

Establishing a Secure Hybrid Cloud with the IBM PureApplication Family

Amit P. Acharya

Tom Bal

Chris Clark

Addison Goering

David Graesser

Angelo Littera

Marcelo Manhaes

Gregory Scott



PureSystems



International Technical Support Organization

**Establishing a Secure Hybrid Cloud with the IBM
PureApplication Family**

February 2016

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

First Edition (February 2016)

This edition applies to Version 2, Release 1, Modification 1 of IBM PureApplication System.

© Copyright International Business Machines Corporation 2016. All rights reserved.

Note to U.S. Government Users Restricted Rights -- Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Notices	vii
Trademarks	viii
IBM Redbooks promotions	ix
Preface	xi
Authors	xi
Now you can become a published author, too!	xiv
Comments welcome	xv
Stay connected to IBM Redbooks	xv
Chapter 1. Enabling a hybrid enterprise	1
1.1 A definition of cloud and hybrid cloud	2
1.2 On premises, off premises, or hybrid	3
1.2.1 Value of hybrid clouds	4
1.2.2 Cloud-enabled or cloud-native applications	5
1.2.3 IBM solutions for cloud-enabled and cloud-native applications	7
1.3 Hybrid cloud is critical to the success of businesses	7
1.4 The IBM PureApplication family approach to hybrid cloud	9
1.4.1 IBM PureApplication System	10
1.4.2 IBM PureApplication Service	13
1.4.3 IBM PureApplication Software	18
1.4.4 Where do the three PureApplication products meet	19
Chapter 2. How to build a hybrid cloud	25
2.1 A five-step roadmap for establishing a hybrid cloud	26
2.1.1 Use cases	26
2.1.2 Assessment	27
2.1.3 Implementation	31
2.1.4 Adoption	39
2.1.5 Continuous optimization	40
2.2 A hybrid infrastructure with the PureApplication family	40
2.3 Patterns and the PureApplication family	50
2.3.1 Pattern engine	50
2.3.2 Choosing the best pattern type for a hybrid cloud	52
2.3.3 Pattern Builder	52
2.3.4 Import and export pattern strategies in PureApplication	54
2.3.5 PureSystems Centre	55
2.4 Open technologies and PureApplication integration	55
2.4.1 Docker	55
2.4.2 Chef	62
2.4.3 OpenStack (Heat and HOT)	67
2.5 Achieving hybrid cloud application portability	68
2.5.1 Virtual patterns and portability	70
2.5.2 Pattern portability with other open technologies	80
2.6 Deploying applications by using a hybrid cloud	82
2.6.1 Using RESTful web services for application integration	82
2.6.2 PureApplication and Bluemix	82
2.6.3 Security at the application level	83

2.7 Achieving business continuity and high availability in a hybrid cloud	85
2.7.1 Principles of high availability	85
2.7.2 Principles of disaster recovery	86
2.7.3 PureApplication family support for high availability and disaster recovery	88
2.7.4 PureApplication System and PureApplication Software backup and restore capabilities	92
2.7.5 Use case scenarios for high availability and disaster recovery in a hybrid cloud scenario	95
Chapter 3. Hybrid use cases	105
3.1 Overview	106
3.2 Development and test environments off premises	107
3.2.1 An extended and dynamic data center	110
3.3 Components off premises and other components on premises	111
3.3.1 The convergence between cloud and mobile	113
3.3.2 IBM MobileFirst Platform on IBM PureApplication Service	114
3.3.3 Implications of this use case	117
3.4 Quick delivery	120
3.5 SMBs that start small on a public cloud and then expand	122
3.6 Start on premises and then move to the hybrid cloud	125
3.6.1 Business value	125
3.6.2 Architecture	126
3.7 Primary on premises and business continuity off premises	129
3.8 Common technology implications of use cases	131
Chapter 4. Implementation considerations	139
4.1 Connectivity	140
4.1.1 IBM PureApplication Service connectivity	140
4.1.2 Reasons that connectivity is required	141
4.1.3 Connectivity use cases	142
4.1.4 Installing and configuring OpenVPN	150
4.1.5 Data center connectivity	154
4.1.6 Network configuration for the TradeLite example	156
4.1.7 Hybrid cloud monitoring of TradeLite	156
4.2 Isolation and security	159
4.2.1 Cloud Groups	159
4.2.2 Networking and Cloud Groups	163
4.2.3 Load balancing	164
4.2.4 Denial-of-service (DoS) protection	165
4.2.5 Securing data at rest	166
4.3 Portability	167
4.3.1 Patterns	167
4.3.2 Docker	172
4.4 Recoverability	176
4.4.1 Backup and recovery in a hybrid cloud	176
4.5 Quick delivery use case with TradeLite	187
4.5.1 Data replication architecture	187
4.5.2 Artifacts to automate deployment	189
4.5.3 Virtual system patterns	190
4.5.4 Required steps to set up data replication	191
4.5.5 CDC management console	193
4.6 Deployment and bringing it all together	200
4.6.1 Test and development off premises and production on premises	201

4.6.2 Application parts off premises, parts on premises, and orchestration.	209
4.6.3 Using PureApplication Software on top of other non-IBM clouds	218
4.6.4 How to cope with an expected or returning increase in load.	221
4.6.5 Checklist	224
Chapter 5. Summary	227
5.1 The meaning of the “cloud” and the “hybrid cloud”	228
5.2 The business value of hybrid clouds	228
5.3 Various cloud or cloud-related technologies	229
5.4 More on business value: Considering the long run	230
5.5 Cloud and enterprise qualities of service	230
5.6 Implementing a hybrid cloud	230
5.7 Paradigm shifts: The hype and the reality.	231
5.8 Reiterating the importance of planning and a BVA	232
5.9 The true promise of the hybrid cloud.	233
Related publications	235
IBM Redbooks	235
Online resources	235
Help from IBM	236

Notices

This information was developed for products and services offered in the US. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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, MD-NC119, Armonk, NY 10504-1785, US

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 jurisdictions 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 provide in any way it believes appropriate without incurring any obligation to you.

The performance data and client examples cited are presented for illustrative purposes only. Actual performance results may vary depending on specific configurations and operating conditions.

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.

Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

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 actual people or business enterprises 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. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at <http://www.ibm.com/legal/copytrade.shtml>

The following terms are trademarks or registered trademarks of International Business Machines Corporation, and might also be trademarks or registered trademarks in other countries.

AIX®	IBM®	PureSystems®
BigFix®	IBM FlashSystem®	Rational®
Bluemix®	IBM MobileFirst™	Redbooks®
Cast Iron®	InfoSphere®	Redbooks (logo)  ®
CloudBurst®	Lotus®	Storwize®
DataPower®	Lotus Notes®	System z®
DB2®	Notes®	Tivoli®
developerWorks®	Optim™	WebSphere®
FlashSystem™	POWER7®	Worklight®
Global Technology Services®	POWER7+™	z/OS®
GPFS™	POWER8®	
HACMP™	PureApplication®	

The following terms are trademarks of other companies:

SoftLayer, and SoftLayer device are trademarks or registered trademarks of SoftLayer, Inc., 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.

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

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

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

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

Find and read thousands of IBM Redbooks publications

- ▶ Search, bookmark, save and organize favorites
- ▶ Get personalized notifications of new content
- ▶ Link to the latest Redbooks blogs and videos

Get the latest version of the Redbooks Mobile App



Download
Now

Android



Promote your business in an IBM Redbooks publication

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

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



ibm.com/Redbooks

About Redbooks → Business Partner Programs

THIS PAGE INTENTIONALLY LEFT BLANK

Preface

This IBM® Redbooks® publication takes you on a hybrid cloud journey with IBM PureApplication® System and Service: from the what, over the why, and to the how. We outline the needs for a hybrid PureApplication cloud and we describe how to build a strategy. We provide advice about the components, including security. Through use cases, we define the need and the strategy for a hybrid cloud implementation with IBM PureApplication System, Software, or Service.

The target audience for this book varies from anyone who is interested in learning more about a true hybrid cloud solution from IBM to strategists, IT architects, and IT specialists who want an overview of what is required to build a hybrid cloud with IBM PureApplication family members.

Authors

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



Amit P. Acharya is a Senior Product Manager for IBM PureApplication System. His role includes influencing the broader portfolio strategy and the product roadmap. He analyzes the market and gathers product requirements that capture client feedback and competitive insights. Before this role, Amit led the strategic quality planning and execution of the IBM Workload Deployer product for private clouds. He has a strong background in enterprise application development and middleware solutions. Amit wrote IBM Redbooks publications about service-oriented architecture (SOA). He contributed to the IBM patent portfolio. He holds a Master's degree in Business Administration from Duke University and a Master's degree in Electrical and Computer Engineering from Purdue University.



Tom Bal is a Senior Certified IT Specialist in the IBM Systems Middleware brand. Tom joined IBM through the Lotus® acquisition in early 2000. For several years, he was part of the services team, assisting clients and IBM Business Partners with Lotus products, including IBM Lotus Notes® and Lotus Workplace. He then became the Technical Pre-Sales Representative for IBM WebSphere® Portal. In January 2012, after two years as a Client IT Architect for the telecommunications industry, Tom rejoined the former IBM Software Group to cover PureApplication System for the region of Europe that includes Belgium, the Netherlands, and Luxembourg. He is a co-author of *IBM PureApplication System Best Practices*, SG24-8145. Tom regularly provides posts for IBM developerWorks® and the Expert Integrated Systems blogs. Tom is a Technical Pre-Sales Engineer for IBM PureApplication products. Tom holds a Master of Arts degree in Translation.



Chris Clark works at IBM as a PureApplication Client Technical Professional. He delights in the contribution of IBM offerings to the improvement of a client's business. His work with PureApplication started in the summer of 2011. Before 2011, he worked with WebSphere offerings, including IBM WebSphere CloudBurst® Appliance. Before IBM, Chris was an Application Programmer who worked directly with operations and users at a manufacturing company. He holds a Masters of Environmental Arts and Sciences degree from the University of Wisconsin Green Bay and a Masters of Business Administration degree from the University of Texas Houston.



Addison Goering is a Certified IT Specialist with the WebSphere Education team. His specialty is the design, development, and delivery of courses in the WebSphere product family. He developed and delivered courses that range from webinars to week-long workshops about products, such as WebSphere Enterprise Service Bus (ESB), IBM Workload Deployer, WebSphere Application Server, WebSphere Business Services Fabric, and WebSphere Business Process Management (BPM). He is the Lead Developer on the WebSphere Education team that develops education about IBM PureApplication System. Addison holds a B.S. in Education from Keene State College in New Hampshire, mainframe certification from DePaul University in Chicago, and several certifications from IBM.



David Graesser is a PureApplication System Technical Specialist with IBM. He focuses on Expert Integrated Systems and cloud computing. He actively supported clients with PureApplication System since its announcement in 2012. Before PureApplication System, David worked with clients and IBM Workload Deployer to design and implement deployment automation solutions. In his current role as a Technical Specialist, David uses his 24 years of experience as a Solution Architect and his work with relational databases and development tools. Dave actively supported clients in UNIX, IBM z/OS®, and Linux environments, specializing in distributed applications, and large warehouse systems.



Angelo Littera is a Senior Technology Architect with IBM Global Technology Services® in Italy. Angelo joined IBM in 1996 and during over 19 years of practical experience, mainly in the Public and Financial sectors, his involvement covered all aspects of the project lifecycle, from engagement to delivery. Angelo is responsible for defining technical solutions in the areas of Cloud Computing and DevOps and coordinates the project activities. Angelo was the technical owner of several complex projects, in which he coordinated working groups of professionals from IBM and non-IBM companies, demonstrating horizontal competence and the ability to quickly acquire vertical skills, when needed. From 1996 to 2006, Angelo was an IT Specialist, and he acquired a deep knowledge of IBM products (WebSphere suite) in the areas of web architecture. Angelo is skilled in application development on the Java platform and Java Platform, Enterprise Edition.



Marcelo Manhaes is a Level 3 Certified IT Specialist for IBM Global Technology Services in Brazil. He has 20 years of experience in IT. He works as an IT Delivery Architect with cloud computing, software automation tools, business analytics, and cognitive computing. He is a teacher in the Universidade Positivo. He was an author in other IBM Redbooks publications, such as *Designing and Coding Applications for Performance and Scalability in WebSphere Application Server*, SG24-7497, and *Adopting IBM PureApplication System V1.0*, SG24-8113. He holds a B.S. in Computer Science from Universidade Federal do Paraná - UFPR and an M.S. in Computer Science from Universidade Tecnológica Federal do Paraná - UTFPR.



Gregory Scott was an academic and programmer before he joined IBM in 2001. He taught Java, XML, web services, SOA, ESB, and related technologies. He became a WebSphere Technical Sales Specialist in 2007, focusing on the WebSphere stack. In that capacity, Gregory published articles for IBM developerWorks about WebSphere Application Server, IBM Rational® Application Developer, and JEE development. Since the beginning of PureApplication in 2012, Gregory was part of the PureApplication team. He recently co-authored an article about how to create JBoss patterns on PureApplication. He has an M.A. from Columbia University and a Ph.D. from the University of Toronto, both in Philosophy.

This project was led by **Margaret Ticknor**, who is an IBM Technical Content Services Project Leader in the Raleigh Center. She primarily leads projects about WebSphere products and IBM PureApplication System. Before Margaret joined IBM Technical Content Services, she 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:

- ▶ Kyle Brown, IBM US, Distinguished Engineer, IBM Cloud Lab Services
- ▶ Jose J De Jesus, IBM US, Executive IT Architect, SWG IBM Global Account
- ▶ Dave Fresquez, IBM US, Service Availability & Performance Mgmt (SAPM) Cloud and Smarter Infrastructure
- ▶ Barry Gower, IBM US, IBM PureApplication Development
- ▶ Larry Heathcote, IBM US, Manager, Application Delivery, IBM Systems Middleware
- ▶ Scott Moonen, IBM US, Senior Software Engineer, PureApplication System
- ▶ Santiago Ortega, IBM US, System Cloud Emerging Technologies
- ▶ Jose Ortiz, IBM US, Senior Technical Staff Member (STSM), PureApplication Service in SoftLayer®, PureApplication System
- ▶ Jim Robbins, IBM US, STSM, IBM PureApplication
- ▶ Rajesh Sankaran, IBM India, Manager, DB2® Patterns
- ▶ Dave Willoughby, IBM US, STSM, Systems Cloud Emerging Technologies
- ▶ Vishwanath Venkataramappa, IBM US, PureApplication Systems Development
- ▶ Bobby Woolf, IBM US, Bluemix® Technical Enablement Specialist

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

- ▶ Karen Lawrence, IBM Redbooks Technical Writer
- ▶ Ann Lund, IBM Redbooks Residency Administrator

Now you can become a published author, too!

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

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

ibm.com/redbooks/residencies.html

Comments welcome

Your comments are important to us!

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

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

ibm.com/redbooks

- Send your comments in an email to:

redbooks@us.ibm.com

- Mail your comments to:

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

Stay connected to IBM Redbooks

- Find us on Facebook:

<http://www.facebook.com/IBMRedbooks>

- Follow us on Twitter:

<http://twitter.com/ibmredbooks>

- Look for us on LinkedIn:

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

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

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

- Stay current on recent Redbooks publications with RSS Feeds:

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



Enabling a hybrid enterprise

This chapter provides an overview of why hybrid enterprise is critical to the success of business and how the IBM PureApplication family of products provides the critical components to build that hybrid enterprise.

The following topics are covered:

- ▶ A definition of cloud and hybrid cloud
- ▶ On premises, off premises, or hybrid
- ▶ Hybrid cloud is critical to the success of businesses
- ▶ The IBM PureApplication family approach to hybrid cloud

1.1 A definition of cloud and hybrid cloud

According to the National Institute of Standards and Technology (NIST), by definition, a cloud needs to adhere to the following criteria:

- ▶ On-demand self-service: The ability for users to interact with a portal-like interface or with a command-line interface (CLI).
- ▶ Broad network access: A ubiquitous network or the *network everywhere* concept.
- ▶ Location-independent resource pooling: You do not need to know the exact location of the resources that are consumed.
- ▶ Rapid elasticity: The capability to easily, smoothly, and rapidly increase and release resources that are required to run a particular application.
- ▶ Measured service, for instance, in a pay-per-use model: Measuring resources that are consumed by applications and billing them correctly.

Details of these criteria are available at this website:

<http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>

If your computing environment meets these criteria, you can call it a form of cloud computing.

NIST recognizes four types of deployment models, or four types of cloud infrastructures that you can provide:

- ▶ Private cloud, for exclusive use by a single organization
- ▶ Public cloud, for use by the general public and maintained by a cloud provider
- ▶ Community cloud, which is used by a community from organizations that have, for example, a considerably similar mission, such as government agencies (versus businesses or educational institutions)
- ▶ Hybrid cloud, which is a combination of cloud infrastructures, each functioning as an individual entity, yet sharing technologies to promote efficiencies

The term *hybrid* is derived from the Latin word *hybrida* or *ibrida*, which describes an offspring of plants or animals of a different variety or species. Recently, more general characterizations are incorporated in the term *hybrid*, which are synonymous to heterogeneous.

In information and communications technology (ICT), the term *hybrid cloud* refers to a cloud solution that involves combining cloud services from different deployment models:

- ▶ On-premises and off-premises services
- ▶ Public and private services
- ▶ Any combinations of all of these types of cloud infrastructure services

Figure 1-1 shows a graphic view of the differences among these cloud deployment models.

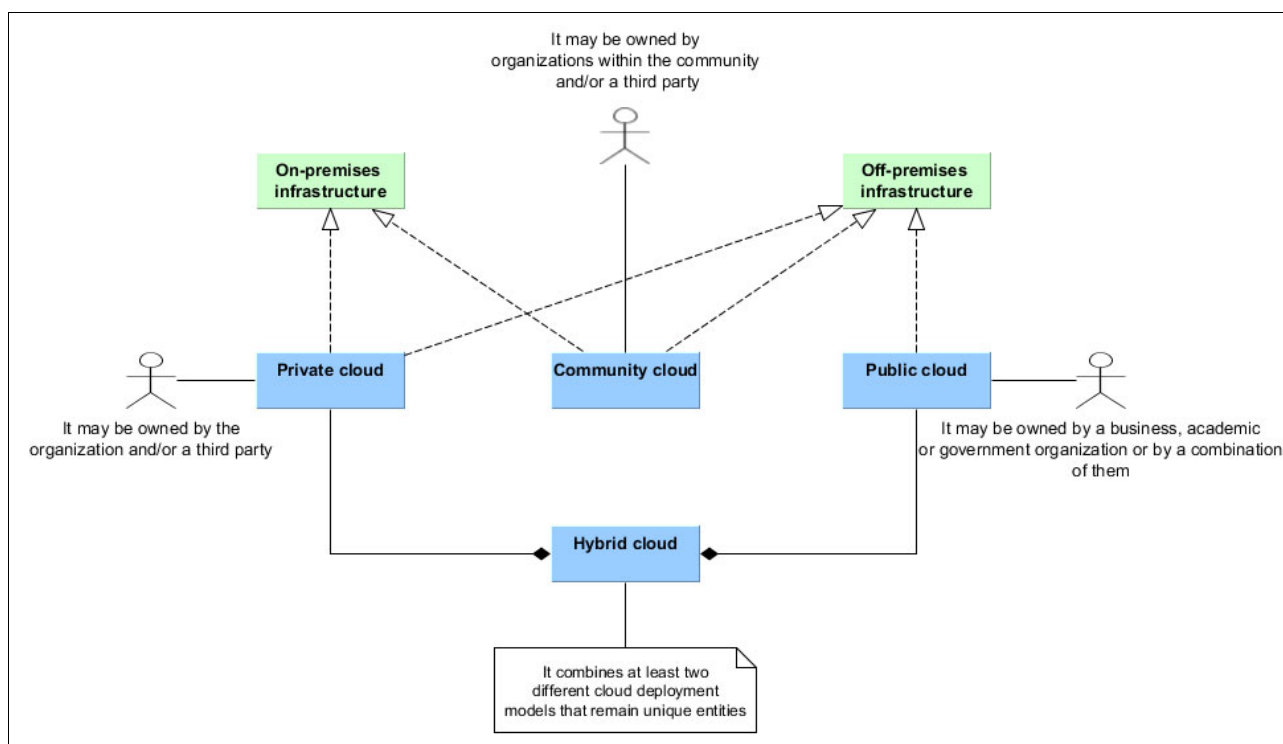


Figure 1-1 Comparison of cloud computing deployment models

The characteristics that are shown in Figure 1-1 offer businesses a mixture of advantages. However, businesses can reap further benefits as their workloads and applications move among these environments, when needed. Alternatively, these businesses can run applications in a cost-effective model, which provides them with even more flexibility and deployment options.

A company that relies on a hybrid cloud architecture, either wholly or in part, can be considered a *hybrid enterprise*.

1.2 On premises, off premises, or hybrid

Being hybrid is a natural consequence of any progressive transformation of an enterprise. Choosing either a complete transformation or no transformation is not a choice because making that choice is not an option. Today, the most common choice is to go hybrid.

Before we explain why the hybrid cloud might be one of the most common scenarios in the future, it is useful to mention its definition by the National Institute of Standards and Technology (NIST)¹:

Hybrid cloud. The cloud infrastructure is a composition of two or more distinct cloud infrastructures (private, community, or public) that remain unique entities, but are bound together by standardized or proprietary technology that enables data and application portability (for example, cloud bursting for load balancing between clouds).

¹ The NIST Definition of Cloud Computing, Special Publication 800-145:
<http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>

1.2.1 Value of hybrid clouds

Several use cases express the value of the hybrid cloud, which is useful in various scenarios. Presently, technologies can provide multiple ways to address your specific needs, including technical, functional, and non-functional needs (for example, government regulations or short-term financial exigencies of your company).

The on-premises model was the traditional model of deployment, which provided control over the deployment of resources, flexibility in choosing infrastructure components, and it followed a strict organization governance and security model that was perfected and put in place for years. This model benefitted the organization in the past few years but due to the advent of social, mobile, and analytics, organizations are required to deliver IT much faster and in an agile manner.

Alternatively, the off-premises (or public cloud) model provides economies of scale, rental economies, and faster time to value.

Many times, organizations debate whether the approach to cloud is on premises *or* off premises. The answer is simple and it is neither. Considering the investments that an organization made in the past, and looking at the benefits of both on premises and off premises, the approach to cloud is hybrid. The hybrid approach provides the agility and time to value while it uses existing investments on premises and the economies of public cloud.

The following hybrid enterprise use cases (not a complete list) demonstrate the assignment of the resources and functionalities on premises and off premises. (In this publication, the discussion of hybrid clouds that consist of two private on-premises clouds, or a public off-premises cloud that connects to a private off-premises cloud, is out of scope.)

Consider these hybrid enterprise use cases:

- ▶ Develop and test on a public cloud and run workloads on premises.
- ▶ Run front-end workloads off premises and back-end workloads on premises.
- ▶ Start your workloads on premises, then migrate to an off-premises cloud as part of a corporate strategy.

The second use case that is listed is the most representative of the hybrid nature. It brings persistent integration at the application level between two worlds, private and public cloud. This use case is also commonly associated with the integration between the world of *systems of engagement (SOE)* and the world of *systems of records (SOR)*.

Geoffrey Moore introduced the concept of SOE in his paper “*Systems of Engagement and the Future of Enterprise IT*”²:

https://www.google.com/?gws_rd=ssl#q=Systems+of+Engagement+and+the+Future+of+Enter

In his paper, Moore comments how social media websites, such as Twitter, Facebook, and Instagram, permanently changed the way that we communicate among ourselves and how people want to interact with the services. Therefore, it is crucial that enterprises are aware of this change, adapting their business models to this new way of communication. From a technology perspective, the SOE represents this shift.

² *Systems of Engagement and the Future of Enterprise IT*, by Geoffrey Moore:
<http://www.aiim.org/Research-and-Publications/Research/AIIM-White-Papers/Systems-of-Engagement>

Conversely, *systems of record* (SOR) represent the traditional way to design systems and applications. With this concept, applications focus on business transactions and strictly depend on discrete pieces of information (*records*). Systems of records rely on the following aspects:

- ▶ Data and process integrity
- ▶ Resilience and high availability
- ▶ Stability and slow evolution

In contrast, SOE might be considered a social media platform, and applications that belong to that context focus on the following aspects:

- ▶ Rapid evolution
- ▶ Real-time connection to other people
- ▶ Emphasis on improving the interaction between a business and its customers
- ▶ Analytics as a way to create or change business models

However, SOE and SOR must be interconnected, complementary worlds. An SOE without an integration to an SOR does not make much sense for traditional enterprise clients with decades of investment and proven reliability in back-end systems. Although, perhaps startup companies or new projects might locate the SOE and the new SOR record on the public cloud (with the need, then, to ensure the same kind of old-fashioned reliability and security). From this perspective, the SOR environment must evolve with the SOE environment so that it becomes larger and more robust, and it can integrate and support the growth of SOE with more traditional information. Do not ignore the SOR when you build an SOE. It is similar to the discussion of stateful versus stateless: companies cannot do one without the other.

1.2.2 Cloud-enabled or cloud-native applications

Cloud solutions are often described as cloud-enabled applications, and the concept of cloud-enabling applications versus cloud-native applications (or *born-on-the-cloud applications*, as they are sometimes called) exists. A place exists for both types of applications. We do not suggest that you attempt to force all of your applications to be all of either type. Certain applications are not designed to run entirely in the cloud, and other applications are not designed to run behind the boundaries of a secure data center.

Cloud-native and cloud-centric applications are applications that are developed specifically for the cloud and usually support the following capabilities:

- ▶ Multitenancy
- ▶ Automatic and elastic resource scaling
- ▶ Integration that is based on open standards
- ▶ *Eventual consistency* for data, which implies that, unlike a bank account that must be the same no matter where you log in to the account, the data might differ for short periods in different parts of the world, without harm, such as for weather forecasts, until all systems are updated consistently and periodically
- ▶ Hyperscale applications to many requests (in the millions) from various devices, such as phones, tables, and the Internet of Things (IoT)

In contrast, existing applications that were originally developed for traditional on-premises IT likely have special requirements or do not support certain cloud capabilities:

- ▶ A limited set of supported platforms, such as an older version of an operating system.
- ▶ A strong integration with on-premises security services (for example, authentication, authorization, and auditing).

- ▶ A strong integration with on-premises application services.
- ▶ Missing support to multitenancy capabilities.
- ▶ High availability and scalability are present, but the automatic and elastic scaling of the resources is missing.
- ▶ Existing investments in automation and frameworks for optimizations that were made over many years.

Key components of the cloud solution

The key question is whether you can take existing applications and eventually transform them into cloud-enabled applications? Is the transformation simple enough to avoid a complete overhaul of the architecture? To be effective, cloud applications must be able to use the entire cloud ecosystem. For any type of cloud that you build, the cloud solution relies on at least a few of the following key components:

- ▶ Integration services at the application, middleware, and physical levels.
- ▶ Fault-tolerant network connectivity or network services.
- ▶ Security-related services.
- ▶ Self-service concepts.
- ▶ Deployment automation.
- ▶ Development operations (DevOps) approach.

Note: *DevOps* is an enterprise capability to enhance the software delivery pipeline by facilitating a close collaboration between developers and IT Operations. DevOps uses software to provide a continuous delivery that is as automated as possible, and that checks code and configuration as they are promoted from environment to environment (such as integration to quality assurance, staging, and production). Finally, by providing the same kind of software tool that all parties can use (whether development or operations), DevOps can alleviate the silos and the inefficient communication that can occur.

- ▶ Modularity on the application level, which is often referred to as *microservices*, is a new upcoming architectural pattern.

Note: *Microservices* refer to an architecture style, which prescribes building large complex software applications by using many small services (or microservices). These microservices are narrowly focused, independently deployable, loosely coupled, language-agnostic services that fulfill a business capability. Multiple microservices communicate with each other by using language-agnostic application programming interfaces (APIs), such as Representational State Transfer (REST).

These microservices are applications in themselves and are often owned by small teams. Unlike the normal practice, the team that coded the microservices is also responsible for its support.

For more information, see these websites:

- ▶ *5 Things to Know about microservices:*

https://www.ibm.com/developerworks/community/blogs/5things/entry/5_things_to_know_about_microservices?lang=en

- ▶ *Microservices from Theory to Practice: Creating Applications in IBM Bluemix Using the Microservices Approach*, SG24-8275:

<http://www.redbooks.ibm.com/redpieces/abstracts/sg248275.html?Open>

1.2.3 IBM solutions for cloud-enabled and cloud-native applications

When you look at IBM technology for cloud-enabling applications, the IBM PureApplication product family provides the quickest way to cloud-enable applications. These products incorporate the components that are listed in “Key components of the cloud solution” on page 6. For in-depth information about this product family, including the specific added value of each product, see 1.4, “The IBM PureApplication family approach to hybrid cloud” on page 9.

Alternatively, the easiest way to deliver cloud-native applications is by the use of IBM Bluemix, which is a platform as a service (PaaS) that is available off premises and on premises. Explaining how to build a hybrid cloud by using either Bluemix environment is beyond the scope of this book. For more information, see the following Bluemix source of information:

<https://ibm.biz/BdHhsg>

Introducing a few cloud solutions can result in reduced costs, limited risks, and a shorter time-to market model (where the products or services are brought to the consumer in a quicker and more agile way). However, be aware that this approach might add another layer of governance to your information and communications technology (ICT) systems. For instance, the hybrid cloud requires a change from the traditional way of managing systems:

- ▶ Performing backup and restore operations
- ▶ Applying system or middleware updates or fixes
- ▶ Monitoring applications or separate systems
- ▶ Implementing security

These functions are performed differently, if only partially, when you use a hybrid cloud solution to run the business. When you bring the enterprise to a cloud, carefully consider these aspects of cloud computing. Several of these aspects are described in Chapter 3, “Hybrid use cases” on page 105.

1.3 Hybrid cloud is critical to the success of businesses

Several of the primary reasons for turning to a cloud solution are described. All of the following reasons share one common element: The environment that you create is intended to provide a blueprint, the basis of which must be identical for all other environments (development, test, acceptance, and production). Within the IBM PureApplication products, we call this blueprint a *pattern*.

A pattern, such as those patterns that are provided by the IBM PureApplication family, is a true asset in this scenario. An example of a pattern is an IBM Integration Bus and Linux, or a WebSphere Application Server cluster with a DB2 database and HTTP servers on IBM AIX®, that can be self-provisioned with a few clicks of the mouse, or with a REST call and two passwords.

Several primary reasons for turning to a hybrid model of deployment are listed:

- ▶ To improve your business *time-to-market model*. Organizations need access to their applications, data, and other business functions. They need the flexibility to create new business opportunities and address business problems. For these clients, it is key to find an environment that is ideal for their problems. Speed and agility for building the supporting IT systems are common drivers for enterprises that are considering the move to an IBM cloud solution, whether it is hybrid, private, or public.
- ▶ The move to a cloud solution relates to *budgets*. Over the last few years, IT divisions saw their overall budgets decrease, yet they are required to maintain an equal level of services and they are expected to expand to further accommodate business growth. With budgets under such pressure, IT divisions look for cost-effective ways to service their lines of business (LOBs). When a cloud service can reduce part of these costs and decrease the level of complexity, the decision is easy.
- ▶ Moving applications to the cloud can *reduce further risk*. For example, when the cloud provider offers an integrated solution, such as networking, storage, and operating systems, the responsibility for those components shifts from the enterprise IT division to the cloud provider. This shift has an impact on the risks that businesses face in the areas of vulnerabilities, maintenance updates, security updates, and so on.
- ▶ *Repeatability*, which is the ability to quickly reproduce an environment for different motivations, is another key driver for a cloud solution, for example:
 - Training: To support training, an environment that is identical to, or at least similar to, the production environment is needed.
 - Troubleshooting: An identical environment is necessary to troubleshoot an issue without touching the production environment.
 - Pre-production deployment testing: Before you apply an update, ensure that the change or update to a part of the architecture does not negatively affect the entire environment.
- ▶ Closely linked to the previous reasons for moving to the cloud is the ability to provide or support *DevOps* services. DevOps services provide the infrastructure and environments for running these applications, in addition to the agility and ability to deploy, redeploy, and maintain applications easily.
- ▶ *Efficiency* is provided by the single dashboard for managing all applications or all instances, including logging and monitoring. The dashboard offers application owners an end-to-end overview of application health.
- ▶ When clients are looking for full, built-in security, isolation, and compliance, a cloud solution can help. Off-premises solutions, with or without managed services, often include security, isolation (physical or logical), and compliance to certain industry standards. Most of the IBM PureApplication family of products offers these capabilities because of the products' tight integration, or options for tight integration, with the underlying hardware.

With the advent of social, mobile, and analytics, clients now demand faster access to data and quicker decision making. Organizations now adopt mobile-first strategies. These expectations disrupt the way that technology delivers and integrates services, and cloud further helps with the adoption of such strategies.

To see the IBM hybrid cloud in action, visit this web page for a demo:

<https://www.youtube.com/watch?v=-s93Xx6wKx4>

In mid 2014, Forrester Consulting conducted their Total Economic Impact study, which examined the potential return on investment (ROI) that enterprises might realize when they deploy an IBM PureApplication System. This study is publicly available and you can download it at one of the following websites:

- ▶ <ftp://ftp.software.ibm.com/software/websphere/pure-app/TEI-IBM-PureApplication-System.pdf>
- ▶ <https://ibm.biz/BdXgDP>

For a summary of enterprise application portability in the cloud, see the following document:
<https://ibm.biz/BdHnB3>

1.4 The IBM PureApplication family approach to hybrid cloud

The IBM PureApplication product family currently consists of the following three products, as shown in Figure 1-2:

- ▶ IBM PureApplication System
- ▶ IBM PureApplication Service
- ▶ IBM PureApplication Software

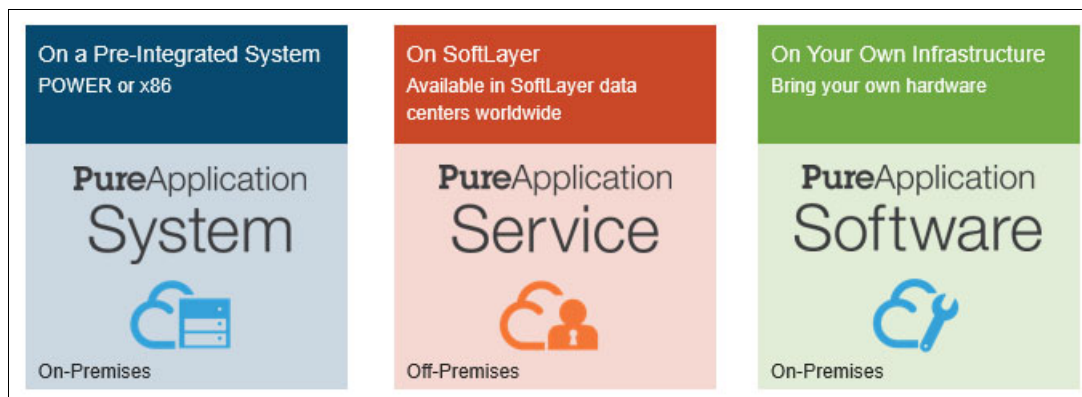


Figure 1-2 Current IBM PureApplication product family

Each product has its own strength and each product complements the others to provide a full approach to build a complex hybrid cloud solution. Whichever product you choose, the user experience remains the same, which makes the transition from one IBM PureApplication product to another IBM PureApplication product easy and straightforward.

The characteristics and the added value for each product are described. For a comprehensive, cross-referenced overview of the PureApplication offerings, see 1.4.4, "Where do the three PureApplication products meet" on page 19.

1.4.1 IBM PureApplication System

IBM PureApplication System is a cloud application platform on a rack. It is a ready-to-use solution that merges the following components:

- ▶ Infrastructure
- ▶ Cloud capabilities
- ▶ System management capabilities
- ▶ Predefined patterns (available without confirmation), such as WebSphere Application Server and DB2, importable patterns from IBM and non-IBM vendors, and built-in tools to easily customize patterns, as long as they run on RHEL 6.x, AIX, or Windows 2012 operating systems

At the *infrastructure* level, PureApplication System is a pre-integrated machine, which includes compute nodes, storage, and networking. Since the inception of the PureApplication System, IBM released three generations of PureApplication System. Currently, five models are available for different architectures (IBM Power or Intel) and sizes, as shown in Table 1-1.

Table 1-1 Different types and models for an IBM PureApplication System

Model	Year	Compute nodes	Memory	Storage	Networking
W1500	2012	32 - 608 Intel CPU cores	0.5 - 2.0 terabytes (TB)	2.4 - 6.4 TB solid-state drive (SSD) 24.0 - 48.0 TB hard disk drive (HDD)	Two 64-port 10-Gigabyte Ethernet switches
W1700		32 - 608 IBM POWER7® CPU cores	1.0 - 9.5 TB	2.4 - 6.4 TB SSD 24.0 - 48.0 TB HDD	Two 64-port 10-Gigabyte Ethernet switches
W2500	2014	32 - 384 Intel CPU cores	16 - 32 gigabytes (GB) per core	2.4 - 6.4 TB SSD 24 - 48 TB HDD	Two 64-port 10-Gigabyte Ethernet switches
W2700		32 - 384 IBM POWER7+™ cores	16 - 32 GB per core	2.4 - 6.4 TB SSD 24 - 48 TB HDD	Two 64-port 10-Gigabyte Ethernet switches
W3700	2015	40 - 200 IBM POWER8® cores	32 GB per core	6.4 TB SSD 96 TB HDD	10-Gigabyte dual top of rack (TOR)

At the *cloud* level, PureApplication System provides several capabilities:

- ▶ Support for growth to larger systems without a system outage
- ▶ Built-in elasticity of application workloads
- ▶ PaaS model support
- ▶ Fast pattern-centric deployment

At the *system management* level, PureApplication System includes features and capabilities that are needed to support the management of the entire platform, in addition to lifecycle management of deployed applications:

- ▶ Integrated and optimized compute, network, and storage resources
- ▶ Monitoring and management tools through an integrated user interface

- ▶ Preinstalled management software
- ▶ Updates for the entire infrastructure stack
- ▶ Centralized access control
- ▶ License tracking
- ▶ Support for IBM and partner content that is optimized for PureApplication System, in addition to third-party software
- ▶ Single point of contact for IBM hardware and software support
- ▶ Patterns and policies that allow the state management (for instance, defining how and when the application needs to scale) that you want

Patterns of expertise

One of the key value propositions of an IBM PureApplication System family of products is the concept of patterns of expertise. Patterns of expertise introduced a new way of providing integrated hardware and software, together with built-in expertise, to drastically simplify the provisioning of cloud applications. Patterns of expertise are the core of the deployment model of the PureApplication products. They are available in three types:

- ▶ The *virtual appliance pattern* provides a way to design a complete black box, including operating system, middleware, and application in a single package. Often, these packages are open virtualization archive (OVA) or open virtualization format (OVF) image files.
- ▶ The *virtual system pattern* provides a system-centric approach to designing a cloud application in terms of topologies of interconnected virtual machines.
- ▶ The *virtual application pattern* provides an application-centric approach to designing a cloud application in terms of high-level components, such as a Java Platform, Enterprise Edition (JEE) application, database, and queues.

Figure 1-3 shows how the three types of virtual patterns serve different purposes and strengths.

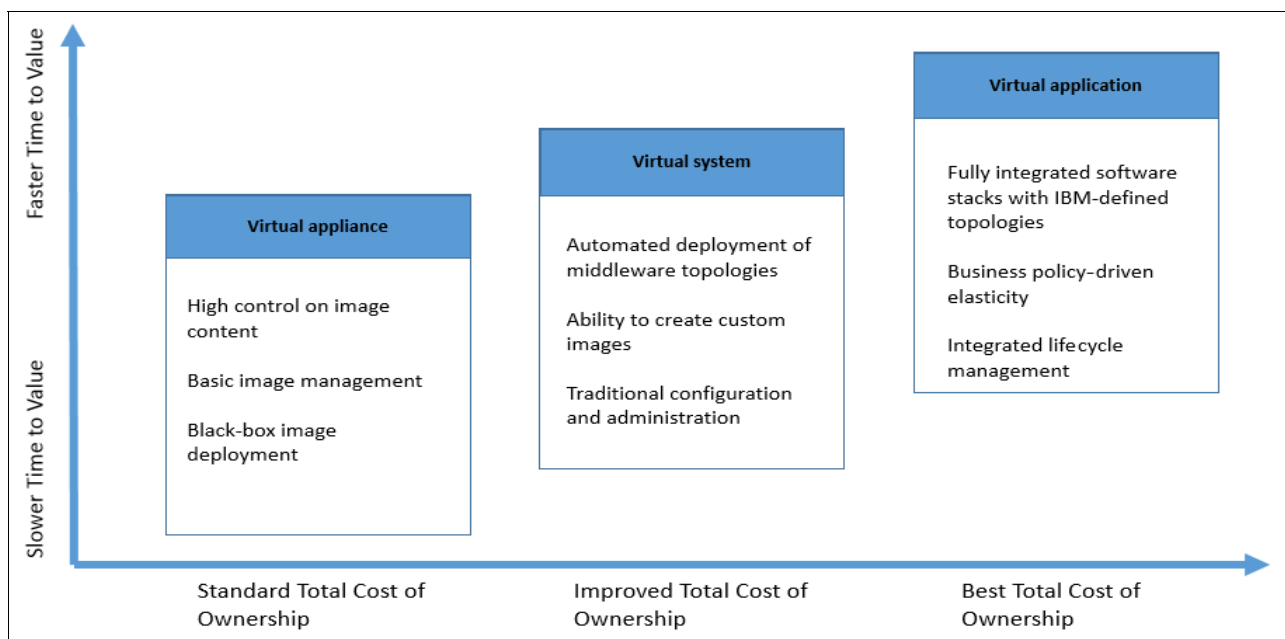


Figure 1-3 Different purposes and strengths that are offered by IBM PureApplication patterns

For a detailed description of the pattern engine and patterns, see 2.3.1, “Pattern engine” on page 50.

One major enhancement that was introduced by IBM PureApplication Version 2.0 was the support for multisystem management and deployment. This feature, which is known as *multi-rack*, offers the opportunity to centrally manage a group of PureApplication Systems with the following benefits:

- ▶ Simplified systems management
- ▶ Simplified (cloud) application portability
- ▶ More flexibility in the implementation of high availability scenario for products, such as IBM WebSphere Application Server, IBM DB2, and IBM Business Process Manager

Multisystem management and deployment capabilities rely on the concepts of a management domain and a deployment subdomain.

The creation of a *management domain* is needed to provide a management relationship between two or more PureApplication Systems, with no limit on the number of systems. The following minimum requirements must be addressed to create a management domain:

- ▶ All PureApplication Systems must interconnect through an IP network.
- ▶ One PureApplication System can belong, at most, to a single management domain.
- ▶ All PureApplication Systems must share Lightweight Directory Access Protocol (LDAP) user registry.

The creation of the *deployment subdomain* is needed for deploying a single virtual pattern across multiple PureApplication Systems that belong to the same management domain. The following minimum requirements must be addressed:

- ▶ Create a deployment subdomain only after at least one management domain is defined.
- ▶ Only two systems can be members of a deployment subdomain.
- ▶ Members of the deployment subdomain must belong to the same management domain.
- ▶ One PureApplication System can belong to, at most, one deployment subdomain.

Multisystem management and deployment capabilities are shown graphically in Figure 1-4.

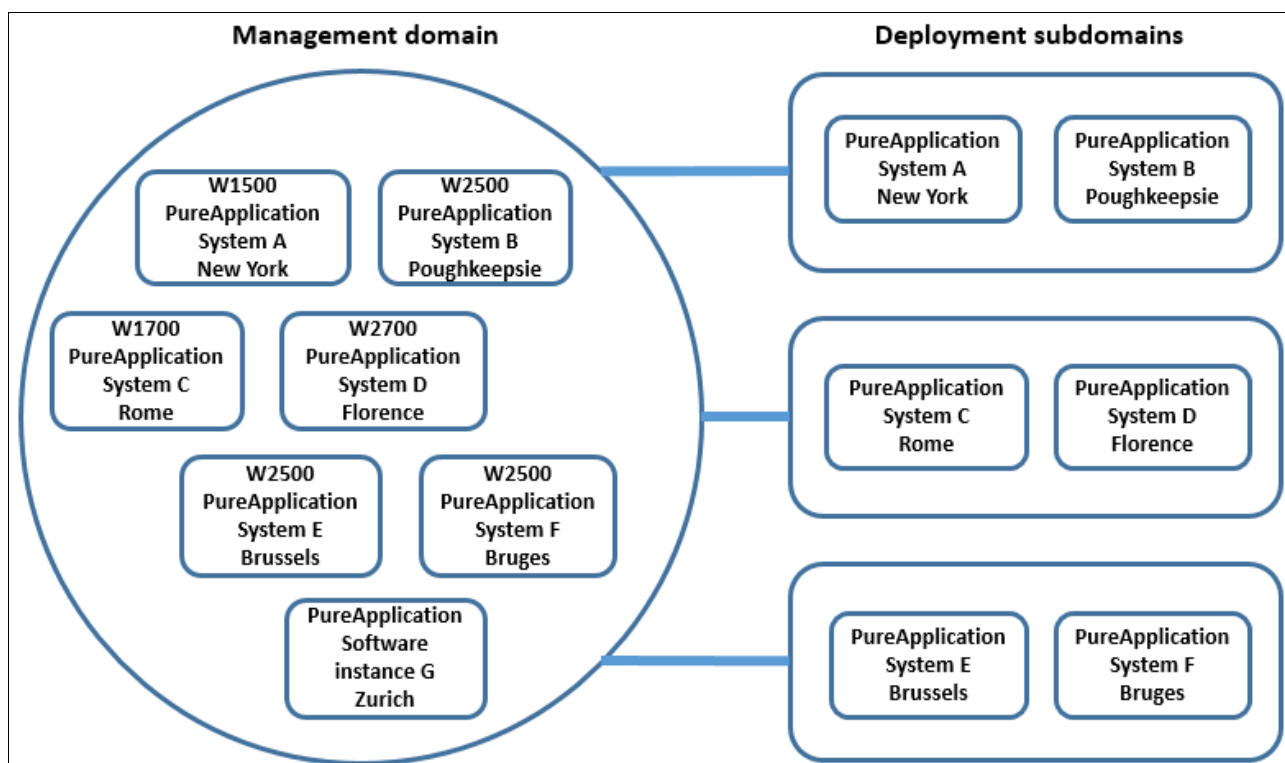


Figure 1-4 Management domains and deployment subdomains explained

Important note: Multisystem deployment is only available for identical types of systems (Power or Intel).

Additional installation and configuration information about multisystem management and deployment is available in *IBM PureApplication System V2.0: Multi-target deployment* in the IBM Knowledge Center:

<https://ibm.biz/BdHEb6>

For more information about how to set up a management domain and deployment subdomain by using IBM PureApplication System, see *Implementing multi-system management and deployment with IBM PureApplication System*:

<https://ibm.biz/BdHEbU>

1.4.2 IBM PureApplication Service

IBM PureApplication Service is a cloud application platform that uses SoftLayer to bring the value and capabilities of the IBM PureApplication System off premises.

This solution was released in 2014 and provides the same workload deployment and lifecycle management capabilities of PureApplication System, but in a dedicated and isolated environment on a public cloud.

PureApplication Service shifts the focus to the public deployment model, and in addition to the virtually identical graphical user interface (GUI) for management and workload functionality that it shares with PureApplication System, PureApplication Service provides a self-service portal where the cloud user is able to perform these functions:

- ▶ Purchase and provision instances
- ▶ Configure CPU cores, memory, storage, and networking functions
- ▶ Support subscription through a purchase order
- ▶ Configure two-way virtual private network (VPN) connections between the premises server and PureApplication Service with the console

The scope of this book is not to provide a detailed user guide of the PureApplication Service self-service portal. However, it is important to understand several functionalities of the self-service portal, such as configuring and modifying instances in the public cloud of PureApplication Service. Figure 1-5 shows the home page of IBM PureApplication Service.

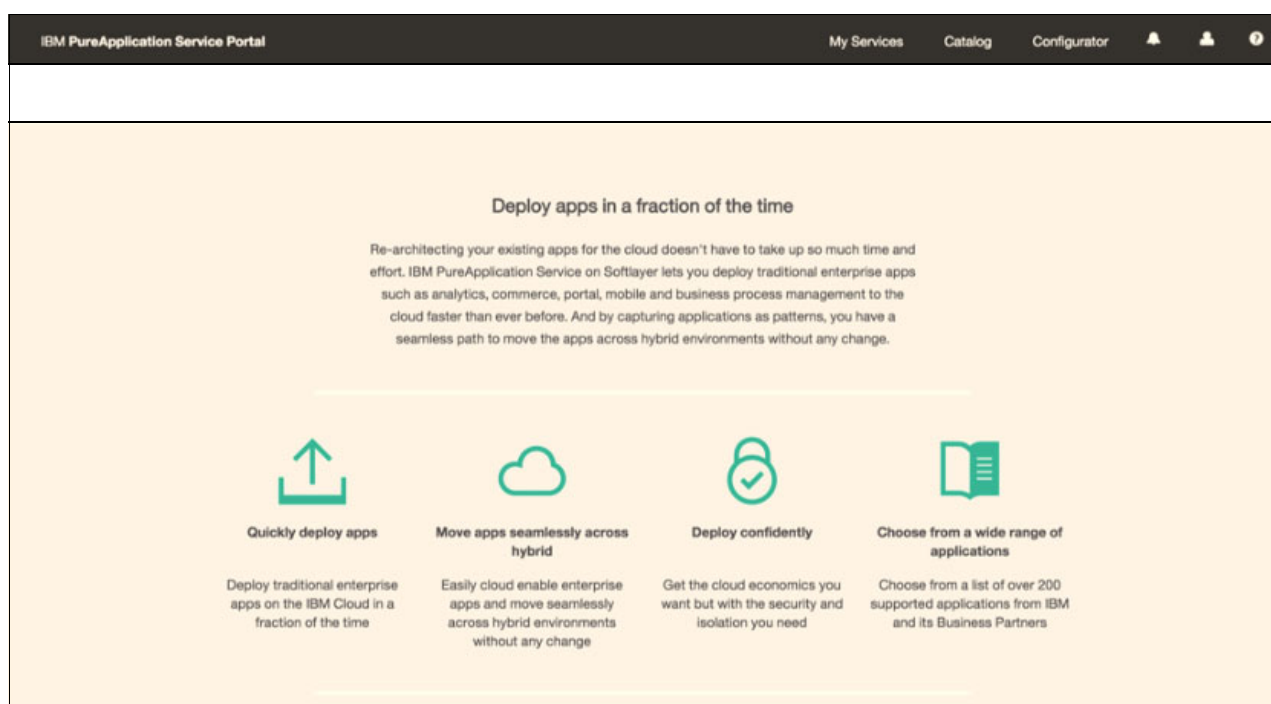


Figure 1-5 The home page of the IBM PureApplication Service self-service portal

The following brief explanations describe each main menu item in Figure 1-5:

- ▶ **My Services:** View your PureApplication Service instances, modify an existing service instance, or navigate to the PureApplication Service console to manage your environments.
- ▶ **Catalog:** Explore the available patterns on PureApplication Service, share patterns with colleagues, and select patterns to include in PureApplication Service instances.
- ▶ **Configurator:** Provision new PureApplication Service instances, choose the data center, add cloud groups, and configure other specifications.
- ▶ **Notifications:** A quick view of your latest notifications and a full history of all notifications.
- ▶ **My Account:** View your personal account, specify notification preferences, and log out of the self-service portal.
- ▶ **Help:** Links to the Help portal with frequently asked questions, tutorial videos, and links to other support materials and communities.

You can provision a new environment on PureApplication Service by selecting **Configurator**. Begin the creation of a draft instance, and the window in Figure 1-6 displays the available options for an environment.

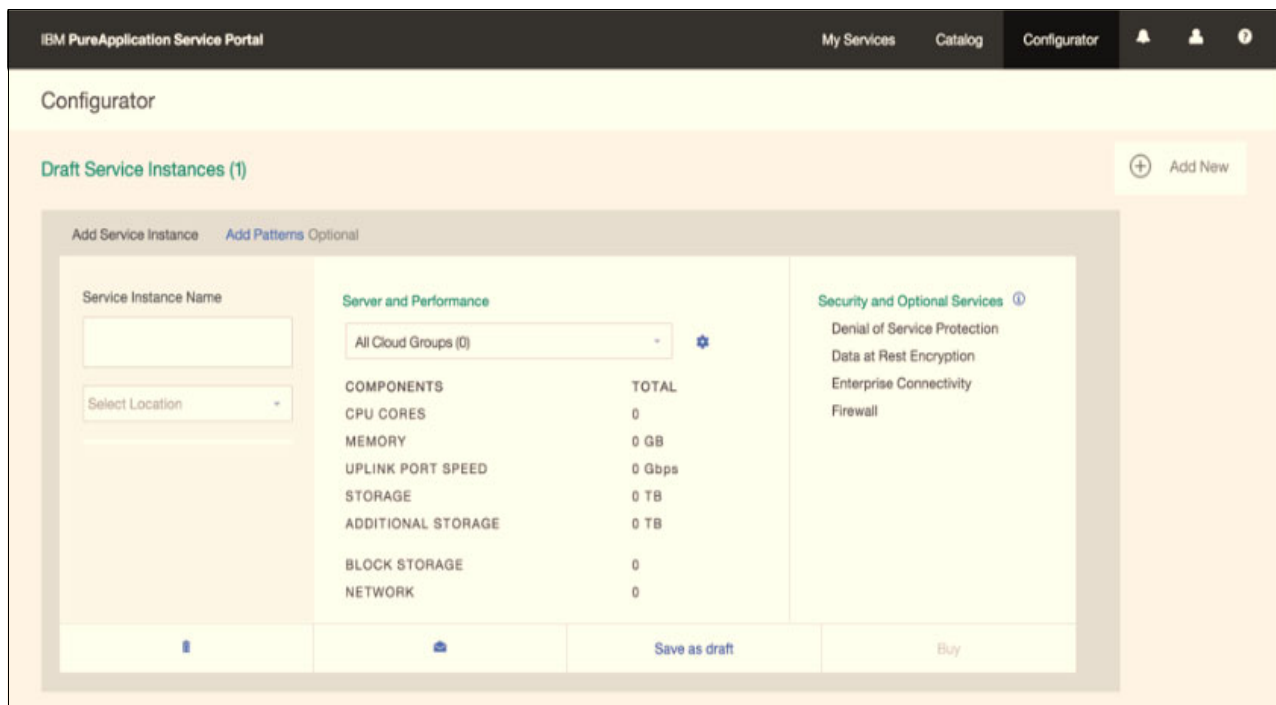


Figure 1-6 Creation of a new instance on PureApplication Service

Figure 1-6 shows a blank draft service instance. Follow these steps to configure the instance:

1. Provide a name to the instance and select the data center location that you need, choosing between 18 data centers in 12 countries around the world. (At the time of this writing, 40 locations exist around the world, with new data centers that are added frequently.)
2. Add cloud groups.
3. Configure security options and network connectivity.

Adding the cloud group to the instance adds resources to the environment:

- ▶ The name of the cloud group
- ▶ The server type and the quantity, choosing between five configurations, including the eSeries 4-core server with 32 GB of RAM, the mSeries 8-core server with either 128 GB or 256 GB of RAM, and the sSeries 16-core server with either 256 GB or 512 GB of RAM
- ▶ Additional storage in 1 TB increments
- ▶ Optional additional block storage, in increments of 100 GB, 250 GB, 500 GB, or 1 TB

Figure 1-7 shows the basic specifications for a cloud group.

The screenshot shows the 'Add Cloud Group' form. It has a 'Specifications' section with fields for 'Cloud Group Name' (set to 'Cloud Group 1'), 'Server Type' (set to 'S-Series 256GB'), 'Number' (set to '1'), and 'Additional Storage' (set to '1 TB'). To the right is a 'COMPONENTS' table with two columns: 'COMPONENTS' and 'TOTAL'. The table lists: CPU Cores (16 Cores), Memory (256 GB), Uplink Port Speed (2 Gbps), Storage (1 TB), and Additional Storage (1 TB). Below the specifications is a 'Block Storage' section with a value of '0'. At the bottom right are 'Cancel' and 'Add' buttons.

COMPONENTS	TOTAL
CPU Cores	16 Cores
Memory	256 GB
Uplink Port Speed	2 Gbps
Storage	1 TB
Additional Storage	1 TB

Figure 1-7 Adding a cloud group to the instance

Figure 1-8 shows how to add individual patterns to the instance that you want to preload on your new instance after it is provisioned. The choice of patterns is limited to the set of patterns for which you have an active entitlement (bring your own license). Additional patterns can be purchased directly in the portal. Adding patterns to your environment takes only one click from the self-service portal.

The screenshot shows the 'Add Patterns' optional section. It has a 'Patterns (11)' list with a filter box. The list includes: Business Process Manager Pattern 8.5.6, Integration Bus HV Edition 10.0.0.0, Business Monitor Pattern 8.5.5, DB2 with BLU 1.2.1.0, Mobile Application Platform Pattern 7.0.0.0, and Operational Decision Management Pattern 8.5.1.1. To the right is a 'Pattern Details' section for the 'Business Process Manager Pattern 8.5.6 VSYS'. It describes the IBM Business Process Manager Pattern and its components. To the right of the details is a 'LICENSE TYPE' section with 'Processor Value Unit', a 'PRODUCT ID' of '5725-L40', and a 'PART NUMBER' of 'D0ZMQLL - BF'. At the bottom are icons for a trash can and an envelope, and buttons for 'Save as draft' and 'Buy'.

Figure 1-8 Adding patterns to the instance

Finally, you can save the instance and keep it in draft state until you are ready to start the provisioning. You can also send an email with the instance details to anyone you choose. As Figure 1-9 shows, after your draft instance is ready, click **Save as draft**, then click **Buy** to start the provisioning.

Add Service Instance

Add Patterns

Optional

Service Instance Name

Demo Instance

SJC01 - San Jose - West Coas

Server and Performance

All Cloud Groups (1)

COMPONENTS

CPU CORES

MEMORY

UPLINK PORT SPEED

STORAGE

ADDITIONAL STORAGE

BLOCK STORAGE

NETWORK

TOTAL

16 Cores

256 GB

2 Gbps

1 TB

1 TB

0

0

Security and Optional Services

Denial of Service Protection

Data at Rest Encryption

Enterprise Connectivity

Firewall

Save as draft

Buy

Figure 1-9 Submit the provisioning of your instance

The catalog of patterns (Figure 1-10) shows samples of available patterns, but the actual list is a real-time view of the patterns and versions that are available for immediate upload to your PureApplication Service instances.

Patterns	Version(s)	Product ID	License Type		
Business Process Manager Pattern	8.5.6	5725-L40	Processor Value Unit		★
Integration Bus HV Edition	10.0.0.0	5725-B72	Processor Value Unit		★
Business Monitor Pattern	8.5.5	5724-M24	Processor Value Unit		★
DB2 with BLU	1.2.1.0	5765-F41	Processor Value Unit		★
Mobile Application Platform Pattern	7.0.0.0	5725-G24	Processor Value Unit		★
Operational Decision Management Pattern	8.5.1.1	5725-I09	Processor Value Unit		★
WebSphere Application Server Pattern	1.0.0.3	5725-A26	Processor Value Unit		★

Figure 1-10 Catalog of patterns

Table 1-2 shows the available infrastructure options on PureApplication Service.

Table 1-2 Available infrastructure options on PureApplication Service

	eSeries (e32)	mSeries (m128)	sSeries (s256)
Processor	Single processor quad core Xeon 3450	Single processor octo core Xeon 2670	Single processor octo core Xeon 2670
Processor speed	2.66 GHz	2.60 GHz	2.60 GHz
Processor cache	8 MB	20 MB	20 MB
Cores	4 cores	8 cores	16 cores
Processor value units (PVUs)	280 PVUs (4 x 70 PVU)	560 PVUs (8 x 70 PVU)	1120 PVUs (16 x 70 PVU)
Memory type	DDR3 registered 1333	DDR3 registered 1333	DDR3 registered 1333
Memory amount	32 GB	128 GB	256 GB
Public bandwidth	Unlimited Bandwidth	Unlimited Bandwidth	Unlimited Bandwidth
Uplink port speed	1 Gbps Private 1 Gbps Public	2 Gbps Private 2 Gbps Public	2 Gbps Private 2 Gbps Public
Storage type	HDD storage area network (SAN) (Internet Small Computer System Interface (iSCSI))	HDD SAN (iSCSI)	HDD SAN (iSCSI)
Storage amount	1 TB	1 TB	1 TB
Storage redundancy	RAID 50	RAID 50	RAID 50

For more information about the IBM PureApplication self-service portal, see the video at this website:

<https://www.youtube.com/watch?v=oYYu33ybybw>

1.4.3 IBM PureApplication Software

IBM PureApplication Software is the most recent member of the PureApplication family. It is a cloud application platform that, like PureApplication System, involves on-premises solutions. It is based on the philosophy of *bring your own hardware*. The focus of PureApplication Software is to build up a cloud platform when you use the existing infrastructure that is available in the data center. The *bring your own hardware* approach is preferable in the following cases:

- ▶ Significant investment exists in the on-premises infrastructure (saving capital expenditures (CAPEX)), in terms of hardware and software. For example, your business uses many software licenses of Red Hat Enterprise Linux (RHEL) or VMware.
- ▶ Significant investments exist in local system management staff, services, and tools (saving operational expenditures (OPEX)), especially on VMware virtualization and infrastructure management.
- ▶ Important use cases require unique infrastructure configurations.

With PureApplication Software, you can deploy PureApplication workloads and lifecycle management capabilities, but on a local infrastructure that is based on VMware or PowerVC. From a technological point of view, this solution adds additional flexibility:

- ▶ Simplification of high availability scenarios
- ▶ Integration with existing external storage
- ▶ More granular access control for cloud and hardware resources

PureApplication Software V2.1.1 can run on your Intel, IBM POWER7, or IBM POWER8 servers. It can also be used on IBM PureApplication System W2500, W2700, W1500, and W1700 appliances. PureApplication Software can also be used on the Azure cloud.

The latest enhancements in Release 2.1.1 are listed:

- ▶ Pattern support to include Docker containers
- ▶ The integration between the PureApplication Software offering and existing Chef servers to use Chef recipes

For more information about the PureApplication Software capabilities, see the announcement letter:

<https://ibm.biz/BdHEbN>

Note: Instances from PureApplication Software can be added to a management domain, but they cannot be part of a deployment subdomain that involves IBM PureApplication Systems.

1.4.4 Where do the three PureApplication products meet

The three PureApplication products are built by using the same core technology. Patterns and the pattern engine are not only at the core of the three solutions, but also the primary way to achieve application portability in a hybrid cloud scenario. Use the patterns to design one time. Then, deploy the applications everywhere, from PureApplication System to PureApplication Software or Service.

In addition to the value of the patterns, all IBM PureApplication products implement the concept of shared services, for example, a shared monitoring or caching service. These services can be used by all virtual machines (VMs) that are running in one environment without a requirement for the pattern designer to provide installation and configuration scripts to integrate the services. When the VM is created, the available shared services are detected and integration is performed.

Each PureApplication family member is complementary and addresses cloud capabilities differently. Table 1-3 offers a comparison of how the PureApplication capabilities match the essential characteristics of a cloud solution.

Table 1-3 Where the IBM PureApplication family products meet

Cloud computing capability	PureApplication System	PureApplication Software	PureApplication Service
On-demand self-service	<p>PureApplication System, Software, and Service provide a web management portal that can implement “on-demand self-service” in a private cloud.</p> <p>PureApplication Service also provides an online, self-service portal to purchase service, which is also known as <i>instances</i>, on a public cloud.</p>		
Broad network access	<p>PureApplication System and PureApplication Software are on-premises cloud platforms. They support the “network access everywhere” by relying on the on-premises infrastructure capabilities.</p>		<p>PureApplication Service is a public cloud service on the internet and it relies on the network capabilities of SoftLayer.</p>
Resource pooling	<p>You can segment PureApplication System resources (compute, network, and storage) into different, isolated groups.</p>		<p>In addition to the environments that PureApplication System and Software provide, PureApplication Service provides a dedicated, isolated environment to the cloud user.</p>
Rapid elasticity	<p>Virtual patterns natively support the elastic scaling of the resources.</p>		
Measured service	<p>PureApplication natively provides tools to monitor and control the usage of the resources.</p>		
Infrastructure as a service (IaaS)	<p>The ability to deploy multiple VMs is part of both a virtual system pattern and a virtual appliance pattern.</p>		
Platform as a service (PaaS)	<p>The virtual application pattern expresses the ability to support PaaS by the PureApplication. The consumer does not manage or control the underlying cloud infrastructure, including network, servers, or storage, but can control the deployed applications and possibly configuration settings for the application-hosting environment. PureApplication Systems plays a key role in this space by providing a preconfigured, open platform for PaaS solutions.</p>		
Software as a service (SaaS)	<p>PureApplication can be used as the building blocks for designing and implementing a cloud computing solution that can provide SaaS offerings.</p>		
Private cloud	<p>PureApplication System and Software represent key building blocks for implementing a private cloud solution.</p>		<p>PureApplication Service is only an off-premises cloud solution and relies on IBM SoftLayer.</p>
Public cloud	<p>Although PureApplication System and PureApplication Software are IBM offerings for on-premises solutions, they act as the basic building blocks from the perspective of a cloud provider that needs to implement a public cloud solution.</p>		<p>PureApplication Service relies on IBM SoftLayer to provide a public cloud solution.</p>
Hybrid cloud	<p>PureApplication System and PureApplication Software are key building blocks for implementing the on-premises side of a hybrid cloud computing scenario.</p>		<p>PureApplication Service is a key solution for the off-premises side of a hybrid cloud computing scenario.</p>

IBM PureApplication family, IBM Bluemix, and IBM Cloud Orchestrator

This section describes how the PureApplication family intersects with other IBM products for a holistic approach to cloud computing and a focus on the hybrid scenario.

Figure 1-11 provides a view of the positioning of the PureApplication family, IBM Bluemix, and IBM Cloud Orchestrator on two dimensions: cloud computing service models and cloud computing deployment models.

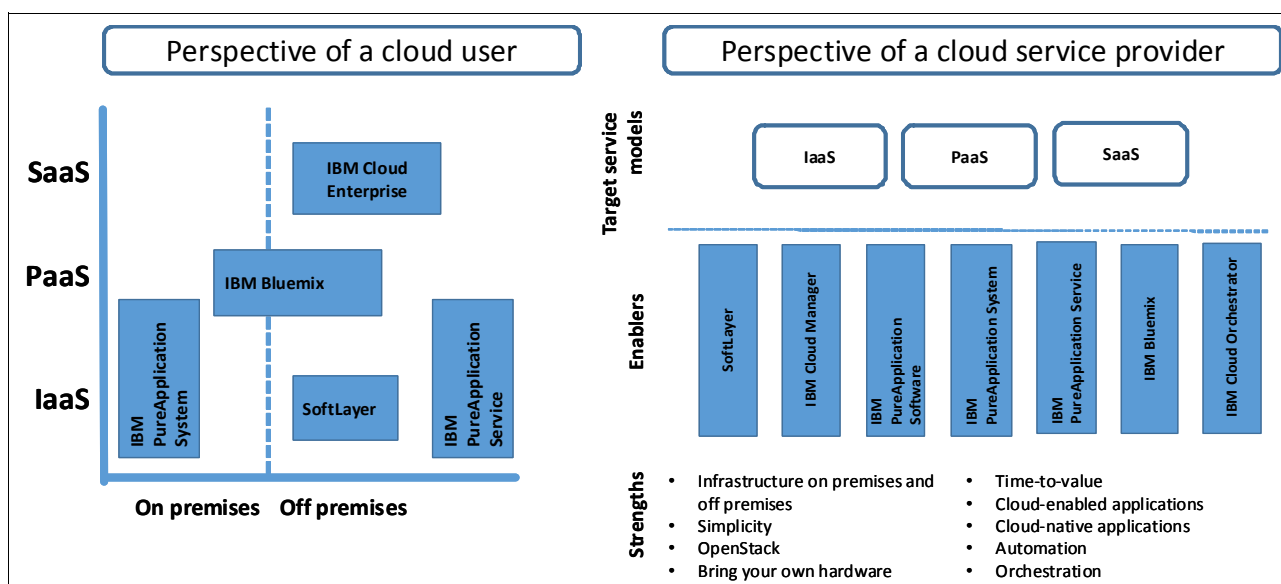


Figure 1-11 A quick positioning guide of the IBM products to approach cloud computing

IBM Cloud Orchestrator (ICO) provides cloud management capabilities to address complex scenarios, for example:

- ▶ An orchestration layer that offers workflow and approval logic is a requirement.
- ▶ The infrastructure is heterogeneous (OpenStack platforms, Intel, IBM Power, IBM System z®, and so on).

IBM Bluemix is a PaaS that can speed up the delivery of cloud-native applications. Bluemix is available as either off premises or “on-premises off premises”:

- ▶ Off premises, either running in a multiple tenant or dedicated environment that is maintained by IBM
- ▶ On-premises off premises, which is also called *Bluemix Local*, either running on dedicated, client-provided, and client-managed hardware, or within IBM PureApplication System

Note: IBM PureApplication System provides a means for running a local instance of IBM Bluemix. At the time of this writing, this method is the *only way* to run a Bluemix environment on top of hardware that is managed by IBM. For this scenario, Bluemix Local requires at least two dedicated compute nodes and storage.

The software on-premises as a service (SaaS) model provides the cloud user the ability to deploy an application that is running on a cloud (private, public, or hybrid). In this case, the cloud user is not responsible for the underlying cloud infrastructure, in terms of hardware, software, licensing, and system management. Beyond consuming SaaS applications, the cloud user might only be able to configure at the application level. From the perspective of the cloud user, IBM provides a public cloud offering with more than 100 SaaS business applications. For more information, see *Reinvent your business with ready-to-use expertise* at this website:

<http://www.ibm.com/cloud-computing/us/en/saas.html>

However, from the perspective of a cloud provider that wants to offer SaaS business applications, both IBM Cloud Orchestrator and the IBM PureApplication products are important building blocks and enable the implementation of the cloud solution (private, public, or hybrid). The same approach is applicable for PaaS, where IBM Cloud Orchestrator, IBM PureApplication System, and IBM PureApplication Software are potential building blocks and enable the implementation of a cloud solution from the perspective of a cloud service provider.

In summary, Figure 1-12 illustrates an IBM perspective for cloud-enabled applications in a hybrid cloud computing environment that uses IBM PureApplication products.

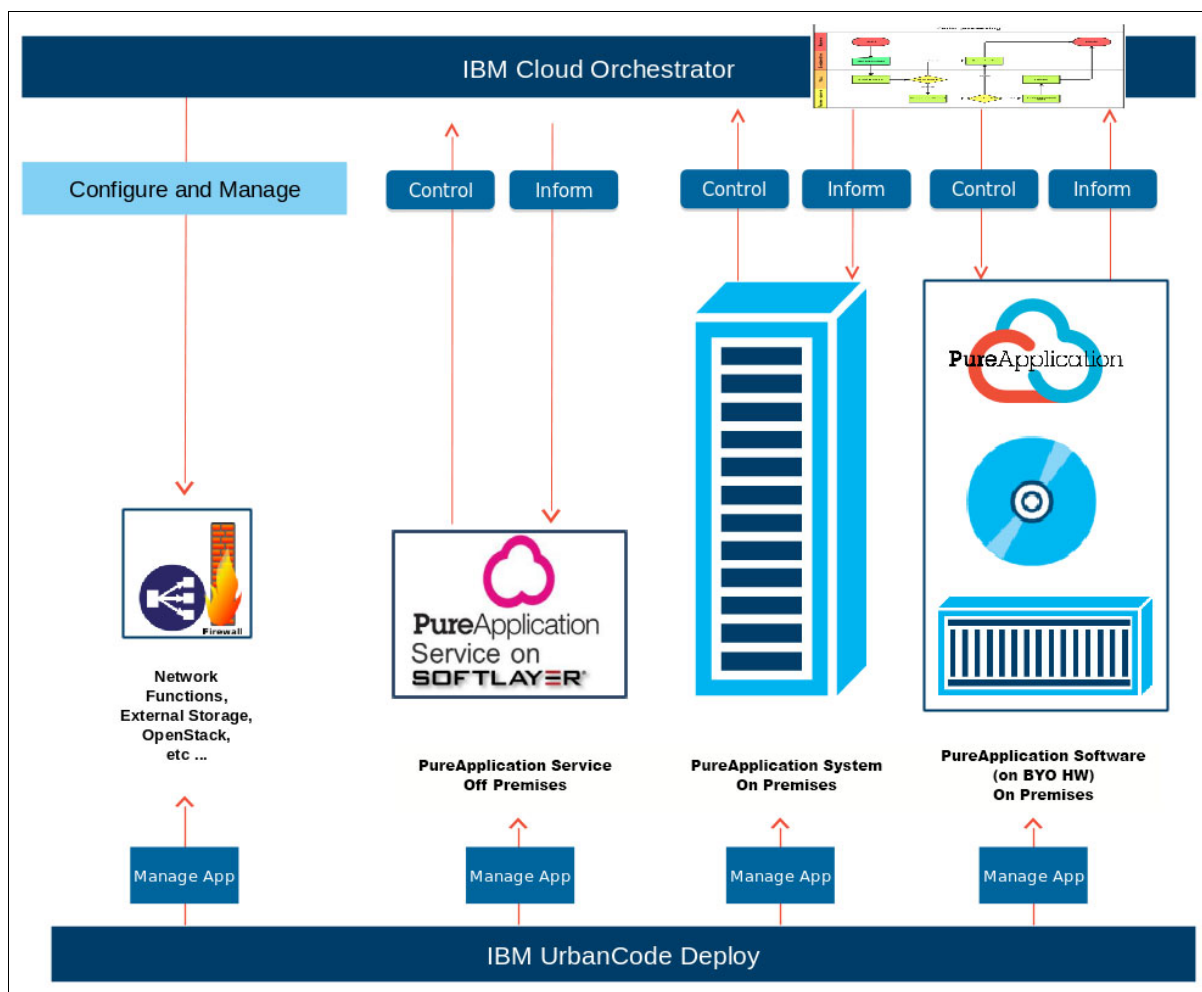


Figure 1-12 Cloud-enabled applications in a hybrid cloud

Building a hybrid cloud requires the consideration of a mixed environment that contains IBM PureApplication Service for the off-premises part, which is combined with either IBM PureApplication Software or IBM PureApplication System for the on-premises portion. The choice between IBM PureApplication Software or IBM PureApplication System depends on personal preference. You can choose either option:

- ▶ Provide and maintain the hardware yourself by using IBM PureApplication Software.
- ▶ Take the integrated hardware, software, and middleware option, and select IBM PureApplication System.

In this publication, IBM PureApplication System and IBM PureApplication Service were combined to build the hybrid cloud solution in the use cases that are described in Chapter 3, “Hybrid use cases” on page 105.

For more information, see the following resources:

- ▶ For more information about the IBM PureApplication product family, see this website:
http://www.ibm.com/ibm/puresystems/uk/en/pf_pureapplication.html
- ▶ For more information about IBM Cloud Orchestrator, see this website:
<https://ibm.biz/BdEZu3>
- ▶ For more information about IBM Bluemix, see *Create, deploy, and manage your apps in the cloud* at this website:
<http://www.ibm.com/cloud-computing/bluemix/index-b.html>
- ▶ For more information about IBM Bluemix on local, see *Expect more from private cloud* at this website:
<http://www.ibm.com/cloud-computing/bluemix/hybrid/local/>



How to build a hybrid cloud

This chapter introduces the concepts for building a hybrid cloud with IBM PureApplication System, IBM PureApplication Software, and IBM PureApplication Service, which are known collectively as the *IBM PureApplication family offering*.

In this chapter, we describe how the IBM PureApplication products can accelerate the adoption of a hybrid cloud, with considerations for infrastructure, security, and networking. We also further cover considerations about infrastructure, middleware, and application programming interfaces (APIs) for hybrid portability. We also include information about interactions between the PureApplication family and open source product offerings and how to provide secure business continuity adoption in hybrid scenarios.

The following topics are the focus of this chapter:

- ▶ A five-step roadmap for establishing a hybrid cloud
- ▶ A hybrid infrastructure with the PureApplication family
- ▶ Patterns and the PureApplication family
- ▶ Open technologies and PureApplication integration
- ▶ Achieving hybrid cloud application portability
- ▶ Deploying applications by using a hybrid cloud
- ▶ Achieving business continuity and high availability in a hybrid cloud

2.1 A five-step roadmap for establishing a hybrid cloud

Every trip requires a good plan before departure if you want to avoid disappointments. The adoption of a hybrid cloud solution is similar to taking a trip and reaching the final destination. With a good strategy, you can maximize the value of your investment. This plan applies to establishing a hybrid cloud also because significant changes will be introduced to your enterprise in the following areas:

- ▶ Infrastructure
- ▶ Security
- ▶ Application development
- ▶ System management

Every change triggers a transformation, and every transformation needs a roadmap.

The purpose of this book is not to fully describe a cloud adoption strategy. However, it is important to focus on important aspects of hybrid cloud adoption.

Figure 2-1 shows a five-step iterative adoption roadmap, and the following text lists the characteristics of each step, providing high-level information about how the IBM PureApplication family offering can meet your requirements.

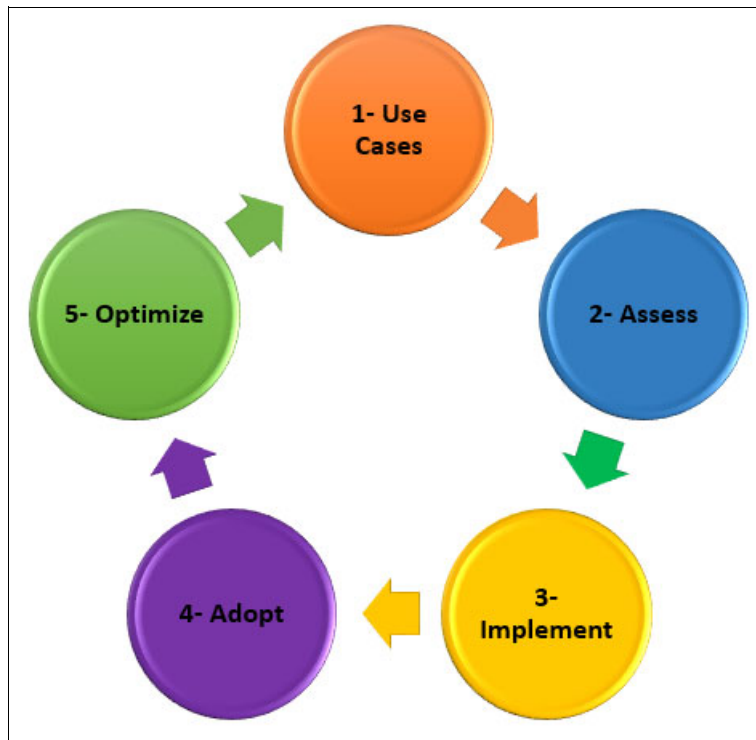


Figure 2-1 A five-step roadmap for establishing a hybrid cloud

2.1.1 Use cases

A successful roadmap requires a clear goal. In this publication, the use cases represent the goals to achieve from a business perspective. The hybrid cloud is the key enabler for implementing the use cases.

