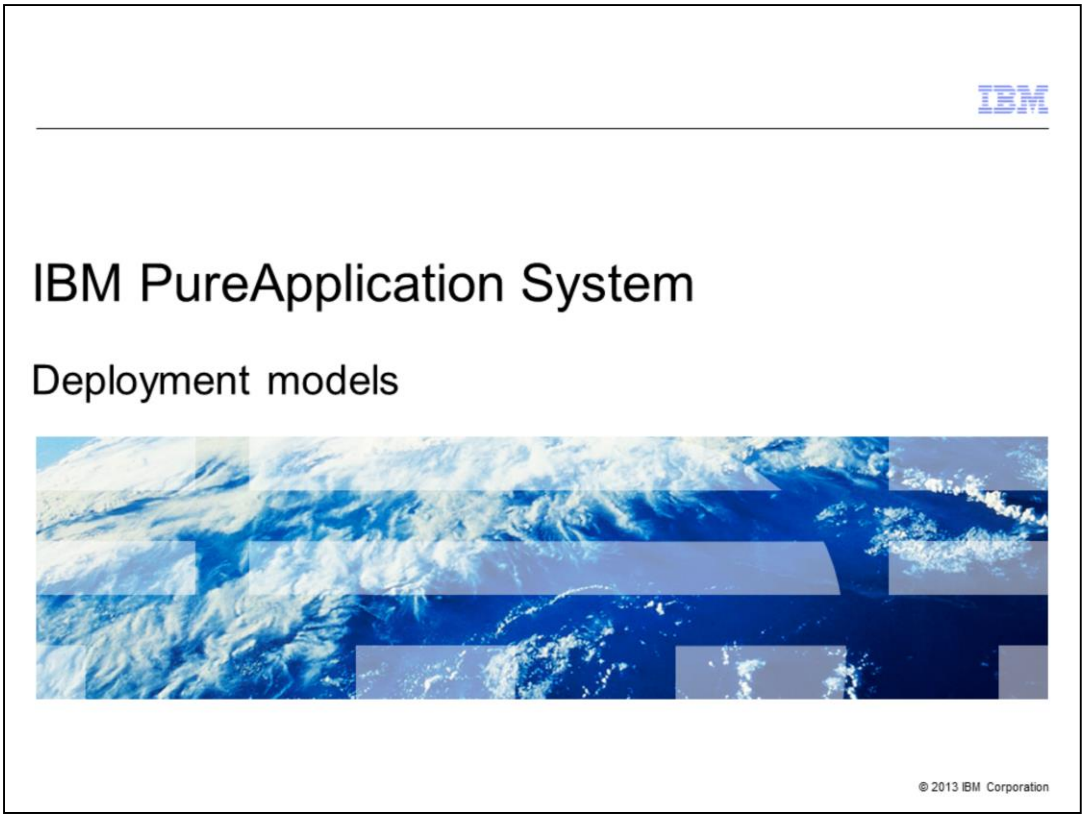
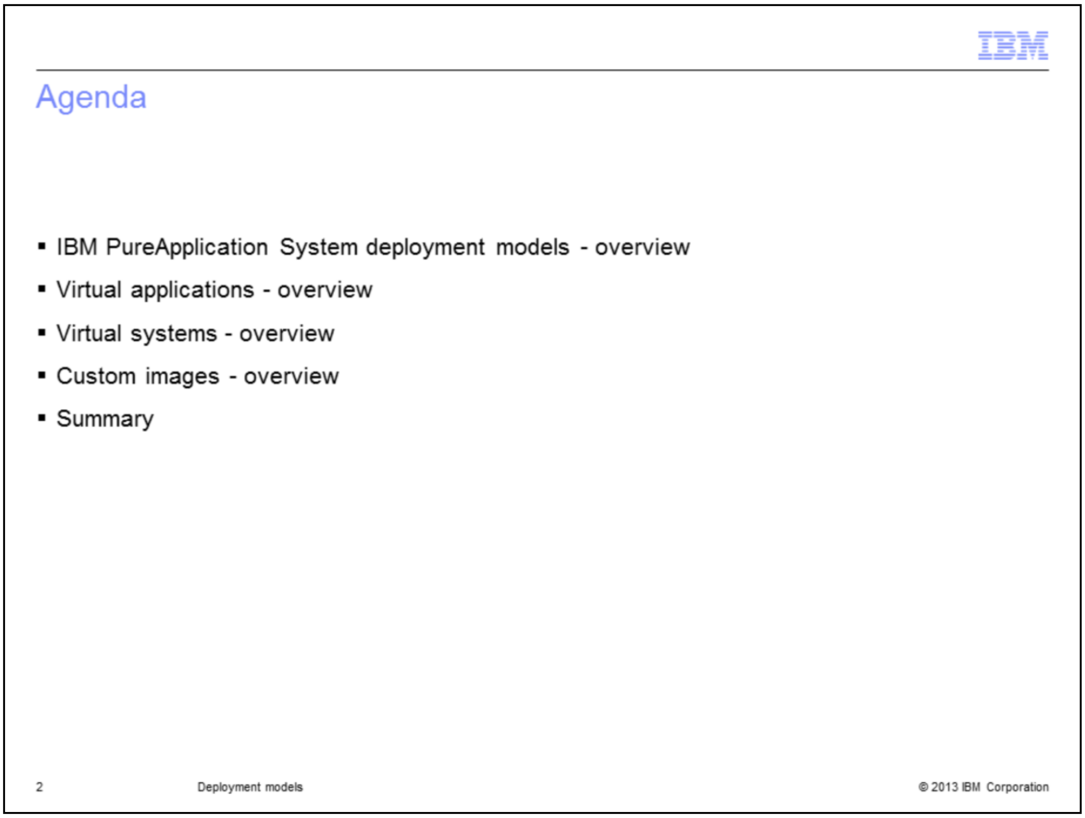


IBM PureApplication System

Deployment models

This presentation will cover the various deployment models available in IBM PureApplication™ System.

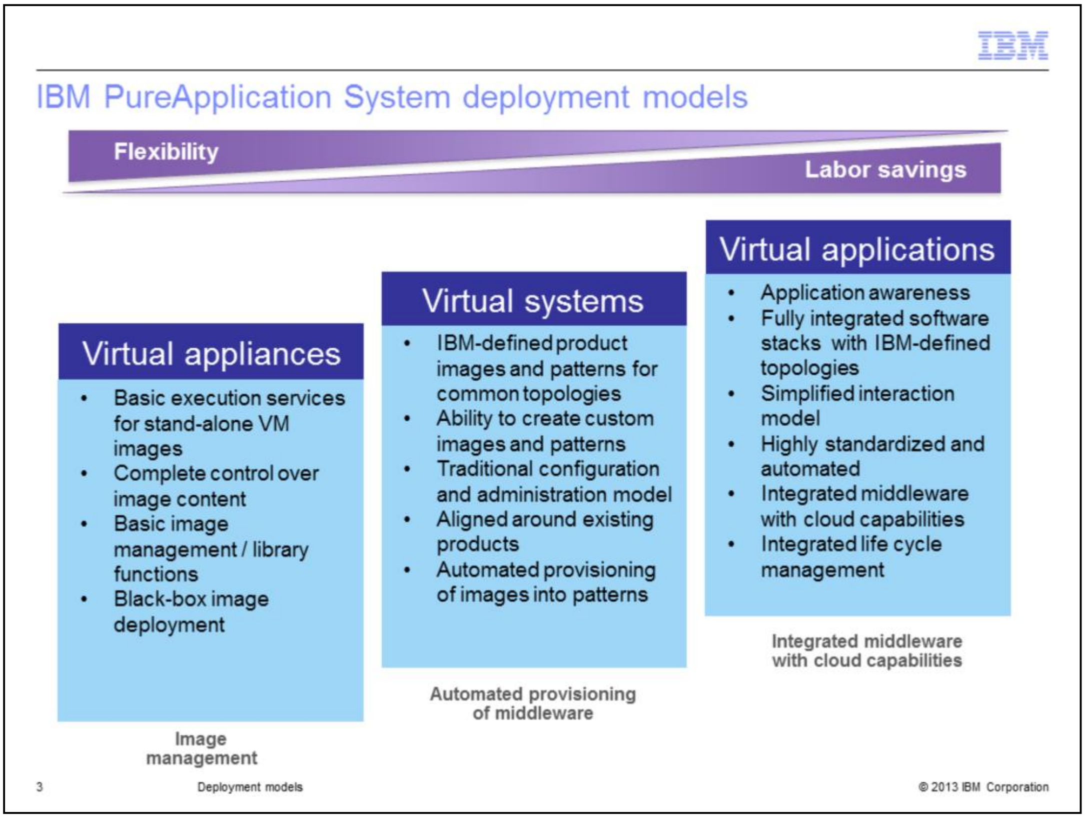


Agenda

- IBM PureApplication System deployment models - overview
- Virtual applications - overview
- Virtual systems - overview
- Custom images - overview
- Summary

2 Deployment models

After a brief overview of the various deployment models, you will see an overview of the virtual application and virtual system deployment models. You will then see how you are able to create custom images if you want to deploy images not currently available from IBM.



IBM PureApplication System deployment models

Flexibility

Labor savings

Virtual appliances

- Basic execution services for stand-alone VM images
- Complete control over image content
- Basic image management / library functions
- Black-box image deployment

Image management

Virtual systems

- IBM-defined product images and patterns for common topologies
- Ability to create custom images and patterns
- Traditional configuration and administration model
- Aligned around existing products
- Automated provisioning of images into patterns

Automated provisioning of middleware

Virtual applications

- Application awareness
- Fully integrated software stacks with IBM-defined topologies
- Simplified interaction model
- Highly standardized and automated
- Integrated middleware with cloud capabilities
- Integrated life cycle management

Integrated middleware with cloud capabilities

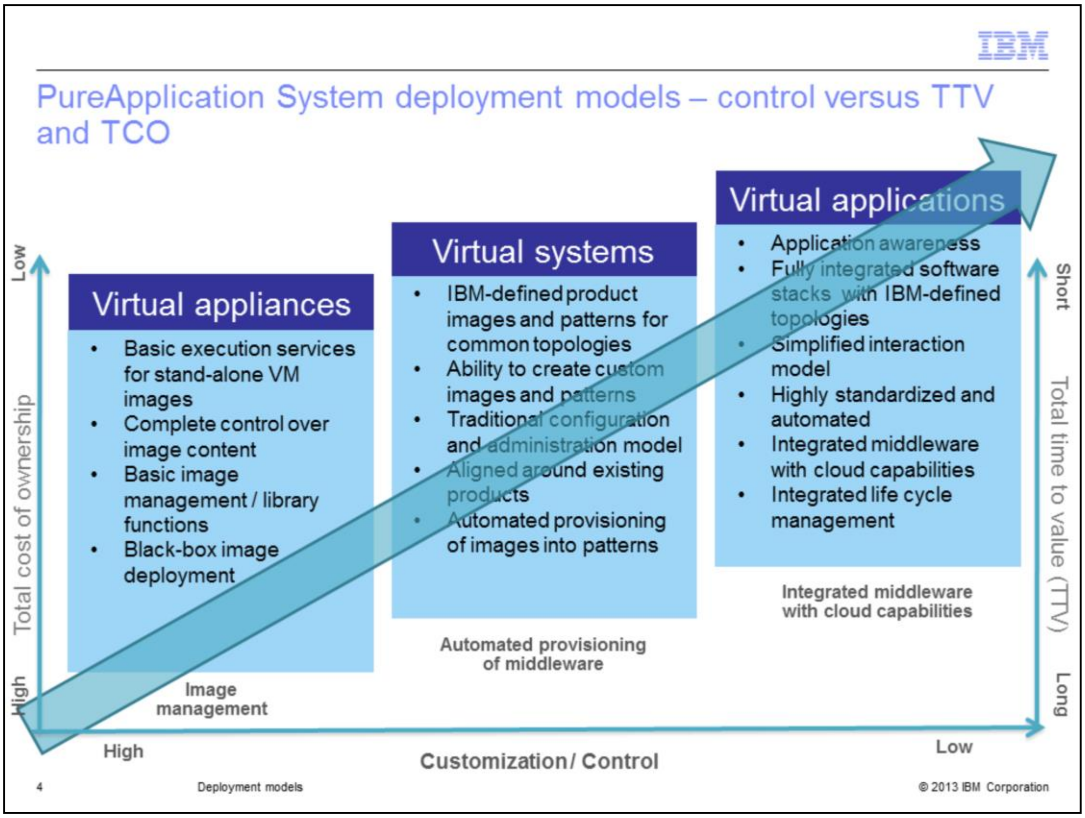
3 Deployment models © 2013 IBM Corporation

PureApplication System provides different deployment models, allowing you to choose which model is best suited for your needs. Starting over on the left with virtual appliances, this is the most generic type of deployment in which you can deploy any generic OVF file. While you have complete control over the image content, PureApplication System knows nothing about it. To PureApplication System, it is a black box so PureApplication System can only provide basic execution services such as stopping and starting the virtual machines where your image is deployed. PureApplication System also provides basic image management functions from the image catalog for virtual appliances.

The more interesting and optimized styles of deployment are virtual systems and virtual applications.

Moving to the right, the virtual systems deployment model uses IBM’s hypervisor edition images with pre-installed middleware. This allows you to define your topology as a pattern, customize it with script packages that represent your application, and finally deploy that pattern into the cloud. This model provides rich customization capabilities and allows fast, automated provisioning of IBM middleware, based on the specific topology that you have defined. For example, you can define a virtual system pattern as a multi-node WebSphere® Application Server topology that contains a deployment manager, one or more custom nodes and an IBM HTTP Server. Once you have that topology defined as a pattern, PureApplication System provisions that for you with minimal effort on your part. That pattern can then be deployed multiple times with the same results each time as the deployment is fully automated. IBM also provides a set of patterns that follow best practices based on years of experience. You can use those patterns as-is or use them as a starting point for your own patterns, customizing them as needed. You can also create your own custom images to be deployed as virtual systems using the IBM Image Construction and Composition Tool. This is useful for the cases where you need an image containing middleware where IBM does not currently provide a hypervisor edition image and for vendor software you want to deploy.

Whereas the virtual systems deployment model is middleware centric, the deployment model on the far right, virtual applications, is application centric. With the virtual application deployment model, the focus turns from the middleware and moves to the application. PureApplication System takes care of what middleware is needed to run those applications and determines the topology based on your artifacts and any policies you specify. PureApplication System then manages the life cycle of the application including growing and shrinking the servers needed to satisfy specified service levels. This deployment model is fully automated like the virtual systems deployment model but adds integrated life cycle management.



