IBM

IBM PureApplication System Best Practices

- Discover best practices for optimal deployments
- Learn how others use PureApplication System
- Get standard implementation examples to follow

Amit P. Acharya
Tom Bal
Kyle Brown
Srinivas Cheemalapati
Rodrigo Costa
Venkata Gadepalli
Rajeev Gandhi
Erdinc Guler
James Kochuba
Rory O'Grady
Jose Ortiz

Redbooks

ibm.com/redbooks

International Technical Support Organization

IBM PureApplication System Best Practices

February 2014

SG24-8145-00

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

First Edition (February 2014)

This edition applies to IBM PureApplication System Version 1.1.

Contents

Notices v
Trademarks vi

Preface vii
Authors vii
Now you can become a published author, too! x
Comments welcome xi
Stay connected to IBM Redbooks xi

Chapter 1. Introduction 1
1.1 Cloud computing 2
1.1.1 Basic principles of cloud computing 3
1.1.2 Cloud service models 5
1.1.3 Cloud deployment models 6
1.2 IBM PureSystems family of products 9
1.2.1 Patterns of expertise 10
1.2.2 IBM PureFlex System 11
1.2.3 IBM PureApplication System 12
1.2.4 IBM PureData System 13
1.3 IBM PureApplication System overview 13
1.3.1 All cloud stack components are already integrated and optimized 14
1.3.2 Configurations 15
1.4 Concepts and terms 17
1.4.1 Patterns 17
1.4.2 Management 18
1.4.3 Problem determination and support 20

Chapter 2. Challenges for IT organizations 23
2.1 Cloud computing and optimization 24
2.1.1 Changing the dynamics of the enterprise 24
2.1.2 Roles and process 25
2.2 Innovation and optimization 26
2.2.1 Ease of use 27
2.2.2 Availability 28
2.2.3 Promotion between environments 28
2.3 Simplicity and consolidation 29

Chapter 3. Organization and role changes to adapt to automation 33
3.1 Introduction 34
3.1.1 The value of patterns of expertise 34
3.1.2 Major steps in adopting PureApplication System 35
3.2 Organizational principles and practices 35
3.2.1 Overall changes 36
3.2.2 Role changes 37
3.2.3 Skill set changes 38
3.3 Roles for PureApplication System implementation 39
3.3.1 Data Center Operations team 40
3.3.2 Application Operations team 41
3.3.3 Database Operations team 42

3.3.4 Finance and Compliance staff 42
3.3.5 Content Enablement team 43
3.3.6 Solution development 45
3.4 Structure of a PureApplication Center of Excellence 48
3.4.1 Task division for Center of Excellence subteams 48
3.4.2 Moving to round-the-clock operations 49

Chapter 4. Workload practices for patterns 51
4.1 Browse the PureSystems Centre for relevant assets 52
4.2 Use a virtual application pattern to add new applications 52
4.3 Migrate the Java application before you use patterns 53
4.4 Use images and patterns as delivered when possible 53
4.5 Decide which tool to use if you customize virtual images 54
4.6 Split patterns along key interfaces 54
4.7 Separate the configuration and application from topology 55
4.8 Automate your lifecycle steps 56
4.9 Think of environments as short-lived assets 57
4.10 Develop and test script packages on the target platform 57
4.11 Set the time zone 58
4.12 Pass passwords as parameters in script files 58

Chapter 5. High availability of the system 61
5.1 Cloud management 62
5.2 Placement algorithm 62
5.3 Cloud configuration 64

Chapter 6. Lifecycle and maintenance 67
6.1 IBM PureApplication System maintenance 68
6.1.1 System components 68
6.1.2 Automated maintenance frameworks 68
6.1.3 Client responsibilities 69
6.2 System maintenance 69
6.3 Workload maintenance 71
6.3.1 Maintaining and updating catalog components 71
6.3.2 Updating a virtual system workload 73

Related publications 75
IBM Redbooks 75
Online resources 75
Help from IBM 77

Notices

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

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

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

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

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

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

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

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

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

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

COPYRIGHT LICENSE:

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

Trademarks

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

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

AIX®
DB2®
developerWorks®
Global Technology Services®
Guardium®
IBM®
IBM PureData™
InfoSphere®
Lotus®
Lotus Notes®
Notes®
POWER®
POWER7+™
PowerVM®
PureApplication™
PureData™
PureFlex™
PureSystems™
Rational®
Redbooks®
Redbooks (logo) ®
Systems Director VMControl™
Tivoli®
WebSphere®

The following terms are trademarks of other companies:

Netezza, and N logo are trademarks or registered trademarks of IBM International Group B.V., an IBM Company.

Intel, Intel Xeon, 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.

ITIL is a registered trademark, and a registered community trademark of The Minister for the Cabinet Office, and is registered in the U.S. Patent and Trademark Office.

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.

Preface

This IBM® Redbooks® publication describes IBM PureApplication™ System preferred practices that are based on IBM client and Business Partner experience. It explains how PureApplication System enables industries to consolidate workloads, increase efficiency, automate routine processes, reduce costs, and become more agile to respond to continually changing business needs.

This book is particularly useful to solution specialists, system or software architects, and the IT teams who implement PureApplication System cloud services.

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, as well as 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. He has written IBM Redbooks publications about service-oriented architecture (SOA) and has 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 Software Group. Tom joined IBM Belgium through the Lotus® acquisition in early 2000. For several years, he was part of the services team, assisting clients and Business Partners with different Lotus products, including IBM Lotus Notes® and Lotus Workplace. He then became the technical presales representative for IBM WebSphere® Portal. In 2012, after two years as a client IT architect for the telecommunications industry, Tom rejoined Software Group to cover PureApplication System for the region of Europe known as Benelux, which includes Belgium, the Netherlands, and Luxembourg. Tom holds a Master of Arts degree in Translation.

Kyle Brown is a Distinguished Engineer with IBM Software Services for WebSphere. He specializes in emerging technologies and provides consulting services, education, and mentoring on Cloud, SOA, object-oriented topics, and Java Enterprise Edition technologies to Fortune 500 clients. Kyle is a co-author of *Enterprise Java Programming with IBM WebSphere* (Pearson Education, 2011) and *Persistence in the Enterprise: A Guide to Persistence Technologies* (IBM Press, 2008). He is also a frequent conference speaker on the topics of PureApplication System, object-oriented design, enterprise Java, SOA, and design patterns.

Srinivas Cheemalapati is a Senior Engineer and Certified IT Specialist in the IBM Public Sector, US Federal division. He joined IBM US in 1998 and has 25 years of experience that includes cloud computing, IT architecture and solutions, server development, and virtualization. Srinivas leads large, complex proposals and provides infrastructure solutions in the areas of cloud computing, data center consolidation and optimization for public sector and US Department of Defense clients. He contributes to delivery excellence by performing technical risk and delivery assessments. He holds a Master's degree in Computer Science from North Carolina State University, and his interests include cloud solution design, big data, Apache Hadoop software, and data analysis.

Rodrigo Costa is a Senior IT Specialist for IBM Global Technology Services® in Sao Paulo, Brazil. He has more than 12 years of experience in the IT and banking industries. He holds a bachelor's degree in Computer Science, and his main focus is on middleware technologies, where he has dedicated the last eight years to working in administration, infrastructure design, implementation, maintenance, and troubleshooting of the IBM WebSphere environment. Rodrigo is also an IBM Certified System Administrator for WebSphere Application Server.

Venkata Gadepalli is a member of the IBM Software Services for WebSphere team. He has more than 14 years of experience in the IT field and has been involved in client engagements involving the WebSphere family of products. His main area of interest is working with first-of-a-kind engagements that involve IBM PureApplication System. Vishy has written numerous papers that have been published both within and outside of IBM. He also co-authored the first and second editions of the *IBM WebSphere Portal Primer* (IBM Press, 2003 and 2005).

Rajeev Gandhi is a Senior Technical Staff Member in IBM Software Services for WebSphere, specializing in PureApplication Systems. He has many years of experience in the IT field. His current area of expertise is providing PureApplication System consulting, services, and education for IBM clients and internal teams.

Erdinc Guler is a Team Lead at IBM Delivery Centre, Central Eastern Europe, in Brno, Czech Republic. He started his IT career 13 years ago while studying aerospace engineering in Ankara, Turkey. He then since gained experience in different areas, from system administration to client engagements, through his employment with an Internet service provider. Erdinc joined IBM in 2009 as a Middleware IT Specialist and has worked mainly on WebSphere products. He currently leads second-level and third-level teams that support middleware technologies in Europe. He has participated in creation of the Global Solution Architecture Repository catalog and IBM PureApplication System testing.

James Kochuba is a Software Architect for IBM Workload Deployer and PureScale Application System. His previous roles in IBM were as the WebSphere Application Server Security Test Architect, on the WebSphere Premium Support team, and in WebSphere Application Server support. He is the technical leader for shared services (for example, logging, monitoring, and caching services) and services (troubleshooting) for the IBM platform-as-a-service products.

Rory O'Grady is a Technical Pre-Sales Specialist at IBM United Kingdom Ltd. After several years in customer service roles, he moved into the software line of business. Rory has experience in development and testing roles, as well as in quality assurance and information development. Previous specialties included software development, testing, and modeling, as well as data management and development preferred practices.

Jose Ortiz is a Technical Lead for IBM Workload Deployer, IBM PureApplication System, and PureApplication Systems Serviceability. He works on the patterns, deployment, and management components.

Additional contributors

Margaret Ticknor, who led the project to produce this book, is an IBM Redbooks Project Leader in the Raleigh Center, in Raleigh, North Carolina, USA. She primarily leads WebSphere and IBM PureApplication System projects. Before joining the International Technical Support Organization, Margaret worked as an IT specialist in Endicott, NY. She attended the Computer Science program at State University of New York at Binghamton.

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

- Kenneth Ausich
- Thomas Blattmann
- Don Carr
- Jonathan Deberdt
- Lei (Joyce) HL He
- Hong Ji
- Mike Law
- David Leigh
- Michael Maison
- Sara Mitchell
- Matthew Sheard
- Andre Tost
- Hendrik van Run
- Yue (Eva) WY Wang
- Joe Wigglesworth
- Bobby Woolf

Thanks to the following people for supporting this project:

- Judith Broadhurst, IBM Redbooks Editor
- Ella Buslovich, IBM Redbooks Graphics Editor
- Deana Coble, IBM Redbooks Technical Writer
- Shari Deiana, IBM Redbooks IT Support
- Elise Hines, IBM Redbooks Technical Writer
- Tamikia Lee, 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

1

Introduction

This IBM Redbooks publication introduces the concepts of the IBM PureApplication System offering, which is a cloud application platform for enterprise applications. It is pre-integrated hardware and software combination that is built by using IBM PureFlex™ hardware.

This book covers the most common solutions, practices, and use cases for PureApplication System success. The book also delves into the architectural decisions that are made within the PureApplication System and how they solve business challenges.

This chapter covers the following topics:

- Cloud computing
- IBM PureSystems family of products
- IBM PureApplication System overview

1.1 Cloud computing

Economic pressures are placing greater importance than ever on IT infrastructures to deliver more value with optimum resources. Businesses need to innovate faster and respond to the market quickly with accelerated development. They also need to reduce capital and operational expenditures, yet maintain a competitive edge. To do all of that, they need to increase flexibility of their IT infrastructures, applications, and services.

Enterprises must also satisfy a broad set of users, both internal and external. Their variances, such as the need to respond to peak capacity quality of service (QoS), strain IT resources. Meeting these requirements demands a new approach to delivering IT services, from metal to middleware and from middleware to the application services.

Cloud computing is that new approach. This is what it involves, according to the National Institute of Standards and Technology (NIST) definition:[1]

> "Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (for example, networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. This cloud model is composed of five essential characteristics, three service models, and four deployment models."

Cloud computing technology is built on the foundation of virtualization technology. The primary goal is to ease the use and delivery of IT services by supporting the essential characteristics that are accessible through a network, such as resource pooling, elasticity, and metered resources. Cloud computing alleviates the traditional way of provisioning resources, which is time-consuming, costly, and involves capital investment, acquisition costs, and procurement, in addition to the operational complexities.

The cloud computing services, as defined by NIST, are built in a stacked fashion, beginning with the infrastructure layer that is referred to as *infrastructure as a service (IaaS)*. At this layer, users request compute servers, storage, and network resources through a self-service portal, and the cloud platform's back end provisions these resources. The provisioned resources, with predefined capabilities, provide the same user experience as the physical infrastructure. With IaaS, IT services are delivered as a subscription service, which eliminates up-front costs and reduces ongoing support costs through automation. That automation can provide a consistent, repeatable, and error-free experience.

One notch above the infrastructure as a service is *platform as a service (PaaS)*. This provides middleware and an application platform as a utility. For example, if you like to provision an IBM WebSphere Application Server, for example, you get a consistent run time merely by clicking a button. To take it further, if you want to replicate that traditional three-tier topology of web server, application server, and a back-end database, PaaS provides the ability to use this topology consistently and repeatedly without errors.

With PaaS, development teams spend their time developing the applications rather than trying to manage the environment so that they can develop applications. This significantly accelerates the time to value and time to market for new applications.

[1] The NIST Definition of Cloud Computing
http://csrc.nist.gov/publications/PubsSPs.html#800-145

With a PaaS cloud environment, you can request platform services for application development through a self-service portal. This approach removes time-consuming manual processes and automates repeated and complex tasks through standardization of the deployment. PaaS also simplifies the post-deployment lifecycle management, because you can patch the systems easily and maintain them.

The sections that follow describe the essential characteristics of cloud computing. They also introduce IBM PureSystems™ offerings, which take cloud computing to the next level with simplicity of deployments, accelerated time to value, agility, and performance.

PureSystems products provide deeper integration of hardware and software with a single user interface enables full lifecycle management of the hardware, software, and the application deployed. The complexity of deploying environments and managing them through proprietary automation adds significant operational cost. Using the PureSystems patterns of expertise instead simplifies deployments, so you can focus on the applications that serve your business.

1.1.1 Basic principles of cloud computing

This section elaborates on the main themes that are relevant to enterprises to achieve cloud computing objectives.

Resource pooling

Resource pooling is the most important concept in a cloud solution. It is a way to achieve efficiencies and to enable resource reuse to satisfy the changing user self-service provisioning requirements.

Cloud computing resources generally refers to the compute, storage, and networking capabilities that are parts of the infrastructure that serves the cloud service.

The *host resources* in the cloud environment that are managed by the cloud management stack are treated as an aggregate pool of virtualized resources that are ready for provisioning.

The *pools* are made up of the compute, storage, and network resources that are part of the cloud infrastructure. The hypervisors report the resources that they are managing to the cloud stack.

The following functions are desirable in an enterprise cloud computing solution:

- Intelligent management
- Scheduling and distributing the workloads on demand
- Meeting quality of service requirements

Resource sharing

The ability to share resources is the basic premise of cloud computing. After the resource pools are identified, the cloud management stack takes ownership of manipulating those resources among the projects, teams, and individual users. Assignment of these resources can be based on quality of service parameters to meet the service levels that the users expect. The resources are dynamic in nature and are reclaimed when users relinquish them to the pool to be available to other users.

The virtualization technology gives you the ability to over-provision computing resources to abide by the policies that your system administrators set by using empirical performance metrics.

Resource isolation

Because of security concerns, users expect isolation of their allocated cloud resources from other user resources. Resource isolation and governance frameworks related to the cloud security model are required to take advantage of the benefits that are provided by a properly implemented cloud environment.

Systems that provide cloud services must support resource isolation through strict role-based access control mechanisms, and they must be tightly aligned with the enterprise security policies. Therefore, logical isolation of resources that meet the security and audit requirements, especially in production environments, is normal rather than an exception, and the cloud platform must support them.

Multi-tenancy

Cloud computing provides metered resources to users who are typically referred to as *tenants*. A tenant can be a single user, an organization, multiple organizations, or a team, depending on the assigned roles. It is rare to find a cloud environment that is entirely dedicated to a single tenant.

A set of logical resources (compute, storage, and networking) from within the shared resource pool of the cloud environment can be allocated to a tenant. There are typically enough resources available in a cloud, through correct capacity planning and thin provisioning mechanisms, to satisfy the needs of more than one tenant. The concept of *multi-tenancy* needs to be understood in the context of the ability to share a common pool of resources in a secure, co-existing manner.

Different tenants need different types of isolation. A cloud tenant might be an application developer or many application developers who are working together or individually. A tenant can also be multiple companies that are hosted on the same cloud by a managed service provider. For multi-tenancy in the cloud, resource isolation is critical.

As an example, application developers need isolation from other tenants that belong to a different organization while they are sharing the same hardware. Managed service providers might have requirements for physical resource isolation between tenants that must be supported. Yet cloud computing platforms must be able to support isolation, security, and multi-tenancy in a holistic manner, because they are related and tightly integrated.

Test and development clouds within the enterprise firewall typically do not have to meet strict regulatory and audit requirements. In such cases, a multi-tenant cloud environment with self-provisioning capability can be used with minimal reliance on the operations team. For production environments, the compliance with security policies, separation, and isolation of resources, both logically and physically, must be considered and must involve the operations team. However, standardization and automation help mitigate reliance on the operations team.

Most of the multi-tenancy issues are closely tied to the security of the platform, the ability to isolate the resources, and the ability to provide evidence that the system can be audited and meet audit and compliance requirements that are critical for enterprises. This book provides guidance and preferred practices for how to enable multi-tenancy on the PureApplication System so that enterprises can confidently use the system to satisfy all concerns in test, development, and production environments.

1.1.2 Cloud service models

Enterprises or managed service providers need flexibility with IT service offerings and the service delivery models that are closely aligned with their business goals. The ability to build hierarchical services provides that flexibility. NIST defines three main service models in the cloud:

- Infrastructure as a service (IaaS)
- Platform as a service (PaaS)
- Software as a service (SaaS)

The subsections that follow explain the functions and the issues that each of these addresses.

By using virtualization as the foundation, cloud computing is entering the mainstream with technologies that support workload consolidation and orchestration. Support for programming and abstraction at all layers of these cloud service models is enabling innovation for the betterment of IT service delivery.

The new programmable virtualized environment is changing the way that workloads are provisioned and managed. This is requiring close integration across the hardware and software domains, with advanced tools to ease the burden on IT operations.

Infrastructure as a service

IaaS is the first stage in building a cloud solution. The precursor to building IaaS is the need for an environment that is already virtualized with hypervisors. The traditional way that users request compute resources places a burden on the IT staff and does not add any value to the business. These repeated tasks, despite the similarity in the requests, need approval every time. That fulfillment approach, which involves manual processes, is time-consuming.

A new approach that automates the fulfillment cycle, but with a strict governance model built around it, makes sense. The ability to request virtual servers through a self-service portal alleviates the burden on the IT operations team and optimizes the ticket-based model for requesting new servers.

IaaS optimizes the provisioning mechanism of servers all the way to the operating system login for users. This leaves the responsibility of installing middleware, configuring it, and enabling application development to the user. When using IaaS, users have complete control of managing the middleware stack and any applications that are running on the servers and for upgrades and patches.

Platform as a service

The main focus for developers is a faster application development lifecycle. They are reluctant to take ownership of managing middleware, because it is more burdensome, time-consuming, and an extra effort that adds no value to their process. When the cloud service provider is managing the middleware and providing application programmable interfaces (APIs) or similar mechanisms to enable direct application development, the service model is referred to as *platform as a service* (PaaS).

The benefits of using this service are that you avoid dealing with the middleware layer and with the operational aspects of managing the system (security patches, bug fixes, new releases, updates, and so on). You can focus mainly on application development by relinquishing control of installation, configuration, and maintenance of the middleware to the hosting provider.