Figure 2-2 introduces the business value of several common use cases:

- ▶ Take advantage of the flexibility that is offered by cloud computing to extend your enterprise applications by adopting a hybrid approach in which the systems of engagement integrate with the systems of records
- ▶ Experiment with new technologies by relying on the agility and speed of acquiring resources on a cloud
- ▶ Achieve cost optimization by moving development and test environments to the public cloud
- ▶ Achieve market expansion and globalization by using the global availability of cloud service provider data centers

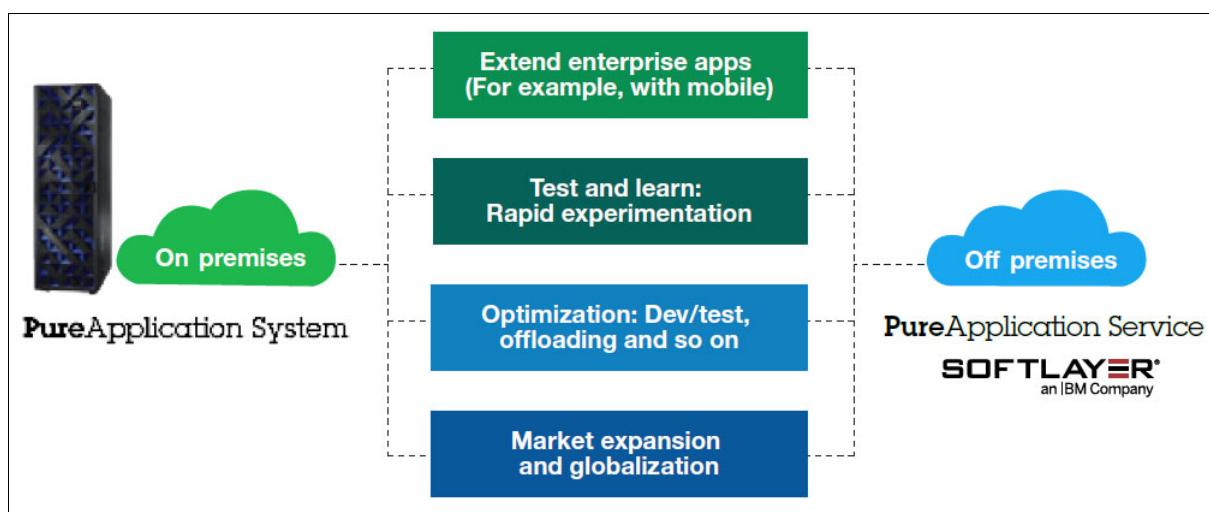


Figure 2-2 Common hybrid cloud use cases

In this book, Chapter 3, “Hybrid use cases” on page 105 describes the business value and technical impacts of a set of specific use cases:

- ▶ Develop, test, and run a pilot on a public cloud, then move the application on premises.
- ▶ Develop front-end applications off premises and back-end applications on premises.
- ▶ Deliver an application for a marketing campaign quickly.
- ▶ Start a small or medium business on a public cloud, then invest in on-premises applications as the business grows.
- ▶ Start your business functions on premises. Move completely to the cloud as part of a corporate strategy.
- ▶ Maintain primary functions on premises and disaster recovery (DR) functions off premises.

2.1.2 Assessment

With your defined goals, the next step is to understand what you have and what you need to achieve your goals (use cases) by initiating a gap analysis workshop. The assessment phase of your analysis covers at least three aspects:

- ▶ Assess the applications that are viable for moving to the hybrid cloud
- ▶ Determine your on-premises and off-premises needs and requirements
- ▶ Decide your system management approach for the hybrid cloud

Assess your applications

The implementation of use cases requires the design, development, and deployment of business applications. However, the development of new cloud-native applications is not always necessary.

An existing application (or part of an application) can become cloud-enabled, and the suitability of this approach depends on many factors, such as your security, availability, accessibility, resiliency, and integration requirements. Therefore, the assessment of existing applications is intended to provide their suitability for moving to the cloud. Applications might not fit the cloud requirements for technological or business reasons.

The dilemma can be addressed by first determining how to address the following questions:

- ▶ Do we develop new cloud-native applications?
- ▶ Do we redesign our existing applications? If so, do we redesign these applications for an on-premises cloud only (for example, taking a WebSphere Application Server application, such as a stock-trading application that runs on bare metal, and moving it to PureApplication System to realize the benefits of self-provisioning, virtualization, and metering)?
- ▶ Alternatively, assuming that we determine that the application can run on the on-premises cloud, can we then realize any benefits from running it on the hybrid cloud, or moving it entirely to the public cloud?

By using an assessment that identifies the applications (cloud-native and cloud-enabled) that implement one or more hybrid cloud use cases, you can develop a project plan. Because business needs drive everything, the plan must consider how critical each application is and how much business value each application generates.

Table 2-1 describes a way to classify your applications in terms of strategic value and critical issues.

Table 2-1 Classification tiers of applications

	Tier 1 applications	Tier 2 applications	Tier 3 applications	Tier 4 applications
Type of application	Most critical	Mission critical	Strategic but not mission critical	Neither strategic or mission critical
Characteristics	<ul style="list-style-type: none">▶ Maximum level of resilience▶ Monitoring at the application level▶ 24x7 support	<ul style="list-style-type: none">▶ High level of resilience▶ Monitoring at the infrastructure level▶ 24x7 support	<ul style="list-style-type: none">▶ Medium level of resilience▶ Basic monitoring▶ Support (business hours)	<ul style="list-style-type: none">▶ Low level of resilience▶ Basic or no monitoring▶ No support or best effort support

Start from the classifications in Table 2-1 on page 28, proceed to Figure 2-3, which provides a larger picture of a hypothetical set of applications, showing the priority of each from the business and technical perspectives.

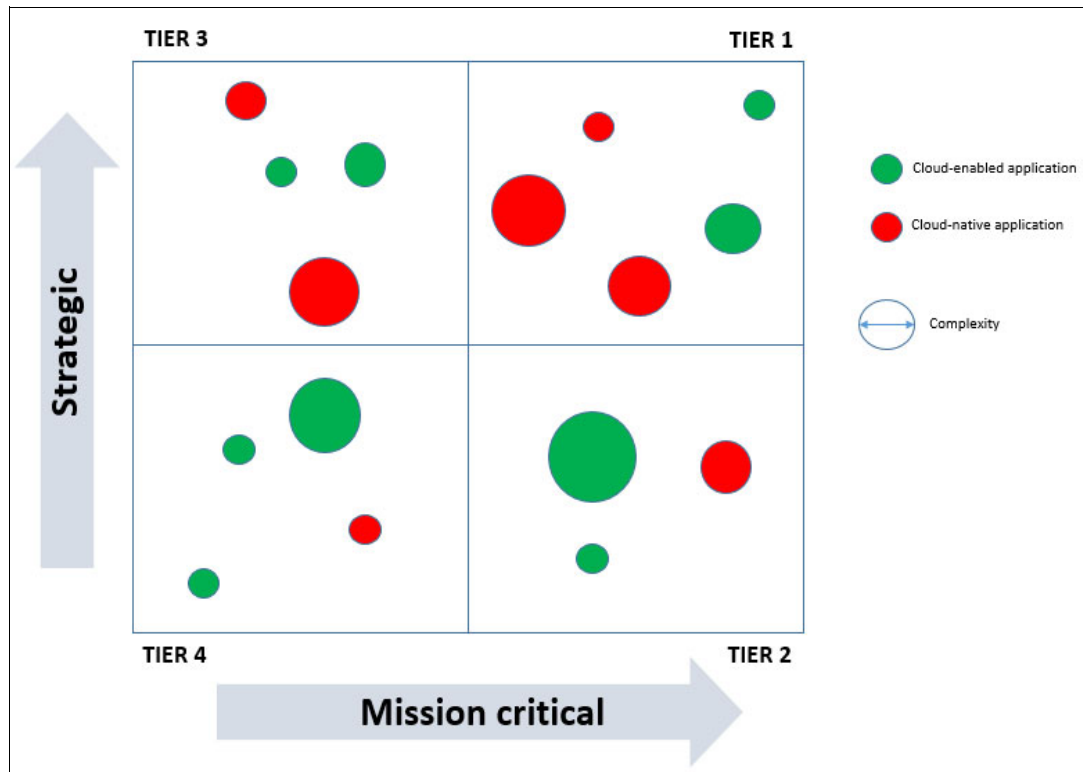


Figure 2-3 Planning your adoption of the hybrid cloud

Most likely, starting with your Tier 4 applications is a safe starting point, if you are working on your first implementation of a hybrid cloud scenario. The risks are low, and you can easily mitigate unforeseen situations. However, every enterprise must build its own specific plan, depending on their business drivers and technical context.

Assess your infrastructure

In a hybrid cloud scenario, the cloud service provider extends your on-premises infrastructure. The size and the nature of the extension depend on the use cases and likely are not constant over time. Moreover, the impact of the hybrid cloud adoption also depends on the cloud service model.

Figure 2-4 shows the differences among between the three cloud service models in terms of responsibilities between the cloud service provider and cloud user. The figure presumes that a public cloud is rented and that an on-premises PureApplication System cloud is used. The area that is labeled *Vendor Manages in Cloud* is the enterprise IT shop that manages its own cloud, and the shop might or might not charge back the line of business (LOB) for usage.

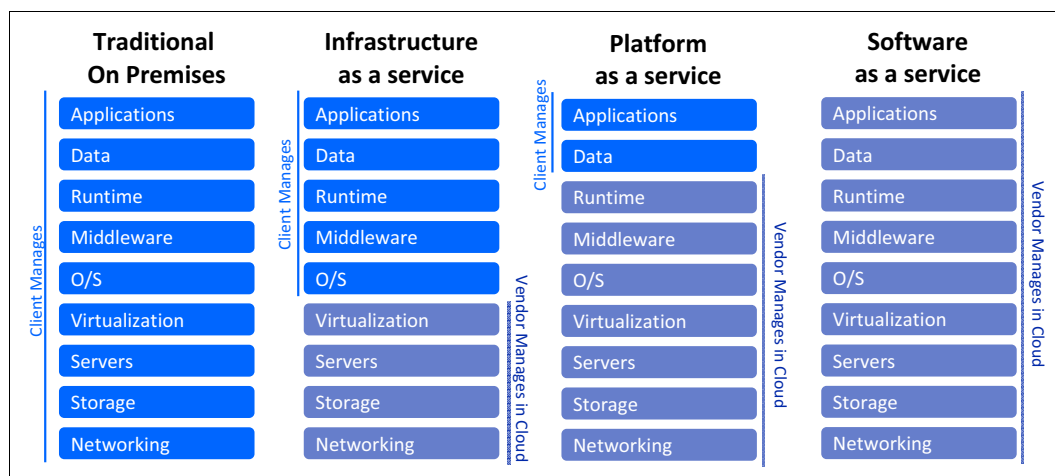


Figure 2-4 Differences among the cloud service models

Each service model affects the everyday activities of the local IT operations team differently and requires different considerations in the choice of the cloud service provider. For instance, if you are choosing a cloud service provider for an infrastructure as a service (IaaS) model, consider the following questions:

- ▶ Does the cloud service provider offer the same operating systems (OSs) that we use on premises?
- ▶ Can we use our system management skills and tools to handle the off-premises infrastructure?
- ▶ Does the cloud service provider offer additional capabilities, such as backup and restore and auto-patching?
- ▶ Can the cloud service provider use bare metal?

The adoption of a platform as a service (PaaS) model requires considerations about the platforms and programming languages that are supported by the cloud service provider. A hybrid cloud scenario based on PureApplication System and PureApplication Service provides many benefits:

- ▶ Highly virtualized, standardized, and homogeneous platforms on both sides: on premises and off premises
- ▶ The possibility of centralized system management across the hybrid cloud:
 - Aligning the catalogs between PureApplication System and PureApplication Service
 - Configuring PureApplication Service instances to use the same Lightweight Directory Access Protocol (LDAP) as the PureApplication Systems on premises
- ▶ Fix management on PureApplication System and PureApplication Service
- ▶ Backup and restore on PureApplication Service (For more information, see 4.4.1, “Backup and recovery in a hybrid cloud” on page 176.)

2.1.3 Implementation

The implementation phase covers several aspects:

- ▶ What is the impact on the software delivery pipeline?
- ▶ How to establish a secure hybrid scenario?
- ▶ What is the impact on networking?
- ▶ What is the impact on my everyday activities?

Software delivery pipeline in a hybrid scenario

The effectiveness of (or the survivability) an enterprise depends on its ability to rapidly transform its business and IT models when necessary. IT departments need to quickly execute their activities to support the continuous business evolution. The efficiency of the software delivery pipeline demonstrates the ability of IT departments to respond to innovation swiftly. Many enterprises adopt a DevOps approach, and we can summarize the goal of DevOps. You can use DevOps to deploy an application to production (or other environment) at any time.

The question then becomes, “How does the hybrid cloud impact my software delivery pipeline?”

Even in a hybrid cloud that is based on PureApplication System and PureApplication Service, you can use the capabilities of UrbanCode software to meet the DevOps requirement to provide continuous delivery. One of the strengths of an IBM solution is that the tools work across both private and public clouds in the same manner. Figure 2-5 shows the automation of platform provisioning in traditional IT models and in cloud service models.

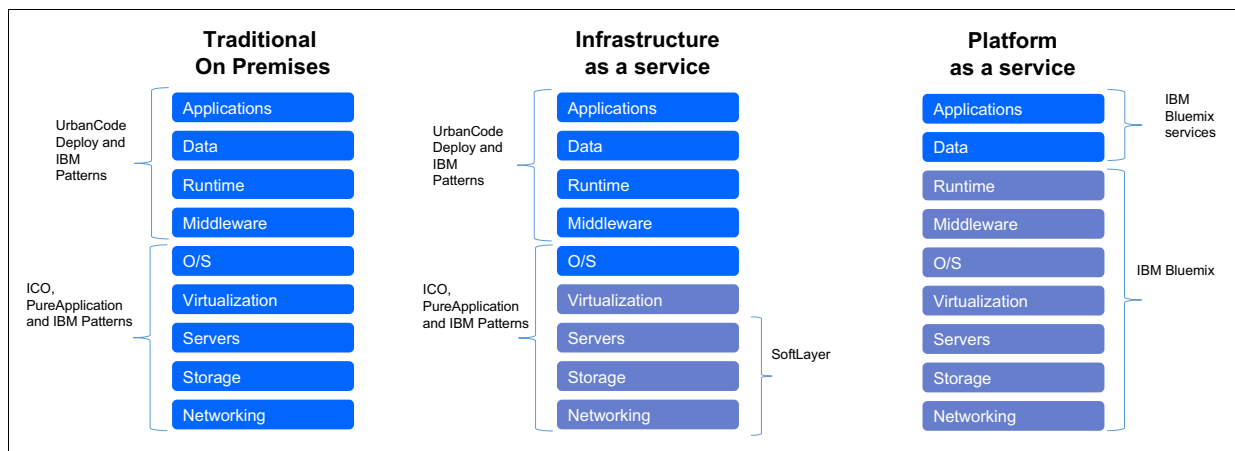


Figure 2-5 Automating platform provisioning in traditional IT models and in cloud service models

To read more about how the PureApplication family addresses DevOps challenges, see the following websites:

- ▶ *IBM PureApplication System V2.0 Integration with IBM UrbanCode Deploy using patterns:*
<https://www.youtube.com/watch?v=kfvCTGD0dI4>
- ▶ *Addressing DevOps Challenges with IBM PureApplication System:*
<https://www.youtube.com/watch?v=g3U6muurqBQ>
- ▶ *Integration of Rational Team Concert (RTC), IBM UrbanCode Deploy & IBM PureApplication System:*
<https://www.youtube.com/watch?v=0thw0r744aw>

To read more about DevOps in a hybrid cloud, see *I need to build and deploy complex hybrid applications that include cloud, mobile, and on-premises components*:

<https://www.ibm.com/cloud-computing/solutions/cloud-devops/hybrid#fbid=WimCeUibOn4>

Security considerations

Security is an ever-challenging issue in IT. A discussion about even the most significant security implications in the cloud requires an entire book. Whether your enterprise is in the cloud or not, IT managers must have a security strategy. The hybrid cloud significantly changes these considerations. For example, the adoption of a public cloud means that you do not have complete control over security. Instead, you must share this responsibility with your cloud service provider.

This book emphasizes the main security issues that an IT manager must consider when an IT manager implements a hybrid cloud. Figure 2-6 illustrates these security issues, which are grouped by major area.

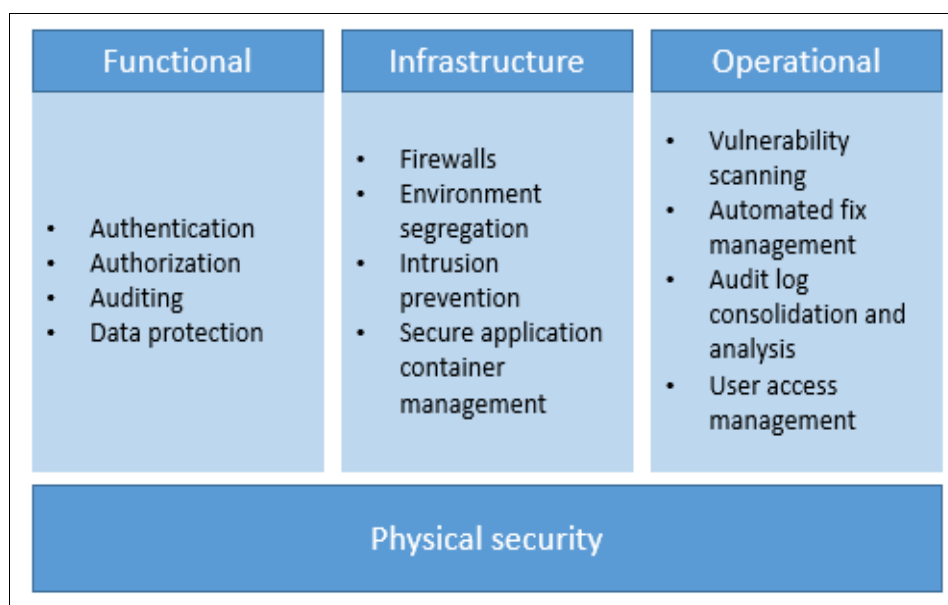


Figure 2-6 Main security considerations on the hybrid cloud

Authentication, authorization, and auditing

Different roles (for example, administrators and developers) access functions and data on a hybrid cloud by using different permissions. On the private side, the authentication, authorization, and auditing usually rely on the well-known on-premises security framework and its policies. Authentication and authorization are implemented on top of a user registry, in which you can create users and groups. Generally, an LDAP implements the user registry.

PureApplication System provides the capability to configure an external LDAP as the user registry, and it provides a deep granularity, in terms of levels of permissions that can be assigned to a group. This approach enables each role to perform one or more specific functions. For auditing, PureApplication System captures comprehensive auditing records, and every record describes who attempted what action on which resources when, from where, and whether the result was successful. Security auditing is enabled by default, and it cannot be disabled.

On the public side of the hybrid cloud, the cloud service provider must provide the authentication, authorization, and audit services. This situation is more delicate because in a multitenant scenario, the cloud service provider must ensure that each cloud user can see and access only its own services.

The PureApplication Service provides dedicated environments to cloud users. It also provides the same capabilities as PureApplication System in terms of authorization, authentication, and audit capabilities. Moreover, it is possible to configure an LDAP master-master topology for both PureApplication System and PureApplication Service. Therefore, you can centralize the user registry for the authentication and authorization services of the management graphical user interface (GUI).

For more information about the security on PureApplication System, see the following resources:

- *Administering users, user groups, and security* (PureApplication System W3700):
<https://ibm.biz/BdHhie>
- *Administering users, user groups, and security* (PureApplication System W2700):
<https://ibm.biz/BdHhib>

For more information about the security on PureApplication Service, see *Administering users, user groups, and security* (PureApplication Service 2.1.1) at this website:

<https://ibm.biz/BdHhip>

Protect your data

How to protect your data in the cloud is probably one of the most common concerns of IT managers. It is necessary to protect the data everywhere and for the whole data lifecycle. In a hybrid scenario, the data can be on premises, off premises, or in transit, and the data lifecycle includes several states, as Figure 2-7 shows.

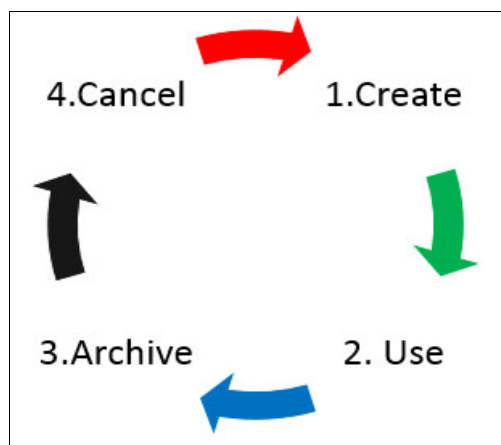


Figure 2-7 Data lifecycle

Table 2-2 describes security considerations for each state of the data lifecycle.

Table 2-2 Data lifecycle and security considerations for each state of the data lifecycle

Phase	Security considerations
Create	<p>Whether you are creating, updating, or deleting your data on premises or off premises, you need to protect the access to the data by applying security services. Authentication, authorization, and auditing are required when the applications access the data and when an administrator accesses the data.</p> <p>On a private cloud, in which all data is contained by the organization, a simple method exists to maintain security compliance. In contrast, on a public cloud, at least three security concerns exist:</p> <ul style="list-style-type: none"> ▶ Where is my data physically located? ▶ How is my data isolated from the data of other cloud users, in a public, multitenant environment? ▶ How can I move my data from off premises to on premises, and vice versa?
Use	Applications and administrators continuously access the data. In certain cases, data is sensitive and needs encryption. Encryption can be applied when the data is in transit or when the data is on premises and off premises.
Archive	Understand the backup and restore strategy of your off-premises data. Also, understand how you can keep a copy of the off-premises data in the on-premises data center.
Cancel	If you terminate your contract with the cloud service provider, is your data canceled? Alternatively, if you cancel several virtual machines (VMs), is your data canceled?

The following benefits relate to a hybrid cloud that is based on PureApplication System and PureApplication Service:

- ▶ The ability to easily establish a secure connection between on premises and off premises. For more information, see Chapter 4, “Implementation considerations” on page 139.
- ▶ PureApplication Service offers dedicated environments, and the data is isolated from the data of other tenants.
- ▶ Table 2-3 shows how PureApplication Service is globally available in SoftLayer data centers.
- ▶ For an explanation of the backup and restore capabilities of PureApplication System and PureApplication Software in a hybrid scenario, see 4.4.1, “Backup and recovery in a hybrid cloud” on page 176.
- ▶ SoftLayer adopts the Department of Defense (DoD) 5220.22-m standards to erase server storage when physical or virtual servers are no longer needed.

Table 2-3 Global availability of PureApplication Service on SoftLayer data centers

Region	Country	City
North America	Canada	Montreal
	Canada	Toronto
	Mexico	Mexico City
	US	Dallas
	US	Dallas
	US	Houston

Region	Country	City
	US	San Jose, CA
	US	Washington, DC
	US	Washington, DC
Europe	France	Paris
	Germany	Frankfurt
	Italy	Milan
	Netherlands	Amsterdam
	United Kingdom	London
Latin America	Brazil	Sao Paulo
Asia	Australia	Melbourne
	Australia	Sydney
	Hong Kong SAR of the PRC	Hong Kong SAR of the PRC
	Japan	Tokyo
	Singapore	Singapore

Note: Table 2-3 on page 34 shows the available PureApplication Service data centers as of the writing of this book (Fall, 2015).

Infrastructure and operational security

From the infrastructure and operations perspectives, Table 2-4 describes the main security considerations.

Table 2-4 Main considerations for infrastructure and operational security

	Security considerations
Infrastructure security	<ul style="list-style-type: none"> ▶ Environment segregation ▶ Firewalls ▶ Intrusion protection ▶ Operating system security hardening
Operational security	<ul style="list-style-type: none"> ▶ Vulnerability scan ▶ Automated fix management ▶ Audit log consolidation and analysis ▶ User access management

A hybrid cloud perspective that is based on PureApplication System and PureApplication Service offers the following benefits:

- ▶ For information about environment segregation and isolation, see 4.2, “Isolation and security” on page 159.
- ▶ SoftLayer and PureApplication Service offer network security. For more information, see these sections:
 - 4.1.1, “IBM PureApplication Service connectivity” on page 140
 - 4.1.5, “Data center connectivity” on page 154
 - 4.2, “Isolation and security” on page 159

- ▶ PureApplication Service offers a denial of service security service that is based on the IBM DataPower® appliance, and you can add additional configuration manually on the appliance.
- ▶ IBM DataPower offers security capabilities, and IBM DataPower is available as a pattern in PureApplication System and PureApplication Service.
- ▶ PureApplication System and PureApplication Service provide user management and security auditing capabilities.

Physical security

Physical security mainly focuses on the security of IT equipment and physical network security.

On a private cloud, you own the physical security for the PureApplication Systems and the data centers in which they are located. The network is protected by the company's firewall.

On a public cloud, PureApplication Service relies on the network-within-a-network topology of SoftLayer for physical network security. This architecture ensures that systems are fully accessible to authorized personnel only.

Two layers are in the SoftLayer network-within-a-network:

- ▶ The public network layer to handle public traffic
- ▶ The private network layer to handle internal traffic

From the private network, it is possible to establish a secure connection to data centers that are on premises, by choosing between different network connection options. For more information, see "Network considerations" on page 38.

Compliance

When you adopt a public cloud, you need to trust the capabilities of your cloud service provider at different levels and likely, security generates many concerns. Enterprises generally need visibility of the security strategy that is adopted by the cloud service providers. Security compliance represents a way for a cloud service provider to make its security strategy visible. Several types of security compliance are available.

Table 2-5 lists the more common types of security compliance.

Table 2-5 Security compliance types

Compliance	Description
Federal Risk and Authorization Management Program (FedRAMP)	<i>The Federal Risk and Authorization Management Program (FedRAMP) is a government-wide program that provides a standardized approach to security assessment, authorization, and continuous monitoring for cloud products and services.^a</i>
Federal Information Security Management Act of 2002 (FISMA)	<p><i>The Federal Information Security Management Act of 2002 ("FISMA", 44 U.S.C. § 3541, et seq.) is a United States federal law that was enacted in 2002 as Title III of the E-Government Act of 2002 (Pub.L. 107 - 347, 116 Stat. 2899). The act recognized the importance of information security to the economic and national security interests of the United States.^b</i></p> <p><i>The National Institute of Standards and Technology (NIST) vision of the FISMA includes^c the following information:</i></p> <ul style="list-style-type: none"> ► <i>Standards for categorizing information and information systems by mission impact</i> ► <i>Standards for minimum security requirements for information and information systems</i> ► <i>Guidance for selecting appropriate security controls for information systems</i> ► <i>Guidance for assessing security controls in information systems and determining security control effectiveness</i> ► <i>Guidance for the security authorization of information systems</i> ► <i>Guidance for monitoring the security controls and the security authorization of information systems</i>
Payment Card Industry Data Security Standard (PCI DSS)	<i>PCI Security Standards are technical and operational requirements that are set by the PCI Security Standards Council (PCI SSC) to protect cardholder data. The Council is responsible for managing the security standards, while compliance with the PCI Security Standards is enforced by the payment card brands. The standards apply to all organizations that store, process, or transmit cardholder data with guidance for software developers and manufacturers of applications and devices that are used in those transactions.^d</i>
Statement on Standards for Attestation Engagements (SSAE) No. 16 (SSAE 16)	<i>Statement on Standards for Attestation Engagements (SSAE) No. 16 is an attestation standard that is put forth by the Auditing Standards Board (ASB) of the American Institute of Certified Public Accountants (AICPA) that addresses engagements that are undertaken by a service auditor for reporting on controls at organizations (that is, service organizations) that provide services to user entities, for which a service organization's controls are likely to be relevant to a user entity's internal control over financial reporting (ICFR).^e</i>

a. Program Overview: <https://www.fedramp.gov/about-us/about/>

b. Federal Information Security Management Act of 2002:

https://en.wikipedia.org/wiki/Federal_Information_Security_Management_Act_of_2002

c. Federal Information Security Management Act (FISMA) Implementation Project:

<http://csrc.nist.gov/groups/SMA/fisma/>

d. How to Be Compliant: Getting Started with PCI Data Security Standard Compliance:

https://www.pcisecuritystandards.org/merchants/how_to_be_compliant.php

e. SSAE 16 Definition: Provided by the SSAE 16 Resource Guide:

<http://www.ssaes16.org/what-is-ssae-16/ssae-16-definition.html>

For more information, see the following websites:

- ▶ *Cloud Security: Iron clad your cloud:*
<http://www.softlayer.com/security>
- ▶ *Compliance: Compliance without complication:*
<http://www.softlayer.com/compliance>
- ▶ *Security and trust in IBM PureApplication System:*
<https://ibm.biz/BdHhJC>

Network considerations

When we think of a hybrid cloud, networking is typically one of the first areas of interest. A hybrid cloud establishes a bridge between a private and public cloud. The network is the foundation of the bridge. The concern about networking is how to deliver a hybrid cloud without affecting performance and availability negatively.

The importance of networking depends on the use cases that you implement. For example, the implementation of an application for which the front-end components are off premises and the back-end components are on premises requires a solid network between the private cloud and the public cloud.

PureApplication Service offers three options to establish a connection with an on-premises site:

- ▶ Open virtual private network (OpenVPN) connection
- ▶ Site-to-site Internet Protocol Security (IPsec) VPN connection
- ▶ Direct Link from the client's network provider to the SoftLayer point of presence (POP)

Every connection option has a specific strength. The first two options are no-charge options. For more information about each type of connection, see 4.1.3, "Connectivity use cases" on page 142.

For more information about the way to set up PureApplication System in a data center network, see *Network design for IBM PureApplication System, Part 1: Connecting the system to the data center network*:

<https://ibm.biz/BdXREM>

System management considerations

In an on-premises scenario, an organization directly handles the system management capabilities. However, the adoption of a public cloud introduces variants, and the need for off-premises system management depends on the specific cloud service model (IaaS, PaaS, or software as a service (SaaS)) and on the capabilities that are offered or managed by the cloud service provider. Other system management considerations are listed:

- ▶ The possibility of unified or integrated management of both the private and public clouds
- ▶ The capability to regularly monitor the usage of the off-premises resources
- ▶ The possibility of a single self-service portal for servicing all environments
- ▶ The possibility of applying local service management tools, such as ticketing, service desk, patching, monitoring, configuration management, and change management, to off-premises resources

A hybrid scenario that is based on PureApplication System and PureApplication Service offers the following capabilities:

- ▶ The ability to consistently align the on-premises and off-premises catalogs of artifacts, such as images and patterns.
- ▶ The ability to use a single monitoring view of on-premises and off-premises processes that are running.
- ▶ IBM introduced an important enhancement to IBM PureApplication System, which is called the *Call Home* feature. In high severity errors (where both hardware and software can be the source of the problem), the PureApplication System automatically opens a service ticket. The PureApplication System also can collect the necessary logs and system configuration information, upload them to the ticket, and accelerate the problem determination and problem solution phases.
- ▶ The possibility to use a shared service to get the lifecycle management platform for Red Hat Enterprise Linux infrastructure, which is provided by Red Hat Satellite.
- ▶ The possibility to use IBM Endpoint Manager and IBM BigFix® to implement a patch management solution for Windows and AIX guest operating systems.

For more information about the system management capabilities of PureApplication Service and PureApplication System, see the IBM Knowledge Center:

- ▶ *Administering users, user groups, and security* (PureApplication Service 2.1.1):
<https://ibm.biz/BdHhi8>
- ▶ *Organizational structure in PureApplication System operations*:
http://www.ibm.com/developerworks/websphere/techjournal/1307_brown2/1307_brown2.html
- ▶ *Six steps to deploying Red Hat Satellite Server with IBM PureApplication System*:
<http://expertintegratedsystemsblog.com/2015/03/six-steps-to-deploying-red-hat-satellite-server-with-ibm-pureapplication-system/>
- ▶ *Adopting IBM PureApplication System V1.0*, SG24-8113:
<http://www.redbooks.ibm.com/abstracts/sg248113.html>

2.1.4 Adoption

Significant shifts require significant attention. Adopting a cloud computing strategy is not only a technology issue because the change influences the organization, too. A project might fail even if the technical complexity is low. Several key success factors for adopting the hybrid cloud are listed:

- ▶ The implementation of a hybrid cloud affects all IT departments. Therefore, the necessary (CxO) sponsorship is crucial to ensure the success of the project.
- ▶ The identification of roles and responsibilities of the involved actors and structures. It is important to understand the responsibilities of the cloud service provider.
- ▶ The adoption of a hybrid cloud can be gradual, and the adoption roadmap can be iterative, but it is crucial to define clear and measurable objectives.
- ▶ Define your communication plan and training programs. For the greatest success, the technical solution needs to be implemented well, fully adopted, correctly used, and managed well.

2.1.5 Continuous optimization

The roadmap to the hybrid cloud does not end. It is iterative. Over time, continuous optimization is required for several reasons:

- ▶ Adopting different cloud service models requires the continuous alignment of your everyday activities, depending on the system management capabilities that are offered by the cloud service provider.
- ▶ The cloud service provider might change its services, or you might want to change cloud service providers.
- ▶ You can use cloud application portability to use more than one cloud service provider.

2.2 A hybrid infrastructure with the PureApplication family

When you build a hybrid cloud, you need to consider important points at the infrastructure level. First, planning and preparation are needed. If you are extending your existing platform into a cloud model after you define a business impact analysis and a cloud readiness assessment, the organization needs to understand which workloads need to extend to the cloud.

A readiness assessment allows the enterprise to understand whether the users and data are ready to migrate to the cloud. From this analysis and a return on investment (ROI) report, the enterprise can decide on the best migration strategy: build, lease (IaaS or PaaS), or subscribe to a cloud service (SaaS). For more information about cloud adoption, see 2.1, “A five-step roadmap for establishing a hybrid cloud” on page 26.

The next step is to select your hardware based on what you are deploying to the cloud. Server, storage, and network requirements need to be considered. The use of converged platforms is the primary reason why clouds are getting smaller. *Converged platforms* refer to using the same standard preconfigured hardware where possible.

Modern data centers are defined as software-defined data center platforms, and it is important when you move to the cloud to think about the logical and virtual controls. By using a hypervisor and network, you can consider storage and compute resources as a software-defined category to supply in data centers. When you build the infrastructure, verify whether those systems help to scale workloads. Scaling workloads is important to guarantee cloud resiliency because it keeps business applications running in a hybrid scenario.

Another important consideration for a hybrid cloud is to understand your replication and distribution mechanism: what you are replicating and to where. Certain cloud solutions deploy to a hybrid cloud to keep the data and applications close to users. Other applications use the hybrid cloud to control bursts and branch locations in a geographic expansion.

End-to-end security is another key point. See “Security considerations on the infrastructure level” on page 42.

Automation and orchestration can be added value components of the hybrid cloud that interact with other areas of the enterprise, such as load balancers or configuration management databases (CMDBs). This whole environment, the “sky,” might include multiple clouds, which are sometimes from different vendors. Resources can be provisioned and deprovisioned on demand according to underlying rules and user permissions by using a single portal that has the same look and feel for various departments or LOBs.

Sometimes, it is important to interrupt an automatic deployment to ensure that, for example, a developer is not self-provisioning too many resources that will be billed to the department at the end of the month, which is a concern for development.

Therefore, over the long term, it is important not to think of the hybrid cloud in isolation from the whole “sky,” and it is wise to work with an orchestration product that accepts open source management systems, providing the users or LOBs the ability to interact, extend, and scale to other cloud platforms. As you plan your hybrid model, ensure that you look at a solution that can work seamlessly, or with minimal configuration, with cloud-based orchestration and automation tools to handle enterprise-wide, long-term considerations.

Consider the following reasons to add IBM Cloud Orchestrator on top of PureApplication deployments:

- ▶ Data center integration
 - Integrates with firewalls, load balancers, and CMDBs
- ▶ Manual process integration
 - Add approval processing
- ▶ Self-service catalog:
 - Provide a customized catalog across the PureApplication family and other offerings
 - Provide a consistent user experience across multiple service domains
- ▶ Cross-cloud orchestration
 - Place workloads across PureApplication and other private and public clouds for optimal quality of service (QoS)

From the initial deployment orchestration to load orchestration or load balancing, application workloads must be balanced so that they support floating demands, such as changes in the user load, changes in the data (location) access, and changes in application instances, while they dynamically reassign required data center resources, including storage, networking, and compute power.

For example, move a particular application off premises when a sudden need arises to support more users for an initially on-premises application. The load balancing platform needs to be able to intelligently move a user instance closer to a user's application access point or data point while it controls resources and application access.

The load balancing platforms tend to be more sophisticated, incorporating application firewall features, controlling on-premises and off-premises resources, and being both physical and virtual.

PureApplication System, Software, and Service provide platform services that encapsulate preconfigured, policy-managed platform services, such as caching elasticity, failover, load balancing, security, database, and middleware.

These services allow faster enterprise-ready application development by using the automation of common platform management tasks, such as environment setup, middleware configuration, and application deployment.

Monitoring and management are important points in maintaining a hybrid cloud environment. Centralized monitoring is important because it allows administrators to delegate controls and permissions while they directly view the systems' behavior.

The GUI from an IBM PureApplication System provides you with monitoring and management tooling for both hardware and software components at various granular levels. PureApplication Service just like any other IBM PureApplication product can also be incorporated into a centralized monitoring solution. Proactive monitoring and management of PureApplication System and Software will reduce the frequency and severity of issues, ensure optimal operation and performance, and minimize the impact on deployed workloads and system availability.

Security considerations on the infrastructure level

Consider several points for security on the infrastructure level as shown in Figure 2-8.



Figure 2-8 Security challenges to move to cloud

In a cloud environment, access expands, responsibilities change, control shifts, and the speed of provisioning resources and applications increases, greatly affecting all aspects of IT security. From these challenges, key cloud security concerns must be considered at the infrastructure level:

- ▶ Manage the registration and control the access of thousands, possibly millions, of cloud users in a cost-effective way
- ▶ Ensure the safety and privacy of critical enterprise data in cloud environments without disrupting operations
- ▶ Provide secure access to applications in the cloud
- ▶ Manage patch requirements for systems (virtual and physical)
- ▶ Provide protection against network threats and vulnerabilities in the cloud
- ▶ Protect VMs by establishing firewalls and encryption
- ▶ Achieve visibility and transparency in cloud environments to find advanced threats and meet regulatory and compliance requirements

The PureApplication family offers a complete set of capabilities and considerations for a secure hybrid cloud.

Functional security

The PureApplication platform provides various functional security capabilities, including admin user authentication, access authorization, auditing of critical operations, and data protection.

Authentication

PureApplication System and PureApplication Software support authentication against an LDAP directory. A central shared LDAP directory infrastructure can facilitate central user account management, allowing for easier integration between one or more PureApplication Systems and PureApplication Software instances. Figure 2-9 illustrates this type of a configuration.

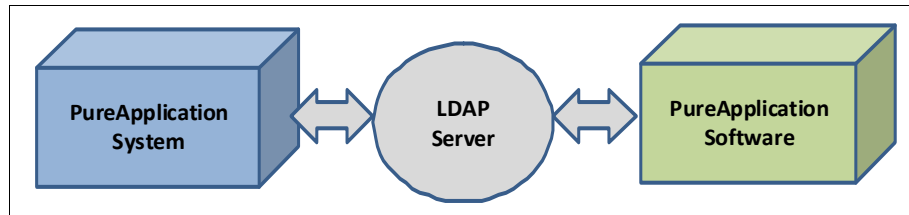


Figure 2-9 LDAP configuration to enable a single authentication point in a hybrid environment

For high availability (HA) on LDAP, you need to configure another LDAP server in a multiple master configuration, for instance, relying on a load balancer as described in Figure 2-10.

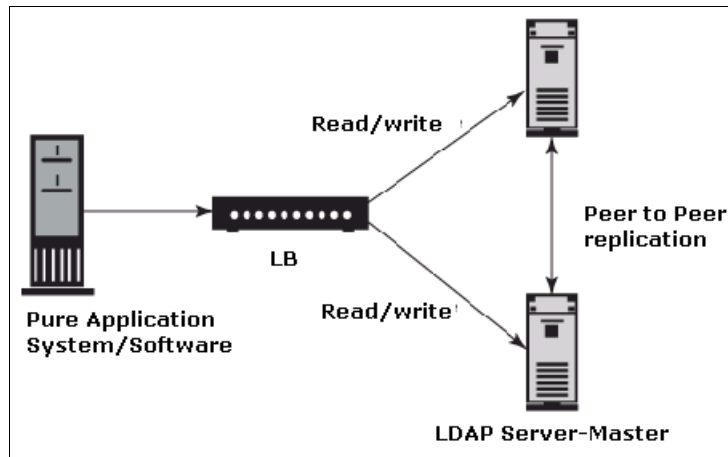


Figure 2-10 HA in LDAP that uses centralized authentication for PureApplication System and Software

For more information, see the following websites:

- *Integrating PureApplication System with LDAP servers* (describes how to configure LDAP in PureApplication System):
<https://ibm.biz/BdHVVHg>
- *Approaches for solving problems with Tivoli Directory Server synchronization* (about LDAP replication topologies):
<http://www.ibm.com/developerworks/tivoli/library/t-tdsync/>

Authorization

In the PureApplication family, when you view details for system users (by using the user interface), you can see counters for various cloud resources. Administrative users are authorized to access these counters, based on access rights that were granted. Depending on which role the user is granted, not all cloud resources are visible on the user interface.

The following resources, which are managed at the console, support access rights and permissions:

- ▶ Cloud groups
- ▶ Internet Protocol (IP) groups
- ▶ Virtual machines
- ▶ Virtual appliances

Auditing

PureApplication audit logs can be enabled for various administrative operations:

- ▶ A compute node that is added or removed
- ▶ A system restore is performed
- ▶ A role or permission is changed for a user (local or LDAP)
- ▶ An IP address or IP group is added or removed

Data protection

The key consideration in hybrid cloud networking is to provide end-to-end security between off-premises and on-premises systems. One option at the infrastructure level is to provide an IBM DataPower Gateway, which provides reverse proxy, Secure Sockets Layer (SSL) termination, and load balancing functions. A DataPower Gateway is typically in the demilitarized zone (DMZ) in dual-firewall environments. Applications can connect directly to a DataPower Gateway, after which requests are forwarded to the back-end systems.

The DataPower Gateway comes in physical and virtual forms, with the following features:

- ▶ High availability
- ▶ Failover
- ▶ Load balancing
- ▶ Message security
- ▶ Data conversion

The DataPower Gateway excels at Extensible Markup Language (XML) and Representational State Transfer-related (RESTful) web services processing. Version 7.2 of DataPower Gateway enhances cloud and on-premises security with support for elliptic curve cryptography (ECC), server name indication, and perfect forward secrecy to protect against both malicious protocols and denial-of-service (DoS) attacks.

IBM API Management is an optional feature that is available with DataPower Gateway. APIs are useful because they can be configured by the developers of the on-premises resources and they can be used by other developers of cloud applications, even from different companies. For instance, by safely exposing certain APIs, a car sales company might allow developers from a bank or loan company to offer easy, seamless loans to potential car customers. DataPower Gateway implements the API mappings, and API Management makes mainframe APIs more consumable.

A scenario for implementing a solution might also include the use of DataPower Gateway as the endpoint of a secure connection, starting with an off-premises application and passing through a Vyatta appliance. The DataPower Gateway can be installed either in the DMZ or in the enterprise trusted zone.

Figure 2-11 shows the IBM DataPower Gateway solution for data protection. SoftLayer also provides DataPower for incoming traffic for certain functionalities, such as protection against DoS.

For more information, see *IBM DataPower Gateway Virtual Edition*:

<https://ibm.biz/BdHVXW>

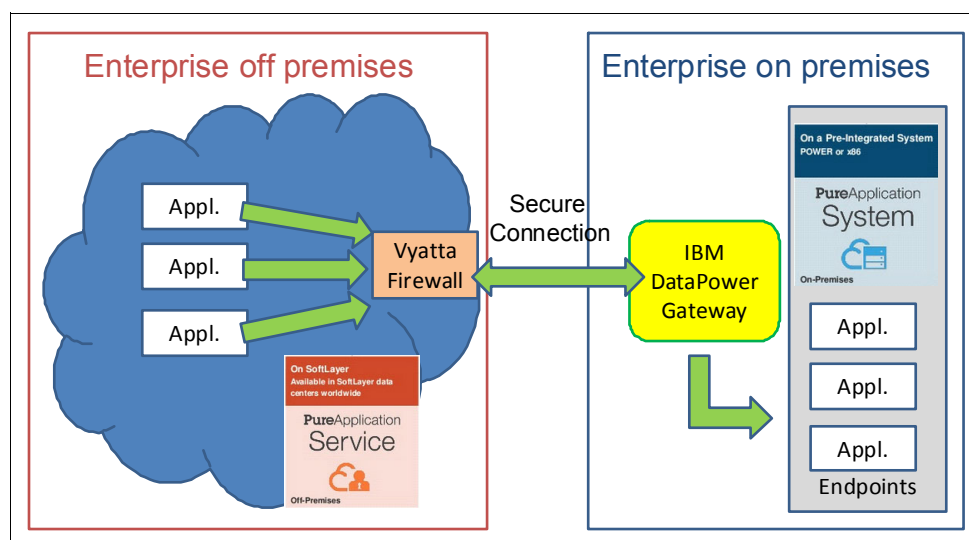


Figure 2-11 The DataPower Gateway solution for data protection

Infrastructure security

The PureApplication family provides various infrastructure security capabilities, which are available after the purchase with no configuration. Consider these infrastructure security capabilities when you enable a hybrid cloud solution.

Environment segregation

With PureApplication System, PureApplication Software, and PureApplication Service, you can create *environment profiles* (runtime environments) with separated resources (IP groups and cloud groups). Also, environment profiles allow groups finer-grained access (with isolation) to the same compute node because only one cloud group can access a compute node.

For instance, because a runtime environment in the PureApplication family is a cloud group, creating three separate environment profiles for production, testing, and development means that three cloud groups are needed for good isolation. A *cloud group* consists of one or more compute nodes and at least one IP group. Compute nodes are hardware resources that are part of the PureApplication family.

Clients can save expenses because if one group needs only four cores of the smallest 16-core Intel compute node, another group can be assigned to the same cloud group but to a different environment profile that uses the other eight cores. This isolation can be set up in development, quality assurance, preproduction, and production.

Figure 2-12 explains the relationships among the components of environment segregation.

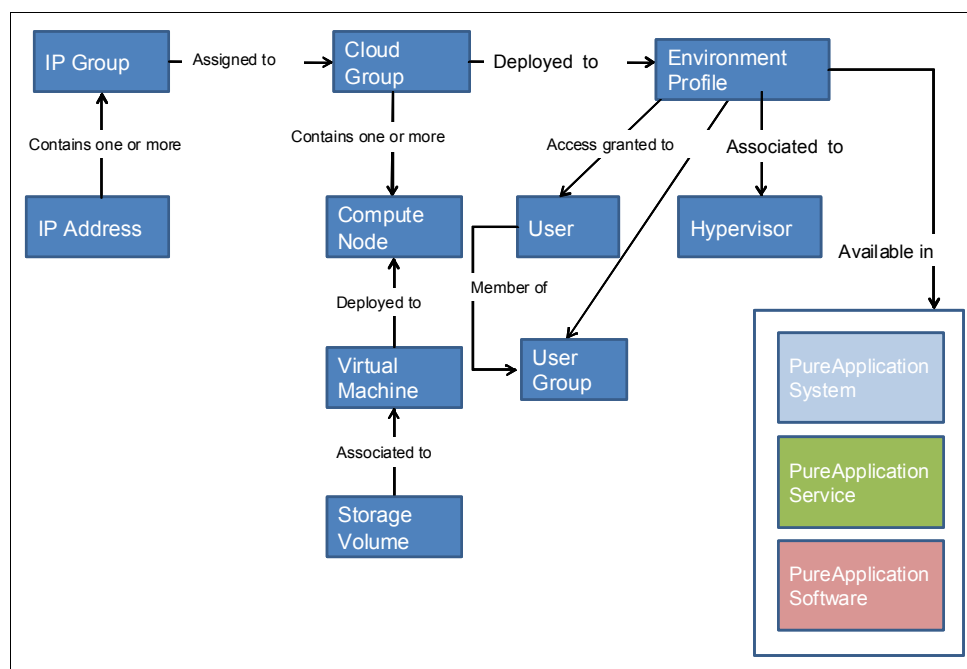


Figure 2-12 Components that are involved in environment segregation

You can clone a complete configuration (pattern) when you move, for example, from development to production. From this standpoint, your application is consistent in the infrastructure for each level of the development lifecycle, and you can adjust for more or less capacity when it is needed.

For more information about these capabilities, see *Integrating IBM PureApplication System into an Existing Data Center*, SG24-8285.

For more information about implementation, see 4.2, “Isolation and security” on page 159.

Firewalls and connectivity

Firewalls are important for restricting access to the PureApplication System in an on-premises network. In a hybrid cloud scenario, by connecting to a public cloud by using the PureApplication Service offering, you can configure corporate firewall rules to PureApplication Service IP addresses and establish a connection by an OpenVPN connection, site-to-site IPsec, a VPN connection, or Direct Link from a client network provider to a SoftLayer POP. In PureApplication System, the enterprise configures firewall IPtables (corporate firewall).

Figure 2-13 shows this scenario.

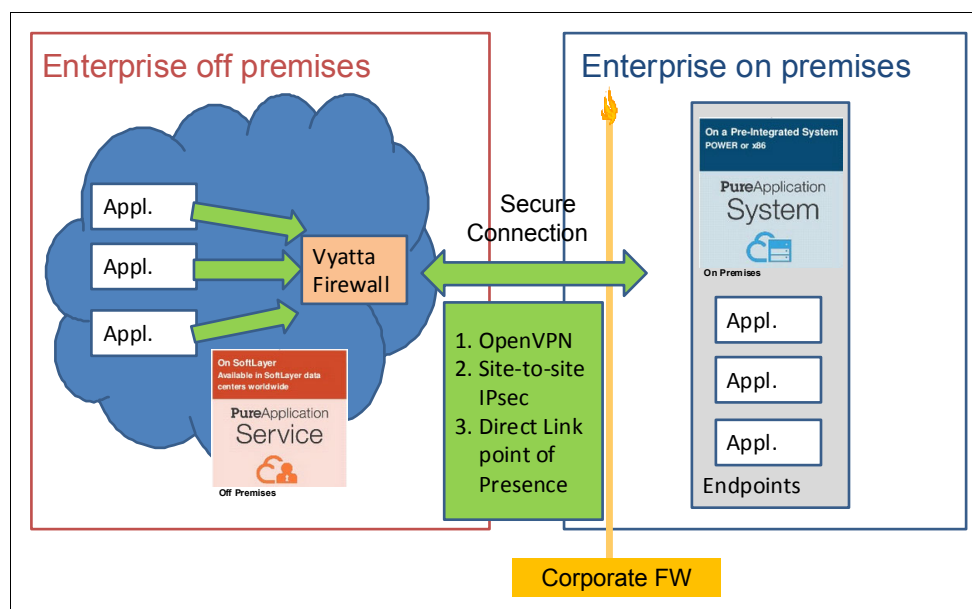


Figure 2-13 Corporate firewall configuration between PureApplication Service and System

For more information about connectivity, see 4.1, “Connectivity” on page 140.

Intrusion protection

Intrusion protection is an important requirement in a hybrid cloud scenario. It can be enabled to discover threats so that they can be addressed. Intrusion protection policies are enabled on corporate firewalls in a client data center environment for PureApplication System and PureApplication Software, for instance. PureApplication Software has a Vyatta service that can be subscribed on SoftLayer to allow firewall enablement.

Secure application container management

In PureApplication, each application can be secured from the other applications when it is deployed on VMs. This security is defined by IPtables. This level of security guarantees isolation of virtual memory, CPU, and file systems’ use with resource limits that are provisioned by each application.

Server security hardening

We recommend that OS administrators harden the network and operating system regularly by using tools, such as IBM Endpoint Manager, which is also known as *BigFix*. The Endpoint Manager Service is a shared service that acts as a relay between an external Endpoint Manager Server and Endpoint Manager clients. These clients are integrated, by default, into your virtual application and virtual system deployments. Endpoint Manager Version 9.0 and later support server security hardening.

For more information, see the following websites:

- *IBM Endpoint Manager Service* (PureApplication Software):
<https://ibm.biz/BdHhsV>
- *IBM Endpoint Manager Service* (PureApplication System):
<https://ibm.biz/BdHhsJ>

- *Installing on Linux systems* (to install IBM EndPoint Manager for Linux Systems):
<https://ibm.biz/BdHhsA>
- *Installing on Windows systems* (to install IBM EndPoint Manager for Windows):
<https://ibm.biz/BdHhsu>

Another service to update operating systems for Red Hat users is the shared service Red Hat Satellite Server.

Red Hat Satellite Server is a lifecycle management platform for the Red Hat Enterprise Linux OS that provides the tools to deploy, update, monitor, and manage systems instances within the PureApplication family that are running inside or outside of the client's environment. This service consumes no VM resources, and it is deployed for each cloud group. This service requires at least one VM on PureApplication System to which the shared service links. The Red Hat Satellite Service is based on the Red Hat Satellite, which provides an OS patch management solution for Red Hat client VMs. It is available with the PureApplication family, and systems can connect to this shared service on premises or off premises.

For more information, see the following websites:

- Red Hat Satellite Service (for PureApplication System):
<https://ibm.biz/BdHhsL>
- *Red Hat Satellite external shared service* (for PureApplication Software):
<https://ibm.biz/BdHhs9>
- *Red Hat Satellite external shared service* (for PureApplication Service):
<https://ibm.biz/BdHhsC>

Operational security

The PureApplication family provides a robust operational security environment with the following controls.

Fix management

PureApplication System, PureApplication Software, and PureApplication Service offer a fix management feature that PureApplication administrators can use to update the firmware. With PureApplication System, the procedure is enhanced because of its design. Each component (including the top-of-rack switches, IBM V7000 Storwize®, and IBM PureSystems® Manager) is duplicated and runs in active and passive modes. The code that integrates these components enables the fix to be applied, essentially, with the click of a button.

PureApplication System contains the intelligence to patch a passive component, then move the relevant work from the active component to the passive component, and make the passive component the active component. Then, the new passive component is patched. This process occurs in the rest of the rack also. A process that used to take hundreds of hours for many shops now takes about 12 hours, and it can be performed when the system is hot (running all workloads).

Audit log consolidation and analysis

Audit log is used to monitor administrative activities. Configure external storage for auditing so that the PureApplication family can automatically push audit record packages to an external storage area when the internal database reaches its threshold. Save audit record packages externally so that you can analyze data offline and also archive data to meet various compliance and regulation requirements.

For a hybrid cloud, you might consider centralized auditing. It is possible to have the same external storage between the PureApplication family products. To use this option, an external server with enough storage capacity is configured to generate complete audit records. A typical use case is monitoring configuration changes in a centralized way in a hybrid cloud scenario that uses PureApplication System and PureApplication Software. To configure an external auditing source, see *Configuring external storage servers*:

http://www.ibm.com/support/knowledgecenter/SSNLXH_2.1.1/doc/systemconsole/t_audconfig.dita?lang=en

User access management

The goal of user access management is to follow the best procedures to administer user permissions for the PureApplication family. User access management achieves compliance with IT regulations and helps protect your environment from internal and external security threats.

When you create users in the PureApplication family, they automatically receive the default permission to deploy objects in the cloud, including virtual system patterns. Any additional permissions need to be assigned manually. When you assign additional permissions, consider the following suggested practices:

- ▶ To prevent the abuse of user rights in your environment, try to minimize assigning multiple management responsibilities to users or user groups. Most importantly, use the separation of duties (SoD) strategy to protect the integrity of the auditor role. Isolate the assignment of auditing permissions to one or more users who do not have other powerful administrative capabilities, such as the system or cloud administration permissions.
- ▶ Auditors are responsible for monitoring normal and abnormal activities in the system. Administrators are responsible for administering resources in the system. These different responsibilities must be assigned to different individuals.

Note: In addition to offering discrete permissions for separating user duties, PureApplication Service implements two other SoD-oriented policies to help you control user activity in the cloud. These policies limit the authority to assign user permissions:

- ▶ Only users with the following roles can make permission assignments:
 - Workload resources administration with Manage workload resources (full permission)
 - Cloud group administration with Manage all cloud groups permission (full permission)
 - Hardware administration with Manage hardware resources permission (full permission)
 - Security administration with Manage security permission (full permission)
 - Auditing with Manage auditing permission (full permission)
- ▶ Users must have at least one of the five full permission administrator roles and the delegation security role to be able to delegate their security roles to other users.

Therefore, you can implement secure user administration access, in which no user can act without a record of the action. All of these measures protect the integrity of your environment and need to be part of your operational security practices.

For more information, see the following resources:

- *Administering users, user groups, and security* (for PureApplication System):
<https://ibm.biz/BdHhsT>
- *Administering users, user groups, and security* (for PureApplication Software):
<https://ibm.biz/BdHhsw>
- *Administering users, user groups, and security* (for PureApplication Service):
<https://ibm.biz/BdHhsk>

Physical security

PureApplication Service relies on the network-within-a-network topology of SoftLayer for physical network security and ensures that systems are fully accessible only to authorized personnel. The public network layer handles public traffic to hosted websites or online resources. The private network layer allows for true out-of-band management through a distinct stand-alone third carrier over SSL, point-to-point tunneling protocol (PPTP), or IPsec VPN gateways. The data center-to-data center network layer provides free and secure connectivity between servers that are housed in separate SoftLayer facilities. Every SoftLayer data center is fully secured with controls that meet Statement on Standards for Attestation Engagements (SSAE) number 16 and industry-recognized requirements, without exception. For more information, see the SoftLayer security compliance page:

<http://www.softlayer.com/compliance>

For PureApplication System or PureApplication Software, you own the physical security for the local instance. Your data center is secured behind your company firewall.

2.3 Patterns and the PureApplication family

We follow the blueprint recommendations that are described in “Assess your applications” on page 28 to cover the pattern engine capabilities for a hybrid cloud. Pattern Builder is covered as a standard tool for working with patterns.

2.3.1 Pattern engine

Pattern engine is a technical platform that supports IBM patterns. A pattern engine represents one of the most important parts of the PureApplication family.

Figure 2-14 shows that you can use pattern engine to configure patterns for deployment in the PureApplication family.

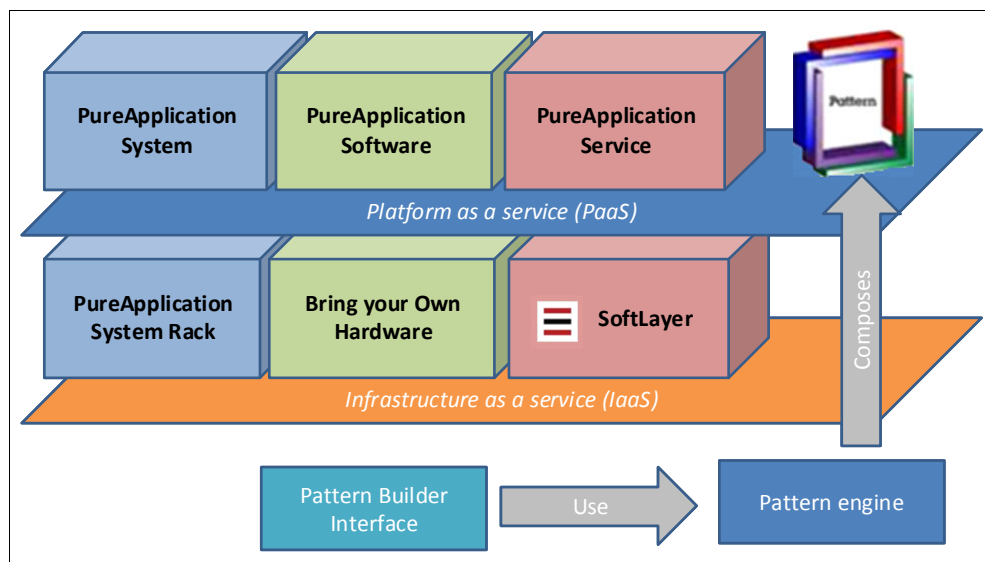


Figure 2-14 Pattern engine composes patterns to use on the PureApplication System family

With patterns, you can move your workload between PureApplication products for cloud provisioning, simplifying your time-to-market deployment, based on business needs and if workloads were built on the same hypervisor. Users can deploy a pattern in one PureApplication System rack, spread to other racks in a multiple system deployment, and also export to PureApplication Software and PureApplication Service. Hybrid cloud use cases, such as system of records and system of engagement integration, can use pattern engine to move workloads between these products.

Pattern engine offers the following deployment methods (*virtual patterns*), which are important in a hybrid cloud scenario.

Appliances

A *virtual appliance* is a preconfigured VM image that can be used or customized. It contains a software stack that is designed to run on a VM platform, which can be a hypervisor type 2 (hosting hypervisor). Virtual appliances represent basic parts for building more complex topologies. Users can add new virtual images to the PureApplication catalog, creating a virtual appliance template and deploying multiple instances of this single template. Virtual appliances are portable, self-contained configurations of a software stack.

For more information about virtual appliances in a hybrid cloud scope, see 2.5, “Achieving hybrid cloud application portability” on page 68.

System patterns

Virtual systems are an important deployment type among the PureApplication products. A *virtual system pattern* is a provisional unit for one or more VMs to be installed, configured, and integrated to implement an application topology. Use virtual system patterns to graphically describe a middleware topology to be built and deployed to the cloud. This pattern is middleware-centric, and users work on this pattern when it needs more customization of middleware components.

By using virtual images or parts from the catalog, in addition to optional script packages and add-ons, you can create, extend, and reuse middleware-based topologies. Virtual system patterns provide control over the installation, configuration, and integration of all of the components that are necessary for your pattern to work. For more information about virtual system patterns on the hybrid cloud scope, see 2.5, “Achieving hybrid cloud application portability” on page 68.

Application patterns

A *virtual application pattern* is an application-centric approach for deploying applications on the cloud. With virtual application patterns, users do not worry about the topology that is required to run their application. Instead, users specify an application (for example, an .ear file) and a set of policies that correspond to the service level agreement (SLA) to achieve. PureApplication pattern engine capabilities then transform that input into an installed, configured, and integrated middleware application environment instance.

The system also automatically monitors application workload demand, adjusts resource allocation, and fine-tunes prioritization to meet your defined policies. Virtual application patterns address specific solutions, incorporating years of expertise and preferred practices from IBM. For more information about virtual application patterns on the hybrid cloud scope, see 2.6, “Deploying applications by using a hybrid cloud” on page 82.

The pattern engine supports OpenStack Heat Orchestration Template (HOT) deployments. For more information, see 2.4.3, “OpenStack (Heat and HOT)” on page 67.

2.3.2 Choosing the best pattern type for a hybrid cloud

The use of virtual application patterns clearly involves the least amount of work and the fewest customizations, for example, for configuring a session timeout for a web application. A virtual application pattern can create an application, and PureApplication will manage the infrastructure that is required to meet a specified QoS. This strategy has the lowest total cost of ownership (TCO) and the shortest time to provision, and you will receive reused patterns from IBM pattern expertise offerings.

However, not all configurations fit easily into an available virtual application pattern type. Sometimes, you need more granular control that is based on your needs for customization, ownership, business continuity, and specific security setups.

With virtual system patterns, you can specify the exact customized topology that you need to support your application, and you can benefit from pattern reusability. With virtual system patterns, you can also take advantage of features, such as auto-scaling.

The general rule for choosing the best pattern for your environment is to focus on the optimization, convenience, and standardization of the pattern because you will need less customization. If you need to control the topology and manage the environment through administrative consoles, the use of virtual system patterns is the best choice.

2.3.3 Pattern Builder

The PureApplication Pattern Builder provides a simple interface for controlling various aspects of patterns and deployments. This tool is available in the PureApplication family. You can create and edit patterns by using this tool, and you can also deploy patterns. You can start Pattern Builder from virtual systems or from virtual application options. The PureApplication System main menu, for instance, contains options to start Pattern Builder.

See Figure 2-15.

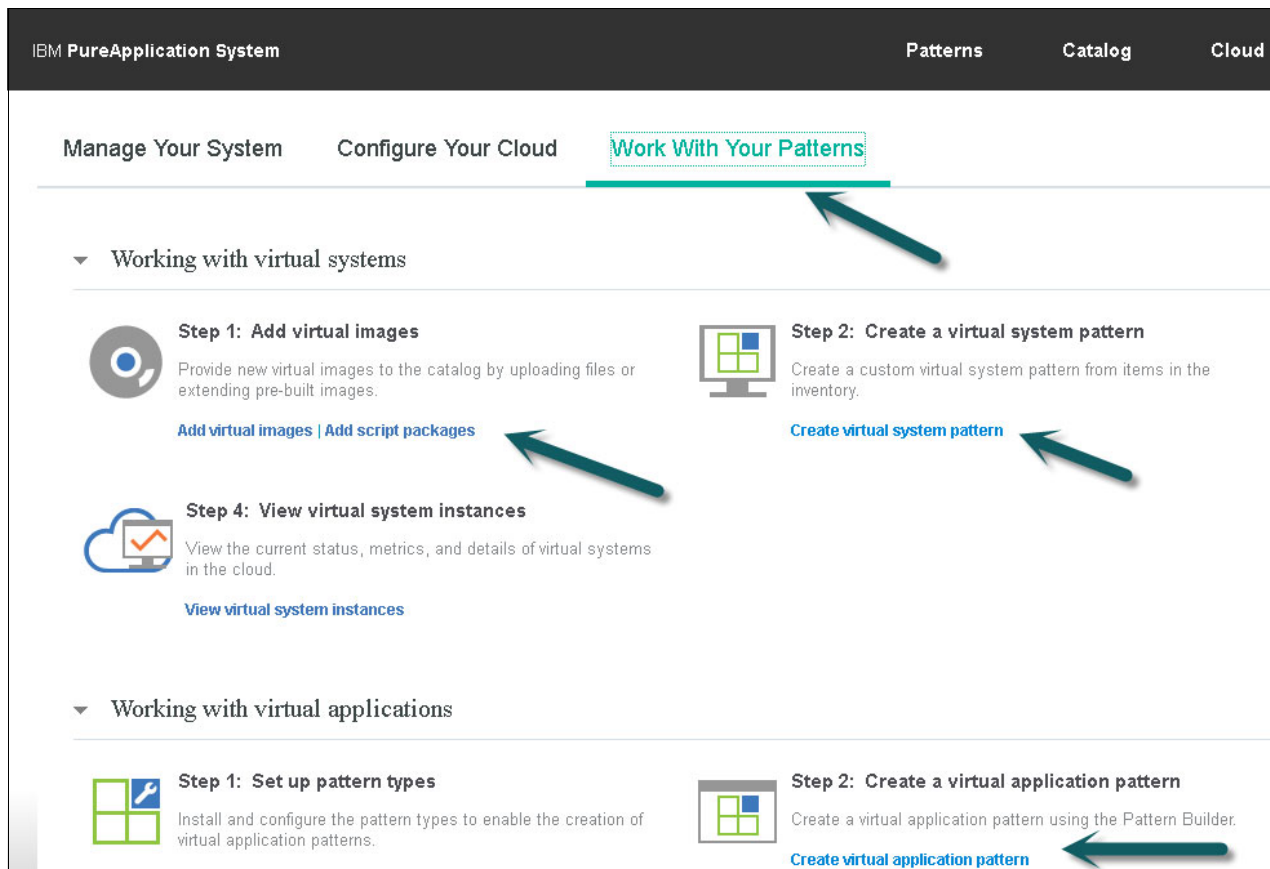


Figure 2-15 Pattern Builder main menu

To start Pattern Builder from the main menu that is shown in Figure 2-15, select either **Add virtual images**, **Add script packages**, **Create virtual system pattern**, or **Create virtual application pattern**. Then, select **Create New**. The following steps depend on your choice of virtual system or virtual application.

In Figure 2-16, you can see the layout of Pattern Builder for a virtual application pattern that uses the option for WebApp Pattern Type 1.0.

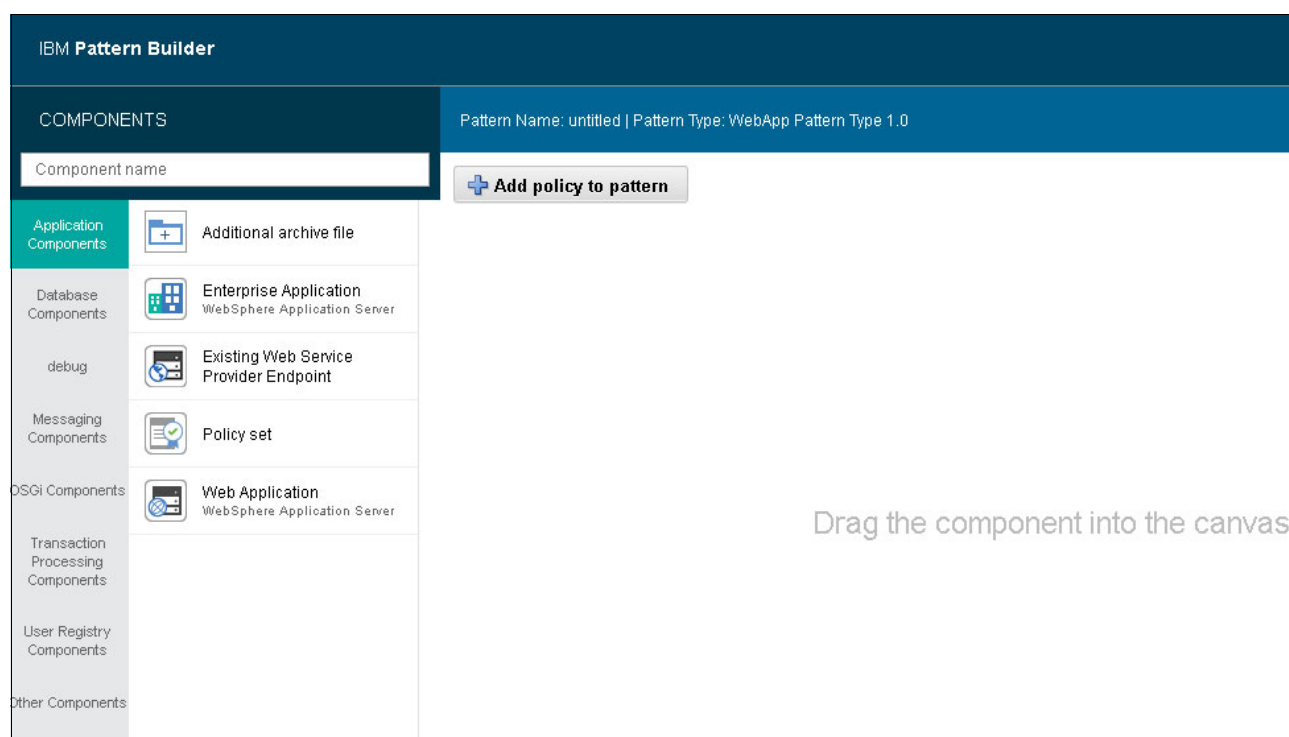


Figure 2-16 Pattern Builder

For more information about virtual patterns and Pattern Builder, see *Working with virtual patterns*:

<https://ibm.biz/BdHhLc>

2.3.4 Import and export pattern strategies in PureApplication

Import and export patterns in the PureApplication family are easy to use, and the best strategy depends on whether you select virtual system patterns or virtual application patterns.

For virtual system patterns, you can use the portal console or command-line interface (CLI) that you can download from the PureApplication System and PureApplication Software home page or from the PureApplication Service home page to which you subscribed.

You can export a virtual system pattern in a hybrid scenario by using, for example, PureApplication System or PureApplication Software on premises, then importing to PureApplication Software off premises when you need to expand your workload outside of your data center. Patterns are visible for all of the PureApplication family products to offer you the flexibility to move workloads from on premises to off premises or vice versa.

For virtual application patterns, you can use the portal console to import and export.

For more information, see 2.3.4, “Import and export pattern strategies in PureApplication” on page 54.

2.3.5 PureSystems Centre

PureSystems Centre is a website from which you can access patterns from IBM and IBM Business Partners, and support for maximizing the benefit of patterns. Pattern updates are available from IBM Fix Central. PureSystems patterns are part of a broad portfolio of solutions that accelerate deployment and simplify management for cloud, business, and infrastructure applications.

For more information, see the following websites:

- ▶ PureSystems Centre (about how to use patterns from IBM and IBM Business Partners):
<http://www.ibm.com/PureSystems/Centre>
- ▶ IBM Fix Central (for updates, including pattern updates):
<http://www.ibm.com/support/fixcentral/>

2.4 Open technologies and PureApplication integration

In this section, we describe the major open technologies on the market today, and the capabilities of the PureApplication family to integrate PureApplication products with other cloud workloads, including non-IBM technology-based cloud workloads. The PureApplication family has a continuous integration strategy for working with open technologies, such as Docker and Chef, in environments that use existing middleware assets, applications, and databases. Also, the pattern engine supports OpenStack Heat and Heat Orchestration Templates (HOT) so that you can deploy and manage HOT documents.

2.4.1 Docker

Docker is an open source product that uses container-based technology. It is a self-sufficient container that enables developers to build, ship, and run applications without changing their environment, such as a notebook or VM in the cloud.

Figure 2-17 shows an overview of Docker capabilities.

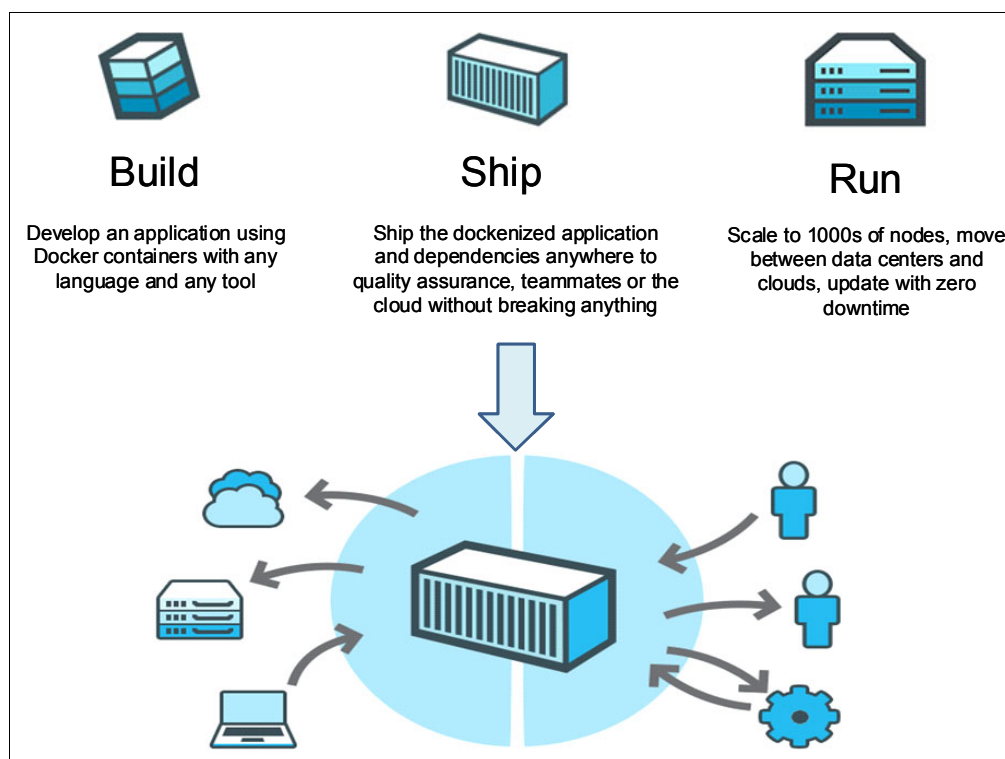


Figure 2-17 Docker's capabilities overview

Docker can manage multiple software stacks, such as a data center VM or public or private cloud (on PureApplication Systems). During the build phase, you can develop an application by using a Docker container, ship in a standard way (in Docker application format), and run workflows. You can also share Docker containers with other people and communities.

Docker versus VMs

Docker is a lightweight alternative for running software in a portable, though less isolated, virtual environment. Docker does not have to be tied to a hypervisor. It is an optional component. Figure 2-18 shows a comparison of the traditional virtualization approach with the Docker container approach.

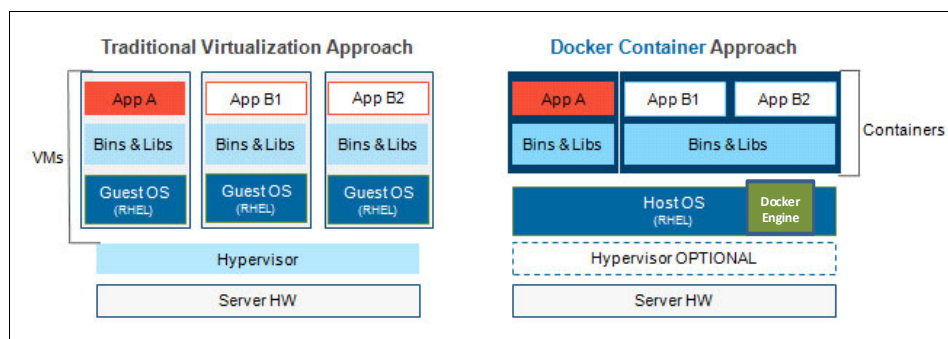


Figure 2-18 Comparison between virtualization with VMs and Docker container

Consider the following points when you compare VM and container approaches. See Table 2-6 on page 57. In addition to the comparisons in Table 2-6 on page 57, Docker with the PureApplication family offers advantages, such as auto-scaling in Docker containers.

Auto-scaling in Docker containers is described in “VMs and Docker together in a hybrid cloud” on page 57.

Table 2-6 VM and container comparison

Attribute	VM	Docker container
Startup time and performance	Slow (minutes to start up) because of hypervisor overhead	Fast (seconds to start up). No hypervisor overhead
Footprint	<ul style="list-style-type: none">▶ Large (nothing shared in terms of software)▶ One guest OS for each VM	Small (OS kernel shared)
Portability	Low	High (Linux)
Isolation and security	High	Medium
Resource constraints	Yes	Yes (CPU and memory)

The following list expands on the information in Table 2-6:

- ▶ **Startup time and performance:** Faster because Docker does not need a hypervisor. VMs always ship with all of the necessary dependencies, such as the VM’s binary files, libraries, and the guest OS.
- ▶ **Footprint:** Docker does not have an OS dependency on deployment so Docker is built on Linux Containers (LXC) low-level functions. This approach yields a smaller footprint than VMs, where middleware and application software link to the OS of each VM instance.
- ▶ **Portability:** Docker has a greater level of portability than VMs. VMs need to be customized by separating the software installation from the configuration steps on a full operating system because applications and middleware are tied to the OS image in those cases.
- ▶ **Isolation and security:** Isolation by using VMs is more secure than isolation with Docker because, based on hypervisor technology, no OS-level or middleware-level sharing occurs.
- ▶ **Resource constraints:** Full OS and memory management are installed in VMs with the associated overhead for virtual device drivers. This architecture causes more resource constraints than Docker because Docker containers execute with the Docker Engine rather than with the hypervisor. Docker has resource constraints that relate to memory and CPU.

VMs and Docker together in a hybrid cloud

In a hybrid cloud scenario, Docker containers can run in VMs because they are separate technologies. In off-premises workloads, Linux VMs with Docker can provide portability, fast deployment, and fast startup from Docker containers, and high isolation in middleware, OS, and capacity management from VMs in the PureApplication family.

