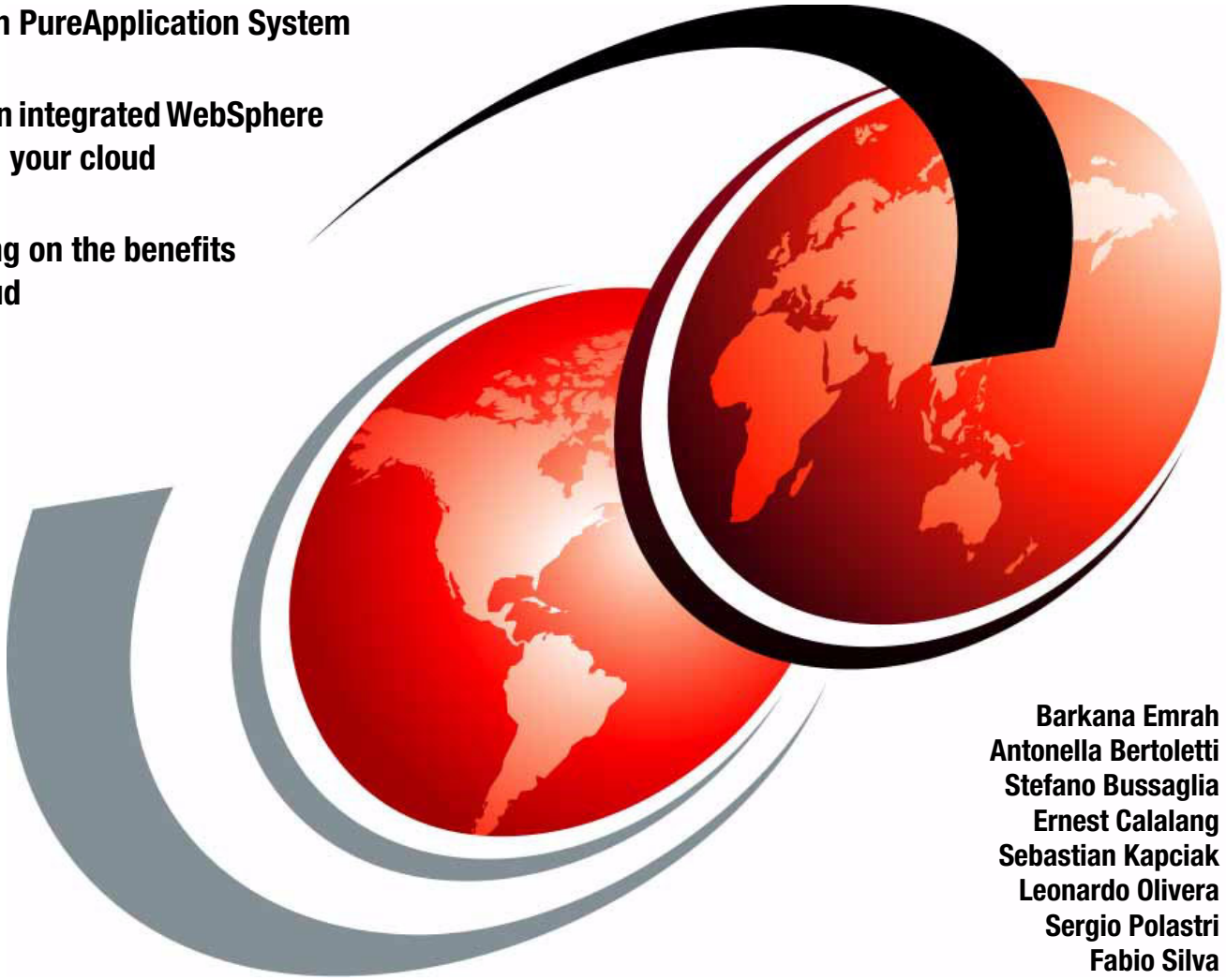
IBM

Rapid Deployment of Integrated WebSphere Solutions in Your Cloud

- Deploying an integrated WebSphere solution on PureApplication System
- Creating an integrated WebSphere solution in your cloud
- Capitalizing on the benefits of the cloud



Barkana Emrah
Antonella Bertoletti
Stefano Bussaglia
Ernest Calalang
Sebastian Kapciak
Leonardo Olivera
Sergio Polastri
Fabio Silva

ibm.com/redbooks

Redpaper

International Technical Support Organization

Rapid Deployment of Integrated WebSphere Solutions in Your Cloud

December 2014

REDP-5132-00

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

First Edition (December 2014)

This edition applies to IBM PureApplication System v1.1.

This document was created or updated on December 15, 2014.

Contents

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.

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®
CICS®
DB2®
Global Business Services®
Global Technology Services®
IBM PureData™
IBM UrbanCode™
IBM®
ILOG®
Passport Advantage®
PureApplication®
PureData™
PureSystems®
Rational®
Redbooks®
Redbooks (logo) ®
Tivoli®
TXSeries®
WebSphere®
Worklight®

The following terms are trademarks of other companies:

Worklight is trademark or registered trademark of Worklight, an IBM Company.

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.

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.

Preface

This IBM® Redbooks® publication uses the same practical solution and is an extension to *Creating Integrated WebSphere Solutions using Application Lifecycle Management*, SG24-8243-00. This paper demonstrates how to take an existing application that was built in a legacy environment, and bring that application to IBM PureApplication® Systems, using preferred practices for deployment and automation. The process is illustrated using a business scenario.

This publication is intended for architects, developers, and administrators who want to know about the next generation of technology that modern IT organizations are moving rapidly towards: Application integration and systems development.

Authors

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

Barkana Emrah is an IBM Certified IT Specialist working in SWG group in Turkey. He joined IBM in May 2008 as ISSW. Since then, Emrah has worked in several industrial areas, specialized in WebSphere Application Server, WebSphere Process Server, WebSphere Portal Server, IBM ILOG®, Business Process Management, and Master Data Management. Before he joined IBM, he was working for ISBANK in Turkey as business analyst and WebSphere Process Server developer.

Antonella Bertoletti is an Executive IT Specialist working for IBM Italy, in Milan. She joined IBM in 1987 and spent the first 10 years of her career in the HW Division (now STG). She is specialized in both the OLTP areas (IBM CICS® and TXSeries®, WebSphere Application Server, CICS Transaction Gateway) and Cloud Computing and Virtualization (IBM PureApplication Systems, IBM Workload Deployer, WAS Hypervisor Edition, WebSphere Virtual Enterprise, WebSphere Compute Grid, WebSphere eXtreme Scale). She has been involved in many different projects using OLTP products, distributed computing transactional technology, and virtualization on several platforms, especially in the Insurance, Banking, and Manufacturing sectors. Antonella is member of the IBM Academy of Technology, and is the PureApplication System Technical Leader for Italy. She has written IBM Redbooks and White Papers on TXSeries, CICS Transaction Gateway, and WebSphere products. She is currently working for the Software Group Division, in the WebSphere Technical Sales and Solutions - Europe IOT team. Antonella holds a degree in Economics from Pavia University.

Stefano Bussaglia is a WebSphere Client Technical Specialist at IBM Switzerland. Before joining IBM, he was working as a Middleware specialist in a bank, developing the backbone with IBM WebSphere software. Stefano has more than 30 years of IT experience, mainly in developing client/server and web applications. His client experiences come from running his own company from more than 10 years. His current area is serving clients from IBM with IBM PureApplication System and WebSphere Applications Server.

Ernest Calalang is a Certified Advisory IT Specialist in the IBM Software Services for WebSphere in the Philippines. He is also an IBM Certified Instructor for the Application Integration and Middleware portfolio. Since joining IBM in 2008, he has six years of industry experience in the banking, telecommunications, and utility sectors. His areas of expertise include enterprise integration using IBM Integration Bus, WebSphere Application Server, WebSphere DataPower, WebSphere Business Process Management, and IBM PureApplication Systems. He holds a degree of Bachelor of Science in Computer Science from the University of Santo Tomas, Philippines.

Sebastian Kapciak is an IBM Certified IT Specialist working for IBM Poland. He has more than 10 years of IT experience in delivering middleware, security, and open source solutions. His areas of expertise are system integration and JEE technologies. Sebastian is certified Open Group TOGAF9 architect and holds a Master's degree in Information Technology from the University of Technology of Warsaw.

Leonardo Olivera is an IT Specialist at IBM Global Services in Uruguay. He has 20 years of experience in application development and systems integration. He holds a degree in Computer Science Engineering from Universidad Católica del Uruguay. His areas of expertise include Java Enterprise Edition architecture, sensor and actuator solutions, and application security. During the last few years, he has been working with mobile application platforms, including IBM Worklight®.

Sergio Polastri is an IBM IT Specialist at IBM Global Business Services® in Brazil. He joined IBM in August 2000 and since then has worked with IBM Rational® products and solutions. He currently leads the Tools team of the Processes Method and Tools Team. His team supports Rational Tools deployment over more than 45 projects in Global Delivery Center (Latin America). He holds a Bachelor's degree in Computer Systems Engineering from Centro Universitario da Cidade do Rio de Janeiro in Brazil, and a Post-Graduate Degree in Computer Network Management from Centro Universitario da Cidade to Rio de Janeiro in Brazil.

Fabio Silva is an IT Specialist working for IBM Global Technology Services® in Brazil. He joined at IBM in January 2007 and spent the first six years of his career supporting WebSphere family and now PureApplication System. His main areas of expertise are automation for WebSphere administration, PureApplication Script Packages, and Plug-ins; infrastructure design in Legacy and Cloud; and implementation, maintenance and problem determination in the web environment. He has more than 15 years of IT expertise in fields, including middleware, Java programming, open source tools, UNIX operating systems, network security, and web hosting environments. He has designed, implemented, and supported various middleware infrastructure and web hosting environments in large public and private organizations. He holds a Bachelor's degree in Compute System Engineering from Centro Universitario da Cidade do Rio de Janeiro in Brazil.

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 contributions to this project:

- Jim Amsden, Senior Technical Staff Member Solution Architect for Rational Government Industry Solutions
- Rajeev Gandhi, STSM, ISSW Tech Consultant
- Derrick W Foley, Manager, WebSphere Cloud Initiatives
- Robbie John Minshall, IBM Software Group, Rational, DevOps, Continuous Delivery, and Cloud Adoption
- Scott Walden, IBM PureApplication System Enablement

Thanks to the following people for supporting this project:

- Deana Coble, IBM Redbooks Technical Writer and Video
- Richard Conway, IT Support
- Karen Lawrene, IBM Redbooks Technical Writer
- Tamikia Lee, IBM Redbooks Residency Administrator
- Mahesh Nagabhushan, IBM Redbooks Graphics Editor
- Ernest Keenan, IBM Redbooks Editor

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 papers to be as helpful as possible. Send us your comments about this paper 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

Introduction

This book has two interconnected goals:

- To show, in step-by-step fashion, how to bring an existing solution to the cloud using IBM PureApplication Systems.
- To demonstrate, by moving the solution to the cloud, how several steps, which are conventionally carried out manually, can be easily automated.

This paper demonstrates how to take an existing application that was built in a legacy environment, and bring that application to PureApplication Systems, using preferred practices for deployment and automation. The process is illustrated by using a business scenario. The scenario is described in detail in *Creating integrated WebSphere solutions using Application Lifecycle Management*, SG24-8243. The book is available at:

http://www.redbooks.ibm.com/abstracts/sg248243.html

This chapter includes the following sections:

- Who this book is intended for
- Cloud consideration
- Bring the solution to the IBM PureApplication Systems
- Overview of the example business scenario

1.1 Who this book is intended for

This publication is intended for architects, developers, and administrators who want to know about the next generation of technology that modern IT organizations are moving rapidly towards: Application integration and systems development. With this rapid change, organizations now try to reach these goals:

- Increase volume output or productivity with fewer people
- Work on a cloud, with a minimal learning curve for hardware and software issues
- Reduce the costs that are associated with software and mobile application development
- Maintain the quality of product offerings
- Decrease deployment time from months to hours

1.2 Cloud consideration

Cloud computing is a flexible way to deliver high-end computing resources that can be shared by multiple software vendors and scaled on demand as the workload grows. Today's businesses need to be agile to develop and deploy applications dynamically to meet both market and client needs. Cloud computing is revolutionizing the way organizations provide, manage, and use IT services. It builds on efficient infrastructures, improving productivity and helping to reduce administrative and capital costs.

1.2.1 The IBM PureSystems solutions

IBM PureApplication System is a member of the IBM Expert Integrated Systems family. What sets this family of systems apart is the ability to include, in the same system, integration compute nodes, storage, networking, and platform middleware for physical and virtual resources. Some specific models of the family, such as IBM PureApplication Systems and IBM PureData™ System, also include built-in expertise from IBM factory pre-configuration, integration, and testing. This simplified experience enables clients to deliver new cloud services and applications faster and with improved economics over traditional IT practices. This paper integrates the solution to IBM PureApplication Systems and describes the integration in terms of the business scenario mentioned in the 1.3, "Overview of the example business scenario" on page 3.

1.2.2 The SoftLayer solution

SoftLayer Technologies, Inc. is a dedicated server, managed hosting, and cloud computing provider. The company, founded in 2005, was acquired by IBM in 2013. SoftLayer offers a high performance cloud infrastructure that clients can acquire. Currently, SoftLayer manages 15 data centers around the world, providing these same services and quality of service.

Clients can select among several offerings according to their needs, thanks to a considerable level of flexibility. This way, every company can create its own dedicated cloud infrastructure in the most suitable data center around the world, and communicate freely and securely over private or public networks.

Recently, IBM announced the availability of IBM PureApplication Service on SoftLayer, allowing the option to choose the IBM PureApplication Systems technology on a SoftLayer cloud infrastructure instead of purchasing a physical box to host in a data center.

A more detailed description of the IBM PureApplication Service on the SoftLayer offering is located in 3.2.2, “Defining resources on IBM PureApplication Service on SoftLayer” on page 26.

1.2.3 Bring the solution to the IBM PureApplication Systems

This paper describes how to create patterns and deploy an integrated application on the PureApplication Systems. This paper demonstrates, in step-by-step fashion, the creation of a new solution platform on the cloud, using virtual application patterns and virtual system patterns. Then you integrate the software products described in *Creating integrated WebSphere solutions using Application Lifecycle Management*, SG24-8243, which focuses on software in the IBM WebSphere Portfolio that is used to integrate businesses and systems. The book is available at:

http://www.redbooks.ibm.com/abstracts/sg248243.html

In the book, all of the steps were done manually, simulating conventional installation and configuration of servers and software. For those organizations that continue to use the conventional method for creating environments, the book describes how you can complete these objectives:

- Create these environments in the cloud
- Capitalize on the benefits of the cloud

In a cloud environment, the client manages the computing resources and the software capabilities on demand. The physical hardware must be an abstraction layer where the resources are dynamically managed and allocated without compromising other resources. One of the best characteristics of the cloud is elasticity; that is, increasing or decreasing the computing capacity, on demand.

For more information about the PureApplication Systems, see the two following publications:

- *Adopting IBM PureApplication System V1.0*, SG24-8113:

  http://www.redbooks.ibm.com/abstracts/sg248113.html
- *IBM PureApplication System Best Practices*, SG24-8145

  http://www.redbooks.ibm.com/abstracts/sg248145.html?Open

1.3 Overview of the example business scenario

The business scenario used for promoting a conventional installation to the cloud takes place at a fictitious organization, the *Redbooks Company Service Desk*. The following is the high-level process overview:

- The environment consists of two different front ends: One web and one mobile
- Both services (web and mobile) trigger the business processes at the Redbooks Company Service Desk.
- The process handles the execution of the business rules and the required approvals from technical support and the financial manager for completing the process
- The process performs the enterprise service bus role and translates the incoming messages so that process management can understand them
- The last important component of the solution is the database, where all claim records are stored and updated throughout the business process lifecycle.

Figure 1-1 shows this process graphically.

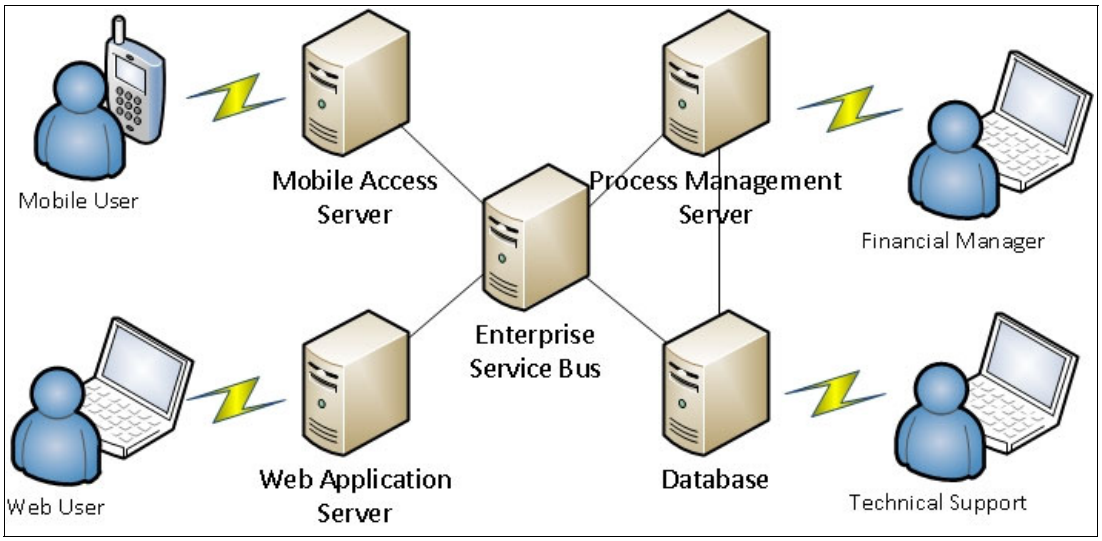


Figure 1-1 Solution overview for the business scenario at the Redbooks Company Service Desk

The following patterns are used to implement the solution on the PureApplication Systems:

- Web application as a virtual system pattern
- Mobile access server as a virtual application pattern
- Enterprise service bus and IBM DB2® as a virtual system pattern
- Process management server as a virtual application pattern
- Operation Decision Manager as a virtual application pattern

2

Patterns of expertise

This chapter describes the standard patterns that are provided by IBM that are used to build the claim solution described in the example business scenario (see 1.3, “Overview of the example business scenario” on page 3). Most standard patterns have many different possibilities when defining topologies. This book addresses the patterns that are used for a runtime environment, similar to those in a production environment. This example does not deal with high availability or disaster recovery for this deployment. To run a secured and scalable system, see 3.3.6, “Adding scaling policies” on page 48.

The chapter includes the following sections:

- Patterns that are used for the solution
- Loading patterns into the system

For more information about patterns of expertise, see Chapter 1 in *Cloud Computing Patterns of Expertise*, REDP-5040.

2.1 Patterns that are used for the solution

The solution requires different middleware components that WebSphere offers in virtual systems and in virtual application patterns. Each product is built upon standard patterns that are augmented with the scripts. The scripts can be obtained from *Creating integrated WebSphere solutions using Application Lifecycle Management*, SG24-8243, available at:

http://www.redbooks.ibm.com/abstracts/sg248243.html

Figure 2-1 shows an overview of the pattern groups that were used to build the solution in the cloud.

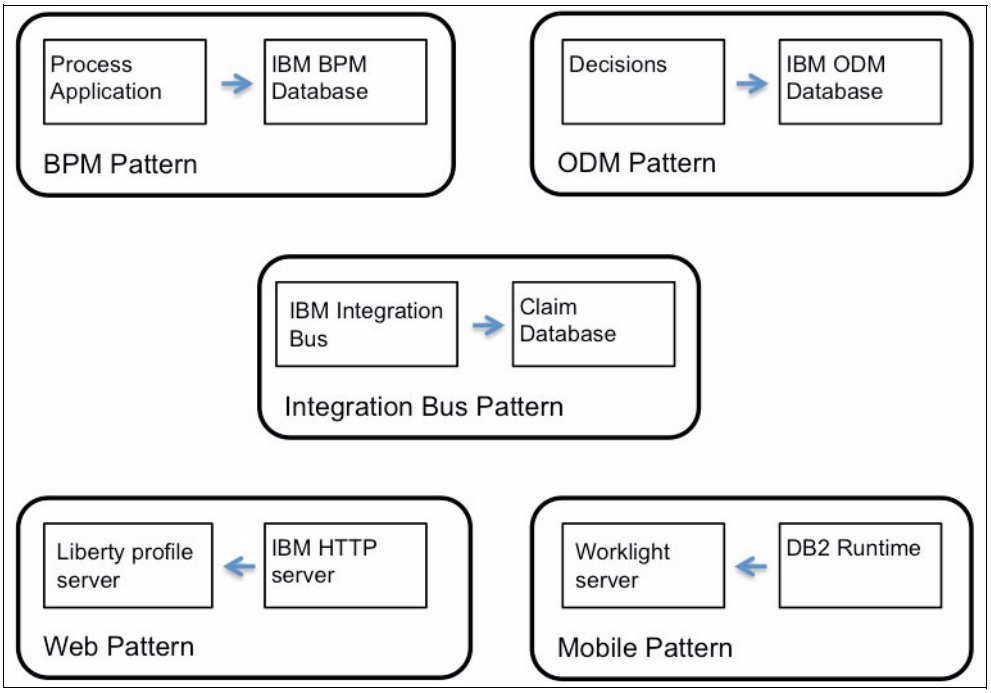


Figure 2-1 Patterns that were used for the cloud solution

2.1.1 Virtual images

Virtual images that are used by the IBM PureApplication System are Open Virtualization Format (OVF) compliant images, with special activation logic to help with deployment. There are a growing number of these virtual images for IBM Software products, which are named Hypervisor Editions.

The most fundamental building blocks for virtual system patterns are the *parts* (described in 2.1.3, “Virtual system patterns” on page 7). The parts are delivered with Hypervisor Edition images (described in 2.1.2, “Hypervisor Edition images” on page 7). This chapter describes the patterns for the different virtual images that are used for the solution. These include the patterns (the Hypervisor Edition images with composing elements) and the parts (used to deliver those elements and images). For more information about virtual system patterns, see Chapter 4 of in *Adopting IBM PureApplication System V1.0*, SG24-8113, available at:

http://www.redbooks.ibm.com/abstracts/sg248113.html

2.1.2 Hypervisor Edition images

An edition image is the delivery of a middleware product that is packaged according to the OVF in an Open Virtualization Format Archive (OVA) file. These images are imported into a virtual image catalog in the PureApplication System.

A Hypervisor Edition image consists of a middleware product (such as WebSphere Application Server) that is preinstalled and pre-configured with an operating system (often Linux or IBM AIX®), and is designed for virtual environments. As an example, the virtual image of WebSphere Application Server includes the following features:

- An operating system
- WebSphere Application Server
- IBM HTTP Server binary files
- WebSphere Application Server profiles
- A combination of code and tuning that is built into the image to optimize the server for a virtual environment

To view virtual images on the PureApplication System Workload console, select **Catalog** → **Virtual Images** (Figure 2-2).

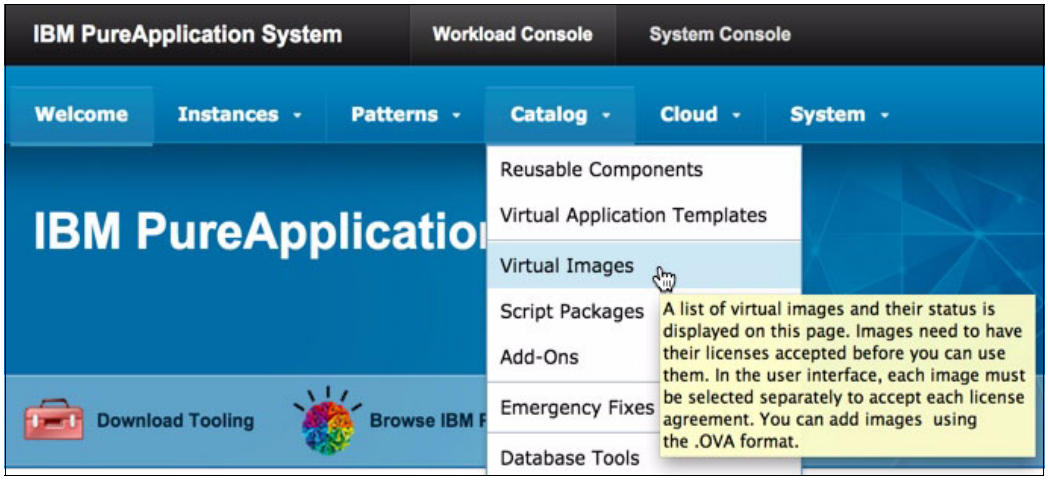


Figure 2-2 Virtual images

2.1.3 Virtual system patterns

Virtual system patterns (VSPs) are repeatable topology definitions that are based on various virtual images, each containing multiple middleware components and applications that are configured to work with each other. VSPs provide flexibility and control over the middleware topology that is deployed.

The fundamental building blocks of VSPs are called *parts*. These parts are delivered with the virtual images and are used, with configuration parameters and other artifacts, such as *script packages* and *add-ons*, to create complex VSPs that are deployed as a single unit.

Learn more about virtual system patterns in Chapter 1 of *Cloud Computing Patterns of Expertise*, REDP-5040.

Editing virtual system patterns

A virtual system pattern can be edited to modify its configuration (associated parts or script packages).

Pattern window view

The Pattern window displays the applicable virtual system patterns for the available virtual images. You can search for and add virtual system patterns. The Pattern window also includes information about the topologies that are associated with each pattern.

To edit the patterns on the PureApplication System, follow these steps:

1. Access the Workload console.
2. Select **Patterns** → **Virtual Systems** (Figure 2-3).

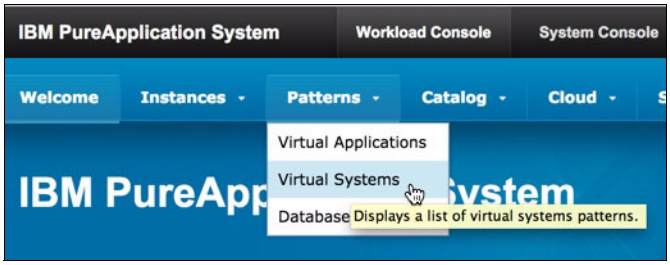


Figure 2-3 Accessing virtual patterns to edit

The window shown in Figure 2-4 is displayed and includes a list of the patterns that are used in virtual systems.

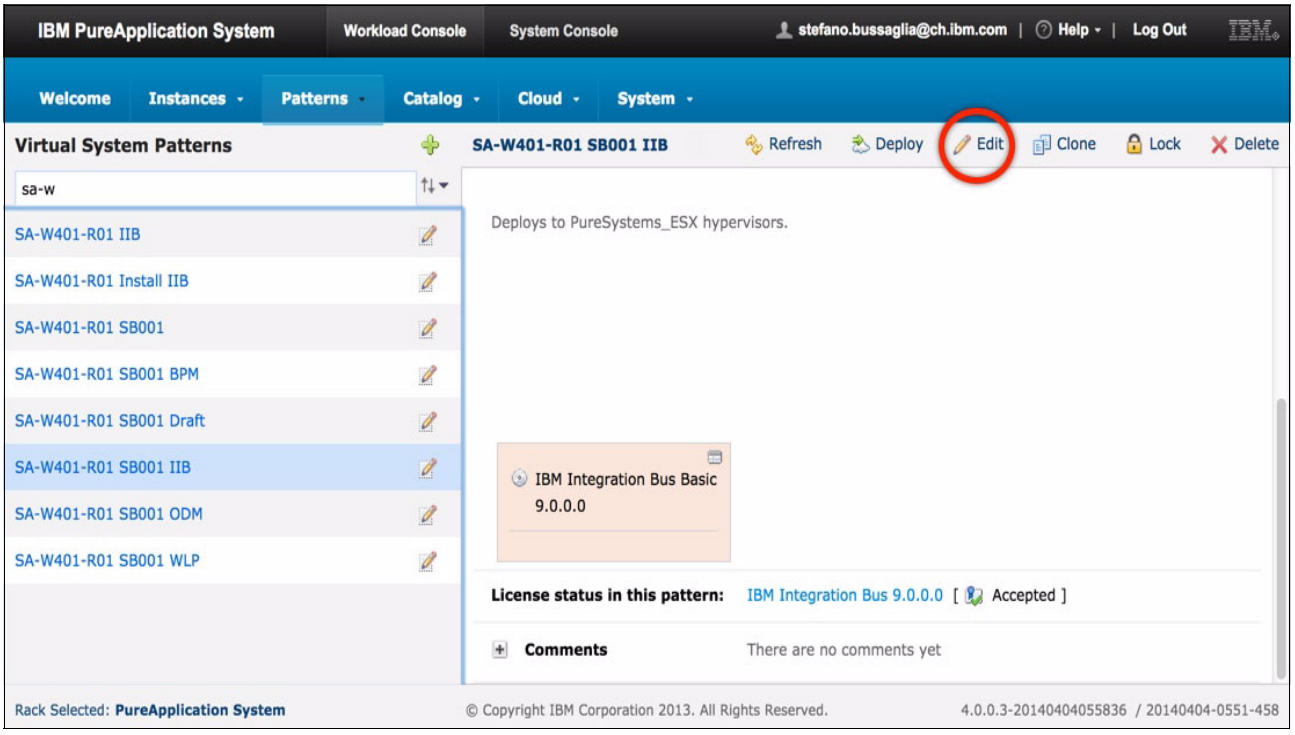


Figure 2-4 Pattern editor window

3. To modify the pattern, select the pattern, click **Edit**, and the pattern editor is displayed.

For more information about editing virtual system patterns, see Chapter 5 in *Adopting IBM PureApplication System V1.0*, SG24-8113, available at:

http://www.redbooks.ibm.com/abstracts/sg248113.html

IBM Integration Bus

The concept of *parts* is discussed in 2.1.3, “Virtual system patterns” on page 7. There are two different parts of the IBM Integration Bus: Basic and Advanced. The business scenario described in this paper (see Chapter 1, “Introduction” on page 1), uses the Basic part, which corresponds to the Standard edition of the Integration Bus. Additionally, it uses a DB2 database component to store the claim data from the business scenario (Figure 2-5).

Figure 2-5 Integration Bus and DB2 database used in the business scenario

For more information about the Integration Bus Hypervisor Edition, see:

http://www-03.ibm.com/software/products/en/integration-bus-hypervisor-edition

Web application

The web application uses WebSphere Liberty Profile and an IBM HTTP Server. The WebSphere Application Server Hypervisor Edition includes components of the Liberty profile and IBM HTTP servers (Figure 2-6).

Figure 2-6 Liberty server and HTTP server used for the web front end

For the web application, the virtual application pattern could have also been used, but the Liberty profile component used in virtual system pattern was chosen for this business scenario. The reason is that, with the virtual system pattern, you can choose the application server profile, whereas the virtual application pattern (VAP) offers only the base application server profile.

2.1.4 Virtual application pattern

With a virtual application pattern, the user can focus only on application requirements and not on the underlying infrastructure that is needed to support the runtime environment. You create the virtual application from the required components, define dependencies among the components, and apply policies to automatically manage the behavior of the application after it is deployed. By using this approach, you can use the VAP elements (such as components, links, and policies) that do not directly map to virtual machines.

A virtual application is defined by a virtual application pattern. It is a complete set of platform resources that fulfills the business need, including:

- Web applications
- Databases
- User registries
- Messaging services
- Transaction processes

Each virtual application pattern is associated with a pattern type, which is a collection of plug-ins that provide these resources and services for a particular business purpose in the form of components, links, and policies. The pattern types, product extensions of the cloud system, and the types of virtual application that you build depend on the pattern types that you enabled.

The differences between virtual systems and virtual applications related to web applications are discussed at:

http://www-01.ibm.com/support/knowledgecenter/SSAJ7T_1.0.0/com.ibm.websphere.patterns.doc/ae/cins_patterns_diff.html

For more information about virtual application patterns, see Chapter 1 in *Cloud Computing Patterns of Expertise*, REDP-5040, available at:

http://www.redbooks.ibm.com/abstracts/redp5040.html?Open

To view virtual application pattern types on the IBM PureApplication System Workload Console, select **Cloud** → **Pattern Types** (Figure 2-7).

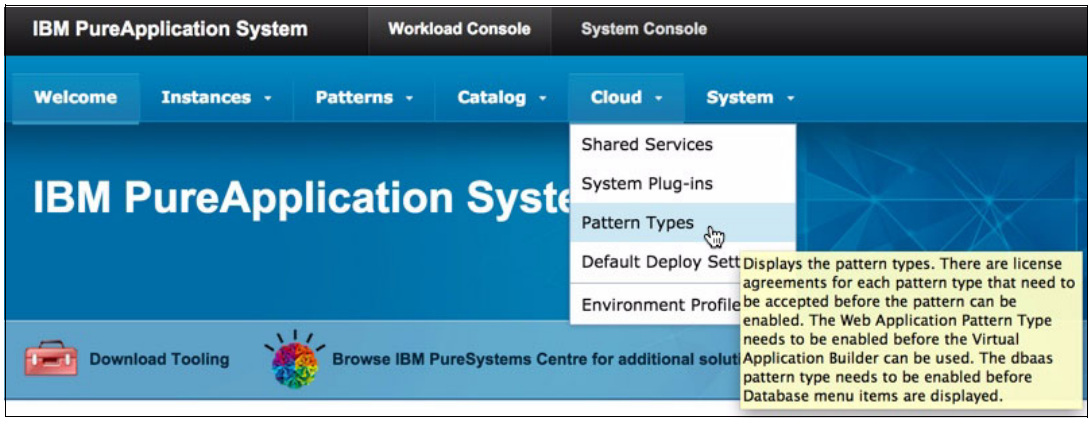


Figure 2-7 Virtual application pattern types

Editing virtual application patterns

To view virtual application patterns on the PureApplication System Workload Console, complete these steps:

1. Select **Patterns** → **Virtual Applications** (Figure 2-8).

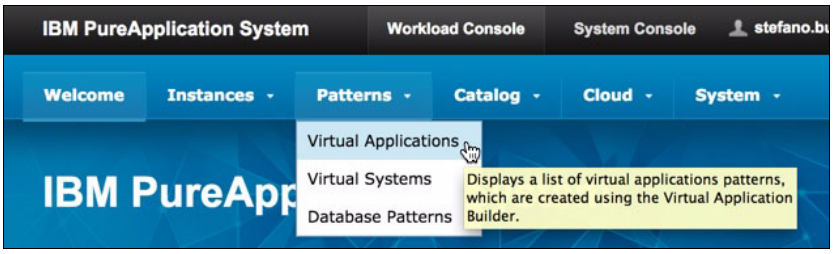


Figure 2-8 Viewing virtual application patterns

2. Click the **Green Plus Sign** to create a new virtual application pattern (Figure 2-9).

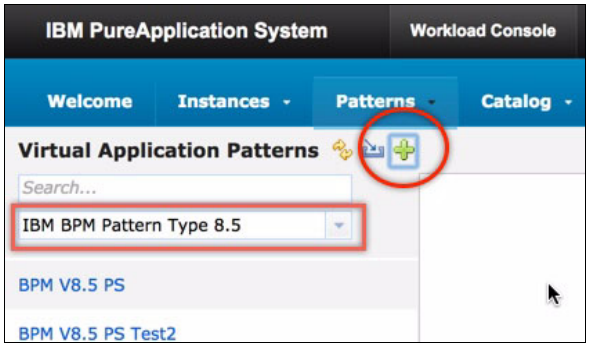


Figure 2-9 Creating a virtual application pattern

Tip: Select the pattern type you want in the drop-down menu to display existing patterns or create a new pattern.

For example, for a new IBM Business Process Manager pattern, select the pattern type, and then select one of the templates for building the environment. The example business scenario uses the template *Process Application* as shown in Figure 2-10.

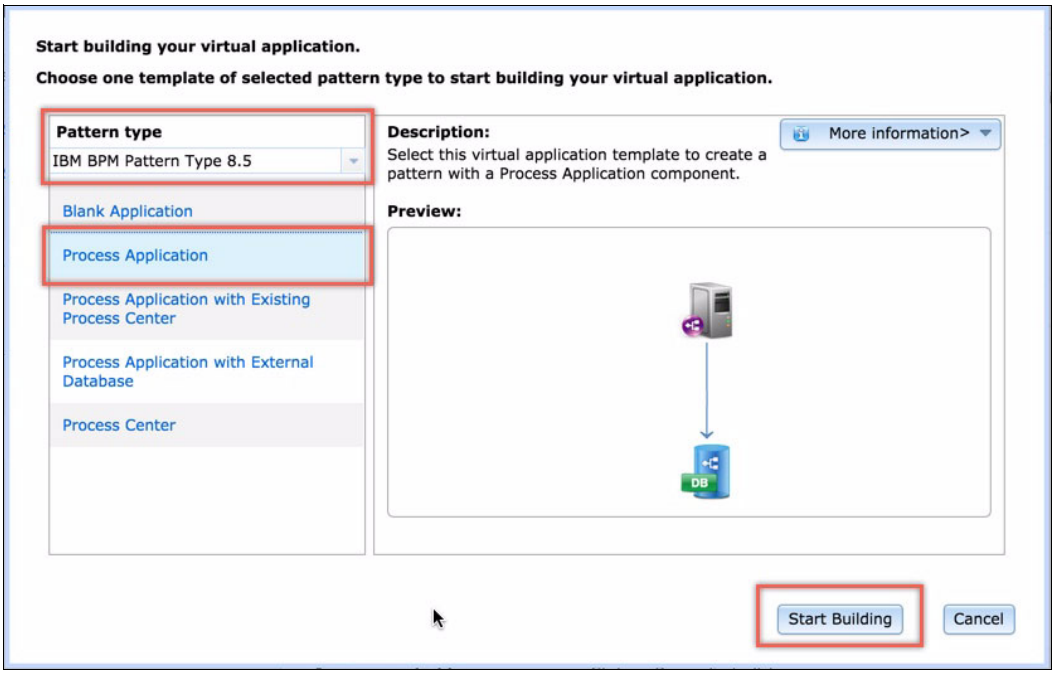


Figure 2-10 Creating a Virtual Application pattern from template

3. Click **Start Building** to open the Virtual Application Builder (Figure 2-11).

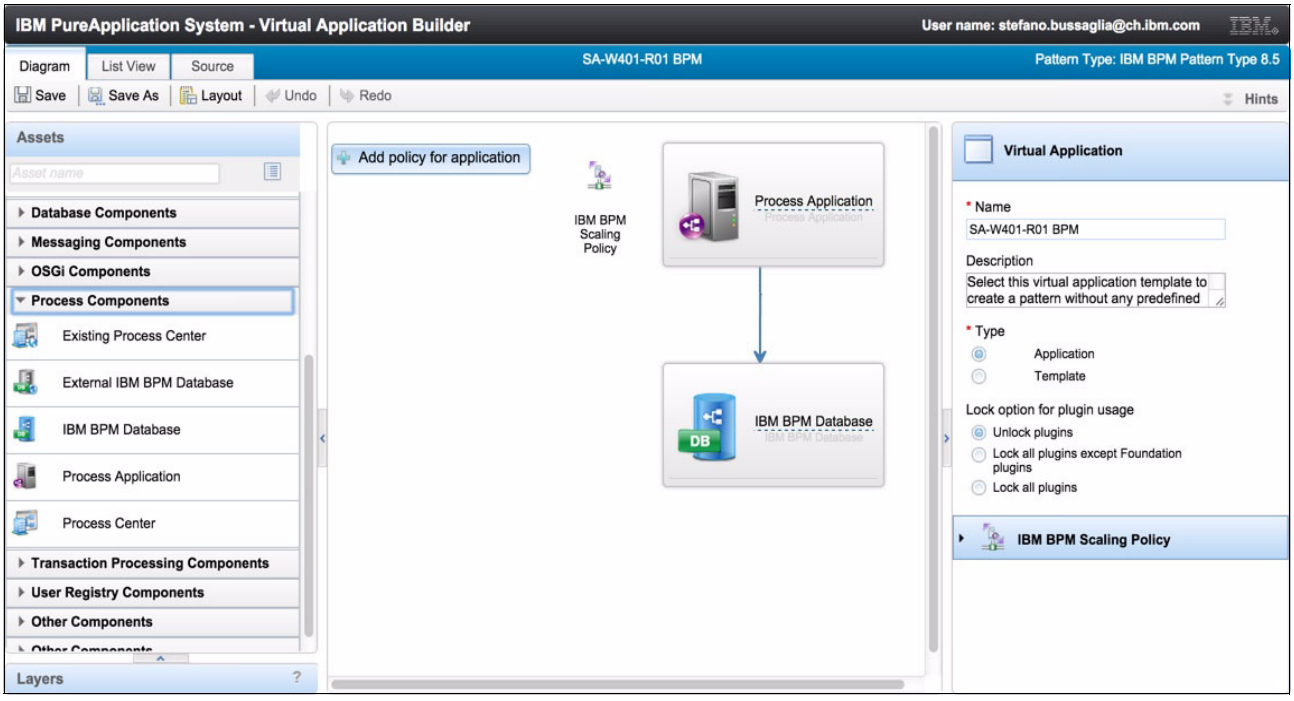


Figure 2-11 Virtual Application Builder

To edit the virtual application pattern, you can also use the Virtual Application Builder by selecting your virtual application pattern and clicking **Open** (Figure 2-12).

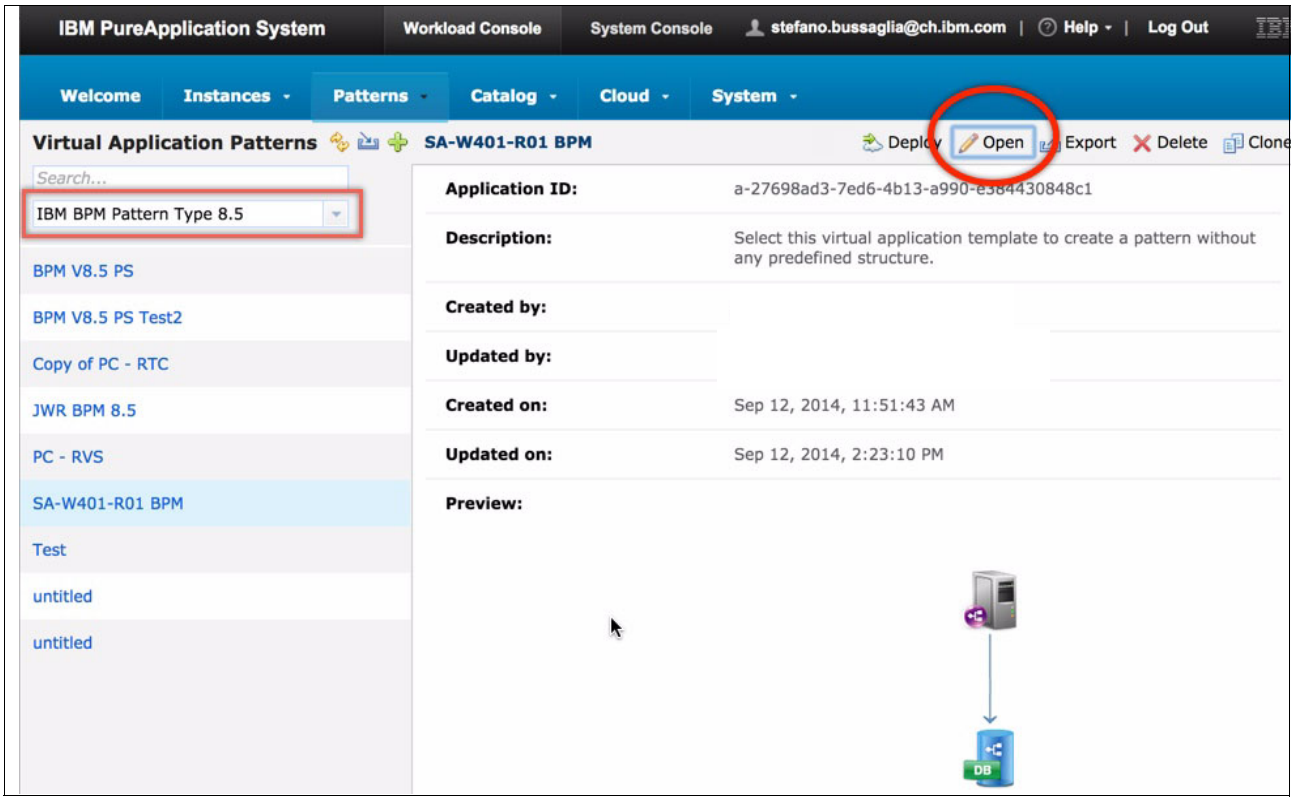


Figure 2-12 Open option of virtual application builder

IBM Business Process Manager Process Center

Business Process Manager (IBM BPM) application pattern is a comprehensive and consumable business process management platform. It provides visibility into, and management of, your business processes. It includes tools (Process Designer) and run times (Process Center and Process Server) to model, assemble, deliver, monitor, and manage the business processes confidently. Use only the necessary runtime components, such as *Process Application* and the *IBM BPM Database* (Figure 2-13).

