Course Guide

WebSphere Application Server V9 Administration in a Federated Environment

Course code WA599 / ZA599 ERC 1.0

IBM Training

November 2016 edition

Notices

This information was developed for products and services offered in the US.

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

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

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

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

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

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 actual people or business enterprises is entirely coincidental.

Trademarks

IBM, the IBM logo, and ibm.com 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 IBM trademarks is available on the web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml.

Contents

Trademarks

The reader should recognize that the following terms, which appear in the content of this training document, are official trademarks of IBM or other companies:

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide.

The following are trademarks of International Business Machines Corporation, registered in many jurisdictions worldwide:

AIX®	DataPower®	DB™
DB2®	developerWorks®	Express®
HACMP™	OS/400®	Rational®
Redbooks®	Tivoli®	WebSphere®
z/OS®		

Intel is a trademark or registered trademark of Intel Corporation or its subsidiaries in the United States and other countries.

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

Windows is a trademark 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.

VMware and the VMware "boxes" logo and design, Virtual SMP and VMotion are registered trademarks or trademarks (the "Marks") of VMware, Inc. in the United States and/or other jurisdictions.

Social® is a trademark or registered trademark of TWC Product and Technology, LLC, an IBM Company.

Other product and service names might be trademarks of IBM or other companies.

Course description

WebSphere Application Server V9 Administration in a Federated Environment

Duration: 1.5 days

Purpose

This course teaches you the skills that are needed to administer IBM WebSphere Application Server V9 in a federated environment.

This release of IBM WebSphere Application Server provides enhanced support for standards (notably Java 7 EE), emerging technology, and a choice of development frameworks.

In this course, you learn how to configure and maintain IBM WebSphere Application Server V9 Network Deployment. You learn how to deploy and create a deployment manager and federate a cell. In addition, you learn how to create a cluster within the federated cell.

Throughout the course, hands-on exercises and demonstrations reinforce lecture content. You gain practical experience with WebSphere Application Server V9 by completing tasks such as creating a deployment manager, federating a stand-alone application server, creating a custom profile, and clustering an existing application server.

Audience

This course is designed for WebSphere administrators who have experience with stand-alone application server environments, and want to learn about creating and managing a federated environment.

Prerequisites

- Basic operational skills for the Linux operating system
- Administrative skills for a web server, such as IBM HTTP Server or Apache
- Basic understanding of cloud concepts, private, public, and hybrid clouds, and specifically traditional on-premises environments
- Completion of course *WebSphere Application Server V9 Administration* (WA590G) or *WebSphere Application Server V9 Administration* (ZA590G), or experience with WebSphere Application Server in a stand-alone environment

Objectives

- Describe the architectural concepts that are related to WebSphere Application Server Network Deployment
- Create a deployment manager instance

- Federate an application server to a cell
- Add a stand-alone application server to a WebSphere Application Server cell
- Cluster an application server within a WebSphere Application Server cell
- Configure WebSphere Application Server SSL security settings
- Deploy applications in clustered environments
- Describe the features of Intelligent Management

Contents

- Intermediate training for administrators
- How to federate a WebSphere Application Server environment
- How to create WebSphere Application Server clusters

Agenda

Note

The following unit and exercise durations are estimates, and might not reflect every class experience.

Day 1

(00:15) Course introduction
(01:00) Unit 1. WebSphere Application Server architecture: Federated
(01:00) Unit 2. Federating a cell
(00:20) Exercise 1. Configuring the lab workstation
(01:30) Exercise 2. Creating a federated cell
(01:30) Unit 3. Workload management
(01:00) Exercise 3. Clustering and workload management

Day 2

(01:30) Unit 4. WebSphere security: SSL
(01:00) Exercise 4. Configuring SSL for WebSphere
(01:00) Unit 5. Overview of Intelligent Management
(00:15) Unit 6. Course summary

Unit 1. WebSphere Application Server architecture: Federated

Estimated time

01:00

Overview

This unit describes the concepts and terminology of a federated environment.

How you will check your progress

- Review questions

References

WebSphere Application Server V9 Knowledge Center

https://www.ibm.com/support/knowledgecenter/SSEQTP_9.0.0/as_ditamaps/was900_welcome_base.html

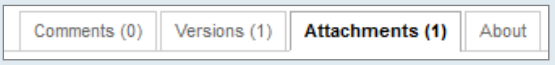
IBM Training

How to check online for course material updates

Note: If your classroom does not have Internet access, ask your instructor for more information.

Instructions

1. Enter this URL in your browser: **ibm.biz**/CloudEduCourses.
2. Find the product category for your course, and click the link to view all products and courses.
3. Find your course in the course list and then click the link.
4. The wiki page displays information for the course. If a course corrections document is available, this page is where it is found.
5. If you want to download an attachment, such as a course corrections document, click the **Attachments** tab at the bottom of the page.



6. To save the file to your computer, click the document link and follow the prompts.

Figure 1-1. How to check online for course material updates

IBM Training

Unit objectives

- Describe the Network Deployment runtime flow
- Describe Network Deployment concepts and terminology, such ascell,node,node agent, anddeployment manager
- Describe the Network Deployment administration flow
- Explain how to manage web servers from WebSphere Application Server

Figure 1-2. Unit objectives

IBM Training

Topics

- Network deployment concepts
- Managing web servers
- Security domains
- Fine grained access
- Additional concepts

Figure 1-3. Topics

1.1. Network deployment concepts

IBM Training

Network deployment concepts

Figure 1-4. Network deployment concepts

Version 9 packaging

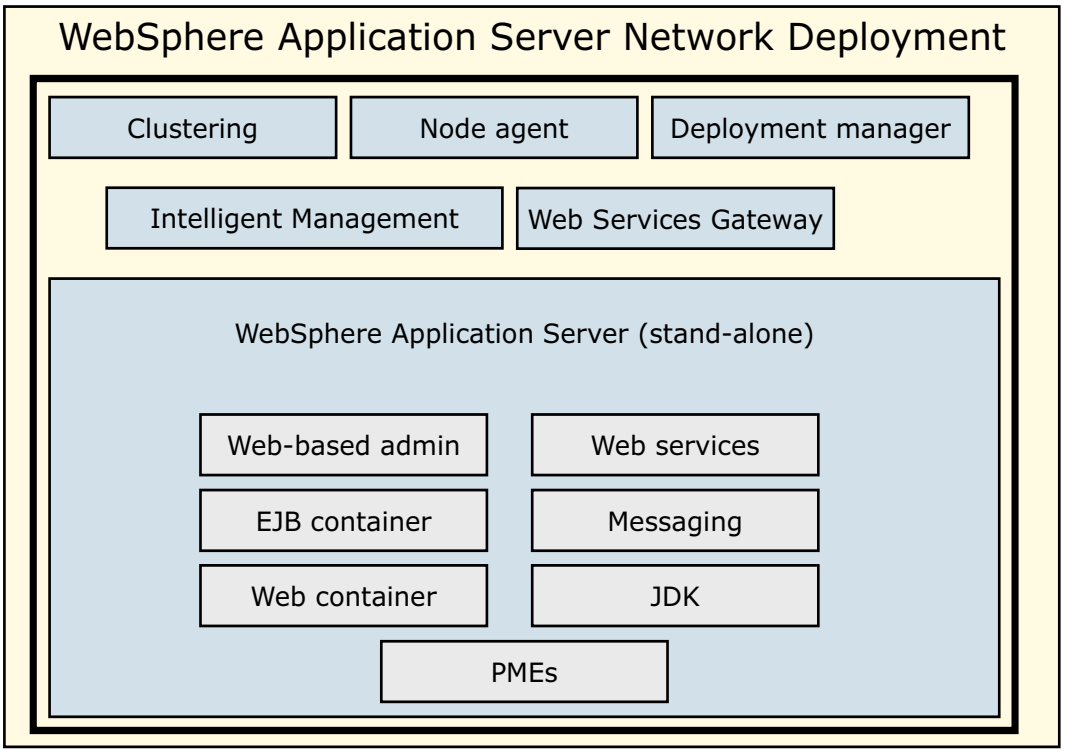


Figure 1-5. Version 9 packaging

This unit focuses on the Network Deployment version of WebSphere Application Server.

Network deployment concepts

- A *deployment manager* (dmgr) process manages the node agents
  - Holds the configuration repository for the entire management domain, called a *cell*
  - Within a cell, the administrative console runs inside the dmgr
- A *node* is a logical grouping of application servers
  - A single *node agent* process manages each node
  - Multiple nodes can exist on a single machine by using profiles

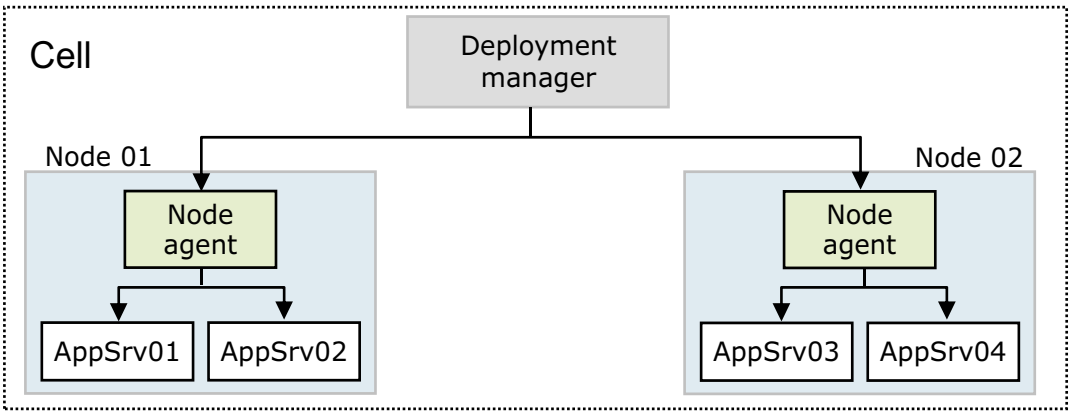


Figure 1-6. Network deployment concepts

The deployment manager is an application server that manages the administrative environment within a cell. As you see later in this unit, a node is represented as a profile. The node agent is an important process that allows for communication of administrative information, such as commands and configuration files, to reach the application servers.

IBM Training

Network deployment runtime flow

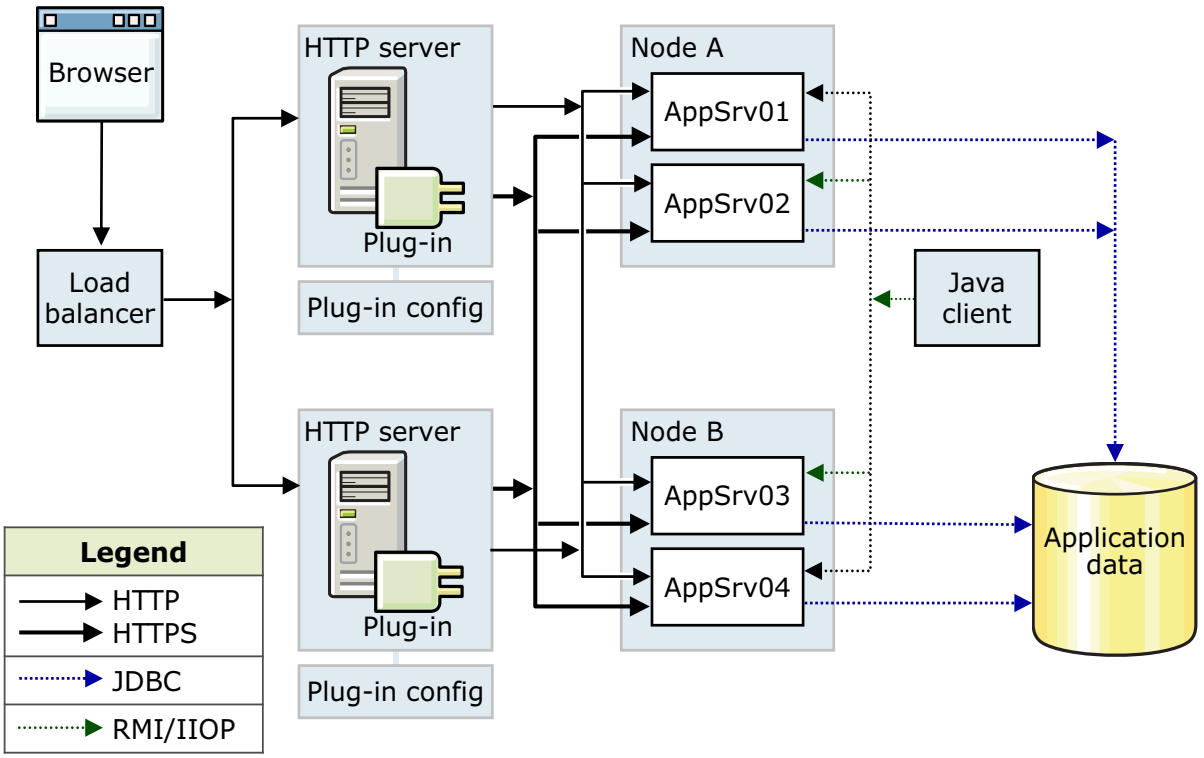


Figure 1-7. Network deployment runtime flow

The main theme with Network Deployment is distributed applications. While the “flow” of an application remains the same, there are significant additions to the runtime of an application. Note the “load balancer”: it allows for multiple HTTP servers. Users point their browsers to the load balancer, and their requests are workload managed to an HTTP server. When a request arrives at one of these HTTP servers, the HTTP server plug-in load balances the request between the application servers that it is configured to serve. When the request enters the application server, the flow is identical to how it was in Express and Base.

The Java client requests to EJBs can also be workload managed so that the requests do not all arrive at one application server.

IBM Training

Administration flow

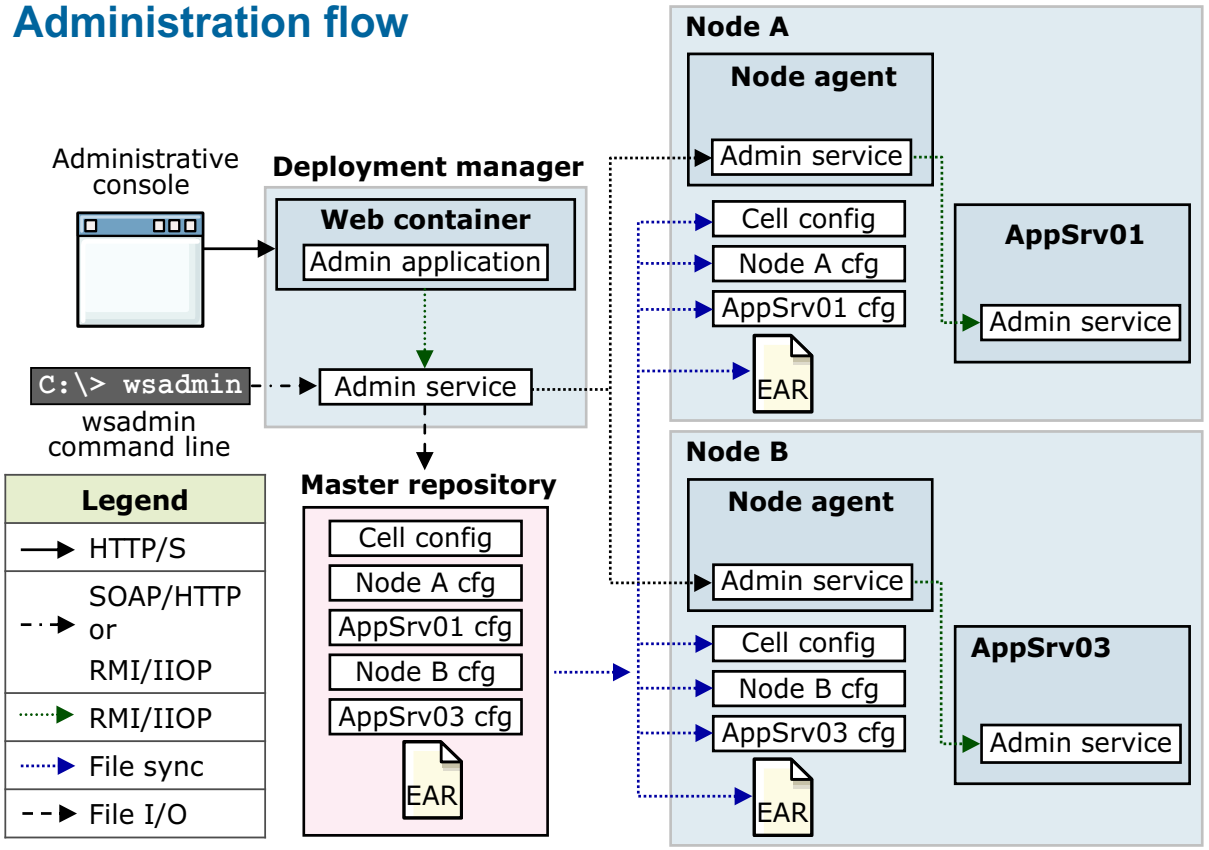


Figure 1-8. Administration flow

The administrative console and wsadmin are the two ways that the environment is administered. However, these tools communicate with the deployment manager and *not* with the application servers directly. The communication flow of these commands is from the tools to the deployment manager, to the node agents, to the application servers. This flow allows for the administration of multiple nodes from a single focal point (the deployment manager). Each node can possibly contain multiple application servers.

There is *one* main (master) repository for the configuration files within a cell, and those files are associated with the deployment manager. All updates to the configuration files go through the deployment manager. Be careful about directly connecting to an application server with wsadmin or the administrative console. Any changes to the configuration files are only temporary and are overwritten with the configuration files from the master files (repository).

IBM Training

File synchronization

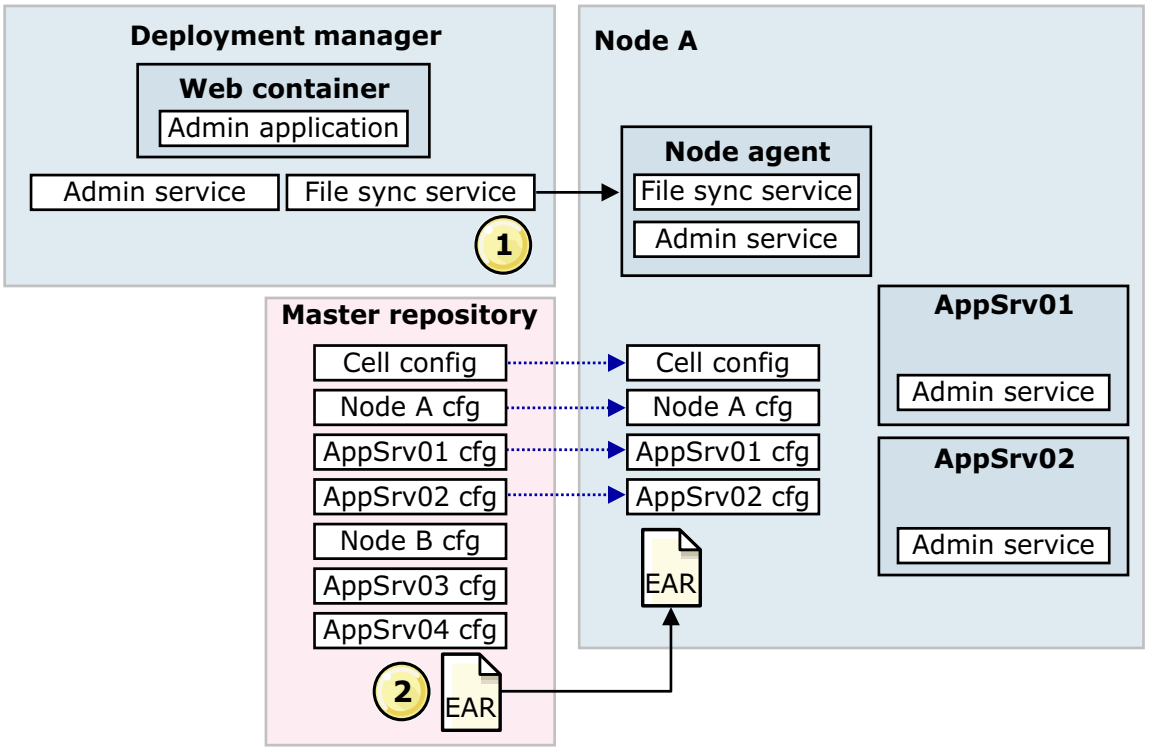


Figure 1-9. File synchronization

Each managed process, node agent, and deployment manager starts with its own set of configuration files. The deployment manager contains the master configuration. Any changes at the node agent or server level are local, and the master configuration overrides them at the next synchronization (update).

1. Node agents synchronize their files with the master copy either automatically or manually. Automatic synchronization can be done at startup or scheduled periodically. Manual synchronization is done with the administrative console or from the command line.
2. During synchronization, the node agent asks for changes to the master configuration. Any new or updated files are copied to the node.