PureApplication System deployment models – control versus TTV and TCO

Virtual appliances

- Basic execution services for stand-alone VM images
- Complete control over image content
- Basic image management / library functions
- Black-box image deployment

Image management

Virtual systems

- IBM-defined product images and patterns for common topologies
- Ability to create custom images and patterns
- Traditional configuration and administration model
- Aligned around existing products
- Automated provisioning of images into patterns

Automated provisioning of middleware

Virtual applications

- Application awareness
- Fully integrated software stacks with IBM-defined topologies
- Simplified interaction model
- Highly standardized and automated
- Integrated middleware with cloud capabilities
- Integrated life cycle management

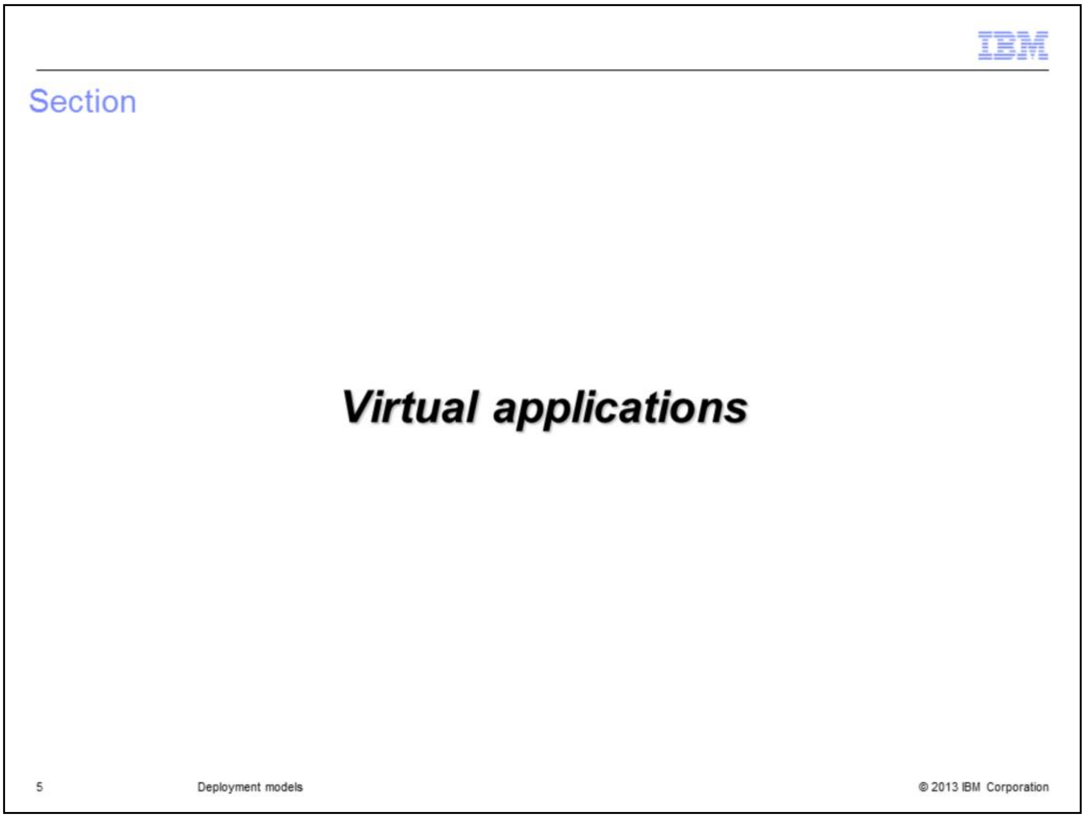
Integrated middleware with cloud capabilities

4 Deployment models © 2013 IBM Corporation

Looking at the same slide but adding time to value (TTV) and total cost of ownership (TCO) on the axis, you see that the customization and control is highest with the virtual appliance model. Along with that though, the virtual appliance model also has the highest total cost of ownership and longest time to value. With the virtual appliance deployment model, you at some point, have to do the standard software installation and configuration on the operating system to create the image to be deployed. While this model is very flexible, giving you complete control over what the image contains, building and maintaining your own images is typically very labor intensive. Since PureApplication System has no knowledge of what the image contains it just provides basic execution services and image management. You are fully responsible for its content and administration. This increases costs and time to value but, again, gives you a high degree of customization.

Moving to virtual systems, you can save time and ownership cost by relying on IBM's pre-built hypervisor edition images for many product capabilities. You still have significant ability to customize your topology deployments here. You can customize through image extension, by defining the specific topology and middleware levels for your application, and by using script packages to customize specific components. Administration is 'business as usual' with the products you deploy. For instance, for WebSphere Application Server, you can use the administration console and wsadmin commands in addition to capabilities provided to you in the PureApplication System administration console. This allows for quick deployment of highly customized middleware however, thus reducing time to value. The cost of ownership also goes down considerably when you rely on IBM to maintain the product images.

Looking at virtual applications, you drastically reduce the total cost of ownership and time to value, because the solutions are pre-built and integrated for a specific use case. Rather than defining topologies, you provide your application artifacts, and PureApplication System will determine the appropriate underlying topology based on the service-level agreement that you provide. While using the virtual application deployment model is the most cost-effective, its high level of standardization and cross-product integration results in fewer product configuration options being exposed for customization. For example, with WebSphere Application Server deployments, you do not have access to the WebSphere administration console. Instead, you have a limited set of customizations available to you through the PureApplication System administration console.

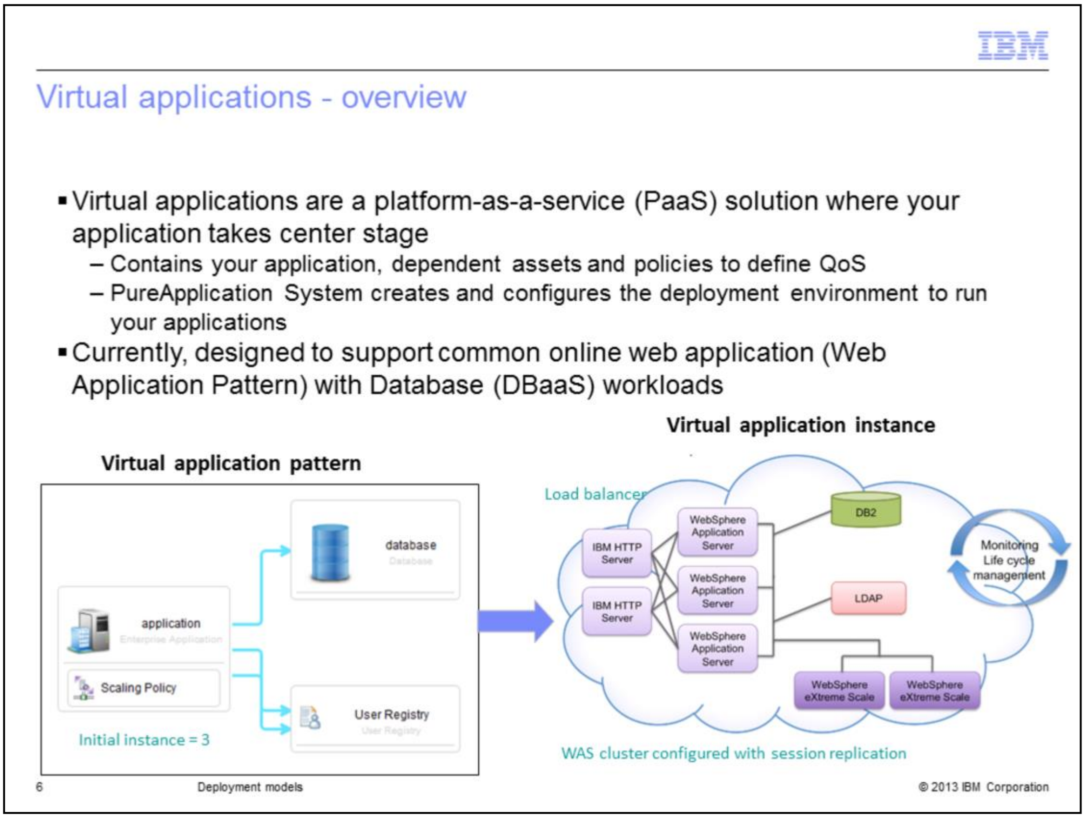


Section

Virtual applications

5 Deployment models © 2013 IBM Corporation

In this next section you will look at an overview of the virtual applications deployment model.



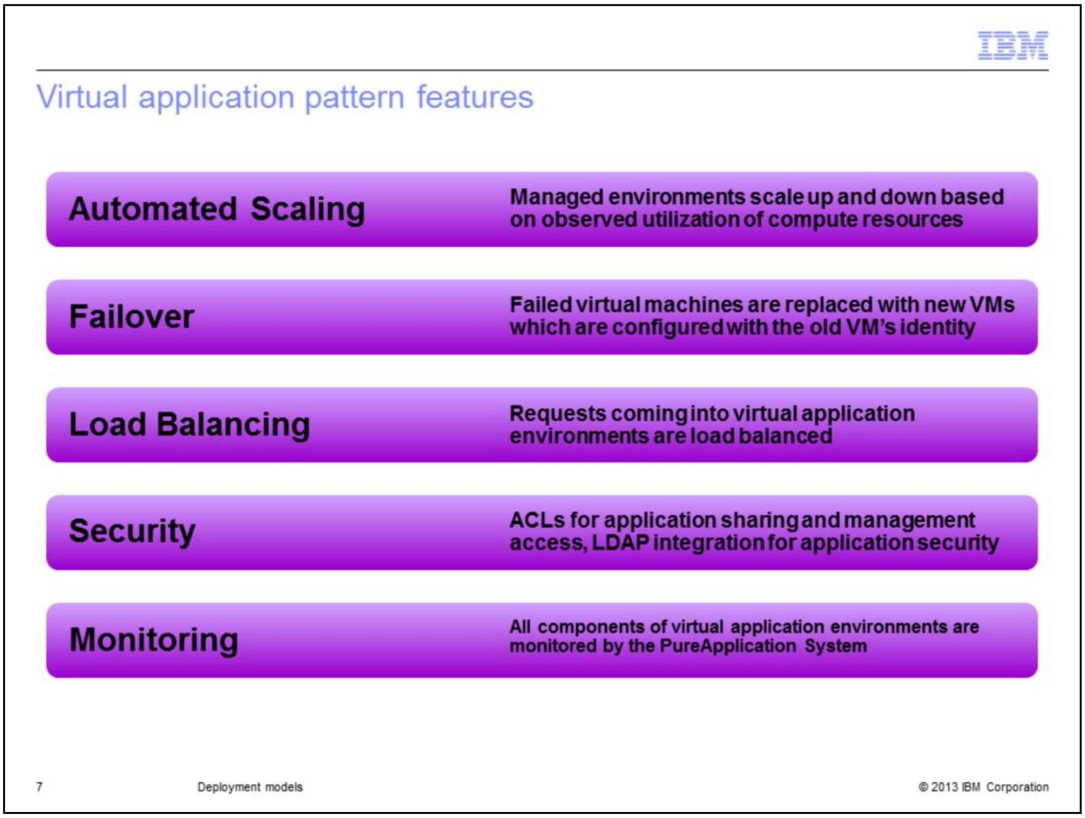
Virtual applications - overview

- Virtual applications are a platform-as-a-service (PaaS) solution where your application takes center stage
  - Contains your application, dependent assets and policies to define QoS
  - PureApplication System creates and configures the deployment environment to run your applications
- Currently, designed to support common online web application (Web Application Pattern) with Database (DBaaS) workloads

6 Deployment models © 2013 IBM Corporation

The virtual application deployment model is a platform-as-a-service (PaaS) model where your application is the focal point. To create a virtual application pattern, you start with your application and define its specific requirements, such as what services it requires and the quality of service (QoS) that is needed. Based on the assets in your virtual application pattern, PureApplication System will, in the background, deploy and configure the appropriate middleware components to run your application. This simplifies the end-to-end process of creating, deploying and configuring the middleware components for your applications. PureApplication System handles it all. Once deployed, PureApplication System also takes care of monitoring the application for you, adding resources as needed, for example, to meet the QoS requirements.

PureApplication System supports most common online web applications as virtual applications with the IBM Web Application Pattern and IBM Database Patterns. The IBM Database Patterns allow databases to be deployed in a Database-as-a-Service (DBaaS) cloud environment on their own or as part of a virtual application instance.



Virtual application pattern features

Automated Scaling — Managed environments scale up and down based on observed utilization of compute resources

Failover — Failed virtual machines are replaced with new VMs which are configured with the old VM's identity

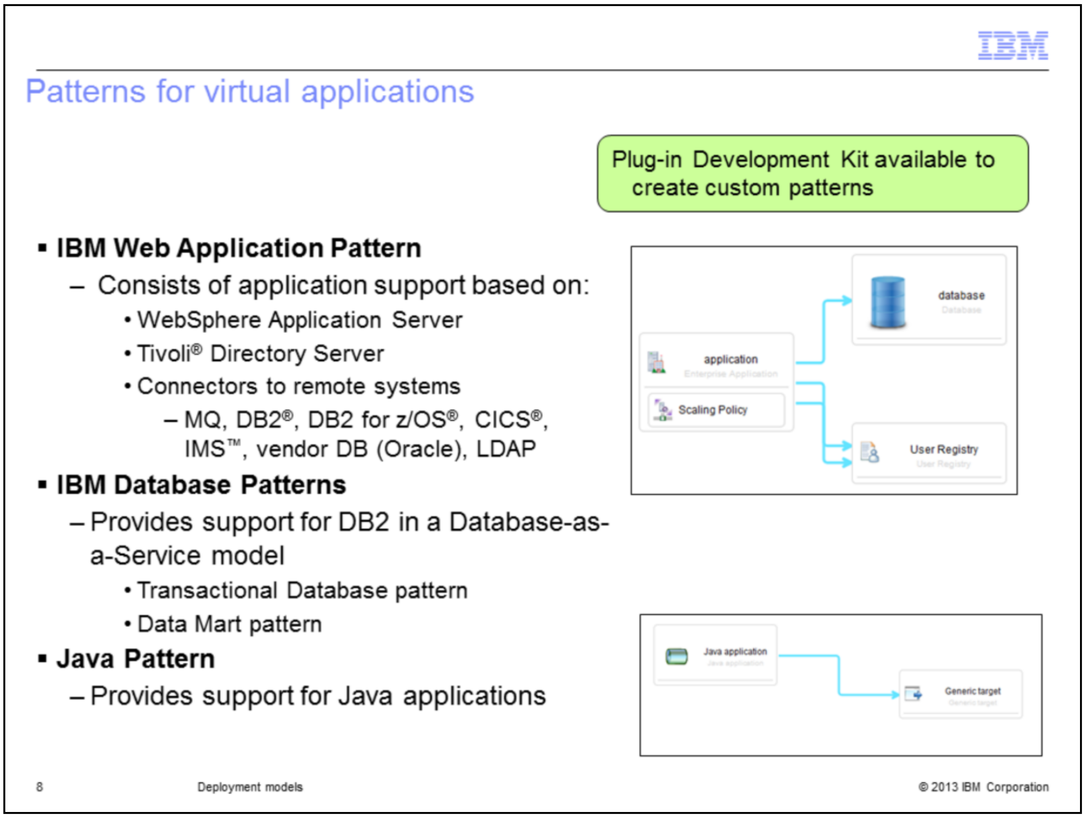
Load Balancing — Requests coming into virtual application environments are load balanced

Security — ACLs for application sharing and management access, LDAP integration for application security

Monitoring — All components of virtual application environments are monitored by the PureApplication System

7 Deployment models © 2013 IBM Corporation

With the virtual application deployment model, you have features to help manage the deployment and life cycle of your application, easing the administrative burden placed on you. You can include a policy in your virtual application pattern that will provide automated scaling, managed by PureApplication System. As PureApplication System monitors the resources on the system, it will scale up and down based on the application load. In addition, PureApplication System will automatically restart your application if a virtual machine has a problem at any time, thus providing application failover automatically. Load balancing is also done for you by Pure Application System when using the proxy shared service in your cloud such that requests being serviced by your virtual application will automatically be distributed across available instances. Security is an important feature in virtual applications, allowing highly secure environments that can easily be integrated with LDAP for application security. Finally, all the various components of the virtual application environments are monitored for you by PureApplication System. This allows you to gain quick access to the status, performance, and resource usage of all areas of your virtual application.