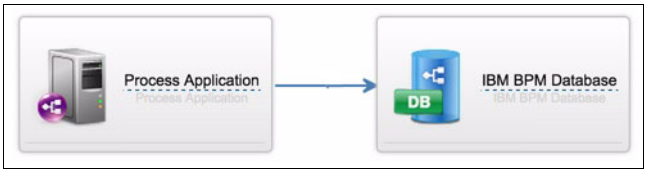


Figure 2-13 The IBM BPM application pattern components

For more information about IBM Business Process Manager patterns, see *Leveraging Virtual Application Patterns for Business Process Management Deployment*, TIPS1022:

http://www.redbooks.ibm.com/abstracts/tips1022.html?Open

IBM Operational Decision Manager Decision Center

The IBM Operational Decision Manager Application Pattern provides application models that enable the Operational Decision Manager (ODM) environment to run on a private cloud. It is optimized to run in a PureApplication System environment.

IBM ODM application pattern is a configuration of Operational Decision Manager V8.5. From the Decision Center, you can centrally control business decision deployments for your organization. IBM ODM application pattern provides capabilities that are configured for typical decision management projects. For ODM, the example uses the Decisions run time and the Operational Decision Manager database components (Figure 2-14).

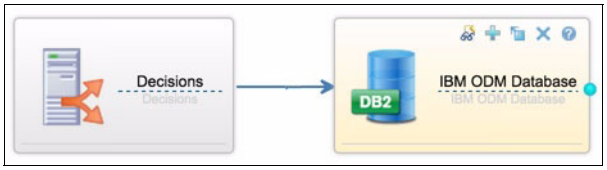


Figure 2-14 Components of the ODM application pattern

Mobile

The mobile application platform pattern type is composed of the IBM Web Application Pattern and the IBM Mobile Application Platform Pattern. The IBM Mobile Application Platform Pattern provides a various number of components, including those shown in Figure 2-15.

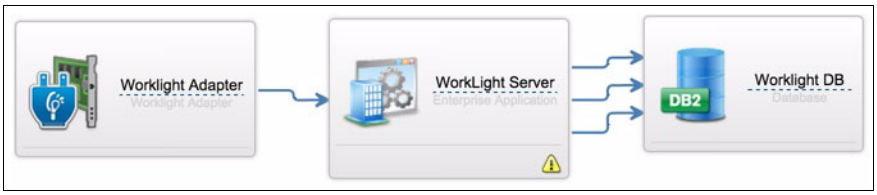


Figure 2-15 Worklight pattern used for mobile

For the business scenario described in this paper (see “Introduction” on page 1), the components Worklight Adapter, Worklight Server, and Worklight DB were chosen as shown in Figure 2-15.

2.2 Loading patterns into the system

Before working with patterns, make sure that they are in the PureApplication System catalog. The PureApplication System is shipped with pre-entitled software with unlimited usage in the same rack.

You can import patterns and OVAs into the PureApplication System using a GUI on the HTTP server *only*, as FTP, SFTP, and NFS are currently not supported.

2.2.1 Pre-optimized and pre-entitled software

In a licensed PureApplication System W1500 (Intel) environment, you have full entitlement to run the following software:

- Full stack monitoring (hardware, OS, entitled middleware)
- Virtualization and virtualization management
- Tooling for creating patterns
- Virtual System Patterns:
  - IBM OS image for Red Hat Linux Systems v2.0.0.3 (RHEL 64-bit v6.3)
  - IBM WebSphere Application Server Hypervisor Edition v7.0.0.29 with IMP[1]
  - IBM WebSphere Application Server Hypervisor Edition v8.0.0.6 with IMP[1]
  - IBM WebSphere Application Server Hypervisor Edition v8.5.5 with IMP[1]
  - IBM DB2 9.7 FP8 Enterprise Server Edition HV[1]
  - IBM DB2 10.1 FP2 Enterprise Server Edition HV[1]
  - IBM DB2 10.5 Advanced Workgroup Server Edition HV[1]
  - Automation Framework HV* (for migrating applications)
- Virtual application patterns:
  - Java Pattern v1.0.1 (64-bit Java 7 SDK)
  - IBM Workload Deployer Pattern for Web Applications v1.0.1 (with Web Application Server v7)
  - IBM Web Application Pattern v2.0.1 (with Web Application Server v8)
  - IBM Transactional Database for Cloud v1.1.0.7 (DB2 9.7 FP8, 10.1 FP2 & 10.5)
  - IBM Data Mart for Cloud v1.1.0.7 (with DB2 9.7 FP8, 10.1 FP2 & 10.5)

2.2.2 Virtual system pattern

You can download any other Hypervisor image from IBM Passport Advantage® and import it into the IBM PureApplication System.

From the console, you can import the image from an HTTP server or an FTP server. Otherwise, you can import the image by using a command-line interface (CLI). For more information about using CLI, see Chapter 5 in the book, *Adopting IBM PureApplication System V1.0*, SG24-8113 at:

http://www.redbooks.ibm.com/abstracts/sg248113.html

With CLI, the image can be imported from a local hard disk drive. The following is an example of a Linux script for importing an OVA file with CLI:

```
$sh pure –h <host> -u <user> -p <password> --acceptcert
>>> deployer.virtualimages.create('/Images/IBM_Integration_Bus_9.0.0.0.ova')
```

Tip: To upload, export, or import patterns (VSP, VAP, or OVAs) with sizes greater than 2 GB, always use the CLI tool.

[1] Intelligent Management Pack, HV-Hypervisor Edition

To import an image, follow these steps from the UI console:

1. To import a new Hypervisor image on the PureApplication System Workload console, select **Catalog** → **Virtual Image** and click the **Green Plus Sign**, as shown in Figure 2-16.

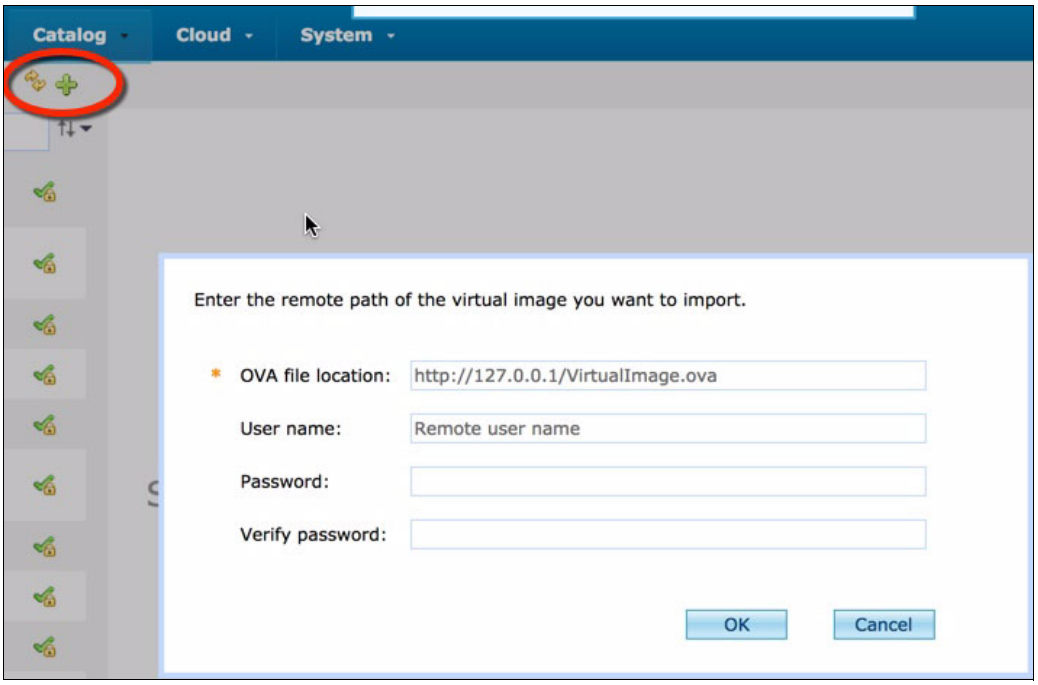


Figure 2-16 Importing the image

After the import, the system automatically registers the new image on the hypervisor. The virtual image is visible in the catalog and has a status of *"License not accepted"* (Figure 2-17).

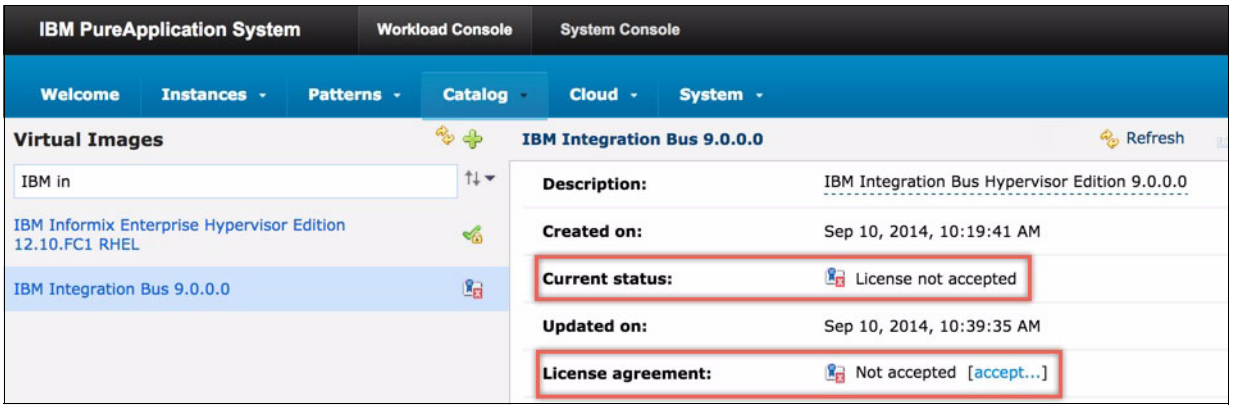


Figure 2-17 Updated catalog

2. Accept the license term conditions (Figure 2-18) to continue.

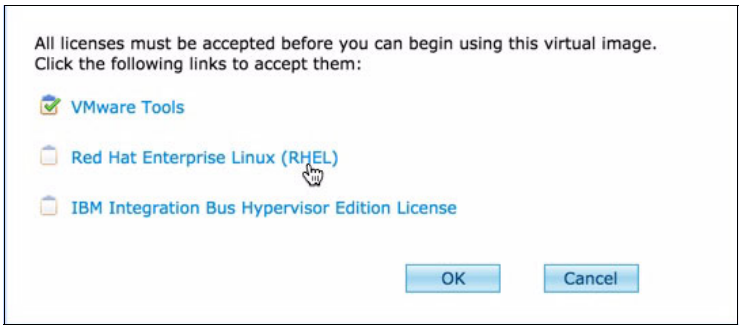


Figure 2-18 Accepting license terms

2.2.3 Virtual application pattern

To import the pattern types for the virtual application, complete these steps:

1. In the PureApplication System Workload console, select **Cloud** → **Pattern Types** (Figure 2-19).

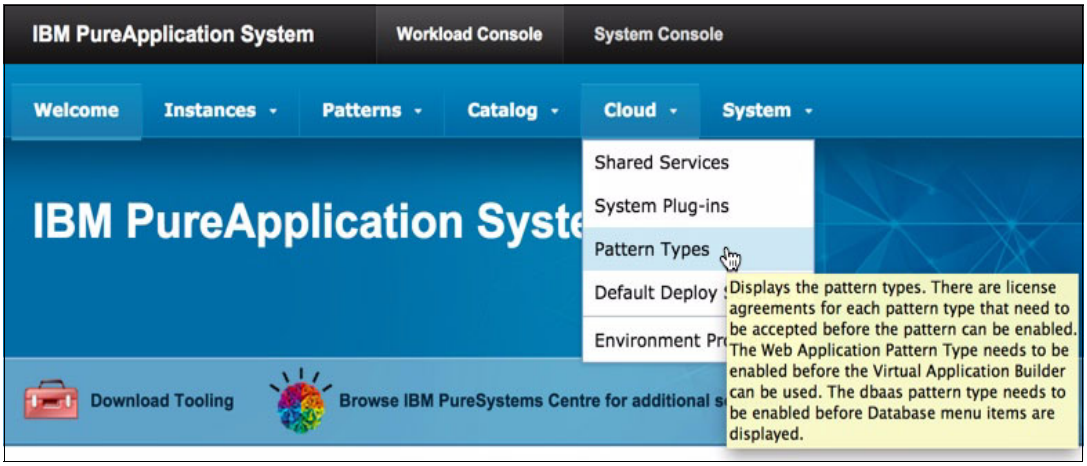


Figure 2-19 Pattern types

2. Click the **Green Plus Sign**, which shows that you have two options to import, either locally (Figure 2-20) or from a remote location (Figure 2-21 on page 18).

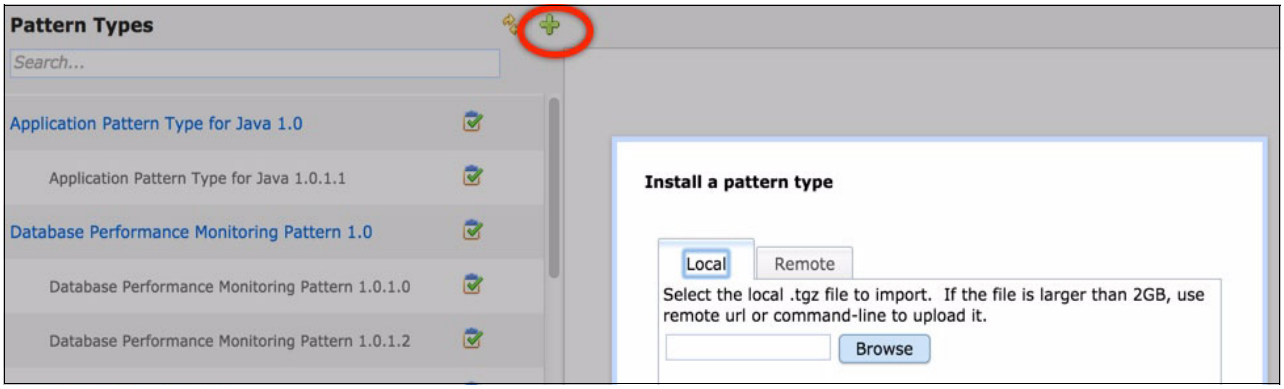


Figure 2-20 Selecting where to import from

3. Select *Local* to import the patterns, or use *Remote* to import from an HTTP server (Figure 2-21).

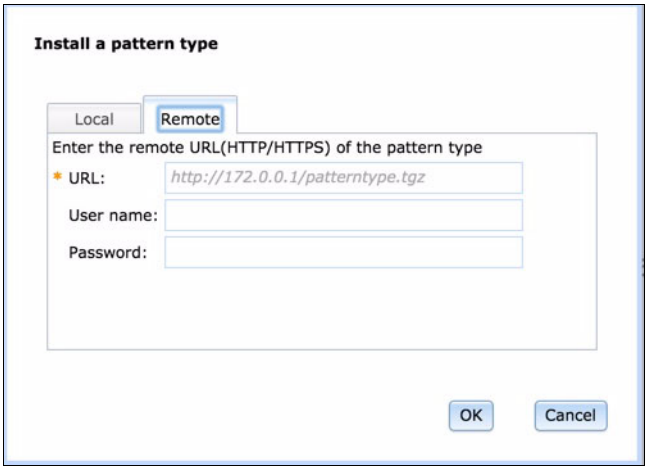


Figure 2-21 Defining a remote location to import from

Note: Most pattern types are shipped with an installation script that you can run from the command line.

For more information about importing and installing pattern types, see *Installing the IBM Mobile Application Platform Pattern Type* at:

http://pic.dhe.ibm.com/infocenter/wrklight/v5r0m6/topic/com.ibm.worklight.help.doc/pureapp/t_pureapp_installing_mobile_application_platform_pattern_type.html

3

Integrating the solution in the cloud

This chapter describes the cloud environment that hosts the solution. A private cloud provided by an IBM PureApplication Systems machine is used. The concepts of cloud resources (including cloud group, IP group, and environment profile) are introduced, specifically regarding this type of Expert Integrated System.

The descriptions in this chapter continue to expand on the concept of patterns. This chapter explains how the different patterns (described in Chapter 2, “Patterns of expertise” on page 5) have been customized to build the middleware infrastructure necessary for the solution.

Further described in this chapter, is the IBM PureApplication Service on SoftLayer as an IBM offering for the public cloud. This is done because it shares the technology used in the PureApplication Systems. The PureApplication Service can house, without any modification, the artifacts that are created for PureApplication Systems.

Finally, this chapter explains how to integrate the IBM Rational UrbanCode technology in a cloud environment, and also demonstrates how to use it to deploy the solution to PureApplication Systems.

This chapter includes the following sections:

- System requirements
- Cloud resources
- Create the necessary patterns for the solution
- Shared Services
- Rapid application deployment using IBM UrbanCode Deploy
- Deploying the solution
- Managing the Virtual Application instances

3.1 System requirements

Table 3-1 provides the minimum requirements of the system for the software used in this chapter's solution. The details of the requirements are described throughout this chapter.

When you deploy a pattern into a cloud group, the system ensures that sufficient memory and processor resources exist for each virtual machine in the pattern.

Review Table 3-1 before starting your solution to ensure that your system has the minimum requirements.

Table 3-1 List of Virtual System Patterns and Virtual Application Patterns

Middleware	CPU	Memory	OS
Liberty profile server 8.5.5.1	1	2048	Linux
IBM HTTP servers 8.5.5.1	1	2048	Linux
IBM Integration Bus Basic 9.0	1	2048	Linux
Process Center deployment manager 8.5.0.1	1	3072	Linux
Process Center custom nodes 8.5.0.1	2	4096	Linux
IBM HTTP Server for Process Center 8.5.0.1	1	2048	Linux
Process Center database 8.5.0.1	2	4096	Linux
ODM deployment manager 8.0.1	2	2048	Linux
Decision Center custom node 8.0.1	2	4096	Linux
ODM DB2 database 8.0.1	2	4096	Linux
IBM Worklight 6.2	2	2048	Linux

3.2 Cloud resources

Before any deployment can be done, you must define the computing resources that hold the application, such as cores, memory, network, and storage. In PureApplication System, this is done by defining the cloud resources.

For more information about how to create a cloud resource in PureApplication Systems, see *IBM PureApplication System Best Practices*, SG24-8145, which is available at:

http://www.redbooks.ibm.com/abstracts/sg248145.html?Open

3.2.1 Defining cloud resources on the IBM PureApplication System

Cloud resources on the IBM PureApplication System are built in a private cloud, which does not go outside of the rack. You first need to define the cloud environment with IP Groups, a Cloud Group, and Environment Profiles.

The solution in this paper demonstrates how to deploy the Claim solution into the cloud, and therefore covers only the production environment. However, for resource isolation and security constraints, all other stages should be separated into their own cloud group, such as development, integration, or quality assurance.

IP addresses

Before defining the necessary cloud group, the IP addresses needed for the solution must be defined.

A range of IP addresses for each network zone must be provided based on of the size calculation. IP groups define the range of IP addresses that the infrastructure uses for different parts of the cloud. To create new groups of IP addresses on the IBM PureApplication System Console, use the following steps:

1. Select **Cloud** → **IP Groups**, as shown in Figure 3-1.

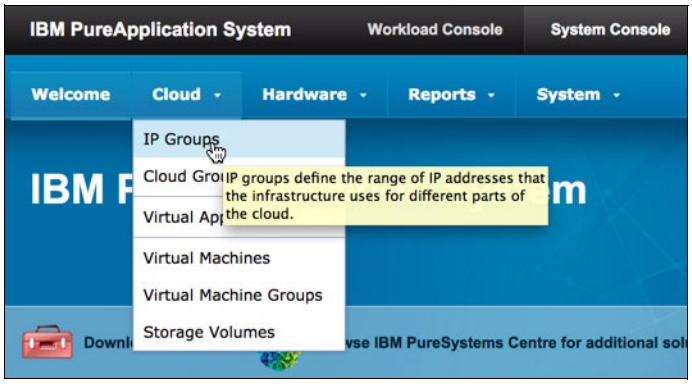


Figure 3-1 Opening IP groups

2. Create an IP Group by clicking the **Green Plus Sign** as shown in Figure 3-2.

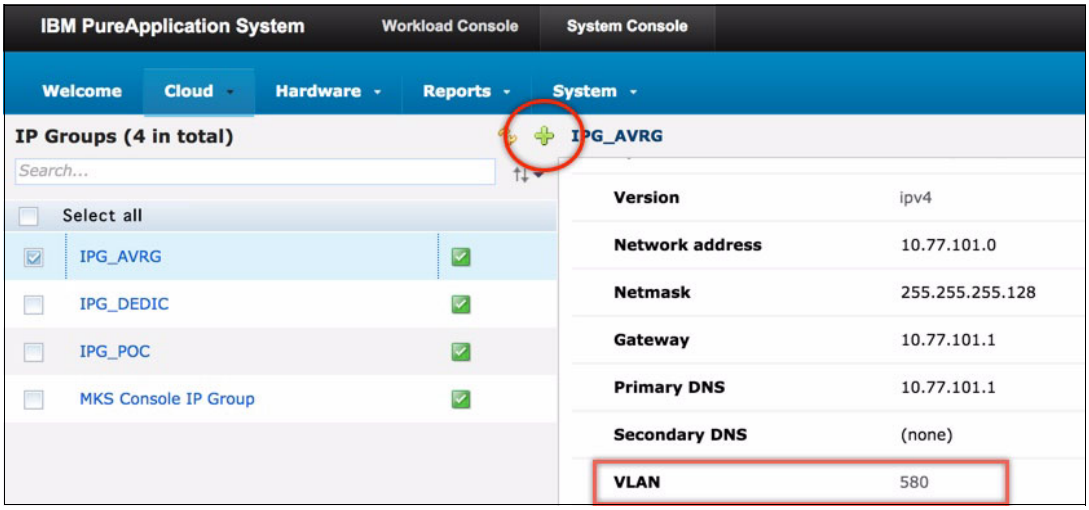


Figure 3-2 Creating an IP Group

3. Add your network information to create an IP address range.

Important: Separate the two IP Groups by the different VLANs as shown in Figure 3-2 on page 21 (VLAN 580 for the intranet and 480 for the Internet).

IP groups

This section addresses how to approach the IP groups and adding IP addresses. To do so, use the following steps:

1. Name both IP groups referencing the name for the solution (titles here are for example):
   - *SA-W401-R01* IP DMZ with VLAN 480
   - *SA-W401-R01* IP CORE with VLAN 580
2. Add the necessary IP addresses, as shown in Figure 3-3. IP addresses can be added by using a specific range or by entering the host names.

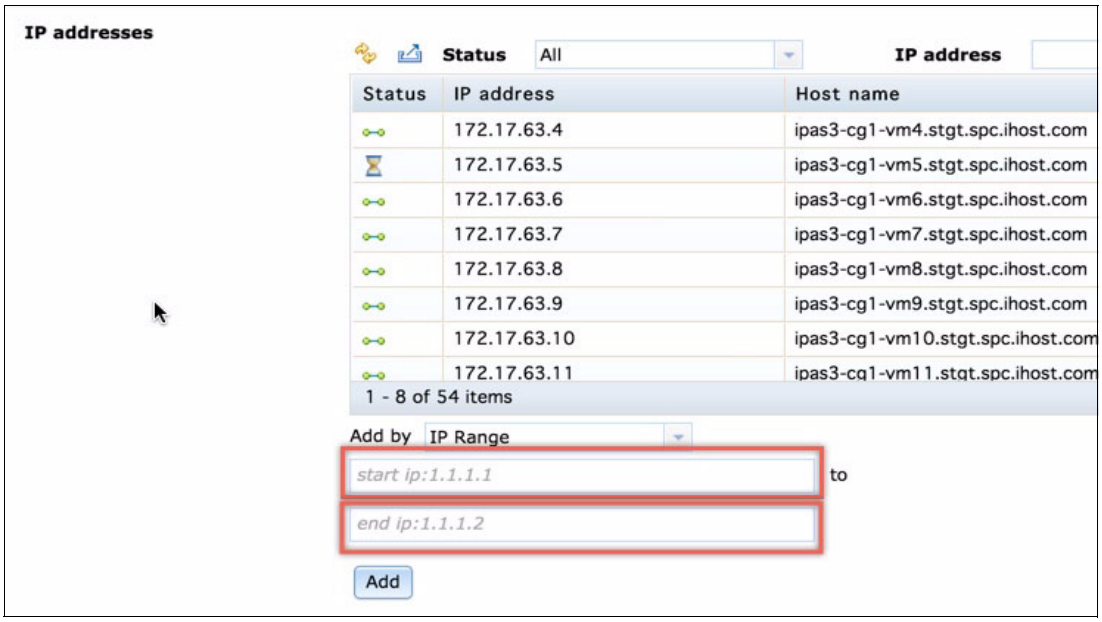


Figure 3-3 Adding IP addresses using a specific range

Cloud groups

In the cloud group, you allocate the computing resource of PureApplication System, such as Compute Nodes.

To create and define a cloud group from the IBM PureApplication System workload console, use the following steps:

1. Select **Cloud** → **Cloud Groups**, as shown in Figure 3-4. This panel organizes compute nodes and IP groups into cloud groups.

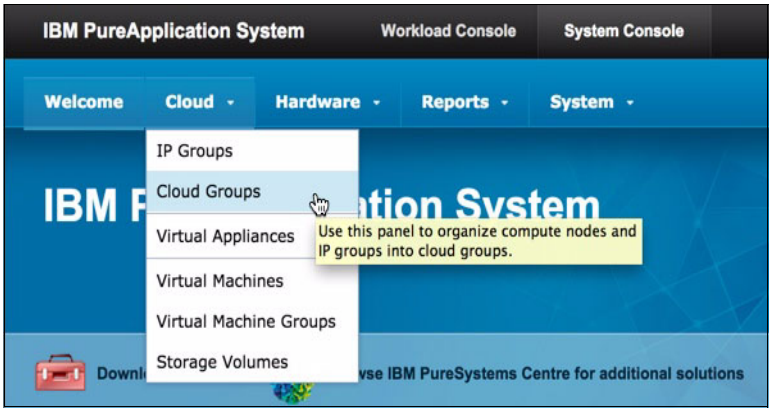


Figure 3-4 Opening cloud groups

2. When defining a new Cloud Group, you need an IP Group that is not assigned to another Cloud Group and unused Compute Nodes available to be attached to the group. For assigning available IP Groups and Compute nodes to the Cloud Group, see Figure 3-5. To define a new cloud group, click the **Green Plus Sign**.

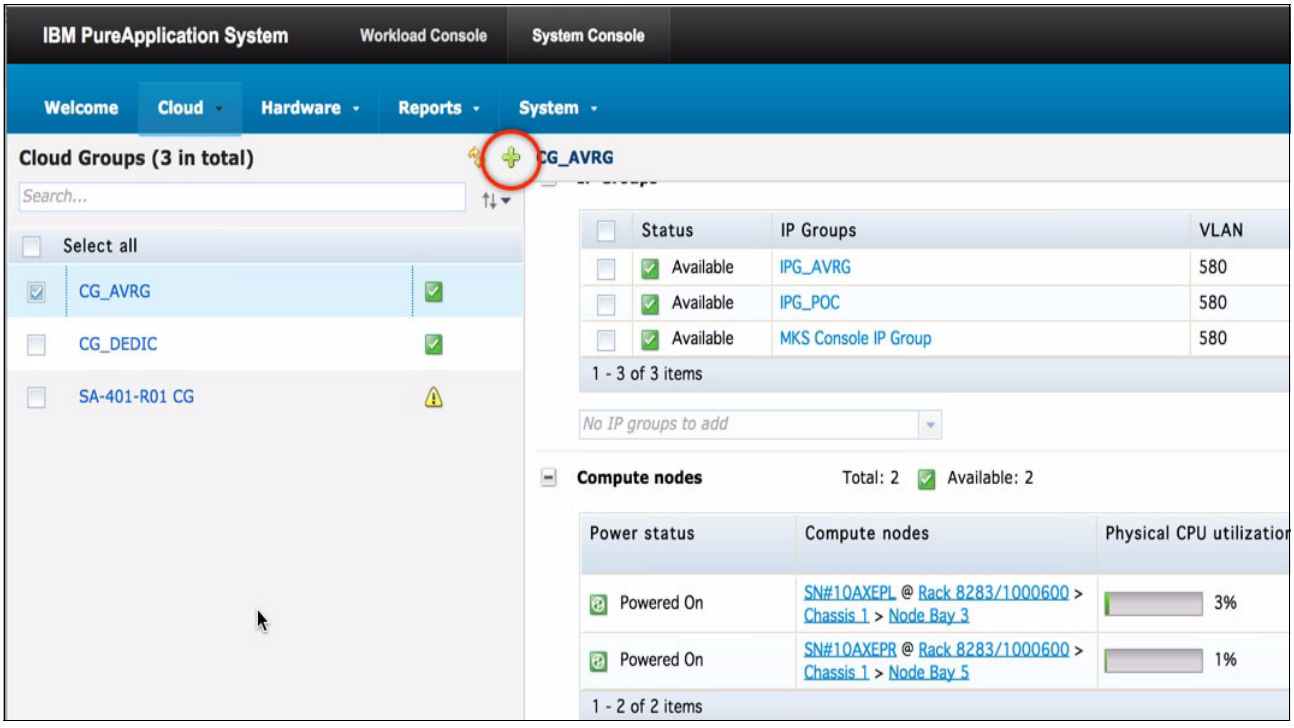


Figure 3-5 Defining a cloud group

3. Name the cloud group for the solution (the following title is an example):

```
SA-W401-R01 CG
```

For more information about how to create a Cloud Group in PureApplication Systems, see *Adopting IBM PureApplication System V1.0*, SG24-8113, which can be found at:

http://www.redbooks.ibm.com/abstracts/sg248113.html

Note: In PureApplication, you can isolate the different network zones, such as internet and intranet, by separating them into different Cloud Groups.

This solution has only one Cloud Group, and the network zones are separated by defining two IP Groups in different VLANs.

Environment Profiles

The environment profiles describe the deployment settings for the platform. You can specify an environment with multiple clouds, and configure specific resources within those clouds.

To create environment profiles, configure on different zones, and configure fields, use the following steps:

1. To create a environment profile on the IBM PureApplication System Workload Console, select **Cloud** → **Environment Profiles**, as shown in Figure 3-6.

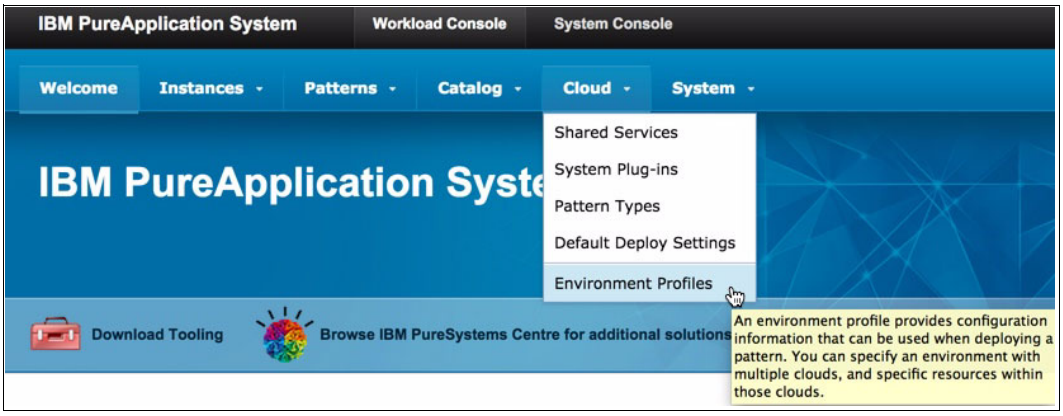


Figure 3-6 Opening an environment profiles

2. Assign a name to distinguish the environment file types (the following names are examples):
   - *SA-W401-R01* EP
   - *SA-W401-R01* EP Pattern

   Note: When creating the two different environment profiles, you can create the first one and then make a clone for the second profile.

See the example settings in Figure 3-7. Note the highlighted fields available that can be specified according to your environment.

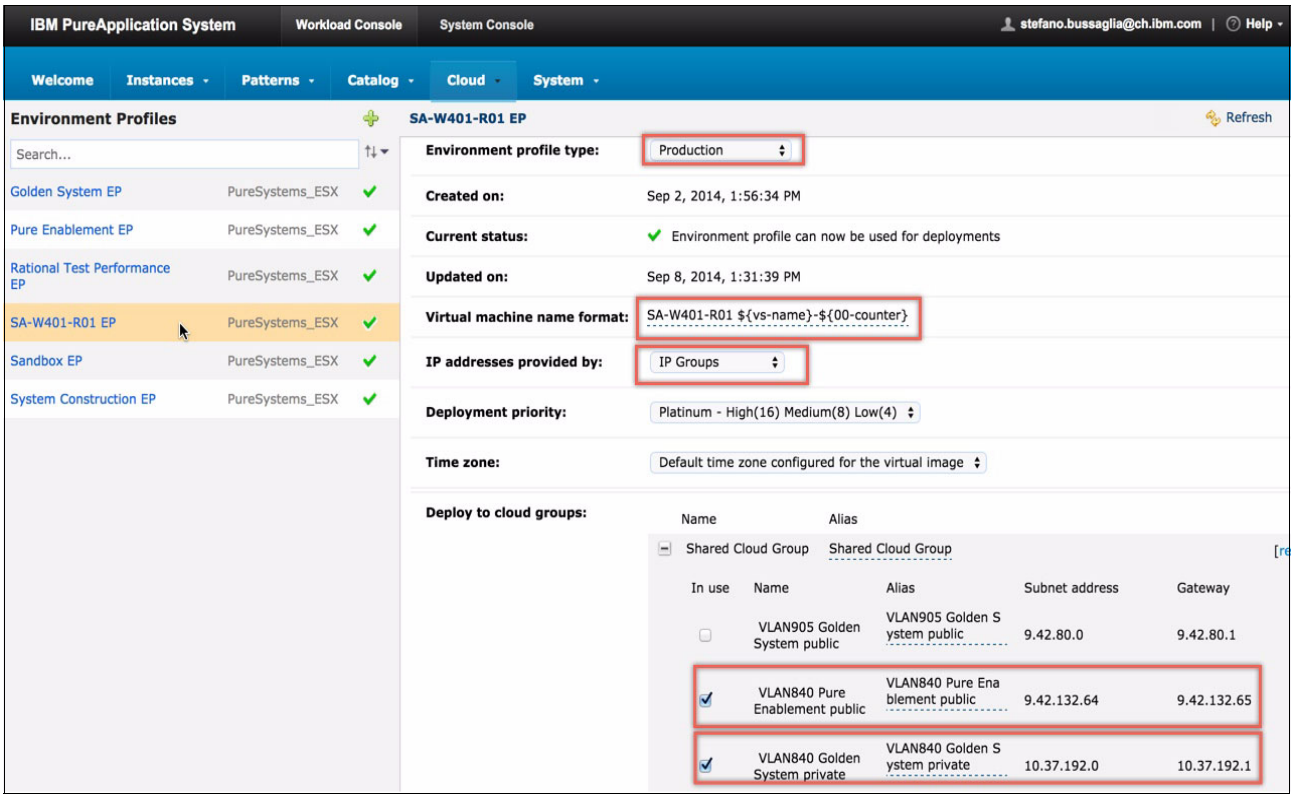


Figure 3-7 Environment profile settings

Fields that are used in Environment Profile

As shown in Figure 3-7, there are many fields to define. The following list notes those fields (and gives suggested changes to the default settings) to set up the environment for the claim solution:

- Environment profile types available include: **All**, **Development**, **Test**, **Quality Assurance**, **Performance**, **Research**, **Production**, and **Pre-Production**. The default is **All**. Select **Production**.

  Tip: If you set the profile type to a specific value, you have the advantage to select only from a certain range of items at deployment time.

- Set a virtual machine name format such as, ***SA-401-R01-${vs-name}-${00-counter}*** The following list explains the name structure:
  - ${*hostname*}: Resolves to the host name of the virtual machine
  - ${*vs-name*}: Resolves to the name of the virtual system; this is not unique, so it must be used with one of the other variables
  - ${*x-counter*}: Resolves to an incremented value managed by IBM Workload Deployer; 'X' represents the number of digits in the counter

- IP addresses provided by: available options are **IP Groups** or **Pattern Deployer**. The default is **IP Groups**.

Because the examples used in this book need two different ranges of IP Groups to deploy in different network zones, **IP Groups** is selected.

Note: Step 2 “Assign a name to distinguish the environment file types (the following names are examples):” noted to create two profiles (*SA-W401-R01* EP and *SA-W401-R01* EP Pattern). For the second Environment Profile (*SA-W401-R01* EP Pattern), **Pattern Deployer** was selected, because you can specify the host name and the wanted IP address.

- Deploy to cloud groups: From all defined IP Groups, select a range of internal and external IP addresses.

Note: For the second Environment Profile (*SA-W401-R01* EP Pattern), select only the internal range for the IP Group.

3.2.2 Defining resources on IBM PureApplication Service on SoftLayer

Before describing how the solution can be replicated on SoftLayer, this chapter briefly explains what IBM PureApplication Service on SoftLayer is and why IBM decided to move in that direction.

IBM PureApplication Service on SoftLayer provides new capabilities to take patterns that have been developed for IBM PureApplication System since 2012, and make them available on a public cloud. This offeringgives customers the option to either choose between (or have both) a cloud infrastructure on-premises (at home) or off-premises (hosted externally). The correct choice for your environment depends on the type of workload the systems must manage. For instance, you can decide to use IBM PureApplication Service on SoftLayer for development and test purposes. You might at first maintain the production on-premises (on either a small cloud environment on IBM PureApplication Service on SoftLayer or on the IBM PureApplication System). Later, you can move from SoftLayer to the IBM PureApplication System when the environment becomes more complex. An interesting usage of the IBM PureApplication Service on SoftLayer offering is when a client has a private cloud on-premises and needs more capacity. The additional resources can be taken from the public cloud, maximizing the capabilities of IBM PureApplication Services on SoftLayer. In such a situation, it is more about where a customer can easily and quickly get what they need to satisfy either the workload or the scalability requirements of their applications.

It is important to know that all the patterns created or customized on an IBM PureApplication System on-premises work unchanged on the IBM PureApplication Service on SoftLayer. There is complete transparency between the two cloud infrastructures, and you can move back and forth as needed. This is valid for both the created patterns and for those that you have purchased.

IBM made available on PureApplication Service on SoftLayer both the patterns technology and some of the infrastructure management coming from IBM PureApplication System. This fluidity between systems means that IBM PureApplication Service on SoftLayer is now a strong programmable environment. Using a set of APIs, you can access SoftLayer infrastructure resources, which are available in the different data centers available around the world. Both the on-premises and off-premises solutions allow the identical interface and experience for developers and operations people.

To summarize, by adopting IBM PureApplication Service on SoftLayer you can run the applications that you have with the cloud economics you want and the isolation you need.

Differences

Although the similarities associate these two types of public and private clouds, there are some differences between IBM PureApplication Service on SoftLayer and IBM PureApplication System, as illustrated in Figure 3-8.

Category	PureApp Service	PureApp System
Pattern Portability	Yes – Same	Yes – Same
Management Platform	Same	Same
Management UI	Same	Same
Pre-entitled WAS, DB2	No	Yes
Compute Speed	1x	1.1x
Starting # of Cores	4 for Express 16 for Standard	32
Core increments	4 for Express 16 for Standard	32
Memory per core	6 gb/core for Express 16 gb/core for Standard	16 gb/core for Mini 32 gb/core for Ent.
Storage	1 TB or more of HDD	26.4 TB on Mini with balanced mix of HDD & SSD 54.4 TB on Enterprise with balanced mix of HDD & SSD
IOPS Estimate	1x	5x
Network Speeds	1 Gbps for Express 2 Gbps for Standard	10 Gbps/compute node

Figure 3-8 Comparison of PureApplication Service on SoftLayer and PureApplication System

The following are additional considerations when comparing the products:

- On PureApplication Service on SoftLayer, WebSphere Application Server and DB2 are not pre-entitled. This means that to load them, you must either use existing licenses or purchase new ones.
- PureApplication System offers high performance storage, networking, and compute resources not currently available on SoftLayer. These resources result in faster deployments, scaling, and overall performance. Depending on the application, performance differences of 10x can be evident.
- PureApplication Service offers a lower entry point and smaller growth paths that provide more flexibility. You can start with four cores and add increments of four cores, and your storage begins with 1 TB and can be increased by intervals of 1 TB of storage. The number of cores and amount of disk space that can be added is unlimited.

SoftLayer Account

After you purchase resources on PureApplication Service, you receive a SoftLayer Account for accessing the login page of the PureApplication Service as illustrated in Figure 3-9. You must install and properly configure a VPN client to access both the public and the private networks provided by PureApplication Service. In the example, OpenVPN is used.

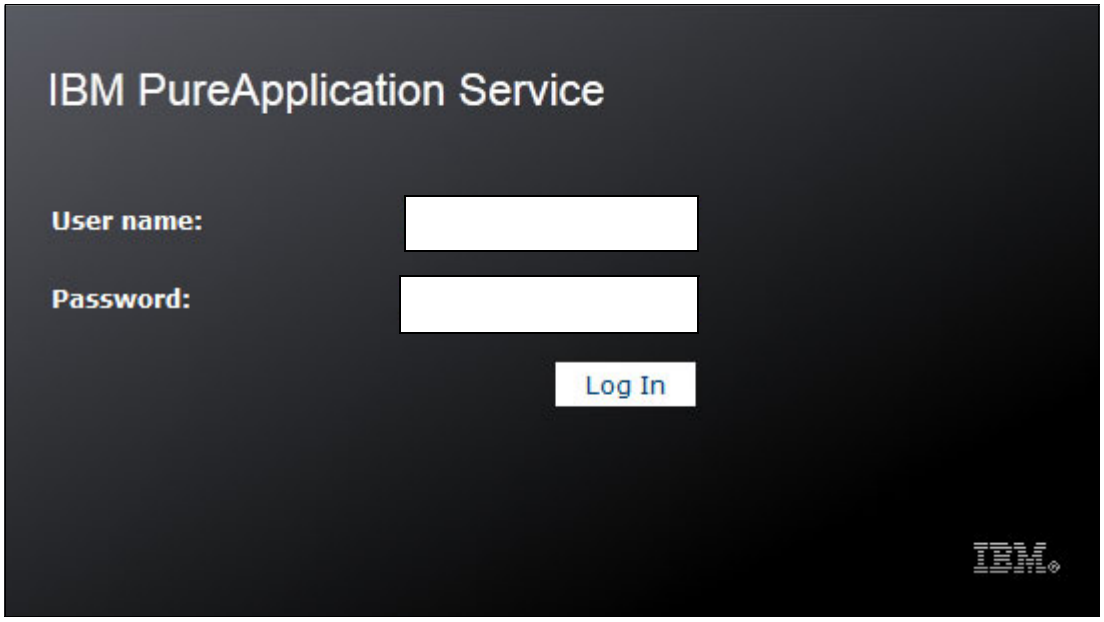


Figure 3-9 Login page

The login allows you to enter the PureApplication Service environment. If you are familiar with the PureApplication interface, you will notice that it is the same. After you create your own patterns (whether Virtual Systems or Virtual Applications), if your user ID has not been associated with a specific Environment Profile, you can deploy what you have created by choosing one of the Environment Profiles available on the system.