Docker components

Figure 2-19 depicts the main Docker components.

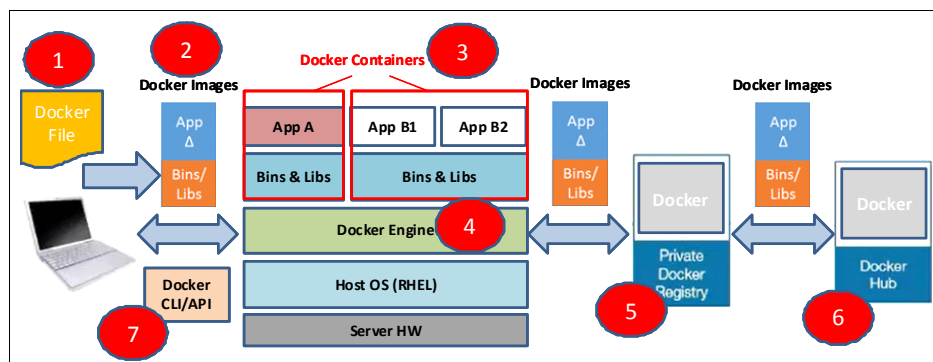


Figure 2-19 Docker components at a glance

The components that are numbered in Figure 2-19 are described in detail.

Docker file

Item 1 in Figure 2-19 shows a Docker file. A *Docker file* documents how to construct the container and what to execute when it is running. A Docker file is a build script that defines the following information:

- ▶ An existing image as the starting point
- ▶ A set of instructions to augment the image (Each instruction results in a new layer in the file system.)
- ▶ A metadata section with the ports that are exposed
- ▶ The command to execute when the image is run

A Docker file is a text file that shows how to build an image. A Docker file is readable. It can be maintained with application source code that can be stored in a software version management (SVM) repository, such as GitHub. A sample Docker file is shown in Figure 2-20.

```
FROM RHEL: 7.0
# add the files we require, jar + WLP files
ADD wlp-developers-extended-8.5.5.2.jar /root/
ADD wlp-developers-runtime-8.5.5.2.jar /root/
ADD JAXWSEJBSample.jar /root/
# install WLP
RUN apt-get update
RUN apt-get install -y default-jre
RUN java -jar /root/wlp-developers-runtime-8.5.5.2.jar --acceptLicense /root/
RUN java -jar /root/wlp-developers-extended-8.5.5.2.jar --acceptLicense /root/
RUN cd /root/wlp && java -jar ../JAXWSEJBSample.jar /root/wlp
EXPOSE 9080
CMD /root/wlp/bin/server run JAXWSEJBSample
```

Figure 2-20 Docker file sample

The Docker sample file in Figure 2-20 on page 58 shows the following information:

- Item 1: The sample base image in the beginning of the file.
- Item 2: Several files, such as simple .jar files or WebSphere Liberty Profile (WLP). The .jar files are moved from the host or another location that is mounted by using the **add** command.
- Item 3: **Run** commands for the installation. The example in Figure 2-20 on page 58 shows that it is installed in a default Java Runtime Environment (JRE). The liberty profile, .jar files, and application code .jar file are listed, too.
- Item 4: This section exposes the container port when it is running.
- Item 5: The **cmd** command shows what starts when the image is instantiated as a container.

Docker images

Item 2 in Figure 2-19 on page 58 shows a Docker file. A Docker image is a layered file system where each layer references the layer beneath it. Figure 2-21 is an example.

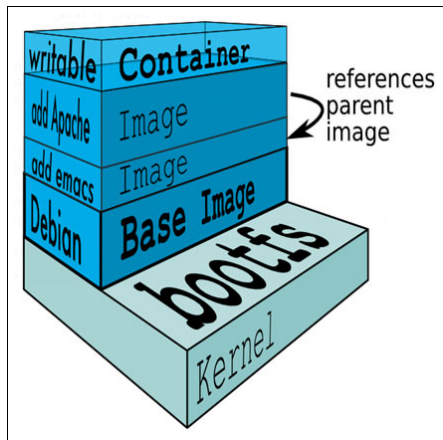


Figure 2-21 Docker image example

Image layers are built on one parent image, and only the differences are stored in each layer. Certain layers are read-only layers, and the container is a writable part of the image.

Docker container

Item 3 in Figure 2-19 on page 58 shows a *container*, which is a runtime instance of an image with an added read/write layer. The image is in build time, and the container is in run time, which is similar to the relationship between a class (*image*) and an object instance (*container*) of the class.

In Figure 2-22, containers c1 - c6 run layered on the OS1 Linux base image. Containers c7 and c8 run on top of the OS2 Linux image. All of the containers share a bootfs kernel image.

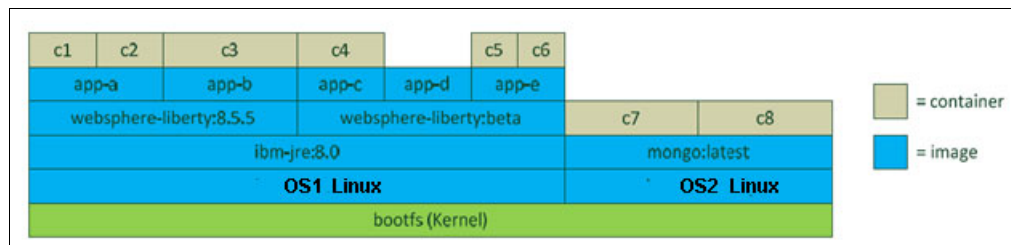


Figure 2-22 Placement of images and containers

Docker Engine

Item 4 in Figure 2-19 on page 58 shows the *Docker Engine*, which is a lightweight run time to run containers. It is closely associated with the host OS.

Docker Registry

Item 5 in Figure 2-19 on page 58 shows a *Docker Registry*, which stores and distributes images centrally. You push images to and pull images from the Docker Registry. A Docker Registry lets team members share images and deploy them to testing, staging, and production environments. The Docker Registry is the core technology behind the Docker Hub. You can run your own registry, for instance, if you want to host your images privately.

Docker Hub

Item 6 in Figure 2-19 on page 58 shows the Docker Hub, which is a hosted registry. It is a cloud service that securely manages images. Docker Hub is used by several companies to host Docker containers. It is a public registry for sharing applications, and it has many containers from the community.

For more information about Docker Hub, see *Build, Ship, and Run Any App, Anywhere*:

<https://hub.docker.com/>

The containers are ready for reuse or use as base images for new containers. By default, Docker supports public images. A Docker Hub for private images is available as a paid service. Docker Hub also has images in the community, for instance, a Liberty container image that only IBM can change.

Docker CLI/API

Item 7 in Figure 2-19 on page 58 shows that the Docker CLI/API is available when Docker is installed on a developer's workstation. For more information, see the following websites:

- ▶ To install Docker on Linux, see *Install Docker Engine*:
<https://docs.docker.com/installation/>
- ▶ For Windows, you can use an option to install by using Docker Toolbox. See *Install Docker for Windows*:
https://docs.docker.com/windows/step_one/

Several basic commands are listed:

- ▶ Docker **login** logs in to a Docker Registry server. If you do not specify a registry server, <https://index.docker.io/v1/> will be used, as default.
- ▶ Docker **pull** pulls an image from the registry.
- ▶ Docker **run** creates a writable container layer over the specified image, and then starts it by using the specified command. You can publish ports by using **-p** switch, and you can mount a volume for data by using **-v** switch.

For more information about the Docker client API, see *Use the Docker command line* at the following website:

<https://docs.docker.com/reference/commandline/cli/>

Advantages of using Docker with PureApplication in a hybrid cloud

PureApplication Systems support the use of Docker containers in patterns on Intel and Linux platforms, and on PureApplication Software and PureApplication Service.

If the Docker pattern type is installed and enabled in the system, you can drop Docker containers from the Pattern Builder as software components onto the canvas of your virtual system patterns. You can reference Docker images that are stored in Docker Hub or in the private Docker Registry that runs on the PureApplication family. In the same way that you configure patterns, configure the high-level parameters, as shown on Figure 2-23.

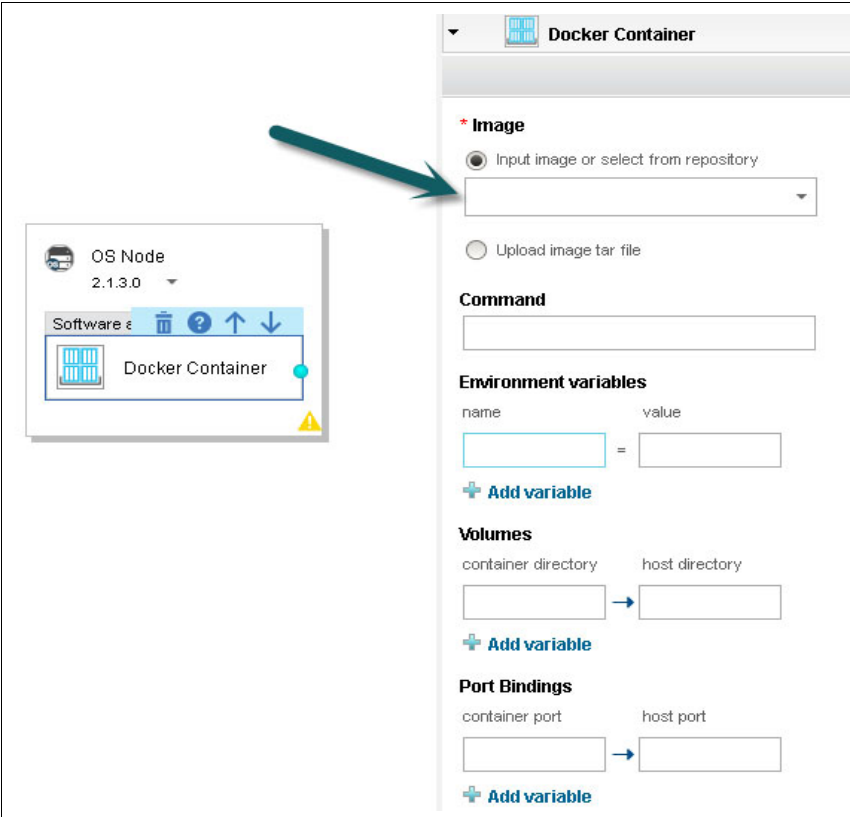


Figure 2-23 Docker container in Pattern Builder as a virtual system pattern

You can even mix containers with patterns to enhance your pattern-based solutions. In addition, IBM provides a Liberty Profile Container in the Docker community. Figure 2-24 shows a Liberty container with a database pattern.

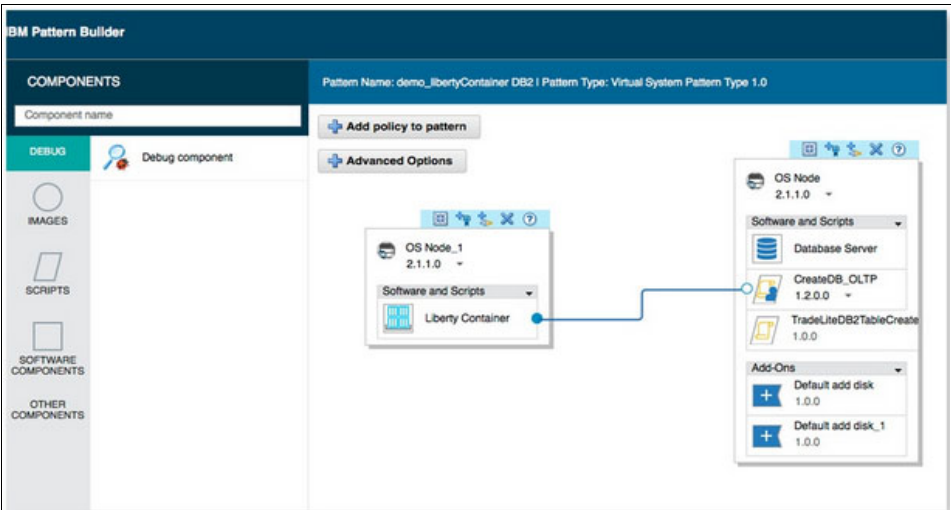


Figure 2-24 Liberty Container profile as a virtual system pattern

With the PureApplication family, you can create and deploy single and multiple container Docker applications, or a multiple node orchestration of Docker containers. In the Pattern Builder, you can connect containers across nodes by linking them. You can also use the PureApplication family to update container images and propagate the changes across containers, and to scale out the number of containers that are created.

In summary, Docker containers on the PureApplication family provide many advantages:

- ▶ Faster application deployment, startup, and scaling
- ▶ Higher-density deployments
- ▶ More efficient use of hardware
- ▶ More seamless workload movement in hybrid and borderless cloud scenarios
- ▶ Ability to access thousands of pre-built applications on Docker Hub
- ▶ Simplified configuration management
- ▶ More consistency across development, test, and production

Docker on the PureApplication family (Linux platform) provides significant business benefits with a focus on a faster time to deliver applications, seamless application portability, and higher confidence and reliability for enterprise applications. The result is an overall reduction in development time and cost. In a hybrid scenario, you can use Docker workloads on PureApplication System and PureApplication Service, for instance.

For more information, see the following websites:

- ▶ *Working with Docker* (for PureApplication System)
http://www.ibm.com/support/knowledgecenter/SSNLXH_2.1.1/doc/iwd/pac_docker_ov.dita
- ▶ *Working with Docker* (for PureApplication Software)
<https://ibm.biz/BdHhuv>
- ▶ *Working with Docker* (for PureApplication Service)
https://www.ibm.com/support/knowledgecenter/SSNS6R_2.1.1/doc/topics/pac_docker_ov.dita?lang=en

2.4.2 Chef

Chef is a popular open source product that is used for DevOps solutions for infrastructure automation and configuration management. You can use DevOps to automate the build and configuration steps for servers. By using Chef, you can perform these tasks:

- ▶ Configure from a base operational system
- ▶ Start the necessary system services
- ▶ Install and configure middleware and applications

Figure 2-25 shows an overview of Chef functionality with its basic components.

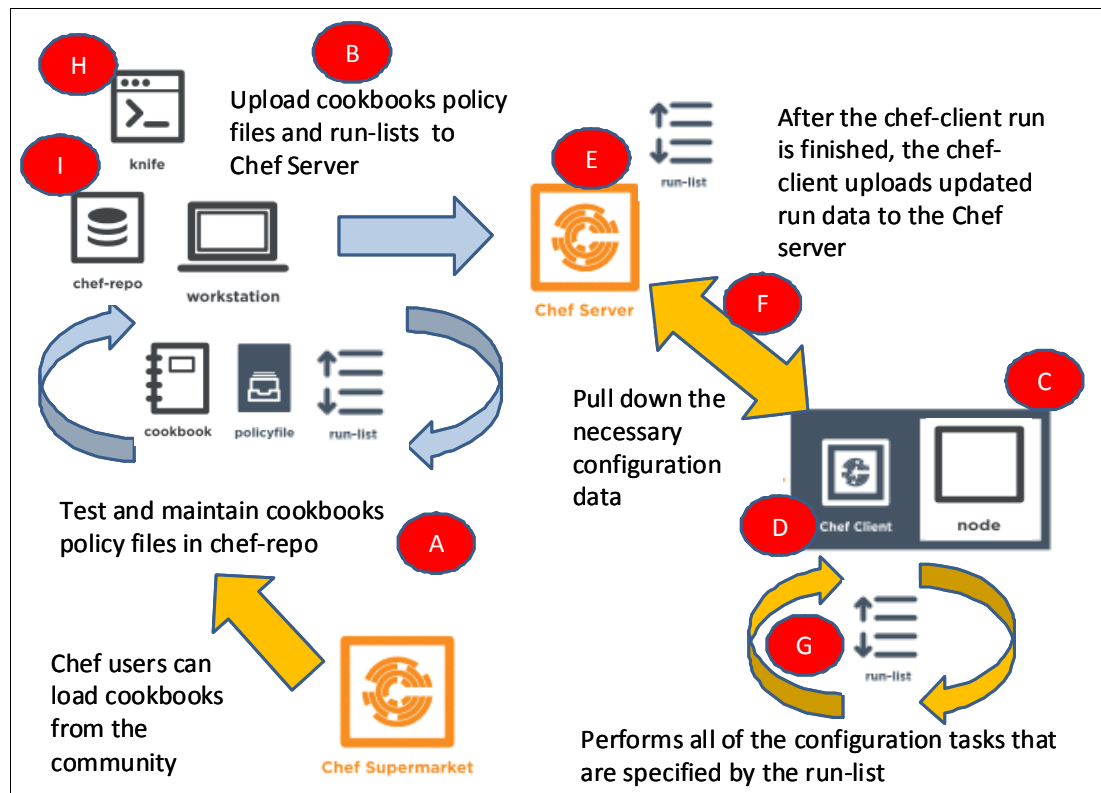


Figure 2-25 Basic Chef framework functionality

The functionality that is shown in Figure 2-25 is explained in the following sequence. The following letters correlate to the letters in Figure 2-25:

- A** One or more workstations can test and maintain cookbooks that are customized by the enterprise or from the community. Cookbooks are available in the Chef supermarket. Also, policy files can be maintained in the workstation.
- B** A workstation uploads cookbooks to the Chef server because the cookbook is the fundamental unit of configuration and policy distribution. One of the most fundamental configuration elements in a cookbook is called the *recipe*. A cookbook is a collection of related recipes.
- C** A node is any machine type (physical, virtual, cloud, network, or storage device) that is controlled by Chef.
- D** A Chef client is installed on every node that is managed by Chef.
- E** Chef servers act as an information hub. They contain cookbooks, recipes, and run-lists that describe how to define, provision, and configure application resources in Chef clients. Cookbooks, recipes, and policy settings can be uploaded to Chef servers from a developer's workstation. Policy settings for the server can be managed by the Chef management console.

F	The Chef client, on a configured node, accesses the Chef server. The client performs a search on historical chef-client run data, pulls necessary configuration data (including a run-list), and executes all configuration tasks from the run-list. When the Chef client finishes, the execution uploads the updated run data to the Chef server.
G	A run-list contains all run data information that is needed for Chef to configure a node (a Chef client) in an ordered list of rules and recipes.
H	Knife is a command-line tool from Chef that maintains run-lists that are updated from the workstation to the Chef server. Also, the Chef management console on the Chef server can maintain run-lists.
I	Chef-repo is a local repository where cookbooks (with recipes), policy files, and run-lists are maintained.

From this basic functionality, two models that you can deploy are available:

- ▶ The Chef client/server model where the Chef server is a central repository for managing configuration data.
- ▶ Chef solo is a simple way to use Chef where workstations can apply cookbooks to a local node for testing. Access to a Chef server is not needed.

Chef is an emergent open standard for deploying images, and it integrates with PureApplication System, PureApplication Software, and PureApplication Service. We provide two scenarios for deploying images in this section. The first scenario uses a client/server, and the second scenario uses client-solo to send Chef assets to a virtual system pattern.

Use case: Deploy an image by using a client server

This use case shows how to integrate an existing Chef infrastructure with PureApplication System by using virtual system patterns on a client/server.

Figure 2-26 shows the solution architecture to integrate PureApplication System and Chef with virtual system patterns.

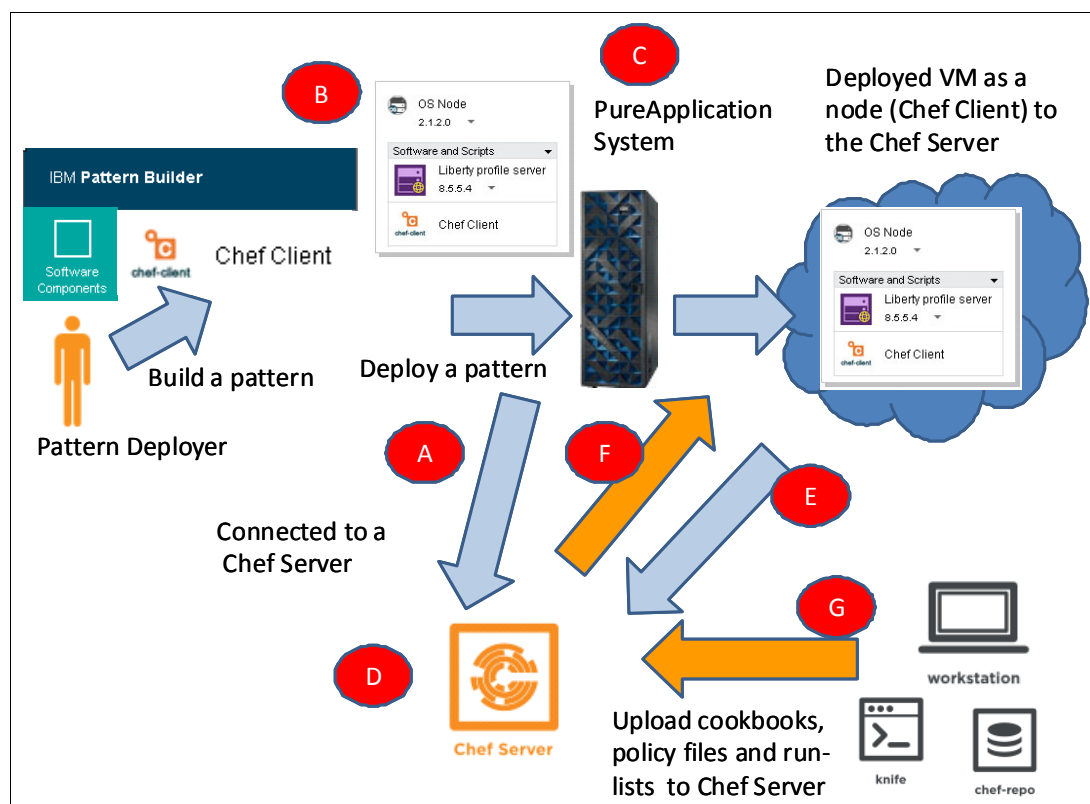


Figure 2-26 Solution to integrate PureApplication System and Chef with virtual system patterns

The solution architecture consists of the following sequence. The following letters correlate to the letters in Figure 2-26.

- A** From PureApplication, the system administrator configures an existing Chef server as an external shared service. The Chef server is deployed to a cloud group in PureApplication System.
- B** The pattern deployer configures Chef clients by using the Pattern Builder tool in a virtual system pattern deployment.
- C** Chef clients are configured in PureApplication System, PureApplication Software, or PureApplication Service in a virtual system pattern. Chef clients are considered nodes in the Chef framework.
- D** The Chef server acts as an external shared service to the PureApplication product, which contains cookbooks, recipes, and run-lists that describe how to define, provision, and configure application resources in Chef clients.
- E** In Chef framework, Chef clients periodically poll the Chef server for updates to the cookbooks or settings. If updates are available, the client pulls the content from the server. In the PureApplication products, you can specify how often the client needs to poll the server for updates.

- F** The Chef server sends the latest versions of the cookbooks and recipes to requesting clients. Each Chef client updates itself with the configuration information that it receives from the server.
- G** A Chef workstation is an external machine, for example, a developer's notebook, and uploads configurations to the Chef server. The Knife tool is regularly used to communicate with the nodes by using Secure Shell (SSH). You can use the Chef-repo structure to test configurations before you export them.

For more information about how to implement this use case, see *Integrating IBM PureApplication System into an Existing Data Center*, SG24-8285:

<http://www.redbooks.ibm.com/abstracts/sg248285.html>

Use case: Deploy an image by using Chef Solo

Transform a Chef asset to a script package by using the Chef Solo approach. Figure 2-27 shows the solution architecture to export Chef assets to PureApplication virtual system patterns. This approach can be used to migrate external Chef configurations from off-premises workloads on PureApplication System.

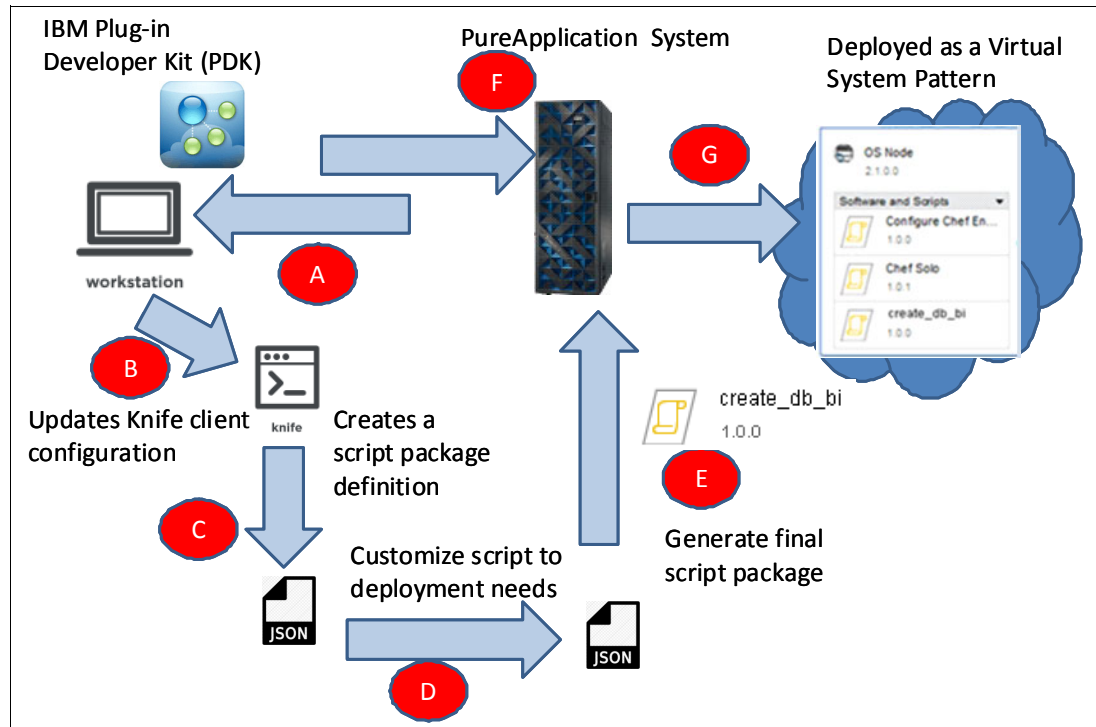


Figure 2-27 Solution architecture for exporting Chef assets to PureApplication virtual system patterns

The following steps explain the export of Chef assets. The following letters correlate to the letters in Figure 2-27 on page 66:

- A** From a developer's workstation, download the script package generator tool from the IBM Workload Plug-in Developer Kit (PDK) of the PureApplication family. The Knife tool is in the PDK in the Chef folder.
- B** Update the Knife client configuration to point to the target PureApplication System (target host name, user, and password).
- C** Generate a script package definition by using Knife.

- D** Customize the script definition file in the JavaScript Object Notation (JSON) format to the deployment that is needed.
- E** Generate the final script package with the script definition file, Chef standard cookbooks, and Chef scripts from the PDK. Upload the final script package to PureApplication System.
- F** Enter the type of Chef pattern that you are exporting to the PureApplication product, such as chef-1.0.0.0.tgz. The pattern type is available in the PDK.
- G** In Pattern Builder, drag the following script packages to a base OS image, and configure parameters on target environment:
 - Configure Chef Environment
 - Chef Solo
 - Custom Chef asset package

Inserting Chef assets into this script package scenario offers the following advantages:

- ▶ You can reuse existing Chef assets from the community (in the Chef Supermarket) on a public repository, or a Chef server that is in the enterprise domain as a private repository.
- ▶ You can build a complex distributed topology with Chef across VMs by using the pattern engine and Chef recipe execution.
- ▶ Pattern engine offers advantages, including auto-scaling, auto-failover, licensing, monitoring, and maintenance.

For more information about how to implement this use case, see *Integrating Chef with IBM PureApplication System, Part 1: Transforming Chef assets to script packages*:

<http://www.ibm.com/developerworks/cloud/library/cl-ps-aim1506-chef-pureappl/index.html>

The pattern engine can use Chef to deploy virtual system patterns in your DevOps solution. Patterns provide information about infrastructure automation, configuration management, and integration.

For more information, see the following websites:

- ▶ *Working with Chef* (PureApplication System W1500 2.1.1):
<https://ibm.biz/BdHhua>
- ▶ *Working with Chef* (PureApplication Software 2.1.1):
<https://ibm.biz/BdHhuG>
- ▶ *Working with Chef* (PureApplication Service 2.1.1):
<https://ibm.biz/BdHhun>

2.4.3 OpenStack (Heat and HOT)

OpenStack is an open source product for cloud operating systems. The current version is named *OpenStack Icehouse*. It provides a set of services for controlling cloud resources in a consistent manner across the data center.

For more information, see Chapter 1, “Architecture”, in the *OpenStack Installation Guide for Ubuntu*:

http://docs.openstack.org/icehouse/install-guide/install/apt/content/ch_overview.html

Heat and HOT overview

Heat, the OpenStack cloud core service, orchestrates multiple composite cloud applications by using the native Heat Orchestration Template (HOT) format by using both an OpenStack-native REST API and a CloudFormation-compatible Query API.

PureApplication support for OpenStack

You can use OpenStack services to communicate with PureApplication System by using IaaS REST APIs, as described on Figure 2-28.

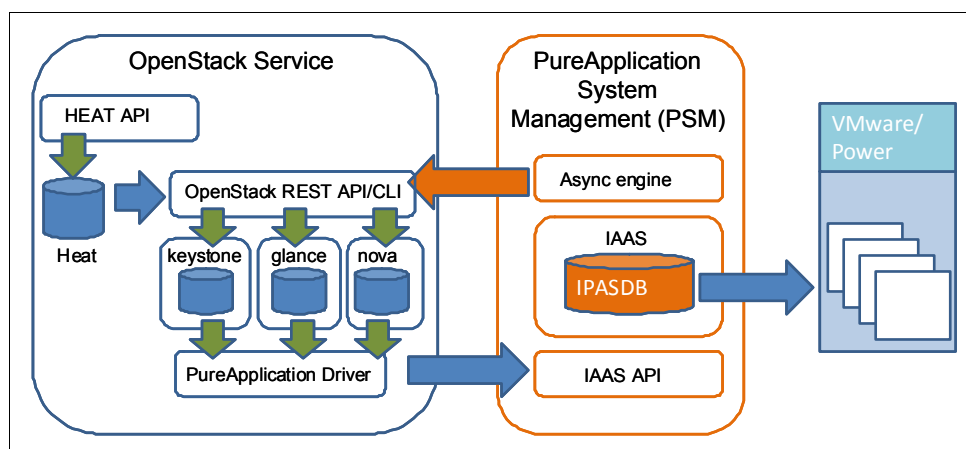


Figure 2-28 Interaction between PureApplication System and OpenStack

In Figure 2-28, an asynchronous engine component in PureApplication System interacts with OpenStack workloads by using REST APIs. The PureApplication driver, which is on the OpenStack services, intercepts requests from the OpenStack native drivers and forwards them to PureApplication System.

PureApplication System enables OpenStack services by supporting the OpenStack Icehouse version. Administrators need to enable the OpenStack service. OpenStack services are offered as a technology preview.

You can use OpenStack services on PureApplication System to deploy OpenStack workloads, improving portability across multiple cloud platforms. For instance, if you want to move from an off-premises workload to on premises (PureApplication System), and you have another cloud provider, you can deploy the workload to PureApplication System by using HOT templates.

To configure OpenStack services and deploy HOT templates to PureApplication System, see *Configuring the OpenStack services* at this website:

<https://ibm.biz/BdHVuv>

2.5 Achieving hybrid cloud application portability

Achieving cloud application portability is one of the most important requirements in a hybrid cloud. For instance, implementing the process of developing, testing, and running a pilot on a public cloud, then moving the application on premises, clearly requires cloud application portability. In this instance, you can start the development of an application in a public cloud by using an infrastructure that is provided by a cloud service provider. Then, you can move the application to a private cloud by using a potentially different infrastructure.

Cloud application portability is a key enabler for different reasons:

- ▶ Front end off premises, back end on premises: If your application management platform covers a hybrid cloud, you need portable patterns to deploy test and production environments to the hybrid cloud. In fact, you might have the same pattern definition at different locations. As examples, you might deploy one pattern on premises (test environment) and deploy the exported and imported pattern off premises (production).
- ▶ Quick delivery: You most likely develop your quick delivery application on premises. Then, you deploy it off premises.
- ▶ A small and medium business (SMB) starts small on a public cloud and expands as it grows. Extending your IT on premises might require that you migrate applications from a public cloud to a private cloud.
- ▶ Start on premises, then move to the hybrid cloud, which is similar to the SMB that starts small and expands. In this case, you might need to move applications from the private cloud to the public cloud.
- ▶ Primary on premises and business continuity off premises: Portable patterns make it possible to rebuild an infrastructure in a different location.

The purpose of cloud application portability is to facilitate moving an application from one cloud service provider to another cloud service provider. This capability can have a different meaning and a different effect, depending on the specific scenario:

- ▶ A division of a large multinational enterprise needs a customer relationship management (CRM) application for deployment to a private cloud. Another division in the same enterprise has this type of an application. Cloud application portability enables this application portability to occur.
- ▶ An enterprise can move an application from its private cloud to a public cloud, for instance, so that the application is available to users around the world by using geographically available cloud service provider data centers.

The second scenario is more complex because moving an existing application also means moving the workload that is associated with the application. You must consider the portability of the data that results from executing the application.

The portability of a cloud application affects your organization at multiple levels: the application level, the data, the platform, and the infrastructure.

Table 2-7 expands on these effects.

Table 2-7 Impacts of cloud application portability

Layer	Impacts
Application and data	<ul style="list-style-type: none">▶ The application directly supports business needs from a technical perspective. An organization can develop a business application internally, or buy it from an external software house.▶ Generally, the data is the information that is stored in a system. A business application usually creates, reads, updates, and deletes data.▶ Data portability is not trivial, and you might need to reformat the data, also.

Layer	Impacts
Platform	A <i>platform</i> is a composition of middleware (for example, application servers, IBM MQ servers, and database servers) that supports an application. All middleware might need contextual configurations to support the specific business application.
Operating systems	The operating systems allow the platform to run. Every operating system might need contextual configurations to support the specific business application.
Infrastructure	This layer includes systems, network, storage, and all necessary configurations to make the infrastructure functional.

So, what makes an application portable? The answer is patterns because they represent the foundation for portability. Virtual patterns make it possible to transform an application into an auto-consistent, coherent, and reusable asset. In these terms and by looking at the hybrid cloud, the patterns are the vehicle for transporting an application from off premises to on premises and vice versa.

A pattern is a blueprint for an application architecture. It captures the essence of an application architecture by using a formal descriptive language. In this way, the pattern preserves all of the information to enable portability. Briefly, two main aspects characterize an application, its topology and its behavior:

- *Topology* defines the architecture of an application. It might be a potentially complex matrix of interlocked components, where components can be VMs, virtual network devices, middleware, or application modules and data. The deployment, availability, and performance of an application strictly depend on the correctness of the wiring between components and on how the components are provisioned and maintained.
- *Behavior* defines the non-functional requirements of an application, in terms of specific constraints, key performance indicators (KPI), and service level objectives (SLOs). These requirements consist of policies, and the pattern engine uses them to map the resources correctly to provide optimization. Examples of common non-functional requirements are high availability and scalability.

In summary, patterns offer a way to automatically build the application architecture that you need to execute a cloud application. Patterns represent a different approach from the approach that is based on backup and restore activities, manual installation and configuration of software, and so forth.

PureApplication System and PureApplication Service implement the concept of patterns, as described on 2.3.1, “Pattern engine” on page 50.

2.5.1 Virtual patterns and portability

This section describes how patterns support cloud application portability. We describe the impact of portability on a sample cloud application that is based on the TradeLite product.

Figure 2-29 shows a hypothetical business application that uses TradeLite and two additional components. The topology of this application includes the following components. The numbers refer to the numbers in Figure 2-29:

- TradeLite as a front-end web application (1) that enables the user to trade stocks, view stock orders and prices, and so on
- A central database (2) that contains data, including stock and trading order data

- ▶ A back-office batch application (3) that works from the central database to perform statistical calculations about trading activities
- ▶ A back-office statistical report web application (4) that shows reports (charts and tables) that are generated by the back-end office batch application

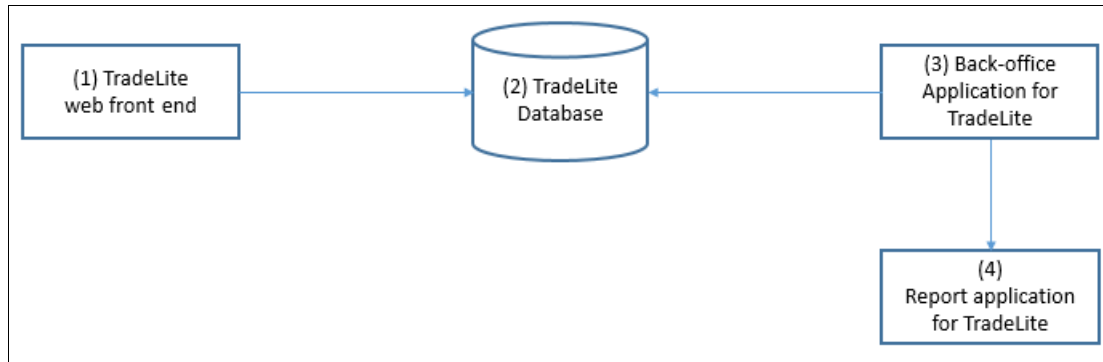


Figure 2-29 Logical architecture of the business application

Patterns capture the entire underlying architecture of the business application. However, how to use the patterns depends on the requirements of every architectural component. Table 2-8 describes the requirements of each component in the hypothetical business application. The numbers refer to the numbers in Figure 2-29.

Table 2-8 Architectural requirements of the business application

Component	Requirements
Web front end (1)	<ul style="list-style-type: none"> ▶ A Java Platform, Enterprise Edition (Java EE)-compliant application that runs on WebSphere Application Server ▶ Needs elastic scalability of the resources (application servers)
Central database (2)	<ul style="list-style-type: none"> ▶ Requires DB2 Enterprise Edition ▶ Needs the capability to deeply personalize the configuration of the database
Back-office batch application (3)	<ul style="list-style-type: none"> ▶ A Java batch application ▶ Packaged by the development team in a Docker container
Statistical reports web application (4)	<ul style="list-style-type: none"> ▶ A web application ▶ Provided as a black-box appliance and packaged in an open virtualization alliance (.OVA) file

For the requirements in Table 2-8 on page 71, the pattern for the business application can be a combination of the following pattern types:

- ▶ A virtual application pattern to design the TradeLite web front end (1)
- ▶ A virtual system pattern to design the central database (2) and the back-office batch application (3)
- ▶ A virtual appliance pattern to design the statistical report web application (4)

Figure 2-30 shows all three virtual patterns in one diagram.

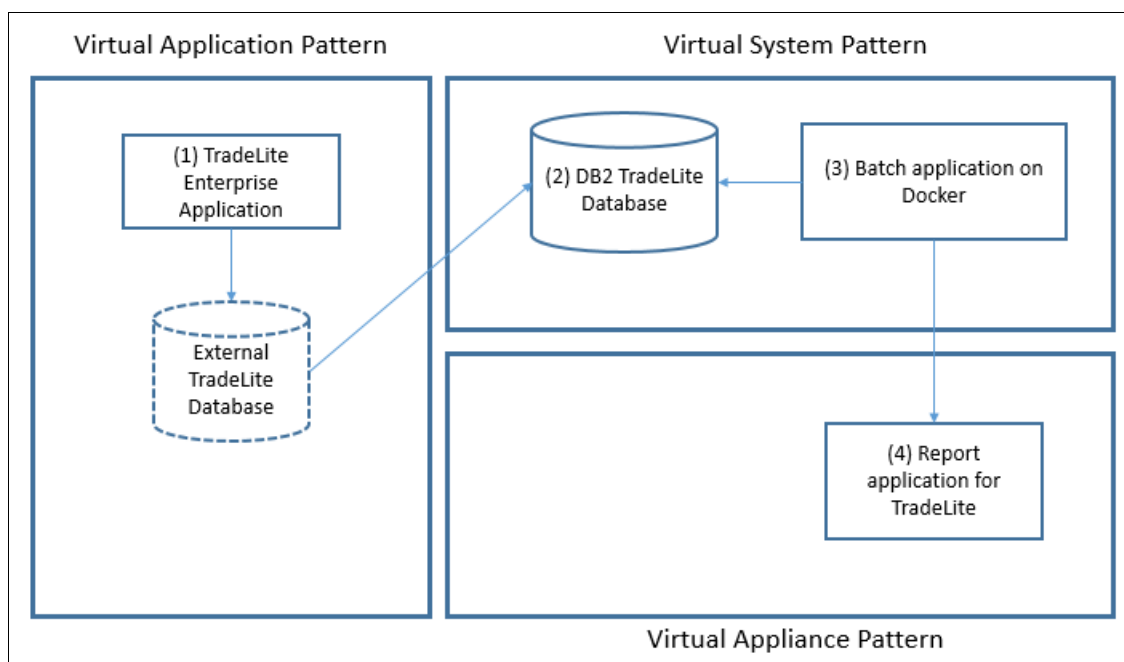


Figure 2-30 Physical architecture of the business application

The next section describes the following topics:

- ▶ How to build the virtual patterns that you want
- ▶ How to deploy them to a PureApplication System
- ▶ How to move them to a PureApplication Service

Building patterns on PureApplication System

Figure 2-31 shows the virtual patterns that you need on PureApplication System to design the architecture of the business application that is based on TradeLite.

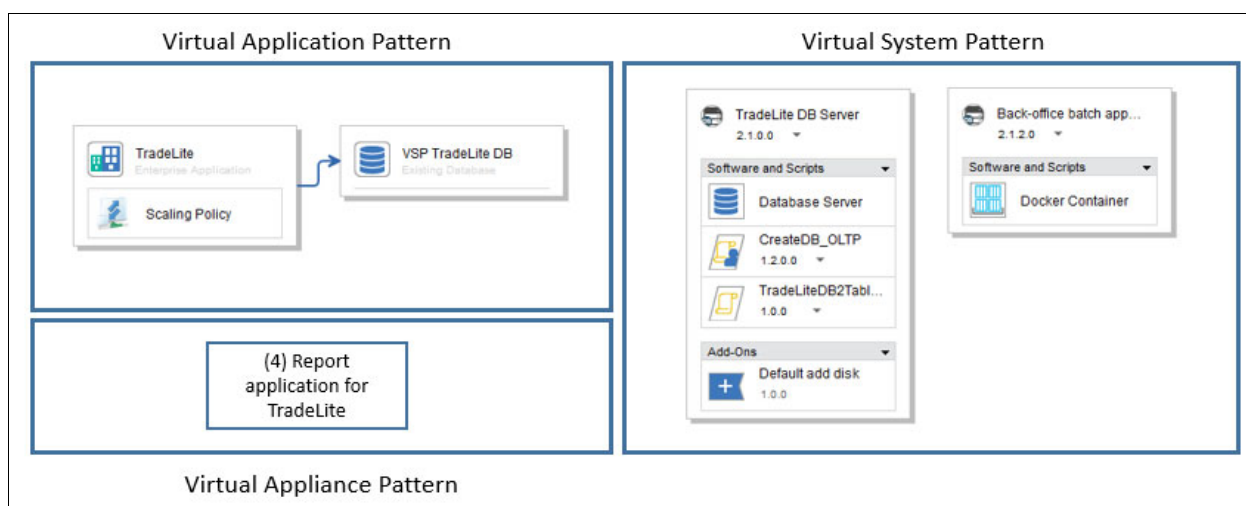


Figure 2-31 The virtual patterns to implement the business application

The *virtual application pattern* contains the enterprise application of the TradeLite web front end. In fact, the database component is merely a reference to the external DB2 system that is implemented by the virtual system pattern.

Use the application-centric approach that is offered by the virtual application pattern so that you can focus only on what you need. You need the enterprise application (to upload the code) and its behavior in terms of scalability (the scaling policy). The PureApplication System is responsible for building, executing, and monitoring the underlying infrastructure.

The *virtual system pattern* contains the DB2 system and the VM to host the Docker container for the back-office batch application. As shown in Figure 2-31, the DB2 system includes script packages and add-ons because they are necessary to execute additional configuration steps at deployment time.

You can use the system-centric approach that is offered by the virtual system pattern so that you can better control the infrastructure. You can choose the version of DB2 that you need, indicate how to handle the Docker container images, and so on.

The *virtual appliance pattern* contains only the VM to host the statistical report web application.

Figure 2-32 shows the workload for this business application after the deployment of the virtual patterns.

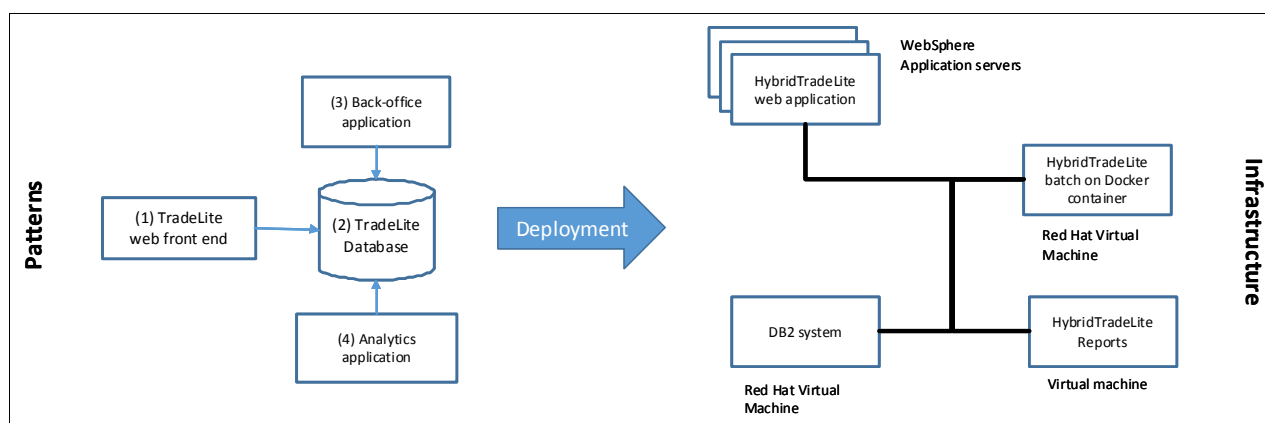


Figure 2-32 From patterns to workload

The portability of this business application from PureApplication System to PureApplication Service can have different effects:

- If you need to deploy a new instance of the business application on PureApplication Service by using the same pattern on PureApplication System, the portability is related to the pattern portability. You need to export the patterns from PureApplication System and import them into PureApplication Service.
- If you need to move the business application from PureApplication System to PureApplication Service, consider the impact of portability at the workload level, too.

Next, we provide more details about how portability affects patterns and workloads.

Portability of virtual patterns

The virtual patterns contain all of the information to make the cloud application portable. The virtual patterns are actual code. If you look at the virtual application pattern diagram in Figure 2-33, then click the **Source** tab, you see that the pattern is code that is defined by formal descriptive language.

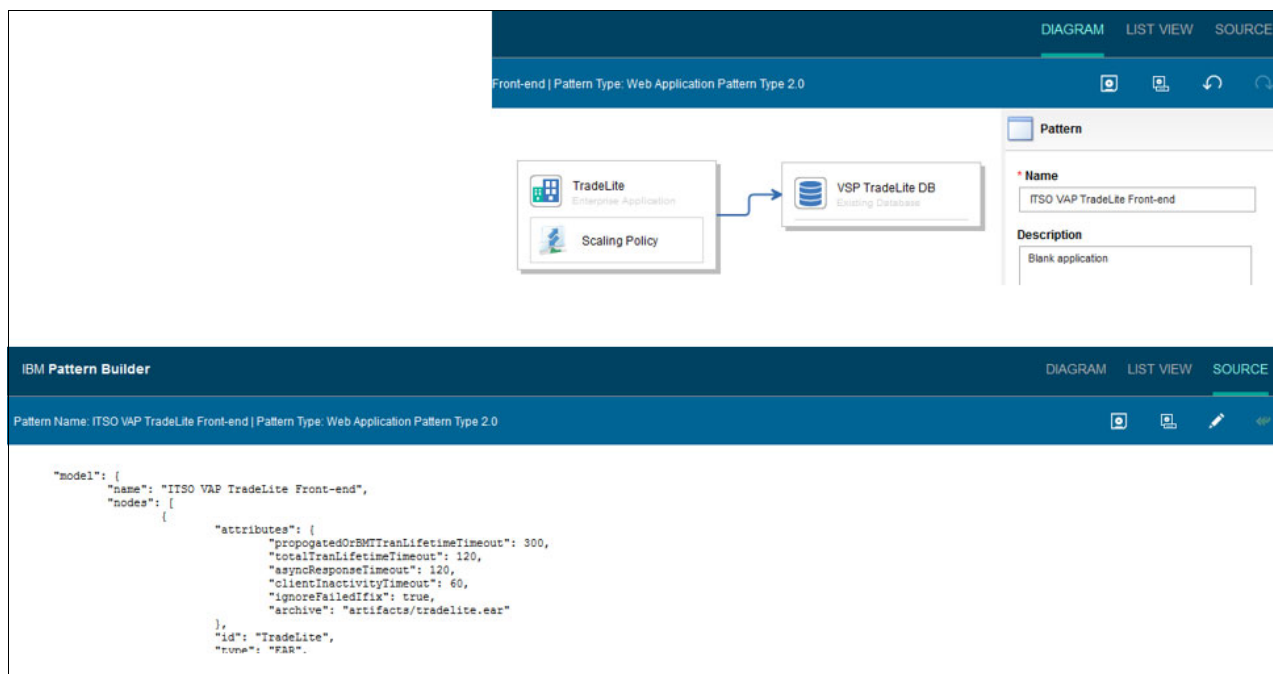


Figure 2-33 The virtual application pattern and related source code

That virtual patterns are code is a key point. This way, sharing patterns between PureApplication System and PureApplication Service is easy by using the import and export features.

From a portability perspective, the virtual patterns create an abstraction between who designs the patterns and the pattern engine that deploys the architecture and underlying infrastructure. The virtual pattern is a sort of contract. It contains what you need and what to expect from the pattern engine. However, the pattern engine needs to have all of the necessary artifacts to deploy a virtual pattern. The artifacts are collected into the PureApplication System (or the PureApplication Service) catalog. These artifacts can be the following objects:

- ▶ Operating system images (for instance, Red Hat Enterprise Edition)
- ▶ Middleware (for instance, IBM WebSphere Application Server 8.5.5)
- ▶ Script packages (for instance, the script to install the DB2 drivers into an instance of WebSphere Application Server)
- ▶ System plug-ins (for instance, the autoscaling feature)
- ▶ Add-ons (for instance, to add a disk to a VM)
- ▶ Predefined pattern types, such as the pattern types for IBM Business Process Manager (BPM) and other IBM middleware products

In summary, a pattern contains many references to artifacts from the catalog. Although, if you are moving a virtual pattern from one PureApplication System to an instance of PureApplication Service, you need to ensure that the system catalogs are aligned.

Figure 2-34 summarizes the need to keep the catalogs of PureApplication System and PureApplication Service in a hybrid cloud aligned.

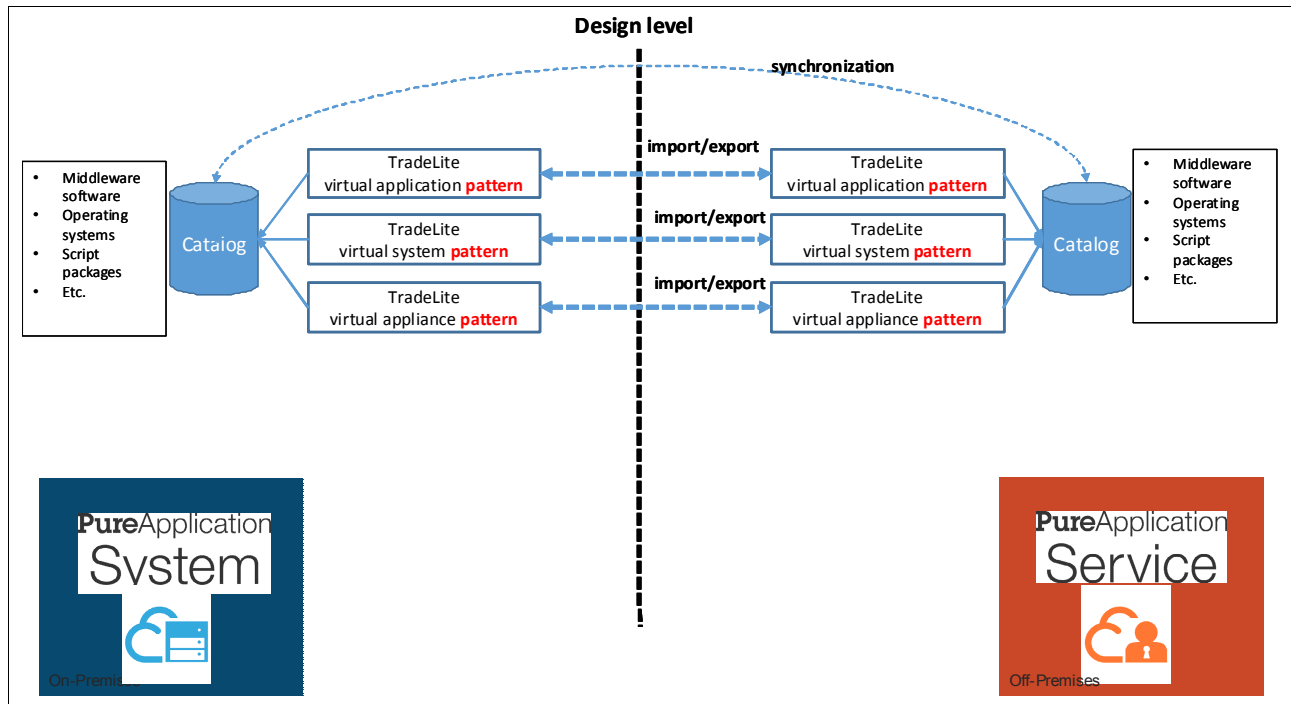


Figure 2-34 Sharing virtual patterns between PureApplication System and PureApplication Service

Through a use case, 4.4.1, “Backup and recovery in a hybrid cloud” on page 176 illustrates several implications of implementation. The description of this use case continues through to 4.6.1, “Test and development off premises and production on premises” on page 201, where we describe how to export patterns on PureApplication Service (off premises) and import them into PureApplication System (on premises) to illustrate what happens if the catalogs of both systems are not aligned.

To see the import and export features in action, watch the following video:

<https://www.youtube.com/watch?v=-s93Xx6wkx4>

Portability of workloads

In this section, the term *workload* is relative to the entire underlying infrastructure of a deployed business application.

Figure 2-32 on page 73 shows how, after the deployment of a TradeLite-based application, a workload is running on PureApplication System (or PureApplication Service):

- ▶ The TradeLite web front-end application is running on a dynamic cluster of WebSphere Application Server instances.
- ▶ Data is accessed by several applications.
- ▶ Several software components exist: WebSphere Application Server and DB2 Enterprise Edition.
- ▶ A Docker container hosts the TradeLite back-office batch application.
- ▶ Several VMs exist with potentially different operating systems.

Consider the implications of moving an entire workload from PureApplication System to PureApplication Service. The use of patterns can simplify sharing an architecture definition of a business application. You can share the patterns of a TradeLite-based application between the PureApplication System and PureApplication Service to deploy the identical workload (same architecture) to the PureApplication Service instance that you need to run the application.

As shown in Figure 2-35, moving a workload from PureApplication System to PureApplication Service creates implications that you need to consider:

- ▶ Moving data from on premises to off premises.
- ▶ If you applied changes to the workload after deployment (for instance, additional configurations to operating systems or middleware software), you need to move these changes to the off-premises site.

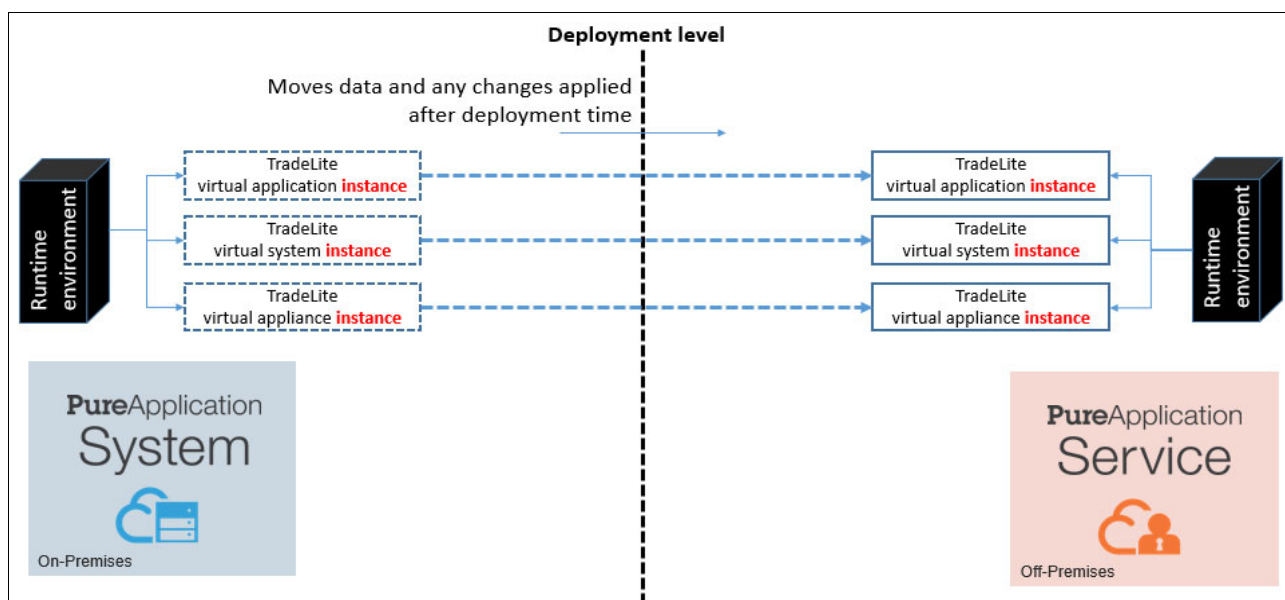


Figure 2-35 Moving the workload for the TradeLite-based application

Moving data is the main issue, which is true in traditional IT environments, too. Applying changes to a workload after it is deployed increases the difficulty of application portability. The patterns must contain all of the necessary configuration steps, simplifying workload portability so that you can focus on data portability.

What types of changes can you apply to running workloads? Thinking of our sample application and looking at Figure 2-30 on page 72, the following changes are possibilities:

- ▶ Make evolutive and maintenance changes to the TradeLite web application.
- ▶ Tune the DB2 system by modifying parameters.
- ▶ Apply fixes to the batch application on Docker.
- ▶ Release a new version of the Report application for TradeLite.

If you need to change the running workload, change the original patterns, which is the best approach. This way, you are sure that if you need to move your workload (or deploy it again), you must redeploy the corresponding patterns.

Changing the patterns can require different types of actions. For example, the following actions might be required in our sample scenario:

- ▶ If you changed the TradeLite web application and this application is a Java EE application, you must update the application pattern by uploading the last release of the enterprise archive file.
- ▶ The TradeLite DB2 system is a VM in a system pattern. Script packages and add-ons help you to execute any type of action at deployment time. If you tuned the DB2 system by changing parameters at run time, you must update the script packages (or create a new script package) that are attached to the DB2 component in the pattern.
- ▶ The TradeLite batch application runs on top of a Docker container. If you change the batch application (for instance, by applying fixes to the code), you must update and commit the changes into your Docker image. You can upload the image to the system pattern or to a private Docker Registry.
- ▶ The report application for TradeLite is packaged in an appliance, so you must use the updated .OVA image file at deployment.

For a description of the configuration of patterns after deployment, see 2.4.2, “Chef” on page 62. We explain the integration between patterns and Chef.

For our sample scenario, Table 2-9 summarizes the assumptions and effects of workload portability for the TradeLite-based workload.

Table 2-9 Workload portability effects on a TradeLite-based application

Layer	Impacts on workload portability
Application and data	<ul style="list-style-type: none"> ▶ Virtual application patterns provide an application-centric approach that simplifies both pattern portability and workload portability. They demand the handling of the complexity of the underlying infrastructure of the pattern, which is the pattern engine. ▶ The pattern contains all of the information for building the correct architecture on PureApplication System (and PureApplication Service). No additional configuration steps are required after deployment. ▶ For data portability, after the deployment of a TradeLite-based application on PureApplication System (and PureApplication Service), you need to move data only from DB2 on premises to the new DB2 off premises.
Platform	The pattern contains all of the information for building the same architecture of TradeLite on PureApplication System and PureApplication Service.
Operating systems	The pattern engine of PureApplication System (and PureApplication Service) implements the infrastructure of the involved patterns.
Infrastructure	

In our scenario, we deployed the TradeLite patterns into the PureApplication System (the private side of our hybrid cloud). We did not apply any changes to the workload after the deployment. This way, we know that by deploying the TradeLite patterns into the PureApplication Service (the public side of our hypothetical hybrid cloud), the workload is identical (same architecture) to the PureApplication System.

We need to address the issue of data portability, as stated in Table 2-9 on page 77. The TradeLite applications create, read, update, and delete data in the DB2 system. One more necessary step is required to move the TradeLite-based workload from PureApplication System to PureApplication Service (or vice versa) after the deployment of the patterns off premises. Copy the data from DB2 on premises and import the data into the new DB2 off premises.

Figure 2-36 describes a migration plan for the workload.

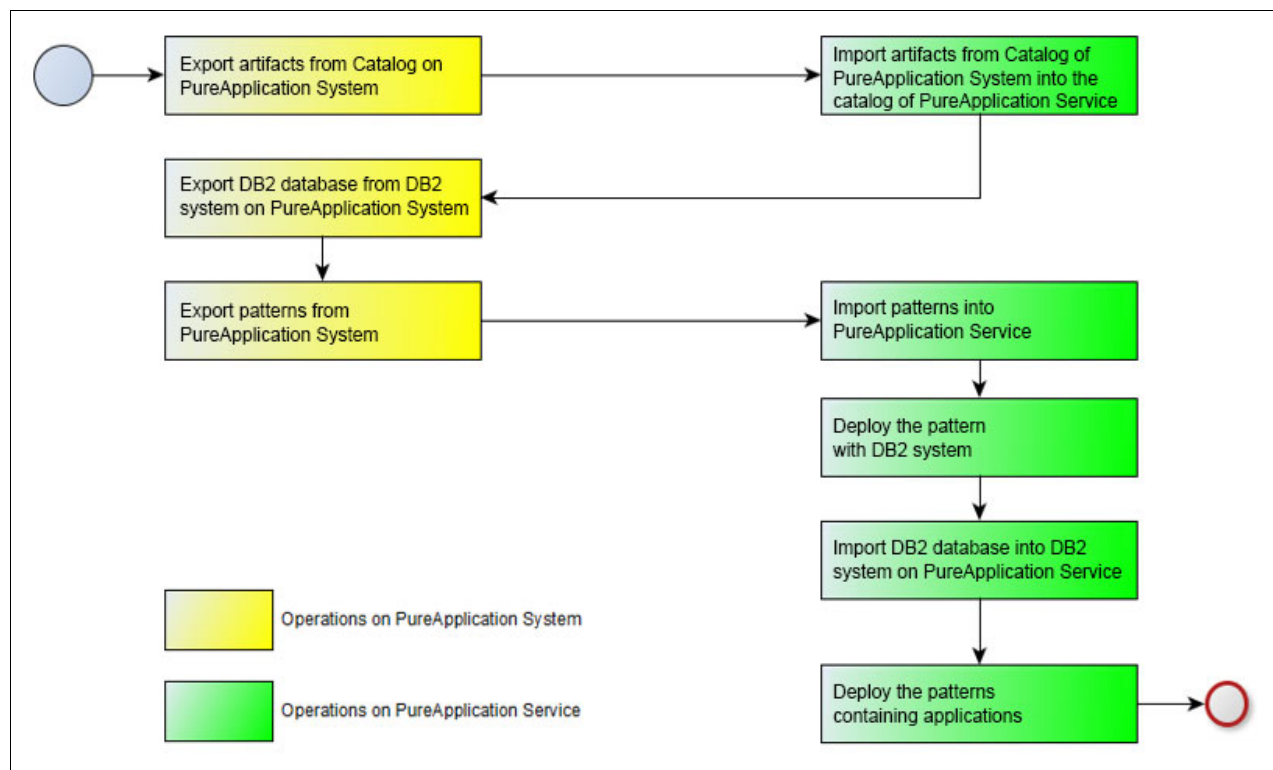


Figure 2-36 Workflow to move the TradeLite-based application to PureApplication Service

Pattern portability in a hybrid cloud

In “Portability of virtual patterns” on page 74, we described the importance of keeping the catalogs aligned in a hybrid cloud-based PureApplication System and PureApplication Service.

Alignment between PureApplication System and PureApplication Service is addressed by using the import and export features. These features can be used to align virtual patterns and artifacts in the catalog, such as script packages and virtual images.

The alignment between PureApplication System and PureApplication Service requires different actions:

- ▶ Virtual patterns can be shared between PureApplication System and PureApplication Service by using the import and export features.
- ▶ Artifacts in the catalog, such as script packages and virtual images, can be shared between PureApplication System and PureApplication Service by using the import and export features.

Docker images need to be centralized in a private (or public) Docker Registry. Use this approach to connect both PureApplication System and PureApplication Service to the same

repository of Docker images. This way, you do not need to upload the container image into a virtual system pattern. You can merely reference the image in the Docker Registry.

For appliance patterns, you need to upload an OVA file into the pattern. However, it is a good practice to have a central repository for all OVA files. To build OVA files, IBM provides the IBM Image Construction and Composition Tool (ICCT).

Use ICCT to build reusable, multiple purpose virtual images, in addition to OVA files. By using ICCT, you can define an OS and any additional software, providing the final virtual image.

In general, the development of cloud platform content, such as patterns, script packages, and VM images, involves different roles:

- **Pattern engineer:** Defines new virtual patterns and script packages, and determines how to change existing patterns.
- **Pattern developer:** By starting from the design of a pattern, as defined by the pattern engineer, the pattern developer creates new virtual patterns or modifies existing virtual patterns.
- **Asset librarian:** Manages the catalog of assets (images, plug-ins, script libraries, and patterns).

For more information about choosing the most effective practices for adopting a PureApplication System, see *IBM PureApplication System Best Practices*, SG24-8145:

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

Figure 2-37 assembles all of these roles into one view.

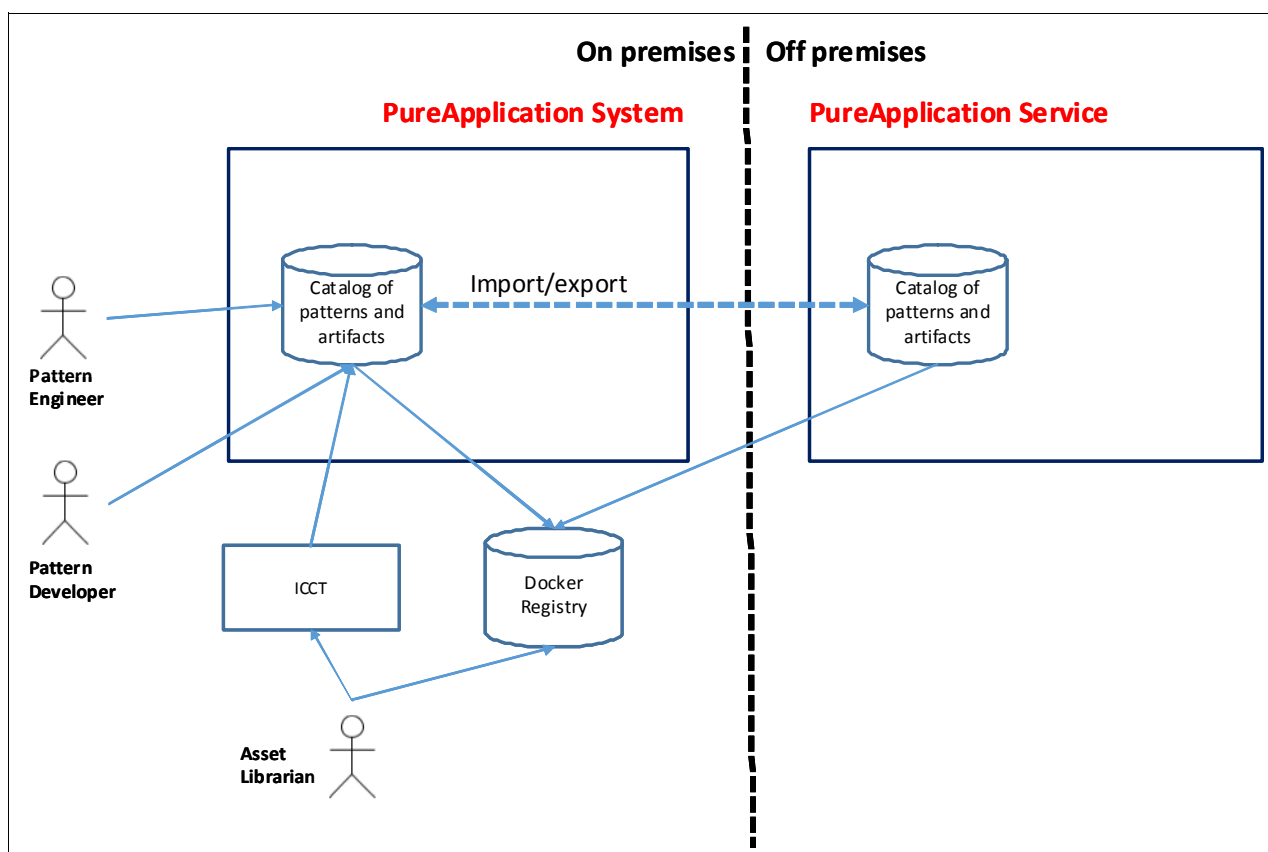


Figure 2-37 Roles for governance around patterns and artifacts

For more information about ICCT, see the following websites:

- ▶ *IBM Image Construction and Composition Tool: Creating Windows images for use in IBM PureApplication System:*
http://www.ibm.com/developerworks/websphere/techjournal/1309_hernandez/1309_hernandez.html
- ▶ *Using IBM Image Construction and Composition Tool:*
<http://www.ibm.com/developerworks/aix/library/au-aix-image-construction/>

2.5.2 Pattern portability with other open technologies

Pattern portability among cloud service providers, who adopt different technologies, is an important issue. Industry-wide standards are emerging to define an open specification for pattern portability. This section introduces two open specifications:

- ▶ Topology and Orchestration Specification for Cloud Applications (TOSCA): IBM, Red Hat, Cisco, Citrix, EMC, and other companies contribute to the TOSCA technical committee. TOSCA is part of the Organization for the Advancement of Structured Information Standards (OASIS).
- ▶ OpenStack HOT: The OpenStack cloud core service orchestrates multiple composite cloud applications by using the native Heat Orchestration Template (HOT) format.

Next, we describe these organizations in more detail.

Topology and orchestration specification for cloud applications

OASIS is a nonprofit consortium that drives the development, convergence, and adoption of open standards for the global information society. The consortium has more than 5,000 participants that represent over 600 organizations and individual members in more than 65 countries.¹

For more information about OASIS, see the OASIS website:

<https://www.oasis-open.org/org>

OASIS established the TOSCA technical committee, which consists of several companies, to define TOSCA. TOSCA is an open specification for enhancing the portability of cloud applications and services.

TOSCA uses a formal descriptive language to describe cloud applications, including their components, relationships, capabilities, dependencies, and requirements. Building a model of a cloud application in TOSCA involves the following tasks:

- ▶ Defining the topology in terms of interconnected components, where every component can be of a specific node type:
 - A *node type* describes a specific component, for instance, a web server, in terms of its attributes and capabilities.
 - The relationships between components have specific attributes.
- ▶ Adding possible deployment artifacts for every component, for instance, a configuration file for a web server.
- ▶ Adding plans, for instance, by using the Business Process Model and Notation (BPMN) standard to define orchestrations for an application.

¹ Source: <https://www.oasis-open.org/org>

Note: The model of a cloud application is packaged into a cloud service archive (CSAR) file. A TOSCA-compliant container can open and interpret a CSAR file and deploy the relative architecture. TOSCA enables cloud application portability across all cloud service providers that comply with the TOSCA specification.

IBM Cloud Orchestrator supports importing, deploying, and exporting service templates based on OASIS TOSCA.

For more information, see the following websites:

- ▶ *Enabling TOSCA support for IBM Cloud Orchestrator:*
<https://ibm.biz/BdHhZu>
- ▶ *OASIS Topology and Orchestration Specification for Cloud Applications (TOSCA) TC:*
https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=tosca

Heat Orchestration Template (HOT)

OpenStack, Heat, and HOT were introduced in 2.4.3, “OpenStack (Heat and HOT)” on page 67. HOT is a specification that describes the infrastructure of a cloud application by using a human readable, formal descriptive language and a simple test file to package the pattern.

Use HOT to define the infrastructure resources of a cloud application:

- ▶ Servers
- ▶ Volumes
- ▶ Security groups
- ▶ Relationships between resources (for instance, the connection between one volume and one server)

Heat uses the HOT model to implement and manage the infrastructure of a cloud application. Therefore, HOT packages are portable across all HOT-compliant cloud platforms.

IBM Cloud Orchestrator supports these components:

- ▶ OpenStack HOT
- ▶ OpenStack Heat stacks, which are called *Heat stacks* or *Stacks*, that are instances of deployed Heat templates

For more information, see the following websites:

- ▶ *Working with Heat templates and stacks:*
<https://ibm.biz/BdHhZ9>
- ▶ *IBM UrbanCode Deploy with Patterns supports the Hot templates:*
<https://ibm.biz/BdHhZC>
- ▶ *Environment Pattern Designer:*
<https://developer.ibm.com/urbandcode/products/urbandcode-deploy-with-patterns/features/environment-pattern-designer/>
- ▶ *Configuring the OpenStack services* (about the PureApplication family that provides the *OpenStack services* feature):
<https://ibm.biz/BdHhZQ>

2.6 Deploying applications by using a hybrid cloud

Cloud portability is an important consideration for building and deploying a hybrid application on a cloud. To reach this goal, the use of Representational State Transfer (REST) web services (or RESTful web services) is a key technology. PureApplication System, PureApplication Software, and PureApplication Service support RESTful APIs.

2.6.1 Using RESTful web services for application integration

RESTful web services are standard for integration. You can use these services to integrate various program languages and workloads that are based on REST technology.

REST standards are important for interoperability between applications. Older applications that are on premises can be enabled by RESTful web services, and they can communicate with cloud-centric applications and services off premises. This approach enables programming language transparency because each source point is unaware of the programming language on each endpoint, and because communication is only by the protocols that are supported by REST: Extensible Markup Language (XML) and JavaScript Object Notation (JSON).

JSON is a lightweight data-interchange format. JSON is easy for human beings to read and write. It is easy for machines to parse and generate. It is also a portable format between different programs on different platforms.

For more information about RESTful web services, see *RESTful Web services: The basics*:

<https://www.ibm.com/developerworks/library/ws-restful/>

2.6.2 PureApplication and Bluemix

IBM Bluemix is an implementation of the IBM Open Cloud Architecture, based on Cloud Foundry, an open source PaaS that you can use to quickly create and deploy applications on the cloud. Cloud Foundry is not vendor-specific, and it does not lock you into proprietary software or cloud infrastructure. You can use Bluemix to quickly create, deploy, and manage cloud applications by using a web interface. Bluemix runs on top of the SoftLayer infrastructure and also on PureApplication System (Bluemix Local).

Bluemix is a good solution for building front-end applications on a public cloud. The integration point with the PureApplication family is through Bluemix cloud integration services. You can use these services to integrate the front end with back-end services that are running on PureApplication System.

With Bluemix, developers can quickly create applications on the front end by using the enterprise-level power of a PureApplication System on a private cloud.

Figure 2-38 shows the Bluemix offerings.

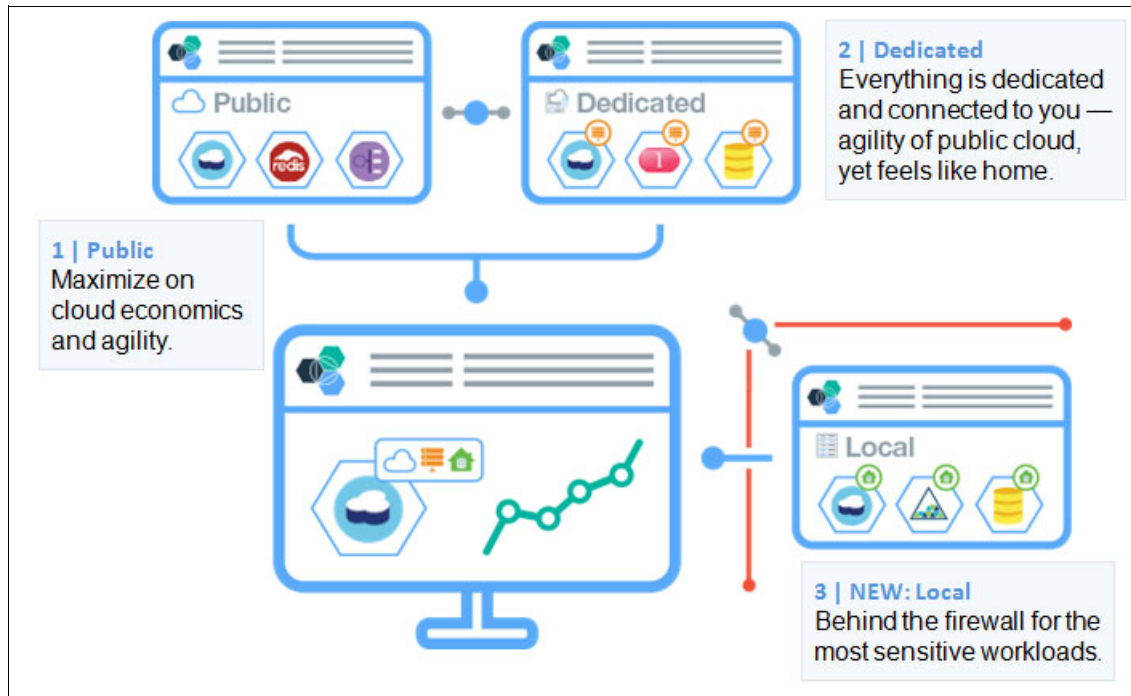


Figure 2-38 Bluemix offerings

Bluemix products include these components:

- ▶ Bluemix Public off premises for developing simple applications. Bluemix Public is dedicated to corporate development and interaction with business partners.
- ▶ Bluemix Local is used to develop more sensitive workloads, for example, those workloads that are associated with sensitive personal information. Bluemix Local runs on top of PureApplication System.

Another important integration is between applications that are deployed on the PureApplication family by using Bluemix services. Many Bluemix services are available, such as business analytics, Internet of Things (IoT), and Watson services.

For instance, Bluemix services can be consumed by PureApplication Service for use in workload patterns.

For more information, see the following websites:

- ▶ *What is IBM Bluemix?*
<http://www.ibm.com/developerworks/cloud/library/cl-bluemixfoundry/>
- ▶ *IBM Bluemix (services catalog):*
<https://console.ng.bluemix.net/catalog/>

2.6.3 Security at the application level

As an application developer, you must include security configurations, including application data protection, for your applications that run on PureApplication System, PureApplication Software and PureApplication Service.

You can use authentication capabilities, such as single sign-on (SSO) and Security Assertion Markup Language (SAML), which is an OASIS open standard for representing and exchanging user identity, authentication, and attribute information. SAML is fast becoming the technology of choice for providing cross-vendor SSO interoperability. These features can be set at the application level by using products, such as WebSphere Liberty Profile.

