Redbooks

ibm.com/redbooks

Integrating IBM PureApplication System into an Existing Data Center

Alan E Booth

Erhan Ekici

Venkata Gadepalli

Rajeev Gandhi

Addison Goering

Ivan Pryanichnikov

Vincent T Tran

Hendrik Van Run

Cloud

PureSystems

IBM

Redbooks

International Technical Support Organization

Integrating IBM PureApplication System into an Existing Data Center

November 2015

SG24-8285-00

Note: Before using this information and the product it supports, read the information in “Notices” on page ix.

First Edition (November 2015)

This edition applies to Version 2.1.0 of IBM PureApplication System.

© Copyright International Business Machines Corporation 2015. All rights reserved.
Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices ix
Trademarks x

IBM Redbooks promotions xi

Preface xiii
Authors xiii
Now you can become a published author, too! xvi
Comments welcome xvii
Stay connected to IBM Redbooks xvii

Chapter 1. Overview 1
1.1 Product overview 2
1.1.1 PureApplication System 2
1.1.2 PureApplication Software 3
1.2 Pre-engineering 3
1.2.1 Pre-engineering is more than patterns of expertise 3
1.2.2 Beware of configuration drift 4
1.3 Tasks, roles, and responsibilities 4
1.3.1 A cross-functional team is the solution 4
1.4 Roadmap for this publication 7

Chapter 2. Data center planning for an IBM PureApplication installation 13
2.1 PureApplication System requirements 14
2.1.1 PureApplication System classes 14
2.1.2 Hardware requirements: PureApplication System x86 Mini configurations 14
2.1.3 Hardware requirements: PureApplication System x86 Enterprise configurations . 18
2.1.4 Hardware requirements: PureApplication System POWER configurations 24
2.1.5 Site-readiness planning 26
2.1.6 Networking overview 26
2.1.7 Cloud groups 27
2.1.8 Environment profiles 28
2.1.9 IP groups 29
2.1.10 PureApplication System internal network overview 29
2.1.11 Ports that are required for a PureApplication System connection 29
2.1.12 Domain Name System server 30
2.1.13 Network time protocol server 30
2.1.14 Networking topologies 30
2.2 PureApplication Software requirements 31
2.2.1 Configuration of the VMware vCenter server and ESXi 32
2.2.2 Installation requirements 32
2.2.3 PureSystems Manager requirements 32
2.2.4 Firewall 34
2.2.5 Compute node requirements 35
2.2.6 Network requirements 36
2.3 How PureApplication VMs connect to the network 36

Chapter 3. IBM PureApplication installation 41
3.1 PureApplication Software in the data center 42

© Copyright IBM Corp. 2015. All rights reserved.

3.1.1 Obtaining PureApplication Software 42
3.1.2 Setting up and installing PureApplication Software. 43
3.1.3 Configuring the vCenter environment 49
3.1.4 Configuring the PureApplication Software connection to vCenter 58
3.1.5 Configuring the PureApplication Software system settings. 58
3.1.6 Configuring and allocating cloud resources 61
3.2 PureApplication System in the data center 63
3.2.1 Physical inspection of the rack 63
3.2.2 Powering on the rack 64
3.2.3 System diagnostic tests 65
3.2.4 Physical network connection between the appliance and the data center 65
3.2.5 Test deployment 66

Chapter 4. IBM PureApplication multi-system environment 69
4.1 Introduction to the multi-system environment on the PureApplication platform 70
4.1.1 Management domain 71
4.1.2 Deployment subdomain 73
4.1.3 Externally managed cloud groups and environment profiles 74
4.2 Configuring a multi-system environment with PureApplication System 74
4.2.1 Rules for a multi-system environment. 74
4.2.2 Required user permissions 75
4.2.3 Management VLAN for an externally managed cloud group 75
4.2.4 Configuring additional IP addresses for cloud management. 77
4.2.5 Creating the management domain 79
4.2.6 Creating the deployment subdomain 79
4.2.7 iSCSI tiebreaker 80
4.2.8 Configuring cloud resources for a multi-system environment 83
4.2.9 Upgrade considerations in the multi-system environment 90
4.2.10 Security 92
4.3 Using a PureApplication System multi-system environment 93
4.3.1 Managing catalog resources 94
4.3.2 Shared services 96
4.3.3 Deploying a pattern by using a multi-system configuration. 97
4.3.4 Security considerations 100
4.3.5 Availability, scaling, and recovery considerations 100

Chapter 5. Storage 103
5.1 Storage principles 104
5.1.1 What a storage volume is 104
5.1.2 Base OS images 104
5.1.3 What block storage and file-level storage are 106
5.1.4 How block storage and file-level storage are used 107
5.1.5 Block storage and file-level storage in PureApplication System 107
5.1.6 Block storage and file-level storage in PureApplication Software 111
5.2 SAN Virtualization Controller for external storage. 112
5.2.1 When to use a SAN Virtualization Controller 112
5.3 Adding additional storage to the PureApplication System 112
5.4 IBM General Parallel File System 113
5.4.1 GPFS topologies. 113
5.4.2 Shared service for GPFS 115
5.4.3 GPFS file systems and file sets 115

Chapter 6. Monitoring 117
6.1 PureApplication System Monitoring overview 118

6.2 System event monitoring by using SNMP 119
6.3 System Monitoring shared services 121
6.3.1 System Monitoring 122
6.3.2 Middleware-specific System Monitoring services 124
6.3.3 Using the PureApplication System Agent 126
6.3.4 Deploying the System Monitoring shared service 127
6.3.5 Deploying other System Monitoring shared services 130
6.3.6 Using System Monitoring shared services 132
6.3.7 Situations in the System Monitoring shared service 143
6.3.8 Forwarding events from System Monitoring shared service 149
6.4 Using Optim for performance monitoring of DB2 151

Chapter 7. Operating system maintenance 155
7.1 Introduction 156
7.2 OS maintenance by using Red Hat Satellite Server 157
7.2.1 Overview 157
7.2.2 Preparing to deploy Red Hat Satellite Server on PureApplication System 158
7.2.3 Deploying Red Hat Satellite Server on PureApplication System 164
7.2.4 Configuring Red Hat Satellite Server on PureApplication System 167
7.2.5 Understanding the scope of the Red Hat Satellite Service shared service 175
7.2.6 Deploying the Red Hat Satellite service shared service 177
7.2.7 Using Red Hat Satellite Server integration on PureApplication System 179
7.3 IBM AIX OS maintenance by using IBM Endpoint Manager 195

Chapter 8. IBM middleware maintenance 205
8.1 IBM Installation Manager Repository 206
8.1.1 Default internal IBM Installation Manager Repository 206
8.1.2 Integrating with an external IBM Installation Manager Repository 207
8.2 Applying fix packs to Virtual System Pattern Instances 213
8.2.1 IBM WebSphere Application Server V8.5.5 213
8.2.2 IBM DB2 10.5 226
8.2.3 IBM MQ V8.0 233
8.3 Including Fix Packs in Virtual System Patterns 237
8.3.1 IBM WebSphere Application Server V8.5.5 238
8.3.2 DB2 10.5 238
8.3.3 IBM MQ 8.0 240

Chapter 9. System and integrated components maintenance 241
9.1 Maintenance overview 242
9.2 Planning system update 245
9.2.1 Deciding on a maintenance path 245
9.2.2 Obtaining the system update procedure 246
9.2.3 Taking backups of the system 246
9.2.4 Maintenance in a multi-system environment 246
9.2.5 Downloading the required software packages 248
9.2.6 Importing the system fix pack 250
9.2.7 Time and workload continuity planning 255
9.3 Preparing the system for an update 258
9.3.1 Activating maintenance mode 258
9.3.2 Disabling the Service and Support Manager 259
9.3.3 Checking the system health status 260
9.4 Applying the system update 265
9.4.1 Applying a system update by using the system console 266
9.4.2 Applying a system update by using the pure.cli 267

9.4.3 Applying compute node updates 268
9.5 Applying an integrated components update (group fix) 270
9.5.1 Dependency checks 270
9.5.2 Updating integrated components by using the system console 270
9.5.3 Updating integrated components by using the pure.cli 271
9.5.4 Upgrading deployed pattern type instances 272
9.6 Performing system health status check (post-upgrade) 272
9.7 Maintenance-related preferred practices 273

Chapter 10. Backup and restore 275
10.1 Data categories 276
10.1.1 Management function data 276
10.1.2 Cloud environment data 276
10.1.3 Workload catalog data 277
10.1.4 Workload (pattern instance) data 278
10.1.5 Application data 278
10.1.6 Data categories example 278
10.2 Types of backups 279
10.2.1 Required permissions for backup and restore 279
10.3 Configuring backup functions 280
10.3.1 Creating backup locations 280
10.3.2 Creating backup configurations 282
10.3.3 Creating a component backup configuration 283
10.3.4 Creating a system backup configuration 285
10.4 Performing backups 286
10.4.1 Performing a system backup 286
10.4.2 Performing a component backup 288
10.4.3 Viewing the backup history 290
10.5 Restoring backup data 291
10.5.1 Restoring system data 292
10.5.2 Restoring component data 293
10.5.3 Alternative backup locations to restore component data 295
10.5.4 Troubleshooting 296
10.6 Backing up and restoring by using the CLI or REST API 296
10.7 Backup instances and application data 296

Chapter 11. Security 299
11.1 System management 300
11.1.1 Management system security 301
11.1.2 System console SSL certificates 301
11.2 Resource isolation 305
11.2.1 Cloud groups 306
11.2.2 IP groups 308
11.2.3 Environment profiles 309
11.2.4 Access controls 309
11.3 System security 310
11.3.1 Disk-based encryption 310
11.3.2 System-level access 310
11.3.3 Roles and permissions 312
11.3.4 User account policies 313
11.3.5 User and User Group management 314
11.3.6 Directory server integration (LDAP) 315
11.3.7 Secure LDAP communication 317

11.4 Workload security 319
11.4.1 LDAP user authentication for OS 320
11.4.2 Patch management 320
11.5 Security monitoring 321
11.5.1 Security and administrative event auditing 321
11.5.2 Enabling security events 324
11.5.3 Forwarding security events 327
11.6 Special situations: Peeling back the covers 328
11.6.1 Network topology 328
11.6.2 Network switches 328
11.6.3 SAN switches 331

Chapter 12. Service and Support Manager 333
12.1 Service and Support Manager overview 334
12.2 Enabling Service and Support Manager 334
12.2.1 Access through a corporate firewall 335
12.2.2 Configuring Service and Support Manager 337
12.2.3 Verifying the Service and Support Manager configuration 341
12.3 Service and Support Manager workflow 343
12.4 Service tickets (Call Home events) 344
12.5 Working with service tickets 345
12.5.1 Add Service Ticket feature 347
12.5.2 Add File to a Service Ticket feature 349
12.5.3 Add Collection Set to a Service Ticket feature 350
12.5.4 Uploading, downloading, and deleting service ticket files 352
12.5.5 Add Local Note feature 352
12.6 List of Call Home events 353

Chapter 13. IBM PureApplication System command-line interface and REST API . 355
13.1 Pure.cli 356
13.1.1 Pure.cli basics 356
13.1.2 Creating and deleting objects 359
13.1.3 Listing objects in a collection 361
13.1.4 Iterating through a collection 361
13.1.5 Finding an object instance by exact name 363
13.1.6 Setting or updating an object attribute 364
13.1.7 Building object collections and instances from REST JSON responses 365
13.2 PureApplication System REST API 365
13.2.1 Anatomy of a RESTful HTTP request 366
13.2.2 HTTP verbs 368
13.3 CLI recipes 370
13.3.1 Importing and loading group fixes 370
13.3.2 Creating users and groups from a CSV list 371
13.3.3 Cleaning old deployments (non-production environments only) 372
13.3.4 Getting full instance reports while filtering by virtual machine name 373
13.3.5 Running a script package 374
13.3.6 Finding VMs in a particular cloud group by using a particular image 376
13.3.7 Creating IP groups, adding an IP range, and attaching an IP group to a cloud group by using a script 378

Chapter 14. Chef 381
14.1 What is Chef 382
14.2 Chef integration with PureApplication System 382
14.3 Configuring the external Chef server shared service 383

14.4 Sample Chef deployment 383
14.5 Debugging a deployment by using a Chef client on PureApplication System. 386

Related publications 397
Online resources 397
Help from IBM 399

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:
IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785 U.S.A.

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs.

© Copyright IBM Corp. 2015. All rights reserved.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. These and other IBM trademarked terms are marked on their first occurrence in this information with the appropriate symbol (® or ™), indicating US registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at http://www.ibm.com/legal/copytrade.shtml

The following terms are trademarks of the International Business Machines Corporation in the United States, other countries, or both:

AIX®
Bluemix™
DB2®
developerWorks®
FlashCopy®
GPFS™
IBM®
IBM Flex System®
Optim™
Passport Advantage®
POWER®
PowerVM®
PureApplication®
PureData®
PureFlex®
PureSystems®
Redbooks®
Redbooks (logo) ®
Storwize®
Tivoli®
Tivoli Enterprise Console®
WebSphere®

The following terms are trademarks of other companies:

Intel, Intel logo, Intel Inside logo, and Intel Centrino logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

UNIX is a registered trademark of The Open Group in the United States and other countries.

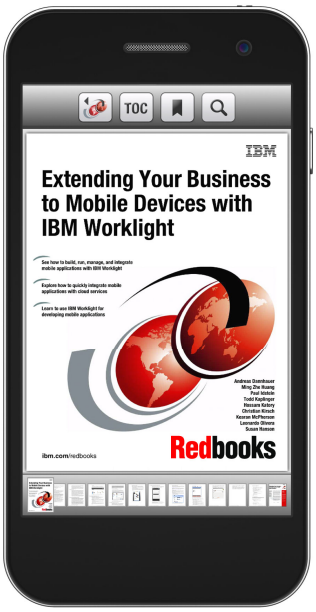
Other company, product, or service names may be trademarks or service marks of others.

Find and read thousands of IBM Redbooks publications

- Search, bookmark, save and organize favorites
- Get up-to-the-minute Redbooks news and announcements
- Link to the latest Redbooks blogs and videos

Get the latest version of the Redbooks Mobile App

iOS **Download Now** Android



Promote your business in an IBM Redbooks publication

Place a Sponsorship Promotion in an IBM® Redbooks® publication, featuring your business or solution with a link to your web site.

Qualified IBM Business Partners may place a full page promotion in the most popular Redbooks publications. Imagine the power of being seen by users who download millions of Redbooks publications each year!

ibm.com/Redbooks
About Redbooks → Business Partner Programs

THIS PAGE INTENTIONALLY LEFT BLANK

Preface

This IBM® Redbooks® publication helps you with the integration of IBM PureApplication® System and IBM PureApplication Software into an existing data center. This publication describes certain scenarios that are considered critical (based on IBM client experiences) for a successful implementation of PureApplication Software or PureApplication System into an existing data center. It covers the planning, installation, and configuration of both PureApplication System and PureApplication Software.

Both PureApplication System and PureApplication Software offer on-premises solutions that use proven patterns to extend your applications, reduce cost and complexity, and ease management.

This book is useful for solution specialists, system or software architects, and the IT teams who need more in-depth knowledge about the integration of PureApplication System and PureApplication Software.

Authors

This book was produced by a team of specialists from around the world working at the International Technical Support Organization, Raleigh Center.

Alan E Booth is a Senior Software Engineer with IBM Software Services for Middleware. He has more than 16 years of experience in the IT field working as a software developer, consultant, and manager. Alan is passionate about customer service and founded the PureStart Program and Technical Account Manager team for PureApplication System.

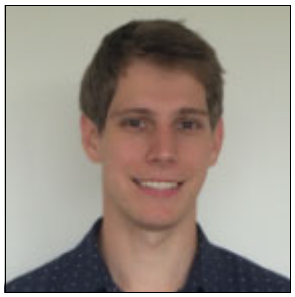
Erhan Ekici is a master certified IT Specialist and works as a technical sales professional at IBM Turkey. He has 15 years of experience in the telecommunication and IT fields. His industry focus is banking and telecommunications, and his technical focus is cloud computing. He holds a master degree in business administration from Istanbul Bilgi University and a Bachelor of Science degree in electronics and communications engineering from Istanbul Technical University.

© Copyright IBM Corp. 2015. All rights reserved.

Venkata Gadepalli is a member of the IBM Software Services for IBM WebSphere® team. He has more than 16 years of experience in the IT field and has been involved in client engagements involving the WebSphere family of products. His main area of interest is working with first-of-a-kind engagements that involve IBM PureApplication System. Vishy has written numerous papers that have been published both within and outside of IBM. He also co-authored the first and second editions of the *IBM WebSphere Portal Primer*, IBM Press, 2003 and 2005, ISBN 1931182132 and ISBN 9781931182232.

Rajeev Gandhi is a Senior Technical Staff Member in the IBM Software Services for WebSphere team. He has more than 28 years of experience in the IT field and has been involved in client engagements working with many of the early releases of IBM middleware products. His main area of focus is IBM PureApplication System. He had been the lead developer on the enablement team for IBM PureApplication System, and works with clients to help them adopt IBM PureApplication Systems. He has presented at several IBM and external conferences, and written articles for IBM developerWorks®. Rajeev holds a Master of Science degree in computer science from the University of Connecticut.

Addison Goering is a Certified IT Specialist with the WebSphere Education team. His main specialty is the design, development, and delivery of courses in the WebSphere product family. He has developed and delivered courses ranging from webinars to week-long workshops on products such as WebSphere ESB, IBM Workload Deployer, WebSphere Application Server, WebSphere Business Services Fabric, and WebSphere BPM. He is the lead developer on the WebSphere Education team that is developing education for IBM PureApplication System. Addison holds a Bachelor of Science degree in education from Keene State College in New Hampshire, a mainframe certification from DePaul University in Chicago, and several certifications from IBM.



Ivan Pryanichnikov is a certified Technical Sales Specialist on the IBM Systems Middleware team in the CEE region, and is based in Moscow, Russia. He has more than 5 years of experience at IBM. Ivan worked on the PureApplication projects for key customers across his region since the announcement of PureApplication System. He holds a degree in control and informatics in technical systems from Moscow Aviation Institute (State University of Aerospace Technologies) and several certifications from IBM.

Vincent T Tran is a software engineer with IBM working on the customer support team for the PureApplication products. Vincent is an insatiable learner and has technical certifications from IBM, Cisco, and VMware. Vincent recently completed his master degree in computer science from the University of North Carolina at Wilmington.

Hendrik Van Run is an Executive IT Specialist within IBM Software Services for WebSphere (ISSW) consulting practice in Hursley. He is the European technical lead for PureApplication System and helps clients evaluate and implement solutions by using this platform. He has advised many clients on issues, such as high availability, performance, and scalability across the WebSphere Application Server product families. Hendrik also has a strong background in performance analysis and optimization within the enterprise. He holds a master degree in physics, is co-author of a number of IBM Redbooks publications, and frequently speaks at conferences.

This project was led by:

- Margaret Ticknor a Redbooks Project Leader in the Raleigh Center. She primarily leads projects about WebSphere products and IBM PureApplication System. Before joining the ITSO, Margaret worked as an IT specialist in Endicott, NY. Margaret attended the Computer Science program at State University of New York at Binghamton.

Thanks to the following people for their support of this project:

- Deana Coble, IBM Redbooks Technical Writer
- Karen Lawrence, IBM Redbooks Technical Writer
- Ann Lund, IBM Redbooks Residency Administrator
- Thomas Edison, IBM Redbooks Graphics Editor
- Wade Wallace, IBM Redbooks Editor

Thanks to the following people for their contributions to this project:

Robert H. Bunn, PureApplication Support Engineer IBM AIM Technical Support and Customer Service, for his review and guidance with Service and Support Manager (Call Home).

Don Carr, IBM US Internal and External Customer Support, for providing help and guidance with the data center planning and system installation.

Wei Lung Chan, WebSphere Cloud Development, for reviewing and providing assistance with the storage section.

Ching-Yun Chao, Ph.D., CISSP Cloud APS Integration Architect IBM Master Inventor, for providing help with security and trust in IBM PureApplication System.

Noel Colon, Technical Account Manager PureApplication System PureStart, for reviewing and providing help with pure.cli and the REST API with IBM PureApplication System.

Jean Gao, Advisory Software Engineer, for providing help with Integrating Chef with IBM PureApplication System.

Yiwen Huang, PureApplication System, IBM China Development Lab (CDL), for providing assistance with system maintenance.

Jianfeng Kong, Staff Software Engineer WebSphere Cloud Initiatives Development, for providing help with Integrating Chef with IBM PureApplication System.

Qi Liu, Advisory Software Engineer, for providing help with integrating Chef with IBM PureApplication System.

Zhao Liu, Advisory Software Engineer, for providing help with integrating Chef with IBM PureApplication System.

Jose L Lopez, Software Developer, for reviewing and providing enhancements to the pure.cli and REST API parts of this publication.

Jeffrey Luo, Test Engineer PureApplication System, for providing help and guidance with the IBM Endpoint Manager (IEM) and operating system (OS) maintenance.

Scott Moonen, Senior Software Engineer, PureApplication System, for providing help with implementing multi-system management and deployment with IBM PureApplication System.

Daniel Mullen, IBM PowerVM® Cloud Development, for his help reviewing and enhancing Chapter 2, “Data center planning for an IBM PureApplication installation” on page 13.

Jim Robbins, STSM and lead developer for IBM PureApplication System, for providing infrastructure help in setting the PureApplication Systems and managing the systems that were used to validate our scenarios, and helping us answer key architecture questions that are related to scenarios and the new functions.

Eduardo Tanaka, Software Test Engineer, IBM Tivoli® Systems Management, for reviewing and providing help with pure.cli and the REST API.

Yue Wang, Advisory Software Engineer, for providing help with integrating Chef with IBM PureApplication System.

Bobby Woolf, IBM Bluemix™ Technical Enablement Specialist, for providing guidance and help with network design for IBM PureApplication System, and about how workloads connect to the network.

Li Yi, Senior Technical Staff Member, for providing help with integrating Chef with IBM PureApplication System.

Now you can become a published author, too!

Here’s an opportunity to spotlight your skills, grow your career, and become a published author—all at the same time! Join an ITSO residency project and help write a book in your area of expertise, while honing your experience using leading-edge technologies. Your efforts will help to increase product acceptance and customer satisfaction, as you expand your network of technical contacts and relationships. Residencies run from two to six weeks in length, and you can participate either in person or as a remote resident working from your home base.

Find out more about the residency program, browse the residency index, and apply online at:

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

We want our books to be as helpful as possible. Send us your comments about this book or other IBM Redbooks publications in one of the following ways:

- Use the online **Contact us** review Redbooks form found at:

  ibm.com/redbooks

- Send your comments in an email to:

  redbooks@us.ibm.com

- Mail your comments to:

  IBM Corporation, International Technical Support Organization
  Dept. HYTD Mail Station P099
  2455 South Road
  Poughkeepsie, NY 12601-5400

Stay connected to IBM Redbooks

- Find us on Facebook:

  http://www.facebook.com/IBMRedbooks

- Follow us on Twitter:

  http://twitter.com/ibmredbooks

- Look for us on LinkedIn:

  http://www.linkedin.com/groups?home=&gid=2130806

- Explore new Redbooks publications, residencies, and workshops with the IBM Redbooks weekly newsletter:

  https://www.redbooks.ibm.com/Redbooks.nsf/subscribe?OpenForm

- Stay current on recent Redbooks publications with RSS Feeds:

  http://www.redbooks.ibm.com/rss.html

1

Overview

IBM PureApplication System and IBM PureApplication Software provide multiple benefits to an enterprise. The decision to integrate either of these products into an existing data center can be a disruptive event. Physical infrastructure, networking, storage, middleware, and systems are all affected in some way.

This publication provides guidance about the integration process so that there is as little disruption as possible. The authors of this book are subject matter experts (SMEs) who have worked with clients during their integration process. The collective experience of these SMEs is captured in this publication.

This chapter also provides an overview of the products and challenges that must be managed when introducing PureApplication into a data center.

This chapter covers the following topics:

- Product overview
- Pre-engineering
- Tasks, roles, and responsibilities
- Roadmap for this publication

© Copyright IBM Corp. 2015. All rights reserved.

1.1 Product overview

The descriptions in this publication are limited to the integration of PureApplication System and PureApplication Software into an existing data center. There is no description of integrating PureApplication Service (another component of PureApplication System) into a data center.

The IBM PureSystems® family of systems accelerates the deployment of enterprise applications and analytic platforms. For more information about the entire family of systems, go to the following website:

http://www.ibm.com/ibm/puresystems/us/en/

There are four subfamilies in the PureSystems family of products:

- PureApplication (including PureApplication System and PureApplication Software)
- IBM PureData®
- IBM PureFlex® System
- Hybrid Cloud

1.1.1 PureApplication System

PureApplication System is often referred to as a *cloud on a rack*. PureApplication System provides all the elements of a cloud in one system: compute nodes, networking, storage, and system management. Figure 1-1 shows a simple view of a PureApplication System. It is designed around the following core principles:

- Built-in expertise that captures what experts do, from infrastructure patterns to application patterns.
- Integration of hardware and software by design in a workload optimized system.
- Simplified experience that makes each part of the lifecycle easier, which means integrated management of the entire system and an open infrastructure of optimized solutions.

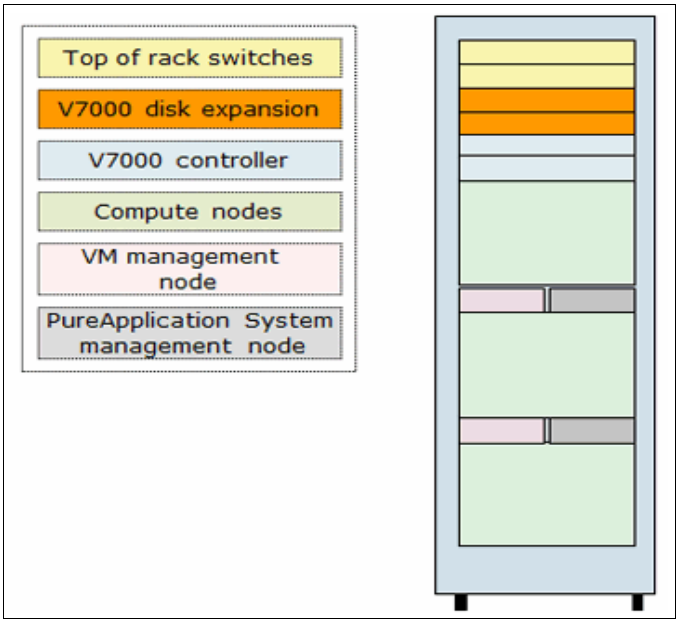


Figure 1-1 PureApplication System hardware overview

1.1.2 PureApplication Software

PureApplication Software is a cloud application platform that accelerates and simplifies the repeatable deployment and lifecycle management tasks of enterprise middleware workloads, such as:

- IBM WebSphere Application Server
- IBM Business Process Manager
- Mobile
- Portal
- Commerce
- Analytics

You can choose to use PureApplication capabilities on an integrated hardware and software PureApplication System, or with PureApplication Software on your x86 hardware.

You can use PureApplication Software to design the infrastructure to your unique specifications. Infrastructure planning and configuration work must be completed before the installation of the PureApplication Software and must meet all prerequisites. Although you must adhere to the prerequisites, PureApplication Software provides flexible compute and storage characteristics so that you can define an infrastructure that meets your unique use cases. With PureApplication Software, you bring your own licenses for OS, hypervisor, and middleware.

1.2 Pre-engineering

PureApplication System (and to some degree, PureApplication Software), along with the IBM patterns of expertise, are often referred to as a *pre-engineered* product, which means that the PureApplication System already adheres to IBM and industry-preferred practices for network design, storage implementation, file system organization, and other components, all the way up to the middleware configuration.

However, pre-engineered systems require thinking, designing, and planning to integrate into a data center.

When faced with the complexity of implementing a PureApplication project in the real world, it is tempting to take shortcuts. Although shortcuts can lead to much improved time to value (TTV), failure to plan for system lifecycle and to understand fully how things must change operationally, can greatly increase the total cost of ownership (TCO) for the project.

1.2.1 Pre-engineering is more than patterns of expertise

Pre-engineering applies to more than patterns. To get the most out of your investment in PureApplication System, you must modernize your approach to the whole data center infrastructure. This approach means pre-allocating DNS and firewall rules, and opening APIs for enrollment systems. This approach means automating everything that is involved. If you must log in to a virtual machine (VM) that is deployed to make it usable, then you have deficiencies in your pattern.

Pre-engineering requires more than downloading a pattern. Patterns of expertise represent the IBM view of preferred practices as they apply to most users. This does not mean that a pattern available from IBM is correct for every situation. Expect that some customization is required to make the default patterns work in your enterprise.

1.2.2 Beware of configuration drift

When you initially have PureApplication System in your data center, there is a temptation to get started right away and deploy as many VMs as possible. Consider this: PureApplication System and PureApplication Software can deploy VMs quickly. After the VMs are deployed, they are released to the application teams to manage as their own. The application teams become comfortable, and rather than providing the deployment team with environment requirements, the deployment team gives the application teams Secure Shell (SSH) access.

Use SSH access for emergency purposes only. The more access an application team has to their VMs, the more *state* is introduced. A principle to consider is if the application teams must access the system by using SSH, then there must be an undefined or uncaptured requirement that must be implemented as part of the pattern. If SSH is a regular part of your PureApplication administrator's duties, then you should reexamine them so you can see the true value of patterns.

1.3 Tasks, roles, and responsibilities

When an organization brings in a PureApplication System or PureApplication Software, it is a disruptive event. PureApplication System can vastly simplify or eliminate many of the routine tasks that consume the time of development, operations, and system administrator staff. Freedom from these repetitive tasks creates a unique opportunity for businesses to leverage fully the technical skills of its organization by assigning them to higher value tasks.

PureApplication System uses specialist roles that an IT organization is already familiar with, such as security, hardware, storage, and cloud administration. It also introduces new concepts and roles, such as patterns and workload administration, which require a combination of generalist skill sets that are likely already found in the IT organization.

1.3.1 A cross-functional team is the solution

PureApplication System combines the major components of a large-scale, virtualized, and distributed system into one package. Tasks that used to take teams of specialists days or weeks to complete can be accomplished by a wider range of people in much less time. This is accomplished by capturing the deep skills of IT specialists inside the system and into the deployable patterns. The knowledge of how to tune a complex piece of middleware or how to scale an application for a specific workload can be encapsulated so that anyone with lesser skill sets (generalist) in the organization can take advantage of it.

PureApplication System allows the organization to distribute roles that cover a broader range of tasks. For example, WebSphere Application Server administrators no longer must spend time tuning their application servers because that task is managed by policies that are defined in simple business terms within a pattern. Administrators can instead branch out and apply their skills to assist application developers and architects who are implementing their patterns. A cross-functional team provides a vehicle for operations and support staff who aspire to expand their expertise and have greater overall influence on an end-to-end application.

As a rule, three people (three roles) should be identified on a cross-functional team as focal points for the integration of PureApplication System or PureApplication Software:

- Executive sponsor
- System administrator
- Workload administrator

Executive Sponsor

Bringing a disruptive technology into an organization can be painful. The Executive Sponsor leads the effort to help the organization migrate and embrace the integration of PureApplication System into the data center. The Executive Sponsor should help to identify and revisit roles, responsibilities, and tasks that have evolved through the life span of the organization. Roles, responsibilities, and tasks do not need to be reinvented, but changes are needed to reap the full benefit of a PureApplication platform.

System Administrator

This person is ultimately accountable for all of the system-level concepts on the rack. *System level* is normally considered the infrastructure from the hypervisor down. The System Administrator does not necessarily perform all of the tasks that are needed, but that person is the focal point for delegating tasks and ensuring that all necessary roles and responsibilities on the system side are fulfilled.

Here are some of the system-level concepts:

- Cloud group design
- IP group definition
- User and group management
- Integration with corporate event monitoring, such as by simple network management protocol (SNMP)
- System-level backup and restore

Workload Administrator

The Workload Administrator is responsible for all of the workload-level concepts on the rack. *Workload level* typically refers to the platform from the middleware up. Similar to the sysTem Administrator, the Workload Administrator does not necessarily perform all tasks, but does bear ultimate responsibility.

Here are some of the workload-level concepts:

- All patterns
- Deployed workloads
- Script packages
- Shared services
- OS patch management

There is not always a clear separation of duties between the System Administrator and the Workload Administrator, so it is critical that these roles work closely together. One gray area between the roles concerns the OS on which deployed workloads run. This area combines both workload and system concepts, and thus must be a joint effort.

Other new PureApplication roles

There are two other roles that should be considered, but are not part of the core roles when integrating PureApplication System or PureApplication Software in your data center. These roles are included here for completeness.

Pattern Architect

Persons in this role define and design the patterns that are needed for all applications. The pattern architect engages with application architects to enable applications in defined patterns.

Pattern Developer

The person in this role works closely with the pattern architect and is ultimately the pattern owner. The pattern developer adapts existing patterns from IBM and IBM Business Partners, creates Virtual System Patterns (VSPs), and creates plug-ins for virtual applications.

Existing IT roles and how they fit in with disruptive technology

What about other areas that are critical to PureApplication System, such as IT security, networking, and storage? Because PureApplication is a disruptive technology, the organization can experience changes to the tasks that are assigned to these roles.

Certain roles must change because of the benefits of the automation that is provided with PureApplication System and PureApplication Software. The reduction in work that is realized by your team allows them to begin working through existing IT backlogs. This reduced effort that is allocated for common tasks also enables your staff to retrain and obtain new skill sets for addressing new business opportunities.

Rack and server operations

In the past, distributed systems were purchased piecemeal and arrived in multiple boxes. Someone was responsible for unpacking, and then racking, stacking, and cabling the servers together. However, PureApplication System ships as a single, pre-assembled, pre-cabled, pre-integrated, pre-tested system. An IBM representative unpacks the system in your data center, connects it to your network and power, starts the system, validates its functions, and configures it to recognize your network.

PureApplication Software is deployed and configured on your infrastructure according to specifications provided by IBM. Therefore, with PureApplication Software, the function of rack and server operations remains. A team is needed to rack, stack, and cable the servers together before PureApplication Software is installed and configured.

Storage administration

Most organizations have an entire team that is dedicated to managing the storage area network (SAN) infrastructure on which their compute hosts work. The team sets up the physical parts of the SAN, is responsible for all of the SAN and logical unit number (LUN) planning and mapping to connect the SAN infrastructure to the hosts, and performs the host and volume configuration, host mapping, storage pool management, and so on. The team is responsible for applying new driver levels, SAN software levels, and so on.

In PureApplication System, the work of the SAN infrastructure team is encapsulated in the system. However, the team plays an important role in configuring block storage replication and in managing external storage through storage volume controllers (SVCs).

With PureApplication Software, the work of the SAN infrastructure team is necessary for the configuration and integration of block storage volumes.

Some SAN administration tasks must be performed during normal operations of PureApplication System and PureApplication Software. These operational tasks are listed below and can be assigned to a member of the cross-functional team or a member of the SAN infrastructure team:

- Determine the appropriate volume size when a new VM is provisioned.
- Expand volume sizes when needed (for example, when a database is close to expanding beyond its originally available volume size).
- Perform backup and restore operations.

Virtualization infrastructure administration

Many organizations have a team that is responsible for its virtualization infrastructure and administration. This team installs hypervisors, installs the virtualization management infrastructure, connects individual hypervisors into the infrastructure, and then maintains the infrastructure.

With PureApplication System, the virtualization infrastructure team gets to work immediately after initial configuration. The scope of the virtualization team is reduced so that they can now manage more virtualized environments, and with the trend of PureApplication System, the team can possibly take on other duties within the cross-functional team.

When configuring PureApplication Software, the virtualization team is critical to the success of integrating PureApplication Software into a data center. A well-designed virtualization infrastructure is critical to the success of the integration project because PureApplication Software *connects*, in a sense, to the infrastructure during installation. The virtualization infrastructure is also central to the daily operation of PureApplication Software.

Network administration

Any modern distributed system is supported by a mass of network cabling. There is an enormous amount of planning and work that goes into setting up a LAN, with work going into connecting distributed host computers to a SAN, to each other, and to rack-mounted switches. Then, someone must configure all of those physical networks, assigning IP addresses to servers, specifying how those map into VLANs, and so on.

The PureApplication System either eliminates those tasks or reduces the amount of effort that is needed to perform those tasks. There is still some planning that must happen to determine how to set up the internal network in your PureApplication System, but after you make those decisions, the implementation proceeds quickly, and nearly everything you need to specify is done during the 4-hour installation window. Likewise, the maintenance of the internal network is a much simpler operation, requiring far fewer hours than the management of a typical network.

1.4 Roadmap for this publication

This publication describes certain scenarios that are critical to a successful implementation of PureApplication Software or PureApplication System in an existing data center. Not all scenarios can be covered, but the collective customer experience of the team was used as the guiding factor.

The focus of the book is not on features and functions. For more information at that level, see IBM Knowledge Center, found at the following website:

`http://www.ibm.com/support/knowledgecenter/SSL5ES/welcome`

This publication guides you from the perspectives of the roles that are described in 1.3, “Tasks, roles, and responsibilities” on page 4 with the experience that was gained from internal and client engagements.

The roadmap that is shown in Figure 1-2 suggests the most appropriate reader (role) for each chapter. The roadmap is divided into two paths:

- Tasks that are associated with planning and installation
- Tasks that are associated with postinstallation

There are chapters and roles that are aligned with each path in the roadmap. Use this roadmap as a guide to determine which area of content might be most useful to you and the role that you play in the integration.

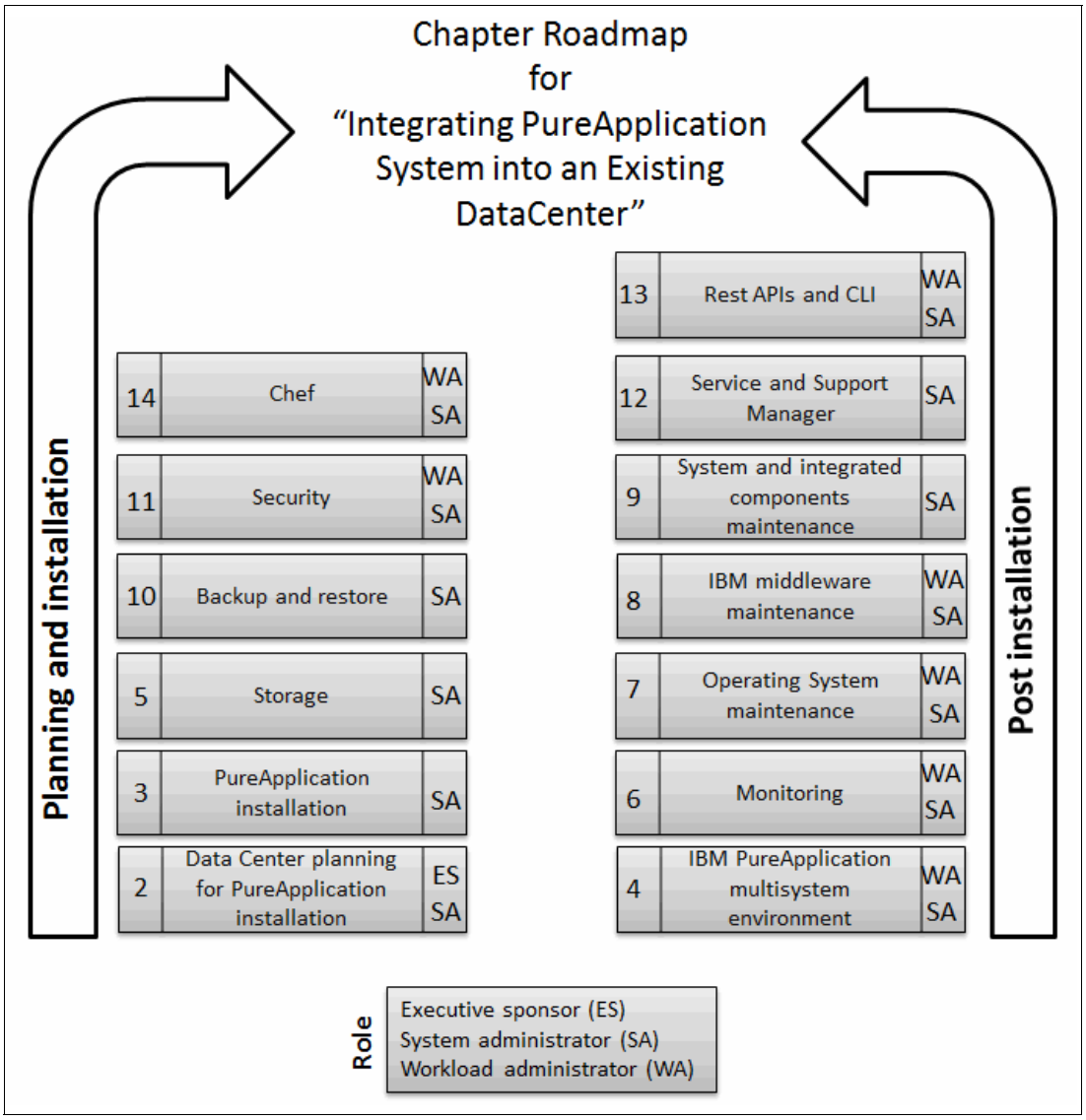


Figure 1-2 Publication roadmap

Data center planning for PureApplication installation

Chapter 2, "Data center planning for an IBM PureApplication installation" on page 13 provides information about data center planning and the requirements for PureApplication System and PureApplication Software installation. Important networking concepts and topologies of PureApplication System and PureApplication Software are described.

Information in this chapter can assist you in planning the installation of PureApplication System and PureApplication Software, and in the integration of these products into an existing environment. The chapter describes management and data interfaces of the system, workload isolation, and other important topics.

Recommended roles: Executive Sponsor and System Administrator

PureApplication installation

Chapter 3, “IBM PureApplication installation” on page 41 provides information about installing PureApplication System and PureApplication Software. The planning and installation requirements for each product are different. They are both on-premises solutions that give you a cloud application platform for end-to-end application lifecycle management. PureApplication System provides a pre-integrated hardware and software platform. PureApplication Software provides a cloud application platform that can run on your hardware.

Recommended role: System Administrator

PureApplication multi-system environment

A multisystem environment expands the benefits of a single system environment. Typically, a large business or one that has more than one geographic location operates in a multi-system environment. This environment consists of two or more systems that are connected to each other physically (by communications equipment) and logically (in a defined relationship, as peer or host to other systems).

Chapter 4, “IBM PureApplication multi-system environment” on page 69 covers multi-rack configuration and deployments with PureApplication System. Topics that are described in this chapter include how to manage catalog content across PureApplication System racks and how to work with multi-target deployments.

Recommended roles: System Administrator and Workload Administrator

Storage

Storage is a key item when considering the integration of PureApplication System or PureApplication Software to a data center. Both PureApplication products support different storage configurations for various architectural reasons.

Chapter 5, “Storage” on page 103 describes basic storage concepts (storage volumes, block storage, and file-level storage) and how these concepts are implemented and used in PureApplication System and PureApplication Software. The chapter also describes external storage options, in particular by using an IBM SAN virtualization controller to connect an existing SAN and adding storage capacity to an IBM Storwize® V7000 storage system.

Recommended role: System Administrator

Monitoring

Both PureApplication System and PureApplication Software provide facilities to perform monitoring of all areas of your system. System, middleware, and hardware monitoring facilities let you gain access to the status, performance, and resource usage of all areas of your integrated solution.

Chapter 6, “Monitoring” on page 117 describes the following topics:

- Event monitoring by using SNMP
- Using the PureApplication agent
- System Monitoring shared services
- Using IBM Optim™ for performance monitoring of IBM DB2®

Recommended roles: System Administrator and Workload Administrator

Operating system maintenance

Operating system maintenance is an ongoing and important task for any enterprise. Chapter 7, “Operating system maintenance” on page 155 describes how to implement RHEL operating system (OS) maintenance on PureApplication. The chapter also describes using IBM Endpoint Manager to perform maintenance on the IBM AIX® operating system.

Recommended role: System Administrator

IBM middleware maintenance

Chapter 8, “IBM middleware maintenance” on page 205 describes how to apply software maintenance to deployed pattern instances, also known as *workloads*, including OS patches and fix packs to IBM middleware. For production workloads, this is an important scenario because simply redeploying a newer version of a pattern is not always wanted or even possible.

Software maintenance can also be included in patterns so that when software is deployed, it comes with more recent OS patches and the IBM middleware fix packs installed. This is sometimes referred to as *pattern maintenance*.

Recommended roles: System Administrator and Workload Administrator

System and integrated components maintenance

Chapter 9, “System and integrated components maintenance” on page 241 describes different aspects of PureApplication System and PureApplication Software maintenance. Sections include information about tools, interfaces, and preferred practices that you can use to maintain and update your PureApplication System and PureApplication Software environment.

The following topics are covered in this chapter:

- Maintenance overview
- Planning system update
- Preparing the system for update
- Applying a system update
- Applying an integrated components update (group fix)
- Maintenance preferred practices

Recommended role: System Administrator

Backup and restore

The PureApplication System backup and restore capabilities provide a simple level of disaster recovery (DR) that you can use if your business continuity requirements are defined as a recovery time objective (RTO) of 48 hours and a recovery point objective (RPO) of 24 hours.

Chapter 10, “Backup and restore” on page 275 describes various types of backups, such as system, component, workload, and application data. PureApplication uses the concepts of backup profiles and backup policies to plan and perform backups. These concepts are explained in this chapter.

Recommended role: System Administrator

Security

Security is a major consideration when introducing systems to a data center. Chapter 11, “Security” on page 299 describes topics that must be considered before the integration of PureApplication System or PureApplication Software. These topics include the following ones:

- Security standards
- System security
- User account policies user management
- Isolation (network and hardware)
- LDAP integration
- Workload security
- Security monitoring
- Log management
- Special scenarios

Recommended roles: Workload Administrator and System Administrator

Service and Support Manager

Service and Support Manager (also known as *Call Home*) is an optional feature of IBM PureApplication System V2.0. PureApplication Software does not have this feature currently. This is an automated service ticket generation process that detects problems based upon high severity hardware or management software events and reports them to IBM Support.

Chapter 12, “Service and Support Manager” on page 333 describes specific details about the Service and Support Manager configuration for IBM PureApplication System.

Recommended role: System Administrator

Command-line interface (pure.cli) and REST API

Chapter 13, “IBM PureApplication System command-line interface and REST API” on page 355 describes the PureApplication command-line interface (pure.cli) and its REST application programming interface (API). The pure.cli is a Jython-like environment for running commands to manage PureApplication System and PureApplication Software. Pure.cli imports two primary packages: admin and deployer.

The admin package contains objects and functions that allow system management of the rack. The deployer package contains resources that are associated with the workload management. The environment enables a high degree of automation by using scripting.

The PureApplication System REST API is a RESTful web service that exposes various PureApplication System (and PureApplication Software) components and resources. Understanding the REST API allows users to manage various components through automation by using any programming languages that support HTTP verbs.

Recommended role: System Administrator and Workload Administrator

Chef

Chef is a configuration manager that allows administrators to automate the building and configuration of servers. PureApplication System and PureApplication Software support the Chef framework. Chapter 14, “Chef” on page 381 describes the integration of Chef with PureApplication System and PureApplication Software.

Recommended role: System Administrator

2

Data center planning for an IBM PureApplication installation

This chapter provides information about data center planning and the requirements for installing IBM PureApplication System and IBM PureApplication Software in an existing data center. It describes the networking concepts and topologies of these products, in addition to the management and data interfaces of the system, workload isolation, and other important topics.

This information helps you to plan for the installation of PureApplication System and PureApplication Software, and to smoothly integrate these solutions into your existing data center.

This chapter covers the following topics:

- PureApplication System requirements
- PureApplication Software requirements
- How PureApplication VMs connect to the network

© Copyright IBM Corp. 2015. All rights reserved.

2.1 PureApplication System requirements

The PureApplication System is a pre-integrated appliance with a wide range of hardware and software components included, and it requires correct placement in the data center. This section describes the primary installation requirements for PureApplication System and helps you to prepare your data center.

2.1.1 PureApplication System classes

All PureApplication System models can be split into two classes: Mini and Enterprise, which are built on Intel or IBM POWER® processors. Table 2-1 provides details.

Table 2-1 PureApplication System classes

Name	CPUs	Storage in TB	Memory, per node in GB	# of chassis
Mini	32, 64, 96, 128	26.4	256	1
Enterprise	32, 64, 96, 128, 160, 192, 224, 320, 384	54.4	512	2

The system can be upgraded within its class in increments of 32 cores. Upgrades to the PureApplication System can be applied without any outage and does not affect running applications.

Upgrades between configurations are performed by IBM representatives.

2.1.2 Hardware requirements: PureApplication System x86 Mini configurations

In this section, you find configurations of PureApplication System Mini racks, including physical dimensions, power requirements, and detailed hardware configurations.

Figure 2-1 shows the hardware configurations of PureApplication System x86 Mini racks.

PureApplication System W2500 'Mini' (x86)		W2500-32	W2500-64	W2500-96	W2500-128
Compute	Processor	8 core, 2.6GHz Intel Ivy Bridge EP processor, 115 W			
	Compute Node	Dual Processor, 16-core, 256GB memory			
	Memory/Compute Node	256GB (8 2x16GB, 1066 MHz, DDR3, LP RDIMMS(1.35V))			
	Network cards/Compute Node	1 x EN4054 – 4 port 10Gb Ethernet Mezz Adapter			
	Fibre cards/Compute Node	1 x FC 3172 - 2 port 8Gb Fibre Channel Mezz Adapter			
	# Compute nodes	2	4	6	8
	# Cores	32	64	96	128
	Memory	0.5 TB	1 TB	1.5 TB	2.0 TB
Storage	Solid State Drive (SSD)	6 x 400GB 2.5 in. SSD (E-MLC)			
	Hard Disk Drive (HDD)	40 x 600 GB 2.5 in. 10k HDD			
	Storage Controller	1 x IBM Storwize V7000 Disk System			
	Storage Expansion	1 x IBM Storwize V7000 Storage Drawer			
	Total Storage SSD	2.4 TB unformatted / 1.6 TB usable			
	Total Storage HDD	24.0 TB unformatted / 21.6 TB usable			
Network	Top of Rack Switches (TOR)	BLADE Network Technologies TOR G8264 switches for Customer Data Center & Rack to Rack communications			
	Transceivers	Choice of: 10GbE Fibre, 1GbE Fibre, 1GbE Copper, Direct Attach Cabling (DAC / Twinax)			
Chassis	Network / Chassis	2 x EN4093 – BNT 10GbE			
	Fibre / Chassis	2 x FC5022– Brocade 48 16GbFC			
	Chassis	1 x Accipiter Chassis			
Power	Power Distribution Unit (PDU)	4 x 30A 1ph - North America 4 x 32A 1ph - International			
Mgmt	Management Nodes	2 x PureSystems Manager (PSM) 2 x Virtualization System Manager (VSM)			
Rack	Rack	2.0 M 19" Enterprise Rack			
Specs	Weight	591 Kg (1300 lb)	605 Kg (1332 lb)	620 Kg (1364 lb)	635 Kg (1396 lb)
	Power	Max: 4.9 kW Typical: 4.3 kW	Max: 5.8 kW Typical: 5.2 kW	Max: 6.8 kW Typical: 6.0 kW	Max: 7.7 kW Typical: 6.8 kW
	Dimensions	Height: 2.015 m (79.3 in) / Depth: 1.130 m (44.5 in) / Width: 0.644 m (25.4 in)			

Figure 2-1 PureApplication System W2500 Mini rack specifications

Figure 2-2 shows the layout of a PureApplication System x86 Mini rack.

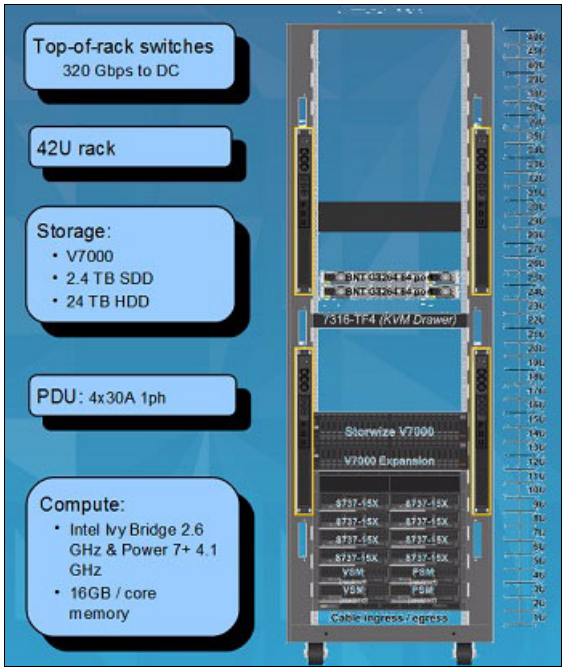


Figure 2-2 Layout of the PureApplication System x86 Mini rack

PureApplication System W2500-32 Mini rack

The PureApplication System W2500-32 (machine type model (MTM) 8283 - 12Y) is the minimum available configuration in the Mini class. As indicated in Table 2-1 on page 14, it has one chassis, 256 GB of memory per compute node, one controller, and one expansion node of IBM Storwize V7000 storage. The physical dimensions and electrical and environmental requirements for this model are provided in Table 2-2, Table 2-3, Table 2-4 on page 17, and Table 2-5 on page 17.

Table 2-2 Dimensions of PureApplication System W2500-32 mini rack

Width	Depth	High	Weight with packaging	Weight without packaging	Rack capacity
25.4 in. (644 mm)	44.5 in. (1130 mm)	79.3 in. (2015 mm)	1450 lbs (659 kg)	1300 lbs (591 kg)	42 EIA units

Table 2-3 Maximum system power[1]

Electrical characteristics	Properties
Power consumption	4.9 kW
kVA	5.0 kVA
Thermal output	16.7 kBTU/hour
Voltage	200 - 240 (nominal) V ac
Frequency	50 Hz or 60 Hz
Phase	1

[1] Power is supplied to one set of power cords only.

Note: Regarding the power consumption in Table 2-3, when planning the electrical system, it is important to use maximum values, which accounts for internal and environmental conditions that result in power consumption increasing beyond typical values.

However, when planning for heat load (see the footnote for Table 2-3 on page 16), you can use the typical values that are specified in Table 2-4.

Table 2-4 Typical system power[2]

Electrical characteristics	Properties
Power consumption	4.3 kW
kVA	4.4 kVA
Thermal output	14.8 kBTU/hour
Voltage	200-240 (nominal) V ac
Frequency	50 or 60 Hz
Phase	1

Table 2-5 Environmental requirements

Environment	Recommended operating	Allowable operating	Non-operating
ASHRAE class		A2	
Airflow direction[a]		Front-to-back	
Temperature		27°C (80.6°F)	
Humidity range	5.5°C (42°F) dew point (DP) to 60% relative humidity (RH) and 15°C (59°F) dew point	-12.0°C (10.4°F) DP and 8% - 85% RH	8% - 85% RH
Maximum dew point		24°C (75°F)	27°C (80°F)
Maximum operating altitude		3050 m (10000 ft)	
Shipping temperature			-40°C - 60°C (-40°F - 140°F)
Shipping relative humidity			5% - 100%

a. Normal cubic feet per minute (CFM) is approximately 1110. Maximum CFM is approximately 2105.

Electromagnetic compatibility compliance

The product is in compliance with the following electromagnetic regulations:

- FCC 47 CFR Part 15
- ICES-003
- EN 55022
- EN 55024

[2] Power is supplied to one set of power cords only.

- CISPR-22
- AS/NZS CISPR-22
- CNS 13438
- VCCI
- GB 9254

Product safety compliance

The product is in compliance with the following safety regulations:

- IEC 60950-1
- UL 60950-1
- CSA 60950-1

Delivery and subsequent transportation of equipment

To accept the new product into your computer room, based on the installation planning information that you provide, you are required to set up the environment with the assistance of an authorized service provider (professional movers or riggers).

Only professional movers or riggers are permitted to transport the equipment. You are responsible for using professional movers or riggers when accepting, relocating, or disposing of equipment.

In anticipation of equipment delivery, prepare the final installation site in advance so that professional movers or riggers can transport the equipment to the computer room at the final installation site. If this is not possible at the time of delivery, make arrangements to have the professional movers or riggers return later to finish the transportation. To perform the required service, the authorized service provider can perform only minimal frame repositioning in the computer room, as needed.

For more information about the requirements for other PureApplication System Mini configurations, such as those listed below, go to the following website:

`https://www.ibm.com/support/knowledgecenter/api/content/nl/en-us/SSCR9A_2.1.0/doc/installguide/planning/ps_sysreq_gen2mini.html`

Here are the machine types for the PureApplication System Mini:

- W2500-32 ‘Mini’ machine type is 8283-12Y
- W2500-64 ‘Mini’ machine type is 8283-14Y
- W2500-96 ‘Mini’ machine type is 8283-16Y
- W2500-128 ‘Mini’ machine type is 8283-18Y

Good to know: The character *1* in the machine type means the number of chassis in the PureApplication System rack.

2.1.3 Hardware requirements: PureApplication System x86 Enterprise configurations

Configurations for the PureApplication System x86 Enterprise are based on the same hardware components as for the PureApplication Mini; however, the x86 Enterprise configurations include a different number of chassis, compute nodes, and storage devices.

Figure 2-3 shows the hardware configurations of PureApplication System x86 Enterprise racks.

PureApplication System W2500 'enterprise' (x86)		W2500-32	W2500-64	W2500-96	W2500-128	W2500-160	W2500-192	W2500-224	W2500-320	W2500-384
Compute	Processor	8 core, 2.6GHz Intel Ivy Bridge EP processor, 115 W								
	Compute Node	Dual Processor, 16-core, 512GB memory								
	Memory/Compute Node	512GB (8 2x32GB, 1066 MHz, DDR3, LP RDIMMS(1.35V))								
	Network cards/Compute Node	1 x EN4054 – 4 port 10Gb Ethernet Mezz Adapter								
	Fibre cards/Compute Node	1 x FC 3172 - 2 port 8Gb Fibre Channel Mezz Adapter								
	# Compute nodes	2	4	6	8	10	12	14	20	24
	# Cores	32	64	96	128	160	192	224	320	384
	Memory	1.0 TB	2.0 TB	3.0 TB	4.0 TB	5.0 TB	6.0 TB	7.0 TB	10.0 TB	12.0 TB
Storage	Solid State Drive (SSD)	16 x 400GB 2.5 in. SSD (E-MLC)								
	Hard Disk Drive (HDD)	80 x 600 GB 2.5 in. 10k HDD								
	Storage Controller	2 x IBM Storwize V7000 Disk System								
	Storage Expansion	2 x IBM Storwize V700 Storage Drawer								
	Total Storage SSD	6.4 TB unformatted / 4.8 TB usable								
	Total Storage HDD	48.0 TB unformatted / 43.2 TB usable								
Network	Top of Rack Switches (TOR)	BLADE Network Technologies TOR G8264 switches for Customer Data Center & Rack to Rack communications								
	Transceivers	Choice of: 10GbE Fibre, 1GbE Fibre, 1GbE Copper, Direct Attach Cabling (DAC / Twinax)								
Chassis	Network / Chassis	2 x EN4093 – BNT 10GbE								
	Fibre / Chassis	2 x FC5022– Brocade 48 16GbFC								
	Chassis	3 x Accipiter Chassis								
Power	Power Distribution Unit (PDU)	4 x 60A 1ph – North America 4 x 32A 3ph - International								
Mgmt	Management Nodes	2 x PureSystems Manager (PSM) 2 x Virtualization System Manager (VSM)								
Rack	Rack	2.0 M 19" Enterprise Rack								
Specs	Weight	717 Kg (1578 lb)	732 Kg (1610 lb)	746 Kg (1642 lb)	761 Kg (1674 lb)	775 Kg (1706 lb)	790 Kg (1738 lb)	805 Kg (1770 lb)	848 Kg (1866 lb)	877 Kg (1930 lb)
	Power	Max: 6.3 kW Typical: 5.5kW	Max: 7.2 kW Typical: 6.4kW	Max: 8.1 kW Typical: 7.2kW	Max: 9.1 kW Typical: 8.0kW	Max: 10.1 kW Typical: 8.9kW	Max: 11.0 kW Typical: 9.6 kW	Max: 11.9 kW Typical: 10.5W	Max: 14.8 kW Typical: 13.0 kW	Max: 16.6 kW Typ.: 14.7 kW
	Dimensions	Height: 2.015 m (79.3 in) / Depth: 1.130 m (44.5 in) / Width: 0.644 m (25.4 in)								

Figure 2-3 PureApplication System W2500 Enterprise rack specifications

Figure 2-4 shows the layout of the PureApplication System x86 Enterprise rack.

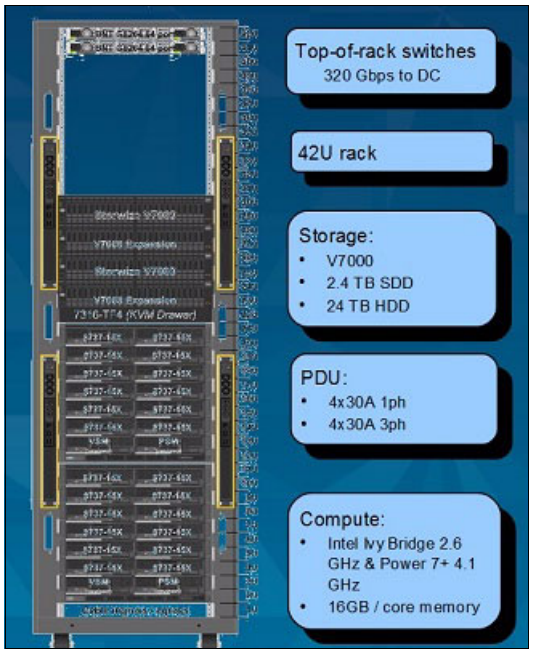


Figure 2-4 Layout of the PureApplication System x86 Enterprise rack

PureApplication System W2500-32 Enterprise

The PureApplication System W2500-32 (machine type model (MTM) 8283 - 2A3) is the minimum available configuration for the Enterprise class of PureApplication System. This model includes two chassis, 512 GB memory per compute node, two controllers, and two expansion nodes of IBM Storwize V7000 storage. The physical dimensions and electrical and environmental requirements for this model are provided in Table 2-6, Table 2-7, Table 2-8 on page 21, and Table 2-9 on page 22.

Table 2-6 Dimensions of the PureApplication System W2500-32 Enterprise rack

Width	Depth	High	Weight with packaging	Weight without packaging	Rack capacity
25.4 in (644 mm)	44.5 in (1130 mm)	79.3 in (2015 mm)	1728 lbs (785 kg)	1578 lbs (717 kg)	42 EIA units

Table 2-7 Maximum system power[3]

Electrical characteristics	Properties
Power consumption	8.0 kW
kVA	8.1 kVA
Thermal output	27.1 kBTU/hour
Voltage	200 - 240 (nominal) V ac
Frequency	50 Hz or 60 Hz

[3] Power is supplied to one set of power cords only.

Electrical characteristics	Properties
Phase	North America: 1P 60A (feature code 6492) Europe: 3P 32A (feature code 6489) Australia/New Zealand: 3P 32A (feature code 6667)

Note: Regarding the power consumption in Table 2-7, when planning the electrical system, it is important to use maximum values, which accounts for internal and environmental conditions that result in power consumption increasing beyond typical values.

However, when planning for heat load (see footnote for Table 2-3 on page 16), you can use the typical values that are specified in Table 2-7 on page 20.

Table 2-8 Typical system power[4]

Electrical characteristics	Properties
Power consumption	7.0 kW
kVA	7.2 kVA
Thermal output	24.0 kBTU/hour
Voltage	200 - 240 (nominal) V ac
Frequency	50 or 60 Hz
Phase	North America: 1P 60A (feature code 6492) Europe: 3P 32A (feature code 6489) Australia/New Zealand: 3P 32A (feature code 6667)

[4] Power is supplied to one set of power cords only.

Table 2-9 Environmental requirements

Environment	Recommended operating	Allowable operating	Non-operating
ASHRAE class		A2	
Airflow direction[a]		Front-to-back	
Temperature		27°C (80.6°F)	
Humidity range	5.5°C (42°F) dew point (DP) to 60% relative humidity (RH) and 15°C (59°F) dew point	-12.0°C (10.4°F) DP and 8% - 85% RH	8% - 85% RH
Maximum dew point		24°C (75°F)	27°C (80°F)
Maximum operating altitude		3050 m (10000 ft)	
Shipping temperature			-40°C - 60°C (-40°F - 140°F)
Shipping relative humidity			5% - 100%

a. Normal cubic feet per minute (CFM) is approximately 1110. Maximum CFM is approximately 2105.

Electromagnetic compatibility compliance

The product is in compliance with the following electromagnetic regulations:

- FCC 47 CFR Part 15
- ICES-003
- EN 55022
- EN 55024
- CISPR-22
- AS/NZS CISPR-22
- CNS 13438
- VCCI
- GB 9254

Product safety compliance

The product is in compliance with the following safety regulations:

- IEC 60950-1
- UL 60950-1
- CSA 60950-1

Delivery and subsequent transportation of the equipment

To accept the new product into your computer room, based on the installation planning information that you provide, you are required to set up the environment with the assistance of an authorized service provider (professional movers or riggers).

Only professional movers or riggers are permitted to transport the equipment. You are responsible for using professional movers or riggers when accepting, relocating, or disposing of equipment.

In anticipation of equipment delivery, prepare the final installation site in advance so that professional movers or riggers can transport the equipment to the computer room at the final installation site. If this is not possible at the time of delivery, make arrangements to have the professional movers or riggers return later to finish the transportation. To perform the required service, the authorized service provider can perform only minimal frame repositioning in the computer room, as needed.

For more information about the requirements for other PureApplication System Enterprise configurations, such as those listed below, go to the following website:

https://www.ibm.com/support/knowledgecenter/#!/SSCR9A_2.1.0/doc/installguide/planning/ps_sysreq_gen2enterprise.dita

Here are the machine types for the PureApplication System Enterprise:

- W2500-32 ‘Enterprise’ machine type is 8283-2A3
- W2500-64 ‘Enterprise’ machine type is 8283-2B3
- W2500-96 ‘Enterprise’ machine type is 8283-2C3
- W2500-128 ‘Enterprise’ machine type is 8283-2D3
- W2500-160 ‘Enterprise’ machine type is 8283-2E3
- W2500-192 ‘Enterprise’ machine type is 8283-2F3
- W2500-224 ‘Enterprise’ machine type is 8283-2G3
- W2500-320 ‘Enterprise’ machine type is 8283-2J3
- W2500-384 ‘Enterprise’ machine type is 8283-2L3

Note: The character *2* in the machine type means the number of chassis in the PureApplication System rack.

2.1.4 Hardware requirements: PureApplication System POWER configurations

Figure 2-5 and Figure 2-6 on page 25 show the hardware specifications for the PureApplication System racks with POWER processors.

PureApplication System W2700 'enterprise' (Power)		W2700-32	W2700-64	W2700-96	W2700-128	W2700-160	W2700-192	W2700-224	W2700-320	W2700-384
Compute	Processor	8 core, 4.1 GHz Power7+								
	Compute Node	Quad Processor, 32-core, 1TB memory								
	Memory/Compute Node	1.0 TB (16 2x32GB, 1066 MHz, DDR3, LP RDIMMS(1.35V))								
	Network cards/Compute Node	2 x Emulex 4-port 10GbE Network Adapter Mezz (Wildcat 2+2)								
	Fibre cards/Compute Node	2 x Qlogic 2-port 8Gb Fibre Channel Expansion Card (Ninja)								
	# Compute nodes	2	2	3	4	5	6	7	10	12
	# Cores	32	64	96	128	160	192	224	320	384
	Memory	1.0 TB	2.0 TB	3.0 TB	4.0 TB	5.0 TB	6.0 TB	7.0 TB	10.0 TB	12.0 TB
Storage	Solid State Drive (SSD)	16 x 400GB 2.5 in. SSD (E-MLC)								
	Hard Disk Drive (HDD)	80 x 600 GB 2.5 in. 10k HDD								
	Storage Controller	2 x IBM Storwize V7000 Disk System								
	Storage Expansion	2 x IBM Storwize V700 Storage Drawer								
	Total Storage SSD	6.4 TB unformatted / 4.8 TB usable								
	Total Storage HDD	48.0 TB unformatted / 43.2 TB usable								
Network	Top of Rack Switches (TOR)	BLADE Network Technologies (BNT) TOR switches for Customer Data Center & Rack to Rack communications								
	Transceivers	Choice of: 10GbE Fibre, 1GbE Fibre or Copper								
Chassis	Network / Chassis	2 x COMPASS – BNT 10Gb								
	Fibre / Chassis	2 x Pharos FC – Brocade 48 port								
	Chassis	3 x Accipiter Chassis								
Power	Power Distribution Unit (PDU)	4 x 60A 1ph – North America 4 x 32A 3ph - International								
Mgmt	Management Nodes	2 x PureSystems Manager (PSM) 2 x Virtualization System Manager (VSM)								
Rack	Rack	2.0 M 19" Enterprise Rack								
Specs	Weight	716 Kg (1575 lb)	716 Kg (1575 lb)	730 Kg (1606 lb)	744 Kg (1636 lb)	758 Kg (1667 lb)	772 Kg (698 lb)	786 Kg (1728 lb)	827 Kg (1820 lb)	855 Kg (1881 lb)
	Power	Max: 5.4 kW Typical: 6.8kW	Max: 5.4 kW Typical: 6.8kW	Max: 6.4 kW Typical:7.7kW	Max: 8.6 kW Typical: 7.3kW	Max: 9.6 kW Typical: 8.3kW	Max: 10.7 kW Typical: 9.3 kW	Max: 11.7 kW Typical: 10.2W	Max: 14.8 kW Typical: 14.1 kW	Max: 16.9 kW Typical:16.0kW
	Dimensions	Height: 2.015 m (79.3 in) / Depth: 1.130 m (44.5 in) / Width: 0.644 m (25.4 in)								

Figure 2-5 PureApplication System W2700 Enterprise rack specifications

PureApplication System W2700 'Mini' (Power)		W2700-32	W2700-64	W2700-96	W2700-128
Compute	Processor	8 core, 4.1GHz Power7+			
	Compute Node	Quad Processor, 32-core, 512GB memory			
	Memory/Compute Node	512GB (16 2x16GB, 1066 MHz, DDR3, LP RDIMMS(1.35V))			
	Network cards/Compute Node	2 x Emulex 4-port 10GbE Network Adapter Mezz (Wildcat 2+2)			
	Fibre cards/Compute Node	2 x Qlogic 2-port 8Gb Fibre Channel Expansion Card (Ninja)			
	# Compute nodes	2	2	3	4
	# Cores	32	64	96	128
	Memory	0.5 TB	1 TB	1.5 TB	2.0 TB
Storage	Solid State Drive (SSD)	6 x 400GB 2.5 in. SSD (E-MLC)			
	Hard Disk Drive (HDD)	40 x 600 GB 2.5 in. 10k HDD			
	Storage Controller	1 x IBM Storwize V7000 Disk System			
	Storage Expansion	1 x IBM Storwize V7000 Storage Drawer			
	Total Storage SSD	2.4 TB unformatted / 1.6 TB usable			
	Total Storage HDD	24.0 TB unformatted / 21.6 TB usable			
Network	Top of Rack Switches (TOR)	BLADE Network Technologies (BNT) TOR switches for Customer Data Center & Rack to Rack communications			
	Transceivers	Choice of: 10GbE Fibre, 1GbE Fibre or Copper			
Chassis	Network / Chassis	2 x COMPASS – BNT 10Gb			
	Fibre / Chassis	2 x Pharos FC – Brocade 48 port			
	Chassis	1 x Accipiter Chassis			
Power	Power Distribution Unit (PDU)	4 x 30A 1ph - North America 4 x 32A 1ph - International			
Mgmt	Management Nodes	2 x PureSystems Manager (PSM) 2 x Virtualization System Manager (VSM)			
Rack	Rack	2.0 M 19" Enterprise Rack			
Specs	Weight	590 Kg (1297 lb)	590 Kg (1297 lb)	604 Kg (1328 lb)	617 Kg (1358 lb)
	Power	Max: 4.9 kW Typical: 4.3 kW	Max: 5.8 kW Typical: 5.2 kW	Max: 6.8 kW Typical: 6.0 kW	Max: 7.7 kW Typical: 6.8 kW
	Dimensions	Height: 2.015 m (79.3 in) / Depth: 1.130 m (44.5 in) / Width: 0.644 m (25.4 in)			

Figure 2-6 PureApplication System W2700 Mini rack specifications

For detailed power and environmental requirements, go to the following websites:

- PureApplication System W2700 Mini racks:

  https://www.ibm.com/support/knowledgecenter/#!/SSCRSX_2.1.0/doc/installguide/planning/ps_sysreq_gen2mini.dita

- PureApplication System W2700 Enterprise racks:

  https://www.ibm.com/support/knowledgecenter/#!/SSCRSX_2.1.0/doc/installguide/planning/ps_sysreq_gen2enterprise.dita

Note: The difference between PureApplication System Intel and POWER configurations is in the types of processors and compute nodes. All other components, such as chassis, storage, and switches, are the same.

2.1.5 Site-readiness planning

Proper site-readiness planning is key for the fast and simple installation of a PureApplication System in your data center. Planning tasks must be carried out before the system is delivered to the data center.

Site-readiness planning includes the following items:

- Site selection
- Transport and delivery
- Static electricity and floor resistance
- Space requirements
- Floor construction and floor loading
- Computer room layout
- Vibration and shock
- Acoustics
- Electromagnetic compatibility
- Computer room location
- Material and data storage protection
- General power information
- Air conditioning determination
- Environmental design criteria

As a preferred practice, use site-readiness planning when preparing scenarios for a new installation, a model upgrade, hardware modifications, or for software modifications.

For the requirements for site-readiness planning, go to the following website:

https://www.ibm.com/support/knowledgecenter/#!/SSCR9A_2.1.0/doc/installguide/planning/siteplan.dita

2.1.6 Networking overview

The PureApplication System is a ready-to-use, private cloud solution that can be used without configuration. It does need to be connected to the data center network to work with the rest of your organization's IT environment.

The PureApplication System and PureApplication Software provide a wide-range of capabilities when integrated into an existing data center.

PureApplication uses *virtual local area networks* (VLANs) to isolate logically network traffic between applications and from system management traffic.

To connect the PureApplication System to your existing infrastructure, several networks are required:

- Management network
- Data network
- Cloud management network

PureApplication uses IPV6 for internal communication between its components. For external communications, the administrator can choose either IPV4 or IPV6.

Cloud management network: The cloud management network is required only when using a multi-system environment or externally managed cloud groups of PureApplication System.

Management network

The management network of a PureApplication System can isolate access to management functions and interfaces of the system. It provides access to the PureApplication management console (GUI), command-line interface (CLI), and to the REST interface. The management network is also used for backup and restore of the PureApplication System and PureApplication Software.

Cloud management network

In addition to the management network, the PureApplication System has a Cloud Management Network for intra-rack communication between its components. The VLANs you use to connect the PureApplication System to your data center network cannot use these internally reserved IDs, as listed in Table 2-10.

Table 2-10 Reserved VLANs for internal communication

Intel racks	Power racks
-	98
1358	1358
3201	3201
4091	4091
4093	4093
4094	4094
4095	4095

Note: PureApplication Software does not reserve any VLANs for internal communication.

Data network

The data network is designed for communications between virtual machines (VMs) running on the PureApplication System and other enterprise resources. For more information, see 3.2.4, “Physical network connection between the appliance and the data center” on page 65.

Most likely, you have more than one data network in your PureApplication environment.

Note: Data networks can be development, test, production, application tier, database tier, and so on.

2.1.7 Cloud groups

The PureApplication *cloud group* is a collection of one or more *compute nodes* and one or more *IP groups*.

There are two primary reasons to use cloud groups in PureApplication System:

- Resource isolation
- Logical aggregation of compute resources

You can use cloud groups to divide logically a PureApplication System into several isolated compute environments working with specific VLANs, ranges of IP address, and other parameters. Each cloud group has its own set of compute nodes and IP addresses, which allow for isolated applications to run on PureApplication. This configuration at the level of compute nodes achieves physical isolation between workloads and ensures that processes running in one cloud group are not affected by processes running in another cloud group.

For more information about cloud groups, see 11.2.1, “Cloud groups” on page 306.

Figure 2-7 shows the relationships between *cloud resources* in a PureApplication System.

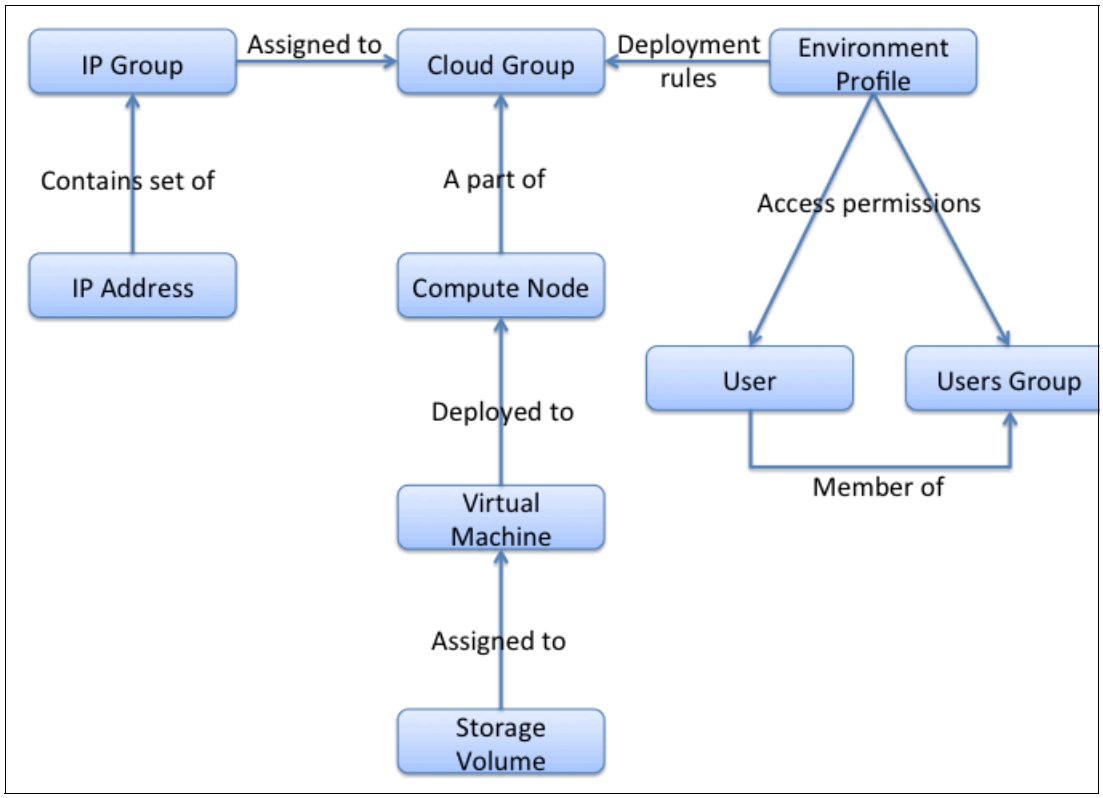


Figure 2-7 Relationships between cloud resources in the PureApplication System

2.1.8 Environment profiles

Environment profiles are policies for deploying patterns into cloud groups. These profiles create a logical isolation of resources by allocating the resources.

An environment profile defines policies for how patterns are deployed to one or more cloud groups and how their instances run in the cloud group. To deploy a pattern, you select a profile for performing the deployment, which, in turn, specifies the cloud groups to which the deployer can deploy patterns. Think of the environment profile as the target to which you are deploying a pattern. The fact that the profile deploys the pattern instance into a cloud group is a system-level detail of which a workload-level user, such as a deployer, does not need to be aware.

For more information, see 11.2.3, “Environment profiles” on page 309.

2.1.9 IP groups

An IP group is a set of IP addresses that are assigned to the cloud group, as indicated in Figure 2-7 on page 28. When deploying a new instance, PureSystems Manager obtains the required number of IP addresses from the IP group.

There are three primary functions of an IP group in a PureApplication System:

- Dynamic sharing of IP addresses across the cloud
- IP address pool isolation
- Logical network isolation

Note: PureSystems Manager can obtain IP addresses from an IP group automatically, or IP address can be set manually by the pattern deployer.

At the start of the deployment, the PureSystems Manager takes IP addresses from the IP group and marks them as In Use. When the instance is deleted, each address is added back to the pool and marked as Available.

Note: VMs can have several network interfaces and several IP addresses.

For more information, see 11.2.2, “IP groups” on page 308.

2.1.10 PureApplication System internal network overview

Like all components of PureApplication System, the internal network is designed for resiliency and redundancy to avoid any single point of failure.

There are three types of the switches in the system:

- Top-of-rack (TOR) switches (IBM System Networking RackSwitch G8264)
- Chassis switches (IBM Flex System® Fabric EN4093R)
- Storage area network (SAN) switches (IBM Flex System FC5022)

There are two TOR switches in the system. Each chassis contains two switches of each type. For more information, see IBM developerWorks at the following website:

http://www.ibm.com/developerworks/websphere/techjournal/1501_woolf/1501_woolf1.html

2.1.11 Ports that are required for a PureApplication System connection

The PureApplication System is designed to connect to the existing data center by using TOR switches.

Each of the TOR switches has 64 ports, which can be configured to use four types of transceivers, depending on your existing infrastructure:

- 1 Gb SFP fiber
- 1 Gb SFP copper
- 10 Gb SFP+ fiber
- 10 Gb SFP+ direct attach copper (DAC)

Note: Ports 1 - 40 and 57 - 63 are configured for internal rack communication. Ports 41 - 56 are available for your data network. Port 64 is designed for your management network.

Figure 2-8 shows the available ports on both TOR switches of PureApplication System.

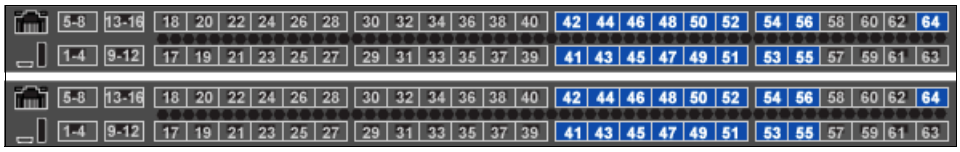


Figure 2-8 TOR switches of the PureApplication System

You can use different ports for your data network to achieve physical isolation between applications.

2.1.12 Domain Name System server

The PureApplication System requires a connection to a Domain Name System (DNS) server to function correctly. The system uses a DNS server to locate network resources during deployments. Therefore, connectivity issues between the PureApplication System and the DNS server can cause increased deployment times and failed deployments.

Note: The DNS server must be configured for both forward and reverse lookups. Connection to a DNS server is required for successful pattern deployments.

For more information, go to developerWorks at the following website:

`http://www.ibm.com/developerworks/websphere/techjournal/1407_woolf1/1407_woolf1.html`

2.1.13 Network time protocol server

Network time protocol (NTP) is a network protocol that is intended to synchronize clocks in computer networks.

An external NTP server is required for correct operation of the PureApplication System. Hypervisors, and the VMs synchronize their time to this server. If the NTP server is inaccessible, all hypervisors use their own hardware clock and sometimes become out of sync, which can cause issues with deployments and unwelcome vulnerabilities.

2.1.14 Networking topologies

As a preferred practice, configure the System Management IP network based on one of two topologies:

- Etherchannel (also known as link aggregation control protocol (LACP))
- Hot standby router protocol (HSRP)

In an Etherchannel or LACP topology, a single cable is connected to each TOR switch. Virtual link aggregation group (VLAG) provides the capability to group multiple links to achieve high availability, redundancy, and increased throughput, as shown in Figure 2-9 on page 31.

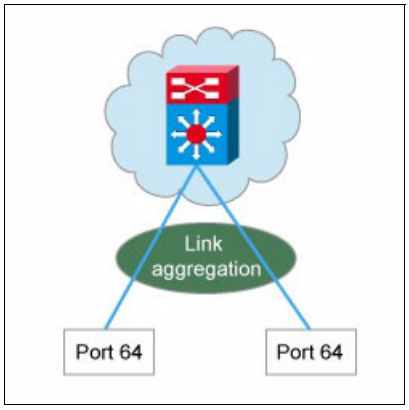


Figure 2-9 Etherchannel (or LACP) topology

Figure 2-10 shows the HSRP topology, in which two cables are connected to the TOR switch and are part of an aggregation. Here you see that there are two aggregations because four cables are connected to the rack.

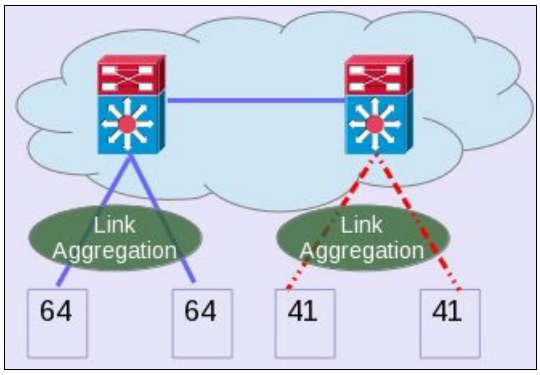


Figure 2-10 HSRP topology

For recommended topologies and communication options for applications, see the following IBM Knowledge Center website:

https://www.ibm.com/support/knowledgecenter/#!/SSCR9A_2.1.0/doc/installguide/planning/networkplan.dita

2.2 PureApplication Software requirements

PureApplication Software is installed into your existing hardware infrastructure. PureApplication Software does not include OS images or middleware.

The following section describes the PureApplication Software requirements and the steps to perform during the installation planning stage.

2.2.1 Configuration of the VMware vCenter server and ESXi

The configuration of the existing VMware infrastructure is required before you perform your IBM PureApplication Software installation. It includes the configuration of the VMware vCenter server and ESXi hosts and can be split into three main steps:

- Configuring a virtual data center
- Configuring ESXi hosts
- Configuring a network

The following sections provide high-level configuration steps. For detailed configuration steps, see the VMware documentation and the IBM Knowledge Center found at the following websites:

- http://pubs.vmware.com/vsphere-51/index.jsp?topic=%2Fcom.vmware.vsphere.doc%2FGUID-1B959D6B-41CA-4E23-A7DB-E9165D5A0E80.html
- https://www.ibm.com/support/knowledgecenter/api/content/nl/en-us/SSL5ES_2.1.0/doc/getstart/cfgvmware.html

2.2.2 Installation requirements

The installation requirements for PureApplication Software Versions 2.1.0.1 and 2.1.0.0 are described in Table 2-11 and Table 2-12.

Table 2-11 Installation requirements for PureApplication Software Version 2.1.0.1

PureSystems Manager Red Hat Enterprise Linux version	vCenter version	ESXi version
x86 64-bit Red Hat Enterprise Linux Version 6.6 with the latest updates	vCenter server V5.1U2a or vCenter server V5.5 U2	ESXi V5.1 U2 or ESXi V5.5 U2

Table 2-12 Installation requirements for PureApplication Software Version 2.1.0.0

PureSystems Manager Red Hat Enterprise Linux version	vCenter version	ESXi version
x86 64-bit Red Hat Enterprise Linux Version 6.6 with the latest updates	vCenter server V5.1U2a	ESXi V5.1 U2

2.2.3 PureSystems Manager requirements

The PureApplication Software Management node is known as *PureSystems Manager* and can be installed both on a VM or in a non-virtual environment, as shown in Figure 2-11 on page 33 and Figure 2-12 on page 33.

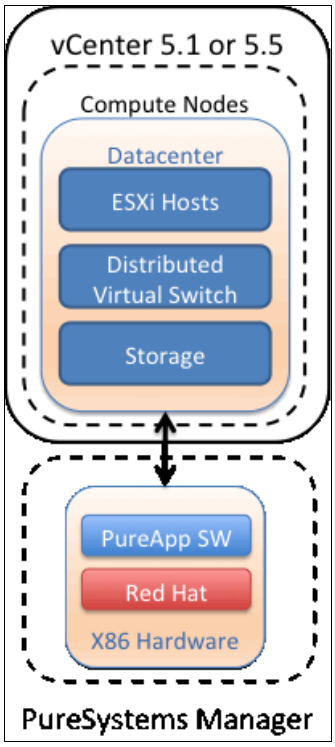


Figure 2-11 PureSystems Manager in a non-virtual environment

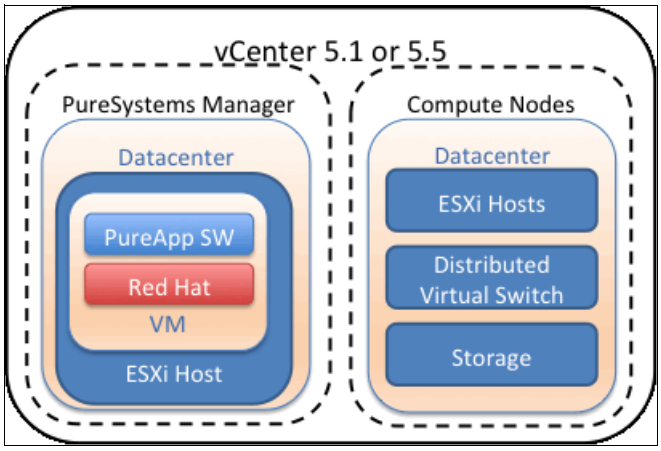


Figure 2-12 PureSystems Manager in a virtualized environment

A dedicated server with Red Hat Enterprise Linux must be prepared to run PureApplication Software. No other enterprise application can be installed other than PureSystems Manager.

Note: You must have the English locale (LANG=en_US.UTF-8) set in Red Hat Enterprise Linux. By default, this setting is configured in the `/etc/sysconfig/i18n` file.

PureApplication Software must be installed and run as the root user on the Red Hat Enterprise Linux operating system.

Here are the requirements for installing PureApplication Software:

- Security-Enhanced Linux (SELinux) must be disabled or in permissive mode (avoid extra warning messages in the logs by setting it to permissive mode).
- CPU: Minimum of eight cores.
- Memory: Minimum of 80 GB (of which up to 40 GB can be swap space).
- Partitions: Three disks with the following characteristics:
  - Root: Minimum of 10 GB
  - Data: Minimum of 1.5 TB
  - Run time: Minimum of 500 GB
- Red Hat YUM with configured YUM repositories Red Hat Enterprise Linux Version 6.6 and extra packages for Enterprise Linux.
- The following Red Hat Package Manager (RPM) packages automatically installed from YUM repositories:
  - compat-libstdc++-33.i686
  - compat-libstdc++-33.x86_64
  - bind-utils.x86_64
  - dhcp.x86_64
  - dnsmasq.x86_64
  - genisoimage.x86_64
  - httpd.x86_64
  - libcgroup.x86_64
  - lsof.x86_64
  - mod_ssl.x86_64
  - mod_security.x86_64
  - mod_security_crs.noarch
  - mod_security_crs-extras.noarch
  - ntp.x86_64
  - openssh-clients.x86_64
  - pam.i686
  - redhat-lsb-core.x86_64
  - unzip.x86_64

2.2.4 Firewall

PureApplication Software can use both IPv4 and IPv6 traffic for internal and external communication. To allow proper communication for your PureApplication Software instance, a loopback interface must be configured to accept all connections for IPv4 and IPv6 traffic. Ports for outbound traffic must be open on the firewall as well.

Table 2-13 and Table 2-14 on page 35 show information about the required ports for inbound traffic and internal communication between components.

Table 2-13 Inbound ports that are allowed through the firewall

Port	Protocol	Purpose
80	TCP	HTTPd
123	UDP	NTPD
443	TCP	HTTPd
8585	TCP	VM to PureApplication Software communication

Port	Protocol	Purpose
9443	TCP	VM to PureApplication Software communication
9444	TCP	VM to PureApplication Software communication

Table 2-14 Ports for internal communication between components

Port	Protocol
657	TCP
2000	TCP
5000, 5001	TCP
5003 - 5011	TCP
7005 - 7007	TCP
8080	TCP
8083	TCP
9080	TCP
9081	TCP
50002	TCP
50003	TCP
1747	UDP
1804	UDP
2123	UDP
2176	UDP
2495	UDP
2517	UDP
2534	UDP
2551	UDP
12347	UDP
12348	UDP

2.2.5 Compute node requirements

PureApplication Software supports configurations with no more than 32 compute nodes. You can have any number of cores in your compute nodes; however, all the compute nodes must have the same number off cores.

The amount of memory per compute not is not limited. PureApplication Software reserves an amount of memory that is equal to the largest compute node for high availability of the cloud group.

Note: Local storage on a single compute node is not supported. However, you can use shared storage that is mapped to the compute node if it is block storage. For more information, see “Block storage” on page 106.

2.2.6 Network requirements

There are three key networks requirements for ensuring that your PureApplication Software instance works correctly:

- The PureSystems Manager has an assigned IPv4 address on the same network as the vCenter server.
- The PureSystems Manager has IPv6 enabled for local loopback.
- The DNS and NTP servers are available.

For more information about PureApplication Software requirements, see the following website:

https://www.ibm.com/support/knowledgecenter/api/content/SSL5ES_2.1.0/doc/getstart/sysreqs.dita?locale=en

2.3 How PureApplication VMs connect to the network

During the data center planning stage, consider the applications and workloads that are deployed in your PureApplication environment. The required network bandwidth depends on the number and types of running applications. Therefore, it helps to plan the number of required physical connections to the data network.

The PureApplication System runs all workloads as VMs in cloud groups. Each VM in the PureApplication System connects to networks that are provided by the cloud group:

- Cloud group management (eth0)
- Application data (eth1)

The PureApplication Software uses an eth0 interface both for cloud group management and application data.

Note: Each VM can have more than one application data interface.

Figure 2-13 on page 37 shows how a VM connects to networks in PureApplication System.

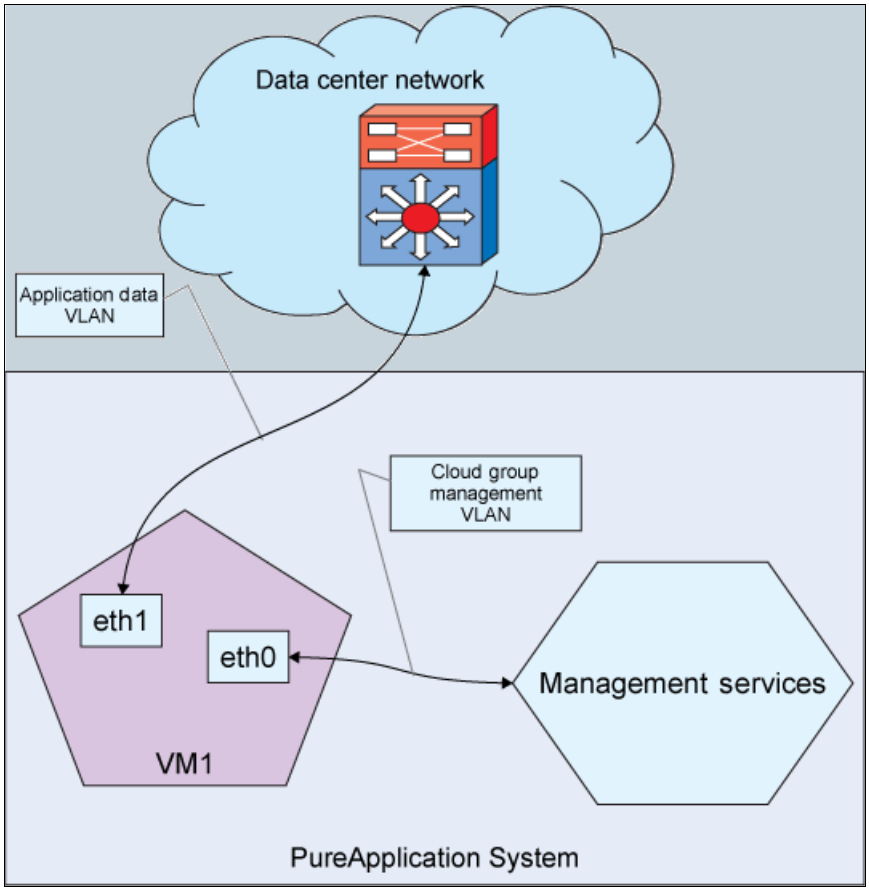


Figure 2-13 Network connections to a VM in PureApplication System

Figure 2-13 shows that eth0 is used for the connection between the VM and the management network of PureApplication System, which allows the system to manage running VMs by using a single UI.

Figure 2-13 also shows that eth1 connects the VM to the application data network. It allows communication between the VM and other enterprise resources outside of the PureApplication System.

IP groups and cloud groups can be used to isolate network traffic between applications in the PureApplication System. Each IP group can use a different VLAN and can be assigned to a different cloud group, which provides the option to isolate your applications according to enterprise standards and policies. Network isolation can even be applied to different parts of a single application when you want the network traffic of each part to be separated from the network traffic of the other parts.

In a typical three-tier architecture (web layer, application, and database), tiers are logically isolated from each other by using different VLANs. As shown in Figure 2-14, this topology can be implemented with IP groups and cloud groups in PureApplication System and in PureApplication Software.

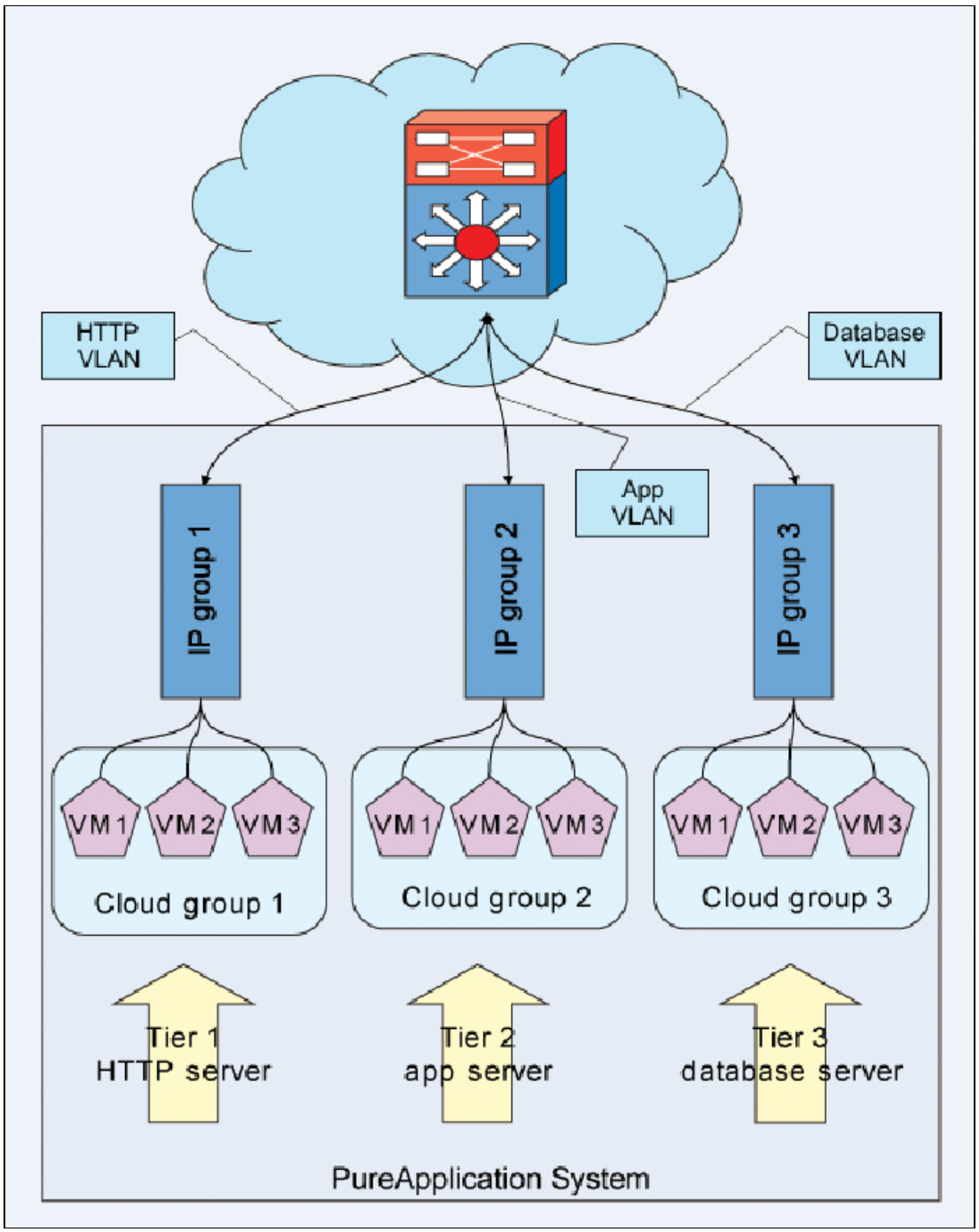


Figure 2-14 Three-tier application architecture with PureApplication System and PureApplication Software

Note: The required VLANs can be assigned to one physical link or to different links, depending on your required bandwidth and enterprise security standards.

For more information about network design, see the developerWorks article, found at:

http://www.ibm.com/developerworks/websphere/techjournal/1501_woolf/1501_woolf2.html

3

IBM PureApplication installation

This chapter describes the installation and configuration of PureApplication System and PureApplication Software. The intent of this chapter is to guide the reader through the process of planning, installing, and configuring both of these PureApplication products. The approach is different for each product.

The difference in PureApplication System and PureApplication Software can be summarized as a choice between customization and standardization. This choice is a common trade-off in cloud architectures and depends on workload. PureApplication System provides a standardized installation and can be available within hours. Because the installation of PureApplication Software is customized, it requires a longer period. You can use PureApplication Software to customize your infrastructure and installation to your specifications.

This chapter covers the following topics:

- PureApplication Software in the data center
- PureApplication System in the data center

© Copyright IBM Corp. 2015. All rights reserved.

3.1 PureApplication Software in the data center

PureApplication Software is installed into your existing hardware infrastructure (see Chapter 2, “Data center planning for an IBM PureApplication installation” on page 13). You can use PureApplication Software to build, run, and manage workloads on your VMware on your hardware. You can use this flexibility to use the existing skills and your investment in VMware virtualization and infrastructure management while taking advantage of the simplification and repeatability of PureApplication workload deployment and management by using patterns.

The following topics are covered:

- Obtaining PureApplication Software
- Setting up and installing PureApplication Software
- Configuring the vCenter environment
- Configuring the PureApplication Software connection to vCenter
- Configuring the PureApplication Software system settings
- Configuring and allocating cloud resources

3.1.1 Obtaining PureApplication Software

The two required packages for PureApplication Software can be downloaded from IBM Passport Advantage®. The current level is PureApplication Software V2.1.0.0. Complete the following steps:

1. Log in to Passport Advantage at the following website:

   `http://www.ibm.com/software/passportadvantage`

2. Go to **Software download & media access**.
3. Search for the eAssembly (source code) by part number *CRT20ML*.
4. Download the required images to a temporary directory:
   - PureApplication Software Installer V2.1.0.0 Multi-platform Multilingual:
     - File name: `pure-app-software-2.1.0.0.tgz`
     - Size: Approximately 6 GB
     - Contents: PureApplication Software installation
   - PureApplication Software Integrated Content V2.1.0.0 Multi-platform Multilingual:
     - File name: `pure-appsw-content-2.1.0.0.tar`
     - Size: Approximately 44 GB
     - Contents: Default data content for PureApplication Software
5. Use a file extraction utility to extract each part into a single temporary directory. The utility must be able to work with large file sizes because PureApplication Software has a compressed file size of 44 GB.

3.1.2 Setting up and installing PureApplication Software

PureApplication Software is installed on Red Hat Enterprise Linux with the latest updates. Red Enterprise Linux can be installed natively on a Linux server or on a virtual machine (VM). The requirements are found in the PureApplication Software IBM Knowledge Center at the following website:

http://www.ibm.com/support/knowledgecenter/SSL5ES_2.1.0/doc/getstart/sysreqs.dita?lang=en

Note: Red Hat Enterprise Linux Version 6.6 was used for this publication.

The Red Hat Enterprise Linux system must be dedicated to running PureApplication Software and no other enterprise applications.

The following example covers the steps that are required for building and using a Red Hat VM.

Building a Red Hat Enterprise Linux (RHEL) 6.6 system VM

The following instructions use vCenter to build an RHEL 6.6 system VM. Complete the following steps:

1. Download the Red Hat Enterprise Linux (RHEL) Version 6.6 server by using the Red Hat customer portal, found at the following website:

   https://access.redhat.com/

2. Create the VM in vCenter by using the following parameters:
   - 40 GB RAM.
   - 64 GB primary HDD.
   - Use the default settings in the RHEL installer. (Exception: HDD partitioning. RHEL does not make the swap large enough. Adjust 40 GB swap for 80 GB RAM total.)

Creating disks and mounts

To create disks and mounts, complete the following steps:

1. After the VM is created and started, attach two HDD devices in independent mode. This is done so that the HDD devices do not affect snapshots of the operating system.
   a. HDD device 1:
      - Size: 500 GB.
      - Partition, format, and mount as `/pureapp-sw-runtime`.

   b. HDD device 2:
      - Size: 1500 GB.
      - Partition, format, and mount as `/pureapp-sw-data`.

2. Create bind mounts so that the directory structure is prepped for the PureApplication Software installation by running the following commands. The entire list is shown in Figure 3-1.
   - **`mount --bind /opt/ibm/data/home/iwd/var/log/purescale /pureapp-sw-runtime`**
   - **`mount --bind /data/system/data/workload/drouter /pureapp-sw-data`**

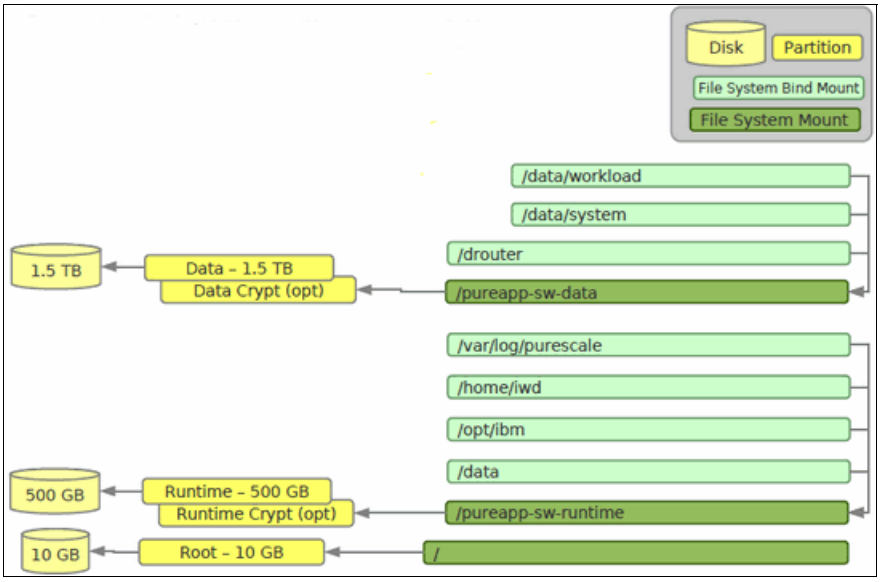


Figure 3-1 PureApplication Software example disk layout

Disabling selinux

Disable security enabled Linux (`selinux`) to allow RHEL configuration and PureApplication Software installation by completing the following steps:

1. Check to see whether `selinux` is enabled by running **`sestatus`**. If the status is *enabled*, proceed to the next step. If the status is *disabled,* go to “Installing the prerequisite Red Hat Package Manager” on page 44.
2. Edit `/etc/selinux/config`.
3. Set `SELINUX=disabled`.
4. Restart.
5. Check to see that `getenforce=Disabled`.

Installing the prerequisite Red Hat Package Manager

You need YUM access to both the RHEL and Extra Packages for Enterprise Linux (EPEL) repositories to install PureApplication Software. The installer checks for required Red Hat Package Manager (RPM) packages, and if they are not found, installs them. If the necessary RPMs are not found, the installer terminates with an error.

If your Linux server or VM is not connected to the RHEL or EPEL repositories, you can use the following method to do so. This method is valid if you deployed an internal satellite server that has RHEL updates on it. You need root access (see 7.2, “OS maintenance by using Red Hat Satellite Server” on page 157).

1. Register the Linux server or VM with the internal Satellite server.
2. After registering, update the system with YUM by running **yum update**. The following list shows the complete list of RPM packages that are installed from the YUM repositories:
   - compat-libstdc++-33.i686
   - compat-libstdc++-33.x86_64
   - compat-libstdc++-33.x86_64
   - bind-utils.x86_64
   - dhcp.x86_64
   - dnsmasq.x86_64
   - genisoimage.x86_64
   - httpd.x86_64
   - libcgroup.x86_64
   - lsof.x86_64
   - mod_ssl.x86_64
   - mod_security.x86_64
   - mod_security_crs.noarch
   - mod_security_crs-extras.noarch
   - ntp.x86_64
   - openssh-clients.x86_64
   - pam.i686
   - redhat-lsb-core.x86_64
   - unzip.x86_64

Enabling firewall ports

Completing the following steps enables certain firewall ports. The loopback interface must be configured to accept all IPv4 and IPv6 traffic. Ports for outbound ports must be open. Specific ports for inbound traffic are allowed through the ports that are shown in Table 3-1.

Table 3-1 Inbound ports that are allowed through the firewall

Port	Protocol	Purpose
80	TCP	HTTPd
123	UDP	NTPD
443	TCP	HTTPd
8585	TCP	VM to PureApplication Software communication
9443	TCP	VM to PureApplication Software communication
9444	TCP	VM to PureApplication Software communication

1. Stop iptables and ip6tables by running the following commands:
   - **service iptables stop**
   - **service ip6tables stop**
2. Check the status of iptables and ip6tables by running the following commands (should not be running):
   - **service iptables status**
   - **service ip6tables status**

3. Open the ports that are listed in Table 3-1 on page 45.
4. Start iptables and ip6tables by running the following commands:
   - **service iptables start**
   - **service ip6tables start**
   - **service iptables status**
   - **service ip6tables status**

Installing PureApplication Software

Installing PureApplication Software involves extracting the `install.tgz` file and running the installation script. Generally, the installation takes about 40 minutes, and the time to install the default data set is about two hours.

Complete the following steps:

1. Extract the PureApplication Software installation `.tgz` file:
   a. Verify that the default data set, `pure-appsw-content-2.1.0.0.tar`, is in the same directory as `pure-app-software-2.10.0.tgz`.
   b. Create a subdirectory in the temporary directory where the PureApplication Software files were downloaded, as described in 3.1.1, “Obtaining PureApplication Software” on page 42.
   c. Change to the subdirectory and extract `pure-app-software-2.10.0.tgz`.
2. Run the installation script to install PureApplication Software. Figure 3-2 on page 47 shows some of the sample output from the installation process. The red boxes in the figure indicate user input (such as installation name or user name and password). Green highlighted text indicates major steps of the installation script.
   a. Run **install.sh -t “software”**.
   b. Provide the following parameters during the installation:
      - Installation name, for example, `ISSWPureAppSW`
      - Admin user name and password

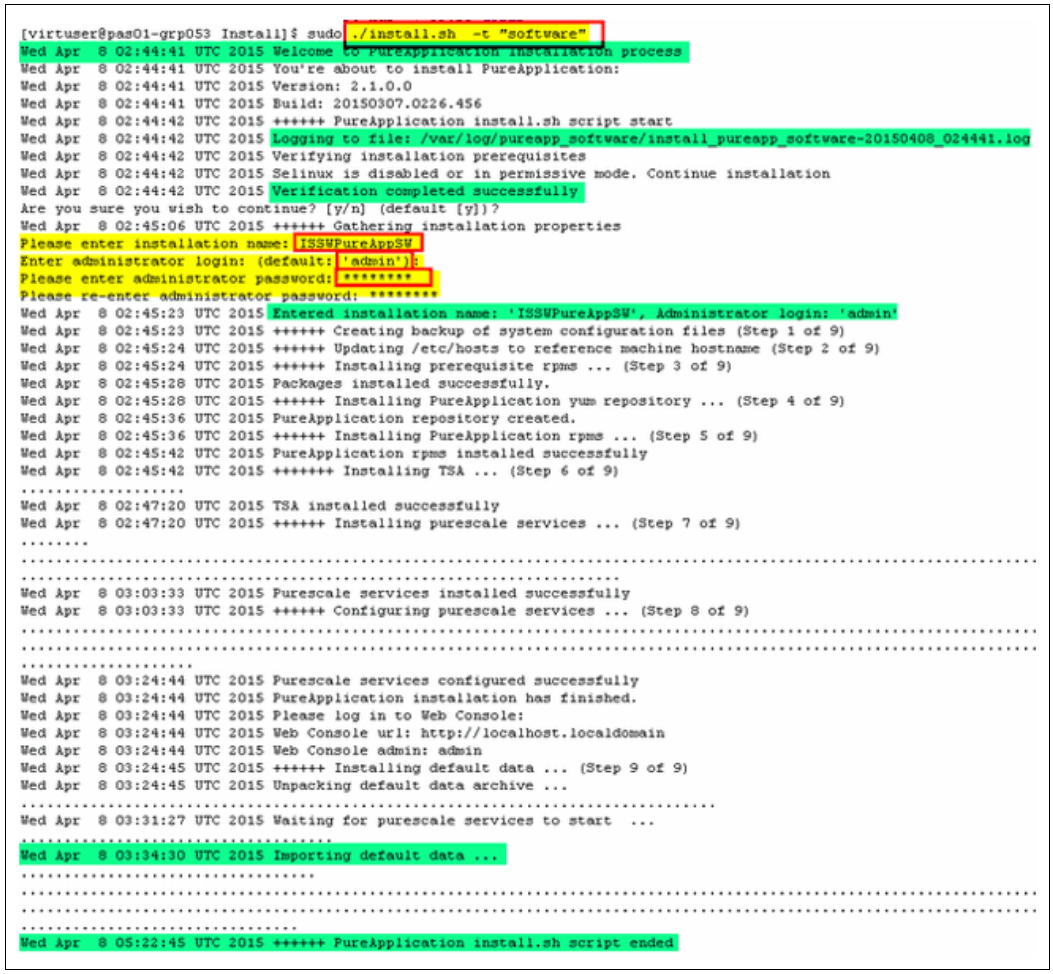


Figure 3-2 Installation program output (red boxes indicate user input; green text indicates major steps)

3. Two important log files are created. Use the **`tail`** function to view these files during the installation process.
   - Installation log file: `/var/log/pureapp_software/install_pureapp_software-{date-build}.log`
   - Content import log: `/var/log/purescale/purescale.log`

Verifying the installation of PureApplication Software

With the installation complete (see the last line of output in Figure 3-2 on page 47), verify the status of the installation by running the following steps:

1. Verify which services are online by opening a browser on the Linux server. If you see the PureApplication Software console, as shown in Figure 3-3, the services are online.

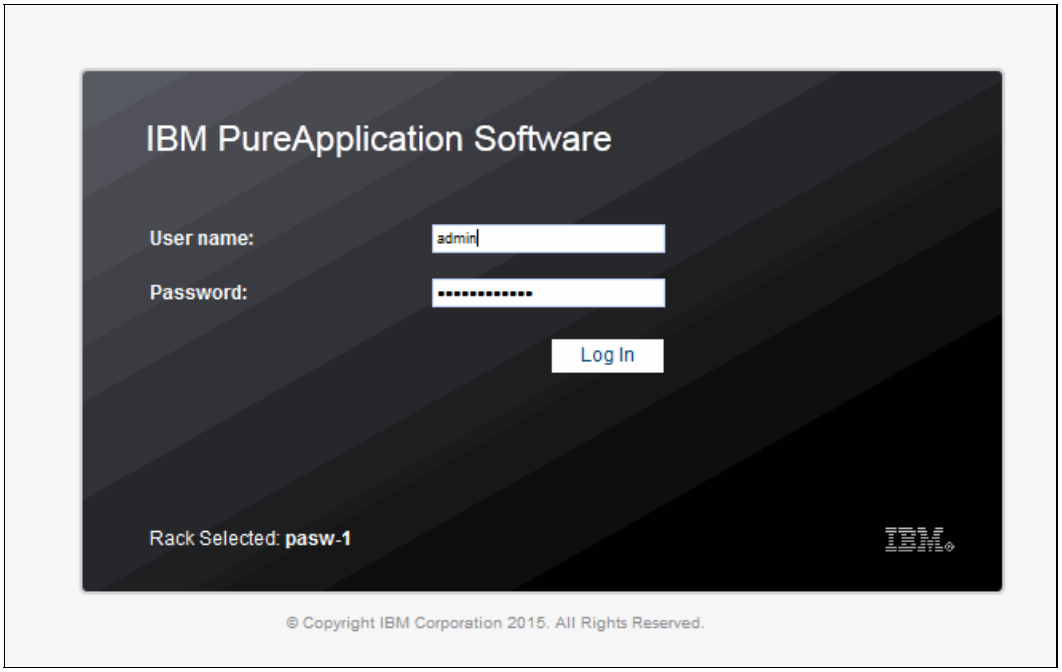


Figure 3-3 PureApplication Software console

2. Log in by using the credentials that are provided during the installation. The PureApplication Software welcome window opens, as shown in Figure 3-4.

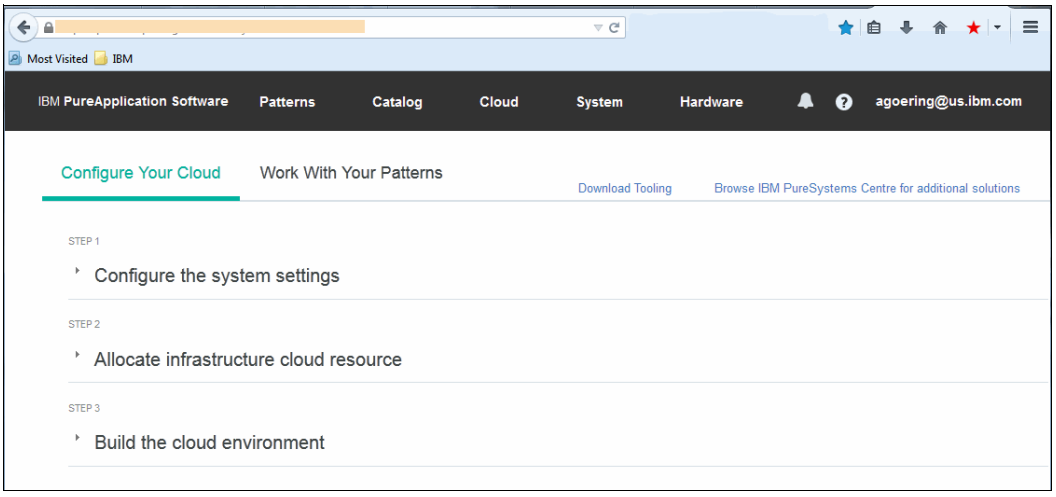


Figure 3-4 PureApplication Software welcome window

3.1.3 Configuring the vCenter environment

PureApplication Software requires the following items, as noted in the IBM Knowledge Center, found at:

http://www.ibm.com/support/knowledgecenter/SSL5ES_2.1.0/doc/getstart/sysreqs.dita?lang=en

- VMware vCenter 5.1 u2a
- VMware ESXi 5.1 U2a

Note: In this publication, VMware vCenter 5.1 u2a was used in a Windows environment.

PureApplication Software communicates with a vCenter server to perform actions in the VMware environment. There are three major steps to configure the vCenter environment so that it communicates with the PureApplication Software:

1. Configure the vCenter data center and user role (see “Configure vCenter data center and user role”).
2. Configure the VMware ESXi host (see “Configuring the VMware ESXi host” on page 52).
3. Configure vCenter networking (see “Configuring vCenter networking” on page 53).

Configure vCenter data center and user role

PureApplication Software can communicate with a single data center in the vCenter environment. To configure your vCenter to PureApplication Software, two assumptions are made:

- The vCenter user that is provided to PureApplication Software accesses only one data center.
- PureApplication Software is the only user that makes changes to the data center.

The first step is to create a data center in vCenter or use an existing data center. All ESXi hosts that are managed by PureApplication Software must be in the data center. If the data center contains any clusters, the clusters are ignored by PureApplication Software and all ESXi hosts that are contained in the clusters. The clusters are ignored because PureApplication Software creates clusters when cloud groups are created in PureApplication Software.

vCenter User role

To create the initial vCenter user role, complete the following steps. It is assumed that you are using the Windows operating system.

1. If you have one data center, assign the administrative user to PureApplication Software.
2. You might have multiple data centers. You must create a vCenter user that has access only to the data center that is used by the PureApplication Software. Create a user specifically for PureApplication Software by completing the following steps. This vCenter user was created for this scenario. The user has login privileges to vCenter.
   a. Create a Windows operating system user. In this scenario, the user is named pasoftware.
   b. Set the password to not expire.
3. Start the vSphere administrative client to connect to vCenter.

4. Create a user role with permissions that allow the user to access solely the PureApplication Software data center:
   a. Go to the Roles view, and click **View** → **Administration** → **Roles**.
   b. Click **Add Role**.
   c. Provide a name for the role. In this scenario, we use PureApp Administrator.
   d. Select **All Privileges**.
   e. Clear **All Privileges** and **Global**.
   f. Click **OK**.

   Figure 3-5 shows a summary of the user attributes.

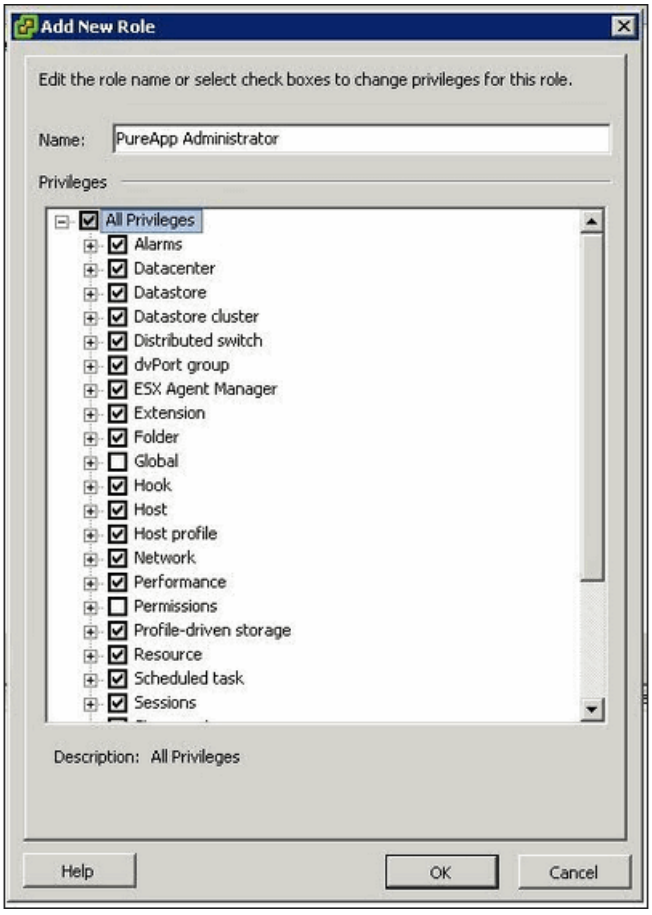


Figure 3-5 New user role configuration in vCenter

5. Assign the user to the data center:
   a. Go to a view where data centers are visible. Click **View** → **Inventory** → **Hosts and Clusters**.
   b. Right-click the data center that is identified for use by PureApplication Software, as shown in Figure 3-6 on page 51.

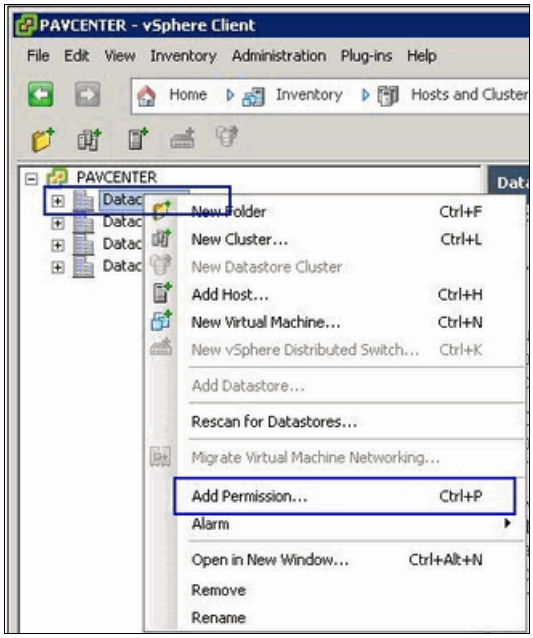


Figure 3-6 New user role permissions

c. Click **Add**.

d. From the Select Users and Groups dialog box, select the PureApplication Software administrative user. In this scenario, we used pasoftware.

e. Click **Add**, and then click **OK**, as shown in Figure 3-7.

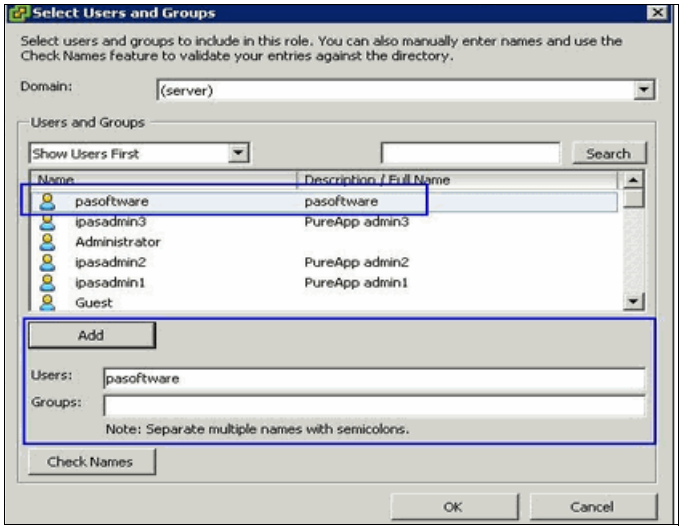


Figure 3-7 Add user

f. From the Assign Permissions dialog box, select the user. In this scenario, we selected pasoftware.

g. From the Assigned Role list, select **PureApp Administrator**.

h. Click **OK**, as shown in Figure 3-8. The user has full access to the PureApplication data center, but no other data centers.

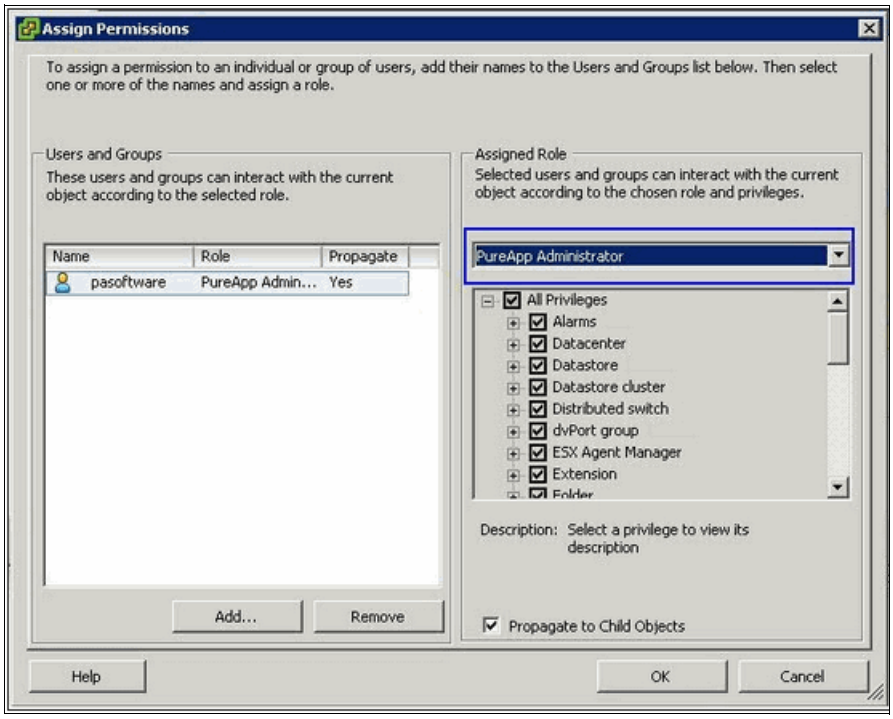


Figure 3-8 Assigned role

Configuring the VMware ESXi host

There are a number of VMware ESXi host configuration requirements. The purpose of this section is to provide guidance in this area.

Requirements

Here are the requirements for configuring the VMware ESXi host:

- Any existing VMware ESXi hosts that are known to vCenter can be moved to the PureApplication Software data center. VMware ESXi hosts must be added to the PureApplication data center for the PureApplication Software to use them.
- Any new VMware ESXi hosts must be added in the PureApplication data center. These hosts should not be part of any cluster in the PureApplication data center.

> **Note:** Do not edit the vSphere clusters that are created by the PureApplication Software. Changing cluster settings or contained objects can result in deployment or runtime failures.

- No VM should be on any ESXi hosts that are used by the PureApplication Software.
- After discovery, any ESXi hosts that are managed by PureApplication Software should not be modified manually.

- The PureApplication Software uses a cache data store for image storage. It is a preferred practice that the data store be 2 TB minimum. This size is only a recommendation and depends on the quantity and size of images in the environment. The data store must be mounted to *all* ESXi hosts in the PureApplication data center. In the example that is shown in Figure 3-9, a data store that is named cache_2TB_1 is mounted to four ESXi hosts in a PureApplication data center.

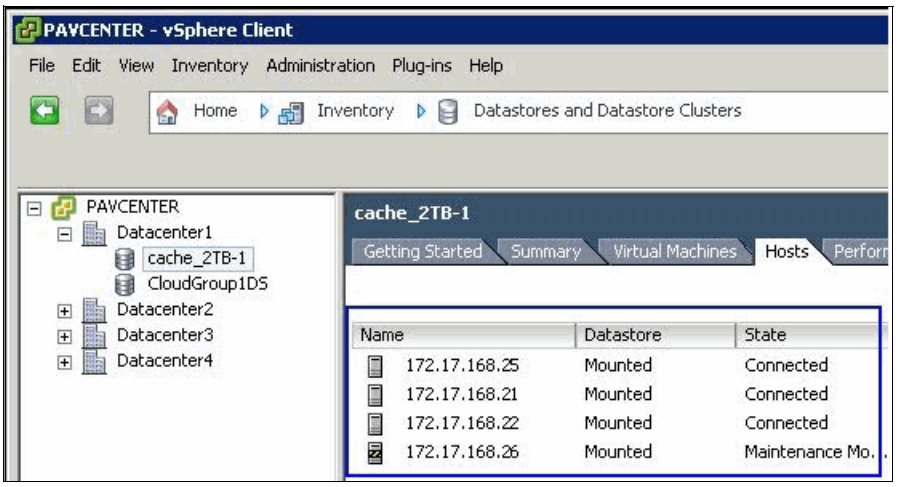


Figure 3-9 Data store that is used to store all images

- To perform deployments, cloud groups in the PureApplication Software need at least one data store. Data stores that are used for cloud group deployments must be accessible to all ESXi hosts that are part of the cloud group.

> **Note:** The PureApplication Software automatically mounts the data stores to ESXi hosts that are moved in and out of cloud groups. However, the LUNs that contain the data store or data stores must be mapped correctly to the ESXi host.

- All VMware ESXi hosts in the PureApplication data center must have **Lockdown Mode** disabled. This configuration is necessary because the PureApplication Software must be able to **ssh** into the ESXi host for various tasks.
- All VMware ESXi hosts in the PureApplication data center must connect to a vSphere distributed switch. Details for this configuration are described in “Configuring vCenter networking”.

Configuring vCenter networking

PureApplication Software is designed to work with a vSphere distributed switch (VDS). The VDS provides a centralized interface for configuring, monitoring, and administering VM access switching for the entire data center. All ESXi hosts in the PureApplication data center must be connected to the VDS. Connect only one VDS in the PureApplication data center.

Complete the following steps:

1. From the vSphere client, click **View** → **Inventory** → **Networking**.
2. Right-click the PureApplication data center and select **New vSphere Distributed Switch...** (Figure 3-10).

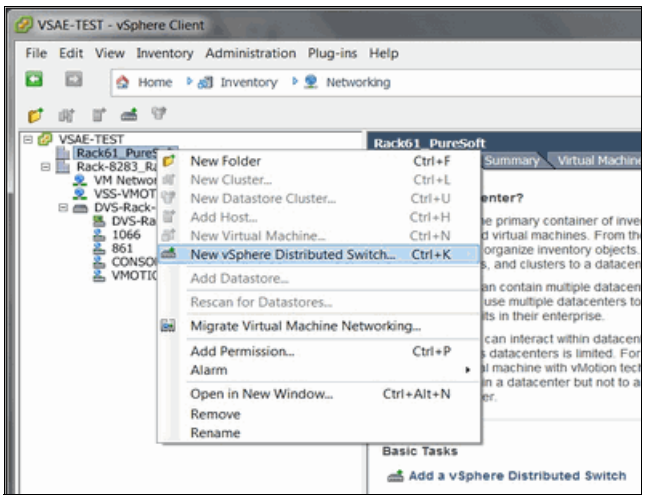


Figure 3-10 Create a vSphere distributed switch

3. Select the default option of **vSphere Distributed Switch Version: 5.0.0** and click **Next**.
4. Select **Add later** when asked when to add hosts to the new switch. Click **Next** (Figure 3-11).

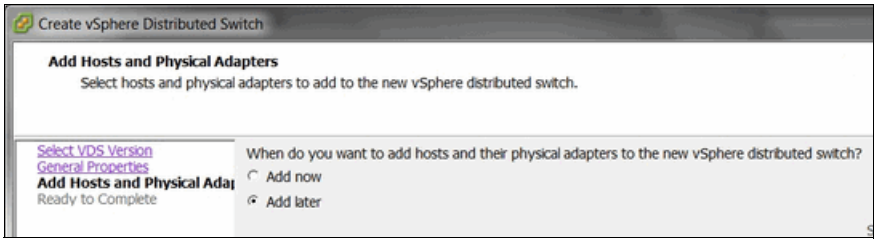


Figure 3-11 Option to add hosts to a switch

5. Enter a name for switch. Use the default value of four uplink ports. Click **Next** (Figure 3-12 on page 55).

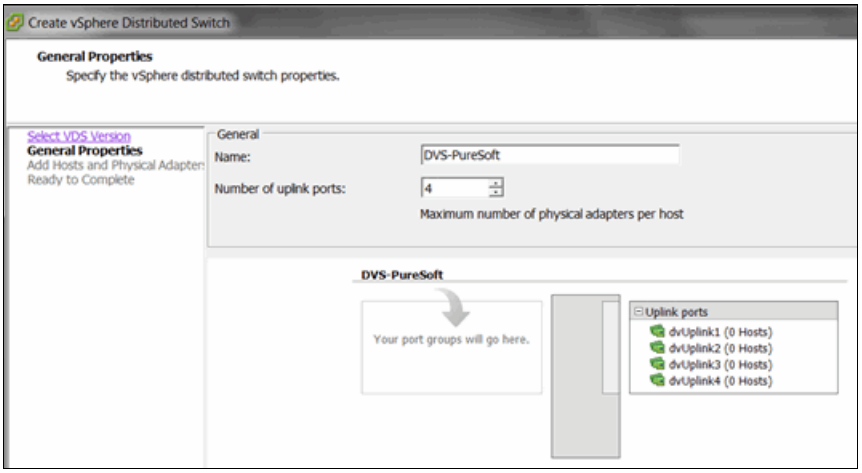


Figure 3-12 Option to add uplink ports

6. Clear **Automatically create a default port group** and click **Finish** (Figure 3-13).

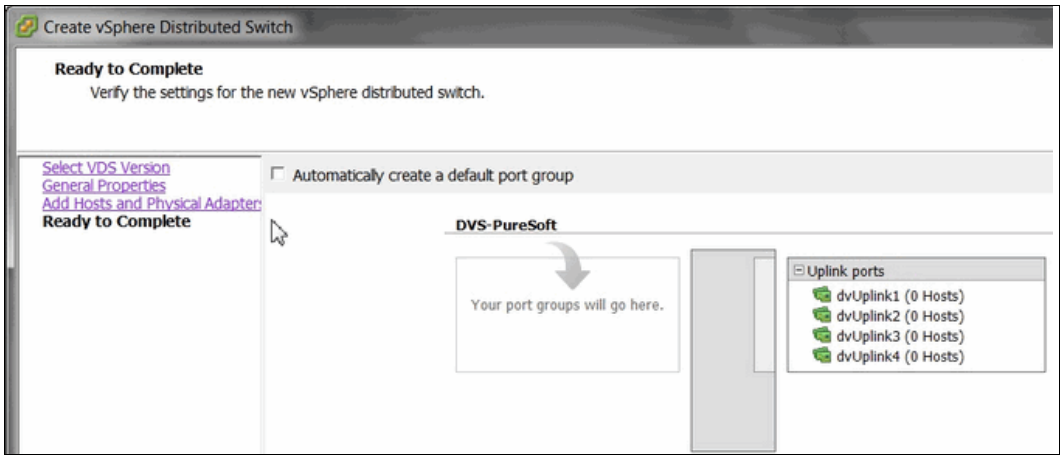


Figure 3-13 Option to create a port group

7. The ESXi server Management Network requires a distributed port group. Right-click the newly created switch and select **New Port Group...** (Figure 3-14).