IBM Training

WebSphere Network Deployment profiles

Benefits of profiles in network deployment

- Think of profiles as representing a node
- Can install multiple profiles on a single machine

All profiles use the same product files

- **Application server** profile (stand-alone)
  - Equivalent to Base or Express application server
  - Has a node name and a cell name property, and corresponding directories
  - Cell directory is overwritten upon federation
- **Deployment manager** profile
  - Creates a deployment manager
- **Custom** profile (managed)
  - Creates a managed node, which by default, is federated into a cell
  - Creates a node agent, but no application servers
- **Cell** profile
  - Creates both a deployment manager and a federated node
- Others

Figure 1-10. WebSphere Network Deployment profiles

The addition of Network Deployment to this discussion does not change the definition of a profile. The WebSphere configuration is still built by creating profiles, which consist of product binary files and configuration files. The profile that is listed as a “stand-alone node” is listed here as an “application server”. The deployment manager profile is added, which is a special type of node that manages the administrative domain of a cell.

1.2. Managing web servers

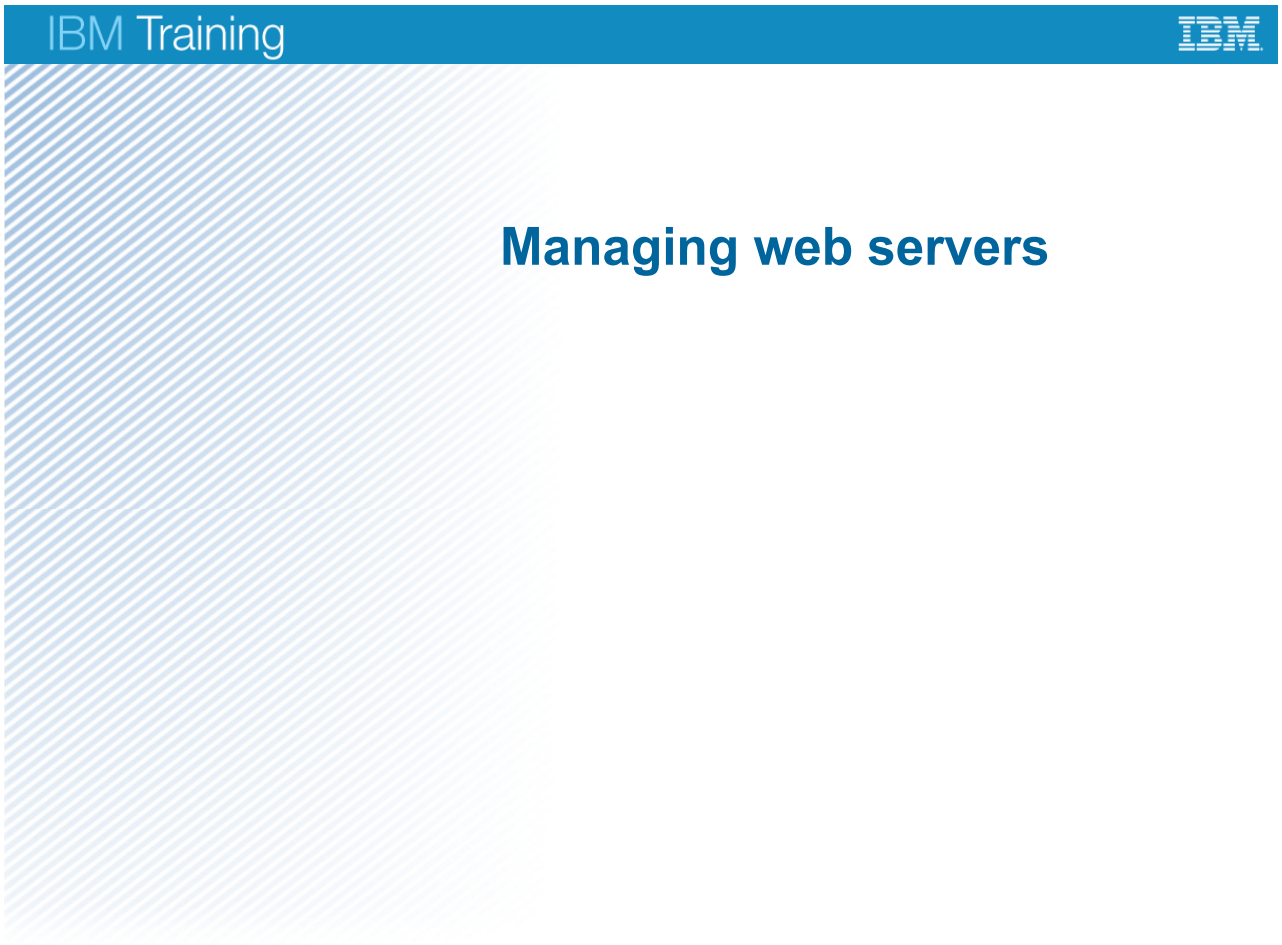


Figure 1-11. Managing web servers

IBM Training

Web servers within a cell

Web servers are customized

- Each web server plug-in is customizable
- Requires a web server definition
  - Defining a web server does not mean that it is managed
- Plug-in properties are defined on a per web server basis
- Each plug-in has a unique `plugin-cfg.xml` generated for it
- A cell level `plugin-cfg.xml` can also be generated

Web servers can optionally be managed

- Web servers can be unmanaged
  - No management is available
- You can manage web servers by:
  - A node agent
  - The IBM HTTP Server administrative process

Figure 1-12. Web servers within a cell

Web servers within a cell are customized. Each web server plug-in is customizable and requires a web server definition. Defining a web server does not mean that it is managed; plug-in properties are defined on a web-server-by-web-server basis. Each plug-in has a unique `plugin-cfg.xml` file that is generated for it; a cell level `plugin-cfg.xml` can also be generated.

Web servers can be managed or they can be unmanaged. For a managed web server, a node agent or the IBM HTTP Server administrative process can manage it.

IBM Training

Managed versus unmanaged nodes

- A managed node is a node that contains a node agent
- An unmanaged node is a node in the cell without a node agent
  - The rest of the environment can be aware of the node
  - Useful for defining HTTP servers as part of the topology
  - Allows creation of different plug-in configurations for different HTTP servers

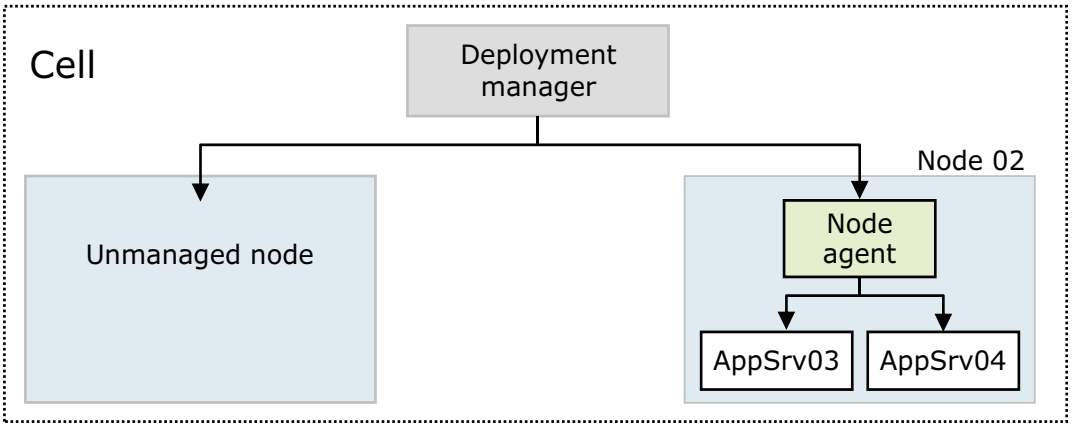


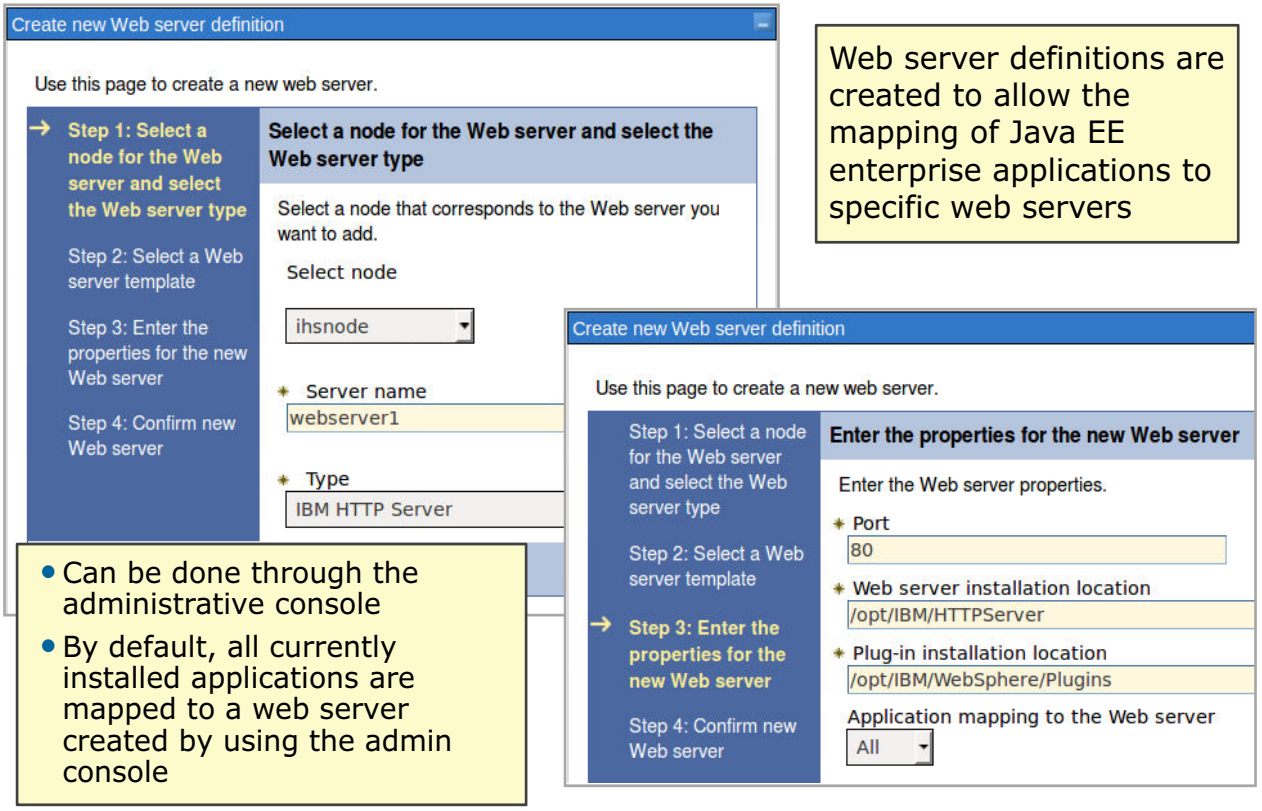
Figure 1-13. Managed versus unmanaged nodes

A node agent is a process that handles communications with the resources within the node.

An unmanaged node has no node agent. It is a reference to a machine somewhere in the topology. Typically an unmanaged node is used to define the location of web servers.

IBM Training

Web server definitions (1 of 2)



Web server definitions are created to allow the mapping of Java EE enterprise applications to specific web servers

- Can be done through the administrative console
- By default, all currently installed applications are mapped to a web server created by using the admin console

Figure 1-14. Web server definitions (1 of 2)

Just as modules for an enterprise application must be mapped to one or more application servers, they also must be mapped to one or more web servers. When you define a web server, the default is that all currently installed applications are mapped to the web server.

Web server definitions (2 of 2)

- Alternatively use the script that is generated during the configuration of the plug-in, which can automate the mapping of all the applications to the web server

  `configure<`*`Web_server_name`*`>.sh` in `<`*`plugin_root`*`>/bin`

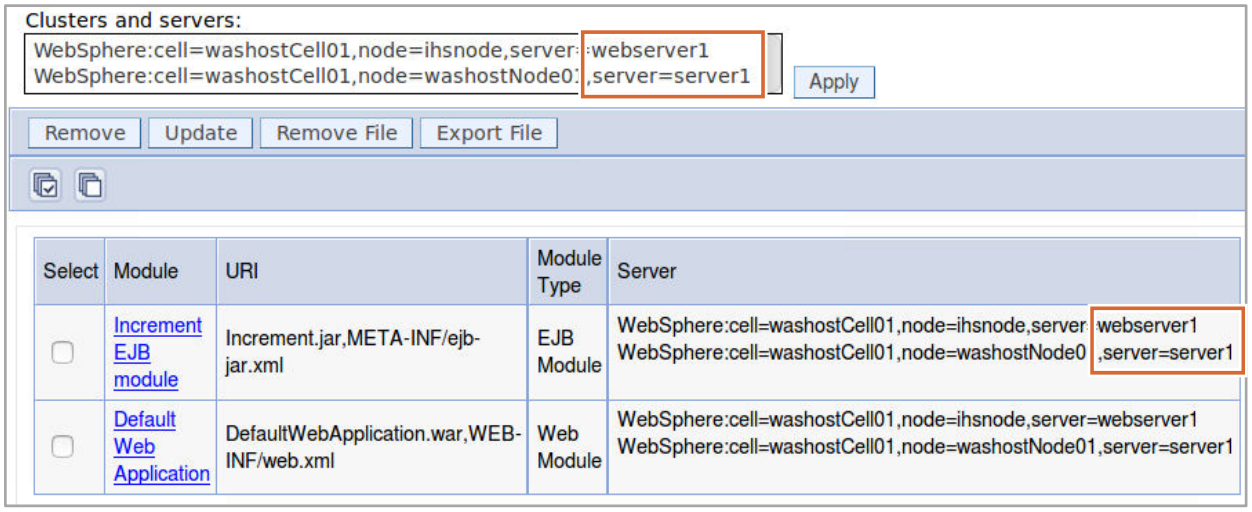
Figure 1-15. Web server definitions (2 of 2)

When the web server plug-in is configured, a script is generated that you can use to define the web server. The script uses the same defaults as defining the web server in the administrative console.

IBM Training

Web server custom plugin-cfg.xml

- Mapping the applications to specific web servers causes the `plugin-cfg.xml` files for only those web servers to include the information for those applications
  - Target-specific web server applications that run in a cell
  - Deployment manager automatically generates them



Application module is explicitly mapped to a web server

Figure 1-16. Web server custom plugin-cfg.xml

After the web servers are defined, any applications that are installed after the web servers are defined must be explicitly mapped to a web server.

IBM Training

Managing plugin-cfg.xml files

`plugin-cfg.xml` files are automatically generated and propagated

- This behavior is the default, but can be changed
- This behavior is configurable through the console

`plugin-cfg.xml` files can be generic to a cell or custom to web server

- Generating a cell generic `plugin-cfg.xml` file
  - Use the command-line script **`<was_root>/bin/GenPluginCfg.sh`**
  - Not available through the console
- Generating a web server custom `plugin-cfg.xml` file
  - Use the administrative console
  - Must map applications to web servers
  - Can customize the plug-in settings of each web server

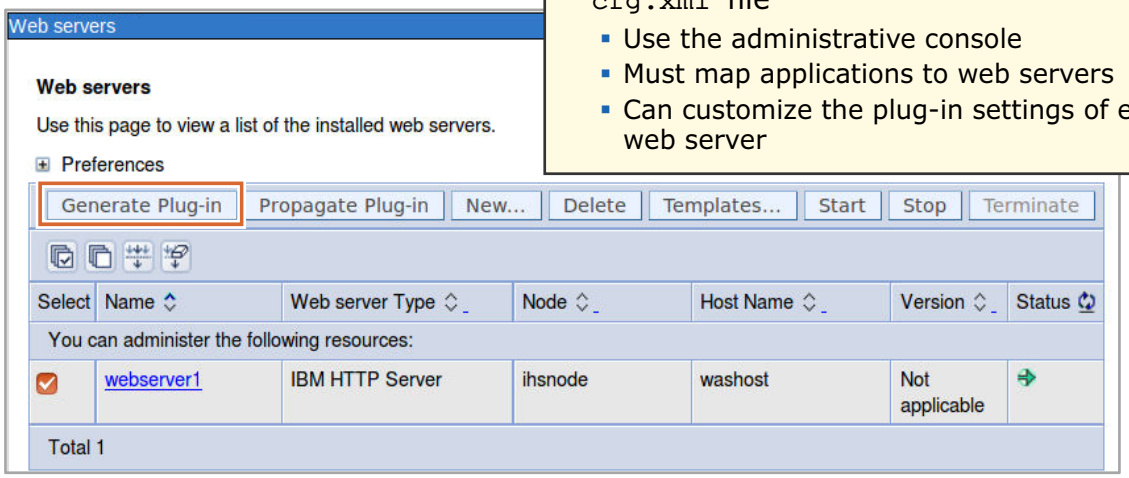


Figure 1-17. Managing plugin-cfg.xml files

The `plugin-cfg.xml` files are automatically generated and propagated. This behavior is the default, but it can be changed; it is configurable through the console.

The `plugin-cfg.xml` files can be generic to a cell or custom to a web server. To generate a cell generic `plugin-cfg.xml` file, use the command-line script: *<was_root>*`/bin/GenPluginCfg.sh`

This action is not available through the console.

To generate a web server custom `plugin-cfg.xml` file, you use the administrative console. You map applications to web servers, and you can customize the plug-in settings for each web server.

IBM Training

Managing web server plug-in properties

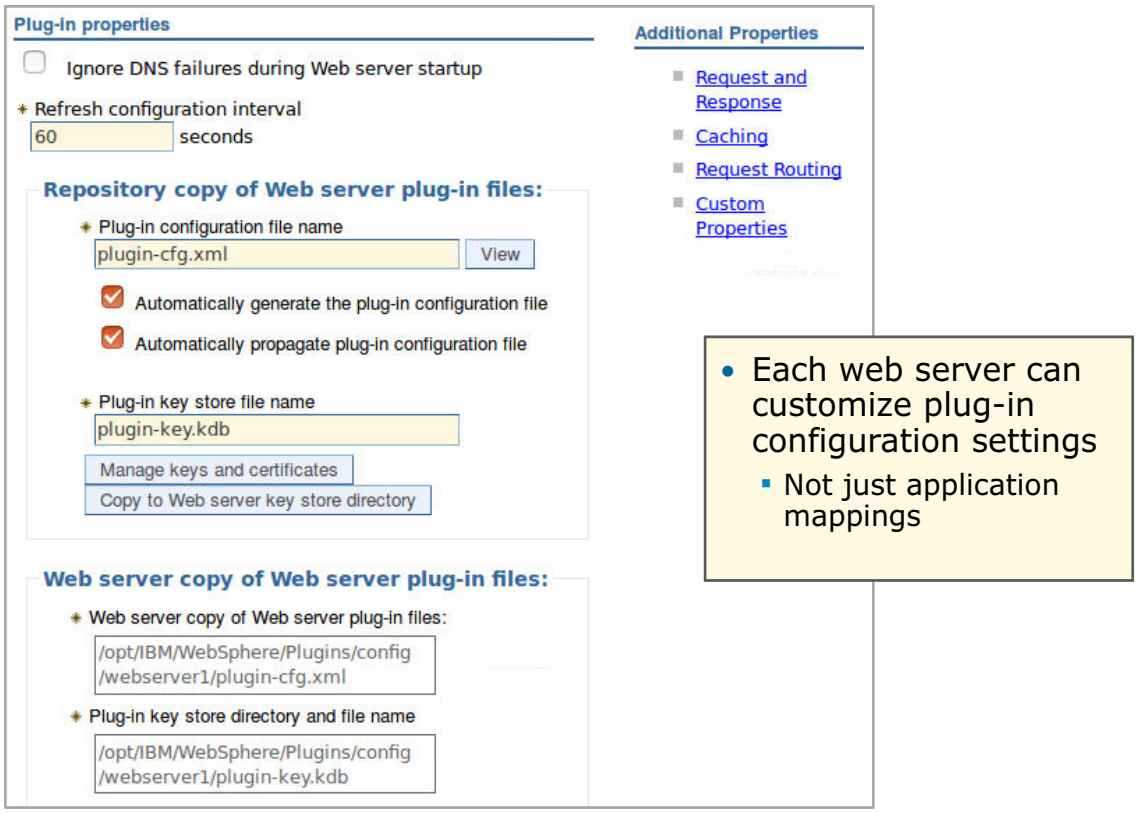


Figure 1-18. Managing web server plug-in properties

The links on the right side of the page under Additional Properties provide access to pages for changing the plug-in properties.