PaaS services can be implemented by the cloud service providers in either public or private cloud environments. The delivery models are described in 1.1.3, “Cloud deployment models” on page 6.

Software as a service

At this level of service, the enterprises and users are using the software that is delivered by the cloud hosting provider or an internal IT organization. Typically, software is provisioned as a canonical implementation that meets specific requirements and that multiple users or tenants can use with minimal configuration changes and account setup. The most familiar examples that are analogous to SaaS are collaboration services, email services, and customer relationship management.

The services are accessed over the Internet with no installation on the client side. Users relinquish complete control over the choice of middleware and infrastructure and use the software application as-is. Requests for implementation of custom requirements are difficult to meet unless they add value to a larger community of SaaS users. The SaaS provider is in complete control of the features and functions that are supported by the service. The software applications are accessed through a browser interface.

In summary, systems that are capable of integrating all of the cloud service layers with integrated tools and orchestrated workflows to support various application workloads are beneficial. This flexibility enables IT staff to provide improved services to their users.

By using the PureApplication System, you can build SaaS services. For examples of how to create these services, see the IBM Redbooks Solution Guide titled *Platform as a Service with IBM PureApplication System*:

http://9.12.6.4/itsoapps/redbooks.nsf/TipLookup/TIPS1019

1.1.3 Cloud deployment models

NIST defines private cloud, public cloud, community cloud, and hybrid cloud as the predominant cloud deployment models. Each of these models serves a specific set of enterprise requirements. The decision for selection of the appropriate model primarily depends on, but is not limited to, these factors within the specific enterprise:

- Security policies
- Elasticity requirements
- Workload sensitivity to latency

The four deployment models of cloud computing environments can be classified according to the levels of access, amount of control over the infrastructure, and the sharing of resources within and outside the enterprise data centers. Table 1-1 on page 7 lists the differences in the ways that the cloud deployment models access the cloud resources and controls.

Table 1-1 Cloud deployment models access information

	Private cloud	Public cloud	Community cloud	Hybrid cloud
Internet access	Restricted and controlled	Allowed	Restricted and controlled	Restricted and controlled
Virtual Private Network (VPN)	Industry norm	Available from certain providers to access virtual privately hosted clouds	Typically mandatory (site-to-site connections are also possible)	Typically used
Security controls	Inside the enterprise firewall	Outside enterprise firewall	Inside enterprise firewall	Inside and outside enterprise firewall
Resources use	Exclusively internal to the enterprise	Publicly shared with other tenants	Shared within Community	Exclusive to internal enterprise with burst capability for public resources

Private cloud

Enterprises with in-house information technology virtualization capabilities are ready to evolve and take advantage of cloud computing technology and the services that it offers. Cloud computing enables enterprises to consolidate workloads. Certain workloads are sensitive to the enterprises, and the security policies restrict the use of cloud hosting services that are outside of the enterprise firewall. When the cloud operating environment is supported within the enterprise firewall, it is referred to as a *private cloud*. There is flexibility and control with the private cloud, and it fits well within the established policies that are specific to an enterprise.

Technology and platform selection and decisions about the cloud requirements are within the purview of the IT department. In a private cloud scenario, the enterprise becomes a cloud service provider that serves only internal users. The private cloud environment is managed with established internal governance models and processes, such as the Information Technology Infrastructure Library (ITIL). Ownership of service level agreements (SLAs) and quality of service remains with the enterprise IT department.

A private cloud can be built to serve any of the cloud service models (IaaS, PaaS, or SaaS). IaaS is the foundation of a private cloud implementation. However, most of the benefits, from an enterprise perspective, are realized when there is automation higher up in the cloud stack. With IaaS implementation, users can request and use only virtual servers. The level of effort to bridge the gap between having a virtual server with an operating system (IaaS) versus having a virtual server that is ready for application development and use (PaaS or SaaS) grows larger with the number of applications to support and manage.

1.2.1, “Patterns of expertise” on page 10 explains more about *patterns* and how the IBM expert integrated systems can bridge the gap. Having the ability to create, deploy, and integrate composite applications through automated workflows that are based on proven practices is new to cloud computing platforms. Having such a system as the foundation for a private cloud eliminates the guesswork in provisioning middleware and software on the servers and builds applications that allow productive use of the IT staff and error-free operations.

Enterprises need to start thinking with a broader perspective when it comes to the choice of platforms for a private cloud. Platforms that can support all of the delivery models with expert integration provide more value than the private cloud solutions that target the IaaS layer alone. Using IBM PureSystems options in a private cloud environment, specifically IBM PureApplication Systems, helps enterprises meet the time-to-market application delivery requirements.

Public cloud

The public cloud gives users access to resources from the Internet, while satisfying the service models and the essential associated characteristics. The ability to request IT resources as a utility from public Internet gives it the “public cloud” name. As Table 1-1 on page 7 shows, access to a public cloud is through secure protocols over the Internet, and all resources are outside of the enterprise firewall. This is in contrast to a private cloud.

The origins of public cloud computing trace back to IT vendors’ abilities to elastically source compute servers and storage resources from within their hosted virtualized infrastructures. The charges for use of the resources are based on actual use, typically on an hourly basis. The public cloud provider is responsible for managing security controls within the hardened data center’s supported auditing capabilities. Economies of scale are achieved by allowing multi-tenancy and providing shared resources from a common resource pool through a self-service portal.

This low-entry model for IT resources is attractive to businesses if it meets their governance frameworks. Public cloud service is also opening avenues for developers and business startups to provide proofs of concepts without hardware or software of their own.

The range of offerings in the public cloud varies from a base IaaS all the way to SaaS offered through a service catalog. The resources are scalable on demand to meet the user or application needs, and they can be accessed from anywhere.

Community cloud

As cloud computing entered the mainstream, cloud vendors developed new deployment models to support specific regulatory compliance, audit, and security requirements. The community cloud provides the economies of scale through multi-tenancy while meeting government agency-specific requirements.

As an example, an IBM Federal community cloud provides IaaS services to government agencies that require a Federal Information Security Management Act (FISMA) moderate level of compliance. The environment supports multi-tenancy and meets the regulatory compliance requirements at the same time. As a result, government agencies get all of the benefits of a public cloud while meeting the FISMA moderate compliance level, which is a common federal government agency regulatory requirement.

Hybrid cloud

Infrastructure capacity planning is always a challenge. Most often, enterprises tend to over-provision the resources to meet business peak requirements or under-provision them, which results in a problem in satisfying needs during business surges. Over-provisioning wastes resources, but under-provisioning results in being unable to provide resources when they are needed. This is true for cloud implementations also. Enterprise decision-makers are looking for the capability to access additional resources to supplement the existing private cloud so that they can avoid new capital investment costs to meet occasional peak demand, yet still scale for future growth needs.

Private cloud solutions are designed with certain estimated capacity and some built-in growth factors. However, as business demand for compute resources increases, the ability to tap into resources outside of the current environment to meet the peak demand is a powerful attraction. This approach gives managers latitude when they are making decisions about whether to invest more capital in infrastructure scalability and growth needs versus using external resources on a pay-per-use basis. A hybrid cloud enables the enterprises to tap into external public or community clouds to meet short-term needs. This capability saves enterprises capital expense and satisfies the peak resources requirements through an operations expense.

Another key driver for using a hybrid cloud for resources outside of the enterprise firewall is the ability to connect to the public cloud through virtual private network (VPN) connections by running the IPSec protocol. This solution provides extra security beyond the normal HTTP and HTTPS access through the public Internet. If enterprise security policies allow this solution, a hybrid cloud is a good model for protection from unpredictable resource demands. However, this solution is ideal for workloads such as development and testing where security requirements are less stringent.

A hybrid cloud can be used for different purposes, for example:

- Resource optimization. To make optimal use of resources, you can deploy development and testing in a public cloud while running the production workloads in a private cloud.
- *Cloud bursting*. To meet sudden peak requirements when a private cloud is constrained by resources, the additional resource requirements can be satisfied by using a public cloud.

With a hybrid cloud, requested resources are transparently satisfied by the internal private cloud in combination with the public cloud or community cloud offerings from vendors, as needed, through secure VPN connections.

1.2 IBM PureSystems family of products

Information technology is becoming more critical for businesses as organization leaders seek to extract more real value from their data, business processes, and other key investments.

IBM introduced the PureSystems family of products to meet this IT challenge by combining these advantages:

- Flexibility of general-purpose systems
- Simplicity of an appliance that is tuned to the workload
- Flexibility of the cloud

This new category of systems represents the collective knowledge of numerous deployments, established preferred practices, innovative thinking, IT leadership, and distilled expertise.

IBM PureSystems products deliver value through three core attributes:

- **Built-in expertise:** These *expert integrated systems* help you address business and operational tasks by reducing manual steps and costs for support and maintenance. Systems capture and automate leading practices and expertise, and this allows solution providers to optimize their applications.
- I**ntegration by design:** All hardware and software components are designed to integrate, tuned in the lab, and prepackaged in the factory for optimal performance and efficiency.
- **Simplified experience:** Systems create efficiencies quickly by eliminating the need to procure, deploy, manage, and support IT components separately.

The PureSystems family includes the following offerings:

- **IBM PureFlex System**, which integrates compute nodes, storage, networking, and fertilization integrates compute nodes, storage, and networking into a single system with a single, unified management console.
- **IBM PureApplication System**, which is a cloud application platform for enterprise applications. It is workload-aware, flexible, and designed to be easy for you to deploy, customize, safeguard, and manage your middleware and other applications.
- **IBM PureData™ System**, a resilient, security-rich, scalable system that is optimized for transaction processing, reporting analytics, operational analytics environments, and Apache Hadoop capabilities.

1.2.1 Patterns of expertise

IBM PureSystems products include *patterns of expertise* that are based on proven practices for accomplishing complex tasks. This comes from expertise developed over decades of client and partner engagements. These preferred practices and expertise are captured, lab-tested, and optimized in a repeatable, policy-driven form.

Patterns of expertise bring efficiency, flexibility, agility, and control to your systems. There are three types of patterns (see Figure 1-1 on page 11) that correspond to different cloud service layers:

- **Application patterns:** Application patterns provide a predefined application architecture and the required platform services, which are deployed and managed by the system according to a set of policies. These patterns help build more robust, scalable, and easy-to-maintain application architectures.
- **Platform patterns:** Platform patterns provide preconfigured and policy-managed platform services, such as caching, elasticity, failover, load balancing, and security monitoring. With these services, platform patterns enable faster and more efficient deployment and management to application services, database, and messaging middleware.
- **Infrastructure patterns:** Infrastructure patterns provide an automated, policy-driven infrastructure management approach across compute, storage, and networking resources. These patterns reduce operational expenses and improve performance by enabling faster hardware configuration and simplifying low-level resource management.

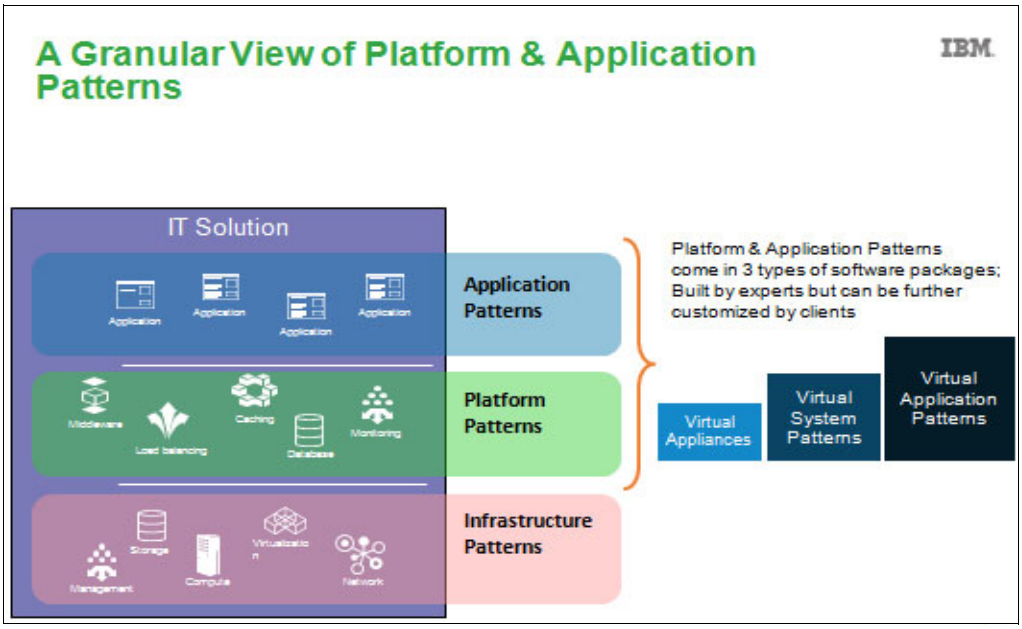

Figure 1-1 Platform and patterns of expertise

The IBM PureSystems Centre web page gives you easy access to PureSystems assets and expertise built into patterns from a broad environment, all efficiently deployable. The patterns that are designed by IBM developers or IBM Business Partners' developers are available in the PureSystems Centre, including explanations of their business value, technical details, and other information. For more information, see the PureSystems Centre:

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

For more information about patterns for the IBM PureApplication System, see the Redbooks publication titled *Creating Composite Application Pattern Models for IBM PureApplication System*, SG24-8146:

http://www.redbooks.ibm.com/Redbooks.nsf/RedpieceAbstracts/sg248146.html?Open

1.2.2 IBM PureFlex System

IBM PureFlex System gives you a simplified experience of control and choice without compromising the needs of your business. It provides a private cloud platform on an IaaS layer with a selection of compute nodes, operating systems, and hypervisors that fit your business requirements.

By using PureFlex System, you can use infrastructure patterns that automate and optimize the deployment and maintenance of your infrastructure. With this automation and optimization, you get the flexibility to handle unexpected demands without increasing resources.

The system has three optimized configurations that are available with the choice of IBM POWER® 7+ or Intel processor-based compute nodes:

- **IBM PureFlex System Express** is designed for small and medium-sized businesses. It is the most affordable entry point.
- **IBM PureFlex System Standard** is optimized for application servers, with supporting storage and networking. It is designed to support your key independent software vendor (ISV) solutions.
- **IBM PureFlex System Enterprise** is optimized for scalable cloud deployments. It has built-in redundancy for highly reliable and resilient operation to support your critical applications and cloud services.

1.2.3 IBM PureApplication System

IBM PureApplication System is a cloud application platform for enterprise applications. It is pre-integrated hardware and software that are built by using IBM PureFlex System hardware. PureApplication System provides a single pane of glass to manage hardware, middleware, and applications. By using the single view, you can manage the lifecycle of the system and workload. You order by only one part number, and PureApplication System is delivered. It is factory-integrated and tested. Therefore, it saves you time in integrating parts that you typically need integrate, otherwise.

IBM PureApplication System helps you achieve the following types of value:

- Agility. PureApplication System provides faster time-to-value by delivering complex application and middleware environments in minutes rather than days. It enables you to respond faster to the market with standard deployments, with lifecycle management capabilities built into the patterns.
- Efficiency. PureApplication System makes your IT systems efficient and optimized, not only from a virtualization standpoint but from physical attributes of the systems (energy-efficient, for example). This applies to middleware and software lifecycle management.
- Simplicity. This is one of the key values of PureApplication System. Simplicity starts from when you order the system by just one part number. The system is integrated, with hardware and software, factory-tested, and ready to power on with network connectivity. PureApplication System provides further simplicity in managing the lifecycle of the entire system. This includes managing and maintaining the hardware, as well as the software. You can find all of the included patterns in one catalog in the IBM PureSystems Centre.
- Control. PureApplication System provides consistent control over the deployments. It enables you to replicate traditional environments on a PureApplication System with a level of control that ranges from using existing technical investments to having PureApplication System handle the application monitoring and management.
- Scalability. PureApplication System can dynamically scale the system, from a hardware and software standpoint. To add extra capacity, you merely need to add the upgrade nodes. The system detects the capacity addition, which is then available for use. You can also set SLAs and attach a scaling policy to the application deployments. This allows the system to add VMs to handle peak loads and then take back resources when the load reduces.
- Reliability. The system is highly available and redundant from hardware and software standpoints. In a compute node failure, the system moves the VM to a different node. Similarly, when a scaling policy SLA is attached to an application, PureApplication System adds a new VM, configures it with middleware, deploys the application, and adds it to the cluster if any VM in the cluster fails.

IBM PureApplication System includes pre-integrated patterns that are the result of practices and expertise gained from decades of managing and optimizing IBM client applications. You can create custom patterns to deliver and manage applications, services, and business processes.

1.2.4 IBM PureData System

Accessing and processing data while it keeps growing in terms of volume, velocity, and variety is a big challenge. Different workload characteristics and application demands come with different hardware and software challenges that are time-consuming and costly.

IBM PureData System provides the ability and simplicity to reduce the complexity of deploying and managing data systems. It delivers data services for both traditional and cloud environments.

The expertise built into PureData System reduces the time, effort, and expertise that are required to build, optimize, and manage data systems. The system is designed, integrated, and optimized for different types of high-performance data workloads.

The following IBM PureData System models are optimized for specific data workloads:

- **IBM PureData System for Transactions** is designed, integrated, and optimized exclusively for online transaction processing (OLTP) workloads. It provides high scalability and reliability and improves performance. It is a perfect companion to IBM PureApplication System when an application requires higher levels of scalability and availability.
- **IBM PureData System for Analytics**, powered by IBM Netezza® technology, is a simple data appliance for serious analytics. It simplifies and optimizes the performance of data services for analytic applications, which enables complex algorithms to run in minutes, not hours. It is designed specifically for running complex analytics on very large volumes of data.
- **IBM PureData System for Operational Analytics** is a data warehouse system that is designed and optimized specifically for operational analytics workloads. This expert integrated system with in-database analytics eliminates the time, cost, and risk that are involved with copying data from the data warehouse to analyze it.
- **IBM PureData System for Hadoop** is built to deliver Apache Hadoop capabilities with easy-to-use analytic tools and visualization for business analysts and data scientists. It provides extensive capabilities with enhanced big data tools for monitoring, development, and integration with many more enterprise systems, such as Netezza, IBM DB2®, IBM PureData System for Analytics, and IBM InfoSphere® Guardium®.

1.3 IBM PureApplication System overview

IBM PureApplication System is an *expert integrated system* (see Figure 1-2 on page 14) that automates the operations that are needed for applications, hardware, and data center infrastructure and the application development, deployment, and adoption of cloud services. The IBM PureApplication System enables you to focus on core applications rather than on the routine tasks of managing and maintaining the infrastructure.

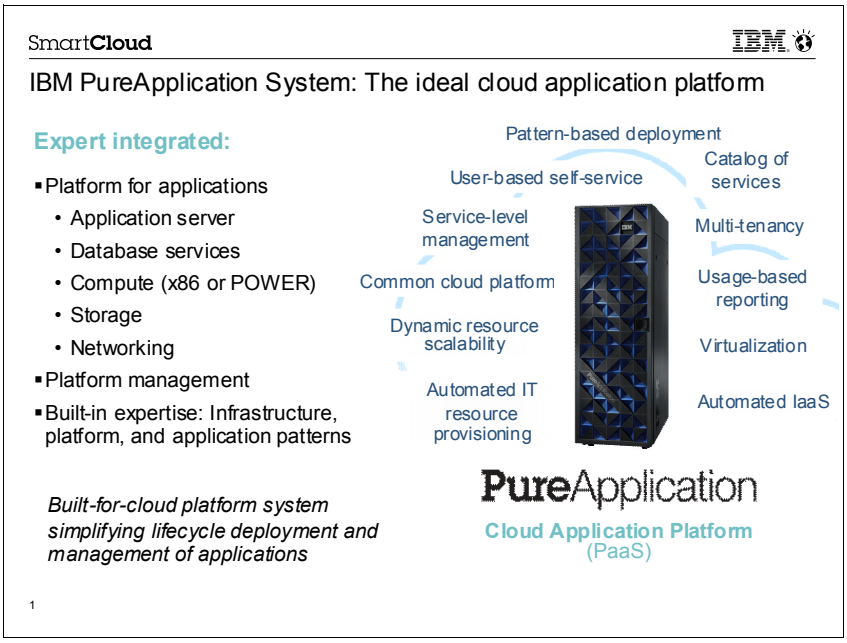

Figure 1-2 IBM PureApplication System is integrated by design

1.3.1 All cloud stack components are already integrated and optimized

IBM PureApplication System provides a PaaS layer because it includes pre-integrated patterns of expertise (see Figure 1-3).