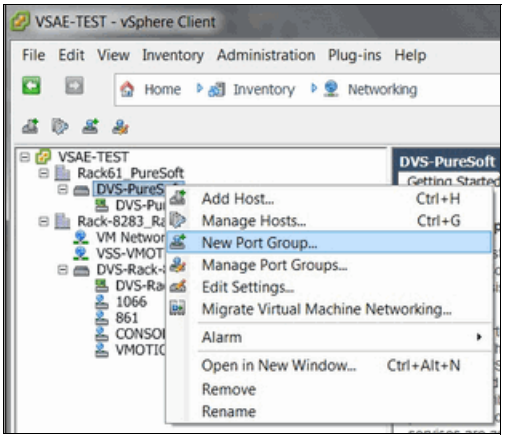


Figure 3-14 Option to create a port group for the Management Network

8. Configure the distributed port group, as shown in Figure 3-15:
   - Name: Enter the VLAN ID.
   - Number of ports: 8192.
   - VLAN type: VLAN.
   - VLAN ID: Obtain the ID from the network administrator.

> **Note:** The VLAN ID is probably the same VLAN that is used for communications between the ESXi hosts and vCenter.

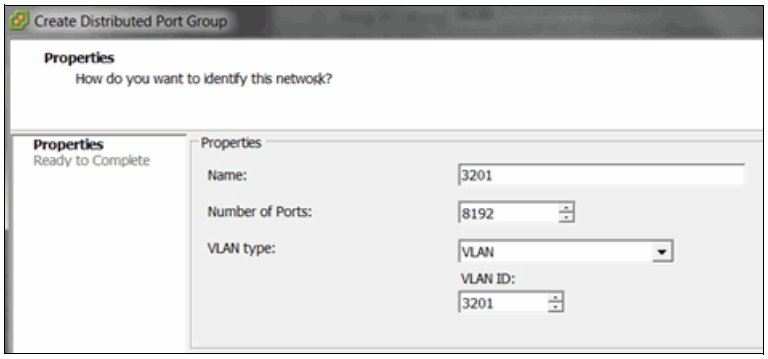


Figure 3-15 Distributed port group configuration

9. Review the distributed port group settings, as shown in Figure 3-16. Click **Finish**.

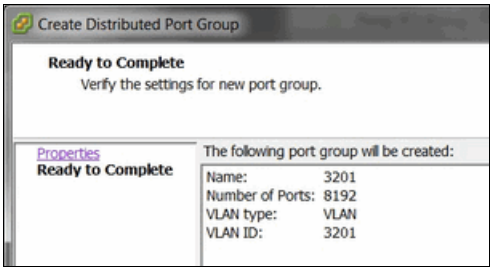


Figure 3-16 Final distributed port group settings

Adding an ESXi host to the distributed switch

> **Important:** Each ESXi host must be added to the distributed switch. Networking on the vSphere standard switch must be migrated. An example is shown in this section. Mistakes can lead to a loss of communication with the ESXi host.

Complete the following steps:

1. From the vSphere client, click **View** → **Inventory** → **Networking**.
2. Select the switch and select the **Configuration** tab.
3. Click **Add Host**.
4. Select the host and NICs to add to the switch (Figure 3-17 on page 57). Click **Next**.

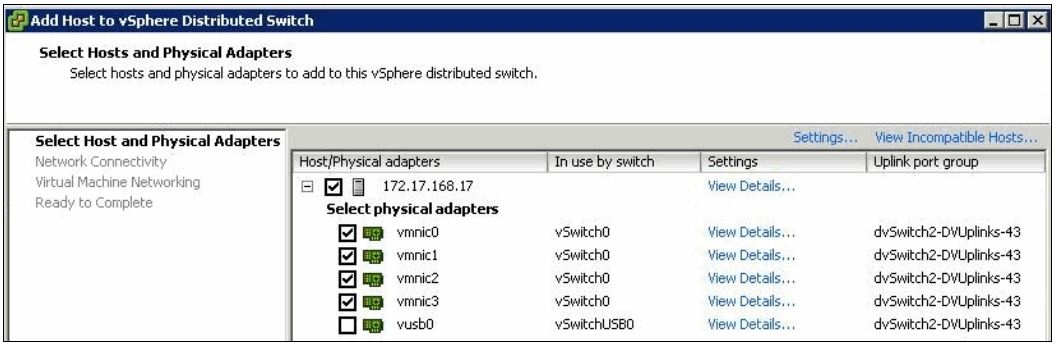


Figure 3-17 Add hosts to distributed switch

5. Verify visually that the vmk virtual adapter for the Management Network is mapped to the correct port group on the distributed switch (Figure 3-18). Click **Next**.

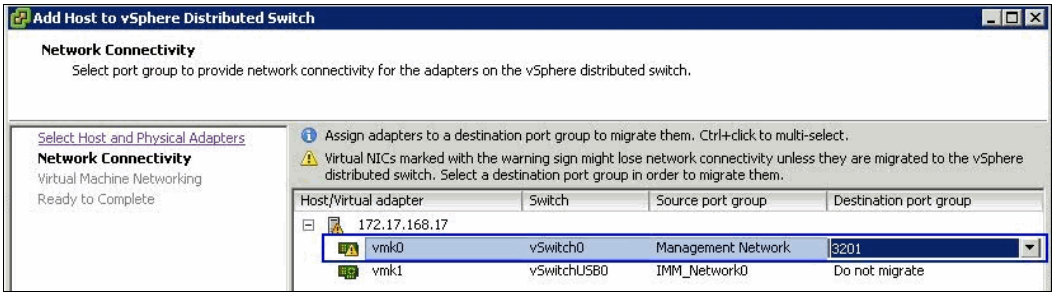


Figure 3-18 vmk virtual adapter selection

6. Because there are no VMs on the ESXi host, there are no VMs to migrate. Click **Next**.
7. Click **Finish**.

Enabling vMotion on the hosts in the cluster

To support load balancing through VM migration, vMotion must be enabled on the hosts in the cluster. A VMKernel adapter must have vMotion enabled for all the ESXi hosts in the data center. An existing VMKernel adapter can be used, or a new one can be created. The same VMKernel must be used for all ESXi hosts.

To enable vMotion on a VMKernel adapter, complete the following steps:

1. From the vSphere client, click **View** → **Inventory** → **Hosts and Clusters**.
2. Select the ESXi host.
3. Select the VDS and click **Manage Virtual Adapters**.
4. Select the VMKernel adapter on which to enable vMotion and click **Edit**.

5. Select **Use this virtual adapter for vMotion** and click **OK** (Figure 3-19). For more information, see the VMware documentation that is found at the following website:

   http://kb.vmware.com/selfservice/microsites/search.do?language=en_US&cmd=displayKC&externalId=1006989

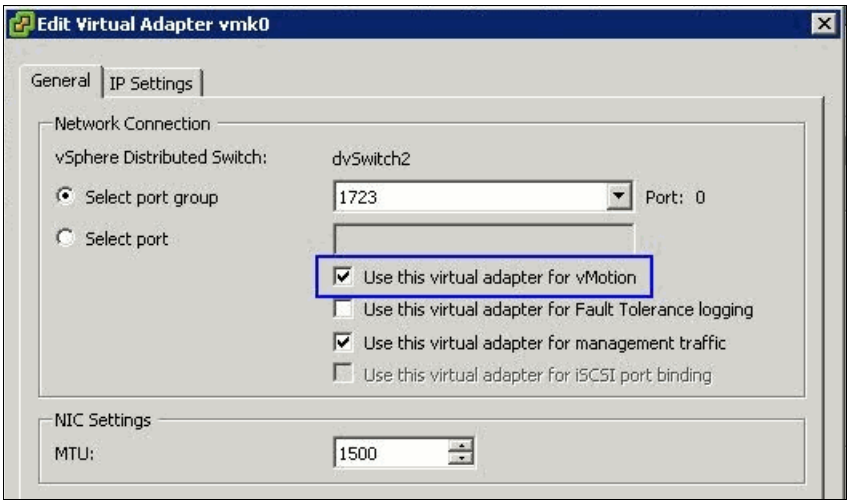


Figure 3-19 Edit virtual adapter

6. Repeat steps 1 on page 57 to 5 for all ESXi hosts.

Note: When you create IP groups in the PureApplication Software, you might need to create additional port groups on the distributed switch. One port group for each VLAN must be created.

3.1.4 Configuring the PureApplication Software connection to vCenter

You should have vCenter and any ESXi hosts configured as shown in 3.1.3, “Configuring the vCenter environment” on page 49 with the following items:

- vCenter data center with assigned ESXi hosts
- VDS
- Storage data stores that are visible to all the hosts, with one data store of around 2 TB named

3.1.5 Configuring the PureApplication Software system settings

There are three system settings that must be configured: Cloud Management IP, Domain Name Service, and NTP.

Configuring the cloud management IP

The cloud management IP is the network setting of the Red Hat server where PureApplication Software is installed. At a different level, this IP address is where the PureSystems Manager is running. If the Red Hat server has multiple NICs, you can specify the NIC to which the VMs and PSM connect.

Complete the following steps:

1. Click **System** → **System Settings** → **Cloud Management IP**.
2. Enter the following information for the Red Hat server hosting the PureApplication Software (Figure 3-20 shows an example):
   a. Management IP
   b. Host Name
   c. Domain Name
   d. Netmask: 255.55.255.0
   e. Default Gateway
   f. VLANS: Specify the VLANs that are used by your IP groups.

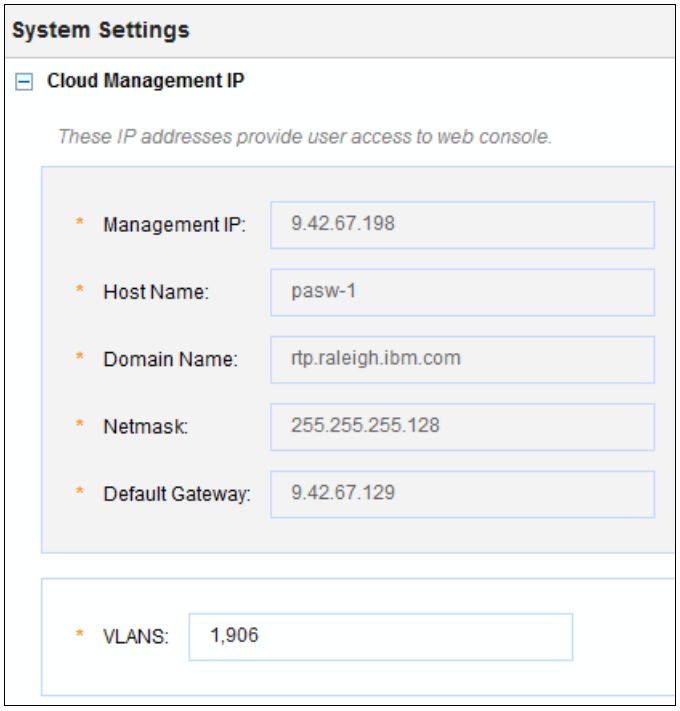


Figure 3-20 Example Cloud Management IP configuration

Configuring the DNS server

Complete the following steps:

1. Click **System** → **System Settings** → **Domain Name Service (DNS)**.
2. Enter the IP address or addresses of your DNS (Figure 3-21 shows an example).

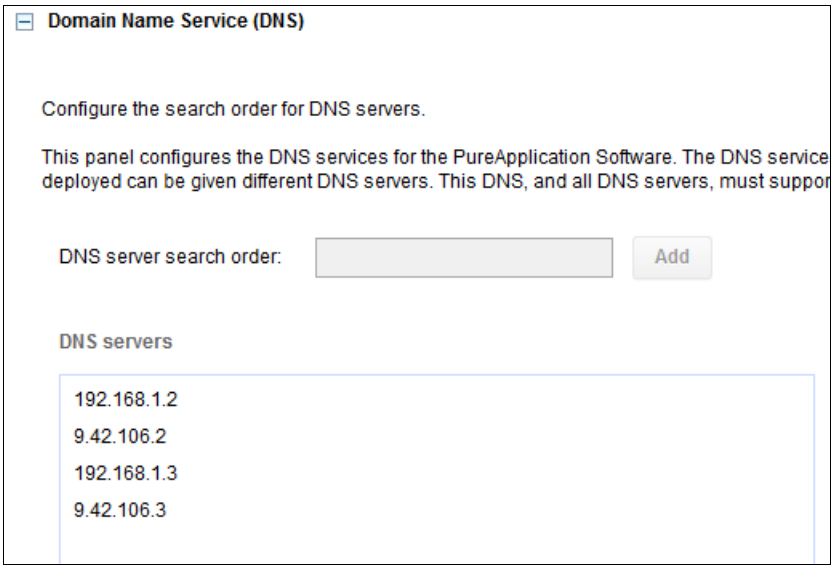


Figure 3-21 Example DNS configuration

Configuring the date and time

Complete the following steps:

1. Click **System** → **System Settings** → **Date and Time**.
2. Enter the address of the NTP server (Figure 3-22 shows an example).

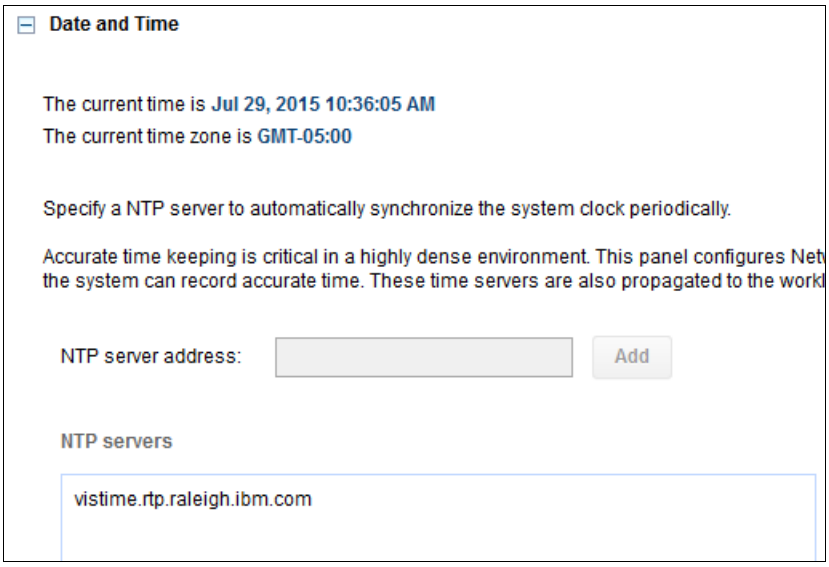


Figure 3-22 Example NTP server configuration

3.1.6 Configuring and allocating cloud resources

This section describes the tasks that an IBM representative performs to set up, power on, test, and make the physical network connection to support the PureApplication System. A test deployment is also described.

Configuring external vCenter servers

Set up the initial vCenter parameters that the PureApplication Software uses for its connection. After the PureApplication Software is connected, it manages the vCenter data center. Other than adding ESXi hosts or storage, you should not change any vCenter configuration data.

Complete the following steps:

1. Click **System** → **System Settings** → **vCenter Access**.
2. Enter the vCenter IP address (do not enter the host name), root user name, and password.
3. Click **Test Connection** to verify the PureApplication Software connection to vCenter. The message, *vCenter connection is successful* displays, along with configuration information, as shown in Figure 3-23.

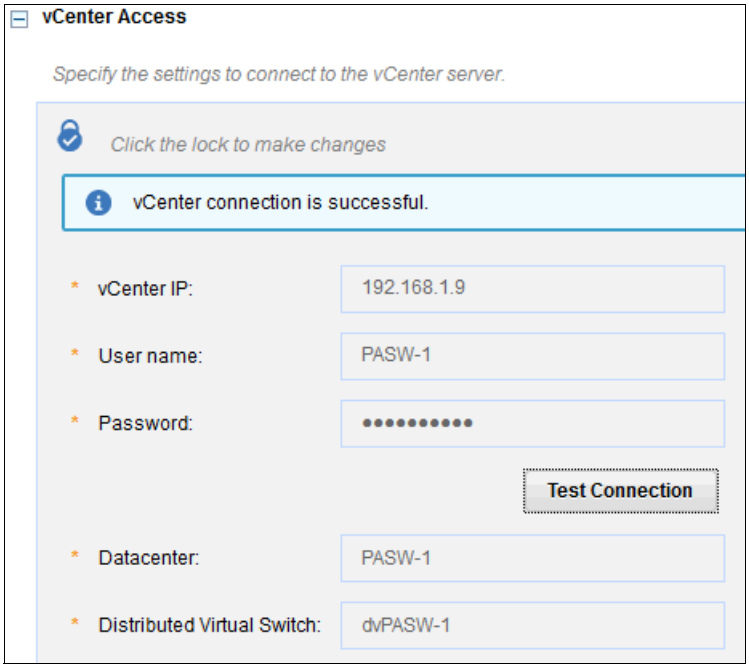


Figure 3-23 Example vCenter configuration

4. Enter the data center and VDS that is specified in “Configuring vCenter networking” on page 53.
5. Save your changes.

Discovering compute resources

Discover the ESXi hosts that were added to the vCenter data center (see “Adding an ESXi host to the distributed switch” on page 56) by completing the following steps:

1. Click **Hardware** → **Compute Resources**.
2. Click the **Discover** icon, as shown in Figure 3-24.

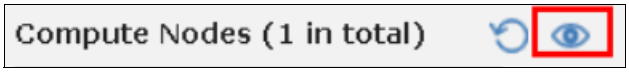


Figure 3-24 Discover ESXi hosts

3. The ESXi hosts that are added to the vCenter data center are shown as a list. Figure 3-25 shows an example.

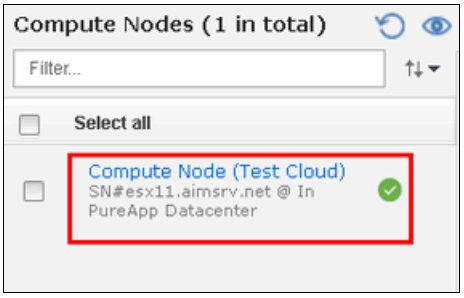


Figure 3-25 List of ESXi hosts

Discovering storage resources

The following procedure discovers all of the ESXi hosts and storage that are added to the vCenter storage (data store). You see the cache_2TB-1 data store that is used for the cache. The remaining data stores can be allocated to the appropriate cloud groups.

1. Click **Hardware** → **Storage Resources**.
2. Click the **Discover** icon, as shown in Figure 3-26.

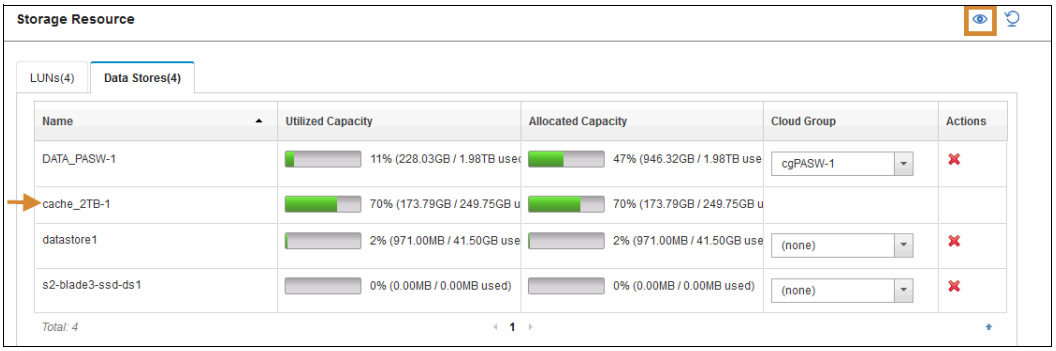


Figure 3-26 List of storage resources

3. Allocate the remaining data stores to the appropriate cloud groups.

3.2 PureApplication System in the data center

PureApplication System is a ready-to-use application platform that is integrated into one rack. It contains all of the required hardware and software components to build a private cloud environment on your data center. Therefore, it can be installed and integrated into an existing environment much faster than traditional solutions. It requires hours, not days or weeks, to install PureApplication System.

The PureApplication System appliance is installed by IBM. This section provides an overview of the installation steps that are performed by the IBM representative in your data center. These steps are provided for your information and understanding.

3.2.1 Physical inspection of the rack

The IBM representative ensures that the data center in which the system is being installed meets or exceeds the requirements for physical space and weight, and is appropriate for providing cooling and environmental considerations. The IBM representative performs a physical inspection of the rack, including the following items:

- Removing shipping brackets from the storage nodes.
- Reseating storage control and expansion canisters.
- Reseating storage power supplies.
- Ensuring that storage racks are cabled correctly and all cables are plugged into the correct locations.
- Confirming that the cables that connect the system power distribution units (PDUs) to all four storage enclosures are connected.
- Ensuring that all disks are correctly seated in the storage nodes.

The IBM representative ensures that all other components of the system are connected and seated correctly, including the following items:

- The management nodes and compute nodes in the chassis.
- Confirming that the PDU cables correctly plug into the IBM RackSwitch G8264 switches and the chassis.
- Ensuring that all I/O modules, Chassis Management Modules (CMMs), fan modules, and power modules are well seated in the back of the chassis.

Note: All cables in PureApplication System are marked with special labels, indicating where the cable should be connected, as shown in Figure 3-27.

As shown in Figure 3-27, each label displays two lines of information:

- The top line displays the *TO* information, which identifies the location of the port to which that end of the cable (the end that bears the label) connects.
- The bottom line displays the *FROM* information, which identifies the location of the port to which the opposite end of the cable (the end that does not bear the label) connects.

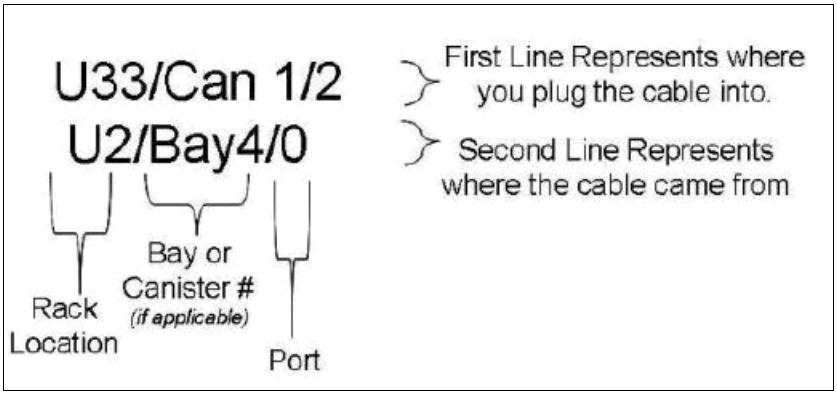


Figure 3-27 Sample cable label in the PureApplication System

When the physical inspection of the rack is complete, the IBM representative powers on the rack.

3.2.2 Powering on the rack

Note: The IBM representative verifies that the circuit breakers for the PureApplication System are powered off before completing these steps.

The IBM representative completes the following steps to power on the PureApplication System:

1. Connect all power cables to electrical outlets.
2. Double-checks that all Storwize V7000 controllers and expansion enclosures are powered off.
3. Turns on the power breaker switch for the PureApplication System.

   System initialization starts, which takes approximately 10 minutes and does not require any intervention.
4. When initialization is complete, verify that all the LEDs of the compute nodes and PureSystems Managers are green and flashing slowly on the front of the chassis.

   Note: System flashes are defined as follows:
   - A slow flash indicates that the node is discovered by the CMM and the device is in standby.
   - A fast flash indicates that the node is initializing.
   - A solid green LED indicates that the node is active.
5. When power is connected to the rack, all expansion and controller enclosures of Storwize V7000 storage are powered on automatically.

6. Proceed to power on each of the PureSystems Managers, one at a time. Press the Power button on the front of each device, as shown in Figure 3-28. Press each Power button only one time.

Note: PureSystem Managers are in Bay 2 of each chassis.

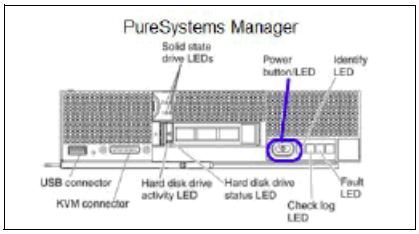


Figure 3-28 PureSystems Manager node

The PureSystems Manager starts first, and then the compute nodes automatically power on.

Note: The process of loading PureSystems Manager takes approximately 45 minutes and depends on the number of compute nodes in the system.

The following links provide information about powering off and restarting the PureApplication System:

- Powering off the system:

  https://www.ibm.com/support/knowledgecenter/#!/SSCR9A_2.1.0/doc/installguide/tpoweroffsys-gen2.dita

- Restarting the system:

  https://www.ibm.com/support/knowledgecenter/#!/SSCR9A_2.1.0/doc/installguide/trestart-gen2.dita

3.2.3 System diagnostic tests

When the PureApplication System is initialized and powered on, the IBM representative performs system diagnostic tests to confirm the overall health of the system. The diagnostic tests enable the specialist to detect major networking and hardware issues.

When the diagnostic tests complete with 0 errors, you can continue with the system installation, which includes the following items:

- VLAN setup
- Port setup
- Link configuration
- Connection to DNS servers
- Connection to NTP servers
- Creation of administrative password

3.2.4 Physical network connection between the appliance and the data center

When physical installation of the PureApplication System is complete, and the diagnostic and other installation steps finish successfully, the IBM representative performs the physical connection between the data center and the appliance.

This procedure includes two steps:

1. Installing transceivers of proper types in top-of-the-rack switches
2. Cabling the system to the data center

This process includes setting up the network connection for the management and data networks, as described in “Management network” on page 27 and “Data network” on page 27, and is done based on the client’s requirements and technical delivery assessment.

Consider which ports are being used by the system for inbound and outbound traffic, and configure the firewall. For more information about the required ports, go to the IBM Knowledge Center found at the following website:

https://www.ibm.com/support/knowledgecenter/#!/SSCR9A_2.1.0/doc/iwd/aar_portsover.dita

3.2.5 Test deployment

When the physical connection between the appliance and the data center is established and configured, the IBM representative deploys a test application to ensure that the PureApplication System operates.

To set up a test deployment, complete the following steps:

1. Go to PureApplication System and click **Patterns** → **Virtual System Patterns**. Select a test pattern from the list, as shown in Figure 3-29.

> **Note:** You can use a stand-alone WebSphere Application Server pattern to test deployment.

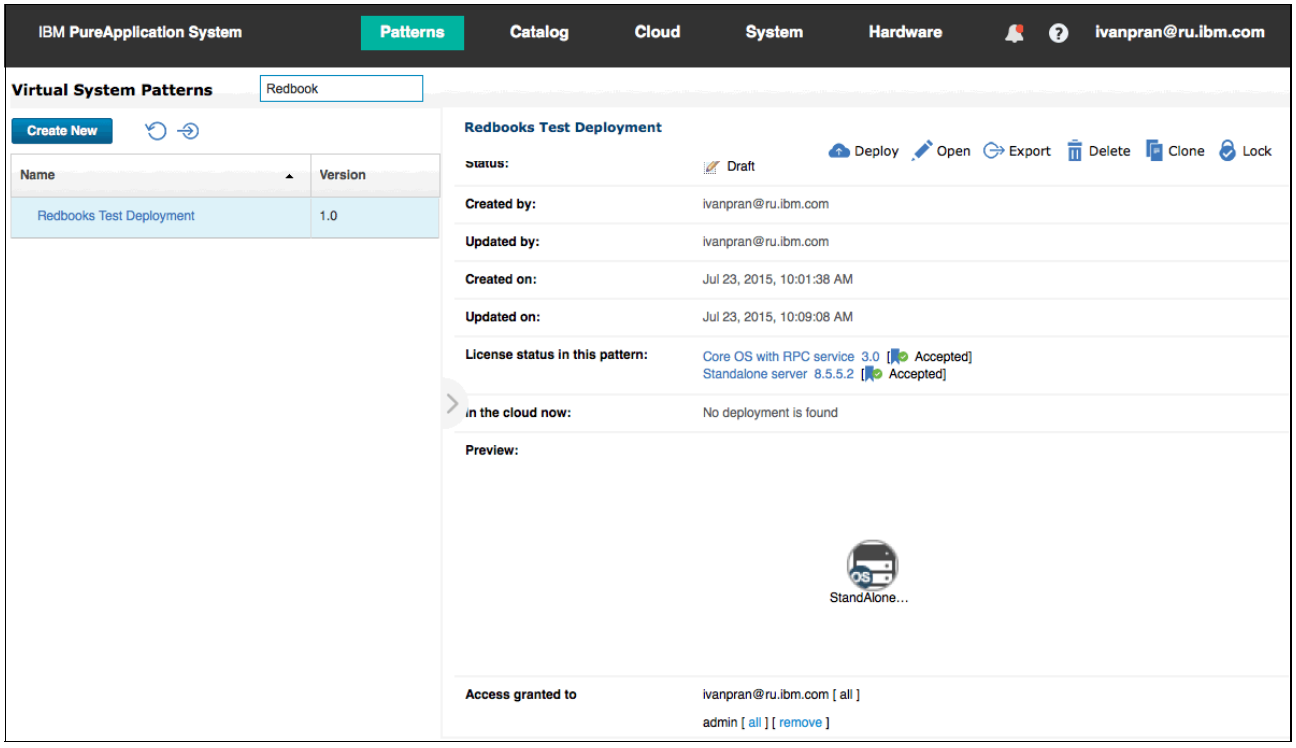


Figure 3-29 Pattern for test deployment

2. Click **Deploy**, and provide the required parameters for the deployment, as shown in Figure 3-30.

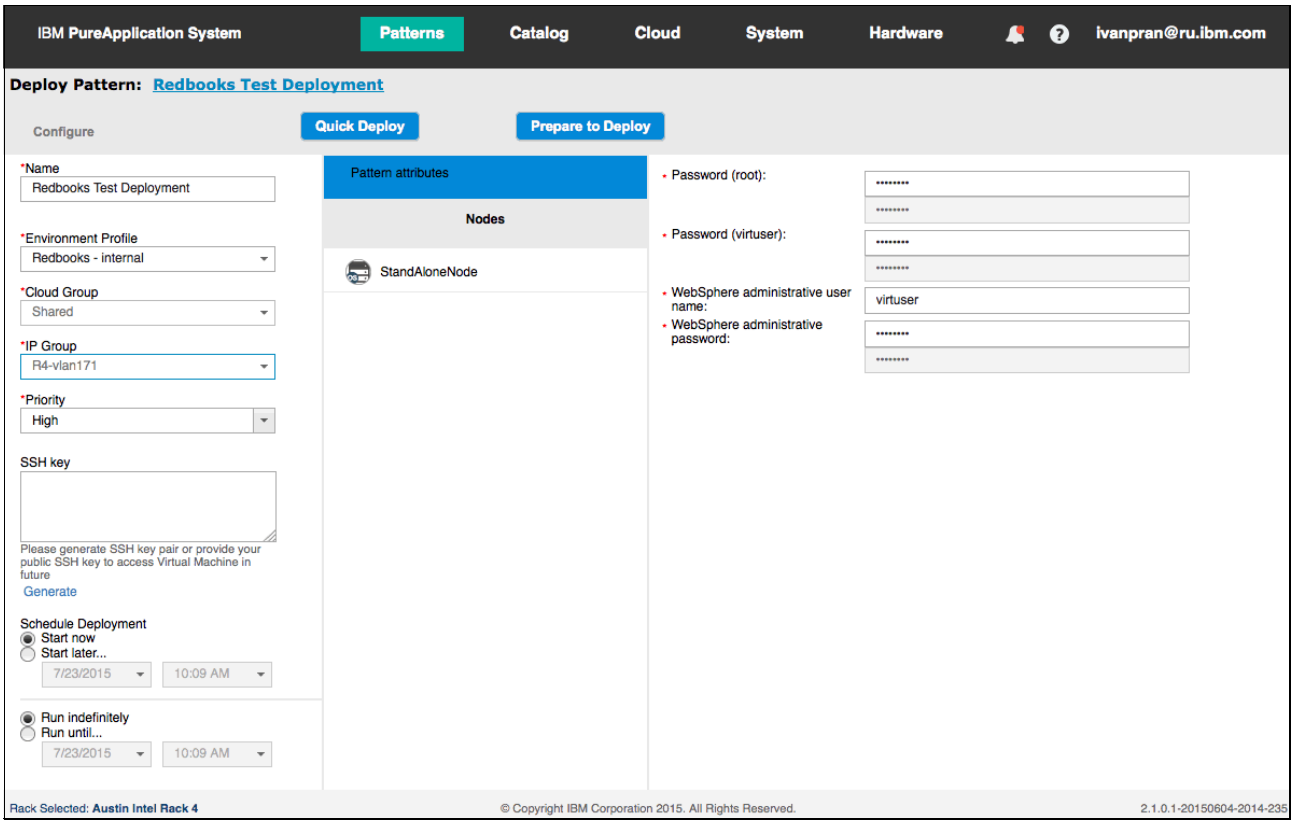


Figure 3-30 Parameters of test deployment

The test deployment is in the **Patterns** → **Virtual System Instances** menu of the GUI now, as shown in Figure 3-31.

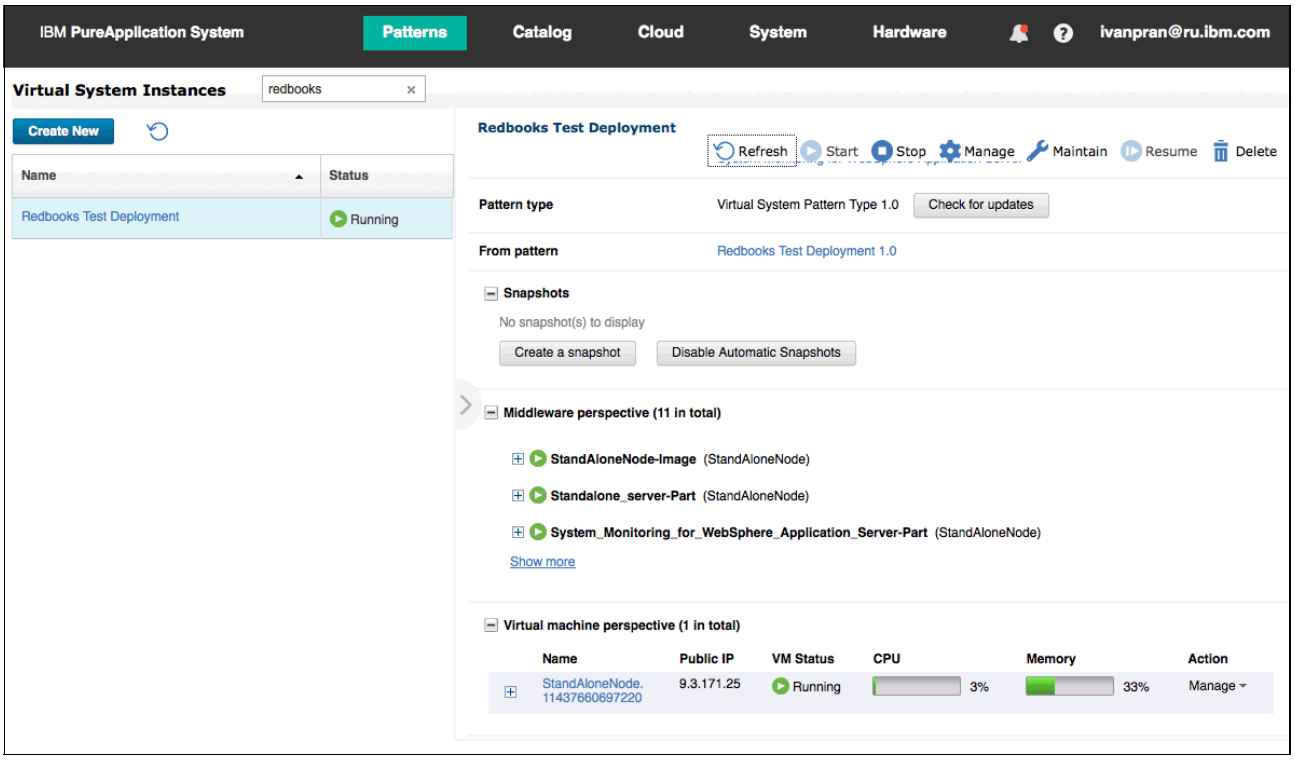


Figure 3-31 Instance of the test deployment

> **Note:** Deployment on a stand-alone WebSphere Application Server takes approximately 12 minutes.

3. Under the **Virtual machine perspective** (see Figure 3-31), click the VM name to view detailed information about the deployed VM and a link to the WebSphere Application Server console. Click this link to ensure that the instance deployed correctly and that its VMs are accessible.

For more information about virtual patterns and deployments in the PureApplication System, go to the following website:

`http://www.ibm.com/support/knowledgecenter/#!/SSCR9A_2.0.0/doc/iwd/par_vsysov.dita`

4

IBM PureApplication multi-system environment

A multi-system environment expands the benefits of a single system environment. Typically, customers who have more than one geographic location operate in a multi-system environment. This environment consists of two or more systems that are connected to each other physically (by communications equipment) and logically (in a defined relationship, as peer or host to other systems).

IBM PureApplication System V2.0 and later supports multi-system management and deployment. This feature was introduced in PureApplication System V2.0 and simplifies the implementation of high-availability configurations of IBM WebSphere Application Server, IBM Business Process Manager, IBM DB2, and others.

Note: For more information about high availability and disaster recovery in PureApplication System, see *Implementing High Availability and Disaster Recovery in IBM PureApplication Systems V2*, SG24-8246.

This chapter covers multi-system configurations and deployments with PureApplication System. It describes how to manage catalog content across PureApplication Systems and shows how to work with multi-target deployments.

The chapter covers the following topics:

- Introduction to the multi-system environment on the PureApplication platform
- Configuring a multi-system environment with PureApplication System
- Using a PureApplication System multi-system environment

© Copyright IBM Corp. 2015. All rights reserved.

4.1 Introduction to the multi-system environment on the PureApplication platform

You can use the multi-system environment on the PureApplication platform to manage environments centrally and deploy their applications across multiple systems by using a single user interface. The application architecture can have multiple virtual machines (VMs) that can be deployed across multiple systems in a flexible manner.

PureApplication System provides tools to simplify the management of multi-target deployments. Multi-system deployments can provide clear benefits when deploying certain IBM products through patterns. You need a minimum of two PureApplication Systems to start.

To configure a multi-system topology and prepare your PureApplication environment for multi-system deployments, you must perform some configuration steps. These steps are described in, 4.2, “Configuring a multi-system environment with PureApplication System” on page 74.

Figure 4-1 shows the relationships between the management domain and the deployment subdomains.

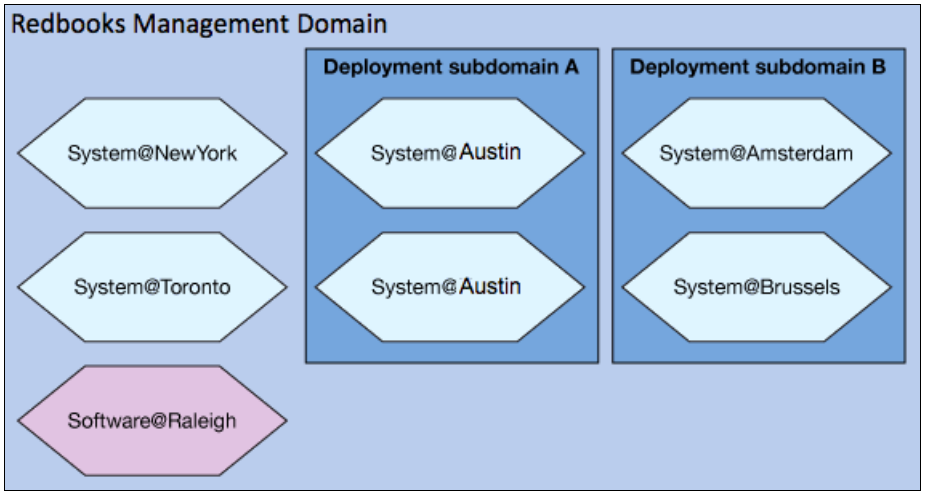


Figure 4-1 Relationship between the management domain and deployment subdomains

The multisystem environment with PureApplication System performs a number of tasks:

- Deploy VMs across multiple locations in a region.
- Determine whether catalog content across multiple system locations and data centers is synchronized.
- Deploy patterns across multiple locations in a deployment subdomain.
- Manage catalog artifacts at different levels of granularity, including details for a single item, across a management domain.
- Edit script packages in the catalog and synchronize the changes to all system locations in the management domain.

4.1.1 Management domain

A management domain creates a management relationship between two or more PureApplication Systems. When created, you can use a management domain to perform catalog content synchronization and user management across its members. An external LDAP repository is required for the management domain because all domain members must have the same set of users and users groups. You also need IP connectivity between the parts of the management domain. You can connect any number of systems in a management domain, and there is no limit to the distance between those systems.

Within a management domain, you can have one or more deployment subdomains.

Trust between systems in a management domain

PureApplication System uses preinstalled IBM self-signed console certificates by default. Your own SSL certificates are also supported. If you configured your own SSL certificate for the console, then you must import this certificate into the truststore for all other systems in the domain.

For more information about certificates in PureApplication System, see the following website:

https://www.ibm.com/support/knowledgecenter/#!/SSCR9A_2.1.0/doc/iwd/security_import_certs.dita

If you use your own SSL certificate, you must import this certificate into the truststore for all other systems in the domain.

Figure 4-2 shows networking between two PureApplication Systems in a management domain.

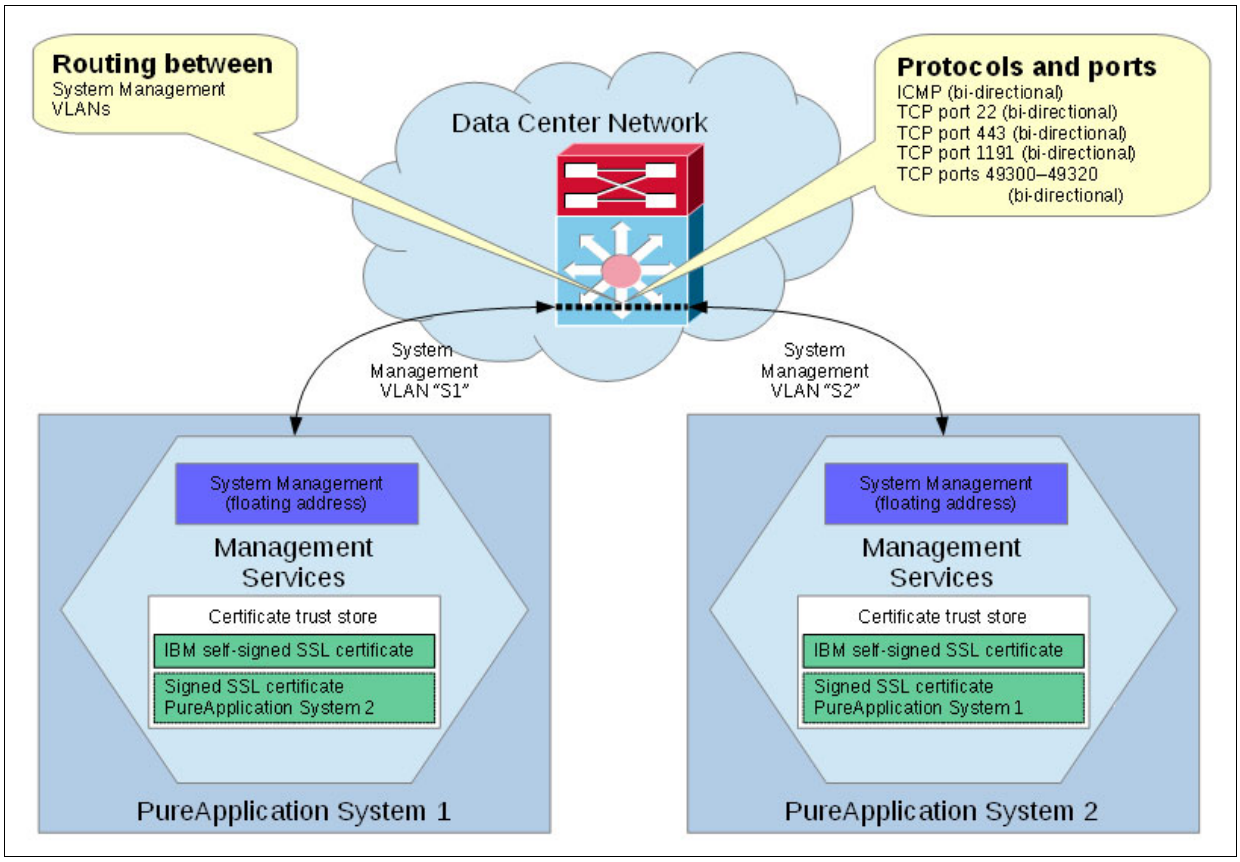


Figure 4-2 Networking within a management domain

To import a console certificate into the truststore for all other systems in the domain, complete the following steps.

Note: This action should be performed before adding systems to your domain or importing a new console certificate, and after every upgrade of the PureApplication Systems in your domain.

1. Save the certificate for either the system that is being added to the domain or whose console certificate is being updated. If you have a copy of the certificate, use its `.crt` file. Alternatively, you can obtain the current certificate from the system by going to the system with your browser. From that point, click the browser's security icon, and export the certificate as a `.crt` file.
2. Use the command-line interface (CLI) to find the location name for the system that is being added to the domain or whose console certificate is being replaced. You can use the following command:

   ```
   $ bin/pure -h RedbooksRack1_IP -u user -p password -c "print
   admin.racks[0].location_name"
   ```

   As the result of this command, you get the serial number of the rack. You see a response similar to this return:

   ```
   1000202
   ```

You can also obtain the location of the system by clicking **Hardware** → **Infrastructure Map**, from the PureApplication System GUI, as shown in Figure 4-3.

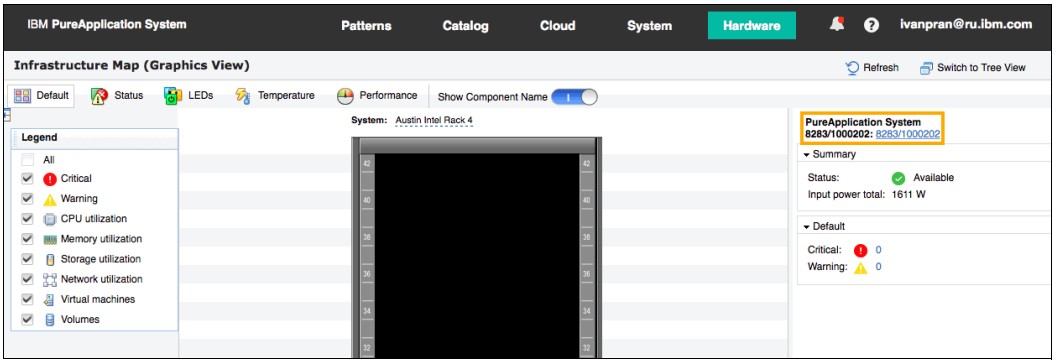


Figure 4-3 System location in the GUI

3. Using the location name that was obtained in step 2 on page 72, use the CLI to import this system's certificate to all other systems in the domain. You can use the following commands to import the certificate:

   - **$ bin/pure -a -h RedbooksRack2_IP -u user -p password -c**
   - **"deployer.peercertificate._import({'certfilepath':'RedbooksRack1.crt','peernam e':'1000202'})"**
   - **$ bin/pure -a -h RedbooksRack3_IP -u user -p password -c**
   - **"deployer.peercertificate._import({'certfilepath':'RedbooksRack1.crt','peernam e':'1000202'})"**

   For these commands, you must have either the hardware administration role with permission to Manage hardware resources, or the security administration role with permission to Manage security. For more information about users roles, see the following website:

   https://www.ibm.com/support/knowledgecenter/#!/SSCR9A_2.1.0/doc/iwd/aat_manusui.dita

4. Repeat this process as needed to import the remaining systems' certificates to other systems in the domain.

4.1.2 Deployment subdomain

You can use a deployment subdomain to deploy patterns and shared services across two PureApplication Systems. You can define any number of deployment subdomains within a management domain.

You can add up to two locations to a subdomain in your management domain configuration. The locations in each subdomain share deployment workloads in your multisystem environment.

Each deployment subdomain can consist of only two PureApplication Systems. Systems in the subdomain must be connected to each other by a low-latency network. Generally, distance between the systems should not exceed 300 kilometers. In addition to these requirements, an external iSCSI tiebreaker is required in case the systems from the deployment subdomain lose connectivity to each other.

4.1.3 Externally managed cloud groups and environment profiles

By default, PureApplication System uses internal IPv6 management VLANs for communication between management nodes (PureSystem Managers), VMs, and shared services running within cloud groups.

To support multi-system deployment, externally managed cloud groups were introduced in Version 2.0. Instead of using an internal IPv6 VLAN for management communication, these cloud groups use an external management VLAN, which is required for management communication between the two systems within a deployment subdomain.

Internally managed cloud groups continue to be supported fully on PureApplication System. However, externally managed cloud groups require multi-system deployment (they have some limitations compared to internally managed cloud groups). Therefore, you must evaluate your deployment needs and carefully plan the allocation of your compute nodes to internally or externally managed cloud groups, or both. You cannot convert existing cloud groups from one type of management to another.

4.2 Configuring a multi-system environment with PureApplication System

This section describes the required steps to set up a multisystem environment with PureApplication System. For the purposes of this publication, two Intel based PureApplication Systems running at the 2.1.0.1 firmware level were used.

4.2.1 Rules for a multi-system environment

The following rules apply when setting up a management domain and deployment subdomains:

- Your system (PureApplication System) or instance (PureApplication Software) can belong to, at most, one domain.
- PureApplication Software instances can belong to a domain but not to a subdomain. (Multi-system deployment is not supported for PureApplication Software.)
- You can create one or more subdomains, which are contained within the domain.
- PureApplication Systems within the domain can belong to, at most, one subdomain within the domain.
- Your subdomain can contain only two systems, but there is no limit to the number of subdomains that you can create in your domain.
- Because of latency limitations for the subdomain communications, the systems in your subdomain should not be more than 300 km apart.
- You must have the same platform type within each management domain. IBM PureApplication System models W1500 and W2500 can coexist in a domain, together with PureApplication Software. Similarly, IBM PureApplication System models W1700 and W2700 can coexist in a domain. However, you cannot have an Intel based system (W1500/W2500) and a POWER-based system (W1700/W2700) in the same management domain.
- After a management domain exists, you cannot remove a system from there if that system is also part of a deployment subdomain. Furthermore, you cannot remove a system from a deployment subdomain without deleting the subdomain.

Note: To delete the subdomain, all pattern instances and shared services that are deployed within that subdomain to externally managed environment profiles first must be deleted. This situation is true even for externally managed deployments whose VMs are on only one of the systems in the subdomain. So, plan and design your multisystem environment before implementation.

4.2.2 Required user permissions

To add a system to the management domain, the user must adhere to the following items:

- Have both full security and full hardware administration permissions on the system from which they perform the addition.
- Provide user credentials having full security and full hardware administration permissions on the system that is being added to the domain.

Adding the system to the domain creates a relationship of trust between the system and all other existing systems in the domain.

All other domain and subdomain operations (create subdomain, add system to subdomain, delete subdomain, and remove system from domain) require only hardware administration authority, and only on the system that is performing the operation.

For more information about security roles in PureApplication System, see the following website:

https://www.ibm.com/support/knowledgecenter/#!/SSCR9A_2.1.0/doc/iwd/aac_user_permissions.dita

4.2.3 Management VLAN for an externally managed cloud group

The first step to set up the multi-system environment with PureApplication System is to create an external VLAN and a link configuring the VLAN to be connected to the PureApplication System. Complete the following steps:

1. Under Required Settings, click **System** → **Network Configuration**. Then, click **VLANs and Links**. Figure 4-4 shows the VLANs and Links tab.

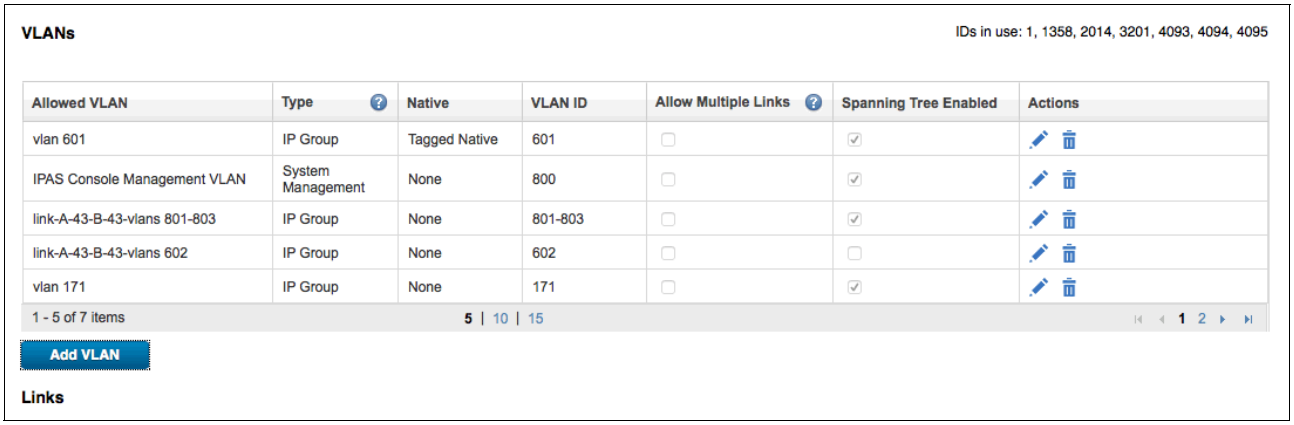


Figure 4-4 VLANs and Links tab of the PureApplication GUI

2. In the VLANs section, add a VLAN by clicking **Add VLAN**.

Note: A VLAN of type External Storage is displayed. This VLAN is predefined on the system for external storage configuration and should not be changed.

The maximum number of VLANs that you can configure on the system is 32.

3. Specify the VLAN name to identify your new VLAN group.
4. Specify the VLAN type by selecting one of the available options:
   - **System Management**: This VLAN type is intended for accessing the console and to provide a way to send all network traffic outside the rack. You can define only one VLAN with this type for the entire system. A single VLAN ID is required for the VLAN that is defined.
   - **IP Group**: This VLAN type is intended for defining IP groups. You can define more than one VLAN with this type, and you can specify ranges of numbers for the VLAN IDs.

     Note: To set up a multi-system environment, you must use the IP Group VLAN type.

5. Specify the native setting by selecting one of the available options:
   - **None**: Use this VLAN type to define a VLAN that is not a native VLAN.
   - **Untagged Native**: This VLAN type is referred to as the native VLAN ID for a trunk port. The native VLAN ID is the VLAN that carries untagged traffic on trunk ports. Use this VLAN type if you do not use `PVID=1` as the native VLAN. You can define more than one VLAN with this type. A single VLAN ID is required for the VLAN defined.
   - **Tagged Native**: Use this VLAN type if you require that all packets are 802.1Q tagged packets. For Cisco, this definition is a global definition. For IBM PureApplication System 8283, this VLAN type is defined on a *per link* basis. You can define more than one VLAN with this type. A single VLAN ID is required for the VLAN defined.
6. Specify the VLAN ID as a single numeric value. Then, specify whether you want the VLAN to allow multiple links.

   When you define a link, each VLAN definition is available for selection once. To allow the use of the VLAN in two links, select the **Allow Multiple Links** check box. Use this option if you have HSRP networks and you need to define the same VLAN in multiple links.
7. Specify whether you want to enable spanning tree for the VLAN.
8. Select the **Spanning Tree Enabled** check box to define Per VLAN Rapid Spanning Tree (PVRST) for the VLAN. PVRST is compatible with RST+.
9. After you select all options, click **Confirm**, then click **Save** to add the VLAN, as shown in Figure 4-5 on page 77.

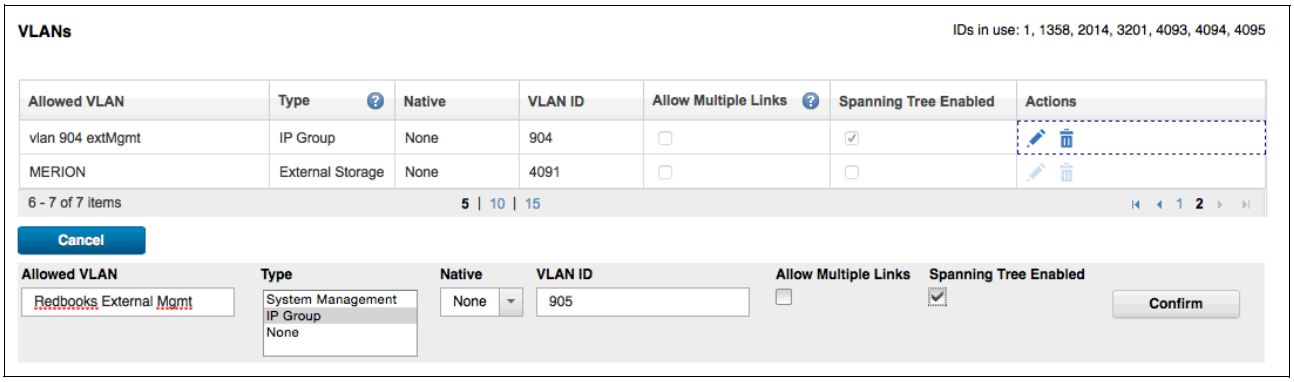


Figure 4-5 Creation of a VLAN in the PureApplication GUI

4.2.4 Configuring additional IP addresses for cloud management

One or two additional IP addresses are required for PureApplication System before it can be added to the management domain. These addresses are used for the workload management of deployed VMs in a multi-system environment. Figure 4-6 shows how the management IP address on the system management VLAN interacts with a VM deployed on PureApplication System.

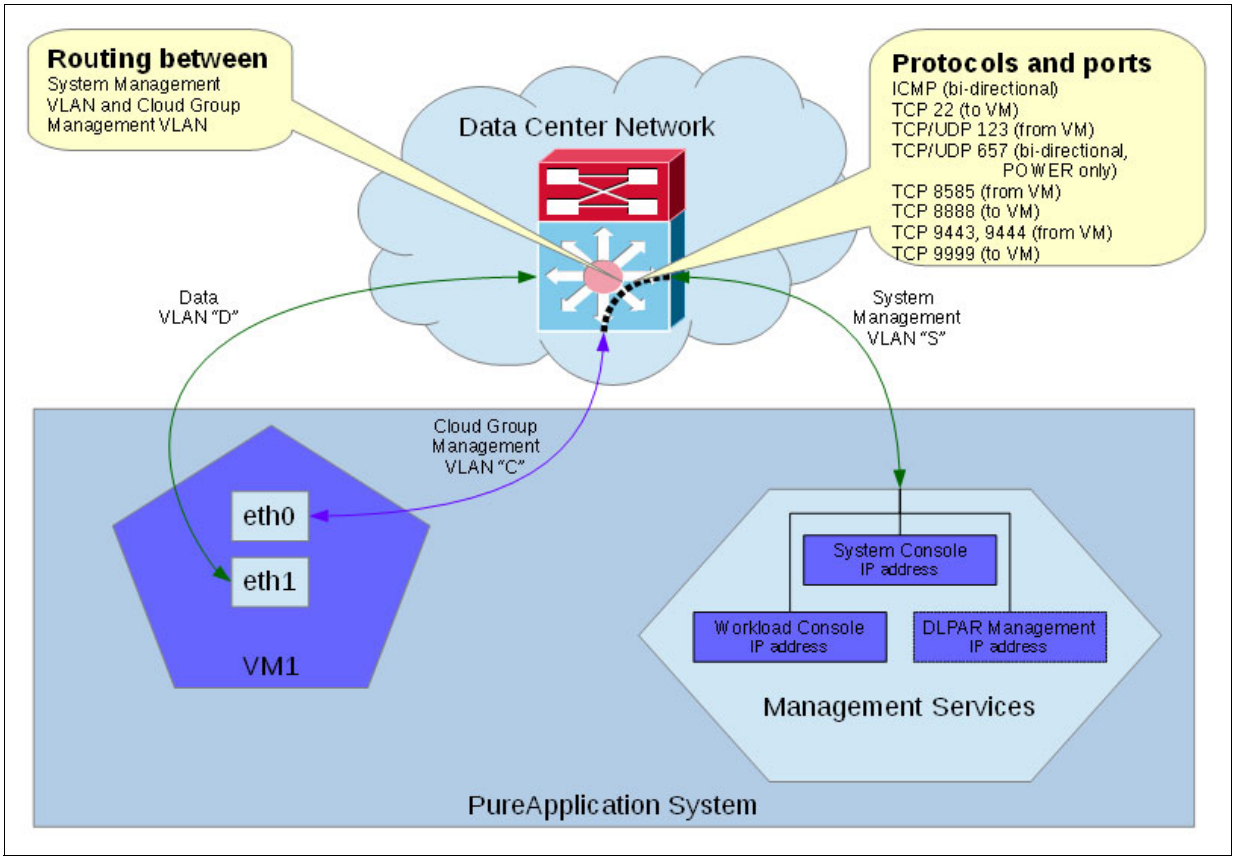


Figure 4-6 Interaction between management IP address and VMs

Additional IP addresses can be configured in the PureApplication GUI by clicking **System** → **Network Configuration**. Then, set the configuration in the Additional IPs for Cloud Management by way of External Networks window, as shown in Figure 4-7.

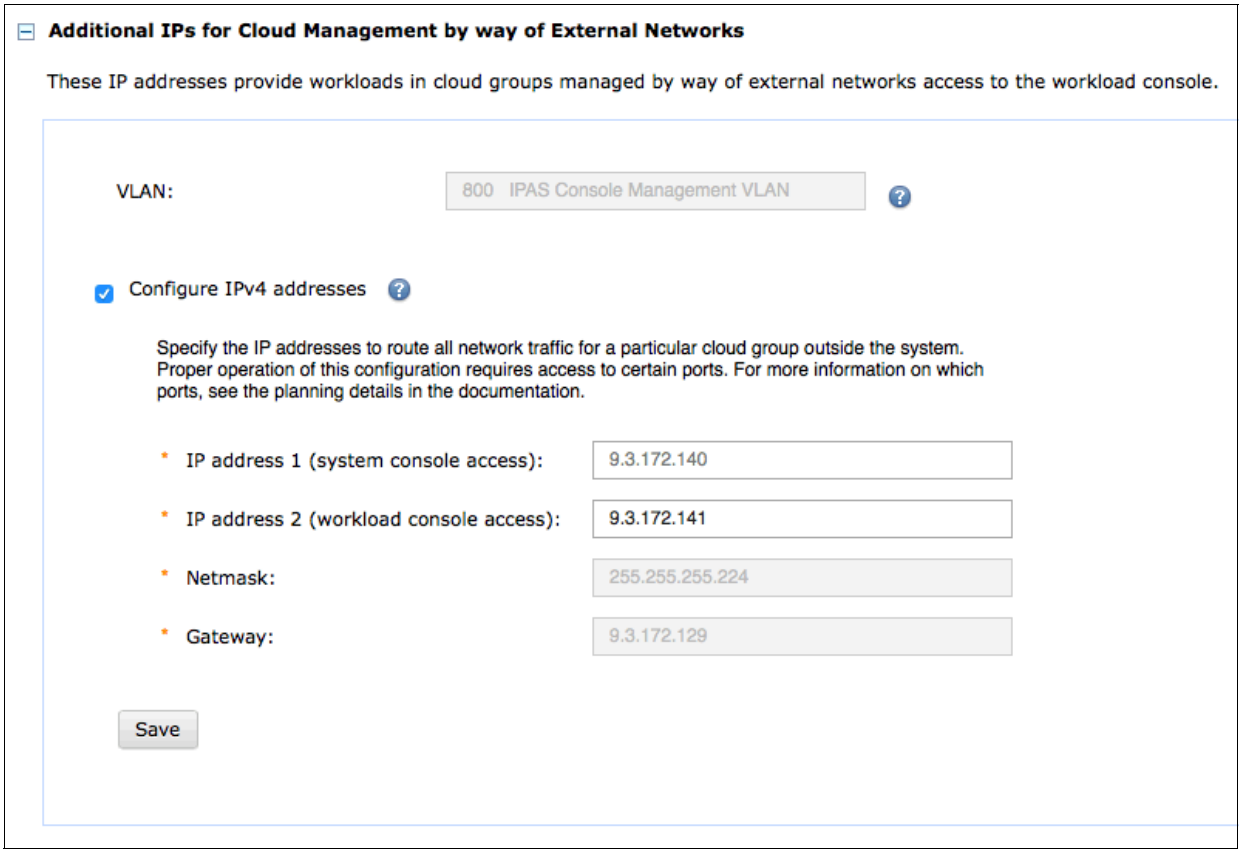


Figure 4-7 Additional IP for Cloud Management section of PureApplication W1500/2500 GUI

PureApplication System on Intel (models W1500 and W2500) must be configured with one additional IP address, as shown in Figure 4-7. PureApplication System on POWER (models W1700 and W2700) must be configured with two additional IP addresses. In both cases, the new addresses must be defined in the same subnet as the existing system management IP addresses. The additional management addresses must be IPv4 addresses only.

Note: Additional IP addresses are not required for PureApplication Software.

You must ensure that the PureApplication Systems (or PureApplication Software instances) in your management domain share IP connectivity between all of their system management IP addresses. If you have a firewall in place between the system management VLANs of both, you should ensure that ICMP traffic and TCP traffic is permitted.

For more information about service and port requirements when configuring an externally managed cloud, see the following website:

https://www.ibm.com/support/knowledgecenter/SSCR9A_2.1.0/doc/systemconsole/t_configcloud.dita?lang=en

4.2.5 Creating the management domain

When your PureApplication System is enabled to be externally managed, you can create a Management domain. Complete the following steps:

1. Click **System** → **Management Domain Configuration**.
2. Click **Create Management Domain**. Then, provide a name for the management domain and select **OK**, as shown in Figure 4-8.

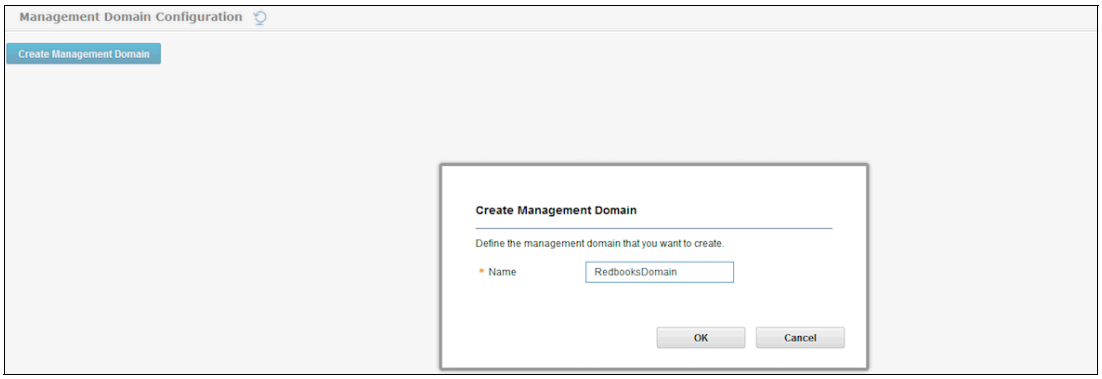


Figure 4-8 New management domain

3. Add systems in the management domain. Select **Add a location**, which adds a system to the domain. Enter the shared floating IP address that you used for the workload console and the system console for the remote system. The user does not have to be an LDAP user, but must be the administrative user that establishes trust.

Note: The user must have permissions for hardware resources and security.

4.2.6 Creating the deployment subdomain

After you create the management domain, you create a deployment subdomain. To do so, complete the following steps:

1. Click **System** → **Management Domain Configuration** and select the **Create Deployment Subdomain** icon, as shown in Figure 4-9.

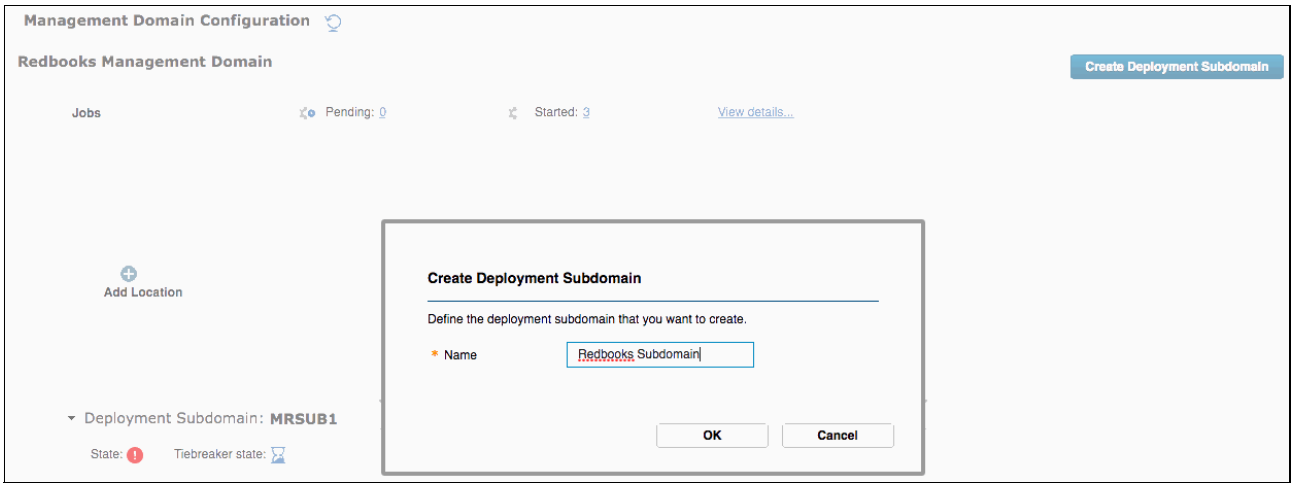


Figure 4-9 New deployment subdomain

2. After you create the deployment subdomain, you can add a location to it as needed to continue your multi-system configuration. You can add a location by either dragging the location into the subdomain, or by clicking the **Add a Location** button in the subdomain.

Note: From the available list of local and remote locations in your management domain, you can add up to two locations to a deployment subdomain. However, a location can be added to only one deployment subdomain. After the location is part of a deployment subdomain, it is no longer available to be added to another deployment subdomain. In addition, only the locations of type system can be added. Locations of type software are not supported.

After you add a location to any deployment subdomain, you cannot remove it. You must delete the deployment subdomain to return the location to the management domain and make it available for use in another deployment subdomain.

4.2.7 iSCSI tiebreaker

When you add the second location to any subdomain, you are prompted to configure the iSCSI tiebreaker, as shown in Figure 4-10.

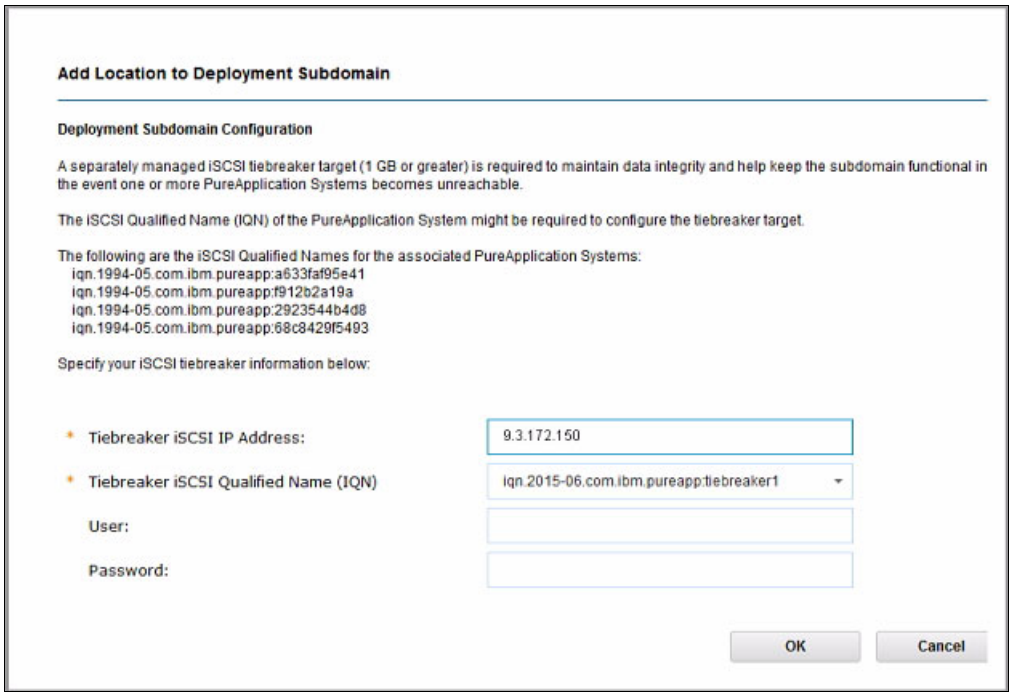


Figure 4-10 Tiebreaker configuration

Note: User and password are optional in the tiebreaker configuration window.

iSCSI tiebreaker is used when the members in the subdomain lose connectivity to each other. The two PureApplication Systems in your subdomain share information about deployed instances in real time. This data is stored in a new partition (LUN) called *SubdomainData* that is created on the master storage node of each system at the time the subdomain is created. At least 512 GB of storage must be available on each system before it can be added to the subdomain. You can verify the amount of available storage capacity by clicking **Hardware** → **Storage devices**, as shown in Figure 4-11.

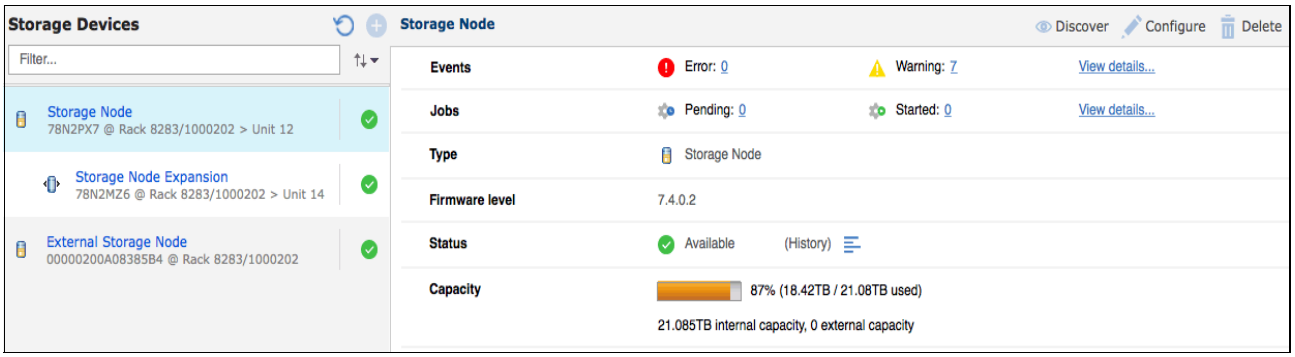


Figure 4-11 Storage Devices tab of the PureApplication System GUI

Systems in the subdomain must be part of a low latency network because IBM GPFS™ is used to synchronize storehouse data between systems. The distance between the systems should not exceed 300 kilometers.

The external iSCSI target is required if your PureApplication Systems lose communication with each other. The tiebreaker resolves the issue by giving control to one of the systems. Your external tiebreaker should have a minimum size of 1 GB and fulfil the same distance and latency requirements.

Figure 4-12 shows the connectivity between two PureApplication Systems and an external iSCSI tiebreaker.

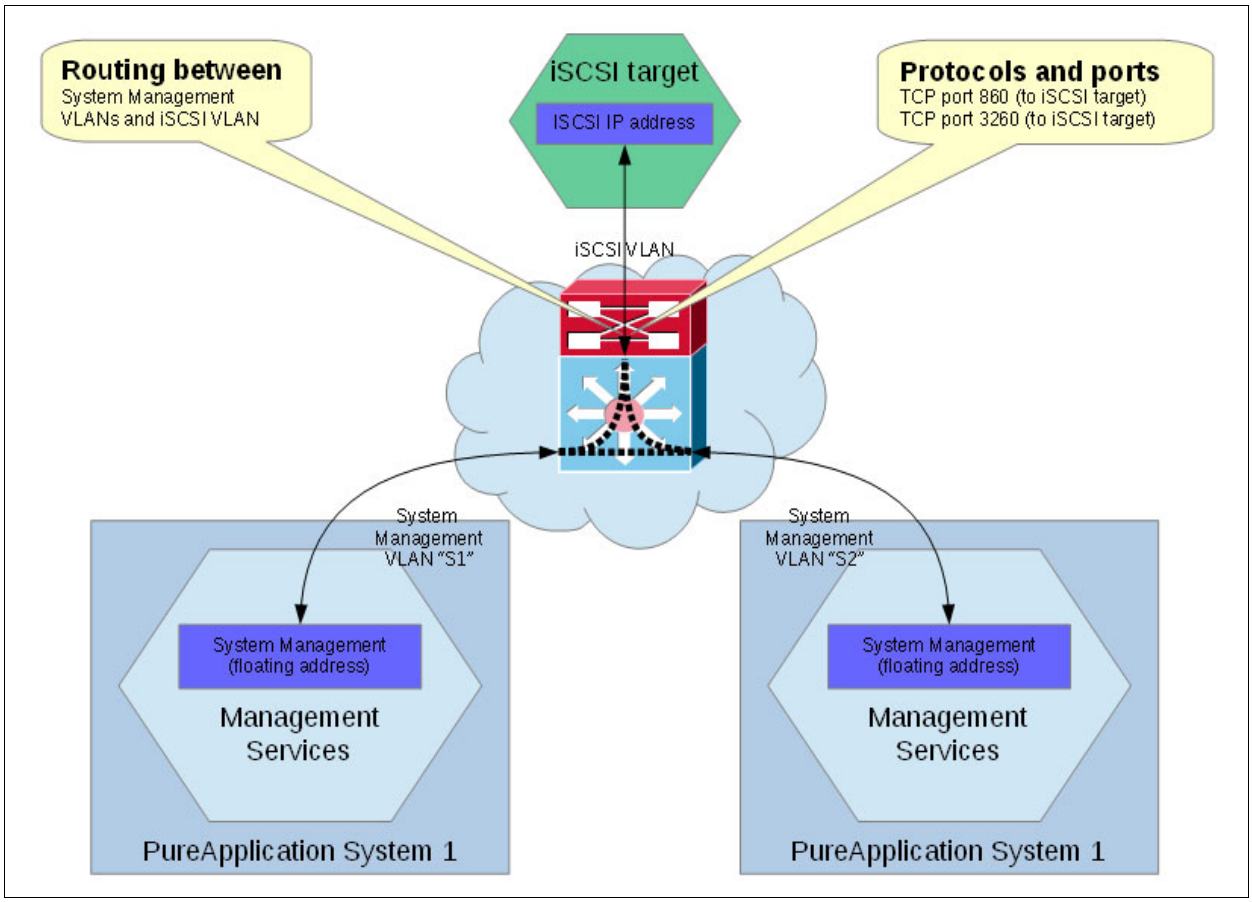


Figure 4-12 Communication with an external iSCSI tiebreaker

If both systems lose access to the tiebreaker but can still communicate with each other, the subdomain availability is not affected. If the systems lose communication with each other but can communicate with the tiebreaker, the winning system still can maintain the subdomain. On systems where the subdomain is unavailable, deployments to external environment profiles are effectively frozen. Existing deployments cannot scale, new deployments cannot be issued, and deployments cannot be deleted.

Note: A preferred practice is to use scsi-target-utils version 1.0.24 or newer on Linux.

The deployment data that is stored and shared on this internal mirror is not encrypted. Because this data is mirrored between the systems (when at rest on the system storage controllers or in transit between the system management IP addresses), it is unencrypted. This data includes secure keys that are used by the system to manage the deployed instances. You should plan for adequate physical security for the systems and adequate network security for the data in transit.

After configuring the tiebreaker and addressing related concerns, you have a new management domain with a deployment subdomain in it, as shown in Figure 4-13 on page 83.

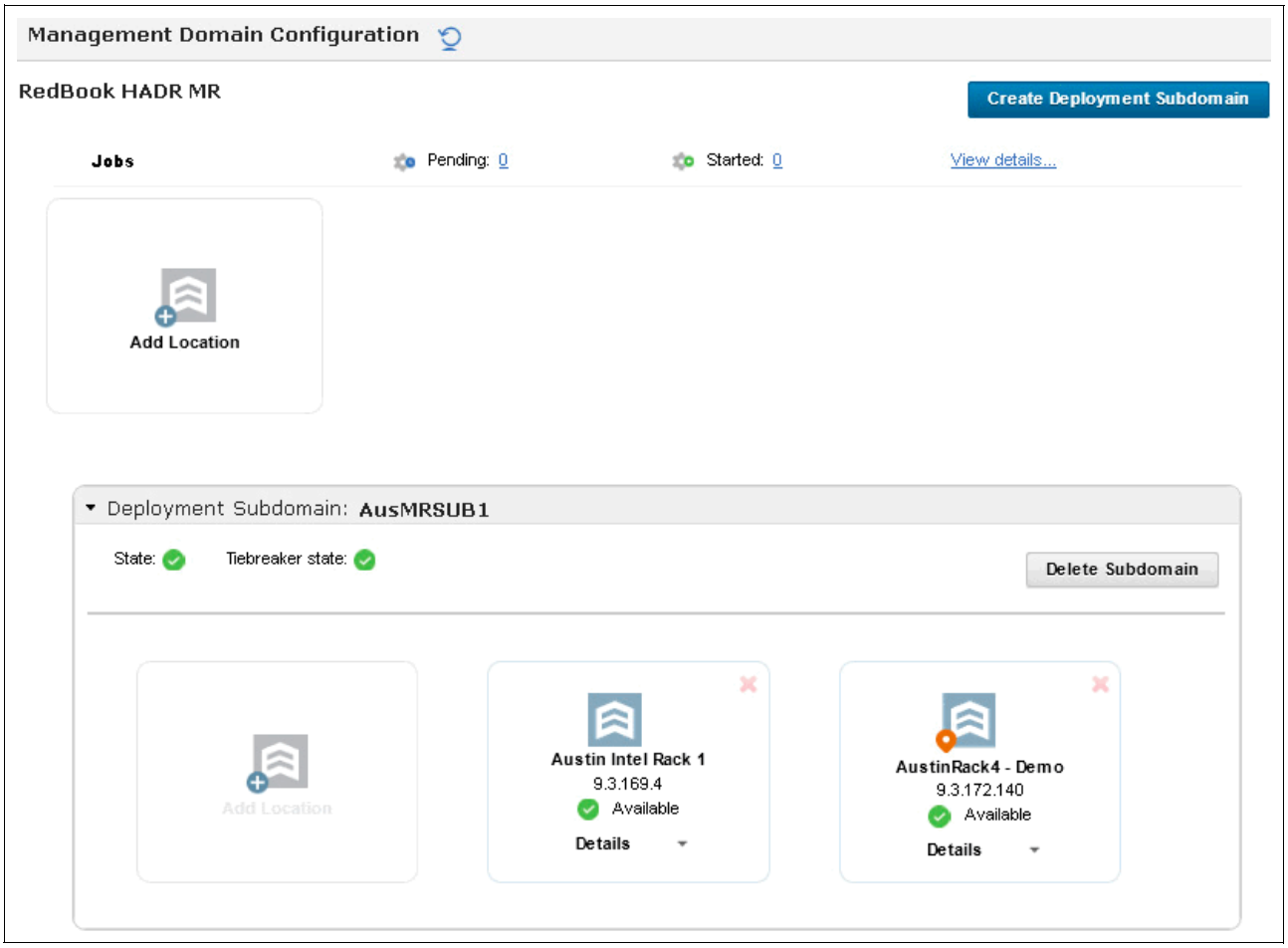


Figure 4-13 Deployment subdomain as a part of management domain

4.2.8 Configuring cloud resources for a multi-system environment

To deploy patterns on the PureApplication System by using a multi-system environment, you must configure *cloud resources*. These resources are noted in the following list:

- IP groups
- Cloud groups
- Environment profiles

IP groups setup

You must configure a new IP group for the purpose of *cloud management*. Complete the following steps:

1. Click **Cloud** → **IP Groups** in the PureApplication GUI. Create an IP Group and, from the Used For drop-down menu, select **Cloud Management**, as shown in Figure 4-14.

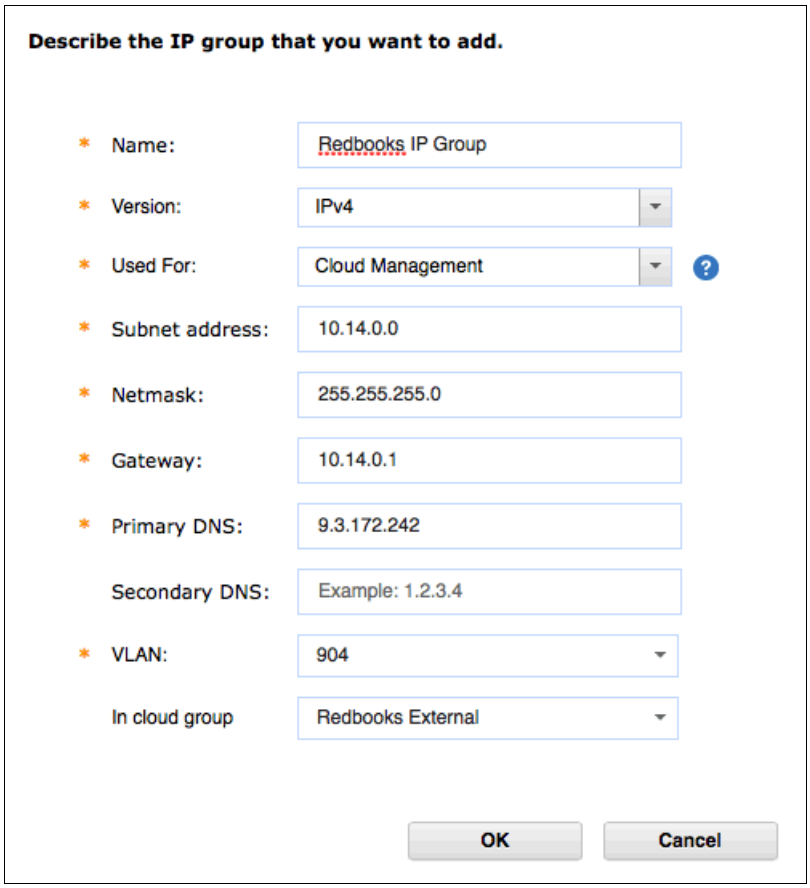


Figure 4-14 New IP group for cloud management

> **Note:** Assign this IP Group to the cloud group management VLAN you configured earlier.

2. Add IP addresses into your new IP group for cloud management. For more information about that process, see the following website:

   https://www.ibm.com/support/knowledgecenter/api/content/nl/en-us/SSCR9A_2.1.0/doc/iwd/srt_addipui.html

3. The default route for your VMs is normally assigned to the first data interface (eth1 or en1). Because of this default, you must now configure additional routes for the management interface eth0 or en0 to ensure that management traffic is not routed over eth1 or en1. You can define additional routes as part of the IP group definition.

You must configure the following routes for each cloud management IP group:

– A subnet route that uses the IP group's gateway address to the network of the local system management VLAN.

– A subnet route that uses the IP group's gateway address to each network that is used by any other cloud management IP group with which this IP group shares an environment profile. This route is required only when the different cloud management IP groups do not share the same VLAN.

There is no need to configure a route to the System Management network of the remote system.

As noted earlier, Layer-3 network routing must be in place within the data center network between the various VLANs. Figure 4-15 is an example of this scenario with two PureApplication Systems.

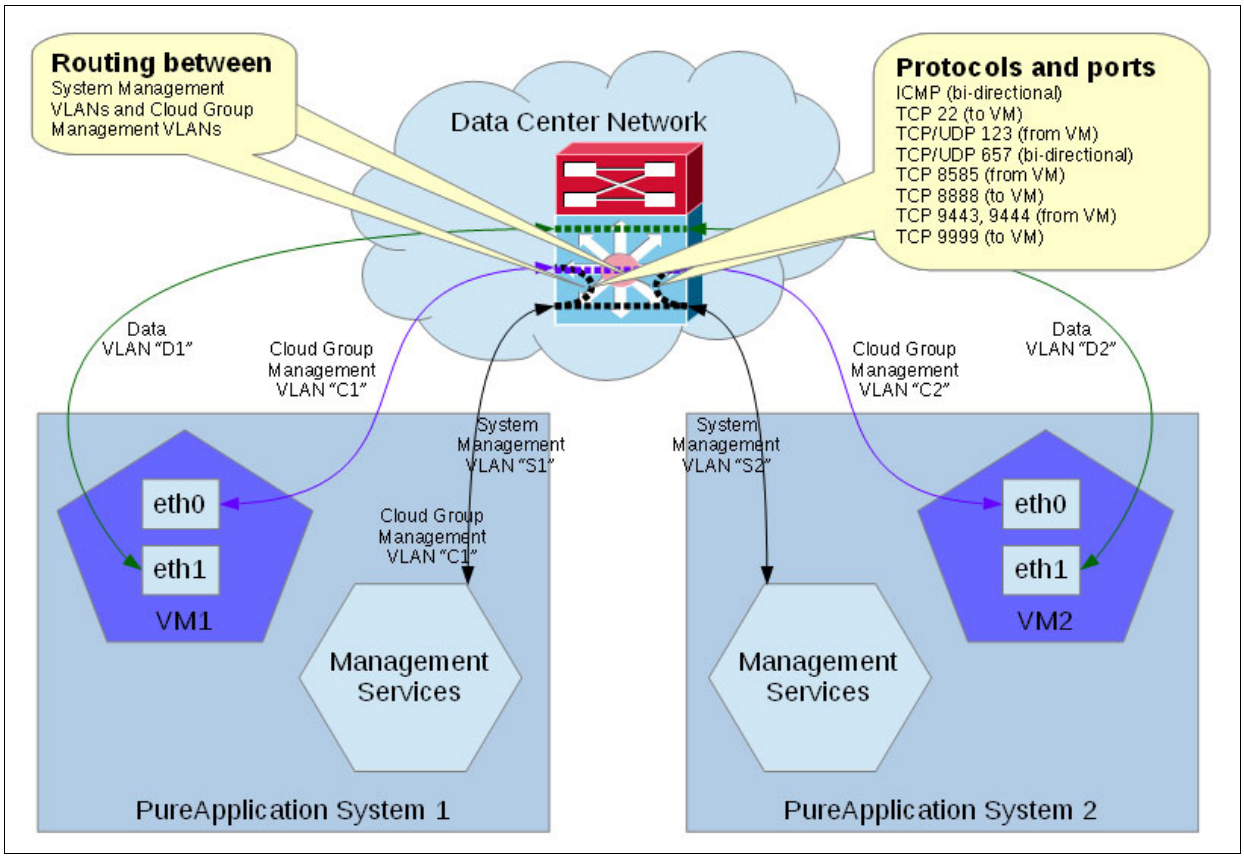


Figure 4-15 Routing between two PureApplication Systems

Figure 4-15 shows example routing for the following VLANs:

- Cloud group management VLAN 1 and System Management VLAN 1
- Cloud group management VLAN 2 and System Management VLAN 2
- Cloud group management VLAN 1 and cloud group management VLAN 2
- System Management VLAN 1 and System Management VLAN 2
- Data VLAN 1 and Data VLAN 2

Cloud groups setup

For cloud group setup, a new option is available that is called *Cloud Management*. The option includes *By way of internal network* and *By way of external network*. For the internal option, management data travels on the internally managed VLAN. For the external option, which is used in a multisystem environment, management data travels on the data center network and requires a cloud management IP group.

To create an external network option, complete the following steps:

1. To create a cloud group for a multi-system environment, click **Cloud** → **Cloud Groups**. Then, from the Cloud Management drop-down menu, select **By way of External Network**, as shown in Figure 4-16.

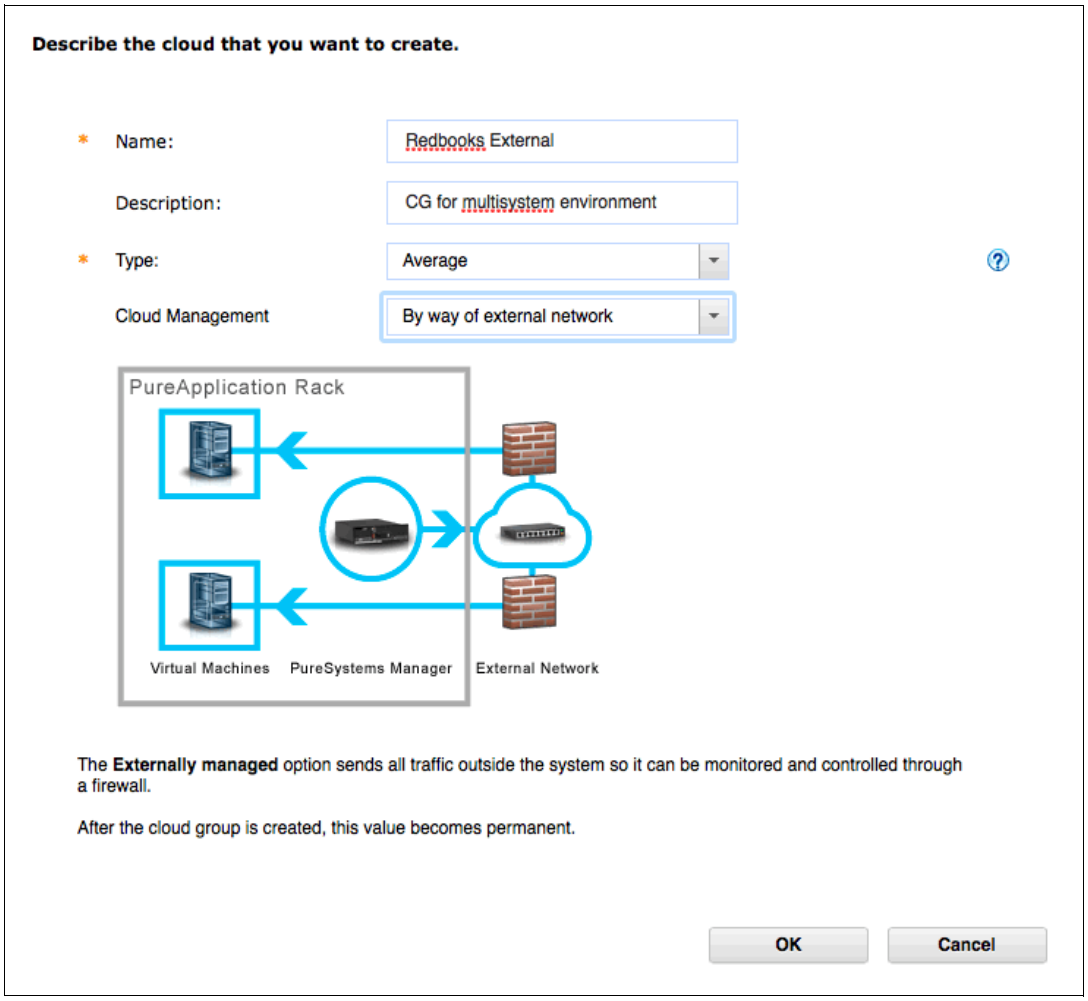


Figure 4-16 New cloud group for multi-system environment

2. To the cloud group, add a minimum of one compute node, one data IP group, and one cloud management IP group. These minimums make your cloud group available for multi-system deployments, as shown in Figure 4-17 on page 87.

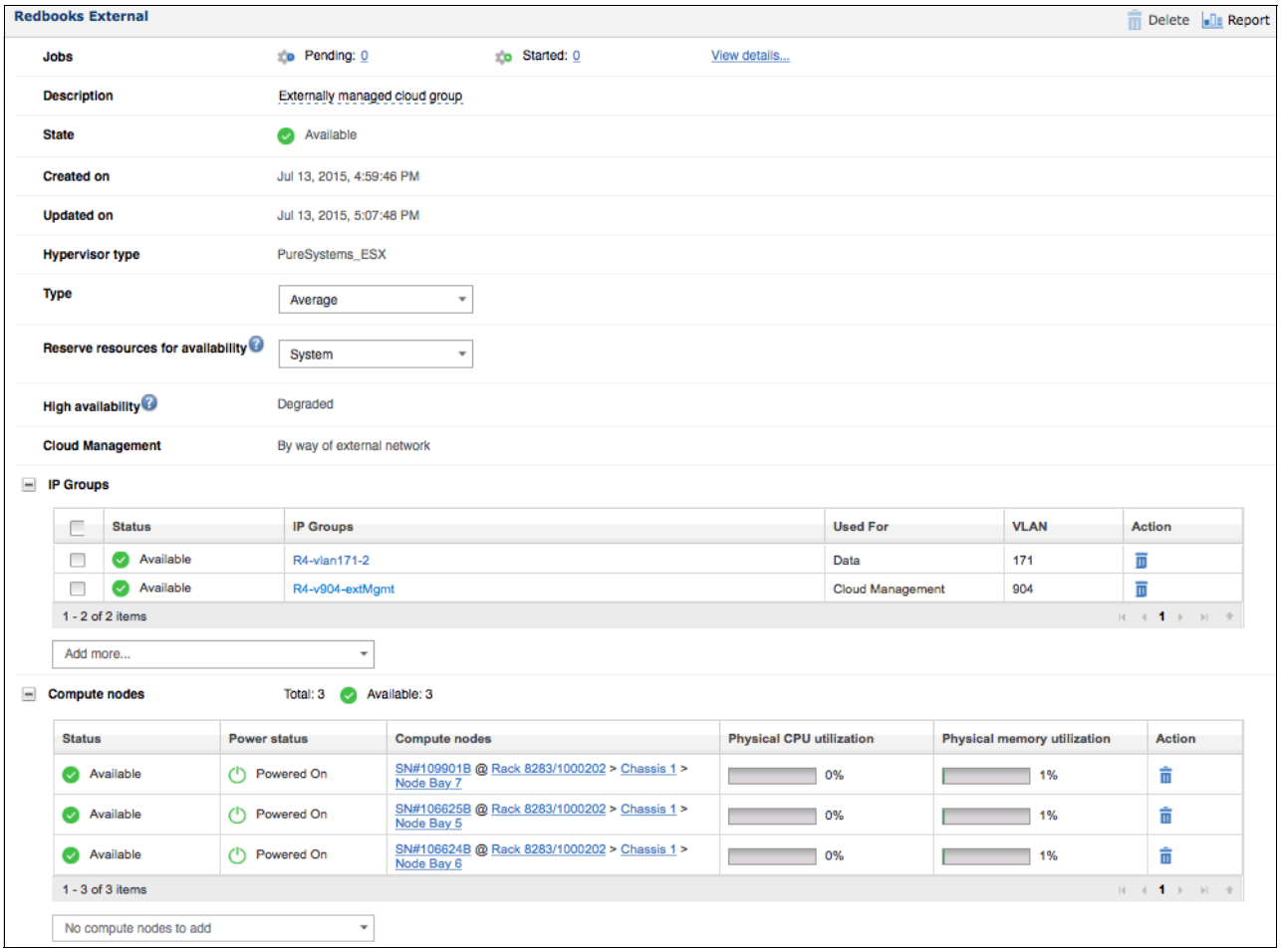


Figure 4-17 New cloud group for multi-system deployments

When you create an externally managed cloud group, you do not select a VLAN ID. Instead, you create one or more IP groups that are designated to be used for cloud management instead of data. These IP groups are used to assign IPv4 addresses to the management interfaces of VMs that are deployed in these cloud groups. As with IP groups that are used for data traffic, the addresses in your cloud management IP groups must have both forward and reverse lookups that are defined in your DNS.

The system does not enforce any requirements for the VLANs that are used for management IP groups. You can create VLANs for VM management use only, or you can share VLANs with your data IP groups. In either case, your externally managed cloud groups are not marked *available* until they have IP groups that are defined for both management and for data.

For more information about configuring a multisystem environment, see the following website:

https://www.ibm.com/support/knowledgecenter/#!/SSCR9A_2.1.0/doc/iwd/msc_configuration_ov.dita

Environment profiles

To tie together the IP group and cloud group for multi-system deployments, you must create a new *environment profile.* To do so, click **Cloud** → **Environment Profiles**, as shown in Figure 4-18.

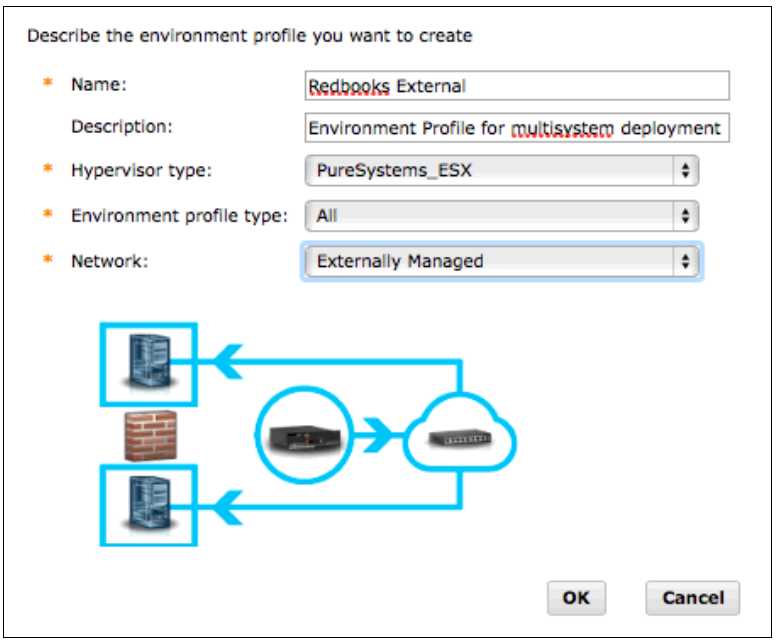


Figure 4-18 Environment profile for multi-target deployments

As shown in Figure 4-18, the option for Network defines whether VMs are deployed to the internally managed network or an externally managed network. For multi-system deployments, you must use the Externally Managed option.

In this process, you also need to specify a minimum of one cloud group to make your new environment profile available, as shown in Figure 4-19.

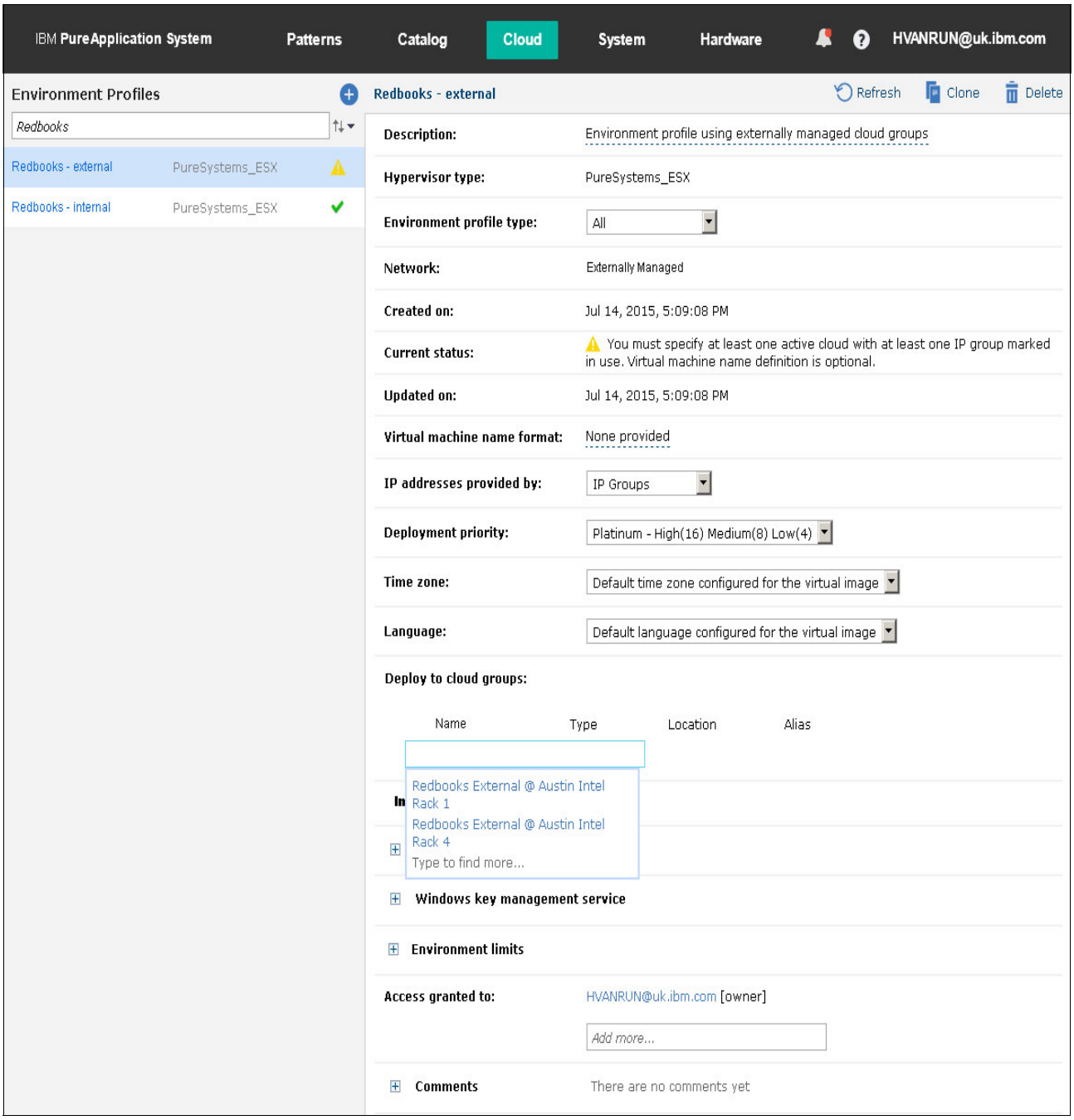


Figure 4-19 Create an environment profile for multi-system deployments

After ensuring the cloud group and network options, and finishing the finer details in the configuration, your environment profile is available for multi-system deployments, as shown in Figure 4-20.

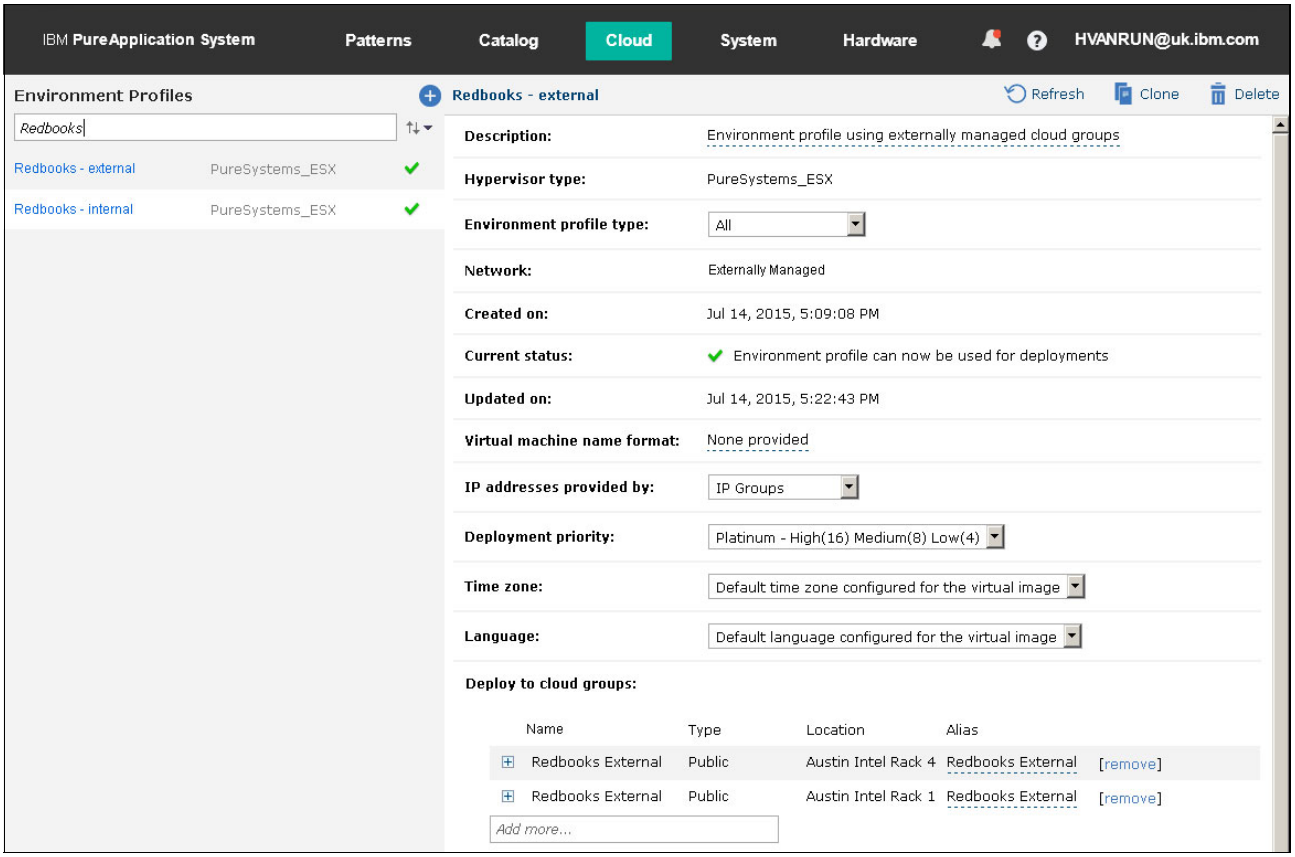


Figure 4-20 Environment profile is ready for multi -system deployments

For more information about how to create environment profiles, see the following website:

https://www.ibm.com/support/knowledgecenter/?lang=en#!/SSCR9A_2.1.0/doc/iwd/ept_creaprof.dita

4.2.9 Upgrade considerations in the multi-system environment

When you upgrade any system, its management console experiences an outage. During this time period, expect the following behaviors:

- You cannot display the shared catalog view of this system's catalog artifacts from any other system.
- Your multi-system deployments continue to run, but the display of the status of VMs is affected. VMs on the system being upgraded, when displayed on the console of another system, can be temporarily out of date or unknown.
- New deployments to externally managed environment profiles cannot connect to shared services when those shared services are originally deployed from the system that is being upgraded.

- Even after the management console becomes available, there are periods during a system upgrade when the upgrade blocks other system jobs from running. Until the upgrade is complete, you should not attempt to copy remote catalog artifacts to the upgrading system or deploy multi-system deployments to that system.

Tip: Plan to upgrade all the systems in your multi-system domain, and especially your multi-system subdomain, in short succession of each other.

If you have upgraded one or more systems in your domain, but not all systems, then you should expect the following behavior for your management domain:

- Shared catalog views of catalog artifacts (such as virtual images and script packages) should display as you expect.
- Copying or synchronizing catalog artifacts from one system to another is supported. However, if the artifact you are copying requires a level of firmware that is not present on the destination system, the copy fails.
- Some catalog artifact types support shared view or only copy on newer firmware levels. In this case, you cannot display these artifacts on systems at an older level or copy to those systems.

  For your deployment subdomain in this situation, existing multi-system deployments continue to operate. You can display their status and manage them from any system in the deployment subdomain. New multi-system deployments can also be issued to the deployment subdomain, but be aware of the following restrictions:

  - New deployment capabilities might not work until all systems in the deployment subdomain are upgraded to the latest firmware.
  - When you deploy, your pattern's required content must be present on the system where a certain pattern part is being deployed. If you upgrade and enable default data on the system with the new firmware, then that same default data is not present on the systems with older firmware. In this case, you cannot deploy from the higher-level systems to the lower-level systems. You have the following options in this case:
    - Plan your window of firmware upgrades and default data updates so that you do not need to issue multi-system deployments until all systems are updated.
    - Issue all multi-system deployments from the system with the lower firmware level. If this same default data is present on the systems with newer firmware, the deployment should succeed.
    - Before upgrading the first system, you can lock the plug-ins in a pattern. After the upgrade and after updating default data, if you deploy that pattern, it uses the older default data, which means you can issue multi-system deployments even from the system with the newer firmware.
- You cannot upgrade running deployments to newer default data until all systems on which the deployments are running are upgraded to the new firmware and default data.

Note: If you are creating a management subdomain between systems of differing release levels, you should initiate the subdomain creation from the system at the older release level

Additional considerations specific to particular versions of PureApplication System are documented as release notes. PureApplication V2.1.0.0 has specific requirements that are listed at the following website:

http://www.ibm.com/support/docview.wss?uid=swg27045032&context=SSM8NY+SSPQ5Q

4.2.10 Security

Similar to a single PureApplication System or PureApplication Software environment, the solution is designed with key security features. These features establish and manage trust across the cloud in a multi-system environment. For more information about security on PureApplication System, see the following website:

https://www.ibm.com/support/knowledgecenter/SSCR9A_2.1.0/doc/iwd/aac_overview_sec.dita?lang=en

Integration with LDAP

All PureApplication Systems and PureApplication Software instances in your management domain must be configured to use the same LDAP configuration, which can be by going to the PureApplication GUI and clicking **System** → **System Security** → **LDAP Settings**, as shown in Figure 4-21.

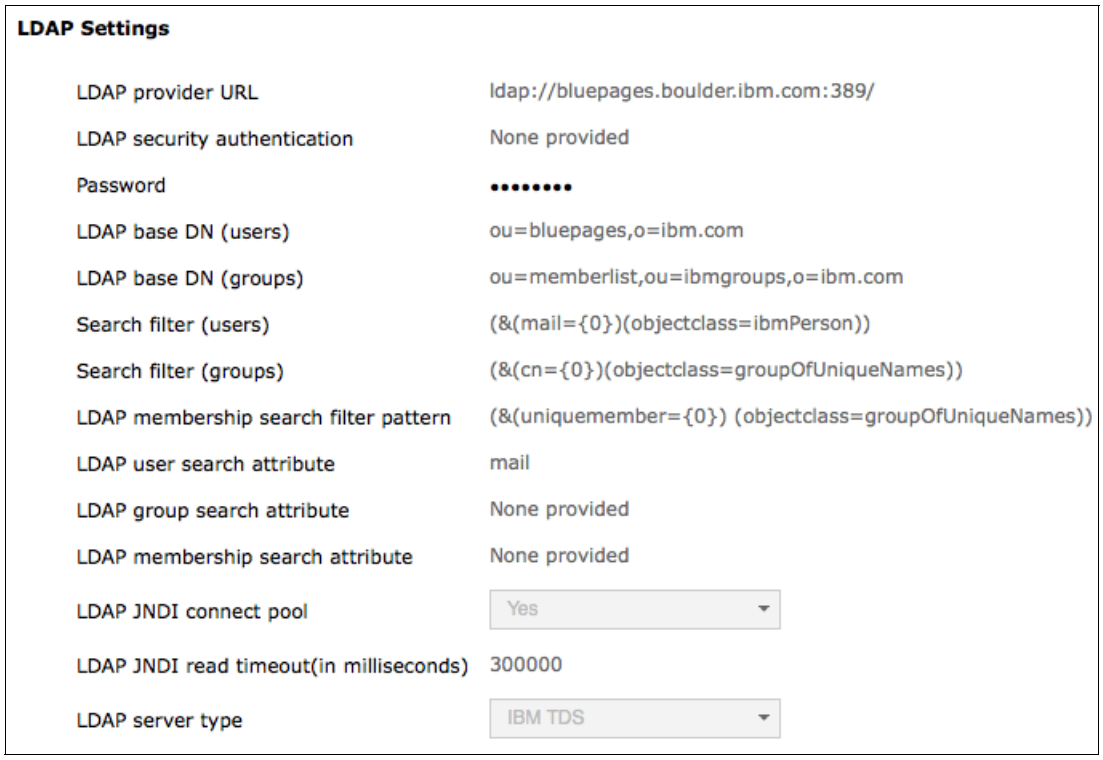


Figure 4-21 LDAP settings in the PureApplication GUI

Consider the following security restrictions when working in the multi-system environment:

- Users who are configured on Lightweight Directory Access Protocol (LDAP) servers that are integrated with the system can work in the multi-system environment. These users are known as LDAP users.
- Configure all systems in the multi-system management domain to use the same LDAP infrastructure.

- LDAP users are onboarded automatically. In the multi-system environment, when one system in the domain initiates a request that spans multiple systems in the domain, the user is created automatically, which is true if the user does not exist on the other systems. During user creation, LDAP groups that are associated with the user are also created if they do not exist. However, the membership for the newly created LDAP group is synchronized with the other systems. A manual synchronization of group membership is required.
- If an LDAP user is deleted from a system by a system administrator, the deletion can be triggered by the originating system to other systems. The deletion process is similar to the process for propagating access control lists (ACLs) and user roles. A manual synchronization is required to complete the deletion process on both systems.
- Security roles of LDAP users and LDAP user groups are not synchronized automatically among the systems. A manual administrative action is required to synchronize security roles to rectify this issue.
- ACLs are not synchronized automatically among the systems except in the case of copying a catalog artifact from one system to another, so manual administrative action is required to synchronize or reconcile ACLs between systems.

Integration with LDAP is required because most of the deployment subdomain activities required an LDAP user.

To check the type of the PureApplication user (LDAP or Internal), go to the PureApplication GUI and click **System** → **Users**, as shown in Figure 4-22.

Figure 4-22 LDAP user in the PureApplication GUI

Note: There are two sets of ACLs that are synchronized between systems: ACLs for multisystem deployments, and ACLs for externally managed environment profiles.

4.3 Using a PureApplication System multi-system environment

Whether deployed to a single system or across multiple systems, deployments to externally managed environment profiles have some dependencies. They depend on some activation and interface changes to establish the externally managed network interface and to reference multiple systems and cloud groups in a single deployed instance, which means that only newer pattern content supports deployment to an externally managed environment profile.

The following restrictions apply to multi-system deployments.

- Patterns that use one or more Hypervisor Edition virtual images cannot be deployed to externally managed cloud groups. As a result, classic (and promoted classic) Virtual System Patterns (VSPs) cannot be used with multi-system deployment. Attempting to deploy yields the following error:

  ```
  CWZKS0417E: Failed to deploy because one or more virtual images do not support
  multi-target deployment.
  ```

- Virtual appliances cannot be deployed to externally managed environment profiles.
- Patterns might not be able to deploy to externally managed environment profiles, which is true if the patterns rely on a plug-in or pattern type that depends (directly or indirectly) on a Foundation pattern type version lesser than Version 2.1.0.0. All plug-ins require Foundation Version 2.1 or greater to signal their compatibility with externally managed networks and multi-system references.

4.3.1 Managing catalog resources

This section describes how you can manage catalog resources in a multi-system environment.

From one console, you can use the PureApplication System to manage catalog resources. Here are those catalog resources:

- Images
- Script-packages
- Add-ons

To see the list of available script packages, from the PureApplication GUI, click **Catalog** → **Script Packages**.

Figure 4-23 shows the list of script packages that are available. In the upper right of the figure, you can see three buttons. These clickable buttons show you the resources in the list for the **Entire domain**, **Subdomain only**, or **Local only**.

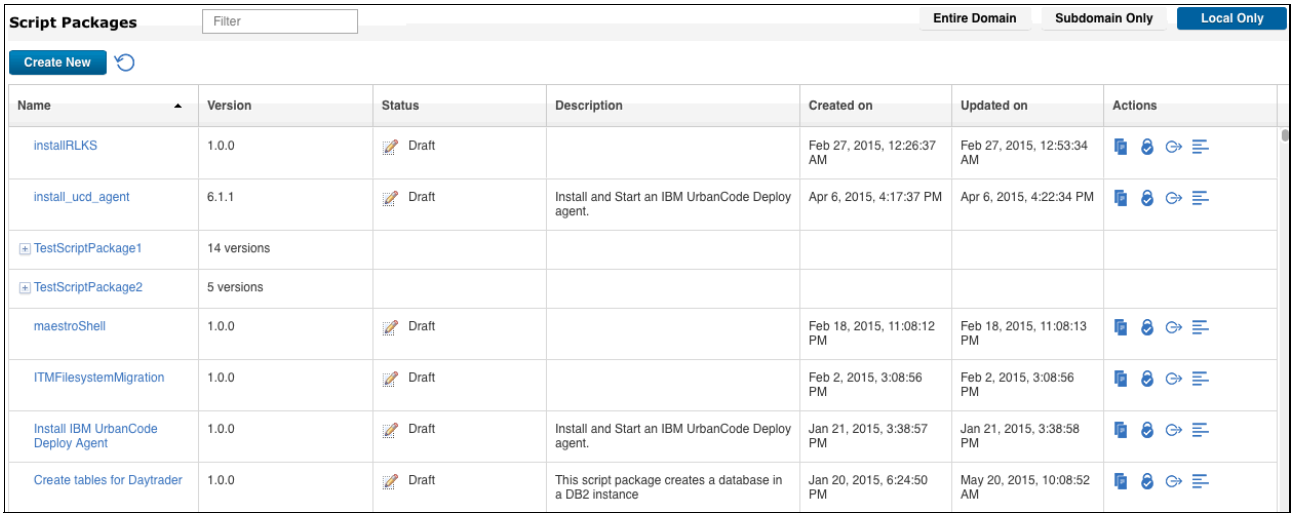


Figure 4-23 List of available script packages

To see the specific locations of a particular script package and the status of that package, click the script package, as shown in Figure 4-24 on page 95.

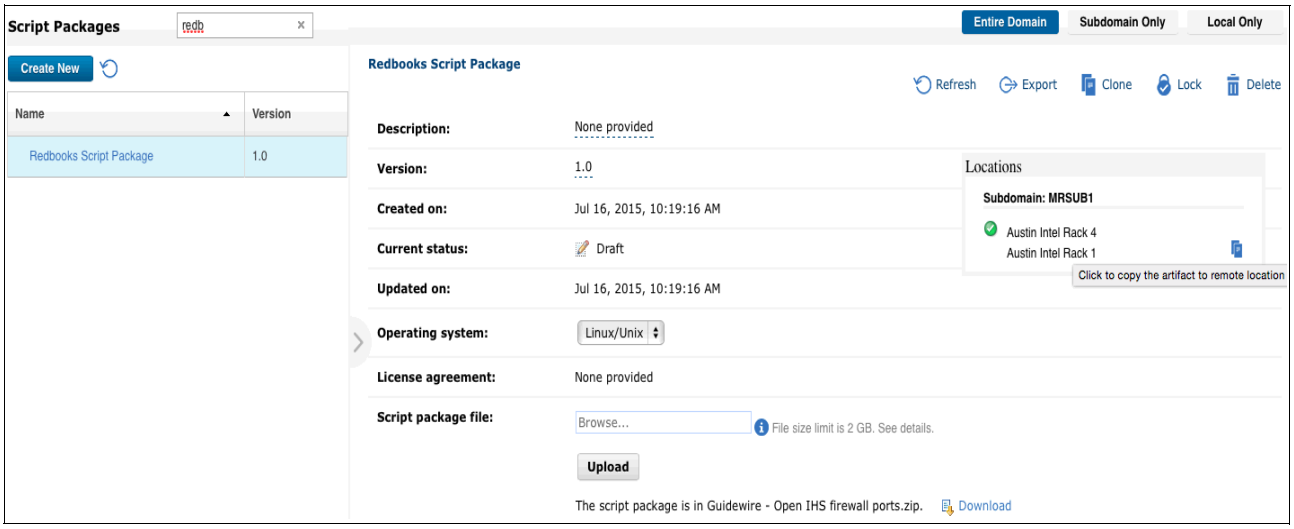


Figure 4-24 Location and status of the script package

Another capability of the multi-system environment that is supported by the PureApplication System is pushing packages to a remote location. The green check shows that the scripts package is available only on the local system, as shown in Figure 4-24. Click **Copy** to push it to a remote location.

Note: You cannot pull content from the remote location.

Figure 4-25 shows a script package that is synchronized between two locations.

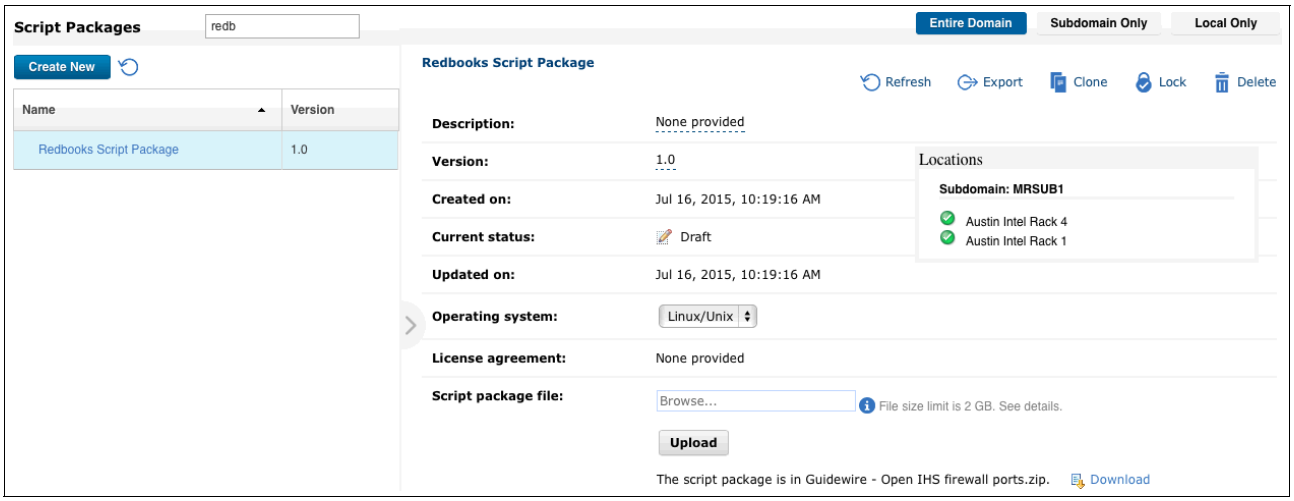


Figure 4-25 Scripts package that is synchronized between two locations

Note: All the capabilities and options that are reviewed in this section for script packages show how you can manage catalog content between two PureApplication Systems. These options are also applicable for other catalog content, such as images and add-ons.

For more information about catalog resources, see the following website:

https://www.ibm.com/support/knowledgecenter/api/content/nl/en-us/SSCR9A_2.1.0/doc/iwd/pcr_scriptpkgfui.html

4.3.2 Shared services

Because externally managed deployments can span multiple cloud groups, the scope of shared service usage should not be limited to a single cloud group. When deploying to an externally managed environment profile, the shared service association is established at the scope of the environment profile. Therefore, when using externally managed environment profiles, use a shared services instance for each of them, which can help isolate applications from one another, or even production from non-production environments. Figure 4-18 on page 88, illustrates the difference in scope for internally and externally managed environment profiles.

This consideration applies to all externally managed environment profiles. This is true regardless of whether your system is part of a deployment subdomain or whether your environment profile currently contains cloud groups from multiple systems. You should plan your externally managed cloud group and environment profile configuration carefully to avoid a multiplication of the number of shared service deployments that are needed.

Figure 4-26 shows the difference between shared services for internally and external managed environment profiles.

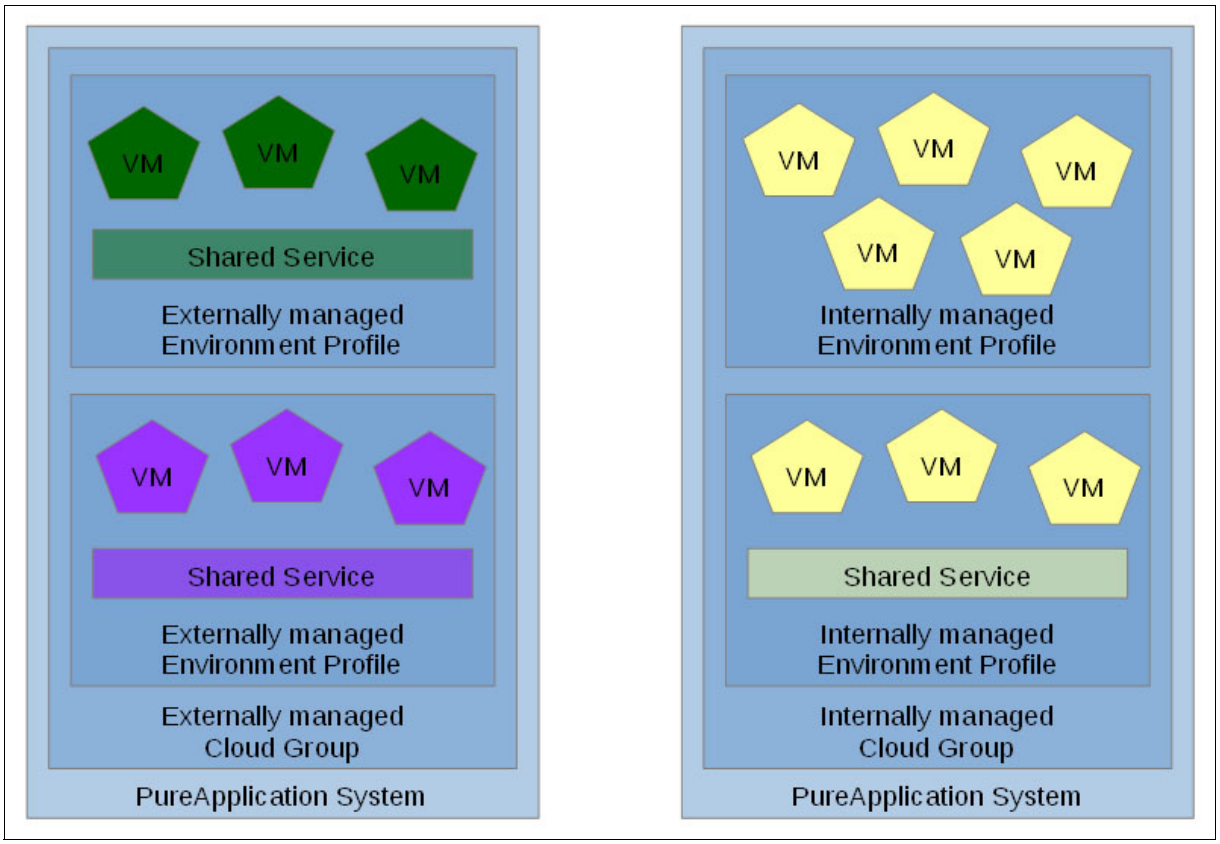


Figure 4-26 Difference in scope of shared services for internal and external managed environment profiles

For more information about shared services, see the following website:

https://www.ibm.com/support/knowledgecenter/#!/SSCR9A_2.1.0/doc/iwd/apt_sharedservices1.dita

4.3.3 Deploying a pattern by using a multi-system configuration

When deploying a pattern across multiple systems in a subdomain, the pattern itself need exist only on the deploying system. However, for each VM that is placed on any given system, that system must contain exact copies of all the pattern artifacts that are required for the following areas and more:

- The specific VM
- The virtual image
- Script packages
- Pattern types
- Plug-ins
- Software components
- Installation manager repository

The system verifies that all the required artifacts are available on both systems as a part of its placement recommendations for the VMs.

To start multi-system deployment with PureApplication System, use the PureApplication GUI and complete the following steps:

1. Click **Patterns** and choose the required pattern.
2. Select your Environment profile for multi-system deployments and click **Prepare to Deploy**, as shown in Figure 4-27.

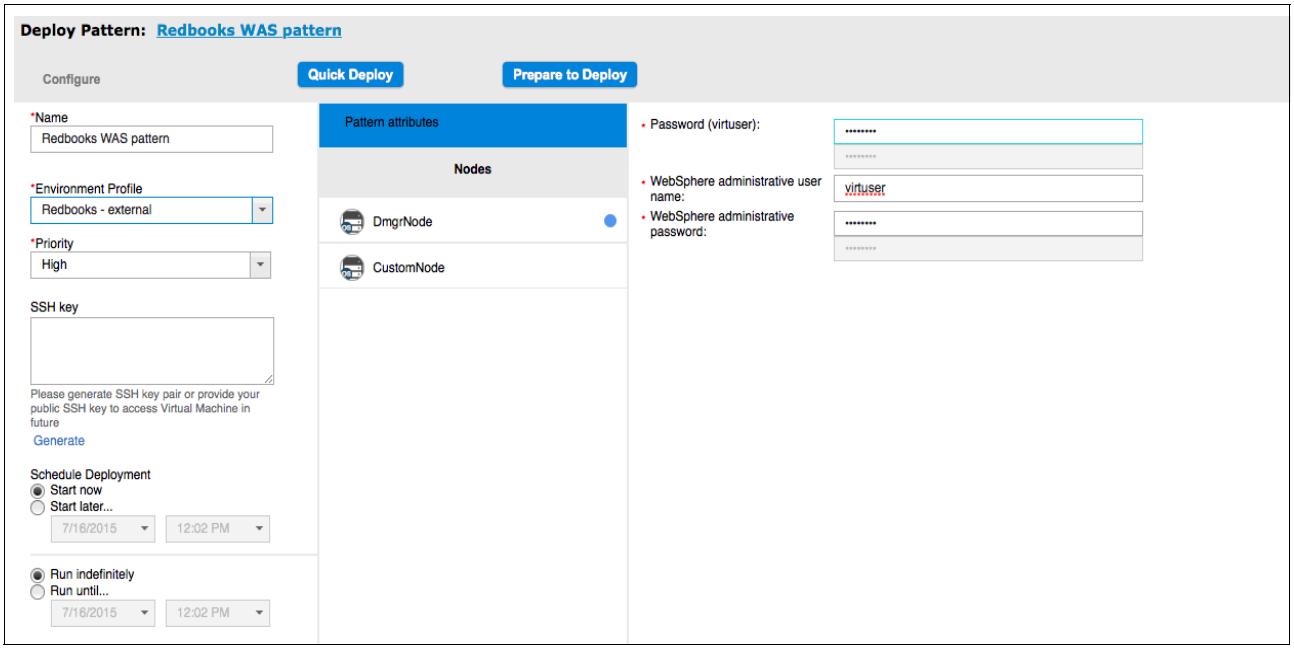


Figure 4-27 Preparation of a multi-system deployment

3. The PureApplication System then prepares deployment recommendations and topology based on the pattern placement policies and available cloud resources, as shown in Figure 4-28.

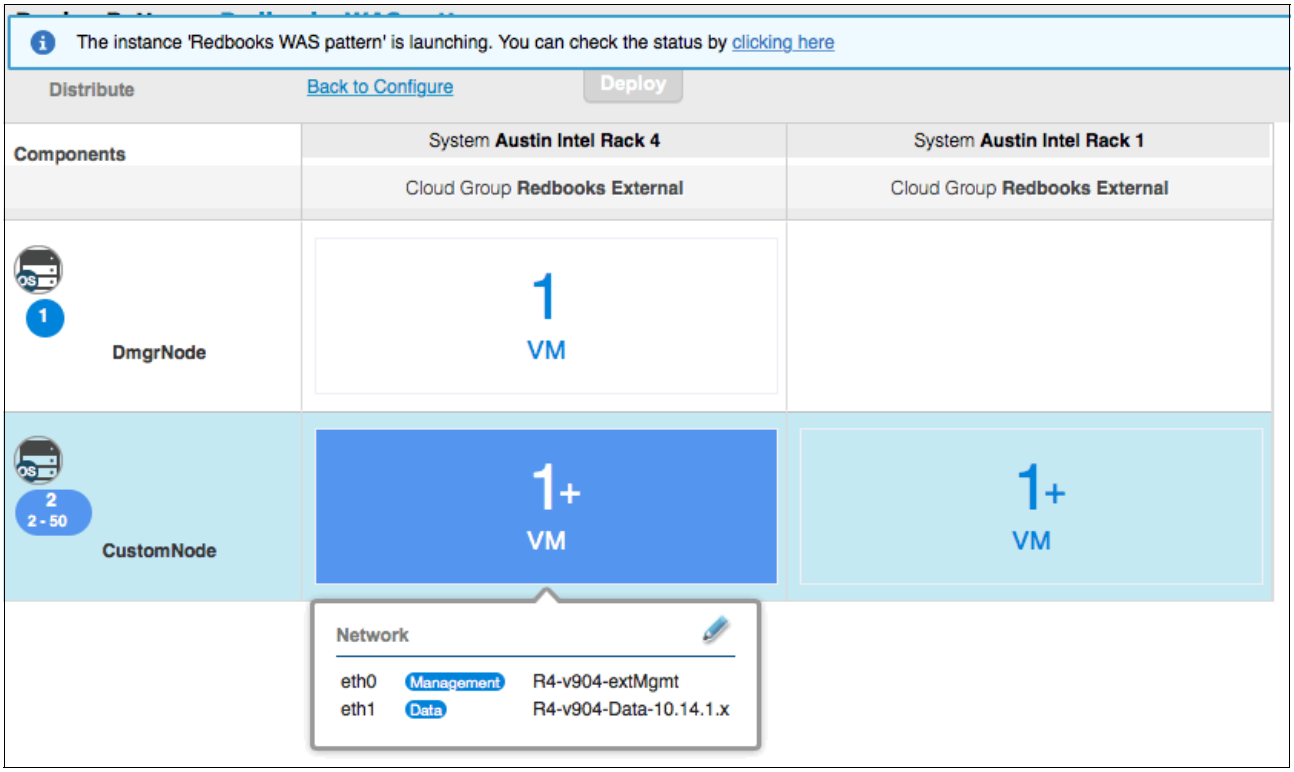


Figure 4-28 Recommended deployment topology of the pattern

4. You can enable the system to choose the placement of all VMs, or you can choose to do the placement manually. To do the placement manually, drag the VMs between the PureApplication Systems in the deployment subdomain, as shown in Figure 4-28.
5. After you click **Deploy**, the system starts deployment and provides you with a link that can be used to check the status of your deployment. By clicking the link, you see information about the deployment, including its history, which can be sorted by location, as shown in Figure 4-29 on page 99.