Patterns for virtual applications

Plug-in Development Kit available to create custom patterns

- **IBM Web Application Pattern**
  - Consists of application support based on:
    - WebSphere Application Server
    - Tivoli® Directory Server
    - Connectors to remote systems
      - MQ, DB2®, DB2 for z/OS®, CICS®, IMS™, vendor DB (Oracle), LDAP
- **IBM Database Patterns**
  - Provides support for DB2 in a Database-as-a-Service model
    - Transactional Database pattern
    - Data Mart pattern
- **Java Pattern**
  - Provides support for Java applications

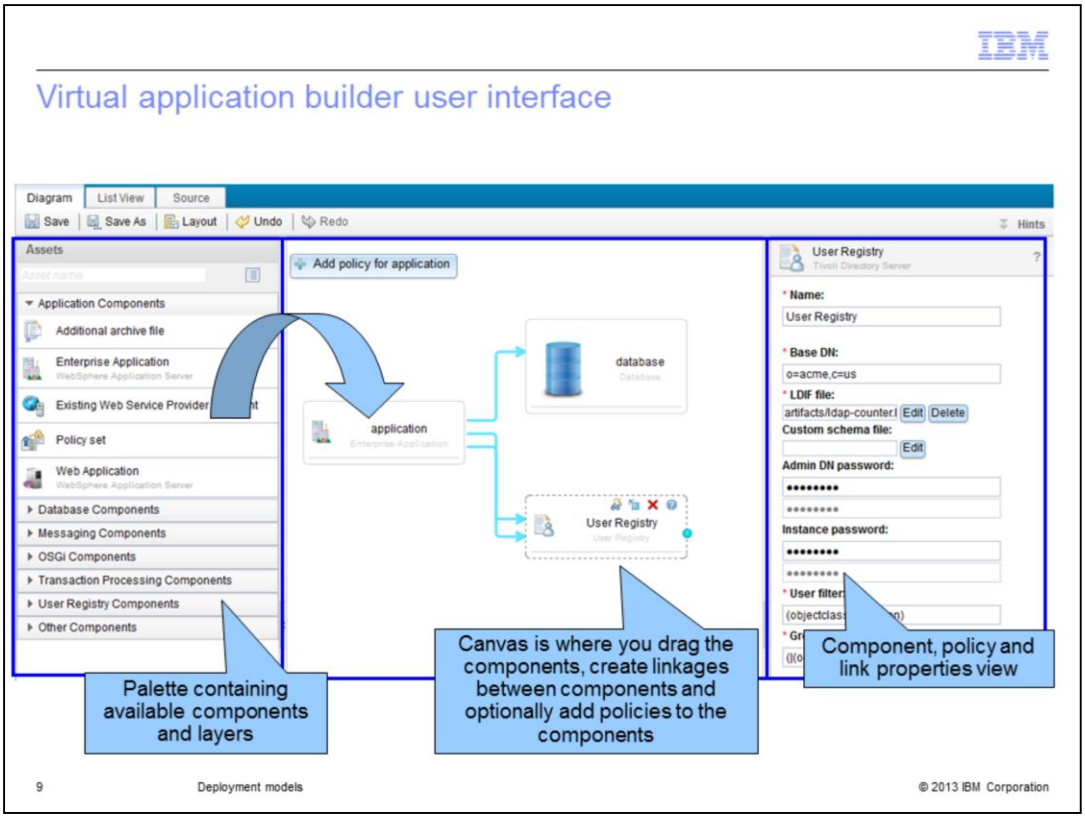
8 Deployment models © 2013 IBM Corporation

There are three types of virtual application patterns included with PureApplication System. They include the IBM Web Application Pattern, the IBM Database Patterns and the Java Pattern. The IBM Web Application Pattern provides a set of components that are typically needed for online web applications. This includes Java Platform Enterprise Edition (Java EE) applications, databases, Lightweight Directory Access Protocol (LDAP) servers, and messaging. These components are based on the products shown on the slide such as WebSphere Application Server, and Tivoli Directory Server. The Web Application Pattern also allows you to incorporate connectors to remote systems, such as MQ, CICS and IMS, into your virtual application pattern.

You can use IBM Database Patterns separately or you can incorporate them into a virtual application pattern based on the Web Application Pattern. The database pattern incorporated into a virtual application pattern is shown on the slide. The IBM Database Patterns provide support for DB2 in a Database-as-a-Service (DBaaS) model, allowing you to simplify and standardize the creation of needed databases. There are actually two different database patterns available, based on your needs. The Transactional Database pattern is primarily used for online transaction processing and is optimized for transactional applications. The Data Mart pattern is primarily used for data warehousing and is optimized for reporting applications.

The Java Pattern provides support for building Java applications. This pattern type provides an easy and fast mechanism for provisioning Java applications. It also has components that allow you to connect to network resources, such as databases and web services.

While these are the patterns included with PureApplication System, there are other patterns that you can import into the PureApplication System. You also have the ability to create your own patterns using the Plug-in Development Kit which is available for download.



Virtual application builder user interface

* Name: User Registry

* Base DN: o=acme,c=us

* LDIF file: artifacts/ldap-counter.l

Custom schema file:

Admin DN password: •••••••• ••••••••

Instance password: •••••••• ••••••••

* User filter: (objectclas... n)

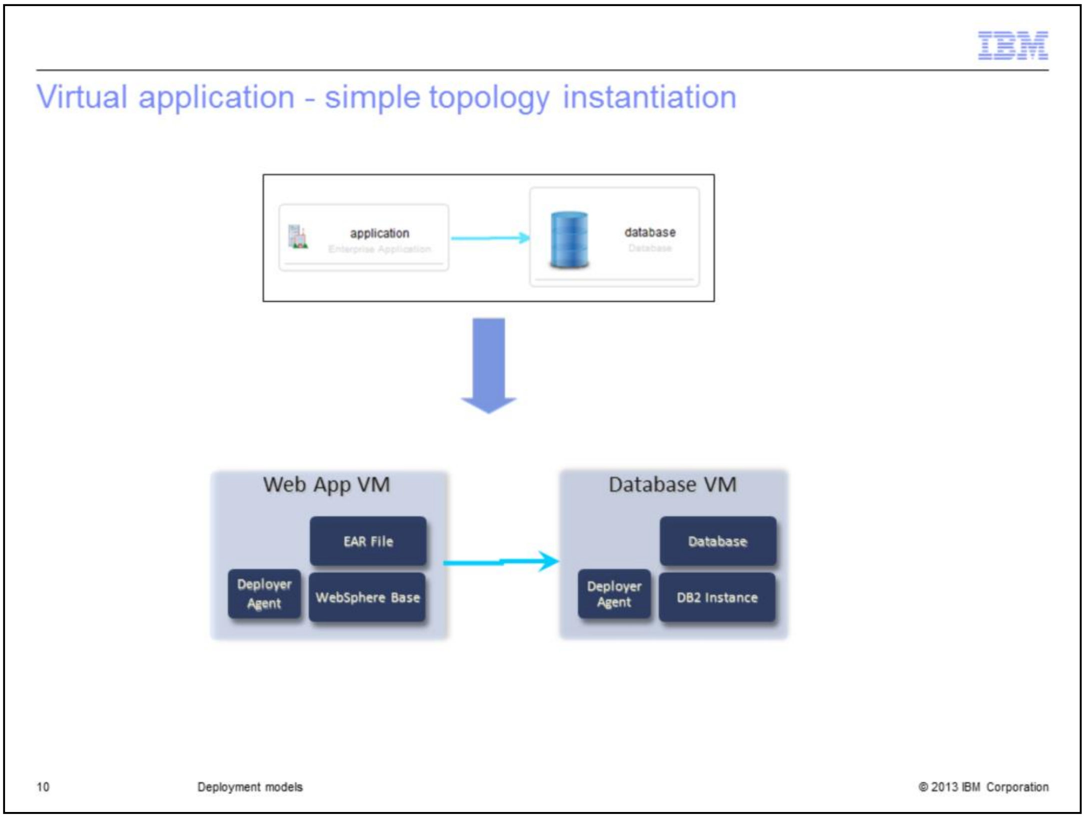
* Gr... (|(o...

Palette containing available components and layers

Canvas is where you drag the components, create linkages between components and optionally add policies to the components

Component, policy and link properties view

This slide is showing the virtual application builder user interface. This is what you use to define your virtual application pattern. Over on the left, you see the components available for your virtual application pattern. This is showing the Web Application Pattern so you see application components like “Enterprise application” and “Web application.” You also see the pattern contains Database Components, Messaging Components and User Registry Components. To build your pattern you drag your application’s needed components on the canvas in the middle of the interface. The canvas here is showing a pattern that includes an enterprise application that connects to a User Registry component and a Database component. Over on the right is where you define the components, telling PureApplication System what enterprise application to deploy, for instance. This slide is showing the properties to use when deploying the user registry that is needed. This includes an LDIF file that will define the LDAP database to be used.



Virtual application - simple topology instantiation

application
Enterprise Application

database
Database

Web App VM

EAR File

Deployer Agent

WebSphere Base

Database VM

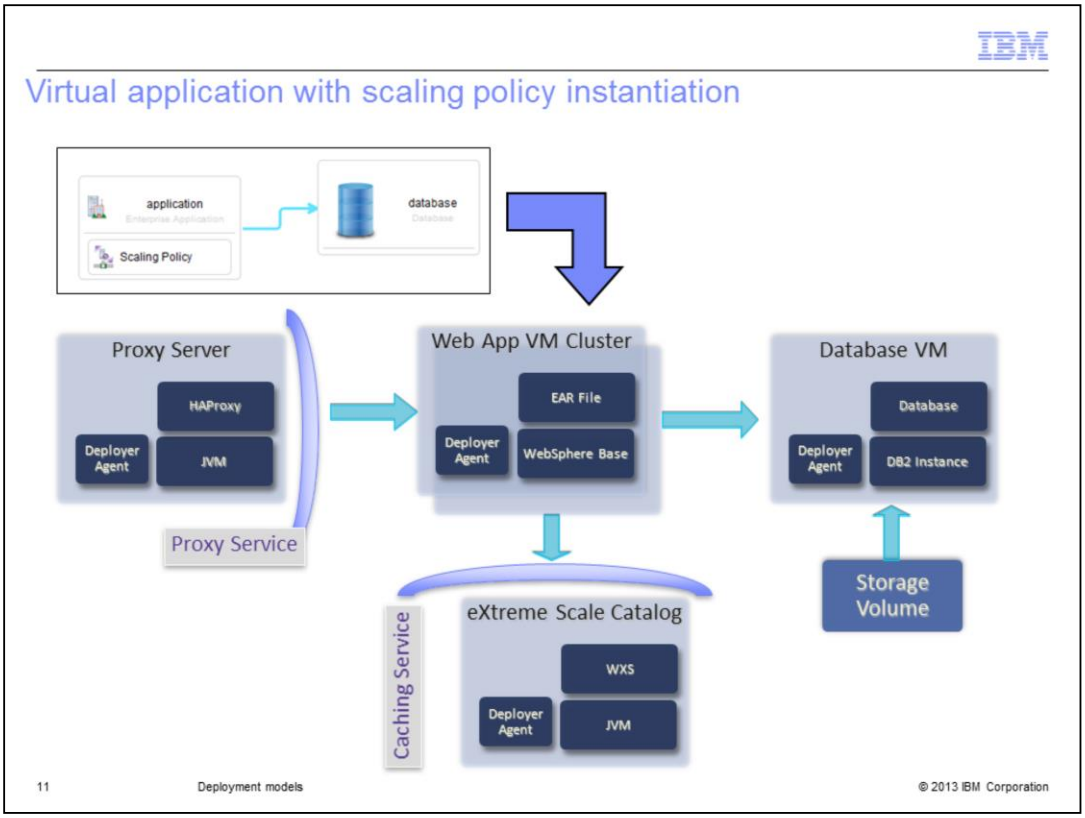
Database

Deployer Agent

DB2 Instance

10 Deployment models © 2013 IBM Corporation

This slide shows how the simple virtual application topology gets instantiated by PureApplication System. The virtual application pattern consists of an enterprise application component and a database component. Each of those components get deployed into a separate virtual machine (VM), along with the needed middleware and a deployer agent which orchestrates the deployment. The middleware that gets installed in the case of the Enterprise Application component is WebSphere Application Server whereas in the case of the database component, DB2 is installed. You see that an .ear file is also installed and a database is created in the respective VMs.

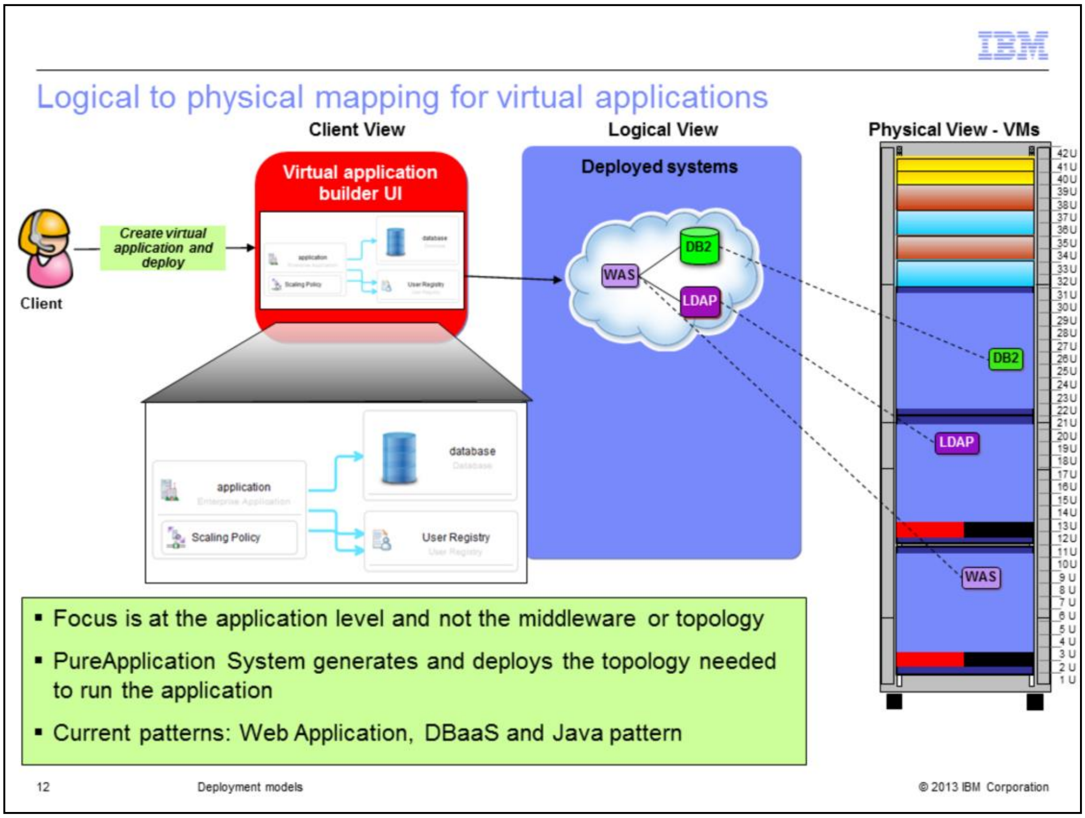


IBM

Virtual application with scaling policy instantiation

11 Deployment models © 2013 IBM Corporation

Just by adding a scaling policy to the virtual application pattern, you get a much more complicated instantiation, all determined by PureApplication System based on your simple virtual application pattern. In this case, your application relies on a couple of shared services, the proxy service and the caching service. The proxy shared service provides routing and load balancing for the multiple web applications that are deployed on your behalf. The scaling policy also requires the use of the caching shared service which allows for highly efficient caching based on WebSphere eXtreme Scale. Note that the shared services are shared among all deployments within a cloud. They are not deployed when this pattern is instantiated. The shared services will already be running and will just be used here.