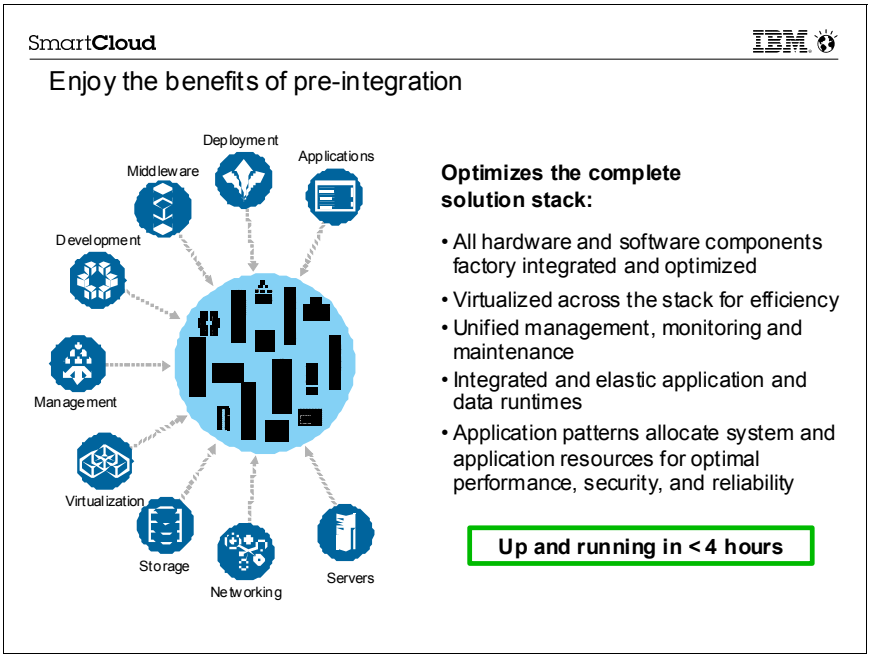

Figure 1-3 Benefits of pre-integration IBM patterns

You can use PureApplication System to create new patterns and to customize the existing ones. You can also use the patterns that are developed by IBM or IBM Business Partners, which are available on the IBM PureSystems Centre web page:

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

1.3.2 Configurations

IBM PureApplication System offers different configuration options to meet the different size and computing power that is needed for applications. There are three classes available, and it is possible to upgrade within the same class without incurring application downtime:

- Small rack by using Intel Xeon E5-2670 processors (W1500-32 and W1500-64)
- Large rack by using Intel Xeon E5-2670 processors (W1500-96 through W1500-608)
- Large rack by using IBM POWER7+™ processors (W1700-96 through W1700-608)

Table 1-2, Table 1-3 on page 16, and Table 1-4 on page 16 give you an overview of the configurations.

For detailed system specifications and site readiness information for IBM PureApplication System W1500 and W1700, see these information centers:

- IBM PureApplication System W1500: Planning information:

  http://pic.dhe.ibm.com/infocenter/psappsys/v1r0m0/topic/com.ibm.ipas.doc/iwd/gst_planning.html

- IBM PureApplication System W1700 - Planning information:

  http://pic.dhe.ibm.com/infocenter/psappsys/v1r0m0/topic/com.ibm.puresystems.appsys.1700.doc/iwd/gst_planning.html

Table 1-2 IBM PureApplication System configurations with Intel processors

PureApplication System W1500 (small rack)	**W1500-32**	**W1500-64**
Compute nodes	2	4
Processor cores	32	64
Memory	512 GB	1 TB
Solid-state drive (SSD) storage	2.4 TB	
Hard disk drive (HDD) storage	24 TB	
Rack	1.3 M 19-inch enterprise rack	
Power	Power distribution unit 4 x 60A 1ph - North America 4 x 32A 1ph - International	
	Maximum: 5.5 kW Typical: 4.68 kW	Maximum: 6.5 kW Typical: 5.52 kW
Weight	365.6 Kg (815 lb)	385.6 Kg (850 lb)
Dimensions	Height: 1.24 m (49 in), Depth: 1.0 m (39.4 in), Width: 0.61 m (24 in)	

Table 1-3 IBM PureApplication System configurations with Intel processors

PureApplication System W1500	**W1500-96**	**W1500-192**	**W1500-384**	**W1500-608**
Compute nodes	6	12	24	38
Processor cores	96	192	384	608
Memory	1.5 TB	3.1 TB	6.1 TB	9.7 TB
SSD storage	6.4 TB			
HDD storage	48.0 TB			
Rack	2.0 M 19-in enterprise rack			
Power	Power distribution unit 4 x 60A 3ph - North America 4 x 32A 3ph - International			
	Maximum: 7.9 kW Typical: 5.9 kW	Maximum: 10.4 kW Typical: 7.8 kW	Maximum: 15.4 kW Typical: 11.6 kW	Maximum: 21.2 kW Typical: 15.9 kW
Weight	936 Kg (2027 lb)	953 Kg (2095 lb)	1016 Kg (2232 lb)	1088 Kg (2391)
Dimensions	Height: 2.015 m (79.3 in), Depth: 1.098 m (43.3 in), Width: 0.644 m (25.4 in)			

Table 1-4 IBM PureApplication System configurations with POWER7+

PureApplication System W1700	**W1700-96**	**W1700-192**	**W1700-384**	**W1700-608**
Compute nodes	3	6	12	19
Processor cores	96	192	384	608
Memory	1.5 TB	3.1 TB	6.1 TB	9.7 TB
SSD storage	6.4 TB			
HDD storage	48.0 TB			
Rack	2.0 M 19-in enterprise rack			
Power	Power distribution unit 4 x 60A 3ph - North America 4 x 32A 3ph - International			
	Maximum: 9.3 kW Typical: 8.7 kW	Maximum: 13.4 kW Typical: 12.4 kW	Maximum: 21.8 kW Typical: 20.0 kW	Maximum: 33 kW Typical: 28.8 kW
Weight	936 Kg (2027 lb)	953 Kg (2095 lb)	1016 Kg (2232 lb)	1088 Kg (2391 lb)
Dimensions	Height: 2.015 m (79.3 in), Depth: 1.098 m (43.3 in), Width: 0.644 m (25.4 in)			

1.4 Concepts and terms

The concepts and terms that are explained in the list that follows give you a better understanding of the remaining chapters in this book. For more detailed explanations, see *Adopting IBM PureApplication System V1.0*, SG24-8113:

http://www.redbooks.ibm.com/Redbooks.nsf/RedbookAbstracts/sg248113.html?Open

- **Compute nodes:** A compute node is a combination of computer hardware components that contains the processor, memory, and network, along with the storage adapters that have access to storage and networking. Each compute node runs a hypervisor to enable multiple VMs to share underlying hardware.
- **IP groups:** An IP group is a logical group of one or more IP addresses that have a virtual local area network (VLAN) ID and the information that is necessary to connect to the network. All of the addresses in the same IP group must belong to the same subnet of the network, indicated by a netmask.
- **Cloud groups:** A cloud group is a virtualized platform that contains one or more compute nodes and one or more IP groups for running workloads. It is a logical group of resources that enables system segmentation and compute node aggregation.
- **Environment profiles:** An environment profile is a set of policies that define how patterns are deployed to cloud groups and how deployed instances run in the cloud group. These profiles specify the cloud groups to which users or user groups can deploy patterns.
- **User groups:** A user group is a set of users that are in the same role. The set contains lists of users and the action permissions that those users have. One of the main functions of a user group is to specify which users can deploy patterns to which environment profile.
- **Shared services:** A shared service is a virtual application pattern that can be used by multiple virtual applications, virtual systems, and virtual appliances. Only one instance of a type of shared services can be deployed in a cloud group. This shared service can be used by all deployments in the cloud group.
- **Prioritization:** Each VM has priority settings that define the importance of the VM relative to all other VMs in the same cloud group. When VMs in the same cloud group require more resources than the cloud group has, the system gives preference to the higher-priority VMs.
- **Resource requirements:** You can use PureApplication System to define the required resources, such as processor count and the amount of virtual memory required for VMs to run correctly.

1.4.1 Patterns

Patterns are a deployable and repeatable form of expertise that is gathered to solve complex tasks. IBM PureApplication System provides different deployment types by using three different pattern types:

- **Virtual system pattern:** This pattern is a provisionable unit of one or more virtual images to be installed, configured, and integrated to implement a topology. These platform type patterns provide more customization capabilities than virtual application patterns and enable fast and automated provisioning of middleware-centric designs.

  For example, you can define a virtual system pattern for a middleware topology that contains a deployment manager, one or more custom nodes, and an IBM HTTP Server. You can define the VM images, the middleware components, the script packages that run to configure the components, and any monitoring agents that you want to include. That pattern can be deployed multiple times, with the same result each time.

- **Virtual application pattern:** This pattern encapsulates the installation, configuration, and integration of middleware, along with the installation and configuration of applications that run on that middleware. These application type patterns use an application-centric approach. Although using virtual application patterns gives you less flexibility than virtual system patterns, deploying your application is quicker and easier.

  When you define a virtual application pattern, you describe the application to use. PureApplication System builds the appropriate infrastructure and deploys the application. These types of patterns are also fully automated and are similar to virtual system patterns, but they add integrated lifecycle management.
- **Virtual appliance pattern:** This pattern combines the operating system, middleware, and applications in one package. Virtual appliances facilitate a quicker transition to cloud services and require significantly fewer installation and configuration tasks than traditional deployment models.

For more information about patterns and deployment models, see Chapter 4 of *Adopting IBM PureApplication System V1.0*, SG24-8113:

http://www.redbooks.ibm.com/Redbooks.nsf/RedbookAbstracts/sg248113.html?Open

1.4.2 Management

IBM PureApplication System provides a unified interface for managing the entire system. This single unified console provides management and monitoring interfaces for system hardware, virtualized storage and networking, license management, audits, and security configuration.

Administration interface types

PureApplication System provides unified system management that is accessible in these three ways:

- Graphical user interface (GUI)
- Command-line interface (CLI)
- Representational state transfer (REST) application programming interfaces (API)

Administration console (GUI)

IBM PureApplication System provides a single management interface for the entire system. You can use this integrated administration console to manage and maintain these aspects of the system:

- Monitoring
- License use
- User auditing
- Workload deployment
- Security configuration

All of these system and workload functions are available in the same integrated console.

There are two aspects of the administration console: system console and workload console.

- **System console** covers system management, administration, and maintenance, such as the following tasks:
  - Hardware management
  - User and group management
  - Auditing
  - Hardware monitoring
  - Problem determination

You can use this part of the console to perform cloud operations, such as creating and managing a cloud or IP groups and environments.

- **Workload console** covers the tasks for the workload such as these examples:
  - Creating and deploying patterns
  - Working with catalogs
  - Managing deployed VMs

  The workload console also provides capabilities to add existing applications to PureApplication System.

Command-line interface (CLI)

The PureApplication System CLI provides a scripting environment that is based in Jython. You can download this CLI from the Welcome page on the PureApplication System administration console (the web GUI). This interface enables you to run complex automation scripts to make maintenance easier. The CLI interface has the same options that the administration console has.

REST API

PureApplication System provides the representational state transfer (REST) application programming interface (API). A subset of PureApplication System functions is available in the REST API. You can access this interface by using the same IP address or host name that you use to reach the web GUI and the CLI. The REST API is supported only over hypertext transfer protocol secure (HTTPS) on port 443.

For more information, see *Adopting IBM PureApplication System V1.0*, SG24-8113:

http://www.redbooks.ibm.com/Redbooks.nsf/RedbookAbstracts/sg248113.html?Open

Monitoring

IBM PureApplication System provides pre-built monitoring capabilities that enable an administrator to monitor hardware, virtual machines, middleware, and applications. You can choose to use the built-in monitoring functions by using the Monitoring console or choose to use existing monitoring solutions by integrating PureApplication System with them.

In PureApplication System, monitoring of system, middleware, and hardware layers enables you to get information about the status, performance, and resource use for both hardware and software.

Maintenance

IBM PureApplication System involves two aspects of maintenance:

- *System maintenance* includes updates for system components, such as hardware and software components, compute nodes, switches, storage, hypervisors, and management software.
- *Workload maintenance* covers updates to software that runs within VMs, such as the operating system, middleware, and applications. This maintenance is applied to the virtual system and virtual applications.

License

PureApplication System contains tools for license management. You can monitor license use on a per-server basis or by using the processor value unit (PVU) model. In addition to real-time monitoring of licenses, PureApplication System allows you to limit license use and to export historical license use information.

For more information, see *Adopting IBM PureApplication System V1.0*, SG24-8113:

http://www.redbooks.ibm.com/Redbooks.nsf/RedbookAbstracts/sg248113.html?Open

Security

PureApplication System provides resource isolation and protection at different levels that establish and manage trust across the entire system. It uses a key to encrypt the contents of the disk, which allows the content to tie with a specific system.

There is only one operating system user ID on the system, but you cannot log on to the system externally with the user ID, because there is no shell available. All of the virtual images and other content that are used for deployment to a VM run in the VM on a hypervisor, not on the system. It is also possible to run security scripts during deployment of a pattern so that your VMs are compliant with your security settings from the first moment.

PureApplication System provides a combination of role-based security access control and resource instance-based access control to balance role-based and more granular resource-level access control. The role-based authorization design is based on both the separation of duty security principle and the least-privileged security principle.

IBM PureApplication System does not provide any capability for users to upload executable code or scripts. It is only possible to update system firmware that is digitally signed, and this update requires an administration authority role.

For more information, see the “Security overview” section of the IBM PureApplication System W1500 Information Center:

http://pic.dhe.ibm.com/infocenter/psappsys/v1r0m0/topic/com.ibm.ipas.doc/iwd/aac_overview_sec.html

1.4.3 Problem determination and support

IBM PureApplication System enables you to access both system logs and workload logs separately, to assist you with troubleshooting.

The system logs track actions that are performed by the system components, such as these examples:

- Hardware
- Firmware
- System components
- Management nodes

System events that are generated by system components are collected by using Simple Network Management Protocol (SNMP), and they can be accessed through the administration console.

PureApplication System has a built-in tool called the infrastructure map that shows the entire rack and components. This tool has two views: graphical view and tree view.

The workload logs contain information that is generated by workloads, such as operating system logs from VMs and middleware runtime logs.

PureApplication System provides a single point of contact to create service requests for all platform components, such as compute, storage, firmware, and VMs. You can submit a problem by using the web-based service request (SR) tool or by calling IBM support.

For more information about troubleshooting, see these publications:

- The “Troubleshooting and support” section of the IBM PureApplication System W1500 Information Center:

  http://pic.dhe.ibm.com/infocenter/psappsys/v1r0m0/topic/com.ibm.ipas.doc/iwd/tst_trouble_overviewnonUI.html

- Chapter 8 in *Adopting IBM PureApplication System V1.0*, SG24-8113:

  http://www.redbooks.ibm.com/Redbooks.nsf/RedbookAbstracts/sg248113.html?Open

2

Challenges for IT organizations

In this chapter, we examine the challenges that businesses and IT organizations face today. We explore the new system capabilities that are requirements for an enterprise to achieve the faster time to market and IT efficiencies and to focus on innovation.

This chapter covers the following key challenges that enterprise decision-makers need to address while designing and implementing their IT strategies:

- Cloud computing and optimization
- Innovation and optimization
- Simplicity and consolidation

2.1 Cloud computing and optimization

Cloud computing service models are structured in a layered fashion to accommodate different use cases. Because the infrastructure as a service (IaaS) model provisions a virtual machine to work with as a minimum, users must still do more work to install and configure middleware and to perform tune-ups to create a fully functional system that is ready to run applications.

Enterprise IT teams and leaders are familiar with IaaS and its capabilities. With IaaS, certain efficiencies are realized through the self-provisioning capabilities. However, businesses cannot realize all of the PureApplication System benefits from IaaS alone. Software developers typically work at the middleware and application layers. Although IaaS provides a platform for compute, networking, and storage resources in an efficient manner, it does not provide any appreciable benefits at the middleware and application layer. Therefore, working only at the IaaS layer does not truly provide a larger incremental benefit for enterprises that deploy application environments frequently for accelerated time to market and faster time to value.

Enterprises get more benefits and a better return on investment with systems that allow the IT team to deploy applications and middleware environments rapidly. Systems that support lifecycle management of middleware and the application layer provide the much-needed gain in efficiency to be able to respond to the market faster. Most cloud computing solutions accomplish integration at the hardware, network, and storage layers and, more recently, at the database layer.

However, significantly more efficiency benefits and cost savings result when the system, not the IT staff, builds, configures, and tunes applications, based on the assessment of workload and system resource capabilities. The ability to support all of the layers of a cloud computing stack within a single integrated system provides a higher value to the enterprise than a pure IaaS cloud system. Savings accrue by using standardization, automation, and elimination of error-prone, manually repeated tasks.

2.1.1 Changing the dynamics of the enterprise

The IT operation and application administration teams spend quite a bit of time building user environments that typically involve many software components and many integration points. The more complex that the middleware and other software are, the more time and effort that are required. The time is spent not only in building the environment but in maintaining it thereafter. This involves managing run times that are supported, but they might not be supported in coming years.

How do you cope with such dynamic shifts in programming environments while ensuring that the applications are providing the necessary business value to the organization? There is a limited use of expertise within an organization, and it is mostly through the knowledge management systems. Application of this expertise still involves reading, understanding, and applying the fundamentals to the current situation and context.

IBM clients gain appreciable benefits are realized when the collective knowledge and experience gained through thousands of field deployments of applications are combined with proven processes into a system-deployable format. Taking advantage of this experience and expertise reduces risk, and automating repetitive, error-prone work reduces cost and increases efficiency. Features that support automated tuning of new workloads and patch deployment with minimal impact to system run time simplify the IT process. Security is enhanced, because the deployments are run against predetermined policies.

Cloud computing that is supported by systems with built-in intelligent workload management software is changing the dynamics of IT in the enterprise. The most successful businesses rely on the following key factors that make this possible:

- Standardization
- Automation
- Reusability of assets across all layers of the cloud stack
- Governance

The ability to check the environment with a unified console, with a single view of the entire integrated system, helps IT administrators immensely in day-to-day operations. It allows them to see relationships among various failures and to coordinate faster resolutions. Think of the ability to correlate the logs of failures for an application server, database, application, hardware at a point in time and being able to pinpoint the cause of the failure, rather than trying to follow the symptoms serially. This ability to track down failure in increasingly complex systems is one of the bottlenecks that prevent responding to failures faster.

The ability to manage a physical server, virtual servers, storage, network, and applications from a single console is another requirement to make the work of the operations and applications teams simpler. This enables IT teams to operate efficiently and spend more time on innovative business transformation initiatives.