For example, in a hybrid cloud, a centralized SSO can be in two WebSphere cells, and it is configured in two different data centers. You can use PureApplication System, PureApplication Software, or PureApplication Service to host the two WebSphere cells. Figure 2-39 shows the use of SSO capability between on-premises and off-premises environments at the application level by using WebSphere Application Server.

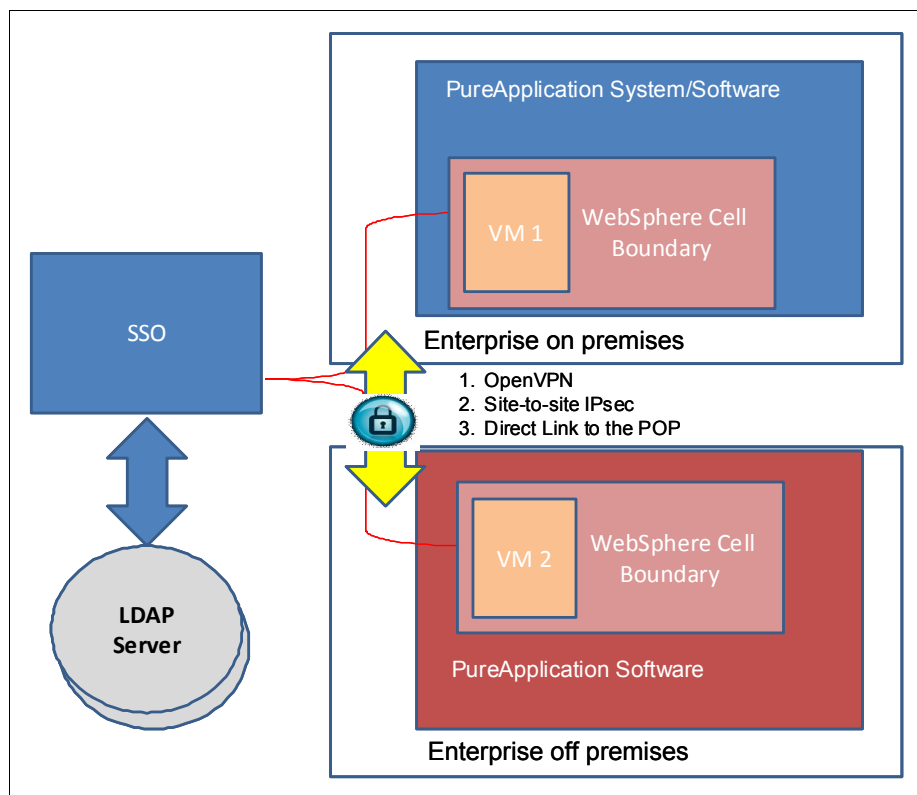


Figure 2-39 SSO between on-premises and off-premises environments at the application level

For example, within the same WebSphere cell, an on-premises environment with PureApplication System or PureApplication Software and an enterprise off-premises environment can exist. In this case, a WebSphere cell is on premises and another WebSphere cell is off premises, each with a central LDAP to configure the WebSphere security scope. For secure communications, we suggest the OpenVPN, site-to-site IPsec, or Direct Link to the POP option.

For more information about SSO capabilities in the cloud, see *Extend single sign-on to the cloud* at this website:

<http://www.ibm.com/developerworks/cloud/library/cl-singlesignoncloud/>

2.7 Achieving business continuity and high availability in a hybrid cloud

Business continuity refers to critical business functions and a pre-planned set of processes to perform in a disaster that affects business. Architecture and design systems that support business continuity are two of most complex issues in software engineering.

Disaster recovery (DR) is a pre-planned, tested, and tried set of processes that must be performed in a catastrophic event that brought several systems or an entire site down. The DR plan describes how to return systems, including websites, to use. If the primary systems cannot be returned to a usable state, the last good state of the existing systems is activated in a secondary location.

Users confuse DR with high availability (HA). DR is a standby infrastructure that is passive and not involved as part of the HA architecture. HA aims to ensure an agreed-to level of operational performance. Usually, DR does not have an identical mirrored capacity from production or HA environments as is used for the primary sites. The goal of a DR architecture is to function only for a limited period until the production systems are recovered, and for business or mission-critical applications only.

For the PureApplication family, a business continuity plan includes addressing the following key measurements for this service:

- ▶ *Recovery point objective (RPO)*: A measure of the maximum time period in which data might be lost from an IT service due to a major incident.
- ▶ *Recovery time objective (RTO)*: The duration of time within which a business process must be restored after a disaster (or disruption) to avoid unacceptable consequences that are associated with a break in business continuity.

Using these measurements, business units can define the business criticality of applications and the best strategy for addressing a disaster. For instance, if a business unit needs an RPO of 12 hours, two backups need to be taken each day, at a minimum. So, a business unit might need an RTO to be restored in minutes for critical applications or systems. For other applications or systems, an RTO of days is acceptable.

The business criticality will define the complexity and the required cost to support a disaster recovery architecture. Balancing RPO and RTO is a key factor in reaching a solution for recovering applications and systems, including workloads, in a disaster.

The architecture of your primary site must allow or include HA and exclude, for instance, a single point of failure (SPOF) elimination.

In the following sections, we describe several HA and DR principles, followed by the PureApplication family capabilities that can be used in hybrid business continuity solutions.

2.7.1 Principles of high availability

The goal of HA is to ensure a consistent level of operational performance even when certain physical or logical parts of your applications' architecture are unavailable. HA is a key point for business continuity and relies on a set of servers that are set up for redundancy, eliminating SPOFs.

Three principles of system design exist in HA engineering:

- ▶ Eliminate any SPOF by adding redundancy to the system so that the failure of one system component does not mean failure for the entire system.
- ▶ Ensure reliable crossover. HA engineering addresses the need for reliable crossover. For example, in multithreaded systems, the crossover point can become a SPOF.
- ▶ Detect failures. A user might not notice a combination of failures, for example, both a SPOF and an unreliable crossover. Therefore, maintenance activities are performed to prevent multiple failures.

When an organization strives for durability and uninterrupted operation without failure, the organization addresses these goals with a highly available system. With highly available systems, you might experience outages for a few seconds or minutes when a failover occurs. In calculating for HA, planned outages are not included. Unplanned outages are included.

HA needs to be considered both from the hardware and workload perspectives. For an overview of hardware and workload HA characteristics for the PureApplication family, see 2.7.3, “PureApplication family support for high availability and disaster recovery” on page 88.

2.7.2 Principles of disaster recovery

DR is a process for reproducing or recovering data center operations in a different data center, following a disaster. Three primary components exist in a DR plan, as shown in Figure 2-40.

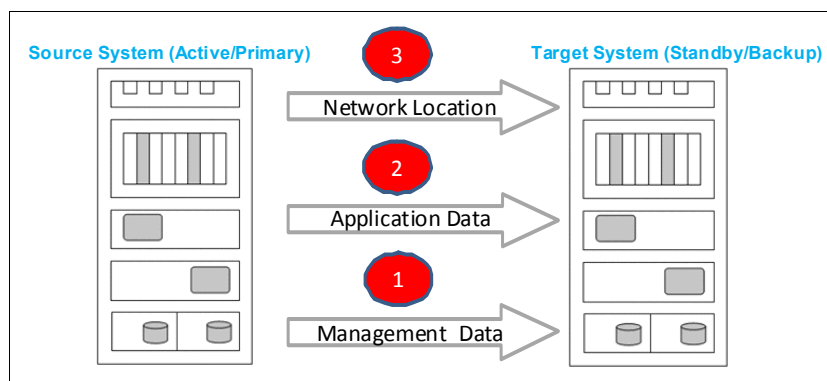


Figure 2-40 Main components of a DR plan

These main components are spread out in the following order:

1. Management data: Management data initially includes machine-specific parameters that relate to node configuration, such as cloud group configuration, virtual local area network (VLAN) ranges, and IP address ranges (IP groups).

All configurations for PureApplication System and PureApplication Software are stored locally in the storage of management nodes. These management nodes also contain all information about virtual system patterns, virtual application patterns, virtual images, pattern types, script packages, and plug-ins.

2. Application data: Application data represents the state of the databases, messages on IBM WebSphere MQ queues, configuration information in the WebSphere Application Server Deployment Manager, and transaction logs. Different clients and different applications within the client systems need different levels of recovery.

Simple applications, such as a search system, require only that other systems are available to accept new requests. More complex systems with more data can be available

after a disaster, such as certain retail client systems. For these systems, for which the latest set of catalog items must be available from the e-commerce system, the client system can then enable new customer traffic to search the catalogs.

At the other end of the DR strategy are systems that require as recent a state as possible, based on business criticality, such as banking with smaller RPOs. For those systems, a requirement might be to rebuild transactions that are occurring, and then capture and reconstruct. The restructure might be automatic (by recovering transaction logs) or manually (through a reconciliation process after the recovery is complete). For these requirement scenarios, it is typical for clients to use several types of recovery of program state information. Each type of recovery is based on client RPO and RTO.

3. Network configuration: Redirect network traffic from the primary system to the backup system.

Approaches for managing disaster recovery

In general, you can perform disaster recovery in a hybrid cloud, in which one site is on premises and another site is off premises. Arranged in order of decreasing RPO and (effectively) increasing cost, these disaster recovery strategies are listed:

- Backup and restore
- File, disk replication
- Shared file system

An RPO scale, in terms of a disaster recovery strategy, is explained in Figure 2-41.

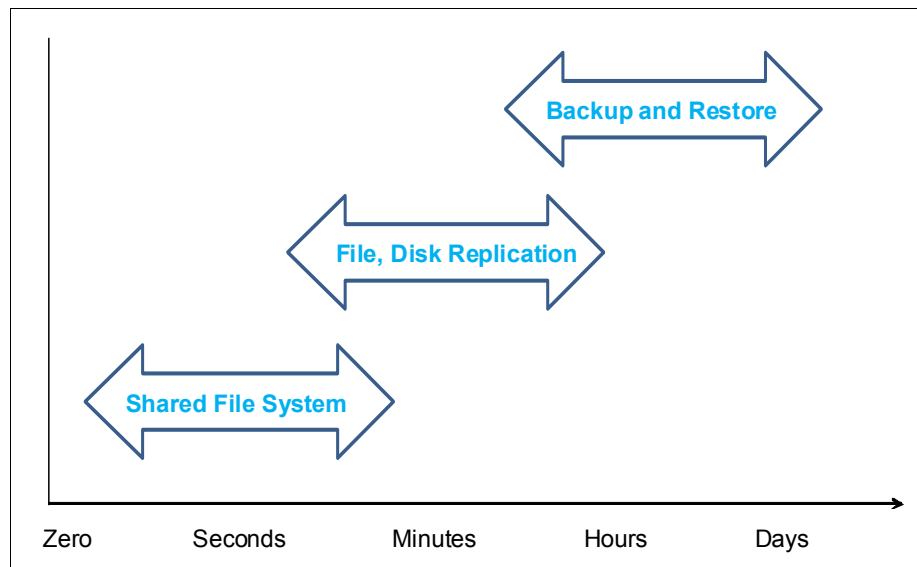


Figure 2-41 RPO scales in terms of time wasted and DR strategies

The shared file system option between a primary site and a recovery site is a faster option in terms of RPO time, followed by file disk replication. The backup and restore option takes the most time.

2.7.3 PureApplication family support for high availability and disaster recovery

PureApplication System and PureApplication Software include capabilities for implementing HA and DR strategies for your workloads. This section describes suggested business continuity solutions by using products in the PureApplication family. Think of PureApplication System and PureApplication Software as your toolbox for implementing these strategies at the application level.

PureApplication System hardware resiliency

Every PureApplication System is designed for HA. Its hardware is resilient. Compute nodes, network controllers, management nodes, virtualization nodes, storage controllers, and storage are all redundant and contribute to a highly available environment. The following list shows the component-level details that are available on a single rack system:

- ▶ Compute nodes: The management system automatically routes around failed cores. If an entire node fails, the system tries to move the VM to another compute node in its cloud group, if space is available.
- ▶ Network controllers: Cables and switches are redundant. The failure of one controller reduces bandwidth. However, service is continuous.
- ▶ Management nodes: Each node has a backup server. A floating IP address is assigned to the active management node (workload deployer).
- ▶ Virtualization nodes: Each node has a backup server.
- ▶ Storage controllers: Each controller has two canisters that service all traffic. If one canister fails, the other canister handles all traffic.
- ▶ Storage: Solid-state drive (SSD) storage and hard disk drive (HDD) storage are configured in a redundant array of independent disks with distributed parity (RAID 5) plus spares. Storage is designed to tolerate two concurrent failures without data loss (after the spares are in use).

High availability for the PureApplication family workloads

Several of the IBM software products that are implemented as patterns for the PureApplication family include built-in HA characteristics:

- ▶ A DB2 high availability disaster recovery (HADR) pattern
- ▶ WebSphere Application Server cell composition with Deployment Manager and built-in script packages
- ▶ Load balancers, such as the WebSphere Application Server On Demand Router and DataPower Gateway XI52 virtual appliance

IBM General Parallel File System (GPFS)

IBM General Parallel File System (GPFS™) is a scalable, high performance file management infrastructure for IBM AIX, Linux, and Windows Server systems. It is based on a shared disk model, which is provided by an underlying storage area network (SAN). GPFS provides fast, reliable access to data from multiple nodes in a cluster environment. Applications can readily access files by using standard file system interfaces, and the same file can be accessed concurrently from multiple nodes. GPFS is designed to provide HA through advanced clustering technologies, dynamic file system management, and data replication.

GPFS deployments and topologies

The IBM pattern for GPFS provides integrated management capabilities so that you can deploy many different file server topologies.

All deployments provide a GPFS manager on a VM that directs and controls the underlying GPFS cluster. From the GPFS manager VM and its application management capabilities, administrators can define file systems, add storage, add file system mirrors, manage security credentials, or apply recovery actions in a failure:

- ▶ GPFS server configurations

Several GPFS server configurations are supported, which you can easily scale to fit the needs of your environment.

- ▶ GPFS client configurations

The IBM pattern for GPFS provides integrated management capabilities so that you can deploy many different file servers.

For instance, In PureApplication System and PureApplication Software, GPFS allows multiple nodes or VMs to concurrently access the same data. GPFS provides a shared file system to support highly available configurations for virtual system and virtual application pattern deployments within the same rack, across racks, and even between a rack and PureApplication Software instance.

Figure 2-42 explains a scenario in which PureApplication A can be a PureApplication System, and PureApplication B can be an instance of either PureApplication System or PureApplication Software.

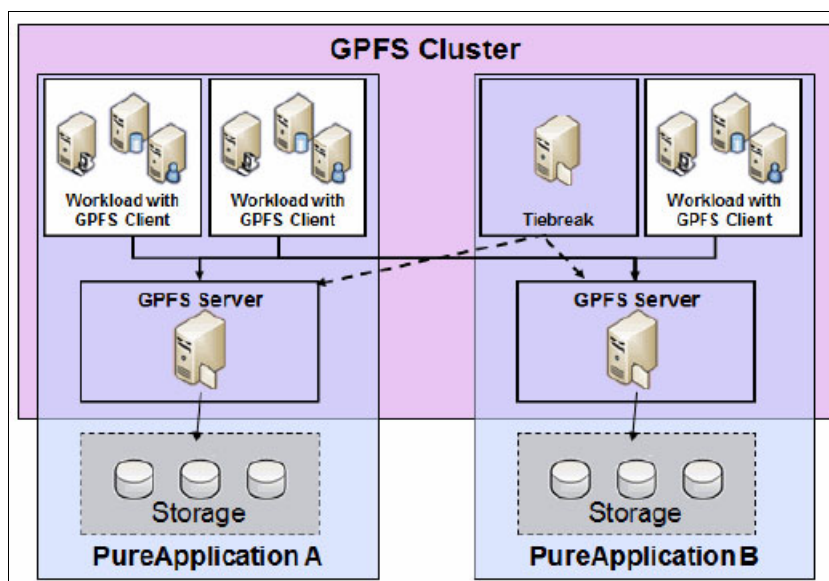


Figure 2-42 A scenario that describes the GPFS cluster feature

For more information, see the following websites:

- ▶ GPFS deployments and topologies (for PureApplication Software):

<https://ibm.biz/BdHhZ3>

- ▶ GPFS deployments and topologies (for PureApplication System):

<https://ibm.biz/BdHhZw>

Shared service for GPFS

To enable workloads to connect to a GPFS server and use shared file systems, the shared service for GPFS must be deployed. The shared service defines the parameters to connect to the GPFS server that connects to shared storage. As a result, any pattern that is deployed to a cloud group into which a shared service is deployed, can use the storage.

You can add a policy to a pattern component in a virtual application pattern or virtual system pattern to define it as a GPFS client. This policy is a GPFS Client Policy, which is supported in PureApplication System and PureApplication Software. When the GPFS Client Policy is added to a pattern component, the GPFS product is installed on the VM at deployment time. And, the configuration is retrieved from the shared service, and the GPFS client is connected to shared file systems that are hosted by the deployed GPFS server pattern.

For more information, see the following websites:

- ▶ *GPFS Client Policy* (For PureApplication System):
<https://ibm.biz/BdHhiA>
- ▶ *GPFS Client Policy* (For PureApplication Software):
<https://ibm.biz/BdHhiJ>
- ▶ *Implementing High Availability and Disaster Recovery in IBM PureApplication Systems V2*, SG24-8246:
<http://www.redbooks.ibm.com/abstracts/sg248246.html>
- ▶ *IBM shared service for GPFS* (for PureApplication Software):
<https://ibm.biz/BdHhiV>
- ▶ *IBM shared service for GPFS* (for PureApplication System):
<https://ibm.biz/BdHhih>

Block storage

A new type of storage volume, *block storage*, was introduced with PureApplication System. Block storage can be cloned on a single rack, or replicated to another rack for HADR solutions. Block storage maximizes storage controller logical unit numbers (LUNs), directly avoiding the VM file system (VMFS) and allowing more capabilities. Volumes can be internal volumes or defined in a block storage replication as a form of DR at the individual storage volume level. Rather than replicating the entire system configuration, you can configure replication of each storage volume between local and remote systems in either direction. Block storage replication removes the need for dedicated DR as an external storage device.

For instance, PureApplication Service supports block storage as a storage resource type and VMFS as a storage resource type. For more information, see *Administering storage resources*:

<https://ibm.biz/BdHhig>

Block storage replication

Available in PureApplication System and PureApplication Software, block storage replication is a form of DR at the individual storage volume level. Rather than replicating an entire system configuration, you can configure replication of each storage volume between local and remote systems in either direction. Block storage replication removes the need for a dedicated DR rack.

For more information, see the following resources:

- ▶ *Administering block storage replication* (for PureApplication Software):
<https://ibm.biz/BdHhIL>
- ▶ *Administering block storage replication* (for PureApplication System):
<https://ibm.biz/BdHhIC>

Block storage REST API

You can use the REST API to manage block storage.

Note: These APIs are intended to be used in the development of software components or script packages that consume block storage or shared block storage.

For more information, see *Block storage REST API* (for PureApplication Software):

<https://ibm.biz/BdHhZk>

External storage

You can attach your own storage to a PureApplication System rack or multiple racks, as shown in Figure 2-43. This concept is similar to connecting two networks through an Ethernet switch, but you connect drives instead of computers. Attaching external volumes to multiple racks allows HA across PureApplication System hardware.

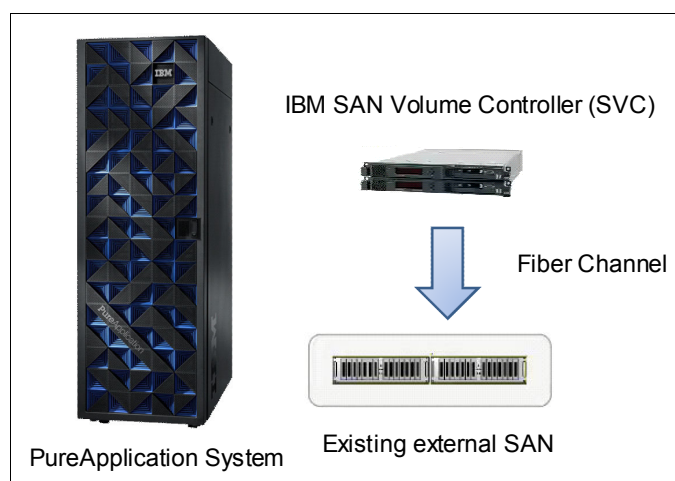


Figure 2-43 External SAN storage

When you use external storage devices, you can assign storage volumes to cloud groups as either block or block shared storage volume types. Block shared storage is available on Intel systems only. You provide the external SAN Volume Controller hardware. For a list of supported SAN Volume Controller hardware, see *Planning to use external storage*:

<https://ibm.biz/BdHhZt>

The PureApplication System provides the following connections:

- ▶ Fibre Channel connections for client storage only for Brocade switches
- ▶ Ethernet connections for SAN volume controller management to PureApplication System racks

Multisystem environment deployment

A *multisystem environment* consists of two or more systems that connect to each other physically (by networking) and logically (in a defined relationship). Systems in a multiple system environment can be in the same or different geographic locations.

PureApplication System provides continuous availability for key applications by deploying across multiple systems in a multiple system environment, as shown in Figure 2-44.

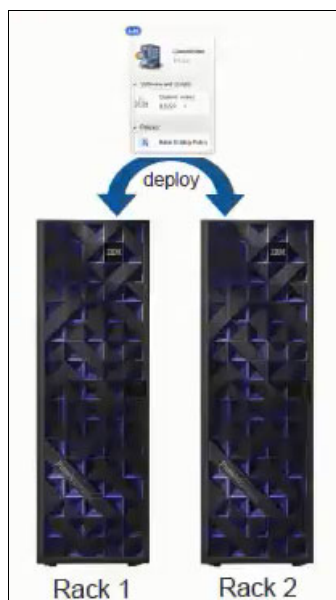


Figure 2-44 Multisystem deployment on multiple racks for use in the same or different geographies

A multiple system environment contains management domains and deployment subdomains. Systems that are part of a multisystem environment are also referred to as *locations*.

You can use a multisystem environment to perform many tasks:

- ▶ Deploying VMs across multiple locations in a region
- ▶ Preparing a consolidated view of pattern artifacts across the racks
- ▶ Building a pattern on any rack by using artifacts from all racks
- ▶ Deploying patterns across multiple locations in a deployment subdomain
- ▶ Preparing a single view to monitor the status of the deployed pattern across the racks
- ▶ Copying virtual system images from one system location to another system location within the management domain

Multisystem deployment provides considerable cost-efficiency through finer-grained replication on a workload-by-workload basis.

2.7.4 PureApplication System and PureApplication Software backup and restore capabilities

In PureApplication System and PureApplication Software, when you build a hybrid scenario, think about backup and restore capabilities both on premises and off premises. This capability is a key factor in business continuity.

PureApplication System backup and restore capabilities

PureApplication System capabilities provide a simple level of DR that you can use if your business continuity requirements are lenient, such as an RTO of 48 hours and an RPO of 24 hours. The general approach, which is outlined in the following list, fulfills the DR requirements that are shown in Figure 2-40 on page 86. The location that is defined for the backup must be a configured, SSH-accessible machine. You can also perform many of the tasks in this list by using REST API commands.

The following options fulfill the DR requirements:

- ▶ Use one rack for production and another rack for non-production (as standby or backup).
- ▶ To replicate management data, perform the following actions:
 - a. On the backup rack, configure a production cloud group for DR that includes IP groups. The IP groups will contain production IP addresses and VLANs.
 - b. The backup rack contains a single compute node.
 - c. The remainder of the backup rack is used for non-production work.
- ▶ Take application data backups of each pattern instance every 24 hours.
- ▶ To fail over the active rack to the backup rack, perform the following actions:
 - a. On the backup rack:
 - i. Store or delete workloads in the non-production cloud group to release resources.
 - ii. Acquiesce, stop, and remove compute nodes from the non-production cloud group.
 - iii. Add compute nodes to the Production cloud group.
 - iv. Deploy patterns for the workloads to be recovered.
 - v. Restore application data backups into the newly deployed workloads.
 - b. Redirect network traffic to make the workloads available, making the active workloads available on the backup.

PureApplication Service backup and restore capabilities

The backup and restore features in PureApplication Service include procedures for capturing the system configuration and helping you to re-create the cloud configuration, restore workload components, and restore application data.

The system backup includes management data that must be backed up, with several other categories of data that must be backed up. These categories include high-level data that is used for managing the whole system, including individual patterns and workloads.

Management data

The management functions control the whole system, virtualizing the hardware into resources for the cloud environment and providing a runtime environment for the workload functions. These management functions contain setup and configuration state data that must be backed up.

Cloud environment data

The cloud environment is based on the management data, and it is defined by creating and configuring cloud components, such as IP groups and cloud groups. It organizes system resources as runtime environments, into which the workloads can be deployed. This cloud environment data must be backed up separately from the system management data.

Workload catalog data

The workload catalog contains the individual patterns and their constituent parts as displayed in the console. Workload components are separate from the cloud environment. They include virtual images, virtual application patterns and pattern types, virtual system patterns, script packages, and plug-ins.

These pattern-level components do not include the deployed pattern instances, but the necessary patterns to deploy these instances again. The workload catalog also includes environment profiles, which are needed to deploy the patterns in the cloud environment. This workload data must be backed up by exporting it from the system.

After the workload catalog data is exported, you can use this data for multiple purposes other than restoring from the backup. You can store this data in a software configuration management (SCM) system to help manage the software development lifecycle of these components. You can also import this data into another PureApplication Service instance so that both instances are configured with the same patterns. You can also use this data in DR situations.

Workload data

Workload data includes deployed pattern instances, the contents of the running VMs, and their relationships. This data, when it is stored on the system after you deploy patterns, can be large for the following reasons:

- ▶ The VMs are large.
- ▶ The applications often load large amounts of data from the databases.
- ▶ The system maintains a significant amount of session data when many concurrent users are working.

When you follow standard middleware installation and application deployment practices, if the runtime environment is lost, re-creating it can be difficult, time-consuming, and error-prone. But with PureApplication Service, if an application environment fails, you can easily replace it by redeploying the pattern and redeploying the application in the pattern instance.

Application data

The internal state of each of the applications that is running as a workload on the system must be backed up. The state is application-specific and typically includes the application databases, but it can also include the application configuration, application file system, logs, and other key application artifacts.

This type of application data backup is always necessary for applications that are running in middleware on traditional hardware. However, for a virtualized application that is running in PureApplication Service, the pattern must include the backup software. Clients can add their own backup software to a pattern by using a plug-in or a script package.

Using the Idera continuous data protection agent

If you purchased Idera Continuous Data Protection (CDP) as part of your PureApplication Service environment for backup and restore, you can add a script package to your virtual system pattern to install the backup agent on your deployed VMs.

CDP is a high-performance, agent-based, backup service from Idera (formerly R1Soft) that you can use to back up workload data from your deployed VMs.

For more information, see the following websites:

- ▶ *Using the Idera Continuous Data Protection agent for backup and restore:*
<https://ibm.biz/BdHhZU>

- *Managing backup and restore* (for PureApplication Service):

<https://ibm.biz/BdHhZN>

Add your backup software

Users can add also your backup software by using scripts in your virtual system pattern. Moreover, a Red Hat Linux Satellite Server or IBM Endpoint Manager can be used also to configure your backup software.

2.7.5 Use case scenarios for high availability and disaster recovery in a hybrid cloud scenario

The following use cases are described on this section:

- DB2 HADR
- WebSphere Application Server cluster
- WebSphere MQ

These use cases refer to a HADR scenario, by using one or more instances of PureApplication System, PureApplication Service, or PureApplication Software, and the capabilities of DB2, WebSphere Application Server, and WebSphere MQ.

DB2 HADR

Figure 2-45 explores DB2 HA in active/standby mode and DB2 DR in active/passive mode that are deployed in a single rack for PureApplication System or your own hardware that is supported for PureApplication Software in the data center.

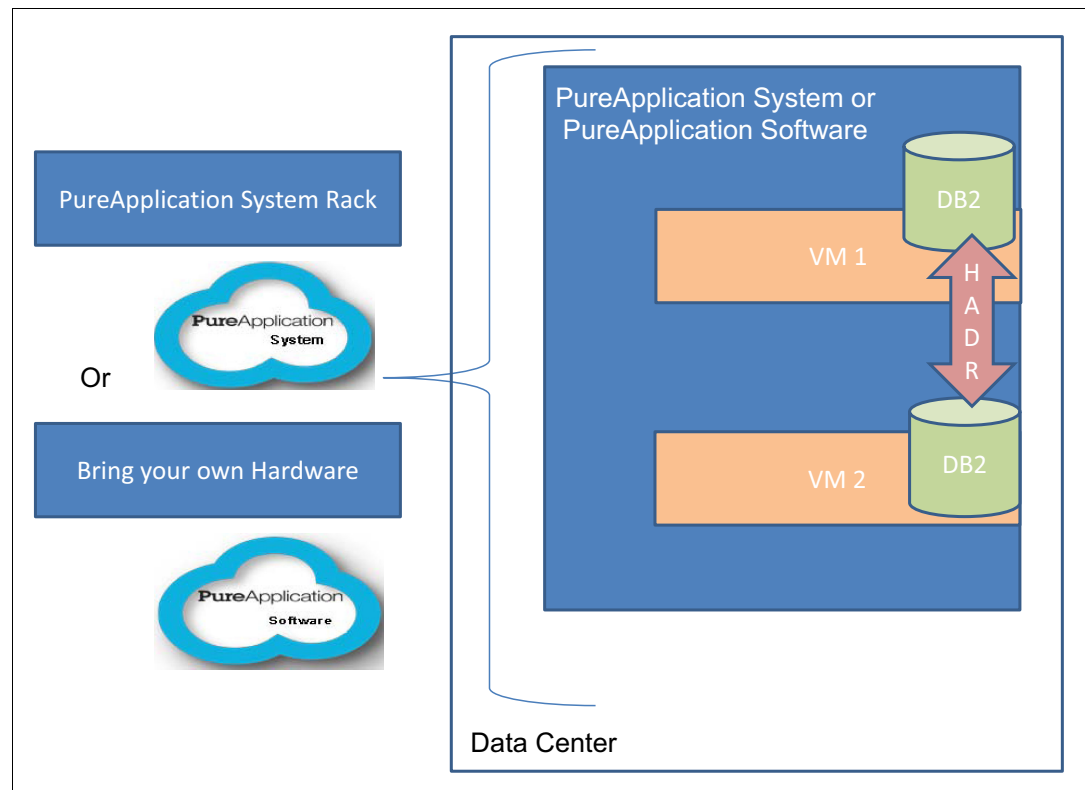


Figure 2-45 DB2 HADR that is deployed on PureApplication System or PureApplication Software

In this HADR scenario, PureApplication System and PureApplication Software help the DB2 product to allow a faster delivery by using virtual system and application patterns. However, HADR can be on DB2 middleware only. In this scenario, no HADR is on the hardware so the user cannot manage a PureApplication System rack failure, a hardware failure for PureApplication Software, or a data center failure.

The scenario, which is shown in Figure 2-46, addresses another HADR solution of hardware independency where DB2 is deployed across a multiple rack IBM PureApplication System in the same data center.

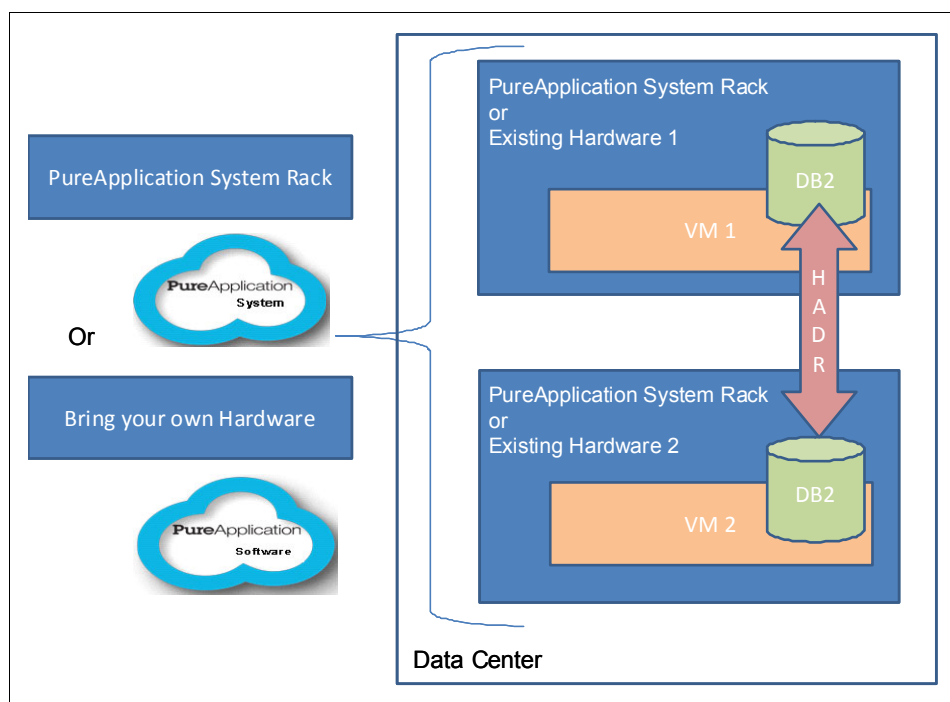


Figure 2-46 DB2 HADR setup across a multi-rack or multiple hardware setup in the same data center

Figure 2-46 shows that PureApplication System can address a single failure of DB2 (physical or virtual instance) on a single rack. However, PureApplication System cannot address a failure at the data center level.

In the last scenario, which is described on Figure 2-47, DB2 HADR is active in a primary data center, with a passive DR setup in the secondary data center. This scenario promotes a complete HADR configuration for DB2, PureApplication System, or PureApplication Software.

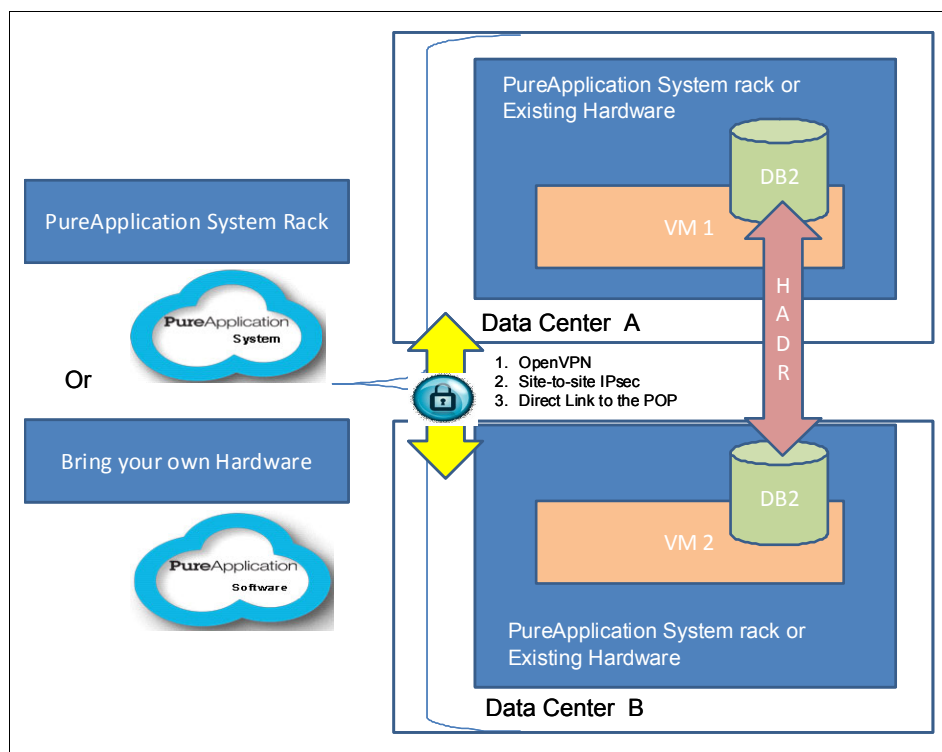


Figure 2-47 DB2 replication spread to two data centers

This scenario can address a DB2 instance failure, PureApplication System rack failure, your own hardware failure for PureApplication Software, and even data center failure. If the entire data center goes down, DB2 HADR instances can be activated in the secondary data center in a short time because the data was replicated from the primary data center to the secondary data center. To be secure, the connection between the two data centers uses OpenVPN or site-to-site IPsec, or it removes the Direct Link to the SoftLayer POP.

Note: The replication scope in this scenario is in the DB2 product.

WebSphere cluster

Figure 2-48 illustrates WebSphere horizontal or vertical clustering (VMs or WebSphere Java virtual machines (JVMs)) within one WebSphere cell on a PureApplication System or a PureApplication Software environment. The clustering relies on transaction logs that are stored on a GPFS file system and a database. Figure 2-48 shows the WebSphere cell boundary in a PureApplication System single rack or your own hardware for PureApplication Software.

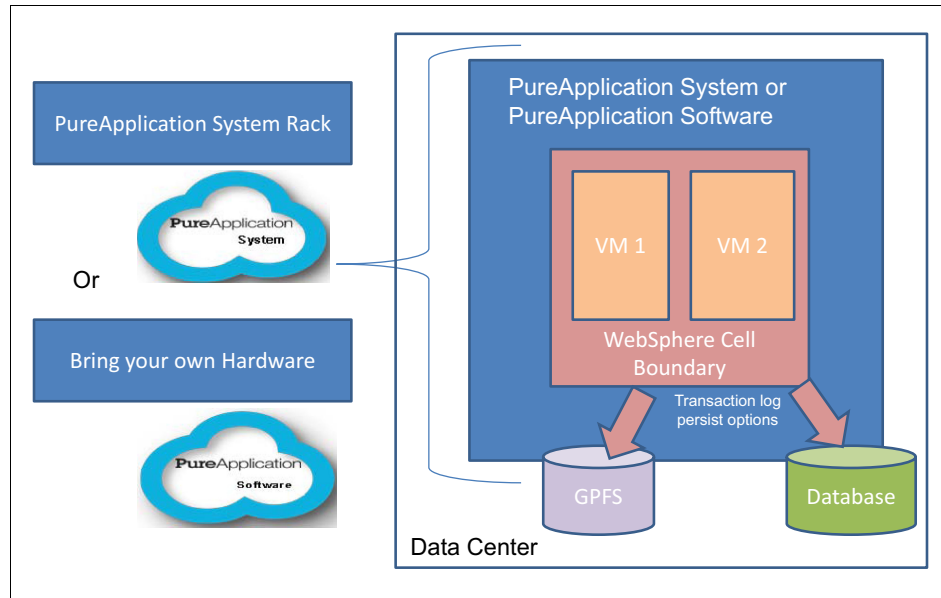


Figure 2-48 WebSphere cell boundary for either PureApplication System or PureApplication Software

This scenario in Figure 2-48 can address WebSphere node (WebSphere JVMs or VMs) failures with other nodes that handle the requests or recover the transactions. However, it cannot address the entire WebSphere cell failure, the rack failure (own hardware), or a data center failure.

Figure 2-49 describes a WebSphere Application Server cell across two PureApplication System racks or your own hardware for PureApplication Software in a unique data center.

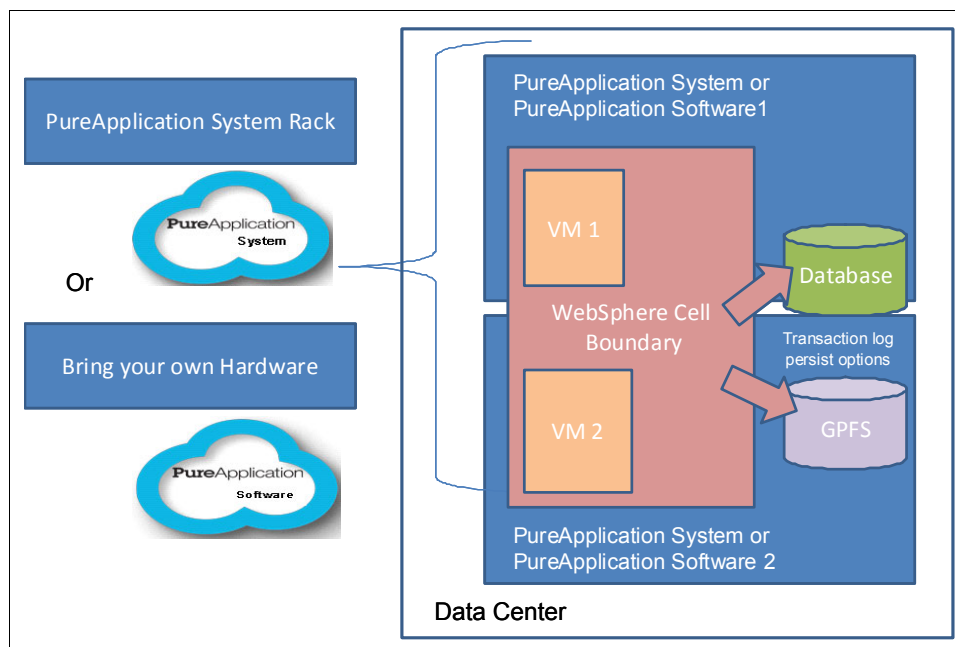


Figure 2-49 WebSphere cell across two PureApplication System racks in a single data center

In this scenario, transaction logs are stored in GPFS or a database. The environment can be a multiple rack deployment of a single pattern across two racks, or a manual catalog pattern synchronization between two PureApplication Software instances.

The scenario in Figure 2-49 can handle a WebSphere node failure (at the VM or at the JVM level) where other nodes take over handling the requests or recovering the transactions. This scenario can also handle PureApplication System or PureApplication Software failures. However, the setup that was used in this scenario cannot cope with an entire data center failure.

Figure 2-50 describes a less disruptive scenario in terms of HADR. Two identical WebSphere Application Server cells are across the primary data center and a secondary data center. This scenario includes a WebSphere Application Server active/passive setup. Transaction logs are stored in GPFS or on a database server that supports multiple cell topology. WebSphere Application Server supports multiple cell topology by using peer-cell topology, for example. To be secure, the connection with the two data centers can be OpenVPN, site-to-site IPsec, or Direct Link to the SoftLayer POP.

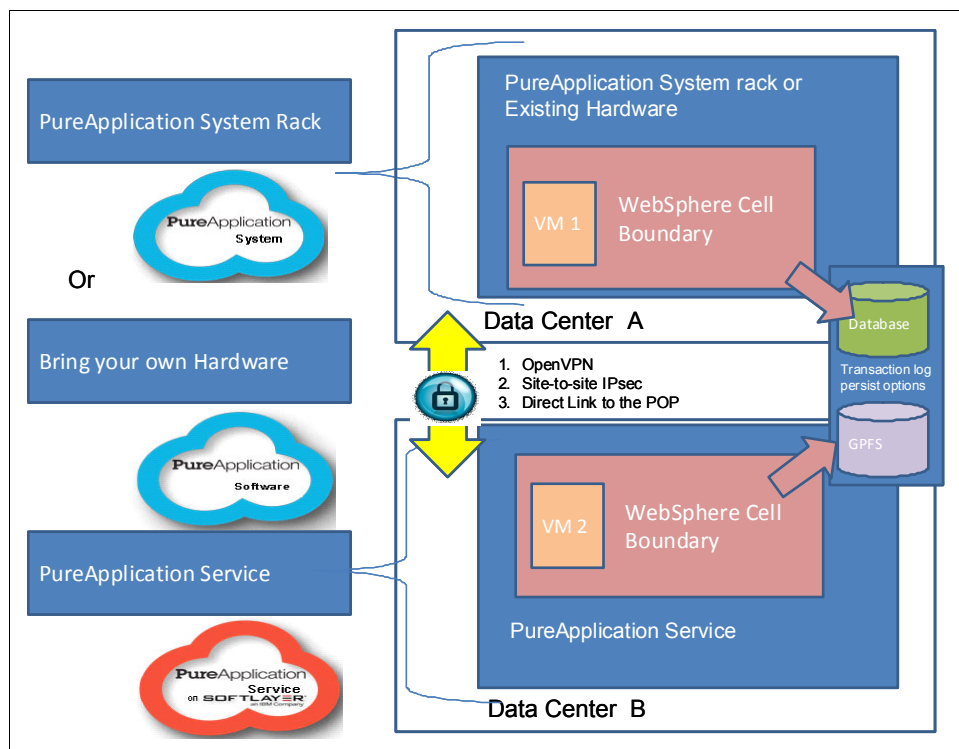


Figure 2-50 Two identical WebSphere Application Server cells on two data centers

This scenario can address a single WebSphere node (VM failure or WebSphere JVM failure), PureApplication System rack failure, failure of your own hardware for PureApplication Software, or data center failure. Based on the failure, if the entire data center is down, the WebSphere cell can be activated in the secondary data center in a short time because the data was replicated from the primary data center to the secondary data center.

In this scenario, PureApplication Service can be involved as Data Center B, for instance.

WebSphere MQ

The scenario for WebSphere MQ in Figure 2-51 explores two queue managers that are deployed in a PureApplication System single rack or your own hardware for PureApplication Software in the data center.

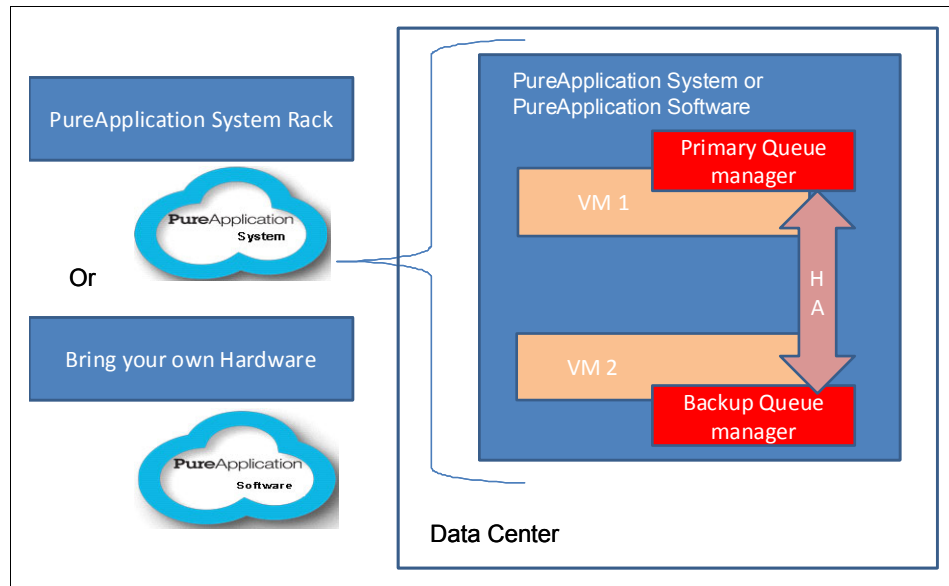


Figure 2-51 WebSphere MQ as two queue managers deployed on the same hardware

This scenario describes WebSphere MQ in active/standby mode in a primary data center. Replication in WebSphere MQ uses an OS replication feature, such as IBM High Availability Cluster Multi-Processing (IBM HACMP™) in AIX, for instance. This approach can address WebSphere MQ primary queue manager (QManager) failure (VM or WebSphere MQ processes) and failures with WebSphere MQ standby handling the messaging. However, it cannot address failures in the entire rack or data center.

Figure 2-52 describes two queue managers that are distributed on two PureApplication System racks, or a PureApplication Software instance that uses your own hardware.

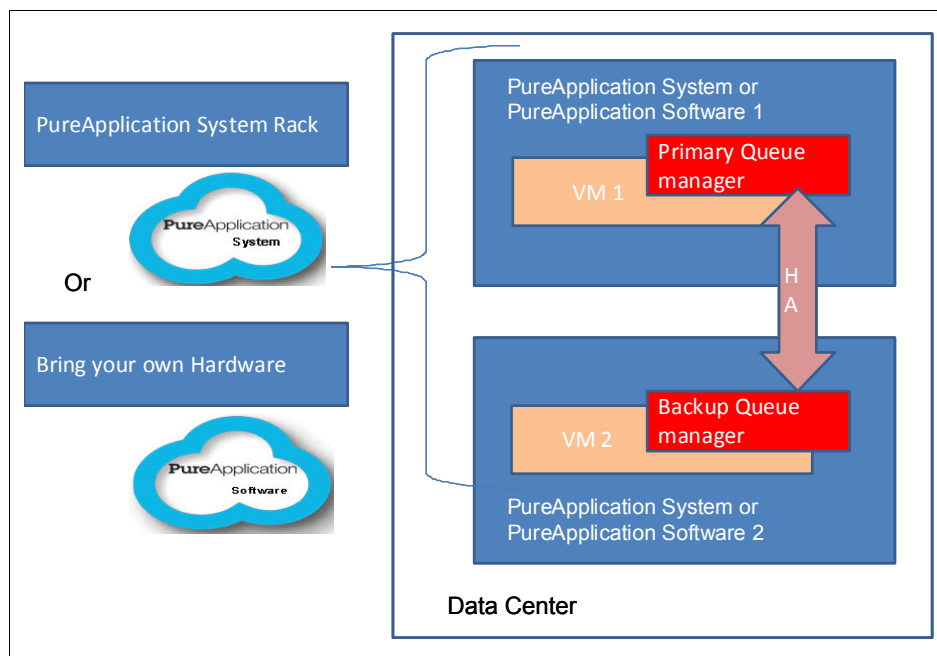


Figure 2-52 Two queue managers that are distributed on different hardware

Figure 2-52 shows active/standby mode in a primary data center. A split pattern for WebSphere MQ uses multisystem deployment for PureApplication System, or manual catalog pattern synchronization between two PureApplication Software instances.

This approach can address a WebSphere MQ primary queue manager (QManager) failure (VM or WebSphere MQ processes) and failures with WebSphere MQ standby handling the messaging. This approach can also address a rack failure. However, it cannot address entire rack or data center failures.

Figure 2-53 describes the last HADR scenario with two WebSphere MQ instances (primary and standby) with queue managers across two racks or two different hardware configurations in different data centers. WebSphere MQ with two queue managers in different data centers with PureApplication System racks or your own hardware for PureApplication Software are shown.

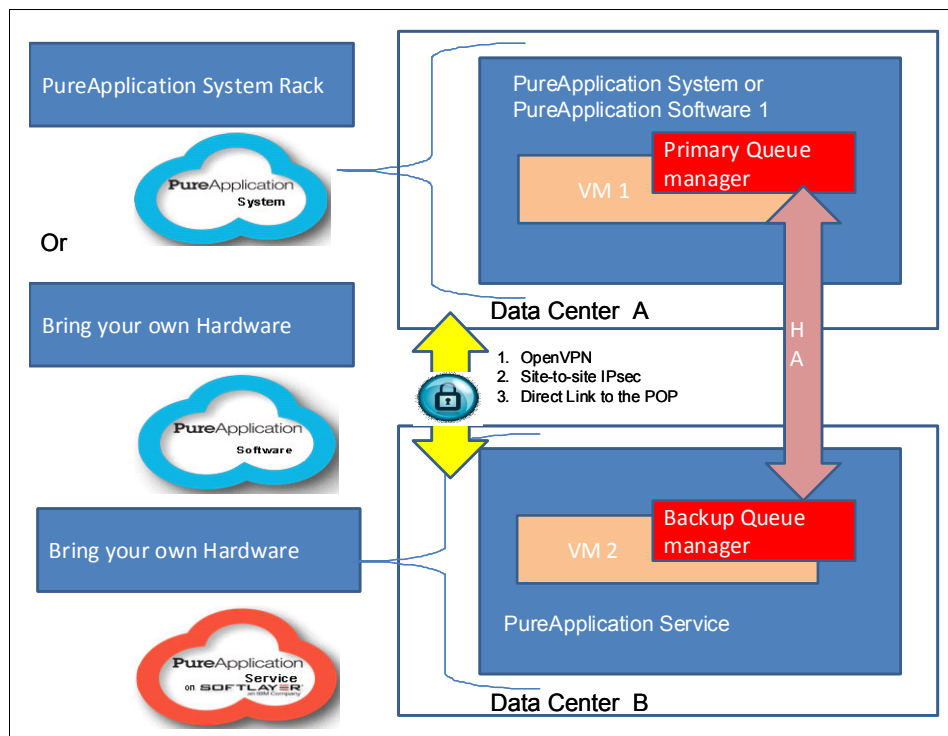


Figure 2-53 WebSphere MQ with two queue managers in different data centers

In this scenario, an active/standby mode for WebSphere MQ is across data centers. You can execute a manual configuration on WebSphere MQ patterns to synchronize the catalog between the PureApplication System rack or PureApplication Software and on the PureApplication Service instance. To be secure, the connection with two data centers can use OpenVPN, site-to-site IPsec, or Direct Link to the SoftLayer POP.

This approach can address WebSphere MQ primary Queue Manager (QManager) failures (VM or WebSphere MQ processes) and failures with WebSphere MQ standby handling the messaging. This approach can also address rack and data center failures. WebSphere MQ can be activated in the secondary data center in a short time because the queue manager data was replicated from the primary data center to the secondary data center.

In this scenario, PureApplication Service can be involved as Data Center B, for instance, and the replication can be performed on the WebSphere MQ scope.

For more information, see *Implementing High Availability and Disaster Recovery in IBM PureApplication Systems V2*, SG24-8246:

<http://www.redbooks.ibm.com/abstracts/sg248246.html>



Hybrid use cases

This chapter describes hybrid cloud use cases that support the use of a hybrid cloud that was built by using IBM PureApplication System and IBM PureApplication Service instances.

The following topics and use cases are presented:

- ▶ Overview
- ▶ Development and test environments off premises
- ▶ Components off premises and other components on premises
- ▶ Quick delivery
- ▶ SMBs that start small on a public cloud and then expand
- ▶ Start on premises and then move to the hybrid cloud
- ▶ Primary on premises and business continuity off premises
- ▶ Common technology implications of use cases

3.1 Overview

This chapter describes many use cases for the hybrid cloud, as introduced in 2.1.1, “Use cases” on page 26. This book does not include an exhaustive list of use case categories because the reasons for using the hybrid model are many and varied, as is leasing a car compared with purchasing a car, which is a common analogy. Using the public cloud is like leasing an automobile. Your reasons for leasing an automobile (or a cloud) rather than purchasing an automobile (or a cloud) depend on your situation.

By using the automobile lease-versus-purchase scenario, it is not cost-effective to purchase an automobile when you visit a distant region briefly. Renting is more cost-effective in this scenario. Alternatively, if you are a salesperson who leases an automobile, and you drive many miles or kilometers each day (more than the contract allows), you pay too much for the use of the automobile. In this scenario, purchasing the automobile is more cost-effective.

The lease-versus-purchase scenario applies to the cloud, too. Consider your usage and business requirements. Even if you can use the public cloud based on government regulations and other considerations, your usage might be lower and your business requirements might be fewer if you move to an on-premises private cloud.

To estimate the total cost of both options (public and private cloud), conduct a business value agreement (BVA) before you decide to run an on-premises or off-premises cloud, or a combination. IBM provides no-charge BVAs that use your own figures for greater accuracy or industry averages for quick estimates. For more information and assistance, contact your IBM sales representative.

Presuming that you need to customize the middleware that is used, and you prefer not to use a software as a service (SaaS) offering, perhaps the infrastructure as a service (IaaS) or platform as a service (PaaS) cloud can meet your requirements.

If you select the IaaS model, you need to install and configure all of the middleware (including application servers, databases, a Lightweight Directory Access Protocol (LDAP) server, and web servers). These tasks are required before any development or testing of new applications can begin.

Or, you can use the PaaS model to develop, test, customize, and implement nearly immediately. Although the configuration of the middleware might be necessary before an application can be implemented or deployed. The PaaS model might be more expensive initially, but it can save you the cost of installing and configuring the middleware. Compare this approach to leasing an automobile for a low price because the automobile has an engine and chassis, but no wheels or transmission. The cost of adding these requirements is more expensive than leasing a fully constructed automobile.

You can use *patterns* in IBM terminology to self-provision in a PaaS environment with only a few clicks of the mouse. See 1.3, “Hybrid cloud is critical to the success of businesses” on page 7. These patterns can be customized in one cloud (for example, an on-premises PureApplication System), exported with the click of a button, and then imported with another click of the button in another cloud (for example, an off-premises IBM PureApplication Service on SoftLayer).

Similarly, you can create an off-premises cloud first and then import it into an on-premises private cloud, as described in the first use case in this chapter (3.2, “Development and test environments off premises” on page 107). By completing a few fields (such as passwords), your cloud can be running in only 7 - 60 minutes.

This time includes middleware software, such as IBM WebSphere Application Server, an IBM Integration Bus, portal, and Business Process Manager, and even non-IBM software, depending on the size of the system that is deployed by the pattern.

A single IBM Integration Bus virtual machine (VM) and the default, customizable WebSphere Broker Manager, can be running in about 6 minutes. A large cluster for Business Process Manager with multiple application servers and HTTP servers all federated automatically into a Deployment Manager, and all started at completion, might take approximately 60 minutes, with the various portal pages all ready to be logged in to, and the administrator console ready for use.

Figure 3-1 shows sample uses cases for the hybrid cloud.

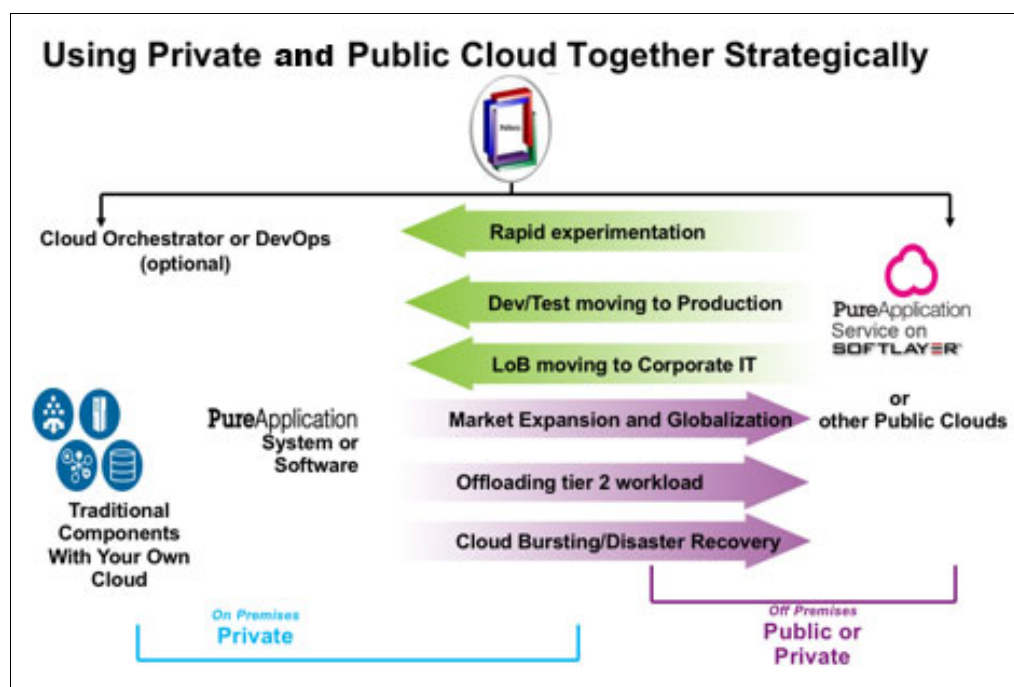


Figure 3-1 A strategy is required when you use a private and public cloud together

Next, we describe several sample use cases.

3.2 Development and test environments off premises

This scenario describes how a financial services company decides to support a line of business (LOB) project that relies on a public cloud service. The company starts with a development environment on the public cloud, and then the environment gradually evolves.

In this scenario, the company uses a hybrid cloud so that the development environment is available on a public cloud. The development environment on the public cloud needs to also be accessible by an external partner who currently performs development work. In time, the development environment can be temporarily expanded with additional resources to support

testing. This scenario leads ultimately to a pilot project, with the results shared with a group of pilot users who perform user testing on the infrastructure. The ability to temporarily add capacity, only when needed, offers significant advantages in this environment. For example, the ability to destroy and redeploy a test or development environment in identical configurations can be added.

In this use case, the financial services company LOB did not opt for the traditional method of running the project with the help of IT because of the time that is necessary for IT to create the correct environment. Typically, in a non-cloud environment, IT must perform these tasks first:

- ▶ Locate resources from existing hardware
- ▶ Acquire new hardware
- ▶ Order, install, configure, and deploy software
- ▶ Make the environment secure and available to the external partner, which involves time-consuming networking administrator services because of security considerations

The potential gains when you use the hybrid aspects of this solution stem, in large part, from the speed in setting up the necessary environment with relevant software. The development and test environments can be easily deployed, destroyed, and redeployed during software development and testing. These activities can now be performed by both internal and external parties by using test data that is provided from the company's on-premises cloud infrastructure or seeded onto an off-premises cloud.

This same scenario applies if the company did not have an on-premises cloud, although the value is more compelling if the company has an on-premises cloud. As Figure 3-2 on page 109 shows, the key architectural element in this use case is the externally accessible environment that is quickly available to the external partner for a limited period.

In the PureApplication family, these requirements match the offerings of PureApplication Service:

- ▶ Rapid provisioning of environments
- ▶ Secure interaction with client and third-party environments, servers, or workstations, through a gateway or virtual private network (VPN)
- ▶ The opportunity to start with a small cloud environment and then grow over time to the infrastructure that is needed to support development, test, acceptance, and potentially production environments

Because of the underlying technology of the PureApplication family, the pattern-based deployment technique offers quick rebuilds and the flexibility to support an agile (development) project approach.

Figure 3-2 shows the setup of development, test, acceptance, and production environments with a hybrid cloud with PureApplication products.

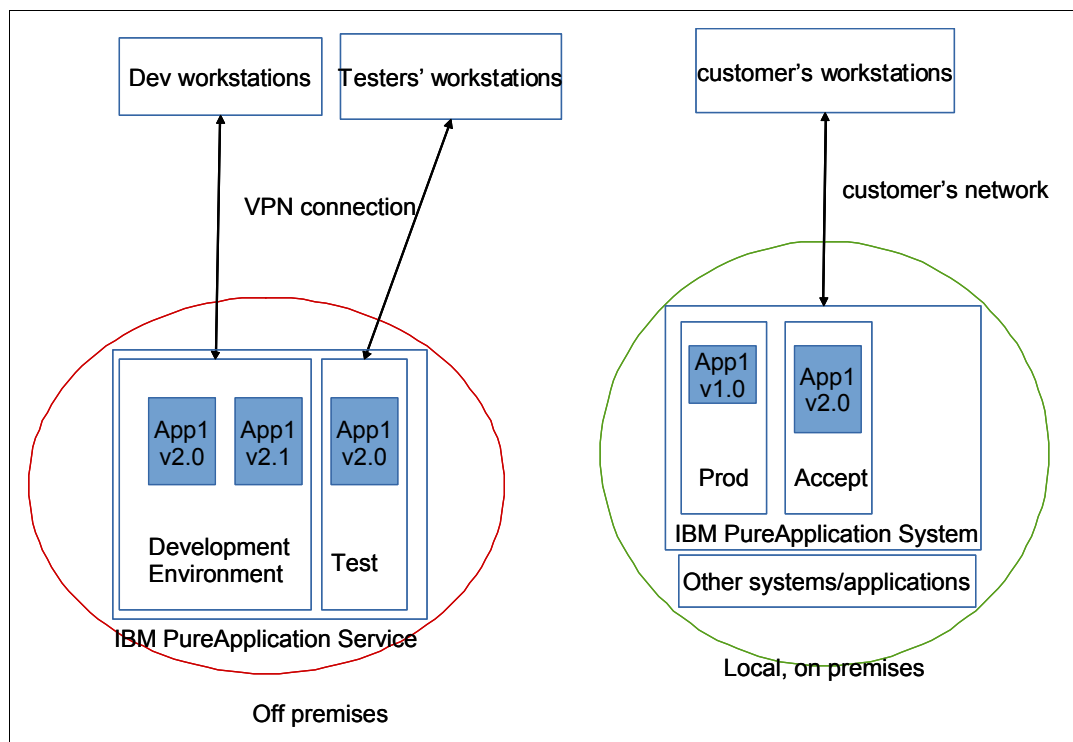


Figure 3-2 Setup of development, test, acceptance, and production environments with a hybrid cloud

For the financial services company, the greatest advantage of a hybrid cloud that is based on the PureApplication family is the ease of use in setting up the required environments. This ease of use is due to the automation that is introduced by patterns and their portability between on premises and off premises, and the ability to expose these environments to internal and external users. And, the hybrid cloud allows all of the participants to approach and deliver the project in an agile way.

Additional advantages were indirectly related to the use of a hybrid cloud. Because of the way that the third party accesses this environment, you are not exposing, or you are limiting the exposure, to your company's internal systems. In fact, the possible third party does not require access to the data centers, databases, or applications directly. In addition, you might consider applying data masking to the test data you use during development, or when you test the application on the off-premises cloud. The IBM portfolio includes products in this area. However, they are outside the scope of this publication. For more information, see IBM InfoSphere® Optim™ Data Privacy at this website:

<https://ibm.biz/BdHhiQ>

3.2.1 An extended and dynamic data center

This use case is one of the proven, easy use cases for the hybrid cloud, especially if your current cloud infrastructure is so full that you cannot add other projects without negatively affecting current resources or without requiring additional hardware, and performing network and storage configuration (assuming that you rolled out your own cloud). If you have, for example, a PureApplication System, IBM can add compute nodes and memory *hot* (while workloads continue to run) to the existing system that become available to new or existing Cloud Groups in about 20 minutes.

If an enterprise does not have an existing on-premises cloud, and even if new hardware does not need to be purchased, sometimes the internal procedures, such as internal approvals from multiple teams, to acquire, configure, and connect the necessary resources are so time-consuming that it can take 2 - 8 months before everything is in place to test a new application.

By being able to quickly rent a public cloud, pay for the required usage for a month or two, and then terminate the project, you can choose the less expensive method.

In this scenario, you install the necessary operating system (OS) and middleware on the public cloud, create, and connect to temporary databases that contain simulated data in the same form as the projected production data, and then develop the application in that environment. Your cloud provider can, like IBM, provide all of the middleware products that need to work together in an accessible manner, requiring minutes rather than days or weeks.

Typically, developers work on their own workstations or notebooks to create their piece of the solution (the application). The public cloud might be the infrastructure that functions first as the integration test, with different developers contributing to the full solution by uploading their source code (for example, as .war or .ear files).

Then, a clone of this whole system might be created and tweaked (adding, for example, CPU/vCPU, memory, or both) to simulate a more production-like setting, so that the next environment of the application lifecycle, perhaps quality assurance (QA) or performance testing, can proceed, still in the public cloud. Again, if you are using an IaaS cloud, you must perform more manual work to create these environments.

With a PaaS cloud, the cloning and tweaking of the required middleware is typically trivial, especially if *patterns* or their equivalents are used. A clone of a complex pattern (with multiple application servers, HTTP servers, and so on) in the PureApplication family can be created with a single click of the mouse and with a unique name for the clone.

When the application is finished and evaluated (typically at the different stages) and deemed useful and ready, it can be brought in-house for the final steps. If new compute resources were needed internally, you can acquire and configure those resources in parallel with the development work, speeding up the process to go live. In any event, depending on whether you have a private cloud, different degrees of manual work must be performed to prepare the internal infrastructure.

If patterns can be created and customized to include even the application (to make a quasi-software-as-a-service offering) in one environment (in this case, the public cloud) and exported and then imported to the other environment (in this case, the on-premises cloud), with a few clicks and a few changes for network configuration and connecting to the different, on-premises databases, you can further reduce the time, cost, and labor to start the running application in a staging environment on premises.

3.3 Components off premises and other components on premises

Developing a business application where several components are off premises and other components are on premises clearly embodies the essence and capabilities of the hybrid cloud:

- ▶ The opportunity to place your workload on the best platforms, choosing between on premises and off premises.
- ▶ Not all enterprises can move completely to the public cloud because the enterprises might need to host part of the data and application on premises for security reasons.
- ▶ Many startups begin on the public cloud, and as they grow, they move to a hybrid cloud to balance costs, security, and performance.

For this use case, we use the analogy of building of a bridge between two countries. Building a bridge requires you to address several considerations:

- ▶ How long will the bridge be?
- ▶ How many vehicles can be on the bridge at one time?
- ▶ How much traffic is anticipated?
- ▶ How will traffic be monitored?
- ▶ What can be transported over the bridge and what cannot be transported?
- ▶ Which security services are needed at the beginning and end of the bridge?
- ▶ What language is spoken at the beginning of the bridge and at the end of the bridge?

Similarly, implementing this use case requires the following considerations:

- ▶ What is the distance between the on-premises data center and the data center of the cloud service provider?
- ▶ How much network bandwidth is required between the front end and the back end?
- ▶ Which types of information are exchanged between the front end and the back end?
- ▶ What security is required by the cloud service provider and the on-premises data center?
- ▶ What communication protocols are used between off-premises and on-premises components?

Figure 3-3 illustrates a generalized architecture of this hybrid use case, where you choose to have front-end components off premises (PureApplication Service) and back-end services on premises (PureApplication System).

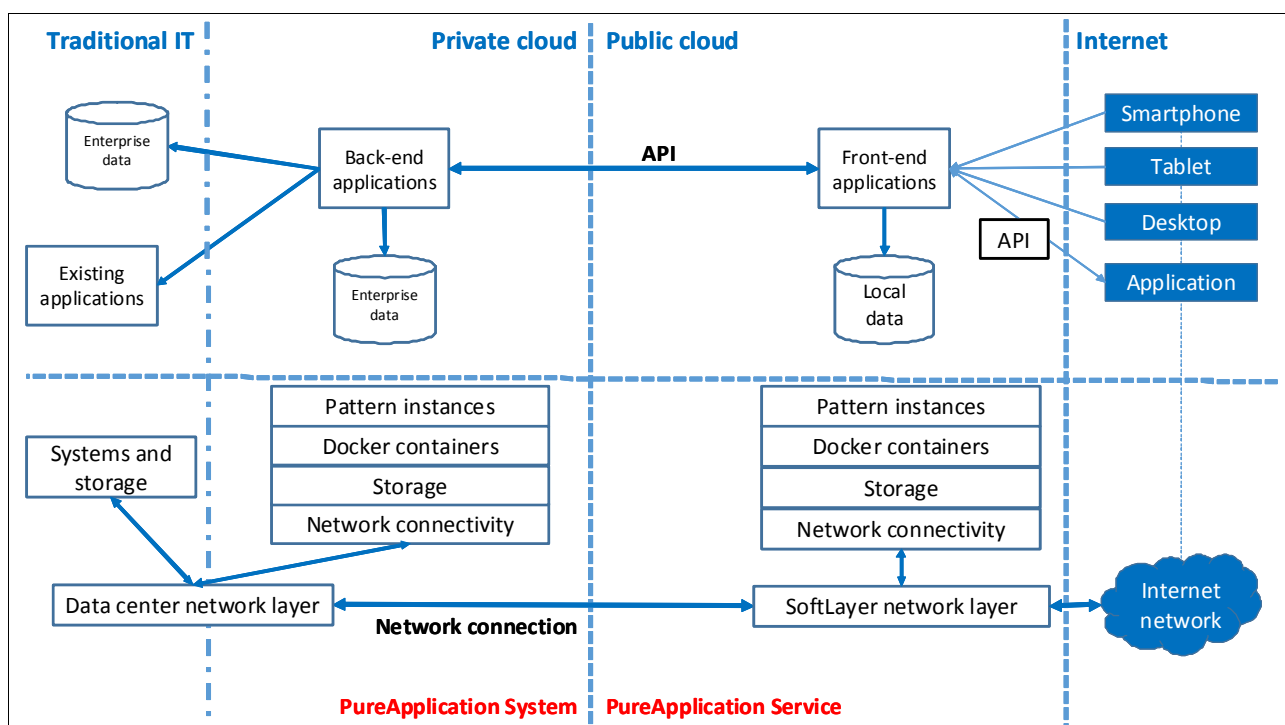


Figure 3-3 A generalized architecture of this use case

Figure 3-3 shows the logical and physical resources on premises and off premises:

- ▶ On premises: Existing applications and enterprise data are placed on top of various possible systems, which are based on different architectures (including IBM Power, IBM System z, and Intel).
- ▶ On premises: Back-end applications and (optionally) enterprise data are placed in a private cloud that is based on PureApplication Systems.
- ▶ Off premises: Front-end applications and data are placed in a public cloud that is based on PureApplication Service.
- ▶ On-premises networking: Connects PureApplication Systems to the traditional IT systems.
- ▶ A secure network layer: Connects the on-premises data center to the off-premises data center.

Consider the following questions:

- ▶ How can I protect the data and applications on the public side?
- ▶ How can I handle application security on the public cloud side?
- ▶ Are the front-end applications scalable, and can I quickly add more resources to my platform on the public cloud?
- ▶ How can I back up my off-premises resources?
- ▶ Can I have a monitoring view of the entire hybrid cloud?

- If I want to move my back-end services to a public cloud, can I?
- Can I have a business continuity strategy to protect my services in a disaster that occurs at the cloud provider data center?

Addressing these considerations depends on the nature of the specific business application (such as business critical issues, frequency of usage, and user types) and its technology context (such as adopted technologies, type of users, and non-functional requirements).

Next, we describe how mobile applications are suitable to be placed on the public side of this use case.

3.3.1 The convergence between cloud and mobile

Cloud computing changed IT. It continues to change IT in the same way that mobile applications are changing IT. People are using smartphones more than traditional personal computers, continuously installing and deleting mobile applications. Petabytes of data are generated every day by mobile users around the world. Enterprises often have to focus on mobile applications. While they define an adoption strategy, it is important to consider that mobile applications differ from traditional business applications. Mobile applications have specific characteristics:

- They are more strategic because users can easily use them almost all of the time, even when not the user is not tethered to a desk.
- They are context-aware because a mobile application must adapt itself to the current context of the user (for example, time zone, language, and location).
- They are always on: 24x7 and 365 days a year.
- They run on unreliable networks. Certain smartphones might provide offline capabilities.
- They must support various devices with a smaller screen, various operating systems, and different hardware (responsive design).
- Smartphones contain sensitive data, and the phones can be lost or stolen.
- Mobile applications require agile development to support the fastest development cycles and continuous optimization that is based on feedback from global users.

One key factor to the successful adoption of mobile is the technology. Cloud computing is probably the best place to handle the massive data and computational resources that support mobile applications.

Based on the convergence of cloud computing and mobile, the next use case describes a more specific use case where the front-end side is based on IBM MobileFirst™ Platform Foundation. This use case was derived from a real project that was performed by a large retail corporation.

3.3.2 IBM MobileFirst Platform on IBM PureApplication Service

The IBM MobileFirst Platform is a suite of products that enables an enterprise to adopt a mobile strategy. You can use the platform to build and deliver mobile applications with its three parts:

- ▶ IBM MobileFirst Platform Foundation: Formerly known as IBM Worklight® Foundation, this product provides the platform and tools to build, administer, and monitor mobile applications.
- ▶ IBM MobileFirst Platform Application Scanning: Provides tools to analyze code (JavaScript, HTML, and Java) to identify security vulnerabilities.
- ▶ IBM MobileFirst Quality Assurance: Contains tools and features to provide quality assurance to mobile applications.

The IBM MobileFirst Platform Foundation is available on PureApplication System and PureApplication Service on SoftLayer. For more information about the Foundation, see *Deploying MobileFirst Server on IBM PureApplication System* at this website:

<https://ibm.biz/BdHAiD>

Figure 3-4 and Figure 3-5 on page 115 describe the hybrid cloud scenario in which the front-end applications are based on IBM MobileFirst Platform.

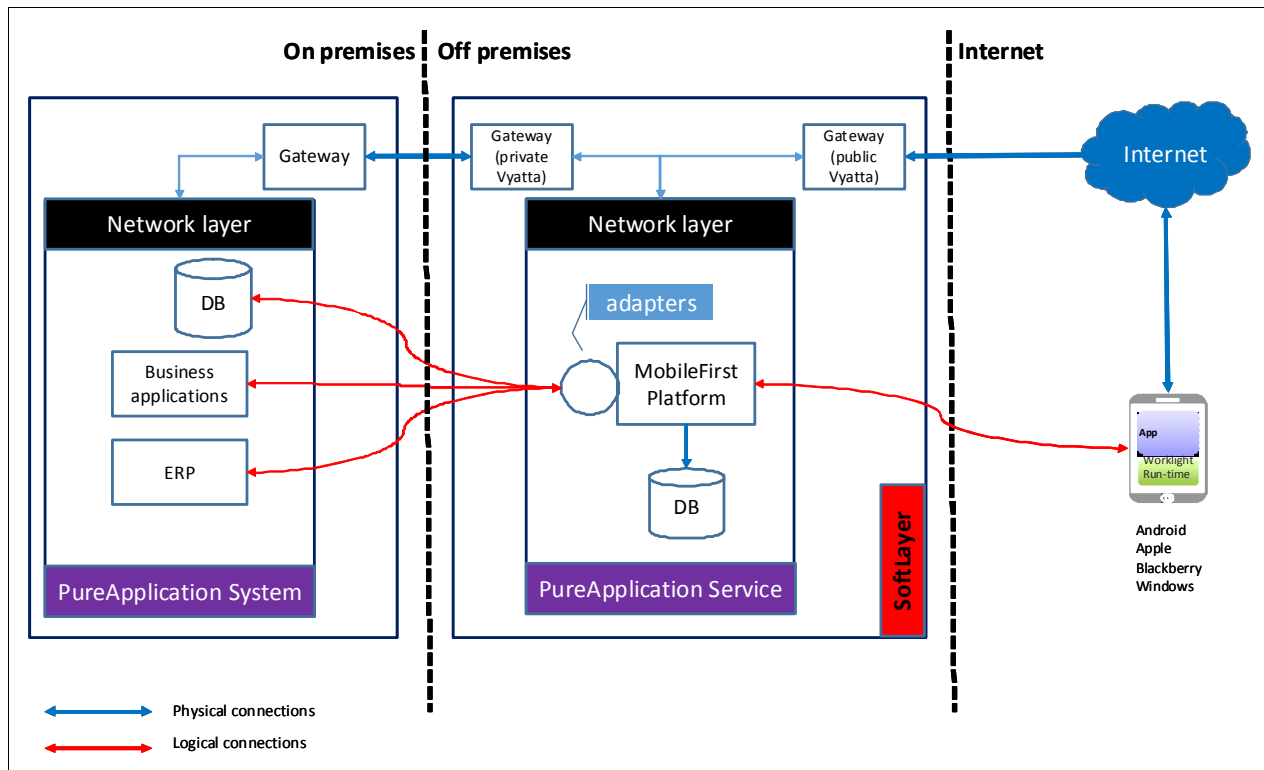


Figure 3-4 IBM MobileFirst Platform on PureApplication Service

Figure 3-4 shows a single component to identify the entire MobileFirst Platform, but the platform includes several components:

- ▶ MobileFirst Server: Represents the middleware tier that acts as a gateway between back-end systems, services, and the mobile client applications
- ▶ MobileFirst API: Related to server-side and client-side application programming interfaces (APIs)
- ▶ MobileFirst Studio: An Eclipse-based tool for developing applications on MobileFirst Platform
- ▶ MobileFirst Console: Provides a management portal and a dashboard that offer you complete control over the MobileFirst Platform
- ▶ MobileFirst Application Center: Acts as an application store within the enterprise

For more information, see these resources:

- ▶ *IBM MobileFirst Strategy Software Approach*, SG24-8191:
<http://www.redbooks.ibm.com/abstracts/sg248191.html?Open>
- ▶ *IBM MobileFirst Platform {dev}* (for information about the integration capabilities of the MobileFirst Platform):
<https://developer.ibm.com/mobilefirstplatform/documentation/integration-7-1/>

An enterprise mobile application requires integration with the back-end services and data. The MobileFirst Platform provides the adapter framework to develop the server-side code for mobile applications. The adapters run on the MobileFirst Server, which makes it possible to establish bidirectional communication to the back-end services and data, and to perform server-side logic.