Resources

To ensure that the allocated resources are in line with what has been requested and purchased, select **Environment Profiles** after opening the Cloud tab in the administration interface.

Click the name of your Environment Profile and open the Environment Limits tab. In this tab, you can check the limits of the environment to which you are connected and how much of the resources you are using, as shown in Figure 3-10. You can also validate the products that you are entitled to by analyzing the list of the products loaded in the catalog of the system and verify that they match what you have purchased.

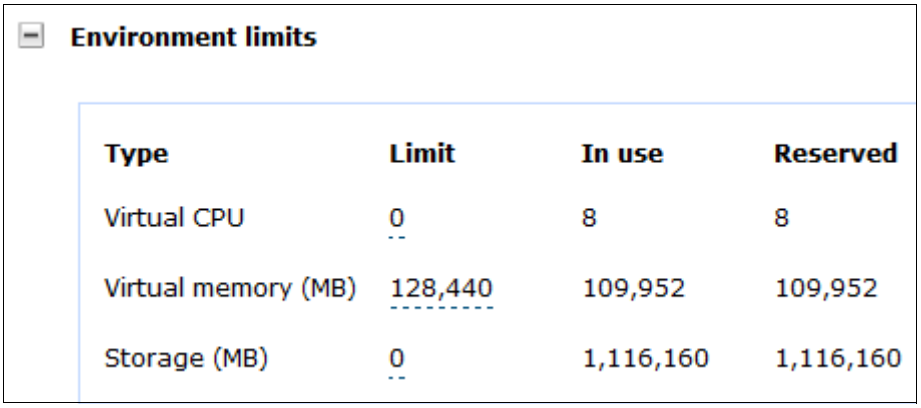


Figure 3-10 Example of environment limits

Patterns that are either created or purchased for being deployed and run in PureApplication Service or in PureApplication System can equally be used in either product without the need for any modification.

Every time that you need to run the patterns that you created for the PureApplication System in PureApplication Service, you can rely on the transparent migration ability that exists between the two system. You do not need to worry where the patterns were created and deployed. To move them, use the command line tool, which you can download either from the PureApplication Service Console or from a PureApplication System.

The *command line tool* can be downloaded by clicking **Download Tooling** in the Welcome tab of the PureApplication Service or of the PureApplication System (Figure 3-11). This tool allows you to *export* the patterns from one system and equally *import* them into the other.

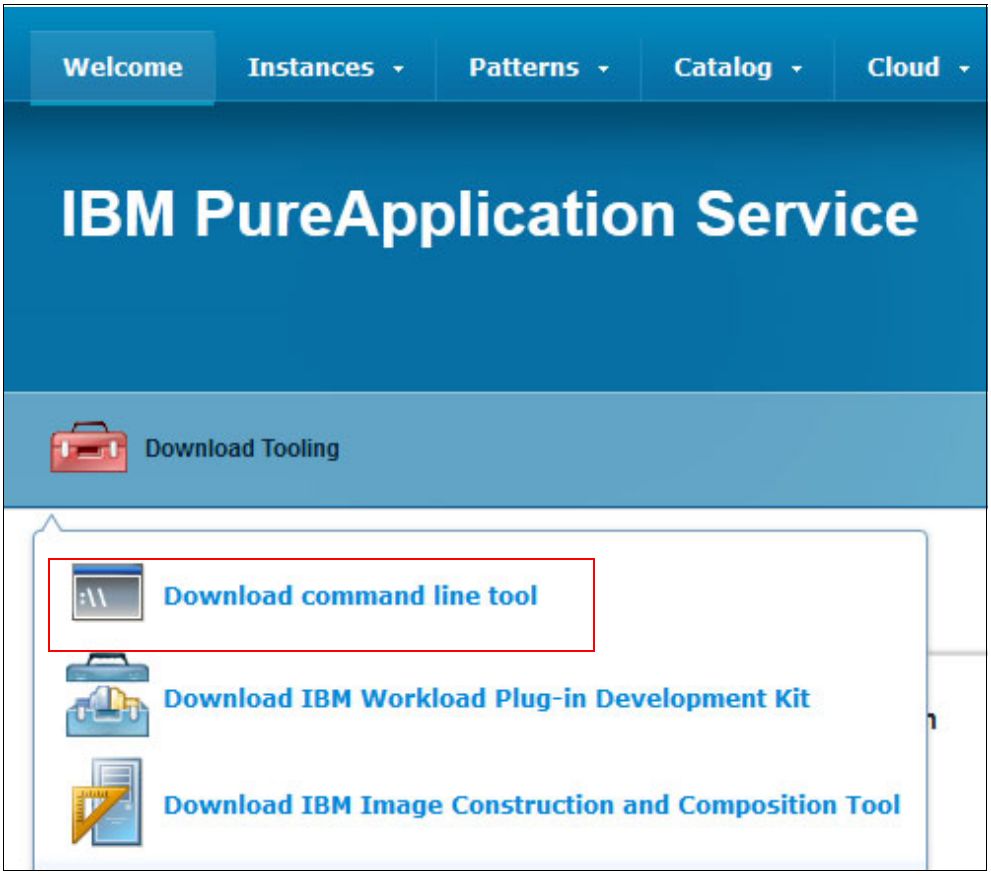


Figure 3-11 The PureApplication Service Welcome tab and Download icon.

3.3 Create the necessary patterns for the solution

This section covers information about building the solution with the patterns that IBM Platform as a Service (PaaS) offers. You can select the ones that best fit your solutions. This chapter uses example options for its solution.

3.3.1 Topology

For the reader's convenience, this topology reflects the possible pattern types that have been created from the solutions in this chapter. Figure 3-12 represents the integrated topology of the overall example solution.

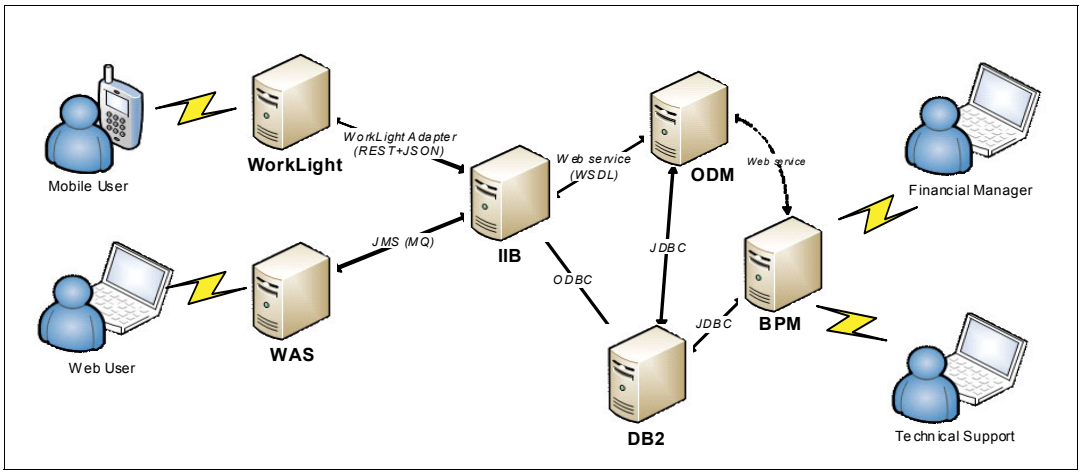


Figure 3-12 Integrated topology

The following solution components are illustrated in Figure 3-12:

- For the User Interface and Presentation Layer:
  - Worklight Server: For managing the requests coming from users who use mobile devices.
  - WebSphere Application Server and Liberty Server (the IBM PureApplication System solution, the example used the Liberty profile included in the WebSphere Application Server license): For managing the requests coming from users who use browsers.
- For the Business Logic Layer:
  - Integration Bus: For managing all the custom integration flows with associated integration logic.
  - Operational Decision Manager (by using Decision Server component): For running all the decision rules that are used in the solution.
  - Business Process Manager (by using Process Center component): For running the business processes defined in the solution.
  - DB2: For hosting the database for the overall solution.

The following list maps the previous listed components to the patterns created for the PureApplication System cloud environment:

- For the User Interface and Presentation Layer:
  - SA-W401-R01 Mobile: A Virtual Application Pattern based on Worklight Server.
  - SA-W401-R01 WLP: A Virtual System Pattern based on WebSphere Liberty Profile.
- For the Business Logic Layer:
  - SA-W401-R01 IIB: A joint Virtual System Pattern based on IBM Integration Bus (IIB) and DB2.

- SA-W401-R01 ODM: A Virtual Application Pattern based on Operational Decision Manager (ODM) for the Decision Center component that can run on multiple virtual machines.
- SA-W401-R01 BPM: A Virtual Application Pattern based on Business Process Manager (IBM BPM).

Mobile, ODM, and IBM BPM use their own DB2 databases to store the internal data necessary for the proper functioning of the products and of their features. For this example, the database for the overall solution is the one accessed by the IIB pattern. Therefore, the DB2 pattern is included in the IIB pattern, thus obtaining a multiproduct joint pattern. This choice is due to the core role that the IIB component plays within the overall architectural design.

To simulate a real production environment, the patterns developed for the User Interface and Presentation Layer will be deployed, together with their components, using a different IP Group compared to the remaining parts of the solution. Mobile, Liberty, and IHS parts are therefore associated to the *VLAN840 Pure Enablement public* IP Group. This association is done because they need to be exposed to the outside. All the other components use the *VLAN840 Golden System private* IP Group.

3.3.2 Pattern types

After the patterns used in the solution are defined, you are going to build the infrastructure that is required for the solution.

Assumption: This section builds the whole solution with the artifacts and scripts defined and used in *Creating integrated WebSphere solutions using Application Lifecycle Management*, SG24-8243, to create the Claim solution version 1.0.0.0. Any future changes (upgrades) to the solution that reflect a new version will be done with UrbanCode.

The first step is to get the system ready to deploy the artifacts. The artifacts are included in the script packages on the Virtual Systems or directly in the parameter window of the Virtual Applications. For any new requirements for the application, use the UrbanCode agent. Each workload pattern has at least one UrbanCode Agent that is installed with a script package at the creation of the virtual machine.

There are five groups of patterns that are used for this chapter's solution. Each group needs several virtual machines that are created at deployment time and represent an instance of the virtual machine group managed by the instance. Therefore, for the system environment 12 virtual machines are needed. See "Patterns that are used for the solution" on page 6 for an overview of the solution.

Tip: The UrbanCode agent is not necessary. However, if you do not use UrbanCode, you must apply any application upgrades or fixes manually or use the support available from the virtual application pattern. For more information, see Chapter 6, "Maintaining the solution" on page 93.

3.3.3 Pattern groups

To build the middleware for the solution, each application environment of IBM IIB, IBM ODM, IBM BPM, web application, and mobile is built separately. Generally, do not build one single pattern for the entire middleware platform because some patterns are virtual systems and

some are virtual applications. Because of this variance, building a single pattern is not possible in version 1 of the PureApplication System.

Note: The pattern described applies to version 1 of PureApplication System. The new virtual system patterns, called `vSystem.next()` pattern, are introduced in version 2 of PureApplication System. For more information about the new pattern engine, see Appendix B., “IBM PureApplication System V2.0” on page 121.

3.3.4 Virtual application patterns

BPM, ODM, and Mobile are on their latest version build with the VAP types available. ODM and IBM BPM were previously offered as Hypervisor Editions in a virtual image.

IBM Business Process Manager

To build the BPM platform, begin by creating a virtual application pattern. Use the Virtual Application Builder and select the assets from the BPM Components on left pane of the Virtual Application Builder to set up the BPM pattern, shown in Figure 3-13.

To name the pattern, use the right pane on virtual application builder. Use the default values for other settings. The names used here are for example purposes (Figure 3-13).

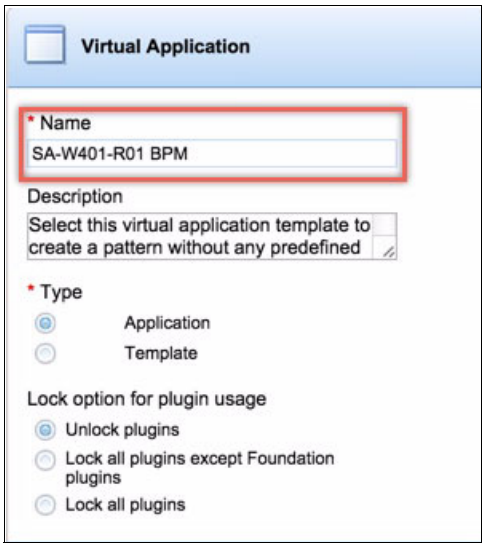


Figure 3-13 Properties of the virtual application pattern

When using the IBM BPM Pattern Type 8.5.0.0, the following components are used:

- Process Application: Process applications running on Process Server
- IBM BPM Database: This component provisions a new database for IBM BPM

To start selecting assets, expand Process Components and follow these steps:

1. Select the assets from the BPM Components to use in the patterns, as shown in Figure 3-14.

Figure 3-14 BPM virtual application components

2. Also shown in Figure 3-14 is how to link your components. Drag the **Process Application component** and **IBM BPM Database component** to the canvas (components are on the left, and you drag them to the canvas located to the right). To link them together, selectthe **Process Application component** and use the blue dot that appears on the right side to link the components. Click and drag the **blue dot** to the **IBM BPM Database** to link the components.
3. Open the properties of the **Process Application** by clicking the component. In the right pane of the Virtual Application Builder, you see the correspondent properties (Figure 3-15 on page 35). Review the following list to assign properties and fields needed:
   - Select **Production** for the Environment Type.
   - Enter a name and password for the administrative user.
   - Change the initial Custom Node Count to 1.
   - Upload the snapshot from the solution's BPM Process Center on the Process Application Package field. Click **Edit** to browse the file.

     Note: The Process Application Package and its creation are described later in this chapter.

   - Decrease the **Number of CPU** to **2.**

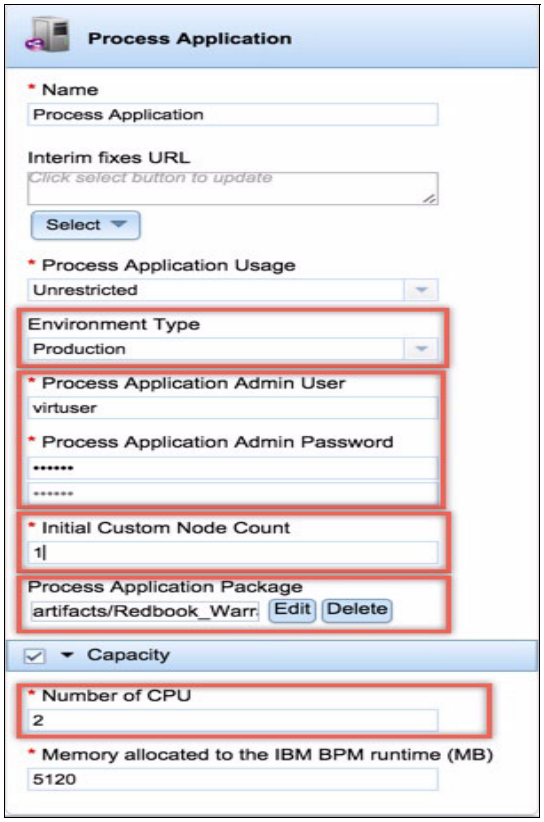


Figure 3-15 Properties of a process application

4. Now focus on the database component in the Virtual Application Builder. Select the **database component** and set **Number of CPU** to **2** in the right pane (only the right pane is shown in Figure 3-16).

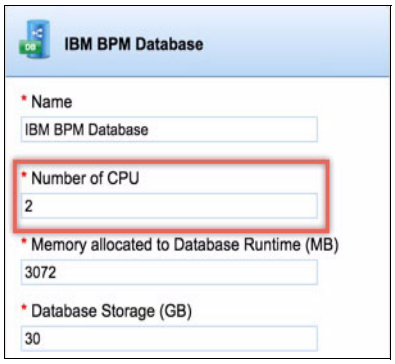


Figure 3-16 Properties of the database component

5. Finish the BPM pattern by saving the Virtual Application pattern.

Installing snapshots on offline process servers

Because the Business Process Server on PureApplication is not connected to the Process Center server as described in *Creating integrated WebSphere solutions using Application Lifecycle Management*, SG24-8243, import the application with a snapshot. The snapshot is available from the Process Center server.

To acquire a snapshot, use the following steps:

1. Log in to the BPM Process Center of the solution, as shown in Figure 3-17 on page 36.

Figure 3-17 Process center login

2. Select your **application** (for example, this chapter uses the **Redbook Warranty Process (RWP)**). You can create a snapshot or select an existing snapshot and export it to an archive file, as shown in Figure 3-18.

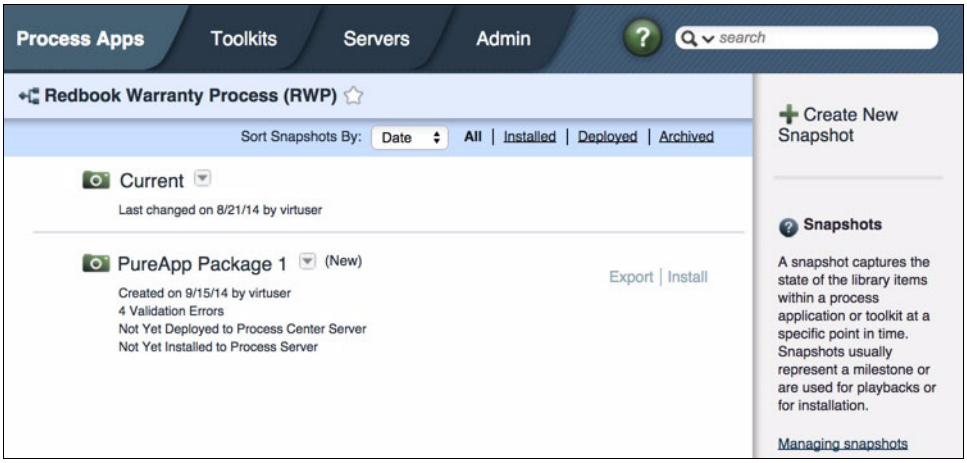


Figure 3-18 Creating a snapshot from the process center

Note: For guidance how to create and extract an installation package to a file, see *Creating an installation package* at the following website:

`http://www-01.ibm.com/support/knowledgecenter/SSFTN5_8.5.5/com.ibm.wbpm.admin.doc/topics/releasing_installing_procs_d1.html?cp=SSFTN5_8.5.5&lang=en`

IBM Operation Decision Manager

In this solution, to add the decisions used for the Claim solution, virtual application patterns were used for ODM. The pattern used is IBM ODM Application Pattern Type 8.5.1.0, which is composed of the following components:

- Decision Center: ODM Decision Center component
- IBM ODM Database: This component provisions a new database for IBM ODM

Expand the IBM ODM Components to select the items for use by following these steps:

1. Use the Virtual Application Builder and select the assets from left pane to set up the IBM ODM Components, as shown in Figure 3-19.

Figure 3-19 ODM virtual application components

2. Drag the **Decision Center component** and the **IBM ODM Database component** to the canvas (on the right). Link the components together by clicking the **Decision Center component**.Use the blue dot that appears on the right side of the selected component to link the components by dragging it to the other component (Figure 3-19).
3. Open the parameter of the Decision Center by clicking the component. The parameters are shown on the right pane as shown in Figure 3-20.
4. Complete the fields as shown in Figure 3-20:
   - Enter a name and password for the administrative use
   - Import the **Extension Model**
   - Import the **Extension Data**

Figure 3-20 Properties of a decision center

5. Within the Virtual Application Builder, select the **IBM ODB Database component** and when the corresponding properties open, set the **Virtual CPU** to **2** in the right pane. The right pane is shown in Figure 3-21.

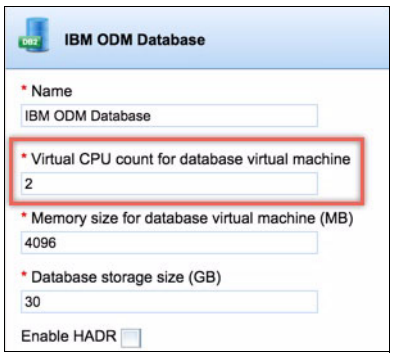


Figure 3-21 Properties of an ODM database

Note: For guidance on how to export a RuleApp archive from the RuleApp project, see *Exporting RuleApp archives* at the following website:

`http://www-01.ibm.com/support/knowledgecenter/SSQP76_8.6.0/com.ibm.odm.dserver.rules.deploying/topics/tpc_res_deploy_rlapp_export_arch.html?lang=en`

Mobile

For the mobile front end, this example uses the IBM Worklight Platform that comes with the virtual application patterns. Worklight is provided in the Virtual Application Patterns of the IBM Mobile Application Platform Pattern Type 6.2.

You also need the correct plug-ins. The following plug-ins are used from the IBM Mobile Application Platform Pattern Type 6.2:

- Worklight Server
- Worklight Adapter
- Worklight Database

Use the following steps in the Virtual Application Builder to create the mobile pattern:

1. Start with the **Application Components** by adding an **Enterprise Application** to the canvas.
2. Select the new component so that the corresponding properties appear on the right pane.
3. Name the Enterprise Application *`WorklightServer`* (or the name of your choice).

4. Include the artifact `ServiceDeskProject.ear` from the solution. To upload the artifact, click **Edit** where you can import the EAR file, as shown in Figure 3-22. Name the Enterprise Application: *`WorklightServer`* (or a name of your choice).

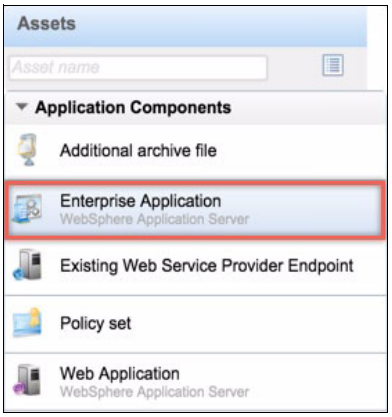


Figure 3-22 Enterprise application component

5. Include the artifact `ServiceDeskProject.ear` from the solution. To upload the artifact, click **Edit** where you can import the EAR file, as shown in Figure 3-23.

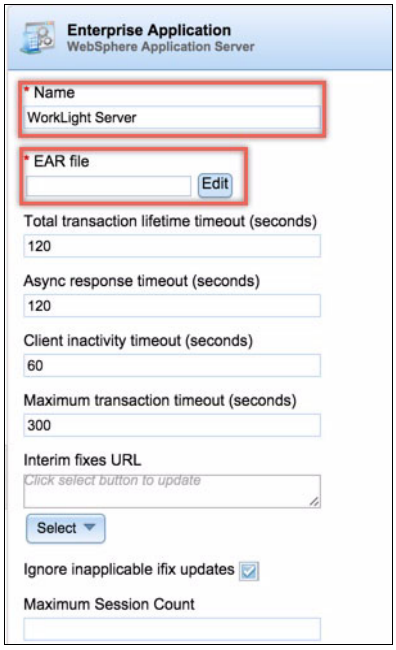


Figure 3-23 Including the EAR file

6. Use the dialog box to import the EAR file, as shown in Figure 3-24.

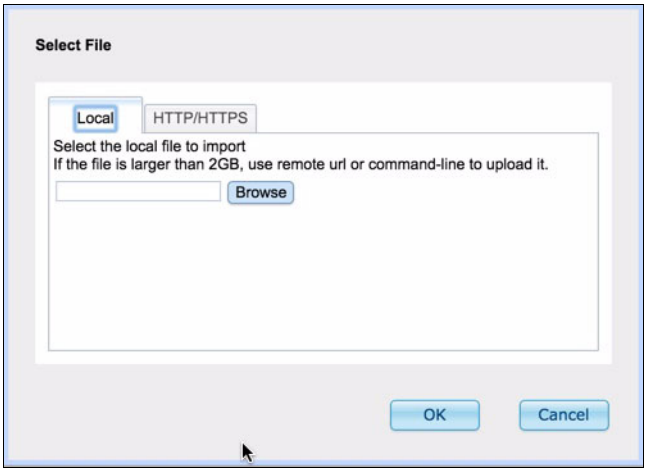


Figure 3-24 Dialog box for importing the EAR file

Note: After the EAR file is imported, the system scans the deployment descriptor and shows a warning that some links are not satisfied.

7. Move your cursor over the alert symbol and hover over messages of unsatisfied links as shown in Figure 3-25.

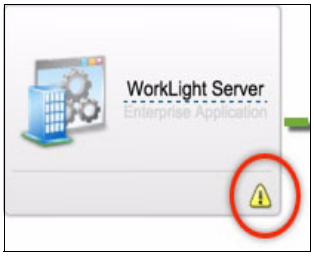


Figure 3-25 Warning messages from a component

8. Next, a database is needed for Worklight where you can define the JDBC connections requested from the EAR file. From the Database Components window, select a **DB2 database** as shown in Figure 3-26.

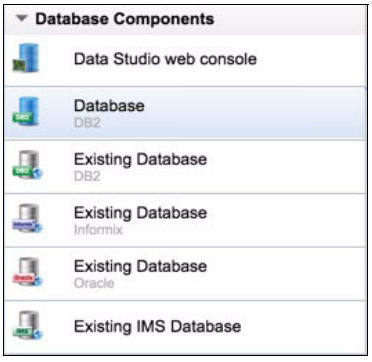


Figure 3-26 Select DB2 component

9. Drag the **database component** from the left pane to the canvas located to the right.

10. Select the **Worklight Server component** and link it to the **database component** by dragging the blue dot to the second component (an arrow is added between those two components).
11. Repeat step 10 for two more times to create a total of three links. Doing so creates links to the database for each JDBC connection when they are later defined. Figure 3-27 shows the links needed for each JDBC connection from the Worklight Server to the database.

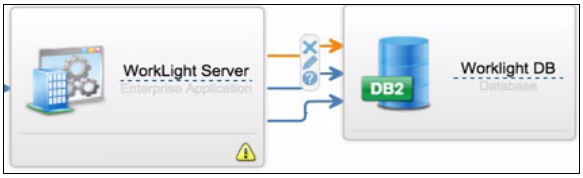


Figure 3-27 Links between components

12. Click one of the JDBC connections to define the link to the database, as shown in Figure 3-28. To edit the JDBC link, click the **drop-down arrow**, and select the JDBC connection in the drop-down list that is defined in the EAR file. The selected link turns orange when it receives the focus (is selected).

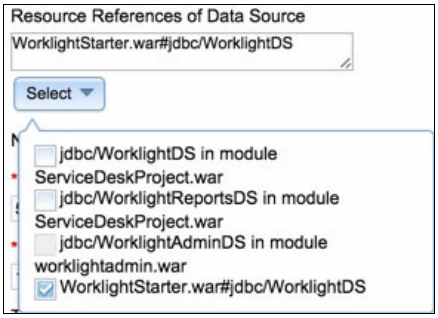


Figure 3-28 Properties of the resource reference

13. Next, add the necessary input into the fields for the DB2 database, as shown in Figure 3-29, and noted in the following list:
    - For the name field, use *Worklight DB* (or a name of your choice)
    - For the **Database Name**, use WRKLGHT
    - For the **Purpose**, select **Production** from the drop-down menu

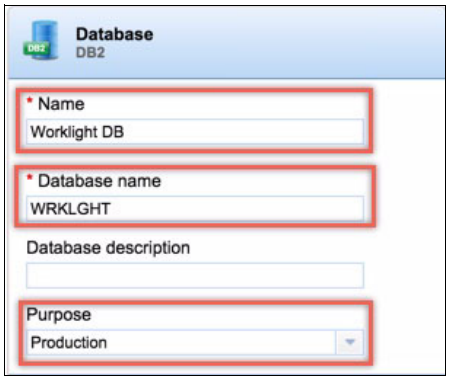


Figure 3-29 Properties of the DB2 component

14. The last component to add is the Worklight Adapter, as shown in Figure 3-30.
    - Open the components from Worklight in the left pane and drag the **Worklight Adapter** to the canvas located to the right (canvas not shown in Figure 3-30).
    - Select the **Worklight Server component** and link it to the **Worklight Adapter component**.
    - Where you can import the adapter file, select the **Worklight Adapter**
    - Import the adapter file: `Backend.adapter`

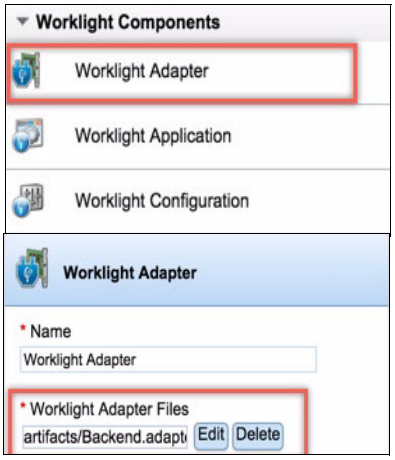


Figure 3-30 Select the worklight adapter

After following these steps, the Worklight virtual application pattern should look like Figure 3-31.

Figure 3-31 Components for mobile

Note: You can also deploy the Worklight application to the cloud directly from Eclipse.

Using Eclipse for Mobile

You can use the Worklight Studio installation to create the mobile application, to directly create a mobile virtual application pattern on PureApplication System, and deploy it directly to the cloud. This process simplifies the application installation because no further configuration is needed in the Worklight server.

To create the application pattern, first check the Worklight deployment settings for PureApplication Systems in Worklight Studio under **Windows** → **Preferences** → **Worklight** → **IBM Mobile Application Platform Pattern**.

A window similar to Figure 3-32 is displayed that shows the settings of the target system where the patterns will be created. In this case, it is a PureApplication System. You can also use other IBM PaaS services, such as IBM Workload Deployer or SoftLayer.

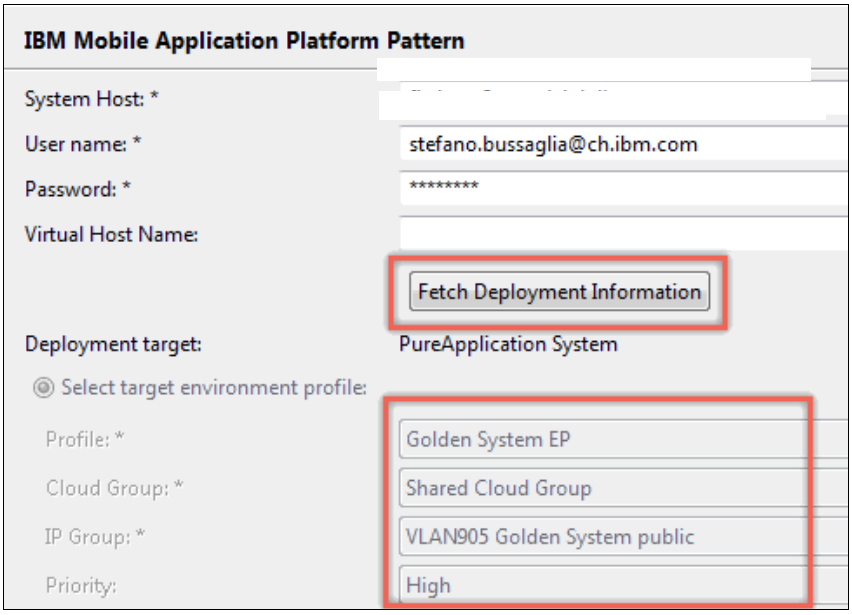


Figure 3-32 Panel for the cloud resource

After defining the target system, click **Fetch Deployment Information** to retrieve the cloud resources from the environment profile of the user provided on the target system (also shown in Figure 3-32). The **Deployment target** shows where the mobile application will be deployed.

Note: You can select the Deployment target from a drop-down list, if the user has more than one environment profiles assigned.

Finally, right-click the Worklight project and select **Run As** → **Deploy project as IBM Mobile Application Platform Pattern**, as shown in Figure 3-33. This process creates the application pattern.

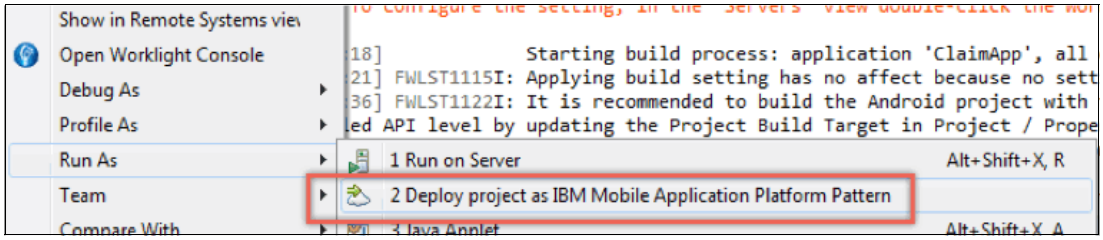


Figure 3-33 Creating a pattern and deploying from eclipse

For more details about Virtual Applications, see Chapter 6 in *Adopting IBM PureApplication System V1.0*, SG24-8113:

http://www.redbooks.ibm.com/abstracts/sg248113.html

3.3.5 Virtual System Patterns

The next middleware components that are used are built within Virtual System Patterns. For example purposes, this book shows both pattern types (VAP and VSP) to demonstrate each option. However, the IBM Integration Bus (IIB) comes only within Virtual System Patterns.

This section shows you how to build the Virtual System Patterns for IIB and the web application for the Claims solution used in this book.

IBM Integration Bus

The IBM Integration Bus patterns come with virtual system patterns. To start system integration, create a virtual system pattern in the pattern editor (**Patterns** → **Virtual systems** page, click the **Green Plus Sign** to create a pattern) and drag the IBM Integration Bus part onto the blank canvas. For specifics about how to create a virtual system pattern, see 2.1.3, “Virtual system patterns” on page 7.

Using the Claim solution (from *Creating integrated WebSphere Solutions using Application Lifecycle Management*, SG248243), you need an application database that stores the claim data. To facilitate the example in this book, a DB2 database is added to the Integration Bus pattern.

The following list notes the required components for creating your virtual system pattern, also shown in Figure 3-34:

- IBM Integration Bus 9.0.0. Hypervisor edition
- IBM Integration Bus Basic
- IBM DB2 Enterprise 10.1.0.2

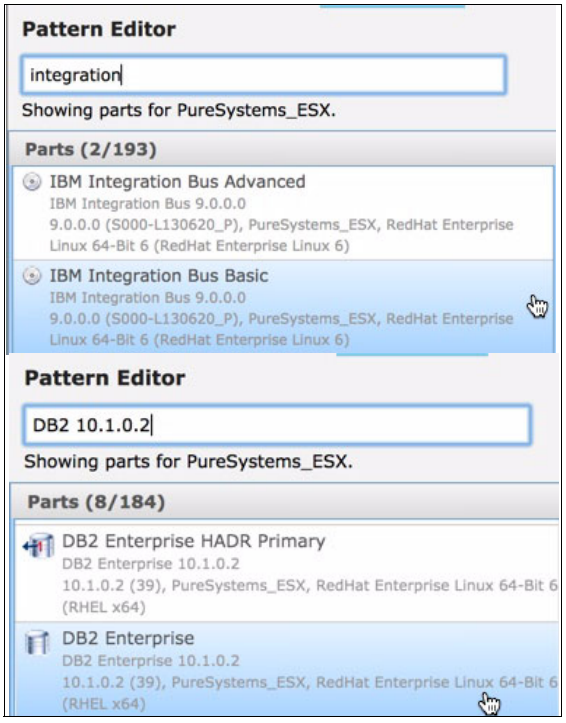


Figure 3-34 Select parts for integration bus

To add the script packages and order the installation sequence, use the following steps:

1. From the pattern editor, after you drag the components shown in Figure 3-34 on page 44 onto the canvas, you see an IBM Integration Bus component. At this point, you can include the script packages, as shown in Figure 3-35. For more details about the script packages, see 3.3.7, “Script packages” on page 50.

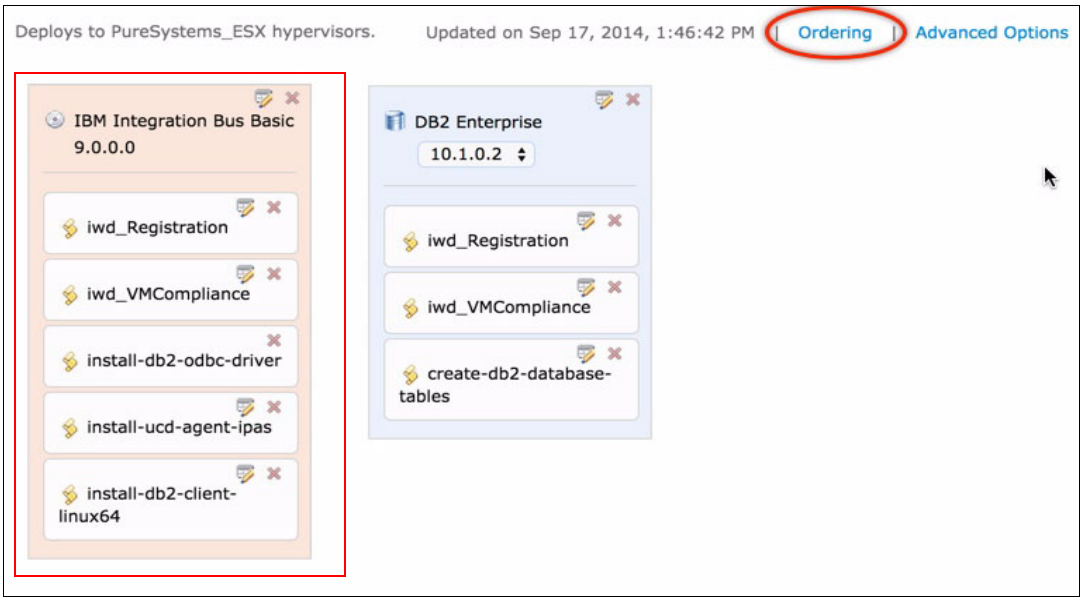


Figure 3-35 Script packages and option to define ordering sequence

2. To control the installation sequence of the components, you can view or change the ordering by clicking the **Ordering** link, then dragging the components to create the installation sequence required, shown in Figure 3-35.

Figure 3-36 shows the installation sequence of IIB and DB2. The sequence starts with the binary files and then the script packages.

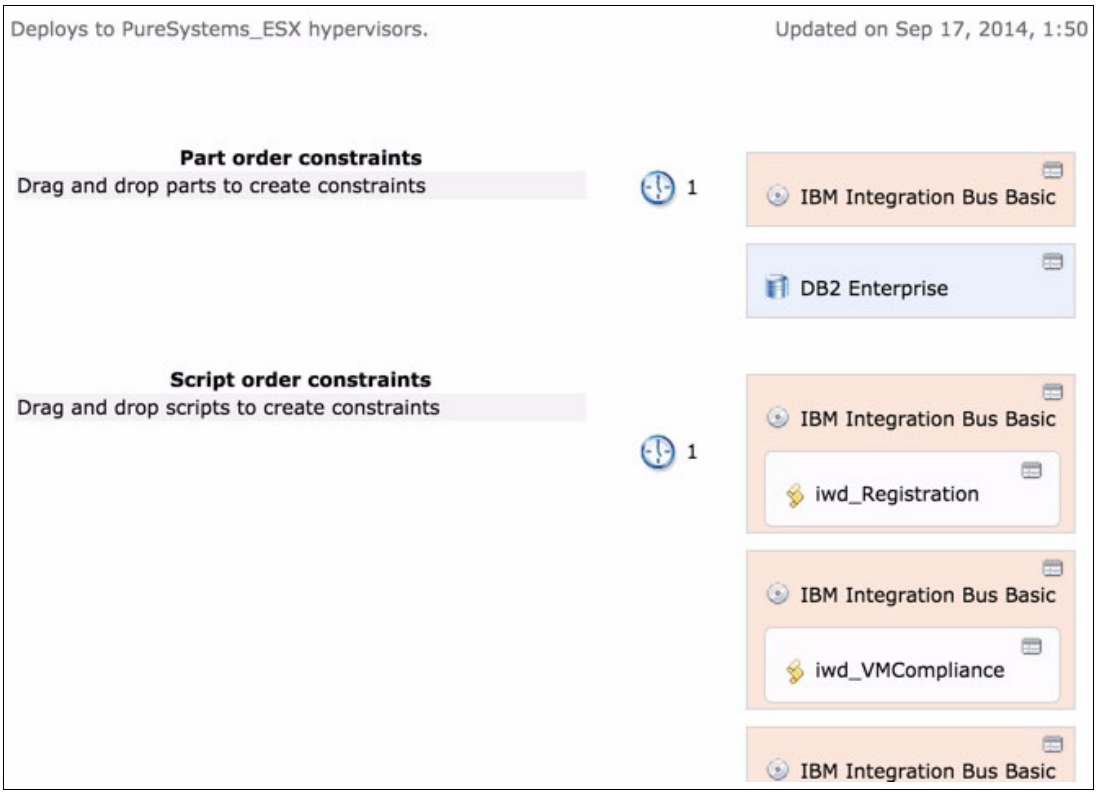


Figure 3-36 Ordering capability of installation sequence

3. To switch to previous pane click the **Topology** link at the upper right of the canvas.

Any additional application installations that are related to the solution are wrapped into script packages that will be started after the installation of the Integration Bus.

Script Packages

You need script packages to do specific tasks when working with the integration bus installation. For all script packages needed for the integration bus and their purposes with the Claim solution, see Table 3-2.

Table 3-2 Script packages for integration bus

Name	Artifact	Description
install-ucd-agent		Script for installing UrbanCode Agent
install-db2-client.linux64		Script for installing the ODBC client to connect the solution database
configure-IIB	ServiceDeskESB.bar	Script for creating Queues and Channels, and importing the bar file to the Integration Bus

Name	Artifact	Description
install-db2-odbc	odbc.ini	Script that configures the ODBC connection
create-db2-database-tables	Tables.ddl	Script for installing the Database and Schema needed for the solution

After including all your script packages, you can close the pattern editor by clicking **Done**. Script packages are explained in more detail in 3.3.7, “Script packages” on page 50.

Web application

The web application includes an HTML generated claim form for the user and controller logic for sending a message that contains the user input to the Integration Bus.

The web application runs on the WebSphere Application Server Liberty Profile. From the pattern editor, to create a virtual system pattern for the web applicationusing the WebSphere Application Server Hypervisor Image, only the Liberty Profile Server and the IBM HTTP Server components are selected, as shown in Figure 3-37.

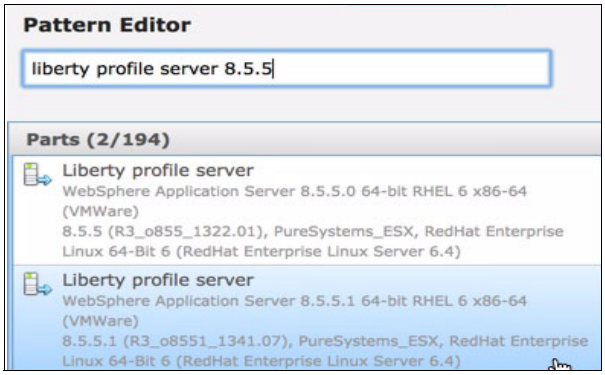


Figure 3-37 Selecting the Liberty profile server

The following parts are used in this solution image (WebSphere Application Server 8.5.5.1 64-bit):

- Liberty profile server
- IBM HTTP servers

After all the parts are selected, your canvas should look like Figure 3-38.

Figure 3-38 Parts for web application

You are now ready to add the script packages to the parts.

Script Packages

Script packages provide the customized parts for the solution. While using virtual system patterns, you also must include several script packages for the Claim solution in this book. Table 3-3 lists the script packages that you need.

Table 3-3 Script packages for web application for Claim solution

Name	Artifact	Description
install-ucd-agent		Script for installing UrbanCode Agent
InstallApp	ServiceDeskWebApp.war	Deploys the war file and server.xml

After you have included the script packages, close the pattern editor by clicking **Done**.

3.3.6 Adding scaling policies

Virtual Application System gives you the opportunity to scale out the application on heavy workloads by adding a copy of the virtual machine or by increasing the amount of CPU in a virtual machine.