With the dynamically changing business and IT environment that is influenced by the increasing use of social media, mobile devices, and analytics, systems must be scalable to meet the current and future needs. Ease of integration with existing IT infrastructure with little impact to the enterprise is essential. Business decision-makers want intelligent systems that enable them to scale business applications while managing IT environments with solid governance and optimization to use resources optimally and to control IT capital and operational expenses.

2.1.2 Roles and process

Traditionally, enterprises have IT teams that are organized in different silos, and that is for the right reasons. Enterprises might have more silos, but these summarize the key ways that most enterprises organize their IT departments:

- The hardware or infrastructure procurement team guides the company's hardware strategy.
- The operations team or a "shared services" team manages and deploys the appropriate operating systems, middleware, and tools on these physical systems to ensure that the internal clients (such as development, QA, and production teams) are serviced appropriately.
- The network team manages the intranet, IT systems and subsystems, and the exit to external networks.
- The security team manages the complete enterprise IT compliance.

Such organizations have worked well, traditionally. But the pressure on enterprises to respond to the changing market faster is now much higher. There is an urgent need to reorganize the internal IT structure to make the enterprise more agile.

Traditionally, the enterprise infrastructure teams are responsible for defining the specifications and sizing of the hardware to support the application needs. The requirements from software teams regarding what is needed at the infrastructure layer are sometimes not detailed. There is much reliance on historical use and performance metrics in sizing the environments.

When IBM staff, Business Partners, or consultants build integrated systems, they consider all that is required to support the middleware, database, and application performance. The data is based on implementations of similar workloads across thousands of engagements.

2.2 Innovation and optimization

Experience with IBM client engagements shows that a specification and design phase of a typical IT infrastructure system takes about 30 days, on average. Procurement of software and hardware that are ordered separately can take anywhere from a week to four weeks, depending on the lead times of the selected components. After integration at the infrastructure layer, even more effort is necessary to optimize the environment to support workloads.

Depending on the complexity of the solution, the deployment phase for a fully functional system can take weeks to months. Development and test environments also take considerable time, sometimes up to a few months. One of the major issues that every IT organization faces is the delay in production deployments. These delays sometimes run weeks due to errors, especially when using modified and nonstandard development, test, and production configurations.

IT teams are constrained in their ability to quickly provision compute, storage, and network resources to support requests from hundreds of developers, possibly from different business units. The developers often have their own customized set of development and off-the-shelf software applications that require provisioning to their target development platform. This support model adversely affects crucial IT staff, leaving them less time to support the more important and continually changing business goals and requirements.

It is common for development and test groups to share resources on virtual platforms. Simplification of dynamic allocation of compute and storage resources to meet multiple project needs and automation of the process makes users more productive and helps the business save money. There is a gap in enterprises today between well-integrated tools that support application-scaling policies and apply maintenance patches to the middleware components.

to the middleware components.Optimization at the infrastructure layer helps, but maximum business benefits are realized at the enterprise level only when a similar approach is taken at the middleware and application provisioning areas, too. The trend is for the developers to define system requirements to meet the application needs and then to provide the requirements to the infrastructure team to build systems accordingly. Sometimes, the requirements are incomplete. This results in allocation of suboptimal resources and affects the ability to meet expected quality of service and service level agreements (SLAs). Lack of sophisticated tools and automation makes it harder for service providers to commit to SLAs in production environments.

From an application perspective, managers want to closely match the development and test environments to the production environment. The ability to replicate a production environment by minimizing the manual effort in creating the environment helps with faster build and test cycles, while also addressing application performance, scalability, and reliability issues. Enterprises greatly benefit from such optimization.

To meet the SLA customization, ongoing tuning of the middleware is also required. In some instances, the initially allocated resources need to be able to scale to meet performance goals needs. In other words, dynamic elasticity in the allocation of resources according to application needs is required.

Managing and monitoring the environments and creating an ability to do that from a single view with multiple vendor components can be time-consuming. It also requires development effort and new skills. Using multiple tools, sometimes from different vendors, results in situations where integration is complex, time-consuming, and prone to finger-pointing about errors that result from incompatibility. Applying fixes separately requires separate testing. Upgrades require months to plan, procure, and test. Migration of applications to and from different environments is another area where IT staff spends considerable time and resources.

Dependency on the IT staff for mundane tasks continually increases. This often slows their ability to meet fast-paced business requirements. When self-service capabilities for repeatedly requested resources are available, the impact on IT staff is minimized. Therefore, it is beneficial to adopt system solutions that mitigate these problems with error-free provisioning and to automate repeated tasks that otherwise add no value to the business.

At the data center level, optimization helps mitigate the concern about rising energy and floor space costs in the data centers. Decision-makers often prefer to focus on the current capacity needs and defer investments to meet future growth requirements. Therefore, systems that allow delaying up-front capital investment for future system growth are desirable. This requires optimized systems, with balanced compute, network, and storage capacities that deliver a specific, well-defined set of capabilities.

2.2.1 Ease of use

Systems that bring new capabilities and innovation must also be easier to install, configure, and tune to realize true runtime improvements. Run time is typically measured from powering on the system to when the system is ready to support application deployments. Traditionally, incorporating new systems in a data center goes through a lifecycle of its own, with the following stages:

- Compatibility issues with different vendor systems
- Complex configurations
- Interoperability problems
- Performance of the system as a whole
- Custom configuration for networking connectivity
- Storage backup
- Security

Enterprise managers also want to ease coordination between the development and operations teams so that new capabilities that are critical for business can be introduced quickly, without delays and with minimal expense. Systems that substitute manual processes at different stages of the software delivery lifecycle with automation that ensures successful deployment are vital. To achieve this, the system needs to provide ways to help with these key aspects:

- Planning for the build
- Integration and deployment of applications into production
- Test environments that are assisted by automation

Standardization is the key to ease of use.

License use tracking is another area that is difficult for enterprises and, if not properly implemented, is an impediment to compliance. Typically, these features are overlooked in system designs, but having tools that are designed to manage licenses and track and report them accurately resolves compliance issues and saves the IT staff valuable time. Automating these tools helps with productivity and eliminates the need for custom or multiple vendor license management tools and integrating them.

2.2.2 Availability

Unplanned downtimes and outages are to be expected in a data center environment. The reasons range from system failures to manual errors. Systems that can restore the applications to serve users with minimal impact after an incident are highly desirable.

Depending on the type of environment, production or otherwise, the systems that serve users must be capable of meeting the expected service levels. Inability to meet SLAs potentially costs enterprises new business and possible penalties on existing service. The goal for a business is always to provide quality services to their users, either internal or external. While they are implementing services, the IT teams are expected to manage the service levels at all times, at both the infrastructure and the application tiers. When incidents arise, resolution time is always a concern. Unplanned downtimes due to power outages or malfunctioning of hardware or applications due to problems with a patch push the IT staff into a reactive mode of operation.

Restoration of the environment to the original state without violating the SLAs is a challenge and another reason that enterprises are looking for help with system designs. There are always concerns about the length of time that it takes to configure the environment and to bring the environment back to the level of function that existed before the outage. This is a significant challenge with disparate and non-cohesive systems. A new system design approach that has software intelligence to automatically restore the system and configure the middleware and applications to their working states is necessary in today's business environment. This is extremely important in production environments so that they continue to meet the SLAs and availability requirements.

The ability to apply pre-configured changes is a major innovation that is employed in integrated systems. It is also a capability and innovation that disparate systems fail to meet.

2.2.3 Promotion between environments

A typical IT project lifecycle involves multiple levels of interactions between infrastructure and software teams. Traditionally, this has worked, but to respond quickly to the market, these expectations now seem to impede the organizational efficiency. Enterprise managers are exploring ways to minimize the time that is spent on the various phases of implementing the IT projects so that teams can improve the deployment times. The ability for software development teams to minimize their effort in the integration, configuration, and tuning of the applications saves time and increases business agility.

Depending upon the size of the enterprise, requests for compute resources and the rate of configuration changes for the development environments are typically high and demanding on IT staff time and resources. Although the nature of the requirements falls below the stringent production environment specifications, it is a strain on the IT staff to address the velocity of requests while concurrently meeting the production, development, and test needs and honoring the service level commitments. Incident resolution for the development environment drops in priority when there are production environment issues to address.

The inability to satisfy the requirements of the development environment prolongs the development cycle, and that affects the production schedules. When this happens, you do not meet business commitments to your customers. That costs money and potentially leads to changes in requirements due to competition and redesign.

A solution for this dilemma is to provide a self-service catalog that allocates the involvement of IT staff to fixing infrastructure issues only, not the mundane tasks of provisioning servers for the development environment. This approach also minimizes the interaction between cross-domain IT groups to satisfy the requests that are coming in from the development groups. Faster resource access improves development cycle times.

Quality assurance is the next step after development, and, many times, most of the errors are discovered here. One of the discoveries is the use of hardcoded variables in the development phase. A closely integrated approach through development and operation integration minimizes the delays. Pre-production environments sometimes are in similar scale to the production environment, so they require testing of the scalability of the solution. Moving seamlessly from pre-production into the production environment is expected. Enterprise teams face challenges in all of these areas, with trying to keep up with the tuning of the applications, deriving standard configurations, and applying standard templates whenever possible, for example.

A new system approach is necessary to address these challenges and bring development and operations under a single umbrella. Integrating the complete software delivery lifecycle that is coupled with infrastructure provisioning brings in new efficiencies that translate into better return on investment for the enterprise.

2.3 Simplicity and consolidation

Enterprise managers are looking for ways to optimize the number of data centers that they operate and efficiently manage the footprint of servers, storage, and networks within the data centers. Typically, the consolidation effort is a long-term strategy to address how IT assets can be better managed and operate more efficiently to improve delivery of IT services to the business users. This strategy also aligns with the enterprise strategy of standardization of infrastructure across locations, which makes management and maintenance easier. Choice of vendor platforms and different technology hardware from the vendors adds another dimension of complexity to the operation and maintenance of the systems. Compute, network, and storage devices from multiple vendors typically have different integration and interoperation capabilities. This is another reason for enterprises to consolidate. Consolidation, consistency, and compatibility also reduce the total cost of ownership.

Enterprises spend a considerable amount of time in the planning and execution of consolidation projects. The efforts vary by the size of the environment and the diversity of the platforms and workloads. Most of the platforms that are used by enterprises today are based on Microsoft Windows, Linux, or UNIX operating systems that run various applications. The middleware software that is used also varies. Organizations are maintaining and supporting multiple levels of operating systems, middleware, and databases. They are spending a considerable amount of money in doing so, rather than investing that money in innovative projects. All of these efforts are still based on people with skills that are supported by a multitude of tools or home-grown scripts.

The concept of patterns and automation is relatively new to IT organizations. The remaining chapters in this book explain the benefits of a pattern-based approach and the new integrated systems that are built on automating expert skills. To realize the benefits of consolidation, a significant amount of time is spent in understanding how the systems interconnect and interact across the various software components. Distributed environments are challenging to manage and maintain without a holistic end-to-end approach to the infrastructure and middleware. Assessment of IT infrastructure against the organization's current (and future) business model is critical to the selection of the correct platforms.

Enterprise managers are looking for ways to focus more on innovation and reduce two key unnecessary IT maintenance costs:

- IT lifecycle management
- Maintenance simplification and reduction of repetitive tasks, not just at the hardware level but higher up in the stack, from metal to middleware and middleware to applications

The ability to create virtual servers, virtual networks, and virtual storage on a collection of physical hosts helps with the consolidation. However, built-in intelligence to manage the virtualized environment by allocating resources based on use and workload characteristics and through provisioning that is based on available resources eases operations and maintenance of systems.

This can take management of IT systems to a new level. IT consolidation optimizes the infrastructure and enables organizations to do more with less.

Businesses realize the benefit only when the IT department is more responsive and able to provision resources faster, error-free, and in a standard and repeatable way while making costs clear to decision-makers. Systems that deliver both IT and business benefits bring value to the organization.

Applications are getting more complex, and the integration points are many. We call the systems that provide tools to implement proven application topologies in a consistent manner, eliminate repetitive manual tuning tasks, and enable asset reuse as *expert integrated systems*.

Figure 2-1 shows the high-level requirements that enterprise clients like to automate to reduce costs.

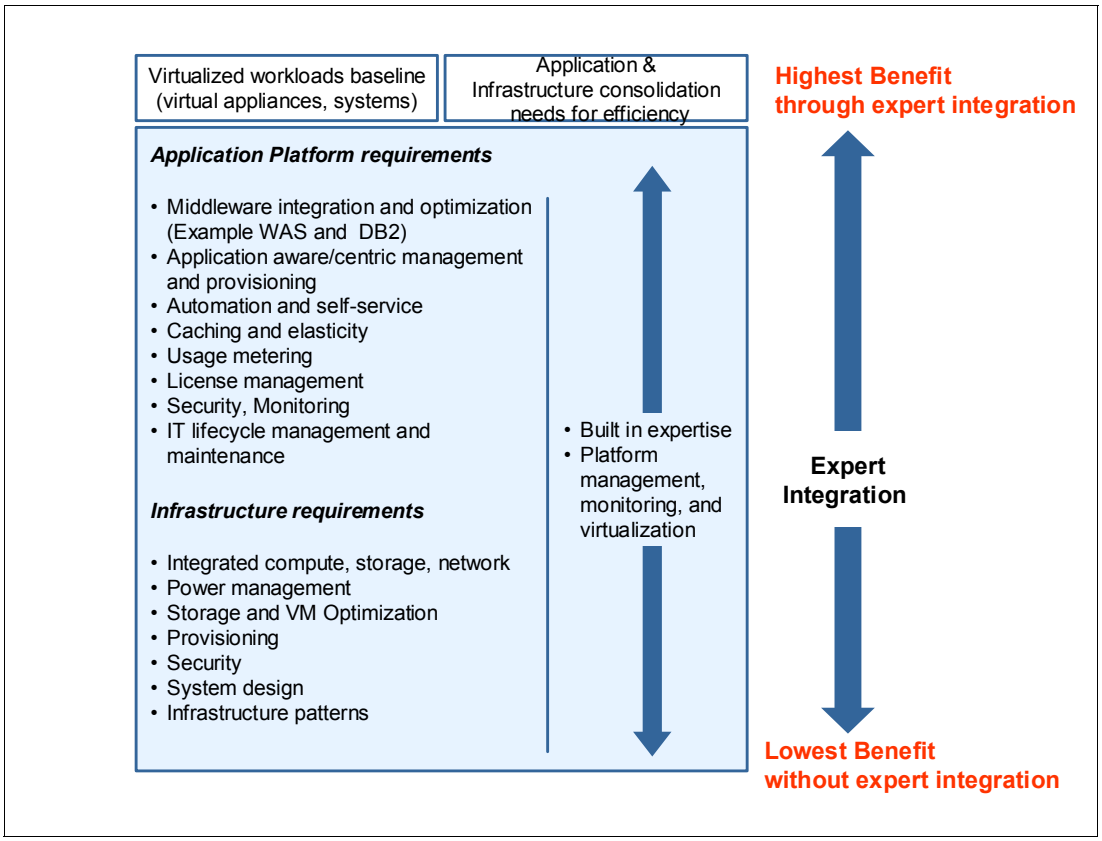

Figure 2-1 Requirements and automation to derive cost benefits

Managers are looking for systems that can support these higher levels of integration, automation, and consolidation. Expert integrated systems address both the application and infrastructure issues.

3

Organization and role changes to adapt to automation

In April 2012, IBM introduced the IBM PureSystems family of *expert integrated systems* with its first two products, IBM PureFlex System and IBM PureApplication System. In November 2012, IBM PureData System, optimized exclusively for delivering data services, joined the PureSystems product family.

As the IBM PureSystems products have been introduced, particularly PureApplication System, many clients have asked what impact this new class of systems might have on their IT organizations. This chapter provides detailed guidance to organization managers who want to adopt PureApplication System.

It covers the following topics:

- Introduction
- Organizational principles and practices
- Roles for PureApplication System implementation
- Structure of a PureApplication Center of Excellence

3.1 Introduction

IBM PureApplication System is a cloud application platform that is designed and tuned specifically for transactional web and database applications. The workload-aware, flexible platform is easy to deploy, customize, safeguard, and manage. It reduces the amount of labor and time that are typically required to support new and existing workloads. PureApplication System implements the platform as a service (PaaS) cloud level. With this platform, you can either create your own patterns of software, middleware, and virtual resources or use patterns that are included. You can provision and share these patterns within a unique framework that is shaped by IT guidelines, industry standards, and years of developing preferred practices, which is why we call IBM PureSystems products *expert integrated systems*.

3.1.1 The value of patterns of expertise

IBM built expert knowledge into PureApplication System with *patterns of expertise*. A pattern can be thought of as a predefined architecture of an application. Each component of the topology comes preinstalled with the OS and middleware, and the pattern is fully integrated across components, preconfigured, and tuned, with built-in hooks to advanced services, such as security, monitoring, and caching.

Patterns of expertise contain proven practices for complex tasks that we have learned from decades of IBM client and partner engagements. We captured, lab-tested, and optimized these patterns, and then built them into PureApplication System. Each system provides several ready-to-use patterns for common enterprise applications, such as IBM WebSphere Application Server software and IBM DB2 database. You can purchase patterns for other middleware, such as IBM WebSphere MQ, IBM Integration Bus, and IBM WebSphere Portal, among many others from IBM and IBM Business Partners. Those are available through the IBM PureSystems Centre:

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

You can also use the built-in editors to easily customize existing patterns or build new ones from ready-to-use components or from custom components that you create. Each pattern can then be deployed multiple times. PureApplication System provides lifecycle management of the deployed patterns, which we refer to as *instances*.

Patterns deliver faster time to value by removing manual steps and automating system delivery. They improve efficiency and simplicity by reducing IT costs and necessary resources and by reducing the amount of in-house expertise that is required for deployment. Patterns also increase your control, because repeatable, optimized deployments reduce the risk of human error.

The five-part series of IBM developerWorks® articles titled “Preparing for PureApplication System” describes the different types of patterns that are available:

http://www.ibm.com/developerworks/views/websphere/libraryview.jsp?search_by=Preparing+for+IBM+PureApplication+System+,+Part

3.1.2 Major steps in adopting PureApplication System

After you make the decision to acquire PureApplication System, your next steps are in these four major areas:

1. Install and set up PureApplication System in the data center.

   There is a set of preinstallation steps to ensure that the data center has the prerequisites. This ensures that the logistics, power requirements, network, and other requirements are met before the arrival of PureApplication System hardware at your site. IBM has a well-defined process for obtaining this information and making it available to the service support representative at installation time. Although IBM representatives perform the installation, your participation is required in obtaining this information and, perhaps, in making changes to ensure that logistics, power, and cooling requirements are met before installation.

2. Determine and develop workloads to run on PureApplication System.

   This involves determining which workloads to move to PureApplication System and developing new patterns or using existing ones to deploy the workloads. This step is performed primarily by your staff, although IBM representatives can assist through service engagements or by helping you establish a Center of Excellence (see 3.4, “Structure of a PureApplication Center of Excellence” on page 48).

3. Deploy and manage workload lifecycles.

   After workloads are deployed, these tasks include managing workload lifecycles, monitoring resources that are used by the workloads, and applying maintenance updates to patterns and to deployed workloads.

4. Manage the PureApplication System.

   This set of tasks involves the overall management and monitoring of PureApplication System to ensure that all workloads in the system run smoothly. Management tasks include creating different cloud groups within the system to provide the isolation for different user groups that are using PureApplication System, handling the security authentication and authorization, auditing, applying system maintenance, and so on. Monitoring tasks include reviewing the overall system use, identifying bottlenecks, and making the necessary adjustments.