Figure 3-5 describes the role of the adapters.

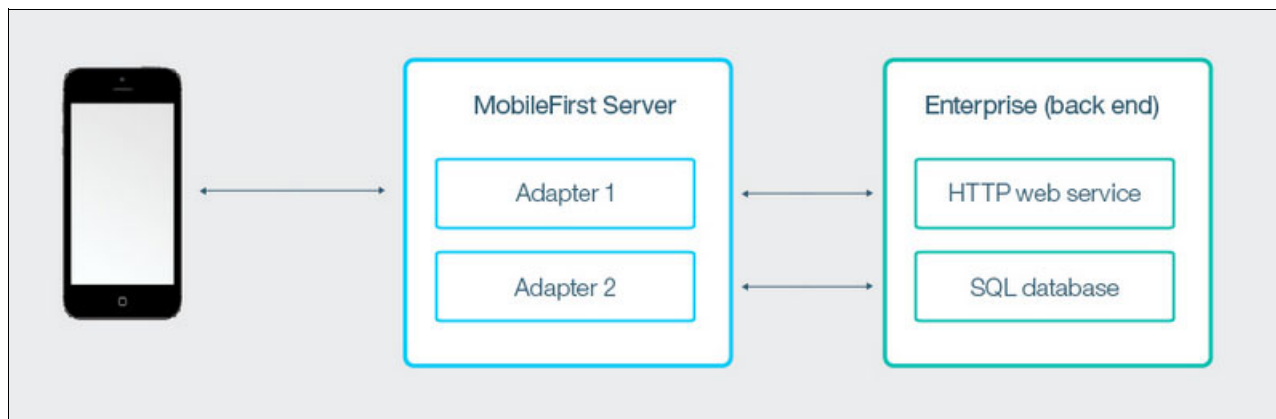


Figure 3-5 The role of the adapters in MobileFirst Platform

For more information, see *Overview of MobileFirst adapters*:

<https://ibm.biz/BdHAis>

MobileFirst Platform provides Java and JavaScript adapters to support the following capabilities:

- ▶ SQL integration to directly interact with enterprise data
- ▶ The establishment of HTTP communication as an instance to provide Representational State Transfer (REST) architectural constraints (RESTful API) and interactions
- ▶ Java Message Service (JMS) to establish an asynchronous connection to the back-end services

Beyond these integration capabilities, MobileFirst Platform also provides an IBM Cast Iron® adapter. The IBM WebSphere Cast Iron Cloud is a virtual or physical appliance that enables integration between applications, wherever applications are (on premises and off premises), so this appliance plays a crucial role in a hybrid cloud scenario.

Figure 3-6 shows the overall architecture of this use case, including the WebSphere Cast Iron Cloud.

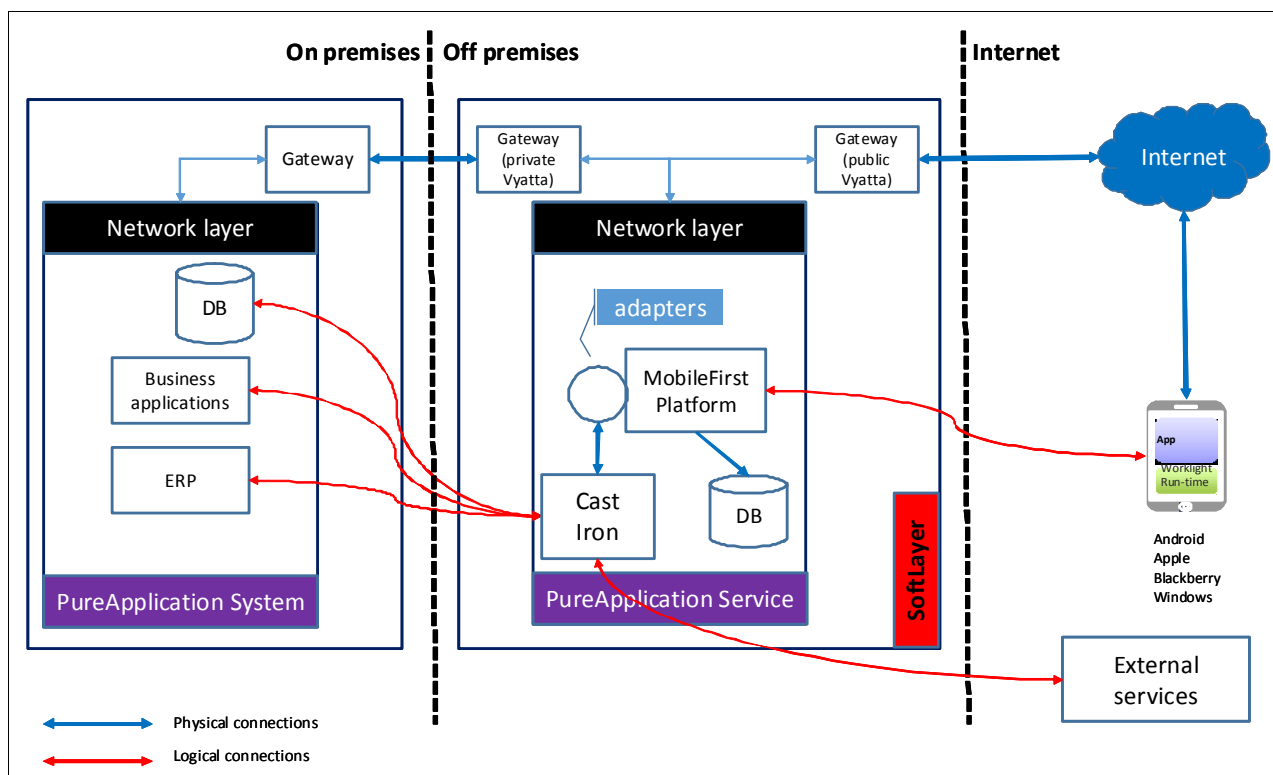


Figure 3-6 Supporting integration between on premises and off premises with WebSphere Cast Iron Cloud

You can use WebSphere Cast Iron Cloud to design the integration capabilities that you need to integrate applications by providing a specific development tool: WebSphere Cast Iron Studio. The approach to the design does not require programming knowledge. It uses a graphical user interface (GUI). Moreover, the design of the integration capabilities uses many pre-built connectors to multiple applications. If connectors are not available for your scenario, it is possible to build custom connectors by using Connector Development Kit.

For more information, see the *IBM Cast Iron Connector Development Kit (CDK) Developer's Guide*:

<https://ibm.biz/BdHAiz>

From a security perspective, WebSphere Cast Iron Cloud connects to the endpoints (for example, applications, databases, and flat files) by using several secure communication protocols:

- ▶ HTTPS (HTTP over Secure Sockets Layer (SSL)) by supporting bilateral authentication, privacy, and integrity
- ▶ Secure web services that are based on SOAP/HTTP over SSL by supporting bilateral authentication, privacy, and integrity
- ▶ File Transfer Protocol (FTP) over Secure Shell (SSH) and FTP over SSL by supporting secure mechanisms for FTP server authentication, privacy, and integrity
- ▶ Databases that rely on SSL for security by supporting a secure mechanism for database access

For more information about WebSphere Cast Iron Cloud, see the following websites:

- ▶ *WebSphere Cast Iron Cloud integration, Version: V7.0:*
<https://ibm.biz/BdHhub>
- ▶ *IBM WebSphere Cast Iron Introduction and Technical Overview, REDP-4840:*
<http://www.redbooks.ibm.com/abstracts/redp4840.html?Openf>
- ▶ *IBM WebSphere Cast Iron Version 7.0 documentation:*
<https://ibm.biz/BdHhup>
- ▶ *Deploying a WebSphere Cast Iron Integration Appliance pattern on IBM PureApplication System:*
http://www.ibm.com/developerworks/websphere/library/techarticles/1501_iyengar/1501_iyengar.html

The next section describes the implications of this use case.

3.3.3 Implications of this use case

Security and networking are key aspects of this use case. Networking embodies the fundamental bridge between front-end applications and back-end services, and the data is on both sides of the bridge. Also, data is in transit on the bridge, and the data probably needs to be encrypted. However, the underlying architecture of this use case is not new. Several traditional scenarios exist in which you need a special focus on networking and security:

- ▶ An active-active, high availability (HA) scenario on geographically distributed data centers
- ▶ Front-end applications in one data center, and back-end services in another data center
- ▶ Multiple tier architectures within one data center, in which the front-end components are physically separated from the back-end components

In the types of use cases that are listed, you can count on high-speed network connections and complete control over the security strategy to adopt. In a hybrid cloud, the situation is different. Networking and security considerations are common to all use cases, and 3.8, “Common technology implications of use cases” on page 131 describes these implications.

This use case has other implications. When you consider application development, where do you need to place development, test, QA, and acceptance environments for mobile application development?

Table 3-1 compares two possible alternatives to this question.

Table 3-1 *Placing test environments for mobile applications*

	PureApplication System on premises	PureApplication Service off premises
MobileFirst Platform support	PureApplication System and PureApplication Service support the IBM MobileFirst Platform, providing predefined patterns.	
Flexibility to obtain more resources	The ability to obtain more local resources depends on the usage of the PureApplication System. If the PureApplication System is not filled with compute nodes, IBM can provide additional CPU and memory while the PureApplication System is hot by simply installing more compute nodes. If all bays are filled, it might be easier to add more resources to the cloud application platform, which is based on PureApplication Software with unused hardware, rather than obtaining an additional PureApplication System. It depends on your local procurement processes, and planning is crucial. With PureApplication Software, you must build all of the additional components (network, storage, hypervisor, and OS). With PureApplication System, it is time to order the needed hardware.	PureApplication Service offers the capability of changing existing instances, and the possibility of obtaining a new instance.
	A pattern-based approach simplifies the deployment and de-provisioning of test environments on the cloud.	
Obtaining production-like environments during the application development and test phase	PureApplication System offers the same capabilities of PureApplication Service for building a MobileFirst Platform. However, it is necessary to locally simulate the off-premises scenario from the network perspective.	PureApplication Service can provide production-like environments.

Another crucial topic for this use case is business continuity.

Mobile applications can be mission-critical, and they might require business continuity. PureApplication Service relies on the availability of PureApplication Service on SoftLayer data centers around the world.

Figure 3-7 shows how you can consider an active-active scenario where two PureApplication Service instances are placed in different PureApplication Service on SoftLayer data centers.

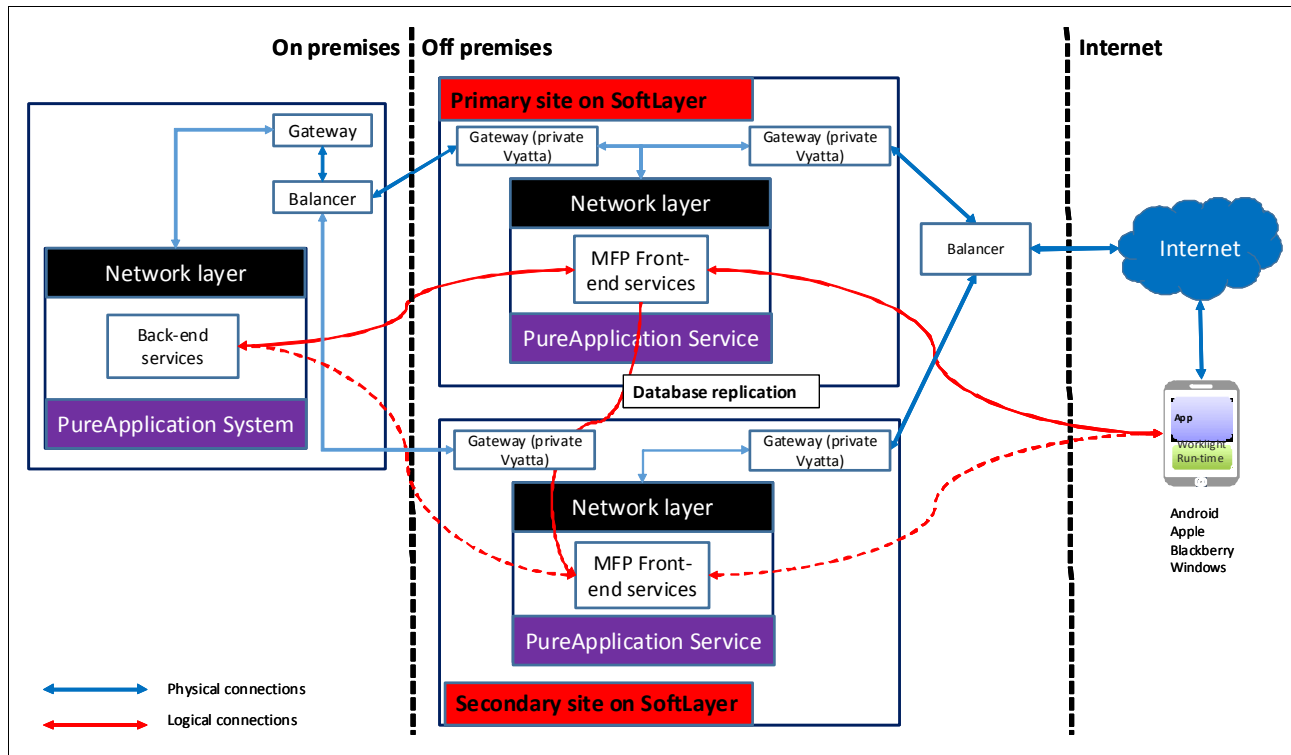


Figure 3-7 Business continuity scenario that is based on PureApplication Service on SoftLayer data centers

You need the following capabilities for the scenario in Figure 3-7:

- ▶ You need global load balancing capabilities. PureApplication Service on SoftLayer provides the necessary services to address this requirement.
- ▶ Communication between off-premises data centers can be a virtual private network (VPN) link over a PureApplication Service on SoftLayer private network.
- ▶ It is necessary to implement an active-active topology with MobileFirst Platform.

Finally, in this context, you can take advantage of the geographically available PureApplication Service on SoftLayer data centers. Consider the following case: Starting from the scenario that is described in Figure 3-4 on page 114, an enterprise might provide a mobile application for online shopping to its customers that are concentrated on their own continent, for example, North America.

By trying to expand their market, the enterprise can extend their mobile online shopping application to another continent, for instance, Europe. If the number of customers in Europe increases dramatically (for example, because of scalability, performance, proximity, and security), the enterprise can quickly shift to an active-active scenario, such as the blueprint that is described in Figure 3-7. The first site is on the PureApplication Service on SoftLayer data center in Dallas, Texas (US). The second site is on the PureApplication Service on SoftLayer data center in London, England (Europe).

Table 3-4 on page 132 describes additional common implications of all of the use cases in this chapter:

- ▶ Security
- ▶ Networking
- ▶ System management
- ▶ Business continuity

For more information, see the following websites:

- ▶ *Load Balancing: Let your servers share the load* about the global load balancing services that are offered by PureApplication Service on SoftLayer:

<http://www.softlayer.com/load-balancing>

- ▶ *How to coordinate and execute an active-active topology with IBM MobileFirst Platform Foundation:*

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

3.4 Quick delivery

Imagine a situation where your company is required to quickly share information for a limited period, and the start and end dates are known in advance. For example, these dates are fixed, such as during a holiday season or for an annual auction.

Not needing to provision an environment on premises reduces time, money, and resources, such as operational expenditures (OPEX) and capital expenditures (CAPEX). You want to benefit from having monitoring and backup and restore capabilities in place. The capability to quickly rebuild previously developed environments is an advantage in this scenario. In commercial concepts, this scenario is comparable to the pop-up retail concept, which is also known as a *pop-up store* or a *pop-up shop*.

From an architecture perspective, because your pop-up store or informative website is an entity on its own, you begin the development and testing on premises, including all of the required components in a pattern. At the correct time, bring the application (web server, database, application, and so on) as one entity to the remote, off-premises platform. When the system is no longer needed, the pattern instances are stopped and deleted. At the end of this period, the off-premises compute environment is stopped and returned to IBM or reused for other purposes. Likewise, temporarily ordering increased capacity for storage or compute power is applicable to this use case.

Benefits of the quick delivery use case are shown:

- ▶ **Time-to-value:** Beneficial because you need to quickly deliver strategic applications for a short period. Time and global availability are two key elements. An instant marketing campaign is an example of this scenario.
- ▶ **Cost optimization and pay-per-use:** You want to allocate the infrastructure resources only when you need the application. When the application is no longer necessary, its resources can be deallocated or used for another purpose.

Applications that are suitable for quick delivery have a specific lifecycle, as Figure 3-8 shows.

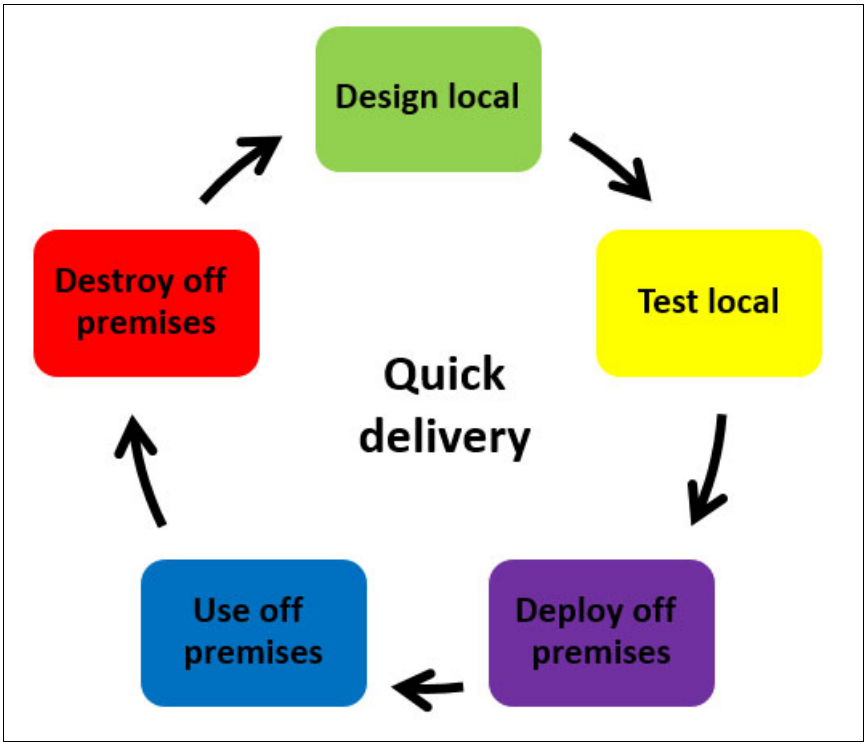


Figure 3-8 Lifecycle of a quick delivery use case

Every step of the process implies considerations, and the time between steps depends on many factors. In one case, you anticipate the design step. The deployment of the application can be significantly shortened or decreased later. In another case, the whole process can be quick. The process is iterative because you might need the application many times (for example, every year), applying changes over time.

Table 3-2 describes implications of every step of the quick delivery lifecycle and how all of the steps can be accommodated by using IBM PureApplication products, patterns for quick deployment, and the IBM PureApplication Service self-service portal.

Table 3-2 Five-step process for approaching the lifecycle of quick delivery

Step	Considerations
Design local	<p>It is possible to design your application off premises. Probably, the design, development, and test are performed locally.</p> <p>However, wherever you decide to develop the quick delivery application, the portability of patterns between PureApplication System and PureApplication Service makes this choice simple.</p>
Test local	<p>Similarly, testing can be conducted on premises or off premises to test the networking capabilities of the public cloud side.</p> <p>The portability of patterns between PureApplication System and PureApplication Service makes it simple to move the quick delivery application between on premises and off premises.</p>

Step	Considerations
Deploy off premises	<p>If you designed and tested your application locally, during the deployment step, you can export the virtual patterns from PureApplication System and import them into the PureApplication Service. Briefly, the deployment step implies the following activities:</p> <ul style="list-style-type: none"> ▶ Allocate the necessary resources on PureApplication Service ▶ Configure network connectivity between PureApplication Service and the on-premises data center ▶ Configure network connectivity between PureApplication Service and the internet ▶ Export the virtual patterns from the PureApplication System ▶ Import the virtual patterns into the PureApplication Service instance ▶ Deploy the virtual patterns
Use off premises	<p>The quick delivery application can create, read, update, and delete data, and data can be on premises and off premises.</p> <p>The quick delivery application has a short life. When you do not need it anymore, you might (optionally) need to move the data that was produced off premises to on premises.</p>
Destroy off premises	Finally, the last step is the deprovisioning of the application.

An example of a quick delivery use case is the *click day* application. The Italian government uses this type of application when many job openings exist of a specific type, for which applications are accepted for one day only through an internet portal. The Italian government organizes a click day for similar recurring events, too.

These quick delivery applications have many concurrent users that are concentrated in a short period. The volume is so high and concentrated that it might look like a denial-of-service (DoS) attack. For this reason, quick delivery applications require a scalable infrastructure and a high-speed connection to sustain the load that is generated by users.

From the perspective of the quick delivery use case, it is possible to locally design, develop, and test the application, where you do not need many resources to implement the test and QA environments. Instead, the public cloud easily offers the resources (compute nodes, storage, and network) to implement the production environment on a pay-per-use basis.

3.5 SMBs that start small on a public cloud and then expand

This client decides to jump-start their business in the public cloud, move to a private cloud with IBM PureApplication System for more stability when the business expands, and eventually bring the entire environment back on premises. In this way, the client can rely on the advantages of portability when they use technology that is based on the common IBM PureApplication technology to build their off-premises and on-premises clouds.

This use case correlates to the phenomenon that is experienced by startups. For a startup, it is easier to initially build their IT on a public cloud because a virtual data center is cheaper than a physical one. However, as the client grows and needs more systems and IT capabilities, the client remains on premises. A hybrid cloud scenario allows them to balance the cost, IT optimization, and security.

Important aspects in this use case are system management capabilities. How will you perform OS-level maintenance and fix middleware? How will you monitor the environment?

When you choose these capabilities, focus on building a product that delivers what is required from day one, and offering enough flexibility to grow and expand with your growing cloud environment. A form of orchestration is required.

Adopting an approach while you keep in mind that a future evolution will introduce significant change requires good discipline. The design and implementation of the initial scenario must constantly consider the needs of the future scenario. If your data center starts on a public cloud, over time you must address many aspects, especially if you are adopting an IaaS model. In an IaaS model, considerations cover the design of the infrastructure to application development and system management.

All considerations must include a future shift to the hybrid cloud. For instance, if you are a startup, you begin with a small data center on a public cloud, and you accept the minimum set of system management capabilities, yet you think about a future on-premises data center. If you are planning to adopt a PaaS model, you depend on the system management capabilities of the cloud service provider. Whatever your initial service model is for the public cloud side, your application development approach must always consider the future hybrid scenario, and cloud application portability is a crucial requirement.

Building the off-premises data center on PureApplication Service offers important benefits:

- ▶ The environment natively provides the system management capabilities that you need.
- ▶ Patterns are available to build an application management platform that can be based on the following products:
 - IBM Rational Collaborative Lifecycle Management
 - IBM Rational Application Developer
 - IBM Rational License Key Server
 - IBM Rational Test Virtualization Server
 - IBM Test Workbench
- ▶ The native pattern engine provides a way to provision and de-provision your workloads.
- ▶ It is possible to consider a business continuity active-active topology by using the global availability of PureApplication Service on SoftLayer data centers.

Note: For more information, see *IBM Software Delivery and Lifecycle Patterns V1.0.1* documentation:

<https://ibm.biz/BdHAjp>

Beginning with your IT running on PureApplication Service, what happens when you need to expand on premises? You need to build a private cloud.

Table 3-3 describes the benefits of basing the private cloud on PureApplication System.

Table 3-3 *Shifting to a hybrid cloud*

Topic	Benefits
Cloud platform	PureApplication Service and PureApplication System share the pattern engine. The portability of patterns makes it possible to move easily from off premises to on premises.
Application lifecycle management (ALM)	<p>You might think of moving ALM on premises or using another ALM platform on premises.</p> <p>ALM is a pattern-oriented implementation on PureApplication Service. Patterns are available on PureApplication System, too, to easily create another ALM platform.</p>

Topic	Benefits
Application migration and portability	<p>In the new hybrid scenario, you might start to develop new applications on premises, and you might migrate several applications from off premises to on premises. Also, your new scenario might need business-to-business integration between off-premises and on-premises components.</p> <p>The adoption of the container virtualization mechanism (Docker) and the microservices architectural style can simplify or mitigate the risks of your changes at the application level.</p> <p>Both PureApplication Service and PureApplication System support Docker containers.</p>

Figure 3-9 includes all of these aspects in a single view.

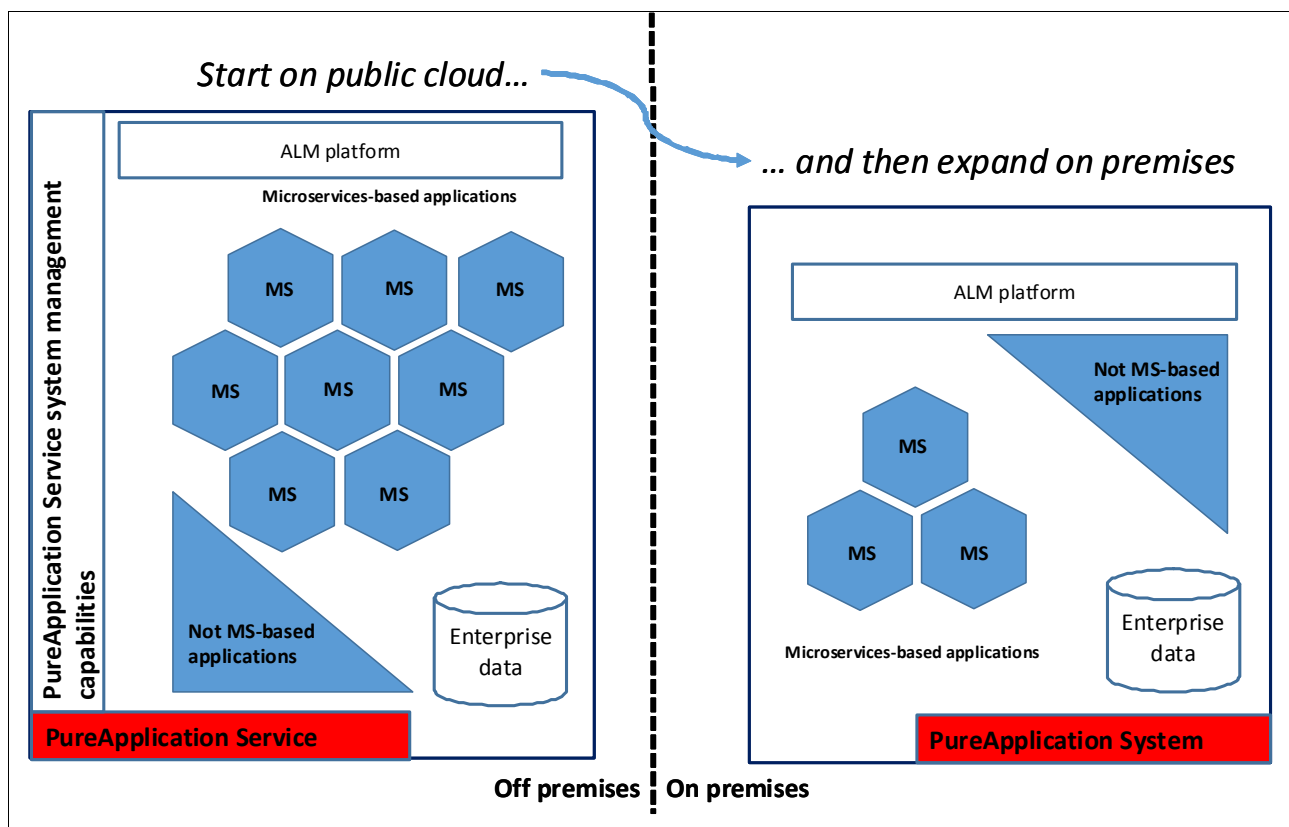


Figure 3-9 Starting on a public cloud and then expanding on premises

The approach to application development is crucial for this use case, and the adoption of microservices and Docker can provide significant benefits. The microservices architectural style prescribes that you build a single software application as a composition of many small microservices, rather than one monolithic application. The microservices style offers the following benefits:

- ▶ Every microservice provides a simple capability. It is auto-consistent. It can be deployed independently of other microservices that belong to the same business application.
- ▶ Microservices are loosely coupled among each other, and they communicate with each other by using language-independent APIs, such as REST.
- ▶ Every microservice is relatively small. It is usually owned by a small team.

As an analogy, the microservices architectural style is similar to applying the divide and conquer concept to the development of a complex software application. The development of the entire application is subdivided into the development of small auto-consistent modules (*microservices*). The final result is a composition of interconnected microservices. However, the adoption of this approach requires certain considerations:

- ▶ If you are starting to develop a new application, it is not always necessary to start with a microservices approach.
- ▶ It is important not to subdivide the software application into too many small microservices because too many microservices increase the complexity of the integration between microservices. Too many microservices can introduce possible latency, making the entire application unstable.
- ▶ Usually, the development of microservices involves many teams. Therefore, governance and coordination are vital to ensure the success of the entire application.
- ▶ The number of resources (for example, database and application servers) that are introduced by microservices increases the importance of an effective software delivery pipeline, including deployment and monitoring automation.

For more information, see *Microservices from Theory to Practice: Creating Applications in IBM Bluemix Using the Microservices Approach*, SG24-8275:

<http://www.redbooks.ibm.com/abstracts/sg248275.html?Open>

Microservices work well on cloud platforms. Microservices take advantage of the flexibility of the cloud in terms of resource availability, improving the scalability of the applications. Combining microservices and Docker can be the best choice to achieve portability of microservices across a hybrid cloud scenario.

For more information about Docker and portability, see 4.3.2, “Docker” on page 172.

3.6 Start on premises and then move to the hybrid cloud

This use case explains the adoption of the cloud by an enterprise on an on-premises basis, starting with a few applications, to an off-premises environment, then moving all applications to the cloud. The business value for this solution, together with the capabilities of the PureApplication family, is explained in this section.

3.6.1 Business value

Imagine the scenario in which an enterprise has a system off premises and concerns about the cost of maintaining the internal services for machines, infrastructure, and technical trainers on the site to the support IT infrastructure. Also, with continuous technology improvements that need to be made to on-premises systems for software licenses and hardware, it is difficult to keep up-to-date. The use of the most promising technologies requires considerable financial investment.

Security concerns are one reason to not move all workloads to an off-premises cloud environment at one time. A reason to move workloads off premises is that off premises is a better environment for interacting with system of engagements applications. The possibility of business opportunities is attractive, and it is motivating enterprises to move to the cloud environment.

Another situation is that of a new enterprise that wants to develop solutions for customers. For example, an enterprise needs a new business analytics application or a business process management (BPM) application, with faster delivery to market. The enterprise does not have time to obtain enough technical skill to build the solution on premises. The operational expense to maintain a solution on premises is another key factor that enterprises want to avoid. It is complex to build and maintain an on-premises security infrastructure.

For specific outsourcing projects, enterprises want an entire packaged solution, without worrying about building and maintaining. These factors lead enterprises to build and use cloud-centric applications and deploy directly to the cloud.

The cloud journey needs to be started with less critical applications, or possibly by moving seasoned business applications to the cloud until all applications are moved. The business value is high because it can reduce IT operational costs without decreasing clients' security.

3.6.2 Architecture

First, follow the cloud architecture blueprints of cloud adoption as described in 2.1, “A five-step roadmap for establishing a hybrid cloud” on page 26. One of the most important points is to start the cloud journey with a map of applications from a business criticality standpoint. Then, check the applications that are cloud-centric or cloud-enabled and those applications that do not have cloud adherence. The applications that are not ready to go to the cloud can be enabled for a cloud adoption program. Figure 3-10 explains the general idea of this approach. The application is cloud-centric.

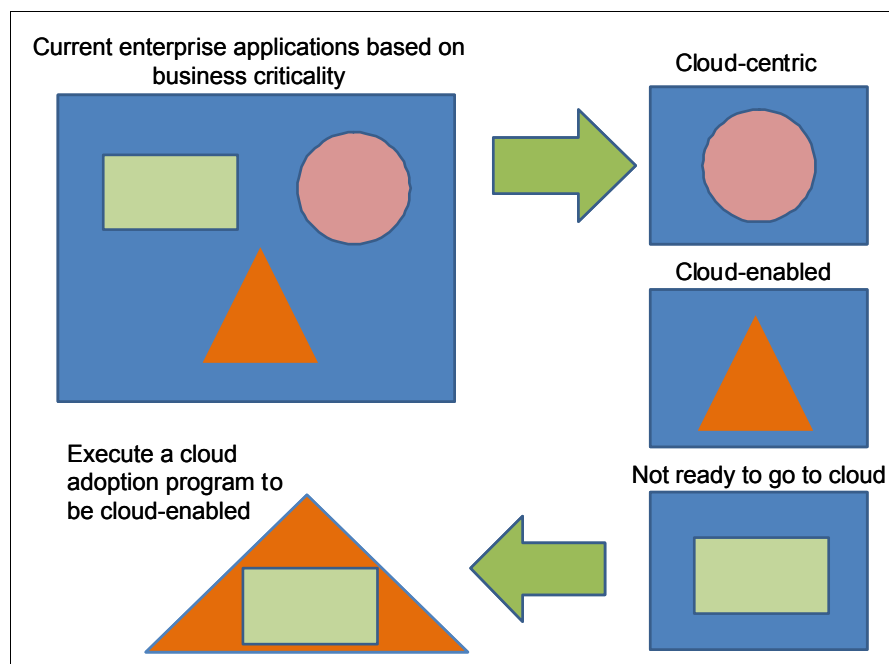


Figure 3-10 Architecture and design approach to moving applications to the cloud

The cloud adoption program maps the application with no cloud adherence and applies the cloud adoption program to use, for example, a web services technology, such as SOAP or RESTful web services. The application components communicate by using JavaScript Object Notation (JSON) or XML, and the communication can be integrated with different programming languages. For IaaS and PaaS, for instance, select the best strategy for choosing a cloud provider or building a cloud structure in the client's data center to help this transformation.

Choose the correct deployment model

Based on the enterprise requirements, an architect needs to choose the correct deployment model, considering all cloud layers, such as IaaS, PaaS, and SaaS.

Figure 3-11 provides an overview of these service models and deployment models.

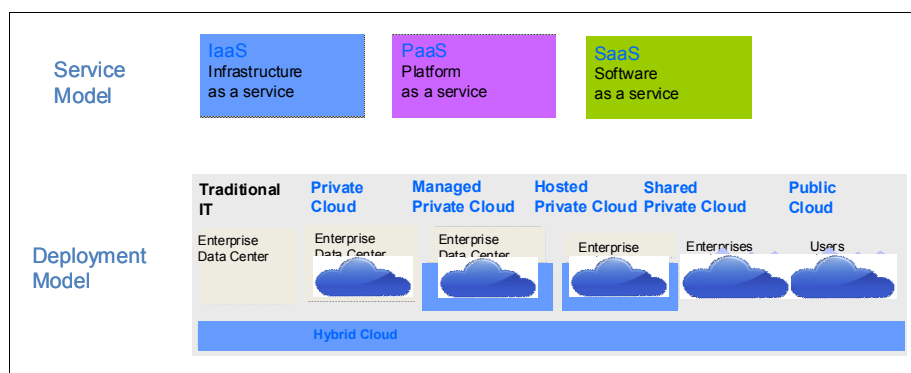


Figure 3-11 Deployment models and service models

In most cases, an enterprise uses either a traditional IT infrastructure in their data center or a private cloud (when cloud adoption is mature), or a managed or hosted private cloud that uses products, such as PureApplication System and PureApplication Software.

The most suitable options for moving business applications to the cloud are listed:

- ▶ A hosted private cloud: This cloud offers a complete data center that is off premises and dedicated to an enterprise. This option is important when an application has high business criticality.
- ▶ Shared private cloud: This cloud offering is suited for applications with less business criticality because this cloud can share an environment with other business applications from other enterprises and with a level of security isolation.

PureApplication family offerings for this use case

The PureApplication family can support this use case. The enterprise starts on premises. The preferred practice is to start the cloud adoption on premises, moving all workloads to the cloud by using PureApplication System or PureApplication Software. By using patterns (2.3, “Patterns and the PureApplication family” on page 50), it becomes easier to move from on premises to an off-premises workload.

Figure 3-12 describes the solution at a high level without including security and networking.

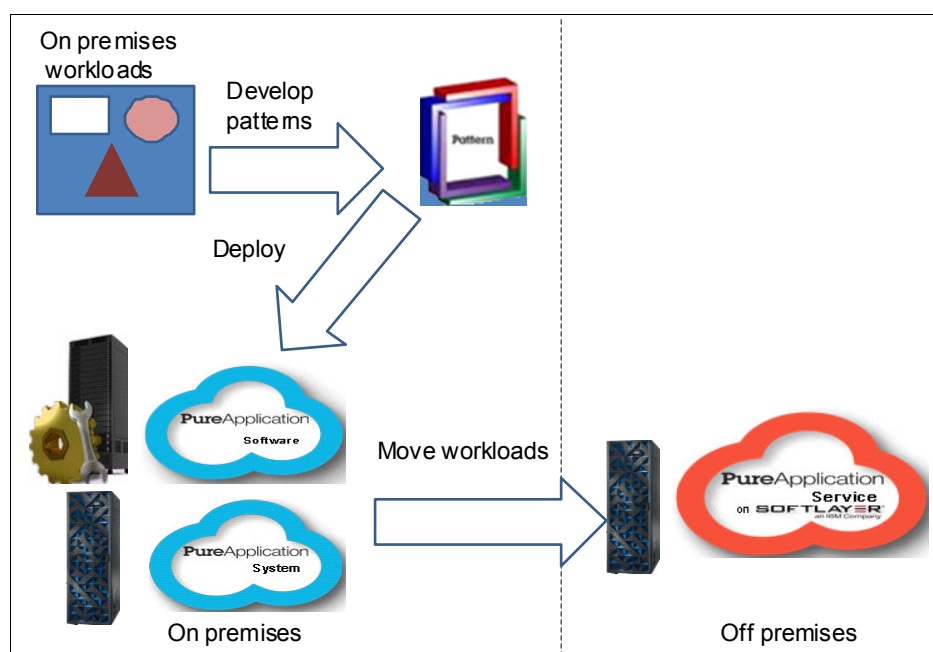


Figure 3-12 Strategy for moving workloads by using the PureApplication family

Enterprise users develop on-premises workloads by using pattern engine and deploy business applications with PureApplication System or PureApplication Software. With the patterns that are created on premises, for example, by using virtual system patterns with virtual images (with encryption features) and automation through script packages, the workload can be ready for a safer transition to PureApplication Software off premises. Typically, safer transition is accomplished by migrating workloads (those workloads that manage sensitive personal information) by using patterns that are protected by cryptographic methods. The workloads can be moved based on business criteria until all workloads are moved to the public cloud.

Considerations in a hybrid or full off-premises cloud scenario

While the enterprise moves workloads to the cloud, certain applications are off premises and other workloads are on premises. Based on this hybrid scenario, consider the following questions:

- ▶ Network considerations: How will the enterprise deal with network communication between on premises and off premises?
- ▶ Security: In a hybrid scenario, when all workloads are off premises, how will you address data security?
- ▶ System management: How will the system be managed in the hybrid scenario and in the off-premises scenario?
- ▶ Business continuity: How will you guarantee high availability and disaster recovery for applications in the hybrid cloud or in a complete off-premises workload?
- ▶ Resource planning: What do you need to run application XYZ?
- ▶ Chargeback: Is (internal) cross-charging required? What are the criteria?

Those points are explained with other use cases in 3.8, “Common technology implications of use cases” on page 131.

3.7 Primary on premises and business continuity off premises

Of great interest is the use case in which an enterprise uses a public cloud as a disaster recovery site for its on-premises data center.

In this use case, the on-premises components of the hybrid cloud are used for production or production-like setups. The off-premises components are used for non-production environments. If a disaster occurs in the client's data centers, the client must temporarily move its most critical applications to the public cloud to use all off-premises resources.

Certain characteristics are particular to this use case. This example is a disaster recovery use case because it is only activated in a disaster or a major outage at production sites. Therefore, we assume that this setup will not recover all applications, but only the necessary applications to keep the business running. These applications are built by using an IBM PureApplication pattern with an application that is fully contained, which means that IBM General Parallel File System (GPFS) or any other dependency to elements outside of the pattern is not necessary.

The preferred setup is illustrated in Figure 3-13 on page 130, in which a hybrid cloud consists of a combination of PureApplication System and IBM PureApplication Service, which are connected through a gateway.

Note: Figure 3-13 describes a use case where applications are on an IBM WebSphere Application Server and databases are in IBM DB2:

- ▶ If you are not familiar with IBM WebSphere Application Server terminology, see *WebSphere Application Server V8.5 Concepts, Planning, and Design Guide*, SG24-8022:
<http://www.redbooks.ibm.com/abstracts/sg248022.html?Open>
- ▶ If you are not familiar with IBM DB2 terminology, see *IBM DB2 9.5 for Linux, UNIX, and Windows documentation* in the IBM Knowledge Center:
<https://ibm.biz/BdHAAe>

Figure 3-13 shows the preferred setup in which a hybrid cloud consists of a combination of PureApplication System and IBM PureApplication Service, which are connected through a gateway.

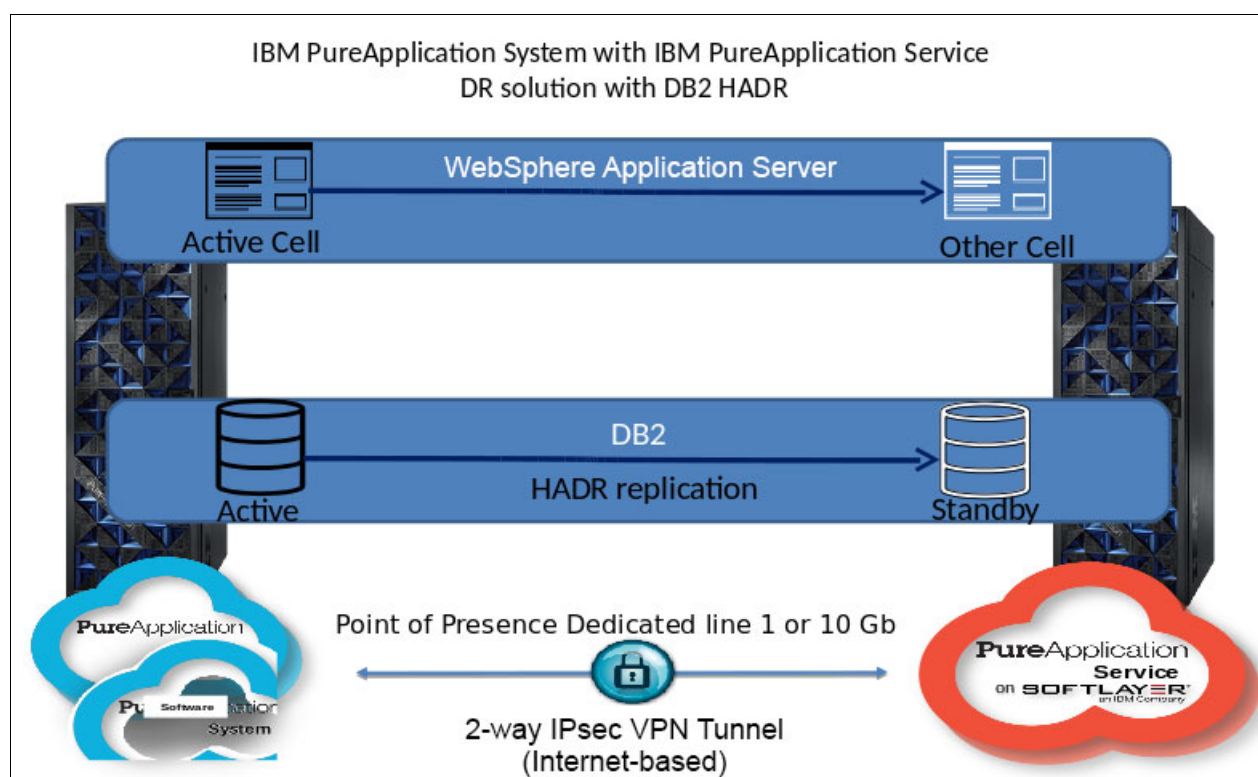


Figure 3-13 A potential disaster recovery use case

The database is kept in sync by using database replication. The choice between a 10 Gb or 1 Gb dedicated line largely depends on the type of applications that you need to run on the IBM PureApplication Service side, the number of those applications, and the amount of data that is required to be synchronized with the standby database.

On the disaster recovery side, the application layer consists of another cell in which only the WebSphere Deployment Manager is running to maintain the synchronization. At least two separate cells are used on this side so the synchronization must be manual by using `wsadmin` scripts or by applying every manipulation to both cells. This approach is not uniquely relevant to a hybrid cloud. It is part of the WebSphere Application Server characteristics. Multiple sources of information are available on the internet about this topic.

For more information, see the following websites:

- ▶ *High Availability and Disaster Recovery Options for DB2 for Linux, UNIX, and Windows*, SG24-7363:
<http://www.redbooks.ibm.com/abstracts/sg247363.html>
- ▶ *IBM WebSphere Application Server V8 Concepts, Planning, and Design Guide*, SG24-7957:
<http://publib-b.boulder.ibm.com/abstracts/sg247957.html?Open>
- ▶ *WebSphere Application Server V8.5 Administration and Configuration Guide for the Full Profile*, SG24-8056:
<http://publib-b.boulder.ibm.com/abstracts/sg248056.html?Open>

The standard setup with load balancers and IP-sprayers is at a higher level in this hybrid cloud infrastructure (not shown in Figure 3-13 on page 130). This setup is commonly used to balance the load between the on-premises production instances and the off-premises disaster recovery site.

When a disaster happens and the site where the IBM PureApplication System or PureApplication Systems in a multiple rack setup are lost, the following events occur:

- ▶ A failover is activated on the database.
- ▶ The application server nodes in the PureApplication Service cell are manually activated.
- ▶ The load is redirected to the PureApplication Service infrastructure.

This failover might also require an enterprise to change its external web traffic routing.

For more information, see the following websites:

- ▶ *WebSphere Application Server V8.5 Administration and Configuration Guide for the Full Profile*, SG24-8056 (includes information about WebSphere Application Server HADR setups):

<http://www.redbooks.ibm.com/abstracts/sg248056.html>

- ▶ *Implementing High Availability and Disaster Recovery in IBM PureApplication Systems V2*, SG24-8246 (includes information about HADR setups with IBM PureApplication Systems):

<http://www.redbooks.ibm.com/abstracts/sg248246.html>

To fully optimize your entire cloud environment, when you build or deploy your disaster recovery topology, consider separating parts. Separate the database from the application servers, for instance, to keep *only* the database running, while you start the application nodes when a reconfiguration is needed. This way, you can stop and store the application server instances. Restarting a stored instance takes only seconds or minutes versus a full restart (of all VMs) or redeployment, or versus a couple of hours to install. In addition, the capacity that is released this way can be repurposed until a real disaster occurs.

3.8 Common technology implications of use cases

This section summarizes all of the common implications of the use cases in this chapter from the perspective of a hybrid cloud that is implemented by using PureApplication System and PureApplication Service.

Table 3-4 groups all of the implications into the following categories:

- ▶ Networking
- ▶ Security
- ▶ System management
- ▶ Application lifecycle management (ALM)
- ▶ Business continuity
- ▶ Other

Table 3-4 Common technology implications of use cases

Topic	Implications
Networking	<p>Choose the correct connection option.</p> <p>PureApplication Service on SoftLayer offers three network options for establishing a connection between on-premises and off-premises data centers. Requirements are not the same for all use cases, especially the following use cases:</p> <ul style="list-style-type: none"> ▶ The front end is off premises, and the back end is on premises. ▶ Quick delivery. ▶ Start on premises, then move to the hybrid cloud. <p>Consider a high-speed connection, such as the point of presence (POP) option. This option is suggested if your company does not allow users to activate a VPN tunnel from their personal or company notebooks.</p> <p>For more information about network options that are offered by PureApplication Service on SoftLayer, see 4.1.3, “Connectivity use cases” on page 142.</p> <p>Application data and management data.</p> <p>Certain use cases require large network bandwidth between on premises and off premises, such as the following scenarios:</p> <ul style="list-style-type: none"> ▶ The front end is off premises, and the back end is on premises. ▶ Quick delivery. <p>In this case, it can be useful to separate the application data traffic from the management data traffic. The management data traffic can be generated by the following activities:</p> <ul style="list-style-type: none"> ▶ Exporting virtual patterns from on premises to off premises and vice versa ▶ Moving Docker images from a Docker Registry on premises to virtual system patterns off premises ▶ Moving Open Virtualization Format Archive (OVA) files from on premises to off premises and vice versa ▶ Moving log files from off premises to on premises <p>Application data traffic is generated by business-to-business integration when front-end components are on premises and back-end services are off premises.</p> <p>Because application data traffic is more critical, it can rely on a POP network connection. However, the management data can rely on different network options.</p>

Topic	Implications
Security	<p>Establish a secure connection between on premises and off premises.</p> <p>The encryption of data in transit is usually mandatory for applications in all of the use cases.</p> <p>If encryption of data at rest is required, particular software patterns components need to be licensed and included in your patterns.</p> <p>All network options that are provided by PureApplication Service on SoftLayer ensure data encryption. For more information, see 4.1.3, “Connectivity use cases” on page 142.</p>
	<p>Denial-of-service attack (DoS), firewalling, and vulnerability scan tests.</p> <p>Conducting security tests in an off-premises infrastructure and on applications is crucial, especially for use cases where your critical applications run on the public cloud.</p> <p>PureApplication Service on SoftLayer offers security services for DOS and firewalling. For more information, see this website:</p> <p>http://www.softlayer.com/security</p>
	<p>Security compliance.</p> <p>PureApplication Service on SoftLayer complies with various security standards. For more information, see 2.1.3, “Implementation” on page 31.</p>
	<p>Segregation and isolation in a multiple tenant environment.</p> <p>PureApplication Service on SoftLayer provides dedicated environments to the cloud users. However, due to security compliance, such as Payment Card Industry Data Security Standard (PCI DSS), you might consider a mixed domain:</p> <ul style="list-style-type: none"> ▶ PureApplication Service instances on SoftLayer ▶ Bare metal servers on a dedicated environment on SoftLayer <p>Bare metal servers can host the critical application components that require physical isolation due to PCI DSS security compliance.</p> <p>For more information about segregation and isolation, see 4.2, “Isolation and security” on page 159.</p>

Topic	Implications
	<p>Authentication, authorization, and auditing.</p> <p>Authentication, authorization, and auditing (AAA) services generally rely on LDAP, and they are applicable to application security and system management activities.</p> <p>A centralized LDAP between PureApplication System and PureApplication Service might simplify user management on the entire hybrid cloud.</p> <p>From the perspective of application security, the issue might be more delicate. Certain use cases might require a single sign-on (SSO) between off-premises components and on-premises components.</p>
	<p>Single sign-on (SSO) considerations in SaaS.</p> <p>In SaaS, consider security concerns, such as the SSO model. For more information about application-level concerns, see 2.6.3, “Security at the application level” on page 83.</p>

Topic	Implications
System management	<p>A consolidated monitoring view.</p> <p>A consolidated monitoring view of the entire hybrid cloud is always useful. For the following specific use cases, it is necessary:</p> <ul style="list-style-type: none"> ▶ The front end is off premises, and the back end is on premises. ▶ Quick delivery. <p>Transactions can involve application components on premises and off premises. The capability to easily correlate activities within the entire hybrid cloud can be crucial.</p> <p>PureApplication System and PureApplication Service can provide a consolidated monitoring view. For more information, see 4.6.2, “Application parts off premises, parts on premises, and orchestration” on page 209.</p>
	<p>OS and middleware fix management.</p> <p>A common strategy for OS and middleware fix management can be applied to the whole hybrid cloud. You can use the Red Hat Satellite service on both PureApplication System and PureApplication Service. For more information, see 2.3.1, “Pattern engine” on page 50.</p>
	<p>Backup and recovery.</p> <p>Backup and recovery are always necessary. However, depending on the use case, the backup and recovery activities might have different implications:</p> <ul style="list-style-type: none"> ▶ Off-premises backups can correlate to on-premises backups ▶ Backups can be used to move environments or data from off premises to on premises and vice versa <p>For more information about the implementation of a backup and recovery strategy with PureApplication System and PureApplication Service, see 4.6.1, “Test and development off premises and production on premises” on page 201.</p>
	<p>OS platform migration considerations.</p> <p>Enterprises that are migrating to a cloud environment need to consider their older applications. Today, the supported platforms are x86 and Power. If enterprises use hardware platforms that are not supported, the applications must be migrated with supported platforms, with a completed business application refactoring lifecycle, if necessary.</p>
Application lifecycle management (ALM)	<p>ALM in a hybrid cloud.</p> <p>The use of ALM that covers the whole hybrid cloud is necessary for all use cases.</p> <p>For more information about ALM and the hybrid cloud, see 2.1.3, “Implementation” on page 31.</p>

Topic	Implications
Business continuity	<p data-bbox="748 254 1187 279">Disaster recovery of the off-premises site.</p> <p data-bbox="748 310 1406 394">If you have mission-critical applications on a public cloud, you must adopt a business continuity strategy for the cloud service provider data center.</p> <p data-bbox="748 426 1406 510">PureApplication Service is available on several SoftLayer data centers around the world, which offers you the opportunity to consider different business continuity approaches:</p> <ul data-bbox="748 514 1406 657" style="list-style-type: none"> <li data-bbox="748 514 1406 598">▶ One active PureApplication Service instance on one SoftLayer data center, and one standby instance in another SoftLayer data center <li data-bbox="748 602 1406 657">▶ Two active PureApplication Service instances that are distributed to different SoftLayer data centers <p data-bbox="748 682 1094 707">Business continuity off premises.</p> <p data-bbox="748 739 1427 850">A holistic approach to a disaster recovery strategy where your secondary site is off premises is a complex topic. However, such a scenario usually requires common requirements, planning, and design, for instance:</p> <ul data-bbox="748 854 1427 1026" style="list-style-type: none"> <li data-bbox="748 854 1427 938">▶ Virtual patterns must cover all that you need to build your infrastructure (operating systems, software, and configuration steps). <li data-bbox="748 942 1427 1026">▶ The off-premises site can count on infrastructure resources from PureApplication Service on SoftLayer too, not only the PureApplication Service instances.
Other	<p data-bbox="748 1056 954 1081">Resource planning.</p> <p data-bbox="748 1113 1427 1194">Resource planning includes the need to address quickly expected or unexpected resources in the immediate future to the need to predict long-term future requirements.</p> <p data-bbox="748 1199 1427 1398">The public (and hosted private) clouds promise to be more agile in acquiring resources and disposing of resources, when compared to the agility of the procurement process of the on-premises side of the hybrid cloud. (Alternatively, adding on-premises resources to the PureApplication System can be as simple as IBM inserting two or more compute nodes in a “hot” rack with available bays.)</p> <p data-bbox="748 1402 1427 1484">Accurate resource planning and the awareness of the scalability that is offered by the entire hybrid cloud solution affects the efficiency of an enterprise’s IT.</p> <p data-bbox="748 1488 1427 1631">Thinking of the use case “start on premises, then move to the hybrid cloud” where the strategy is to move the entire data center to the cloud, we realize that the form and success of the enterprise’s life relies on the resource planning and the scalability of the cloud service provider.</p> <p data-bbox="748 1635 1427 1778">Chapter 1, “Enabling a hybrid enterprise” on page 1 describes the way to acquire new resources on PureApplication Service, and how the monitoring features of the PureApplication family can support the capacity management and resource planning practices.</p>

Topic	Implications
Other	<p>Chargeback (and showback).</p> <p>IT was considered a cost center for decades. The necessity to adopt a chargeback model is historical. However, applying this mechanism to a traditional data center is not trivial. Moreover, think of the new rise and diffusion of server, network, and storage virtualization. It is almost certain that understanding how to assign the cost of its shared physical resources and application processing accurately to the LOB can get more difficult. However, the cloud is changing the game.</p> <p>The cloud computing model gives you the option of a chargeback approach, which is part of the definition of a cloud, and the cloud makes it easy to acquire new IT resources. Ironically, though, this option exposes an enterprise to the risk of an unjustified explosion of resources and charges, which already happened with IT shops that forgot to turn off the resources. Alternatively, the cloud can simplify the chargeback analysis, which is true for the off-premises side of the hybrid cloud, where it is easier to isolate the cost of resources for a single project.</p> <p>To go back to the described use cases, certain scenarios require a complex chargeback analysis due to the hybrid nature of IT, as in the case of the “front end off premises, back end on premises” use case. Other scenarios require the transition from one chargeback technique to another chargeback technique, as in the case of the “start on premises, then move to the hybrid cloud” use case.</p> <p>The PureApplication family provides tools and reports to support the chargeback. For more information, see the IBM Knowledge Center:</p> <p>https://ibm.biz/BdHhin</p>



Implementation considerations

Earlier chapters in this book outline several general concepts about the cloud, introduce the ideas and concepts of an adoption roadmap, and focus on the requirements to build a hybrid cloud by using IBM PureApplication products. Also, Chapter 3, “Hybrid use cases” on page 105, includes many use cases, each of which is illustrated with practical approaches.

This chapter describes more of the practical aspects of deploying middleware by using a hybrid cloud solution. The goal is to present a comprehensive overview of key questions to ask, topics to consider, and the respective answers when you bring it all together in a hybrid cloud solution, which is mapped to the use cases in Chapter 3, “Hybrid use cases” on page 105. Although we illustrate several step-by-step tasks in this chapter, this chapter is not intended to be a complete how-to tutorial.

This chapter covers the following topics:

- ▶ Connectivity
- ▶ Isolation and security
- ▶ Portability
- ▶ Recoverability
- ▶ Quick delivery use case with TradeLite
- ▶ Deployment and bringing it all together

4.1 Connectivity

Hybrid computing is a contemporary extension of distributed computing. With PureApplication System and PureApplication Service on SoftLayer, computation occurs in a common, well-maintained, and supported computing management framework. PureApplication Service extends your data center topology to the resources and capabilities of IBM SoftLayer.

For this book, a hybrid cloud involves servers that are housed in at least two data centers. Theoretically, it might be possible to have two different clouds in one data center, but that design is not within the scope of this book. (Two PureApplication Systems, which are used in one data center for multi-rack deployment, are considered as one cloud.) The client's data center is on premises, and the infrastructure of PureApplication Service on SoftLayer is off premises. The internet provides the public network for communication to flow between on-premises and off-premises systems.

PureApplication Service on SoftLayer has a dedicated team that is responsible for the configuration of the network on the off-premises side. This group is referred to as the *PureApplication Service on SoftLayer DevOps Team*, or *DevOps Team*. This team will contact you for information about how to communicate with your on-premises networking team. Together, you will establish a configuration for server-to-server communication.

Note: When you place an order for PureApplication Service, be sure to note any existing SoftLayer accounts so that the DevOps Team can configure gateways and firewalls to integrate your existing environments into the new service.

4.1.1 IBM PureApplication Service connectivity

IBM PureApplication Service is provisioned on an IBM SoftLayer cloud. IBM PureApplication Service provides pattern-based workload deployment services that use IBM middleware, non-IBM middleware, and applications that can be deployed and managed easily. IBM PureApplication Service contains various components, including a pattern deployment engine, which is responsible for the deployment of patterns, hypervisors, and storage.

Figure 4-1 shows a typical PureApplication Service instance.

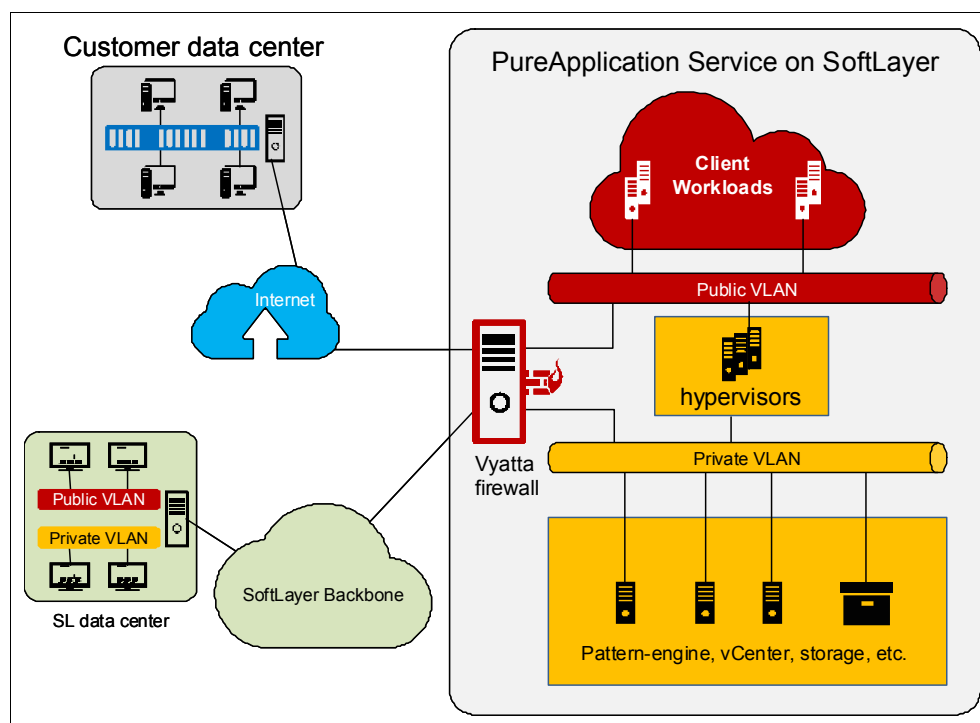


Figure 4-1 Typical PureApplication Service instance

Two networks are in a PureApplication Service instance:

- ▶ The private network contains, for example, the pattern-engine, vCenter, and storage.
- ▶ The public network contains client workloads.

The private network also connects to a high-speed SoftLayer backbone network. Servers in the private network from one SoftLayer instance to another SoftLayer instance use the high-speed SoftLayer backbone network to communicate.

When a PureApplication Service instance is provisioned, you will be able to access and manage the pattern deployment engine. You also need to access the deployed middleware and application instances. In certain cases, the deployed instances need access to services outside of the IBM SoftLayer cloud.

This chapter describes different options that are available for you to gain access to the IBM PureApplication Service instance and deployed instances that are running inside the IBM SoftLayer cloud. This chapter also provides connection options for deployed instances to access services outside of the IBM SoftLayer cloud where the instance is deployed.

4.1.2 Reasons that connectivity is required

Connectivity between IBM PureApplication Service and your data center site is required for the following reasons:

- ▶ Deployment automation software that is running in your data center needs to manage IBM PureApplication Service.
- ▶ Hosts that are running in your data center need to perform pattern deployment.
- ▶ Hosts that are running in your data center need to access deployed instances.

- ▶ Deployed instances must access database services, Lightweight Directory Access Protocol (LDAP) services, and domain name system (DNS) services that are outside of the IBM SoftLayer cloud. For example, your data center already has those services and wants to reuse them.

4.1.3 Connectivity use cases

Several options are available for establishing connectivity between your site and a PureApplication Service site:

- ▶ OpenVPN connection
- ▶ Site-to-site Internet Protocol Security (IPsec) virtual private network (VPN) connection
- ▶ Direct Link from your network provider to SoftLayer point of presence (POP)

The use cases in this chapter have different qualities of service. Use the following sections to guide you in determining the use case that is more appropriate for your environment.

OpenVPN client connection

An OpenVPN server is configured on the IBM SoftLayer site where PureApplication Service is provisioned. At the same time, client-side certificates and configuration files are generated and packaged. The client-side certificates and configuration files are sent to you after PureApplication Service provisioning is complete. You are responsible for installing an OpenVPN client and using the provided configuration files to establish an OpenVPN connection.

An OpenVPN client needs to be set up in each host in your data center that needs access to PureApplication Service by using OpenVPN. The same client-side configuration files and certificates can be used for every connection from your data center hosts. As shown in Figure 4-2, this approach provides a unidirectional client to the PureApplication Service connection.

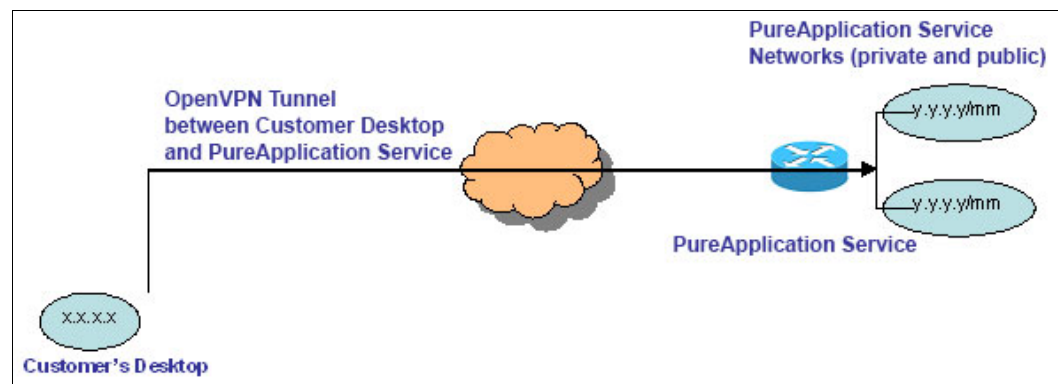


Figure 4-2 Unidirectional client to PureApplication Service connection

When the VPN is established, OpenVPN Server on PureApplication Service pushes the routes that are required for you to access both PureApplication Service private and public networks. Firewall rules are also set up on the gateway so that you can access the pattern engine in the private network and client workloads in the public network, directly from your desktop.

Scenario 1

A client wants to establish a one-way connection to the PureApplication Service site from a desktop. For this scenario, Figure 4-3 depicts how your notebook can use a VPN tunnel to connect to the SoftLayer network.

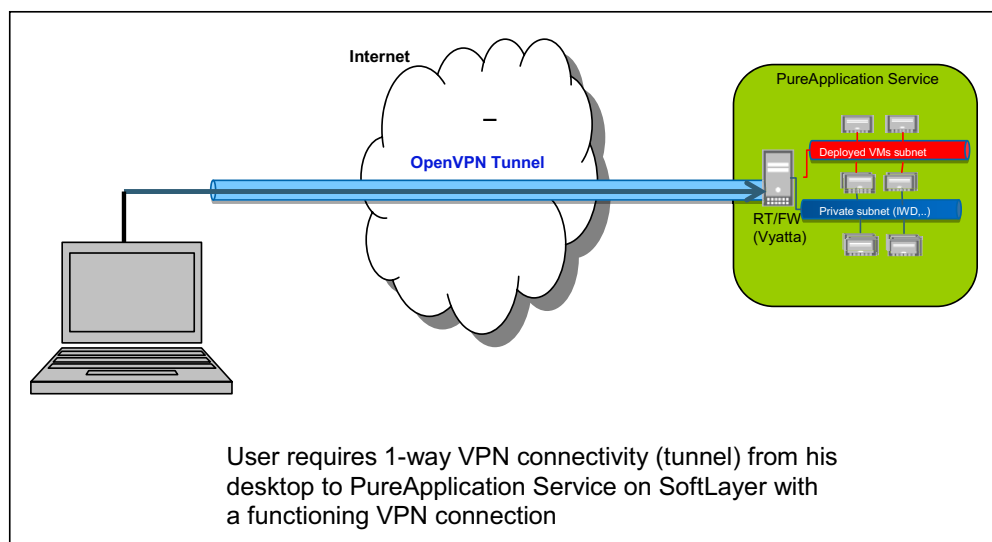


Figure 4-3 Client-initiated VPN tunnel

Drawbacks

A considerable drawback of this option is that it is unidirectional, and it needs to be configured on every client system that needs to connect to PureApplication Service. Also, this option does not allow connectivity of deployed instances to resources outside of the cloud in which they are deployed.

Site-to-site IPsec VPN connection

After PureApplication Service is provisioned, you optionally can establish a site-to-site IPsec VPN connection by submitting an IPsec VPN Request Form, which contains the required information to set up a site-to-site IPsec VPN. When it is returned to you, it contains the PureApplication Service site information. We suggest that you specify your site information in the IPsec VPN Request Form and send it back to the IBM DevOps team. You and the IBM DevOps team will work together to establish the site-to-site IPsec VPN between the two sites.

In this case, a site-to-site IPsec VPN connection is established between the gateway at your site and the gateway at the PureApplication Service side.

After the site-to-site VPN is configured, communication is established between each subnet on one site to all of the other subnets on the other site.

One tunnel is created for each subnet-to-subnet connection, as shown in Figure 4-4.

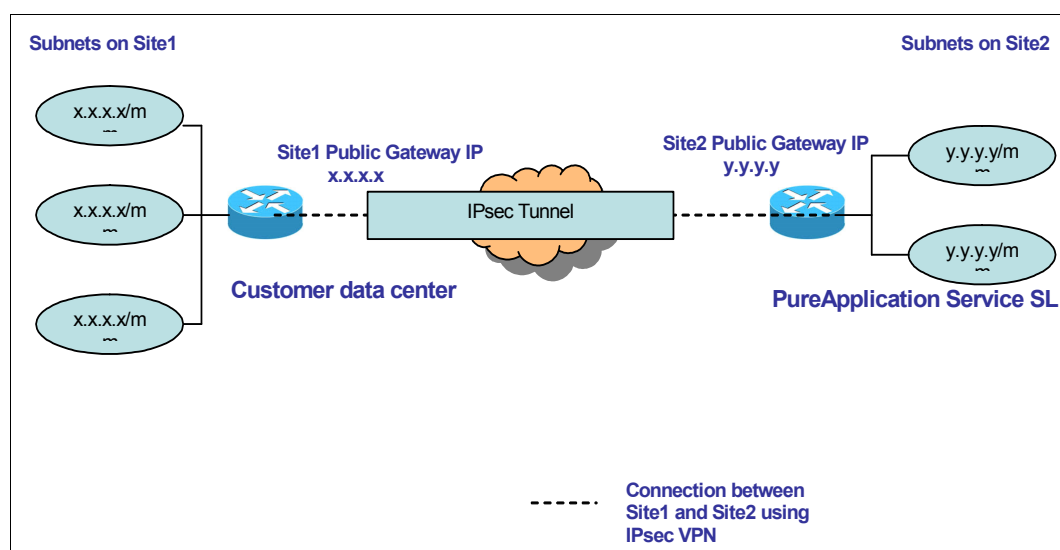


Figure 4-4 Site-to-site connection by using IPsec VPN

The following sections describe different scenarios in which this option can be used.

Scenario 1

A user wants to establish a connection between an onsite data center and the PureApplication Service instance. This type of connection is required mainly for you to access the PureApplication Service management console and the deployed applications on the PureApplication Service site. This type of connection is also required for deployed applications at the PureApplication Service site to access enterprise applications that are running in your data center for databases and central services, including LDAP. See Figure 4-5.

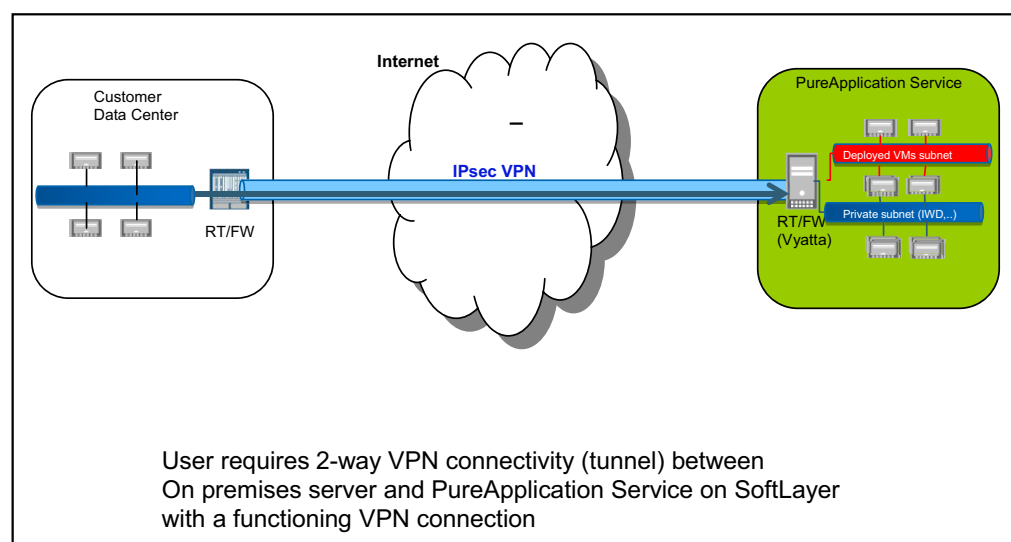


Figure 4-5 An on-premises and PureApplication Service on SoftLayer connection

Scenario 2

A user wants to establish a connection between a SoftLayer instance and the PureApplication Service instance. This type of connection is required for applications that run in a SoftLayer Data Center (for example, IBM Bluemix applications that run inside SoftLayer) to access PureApplication Service. See Figure 4-6.

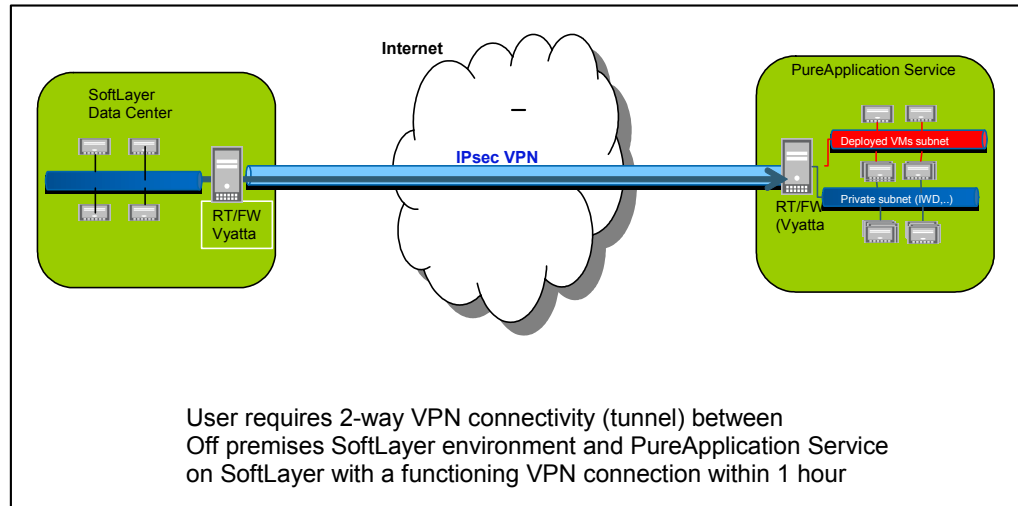


Figure 4-6 Bidirectional VPN tunnel

Scenario 3

A user wants to expand an existing PureApplication Service instance by creating another PureApplication Service instance and establishing a connection between them. This type of connection can be used to expand the PureApplication Service network capacity, as shown in Figure 4-7 on page 146.

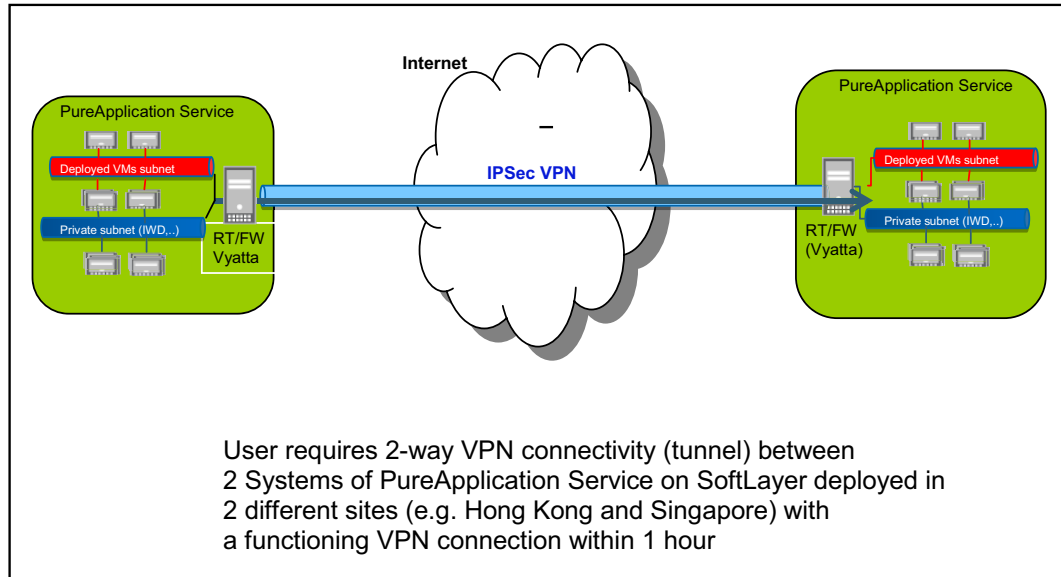


Figure 4-7 SoftLayer to SoftLayer VPN tunnel

Drawbacks

The process of establishing a site-to-site IPsec VPN connection involves multiple vendor equipment and multiple teams from each company. Setting up the connection requires coordination and clear lines of communication between your team and the IBM DevOps team. Because this connection is an internet connection, the speed depends on the distance between the two sites and any internet latency.

Establishment of site-to-site IPsec VPN

In PureApplication Service, the process of establishing a site-to-site IPsec VPN is a manual process, and it requires that you and IBM work together. Follow these steps to set up the site-to-site IPsec VPN connection:

1. IBM creates an IPsec VPN Request Form and sends it to you.
2. You add your network information to the IPsec VPN Request Form and return it to IBM.
3. You and IBM set up a call to establish site-to-site IPsec VPN.
4. IBM sets up IPsec VPN on a PureApplication Service site.
5. You set up IPsec VPN at your site.
6. Verify the IPsec VPN connection.

Figure 4-8 is a sample IPsec VPN Request Form that IBM sent to a client.

	IBM PureApplicationService	Customer Site
Gateway IP	150.10.10.1	
IKE phase 1		
Key Exchange Method (AES/DES/3DES)	AES256	
Hashing (SHA/MD5)	SHA	
Authentication (PSK/x509)	PSK	
Lifetime in mins	14000m	
DH Group (2, 5)	2	
IKE Phase 2		
Encryption (AES/DES/3DES)	AES256	
Hashing (SHA/MD5)	SHA	
PFS (enable/disable)	Enable	
PFS DH Group (2, 5)	2	
Lifetime in secs	3600	
Protocol (ESP/AUTH)	ESP	
Mode (Tunnel/Transport)	Tunnel	
Encryption Domains	10.100.10.0/24,150.10.10.0/24	
Firewall Rules	10.100.10.2:80,443	
	150.10.10.0/24:80,443,22	

Figure 4-8 A sample, completed IPsec VPN Request Form that was sent to a client

Figure 4-9 is an example IPsec VPN Request Form that was returned to IBM from a client.