In Virtual Application Patterns, you can define more scaling policies. A scaling policy can be defined for the whole pattern or individually for some components.

Note: All the patterns used to deploy the Claim solution are without scaling policies.

Policies for the Virtual Application Pattern

Not all pattern components have scaling policies available. In the Virtual Application Pattern Builder, to obtain the policies of a component, click the **Blue Plus Sign** at the upper right of the component. A menu appears with the inherited policies, as shown in Figure 3-39.

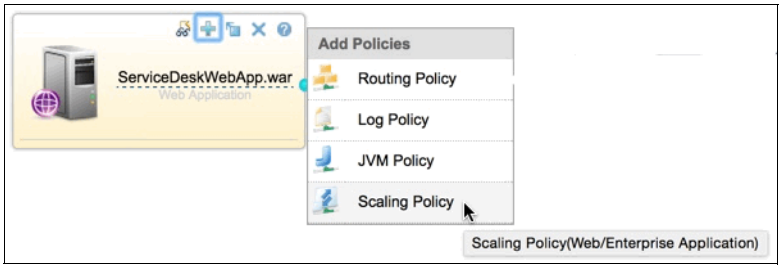


Figure 3-39 Component policies

To use a policy for the entire pattern, use the option shown at the upper left of the canvas of the Virtual Application Pattern Builder, as shown in Figure 3-40.

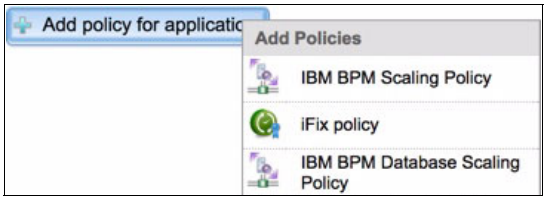


Figure 3-40 Defining the policy of the pattern

Scaling policy for BPM

There are many options to scale the BPM application in the Virtual Application Pattern. This section explains a sample that can be added for the Claim solution. The properties of the scaling policy are shown in the right pane of the Virtual Application Pattern Builder. The scaling policy is related to the BPM Process Application component. The following list notes several options that can be used to scale applications on heavy workloads, as shown in Figure 3-41:

- CPU count for each Custom Node
- Maximum Custom Node count: (this means that the system will clone and add virtual machines)
- The percent threshold the CPU must exceed for scaling to engage
- Wait time before scaling (in seconds)
- Options for how the system scales

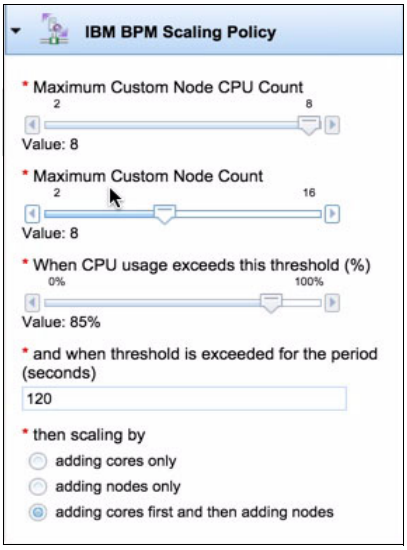


Figure 3-41 Properties of the scaling policy for BPM

For more information about scaling policies, see the following website:

http://www-01.ibm.com/support/knowledgecenter/SSNLXH_1.0.0/com.ibm.ipas.doc/iwd/apt_policyov.html

3.3.7 Script packages

Script packages provide a way for you to customize parts in virtual system patterns, beyond the customizations that are already provided by IBM PureApplication Systems. These parts usually need further customization to meet the needs of a specific application. For example, creating and deploying a virtual system pattern using the Liberty profile part automatically installs the Liberty profile during the deployment. However, you also need to install the UrbanCode agent. Extra tasks, such as installing the UrbanCode agent, are done through script packages. As Figure 3-42 illustrates, when editing a virtual system pattern in the Pattern Editor, you can attach a script package to a part by dragging it directly onto that part.

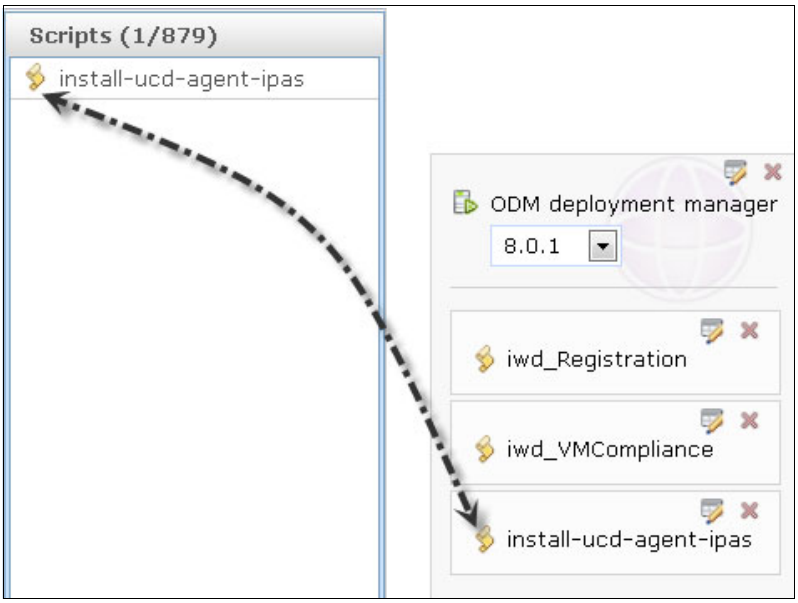


Figure 3-42 Dragging the script package

A part can contain multiple script packages and you can order which parts and script packages run first throughout the pattern by using the Ordering view. This view lets you drag parts and scripts to establish their execution order.

Several customization techniques can be done using a script package approach:

- Define your own scripts to configure the deployed environment
- Configure any middleware software, such as WebSphere Application Server, Liberty, even a DB2, and so on.
- Customize the preinstalled virtual images and patterns

A script package is an archive (.zip) file that contains artifacts that you want to be run, and artifacts that you want to be run upon. Script packages can be as simple as a single script file or as complex as an entirely new product. The content of a script package is not defined or restricted by the PureApplication System. The scripts included in a script package define the requirements for the script package. As shown Figure 3-43 on page 51, the sample script package is formed as a file called `ibm-ucd-agent.zip` and is composed of three types of files:

- UCD-Agent_Installation.zip file, which represents the UrbanCode Agent to be installed
- Shell script file, which contains the logic to run the UrbanCode Agent installation.
- JSON file, which represents the configuration of the script package, such as name, place where the script package will be run, location of the logs, which shell script will be run, and many other parameters.

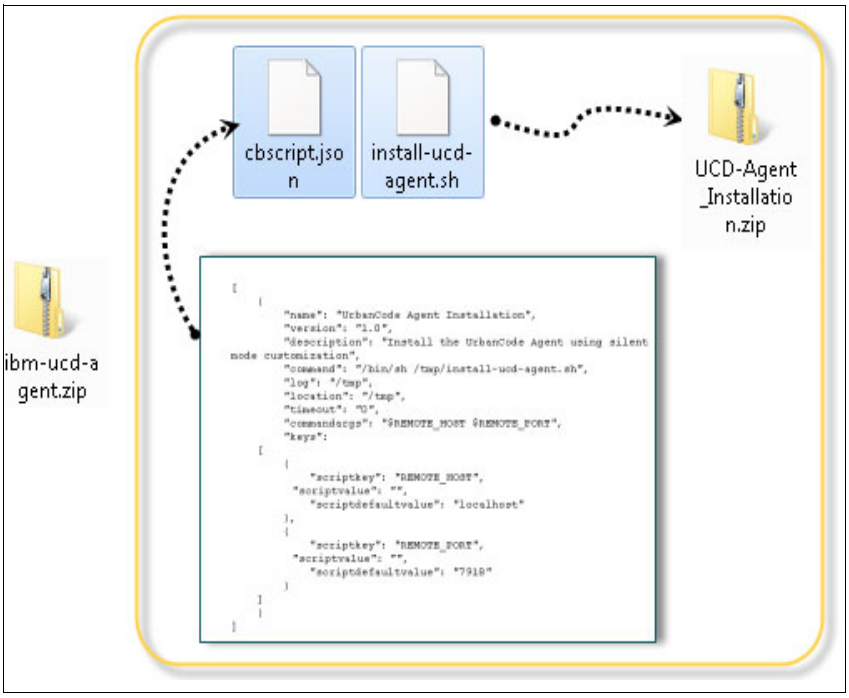


Figure 3-43 Script package structure files

Patterns are made up of virtual image parts, and each virtual image part gets its own virtual machine to run in when it is deployed. A script package is associated with a virtual image part. Upon deployment, the virtual image part becomes a virtual machine, and the script package that is associated with the virtual image part gets run on that virtual machine. Associating a script package with a virtual image part is as easy as dragging the script package onto the virtual image part.

During the deployment process, the PureApplication System first creates and configures a virtual machine for each virtual image part. After the topology is configured and running, PureApplication System will Secure Shell (SSH) over as root context to the virtual machine for which the script package is targeted. The PureApplication System then unpacks the archive and runs the scripts defined within.

Appendix A, “Script packages” on page 113 contains examples on how to create the script package for the solution. This chapter now describes the steps that are involved in creating and deploying a script package into the cloud.

To create a script package from the PureApplication System Workload console as shown in Figure 3-44 on page 52, complete these steps:

1. Select **Catalog** → **Script Packages**.
2. From Script packages window, click the **Green Plus Sign** to create a script.
3. Name your script, `install-ucd-agent-ipas`, and click **OK**.

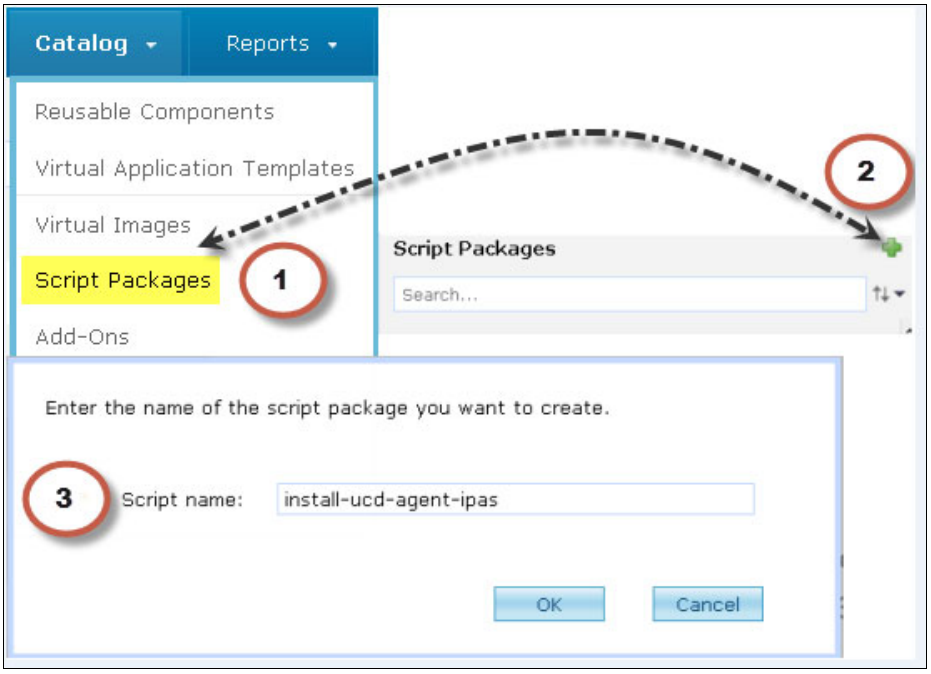


Figure 3-44 Steps for creating a script package from the GUI

4. Click **Browse** and select your script package, then click **Upload**. The amount of time this operation takes depends on the size of your archive and the speed of your network connection. When this step completes, you receive a successful confirmation message below the **Upload** button. See Figure 3-45.

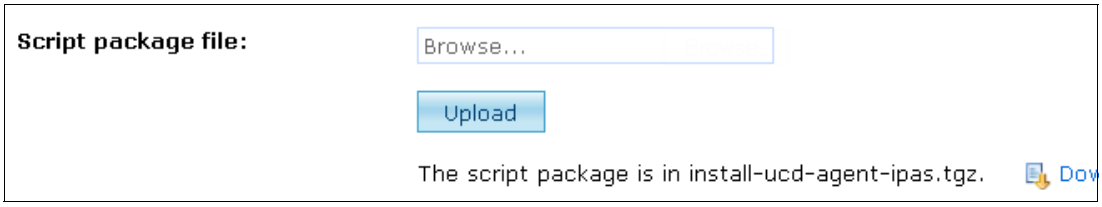


Figure 3-45 Uploading the script package

5. To create the environment variable, click **Add**. See Figure 3-46.

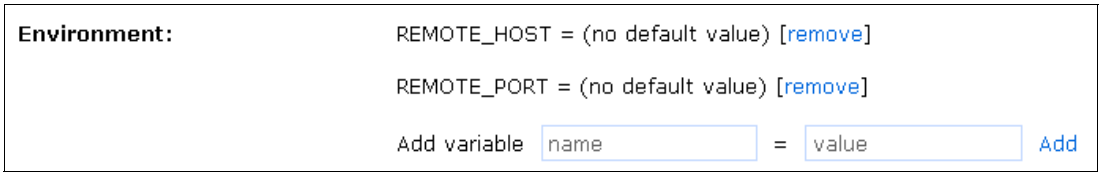


Figure 3-46 Adding variables

6. Normally, when using JSON, all fields are populated. However, the PureApplication System allows you to apply updates instead of redeploying the script package again with the new `cbscript.json`. See Figure 3-47.

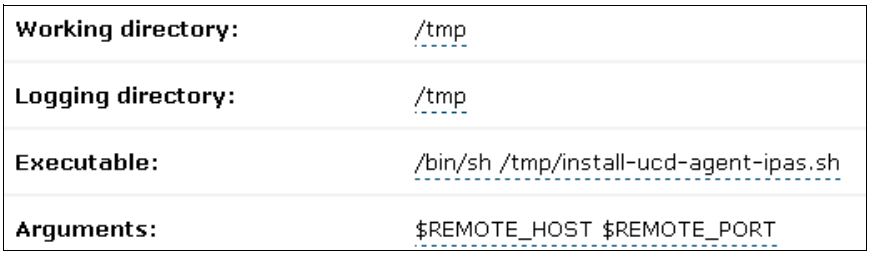


Figure 3-47 Additional information

The **Logging directory** field, shown previously in Figure 3-47, tells PureApplication System the directory that is used to store logs that are produced by the script package. This is not input to the executable file to tell it where to place logs. By defining the log location in the **Logging directory** field, you are able to use PureApplication System to view these logs.

Note: Because the content of the script package is running on Linux, if you create the `.json` and script files in Windows, make sure that you use an editor that supports UNIX style text files. You must save these files with an editor that can write a single Line Feed (LF) character to signify a new line (UNIX mode) instead of a typical Windows-style carriage return-line feed (CRLF). Otherwise, your scripts will not run in Linux. There are many editors available at no extra cost that support saving text files in the Linux format.

Parameters

Script packages also let you specify parameters that get passed as environment variables to the part so that you can make the scripting configurable and reusable. As Figure 3-49 on page 54 illustrates, this enables a single script to be used for multiple parts, requiring only that the person deploying the pattern change specific values. This also makes the parts more reusable. Similarly, you can save a set of connected parts and their scripts as a single reusable virtual system pattern.

Package the script files

After creation, the scripts and the JSON files can be compressed into an archive file such as `.zip` or `.tgz` files. You can include many other files that could be part of the script package component and add all of them in the root directory.

The following command is an example how to create a `.tgz` file:

```
tar -cvzf deploy-redbook-app.tgz deploy-redbook-app.sh
deploy-redbook-app.json ServiceDeskWebApp.war server.xml
```

Discover the properties of each part

There are some situations in which you must work with a variable that is not known until deployment time. An example is the host name of the part. Some of the part's variables, such as host name, are hidden. When you open the part properties and view the field labels, the name of the label is not always the same as the co-related properties. This section tells you what information you need and how to use that information.

To discover the properties of the part, you need to have the following information and tool:

- The part name
- The Virtual System Pattern where the part is located
- The command line tool

To find the name of the part, use the following steps:

1. Click **Menu** → **Patterns** → **Virtual System Patterns**.
2. Select the target **Virtual System Pattern**.
3. Select the part and click the **Properties** icon, as shown in Figure 3-48.

In this example, the name of the part is `LibertyPart`.

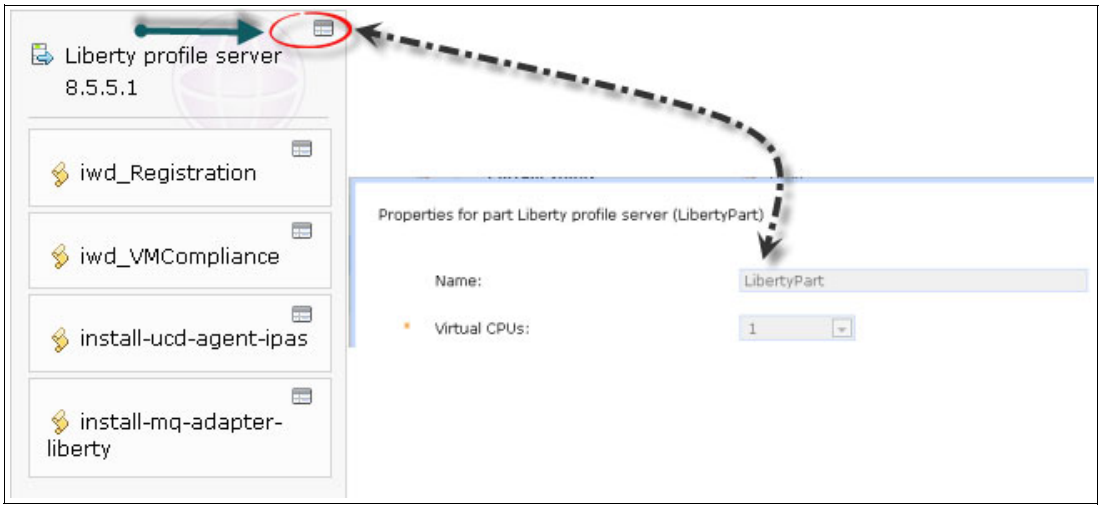


Figure 3-48 Discovering a part name

Next, you need the command line tool, use the following steps to download the tool:

1. Go to the IBM PureApplication System welcome window.
2. Click **Download Tooling**.
3. Click the item, then click **Download command line tool**, as shown in Figure 3-49.
4. Unpack it into any directory.

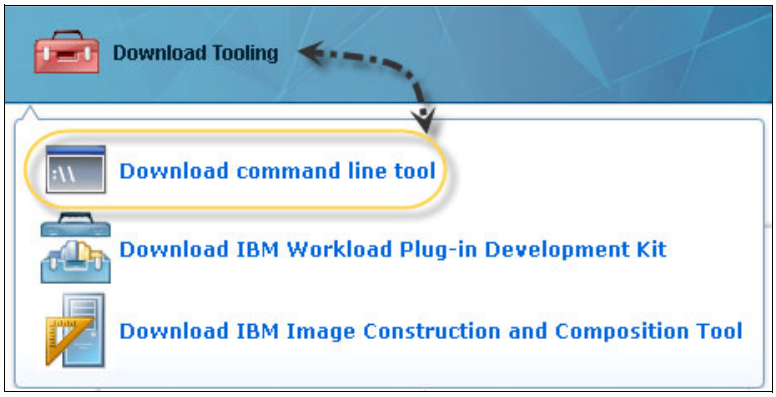


Figure 3-49 Downloading the command line tool

5. Change to the directory where you downloaded the CLI and issue the following command:

   ```
   ./deployer -h <host> -u <userID> -p <password>, -f
   ../samples/listPatternProperties.py
   ```

   This command prints a list of Virtual System Patterns. At the end, it asks you to type the index number of the Virtual Systems Pattern. You want to enter the index for the VSP that contains the part you are seeking the properties for, as shown in Example 3-1.

Example 3-1 Pattern properties

```
select a pattern: 1098
part properties/script parameters for SA-W401-Liberty-Hypervisor
part-0.HWAttributes.numvcpus (default=1)
part-0.HWAttributes.memsize (default=2048)
part-0.HWAttributes.physcpures (default=false)
part-0.HWAttributes.physmemres (default=false)
part-0.ConfigWAS.liberty_server_name (default=server)
part-0.ConfigWAS.was_im_repo_location
part-0.ConfigWAS.was_im_repo_user
part-0.ConfigWAS.was_im_repo_pass
part-0.ConfigPWD_ROOT.password (default=saw401r01)
part-0.ConfigPWD_USER.username (default=virtuser)
part-0.ConfigPWD_USER.password (default=saw401r01)
part-0.ConfigVNC.enabled (default=true)
part-0.script-0.CLOUD_HOSTNAME (default=fit-iwd-1.rtp.raleigh.ibm.com)
part-0.script-1.SLEEPING_HOURS (default=12)
part-0.script-2.REMOTE_HOST
part-0.script-2.REMOTE_PORT
```

Note: The properties that are listed in Example 3-1 can be used in your variables as default values.

Now you have the properties list of the part. You can use any properties noted. An example is the property `ConfigWAS.was_im_repo_location`. This property is the path where the Liberty Profile will be installed. You can now create a variable in the script package and add this property as a default value. Note the value `part-0`, must be replaced by the name of the appropriate part.

During the creation of the script package, it is possible to add variables, as shown in step 5 on page 52 (environment variable). The advantage of this is that you are able to complete the default fields of the variable with the properties of the part, as shown in Figure 3-50.

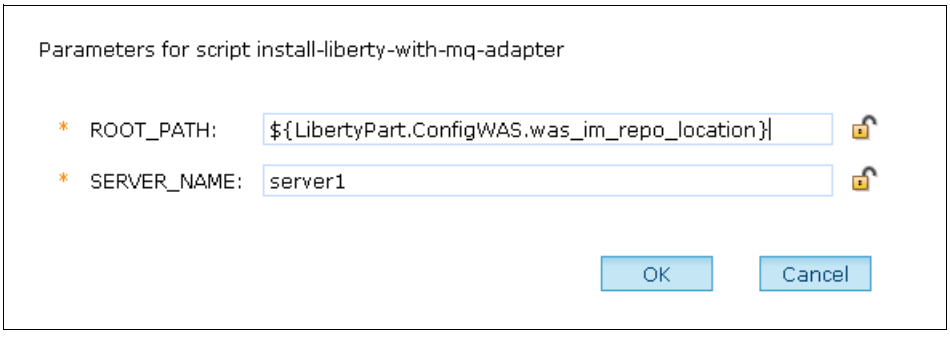


Figure 3-50 Variable using the same value of the part properties

3.4 Shared Services

You can define the Shared Services like a predefined virtual application pattern that is deployed in the cloud and can be shared by multiple applications, including virtual applications, virtual systems, and virtual appliances. Shared services are included with the product and in the IBM Foundation Pattern type. Shared Services are visible only if you own the authorization role either as Workload resources administrator with full permissions or as cloud administrator. In general, the operational users belong to the Workload resources administration category.

Shared services are built on the multitenant architecture model, so a single deployment can provide service to multiple tenants or to multiple client applications.

It is possible to deploy shared services either at the cloud group level or at the environment profile level. However, regardless of which deployment level is used, only a single instance of a type of shared service can be deployed. If the deployment occurs in an environment profile, all the application deployments done in the profile can use that shared service. The same happens in the case of deployment of the shared service in a cloud group.

You can deploy the following Shared Services as needed:

- Caching Shared Service
- Database Performance Monitor Shared Service
- Elastic load balancing proxy Shared Service
- IBM Endpoint Manager Shared Service
- Red Hat OS Update Shared Service
- System Monitoring Shared Service

The example uses the Caching Shared Service. This service is one of the most commonly used.

Note: It is out of the scope of this publication to explain how to develop a Shared Service. For more information about Shared Services, see the specific documentation at the following website:

http://pic.dhe.ibm.com/infocenter/psappsys/v1r1m0/index.jsp?topic=/com.ibm.puresystems.appsys.1500.doc/systemconsole/c_syswelcomeipassc.html

3.4.1 Caching Shared Service

The Caching Shared Service is a shared service for virtual deployments to store, share, and access caching information among components. This shared service also offers auto-scale capabilities so that new component instances can be created or deleted without retransmitting information.

The Caching Shared Service is based on the WebSphere eXtreme Scale technology, which provides highly efficient caching. Such a shared service is self managed and highly available.

Each deployed instance of the Caching Shared Service contains the three virtual machines as noted in the following list:

- Caching-Master: This virtual machine hosts the shared service administrative API and the WebSphere eXtreme Scale catalog and container processes.
- Caching-Catalog: This virtual machine hosts the WebSphere eXtreme Scale catalog (redundancy) and container processes.

- Caching-Container: This virtual machine hosts the WebSphere eXtreme Scale container processes.

When deploying the Caching Shared Service, you can specify both the instance size and the number of instances. By doing so, you indicate the size and initial number of virtual machines in the cloud that the system devotes to the Caching Shared Service.

To manage the Caching Shared Service, use the following steps:

1. Select **Instances** → **Virtual Applications** from the Workload console.
2. Choose a deployed virtual application instance from the list.
3. Click **Manage**.

Deploying the Caching Shared Service

To deploy the Caching Shared Service, complete the following steps:

1. Open the IBM PureApplication System Workload console.
2. Select the Cloud tab.
3. Select the **Shared Services** option to access the Shared Services page.
4. In the Caching Services section, select the Caching Shared Service to deploy.
5. To start the deployment, click the **Deploy option** on the toolbar. The Configure and deploy a shared service window is displayed as shown in Figure 3-51.

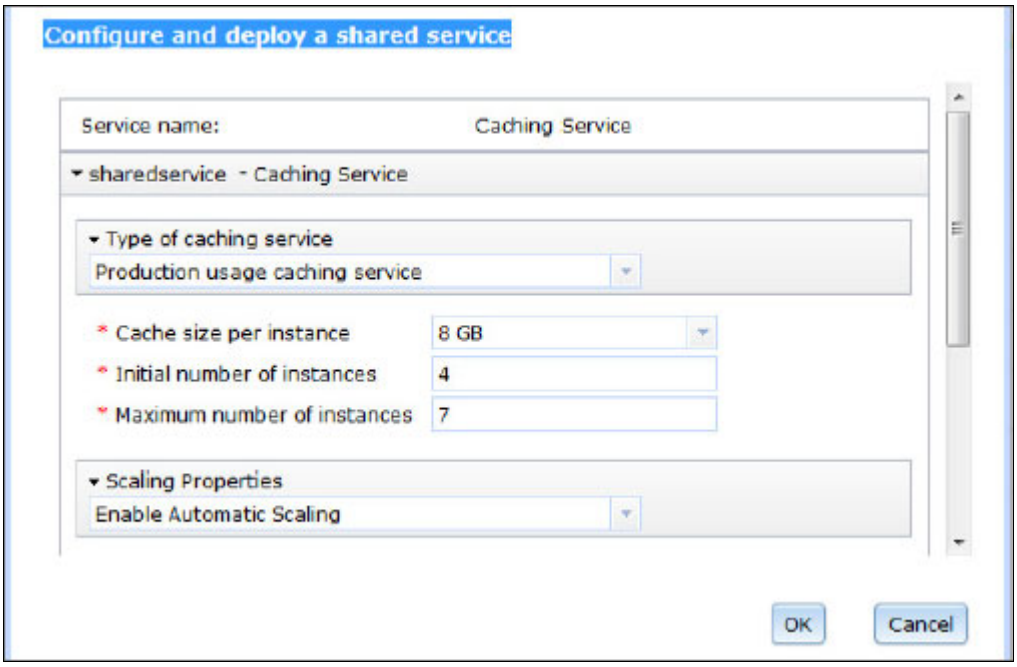


Figure 3-51 Configuring and deploying a shared service

6. Fill or adjust all the parameters according to your needs and click **OK.** A window appears in which extra configuration information is required (Figure 3-52).

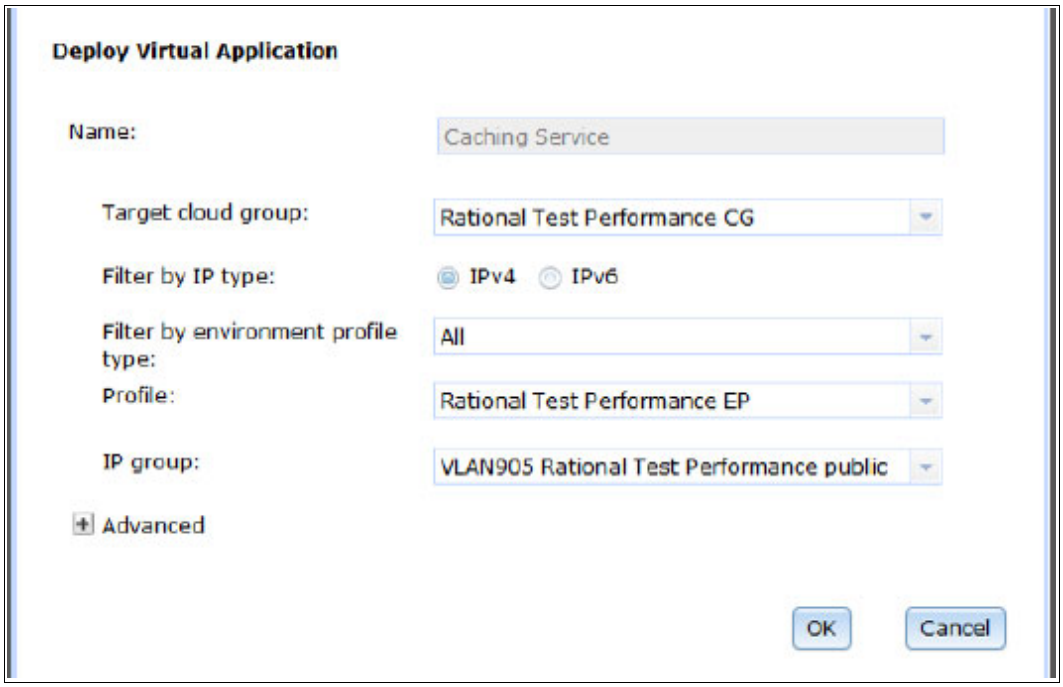


Figure 3-52 Deploying the virtual application

7. Complete the following fields:
   - **Name** field: Enter the name of the application that is being shared as a service. You cannot use more than two consecutive underscore characters in a shared service name.
   - **Target cloud group** field: Select the group that is associated with the shared service.
   - **Filter by IP type** field: Select the IP type (**IPv4** or **IPv6**) that is associated with the shared service.
   - **Filter by environment profile type** field: Select the appropriate category of environment profiles from the drop-down list.
   - **Profile** field: Select the environment profile that contains the IP group dedicated to the caching instances from the drop-down list.
   - **IP group** field: Select the dedicated IP group created specifically for the caching instances from the drop-down list.
8. Click **OK** when done. By clicking **OK**, the system starts the deployment of the shared service, and starts the virtual machines after their creation is completed.
9. To connect to the deployed virtual machines using SSH, click **Advanced**. Specify the SSH public key in the **SSH Key** field. If you do not have an existing SSH key pair, you can generate one for reuse with other deployments by clicking **Generate**. The **SSH Key** field is then populated with the generated public key. Click **Download** to save the private key to your local system.

Note: To stop the Shared Service, click the **Stop** icon in the toolbar of the page for the selected Shared Service.

Viewing Shared Service details

To view the details about the Shared Service, such as the ID, description, creation date, pattern type, and so on, use the following steps:

1. Open the IBM PureApplication System Workload console.
2. Open the **Cloud** tab and choose **Shared Services**.
3. Select the shared service for which you want to review the details

Updating Shared Service instances

To update shared services, from the workload console, perform the following steps:

1. Click **Instances** → **Shared Services**.
2. Select the shared services instance that you want to check for upgrades and click **Check for upgrades**

Important: If the shared service instance is deployed in multiple clouds, remember that each shared service instance in each cloud must be updated individually.

3.5 Rapid application deployment using IBM UrbanCode Deploy

This section describes how to use UrbanCode Deploy to prepare, build, and run your deployment process. If you are not familiar with UrbanCode, see the following website for examples, videos, and documentation on UrbanCode:

https://developer.ibm.com/urbancode

This section covers all steps to deploy your application and configuration in Liberty using UrbanCode.

One of the concepts of the PureApplication System is building your environments the easy way, even when applying for continuous integration when using UrbanCode. IBM UrbanCode™ Deploy relieves the challenge of manual deployment by providing tools that improve the speed of the development process using automation and improved reliability. There are also features for deploying to a specific environment before production. If you do not have UrbanCode installed, see the following website for instructions:

http://pic.dhe.ibm.com/infocenter/ucdeploy/v6r0/index.jsp?topic=%2Fcom.ibm.udeploy.install.doc%2Ftopics%2Finstall_ch.html

Critical information for this section and steps: It is assumed from this point that you have UrbanCode installed and set up. This paper uses the solution that is used in *Creating integrated WebSphere solutions using Application Lifecycle Management*, SG24-8243. The following section continues from the steps in Chapter 6, Item 6.4 Deploy the Application on Liberty.

Also, the following steps assume that you have the Component, Component Process, Application, and Application Process created in UrbanCode. If not, see Chapter 6 of *Creating integrated WebSphere solutions using Application Lifecycle Management*, SG24-8243.

3.5.1 Creating an environment

Using the UrbanCode web page and the following steps, you can create an environment:

1. Click **Applications** → **Liberty Application**.
2. On the Environment tab, click **Create New Environment**, as shown in Figure 3-53.

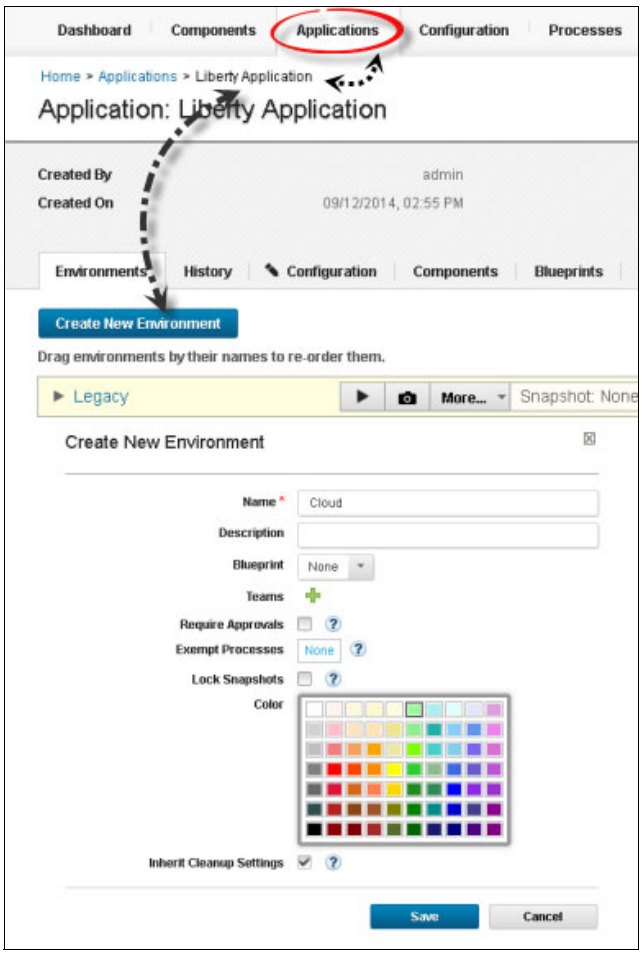


Figure 3-53 Creating an environment

3. In the **Name** field, enter a name for the new environment. This example uses `Cloud`, as shown in Figure 3-53.
4. Click **Save**.

Now, you have a environment created in UrbanCode. However, you need to define which resource the Component Process uses to deploy your war file in Liberty. It is important to next verify whether the resource exists and review its configuration and agent.

Check for the agent of the resource for the new environment

To check the agent assigned to the resource for the created environment, use these steps:

1. Go to the UrbanCode web page.
2. Select the **Resources** tab.

3. Select the **Agents** tab and check whether the agent was installed in your Liberty Server during the deployment of the VSP. For more information about how to install the agent manually, see Chapter 6.2.1 in *Creating integrated WebSphere solutions using Application Lifecycle Management*, SG24-8243. This publication is available at the following website:

   http://www.redbooks.ibm.com/abstracts/sg248243.html

Now that you have an agent for the resource, the resource can be added to the environment inside the UrbanCode Application. The UrbanCode Application deploys the war file to the agent. The next sections cover the following steps within the UrbanCode Application:

- Adding the agent to the resource of the created environment
- Adding the resource to the created environment
- Deploying the war file to the agent

Add an agent to the Liberty-Group resource

To add the agent to your resource, use the following steps, shown in Figure 3-54:

1. Select the **Resource tab**.
2. Select the Resource, **Liberty-Group**.
3. Using the **Actions** menu, select **Add Agent**.
4. After the pop-up window opens, choose the agent where Liberty is located and click **Save**.

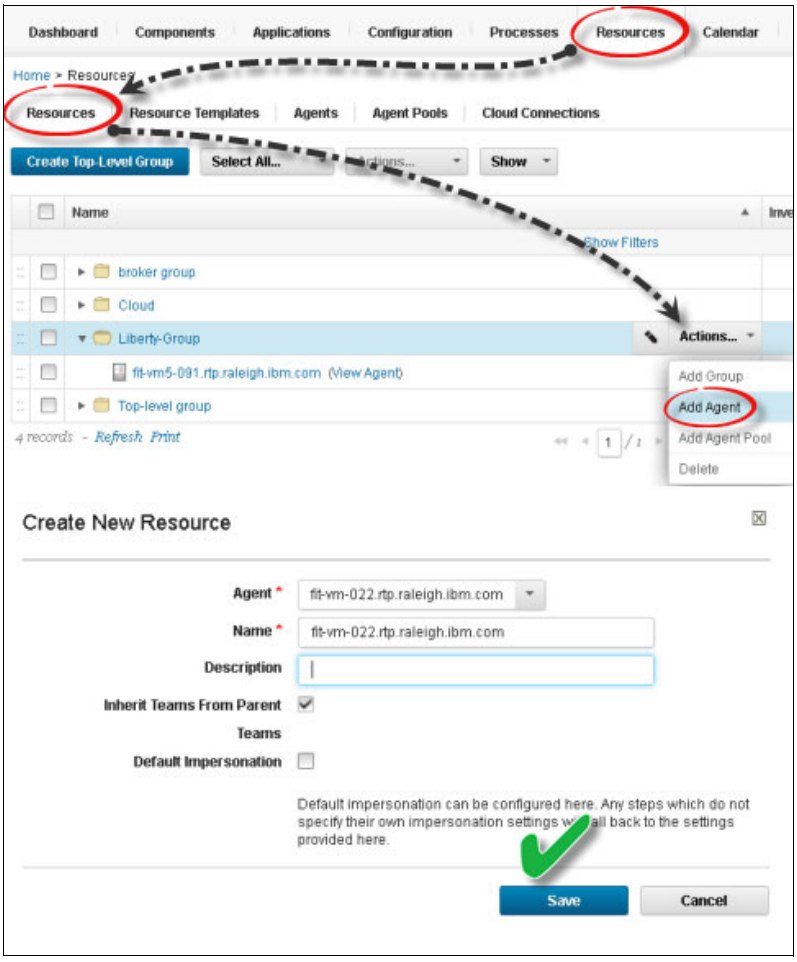


Figure 3-54 Adding an agent

Add the resource to the created environment

You now have an agent for the resource. Before you deploy the war file, you must attach the agent for that resource to the environment. This example uses the Cloud environment (specifically the environment that was created in 3.5.1, “Creating an environment” on page 60).

To add the resource to the Cloud Environment, use the following steps, shown in Figure 3-55:

1. Within the UrbanCode Application, select the **Applications tab** → **Environment** tab.
2. Select the **Cloud environment**, created on “Creating an environment” on page 60.
3. Click **Add Base Resource**.
4. Choose the Agent associated to the Liberty server.

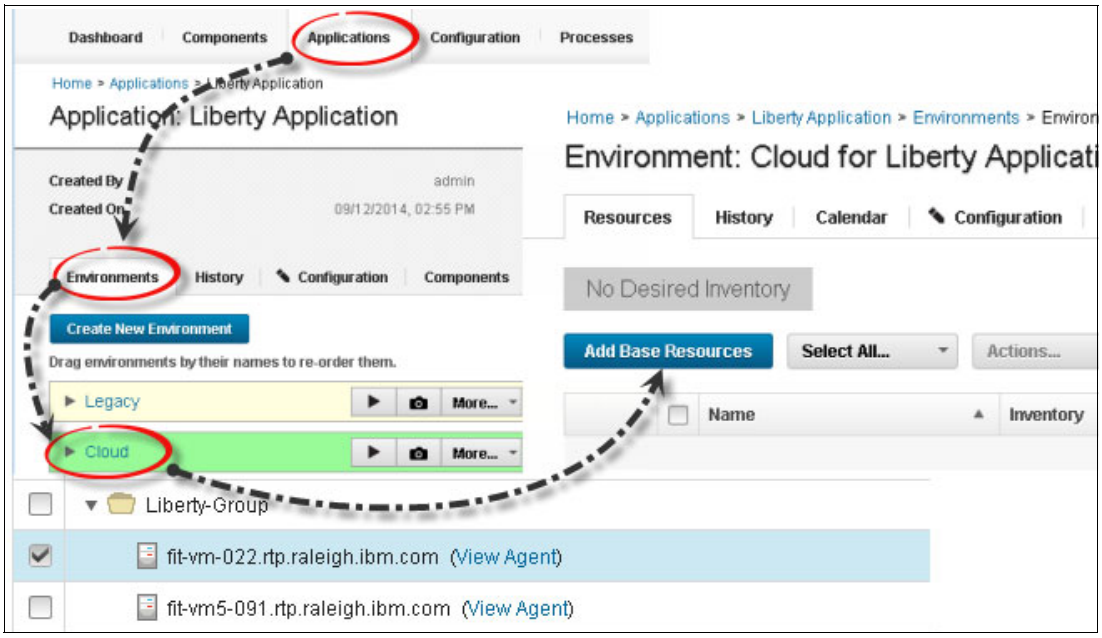


Figure 3-55 Adding resources

Deploy WAR file to the Cloud Environment

To deploy a WAR file, you must have already created an application process. This chapter assumes that you followed the steps in *Creating integrated WebSphere solutions using Application Lifecycle Management*, SG24-8243, Chapter 6, Item 6.4.

To run the application process, use the following steps:

1. Within the UrbanCode Application, click the **Application** tab.
2. Select **Liberty Application**.
3. Click the **Process** tab.
4. Select the environment called **Cloud**.

5. Click the **Request Process** button, shown in Figure 3-56

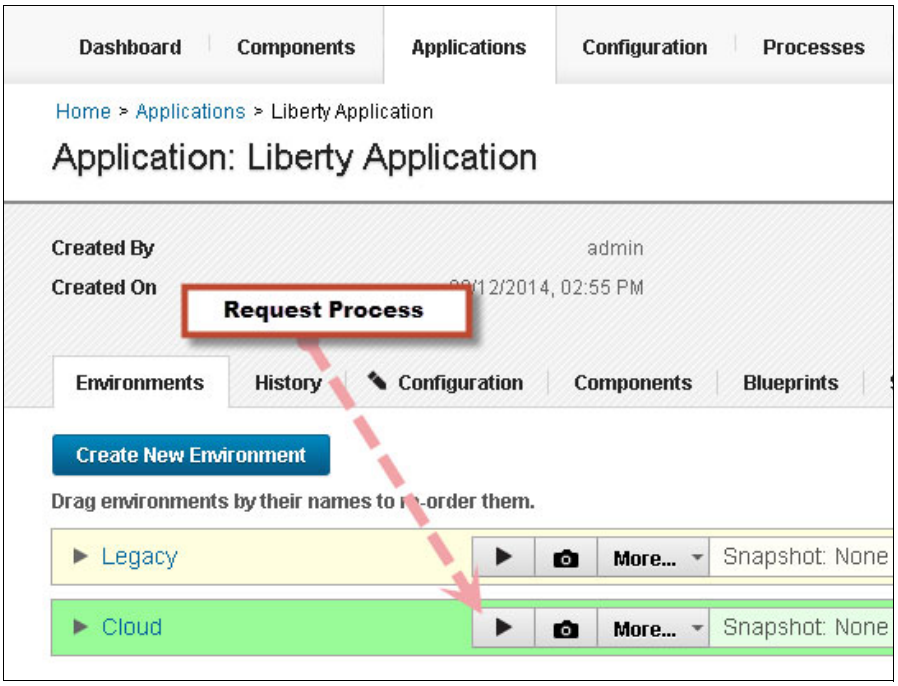


Figure 3-56 Running the application process, Request Process button

3.6 Deploying the solution

This section describes how the different parts of the solution are placed into the cloud. It also distinguishes what is different when the solution is deployed in a private cloud or on public cloud.