IBM Training

Virtual hosts

- Configuration that allows a host machine to resemble multiple host machines
- Each virtual host has a logical name and one or more host aliases
- Several default virtual hosts exist, including:
  - default_host: Used for accessing the default applications
    Example: `http://localhost:9080/snoop`
  - admin_host: Used for accessing the administrative console
    Example: `http://localhost:9060/ibm/console`

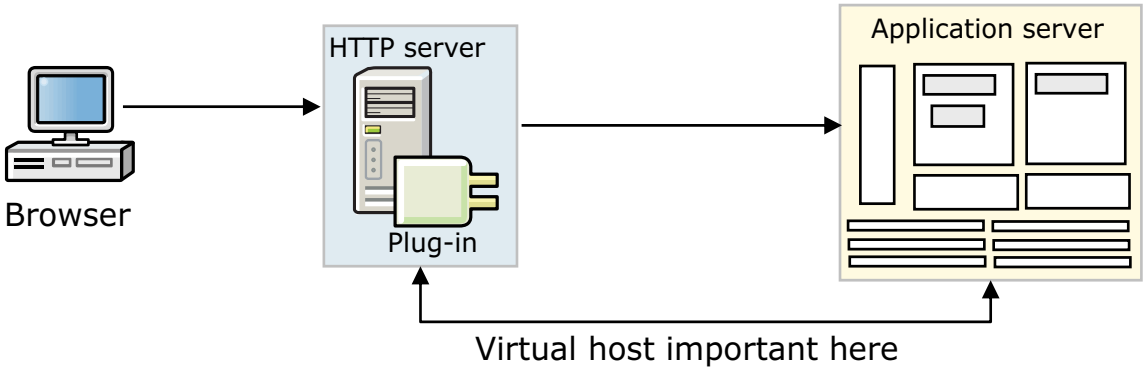


Figure 1-19. Virtual hosts

The virtual hosts definition for the application server determines *not* which ports are listened to, but what ports are acceptable for the incoming URL. Since most requests come in from the external HTTPd, the ports that are specified either explicitly or implicitly (because of the protocol used) are 80 and 443. If an administrator wants to allow other ports to be used, the ports must be specified on the virtual host to which the application is mapped. If direct access is required to an application (without going through an external web server), adding the port that the application server accepts for incoming requests to the virtual host definition makes that possible. More specifically, if an application server listens to port 9084 and direct access was required (`http://<hostname>:9084/<uri>`), then adding 9084 to the virtual host definition makes that possible.

A virtual host is a configuration that allows a host machine to resemble multiple host machines.

- Allows one machine to support multiple applications
- Associated with the cell, not a single node
- Allows plug-in to route requests to the correct servers

Each virtual host has a logical name and one or more host aliases.

- Each alias is a host name and port combination (allows wildcards)

- Example: `*:80`, `*:443`, `*:9080`, `*:9060`

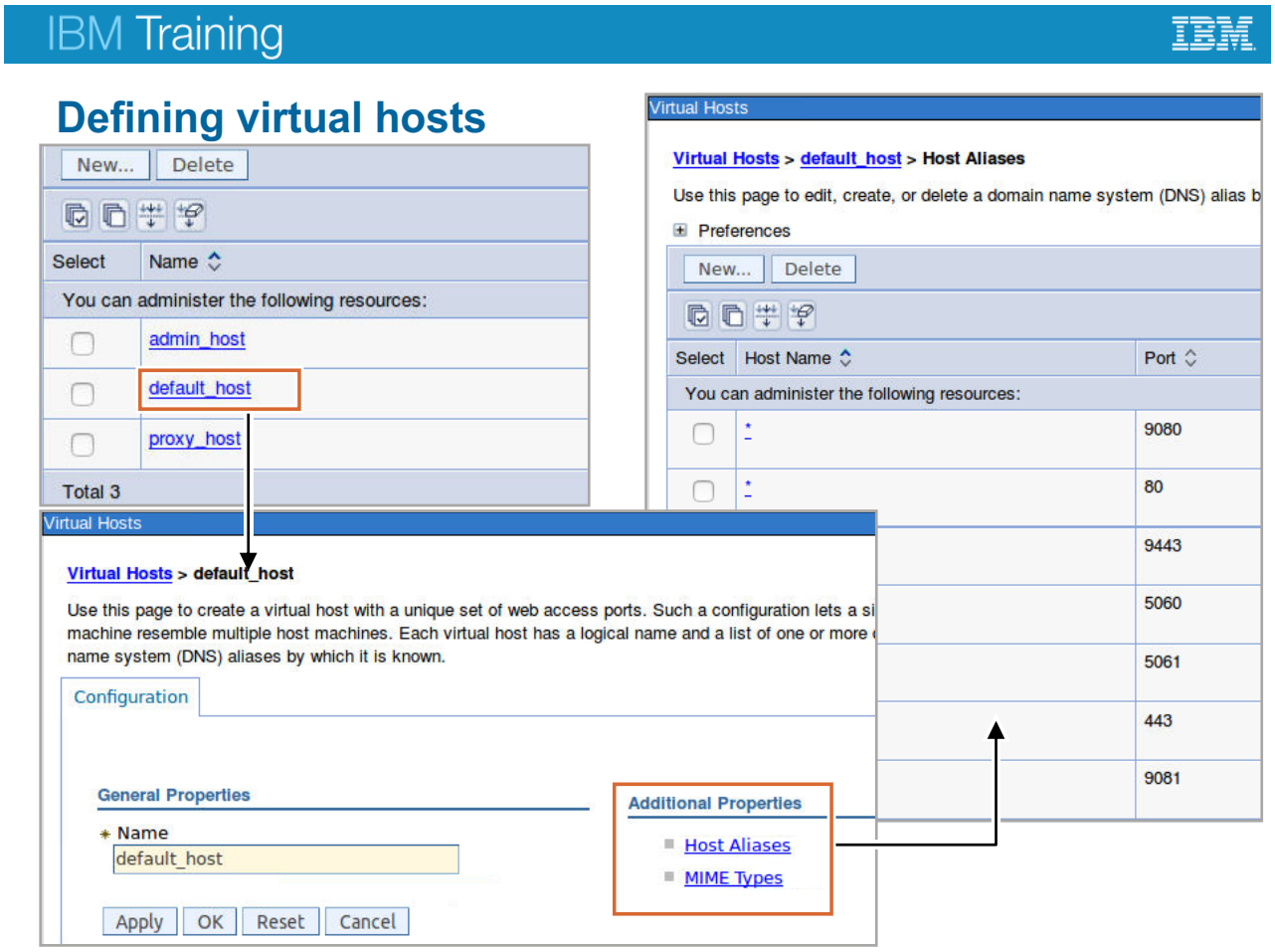


Figure 1-20. Defining virtual hosts

This slide shows the definition for “default_host.” Notice that the host aliases are specified with the “*” symbol. Using the “*” symbol means that a request for any host name on ports 9080, 80, and 9443 is forwarded to this host. When applications are installed, web modules within those applications must be mapped to a virtual host.

IBM Training

Managing web servers with WebSphere

Deployment manager can manage external web servers

- IBM HTTP Server (special case, no node agent needed)
  - Deployment manager can distribute `plugin-cfg.xml` files to web server machines
  - Can be started and stopped
  - Can edit the `httpd.conf`
- Other web servers (node agent needed)
  - Can have `plugin-cfg.xml` files that are automatically distributed to them
  - Can be started and stopped

Web servers are defined within WebSphere cell topologies

- Managed node (local) or unmanaged node (remote)
  - Managed nodes use a node agent to control the web server
  - Unmanaged nodes use the IBM HTTP Server Admin Service instead of a node agent to control the web server

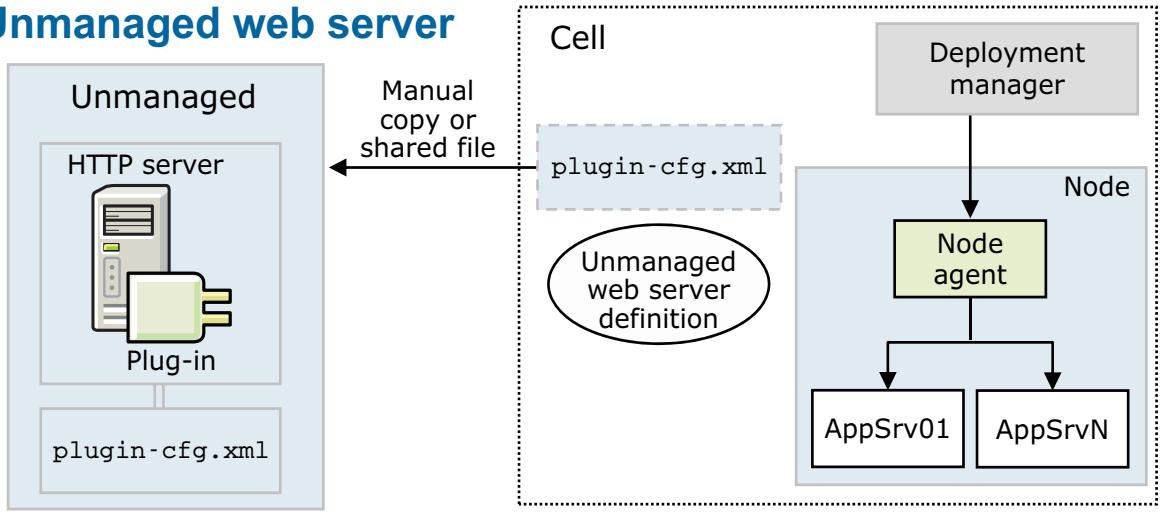
Figure 1-21. Managing web servers with WebSphere

There are three types of WebSphere Application Server nodes upon which you can create and manage a web server. Over the next several pages, three common web server management scenarios are presented:

- Using a web server as an unmanaged node
- Using IBM HTTP Server as an unmanaged node
- Using a web server as a managed node

IBM Training

Unmanaged web server



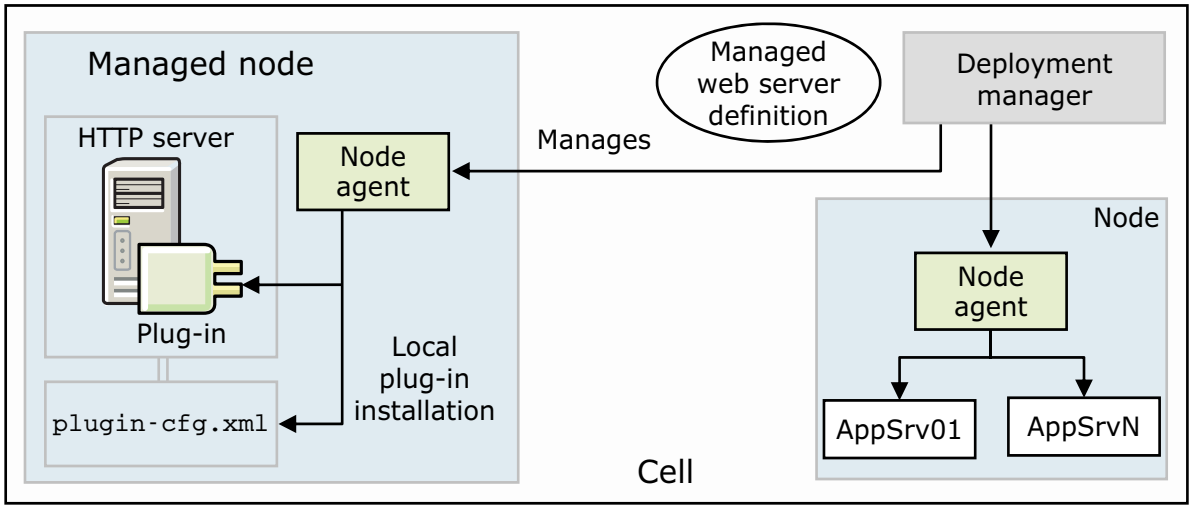
- WebSphere node agent does not manage web server (other than IBM HTTP Server)
  - Allows WebSphere system administrator to create custom plug-in files for a specific web server
  - Application mappings
  - SSL certificates
- Manually copy or use FTP to transfer the plug-in configuration file from the deployment manager machine to the web server machine

Figure 1-22. Unmanaged web server

The web server is registered as an unmanaged node in this WebSphere configuration. This scenario is common for web servers that are installed outside the firewall or in a DMZ where no WebSphere Application Server exists. The implication with this scenario is that all management of the web server occurs manually, which is outside the control of WebSphere Application Server.

IBM Training

Managed web server on a managed node (local)



- Install a web server on a managed node
- Create a web server definition within the dmgr
- Node agent receives commands from dmgr to administer the web server
- `plugin-cfg.xml` file is propagated through the file synchronization service and is in the `config` directory
- Warning: Security issues if this configuration spans a DMZ

Figure 1-23. Managed web server on a managed node (local)

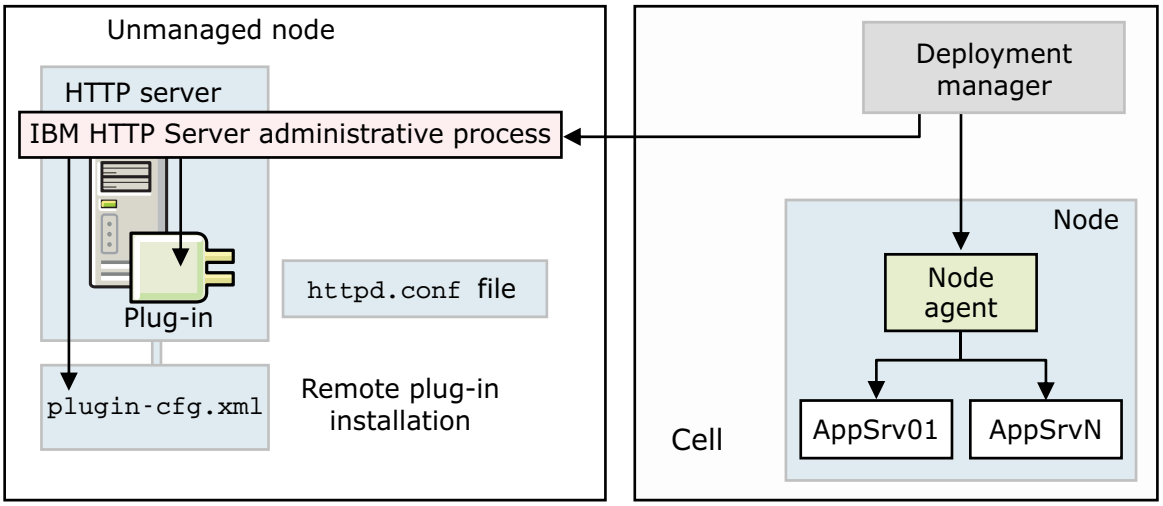
The deployment manager manages the web server through the node agent. In this way, you can start or stop the web server and automatically push the plug-in configuration file to the web server from the deployment manager. This configuration can be used when the web server is installed on the same machine as the WebSphere Application Server is installed. It is a common scenario for behind a firewall where a WebSphere node can be installed.

A node agent communicates with the web server from the administrative tools of WebSphere.

It might be undesirable to use this configuration, since access to the node agent in a DMZ can compromise security.

IBM Training

IBM HTTP Server as unmanaged node (remote)



- The IBM HTTP Server administrative process provides administrative functions for IBM HTTP Server within WebSphere
  - Able to start, stop IBM HTTP Server, make configuration changes to `httpd.conf`, and automatically push the plug-in configuration file to IBM HTTP Server machine
  - Node agent is not needed on the web server machine

Figure 1-24. IBM HTTP Server as unmanaged node (remote)

IBM HTTP Server can be managed completely from the deployment manager (dmgr). The dmgr communicates with the IBM HTTP Server administrative process that runs on the node with IBM HTTP Server. There are two Apache instances (processes) on the IBM HTTP Server machine: one running the administrative services and one containing the plug-in.

UNIX

On UNIX operating systems, the user ID under which the IBM HTTP Server administrative process runs must have write permissions to the `plugin-cfg.xml` file. By default, when IBM HTTP Server is installed, root owns the `plugin-config.xml` file, and only root has write access to it. The result is a failure with propagation of the `plugin-cfg.xml` file when it is run from the administration tools.

IBM HTTP Server administration overview

- Direct administration of IBM HTTP Server by manually editing `httpd.conf`
- IBM HTTP Server has no web-based console, as previous versions did

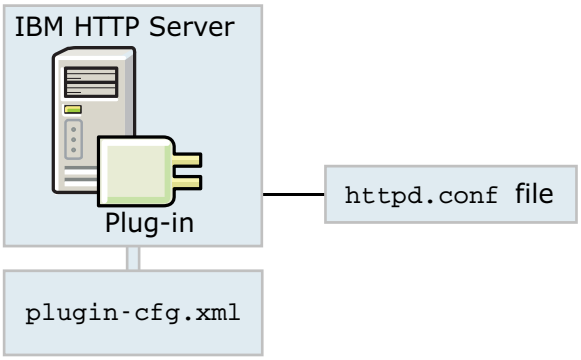


Figure 1-25. IBM HTTP Server administration overview

To administer IBM HTTP Server, you must either manually make updates to the `httpd.conf` file, or use the administrative console.

IBM Training

IBM HTTP Server administration server

- IBM HTTP Server administration server runs as a separate instance of IBM HTTP Server
- Administrative component for IBM HTTP Server includes:
  - IBM HTTP Server administration configuration file (`admin.conf`)
  - Default port for the IBM HTTP Server administration server is 8008
- IBM HTTP Server administration authentication password file (`admin.passwd`)
  - Initially blank, which prohibits access to IBM HTTP Server administration
  - Administrator updates IBM HTTP Server admin password file by using:
    `> htpasswd -cm ..\conf\admin.passwd <user_name>`
- To start and stop the administration server:
  - `<ihs_root>/bin/adminctl start`
  - `<ihs_root>/bin/adminctl stop`
  - Or Windows service

Figure 1-26. IBM HTTP Server administration server

The IBM HTTP Server administration server acts like a lightweight node agent on the IHS machine. The administration server allows the WebSphere Application Server to control the IHS environment server and propagate plugin files.

1.3. Security domains

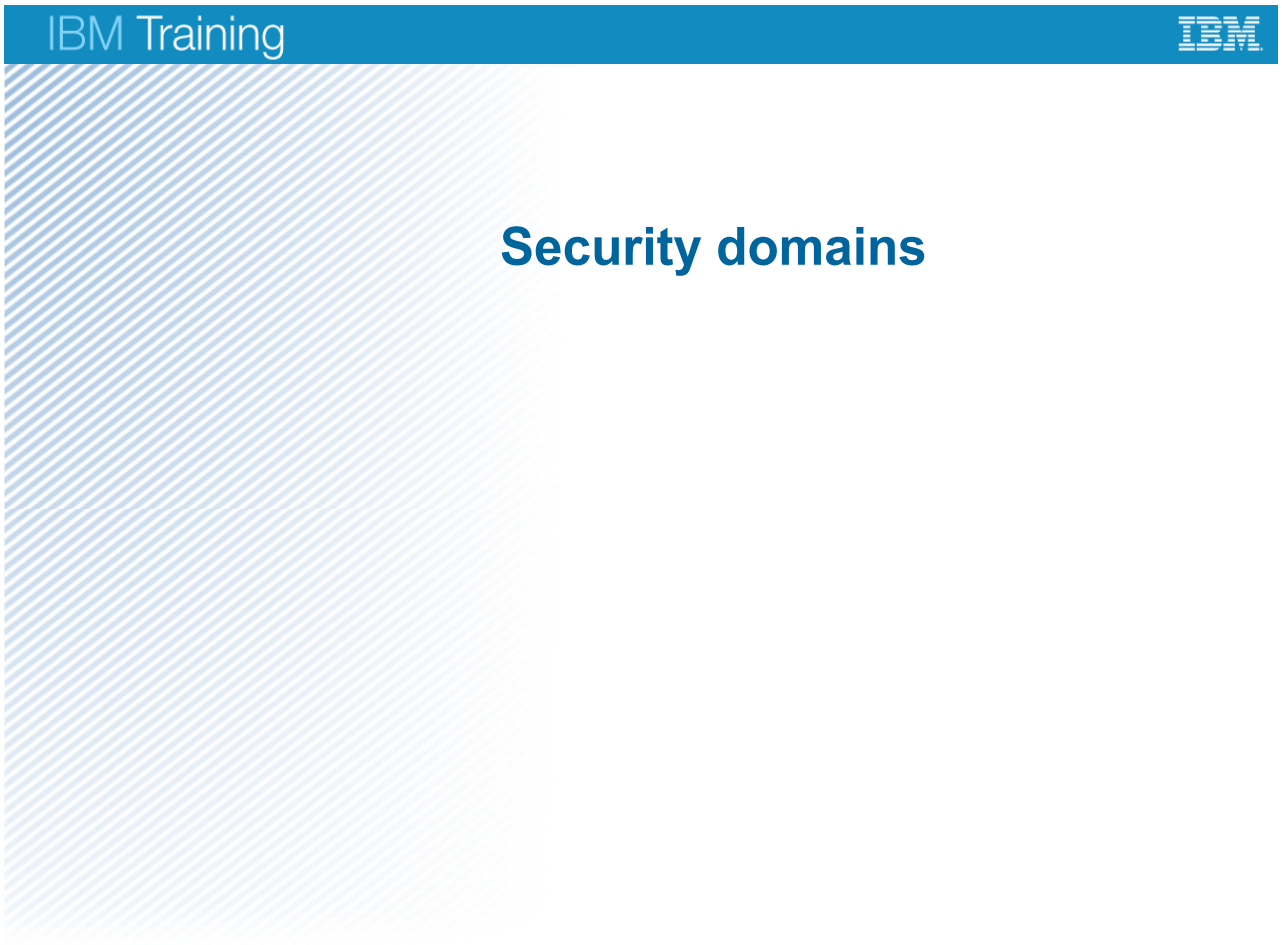


Figure 1-27. Security domains

Security domains

- Multiple security domains are supported
  - Can create different security configurations and assign them to different applications
  - Can configure different security attributes for both administrative and user applications within a cell environment
  - Can configure different applications to use different security configurations by assigning the servers, clusters, or service integration buses to the security domains
- Only users that are assigned to the administrator role can configure multiple security domains
- For example, with security domains, it is possible to have different user registries that are configured for distinct parts of the cell

Figure 1-28. Security domains

This screen shows a security configuration that is defined at a cell level.