The remainder of this chapter describes the types of organizational structures that are necessary to successfully adopt PureApplication Systems in an IT organization. This is described in the context of building a Center of Excellence for PureApplication System within your organization. Also included are descriptions of the IT team members’ roles and skills to consider when adopting PureApplication System.

3.2 Organizational principles and practices

Adopting PureApplication System requires changes to your IT organization. These are welcome changes, because many of the challenges that IT organizations typically spend a great deal of effort in addressing either do not exist when you are using an *expert integrated system* or are reduced in scope and effect. The following sections describe particular aspects of how your IT organization is likely to change.

3.2.1 Overall changes

Based on observations, the general principles referred to in this IBM developerWorks article are likely to lead to changes in your organization:

“Aligning organizations to achieve integrated system benefits with IBM PureApplication System”

http://www.ibm.com/developerworks/cloud/library/cl-ps-aim1305_alignorg/

These are the three key changes to expect:

- You are going to need more generalists and fewer specialists.

  This is probably the biggest overall change to your organization. The approach that PureApplication System takes folds all of the major parts of a large-scale, virtualized, distributed system into one package. As a result, you need a split between a larger team of lower-cost generalists who can handle simple deployment and monitoring tasks, and a much smaller team of high-skilled specialists that address specific issues. Tasks that now take teams of specialists days to accomplish can be accomplished by lower-cost, lower-skilled generalists in much less time. Likewise, when problems do occur in your application, the unified console and unified logging and monitoring capabilities make it possible for a single specialist with a higher skill set to debug and solve those problems faster, without having to call in an outside team to deal with the issue.

- Successful integrated organizations implement DevOps.

  A consequence of adopting PureApplication System is that your organizational structure tends to become less compartmentalized and more integrated. This observation is at the heart of a new set of DevOps initiatives from IBM Rational® software and IBM Tivoli® software, called SmartCloud Continuous Delivery. For more information, see DevOps: Continuous delivery of software-driven innovation:

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

  Automated deployment to PureApplication System is a key part of this initiative. But continuous software delivery is not the only benefit of decreased compartmentalization. Through increased collaboration, your organization gets better at planning projects, because there are fewer things that can go wrong when the systems are pre-integrated and optimized with patterns of expertise. Likewise, you find smaller teams working more closely together on problem resolution. Rather than requiring four to six specialists to separately obtain the data, interpret the data, identify the problem, and fix the problem, a smaller number of generalists can perform all of those steps. As part of this change, troubleshooting becomes more about looking at trends and looking for patterns in the data, especially those that derive from implications of a shared infrastructure.

- Everyone does less double-checking to find out whether everything is the same and more to make everything the same from the beginning.

  Today, development teams spend a great amount of effort trying to determine whether all of the various development environments in a software development lifecycle have the same configuration. For a description of the standard environments in a typical development lifecycle, see the developerWorks article titled “IBM WebSphere Developer Technical Journal: The Ideal WebSphere Development Environment:”

  http://www.ibm.com/developerworks/websphere/techjournal/0312_beaton/beaton.html

  Internal IBM studies show that a high percentage of bugs that are found in production are actually bugs that were introduced or reintroduced through configuration changes in test environments but then not conveyed forward to production environments. There are many sources of these changes: operating system patches, middleware fix packs, or simply configuration changes applied to the middleware, such as transaction timeouts or Java

Database Connectivity (JDBC) pool sizes made from the IBM WebSphere Application Server console. The reason that these configurations drift apart over time is that each environment is a long-lived, hard-to-construct assembly of many parts.

After it is built, developers and administrators avoid tearing down an environment and replacing it, because the time and effort that are required in building an entire environment are so extensive. Instead, one of the design principles that underlie PureApplication System is that you put more effort into specifying the templates for your enterprise software configurations (for example, patterns) and less time into managing the instances after they are constructed. If all of the latest configuration information is captured in a pattern, it is simple to re-create a new instance of that pattern. This means that different environments are all built from the same pattern, so they all start from an identical place.

3.2.2 Role changes

Another observation, from our experience with clients, is that certain roles need to change as a result of the benefits of the automation that PureApplication System provides. The reduction in work that we describe in this section enables your team to begin working through existing IT backlogs without increasing your IT budget. The reduced effort that is needed for common tasks also enables your staff to retrain and develop the new skill sets that help you address new business opportunities. These are the areas where you save time and work:

- Less work is needed to rack and stack servers.

  One of the questions that clients ask when they first find out about PureApplication System is “So who assembles it?” In the past, distributed systems were always purchased piecemeal and arrived in multiple boxes that someone had to unpack and then rack, stack, and cable the servers together.

  PureApplication System does not arrive as a large stack of boxes. The entire rack is delivered as a single, pre-assembled, pre-cabled, pre-integrated, and pre-tested system. IBM representatives unpack the system in your data center, connect it to your network and power, and then boot the system, validate that it functions correctly, and configure it to recognize your network — all within four hours. And when you decide that you want a server upgrade to add more processing speed and memory, IBM service representatives do that in about an hour. You do not even have to incur downtime.

- Less expertise is needed for managing storage.

  Before PureApplication System, you needed an entire group that was dedicated to managing the storage area network (SAN) infrastructure that your compute hosts worked from. In addition to setting up the physical parts of the SAN, someone had to handle all of the SAN and logical unit number (LUN) planning and the mapping to connect the SAN infrastructure to your hosts and then to configure the host and volume, host mapping, storage pool management, and so on. Then, after the infrastructure was set up, it had to be constantly cared for and managed to keep up with new driver levels, SAN software levels, and so on.

In PureApplication System, this is all done for you. The only storage tasks that anyone performs during normal operations are to decide on the appropriate volume size when a new VM is provisioned, to expand volume sizes when needed (for example, when a database is close to expanding beyond its originally available volume size), and to perform backup and restore operations.

- Less work is needed to care for your virtualization infrastructure.

  Another major change with PureApplication System from traditional ways of doing things is that you do not need someone to set up your virtualization infrastructure. If you were building a virtualized environment on your own, you would need someone to install

hypervisors, install the virtualization management infrastructure, connect the individual hypervisors to the infrastructure, and then maintain the infrastructure. Again, all of this already works upon delivery of your PureApplication System. The scope of work for the virtualization team is reduced, so they have time to manage more virtualized environments. With the trend of PureApplication System implementations, the team can also take on other IT duties.

- Less work is needed in network management.

  Any modern distributed system is supported by a mass of network cables. There is an enormous amount of planning and work that goes into setting up a LAN, with work required for connecting distributed host computers to a SAN, to each other, and to rack-mounted switches. Then, someone must configure all of those physical networks, assign IP addresses to servers, specify how those map to VLANs, and so on. PureApplication System either eliminates those tasks or reduces the amount of effort that is needed to perform those tasks. There is still some planning that is necessary to determine how you are going to set up the internal network in your PureApplication System. But after you make those decisions, the implementation proceeds quickly, and nearly everything you need to specify is done during the four-hour installation. Likewise, the care and feeding of the internal network is a much simpler operation, requiring far fewer staff hours than the management of a typical network.

3.2.3 Skill set changes

Although the need for certain skill sets is lessened with the adoption of IBM PureApplication System, others become more prominent. So you will need to modify some of the existing roles and add new ones along these lines:

- As the lifetime of systems changes, the types of administrative actions that you perform also change.

  As an example of what people mean when they say “patterns are things where the definition is more important than the implementation,” one change that we find with clients that adopt the WebSphere Application Server patterns in PureApplication System is that the lifetime of a WebSphere Application Server cell changes. This is a ramification of the practice that is described in the developerWorks article titled “Preparing for IBM PureApplication System, Part 2: Is your application ready to become virtual?”

  http://www.ibm.com/developerworks/websphere/library/techarticles/1204_brown/1204_brown.html

  This requires a change in how you view environments. Rather than thinking of them as long-lived assets, they are now transient assets. Your WebSphere Application Server system administrators get used to working on more, smaller cells with shorter lifetimes. In the old way of doing things, a WebSphere Application Server cell was generally created to represent a development environment, and then that same cell typically lasted, in the same form, until it was replaced when a major WebSphere Application Server upgrade was adopted for that project. That time span was usually one to four years.

  For some PureApplication System clients, the average lifetime of a cell drops to a few months (three or fewer in some cases). Therefore, your WebSphere Application Server system administrators (and system administrators of all types of middleware) spend less time modifying live systems in the GUIs of their respective tools and more time writing scripts to execute on the running systems that are pattern instances and creating new instances of those patterns.

- You need people who are good at abstraction to create content for standard patterns.

  So what do the people who used to manage your virtualization infrastructure do, instead? They do the very useful work that helps your business get things done better and faster.

For example, remember the benefits of keeping your different development and test environments in synchronization and keeping them up-to-date with the latest versions of OS patches, and so on? That kind of global planning work requires ongoing effort from people who are experts in building and modifying hypervisor images. PureApplication System comes with the Image Construction and Composition Tool (ICCT) from Tivoli software, which aids administrators with that kind of work.

For more information, see the “Build custom virtual images with the IBM ICCT” blog entry on developerWorks:

https://www.ibm.com/developerworks/mydeveloperworks/blogs/complete_clouddev/entry/icct_cloud_tool?lang=en

As that blog explains, this also applies to the need to centralize other types of content, such as standardized virtual system patterns and virtual application plug-ins. This kind of content creation requires people who are skilled in abstraction to design and who can develop the minimum set of long-lived assets for your organization.

- Everything happens faster, which requires more discipline and coordination in your software development lifecycle (SDLC) processes.

  In a business-as-usual environment, without patterns, constructing a Development, System Integration Test, or User Acceptance Test environment can take weeks or months. With the pattern capability provided by PureApplication System, this entire process can be shortened to days or hours.

  That means that the amount of “wiggle room” in existing build and deployment processes goes away, because any missteps become visible immediately. The manual “desk checking” time that is often used to find and correct process problems is also reduced.

  One place where this becomes very important is in application deployment automation. It does not help your organization’s overall efficiency if the time that it takes to deploy your middleware is reduced to a matter of a few hours, but then it takes days of manual trial-and-error work to deploy your application to that middleware and configure it appropriately. Thus, building automation to improve the efficiency and repeatability of this task is an important part of improving the efficiency of your overall software development lifecycle.

  Another aspect that is just as important is to improve the communications between the team that builds and deploys your patterns and your security team. Again, it does no good if a new environment can be deployed in hours if it takes days or weeks to process the paperwork to open the appropriate ports through your corporate firewalls so that people can use that new environment.

3.3 Roles for PureApplication System implementation

Next, we define a new set of roles that are common to IT organizations that use PureApplication System to maximum advantage. We also describe how existing roles are redefined, stemming from those changes, in these key areas:

- Data center operations
- Application operations
- Finance and compliance
- Content enablement
- Solution development

Note:
These are simply roles, not job descriptions.

It is valid to expect that, in some cases, a single person might carry out multiple roles. In other cases, it might take an entire team to fulfill a single role in an organization. Also, the roles that are defined here relate specifically to management of workloads that are running on PureApplication System. When clients also maintain their traditional infrastructures, their existing roles continue to play a part in that management, of course. Therefore, some of these existing roles might be of only a consulting or secondary nature as they relate to PureApplication System.

3.3.1 Data Center Operations team

This team focuses on the system aspect of PureApplication System, not on the workloads that run on the system. They are responsible for the daily operation of the system and the operation of other systems in the data center. The primary roles that we describe here are directly responsible for interacting with the PureApplication System, but the secondary roles are usually more consultative in nature and might not directly interact with the system.

Primary roles

These are the primary roles for interacting with the PureApplication System:

- PureApplication System Administrator

  The System Administrator owns the system and is responsible for keeping the system up and functioning. This role determines the integration of PureApplication System with existing systems, such as external monitoring, reporting, security, and so on. This person enforces adherence to the organization's IT standards for this integration and uses automation to do that where possible, to limit human error. During the installation and setup phase, this person works with the IBM service representatives who do the onsite setup. The person in this role receives the master passwords for the system that are created as part of the setup process. The System Administrator then uses those security credentials to create other IDs on the system and assign those IDs to other roles.

- Physical Cloud Administrator

  Also referred to as a Physical Cloud Architect, this person configures and manages the physical cloud assets in the PureApplication System. This includes these tasks:

  - Creating new isolated clouds within the system (with the quality of service needed for different internal or external groups that use the system)
  - Managing shared services in the cloud as required by applications
  - Creating IP groups with the VLAN assigned by the Network Administrator
  - Troubleshooting cloud infrastructure
  - Monitoring the resource use on the different clouds
  - Adjusting the resources that are assigned to those clouds, based on the needs of the users of those cloud services
  - Working with the different departments

- Network Administrator

  Network Administrators are responsible for the network components within a WAN or LAN infrastructure. They might plan, design, and deploy networks, plus they might perform these activities:

  - IP address assignment
  - IP address management
  - Routing implementation
  - Switch and network server administration

From a PureApplication System point of view, responsibilities include assigning VLANs for the different IP groups to create the necessary isolation, and then assigning the ports for those VLANS on the top-of-rack switches for data network and for network management. Network administrators work with physical cloud administrators to provide the VLANs and IP groups, as needed, to create clouds and to ensure network compliance with the organization's IT policies. During the initial setup phase of implementing PureApplication System, Network Administrators are consulted often about making decisions regarding what ports to use. However, during the day-to-day work of managing PureApplication system, they are consulted only periodically as the Physical Cloud Administrators need to revisit the allocation of network resources.

- Security Administrator

  Security Administrators ensure that the system is secure, manage the LDAP configuration for user and group authentication, manage users and groups within the system, and provide the necessary authorization to the different functions and tasks of the system. They also make sure that the system complies with the security policies of the organization and that corporate security policies are followed in the implementation of the workloads that run on the system. As with Network Administrators, the Security Administrators are consulted often during the initial setup phase while initial policies are set in place and the initial connection to the corporate security infrastructure is set up. They continue to be active periodically as the day-to-day management begins.

Secondary roles

These are the secondary roles for interacting with the PureApplication System:

- Facility Operations Manager

  The Facility Operations Manager is responsible for the physical maintenance and management of a data center, including these tasks:

  - Managing electrical load
  - Capacity management
  - Managing physical floor load capacity
  - Overseeing heating and cooling to maintain optimum operating temperatures for computing equipment

  With respect to PureApplication System, they work with the IBM Client Technical Professional (CTP) on the initial planning, delivery, placement, and installation of the system and the physical connection to the power grid to ensure that all of the facilities, HVAC, and power requirements are met.

- Storage Administrator

  The Storage Administrator designs, plans, implements, administers, and provides support for SAN storage systems. PureApplication System does not currently use or connect to external SAN storage systems (except through the Network File System), because it contains an internally managed SAN. Therefore, this is primarily a consultative role.

- Data Center Operations

  This group proactively monitors computing systems and the network infrastructure. They provide change management and problem management to manage and correct network issues. With PureApplication System, the Data Center Operations team can monitor Simple Network Management Protocol (SNMP) events sent to an external monitoring tool.

3.3.2 Application Operations team

This team creates and maintains environments for different application development groups by using the racks to deploy and manage their workloads:

- Application Cloud Administrator

  This is a new and emerging role that is specific to PureApplication System. This person creates and manages environment profiles and maintains users who have rights to deploy and manage applications that are running on an Environment profile.

- Middleware Administrator

  This is usually an existing role. This person manages middleware products and environments, including performing new installations, upgrades, maintenance and tuning, monitoring, and capacity planning. This role also provides technical expertise during application deployments and assists in troubleshooting application and platform issues. This role is reduced in scope by PureApplication System, because new installations and upgrades are handled through patterns.

3.3.3 Database Operations team

A physical Database Administrator is responsible for the physical aspects of database administration. In a traditional environment, this includes several tasks:

- Database management system (DBMS) installation
- Configuration
- Patching and upgrades
- Backups, restorations, and refreshes
- Performance optimization
- Maintenance
- Disaster recovery

This role changes when PureApplication System is considered. DBMS installation, configuration, and patching becomes a much simpler operation, especially in cases where you can use IBM DB2 as a service (database as a service, or DBaaS). Even in cases where a DB2 virtual system pattern is created, those tasks become simpler. For DBaaS, the physical Database Administrator becomes responsible for the definition of customer-specific workload standards. Backup and restore processes are also simplified by the built-in support for these aspects that are part of DBaas.

For more information, see the presentation titled, “IBM Workload Deployer Database-as-a-Service (DBaaS) patterns:”

http://publib.boulder.ibm.com/infocenter/ieduasst/v1r1m0/topic/com.ibm.iea.iwd/iwd/3.1/Patterns/WD31_DBaaSPatterns.pdf

3.3.4 Finance and Compliance staff

People in these secondary roles do not directly interact with the system, but they receive information from the Data Center Operations team:

- Business Manager

  This role is responsible for billing management and overall financial responsibility for the machines in the data center. In PureApplication System, this person creates charge-backs from use reports, where needed.

- Audit Manager

  In many organizations, especially financial organizations with fiduciary responsibility that answer to governmental authorities, there are often standard audits of access logs or system use that must be run periodically. The Audit Manager oversees the audit data from the system and acts upon it, as appropriate.

3.3.5 Content Enablement team

This team (often called the Corporate Engineering team) defines and controls the standards that the organization uses for OS, middleware, database, security, and other aspects of system management. This is a preferred practice for reusable pattern content that is to be implemented by a centralized team. This team is the best place for those content creators. Therefore, this team is responsible for creating or certifying shared resources that content developers use, such as patterns, images, OS packages, security packages, and so on.

Primary roles

There are several primary content enablement roles:

- Pattern Engineer

  A Pattern Engineer is responsible for adapting existing PureApplication System patterns to the needs of your customer or for defining new patterns. This is an important new role that defines all of the patterns to be used with PureApplication System, based on the needs of different application groups. It includes these tasks:

  - Defining new virtual system patterns and script packages
  - Deciding how to change existing patterns
  - Integrating the work of third parties into these systems

  This role also includes defining plug-ins for new types of virtual application patterns, where needed. Pattern Engineers implement the preferred practices of pattern development. They make the pattern one that is shared across multiple similar environments and create the transient application information to specify at deployment time. They make the decision about when to use an existing pattern for a given application need or whether application needs require a new pattern. Consulting with the Corporate Security Engineer, they ensure that the corporate standards for OS, security, and middleware are included and satisfied in their patterns.

- Pattern Developer

  Based on the design of the pattern, as defined by the Pattern Engineer, Pattern Developers create new virtual system patterns and script packages, modify existing patterns, and integrate the work of third parties into these systems. This role includes creating plug-ins for new types of virtual application patterns, where needed. This person is also responsible for updating and maintaining these patterns and plug-ins throughout their lifecycles.

- Asset Librarian

  This person manages the catalog of assets (images, plug-ins, script libraries, and patterns) across the organization's set of PureApplication Systems. The Asset Librarian makes sure that the catalog is current and that it contains the correct set of resources for the purpose (development, test, or production) for which each PureApplication System is used. If you follow the practice described in the article about using a standardized version control system for long-lived assets, this role keeps the PureApplication System catalog in sync with the correct versions of the assets that are held in the version control system.

  For more information, see "Preparing for IBM PureApplication System, Part 2: Is your application ready to become virtual?" on developerWorks:

  http://www.ibm.com/developerworks/websphere/library/techarticles/1204_brown/1204_brown.html

- Deployment Automation Developer

  This role is responsible for writing and maintaining automation for application installation and configuring and integrating this into the patterns. The person in this role can write custom automation or can manage the integration by using deployment automation tools,

such as IBM UrbanCode Deploy. Because the tools used for deployment automation need to integrate closely with existing continuous integration processes, this person needs to work with the teams that manage continuous integration. Finally, the Deployment Automation Developer needs to collaborate with the application teams and the patterns developer and the architect as they set up deployment automation that complies with corporate policies for deploying code to different types of middleware.