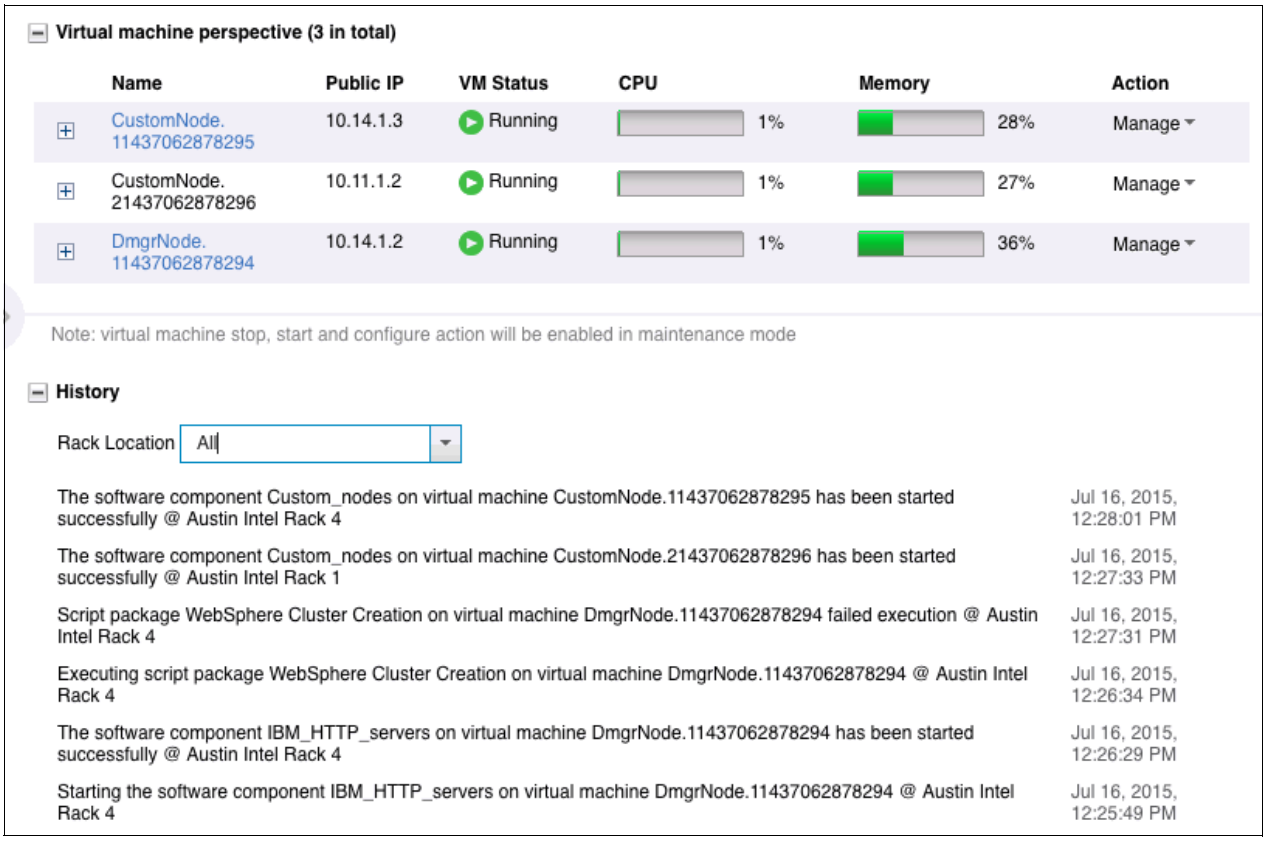


Figure 4-29 Deployment status and history

Note: The placement of VMs across the cloud groups of both systems within the deployment subdomain is set at deployment time. After being deployed, VMs cannot be moved from a cloud group on one system to a cloud group on the other system.

Placement policies

Placement policies are used to determine where parts are to be placed in the multi-target deployment. You can specify level as a location or cloud group. You can place VMs across systems or cloud groups. You can also specify *collocation* or *anti-collocation*. Collocation means that the VMs are in the same cloud group or system. Anti-collocation means that the VMs are spread across cloud groups or systems. You can choose *soft constraint* or *hard constraint*. Soft constraint means that you have a preference about policies and the PureApplication System does not follow all placement policies (for example, if there are not enough resources) and deploy the pattern anyway. Hard constraint means that you want policies enforced and the PureApplication System aborts the deployment if it is impossible to fulfill all the policies. If you do not specify a placement policy, then the default is location, anti-collocated, and soft.

Policies allow the pattern deployer to specify parameters for the pattern and its components. Placement policy options are shown in Figure 4-30.

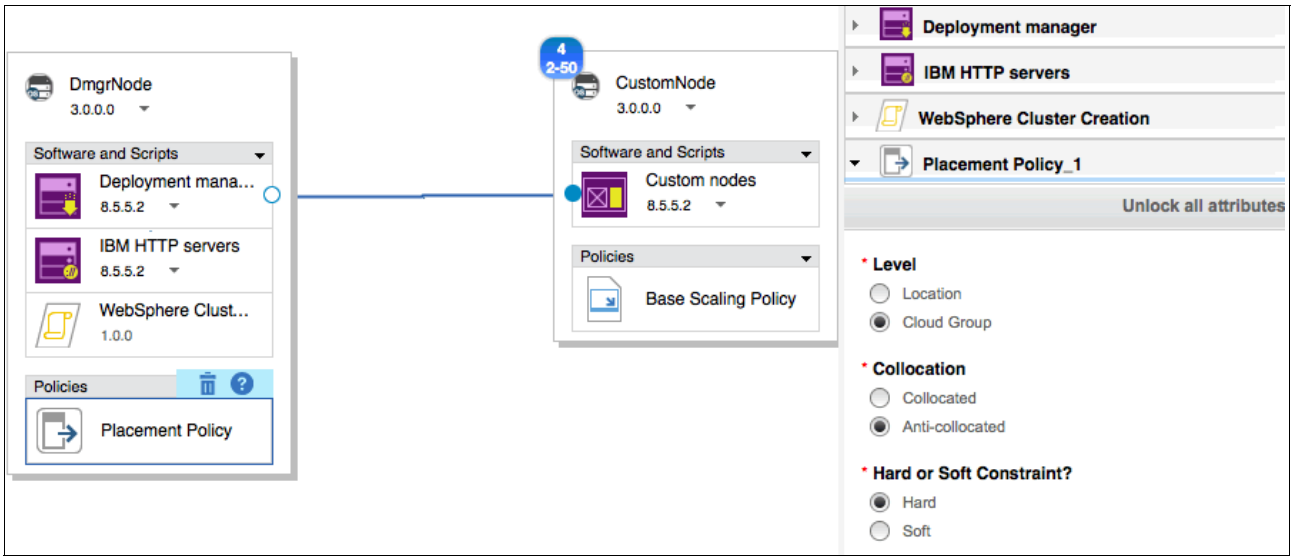


Figure 4-30 Placement policy for the pattern

4.3.4 Security considerations

To corroborate user identity between the systems, all workload-related operations that have the potential to span multiple systems require an LDAP user. To create, view, or manage externally managed profiles or to create, view, or manage deployments to these profiles, you need the following items:

- To be granted appropriate access on each system
- To be logged in as an LDAP user

Resources are hidden from non-LDAP users, regardless of those users' access levels and roles. Access controls for multi-system environment profiles and deployments are automatically synchronized between the systems in the subdomain.

4.3.5 Availability, scaling, and recovery considerations

Multi-system deployment provides a foundation for high availability of pattern instances across systems or data centers. However, this availability depends on the design of your application and any services it is using. The application and its services must be correctly designed and configured to support high availability to tolerate the loss of access to the VMs on one system.

When a VM fails, if it is considered non-persistent, the system, in many cases, attempts to recover it by cloning a new VM. For multi-system deployments, this recovery is attempted only on the same system as the original failing VM. Background information about persistent and non-persistent VMs is documented in the PureApplication IBM Knowledge Center at the following website:

`http://www.ibm.com/support/knowledgecenter/SSCR9A_2.1.0/doc/virt_app_patterns/ap/apc_trbpattern.dita`

Horizontal scaling for a multi-system deployment attempts to balance the VMs across the cloud groups and systems. However, horizontal scaling cannot deploy a new VM to a cloud group or system on which an instance of that VM was not originally deployed. Therefore, to achieve high availability, you must ensure that the placement for your initial deployment discovers a minimum of one instance of your scalable VMs on each system.

An entity that is called the *shared service provider* exists for each shared service. However, it exists only on the system from which the shared service deployment was initiated. If this deploying system is temporarily unavailable to the subdomain, new shared service clients cannot connect to the shared service, even if the shared service VMs on the remaining system still can service clients and even if the client VMs are all on the surviving system.

5

Storage

Storage should be a key discussion point when considering the integration of IBM PureApplication System or IBM PureApplication Software into a data center. Both PureApplication products support different storage configurations for various architectural reasons.

This chapter provides descriptions about basic storage concepts and how those concepts are applied to PureApplication System and PureApplication Software. The first part of the chapter describes basic concepts such as storage volumes, block storage, and file-level storage. There is an additional description about how these concepts are implemented and used in PureApplication System.

The second part of the chapter describes external storage options, in particular by using an IBM SAN Virtualization Controller to connect an existing SAN. The other external storage option that is described is how to add storage capacity to the IBM Storwize V7000 storage system.

The final part of the chapter provides a description about IBM General Parallel File System (GPFS) topologies.

This chapter covers the following topics:

- Storage principles
- SAN Virtualization Controller for external storage
- Adding additional storage to the PureApplication System
- IBM General Parallel File System

© Copyright IBM Corp. 2015. All rights reserved.

5.1 Storage principles

The following sections are provided as the background and foundation for this chapter. It is important to understand several basic storage concepts before proceeding with this chapter. The sections that follow describe basic concepts such as storage volumes, block storage, and file-level storage. There are additional descriptions about how they are implemented and used in PureApplication System.

5.1.1 What a storage volume is

When a pattern is deployed by using PureApplication System, the result is one or more virtual machines (VMs). Each VM has at least one or more volumes attached. A volume is a logical disk drive that typically represents a single partition of a drive.

With PureApplication System, all VMs start from a volume that is internal to the rack. Secondary volumes can be RAW, VMFS, Block, or Block Shared. For more information about these secondary volumes, see 5.1.5, “Block storage and file-level storage in PureApplication System” on page 107.

On a *POWER-based* PureApplication System, either RAW or block volumes are available. Both RAW and block volumes are implemented as individual Logical Unit Numbers (LUNs). Boot volumes are individual LUNs.

On an *Intel based* PureApplication System, RAW, VMFS, Block, or Block Shared volumes are available. Each VMware data store is a LUN. Individual VMFS volumes are stored as a vmdk inside a data store. Just like POWER-based systems, RAW, Block, and Block Shared volumes are implemented as individual LUNs. Boot volumes are always VMFS.

Definition: A *Logical Unit Number (LUN)* is a unique identifier to designate an individual or collection of physical or virtual storage devices that run input/output (I/O) commands with a host computer, as defined by the Small System Computer Interface (SCSI) standard

5.1.2 Base OS images

Base OS images that are shipped with PureApplication System use Logical Volume Manager (LVM) to manage volumes. Administrators can use the LVM to manage volumes and file systems more independently. This capability provides more flexibility in how volumes are grouped and managed. This capability applies to both deployment and operations throughout the lifecycle of a deployed VM. Figure 5-1 on page 105 shows the block LVM definition in the default Red Hat OS image.

```
-bash-4.2# lvm lvdisplay
  --- Logical volume ---
  LV Path                /dev/rhel/root
  LV Name                root
  VG Name                rhel
  LV UUID                AxVaoH-ijfF-sVB7-d0Lq-Ya5R-d3lj-czANdd
  LV Write Access        read/write
  LV Creation host, time vm-012-184.rainmaker.raleigh.ibm.com, 2014-11-10 22:41:
59 +0000
  LV Status              available
  # open                 1
  LV Size                9.80 GiB
  Current LE             2509
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     8192
  Block device           253:0

  --- Logical volume ---
  LV Path                /dev/rhel/swap
  LV Name                swap
  VG Name                rhel
  LV UUID                OCxt6z-hxum-XysM-XJs2-3AdG-L7Xt-ntrlCE
  LV Write Access        read/write
  LV Creation host, time vm-012-184.rainmaker.raleigh.ibm.com, 2014-11-10 22:42:
01 +0000
  LV Status              available
  # open                 2
  LV Size                1.95 GiB
  Current LE             500
  Segments               1
  Allocation             inherit
  Read ahead sectors     auto
  - currently set to     8192
  Block device           253:1
```

Figure 5-1 Block LVM definition in the default Red Hat OS image

The default layout of the IBM OS Image base OS is a single volume. On Intel, the layout is `/dev/sda`. This volume is split into two partitions: `sda1` and `sda2`. `sda1` is the `/boot` partition and `sda2` is used for LVM. `sda2` is part of the volume group `rhel`. Volume group `rhel` (see Figure 5-2 for the default volume group definition) is split into two logical volumes: `rhel-root` and `rhel-swap`.

```
-bash-4.2# lvm vgdisplay
  --- Volume group ---
  VG Name               rhel
  System ID
  Format                lvm2
  Metadata Areas        1
  Metadata Sequence No  3
  VG Access             read/write
  VG Status             resizable
  MAX LV                0
  Cur LV                2
  Open LV               2
  Max PV                0
  Cur PV                1
  Act PV                1
  VG Size               11.75 GiB
  PE Size               4.00 MiB
  Total PE              3009
  Alloc PE / Size       3009 / 11.75 GiB
  Free  PE / Size       0 / 0
  VG UUID               Kg2Xhs-XGtv-NVNe-wEds-1S17-8fwz-9oq2Yl
```

Figure 5-2 Default volume group information

Figure 5-3 shows details of the block device layout for the default Linux base OS.

```
-bash-4.2# lsblk
NAME          MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
sda             8:0    0   12G  0 disk
├─sda1          8:1    0  250M  0 part /boot
└─sda2          8:2    0 11.8G  0 part
  ├─rhel-root 253:0    0  9.8G  0 lvm  /
  └─rhel-swap 253:1    0    2G  0 lvm  [SWAP]
sr0            11:0    1  370K  0 rom
-bash-4.2#
```

Figure 5-3 Block device layout for the default Linux base OS

5.1.3 What block storage and file-level storage are

There are two main types of storage technology in use today: block storage and file-level storage. Nearly every time a file is accessed, data is accessed at the file level and at the block level. The amount of abstraction between the OS and the storage media where files are stored varies widely. There can be multiple physical devices and different protocols that are involved when getting files from storage.

In essence, block storage is the physical storage and file-level storage is how the physical storage is organized and accessed. The next two sections take a closer look at block storage and file-level storage.

Block storage

Block storage is raw volumes that are exposed to the OS. They must be partitioned and formatted before they can be used for file storage. Block storage is exposed to an operating system either as a direct connection or through a storage protocol. Examples of block storage protocols are Fibre Channel, iSCSI, and FCoE.

In a SAN solution, block devices are exposed directly to the operating system. The SAN administrator creates a LUN and makes it available to one or a group of hosts through the storage fabric.

File-level storage

File-level storage is familiar to anyone who uses a computer. The folders, paths, and permissions of the file system are all concepts that are part of file-level storage.

File-level storage is exposed in the OS where the file system is. Network-attached storage solutions such as NAS, NFS, and Samba all present file-level storage to the user. In an NAS solution, the block device is abstracted from the user. The RAID level, partitioning, and formatting are all handled by the host device. The user is not aware of this setup.

Definitions:

Network-attached storage (NAS) is a type of dedicated file storage device that provides local area network (LAN) nodes with file-based shared storage through a standard Ethernet connection.

Network File System (NFS) is a distributed file system protocol that allows a user on a client computer to access files over a network much like local storage is accessed.

Samba is an open source, no-charge software suite that provides file and print services to SMB/CIFS clients. Samba is freely available and allows for interoperability between Linux and UNIX servers and Windows based clients.

5.1.4 How block storage and file-level storage are used

This section covers block and file storage, and the difference between the two types.

Your notebook

A typical notebook has a single hard disk drive (HDD) that is connected to the rest of the computer through the PCIe bus. When the HDD is new, it first must be partitioned. Partitioning creates the C: and perhaps D: and E: drives.

Although the HDD has the C:, D:, and E: partitions, this does not mean that they are usable for file storage. They must be formatted with a file system such as FAT32 or NTFS. After they are formatted, the partitions are ready to be used as C:\ D:\ and E:\ drives.

To summarize, the HDD is the block device. The file system contains the file devices. The bus in between them implements the storage protocol that connects the block device to the OS and thus makes the file systems available.

Definitions:

FAT32 is the most current version of the File Allocation Table that the operating system uses to find files on a disk. Due to fragmentation, a file may be divided into many sections that are scattered around the disk. The FAT tracks all these pieces.

New Technology File System (NTFS) is a high-performance and self-healing file system that is proprietary to the Windows platform. It supports file-level security, compression, and auditing. It also supports large volumes and storage solutions, such as RAID. Its most important features are data integrity (transaction journal) and the ability to encrypt files and folders.

Linux system

Consider the same notebook, but with Linux installed on it. It still has that same HDD that is the block device, in this case, `/dev/sda`. If you create partitions `/dev/sda1` and `/dev/sda2`, then you can format them by using a file system such as EXT3. After the file systems are created, they are ready to be mounted as part of the overall file system for your Linux box.

5.1.5 Block storage and file-level storage in PureApplication System

Storage volumes in a large system like PureApplication System work much the same way as described for your notebook or a Linux system. PureApplication System might introduce new terms such as SAN, Fibre Channel, Block Storage, RAW, and LUN, but in reality the concepts are not new at all. The concepts might be unfamiliar if you are new to PureApplication System, but they are not new. They are analogous to the notebook sitting on your desk.

PureApplication System has four types of storage devices that can be used and attached to your VM: VMFS, RAW, Block, and Block Shared. This section looks at each type in more detail.

VMFS

VMFS is VMware's proprietary file system. VMware storage is defined as data stores. The file system within each data store is VMFS. On Intel based PureApplication Systems, the boot volume for each deployed VM is in a VMFS data store. Multiple ESXi hosts can share each data store, which are isolated on cloud group boundaries so that each cloud group has its own set of data stores.

PureApplication System handles storage placement for new deployments. If there is no suitable data store for a new deployment, a data store is created. The default size for new data stores is a 1.8 TB LUN on a Storwize V7000 storage system. Placement continues to use the available data stores until they are above a capacity threshold. After that threshold is reached, another new 1.8 TB LUN is created and used as a new data store.

It is possible to have circumstances where a particular data store becomes too full and certain actions start to fail because of insufficient capacity. The capacity algorithm tries to balance future growth with current density in the storage arrays.

RAW

RAW volumes are simply LUNs on a Storwize V7000 storage system that are created and available to a cloud group. RAW volumes cannot be moved between cloud groups, and they lack other advanced features that are available with Block Storage volumes.

Block

Block volumes are also RAW LUNs on a Storwize V7000 storage system. They can also be LUNs that are defined on some other attached storage device and made available through a SAN Volume Controller.

External block volume can be moved between cloud groups. For example, block volumes can be attached to a VM in cloud group A. After it is unmounted and detached, it can be moved to cloud group B.

Block volumes also can be imported, exported, cloned through IBM FlashCopy®, and grouped into consistency groups. PureApplication System allows block volumes to be replicated between two PureApplication Systems, which can be done synchronously over metropolitan distances or asynchronously otherwise. These features are often used as key components in high availability (HA) and disaster recovery (DR) scenarios.

Definition: *FlashCopy* is an IBM feature that is supported on various IBM storage devices that makes it possible to create near-instantaneous point-in-time snapshot copies of entire logical volumes or data sets.

Block Shared

Block Shared devices have all the same qualities as a block device, but have the unique ability to be attached to more than one VM at a time if the VMs are on different compute nodes. Block Shared devices are used by the GPFS pattern for synchronous access and data consistency.

Add-ons

Add-ons are specialized scripts that customize your VM configuration. Add-ons provide fine-tuning for hardware and operating system configurations.

At deployment time, all add-on operations are run before the custom script packages are run. The order in which add-ons are run at deployment cannot be specified in a Virtual System Pattern (VSP), unlike scripts and parts. Add-ons always run as the system is created. They are not initiated by users, and they do not run at deletion. Depending on the type of add-on, special processing is performed during deployment to add the additional hardware to the VM as needed.

Add Disk

Add Disk does exactly as its name implies: It adds a disk. The result is similar to a virtual volume. Depending on the platform and the add-on that is used, the disk might be a new VMFS volume in a data store or it might be an additional LUN.

A word of caution: The middleware team should work with the UNIX SMEs. Using an Add Disk add-on *is not* the same as adding a directory or a mount point. From a resource impact perspective, there is a limit to the number of volumes and LUNs that can be attached to a single VM. The SCSI bus in the VM can handle only 14 devices. If you want to separate file systems, you can accomplish that goal with partitions in a single volume.

Block Volume

Block volumes can be specified and automatically attached to a VM at deployment time. Because block volumes have a decoupled lifecycle from the deployed instance to which they are attached, this can be a powerful concept. It is possible to have a block volume that is created and attached to a temporary system. While it is attached to the temporary system, the block volume can be partitioned and formatted as needed. The block volume can be pre-populated according to the needs of future deployments. After the volume is set up as wanted, it can be detached from the temporary system and reused with other new deployments. If needed, you can make multiple FlashCopy copies of the block volume and use it for many purposes, such as testing or software installation. Another popular scenario for block volumes is backup and restore, which is described in Chapter 10, “Backup and restore” on page 275.

Block volumes can also be imported and exported. In this way, you can have pre-built file systems that are available to be attached to any system with which you want to use them.

Groups of block volumes are added to block volume groups so that they can be managed together. Block volume groups are implemented as consistency groups within the Storwize V7000 storage system.

Viewing Volumes and Volume Groups

You can view the table of the volumes that are known to the rack or to the software instance. You can see the device type, location, and VMs by using the volume, and the volume group to which the volume belongs.

To view the volumes on a PureApplication System, complete the following steps:

1. From the console, click **Cloud** → **Storage** → **Volumes**. A list of volumes is displayed, as shown in Figure 5-4.

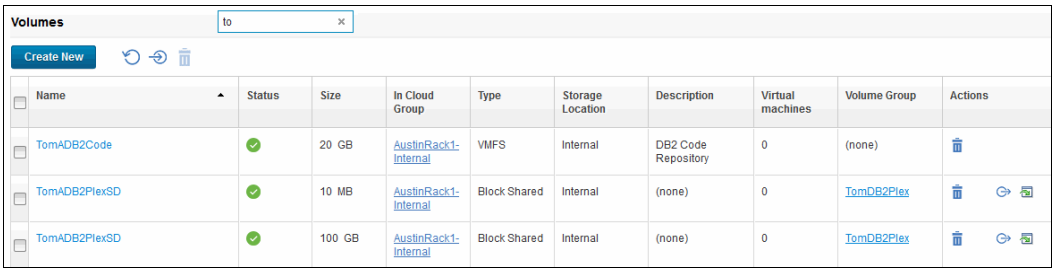


Figure 5-4 List of volumes

2. Click one of the volumes to view its details, as shown in Figure 5-5. From here, you can see the volume type (VMFS, Block, or Block Shared). You can also see to which volume group the volume belongs.

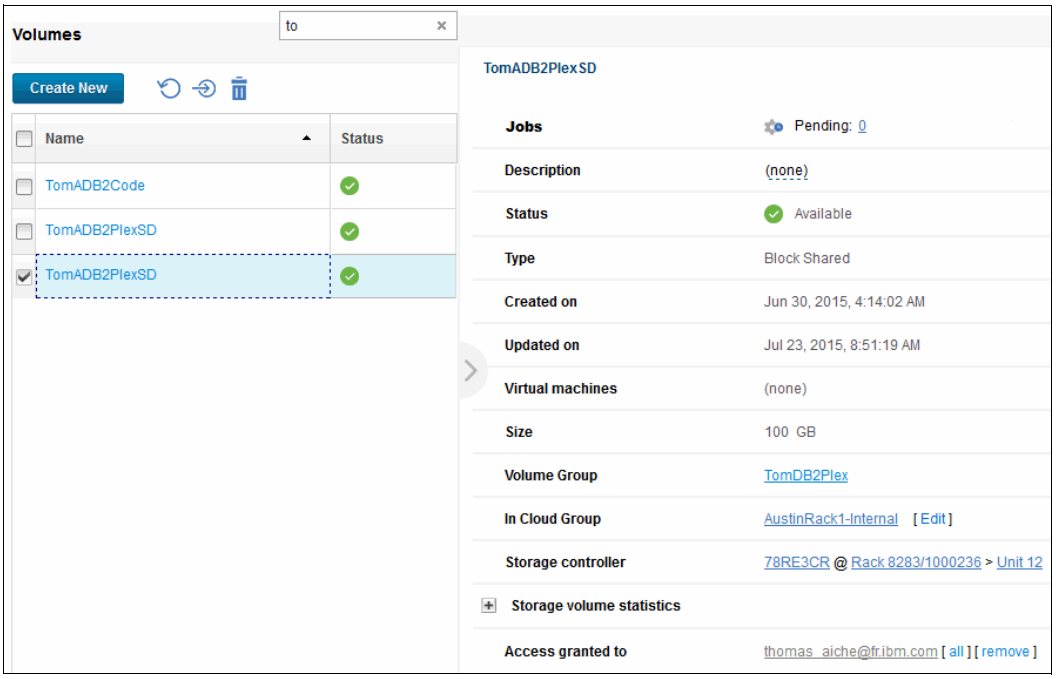


Figure 5-5 Volume details

To view the volume groups on a PureApplication System, complete the following steps:

1. From the console, click **Cloud** → **Storage** → **Volume Groups**. A list of volume groups is displayed, as shown in Figure 5-6.

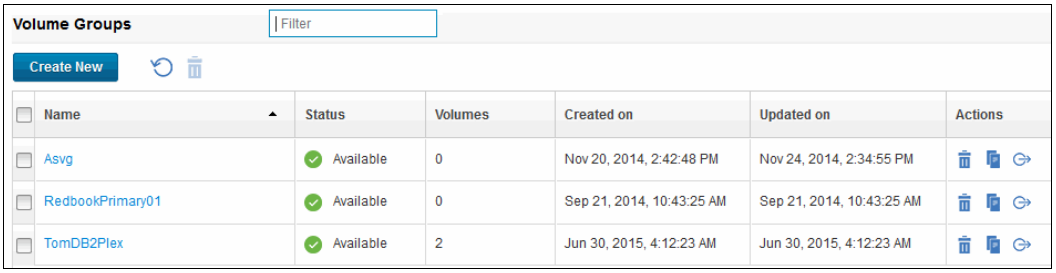


Figure 5-6 List of volume groups

2. Click one of the volume groups to view its details, as shown in Figure 5-7 on page 111.

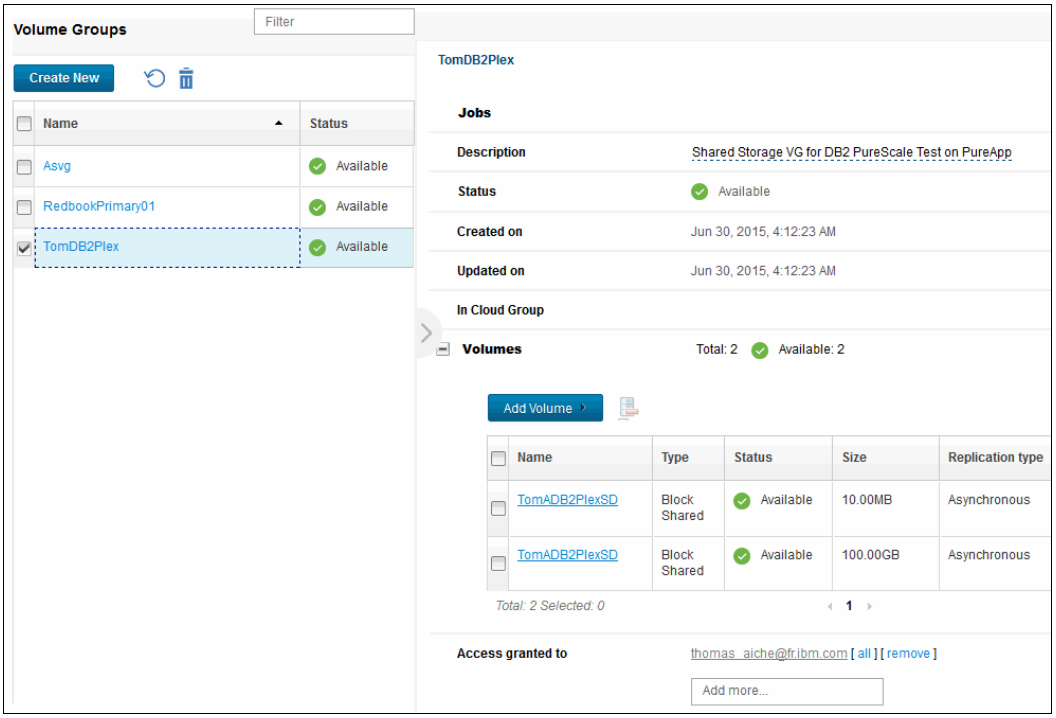


Figure 5-7 Volume group details

5.1.6 Block storage and file-level storage in PureApplication Software

Section 5.1.5, “Block storage and file-level storage in PureApplication System” on page 107 described using block storage and file-level storage principles as they apply to PureApplication System. Because the infrastructure of PureApplication Software is fundamentally different than PureApplication System, block storage and file-level storage are handled differently.

The SAN administrator is responsible for creating LUNs and making them available to the deployment host or host group. After they are available, PureApplication Software can initiate a discovery process. The discovery step is performed by VMware, so it is easy to watch how the process flows in vCenter.

If you create VMFS data stores and make them available to PureApplication Software, then PureApplication can create VMFS volumes in the data store.

When a new VM is deployed, the boot volume for the VM is a `.vmdk` file that is stored in the data store. Other volumes that are created for the VM can be stored in the data store and managed by PureApplication Software. You can also create LUNs in the SAN and use them the same way that you use any other block volume.

An additional option that is available with PureApplication Software is the ability to use iSCSI to create block volumes. iSCSI can be used for block devices and for creating the data stores that you attach to your VMware hosts.

Definition: *iSCSI* is a lightweight block storage protocol that uses the data network through TCP/IP.

5.2 SAN Virtualization Controller for external storage

If you want to use an existing SAN by connecting it to PureApplication VMs, you can implement a SAN Virtualization Controller. The addition of a SAN Virtualization Controller allows you to access potentially the VMs at high IOPS rates. The SAN Virtualization Controller allows you to expand storage and potentially achieve higher data throughput than with the standard Storwize V7000 storage system in the rack.

5.2.1 When to use a SAN Virtualization Controller

There are two general use cases when you want to use a SAN Virtualization Controller:

1. DBaaS workloads that demand high IOPS.
2. Block devices that are shared between racks or even data centers can be facilitated through SAN Virtualization Controller.

Pros

The addition of a SAN Virtualization Controller allows you to access greater amounts of storage than what is available in the PureApplication System:

- Nearly unlimited storage for workloads
- Increased compute density
- Ultimate flexibility in options for both HA and DR scenarios
- Fast throughput and access to data for applications that demand high IOPS

Cons

Infrastructure complexity and increased cost are the drawbacks to implementing a SAN Virtualization Controller. Although the SAN Virtualization Controller appliance is relatively inexpensive, it is licensed by each TB of data that is virtualized.

- Adding a SAN Virtualization Controller and connecting it to an EMC or NetApp appliance adds to the complexity of the solution.
- Additional hardware and firmware must be tested to ensure compatibility.
- Additional vendors must be contacted for technical support.

5.3 Adding additional storage to the PureApplication System

Another option for existing clients is the ability to add native storage capacity to the Storwize V7000 storage system in the PureApplication System racks through an additional feature code and is performed in the data center. The task adds one or more additional expansion units to the Storwize V7000 storage system.

Depending on your system generation, there are different requirements:

- Gen 1 systems: There is no space for an additional enclosure. You can add storage enclosures, but they must be physically mounted in another frame. The additional frame must have PDUs for the storage array and be physically close to the PureApplication System frame. Unlike fiber cables, Direct Attach Cable (DAC) cables are not long, so proximity is vital.
- Gen 2 systems: The expansion units are powered by the PDUs in the rack and then connected to the Storwize V7000 array through a daisy-chained DAC connection.

Definition: *Daisy chain* is a wiring scheme in which multiple devices are wired together in sequence or in a ring.

- Gen 3 systems: If there is space in the frame, you can take advantage of the built-in PDUs.

5.4 IBM General Parallel File System

General Parallel File System (GPFS) is a high-performance, shared-disk file management system that provides reliable access to multiple nodes in a cluster environment.

GPFS allows the same file to be accessed from multiple different clients. GPFS is built to be redundant so that the file system remains active even if the host nodes become unavailable or inoperable.

PureApplication System provides the IBM Pattern for GPFS, which the system administrator can deploy as a GPFS cluster. Figure 5-8 shows an example of a GPFS cluster. The GPFS cluster provides high-performance, highly available storage to the middleware and applications that are deployed in the PureApplication System environment.

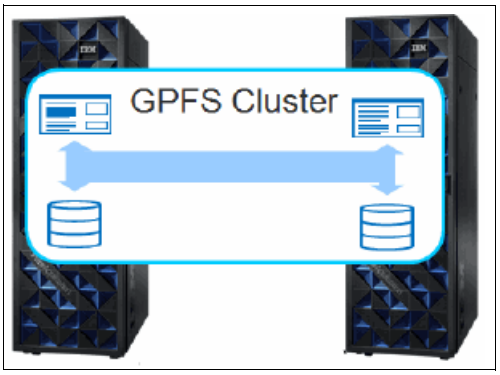


Figure 5-8 GPFS cluster example

The system administrator is responsible for the following actions:

- Deploying the GPFS Server pattern
- Attaching storage
- Defining file systems
- Deploying the GPFS shared service

5.4.1 GPFS topologies

Using the IBM Pattern for GPFS, the system administrator can deploy a number of different file server topologies. All deployments provide a GPFS Manager VM that directs and controls the underlying GPFS cluster. From the GPFS Manager VM and its application management capabilities, administrators can define file systems, add storage, add file system mirrors, manage security credentials, apply recovery actions in the case of a failure, and attach a passive replica.

The GPFS Server is the first component that is deployed when you use a GPFS shared file system topology. This component is deployed as a virtual application by using the GPFS Pattern Type. When you deploy a GPFS Server component, configure the GPFS Server nodes and attach the list of disks that are used by the GPFS shared file system. This virtual application pattern supports the following configurations:

- Primary
- Mirror
- Tiebreaker
- Passive

There are two GPFS topologies that are worth describing: Active/Active and Active/Passive.

Active/Active

The *Active/Active GPFS deployment* is also referred to as active-active mirroring and is considered a high availability scenario. This topology includes a *GPFS Primary server* along with attached *GPFS Mirror* and *GPFS Tiebreaker* servers. The tiebreaker server takes care of quorum and decides which server is primary. The tiebreaker also takes over the primary role of serving clients if either the primary or mirror server becomes unstable. The number of GPFS mirror nodes must match the number of primary nodes.

The size of the mirror and primary storage volumes must match as well. It is interesting to note that the block storage volumes are separate and not aware of each other. GPFS is responsible for the synchronization of data in the two volumes.

Active/Active mirroring is used for disaster recovery scenarios over shorter distances (less than 300 km). Figure 5-9 shows the three GPFS server configurations that are deployed to build this high availability scenario.

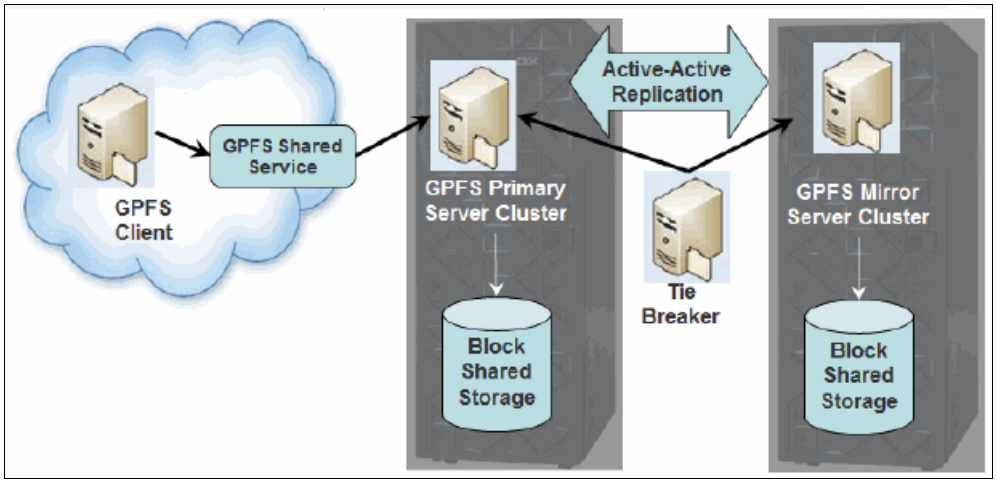


Figure 5-9 Active/Active GPFS deployment

Active/Passive

An *Active/Passive GPFS deployment* includes a *GPFS Primary server* configuration and a *GPFS Passive server* configuration. For replication, this configuration uses block storage replication, which must be set up manually. This configuration, which is shown in Figure 5-10 on page 115, is used over larger distances (less than 8000 km).

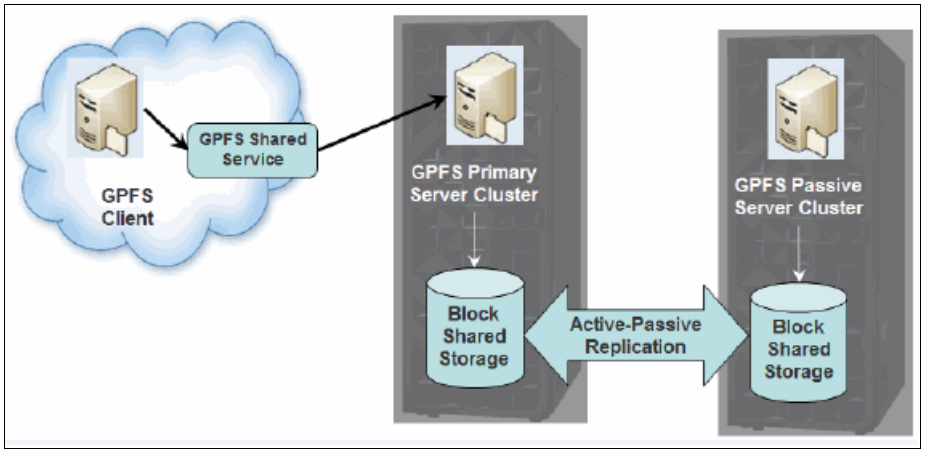


Figure 5-10 Active/Passive GPFS deployment

In the approach that is shown in Figure 5-10, the two GPFS server configurations are not aware of each other. However, the volumes are aware of each other (synchronized) through a block storage replication approach.

5.4.2 Shared service for GPFS

This shared service is deployed so that GPFS clients in the same cloud group can communicate with a deployed GPFS Server. To communicate with a GPFS Server, GPFS clients must know the client key, which contains the IP address and key that is retrieved from the Primary GPFS Manager server. The shared service for GPFS simplifies the GPFS client deployments by providing information about available GPFS Servers. As a prerequisite for this shared service, you must deploy a GPFS Server configuration by using IBM Pattern for GPFS.

5.4.3 GPFS file systems and file sets

GPFS Servers can host multiple file systems. File systems are attached to one or more block storage volumes. On the GPFS Server, file systems are mounted by running the following commands:

- **`/gpfs/fileSystemName1`**
- **`/gpfs/fileSystemName2`**

Within each file system, you define file sets that are treated as subdirectories within the file system. File sets are created in the client VM, either through a GPFS client policy or post deployment. Even though file set creation operation is run on the client side, the file set directory is created on the server VMs. A directory is created for each file system and file set in the client VM by running the following commands:

- **`/gpfs/fileSystemName1/fileSet1`**
- **`/gpfs/fileSystemName1/fileSet2`**
- **`/gpfs/fileSystemName2/fileSet1`**
- **`/gpfs/fileSystemName2/fileSet2`**

These shared file directories are linked to local file directories. This convention makes it easier for workloads to reference those directories. For example, a shared file directory might look like the following example:

```
/WASTranlogs links to /gpfs/fileSystemName1/fileSet1
```

A maximum of 14 storage volumes can be attached to all file systems for a GPFS Server. File systems must be created before a client can use them.

6

Monitoring

IBM PureApplication System provides integrated, layered monitoring of all areas of your integrated solution. You can use system, middleware, and hardware monitoring facilities to gain quick access to the status, performance, and resource usage of all areas of your integrated solution.

This chapter covers the following topics:

- PureApplication System Monitoring overview
- System event monitoring by using SNMP
- System Monitoring shared services
- Using Optim for performance monitoring of DB2

© Copyright IBM Corp. 2015. All rights reserved.

6.1 PureApplication System Monitoring overview

PureApplication System provides integrated, layered monitoring of all areas of your integrated solution. The PureApplication System Monitoring operates at three distinct levels:

- Hardware

  Monitoring at the hardware level includes compute nodes, management nodes, storage system, network switch, and light-emitting diode (LED) status. The system provides an infrastructure map that contains all the important monitoring information in one place. Monitoring of the hardware is critical to the health of your system. If a hardware component is not functioning correctly, then it is possible that workloads might not function correctly.

- System

  PureApplication System provides a comprehensive System Monitoring service that provides the monitoring infrastructure to collect performance and availability information. You can use this shared service to monitor the operating system (AIX, Linux, or Windows) and database and workload deployments. Information from the shared service is available through a portal. Use the System Monitoring shared service to troubleshoot your system and make business decisions about your hardware and service usage.

- Middleware

  Middleware monitoring provides you the most specific view of individual middleware workloads and a detailed view of how each workload fits into the bigger picture of your system. This level of monitoring allows a fine-grained view into your workloads and instances. This level of monitoring requires that you deploy monitoring shared services. Agents that are installed on the middleware are used to monitor and collect information

Figure 6-1 shows the three monitoring levels.

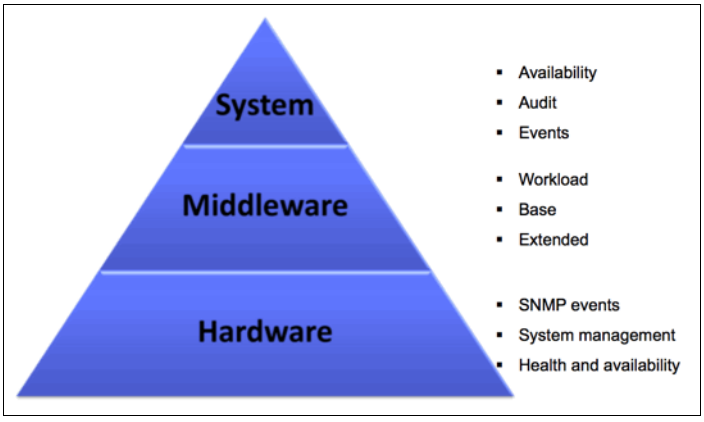


Figure 6-1 PureApplication System Monitoring layers

Middleware and system level monitoring are enabled by forwarding events from the System Monitoring shared service instance. These events are sent to either PureApplication Events (by using an internal PureApplication receiver) or forwarded to an external monitoring solution by using Event Integration Facility (EIF) events.

6.2 System event monitoring by using SNMP

Almost all events that are related to hardware, whether they are the PureApplication System rack, top of rack network switches, chassis, compute nodes, or storage, are generated and can be viewed within PureApplication itself by using the PureApplication System console and clicking **System** → **Events**. Figure 6-2 shows the Events window.

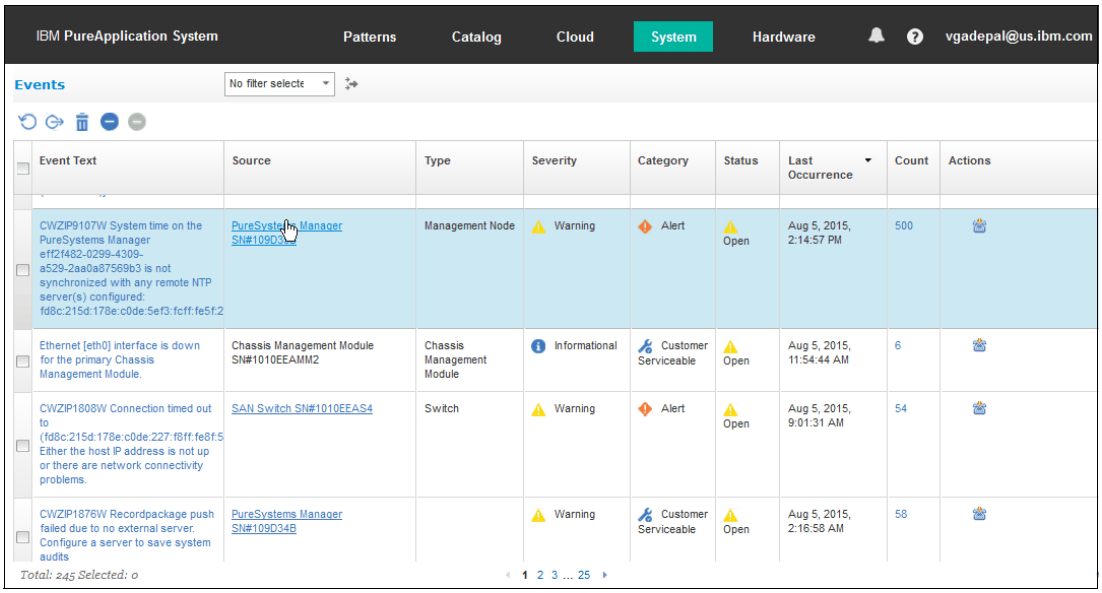


Figure 6-2 Events UI window

You can set filters, such as event type, severity, category, and time interval, to refine the list of system events that are displayed in this view. When looking at this window in the PureApplication System console, the two main columns that you should be paying attention to or, along with the problem text, is “Severity” and “Category”. When looking at this PureApplication System console window, the columns to pay attention to are the Event Text, Severity, and Category.

The Severity column has one of the following values:

- Critical

  Critical events are those events that require assistance from IBM Software Support. However, in the case of some critical events, the system can continue to operate successfully because of the redundancy of the hardware, the networking, and the overall architecture. If a critical event occurs repeatedly, indicated by the occurrence number in the Count column, contact IBM Software Support and gather the appropriate logs.

- Warning

  Warning events provide useful information within the context of the system and are often used by IBM Software Support to diagnose problems that are identified by failures. These events can reflect situations that might eventually lead to a critical event if the automated recovery steps fail. This type of event represents state changes within the system that, although not causing a critical issue, might be an indication of a problem. The system administrator can track this event and use the information that is provided to take corrective action if necessary.

- Informational

  This type of event represents state changes within the system, but is not necessarily a cause for concern or action. For example, when a virtual machine (VM) is migrated from one compute node to another because of maintenance or other activities, an informational event is emitted. The system administrator can track this event, if necessary.

- Unknown

  These are events that are emitted by a device within the system that has no severity that is assigned to it. These are harmless usually.

The Category column indicates the level of response to the severity and the event. Here are the options:

- Alert

  Events of this category can have an associated warning severity.

- Resolution

  Events of this category can have an associated informational severity, which indicates a state change occurred within the system, but it is not necessarily a cause for concern or action. Resolution events might indicate that an earlier alert, which indicated some problem or potential problem, was corrected. In such cases, the resolution event is correlated to the earlier alert events and both the original alerts and the resolution are automatically closed.

- Call Home

  These types of events typically indicate a failure that requires urgent attention. A record is created on the Problems window for each Call Home event. Call Home events that are not critical represent a piece of hardware that detected a failure but continues to work for a period. Even though attention to these events is less urgent, they can result in a critical failure and the hardware might become inoperable if ignored. As a preferred practice, contact IBM Software Support for any Call Home event. For more information about the Call Home feature and its configuration, see Chapter 12, “Service and Support Manager” on page 333.

- Customer serviceable

  Events of this category can include required physical checks. If the physical checks do not uncover an environmental problem that you can resolve, contact IBM Software Support for further assistance while collecting the correct hardware logs and resolving the problem.

These events can also be forwarded to an external event receiver, such as Netcool/OMNIbus or HP Openview. These events are sent as SNMP events. In order for a third-party monitoring tool to consume and display PureApplication events, they must import the PureApplication Management Information Base (MIB), which is made available through the PureApplication Console. The MIB can be downloaded by clicking **System** → **System Settings** → **Events**, as shown in Figure 6-3 on page 121.

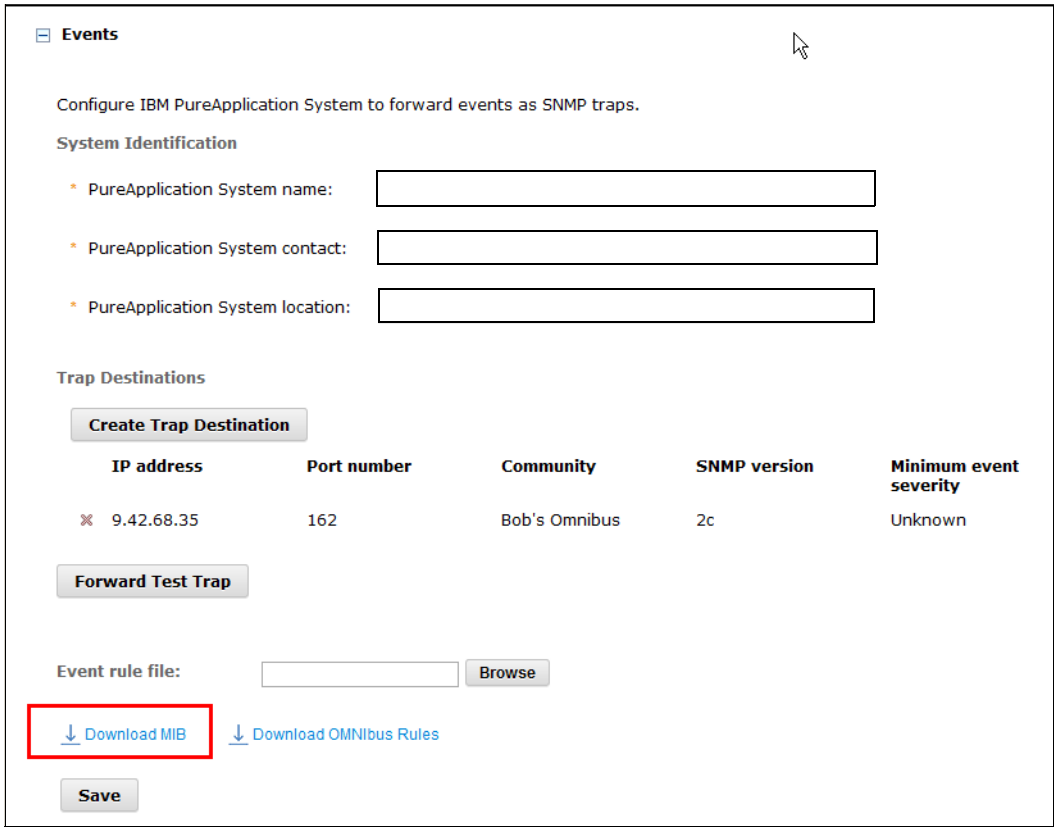


Figure 6-3 Download the PureApplication MIB

After the MIB is downloaded and imported into an external monitoring receiver, a trap destination must be created to forward the events to the monitoring system.

6.3 System Monitoring shared services

The System Monitoring shared services can provide monitoring of the operating system (OS) and some IBM middleware products, such as IBM WebSphere Application Server and IBM DB2. To understand the mechanics, it is important to distinguish between the System Monitoring shared services that are provided with PureApplication System.

6.3.1 System Monitoring

By default, this shared service deploys the Version 6.2.3 Fix Pack 5 of the IBM Tivoli Monitoring products. When the shared service is deployed, the following components are deployed across three VMs (see Figure 6-4):

- Hub-Tivoli Enterprise Monitoring Server
- Tivoli Enterprise Portal Server
- Remote-Tivoli Enterprise Monitoring Server
- Data-Warehouse

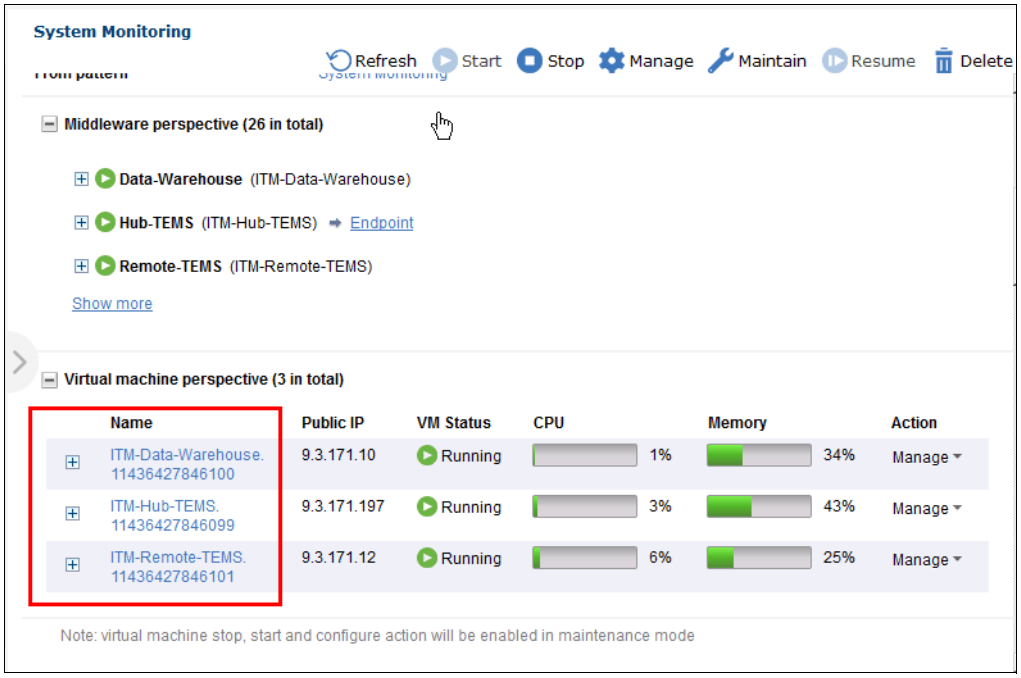


Figure 6-4 Components that are deployed from the shared service

Figure 6-5 on page 123 shows the architecture of the System Monitoring shared service in the PureApplication System.

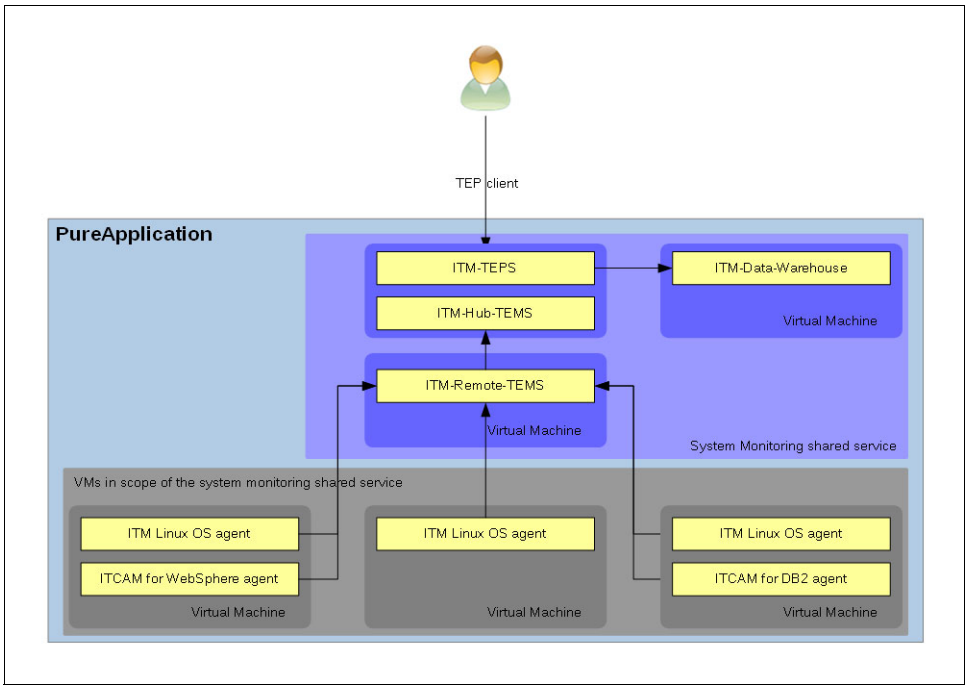


Figure 6-5 Architecture of the System Monitoring shared service in PureApplication System

When this shared service is deployed on PureApplication System, the IBM Tivoli Monitoring agent that monitors the OS and workload on each VM within the *scope* of the shared service is automatically integrated with the Remote-Tivoli Enterprise Monitoring Server of the shared service. Metrics from all those VMs automatically flow to the Hub-Tivoli Enterprise Monitoring Server. The metrics can be used to generate events and are stored so they can be analyzed later.

Note: The scope of shared services depends on whether the environment profile that is used to deploy the shared service is internally or externally managed (as shown in Figure 6-6).

The use of an externally managed environment profile is required when using externally managed cloud groups. The use of externally managed cloud groups has a number of advantages. For more information, see the developerWorks article at the following website:

http://www.ibm.com/developerworks/websphere/techjournal/1501_woolf/1501_woolf2.html

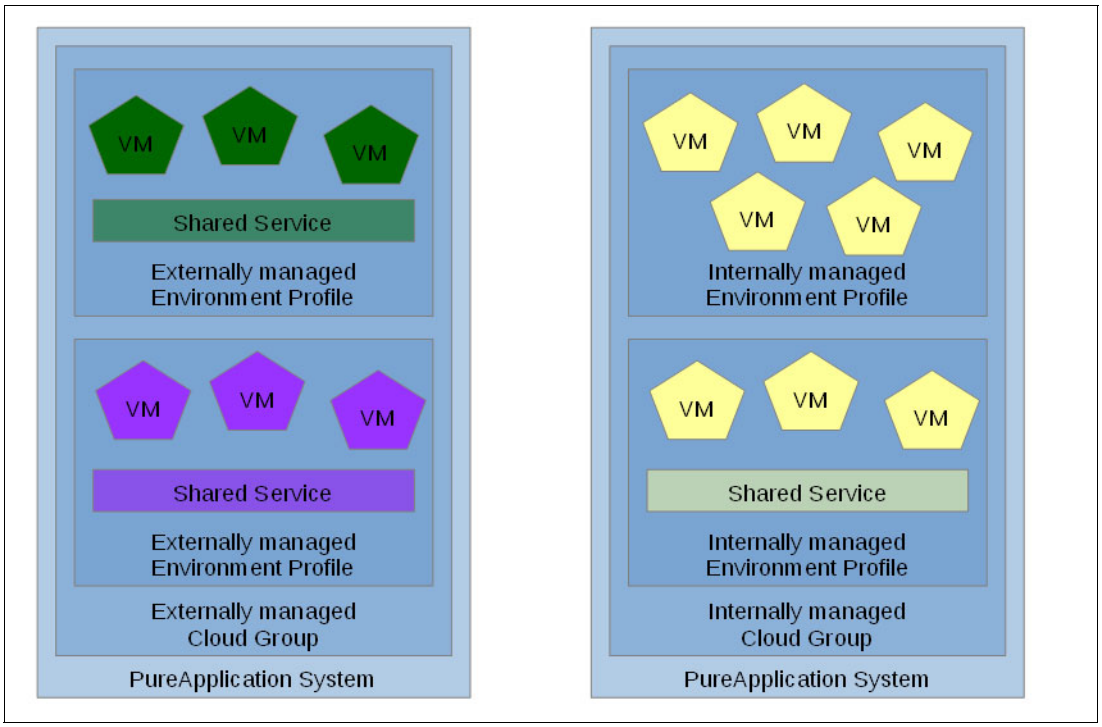


Figure 6-6 Understanding the scope of shared service instances

6.3.2 Middleware-specific System Monitoring services

In addition to the System Monitoring service, PureApplication products ship with specific shared services for each of the middleware products in which a customer might be interested. The customer is entitled to using monitoring shared services for WebSphere Application Server, HTTP Serve, and DB2. These shared services are deployed as zero VM instances, which means that these shared services do not result in a new VM being created, but they act as conduits for monitoring data to be forwarded to the Monitoring Service. Here is an explanation of some of the shared services that are shipped with PureApplication products:

- System Monitoring for HTTP Servers

  This shared service enables the IBM Tivoli Composite Application Monitoring agent for HTTP Servers on VMs that run IBM HTTP Server.

- System Monitoring for WebSphere Application Server

  This shared service enables the Tivoli Composite Application Monitoring agent for WebSphere Application Server on VMs that run a WebSphere Application Server node. The nodes can be a stand-alone server, a custom node, or a deployment manager.

- System Monitoring for DB2 Application Server

  This shared service enables the Tivoli Composite Application Monitoring agent for DB2 on VMs that run DB2.

  Note: This is only applicable to DB2 10.5 virtual system instances that are deployed by using Version 1.2.2.0 of the IBM DB2 with the BLU acceleration pattern type. Only Virtual System Patterns (VSPs) that use that pattern type are supported. No support is provided for “classic” DB2 VSPs.

  This pattern type is not installed by default with the 2.1.0.1 firmware, but can be downloaded separately. For more information, see the blog post at the following website:

  https://www.ibm.com/developerworks/community/blogs/explorepatterns/entry/New_version_of_DB2_with_BLU_Acceleration_Pattern_includes_DB2_10_5_Fix_Pack_5?lang=en

- System Monitoring for WebSphere MQ

  This shared service enables the Tivoli Composite Application Monitoring agent for WebSphere MQ on VMs that run WebSphere MQ.

  Note: At the time of writing, only the “classic” WebSphere MQ 7.5 Hypervisor Edition VSP is supported. VSPs that use the IBM MQ Advanced 8.0 pattern type are not supported.

- System Monitoring for WebSphere Message Broker

  This shared service enables the Tivoli Composite Application Monitoring agent for WebSphere Message Broker on VMs that run WebSphere MQ.

Before deploying any of these shared services, consider the *integration* of the System Monitoring shared service within the data center. By default, an entire set of IBM Tivoli Monitoring components is deployed on PureApplication System. This is a valid option for customers who do not have IBM Tivoli Monitoring deployed outside of PureApplication System. However, customers might want to integrate if they have an existing IBM Tivoli Monitoring environment. To facilitate integration, the System Monitoring shared service can be deployed in two different manners.

- Internal

  This is the default option and deploys the full set of IBM Tivoli Monitoring components that are listed earlier in this section.

  Note: This option requires integration of the IBM Tivoli Monitoring environment on PureApplication System within your data center. In particular, events from IBM Tivoli Monitoring must be routed so they can be picked up by the operations team.

- External

  This option deploys only the Remote-Tivoli Enterprise Monitoring Server IBM Tivoli Monitoring component on PureApplication System, but forwards all its metrics from agents on VMs that are in scope of the shared service to an existing Hub-Tivoli Enterprise Monitoring Server environment outside PureApplication System. One advantage of this approach is that you can leverage the integration of the existing environment within your data center. This approach allows monitoring of multiple racks along with other IT resources within the data center from a single IBM Tivoli Monitoring environment.

  In addition, the monitoring console can be viewed from the integrated PureApplication console. VM performance characteristics can be viewed. Figure 6-7 shows the architecture of its implementation in PureApplication System.

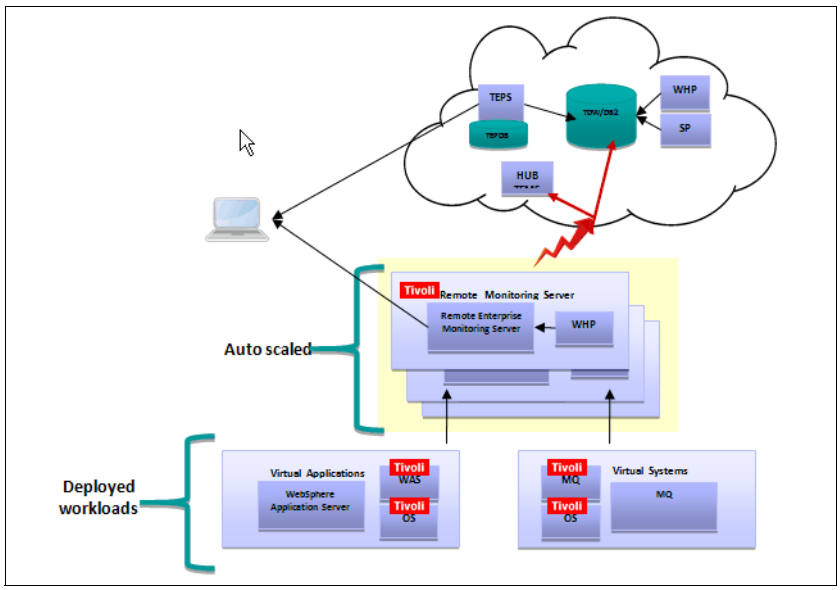


Figure 6-7 External configuration of the System Monitoring shared service

Note: This option comes with specific requirements for the version of the existing Hub-Tivoli Enterprise Monitoring Server environment. The currently supported version is V6.2.3 FP5 or later.

6.3.3 Using the PureApplication System Agent

You can use the PureApplication System Agent to monitor and troubleshoot the PureApplication System. The System Monitoring shared service is the base for this monitoring agent.

The PureApplication System Agent is automatically deployed with the System Monitoring shared service. The PureApplication System Agent collects monitoring data from the PureApplication System and monitoring data is presented in a workspace of the PureApplication System Monitoring Portal. The monitoring data indicates the overall health status of the PureApplication System. You can use the data to troubleshoot problems in your system and prevent problems from occurring.

The shared service can be deployed either as an internal System Monitoring shared service or an external System Monitoring shared service.

When the System Monitoring shared service is started internally, the PureApplication System Agent is deployed on the VM that hosts the Tivoli Enterprise Monitoring Server Hub.

In the case of an externally managed System Monitoring shared service, an instance of the PureApplication System Agent is deployed on each VM. Only one instance of the PureApplication System Agent is started. To monitor PureApplication System as another component in the data center, you can install the PureApplication System Agent on a system that is not part of PureApplication System. However, this is not a preferred option because of outdated architecture. The preferred architecture is to use System Monitoring in external mode (with embedded PureApplication System Agent) to connect to an enterprise IBM Tivoli Monitoring environment. If you already have an existing IBM Tivoli Monitoring environment with a monitoring server, a portal server, and a desktop portal, or a browser portal, you can choose this option.

6.3.4 Deploying the System Monitoring shared service

You can deploy the System Monitoring Shared Service by using the PureApplication System console. On the system console, click **Patterns** → **Shared Services** to display the list of Shared Services patterns. Before you deploy the shared service, verify that the user that is deploying the shared service has the following roles assigned:

- Allow delegation when full permission is selected
- Workload resources administration role with permission to manage workload resources (Full permission)
- Cloud group administration role with permission to view all cloud groups (Read-only)
- Hardware group administration with permission to view all hardware resources (Read-only)
- Security Administration role with permission to view users/groups (Read-only)

Assuming that you are using PureApplication System V2.1.0.1 firmware, you should explicitly confirm that the following software components are also present and enabled on the system:

- IBM OS image for Red Hat Linux Systems Version 2.1.2.0 build 79 or higher
- System Monitoring pattern type Version 1.0.4.0 or higher

To deploy the System Monitoring shared service, complete the following steps:

1. From the PureApplication Console, click **Patterns** → **Shared Services**.
2. From the list of shared services, expand **Monitoring Services** and select **System Monitoring 1.0.4.0**. Click **Deploy**.

3. You are prompted for the details to deploy the monitoring shared service (Figure 6-8).

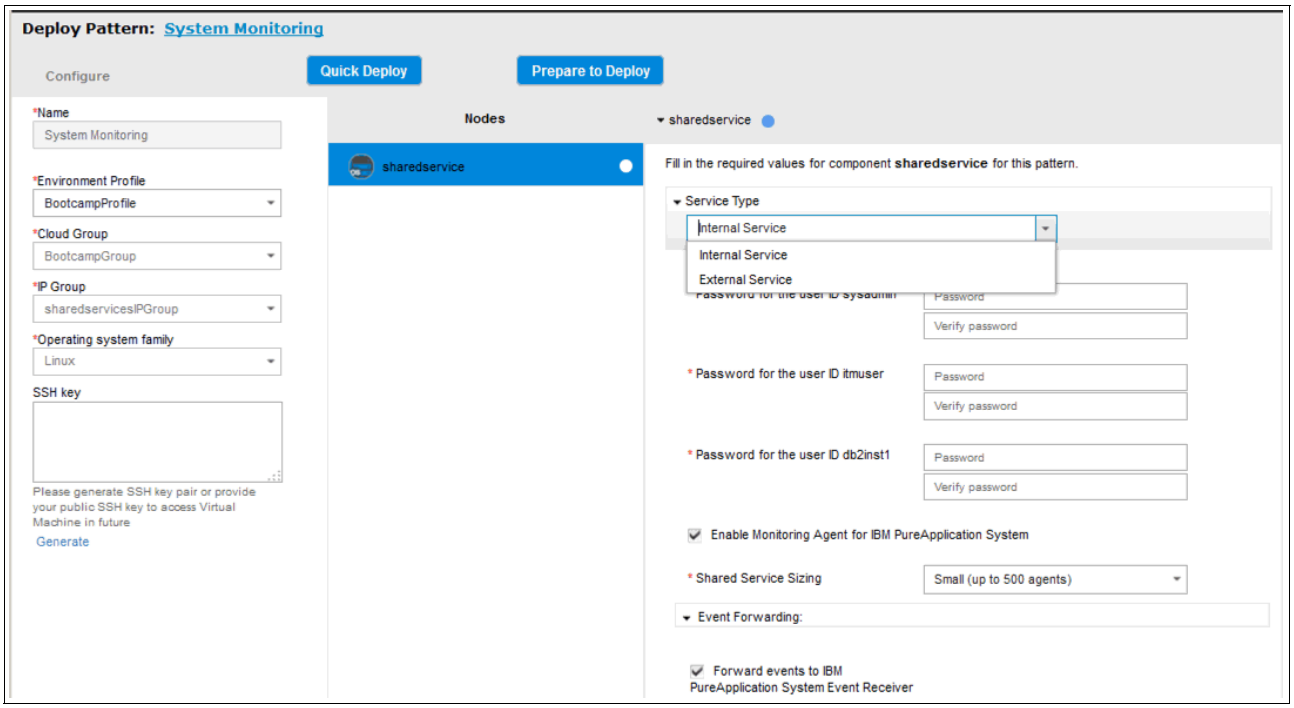


Figure 6-8 Deploy the System Monitoring shared service

4. If you are forwarding monitoring data to an external Tivoli Monitoring Server under Service Type, select **External Service**. If you want to use the internal Tivoli Monitoring capability within PureApplication System, select **Internal Service**.
5. If you select **External Service**, provide the details of the external monitoring server (Figure 6-9 on page 129).

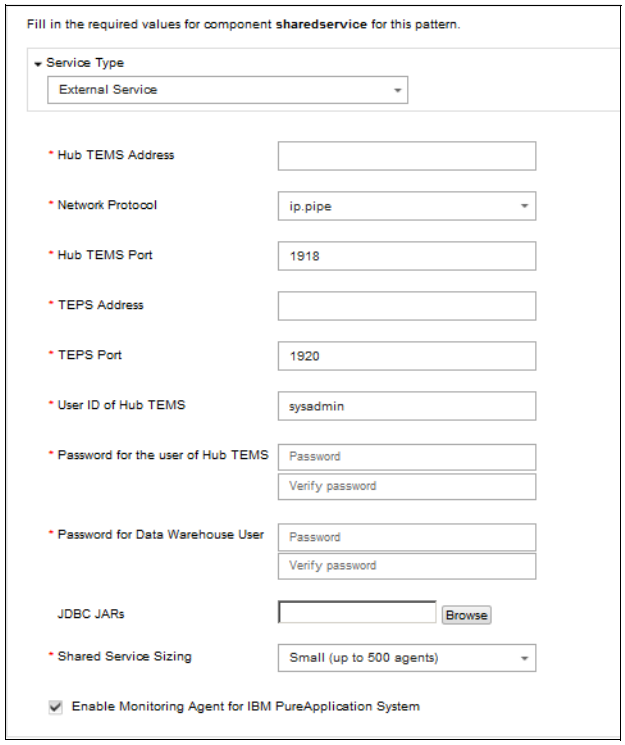


Figure 6-9 Deploy a Monitoring Shared Service to point to an external server

6. In the case of deploying the shared service as an internal monitoring system, the following information must be provided:
   - Password for the user ID sysadmin

     This is the password for the *sysadmin* user ID. This user ID is used for internal interaction with the HTEMS. Enter the same password again in the verification field. The password must be 15 characters or less.
   - Password for the user ID itmuser

     Enter a password for the *itmuser* user ID. This user ID is used to collect historical data. Enter the same password again in the verification field.
   - Password for the user ID db2inst1

     Enter a password for the *db2inst1* user ID. This user ID is used to create and maintain the data warehouse database. Enter the same password again in the verification field.
   - Shared Service Sizing

     Select the expected size of the internal System Monitoring shared service. The size is based on the number of monitoring agents that you plan to connect to the monitoring servers in that shared service. For more information about how this can be decided, see the *Deploying a System Monitoring shared service in internal mode* topic in the IBM Knowledge Center at the following website:

     http://www.ibm.com/support/knowledgecenter/SSCR9A_2.1.0/doc/ITM_shared_service/fac_deploy_monitoring_service.dita?cp=SSCR9A_2.1.0%2F6-3-6-0-0r

7. After you enter all the values, click **Quick Deploy** or **Prepare to Deploy**, depending on whether you are starting this shared service within a single rack or across multiple racks. The reason to choose one over the other depends on whether you are using an externally managed environment profile. If you use an externally managed profile and if you have multiple data and cloud management IP groups in that environment profile, you should always proceed to Prepare to Deploy so you can verify that the correct IP groups are used for deployment. In the case of an internally managed environment profile, you are presented with a list of data IP groups if your environment profile contains multiple IP groups.
8. After the deployment completes, view the running shared service by clicking **Patterns** → **Pattern Instances** → **Shared Service*s***, as shown in Figure 6-10.

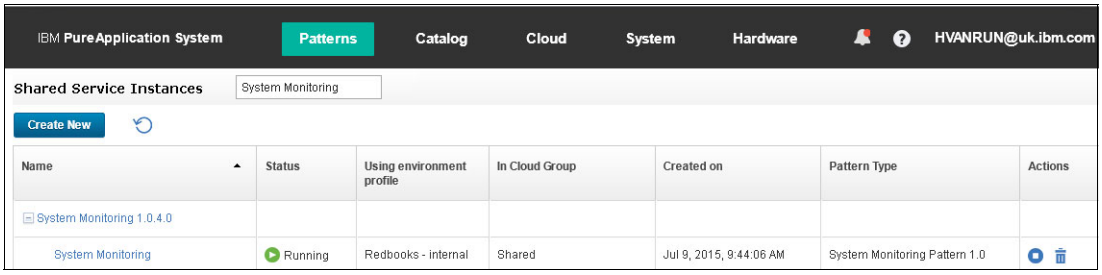


Figure 6-10 Deployed System Monitoring shared service instance

6.3.5 Deploying other System Monitoring shared services

You can extend the System Monitoring shared service to collect performance and availability data about middleware servers running in a cloud group's workloads. An explanation of the shared services that are available with PureApplication is in 6.3, "System Monitoring shared services" on page 121.

Before you deploy any of the middleware-related shared services, the base monitoring shared service must be deployed and in the *Running* state. To deploy any of the middleware shared services, complete the following steps:

1. Go to the PureApplication System console and click **Patterns** → **Shared Services**.
2. Select the Shared Service that you are interested in, for example, **System Monitoring for HTTP Servers**. Click **Deploy**.
3. The Configure and deploy a shared service dialog box opens (Figure 6-11 on page 131). It has the following options:
   - **Auto-start monitoring agent when deployed**: Enable this option to start the monitoring agent for IBM HTTP Server automatically when the HTTP server instances are started. This option is enabled by default.
   - **Restart IHS servers automatically**: Enable this option to restart monitored HTTP server instances.
   - **Auto-start monitoring agent on all deployments**: Enable this option to start all monitoring agents that are stopped.
   - **Start monitoring agent when existing deployments discover Shared Service for the first time**: Enable this option to install and start the monitoring agent on existing deployments that are not configured to forward monitoring data when they are deployed. This setting has no effect unless you also enable the **Auto-start monitoring agent when deployed** option. If you disable this option, the monitoring agent is installed on existing deployments, but you must manually start the agent.

4. Accept the defaults, and click **OK** to deploy the shared service into the correct cloud group.

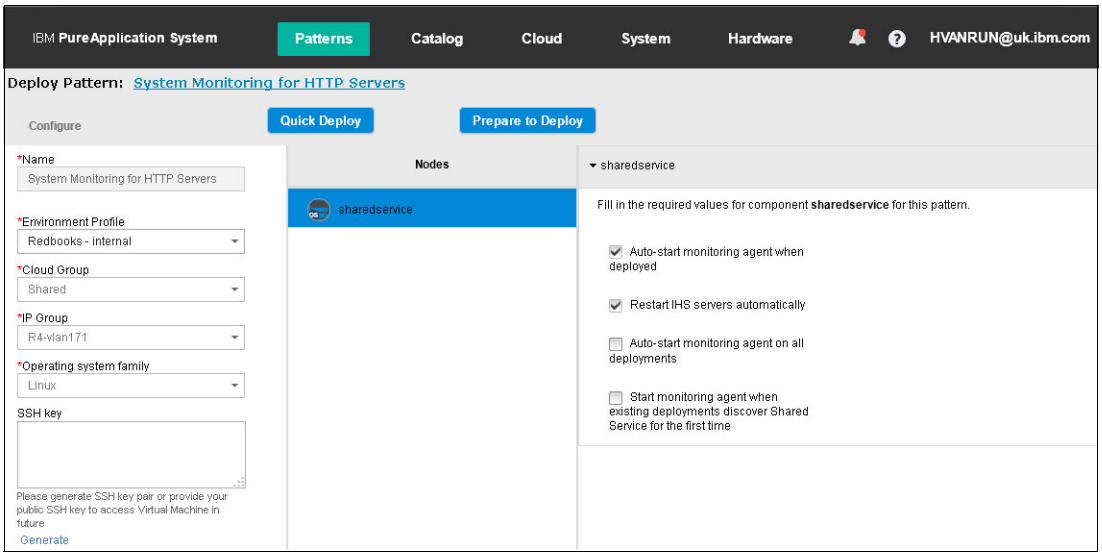


Figure 6-11 Deploying System Monitoring for HTTP Servers shared service

5. Complete steps 2 on page 130 to 4 to deploy any of the following shared services.
   - System Monitoring for WebSphere Application Server.
   - System Monitoring for DB2 Application Server.
   - System Monitoring for WebSphere MQ.
   - System Monitoring for WebSphere Message Broker.

Note: WebSphere MQ and WebSphere Message Broker shared services integrate only with classic VSPs.

Note: To use the System Monitoring for DB2 Application Server shared service with DB2 VSPs, you must have the IBM DB2 with BLU Acceleration Pattern type Version 1.2.2.0 installed and enabled.

This pattern type is not installed by default with the Version 2.1.0.1 firmware, but can be downloaded separately. For more information, see the blog post at the following website:

https://www.ibm.com/developerworks/community/blogs/explorepatterns/entry/New_version_of_DB2_with_BLU_Acceleration_Pattern_includes_DB2_10_5_Fix_Pack_5?lang=en

The deployment of these additional shared services does not involve deployment of actual VMs. Instead, they simply provide a signal to the System Monitoring shared service and specific IBM middleware on deployed VMs to be monitored. Effectively, this enables the automatic deployment and configuration of the corresponding Tivoli Composite Application Monitoring agent for the IBM middleware product.

Figure 6-12 shows all the System Monitoring shared services deployed. All the services have the same scope of cloud group *Shared*, which results from their deployment by using the internally managed environment profiled *Redbooks - internal*.

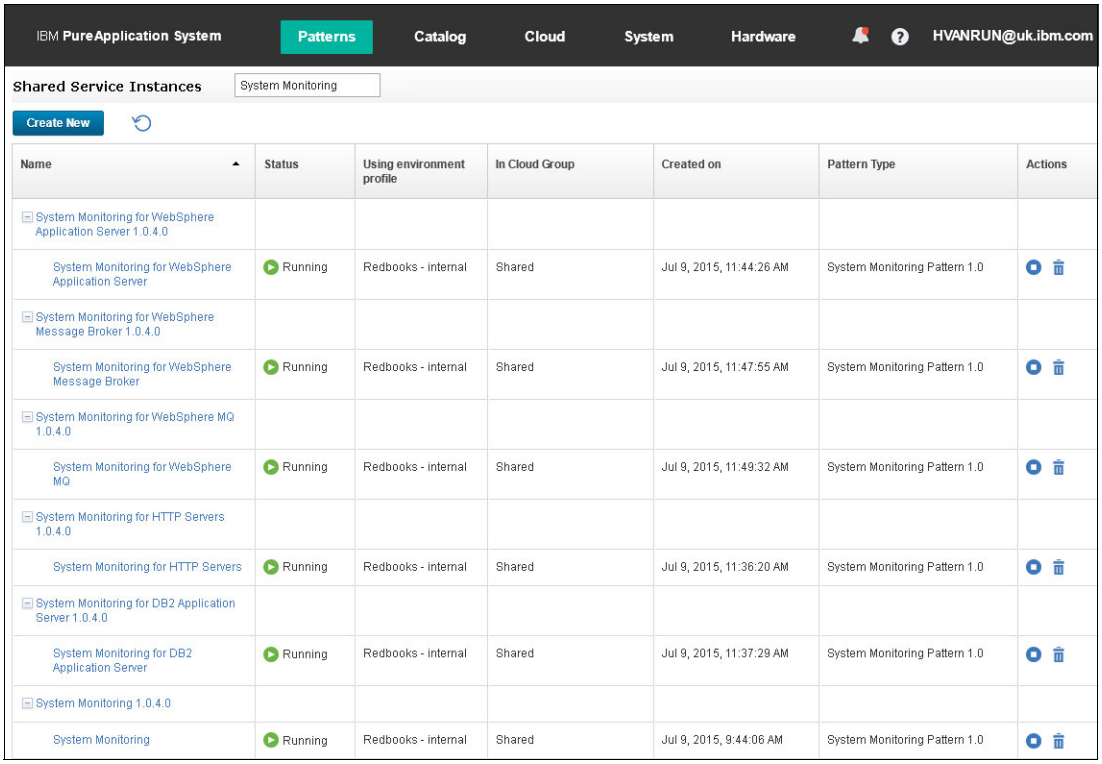


Figure 6-12 Overview of all System Monitoring shared services deployed

6.3.6 Using System Monitoring shared services

After the System Monitoring shared services deploy successfully, PureApplication System Monitoring Portal gives you access to the data that is collected by the monitoring shared services (monitoring agents). This section describes using the System Monitoring shared services.

Starting the PureApplication System Monitoring Portal

After the internal System Monitoring shared service is deployed, you can start the PureApplication System Monitoring Portal from the PureApplication System console.

Note: The PureApplication System Monitoring Portal is effectively a repackaged version of the Tivoli Enterprise Monitoring Portal client.

Note: Your client workstation must meet certain requirements. You need to have a 32-bit Java Runtime Environment (JRE) installed. For more information, see the *Viewing monitoring data for an internal System Monitoring shared service* topic at the IBM Knowledge Center, found at the following website:

`http://www.ibm.com/support/knowledgecenter/SSCR9A_2.1.0/doc/ITM_shared_service/fac_launch_tep_webstart.dita`

Alternatively, you can download and install a supported IBM JRE, as documented in the IBM Knowledge Center at the following website:

`http://www.ibm.com/support/knowledgecenter/SSCR9A_2.1.0/doc/agent_IPAS/fac_launch_tep_webstart_java.dita`

Note: Version 1.0.4.0 of the System Monitoring shared service is not compatible with Oracle Java 8 or Oracle Java 7 update 51 or higher.

To start the PureApplication System Monitoring Portal, complete the following steps:

1. Using the PureApplication System console, click **Patterns** → **Pattern instances** → **Shared services**. Find the System Monitoring shared service instance.
2. In the Middleware perspective, find **Hub-TEMS**. Click **Endpoint** (Figure 6-13).

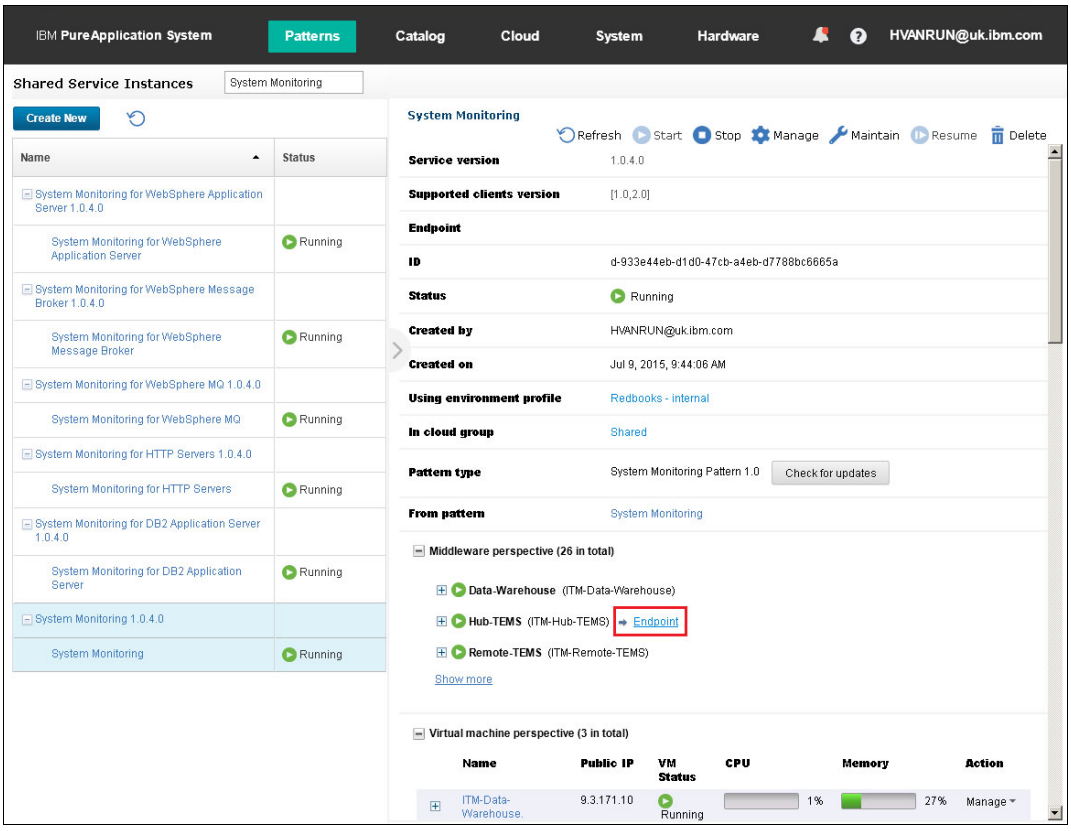


Figure 6-13 Find the Endpoint link to start the System Monitoring Portal client

3. In the dialog box, click the link for the PureApplication System Monitoring Portal (Figure 6-14).

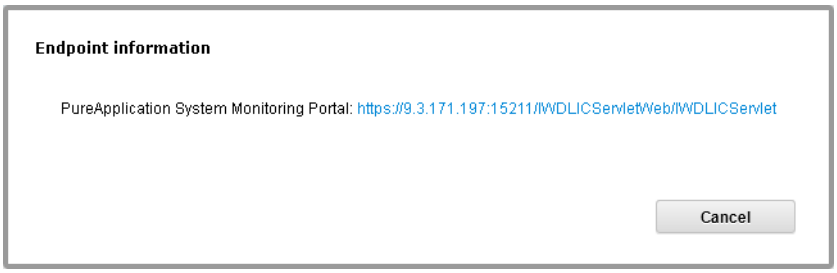


Figure 6-14 Start the PureApplication System Monitoring Portal

4. After the link is clicked, the Java Web Start application is started. Upon first start, the application is downloaded, as shown in Figure 6-15.

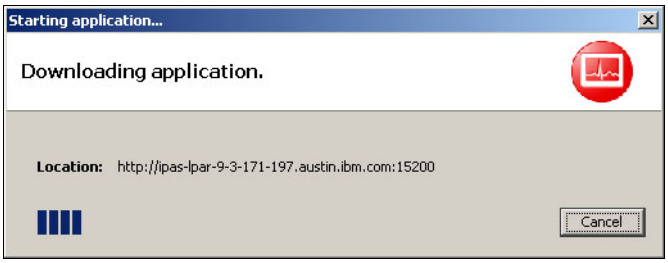


Figure 6-15 Download the Java Web Start application upon first start

5. Typically, a Security Warning dialog box opens, as shown in Figure 6-16. This dialog box opens because even though the application has a digital signature from a trusted certificate authority, its associated `.jnlp` file does not. Click **Run** to continue starting the application.

> **Note:** Select the **Do not show this again for apps from the publisher and location above** option. This action ensures that the Security Warning is not shown when you start the IBM PureApplication System Monitoring Portal again from your client workstation.

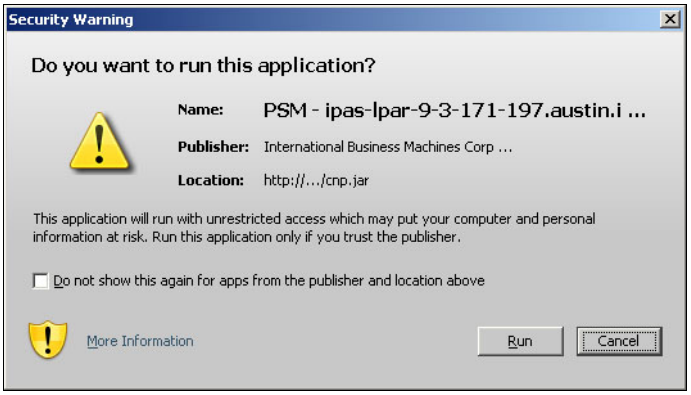


Figure 6-16 Accept the security warning

6. With the security warning accepted, the PureApplication System Monitoring application starts. You see the welcome window that is shown in Figure 6-17 on page 135.

Note: The version that is shown here does not match the version of the deployed shared service. This is a known problem with the System Monitoring shared service Version 1.0.4.0.

Figure 6-17 PureApplication System Monitoring welcome window

7. You should receive a security alert, as shown in Figure 6-18. Optionally, click **View Certificate** to examine the certificate. Select one the following options and click **OK** to continue.
   - **Accept this certificate permanently**
   - **Accept this certificate temporarily for this session only**

   Note: The application is attempting to perform a single sign-on by using the credentials that you logged on with to the PureApplication System web console. However, the SSL certificate that is installed on the System Monitoring Portal is a self-signed certificate that by default is not trusted by the Java Web Start application.

Figure 6-18 Accept the self-signed certificate from the Tivoli Enterprise Portal

8. The IBM PureApplication System Monitoring Portal window opens (Figure 6-19 on page 137).

   Note: The name of the window that is shown in Figure 6-19 includes the user name that starts the IBM PureApplication System Monitoring Portal from the PureApplication System web console, which indicates that single sign-on was used.

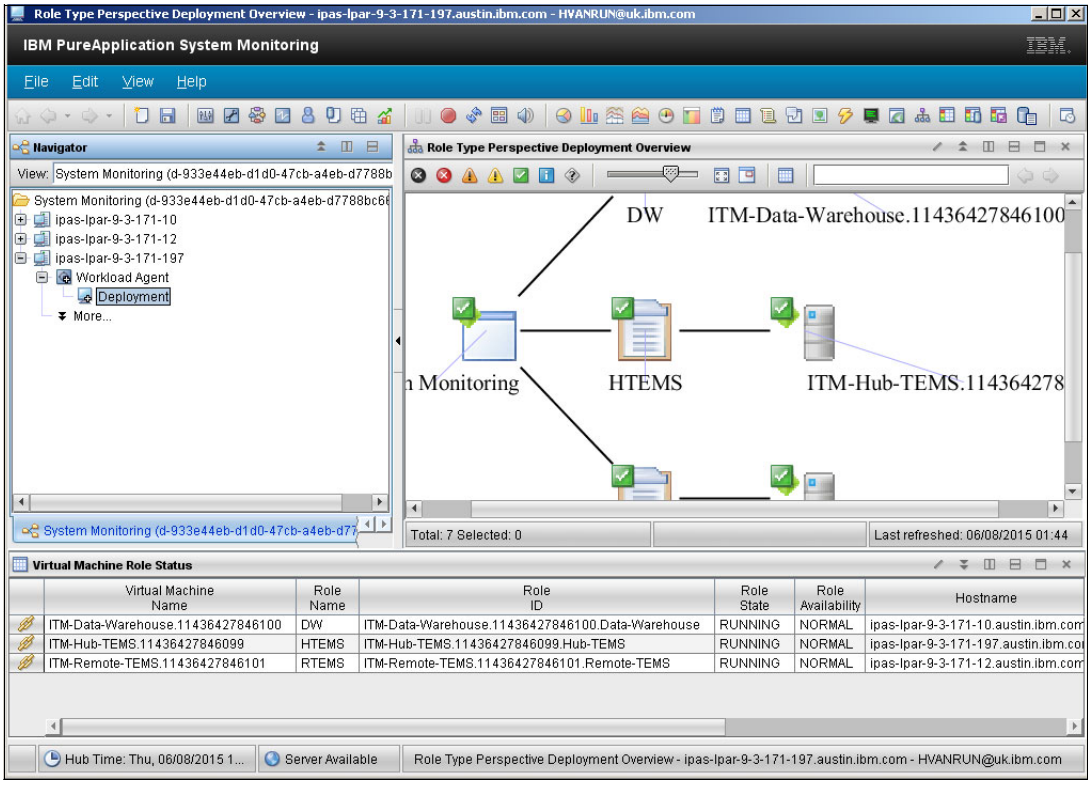


Figure 6-19 IBM PureApplication System Monitoring Console

Monitoring the operating system

To monitor the Operating System Level Metrics, complete the following steps:

1. From the Navigator of the IBM PureApplication System Monitoring console, click **Physical View**, as shown in Figure 6-20.

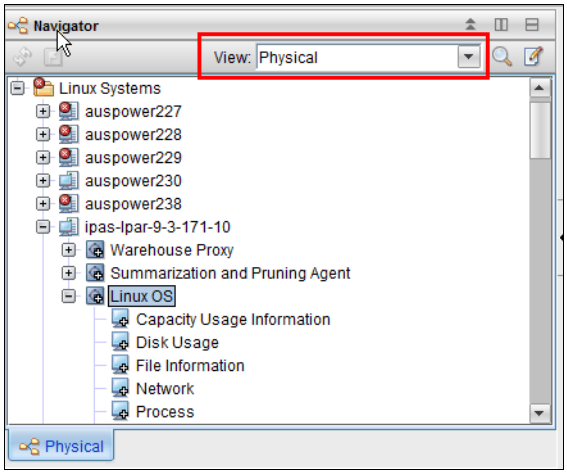


Figure 6-20 Select the Physical view in the IBM PureApplication System Monitoring Console

2. Expanding **Linux Systems** in the Navigator view (Figure 6-21) displays all the VMs that are deployed within the scope of the System Monitoring shared service instance.

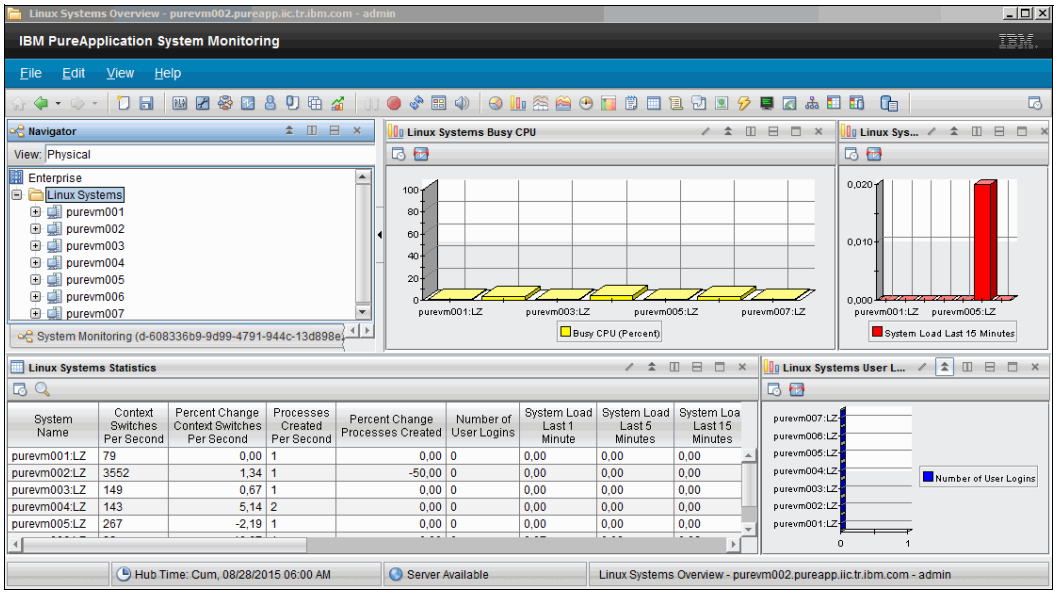


Figure 6-21 List of Linux systems in the Navigator view

3. Expand **Linux Systems** to display all the VMs in PureApplication System.
4. Select the VM in which you are interested. Expand **Linux OS** under the VM that is selected.
5. View the metrics that you are interested in at the operating system level for that VM. Figure 6-22 shows an example.

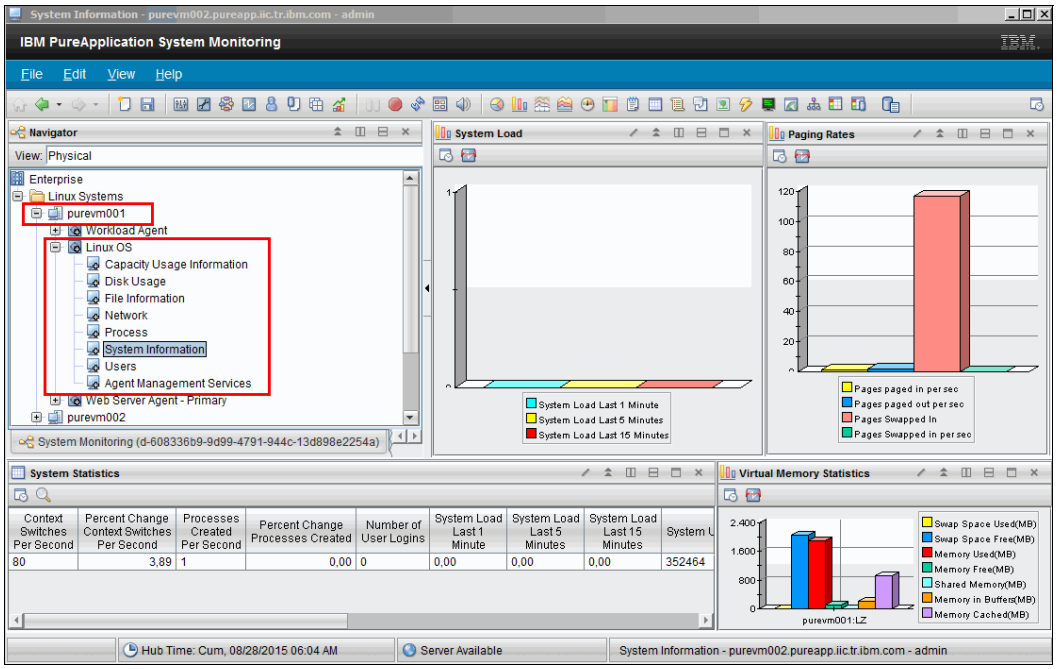


Figure 6-22 Monitor the operating system

6. Click the different system resources under the Linux OS, such as Disk Usage or Process, to view all the performance metrics.

Monitoring WebSphere Application Server

The prerequisite for monitoring WebSphere Application Server instances is the System Monitoring for WebSphere Application Server shared service. This shared service must be deployed and running. When monitoring a particular VM running WebSphere Application Server, it is better to go to the virtual system instance for the pattern and then select the VM. The monitoring portal starts with the data pertaining to the particular VM you are monitoring.

Complete the following steps:

1. From the Instances window, select the instance that you want to monitor.
2. Expand the VM perspective. Expand the VM that you want to monitor. Figure 6-23 shows an example.

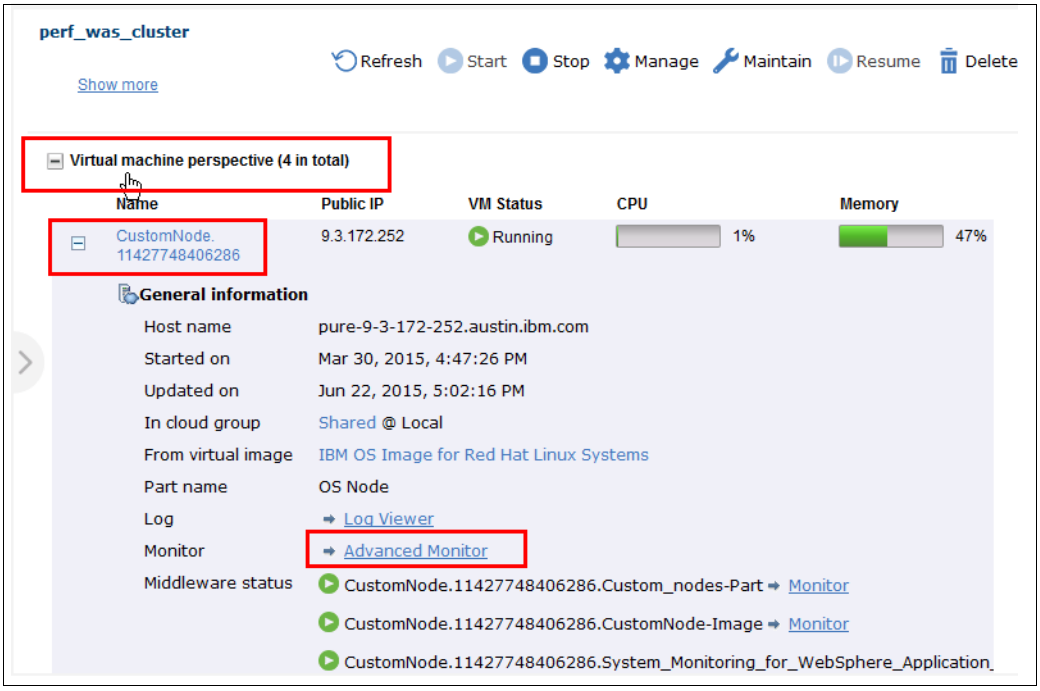


Figure 6-23 Monitor Virtual System Instances window

3. Click the **Advanced Monitor** link to start the Tivoli Monitoring console with a context-specific view to that VM, as shown in Figure 6-24.

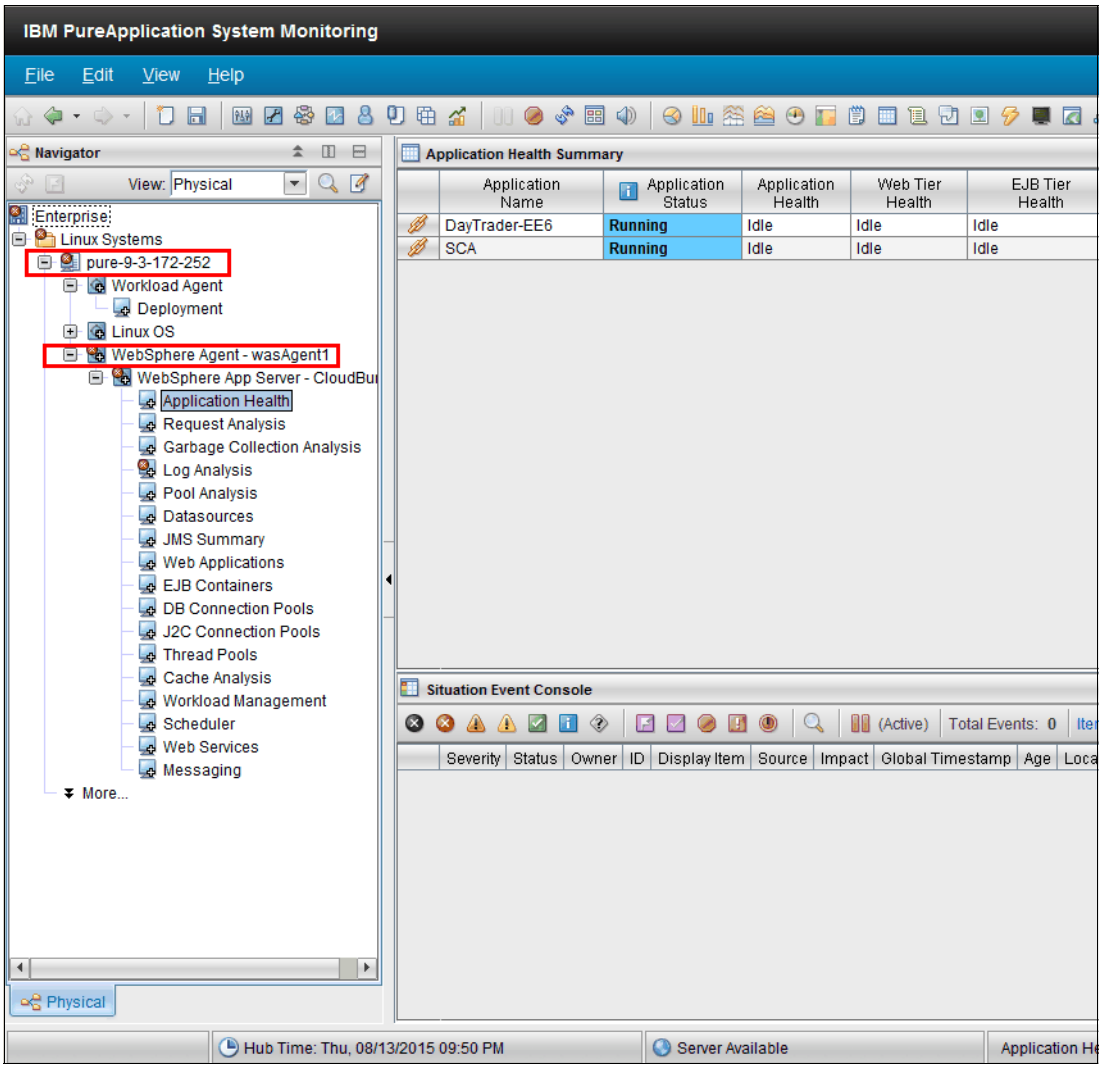


Figure 6-24 Monitor WebSphere Application Server

Monitoring IBM HTTP Server

The prerequisite for monitoring IBM HTTP Server is a successful deployment of Shared Service for IBM HTTP Server. To monitor an IBM HTTP Server, complete the following steps:

1. Click **Patterns** → **Virtual System Instances**.
2. From the Instances pane, select the instance to monitor.
3. Expand the VM perspective. Click the **Advanced Monitor** link next to Monitor, as shown in Figure 6-25 on page 141.

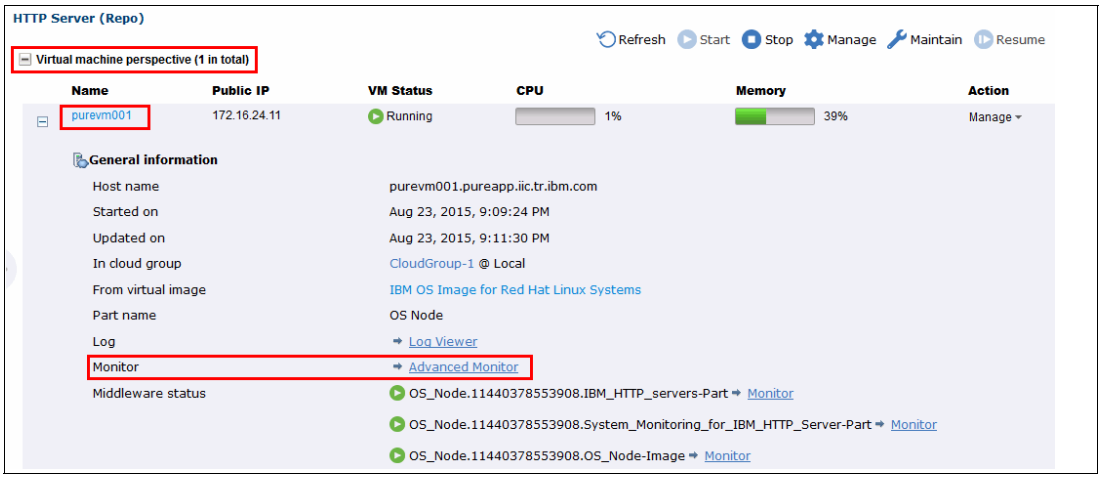


Figure 6-25 Use the Advanced Monitor link from the virtual system instance

4. Click the **Advanced Monitor** link to start the PureApplication Monitor console. Logging in to the console starts a context-sensitive Physical View within the navigator. The VM hosting the HTTP Server is focused.

5. Expand **WebServer Agent** and select **httpd** to view various web server-related metrics, as shown in Figure 6-26.

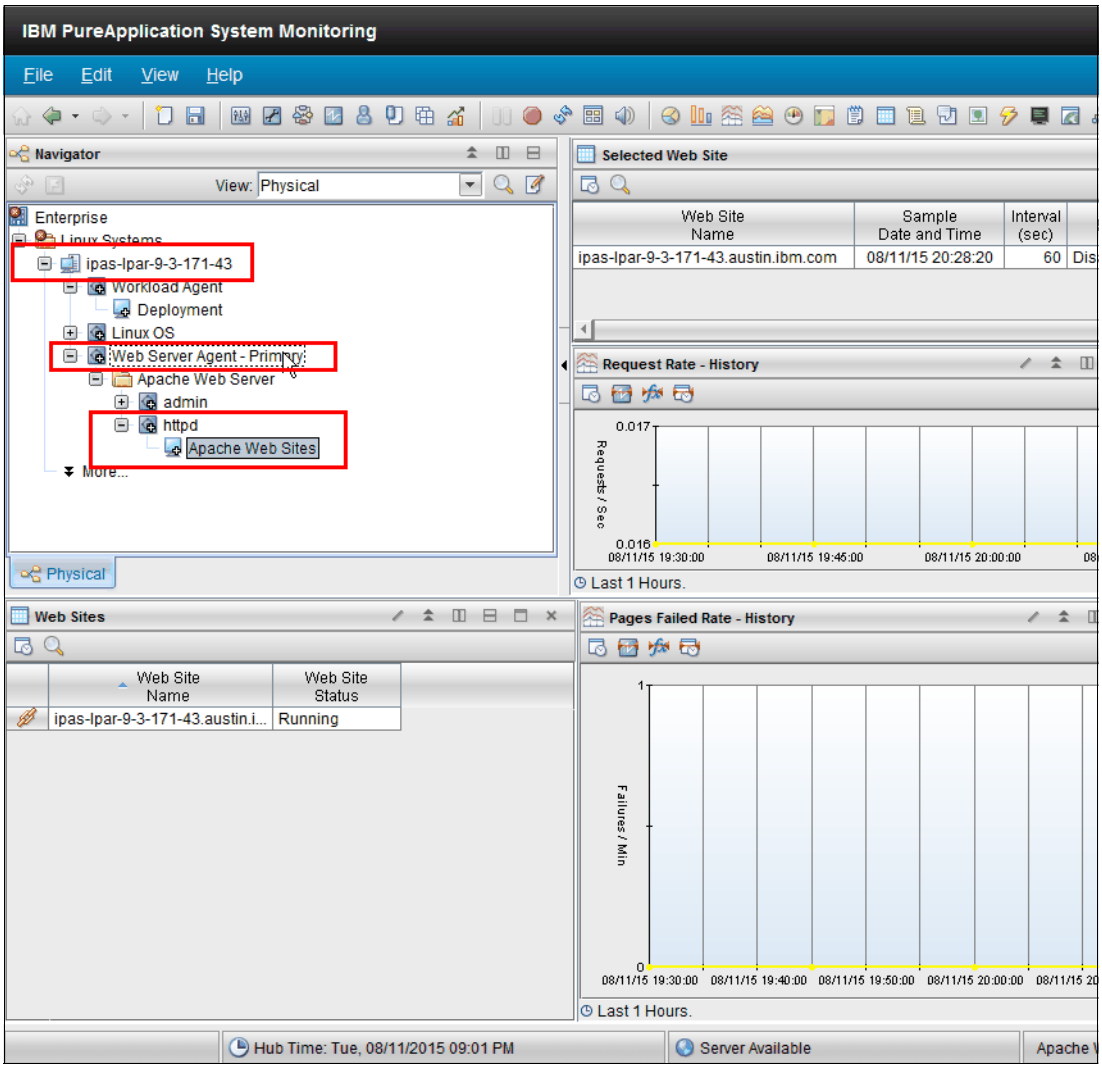


Figure 6-26 Monitor the Console WebServer View

Monitoring DB2

To monitor DB, complete the following steps:

1. From the Navigator view of the PureApplication Monitoring Console, select the VM that hosts your DB2 database instance and expand it.
2. If the VM hosts a DB2 database, the name should be similar to DB2-<Your Instance Name>:Machine Name.
3. Expand the database view, as shown in Figure 6-27 on page 143, to view the available DB2 database metrics.

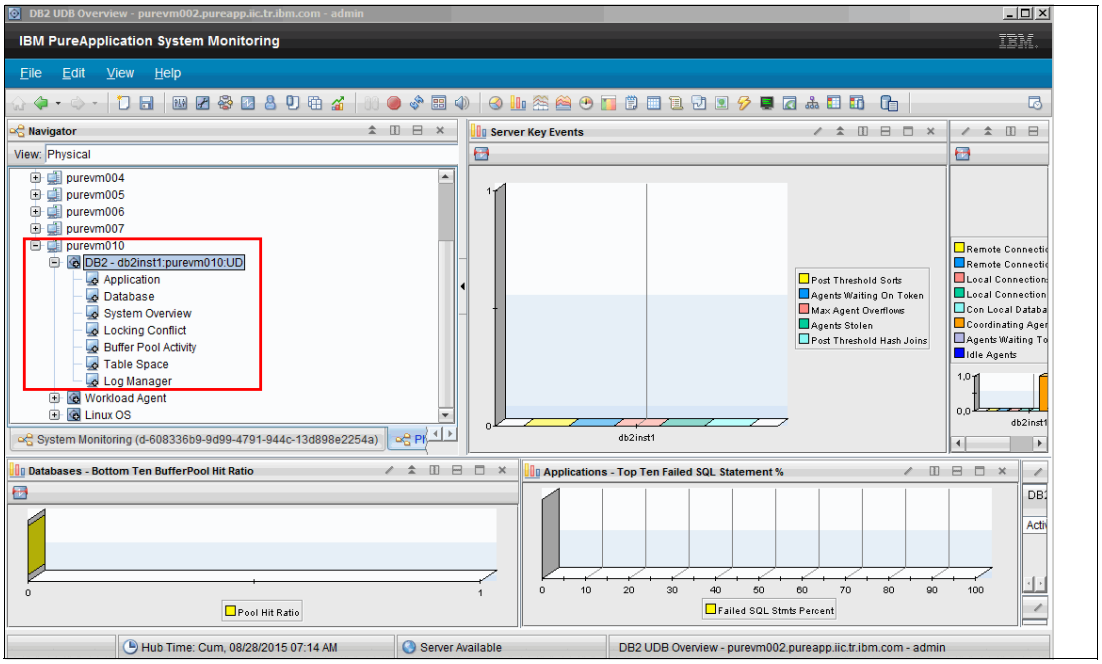


Figure 6-27 DB2 database instance metrics that are available

6.3.7 Situations in the System Monitoring shared service

As described in 6.3.6, “Using System Monitoring shared services” on page 132, the System Monitoring shared service can collect various metrics, including metrics about the operating system and many IBM middleware products. You can use the PureApplication System Monitoring Portal to examine those metrics. However, what is typically needed in production environments is something that evaluates some metrics against certain thresholds. For example, you might want to know when a file system is using over 90% of its capacity, or when the lock wait time in DB2 exceeds a certain number of seconds.

You can define *situations* in the PureApplication System Monitoring Portal to achieve these goals. A situation defines a certain set of conditions for a number of metrics. When those conditions are met, an *event* is automatically generated. Events can have different states, for example, “Warning” or “Critical”. These events from a System Monitoring shared service instance can be forwarded to another system, which allows for operations teams to be automatically notified of those events. For more information about event forwarding, see 6.3.8, “Forwarding events from System Monitoring shared service” on page 149.

For more information about situations, go to the PureApplication System IBM Knowledge Center, found at the following website:

https://www.ibm.com/support/knowledgecenter/SSCR9A_2.1.0/doc/ITM_portal/adminuse/situation_info_tep.dita

Defining a new situation for a DB2 instance

To illustrate how situations work, consider an example that has a DB2 virtual system instance running a DB2 10.5 database instance that is an online transaction processing database (OLTPDB). The administrator decides to track lock wait times. Excessive lock wait times are not preferable in online transaction processing applications because they can slow down application transactions. To define a new situation that automatically generates a *Warning* event when a lock time exceeds 1000 milliseconds (ms), complete the following steps:

1. Make sure that you have an IBM DB2 10.5 Virtual System Instance running. It must be deployed within the *scope* of the System Monitoring shared services instance. For more information, see Figure 6-6 on page 124.
2. Start the PureApplication System Monitoring Portal, as described in “Starting the PureApplication System Monitoring Portal” on page 132.
3. Select **Physical** under **View**, expand **Enterprise**, and find the system corresponding to your DB2 10.5 Virtual System Instance. Expand the system and find the DB2 instance, as shown in Figure 6-28.

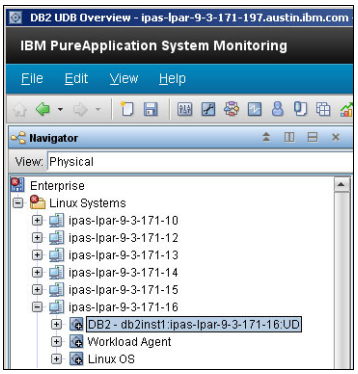


Figure 6-28 Find the DB2 instance in the PureApplication System Monitoring Portal

4. Right-click the DB2 instance and select **Situations**. This action opens the *Situation Editor*, as shown in Figure 6-29.

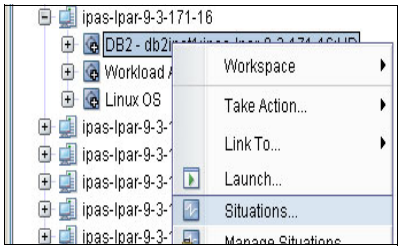


Figure 6-29 Start the Situation Editor

5. Now, right-click **DB2** and select **Create New**.
6. Click **Add conditions** and select the following attributes, as shown in Figure 6-30 on page 145:
   - Condition Type: Attribute Comparison
   - Attribute Group: DB2 Locking Conflict
   - Attribute Item: Lock Wait Time

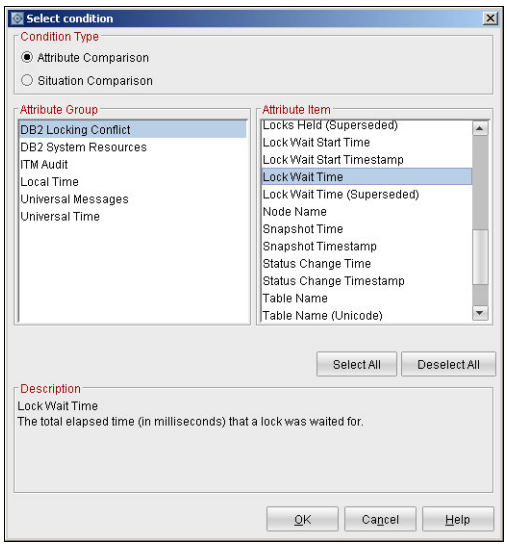


Figure 6-30 Specify a condition for the situation

7. Complete the rest of the situation by entering the information that is listed below. When complete, your situation should like Figure 6-31.
   - Name: Lock_wait_1000_ms
   - Description: Issue a warning when a lock wait time exceeds 1000 ms.
   - Sampling interval: 30 seconds
   - State: Warning
   - Run at startup: Enabled

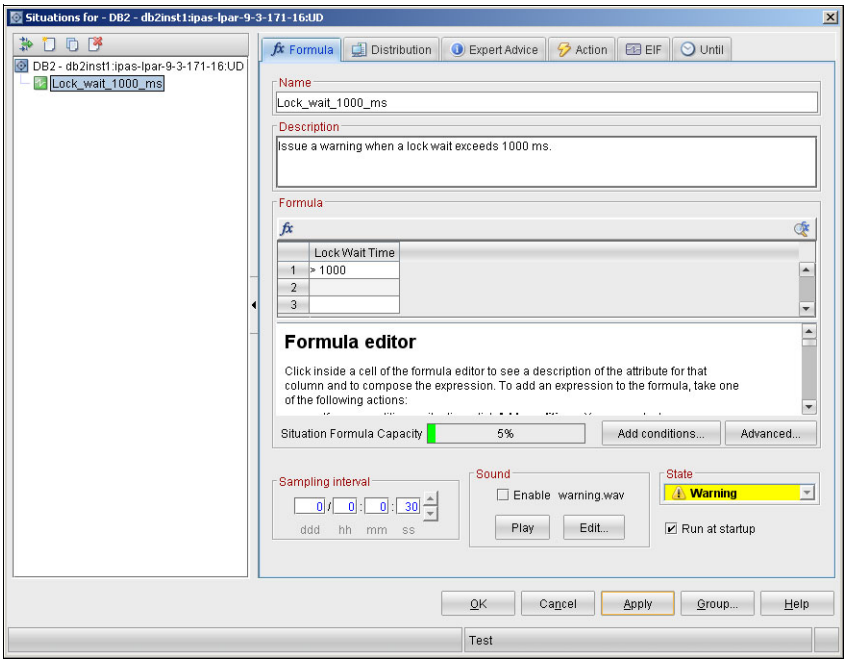


Figure 6-31 The complete definition for the situation

8. Select the **EIF** tab and configure this situation for event forwarding, as shown in Figure 6-32. In particular, specify the following values:
   - Forwarded events to EIF Receiver: Enabled
   - Severity: Warning
   - Assigned EIF Receivers: extTEC1

> **Note:** The EIF Receiver “extTEC1” is configured by default to forward events to the PureApplication System event receiver. For more information, see 6.3.8, “Forwarding events from System Monitoring shared service” on page 149.

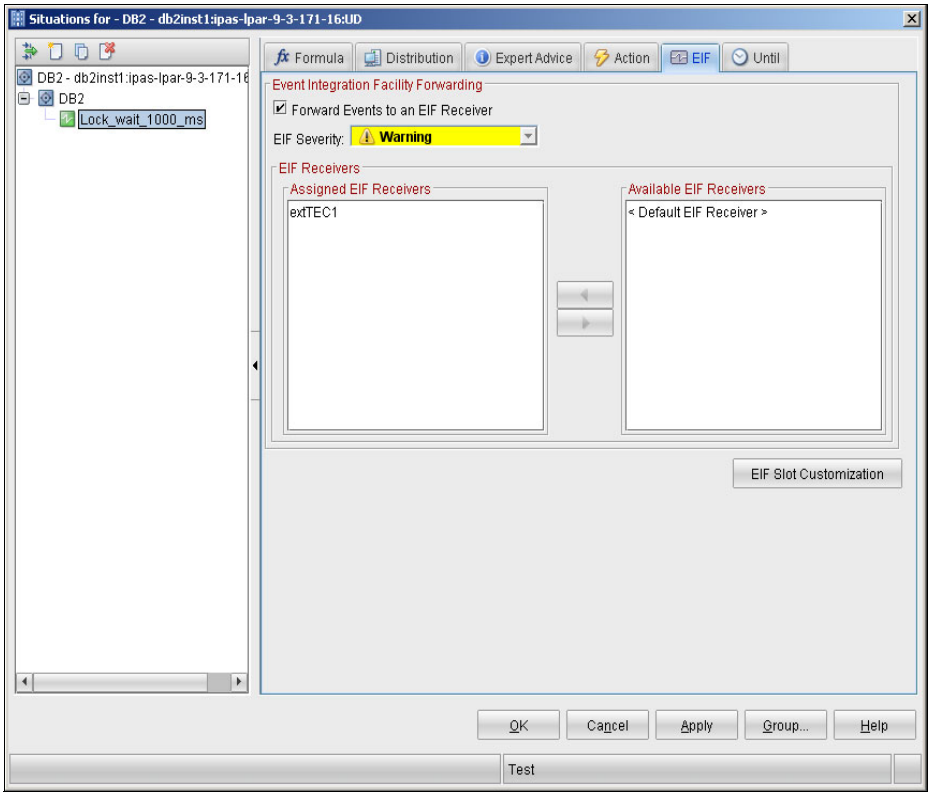


Figure 6-32 Enable EIF event forwarding for the situation

9. Click **OK** to save and enable the new situation.
10. Now, open two DB2 client sessions to the DB2 database OLTPDB that is hosted by the DB2 10.5 Virtual System Instance, as shown in Example 6-1.

Example 6-1 Open a DB2 client session to the OLTPDB on the DB2 10.5 Virtual System Instance

```
login as: root
root@9.3.171.16's password:
-bash-4.1# su - db2inst1
[db2inst1@ipas-lpar-9-3-171-16 ~]$ db2 connect to oltpdb

   Database Connection Information

 Database server        = DB2/LINUXX8664 10.5.5
 SQL authorization ID   = DB2INST1
```

```
 Local database alias   = OLTPDB
[db2inst1@ipas-lpar-9-3-171-16 ~]$
```

11. From the first session, run the commands that are shown in Example 6-2.

Example 6-2 Create a simple table and insert a record into the OLTPDB database

```
[db2inst1@ipas-lpar-9-3-171-25 ~]$ db2 "CREATE TABLE AUTHOR (AUTHOR_NUMBER INT
NOT NULL PRIMARY KEY, AUTHOR_NAME VARCHAR(20) NOT NULL)"
[db2inst1@ipas-lpar-9-3-171-25 ~]$ db2 +c "INSERT INTO AUTHOR (AUTHOR_NUMBER,
AUTHOR_NAME) VALUES (1, 'Margaret Ticknor')"
```

12. Run the command that is shown in Example 6-3 from the second session. The command does not return because this insert is waiting for a lock from the insert started by the other session. The lock that is held by the other session remains until it issues a commit or rollback.

Example 6-3 Perform the insert from the first session with auto commit disabled

```
[db2inst1@ipas-lpar-9-3-171-25 ~]$ db2 +c "INSERT INTO AUTHOR (AUTHOR_NUMBER,
AUTHOR_NAME) VALUES (1, 'Kyle Brown')"
```

13. Make sure to wait at least a couple of seconds and then run **db2 commit** from the first session.

Example 6-4 Commit the insert from the first session after a few seconds

```
[db2inst1@ipas-lpar-9-3-171-16 ~]$ db2 commit
DB20000I  The SQL command completed successfully.
```

14. The command from the second session returns an error regarding a duplicate key, as shown in Example 6-5. The second session has been waiting for a lock all this time. By the time it obtained the lock, it was no longer able to perform the insert because the value with key 1 had been inserted by the first session.

Example 6-5 Second session returns with an error regarding a duplicate key

```
[db2inst1@ipas-lpar-9-3-171-16 ~]$ db2 +c "INSERT INTO AUTHOR (AUTHOR_NUMBER,
AUTHOR_NAME) VALUES (1, 'Kyle Brown')"
DB21034E  The command was processed as an SQL statement because it was not a
valid Command Line Processor command.  During SQL processing it returned:
SQL0803N  One or more values in the INSERT statement, UPDATE statement, or
foreign key update caused by a DELETE statement are not valid because the
primary key, unique constraint or unique index identified by "1" constrains
table "DB2INST1.AUTHOR" from having duplicate values for the index key.
SQLSTATE=23505
```

This action should have triggered the situation that is previously defined. You should see a “Warning” event in the PureApplication System Monitoring Portal, as shown in Figure 6-33.

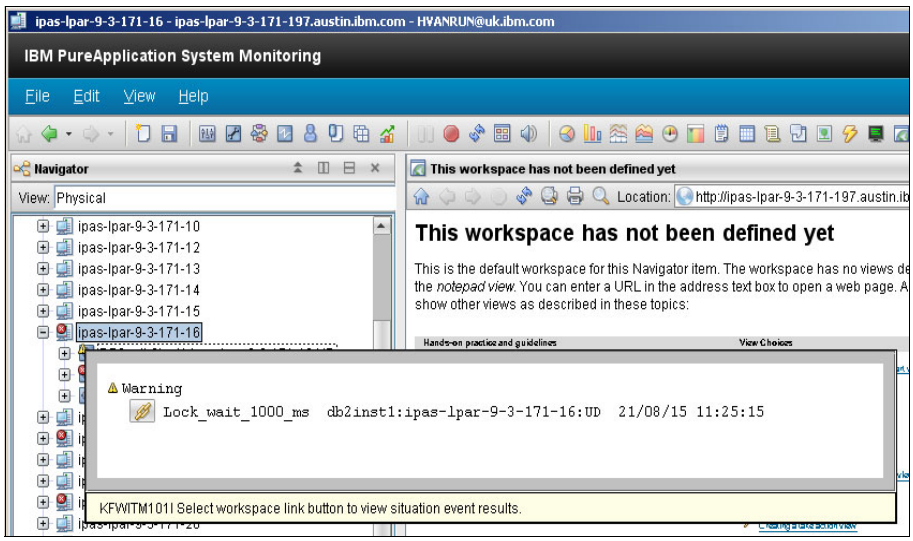


Figure 6-33 The “Warning” situation in the PureApplication System Monitoring Portal

15. As described in “Forwarding to PureApplication System Event Receiver” on page 149, events are forwarded to the PureApplication System event receiver by default. Because you enabled EIF forwarding for the situation, this event should be visible from the PureApplication web console. To confirm that this event was indeed forwarded, click **System** → **Events**, as shown in Figure 6-34.

> **Note:** You can use the filters to limit the number of events that are shown in the PureApplication System console events window.
>
> **Note:** The time stamps of the events that are reported in the PureApplication System Monitoring Portal and the PureApplication web console differ by exactly two hours. The PureApplication web interface was used from a system that uses Central European Time (CET), and the System Monitoring shared service was deployed by using an environment profile with a time zone of “Default time zone configured for the virtual image”. As a result, the VMs hosting the shared service are set to Coordinated Universal Time (UTC), which is exactly two hours behind CET.

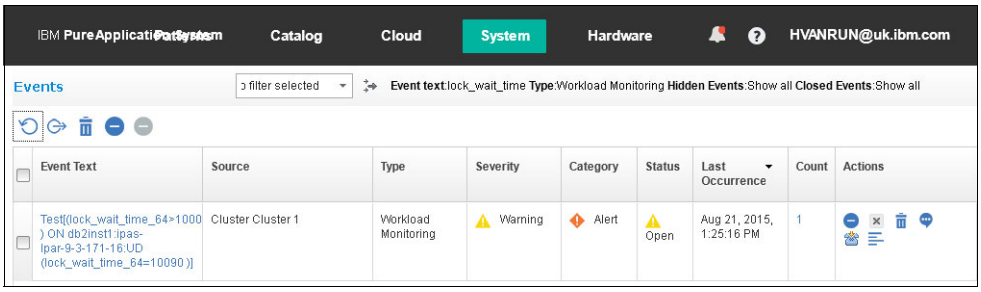


Figure 6-34 Forwarded event in PureApplication System console

6.3.8 Forwarding events from System Monitoring shared service

This section describes forwarding events from the System Monitoring shared service.

Forwarding to PureApplication System Event Receiver

Forward to PureApplication System Event Receiver is an option at deployment time of the System Monitoring shared service, but this option also can be changed after deployment time.

To set this option, complete the following steps:

1. From the shared service instance, click **Manage** to open the Instance Console.
2. Click **Operations** to go to the Operations tab.
3. From the menu on the left, select **ITM-Hub-TEMS.Hub-TEMS**. In the right pane, expand **Fundamental** → **Add Event Forwarding**.
4. Select **Forward events to IBM PureApplication System Event Receiver**, as shown in Figure 6-35, and click **Submit**.

> **Note:** The changes are made after the operation completes.

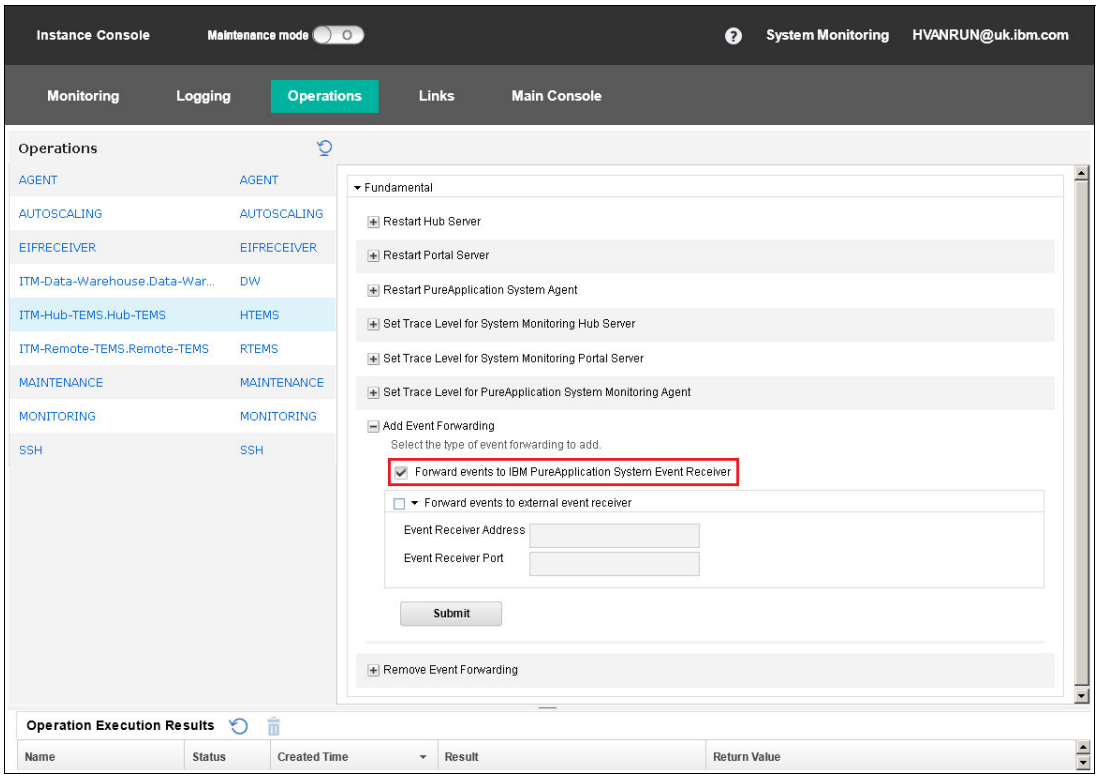


Figure 6-35 Enable forwarding of events to PureApplication System Event Receiver

When event forwarding to the PureApplication System event receiver is enabled, events from the System Monitoring shared service are listed in the PureApplication events table. You can see an example of this task in Figure 6-36.