Security configurations

- Traditionally, the security configuration was defined at a cell level
  - A side effect was all elements of the cell shared the exact same security configuration

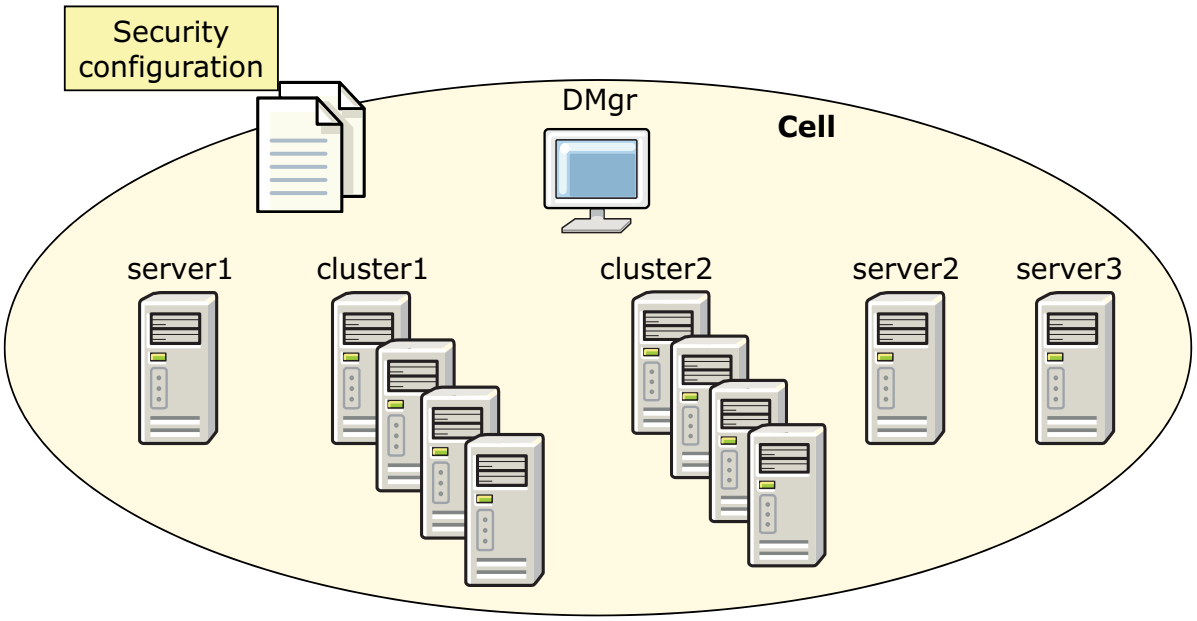


Figure 1-29. Security configurations

This screen shows not only a security configuration that is defined at a cell level, but more configurations for specific scopes. This configuration means that it is possible to define security behavior specific to certain scopes, which is different from what is configured for the cell level.

Security domains

- With security domains, it is possible to have a cell level security configuration, and multiple other security configurations at different scopes

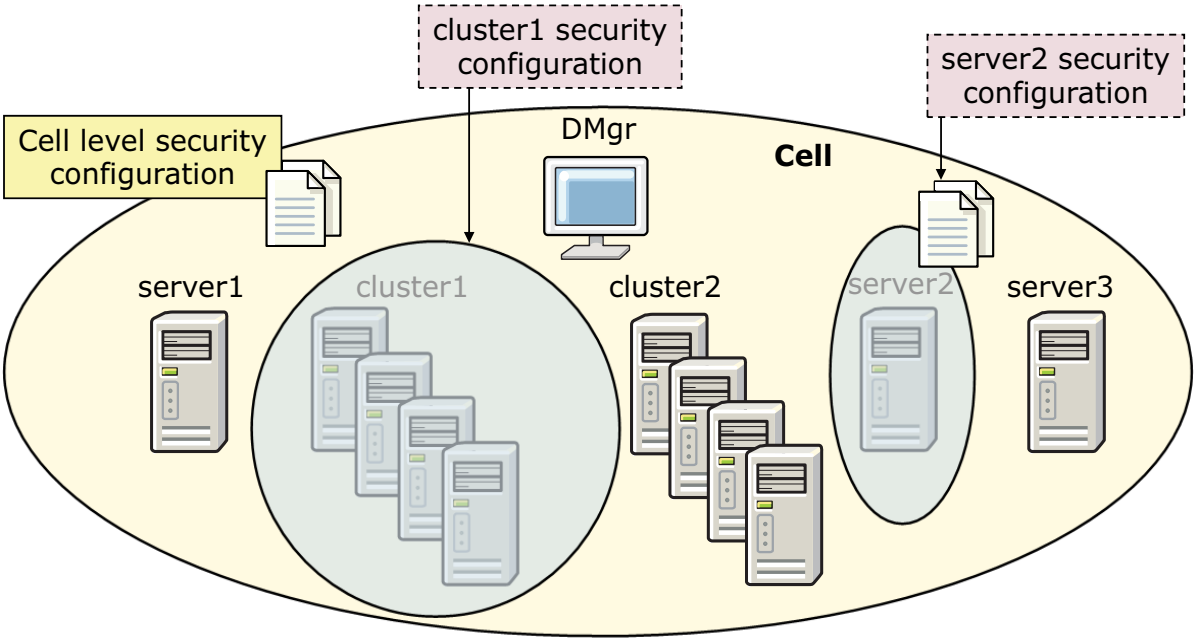


Figure 1-30. Security domains

These diagrams show the use of the administrative console to create a security domain.

IBM Training

Creating a security domain

- Use the console to create a security domain
  - **Security > Security domains > New**

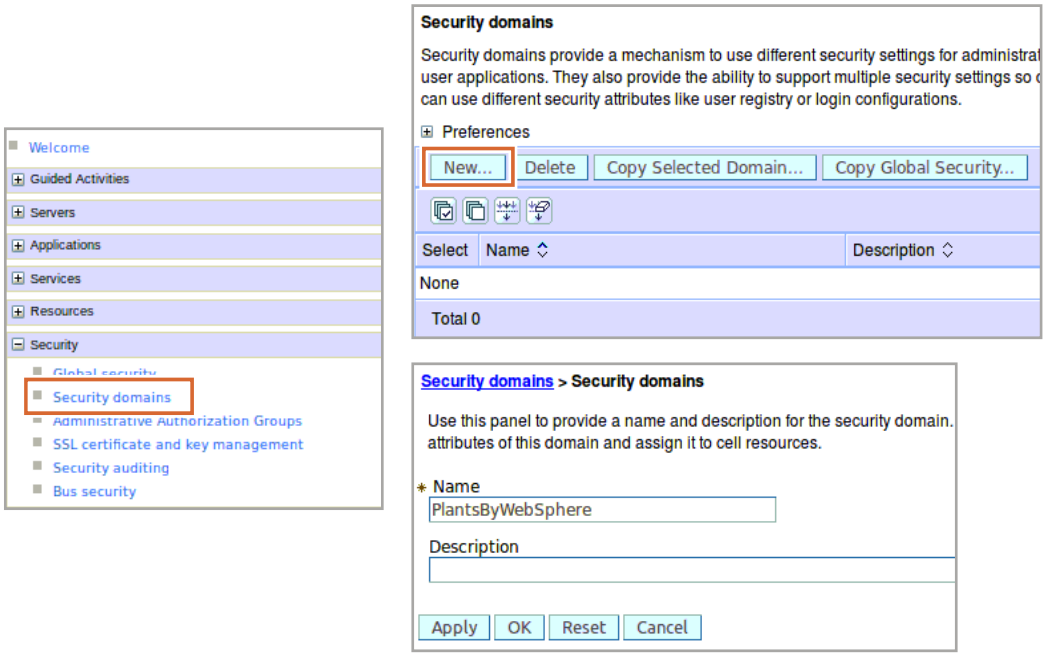


Figure 1-31. Creating a security domain

These diagrams show some of the different levels for which security domains can be used. Examples can include having different user registries for different security domains. It is also possible to change settings for application security, Java 2, user registries (User Realm), and others.

IBM Training

Configuring a security domain

- Define a scope and configure the attributes
  - It is possible to enable application security for only the PlantsByWebSphere

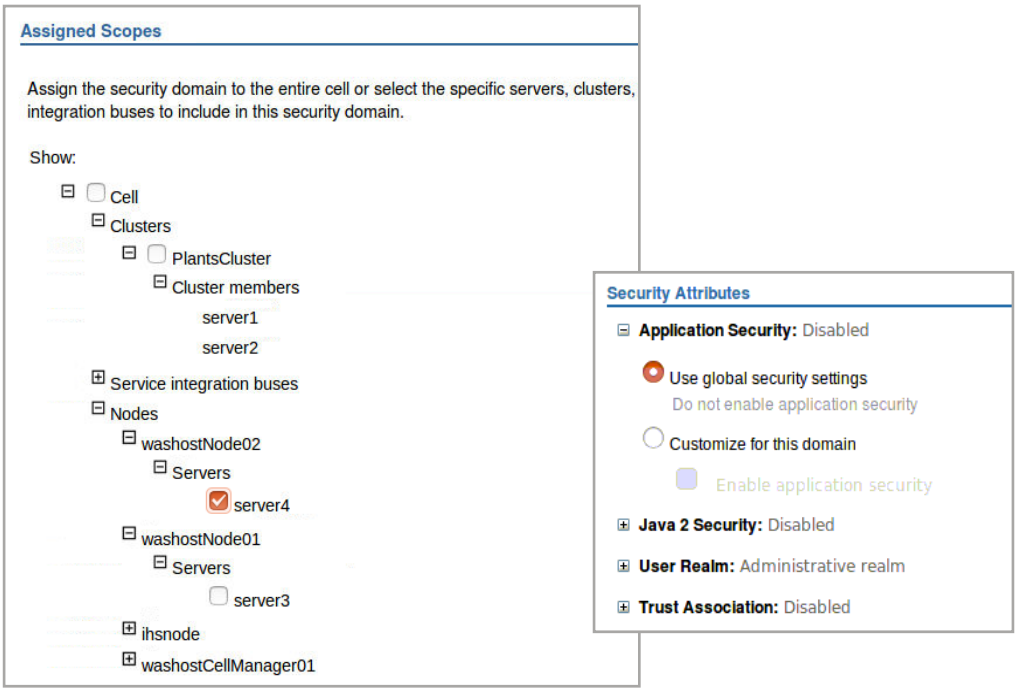


Figure 1-32. Configuring a security domain

These diagrams show the possible scopes and security attributes for the security domains.

1.4. Fine grained access

IBM Training

Fine grained access

Figure 1-33. Fine grained access

IBM Training

Traditional role-based administrative access

- Traditionally, administrators were mapped to roles which controlled what the administrative user might do at a cell level

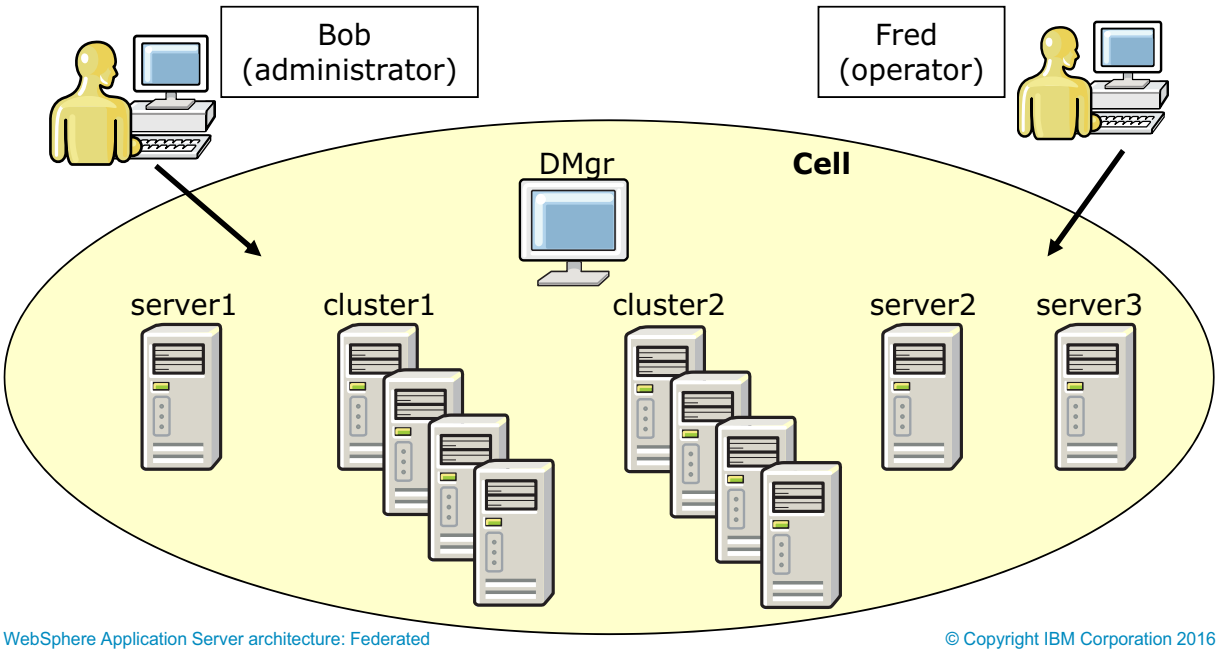


Figure 1-34. Traditional role based administrative access

Normally, and administrative user has their permissions defined at a cell level. If an administrator has operator role assigned to them, they have operator role for the entire cell.

IBM Training

Fine grained administrative access

- Fine grained administrative access allows you to define administrative access to specific parts of your cell

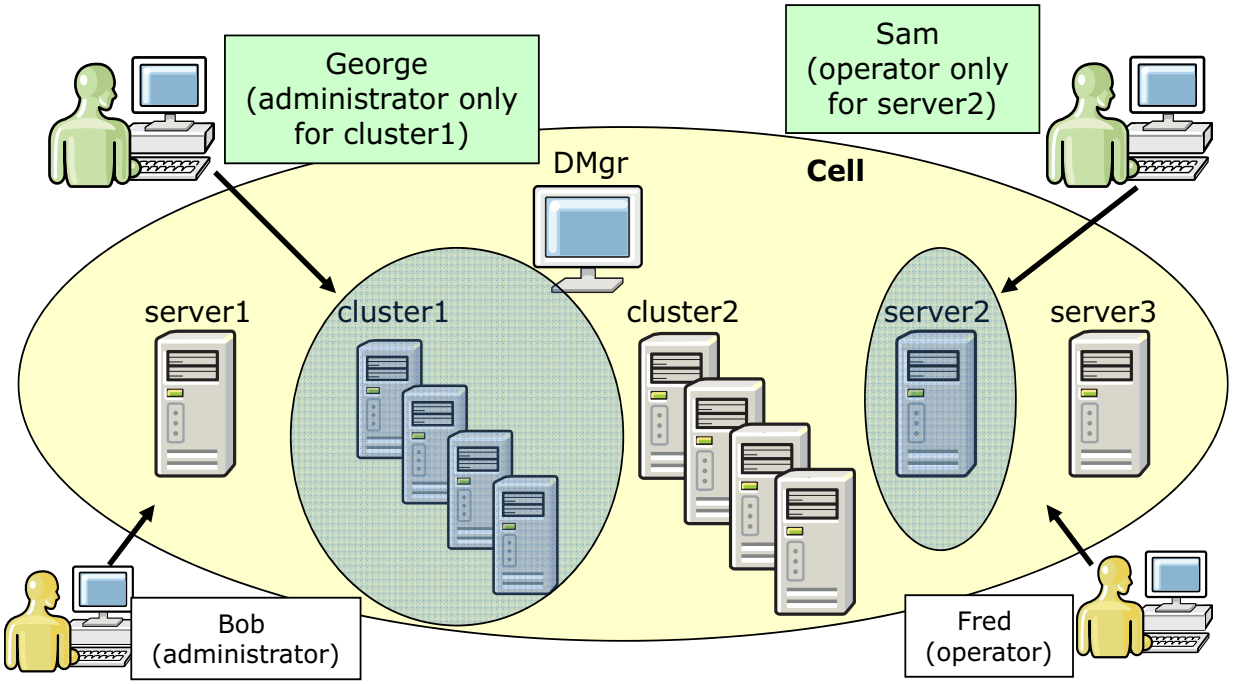


Figure 1-35. Fine grained administrative access

With fine grained access, it is possible to define an administrator as an operator for only a subset of the cell rather than the entire cell. In this example, the administrators Bob and Fred have roles defined at the cell level. Administrators George and Sam have roles defined on only a portion of a cell.

IBM Training

Administrative Isolation

- WebSphere has always supported multiple administrators with different permissions at the cell level
  - Administrative roles, by default, are defined at a cell level
- With fine grained access, administrative roles can be defined at a scope less than the cell
  - Example: George has administrative rights on cluster1, but does not have the ability to affect any other parts of the cell.
  - Example: Sam has operator rights for server2, but cannot do anything anywhere else in the cell.
  - Scopes include: Node, node group, server, cluster.

Figure 1-36. Administrative Isolation

Administrative roles can be defined at the cell level. With fine grained access, administrative users can also have administrative roles assigned at scopes other than the cell. Scopes includes nodes, node groups, server, and clusters.

1.5. Additional concepts

IBM Training

Additional concepts

Figure 1-37. Additional concepts

IBM Training

Flexible management

- Loose management coupling
- Coordinates management across a group of endpoints
  - One job to install application across a number of nodes
- Can manage through administrative agent or deployment manager

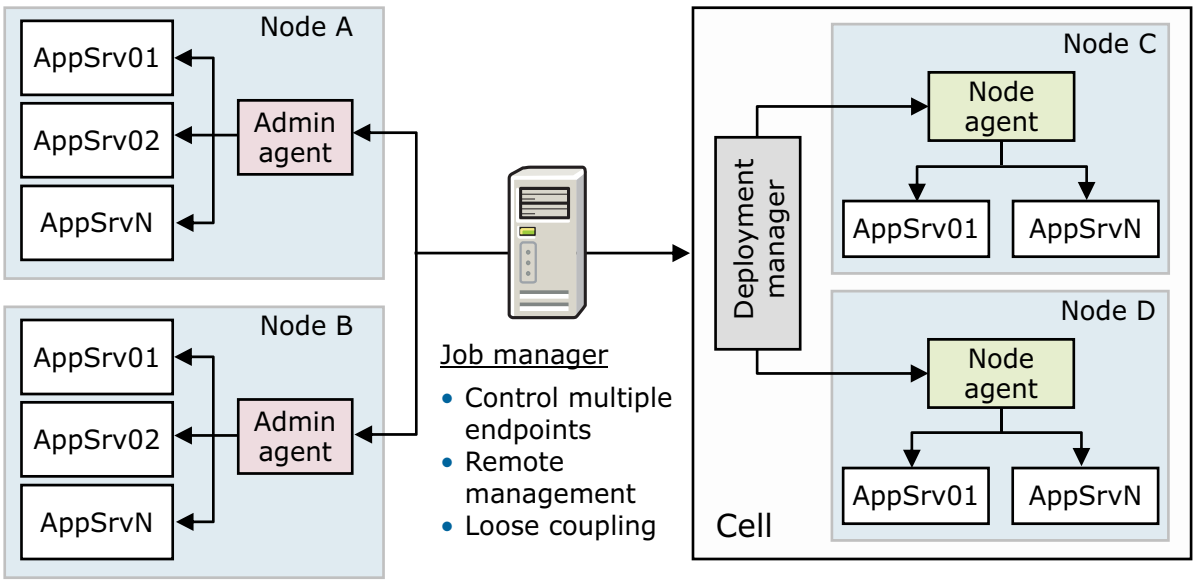


Figure 1-38. Flexible management

Flexible management is an approach that allows an administrator to manage multiple application servers or cells through a loose asynchronous interface. Flexible management is covered later in this course.