	IBM PureApplicationService	Customer Site
Gateway IP	150.10.10.1	200.10.10.1
IKE phase 1		
Key Exchange Method (AES/DES/3DES)	AES256 -> 3DES	3DES
Hashing (SHA/MD5)	SHA	SHA
Authentication (PSK/x509)	PSK	PSK
Lifetime in mins	14000m	14000m
DH Group (2, 5)	2	2
IKE Phase 2		
Encryption (AES/DES/3DES)	AES256	AES256
Hashing (SHA/MD5)	SHA	SHA
PFS (enable/disable)	Enable	Enable
PFS DH Group (2, 5)	2	2
Lifetime in secs	3600	3600
Protocol (ESP/AUTH)	ESP	
Mode (Tunnel/Transport)	Tunnel	
Encryption Domains	10.100.10.0/24,150.10.10.0/24	10.200.10.0/24
Firewall Rules	10.100.10.2:80,443	10.200.10.0/24:636,1521
	150.10.10.0/24:80,443,22	

Figure 4-9 A sample IPsec VPN Request Form that was returned from a client to IBM

Use Direct Link from your network to an IBM SoftLayer point of presence

A site-to-site connection that uses IPsec VPN can be slow and unreliable. Certain clients do not like the connection over the internet, even though a secure tunnel exists between the two sites. To overcome the deficiencies of a site-to-site IPsec VPN connection, the Direct Link option is provided for clients that must have a secure, reliable, high-speed connection.

IBM SoftLayer data center locations are interconnected by a dedicated private SoftLayer backbone network. The SoftLayer backbone network provides a high-speed, private network connection between various SoftLayer data centers. Many points of presence (POPs) are in the SoftLayer backbone network. POP is where you can connect your network with the SoftLayer backbone network.

Figure 4-10 shows how your data center connects to PureApplication Service in a SoftLayer data center by using both IPsec VPN and Direct Link. This dual connection strategy provides a backup connection if the Direct Link is unavailable.

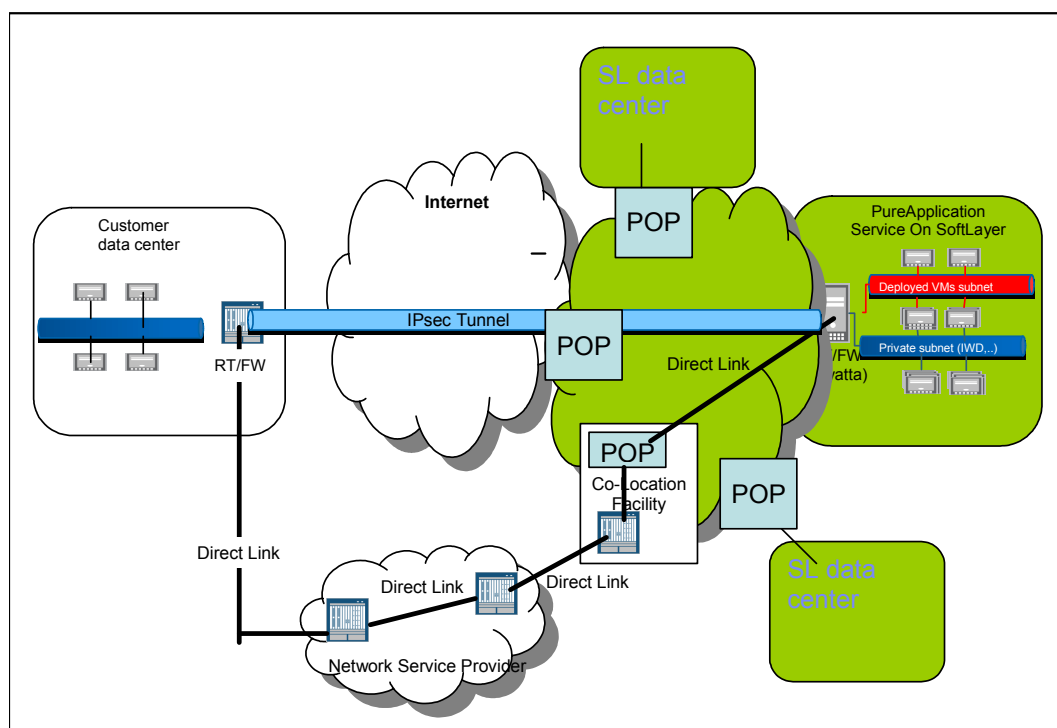


Figure 4-10 Connecting to the SoftLayer cloud by using Direct Link

The IPsec tunnel is established over the Internet. Direct Link is established by connecting your data center to a POP with your network service provider using a dedicated link. After your data center is connected to the SoftLayer POP, a connection to any SoftLayer site from your network is established by using the SoftLayer backbone network.

The following steps establish your site to a PureApplication Service site connection by using POP:

1. Submit a request to IBM for a Direct Link. The request needs to contain the POP site that you want to connect to and the speed at which you want to connect (1 Gbps or 10 Gbps).
2. Contact your network service provider to establish a dedicated link from your site to the POP. The dedicated link can be T1 or T3 lines, depending on your network service provider.
3. Install your router at the POP.
4. Request a connection for your router to the SoftLayer router by contacting the POP provider. Use a cross-connect fiber link (1 Gbps or 10 Gbps).

We suggest that you use an IPsec tunnel on top of a Direct Link to ensure that your data is secure.

Drawbacks

The Direct Link takes the longest time to implement because it involves different providers to supply and install a physical cable along the path from your data center to a POP. You incur the cost of installing the physical cables and any monthly rates that the network provider charges.

4.1.4 Installing and configuring OpenVPN

After PureApplication Service is provisioned, you receive a welcome email with information about setting up the OpenVPN client and the URL of the PureApplication Service management console. You can set up the OpenVPN client, connect to the PureApplication Service management console, and start deployment.

Installing OpenVPN for Windows

Today, most software is 64 bit. OpenVPN is no different. For Windows 7 systems and later, download the 64-bit version of OpenVPN. The software is available at no charge to use. You can download the software at <http://www.openvpn.net>.

Follow these steps:

1. Select **Installer (64-bit), Windows Vista and later version**, which is depicted in Figure 4-11.

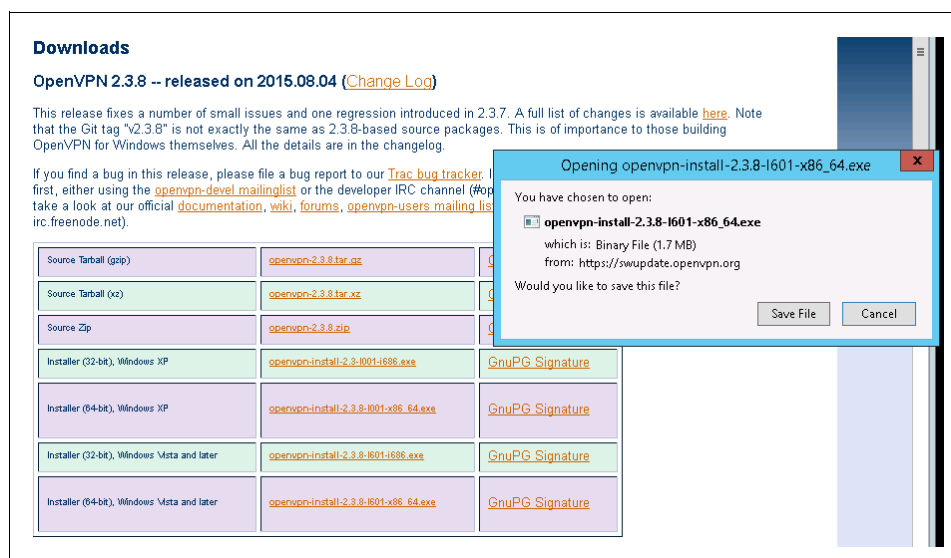


Figure 4-11 Downloading OpenVPN

- OpenVPN uses the standard Windows installer. Start the installation process and select all of the default values, as shown in Figure 4-12.

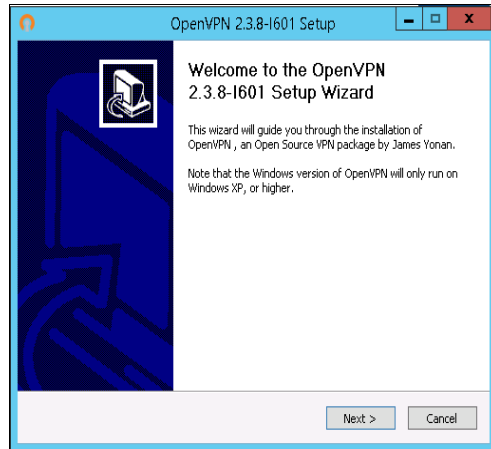


Figure 4-12 The Setup Wizard for OpenVPN

- When the installer completes, an icon shows on the desktop. To start the VPN client, right-click the icon, and select **Run as administrator**. Even though the VPN client starts, no graphical user interface (GUI) appears. However, an OpenVPN icon shows in the system tray.
- When the icon turns green, the VPN client is ready. To connect to PureApplication Service, use Windows Explorer to locate the configuration files that you received when PureApplication Service was provisioned. Right-click the .ovpn file type, and select **Start OpenVPN on this config file**, as shown in Figure 4-13.

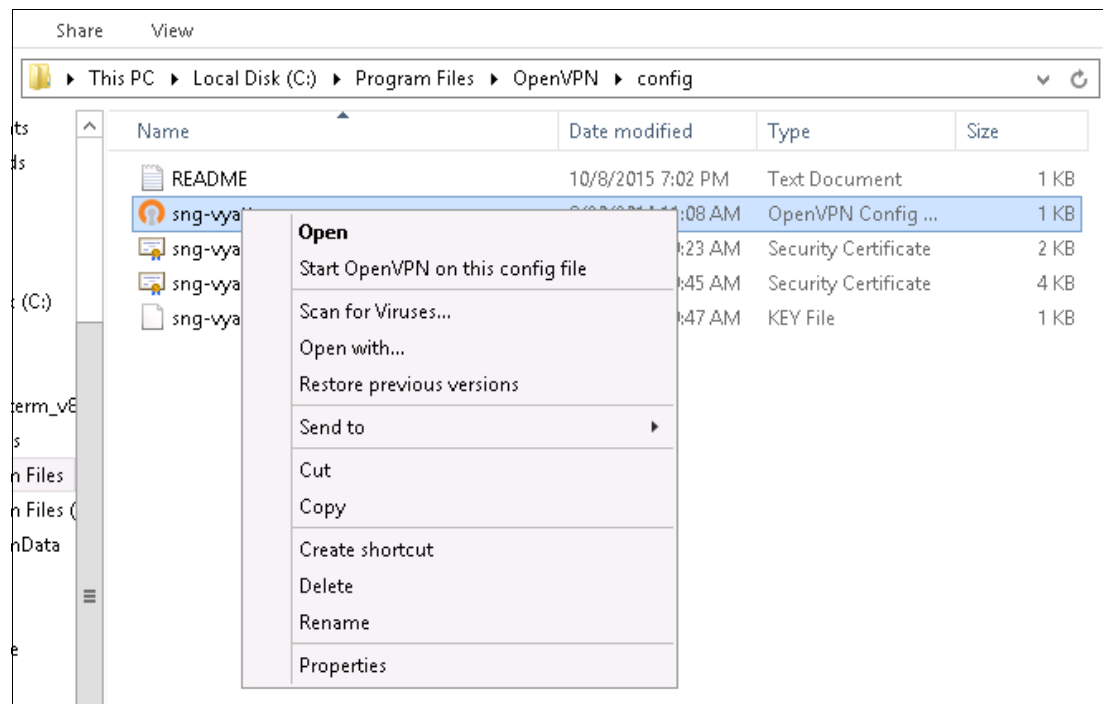


Figure 4-13 Starting a OpenVPN session

- VPN starts in a new command window. When Initialization Sequence Complete is visible, the VPN tunnel is active. You can open a browser window and connect to the PureApplication Service console. Figure 4-14 shows that initialization completed. To terminate the tunnel, press the F4 key or terminate the command window.

```

[C:\Program Files\OpenVPN\config\sng-vyatta.ovpn] OpenVPN 2.3.8 F4:EXIT F1:...
Thu Oct 08 19:28:40 2015 C:\Windows\system32\route.exe ADD 119.81.33.192 MASK 25
5.255.255.224 10.255.255.1
Thu Oct 08 19:28:40 2015 ROUTE: CreateIpForwardEntry succeeded with dwForwardMet
ric1=20 and dwForwardType=4
Thu Oct 08 19:28:40 2015 Route addition via IPAPI succeeded [adaptive]
Thu Oct 08 19:28:40 2015 C:\Windows\system32\route.exe ADD 119.81.108.32 MASK 25
5.255.255.224 10.255.255.1
Thu Oct 08 19:28:40 2015 ROUTE: CreateIpForwardEntry succeeded with dwForwardMet
ric1=20 and dwForwardType=4
Thu Oct 08 19:28:40 2015 Route addition via IPAPI succeeded [adaptive]
Thu Oct 08 19:28:40 2015 C:\Windows\system32\route.exe ADD 119.81.107.64 MASK 25
5.255.255.224 10.255.255.1
Thu Oct 08 19:28:40 2015 ROUTE: CreateIpForwardEntry succeeded with dwForwardMet
ric1=20 and dwForwardType=4
Thu Oct 08 19:28:40 2015 Route addition via IPAPI succeeded [adaptive]
Thu Oct 08 19:28:40 2015 C:\Windows\system32\route.exe ADD 119.81.30.128 MASK 25
5.255.255.224 10.255.255.1
Thu Oct 08 19:28:40 2015 ROUTE: CreateIpForwardEntry succeeded with dwForwardMet
ric1=20 and dwForwardType=4
Thu Oct 08 19:28:40 2015 Route addition via IPAPI succeeded [adaptive]
Thu Oct 08 19:28:40 2015 C:\Windows\system32\route.exe ADD 119.81.34.160 MASK 25
5.255.255.224 10.255.255.1
Thu Oct 08 19:28:40 2015 ROUTE: CreateIpForwardEntry succeeded with dwForwardMet
ric1=20 and dwForwardType=4
Thu Oct 08 19:28:40 2015 Route addition via IPAPI succeeded [adaptive]
Thu Oct 08 19:28:40 2015 Initialization Sequence Completed

```

Figure 4-14 OpenVPN started

Setting up OpenVPN for CentOS

Linux on the desktop is now mainstream. One common Linux distribution is CentOS. In this example, a CentOS 7.1 virtual machine (VM) is used to illustrate an OpenVPN installation. Follow these steps:

- A CentOS installation does not provide the required Extra Packages for Enterprise Linux (EPEL) package library. Use the **yum** command to add the EPEL to the repository list as shown in Example 4-1.

Example 4-1 Obtain EPEL (an OpenVPN installation prerequisite)

```
$sudo yum --assumeyes install epel-release
```

The command in bold in Example 4-2 provides a repository list that includes the EPEL.

Example 4-2 Validate that the EPEL is available

```

$sudo yum repolist
Loaded plugins: fastestmirror, langpacks
Loading mirror speeds from cached hostfile
* base: mirror.team-cymru.org
* epel: mirror.umd.edu
* extras: mirrors.chkhosting.com
* updates: bay.uchicago.edu
repo id          repo name          status
!base/7/x86_64   CentOS-7 - Base    8,652
!epel/x86_64     Extra Packages for Enterprise Linux 7 - 8,536
!extras/7/x86_64 CentOS-7 - Extras   214
!updates/7/x86_64 CentOS-7 - Updates  1,499
repolist: 18,901

```

2. Now that the EPEL is defined to the CentOS instance, you can issue the **sudo yum --assumeyes install openvpn** command to install the OpenVPN client. Example 4-3 illustrates the installation of the VPN client.

Example 4-3 Highlights of the messages from yum when you install OpenVPN

```
sudo yum --assumeyes install openvpn
...
Resolving Dependencies
--> Running transaction check
---> Package openvpn.x86_64 0:2.3.8-1.el7 will be installed
--> Processing Dependency: libpkcs11-helper.so.1()(64bit) for package:
openvpn-2.3.8-1.el7.x86_64
Transaction test succeeded
...
Running transaction
  Installing : pkcs11-helper-1.11-3.el7.x86_64
1/2
  Installing : openvpn-2.3.8-1.el7.x86_64
2/2
  Verifying : pkcs11-helper-1.11-3.el7.x86_64
1/2
  Verifying : openvpn-2.3.8-1.el7.x86_64
2/2

Installed:
openvpn.x86_64 0:2.3.8-1.el7

Dependency Installed:
pkcs11-helper.x86_64 0:1.11-3.el7
```

Complete!

3. The PureApplication Service on SoftLayer DevOps team provides you with the following files:
 - sng-vyatta-ca.crt
 - sng-vyatta-general.crt
 - sng-vyatta-general.key
 - sng-vyatta.ovpn
4. A convenient location for these files is /etc/openvpn. You might choose to add the following alias (Example 4-4) to your ~/.bashrc file for convenience.

Example 4-4 Alias to invoke OpenVPN

```
alias openvpn='sudo openvpn --config /etc/openvpn/sng-vyatta.ovpn --ca
/etc/openvpn/sng-vyatta-ca.crt --key /etc/openvpn/sng-vyatta-general.key
--cert /etc/openvpn/sng-vyatta-general.crt'
```

5. Start OpenVPN from a terminal session, as shown in Example 4-5.

Example 4-5 Start OpenVPN command for CentOS

```
$openvpn
Fri Oct 16 15:09:31 2015 OpenVPN 2.3.8 x86_64-redhat-linux-gnu [SSL (OpenSSL)]
[LZO] [EPOLL] [PKCS11] [MH] [IPv6] built on Aug  4 2015
Fri Oct 16 15:09:31 2015 library versions: OpenSSL 1.0.1e-fips 11 Feb 2013, LZ0
2.06
...
Fri Oct 16 15:09:46 2015 Initialization Sequence Completed
```

6. You can now open a browser window and connect to the PureApplication Service console.
7. To end the VPN session, you can terminate the terminal session or issue the Ctrl+C command at the command prompt.

4.1.5 Data center connectivity

After the OpenVPN client is set up, you can access the PureApplication Service on SoftLayer environment as a client. That is, the communication is one way. The client must initiate the connection to a server that is in the SoftLayer network. To allow deployed applications to access SoftLayer resources, you need to create an IPsec tunnel to connect the two network infrastructures. You connect by using an IPsec gateway appliance or a deployed VM in an on-premises PureApplication System.

An IPsec gateway appliance is the preferred method to connect the two networks. However, you can use a deployed VM gateway to establish an IPsec VPN gateway to the SoftLayer network quickly. The use of a hardware-based IPsec tunnel can provide better performance and a better long-term solution at the expense of procuring hardware and facing a longer lead time.

On-premises applications might need to augment the VM kernel IP routing table information that was deployed by PureApplication System with a route to your gateway. This step requires root authority. The command is shown in Example 4-6.

Example 4-6 Route command for VM-based gateway

```
route add -net <high_order_subnet_octets>.32/27 gw <gateway_high_order_octets>.206
dev eth1
```

The DevOps Team will provide information that is specific to your configuration.

An alternative to directly configuring the deployed VM's network configuration is to add the additional routing information to the IP group. A route to each subnet in the SoftLayer environment can be added to the routing information in the IP group.

Follow these steps to add a network route:

1. From the PureApplication Service console, navigate to **Cloud** → **IP Groups**, and click the IP group that you want.
2. Scroll down to **Routes** and click the plus sign (+).

3. Enter the following values, as shown in Figure 4-15:
 - Subnet value in dotted decimal notation
 - Netmask for the subnet
 - IP address of the gateway
4. Select **OK** to save the new route.

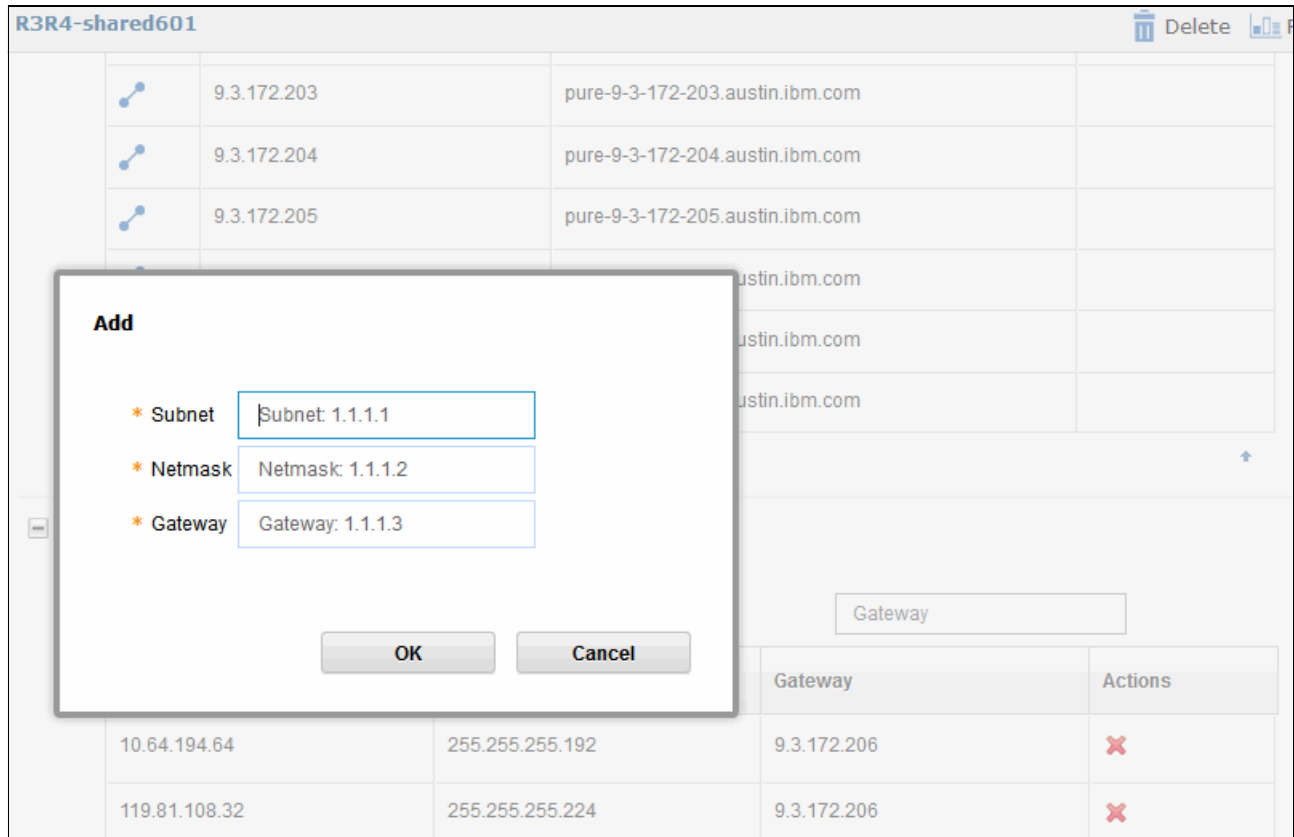


Figure 4-15 Configuring a network route for an IP group

The off-premises deployed VM does not need route statements. They are provided by the SoftLayer infrastructure.

Each network side of your hybrid cloud has a DNS service. Your DNS service contains entries for IP addresses within your network, and the SoftLayer DNS contains entries for the IP addresses within the SoftLayer network. PureApplication Service on SoftLayer does not update its DNS with your network IP addresses. You can choose to update your DNS with IP entries of addresses within the SoftLayer network.

Network Time Protocol (NTP) is another common service that is provided by your data center. An accurate time is critical because it provides the only frame of reference among all devices in the network. Without an accurate time, synchronizing transactions and the correlation of log files is not possible. SoftLayer provides the robust NTP infrastructure for PureApplication Service.

4.1.6 Network configuration for the TradeLite example

Communication about the network frequently benefits from a diagram that contains the components of immediate interest and how these pieces relate to one another. Figure 4-16 illustrates a high-level view of the network that was used for the TradeLite application. PureApplication System hosts are on a local area network (LAN) that is separate from the PureApplication Service network segment. For this project, PureApplication Service and PureApplication System are in different countries. By using the Internet as our public network, the two-way communication occurs in a secure and private manner within an IPsec tunnel.

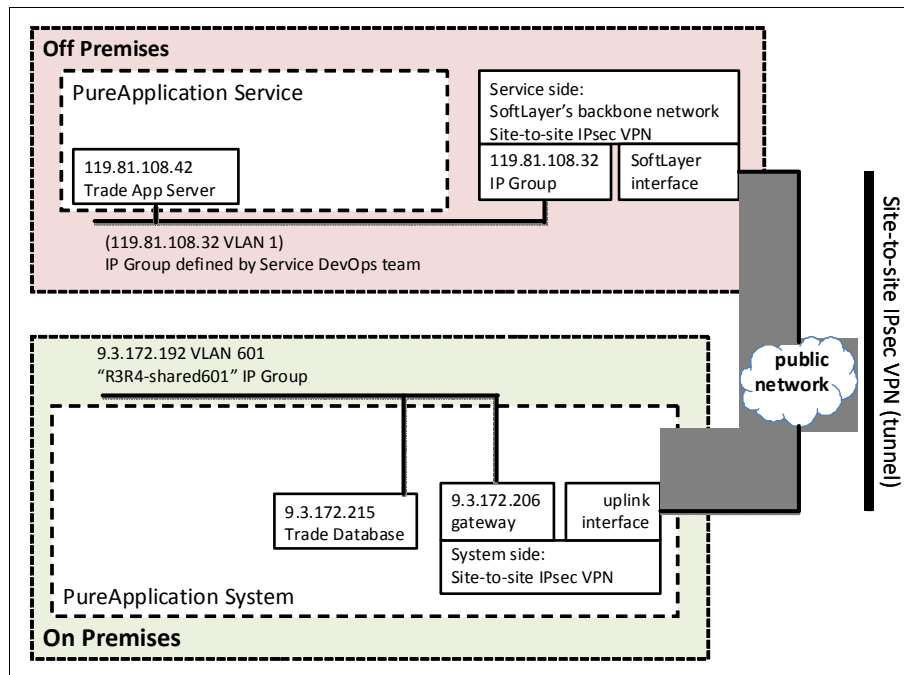


Figure 4-16 Network that connects PureApplication System to PureApplication Service on SoftLayer

On the PureApplication System side, a Red Hat Enterprise Linux (RHEL) host is configured to act as the network gateway. An Openswan VPN gateway provides the IPsec implementation. The local PureApplication System administrator can configure the gateway quickly, enabling the project to proceed without delay. In turn, the gateway administrator works with the SoftLayer DevOps team to validate the functionality of the IPsec tunnel.

4.1.7 Hybrid cloud monitoring of TradeLite

This section describes the monitoring that is provided to the running TradeLite application. It is straightforward to extend the modular design of monitoring in PureApplication Service on SoftLayer and PureApplication System to provide a hybrid cloud monitoring solution. See *Integrating IBM PureApplication System into an Existing Data Center*, SG24-8285, for chapters that describe the monitoring and integration of PureApplication System in an existing data center. We used this book extensively to set up our monitoring environment. This book is available at the following website:

<http://www.redbooks.ibm.com/abstracts/sg248285.html>

Our hybrid cloud monitoring solution starts with validating that PureApplication Service on SoftLayer and PureApplication System are at the same level of the system monitoring pattern type. In our case, both PureApplication System and PureApplication Service monitoring are

at version 1.0.5.0. In the PureApplication Administration console, we selected **Cloud** and selected **Pattern Types** to obtain this information.

Hybrid cloud monitoring is asymmetric. That is, one side of the shared service pattern deployment solution is in *internal mode*, and the other is in *external mode*.

Internal mode consists of three services:

- ▶ Remote monitoring server
- ▶ Hub monitoring server
- ▶ Performance Data warehouse

When a pattern is deployed, it contains monitoring agents that are aware of and connect to the internal mode remote monitoring server. For more information about PureApplication monitoring, see “Monitoring” on page 211.

External mode is a subset of internal mode. It has one or more remote monitoring servers that connect to an external hub monitoring server and data warehouse. External mode is started after the internal mode because it needs the network locations of the hub monitoring server and data warehouse.

Important: When you start the monitoring service in external mode, use the IP address and not the host name of the remote monitoring server.

Because of the common code base in the PureApplication offering, our decision of where to host the internal mode was made based on which of the two environments has more resources. For us, more computing resources were available on the PureApplication System.

With a foundational understanding of our hybrid cloud monitoring infrastructure, it is time to consider the flow of network traffic. We discovered that it was useful to sketch the network components. Every environment is different, both in terms of the locations and how the network technology changes over time. No single person is responsible for everything in your sketch. In the rare case of a PureApplication System or PureApplication Software environment in a sole proprietorship, you will communicate with the PureApplication Service on SoftLayer DevOps team.

A sketch of the network components for monitoring TradeLite is shown in Figure 4-17.

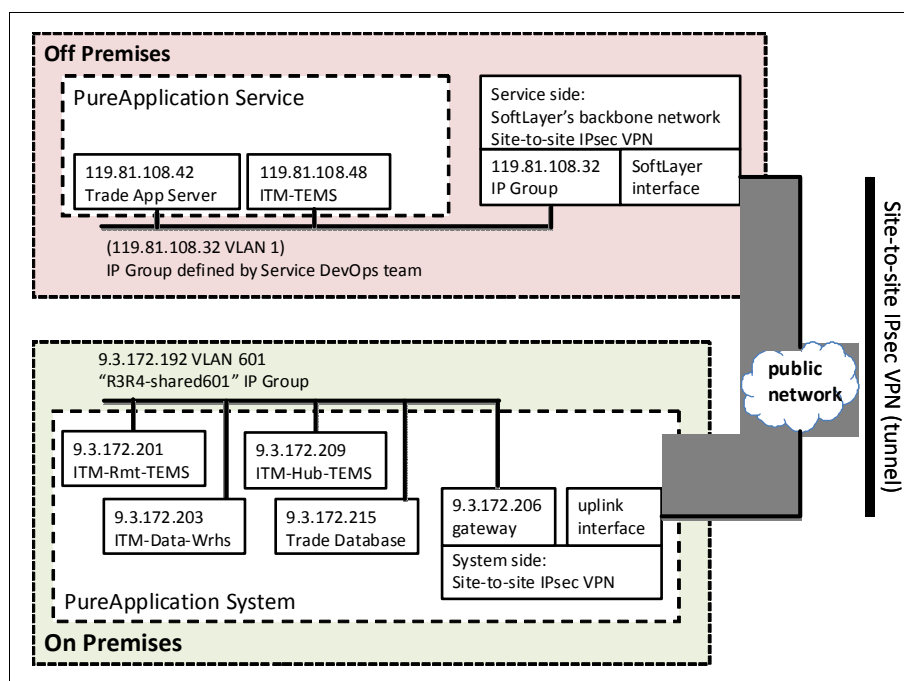


Figure 4-17 Sketch of the network components for monitoring TradeLite

Like we did, you might experience challenges in the network. In our case, the on-premises gateway needed to be updated with routing information, which required interaction with the personnel at the remote location. The sketch that is shown in Figure 4-17 helped us in this communication. We also prefaced our **ping** and **ssh** tests with the following identifiers:

```
"date ; hostname --fqdn ;"
```

That is, when we looked at our terminal session, we knew what happened at what time and on which host. When we used a Windows machine, it was beneficial to install the Cygwin environment. For more information, see the following website:

<http://www.cygwin.org>

If you deployed a PureApplication pattern before, the process will be familiar. If not, the IBM Knowledge Center is helpful. See *Deploying a System Monitoring shared service in internal mode*:

<https://ibm.biz/BdHhuJ>

After the deployment of on-premises and off-premises monitoring is complete, you can discover what can be seen. The PureApplication System monitoring portal provides both high-level and detailed information. See *Integrating PureApplication System into an Existing Data Center*, SG24-8285. This book includes a chapter about the use of system monitoring shared services. It describes how to best use the information that is provided by the portal. This book is available at the following website:

<http://www.redbooks.ibm.com/abstracts/sg248285.html>

Figure 4-18 is a side-by-side view of two workspaces from the monitoring shared service on the PureApplication System monitoring portal. This view shows workloads from both PureApplication System and PureApplication Service.

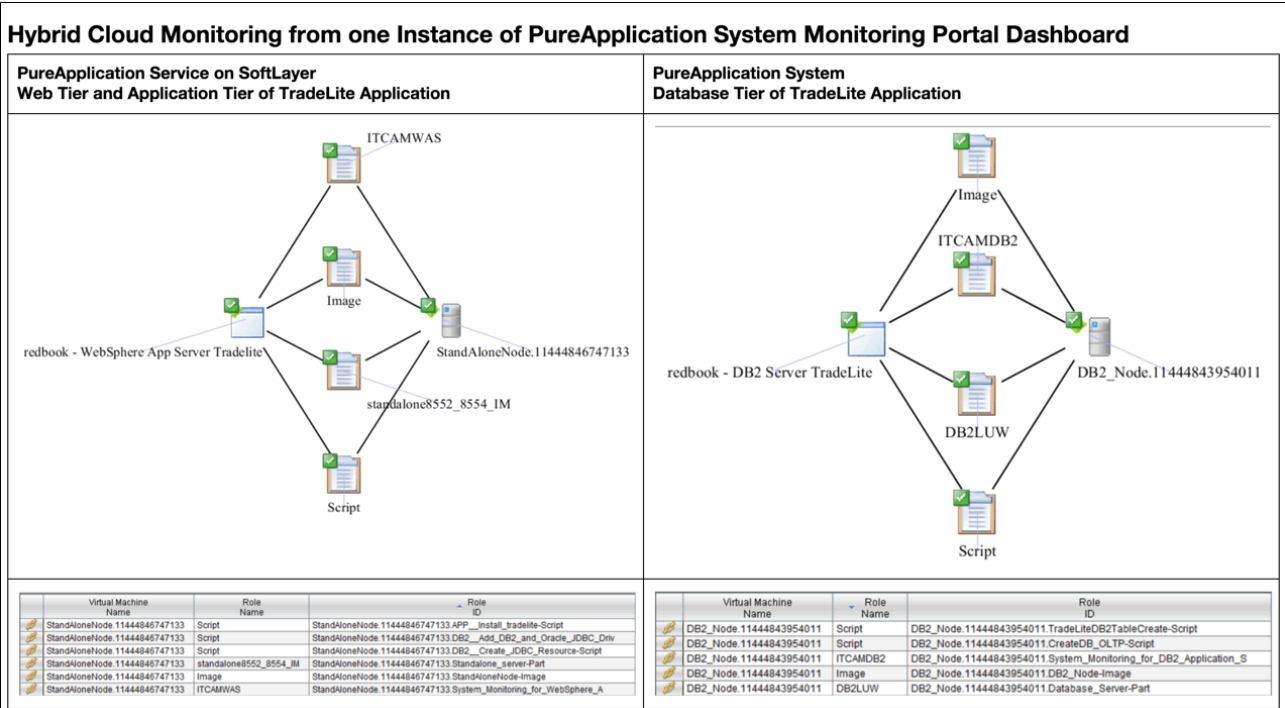


Figure 4-19 shows how a PureApplication Service instance is managed by a *PureSystems Manager* (PSM). The PSM is the user interface (UI) console that you work with to deploy and manage the lifecycle of deployed workloads.

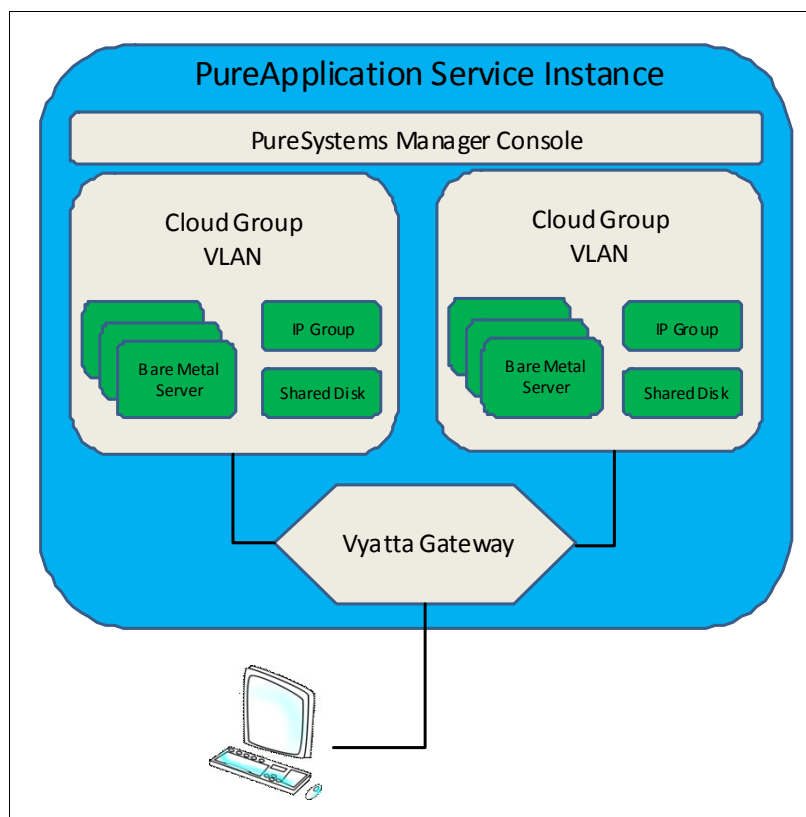


Figure 4-19 PureApplication Service components

Under the PSM, an instance also contains these components:

- ▶ Cloud Groups
- ▶ Bare metal servers, which are also known as compute nodes
- ▶ IP Groups
- ▶ Shared disk storage
- ▶ Network virtual local area networks (VLANs)
- ▶ A Vyatta gateway appliance

Within the PureApplication family, the highest level of isolation is a Cloud Group. A Cloud Group is a collection of one or more compute nodes and one or more IP groups, similar to a logical computer. A Cloud Group accomplishes two main goals:

- ▶ System segmentation: A Cloud Group divides a PureApplication System or PureApplication Service into one or more logical computers. Cloud Groups run isolated from each other.
- ▶ Compute node aggregation: A Cloud Group will group one or more compute nodes, with at least one IP group, into a logical computer that can have greater capacity than a single node.

Cloud Groups provide additional isolation for applications that run in PureApplication Service. A Cloud Group is a collection of physical servers or compute nodes. The physical separation of CPU and memory protects the running applications from being affected by applications that are running in other Cloud Groups. A single misbehaving application can consume a large amount of CPU, which affects all of the running applications in that Cloud Group. Moving high-value applications into your own Cloud Group can ensure that applications have enough CPU and memory resources.

Workloads in a Cloud Group can share the virtual network by using the same VLAN, or they can be isolated from other workloads by using different VLANs. The default configuration in PureApplication Service is for all workloads to share a VLAN.

A particular compute node can belong to only one Cloud Group. Typically, a Cloud Group contains at least two compute nodes so that workloads keep running in a node failure. Using two or more compute nodes eliminates a single point of failure (SPOF).

Cloud Groups can be customized by creating one or more IP pools for assigning IP addresses to VMs for deployment. An IP group is a logical grouping of one or more IP addresses with networking information (such as DNS and subnet). An IP group and, by default, its IP addresses, cannot be shared across Cloud Groups. VMs for deployments that are targeted to a Cloud Group are assigned an IP address from the IP pools that are assigned to the Cloud Group.

Cloud Groups on PureApplication Service are established by the DevOps team. The capacity of a Cloud Group can be expanded by adding additional servers to the Cloud Group. Each server in a Cloud Group must be the same size and capacity. If a Cloud Group consists of compute nodes with 16 cores and 256 GB of RAM, any additional nodes that are added to the Cloud Group must be sized identically.

PureApplication Service offers a range of sizes for compute nodes. Sizes range from 4 cores for eSeries Servers up to 40 cores for vSeries servers. The amounts of memory range in size from 32 GB of RAM for eSeries servers, and up to 1024 GB of RAM for vSeries servers.

Note: Check with your PureApplication sales specialist for a current list of available configurations.

Consider high availability when you determine the size of the servers. Two or more smaller servers can provide higher availability than a single server. In a node failure, PureApplication Service restarts all of the workloads from the failed node on the surviving nodes. If the capacity is insufficient, the highest-priority workloads are started first, continuing with the lower-priority workloads until resources are exhausted. If you do not require a reduction in application capacity during a node failure, ensure that you size the environment with enough additional capacity to handle the failure of a compute node.

For a more detailed discussion about Cloud Groups and runtime environments on PureApplication System, see *Managing application runtime environments in IBM PureApplication System*:

http://www.ibm.com/developerworks/websphere/library/techarticles/1210_woolf/1210_woolf.html

Figure 4-20 shows a PureApplication Service instance with Cloud Groups.

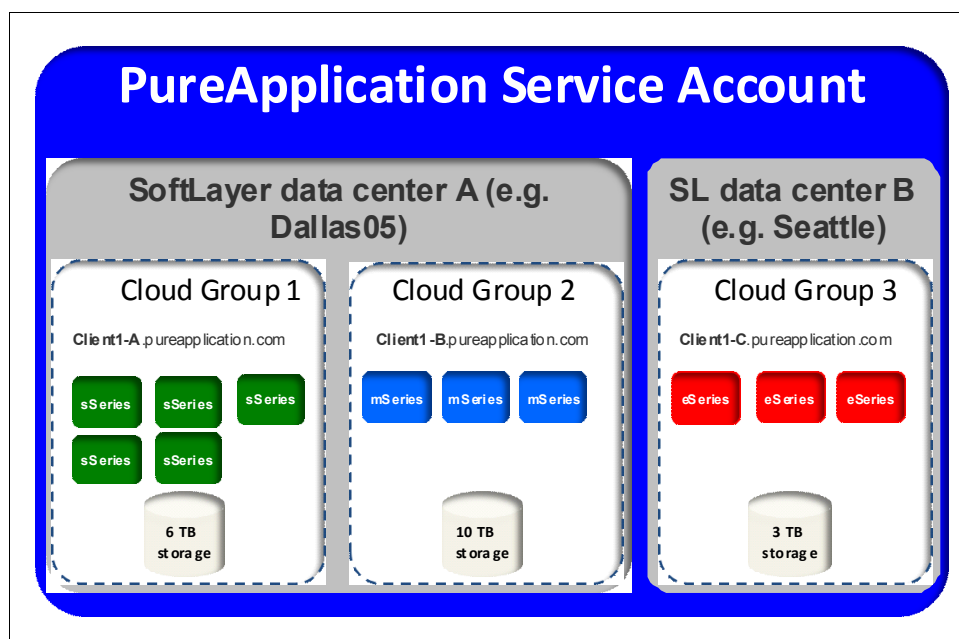


Figure 4-20 PureApplication Service instance with Cloud Groups

Cloud Groups can also implement isolation between different lifecycles within a software development lifecycle. Most clients do not mix workloads among development, test, and production. These workloads are typically separate from each other and might be required to be on separate hardware. By using Cloud Groups, you can implement one Cloud Group for development, one Cloud Group for quality assurance (QA), and one Cloud Group for production. Figure 4-20 depicts a configuration with test and production Cloud Groups in one data center, and disaster recovery (DR) in a separate data center.

Another reason for using Cloud Groups is to maximize software licenses. The text from Figure 4-21 is from the IBM sub-capacity software licensing site. It describes how software is licensed by using *sub-capacity licensing*.

Sub-capacity licensing lets you license a PVU-based software program for less than the full processor core capacity of the server, when the software program is deployed in an eligible virtualization environment.

IBM licenses on the basis of whole processor cores, based on the lower of the physical or the virtual cores. The minimum requirement for software licensing is the appropriate number of PVU license entitlements for one processor core. This licensing is based on the processing capacity (expressed in PVUs) available to the IBM middleware. IBM licenses to the lower of the sum of virtual capacity or the full (physical) capacity of the server. Reference the Virtualization Licensing Counting Rules for details and examples on calculating sub-capacity license requirements in various scenarios.

Figure 4-21 Sub-capacity licensing

In simple terms, the text in Figure 4-21 on page 162 means that you need to license each VM that uses vCPUs as though it were a physical core, until you reach the capacity of the server. For a Cloud Group with two 16-core compute nodes, we license an IBM product for up to 32

vCPUs. At 32 cores, the entire Cloud Group is entitled, and no additional entitlements are required. Licensing an IBM product for all 32 cores results in many times more software licenses than required by the application. Creating a small Cloud Group of two 4-core servers enables you to license eight cores and entitle the entire zone.

For example, if a client deploys IBM Digital Experience (IDX), which was known as WebSphere Portal, the client can create a Cloud Group with two 4-core compute nodes that total eight cores. The client can then license IDX for all eight cores. When the entire Cloud Group is entitled, the client can deploy as many IDX VMs as the Cloud Group will support. By using a 3:1 ratio of vCPU to physical, a client can potentially deploy 24 vCPUs and still have acceptable performance. The performance differs in each environment.

4.2.2 Networking and Cloud Groups

Up to this point, we focused on providing isolation at the CPU and memory level. We take for granted that the virtualization layer will provide isolation between the deployed VMs. The next layer of isolation to describe is the network layer. The *network layer* is the most challenging layer and provides the greatest risk for intrusion from unauthorized users.

A Cloud Group is a defined mechanism so that you can add a deployment area in a PureApplication Service instance. Each Cloud Group runs with your own VLAN, which provides the separation of network traffic from other Cloud Groups. The Vyatta firewall can be configured to further protect critical applications.

A typical example of using multiple Cloud Groups is a 3-tier application. All web servers are in one Cloud Group. Java Platform, Enterprise Edition (Java EE) application servers are in the second Cloud Group. Database servers are in the third Cloud Group. Firewalls can be used to provide access to only those users or applications that require access.

The Vyatta gateway provides the IPsec tunnel termination point. This tunnel creates the secure connection between your data center and SoftLayer. The Vyatta appliance also contains firewall and network address translation (NAT) policies that further protect the SoftLayer environment from the outside world. You can configure the firewall and NAT policies by using the PureApplication Service UI or a Representational State Transfer (REST) application programming interface (API). The UI provides a self-service interface to configure and maintain the network configuration.

NAT policies are most commonly used when subnet addresses do not match between your internal network and an external network. The configuration of the network falls into one of three use cases:

- ▶ Local subnets and remote subnets for the IPsec VPN do not conflict.
- ▶ Local subnets and remote subnets for the IPsec VPN connection conflict.
- ▶ Local subnets for the IPsec VPN connection conflict with subnets that are used at the remote site.

In each of the use cases, you can configure the firewall and NAT by using firewall and NAT policies.

Figure 4-22 shows the required parameters to configure a NAT policy. Before you configure NAT and firewall information, check with the SoftLayer DevOps team and your local network team to verify the configuration settings.

Firewall Policies

Create New

Rule Number	Action	Sc
700	accept	10
701	accept	10
900	accept	9.3
884	accept	8.8

Total: 7 Selected: 0

NAT Policies

Create New

Rule Number	Exclude	Disable
2	true	false
3	true	false
4	true	false
5	true	false

Add NAT policy

* NAT Type:

destination

* Rule Number (1 - 9999):

* Exclude:

false

* Disable:

false

Source Address:

any

Source Port:

any

Destination Address:

any

Destination Port:

any

Translation Address:

any

Translation Port:

any

* Protocol:

all

OK

10.66.245.96/27	any	158.85.80.48/28	any	any
-----------------	-----	-----------------	-----	-----

Figure 4-22 Configuring NAT policies

4.2.3 Load balancing

Load balancing is a way to distribute processing and communications evenly across multiple servers in a data center so that a single device does not carry an entire load. Load balancing is essential in situations in which it is difficult to predict the number of requests that might be issued to a server. Load balancing can distribute requests that are made to a single server to ease the load, minimize latency, and address other issues. With SoftLayer load balancing, you can employ various balancing methods to distribute traffic, including persistent and non-persistent options that can be changed, activated, and deactivated at any time.

Local load balancing

SoftLayer load balancing uses industry-standard techniques for balancing traffic among two or more servers, including round-robin, lowest latency, least connections, shortest response, and IP persistence. Load balancing can be activated and configured in real time, with servers added to or removed from the balancing pool on demand, with little or no downtime. As the client, you size the configuration, based on the anticipated number of concurrent connections. Local load balancing can route to public IP addresses in a single data center only.

Global load balancing appliance

The Citrix NetScaler VPX distributes traffic between your servers in one or multiple SoftLayer data centers. These multifunction network appliances can load balance DNS-based local and global loads so that you completely control how your client traffic is balanced among your servers. The VPX appliance is a multitenant appliance that provides economical price performance.

High availability dedicated load balancers

You can obtain the same capability as the global load balancing appliance with the added performance of a dedicated load balancer. Citrix MPX load balancers are available for environments with higher capacity. These load balancers are also available with a high availability option that features failover protection and automatic fallback.

4.2.4 Denial-of-service (DoS) protection

A denial-of-service (DoS) attack is an attempt to make a machine or network resource unavailable to its intended users by flooding the user with network traffic to overwhelm the target. Protection against a DoS can be provided in a PureApplication Service environment. If you want a (virtual) protective gateway between your WebApp VM and internet users, you can request this service by using the self-service portal. This protective gateway regulates network traffic and limits requests, as a countermeasure against DoS attacks.

When you want to request a protective gateway, open a service ticket, and provide the following information:

- ▶ System description:
 - PureApplication Service account number
 - Name of company that is licensed to use PureApplication Service
- ▶ Requester's name, phone, and email address.
- ▶ IP address/URL that is used to access the PureApplication Service console.

- ▶ Enter the endpoint (URL, including IP, port number, and context root) of the VM. Note: You can create as many VM endpoint sections as required:
 - Payload size protection (yes/no): Enter yes or no if you want this feature for the system. If yes, complete the following information:
 - Payload threshold: What threshold do you want to enforce for the maximum HTTP request payload size? Specify a positive integer number of bytes. Note: This threshold is set one time for the DoS protection feature.
 - Message count protection (yes/no): Enter yes or no if you want this feature. If yes, complete the following information. Note: You can create as many message count protection sections as required:
 - Message count protection rule: Enter either Each requesting IP separately or All requests regardless of the requesting IP.
 - HTTP methods for protection rule. Choose where the message count protection rule is to be placed on all or specific methods. Enter all of the HTTP method types, or list the specific HTTP method types (for example, PUTs, GETs, and POSTs).
 - What is the maximum number of requests? Determine the maximum number of requests the protection rule will allow through the gateway to your VM for each unit. Supply a rate as an integer number of requests, for an integer number of milliseconds. Use this format:
 Allow _____ requests every _____ milliseconds

The DevOps team will deploy and configure the DoS (virtual) protective gateway when they receive this information. When new endpoints are deployed or existing endpoints are deleted, you can open a service ticket to update the protective gateway with the new configuration.

4.2.5 Securing data at rest

Another aspect of security is securing data at rest. The PureApplication family supports encryption of data at rest by using SPxBitFiler-IPA from Security First Corp. The encryption technology that is available from Security First Corp. enables virtual system patterns to easily incorporate the encryption of on-disk data. The encryption solution is based on the Federal Information Processing Standard (FIPS) 140-2 certified cryptographic module, which has the following benefits:

- ▶ Encryption for virtual application patterns and virtual system patterns is easily installed and configured with only a few mouse clicks.
- ▶ Define the directory paths to be encrypted. Identify directory paths either for removal of encryption or to prevent encryption by using simple configuration parameters that you specify. Change encryption parameters at any time by running the customized script packages on demand and specifying new encryption parameters, as needed.
- ▶ Configure and apply encryption policies to your virtual application patterns, either applied globally to all components in the pattern, or targeted to individual pattern components.
- ▶ Installation, configuration, and encryption do not require system reboot or software application interruption.
- ▶ Granular integration control protects current and future data in targeted directories.

- ▶ IBM Encryption Pattern for Security First SPxBitFiler-IPA uses the power of Security First Corp. Cryptographic Splitting, including Advanced Encryption Standard 256 (AES 256) in a FIPS 140-2 certified cryptographic module that meets Health Insurance Portability and Accountability Act (HIPAA), Health Information Technology for Economic and Clinical Health (HITECH) and Federal Information Security Management (FISMA) compliance requirements for data at rest.
- ▶ All cryptographic functions use National Institute of Standards and Technology (NIST) Special Publication (SP) 800-131A-allowed algorithms and key strength for all cryptographic functions.
- ▶ No additional external server or services are required.
- ▶ Key management is automatic and transparent, requiring no user intervention.
- ▶ Additional directories and files can be encrypted at any time, even during normal operation of the system.
- ▶ Exceptions can be defined in encrypted directories for data that needs to remain cleartext.

4.3 Portability

The topic of installation and configuration of both operating systems and middleware in support of business applications inspires many approaches. A challenge each approach must face is how to make the environment, and with it, the application, portable across multiple platforms. In this section, we take a close look at two contemporary techniques for providing a portable infrastructure to run an application:

- ▶ Patterns
- ▶ Docker

4.3.1 Patterns

IBM announced and delivered pattern technology in 2009. Pattern technology grew out of requests from application owners for an innovation to provide a repeatable, reliable means to stand-up application runtimes. After the initial experience with the benefits of patterns, the hybrid cloud use case was requested.

For this project, we are provided with off-premises resources (PureApplication Service on SoftLayer) and on-premises resources (PureApplication System). With our secure network connection to this heterogeneous computing infrastructure in place, let's consider an example of an application that relies on resources that are deployed in PureApplication Service on SoftLayer and deployed in PureApplication System.

TradeLite is a 3-tier web application that is provided as a sample in WebSphere Application Server. In this example, the Web server tier is an internal process, a transport chain, of the WebSphere Application Server. The sample is included as part of the entitled content in both PureApplication Service on SoftLayer and PureApplication System.

The first pattern deployment is on PureApplication Service on SoftLayer for the database, DB2, which is depicted in Figure 4-23.

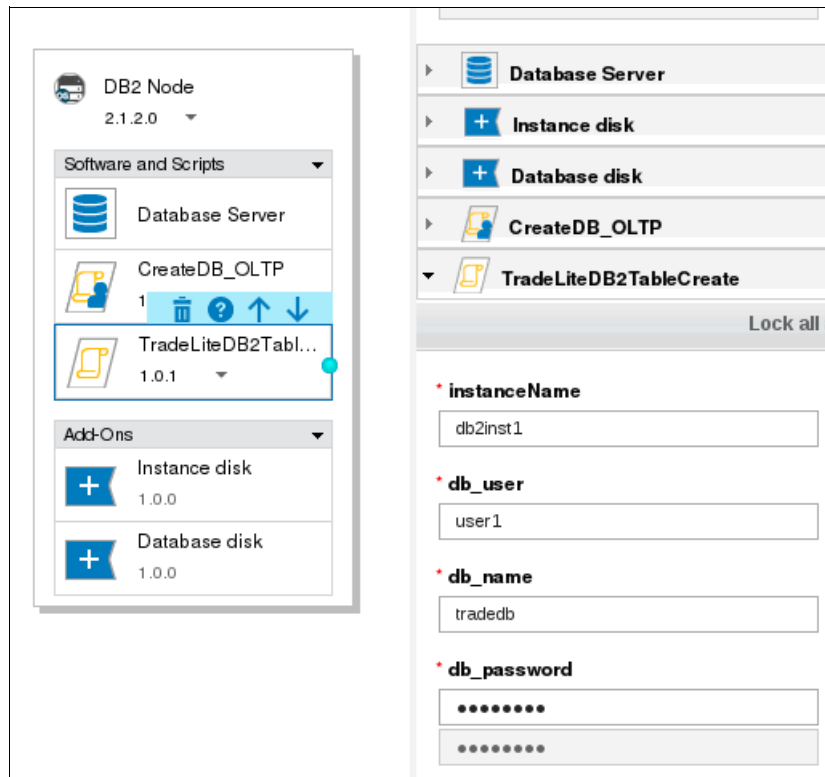


Figure 4-23 Pattern for the TradeLite database tier for PureApplication Service on SoftLayer

With the network address from this first deployment instance, the application server is deployed next on PureApplication System, as shown in Figure 4-24.

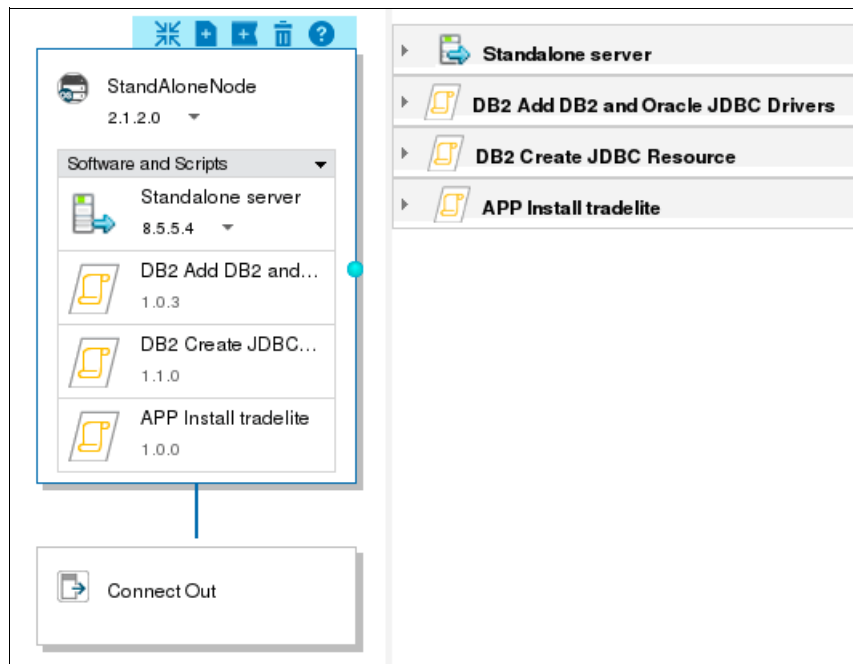


Figure 4-24 Pattern for the TradeLite web and application tier for PureApplication System

The configuration automation of TradeLite existed before pattern technology. We are able to reuse this WebSphere wsadmin and the operating shell script work in a script package (in this case, APP Install tradelite in Figure 4-24 on page 168). A *script package* is a collection of one or more shell scripts, wsadmin commands, operating system (OS) executable commands, or binary files of third-party software, contained in a compressed file. The only required file, cbscript.json, contains metadata about optional parameters to pass to the script and, most importantly, how to start execution. A simple script package can then consist solely of the cbscript.json file, which invokes a copy of a shell script that you already use.

With script packages that are primed to perform the configuration work, we are asked to provide the wsadmin script and shell script variable values at deployment time:

- ▶ dbHostName: a.bc.215
- ▶ dbUserID: db2inst1
- ▶ dbPassword: *****
- ▶ DataSourceName: TradeDN
- ▶ DataSourceJNDI: jdbc/TradeDataSource
- ▶ dbDatabaseName: tradedb
- ▶ dbPort: 5000

Figure 4-25 shows deployment time configuration values for the TradeLite application server on PureApplication System to establish a connection to the TradeLite database server on PureApplication Service on SoftLayer.

The screenshot displays the 'Prepare to Deploy' tab in the IBM PureApplication System interface. On the left, a sidebar shows 'Pattern attributes' and a list of nodes under the heading 'Nodes'. The 'StandAloneNode' is selected and highlighted in blue. Below the nodes list is a 'Connect Out' button. The main area on the right is titled 'StandAloneNode' and contains the following configuration sections:

- StandAloneNode configuration:**
 - Fill in the required values for component **StandAloneNode** for this pattern.
 - Virtual CPUs:** 2 (dropdown menu)
 - Memory size (MB):** 4096 (text input)
- Standalone server:**
 - DB2 Add DB2 and Oracle JDBC Drivers (expandable section)
 - DB2 Create JDBC Resource (expandable section)
- DB2 Create JDBC Resource configuration:**
 - Fill in the required values for component **DB2 Create JDBC Resource** for this pattern.
 - DSNAME:** TradeDN (text input)
 - DB_HOST:** 9.3.172.215 (text input)
 - DB_NAME:** tradedb (text input)
 - DB_JNDI:** jdbc/TradeDataSource (text input)
 - DB_USERID:** db2inst1 (text input)
 - DB_PASSWORD:** (password field with masked characters)
 - DB_PORT:** 50000 (text input)
 - DB_TYPE:** DB2 (XA) (dropdown menu)
 - WEB_CLUSTER_NAME:** (text input)

Figure 4-25 Deployment time configuration values for the TradeLite application server

The Pattern Builder summarizes our declared infrastructure configuration in metadata that is stored in the catalog. We obtained the pattern and imported it into PureApplication Service on SoftLayer and then PureApplication System. Then, we removed the pattern components that were not needed and saved our location-appropriate pattern.

Within a few minutes, we accomplished a great deal. The Pattern Builder enabled us to modify the deployment for a monolithic infrastructure to a distributed cloud infrastructure. With script package reuse, we configured our hybrid cloud efficiently and consistently.

Each member of the PureSystems family contains a copy of the PureSystems Manager activation engine. The activation engine reads the metadata that was created by the Pattern Builder in each pattern and translates the information from a platform-neutral implementation to a platform-specific implementation. The translation considers the different characteristics of each environment. This method allows each pattern to be portable between different environments.

The TradeLite application is now deployed off premises and on premises, as shown in Figure 4-26. We validate that all is as expected by adding a Trading Account for the IBM Redbooks team, for testing.

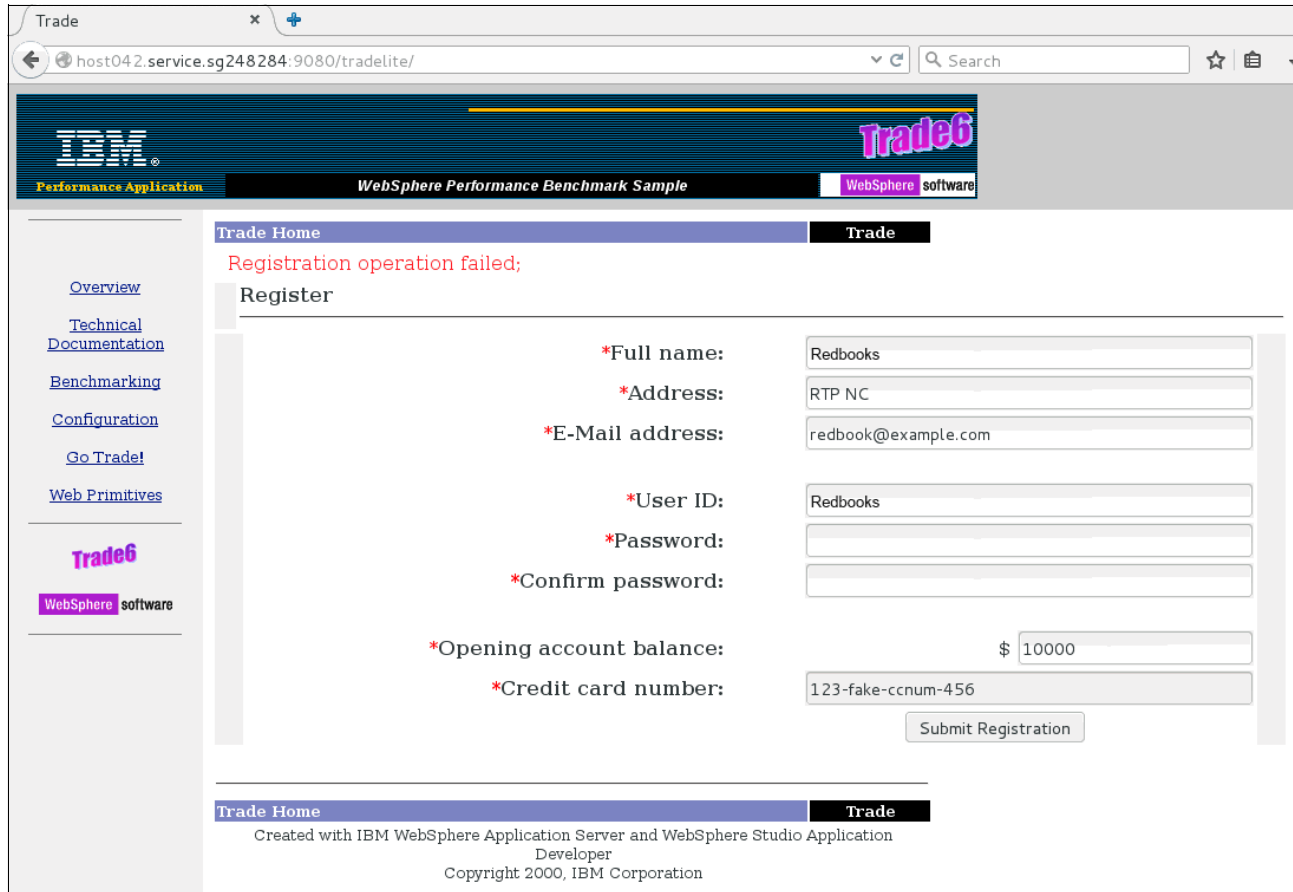


Figure 4-26 TradeLite is up and running across the hybrid cloud

Validation that the application behaves as anticipated is obtained with IBM Data Studio, where we can view the database row that lists the new account.

Figure 4-27 shows from the web and the application tier on PureApplication Service on SoftLayer that our new account is preserved as a database record in PureApplication System.

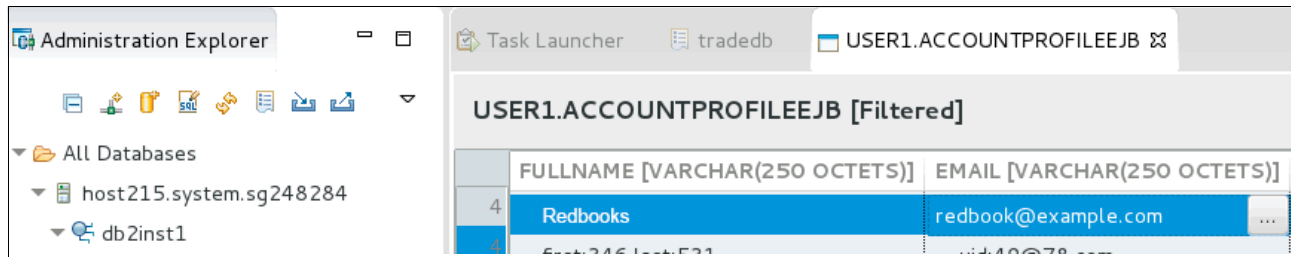


Figure 4-27 New account shows as a database record in PureApplication System

4.3.2 Docker

Docker is a rapidly evolving technology. PureApplication Software, PureApplication Service, and PureApplication System all support Docker in a common and consistent manner. Docker started as an improved means for working with the Linux kernel isolation features, namely, control groups and name spaces. It is only available today in the Intel versions of the PureApplication family.

For more information, see the following websites:

- *Working with Docker* (for PureApplication Service)
<https://ibm.biz/BdHhuu>
- *Working with Docker* (for PureApplication Software)
<https://ibm.biz/BdHhuu>
- *Working with Docker* (for PureApplication System)
<https://ibm.biz/BdHhuC>

Our hybrid cloud provides common components both on premises and off premises so that we can choose the distribution of components that best aligns to our business objectives. In this illustration, we locate components based on the availability of capacity. We extended the example *Creating and Deploying Patterns with Docker Containers in IBM PureApplication* at the following website:

<https://www.youtube.com/watch?v=qfUPpxCgK18>

We expand on this video and move to a hybrid cloud. The following diagram in Figure 4-28 provides a high-level view of the components and their relationships for our Docker hybrid cloud.

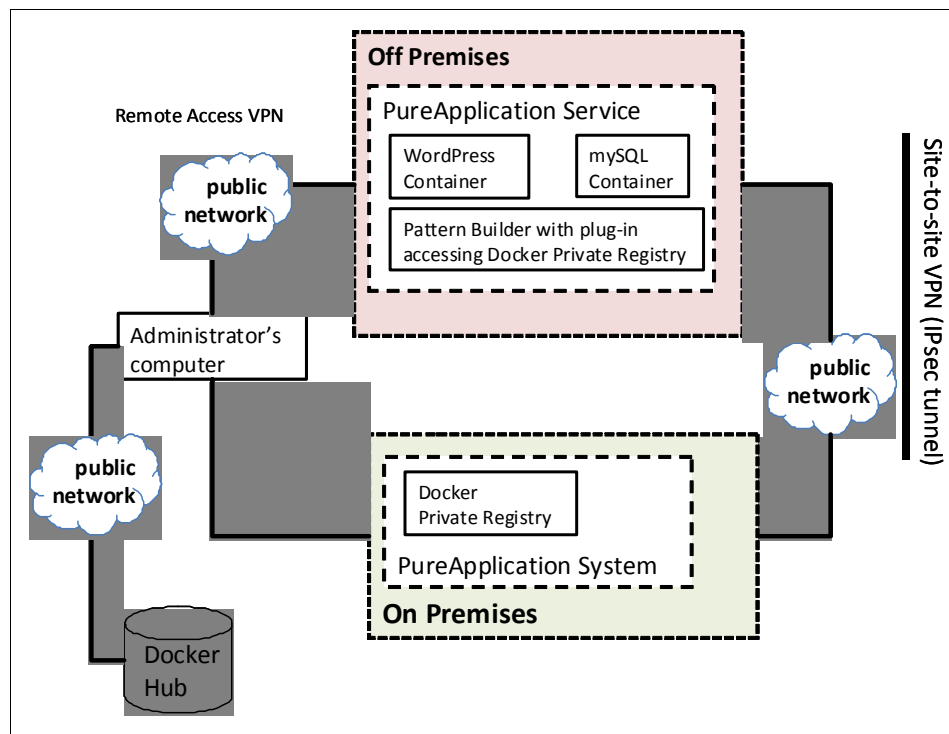


Figure 4-28 Docker in a hybrid cloud

Our first task is to stand up an instance of the PureApplication Docker Private Registry Pattern, which is described at the following website:

<https://ibm.biz/BdHhLD>

We accept the default values for the deployment of this pattern. An Internet Protocol version 4 (IPv4) address is provided from the PureApplication System IP Group that also contains our gateway to PureApplication Service on SoftLayer. We also use the default value for the IP port: 5000.

With the Docker Private Registry running, we configured the PureApplication Service on SoftLayer so that its Pattern Builder is able to reach the content that is hosted on premises.

From the PureApplication Service on SoftLayer Console start page, we perform the following tasks:

1. Select **Catalog** → **Tools** → **Docker Registry**.
2. Enter the address of the on-premises Docker Private Registry.
3. In the text box, enter the Registry URL along with port 5000.
4. Select **Test Connection**, as shown in Figure 4-29 on page 174.

Docker Registry

Connected to Docker Registry

Registry URL

9.3.172.204:5000

Internal registry URL

Registry protocol

http

Registry CA certificate

Browse

Registry user name

Registry password

Password

Verify password

Docker bridge address

10.20.42.1/24

Test Connection

Save

Figure 4-29 Configure the IP address of the Docker Private Registry

With the completion of this configuration task, the Docker Private Registry on PureApplication System is known to PureApplication Service on SoftLayer. The Pattern Builder on PureApplication Service on SoftLayer will access the Docker Private Registry when it is asked for the collection of available Docker images. The network communication between these build pieces of the hybrid cloud can be validated at any time by returning to this page and selecting **Test Connection**.

The administrator can obtain Docker images from the Docker Hub by using the following Docker commands. We showed only a few of the status messages from Docker Hub in Example 4-7 to save space.

Example 4-7 Commands to obtain Docker images

```

$sudo docker pull wordpress
Using default tag: latest
latest: Pulling from library/wordpress
$sudo docker pull mysql
Using default tag: latest
latest: Pulling from library/mysql

```

The curated Docker Hub content on the administrator's notebooks is now pushed to the on-premises PureApplication System Docker Private Registry, as shown in Example 4-8 on page 175.

Example 4-8 Commands to push content to PureApplication System Docker Private Registry

```
$sudo docker daemon --insecure-registry 9.3.172.204:5000
$sudo docker tag wordpress:latest 9.3.172.204:5000/sg248284/wordpress
$sudo docker push 9.3.172.204:5000/sg248284/wordpress

$sudo docker tag mysql:latest 9.3.172.204:5000/sg248284/mysql
$sudo docker push 9.3.172.204:5000/sg248284/mysql
```

Use the **docker search** command to validate your results, as shown in Example 4-9.

Example 4-9 The docker search command

```
$sudo docker search 9.3.172.204:5000/sg248284
NAME                DESCRIPTION    STARS     OFFICIAL   AUTOMATED
sg248284/wordpress                0
sg248284/mysql                  0
```

The Pattern Builder on the off-premises PureApplication Service on SoftLayer is now ready for use.

You can work with the Pattern Builder in the standard way. The Docker software component is dragged to the composition canvas. Meaningful names are keyed into the appropriate fields. The input for the Docker image is fetched by the Pattern Builder from our on-premises PureApplication System, which is unique in the hybrid cloud configuration. The Pattern Builder user receives the benefit of a common registry and the work of the administrator to set up this configuration, as shown in Figure 4-30.

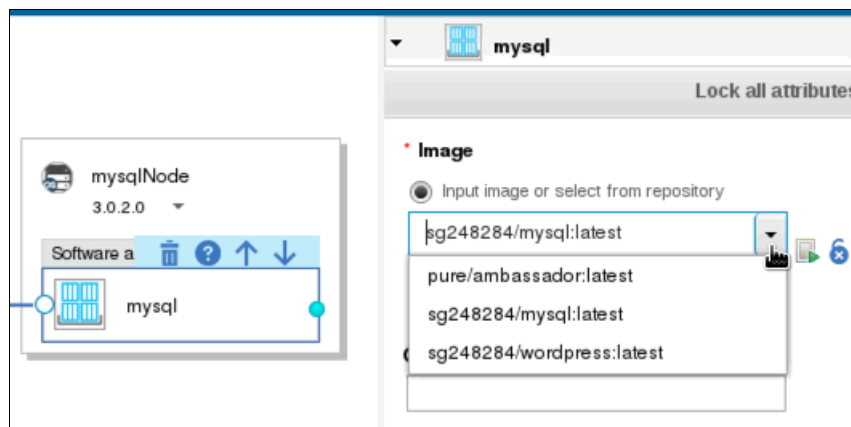


Figure 4-30 Docker-configured Pattern Builder

We finish the construction of the pattern, and then deploy it to the off-premises PureApplication Service on SoftLayer environment, as shown in Figure 4-31 on page 176. Docker images are now part of the solution set for deploying business value applications with PureApplication Service on SoftLayer. In our hybrid cloud, the option is available to reverse roles and place the Docker Private Registry on PureApplication Service on SoftLayer and construct and deploy with PureApplication System.

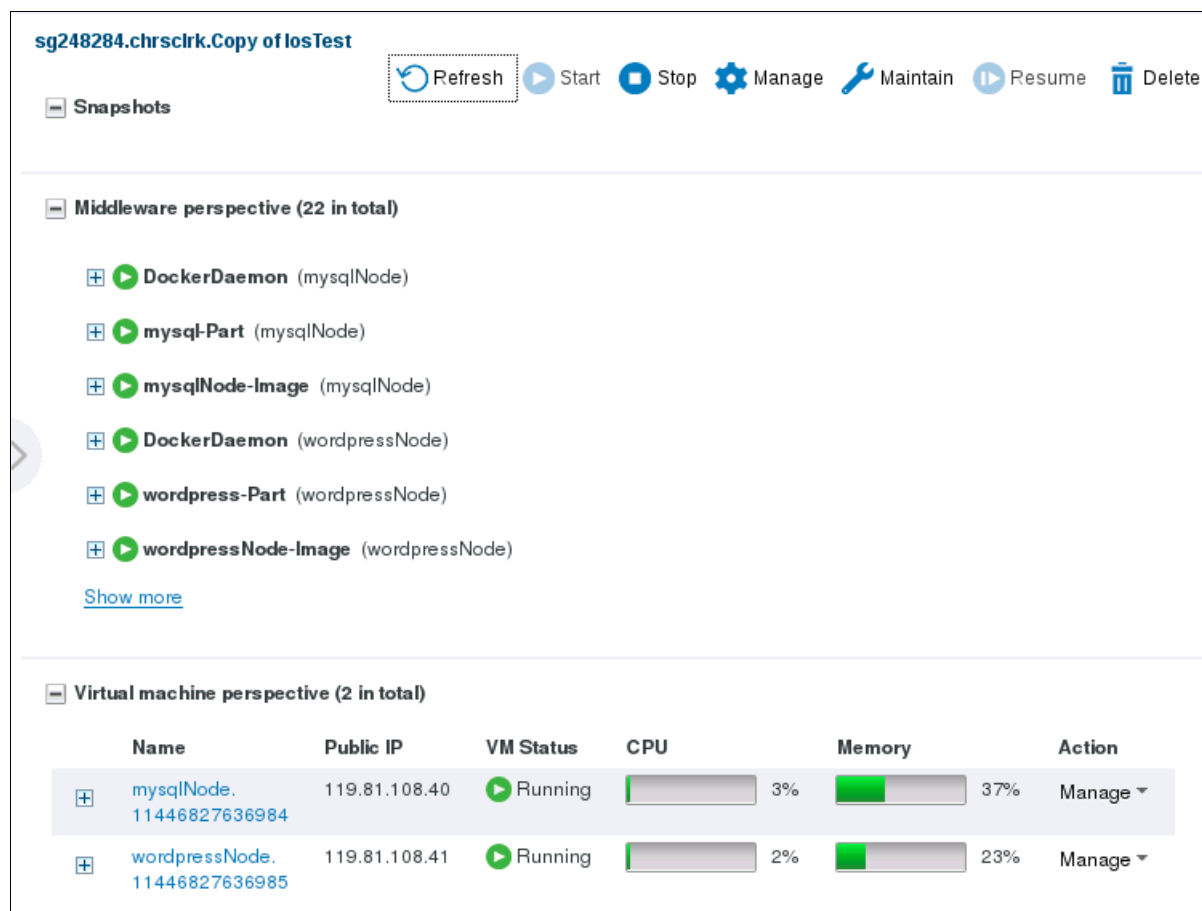


Figure 4-31 Docker deployment

4.4 Recoverability

The traditional backup and recovery process is about creating an exact copy of the runtime environment, with the ability to restore that copy in the future. Its focus is on copying images versus understanding how the environment was built and being able to re-create the environment. The PureApplication family focuses on archiving the components that were used to create the environment and the knowledge to re-create the environment when required.

Let's first look at how we can back up and restore a PureApplication Service configuration, its data, and patterns in the system. Because this data is slow-changing, we can use a point-in-time backup scenario. We will then look at how we can back up user data that is consumed and generated by the application.

4.4.1 Backup and recovery in a hybrid cloud

In Chapter 2, "How to build a hybrid cloud" on page 25, we reviewed the capabilities of backup and restore for PureApplication System and PureApplication Software. This chapter focuses on the required steps to back up and restore PureApplication Service. For a more in-depth discussion of backup and recovery, see 2.7.4, "PureApplication System and PureApplication Software backup and restore capabilities" on page 92.

System backup

PureApplication Service provides a built-in capability to back up and restore system and management data. The backup process stores three types of data:

- ▶ User IDs and passwords

Users that are defined in PureApplication Service are stored in this backup. The backup location is encrypted by default to safeguard the login information. The user ID archive must be restored by the DevOps team.

- ▶ System management data

This data includes information to set up the runtime environment for deployed VMs. The runtime information includes networking information, IP groups, and hypervisor configurations. Information about deployed workloads is also stored in this archive. *The system management archive can be restored only by the DevOps team.*

- ▶ Workload catalog data

All of the artifacts that will eventually be deployed in the runtime environment are stored in the workload catalog. The catalog includes base images, including virtual appliances, hypervisor images, base OS images, and user images that were created through the “extend and capture” process. In addition to the base images, pattern types (*ptypes*), virtual system patterns, and script packages are stored in the workload catalog. This list is not an exhaustive list of objects in the workload catalog, but it provides you with an idea of the types of artifacts that are stored in the catalog.