Note: All forwarded events are of type “Workload Monitoring”. This designation makes it easy to distinguish them from other events from the PureApplication System or Software instance.

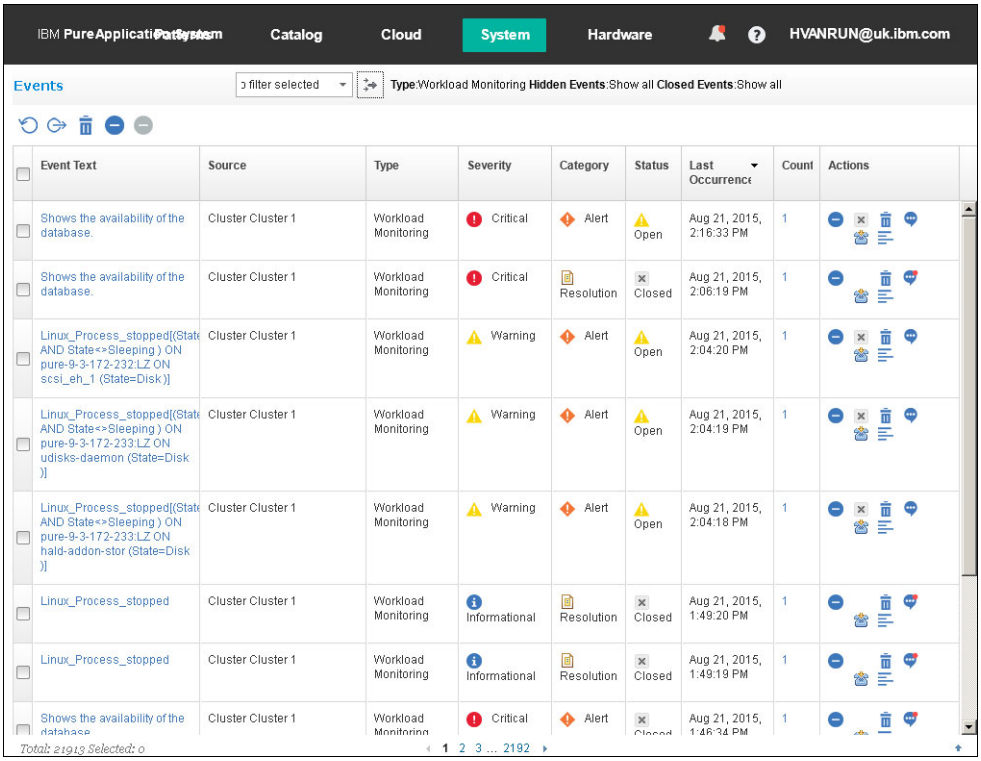


Figure 6-36 Forwarded events of type “Workload Monitoring” in PureApplication System events table

Forwarding events to an external system

Some customers prefer to separate System Monitoring events from PureApplication events. However, this means that events must be forwarded directly from the System Monitor, as shown in Figure 6-37.

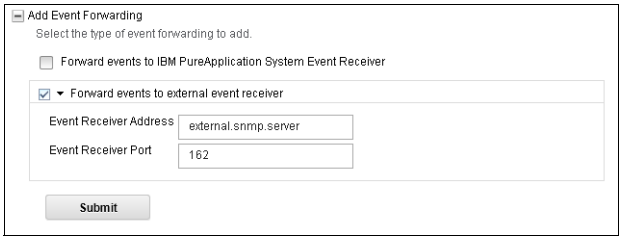


Figure 6-37 Enable forwarding of events to an external SNMP trap destination

Event Integration Facility (EIF) is an IBM Tivoli proprietary standard. Compared to SNMP, EIF allows for more detailed event information to be forwarded to an external, centralized service-level management (SLM) system, such as IBM Tivoli Netcool/OMNIbus.

As stated in the IBM Knowledge Center for Tivoli Composite Application Manager for Applications 7.2.0: “Use the EIF (Event Integration Facility) tab to forward situation events to one or more EIF receivers (such as an IBM Tivoli Enterprise Console® Server or Netcool/OMNIbus Probe for Tivoli EIF).” Figure 6-38 shows the details.

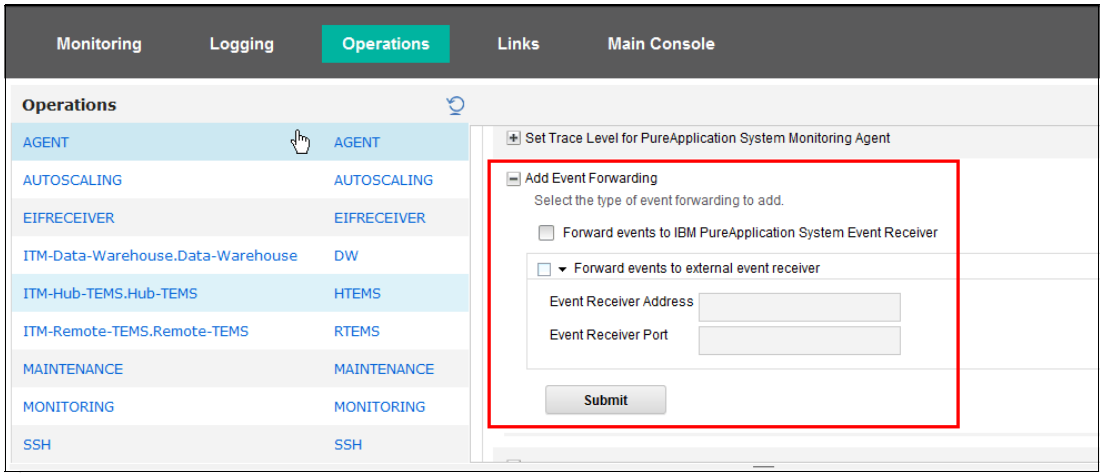


Figure 6-38 Enable forwarding of events by using EIF

6.4 Using Optim for performance monitoring of DB2

PureApplication System is delivered with a shared service that provides the monitoring infrastructure that is required to facilitate the collection of performance and availability information for DB2. With this information, you can troubleshoot databases and make business decisions regarding your service usage. The *Database Performance Monitor* is a tool for assisting with troubleshooting performance problems in DB2 and not necessarily for actively monitoring DB2. For monitoring DB2, use Shared Service for DB2 and monitor the events in the IBM Tivoli Monitoring Console. For more information, see “Monitoring DB2” on page 142. To deploy The Performance Monitoring shared service, complete the following steps:

1. On the PureApplication System console, click **Patterns** → **Shared Services**.

2. Select **Monitoring Services** → **Database Performance Monitoring**. Click **Deploy** to open the pattern deployment window, as shown in Figure 6-39.

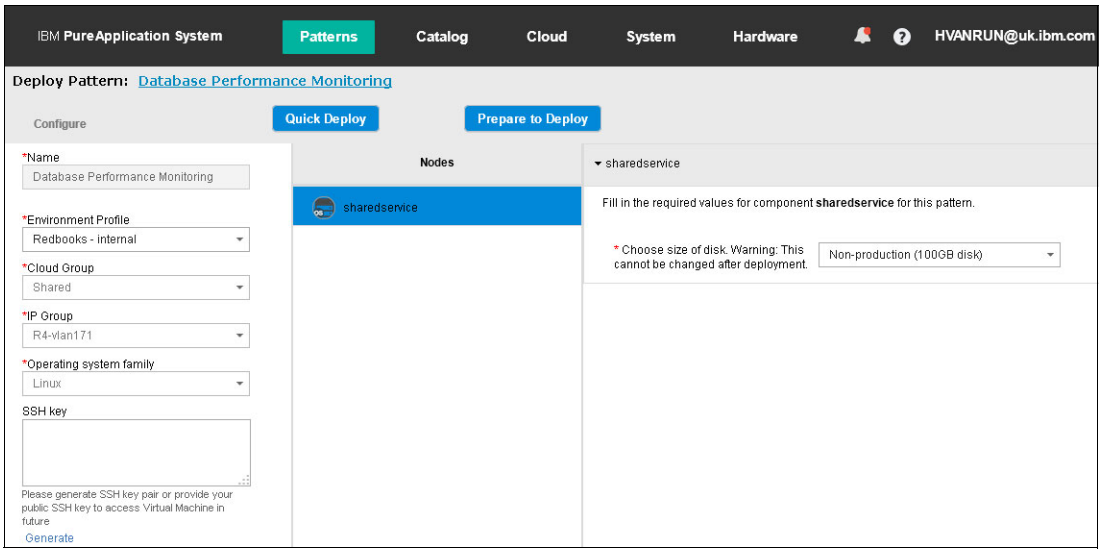


Figure 6-39 Deploy the Database Performance Monitoring Shared Service

After the Database Performance Monitoring Shared Service is deployed and running, complete the following steps to use the performance monitoring tool:

1. On the PureApplication System console, click **Patterns** → **Patterns Instances** → **Shared Services**.
2. Expand **Database Performance Monitoring** and select the instance that you want to monitor.

3. From the Instances detail window, click the **Endpoint** link under Middleware Perspective to start the Performance Monitor, as shown in Figure 6-40.

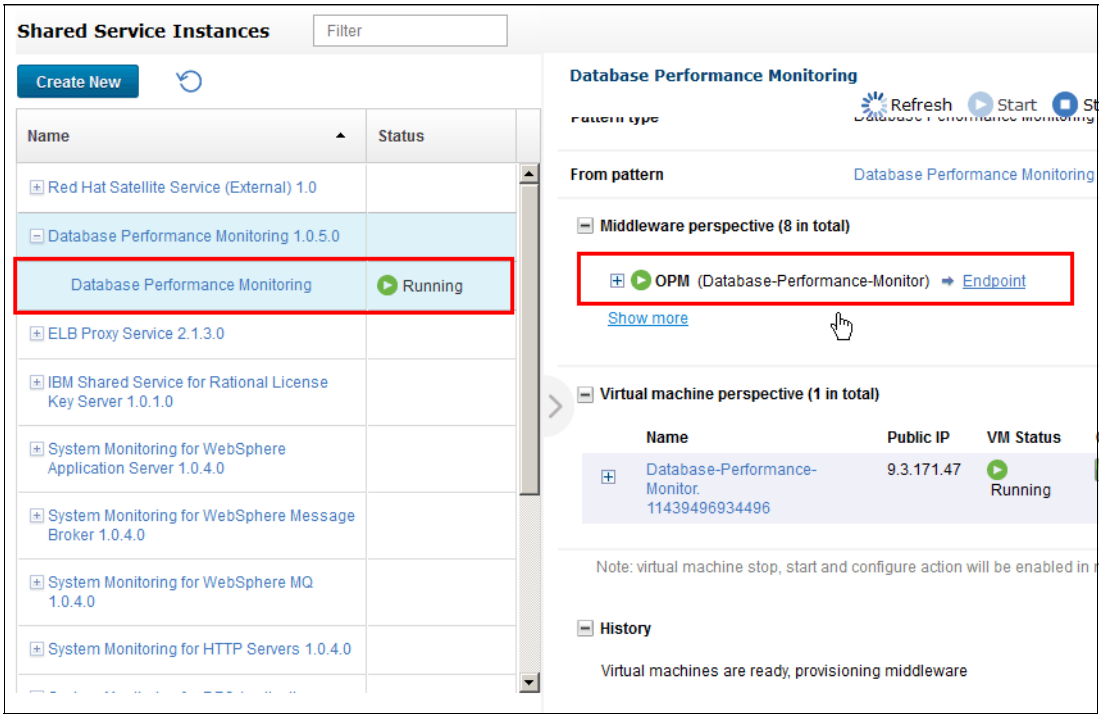


Figure 6-40 Database Performance Monitoring

4. Click the **Endpoint results** to open the Database Performance Monitor window. You can use this window to view any database instance that is deployed into the cloud group in which the shared service was deployed. From here, you can perform operations to tune a particular database, as shown in Figure 6-41.

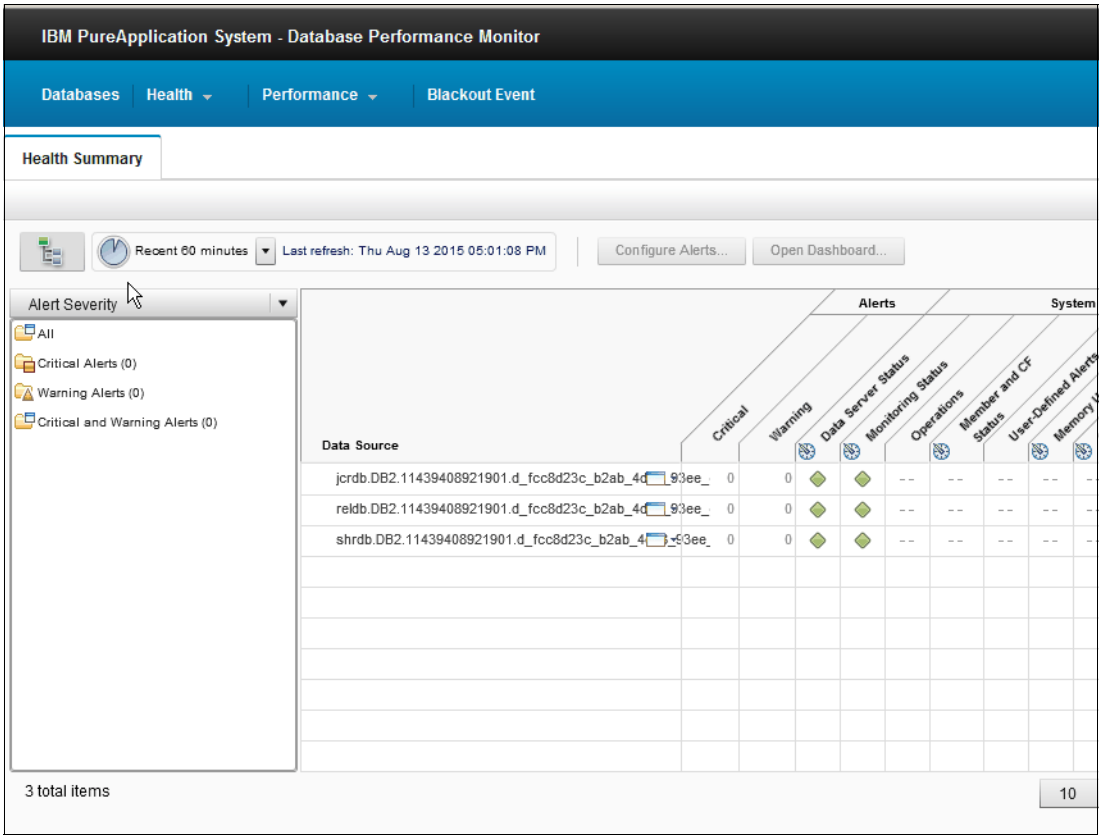


Figure 6-41 Database Performance Monitor

7

Operating system maintenance

This chapter describes how to apply software maintenance to the operating system (OS) of deployed pattern instances, which includes virtual system instances and virtual application instances, also referred to as *workloads*.

The ability to apply software maintenance to a running, production-level OS in a consistent and repeatable manner is a common and sensible requirement. Simply redeploying a newer version of a Virtual System Pattern (VSP) or a virtual application pattern is often not wanted or possible.

As an example, recall the various open source secure sockets layer (OpenSSL) vulnerabilities that recently were discovered, as described at the following website:

`https://www.openssl.org/news/vulnerabilities.html`

Most of these vulnerabilities required urgent patching, which reinforces the need to have a solution for OS maintenance in place. Updating and redeploying patterns to patch systems is both time-consuming and prone to error.

This chapter covers the following topics:

- Introduction
- OS maintenance by using Red Hat Satellite Server
- IBM AIX OS maintenance by using IBM Endpoint Manager

© Copyright IBM Corp. 2015. All rights reserved.

7.1 Introduction

When planning for OS maintenance on IBM PureApplication System or in an IBM PureApplication Software environment, you need a mechanism with the following characteristics:

- Repeatable
- Consistent
- Scalable

You need the ability to apply OS maintenance to some or all virtual system and virtual application instances, or, to be more precise, to the OS that is installed on each of the corresponding virtual machines (VMs). Having said that, these might not be the only VMs that run an OS on PureApplication System. Some shared service instances also run on one or more VMs, so OS maintenance must be applied to them.

The solution for OS maintenance depends on the OS that is used on the PureApplication platform. Table 7-1 lists the various platforms and OSs and the corresponding solutions.

Note: Most clients use a single OS across their PureApplication platform. In the case of a hybrid model, however, multiple solutions for OS maintenance might need to be implemented. A detailed description about hybrid models is outside of the scope of this book.

Table 7-1 Solutions for OS maintenance depend on the PureApplication platform and OS

Platform	OS	Solution for OS maintenance
PureApplication Software	Red Hat Enterprise Linux (RHEL) 6 or RHEL 7	Red Hat Satellite Server
PureApplication Software	Windows Server 2008 R2 or Windows Server 2012	IBM Endpoint Manager
PureApplication System models W1500/W2500 (Intel)	RHEL 6 or RHEL 7	Red Hat Satellite Server
PureApplication System models W1500/W2500 (Intel)	Windows Server 2008 or Windows Server 2012	IBM Endpoint Manager
PureApplication System models W1700/W2700 (POWER)	IBM AIX	IBM Endpoint Manager

Note: IBM also supports the use of Red Hat Update Infrastructure for OS maintenance on PureApplication System. However, both IBM and Red Hat *deprecated* Red Hat Update Infrastructure. Use Red Hat Satellite Server for OS maintenance of RHEL 6 and RHEL 7 on PureApplication System.

For a full comparison of Red Hat Update Infrastructure and Red Hat Satellite Server on PureApplication System, see the IBM Knowledge Center at the following website:

http://www.ibm.com/support/knowledgecenter/SSCR9A_2.1.0/doc/iwd/c_redhat_services_container.dita

As Table 7-1 on page 156 shows, here are the two solutions for OS maintenance on PureApplication System:

- OS maintenance by using Red Hat Satellite Server
- IBM AIX OS maintenance by using IBM Endpoint Manager

7.2 OS maintenance by using Red Hat Satellite Server

This section describes the use of Red Hat Satellite Server for OS maintenance on PureApplication System.

7.2.1 Overview

PureApplication System can automatically register the RHEL OS on deployed VMs with an existing Red Hat Satellite Server, which is done by deploying the Red Hat Satellite Service (External) shared service, which automatically configures the RHEL OS of VMs on PureApplication System to use the Red Hat Satellite Server. When you deploy this shared service, you must specify the URL of the Red Hat Satellite Server and the Activation Key to be used for registration. This key is used to register when the RHEL OS on the VMs registers with the Red Hat Satellite Server.

The following sections describe how to implement RHEL OS maintenance on PureApplication System. Here is a brief outline of the steps and the corresponding sections in this chapter:

- 7.2.2, “Preparing to deploy Red Hat Satellite Server on PureApplication System” on page 158
- 7.2.3, “Deploying Red Hat Satellite Server on PureApplication System” on page 164
- 7.2.4, “Configuring Red Hat Satellite Server on PureApplication System” on page 167
- 7.2.5, “Understanding the scope of the Red Hat Satellite Service shared service” on page 175
- 7.2.6, “Deploying the Red Hat Satellite service shared service” on page 177

Note: Some clients already have a Red Hat Satellite Server environment in place. For example, you might use it to apply OS maintenance to RHEL OS instances outside of PureApplication System. PureApplication System can directly integrate with that environment.

If this situation applies to your organization, go to 7.2.5, “Understanding the scope of the Red Hat Satellite Service shared service” on page 175.

With the Red Hat Satellite Server shared service in place, go to 7.2.7, “Using Red Hat Satellite Server integration on PureApplication System” on page 179 for a demonstration of performing OS maintenance and the general installation of Red Hat packages.

7.2.2 Preparing to deploy Red Hat Satellite Server on PureApplication System

If you do not have an existing Red Hat Satellite Server available, you can easily set up one on PureApplication System. IBM ships a VSP that is named *Red Hat Satellite Server 5.6*, which automatically installs and configures Red Hat Satellite Server 5.6. This pattern is shown in Figure 7-1. This chapter provides the deployment process for use during OS maintenance.

Note: There are multiple versions of the Red Hat Satellite Server 5.6 pattern. At the time of writing, Version 1.0.0.3 is the most recent, and it was shipped with PureApplication System V2.1.0.1 firmware.

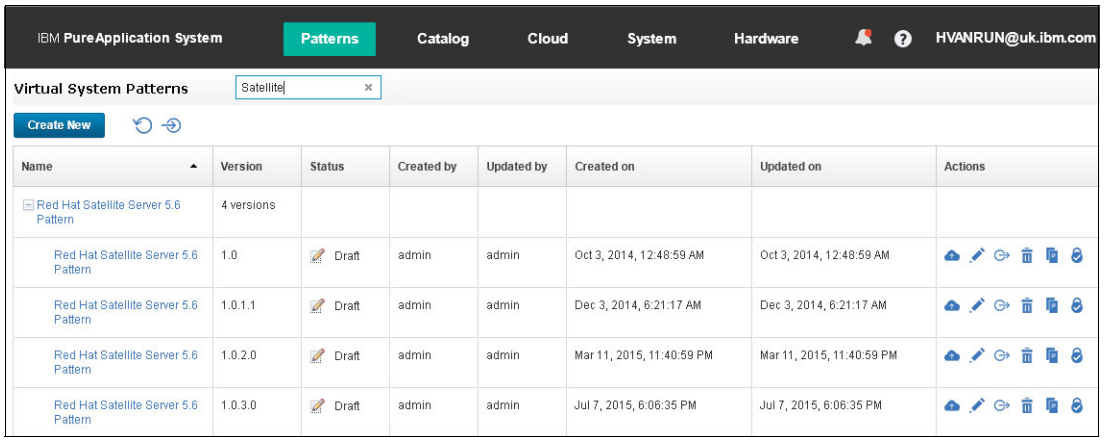


Figure 7-1 Red Hat Satellite Server 5.6 Virtual System Patterns in PureApplication System

This section describes the deployment of a VSP. Figure 7-2 shows what deployment of this pattern on PureApplication System entails.

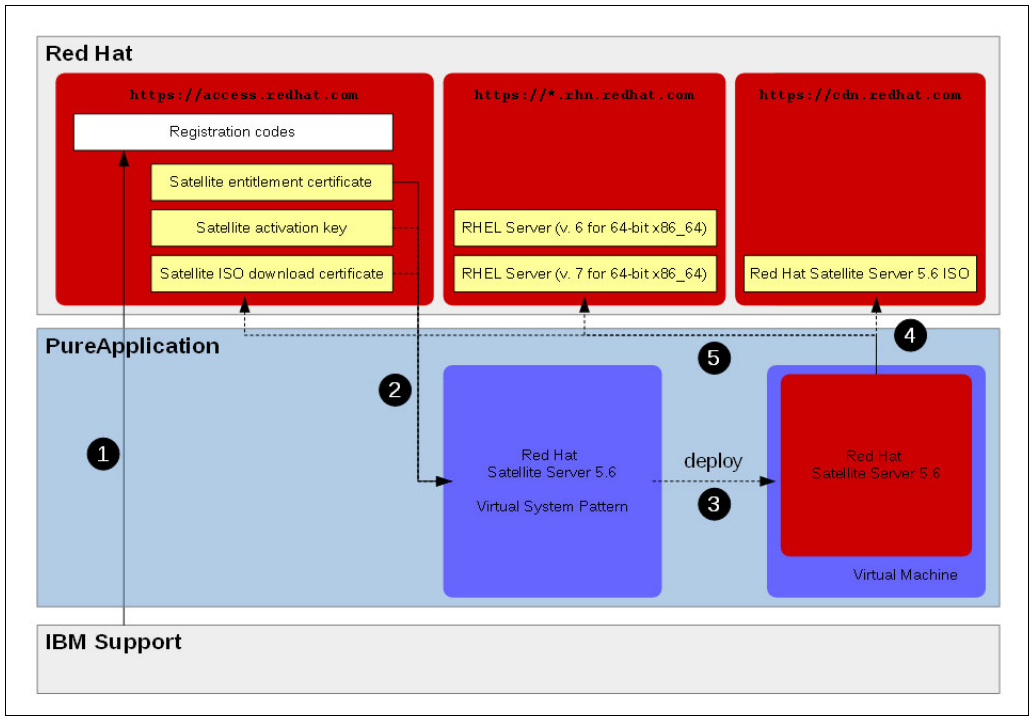


Figure 7-2 Overview of Red Hat Satellite Server 5.6 Virtual System Pattern on PureApplication System

Note: As a prerequisite, you need an active Red Hat account. To create one, go to the following website:

https://access.redhat.com

Here are the high-level steps for deploying a VSP:

1. Request and receive the Red Hat registration codes for PureApplication System from IBM Support and access your Red Hat account by providing the registration codes.
2. Obtain the following items from Red Hat to deploy the Red Hat Satellite Server 5.6 VSP on PureApplication System:
   - Satellite entitlement certificate
   - Satellite activation key
   - Satellite ISO download certificate

   Note: The Satellite ISO download certificate is a PEM file that IBM Support provides to you. You do not need to obtain this certificate from Red Hat yourself.
3. Deploy the Red Hat Satellite Server 5.6 VSP.
4. During deployment, the VSP downloads an ISO file from https://cdn.redhat.com that contains the Red Hat Satellite Server 5.6 installation binary file1s.
5. After the VSP installs Red Hat Satellite Server 5.6, the Satellite Server is registered with Red Hat. With the registration in place, all packages for the following Red Hat Software Channels are downloaded to the local Red Hat Satellite Server:
   - RHEL Server (v.6 for 64-bit x86_64)
   - RHEL Server (v.7 for 64-bit x86_64)

Note: This section focuses on the two most important aspects to deploy the Red Hat Satellite Server 5.6 VSP:

- Obtaining the Red Hat Satellite Server certificates and activation key
- Ensuring network connectivity to the internet

To see all the pre-requirements for the deployment of the Red Hat Satellite Server 5.6 VSP, go to the IBM Knowledge Center at the following website:

http://www.ibm.com/support/knowledgecenter/SSCR9A_2.1.0/doc/iwd/c_redhat_satlitesrv_reqs_2.dita

Obtaining the Red Hat Satellite Server certificates and activation key

As shown in Figure 7-2 on page 158, the deployment of the Red Hat Satellite Server 5.6 VSP requires two certificates and an activation key from Red Hat. Before you can generate these certificates and key on the Red Hat portal, you first must obtain the registration code for your PureApplication Platform from IBM by completing the following steps:

1. Open an IBM Problem Management Report (PMR) by using one of the IBM Support channels by going to the following website:

   https://www.ibm.com/support/entry/portal/support
2. Await an email from IBM Support, which contains the registration codes and detailed documentation.

To obtain the two certificates and the activation key for deployment of the Red Hat Satellite Server 5.6 VSP, complete the following steps:

1. Obtain the Red Hat Satellite activation key. The activation key is used to register the Red Hat Satellite Server 5.6 with Red Hat. To obtain the key, go to the following website:

   http://www.ibm.com/support/knowledgecenter/SSCR9A_2.1.0/doc/iwd/t_create_redhat_satlitesrv_actkey.dita

2. Obtain the Red Hat Satellite ISO download certificate. The Red Hat Satellite Server pattern downloads an ISO file containing the Red Hat Satellite Server installation binary files. This certificate is required to be able to download the file from Red Hat. To obtain the certificate, go to the following website:

   http://www.ibm.com/support/knowledgecenter/SSCR9A_2.1.0/doc/iwd/t_create_redhat_satlitesrv_dwnldcert.dita

3. Obtain the Red Hat Satellite entitlement certificate. This certificate from Red Hat contains the precise set of entitlements that are attributed to your PureApplication environment. To obtain the certificate, go to the following website:

   http://www.ibm.com/support/knowledgecenter/SSCR9A_2.1.0/doc/iwd/t_create_redhat_satlitesrv_certificate.dita

Ensuring connectivity to Red Hat

As we saw in Figure 7-2 on page 158, the deployment of the Red Hat Satellite Server 5.6 VSP requires internet connectivity to Red Hat. Figure 7-3 shows the connectivity requirements in more detail.

Figure 7-3 Internet access from Red Hat Satellite Server 5.6 on PureApplication System

Red Hat websites

The VM that is hosting the Red Hat Satellite Server on PureApplication System needs access, by using HTTPS, to the following Red Hat websites:

- https://cdn.redhat.com
- https://access.redhat.com
- https://xmlrpc.rhn.redhat.com
- https://rhn.redhat.com
- https://satellite.rhn.redhat.com
- https://content-xmlrpc.rhn.redhat.com
- https://content-web.rhn.redhat.com
- https://content-satellite.rhn.redhat.com

Note: When possible, configure access to https://*.redhat.com. If the network connectivity requirements dictated by Red Hat change over time, no additional network changes are required.

(Optional) Security First

As an option, you can install Security First. This pattern provides data-at-rest encryption. You can access it at the following website:

https://ipayum.securityfirstcorp.com

Note: Access to Security First is only applicable when you purchase and install the *IBM Encryption Pattern for Security First SPxBitFiler-IPA* from Security First. For more information, go to the following website:

http://www.ibm.com/software/brandcatalog/puresystems/centre/details?uid=PPA_D10NOLLN

Verifying connectivity

Verify the connectivity of a VM that is in the IP group to which you plan to deploy the Red Hat Satellite Server 5.6 VSP. This task ensures that the VM can access the URLs that are listed in “Red Hat websites” on page 161. Use a tool such as **`wget`** or **`curl`**.

Many clients do not allow a direct connection to the internet from their data center. Instead, a *forward proxy* must be used to connect to the internet. The deployment of the Red Hat Satellite Server 5.6 VSP supports the use of a forward proxy; however, it *must* be configured *without SSL termination* for all outbound connections to the sites listed in “Red Hat websites” on page 161.

Note: The IBM Knowledge Center mentions that a reverse proxy is not supported. However, a *forward* proxy is supported if no *SSL termination* is performed on the outbound connections. For more information, go to the following website:

http://www.ibm.com/support/knowledgecenter/SSCR9A_2.1.0/doc/iwd/c_testconn_redhat_satlitesrv.dita?lang=en

Sometimes, it is valuable to *prove* that SSL termination is turned off in your proxy server. Example 7-1 shows how you can use **curl** to accomplish this task.

Note: When performing this task, pay attention to the issuer of the SSL certificate. If it looks like that the issuer originated from your own proxy server, that means that the SSL connection was terminated.

The issuer in Example 7-1 is the correct one. Akamai, not Red Hat, is the issuer, confirming that Red Hat uses Akamai's Content Distribution Network for https://access.redhat.com. Here is the line of interest:

```
issuer: C=NL; L=Amsterdam; O=Verizon Enterprise Solutions; OU=Cybertrust;
CN=Verizon Akamai SureServer CA G14-SHA1
```

Example 7-1 Using curl to access https://access.redhat.com through a proxy server

```
[virtuser@pure-9-3-172-231 ~]$ curl -v -k -I -x proxy.emea.ibm.com:80
https://access.redhat.com/
* About to connect() to proxy proxy.emea.ibm.com port 80 (#0)
*   Trying 9.139.247.81... connected
* Connected to proxy.emea.ibm.com (9.139.247.81) port 80 (#0)
* Establish HTTP proxy tunnel to access.redhat.com:443
> CONNECT access.redhat.com:443 HTTP/1.1
> Host: access.redhat.com:443
> User-Agent: curl/7.21.3 (x86_64-unknown-linux-gnu) libcurl/7.21.3 OpenSSL/1.0.1e
zlib/1.2.3
> Proxy-Connection: Keep-Alive
>
< HTTP/1.1 200 Connection established
HTTP/1.1 200 Connection established
<

* Proxy replied OK to CONNECT request
* successfully set certificate verify locations:
*   CAfile: /etc/pki/tls/certs/ca-bundle.crt
  CApath: none
* SSLv3, TLS handshake, Client hello (1):
* SSLv3, TLS handshake, Server hello (2):
* SSLv3, TLS handshake, CERT (11):
* SSLv3, TLS handshake, Server finished (14):
* SSLv3, TLS handshake, Client key exchange (16):
* SSLv3, TLS change cipher, Client hello (1):
* SSLv3, TLS handshake, Finished (20):
* SSLv3, TLS change cipher, Client hello (1):
* SSLv3, TLS handshake, Finished (20):
* SSL connection using AES256-SHA
* Server certificate:
*        subject: C=US; ST=NC; L=Raleigh; O=Red Hat, Inc.; OU=Web Operations;
CN=access.redhat.com
*        start date: 2014-10-17 20:30:01 GMT
*        expire date: 2015-10-17 20:29:57 GMT
*        subjectAltName: access.redhat.com matched
*       issuer: C=NL; L=Amsterdam; O=Verizon Enterprise Solutions; OU=Cybertrust;
CN=Verizon Akamai SureServer CA G14-SHA1
*        SSL certificate verify ok.
> HEAD / HTTP/1.1
```

```
> User-Agent: curl/7.21.3 (x86_64-unknown-linux-gnu) libcurl/7.21.3 OpenSSL/1.0.1e
zlib/1.2.3
> Host: access.redhat.com
> Accept: */*
>
< HTTP/1.1 200 OK
HTTP/1.1 200 OK
< Server: Apache
Server: Apache
< X-Powered-By: PHP/5.3.3
X-Powered-By: PHP/5.3.3
< Content-Language: en
Content-Language: en
< X-Cnection: close
X-Cnection: close
< Content-Type: text/html; charset=UTF-8
Content-Type: text/html; charset=UTF-8
< Cache-Control: no-cache, must-revalidate, post-check=0, pre-check=0
Cache-Control: no-cache, must-revalidate, post-check=0, pre-check=0
< Expires: Tue, 21 Jul 2015 14:22:26 GMT
Expires: Tue, 21 Jul 2015 14:22:26 GMT
< Date: Tue, 21 Jul 2015 14:22:26 GMT
Date: Tue, 21 Jul 2015 14:22:26 GMT
< Connection: keep-alive
Connection: keep-alive
* no chunk, no close, no size. Assume close to signal end

<
* Closing connection #0
* SSLv3, TLS alert, Client hello (1):
```

7.2.3 Deploying Red Hat Satellite Server on PureApplication System

With all the pre-requirements in place, deploying the VSP is relatively straightforward. To do so, complete the following steps:

1. Log in to the PureApplication System web console and click **Patterns** → **virtual system patterns**.
2. Find **Red Hat Satellite Server 5.6 Pattern version 1.0.0.3** and click the deployment icon. The Deploy Pattern window opens, as shown in Figure 7-4.

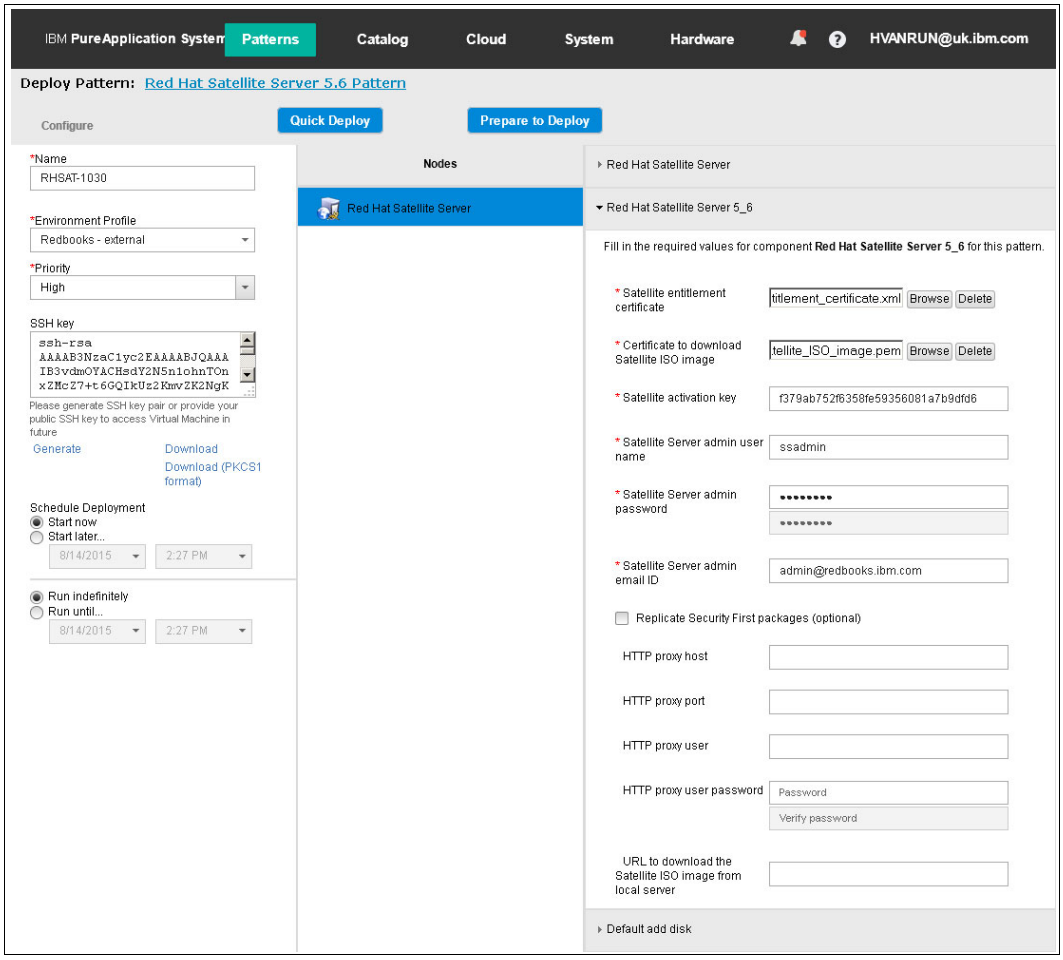


Figure 7-4 The Deploy Pattern window for deploying the Red Hat Satellite Server 5.6 pattern

3. Complete the Name and Priority fields in the Configure window.
4. Make a selection for the **Environment Profile** field. You must assign an IP group to the first *data* Network Interface Card (NIC). Addresses from this IP group must have connectivity to the internet, as described in “Ensuring connectivity to Red Hat” on page 160.

Note: The first data NIC corresponds to eth1 for VMs that are deployed on PureApplication System, but to eth0 for VMs on PureApplication Software.

Note: Take extra care if your environment profile contains more than just that one data IP group:

- When using an *internally managed* environment profile, you can immediately specify the IP group that is used in the Configure window.
- When using an *externally managed* environment profile, you must proceed to the Prepare to Deploy window and assign the correct IP group from your environment profile.

5. Enter the following information into the Deploy Pattern window for the Red Hat Satellite Server 6_6 software component of the Red Hat Satellite Server node:
   - Satellite entitlement certificate

     The Red Hat Satellite Server entitlement certificate that you obtained from Red Hat, ad described in “Obtaining the Red Hat Satellite Server certificates and activation key” on page 159. It is an XML file. For more information, see the following website:

     https://access.redhat.com/documentation/en-US/Red_Hat_Network_Satellite/5.2/html/Installation_Guide/ch-entitlements.html
   - Certificate to download Satellite ISO image

     This is the Red Hat Satellite Server ISO download certificate that you obtained from Red Hat, as described in “Obtaining the Red Hat Satellite Server certificates and activation key” on page 159. It is an XML file.
   - Satellite activation key

     This is the Red Hat Satellite Server activation key that you obtained from Red Hat, as described in “Obtaining the Red Hat Satellite Server certificates and activation key” on page 159. It is an XML file.
   - Satellite Server admin user name

     Enter your Satellite Server admin user name.
   - Satellite Server admin password

     Enter your Satellite Server admin password.
   - Satellite Server admin email ID

     Enter your Satellite Server admin email ID.
6. In the right pane, specify a password for the root and virtuser users.

7. Click **Prepare to Deploy** and ensure that the correct data IP group is assigned, as shown in Figure 7-5.

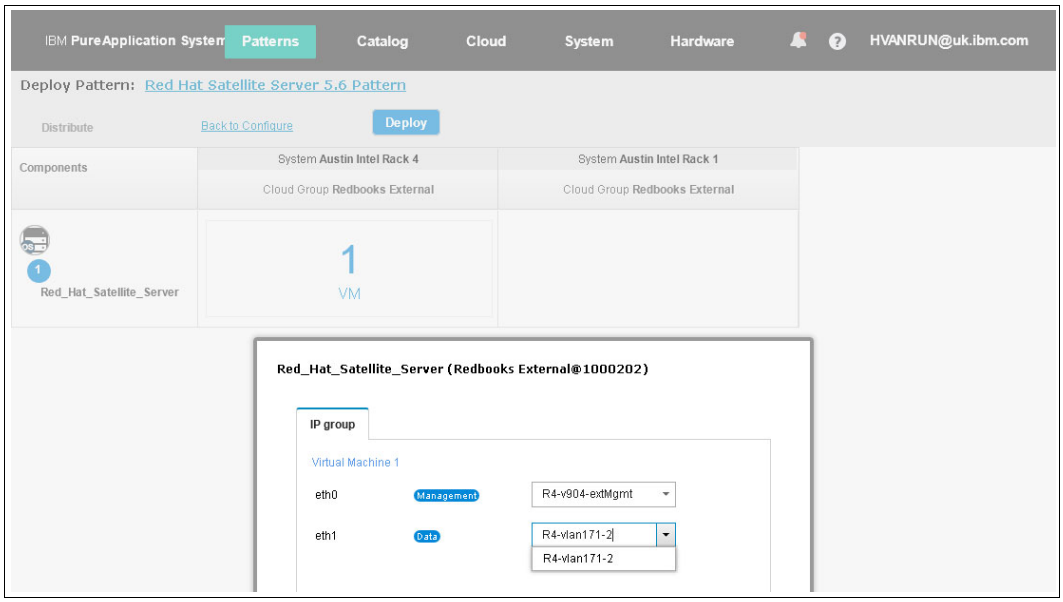


Figure 7-5 Select the correct data IP group to be used at deployment time

8. If you are using an *internally managed environment profile*, you can directly proceed with deployment by clicking **Quick Deploy**. If you are using an *externally managed environment profile*, select one of the IP groups that are defined in your profile.

Deployment of the VSP depends on the speed of your internet connection. Figure 7-6 shows the History of a successfully deployed Virtual System Instance. As you can see, deployment took over 20 hours to complete.

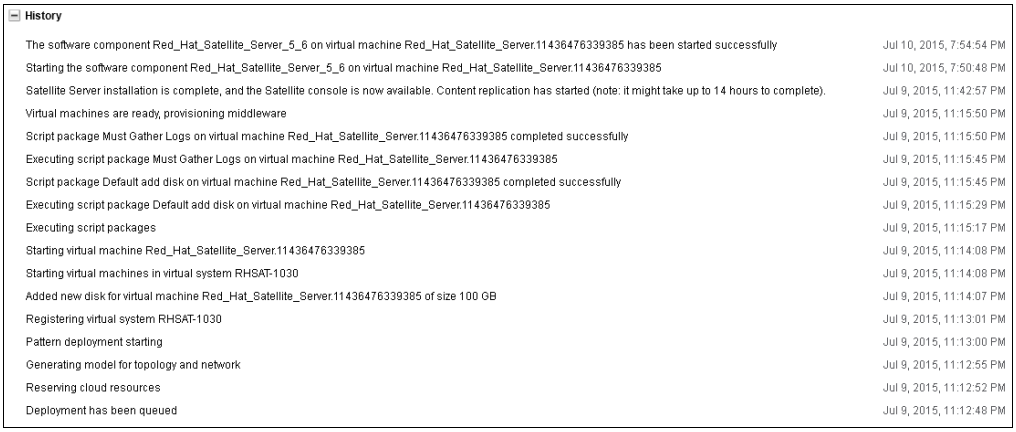


Figure 7-6 Sample of a successfully deployed Virtual System instance

With the process complete, the deployed Virtual System instance looks like the window that is shown in Figure 7-7 on page 167.

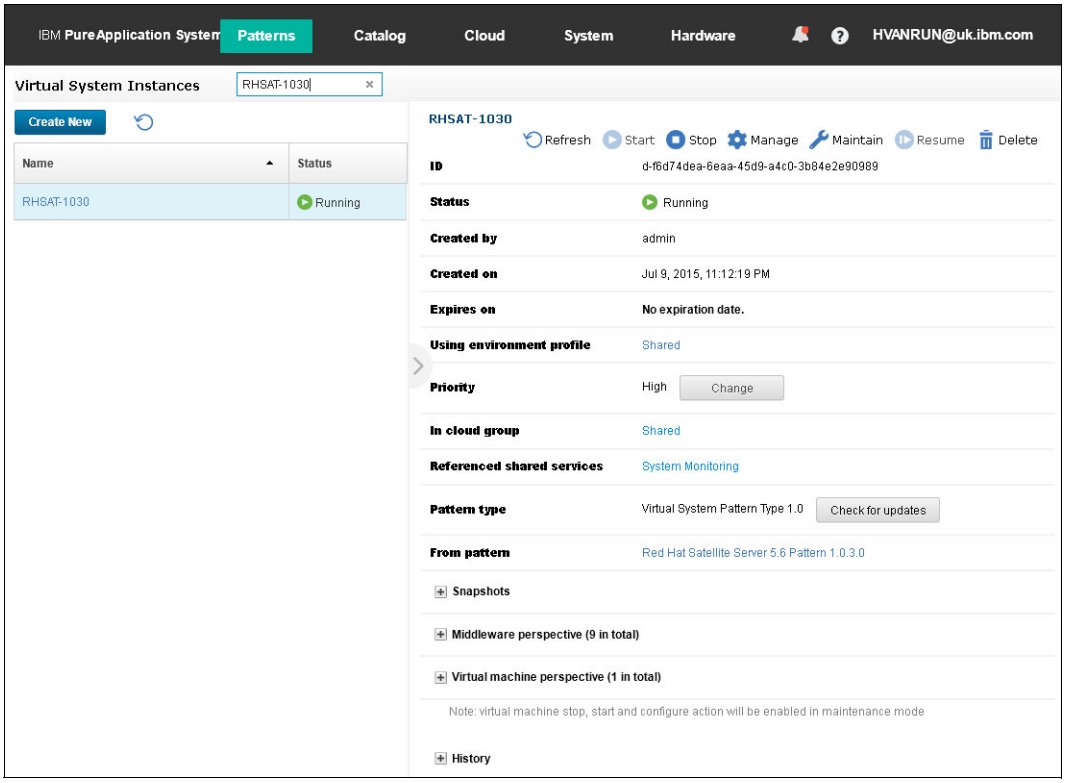


Figure 7-7 Deployed and running Red Hat Satellite Server Virtual System instance

7.2.4 Configuring Red Hat Satellite Server on PureApplication System

The Red Hat Satellite Service shared service automatically registers the RHEL OS of all VMs that are within the scope of the shared service with a Red Hat Satellite Server, as described in 7.2.6, “Deploying the Red Hat Satellite service shared service” on page 177. During the registration, a specific activation key that is defined in the shared service is used. So, you must configure at least one such activation key in the Red Hat Satellite Server you deployed.

Creating the activation key and a system group

The first task is to create an activation key. This key is associated with the Systems Group, which is also created in this section. Complete the following steps:

1. Log in to the Red Hat Satellite Server GUI, shown in Figure 7-8, by using the credentials that you specified during the deployment of the VSP (described in 7.2.3, “Deploying Red Hat Satellite Server on PureApplication System” on page 164).

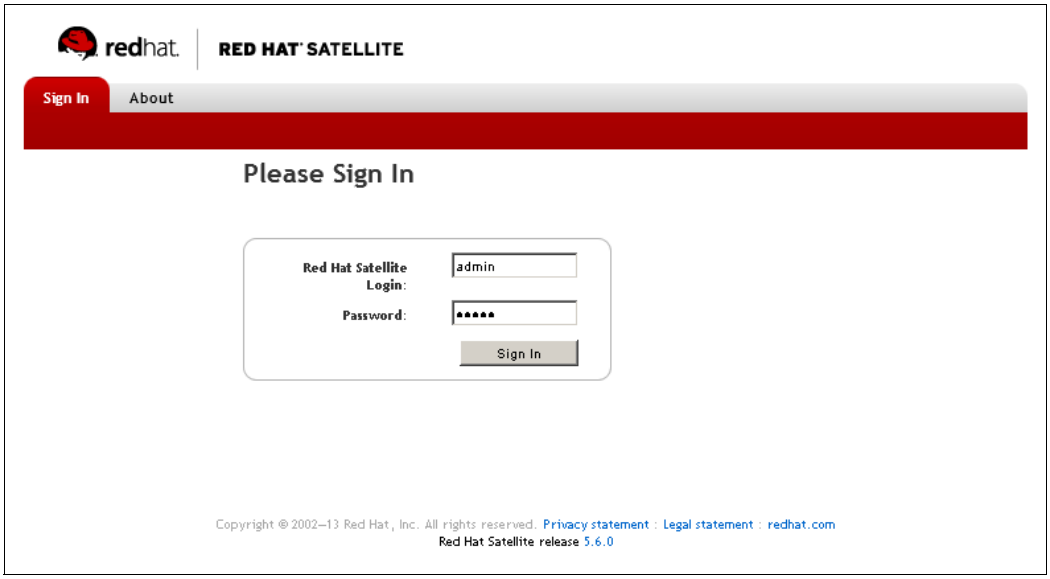


Figure 7-8 Log in to the Red Hat Satellite Server GUI

2. Click **System** → **Activation Keys**, as shown in Figure 7-9, and click **create new key** in the upper right corner.

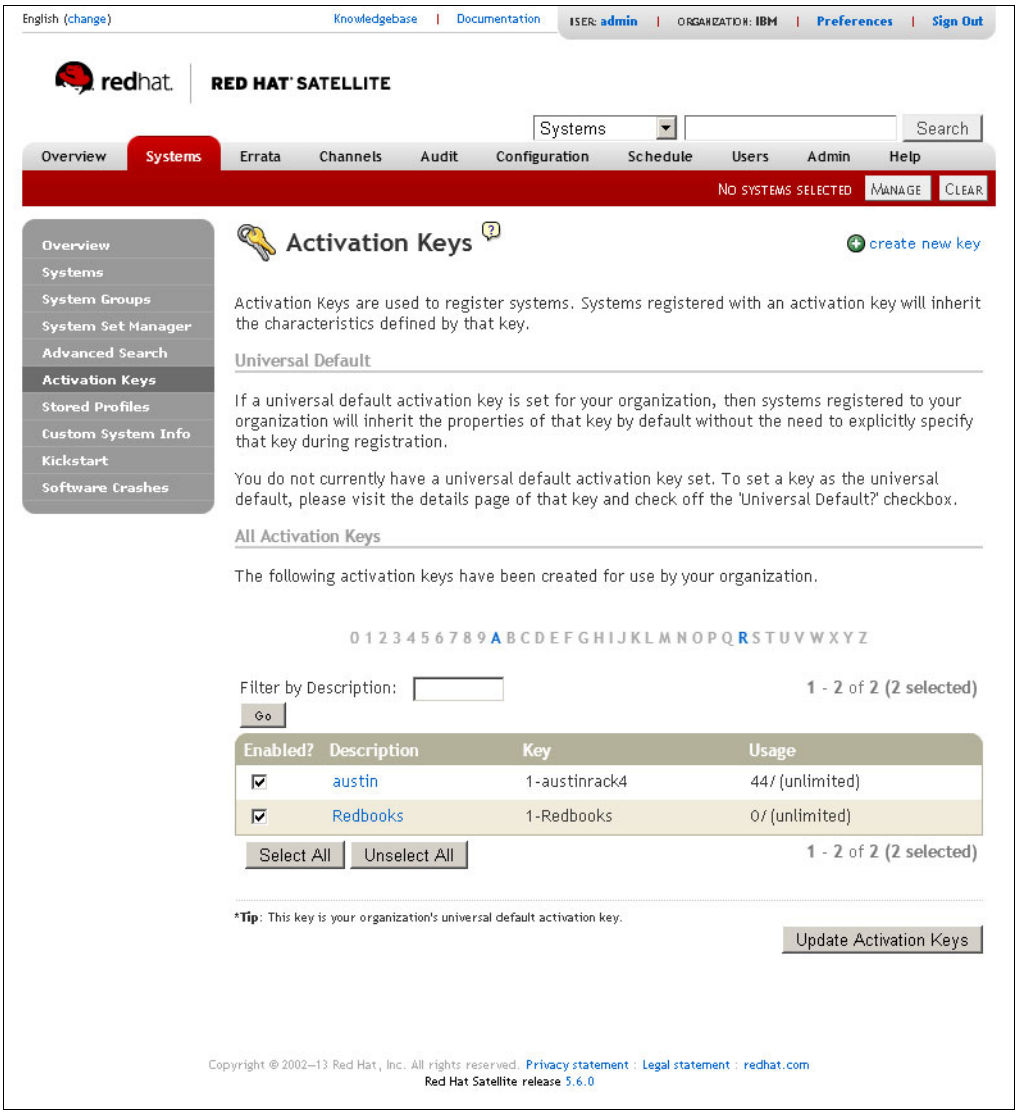


Figure 7-9 Create an activation key in the Red Hat Satellite Server GUI

3. Enter a **Key** and **Description**, as shown Figure 7-10. Ensure that the Red Hat Satellite Default channel is selected.

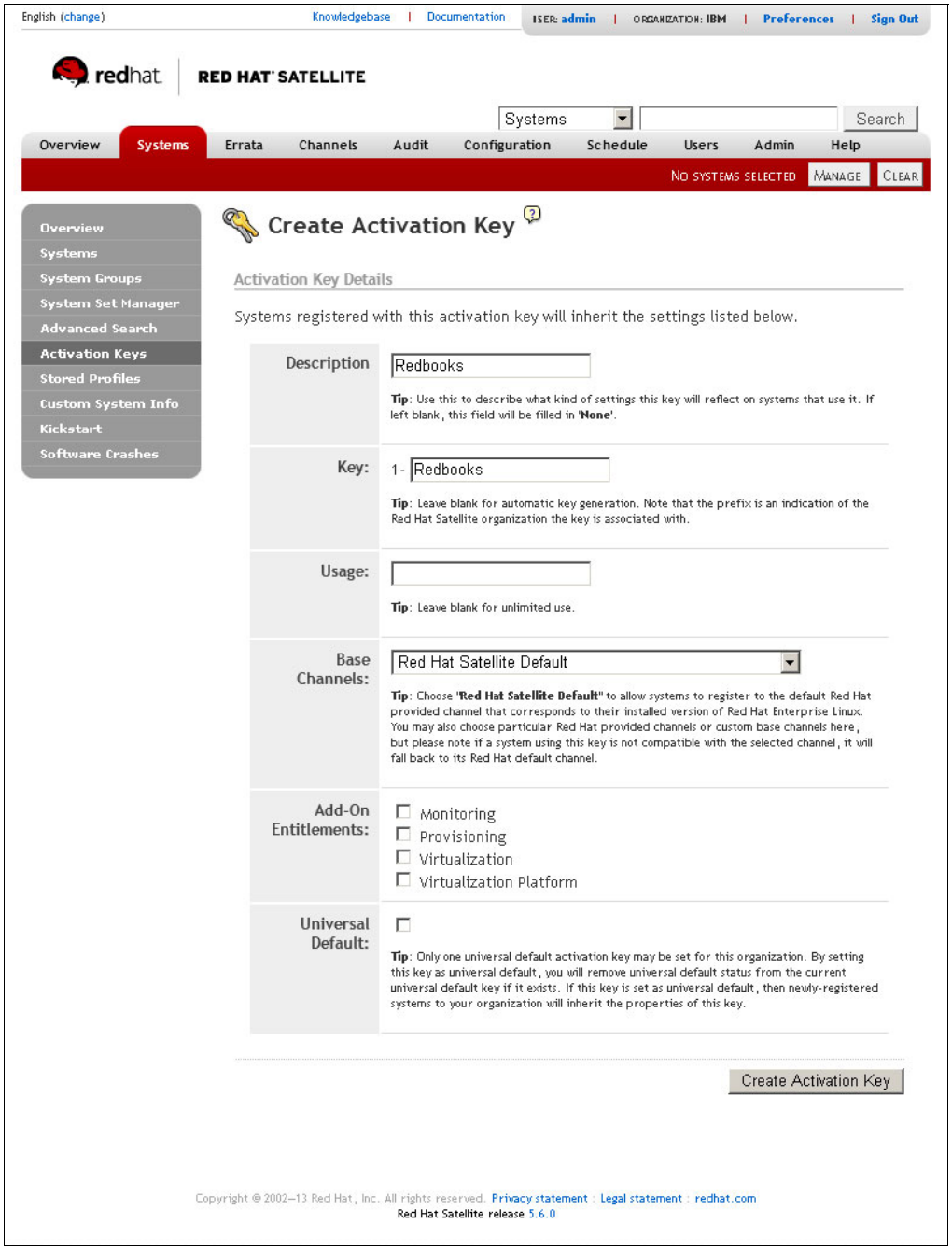


Figure 7-10 Provide information for the new activation key

4. Click **Create Activation Key** to create the key.
5. Associate the activation key with the software channels. Click **Child Channels**, and *at a minimum*, ensure that the following two child channels are selected:
   - RHN Tools for RHEL (v.6 for 64-bit x86_64)
   - RHN Tools for RHEL (v.7 for 64-bit x86_64)

As shown in Figure 7-11, for this example, we also selected the child channel Security First kernel RPMS for RHEL (v.6 for 64-bit x86_64).

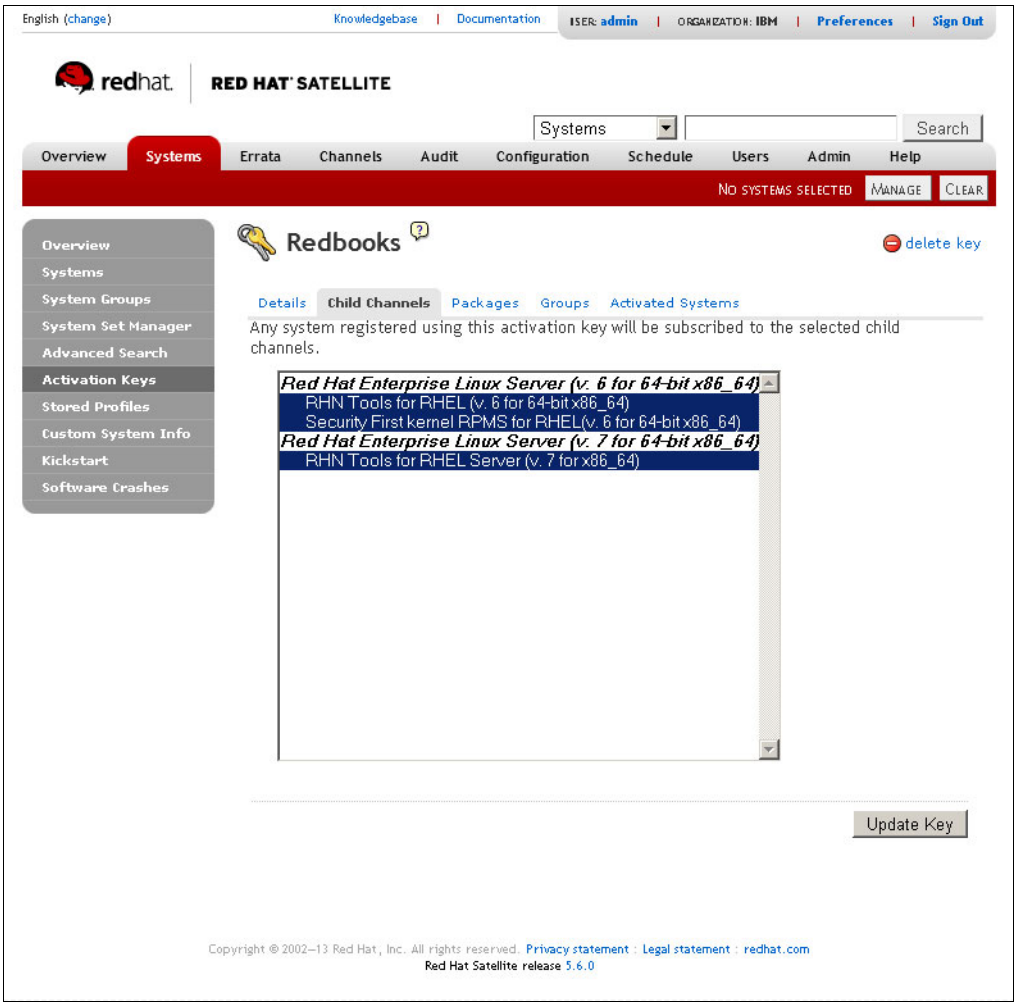


Figure 7-11 Configure an additional child channel for the newly created activation key

6. Click **Update Key** to apply these changes.

Note: Without the child channels for RHN Tools for RHEL, PureApplication System cannot automatically install and configure `osad`. This daemon is required to push out package updates. For more information, see “Applying updates to deployed VMs from the Red Hat Satellite GUI” on page 184.

With the new activation key created, you can now configure a new Systems Group and associate it with the new activation key.

Note: This task is not strictly necessary, but it can greatly simplify managing a large number of VMs that are all registered with the Red Hat Satellite Server by using the same Activation Key.

With all the VMs conveniently grouped, rolling out updates, for example, to packages, can be applied to all VMs at one time. For more information about this topic, see “Applying updates to deployed VMs from the Red Hat Satellite GUI” on page 184.

7. Click **Systems** → **System groups** and click **create new group**, as shown in Figure 7-12.

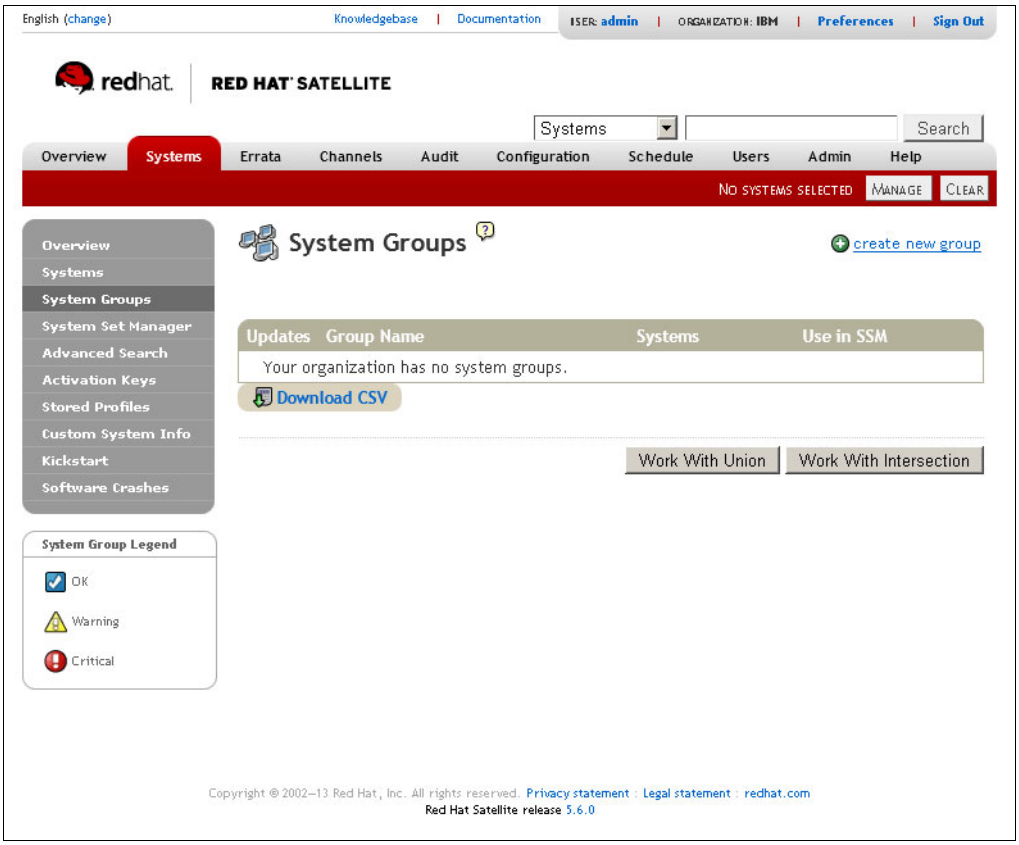


Figure 7-12 Create a System Group

8. Provide a **Name** and **Description** for the new System Group, as shown in Figure 7-13 on page 173, and click **Create Group**.

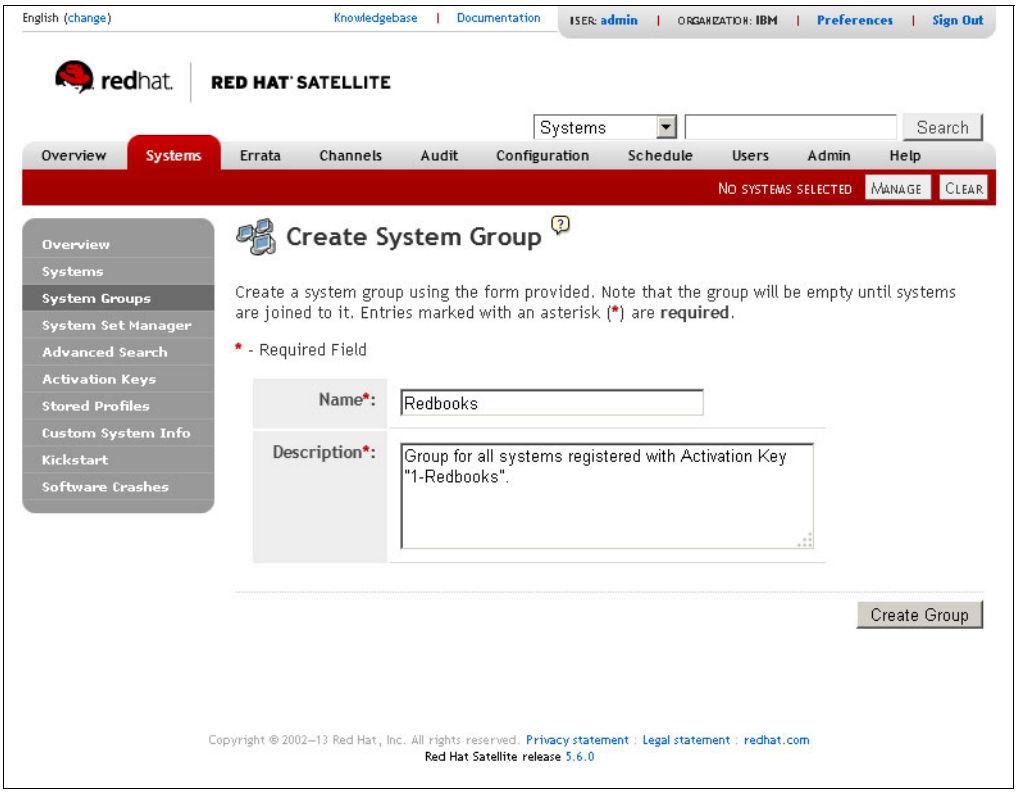


Figure 7-13 Provide a Name and Description for the new System Group

9. With the new System Group created, click **Systems** → **Activation Keys** and select the Activation Key *Redbooks* that was created in step 4 on page 170.

10. In the pane on the right, click **Groups** and click **Join**, as shown in Figure 7-14.

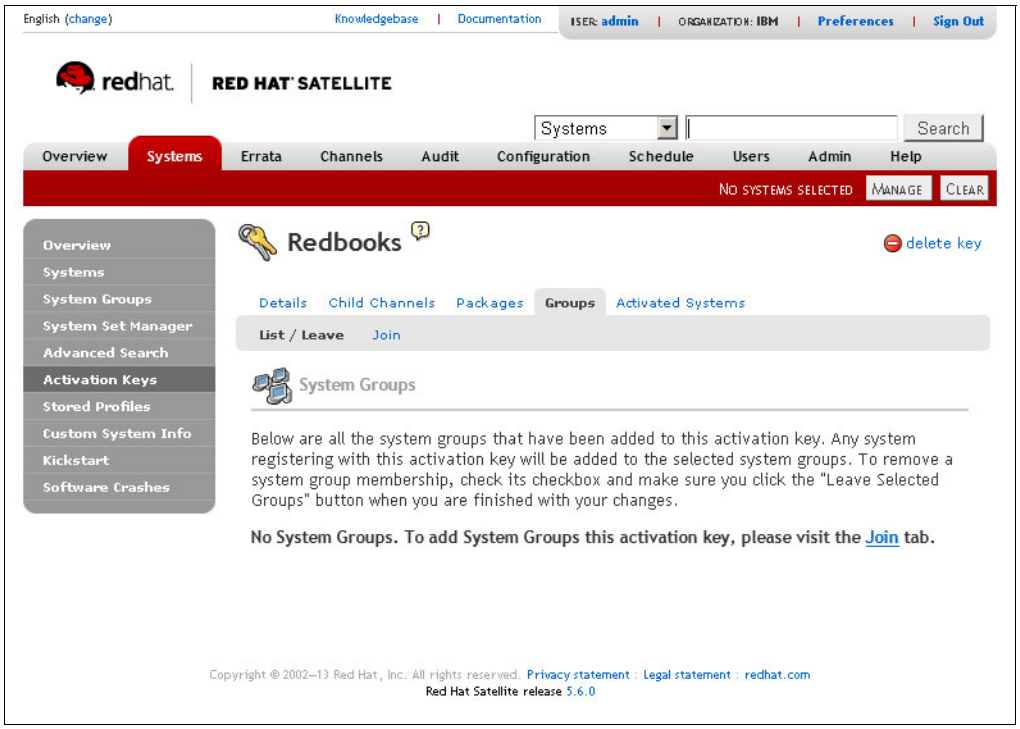


Figure 7-14 System Groups for the activation key Redbooks

11. Now, select the System Group *Redbooks* and click **Join Selected Groups**, as shown in Figure 7-15 on page 175.

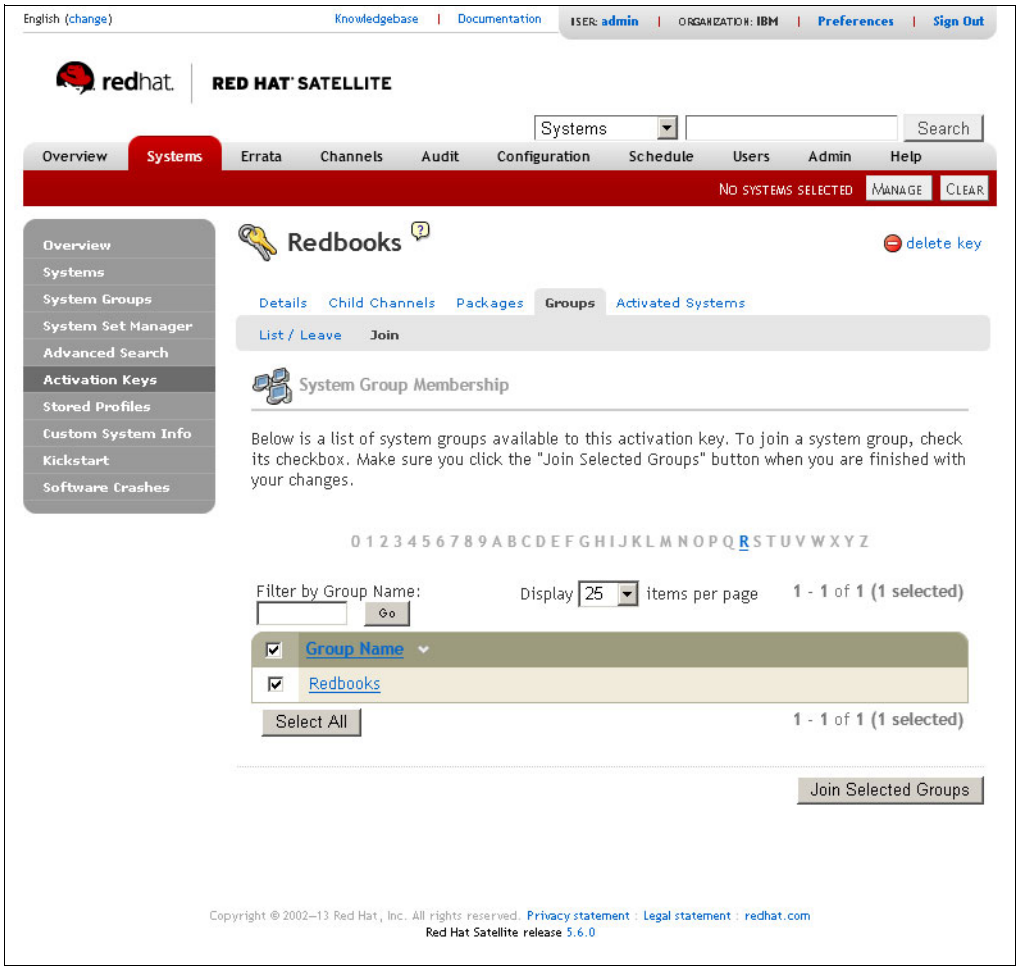


Figure 7-15 Activation key Redbooks joining the System Group Redbooks

This completes the association of the activation key *Redbooks* with System Group *Redbooks*. Any RHEL OS on a VM that automatically registers by using this key automatically joins the System Group *Redbooks*.

Note: Although a single activation key is required, there are many scenarios where multiple activation keys can help. Each activation key can, in turn, be associated with its own System Group. For more information, see Figure 7-16 on page 176.

7.2.5 Understanding the scope of the Red Hat Satellite Service shared service

When deploying a shared service on PureApplication System, consider the *scope*, that is, consider the set of VMs with which the shared service instance interacts.

Historically, the scope of a shared service is the cloud group where the shared service is deployed. All VMs that run inside that cloud group are considered in scope. However, since the release of PureApplication System 2.0, the scope is no longer always the cloud group. Instead, when using an externally managed environment profile to deploy a shared service, the scope of that shared service is the environment profile. Only VMs deployed by using that same environment profile are now in scope of the shared service.

Table 7-2 shows how the scope of a shared service varies depending on the type of environment profile that is used.

Table 7-2 The scope of shared services is determined by the type of environment profile

Environment profile	Cloud management of cloud group	Scope
Internally managed	By way of internal network	Cloud group
Externally managed	By way of external network	Environment profile

When applied to the Red Hat Satellite Service (external) shared service, the scope is important. By deploying multiple shared service instances where each uses a unique key, the VMs on PureApplication System can easily be grouped in the Red Hat Satellite Server. For example, you might want to have separate groups for production and non-production VMs, as shown in Figure 7-16.

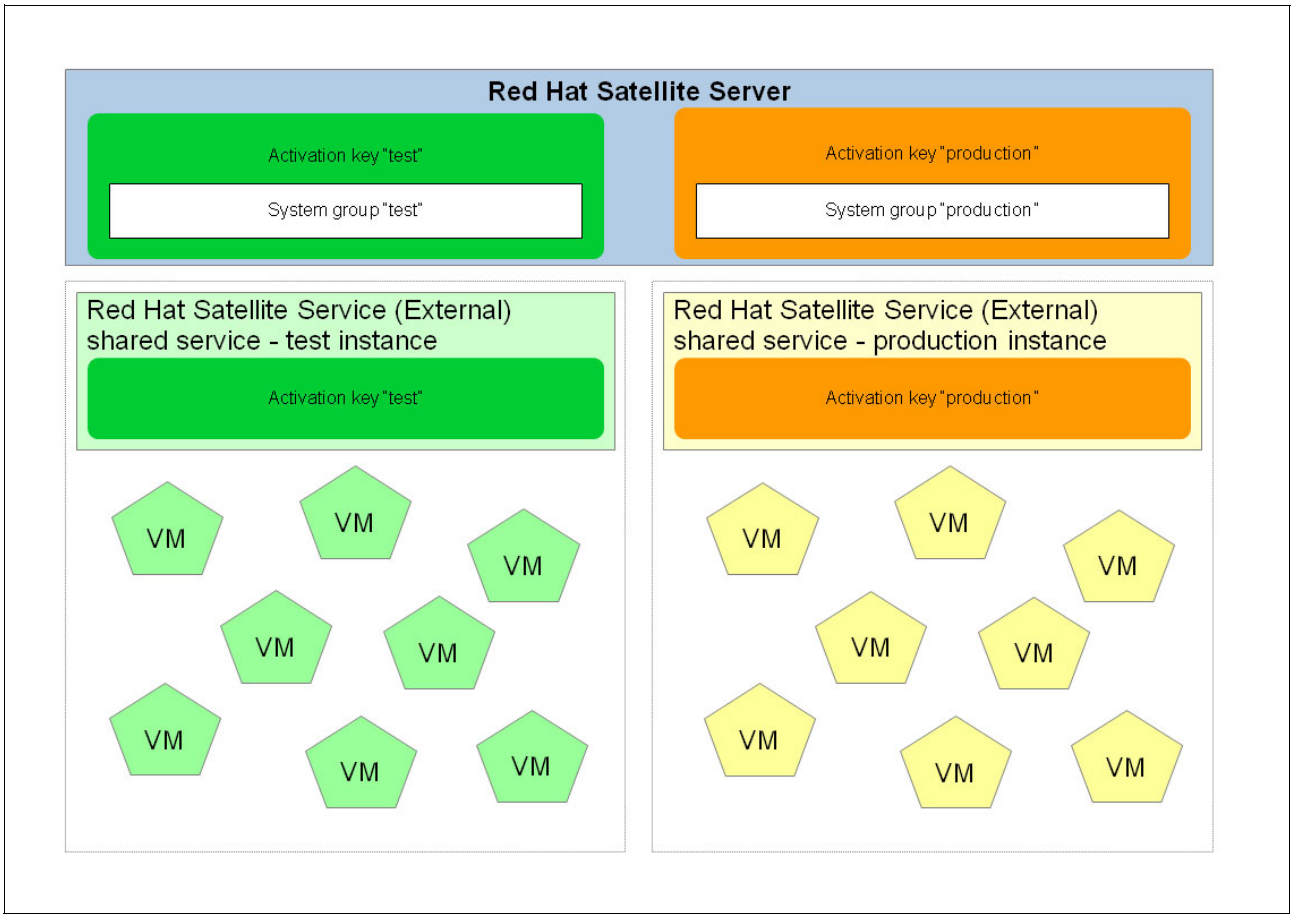


Figure 7-16 Using two shared service instances each with a unique activation key

7.2.6 Deploying the Red Hat Satellite service shared service

Before you deploy the Red Hat Satellite Server (external) shared service, ensure that you have the following items in place:

- The URL for the Red Hat Satellite Server
- At least one activation key that is defined in your existing Red Hat Satellite Server
- *(Optional)* HTTP proxy server details for accessing your Red Hat Satellite Server

You are now ready to deploy the Red Hat Satellite Service (external) shared service. Complete the following steps:

1. From the console, click **Patterns** → **Pattern instances** → **Shared service** and click **Create New**, as shown in Figure 7-17.

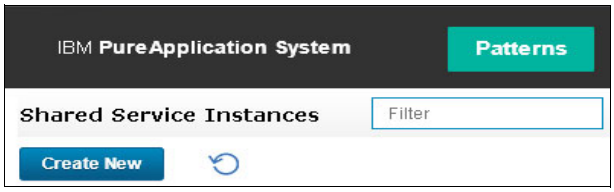


Figure 7-17 Create a shared service (external) instance

2. Locate the shared service **Red Hat Satellite Service (External)** in the list and **click OK**, as shown in Figure 7-18.

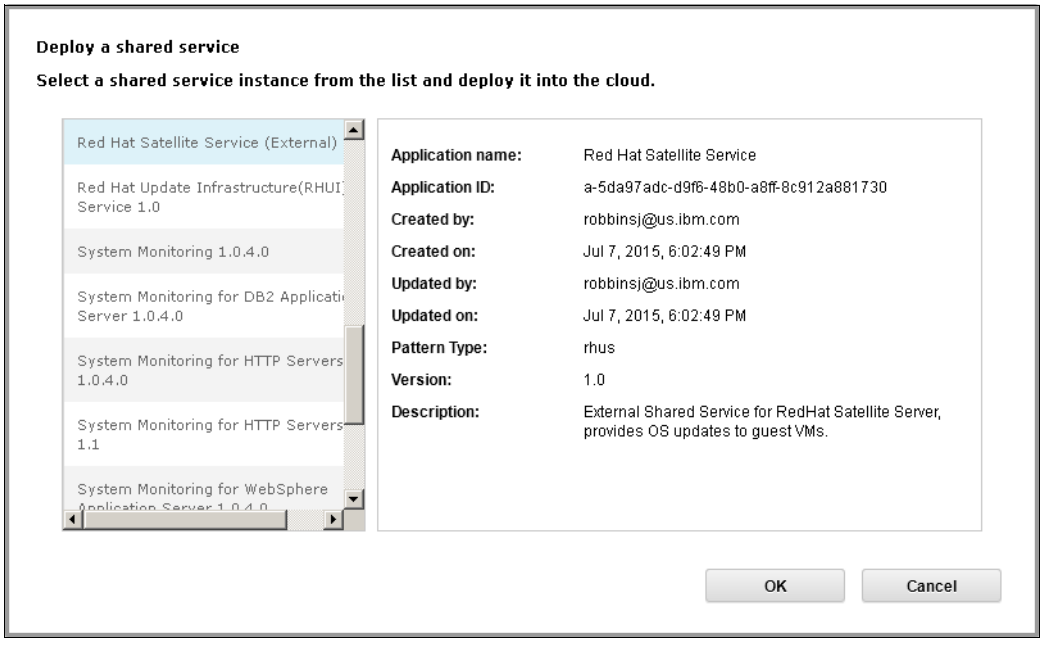


Figure 7-18 Select the Red Hat Satellite Service (external) shared service for deployment

3. In the deployment pane, complete the following steps:
   a. Provide a Name for the shared service instance.
   b. Choose the environment profile that is used for deployment. In this example, we opted to deploy to an externally managed environment profile that is named *Redbooks - external*.

Note: Because this is an external shared service, no VMs are deployed. Hence, there is no need to provide an SSH (public) key.

c. Enter the URL for your Red Hat Satellite Server into the **URL to connect to Satellite Server host** field.

Note: You must provide either the IP address or the Fully Qualified Domain Name (FQDN) of the host.

d. Enter the Activation Key into the **Satellite Server activation key** field. In our example, we set the activation key to match what we defined in the Red Hat Satellite Server (*1-Redbooks*).

e. *(Optional)* Enter the details for your HTTP proxy server.

f. Click **Quick Deploy** to prepare the deployment, as shown in Figure 7-19. The **Prepare to Deploy** button is not valid here because this is an external shared service, and no actual VMs are deployed.

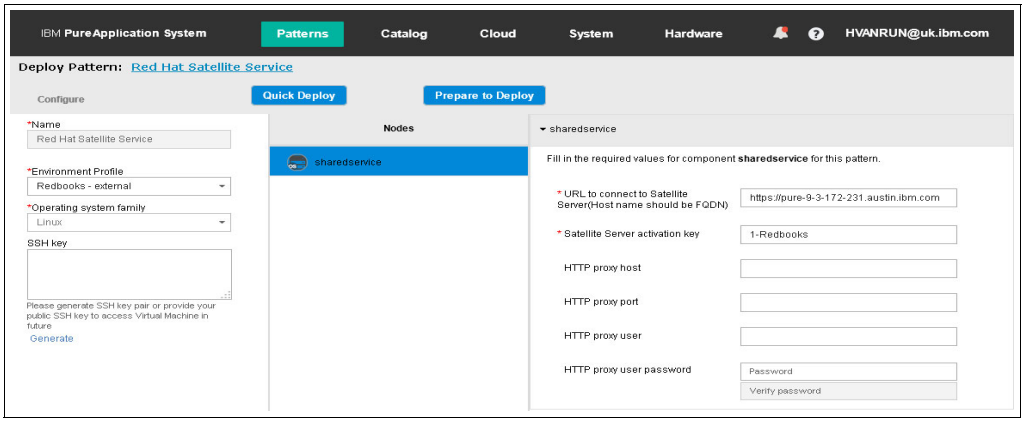


Figure 7-19 Deploying the Red Hat Satellite Service (external) shared service

4. When the deployment is complete, the shared service instance can be viewed by clicking **Patterns** → **Pattern instances** → **Shared services**, as shown in Figure 7-20.

Note: The shared service that we deployed in our example is the first one in the list. No cloud group is mentioned for this shared service instance because we deployed it by using an *externally managed* environment profile.

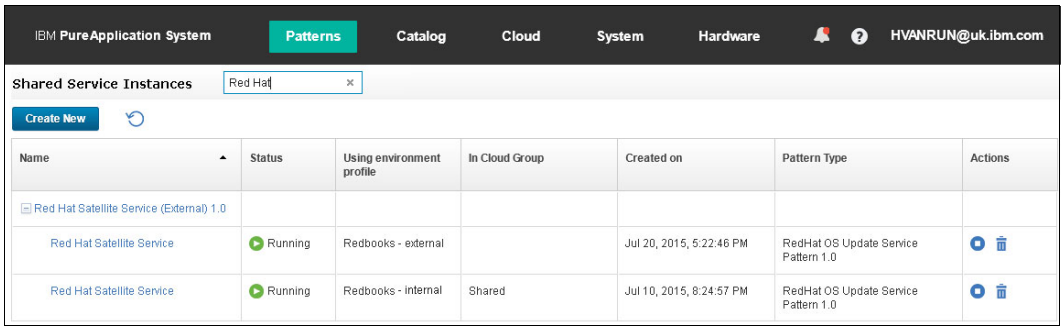


Figure 7-20 Deployed Red Hat Satellite Service (external) shared service instance

Note: If you want to delete and re-deploy the Red Hat Satellite Server shared service, then the OS of all VMs in scope automatically re-register themselves with the Red Hat Satellite Server that is specified in the newly deployed shared service instance.

7.2.7 Using Red Hat Satellite Server integration on PureApplication System

With the Red Hat Satellite Server shared service instance in place, each RHEL OS that is deployed in the scope of that shared service automatically registers with the Red Hat Satellite Server. Here are a few key scenarios that are enabled by the registration with the Red Hat Satellite Server:

- "Using yum to install packages from a deployed VM" on page 182
- "Applying updates to deployed VMs from the Red Hat Satellite GUI" on page 184
- "Installing and updating packages at Virtual System Pattern deployment time" on page 191

Before you can get started, you first must deploy a simple VSP, as described in "Deploying a simple Virtual System Pattern on PureApplication System" on page 179.

Deploying a simple Virtual System Pattern on PureApplication System

In this example, you create and deploy a simple VSP that consists of two VMs running Red Hat Enterprise Linux 6.6. Here is a summary of the steps:

1. Create a VSP that is called *Multisystem core OS*, as shown in Figure 7-21. The pattern consists of two VMs, each running the IBM Virtual Image for Red Hat Linux version 2.1.2.0 (which corresponds to RHEL 6.6).

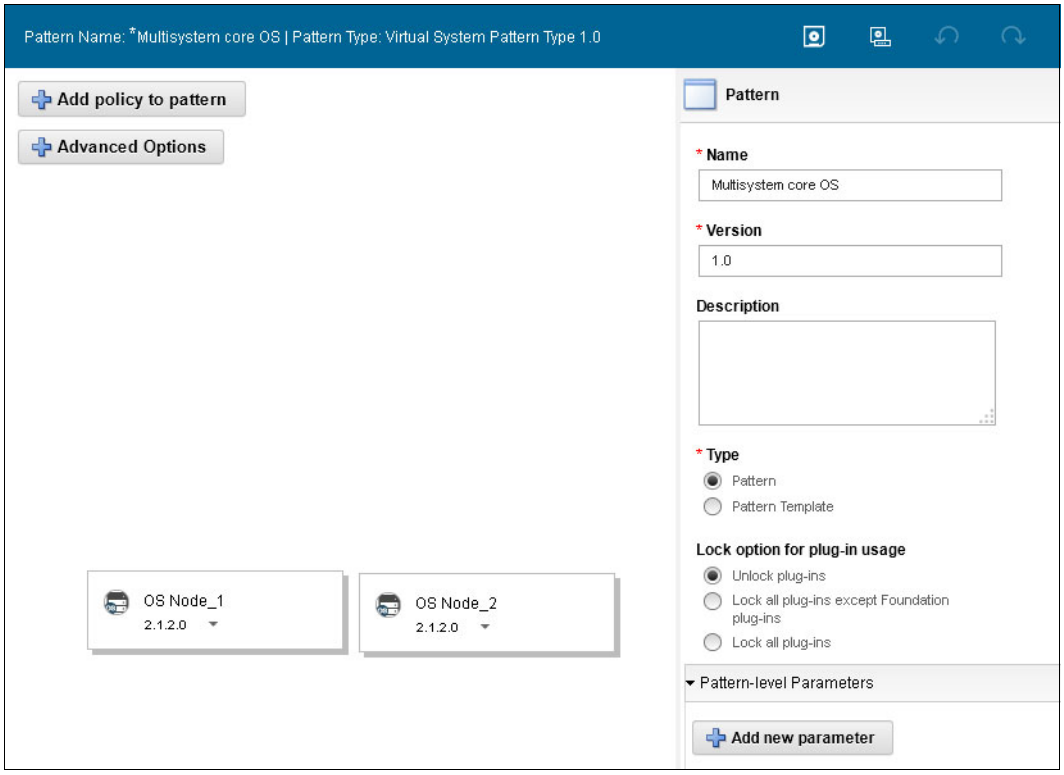


Figure 7-21 Simple Virtual System Pattern consisting of two VMs running RHEL 6.6

2. Deploy this pattern by using the environment profile *Redbooks - external*, as shown in Figure 7-22. Click **Prepare to Deploy**.

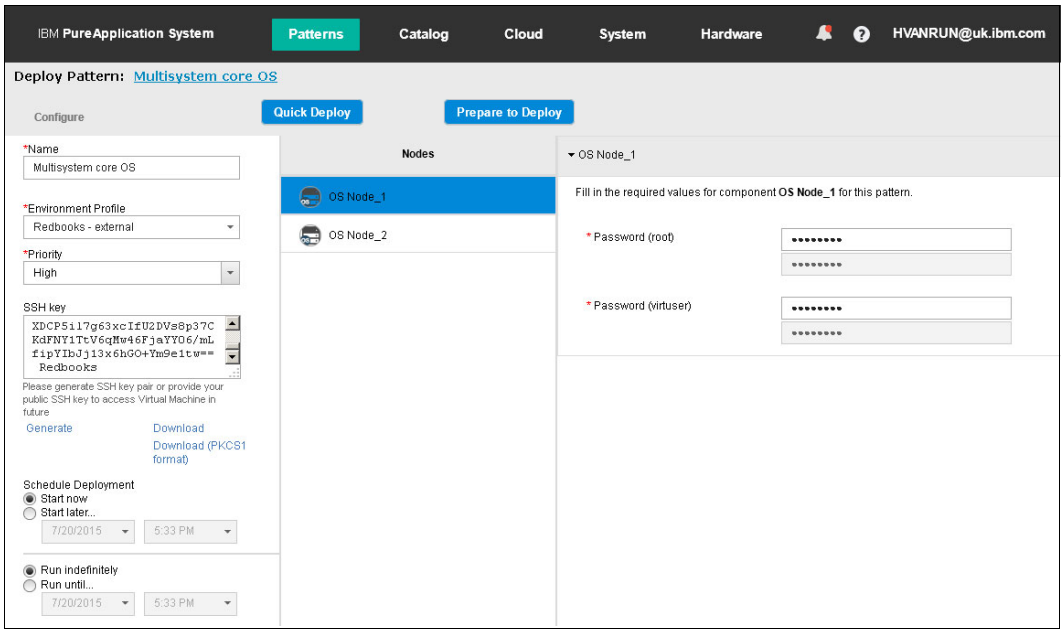


Figure 7-22 Deploy the Virtual System Pattern by using environment profile Redbooks - external

3. Assign each VM to a different PureApplication System within the deployment subdomain, as shown in Figure 7-23.

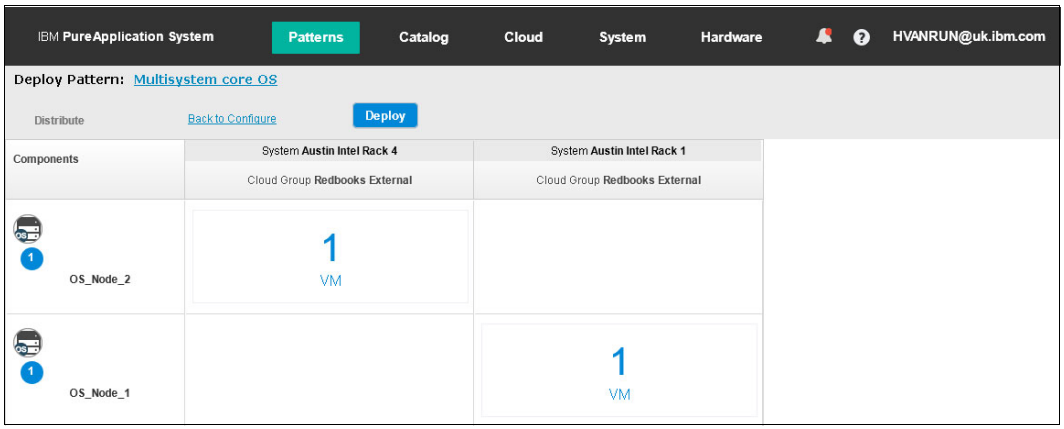


Figure 7-23 Assign VMs to different PureApplication Systems within the deployment subdomain

4. When the deployment is complete, obtain the host name, IP address, and PureApplication System of each VM, as shown in Figure 7-24 on page 181. They are listed in Table 7-3 on page 181 because you need them in the next section.

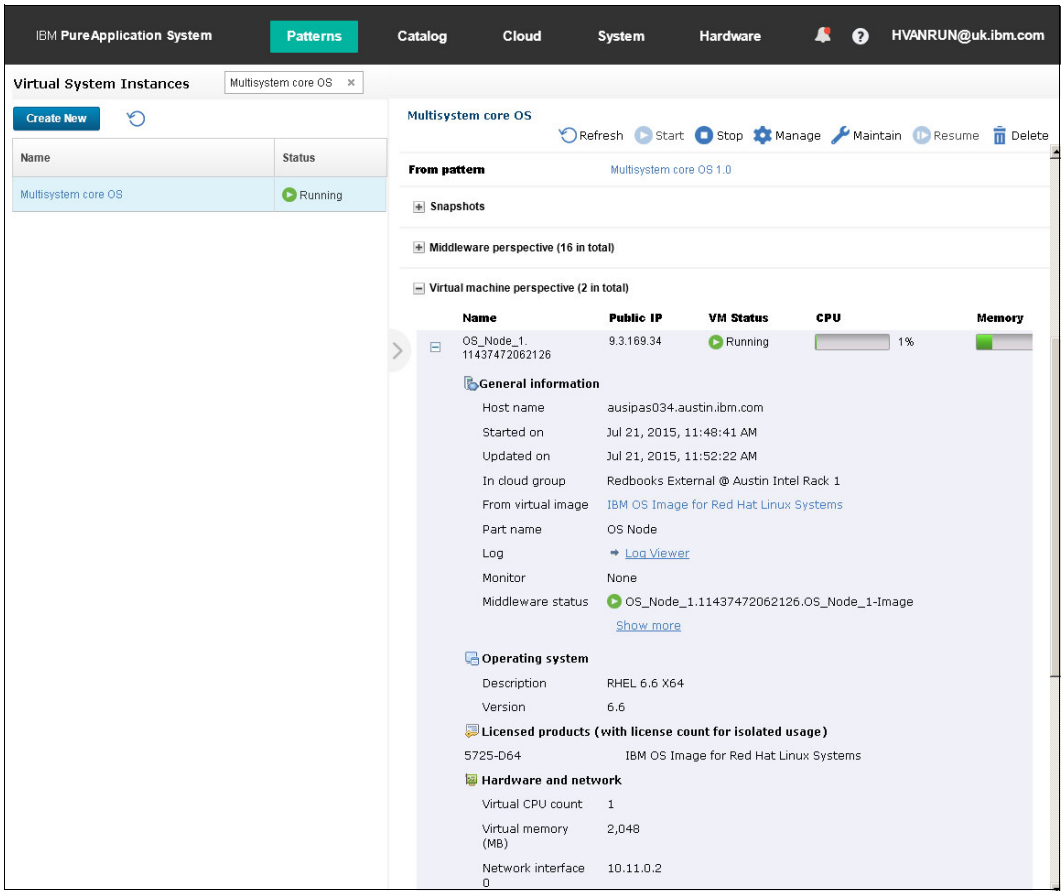


Figure 7-24 Obtain the host name, IP address, and PureApplication System for each VM

Table 7-3 VMs and corresponding host names as deployed on PureApplication Systems

VM	Host name	IP address	PureApplication System
OS_Node_1	ausipas034.austin.ibm.com	9.3.169.34	Austin Intel rack 1
OS_Node_2	ipas-lpar-9-3-171-72.austin.ibm.com	9.3.171.72	Austin Intel rack 4

With the deployment complete, both RHEL OSs on both VMs are automatically registered with the Red Hat Satellite Server. They use the activation key *1-Redbooks*, as configured when you deployed the Red Hat Satellite Service shared service instance to the environment profile *Redbooks - external*.

Note: The exact inner workings of the registration of the deployed VMs with the Red Hat Satellite Server defined in the shared service are beyond the scope of this book. However, it is worth highlighting that the system plug-in *rhusclient* is associated with each VM that is deployed. This system plug-in performs the registration with the Satellite Server and logs to the following file on each VM (the name *OS_Node* and the IDs will vary):

`/opt/IBM/maestro/agent/usr/servers/OS_Node.11437482855011/logs/OS_Node.11437482855011.RHUSClient/trace.log`

Using yum to install packages from a deployed VM

When the VMs are registered with Red Hat Satellite Server, administrators can use **yum** to install new packages by using a command-line interface (CLI).

Note: By default, this package is not installed on the Virtual Image “IBM OS image for Red Hat Linux Systems” version 2.1.2.0 (which is the RHEL 6.6 Virtual Image that ships with PureApplication System 2.1.0.1).

Here are the steps to log in to one of the first VMs that are listed in Table 7-3 on page 181 and install the package for Telnet:

1. Use an SSH client to log in to the VM as virtuser, as shown in Example 7-2. Here, we opt to authenticate with the corresponding SSH private key.

   Note: Authenticating with a private SSH key requires that the corresponding public SSH key be set. In this example, we configured the public SSH key at deployment time of the Virtual System, as described at the following website:

   http://www.ibm.com/support/knowledgecenter/SSCR9A_2.1.0/doc/iwd/pat_ssh_vsys.dita

2. Switch to root by running **sudo su -**, as shown in Example 7-2.

   Example 7-2 Log in to the VM as root by using SSH

```
login as: virtuser
Authenticating with public key "Redbooks" from agent
[virtuser@ausipas034 ~]$ sudo su -
-bash-4.1#
```

3. Run **yum list installed telnet**, as shown in Example 7-3. This command confirms that Telnet is not installed.

   Example 7-3 Confirm that a package for Telnet is not installed

```
bash-4.1# yum list installed telnet
Loaded plugins: rhnplugin, security
This system is receiving updates from RHN Classic or RHN Satellite.
Error: No matching Packages to list
```

4. Run **yum info telnet**, as shown in Example 7-4. This command confirms that Telnet is available from the rhel-x86_64-server-6 repository.

   Example 7-4 Confirm the availability of the Telnet package from the repository in Red Hat Satellite Server

```
bash-4.1# yum info telnet
Loaded plugins: rhnplugin, security
This system is receiving updates from RHN Classic or RHN Satellite.
Available Packages
Name        : telnet
Arch        : x86_64
Epoch       : 1
Version     : 0.17
Release     : 48.el6
Size        : 109 k
Repo        : rhel-x86_64-server-6
```

```
From repo   : rhel-x86_64-server-6
Summary     : The client program for the Telnet remote login protocol
License     : BSD
Description : Telnet is a popular protocol for logging into remote systems over
            : the Internet. The package provides a command line Telnet client
```

5. To install the package for Telnet, run **yum -y install telnet**. The **-y** flag automatically answers *yes* to any questions, making this a convenient command for automating the installation of a package. Example 7-5 shows the output.

Example 7-5 Install the package for Telnet

```
bash-4.1# yum -y install telnet
Loaded plugins: rhnplugin, security
This system is receiving updates from RHN Classic or RHN Satellite.
Setting up Install Process
Resolving Dependencies
--> Running transaction check
---> Package telnet.x86_64 1:0.17-48.el6 will be installed
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================
=
 Package      Arch         Version             Repository                   Size
================================================================================
=
Installing:
 telnet       x86_64       1:0.17-48.el6       rhel-x86_64-server-6        58 k

Transaction Summary
================================================================================
=
Install       1 Package(s)

Total download size: 58 k
Installed size: 110 k
Downloading Packages:
telnet-0.17-48.el6.x86_64.rpm                             |  58 kB     00:00
Running rpm_check_debug
Running Transaction Test
Transaction Test Succeeded
Running Transaction
  Installing : 1:telnet-0.17-48.el6.x86_64                                  1/1
  Verifying  : 1:telnet-0.17-48.el6.x86_64                                  1/1

Installed:
  telnet.x86_64 1:0.17-48.el6

Complete!
```

6. With the installation successfully complete, run **yum list installed telnet** again. The output that is shown in Example 7-6 confirms that the package for Telnet is installed.

Example 7-6 Confirm that Telnet is now installed

```
bash-4.1# yum list installed telnet
Loaded plugins: rhnplugin, security
This system is receiving updates from RHN Classic or RHN Satellite.
Installed Packages
telnet.x86_64                1:0.17-48.el6
@rhel-x86_64-server-6
```

7. Now you can use Telnet, as shown in Example 7-7.

Example 7-7 Use Telnet on the VM

```
-bash-4.1# telnet
telnet> quit
```

Applying updates to deployed VMs from the Red Hat Satellite GUI

There is a more powerful mechanism for applying updates to packages on a deployed VM. Administrators can use the Red Hat Satellite Server GUI to schedule the installation of updates on a VM. Therefore, there is no longer a need to log in to a VM and apply updates by running **yum update**.

From the Red Hat Satellite Server GUI, administrators can also work with groups of systems to schedule the installation of updates. As described in 7.2.4, “Configuring Red Hat Satellite Server on PureApplication System” on page 167, we configured our activation key *Redbooks* so that, upon registration, all systems automatically join the system group *Redbooks*. The systems group in turn enables an administrator to apply updates to all systems in this group.

Note: Although this section describes how to apply updates to packages through the Red Hat Satellite Server GUI, the same mechanism can be used to install (or remove) packages.

Start by using CLI to discover which packages are in need of updates. With that information, you can apply the updates by completing the following steps:

1. Log in to one of the VMs and run **yum list updates**, as shown in Example 7-8.

Example 7-8 Available updates

```
-bash-4.1# yum list updates
Loaded plugins: rhnplugin, security
This system is receiving updates from RHN Classic or RHN Satellite.
Updated Packages
abrt.x86_64
2.0.8-26.el6_6.1                                  rhel-x86_64-server-6
abrt-addon-ccpp.x86_64
2.0.8-26.el6_6.1                                  rhel-x86_64-server-6
2.0.8-26.el6_6.1                                  rhel-x86_64-server-6
0.12.4-4.el6_6.1                                  rhel-x86_64-server-6
3.7.19-260.el6_6.5                                rhel-x86_64-server-6
selinux-policy-targeted.noarch
3.7.19-260.el6_6.5                                rhel-x86_64-server-6
```

```
tzdata.noarch                                2015e-1.el6
rhel-x86_64-server-6
-bash-4.1#
```

2. Log in to the Red Hat Satellite Server GUI, as described in 7.2.4, “Configuring Red Hat Satellite Server on PureApplication System” on page 167.
3. Go to **System Groups**. You see the System Group *Redbooks*, as shown in Figure 7-25. This group contains two systems. These are the two VMs that we deployed in 7.2.7, “Using Red Hat Satellite Server integration on PureApplication System” on page 179.

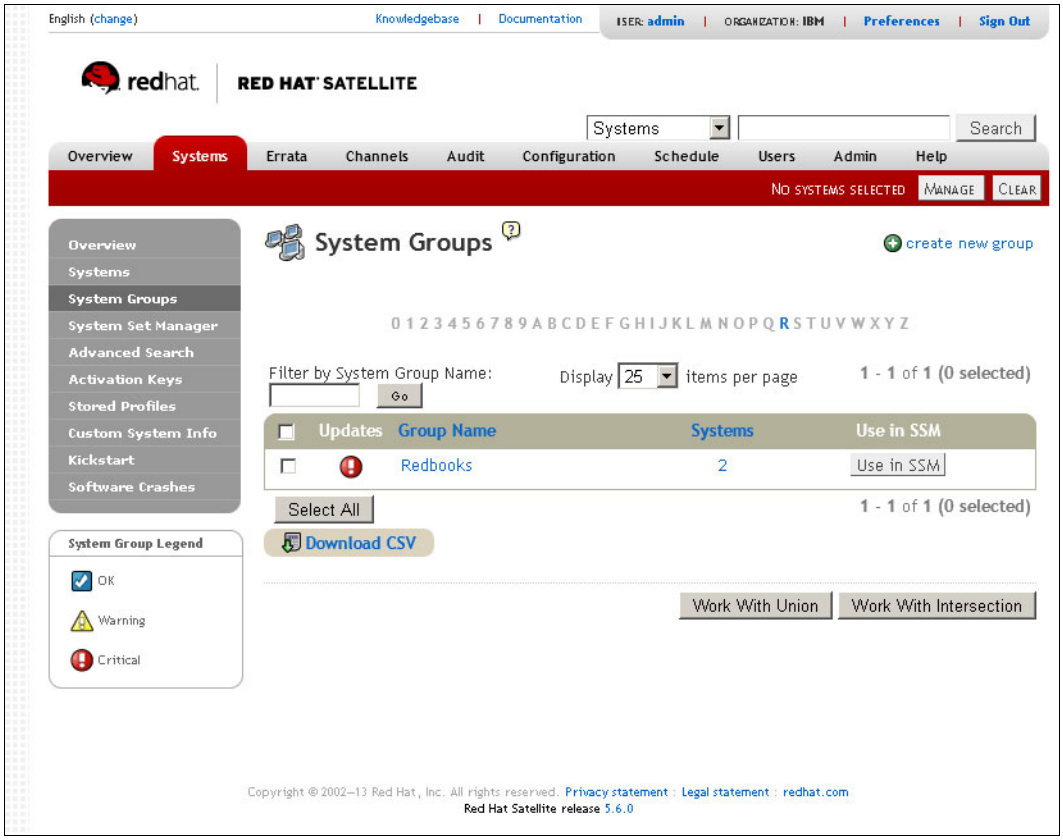


Figure 7-25 Examine all System Groups through the Red Hat Satellite Server GUI

4. Click the System Group *Redbooks* to view more specifics, as shown in Figure 7-26.

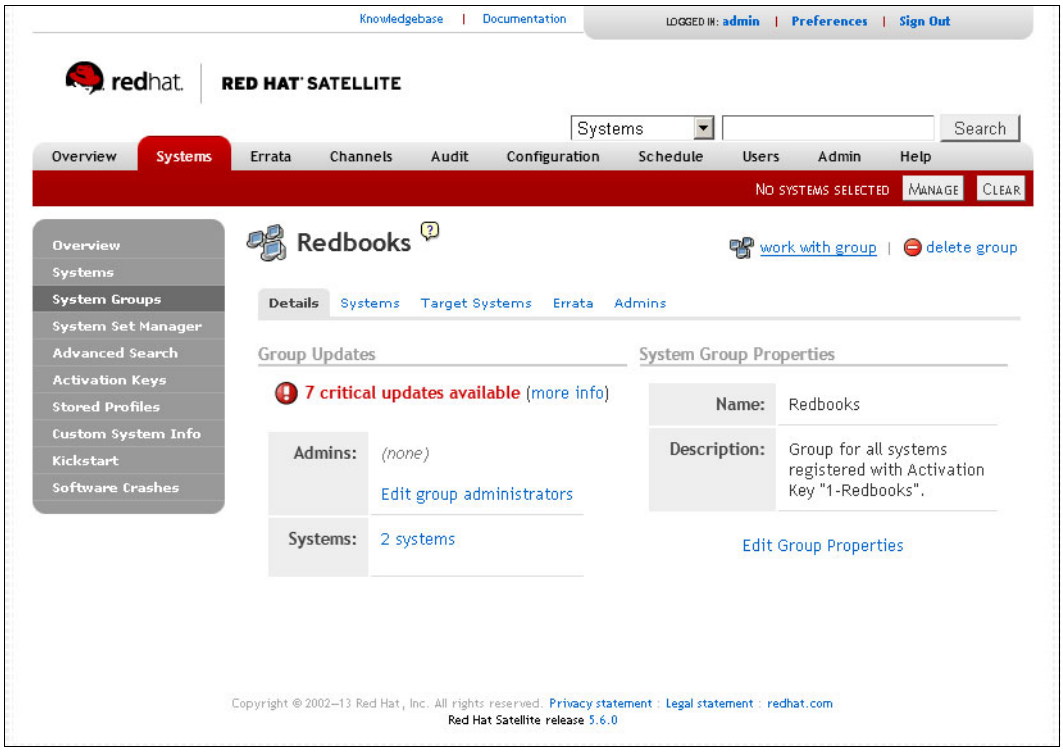


Figure 7-26 The System Group Redbooks

5. You want to work with all systems that are in the System Group, so click **work with group** (upper right corner), which adds all systems to the *System Set Manager*, as shown in Figure 7-27 on page 187.

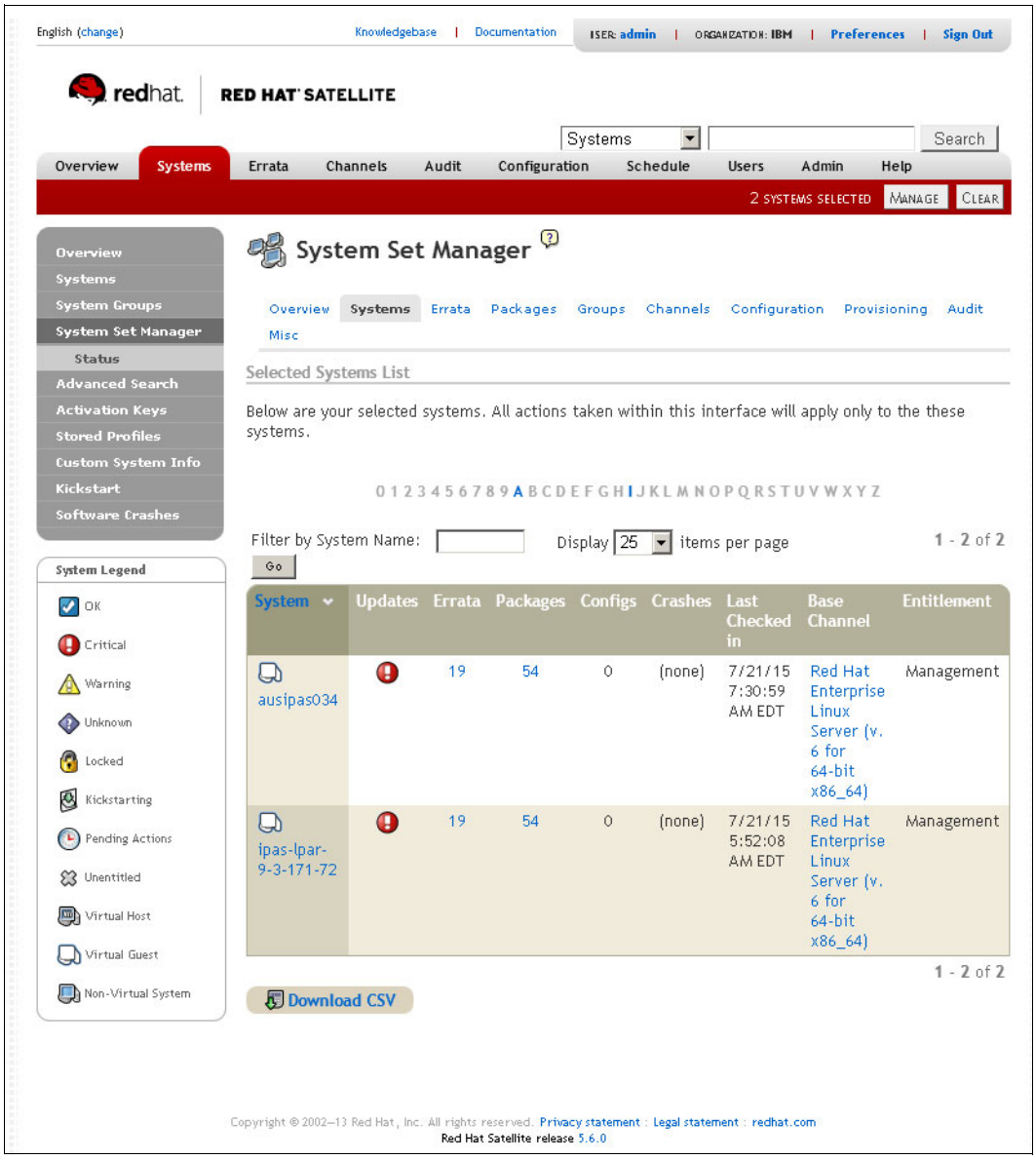


Figure 7-27 Both systems are now set up to be used by the System Set Manager

6. There are 19 applicable errata that can be applied to the systems. Click the **Errata** tab to review them.

7. Apply all errata to both systems. Click **Select All** and then **Apply Errata**, as shown in Figure 7-28.

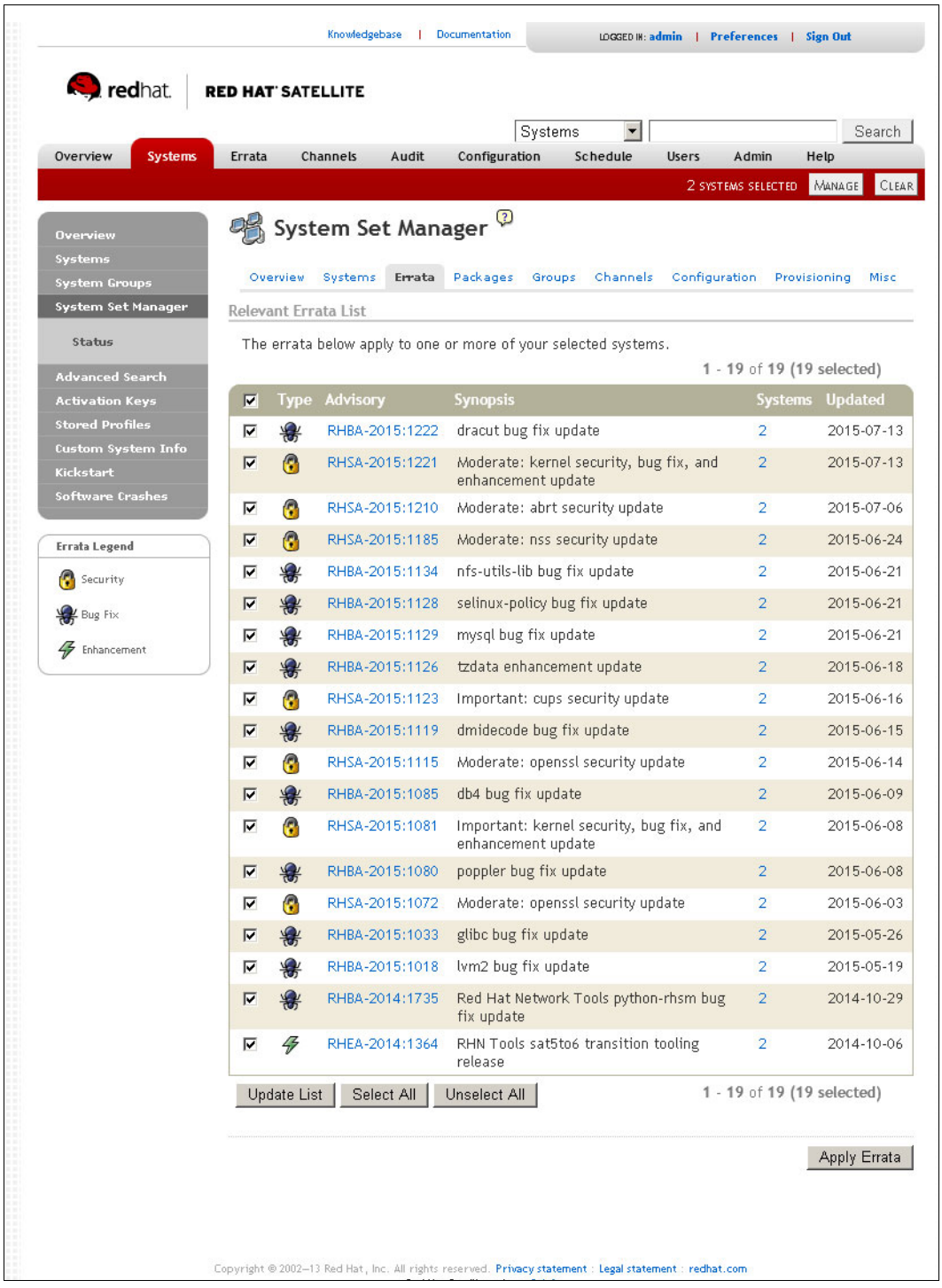


Figure 7-28 Apply all errata that are applicable to both systems in the System Set Manager

8. Schedule the actions to apply all errata as soon as possible and click **Schedule Updates**, as shown in Figure 7-29 on page 189. There can be scenarios where scheduling the actions to be carried out later is preferable.

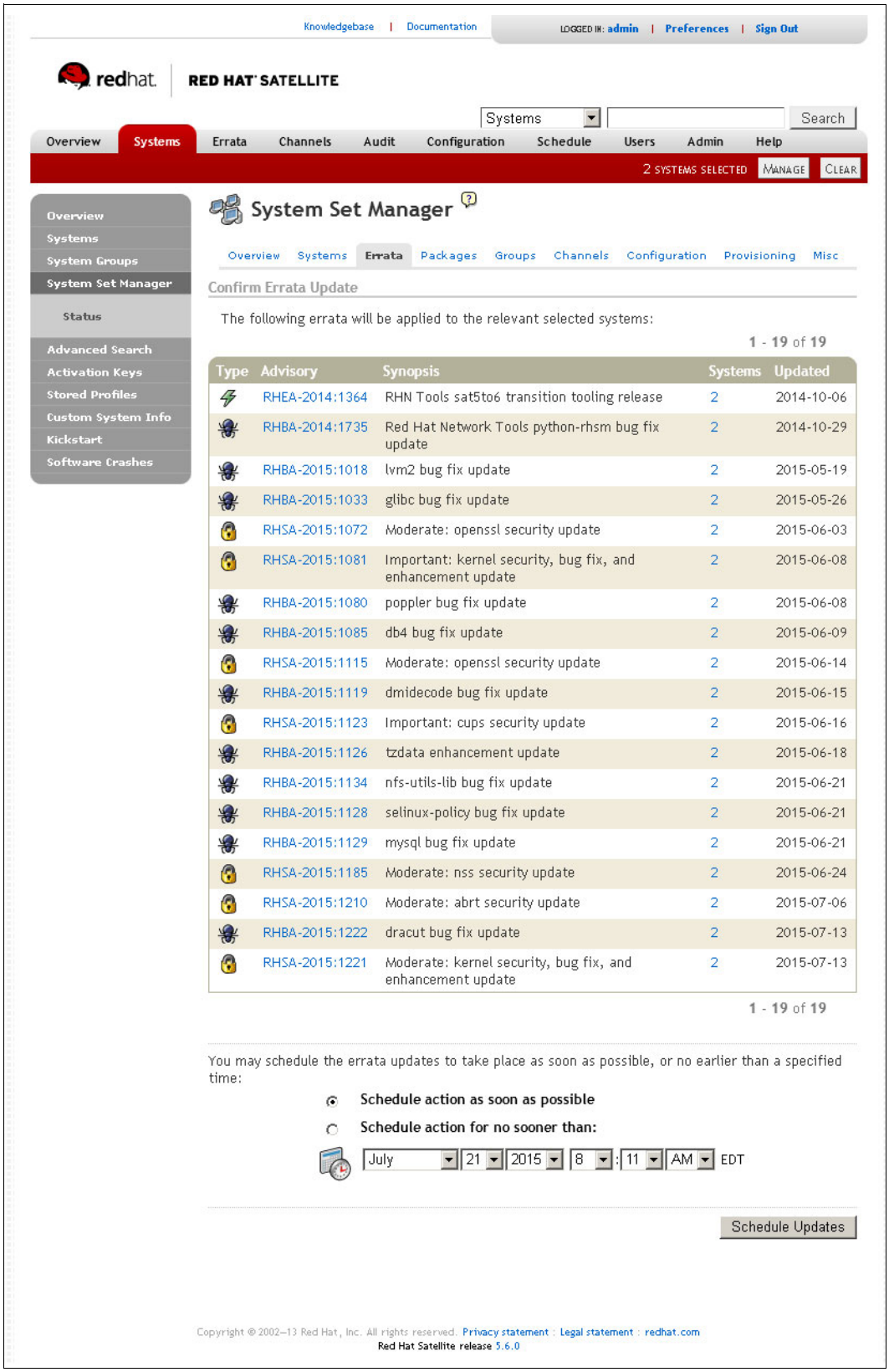


Figure 7-29 Schedule the actions to apply all errata as soon as possible

9. Click **Schedule** to examine the status of the scheduled actions. Because you scheduled the errata to be applied as soon as possible, click **Completed Actions**.
10. Within a few minutes, you see all of the entries (in this case, 19) listed in the Completed Actions section. All errata are applied on two systems, as shown in Figure 7-30.

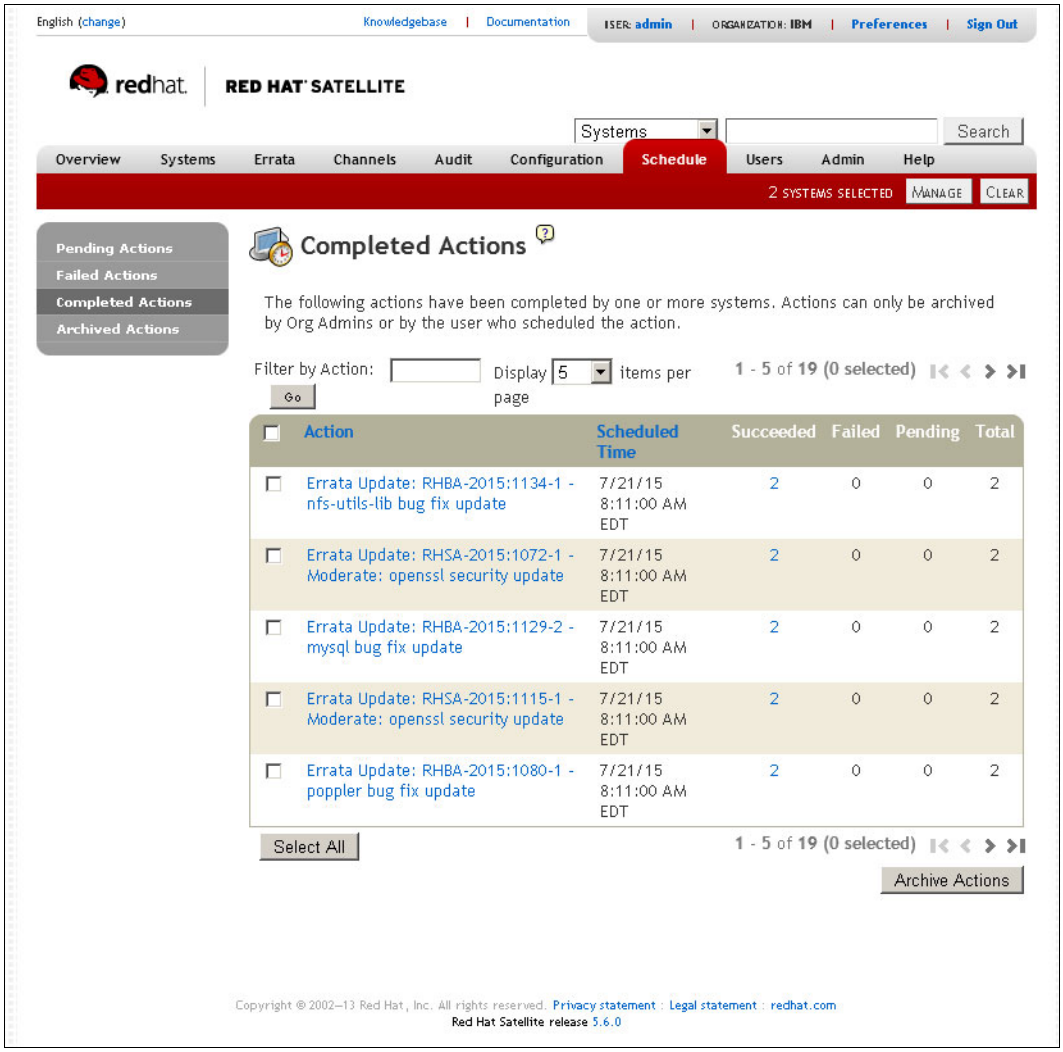


Figure 7-30 A total of 19 completed actions on both systems confirms that all errata were applied

Examine the System Group *Redbooks* again.

11. Click **Systems** → **System Groups**. The status of the System Group reports a status of **OK**, as shown in Figure 7-31 on page 191. This confirms that there are no more errata that are applicable to either of the two systems in the System Group *Redbooks* at this time.

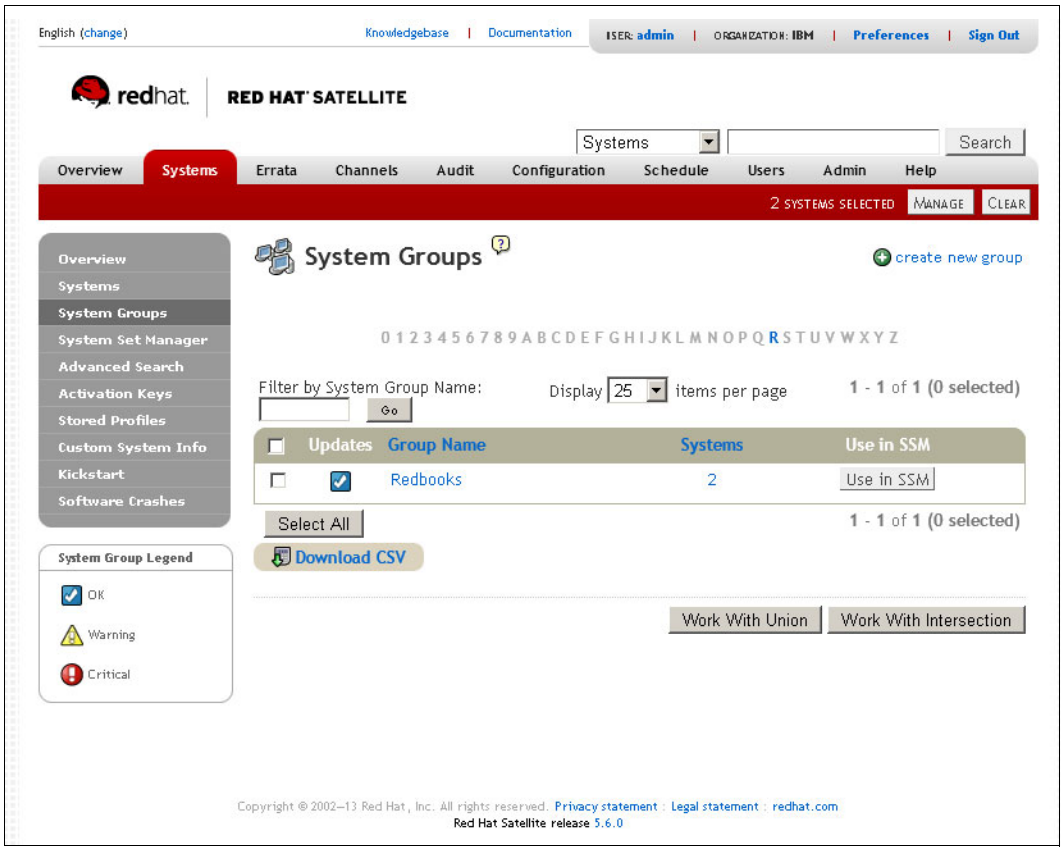


Figure 7-31 The reporting status for System Group Redbooks is OK

12. Log in to one of the VMs again and run **`yum list updates`**. As shown in Example 7-9, no updates for any packages are available.

Example 7-9 List all applicable updates for the installed packages

```
-bash-4.1# yum list updates
Loaded plugins: rhnplugin, security
This system is receiving updates from RHN Classic or RHN Satellite.
-bash-4.1#
```

Installing and updating packages at Virtual System Pattern deployment time

In addition to installing and updating packages on deployed VMs, you can also perform these actions at the deployment time of a VSP. The software component *Red Hat Package Manager* can install a set of packages on a VM within a VSP. In addition, this software component can be configured to update all packages to the latest version.

As an example, you can build and deploy a simple VSP by completing the following steps:

1. Create a VSP that is named *Core OS with Firefox and Telnet*, consisting of a single VM with Red Hat Enterprise Linux 6.6. Use the IBM OS image for Red Hat Linux Systems 2.1.2.0.
2. Add the software component Red Hat Package Installer to the VM. Configure the software component with the following information:
   - Yum Package list: `firefox,telnet`
   - Fail deployment if above packages are not installed: Enabled
   - Update VM with latest OS patches: Enabled
3. The VSP is shown in Figure 7-32. Click **save** to save your pattern.

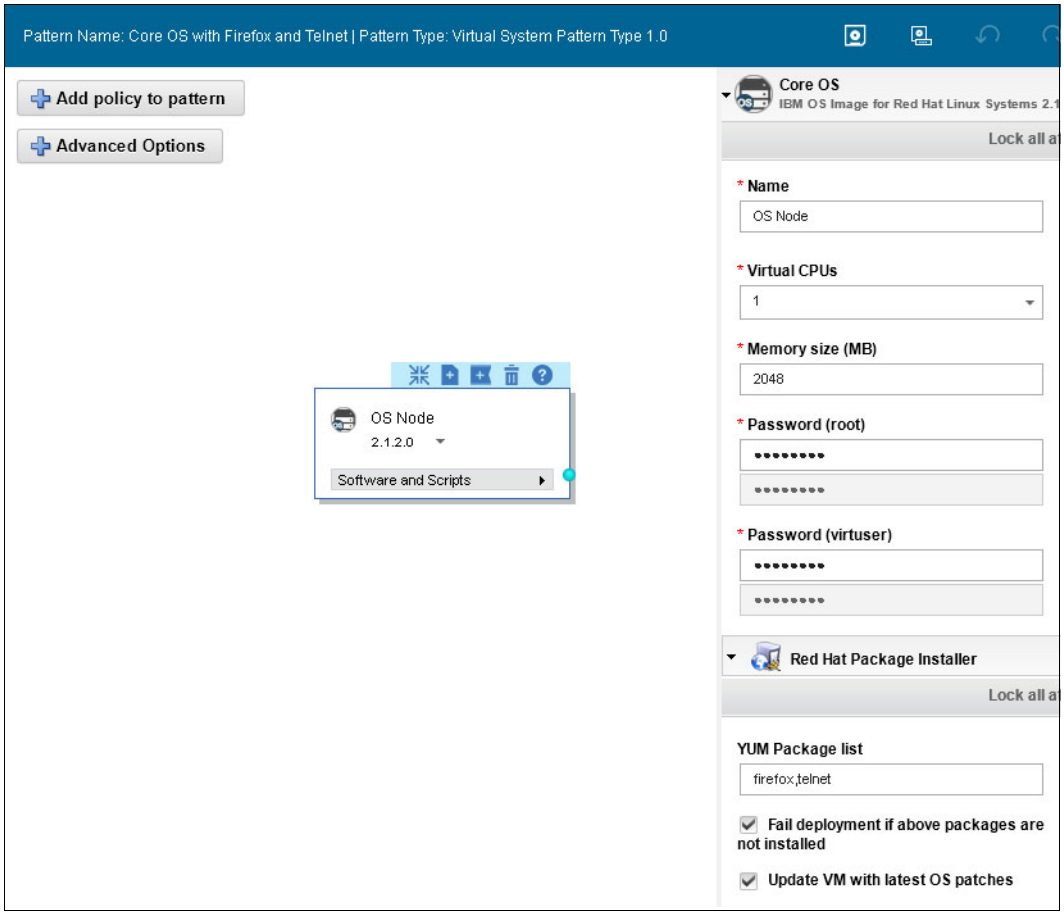


Figure 7-32 Virtual System Pattern by using the software component Red Hat Package Installer

4. Deploy the VSP by using the environment profile *Redbooks - external*, as shown in Figure 7-33 on page 193. This profile ensures that the VM automatically registers with the Red Hat Satellite Server that was configured during the deployment of the Red Hat Satellite Service (external) shared service (see 7.2.6, “Deploying the Red Hat Satellite service shared service” on page 177).

Note: The inner workings of the software component Red Hat Package Installer are beyond the scope of this book. However, it is worth highlighting that this software component logs to the following file on each VM (the name `OS_Node` and the IDs vary):

`/opt/IBM/maestro/agent/usr/servers/OS_Node.11437482855011/logs/OS_Node.11437482855011.Red_Hat_Package_Installer-Part/trace.log`

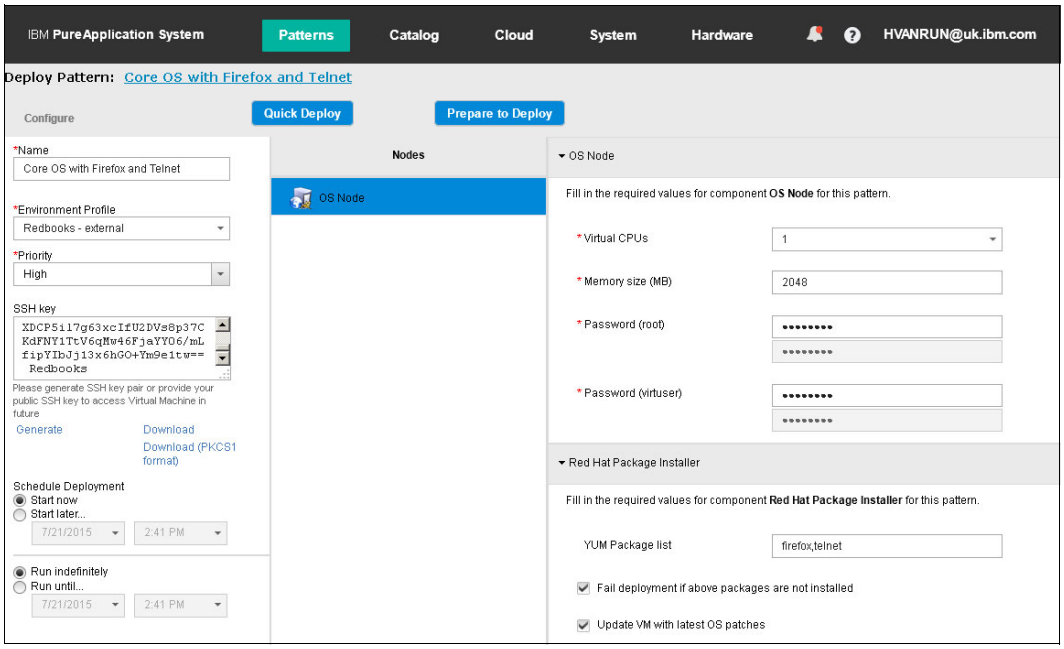


Figure 7-33 Deploy the Virtual System Pattern that is named Core OS with Firefox and Telnet

5. When the deployment is complete, obtain the host name and IP address from the Virtual System Instance, as shown in Figure 7-34. You need this information later. In this case, the VM is assigned host name `ipas-lpar-9-3-171-70.austin.ibm.com` and IP address `9.3.171.70`.