IBM Training

Centralized installation manager (CIM)

- Simplifies the installation and maintenance of application servers within a Network Deployment cell
- Install, update, uninstall version 9 and all Installation Manager installable products

Figure 1-39. Centralized installation manager (CIM)

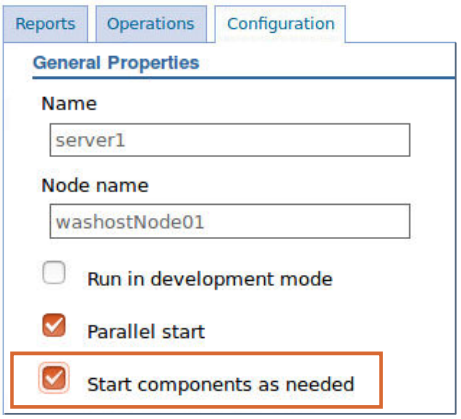
The CIM pushes the product binary files or maintenance to the remote targets and starts the standard installer or update installer tool to complete the installation or update on the targets, allowing you to:

- Download interim fixes and fix packs from IBM support directly to the CIM repository
- Install interim fixes and fix packs on target nodes with the Network Deployment cell
- Monitor download and installation status of packages through the administrative console

IBM Training

Intelligent runtime provisioning

- Dynamic start of server components that are based on application needs
- Reduces runtime footprint; less memory required
- Can significantly reduce startup times
- Disabled by default



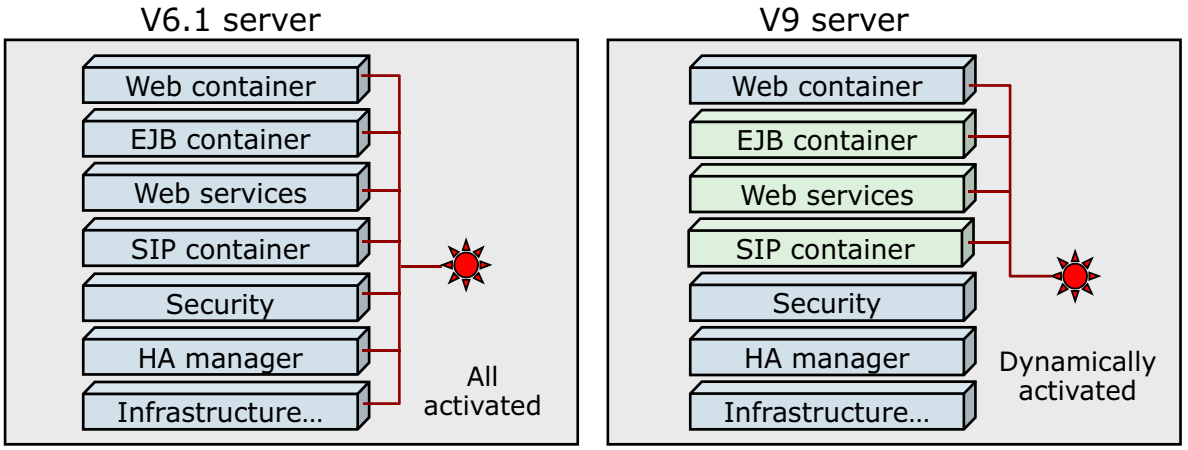


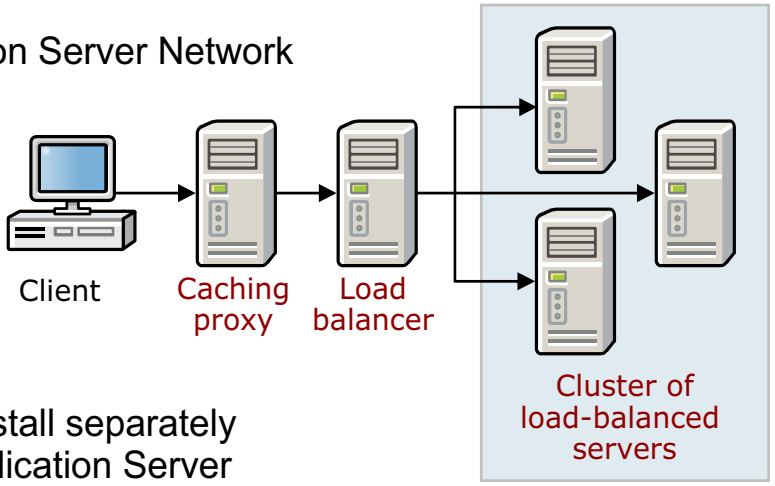
Figure 1-40. Intelligent runtime provisioning

Intelligent runtime provisioning allows the application server to start faster and with less memory because it loads only those components that are required. As other services are needed, they are loaded on demand.

IBM Training

Edge Components

- WebSphere Application Server Network Deployment package contains the following Edge Components functions:
  - Load balancer
  - Caching proxy



- Edge Components install separately from WebSphere Application Server
- Load balancer is responsible for balancing the load across multiple servers that can be within either local area networks or wide area networks
- Purpose of caching proxy is to reduce network congestion within an enterprise by offloading security and content delivery from web servers and application servers

Figure 1-41. Edge Components

Edge Components are included the Network Deployment package. The Edge Components include a load balancer and a caching proxy. The load balancer distributes incoming client requests across servers, balancing workload and providing high availability by routing around unavailable servers. The caching proxy can satisfy subsequent requests for the same content by delivering it directly from the local cache, which is much quicker than retrieving it again from the content host. Cacheable content includes static web pages and JSP files with dynamically generated but infrequently changed fragments.

IBM Training

Intelligent Management

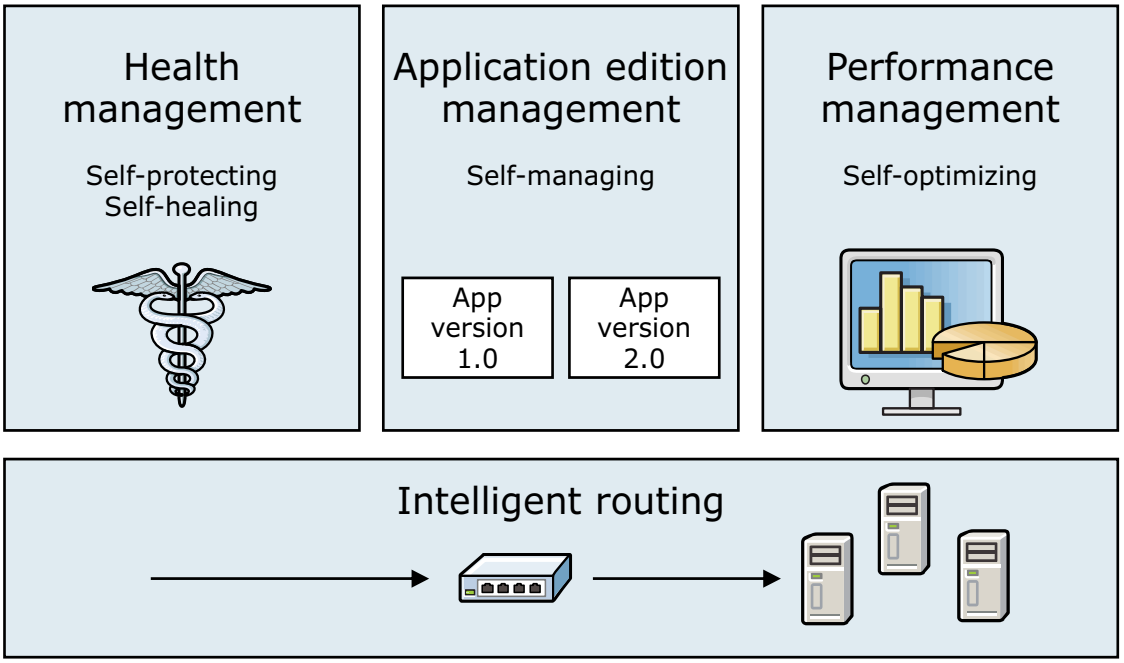


Figure 1-42. Intelligent Management

Intelligent Management provides a virtualized infrastructure that redefines the traditional concepts of Java Platform, Enterprise Edition (Java EE) resources and applications and their relationships with one another. This application infrastructure virtualization facilitates the ability of the product to automate operations in an optimal manner, increasing the quality of service. By introducing an automated operating environment with workload management, you can reduce total cost of ownership by using less hardware to do more work.

IBM Training

Unit summary

- Describe the Network Deployment runtime flow
- Describe Network Deployment concepts and terminology, such ascell,node,node agent, anddeployment manager
- Describe the Network Deployment administration flow
- Explain how to manage web servers from WebSphere Application Server

Figure 1-43. Unit summary

IBM Training

Review questions

1. A process that handles communications with the resources within the node is _____.
2. What is the process when the node agent checks for changes to the master configuration?
3. What is a configuration that allows a host machine to resemble multiple host machines?
4. What defines the runtime environment for either the deployment manager or the application server?

Figure 1-44. Review questions

Write your answers here:

1.

2.

3.

4.

IBM Training

Review answers

1. A process that handles communications with the resources within the node is the _____.
   The answer is Node agent.
2. What is the process when the node agent checks for changes to the master configuration?
   The answer is File synchronization.
3. What is a configuration that allows a host machine to resemble multiple host machines?
   The answer is Virtual host.
4. What defines the runtime environment for either the deployment manager or the application server?
   The answer is Profiles.

Figure 1-45. Checkpoint answers

Unit 2. Federating a cell

Estimated time

01:00

Overview

In this unit, you learn the process of federating a base profile into a cell and the administration of a multi-node distributed environment. You learn the process of creating a deployment manager and federating base profiles into a cell. You also learn how to add a node by using commands or the administrative console, and how to manage a web server with the administrative console.

How you will check your progress

- Review questions
- Lab exercises

References

WebSphere Application Server V9 Knowledge Center

https://www.ibm.com/support/knowledgecenter/SSEQTP_9.0.0/as_ditamaps/was900_welcome_base.html

IBM Training

Unit objectives

- Describe WebSphere Application Server cell concepts
- Describe and create the deployment manager profile
- Describe and create other profile types
- Describe custom profiles and automatic federation
- Describe the directories and configuration files for profiles
- Add a node by using commands or the administrative console
- Compare the deployment manager administrative console with the base administrative console
- Compare managed and unmanaged nodes
- Manage a web server by using the administrative console

Figure 2-1. Unit objectives

WebSphere cells

- A WebSphere cell defines an administrative domain
  - Available in WebSphere Application Server Network Deployment
  - A deployment manager provides centralized administration for entire cell
  - A cell is created as a profile
  - Nodes run application components in application servers

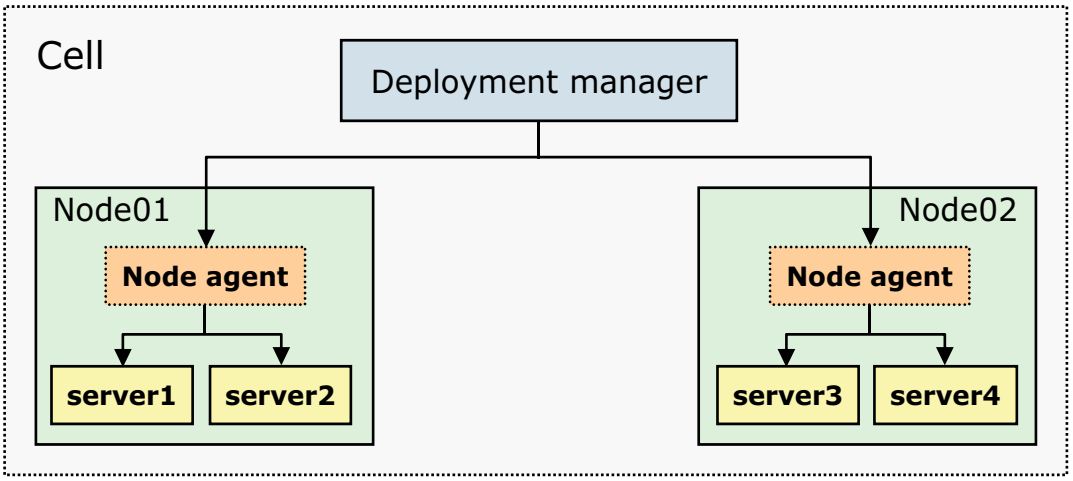


Figure 2-2. WebSphere cells

A WebSphere cell defines an administrative domain. A cell is a grouping of nodes into a single administrative domain. A cell can consist of multiple nodes, all administered from a deployment manager server. When a node becomes part of a cell (a federated node), a node agent server is created on the node to work with the deployment manager server to manage the WebSphere Application Server environment on that node. A cell includes the following characteristics:

- It is available in WebSphere Application Server Network Deployment.
- A deployment manager provides centralized administration for the entire cell.
- A cell is created as a profile.
- Nodes run application components in application servers.

The graphic includes the following cell environment topology:

- A cell encapsulating:
  - Deployment manager with connections to the node agents of two nodes
  - Node01:
    - Node agent with connections to the application servers of the node
    - server1

  - server2
- Node02:
  - Node agent
  - server3
  - server4

IBM Training

WebSphere Application Server process types

- Application server
  - Provides the functions that are required to support and host user applications
  - Runs on only one node, but one node can support many application servers
- Node agent
  - Created and installed when a node is federated into a cell
  - Works with the deployment manager to do administrative activities on the node
- Deployment manager
  - Administers multiple application servers from one centralized manager
  - Works with the node agents on each node to manage all the servers in a distributed topology
  - Application server nodes are federated with the deployment manager before the deployment manager can manage them

Federating a cell

Figure 2-3. WebSphere Application Server process types

There are three main types of WebSphere managed processes that make up a cell. These server types interact to do system administration.

- Application server
  - A WebSphere Application Server provides the functions that are required to support and host user applications.
  - An application server runs on only one node, but one node can support many application servers.
- Node agent
  - When a node is federated, a node agent is created and installed on that node.
  - The node agent works with the deployment manager to do administrative activities on the node.
- Deployment manager
  - With the deployment manager, you can administer multiple nodes from one centralized manager.
  - The deployment manager works with the node agent on each node to manage all the servers in a distributed topology.

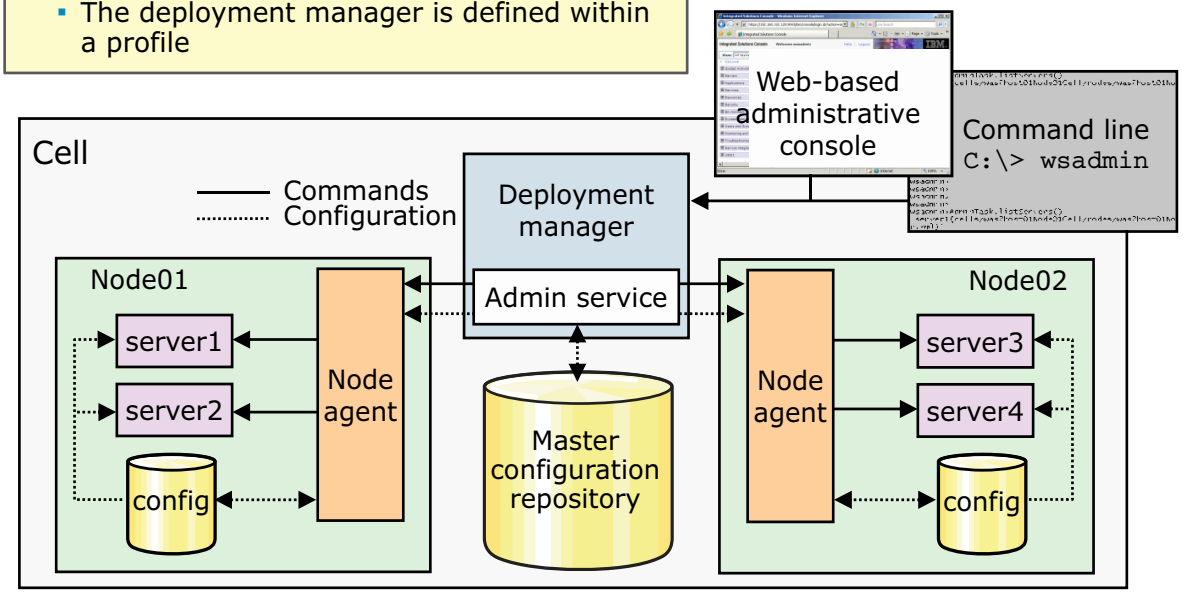
- Application server nodes are federated with the deployment manager before the deployment manager manages them.

IBM Training

Network deployment concepts

- Deployment manager (dmgr)
  - Manages the node agents
  - Holds the configuration repository for the entire management domain, called a **cell**
  - Administrative service runs inside the dmgr
  - The deployment manager is defined within a profile

- Node
  - Logical grouping of servers
  - A single node agent process manages it
  - Each node is defined within a profile



Federating a cell

Figure 2-4. Network deployment concepts

Deployment manager (dmgr):

- The deployment manager works with the node agent on each node to manage all the servers in a distributed topology.
- The deployment manager holds the configuration repository for the entire management domain, called a cell.
- An administrative service runs inside the deployment manager.
- The deployment manager is defined within a profile.

Node:

- A node is a logical group of WebSphere Application Server-managed server processes that share a common configuration repository.
- A single node agent process manages a node.
- A node is associated with a single WebSphere Application Server profile.
- A WebSphere Application Server node does not necessarily have a one-to-one association with a system. One computer can host arbitrarily many nodes, but a node cannot span multiple computer systems.

- A node can contain zero or more application servers.

Configuration repository:

- When a node is part of a cell, the configuration and application files for all nodes in the cell are centralized into a cell master configuration repository.
- The deployment manager server manages the centralized repository and synchronizes the repository to local copies that are held on each node.
- The local copy of the repository that is given to each node contains just the configuration information that the node needs, not the full configuration that the deployment manager maintains.
- When a deployment manager is registered with a job manager, the deployment manager continues to manage the centralized configuration repository.

The graphic includes the following cell environment topology:

- A cell encapsulating:
  - Deployment manager with connections to the node agents of two nodes
    - Administrative service is shown within the deployment manager
  - Master configuration repository with a link to the deployment manager
  - Web-based administrative console that uses the deployment manager administrative console to demonstrate administration of the cell
  - Node01:
    - Node agent with connections to the application servers of the node
    - Local configuration repository with configuration links to the node agent and the application servers within the node
    - server1
    - server2
  - Node02:
    - Node agent with connections to the application servers of the node
    - Local configuration repository with configuration links to the node agent and the application servers within the node
    - server3
    - server4

IBM Training

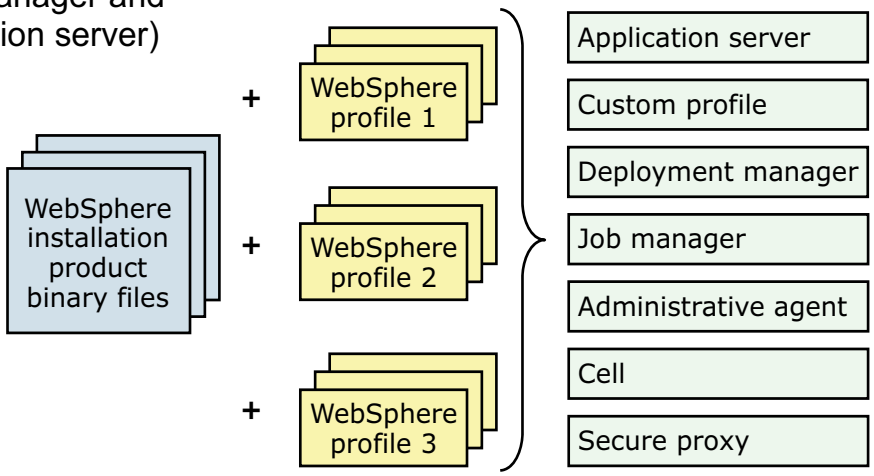
Profiles in network deployment

Profiles represent the nodes

- Multiple nodes can be installed on a single computer
- Nodes can contain a single stand-alone application server
- Nodes can be federated into a cell

Each profile uses the same product files regardless of type:

- Cell (deployment manager and a federated application server)
- Management
  - Administrative agent
  - Deployment manager
  - Job manager
- Application server
- Custom profile
- Secure proxy



Federating a cell

Figure 2-5. Profiles in network deployment

A profile defines the application server configuration and runtime environment. The profile includes all of the files that the server processes in the runtime environment and that you can change. After installing the core product files for the Network Deployment product, you must create a profile.

- Profiles represent the nodes:
  - Multiple nodes can be installed on a single computer.
  - Nodes can contain a single stand-alone application server.
  - Nodes can be federated into a cell.
- Each profile uses the same product files regardless of type:
  - Cell (deployment manager and a federated application server)
  - Management
    - Administrative agent
    - Deployment manager
    - Job manager
  - Application server

- Custom profile
- Secure proxy

The graphic includes a diagram that includes:

- WebSphere installation product files (used by all profiles)
- Three sets of profiles (which can be any of seven profile environment types):
  - WebSphere profile 1
  - WebSphere profile 2
  - WebSphere profile 3
- Here is a list of seven profile environment types:
  - Cell
  - Administrative agent
  - Deployment agent
  - Job manager
  - Application server
  - Custom profile
  - Secure proxy

Application server profile

- Application server profiles provide a base installation
- Application servers in the network deployment product can run as:
  - Part of managed nodes in a cell
  - Stand-alone application servers
- Multiple application server profiles can be created on a single computer
- Each application server profile can be federated into a cell
- Multiple base profiles on a single computer can be federated:
  - Into the same cell
  - Into different cells
  - Remain stand-alone

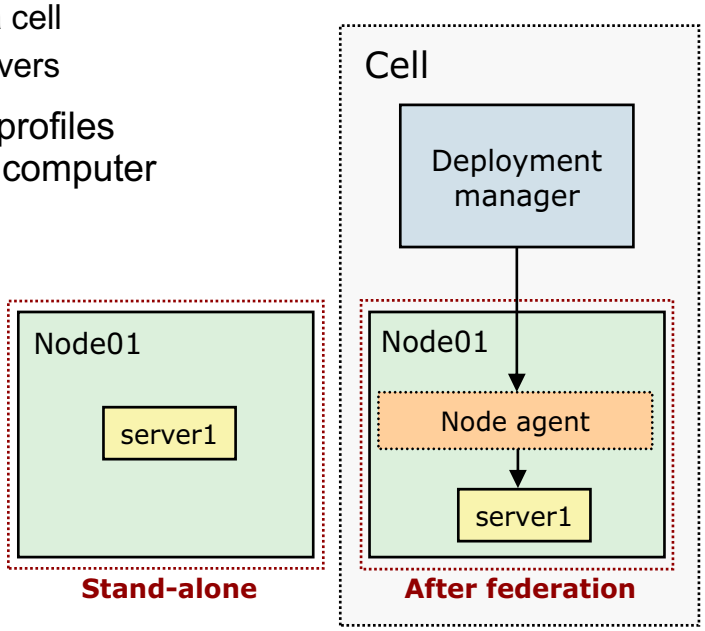


Figure 2-6. Application server profile

Use the application server to make enterprise applications available to the Internet or to an intranet. Application server profiles provide a base installation.