3.6.1 Deployment concepts

This paper covers two kinds or clouds: Private clouds and public clouds.

Private cloud

The PureApplication System provides a private cloud. The infrastructure is provisioned for an exclusive organization or department, which introduces higher levels of security and privacy into the cloud environment. The computing resources can be hosted internally or externally and managed by the organization or a third party. Usually, the access to a private cloud is restricted, with servers that are standing behind multiple firewalls and reachable only through a virtual private network (VPN).

Public cloud

In this case, the chapter uses SoftLayer where the infrastructure is made available for the general public and is offered by organizations in a no cost or pay-per-use model. The computing resources are hosted by the service provider and shared between different customers (multi-tenancy). Access to the cloud environment is provided through an ordinary Internet connection.

3.6.2 Deploy on the PureApplication System

In the PureApplication System, you have control of where you put the parts of the solution. This chapter's solution has, for security reasons, two different network zones (for example a DMZ zone and an intranet zone). In the PureApplication System, you can isolate the resource by defining different cloud groups. Figure 3-57 illustrates how to isolate the computing resource.

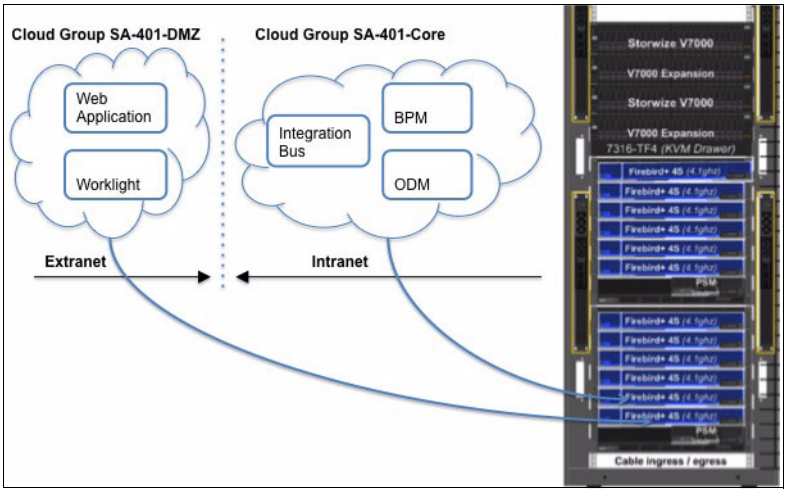


Figure 3-57 Deployment in different cloud groups

For more information about cloud computing concepts, see Section 3.1.1 of *Adopting IBM PureApplication System V1.0*, SG24-8113, at the following website:

http://www.redbooks.ibm.com/abstracts/sg248113.html

Note: This chapter's example splits the network zones by defining two different IP groups within different VLANs.

3.6.3 Deploy the patterns to the cloud group

Now, deploy the solution by choosing the pattern type, Virtual System, and creating the virtual machines to run the applications. Deploying is made simple in the PureApplication System. Navigate to Virtual System Patterns by filtering your patterns and click **Deploy**, as shown in Figure 3-58.

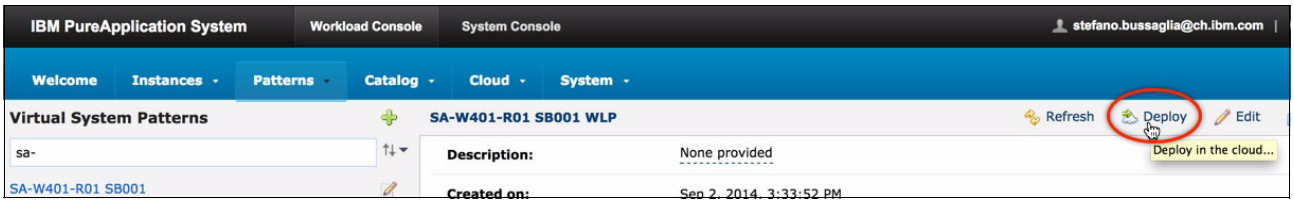


Figure 3-58 Deploying the solution

In the deployment dialog window, you can specify the specific parameters to modify the standard values, noted in the following steps and shown in Figure 3-59 on page 65.

1. Provide a relevant name for the instance that manages the virtual machines.
2. From the **Choose environment** field, select your prepared environment profile to determine the resource that is used for the deployment.

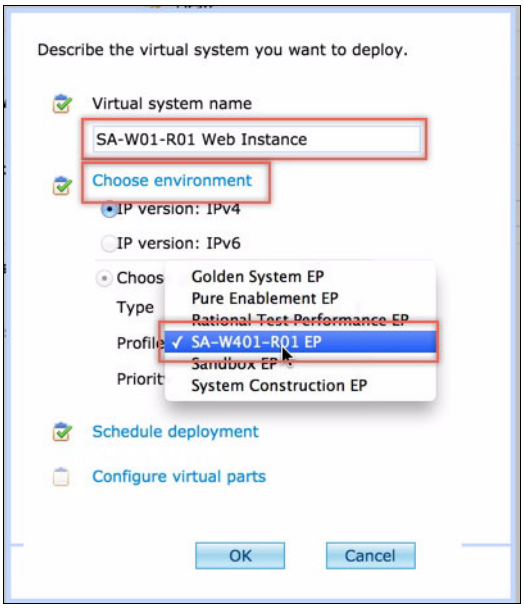


Figure 3-59 Deployment parameters

3. From the same window, if you select type **Production**, only the environment profiles are shown from that group, see Figure 3-60.

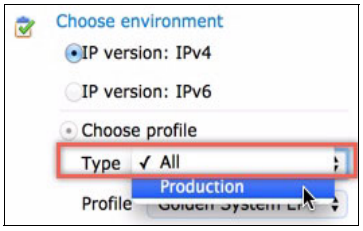


Figure 3-60 List production environment profiles

Next, configure each virtual part in your deployment. Each part results in a new virtual machine. A separate dialog appears to specify the requested parameter for each part. In this example, you want to place the web application into the Internet network zone. Therefore, select the IP addresses provided for the public sector, as shown in Figure 3-61.

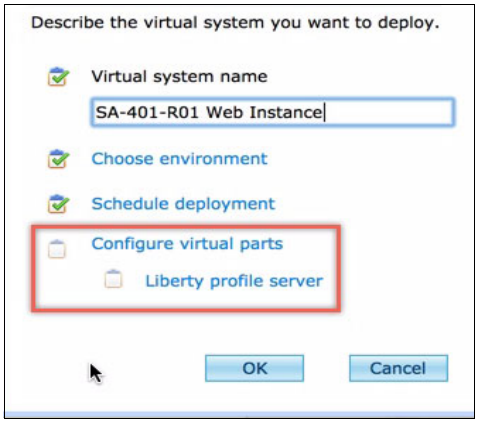


Figure 3-61 Virtual part parameters

Figure 3-62 demonstrates the selection of the IP group where the virtual machine receives an IP address. Select **public** for the internet zone and **private** for the intranet.

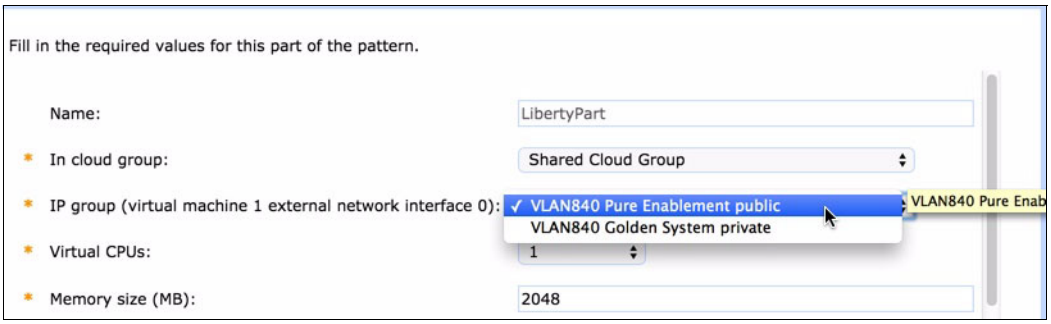


Figure 3-62 Selecting IP group for internet zone

Finally, you can generate an SSL key to access the virtual machine using SSH. Open the **Advanced** section, and select **Generate and Download,** as shown in Figure 3-63.

Figure 3-63 SSL key generation

Note: Virtual machines provisioned by Virtual Application can only be accessed with the SSL key. This is different from the Virtual System, where the virtual machine can be accessed with the user name and password.

3.6.4 Default deploy settings

Virtual Application uses a default operating system image for the virtual machine. To view the images, open the IBM PureApplication System Console and select **Cloud** → **Default Deploy Settings**, as shown in Figure 3-64.

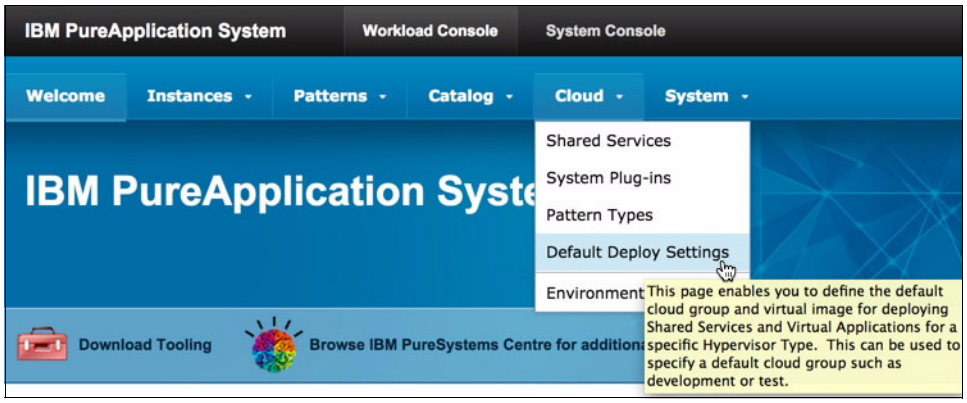


Figure 3-64 Default deploy settings of the virtual applications

In Figure 3-65, you see the OS image that the Virtual Application uses for the virtual machine. To add another default image (for example, to create a virtual machine with a different operating system), select **Add**.

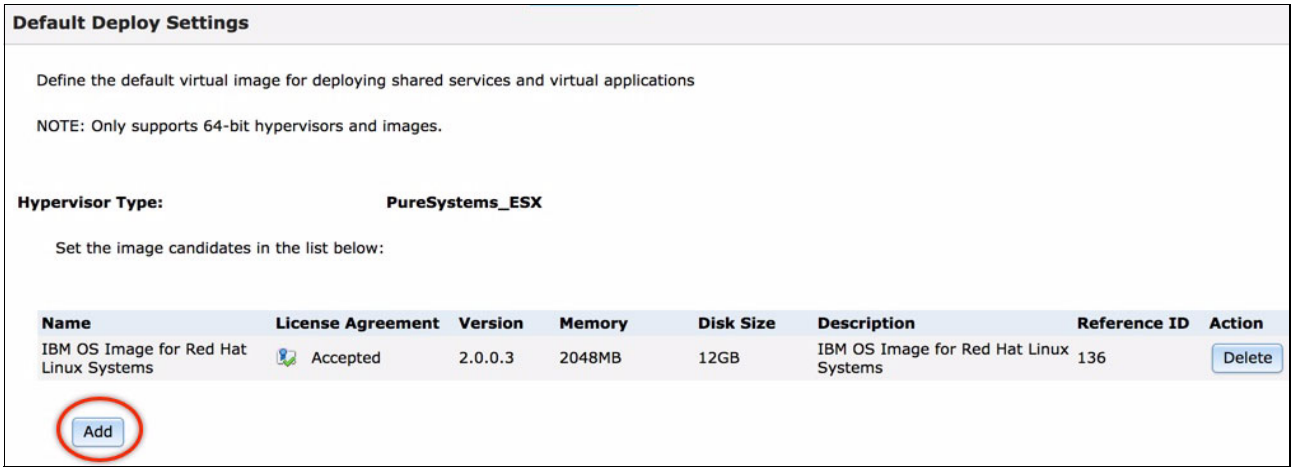


Figure 3-65 Default OS image for virtual systems

Another window opens where you can select from a list of images. These are available to add a second default deploy image, as shown in Figure 3-66.

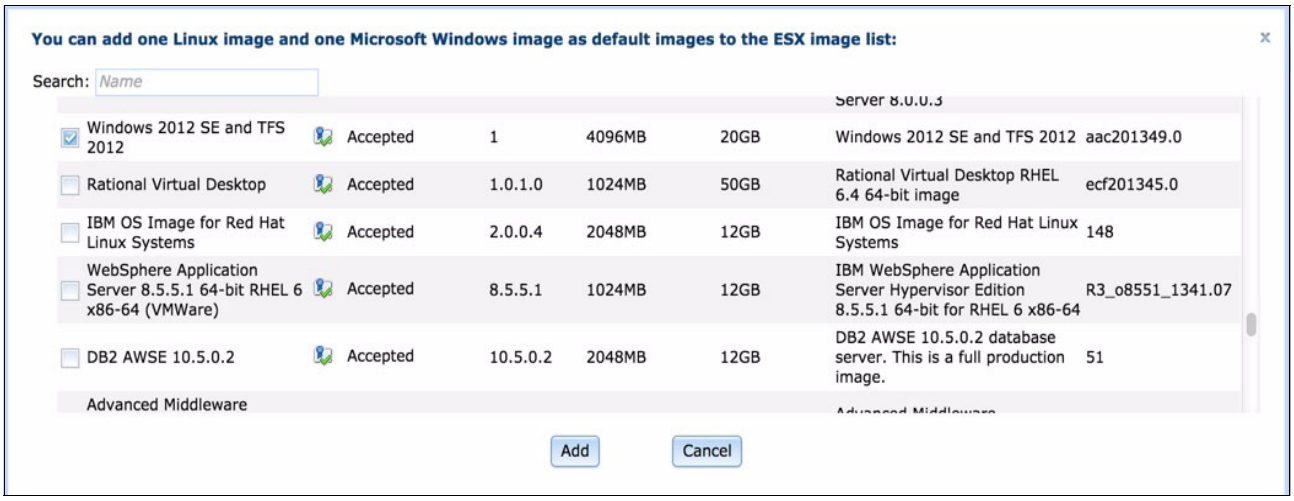


Figure 3-66 List of default deploy images

Note: You can have only one Linux and one Windows image set as default deploy.

Next, choose which of those two images you use to create the virtual machine for a virtual system instance, as shown in Figure 3-67.

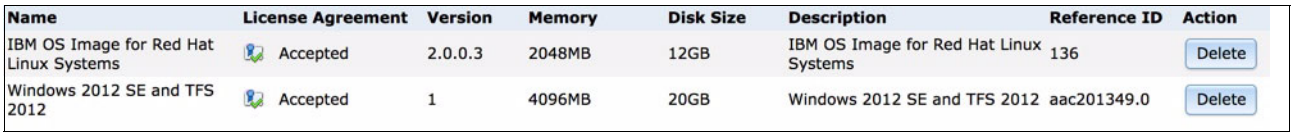


Figure 3-67 Default deploy images for virtual applications

3.6.5 Creating the instance of the Virtual Application

In the deployment dialog window of PureApplication System workload console, after you click **OK**, the system deploys the Virtual Application by creating an instance of a group of virtual machines.

To see the instance start from deployment, use the following steps:

1. Open the IBM PureApplication System workload console and select **Instances** → **Virtual Applications**, as shown in Figure 3-68.

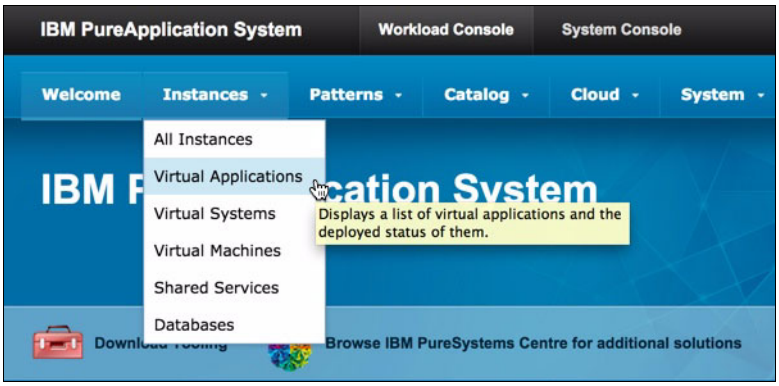


Figure 3-68 List of virtual applications and deploy status

2. Select the pattern types you want where the instance appears. The status in this example is `Preparing middleware`, as shown in Figure 3-69.

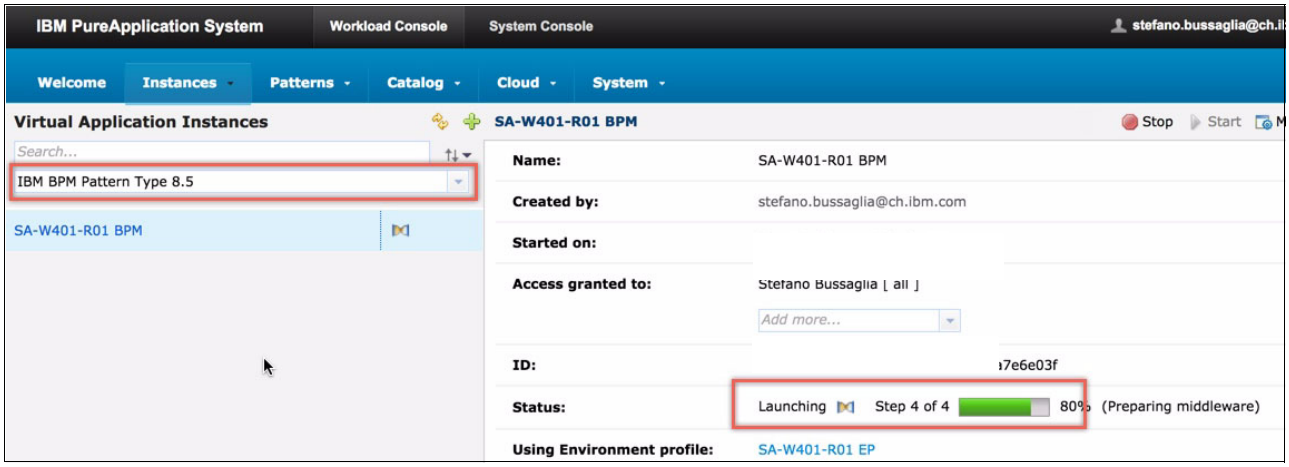


Figure 3-69 Preparing middleware status

3. Below the status indicator, you will find the Virtual machine perspective section. When you open this section you can see that six virtual machines have been created for BPM, as shown in Figure 3-70.

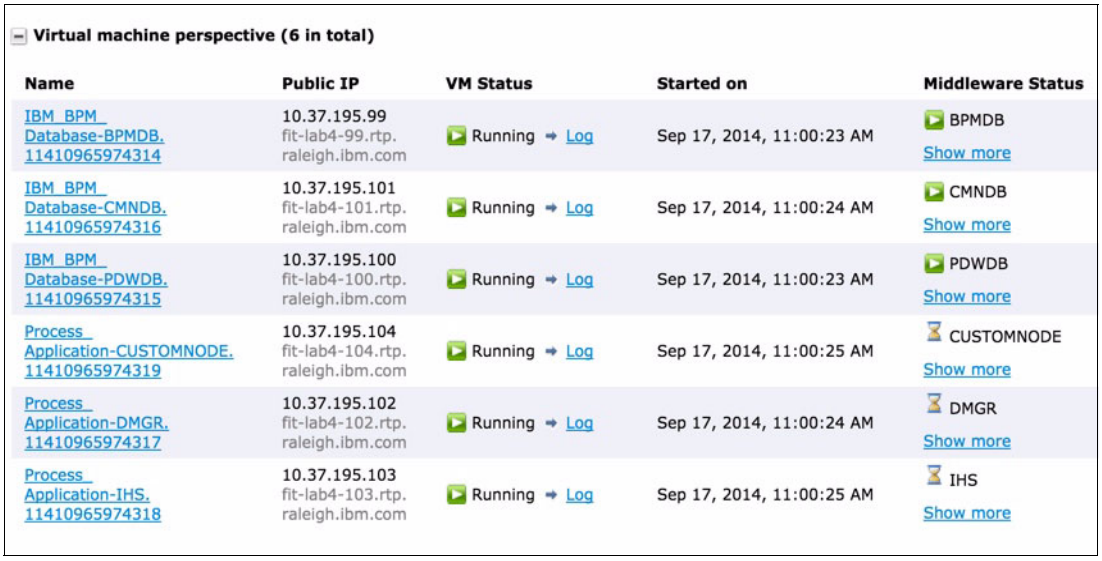


Figure 3-70 Virtual machine perspective

Note: It took less than 6 minutes to create the whole middleware for BPM with six virtual machines, as shown in Figure 3-71.

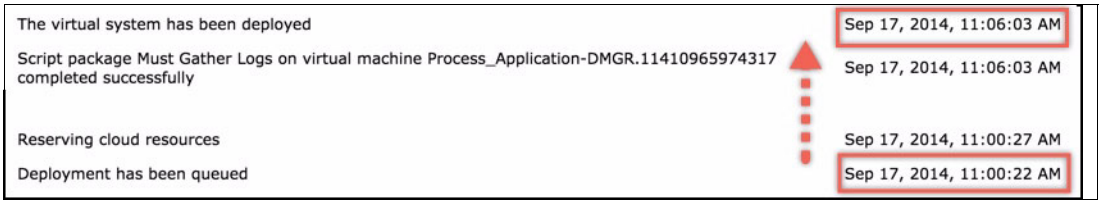


Figure 3-71 History of deployment steps

The next section covers deploying the other Virtual Systems Patterns for ODM and Mobile. The deployment can be done sequentially or in parallel.

3.6.6 Deploy an instance for Virtual Systems

The difference between deploying Virtual Systems and Virtual Applications is minimal. When deploying a Virtual System, you enter the input that is required by the script packages and the SSL key is not implemented. Virtual Applications do not use script packages.

In this next solution, the example scenario from *Creating integrated WebSphere solutions using Application Lifecycle Management*, SG24-8243, is used. It is important to mention that the DB2 database in the Integration Bus pattern receives a defined host name to link in the claim application.

Navigate to the Virtual System pattern, select the **IBM Integration Bus Virtual System Pattern**, and click **Deploy**. A dialog box to describe the deployment of the virtual system is displayed.

Use the following steps to describe the virtual system, as shown in Figure 3-72:

1. Select **Choose environment → Choose profile.** From the drop-down list, select the environment profile **SA-W401-R01 EP Pattern**.
2. Select **Configure** to choose the configurable parts.

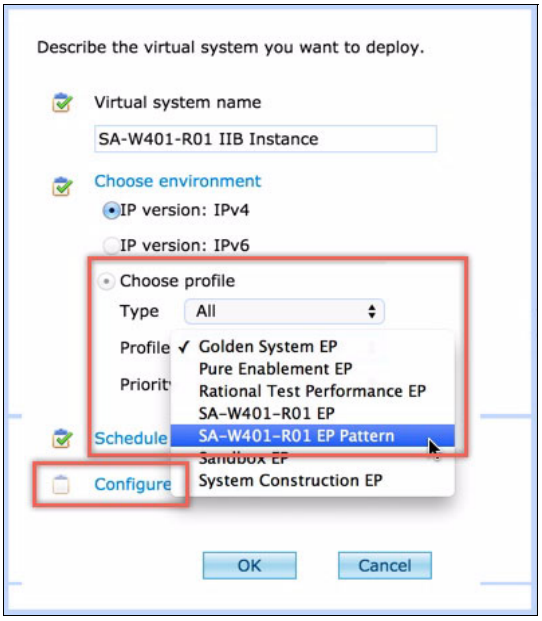


Figure 3-72 Selecting an environment profile with pattern deployment

3. Select **DB2 Enterprise** to define the host name, as shown in Figure 3-73.

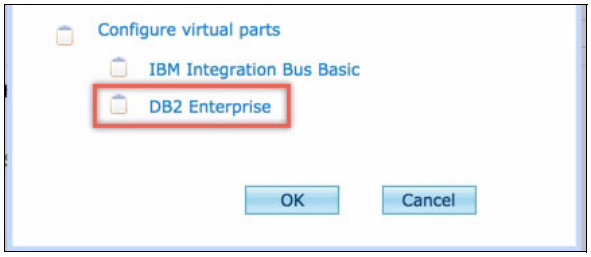


Figure 3-73 Selecting a DB2 part to specify a host name

4. Specify the *host name* and *IP address* that you want for the DB2 instance. Use the drop-down menu to select the IP group, as shown in Figure 3-74.

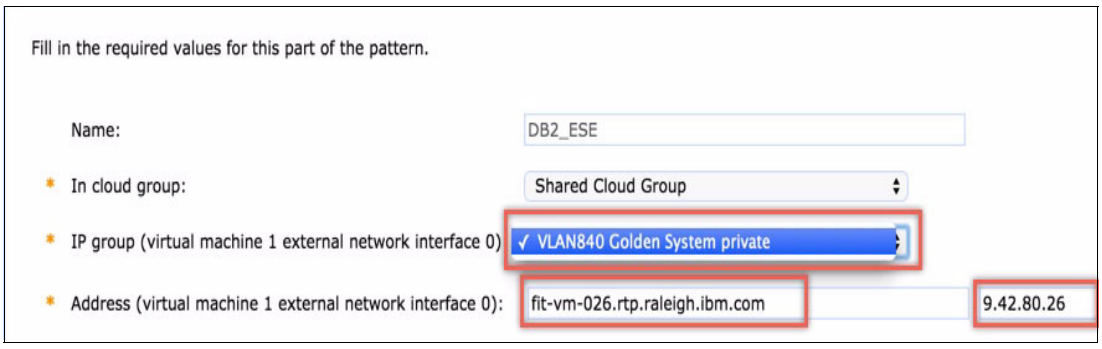


Figure 3-74 Properties of DB2 virtual machine

Note: There is only one item in the drop-down list for the IP group because you selected only one IP Group in the Environment Profile SA-W401-R01 EP Pattern.

3.7 Managing the Virtual Application instances

This section gives you a short description about the management functions of virtual instances deployed, and therefore virtual machines created.

3.7.1 Virtual System

IBM PureApplication System offers you management functions for the virtual system instance.

Virtual system instance

You must first view the instance to manage it. To view the instance you want to manage, from the Workload console, select **Instances** and select the specific virtual instance. In the menu bar of virtual system pane, you will find the management commands.

You can manage the virtual instance with the following functions:

- **Start** or **Stop**: Starts or stops the instance and its virtual machines
- **Service**: Apply a fix-pack
- **Store**: Stores all virtual machines (only active when the instance is stopped)
- **Delete**: Deletes the instance including its virtual machines

The management functions of the virtual instance are in the virtual instance pane in the menu bar in the upper right. To manage a virtual machine individually, open the *Virtual machines* view, in the lower right, and select **Manage** → **More** as shown in Figure 3-75.

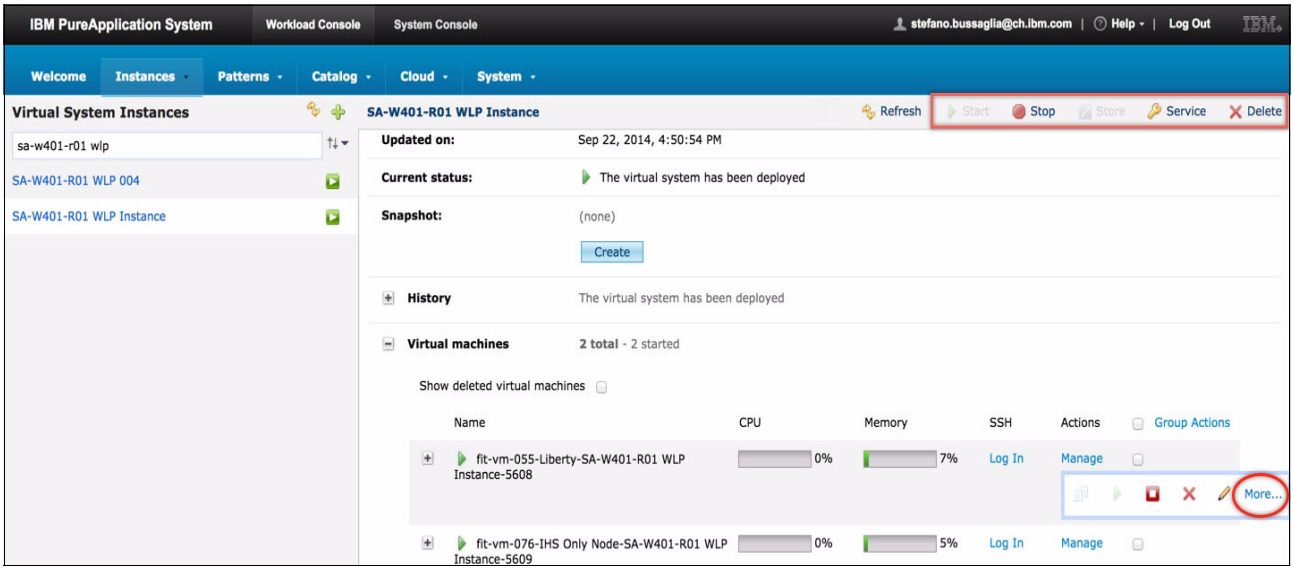


Figure 3-75 Managing a virtual instance

Virtual machine

Each virtual machine has individual management functions. Click the virtual machine that you want to change to access the *Virtual Machines* pane. The following list notes the management functions available, as shown in Figure 3-76:

- Console
- Power On
- Power Off
- Restart
- Delete
- Configure

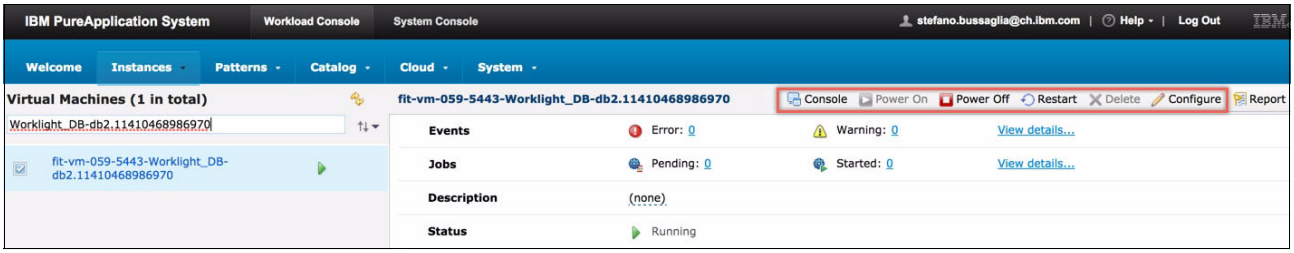


Figure 3-76 Virtual machine management functions

Configure computing resource

Select **Configure** on menu bar to open the configuration window. There you can increase or decrease CPU count or Virtual memory, as shown in Figure 3-77.

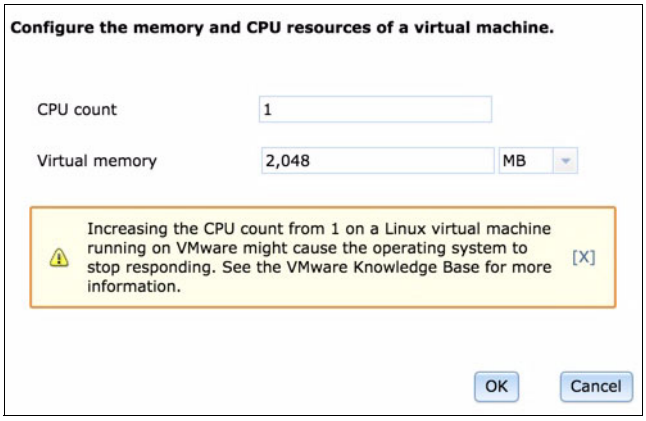


Figure 3-77 Configuring computing resource of the VM

Storage volume

In the Virtual machines window, open the **Operating system volumes** view to see the default added hard disks and their capacity. Select a system Storage volume from the drop-down menu to add to the virtual machine, as shown in Figure 3-78.

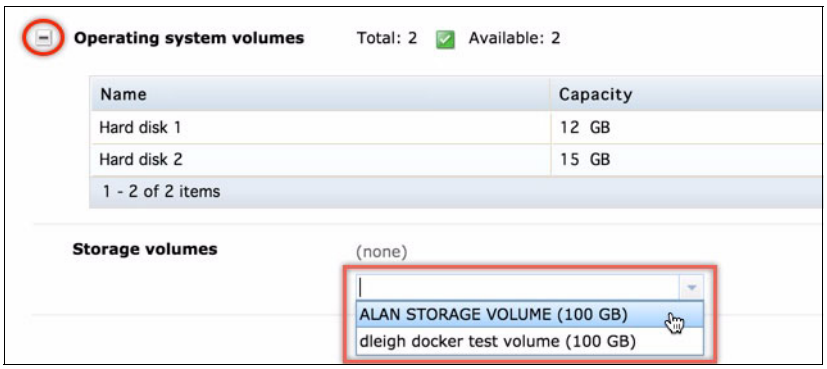


Figure 3-78 Adding storage volumes to the VM

Note: Storage volumes must be defined previously in PureApplication System before they can be added to a virtual machine. See the following link for further information:

http://pic.dhe.ibm.com/infocenter/psappsys/v1r0m0/topic/com.ibm.ipas.doc/systemconsole/t_adminstoragevolume.html

IP address

In the Operating System volume view, the IP address of the virtual machine is located beneath the Storage volumes section. There, you see the IP address of the virtual machine. If you need to change the IP address, you can remove the current address or select available IP addresses from the drop-down list, as shown in Figure 3-79.

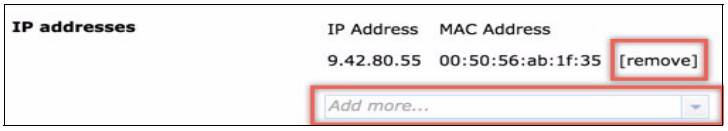


Figure 3-79 Changing the IP address of a virtual machine

Note: You can add more than one IP addresses to virtual machines. You can also select an IP address from an IP group of the used environment profile.

3.7.2 Virtual application

The virtual *application* instance is optimized for IBM PureApplication System and has extra capabilities to manage the virtual instance beyond those previously noted for the virtual system instance.

Virtual application instance

To see the management functions of the virtual application instance, open the workload console, and in the **Instance** menu, select the **Virtual Application Instance**. The **Manage function** button at the top of the view) is added to the virtual application instance management capabilities as shown in Figure 3-80.

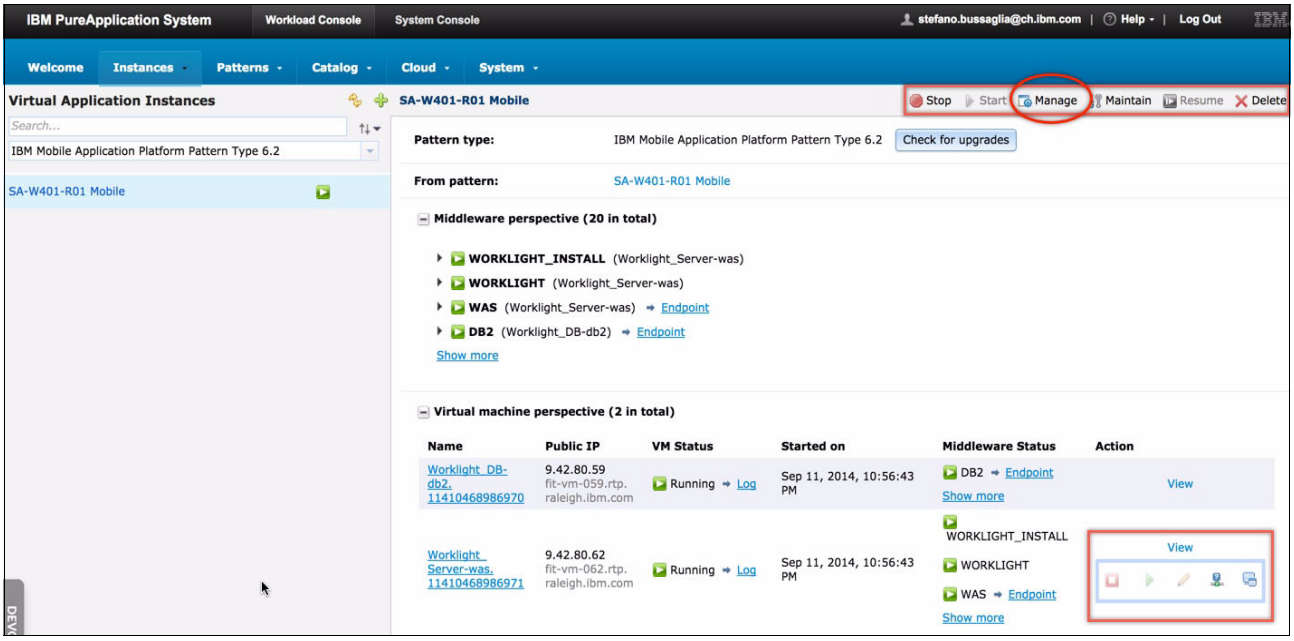


Figure 3-80 Management functions for the virtual application instance

After you click **Manage**, a new tabbed window appears in the browser with the Virtual Application.

For more information about the management capabilities for Virtual Applications, see the following website:

http://www-01.ibm.com/support/knowledgecenter/SSNLXH_1.0.0/com.ibm.ipas.doc/workloadconsole/t_mngpatterns.html?lang=en

Operations

In the Virtual Application Console, you can review the operations available for the virtual application instance.

To access the Virtual Application Console, click **Manage** and accept the HTTPS connection to open a new browser tab, as shown in Figure 3-81.

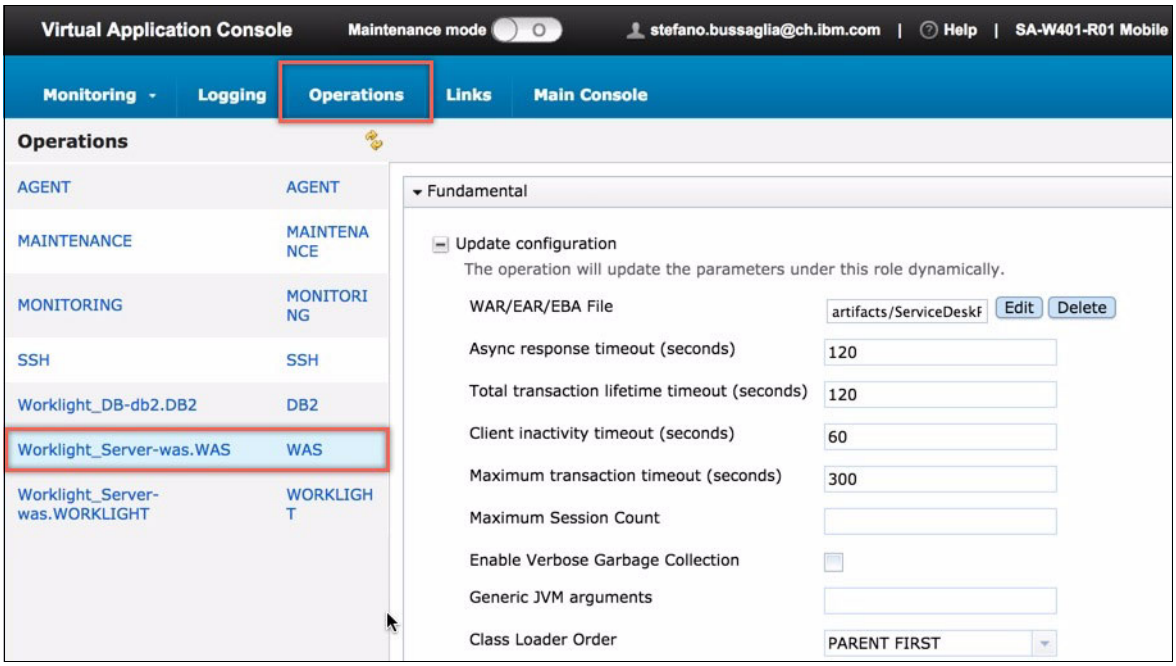


Figure 3-81 Virtual application console operations

To work with the fundamental operations of the virtual application instance, select the **Operations** tab and then select the instance for the WebSphere Application Server agent of the Worklight Server, shown in Figure 3-81.

One useful operation is adding another SSH public key to access the virtual machine, as shown in Figure 3-82.

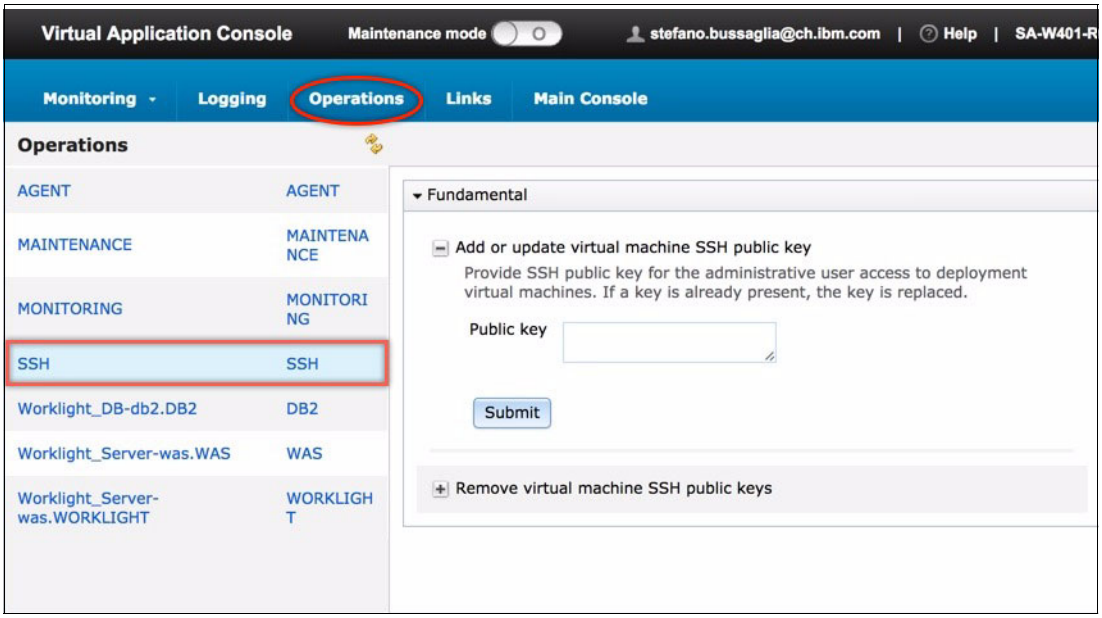


Figure 3-82 Managing the SSH public key

For more information about extra operations available in virtual application, see *Cloud Computing Patterns of Expertise*, REDP-5040, which is available at:

http://www.redbooks.ibm.com/redpapers/pdfs/redp5040.pdf

Middleware perspective

In the Virtual Application Instance window, there is a middleware perspective of the virtual application instance. Figure 3-83 shows this view.