Secondary roles

There are also several secondary roles for Content Enablement:

- OS Engineer

  In IT environments that use standard virtualization (not PureApplication System), this is a critical role. OS Engineers decide upon and publish the base VMware or other hypervisor images that all application development and other teams use for middleware installation and configuration. With PureApplication System, this role diminishes, because the base images are included and the middleware is preinstalled, with configuration provided by scripts or by policies. Therefore, this role becomes more of a planning and consultative role that involves understanding fix packs, security patches, and third-party agents that are installed by scripts that run in all virtual systems as part of a corporate standard installation or that are added to virtual applications through the IBM Patterns Development Kit. They serve in a consulting role with Pattern Developers to ensure that standards are implemented as part of the pattern.

- Database Engineer

  This role sets corporate guidelines for database implementation and use. These guidelines can be implemented as scripts in virtual systems or as workload standards in DBaaS systems. This person often directs the work of the physical Database Administrator, who is responsible for implementation of those scripts or workload standards.

- Middleware Engineer

  As with the other planner roles, Middleware Engineers set guidelines for which versions of middleware are current. They might also be responsible for planning middleware upgrades. With PureApplication System, this is primarily a consultative role carried out in concert with the Application Operations team.

- Network Engineer

  A Network Engineer designs and implements corporate networks. This role must take into account bandwidth capacity, infrastructure, and security requirements. Network Engineers can also be responsible for planning potential network expansion, which often involves network modeling, along with traffic prediction and management. Because PureApplication System contains many network components that are built in, this role is related to planning the connection between PureApplication System and the existing corporate network. The Network Engineer often directs the work of Network Administrators.

- Corporate Security Engineer

  This role determines security requirements, plans compliance with information security standards, and plans and conducts system vulnerability analyses and risk assessments. In addition, this role plans security systems that are related to networking infrastructure (such as firewalls and related security and network devices), designs public key infrastructures, and guides the implementation of security systems. With respect to PureApplication System, this might include determining the integration of PureApplication System with existing firewall systems, selecting security products for integration with PureApplication System, and validating the compliance of PureApplication System with existing corporate security standards.

- Build Engineer

  This existing role is responsible for setting up and maintaining the build and continuous integration environments for the organization. The Build Engineer needs to work closely with the Deployment Automation Engineer to define the connections between the organization's build and deployment processes and the new PureApplication System environment's deployment capabilities.

3.3.6 Solution development

This team (which might be multiple teams) is responsible for developing, testing, and maintaining applications for deployment either on PureApplication System or on other platforms for which PureApplication System is being used as a development platform.

Application development

The following are the application development roles:

- Application Software Engineer

  This existing role is changed slightly by PureApplication System implementation. Software engineers plan, design, and implement the topology and implementation of application systems. They are responsible for middleware selection and for overall application system design. With PureApplication System, the role changes to enable software engineers to start from a set of common pre-built patterns that describe template application topologies. Therefore, an application engineer might select patterns from a patterns catalog and deploy new instances to an environment. Those instances are provided through an Environment profile. This results in better repeatability between different environments (development, test, and production). Engineers can then focus on the internal design of the application software for which they are responsible.

- Application Security Engineer

  This role is responsible for the design and implementation of security solutions within a single application. This often includes addressing concerns such as login, single sign-on, authentication, and authorization within an application. As with the Application Software Engineer, PureApplication System adoption simplifies this role, because there is a greater potential of reuse of existing design solutions that are captured as repeatable patterns.

- Application Developer

  The responsibilities of this existing role are not significantly changed by PureApplication System. One difference, though, is that when development and test systems are deployed to PureApplication System, developers have a consistent place to test their code. There is another change worth noting if Application Developers in your organization use the WebSphere Application Server administration console (or other middleware user interface) for deployment, and for setting server configurations: Rather than using the console, it is better to specify all configurations as script packages or policies so that consistency can be maintained across all environments.

- Application Tester

  This is another existing role that is little changed by PureApplication System, except that testers work within a test environment that is consistent with development and production environments. One small change is that after a test system is described in PureApplication System, it does not need to remain running during times when active testing is not taking place. It can be "turned off" by stopping the corresponding pattern instance, and then easily and quickly "turned on" any time by restarting the pattern instance. This enables much better overall system use and better resource allocation.

Database development

The following are the database development roles:

- Database Application Developer

  This existing role uses tools from your database vendors and creates application-specific DDLs, queries, stored procedures, and other application information storage and retrieval artifacts. There is little change in this role when you adopt PureApplication System.

- Database Application Administrator

  This existing role administers and tunes client-specific databases, updates existing databases to new DDLs, migrates data, and tunes application-specific queries and stored procedures for optimal performance. There is little change to this role with PureApplication System.

Table 3-1 on page 47 shows which of these roles are high, medium, and low touch in terms of how much of the person's time is spent on supporting PureApplication System:

- High: Once every 1 - 7 days
- Med: Once every 15 - 90 days
- Low: Less than once every 90 days or when needed

Table 3-1 Roles and support times for suggested organizational structures

Role	Primary or consulting	New role for PureApplication	Touch level
Data center operations			
PureApplication System Admin	P	N	High
Physical Cloud Admin	P	Y	Med
Security Admin	C	N	Low
Network admin	C	N	Med (Medium)
Facilities Ops	C	N	Low
Storage Admin	C	N	Low
Data Center Ops (Network Ops Center)	C	N	Med
Application operations			
Application Cloud Admin	P	Y	Med
Middleware Admin	P	N	High
Finance and compliance			
Business Manager	P	N	Low
Audit Manager	P	N	Low
Content enablement			
Pattern Engineer	P	Y	High
OS Engineer	C	N	Low
Network Engineer	C	N	Low
Security Engineer	C	N	Low
Database Engineer	C	N	Low
Middleware Engineer	C	N	Med
Pattern Developer	P	Y	High
Asset Librarian	P	Y	High
Deployment Automation Developer	P	Y	High
Build Engineer	C	N	Med
Solution development			
Application Software Engineer	P	N	High
Application Security Engineer	C	N	Med
Application Developer	P	N	High
Application Tester	P	N	High
Database Application Developer	P	N	High
Database Application Administrator	P	N	High

One of the key aspects of adopting a new technology is that such a major change cannot be accepted by an organization all at once. It takes time to transition roles and responsibilities between parts of an organization, and it takes time to train people to take on new roles. Moreover, with any new technology, there is an inevitable learning curve that an organization must go through to become productive with the new technology.

To help you overcome this learning curve, a standard organizational effort that IT teams can put in place is a Center of Excellence (COE). IBM has a long track record of helping IT teams adopt new technology by introducing Centers of Excellence (beginning with Java, through service-oriented architecture, or SOA). We continue by helping clients establish these centers for PureApplication System.

3.4 Structure of a PureApplication Center of Excellence

Early in the adoption of PureApplication System, the members of your Center of Excellence can take on most of the new roles that are described in 3.3, “Roles for PureApplication System implementation” on page 39. However, a critical principle and a responsibility of a COE is to, in effect, “work itself out of a job,” as opposed to becoming a department that grows larger over time. Instead, the COE focuses on training and equipping other teams to take over many of the tasks that are associated with PureApplication System.

The COE is where new knowledge about PureApplication System is acquired in the organization, and it is responsible for spreading that knowledge to the other teams. For example, Application Development teams need training in what patterns are, how they affect their development, and how to perform automated deployment by using the workload console. Systems Administrators on the Operations team need training in how to read the logs and graphs in the system console.

IBM representatives can help by providing initial individualized and large-group training for these roles, but the day-to-day support needs to come from the COE. After the key work of setting up policies for managing and governing assets and then training the various teams in what those policies are and how they are implemented in everyday operations is complete, the COE can begin to wind down to only the core Content Enablement aspect.

In general, the COE works best when split into two teams:

- Content team
- Infrastructure team

This follows the overall split of PureApplication System into the system console and workload console. It is also based on a natural division of labor that most organizations already follow.

3.4.1 Task division for Center of Excellence subteams

The Content team is responsible for managing and maintaining the patterns catalog. That includes obtaining patterns and new pattern versions from IBM, creating and maintaining script packages, creating customer-specific pattern versions, and maintaining all pattern versions. They are also involved in helping Operations maintain existing pattern instances. They work almost exclusively in the workload console. The Content team is also responsible for governance of the shared intangible assets (patterns, images, and script packages).

The Infrastructure subteam is responsible for configuration and management of the overall rack and its internal infrastructure. That includes both the physical cloud infrastructure elements and the virtual cloud infrastructure elements described earlier. This team is

responsible for monitoring the PureApplication System infrastructure, managing security policies, and handling backups and restores, capacity management and management of the shared physical resources.

Considering the preferred practices that we describe in this chapter and those in the article cited at the end of this section, the centralized Content team is responsible for building all virtual system patterns. Likewise, if any new virtual application pattern components are to be built by using the Patterns Development Kit, it is best that this team is responsible for that development. Given this set of responsibilities, we advise that the Content team include at least a small number of experts in each discipline who are responsible for building and maintaining the standard patterns for each technology. For example, this might include experts for middleware, master data management (MDM), business process management, and so on). They do so with input from the application development team, which provides requirements for the new patterns. The Content team functions best with help and guidance from IBM experts. This is especially true during the startup phase of the COE, where they can provide specific training in PureApplication System pattern development.

For more information, see “Best practices for patterns adoption in IBM PureApplication System” on developerWorks.

http://www.ibm.com/developerworks/websphere/techjournal/1307_brown1/1307_brown1.html

3.4.2 Moving to round-the-clock operations

In most cases, the COE needs to take on many of the operations roles until the Operations team learns how to carry out those functions. Therefore, the COE needs to provide training for the Operations team on what happens day-to-day, such as these processes:

- Deploying production instances
- Monitoring the pattern instances and infrastructure
- Applying maintenance updates
- Handling backup and restore operations
- Overseeing governance
- Managing system capacity

As part of the transition plan, the COE needs to set up specific escalation procedures so that the Operations team knows when to contact the COE for incident resolution. Likewise, the COE needs to set up asset governance policies for implementation by the Operations team. The COE also defines the changes to the Information Technology Infrastructure Library (ITIL) processes for the Operations team to follow.

An initial step toward the new approach is to adopt procedures to ensure that transient assets are deprovisioned and returned at appropriate times. These procedures reinforce the concept that environments are not long-lived. This enables the use of a self-service model for the development and test systems through which application development teams can deploy new pattern instances of standard patterns. They can use these instances freely within the bounds of the policies set by the Patterns team regarding what can be changed, how application artifacts are managed and packaged, and how long instances can be kept in place without being used.

Likewise, when the Patterns team develops a new version of a standard pattern, the Operations team can then put a time limit on how long the previous version is to be supported in development and test environments. It is the responsibility of the Application Development team to provision the new version of the pattern and to automatically deploy application resources (EAR files, configuration files, and so on) for the new version within that time limit of support.

A final step is to automate these processes. When the governance processes are established and automated, so that they can be efficiently carried out by the Applications Development and Operations teams, this enables the COE infrastructure team to plan a graceful exit from the stage. Similarly, the Patterns team can decrease in size over time, as fewer new patterns are built or customized. This happens after the Applications Development teams move to adopting off-the-shelf patterns or to reusing existing customized patterns.

4

Workload practices for patterns

The pattern-based deployment approach of IBM PureApplication System has changed the relationship between system development and operations and the roles within those teams. Organizational and operational practices must be adopted to maximize the potential of PureApplication System and to realize the efficiency and total cost of ownership benefits that are possible.

This chapter outlines the preferred practices to help you be successful in integrating PureApplication System into your IT governance policies and data center management strategies. It covers the following practices:

- Browse the PureSystems Centre for relevant assets
- Use a virtual application pattern to add new applications
- Migrate the Java application before you use patterns
- Use images and patterns as delivered when possible
- Decide which tool to use if you customize virtual images
- Split patterns along key interfaces
- Separate the configuration and application from topology
- Automate your lifecycle steps
- Think of environments as short-lived assets
- Develop and test script packages on the target platform
- Set the time zone
- Pass passwords as parameters in script files

4.1 Browse the PureSystems Centre for relevant assets

In the PureSystems Centre, you can access patterns from IBM and IBM Business Partners, updates to systems and patterns, and expertise for maximizing the benefit of systems and patterns. PureSystems *patterns* are part of a broad portfolio of solutions that accelerate deployment and simplify management for cloud, business, and infrastructure applications. Most of the patterns and solutions released by IBM or partners are available through the PureSystems Centre:

https://www-304.ibm.com/software/brandcatalog/puresystems/centre/

4.2 Use a virtual application pattern to add new applications

When designing a new application, consider using a *virtual application pattern* (VAP). These included patterns provide additional capabilities, such as more granular control over elasticity and built-in monitoring. A VAP enables you to take advantage of an existing supported OS and middleware layer so that you can focus only on the application that you are deploying.

Figure 4-1 on page 53 shows the design palette for building a VAP. You provide only the application enterprise archive (EAR) file, database Data Definition Language (DDL) file, and LDAP details to deploy. You can add policies to control the runtime behavior. When you deploy this pattern, the system provisions environments for the application server, database, and connections to the defined user registry.

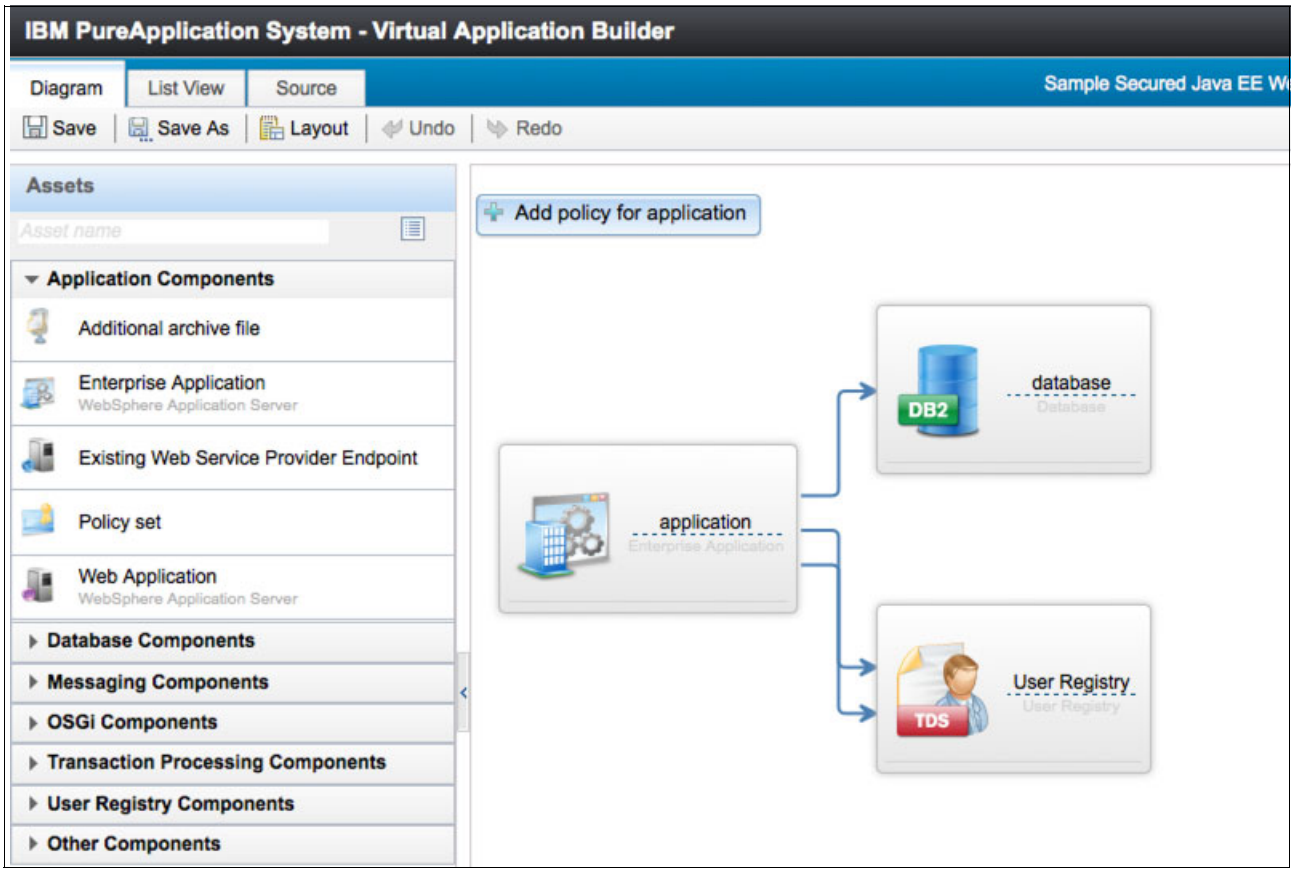

Figure 4-1 VAP design palette

No knowledge of the underlying OS or middleware configuration is required. There are two advantages to this approach:

- It minimizes the number of unique environments that need to be managed to keep the focus on the individual applications.
- It leads to a shorter development and deployment cycle, resulting in a faster time to market, which ultimately lowers costs.

Developers can use the IBM Workload Plug-in Development Kit (PDK) to build the components that they need to build virtual application patterns. It is available at no charge:

`http://ibm.co/1iObTxI`

The IBM video about Virtual Application Patterns in PureApplication System is also helpful:

`http://www.youtube.com/watch?v=ysi_asvzcxw`

Figuring out how to best meet your organization's needs requires thought and planning, of course.

4.3 Migrate the Java application before you use patterns

Although their inherent simplicity and ease of use make virtual application patterns attractive, when you are working with an existing Java Platform, Enterprise Edition (Java EE) application, keep the following factors in mind:

- Ensure that the application runs on a version of middleware that PureApplication System supports.
- If you are running Java EE on IBM WebSphere Application Server Version 6.1, be aware that V6.1 reaches end of service (EOS) soon.
  - WebSphere Application Server 6.1 is not available or supported on PureApplication System installations.
  - You must migrate Java EE to WebSphere Application Server V7.0, V8.0, or V8.5.
  - Using the WebSphere Application Migration Toolkit helps when migrating Java EE applications to a newer version of WebSphere Application Server. It quickly identifies problematic code in the application.

  For more information, see the IBM WebSphere Application Migration Toolkit web page:

  `http://www.ibm.com/developerworks/websphere/downloads/migtoolkit/index.html`

4.4 Use images and patterns as delivered when possible

Avoid unnecessary customizing, because IBM releases fixes and updates that can be applied directly only to as-delivered materials. Therefore, use the provided WebSphere Application Server Hypervisor Edition images, rather than build your own. Unnecessary customization of images can fill disk space and create confusion when selecting the correct image to use to build a pattern.

Also, remember that every time a new image is created, it must be maintained and managed as an individual unit. This means that OS, middleware, and other patches must be applied

directly to the image. This has an impact on overall management overhead, so there is a trade-off when considering when to extend an existing image. Does reducing deployment time outweigh the increase in management overhead?

Before customizing an image by using Extend Capture functionality or using the Image Construction Composition Tool (ICCT), consider whether you can do what you want to do by using a script. It is best to write scripts in a way that they fetch the necessary components from a centralized repository when the script runs.

4.5 Decide which tool to use if you customize virtual images

Although the guidance in 4.4, "Use images and patterns as delivered when possible" on page 53 discourages the creation of new images, you might occasionally need to customize a virtual image.