- Application servers in the Network Deployment product can run as:
  - Managed nodes in a deployment manager cell
  - Stand-alone application servers
- Multiple application server profiles can be created on a single computer.
- Each application server profile can be federated into a cell.
- Multiple base profiles on a single computer can accomplish the following tasks:
  - Be federated into the same cell
  - Be federated into different cells
  - Remain a stand-alone profile

An important product feature is the ability to scale up a stand-alone application server profile by adding the application server node into a deployment manager cell. Multiple application server processes in a cell can deploy an application that is in demand. You can also remove an application server node from a cell to return the node to the status of a stand-alone application server.

Each stand-alone application server can optionally have its own administrative console application, which you use to manage the application server. You can also use the wsadmin scripting facility to complete every function that is available in the administrative console application.

No node agent process is available for a stand-alone application server node unless you decide to add the application server node to a deployment manager cell. Adding the application server node to a cell is known as federation. Federation changes the stand-alone application server node into a managed node. You use the administrative console of the deployment manager to manage the node. If you remove the node from the deployment manager cell, then use the administrative console and the scripting interface of the stand-alone application server node to manage the process.

The graphic includes the following cell environment topology:

- Unfederated Node01 containing a stand-alone application server, server1
- A cell encapsulating:
  - Deployment manager
  - Federated Node02
    - Node agent with connections to application servers within the node
    - server2
    - server3

Deployment manager profile

- Is used to create a deployment manager process (dmgr)
- Can exist on an independent computer
- Can exist on a computer with other profiles
- Provides centralized administration of managed application server nodes and custom nodes as a single cell

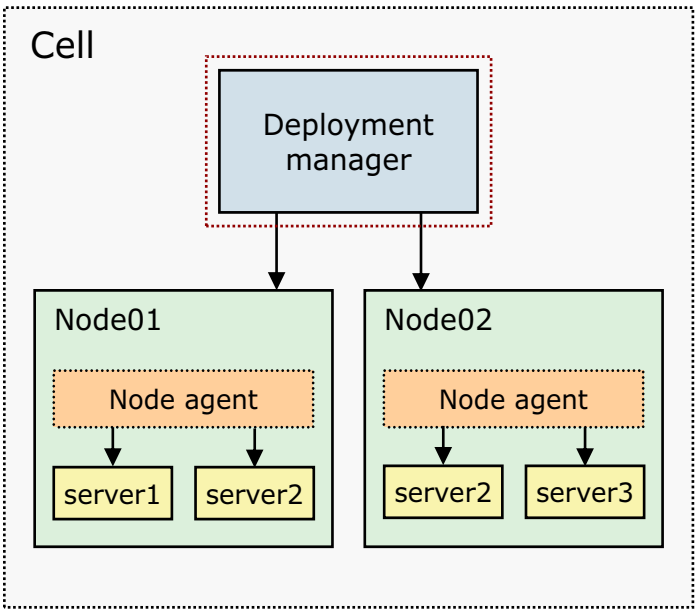


Figure 2-7. Deployment manager profile

The deployment manager profile provides the necessary configuration files for starting and managing the deployment manager server that it contains. The profile also provides everything necessary to configure and manage WebSphere Application Server profiles, or nodes, that are in the deployment manager cell.

The deployment manager profile contains an application server with a server name of dmgr. The dmgr application server is a special application server that contains the deployment manager. The dmgr server contains the Network Deployment administrative console application and the Network Deployment file transfer application. These applications enable the distributed management of one or more WebSphere Application Server profiles, or nodes.

The features of the deployment manager profile are provided here:

- Is used to create a deployment manager process (dmgr)
- Can exist on an independent computer
- Can exist on a computer with other profiles
- Provides centralized administration of managed application server nodes and custom nodes as a single cell

The graphic includes the following cell environment topology:

- A cell encapsulating:
  - Deployment manager with connections to two nodes
  - Federated Node01:
    - Node agent with connections to application servers within the node
    - server1
    - server2
  - Federated Node02:
    - Node agent with connections to application servers within the node
    - server2
    - server3

Custom profile

- A custom profile creates a node without an application
- Automatically federated into a cell during profile creation by default
- No application servers are created during profile creation
- Use the deployment manager administrative console to create servers and clusters on the federated node
- Consider a custom profile as a production-ready shell, ready for customization to contain your servers and applications

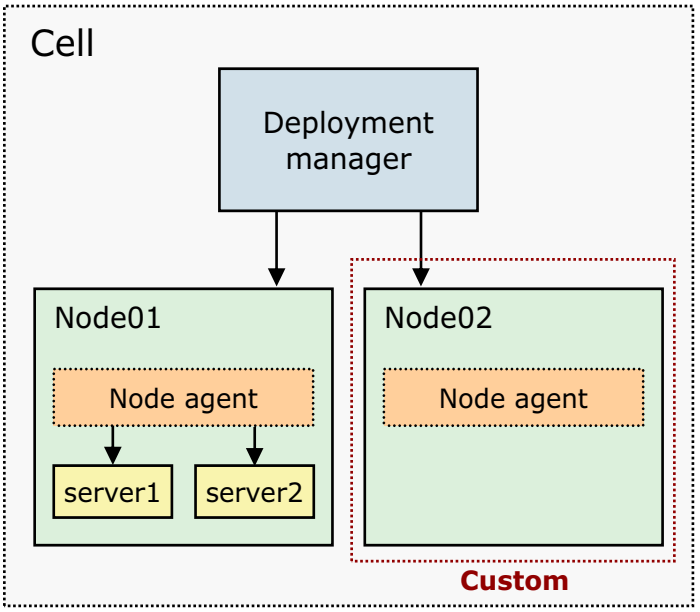


Figure 2-8. Custom profile

Use the custom profile, which belongs to a deployment manager cell, to make enterprise applications available to the Internet or to an intranet under the management of the deployment manager. A custom profile does not have its own administrative console or scripting interface. You cannot manage the node directly with the wsadmin scripting facility.

A custom profile does not include default applications or a default server as the application server profile does. A custom profile is an empty node. Add the node to the deployment manager cell. Then, you can use the administrative interface of the deployment manager to customize the managed node by creating clusters and application servers.

In summary:

- A custom profile creates a node without an application.
- A custom profile automatically is federated into a cell during profile creation by default. In the process, the node agent process is then instantiated on the newly managed node.
- No application servers are created during profile creation.
- Use the deployment manager administrative console to create servers and clusters on the federated node.

- Consider a custom profile as a production-ready shell, ready for customization to contain your servers and applications.

The graphic includes the following cell environment topology:

- A cell encapsulating:
  - Deployment manager with connections to two nodes
  - Federated Node01:
    - Node agent with connections to application servers within the node
    - server1
    - server2
  - Federated Node02 (custom profile):
    - Node agent
    - No application servers

Creating profiles

Profile Management Tool

- Start menu (Windows only)
- Started from installation wizard
- Launch command-line tool **pmt.sh**
  - *<was_root>*/bin/ProfileManagement/
  - Similar command exists for UNIX
  - Wizard in First steps console

manageprofiles

- Command-line tool
- Use **manageprofiles -silent** option to create profiles in silent mode
- Other **manageprofiles** options include: **-listProfiles -delete**

Figure 2-9. Creating profiles

You can use the Profile Management Tool or the `manageprofiles` command to create profiles.

- Profile Management Tool:
  - The tool is started in several ways:
    - Start menu (Windows only)
    - Started from the installation wizard
    - Started with the command-line tool: `pmt.sh`
      - In *<was_root>*/bin/ProfileManagement/
      - A similar command exists for UNIX
    - The wizard is available from the First steps console
  - The graphic displays the Profile Management Tool Environment Selection window. Five environments can be selected from the listing:
    - Cell (deployment manager and a federated application server)
    - Management
    - Application server

  - Custom profile
  - Secure proxy (configuration-only)

  The Profile Management Tool refers to profiles as **environments** because selecting **cell** results in creating two profiles: a deployment manager profile and a federated application server profile. There is really no such thing as a cell profile. It is just a fast way to build a cell. Additionally, there is no such thing as a management profile. There are three types of management profiles: administrative agent, deployment manager, and job manager.

- manageprofiles
  - manageprofiles is a command-line tool.
  - You can use the `manageprofiles -silent` option to create profiles in silent mode.
  - Other `manageprofiles` options include: `-create`, `-listProfiles`, and `-delete`
  - The graphic displays the following example:

```
manageprofiles -create -profileName profile3 -profilePath
"/opt/IBM/WebSphere/Appserver/profiles/profile3" -templatePath
"/opt/IBM/WebSphere/Appserver/profileTemplates/default" -nodeName washost01Node03
-cellName washost01Cell03 -hostname washost01
```

The resulting output produces the following message:

```
"INSTCONFSUCCESS: Success: Profile profile3 now exists."
```

Profiles can be silently created in the following ways:

- As part of a silent WebSphere installation process
- Manually by using the `-silent` option with the pctWindow utility

Silent creation of profiles requires a response file, which should be customized:

- Dmgr: `responsefile.pct.NDdmgrProfile.txt`
- Application Server: `responsefile.pct.NDStand-aloneProfile.txt`
- Custom: `responsefile.pct.NDmanagedProfile.txt`

IBM Training

Profile Management Tool: Launch and create

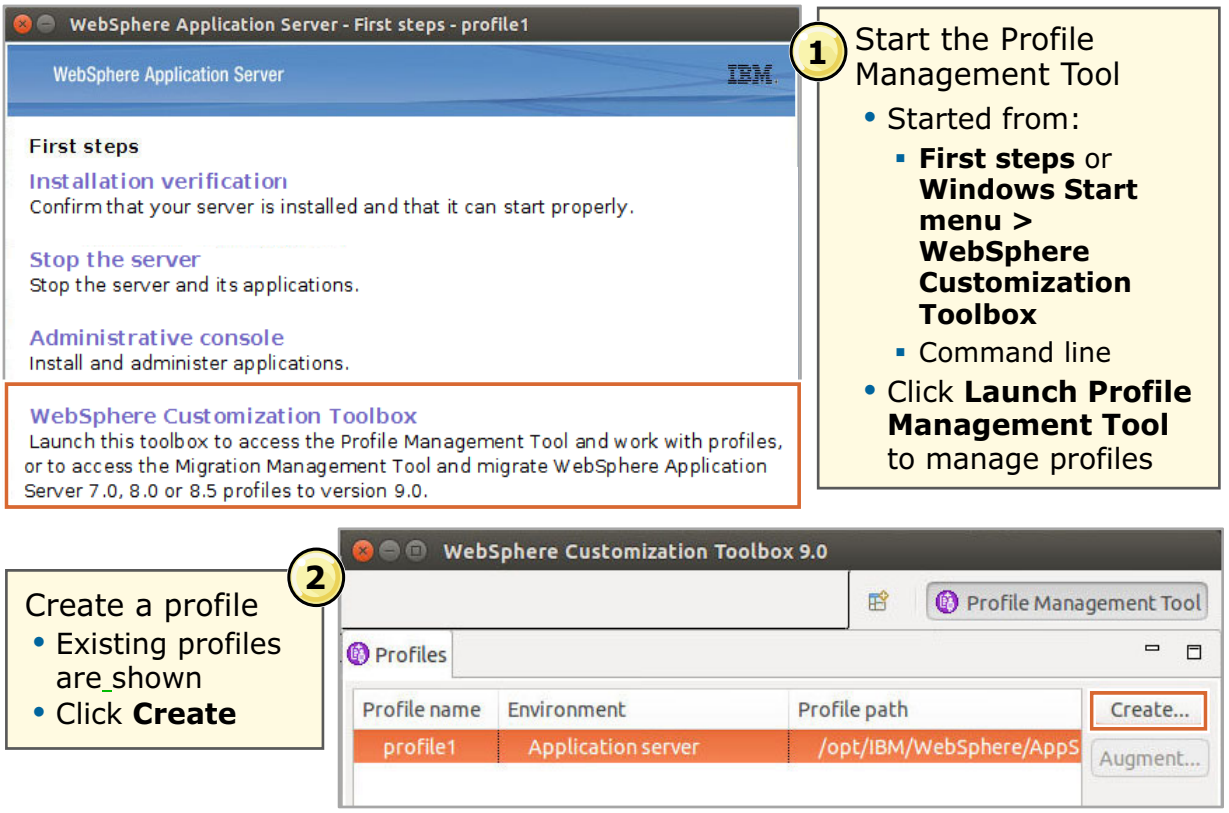


Figure 2-10. Profile Management Tool: Launch and create

1. To create a profile, launch the Profile Management Tool.
   - The tool can be started in the following ways:
     - Following installation, if selected on the installation results window; the Profile Management Tool welcome window displays
     - From the command line with `pmt.sh`
   - As soon as it is started, click **Launch Profile Management Tool** to manage profiles.
2. The Profile Management Tool provides a listing of existing profiles and an option to create a profile.
   - The profile list displays existing profiles. In the example, profile1 is shown.
   - To begin the process of creating a profile, click **Create**.

IBM Training

Profile Management Tool: Environment and server type

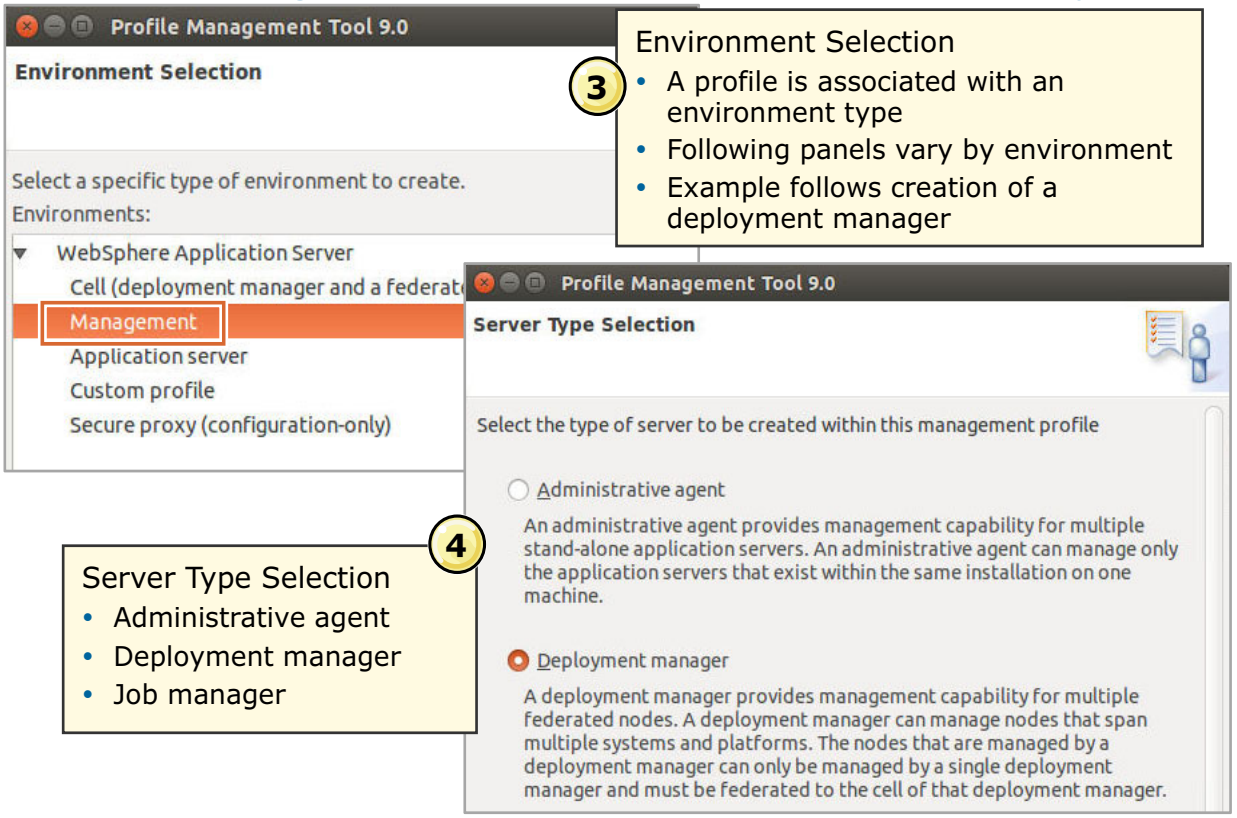


Figure 2-11. Profile Management Tool: Environment and server type

3. Use the options from the Environment Selection window to choose the type of environment that is associated with your new profile. A profile is associated with one of seven environment types.

   - **Cell (deployment manager and a federated application server)**
     A cell environment creates two profiles: a management profile with a deployment manager and an application server profile. The application server is federated to the cell of the deployment manager.
   - **Management**
     A management profile provides the server and services for managing multiple application server environments. The administrative agent manages application servers on the same computer. The Network Deployment edition also includes a deployment manager for tightly coupled management and a job manager for loosely coupled management of topologies that are distributed over multiple computers.
   - **Application server**
     An application server environment runs your enterprise applications. WebSphere Application Server is managed from its own administrative console and functions independently from all other application servers.
   - **Custom profile**
     A custom profile contains an empty node, which does not contain an administrative console

or servers. The typical use for a custom profile is to federate its node to a deployment manager. After federating the node, use the deployment manager to create a server or a cluster of servers within the node.

- **Secure proxy (configuration-only)**
  A secure proxy configuration-only profile is for use with a DMZ secure proxy server. You cannot start the secure proxy server on the Network Deployment installation. This configuration-only profile is intended to be used only to configure the profile by using the administrative console.

4. When you select the management environment, the Server Type Selection window is displayed. You can select one of three types of management servers to create.

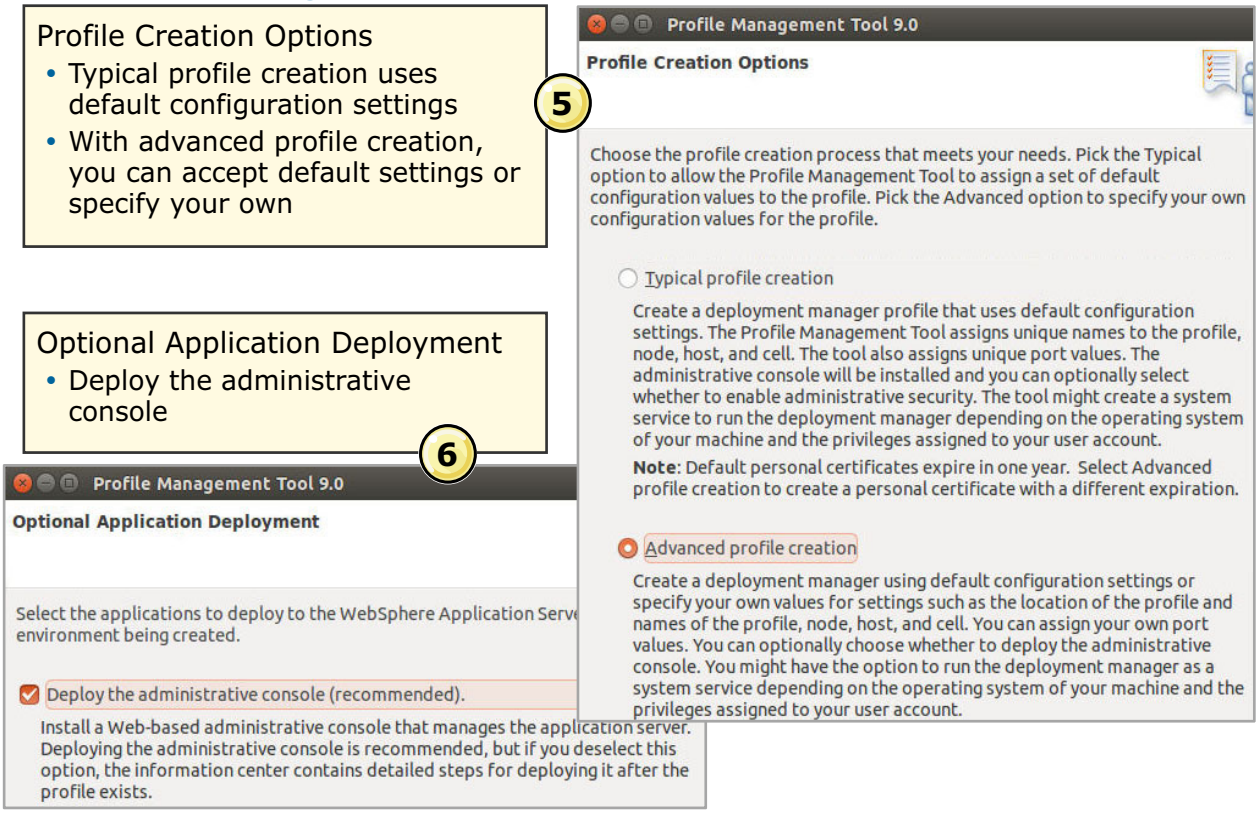
   - **Administrative agent**
     An administrative agent provides management capabilities for multiple stand-alone application servers. An administrative agent can manage only the application servers that exist within the same installation on one computer.
   - **Deployment manager**
     A deployment manager provides management capabilities for multiple federated nodes. A deployment manager can manage nodes that span multiple systems and platforms. A single deployment manager can manage a node, and the node must be federated to the cell of that deployment manager.
   - **Job manager**
     A job manager provides management capabilities for multiple stand-alone application servers, administrative agents, and deployment managers. The job manager can manage nodes that span multiple systems and platforms. If a job manager manages a node or nodes, other job managers can also manage those nodes.