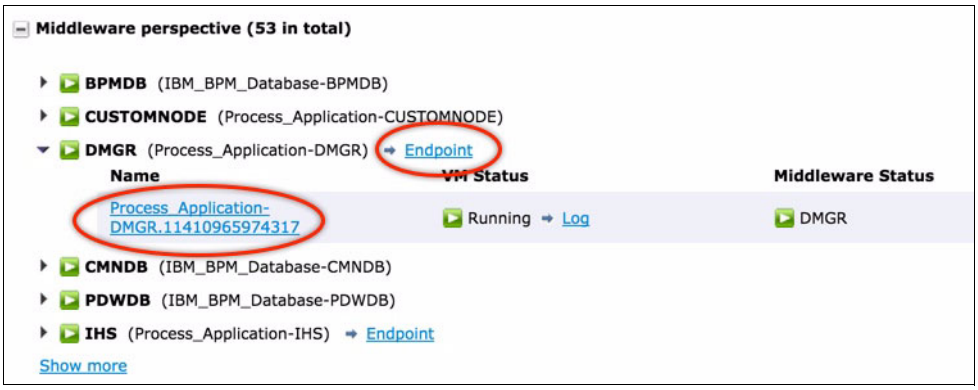


Figure 3-83 View of middleware perspective

Some middleware components have an *Endpoint* link for direct access. Clicking this link gives access to the ***Virtual machine's* pers**pective to view the group of virtual machines used for the middleware, also shown in Figure 3-83.

Virtual machine

Management of the virtual machine of the virtual application instance is the same as for the virtual system instance (see "Virtual system instance" on page 72).

The Claim solution is now deployed in several instances of virtual systems and virtual applications, and can be accessed in the cloud.

4

Monitoring

This chapter describes the integrated monitoring capabilities of IBM PureApplication System. These capabilities help you to achieve these goals:

- Identify and accelerate root cause analysis and problem determination
- Achieve valuable insight into application and system health

This chapter includes the following sections:

- Monitoring overview of PureApplication System
- Monitoring examples

The PureApplication System provides comprehensive features for monitoring system activities. Integrated components provide a single place to track all events and check the status of the system. The system provides some standard interfaces that help to integrate your system into an existing monitoring infrastructure.

For more information about monitoring, see Chapter 2 in *Adopting IBM PureApplication System V1.0*, SG24-8113, which is available at:

http://www.redbooks.ibm.com/abstracts/sg248113.html

4.1 Monitoring overview of PureApplication System

This section describes the integrated monitoring capabilities of the PureApplication System, and identifies how this monitoring accelerates root cause analysis and problem determination.

4.1.1 Business value

The PureApplication System provides integrated monitoring for aspects from system hardware through running workloads and maintaining databases:

- Hardware
- Operating system
- Middleware
- Application
- Database

With PureApplication System integrated monitoring, no installation or configuration is required to begin using the system.

Monitoring delivers insight into the operations and performance of your system in these ways:

- Facilitating problem determination and root cause analysis
- Providing information to assist with understanding current utilization and planning for future capacity needs

4.1.2 Role based drill-down

Different roles have the ability to drill down to their related monitoring tasks:

- Deployers:
  - Can drill down only into their middleware and database metrics
- Monitor operators:
  - Allow access to all user deployments
- Monitor administrators of cloud and hardware:
  - Have Operator access
  - Allowed to see all hardware

4.1.3 Monitoring layers

With the PureApplication System, the monitoring of system, middleware, and hardware layers enables you to gather information about the status, performance, and resource use for both hardware and software. Figure 4-1 shows the monitoring tasks of each of these layers.

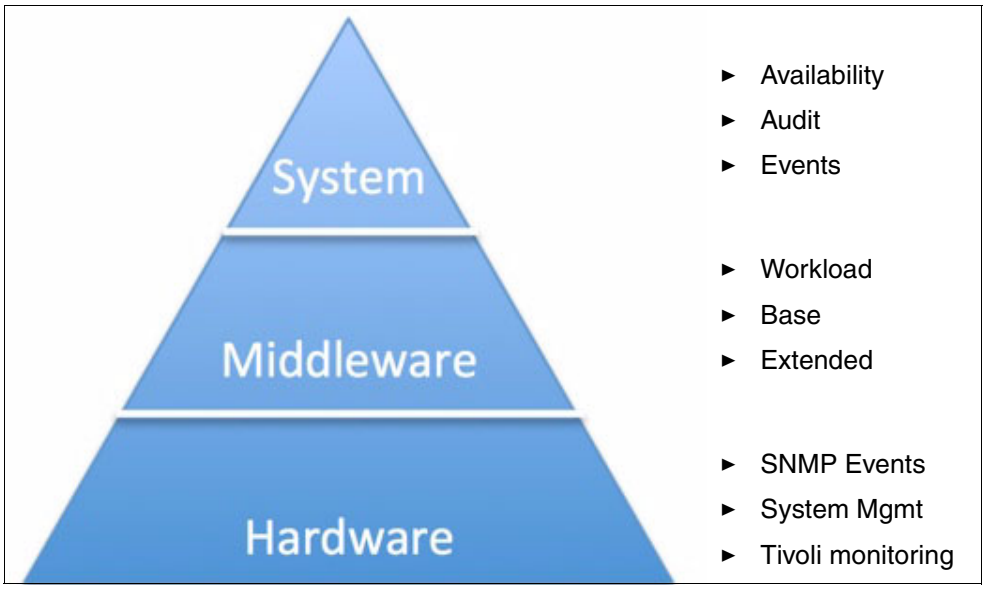


Figure 4-1 Monitoring layers

Monitoring the system

To view monitoring data, you can use the monitoring agent for the PureApplication System or install it on a computer that is not part of the product. The Monitoring Agent for the PureApplication System is deployed automatically with the System Monitoring shared service.

Monitoring using the System Monitoring shared service

When you deploy the System Monitoring shared service, an instance of the Monitoring Agent for the PureApplication System is deployed and configured automatically. You can view the data in the PureApplication System Monitoring portal. The advantage of using this option is that the process is relatively simple and does not require an existing IBM Tivoli® Monitoring environment.

Monitoring from a remote environment

Another option is to install the Monitoring Agent for the PureApplication System on a computer that is not part of the product. 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. The advantage of installing the monitoring agent outside of the product is that the agent can collect statistics for hardware resources and also for deployments in cloud groups in the product.

Monitoring the middleware

You can monitor workloads and instances, for example virtual system instances and virtual appliance instances, with PureApplication System.

Facilities for monitoring workloads and instances are accessible from the Workload console. For the monitoring facilities to be displayed in the Workload console, you must deploy the monitoring shared services.

Instances are managed by the PureApplication System, and can be serviced and accessed through the console. The following instances are dynamic and are managed by the PureApplication System:

- Monitoring Agent for IBM WebSphere Application Server

  The Monitoring Agent for the IBM WebSphere Application Server provides a systems management solution for multiple versions of the WebSphere Application Server. Using this monitoring agent, you can monitor multiple WebSphere Application Servers running on the same physical node. Each application server must be configured with its own Monitoring Data Collector for WebSphere.

- Monitoring Agent for Workloads

  The Monitoring Agent for Workloads provides the capability to monitor tasks that are specific to the workloads that you deploy.

- Monitoring Agent for IBM HTTP servers

  The Monitoring Agent for IBM HTTP servers provides a systems management solution for your web servers for distributed platforms. Using the Monitoring Agent for IBM HTTP servers, you can monitor multiple web servers that are running on the same physical node.

- Database Performance Monitoring

  Use Database Performance Monitoring to set and manage database alerts, and to monitor database performance metrics.

Monitoring the hardware

You can monitor various hardware devices that are included in the system. Facilities for monitoring hardware are accessible from the System console.

You can also use the PureApplication System Monitoring Agent to monitor hardware resources:

- Monitoring compute nodes:

  Monitor compute node components, such as microprocessors, memory, and Ethernet controllers. You can monitor the compute nodes in the system for details, including CPU resources, memory usage.

- Monitoring management nodes:

  You can use management nodes to monitor details about the system.

- Monitoring Flex System Enterprise Chassis:

  In the left pane of the console, you can view a list of all chassis. The details for each chassis include events, jobs, type, status, machine type, energy information, location, temperature, and LEDs. There are expandable tables for the various hardware components that exist in each chassis, including compute nodes, Chassis Management Modules (CMMs), network switches, management nodes, power supplies, and fans.

- Monitoring storage devices:

  By monitoring the storage devices in your system, you can view various details, such as associated events and jobs, firmware levels, status, capacity, and the use and allocation of disks and volumes.

- Monitoring network devices:

  Monitor the status and attributes of the network devices on your system. Regular monitoring helps to ensure that you are up-to-date on any changes to the system.

- Monitoring virtual networks:

  Monitor virtual networks on the system to ensure that they function correctly.

4.2 Monitoring examples

The following are some examples of the PureApplication System monitoring capabilities.

After business process management (BPM) patterns are deployed, the environment has four virtual machines (Figure 4-2).

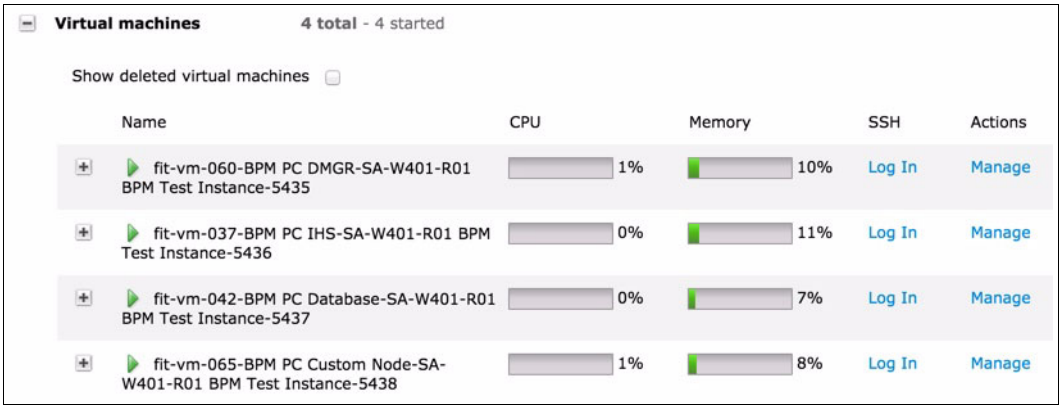


Figure 4-2 List of virtual machines

For each virtual machine, there are several options for gathering monitoring information.

During deployment, you can inspect the script packages to determine whether they finished successfully. If not, you can directly watch the `stdout` or the `stderr` log file from each script package (Figure 4-3).

Figure 4-3 Inspection of script packages

When deployment is complete, you can inspect the health of the virtual machine. To learn more about the health of a virtual machine, click **Manage** → **More** to view the next menu level (Figure 4-4).

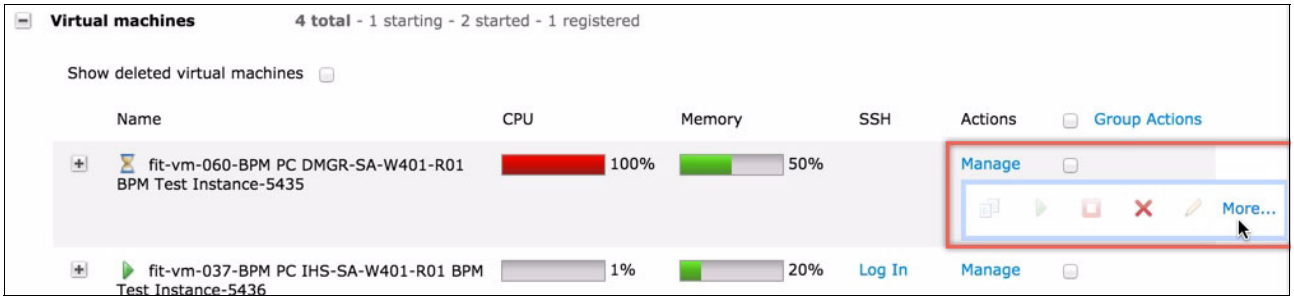


Figure 4-4 Virtual system health check

The next window provides an overview for the selected virtual machine. To view reports about the virtual machine, click **Report** (Figure 4-5).

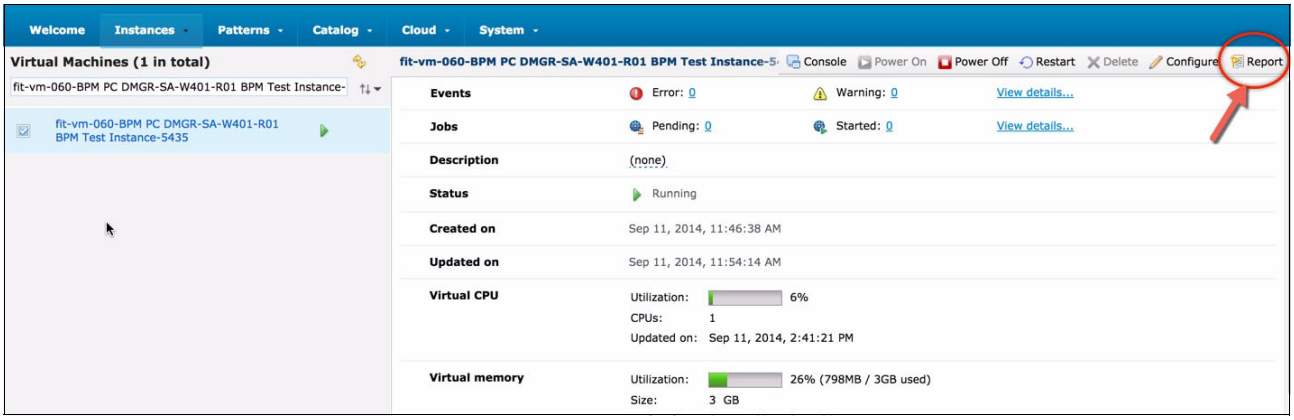


Figure 4-5 Generating reports

Figure 4-6 shows the CPU consumption of a virtual machine.

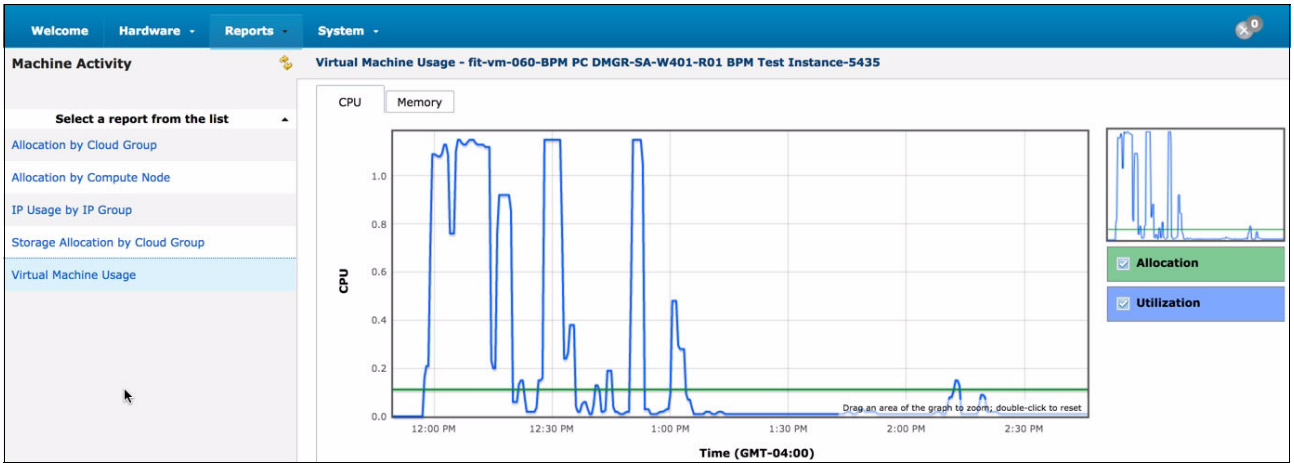


Figure 4-6 CPU consumption

4.3 External monitoring

The PureApplication System has a System Monitoring feature called the External System Monitoring service. This feature allows you to push the workload monitoring agent data to an external IBM Tivoli Monitoring data center. This makes the integration of the PureApplication System with the data center tighter and smoother.

For more information about the External System Monitoring service, visit:

http://www-01.ibm.com/support/knowledgecenter/SSCR9A_2.0.0/doc/ITM_shared_service/r_itm_shared_service.dita

5

Licensing

Many of the products that can be installed on IBM PureApplication System require proper licensing. The PureApplication System provides options that help with monitoring and administering your licensed products, both IBM and non-IBM products.

This chapter describes how to define, manage, and control licenses for the products installed on the PureApplication System. It describes the available reports and how to use them to charge departments in your organization with applicable license fees.

This chapter includes the following sections:

- Product license administration and control
- Reports

5.1 Product license administration and control

You can administer and monitor product licenses with PureApplication System. This can be done for products that are bundled with the system (for example, WebSphere Application Server, DB2, and HTTP servers) and for products that are purchased independently.

The license awareness feature of the PureApplication System prevents you from exceeding your allowed licenses, and notifies you when you are close to your license limit.

The PureApplication System reporting feature displays your license usage. For those with access to the subcapacity license contract that makes the IBM License Metric Tool available, you can report system usage with this tool as well.

Note: Users who are permitted to use the IBM License Metric Tool will find it in the Workload Management component of the PureApplication System, and can track server licenses on the system. Only those users who are assigned the Cloud Group Administrator role or the Hardware administration role with permission to manage hardware resources (full permission) can access this tool.

In the PureApplication System, license information is refreshed every 5 minutes, so changes made might not be reflected immediately.

5.1.1 License awareness

To monitor license usage and to confirm that the system does not exceed the number of owned licenses, license awareness needs to be configured.

Note: You can use the system console, the command-line interface (CLI), or the Representational State Transfer (REST) API to complete this task. The method that is shown as an example in this paper is based on using the System Console. For details about the other two possible ways to operate, see the specific documentation on the following website:

http://pic.dhe.ibm.com/infocenter/psappsys/v1r1m0/index.jsp?topic=%2Fcom.ibm.puresystems.appsys.1500.doc%2Fwelcome%2Fwelcome.html

To configure license awareness, complete these steps:

1. Access the System console.
2. Select **System** → **Product Licenses**.

3. In the License awareness section, select **Notify virtual image owners when license usage reaches the thresholds set below** (Figure 5-1).

Figure 5-1 Selecting to notify virtual image owners when license usage reaches thresholds

4. For IBM products, navigate to the IBM Software Catalog product licenses tab and expand every product that is listed.
5. Set the number of owned licenses in the **Licenses owned field** to tell the system the exact maximum number of licenses it can use during the autonomic creation and allocation of the virtual instances according to both the workload and the scaling requests. The unit of measurement is expressed in processor value units (PVUs), which are the standard way to quantify software licenses purchase (Figure 5-2).

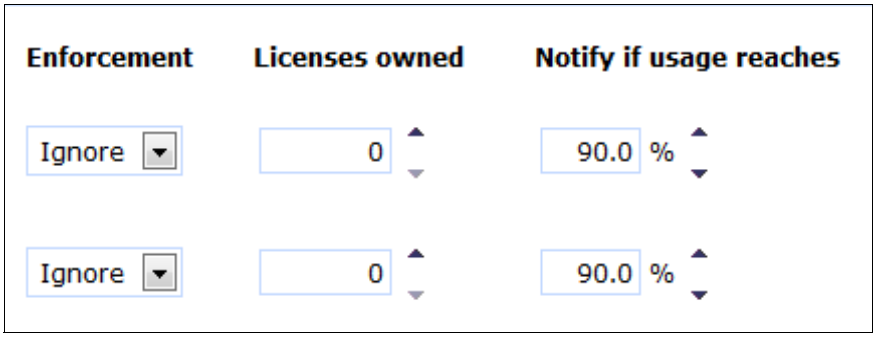


Figure 5-2 Setting licenses thresholds

Note: Because the displayed products are from the PureApplication System catalog, it is good practice to periodically download and import the catalog so that the system has the latest licensing data. Open the IBM Software Catalog product licenses tab and expand the **Update IBM Software Catalog** section. Click **Download IBM Software Catalog** to download the catalog and import it by clicking **Browse**. Finally, click **Update**.

6. For each product listed, set the **Notify if usage reaches** field. This field specifies the percent of licenses that are used per licenses owned. Surpassing this threshold triggers a notification to all users who are assigned the **Create new catalog content** permission.
7. Set the **Enforcement** field for each product listed. There are three possible values:
   a. **Ignore:** No enforcement action is taken. Deployments continue without obstacles, but license usage is still monitored.
   b. **Warn:** An error is logged in the audit logs, and a warning message is added to the virtual system instance history. Email notifications are sent to communicate the warning. Deployments continue without obstacles, but license usage is still monitored.
   c. **Enforce:** Deployments of new virtual system instances or virtual machines fail with allocation errors due to the lack of available licenses. Email notifications are sent to communicate the unsuccessful deployment.

After each product is configured, the usage of the virtual systems distributed by the PureApplication System is displayed. If the license threshold is exceeded, notifications are issued.

If you need to monitor and control the use of products that are not available in the form of PureApplication patterns, or not present in the IBM Catalog, navigate to the *Non-catalog product licenses* tab and add licenses for those products. Then, configure them by following Step 5 through Step 7.

Note: With specific reference to the PureApplication System solution in this paper, from a product licenses point of view, it is important to note these considerations:

- IBM WebSphere Application Server Network Deployment Hypervisor Edition (which contains Liberty license) and IBM DB2 Enterprise Server Edition are included in the price of the PureApplication System. *Therefore, PureApplication provides unlimited licensing for all cores installed on the system.*
- All other software components of the solution must be expressly purchased. For each of them, a specific pattern exists:
  - IBM Mobile Application Platform Pattern
  - IBM Operational Decision Manager Virtual Application Pattern
  - IBM Business Process Manager Pattern
  - IBM Integration Bus Hypervisor Edition

These patterns can be bought for all cores on the system or, alternatively, for a subset of the cores, according to the subcapacity contract. In both cases, the number of PVUs associated with the cores needs to be configured in the license awareness feature as explained in 5.1.1, “License awareness” on page 88.

5.2 Reports

The PureApplication System offers various reports that can be downloaded and used for planning purposes. The following reports are available. All of them are accessible by selecting the Reports tab in the Workload console (Figure 5-3).

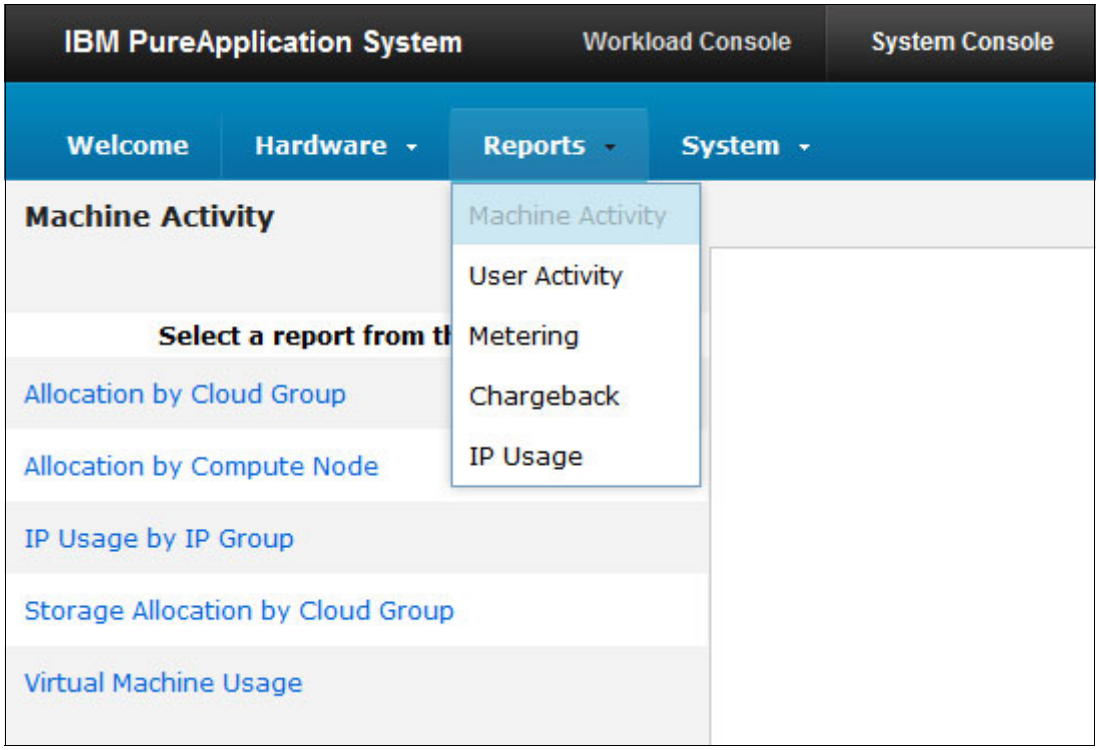


Figure 5-3 Reports tab

- **Machine Activity** reports: These are system usage reports that are used to track the physical and virtual resources that are used in the cloud. The following subsets can be generated:
  - **Allocation by Cloud Group**: This report shows the data for tracking application resources, historic usage patterns, and averages and future trends of cloud groups. You can view resource allocations categorized by CPU, memory, and instances.
  - **Allocation by Compute Node**: This report shows historic usage patterns, averages, and future trends of CPU and memory use, and the associated cloud group of all compute nodes on the rack. You can use the report to identify compute nodes that are not used at their capacity and those compute nodes that are overloaded. Resource allocations are categorized by CPU and memory.
  - **IP Usage**: This report shows the IP address usage by IP groups in the system. This report provides filtering capabilities for easy problem determination.
  - **Storage Allocation by Cloud Group**: This report shows the storage allocation by cloud group in the system. From this report, you can quickly find resources that are allocated to an application and how critical those resources are in terms of business affect. It is important to point out here that if utilization is less than or higher than the established limits in your organization, you can rebalance regarding CPU and memory.
  - **Virtual Machine Usage**: With this report, you can inspect, customize, and print reports for virtual machines and review the details, including CPU utilization, memory allocation, and disk allocation.

- **User Activity**: This report shows the resources that are used by each user account and include active virtual machines, reserved storage, reserved CPU, and reserved memory. The User Activity report documents the physical and virtual usage that is in bad condition during the specific time and date range set by the person who submits the report generation request.
- **Metering**: This report is available to those who have a specific contract agreement with IBM for CPU metering. These reports can display CPU usage over a specific period. Based on the agreed billing cycle (weekly, monthly, or quarterly), the Metering Report is sent to IBM and the user is billed based on CPU consumption.
- **Chargeback**: This report shows the usage of resources based on the metrics that have been selected. Chargeback is a model that assigns the costs of IT services to the business unit that is using those services. IT chargeback systems are intended to shift responsibility to users and get them to treat IT services as a utility, therefore encouraging them to moderate use.
- I**P Usage**: This report displays a list of resources that are available and their status per IP address. Visible details include the IP group that the address belongs to, the instance name, the host name, and the status of that instance.

Note: The ability to view reports is contingent upon roles and access. The following roles are required for each report type:

- Machine Activity report: The Cloud group administrator authorization role or the Hardware administrator role with read-only access for the various resources is required.
- User Activity report: The Cloud administration role is required.
- Metering report: Administrator authorization is required.
- Chargeback report: The Security administration role with permission to view users and groups (read-only) or the Workload resources administration role with permission to view all workload resources (read-only) is required.
- IP Usage report: The Cloud administration role is required.

6

Maintaining the solution

This chapter introduces the concepts of upgrades, emergency fixes, and backups, specifically regarding IBM PureApplication System.

It also explains how to maintain a solution that is delivered by using the Expert Integrated System. For information about the components and resources the business scenario is composed of, see 3.3.7, “Script packages” on page 50.

Note: The general term *upgrades* refers to system updates, virtual images updates, and virtual patterns updates. These different types of upgrades are detailed throughout this paper.

This chapter includes the following sections:

- Upgrades
- Emergency fixes
- Backup and restore

6.1 Upgrades

To maintain your existing PureApplication System environment, download and install the latest available fixes and updates.

6.1.1 System updates

The system maintenance task includes an automated process that applies updates and ensures that the virtual machine instances that were previously created on the system remain in their original state and are still available without requiring user interaction. This automated process applies updates to management software and hardware firmware in a single fix pack. During the fix pack installation, updates are applied to management components, VMware vCenter, network switches, storage firmware, compute node firmware, and so on, to ensure that all versions are compatible. However, not every system fix pack includes updates for all components in the system. Therefore, the entire process might not be required from release to release.

To download the System Fix Packs, follow these steps:

1. Navigate to the IBM PureSystems® Center at:

   https://www.ibm.com/software/brandcatalog/puresystems/centre/

2. Click the **PureApplication System** button (the button will change from gray to green) and then click **System Updates**.
3. Click the tab (either W1500 or W1700) that applies to your PureApplication System.
4. Expand the section for your version of the system (for example, PureApplication System V1.1.x Updates).
5. Click the System Fix Pack that you are interested in. Some links point to the location where you can directly download the packages, and other links to a document with download instructions. Usually, you are redirected to the IBM Fix Central website at the following link:

   https://www.ibm.com/support/fixcentral/

   It requires authentication if you are not authenticated yet. Follow the instructions and then click **Download**.

A system update must be installed and completed by an IBM specialized technician.

6.1.2 Virtual images and patterns updates

For updating virtual images and patterns, fixes and upgrades are downloaded from the PureSystems Center, mentioned in 6.1.1, “System updates” on page 94. From that website, select **Group Fix Updates**.

You can select some or all of the independent fixes, as needed. The group fix contains updates for pattern types, add-ons, and shared services. When you download a group fix, all of the `.tgz` files with updated add-ons and shared services are also downloaded.

Update the base operating system virtual images

When an update to a base operating system virtual image is downloaded, and it is a newer version, it is downloaded as a new virtual image. The virtual image must be imported into the catalog, and then used as base image for new deployments.

Important: To perform this task, the user needs to be in the Create new catalog content role or the Workload resources administration role, and have full permissions.

To update a base operating system virtual image, complete these steps:

1. From the Workload console, click **Catalog** and select **Virtual Images** (Figure 6-1).

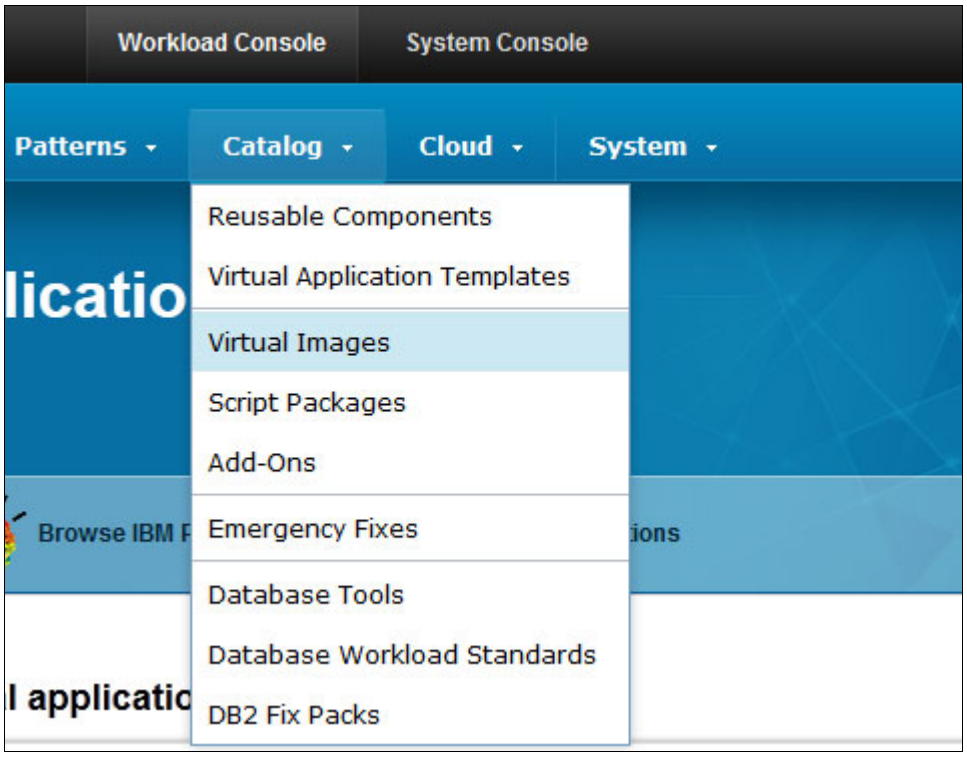


Figure 6-1 Updating an operating system virtual image

2. Click the **Green Plus Sign** on the toolbar to upload the new virtual image (Figure 6-2).

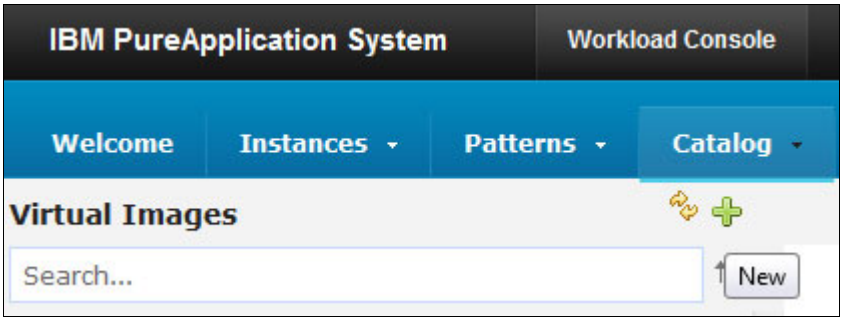


Figure 6-2 Updating an operating system virtual image

3. Click **New**, and a window displays on the canvas. Enter the remote path of the virtual image (in OVA format) to import. This field is required. The **User name** and **Password** fields are optional.

4. Click **OK** (Figure 6-3).

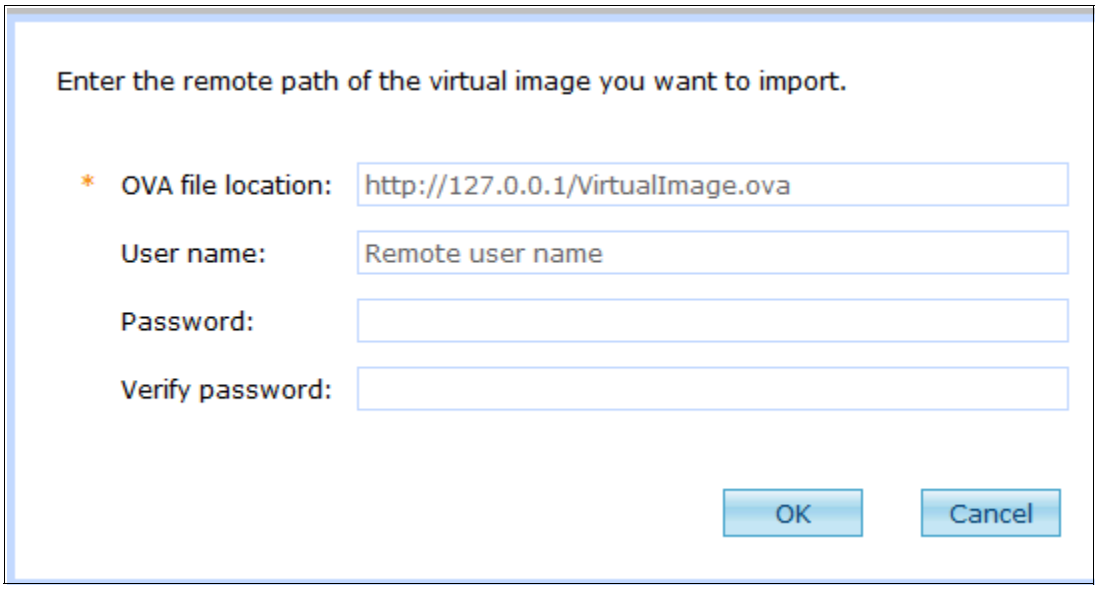


Figure 6-3 Updating an operating system virtual image

Update virtual application patterns

This section describes how to update virtual application patterns.

Important: Before patterns can be updated, your PureApplication System must be upgraded to the latest system update maintenance.

The IBM Foundation Pattern must be updated before any other patterns are updated. When you update it, you must also update the DB2 with BLU Acceleration Pattern.

You must be assigned the Create new catalog content role or the Workload resources administration role with full permissions to complete this task.

To update virtual application patterns, complete these steps in the Workload console:

1. Click **Cloud** → **Pattern Types** (Figure 6-4).

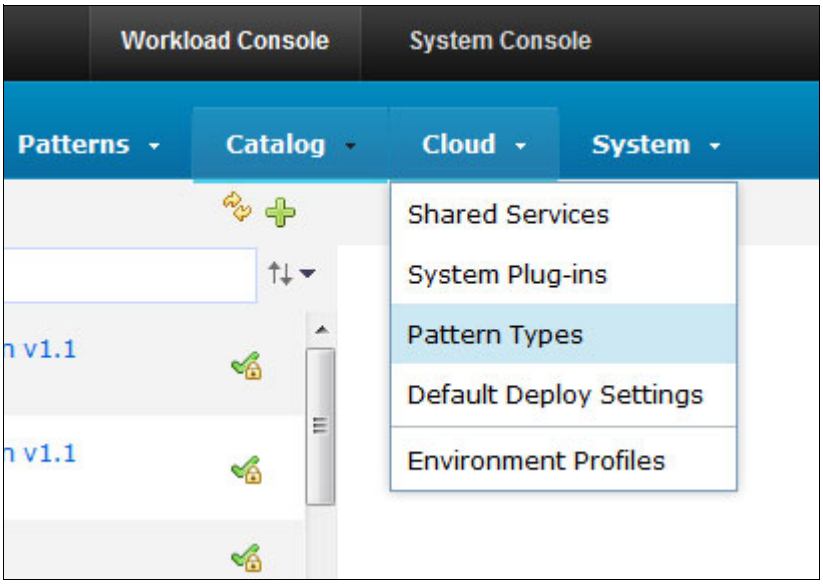


Figure 6-4 Updating virtual application patterns

2. In the next window, click the **Green Plus Sign** on the toolbar to install a new pattern (Figure 6-5).

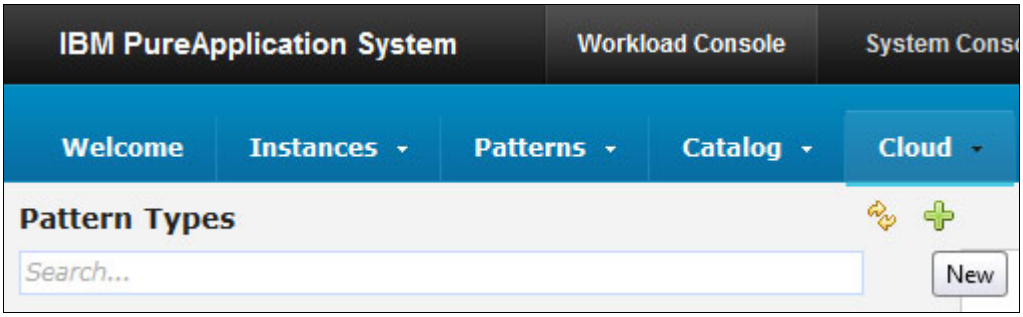


Figure 6-5 Updating virtual application patterns

3. Click **New**, and a window is displayed on the canvas. Browse for the .tgz file to import as a pattern type and click **OK** (Figure 6-6).

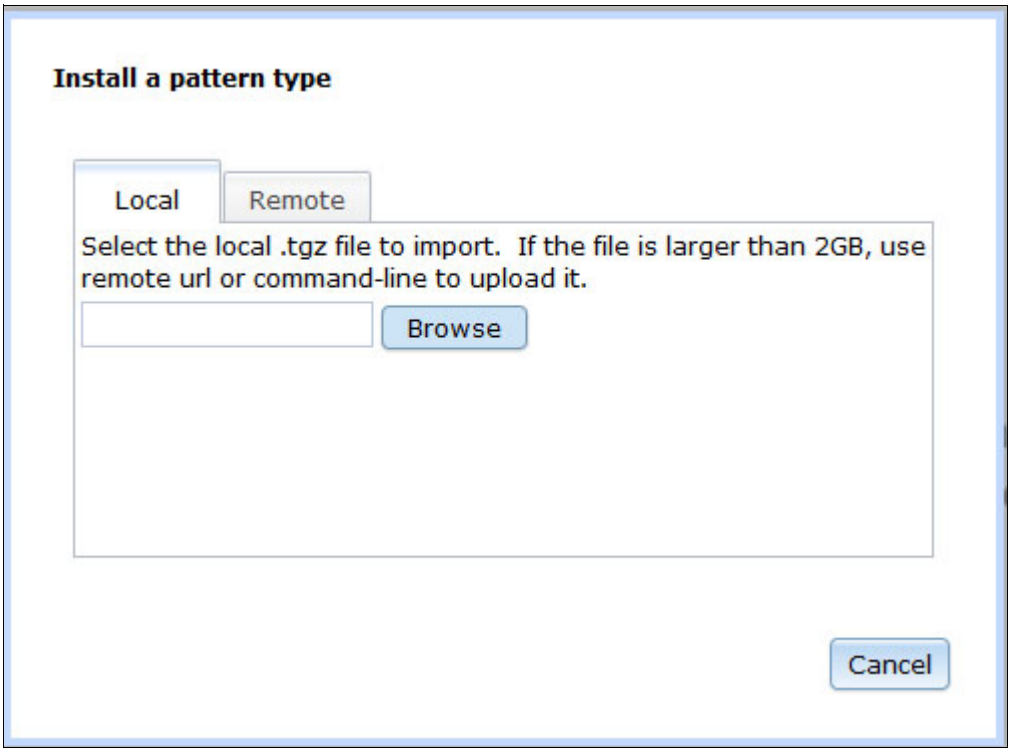


Figure 6-6 Updating virtual application patterns

The next window describes a restriction on the size of the file to import, based on a limitation of most web browsers. As alternatives, you can use:

- The remote URL (Figure 6-3 on page 96)
- The CLI (see "Update virtual application patterns using the command-line Interface (CLI)" on page 98)

4. With the import complete, click **Cloud** → **Pattern Types** to verify that the pattern type is updated. Pattern types are listed with the new version number.

Update virtual application patterns using the command-line Interface (CLI)

To update virtual application patterns using CLI, complete these steps:

1. Log in to the PureApplication system console with IP address `http://PureApplication<IP>`.
2. On the Welcome page, click the **Download Tooling** icon, and download the command line tool file to a temporary directory.
3. Extract the CLI from the `pure-cli-1.1.0.x.zip` file with the **`unzip`** command.
4. Log in to the PureApplication System appliance from the `bin` directory of the CLI, as shown in Example 6-1.

Example 6-1 Logging in to the PureApplication System appliance

```
#cd pure.cli/bin
#./pure -h Appliance_IP -u Admin_ID -p Admin_PW
```

In this example:

- *Appliance_IP* is the IP address/host name of the appliance
- *Admin_ID* is the administrator ID for the appliance
- *Admin_PW* is the administrator password for the appliance

5. Run the following command to update the type.

```
>>>deployer.patterntypes.create("/tmp/ptype-x.x.x.x.tgz")
```

6. Press Enter to submit the command.

Update shared service instances

Important: If a shared service instance is deployed in multiple clouds, you must update each shared service instance in each cloud individually.

To use the Workload console to update the shared service instances, complete these steps:

1. From the Workload console, click **Instances** → **Shared Services** (Figure 6-7).

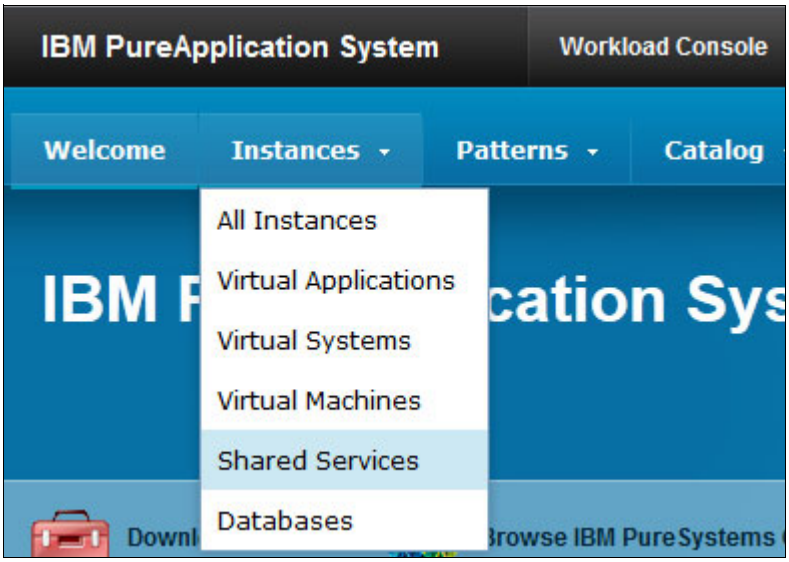


Figure 6-7 Update Shared Services

2. On the left side of the next window, select the service instance to update, and then click **Check for upgrades** in the Pattern type section. This process determines whether later versions of the plug-ins are available. A message prompts you to confirm the operation.
3. Click **Update** and then click **Upgrade.** By clicking **Upgrade**, the update begins, and the status of the instance changes to `launching and maintaining`.
4. Wait until the status of the instance changes to `running and maintaining`, indicating that the update process is complete.
5. To commit the update, click **Commit** in the upper right of the window. The update process is committed, and the instance of the shared service cannot roll back to the previous version. The status of the service instance changes to `running`.