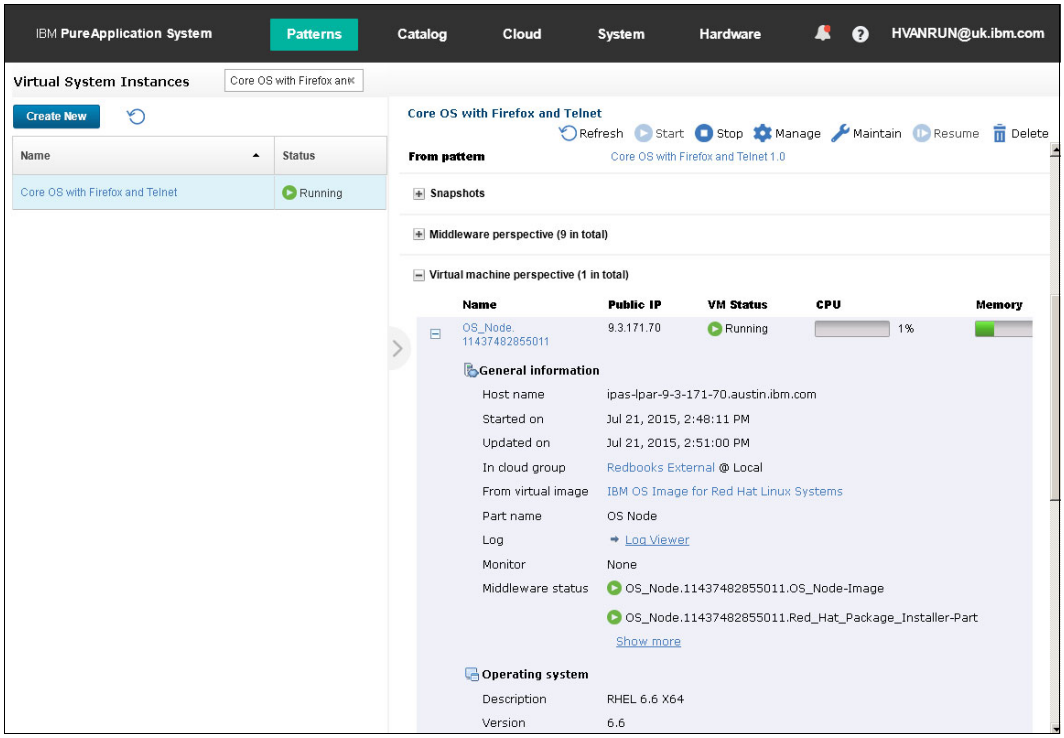


Figure 7-34 Obtain the host name and IP address of the deployed VM

6. Log in to the VM as virtuser and switch to the root user by running **`sudo su -`**.
7. Run **`yum list installed firefox telnet`** to confirm that both packages are installed, as shown in Example 7-10.

Example 7-10 Confirm that Firefox and Telnet are installed

```
-bash-4.1# yum list installed firefox telnet
Loaded plugins: rhnplugin, security
This system is receiving updates from RHN Classic or RHN Satellite.
Installed Packages
firefox.x86_64                          38.1.0-1.el6_6
@rhel-x86_64-server-6
telnet.x86_64                           1:0.17-48.el6
@rhel-x86_64-server-6
```

8. Run **`yum list updates`** to review updates that are applicable to any of the installed packages. In this example, we configured the software component Red Hat Package Installer to Update VM with the latest OS patches. As expected, no updates are listed, as shown in Example 7-11.

Example 7-11 Confirm that there are no pending updates for any packages

```
-bash-4.1# yum list updates
Loaded plugins: rhnplugin, security
This system is receiving updates from RHN Classic or RHN Satellite.
```

You can log in to the Red Hat Satellite Server GUI and examine the newly registered VM that is deployed. Because the latest updates are applied at deployment time, no errata are applicable, as shown in Figure 7-35. The system ipas-lpar-9-3-171-70 corresponds to the VM that was deployed.

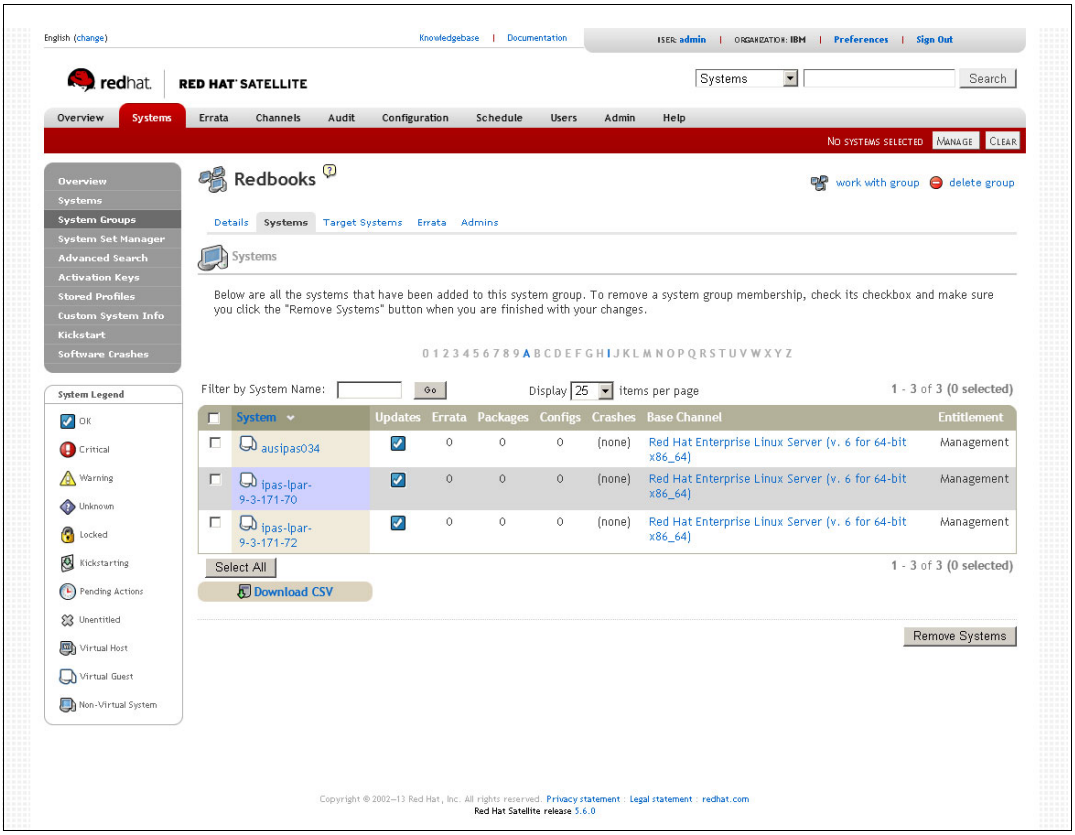


Figure 7-35 No errata or packages are applicable to the newly deployed VM

7.3 IBM AIX OS maintenance by using IBM Endpoint Manager

IBM Endpoint Manager is a solution for addressing highly diverse, distributed, and complex IT environments. It provides real-time visibility and control through a single infrastructure, single agent, and single console for policy-based management capabilities, including asset discovery and inventory, patch management, OS deployment, remote desktop control, endpoint protection, security configuration, and vulnerability management.

PureApplication System and PureApplication Software do not perform any patch management. PureApplication System provides an Endpoint Manager shared service that you can use to access Endpoint Manager running remotely on another system. For the purposes of this publication, the Endpoint Manager shared service is deployed on a POWER-based PureApplication System. The Endpoint Manager server runs on an Intel based system. Installation of the Endpoint Manager server is beyond the scope of this publication.

After the PureApplication System or PureApplication Software is deployed, clients (deployed workloads) are registered to the Endpoint Manager server. Patch management is performed on the Endpoint Manager server from its console.

From PureApplication System or PureApplication Software, the system administrator or workload administrator deploys the Endpoint Manager shared service, which is configured to point to the Endpoint Manager server. Any workloads that are deployed on the same cloud group as the Endpoint Manager shared service are automatically connected to the Endpoint Manager server through the Endpoint Manager shared service.

Note: PureApplication System and PureApplication Software clients are entitled to install and use an Endpoint Manager server for the purpose of AIX and Windows OS patch management.

Deploying the Endpoint Manager shared service

The Endpoint Manager shared service is shipped with PureApplication System and PureApplication Software. and is included in the Foundation Pattern type. Endpoint Manager acts as a relay between an external Endpoint Manager server and Endpoint Manager clients. By default, Endpoint Manager clients are integrated into virtual application and virtual system deployments.

After you deploy the Endpoint Manager service in a cloud group, the Endpoint Manager clients automatically connect to the shared service to receive OS updates and to install patches on VMs. For more information about the Endpoint Manager console and patch management, see “Accessing the Endpoint Manager server” on page 201.

The Endpoint Manager service acts as a relay to receive updates and install patches on clients. You must use the Endpoint Manager console to deliver fixes and perform other administrative tasks on the Endpoint Manager server.

When you deploy the Endpoint Manager service in a cloud group, you configure the shared service by pointing to a *masthead file*. The masthead file contains connection information about the Endpoint Manager server, including the host name, security certificates, server configuration details, and more. The Endpoint Manager clients, in the cloud group where the shared service is deployed, automatically discover the shared service and import the masthead file with no manual configuration.

To deploy the Endpoint Manager service, complete the following steps:

1. Click **Patterns** → **Patterns** → **Shared Services**. A list of available shared services displays (Figure 7-36).

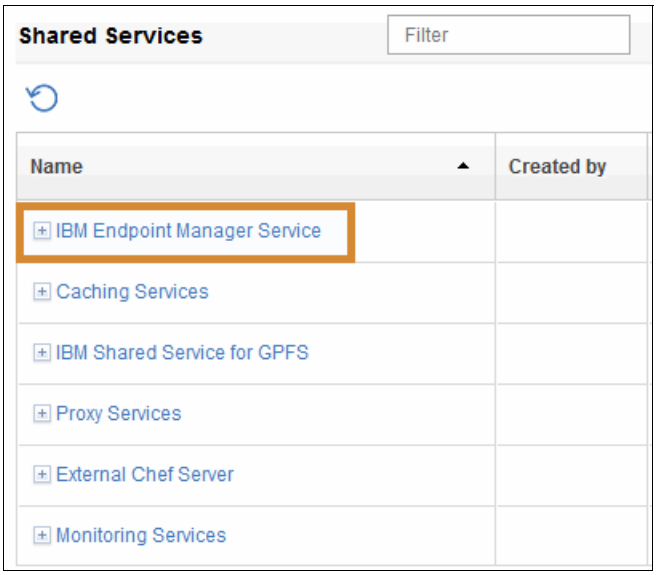


Figure 7-36 List of shared services

2. Expand **Endpoint Manager Service**. The latest version of the shared service is listed. Select the shared service. Details of the service are shown in the right pane, as shown in Figure 7-37.

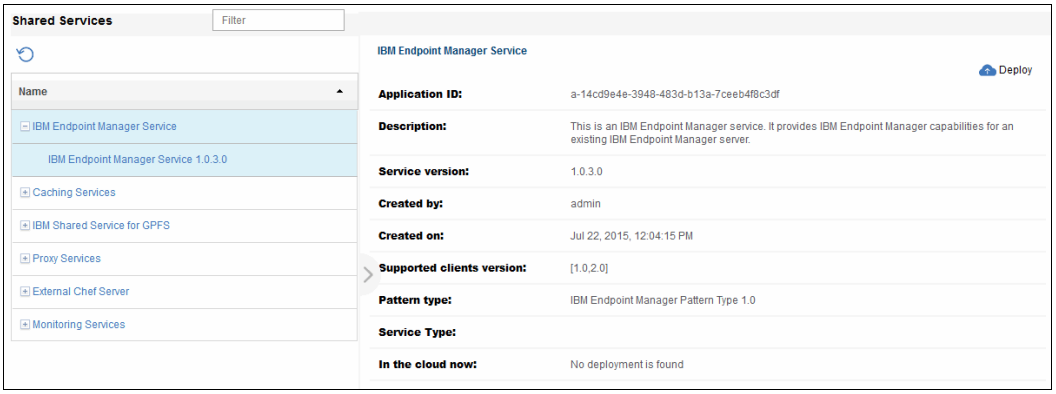


Figure 7-37 Endpoint Manager service details

3. The service is not deployed. To deploy it, click the **Deploy** icon in the upper right corner of the window.
4. Requirements for the deployment are minimal. The masthead file (or location of the masthead file) is needed for the deployment. You can obtain this location from the group that manages the Endpoint Manager server. In the deployment window, the masthead file is required. The relay files are optional.

Note: In the database, use the Endpoint Manager Administration Tool to export the masthead from the database by completing the following steps:

- Click **Start Menu** → **All Programs** → **IBM Endpoint Manager** → **IBM Endpoint Manager Administration Tool**.
- Point to the deployment's `license.pvk` site signing key. Click **OK** and enter the password.
- On the Masthead Management tab, click **Export Masthead** to save `masthead.afxm` to your desktop.

The `ActionSite.afxm` is on the Endpoint Manager server and is found in one of the following site version folders at `C:\Program Files\BigFix Enterprise\BES Server\sitearchive\actionsite\archiveDir\actionsite_<nnn>`, where `nnn` = actionsite version #:

- `C:\Program Files\BigFix Enterprise\BES Server`
- For 64-bit OSs: `C:\Program Files(x86)\BigFix Enterprise\BES Server`

`ActionSite.afxm` on the BES clients is in `C:\Program Files\BigFix Enterprise\BES Client`.

5. Enter the location of the masthead file. In this example, the masthead file is on the Endpoint Manager server, as shown in Figure 7-38. It can also be on the local file system. Click **OK**.

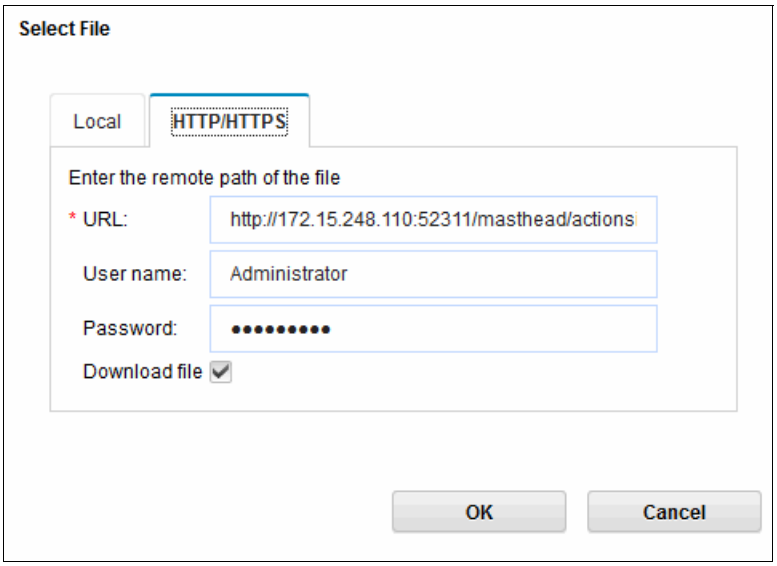


Figure 7-38 Masthead file selection

6. Click **Prepare to Deploy**. The system prepares the deployment pattern.

7. Click **Deploy**, as shown in Figure 7-39.

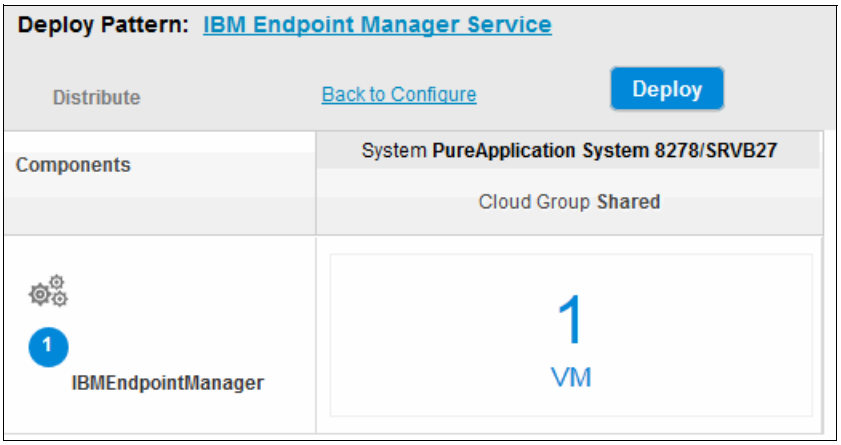


Figure 7-39 Deploy Endpoint Manager

8. With the service deployed, you see an instance of the shared service. Click **Patterns** → **Pattern Instances** → **Shared Services**. The instance is displayed in the list, as shown in Figure 7-40.

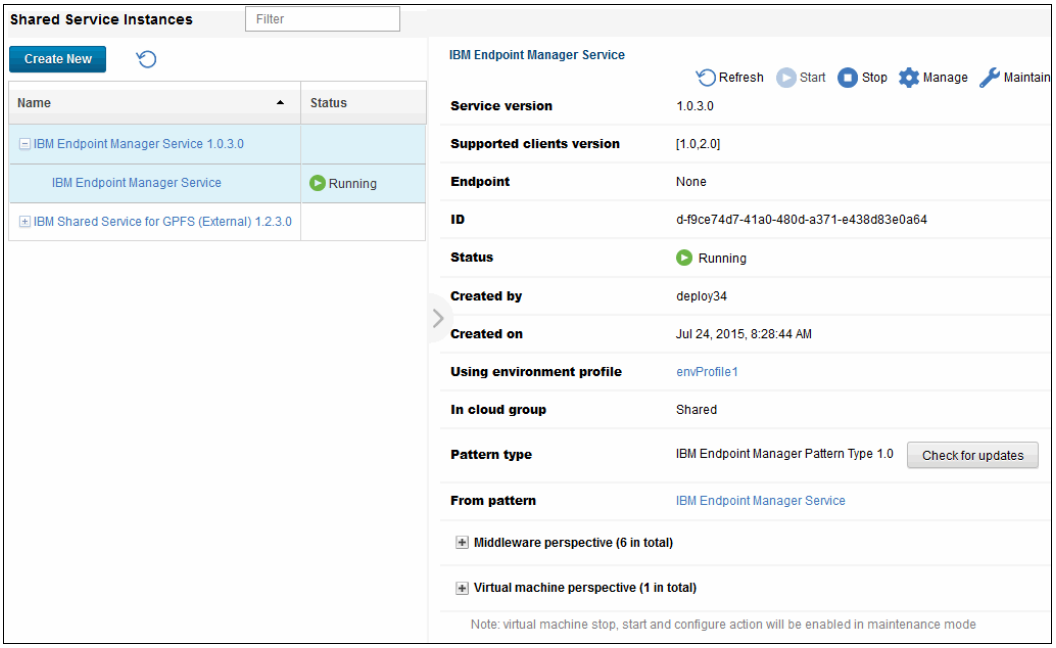


Figure 7-40 Instance of the Endpoint Manager service

9. Expand **Virtual machine perspective**. The deployment created one VM, as shown in Figure 7-41.

Figure 7-41 VM created by the Endpoint Manager service deployment

Instances and the Endpoint Manager service

The Endpoint Manager client is part of any virtual image that is supplied by PureApplication System. The client adds a background controller service that is set to run automatically. After the shared service is deployed, any virtual system instances that you deploy reference the shared service, as shown in Figure 7-42 on page 201. The shared service and Endpoint Manager client work together to discover and apply patches from the Endpoint Manager server.

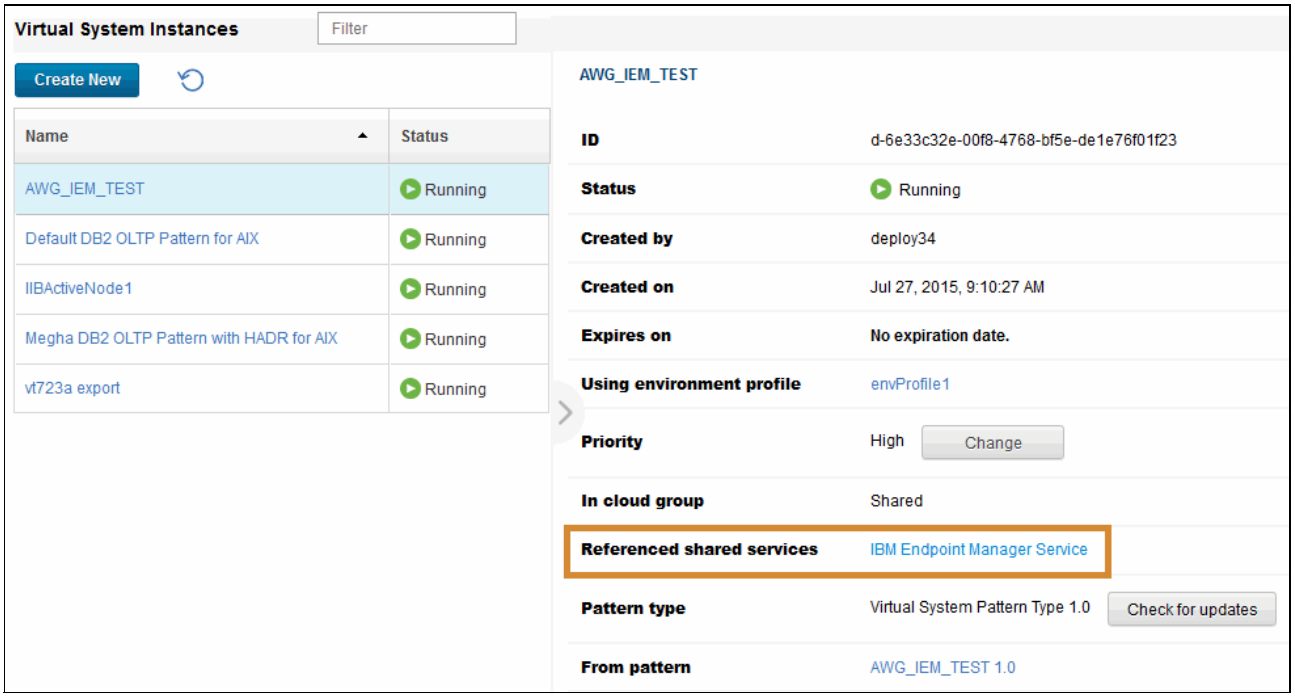


Figure 7-42 Virtual system with reference to the Endpoint Manager service

Accessing the Endpoint Manager server

The Endpoint Manager server console is used by the operator to monitor and repair networked computers running the Endpoint Manager client. You can run the Endpoint Manager console on a 64-bit Windows OS that has network access to the Endpoint Manager server. There are two kinds of authorized console users:

- Operators: Operators manage the day-to-day operation of the program, including fixlet management and action deployment, subject to the management rights that are assigned by a site administrator or master operator.
- Master operators: Master operators have the added authority to assign management rights to other console operators.

Endpoint Manager for patch management is a console view that provides an automated patching process. This view gives you near real-time visibility and enforcement to deploy and manage patches to all distributed endpoints. Some of the features are that Endpoint Manager can accomplish the following tasks:

- Automatically manage patches to hundreds of thousands of endpoints for multiple OSs and applications, regardless of location, connection type, or status
- Apply only the correct patches to the correct endpoint
- Visibility into patch compliance with near real-time monitoring and reporting
- Near real-time visibility and control from a single management console

The Endpoint Manager console has several views that are useful to the system administrator of a PureApplication System or PureApplication Software installation. This publication specifically describes patches and fixlets for IBM AIX 7.1. Screen captures and descriptions in this section might vary from the installation and configuration of your console.

Patch Management

Patches can be found in this view, as shown in Figure 7-43. Patches are categorized by vendor and version number. Patches are added to the Endpoint Manager server by an administrator and managed from the console. PureApplication System does *not* perform any patch management.

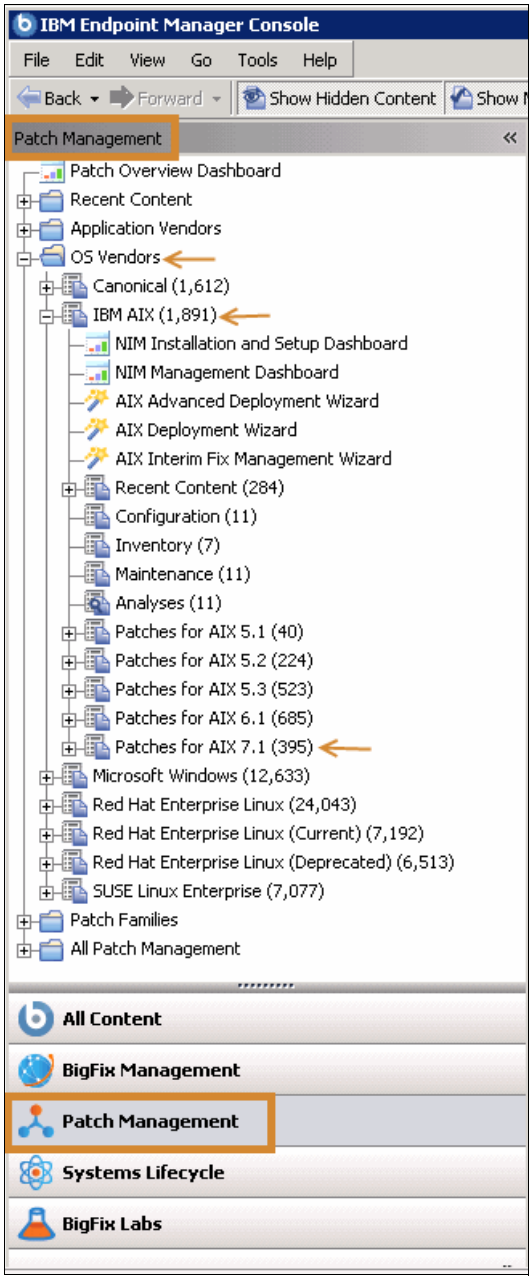
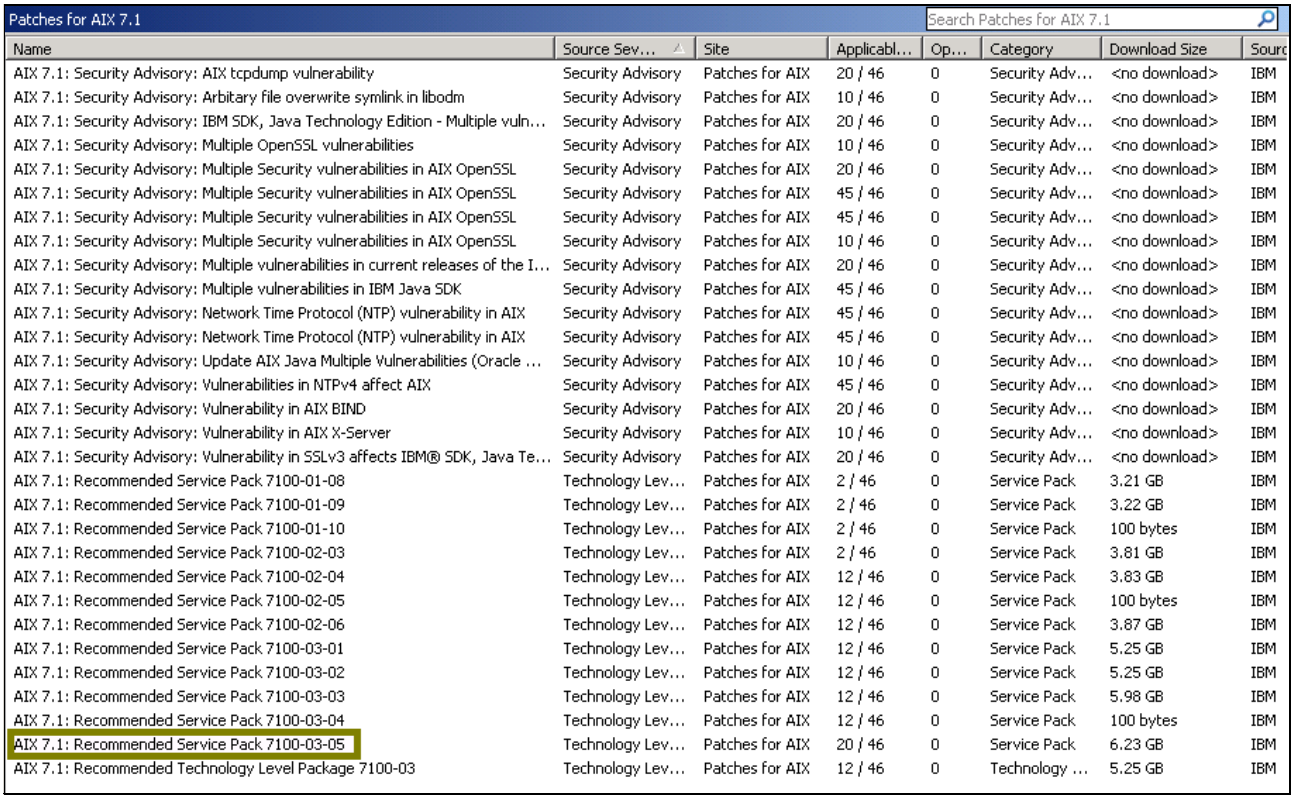


Figure 7-43 Patch management for IBM AIX 7.1

For AIX, patches can include service advisories, service packs, APARs, and bulletins. For a sample of patches, see Figure 7-44 on page 203.



Name	Source Sev...	Site	Applicabl...	Op...	Category	Download Size	Sourc
AIX 7.1: Security Advisory: AIX tcpdump vulnerability	Security Advisory	Patches for AIX	20 / 46	0	Security Adv...	<no download>	IBM
AIX 7.1: Security Advisory: Arbitary file overwrite symlink in libodm	Security Advisory	Patches for AIX	10 / 46	0	Security Adv...	<no download>	IBM
AIX 7.1: Security Advisory: IBM SDK, Java Technology Edition - Multiple vuln...	Security Advisory	Patches for AIX	20 / 46	0	Security Adv...	<no download>	IBM
AIX 7.1: Security Advisory: Multiple OpenSSL vulnerabilities	Security Advisory	Patches for AIX	10 / 46	0	Security Adv...	<no download>	IBM
AIX 7.1: Security Advisory: Multiple Security vulnerabilities in AIX OpenSSL	Security Advisory	Patches for AIX	20 / 46	0	Security Adv...	<no download>	IBM
AIX 7.1: Security Advisory: Multiple Security vulnerabilities in AIX OpenSSL	Security Advisory	Patches for AIX	45 / 46	0	Security Adv...	<no download>	IBM
AIX 7.1: Security Advisory: Multiple Security vulnerabilities in AIX OpenSSL	Security Advisory	Patches for AIX	45 / 46	0	Security Adv...	<no download>	IBM
AIX 7.1: Security Advisory: Multiple Security vulnerabilities in AIX OpenSSL	Security Advisory	Patches for AIX	10 / 46	0	Security Adv...	<no download>	IBM
AIX 7.1: Security Advisory: Multiple vulnerabilities in current releases of the I...	Security Advisory	Patches for AIX	20 / 46	0	Security Adv...	<no download>	IBM
AIX 7.1: Security Advisory: Multiple vulnerabilities in IBM Java SDK	Security Advisory	Patches for AIX	45 / 46	0	Security Adv...	<no download>	IBM
AIX 7.1: Security Advisory: Network Time Protocol (NTP) vulnerability in AIX	Security Advisory	Patches for AIX	45 / 46	0	Security Adv...	<no download>	IBM
AIX 7.1: Security Advisory: Network Time Protocol (NTP) vulnerability in AIX	Security Advisory	Patches for AIX	45 / 46	0	Security Adv...	<no download>	IBM
AIX 7.1: Security Advisory: Update AIX Java Multiple Vulnerabilities (Oracle ...	Security Advisory	Patches for AIX	10 / 46	0	Security Adv...	<no download>	IBM
AIX 7.1: Security Advisory: Vulnerabilities in NTPv4 affect AIX	Security Advisory	Patches for AIX	45 / 46	0	Security Adv...	<no download>	IBM
AIX 7.1: Security Advisory: Vulnerability in AIX BIND	Security Advisory	Patches for AIX	20 / 46	0	Security Adv...	<no download>	IBM
AIX 7.1: Security Advisory: Vulnerability in AIX X-Server	Security Advisory	Patches for AIX	10 / 46	0	Security Adv...	<no download>	IBM
AIX 7.1: Security Advisory: Vulnerability in SSLv3 affects IBM® SDK, Java Te...	Security Advisory	Patches for AIX	20 / 46	0	Security Adv...	<no download>	IBM
AIX 7.1: Recommended Service Pack 7100-01-08	Technology Lev...	Patches for AIX	2 / 46	0	Service Pack	3.21 GB	IBM
AIX 7.1: Recommended Service Pack 7100-01-09	Technology Lev...	Patches for AIX	2 / 46	0	Service Pack	3.22 GB	IBM
AIX 7.1: Recommended Service Pack 7100-01-10	Technology Lev...	Patches for AIX	2 / 46	0	Service Pack	100 bytes	IBM
AIX 7.1: Recommended Service Pack 7100-02-03	Technology Lev...	Patches for AIX	2 / 46	0	Service Pack	3.81 GB	IBM
AIX 7.1: Recommended Service Pack 7100-02-04	Technology Lev...	Patches for AIX	12 / 46	0	Service Pack	3.83 GB	IBM
AIX 7.1: Recommended Service Pack 7100-02-05	Technology Lev...	Patches for AIX	12 / 46	0	Service Pack	100 bytes	IBM
AIX 7.1: Recommended Service Pack 7100-02-06	Technology Lev...	Patches for AIX	12 / 46	0	Service Pack	3.87 GB	IBM
AIX 7.1: Recommended Service Pack 7100-03-01	Technology Lev...	Patches for AIX	12 / 46	0	Service Pack	5.25 GB	IBM
AIX 7.1: Recommended Service Pack 7100-03-02	Technology Lev...	Patches for AIX	12 / 46	0	Service Pack	5.25 GB	IBM
AIX 7.1: Recommended Service Pack 7100-03-03	Technology Lev...	Patches for AIX	12 / 46	0	Service Pack	5.98 GB	IBM
AIX 7.1: Recommended Service Pack 7100-03-04	Technology Lev...	Patches for AIX	12 / 46	0	Service Pack	100 bytes	IBM
AIX 7.1: Recommended Service Pack 7100-03-05	Technology Lev...	Patches for AIX	20 / 46	0	Service Pack	6.23 GB	IBM
AIX 7.1: Recommended Technology Level Package 7100-03	Technology Lev...	Patches for AIX	12 / 46	0	Technology ...	5.25 GB	IBM

Figure 7-44 Patches for IBM AIX 7.1

The details and description of each patch are available as shown in Figure 7-45. You can also see which computers (VMs) must have the patch applied.

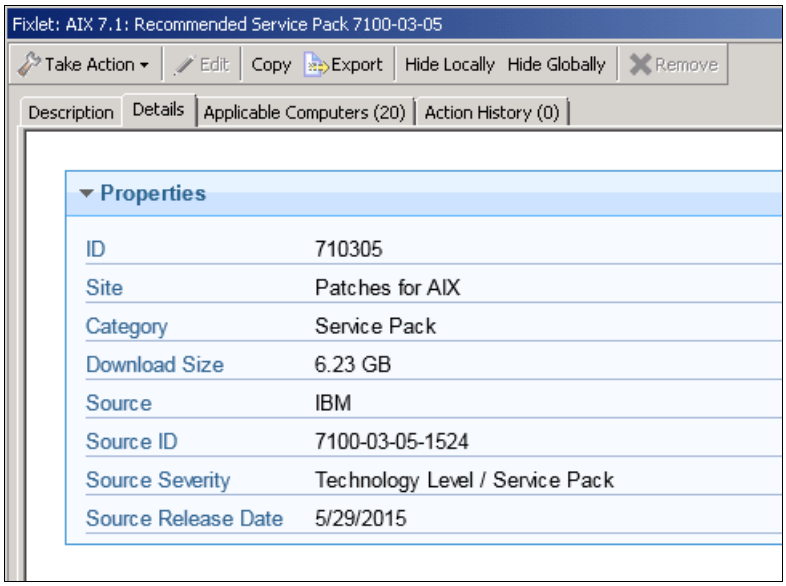


Figure 7-45 Patch details

All Content

As the view name implies, you can see all the content from this view. This view is useful for finding specific VMs that have their OS patches managed by Endpoint Manager. In the example that is shown in Figure 7-46, you see a list of all the managed VMs.

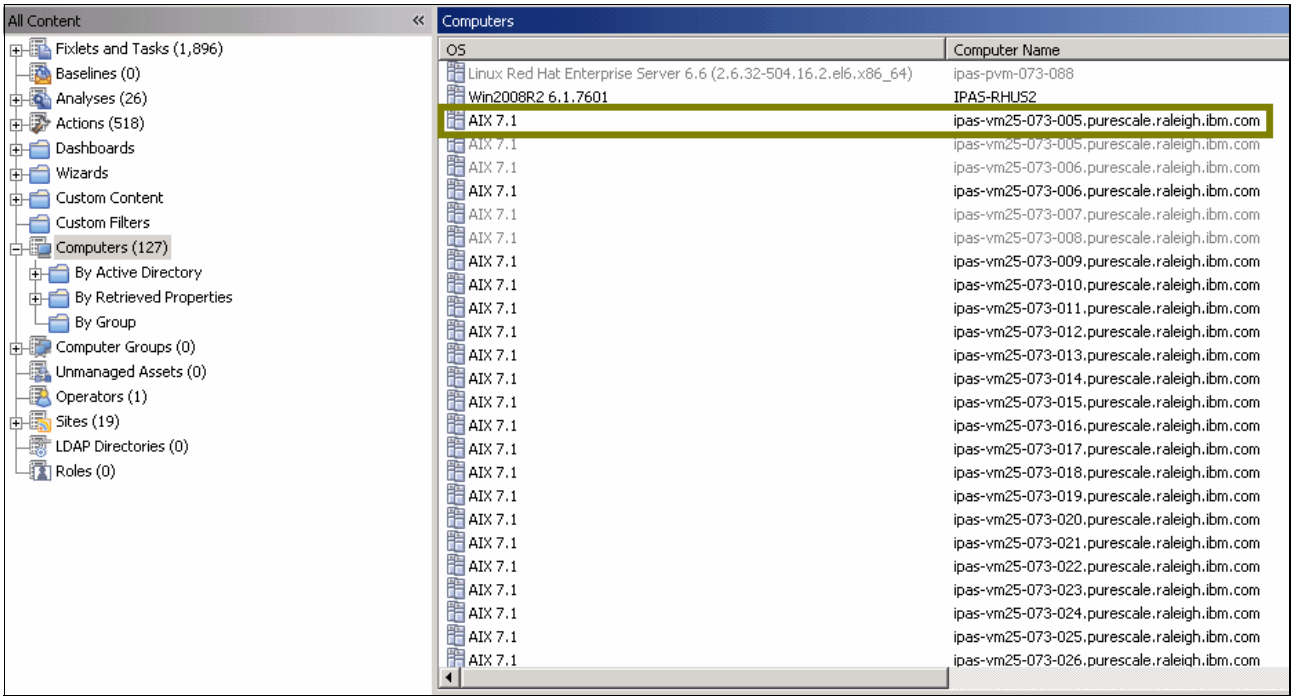


Figure 7-46 All Content view - list of all managed VMs

As you select an individual computer, the details of that VM are shown. You can view the fixlets and tasks that are relevant to that VM, as shown in Figure 7-47.

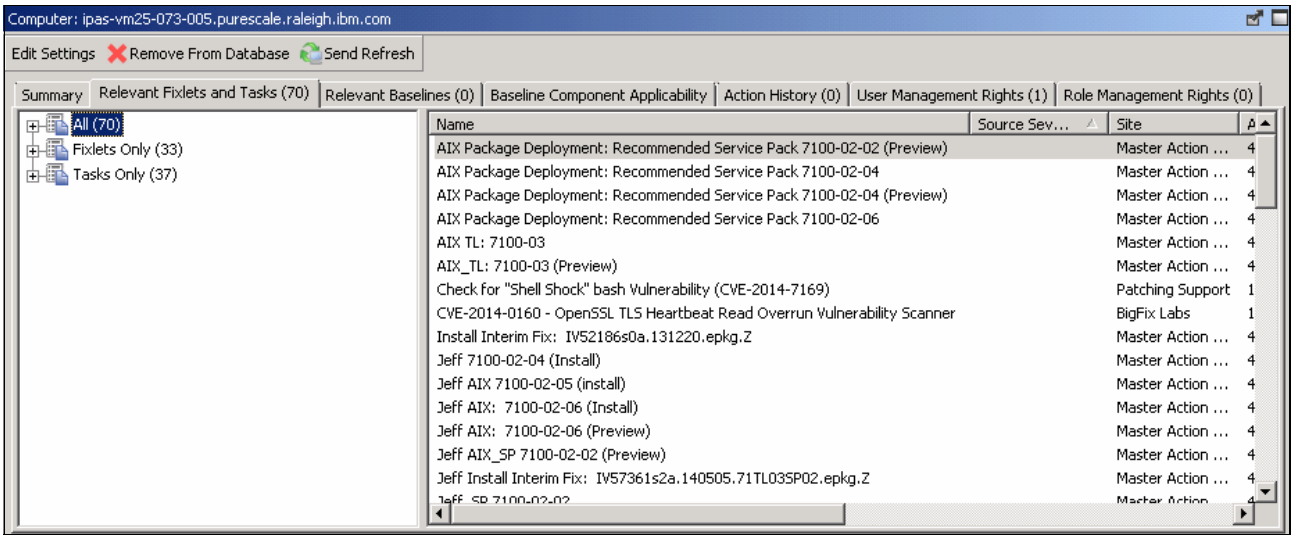


Figure 7-47 VM details

8

IBM middleware maintenance

This chapter describes how to apply software maintenance to IBM middleware on deployed pattern instances. For production workloads, this is an important scenario because continuously redeploying a newer version of a pattern is not always preferable or even possible.

Sometimes, IBM middleware maintenance can also be included in patterns. This approach ensures that newly deployed pattern instances come with the correct IBM middleware fix packs installed.

This chapter explains how fix packs can be applied to a select subset of IBM middleware that can be deployed on IBM PureApplication System or IBM PureApplication Software. The chapter focuses on a few key products, including IBM WebSphere Application Server, IBM DB2, and IBM MQ.

This chapter covers the following topics:

- IBM Installation Manager Repository
- Applying fix packs to Virtual System Pattern Instances
- Including Fix Packs in Virtual System Patterns

© Copyright IBM Corp. 2015. All rights reserved.

8.1 IBM Installation Manager Repository

One important aspect of IBM middleware maintenance on the PureApplication platform is that some products rely on an IBM Installation Manager Repository. Examples of such products include IBM WebSphere Application Server and IBM Business Process Manager (IBM BPM).

Before going into detail about how IBM Installation Manager Repository is implemented in PureApplication System, here is some relevant terminology:

- IBM Installation Manager Repository

  A repository that IBM Installation Manager uses to find the files for installing, modifying, rolling back, updating, or uninstalling certain IBM products. The repositories are specified in the Installation Manager preferences by a URL containing the `repository.config` file. When you download a product from IBM Passport Advantage or IBM Fix Central, the extracted installation files contain the `repository.config` file for the product.

- Composite IBM Installation Manager Repository

  A repository that in turn references one or more other Installation Manager repositories. By simply editing the `repository.config` file of the composite repository, you can quickly and easily change which products and versions the repository includes.

- IBM Packaging Utility

  A no-cost companion tool for IBM Installation Manager that you can use to create repositories containing one or more IBM products. The tool accepts web repositories or local repositories as input and it produces a (local) repository as output.

IBM PureApplication System V2.0 and higher come with a built-in (internal) IBM Installation Manager Repository. However, clients can configure PureApplication Systems to use an existing external IBM Installation Manager Repository. This chapter covers both scenarios in more detail in these sections:

- 8.1.1, “Default internal IBM Installation Manager Repository” on page 206
- 8.1.2, “Integrating with an external IBM Installation Manager Repository” on page 207

8.1.1 Default internal IBM Installation Manager Repository

By default, a built-in (internal) IBM Installation Manager Repository is installed and configured on PureApplication System V2.0 and higher. This is a composite IBM Installation Manager Repository that is within the PureApplication System environment; it is also known as the PureApplication IBM Installation Manager Repository. This internal IBM Installation Manager Repository can be used by IBM products that use IBM Installation Manager for product installation and maintenance.

Figure 8-1 on page 207 shows where to find the IBM Installation Manager Repository. On PureApplication System, the internal IBM Installation Manager Repository is automatically populated with a category (or location) called WebSphere.

Note: Not all IBM software uses Installation Manager products. For example, IBM DB2 and IBM MQ use a different installation mechanism.

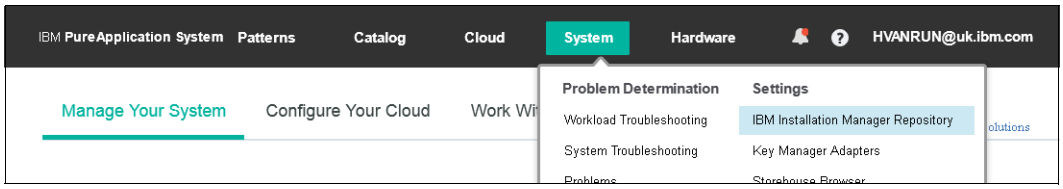


Figure 8-1 Go to the IBM Installation Manager Repository in PureApplication System

In general, the structure in the IBM Installation Manager Repository resembles Example 8-1.

Example 8-1 Structure of IBM Installation Manager Repository

```
--Category Information
   --Product Information
      --Product Version Information (Installation Package)
```

When examining the category WebSphere in the IBM Installation Manager Repository, you see several products, as shown in Figure 8-2.

Note: What is shown in Figure 8-2 is what is present in the Internal IBM Installation Manager Repository only. Even when you integrate with an External IBM Installation Manager Repository, this window shows only what is present in the Internal IBM Installation Manager Repository.

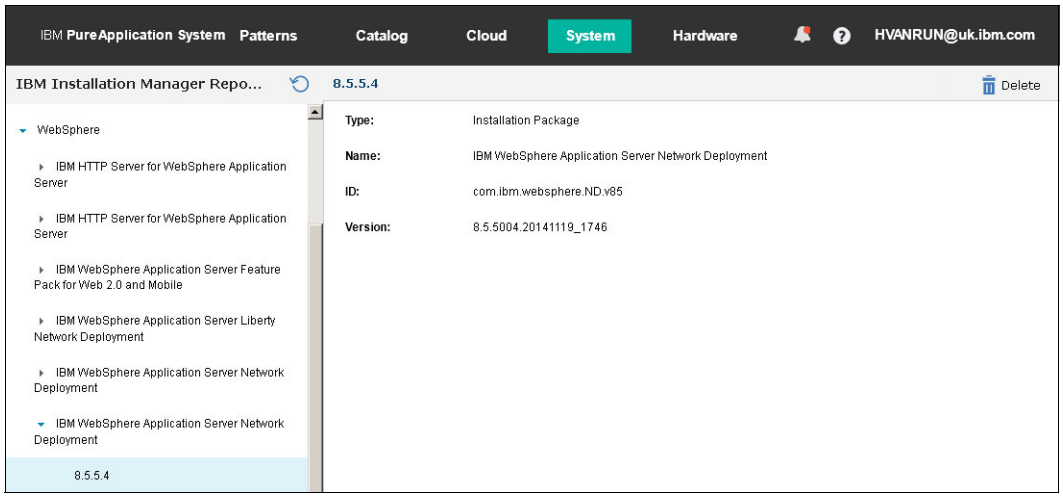


Figure 8-2 Examine the products and versions in the IBM Installation Manager Repository

8.1.2 Integrating with an external IBM Installation Manager Repository

Some organizations already have one or more IBM Installation Manager Repositories in place. PureApplication System supports integration with an existing IBM Installation Manager Repository. This external IBM Installation Manager Repository can be a composite repository, consisting of range of IBM products, versions, fix packs, and interim fixes. Configuring PureApplication System to use an existing external IBM Installation Manager Repository eliminates the need to keep the internal IBM Installation Manager Repository populated with the latest packages from IBM. This technique in turn helps reduce the operational effort to maintain IBM Installation Manager Repositories.

By default, PureApplication System uses its internal IBM Installation Manager Repository. Figure 8-3 shows the integration between PureApplication System and its IBM Installation Manager Repository, which is used in four cases:

- Importing Pattern Types
- Creating Patterns
- Deploying new Pattern Instances
- Applying maintenance to deployed Pattern Instances

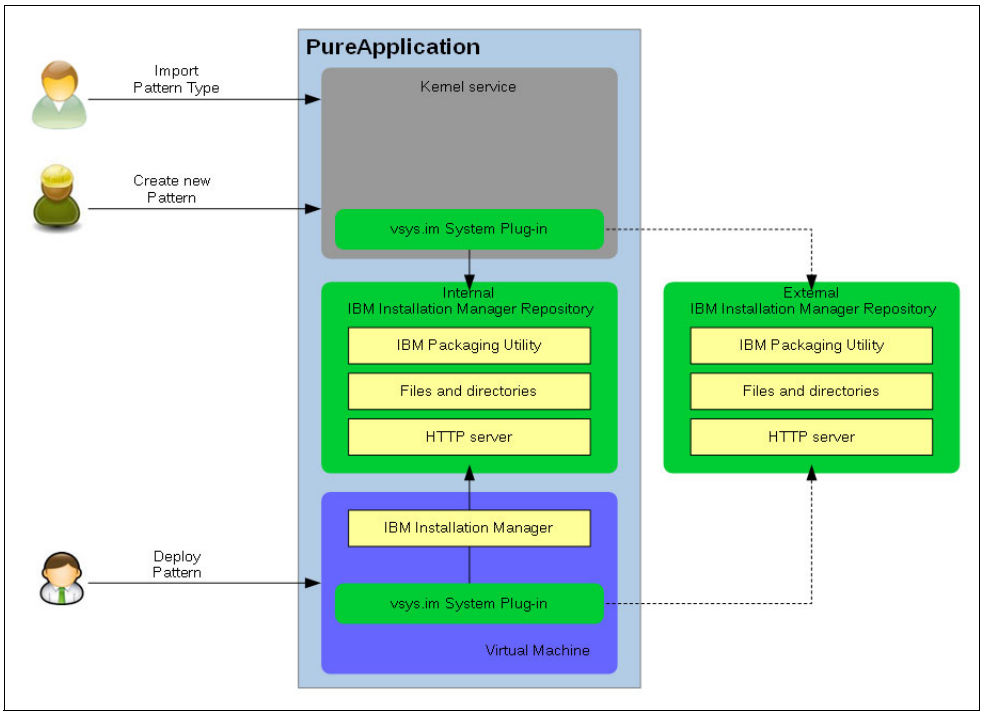


Figure 8-3 IBM Installation Manager architecture in PureApplication

Integration with an external IBM Installation Manager Repository comes with specific network connectivity requirements. Table 8-1 lists those requirements in more detail.

Table 8-1 List the connectivity requirements for integration with an external IBM Installation Manager Repository

Source	Destination (protocol)	Network
PureApplication System management nodes	External IBM Installation Manager Repository (HTTPS)	System Management Network
All deployed virtual machines (VMs)	External IBM Installation Manager Repository (HTTPS)	Data Networks

Assuming that the external IBM Installation Manager Repository is in place and all connectivity requirements are met, complete the following steps to configure PureApplication to use an external IBM Installation Manager Repository.

Note: PureApplication System either uses its internal IBM Installation Manager Repository or an external one. It does not support a hybrid configuration.

1. Click **Catalog** → **System Plug-ins** to find the System Plug-in `vsys.im`.
2. In the right pane, select **All** from the drop-down list and enter “vsys.im” as the filter.

3. Select the latest enabled version of the System Plug-in `vsys.im` (Figure 8-4).

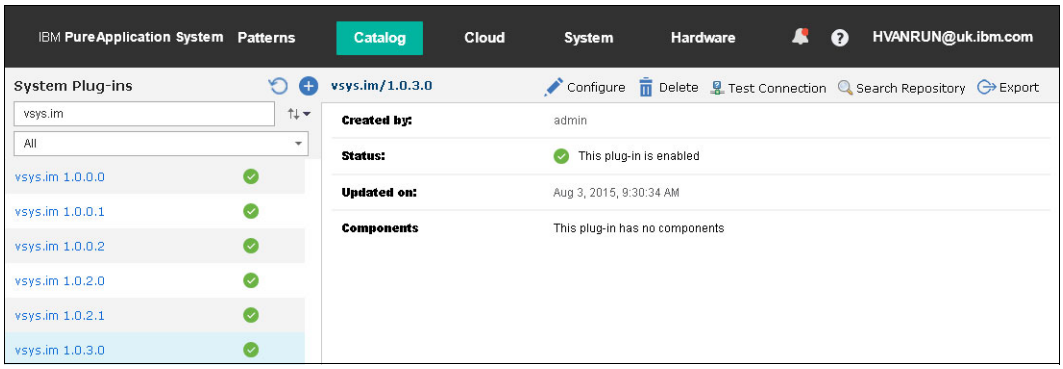


Figure 8-4 System Plug-ins

4. Click **Configure** to configure the external IBM Installation Manager Repository.
5. Enter the following information into the window that is shown in Figure 8-5 and click **Update**:
   - External Repository URL: `http://<your_repository_server>/some/url`

     Note: The URL provided here must point to the directory containing the `repository.config` file.

     Note: Do *not* use a URL that begins with `https`. SSL is a supported protocol, but you cannot import the SSL certificate of your repository into the SSL truststore. Without the certificate in the truststore, you get the following error when you click **Update**:

     ```
     CWZPL0376X: Failed to update the plug-in config due to the problem:
     System cannot connect to the external Installation Manager repository by
     using URL: https://<your_repository_server>/some/url
     ```

   - (optional) User ID: A user ID with the authorization to access your IBM Installation Manager Repository
   - (optional) Password: Password for the user ID

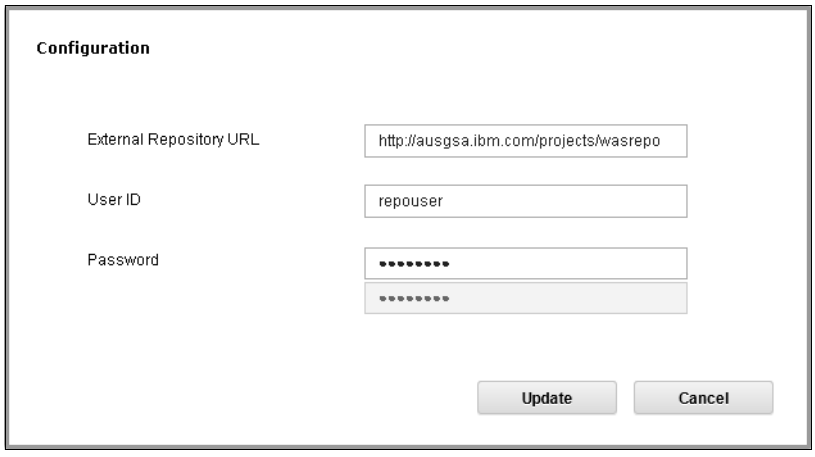


Figure 8-5 Enter details to configure integration with an external IBM Installation Manager Repository

Note: It can take a minute or so before the actual configuration of the System Plug-in `vsys.im` is updated. You can look at the field “Updated on” to confirm that it completed the update.

6. Click **Test Connection** to validate that a connection to your external IBM Installation Manager Repository can be established. When successful, you should see something similar to Figure 8-6.

Note: The connection test attempts to establish a connection to the external IBM Installation Manager Repository from one of the PureApplication management nodes. This action does not prove or test that a deployed VM can access the external IBM Installation Manager Repository.

Figure 8-6 Test the connection to the external IBM Installation Manager Repository

7. Log on to a deployed VM and perform a simple connectivity test to your external IBM Installation Manager Repository. Example 8-2 shows the result of a successful test by running **`curl`**.

Note: Network connectivity issues from a deployed VM to the external IBM Installation Manager Repository can be *hard* to debug. For more information, see the post on the dwAnswers forum at the following website:

`https://developer.ibm.com/answers/questions/196669/websphere-virtual-system-pattern-deployment-report.html#answer-196670`

Example 8-2 Test the connection to the external IBM Installation Manager Repository from a VM

```
[virtuser@ipas-lpar-9-3-171-43 ~]$ curl -u repouser
http://ausgsa.ibm.com/wasrepo/
Enter host password for user 'repouser':
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<HTML>
 <HEAD>
  <TITLE>Index of /projects/wasrepo</TITLE>
 </HEAD>
 <BODY>
<H1>Index of /projects/wasrepo</H1>
<PRE><IMG SRC="/icons/blank.gif" ALT="    "> <A HREF="?N=D">Name</A>
<A HREF="?M=A">Last modified</A>       <A HREF="?S=A">Size</A>  <A
HREF="?D=A">Description</A>
<HR>
<IMG SRC="/icons/back.gif" ALT="[DIR]"> <A HREF="/projects/">Parent
Directory</A>        06-Apr-2015 09:28       -
```

```
<IMG SRC="/icons/text.gif" ALT="[TXT]"> <A
HREF="repository.config">repository.config</A>       28-Jul-2015 09:32    7.8k
</PRE><HR>
<ADDRESS>Apache/1.3.42 Server at ausgsa.ibm.com Port 443</ADDRESS>
</BODY></HTML>
```

It is a good idea to look for specific a package in the external IBM Installation Manager Repository from the PureApplication System user interface. You can accomplish this task by completing the following steps:

1. Click **Catalog** → **System Plug-ins** and select the latest enabled version of the System Plug-in `vsys.im`.
2. Click **Search Repository** and under “Offering ID or Version”, enter (for example) **c**om.ibm.websphere.ND.v85_8.5.5. Click **OK** to perform the search and display all IBM WebSphere Network Deployment V 8.5.5 packages that are present in the external IBM Installation Manager Repository. Figure 8-7 shows the packages from this search. This action proves that there is not only connectivity, but also that certain packages can be found.

Note: When updating the External IBM Installation Manager Repository with new offerings, they might not be visible in PureApplication System right away. The list of offerings from the External IBM Installation Manager Repository is stored in a cache, which refreshes every 30 or 60 minutes.

You can force a refresh of the cache by completing the following steps:

a. Click **Catalog** → **System Plug-ins**.

b. Select the `vsys.im` System Plug-ins and click **Configure**.

c. Do not make any changes. Click **Update**.

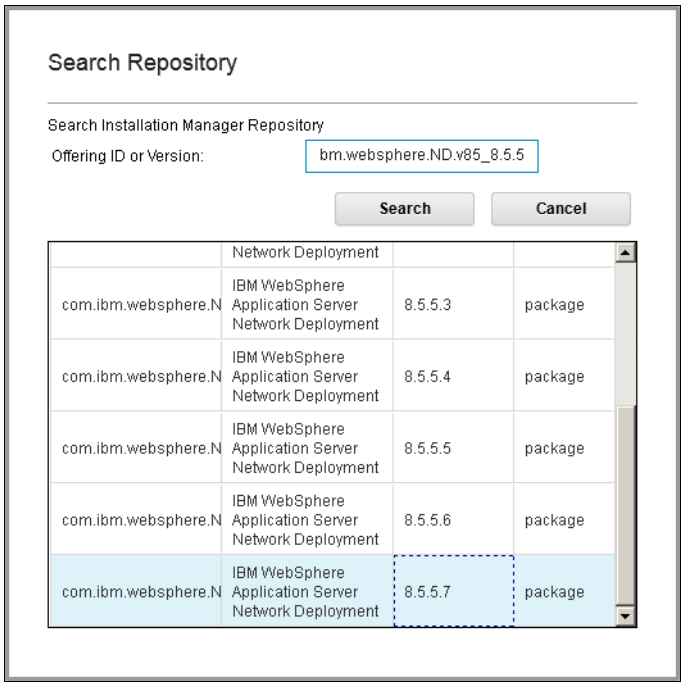


Figure 8-7 Search for packages in the external IBM Installation Manager Repository from PureApplication

3. Open the Pattern Editor and create a simple Virtual System Pattern (VSP) by using an IBM WebSphere Network Deployment V8.5.5 Software Component. Figure 8-8 shows all fix packs that are listed in the search results (previously shown in Figure 8-7 on page 211) as now available.

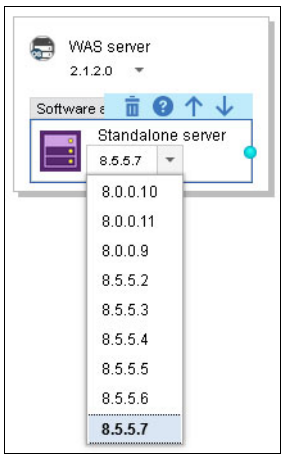


Figure 8-8 All WebSphere V8.5.5 fix packs from the external IBM Installation Manager Repository are available

4. To prove that the WebSphere Application Server Version 8.5.5 fix packs are not retrieved from the internal IBM Installation Manager Repository, quickly review the fix packs present by clicking **System** → **Installation Manager Repository**, as shown in Figure 8-9. With only Fix Pack 4 listed, this proves that the other fix packs that are available in the Pattern Editor are indeed obtained from the external IBM Installation Manager Repository.

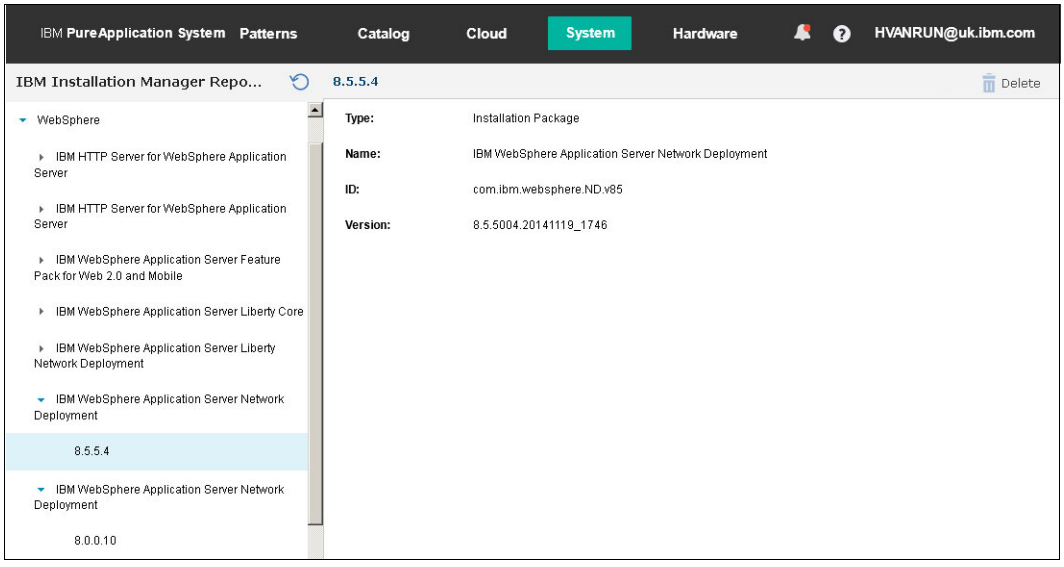


Figure 8-9 WebSphere V 8.5.5 Fix Packs present in internal IBM Installation Manager Repository

The following steps complete the configuration of an external IBM Installation Manager Repository in PureApplication System. For the remainder of this chapter, you work with the internal IBM Installation Manager Repository, meaning that the external IBM Installation Manager Repository is disabled.

1. Click **Catalog** → **System Plug-ins** and select the latest enabled version of the System Plug-in `vsys.im`.

2. Click **Configure** and clear all the fields in the dialog box that opens (Figure 8-10). Click **Update** to apply the changes.

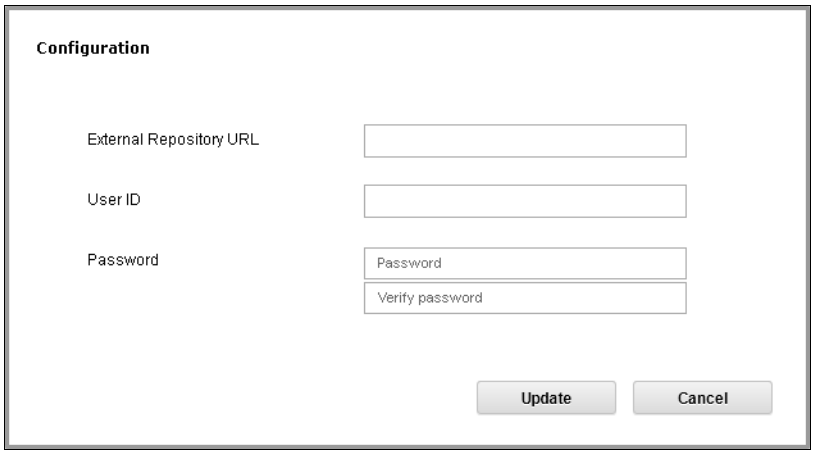


Figure 8-10 Leave all fields empty to disable the external IBM Installation Manager Repository

8.2 Applying fix packs to Virtual System Pattern Instances

This section demonstrates how to apply fix packs to IBM middleware running on *deployed* VSP Instances. As stated at the start of this chapter, this chapter covers a few IBM middleware products.

Table 8-2 lists the products that covered in this section. The exact mechanism and steps to apply a fix pack differ for each of the three IBM middleware products that are listed.

Table 8-2 Apply fix packs to various IBM middleware products on PureApplication System

IBM middleware product	Uses IBM Installation Manager	Uses PureApplication MAINTENANCE operation
IBM WebSphere Application Server 8.5.5	Yes	Yes
IBM DB2 10.5	No	Yes
IBM MQ V8.0	No	No

8.2.1 IBM WebSphere Application Server V8.5.5

In this section, you work with the internal IBM Installation Manager Repository where applicable. The instructions to upload a new WebSphere Application Server V 8.5.5 fix pack were written with this in mind. All other steps are transparent, so it does not make a difference whether you are using an internal or external IBM Installation Manager Repository.

Deploying a simple WebSphere Application Server 8.5.5 Virtual Server Pattern

Started by deploying a simple VSP with a single VM. On this VM, deploy the WebSphere Application Server Software Component for a stand-alone server. This software component corresponds to the Pattern Type “IBM WebSphere Application Server Network Deployment Patterns” Version 1.0.0.4, as shown in Figure 8-11.

Note: By default, PureApplication System Version 2.1.0.1 comes with “IBM WebSphere Application Server Network Deployment Patterns” pattern type Version 1.0.0.3. Version 1.0.0.4 can be downloaded separately. For more information, go to the following website:

https://www.ibm.com/developerworks/community/blogs/explorepatterns/entry/WebSphere_Application_Server_1_0_0_4_PureApplication_pattern_types_are_now_available?lang=en

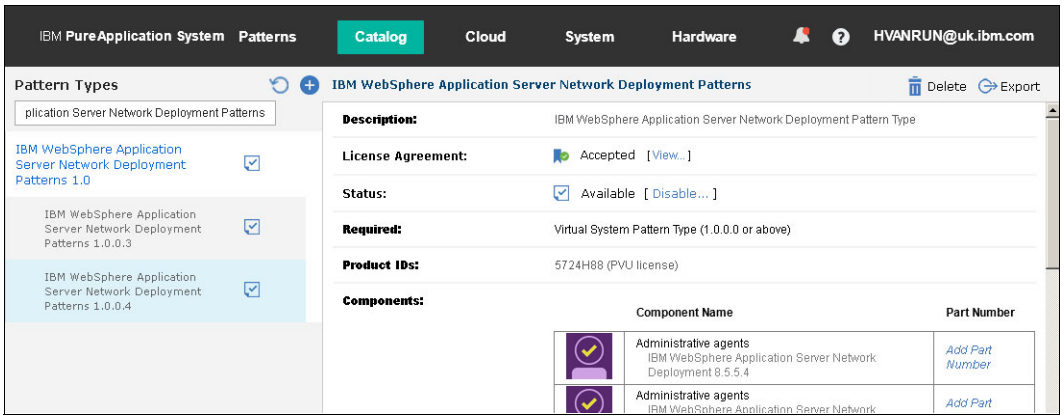


Figure 8-11 IBM WebSphere Application Server Network Deployment Patterns Version 1.0.0.4

The version of WebSphere Application Server Version 8.5.5 that is deployed by the Software Components of the pattern type depends on what is available from the IBM Installation Manager Repository. The most recent version of WebSphere Application Server loaded in the internal IBM Installation Manager Repository in PureApplication System V2.1 is V8.5.5.4. This statement is confirmed by the version in the drop-down box of the Software Component (Figure 8-12).

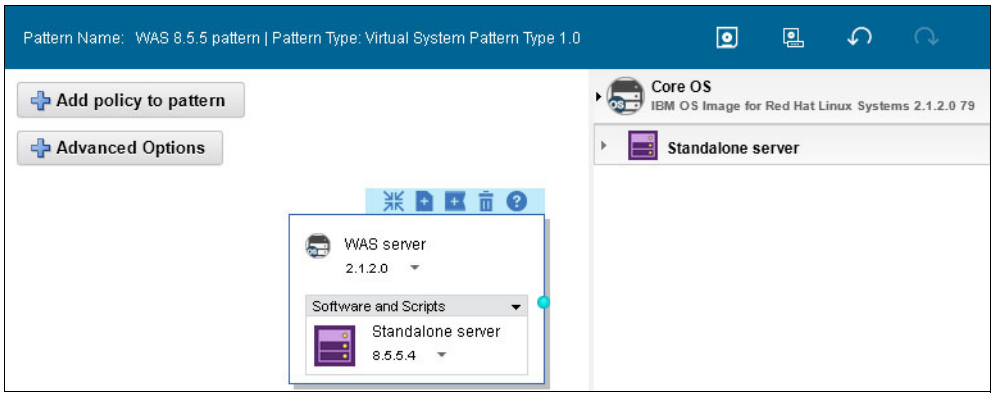


Figure 8-12 Simple Virtual System Pattern that uses WebSphere Application Server V 8.5.5 Fix Pack 4

After it is deployed, you can quickly log on to the VM and validate the version of WebSphere Application Server V8.5.5 that is installed. Log on as the WebSphere Application Server administrator and run **versionInfo.sh** (Example 8-3).

Example 8-3 Determine the exact version of WebSphere Application Server by running versionInfo.sh

```
[virtuser@ipas-lpar-9-3-171-18 bin]$ ./versionInfo.sh
WVER0010I: Copyright (c) IBM Corporation 2002, 2012; All rights reserved.
WVER0012I: VersionInfo reporter version 1.15.1.48, dated 2/8/12

--------------------------------------------------------------------------------
IBM WebSphere Product Installation Status Report
--------------------------------------------------------------------------------

Report at date and time July 30, 2015 3:33:20 PM UTC

Installation
--------------------------------------------------------------------------------
Product Directory        /opt/IBM/WebSphere/AppServer
Version Directory        /opt/IBM/WebSphere/AppServer/properties/version
DTD Directory            /opt/IBM/WebSphere/AppServer/properties/version/dtd
Log Directory            /home/virtuser/var/ibm/InstallationManager/logs

Product List
--------------------------------------------------------------------------------
HV                       installed
ND                       installed

Installed Product
--------------------------------------------------------------------------------
Name            IBM WebSphere Application Server Network Deployment Patterns
Version         1.0.0.4
ID              HV
Build Level     v10X_1524.01
Build Date
Architecture    x86-64 (64 bit)

Installed Product
--------------------------------------------------------------------------------
Name                  IBM WebSphere Application Server Network Deployment
Version               8.5.5.4
ID                    ND
Build Level           cf041446.03
Build Date            11/19/14
Package               com.ibm.websphere.ND.v85_8.5.5004.20141119_1746
Architecture          x86-64 (64 bit)
Installed Features    IBM 64-bit WebSphere SDK for Java
                      WebSphere Application Server Full Profile
                      EJBDeploy tool for pre-EJB 3.0 modules
                      Embeddable EJB container
                      Stand-alone thin clients and resource adapters
Optional Languages    German
                      Russian
                      Korean
                      Brazilian Portuguese
                      Italian
```

```
                    French
                    Hungarian
                    Spanish
                    Simplified Chinese
                    Czech
                    Traditional Chinese
                    Japanese
                    Polish
                    Romanian

--------------------------------------------------------------------------------
End Installation Status Report
--------------------------------------------------------------------------------
```

You can log on to the WebSphere Integrated Solutions Console by using a browser and validate the WebSphere Application Server version that is installed, as shown in Figure 8-13.

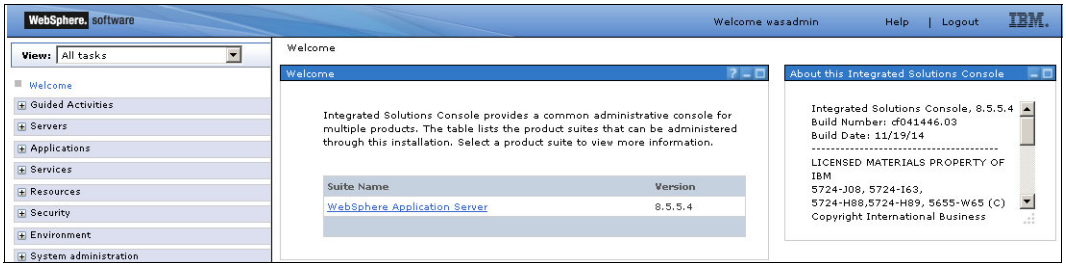


Figure 8-13 Confirm the deployed version of WebSphere Application Server

Figure 8-13 shows that WebSphere Application Server V8.5.5 Fix Pack 4 is installed.

Downloading WebSphere Application Server V8.5.5 Fix Pack 5 from IBM Fix Central

In this example, you upgrade to WebSphere Application Server V8.5.5 Fix Pack 5. Before you can do so, you must download the corresponding fix pack from IBM Fix Central. The instructions include additional details to ensure that you can quickly find the files that you need. Complete the following steps:

1. Open your browser, go to Fix Central at the following website, and log on by using your IBM ID:

   https://www.ibm.com/support/fixcentral/

2. Enter the following details to find applicable fixes for WebSphere Application Server V8.5.5.4 (Figure 8-14 on page 217). Click **Continue**.
   - Product selector: WebSphere Application Server
   - Installed Version: 8.5.5.4
   - Platform: Linux 64-bit/x86-64

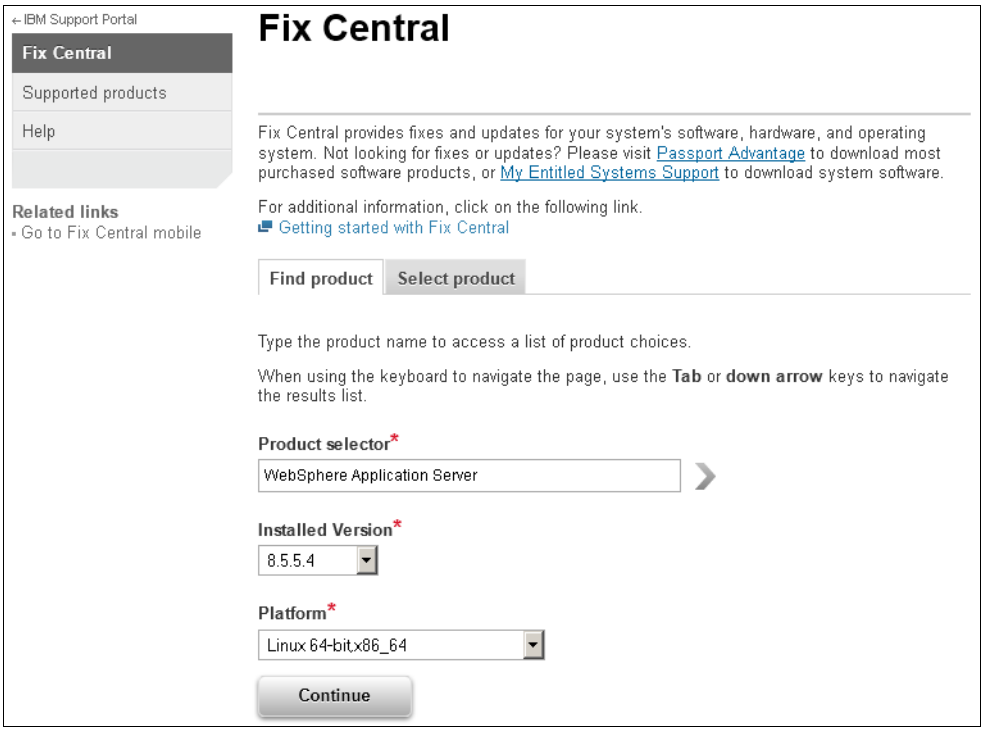


Figure 8-14 Enter information to find applicable fixes for WebSphere Application Server 8.5.5.

3. Optionally, narrow down the list of fixes by selecting the following options (Figure 8-15):
   - Fix status: Available
   - And Platform: Linux 64-bit/x86-64
   - And Applies to: 8.5.5.4
   - And Fix type: Fix pack

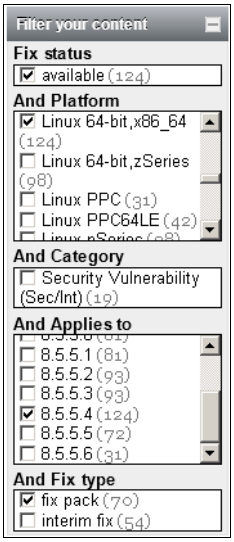


Figure 8-15 Narrow down the list of fixes on Fix Central

4. Select all the files that are listed in Table 8-3. Click **Continue**.

Table 8-3 Files for WebSphere Application Server V8.5.5 Fix Pack 5 to be downloaded from IBM Fix Central

File	Size in bytes	Comments
8.5.5-WS-WAS-FP0000005-part1.zip	897952246	Needs to be combined into a single .tgz file.
8.5.5-WS-WAS-FP0000005-part2.zip	1725356354	
8.5.5-WS-WCT-FP0000005-part1.zip	155127498	Needs to be combined into a single .tgz file.
8.5.5-WS-WCT-FP0000005-part2.zip	1588735641	
7.0.8.10-WS-IBMWASJAVA-part1.zip	656710627	Needs to be combined into a single .tgz file.
7.0.8.10-WS-IBMWASJAVA-part2.zip	1615232795	
7.1.2.10-WS-IBMWASJAVA-part1.zip	693563888	Needs to be combined into a single .tgz file.
7.1.2.10-WS-IBMWASJAVA-part2.zip	1299299840	
8.5.5-WS-WASSupplements-FP0000005-part1.zip	1052850261	Needs to be combined into a single .tgz file.
8.5.5-WS-WASSupplements-FP0000005-part2.zip	1539822787	
8.5.5-WS-LIBERTYPROFILE-FP0000005.zip	696284316	Needs to be converted to a .tgz file.

5. Table 8-3 states that most of the archive files must be combined into a single .tgz file. Example 8-4 shows how to combine the designated files into a single file.

Example 8-4 Combine .zip files into a single .tgz file

```
-bash-4.1# ls -rtl 8.5.5-WS-WAS-FP0000005*.zip
-rw-r--r-- 1 root root  897952246 Feb 27 18:45 8.5.5-WS-WAS-FP0000005-part1.zip
-rw-r--r-- 1 root root 1725356354 Feb 27 19:05 8.5.5-WS-WAS-FP0000005-part2.zip
-bash-4.1# mkdir 8.5.5-WS-WAS-FP0000005
-bash-4.1# cd 8.5.5-WS-WAS-FP0000005
-bash-4.1# unzip ../8.5.5-WS-WAS-FP0000005-part1.zip
-bash-4.1# unzip ../8.5.5-WS-WAS-FP0000005-part2.zip
-bash-4.1# tar -cvzf ../8.5.5-WS-WAS-FP0000005.tgz *
-bash-4.1# cd ..
-bash-4.1# ls -rtl 8.5.5-WS-WAS-FP0000005*.tgz
-rw-r--r-- 1 root root 2621612238 Jul 29 16:23 8.5.5-WS-WAS-FP0000005.tgz
```

Note: The capacity of the Internal IBM Installation Manager Repository on PureApplication System is limited. You can build your own repositories by using the IBM Packaging Utility. For more information, see the following website:

https://www.ibm.com/support/knowledgecenter/SSSH27_8.0.1/com.ibm.rational.clearcase.cc_ms_install.doc/topics/c_packaging_intro_rsar.htm

You still must build a .tgz file that you can import by using the PureApplication System CLI, as described in “Uploading WebSphere Application Server V8.5.5 Fix Pack 5 files to an Internal IBM Installation Manager Repository” on page 219.

For more information how to build WebSphere Application Server Patterns with IBM Installation Manager repositories, see the following website:

http://www.ibm.com/support/knowledgecenter/#!/SSAJ7T_1.0.0/com.ibm.websphere.waspatt20nd.doc/ae/tins_patternsB_configuring.html

Uploading WebSphere Application Server V8.5.5 Fix Pack 5 files to an Internal IBM Installation Manager Repository

Now, you are ready to upload the `.tgz` files containing the WebSphere Application Server V8.5.5 Fix Pack 5 files to the Internal IBM Installation Manager Repository on PureApplication. Log on to the PureApplication web console and complete the following steps.

Note: Make sure that the user has the role “Workload resources administration - Manage Workload resources (Full permission)” assigned. This designation is *required* to upload packages to the Internal IBM Installation Manager Repository on PureApplication System.

Without this role, the **`deployer.imrepositories.uploadPackageFromLocal`** command that you use here results in the following error message:

```
IOError: User is not authorized to perform this operation. You must have
APPLIANCE_ADMIN permission.
```

1. Click **System** → **IBM Installation Manager Repository**, as shown in Figure 8-16.

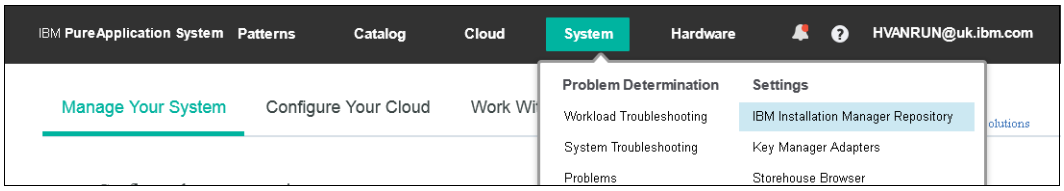


Figure 8-16 Go to the Internal IBM Installation Manager Repository on PureApplication

2. Confirm the name of the category in the Internal IBM Installation Manager Repository for WebSphere. The category name for this is WebSphere by default, as shown in Figure 8-17.

Note: The category WebSphere contains the offering ID files for “IBM WebSphere Application Server Network Deployment” V8.5.5.4 and V8.5.0.0. So, when importing a Fix Pack for WebSphere Application Server V8.5.5, you must use the same category of WebSphere. Importing the fix pack into its own category does not work.

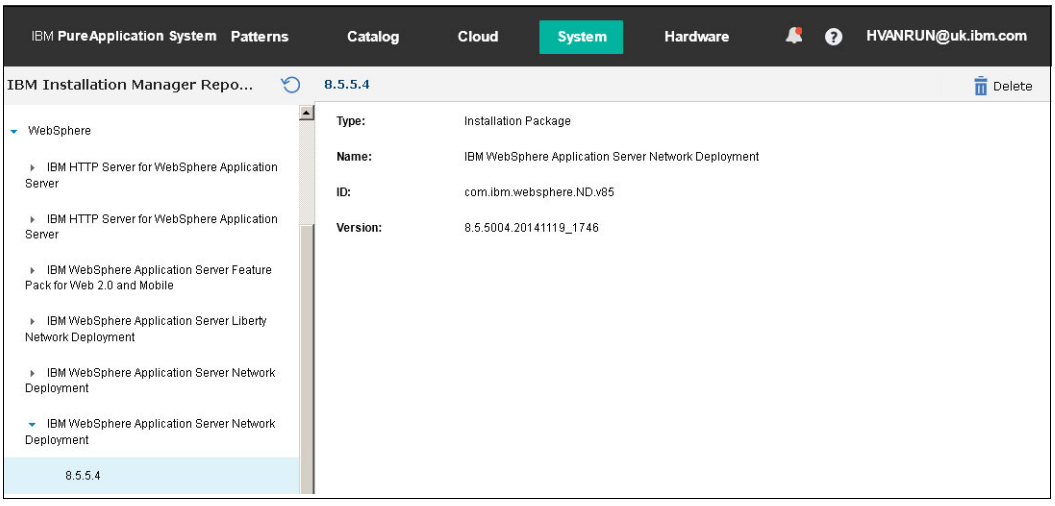


Figure 8-17 Confirm the name of the location in the IBM Installation Manager Repository

3. Use the PureApplication System command-line interface (CLI) to import the WebSphere Application Server V8.5.5 Fix Pack 5 files. Start with the `8.5.5-WS-WAS-FP0000005.tgz` file and run the command that is shown in Example 8-5 to import it. Here, `WebSphere` denotes the category in the Internal IBM Installation Manager Repository where the file is imported.

Example 8-5 Upload WebSphere Application Server V8.5.5 Fix Pack 5 to the built-in repository

```
>>>
deployer.imrepositories.uploadPackageFromLocal('WebSphere','/repository/8.5.5-W
S-WAS-FP0000005.tgz')
{'success': ['com.ibm.websphere.BASE.v85_8.5.5005.20150220_0158',
'com.ibm.websphere.BASETRIAL.v85_8.5.5005.20150220_0158',
'com.ibm.websphere.DEVELOPERS.v85_8.5.5005.20150220_0158',
'com.ibm.websphere.DEVELOPERSILAN.v85_8.5.5005.20150220_0158',
'com.ibm.websphere.EXPRESS.v85_8.5.5005.20150220_0158',
'com.ibm.websphere.EXPRESSTRIAL.v85_8.5.5005.20150220_0158',
'com.ibm.websphere.ND.v85_8.5.5005.20150220_0158',
'com.ibm.websphere.NDDMZ.v85_8.5.5005.20150220_0158',
'com.ibm.websphere.NDDMZTRIAL.v85_8.5.5005.20150220_0158',
'com.ibm.websphere.NDTRIAL.v85_8.5.5005.20150220_0158'], 'fail': []}
```

4. Repeat the previous step for all the `.tgz` files that were created earlier.
5. Confirm that all fixes for Fix Pack 5 are now visible in the IBM Installation Manager Repository on PureApplication System, as shown in Figure 8-18.

Note: Figure 8-18 on page 220 shows only the package for IBM WebSphere Application Server Network Deployment. However, other packages in the repository are updated in a similar manner and also show the presence of Version 8.5.5 Fix Pack 5.

Note: You can also explore the package that is present in the Internal IBM Installation Manager Repository by running **`deployer.imrepositories.listPackage()`**. However, this command returns all packages in JSON format, so it is not usable without a CLI script to parse or filter the data that is returned.

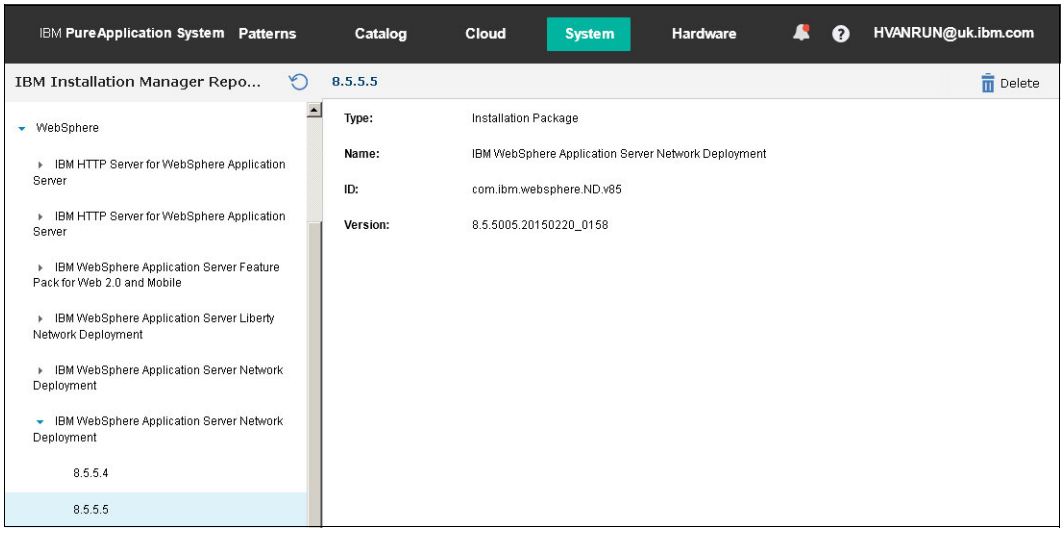


Figure 8-18 Confirm that the fixes are imported into the internal IBM Installation Manager Repository

Note: For information about obtaining files for WebSphere Application Server V8.5.5. Fix Pack 2 and WebSphere Application Server V8.0.0 Fix Pack 9 shipped with PureApplication System V2.0, see the blog post at the following website:

https://www.ibm.com/developerworks/community/blogs/explorepatterns/entry/Deploying_WAS_8552_or_8009_on_PureApplication_V21_or_higher?lang=en

Applying Fix Pack 5 to WebSphere Application Server V8.5.5 Pattern Instance

With the WebSphere Application Server V8.5.5 Fix Pack 5 loaded into the internal IBM Installation Manager Repository, you can apply it to a deployed WebSphere Application Server 8.5.5 pattern instance by completing the following steps:

1. Find the corresponding WebSphere Application Server V8.5.5 Virtual System Instance that was deployed earlier with Fix Pack 4. Click **Manage** (Figure 8-19), which opens the Instance Console.

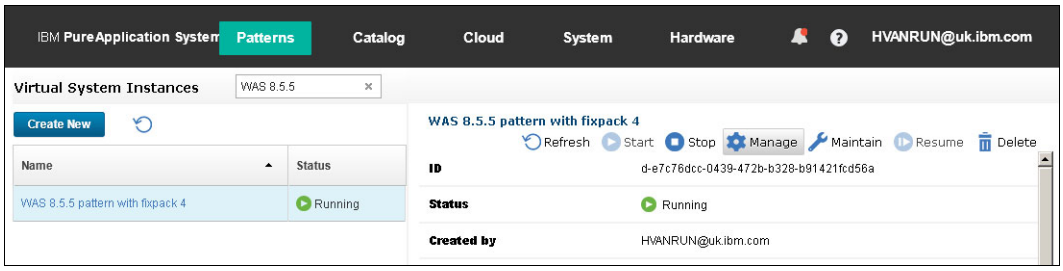


Figure 8-19 The deployed WebSphere Application Server V8.5.5 Virtual System Instance with Fix Pack 4

2. In the Instance Console, use the **Maintenance mode** slider (Figure 8-20) at the top to put the Pattern Instance into maintenance mode.
3. Click **Operations** and select the operation **MAINTENANCE** from the list. In the right pane, under “Fundamental”, open the “Maintenance fix packs” section (Figure 8-20).

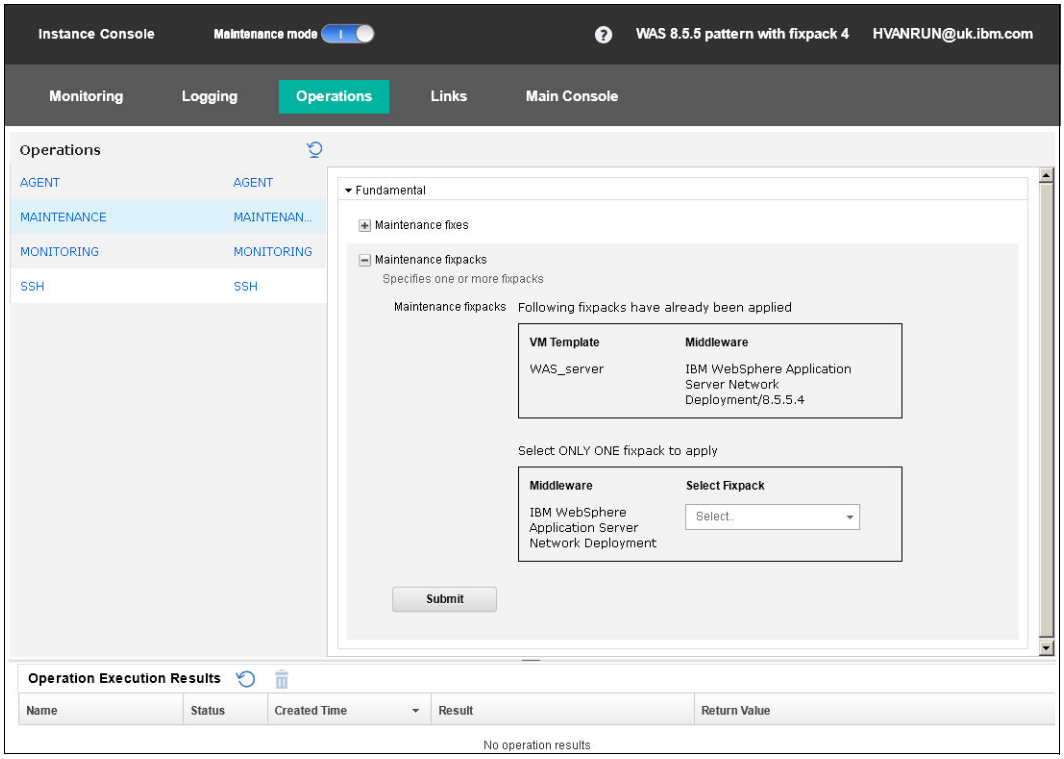


Figure 8-20 The MAINTENANCE operations that are available from the Instance Console

4. Under “Select Fix pack”, you should see 8.5.5.5 listed. Select it and click **Submit** (Figure 8-21).

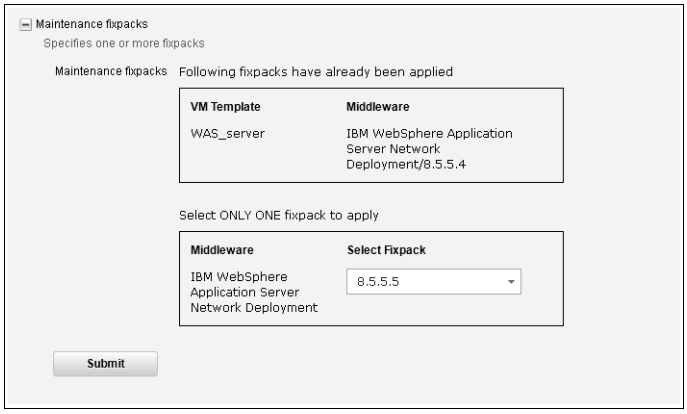


Figure 8-21 Apply 8.5.5.5 to IBM WebSphere Application Server Network Deployment

5. You see the operation listed under Operation Execution Results (Figure 8-22 on page 223).

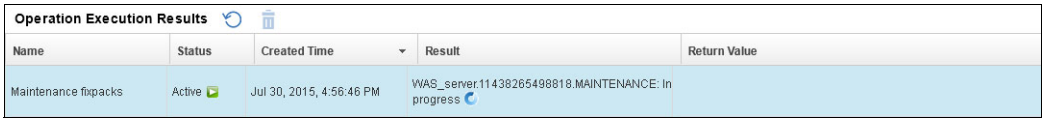


Figure 8-22 Operation "Maintenance fix packs" active

6. After a few minutes, the operation completes (Figure 8-23), which means that Fix Pack 5 was applied to WebSphere Application Server V8.5.5.

Figure 8-23 Operation "Maintenance fix packs" completed

7. Use the **Maintenance mode** slider (at the top of the Pattern Instance) to get out of maintenance mode.
8. To confirm that Fix Pack 5 is installed, log on to the WebSphere Integration Solutions Console (Figure 8-24).

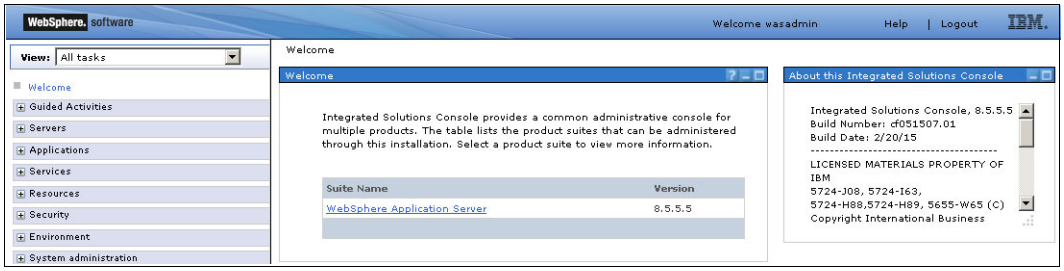


Figure 8-24 Confirm that WebSphere Application Server V8.5.5.5 is installed

9. You can also run **`versionInfo.sh`** (Example 8-6) to confirm that Fix Pack 5 is installed.

Example 8-6 Run versionInfo.sh to confirm that Fix Pack 5 is installed

```
[wasadmin@ipas-lpar-9-3-171-46 bin]$ ./versionInfo.sh
WVER0010I: Copyright (c) IBM Corporation 2002, 2012; All rights reserved.
WVER0012I: VersionInfo reporter version 1.15.1.48, dated 2/8/12

--------------------------------------------------------------------------------
IBM WebSphere Product Installation Status Report
--------------------------------------------------------------------------------

Report at date and time July 29, 2015 8:17:45 PM UTC

Installation
--------------------------------------------------------------------------------
Product Directory        /opt/IBM/WebSphere/AppServer
Version Directory        /opt/IBM/WebSphere/AppServer/properties/version
DTD Directory            /opt/IBM/WebSphere/AppServer/properties/version/dtd
Log Directory            /home/virtuser/var/ibm/InstallationManager/logs

Product List
--------------------------------------------------------------------------------
```

```
HV                      installed
ND                      installed

Installed Product
------------------------------------------------------------------------------
Name            IBM WebSphere Application Server Network Deployment Patterns
Version         1.0.0.4
ID              HV
Build Level     v10X_1524.01
Build Date
Architecture    x86-64 (64 bit)

Installed Product
------------------------------------------------------------------------------
Name                  IBM WebSphere Application Server Network Deployment
Version               8.5.5.5
ID                    ND
Build Level           cf051507.01
Build Date            2/20/15
Package               com.ibm.websphere.ND.v85_8.5.5005.20150220_0158
Architecture          x86-64 (64 bit)
Installed Features    IBM 64-bit WebSphere SDK for Java
                      WebSphere Application Server Full Profile
                      EJBDeploy tool for pre-EJB 3.0 modules
                      Embeddable EJB container
                      Stand-alone thin clients and resource adapters
Optional Languages    German
                      Russian
                      Korean
                      Brazilian Portuguese
                      Italian
                      French
                      Hungarian
                      Spanish
                      Simplified Chinese
                      Czech
                      Traditional Chinese
                      Japanese
                      Polish
                      Romanian

------------------------------------------------------------------------------
End Installation Status Report
------------------------------------------------------------------------------
```

10. Figure 8-25 shows a *snapshot* with a description of RM07370. Click **Delete** to delete this snapshot because you no longer need it; you confirmed that the Fix Pack is installed.

> **Note:** By default, PureApplication automatically takes a *snapshot* before applying a WebSphere Fix Pack through the MAINTENANCE operation. By restoring this snapshot, you can effectively roll back the installation of the fix pack.

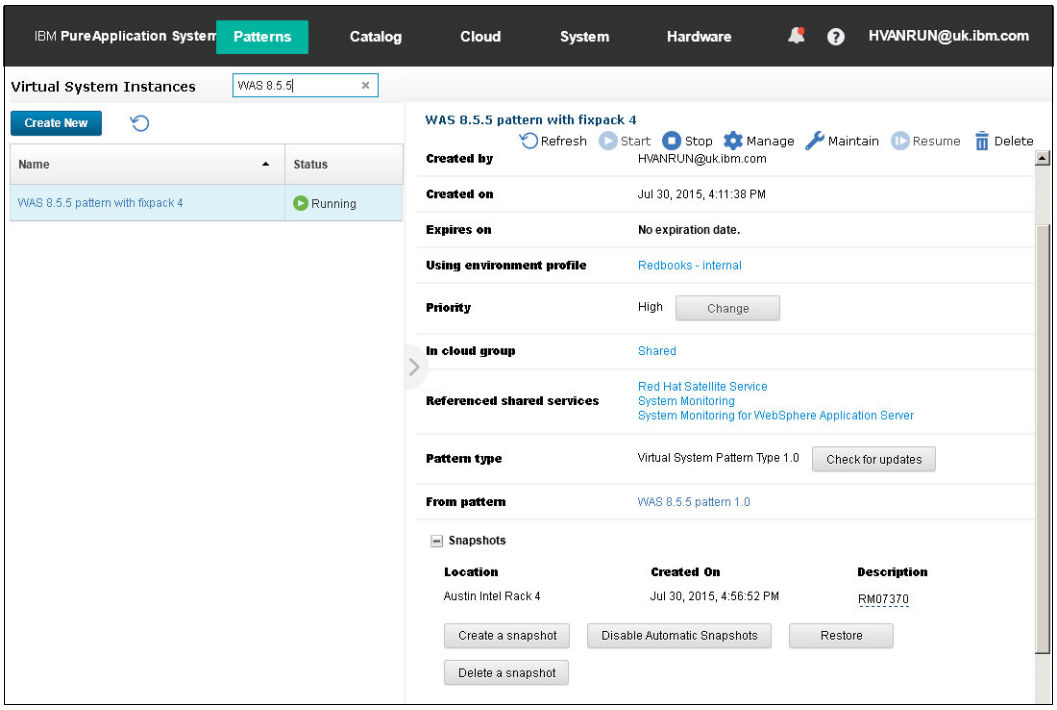


Figure 8-25 The snapshot that was automatically taken as part of the MAINTENANCE operation

8.2.2 IBM DB2 10.5

This section describes deploying a simple DB2 10.5 VSP and handling fix packs.

Deploying a simple DB2 10.5 Virtual System Pattern

Start by deploying a simple VSP with a single VM running DB2 10.5. The DB2 Software Component corresponds to the Pattern Type “IBM DB2 with BLU Acceleration Pattern”. Version 1.2.1.0 of this Pattern Type deploys DB2 10.5 Fix Pack 4, as shown in Figure 8-26. This version of the DB2 Pattern Type was released at the same time as PureApplication System V2.1.0.1.

Note: IBM released a new version of the “IBM DB2 with BLU Acceleration Pattern” on 26 July 2015. It comes with DB2 10.5 Fix Pack 5 preinstalled and can be downloaded from Fix Central. For more information, go to the following website:

https://www.ibm.com/developerworks/community/blogs/explorepatterns/entry/New_version_of_DB2_with_BLU_Acceleration_Pattern_includes_DB2_10_5_Fix_Pack_5?lang=en

In this example, it is not installed because it is not possible to demonstrate how to apply a more recent Fix Pack (DB2 10.5 Fix Pack 6 was not available at the time of writing).

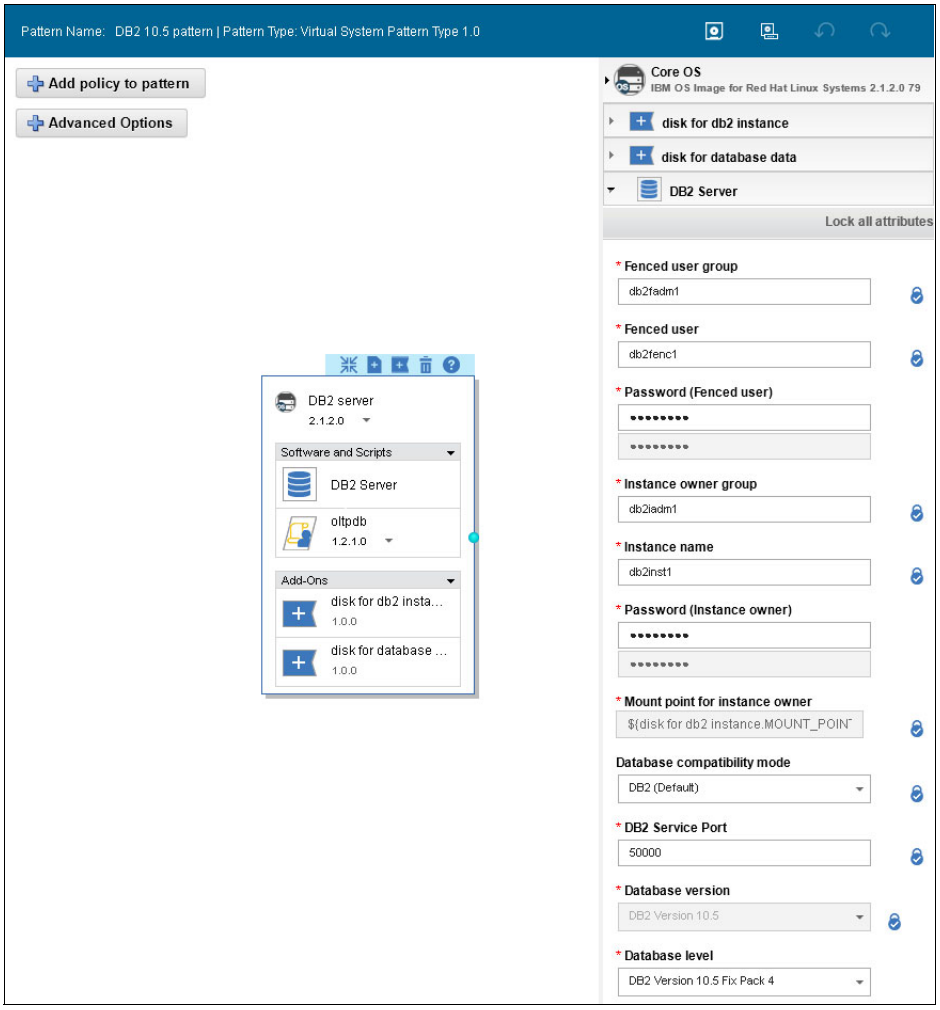


Figure 8-26 Simple Virtual System Pattern with DB2 10.5 Fix Pack 4

After the VSP is deployed, you can log on to the VM and validate the version of DB2 that is installed. Log on as the instance owner and run the command that is shown in Example 8-7.

Example 8-7 Determine the exact version of DB2 by running db2level

```
[db2inst1@ipas-lpar-9-3-171-44 ~]$ db2level
DB21085I  This instance or install (instance name, where applicable:
"db2inst1") uses "64" bits and DB2 code release "SQL10054" with level
identifier "0605010E".
Informational tokens are "DB2 v10.5.0.4", "special_33141", "IP23623_33141", and
Fix Pack "4".
Product is installed at "/opt/ibm/db2/V10.5".
```

Another way of confirming the DB2 version is to establish a connection to the DB2 database that is running on the VM. Using the same instance owner, connect to the instance, as shown in Example 8-8.

Example 8-8 Determine the exact version of DB2 by establishing a client connection

```
[db2inst1@ipas-lpar-9-3-171-44 ~]$ db2 connect to oltpdb

   Database Connection Information

 Database server        = DB2/LINUXX8664 10.5.4
 SQL authorization ID   = DB2INST1
 Local database alias   = OLTPDB

[db2inst1@ipas-lpar-9-3-171-44 ~]$ db2 connect reset
DB20000I  The SQL command completed successfully.
```

Both mechanisms confirm that DB2 10.5 Fix Pack 4 is installed.

Downloading DB2 10.5 Fix Pack 5 from IBM Fix Central

In this example, you upgrade to DB2 10.5 Fix Pack 5. Before you can do so, you must download the corresponding fix pack from IBM Fix Central. Details are included to help ensure that you can quickly find the file that you need. Complete the following steps:

1. Open your browser, go to Fix Central, which is found at the following website, and log on by using your IBM ID:

   https://www.ibm.com/support/fixcentral/

2. Enter the following information to find applicable fixes for DB2 10.5, as shown in Figure 8-27. Click **Continue**.
   - Product selector: DB2 for Linux, UNIX, and Windows
   - Installed Version: 10.5
   - Platform: Linux 64-bit/x86-64

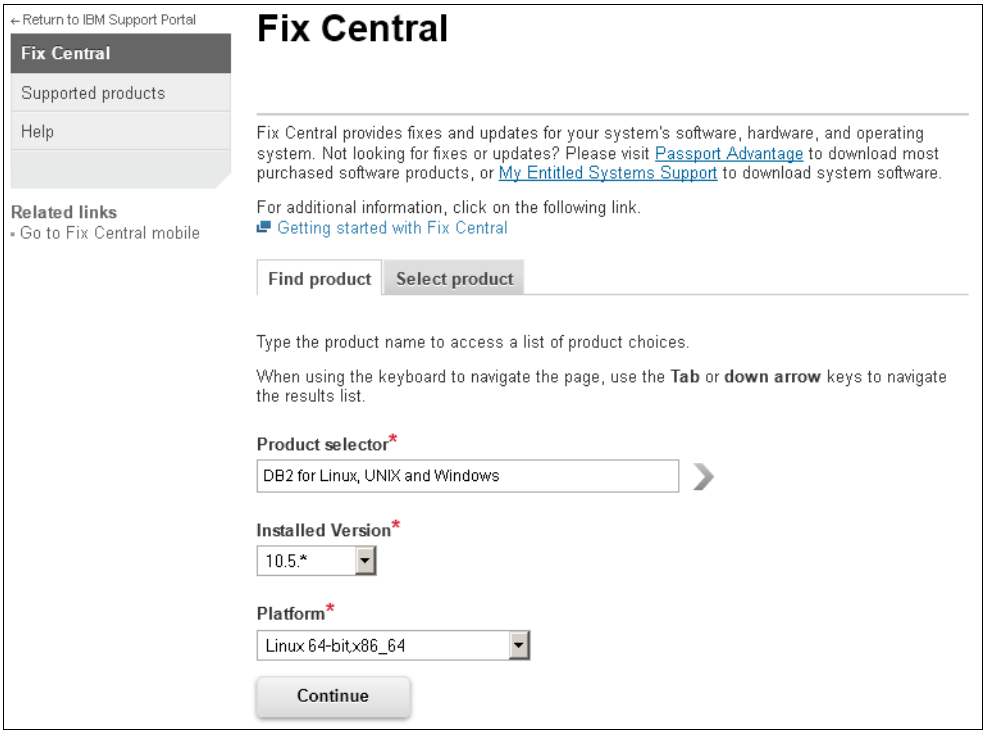


Figure 8-27 Enter information to find applicable fixes for DB2.

3. Optionally, narrow down the list of fixes by selecting theses options (see Figure 8-28):
   - Fix status: Available
   - And Platform: Linux 64-bit/x86-64
   - And Applies to: 10.5.0.4
   - And Fix type: Fix pack

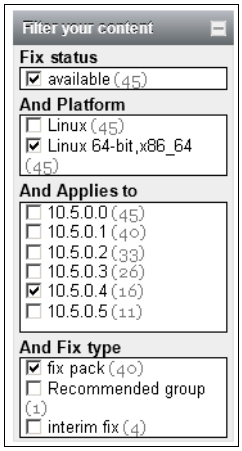


Figure 8-28 Narrow down the list of fixes on Fix Central

4. Find the `DB2-linuxx64-universal_fixpack-10.5.0.5-FP005` file, which corresponds to the DB2 Universal Fix Pack that is shown in Figure 8-29. Select the check box next to the fix and click **Continue**.

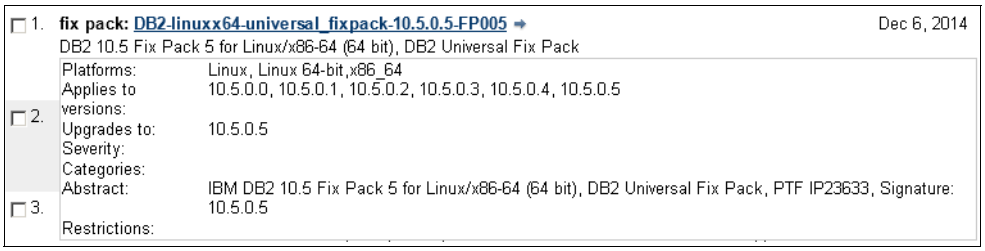


Figure 8-29 Find the fix for DB2 10.5 Fix Pack 5

5. You can now download the `v10.5fp5_linuxx64_universal_fixpack.tar.gz` file, as shown in Figure 8-30.

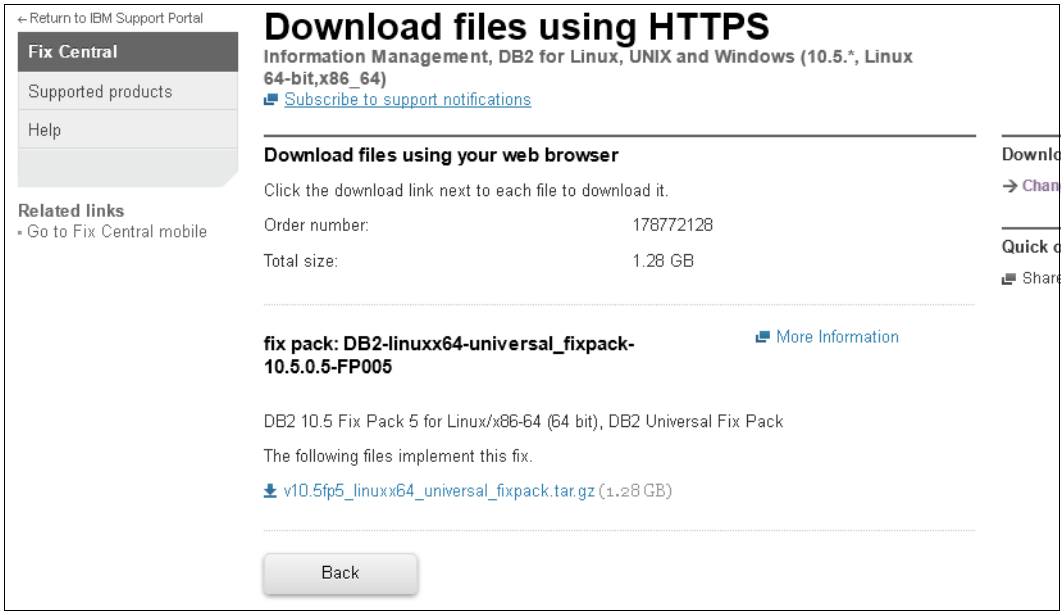


Figure 8-30 Download the DB2 Fix Pack from IBM Fix Central

6. Validate that the downloaded file is not corrupted by running the commands that are shown in Example 8-9. If the result is 0, the `.tar.gz` file is fine.

Example 8-9 Verify that the downloaded file is not corrupted

```
-bash-4.1# gunzip -c v10.5fp5_linuxx64_universal_fixpack.tar.gz | tar t >
/dev/null
-bash-4.1# echo $?
0
```

Now, you can upload the Fix Pack to the PureApplication System.

Uploading DB2 10.5 Fix Pack 5 to PureApplication System

To upload the fix pack to PureApplication System, log on to the PureApplication System web console and complete the following steps.

Note: Make sure that your user has sufficient permission to create new catalog content.

1. Click **Catalog** → **DB2 Fix Packs** (Figure 8-31).

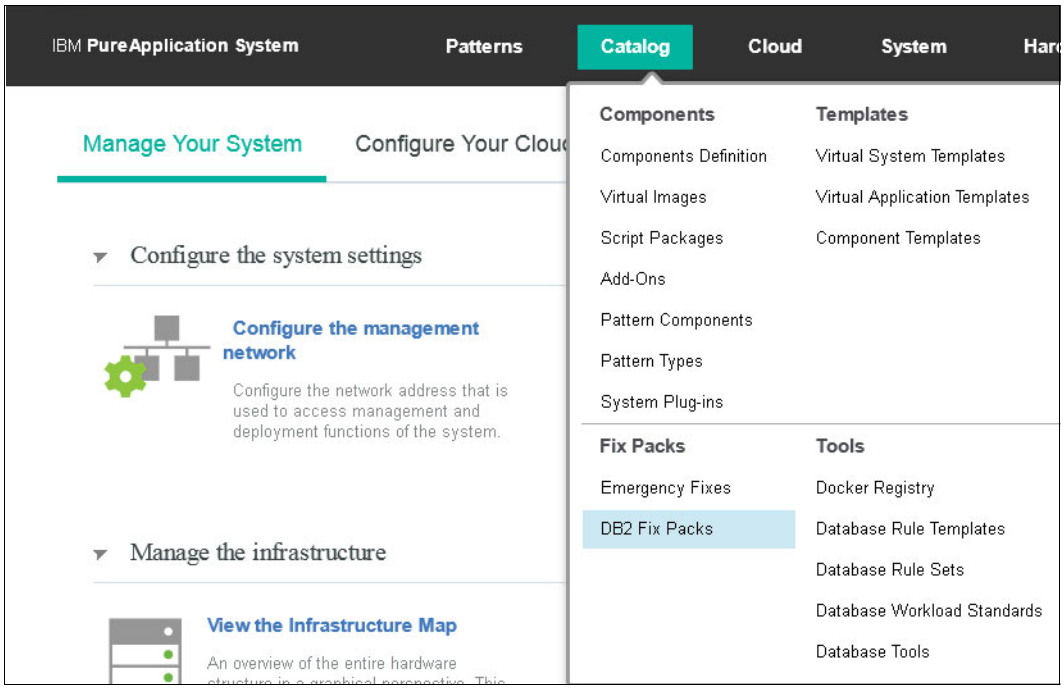


Figure 8-31 Go to DB2 Fix Packs in PureApplication System

2. Click the “+” icon to create a DB2 Fix Pack. Enter the following information in the dialog box that is shown in Figure 8-32. The information that is entered here is your choice, but it is preferable to use a sensible name and description. Click **Save** to create the entry.
   - DB2 Fix Pack name: DB2 10.5 Fix Pack 5
   - Description: DB2 10.5 Fix Pack 5 for Linux/x86-64 (64 bit)

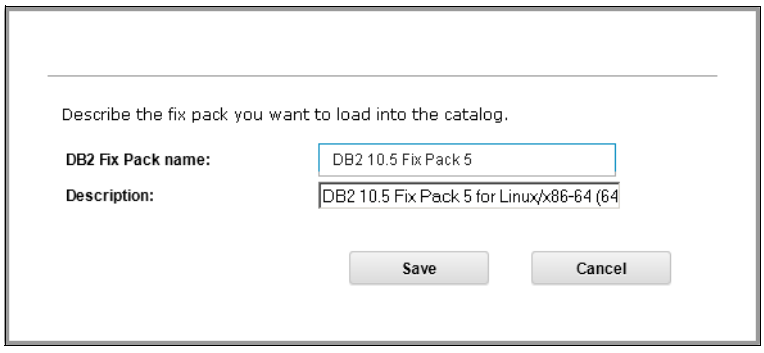


Figure 8-32 Create a DB2 Fix Pack entry in PureApplication System

3. With the new DB2 Fix Pack entry selected, click **Browse** and select the `v10.5fp5_linuxx64_universal_fixpack.tar.gz` file that you downloaded earlier.

4. Wait for the upload to complete (Figure 8-33).

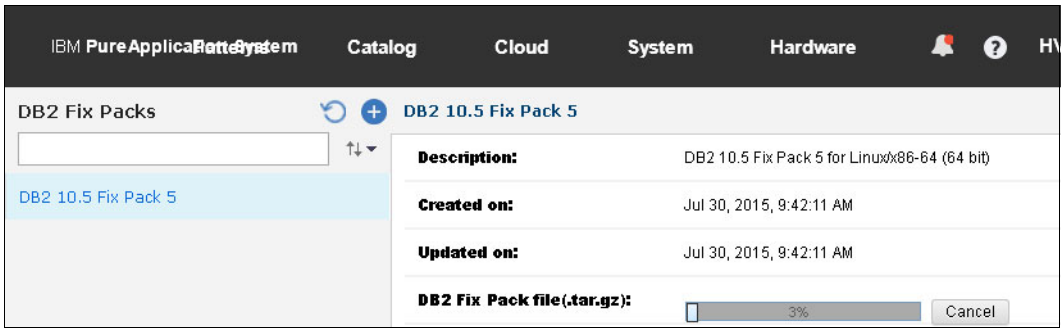


Figure 8-33 Upload the DB2 Fix Pack file

With the DB2 Fix Pack uploaded to the catalog in PureApplication Systems, you can apply it to a deployed DB2 pattern instance.

Applying Fix Pack 5 to a deployed DB2 10.5 pattern instance

To apply Fix Pack 5 to the deployed DB2 10.5 pattern instance, complete the following steps:

1. Find the corresponding DB2 10.5 Virtual System Instance that was deployed earlier with Fix Pack 4. Click **Manage** (Figure 8-34), which opens the Instance Console.

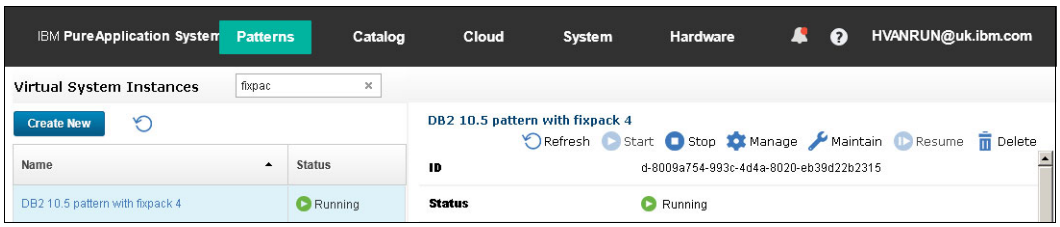


Figure 8-34 Find the DB2 10.5 Virtual System Instance

2. In the Instance Console, use the slide **Maintenance mode** at the top to put the Pattern Instance in maintenance mode.

3. Click **Operations** and select the operation **DB2_server.DB2_Server-Part** from the list. In the right pane, open the section **Apply a DB2 fix pack** that is listed under “Fundamental”, as shown in Figure 8-35.

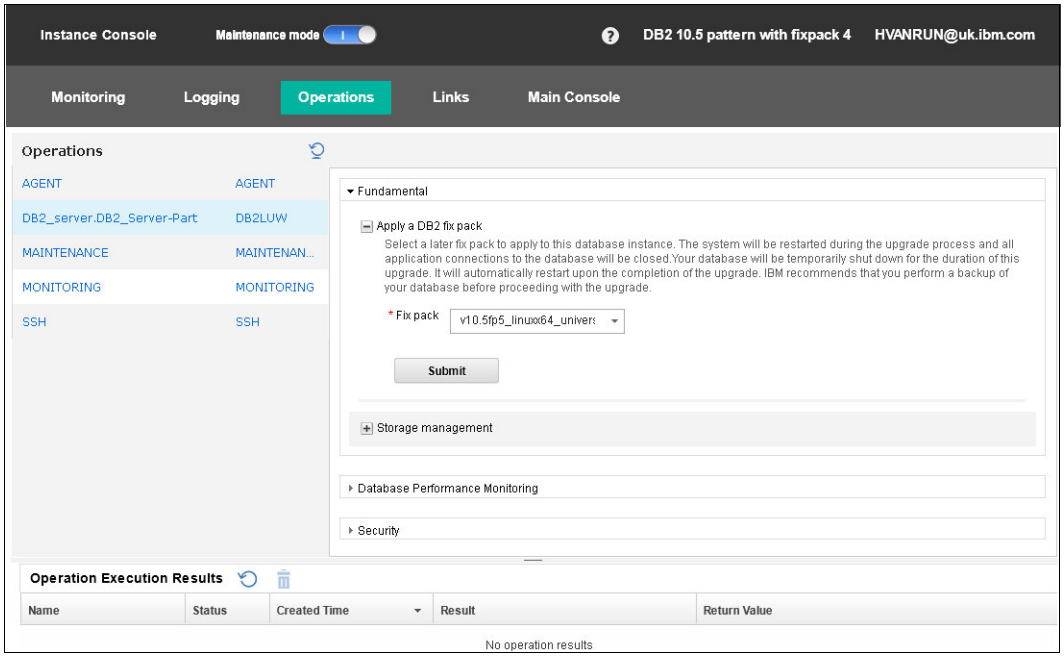


Figure 8-35 Instance Console to apply DB2 Fix Pack

4. Select `v10.5fp5_linuxx64_universal_fixpack.tar.gz` from the drop-down list and click **Submit**. This action starts the operation, which should appear as active under “Operation Execution Results” (Figure 8-36).

Figure 8-36 Operation “Apply DB2 fix pack” active

5. After a couple of minutes, the operation should complete, meaning that Fix Pack 5 now is applied to DB2 10.5 (Figure 8-37).

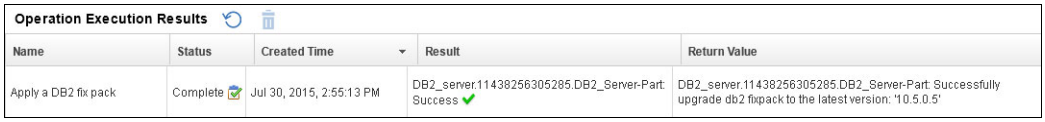


Figure 8-37 Operation “Apply DB2 fix pack” complete

6. Use the **Maintenance mode** slider at the top of the window to take the Pattern Instance out of maintenance mode.
7. Log on to the VM as the instance owner and run the command that is shown in Example 8-10 to confirm that Fix Pack 5 for DB2 10.5 is installed.

Example 8-10 Determine the exact version of DB2 by running db2level

```
[db2inst1@ipas-lpar-9-3-171-48 ~]$ db2level
DB21085I  This instance or install (instance name, where applicable:
```

```
"db2inst1") uses "64" bits and DB2 code release "SQL10055" with level
identifier "0606010E".
Informational tokens are "DB2 v10.5.0.5", "s141128", "IP23633", and Fix Pack
"5".
Product is installed at "/opt/ibm/db2/V10.5".
```

8.2.3 IBM MQ V8.0

In this section, you work with the Pattern Type “IBM MQ Advanced Virtual System Pattern Type” Version 1.0.0.0. Having this pattern installed (Figure 8-38) is a requirement for the scenario that is documented in this section.

Note: Unlike the WebSphere and DB2 Pattern Types, this Pattern Type does not come pre-entitled and preinstalled on PureApplication System. Only clients with entitlement for IBM MQ Advanced can download and install this Pattern Type.

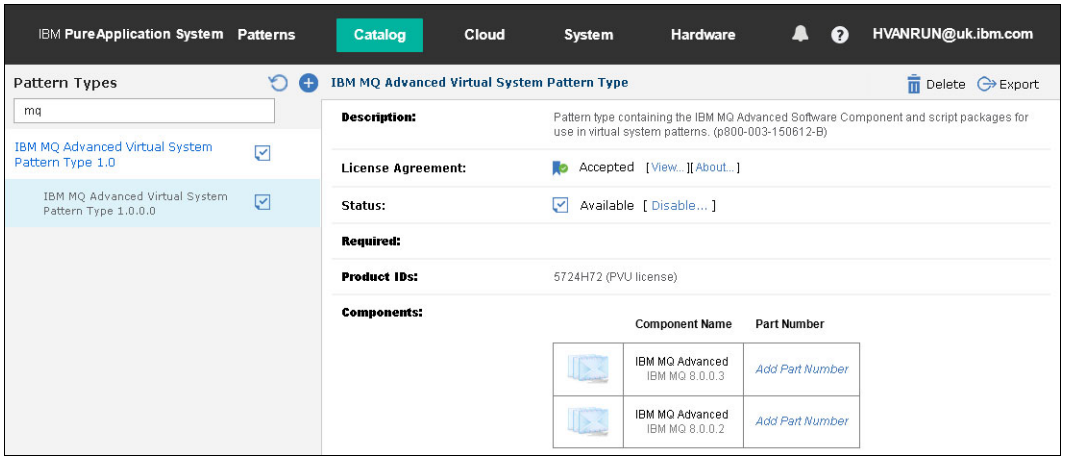


Figure 8-38 IBM MQ Advanced Virtual System Pattern Type 1.0.0.0

Deploying a simple IBM MQ V8.0 Virtual System Pattern

The goal in this section is to deploy a simple VSP with a single VM. On this VM, deploy the IBM MQ Advanced Software Component. This Software Component belongs to the Pattern Type “IBM MQ Advanced Virtual System Pattern Type” Version 1.0.0.0. Figure 8-39 shows what the VSP looks like.

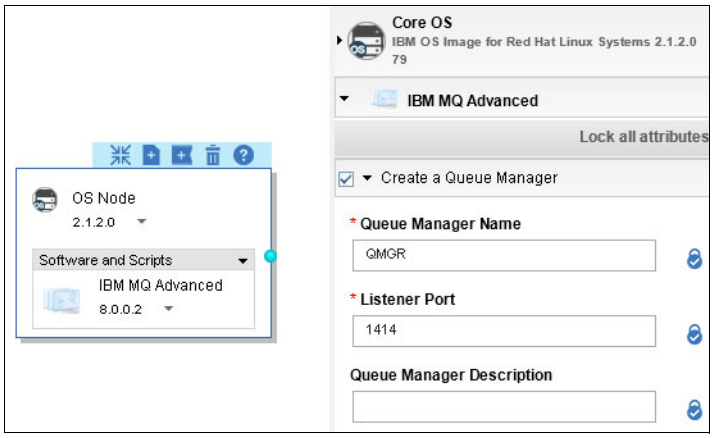


Figure 8-39 IBM MQ 8.0 Virtual System Pattern

After the pattern is deployed, you can log on to the VM and validate the version and Fix Pack of IBM MQ that is installed. Log on as IBM MQ administrator mqm and run the **dspmqver** command that is shown in Example 8-11. This command confirms that the VSP deployed IBM MQ V8.0 Fix Pack 2.

Example 8-11 Verify the version and Fix Pack of IBM MQ

```
-bash-4.1$ dspmqver
Name:        WebSphere MQ
Version:     8.0.0.2
Level:       p800-002-150217.2
BuildType:   IKAP - (Production)
Platform:    WebSphere MQ for Linux (x86-64 platform)
Mode:        64-bit
O/S:         Linux 2.6.32-504.16.2.el6.x86_64
InstName:    Installation1
InstDesc:
Primary:     Yes
InstPath:    /opt/mqm
DataPath:    /var/mqm
MaxCmdLevel: 801
LicenseType: Production
```

Applying Fix Pack 3 to a deployed IBM MQ V.0 Virtual System Instance

With the IBM MQ 8.0 Virtual System Instance deployed, you can apply Fix Pack 3 by completing the following steps:

1. Find the corresponding IBM MQ operation8.0 Virtual System Instance with Fix Pack 2 that you deployed earlier. Click **Manage** (Figure 8-40 on page 235) to open the Instance Console.

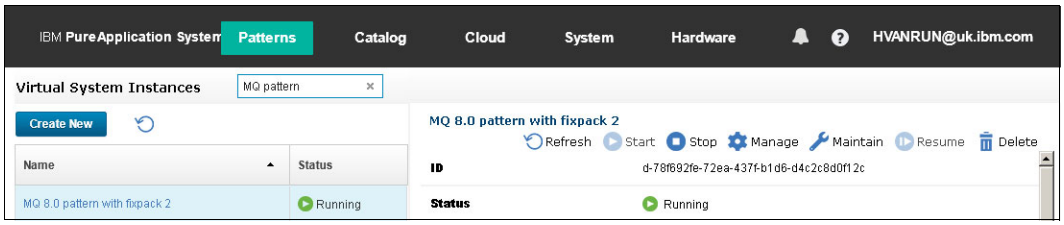


Figure 8-40 The deployed IBM MQ 8.0 Virtual System Instance with Fix Pack 2

2. In the Instance Console, use the slide **Maintenance mode** at the top to put the Pattern Instance in maintenance mode.
3. Click **Operations** and select the **MAINTENANCE** operation from the list. In the right pane, open the section **Maintenance fixpacks** and for “IBM MQ”, select **Fix Pack 8.0.0.3** from the drop-down list, as shown in Figure 8-41.

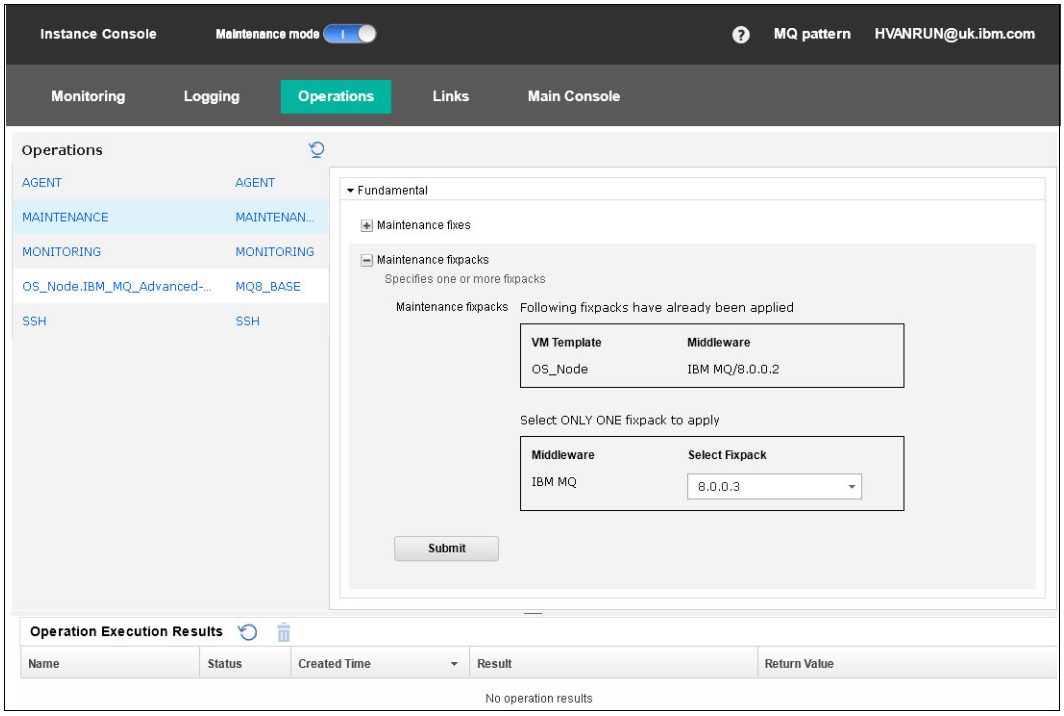


Figure 8-41 Select the Version 8.0.0.3 Fix Pack for IBM MQ under MAINTENANCE operations

4. With Fix Pack 8.0.0.3 selected, click **Submit** to apply this fix pack to the Virtual System Instance. This action starts an operation that is shown under “Operation Execution Results” (Figure 8-42).

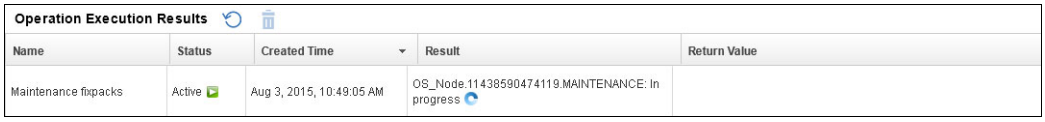


Figure 8-42 Operation “Maintenance fixpacks” active

5. After a few minutes, the operation completes (Figure 8-43). This window shows that Fix Pack 3 is applied to IBM MQ V8.0.

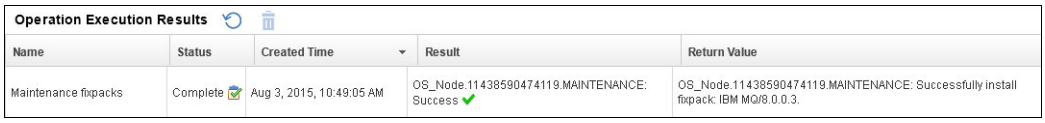


Figure 8-43 Operation "Maintenance fixpacks" completed

6. Use the **Maintenance mode** slider (at the top of the window) to take the Pattern Instance out of maintenance mode.
7. Log on to the VM as the IBM MQ administrator mqm and run the **dspmqver** command to confirm that Fix Pack 3 indeed is applied. Example 8-12 shows the expected output from **dspmqver**.

Example 8-12 Output from the dspmqver command after IBM MQ V8.0 Fix Pack 3 is applied

```
-bash-4.1$ dspmqver
Name:        WebSphere MQ
Version:     8.0.0.3
Level:       p800-USER-L150501.8
BuildType:   IKAP - (Production)
Platform:    WebSphere MQ for Linux (x86-64 platform)
Mode:        64-bit
O/S:         Linux 2.6.32-504.16.2.el6.x86_64
InstName:    Installation1
InstDesc:
Primary:     Yes
InstPath:    /opt/mqm
DataPath:    /var/mqm
MaxCmdLevel: 802
LicenseType: Production
```

8. Figure 8-25 on page 225 shows a *snapshot* with a description of RM07370. Click **Delete** to delete this snapshot because you no longer need it now that you confirmed that the Fix Pack is successfully installed.

Note: By default, PureApplication System automatically takes a *snapshot* before applying a WebSphere Fix Pack through the MAINTENANCE operation. By restoring this snapshot, you can effectively roll back the installation of the fix pack.