For more information about the categories of data to back up, see *Categories of data to be backed up*:

<https://ibm.biz/BdHhLF>

When PureApplication Service is first configured, the DevOps team creates three backup profiles, one for each type of data: user IDs, system management, and workload catalog. Each profile is scheduled to run daily, starting at 2:00 AM. When a backup profile starts, you can view the status of the backup request by selecting the backup process from the job queue. A status icon shows the status.

Figure 4-32 shows an example of a profile status.

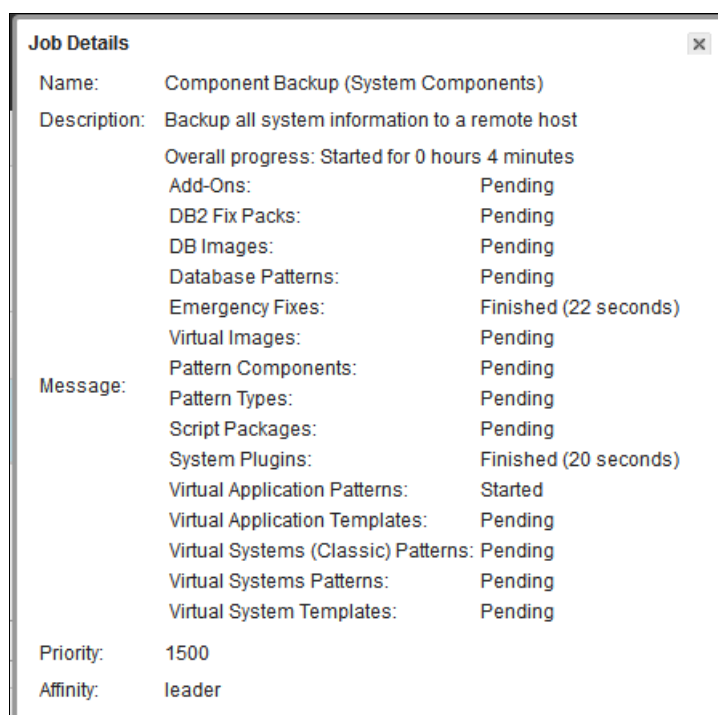


Figure 4-32 Backup profile status

The DevOps team will configure a target location to store backup data as part of the initial setup of PureApplication Service. When the first backup is run, a full backup image is taken. Subsequent backups are delta backups. The system always checks the target location for a full backup to determine whether to take a full or delta backup. Consider changing the remote location, perhaps to another directory location on the same remote backup system. This change causes a new baseline backup to be generated, which can occupy less space than many delta backups that are taken over time.

You can specify a new target location at any time. A secondary backup location might be useful for storing a copy of patterns and script packages at your site to provide an additional level of backup protection. Careful consideration needs to be taken when you create additional target locations in your data center. Backup and recovery time can be affected by network bandwidth availability from SoftLayer to your data center. A backup profile can move a potentially large amount of data across the network, directly affecting running applications.

Follow these steps to create a new backup target location:

1. Navigate to **Systems** → **Backup and Restore**.
2. Under “Locations have been set up and verified”, click **Change**.
3. The panel to create a new location appears. Click **Create New**.

4. Enter the following values:
 - Name of the backup location.
 - Host name of the backup server. Secure Shell (SSH) must be enabled for this server.
 - Directory path for the backup files
 - Port.
 - User ID to access the backup server.
 - The authentication type. Optional: You can use either a password or a private key. If you select a private key, you are prompted to either generate a new key or import an existing key.
 - Select whether you want to enable encryption. You must enable encryption for backups that include user or group data.

Figure 4-33 shows a new backup target location as defined by a client. Only the DevOps team has the private key for the original target. Data can be lost if the original target location is deleted.

Set backup location

Use this table to add and delete backup locations and modify settings as needed. The location that you define must be on a UNIX or Linux server.

Create New


Name	Host name	Path	Port	User name	Authentication type	Encryption
SysVol-83794074	10.0.0.6	/backup	22	devops	Password	<input checked="" type="checkbox"/>
1445587163925			22		<div> <input type="radio"/> Private key <input checked="" type="radio"/> Password </div> <div> <input type="password"/> </div> <div> <input type="password"/> </div> <div>  Using a private key is the preferred method for authentication. </div>	<input type="checkbox"/>

Figure 4-33 Backup target location

You can create your own custom backup profiles. Each profile can be tailored to back up the required workload data only.

Follow these steps to create a backup configuration:

1. Navigate to **Systems** → **Backup and Restore**, and click **Create New**.
2. Enter the following values:
 - A name for the backup profile.
 - Select a target location from the drop-down list.
 - Select the type of backup: system or component.
 - Select when to run the backup. You can run the backup either on demand or by a schedule that you define.
 - Select any users to notify when the backup completes.

Figure 4-34 shows the configuration panel to select workload components for a backup profile.

Backup Profiles - Create new configuration

Backup enabled

Name:

Location:

Type:

System backup

Component backup

Workload

Cloud

Security

Add-ons

Database patterns

DB2 fix packs

Database images

Emergency fixes

Pattern components

Pattern types

Script packages

System plug-ins

Virtual application patterns

Virtual application templates

Virtual images

Virtual system patterns

Virtual system patterns (Classic)

Virtual system templates

Backup Schedule:

On demand

Scheduled

Start backup date

Start backup time

Scheduled backup to repeat

Daily

*

End backup date

Available:

Denton Fendor

Gail Storm

Wase First

Selected:

Notify:

Figure 4-34 Backup profile configuration tab

You can use the PureApplication Service UI to restore workload objects from archives that are contained at a target location. For example, the restore process can be used to restore script packages or OS images that were deleted from the catalog in error. System management data and user ID data must be restored by the DevOps team. To request a restore of user ID or system management data, open a service request with IBM Support. After the ticket is received, the DevOps team will contact you to review the request and schedule the restore.

Follow these steps to restore component backups:

1. Navigate to **System** → **Backup and Restore**.
2. Select the required location and click **Restore Data**.
3. From the Restore configuration window, select the components to restore and click **Restore**.

Figure 4-35 shows a list of locations that you can select when you restore component data.

Restore

Restore data

Set alternate location

Location	Size	Status
<div><div></div>SysVol-83794074</div>	70.644 GB	<div><div></div>Available</div>

Total: 1 Selected: 1

< 1 >

5 | 10 | 25 | All +

Figure 4-35 Restore locations

When you expand a component category, you can select individual objects. See Figure 4-36.

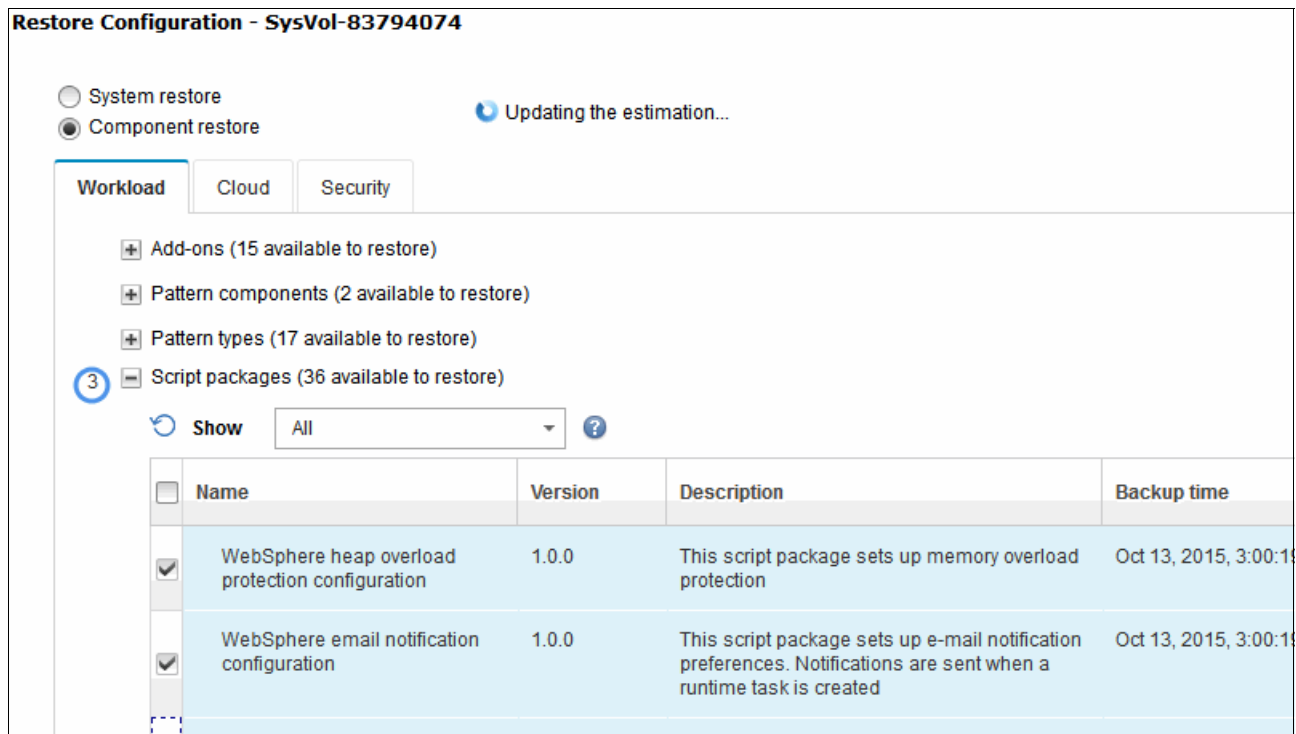


Figure 4-36 Restore component selection

Workload data backup

The Workload Backup option is an agent-based backup service so that you can back up and restore workload data from your PureApplication Service VMs. Use this service to manage and configure the backup and restoration of workload VM data, including specific VM directories and files. Each client has its own installed and configured Idera Backup Server, which consists of a dedicated SoftLayer virtual system. You choose the number of agent license packs that you require and the necessary storage size for your backup when you order this option. The Idera Backup Server (Figure 4-37 on page 182) can be used. You can include the Idera Backup and Restore script package in the pattern definition to automatically install the Idera backup agent on the VM at deployment. After the VM is deployed, the workload VM is automatically registered with the Idera Backup Server and the workload VM shows in the Idera Backup Server web console Servers list.

You can manage your workload VM agents, configure backup retention policies, encryption, compression, and archival settings through the Idera Backup Server web console. You can also restore your backups through the web console, including the ability to restore specific workload VM files and also restore data from one agent VM to another agent VM.

Figure 4-37 shows the Idera Backup Server.

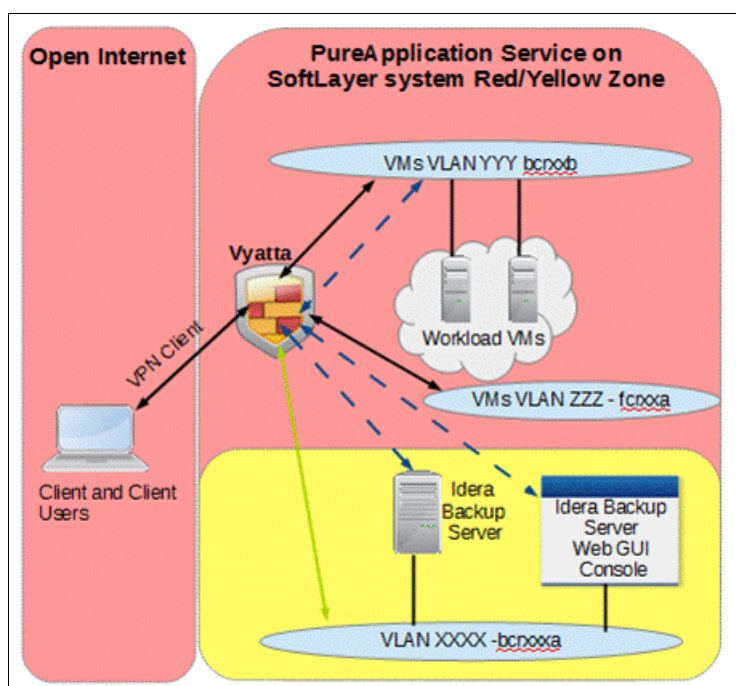


Figure 4-37 Idera Backup Server architecture

The Workload Backup Service consists of five components:

- ▶ Idera Backup Server: SoftLayer Virtual Server
- ▶ Access to the Idera web console
- ▶ Idera backup and restore script packages
- ▶ Idera agent licenses
- ▶ Storage: Additional configured backup storage

Each VM that requires backup services must have an agent license. The licenses can be purchased in packs of 1, 5, 10, or 25. The agent fee is charged as a monthly charge. Additional agent licenses can be purchased at any time. Backup storage is ordered in 0.1 TB increments.

The DevOps team configures the default backup volume with the amount of storage that is purchased by the client. The backup volume is responsible for defining the location of the disk space that stores the backup data from each server, and the backup volume defines the encryption and compressions settings. You configure and schedule your backup retention policy, archive schedule, recovery points, and restores by using the Idera web console.

Figure 4-38 shows the Idera web console.

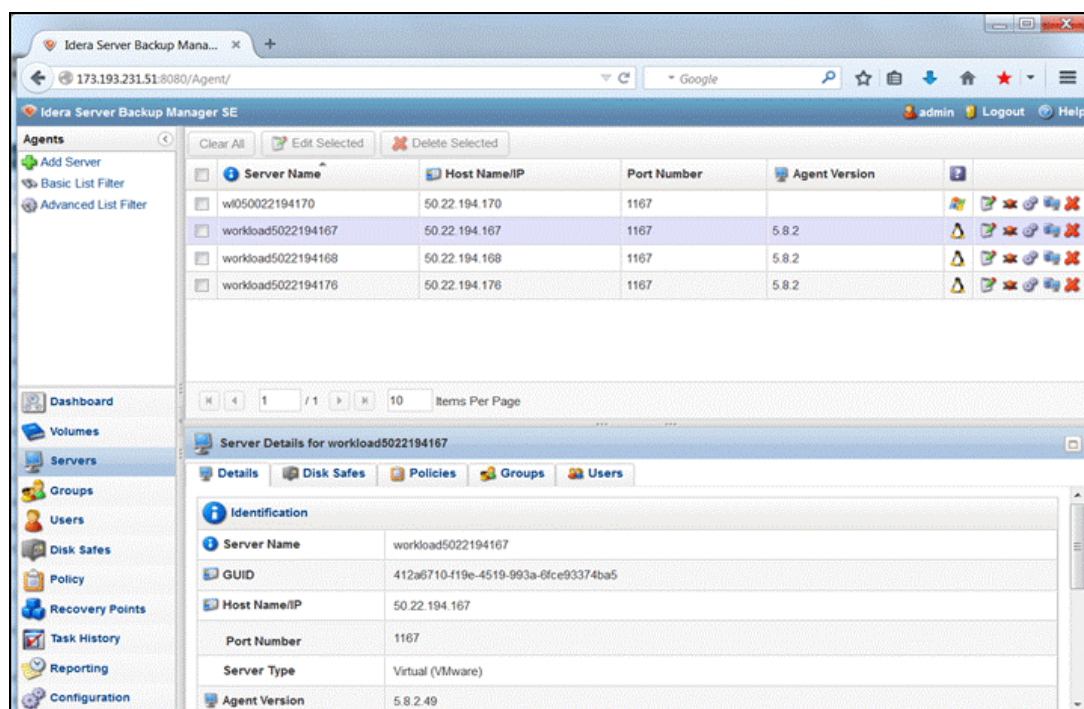


Figure 4-38 Idera web console

The Idera backup and restore script package installs the Idera agent and registers the agent to the Idera backup server at pattern deployment time. The script package supports VSys and VSys Classic Red Hat Enterprise Linux (RHEL) and Microsoft Windows patterns. The Idera backup and restore script package is uploaded to your PureApplication Service environment by the DevOps team. After the script package is loaded, you receive the required parameters to run the script package:

- ▶ Idera backup server IP
- ▶ Idera web console user name
- ▶ Idera web console password

The script package is added to the OS node where the backup agent will run, as shown in Figure 4-39.

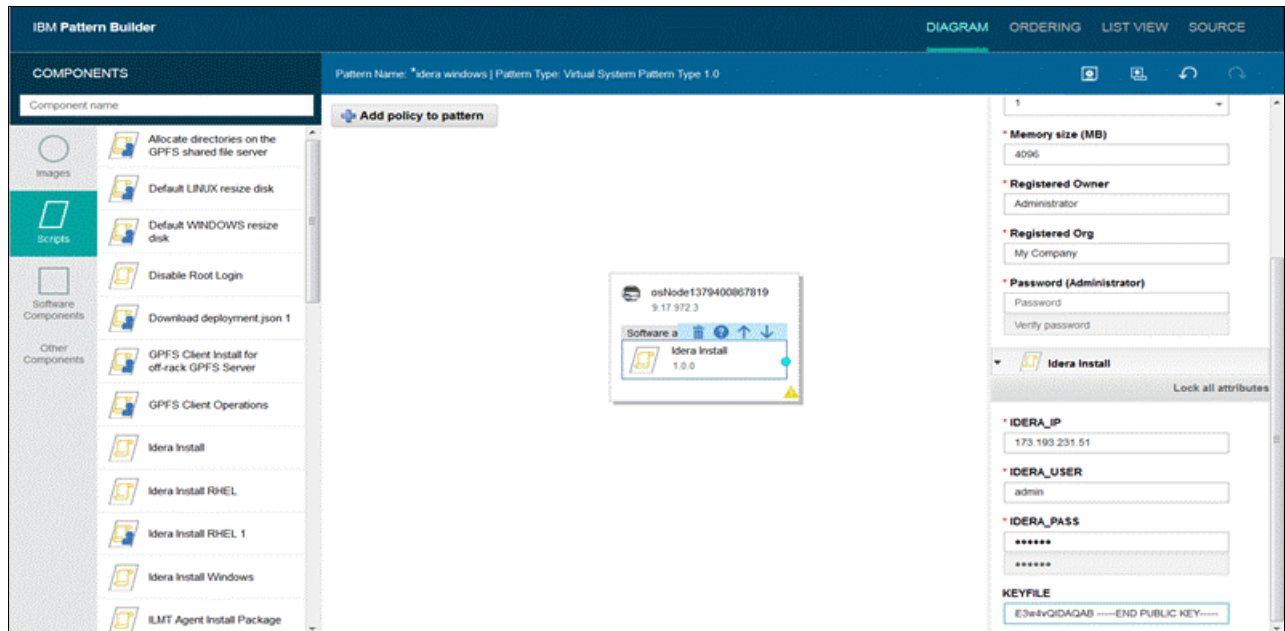


Figure 4-39 Pattern Builder with Idera script package

When you deploy a Windows pattern, you copy the public key value from the Idera web console GUI configuration page for the script package KEYFILE parameter. The public key is generated by SoftLayer when the Idera backup server is ordered. See Figure 4-40.



Figure 4-40 The KEYFILE parameter for Windows

When the VM is deployed, the agent registers with the Idera backup server. When the agent shows in the Idera web console, the administrator can create a disk safe. The *disk safe* is responsible for backup storage for that VM and defines the encryption and compression settings, as shown in Figure 4-41.

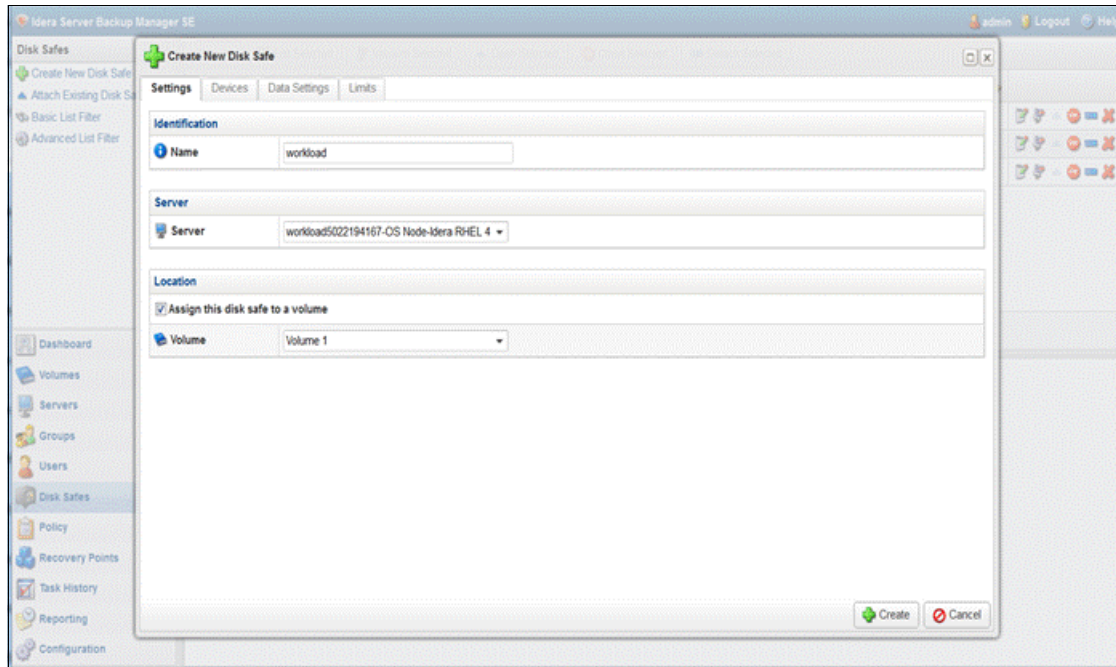


Figure 4-41 Idera disk safe

You can create custom retention policies for each backup schedule. Replication schedules can be set by the hour, day, week, month, or year, as shown in Figure 4-42.

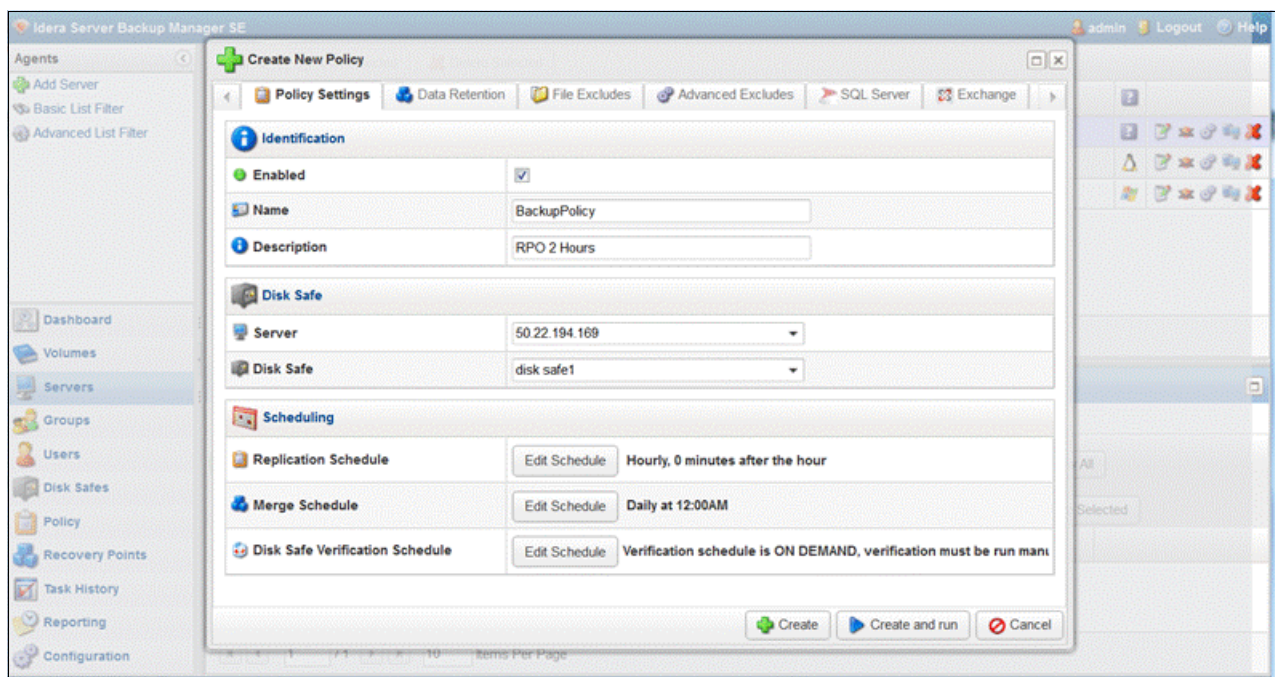


Figure 4-42 Backup retention policy

A list of directories is included in the policy definition. This list of directories is excluded from the backup so that you can include only the data that is relevant to the backup. See Figure 4-43.

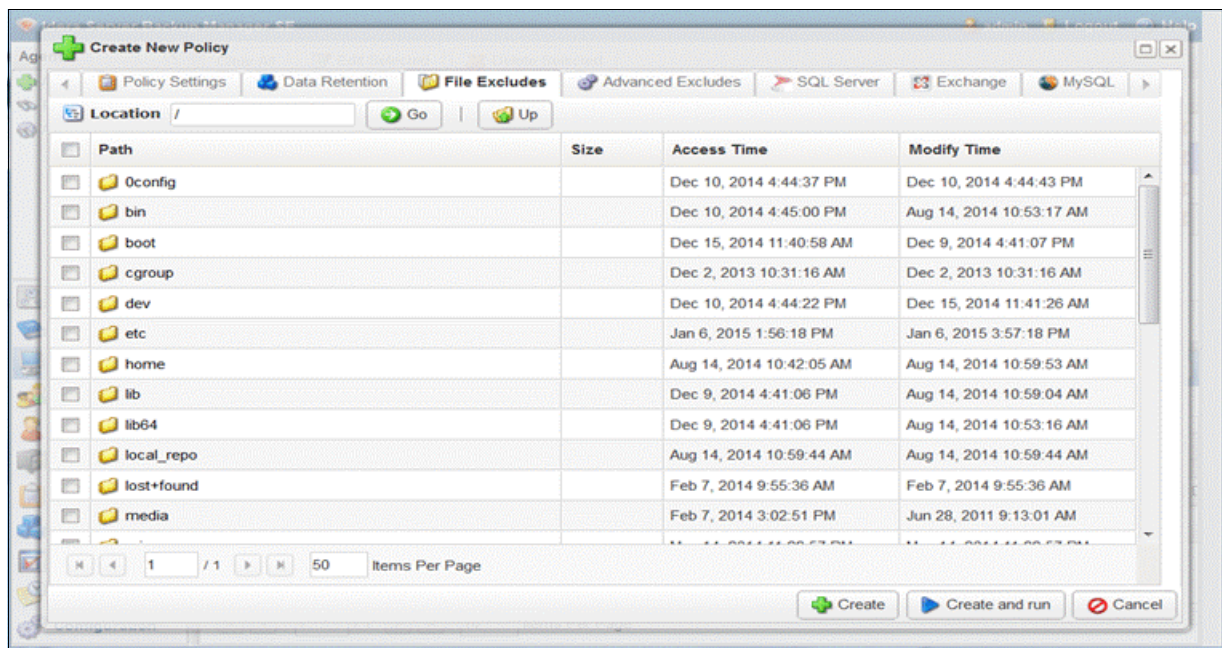


Figure 4-43 Exclude file selection

To restore from a previous backup, locate the recovery point that you want. Select the folder browse icon to the right of the recovery point, as shown in Figure 4-44.

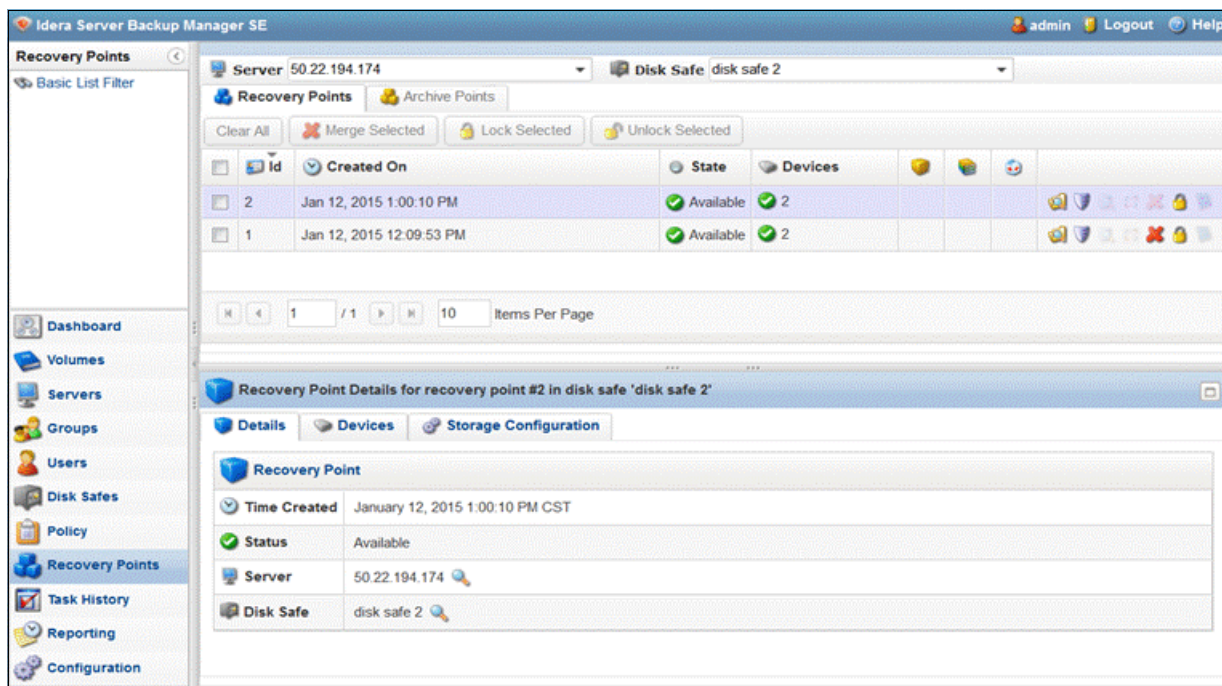


Figure 4-44 Backup recovery point

You can select the entire recovery point or individual files to restore. After you identify the files to restore, select the target agent to restore the files. Files can be restored from an agent on one VM to an agent on a different VM. See Figure 4-45.

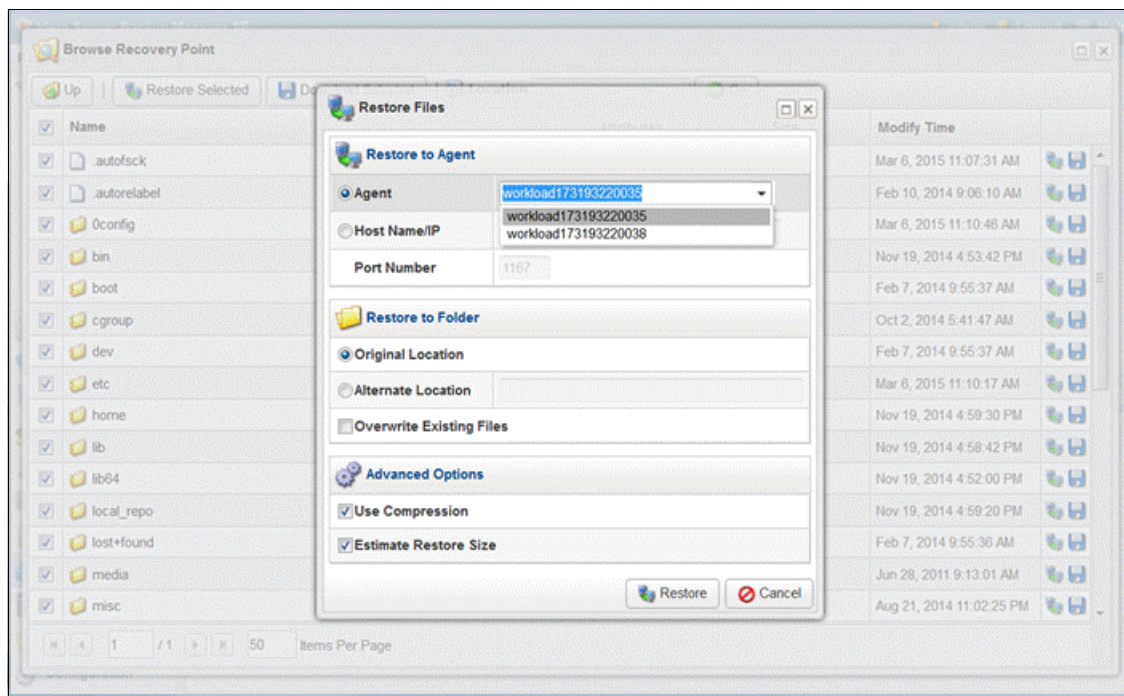


Figure 4-45 Select Restore Agent

For more information, see these websites:

- ▶ *Managing backup and restore* (for PureApplication Service):
<https://ibm.biz/BdHhuP>
- ▶ The Idera server backup Documentation page:
<http://wiki.r1soft.com/display/ServerBackup/Documentation>

4.5 Quick delivery use case with TradeLite

This section explores the use case of delivering an application quickly by using PureApplication Service. In this use case, a new application, TradeLite, is deployed to an off-premises cloud. The main challenge in this use case is data locality. Accessing data remotely across the IPsec tunnel negatively affects the application's performance. Ideally, we want all data to be accessed from local DB2 datastores. This use case focuses on how we can replicate data from the PureApplication Service environment to the client's data center.

4.5.1 Data replication architecture

The Information Management portfolio has multiple ways to replicate data between two sites. The two most common methods are by using either Queue Replication (QRep) or Change Data Capture (CDC). Both of these products allow bidirectional data replication by using a change capture agent that scans the database log files for transactions. The benefit of using a log capture agent is that the replication process does not slow down transactions that are running simultaneously with the replication.

QRep provides the lowest replication latency and the lowest impact on system performance. One negative aspect of QRep is that the QRep setup is more complex and QRep costs more to acquire. CDC provides an environment that is simple to set up and costs less. One negative aspect of CDC is its greater impact on system performance. In our scenario, large data volumes are not an issue, so CDC replication is a good fit.

CDC replication consists of the following components:

- ▶ The CDC agent: This software is installed on the DB2 server and manages the data capture and apply process.
- ▶ Access server: This server manages the subscription sets. After the data replication subscription sets are started, the two DB2 systems communicate directly with each other.
- ▶ Client-based UI: This UI manages the configuration of the data replication environment.

Figure 4-46 shows the components in our replication use case.

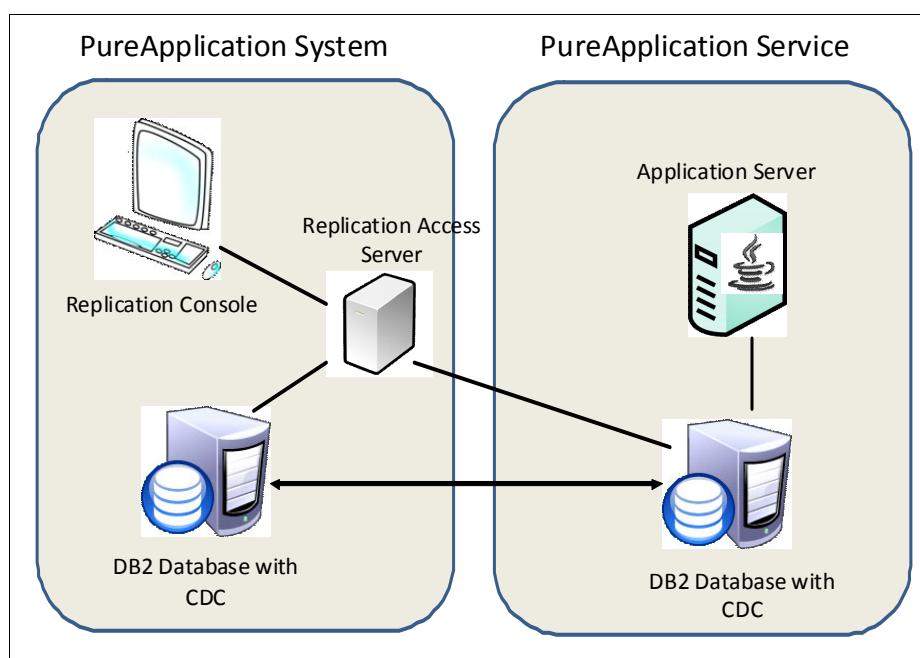


Figure 4-46 CDC components of our replication use case

An in-depth description of data replication is beyond the scope of this book. However, we describe the high-level required steps to set up the replication. For more information, see the following books:

- ▶ *High Availability and Disaster Recovery Options for DB2 for Linux, UNIX, and Windows*, SG24-7363:
<http://www.redbooks.ibm.com/abstracts/sg247363.html?Open>
- ▶ *Smarter Business: Dynamic Information with IBM InfoSphere Data Replication CDC*, SG24-7941:
<http://www.redbooks.ibm.com/abstracts/sg247941.html?Open>

4.5.2 Artifacts to automate deployment

The PureApplication family does not support CDC data replication without configuration. With only a few configuration steps, you can automate most of the process by using script packages and virtual system patterns. Two custom script packages are used in this use case:

- Install and configure the replication access server
- Install the change data agent on DB2

The required binary files to install the necessary components are included in each script package. The binary files are 150 MB - 200 MB. Although these binary files are larger than most script packages, they are within the suggested practice of keeping script packages under 500 MB.

The script package that is named `Configure Access Server` installs the access server on a base OS by using a response file and a silent installation. After the access server is installed, the instance is started. No other additional configuration setup is required for the access server. Example 4-10 shows the options file for the silent installation.

Example 4-10 Silent installation response file for the replication access server

```
# This file was built by the Replay feature of InstallAnywhere.  
# It contains variables that were set by Panels, Consoles, or Custom Code.  
#Has the license been accepted  
#-----  
LICENSE_ACCEPTED=TRUE  
  
#Choose Access Server Port  
#-----  
as.port=10101  
  
#Choose Install Folder  
#-----  
USER_INSTALL_DIR=/opt/IBM/InfoSphereChangeDataCapture/AccessServer
```

The CDC agent script package installs the DB2 agent to capture DB2 transactions from the system log. The software is installed by using a silent installation with an options file. This script package does not configure the agent. Additional configuration is required to configure and start the agent. See Example 4-11. The required steps to configure the CDC agent are described in 4.5.4, “Required steps to set up data replication” on page 191.

Example 4-11 Silent installation response file for the CDC agent

```
# Thu Mar 26 23:46:01 UTC 2015  
# Replay feature output  
# -----  
# This file was built by the Replay feature of InstallAnywhere.  
# It contains variables that were set by Panels, Consoles, or Custom Code.  
#Indicate whether the license agreement was accepted  
#-----  
LICENSE_ACCEPTED=TRUE  
  
#Choose Install Folder  
#-----  
USER_INSTALL_DIR=/opt/IBM/InfoSphereChangeDataDelivery/ReplicationEngineforIBMDB2  
  
#Install
```

```
#-----  
-fileOverwrite_/opt/IBM/InfoSphereChangeDataDelivery/ReplicationEngineforIBMDB2/pr  
operties/version/IBM_InfoSphere_Change_Data_Delivery-10.2.1.swtag=Yes  
  
#Install Complete  
#-----
```

4.5.3 Virtual system patterns

This use case uses two virtual system patterns. The first pattern sets up the DB2 database and Access Server on the PureApplication System. The deployment in a development environment is simplified if the database and the access server are in the same pattern. However, these components probably need to be separated for a production environment. Figure 4-47 shows the pattern that is used to deploy the on-premises servers.

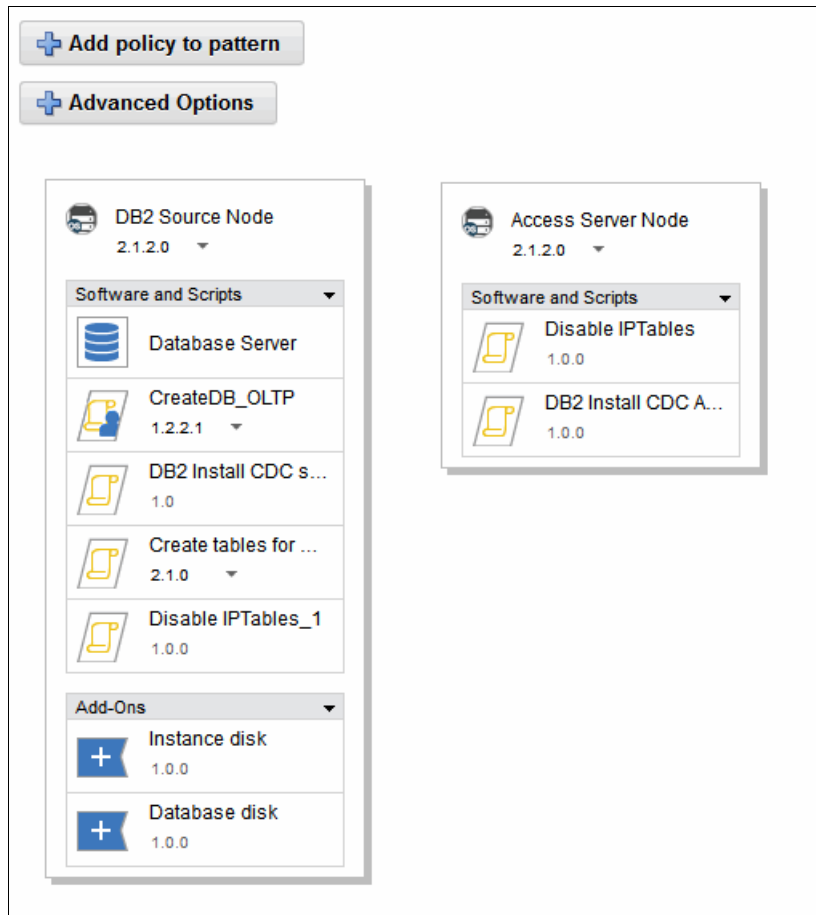


Figure 4-47 Virtual system pattern for on-premises servers

The second pattern deploys the application server and DB2 database. Our environment uses a simple stand-alone WebSphere Application Server environment. A production-level environment uses HTTP servers with stand-alone or clustered application servers.

Figure 4-48 shows the pattern that is used to deploy the off-premises DB2 server and application server.

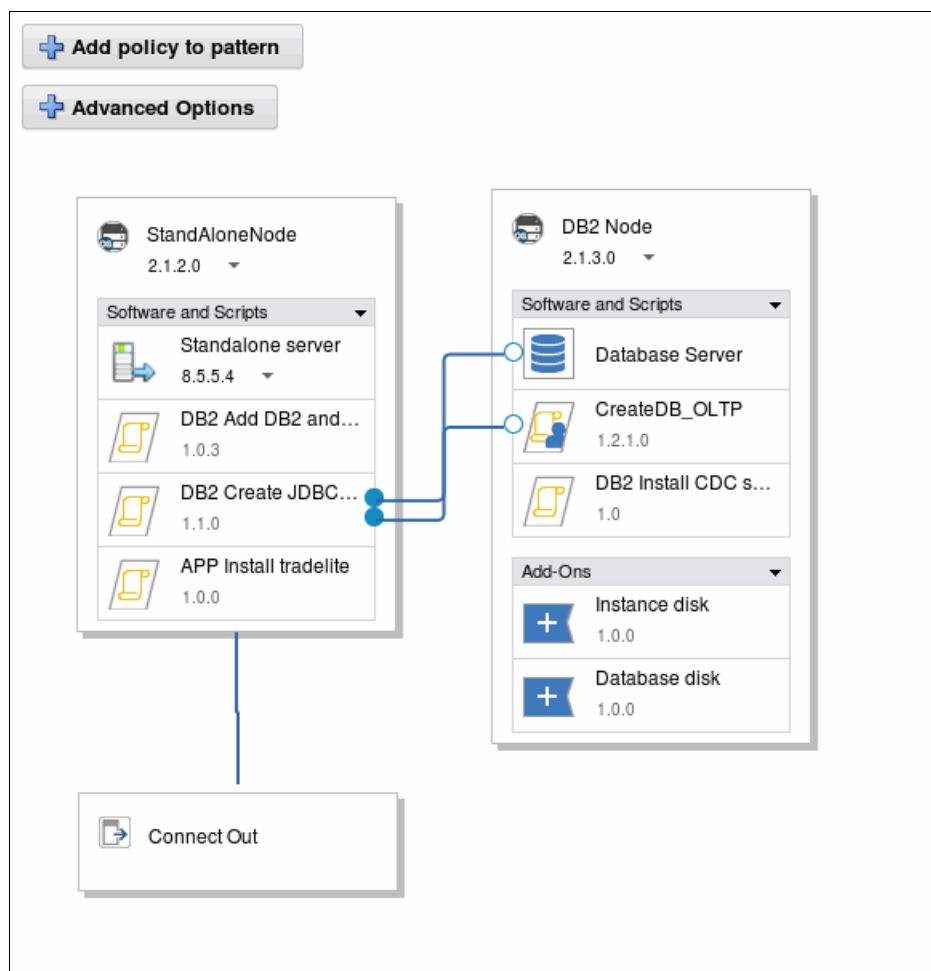


Figure 4-48 Virtual system pattern for off premises

4.5.4 Required steps to set up data replication

After both patterns are deployed, a few manual steps are required to complete the deployment:

1. Configure the DB2 CDC capture agent.
2. Catalog each DB2 instance in the CDC console.
3. Create a subscription set for data replication.
4. Start data replication.

The CDC agent must be configured for each DB2 server. To configure DB2, perform the following steps:

1. Log in to the VM as the DB2 instance owner.
2. Create the schema for the CDC control tables:

```
db2 create schema CDC
```

3. Create the table load directory:

```
mkdir /db2fs/load
```

4. Change to the CDC directory:

```
cd /opt/IBM/InfoSphereChangeDataDelivery/ReplicationEngineforIBMDB2/bin/
```

5. Run the **configure** command:

```
./dmconfigurets
```

Example 4-12 shows the required steps to configure and start the CDC agent.

Example 4-12 Sample output for the CDC agent configuration

```
[db2inst1@pure-9-3-172-217 bin]$ ./dmconfigurets
Welcome to the configuration tool for IBM InfoSphere Change Data Capture (IBM
DB2). Use this tool to create instances of IBM InfoSphere Change Data Capture (IBM
DB2).
```

```
Press ENTER to continue...
```

```
Initializing. Please wait...
```

```
CONFIGURATION TOOL - CREATING A NEW INSTANCE
```

```
-----
```

```
Enter the name of the new instance: cdcinst1
```

```
Enter the server port number [10901]:
```

```
Enter the auto-discovery port number or type 'DISABLE' [DISABLE]:
```

```
Staging Store Disk Quota is used to limit the disk space used by IBM InfoSphere
Change Data Capture staging Store. If this space is exhausted, this instance may
run at a lower speed. The minimum value allowed is 1 GB.
```

```
Enter the Staging Store Disk Quota for this instance (GB) [100]: 5
```

```
Enter the Maximum Memory Allowed for this instance (MB) [1024]:
```

```
Select y to use JMS or TCP/IP engine communication connection, select n to use TCP
only engine communication connection (y/n) [n]:
```

```
Select a DB2 Instance
```

```
1. db2inst1
```

```
2. Other...
```

```
Select a DB2 Instance [db2inst1]: 1
```

```
Select a database name
```

```
1. TRADEDB
```

```
2. Other...
```

```
Select a database name: 1
```

```
Would you like to configure advanced parameters (y/n) [n]:
```

```
Enter the username: db2inst1
```

```
Enter the password:
```

```
Enter a database schema for metadata tables or press ENTER to list schemas: CDC
```

```
Would you like to specify a refresh loader path (y/n) [y]:
```

```
Enter the refresh loader path: /db2fs/load
```

```
Creating a new instance. Please wait...
```

```
Instance cdcinst1 was successfully created.
```

```
Would you like to START instance cdcinst1 now (y/n)?y
```

```
Starting instance cdcinst1. Please wait...
```

```
Instance cdcinst1 started successfully. Press ENTER to go to the Main menu...
```

```
MAIN MENU
```

```
-----
```

```
1. List Current Instances
```

```
2. Add an Instance
```

```
3. Edit an Instance
```

```
4. Delete an Instance
```

```
5. Consolidate Instances
```

```
6. Exit
```

```
Enter your selection:6
```

```
Exiting...
```

```
[db2inst1@pure-9-3-172-217 bin]$
```

4.5.5 CDC management console

After you configure the agents, you perform all of the additional configuration by using the CDC management console. The management console is a Windows utility that interacts with the access server to configure and monitor the replication process.

Follow these steps for the remaining configuration:

1. When you start the console, you are prompted to change your password if it does not meet the password strength requirements, as shown in Figure 4-49.

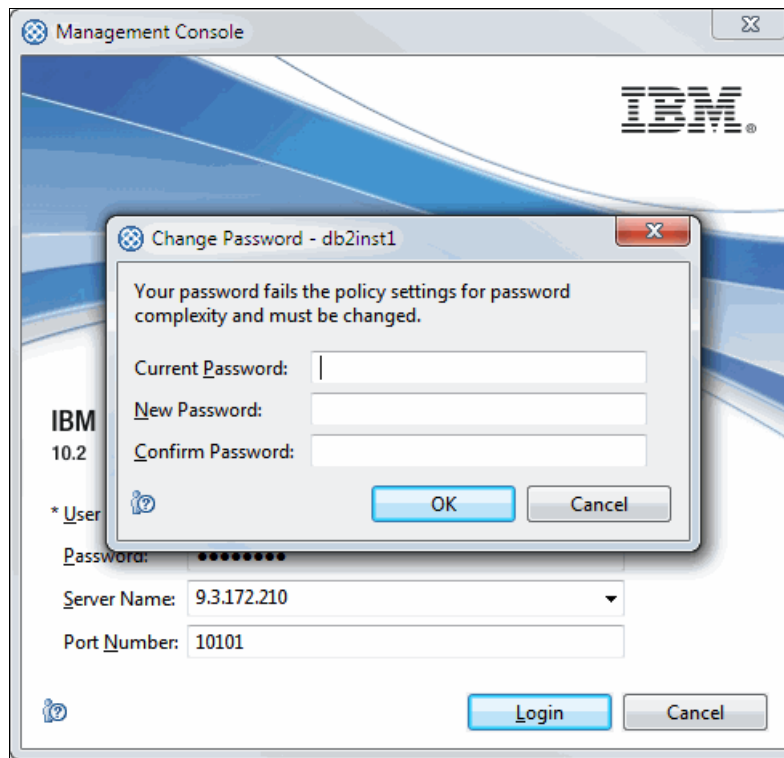


Figure 4-49 CDC management console window to change your password

2. After you successfully log in to the server, you need to define the DB2 datastore. On the Access Manager tab, right-click and select **New Datastore**. Provide a name for the datastore and its host name. Click **Ping** to test the connection. Enter the necessary information in the Properties section. You need to define the DB2 server for both the service and the system. See Figure 4-50.

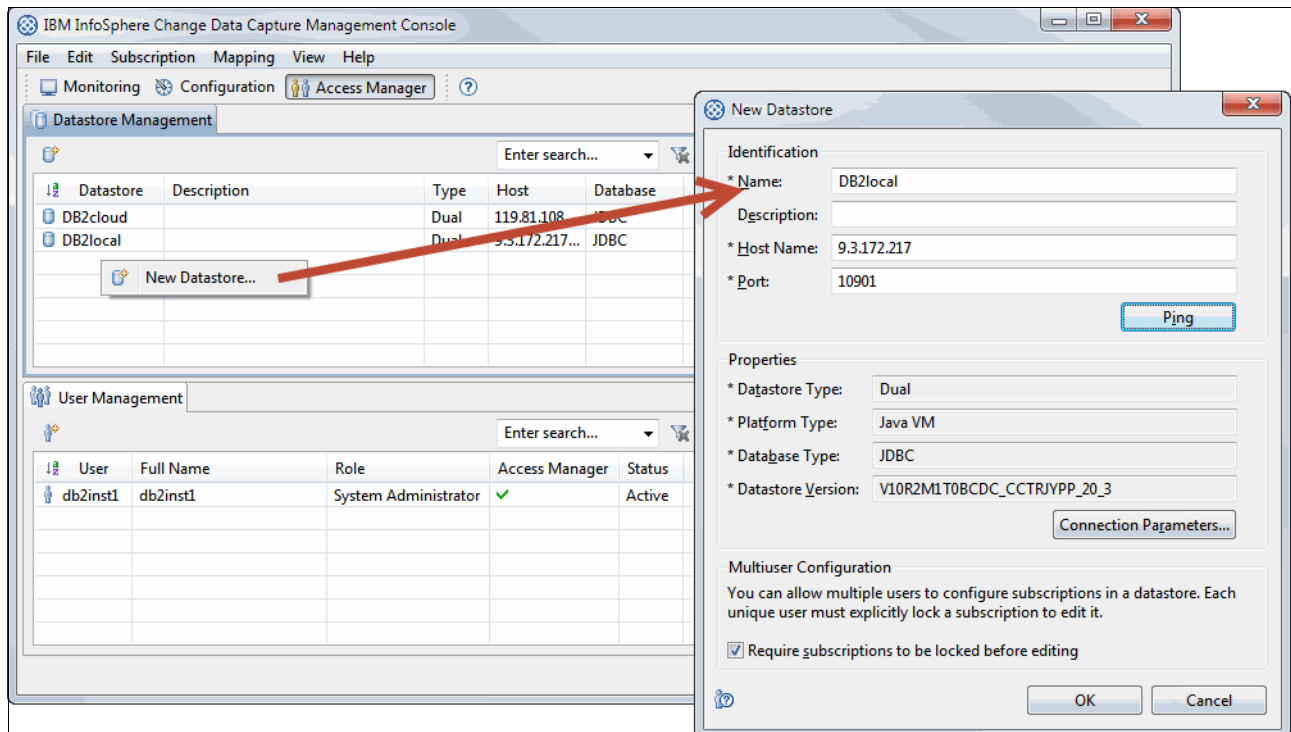


Figure 4-50 Create a datastore

3. On the Configuration tab, under Subscriptions, right-click **Default Project**, and select **New Subscription**. Provide a name for the subscription set, and select the source and target datastores. See Figure 4-51.

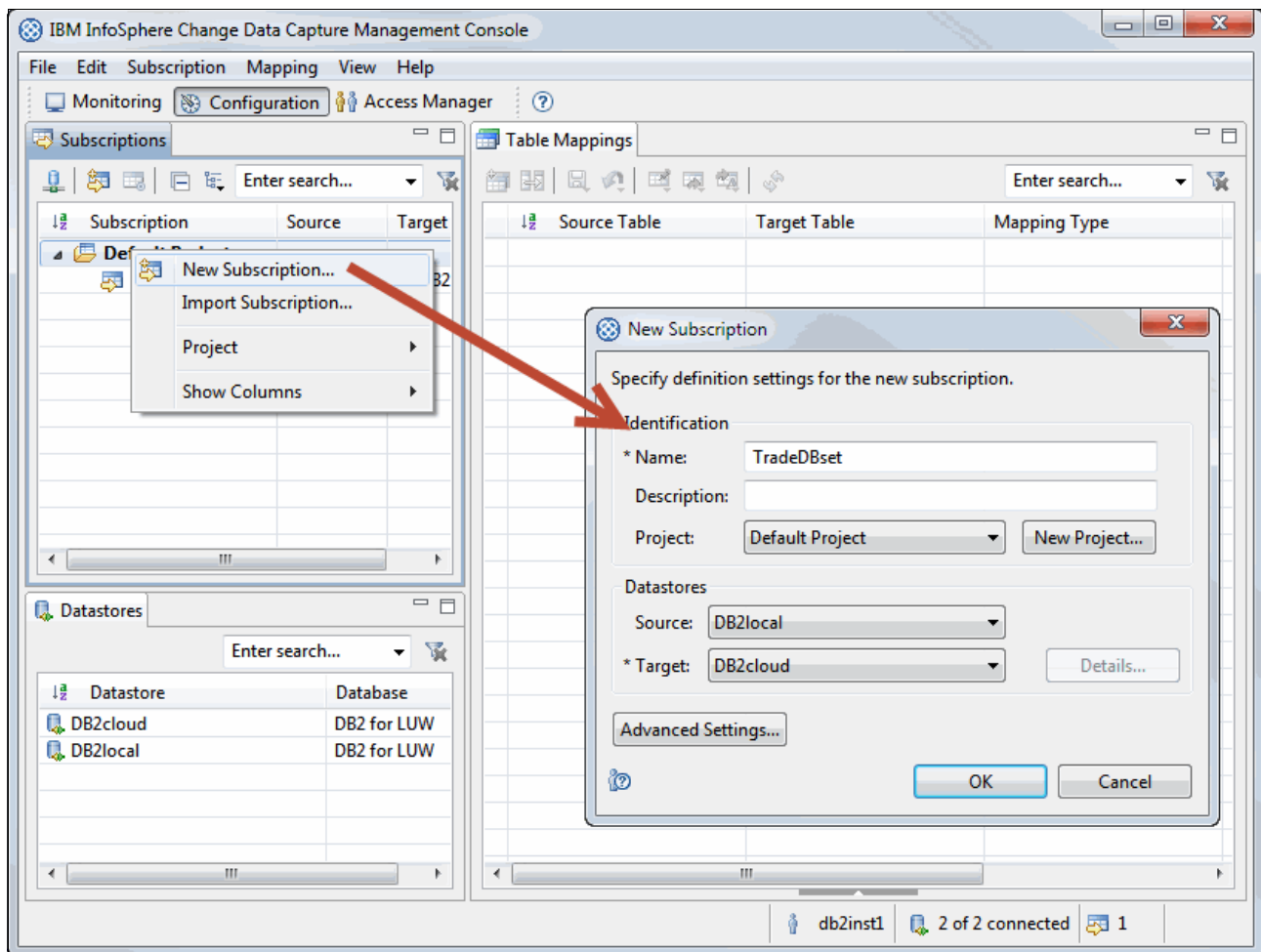


Figure 4-51 Create the replication subscription

4. Use the Map Tables window to select the mapping type. Accept the default values by clicking **Next**.

5. The next option is to choose the source table to replicate. On the Select Source Tables window, expand the schema that you want. Select the source table to replicate, as shown in Figure 4-52.

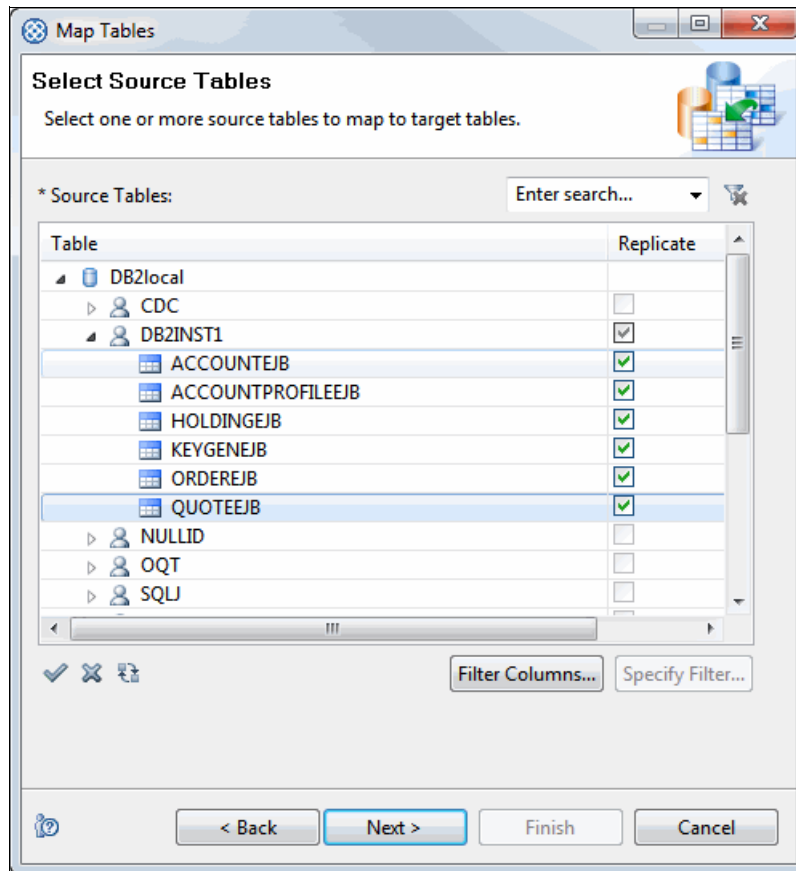


Figure 4-52 Source table selection

6. After you select the source table to replicate, the Complete Mappings window prompts you to select the target table mapping. After you match the first table, the system attempts to map all remaining tables by mapping tables with the same name. Accept the defaults to complete the creation of the subscription set. See Figure 4-53.

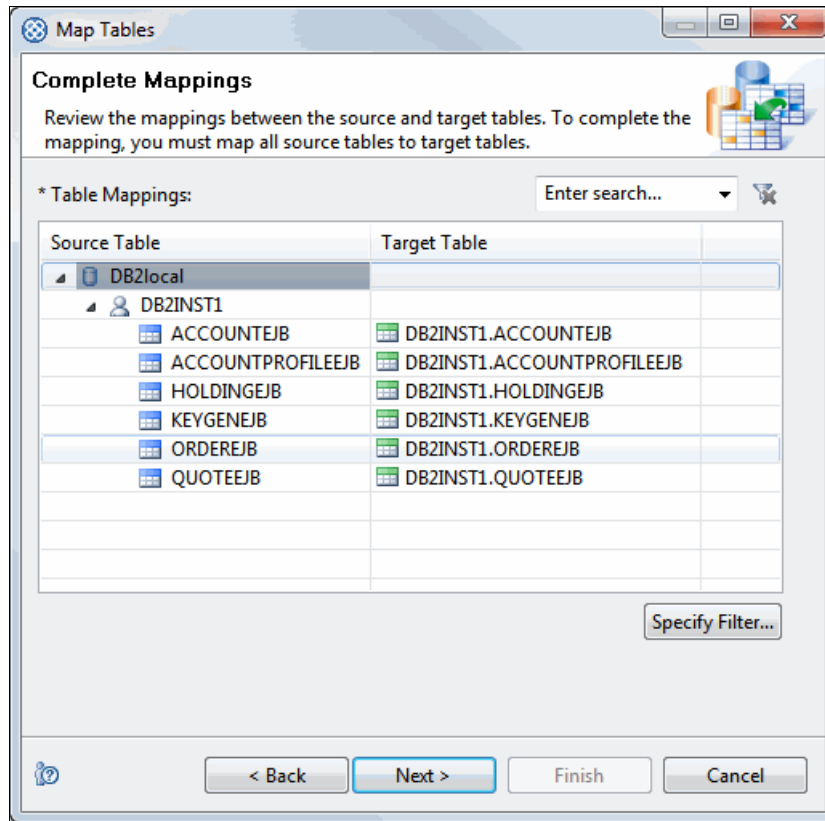


Figure 4-53 Target table mapping

- Now that the subscription set is complete, start data replication. On the Monitoring tab, right-click the subscription set and select **Start Mirroring**. Accept the default mirroring method of **Continuous**. After a few minutes, the subscription set changes from an inactive state to a mirroring state. See Figure 4-54.

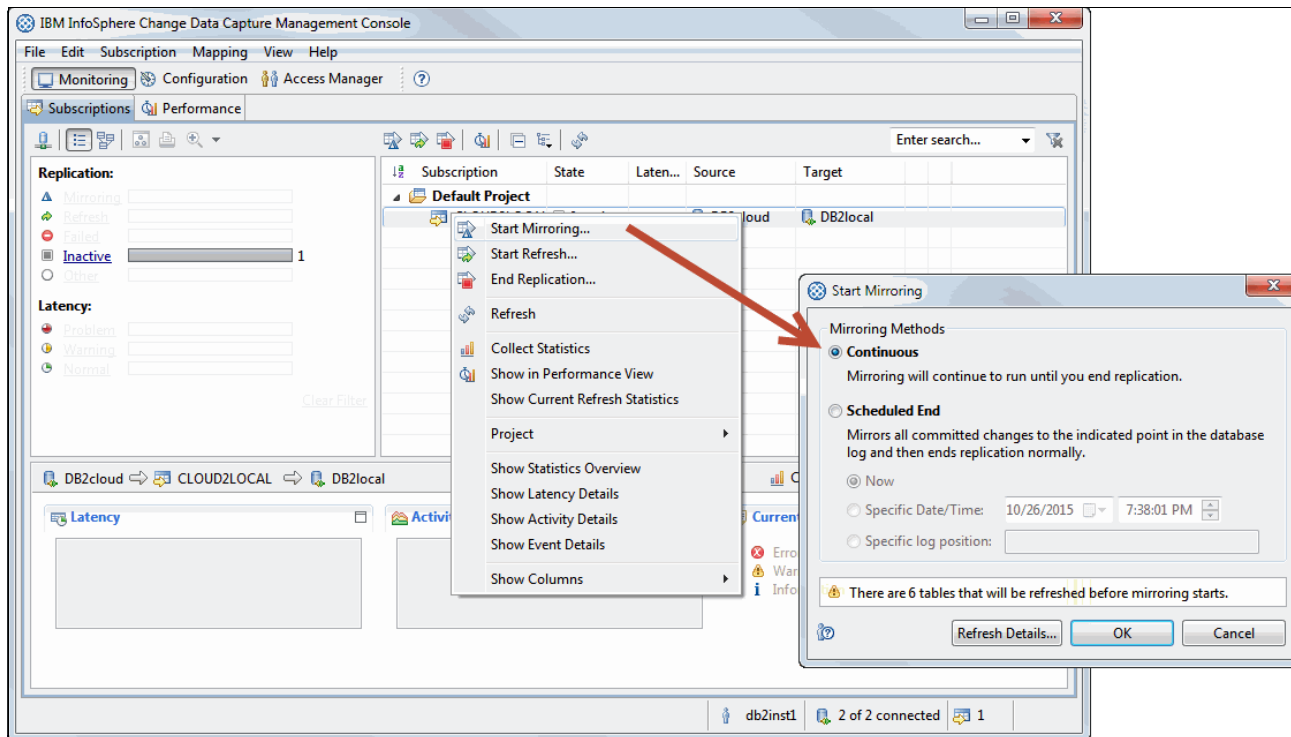


Figure 4-54 Starting data replication

- The application is ready to test. Any updates that are made by the TradeLite application are replicated to the local copy. You can set up bidirectional replication also so that updates from the local copy are replicated to the remote copy.

Because the TradeLite application is interacting with the local copy of data, the speed of the communication link does not affect performance. Optionally, you can choose to not back up the off-premises data because you are replicating data to the local copy.

9. Figure 4-55 shows the performance of the data replication. On the chart, you can see a spike in the network traffic when the application refreshes the database with new data.

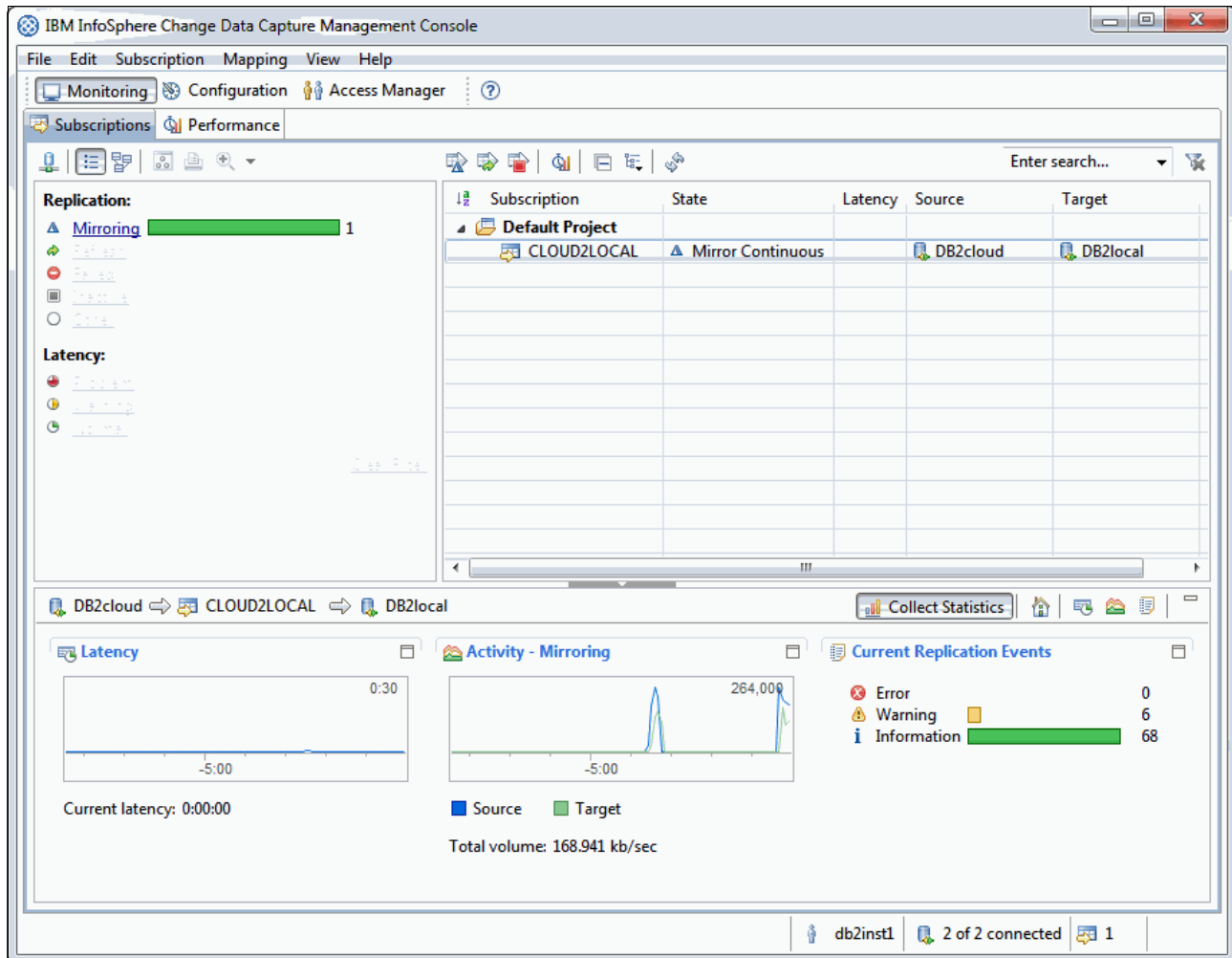


Figure 4-55 Monitoring replication performance

4.6 Deployment and bringing it all together

With important aspects complete, such as connectivity, isolation, security, and disaster recovery, we can define and describe how to bring all of these aspects together and share insights about deploying to your hybrid cloud.

To determine how to choose between on premises and off premises, either separate or together, consider where a specific part of your solution or architecture will run and in which development stage the application will be. Consider also any options to combine IBM PureApplication technology with other vendors' cloud platforms.

How to size your cloud is an easy question to address. If your budget is limited, the best advice is to start small, ensuring that your cloud solution matches at least the minimal requirements for the products or applications that you want to run in that cloud. It is better to allow at least a little margin, so that you can benefit from more storage, more memory, or more compute power in a spike in the load. The best answer to the question of sizing your workload is that it depends on your environment:

- ▶ If you are looking for a business continuity solution, you require a capacity that is at least close to the capacity that you have on premises.
- ▶ If you are looking for a pure disaster recovery solution, you need an environment that is large enough to sustain your most basic and important business tools and services.
- ▶ If you are looking for a development and testing (hybrid) cloud environment, opt for an environment that slightly exceeds the maximum that is needed to run, for instance, your performance tests.

Note on sizing: For the IBM PureApplication Service, the environment flattens out on entire compute nodes (4, 8, or 16). You cannot order an environment of 10 cores. For an overview of the offering, see Chapter 1, “Enabling a hybrid enterprise” on page 1.

4.6.1 Test and development off premises and production on premises

The most frequent use case is to create, test, and deploy an application, then repeat by using an iterative approach, and finally pushing out the application into production.

To obtain this environment, the patterns that are used in the PureApplication family represent the ideal model. They are portable between among cloud environments. They are easy to build and maintain. And, they facilitate identical deployments of your application architecture in different cloud environments.

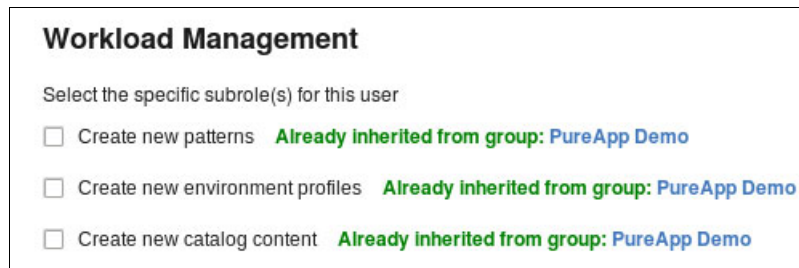
Consider the following key points:

- ▶ The need for customization: You most likely need to include, for instance, security features to support authentication with your corporate user and group repository.
- ▶ The need for further tweaking and tuning: The IBM predefined settings within a pattern are acceptable to start. However, the IBM predefined settings require further changes to become production-ready.
- ▶ The ability to export and import between off-premises environments and on-premises environments by using an identical GUI to export from test or development environments: You can use this capability to import the artifacts back into your on-premises environment. This capability is a unique feature.
- ▶ Use the support for DevOps tools, such as UrbanCode Deploy, to use tools, such as Jenkins and GIT, to track changes by using external repositories.

The following requirements for this scenario are listed:

- ▶ An (administrative) workstation with access to both environments
- ▶ The required rights to import and export patterns and parts, if needed
- ▶ Version management and strict control of the artifacts in the PureApplication catalog

To export items from and import items into the catalog, you need the rights to update, add, and preferably delete artifacts, as shown in Figure 4-56.



The screenshot shows a 'Workload Management' section with a heading 'Select the specific subrole(s) for this user'. Below this, there are three unchecked checkboxes, each followed by a green status message: 'Create new patterns' with 'Already inherited from group: PureApp Demo', 'Create new environment profiles' with 'Already inherited from group: PureApp Demo', and 'Create new catalog content' with 'Already inherited from group: PureApp Demo'.

Figure 4-56 Necessary rights to export from and import into the IBM PureApplication catalog

Use the following brief and limited steps to create, export, import, and deploy a pattern when you move it from an IBM PureApplication Service to IBM PureApplication System. This scenario assumes that the PureApplication Service instance is the testing environment and that PureApplication System is the production environment.

Follow these steps:

1. Start in the pattern designer (Figure 4-57 on page 203), where the pattern is developed or created, which becomes the blueprint of the application.

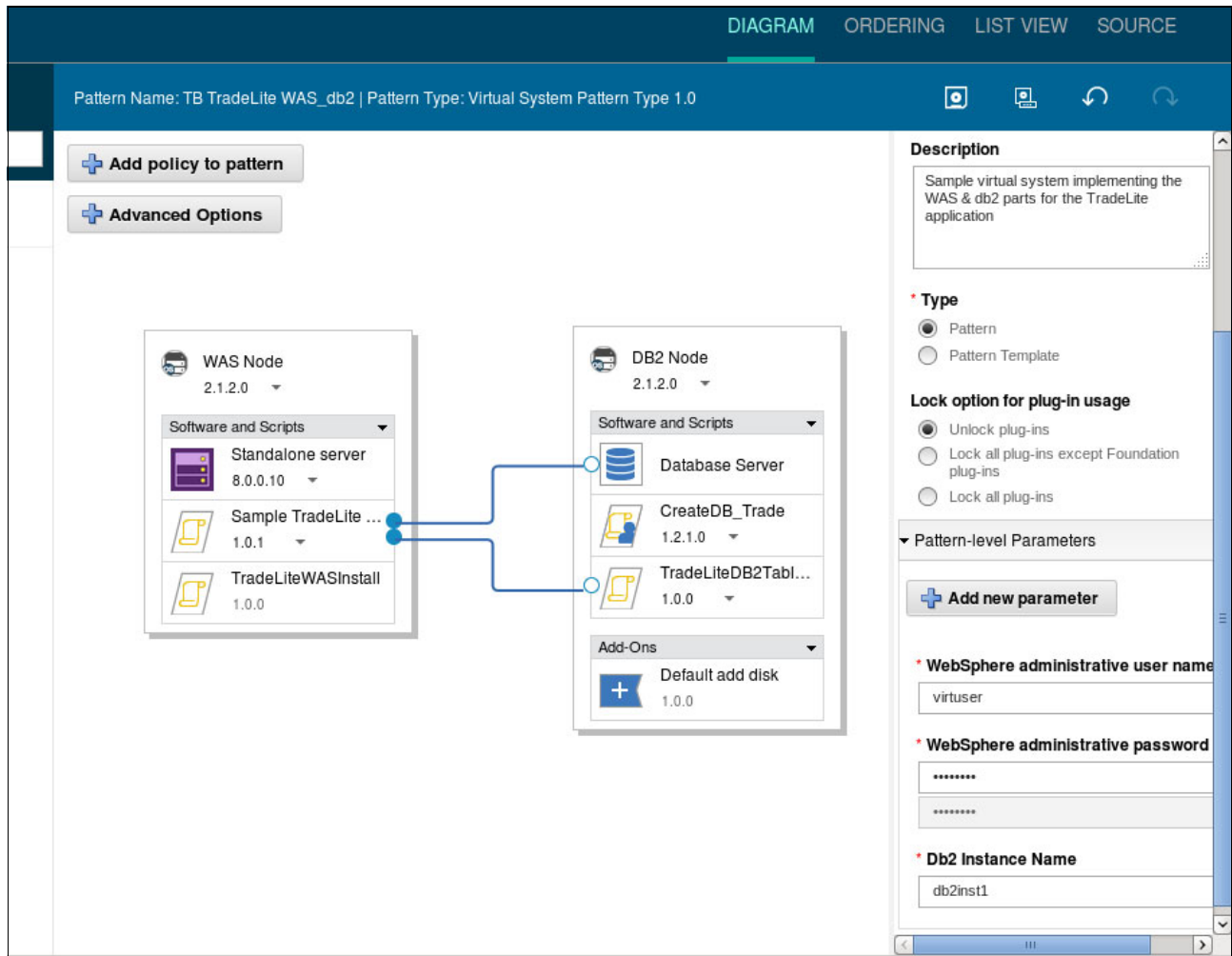


Figure 4-57 Pattern designer in the PureApplication Service GUI

This pattern is a blueprint for an application that relies on an application server, a single node, and a database server. Scripts are available for the following tasks:

- Deploy the application.
 - Create the database.
 - Create the tables.
 - Link the application to the database.
2. After the pattern satisfies the requirements, the pattern can be exported to the other part of the hybrid cloud, as shown in Figure 4-58 on page 204.