Note: To roll the shared service instance back to the previous version, click **Revert**.

Update deployed pattern types

Important: To perform this task, the user needs to be in the Workload resources administration role, with permission to Manage workload resources, and have full permissions.

You can update the deployed pattern types using either the Workload console or CLI. This section describes the procedure for the Workload console:

1. Access the Workload console,
2. Select **Instances** → **Virtual Applications** (Figure 6-8).

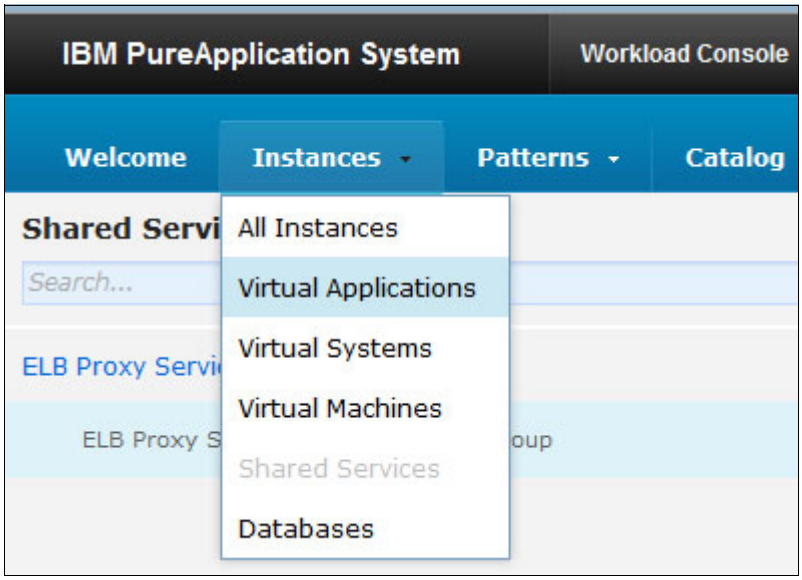


Figure 6-8 Update deployed patterns

3. From the list that is displayed, select the virtual application to update.
4. In the Pattern type section, click **Check for upgrades**. This example uses the virtual application named S*A-W401-R01 Mobile,* which is part of the example solution.

   If an update is available, a list displays of the plug-ins and virtual machines that the update will affect. Also specified is the type of update available:

   – Rolling: For which no downtime is expected.
   – Batch: For which downtime is expected.

The result of the check for the example virtual application is shown in Figure 6-9.

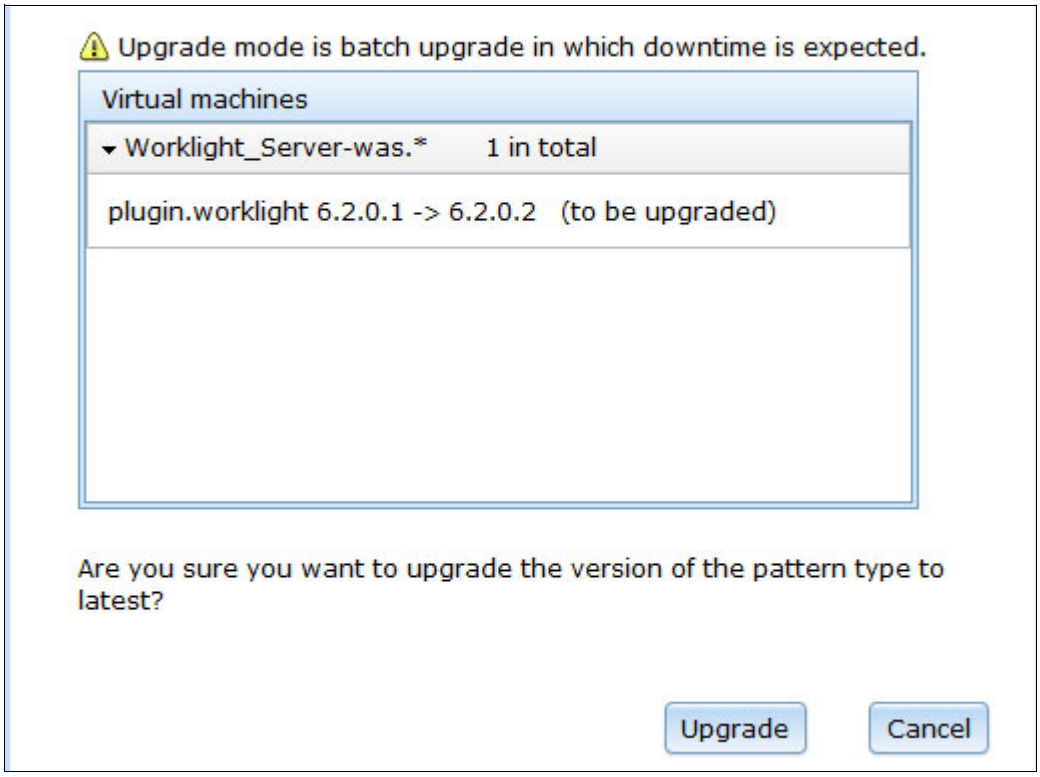


Figure 6-9 Updating deployed patterns

If you do decide to update, the pattern type backs up the data, based on the configuration in the pattern type plug-ins, and then the update changes are applied. The update can take some time.

When the update is successful, the Status field arrow turns green. Review the updated deployment and click **Commit** to update or **Revert** to discard the changes that were made in the update.

6.2 Emergency fixes

Emergency fixes are used to update the product binary files of deployed instances. They can be individual fixes or a group of fixes that are associated with particular virtual system or virtual application types.

Emergency fixes are delivered by IBM or by a provider and, as with the other updates, they must be downloaded before adding them to the catalog.

Interim fixes and fix packs are applied to virtual system instances as emergency fixes.

It is also possible to create and package your own fixes to apply custom maintenance to virtual system instances.

Important: To perform this task, the user needs to be in the Create new catalog content role or the Workload resources administration role, with full permissions.

After any Emergency fixes are downloaded from the PureSystem Center, you can add them to the catalog, and then apply them to the deployed instances.

6.2.1 Add emergency fixes to the catalog

From the Workload console, follow these steps to add emergency fixes to the catalog:

1. Click **Catalog**, then select **Emergency fixes** (Figure 6-10).

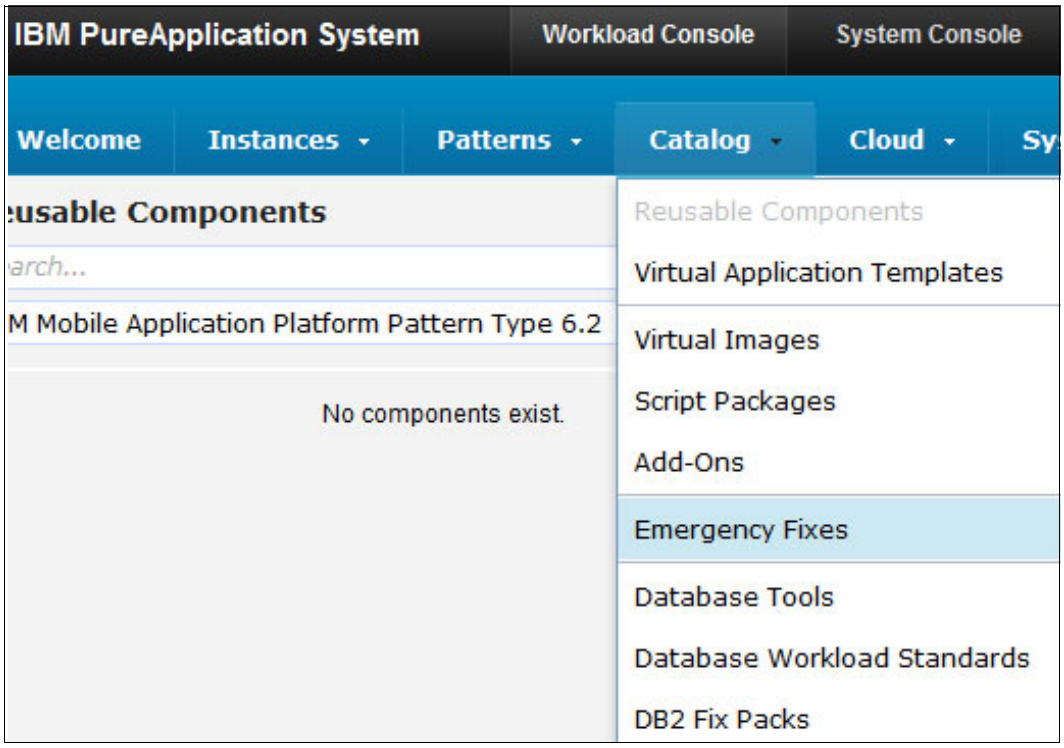


Figure 6-10 Adding emergency fixes

2. Click the **Green Plus Sign** on the toolbar to load the Emergency fix into the catalog (Figure 6-11).

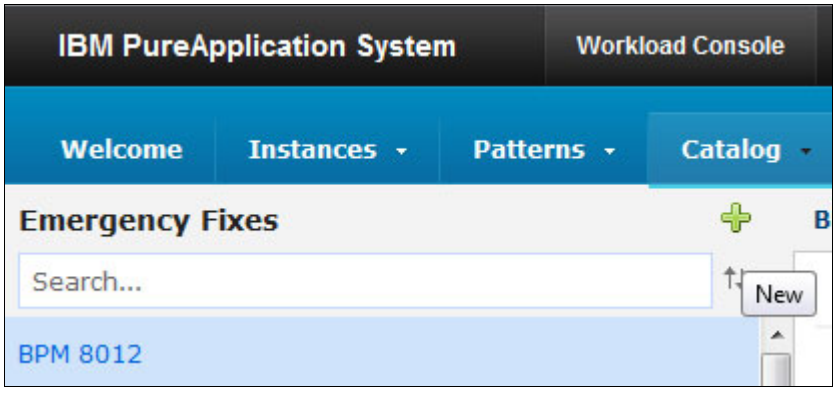


Figure 6-11 Adding Emergency fixes

3. In the next window, on the right side of the canvas, describe the fix to upload by assigning a unique name to it, and an optional description. Use a meaningful name so that the type of fix and the software that it applies to are easily identifiable.
4. Click **OK**.
5. Click **Browse**, select the file to upload, and click **OK**.

   For security reasons, only `.zip`, `.tgz`, and `.pak` files can be uploaded.
6. Click **Upload** to import the file.
7. In the **Severity** field, identify and describe the reason for the urgency for applying this emergency fix (Figure 6-12).
8. In the **Applicable to** field, select virtual images or plug-ins for which this emergency fix is applicable (Figure 6-12).

   An emergency fix is only available for instances that were created using the virtual images listed in the **Image** field. Similarly, a fix is only available for the plug-ins that are listed in the **Plug-ins** field.

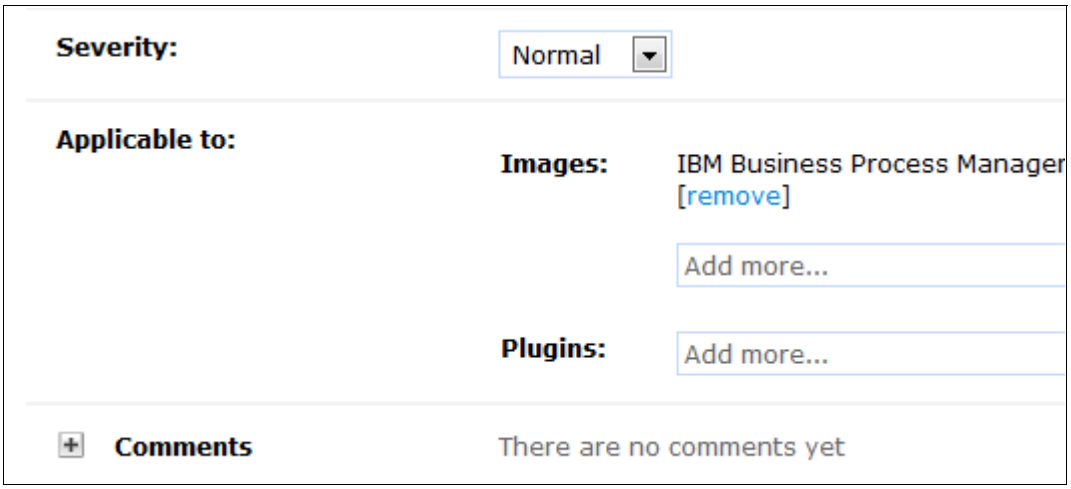


Figure 6-12 Adding emergency fixes

Note: Figure 6-12 shows an example IBM Business Process Manager Advanced 8.0.1.0 running on RHEL 6 x64 (VMWare) RedHat Enterprise Linux 64-Bit (RedHat Enterprise Linux 6) in the **Applicable to** field. This means that every new instance of a Business Process Manager virtual application pattern that is created on the system will contain the indicated fix. You must instead explicitly apply the fix to those instances already created and deployed. This is the subject described in **6.2.2, "Apply emergency fixes to deployed virtual system instances"** and **6.2.3, "Apply emergency fixes to deployed virtual application instances" on page 105**.

6.2.2 Apply emergency fixes to deployed virtual system instances

Interim fixes and fix packs are applied to virtual system instances as emergency fixes. This occurs when a virtual system instance requires maintenance to the product code or to the operating system.

Important: To perform this task, you must be granted all access to the virtual system instance or have the Workload resources administration role with full permissions.

To apply emergency fixes to deployed virtual system instances, complete these steps in the Workload console:

1. Click the **Instances** tab, and then click **Virtual Systems** (Figure 6-13).

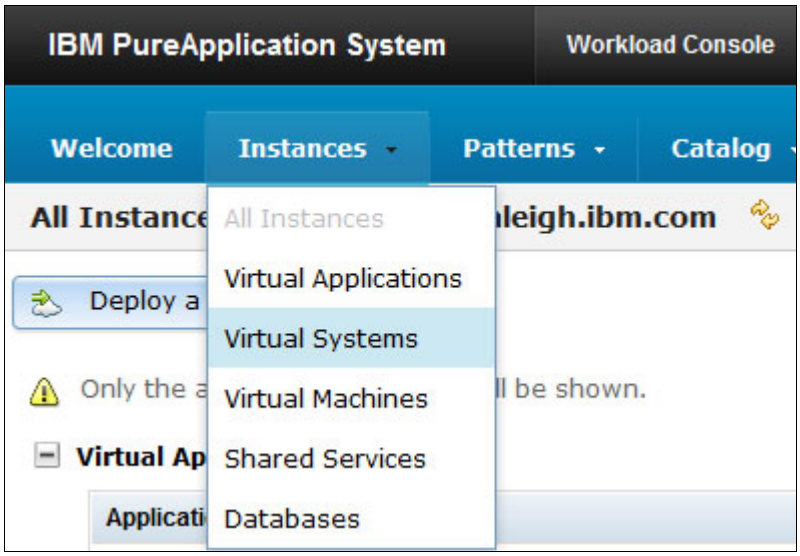


Figure 6-13 Applying emergency fixes to virtual systems

2. Select the virtual system instance to apply the service to.

   The example uses the instance named *SA-W401-R01 BPM Test Instance*.

3. Click the **Service** icon (upper right corner of the canvas). Describe your service request and schedule the service for this virtual system instance (Figure 6-14).

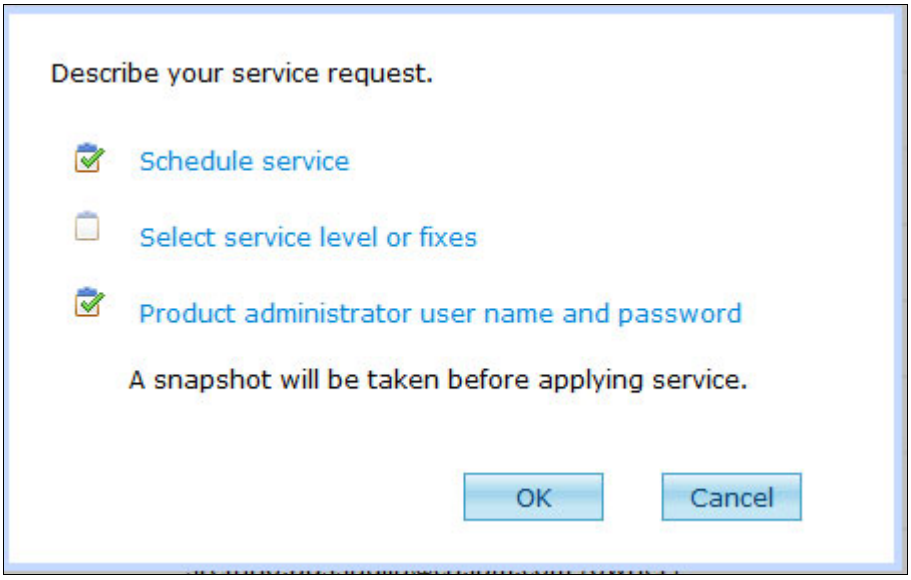


Figure 6-14 Applying emergency fixes to virtual systems

For the option to **Schedule service**, you can select either:

- **Service now**: Schedule the service request to start immediately
- **Schedule service**: Specify when the service is to be applied

For the option to **Select service level or fixes**, one of the following available services must be selected:

- **Move to service level**: Any available fix packs that can be used to update the virtual system instance
- **Apply emergency fixes**: Any available emergency fixes that can be used with the virtual system instance

For the **Product administrator user name and password** option, enter an authorized user name and password

4. Click **OK** to finalize the service request.

6.2.3 Apply emergency fixes to deployed virtual application instances

After a virtual application is deployed, to apply fixes from the catalog, you need to update the instance of the virtual application that was previously created.

Use this procedure to upgrade the base operating system virtual image or upgrade a lifecycle script in the virtual application instance.

Important: To perform this task, you must own one of these authorization roles:

- Workload resources administration role, with permission to Manage workload resources (full permission)
- Workload resources administration role, with permission to View all workload resources (read-only)
- Create new patterns
- Create new catalog content
- Create new environment profiles
- IBM License Metric Tool user

You can apply two types of emergency fixes:

- An update to the base operating system virtual image, by applying the fix to the virtual image. However, remember that, by applying an emergency fix, you cannot update the base operating system virtual image to a newer version (for example, from version 5.1.0.2 to version 5.1.0.3).
- An update to a role used by a virtual application instance.

To apply emergency fixes to deployed virtual application instances, complete these steps in the Workload console:

1. Click **Instances**, then select **Virtual Applications** (Figure 6-15).

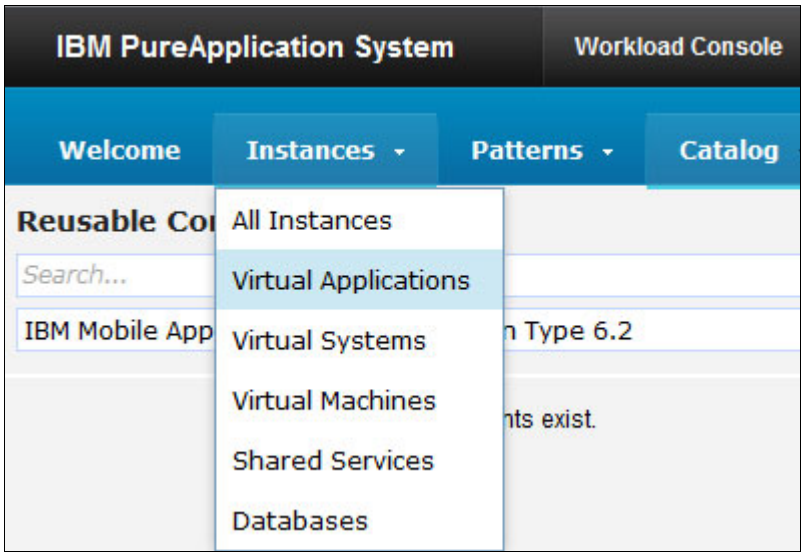


Figure 6-15 Applying emergency fixes to Virtual Applications

2. Select the virtual application instance to update.
3. Click the **Manage** icon on the toolbar to go to the Virtual Application console. The figures use an instance that was created for the example solution: Virtual application instance *SA-W401-R01 Mobile.*
4. In the Virtual Application console, click the Operations tab (Figure 6-16).

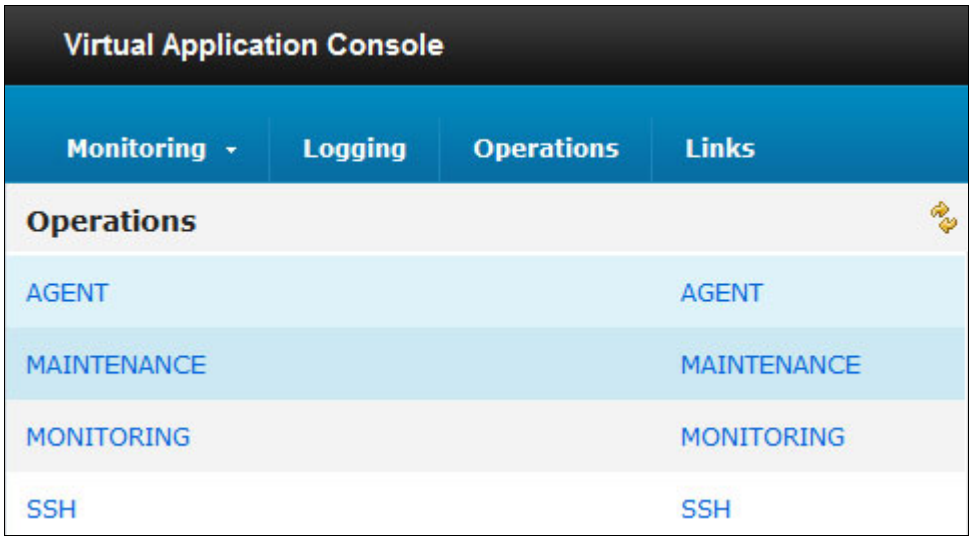


Figure 6-16 Applying emergency fixes to virtual applications

5. Click **MAINTENANCE**, and the window in Figure 6-17 is displayed.

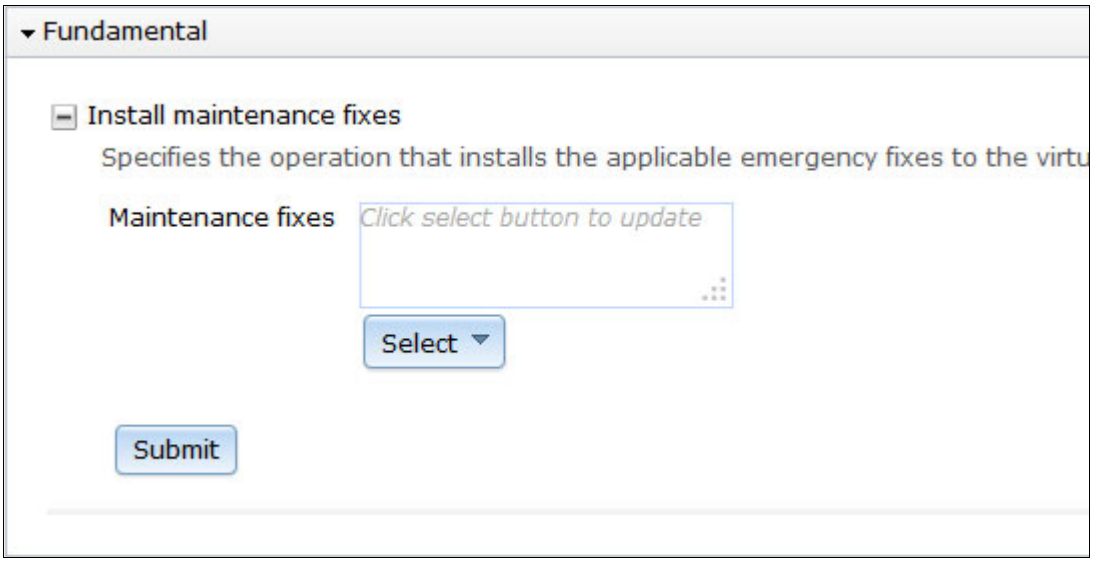


Figure 6-17 Applying emergency fixes to virtual applications

> **Note:** You can verify that the virtual application instance name that you are working on, and its status, by looking at the toolbar on the Virtual Application console.

6. Click **Select** and choose the fixes to apply.
7. Click **Submit** to complete the task. The status of the installation is displayed in the Operation Execution Result section.

6.3 Backup and restore

Using the backup process, you can save and transfer your system configuration to a specified backup storage server. There a two different types of backups:

- Initial backup: Mandatory backup that provides the baseline for future backups. All subsequent backups are delta backups.
- Delta backup: A copy of the system configuration that includes all of the changes since the last baseline backup.

The system automatically detects the type of backup to run by checking the backup storage server for the presence of a baseline backup. A baseline backup is necessary the first time, but you can create a new baseline backup at any time by configuring the location in the backup and restore system settings.

Data and artifacts can be restored at any time if, for example, artifacts are corrupted or if there has been an accidental deletion.

6.3.1 Backup

You can perform two types of backups:

- *Manual:* The **Backup now** button starts a manual backup of your configuration and data to the remote location that you defined. Other jobs in the system are placed on hold during the backup.
- *Scheduled:* The option to **Enable scheduled backups** is used to configure a schedule of automatic backups. Select a backup start time and one or more days of the week to run the backup operation. Once the backup is scheduled the system will automatically run the backup of the local data on the selected days of the week. Other jobs in the system are placed on hold during the backup.

To set up a backup on the PureApplication System, follow these steps:

1. Access the System console.
2. Click **System**, then select **Settings** (Figure 6-18).

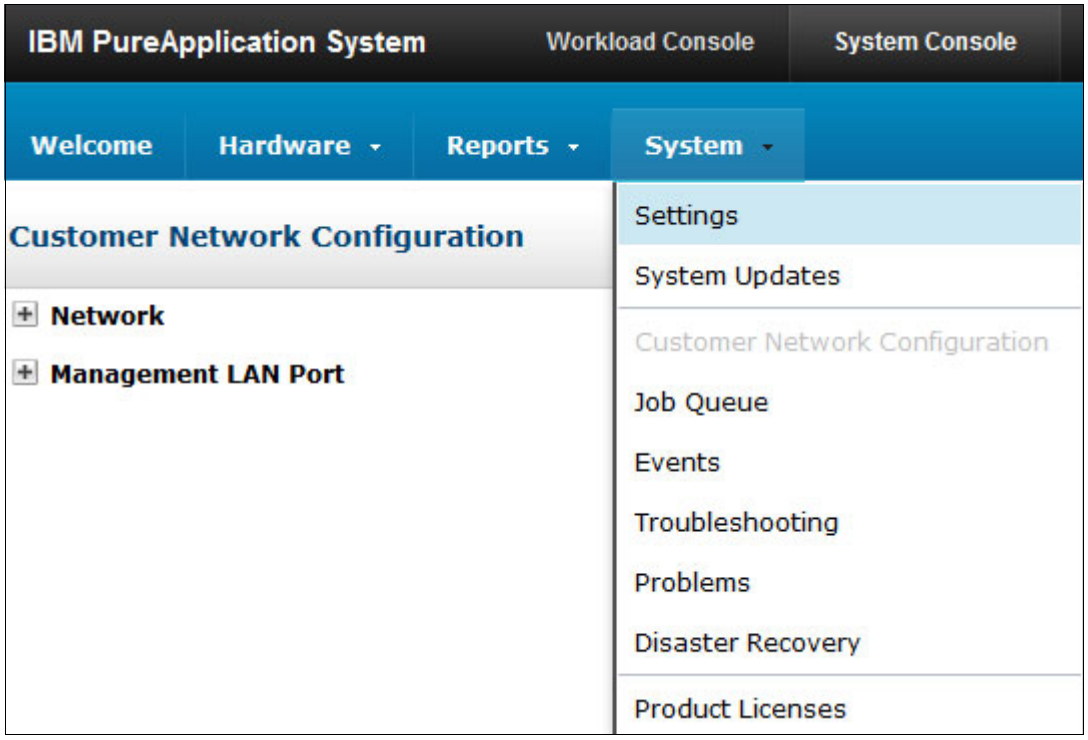


Figure 6-18 Backup the system.

3. In the next window, expand the **System Backup and Restore** section of the window (Figure 6-19).

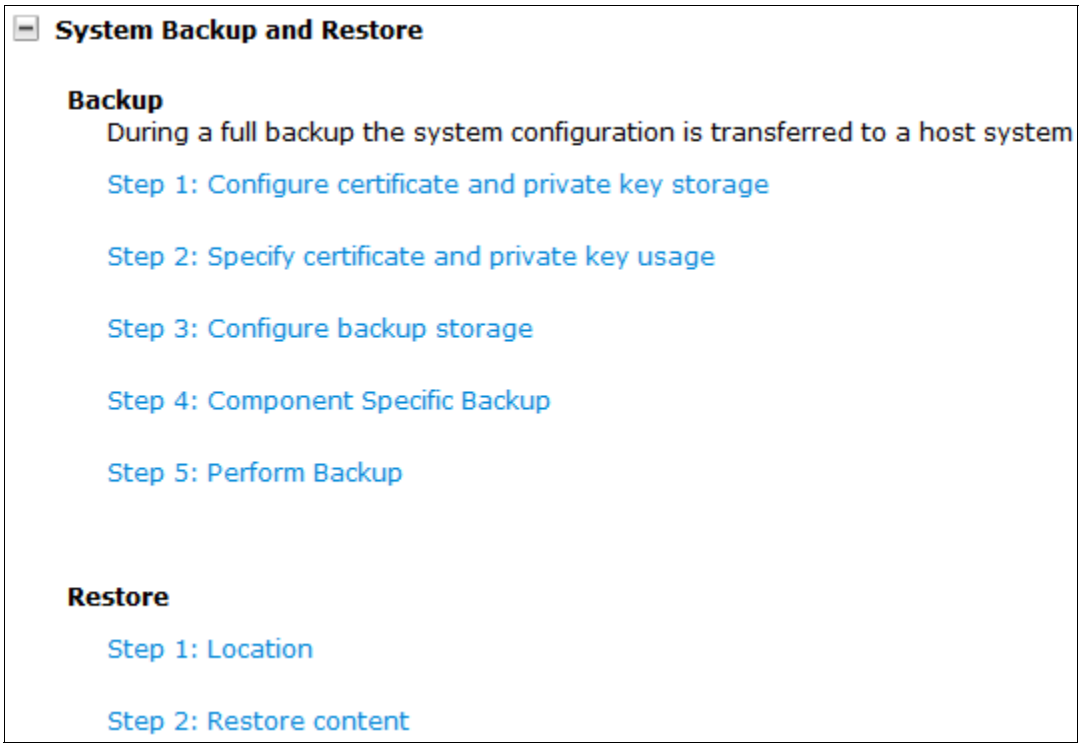


Figure 6-19 Backup of the system

As shown in Figure 6-19, complete these steps to define the backup:

- **Step 1: Configure certificate and private key storage:** Specify the secure location to which the certificate and private key are saved. This is used later to decrypt the backup during the restore process.
- **Step 2: Specify certificate and private key usage:** Provide or generate the certificate and private key to be used to protect your backup.
- **Step 3: Configure backup storage:** Specify where backup artifacts are stored. The location and credentials should be different from those used to store the private key pair.
- **Step 4: Component Specific Backup:** *(optional step)* Select the types of components (such as images, script packages, add-ons, or patterns) that are in your PureApplication System environment and that you want to back up.

– **Step 5: Perform Backup:** Click this step to back up your system (Figure 6-20).

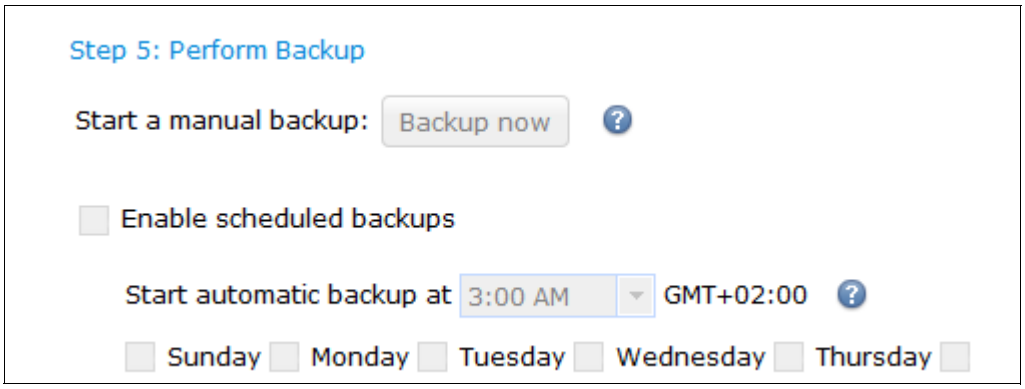


Figure 6-20 Back up the system

Restore

You can perform limited restore tasks to selectively restore the system and the workload data.

An IBM Service Technician is required to perform a system level restore from the backup archives. The system level restore procedure includes extracting the data from the backup storage server, decrypting the backup information, and verifying the integrity of the data by using the metadata that is present in the backup data.

For you to work with the restore functions, follow these steps:

1. Access the System console.
2. Click **System**, then select **Settings**.
3. Expand the **System Backup and Restore** section.
4. *If you included virtual images in your backup*, follow this step. Otherwise, go to Step 5 on page 111:
   a. Click **Restore now** under **System Backup and Restore**.

Note: A restore can be done from the current backup location, or from another remote backup location, by specifying the location of the certificate and private key, and the location of the backup to restore.

b. Click **Save and test connection** (Figure 6-21) to save the new backup location information and check that the connection is successful.

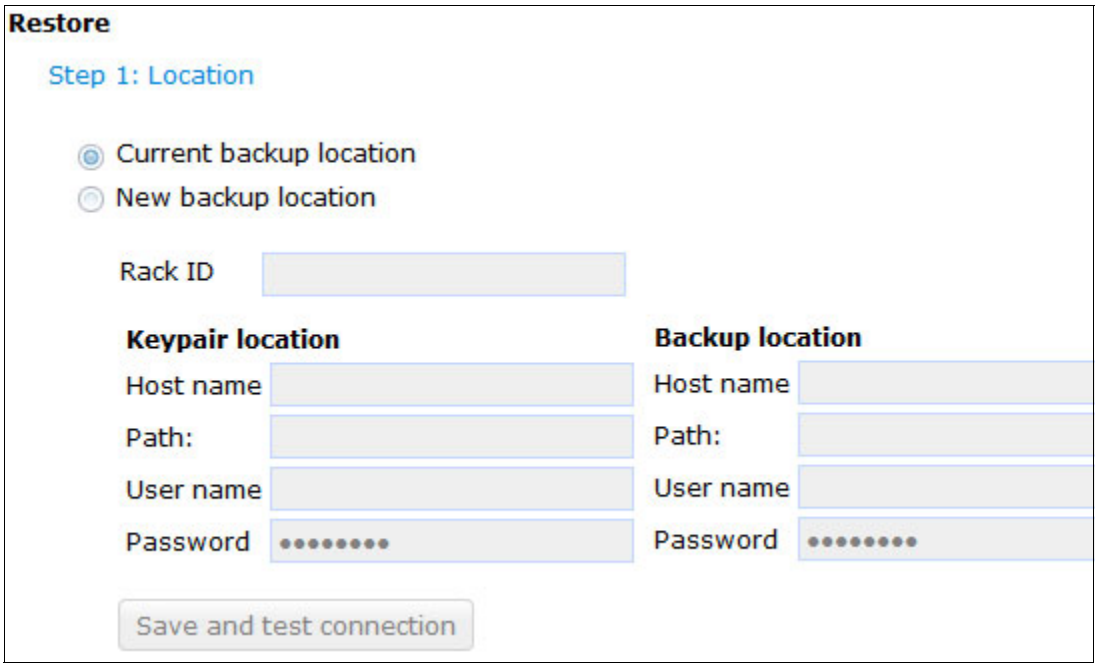


Figure 6-21 Restore data

5. *If you did not include virtual images in your backup*, populate the fields in the **Restore** section.

> **Note:** You can select a specific component type (such as virtual images, workload configurations, add-ons, script packages, or patterns) and one or more artifacts from the available list to restore. A table is displayed from which you can select a component type, and view the suitable components that have been previously backed up.

6. Click **Restore** to begin the process (Figure 6-22).

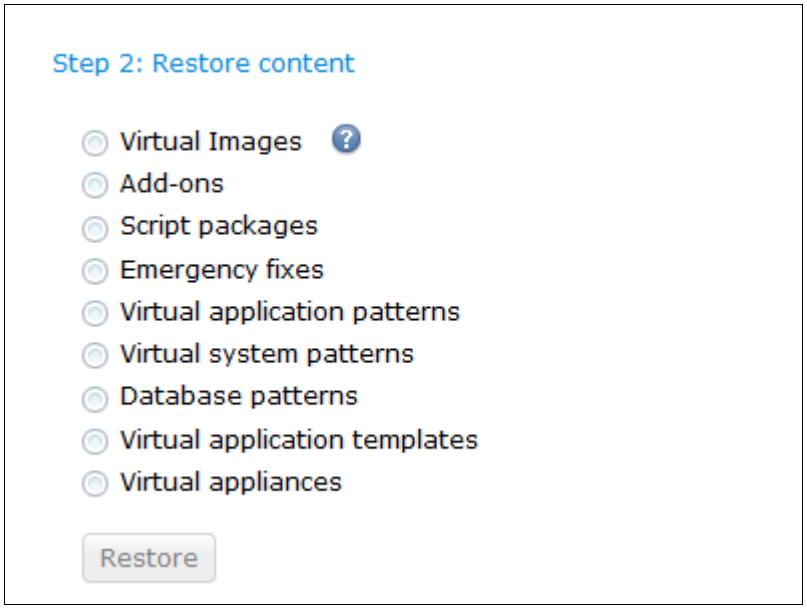


Figure 6-22 Restore components and artifacts

A

Script packages

The intention of using script packages is to further enable you to customize your environment beyond the customization provisions that are standard in IBM PureApplication System. This Appendix provides a list of script packages used in the example solution to deploy parts into the virtual system pattern. A script package is designed to customize the behavior of parts within a pattern.

The script package is an archive file (`.zip`, `.tgz`, `.tar.gz`) that contains shell scripts or Jython scripts and a JavaScript Object Notation (JSON) file. The JSON file describes the script package, and points to the location of the main script, among other items. The JSON file must be in the root directory of the script package.

The goal is to show step-by-step how to create script packages. Therefore, this appendix focuses specifically on script packing tasks and is not intended to teach shell scripting or Jython programming. The reader is expected to have a basic understanding of virtual patterns and how they are built from parts and scripting.

After the script package is linked to the part, the PureApplication System unpacks the file during the execution of the deployment in the folder that you define in the JSON file. The script then develops all of the logic using that folder definition. You can also create subfolders and attach extra files, such as `.jar` or `.xml`, or an entire installation of software, such as Tomcat.

This appendix includes the following sections:

- Install the UrbanCode agent
- Deploy the war file and server.xml on Liberty
- Deploy the war file and server.xml on Liberty
- Configure and deploy IBM Integration Bus
- Create the database and tables
- Install the DB2 client

Install the UrbanCode agent

For the example solution, a script package was created to install the UrbanCode agent using the following steps:

1. Create the folder `install-ucd-agent-ipas`.
2. Create the shell script `install-ucd-agent-ipas.sh` (Example A-1).

Example A-1 Example of install-ucd-agent-ipas.sh

```
#!/bin/sh
#*********************************************************
# Source File Name = install-ucd-agent-ipas.sh
# Function = Install the UrbanCode Agent
# Author = Fabio Santos B. da
#*********************************************************

echo "[IBM UCD]: Unzip ibm-ucd-agent*.zip "
unzip -q ibm-ucd-agent-6.0.0.428559.zip

echo "[IBM UCD]: Configuring installation properties "

export UREMOTE_HOST=$1
export UREMOTE_PORT=$2

echo "[IBM UCD]: create install.properties file "

echo "version=6.0.0.424783" >> /tmp/install.properties
echo "nonInteractive=true" >> /tmp/install.properties
echo "agent.mutual_auth=false" >> /tmp/install.properties
echo "java.home=/opt/IBM/ibm-java-i386-60" >> /tmp/install.properties
echo "agent.home=/opt/ibm-ucd/agent" >> /tmp/install.properties
echo "agent.name=`hostname -s`" >> /tmp/install.properties
echo "agent.jms.remote.host=$UREMOTE_HOST" >> /tmp/install.properties
echo "agent.jms.remote.port=$UREMOTE_PORT" >> /tmp/install.properties

echo "[IBM UCD]: Install the UrbanCode Agent "
/tmp/ibm-ucd-agent-install/install-agent-from-file.sh /tmp/install.properties

echo "[IBM UCD]: Start UbanCode Agent"
/opt/ibm-ucd/agent/bin/ibm-ucdagent start

echo "[IBM UCD]: Installation Completed Successfully"
```

3. Create the JSON file, `cbscript.json` (Example A-2).

Example A-2 Example JSON file

```
[
    {
        "name": "UrbanCode Agent Installation",
        "version": "1.0",
        "description": "Install the UrbanCode Agent using silent mode
customization",
```

```
        "command": "/bin/sh /tmp/install-ucd-agent.sh",
        "log": "/tmp",
        "location": "/tmp",
        "timeout": "0",
        "commandargs": "$REMOTE_HOST $REMOTE_PORT",
        "keys":
    [
        {
            "scriptkey": "REMOTE_HOST",
         "scriptvalue": "",
            "scriptdefaultvalue": "localhost"
        },
        {
            "scriptkey": "REMOTE_PORT",
         "scriptvalue": "",
            "scriptdefaultvalue": "7918"
        }
    ]
    }
]
```

4. Package the script files by using the steps in “Package the script files” on page 53.

Deploy the war file and server.xml on Liberty

The purpose of the script package is to install the application during the deployment of the environment. This example involves the files `ServiceDeskWebApp.war` and `server.xml` that need to be installed in Liberty. To deploy them, complete these steps:

1. Create the folder `deploy-redbook-app` and copy `ServiceDeskWebApp.war` and `server.xml` to this folder.
2. Create the file `deploy-redbook-app.sh` (Example A-3).

Example A-3 Example deploy-redbook-app.sh file

```
#!/bin/sh
#************************************************************
# Source File Name = deploy-redbook-app.sh
# Function = Deploy the war and server.xml file to install
#            the application during the creation of the
#            environment
# Author = Fabio Santos B. da
#************************************************************

export LIBERTY_SERVER_PATH=$1

cp server.xml $LIBERTY_SERVER_PATH
cp ServiceDeskWebApp.war $LIBERTY_SERVER_PATH/dropins
```

3. Create the file `cbscript.json` (Example A-4).

Example A-4 JSON script for deploying the war file

```
[
    {
        "name": "Redbook Application Deployment",
        "version": "1.0",
        "description": "Deploy the RedBook Application",
        "command": "/bin/sh /tmp/deploy-redbook-app.sh",
        "log": "/tmp",
        "location": "/tmp",
        "timeout": "0",
        "commandargs": "$LIBERTY_SERVER_PATH",
        "keys":
    [
        {
            "scriptkey": "LIBERTY_SERVER_PATH",
         "scriptvalue": "",
            "scriptdefaultvalue": "/opt/ibm/liberty/wlp/usr/servers/server1"
        }
    ]
    }
]
```

4. Copy the files `ServiceDeskWebApp.war` and `server.xml` to the `/tmp` folder defined in `cbscript.json`.
5. Create the script package `deploy-app-redbook.tgz` or `.zip` following the steps in “Package the script files” on page 53.