The ICCT uses Extend and Capture. However, the tool provides the following additional capabilities that are not available with Extend and Capture:

- Visibility: Keeps a record of the software bundles that you have added
- Reuse: Same bundle can be added to multiple images
- Variability: Ability to add *deploy-time parameters* directly to an image
- Repeatability: Same customizations can be reproduced with the touch of a button
- Prerequisite checking: Up-front checks for OS and software compatibility
- Portability: Same images can be constructed across multiple clouds
- Usability: Start with existing images
  - Use your own customized OS
  - Your images need to be supported by PureApplication System
- Community content. Sample bundles available

Given the advantages, if you cannot use scripts, it is better to use the Image Construction Composition Tool if you need to customize the images that are included with PureApplication System.

4.6 Split patterns along key interfaces

A common question is: "How big does a pattern need to be, and what does the scope of applications need to be?" For example, must all applications be contained within a single pattern, or can an application spread across patterns? In general, there is no single answer to these questions, because the term *application* means different things to different people and organizations. In some organizations, a single application might be only a single EAR file running on an application server. In other organizations, a single application might spread over multiple run times, including database runtimes, application server runtimes, portal server runtimes, and enterprise service bus (ESB) runtimes.

So instead, the questions need to be rephrased to, "How do you determine the correct scope of a virtual system pattern?" and "How do you determine the correct scope of a virtual application pattern?" For these more tightly defined questions, there are basic principles that you can apply.

Essentially, a single pattern must contain only components that are directly related to each other. It cannot contain components that are shared by any components outside of the pattern unless all of those components are shared.

Take the example of two applications: a Web Banking application and a Wealth Management application. They each have EAR files that are deployed to application servers. They both access some common information in a shared Customer Accounts database. However, the Wealth Management application also relies on information from a Portfolios database that the Web Banking application does not use. In this case, it is better to divide these applications into three patterns, along the following lines:

- Pattern 1: Web Banking application servers and web servers
- Pattern 2: Wealth Management application servers, web servers, and Portfolios database
- Pattern 3: Customer Accounts database

Now, any changes that happen to the individual applications are localized to the specific patterns that are affected by those changes. If a change needs to be made to the Web Banking application, the Wealth Management application continues to function unhindered, and vice versa.

4.7 Separate the configuration and application from topology

When you build a *virtual system pattern* (VSP), you are essentially describing a general topology where you can deploy transient application assets. If you build a clustered WebSphere Application Server pattern that connects to a stand-alone instance of IBM DB2 Enterprise Server Edition, the pattern can be used for many different applications that require that same topology. What differs for each particular application is detailed application configuration information, such as Java virtual machine (JVM) configuration (heap size, and so on) and critical application assets, such as EAR files and DDL. In a large organization with many applications, if each application builds its own pattern, the organization ends up with a proliferation of nearly identical patterns in the library. From an asset management perspective, it becomes challenging to manage, maintain, and track all of those patterns.

Therefore, it is best that the VSPs not include purely application-centric information in the scripts that they run. Instead, the scripts can read in that kind of specific information when they run during the deployment process, as configuration from a shared repository of information or from instance-specific locations that are specified as part of the deployment process. This encourages reuse of each virtual system pattern and reduces the total number of virtual system patterns that must be managed. This also fits with the tooling and policies that most development and operations teams already have in place for handling application builds and automated deployment. It then becomes a matter of taking the tools that the IT department is already productive with and integrating them into the PureApplication System structure.

PureApplication System includes the Advanced Middleware Configuration tool, which can be used to separate configuration information from the specific deployments. For more information, see the IBM developerWorks article titled “Preparing for IBM PureApplication System, Part 4: Onboarding applications to the cloud using the Advanced Middleware Configuration tool V1.1:”

http://www.ibm.com/developerworks/websphere/library/techarticles/1204_akeley/1204_akeley.html

Another possibility is to use IBM UrbanCode Deploy to automate artifact deployment to your pattern instances. For more information, see the UrbanCode Deploy web page:

http://www.urbancode.com/html/products/deploy/

4.8 Automate your lifecycle steps

PureApplication System contains two broad classes of assets: transient assets and shared assets. Transient assets, in particular virtual system pattern instances, have a specific lifecycle that is managed by an operations team (in production), or by a development team (in development), or by a combined development and operations teams in both environments. Most of the steps in this lifecycle can be automated. Consider the example of the lifecycle of a virtual system pattern instance shown in Figure 4-2.

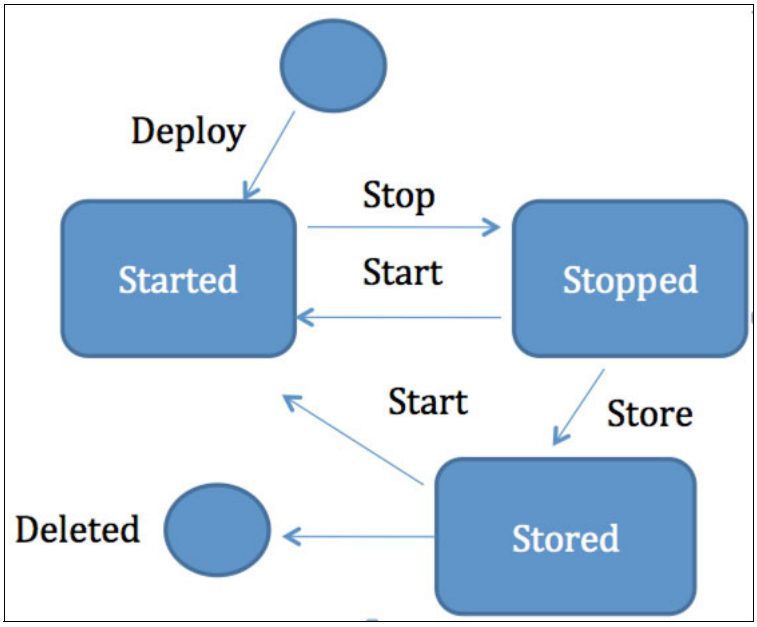

Figure 4-2 Lifecycle of a virtual system pattern instance

Although it does not consider the possibility of patching a running virtual system instance, in this simple example, a newly provisioned pattern instance (for example, in a development and test environment begins in the *started* state). At any time, an administrator can stop the virtual system instance. This does not release the reserved resources in the cloud, but does terminate the virtual machines that make up the virtual system. The administrator can then *store* the virtual system, which does release the resources but keeps the virtual system instance available for redeployment at any time. All of these functions are available through the **Instances** → **Virtual Systems** menu in the Workload console.

An IT organization must set up policies regarding how long an instance is to stay in each of these states and under what conditions the instance moves from one state to another. If an instance is not used for a period of time (which can be verified by referring to the HTTP server's web logs), it can be stopped. It can remain in that stopped state for a fixed period of time before the owner restarts it, or it can be moved to the stored state. Finally, if an instance is in the stored state and has not been started for some time, it can be deleted.

After these policies are established and validated over time, it is best to automate the policies. This guarantees that they are enforced consistently and that the shared resources of the PureApplication System are used efficiently. The PureApplication System command-line interface (CLI) and REST API contain functions that you can use to query the state of a virtual system pattern instance and to start, stop, store, or delete that instance. You can either write scripts to automate these functions and implement the defined policies, or you can write programs that manage this through the REST interfaces.

The CLI and REST interfaces also contain functions to automate the patching and fix application processes for virtual system patterns. That is a more complex case, so you must handle it with care. In many cases, it is better to simply replace the instance with another one at the newer patch level.

4.9 Think of environments as short-lived assets

Many existing IBM clients have WebSphere Application Server cells and other system environments with long product lifetimes. Usually, it takes a major version upgrade or something of similar magnitude to motivate a team to decommission an environment and create a new one. These are often long and complex processes in conventional environments. However, PureApplication System makes the creation of a new environment a fast and painless process. This leads to a major change in the way that environments are viewed, based upon the amount of effort that is required to create or re-create the environment.

If all of the important configuration information is stored in a pattern, as opposed to only in the configuration files of the cell, it is easier to simply re-create the cell for any significant change to the configuration. This is true even when you are merely adopting a new point release of WebSphere Application Server or other middleware. In PureApplication System, the two cells can even run side-by-side for a few days or weeks until it is apparent that the new cell is functioning normally. At that point, the old cell can be deleted, with the assurance that it can be re-created at any time if necessary. Another change that we have seen is that clients were building only a few, large cells because of the complexity of creating the cells. However, when this can be done simply from a pattern, it becomes possible for each project to be contained within its own cell, which results in smaller cells and more of them.

The major change to the way in which operations teams work is that when they adopt this pattern, they must move all administrative aspects into scripts. Although you can perform an administrative action, such as setting a tuning parameter or installing an application, in the console for that middleware (the WebSphere Application Server console, for example), it is best to also create a script that performs that same action and attach that script to the virtual system pattern definition. This enables the re-creation of the environment exactly at any time and ensures that you can quickly adapt to both planned and unplanned failures.

4.10 Develop and test script packages on the target platform

If you write any customization scripts, write them on the target platform. Even Linux and IBM AIX® platforms, the two that PureApplication System supports, have subtle differences in the way that the shell behaves. Also, some of the utilities that work on Linux might not be available on AIX.

Another common problem is scripts that were written in Notepad on a Microsoft Windows platform. When you upload the script to Linux, it adds control characters at the end, and that causes scripts to fail. You can use utilities (such as dos2unix) to convert those scripts.

In terms of debugging, add as much debugging as possible, because that is the only way of figuring out problems. You can usually view log files in the PureApplication System console, as shown in Figure 4-3 on page 58.

Figure 4-3 Log files

Another thing to keep in mind is that any scripts that are added to a pattern are executed as *root* user on the virtual machines. This might cause problems, because middleware such as WebSphere Application Server runs as non-root user, so any directories that have been touched by the script that is running under *root* user are not accessible. Therefore, the preferred practice is to switch the user from within your shell script when necessary, as Example 4-1 shows.

Example 4-1 Switch user command

```
su virtuser -c "./nextShellScript.sh"
```

4.11 Set the time zone

You can set your local time zone in PureApplication System by selecting **System Console** → **System** → **Settings**. However, any virtual machines that are deployed as part of a pattern deployment are usually set to use UTC as the time zone. This can cause confusion in some cases, especially when you are trying to correlate logs with systems running outside of PureApplication System. Use the simple script shown in Example 4-2 to force the deployed VM to use the same time zone as PureApplication System.

Example 4-2 Simple script to change time zone

```
#!/bin/sh
echo "export TZ=CET-1CEST-2,M3.5.0/02:00:00,M10.5.0/03:00:00" >
/etc/profile.d/TZ.sh
```

4.12 Pass passwords as parameters in script files

There are several occasions when you need to externalize parameters that are used during the execution of scripts when you need them to customize a pattern deployment. By default, before you run a script package in your virtual machine environment, the script package parameters are written automatically to the `/etc/virtualimage.properties` file. If you want to manually run the script again later, you must use the `/etc/virtualimage.properties` file as the source to repopulate your environment with the parameters that the script needs to run successfully.

This is not a good practice. The environment properties that are written to the `/etc/virtualimage.properties` file must be only those that are defined or provided by the virtual image. Design your script packages so that they do not depend on those parameters being available in this file.

However, there might be other situations when you do not want the script package parameters to be written to the `/etc/virtualimage.properties` file. For example, if your script package parameters include password parameters, such as parameters that contain the text

string password in the name or parameters of the password type, the values for these password parameters are written to `virtualimage.properties` file in clear text. That might be a security concern in your environment.

As a preferred practice, in most situations, prevent script package parameters from being written to the `/etc/virtualimage.properties` file by including the `envonly` (environment only) attribute in the `extendedattributes.json` file in your script package.

Define the `envonly` parameter in the `extendedattributes.json` file, as shown in Figure 4-4.

```
{
        "envonly": "true"
```

Figure 4-4 Setting the envonly parameter

If the `envonly` value is set to `true`, script package parameter values are set only in the environment and are not written to the `/etc/virtualimage.properties` file. This is the preferred setting. If the value is set to `false`, script package parameter values are set in the environment and are written to the `/etc/virtualimage.properties` file. This is the default value. If this file is not included in your script package, the default value of `false` is assumed.

For more information, see the IBM developerWorks article titled “Best practices for patterns adoption in IBM PureApplication System:”

`http://www.ibm.com/developerworks/websphere/techjournal/1307_brown1/1307_brown1.html`

5

High availability of the system

High availability (HA) in IBM PureApplication System is divided into two aspects: hardware redundancy and software configuration. Hardware redundancy provides the ability to have another component of the same type take over the work. The configuration of the software is also taken into account for this redundancy. If the software running on the system detects a failure, the partner component can take over the work. PureApplication System configures the hardware and hypervisor in HA pairs to maintain redundancy in the system.

This chapter focuses on how PureApplication System helps provide highly available clouds and how patterns take advantage of this configuration. The guidance also considers the placement algorithm that is used in PureApplication System in addition to the different types of models.

This chapter covers the following topics:

- Cloud management
- Placement algorithm
- Cloud configuration

5.1 Cloud management

To make a cloud highly available, the System Administrator needs to include at least two compute nodes in the cloud. This initial step provides hardware redundancy from a compute node perspective, in addition to specific configuration driven into the virtualization components to make them highly available and able to recognize their partners.

One more piece of information to keep in mind is the PureApplication System type. A mini system includes a single chassis, but enterprise systems include three chassis. In an enterprise system, you can ensure high availability by configuring the cloud to span these chassis in case there is a chassis failure.

The other component that you must understand is the placement algorithm use in PureApplication System for resource use. The processor and memory portion of the algorithm is on a cloud basis and takes into account the number of compute nodes that are assigned on each cloud. It is also model-specific, so Intel and IBM PowerVM® systems behave differently in terms of resource reservation and allocation.

5.2 Placement algorithm

In PureApplication System, the placement algorithm ensures that there are sufficient resources during deployments by starting and stopping virtual machines.

When you deploy a pattern into a cloud group, the system ensures that sufficient memory and processor resources exist for each virtual machine in the pattern. To calculate the total amount of memory and processor resources that are available for new deployments, the following resources are subtracted:

- The memory and processor resources that are used by the hypervisor
- The memory and processor resources that are reserved for migrating virtual machines when a compute node is suspended
- The memory and processor resources used by virtual machines on the compute node

The remaining amount of memory and processor resources is available for new deployments. This amount might differ from the amount of memory and processor use of a compute node.

Specific amounts of memory and processor resources are reserved for migrating virtual machines when a compute node is suspended. The reserved memory and processor resources are not available for new deployments, but they might be used when a compute node is suspended, such as when a compute node is removed from a cloud group or is placed into maintenance mode. The amount of memory and processor resources that are reserved on a compute node is based on the number of compute nodes within the cloud group. If there is a single compute node in a cloud group, no memory and processor resources are reserved. If there are two or more compute nodes in a cloud group, the amount of memory and processor resources for a single compute node are reserved and divided evenly among all of the compute nodes in the cloud group, minus the memory and processor resources that are used by the hypervisor.

On an Intel system, the calculations that follow are taken into consideration. Use these guidelines to calculate the amount of memory and the cores.

Calculations: The numbers that are used in these calculations are based on a fully populated compute node. The total amount of memory and the number of processor cores are the values that are returned by VMware.

To calculate the amount of memory available for deployments, in MB, use the following examples. For these calculations, we used the lowest value of 262,116.

- For one compute node:
  `262,116 - 6,144`

- For more than one compute node:
  `(262,116 - 6,144) * ((#_compute_nodes - 1) / #_compute_nodes)`

Reserved memory: 6 GB of memory are reserved for VMware.

To calculate the number of cores available for deployments, use the following examples:

- For one compute node:
  `16 - 1.6`

- For more than one compute node:
  `(16 - 1.6) * ((#_compute_nodes - 1) / #_compute_nodes)`

Reserved cores: 10% of the cores are reserved for VMware.

For each VM that is deployed, VMware allocates a certain amount of overhead in addition to the amount of memory that is used by the VM. Based on various data points, you can calculate the amount of overhead, in MB, by using this example:

`13.87 + (4.5 * #_virtual_cpus) + (0.0075 * virtual_memory_in_MB)`

You might need to adjust the calculations, based on new data points.

On an IBM PowerVM system, take the calculations that follow into consideration.

Calculations: The numbers that are used in these calculations are based on a fully populated compute node. The total amount of memory and the number of processor cores are the values that are returned by IBM Systems Director VMControl™.

To calculate the amount of memory available for deployments, in MB, follow these steps. This example uses a value of 524,288 MB for the memory per compute node:

Start with the memory per compute node, and calculate 6% for memory overhead:

`524,288 * .06 = 31,457`

To calculate the amount of overhead, round the result up to the nearest 512 MB value:

- `31,457 / 512 = 61.44`
- `62 * 512 = 31,744`

Calculate the amount of available memory, in MB:

`524,288 - 31,744 = 492,544`

Memory overhead: The specified memory overhead of 6% is an estimated value, which is based on a typical workload. This value varies based on factors such as the number of I/O devices, memory fragmentation, and the number of virtual machines.

For each VM that is deployed, an overhead of 12.5% is allocated for the processor. That is, 28 physical processors are available for VM deployment.

5.3 Cloud configuration

It is important to keep the model and type of the system in mind when you are making decisions about cloud environments. If high availability is important, the number of chassis and number of potential compute nodes in the model are also important factors to consider. The following tables describe the configuration, based on the model and type, the compute node, and the core count.

We suggest a minimum of three compute nodes for each cloud and no more than three clouds in a system. For compute nodes, that gives you extra safety. If one compute node fails, there is redundancy in the system.

Table 5-1 shows the cloud:compute node suggestions for mini systems.

Table 5-1 Cloud compute node suggestions for mini systems

Number of clouds	Type (brand)	Number of compute nodes in system	Number of cores	Number of compute nodes per cloud
1	Intel	2	32	2
1 2 (not advisable)	Intel	4	64	4 2
1 2	Intel	6	96	6
1 2	Intel	8	128	8 4
1	IBM Power	2	32	2
1	IBM Power	2	64	2
1	IBM Power	3	96	3
1 2	IBM Power	4	128	4 2

Table 5-2 shows the cloud:compute node suggestions for enterprise systems.

Table 5-2 Cloud to compute node suggestions for enterprise systems

Number of clouds	Type (brand)	Number of compute nodes in system	Number of cores	Number of compute nodes per cloud
1 2 3	Intel	6	96	6 3 2
1 2 3	Intel	8	128	8 4 3, 3, 2
1 2 3	Intel	10	160	10 5 3, 3, 4
1 2 3	Intel	12	192	12 6 4
1 2 3	Intel	14	224	14 7 5, 5, 4
1 2 3	Intel	24	384	24 12 8
1 2 3	Intel	38	608	38 19 12, 12, 14
1	IBM PowerVM	3	96	3
1 2	IBM PowerVM	4	128	4 2
1 2	IBM PowerVM	5	160	5 3, 2
1 2 3	IBM PowerVM	6	192	6 3 2
1 2 3	IBM PowerVM	7	224	7 3, 4 3, 2, 2
1 2 3	IBM PowerVM	12	384	12 6 4
1 2 3	IBM PowerVM	19	608	19 10, 9 6, 6, 7

6

Lifecycle and maintenance

This chapter focuses on IBM PureApplication System maintenance and how to plan before, during, and after applying fixes to the system. It explains the built-in expertise that manages updates through an easier, automated workflow. The information includes links to details about what is updated when fixes are applied to the system, so that there are no surprises.

System and workload updates are covered in the following topics:

- IBM PureApplication System maintenance
- System maintenance
- Workload maintenance

6.1 IBM PureApplication System maintenance

Maintaining a system involves coordinating multiple roles and actions to avoid major issues and outages of running applications. The pieces of the system must be updated periodically, and that requires different actions to keep the system running properly. Maintaining those pieces requires a close review of dependences to ensure that there are no surprises.