IBM Training

Profile Management Tool: Options

Profile Creation Options
- Typical profile creation uses default configuration settings
- With advanced profile creation, you can accept default settings or specify your own

Optional Application Deployment
- Deploy the administrative console



Federating a cell

Figure 2-12. Profile Management Tool: Options

5. Configuration values can be assigned based on default configuration values, or you can specify your own configuration values. Use the options in the Profile Creation Options window to choose which option best fits your needs.

   - **Typical profile creation**
     This option creates an application server profile that uses default configuration settings. The Profile Management Tool assigns unique names to the profile, node, and host. The tool also assigns unique port values.

     The administrative console and the default application are installed. You can optionally select whether to enable administrative security.

     The tool might create a system service to run the application server, which depends on the operating system of your computer and the privileges that are assigned to your user account.

     Typical profile creation is the default.

   - **Advanced profile creation**
     This option uses default configuration settings to create an application server or specifies your own values for settings, such as the location of the profile and names of the profile, node, and host.

You can assign your own port values. You can optionally choose whether to deploy the administrative console and sample applications, and create a web server definition.

There is an option to run the application server as a system service, which depends on the operating system of your machine and the privileges that are assigned to your user account.

6. From the Optional Application Deployment window, you can select the applications to deploy to the WebSphere Application Server environment that is being created. The options available from this window depend on the type of profile you are creating. In the example, a deployment manager profile is being created. There is only one option to choose for this type of profile.
   - **Deploy the administrative console (recommended)**
     This option installs a web-based administrative console that manages the application server. Deploying the administrative console is suggested, but if you clear this option, the information center provides detailed steps for deploying it after the profile exists. This option is selected by default.

IBM Training

Profile Management Tool: Names and location

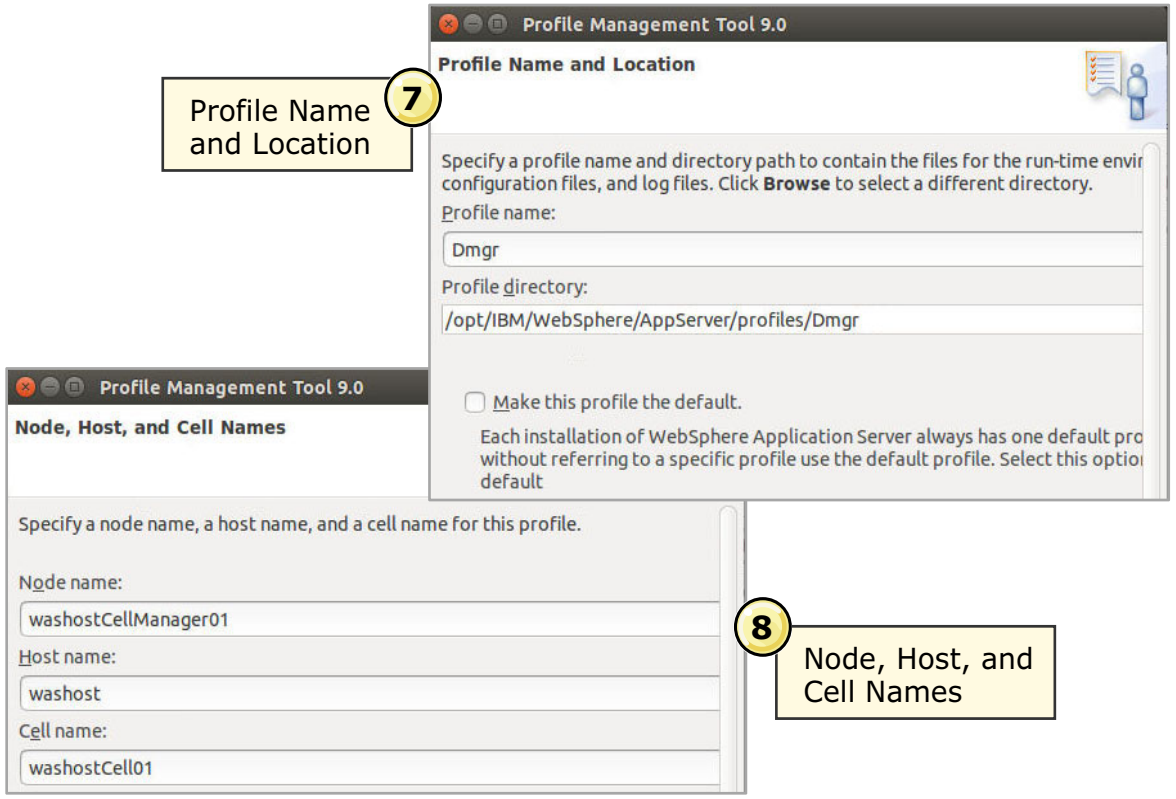


Figure 2-13. Profile Management Tool: Names and location

7. From the “Profile Name and Location” window, the profile name and profile directory path are specified. The directory that is named contains the files for the runtime environment, such as commands, configuration files, and log files.
   - **Profile name**
     The default name can be changed as you choose.
   - **Profile directory**
     The default directory can be changed as you choose.
   - Depending on the profile that being created, more options can be specified by selecting the appropriate check box. In this example, a deployment manager profile has one option.
     - **Make this profile the default**
       Each installation of WebSphere Application Server always has one default profile. Commands that run without referring to a specific profile use the default profile.
8. Use the options in the “Node, Host, and Cell Names” window to specify the following options:
   - **Node name**
     A node name is used for administration. If the node is federated, the name must be unique within the cell.

- **Host name**
  A host name is the Domain Name System (DNS) name (short or long) or the IP address of the computer.
- **Cell name**
  A cell name is a logical name for the group of nodes that the deployment manager manages.

IBM Training

Profile Management Tool: Security

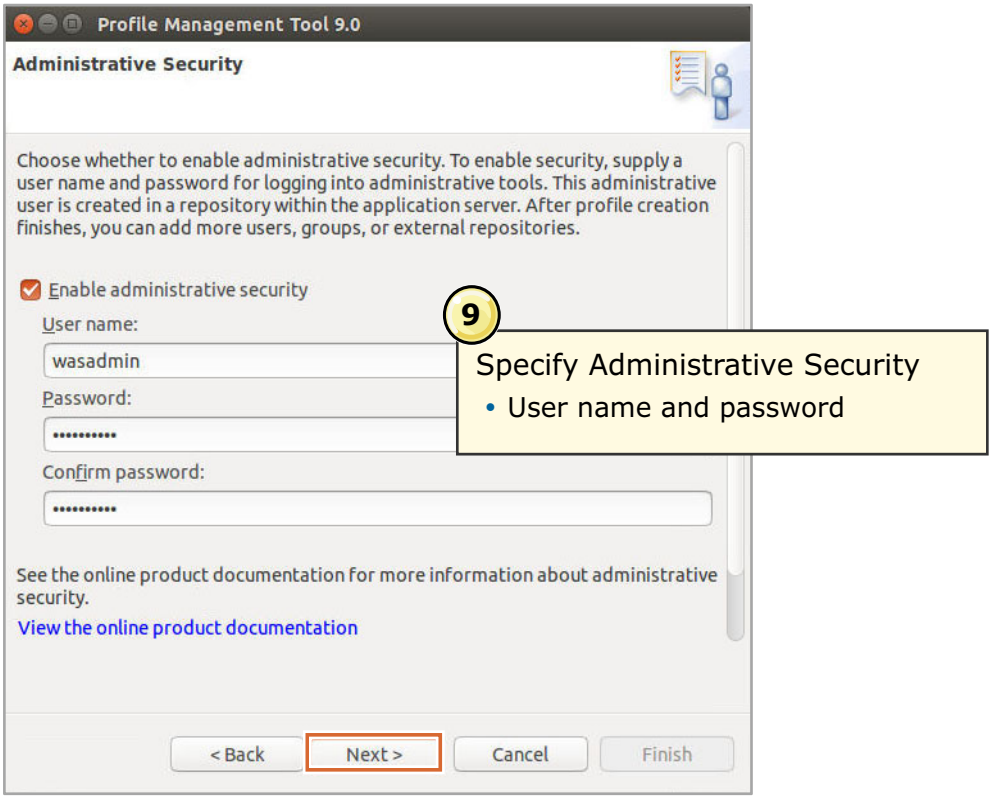


Figure 2-14. Profile Management Tool: Security

9. From the Administrative Security window, choose whether to enable administrative security. If enabled, the administrative user that is specified is created in a repository within the application server. After profile creation finishes, you can add more users, groups, or external repositories.
   - **Enable administrative security**
     Select this check box to enable administrative security.
   - **User name**
     If enabling administrative security, provide a user name for administrative security.
   - **Password**
     If enabling administrative security, provide a password. The password that you enter must be confirmed before proceeding.

IBM Training

Profile Management Tool: Security certificate (1 of 2)

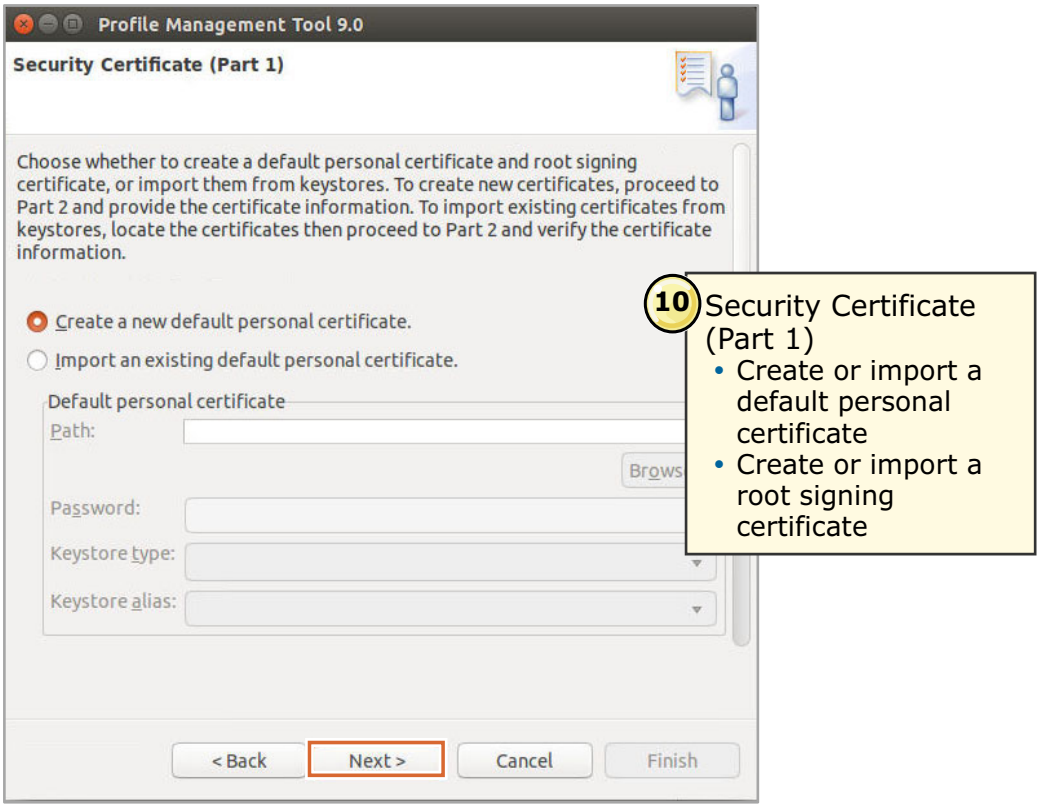


Figure 2-15. Profile Management Tool: Security certificate (1 of 2)

10. From the Security Certificate (Part 1) window, choose whether to create a default personal certificate and root signing certificate, or import them from keystores. To create certificates, proceed to security certificate (part 2) and provide more certificate information. To import existing certificates from keystores, locate the certificates; then, proceed to security certificate (part 2) and verify the certificate information.
    - There are two options for a default personal certificate:
        - Click **Create a new default personal certificate**:
          If this option is chosen, you provide certificate information about the Security Certificate (part 2).
        - Click **Import an existing default personal certificate** and provide the following details:
            - **Path**
              Specify the directory location of the default personal certificate.
            - **Password**
              Specify the password for the default personal certificate.
            - **Keystore type**
              There are four keystore types to choose from the list. The options are: JKS, JCEKS, PKCS12, and CMSKS.

- **Keystore alias**

- There are two options for a root signing certificate:
  - Click **Create a new root signing certificate**:
    If this option is chosen, you provide certificate information about the Security Certificate (part 2) window.
  - Click **Import an existing root signing certificate** and provide the following details:
    - **Path**
      Specify the directory location of the root signing certificate.
    - **Password**
      Specify the password for the root signing certificate.
    - **Keystore type**
      There are four keystore types to choose from the list. The options are: JKS, JCEKS, PKCS12, and CMSKS.

IBM Training

Profile Management Tool: Security certificate (2 of 2)

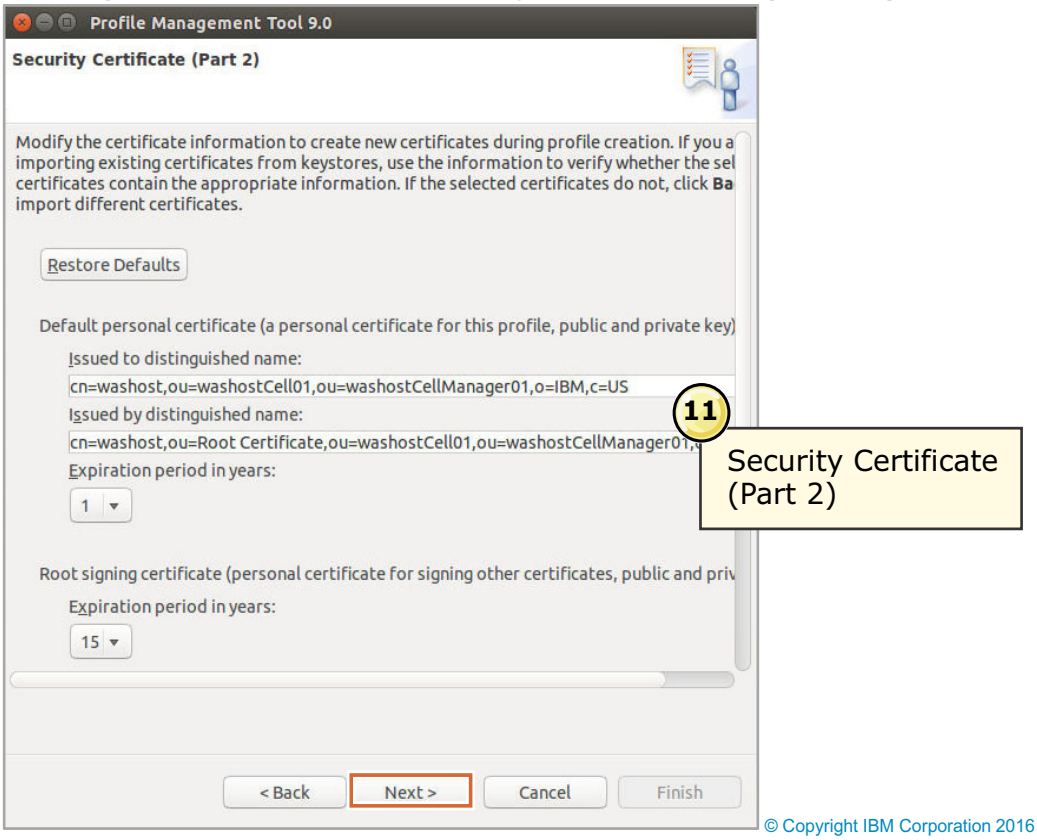


Figure 2-16. Profile Management Tool: Security certificate (2 of 2)

11. On the Security Certificate (Part 2) window, you can modify the certificate information to create certificates during profile creation. If you are importing certificates from keystores, use the information to verify whether the selected certificates contain the appropriate information.
    - The following information must be provided for a default personal certificate (a personal certificate for this profile, public and private key):
      - **Issued to distinguished name**
        The default distinguished name is: `cn="computer IP address", ou="host_name"Node01Cell,ou="host_name"Node01,o=ibm,c=US`
      - **Issued by distinguished name**
        The default distinguished name is: `cn="computer IP address",ou=Root Certificate,ou="host_name"Node01Cell,ou="host_name"Node01,o=ibm, c=US`
      - **Expiration period in years**
        The default is one year. You can choose 1–15 years.
    - The following information must be provided for a root signing certificate (a personal certificate for signing other certificates, public and private key):
      - **Expiration period in years**
        The default is 15 years. You can choose between 15, 20, and 25 years.

- The default keystore password must be specified. The default password, `WebAS`, is supplied in the password field and in the confirmation field.

  The default value for the keystore is documented in the information center and should be changed to protect the security of the keystore files and SSL configuration.
- If you make a mistake after making any configuration changes, you can always restore the default values by clicking **Restore Defaults**.

IBM Training

Profile Management Tool: Ports

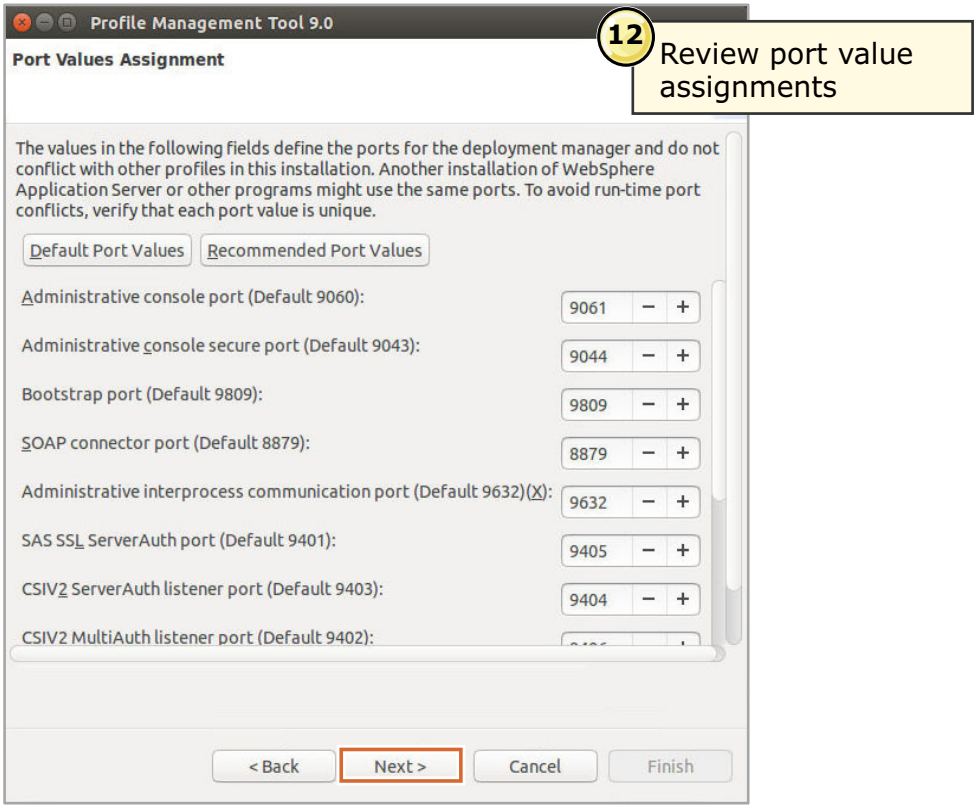


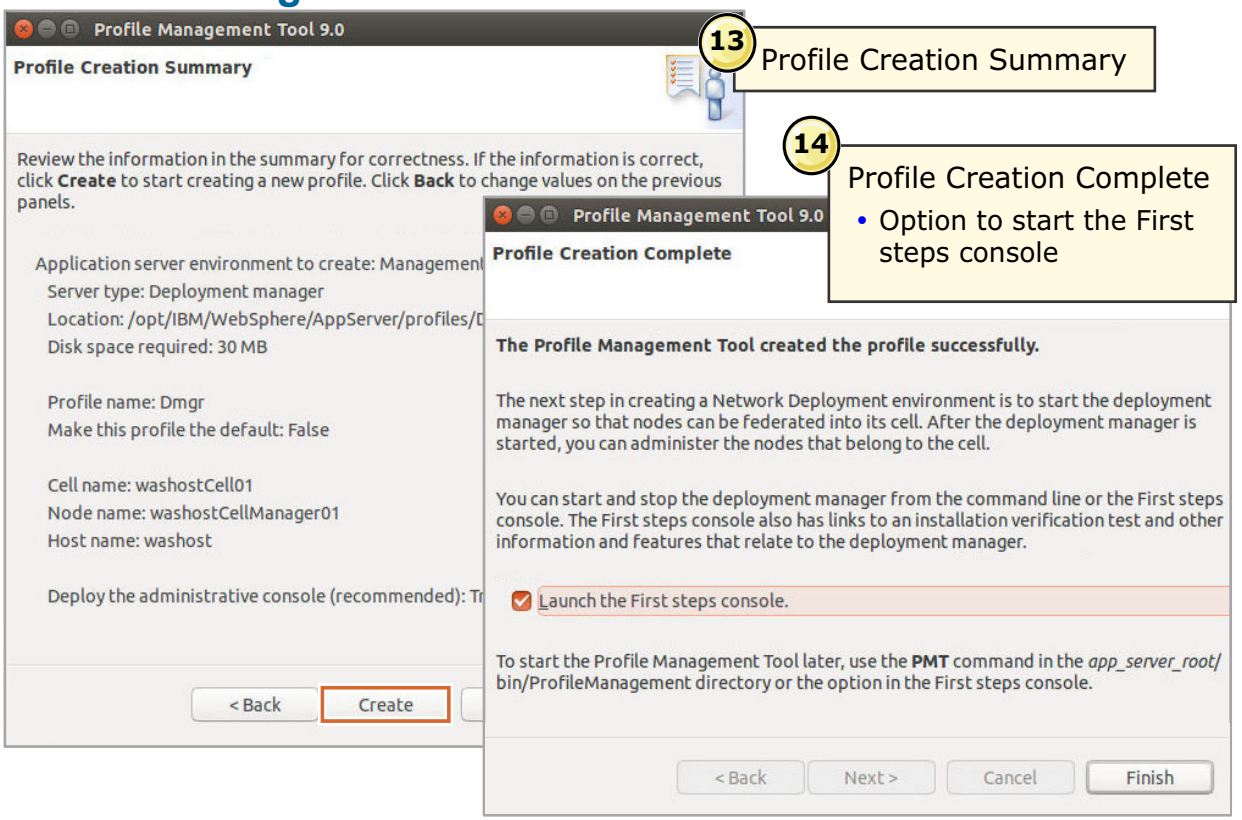
Figure 2-17. Profile Management Tool: Ports

12. The port values assignment window lists the values of the ports that the application server uses. The values that define the ports do not conflict with other profiles in the current installation.
    - Three options are available for you to choose ports:
      - Manually enter the port value for each port.
      - Click **Default Port Values** to populate the port values with the default port values.
      - Click **Recommended Port Values** to populate the port values based on suggested values that the Profile Management Tool calculates.