Configure and deploy IBM Integration Bus

The example script package included two additional files in the script package: `odbc.ini` and `ServiceDeskESB.bar`.

The shell script is expected to accomplish these tasks:

- Copy the `odbc.ini` file to the folder `/opt/ibm/mqsi/9.0.0.0/`.
- Create WebSphere MQ queues and channel.
- Run the command to deploy the `ServiceDeskESB.bar` file:

  a. Create the folder `configure-IIB` and copy the `ServiceDeskESB.bar` file and `odbc.ini` file to the folder.

  b. Create the file `configure-IIB.sh` (Example A-5).

Example A-5 Example configure-IB.sh file

```
#!/bin/sh
#*******************************************************
# Source File Name = configure-IIB
# Function = Configure IIB
# Author = Fabio Santos B. da
#*******************************************************
IIB_USER=$1
IIB_PWD=$2
```

```
cp odbc.ini /opt/ibm/mqsi/9.0.0.0/odbc.ini

# put this in the profile
export LD_LIBRARY_PATH=/opt/mqm/java/lib:$LD_LIBRARY_PATH
export ODBCINI=/opt/ibm/mqsi/9.0.0.0/odbc.ini
# create the WebSphere MQ queue
echo "DEFINE QLOCAL('WEB_IN') PUT(ENABLED) GET(ENABLED)" | runmqsc IB9QMGR
echo "DEFINE QLOCAL('TRIGGER_PROCESS_Q') PUT(ENABLED) GET(ENABLED)" | runmqsc
IB9QMGR

# create the channel and additional access to channel
echo "DEFINE CHANNEL(WAS.CONN) CHLTYPE(SVRCONN) TRPTYPE(TCP) MCAUSER('mqm')" |
runmqsc IB9QMGR
echo "SET CHLAUTH(WAS.CONN) TYPE(BLOCKUSER) USERLIST(ALLOWANY)" | runmqsc
IB9QMGR

mqsisetdbparms IB9NODE -n SG248243 -u $IIB_USER -p $IIB_PWD
mqsicvp IB9NODE -n SG248243

# deploy the bar
mqsideploy -i localhost -p 1414 -q IB9QMGR -e default -a ServiceDeskESB.bar -m
-w 60
```

6. Create the JSON file `cbscript.json` (Example A-6).

Example A-6 JSON script for configure IIB

```
[
    {
        "name": "Configure IIB",
        "version": "1.0",
        "description": "Configure IIB",
        "command": "/bin/sh /tmp/configure-IIB.sh",
        "log": "/tmp",
        "location": "/tmp",
        "timeout": "0",
        "commandargs": "$IIB_USER $IIB_PWD",
        "keys":
    [
        {
            "scriptkey": "IIB_USER",
         "scriptvalue": "",
            "scriptdefaultvalue": "${AdvancedPart.ConfigPWD_USER.username}"
        },
        {
            "scriptkey": "IIB_PWD",
         "scriptvalue": "",
            "scriptdefaultvalue": "${AdvancedPart.ConfigPWD_USER.password}"
        }
    ]
    }
]
```

7. Create the script package `configure-IIB.tgz` or `.zip` by following the steps in “Package the script files” on page 53.

Create the database and tables

For this script package, create the DB2 database and tables for the application. Use a Data Definition Language (DDL) file and issue several db2 commands in the shell script. Complete these steps:

1. Create the folder `create-app-database`.
2. Create the file `sql.ddl` (Example A-7).

Example A-7 Example sql.dll file

```
CREATE TABLE SAW_STATUS (
      "STATUS_ID" BIGINT NOT NULL,
      "STATUS_NAME" CHAR(20) NOT NULL
   )

   DATA CAPTURE NONE;

ALTER TABLE SAW_STATUS ADD CONSTRAINT "CC1408484744159" PRIMARY KEY
("STATUS_ID");

COMMENT ON TABLE SAW_STATUS IS 'SG24-8243 Redbook Status table';

insert into SAW_STATUS values (0, 'PENDING');
insert into SAW_STATUS values (1, 'APPROVED');
insert into SAW_STATUS values (2, 'REJECTED');

CREATE TABLE SAW_REQUESTS (
      "REQUEST_ID" BIGINT NOT NULL GENERATED ALWAYS AS IDENTITY ( START WITH 1
INCREMENT BY 1 MINVALUE 1 MAXVALUE 9223372036854775807 NO CYCLE NO CACHE),
      "SERIAL_NUMBER" CHAR(20) NOT NULL,
      "NAME" CHAR(100) NOT NULL,
      "TRACKING_NUMBER" CHAR(20) NOT NULL,
      "ADDRESS" CHAR(150) NOT NULL,
      "STATUS_ID" BIGINT NOT NULL,
      "CREATION_TIME" TIMESTAMP,
      "REMARKS" CHAR(200),
      "COMMENTS" CHAR(200)
   )

   DATA CAPTURE NONE COMPRESS YES;

ALTER TABLE SAW_REQUESTS ADD CONSTRAINT "CC1408485016720" PRIMARY KEY
("REQUEST_ID");

ALTER TABLE SAW_REQUESTS ADD CONSTRAINT "CC1408485022210" FOREIGN KEY
("STATUS_ID") REFERENCES SAW_STATUS("STATUS_ID");

COMMENT ON TABLE SAW_REQUESTS IS 'SG24-8243 Redbook Requests table'
```

3. Create the shell script `create-app-database.sh` (Example A-8).

Example A-8 Example create-app-database.sh file

```
#!/bin/sh
#*********************************************************
# Source File Name = create-app-database.sh
# Function = Create the Database from Application
# Author = Fabio Santos B. da Silva
#*********************************************************
DB_USER=$1
DB_PWD=$2

db2 create database SG248243 using codeset UTF-8 territory en
db2 connect to SG248243 user $DB_USER using $DB_PWD
db2 -tvf sql.sql
```

4. Create the JSON file `create-app-database.json` (Example A-9).

Example A-9 Example create-app-database.json file

```
[
    {
        "name": "Create the Redbook database and tables",
        "version": "1.0",
        "description": "Create the Redbook database and tables",
        "command": "/bin/sh /tmp/create-app-database.sh",
        "log": "/tmp",
        "location": "/tmp",
        "timeout": "0",
        "commandargs": "$DB_USER $DB_PWD",
        "keys":
    [
        {
            "scriptkey": "DB_USER",
         "scriptvalue": "",
            "scriptdefaultvalue": "${AdvancedPart.ConfigPWD_USER.username}"
        },
        {
            "scriptkey": "DB_PWD",
         "scriptvalue": "",
            "scriptdefaultvalue": "${AdvancedPart.ConfigPWD_USER.password}"
        }
    ]
    }
]
```

5. Create the script package `create-app-database.tgz` or `.zip` following the steps in “Package the script files” on page 53.

Install the DB2 client

This script package installs the DB2 client. To prepare the script, complete these steps:

1. Create the folder `install-db2-client`.
2. Download the IBM Data Server Runtime Client package from the following website and copy it to the folder created in Step 1:

   http://www-01.ibm.com/support/docview.wss?uid=swg21385217
3. Create the shell script `install-db2-client.sh` (Example A-10).

Example A-10 Example install-db2-client.sh file

```
#!/bin/sh
#*********************************************************
# Source File Name = install-db2-client.sh
# Function = Install DB2 Client
# Author = Fabio Santos B. da Silva
#*********************************************************

tar -xvzf /tmp/ibm_data_server_driver_package_linuxx64_v10.5.tar.gz /tmp

/tmp/dsdriver/installDSDriver
```

4. Create the `cbscript.json` file (Example A-11).

Example A-11 Example cbscript.json file

```
[
    {
        "name": "Install DB2 Client",
        "version": "1.0",
        "description": "Install DB2 Client",
        "command": "/bin/sh /tmp/install-db2-client.sh",
        "log": "/tmp",
        "location": "/tmp",
        "timeout": "0",
        "commandargs": "",
        "keys":

]
```

5. Create the script package `install-db2-client.tgz` or `.zip` following the steps in “Package the script files” on page 53.

B

IBM PureApplication System V2.0

This appendix provides a brief description of some of the changes and improvements introduced by IBM PureApplication System v2.0 as compared to the version v1.1. Note that the terminology for versions in this chapter also uses the term classic virtual system pattern (VSP) to refer to previous versions 1.x, and new VSP for Version 2.0.

This appendix covers only the differences relevant to the example solution specifically developed for this paper.

For an overview of IBM PureApplication System v2.0, and a complete understanding of all the new features it offers, see the IBM Knowledge Center at the following link:

http://pic.dhe.ibm.com/infocenter/psappsys/v1r1m0/index.jsp?topic=/com.ibm.puresystems.appsys.1500.doc/systemconsole/c_syswelcomeipassc.html

This appendix includes the following sections:

- Pattern development
- Next pattern generation
- Bring your own operating system
- Virtual application pattern building with components
- Virtual application patterns, pattern components
- Backup and Restore

Pattern development

Version 1.1 of the IBM PureApplication System provided two different ways for customers to model their solutions:

- The first is with VSPs. VSPs are topology- centric patterns that require virtual images. They allow more user control over customizing the topology.
- The second way is with Virtual Application Patterns (VAPs). VAPs are application-centric patterns that offer higher-level abstractions.

In version 2.0, VSPs and VAPs coexist. The new VSP architecture has converged with the VAP architecture, resulting in a single pattern engine. Additionally, version 2.0 keeps the term classic VSP that refers to virtual system patterns based on the previous architecture.

The following terms are used interchangeably in this chapter:

- Version 1.0 or classic VSP
- Version 2.0 or new VSP

Next pattern generation

In version 1.0, the underlying architecture for VSPs was much different from VAPs. This caused challenges such as image sprawl, lack of parity with VAPs for management, difficulty for content providers to reuse software, complex processes to build custom hypervisor images, and licensing issues due to the bundled OS. Now, in version 2.0, the VSP architecture has converged with the VAP architecture, resulting in a single pattern engine.

Hypervisor image split

Classic VSP patterns delivered both the software and the OS in a single hypervisor image. In version 2.0 with the converged patterns, this model is split into two components as shown in Figure B-1. Now, software is being delivered separately with software packages and a base OS hypervisor image. In essence, the model is moving from an image-based model to a software component model.

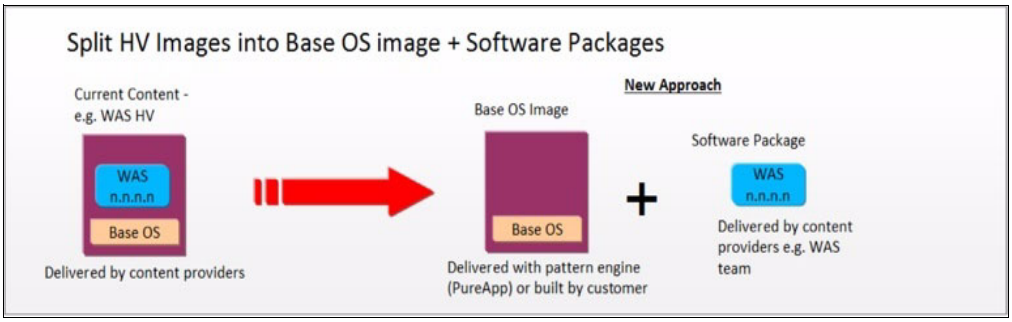


Figure B-1 Splitting of HV Image

Classic virtual system pattern

Figure B-2 shows a comparison of a classic VSP pattern and a new VSP pattern. The classic pattern deploys a two node topology (including a WebSphere Application Standalone server and DB2 Enterprise server) with script packages and a single add-on to the new architecture. The original stand-alone server part combined the base OS image with the WebSphere Application Server software. In the new VSP pattern, this is now split into images (Core OS_WAS and Core OS_DB2) and software components (Standalone server and DB2 Enterprise). The script packages and add-ons remain the same. Further enhancements include adding dependency, data links, and policies.

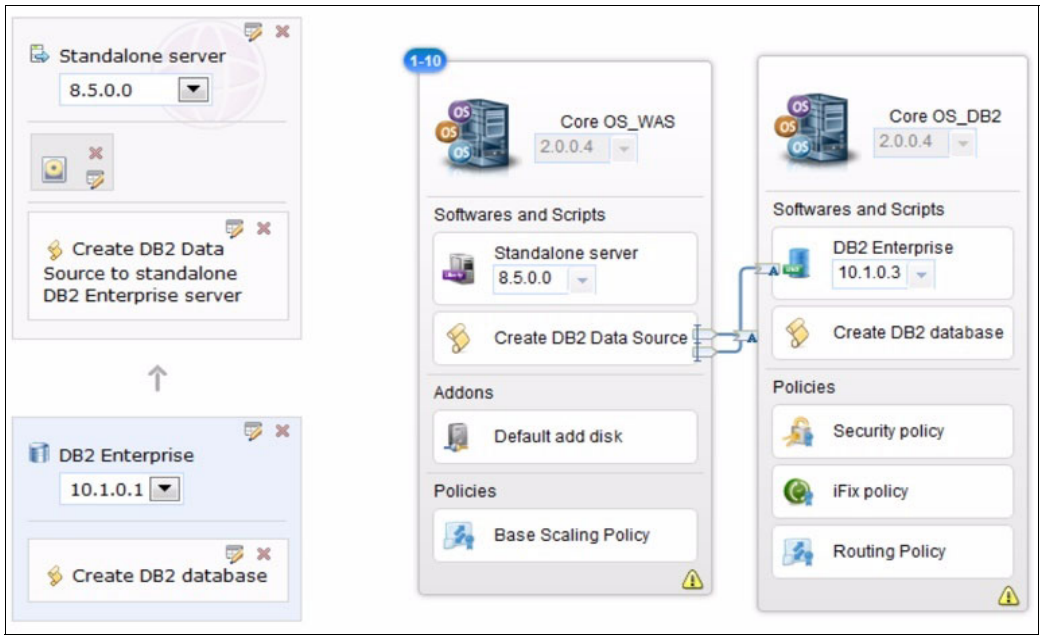


Figure B-2 Comparison of classic VSP and new VSP

Bring your own operating system

When you construct a pattern in version 2, you use the base OS component and drag software packages and script packages onto the OS component. Figure B-3 shows two OS nodes, each with their own software components and script components. You can swap your own RedHat and AIX based images for the default base OS image. Then, the pattern builder allows you to construct a pattern by layering software packages on top of your own custom OS images.

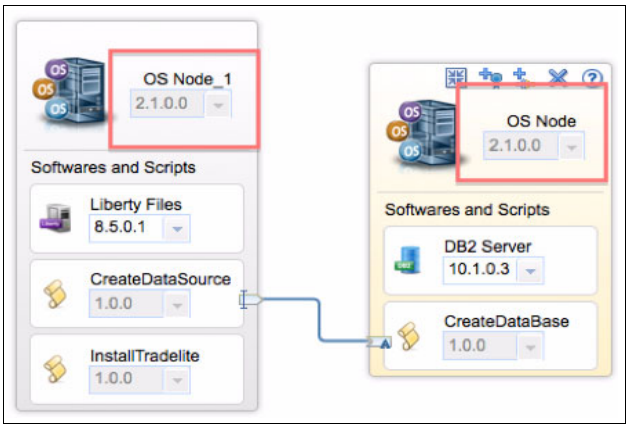


Figure B-3 Use your own operating system

Virtual application pattern building with components

As previously noted, the virtual application pattern in version 1.1 has certain opportunities for improvements in the following areas:

- Complex topologies are difficult or impossible
- No ability to explicitly select the base OS image
- Developing the component plug-ins is difficult

The new VAP pattern engine in version 2addresses some of these issues with components that provide the following advantages:

- VSPs can select a specific OS image
- Components can encapsulate VSPs
- VAP builder includes components as available assets
- Middleware implementation has a simpler path using software components and script packages now available in the VAP deployment environment

The relationship between the VAP and VSP components is shown in Figure B-4.

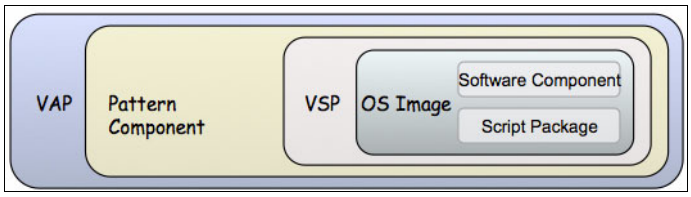


Figure B-4 Relationship between objects

Virtual application patterns, pattern components

This section discusses how pattern components can be bridged between VSPs and VAPs using the new abilities with version 2.

To create a plug-in using the Plug-in Development Kit (PDK) and then to expose the plug-in components on the application builder for use in building virtual applications takes time and effort. Pattern components streamline your efforts as they do not need the PDK to expose a component on builder.

Simply define a virtual system pattern that represents your topology, then expose that as a pattern component on the virtual application builder. You can define the category for the pattern component in the palette. You can also define inputs and outputs for the component, which makes it easy to do the wiring between components in the pattern. For example, a database can output the database credentials and the application server then can input those credentials to configure the database for the application.

Figure B-5 shows a sample of a web application component.

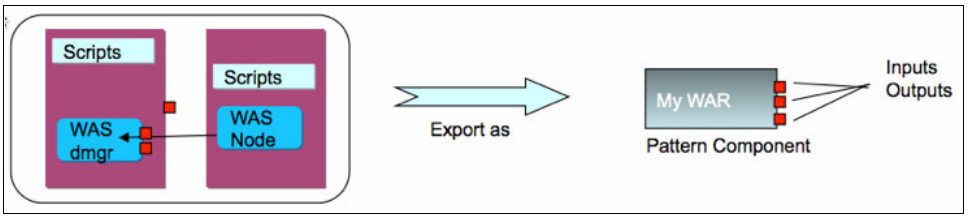


Figure B-5 Pattern component with input and output credentials

You can also define a hierarchy of components for building a pattern of patterns, as shown in Figure B-6.

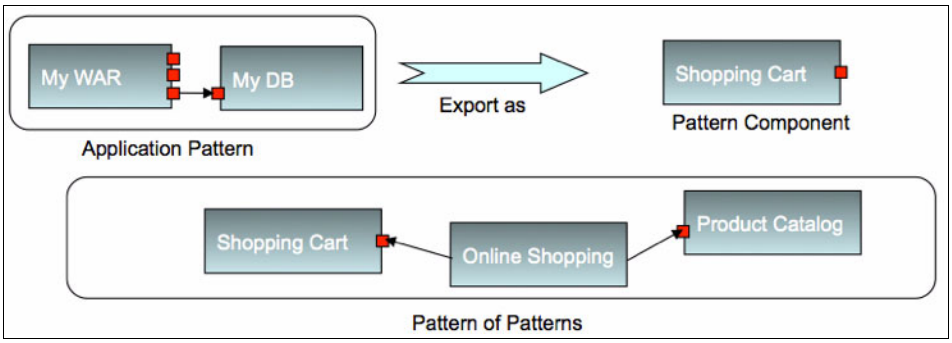


Figure B-6 Hierarchy of pattern components

Virtual application patterns deployment

When you build a virtual application that contains pattern components, at deployment time, the components are recursively unfolded. In this example, there are three pattern components in the pattern and each maps to a topology. At deployment time, each component is unfolded so the result is a single topology composition.

This abstractionsimplifies things for the application developer who can focus on pattern components without being concerned about the underlying topology for the infrastructure. Figure B-7 shows the VAP deployment hierarchy.

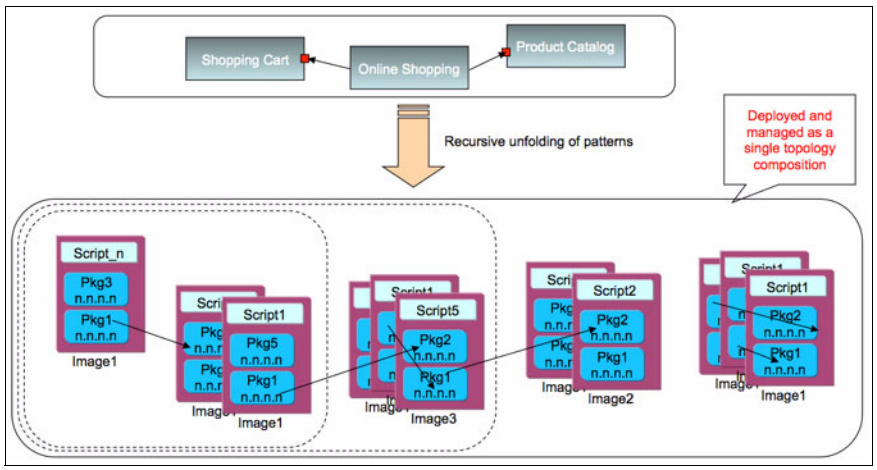


Figure B-7 Virtual application patterns, deployment hierarchy

Virtual applications and pattern components

Pattern content providers create virtual application components by using the PDK. Pattern components are a new feature for virtual applications that are now supported along with components, links, and policies. Pattern components make it easy to create a virtual application pattern using a virtual system topology pattern. Therefore, the pattern creator can build a topology without using the PDK, and the pattern deployer can select the topology to use at deployment time.

The following list notes the different roles of the pattern provider:

- To create virtual application components (content provider creates plug-ins using the PDK)
- To create pattern components: Pattern components are a new addition to virtual applications and support the following abilities (supported only in VSP of version 2)
  - Reduce the need to create components using PDK
  - Represent a virtual system or virtual application pattern as a component
  - Allow reuse of virtual system contents for virtual applications
  - Allows the pattern deployer to select the topology
- To support virtual application patterns

Virtual application components are definable by using the PureApplication System workload console. To do so, select **Catalog** → **Components Definition**, as shown in Figure B-8.

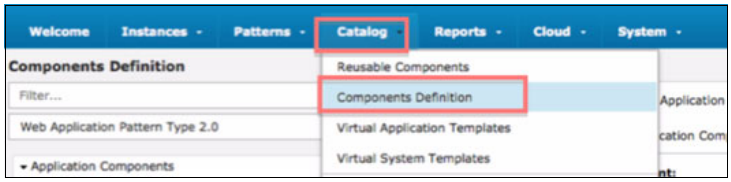


Figure B-8 Components definition using the PureApplication System workload console

To define a pattern component, a workload administrator uses the component definition menu item. Here you define the pattern component name, inputs, outputs, and targets for the component. In this example, an environment target allows the pattern deployer to select the target environment (either production or development). Each environment then is mapped to a target virtual system pattern that defines the topology for that environment, as shown in Figure B-9

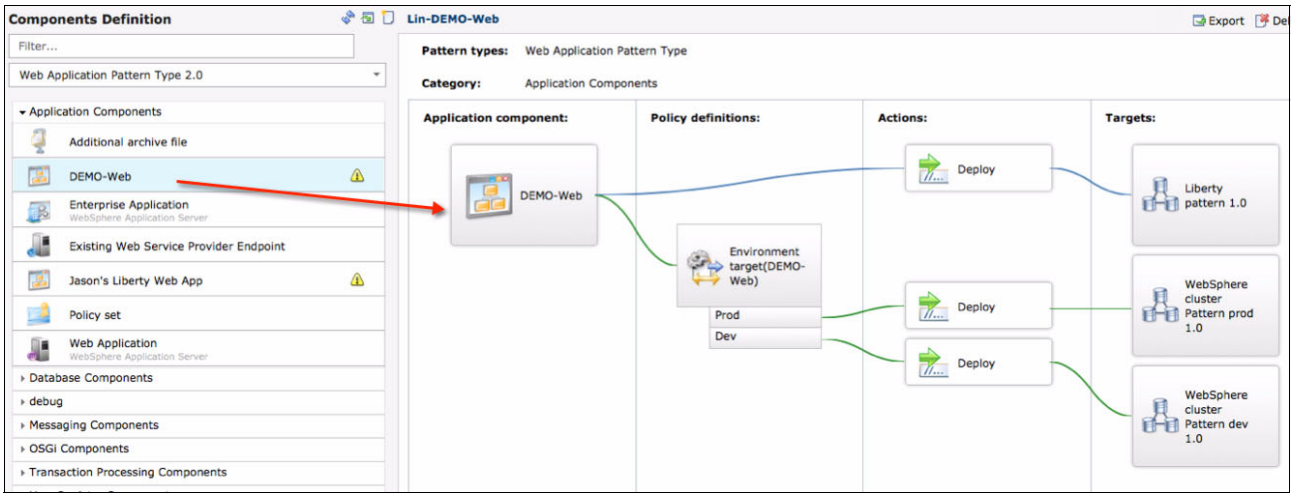


Figure B-9 Pattern component definition

The following list notes some available component definition options:

- Define pattern type, version, and category
- Define input and output
- Define environment target
- Define mapping target pattern

Backup and Restore

Backup and restore is one of the areas where IBM PureApplication System V2.0 has significant improvements. These improvements are explained individually in the following sections:

- Improved GUI functions
- Diverse backup locations
- Enhanced backup profile
  - Component backup supports additional artifact types

Improved GUI functions

The graphical user interface (GUI) has been rewritten with a dedicated *Backup and Restore* page. After opening the PureApplication Console, the page is available by accessing the System Console under the System tab. See Figure B-10.

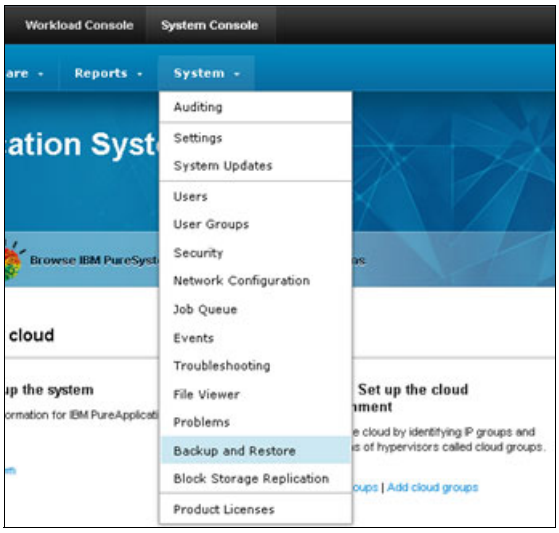


Figure B-10 Backup and Restore initial page

Now it is easy to discover if your user ID has the necessary privileges to work with backup and restore. If you do not have sufficient access, you cannot see the Backup and Restore page in the list shown in Figure B-10. Admin level privileges or the equivalent are required.

Diverse backup locations

You are no longer limited to a single backup server. Now, you can back up what you want and where you want, choosing between two main types of backup: Encrypted or unencrypted.

To start a backup, you need to define *where* to back up data and configure the backup location. To start that process, complete the following steps:

1. From the System Console, select **System** → **Backup and Restore**.
2. Click **Change** to open the backup locations window.

3. Clicking **Create New** as shown in Figure B-11.

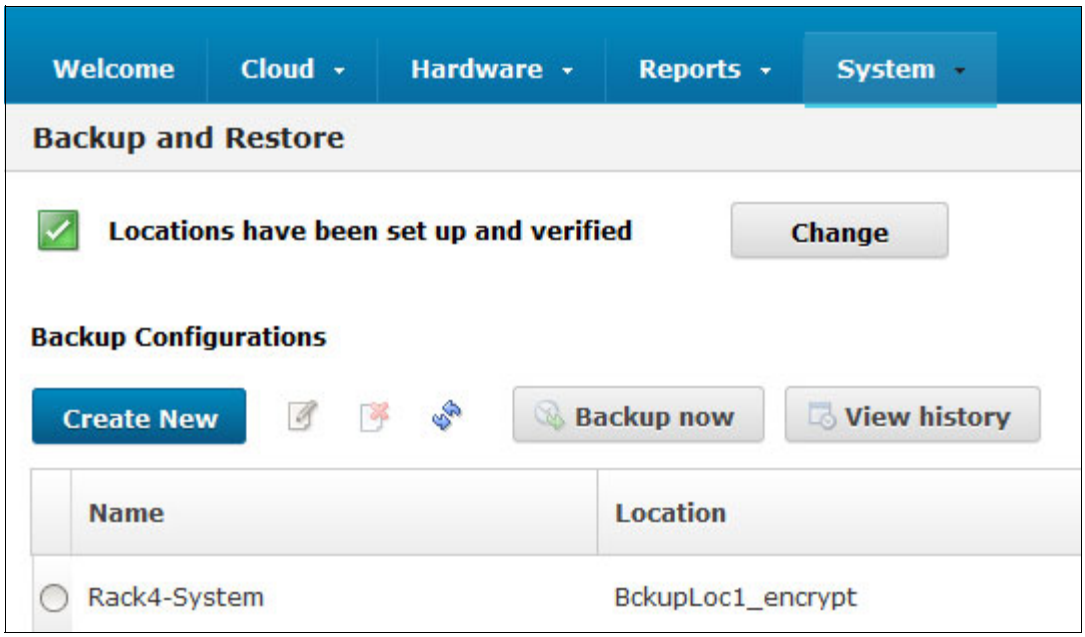


Figure B-11 Backup and restore main page

4. Enter the host name, ip-address, login credentials, SSH port numbers, and so on. As you complete the fields and save the information, you can also set and manage backup locations, as shown in Figure B-12.

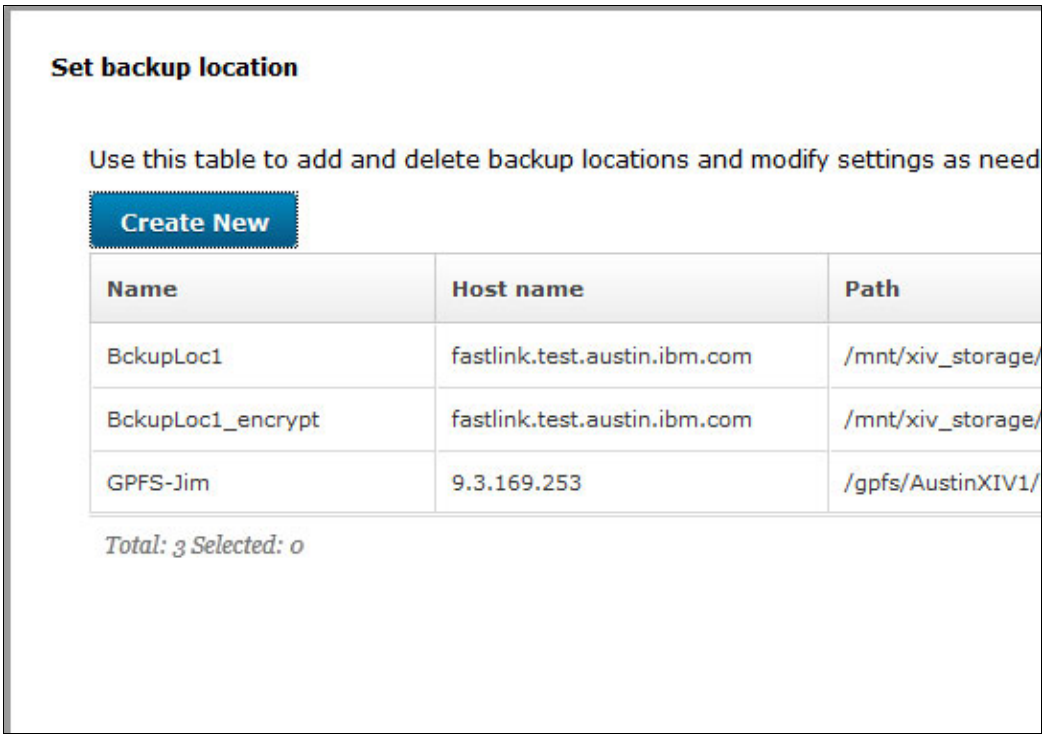


Figure B-12 Create backup locations

Enhanced backup profile

The backup profile now allows you to define what to back up, when the backup is run, and where the backup files are stored. Each backup profile has its own backup schedule. Additionally, IBM PureApplication System v2 has a clear separation between *System Backup* and *Component Backup* (different from v1.1).

To create a backup profile, complete these steps:

1. Open the Backup and Restore main page.
2. Click **Create New** to open the backup profile window.
3. Select the **type** of backup (System Backup or Component Backup).
4. At that point, you are requested to provide a unique name for the profile to select one of the backup locations previously defined, and to indicate an automatic backup schedule (if wanted).

   Figure B-13 shows the *system backup* page.

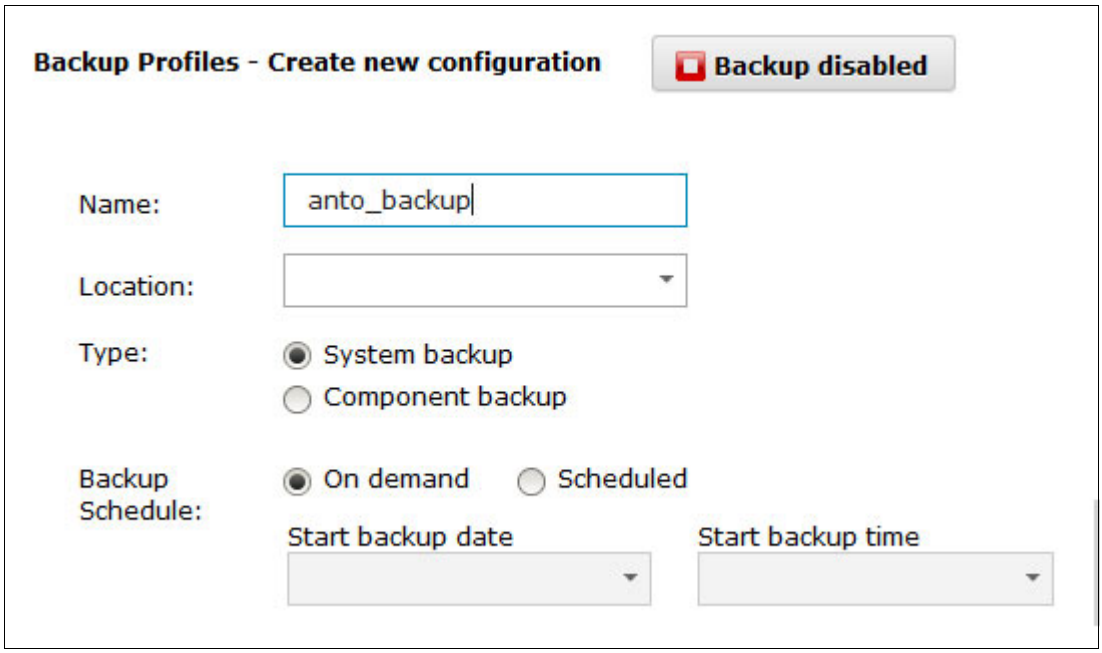


Figure B-13 System backup fields

Figure B-14 shows the *component backup* page. The other tasks that are permitted are **Edit an existing profile**, **View history of a profile** (start time, duration, size, success or fail), and **Delete a profile**.

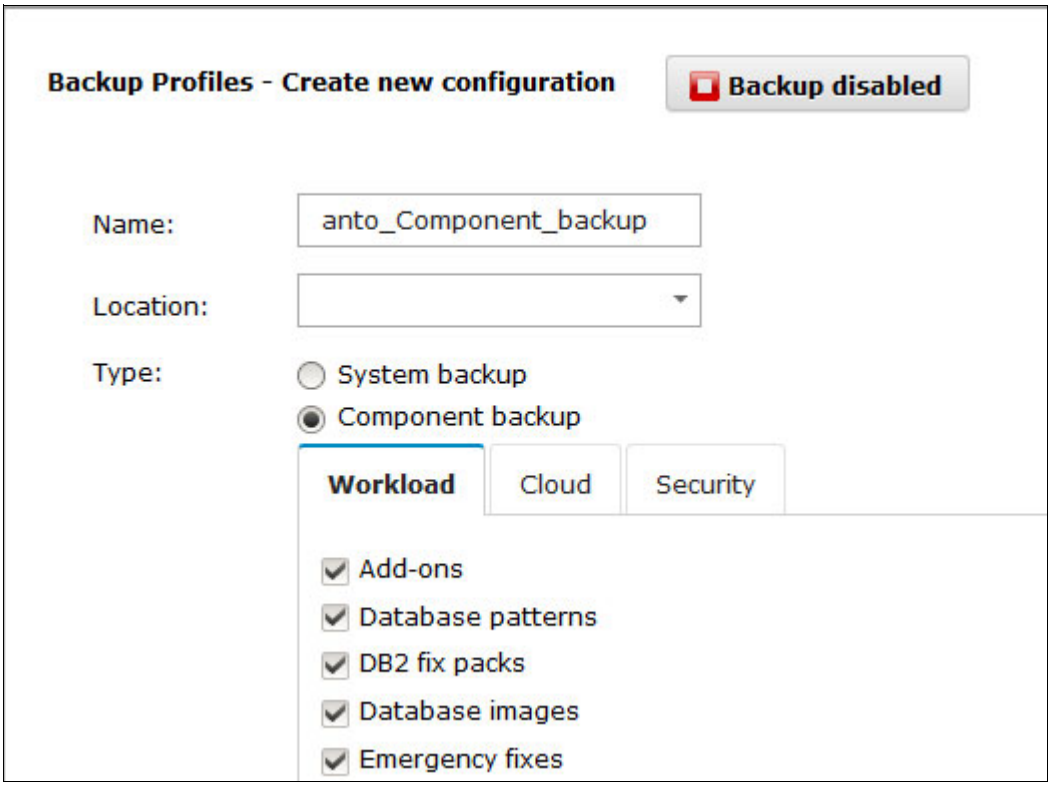


Figure B-14 Component backup fields

Additional artifact types under component backup are supported

To restore an artifact, due to an accidental deletion for example, from the System Console, complete these steps:

1. Click **System** → **Backup and Restore**.
2. Select **restore location** from the Restore section in the Backup and Restore main page.
3. Select the component type of the artifact you want to restore.
4. Select **Only new or changed** so that the table view will then show only the patterns that are missing on the rack.

5. Finally, select one, and click **Restore data** as shown in Figure B-15.

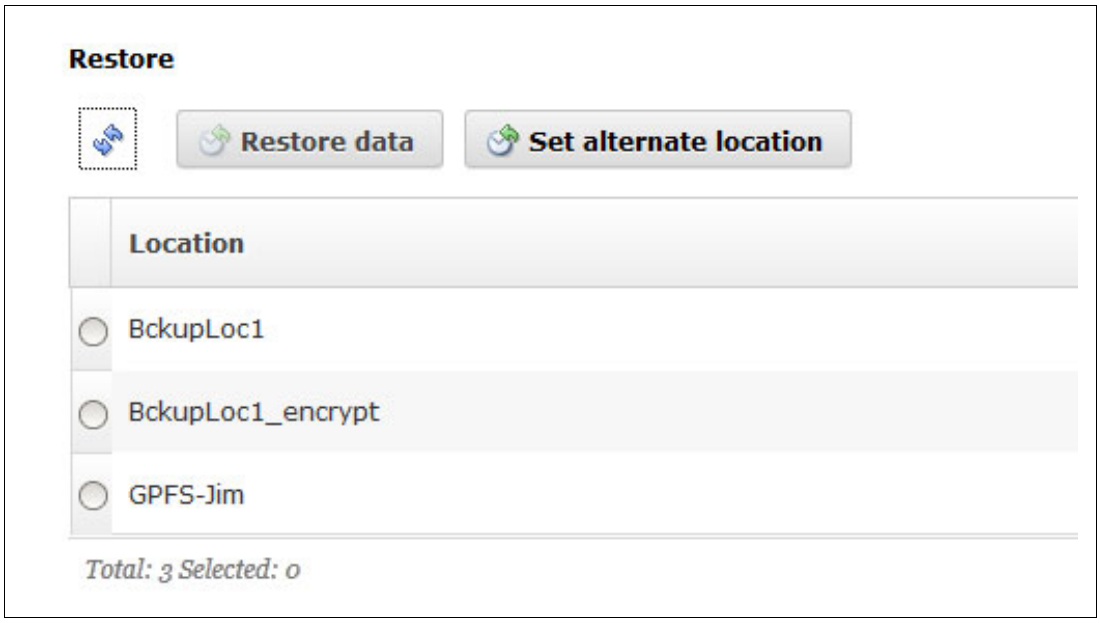


Figure B-15 Example of Restore section with the list of restore locations options.

There are now twenty-two (22) different artifact types supported in the Component Backup. This new approach, an enhancement in version 2, in practical terms allows you to back up everything except the customer deployments (the VMs).

Related publications

The publications listed in this section are considered particularly suitable for a more detailed discussion of the topics covered in this paper.

IBM Redbooks

The following IBM Redbooks publications provide additional information about the topic in this document. Note that some publications referenced in this list might be available in softcopy only.

- *Adopting IBM PureApplication System V1.0*, SG24-8113
- *Cloud Computing Patterns of Expertise*, REDP-5040
- *Creating integrated WebSphere solutions using Application Lifecycle Management*, SG24-8243.
- *IBM PureApplication System Best Practices*, SG24-8145
- *Leveraging Virtual Application Patterns for Business Process Management Deployment*, TIPS1022

You can search for, view, download or order these documents and other Redbooks, Redpapers, Web Docs, draft and additional materials, at the following website:

ibm.com/redbooks

Online resources

These websites are also relevant as further information sources:

- Differences between virtual systems and virtual applications are discussed at:

  http://www-01.ibm.com/support/knowledgecenter/SSAJ7T_1.0.0/com.ibm.websphere.patterns.doc/ae/cins_patterns_diff.html

- Information on importing and installing pattern types:

  http://pic.dhe.ibm.com/infocenter/wrklight/v5r0m6/topic/com.ibm.worklight.help.doc/pureapp/t_pureapp_installing_mobile_application_platform_pattern_type.html

- Creating an installation package at the following website:

  http://www-01.ibm.com/support/knowledgecenter/SSFTN5_8.5.5/com.ibm.wbpm.admin.doc/topics/releasing_installing_procs_D.html?cp=SSFTN5_8.5.5&lang=en

- Exporting RuleApp archives at the following website:

  http://www-01.ibm.com/support/knowledgecenter/SSQP76_8.6.0/com.ibm.odm.dserver.rules.deploying/topics/tpc_res_deploy_rlapp_export_arch.html?lang=en

- Scaling policies:

  http://www-01.ibm.com/support/knowledgecenter/SSNLXH_1.0.0/com.ibm.ipas.doc/iwd/apt_policyov.html

- How to use UrbanCode Deploy:

  https://developer.ibm.com/urbancode

- Management capabilities for Virtual Application pattern:

  http://www-01.ibm.com/support/knowledgecenter/SSNLXH_1.0.0/com.ibm.ipas.doc/workloadconsole/t_mngpatterns.html?lang=en

- External System Monitoring service:

  http://www-01.ibm.com/support/knowledgecenter/SSCR9A_2.0.0/doc/ITM_shared_service/r_itm_shared_service.dita

- PureSystem Centre:

  https://www.ibm.com/software/brandcatalog/puresystems/centre/

- IBM Fix central:

  https://www.ibm.com/support/fixcentral/

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Rapid Deployment of Integrated WebSphere Solutions in Your Cloud

Deploying an integrated WebSphere solution on PureApplication System

Creating an integrated WebSphere solution in your cloud

Capitalizing on the benefits of the cloud

This IBM Redbooks publication uses the same practical solution and is an extension to *Creating Integrated WebSphere Solutions using Application Lifecycle Management*, SG24-8243-00. This paper demonstrates how to take an existing application that was built in a legacy environment, and bring that application to PureApplication Systems, using preferred practices for deployment and automation. The process is illustrated using a business scenario.

This publication is intended for architects, developers, and administrators who want to know about the next generation of technology that modern IT organizations are moving rapidly towards: Application integration and systems development.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

IBM Redbooks are developed by the IBM International Technical Support Organization. Experts from IBM, Customers and Partners from around the world create timely technical information based on realistic scenarios. Specific recommendations are provided to help you implement IT solutions more effectively in your environment.

For more information:
ibm.com/redbooks

REDP-5132-00