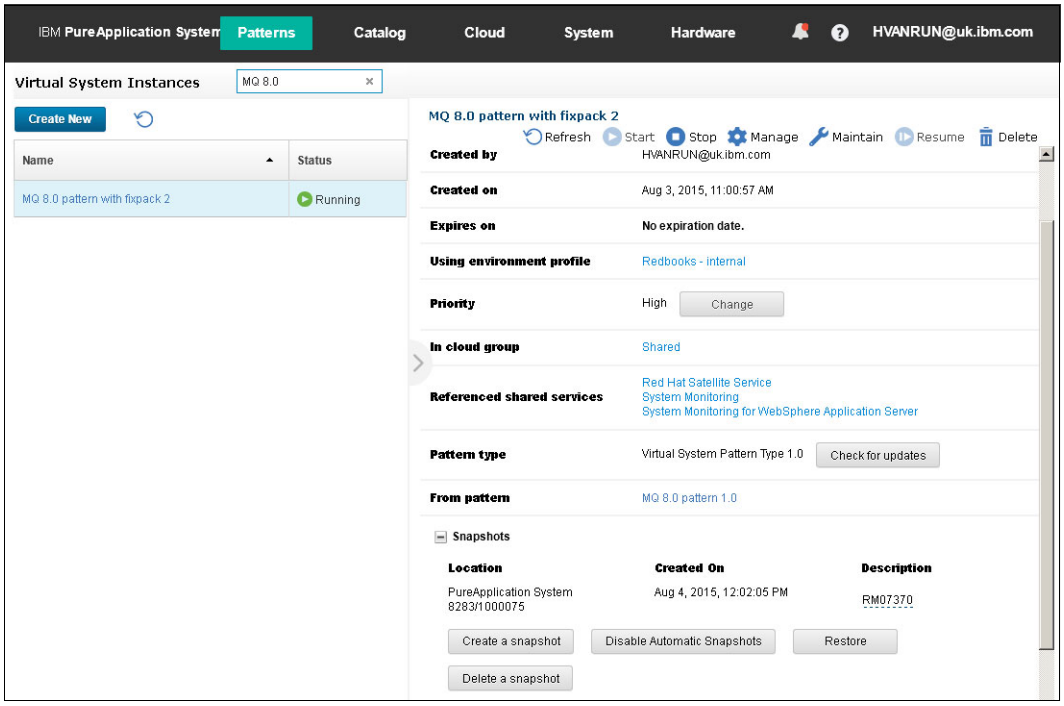


Figure 8-44 The snapshot that was automatically taken as part of the MAINTENANCE operation

8.3 Including Fix Packs in Virtual System Patterns

You might need to apply fix packs to IBM middleware running instances, and include specific fix packs in a pattern. This approach can help accelerate the roll-out of Pattern Instances and increase consistency across environments that are used for development, test, acceptance, and production.

8.3.1 IBM WebSphere Application Server V8.5.5

Assuming you have the correct packages that are loaded into the IBM Installation Manager Repository, you can easily change the fix pack that is applied at deployment time. All IBM WebSphere Application Server Network Deployment (WebSphere Application Server ND) Software Components support this model. Figure 8-45 shows an example of how this task is done for the WebSphere Application Server ND Standalone server Software Component.

Note: The WebSphere Application Server ND Software Components are special in the sense that they support two major versions, Version 8.0 and Version 8.5.5. For each of those major versions, you can select the appropriate Fix Pack in the Pattern Editor.

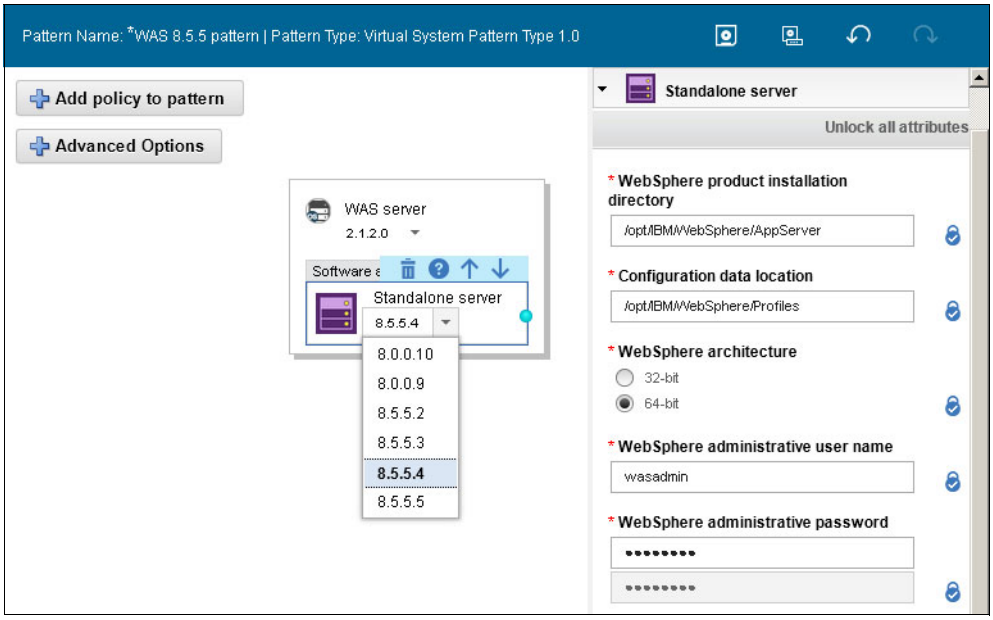


Figure 8-45 Select the version of IBM WebSphere Application Server ND in the Pattern Editor

8.3.2 DB2 10.5

Unfortunately, you cannot control the DB2 10.5 Fix Pack that is applied at deployment time from the VSP currently. The Pattern Editor suggests that you can change this for the DB2 Software Component, as shown in Figure 8-46 on page 239. However, if this is not the case, the version of the *IBM DB2 with BLU Acceleration Pattern* Pattern Type determines the DB2 10.5 Fix Pack.

Note: IBM intends to improve this situation by providing a choice for the DB2 10.5 Fix Pack of the Software Component from within the Pattern Editor.

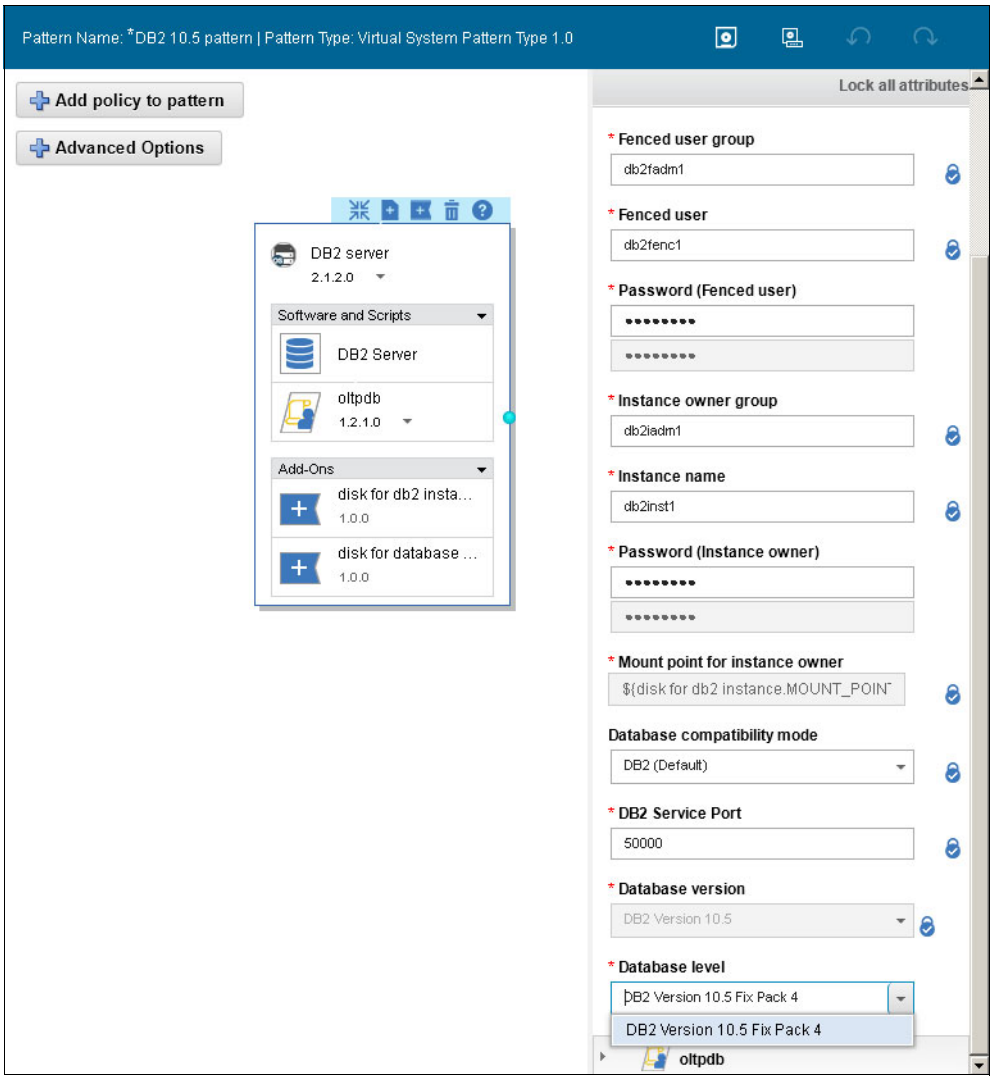


Figure 8-46 You cannot change the Fix Pack of DB2 10.5 in the Pattern Editor

The exact version mapping is shown in Table 8-4.

Table 8-4 The version of DB2 10.5 Fix Pack depends on the version of the Pattern Type

IBM DB2 with BLU Acceleration Pattern version	DB2 10.5 Fix Pack
1.2.0.0	3
1.2.0.1	3
1.2.0.2	3
1.2.0.3	4
1.2.1.0	4
1.2.2.0	5

8.3.3 IBM MQ 8.0

The *IBM MQ Advanced Virtual System Pattern Type* provides the IBM MQ Advanced Software Component. Even though it does not rely on IBM Installation Manager, it does provide a choice regarding the version of IBM MQ that is installed.

In Version 1.0.0.0 of the Pattern Type, both IBM MQ V8.0.0.2 and IBM MQ V8.0.0.3 are available from the drop-down box in the Pattern Editor, as shown in Figure 8-47.

Note: IBM plans to keep adding to the list of versions available.

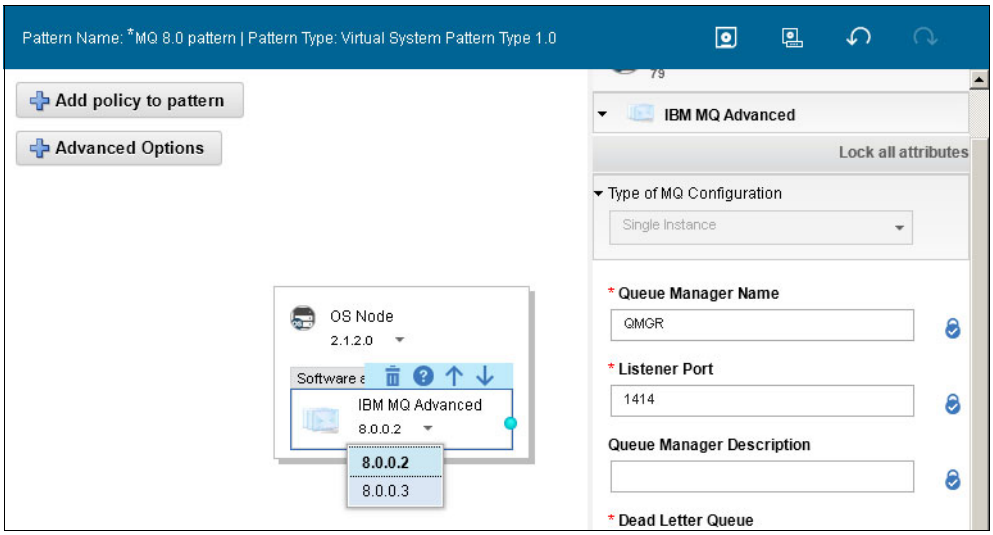


Figure 8-47 Select the version of IBM MQ from within the Pattern Editor

9

System and integrated components maintenance

One of the biggest advantages of using IBM PureApplication System is its integrated design and simplicity. It brings network, storage, compute nodes, and software (both management and middleware software) together with IBM expertise. Deploying middleware systems on the PureApplication platform takes minutes rather than days or weeks, and with only a few clicks. This same concept is also true when system and content maintenance come into play.

Although there are different hardware and software components in PureApplication System, such as network switches, storage devices, compute nodes, and virtualization management software, you do not need to update all of these components (including both hardware firmware and management software) individually. Instead, PureApplication System is treated as one integrated system, and IBM delivers updates as a single file. This file contains all of the updates for the management software and hardware firmware on the system.

This chapter describes aspects about PureApplication System and PureApplication Software maintenance. Sections include information about tools, interfaces, and preferred practices for maintaining and updating your PureApplication System and PureApplication Software environment.

This chapter covers the following topics:

- Maintenance overview
- Planning system update
- Preparing the system for an update
- Applying the system update
- Applying an integrated components update (group fix)
- Performing system health status check (post-upgrade)
- Maintenance-related preferred practices

© Copyright IBM Corp. 2015. All rights reserved.

9.1 Maintenance overview

To take advantage of new features, improvements, bug fixes, and new application releases of integrated components, upgrade your system and keep it updated.

When you upgrade to the latest PureApplication platform, you avoid bugs that were in a prior release, and you also benefit from new features and added improvements to usability and system performance.

The hardware and software components in PureApplication System, such as network switches, storage devices, compute nodes, and virtualization and system management software, are described in 1.1.2, “PureApplication Software” on page 3. In a traditional, distributed environment, system administrators apply hardware firmware to individual hardware components, and each of those upgrades requires specific skill sets and planning for resources, time, and logistics. This translates to added costs, efforts, and risks.

The PureApplication System has three different types of maintenance packages:

- System update (system fix pack)
- Component update (group fix)
- Emergency fix pack, which can be applied to the system to update a smaller group of system components

Here are details about these maintenance packages.

Note: Another type of maintenance package is that of virtual image and integrated pattern updates. These are primarily about workload maintenance. For more information about these topics, see the following chapters:

- Chapter 7, “Operating system maintenance” on page 155
- Chapter 8, “IBM middleware maintenance” on page 205

- System update (system fix pack)

  This single file contains all updates for the management software and hardware firmware on the system. Specifically, this file includes updates for the following system components:

  - PureApplication System management software
  - Virtualization system software
    - Virtual System Manager for Intel Systems (VSM)
    - Power FlexSystem Manager (FSM)
  - Firmware updates for the following components:
    - Compute nodes
    - Chassis
    - Network switches
    - Storage devices
    - Power

  System maintenance is used to update nearly everything that makes up the system, except guests running virtual machines (VMs) and the middleware that runs on them. Updates are also provided to hypervisors running on compute nodes.

  Some system packages include updates only for some system components, that is, the ones for which an update is needed.

- Component update (group fix)

  The group fix is an update package for integrated components. It contains updated pattern types for shared services and integrated components of the system. A group fix can contain updates to the following software components, based on your environment and system software level:

  - Foundation Pattern Type
  - WebSphere Pattern Type
  - Sample pattern for vNext
  - Application Pattern Type for Java
  - System Monitoring Support for SOA Pattern Type
  - Database Performance Monitoring Tools Pattern Type
  - Database Performance Monitoring Pattern
  - IBM Tivoli Monitoring Pattern
  - Tivoli Endpoint Manager
  - IBM Image Construction and Composition Tool Pattern Type
  - General Parallel File System Pattern Type

  Some group fix packages include updates only for some software components, that is, the ones for which an update is needed. Additionally, updates to integrated components might be delivered as multiple files (one for each of the pattern types or integrated components). For more information, see “Integrated components (group fix) maintenance overview” on page 244.

- Emergency fix pack, which can be applied to the system to update a smaller group of system components

  Emergency fix packs update a smaller group of the system, such as a specific shared service. Because of the nature of emergency fix packs, they can be individual fix packs or a group of fix packs that are associated with particular parts of the system.

System maintenance overview

The installation of system updates includes an automated process that applies updates and ensures VM instances that are created on the system remain in their original state and available without requiring user intervention, which means that system updates do not disrupt running VMs and do not require downtime of the workload.

During the automated process, updates are applied to the PureSystems Managers, and the hardware firmware, if applicable. The leader, PureSystems Manager (PSM), is updated first. During this time, the console is unavailable. After the first PSM is updated, the second PSM is updated.

During the update of both PSMs, new instances cannot be deployed. Existing workload instances remain available for use during the update.

Some fix packs include compute node firmware updates. For these updates, system updates make compute node firmware available for the compute nodes, but do not apply them automatically during the system update. For these updates, after the system update completes, compute node updates can be applied separately. You can view your compute node firmware updates by clicking **System** and then clicking the **System Maintenance Compute Node update status** tab, as shown in Figure 9-1.

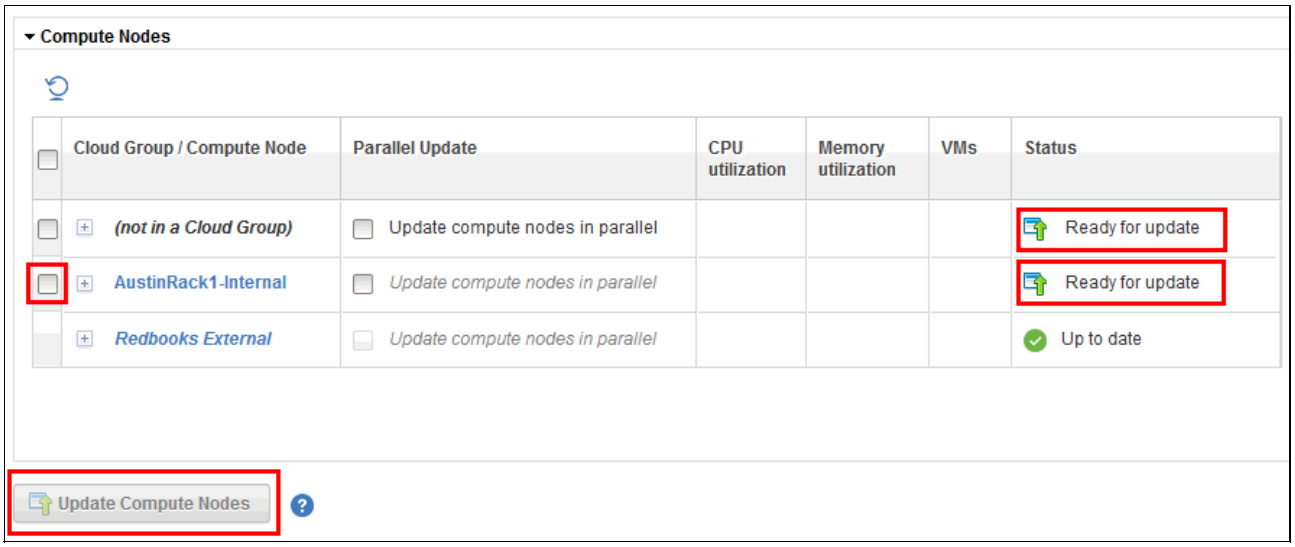


Figure 9-1 Compute node update status

An outage might occur during the update for a compute node for which an instance evacuation is required but the cloud group does not have enough physical capacity for the evacuation to occur. The cloud group is not highly available, and the updates cannot complete without affecting the instances. Some instances are stopped and cannot be restarted until the updates are complete and cloud group resources are fully restored. For more information about compute node updates, see 9.2.7, “Time and workload continuity planning” on page 255.

All system maintenance operations are configured and applied by using the PureApplication System console or the pure.cli interface. To go to the maintenance page, click **System** → **System Maintenance** from the PureApplication System console.

Integrated components (group fix) maintenance overview

Group fix maintenance packages contain updated pattern types for shared services. Shared services pattern types come with PureApplication System by default. The type and version of pattern types can vary, depending on PureApplication environment, such as PureApplication System with Intel platform and PureApplication Software.

Applying a group fix update is done online and does not disrupt any running workload on the system during the update process. However, after the group fix is installed, you might need to upgrade your deployed pattern type instances to the latest level. Upgrading deployed pattern type instances and shared services are described in 9.5.4, “Upgrading deployed pattern type instances” on page 272.

IBM delivers integrated component updates as multiple files (one for each of the pattern types or integrated components), in contrast to the single file PureApplication System updates. There might be some dependencies between those pattern types. For example, most of the pattern types depend on the foundation pattern type and therefore, the foundation pattern type must be installed or updated first.

9.2 Planning system update

Planning before applying updates saves time and mitigates risks in the update process.

Note: IBM regularly publishes system maintenance documentation for fix packs at the IBM Support Fix Central website (http://www.ibm.com/support/fixcentral). Starting with System Fix Pack 2.1.0.2, you can find system maintenance for both W1700/W2700 and W1500/2500 on the IBM Support Fix Central website. Before the 2.1.0.2 fix pack level, ask your IBM representative for system maintenance documentation.

With each fix pack, IBM adds new functions and makes improvements to the PureApplication System. Fix packs can provide some new capabilities for a specific component or change the behavior of some functions. Review details of fix packs to ensure that you do not encounter issues with new functions or improvements in your environment.

Some bug fixes can change the behavior of a function or cause some functions to work differently than designed. To prevent problems or workload disruption, bug fixes and improvements must be reviewed carefully before being applied. It is important to know what is going to change and what is needed to be done after the system update.

For each upgrade, review the Release Notes and the “What’s new” section in the IBM Knowledge Center. You can find these items at the following website:

- IBM PureApplication System Release Notes by version:

  http://www.ibm.com/support/docview.wss?uid=swg27039159
- IBM PureApplication Software Release Notes by version:

  http://www.ibm.com/support/docview.wss?uid=swg27045293
- IBM PureApplication System W2700 IBM Knowledge Center - What’s new:

  http://www.ibm.com/support/knowledgecenter/SSCRSX_2.1.0/doc/iwd/gsr_changes.dita?lang=en
- IBM PureApplication System W2500 IBM Knowledge Center - What’s new:

  http://www.ibm.com/support/knowledgecenter/SSCR9A_2.1.0/doc/iwd/gsr_changes.dita?lang=en
- IBM PureApplication Software IBM Knowledge Center - What’s new:

  http://www.ibm.com/support/knowledgecenter/SSL5ES_2.1.0/doc/iwd/gsr_changes.dita?lang=en

Note: When planning your updates, contact an IBM Service Representative and ask if IBM Service Representative (ibmeng) credentials are required to apply the fix packs (before the 2.1.0.2 level) you are considering for your PureApplication model.

9.2.1 Deciding on a maintenance path

As part of maintenance planning, your administrator must select a maintenance path for the upgrade operation, as the maintenance path might differ among environments. The IBM Support websites (see the links in this section) provide regular updates for the maintenance paths of PureApplication System and PureApplication Software environments for your use in maintenance planning.

There are some restrictions for upgrades, and you should follow the supported maintenance paths for your environment. For example, you cannot upgrade your system from level 2.0.0.x to level 2.1.0.1 until you first upgrade to level 2.1.0.0.

The following documents can be used to determine your maintenance path:

- PureApplication System maintenance paths that are supported for each product version:

  http://www.ibm.com/support/docview.wss?rs=3023&uid=swg27039794

- PureApplication Software maintenance paths that are supported for each product version:

  http://www.ibm.com/support/docview.wss?uid=swg27045473

If you are not sure about the maintenance path for your environment, contact your IBM PureApplication Service Representative (see your product documentation) or open a Service Request (SR) (https://www.ibm.com/support/servicerequest).

9.2.2 Obtaining the system update procedure

The information in this book is intended to be a general guide for system update procedures, rather than a step-by-step system update installation guide. Although system update installation procedures can vary and have different or extra steps, depending on the fix pack release or PureApplication model, most general practices still apply, regardless of the chosen system fix pack to be applied.

Note: IBM regularly publishes system maintenance documentation for fix packs on the IBM Support Fix Central website (http://www.ibm.com/support/fixcentral). Starting with System Fix Pack 2.1.0.2, you can find system maintenance for both W1700/W2700 and W1500/2500 on the IBM Support Fix Central website. Before the 2.1.0.2 fix pack level, ask your IBM representative for system maintenance documentation.

9.2.3 Taking backups of the system

Backup features in the IBM PureApplication System include procedures for capturing the system configuration, cloud configuration, and workload catalog data. You can use restore features to re-create the cloud configuration, restore workload catalog components, and restore the system configuration.

Before the system and integrated component update, you should take system component and workload component backups. For more information about taking backups, see Chapter 10, “Backup and restore” on page 275.

9.2.4 Maintenance in a multi-system environment

If you are using the PureApplication multi-system setup and intend to upgrade the systems in your management domain, you can expect the following behaviors both during the upgrade and while your systems are at differing firmware levels.

During the upgrade

When upgrading any system, the management nodes experience an outage for approximately one to two hours. As management communication of multi-system members occurs between management nodes, they cannot talk to nodes that are being updated during the outage.

During the outage, the following items are true:

- You cannot display the shared catalog view of the system catalog artifacts from any other system.
- Multi-system deployments continue to run, but the status of VMs on that system as displayed on the console of another system might be temporarily out of sync or unknown.
- New deployments to externally managed environment profiles cannot connect to shared services if those shared services were originally deployed from the system that is being upgraded.

After the management console becomes available, there are times during a system upgrade during which the upgrade blocks other system jobs from running. Until the upgrade is complete, do not attempt to copy remote catalog artifacts to the system being upgraded or deploy multi-system deployments to that system.

When your systems are at different firmware levels

Tip: Plan to upgrade all the systems in your multi-system domain, and especially your multi-system subdomain, in short succession to each other to avoid issues.

If you upgraded one or more systems in your domain, but not all systems, then you can expect the following behavior on your management domain:

- Shared catalog views of catalog artifacts, such as virtual images and script packages, display as you expect.
- Copying or synchronizing catalog artifacts from one system to another is supported, but the copy fails if the artifact you are copying requires a level of firmware not present on the destination system.
- Some catalog artifact types might support a shared view and copying only on the newer firmware level. In this case, you cannot display these artifacts on systems at an older level or copy to those systems.

For your deployment subdomain, existing multi-system deployments continue to operate, and you can display their status and manage them from any system in the deployment subdomain. New multi-system deployments can also be issued to the deployment subdomain, but be aware of the following restrictions:

- New deployment capabilities might not be able to be exercised until all systems in the deployment subdomain are upgraded to the latest firmware.
- When you deploy, content that is required by your pattern must be present on the system where a given pattern part is being deployed. If you upgrade and enable default data on the system with the new firmware, then that same default data is not present on the systems with older firmware, so that you cannot deploy from the higher-level systems to the lower-level systems. You have the following options in this case:
  - Plan your window of firmware upgrades and default data updates so that you do not need to issue multi-system deployments until all systems are updated.
  - Issue all multi-system deployments from the system with the lower firmware level. If this same default data is present on the system with newer firmware, you can expect a successful deployment.
  - Before upgrading the first system, you can lock the plug-ins in your pattern. After the upgrade and after updating the default data, if you deploy this pattern, it still uses the older default data, which means that you can issue multi-system deployments even from the system with the newer firmware.

You cannot upgrade these running deployments to newer default data until all systems on which the deployments are running are upgraded to the new firmware and default data.

After the upgrade

Some system update releases can include Oracle Java truststore updates. In this case, the Java default truststore is upgraded during the system maintenance installation, making it necessary to reimport remote or peer location certificates in a multi-system environment.

9.2.5 Downloading the required software packages

After determining the best maintenance path for your environment, download the required update packages for your environment.

For a PureApplication System environment

In a PureApplication System environment, complete the following steps:

1. Access the PureSystems Center at the following website:

   `https://www.ibm.com/software/brandcatalog/puresystems/centre/`

2. Click the **System updates** tab.
3. Click **PureApplication System**.
4. Click the **W1500/2500** tab or the **W1700/2700** tab based on your platform.
5. Expand the section for the current version of your system. For example, **PureApplication System V2.1.x Updates**.
6. Click **PureApplication System Fix Pack 2.x.x** and download the System fix pack from IBM Fix Central. Note the file name format of `pure-appsys-intel_<version>.ifp or pure-appsys-power_<version>.ifp`.
7. Click **Group Fix for 2.x.x** and download group fixes from Fix Central.
8. If you also want to update the IBM base operating system image or virtual pattern images, expand the **Virtual Image and Integrated Pattern Updates** section on the PureSystem Center System Update website, and then download the needed images. You can also download other virtual images or integrated patterns from here. You can download each update separately.

   A sample group fix is shown in Figure 9-2 on page 249.

Figure 9-2 A sample group fix

Note: Use the entitlement key that you received with your purchase agreement to download updates from Fix Central that require entitlement.

For a PureApplication Software environment

In a PureApplication Software environment, complete the following steps:

1. Access the IBM Support Portal: Fix Central at the following website:

   `http://www.ibm.com/support/fixcentral/`

2. Click the **Select product** tab.
3. Select **PureSystems** from Product Group.
4. Select **PureApplication Software** from the **Select from PureSystems** drop-down list.
5. Select your installed version.
6. Select your platform.
7. Click **Continue**.
8. Select **Browse for fixes** and click **Continue**.
9. Select the **IBM PureApplication Software upgrade V2.x** fix pack and group fixes that are applicable to your environment. Note the file name format of `pure-appsys-software_<version>.ifp`.

10. Click **Continue**.
11. **Accept** the agreement and click **Download**.

Note: Use the entitlement key that you received with your purchase agreement to download updates from Fix Central that require entitlement.

9.2.6 Importing the system fix pack

Note: You must be assigned the Hardware Administration role to add a fix pack to the system.

The size of the system fix pack file is ~ 6 GB - 20 GB, depending on your current version and platform. Most browsers and command-line tools limit uploads to about 2 GB or a maximum of 4 GB. The system fix pack file exceeds this limit. So, you cannot upload it from your local hard disk drive by using the browser interface or command-line tools. Instead, PureApplication System gives you other options:

- Put the system fix file on an HTTP or HTTPS server and provide the URL to the browser interface or command-line tool (pure.cli).
- Put the system file on an SSH server and provide the IP address, path, and credentials on the browser interface or command-line tool (pure.cli).

Note: If you do not have access to an external SSH or HTTP or HTTPS server, you can deploy one of them on your PureApplication System and use it as an HTTP, HTTPS, or SSH server to place the system fix pack file on. This approach works only if the management network can reach the data network.

Importing the system fix pack by using the system console

To import the system fix file by using the PureApplication System or PureApplication Software console, complete the following steps:

1. Put the system fix pack file onto an HTTP, HTTPS, or SSH server that is accessible from the PureApplication management nodes. HTTP, HTTPS, or SSH traffic should be open between the PureApplication management nodes and SSH, HTTP, or HTTPS server on the firewall.
2. Click **System** → **System Maintenance** on the PureApplication System console.
3. Determine whether there are any fix packs already uploaded or left from a previous upgrade. If so, click the **delete** icon that is shown in Figure 9-3 to delete them. Otherwise, the **Add Fix Pack** button remains deactivated and you cannot add the fix pack to the system.

Figure 9-3 Delete fix packs from previous upgrades

4. Click **Add Fix Pack** on the System maintenance window, as shown in Figure 9-4 on page 251.

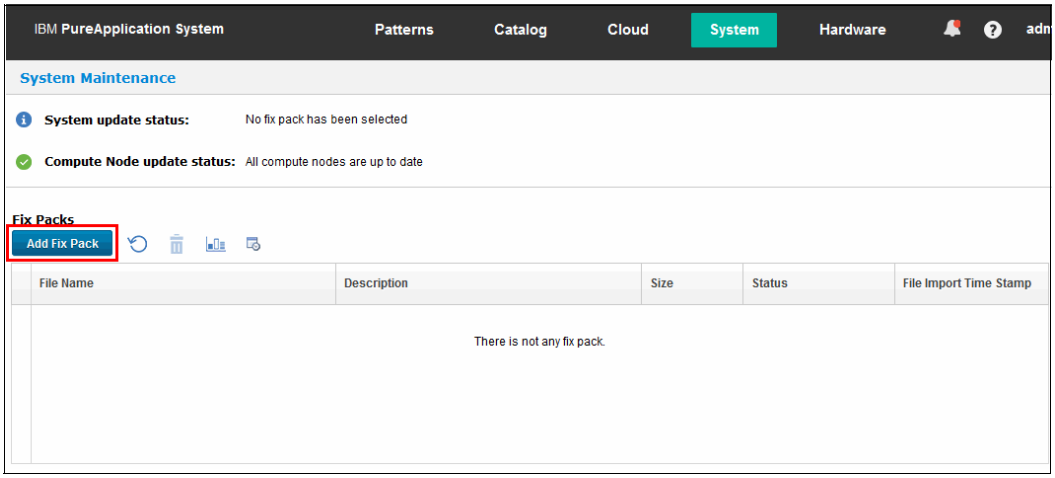


Figure 9-4 Add Fix Pack to the system

5. The Add Fix Pack window opens. There are two options, **HTTP/HTTPS** or **SCP.**
   - If you put the system fix file on an HTTP or HTTPS server, complete the following steps:
     i. Select the **HTTP/HTTPS** tab.
     ii. Enter the URL.
     iii. Indicate whether authentication is enabled on the HTTP server.
     iv. Provide the HTTP/HTTPS credentials.

     Figure 9-5 shows this window.

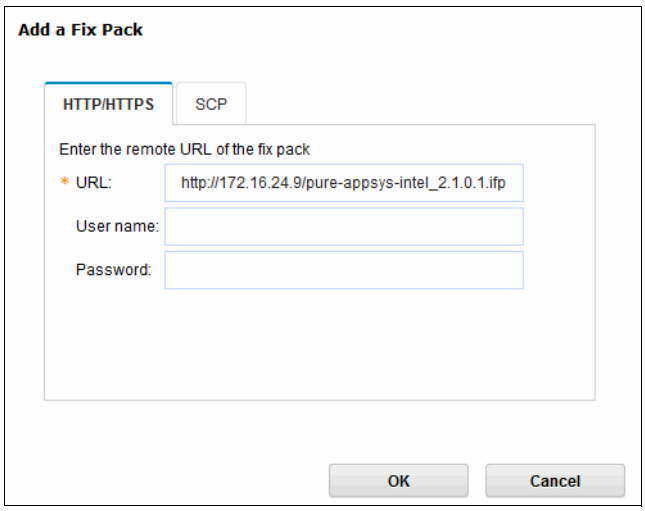


Figure 9-5 Add Fix Pack windows with HTTP tab selected

   - If you put the system fix file on a SSH/SCP server, complete the following steps:
     i. Select the **SCP** tab.
     ii. Provide the IP address of the SSH/SCP server.
     iii. Provide the SSH/SCP port number.

iv. Provide the SSH/SCP server credentials.

v. Provide the full path of the system fix file.

Figure 9-6 shows this window.

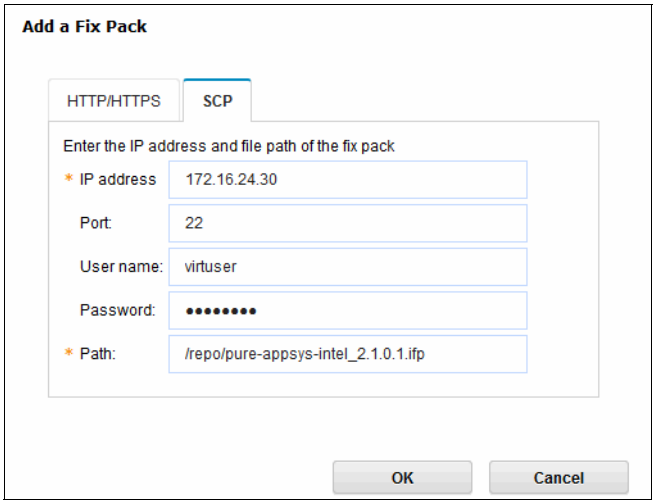


Figure 9-6 Add Fix Pack windows with SCP tab selected

6. Click **OK**. You can close the window.
7. The system fix pack starts uploading to the PureApplication manager node, as shown in Figure 9-7. Because PureApplication uses jobs, the upload continues in a way that is similar to a background job.

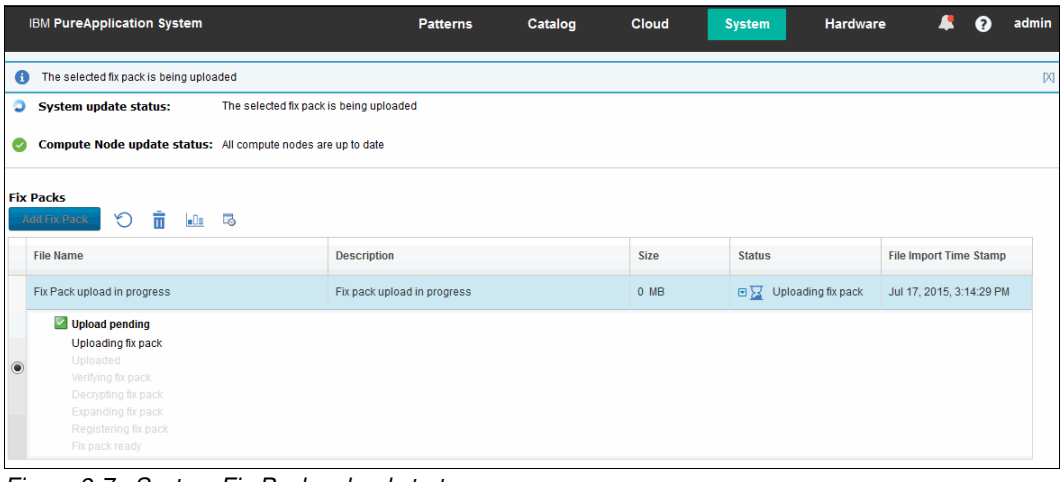


Figure 9-7 System Fix Pack upload starts

8. During the import, the system performs the following steps:
   a. Uploads the fix pack.
   b. Verifies the fix pack.
   c. Decrypts the fix pack.
   d. Expands the fix pack.
   e. Registers the fix pack.

9. When the upload finishes, the fix pack is ready to update the system, as shown in Figure 9-8.

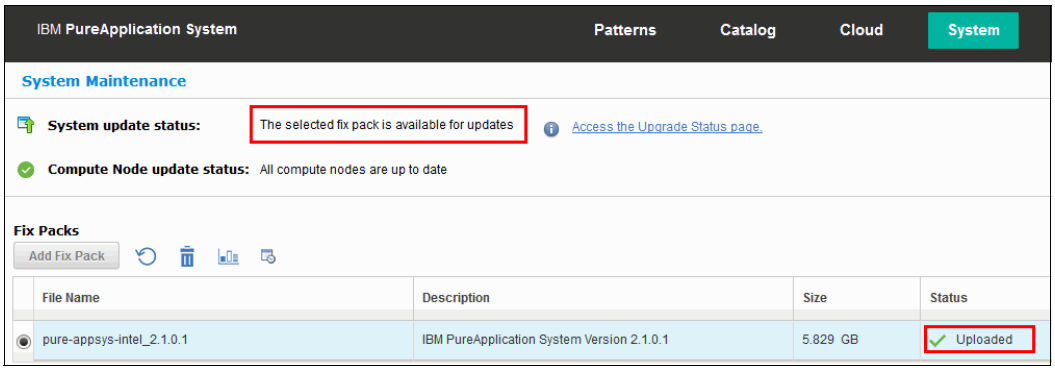


Figure 9-8 Fix pack is uploaded and ready for the update operation

10. After the fix pack is uploaded, click the **System Updates** tab (shown in Figure 9-9) to see which components will be updated and how long each update will take. The components and estimated time frames are used in the system update planning phase, as described in “9.2.7, “Time and workload continuity planning” on page 255”.

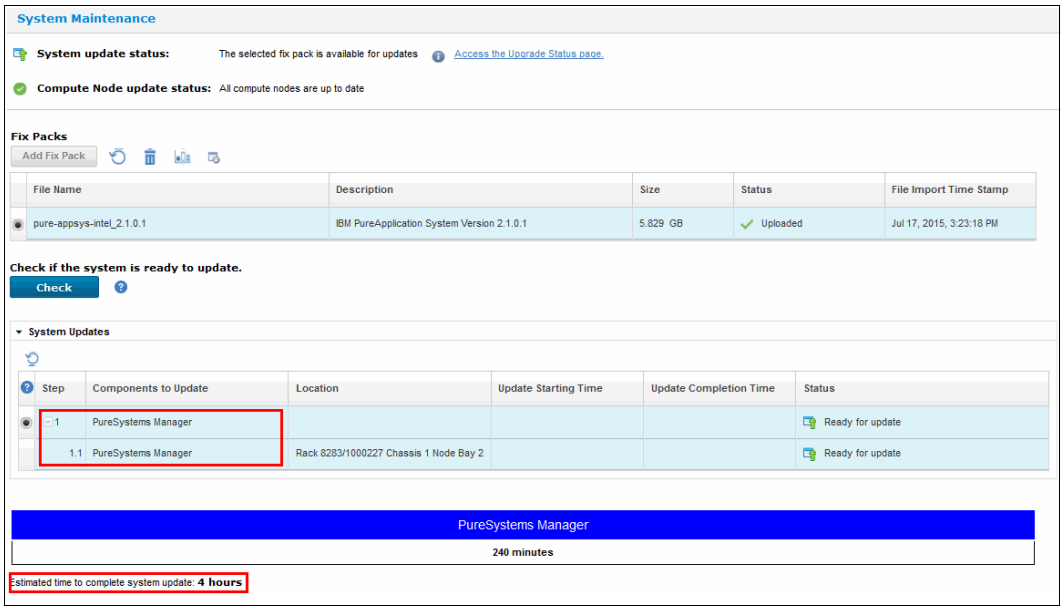


Figure 9-9 Components to be updated and estimated time of updates

Importing the system fix pack by using pure.cli

You can import the system fix pack file by using the command-line interface (CLI) in a PureApplication System or PureApplication Software environment.

Note: pure.cli is described in detail in Chapter 13, “IBM PureApplication System command-line interface and REST API” on page 355”.

To import a system fix file into PureApplication System or PureApplication Software, complete the following steps:

1. Put the system fix pack file onto an HTTP or SSH server that is accessible from the PureApplication management nodes. HTTP, HTTPS, or SSH traffic should be open between the PureApplication management nodes and HTTP, HTTPS, or SSH server on the firewall.
2. Open the CLI from your client, as shown in Example 9-1.

Example 9-1 Open PureApplication CLI

```
# cd /directory/path/to/purecli/bin directory
# ./pure -h <PureApplication_Management_IP> -u admin
password:
>>
```

3. There are two options for importing a system fix file by using SSH/SCP, HTTP, or HTTPS:
   - Use the sample CLI code for SSH, as shown in Example 9-2.

Example 9-2 Import the system fix file by using SSH

```
Welcome to the IBM PureApplication System CLI.  Enter 'help' if you
need help getting started.
>>> admin.fixpacks.list
<bound method FixPacks.list of [  ]>
>>> fixfile = {
... "access_type":"scp",
... "remote_ipaddress":"172.16.24.30",
... "remote_port":"22",
... "remote_path":"/repo/pure-appsys-intel_2.1.0.1.ifp",
... "remote_username":"root",
... "remote_password":"passw0rd"
... }
>>> fp = admin.fixpacks.create(fixfile)
>>> fp.waitFor()
>>>
```

   - Use the sample CLI code for HTTP or HTTPS, as shown in Example 9-3.

Example 9-3 Import the system fix file by using HTTP or HTTPS

```
Welcome to the IBM PureApplication System CLI.  Enter 'help' if you
need help getting started.
>>> admin.fixpacks.list
<bound method FixPacks.list of [  ]>
>>> fixfile = {
... "access_type":"url",
... "remote_url":"http://172.16.24.30/pure-appsys-intel_2.1.0.1.ifp",
... "remote_username":"",
... "remote_password":""
... }
>>> fp = admin.fixpacks.create(fixfile)
>>> fp.waitFor()
>>>
```

4. After the import, check the status of the fix file, as shown in Example 9-4 on page 255.

Example 9-4 Verify the status of the fix file after the file import completes

```
Welcome to the IBM PureApplication System CLI.  Enter 'help' if you
need help getting started.
>>> admin.fixpacks.list
<bound method FixPacks.list of [
  {
    "change_plan": (nested object),
    "created_time": "Jul 21, 2015 1:50:22 AM",
    "description": "IBM PureApplication System Version 2.1.0.1",
    "fixes": None,
    "id": "fd1cd2ac-c8d5-4431-b1e6-ee0e200e2b1c",
    "name": "pure-appsys-intel_2.1.0.1",
    "short_description": "PureApplication System 2.1.0.1",
    "size": 6258511872,
    "state": "ready",
    "unique_size": 6258511872,
    "updated_time": "Jul 21, 2015 1:59:23 AM",
    "url": "/admin/resources/fix_packs/fd1cd2ac-c8d5-4431-b1e6-ee0e200e2b1c"
  }
]>
>>>
```

5. Check the value of the State field:
   - If the State field shows `ready`, the import operation was performed successfully.
   - If the State field shows `failed`, the import operation failed. You must fix the problem, and then delete the failed fix pack from the catalog, as shown in Example 9-5.

   Example 9-5 Delete the failed fix pack from the system

   ```
   >>> fp = admin.fixpacks[0]
   >>> admin.fixpack.delete(fp)
   >>> admin.fixpack.list
   <bound method FixPacks.list of [  ]>
   >>>
   ```

The system fix file import can fail if there is a network problem between the SCP or HTTP server and the PureApplication manager nodes, or the wrong information was provided about the SCP or HTTP details. To resolve this issue, fix the problem, and then repeat steps 1 on page 254 - 5.

9.2.7 Time and workload continuity planning

The time that is required to update a system varies based on usage and requirements, and the number and duration of stopping points that you set during the automated system updates operation.

If a system fix pack does not include any compute node firmware update packages, workload instances and running VMs continue running and are not disrupted by the update operation. Only the system console is inaccessible for 60 - 120 minutes. During this outage, the status of the update to the leader PureSystems Manager can be monitored from the Upgrade Status page on the non-leader PureSystems Manager (for example, `https://<IP Address of Second PSM>/upgrade`). During the update to both PureApplication System Managers, new instances cannot be deployed.

To see which components will be updated and how long each update will take, click **System** → **System Maintenance** from the PureApplication System console, and click the **System Updates** tab, as shown in Figure 9-9 on page 253. Based on the time estimation for each component and the estimated time to complete the system update, you can plan your system update.

Note: For some updates, and based on the version, the estimated time to complete the system update can exceed the customer maintenance window. In this case, apply some of the updates and apply the remaining part in another maintenance window. To do so, PureApplication System maintenance feature has stopping points that you can select for the system update.

Workload continuity and compute node update

Note: There are no compute node firmware updates for PureApplication Software because it is not an appliance. PureApplication Software uses the compute resources from other virtualized environments, such as VMware or PowerVM.

When compute node updates are included in the fix pack, an outage can occur during the updates when an instance evacuation is required and the cloud group does not have enough physical capacity for the evacuation to occur. If the cloud group is not highly available, the updates cannot complete without affecting the instances. Some instances are stopped and cannot be restarted until the updates are complete and the cloud group resources are fully restored.

Optionally, a cloud group can be set to reserve resources for high availability use. This option reserves computing resources (CPU and memory) within the cloud group that are equivalent to one compute node. The reserved capacity in a cloud group containing N compute nodes is 1 / N of the resources (CPU and memory) on each compute node.

If the high availability option for reserve resources is enabled in the cloud group, the evacuation of VM instances from one compute node to another, if required, always completes successfully without impacting the VM instances because the required resources in the cloud group are set aside in advance.

When planning system update time and workload continuity, take the following scenarios into account, depending on your environment and cloud group configuration, and then plan your system update:

1. If a system fix pack update does not include any compute node updates:
   - All workload instances and VMs run without any impact during the update.
   - The only service outage is on the PureApplication manager nodes. During this time, new instances cannot be deployed.
2. If a system fix pack update does include compute node updates:
   - If there is no VM (running or stopped) on the compute node, the updates can complete. There is no need to take any action or planning before the update.
   - If cloud group high availability is active and the reserve resources for high availability option is enabled for the cloud group:
     - All workload instances and VMs run without any impact during the update.
     - The only service outage is on the PureApplication manager nodes. During this time, new instances cannot be deployed.

– If cloud group high availability is inactive, one of two situations regarding outage and evacuation can occur during the system update process:

  • When an evacuation is required and the cloud group has enough physical capacity for the evacuation to occur, as determined by the cloud group type, the updates can complete without impacting the running VM instances.

    The only service outage is on the PureApplication manager nodes. During this time, new instances cannot be deployed.

  • When an evacuation is required and the cloud group does not have enough physical capacity for the evacuation to occur, compute node updates cannot complete without affecting the VM instances. Depending on the compute capacity, some instances are stopped and cannot be restarted until the updates are complete and the cloud group resources are fully restored. The VMs are stopped by their *priority* value first (VMs with low priority stop before VMs with high priority). This situation can be prevented by the system or workload administrator by stopping and storing extraneous VMs before running the compute node update.

    An outage occurs on the PureApplication manager nodes and workload instances. The length of the workload instance outage depends on the number of compute nodes and number of VMs on each compute node.

If you have a cloud group that does not have enough physical capacity for the evacuation to occur, you can temporarily move a compute node from a different cloud group with active high availability (if any) to the specified cloud group and then perform the update. Do not forget to remove the temporary compute node and reassign it to its original cloud group after the compute updates complete.

Note: System updates make compute node firmware available for compute nodes, but *they are not applied automatically* during the system update. If there is a compute node firmware update in a system fix pack file, then, after the system update completes, the compute node updates must be applied separately. For more information, see 9.4.3, “Applying compute node updates” on page 268.

Regarding VMs with the General Parallel File System (GPFS) (pinned VMs): When compute node updates are included in the fix pack and you must update compute nodes with pinned VMs, consider the following items:

- If you are running Active-Passive with only a single-server VM in each virtual application instance, your file system goes offline when the Active instance is updated. You can avoid this situation by updating the passive virtual application, then starting the manual failover procedure that is described in the *Configuring an Active / Passive GPFS deployment* topic at the IBM Knowledge Center at the following website:

  `http://www.ibm.com/support/knowledgecenter/SSCR9A_2.0.0/doc/GPFS12/gpfs_deploy_active_passive.dita`

- If you deployed an Active-Active cluster with one server VM for each virtual application instance, but you deployed it to a single cloud group with a single compute node, the compute node cannot be upgraded without causing an outage of the GPFS file system. For optimal results, deploy Active-Active clusters to separate cloud groups with separate compute nodes if two or more PureApplication System racks are not available.

The time that is required for a compute node update can vary between each release of fix packs and depends on the total number of compute nodes in a PureApplication System. IBM provides estimated durations based on the number of compute nodes, and this information is included in each PureApplication System update document. Ask your IBM representative for this information.

9.3 Preparing the system for an update

Before installing a system maintenance update, some actions are needed on the system. General actions are included here. However, each system fix pack release can require additional steps, depending on the PureApplication platform and model and current version of the system. Have the system update procedure of the fix pack level you are going to apply available for a complete list of steps. For more information, see 9.2.2, “Obtaining the system update procedure” on page 246.

9.3.1 Activating maintenance mode

Before beginning the update process, put the system in maintenance mode. This is the only way of preventing other users from using the system while the fix is being applied. It also prevents issues that might arise because of similar conflicts.

While maintenance mode is enabled, you cannot make new requests, including but not limited to the following ones:

- Deployment of new virtual system instances
- Deployment of new virtual application instances
- Deployment of new shared service instances
- Deployment of new databases
- Start, stop, delete, and maintenance of Virtual System Instances or VMs
- Start, stop, delete, and maintenance of Shared Service Instances
- Start, stop, delete, and maintenance of Virtual Application Instances
- Start, stop, delete, and maintenance of database instances
- Creation of new IP groups
- Creation of new cloud groups
- Creation of or removing storage volumes and volume groups
- Creation of new environmental profiles
- Changing the Default Deploy settings
- Changing the Network configuration (Add VLAN, Add Link, and so on)
- Creation of new emergency fix in the catalog
- Adding or removing patterns, pattern components, plug-ins, and so on

However, compute node operations such as quiesce, maintain, and initialize are allowed when maintenance mode is enabled because if you are going to update the compute nodes, those actions are required for performing the compute node firmware update.

Maintenance mode is also a way of notifying system users about a planned upgrade or system maintenance. When enabled for a specific period, it shows a notice at the top of the page with the end date and time for the maintenance operation.

To put the system in maintenance mode, complete the following steps:

1. Click **System** → **System Settings** from the PureApplication System console.
2. Click to expand the **Maintenance mode** tab.
3. Provide a start date and time and an end date and time, as shown in Figure 9-10 on page 259.

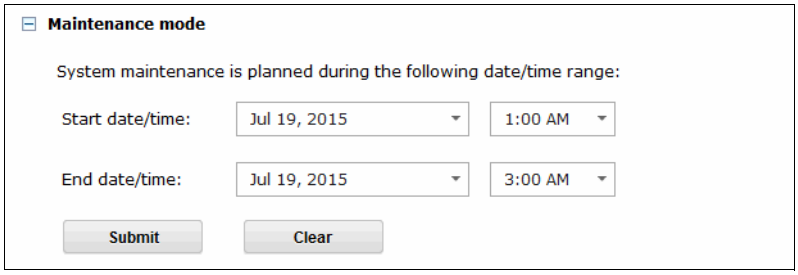


Figure 9-10 Enable maintenance mode

4. Click **Submit**.
5. After maintenance mode is enabled, the system notifies all users by displaying a notice at the top of the window, as shown in Figure 9-11.

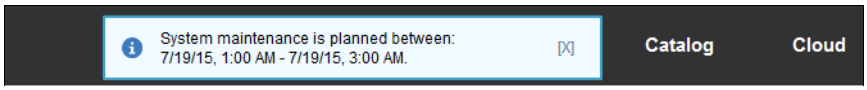


Figure 9-11 Maintenance mode notification window

If the system update ends before the maintenance end date and time, click **Stop maintenance**, as shown in Figure 9-12.

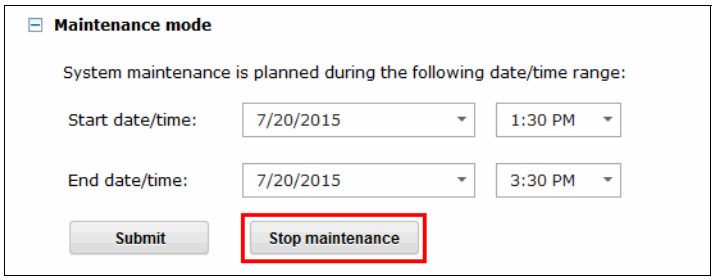


Figure 9-12 Stop maintenance mode

9.3.2 Disabling the Service and Support Manager

If the Service and Support Manager (Call Home) is enabled, disable it to prevent service tickets (PMRs) from being automatically opened during the system maintenance process. Complete the following steps:

1. Click **System** → **System Settings** from the PureApplication System console.
2. Click the **Service and Support Manager** tab.

3. Select **Do not collect troubleshooting information and do not open a service request option**, as shown in Figure 9-13.

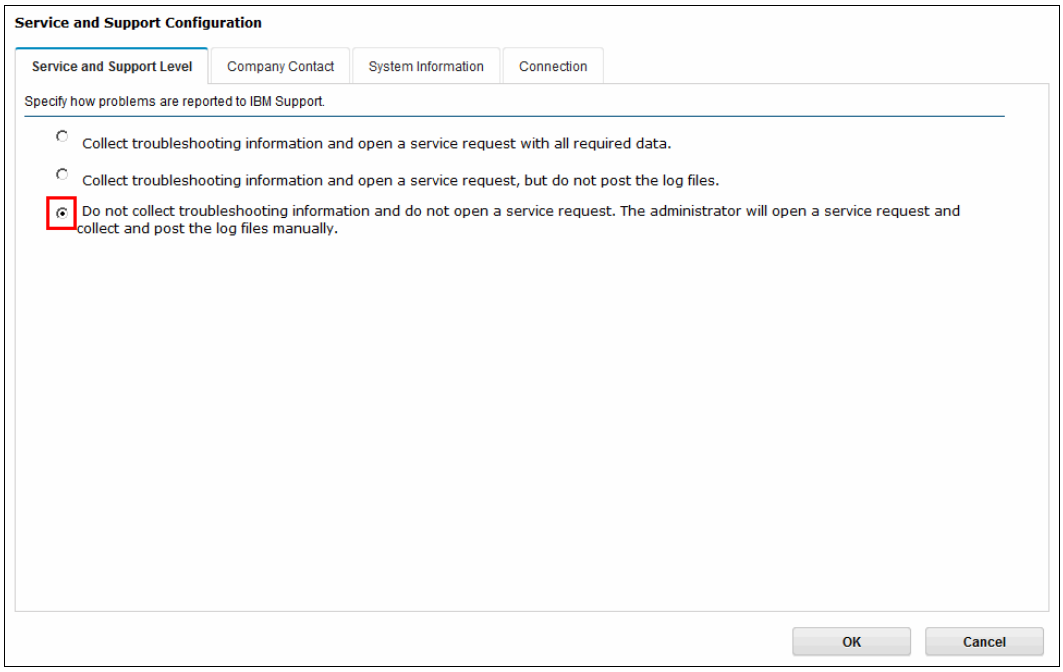


Figure 9-13 Disable the Service and Support Manager

Note: Be sure to re-enable the Service and Support Manager after the system update is complete.

9.3.3 Checking the system health status

As part of update planning, the system administrator must check the system health before any update operation and resolve any problems. There are three main categories for health status:

- PureApplication System health check
- PureApplication update readiness check
- PureApplication workload and catalog health check

The next section describes these categories in more detail.

System health check

A system health check is needed and any issues must be resolved. Perform a system health check 1 - 2 weeks before an upgrade to allow enough time to resolve any issues.

To perform a system health check, complete the following steps:

1. Click **System** → **System Troubleshooting** from the PureApplication System console.
2. Click **Collect System Logs** in the System Logs tab.
3. Select **System Health Check** from the list of options (Figure 9-14 on page 261) and click **Submit**.

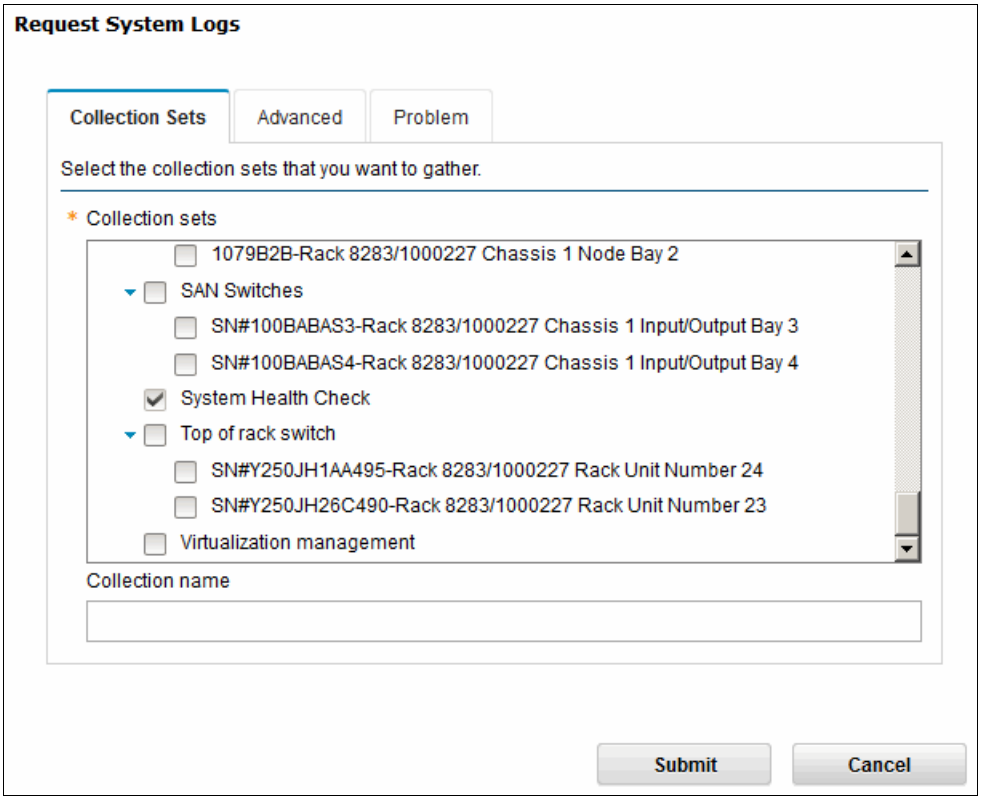


Figure 9-14 Request System Logs - System Health Check Logs

4. After the log collection is complete, click **System** → **File Viewer**.
5. Expand **PureSystem Manager** → **System Logs** → **Health_<date>.tgz**, and then click the `system_health.txt` file, as shown in Figure 9-15.

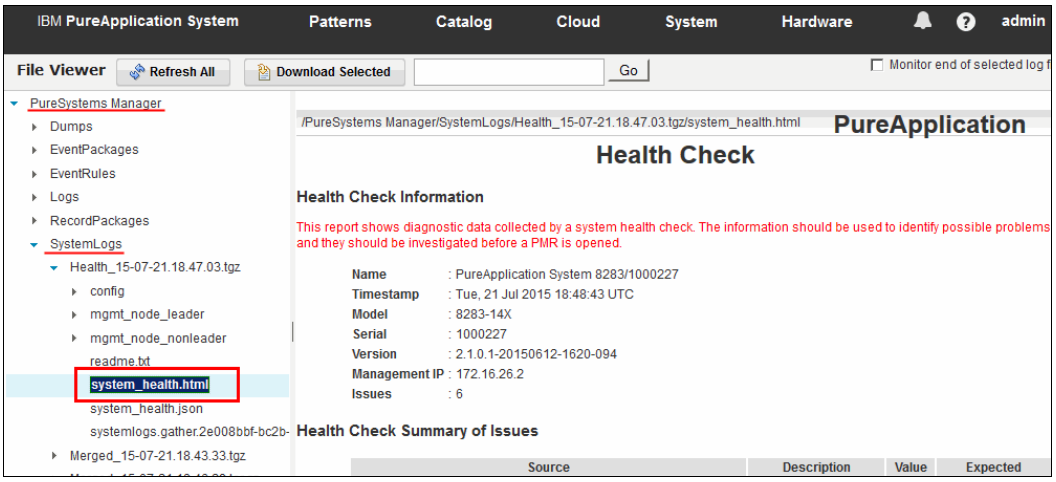


Figure 9-15 System_health.txt file

6. Resolve any issues that are found, and then rerun the system health check to ensure that there are no major issues to address.

System check for update readiness

A system check for update readiness is needed after the system fix pack import operation is completed. Resolve any issues that are found; otherwise, the **Check** button on the system maintenance page is not enabled. Perform a system check for update readiness 1 - 2 weeks before an upgrade to allow enough time to resolve any issues.

To perform a system check for update readiness, complete the following steps:

1. Click **System** → **System Maintenance** from the PureApplication System console.
2. Click **Check**, as shown in Figure 9-16.

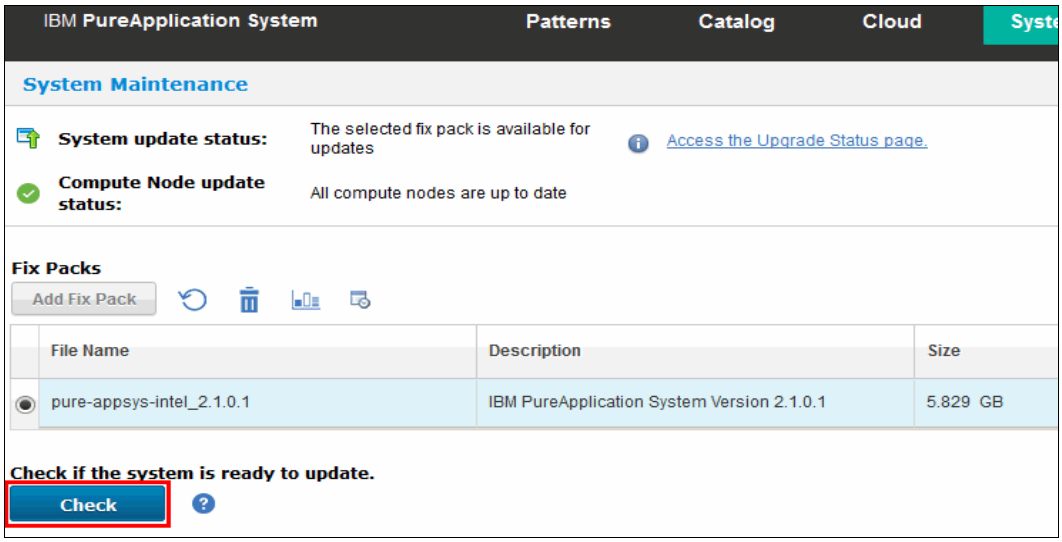


Figure 9-16 System Status Check button

Note: The *check operation* is not a background operation. Do not click any other link on the page while the operation is running. The status check takes about 5 - 10 minutes to complete.

3. After this task finishes, you see the result, as shown in Figure 9-17.

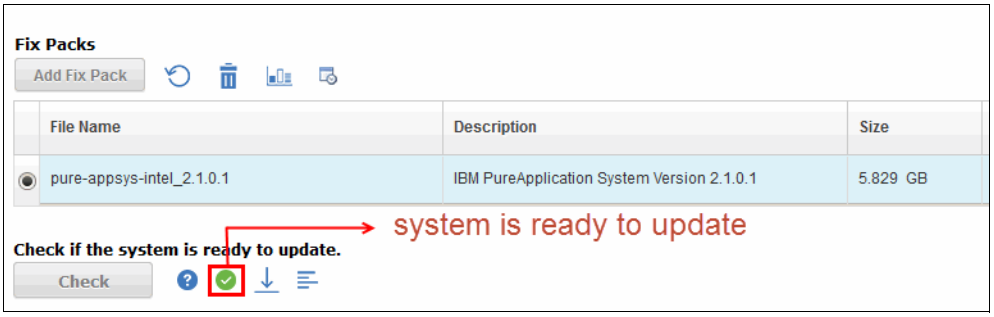


Figure 9-17 Check system status complete

4. If the operation fails, you see the result that is shown in Figure 9-18 on page 263.

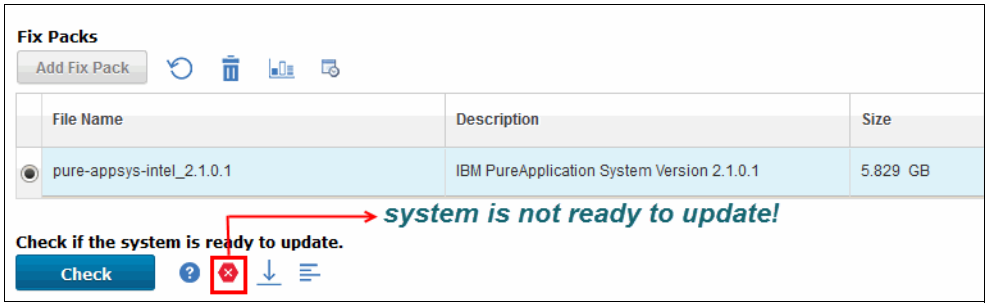


Figure 9-18 Check system status failed

For a failed check status, complete the following steps:

a. Click the **Show detailed results for the status check** icon, as shown in Figure 9-19.

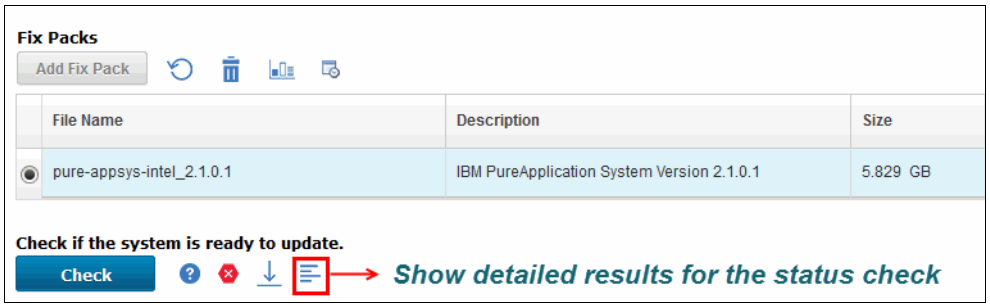


Figure 9-19 Show detailed results icon

b. A window with detailed results opens, as shown in Figure 9-20.

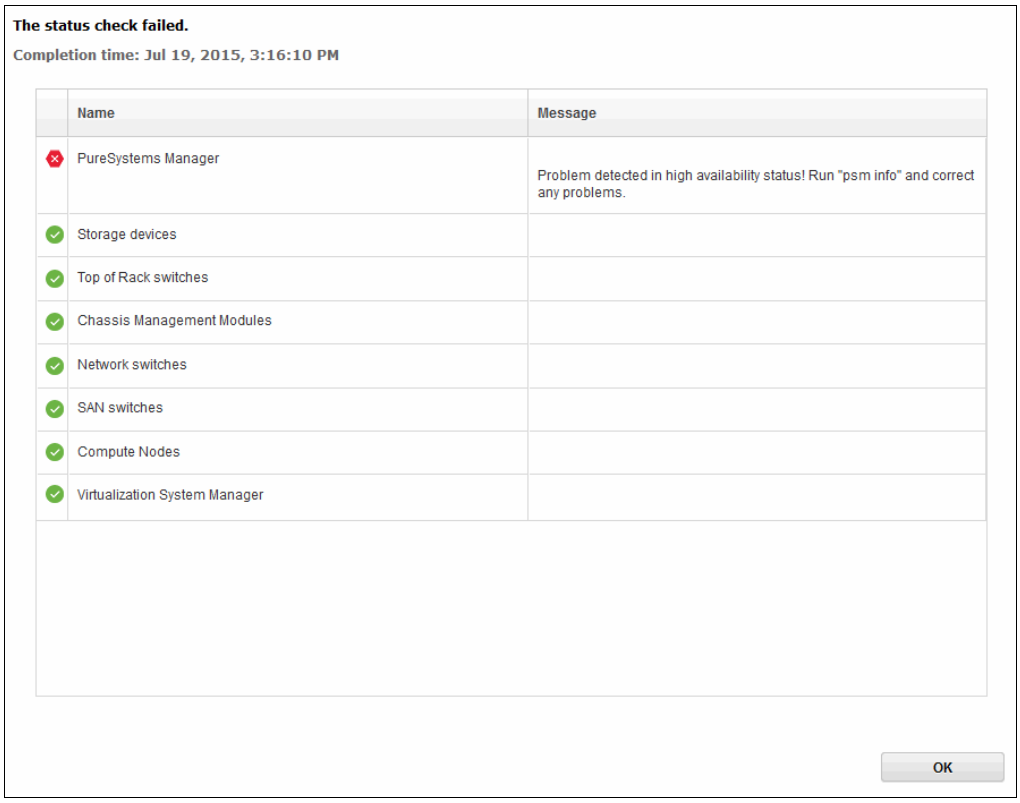


Figure 9-20 Detailed results for the status check

Note: Do not continue the update process unless the issues are fixed. Otherwise, the system update can fail and cause unexpected issues.

c. Click **OK** and go back to previous window.

d. Click the **Download the system check logs** icon, as shown in Figure 9-21.

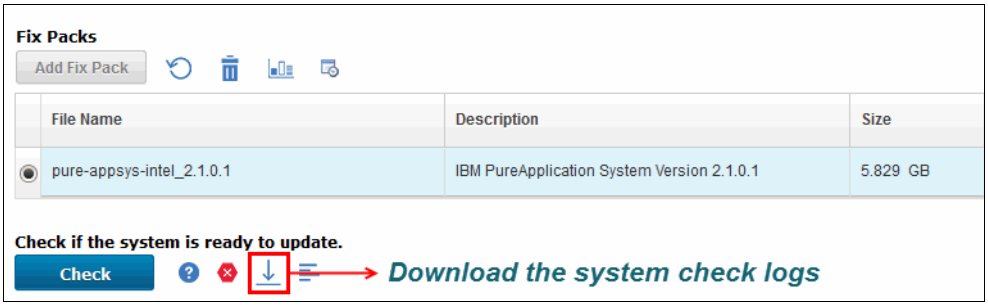


Figure 9-21 Download the system check logs

e. To solve the issues, open an IBM Service Request, attach the system check log file, and submit the Service Request to your IBM PureApplication Service Representative.

Workload and catalog health check for the update

Note: Only an administrator with full permissions to manage workload resources can perform workload and pattern checks.

The status of the virtual patterns, pattern types, virtual images, shared services, and deployed instances must be confirmed before any system and integrated component updates. During the workload and pattern checks, ensure that no page errors or blank windows are displayed on the system. To run a workload and catalog health check, complete the following steps:

1. Check the status of the virtual patterns:
   a. Click **Catalog** → **Pattern Types** from the PureApplication System console.
   b. Note of any pattern types that are in an unavailable or broken state. Delete or make them available before the update.
2. Check the status of the deployed instances:
   a. Click **Patterns** → **Pattern Instances (VIEW ALL)** from the PureApplication System console.
   b. Note any deployments that are in an error state. These deployments might not be able to be evacuated during the update of the compute node firmware, so you might need to stop or delete them.
3. Check the status of the compute nodes by clicking **Hardware** → **Compute Nodes** from the PureApplication System console, and ensure that the state of each compute node is displayed as *Available*.
4. Check the status of the management nodes by clicking **Hardware** → **Management Nodes** from the PureApplication System console, and ensure that the state of each management node is displayed as *Available*.

PureApplication Software: There is no compute node firmware updates for PureApplication Software because it is not an appliance. PureApplication Software uses compute resources of other virtualized environments, such as VMware or PowerVM.

There is no Management Nodes link in a PureApplication Software environment. Use Management Resource on the Hardware tab.

9.4 Applying the system update

This section presumes that anyone who is going to apply a system maintenance fix pack has read 9.2, “Planning system update” on page 245 and has followed the required steps before beginning the update operation.

Note: Before the installation of the system maintenance update, some actions are needed on the system. The general actions are listed in 9.4.1, “Applying a system update by using the system console” on page 266. However, each system fix pack release can require additional steps, depending on the PureApplication platform and model and the current version of the system. For a complete list of preparation steps, see the system update procedure of the fix pack level you are going to apply. For more information, see 9.2.2, “Obtaining the system update procedure” on page 246.

9.4.1 Applying a system update by using the system console

To apply a system update fix pack to the system from the system console, complete the following steps:

1. Click **System** → **System Maintenance** from the PureApplication System console to access system maintenance window.
2. Confirm that the system is ready for the updates to be installed by completing the following steps:
   a. Click **Check** again, even if you have already done so. (For more information, see “System check for update readiness” on page 262.)
   b. After the operation runs, click **Detailed status** to view the results. (For more information, see “System check for update readiness” on page 262.). Then, click **OK** to proceed.

   Failed check: If the check fails, do not proceed with the update. Resolve any issues, or the system update can fail and cause unexpected issues.

3. Select a step in the table under the **System Updates** tab to set the appropriate stopping point in the automated updates process, as shown in Figure 9-22. By setting stopping points, partial updates can be applied to individual components in shorter maintenance windows.

   If you select to stop at Step 3, for example, then Steps 1, 2, and 3 are completed before the automated updates process stops. Some fix packs can include only a PureSystem Manager update. In this case, there is only a Step 1, with no stopping point.

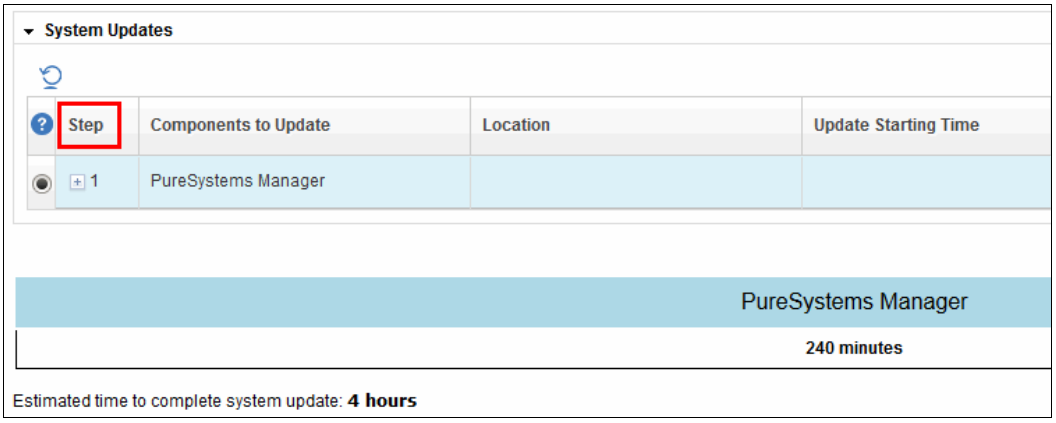


Figure 9-22 Select a step for the system fix pack update

4. Click **Update System**, and then click **OK**.
5. Accept the licensing terms by clicking the name of each license. Then, click **OK** to proceed.
6. The update, as shown in Figure 9-23 on page 267, starts in a few minutes. Updates to the firmware on the leader PureSystems Manager are installed first.

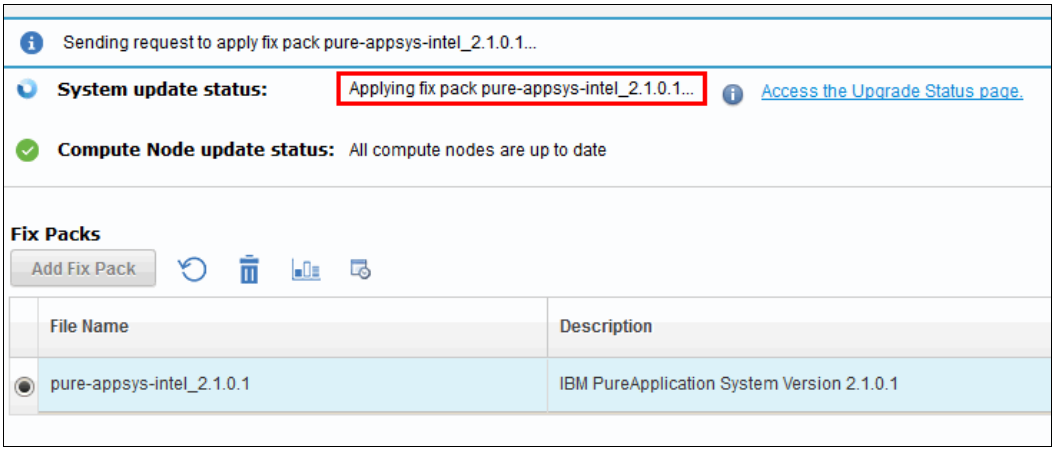


Figure 9-23 The system fix pack update begins

7. During the system outage, monitor the status of the PureSystems Manager update by clicking the **Access the Upgrade Status page** link, as shown in Figure 9-23.
8. After the leader PureSystems Manager is updated, click the **console** link to return to the console. If you are using the service notebook, the system console link does not work. Instead, go to `https://console/systemconsole/system/systemupdates`.
9. Proceed with the remainder of the system maintenance installation. See the details in the Update Progress section for information about the status. A new status is displayed each time a new change plan job runs. Refresh your browser window to see the status. The section is cleared when the fix pack is deleted.

> **Failure:** If a failure occurs, a log collection set containing the logs that are relevant to the failed component is automatically generated. You can download the log collection set by clicking **System** → **System Troubleshooting** on the system console.

10. When the automated system update is complete, update the compute nodes. Recall from "System maintenance overview" on page 243 that compute node updates are not applied automatically. They must be applied separately. For more information, see 9.4.3, "Applying compute node updates" on page 268.

9.4.2 Applying a system update by using the pure.cli

To apply a system update fix pack to the system by using CLI, complete the following steps:

1. Open the CLI from your client by running the following command:

```
# cd /directory/path/to/purecli/bin directory
# ./pure -h <PureApplication_Management_IP> -u admin
password:
>>
```

2. Use the following commands to apply the system fix pack:

```
>>> admin.fixpacks.list
<bound method FixPacks.list of [
  {
    "change_plan": (nested object),
    "created_time": "Jul 19, 2015 8:50:23 PM",
    "description": "IBM PureApplication Software Version 2.1.0.1",
```

```
    "fixes": None,
    "id": "82f8c5ef-ee77-4720-b256-336de4250c1e",
    "name": "pure-appsys-software_2.1.0.1",
    "short_description": "PureApplication Software 2.1.0.1",
    "size": 4509208576,
    "state": "ready",
    "unique_size": 4509208576,
    "updated_time": "Jul 19, 2015 8:56:16 PM",
    "url": "/admin/resources/fix_packs/82f8c5ef-ee77-4720-b256-336de4250c1e"
  }
]>
>>> fp = admin.fixpacks[0]
>>> admin.fixpack.install(fp)
{u'fix_packs_id': u'91ea3bc4-0829-45ad-93b0-f19ab75d1917', u'updated_time_raw':
1437093441862L, u'total_steps': 1L, u'state': u'ready', u'applying_roles':
u'IBM_CE,HARDWARE_ADMIN_WRITER', u'version': u'2.1.0.0', u'breakpoint':
u'General.Category', u'jobs': [u'8212cc9f-1024-4cf6-bb6c-c9726a973adb'],
u'created_time': u'Mon 20 Jul 2015 20:36:31.077 EDT', u'created_time_raw':
1437093391077L, u'completed_steps': 0L, u'id':
u'/admin/resources/change_plans/52cc9698-0231-4aee-bf10-2109ca770442',
u'updated_time': u'Mon 20Jul 2015 20:37:21.862 EDT', u'valid_breakpoints':
u'General.Category'}
>>>
```

3. You can monitor the status of upgrade from the system console or check the status of each change request in the change plan (the change plan can be seen in the job queue).

9.4.3 Applying compute node updates

Compute node updates are available for and applicable only to PureApplication System environments. PureApplication Software system updates do not include any compute node updates because compute nodes are managed by external virtualization systems.

The steps for updating the compute nodes can vary depending on the state and cloud group membership of the compute nodes. The differences are described in the following list:

- Unassigned

  No VMs are deployed to unassigned compute nodes; therefore, the compute nodes can be safely updated.

- Assigned to a cloud group with active high availability

  The VM instances can be evacuated from one compute node to another with no impact because the required resources within the cloud group are set aside in advance.

- Assigned to a cloud group with inactive high availability

  The cloud groups do not have reserved resources to host the VM instances when the compute node is in maintenance mode. You can assign an additional compute node to the cloud group, and evacuate the VM instances to this compute node. Another option is to stop and store the VM instances for the duration of the update process.

Review the following information about updating compute nodes:

- If you choose to update compute nodes in parallel, there cannot be any VMs on each compute node.
- If you do not choose to update compute nodes in parallel, the compute node must be assigned to a cloud group with active high availability and must not have pinned VMs.

- If the parallel update or the serial update options do not work, stop and store some VMs. You still must run the compute node update from the System Maintenance window to load the new VMware ESXi master image on the Storwize V7000 components.

Note: Before applying the compute node updates, read “Workload continuity and compute node update” on page 256. Otherwise, an outage can occur and disrupt your running workload.

To update compute nodes, complete the following steps:

1. Click **System** → **System Troubleshooting** from the PureApplication System console.
2. Expand the **Compute Nodes** tab at the bottom of the window.
3. Select the compute node or nodes to be updated.
4. Click **Update Compute Nodes**, as shown in Figure 9-24.

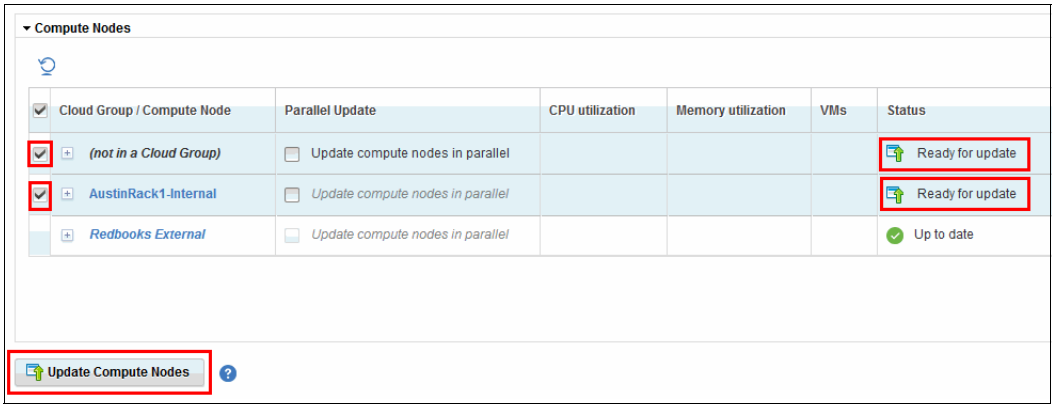


Figure 9-24 Update the compute nodes

5. You can monitor the status of compute node updates in this window.
6. Click **Update Compute Nodes** to view the log file. Click **Download log** if wanted, as shown in Figure 9-25.

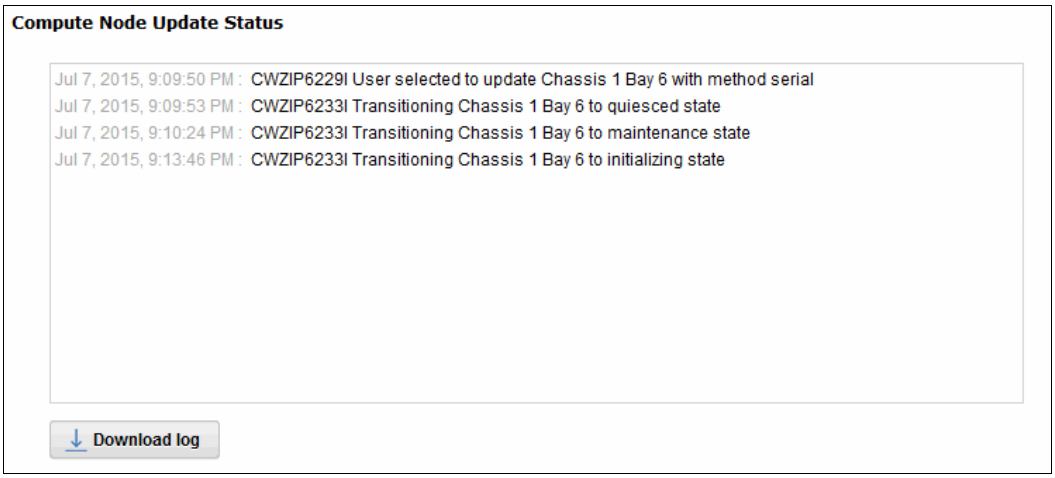


Figure 9-25 Compute Nodes Update Status window

9.5 Applying an integrated components update (group fix)

You can update pattern types by using the system console or CLI of the PureApplication System or PureApplication Software. If you are not familiar with the PureApplication CLI, see Chapter 13, “IBM PureApplication System command-line interface and REST API” on page 355.

9.5.1 Dependency checks

Some pattern types depend on other pattern types. For example, System Monitoring for SOA (itmsoa) depends on the System Monitoring (itm) pattern type. After the update of System Monitoring for SOA pattern type, you cannot enable it if the System Monitoring (itm) pattern type is not updated and enabled first. Similarly, the Database Performance Monitoring Tools pattern type (opmtools) depends on the Database Performance Monitoring pattern type (opm).

As a fundamental pattern type, upload and enable the Foundation Pattern Type first, as all other pattern types depend on the Foundation pattern type.

Aside from those pattern type dependencies, there are a few other pattern types that are also specific to the PureApplication platform. Red Hat Update Service (rhus) is one of them. It can be installed and used only on PureApplication System W1500/2500, and not on PureApplication System W1700/2700.

At the time of writing, the following pattern types are available only for PureApplication System W1500/2500 and PureApplication Software.

- Red Hat Update Service
- Docker Pattern Type

9.5.2 Updating integrated components by using the system console

You can import the updated pattern types to the PureApplication System and PureApplication Software from the system console by completing the following steps:

1. Click **Catalog** → **Pattern Types** from the PureApplication System console.
2. Click the **New** icon on the toolbar.
3. Click **Browse** to select the `.tgz` file to import as a pattern type.
4. After the upload is complete, click **Accept license** in the right pane. (If the status is unavailable, click **Enable**.)
5. The pattern type status is available, as shown in Figure 9-26. If it is not, you cannot deploy any instances by using that pattern type.

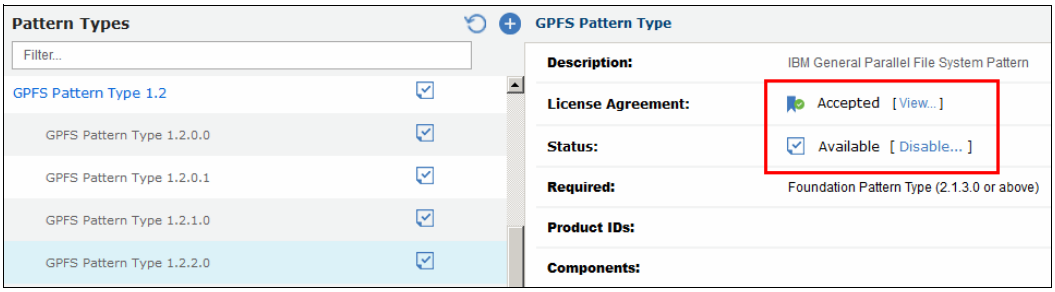


Figure 9-26 Pattern type details

Note: Most modern browsers limit uploads to about 2 GB or a maximum of 4 GB. Some pattern type update files exceed this limit. So, they cannot be uploaded from your local hard disk drive by using a browser. Instead, use the pure.cli to upload them.

9.5.3 Updating integrated components by using the pure.cli

The more practical way of updating integrated components and pattern types is to use CLI. To update your pattern types by using CLI, complete the following steps:

1. Put your integrated components update files into a directory on a Linux or UNIX system.
2. Open the CLI and connect to the PureApplication management IP address, as shown in Example 9-6.

Example 9-6 Connect to the PureApplication management IP

```
# cd /directory/path/to/purecli/bin
# ./pure -h <PureApplication_Management_IP> -u admin
password:
>>
```

3. Replace x.x.x.x with the appropriate version number and run the command that is shown in Example 9-7. Ensure that you specify the correct location of the pattern type `.tgz` file, or the command fails.

   Here are the file names for the pattern types:

   - `webapp-x.x.x.x.tgz`
   - `foundation-x.x.x.x.tgz`
   - `dbaas-x.x.x.x.tgz`
   - `javaapp-x.x.x.x.tgz`

Example 9-7 Command to update the pattern type

```
>>> deployer.patterntypes.create("/tmp/foundation-x.x.x.x.tgz")
```

4. After the update command runs successfully, the information that is shown in Example 9-8 is displayed on the pure.cli console.

Example 9-8 Display after successful execution

```
{
"description": "DESCRIPTION".
"name": "NAME".
"shortname": "ptype".
"status": "avail".
"version": "x.x.x.x"
}
>>>exit
```

5. You might need to accept the license and then enable the pattern type. An example of accepting the license and enabling the pattern type for System Monitoring pattern type is shown in Example 9-9.

Example 9-9 Accept the license and enable the pattern type

```
>>> ptype = deployer.patterntypes.list({'shortname':'itm','version':'1.0.4.0'})[0]
>>> ptype.acceptLicense()
```

```
>>> ptype.enable()
>>>
```

6. Repeat steps 2 on page 271 - 4 on page 271 for each pattern type that you are going to update and enable.

You can find a well-written, ready to use CLI recipe for updating pattern types in 13.3.1, “Importing and loading group fixes” on page 370.

You can find more information about updating pattern types at the IBM Knowledge Center at the following website:

http://www.ibm.com/support/knowledgecenter/SSCRSX_2.1.0/doc/iwd/apt_updatevapp.dita?lang=en

9.5.4 Upgrading deployed pattern type instances

There are several situations where you might want to apply the updates from a pattern type to a deployed pattern. For example, if you update a pattern type, you might want to apply the changes to a pattern that is based on the pattern type. To update deployed pattern type instances, see the IBM Knowledge Center at the following website:

http://www.ibm.com/support/knowledgecenter/SSCRSX_2.1.0/doc/iwd/apgst_upgradepattern.dita?lang=en

Updating shared service instances

The IBM Knowledge Center has instructions about updating shared service instances. To update shared service instances, see the IBM Knowledge Center found at the following website:

http://www.ibm.com/support/knowledgecenter/SSCRSX_2.1.0/doc/workloadconsole/t_upgradess.dita?lang=en

9.6 Performing system health status check (post-upgrade)

As part of the maintenance procedure, the system administrator performs system health status checks after an upgrade operation to resolve any problems that are found:

- Post-upgrade system health check.

  For more information about performing the health check, see 9.3.3, “Checking the system health status” on page 260.

- PureApplication workload and catalog health check.

  After the system health check, perform a workload and catalog health check. For more information, see “Workload and catalog health check for the update” on page 265.

- Check system events.

  Check for any changed or new events that might occur after you complete the system update procedure. Warning events might be produced as hardware components are updated and restarted. After the system is updated, review all newly generated events and delete the ones that originated from the system update actions.

- Perform a test deployment.

  Perform a test deployment and validate that the deployment is successful and the instance is running without issues. After this validation, be sure to delete that instance.

9.7 Maintenance-related preferred practices

This section describes preferred practices for installing maintenance updates for systems and integrated components:

- Plan your upgrade with an IBM Service Representative.

  Before planning your updates, contact an IBM Service Representative to determine whether credentials for an IBM Service Representative (ibmeng) are required for the fix pack level you plan to apply.

- Keep your system up-to-date.

  To take advantage of new features, improvements, bug fixes, and new application releases of integrated components, upgrade and keep your system updated.

  When you upgrade to the latest IBM PureApplication System platform, you avoid the risk of security issues that arise from using an older release of the PureApplication platform, and you benefit from new features, improvements, usability, and system performance.

- Check the release Notes and new functions.

  Before an update, review *What's new in this release* and the release notes for that particular release or fix pack. Reviewing the details of the new functions and improvements helps to ensure that you do not encounter surprises with the new features, so that you can adjust your update plan and post-update actions.

- Update the test and development environment first.

  If you have another PureApplication System in your test environment, upgrade your test environment first. By doing so, the issues that might be encountered during the test environment upgrade can be resolved before upgrading your production environment.

- Take a backup before the update.

  Before the system fix pack and integrated component update, take system component and workload component backups. It is your responsibility to configure backups and take system and component backups. Also, use your existing backup tools to take to a backup of the data within workload instances.

- Check the state of the hardware.

  Perform an exhaustive health check of the hardware before you begin updating the system. Make sure that there are no failed drives, no failed devices, and no amber LEDs. If you find failures or errors, record the details of each one so you can recheck for them after the system is updated.

- Do not update any system that has issues.

  As part of planning and preparing for the upgrade, apply a system health check and an update readiness check. If one or more of the checks fails, do not proceed with the update, but resolve the issues first. Otherwise, the system update can fail and cause unexpected issues.

- Set maintenance mode.

  Before you install system maintenance, put the system in maintenance mode to prevent operations from continuing to run even if the system maintenance process fails.

- Delete system events

  Export all current important and critical events before you update the system. Check for any changed or new events that might occur after you complete the system update procedure.

Warning events might be produced as hardware components are updated and restarted. After the system is updated, review all newly generated events and delete the ones that originated from the system update actions.

- Maintain resources.

  If you have a cloud group that does not have enough physical capacity for the evacuation to occur during the compute node update, you can temporarily move a compute node from a different cloud group with active high availability (if any) to the specified cloud group and then perform the update. Do not forget to reassign the temporarily moved compute node to its original cloud group.

  Cloud group membership of compute nodes is expected to be the same before and after you install maintenance updates. Ensure that there are no changes in the configurations after the system is updated.

- A short maintenance window.

  For some system fix pack releases, the estimated time to complete the system update can exceed your maintenance window. A preferred practice is to apply some of the updates during one maintenance window, then apply the remaining updates during another maintenance window. To do so, use the PureApplication System maintenance feature stopping points so you can select stopping points for the system update.

- Pay attention to any compute node updates.

  Some system fix packs include compute node firmware updates, and some do not. When compute node updates are included in the fix pack, an outage can occur during the updates of the compute node when an instance evacuation is required and the cloud group does not have enough physical capacity for the evacuation to occur. The cloud group is not highly available and the updates cannot complete without affecting the instances. Some instances are stopped and cannot be restarted until the updates are complete and the cloud group resources are fully restored. So, be careful, and read 9.2.7, “Time and workload continuity planning” on page 255 before starting the system fix pack update. A good plan saves time and mitigates risks.

- Use the system maintenance history.

  To report upgrade activities, which can include storage, network switch, management nodes, and virtualization system, use the maintenance history table. This table shows how long the system update took and the start and end dates and times for the update of each component. You can also use maintenance history to track all the upgrades that are applied to the system.

10

Backup and restore

You can run backup and restore operations on your system configuration in the IBM PureApplication System and PureApplication Software environments. You can run scheduled backups or run backups on demand, and restore backed up data as needed. This chapter provides guidance on backing up and restoring various types of data that are associated with PureApplication System and PureApplication Software. Specifically, the chapter outlines how to back up and restore component data and system data.

This chapter covers the following topics:

- Data categories
- Types of backups
- Configuring backup functions
- Performing backups
- Restoring backup data
- Backing up and restoring by using the CLI or REST API
- Backup instances and application data

© Copyright IBM Corp. 2015. All rights reserved.

10.1 Data categories

When considering the backup of the PureApplication System, there are several data categories that must be taken into account. Figure 10-1 shows the different data categories that must be addressed in the backup and restore of the system and the workload.

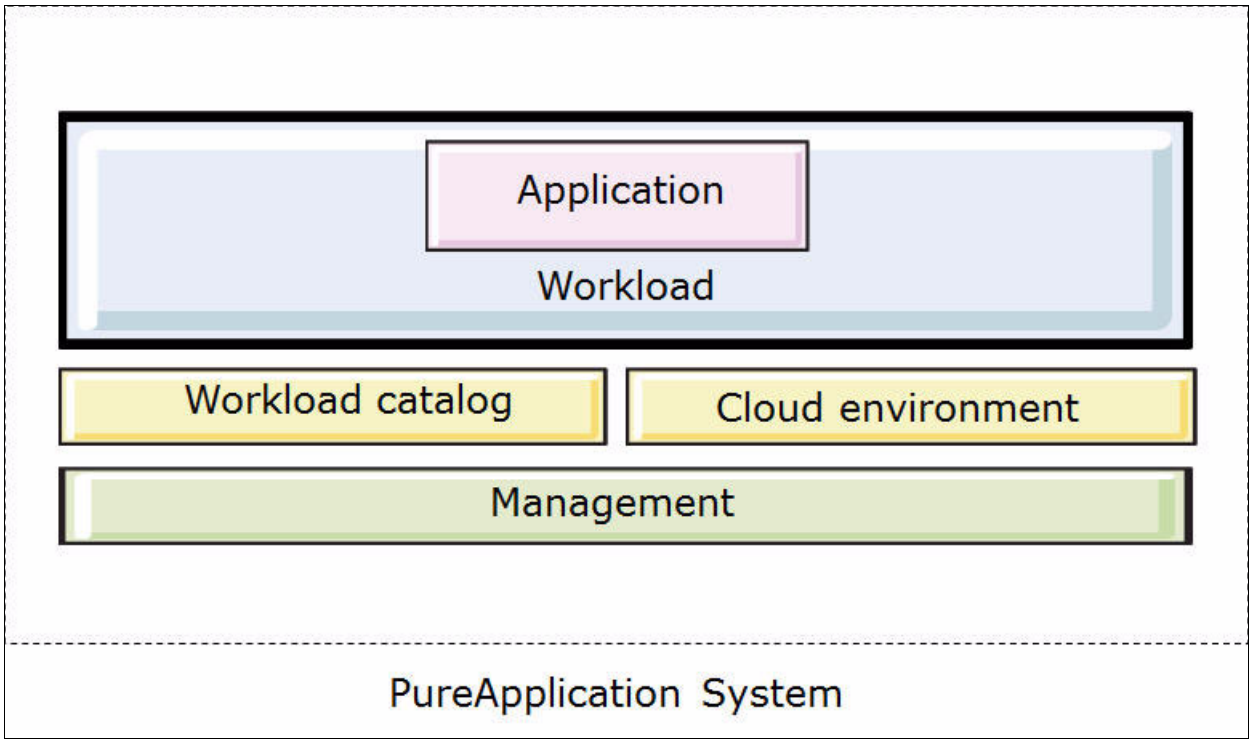


Figure 10-1 PureApplication data categories

10.1.1 Management function data

PureApplication System management functions control the entire system, virtualizing the hardware into resources for the cloud environment and providing a runtime environment for the workload functions. These management functions contain setup and configuration data that must be backed up.

10.1.2 Cloud environment data

The cloud environment is defined by creating and configuring cloud components, that is, IP groups and cloud groups, as shown in Figure 10-2 on page 277. This configuration organizes the system's resources by runtime environments into which the workloads can be deployed.

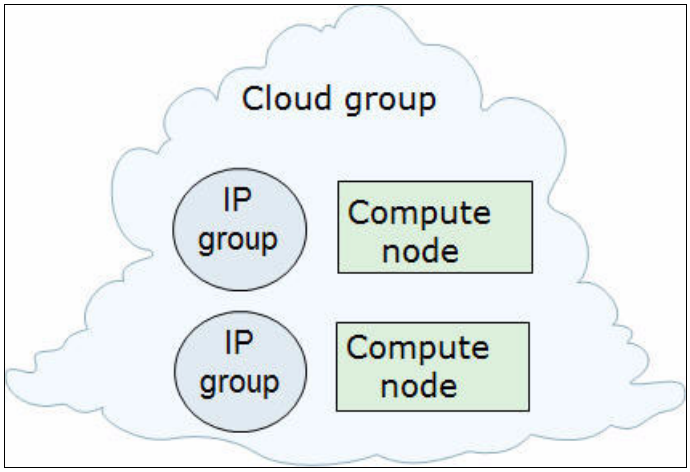


Figure 10-2 IP and cloud groups

Cloud configuration data must be backed up separately from the system's management data. Creating cloud resources is described in Chapter 3, “IBM PureApplication installation” on page 41.

Note: A *cloud group* is made up of a logical group of compute nodes, where the virtual machines (VMs) of the pattern instances are deployed. By having multiple cloud groups, PureApplication System and PureApplication Software provide multi-tenancy and a level of isolation for different teams to deploy their patterns

Cloud groups contain one or more *IP groups* for assigning IP addresses to the VMs of deployed patterns. An IP group is a range of IP addresses that are either assigned to the VMs by the system or can be specified by the deployer. During the creation of IP groups, you provide the VLAN that is configured to connect to the external data center, the gateway, the netmask, and DNS. You then define a pool of IP addresses within the subnet (an IP group within a subnet) that is available to PureApplication System.

10.1.3 Workload catalog data

The workload catalog contains the individual workload components and assets that are shown in the PureApplication console under Catalog and Patterns. The catalog contains the following items:

- Virtual images
- Virtual application patterns and pattern types
- Plug-ins,
- Virtual System Patterns (VSPs)
- Script packages
- Add-ons
- Fixes

These pattern-level components do not include the deployed pattern instances, but rather the patterns and the assets that are needed to deploy these instances. The workload catalog also includes environment profiles, which are needed to deploy the patterns into the cloud environment.

10.1.4 Workload (pattern instance) data

Workload data is the deployed patterns instances, the contents of the running VMs, and their relationships. Workload data is a considerable portion of the data that is stored by PureApplication System and PureApplication Software. Workload data is voluminous because VMs are large and applications often load much data from their databases. Workload data can grow when applications maintain much session data when they have a many concurrent users.

When following standard middleware installation and application deployment practices, if the runtime environment is lost, re-creating it is difficult, time-consuming, and error-prone. Because of patterns In PureApplication System, workload data is fairly easy to replace, so backup and restore is not as important.

With PureApplication System and PureApplication Software, if an application environment fails, workload managers can easily replace it by redeploying the pattern and redeploying the application into the pattern instance. Therefore, on PureApplication System and PureApplication Software, workload data does not need to be backed up.

10.1.5 Application data

The internal state of each application running as a workload on the system must be backed up. The state is application-specific, and typically is the application's databases, but it can also be any application-specific state, such as the following ones:

- Application configuration
- Application file system
- Logs
- Other key application artifacts

This backup is the same kind of application backup that is needed for applications running in middleware on traditional hardware. The difference for a virtualized application running in PureApplication System and PureApplication Software is that the pattern must include the backup software or agent to back up application-specific data to external storage.

10.1.6 Data categories example

Figure 10-3 on page 279 illustrates a singular set of management data:

- A cloud environment that is divided into two cloud groups. (They contain other components, such as IP groups.)
- A workload catalog containing two patterns. (These patterns might be VSPs or virtual application patterns. They contain lower-level components, such as virtual images, script packages, and plug-ins.)
- Two patterns that are deployed as workloads, each deployed into a cloud group.
- Application data in each of the workloads.

If any or all of the data on the system is lost or corrupted, that loss can cause some or all of the system to not function correctly. The goal of the backup is to enable you to replace the flawed data, restoring it to its previous state, so that the affected parts of the system once again function correctly.

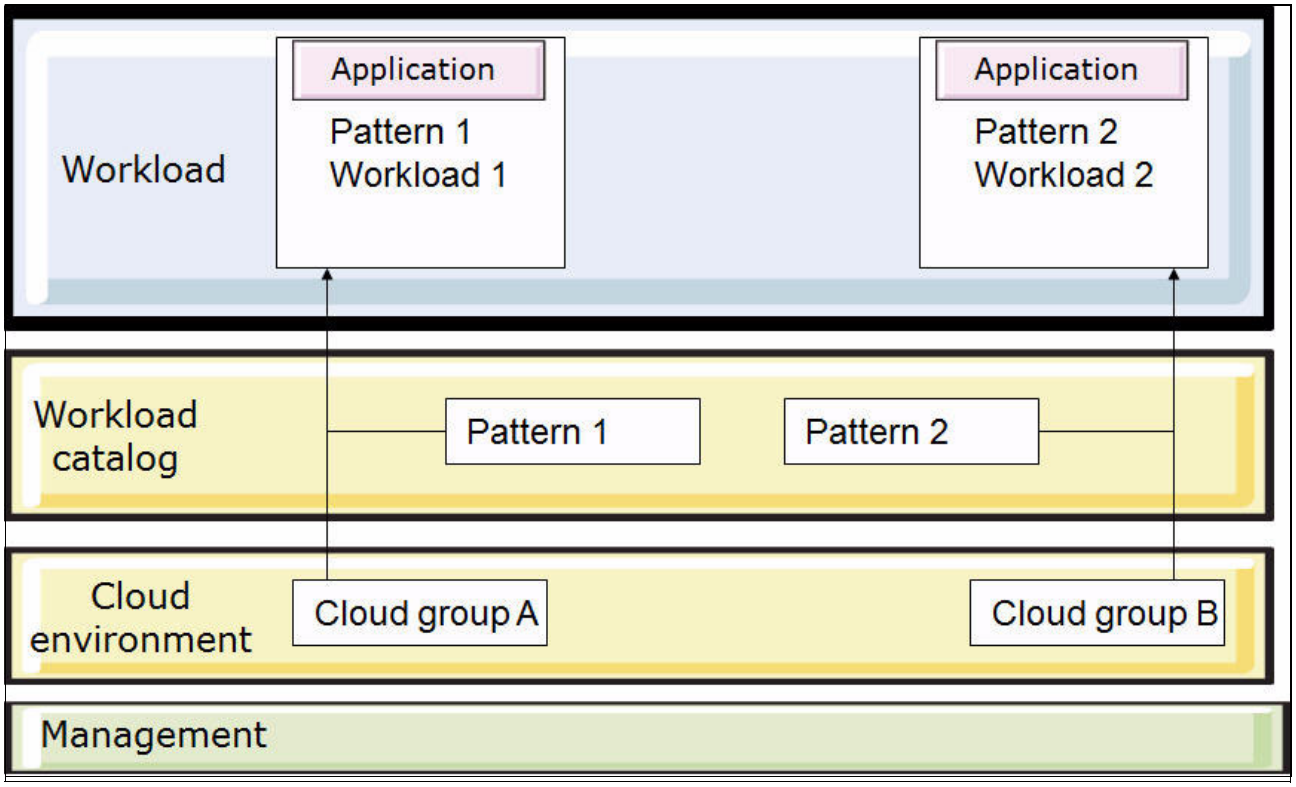


Figure 10-3 Data category example

10.2 Types of backups

PureApplication System and PureApplication Software provide robust backup and restore functions to back up the data to an external Secure Copy Protocol (SCP) server, and the ability to restore them from the backup server. Backup files can be encrypted and authentication can be a user ID and password or by using private key-based authentication.

There are two types of backups:

- System backup

  This is a simple monolithic snapshot for restoring the entire system. It includes management, cloud environment, and some of the workload catalog data. Virtual images, workload instances, and application data are not included.

- Component backup

  This type of backup stores some of the system and all the workload artifacts in individual files. It can be restored individually, copied, stored in source code control, and so on.

10.2.1 Required permissions for backup and restore

Users and groups with different role-based permissions have access to different resources in PureApplication System and PureApplication Software. Here is the list of roles and permissions that are needed to access backup and restore functions. The user must have all these permissions to perform backup and restore functions.

- Workload administration full permission
- Cloud group administration full permission

- Hardware administration full permission
- Security administration full permission

10.3 Configuring backup functions

This section describes how to set up and configure the backup function of PureApplication System and PureApplication Software.

Here are the high-level steps to activate the backup function for either a system or component backup. Subsequent sections describe the details of each of these steps.

1. One or more external backup servers (running Linux, AIX, or Windows) must be configured for Secure Shell (SSH) protocol capability to accept connections and transfer data from remote systems. The backup server must have enough storage to perform backups of system or components. The same backup server can be for used multiple backups in different directories and encryption options. For more information about the backup server requirement, see the PureApplication System IBM Knowledge Center at the following website:

   http://www.ibm.com/support/knowledgecenter/SSCRSX_2.1.0/doc/iwd/brt_remote_system_info.dita?lang=en)
2. From the PureApplication System backup and restore console, create one or more backup locations, pointing to the backup server that is defined in step 1.
3. From the PureApplication System backup and restore console, create one or more backup configurations by using the defined backup locations, type of backup (system or components), and the backup schedule.

10.3.1 Creating backup locations

The assumption in this section is you are creating a backup location for the first time. Later, backup locations can be changed or added as needed.

To create a backup location, complete the following steps:

1. Click **Systems** → **Backup and Restore**, and then click **Setup**.
2. The window to create a location opens. Click **Create New**.
3. Enter the following values:
   a. Provide a meaningful name for the backup location.
   b. Provide the host or IP address of the external backup server.
   c. Provide the path where the data files are backed up. If the directory does not exist in the backup server, it is created.
   d. Provide the user name to access the backup server. The user must have access to the path where the backup files are created.
   e. The authentication type can be either a user ID and password or a private key. Based on the kind of authentication, provide either the password for the user, or add the private key with a maximum length of 8192 bits.

f. Optionally, enable encryption for the backup data. For system backup or component backup that includes security users and groups, encryption must be enabled. The encryption certificate, private key, and paraphrase can either be provided or the system can generate one. If encryption is enabled, another dialog box opens under the table for you to provide additional certificate and private key settings for your backup location:

   i. Use existing private key and certificate.

   ii. Generate new private key and certificate: When you select this option, you are prompted to enter a password (and enter it a second time for verification). The new certificate and private key are generated when you click **Generate**. The expiration date for this certificate is set automatically to 366 days from the date of creation.

   iii. Upload your own private key and x509 compliant certificate.

   iv. Renew expired certificate using existing key.

For this publication, two backup locations were created. Both backup locations point to the same server, but use different directories and encryption options:

1. Figure 10-4 shows the backup location that is called *RedbooksBackupComponents* with no encryption. This location is used for component backup.

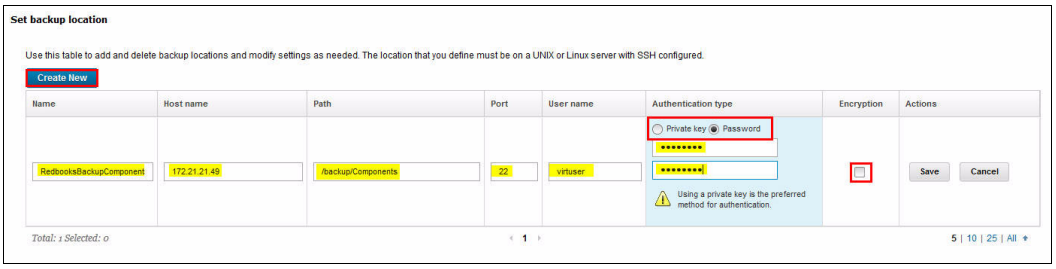


Figure 10-4 Backup location with no encryption

2. Figure 10-5 shows the backup location that is called *RedbooksBackupSystem* with encryption enabled. This location is used for system backup. This option uses PureApplication to generate the certificate, and a key that is based on the paraphrase that is provided. To upload your own certificate and key, you can use the **Upload your own private key and certificate** option, as shown in figure Figure 10-6.

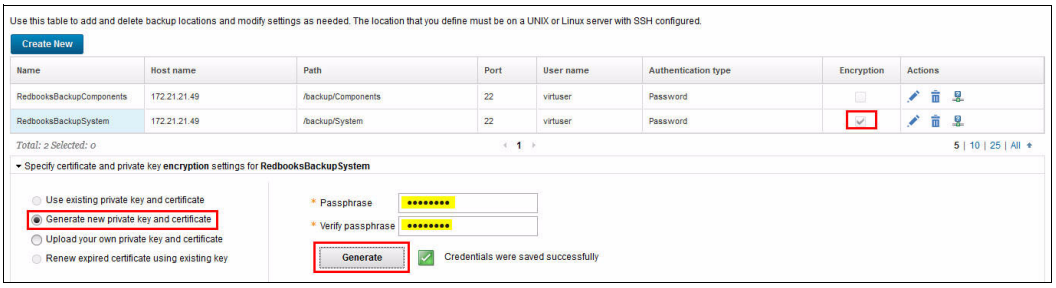


Figure 10-5 Backup location with encryption

Figure 10-6 Option to upload you own private key and certificate

After they are created, the backup locations should look Figure 10-7. From this window, you can edit or delete the location, and test the connection to the backup server.

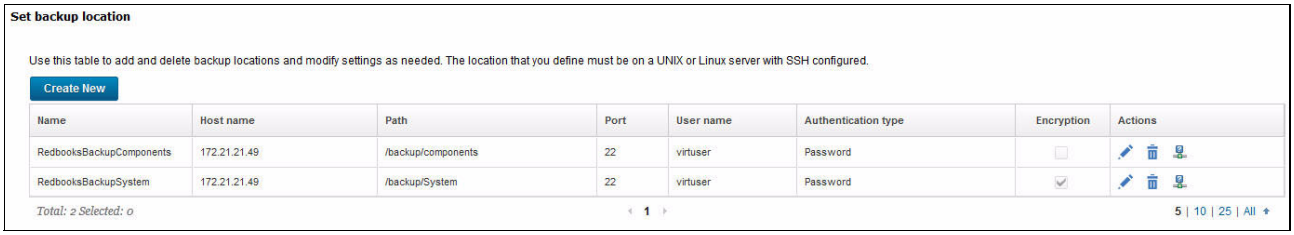


Figure 10-7 List of backup locations

After backup locations are created, the next step is to create backup configurations pointing to the backup locations.

10.3.2 Creating backup configurations

To create a backup configuration, complete the following steps:

1. Click **Systems** → **Backup and Restore**.
2. Click **Create New**.

3. Enter the following values:
   a. Provide a meaningful name for the backup configuration.
   b. Choose any backup *Location* that is created.
   c. Specify the type of the backup: system or component backup.
   d. Specify the backup schedule either “on demand” or a fixed schedule. The fixed scheduled backup has a start date, backup time, daily or weekly schedule, and, optionally, an end date.
   e. Optionally, you can select the users who are notified after the backup is complete. The PureApplication System “Mail delivery” SMTP server must be configured to enable the notification through email. The notification provides status information about complete or incomplete backup jobs.

10.3.3 Creating a component backup configuration

If component backup is selected, you can select several resources in the Workload, Cloud, and Security categories:

- Workload category:
  - Add-ons
  - Database images
  - Database patterns
  - DB2 fix packs
  - Emergency fixes
  - Pattern components
  - Pattern types
  - Script packages
  - System plug-ins
  - Virtual application patterns
  - Virtual application templates
  - Virtual images
  - Virtual system patterns
  - Virtual system patterns (Classic)
  - Virtual system templates
- Cloud category:
  - Cloud groups
  - Environment profiles
  - IP groups
  - Volume groups
  - Virtual appliances
- Security category (requires an encrypted backup location):
  - Users
  - User groups

For this publication, the following component backup was created by completing the following steps:

1. Click **System** → **Backup and Restore**.
2. Click **Create New.**

3. Create the component backup configuration that is named *RedbooksBackupComponents*. As different resources are selected, the backup data storage and estimated time changes. Your configuration should look similar to Figure 10-8.

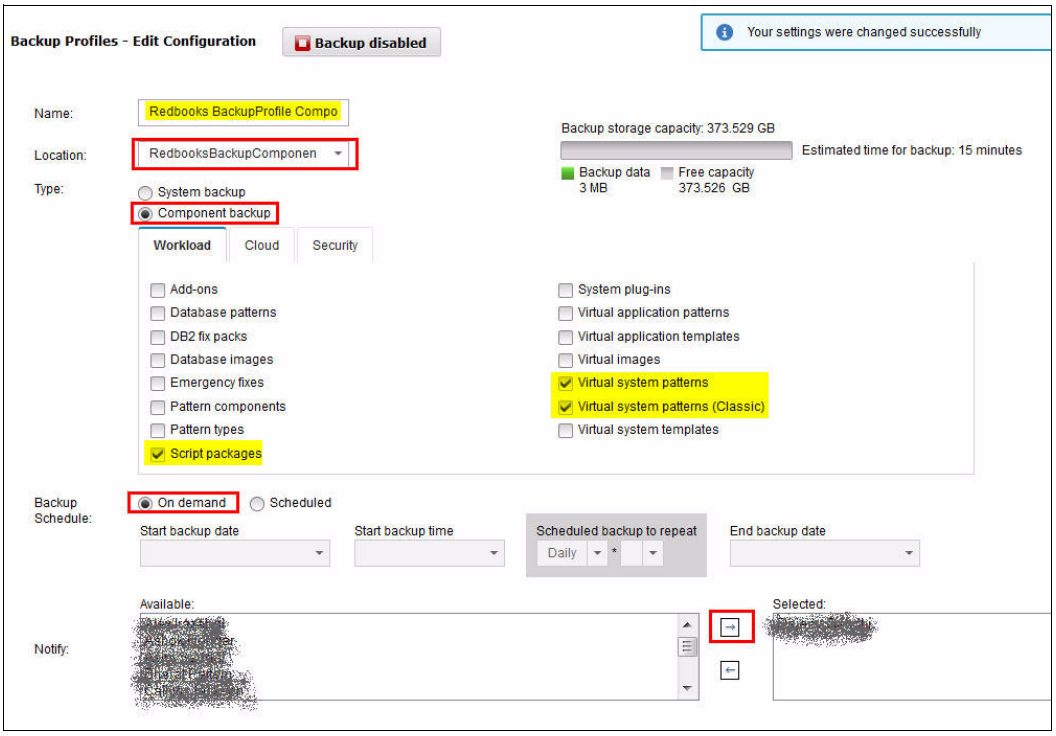


Figure 10-8 Backup profile workload components

a. For the workload components, the **Script packages**, **Virtual system patterns**, and **Virtual system patterns (Classic)** are selected.

b. For the cloud components, which are shown in Figure 10-9, the cloud groups, environment profiles, IP groups, and one storage volume group containing some storage volumes are selected. The backup of the storage volume in the volume group allows the backup of the storage data on the external server.

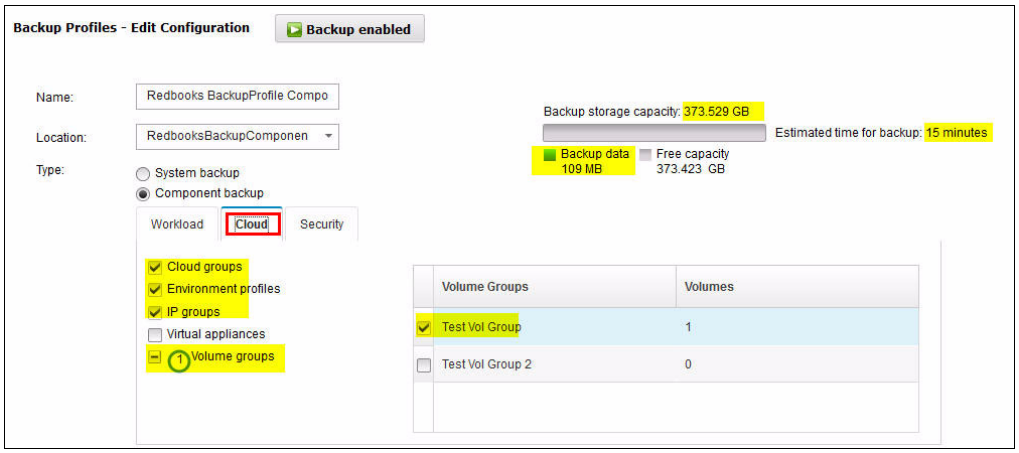


Figure 10-9 Backup profile - cloud components

c. The backup schedule is set to **on demand**.

d. Enable the backup by clicking **Backup disabled**.

10.3.4 Creating a system backup configuration

For this publication, the following backup configuration was created by using an encrypted backup location and completing the following steps:

1. Click **System** → **Backup and Restore**.
2. Click **Create New**.
3. Create the system backup configuration *Redbooks BackupProfile System*, as shown in Figure 10-10.

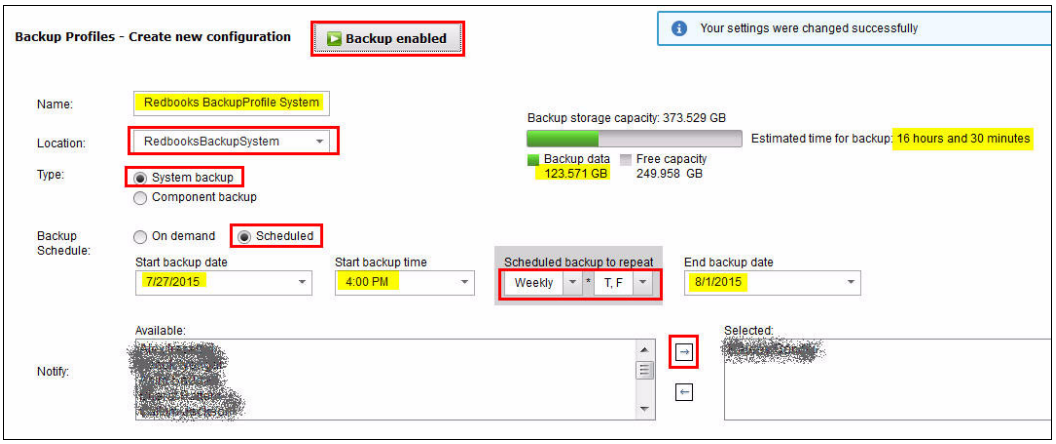


Figure 10-10 Backup profile - system

4. The location **RedbooksBackupSystem** is selected for system backup. The location must be encrypted for system backup.
5. A fixed schedule of every Wednesday and Friday at 12:15 am was configured with a start and end date.

10.4 Performing backups

After the backup resources (locations and configurations) are defined, both backup configurations are enabled and active, and can back up the resources based on the schedule. Figure 10-11 shows the backup configurations that are enabled in PureApplication System.

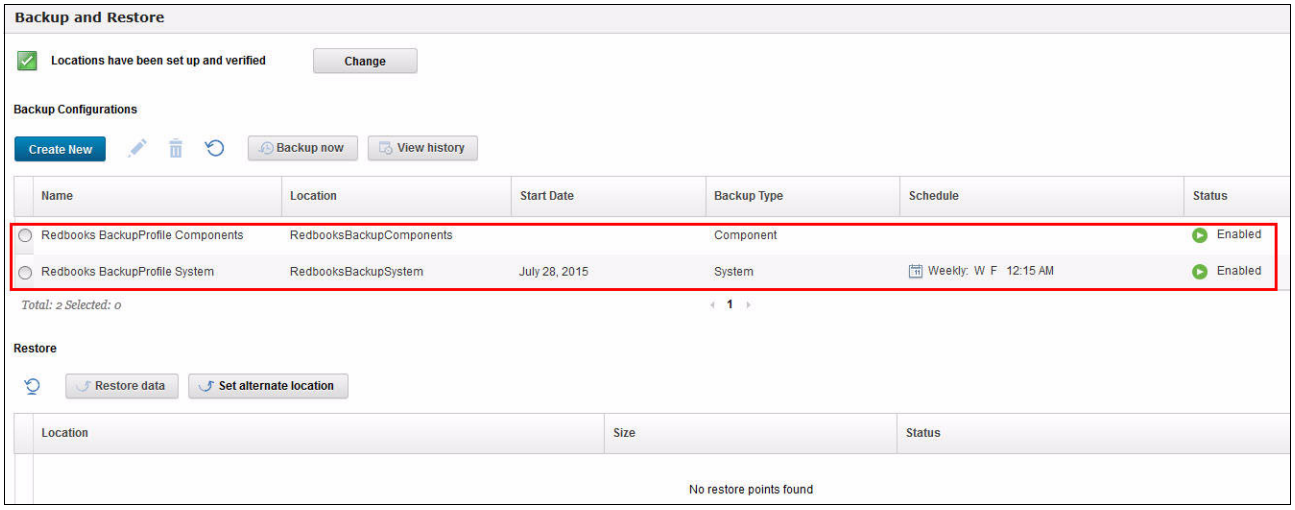


Figure 10-11 List of backup configurations

The backups that are configured to run at a scheduled time are started automatically by PureApplication System.

You can also perform on-demand backups simply by selecting the backup configuration and clicking **Backup now**, as shown in Figure 10-12. The backup job is created and progress is shown on the console.

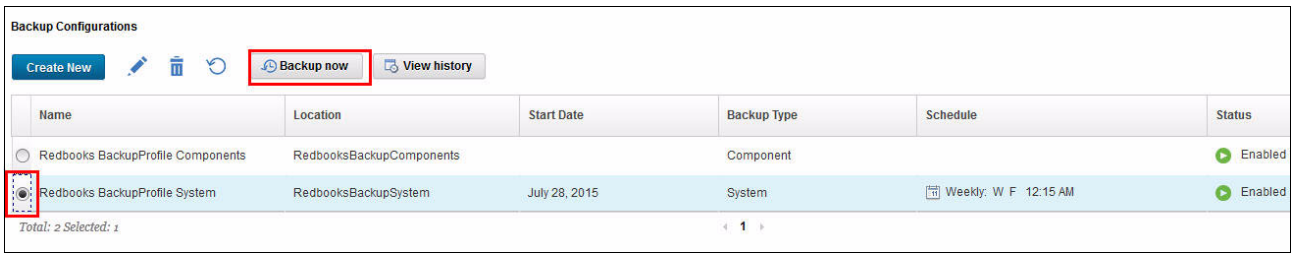


Figure 10-12 Immediate backup option

10.4.1 Performing a system backup

When backing up system data, the first backup provides the baseline backup data and the subsequent backups are delta backups. The subsequent backups store only the changes to the system configuration since the last baseline backup. The backup process automatically runs either the baseline or delta backups based on the data on the backup server. Over time, the delta backup size increases because it is the delta against the last baseline, and not the last delta backup. At that time, a new baseline can be created.

A system backup to a new location creates a baseline backup.

The system backup process runs in three jobs:

1. Backup 1 of 3: This job initiates the backup process and queues the second of the three jobs to start.
2. Backup 2 of 3: This job prepares the data for backup. To ensure a consistent copy of management databases and other artifacts, new tasks are blocked or placed in a pending state while this job runs. This job might run for 10 - 15 minutes based on the resources on the rack. After all target data is defined, the third job starts.
3. Backup 3 of 3: This job compresses, encrypts, and transfers the target data to the specified remote backup system. This job is not a blocking job and other changes to the console are allowed during this stage.

While the system backup is performed, you can monitor the progress of the backup from the console, as shown in Figure 10-13.

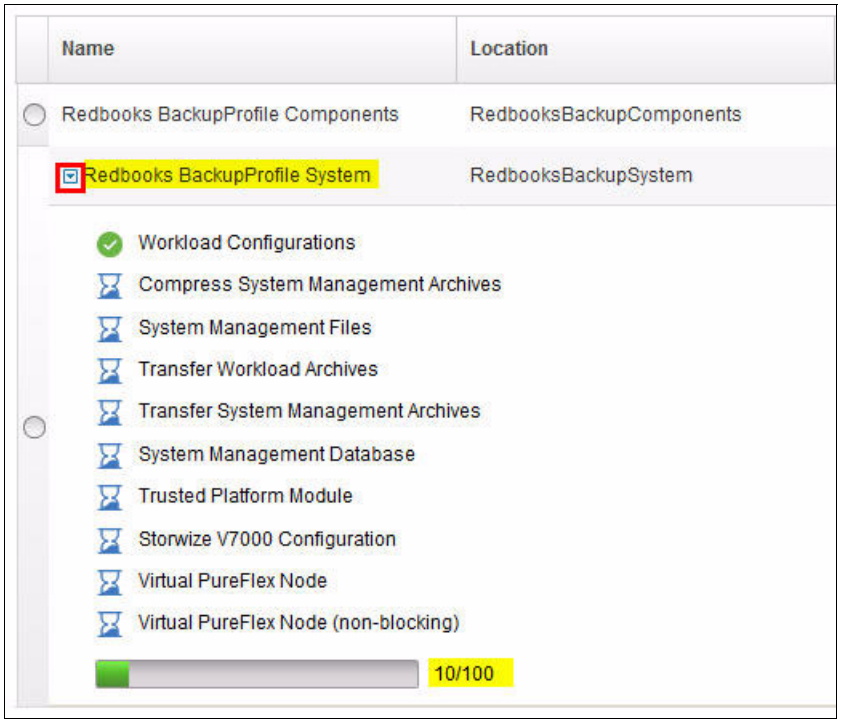


Figure 10-13 Monitor the backup job progress

System-level backup directory structure

System-level data is backed up to the backup server and organized in a unique directory structure. The general directory structure for system level backup data is shown in the following sections.

System-level data that is associated with internal IBM Workload Deployer functions

The system-level data that is associated with internal IBM Workload Deployer functions is found in the following location:

```
/<backup_location_path>/<rack_ID>/<iwd-version number like
5.1.0.1>/backupStorage/<iwd_artifact>/<timestamp>
```

`<iwd-artifact>` represents several artifacts.

PureSystems Manager data

The PureSystems Manager (PSM) data is found in the following location:

`/<backup_location_path>/<rack_ID>/backupStorage/<timestamp>`

The `<timestamp>` folder contains full baseline system-level backups of PSM for each backup operation in several directories (databases, key, IBM Tivoli Provisioning Manager, and Storwize V7000 storage system).

Pattern type data

The pattern type data is found in the following location:

`/<backup_location_path>/<rack_ID>/backupStorage/ptype/<ptype_name>/<timestamp>`

Subdirectories in this directory represent folders for each pattern type.

Volume group data

Volume group data is found in the following location:

`/<backup_location_path>/<rack_ID>/backupStorage/volumegroup/<timestamp>`

All time stamps are in milliseconds. The details for the file system and how to convert the time stamp into a readable string is available in the PureApplication System IBM Knowledge Center at the following website:

`http://www.ibm.com/support/knowledgecenter/SSCRSX_2.1.0/doc/iwd/brc_backup_dir_system.dita?lang=en>`

10.4.2 Performing a component backup

The first time components are backed up, a baseline is created for all the resources that are selected for component backup. For subsequent component backups, only changes are added to the backup. Component backups are *Automated Component Export* operations. A component backup is a single job that is non-blocking, as shown in Figure 10-14.

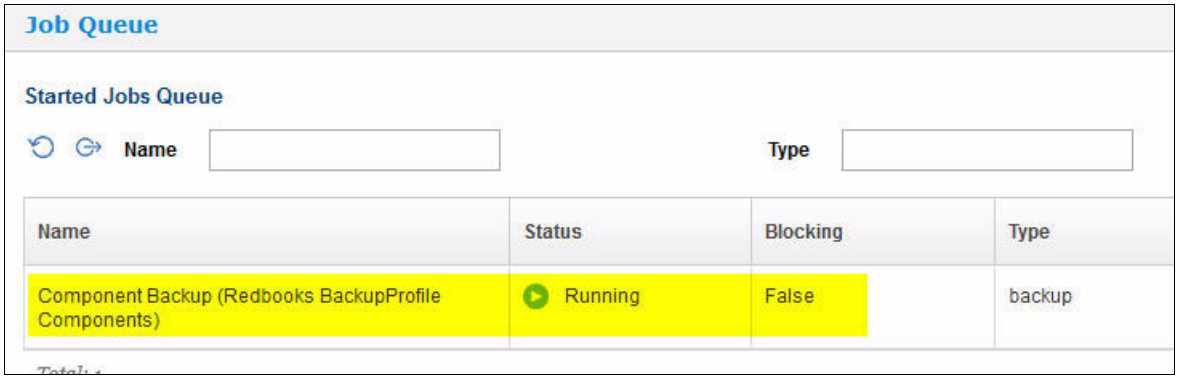


Figure 10-14 Component backup job

While the component backup is performed, you can monitor the progress of the backup from the console, as shown in Figure 10-15 on page 289.

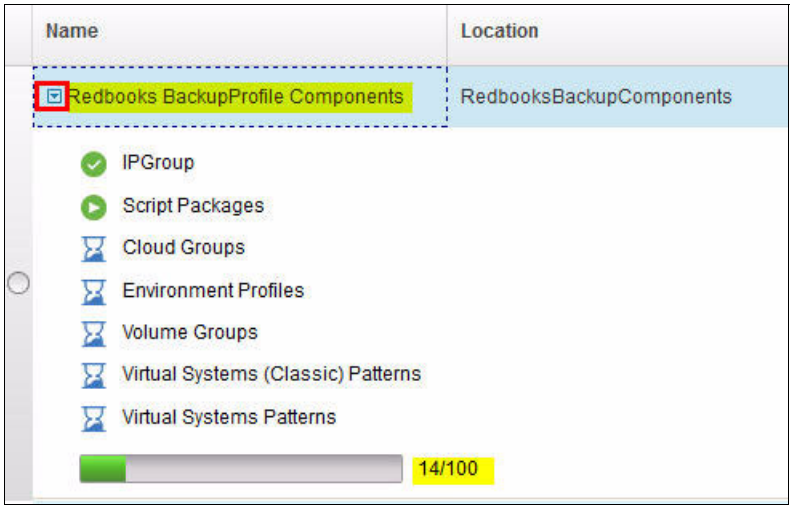


Figure 10-15 Component backup job progress

On the backup server, each type of component resource has its own directory within the specified backup file system. Storage volumes that are contained in the storage volume group that are selected for backup are backed up with compression.

Some examples of directories for component resources on the backup server are shown in Figure 10-16. This example is based on the components that are selected for this publication's scenario for the component backup in Figure 10-8 on page 284.

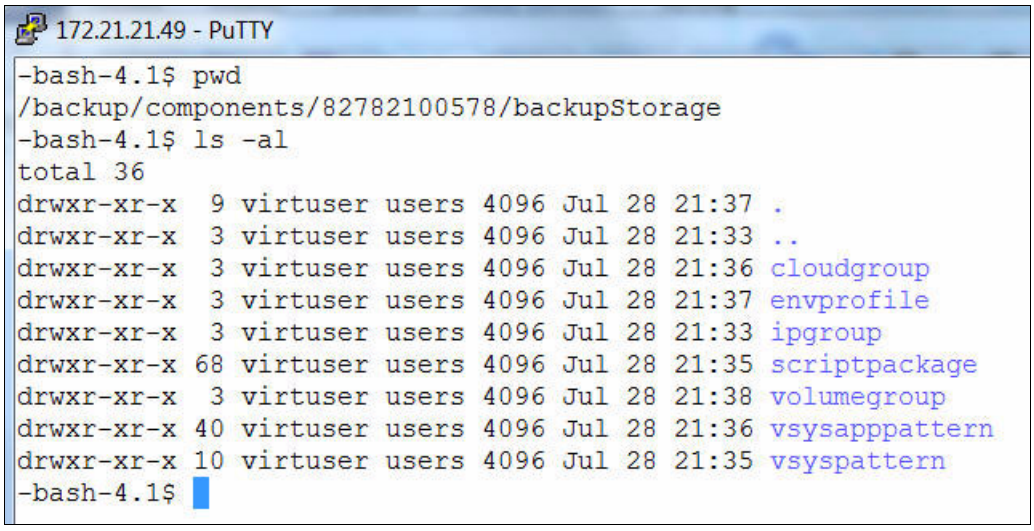


Figure 10-16 Component backup directories

Component-level backup directory structure

Component-level data is backed up to the backup server and organized in a unique directory structure. Here is the general directory structure for component backup data:

`/<backup_location_path>/<rack_ID>/backupStorage/<comp_type>/<internal_ID>/<timestamp>`

The `<comp_type>` is the component resource being backed up, such as an add-on, script packages, and vsyspatterns.