6.1.1 System components

The following components make up the system:

- Management software:
  - IBM PureSystem Management
  - Intel VirtualSystem Management or IBM Power FlexibleSystem Management
- Hardware firmware:
  - Chassis
  - Network
  - Storage
  - Compute nodes

6.1.2 Automated maintenance frameworks

IBM PureApplication System automated maintenance is based on two frameworks for update processes to provide the required flexibility. The frameworks can be broken down into two scenarios:

- System
- Workload

You can look at PureApplication System as an appliance for the *system* scenario of the maintenance. It provides built-in expertise that helps reduce the manual work that updates normally involve.

PureApplication System treats each pattern and pattern instance as a *workload* scenario to help provide a separate management lifecycle to the deployed applications and future deployments.

The workload consists of the following components:

- Deployed VMs:
  - Operating system
  - Middleware
  - Application
  - Data
- Virtual application patterns
- Virtual system patterns
- Virtual appliance patterns
- Database patterns

The PureSystems Centre offers a convenient way to get optimized content, fixes, updates, and access to IBM and IBM Business Partner expertise:

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

6.1.3 Client responsibilities

As the client and owner of the system, it is your responsibility to configure backup tools for the entire machine, the artifacts in the catalog, and the data within the virtualized environments. You can use traditional backup tools and techniques, such as those offered through IBM Tivoli Storage Manager to back up your IBM DB2 database. It is also your responsibility to deploy the data backup agents by using a custom script package that is configured for each instance, to offload a copy of the backup to remote storage, and to ensure proper functioning.

System-wide backup is covered in 2.9 and 9.5 of *Adopting IBM PureApplication System V1.0*, SG24-81130:

http://www.redbooks.ibm.com/Redbooks.nsf/RedbookAbstracts/sg248113.html?Open

For more information about system-wide backup, see “Backing up the system” in the IBM PureApplication System W1700 Information Center:

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

Another preferred practice is inventory management. Be sure to answer these questions:

- Which instances are running?
- Who owns each deployment or catalog component?
- Does upgrading or updating a running instance require a formal change process?

Some of these items are part of traditional asset management, which is not an inherent part of IBM PureApplication System. Therefore, asset management is not covered in this book. However, IT asset management can be part of your larger infrastructure environment. For information, see these IBM Redbooks publications:

- *Deployment Guide Series: Tivoli IT Asset Management Portfolio*, SG24-7602

  http://www.redbooks.ibm.com/abstracts/sg247602.html
- *IT Asset Management Processes using Tivoli Asset Manager for IT*, SG24-7601

  http://www.redbooks.ibm.com/abstracts/sg247601.html

6.2 System maintenance

The system is treated as an appliance. Therefore, there is a single package that can update all of the software and firmware that make up the system. This single package is referred to as a *fix pack*. IBM sends a new fix pack periodically to make sure that all of the hardware firmware and management software is kept up-to-date and to provide new features and improved stability.

There are also emergency fix patches that can be applied, similar to a fix pack, to update a smaller group of the system in a coordinated fashion.

The system has another package called a *group fix*, which contains the internal components of the system workload pieces, such as monitoring, Advanced Middleware Configuration for PureApplication System, Java patterns, and so on. This group fix provides the recommended workload fixes as a base for other workloads to use.

Currently, an IBM service representative comes to your site and applies the system fix pack. In the future, you will be able to load the system fix packs and apply them to your system. However, these same practices still apply no matter who applies the system fix pack.

Before applying the fix pack, review what is new in the fix pack level in this information center web page, so that you understand what is going to happen when the system is updated.

For more information, see “What’s new in this release” in the IBM PureApplication System W1700 Information Center:

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

“What’s new in this release” explains the new features that the fix pack will add to the system. These fixes do not change what you have done previously with the system, but they can help simplify a particular action or provide more capabilities. For example, in PureApplication System Version 1.1.0.1, there is a new feature in system monitoring called the *External System Monitoring service*. This feature allows you to push the workload monitoring agent data to an external IBM Tivoli Monitoring data center. It makes integration of the PureApplication System with the data center tighter and smoother.

Reviewing the details on that web page helps ensure that you do not encounter surprises with the new features, so that you can plan for those changes.

Before applying any of them, be sure to review the fix list and release notes for each fix pack on the *Related information for IBM PureApplication System recommended fixes* web page:

http://www.ibm.com/support/docview.wss?uid=swg27039159

The fix list and release notes provide details about what was fixed, not the new features. Most of these are not very exciting, because they make something that was broken work. But there can be a few items that correct something that worked one way but now works differently, as it was designed to work. These fixes are the surprise items for many people, because the PureApplication system worked for them one way and, after applying the upgrade, it stopped working. That is why it is so important to review the fix list and release notes before proceeding, so you know what to expect.

For example, a REST API that someone found in the PureApplication System, which was not documented in the information center, changed where security restricted the call that was open for others. This was a good fix that secured the system, but when someone wrote a script based on that REST API and then applied the fix, the script no longer worked. That caused frustration and required problem-solving time. So be sure to review the list of fixes to see whether any of the items changed in areas that you might have discovered and used.

After the upgrade, review the individual levels of the system to verify that they were all updated. To find the individual piece levels that make up the system, use the System Console, and select the **Setting menu** from the **System** drop-down menu. On the Setting page, find the System Maintenance Report section, which lists all of the levels. You can download this list as a CSV file so that you can sort it and mark the correct levels.

Next, check the expected levels of the individual pieces that make up the system for the specific PureApplication System fix pack level on the Supported firmware and software in PureApplication System web page:

http://www.ibm.com/support/docview.wss?uid=swg27038152

Doing this comparison verifies that the update properly updated all of the pieces to the correct levels and did not skip anything.

6.3 Workload maintenance

Workload maintenance has a packaging release that is similar to the system maintenance, where each workload product, hypervisor image, or pattern type, has a fix pack release. A group fix can also be released. It is a list of recommended fixes for that specific pattern's middleware or OS. The workload maintenance can also have emergency fixes to patch items quickly.

The major difference to remember is that workload maintenance gives you more control of which new deployments of a pattern or existing pattern instances get the updates. With this control, you can move new deployments to the newer updates, yet let existing pattern instances roll forward based on their schedules.

With PureApplication System, the system is preloaded and pre-entitled to specific software. For each system fix pack, IBM publishes the fix levels for the fixes that are released. This explains what new levels of pre-entitled content are available for your workloads. It also helps you verify whether your system is up-to-date on a specific workload that is included with the system. Check the PureSystem Centre for the latest updates for all other workloads that you added to the system.

For more information, see the System maintenance and pre-entitled component updates on the PureApplication System Version 1.1.0.1 web page:

http://www.ibm.com/support/docview.wss?uid=swg27039773

6.3.1 Maintaining and updating catalog components

Important: The system administrator's job can become overly complex by allowing too many people to generate too many catalog components. It is best to keep creation rights and the number of different patterns under control. Consider adding different configuration parameters between the different environments. At deployment time, you can use the same deployment scripts and rely on an internal web distribution server to retrieve the configuration data for a particular environment.

Emergency fixes

Virtual system instances can require maintenance to the product code and the operating system. Rather than accessing each virtual machine, you can apply interim fixes and fix packs directly from PureApplication System. Download the fixes from the IBM PureSystems Centre, and then upload them to the catalog. When you start a virtual system instance, you can apply the emergency fixes to all of the virtual machines in the virtual system instance directly from the console.

Emergency fixes are used to update the product binaries of deployed instances. Emergency fixes can be individual fixes or a group of fixes that are associated with particular virtual system or virtual application types. Use the workload console to complete this task.

For more information, see "Adding emergency fixes to the catalog" in the IBM PureApplication System W1700 Information Center:

http://pic.dhe.ibm.com/infocenter/psappsys/v1r0m0/index.jsp?topic=%2Fcom.ibm.puresystems.appsys.1700.doc%2Fiwd%2Fpct_add_ef.html

Virtual system patterns

There are several updates that you can apply to a virtual system pattern:

- Updates to script packages.
- Emergency fixes for deployed instances.
- Updates to a new service level of the image from the provider (for example, upgrading from WebSphere Application Server V8.5.0.1 to V8.5.0.3). This is usually on a fixed update schedule, typically once each quarter.
- Newer releases of the image part (for example, moving from WebSphere Application Server V8.x to V8.5.x). You can quickly test the pattern to verify that the scripts and applications work on the new release.

As a prerequisite, the pattern needs to be in PureApplication System, and you need access to the pattern. A user needs to have Create and Edit catalog permissions to edit the pattern.

Virtual application pattern

The virtual application pattern has two types of updates to the pattern:

- Emergency fixes for the middleware that is used in the pattern (IBM provides those fixes for WebSphere Application Server).
- Updates that are available to the pattern types or one or more plug-ins of the virtual application pattern. This is usually on a fixed update schedule, and typically once each quarter.

As a prerequisite, the pattern needs to be in PureApplication System, and you need access to the pattern. A user needs to have Create and Edit catalog permissions to edit the pattern.

Important: Virtual *application* pattern types always reference the latest plug-ins unless they are locked before a new plug-in is imported. Virtual *system* patterns behave differently. Virtual system patterns maintain a reference to the specific part and script packages so that new versions are not automatically applied.

After completing this task, future deployments of the pattern pick up the changes. Existing instances must be either updated or redeployed.

Virtual system and virtual application instances

You can use PureApplication System console to update the deployed instance with an emergency fix pack or move to the newer service level of the image. It takes a snapshot of the virtual machines that comprise the deployed virtual system pattern before applying the fix. You can use that information to restore the system to the original state if there are problems with applying the fix. As a prerequisite, the emergency fix for the newer service level must be in the catalog, and you must have access rights to the deployed instance.

After completing this task, the deployed instances of a pattern have updated script packages and fix packs that are ready to test.

Emergency fixes: Emergency fixes are uploaded to the catalog and can be applied to virtual systems and virtual application instances. You can also create or upload your own emergency fixes.

For a virtual system instance, newer middleware images can also be uploaded to the catalog. This requires setting up a service repository for WebSphere Application Server V8 or later.

When you have that repository, you can update previously deployed virtual system instances from V7.0.0.23 to V7.0.0.25.

For service level updates, you do not need to upload an emergency fix.

6.3.2 Updating a virtual system workload

After the pattern has been deployed as a virtualized environment, it behaves like a traditionally installed, non-virtualized environment. You can continue with the update and upgrade processes, tools, and techniques that you are accustomed to using, or you can change to the scrap-and-replace approach to get all the advantages of the IBM PureApplication System.

Ask yourself this: How much time and effort does it cost your system administrator to scrap the old environment and redeploy the new, updated, or upgraded pattern? Of course, this applies primarily to when your application can handle an outage. Quite often, environments cannot handle an outage, even for nonproduction deployments.

Think about the following scenario: You want to benefit from a fix in the newest release of WebSphere Application Server. Your application runs on top of a clustered environment. The traditional approach is to apply that fix instance by instance, including the Deployment Manager. The load on the application and the time that it takes to copy files over, make changes to your configuration files, and stop and restart the various instances depends on the size of your cluster. This is a procedure that is tedious and must meet the following conditions:

- Meticulously tested and tried
- Properly and thoroughly documented (including rollback scenarios)
- Tested and tried elsewhere

Finally, this procedure can be prone to human error, which can lead to investing even more effort and time. With PureApplication System, your system administrator has the option to completely wipe the existing environment and replace it with a fresh one, without risk of making a typo, running the wrong configuration script, or applying a step to the wrong component. Of course, if your environment relies on a database or a file system, you need to include these in your redeployment steps.

Even if your application cannot handle a scrap-and-replace outage, you can still consider the scratch-and-redeploy option. However, investigate whether you need to include these steps:

- Restoration of a database from a backup
- Reconfiguration to change the existing data for use with the newer version when the database structure and content are specific to the product version

Consider the case of an enterprise product that runs on top of WebSphere Application Server. Moving this infrastructure from V8.x to V8.x+1 involves the following steps, at a minimum:

1. Copy the binary files to all instances that need to be updated.
2. Remove a member from the cluster by removing the server so that you can update this instance.
3. Apply the update to the first instance, and run scripts on the application instances and the database instance.
4. Prepare for a rollback if there is an error.

After all of the various parts of the architecture have been updated, re-enable the cluster synchronization. Stopping and restarting services, manually changing the configuration files, and running configuration scripts leaves much room for human error and unforeseen accidents. That slows the upgrade process.

Here is the same scenario on IBM PureApplication System:

1. From the IBM website, upload the new version of the WebSphere pattern into your local catalog.
2. If this is a new pattern, define the architecture that you want, the number of instances, the database configuration, and so on in the PureApplication System test environment. If this is not a new pattern, flip the version switch in the existing pattern, and you are finished.
3. Deploy the pattern to check where your custom pattern requires particular tweaks and company-specific changes. If you need to reuse particular artifacts, such as pages and data, write scripts for those so that they are injected in the database during or after pattern deployment. You can restore the contents of the database from an earlier backup.
4. **Important step:** Define a cutoff moment where no new updates to the old environment are allowed.

Monitoring: PureApplication System gives you several options to monitor what is going on at deployment time. There are various system logs and, for each script, if you configure a separate `systemout` log file, a `systemerr` log is generated.

5. If this pattern fits your needs, make it available to the system deployer so that person can deploy it in the production cloud group.
6. Schedule production deployment automatically or manually. Make sure that you use the database export from shortly after the cutoff moment and include both provided and custom scripts that work on the database. The data model might need to be migrated or converted from the older version to the newer version. Consider whether this can be automated and can be included in the pattern deployment as a separate step.
7. Balance the load from the previous environment, and stop the old environment. Consider exporting it and backing it up.
8. Delete that environment from your system to freeing resources for other uses.

Compare both approaches, and assess the complexity of both operations. They result in the same outcome. Then, consider the time, complexity, and the elements where human errors can occur during each of those approaches.

Related publications

The publications that are listed in this section are particularly suitable for more detailed information about the topics covered in this book.

IBM Redbooks

The following IBM Redbooks publications provide additional information about the topics in this book. Some publications that are referenced in this list might be available in softcopy only.

- *IBM PureFlex System and IBM Flex System Products and Technology*, SG24-7984
- *Adopting IBM PureApplication System V1.0*, SG24-8113
- *Getting Started with IBM Tivoli Monitoring 6.1 on Distributed Environments,* SG24-7143
- *Creating Composite Application Pattern Models for IBM PureApplication System*, SG24-8146
- *IT Asset Management Processes using Tivoli Asset Manager for IT,* SG24-7601-00
- *Deployment Guide Series: Tivoli IT Asset Management Portfolio*, SG24-7602-00
- *Implementing the IBM Storwize V7000 V6.3*, SG24-7938
- *Deploying IBM Flex System into a Cisco Network*, REDP-4901

You can search for, view, download, or order these and other Redbooks and Redpapers publications, Web Docs, drafts, and additional materials, on the Redbooks website:

ibm.com/redbooks

Online resources

These websites are also relevant as further information sources:

- IBM PureApplication System W1500 Information Center: Planning information

  http://pic.dhe.ibm.com/infocenter/psappsys/v1r0m0/topic/com.ibm.ipas.doc/iwd/gst_planning.html

- IBM PureApplication System W1700 Information Center

  http://pic.dhe.ibm.com/infocenter/psappsys/v1r1m0/index.jsp?topic=%2Fcom.ibm.puresystems.appsys.1700.doc%2Fwelcome%2Fwelcome.html

- IBM PureApplication System Information Center: Security overview

  http://pic.dhe.ibm.com/infocenter/psappsys/v1r0m0/topic/com.ibm.ipas.doc/iwd/aac_overview_sec.html

- Application Migration Tool for WebSphere Version Migration

  http://public.dhe.ibm.com/software/dw/wes/migrationtoolkit/ApplicationMigrationTool_en_US.3.5.1_WebSphere_v2v.pdf

- Introduction to Advanced Middleware Configuration

  http://pic.dhe.ibm.com/infocenter/psappsys/v1r0m0/topic/com.ibm.ipas.doc/RAF/com.ibm.raf.install.doc/topics/c_amc_intro.html

- Extending and capturing virtual images (extend an IBM Hypervisor Edition image)

  http://pic.dhe.ibm.com/infocenter/psappsys/v1r0m0/topic/com.ibm.ipas.doc/iwd/pct_extend_vi.html

- Working with the IBM Image Construction and Composition Tool

  http://pic.dhe.ibm.com/infocenter/psappsys/v1r0m0/topic/com.ibm.ipas.doc/ICON/topics/iwd_cicn_overview.html

- Policies

  http://pic.dhe.ibm.com/infocenter/psappsys/v1r0m0/topic/com.ibm.ipas.doc/iwd/apt_policyov.html

- Applying maintenance upgrades to firmware, virtual systems, and shared services

  http://pic.dhe.ibm.com/infocenter/psappsys/v1r0m0/index.jsp?topic=/com.ibm.ipas.doc/iwd/aat_updating_firmware.html

- IBM PureSystems Centre

  http://www.ibm.com/PureSystems/Centre

- A tour of the hardware in IBM PureApplication System (differences and similarities between Intel and Power versions of IBM PureApplication System):

  http://www.ibm.com/developerworks/cloud/library/cl-ps-aim1302-hardwarepureapp/

- PureApplication System W1500 specifications

  http://pic.dhe.ibm.com/infocenter/psappsys/v1r0m0/topic/com.ibm.ipas.doc/installguide/planning/ps_sysreq.html

- Site-readiness planning

  http://pic.dhe.ibm.com/infocenter/psappsys/v1r0m0/topic/com.ibm.ipas.doc/installguide/planning/purescaleplankickoff.html

- Managing administrative access in IBM PureApplication System

  http://www.ibm.com/developerworks/websphere/library/techarticles/1211_woolf/1211_woolf.html

- Configuring LDAP security settings for system console users and groups

  http://pic.dhe.ibm.com/infocenter/psappsys/v1r0m0/topic/com.ibm.ipas.doc/iwd/aat_secmaint.html

- High availability topologies for IBM PureApplication System

  http://www.ibm.com/developerworks/websphere/library/techarticles/1206_brown/1206_brown.html

- IBM developerWorks articles about PureApplication System

  http://www.ibm.com/developerworks/puresystems/index.html

- Offering announcement for Version 1.1, which lists a simplified disaster recovery solution

  http://www.ibm.com/common/ssi/cgi-bin/ssialias?infotype=AN&subtype=CA&htmlfid=897/ENUS213-157&appname=USN

- System maintenance and pre-entitled component updates in PureApplication System Version 1.1.0.1

  http://www.ibm.com/support/docview.wss?uid=swg27039773

- IBM PureApplication System Monitoring Demonstration (video)

  http://www.youtube.com/watch?v=WECV9cdzSfE&list=PLA5O8B797DAECDBDF&index=11

Help from IBM

IBM Support and downloads

ibm.com/support

IBM Global Services

ibm.com/services

(0.2”spine)
0.17”<->0.473”
90<->249 pages

IBM PureApplication System Best Practices

Discover best practices for optimal deployments

Learn how others use PureApplication System

Get standard implementation examples to follow

This IBM Redbooks publication describes IBM PureApplication System preferred practices that are based on IBM client and Business Partner experience. It explains how PureApplication System enables industries to consolidate workloads, increase efficiency, automate routine processes, reduce costs, and become more agile to respond to continually changing business needs.

This book is particularly useful to solution specialists, system or software architects, and the IT teams who implement PureApplication System cloud services.

INTERNATIONAL TECHNICAL SUPPORT ORGANIZATION

BUILDING TECHNICAL INFORMATION BASED ON PRACTICAL EXPERIENCE

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

For more information:
ibm.com/redbooks

SG24-8145-00 ISBN 0738439061