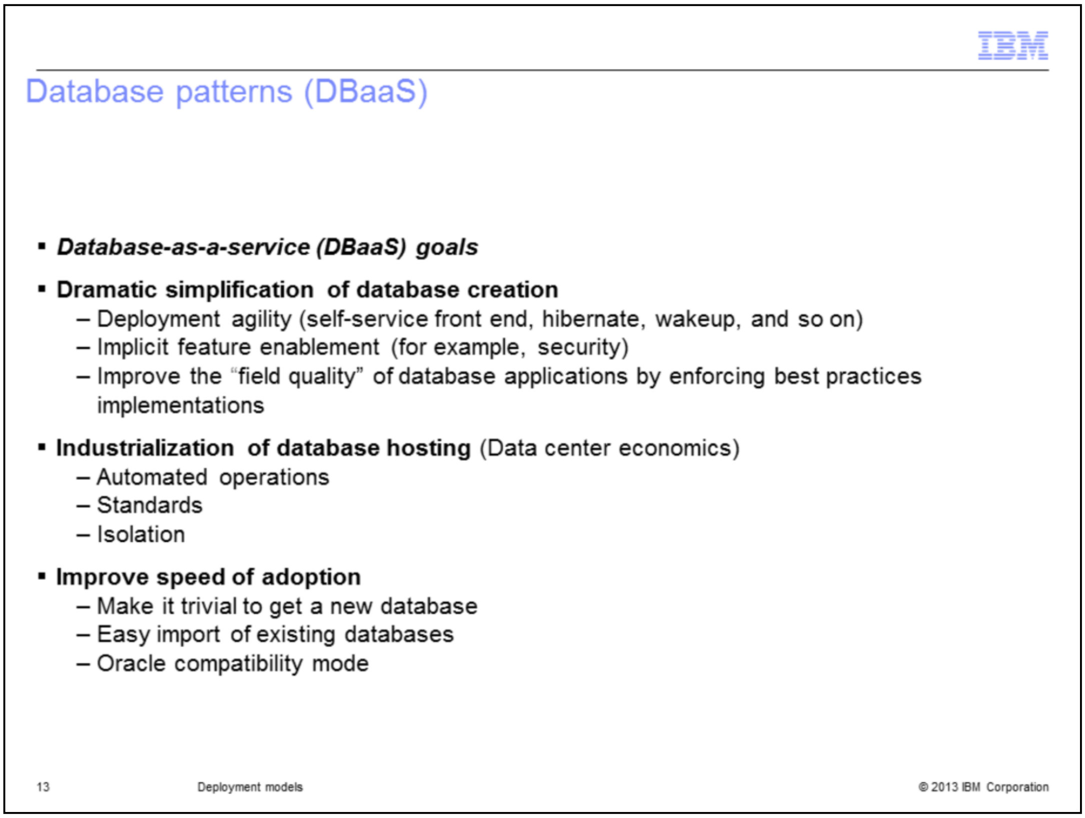


Logical to physical mapping for virtual applications

- Focus is at the application level and not the middleware or topology
- PureApplication System generates and deploys the topology needed to run the application
- Current patterns: Web Application, DBaaS and Java pattern

12 Deployment models © 2013 IBM Corporation

This slide is showing that from the client view, you just need to create a virtual application pattern using the virtual application builder user interface and deploy it. The focus for you is at the application level and you do not make any decisions about the topology or the middleware deployed to run your application. The logical view shows that a WebSphere instance, a DB2 instance and an LDAP instance are deployed into the cloud for this particular virtual application pattern. What this entails physically is shown on the right where three separate virtual machines are created in the PureApplication System rack. PureApplication System determines what is needed topology-wise and where each of the virtual machines are deployed, based on the application you want to run. What this picture does not show is the shared services that are also used by this virtual application because of the scaling policy that was added to the pattern. Those virtual machines already exist in the cloud and are used in this deployment as well.

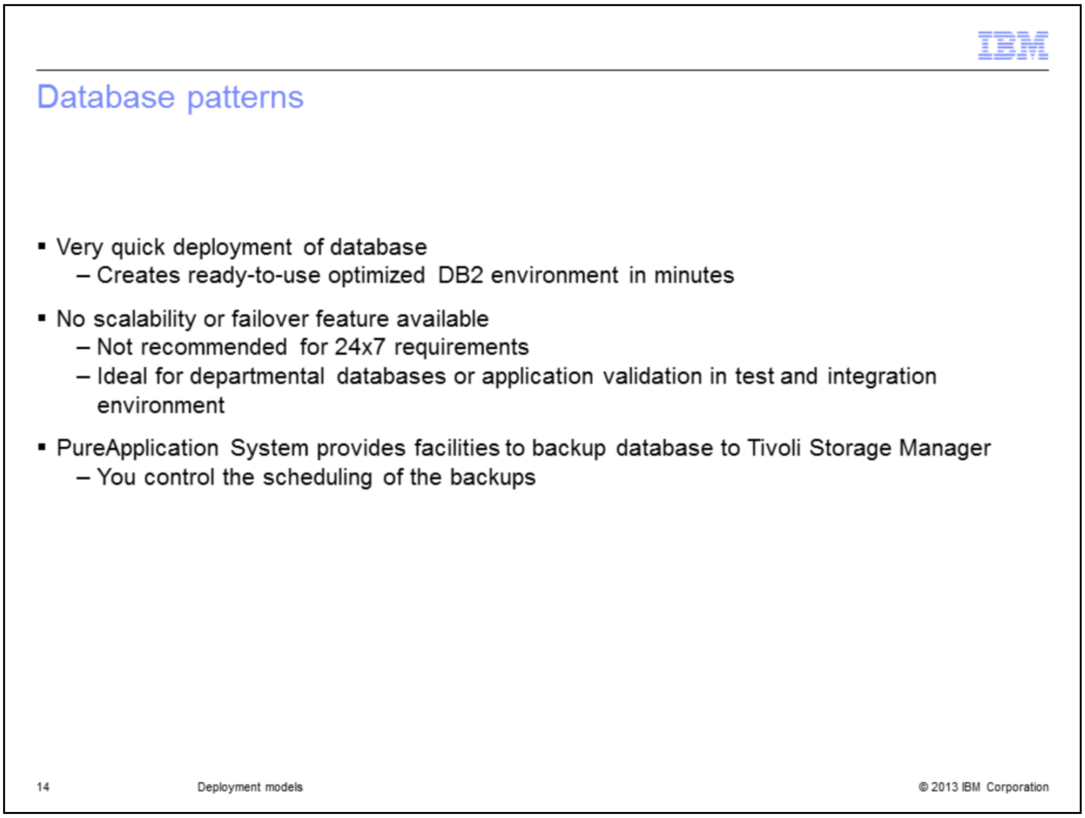


Database patterns (DBaaS)

- ***Database-as-a-service (DBaaS) goals***
- **Dramatic simplification of database creation**
  - Deployment agility (self-service front end, hibernate, wakeup, and so on)
  - Implicit feature enablement (for example, security)
  - Improve the “field quality” of database applications by enforcing best practices implementations
- **Industrialization of database hosting** (Data center economics)
  - Automated operations
  - Standards
  - Isolation
- **Improve speed of adoption**
  - Make it trivial to get a new database
  - Easy import of existing databases
  - Oracle compatibility mode

13 Deployment models © 2013 IBM Corporation

Looking at the database patterns, or database-as-a-service (DBaaS), component of virtual applications, you will find that you can dramatically reduce the time and cost necessary to deploy and use databases in your applications. DBaaS provides a dramatic simplification in database creation. Database administrators are busy and this model gives you a self-service option. By providing some DDL, you can easily and quickly have a database deployed. The database patterns provided give you “best practice” implementations and implicit feature enablement, such as security, automatically. As a result, you need not become an expert in database administration. You can let PureApplication System worry about the details. The DBaaS model also provides automated operations such as backups and restart on failure. By deploying your own database for testing, you easily get isolation from others’ testing. With DBaaS, it becomes trivial to get a new database.

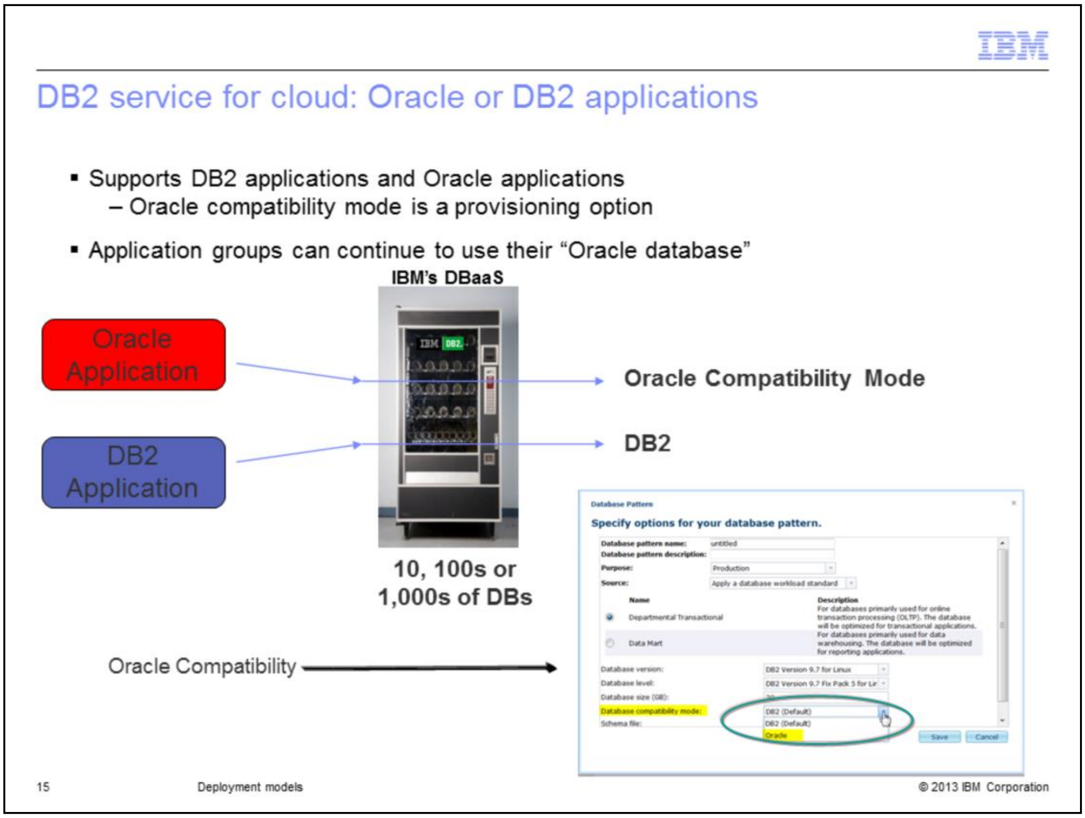


Database patterns

- Very quick deployment of database
  - Creates ready-to-use optimized DB2 environment in minutes
- No scalability or failover feature available
  - Not recommended for 24x7 requirements
  - Ideal for departmental databases or application validation in test and integration environment
- PureApplication System provides facilities to backup database to Tivoli Storage Manager
  - You control the scheduling of the backups

14 Deployment models © 2013 IBM Corporation

Database patterns in PureApplication System allow you to quickly deploy an optimized DB2 database. One drawback to this model at this time, however, is there is no scalability or failover currently available for databases deployed in PureApplication System. If PureApplication System detects a failure in a database it has deployed, it will try to restart it once but there is no additional failover available. For this reason, it is not at this time recommended for 24x7 requirements. It is ideal for departmental databases and application validation in a test and integration environment. PureApplication System does provide a facility to backup your databases to Tivoli Storage Manager, however. You control the scheduling of the backups.



DB2 service for cloud: Oracle or DB2 applications

- Supports DB2 applications and Oracle applications
  - – Oracle compatibility mode is a provisioning option
- Application groups can continue to use their “Oracle database”

Database Pattern

Specify options for your database pattern.

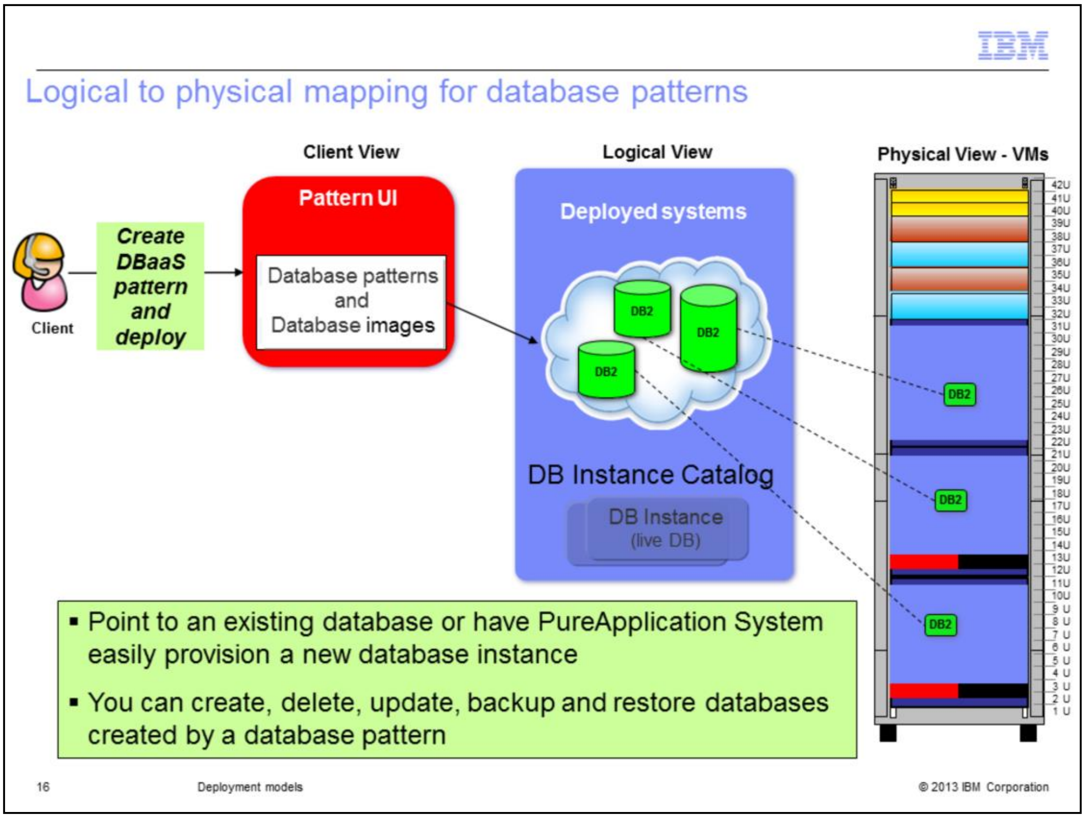
Database pattern name: untitled
Database pattern description:
Purpose: Production
Source: Apply a database workload standard

	Name	Description
●	Departmental Transactional	For databases primarily used for online transaction processing (OLTP). The database will be optimized for transactional applications.
○	Data Mart	For databases primarily used for data warehousing. The database will be optimized for reporting applications.