The <internal_ID> is a simple integer value that is assigned sequentially as new components are defined

10.4.3 Viewing the backup history

From the Backup and Restore console, you can view the history of all the backups for each backup configuration. Figure 10-17 shows the *Redbooks BackupProfile System* configuration.

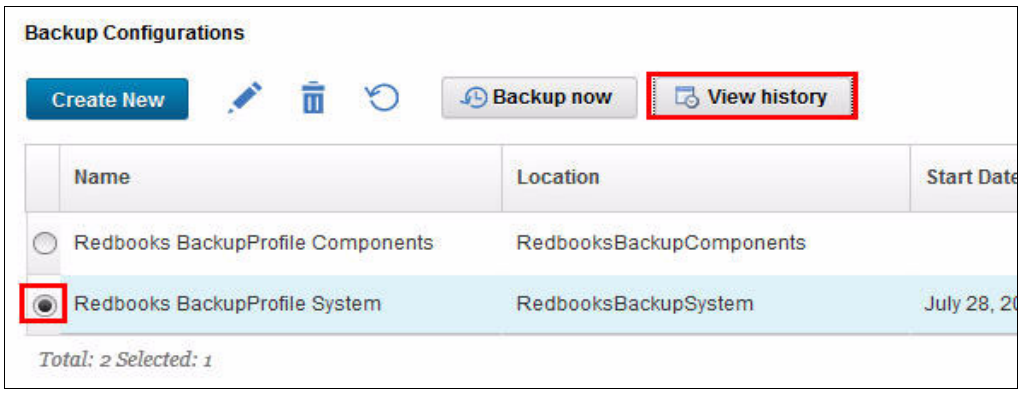


Figure 10-17 View Backups

If you click **View History**, the results of the system backup history are shown in Figure 10-18. The delta backup is smaller and faster than the earlier baseline backup (dated July 27 in the figure).

Figure 10-18 History of system backups

Similarly, Figure 10-19 shows the history of the component backup of *Redbooks BackupProfile Components*. In this example, the oldest backup has data. Subsequent backups have no data (size 0) because no changes were made to the components being backed up.

Figure 10-19 History of component backups

Preferred practices for backing up data

This section lists some preferred practices for backing up system and component data.

System backups

Here are some preferred practices for system backups:

- Run backups regularly, either daily or few times a week. The first backup takes longer, but subsequent delta backups run quickly.
- Once a week, archive a new baseline and delete some of the older ones. Over time, the delta backup size increases because it is the delta against the last baseline, and not the last delta backup. At that time, create a new baseline. As the time duration for the delta operations get close to the baseline backup, this is another indication that you should create a new baseline.

Component backups

Here are some preferred practices for component backups:

- Run backups on a regular schedule or on demand as needed. Component-level data is backed up only if there are changes since the previous backup.
- Over time, perform general archiving and pruning tasks on the backup server to delete data that is no longer needed and free up additional storage. The backed up data can be archived on a tape or other backup solutions such as SAN. Ensure that encryption keys for archived backups are stored safely and are not lost.

10.5 Restoring backup data

After system and component data is backed up to the backup server location, it can be restored from the console. Figure 10-20 shows the restore section of the console. The list of backup locations that have backed up data is visible in the table. For this example, the *RedbooksBackupSystem* and *RedbooksBackupComponents* locations are available. If there were backup locations but no data that is backed up to those locations, they do not appear in the Restore location table. Along with the location, the size of the data and the availability status is shown. You may also set an alternative location.

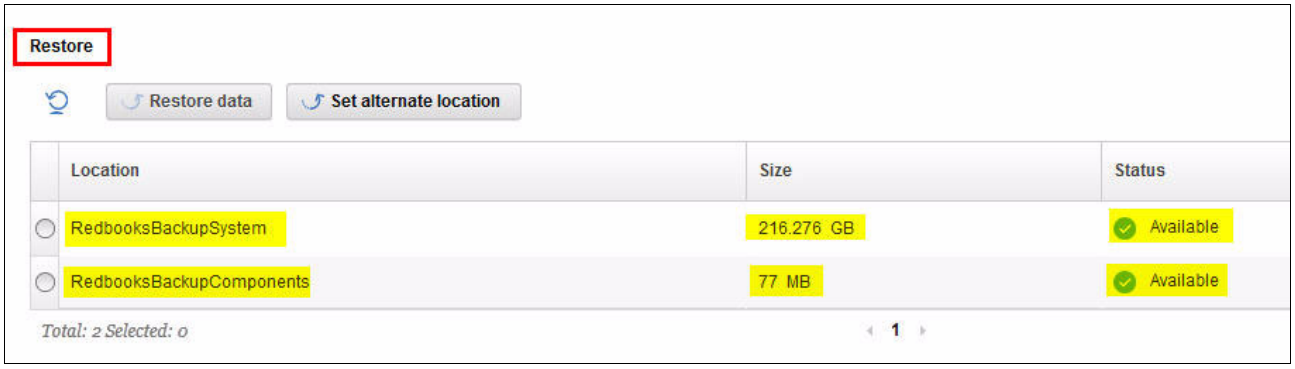


Figure 10-20 Restore location table

From the restore table view, you can select the location from where the backup data is to be restored, as shown in Figure 10-21. Click **Restore Data**.

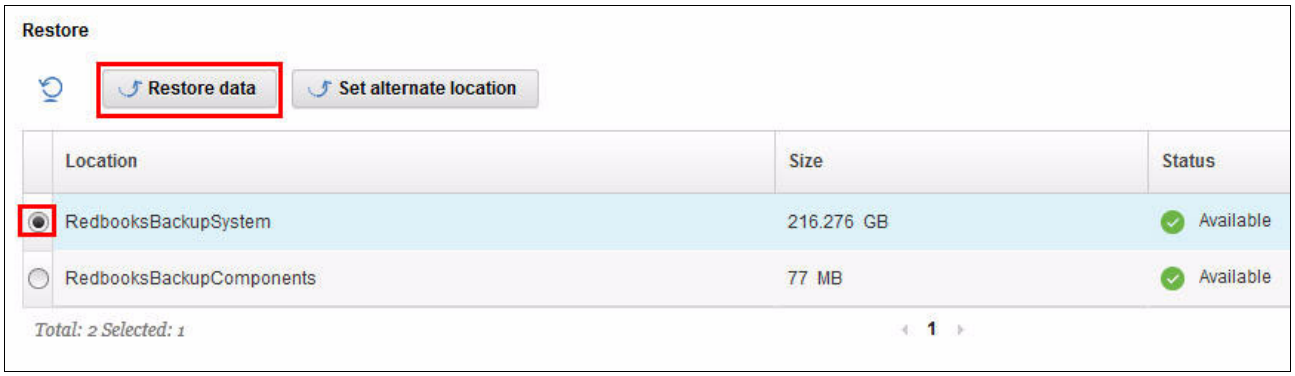


Figure 10-21 Restore location

The *Restore Configuration* window for the selected backup location opens, as shown in Figure 10-22.

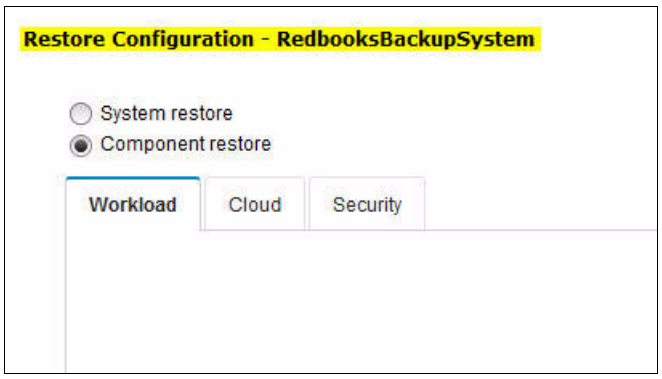


Figure 10-22 Restore configuration

10.5.1 Restoring system data

System restore is used for critical issues, such as internal PSM database corruption, loss of both PSMs, and other unrecoverable conditions. System restore is not to be used lightly. This function is used for disaster situations where the complete system must be restored from the backed up data.

Only authorized IBM Support representatives can and should perform system-level restore operations. If you must perform the restore yourself, do so *only* under the guidance of IBM Support. Contact IBM Support for assistance in restoring system-level data. The IBM representative can back up from the baseline system backup or from any delta backup (which also includes the baseline backup). During the backup, the PSM is offline.

Figure 10-23 on page 293 shows the system restore window.

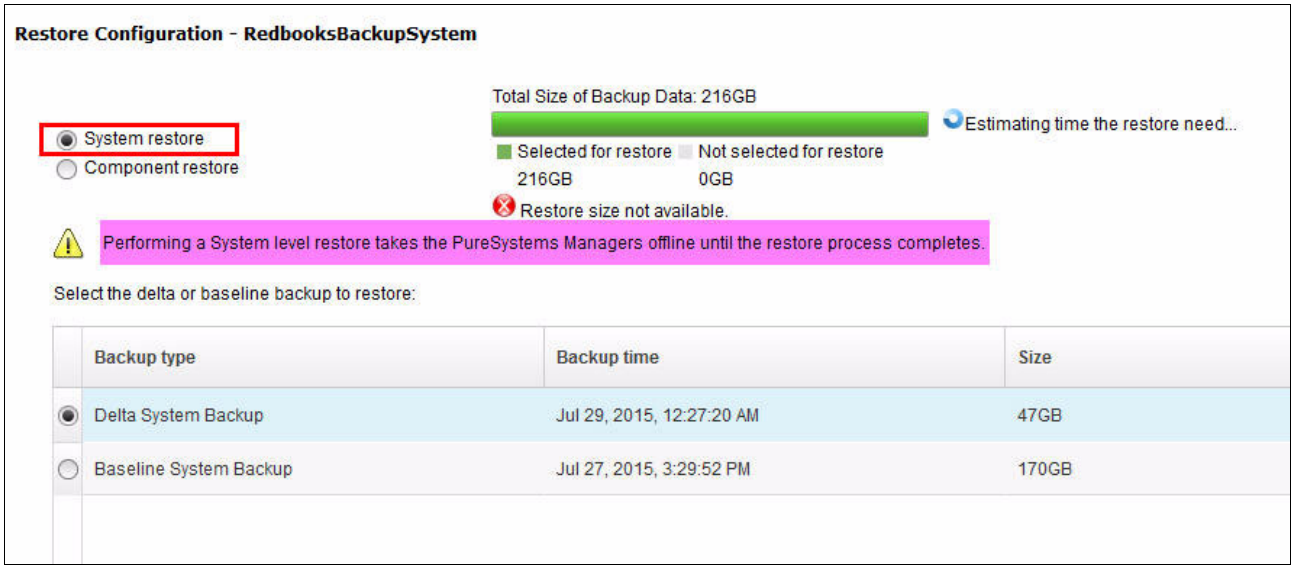


Figure 10-23 Restore system configuration window

10.5.2 Restoring component data

You can select a backup location that has component data to restore all or a subset of components that are backed up, which allows a much granular restore of the component resources.

To restore the example components in this publication, complete the following steps:

1. Click **System** → **Backup and Restore**.
2. Select the *RedbooksBackupComponents* location (where component resources are backed up). *Restore Configuration - RedbooksBackupComponents* is displayed.
3. Select **Component Restore**. All the components that can be restored are shown, as shown in Figure 10-24.

Figure 10-24 Restore components

4. From this window, you can select the components to be restored.

You can use the restore function to display all or only new or changed component resources from what is in the PureApplication System. Figure 10-25 shows the list of component resources that are in the backup, but either changed or new to the rack. Component backups are *Automated Component Export* operations. Virtual Images and Virtual Appliances create additional jobs to perform the OVA or Image import operations, and appear as additional jobs in the job queue.

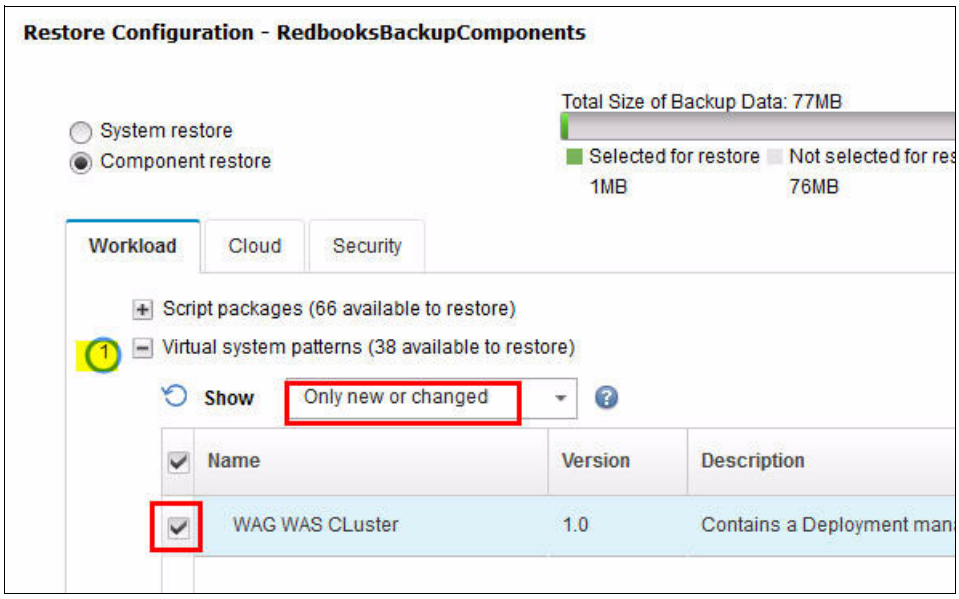


Figure 10-25 List of component resources for restore

Click **Restore** to start the restore. Progress can be monitored in the job queue, as shown in Figure 10-26. If the restore selection includes a storage volume group that has storage volumes, those volumes are restored. This function provides another option of backing up the storage and restoring it at a later point.

Figure 10-26 Monitor the restore progress

10.5.3 Alternative backup locations to restore component data

There might be situations where you must restore data from an alternative backup location that is another PureApplication System. It is also possible that you do not have a component backup location that is defined for the PureApplication System, but defined a location for another rack from where you want to restore the component data. It is valuable to have the ability to restore components from an alternative backup location belonging to another PureApplication System. This capability can be used to copy components from one system to another, especially in disaster recovery (DR) scenarios where the same pattern and assets might be needed in the DR system.

You can use PureApplication System restore to set an alternative backup location by clicking **Set alternate location** in the Restore section of the Backup and Restore console. In addition to the settings to create a backup location, you need the serial number of the alternative rack (which can be obtained from the console Hardware Infrastructure Map view). Figure 10-27 shows the fields for the alternative backup location called *Redbooks-AlternateLocation*.

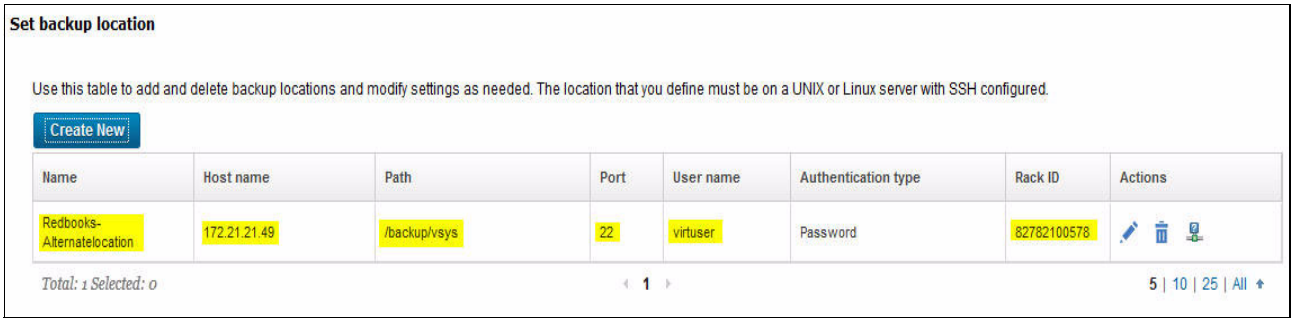


Figure 10-27 Example of an alternative backup location

After the alternative backup location is set, it is visible in the Restore table upon refresh, as shown in Figure 10-28.

Figure 10-28 Restore from an alternative backup location

To restore all or selected components from another rack to this rack, complete the following steps:

1. Select the *Redbooks-AlternateLocation.*
2. Click **Restore data**

10.5.4 Troubleshooting

Backup tasks create jobs that can be viewed in the Event console. Event jobs start with a text string of “CWZIP95”. Using this string as a search filter provides a way to look at the events that are associated with backup jobs.

Failed backup or restore jobs automatically trigger the *Backup and Restore Log Collection Set*. You can view them in the System troubleshooting. You can download the log collection from the System Console, and upload it into the Problem Management Record (PMR).

10.6 Backing up and restoring by using the CLI or REST API

Although the backup and restore function that is provided by PureApplication System should be the primary way of backing and restoring components, there is a rich set of CLI and REST API that can be used to selectively back up and restore resources within PureApplication System.

Chapter 12, “Service and Support Manager” on page 333 provides the basics of developing scripts by using these APIs.

There are some scripts that are available in the samples directory of the CLI tool. You can write custom scripts to perform the backup and restore.

10.7 Backup instances and application data

PureApplication System does not back up deployed instances. There is a mechanism to create one snapshot of the deployed instance from the instance console, which makes a copy (snapshot) of all the VMs in the instance. For example, if a WebSphere Application Server cell had a Deployment Manager, one HTTP Server, and two custom nodes with a total of four VMs, the snapshot makes a copy of all four VMs. Later, the snapshot can be restored.

There is no built-in continuous backup function of the instance or any application data within the VM. This task is the responsibility of the client. Most existing backups or continuous backups of the application data can be reused. If there are agents that provide those functions, those agents can be added into the pattern, so the VMs have the agents that can interact with the backup system that is used within the data center.

Here are some examples of application-specific data:

- Database
- Log files
- Transaction logs
- Configuration files, which might not be needed if the configuration of the middleware is automated

Solutions such as IBM Tivoli Storage Manager or other third-party backup solutions can be adopted. PureApplication System provides a Tivoli Storage Manager plug-in for database backups to an external Tivoli Storage Manager server. After it is configured, the backup of the database can be scheduled.

You can use IBM Endpoint Manager to inject backup agents, such as Tivoli Storage Manager agent fixlets, in VMs. Figure 10-29 shows a high-level diagram of using IBM Endpoint Manager to inject agents. Alternatively, agents can be part of the pattern so that they are added to the deployed VMs.

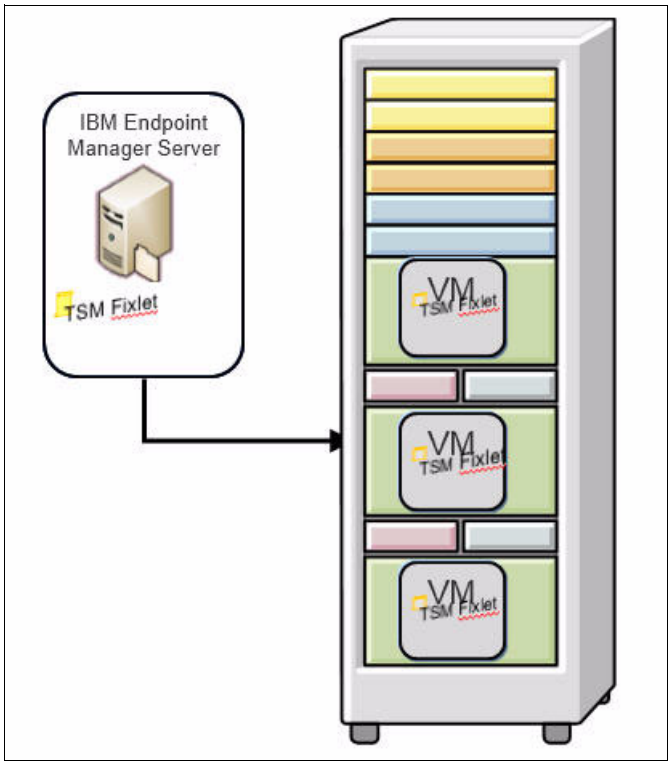


Figure 10-29 Use IBM Endpoint Manager to inject agents

11

Security

The IBM PureApplication platform provides a secure environment for deploying and maintaining workloads. Both IBM PureApplication System and IBM PureApplication Software have security models that provide workload and management isolation. In addition, both provide mechanisms to collate workloads from a management perspective, but isolate workloads from a physical and logical perspective. You can use this approach to meet stringent security policies and use the PureApplication platform with sensitive and regulated data.

This chapter provides a look at the facilities that PureApplication System and PureApplication Software provides that you can use to integrate them into your existing security framework.

This chapter covers the following topics:

- System management
- Resource isolation
- System security
- Workload security
- Security monitoring
- Special situations: Peeling back the covers

© Copyright IBM Corp. 2015. All rights reserved.

11.1 System management

The PureApplication System is managed by the PureApplication System Manager (PSM). The PSM is accessible by administrators through the management network connection. On the PureApplication System, this connection is typically through ports 64 of the Top of Rack (TOR) switches. The management connection is not a typical out of band management connection; it is a highly available aggregated link between the rack and the customer data center.

Each PureApplication System has two PSMs, each of which have a static IP address that is accessible over the management network. The leader PSM is assigned a floating IP address. The floating IP address is what administrators typically use to access the system management interface (also known as the *system console*).

PureApplication Software also has a PSM. The PSM consists of the software stack that is installed on a Red Hat Linux system that is provided by the customer. This PSM is physically isolated from the hypervisors that host deployed workload virtual machines (VMs).

Whether working with PureApplication System or PureApplication Software, the PSM has processes that need access to the VMs for management purposes. Depending on your security needs, this access can be done in a couple of ways.

First, the management node (PSM) talks to VMs through a management Ethernet interface (eth0 or en0) on the VMs. This activity is done on a virtual LAN (VLAN) that is used only within the rack for each cloud group. All communication is done internal to the rack over Internet Protocol version 6 (IPv6) addresses.

If you are on the system and using externally managed cloud groups, you provide IP addresses for cloud management. These IP addresses that are used for cloud management are in separate IP Groups from the data IPs that you use for access to the VMs. Each externally managed cloud group can have multiple external management IP groups. This approach allows VMs within a cloud group to be logically isolated. In this scenario, management traffic from the PSM exits the rack on the management ports and is routed to the VMs through the data ports.

If you are working with PureApplication Software, all VM management traffic goes over the data network. It is possible to have multiple data networks that are defined to achieve the level of logical isolation that you need.

After workload VMs are deployed, they can be managed through the PureApplication System system console. Most middleware patterns provide their own user interface consoles. These consoles can be accessed through the PureApplication System system console or directly. In many enterprises, it is preferable to have a separation of duties between the system and the middleware. Using this approach, the PureApplication workload administrator does not need access to the PureApplication System system console. In fact, they do not need to know where their new environment is hosted.

11.1.1 Management system security

To help protect the integrity and confidentiality of the management system, PureApplication System compute nodes encrypt the data and binary images on the hard disk drive (HDD) and solid-state disk (SSD) by using the Advanced Encryption Standard (AES) encryption algorithm with a 256-bit key size. Encryption keys are not stored on the disk, but are stored instead in the Trusted Platform Module (TPM) on the system management computer circuit board. Configuration data and binary images are not visible to operating system (OS) users. OS users log in to a restricted shell that does not support shell scripts. OS users cannot start or stop OS processes, and they do not have access to file systems.

11.1.2 System console SSL certificates

Hypertext Transfer Protocol Secure (HTTPS) is an internet communication protocol that protects the integrity and confidentiality of data that is transferred between the client and server. The IBM PureApplication platform uses HTTPS to give users access to the system console for performing both administrative and user actions in a secure way.

Instead of using the default Secure Sockets Layer (SSL) certificates that are generated by IBM, as a security administrator, you might prefer to use enterprise SSL certificates of your organization. By doing so, you ensure that the PureApplication platform is in compliance with your corporate security policies.

You can import your system console SSL certificates and keys into the PureApplication platform without any assistance from an IBM service representative. You can also restore the original console SSL certificate and key that was in the system before they were updated.

System console SSL certificates can be imported into the PureApplication platform by using the following mechanisms:

- IBM PureApplication System system console
- IBM PureApplication System command-line interface (pure.cli)
- IBM PureApplication System REST API

This publication shows how to import system console SSL certificates by using the PureApplication System system console. For other configuration methods, such as using CLI or REST API, go to the IBM PureApplication System IBM Knowledge Centers:

- Using the CLI for system management, SSL:

  `http://www.ibm.com/support/knowledgecenter/SSCRSX_2.1.0/doc/clihelp/import_restore_cert_key.dita?lang=en`

- REST API for system management, SSL:

  `http://www.ibm.com/support/knowledgecenter/SSCRSX_2.1.0/doc/psapsys_restapi/rar_ssl_certs.dita?lang=en`

Note: To import SSL certificates into the PureApplication platform, you must be assigned the Security administration role with permission to manage security (Full permission).

To import SSL certificates into the PureApplication platform, complete the following steps:

1. Click **System** → **System Security** from the PureApplication System system console.
2. Expand **Import Console Certificate and Key** (Figure 11-1).

Figure 11-1 Import Console Certificate and Key

3. In the Certificate File field, browse to and select the certificate file to be imported. The certificate file must have a `.crt` file extension.

   The certificate file should be in privacy enhanced mail (PEM) format. PEM is a standard for sending secure email over the Internet. The first few lines of a valid certificate file in PEM format look similar to Example 11-1.

Example 11-1 Beginning of a valid certificate file in privacy enhanced mail format

```
-----BEGIN CERTIFICATE-----
MIIC9zCCAd8CAmmrMA0GCSqGSIb3DQEBBQUAMEExDDAKBgNVBAoTA0lCTTEYMBYG
```

4. In the Private Key File field, browse and select the key file to be imported. The key file must have a `.key` file extension.

   The key file should be in PEM format. The first few lines of a valid key file in PEM format should look similar to Example 11-2.

Example 11-2 Beginning of a valid key file in privacy enhanced mail format

```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEA0RLL5O6Or3PsiigRwekXlibcfw4At6E4vqbdhbAB/ErfV/gi
```

5. (Optional) You can also restore the original console SSL certificate and key that was in the system before they were updated. To restore the original certificate and key files that were included with the system, select **Restore back to original certificate and private key**.
6. Click **Submit**, as shown in Figure 11-2.

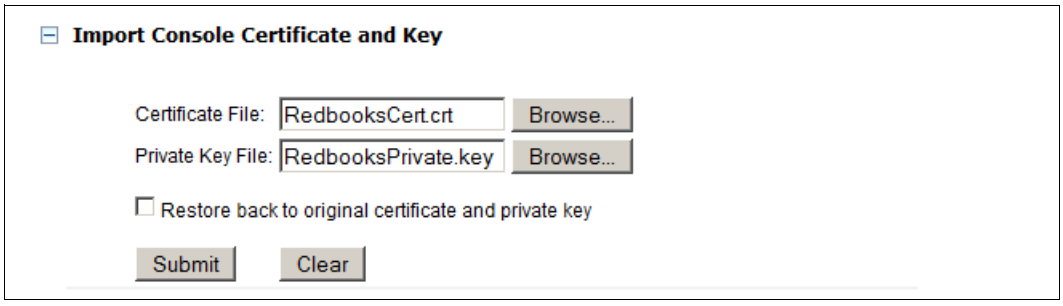


Figure 11-2 SSL certificate and private key files selected

7. After you click **Submit**, the console restarts approximately 10 seconds after the request is completed and the following message displays on the console:

   Your settings were changed successfully

8. Click the web browser refresh icon to confirm that PSM is working correctly with the updated SSL certificates.
9. Update the certificate and key on the other PSM. Perform a failover to the other PSM, and repeat steps 1 on page 302 - 8 by using the same certificate and key files.

Note: Do not forget to put the system in maintenance mode to prevent compute nodes from temporarily moving to a quiesced state before you initiate a failover. You can perform a PSM failover by using one of the following methods:

- Run a PSM failover command on the leader management node.
- Power off the leader management node from the console.
- Run a PSM restart command on the leader management node (or PSM restart remote from the non-leader management node).
- Soft press the leader PSM on the physical rack.

For more information about user-initiated failover operation, go to the IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSCRSX_2.1.0/doc/clihelp/appsysadmincli-psm.dita

Select the PureApplication System type, such as **W2700 2.1.0** → **Troubleshooting and support** → **Using diagnostic tools** → **PureApplication System Administrative Interface** → **psm command**.

10. Download the CLI interface tool again and connect to PSM to ensure that it is updated with the new certificate and private key file.

Note: If you cannot access the console after completing this procedure, contact your IBM service representative.

Updating peer certificates in a multi-system environment

If you are using the PureApplication multi-system setup and updated the SSL certificates in one of the systems in your management domain, you should update the other system's truststore database with the updated certificates. To do so, complete the following steps:

1. Open the CLI from your client and connect to the node on which you updated the SSL certificates, as shown in Example 11-3.

Example 11-3 Open the CLI from your client and connect to the node on which you updated the SSL certificates

```
# cd /directory/path/to/purecli/bin directory
# ./pure -h <PureApplication_Management_IP> -u admin
password:
>>
```

2. Obtain the rack location name (peername), as shown in Example 11-4.

Example 11-4 Rack location name

```
>>> admin.racks[0].location_name
1000202
```

3. Open the CLI from your client, and connect to the other node in the multi-rack setup, as shown in Example 11-5.

Example 11-5 Open the PureApplication CLI and connect to the other node

```
# cd /directory/path/to/purecli/bin directory
# ./pure -h <PureApplication_Management_IP_of_otherNode_in_Multi> -u admin
password:
>>
```

4. Check the peer certificates on the system, as shown in Example 11-6.

Example 11-6 List the peer certificates on the system

```
>>> deployer.peercertificate._list()
[
  {
    "algorithmName": "SHA1withRSA",
    "alias": "ipas:1000202",
    "issuer": "CN=IBMPureSystems, OU=IBM PureSystems, O=IBM",
    "md5": "E19E3E88421D90698D76BEFBB69034B7",
    "notAfter": "Mon Mar 06 02:48:52 UTC 2034",
    "notBefore": "Tue Mar 11 02:48:52 UTC 2014",
    "owner": "CN=IBMPureSystems, OU=IBM PureSystems, O=IBM",
    "sha1": "C3DDE64B69BE11D7030B06AF860779A53E9829C8",
    "sha256":
"08DBD9AEE224CC10427F2B2AD130BE701DFCF8F0F4792D0300E0E53740C3B3EE"

  }
]
>>>
```

5. List the peer certificate by name, as shown in Example 11-7.

Example 11-7 List peer certificates

```
>>> deployer.peercertificate._get('1000202')
{
    "algorithmName": "SHA1withRSA",
    "alias": "ipas:1000202",
    "issuer": "CN=IBMPureSystems, OU=IBM PureSystems, O=IBM",
    "md5": "E19E3E88421D90698D76BEFBB69034B7",
    "notAfter": "Mon Mar 06 02:48:52 UTC 2034",
    "notBefore": "Tue Mar 11 02:48:52 UTC 2014",
    "owner": "CN=IBMPureSystems, OU=IBM PureSystems, O=IBM",
    "sha1": "C3DDE64B69BE11D7030B06AF860779A53E9829C8",
    "sha256":
"08DBD9AEE224CC10427F2B2AD130BE701DFCF8F0F4792D0300E0E53740C3B3EE"
}
```

6. Import the certificate file, as shown in Example 11-8 on page 305.

Example 11-8 Import a peer certificate

```
>>> deployer.peercertificate._import({'certfilepath':'C:\\RedbooksCert.crt',
'peername':'1000202'})
```

11.2 Resource isolation

One key concept of cloud computing is *resource sharing*. System users or applications all share the resources that are available on the cloud. Resource sharing is also a fundamental requirement for having multi-tenant systems. As a cloud environment, the PureApplication platform is no exception. Resources, such as processors, memory, storage, network, and compute nodes, are all shared among cloud resource consumers. Sharing resources might raise questions of data confidentiality, security, and resource contention. In the PureApplication platform, the system provides secure and protected resources against all of those security threats.

Although resource sharing offers major cost benefits to business, the ability to isolate resources to protect data confidentiality, integrity, and privacy is the main concern of enterprise customers. To provide an environment on which enterprises can have a secure IT environment for their infrastructure and applications, PureApplication System and PureApplication Software provide resource isolation and data segregation through networking, cloud groups, and access control.

From the network perspective, PureApplication System isolates management traffic and customer data traffic by using separate external physical networks. An additional level of resource isolation among deployments is supported through the construction of cloud groups, IP groups, and environment profiles. Figure 11-3 shows the logical and physical resource isolation.

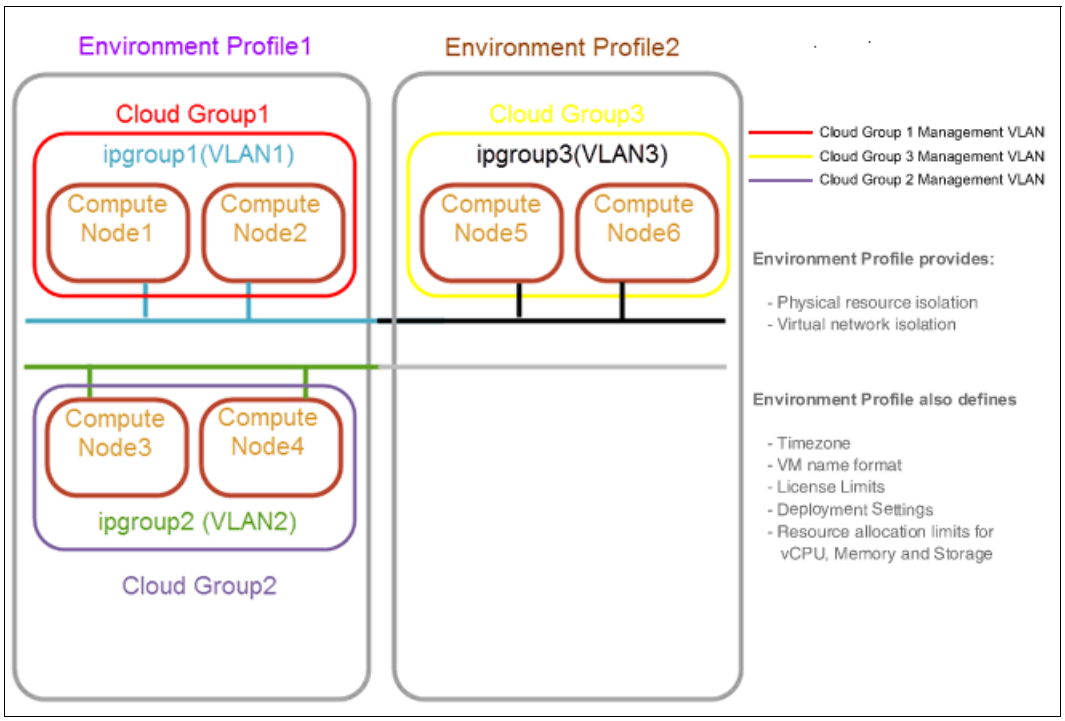


Figure 11-3 Resource Isolation in the PureApplication platform

However, if you are working with PureApplication Software, management traffic goes over the data network. It is possible to have multiple data networks that are defined to achieve the level of logical isolation that you need.

11.2.1 Cloud groups

The PureApplication platform groups computing resources into cloud groups. Cloud groups are a collection of hypervisors, storage, and IP groups. Cloud groups provide physical isolation for workload run times. They allow different application tiers to have dedicated processors.

Different PureApplication users take advantage of cloud groups in unique ways:

- Some clients prefer to isolate workloads based on data classification.
- Other clients prefer to isolate their workloads by line of business or application tier.
- Other clients choose to have large cloud groups with all their applications together.

These customers rely on the logical isolation that PureApplication can provide in concert with its ability to balance dynamically workloads and arbitrate resource contentions with runtime priority settings.

In PureApplication System, cloud groups can be an internally managed cloud group or externally managed cloud group. Figure 11-4 shows those cloud group types.

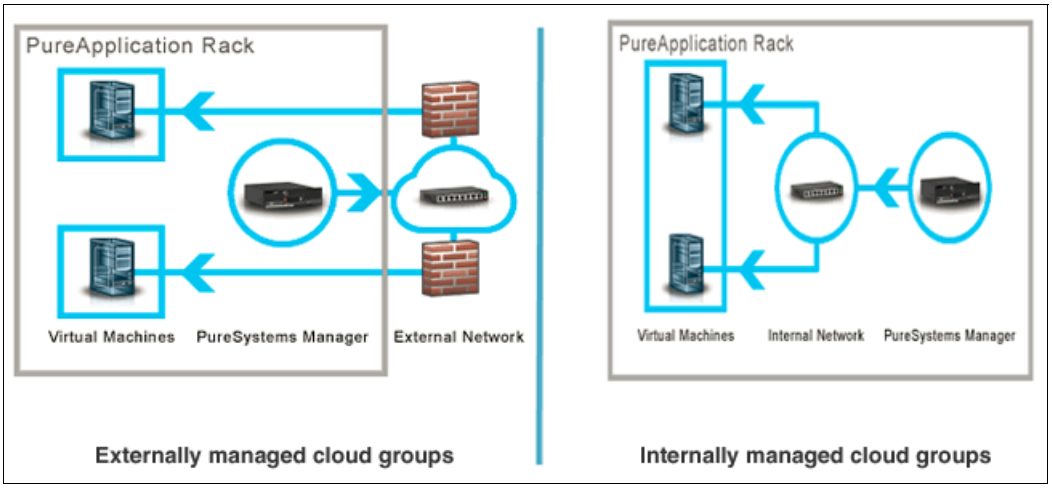


Figure 11-4 High-level data flow of externally and internally managed cloud groups

When you create a cloud group, you specify whether the cloud is to be managed internally or externally, as shown in Figure 11-5 on page 307.

For internally managed cloud groups, you specify a management VLAN for the cloud group. All VMs that are deployed have an IPv6 address that is generated and assigned for management traffic. For externally managed cloud groups, you provide one or more VLANs for VM management.

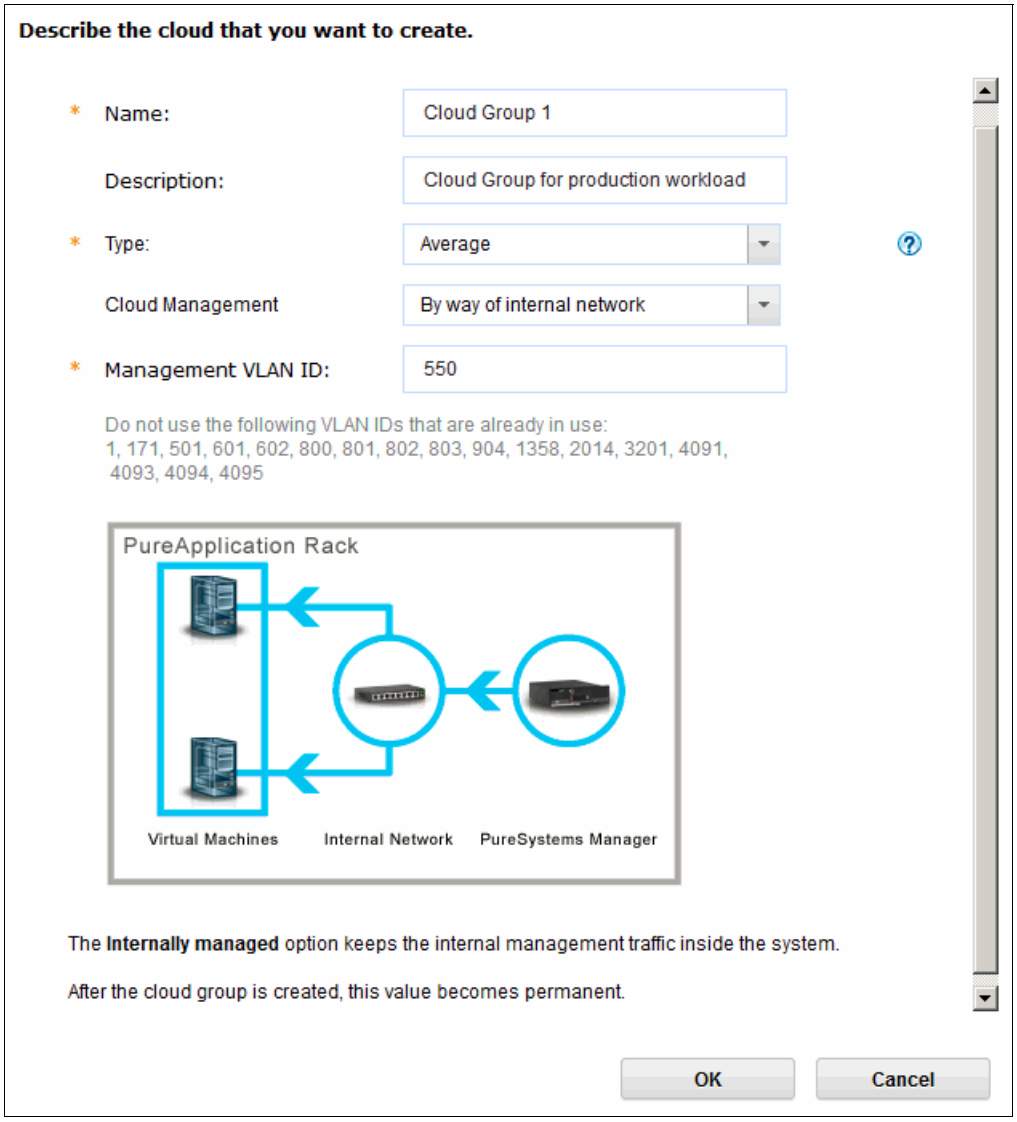


Figure 11-5 Describe a new cloud group

When a cloud group is created, the management VLANs are automatically assigned to the correct ports in the TOR switches and chassis switches. In addition, the management VLAN is also added to the Virtual I/O Server (VIOS) (for PureApplication System W1700/2700) or the Distributed Virtual Switches (DVSs) (for PureApplication System W1500/2500).

IBM Knowledge Center has documented instructions for creating and adding cloud groups. For more information about adding cloud groups, see the IBM Knowledge Center at the following website:

http://www.ibm.com/support/knowledgecenter/SSCRSX_2.1.0/doc/iwd/cgt_cgvmwcre.dita?lang=en

Storage isolation is also set up when you create a cloud group. In the PureApplication System, a host group is created and the host bus adapters (HBAs) of the compute nodes are added automatically. In PureApplication Software, the SAN administrator is responsible for ensuring that the host groups are created correctly and that the storage is discoverable by VMware.

11.2.2 IP groups

Whenever the PureApplication platform deploys VMs, it assigns IP addresses to the VMs so that VMs can be reached over the designated network. The PureApplication platform has those IPs assigned to VMs from IP groups.

IP groups are a way of differentiating sets of IPs within a subnet. Subnets are useful because you can use them to provision firewall rules for a group of IPs ahead of time so that they are ready for use when needed. By having extra IPs in a pool that are configured, you can have a continuous availability strategy that still maintains security standards.

When you create an IP group, you specify settings such as gateway, subnet, and domain name system (DNS) for use in connecting to the enterprise network and VLAN for the network traffic. You then define a pool of IP addresses within the specified subnet, as shown in Figure 11-6.

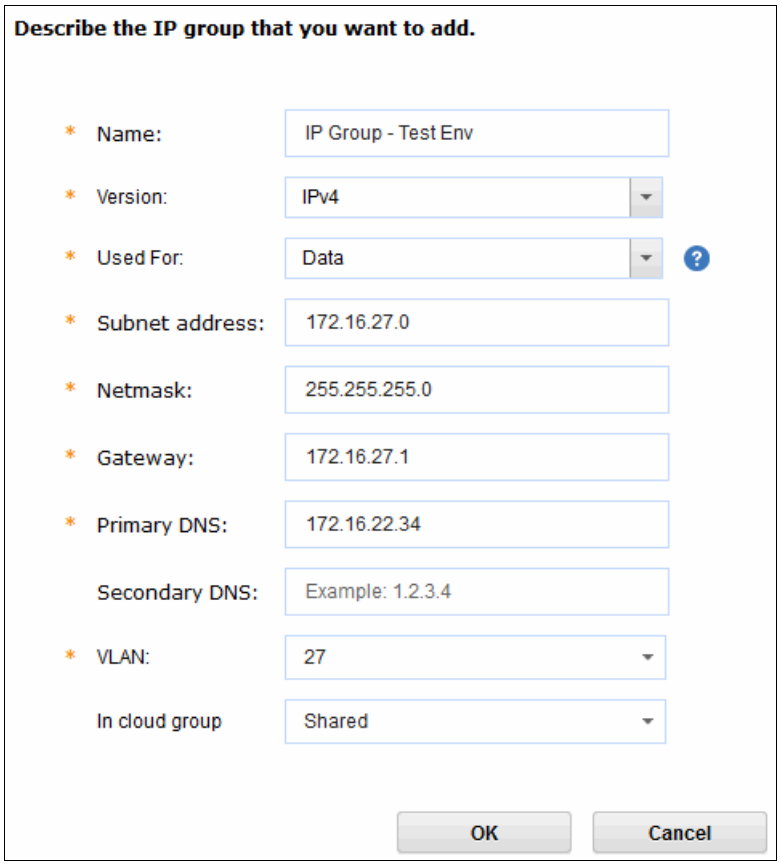


Figure 11-6 Describe an IP group

You can use IP groups to partition a subnet. You can also partition the subnet into smaller groupings. You can create IP groups with duplicate subnet information. You can enter IP addresses by specifying a range in the console or by using the CLI.

A single deployment can use IPs from multiple IP groups. This way, you can isolate your web, application, and data tiers. They can be on different ranges of the same subnet or on separate VLANs.

IBM Knowledge Center has documented instructions for creating and administering IP Groups, which can be found at the following website:

http://www.ibm.com/support/knowledgecenter/SSCRSX_2.0.0/doc/iwd/srt_admipui.dita?lang=en

11.2.3 Environment profiles

The PureApplication platform environment profiles bring various deployment configurations, cloud groups (include physical compute nodes and virtual network settings, such as IP groups and VLANs), and resource allocation limits (virtual processor limits, memory limits, and storage limits) all together and group them as a profile. In doing so, environment profiles provide resource isolation through a grouping of cloud groups (and physical compute nodes) and IP groups. Through a combination of cloud groups and IP groups, an environment profile can have virtual network isolation from other environment profiles, both management networks and customer data networks.

An environment profile can do the following tasks:

- Define the operational environments, for example, production, development, test, or quality assurance.
- Define a VM name format within the operational environment.
- Specify whether the PureApplication platform or the pattern deployer provides the IP address of the deployment (auto IP assignment from a pool or manual IP assignment in deployment).
- Segment the clouds and IP groups within the clouds to specific environments.
- Assign aliases to the cloud resources, such as clouds and IP groups.
- Assign sections within the clouds to specific users or groups.
- Define a specific time zone for use in deployments.

11.2.4 Access controls

Resource isolation through cloud groups, IP groups, and environment profiles provides cloud resource isolation in terms of both networking and compute resources. However, an enterprise might need to have fine-grained access control to set up environment profiles and isolate cloud resources so that different users or groups of users are permitted to use a different set of cloud resources.

PureApplication platform management provides workload resource and cloud resource isolation such that users can deploy only workload patterns to cloud resources to which they are explicitly granted access rights.

User access to cloud resources is managed by using the environment profiles. Users must be explicitly granted read access rights to an environment profile to deploy virtual application patterns and Virtual System Patterns (VSPs) to that environment profile. Workload administrators with full permission roles must have read access rights to cloud resources to accomplish the following tasks:

- Create an environment by using those cloud resources.
- Grant other users access rights to the environment profile.
- Deploy shared services to the environment profile.

11.3 System security

The PureApplication platform includes the following key security features:

- The root key for disk-based encryption is tied to the system, and is private and protected.
- No user-provided logic can be run on the system.
- Local users with full hardware administrative permissions can use a Secure Shell (SSH) connection to manage the PSMs.
- Monitoring of security-related events can be enabled on the system.
- Account policies can be created for local users.
- Your own console SSL certificates and keys can be imported into the system.
- Users and user groups can be synchronized in a multi-system environment.
- You can manage your console session idle timeout.

11.3.1 Disk-based encryption

A key is used to encrypt the contents of the SSD and the HDDs. The key is specific to each system and cannot be modified. All the local file systems, such as the SSD and the HDDs, are encrypted. By encrypting the file systems, the information is tied to the specific system and is unreadable even if it is physically compromised.

Here are the contents of the SSD:

- OS
- System software
- Root passwords of the deployed VMs
- Hypervisor management endpoint passwords
- Secure access keys

To protect this sensitive information, there is no user access to the SSD.

The HDDs are in user serviceable trays that can be removed. However, encryption ensures that any person attempting to access the system to remove these drives cannot access the data.

11.3.2 System-level access

During the life of the PureApplication System, there will be times when IBM representatives must access components within the rack for maintenance. In these cases, a system administrator must grant IBM the right to log in.

The IBM service representative account provides service and support personnel access to the hardware within the rack. This approach allows for advanced troubleshooting and problem resolution. This approach does not allow IBM to see your workloads or your data. This situation provides only the minimum access for servicing the system.

As a system administrator, to grant IBM the right to log in to the system, complete the following steps:

1. Click **System** → **System Troubleshooting** from the PureApplication System system console.
2. Select the **Enable IBM service representative account access** check box at the top of the window.

   A new Secret key text field appears in the window, as shown in Figure 11-7.

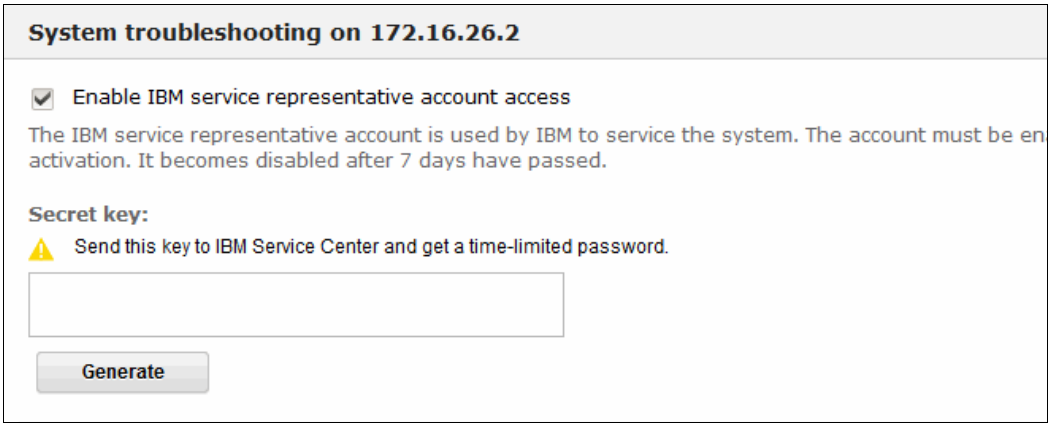


Figure 11-7 Secret Key generation window for IBM service representative account access

3. Click **Generate** to generate the secret key. The secret key is generated by the system, as shown in Figure 11-8.

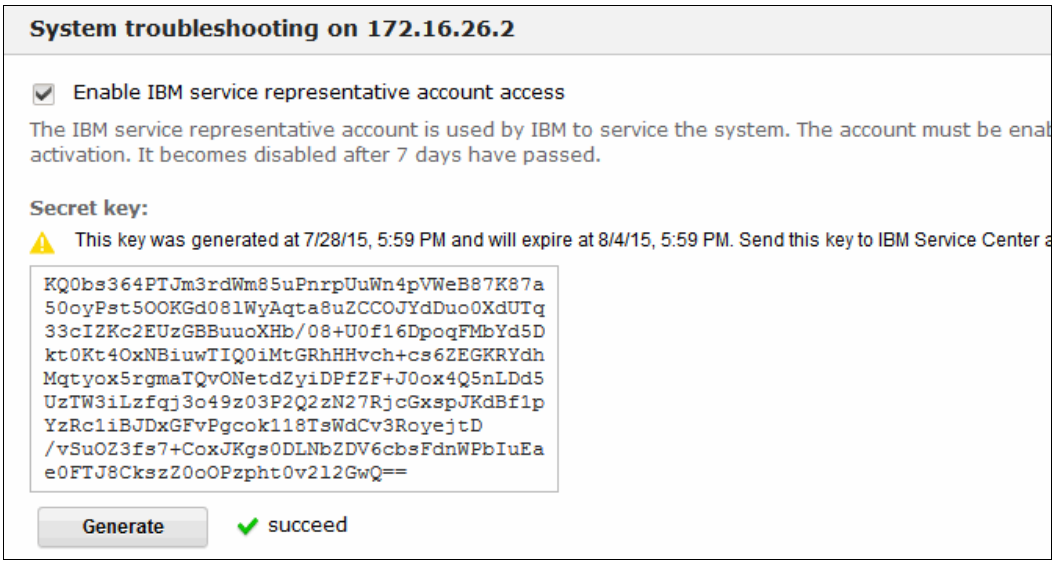


Figure 11-8 Generated secret key

4. Copy the secret key and send it to the IBM service representative.

5. After the IBM service representative finishes the service tasks, disable the IBM service representative account access by completing the following steps:
   a. Click **System** → **System Troubleshooting** from the PureApplication System system console.
   b. Clear the **Enable IBM service representative account access** check box at the top of the window.

Note: The IBM service representative account must be enabled before the IBM service representative can log in to take corrective actions. The account remains enabled for 7 days from the time of activation. It becomes automatically disabled after 7 days pass unless the system administrator manually disables it.

11.3.3 Roles and permissions

The PureApplication platform provides security role-based access control (RBAC) at both the system level and resource level. This combination of access control provides a balance between security roles at the overall system level and more granular access control at the resource level, for example, individual cloud groups, IP groups, virtual appliances, and compute nodes. The role-based authorization design is based on both the separation of duty security principle and the least privileged security principle.

System-level roles

Access control permissions for the PureApplication platform are divided into major functional areas and assigned according to the administrator roles as follows:

- Workload resources administration
- Cloud group administration
- Hardware administration
- Auditing
- Block storage replication administration
- Security administration

If a user does not have permission to complete a task, the menu items are not presented to them.

Users can be granted management responsibility of one or more of these functional areas. Within each area, an administrator can be granted either a read-only security role or a full permission security role. An administrator with a full permission security role can perform all administrative actions of that area, and an administrator with a read-only security role can monitor the configuration and status, but cannot make changes.

No single administrator needs to have all areas of administrative authorities. The preferred practice is to divide the management responsibilities among administrators so that no single administrator has all the administrator roles. For example, a user who has the auditing role is responsible for managing security auditing configuration, archiving security auditing records, monitoring, and analyzing any abnormality in system operations. An auditing administrator ideally should not have other areas of administrative responsibility, both to avoid any conflicts of interest and to be able to hold effectively users accountable for their actions.

An administrator with any one of the full permission security roles can delegate his own security roles to other users if that the administrator also has a delegation security role, as shown in Figure 11-9 on page 313.

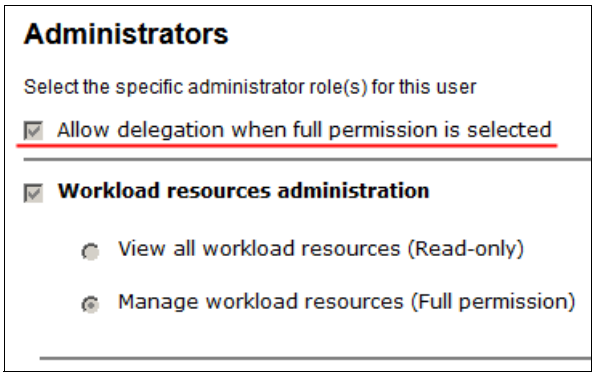


Figure 11-9 Delegation

Resource-level roles

Resource-level roles refer to the security roles and access control permissions that are required to administer individual resources in the system, for example, cloud groups, IP groups, compute nodes, and storage devices. At the resource level, administration responsibility for the workload resources administration, cloud group administration, hardware administration, and security administration functional areas is further divided into subroles to support the least privileged security principle. For example, users and groups who are responsible for creating and administering deployment patterns can be granted the Create new patterns security role and not the full-permission Workload resources administration writer role.

This model of providing more granular access control at the resource level is designed to limit access to only those resources that must be managed at a particular time. The level of access (read, write, all, or none) that is assigned to specific resources can be selected individually. For example, a user may be granted read access to some resources, and write access to other resources.

Roles and permissions are documented in the IBM Knowledge Center. For more information about each security role and its restrictions, see the following website:

http://www.ibm.com/support/knowledgecenter/SSCRSX_2.1.0/doc/iwd/aac_user_permissions.dita?lang=en

11.3.4 User account policies

User account policies are used to protect user accounts from being used by attackers to intrude into enterprise systems. User account policies include a failed login attempt policy, an account lock-out policy, a password policy and password expiration policy, and so on.

Having a well-prepared user account policy provides security protection for the PureApplication platform from the security threats, and provides protection for your users' passwords by forcing a password policy.

The PureApplication platform has the user account policies feature only for local users. If your PureApplication platform is configured to authenticate users on a Lightweight Directory Access Protocol (LDAP) server, some policies do not apply even if they are enabled. For example, users who violate the "Number of allowed consecutive failed attempts to authenticate" policy are not locked out if they are LDAP users.

Note: User account policy configuration steps require an account with the security administration role with permission to manage security (full permission).

To configure user account policies, complete the following steps:

1. Click **System** → **System Security** from the PureApplication System system console.
2. In the User Account Policies section, select or clear the check boxes in front of the individual policies to enable or disable them, as shown in Figure 11-10.

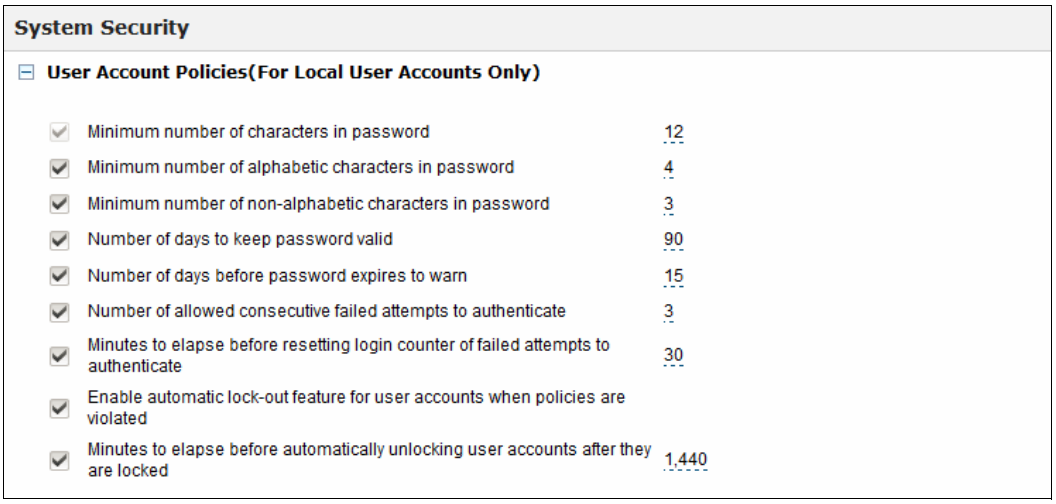


Figure 11-10 User Account Policies on the PureApplication platform

3. For enabled policies, click the values beside the individual policies to change their values. The minimum number of characters in the password policy is always enabled and requires a value of at least 1.
4. Click **Save** beside each value that you update.

IBM Knowledge Center has documented descriptions of each value of the policy items. For descriptions of each value, go to the following website:

`http://www.ibm.com/support/knowledgecenter/SSCRSX_2.1.0/doc/iwd/aac_user_policies.dita?lang=en`

11.3.5 User and User Group management

User and User Group management is a vital feature of all multi-user systems and one of the core areas where IT security comes into play. Moreover, almost all of the industry regulations have something to say about it and mandate some strict IT policies regarding user management, access rights, and permissions.

The PureApplication platform provides detailed User and Group management features that can help you protect your environment from internal and external security threats and achieve compliance with IT policies.

Preferred security practices

As a preferred security practice, always try to minimize the assignment of multiple management responsibilities to users or user groups and give them only what they need. Putting users and groups on a need-to-have basis provides a more secure environment and a more manageable environment.

When you create users in the PureApplication platform, they automatically receive the default permission to deploy objects, such as VSPs, into the cloud. You must manually assign additional permissions to users.

When making these additional permission assignments, consider using the separation of duties (SoD) strategy to protect the integrity of the auditor role. Isolate the assignment of auditing permissions to one or more users who do not have other powerful administrative capabilities, such as the system or cloud administration permissions.

Auditors are responsible for monitoring activities and detecting any abnormal activities in the system, and system administrators are responsible for administering resources in the system. These different responsibilities must be assigned to different individuals.

PureApplication System implements two other SoD-oriented policies to help you control user activity in the cloud. These policies limit the authority to assign user permissions.

- Only users with the following roles can make permission assignments:
  - Workload resources administration with Manage workload resources (full permission)
  - Cloud group administration with Manage all cloud groups (full permission)
  - Hardware administration with Manage hardware resources (full permission)
  - Security administration with Manage security (full permission)
  - Auditing with Manage auditing (full permission)
- Users must have at least one of the five full permission administrator roles and the delegation security role to delegate their security roles to other users.

Thus, you can create a sound SoD implementation in which no single user can perform any action that is not recorded. All of these measures protect the integrity of your environment.

Managing system users

IBM Knowledge Center has documented instructions regarding managing system users. You can find information about managing system users at the following website:

http://www.ibm.com/support/knowledgecenter/SSCRSX_2.1.0/doc/iwd/aat_manusui.dita?lang=en

Managing user groups

IBM Knowledge Center has documented instructions regarding managing user groups. You can find information about managing user groups at the following website:

http://www.ibm.com/support/knowledgecenter/SSCRSX_2.1.0/doc/iwd/aat_manugui.dita?lang=en

11.3.6 Directory server integration (LDAP)

You can use the PureApplication platform to integrate with an existing LDAP server for user and group management. You can use this approach to use standard user authentication credentials. You can centrally manage group membership.

The PureApplication platform supports Generic LDAP, IBM Tivoli Directory Server, and Microsoft Active Directory, which implement the LDAP Version 3 specification RFC 4511. When using a secure LDAP server that is load balanced, be careful to accept the base security certificate. For more information about secure LDAP connections and certificates, see 11.3.7, “Secure LDAP communication” on page 317.

The PureApplication platform reads user and group entries from LDAP servers. PureApplication System can search for users, groups, and users as static members of groups. The system does not support environments in which LDAP groups are organized as attributes of users, either static or dynamic.

Note: If your PureApplication platform is configured to authenticate users on an LDAP server, some user account policies do not apply even though they are enabled.

IBM Knowledge Center has documented instructions regarding integrating the PureApplication platform with various LDAP servers. You can find the integration instructions for IBM PureApplication System W2700 v2.1.0.0 at the following website:

- Integrating with IBM Directory Server

  http://www.ibm.com/support/knowledgecenter/SSCRSX_2.1.0/doc/iwd/ldap_integrate_ids.dita?lang=en

- Integrating with Microsoft Active Directory

  http://www.ibm.com/support/knowledgecenter/SSCRSX_2.1.0/doc/iwd/ldap_integrate_mad.html?lang=en-us

- Integrating with Oracle LDAP Server

  http://www.ibm.com/support/knowledgecenter/SSCRSX_2.1.0/doc/iwd/ldap_integrate_ols.html?lang=en-us

- Integrating with LDAP Server Cluster

  http://www.ibm.com/support/knowledgecenter/SSCRSX_2.1.0/doc/iwd/ldap_integrate_cluster.html?lang=en-us

For other models and different versions of the PureApplication platform, you can find the instructions at the IBM Knowledge Center by selecting the PureApplication System type, such as **W2500 2.1.0** → **Administering the system** → **Administering users and security** → **Integrating PureApplication System with LDAP servers**.

A sample configuration for IBM Directory Server is shown in Figure 11-11 on page 317.

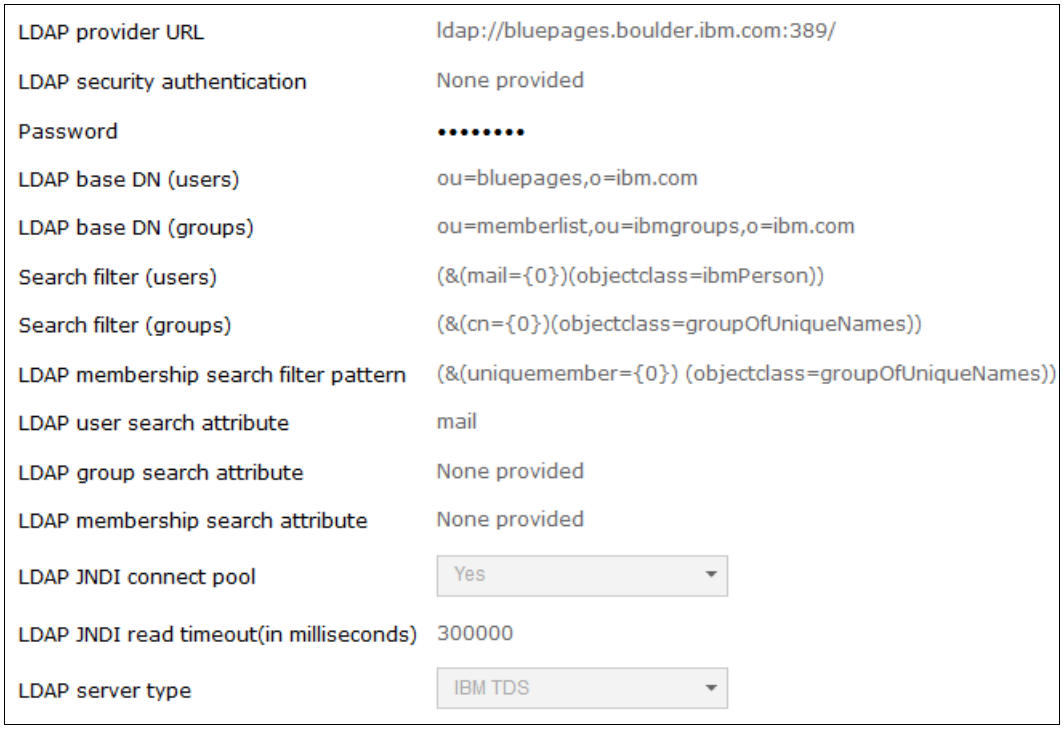


Figure 11-11 Sample configuration for IBM Directory Server

11.3.7 Secure LDAP communication

When you integrate PureApplication System or PureApplication Software with an LDAP server, The PureApplication platform reads user and group entries from the LDAP server and can search for users, groups, and users as static members of groups. Data traffic between the PureApplication platform and LDAP server carries sensitive information. Because of this sensitive data, your corporate security policy might require encrypted connections from the PureApplication platform to the LDAP server. Otherwise, data is carried in plain text, which is an insecure way to carry sensitive data.

For a secure LDAP connection, LDAP Server provides confirmation to the PureApplication platform that it connected to the correct server for LDAP queries and that there is no man-in-the-middle attack in progress. To have such a secure LDAP connection, the PureApplication platform should have trustworthy copies of the appropriate X.509 signer (CA) certificates of the LDAP servers so that it can perform correct validation checks during the initial communication.

As a preferred security practice, you should always use a secure LDAP connection (LDAPS) wherever it is available to protect sensitive data.

To configure a secure LDAP connection between the PureApplication platform and LDAP Server, complete the following steps:

1. Click **System** → **System Security** from PureApplication System system console.
2. Expand **LDAP Settings**.

3. Enter the LDAPS protocol, LDAP server host name, and port number on the LDAP provider URL field, as shown in Figure 11-12.

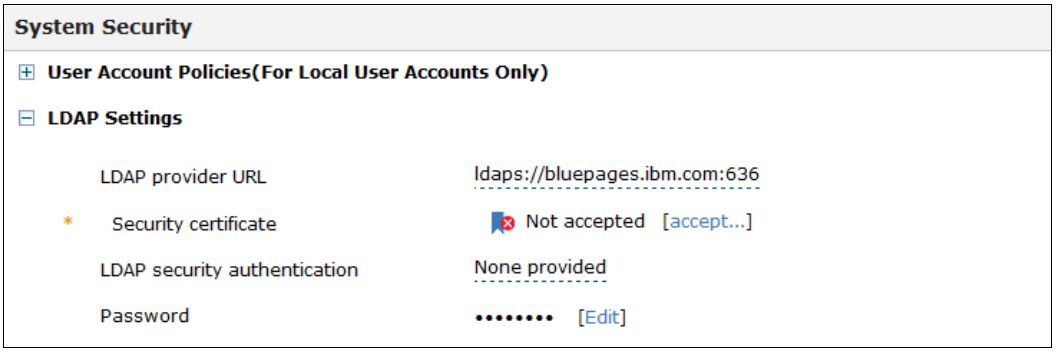


Figure 11-12 Provide an LDAP provider URL

4. After you provide a valid secure LDAP provider URL, a new field named “Security certificate” is displayed under the LDAP provider URL row. Click **accept** in the Security certificate field. A certificates window opens, as shown in Figure 11-13.

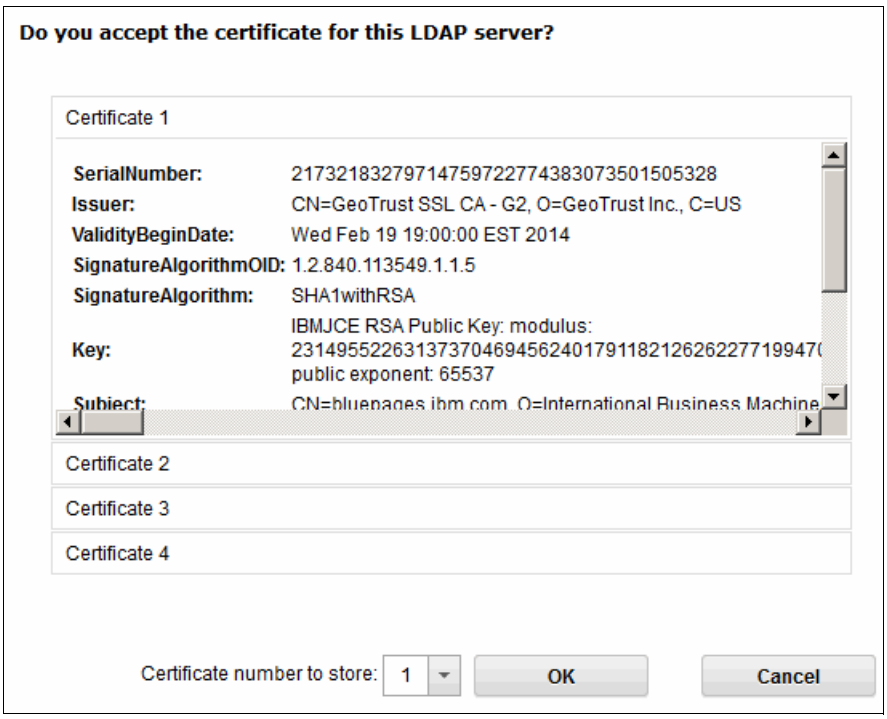


Figure 11-13 LDAP server X.509 certificates

5. Click **OK** to accept the certificate.

Note: Use the “Certificate number to store” drop-down menu to allow the system to trust a clustered LDAP environment on which each LDAP server has a unique X.509 certificate that is issued by a common certificate authority (CA). By configuring the PureApplication platform to trust the common CA, by default the system trusts all certificates that are issued by the trusted CA.

6. The security certificate field shows the status (Figure 11-14 on page 319).

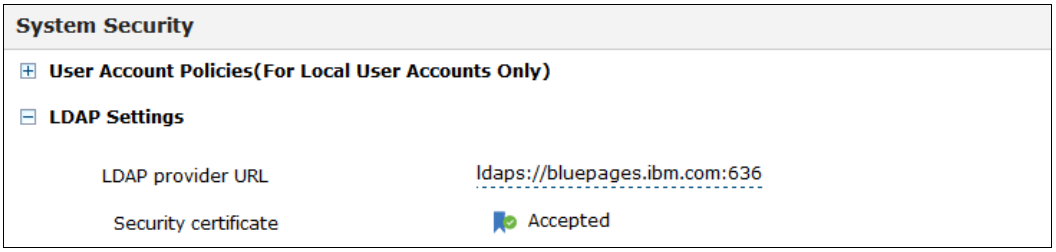


Figure 11-14 Security certificate accepted

For the remaining configuration settings, see 11.3.6, “Directory server integration (LDAP)” on page 315.

11.4 Workload security

Workload is a broad term. It can include OS, runtime environment, middleware products, applications that run on these middleware, and agents for various tasks, such as backup agents, monitoring agents, and security agents. All of those different software components require different security standards and practices. Moreover, there is no one single security standard or security policy with which workload should be in compliance. Every enterprise has different requirements and IT policies regarding security.

The PureApplication platform has powerful and flexible features and tools hat you can use to implement your security policies and customize OS and middleware without any problem.

Workloads that are deployed by the PureApplication platform are based on a standard OS image. The OS image can be customized by the client to meet their needs. For Virtual System, Virtual Application, and Shared Service patterns, a common OS image is used. On PureApplication System, a default IBM OS Image is provided. For PureApplication Software, this is optional. On either platform, it is optional to use your own image with deployments.

The OS image that comes with the PureApplication platform by default can be customized to meet your IT standards and workload needs. Also, you can create a pattern component and add it to your pattern for customized workloads. In the PureApplication platform, you can achieve customization by using one of the following methods:

- Use script packages or add-ons.
- Use a plug-in development kit (PDK) to create software components.
- Use the Extend and Capture feature of the PureApplication platform to create customized OS images.
- Use IBM Image Construction and Composition Tool (ICCT) to create customized OS images.

OS customization can include configuration changes, service activation and deactivation, and software installation, such as installing a security software agent. Instead of providing instructions for each of those tasks, this section provides general guidance, a high-level solution, and the tools that are used to implement the solution in this publication.

Depending on the requirement of the customization, you can use script packages, add-ons, software components, the Extend and Capture feature, or ICCT to implement your corporate security solution or configure the system so that it can be compliant with your IT policies and meet your needs.

IBM Knowledge Center has documented instructions for using script packages, add-ons, the Extend and Capture feature, and the ICCT tool. For more information, see the following topics in the IBM Knowledge Center:

- Managing Script Packages

  http://www.ibm.com/support/knowledgecenter/SSCRSX_2.1.0/doc/iwd/pct_scipacui.dita?lang=en

- Managing add-ons

  http://www.ibm.com/support/knowledgecenter/SSCRSX_2.1.0/doc/iwd/aot_addonsui.dita?lang=en

- Plug-in Development and Creating Software Components

  http://www.ibm.com/support/knowledgecenter/SSCRSX_2.1.0/doc/iwd/pgr_pdk.dita?lang=en

- Extending and Capturing virtual images

  http://www.ibm.com/support/knowledgecenter/SSCRSX_2.1.0/doc/iwd/pct_extend_vi.dita?lang=en

- Working with IBM Image Construction and Composition Tool

  http://www.ibm.com/support/knowledgecenter/SSCRSX_2.1.0/doc/ICON/topics/iwd_cicn_overview.dita?lang=en

11.4.1 LDAP user authentication for OS

Authenticating a deployed workload OS with an enterprise LDAP directory server is a common integration scenario for many enterprise IT organizations. Most of the modern OSs support LDAP authentication for user and group management of the system. However, there is no one single authentication scenario because there are different LDAP servers and different configuration steps for different OSs on the market.

The PureApplication platform OS images use local user and group management by default. The workload OS that is deployed with each VM on the PureApplication platform does not have LDAP integration by default. However, anyone who needs LDAP integration for workload OS can easily add LDAP integration to the default OS by using customization tools and methods that are provided by the PureApplication platform.

The easiest way to configure LDAP integration for workload OS is to use script packages. You can create script packages to install software and then modify OS configuration files.

The IBM developerWorks article *Integration of OS authentication with Microsoft Active Directory on IBM PureApplication System* can help you with integration with a Microsoft Active Directory and provides a solid base even if your corporate LDAP server is different than Microsoft Active Directory. To access this article, go to the following website:

http://www.ibm.com/developerworks/websphere/techjournal/1410_vanrun/1410_vanrun.html

11.4.2 Patch management

Every enterprise should closely monitor the systems and applications against possible vulnerabilities and problems. Otherwise, those vulnerabilities can be used by attackers to access data or damage the system. The enterprise can suffer as the result of those attacks unless they have a good patch management policy and preferred practices.

Software patching is the process of fixing software problems and replacing the vulnerable parts of the software with updated ones. Generally, patches are discovered after the software components are delivered to the customer or made available for general use.

Because PureApplication System includes hardware components, network devices, compute nodes, OS, and middleware software, there are different patches that are applicable to the PureApplication environment. The patches can be grouped into four main categories:

- OS patches
- Middleware software patches
- Firmware patches
- System management software patches

IBM delivers firmware patches and system management software patches as a system update fix pack file for the PureApplication platform. This system update fix pack includes updates for hardware firmware (such as network switch, chassis, storage, compute node, and SAN switch) and management software.

This publication covers patch management in the following chapters:

- See Chapter 7, “Operating system maintenance” on page 155 for PureApplication platform OS patches
- See Chapter 8, “IBM middleware maintenance” on page 205 for PureApplication platform middleware maintenance.
- See Chapter 9, “System and integrated components maintenance” on page 241 for PureApplication platform system update fix packs.

As a preferred security and system management practice, always keep your systems and applications up-to-date by applying updates from IBM.

11.5 Security monitoring

Security monitoring is the one of the most important tasks to be performed in an environment such as the PureApplication platform, which brings infrastructure and middleware software together as an expert-integrated system. Because of that importance, collecting, analyzing, and taking necessary actions for security events and audit data is vital. Security monitoring provides collection and analysis of the data and helps you detect indications and warnings of any possible misuse and intrusion attempts so that you can take preventive actions and respond quickly.

11.5.1 Security and administrative event auditing

Because of enterprise compliance to regulatory policies, an auditing function is a must have feature. PureApplication System and PureApplication Software have an auditing function that is used for storing and recording activities about user actions, administrative actions, and security-related events that occur on the system. PureApplication System and PureApplication Software provide all the required information to describe who performed what action on which resource when, from where (source address), and whether the attempt was successful or not, and if not, what caused the failure.

The auditing function provides accountability, which is a mandatory regulatory requirement for enterprises. By using the auditing data that PureApplication System and PureApplication Software provide, users and administrators are held accountable for the actions they perform on the system. These users and administrators actions include deployment-related actions and all administrative actions that affect data confidentiality, integrity, and system availability.

From the security perspective, audit functions provide all the required information as audit data that is packaged for you to analyze and determine whether any incident that compromised your system occurred. This technique provides security protection of your infrastructure from security threats and all the information regarding auditing. This information can be used to improve your security system and enable you to take all necessary security measures immediately against threats both from the internal and external network.

Security auditing is a fundamental part of regulatory policies for the enterprises to protect data confidentiality, integrity, and system availability. For this reason, the audit function in PureApplication System and PureApplication Software is enabled by default, and nobody including the system administrators, can disable it.

PureApplication System and PureApplication Software store audit records in an internal database. After the number of records exceeds a threshold in the internal database, one of the followings actions can occur:

- If there is external storage configured, the system exports those records to external storage as audit packages, including two files: a CSV file with records, and its checksum file.
- If there is no configured external storage, the system exports those records as an audit package that includes a CSV file with records and its checksum file, and places it on internal storage. PureApplication System and PureApplication Software regularly clean up those auditing packages to stay within file system space constraints. This cleanup can lead to a loss of security event audit records if an external storage server is not configured on the system.

Note: By configuring an external storage server for archiving audit record packages externally, you can analyze event logs offline and automate the event log archival to meet various compliance and regulation requirements process. Communication with external storage servers is protected by using SSH.

Configuring an external storage server

Configure an external storage server so that PureApplication System or PureApplication Software can automatically push audit record packages to an external storage area when the internal database has reached its threshold.

Note: You must be assigned the System level, Auditing role with permission to Manage auditing (Full permission) to configure an external storage server for audit packages.

An external storage server for audit packages is configured from the External Storage Server tab, which is accessed by clicking **System** → **Auditing** from PureApplication System system console.

The PureApplication platform IBM Knowledge Center has documented instructions for the configuration of an external storage server. That information can be found at the following website:

`http://www.ibm.com/support/knowledgecenter/SSCRSX_2.1.0/doc/systemconsole/t_audconfig.dita?lang=en`

Here are some important points to consider when configuring an external storage server for audit packages:

- Always use Rivest-Shamir-Adleman (RSA) key encryption instead of user ID and password authentication to better secure your external storage server.
- Make sure that the external storage server has enough space to store archived security audit packages according to your audit policy. The size of an audit package depends on the number of records it stores, and the number of audit packages that the system generates depends on the activities that are performed on the system. An average size for an audit package is 20,000 records (about 1 MB). You can plan storage space by taking that size into account.
- The External Storage Server (The Secure Shell server) RSA public key size cannot exceed 4096 bits.
- The Public key (external storage server) field might intuitively mislead you to provide the public key of the account (the user account on the external storage system) that you use for the SSH connection. Pay attention to this field and provide a public key of the external storage server (SSH Host public key), not the public key of the account on the external storage server. The `/etc/ssh` directory is usually the location of the SSH host public key (`ssh_host_rsa_key.pub`), but the location can vary depending on the system. The reason for using the SSH host public key instead of using the public key of the account is to provide more secure communication.

Note: The SSH Host key provides a fingerprint that the SSH client (PSM in our case) can use to validate that it is connecting to the server that it thinks it is. The SSH client software checks the server's (host) public key against data that the server sends encrypted with the host private key. This technique requires the host SSH client to have a copy of SSH host public key.

A sample external storage server configuration is shown in Figure 11-15.

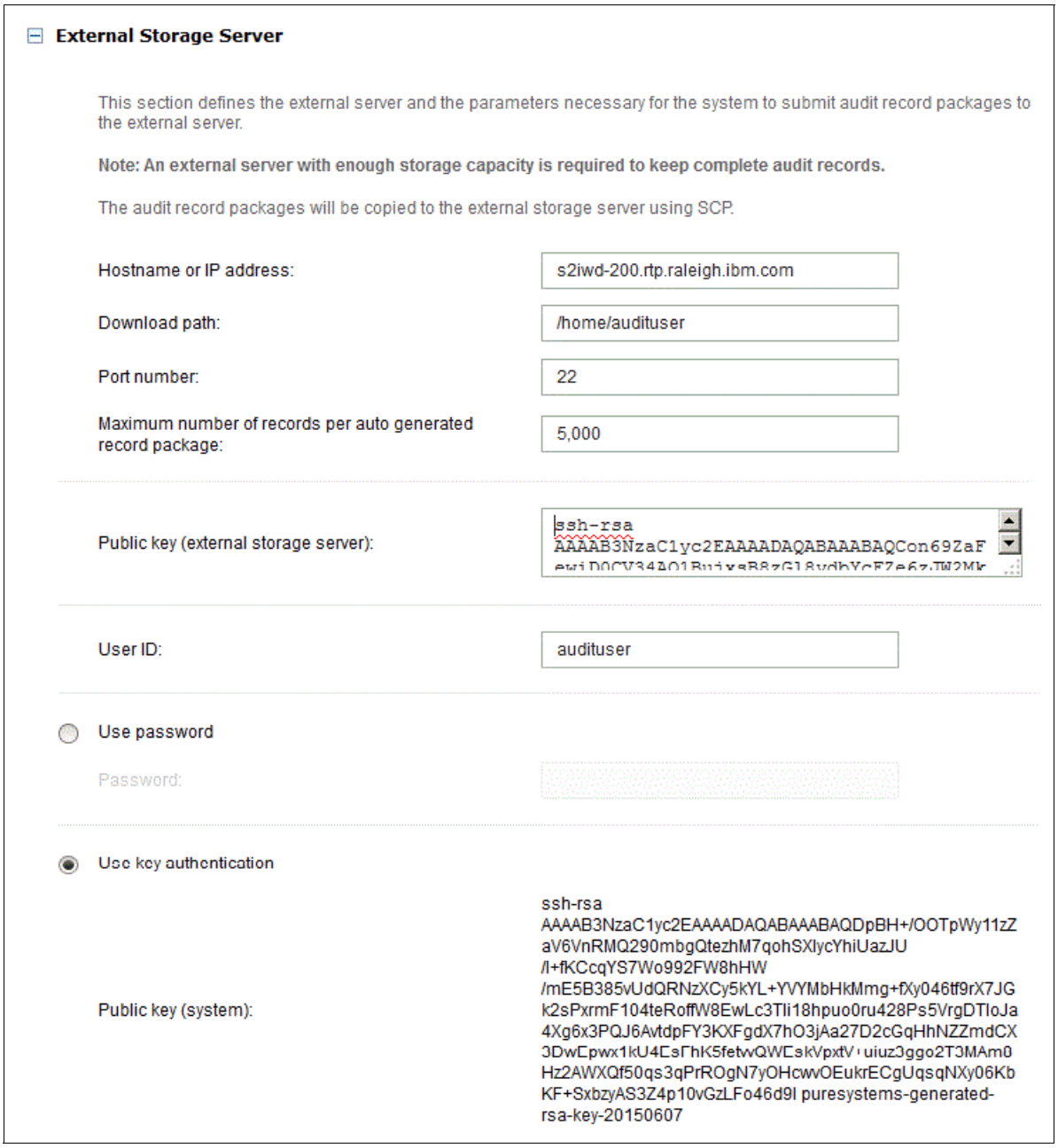


Figure 11-15 A sample external storage server configuration for audit packages

11.5.2 Enabling security events

The auditing system stores and records all activities about user actions, administrative actions, and security-related events that occur on the system. However, those activities cannot be monitored on the event console in real time by default. If you want to monitor security-related events on the system console, PureApplication System and PureApplication Software can be configured to send security-related events to the console in real time. By sending events in real time, you can learn about potential security breaches when they occur on the system.

Security events in the following categories can be sent to system console:

- System restore
- Subnets (IP groups)
- Roles
- User account policies
- User token
- Users
- Groups
- Group memberships
- IP address configurations
- LDAP configurations
- Audit configurations
- Compute nodes
- Email configurations
- SNMP configurations

Moreover, the system administrator or a user with the audit management role can create security events in the system for a particular user's logon and logoff actions.

Activating security event monitoring

To configure PureApplication System or PureApplication Software to send security-related events to the console in real time, complete the following steps.

Note: You must be assigned the System level, Auditing role with permission to Manage auditing (Full permission) to perform these steps. Administrators who are assigned the Auditing role with permission to View all auditing reports (Read-only) can only view security monitoring configurations.

1. Click **System** → **Auditing** from the PureApplication System system console.
2. Expand **Security Monitoring**.
3. To enable a security event in the table, complete the following steps:
   a. In the **Category** column, locate the security event that you want to monitor.
   b. Select the check box in the **Enable** column.
   c. To refresh the table, click the **Refresh** icon above the table.
   d. (Optional) To reset the table to its default content, click the **Reset** icon above the table.

After the configuration is complete, the security monitoring table looks similar to Figure 11-16.

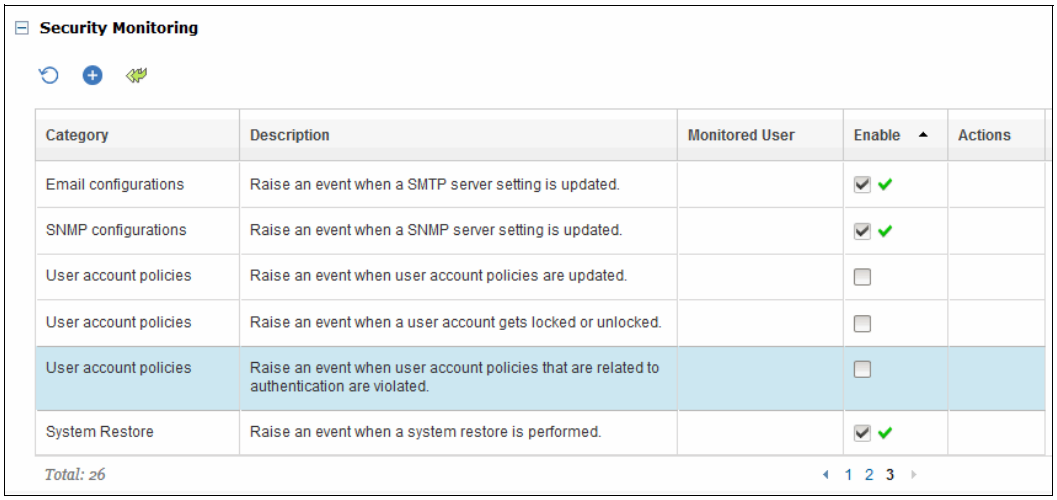


Figure 11-16 Enable security monitoring

Activating security events for a specific user

You can create security events based on a specific user's logging in or logging off the system actions. By doing so, you can monitor a specific user on the system.

To create a security event for a specific user, complete the following steps:

1. Click **System** → **Auditing** from the PureApplication System system console.
2. Expand **Security Monitoring**.
3. Click the **plus sign icon** (above the table).
4. In the new window, select a user from the list.
5. Add a description of the event.
6. Select the user action that you want to monitor, for example, the action of logging in or logging off the system (Figure 11-17).

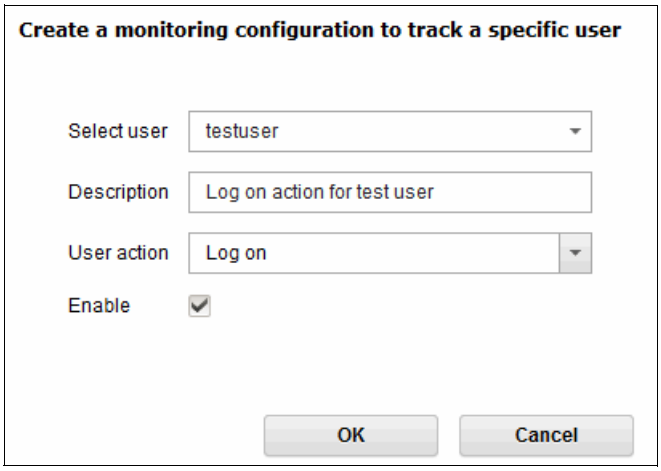


Figure 11-17 Create a security event for a specific user

7. Select **Enable** to enable the security monitoring event.

8. Click **OK**. Details about the new event are added to the table, as shown in Figure 11-18.

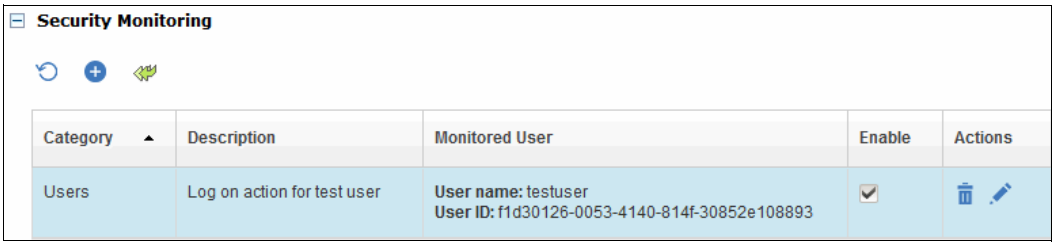


Figure 11-18 Security event added to the table

11.5.3 Forwarding security events

PureApplication System and PureApplication Software have an event forwarding feature. You can use this feature to forward Simple Network Management Protocol (SNMP) traps to an enterprise monitoring system in your environment. When creating an SNMP trap destination, you can filter the events so that only security events are forwarded to the trap destination you create.

To configure security events forwarding, complete the following steps:

1. Click **System** → **System Settings** from the PureApplication System system console.
2. Expand **Events**.
3. Click **Create trap destinations**.
4. Provide the trap destination information, such as IP address, port number, community string, SNMP version, and filter (Figure 11-19).

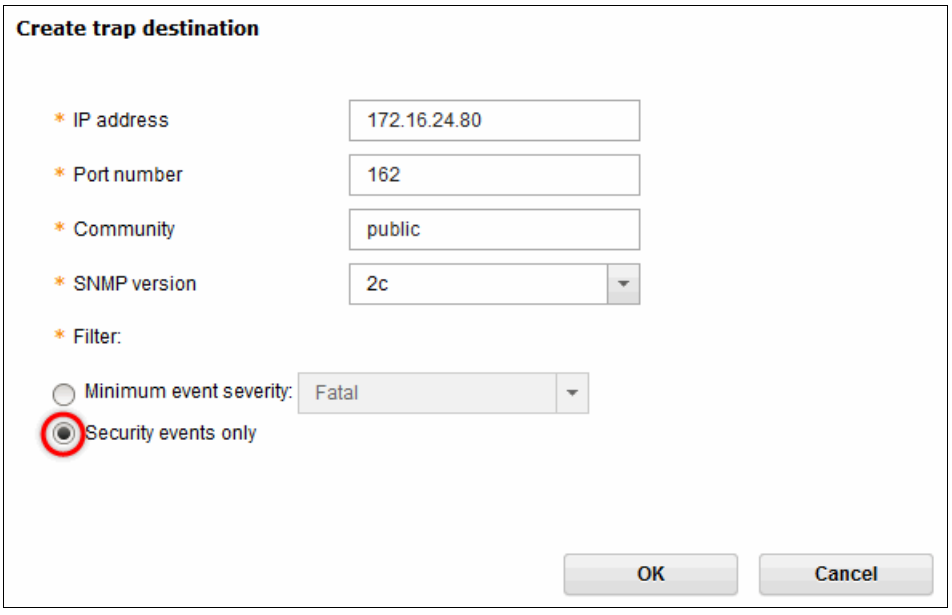


Figure 11-19 Create trap destination with filtering security events only

11.6 Special situations: Peeling back the covers

Most enterprises have groups to manage switches for the data and storage networks. PureApplication System is a pre-integrated appliance with network and storage switches. However, PureApplication System does not bring any extra impact or workload in terms of network management and configuration because PureApplication System is a pre-integrated and pre-configured appliance.

The network or SAN team must understand the configuration of the switches or check port status. PureApplication System provides read-only access to network and SAN switches so that network or SAN team can get what they need.

11.6.1 Network topology

If you click **Hardware** → **Network Topology** in the PureApplication System system console, you can see a table that lists the virtual network links within the rack (Figure 11-20). You can use this view to see each VLAN that the links carry. You can see how the connections are mapped to IP groups, compute nodes, and to compute nodes.

Link Connections

Click a row on the Link Connections table to see the associated compute nodes information in the Compute Nodes tabl

Link: Filter

Link	Ports	VLAN ▲	VLAN ID	IP Group	Cloud Group	Compute Node Serial Number
IPAS Console Management Link	64	IPAS Console Management VLAN	26	...	...	...
link-A-42-B-42	42	link-A-42-B-42-vlans 24	24	ipGroupBig	uCloud	SN#1079AEB
link-A-42-B-42	42	link-A-42-B-42-vlans 24	24	IpGroup1	Shared	SN#1079B0B
link-A-42-B-42	42	link-A-42-B-42-vlans 24	24	IpGroup1	Shared	SN#1079ACB
link-A-42-B-42	42	link-A-42-B-42-vlans 24	24	IpGroup1	Shared	SN#1079ADB
link-A-44-B-44	44	link-A-44-B-44-vlans 27	27	IpGroup2	uCloud	SN#1079AEB
link-A-44-B-44	44	link-A-44-B-44-vlans 27	27	MKS Console IP Group	...	...

Figure 11-20 Link Connections

In the same window with Network Topology, you can also see a table that is called *Compute Nodes* that shows which VMs are on each compute node. You can use this capability to see the high-level data flow within the rack.

11.6.2 Network switches

For a more detailed view, you can look at the individual network devices. Although PureApplication System does not allow direct management of the switches, it does provide limited access to run certain commands.

In the system console, click **Hardware** → **Network Devices**, which opens a menu of the switches in the rack. You can choose either of the TOR switches or any of the switches in the chassis.

The TOR that is used in PureApplication System is a Blade Network Technology (BNT) Networking Operating System RackSwitch G8264. You can see the firmware and software levels by clicking **Hardware** → **Network Devices** in the system console.

Running the “show vlan” command from the user interface displays the VLAN configuration for the switch (Figure 11-21).

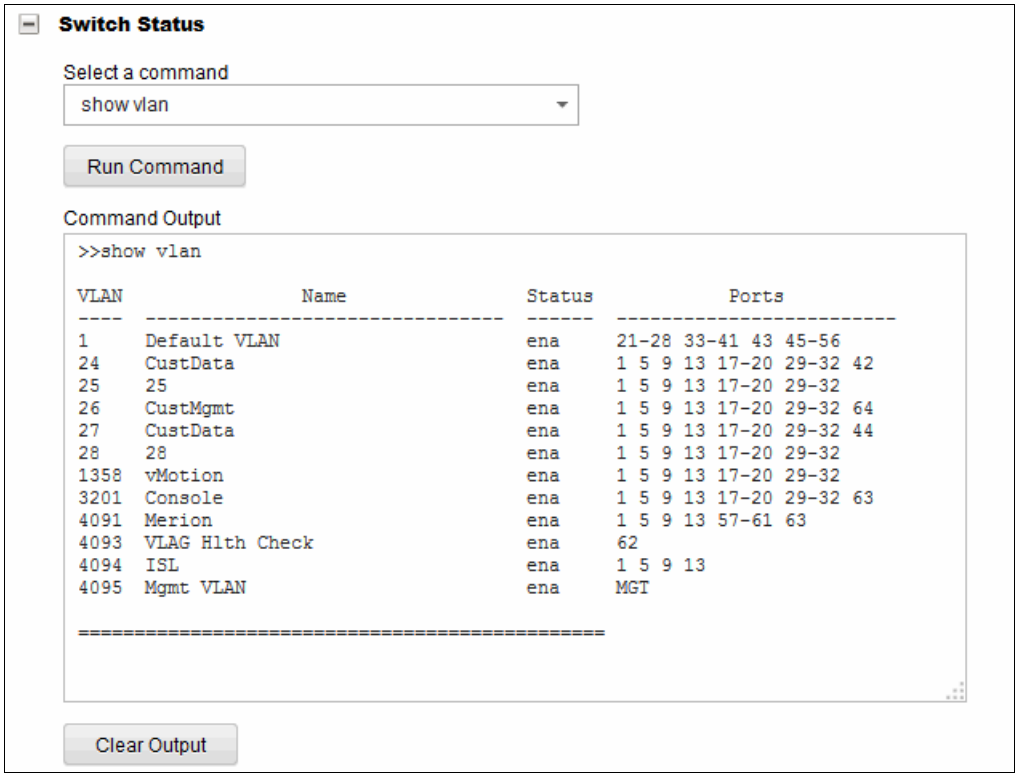


Figure 11-21 Running a read-only command on network switches

You can view the “show vlan” command output to verify exactly how each VLAN is carried in and out of the TOR switches.

The other command that you can use on the TOR switches is “show mac-address-table”. This command shows you all of the Media Access Control (MAC) addresses that it has cached (Figure 11-22). This list tells you all the MAC addresses along with the port and VLAN information that is known. This capability is useful for determining where communication is lost between the PureApplication System and customer data networks if there is a network problem.

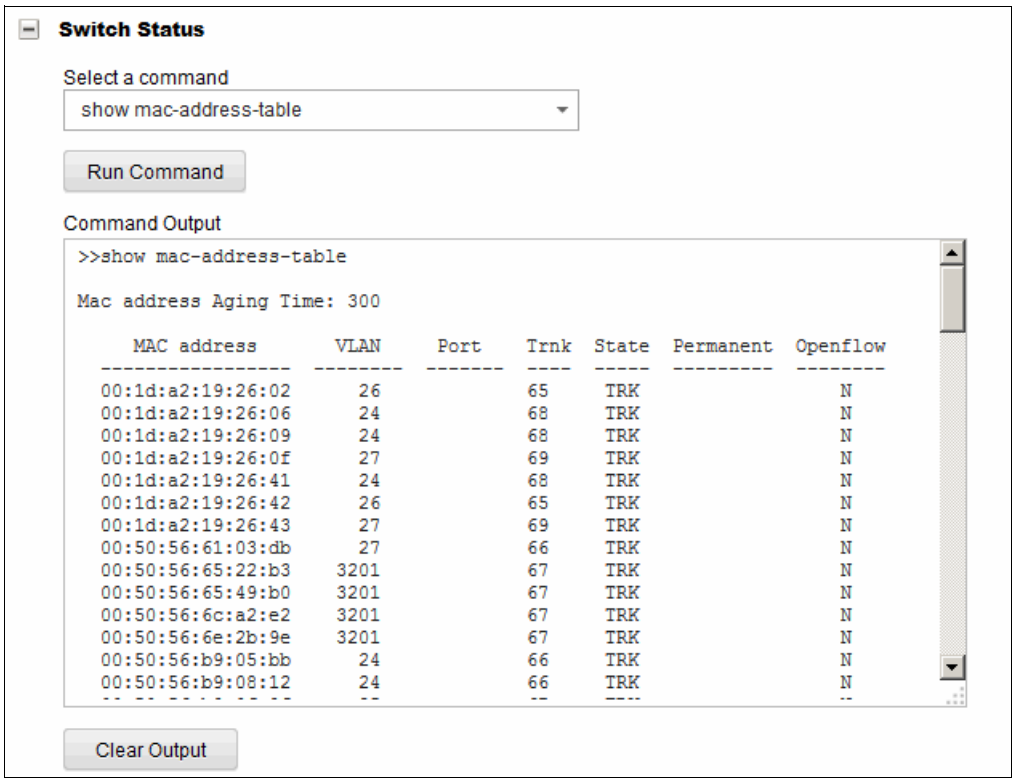


Figure 11-22 show mac-address-table command output

Chassis Network Switch is a model EN4093 Ethernet switch. It has some of the same commands available as the TOR. The most useful is “show vlan” (Figure 11-23 on page 331).

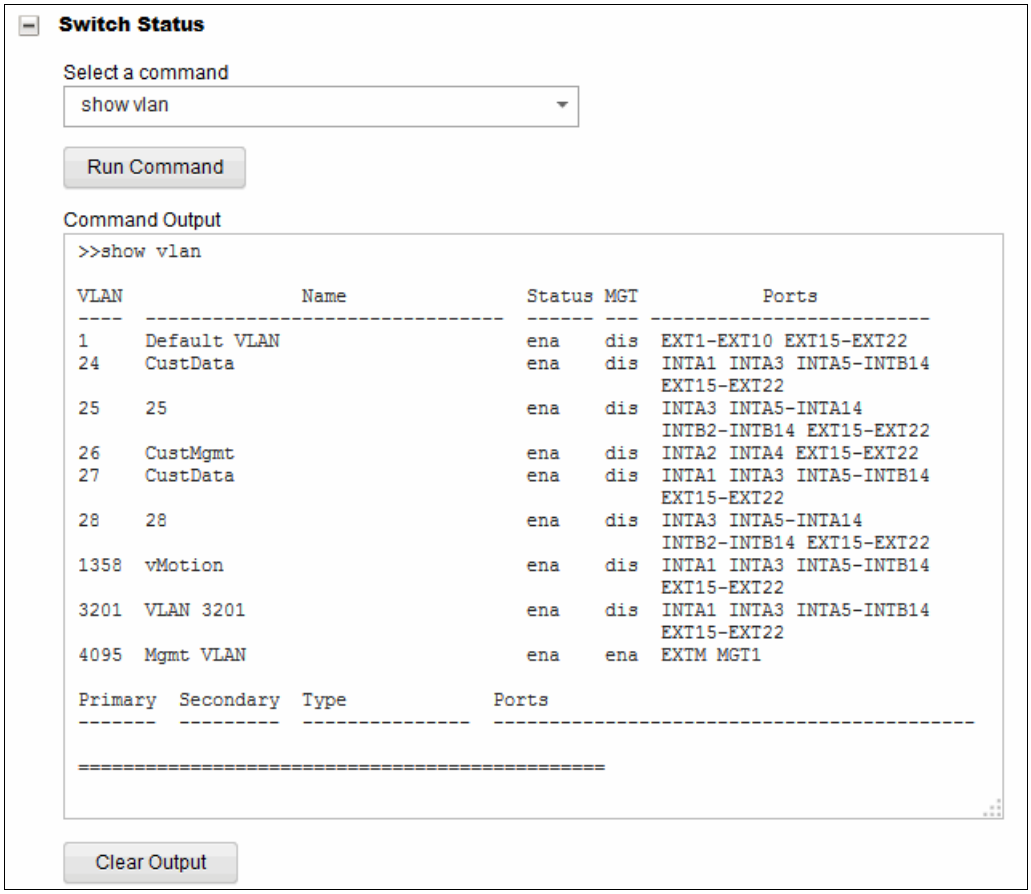


Figure 11-23 Run show vlan command on the chassis switch

11.6.3 SAN switches

The SAN switch is a IBM Flex System FC5022 12-port 16Gb ESB SAN Scalable Switch.

You can run commands to see the setup of the switch and the zone configuration. From the zone information, you can see the security configuration that keeps the volumes for each cloud group separate from the others.

This capability is even more interesting when using PureApplication System features that require Fibre Channel connectivity external to the rack that includes block storage replication and external block storage. In these cases, you see the port configuration in the chassis SAN switches for the additional ports.

For Block Storage Replication, the additional ports plug into the SAN switches on another PureApplication System rack. For external storage, these ports plug directly into a Storage Volume Controller. In either case, the additional ports show up in the SAN switch configuration so that you can verify that the settings are valid and match your security standards.

For more information about storage, see Chapter 5, “Storage” on page 103.

12

Service and Support Manager

Service and Support Manager (also known as Call Home) is an optional feature that comes with IBM PureApplication System V2.0 or later. It is an automated service ticket generation process that detects problems based on hardware or management software Call Home tagged events and reports them to IBM Support.

Service and Support Manager is available just for the PureApplication System, and it helps to streamline and simplify the support process with automated problem reporting and log or trace file collection. At the time of writing, PureApplication Software does not have this feature. However, IBM plans to add a Call Home feature to PureApplication Software for future releases.

This chapter describes specific details about the Service and Support Manager configuration for PureApplication System.

This chapter covers the following topics:

- Service and Support Manager overview
- Enabling Service and Support Manager
- Service and Support Manager workflow
- Service tickets (Call Home events)
- Working with service tickets
- List of Call Home events

© Copyright IBM Corp. 2015. All rights reserved.

12.1 Service and Support Manager overview

Service and Support Manager improves and simplifies the support process by reporting critical problems to IBM Support when a problem occurs on the system.

Service and Support Manager provides the following benefits to your organization:

- Notifies IBM automatically about critical problems
- Raises awareness of problems immediately
- Avoids unplanned service disruption
- Reduces the overall duration of service and support calls
- Reduces the costs by automating key components of the support process

Service and Support Manager allows PureApplication System to open a service ticket with IBM Support and automatically collect required trace logs when a hardware or management software Call Home tagged event occurs on the system. Trace information can be automatically uploaded to IBM Support to expedite problem resolution. Alternatively, you may choose to upload the trace files manually.

Specifically, Service and Support Manager enhances problem reporting with the following services:

- Automatic service ticket generation based on events to compute nodes, network, storage, power, cooling, and management software failures
- Automatic log collection and upload
- Collection of system configuration information

Service and Support Manager can generate a report that shows the service tickets that are generated by Call Home. System administrators can track actions with the comment feature. System administrators can use Call Home to add additional files that are relevant to the service ticket and upload those files to IBM Support servers.

12.2 Enabling Service and Support Manager

It is a preferred practice to enable and use Service and Support Manager to get the most benefit from PureApplication System support. However, Service and Support Manager is not enabled by default. If you want to enable it, you must configure the appropriate settings on the system.

Service and Support Manager can be configured by using the following services:

- IBM PureApplication System system console
- IBM PureApplication System command-line interface (pure.cli)
- IBM PureApplication System REST API

This publication shows how to configure Service and Support Manager by using the PureApplication System system console. For other configuration methods, such as using CLI or REST API, go to the IBM PureApplication System IBM Knowledge Center, found at the following website:

http://www.ibm.com/support/knowledgecenter/SSCRSX_2.1.0/pv_welcome.html

12.2.1 Access through a corporate firewall

To configure Service and Support Manager, PureApplication System needs access to the host names and IP addresses of the IBM Support servers. If your PureApplication System is at Version 2.1.0.1 or later level, use the host names, IP addresses, and port numbers that are shown in Table 12-1. Otherwise, use the host names, IP addresses, and port numbers that are shown in Table 12-2. The port numbers should be opened on the corporate firewall.

Table 12-1 Host names, IP addresses, and port numbers for PureApplication System V2.1.0.1 or later

Host name	IP addresses	Port number
esupport.ibm.com	129.42.56.189 129.42.58.189 129.42.60.189 129.42.54.189	443

Table 12-2 Host names, IP addresses, and port numbers for PureApplication System V2.0 and V2.1

Host name	IP addresses	Port number
eccgw01.boulder.ibm.com	207.25.252.197	443
eccgw02.rochester.ibm.com	129.42.160.51	443
www-945.ibm.com	129.42.26.224 129.42.34.224 129.42.42.224 129.42.50.224	443
www.ecurep.ibm.com	192.109.81.20	443
www.ibm.com	129.42.56.216 129.42.58.216 129.42.60.216	443 80 (optional)
www-03.ibm.com	204.146.30.17	443 80 (optional)
esupport.ibm.com	129.42.56.189 129.42.58.189 129.42.60.189 129.42.54.189	443 80 (optional)

Note: PureApplication System V2.0 and V2.1 use the same destination IP addresses and port numbers.

Each PureApplication System has two PureSystem Manager (PSM) nodes. Both PSMs share a floating IP address and each PSM has a fixed primary IP address and a fixed secondary IP address. These IP addresses are critical because they are used to connect to IBM for Call Home events.

You must allow the source IP addresses to connect to the destination IP addresses that are listed in Table 12-1 on page 335 or Table 12-2 on page 335. To verify these IP addresses, complete the following steps:

1. Click **System** → **Network Configuration**.
2. From the System Management IP section, review and verify the values for the following fields, as shown in Figure 12-1:
   - IPv4 address (floating IP address)
   - Primary IPv4 address
   - Secondary IPv4 address

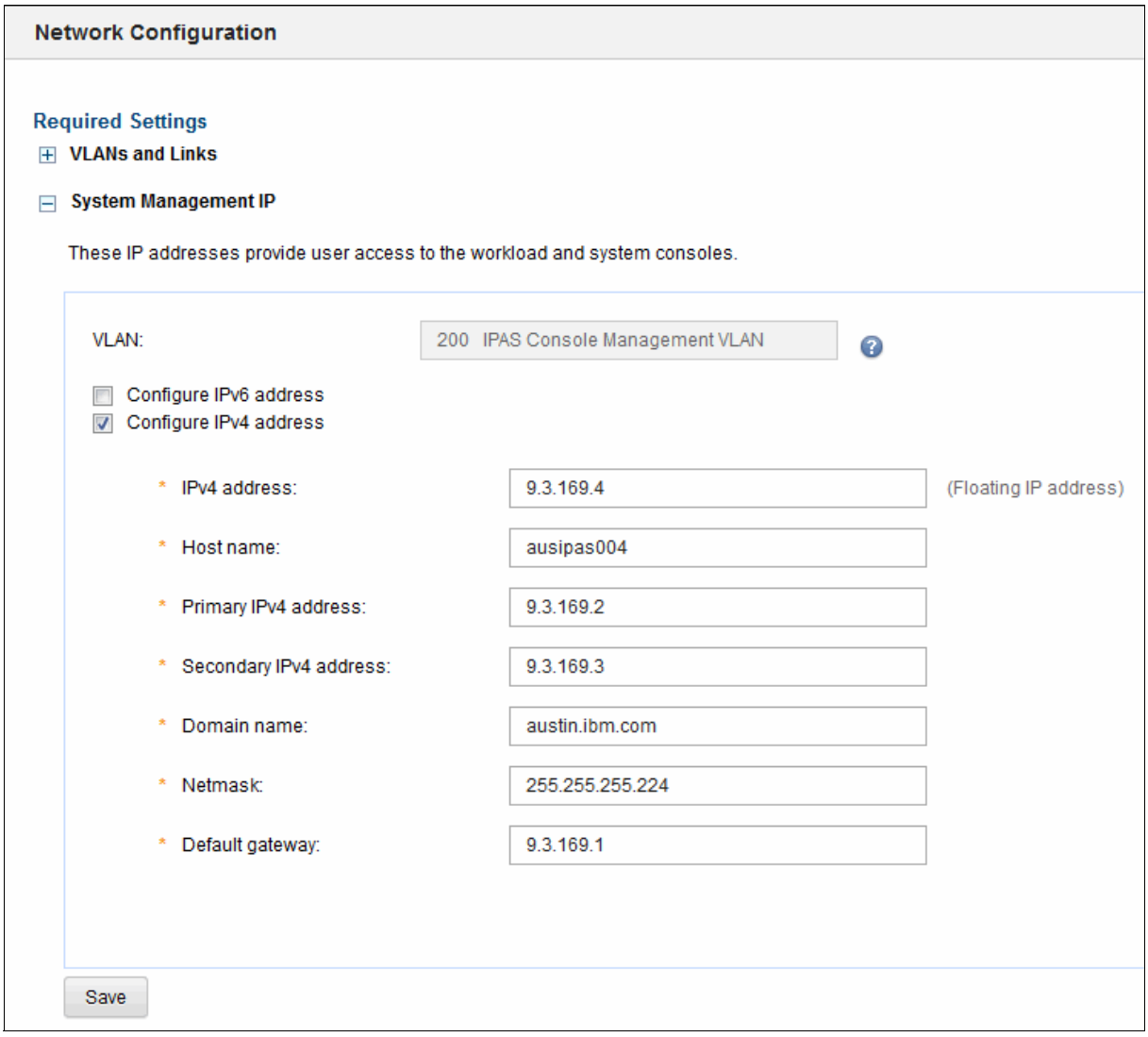


Figure 12-1 Get System Management IPs

You can always check that the Service and Support Manager requirements are up to date by going to the IBM Knowledge Center:

http://www.ibm.com/support/knowledgecenter/SSGRP3_2.1.0/doc/systemconsole/c_ss_req.dita

Select the the PureApplication System type, such as **W1700 2.1.0** → **Administrating the system** → **Configuring the system** → **Configuring system settings** → **Configuring Service and Support Manager** → **Service and Support Manager requirements.**

12.2.2 Configuring Service and Support Manager

Service and Support Manager is an optional feature that comes with PureApplication System so Service and Support Manager is not enabled by default. To enable it, you must configure the appropriate settings.

The Service and Support Manager configuration steps that are listed below should be performed by a user with hardware administration role with permission to manage hardware resources (full permission). Complete the following steps:

1. Click **System** → **System Settings**.
2. Expand **Service and Support Manager**.
3. In the Service and Support Level section, select the **Edit** icon and select the appropriate service level, as shown in Figure 12-2.

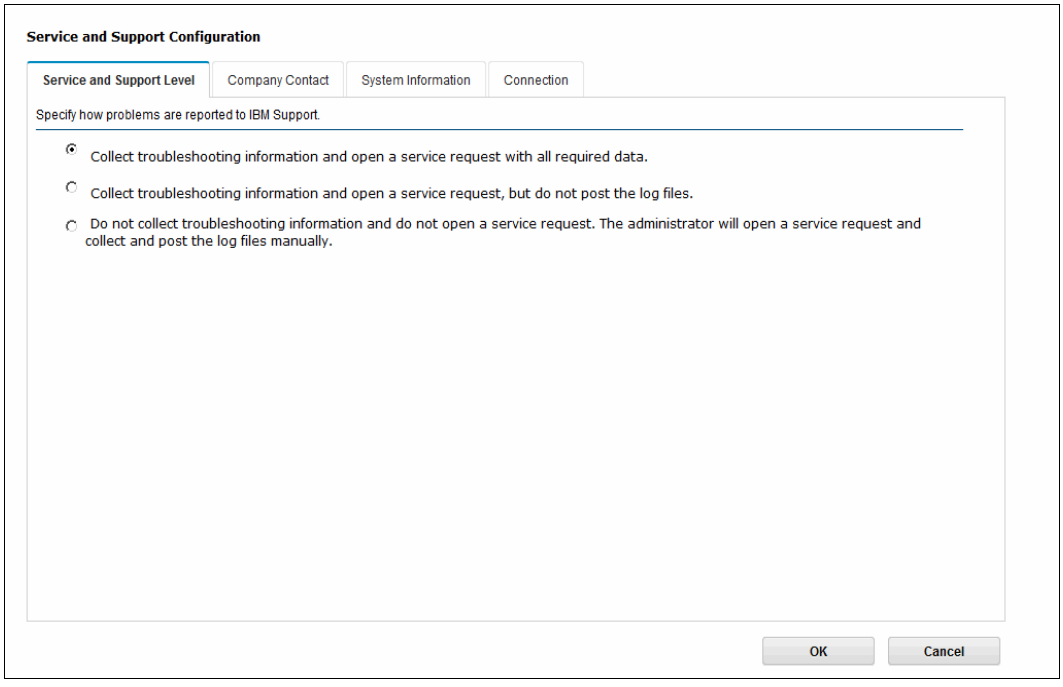


Figure 12-2 Service and Support level

Note: Make your selection based on the workflow of each option, which are summarized below:

- Collect troubleshooting information and open a service request with all required data.

  With this option, when a Call Home event is detected, a service ticket is opened automatically. The context-specific log collection set is generated immediately and then uploaded when it becomes available. Log collection times vary. This option is the preferred option because it expedites service ticket processing with IBM Support.

- Collect troubleshooting information and open a service request, but do not post the log files.

  With this option, when a Call Home event is detected, a service ticket is opened and the context-specific log collection set is immediately generated. However, the collection set is not uploaded to IBM Support. IBM Support acknowledges the service request and returns the service request, requesting that the log collection be uploaded.

  With this option, you can download the collection set for review before manually uploading it to IBM Support. Because this option is not fully automated, it might delay sending the required information to IBM Support.

  Collection sets do *not* remain available on the system indefinitely and they must be uploaded before they expire. The expiration date is typically within four days, but if storage space is required, collection sets might be removed sooner.

- Do not collect troubleshooting information and do not open a service request. The administrator will open a service request and collect and post the log files manually.

  With this option, Call Home events are set with the category Call Support on the console Events page.

  The administrator must open a service ticket and request collection of log files manually. IBM Support acknowledges the service request and returns the service request, requesting that the log collection be uploaded.

  This option is not preferred because of the time delay between event detection and manual collection of log information, which might cause important debug information to be missed because collection sets expire after four days. If there is not enough disk space on the system, they can be deleted automatically before four days.

4. Select the **Company Contact** tab and enter the information about the contact person and the company, as shown in Figure 12-3. Here are the required fields are:
   - Contact and company names
   - Telephone number
   - Country or region

   The contact is the person that IBM Support contacts. The person that is listed should be a representative from your company that can work with IBM Support to determine the problem and next steps. If you use a group email ID, or group phone number, test that IBM Support can reach the appropriate person quickly, and that the email is monitored.

   Enter the information in English. National language characters are not supported.

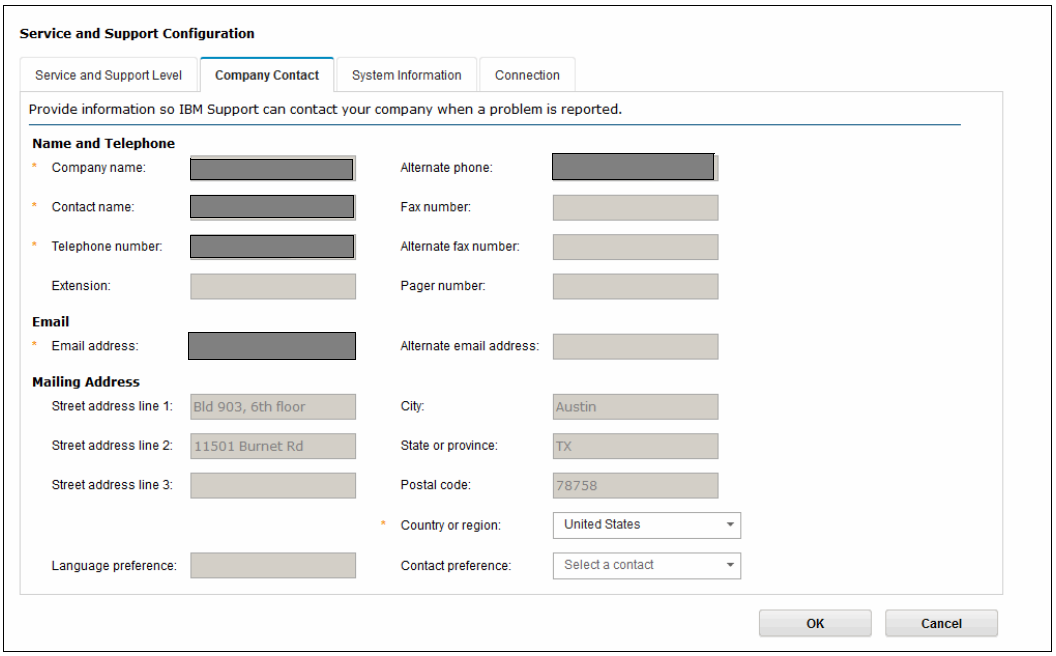


Figure 12-3 Company Contact tab

5. Select the **System Information** tab and enter the information about the system location, as shown in Figure 12-4. Here are the required fields:
   - Telephone number
   - Country or region
   - Street address
   - City
   - State or province
   - Postal code
   - Building

   The System Information tab is used for the data center address, which is used if an onsite service call is needed. Use the data center address for both the company contact and system information to make sure that service calls are processed quickly.

   Enter the information in English. National language characters are not supported.

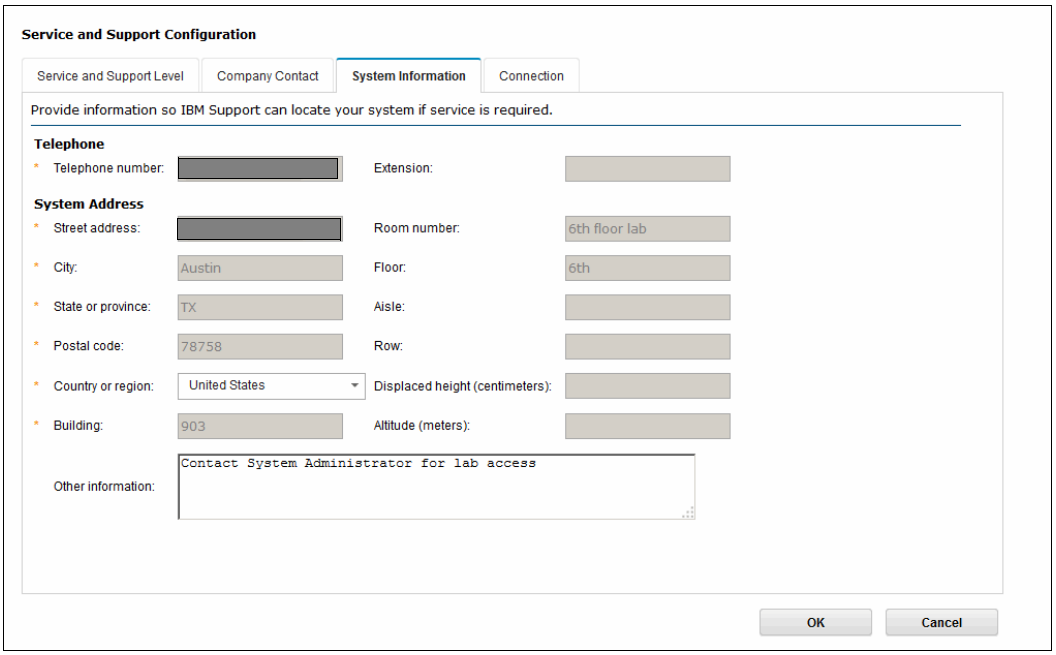


Figure 12-4 System Information

6. Select the **Connection** tab and specify the connection settings to be used by PureApplication System to contact IBM Support. Configure the following settings:
   - If PureApplication System has direct access to the internet, select **Connect to the Internet directly**.
   - If PureApplication System must use a proxy server for HTTP or HTTPS communication with IBM Support, specify the required HTTP settings and proxy server authentication credentials, if required. (Figure 12-5 on page 341).

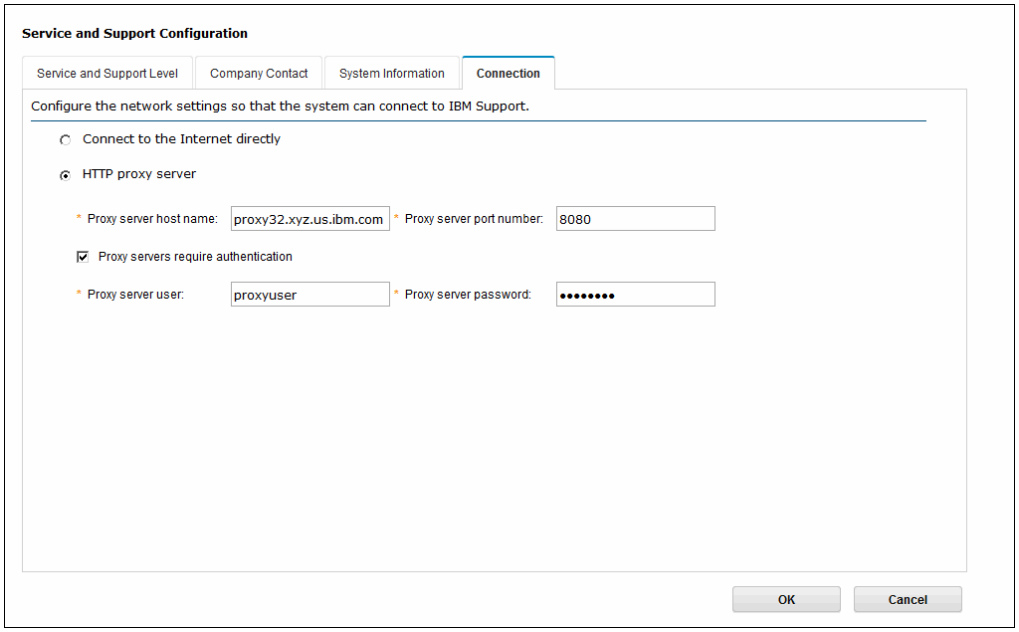


Figure 12-5 Configure the connection

7. Click **OK**.

Note: In some cases, customer proxy servers are configured to terminate SSL connections and re-encrypt them with their own certificates and then initiate a new SSL connection to the destination. This sort of proxy configuration can break the Create Test Ticket function, but the Test Connection function works fine. The Create Test Ticket function requires that the proxy server must not terminate the SSL connection and pass it through with the originally intended certificates.

12.2.3 Verifying the Service and Support Manager configuration

After Service and Support Manager is configured with all the required information, you can verify the configuration by testing the internet connection and then test a service ticket creation. Complete the following steps:

1. Click **System** → **System Settings**.
2. Expand **Service and Support Manager**.

3. At the bottom of Service and Support Manager section, there are two buttons: **Test Connection** and **Send Test Service Tickets**. Click **Test Connection**.
   - If the test is successful, the “Call Home connection test completed successfully” message is displayed as shown in Figure 12-6.

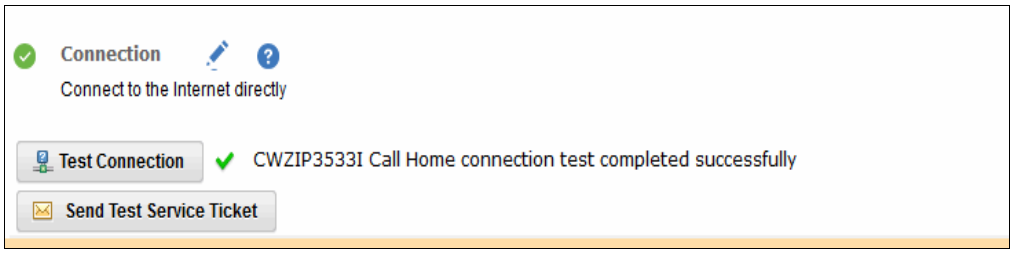


Figure 12-6 Test Connection

   - If the test fails, check the firewall and connection information.
4. After the connection is successfully tested, click **Send Test Service Ticket**.

   If the connection is successful, the “Call Home test service ticket send successfully” message is displayed, as shown in Figure 12-7.

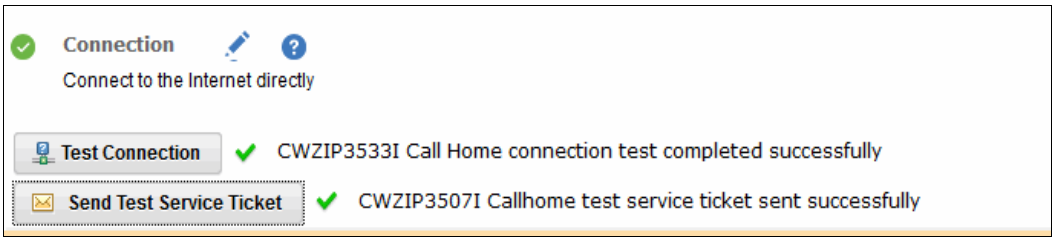


Figure 12-7 Send Test Service Ticket

5. After the Call Home test service ticket is sent successfully, there is a problem record on Problems window. You can view this record by completing the following steps:
   a. Click **System** → **Problems**.
   b. Check to see whether the problem record for the test service ticket was created, as shown in Figure 12-8.

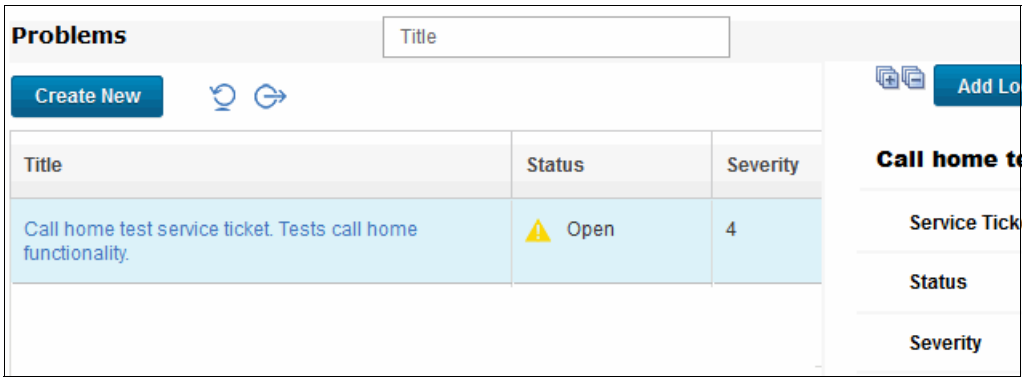


Figure 12-8 Call Home test service ticket problem record

6. Having confirmed that the problem record for test service ticket was created, the Service and Support Manager functionality is verified.

12.3 Service and Support Manager workflow

After the PureApplication event engine detects a problem based on Call Home tagged events to hardware or management software components, PureApplication System follows this workflow:

1. The PureApplication System event engine processes all events. If it finds Call Home tagged events, it makes a Call Home request (for the events that are not a duplicate of existing open problems).
2. The Call Home request is received by the Service and Support Manager.
3. Based on the configuration, the Service and Support Manager collects problem details, system information, and required log and trace files.
4. The Service and Support Manager creates a service ticket by communicating to IBM Support Servers. If it fails, it retries up to three times. If it fails three times, Service and Support Manager creates a problem record on the local system without any service ticket number (PMR Number) assigned to it. Then, another critical event is created and displayed on the Events page of PureApplication System, as shown in Figure 12-9.

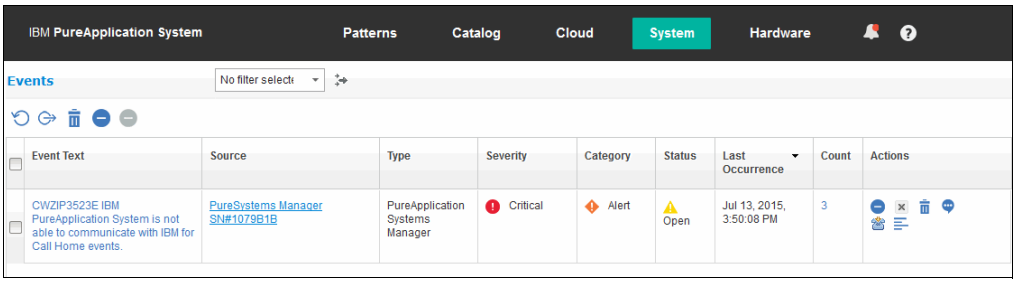


Figure 12-9 Event for a failed Call Home attempt

Note: A user can update a problem record that does not have an assigned service ticket number (PMR Number) by clicking **Add Service Ticket** for that specific problem record.

After the record is selected, the user has two choices for updating a problem record:

- Create a service ticket
- Import existing service ticket

You can find more information about this process in 12.5.1, “Add Service Ticket feature” on page 347.

5. If a service ticket is created successfully, the service ticket number (PMR number) is added to the problem record
6. A request closure is not an automatic procedure like all other BM Support tickets. A customer should request or confirm closure. When the problem is fixed, the customer can request closure by completing the following steps:

   a. Clicking **Close** in the Problems window.

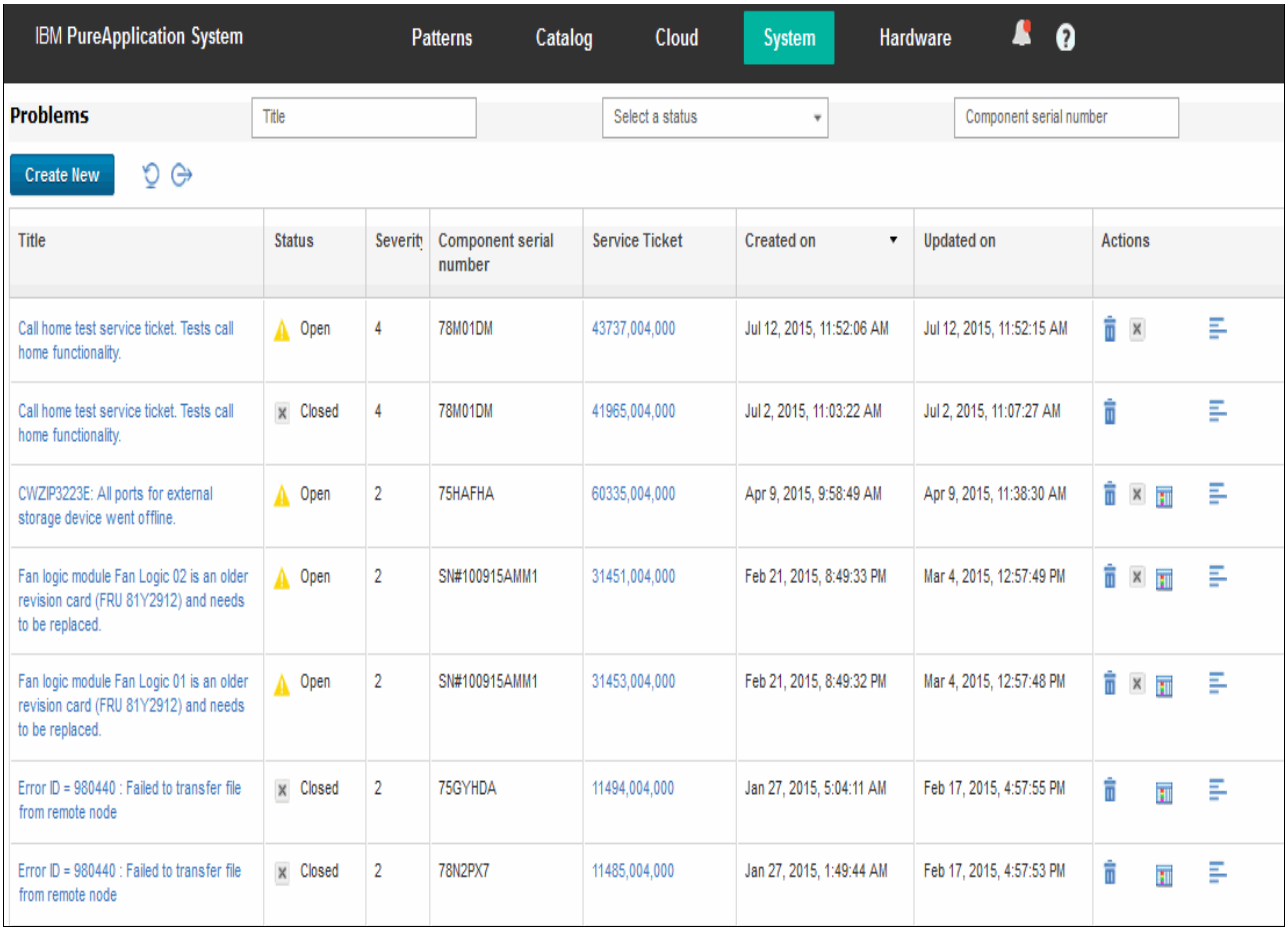
   b. Add a comment to the ticket that the client requested closure.