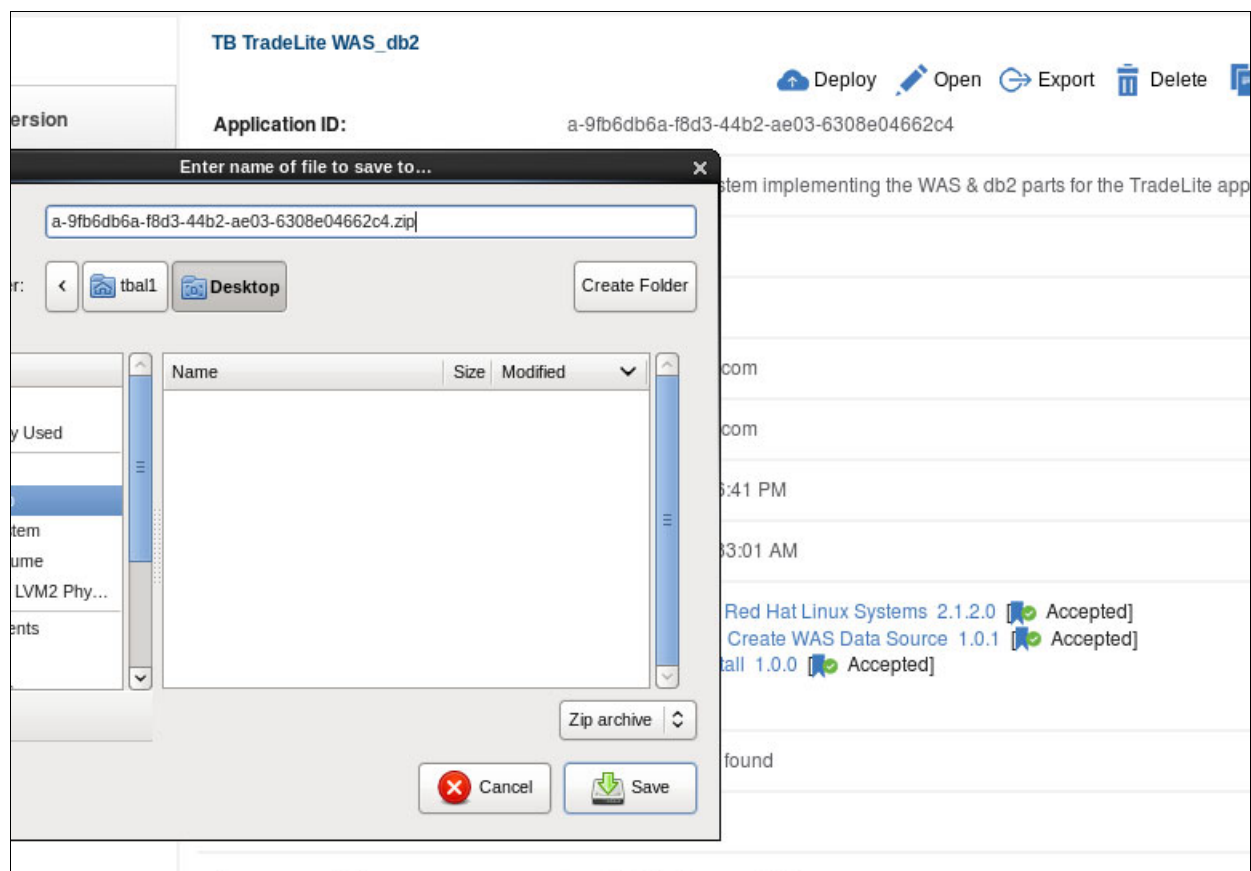


Figure 4-58 Exporting a pattern from the PureApplication Service catalog

3. The export operation creates an archive (compressed) file, which contains .json files that define the application architecture and components.

4. Import the pattern into the catalog on the PureApplication System (Figure 4-59).

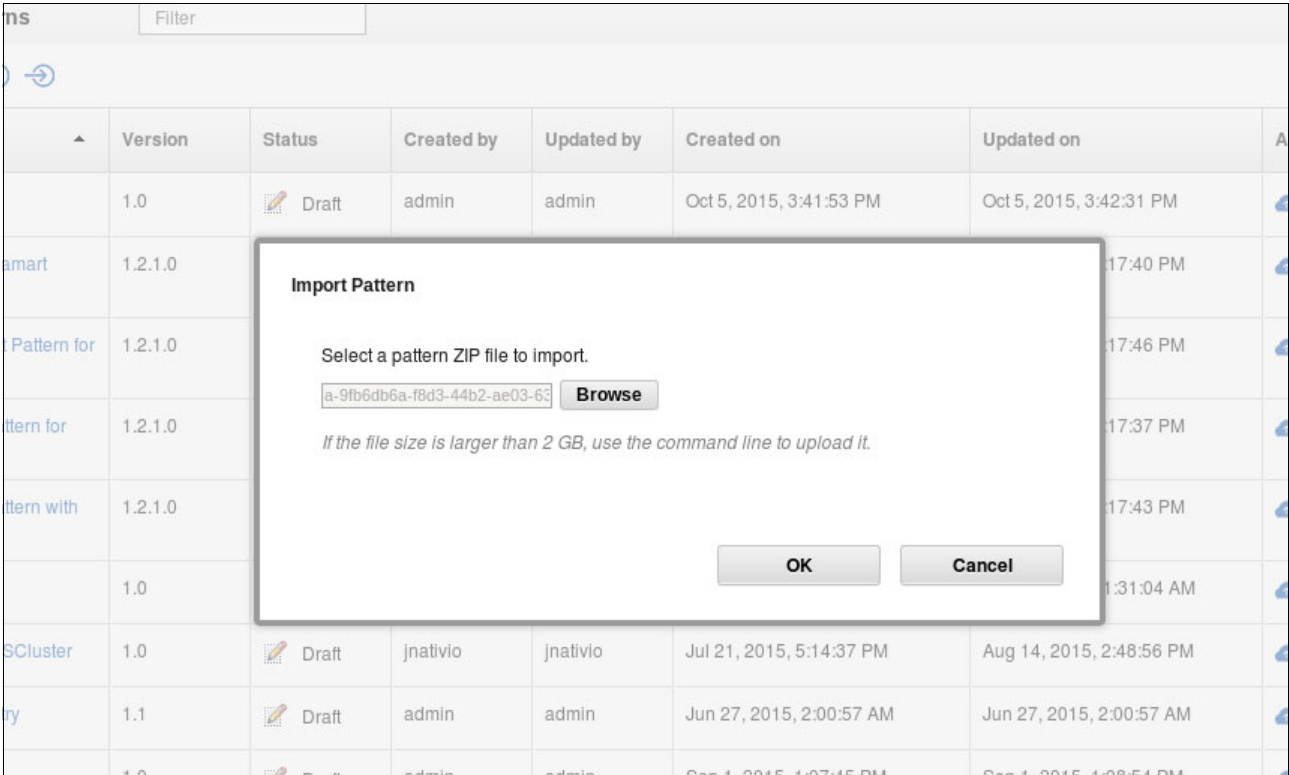


Figure 4-59 Importing a pattern into PureApplication System

If, during the import process, a part is missing, the GUI reflects this missing part. If, during the import, a duplicate is detected, the system flags it with a message, as shown in Figure 4-60.

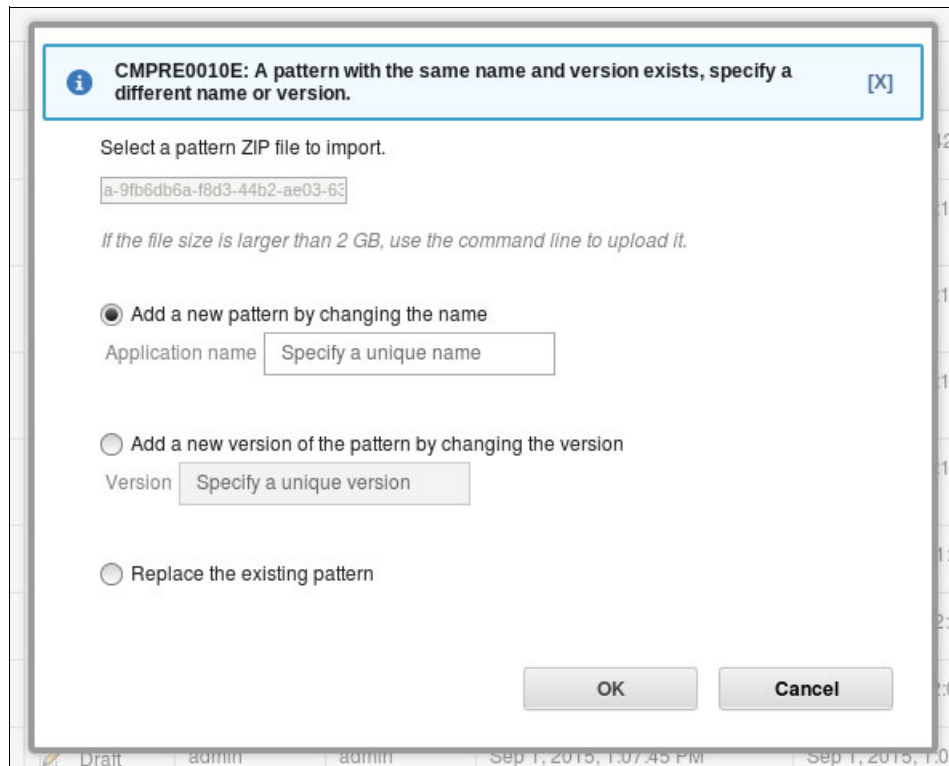


Figure 4-60 A pattern with an identical name is identified and the options are presented

Note: The name of the pattern is also stored in the exported archive file.

5. When all parts are in place, you can redeploy the pattern on PureApplication System, as shown in Figure 4-61.

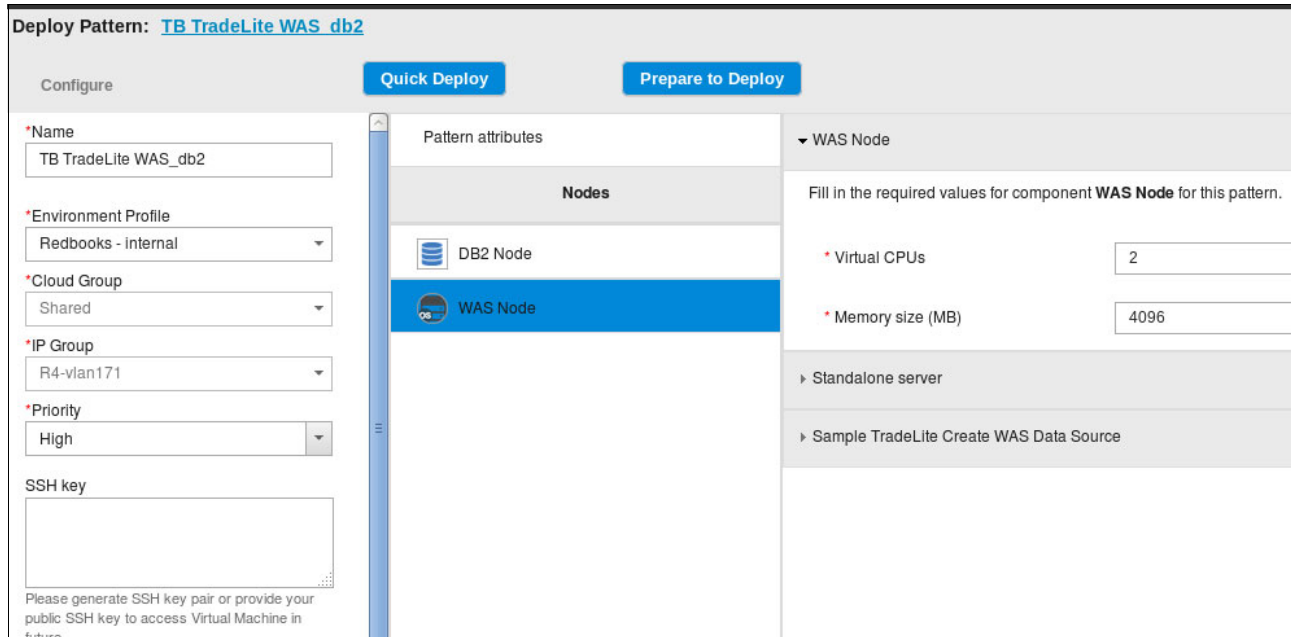


Figure 4-61 Deploying the pattern on the PureApplication System side of our hybrid cloud environment

Exporting and importing for either a virtual application or a virtual system pattern only work if the same components are installed in both environments. Those components also need to be at the same level. For instance, when you use a database, such as DB2, ensure that the necessary catalog artifacts (same product type and version) to install that database and DB2 server are available on both environments.

In particular, when you use a virtual system, pay attention to script names and other part names because they must match. When parts do not match or parts are missing, a warning message is shown, as illustrated in Figure 4-62.

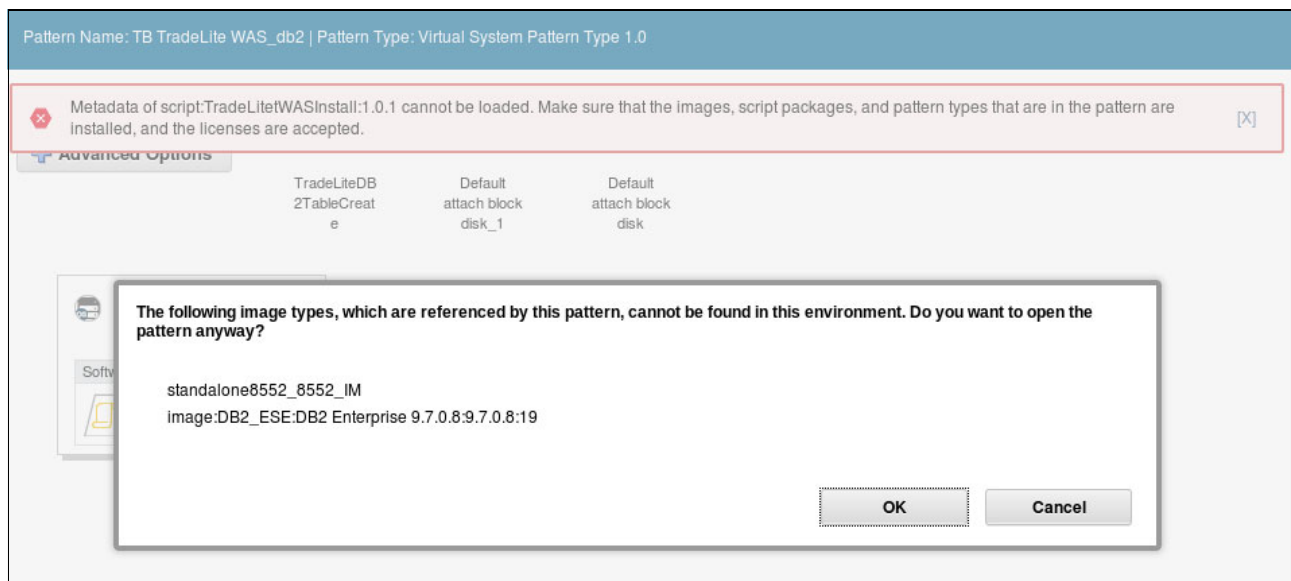


Figure 4-62 Metadata error message when particular pattern parts are missing

Preferably, the following components must be identical for both environments:

- ▶ Version of the PureApplication Software
- ▶ Versions of the software packages (middleware) that you are entitled to
- ▶ Default data set that comes with a particular version

For example, if you move a virtual application pattern that relies on a Foundation Pattern Type that is higher than the Foundation Pattern Type that was available when you created the pattern, the deployment and attachment to certain services fail. The frameworks for deployment no longer match.

When you import a pattern into a system where certain components, such as pattern types, script packages, or any other components, are missing or are at a lower level, the system shows a yellow triangle in the GUI. Also, consider the difference in access rights when you use different users for both cloud environments.

Note: What is the default data set or software package that a client is entitled to?

The *default data set* is often referred to as the *collective* of catalog artifacts that are shipped in combination with a particular version of PureApplication System, PureApplication Software, or PureApplication Service. The default data set is usually a large archive file (on average, 40 GB).

This file set contains, for instance, pattern types and scripts to implement new features that are associated with that version of the PureApplication middleware management or deployment engine. This file set requires a separate upload to the PureApplication catalog, which is usually performed and maintained by your staff, except for PureApplication Service, for which the base catalog artifacts are maintained by the IBM DevOps team that manages the IBM offering.

Often, the pattern that is used for development is different from the pattern that is used for testing or production. Manual changes might be required. For example, a pattern that is deployed on a physical system might require a storage volume that is external or internal to PureApplication System. Another example involves the location of the various parts of your architecture. Is the database or web server VM internal or external? Several of these configurations might not be easily transferable between two different cloud environments.

The deployment engine automatically includes references to the various shared services that are available. Therefore, during deployment, do not manually intervene and change, tweak, or tune the references to monitoring, caching, or Red Hat Satellite Service. All of this work is done for you. It is another advantage of building a hybrid cloud by using PureApplication products.

For more information about how to integrate an IBM PureApplication solution, see these books:

- ▶ *Integrating an IBM PureApplication Environment*, TIPS1328
<http://www.redbooks.ibm.com/abstracts/tips1328.html?Open>
- ▶ *Integrating IBM PureApplication System into an Existing Data Center*, SG24-8285
<http://www.redbooks.ibm.com/Redbooks.nsf/RedbookAbstracts/SG248285.html>

The steps, tools, and screen captures to import and export a virtual system differ from the steps, tools, and screen captures of a virtual application, with or without using the Plug-in Development Kit (PDK) from the Eclipse development environment.

Ideally, you can set up a management domain for PureApplication System and PureApplication Software. You can use the command-line interface (CLI) tool to keep the catalogs of your PureApplication System synchronized with PureApplication Service. Exporting pattern types and parts, importing them at the other end, and comparing versions of patterns are what you are looking for. You can download the CLI from the welcome page in the PureApplication GUI.

4.6.2 Application parts off premises, parts on premises, and orchestration

Your hybrid cloud solution is built. How do you decide which applications or what kind of workload you bring on premises or take off premises? And what can you rely on to orchestrate this process?

A common use case for using a hybrid cloud is when a less critical workload or less critical applications are placed in an off-premises cloud. Often, this workload or the applications are moved so that external IT service providers can access the remote environment easily for training, development, and testing.

It is not possible to outline a single checklist to apply to all client cases because each case depends on various factors. Certain clients are driven toward the hybrid cloud solution because, in their environment, it provides the ideal balance between capital expenditures (CAPEX) and operating expense (OPEX).

The products that we focus on do not contain a data center orchestration engine or data center orchestration layer. If a true orchestration layer is required for your data center, IBM Cloud Orchestrator (ICO) integrates with one or more PureApplication Systems and your off-premises infrastructure to add orchestration to the capabilities of your on-premises cloud solution.

For more information, see these resources:

- ▶ *IBM Cloud Orchestrator: Transform your IT department into a self-service organization:*
<https://ibm.biz/BdEZu3>
- ▶ *IBM Cloud Orchestrator V2.5 documentation:*
<https://ibm.biz/BdHhj6>
- ▶ *Welcome to IBM Cloud Orchestrator* (IBM Bluemix wiki):
<https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20SmartCloud%20Orchestrator/page/Welcome>

Other orchestration tools and techniques can also be used in combination with the various APIs that are provided for the PureApplication family.

Next, we describe several standards that relate to cloud environments. Your decision to run a particular application, or part of an application, either on premises or off premises, might be imposed by requirements to comply with certain business or industry standards.

Note: Ultimately, it is your responsibility, as the client, to ensure compliance with standards. IBM provides the parts to build a hybrid cloud, but it is your choice and obligation to make and keep your cloud environment compliant with standards.

Compliance with the Payment Card Industry Data Security Standard (PCI DSS) is one of the industry standards that greatly affects deciding the location of parts of your application architecture. A description of the various requirements for PCI DSS compliance is beyond the scope of this publication. However, for more information, see the website for the PCI Security Standard Council:

<https://www.pcisecuritystandards.org/>

The most important consideration is the obligation to protect sensitive customer data. When you evaluate how to protect sensitive customer data in a hybrid cloud, several options exist:

- ▶ Keep the database on premises
- ▶ Provide encryption on all communication interfaces
- ▶ Provide encryption for data-at-rest by using, for instance, the IBM Encryption Pattern for Security First SPxBitFiler-IPA

More information about this pattern is included in 4.2.5, “Securing data at rest” on page 166 and in the PureSystems Centre, which is the pattern application catalog for IBM PureApplication products:

- ▶ *PureSystems Centre: IBM Encryption Pattern for Security First SPxBitFiler-IPA:*

<https://ibm.biz/BdHhLX>

- ▶ *HHS.gov: Improving the health, safety, and well-being of America:*

<http://www.hhs.gov/ocr/privacy/hipaa/understanding/srsummary.html>

The Health Insurance Portability and Accountability Act (HIPAA) defines regulations and rules to protect the personal information and processes, and the platform or hardware that is used. For instance, one of the HIPAA requirements is that cloud providers cannot have access to patients’ protected health information (PHI). Instead, the cloud provider must secure, build, maintain, and monitor the infrastructure on which the sensitive information resides as data in motion or at rest.

Another stipulation for HIPAA compliance is the obligation for federal applications to use only American citizens work with data from American citizens or have access to related log files.

Although the PureApplication family of products did not undergo a formal HIPAA compliancy test, it is safe to state the following information:

- ▶ For PureApplication System, the isolation that is built into the product for the storage and network parts provides sufficient separation to withstand a HIPAA compliance audit for the platform.
- ▶ For PureApplication Software, considering that the hardware is not as tightly integrated with the middleware management and the applications that are running on top of it, the wanted level of separation is sufficient to comply with HIPAA guidelines. Whether the cloud that is built with this product is entirely compliant fully depends on the way that the hardware, including the network, is managed, too.
- ▶ For PureApplication Service, on top of the characteristics from PureApplication Software, both the internal IBM provisioning processes and the IBM staff also affect the HIPAA compliance. When you purchase the PureApplication Service offering, you can select the location for your environment. Whether a US-based data center is used is entirely your choice. When the service is running, it has the same characteristics as its on-premises equivalents.

In summary, if you want your hybrid cloud to comply with any standard, you need to check the various components of your solution and check for compliancy for each component separately. For more information and assistance to achieve, certify, and maintain HIPAA compliance for your IBM cloud environment, contact your IBM sales team.

Monitoring

Monitoring is an integrated part for all of the IBM PureApplication on-premises offerings. Every enterprise requires hybrid cloud monitoring to be part of a true hybrid cloud solution.

Monitoring is a shared service that is available to all IBM PureApplication cloud environments. If you combine a PureApplication System with a PureApplication Service environment to build your hybrid cloud, two scenarios are possible:

- ▶ Scenario one: You already use a monitoring product that is running outside of the PureApplication environment and you want to use it.
- ▶ Scenario two: You do not have a monitoring solution, or you want to use the PureApplication monitoring integrated shared service to monitor your combined on-premises and off-premises cloud components.

For scenario one, you need to configure both environments to relay to the outside monitoring environment by using a standard, such as Simple Network Management Protocol (SNMP), or you can rely on an Event Integration Facility (EIF) event forwarding and capturing capability.

Figure 4-63 shows the EIF option that is named IBM Netcool/OMNibus (part of the IBM Tivoli® Management Framework), or any SNMP-capable or EIF-capable product.

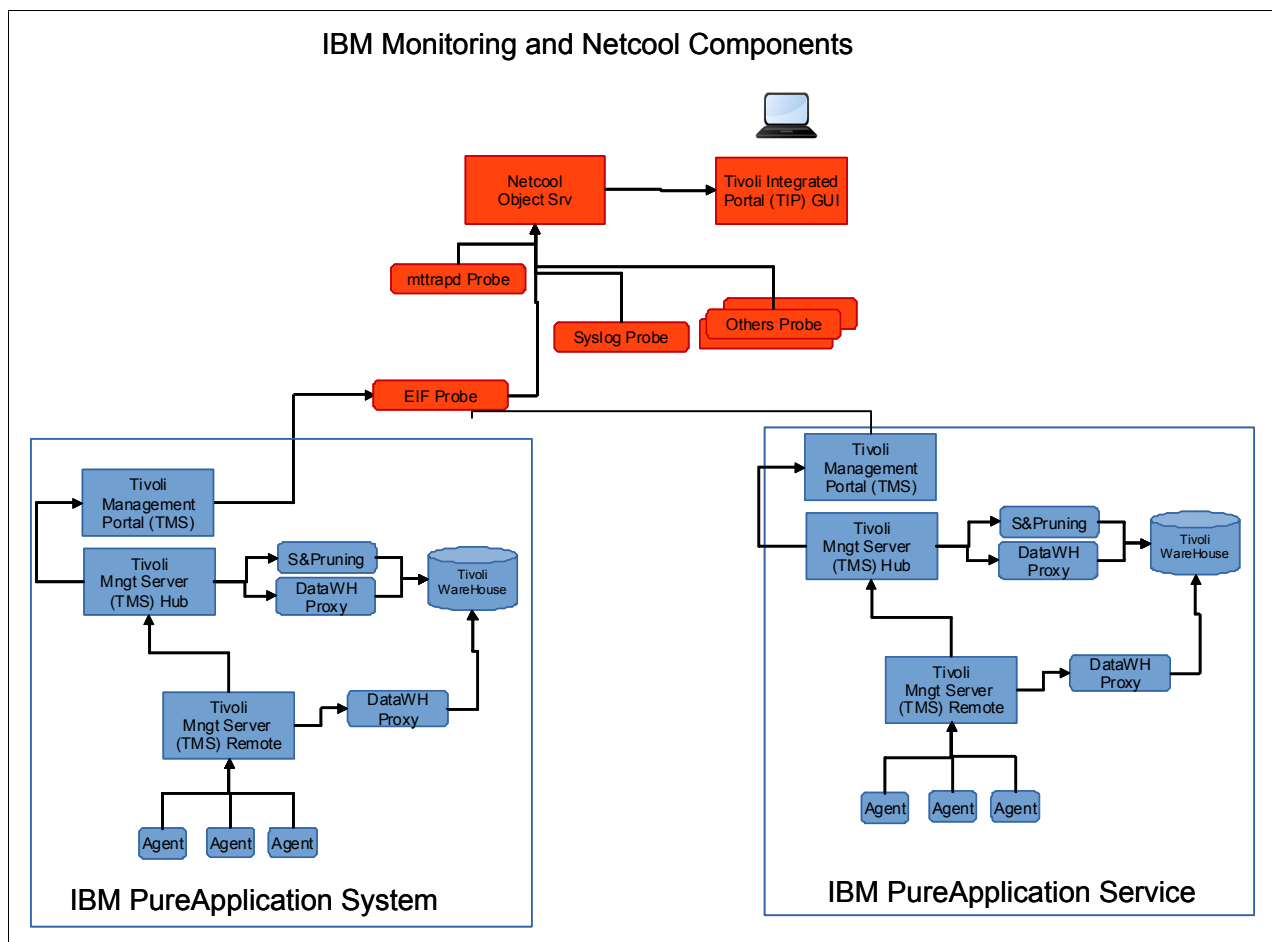


Figure 4-63 EIF or SNMP forwarding sample architecture that uses Netcool/OMNibus

By using these standard protocols, a PureApplication event can be sent to another monitoring tool, which can interpret and understand the event, then suggest potential next steps.

For more information about SNMP, see *SNMP traps*:

<https://ibm.biz/BdHhut>

For EIF, you can import the IBM PureApplication Management Information Base (MIB) files into an EIF event receiver-capable product, such as NetCool/OMNibus or one of several non-IBM products that support the same standards.

For more information, see the following websites:

- *IBM Tivoli Netcool/OMNibus*:

<https://ibm.biz/BdHhuk>

- *PureApplication integration with existent monitoring*:

<http://expertintegratedsystemsblog.com/2012/10/pureapplication-integration-with-existent-monitoring/>

- *Tivoli Event Integration Facility*:

<https://ibm.biz/BdHhu6>

- *Configuring event forwarding:*

<https://ibm.biz/BdHhuU>

Scenario two does not rely on an existing product outside of the IBM PureApplication instances. This scenario requires reconfiguration of the shared monitoring services. Figure 4-64 illustrates this setup.

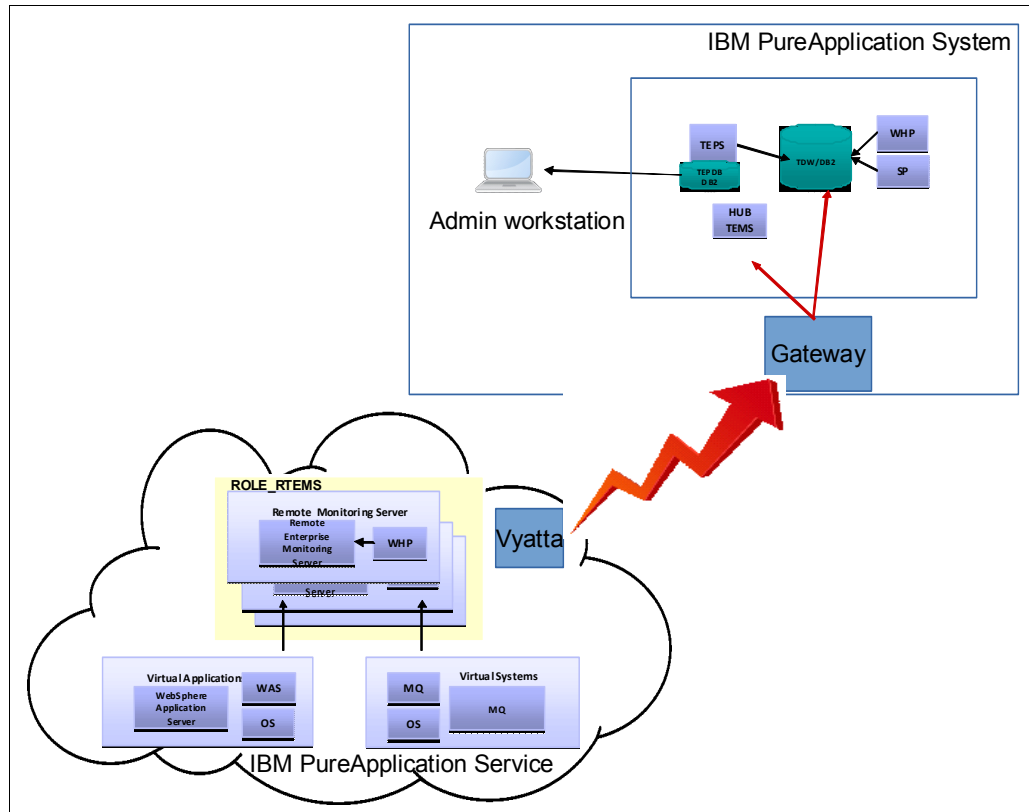


Figure 4-64 Linking PureApplication Service monitoring to the PureApplication System shared service

Consider the following points:

- Ideally, both environments are at the same PureApplication version level.
- Ideally, both environments rely on the same monitoring software level.
- For the license, no restriction exists to monitor applications that are running on any PureApplication infrastructure.
- From an IBM Support perspective, both solutions are supported so this solution is a combination of both services.
- Technically, you can integrate the off-premises Remote Tivoli Enterprise Monitoring Server (ITM-Remote-TEMS) into the Hub-Tivoli Enterprise Monitoring Server (ITM-Hub_TEMS) that is running on premises with two caveats or conditions:
 - From a security perspective, the monitoring portal allows only incoming communication on PureApplication System from VMs or instances within its own Cloud Group. You need to modify the built-in configuration for IPtables to change this security feature and open it for ports 1918 and 1920.
 - Usually, a lack of a common DNS exists between PureApplication System, which uses the company's internal DNS, and PureApplication Service, which cannot query the same internal DNS because it is outside of the company's internal network domain.

Figure 4-65 shows how to set up the monitoring shared service on the PureApplication Service instance.

IBM PureApplication Service

Deploy Pattern: [System Monitoring](#)

Configure

Quick Deploy

Prepare to Deploy

*Name

System Monitoring

*Environment Profile

PSM3EnvProfile1

*Cloud Group

PSM3CloudGroup1

*IP Group

119.81.108.32

*Operating system family

Linux

Base image

IBM OS Image for Red Hat Linux Sy:

SSH key

JL+o+Fu6FXDCP5i17g63xcIfU2DV
8p37CKdFNY1TtV6qMw46FjaYY06/m
LfipYIbJj13x6hG0+Ym9e1tw==
Redbooks

Please generate SSH key pair or provide your public SSH key to access Virtual Machine in future

Generate

Download

Download (PKCS1 format)

Nodes

sharedservice

External Service

* ITMSS Version

* Hub TEMS Address

* Network Protocol

* Hub TEMS Port

* TEPS Address

* TEPS Port

* User ID of Hub TEMS

* Password for the user of Hu

* Password for Data Wareh

JDBC JARs

* Shared Service Sizing

☒ Enable Monitoring Agent

Rack Selected: SNG01-112 (SERV/psm3-sng01-dev)

© Copyright IBM Corporation

Figure 4-65 Setting up the monitoring shared service on the PureApplication Service instance

To cope with the first caveat (page 213), use a program, such as `ssh`, to log in to the monitoring portal VM (the machine that is running the ITM-Hub-TEMS), and change the setting for the IPtables to allow in-bound communications on port 1918, coming from sources outside of the Cloud Group into which the shared service was implemented.

You can address the second caveat (page 213) in multiple ways:

- ▶ Add an Address Resolution Protocol (ARP) entry for the DNS server that is in SoftLayer to the ITM-Hub-TEMS network configuration. This task requires a configuration change on your gateway device or service, in addition to the OS for the ITM-Hub-TEMS.
- ▶ Populate the corporate DNS server with entries of the SoftLayer IP addresses. This solution is a permanent fix. However, this solution requires lead time to set up. You might experience issues with conflicting submits between SoftLayer and your network that make this option impossible.
- ▶ Set up a non-authoritative DNS server for the PureApplication System only. You can then add the SoftLayer IP addresses to the local DNS server. For production environments, you need to add DNS redundant servers. You will need lead time to set up the non-authoritative DNS servers.
- ▶ Add the SoftLayer IP addresses to the `/etc/hosts` file of the ITM-Hub-TEMS server. This fix is temporary. It is not a permanent solution. If you delete the monitoring service and redeploy the service, you must update the `/etc/hosts` file with the newly assigned host names and IP addresses. In addition, you must open ports on the potables firewall, too. T These steps are not tasks that you perform frequently.

To validate that the monitoring setup is working, open the monitoring portal on PureApplication System. The Physical Navigator shows you the external ITM-Remote-TEMS instances and reports back about the events that are happening. Figure 4-66 shows a series of failed logins that are reported.

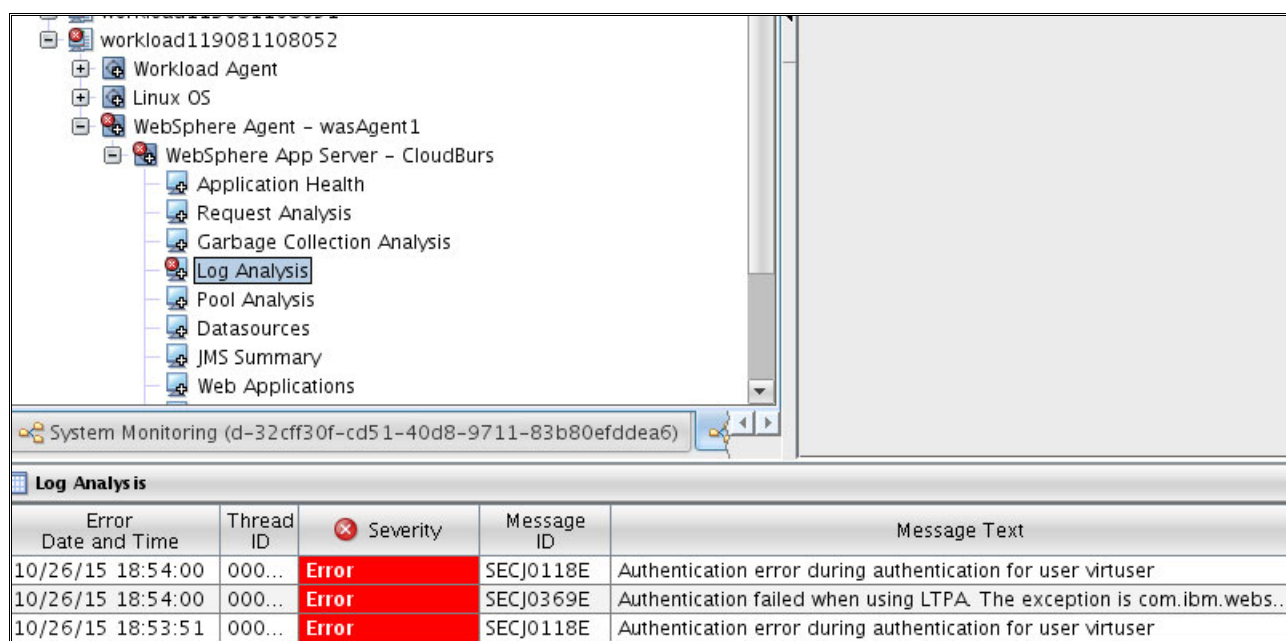


Figure 4-66 IBM Tivoli Monitoring reporting on failed logins for an application that is running on PureApplication Service

When you deploy a pattern, one or more VMs are created. These machines automatically perform these tasks:

- ▶ Pick up the presence of the shared monitoring service
- ▶ Register itself to the shared monitoring service
- ▶ Report the progress on the deployment of the instances

Figure 4-67 displays part of the PureApplication Monitoring portal on the right. On the left, the PureApplication Service GUI shows a new instance in a launching state.

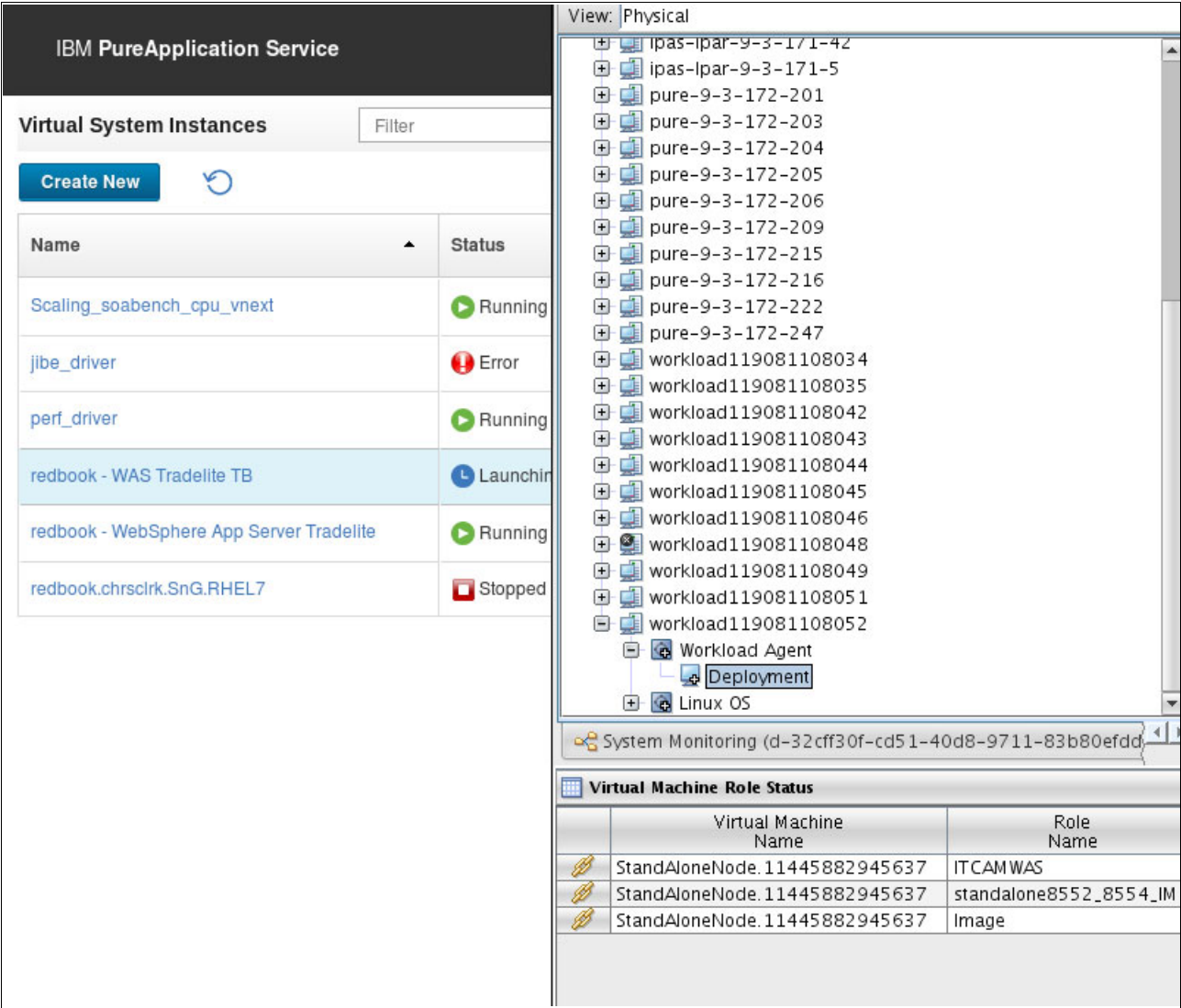


Figure 4-67 Tracking VMs that are launching

For more information about monitoring and other integration points, see the chapter about this topic in *Integrating IBM PureApplication System into an Existing Data Center*, SG24-8285: <http://www.redbooks.ibm.com/abstracts/sg248285.html>

OS-level maintenance

In *Integrating IBM PureApplication System into an Existing Data Center*, SG24-8285, a chapter explains the installation and configuration steps to start the Red Hat Satellite Server shared service. For your hybrid cloud solution, consider setting up the shared service on both the PureApplication System side and the PureApplication Service side. Set up the instance of the shared service on the PureApplication Service to rely on the Red Hat Satellite Server that is running within the company, behind the gateway that links PureApplication System to PureApplication Service. Or, if the capacity is sufficient in the IBM PureApplication Service environment, consider deploying a Red Hat Satellite Server VM in the IBM PureApplication Service environment, also.

Figure 4-68 shows PureApplication System and PureApplication Service relying on the same Red Hat Satellite Server instance.

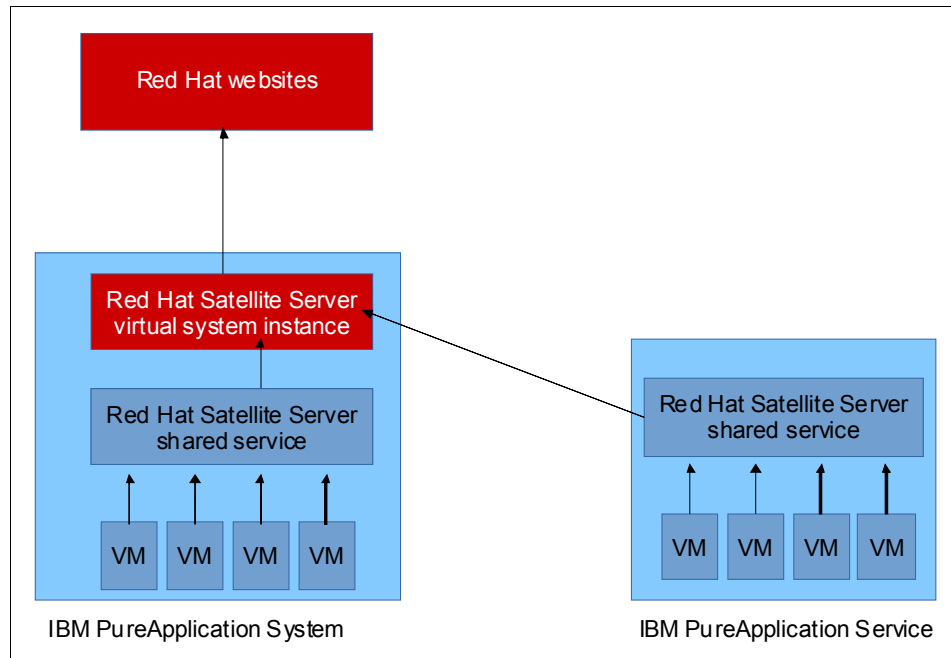


Figure 4-68 PureApplication System/PureApplication Service on one Red Hat Satellite Server instance

The first scenario, as shown in Figure 4-68, reduces the impact on the storage and compute requirements for PureApplication Service. This scenario puts more load in the network between the on-premises environment and the off-premises environment.

The second scenario, as shown in Figure 4-69, requires less network interaction between PureApplication System and PureApplication Service, but it needs additional storage and compute capacities in the PureApplication Service environment.

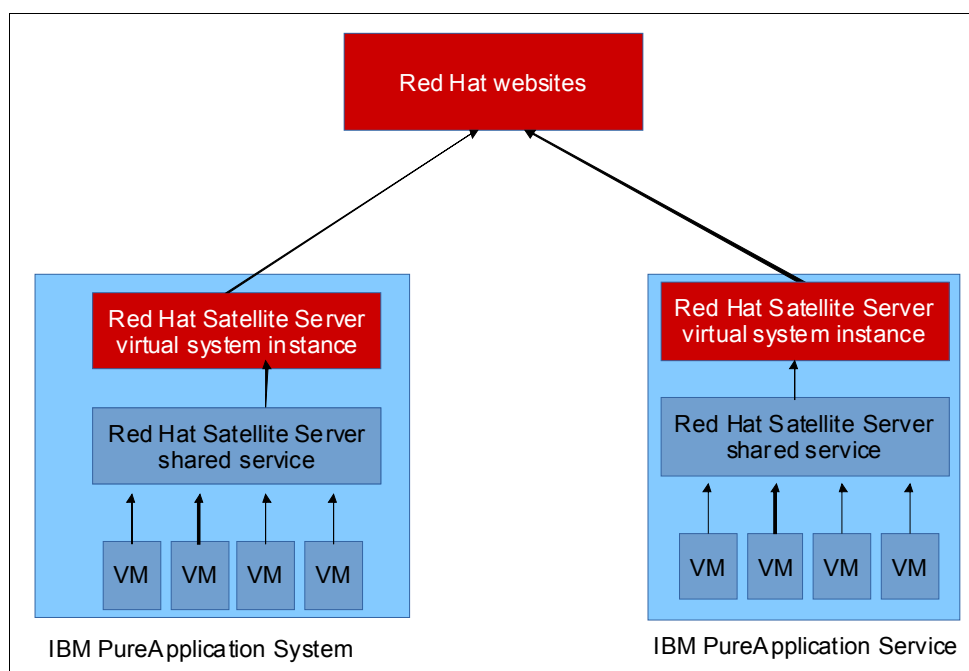


Figure 4-69 PureApplication System and PureApplication Service with Red Hat Satellite Server VMs

If you use IBM Endpoint Manager, consider a similar setup to Figure 4-69.

4.6.3 Using PureApplication Software on top of other non-IBM clouds

This book primarily focuses on building a hybrid cloud by using PureApplication System and PureApplication Service. You can achieve similar results by using, for example, IBM PureApplication Software for Azure or IBM PureApplication Software on VersaStack. This section highlights alternative cloud solutions that you can use in combination with the IBM PureApplication platform:

- ▶ Azure (Microsoft)
- ▶ VersaStack (Cisco)

IBM PureApplication Software on Azure

The IBM preferred route for running IBM middleware on a Microsoft cloud is to use IBM PureApplication Software on Azure. For more information, see these resources:

- ▶ *IBM and Microsoft to offer greater choice in the hybrid cloud:*
<https://news.microsoft.com/2014/10/22/msibmpr/>
- ▶ *IBM PureApplication Software for Azure documentation:*
<https://ibm.biz/BdHhu5>
- ▶ *PureApplication Software on Azure Version 2.1 Download Document:*
<https://ibm.biz/BdHhL4>
- ▶ *PureApplication family of products comparison:*
<https://ibm.biz/BdHhLi>

The two greatest differences between PureApplication Software on Azure and the other IBM PureApplication family products relate to the cloud environment. The PureApplication Software on Azure cloud relies on one public host name or one public IP address. Multiple VMs are all identified through a single host name while they map different ports for different VMs. This approach differs from the other IBM approaches.

The other difference is that PureApplication Software on Azure deploys only VMs that are running Microsoft Windows. You can choose among these products:

- ▶ Windows 2008 R2 Data Center
- ▶ Windows 2012 Data Center
- ▶ Windows 2012 R2 Data Center

You must consider certain characteristics when you use PureApplication Software on Azure because of their impact on specific patterns or IBM middleware products.

Figure 4-70 illustrates the architecture for PureApplication Software on Azure.

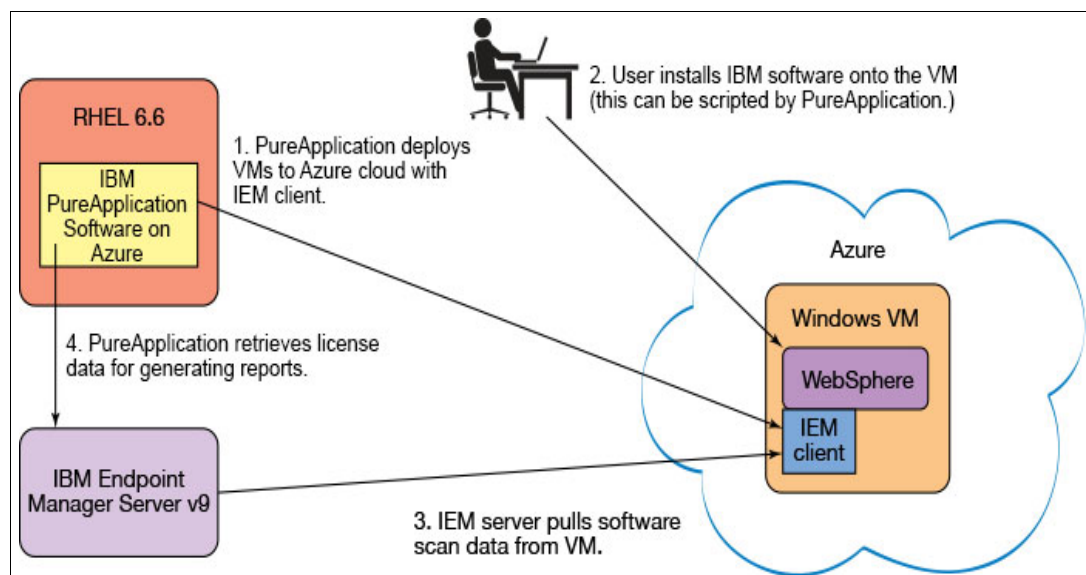


Figure 4-70 PureApplication Software on Azure flow

In the architecture in Figure 4-70, you as the client must provide a Red Hat Enterprise Linux 6.6 instance on which the PureApplication Software part is installed. The instance also acts as a repository for the default data set, which includes only the Pattern Foundation pattern type. Additionally, you must run an Endpoint Manager (IEM) server for which an entitlement was received when you purchased the PureApplication Software on Azure service. Installing the IEM server is a prerequisite because this installation is the only supported way to track licenses and software usage to run IBM middleware on an Azure cloud.

Just as for PureApplication Service, connecting to the Azure cloud also requires a site-to-site VPN connection as described in *Virtual Network documentation*:

<https://azure.microsoft.com/en-us/documentation/services/virtual-network/>

Problem determination is a combined effort. Use IBM for IEM, the pattern creation, and deployment, and use Microsoft for the OS-level product support. For all Internet Control Message Protocol (ICMP) traffic that is blocked on Azure, use the Microsoft Azure web portal to discuss networking issues with Microsoft.

For more information, see the following websites:

- ▶ *Microsoft Azure: The cloud for modern business:*
<http://azure.microsoft.com/>
- ▶ *IBM PureApplication Software for Azure documentation:*
<https://ibm.biz/BdHhu5>
- ▶ *Enabling license scanning with IBM Endpoint Manager and IBM PureApplication Software on Microsoft Azure:*
http://www.ibm.com/developerworks/websphere/library/techarticles/1504_hall/1504_hall.html
- ▶ *Using IBM PureApplication script packages and Microsoft Azure storage to install and configure software:*
http://www.ibm.com/developerworks/websphere/library/techarticles/1504_gower/1504_gower.html?ca=drs-

IBM PureApplication environment on VersaStack

A second non-IBM cloud on which to run an IBM PureApplication environment is the Cisco VersaStack infrastructure or any Cisco Unified Computing System (UCS).

For more information, see the following websites:

- ▶ *5 Things to Know about PureApp Software on VersaStack:*
https://www.ibm.com/developerworks/community/blogs/5things/entry/5_Things_to_Know_about_PureApp_Software_on_VersaStack?lang=en
- ▶ *IBM PureApplication Software Version 2.1 on VersaStack: Designing and Implementing PureApplication Software on VersaStack, REDP-5258:*
<http://www.redbooks.ibm.com/abstracts/redp5258.html?Open>
- ▶ *VersaStack Solution:*
<http://www.cisco.com/go/versastack>
- ▶ *VersaStack:*
<http://www.ibm.com/versastack>

Figure 4-71 shows the required architecture to run an IBM PureApplication Software environment on a VeraStack solution.

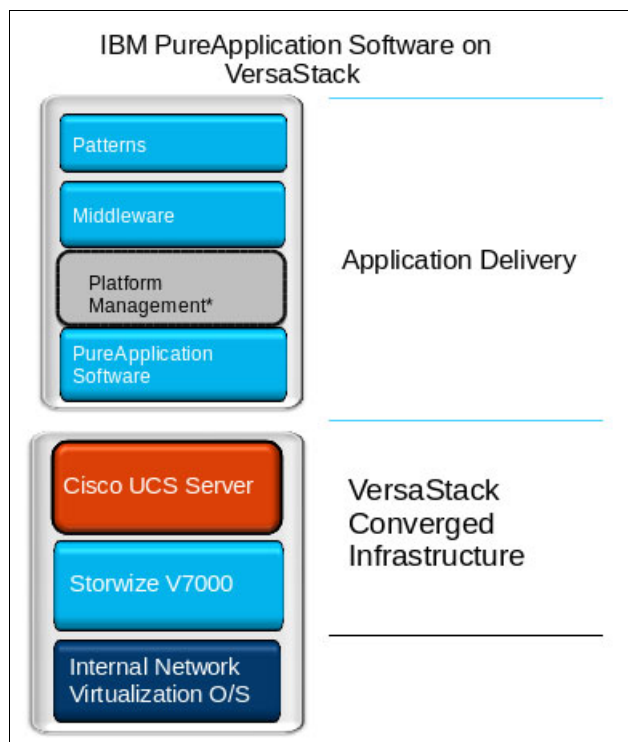


Figure 4-71 PureApplication Software on VeraStack

VeraStack is an integrated Cisco infrastructure solution on top of which PureApplication Software is deployed. It is a viable alternative for clients that standardized on Cisco hardware if they want the benefits of an IBM PureApplication solution in a tightly integrated system that allows deployments of IBM and non-IBM middleware by using either a virtual system or virtual application pattern. This solution can also include an IBM FlashSystem™ V9000 to achieve a higher level of data center power and efficiency. Your IT team is still responsible for the support, testing, and maintenance of the hardware, firmware, and virtualization layer. The following requirements exist for that virtualization layer:

- ▶ Management requires VMware vCenter v5.1 or 5.5 on RHEL 6.6 (minimal 8-core and 64 GB of memory).
- ▶ To run the workloads or applications, you can rely on any hardware that is supported by the vCenter version that is used.

YouTube has several videos about how to set up the integration and collaboration of IBM and Cisco products.

4.6.4 How to cope with an expected or returning increase in load

The cloud implies elasticity. How do you build a hybrid cloud that allows or supports elasticity? This section covers scenarios for increasing your hybrid cloud, or parts of it.

How do you grow the cloud build to rely on an IBM PureApplication System? Purchase additional compute nodes, in pairs, if you need more compute power. And, if you need more storage, you can expand the internal storage by attaching the IBM PureApplication System to externally available storage by using an IBM SAN Volume Controller (SVC).

For more information, see *Planning to use external storage*:

<https://ibm.biz/BdHhLj>

How do you increase the cloud environment capacity of PureApplication Software for IBM PureApplication on VersaStack or Azure? The three IBM PureApplication Software offerings rely on hardware that is managed externally. Increasing the capacity of the VMware ESX infrastructure underneath is the only solution. This solution applies to increases in both storage and compute capacity.

How do you increase the cloud environment capacity of PureApplication Service? Because the infrastructure is based on, derived from, provided by, and managed by the IBM SoftLayer team, the process for increasing the compute capacity or the storage capacity is accomplished differently. IBM provides you with a single interface to manage all of your PureApplication Service instances, which is called the *self-service portal*. This portal acts as a one-stop point of contact for all matters that relate to the PureApplication Service environment. The following tasks are executed or requested through the portal:

- ▶ Uploading or adding patterns from the IBM catalog
- ▶ Adding resources to a PureApplication Service instance
- ▶ Removing resources from a PureApplication instance
- ▶ Accessing product documentation
- ▶ Tracking usage metering and billing
- ▶ Contacting IBM Support if an issue arises

This self-service portal is the preferred tool for requesting additional capacity on demand to cope with anticipated temporary spikes in your application load that runs on PureApplication Service. See Figure 4-72.

The screenshot displays the IBM PureApplication Service Portal Configurator. The top navigation bar includes 'IBM PureApplication Service Portal', 'My Services', 'Catalog', and 'Configurator'. The main section is titled 'Configurator' and 'Draft Service Instances (1)'. It features three tabs: 'Add Service Instance', 'Add Patterns', and 'Optional'. The configuration is divided into three columns:

- Service Instance Name:** Includes a text input field with 'Demo Instance' and a dropdown menu showing 'SJC01 - San Jose - West Coas'.
- Server and Performance:** Includes a dropdown for 'All Cloud Groups (1)' and a table of components.
- Security and Optional Services:** Includes a list of services: 'Denial of Service Protection', 'Data at Rest Encryption', 'Enterprise Connectivity', and 'Firewall'.

The table in the 'Server and Performance' section is as follows:

COMPONENTS	TOTAL
CPU CORES	16 Cores
MEMORY	256 GB
UPLINK PORT SPEED	2 Gbps
STORAGE	1 TB
ADDITIONAL STORAGE	1 TB
BLOCK STORAGE	0
NETWORK	0

At the bottom, there are four buttons: a trash icon, an envelope icon, 'Save as draft', and 'Buy'.

Figure 4-72 The self-service portal to create a draft instance to increase capacity on IBM PureApplication Service

You can request an increase in compute capacity for a short project, for example, by deploying additional resources (4, 8, or 16 compute nodes). You will be billed monthly, as depicted in Figure 4-73. When the IBM DevOps team receives the request for additional capacity for the data center in which your current PureApplication Service is running, provisioning in the SoftLayer data center starts. It takes 2 - 3 business days to provision the new hardware and link it to your self-service portal and the existing PureApplication Service instance. The ordering process is identical to the existing PureApplication Service ordering process. You will receive an email when the requested infrastructure is available.

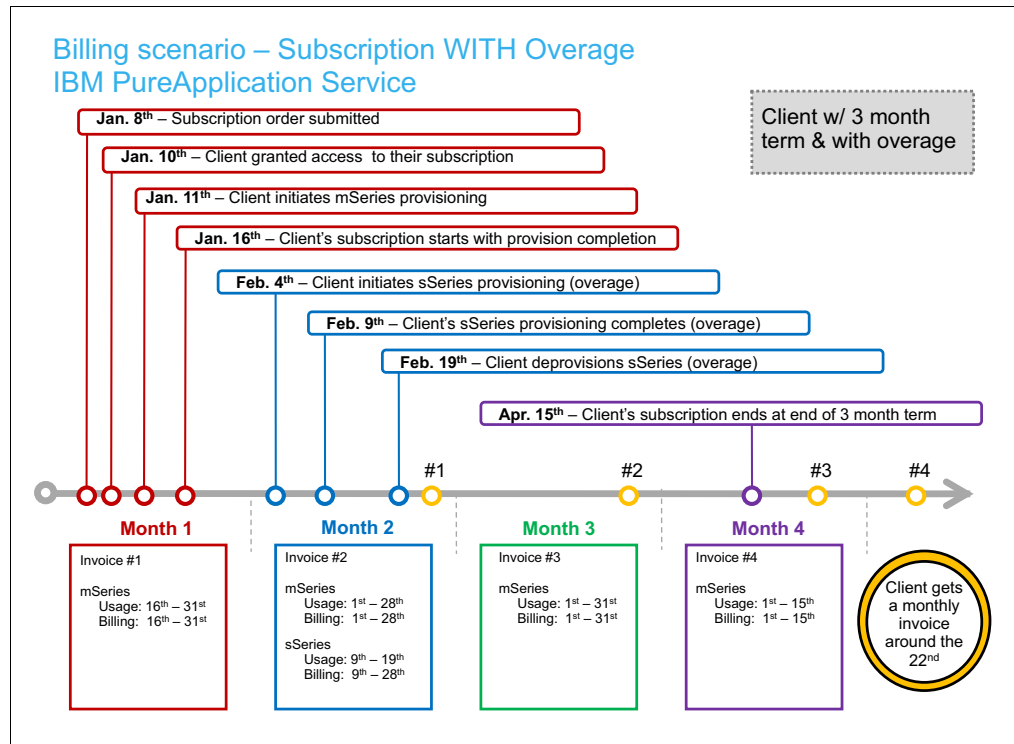


Figure 4-73 For clients with the Overage subscription with IBM PureApplication Service

Figure 4-74 lists what you as the client need to purchase to benefit from the Overage subscription with PureApplication Service.

Billing scenario – Subscription WITH Overage IBM PureApplication Service		
Scenario: Client has signed up 3 months subscription for 1 x 8 core Client does intend to spin up 1 x 16 core for few days for a competitive project		
SQO Order details:		
PN	Description	Quantity
D1HANLL	IBM PureApplication Service Subscription PER MONTH	7790
D1HAPLL	IBM PureApplication Service Subscription OVERAGE	1
TBD	IBM PureApplication Service Subscription Support PER MONTH	1
D1B5NLL	IBM PureApplication Service SLA (Service Level Agreement)	1
D1H9ILL	IBM PureApplication Service C500- 8-128 (8 core 128 GB) Platform and Infrastructure	NA
D1H9HLL	IBM PureApplication Service C500- 16-256 (16 core 256GB) Platform and Infrastructure	NA

Figure 4-74 Sample set of part numbers for a client to benefit from the Overage subscription

As of this writing, we identified only two restrictions for building an off-premises cloud by using PureApplication Service:

- ▶ The capacity of one data center (a limitation only if you rely on one physical location)
- ▶ The available budget

4.6.5 Checklist

In general, checklists are used to provide guidance. Based on our experience, we compiled a checklist. The following checklist relates to building a hybrid cloud by using PureApplication System and PureApplication Service:

- ▶ Access to a system with available capacity.
- ▶ Access to a service instance with available capacity.
- ▶ A workstation with all of the administrative tools:
 - CLI
 - Image Construction and Composition Tool (ICCT)
 - A browser
- ▶ A gateway that is configured between both environments (back and forth).
- ▶ For IBM PureApplication Service, check with IBM Support or use the System Support Program.

- ▶ For IBM PureApplication System, check with your IT department that manages the firewall and other security devices.
- ▶ List of ports to use:
 - Application
 - Database
 - Web server
 - Monitoring
 - Port 80
- ▶ Gateway that is configured to route traffic for a specific IP group or range.
- ▶ DNS or changes to the (local, on service) hosts file to include gateways and hosts.
- ▶ Ensure that at least one Cloud Group with sufficient IP addresses is available on both sides.
- ▶ Monitoring shared services are deployed on PureApplication System.
- ▶ Remote monitoring shared service that is deployed on PureApplication Service requires the following information:
 - User names.
 - Passwords.
 - Secure Shell key (ssh-key).
 - IP addresses or host names from the on-premises shared services, Hub-TEMS.
 - The application.
 - Patternized.
 - All pattern parts are identified and accounted for.
 - Transfer of those parts from the source environment to the target environment.



Summary

This book introduced various areas of hybrid cloud in general and a PureApplication hybrid cloud in particular. The success of creating and using a hybrid cloud efficiently depends not only on technical matters, but on other areas, such as economic value, political concerns, and cultural concerns. The following topics are described in this chapter:

- ▶ The meaning of the “cloud” and the “hybrid cloud”
- ▶ The business value of hybrid clouds
- ▶ Various cloud or cloud-related technologies
- ▶ More on business value: Considering the long run
- ▶ Cloud and enterprise qualities of service
- ▶ Implementing a hybrid cloud
- ▶ Paradigm shifts: The hype and the reality
- ▶ Reiterating the importance of planning and a BVA
- ▶ The true promise of the hybrid cloud

5.1 The meaning of the “cloud” and the “hybrid cloud”

In this book, we clarified the notion of “cloud.” IBM generally follows the National Institute of Standards and Technology (NIST) definition, which includes these conditions:

- ▶ Pooled resources
- ▶ Self-provisioning
- ▶ Metered usage (implying pay for what you use)
- ▶ Elasticity (of resources)
- ▶ Broad access (especially through the internet)

Hybrid cloud is a deployment model that consists of two or more clouds (private, public, or community) that remain unique entities, but that can interoperate by using standard or proprietary protocols. In this book, we assumed that the two clouds are different offerings from the PureApplication family: PureApplication System (on premises) and PureApplication Service (off premises).

5.2 The business value of hybrid clouds

We discussed the business value of clouds, and especially hybrid clouds, and suggested that it can be as varying as the business value of:

- ▶ Leasing a car
- ▶ Owning a car
- ▶ Both owning a car and sometimes renting a car, too, which is the analog of a hybrid cloud

Off-premises clouds are typically called public and often simply “cloud,” perhaps because the first cloud was a public one. They can be shared or not (with dedicated hardware), all of which is owned by the off-premises vendor and paid for on a rent or lease model, with dedicated hardware probably being more expensive. In the shared model, different enterprises are given virtual machines on common hardware.

An on-premises cloud, which is often called a *private cloud*, is owned by the enterprises themselves, in their own data centers, which gives them the same advantage as public clouds. So, for example, different lines of business (LOBs) can self-provision their infrastructure as a service (IaaS) or their platform as a service (PaaS) and be billed correctly, if a chargeback mode is wanted.

A mixed case also exists (but not hybrid in the sense that we use the term). If the cloud is hosted in another company’s (public) data center, but owned by the enterprise, it is a “*hosted private cloud*.” In this case, part of the total cost is a rental charge even though the enterprise owns part of the resources.

The following business drivers are benefits for a cloud or hybrid cloud (Figure 5-1):

- ▶ Improved time to market, especially with self-provisioning
- ▶ More automation
- ▶ Repeatability and standards (of the enterprise, not only of national or IT standards)
- ▶ Easier management

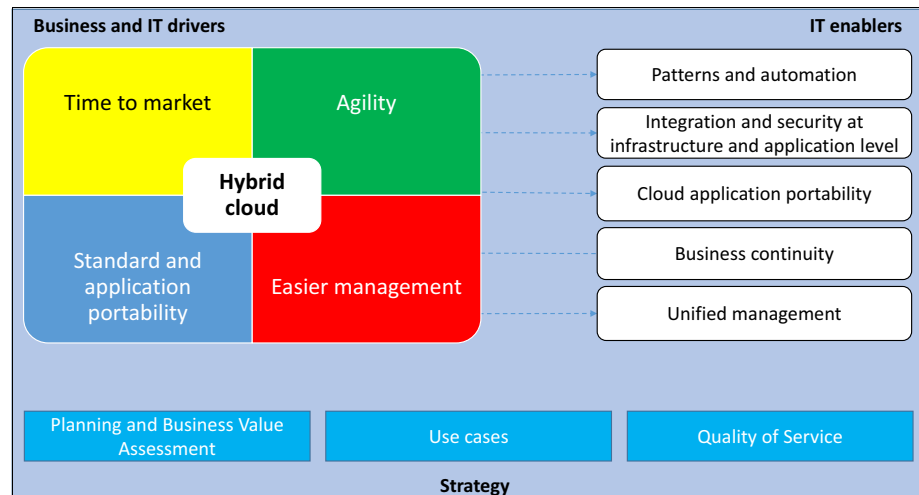


Figure 5-1 Business drivers, IT enablers, and strategy aspects

These business drivers all lead to reduced risk and to reduced costs.

5.3 Various cloud or cloud-related technologies

From the perspective of explaining the whole “sky”, that is, the entire IT environment in which a cloud or hybrid cloud exists, we introduced support by IBM and the PureApplication family for open source technologies, such as Chef and Docker. We also introduced the concept of “orchestration” in this context (which might allow self-provisioners to deploy to any number of disparate clouds, and not only to, PureApplication family).

IBM Cloud Orchestrator, for instance, gives a single, easy-to-use portal to the whole enterprise, with deployment rules and privileges that are hidden from the users. During self-provisioning, the users are automatically given only the resources (operating system (OS) at the minimum, but maybe middleware or applications, with CPU, memory, and storage) that they are entitled to, in the particular cloud they are entitled to (whether public or private).

We also differentiated between the IBM cloud solution that is called Bluemix, which is used for cloud-native applications (applications that are born and run entirely on the cloud, perhaps connected to back-end systems or databases), and PureApplication offerings, which are designed primarily (but not exclusively) to cloud-enable the rigorous systems or processes that enterprises spent years of effort perfecting and protecting.

5.4 More on business value: Considering the long run

Just as no simple answer exists to the question for all families, “Do I own a car, lease a car, or take taxis on occasion?”, no simple answer exists to the question, “Does my enterprise adopt a private cloud, a public cloud, or the most important question here, a hybrid cloud?” We offered six common use cases where a hybrid cloud might provide technological and economic advantage.

To determine the outcome, an enterprise needs to perform its due diligence, consider as many factors as is reasonable and, in our opinion, carry out a Business Value Assessment (usually done with the assistance of IBM sales teams and the Business Value Assessment (BVA) specialists) to determine which option is most suitable and cost-effective in the end.

One of the strengths of the IBM “patterns” is that they are portable between the various clouds. We showed how to port a pattern by using the Export and Import functions that are natively built into PureApplication offerings to move between PureApplication System (on premises) and PureApplication Service (off premises).

5.5 Cloud and enterprise qualities of service

While we are on the point about qualities of service (QoS), we think that it is little recognized that IBM went slowly and surely to provide the exhaustive QoS that enterprises need, if they are to move to hybrid clouds. IBM worked this way by drawing on its knowledge from having a quasi-cloud model with the mainframe for over 40 years. That is, the mainframe gives four of the five conditions of the NIST cloud, except for the broad network access, even though the mainframe can be fronted nowadays by web technologies and so might arguably also be called a private cloud.

In short, choices must focus on building a solution that delivers what is required initially while offering the flexibility to expand with an evolving cloud environment, all the time while solving the full range of issues that an enterprise must cope with, not simply allowing the quick self-provisioning of environments. IBM with its vast experience in enterprise technologies is now positioned well to offer ideal hybrid solutions that are particularly tailored for individual clients.

5.6 Implementing a hybrid cloud

With the sample six use cases, we provided the initial rationale for these fairly obvious use cases. Chapter 4, “Implementation considerations” on page 139, showed the implementation that covered the following various issues:

- ▶ Portability
- ▶ Monitoring across the hybrid cloud from a single “pane-of-glass” perspective
- ▶ Backup of patterns and management data, such as access control lists (ACLs)
- ▶ Backup of virtual machines
- ▶ Database replication
- ▶ Networking issues

All of this information gives you a much better sense of the kinds of issues that you must resolve if you take the journey into hybrid clouds. It also gives the technical issues to consider while you plan for a hybrid cloud and compare vendors' solutions. Not all potential pain points were necessarily covered for every possible client. The design of PureApplication System is arguably a paradigm shift in computing, to use a concept from science.

We did not describe in detail how duplicating all of the components, the switches, virtual storage, compute nodes, virtualization managers, and pattern engines, with writing the code that integrates them all together, allows IBM to patch the entire system with a couple of clicks without downtime. The following tasks summarize this feature:

- ▶ A passive component is upgraded, then the relevant workload is moved to it.
- ▶ The active component now becomes passive and gets patched.
- ▶ The system round-robins likewise through the whole rack so that the entire system is completely patched in about 12 hours, saving hundreds of manual hours each year.

5.7 Paradigm shifts: The hype and the reality

As most IT professionals who were in IT for decades know, paradigm shifts are not new. The possible, if narrowly restricted, advantages of distributed servers and client/server architectures became known by the 1990s, and similarly with the introduction of XML, itself starting in the early 2000s.

To use the introduction XML to see several parallels with the new developments in cloud, XML was the hot technology that seemingly was going to solve almost all problems in IT, or at least almost all integration problems, as the hype often went. Given that all computer languages can read and write text, and given that XML was plain text, any system on one hardware or software platform can exchange data with any other system.

Before, different computer languages were not able to directly communicate, unless difficult integration standards, such as Common Object Request Broker Architecture (CORBA), were learned and implemented.

However, the reality was that enterprise adoption of XML, including the typical variations that are called web services and service-oriented architecture (SOA), was slow to get started. Several of the reasons were security holes in XML, performance considerations (what were a few bytes of data before were now hundreds of bytes because of the need to wrap the data, such as a bank balance, in the XML/web services wrapper, such as SOAP) and the like.

Eventually, after many years, the XML technologies, including appliances, such as DataPower, matured and found their place in enterprise shops, but only after significant planning, rethinking processes and traditions, training and implementing new related technologies and (sometimes conflicting) standards and testing.

The cloud in general and the hybrid cloud in particular, with the capabilities, such as elasticity, self-provisioning, and metered usage, promises much in terms of faster time to market, easier management, and less cost. However, arguably the issues will be much the same as, if not more complex than, XML, because that was essentially a software matter.

Cloud, and especially hybrid clouds, now cut across all areas of IT, including applications, especially if you want a PaaS or software as a service (SaaS) environment.

Building your whole infrastructure from scratch, analogous to building your car from scratch, was the accepted way of creating the environments that were needed for middleware and applications. It was the only way enterprises knew since the 1990s until about 2007, when IBM and other companies began to plan and create the integrated cloud offerings (part of which were IaaS and part of which were PaaS).

Why build a car, though, from piece parts when you can buy it whole, or at least mostly constructed with a few options to be added? Not until IaaS integrated systems appeared, though, did clients realize they can buy the car half-assembled with compute nodes, storage, and networking and finish the rest (adding the middleware, application, and monitoring).

With PaaS and SaaS, you get the car almost completely assembled or completely assembled. Although in the latter case, you do not have, in IT terms, options to customize the middleware or application in depth, which might not be acceptable for the typical large enterprise and that is why a PaaS is often the solution that most clients want.

The benefits are self-evident with a premanufactured car, for what consumer nowadays builds its own car from scratch with parts that are bought from an automobile store? It takes little insight to see, analogously, the inherent benefits for IT with integrated cloud solutions.

5.8 Reiterating the importance of planning and a BVA

The point that the adoption of cloud will be probably more complex than, or at least as complex as, the adoption of XML technologies returns us to planning. Whether the public cloud will be more cost-effective than a private cloud in the end, and whether the hybrid cloud will be more cost-effective than either of the two solo options, can be determined with significant certainty only by performing a Business Value Assessment (BVA) and considering all of the factors for total cost of ownership (TCO), as opposed to an often misleading and short-sighted total cost of acquisition.

Enterprises also need to ensure that governmental regulations do not, for instance, forbid you from storing data in the public cloud (depending on where its servers are). In addition, enterprises need to ensure that the company that is running their public cloud has the resources, processes, mechanisms, and insurance to cover the loss of data centers in a disaster or human accident.

Is your data lost if the data center is lost? In addition, how do you force the company to destroy private data, with proof, if you choose to leave their cloud and what if they are reluctant to provide the proof? Because the private data is your responsibility, you must be confident that it will always be handled correctly. The quality of the data center and of the company that is running it, then, becomes paramount.

Therefore, to return to the point that hybrid clouds might be much more complicated than XML, not only IT architects, operations staff, application development leads, security experts, and database administrators need to get involved in ensuring the protection of their traditional areas in the new scenario, but those individuals who are expert in data center design and maintenance need to be consulted.

Evaluating clouds, and especially hybrid clouds, including all of their parts and processes, can be a daunting task for those individuals who are coming into the hybrid cloud for the first time, or even the second time.

5.9 The true promise of the hybrid cloud

Just like XML about 12 years ago, cloud in general and hybrid cloud in particular are beginning to show their great promise. However, this book shows that creating a hybrid cloud with all the necessary QoS for an enterprise is not a trivial matter, and clearly added complexity exists in moving to a cloud. Fortunately, cloud offerings, such as the PureApplication family, facilitate much of the work for you.

We hope the book provided a good introduction into the necessary considerations to ensure the success of any transformation that in part uses, or results in, a hybrid cloud, and that we sketched at least one way in which IBM can help you transform your environment with PureApplication offerings.

Related publications

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

IBM Redbooks

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

- ▶ *Integrating IBM PureApplication System into an Existing Data Center*, SG24-8285
- ▶ *Adopting IBM PureApplication System V1.0*, SG24-8113
- ▶ *Implementing High Availability and Disaster Recovery in IBM PureApplication Systems V2*, SG24-8246
- ▶ *Integrating an IBM PureApplication Environment*, TIPS1328
- ▶ *Microservices from Theory to Practice: Creating Applications in IBM Bluemix Using the Microservices Approach*, SG24-8275
- ▶ *High Availability and Disaster Recovery Options for DB2 for Linux, UNIX, and Windows*, SG24-7363
- ▶ *IBM MobileFirst Strategy Software Approach*, SG24-8191
- ▶ *IBM WebSphere Cast Iron Introduction and Technical Overview*, REDP-4840

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:

- ▶ White paper “Systems of Engagement and the Future of Enterprise IT”:
https://www.google.com/?gws_rd=ssl#q=Systems+of+Engagement+and+the+Future+of+Enterprise
- ▶ Bluemix sources of information:
<http://www.ibm.com/software/ebusiness/jstart/bluemix/>
- ▶ IBM Hybrid cloud demo:
<https://www.youtube.com/watch?v=-s93Xx6wx4>
- ▶ IBM PureApplication family:
http://www.ibm.com/ibm/puresystems/uk/en/pf_pureapplication.html
- ▶ IBM Cloud Orchestrator product positioning page:
ibm.co/1p902ji
- ▶ IBM Cloud Orchestrator Knowledge Center:

https://www.ibm.com/support/knowledgecenter/SS4KMC_2.5.0/com.ibm.ico.doc_2.5/kc_welcome-ico25.html

- ▶ IBM Cloud Orchestrator wiki:

<https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20SmartCloud%20Orchestrator/page/Welcome>

- ▶ IBM Bluemix:

<http://www.ibm.com/cloud-computing/bluemix/index-b.html>

- ▶ IBM Bluemix Local:

<http://www.ibm.com/cloud-computing/bluemix/hybrid/local/>

- ▶ Virtual patterns and Pattern Builder:

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

- ▶ Docker client application programming interface (API):

<https://docs.docker.com/reference/commandline/cli/>

- ▶ Docker on PureApplication System:

http://www.ibm.com/support/knowledgecenter/SSNLXH_2.1.0/doc/iwd/pac_docker_ov.dita

- ▶ PureApplication Software:

https://www.ibm.com/support/knowledgecenter/SSL5ES_2.1.0/doc/iwd/pac_docker_ov.dita

- ▶ PureApplication Service:

https://www.ibm.com/support/knowledgecenter/SSNS6R_2.1.1/doc/topics/pac_docker_ov.dita?lang=en

- ▶ Configure Open Stack services and deploy HOT templates into PureApplication System:

https://www.ibm.com/support/knowledgecenter/SSCRSX_2.1.0/doc/systemconsole/t_configopenstack.dita

- ▶ Configure Lightweight Directory Access Protocol (LDAP) in PureApplication System:

https://www.ibm.com/support/knowledgecenter/SSCRSX_2.1.0/doc/iwd/ldap_integrate_overview.dita

- ▶ Two analyst reports on hybrid cloud:

<http://www.ibm.com/common/ssi/cgi-bin/ssialias?subtype=WH&infotype=SA&htmlfid=MSL03008USEN&attachment=MSL03008USEN.PDF>

http://www.ibm.com/common/ssi/cgi-bin/ssialias?subtype=WH&infotype=SA&appname=STGE_WA_ZA_USEN&htmlfid=WAW12378USEN&attachment=WAW12378USEN.PDF

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

Redbooks

Establishing a Secure Hybrid Cloud with the IBM PureApplication Family

SG24-8284-00

ISBN 0738441252



(0.5" spine)

0.475" <-> 0.873"

250 <-> 459 pages



SG24-8284-00

ISBN 0738441252

Printed in U.S.A.

Get connected