Database version: DB2 Version 9.7 for Linux
Database level: DB2 Version 9.7 Fix Pack 5 for Lir
Database size (GB):
Database compatibility mode: DB2 (Default)
Schema file:

15 Deployment models © 2013 IBM Corporation

In cases where you might currently need Oracle databases, PureApplication System supports an Oracle Compatibility mode, allowing you to support your Oracle applications easily as well. When defining your database pattern, you can select the Oracle compatibility mode, as shown on the slide. Using DBaaS for database deployment then allows for a single skill set for your application group.



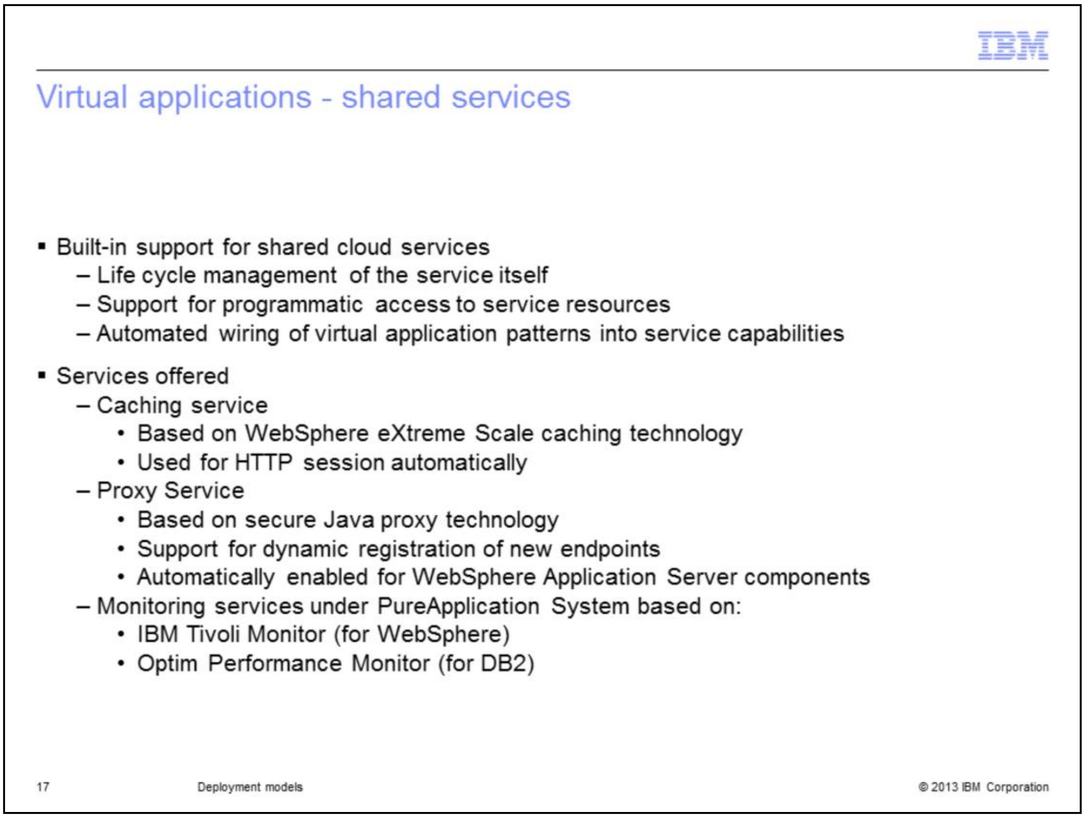
Logical to physical mapping for database patterns

Client View
Logical View
Physical View - VMs
Client
Create DBaaS pattern and deploy
Pattern UI
Database patterns and Database images
Deployed systems
DB2
DB2
DB2
DB Instance Catalog
DB Instance (live DB)
DB2
DB2
DB2
42U 41U 40U 39U 38U 37U 36U 35U 34U 33U 32U 31U 30U 29U 28U 27U 26U 25U 24U 23U 22U 21U 20U 19U 18U 17U 16U 15U 14U 13U 12U 11U 10U 9 U 8 U 7 U 6 U 5 U 4 U 3 U 2 U 1 U

- Point to an existing database or have PureApplication System easily provision a new database instance
- You can create, delete, update, backup and restore databases created by a database pattern

16 Deployment models © 2013 IBM Corporation

Looking at Database-as-a-Service from the client view, you again just need to create a pattern, this time a database pattern, and deploy it. One of the options when creating your database pattern is pointing to an existing image to “copy” it for your new deployment. The logical view shows that a new database is created for each deployment and each of those database deployments translates to a separate virtual machine created in the PureApplication System rack. Once created, you can delete, update, backup and restore the deployed databases.

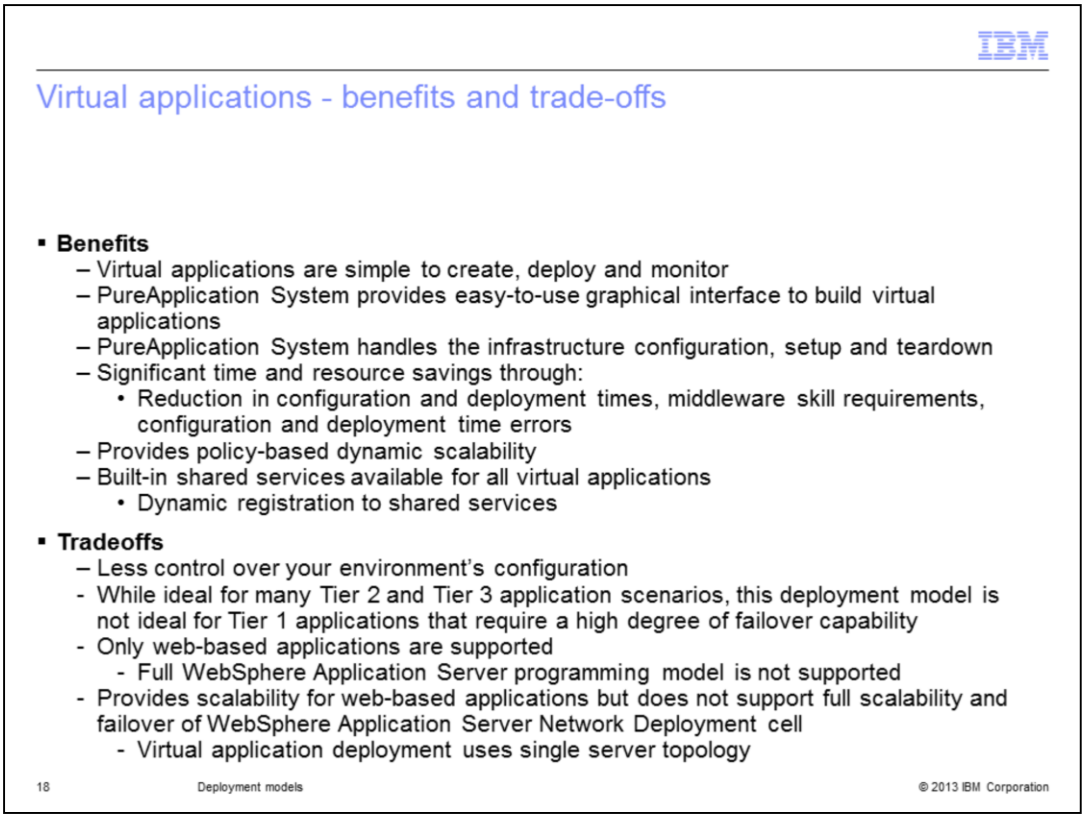


Virtual applications - shared services

- Built-in support for shared cloud services
  - – Life cycle management of the service itself
  - – Support for programmatic access to service resources
  - – Automated wiring of virtual application patterns into service capabilities
- Services offered
  - – Caching service
    - Based on WebSphere eXtreme Scale caching technology
    - Used for HTTP session automatically
  - – Proxy Service
    - Based on secure Java proxy technology
    - Support for dynamic registration of new endpoints
    - Automatically enabled for WebSphere Application Server components
  - – Monitoring services under PureApplication System based on:
    - IBM Tivoli Monitor (for WebSphere)
    - Optim Performance Monitor (for DB2)

17 Deployment models © 2013 IBM Corporation

Virtual applications have the added benefit of some built-in shared cloud services. All services are shared by virtual applications that are deployed to the same cloud group. These shared services include the caching and proxy service mentioned on a previous slide in addition to some monitoring services for various components that you might have as part of your virtual application. As mentioned previously, the caching service is based on WebSphere eXtreme Scale caching technology and, if started in the cloud, it is used automatically for HTTP sessions in your virtual applications. The elastic load balancer proxy shared service provides a front end to virtual applications in the cloud by balancing the load across the deployed instances of your virtual applications. When the elastic load balancer (ELB) service is deployed and running in a cloud group, all new deployments of virtual applications are auto-wired to use ELB automatically upon virtual application deployment. There is also a shared monitoring service that allows you to easily monitor your various virtual application components, including DB2 and WebSphere Application Server components. As shown on the slide, the monitoring services are based on IBM Tivoli Monitor technology for WebSphere components and on Optim Performance Monitor technology for database components. Your virtual applications automatically take advantage of these services as needed when they are started in the cloud you deploy your virtual application to. PureApplication System will take care of the life cycle management of these shared services, restarting them as needed on failure, for instance.



IBM

Virtual applications - benefits and trade-offs

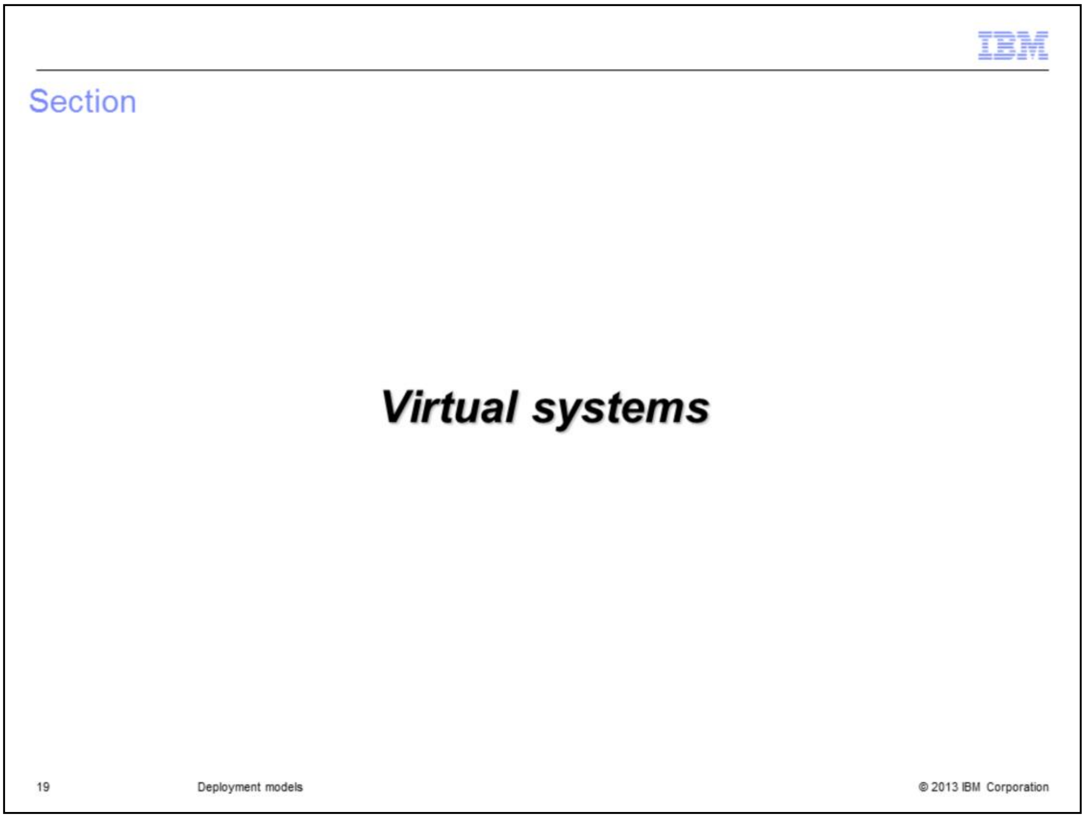
- **Benefits**
  - Virtual applications are simple to create, deploy and monitor
  - PureApplication System provides easy-to-use graphical interface to build virtual applications
  - PureApplication System handles the infrastructure configuration, setup and teardown
  - Significant time and resource savings through:
    - Reduction in configuration and deployment times, middleware skill requirements, configuration and deployment time errors
  - Provides policy-based dynamic scalability
  - Built-in shared services available for all virtual applications
    - Dynamic registration to shared services
- **Tradeoffs**
  - Less control over your environment's configuration
  - While ideal for many Tier 2 and Tier 3 application scenarios, this deployment model is not ideal for Tier 1 applications that require a high degree of failover capability
  - Only web-based applications are supported
    - Full WebSphere Application Server programming model is not supported
  - Provides scalability for web-based applications but does not support full scalability and failover of WebSphere Application Server Network Deployment cell
    - Virtual application deployment uses single server topology

18 Deployment models © 2013 IBM Corporation

While virtual applications are the quickest way to deploy applications and the most cost effective deployment model, it does have trade-offs which might offset some of the benefits for you. Your usage scenario determines if the benefits outweigh the tradeoffs. Some benefits of using virtual applications include not having to worry about the infrastructure configuration. PureApplication System handles that for you, reducing much of the complexity. This reduces the required time needed to configure and deploy your applications. Because PureApplication System handles the environment setup for you in addition to monitoring its health, there is also a decrease in middleware skills requirements. Having PureApplication System handle the configuration also cuts down on configuration and deployment time errors, which again helps reduce your application deployment time.

One potential tradeoff with the virtual application deployment model is the lack of control over your environment's configuration. If you require a specific configuration for your application you might not be able to use the virtual applications deployment model because of the limited number of customizations available. In addition, while there is some failover capability built into virtual applications, the database components have limited failover. As a result, it is not ideal for Tier 1 applications at this time where a high degree of failover capability is required. Currently, only web-based applications are supported in the virtual application deployment model. Since the full WebSphere Application Server programming model is not supported, you also might not be able to use virtual applications for your deployments if your application uses some unsupported functions. As an example, remote enterprise beans are not supported in the virtual application deployment model. Finally, while the virtual application deployment model provides scalability for your web-based applications, it does this with a single server topology. As a result, the full scalability available in the Network Deployment configuration is not available. This means that quality of service (QOS) features such as the HA manager, core groups and DRS are not supported in this environment.

So again, you need to look at your particular application to see if you are able to take advantage of this particular deployment model.