   The problem is marked and set to the Closed status. The service ticket must be closed by IBM.

12.4 Service tickets (Call Home events)

The PureApplication System system console Problems window contains information for all Call Home events that occurred on the system. These events represent problems that require assistance from IBM Support.

To list all the problems for Call Home events, click **System** → **Problems**. The system problem records for Call Home events are listed, as shown in Figure 12-10.



Title	Status	Severity	Component serial number	Service Ticket	Created on	Updated on	Actions
Call home test service ticket. Tests call home functionality.	Open	4	78M01DM	43737,004,000	Jul 12, 2015, 11:52:06 AM	Jul 12, 2015, 11:52:15 AM	
Call home test service ticket. Tests call home functionality.	Closed	4	78M01DM	41965,004,000	Jul 2, 2015, 11:03:22 AM	Jul 2, 2015, 11:07:27 AM	
CWZIP3223E: All ports for external storage device went offline.	Open	2	75HAFHA	60335,004,000	Apr 9, 2015, 9:58:49 AM	Apr 9, 2015, 11:38:30 AM	
Fan logic module Fan Logic 02 is an older revision card (FRU 81Y2912) and needs to be replaced.	Open	2	SN#100915AMM1	31451,004,000	Feb 21, 2015, 8:49:33 PM	Mar 4, 2015, 12:57:49 PM	
Fan logic module Fan Logic 01 is an older revision card (FRU 81Y2912) and needs to be replaced.	Open	2	SN#100915AMM1	31453,004,000	Feb 21, 2015, 8:49:32 PM	Mar 4, 2015, 12:57:48 PM	
Error ID = 980440 : Failed to transfer file from remote node	Closed	2	75GYHDA	11494,004,000	Jan 27, 2015, 5:04:11 AM	Feb 17, 2015, 4:57:55 PM	
Error ID = 980440 : Failed to transfer file from remote node	Closed	2	78N2PX7	11485,004,000	Jan 27, 2015, 1:49:44 AM	Feb 17, 2015, 4:57:53 PM	

Figure 12-10 System Problem records

The Problems window shows the following details for each service ticket:

- Title
- Status (Open, Closed)
- Severity (1, 2, 3, or 4)
- Component Serial Number
- Service Ticket Number (PMR Number)
- Service Ticket Creation Time
- Update Time
- Actions (Delete, Close, Link to Events Page, and Show Details)

If you select the title of a problem record, all information about that problem is displayed in the right pane, as shown in Figure 12-11 on page 345.

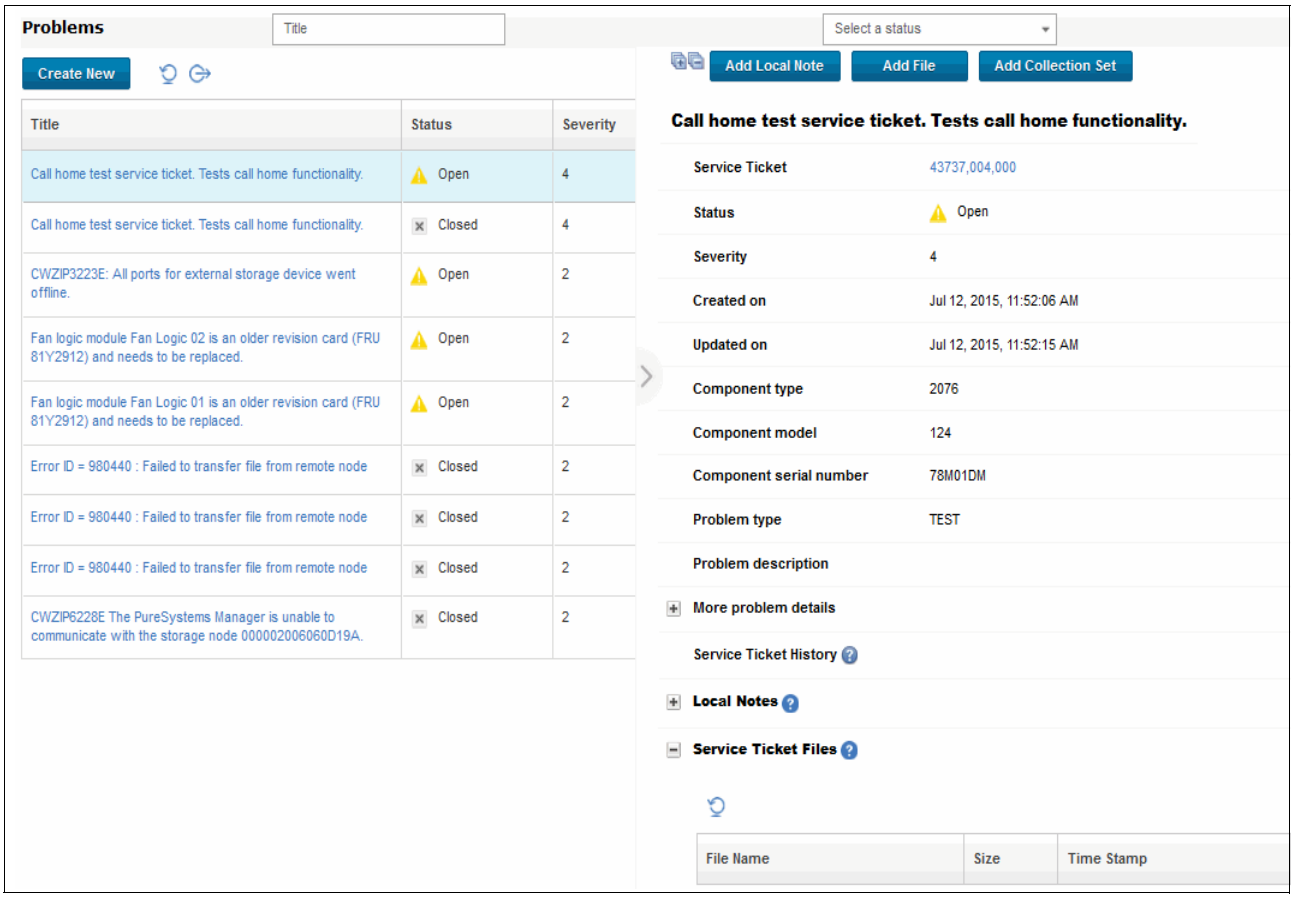


Figure 12-11 Problem details

12.5 Working with service tickets

After Service and Support Manager creates service tickets, the Problems window on the system console can be used to interact with service tickets and IBM Support. The interactions include but not limited to the following items:

- Add a service ticket number (see “12.5.1, “Add Service Ticket feature” on page 347”)
- Add file
- Add log collection sets
- Add local note
- Close service tickets
- Delete service tickets
- Listing of all service tickets

The PureApplication System system console Problems window contains information for all Call Home events that occurred on the system. These events represent problems that require assistance from IBM Support.

From this window, you work with problems for all Call Home events and interact with problem records, such as adding a service ticket, uploading collection sets, and posting comments.

Here are some of the actions for all problem records:

- Listing all problem records
- Sorting problem records based on title, status, severity, component serial number, service ticket, and created on and updated on fields of the records
- Taking actions, such as delete, close, link to event page, add service ticket, and show details for a specific problem record

It is possible to work with a specific problem record by clicking the title of the problem record in the list. When you click the title of any problem record, a new pane opens on the right, as shown in Figure 12-12.

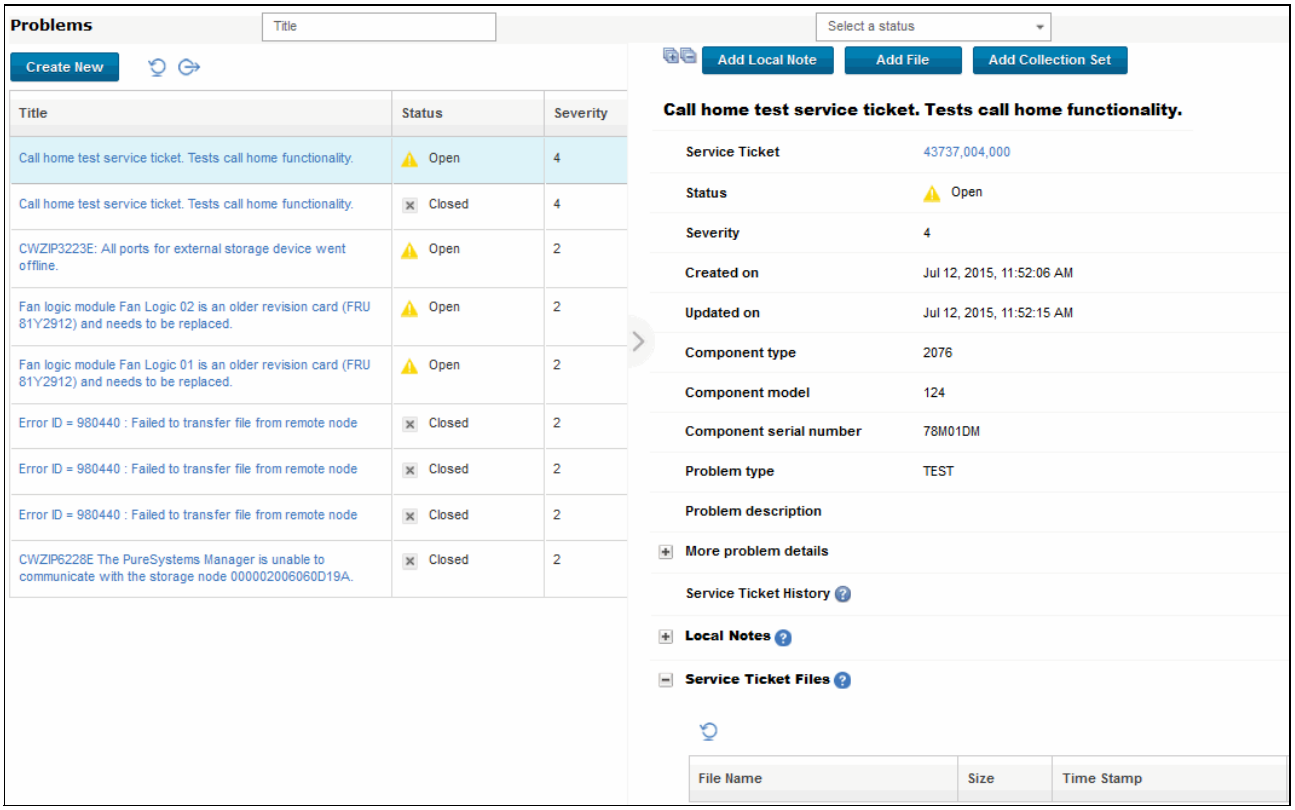


Figure 12-12 Details for a specific problem record

In the problem details section of a specific problem record, the following information is displayed:

- Service Ticket

  Shows the service ticker number. If there is no assigned service ticket number because of communication problem, the **Add Service Ticket** button is displayed.
- Status (Open or Closed)
- Severity (1, 2, 3, or 4)
- Created on
- Updated on
- Component type

  Shows the component type for the problem record. Component type can be racks, management_nodes, tor_network_switch, and so on.

- Component model
- Component serial number
- Problem type

  Shows the type of the problem. It can be BNTTOR, IPASSW, V7000, and so on. It is based on the nature and component source of the problem.
- Problem description
- More problem details

  Shows further details about the problem, such as message key, error code, event text, system state, and the original address of the component.
- Events

  List of events about the problem.
- Service Ticket History

  Shows the service ticket history. Lists service ticket creation, log collection, file add or delete action date, and so on.
- Local Notes

  These notes are on the local system and are not attached to the service ticket. To add notes to the service ticket, select the **service ticket link** at the top.
- Service Ticket Files

  You can select files from local and system logs, and upload them to the IBM service ticket system.

12.5.1 Add Service Ticket feature

You can add a service ticket to an existing problem record that does not have an assigned service ticket. For problem records with unassigned service tickets, see the following note.

Note: For a list of cases for a problem record with unassigned service ticket, the following process occurs:

1. After a Call Home event is processed by the PureApplication event engine, the event engine calls the Service and Support Manager, and then the Service and Support Manager creates a service ticket by communicating to IBM Support Servers. If it fails to communicate to the IBM Support servers, it retries up to three times. If it fails three times, Service and Support Manager creates a problem record on the local system *without any service ticket number* (PMR Number) assigned for it.
2. If Service and Support Level is configured as **Do not collect troubleshooting information and do not open a service request** when the Service and Support Manager was configured, problem records do not have an assigned service ticket.

To add a ticket, complete the following steps:

1. Click **System** → **Problems** on the PureApplication System system console. The system problem records for Call Home events are listed, as shown in Figure 12-10 on page 344.
2. Select the title of one of the problem records to see the details of the problem in the right pane.

3. Click **Add Service Ticket**, as shown in Figure 12-13.

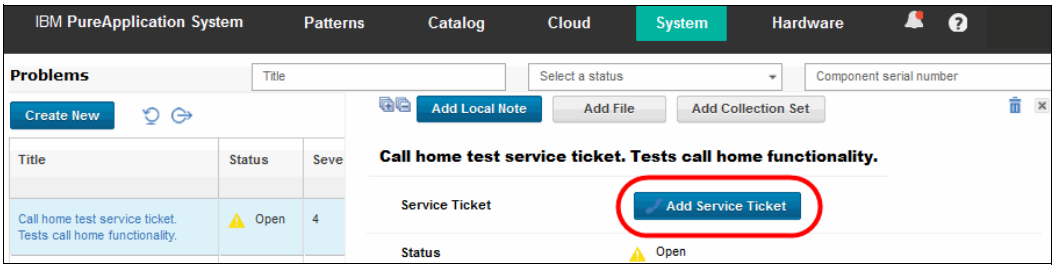


Figure 12-13 Add Service Ticket button

The Add Service Ticket window opens, as shown in Figure 12-14.

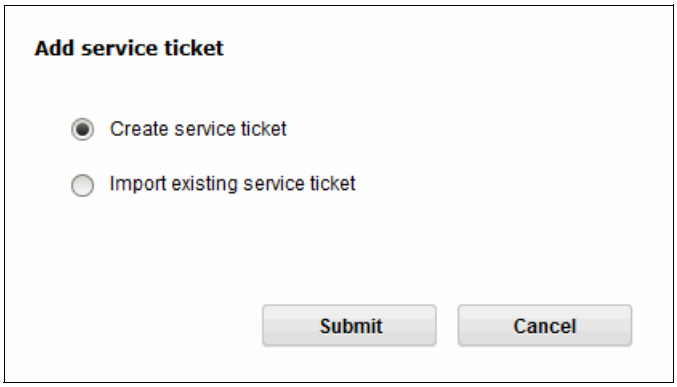


Figure 12-14 Add Service Ticket window

4. Select either **Create service ticket** to create a service ticket (Figure 12-14) or **Import existing service ticket**, as shown in Figure 12-15. If you select **Import existing service ticket**, enter the service ticket number (PMR number).

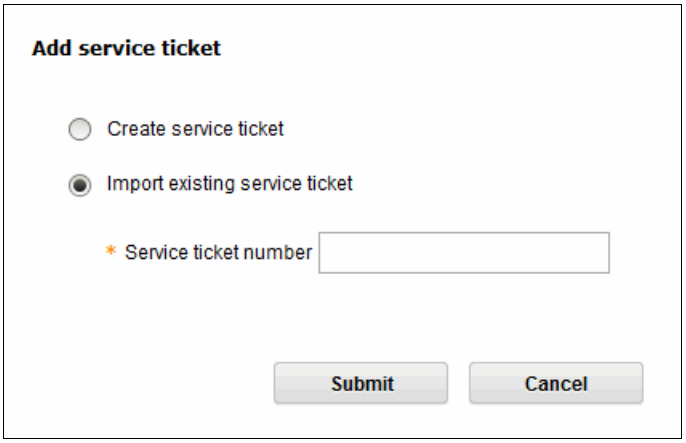


Figure 12-15 Import the existing service ticket window

5. Click **Submit**.

Note: The service ticket number length is 11, and its data format is 00000,000,000. It is also known as a Problem Management Record (PMR) number.

12.5.2 Add File to a Service Ticket feature

You can add files to a service ticket and then upload those files to IBM Support or download them to your local workstation. However, the Add File and Add Collection Set features work with problems records only if that problem record has an assigned service ticket number. Otherwise, the Add File and Add Collection Set features are disabled by default.

To add files to a service ticket, complete the following steps:

1. Click **System** → **Problems** on the PureApplication System system console. The system problem records for Call Home events are listed, as shown in Figure 12-10 on page 344.
2. Select the title of the problem record to see the details of the problem in the right pane.
3. Click **Add File** either at the top of the window or under the service ticket files tab, as shown in Figure 12-16.

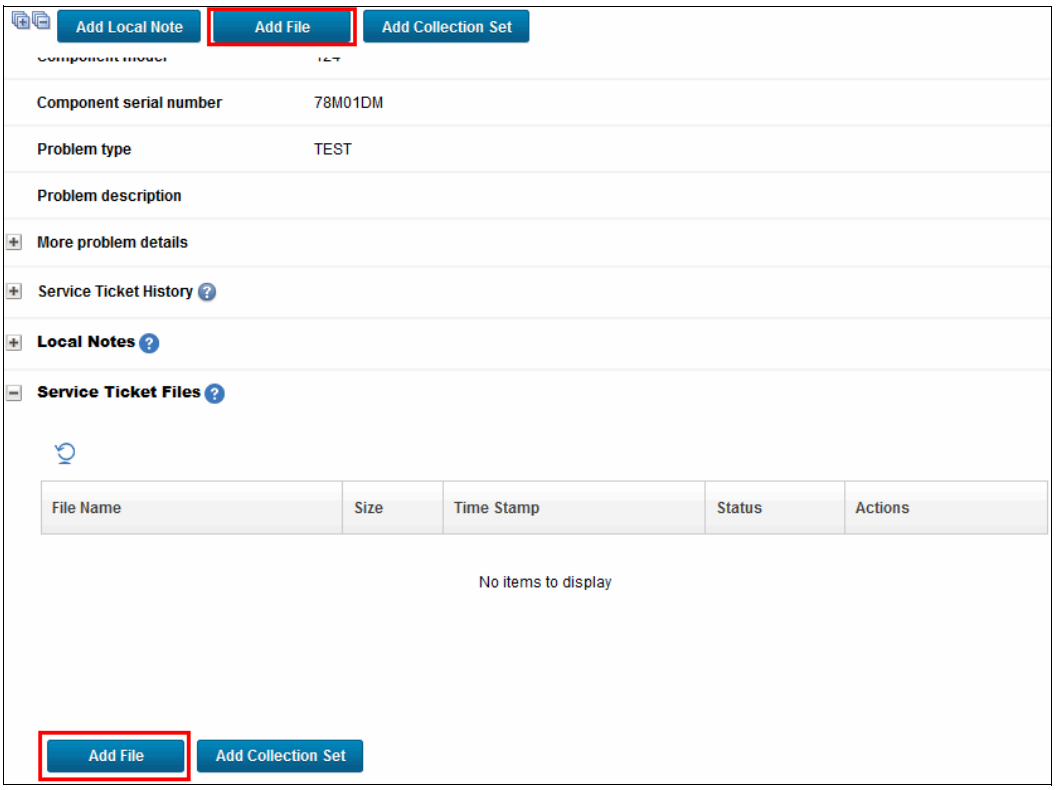


Figure 12-16 Add File buttons on the problem details window

4. The Import file to upload window opens, as shown in Figure 12-17.

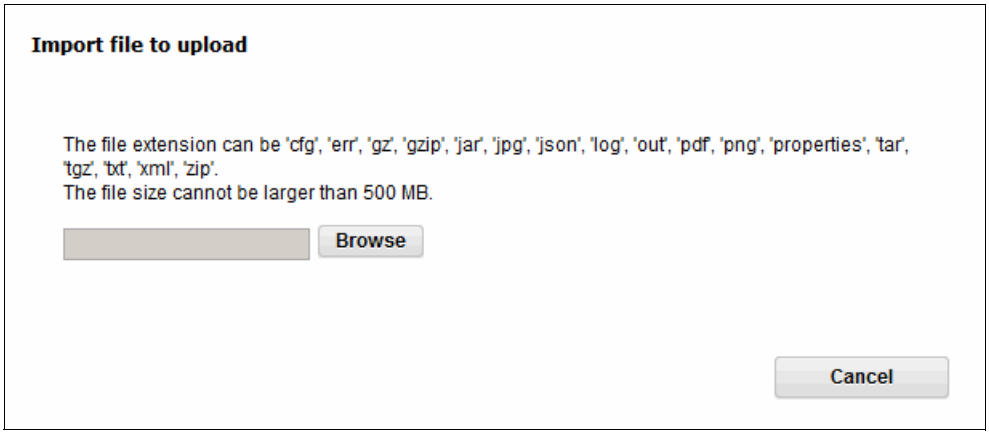


Figure 12-17 Add File window

5. Click **Browse** to select the file on your local workstation, and then select **OK**.
6. After the file uploads successfully on to PureApplication System, you can see your file under the Service Ticket Files tab, as shown in Figure 12-18.

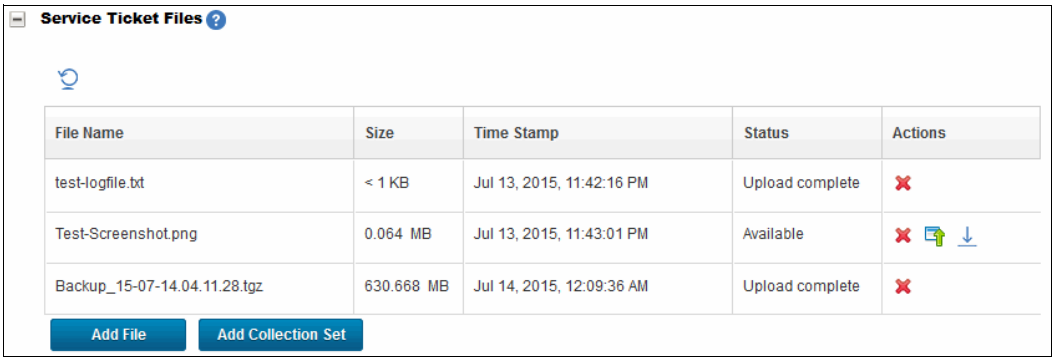


Figure 12-18 Service Ticket Files

12.5.3 Add Collection Set to a Service Ticket feature

You can add collection sets to a service ticket and then upload those files to IBM Support or download to your local workstation. However, Add File and Add Collection Set features work with problems records only if the problem record has an assigned service ticket number. Otherwise, the Add File and Add Collection Set features are disabled by default.

To add collection sets to a service ticket, complete the following steps:

1. Click **System** → **Problems**. The system problem records for Call Home events are listed, as shown in Figure 12-10 on page 344.
2. Select the title of one of the problem records to see the details of the problem in the right pane.
3. Click **Add Collection Set** either at the top or under the service ticket files tab, as shown in Figure 12-19 on page 351.

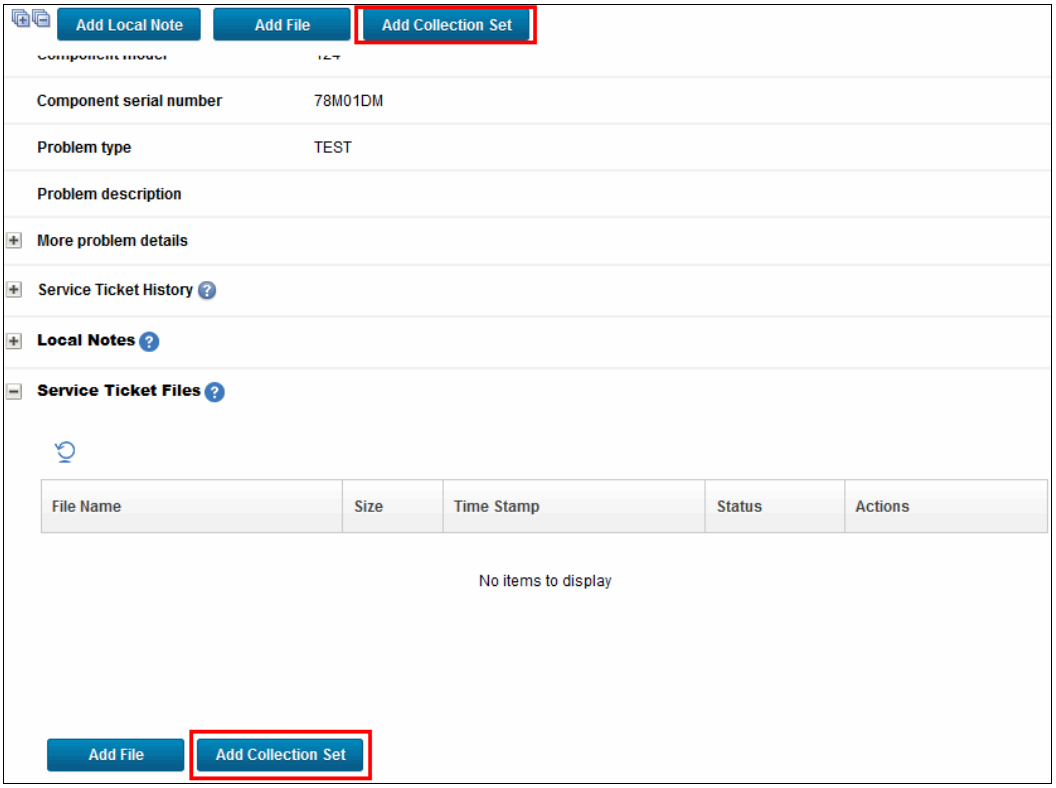


Figure 12-19 Add Collection Set buttons

Note: Before adding collection sets to a service ticket, you should generate collection sets by clicking **System Console** → **System Troubleshooting** → **System Logs** → **Request System Logs** → **Collection Sets**. Collection sets expire after four days. If there is not enough disk space on the system, they can be deleted automatically before four days.

4. Select the collection set you want to add and click **OK**.

12.5.4 Uploading, downloading, and deleting service ticket files

After you add files or collection sets to a service ticket, you can manage those files and collection sets from the Service Ticket Files tab, as shown in Figure 12-20.

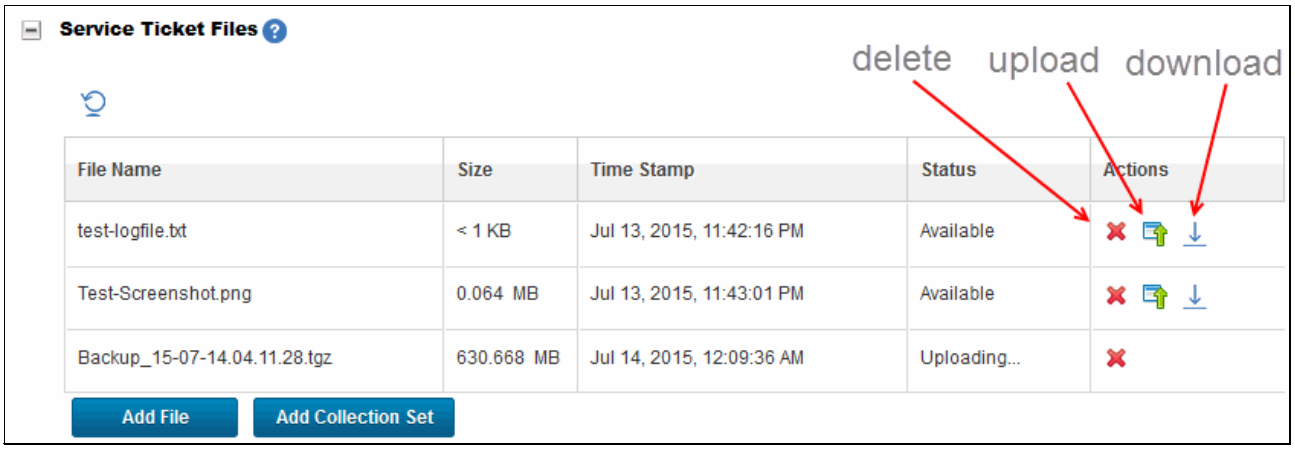


Figure 12-20 Work with service ticket files

12.5.5 Add Local Note feature

Local notes are on the local system, and they are not attached to the service tickets. Local notes are for the local system administration team to work together on problems.

Note: If you want to add notes to the service ticket, you can select the service ticket number link at the top, which takes you to the IBM Service Request tool on the internet.

To add local notes to a service ticket, complete the following steps:

1. Click **System** → **Problems**. The system problem records for Call Home events are listed, as shown in Figure 12-10 on page 344.
2. Select the title of the problem record to see the details of the problem in the right pane.
3. Click **Add Local Note**, either at the top or under the local notes tab, as shown in Figure 12-21.

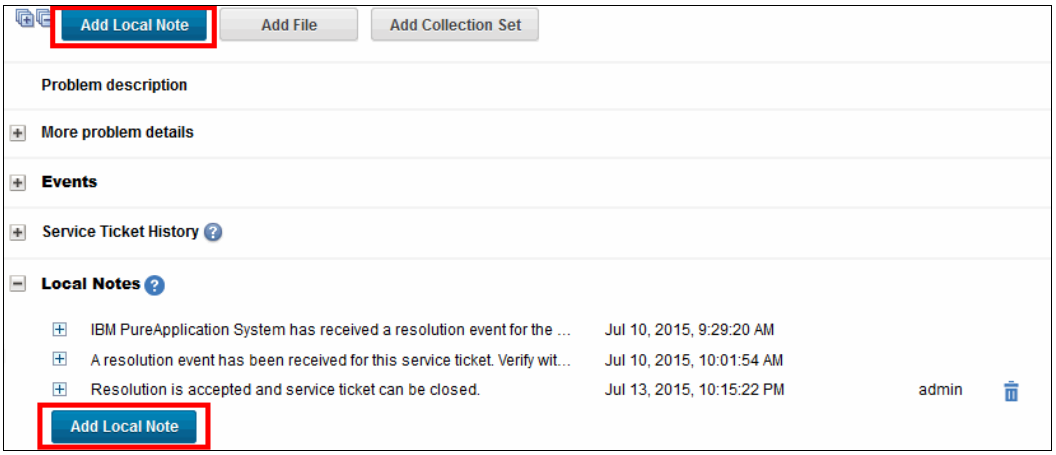


Figure 12-21 Add Local Note buttons

4. The Comment(s) window opens, as shown in Figure 12-22.

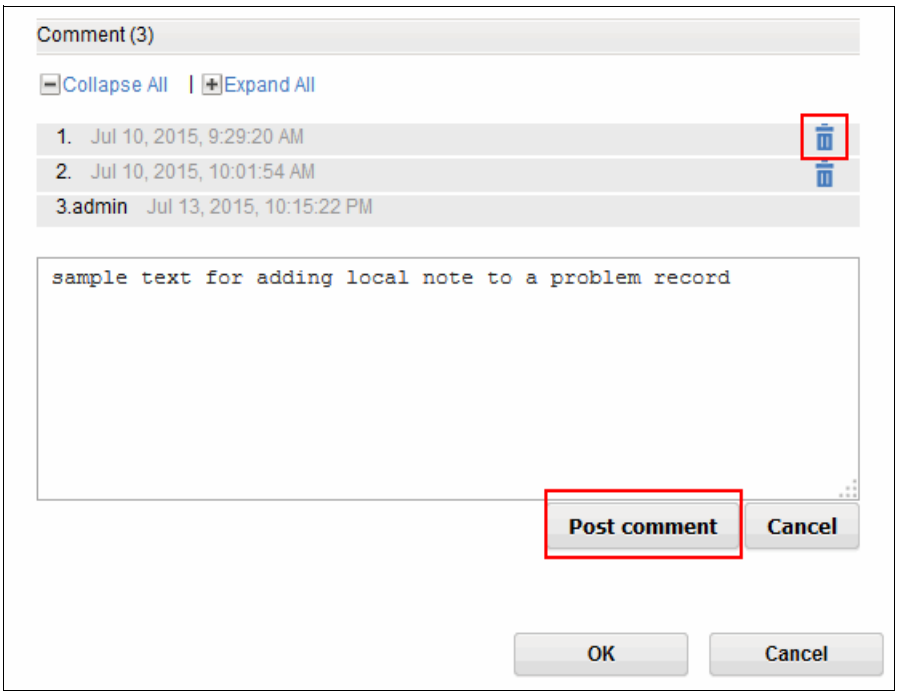


Figure 12-22 Add Local Note (Comments) window

5. Type your comment and click **Post comment**.
6. In the same window, you can add another comment or delete previous comments by selecting the delete icon at the upper right.
7. Click **OK**.

12.6 List of Call Home events

Reference information for a list of Call Home events provides a consolidated view of all Call Home events that can be received when system or hardware failures occur. That reference list can be found in the PureApplication System v2.x IBM Knowledge Center:

`http://www.ibm.com/support/knowledgecenter/SSGRP3_2.1.0/doc/iwd/ch-event-ref.dita`

Select your PureApplication System type, such as **W1700 2.1.0** → **Troubleshooting and support** → **Events and error messages** → **Call Home Event Reference**.

13

IBM PureApplication System command-line interface and REST API

This chapter describes the PureApplication System command-line interface (pure.cli) and its REST API.

This chapter covers the following topics:

- Pure.cli
- PureApplication System REST API
- CLI recipes

© Copyright IBM Corp. 2015. All rights reserved.

13.1 Pure.cli

The pure.cli is a Jython-like environment where you run commands to manage the PureApplication System. The environment enables a high degree of automation through scripting. Jython is a Python implementation of Java. As such, a Jython script can import both Python modules and Java classes. Pure.cli can take advantage of the vast number of the Python modules while being portable because of its ability to run on Java virtual machines (JVMs).

13.1.1 Pure.cli basics

Pure.cli imports two primary packages: admin and deployer. The admin package contains objects and functions that users can use to manage the system side of the rack. The deployer package contains resources that are associated with the workload side of the rack. In Version 2.1, this distinction is no longer obvious on the UI. However, the underlying architecture remains the same regarding the pure.cli.

Because the pure.cli is implemented on top of Jython and resource information is stored as JSON objects, native Python data types, such as dict, list, set, and tuple, are the primary containers.

Within the PureApplication System packages, a collection of object instances, such as a collection of users, can be referenced. Example 13-1 shows a command-line interface (CLI) call to reference the global collection of `admin.users`.

Example 13-1 Reference an object of <class 'admin.resources.user.Users'>

```
>>> admin.users
```

Example 13-2, shows a CLI call to reference the global collection of `admin.users` as a Python list.

Example 13-2 Reference an object of <type 'list'>

```
>>> admin.users.list()
```

The first command returns a generic collection object (`admin.resources.user.Users`), and the second returns a Python list of users. The generic collection cannot use native Python list functions. For example, the slice notation works only after you call **`list()`** on the resource, as shown in Example 13-3.

Example 13-3 The collection object referenced by admin.users does not support slice notation

```
>>> admin.users.list()[4:5]
[{
  "auth_mode": "internal",
  "bidi_national_calendar": None,
  "bidi_text_direction": None,
  "created_time": "Nov 17, 2014 2:35:39 PM",
  "current_status": None,
  "email": "test@user.com",
  "groups": (nested object),
  "id": "31fe4acf-9dff-4037-b050-ac8447187cdf",
  "name": "Default Admin",
  "password": (write-only),
```

```
  "roles": (nested object),
  "shellaccounts": (nested object),
  "url": "/admin/resources/users/31fe4acf-9dff-4037-b050-ac8447187cdf",
  "user_id": "admin"
}]
>>> admin.users[4:5]
Traceback (most recent call last):
  File "<stdin>", line 1, in <module>
  File
"/jumper/pure.cli/lib/2.0.0.1-201506101944840/deployer/resources/resource.py",
line 498, in __getitem__
    raise TypeError('unsupported operand type for __getitem__: %s' % type(key))
TypeError: unsupported operand type for __getitem__: <type 'slice'>
```

Both of these types are equally useful. You can use the generic collection to reference an object instance by using an index or search for an object instance by using a key, as shown in Example 13-4. You can use the Python list `admin.users.list()` to use Python constructs such as slice and list comprehension, as shown in Example 13-4.

Example 13-4 The admin.users collection object supports reference by index and fuzzy search by ID

```
>>> admin.users[4]
{
  "auth_mode": "internal",
  "bidi_national_calendar": None,
  "bidi_text_direction": None,
  "created_time": "Nov 17, 2014 2:35:39 PM",
  "current_status": None,
  "email": "test@user.com",
  "groups": (nested object),
  "id": "31fe4acf-9dff-4037-b050-ac8447187cdf",
  "name": "Default Admin",
  "password": (write-only),
  "roles": (nested object),
  "shellaccounts": (nested object),
  "url": "/admin/resources/users/31fe4acf-9dff-4037-b050-ac8447187cdf",
  "user_id": "admin"
}
>>> admin.users["labadmin"]
[{
  "auth_mode": "internal",
  "bidi_national_calendar": None,
  "bidi_text_direction": None,
  "created_time": "Nov 21, 2014 7:41:27 PM",
  "current_status": None,
  "email": None,
  "groups": (nested object),
  "id": "5ba0e8d4-4150-4e76-8ae3-6d7df0907424",
  "name": None,
  "password": (write-only),
  "roles": (nested object),
  "shellaccounts": (nested object),
  "url": "/admin/resources/users/5ba0e8d4-4150-4e76-8ae3-6d7df0907424",
  "user_id": "labadmin"
}]
```

Effectively, `admin.users[“labadmin”]` is identical to `admin.users.list({“user_id”:”labadmin”})`, as shown in Example 13-5. Each object collection has a default key that a user can use to search, thus enabling calls such as `admin.users[“labadmin”]`. This default key is typically the human-friendly name of the object instance as displayed on the web UI. In the case of the user object, the key is `user_id`.

Example 13-5 admin.users.list({‘user_id’:’labadmin’}) returns the same list as admin.users[‘labadmin’]

```
>>> admin.users.list({'user_id':'labadmin'})
[{
  "auth_mode": "internal",
  "bidi_national_calendar": None,
  "bidi_text_direction": None,
  "created_time": "Nov 21, 2014 7:41:27 PM",
  "current_status": None,
  "email": None,
  "groups": (nested object),
  "id": "5ba0e8d4-4150-4e76-8ae3-6d7df0907424",
  "name": None,
  "password": (write-only),
  "roles": (nested object),
  "shellaccounts": (nested object),
  "url": "/admin/resources/users/5ba0e8d4-4150-4e76-8ae3-6d7df0907424",
  "user_id": "labadmin"
}]
```

When you search by a key, it returns a list. The output in Example 13-5 is a list of one instance. To reference the user instance `labadmin`, you must use `admin.users[‘labadmin’][0]`. When searching for instances in this fashion, you produce a *fuzzy search*. If the key you passed in is not unique or is a partial match for another instance, it returns a list of those instances, as shown in Example 13-6.

Example 13-6 Fuzzy search returns all instances whose user_id field contains the search string

```
>>> admin.users["admin"]
[{
  "auth_mode": "internal",
  "bidi_national_calendar": None,
  "bidi_text_direction": None,
  "created_time": "Nov 17, 2014 2:35:39 PM",
  "current_status": None,
  "email": "test@user.com",
  "groups": (nested object),
  "id": "31fe4acf-9dff-4037-b050-ac8447187cdf",
  "name": "Default Admin",
  "password": (write-only),
  "roles": (nested object),
  "shellaccounts": (nested object),
  "url": "/admin/resources/users/31fe4acf-9dff-4037-b050-ac8447187cdf",
  "user_id": "admin"
}, {
  "auth_mode": "internal",
  "bidi_national_calendar": None,
  "bidi_text_direction": None,
  "created_time": "Nov 21, 2014 7:41:27 PM",
  "current_status": None,
```

```
  "email": None,
  "groups": (nested object),
  "id": "5ba0e8d4-4150-4e76-8ae3-6d7df0907424",
  "name": None,
  "password": (write-only),
  "roles": (nested object),
  "shellaccounts": (nested object),
  "url": "/admin/resources/users/5ba0e8d4-4150-4e76-8ae3-6d7df0907424",
  "user_id": "labadmin"
}]
```

To list the available functions and properties of a certain collection or object, you can call **help()**, shown in Example 13-7.

Example 13-7 The help() function displays a brief definition, methods, and attributes available to an object

```
>>> help(admin.users)
A Users object represents the collection of users defined to the
system.  Objects of this type are used to iterate over,
list and search for users in the system.

Additional help is available for the following methods:
   admin, __contains__, create, delete, __delitem__, __getattr__,
   __getitem__, __iter__, __len__, list, __lshift__, __repr__, __rshift__,
   self, __str__, __unicode__
```

13.1.2 Creating and deleting objects

Classes in the admin and deployer packages implement the **create()** and **delete()** functions. To learn more about the usage of class-bound functions, call **help()** on that class. Example 13-8 shows the creation of a Windows Server 2012 R2 virtual image through a local OVA import. You can create instances of many types in this same fashion.

Example 13-8 The create() function can be called to create an object instance

```
>>> deployer.virtualimages.create('/jumper/IPAS-OVAs/Windows2012R2_maestro.ova')
{
  "acl": (nested object),
  "advancedoptionsaccepted": "F",
  "build": "",
  "created": Jul 14, 2015 1:49:25 PM,
  "currentmessage": None,
  "currentmessage_text": None,
  "currentstatus": "RM01036",
  "currentstatus_text": "Queued",
  "description": "",
  "hardware": None,
  "id": 89,
  "license": (nested object),
  "licenseaccepted": "F",
  "metaSignature":
"30E39FA8ACDBF2FAD8C333161D704A6394FB04D0843530E25F34EFB02B6938182DF2C60B7E9BE3D9F
616EFB7B6D4DF12C60D4CEA9F869E78F2E32F0E6D546593",
  "name": "New virtual image 1436896165218-497",
```

```
  "operatingsystemdescription": None,
  "operatingsystemid": 0,
  "operatingsystemversion": None,
  "owner": (nested object),
  "parts": (nested object),
  "pmtype": "Unknown",
  "productids": (nested object),
  "servicelevel": "0",
  "signature":
"008D2E07A57C211AA4BFE9A017D3A6BDDAC0539ED6E716A77D0771242D7BF66CBF95229AEDDA6ACD3
B026C60559E9B4E7C8238FF3E2BB9E80185AFDB86C14C4B",
  "updated": Jul 14, 2015 1:49:25 PM,
  "version": "0"
}
>>>
```

This create() call in Example 13-8 on page 359 returns the new virtual image instance in JSON format. You can check the status of the import with a *while loop* that is similar to the sample that is shown in Example 13-9. This loop checks the status of the import job every five seconds and prints to the console. The loop exits if the job fails or finishes.

Example 13-9 Sample while loop

```
>>> import time
>>> inst = deployer.virtualimages.list({'id':89})[0]
>>> while True:
...  print ('%s Importing... | Status: %s'
%(time.strftime("%c"),inst.currentstatus_text))
...  if inst.currentstatus_text == 'Failed':
...    print 'Import has failed. Please check the Jobs Queue for error logs.'
...    break
...  elif inst.currentstatus_text == 'Available':
...    print 'Finished'
...    break
...  time.sleep(5)
...
Tue Jul 14 14:23:29 2015 Importing... | Status: Queued
Tue Jul 14 14:23:37 2015 Importing... | Status: Processing
Tue Jul 14 14:23:45 2015 Importing... | Status: Processing
Tue Jul 14 14:23:52 2015 Importing... | Status: Processing
...
```

You can change the sleep time (in seconds) to fit your needs. Complementary to the **create()** function, you can also delete the image that is created in Example 13-8 on page 359 by using the **delete()** function. Example 13-10 shows the expected behavior and return of this function.

Example 13-10 Use delete() to delete an object instance

```
>>> deployer.virtualimages.delete(89)
>>> deployer.virtualimages.list({'id':89})
[]
>>>
```

The **delete()** function takes a resource instance ID as its argument (89, in this case).

13.1.3 Listing objects in a collection

You can list all the objects in a collection, and list a subset of the collection or an instance by index. For the usage of the **`list()`** function and its expected return, see Example 13-11.

Example 13-11 Both deployer.users and deployer.users.list() return a list of all users in the system

```
>>> len(deployer.users)
23
>>> len(deployer.users.list())
23
>>> type(deployer.users)
<class 'deployer.resources.user.Users'>
>>> type(deployer.users.list())
<type 'list'>
>>> deployer.users
...
<abbreviated>
...
>>> deployer.users.list()
...
<abbreviated>
...
>>>
```

As mentioned in 13.1.1, “Pure.cli basics” on page 356, both `deployer.users` and `deployer.users.list()` return a collection of users. However, the latter returns a Python list object. You can use the Python list to use native Python list functions and constructs, such as slice and list comprehension.

13.1.4 Iterating through a collection

Iterating through a PureApplication System resource collection is simple. Standard Python flow control statements are available:

- If-elif-else
- For
- While

Example 13-12 shows a few samples of a `for` statement and nesting an `if` statement within a `for` loop.

Example 13-12 Sample usage of flow control statements

```
>>> for inst in deployer.virtualapplications:
...  print '%s | Status: %s' %(inst.deployment_name,inst.status)
...
Hadoop by Chef - Zero Config | Status: STOPPED
MJM CustomOS | Status: RUNNING
Spark on Chef - Single Slave | Status: STOPPED
UrbanCode Deploy with Patterns | Status: RUNNING
2 VMs + 1 Script - On 1 | Status: RUNNING
Hadoop Cluster - Regular Config | Status: RUNNING
2 VMs + 1 Script - On Demand | Status: RUNNING
WordPress - Medium Rare | Status: STOPPED
5 VMs + 5 Scripts | Status: RUNNING
```

```
Spark on Chef - Run List | Status: STOPPED
MJM CustomOS2 | Status: RUNNING
Node.js by Chef - Zero Config | Status: STOPPED
5 VMs + 5 Scripts - 2 scale 5 | Status: RUNNING
Red Hat Satellite Server 5.6 Pattern | Status: RUNNING
Standalone Spark on Chef - New Repo | Status: LAUNCHING
Red Hat Satellite Service | Status: RUNNING
System Monitoring | Status: RUNNING
External Chef Server | Status: RUNNING
>>> for inst in deployer.virtualapplications:
...  if inst.status == 'RUNNING':
...   print '%s | Status: %s' %(inst.deployment_name,inst.status)
...
MJM CustomOS | Status: RUNNING
UrbanCode Deploy with Patterns | Status: RUNNING
2 VMs + 1 Script - On 1 | Status: RUNNING
Hadoop Cluster - Regular Config | Status: RUNNING
2 VMs + 1 Script - On Demand | Status: RUNNING
5 VMs + 5 Scripts | Status: RUNNING
MJM CustomOS2 | Status: RUNNING
5 VMs + 5 Scripts - 2 scale 5 | Status: RUNNING
Red Hat Satellite Server 5.6 Pattern | Status: RUNNING
Red Hat Satellite Service | Status: RUNNING
System Monitoring | Status: RUNNING
External Chef Server | Status: RUNNING
>>>
```

Example 13-12 on page 361 uses a standard for the statement to iterate through all instances of the `deployer.virtualapplications` collection. You can also use the **`deployer.utils.findAll()`** function along with the Python **`lambda`** function, as shown in Example 13-13.

Example 13-13 deployer.utils.findAll() and the lambda function alternative to the for statement

```
>>> from deployer import utils
>>> print('\n'.join([x.deployment_name +' | Status: '+ x.status for x in
utils.findAll(lambda i: i.status == 'RUNNING', deployer.virtualapplications)]))
Base OS Instance | Status:  RUNNING
MJM CustomOS | Status:  RUNNING
UrbanCode Deploy with Patterns | Status:  RUNNING
2 VMs + 1 Script - On 1 | Status:  RUNNING
Hadoop Cluster - Regular Config | Status:  RUNNING
2 VMs + 1 Script - On Demand | Status:  RUNNING
5 VMs + 5 Scripts | Status:  RUNNING
Script Test | Status:  RUNNING
MJM CustomOS2 | Status:  RUNNING
5 VMs + 5 Scripts - 2 scale 5 | Status:  RUNNING
Red Hat Satellite Server 5.6 Pattern | Status:  RUNNING
System Monitoring | Status:  RUNNING
External Chef Server | Status:  RUNNING
>>>
```

13.1.5 Finding an object instance by exact name

Because of fuzzy logic, when you search for an object by its key by using the **list({<key>:<value>})** function, you might get more than one object instance whose substring matches your search string. To get around this issue, you may construct a method such as one that is shown in Example 13-14. The **getExactObjectByName()** function takes the key, the value, and the collection, and returns the instance whose key:value pair matches the provided parameters exactly.

Example 13-14 getExactObjectByName() function

```
def getExactObjectByName(name, arrayOfObjects, propToLookFor):
    """
    Gets a specific object by name.

    Parameters:
    name (String): Name to match
    arrayOfObjects (Array): Object to compare to.
    propToLookFor (String): Property to match against (i.e. username, name)

    Returns:
    result (JSON Object): Object matching the exact name.
    """
    match = None

    # Get number of letters in the name.
    nameLen = len(name)

    # Loop through the array of objects.
    for i in arrayOfObjects:

        # Construct and evaluate the command with the specified property.
        cmdToEval = 'i.' + propToLookFor
        prop = eval(cmdToEval)

       # If the name is a substring of the matched object, and starts at index 0,
then, check its length.
        # If the length is the same than the original, it is an exact match.
        if prop.find(name) == 0:
            if len(prop) == nameLen:
                match = i
                break
    return match
```

13.1.6 Setting or updating an object attribute

You can use standard Python operators to set, update, or append values to a variable or attribute. Table 13-1shows some commonly used operators.

Table 13-1 Operators that are commonly used to update object attributes

Operator	Purpose	Example
<<	Left bitwise shift Used to add elements to certain nested attributes	`>>> admin.groups[3].roles << saved_roles`
>>	Right bitwise shift Used to subtract elements from certain nested attributes	`>>> admin.groups[3].roles >> saved_roles`
=	Assignment	`>>> admin.groups[3].description = 'Hello World'`
+=	Append	`>>> admin.groups[3].users += admin.users[1]`

Example 13-15 demonstrates how the bitwise shift operators can be used to set and remove roles to and from a user group.

Example 13-15 Bitwise shift operators

```
>>> admin.groups[3]
{
  "created_time": "Jul 7, 2015 1:11:59 PM",
  "description": None,
  "id": "c9a2f051-d827-4a64-8e77-3e2ef93cbff5",
  "member_location": "internal",
  "name": "backup_users",
  "roles": (nested object),
  "updated_time": "Jul 7, 2015 1:12:41 PM",
  "url": "/admin/resources/user_groups/c9a2f051-d827-4a64-8e77-3e2ef93cbff5",
  "users": (nested object)
}
>>> ALL_ROLES = ['CLOUD_ADMIN', 'APPLIANCE_ADMIN', 'CATALOG_CREATOR',
'PATTERN_CREATOR', 'ILMT_USER','CLOUD_ADMIN_READONLY', 'APPLIANCE_ADMIN_READONLY',
'PROFILE_CREATOR', 'REPORT_READONLY', 'AUDIT_READONLY', 'AUDIT', 'HARDWARE_ADMIN',
'SECURITY_ADMIN', 'SECURITY_ADMIN_READONLY', 'ROLE_ADMIN',
'HARDWARE_ADMIN_READONLY', 'CLOUDGROUP_ADMIN', 'CLOUDGROUP_ADMIN_READONLY',
'USER_ADMIN_READONLY', 'DR_ADMIN', 'DR_ADMIN_READONLY', 'CLOUDGROUP_MANAGER',
'HARDWARE_MANAGER']
>>> admin.groups[3].roles
['CLOUD_ADMIN', 'APPLIANCE_ADMIN', 'CATALOG_CREATOR', 'PATTERN_CREATOR',
'ILMT_USER', 'CLOUD_ADMIN_READONLY', 'APPLIANCE_ADMIN_READONLY',
'PROFILE_CREATOR', 'REPORT_READONLY', 'HARDWARE_ADMIN', 'HARDWARE_ADMIN_READONLY',
'CLOUDGROUP_ADMIN', 'CLOUDGROUP_ADMIN_READONLY', 'USER_ADMIN_READONLY',
'CLOUDGROUP_MANAGER', 'HARDWARE_MANAGER']
>>> saved_roles = ['CLOUD_ADMIN', 'APPLIANCE_ADMIN', 'CATALOG_CREATOR',
'PATTERN_CREATOR', 'ILMT_USER', 'CLOUD_ADMIN_READONLY',
'APPLIANCE_ADMIN_READONLY', 'PROFILE_CREATOR', 'REPORT_READONLY',
'HARDWARE_ADMIN', 'SECURITY_ADMIN', 'SECURITY_ADMIN_READONLY',
```

```
'HARDWARE_ADMIN_READONLY', 'CLOUDGROUP_ADMIN', 'CLOUDGROUP_ADMIN_READONLY',
'USER_ADMIN_READONLY', 'CLOUDGROUP_MANAGER', 'HARDWARE_MANAGER'] #saves roles for
later restoring
>>> admin.groups[3].roles >> ALL_ROLES #removes all roles from the group
>>> admin.groups[3].roles
[]
>>> admin.groups[3].roles << saved_roles #re-adds saved roles
>>> admin.groups[3].roles
['CLOUD_ADMIN', 'APPLIANCE_ADMIN', 'CATALOG_CREATOR', 'PATTERN_CREATOR',
'ILMT_USER', 'CLOUD_ADMIN_READONLY', 'APPLIANCE_ADMIN_READONLY',
'PROFILE_CREATOR', 'REPORT_READONLY', 'HARDWARE_ADMIN', 'HARDWARE_ADMIN_READONLY',
'CLOUDGROUP_ADMIN', 'CLOUDGROUP_ADMIN_READONLY', 'USER_ADMIN_READONLY',
'CLOUDGROUP_MANAGER', 'HARDWARE_MANAGER']
```

13.1.7 Building object collections and instances from REST JSON responses

There are some rare instances where an object attribute exists in a REST resource but not a CLI resource. If you must use these attributes to form a relationship (such as finding Virtual System Classic virtual machines (VMs) that are on certain Cloud Group; the Classic VMs and Cloud Group relationship does not exist in the CLI), you must pass in these REST resources. You can use the built-in helpers `deployer.http` and `deployer.prettify` to get direct JSONs from HTTP REST responses. From there, the JSON can be casted to whatever collection you prefer or to a Python list.

Example 13-16 demonstrates a simple function that runs an HTTP GET and saves the response to a list that is usable by the CLI. You can pass in the URI to make the function more dynamic.

Example 13-16 getThings() function turns a REST JSON response into a list that is usable by the CLI

```
from deployer import http, prettify
def getThings():
        url ="<REST-URI>"
        json = http.get(uri)
        result = prettify.PrettyList(json)
        return result
```

You can use the `deployer.http` helper to run the standard HTTP verbs (GET, PUT, POST, and DELETE), and the `deployer.prettify` object parses in and formats the output in a usable form to the CLI.

13.2 PureApplication System REST API

The PureApplication System REST API is a RESTful web service that exposes various PureApplication System components and resources. It is implemented over HTTP, so by using standard HTTP methods (**GET**, **POST**, **PUT**, and **DELETE**), users can manage these system components by using automation through the Jython-like pure.cli scripting environment or any other programming language that supports HTTP.

The PureApplication System is REST driven. Every click event on the UI translates to one or more HTTP REST requests. Understanding the REST API allows users to manage various components through automations by using any programming languages that support HTTP verbs.

13.2.1 Anatomy of a RESTful HTTP request

A RESTful HTTP request is composed of the following items:

- A verb or method (**GET**, **PUT**, **POST**, or **DELETE**)
- A resource URI (`/resources/instances`)
- Request headers
- Message body ({“id”:”xyz”}

In programming semantics, the four HTTP verbs are synonymous with functions or methods. The resource URI is the target object on which the method is called. The request headers are the parameters. The message body represents the instructions. Depending on the method that is called, the message body is used in different ways.

Figure 13-1 shows an HTTP **GET** call to an instance of the resource `virtual Applications`. This resource corresponds to a Virtual System instance (`vSys.Next`) or a Virtual Application instance on the PureApplication System web UI. The message Body tab is not available because **GET** methods do not have bodies.

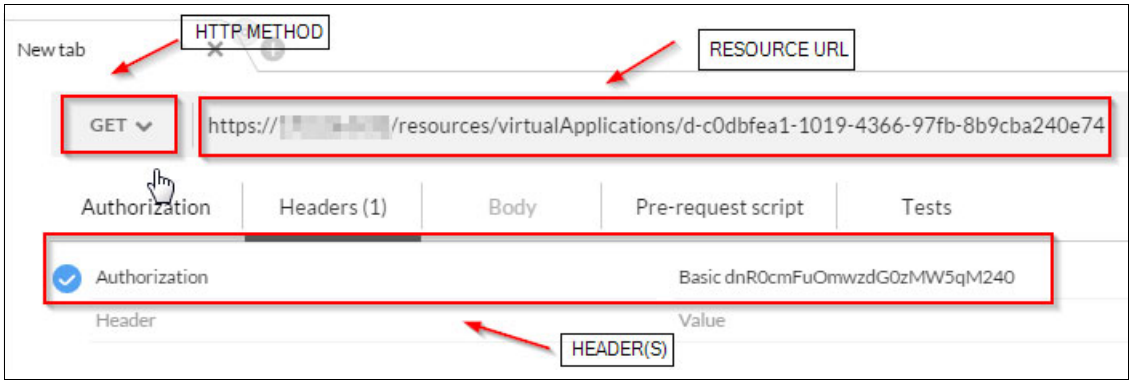


Figure 13-1 Elements of an HTTP GET request

Take a closer look at what is being sent to the server by examining the raw HTTP request in Figure 13-2 on page 367.

```
GET /resources/virtualApplications/d-c0dbfea1-1019-4366-97fb-8b9cba240e74
HTTP/1.1
Host: 172.26.0.32:443
Accept: */*
Accept-Encoding: gzip, deflate, sdch
Accept-Language: en-US,en;q=0.8
Authorization: Basic dnR0cmFuOmwzdGOzMW5qM240
Cache-Control: no-cache
Cookie:
SimpleToken=GIsFD8TNbvoKJnmfI4gqcr9BLJUy1Zj7UIsBUjJ8uW3EQmzan9OZ9XKFBJxCYJJF4RV
xRxHcTZ7B4we5kOAZmpjLvDtp1RcXKz7D3q/Qvb74eFXz5YOEi5viSKo7UIRowQarx1AtzNfraBA6a2
D1oiKtWIGyWfY5AI++LDZXxysEt4BFQcEUv8UCYE4lxbOV;
zsessionid=1436542260232b2ff272e3eb153fe534aa8cc9180ad201cefebf5c83fb9b
CSP: active
Postman-Token: c3145afc-35f3-b972-e9d3-5dae1518ba16
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/43.0.2357.132 Safari/537.36

HTTP/1.1 200 OK
Cache-Control: no-cache
Cache-Control: private
Cache-Control: must-revalidate
Cache-Control: max-age=0
Connection: Keep-Alive
Content-Encoding: gzip
Content-Type: application/json
Date: Fri, 10 Jul 2015 15:41:18 GMT
Keep-Alive: timeout=70, max=500
Transfer-Encoding: chunked
Vary: User-Agent,Accept-Encoding
```

Figure 13-2 Basic headers of a REST HTTP request

Here is a breakdown of the essential fields of an HTTP request:

- `GET /resources/virtualApplications/d-c0dbfea1-1019-4366-97fb-8b9cba240e74 HTTP/1.1`
  - First, the client defines the HTTP Method, Resource URI, and HTTP version.
  - The HTTP version is often auto-negotiated by the client and server.
  - You as a user should be responsible only for defining the METHOD and the Resource URI.
- `Host: 172.26.0.32:443`
  - This is technically a request header field. However, it is derived from the resource URL, so you do not have to explicitly set it as a header.
  - The destination and protocol port of the request.
  - The user must define this as part of the Resource URL, as shown in Figure 13-1 on page 366.
  - Together with the protocol designation (port 443 for HTTPS) and the URI, this field forms the Resource URL.

- `Accept: */*`
  - This is a request header field.
  - This field tells the server which media type is acceptable as reply.
  - This particular HTTP Request accepts all media types as reply.
- `Accept-Encoding: gzip, deflate, sdch`

  This header tells the server which encoding is acceptable.
- `Accept-Language: en-US,en;q=0.8`

  This header tells the server which language is acceptable. PureApplication System REST API supports the same languages and locales that are available on the UI and CLI.
- `Authorization: Basic dnROcmFuOmwzdGOzMW5qM240`

  In basic user name and password authentication scheme, "`dnROcmFuOmwzdGOzMW5qM240`" is a Base 64 encoding of your user name and password string that is separated by a colon (such as Username: Coffee; Password: milkandsugar; the final encoding comes from the result of BASE64ENC(Coffee:milkandsugar).

PureApplication System supports only the basic authentication method. After the server receives the first basic authentication header, it provides the client with a `zsessionid` header and a SimpleToken header. Any subsequent request by the client should include these two headers to avoid the need for reauthentication.

13.2.2 HTTP verbs

This section describes the four HTTP methods that the PureApplication System REST API implements.

GET

This method is called every time that your browser fetches the content of a webpage through HTTP or HTTPS. **`GET`** is considered a safe method because it is read-only. No changes are made to the resources on the server. **`GET`** calls do not have message bodies. This method can be submitted against a resource instance or a resource collection:

- **`GET /resources/virtualApplications/`**

  This method returns a JSON list of `virtualApplication` instances.
- **`GET /resources/virtualApplications/d-c0dbfea1-1019-4366-97fb-8b9cba240e74`**

  This method returns the JSON of the `virtualApplications` instance.

PUT

This method updates the resource instance's fields with the message body, as shown in Example 13-17. It is submitted against the resource instances and not the collection:

```
PUT /resources/virtualApplications/d-c0dbfea1-1019-4366-97fb-8b9cba240e74
```

Example 13-17 The message body of the PUT request

```
{
   "can_resume" : "true"
}
```

This method updates the field **“**`can_resume`**”** of the virtual system instance with the deployment ID above to **“**`true`**.**” This attribute corresponds to the **Resume** icon on the Virtual System Instances page on the UI.

If the resource instance does not exist, the **`PUT`** call creates an instance by using the fields in the message body. The **`PUT`** call requires a message body. Even though you can **`PUT`** an empty message body, the server’s implementation of the resource determines whether to return a status code `204 No Content` or an Internal Error with status code `400 Bad Request`. This is an idempotent method.

POST

This method creates a resource instance. It is submitted to the resource collection, indicating that a new instance of the same resource type should be created. A **`POST`** call generally has a message body (Example 13-18).

```
POST
/resources/applicationPatterns/a-d3a4c9c1-d402-4740-94f6-fe1b41babd4d/virtualAppli
cations/
```

Example 13-18 Message body of the POST request

```
{  "deployment_name": "Base OS",
   "model": {
      "model": {
         "description": "",
         "nodes": [{
            "id": "OS Node",
            "attributes": {
               "ConfigPWD_USER.password": "<xor>Lz4sLChvLTs=",
               "HWAttributes.memsize": 3072,
               "HWAttributes.numvcpus": 1,
               "ConfigPWD_ROOT.password": "<xor>Lz4sLChvLTs="
            },
            "type": "image:OS Node:IBM OS Image for AIX Systems:2.1.1.0:13",
            "locked": []
         },
         {
            "id": "Default add user",
            "attributes": {
               "USERNAME": "virtuser",
               "PASSWORD": "<xor>PToxMzYx"
            },
            "type": "add-on:Default add user:1.0.0",
            "locked": []
         }],
         "app_type": "application",
         "name": "Base OS",
         "patterntype": "vsys",
         "links": [{
            "id": "HostedOnLink_1",
            "source": "Default add user",
            "target": "OS Node",
            "attributes": {

            },
            "type": "HostedOnLink"
```

```
            }],
            "locked": false,
            "readonly": false,
            "version": "1.0",
            "patternversion": "1.0",
            "mixinArgs": null
         },
         "layers": [{
            "id": "layer",
            "nodes": ["OS Node",
            "Default add user"]
         }]
      },
      "cloud_group": "1",
      "ip_group": "2",
      "environment_profile_id": "1",
      "priority": "high",
      "placement_only": true
}
```

DELETE

This method deletes the resource instance or resource collection. It can be submitted against a collection or an instance. A **DELETE** call does not have a message body.

```
DELETE /resources/virtualApplications/d-c0dbfea1-1019-4366-97fb-8b9cba240e74
```

13.3 CLI recipes

This section includes some example scripts that can be used *as is* or modified to fit your requirements. Testing and environmental differences should always be considered.

13.3.1 Importing and loading group fixes

You can use the script in Example 13-19 to load group fixes in the form of GZIP Compressed Tar Archive files (tgz). You can either hardcode the variable *mypath* or pass it in as an argument by using *sys.argv*. This path is the OS path to the directory that contains all of your pattern types. You can download the entitled pattern types as part of a release's group fix from Fix Central. For more information about entitled fixes, see the following website:

http://www.ibm.com/support/docview.wss?uid=swg27039159

Example 13-19 Import pattern type tgz files if the pattern type is not already installed on the rack

```
from os import listdir
from os.path import isfile, join
import deployer, sys, re, datetime

mypath = '/jumper/files/dd-intel4RB/ptypes/'
#onlyfiles = [ join(mypath,f) for f in listdir(mypath) if isfile(join(mypath,f)) ]
installed = set()

'''
Get list of installed ptypes
'''
for pi in deployer.plugins:
    installed.add(pi.patterntype_package)
insList = list(installed)

'''
```

```
Return true if ptype exists on rack
Relies on tgz naming convention and patterntype_package name
'''
def exist(tgzname):
    if re.sub('.tgz','',re.sub('-','/',tgzname)) in insList:
        return True
    else:
        return False

toUpdate = []               # to store tgz not yet exist on rack and to be updated
toUpdatePath = []           # to store path to those tgz
for f in listdir(mypath):
    if isfile(join(mypath,f)) and not exist(f):
        toUpdatePath.append(join(mypath,f))
        toUpdate.append(f)

cr = datetime.datetime.now().strftime('%Y/%m/%d %H:%M:%S')
print '\n'
print 'Script started %s' %(cr)
print 'ptypes to be updated: \n'
print '++++++++++++++++++++++++'
for thing in toUpdate:
    print thing
print '++++++++++++++++++++++++'
print '\n'
print 'ptypes update started...'
for pt in toUpdatePath:
    k = re.compile("%s(.*)" %(mypath)).search(pt)
    name = k.group(1)
    try :
        print "uploading %s" %(name)
        t = deployer.patterntypes.create(pt)
        t[0].acceptLicense()
        t[0].enable()
    except :
        excType = sys.exc_info()[0]
        excValue= sys.exc_info()[1]
        if "zero.json.JsonParserException: Decoding java.io.InputStreamReader" in str(excValue):
            p = re.compile("%s(.*)" %(mypath)).search(pt)
            print 'ptype %s is probably already loaded (with no plugins),  please check the UI' %(p.group(1))
        else:
            print "Caught an exception of type" + str(excType) + ":" + str(excValue)
        #end try
#endFor
end = datetime.datetime.now().strftime('%Y/%m/%d %H:%M:%S')
print 'ptypes update completes at %s' %(end)
print '\n'
```

13.3.2 Creating users and groups from a CSV list

By using the script in Example 13-20, you can pass in a CSV file of users and create a user group, and add users to an existing user group. Here are the CSV fields:

- First name
- Last name
- userid
- email

This script is useful if there is a large list of users that must be added to one or more racks.

Example 13-20 Take a CSV list of users and add them to user-specified groups

```
from deployer import utils
import csv, admin

userList = []

csvFile = open('Users.csv', 'rb')


try:
    spamReader = csv.reader(csvFile)
    for row in spamReader:
        print row
        userList.append(row)
finally:
```

```
    csvFile.close()
    # print userList
GROUP_NAME = 'GROUP1'

ALL_ROLES = ['CLOUD_ADMIN', 'APPLIANCE_ADMIN', 'CATALOG_CREATOR', 'PATTERN_CREATOR', 'ILMT_USER',
'CLOUD_ADMIN_READONLY', 'APPLIANCE_ADMIN_READONLY', 'PROFILE_CREATOR', 'REPORT_READONLY',
'AUDIT_READONLY', 'AUDIT', 'HARDWARE_ADMIN', 'SECURITY_ADMIN', 'SECURITY_ADMIN_READONLY',
'ROLE_ADMIN', 'HARDWARE_ADMIN_READONLY', 'CLOUDGROUP_ADMIN', 'CLOUDGROUP_ADMIN_READONLY',
'USER_ADMIN_READONLY', 'DR_ADMIN', 'DR_ADMIN_READONLY', 'CLOUDGROUP_MANAGER', 'HARDWARE_MANAGER']

GUEST_ROLES = ['CLOUD_ADMIN', 'APPLIANCE_ADMIN', 'CATALOG_CREATOR', 'PATTERN_CREATOR',
'ILMT_USER', 'CLOUD_ADMIN_READONLY', 'APPLIANCE_ADMIN_READONLY', 'PROFILE_CREATOR',
'REPORT_READONLY', 'HARDWARE_ADMIN_READONLY', 'CLOUDGROUP_ADMIN', 'CLOUDGROUP_ADMIN_READONLY',
'CLOUDGROUP_MANAGER']

def addLDGroup(GROUP_NAME):
    if not utils.find(lambda r: r.name == GROUP_NAME, admin.groups[GROUP_NAME]):
        u = admin.groups << {
            "member_location": "ldap",
            "name": GROUP_NAME
        }
    admin.groups[GROUP_NAME][0].roles >> ALL_ROLES
    if GROUP_NAME == 'GROUP2':
        admin.groups[GROUP_NAME][0].roles << GUEST_ROLES
    else:
        admin.groups[GROUP_NAME][0].roles << ALL_ROLES


addLDGroup(GROUP_NAME)

# group = admin.groups[GROUP_NAME][0]# only need 1 group
for user in userList:
    if not utils.find(lambda r: r.user_id == user[2], admin.users[user[2]]):  # verify user does
not already exist
        u = admin.users << {  # add user to rack
                              "auth_mode": "ldap",
                              "email": user[3],
                              "name": user[0] + " " + user[1],
                              "user_id": user[2]
                              }
        print "User %s created" % (user[0],)
```

13.3.3 Cleaning old deployments (non-production environments only)

The script in Example 13-21 cleans all deployments older than 30 days (excluding shared services). You can change the variable *ttl* to change the time to live. To exempt deployments from being automatically cleaned, users can add the **"**DND**"** string to the deployment name (this safe string can be changed). This script should be used *only on non-production or test environment*. The purpose of this script is to clean or purge sufficiently old deployments to recover system resources. This script can be automated as a weekly **cron** job.

Example 13-21 Clean an old deployment that matches a certain time to live duration

```
from datetime import datetime, timedelta
import time, sys

#sys.path.append("C:\Users\Vincent\PycharmProjects\Dummy\2.1.0.0-201504081723317")
import deployer

ttl = timedelta(days=30)
timeout = 900  # 15 minutes
sleepInterval = 60  # 1 minute
deletionInterval = 60  # to avoid concurrency problem
allowedCheckTime = timeout / sleepInterval



def cleanApp():
    for instance in deployer.virtualapplications:
        if isAppPerm(instance):
            print instance.deployment_name.encode(
                        "UTF-8") +' will not be deleted. It has the "Do Not Delete" flag or is a Shared Service.'
#        elif instance.status != 'STARTED' or instance.status != 'ERROR':
#            print instance.deployment_name.encode("UTF-8")+ ' might be stopping or deleting'
        else:
            d1 = datetime.strptime(instance.start_time,"%Y-%m-%dT%H:%M:%S.")
```

```
            if datetime.now() - d1 > ttl:
                try:
                    instance.delete()
                    print '###### Deleting '+instance.deployment_name+' #####'
                    time.sleep(deletionInterval)
                except Exception, err:
                    print datetime.now().isoformat(
                        ' ') + " -- " + "Error deleting " + instance.deployment_name.encode(
                        "UTF-8") + " manual intervention required."
                    print "Exception: " + str(err)
            else:
                print instance.deployment_name.encode(
                        "UTF-8") +' will not be deleted. It is not old enough.'

def cleanSys():
    for instance in deployer.virtualsystems:
        if isSysPerm(instance):
            print instance.name.encode(
                        "UTF-8") +' will not be deleted. It has the "Do Not Delete" flag.'
        else:
            if datetime.now() - datetime.utcfromtimestamp(instance.updated) > ttl:
                try:
                    instance.delete()
                    print '##### Deleting '+instance.name + ' #####'
                    time.sleep(deletionInterval)
                except Exception, err:
                    print datetime.now().isoformat(
                        ' ') + " -- " + "Error deleting " + instance.name.encode(
                        "UTF-8") + " manual intervention required."
                    print "Exception: " + str(err)
            else:
                print instance.name.encode(
                        "UTF-8") +' will not be deleted. It is not old enough.'
def isAppPerm(app):
    if 'DND' in app.deployment_name or 'Tiebreaker' in app.deployment_name or app.app_type == 'service':
        return True
    return False


def isSysPerm(sys):
    if 'DND' in sys.name or 'Tiebreaker' in sys.name:
        return True
    return False


def clean():
    print '*****Cleaning vApp and vSys.Next (excluding Shared Services)*****'
    cleanApp()

    print '*****Cleaning vSys.Classic******'
    cleanSys()

clean()
```

13.3.4 Getting full instance reports while filtering by virtual machine name

The **getReport** function in Example 13-22 uses the `deployer.http` helper to send a **GET** HTTP request to the server and parse the response (in this case, the instance reports) to an output file. The function takes a string argument and a CSV file argument. It then generates a VM usage report to contain all VMs whose name contains the supplied search string. This script can be complementary to the Virtual Machine Usage Report function on the PureApplication System user interface (`/systemconsole/reports/machine-activity`). You can use this script to tailor the usage reports with more fine-grained search parameters.

Example 13-22 Generate a VM usage report for VMs whose name contains the string "ODR"

```
from deployer import http, utils

def getReport(somestring, somefile):
    x = admin.virtualmachines
    uri = '/admin/resources/instancereport?'
    iString= ''
```

```
    tag= 'instid='
    delim = '&'
    somefile = file(somefile, 'wb')
    docclose = True
    for thing in x:
        if somestring in thing.name:
            iString+=tag+thing.id+delim
    http.get(uri+iString+'csv=true',
responseHandler=utils.curryFunction(utils.getResponseHandler, somefile))
    if docclose:
        somefile.close()

getReport('ODR', 'test.csv')
```

13.3.5 Running a script package

The script in Example 13-23 takes a single argument (deployment ID). It then finds eligible VMs of the specified deployment and allows the user to select one. After the VM is selected, it finds the VM's eligible script packages (execution mode 2 and 3) and allows the user to run one. After the script is ran, a status indicator informs the user whether the script is `RUNNING`, `FAILED`, or `DONE`.

Example 13-23 Run a script package and output the status to stdout

```
from deployer import http, prettify
import time, datetime, sys

scriptName = None
myscript = None
myvm = None
mystatus = None
cmdarg = sys.argv

def echo_wait(string):
    ts = datetime.datetime.fromtimestamp(time.time()).strftime('%H:%M:%S')
    sys.stdout.write("\r\033[K[%s]: %s" % (ts, string))
    sys.stdout.flush()

def echo_log(string):
    ts = datetime.datetime.fromtimestamp(time.time()).strftime('%H:%M:%S')
    sys.stdout.write("[%s]: %s" % (ts, string))
    sys.stdout.flush()
    sys.stdout.write("\n")
    sys.stdout.flush()

if len(cmdarg) != 2:
    print "Usage:"
    print "  testSP.py deployment-ID "
    sys.exit(1)


depID = str(cmdarg[1])

if depID[0] != 'd' or len(depID)!=38:
    print "Invalid parameter: Deployment ID should start with the letter d"
    sys.exit(1)

depl = deployer.virtualapplications.list({'id':str(depID)})

if len(depl) == 1 :
    dep = depl[0]
```

```
    vms = dep.virtualmachines.list()
else:
    print 'The deployment ID does not exist, please verify'
    sys.exit(1)

print " VM(s) in Deployment: "
#print vms
validID = []
indx = 0
for v in vms:
    indx += 1
    validID.append(indx)
    print 'VM ID: %s | VM Name : %s' %(indx, v.name)
print " Please select the VM you'd like to run a  script package on"

while True:
    try:
        idt = int(input("VM ID: "))
    except ValueError:
        print("Sorry, I didn't understand that. Please re-enter VM ID.")
        continue

    if idt not in validID:
        print("Please enter a VM ID from the valid IDs above.")
        continue
    else:
       #we're ready to exit the loop.
        break
idt -=1
myvm = dep.virtualmachines.list()[idt]
#print myvm.name
validSP= []
print "Script package(s) in VM:"
for p in myvm.scripts:
#    print p['name']
    if p['execmode'] in [2,3]:
        validSP.append(p['id'])
        print 'Script ID: %s | Script Name: %s' %(p['id'], p['name'])
print " Please select the script package you'd like to execute"

while True:
    try:
        spid = int(input("Script ID: "))
    except ValueError:
        print("Sorry, I didn't understand that. Please re-enter Script ID.")
        continue

    if spid not in validSP:
        print("Please enter a Script ID from the valid IDs above.")
        continue
    else:
       #we're ready to exit the loop.
        break
myscript = (i for i in myvm.scripts if i['id'] == spid).next()
#print spid
#print(myscript['name'])

d = dict([('script',myscript)])
myvm.executeScript(d)

while not mystatus:
    for history in myvm.scripthistory:
        if int(history.get('scriptid')) == int(myscript.get('scriptId')):
            myrc = history.get('scriptstatus')[len(history.get('scriptstatus')) -
1].get('currentstatus')
            if myrc in ["RMO5010","RMO5009","RMO1012"]:
```

```
                result = "Executing Script: %s [DONE]" % myscript.get('label')
                echo_log("%s " % result)
                sys.stdout.write("\n")
                sys.stdout.flush()
                mystatus = True
            elif myrc in ["RM05006", "RM05007","RM05008","RM01013"]:
                result = "Executing Script: %s [FAILED]" % myscript.get('label')
                echo_log("%s " % result)
                sys.stdout.write("\n")
                sys.stdout.flush()
                mystatus = True
            else:
                result = "Executing Script: %s [RUNNING] %s" % (myscript.get('label'), myrc)
                echo_wait("%s " % result)
                time.sleep(1)
                echo_wait("%s ." % result)
                time.sleep(1)
                echo_wait("%s .." % result)
                time.sleep(1)
                echo_wait("%s ..." % result)
                time.sleep(1)
```

13.3.6 Finding VMs in a particular cloud group by using a particular image

This use case can be useful to administrators who must manage and report license usage on a particular environment. You can use the script in Example 13-24 to select interactively a cloud group and a virtual image search parameter. The script can be modified to search on an environment profile instead of cloud group. You can also add additional search parameters. This script takes no argument at run time. It interactively passes in search parameters from the user and finds VMs that use a certain virtual image and is deployed in a certain cloud group.

For more information, see the IBM Knowledge Center found at the following website:

http://www.ibm.com/support/knowledgecenter/SSCR9A_2.1.0/doc/iwd/cct_usingcli.dita

Example 13-24 Find VMs that use certain virtual image and are deployed in certain cloud group

```
import deployer, admin
from deployer import utils, http, prettify

mycloud = None
myimage = None
allVMs = []
allVMCs = []

'''
Store VMs from vsys.next/vapp instances to allVMs list
'''
for inst in deployer.virtualapplications:
    if inst.app_type != "service":
        for vinst in inst.virtualmachines:
            allVMs.append(vinst)

'''
Store VMs from vsys classic instances to allVMCs list
'''
for inst in deployer.virtualsystems:
    for vinst in inst.virtualmachines:
        allVMCs.append(vinst)


def search(cloudid, imageid):
    if cloudid == mycloud.id and imageid == myimage.id:
        return True
    else:
        return False


def getVM():
```

```
    res1 = utils.findAll(lambda vm: search(vm.environment['cloudid'], vm.virtualimage.id), allVMs)
    res2 = utils.findAll(lambda vm: search(getClassicCG(vm.virtualsystem.id), vm.virtualimage.id), allVMCs)
    res = res1 + res2
    return res

'''
Store virtual system classic instances store their cloud group ID differently
This is the http helper mentioned in section 13.1.6
'''
def getClassicCG(vsysid):
    json = http.get("/resources/instances/%s/virtualMachines" % (vsysid))
    result = prettify.PrettyList(json)
    return result[0]['cloudid']


print 'Usage: '
print '     findVM.py takes no arguments'
print '     Input your search parameters from prompt'
print '     This script will return a list of VMs that belongs to the Cloud Group and using a Virtual Image that you
specify'
print '\n'
print '========Cloud Group Selection======='
print '\n'
print 'Available Cloud Groups on this system:'
print '====================================='
validCG = []
for cg in deployer.clouds:
    validCG.append(cg.id)
    print 'ID: %s | Name: %s | Is External: %s' % (
    cg.id, cg.name, admin.clouds.list({'id': str(cg.description)})[0].is_public)
print '====================================='
print '\n'
print 'Please select a Cloud Group search parameter by entering its ID:'
while True:
    try:
        cid = int(input("Cloud Group ID: "))
    except ValueError:
        print("Sorry, I didn't understand that. Please re-enter Cloud Group ID.")
        continue

    if cid not in validCG:
        print("Please enter a Cloud Group ID from the valid IDs above.")
        continue
    else:
        # we're ready to exit the loop.
        break
mycloud = deployer.clouds.list({'id': cid})[0]
print '\n'
print 'You have selected Cloud Group %s' % (mycloud.name)
print '\n'
print '========Image Selection======='
print '\n '
print 'Available Images on this system:'
print '====================================='
validIMG = []
for img in deployer.virtualimages:
    validIMG.append(img.id)
    print 'ID: %s | Name: %s | Version: %s' % (img.id, img.name, img.version)
print '====================================='
print 'Please select an Image search parameter by entering its ID:'
while True:
    try:
        iid = int(input("Image ID: "))
    except ValueError:
        print("Sorry, I didn't understand that. Please re-enter Image ID.")
        continue

    if iid not in validIMG:
        print("Please enter a Image ID from the valid IDs above.")
        continue
    else:
        # we're ready to exit the loop.
        break
myimage = deployer.virtualimages.list({'id': iid})[0]
print '\n'
print 'You have selected Image %s %s' % (myimage.name, myimage.version)

matchedVMs = getVM()
if len(matchedVMs) > 0:
    print '===========Results============'
    for resvm in matchedVMs:
        print "VM Name: %s | Parent Deployment: %s | Deployment Status %s" % (
        resvm.name, resvm.virtualsystem.deployment_name, resvm.virtualsystem.status)
```

```
    print '============================='
elif len(matchedVMs) == 0:
    print 'No Match. Please run again and try different search Parameters'
else:
    print 'This should not happen.'
```

13.3.7 Creating IP groups, adding an IP range, and attaching an IP group to a cloud group by using a script

The script in Example 13-25 takes a CSV file and parses it in as a dict. It creates an IP group. For each IP group, it adds a defined IP range and attaches the IP group to a cloud group. Here are the expected headers for the CSV file:

- IP Group Name
- Subnet
- Mask
- Gateway
- VLAN ID
- DNS
- Secondary DNS
- IPversion
- Start IP
- End IP
- Cloud Group

You can change the key name in the script to fit your own CSV formatting.

Example 13-25 IPG2CY.py parses in IP group information from a CSV

```
import admin, csv, datetime, time
from deployer import utils


'''
Storing global set of rack IP
In case users need to check for existing IPs and split the range
'''
# ipset = set()
# for x in admin.ipgroups:
#     for y in x.ips:
#         ipset.add(y.ip)
#


# def exists(ip):
#     if ip in ipset:
#         return True
#     else:
#         return False



def createIPG(name, subnet, mask, gateway, vlan_id, dns, secondary_dns='', ip_version='ipv4'):
    ipg = admin.ipgroups << {
    "name": name,
    "subnet": subnet,
    "mask": mask,
    "gateway": gateway,
    "vlan_id": vlan_id,
    "dns": dns,
    "secondary_dns": secondary_dns,
    "ip_version": ip_version
    }


def populateIPG(name, from_ip, to_ip,):
    ipg = utils.find(lambda i: i.name == name, admin.ipgroups)
    ipg.ips.create(from_ip,to_ip)


def attach2CG(ipgname, cloud):
```

```
    ipg = utils.find(lambda i: i.name == ipgname, admin.ipgroups)
    cg = utils.find(lambda i: i.name == cloud, admin.clouds)
    ipg.cloud = cg


f = open('ipg2.csv', 'rb')
data = csv.DictReader(f)
for l in data:
    name = l['IP Group Name']
    subnet = l['Subnet']
    mask = l['Mask']
    gateway = l['Gateway']
    vlan_id = l['VLAN ID']
    dns = l['DNS']
    secondary_dns = l['Secondary DNS']
    #ip_version
    startip = l ['Start IP']
    stopip = l['End IP']
    cg = l['Cloud Group']
    createIPG(name,subnet,mask,gateway,vlan_id,dns,secondary_dns)
    print ('[%s]: IP Group %s is created') %(datetime.datetime.fromtimestamp(time.time()).strftime('%H:%M:%S'), name)
    populateIPG(name,startip,stopip)
    print ('[%s]: IP Range %s - %s is added') %(datetime.datetime.fromtimestamp(time.time()).strftime('%H:%M:%S'),
startip, stopip)
    attach2CG(name,cg)
    print ('[%s]: IP Group %s is attached to Cloud Group %s')
%(datetime.datetime.fromtimestamp(time.time()).strftime('%H:%M:%S'), name, cg)
f.close()
```

14

Chef

This chapter introduces Chef, which is a configuration management tool, and its integration with IBM PureApplication System. You can find more information about Chef, including its terminology, by going to the following website:

https://docs.chef.io/

This chapter covers the following topics:

- What is Chef
- Chef integration with PureApplication System
- Configuring the external Chef server shared service
- Sample Chef deployment
- Debugging a deployment by using a Chef client on PureApplication System

© Copyright IBM Corp. 2015. All rights reserved.

14.1 What is Chef

Chef is a configuration manager that lets users automate the build and configuration of servers. A standard Chef framework is composed of the following components:

- Chef workstation: The development node. The Chef workstation maintains the Chef repository, which holds the developmental copies of the cookbooks. Users can develop cookbooks and add them to the Chef local repository on a workstation, or synchronize the Chef repository with a version control system, such as Git. Alternatively, users also can download community maintained cookbooks. Cookbooks, roles, and data bags are uploaded to the Chef server from the workstation.
- Chef server: The centerpiece of the Chef framework. The Chef server stores all configuration data (such as cookbooks, roles, and data bags) and makes them available to the Chef client that is hosted on the nodes.
- Chef client: A configuration agent that runs on the nodes. The Chef client is installed at deployment time and is used to configure the node per the specified cookbook definition that is retrieved from the Chef server. After the initial configuration, the Chef client maintains communication with the Chef server. If the Chef client receives cookbook configuration changes (such as a cookbook update) from the Chef server, it reconfigures the node per the new cookbook definition.
- Node: The server that is configured and maintained by the Chef client by using a cookbook definition that is retrieved from the Chef server.

14.2 Chef integration with PureApplication System

PureApplication System supports the Chef framework. On PureApplication System, a Chef server shared service can be deployed to a cloud group. This shared service acts as an abstract pointer to an organization entity on a Chef server. This action creates a separation of concerns between the PureApplication System hardware and resource administrator, pattern deployer, the pattern developer, and the Chef administrator.

- The PureApplication System hardware and resource administrator continues to manage PureApplication resources without exposure to the Chef framework.
- The pattern developer is responsible for defining and testing the nodes, pattern parameters, chef roles, and recipe at deployment time.
- The pattern deployer's role remains unchanged. The pattern deployer can redefine deployment-time parameters and recipes of a pattern, or deploy the pattern as defined by the pattern developer.
- The Chef administrator is responsible for developing and maintaining the cookbook from the Chef workstation, and for indirectly managing the workload by updating the cookbook repository on the Chef server.

14.3 Configuring the external Chef server shared service

To begin integrating an existing Chef infrastructure with PureApplication System, configure and deploy an external Chef server shared service. You must meet the following prerequisites:

- The Chef workstation must be installed and configured by following the instructions that are found at the following website:

  https://docs.chef.io/install_dk.html

- The Chef server must be installed and configured by following the instructions that are found at the following website:

  https://docs.chef.io/install_server.html

To deploy an external Chef server shared service, the following items are required:

- A Chef server HTTPS URL, with a path to the organization, for example:

  https://chefdev.rtp.raleigh.ibm.com/organization/ipas.

- An organization validation key, which can be found in the organization's Starter Kit that is generated from the Chef Management UI after the compressed file is downloaded and unpacked. The validation key file is in the following directory:

  chef-repo/.chef/<organization-name>-validator.pem.

- A Chef server SSL certificate, which is in the following directory:

  /var/opt/opscode/nginx/ca/<chef_server_fqns>.crt on the Chef server.

The following fields are optional:

- Proxy URL for the Chef client (optional): Needed if the Chef client on the workload nodes cannot reach external resources, such as the Chef server.
- No proxy for (optional): A comma-separated list of servers for which the Chef client does not use the proxy (for example, resources on the same network).
- Gem repository URL for Chef recipes (optional): A URL pointing to the Gem repository (the default is http://rubygems.org/).
- Proxy URL for Gem repository (optional): The proxy URL that is used exclusively to access the Gem repository.

Before your first Chef deployment, you must load an appropriate package to the Chef client plug-in configuration window from your local machine by completing the following steps:

1. Click **Catalog** → **System Plug-ins**.
2. Select the **chefclient/x.x.x.x** plug-in from the pattern type **Chef Pattern type x.x.x.x**.
3. Click **Configure** and upload the appropriate package for your operating system (OS).

14.4 Sample Chef deployment

You can add the Chef client to the nodes of any virtual system pattern (VSP). Whether you have an existing Chef infrastructure and configuration assets or not, integrating Chef with PureApplication System is easy. By using the Chef client pattern type, PureApplication System users can install the Chef client on any nodes that must receive periodic configuration updates or nodes that can benefit from automated configuration.

Complete the following steps:

1. Click **Virtual System patterns** and select or clone one of your existing patterns.

   For clarity, Figure 14-1 demonstrates a simple pattern. It is composed of a Base OS node named *workstation*. Drag the Chef client from the Software component palette. The Debug component is useful because you can use it to resume on error during deployment, if needed.

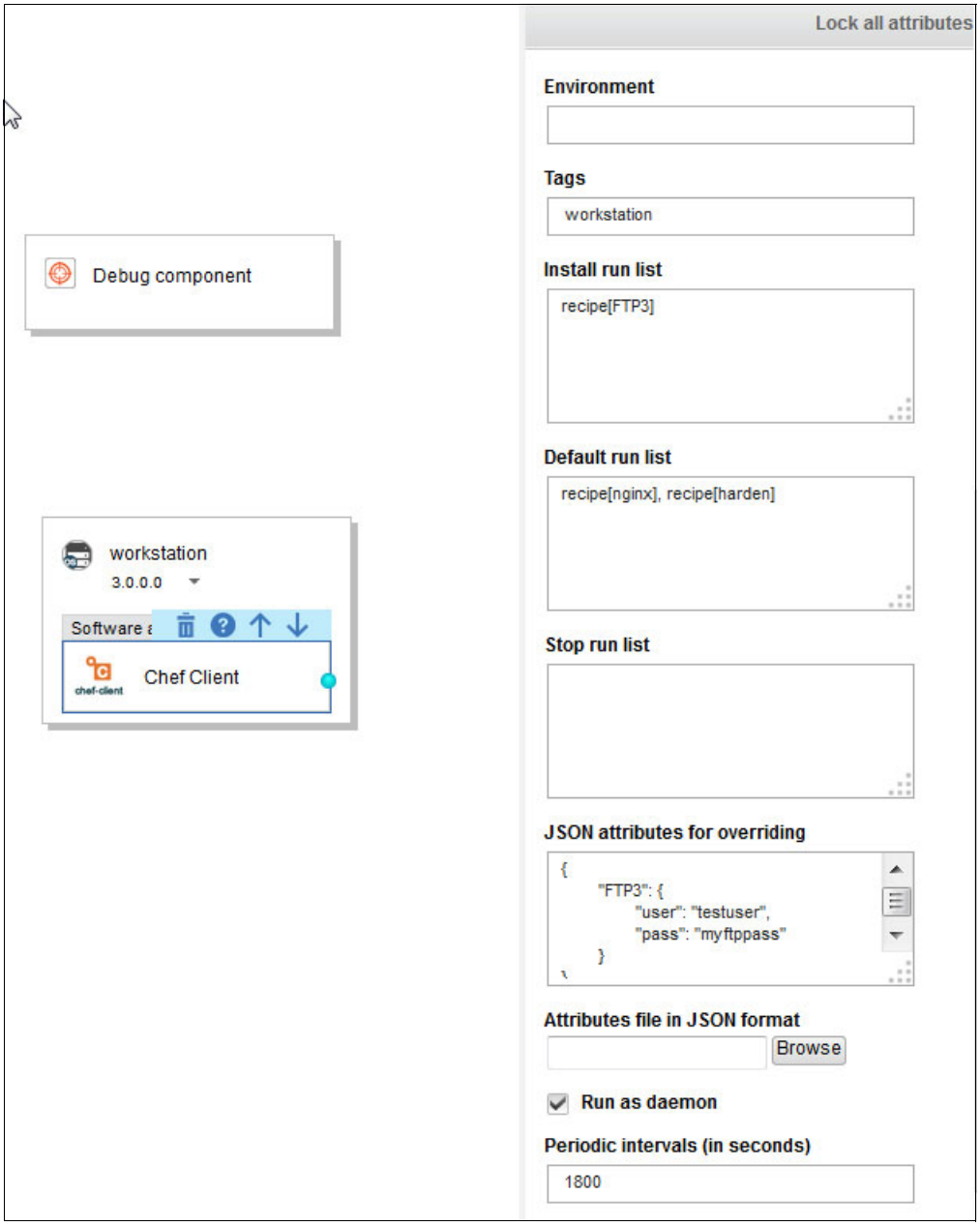


Figure 14-1 Example deployment with the Chef client

Figure 14-1 shows the following terms:

- Environment

  In this field, you can specify the Chef environment and the constraints to which the node is to adhere. A blank Environment field places the node in the Chef's `_default` environment. A node takes only one environment.

- Tags

  This field specifies the tags to be associated with the node. These tags are custom descriptions that the Chef administrator can use to find, group, and manage nodes easily. The example in Figure 14-1 on page 384 has a single tag, but the field can take multiple tags that are separated by commas.

- Install run list
  - The term *run list* is a Chef concept. PureApplication System further divides the run list into three separate lists to be run at different times during the node lifecycle.
  - The install run list is run one time only in the node's lifecycle during the middleware configuration step, just after the Chef client is installed. This is a special use of run list because users can use it to pass in dynamic attributes. In Example 14-1, cookbook FTP3 has only one simple default recipe.

Example 14-1 FTP3/recipes/default.rb - the default recipe in the FTP3 cookbook

```
#
# Cookbook Name:: FTP3
# Recipe:: default
#
# Copyright 2015, My Corporation
#
# All rights reserved - Do Not Redistribute
#

execute 'ftp3_bootstrap' do
 command "wget -qO- --no-check-certificate
https://companyx.net/pub/bootstrap/bootstrap.sh | /bin/bash"
end

execute 'ftp3_reg' do
 command "register_bootstrap_mode --force --username=#{node['FTP3']['user']}
--password=#{node['FTP3']['pass']}"
end
```

  - The install run list should include any recipe for which the configurations are expected to persist after a system restart or are not idempotent (for example, direct modifications to `/etc/sysconfig/ip` tables).
  - The recipe in Example 14-1 runs two commands. The second command passes in two attributes: `user` and `pass`. These Chef attributes are defined in the `FTP3/attributes/default.rb` file, as shown in Example 14-2. However, they are set to empty strings to not include hardcoded login credentials.

Example 14-2 FTP/attributes/default.rb

```
# FTP3 username and password attribute
# default is blank
# must pass in via IPAS JSON or middleware config will fail due to Install run
list failing

default['FTP3']['user']         = ""
default['FTP3']['pass']         = ""
```

– To pass in these attributes at deployment time, you can use the JSON attributes for overriding field or the Attributes file in JSON format field. Example 14-3 shows an example of the JSON that can be passed in for this particular example. The syntax of the JSON depends on how you define the attributes in your cookbook. The top level of the JSON defines the recipe. (In this sample cookbook, only one default recipe is included, and its name is the same as the cookbook name.) The second level defines the attributes. The use of double quotation marks is expected here. Otherwise, an exception is thrown at deployment time (for more information about debugging, see 14.5, “Debugging a deployment by using a Chef client on PureApplication System” on page 386).

Example 14-3 JSON with overriding attributes

```
{
   "FTP3": {
      "user": "testuser",
      "pass": "myftppass"
   }
}
```

- Default run list
  - This run list is run after the node finishes starting and after every node restarts. If the Chef client is running as a daemon, the run list runs periodically. The interval is defined in the Periodic intervals field.
  - This run list is appropriate for any idempotent configurations and configurations that might not persist through restart. This run list also runs when you run **`chef-client`**.
- Stop run list

  This run list is run when the node is stopped.

The deployment in Example 14-1 on page 385 runs the FTP3 recipe and point a deployment to a YUM package repository at deployment time. Then, it runs the recipe nginx and hardens to install and configure the nginx http server and hardens the node SSH configuration by disallowing SSH access by password login. Then, it updates the `authorized_keys` file with only authorized users’ public keys.

Because the Chef client is running as a daemon with 1800-second intervals (30 minutes), the **`chef-client`** command runs every 30 minutes to pick up and apply any configuration changes of the cookbooks in the default run list. If the node run list is modified by a Chef administrator that uses either the Management UI on the Chef server or the **`knife`** command on the Chef workstation, these **`chef-client`** actions also apply those cookbook changes.

14.5 Debugging a deployment by using a Chef client on PureApplication System

If a cookbook was reasonably tested and all dependencies resolve, they also work without any problem on PureApplication System. However, do not overlook slight environment differences because they can cause one or more dependencies to not resolve as expected. In a PureApplication System deployment, the Chef client installation and configuration occurs as the last step of middleware configuration. If your deployment hangs or fails at this step, one or more Chef cookbooks from either the install run list or the default run list are the likely culprits. To aid debugging and testing effort, add the Debug component to your pattern.

Figure 14-2 shows the location of the Debug component on the IBM Pattern Builder. You can drag it to anywhere within the canvas. It is a stand-alone component and does not need to be connected to any other components or nodes.

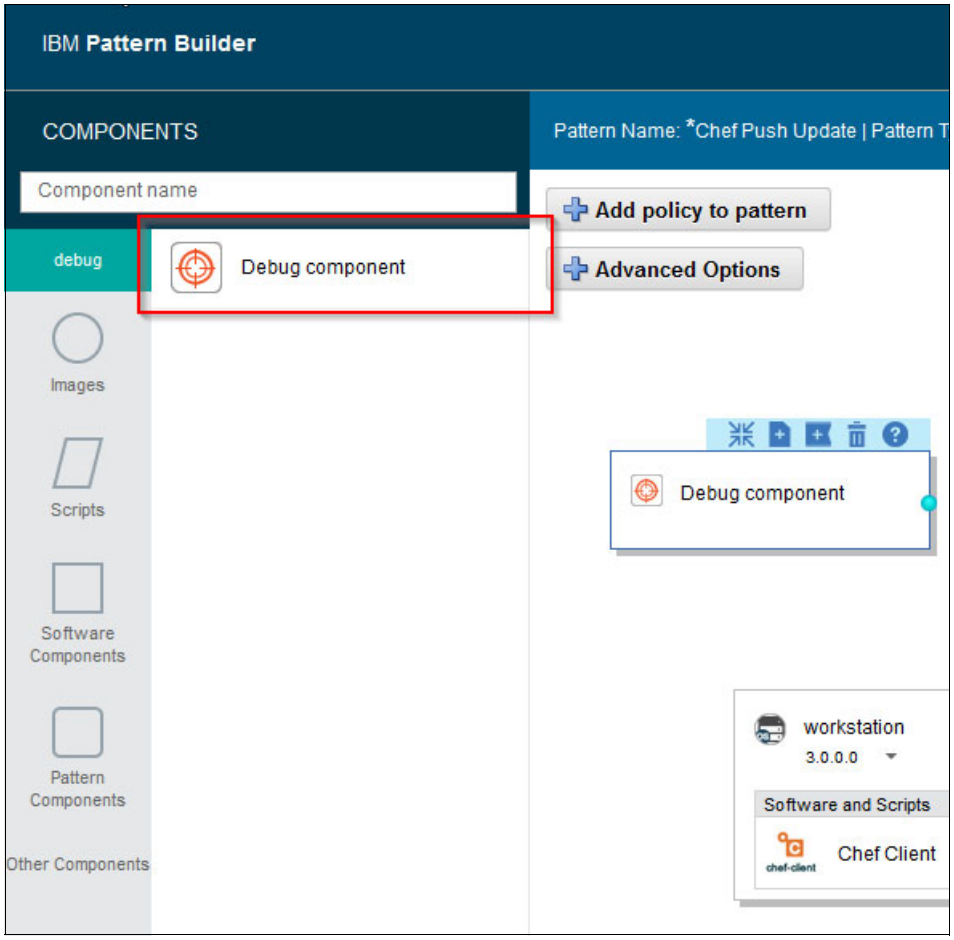


Figure 14-2 The debug component can be added to any pattern by using the IBM Pattern Builder

After you add the debug component, you can explore additional options on the right side of the canvas by clicking the component within the canvas, as shown in Figure 14-3.

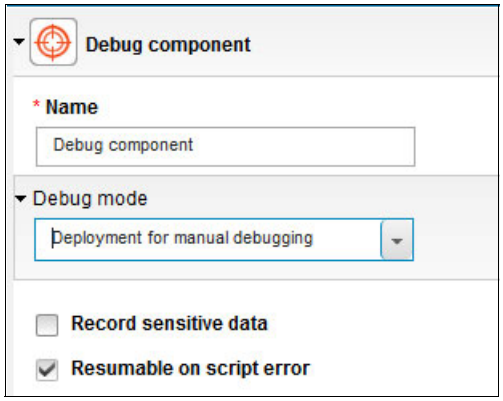


Figure 14-3 Additional options for the debug component

The settings in Figure 14-3 on page 387 suspend the deployment on error so that the deployer can review the error logs and correct any recoverable errors (for example, missing or invalid parameters or file system permissions issues). You can use the Resumable on script error function to resume the deployment after correcting the issues by running the following script within the problem virtual machine (VM):

```
/0config/nodepkgs/common/scripts/pdk-debug/resume.sh
```

Deployment

Here is an example problematic deployment attempt and how it can be debugged.

First, start the deployment of the pattern, as shown in Figure 14-4.

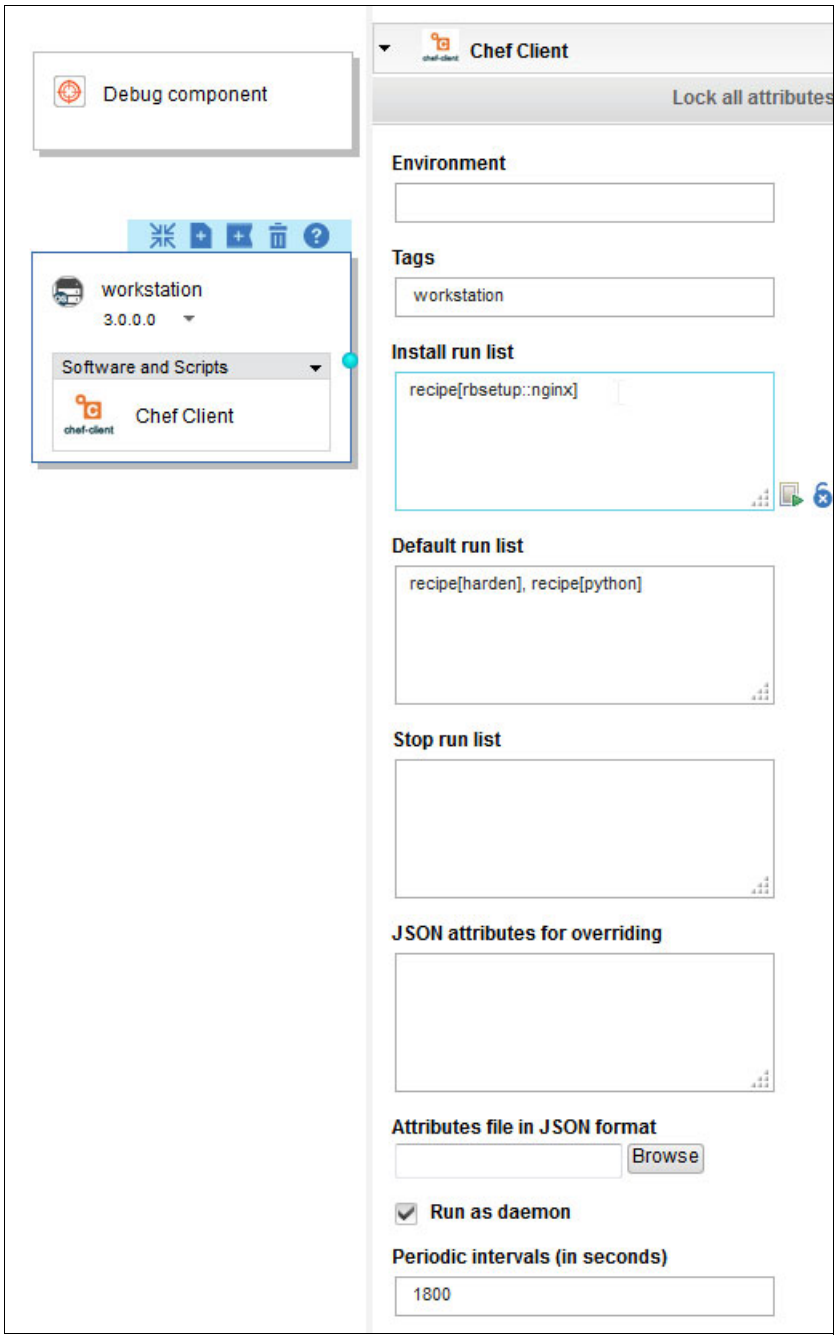


Figure 14-4 Example of a problematic deployment

The pattern in Figure 14-4 on page 388 was deployed *as is*, with the exception of the user name and password attributes for the FTP3 recipe that is shown in Example 14-3 on page 386.

Deployment error on the GUI

After a few minutes, the deployment progresses to Step 4 of 4: Preparing middleware (see the Status field and progress bar in Figure 14-5). An error is encountered here, so the debug component suspended its run. The middleware perspective in Figure 14-5 indicates which node is hung. This information is especially useful when a deployment has multiple nodes.

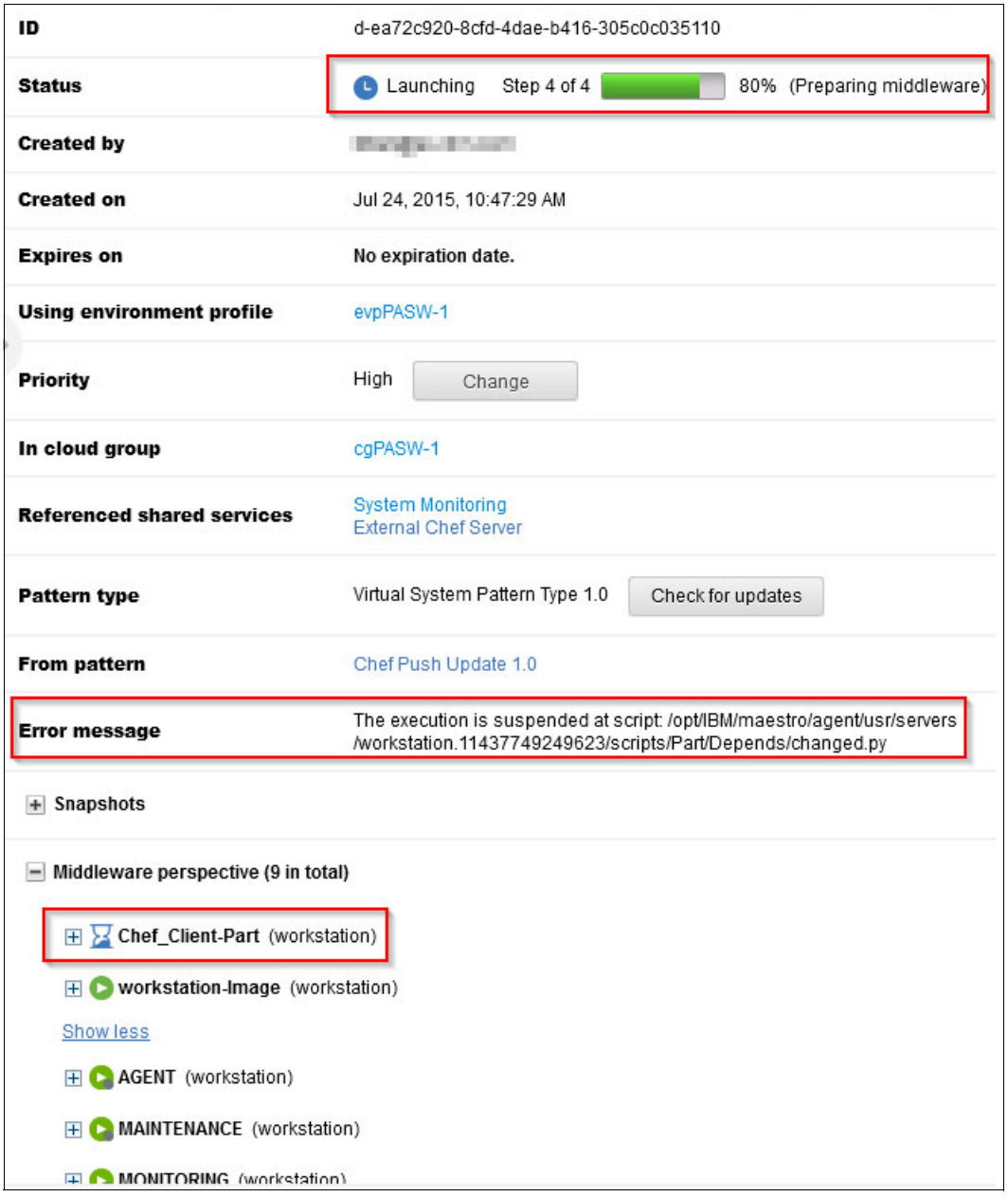


Figure 14-5 Virtual system instance detail window showing the Debug component in action

From this same view (Figure 14-5 on page 389), you can scroll down to the History section, where you can find more information. Figure 14-6 shows the deployment history of this instance. If a failure occurs on multiple nodes, the History section indicates the specific nodes to view.

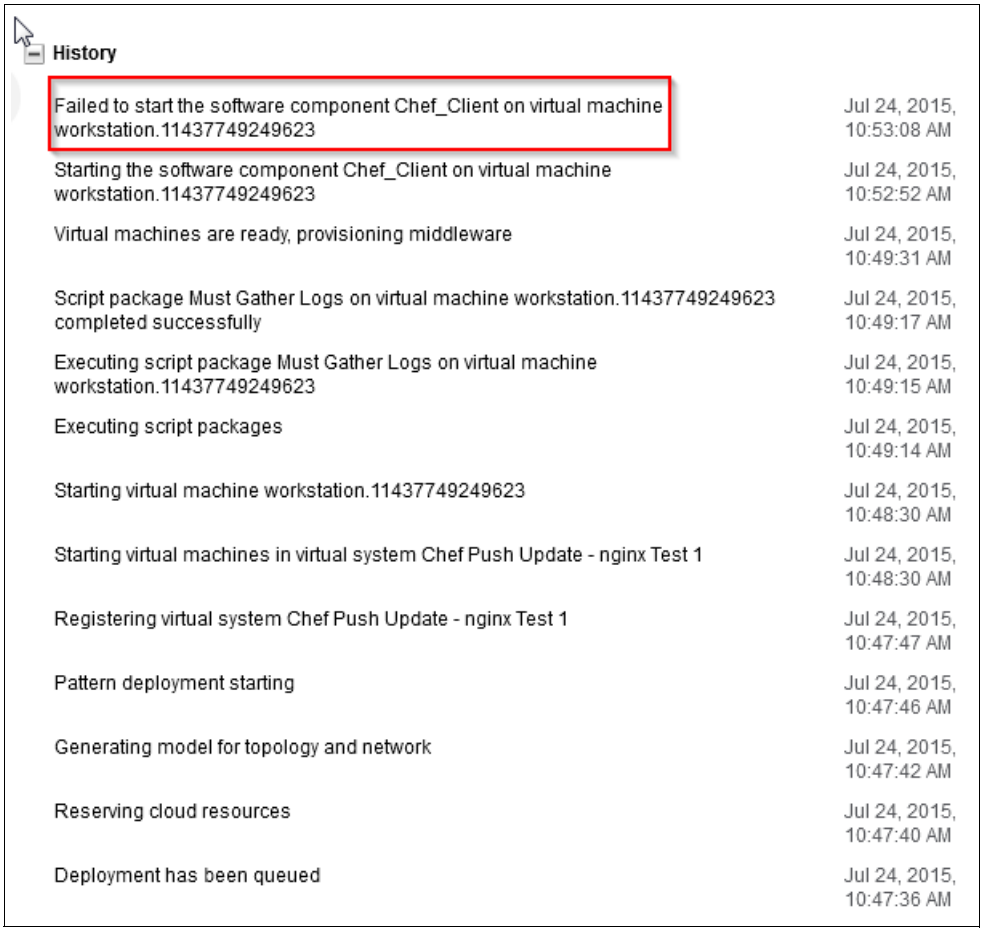


Figure 14-6 The deployment history is useful during debugging

Now, you know that the deployment encountered an error during the installation and configuration of the Chef client. The Chef client is contained in the Chef_Client part of the middleware role of the workstation node. Look at the relevant logs for this role to proceed.

Log review

To view the logs for a particular virtual system deployment, click the **Manage** icon at the upper right corner of the virtual system instance detail window, as shown in Figure 14-7. The instance console opens.

Figure 14-7 The Manage icon opens the Instance Console

Click the **Logging** tab to open the log viewer, which is shown in Figure 14-8.

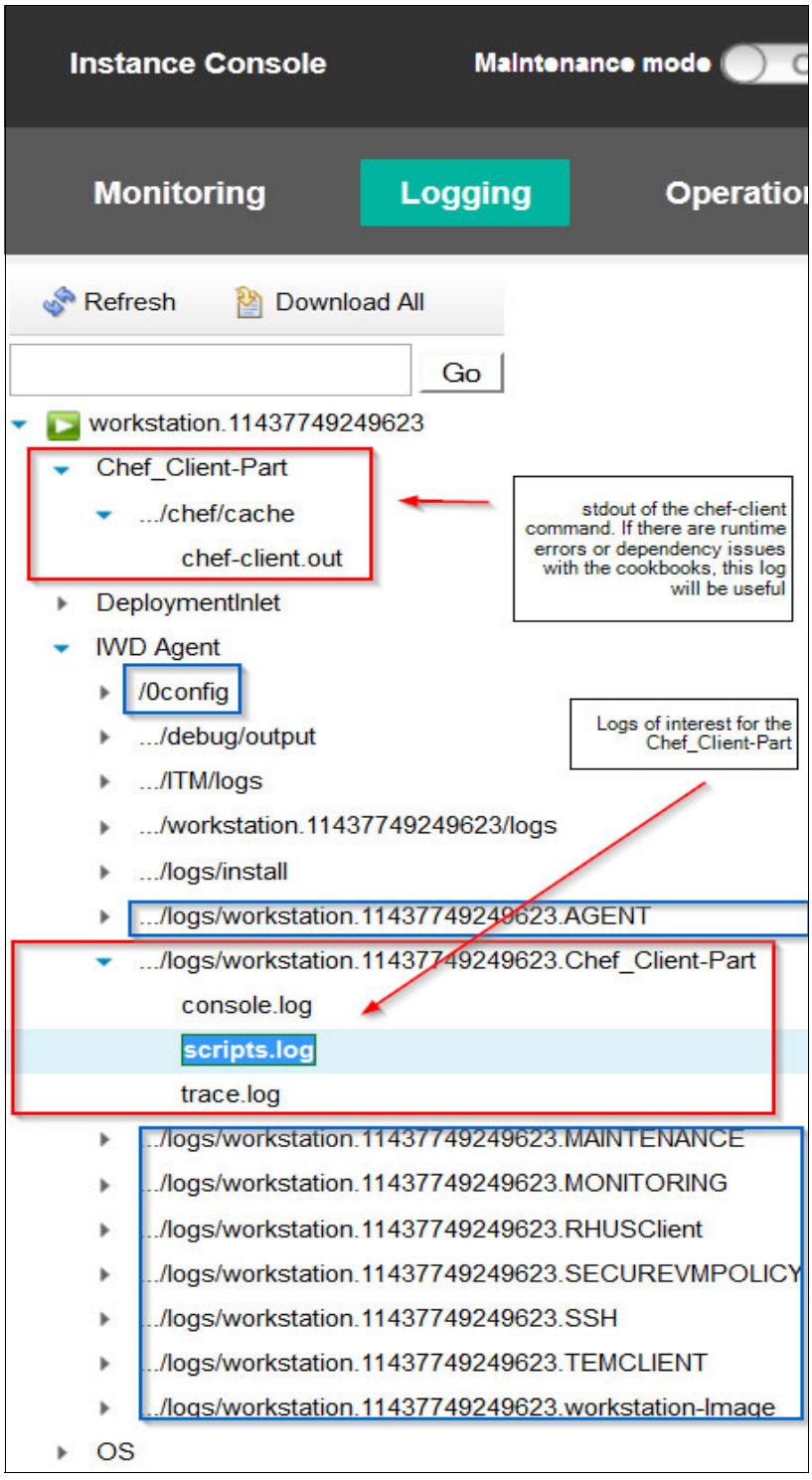


Figure 14-8 Logs of interest from the log viewer

Tip: With the exception of the `0config` directory, the logs in the blue boxes in Figure 14-8 represent other middleware roles of the deployments. If there are errors during the middleware configuration step, look at these places.

Note: The `0config` directory contains logs that pertain to the VM configuration step of the deployment. The state of the VM status column in the UI correlates to this `0config.log` file.

The `scripts.log` log file logs data that pertains to the script files that run the **`chef-client`** commands. Example 14-4 shows an excerpt of the `scripts.log` file. The error message is highlighted in red. The offending command is highlighted in blue.

Example 14-4 Excerpt of scripts.log

```
[Fri 24 Jul 2015 02:52:46 PM UTC] Chef_Client-Part Part/start.py 139911279023936
pid=14426 DEBUG Invoking script:
/opt/IBM/maestro/agent/usr/servers/workstation.11437749249623/scripts/ChefClient/s
tart.py
[Fri 24 Jul 2015 02:52:46 PM UTC] Chef_Client-Part Part/start.py 139911279023936
pid=14426 DEBUG Working dir:
/opt/IBM/maestro/agent/usr/servers/workstation.11437749249623/pyworkarea/requests/
1450500946818800934
[Fri 24 Jul 2015 02:52:46 PM UTC] ChefClient/start.py 139911279023936 pid=14426
INFO Executing Chef Client ...
[Fri 24 Jul 2015 02:52:46 PM UTC] maestro 139911279023936 pid=14426 DEBUG
VM_INSTALLED == False (node part)
[Fri 24 Jul 2015 02:52:46 PM UTC] ChefClient/chefutil.py 139911279023936 pid=14426
INFO knife command is:knife tag create s2iwd-214_workstation.11437749249623
workstation -u s2iwd-214_workstation.11437749249623 -c /etc/chef/client.rb -y
[Fri 24 Jul 2015 02:52:46 PM UTC] ChefClient/chefutil.py 139911279023936 pid=14426
INFO Executing: su - -c "knife tag create s2iwd-214_workstation.11437749249623
workstation -u s2iwd-214_workstation.11437749249623 -c /etc/chef/client.rb -y"
[Fri 24 Jul 2015 02:52:46 PM UTC] maestro 139911279023936 pid=14426 DEBUG WARNING:
shell=True used for subprocess command execution; might be insecure.
[Fri 24 Jul 2015 02:52:49 PM UTC] ChefClient/chefutil.py 139911279023936 pid=14426
DEBUG Created tags workstation for node s2iwd-214_workstation.11437749249623.

[Fri 24 Jul 2015 02:52:49 PM UTC] ChefClient/start.py 139911279023936 pid=14426
INFO Register Chef Client Environment and Attribute...
[Fri 24 Jul 2015 02:52:49 PM UTC] ChefClient/chefutil.py 139911279023936 pid=14426
INFO Executing: su - -c "chef-client  -j "/etc/chef/attributes.json" --logfile
/var/chef/cache/chef-client.out"
[Fri 24 Jul 2015 02:52:49 PM UTC] maestro 139911279023936 pid=14426 DEBUG WARNING:
shell=True used for subprocess command execution; might be insecure.
[Fri 24 Jul 2015 02:53:01 PM UTC] ChefClient/chefutil.py 139911279023936 pid=14426
INFO knife command is:knife node run_list set s2iwd-214_workstation.11437749249623
"recipe[harden], recipe[python]" -u s2iwd-214_workstation.11437749249623 -c
/etc/chef/client.rb -y
[Fri 24 Jul 2015 02:53:01 PM UTC] ChefClient/chefutil.py 139911279023936 pid=14426
INFO Executing: su - -c "knife node run_list set
s2iwd-214_workstation.11437749249623 "recipe[harden], recipe[python]" -u
s2iwd-214_workstation.11437749249623 -c /etc/chef/client.rb -y"
[Fri 24 Jul 2015 02:53:01 PM UTC] maestro 139911279023936 pid=14426 DEBUG WARNING:
shell=True used for subprocess command execution; might be insecure.
```

```
[Fri 24 Jul 2015 02:53:01 PM UTC] ChefClient/chefutil.py 139911279023936 pid=14426
DEBUG su: user recipe[python] -u s2iwd-214_workstation.11437749249623 -c
/etc/chef/client.rb -y does not exist
```

According to the debug message in Example 14-4 on page 392, the **su** command is evaluating part of the **knife** command (see the blue text and the white space in **recipe[harden], recipe[python]**) as the user parameter. In Example 14-5, a quick test is run on the CLI to confirm this behavior.

Example 14-5 Linux console test to confirm the behavior or the su command

```
[root@playbox recipes]# su - -c "echo "hello, world""
su: user world does not exist
[root@playbox recipes]# su - -c "echo "hello,world""
hello,world
[root@playbox recipes]# su - root -c "echo "hello, world""
hello,
```

From the output that is shown in Example 14-5, it is apparent that the code does not handle white space between recipes. If you see Figure 14-4 on page 388, you can see that there is white space between two of the recipes in the Default run list.

To correct this issue, either delete the deployment and redeploy it without white spaces between recipes, or modify the parameters directly in the JSON files from which the code is reading. The first option is easier and safer, but requires redeployment. The second option requires modification of the `parameter.json` file on the server side (PureApplication System Storehouse Browser) and client side (the VM downloaded the `parameter.json` file from the server during the initial deployment).

In this example, choose the second option to see how you can resume from an error by using the debugging component. To edit `parameter.json` on the server-side, click **System** → **Storehouse Browser** and expand **user** → **deployments** → **[deployment id]**, as shown in Figure 14-9. The deployment ID can be found on the virtual system instance detail page (ID field).

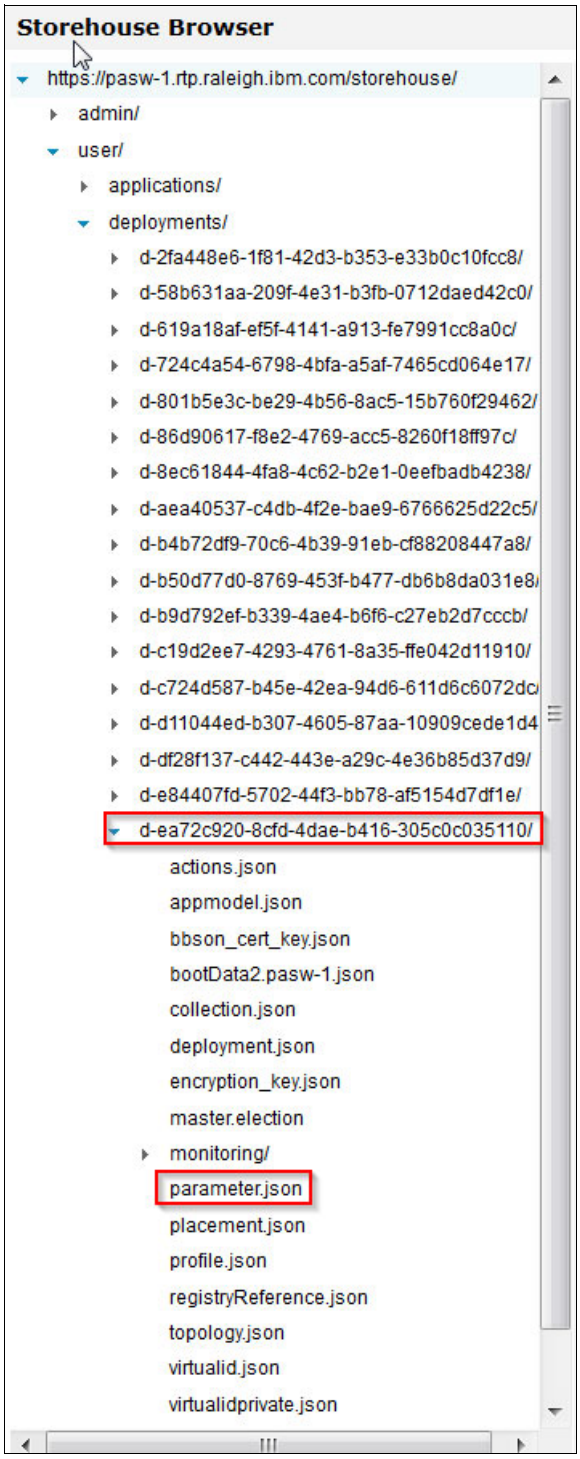


Figure 14-9 Storehouse browser location where the server-side parameter,json is found

Alternatively, you can download `parameter.json` by using an HTTP **`GET`** request, as shown in Example 14-6.

Example 14-6 HTTP GET request to download parameter.json from the storehouse

```
GET
/storehouse/user/deployments/d-ea72c920-8cfd-4dae-b416-305c0c035110/parameter.json
```

In this example, save two copies of the `parameter.json` file (one as `parameter.json.bak`). Make the edit that is shown in Example 14-7 to the `parameter.json` file to remove the white space between the two recipes in the Default run list.

Example 14-7 Stream edit to remove white space

```
-bash-4.2# sed -i 's/recipe\[harden\],
recipe\[python\]/recipe\[harden\],recipe\[python\]/g ./parameter.json
```

After fixing `parameter.json`, use an HTTP **`PUT`** command to upload it back to the storehouse, as shown in Example 14-8.

Example 14-8 HTTP PUT command to upload the fixed parameter.json to the storehouse

```
PUT
/storehouse/user/deployments/d-ea72c920-8cfd-4dae-b416-305c0c035110/parameter.json

Body: { <content of the json> }
```

Before you can resume the failing script, you must edit `parameter.json` and any references of the Default run list on the client side. Access the VM by using SSH. Run the commands in Example 14-9 to fix all references of the Default run list on the VM and resume deployment.

Example 14-9 Fix all references of the Default run list on the VM and resume deployment

```
-bash-4.2# grep -rl --include="*.json" "recipe\[harden\], recipe\[python\]" / |
xargs sed -i 's/recipe\[harden\],
recipe\[python\]/recipe\[harden\],recipe\[python\]/g'
-bash-4.2# /Oconfig/nodepkgs/common/scripts/pdk-debug/resume.sh
the dump file:
/opt/IBM/maestro/agent/usr/servers/workstation.11437746372282/pyworkarea/requests/
3381167376413375974/dump.json has been removed, the execution should be resumed in
seconds...
```

In Example 14-8, The **`grep`** command finds all `*.json` files with the recipe reference, and the **`sed`** command removes the white space between the recipes.

After the deployment resumes, it completes and goes into the Running state, as shown in Figure 14-10.

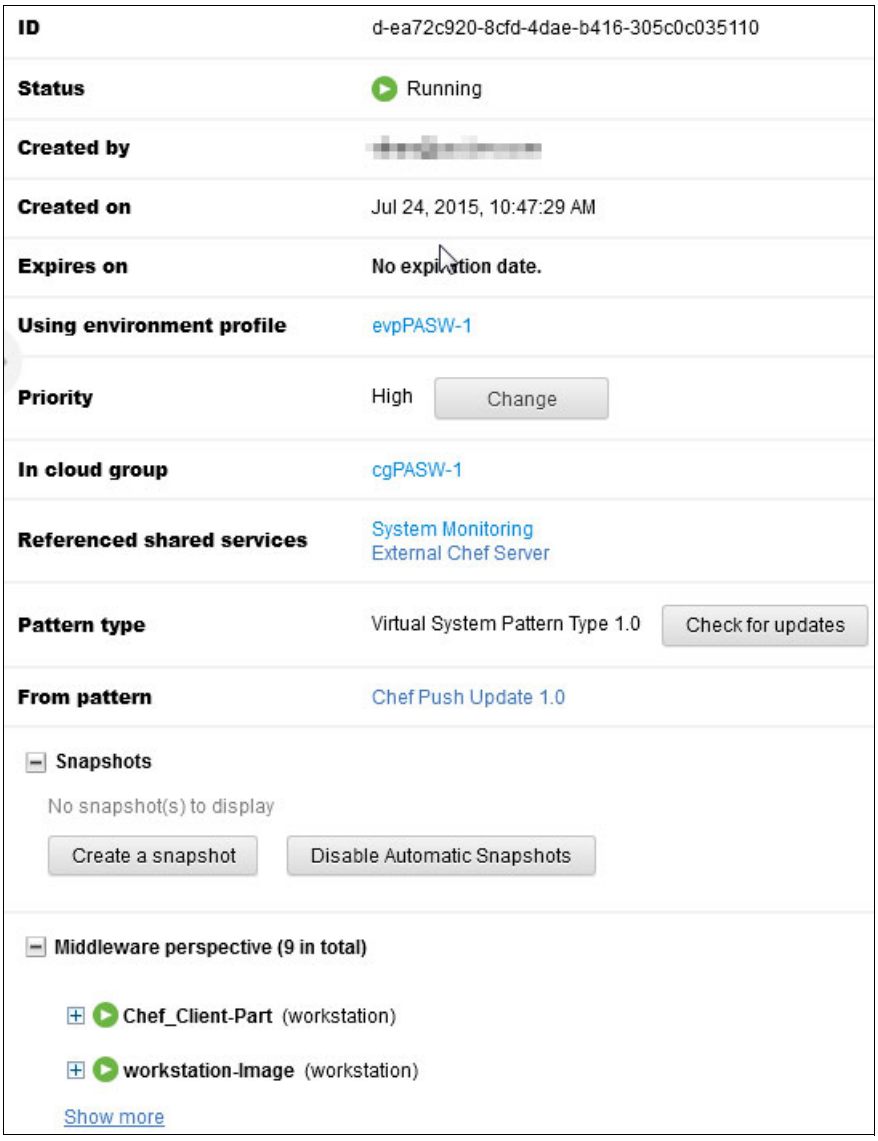


Figure 14-10 Deployment complete

Related publications

The publications that are listed in this section are considered suitable for a more detailed description of the topics that are covered in this book.

Online resources

These websites are also relevant as further information sources:

- Certificates in IBM PureApplication System:

  https://www.ibm.com/support/knowledgecenter/#!/SSCR9A_2.1.0/doc/iwd/security_import_certs.dita

- Chef workstation installation:

  https://docs.chef.io/install_dk.html

- Create Environment profiles:

  https://www.ibm.com/support/knowledgecenter/?lang=en#!/SSCR9A_2.1.0/doc/iwd/ept_creaprof.dita

- How IBM PureApplication System uses DNS:

  http://www.ibm.com/developerworks/websphere/techjournal/1407_woolf1/1407_woolf1.html

- IBM PureApplication Software IBM Knowledge Center - What's new:

  http://www.ibm.com/support/knowledgecenter/SSL5ES_2.1.0/doc/iwd/gsr_changes.dita?lang=en

- IBM PureApplication Software Release Notes by version:

  http://www.ibm.com/support/docview.wss?uid=swg27045293

- IBM PureApplication System Enterprise configuration:

  https://www.ibm.com/support/knowledgecenter/#!/SSCR9A_2.1.0/doc/installguide/planning/ps_sysreq_gen2enterprise.dita

- IBM PureApplication System Mini configurations:

  https://www.ibm.com/support/knowledgecenter/api/content/nl/en-us/SSCR9A_2.1.0/doc/installguide/planning/ps_sysreq_gen2mini.html

- IBM PureApplication System Release Notes by version:

  http://www.ibm.com/support/docview.wss?uid=swg27039159

- IBM PureApplication System W2500 IBM Knowledge Center - What's new:

  http://www.ibm.com/support/knowledgecenter/SSCR9A_2.1.0/doc/iwd/gsr_changes.dita?lang=en

- IBM PureApplication System W2700 Enterprise racks:

  https://www.ibm.com/support/knowledgecenter/#!/SSCRSX_2.1.0/doc/installguide/planning/ps_sysreq_gen2enterprise.dita

© Copyright IBM Corp. 2015. All rights reserved.

- IBM PureApplication System W2700 IBM Knowledge Center - What's new:

  http://www.ibm.com/support/knowledgecenter/SSCRSX_2.1.0/doc/iwd/gsr_changes.dita?lang=en
- IBM PureApplication System W2700 Mini racks:

  https://www.ibm.com/support/knowledgecenter/#!/SSCRSX_2.1.0/doc/installguide/planning/ps_sysreq_gen2mini.dita
- IBM PureSystems Center:

  https://www.ibm.com/software/brandcatalog/puresystems/centre/
- The IBM PureSystems family of systems:

  http://www.ibm.com/ibm/puresystems/us/en/
- IBM Support Portal - Fix Central:

  http://www.ibm.com/support/fixcentral/
- Passport Advantage:

  http://www.ibm.com/software/passportadvantage
- Persistent and non-persistent virtual machines:

  http://www.ibm.com/support/knowledgecenter/SSCR9A_2.1.0/doc/virt_app_patterns/ap/apc_trbpattern.dita
- Red Hat Customer Portal:

  https://access.redhat.com/
- Red Hat Satellite entitlement certificate:

  http://www.ibm.com/support/knowledgecenter/SSCR9A_2.1.0/doc/iwd/t_create_redhat_satlitesrv_certificate.dita
- Security roles in IBM PureApplication System:

  https://www.ibm.com/support/knowledgecenter/#!/SSCR9A_2.1.0/doc/iwd/aac_user_permissions.dita
- Site-readiness planning:

  https://www.ibm.com/support/knowledgecenter/#!/SSCR9A_2.1.0/doc/installguide/planning/siteplan.dita
- Topologies andcommunication options for applications:

  https://www.ibm.com/support/knowledgecenter/#!/SSCR9A_2.1.0/doc/installguide/planning/networkplan.dita
- Users roles:

  https://www.ibm.com/support/knowledgecenter/#!/SSCR9A_2.1.0/doc/iwd/aat_manusui.dita
- VMware documentation:

  http://pubs.vmware.com/vsphere-51/index.jsp?topic=%2Fcom.vmware.vsphere.doc%2FGUID-1B959D6B-41CA-4E23-A7DB-E9165D5A0E80.html

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Redbooks

Integrating IBM PureApplication System into an Existing Data Center

SG24-8285-00

ISBN 0738441120

IBM

(0.5” spine)
0.475”<->0.873”
250 <-> 459 pages

SG24-8285-00

ISBN 0738441120

Printed in U.S.A.

Get connected