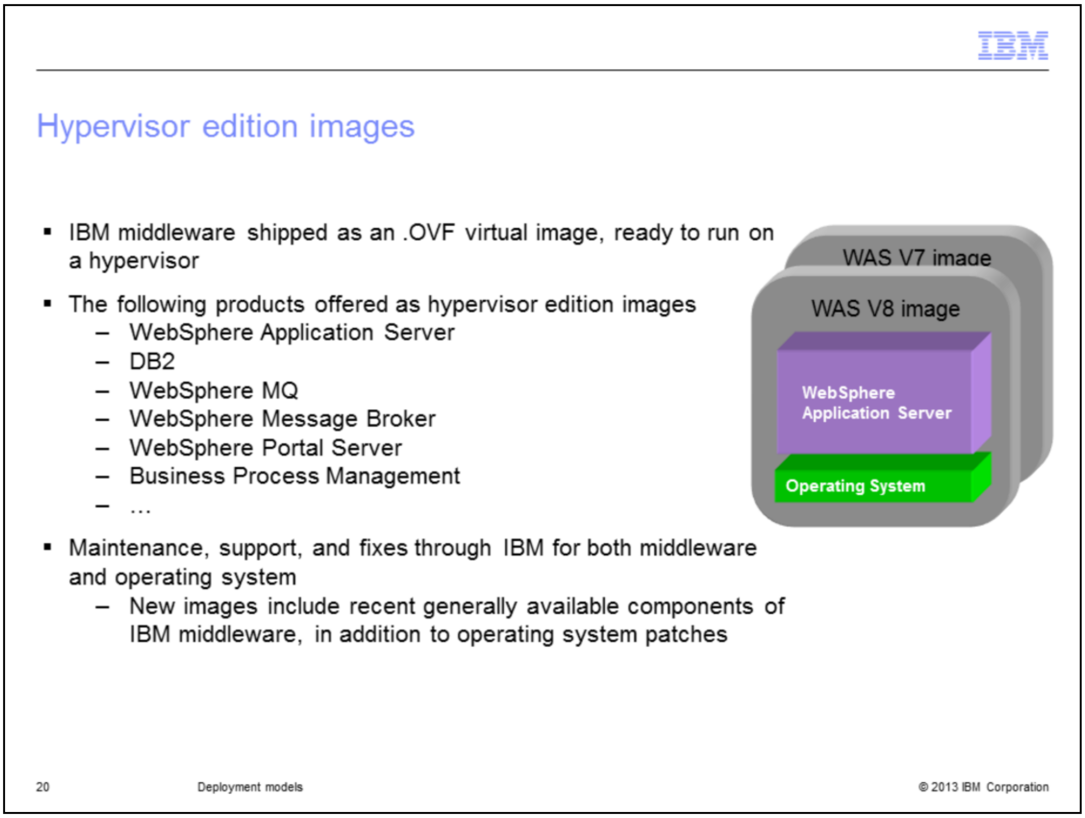


Section

Virtual systems

19 Deployment models © 2013 IBM Corporation

In this next section, you will review the virtual system deployment model.



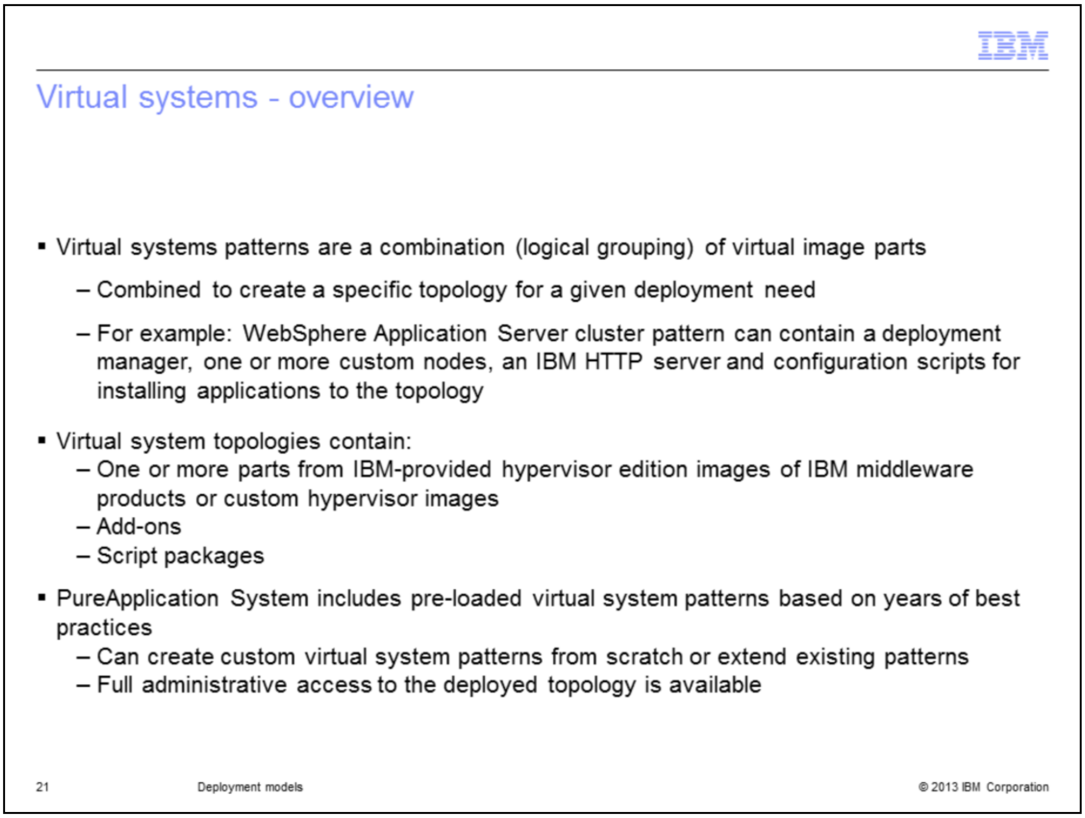
IBM

Hypervisor edition images

- IBM middleware shipped as an .OVF virtual image, ready to run on a hypervisor
- The following products offered as hypervisor edition images
  - WebSphere Application Server
  - DB2
  - WebSphere MQ
  - WebSphere Message Broker
  - WebSphere Portal Server
  - Business Process Management
  - …
- Maintenance, support, and fixes through IBM for both middleware and operating system
  - New images include recent generally available components of IBM middleware, in addition to operating system patches

20 Deployment models © 2013 IBM Corporation

Previously, if you wanted to use WebSphere Application Server in a virtualization context, you were forced to build and own these images, which involved managing two parallel code paths: the operating system and the middleware. Now, IBM is building and supporting the entire virtual image for you in the form of hypervisor edition images. Hypervisor edition images contain middleware that is shipped as an .OVF virtual image that is ready to run on a hypervisor. The slide shows a partial list of hypervisor edition images that are available from IBM. They include WebSphere Application Server, DB2 and WebSphere MQ, just to name a few. With the hypervisor edition images, maintenance and fixes are provided by IBM for both the middleware and operating system components.



Virtual systems - overview

- Virtual systems patterns are a combination (logical grouping) of virtual image parts
  - Combined to create a specific topology for a given deployment need
  - For example: WebSphere Application Server cluster pattern can contain a deployment manager, one or more custom nodes, an IBM HTTP server and configuration scripts for installing applications to the topology
- Virtual system topologies contain:
  - One or more parts from IBM-provided hypervisor edition images of IBM middleware products or custom hypervisor images
  - Add-ons
  - Script packages
- PureApplication System includes pre-loaded virtual system patterns based on years of best practices
  - Can create custom virtual system patterns from scratch or extend existing patterns
  - Full administrative access to the deployed topology is available

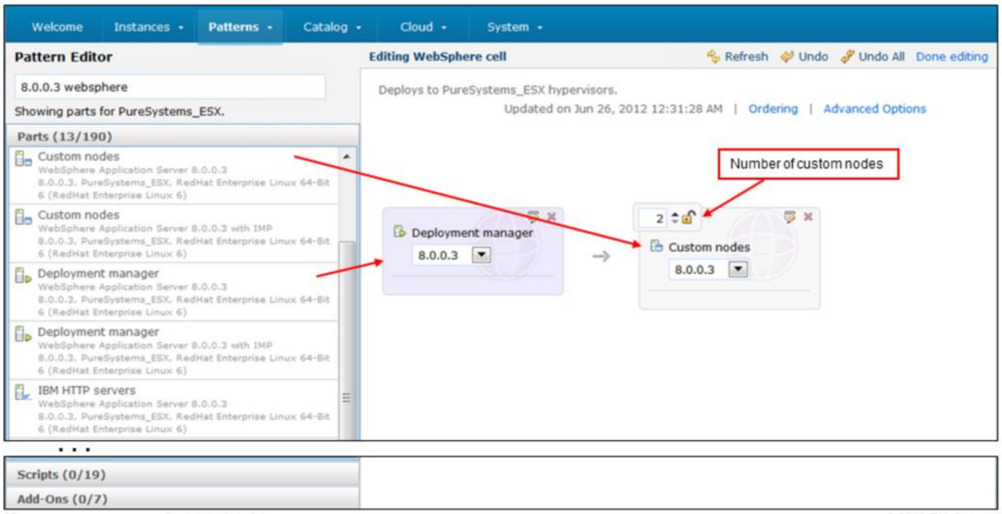
21 Deployment models © 2013 IBM Corporation

Whereas virtual application patterns are focused on the application artifacts, virtual system patterns focus on the topology. With virtual systems, you combine various virtual image parts to create a specific topology where your application can then be deployed. An example WebSphere Application Server topology can contain an HTTP server, a deployment manager and multiple custom nodes, each which might be at different versions. You can also include configuration scripts that will run during the deployment to further configure your topology and install applications. You can also add add-ons which allow you to add additional disks or users to your deployment.

PureApplication System includes some pre-loaded virtual system patterns that are based on years of experience and incorporate best practices in the topologies they define. You can use these pre-loaded patterns as-is, extend them to meet your particular needs or create your own custom patterns from scratch. One big difference in the virtual system deployment model is that you have full administrative access to the deployed middleware. As an example, while in the virtual application deployment model, you cannot access the WebSphere administration console, it is fully available to you in the virtual system deployment model.

Virtual system pattern assembly user interface

- Virtual system pattern assembly by dragging from the palette over to the canvas
  - Virtual image parts
  - Script packages
  - Add-ons



8.0.0.3 websphere

Showing parts for PureSystems_ESX.

Deploys to PureSystems_ESX hypervisors.

Updated on Jun 26, 2012 12:31:28 AM

Custom nodes
WebSphere Application Server 8.0.0.3
8.0.0.3, PureSystems_ESX, RedHat Enterprise Linux 64-Bit 6 (RedHat Enterprise Linux 6)

Custom nodes
WebSphere Application Server 8.0.0.3 with IMP
8.0.0.3, PureSystems_ESX, RedHat Enterprise Linux 64-Bit 6 (RedHat Enterprise Linux 6)

Deployment manager
WebSphere Application Server 8.0.0.3
8.0.0.3, PureSystems_ESX, RedHat Enterprise Linux 64-Bit 6 (RedHat Enterprise Linux 6)

Deployment manager
WebSphere Application Server 8.0.0.3 with IMP
8.0.0.3, PureSystems_ESX, RedHat Enterprise Linux 64-Bit 6 (RedHat Enterprise Linux 6)

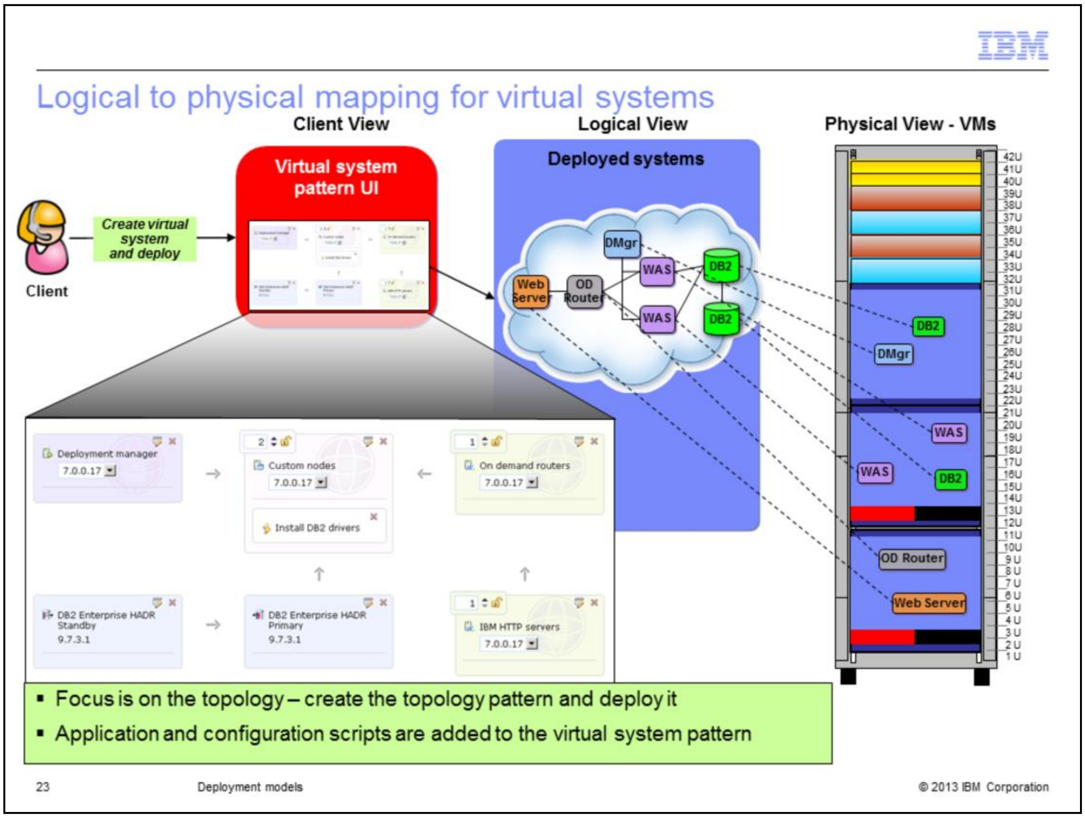
IBM HTTP servers
WebSphere Application Server 8.0.0.3
8.0.0.3, PureSystems_ESX, RedHat Enterprise Linux 64-Bit 6 (RedHat Enterprise Linux 6)

. . .

22 Deployment models © 2013 IBM Corporation

Shown on the slide, the virtual system pattern editor is an easy drag-and-drop interface to create your virtual system topology. If this is a new pattern you are presented with a blank canvas on the right and a palette of virtual image parts on the left. The palette on the left contains your virtual image parts, script packages and add-ons which you will use to assemble your pattern. You create a pattern by dragging parts from the palette and dropping them onto the canvas. In order to add multiple parts at the same configuration point, you increment the number located in the upper left corner of the part when available. There is also an option to choose which virtual image will make up this pattern. The virtual image will dictate the available virtual image parts that you can use to make up your pattern.

The example on this slide shows three nodes consisting of a deployment manager node managing two custom nodes.



Logical to physical mapping for virtual systems

Deployment manager
7.0.0.17

2
Custom nodes
7.0.0.17
Install DB2 drivers

1
On demand routers
7.0.0.17

DB2 Enterprise HADR Standby
9.7.3.1

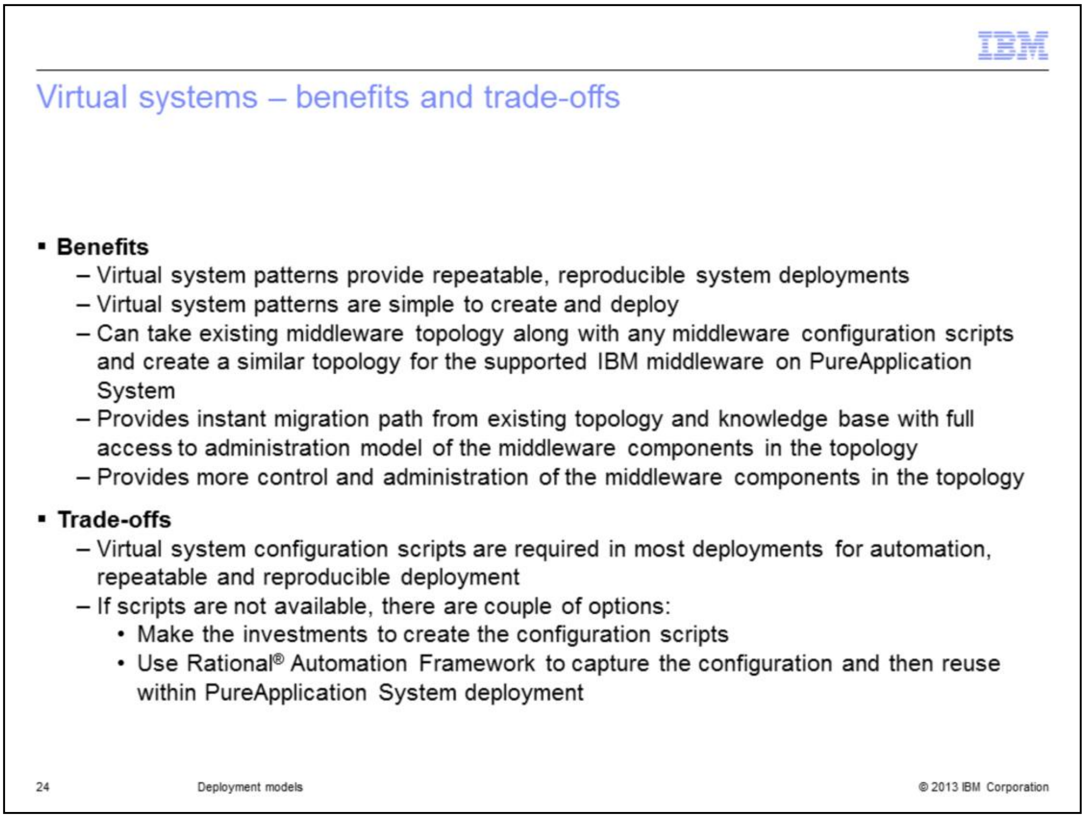
DB2 Enterprise HADR Primary
9.7.3.1

1
IBM HTTP servers
7.0.0.17

- Focus is on the topology – create the topology pattern and deploy it
- Application and configuration scripts are added to the virtual system pattern

23 Deployment models © 2013 IBM Corporation

Showing the virtual system deployment model from the client view, you see that you just need to create a virtual system pattern and deploy it. Rather than focusing on the application here though, you will focus instead on the topology of the system to be deployed. In order to deploy an application in this model, you need to provide application and configuration scripts rather than letting PureApplication System do it for you automatically. The virtual system pattern here is rather complicated and as such translates in the logical view to seven distinct instances. There is an instance created for each part contained in your virtual system pattern. Here this includes a deployment manager, two custom nodes, an on-demand router, a DB2 standby instance, a DB2 primary instance and an HTTP server. Finally, this translates to seven distinct virtual machines that are created for you with the associated virtual image middleware in the PureApplication System rack. Unlike virtual application deployments, you determined this topology very explicitly in your pattern definition.



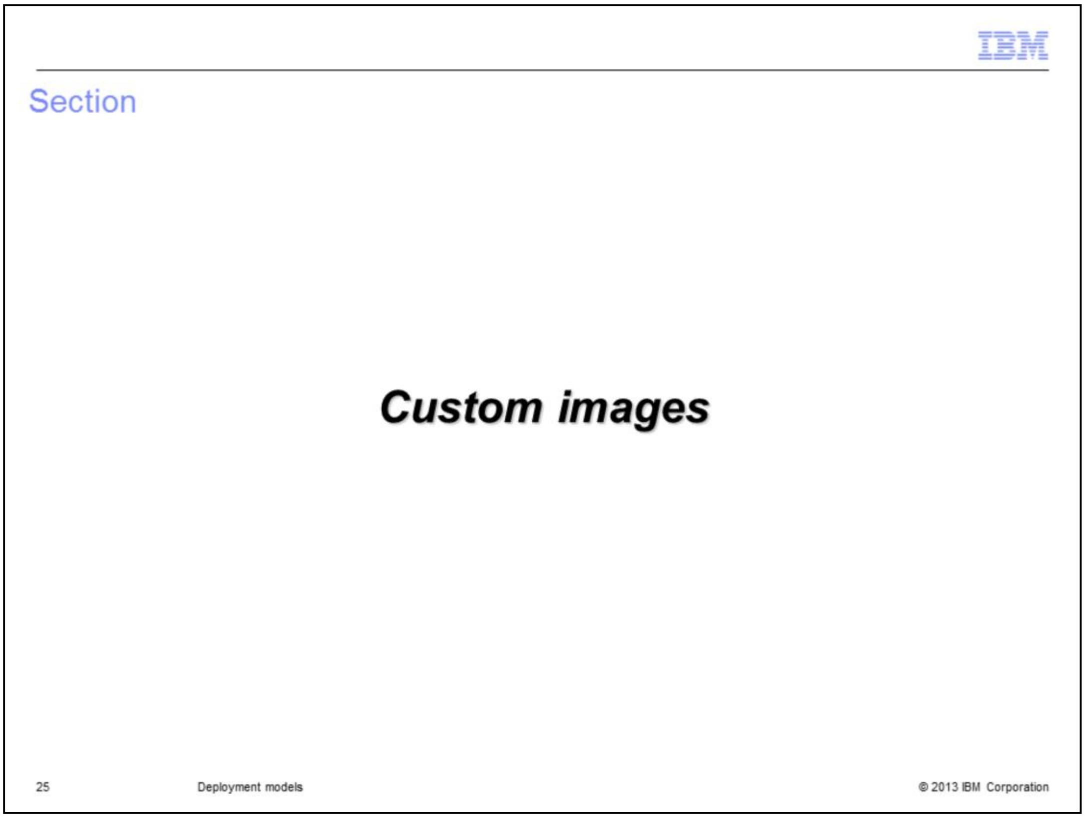
Virtual systems – benefits and trade-offs

- **Benefits**
  - Virtual system patterns provide repeatable, reproducible system deployments
  - Virtual system patterns are simple to create and deploy
  - Can take existing middleware topology along with any middleware configuration scripts and create a similar topology for the supported IBM middleware on PureApplication System
  - Provides instant migration path from existing topology and knowledge base with full access to administration model of the middleware components in the topology
  - Provides more control and administration of the middleware components in the topology
- **Trade-offs**
  - Virtual system configuration scripts are required in most deployments for automation, repeatable and reproducible deployment
  - If scripts are not available, there are couple of options:
    - Make the investments to create the configuration scripts
    - Use Rational® Automation Framework to capture the configuration and then reuse within PureApplication System deployment

24 Deployment models © 2013 IBM Corporation

One of the primary benefits of virtual system deployments is that it provides a repeatable, reproducible system deployment for you with the click of a button. The virtual system patterns are simple to create and deploy but it does take a little more work on your part than a virtual application deployment. In order to install your applications during the virtual system deployment, you need to provide script packages that you have created. With the virtual system deployments, however, you can take your existing middleware topologies and reproduce it as a virtual system pattern. This is possible because the virtual system deployment model allows for far more customization than the virtual application deployment model. In the virtual system model, you have full access to the administration model of the middleware components in the topology which provides more control and the ability for you to manage the middleware components in the topology.

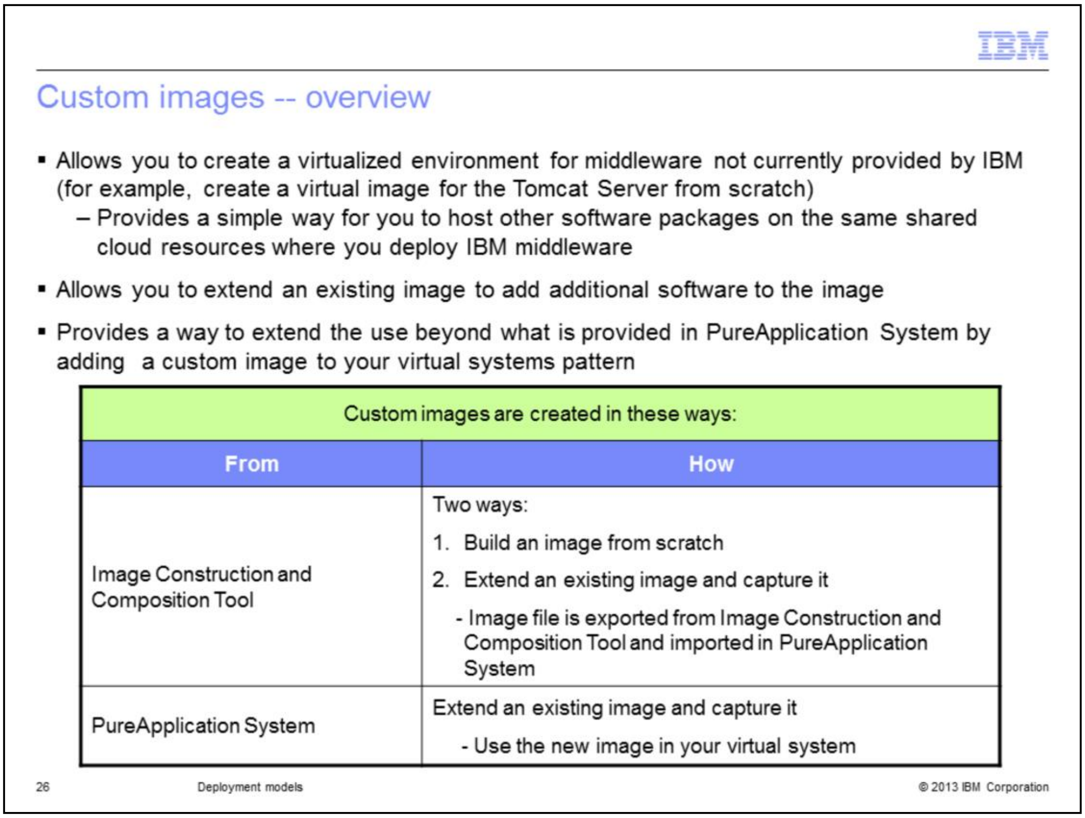
Unlike virtual applications where you do not have the ability to do much customization, virtually all virtual system deployments will require script packages in order to automate the configuration of your virtual system and the deployment repeatable. This might require an investment on your part to create these script packages. Another option is to use Rational Automation Framework to capture an existing configuration that can then be reused for PureApplication virtual system deployments.



Section

Custom images

This next section will introduce you to custom images which allow you to create virtual images for software not currently provided by IBM.



IBM

Custom images -- overview

- Allows you to create a virtualized environment for middleware not currently provided by IBM (for example, create a virtual image for the Tomcat Server from scratch)
  - Provides a simple way for you to host other software packages on the same shared cloud resources where you deploy IBM middleware
- Allows you to extend an existing image to add additional software to the image
- Provides a way to extend the use beyond what is provided in PureApplication System by adding a custom image to your virtual systems pattern

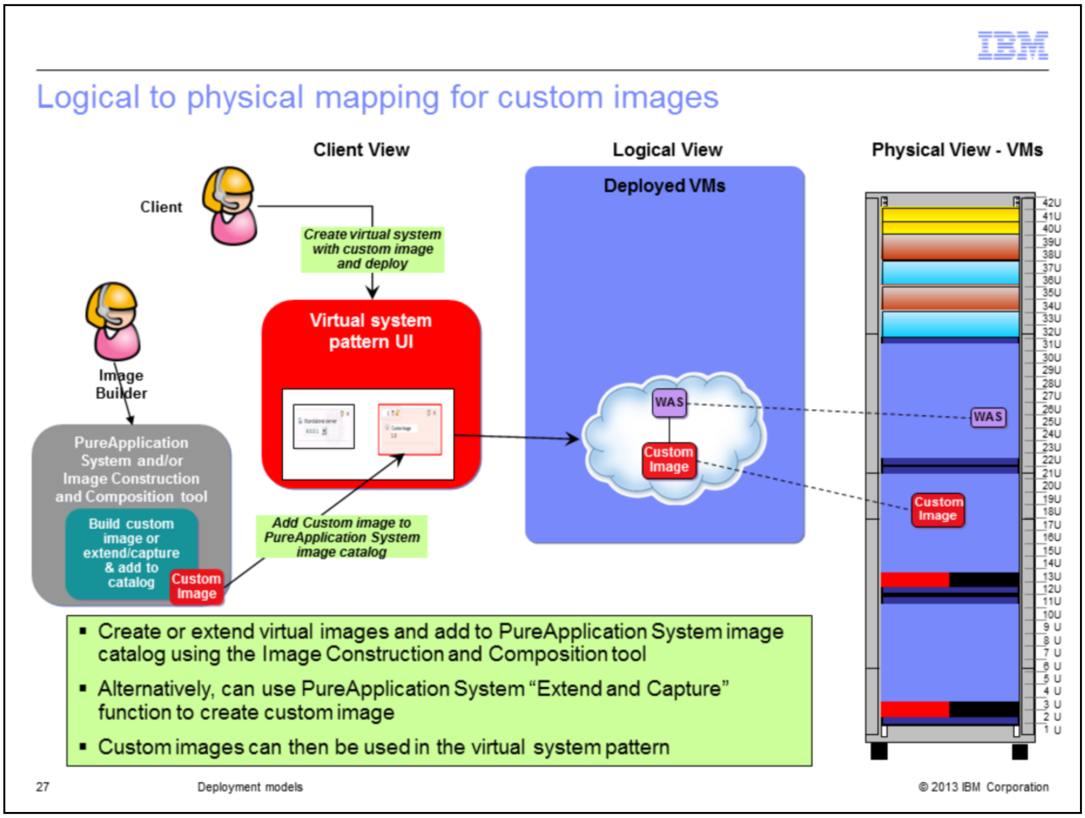
Custom images are created in these ways:

From	How
Image Construction and Composition Tool	Two ways: 1. Build an image from scratch 2. Extend an existing image and capture it - Image file is exported from Image Construction and Composition Tool and imported in PureApplication System
PureApplication System	Extend an existing image and capture it - Use the new image in your virtual system

26 Deployment models © 2013 IBM Corporation

Custom images can be deployed as virtual system patterns in PureApplication System. Custom images allow you to create virtualized environments for middleware that is not currently available as hypervisor edition images from IBM or extend a provided image with additional software that is required. To create a new image from scratch, you can use the Image Construction and Composition Tool which is downloadable from PureApplication System on the Welcome Page of the administration console. A virtual application deployment for the Image Construction and Composition tool is also available in PureApplication System. The virtual application deployment gives you a fully-functional Image Construction and Composition tool instance at the push of a button. With the download option of the tool, you need to install it yourself in a Linux environment somewhere.

Custom images allow you to run vendor software using the same shared cloud resources being used for your IBM middleware deployments. To extend an existing virtual image with additional software, you can also use the “extend and capture” functionality found in PureApplication Systems.



Logical to physical mapping for custom images

Client View
Logical View
Physical View - VMs
Client
Create virtual system with custom image and deploy
Virtual system pattern UI
Deployed VMs
WAS
Custom Image
Image Builder
PureApplication System and/or Image Construction and Composition tool
Build custom image or extend/capture & add to catalog
Custom Image
Add Custom image to PureApplication System image catalog
WAS
Custom Image
42U
41U
40U
39U
38U
37U
36U
35U
34U
33U
32U
31U
30U
29U
28U
27U
26U
25U
24U
23U
22U
21U
20U
19U
18U
17U
16U
15U
14U
13U
12U
11U
10U
9 U
8 U
7 U
6 U
5 U
4 U
3 U
2 U
1 U

- Create or extend virtual images and add to PureApplication System image catalog using the Image Construction and Composition tool
- Alternatively, can use PureApplication System "Extend and Capture" function to create custom image
- Custom images can then be used in the virtual system pattern

27 Deployment models © 2013 IBM Corporation

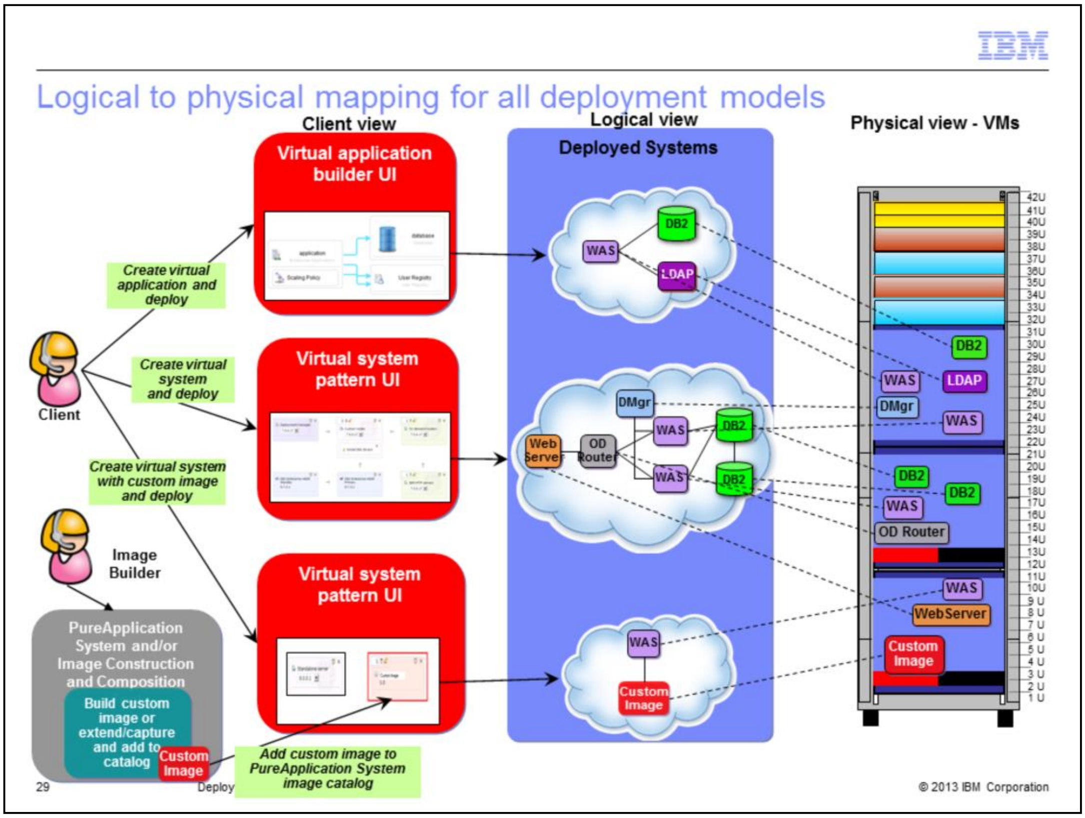
Using the Image Construction and Composition Tool, an image builder who is familiar with the software that you require a virtual image for can build a custom image that can then be deployed into the PureApplication System cloud. Another option to build a custom image is using the extend/capture functionality that is available for the catalog images on PureApplication System. Either option is highly integrated with the PureApplication System. Once you have a custom image created and available in the image catalog on the system, you can easily create a virtual system pattern and deploy it into the cloud. You see in the example, a custom image part has been included in a pattern with a WebSphere stand-alone part. This translates to two instances, one for your custom image and one for WebSphere. Each of these instances then translate to a separate virtual machine on the physical rack.

Section

Bringing it all together

28 Deployment models © 2013 IBM Corporation

In this next section, you will receive a glimpse of the different options in one place.



Logical to physical mapping for all deployment models

Client view

Logical view

Physical view - VMs

Deployed Systems

Virtual application builder UI

Create virtual application and deploy

Virtual system pattern UI

Create virtual system and deploy

Create virtual system with custom image and deploy

Virtual system pattern UI

Client

Image Builder

PureApplication System and/or Image Construction and Composition

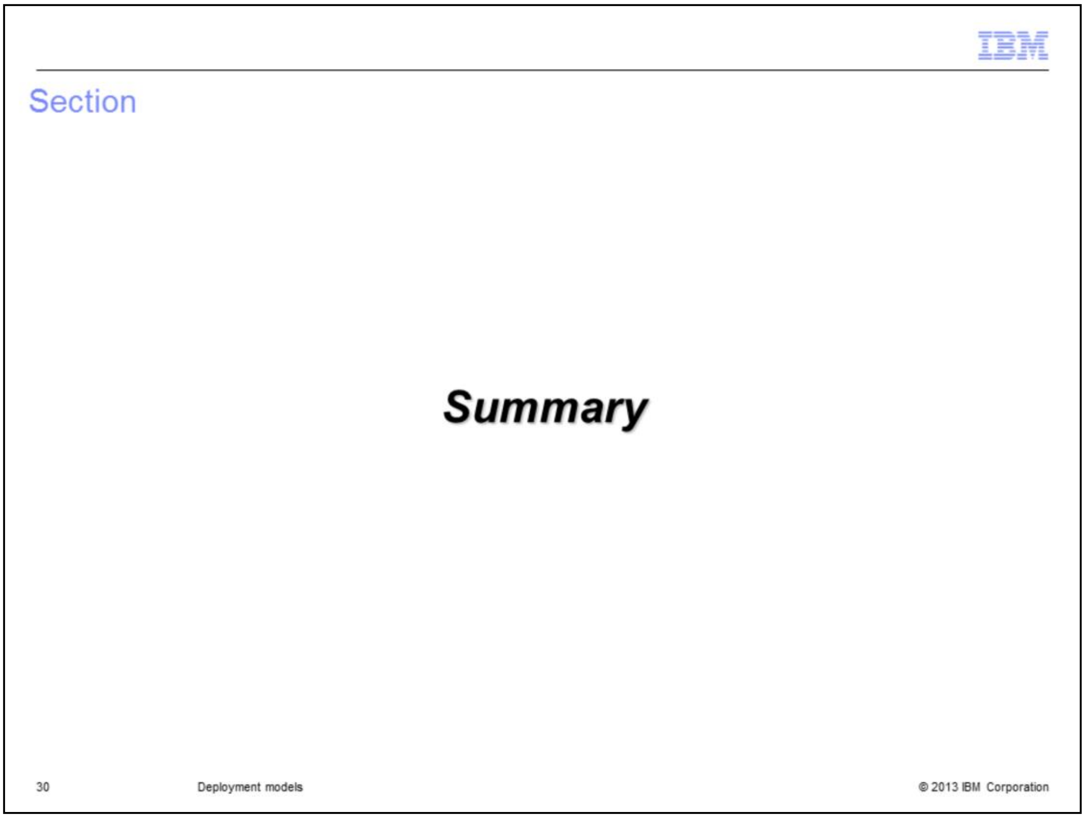
Build custom image or extend/capture and add to catalog

Add custom image to PureApplication System image catalog

29 Deploy

© 2013 IBM Corporation

Here you see the two deployment models, virtual applications and virtual systems in one place. While both options give you an easy-to-use interface for pattern creation, you determine the topology of the deployed system in the virtual system deployment model. With the virtual application deployment model, you are allowing PureApplication System to make the topology decisions for you, giving up some control and customization. The bottom example on the slide is showing the custom image option which gives you flexibility on what middleware you have available to you for deployment. You see though that all of these can coexist in the PureApplication System, giving you a lot of flexibility with your deployments, allowing you to choose different models depending on your application needs.

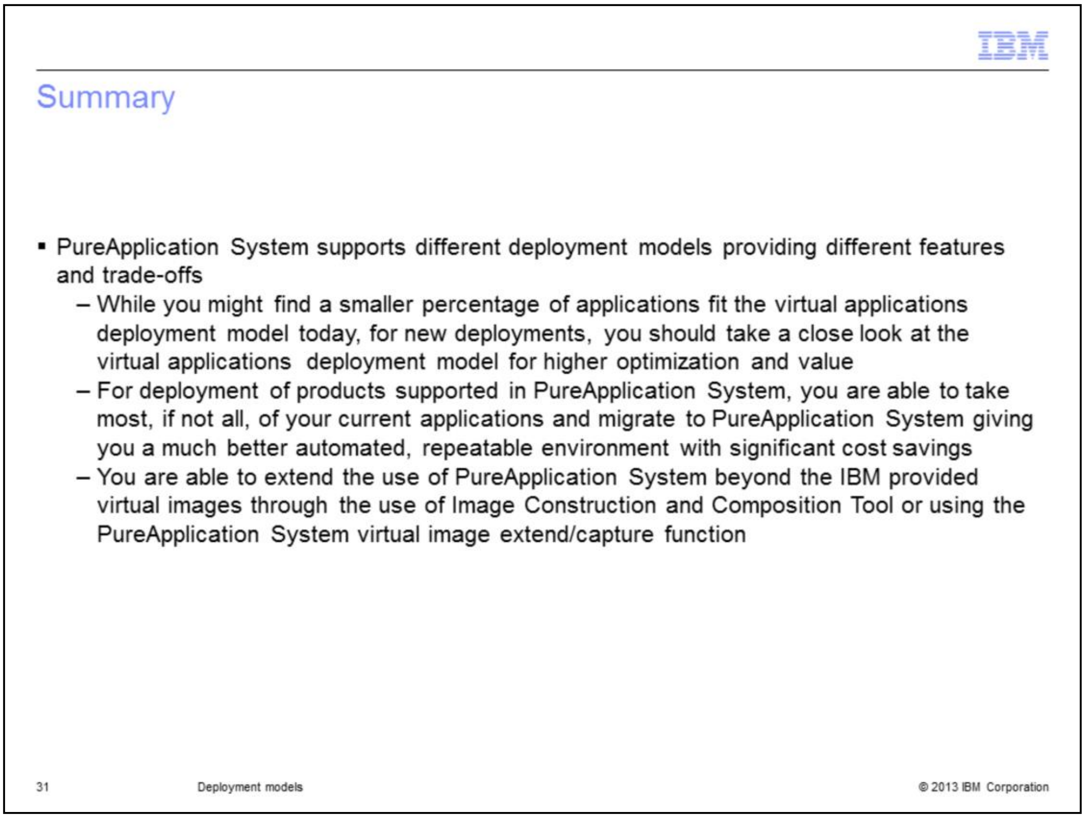


Section

Summary

30 Deployment models © 2013 IBM Corporation

The next section will provide a quick summary of the deployment models.

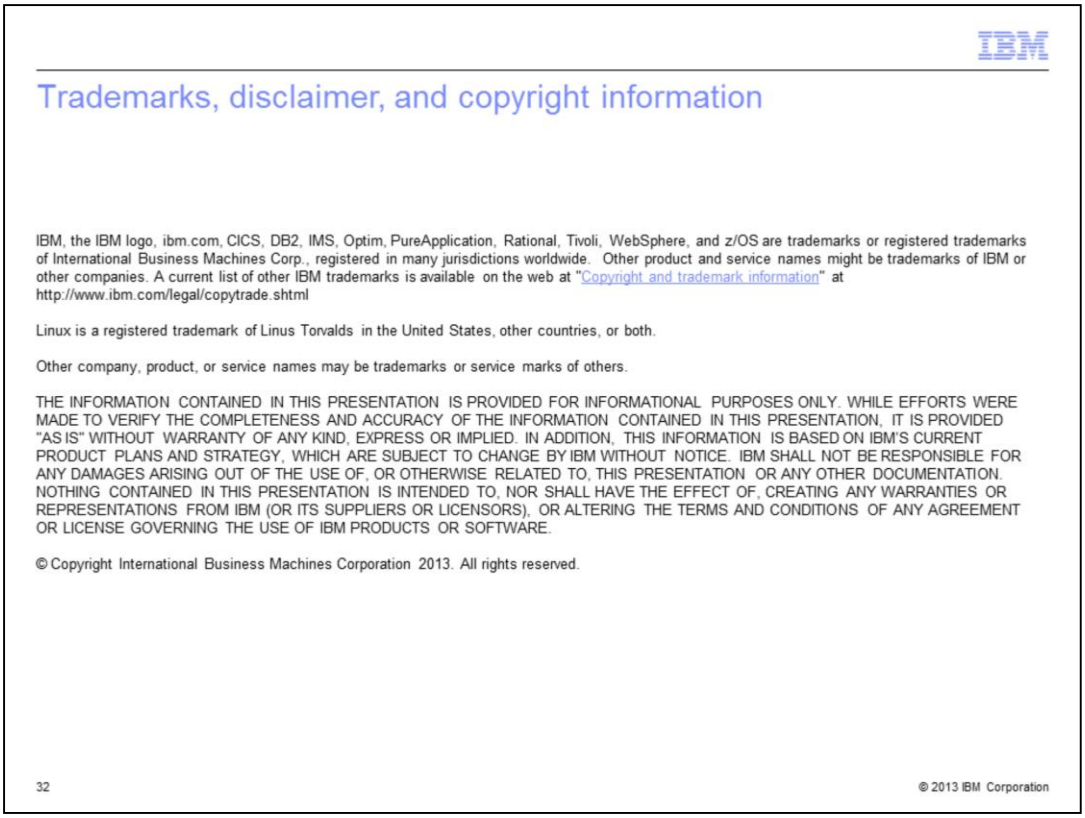


Summary

- PureApplication System supports different deployment models providing different features and trade-offs
  - While you might find a smaller percentage of applications fit the virtual applications deployment model today, for new deployments, you should take a close look at the virtual applications deployment model for higher optimization and value
  - For deployment of products supported in PureApplication System, you are able to take most, if not all, of your current applications and migrate to PureApplication System giving you a much better automated, repeatable environment with significant cost savings
  - You are able to extend the use of PureApplication System beyond the IBM provided virtual images through the use of Image Construction and Composition Tool or using the PureApplication System virtual image extend/capture function

31 Deployment models © 2013 IBM Corporation

As you just saw on the last picture, PureApplication System supports different deployment models, allowing you to select one based on your application needs. You saw some of the benefits and trade-offs of the different deployment models in this presentation. While your current applications might not fit in the virtual application model, as you start new projects, you might be able to take advantage of its benefits. Those benefits include higher optimization and quicker time to value. For existing applications that do not fit in the virtual application model, you can use the virtual systems deployment model and achieve a highly automated, repeatable environment allowing for significant cost savings. Finally, if you have applications using middleware not currently available as IBM hypervisor edition images, you have the option to create custom images. You can create custom images with the Image Construction and Composition Tool or by using the extend/capture function available in PureApplication System.



Trademarks, disclaimer, and copyright information

IBM, the IBM logo, ibm.com, CICS, DB2, IMS, Optim, PureApplication, Rational, Tivoli, WebSphere, and z/OS are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml

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

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

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.