    Another installation of WebSphere Application Server or other programs might use the same ports. To avoid runtime port conflicts, verify that each port value is unique. If the Profile Management Tool detects conflicts, a window displays at the top of the window and list the ports for which activity is detected.

IBM Training

Profile Management Tool: Results and exit



Federating a cell

Figure 2-18. Profile Management Tool: Results and exit

13. When all Profile Management Tool windows are completed, the Profile Creation Summary window displays.
    - The summary information is based on the selections you made previously.
    - Review the summary for correctness before proceeding with the profile creation.
14. Review the results that are shown on the Profile Creation Complete window. If you want to launch the First steps console, select the **Launch the First steps console** check box. The First steps console is then started when you click **Finish**.

    The window also displays information that is based on the type of profile that was created. In the figure above, the window provides information about an application server.

    When you click **Finish** from the profile creation complete window, you are returned to the Profile Management tool welcome window. The profile that you created is now shown in the profile listing. You can continue to create more profiles or exit the tool. To exit the Profile Management Tool, click **File > Exit**.

IBM Training

Profile creation: Command-line tool

The `manageprofiles` script supports a number of functions:

- Create stand-alone application server profiles
  `manageprofiles -create`
- List all profiles
  `manageprofiles -listProfiles`
- Delete profiles
  `manageprofiles -delete -profileName`

Figure 2-19. Profile creation: Command-line tool

Use the `manageprofiles` command to create, delete, augment, back up, and restore profiles, which define runtime environments. The manageprofiles script supports a number of functions:

Create stand-alone application server profiles:

- manageprofiles -create

  An example is:

```
manageprofiles -create -profileName profile3 -profilePath
"/opt/IBM/WebSphere/Appserver/profiles/profile3" -templatePath
"/opt/IBM/WebSphere/Appserver/profileTemplates/default" -nodeName washost01Node03
-cellName washost01Cell03 -hostname washost01
```

  If successful, the output displays the message: `"INSTCONFSUCCESS: Success: Profile3 now exists."`

List all profiles:

- `manageprofiles -listProfiles`

  An example is:

- `manageprofiles -listProfiles`

The output displays a list of profiles, such as: `[profile1, DmgrProfile, profile2, profile3]`

Delete profiles:

- `manageprofiles -delete -profileName`

  Here is an example:

- `manageprofiles -delete -profileName profile3`

  The output displays the following message if successful: `INSTCONFSUCCESS: Success: The profile no longer exists`.

  Deleting a profile leaves a number of files behind, including the `logs` directory. To delete all of these files, the profile must be deleted manually.

Directory structure

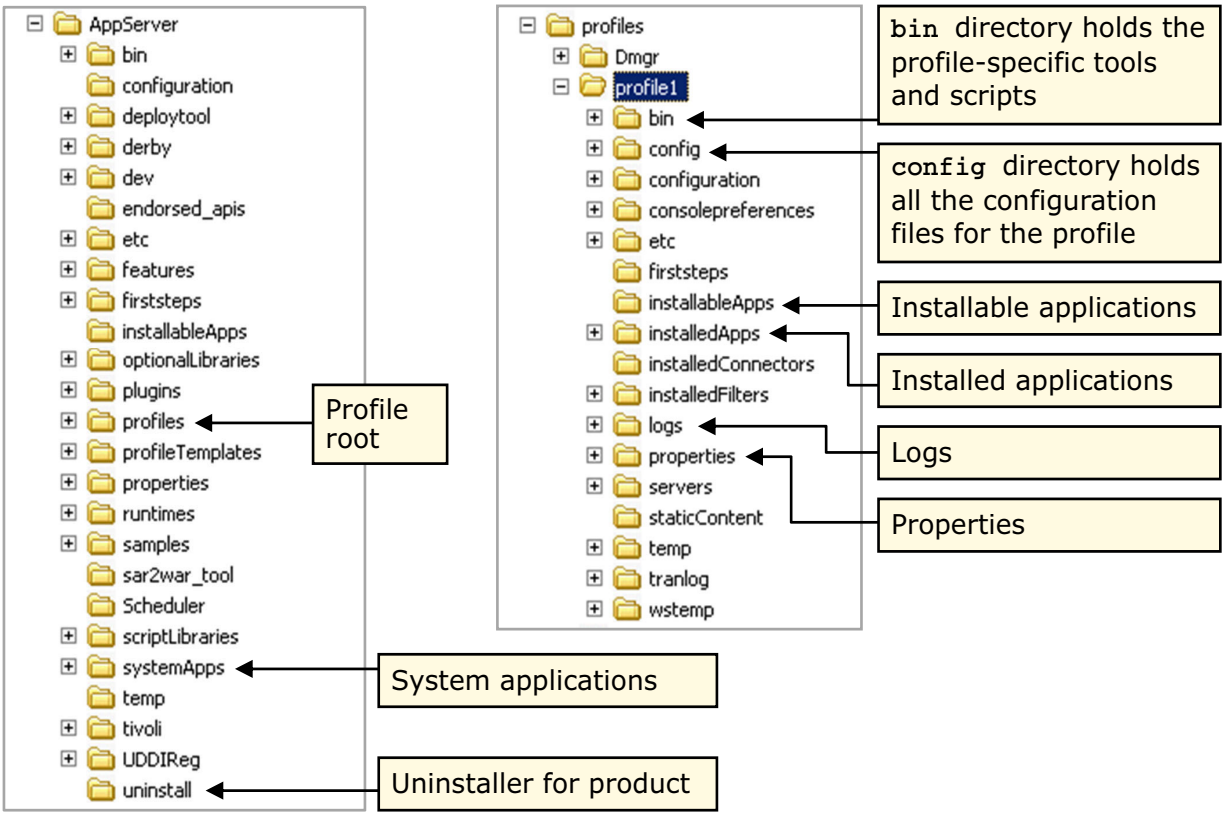


Figure 2-20. Directory structure

The following file paths are default locations. You can install the product and other components or create profiles in any directory where you have write access. Multiple installations of WebSphere Application Server products or components require multiple locations.

- WebSphere Application Server: **/opt/IBM/WebSphere/AppServer**
- Profile root: *<was_root>*/**profiles/**
- Profiles: *<was_root>*/**profiles/profile_name**
- System applications: *<was_root>*/**systemApps**
- Uninstaller for the product: *<was_root>*/**uninstall**

The directory of each profile contains the same standard directories, which include:

- `bin`
  The `bin` directory holds the profile-specific tools and scripts.
- `config`
  The `config` directory holds all the configuration files for the profile.
- `installableApps`
  The `installableApps` directory holds applications that can be associated with the profile, but are not currently installed.

- `installedApps`
  The `installedApps` directory holds all installed applications for the profile.
- `logs`
  The `logs` directory holds all log files that are associated with the profile.
- `properties`
  The `properties` directory holds properties files that are associated with the profile.

IBM Training

Server commands review

- WebSphere commands are profile aware
  - Many WebSphere commands have a `-profileName` option
  - Or issue the commands from the appropriate directory:
    `<profile_root>/<profile_name>/bin`
- If no profile is used, the default profile is assumed
  - Only one default profile can exist
  - Unless otherwise manually set, the first profile that is created is the default profile
- Examples (from `<was_root>/bin`):
  - `startServer server1 -profileName profile1`
  - `startManager -profileName DmgrProfile`
  - `stopServer server1` (assumes default profile)

Figure 2-21. Server commands review

WebSphere commands are profile aware. There is a `-profileName` option on many WebSphere V9 commands to specify that the profile or the command can be issued from the appropriate directory without specifying a profile name. For example: `<profile_root>/<profile>/bin`

If no profile is specified, the default profile is assumed. Keep in mind that there can be only one default profile. Unless otherwise manually set, the first profile that is created is the default profile. It is suggested that you always specify the name of the profile.

Examples of server commands include:

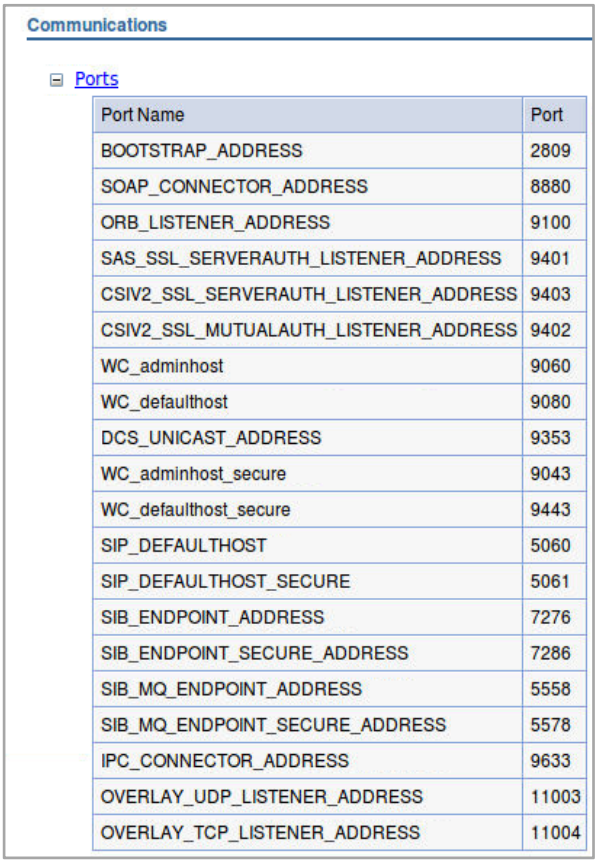
```
startServer server1 -profileName profile1
startManager -profileName DmgrProfile
stopServer server1
```

IBM Training

Profile precautions

When multiple profiles are created on a single computer, be careful:

- Use the correct profile `bin` directory to issue the following commands:
  - `startServer`
  - `stopServer`
  - `serverStatus`
- Be aware of possible port conflicts for node agents and application servers
- A single computer can have multiple server1 instances
- Ensure that consistent host names within a computer are used



Port Name	Port
BOOTSTRAP_ADDRESS	2809
SOAP_CONNECTOR_ADDRESS	8880
ORB_LISTENER_ADDRESS	9100
SAS_SSL_SERVERAUTH_LISTENER_ADDRESS	9401
CSIV2_SSL_SERVERAUTH_LISTENER_ADDRESS	9403
CSIV2_SSL_MUTUALAUTH_LISTENER_ADDRESS	9402
WC_adminhost	9060
WC_defaulthost	9080
DCS_UNICAST_ADDRESS	9353
WC_adminhost_secure	9043
WC_defaulthost_secure	9443
SIP_DEFAULTHOST	5060
SIP_DEFAULTHOST_SECURE	5061
SIB_ENDPOINT_ADDRESS	7276
SIB_ENDPOINT_SECURE_ADDRESS	7286
SIB_MQ_ENDPOINT_ADDRESS	5558
SIB_MQ_ENDPOINT_SECURE_ADDRESS	5578
IPC_CONNECTOR_ADDRESS	9633
OVERLAY_UDP_LISTENER_ADDRESS	11003
OVERLAY_TCP_LISTENER_ADDRESS	11004

Federating a cell

Figure 2-22. Profile precautions

When multiple profiles are created on a single computer, be careful:

- Use the correct profile `bin` directory for the following commands:

  `startServer`
  `stopServer`
  `serverStatus`

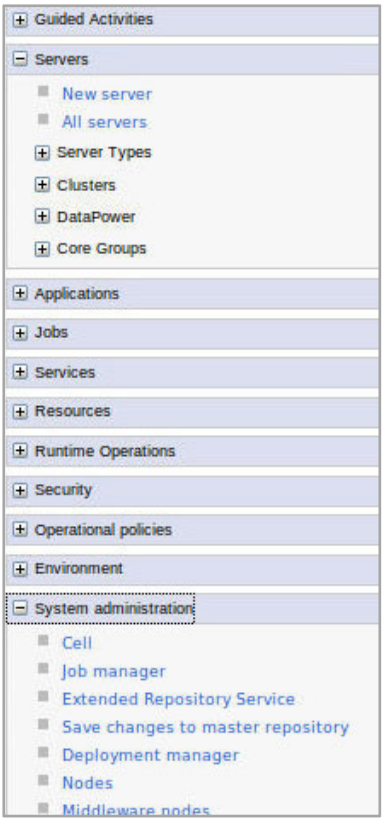
- Be aware of possible port conflicts for node agents and application servers. The graphic displays a list of ports that are shown in the administrative console for a resource.
- There can be multiple server1 instances on a single computer.
- Ensure that consistent host names within a computer are used.

  Be careful when using the Profile Management Tool. It is possible that it preinstalls a default host name by adding the default DNS suffix to the short machine name, which can cause problems if other profiles used only the short host name.

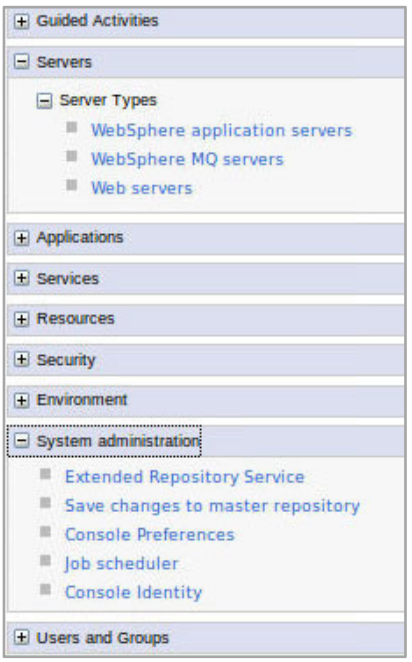
  It does not matter which form is used (short name or fully qualified name), on the condition that the name is used consistently. For example, `washost` and `washost.ibm.com` are different.

IBM Training

Deployment manager console versus stand-alone console



Stand-alone



Deployment manager

- Deployment manager administrative console has more functions for administration of the cell

Figure 2-23. Deployment manager console versus stand-alone console

The deployment manager administrative console provides more tasks for administration of a cell that the stand-alone administrative console does not provide. The example demonstrates a few differences between the deployment manager and stand-alone administrative consoles.

- Under **Servers**
  The deployment manager provides more server task options for managing clusters, DataPower, and core groups.
- Under **System administration**
  The deployment manager provides management tasks that are associated with administering a cell, nodes, node agents, and node groups.

Common command-line tools

- In several directories:
  - *<was_root>*/bin
  - *<profile_root>*/*<profile_name>*/bin
- Tools include:

Command	Function
addNode	Add a node to a cell
syncNode	Synchronize a node with the cell configuration
removeNode	Remove a node from a cell
cleanupNode	Cleans up a node configuration from the cell repository
startNode	Start the node agent
stopNode	Stop the node agent
startManager	Start the deployment manager
stopManager	Stop the deployment manager

Figure 2-24. Common command-line tools

Several tools are commonly used in a cell environment. The command-line tools are in more than one directory:

<was_root>/bin
<profile_root>/*<profile_name>*/bin

The tools include:

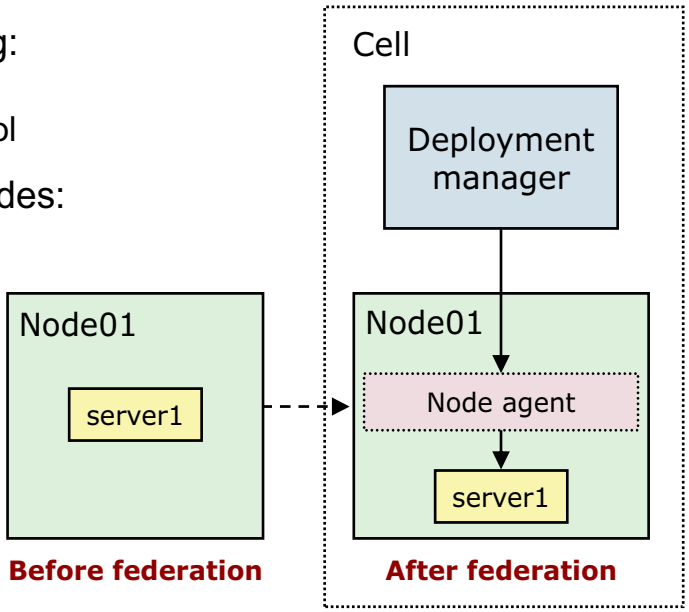
- addNode
  The addNode command incorporates an application server installation into a cell. Depending on the size and location of the new node you incorporate into the cell, this command can take a few minutes to complete.
- syncNode
  The syncNode command forces a configuration synchronization to occur between the node and the deployment manager for the cell in which the node is configured.
- removeNode
  The removeNode command returns a node from a Network Deployment distributed administration cell to a stand-alone application server installation. The removeNode command removes the node-specific configuration from the cell. This command does not uninstall any applications that are installed as the result of running an addNode command.

- `cleanupNode`
  The `cleanupNode` command cleans up a node configuration from the cell repository. Use this command to clean up a node if you have a node that is defined in the cell configuration, but the node no longer exists.
- `startNode`
  The `startNode` command reads the configuration file for the node agent process and constructs a launch command. You do not have to use a user name and password with the `startNode` command because this command starts a server process but does not invoke an MBean method.
- `stopNode`
  The `stopNode` command reads the configuration file for the Network Deployment node agent process and sends a Java Management Extensions (JMX) command that tells the node agent to shut down. By default, the `stopNode` command waits for the node agent to complete shutdown before it returns control to the command line.
- `startManager`
  The `startManager` command starts the deployment manager. You do not have to use a user name and password with the `startManager` command because this command starts a server process but does not invoke an MBean method.
- `stopManager`
  The `stopManager` command stops the deployment manager. It sends a Java Management Extensions (JMX) command to the manager to shut down. By default, the `stopManager` command waits for the manager to complete the shutdown process before it returns control to the command line.

IBM Training

Adding a node to a cell

- Add a node to a cell by using:
  - Administrative console, or
  - `addNode` command-line tool
- Adding a node to a cell includes:
  - Creation of a backup of current configuration
  - Connection to the deployment manager
  - Configuration of the node agent
  - Addition of applications of node to cell configuration



- After the node is added:
  - Use `startNode` to start the node agent
  - Use `syncNode` to synchronize a node

Federating a cell

Figure 2-25. Adding a node to a cell

The process of adding a node to a cell is known as federation. The process of federation creates a managed node with an application server and a node agent that belongs to a deployment manager cell.

A node is added to a cell by using:

- The deployment manager administrative console, or
- The `addNode` command-line tool

The process of adding a node to a cell includes:

- Creation of a backup of the current configuration
- Connection to the deployment manager
- Configuration of the node agent
- Addition of applications of the node to the cell configuration

After the node is added to the cell, the following commands can be used:

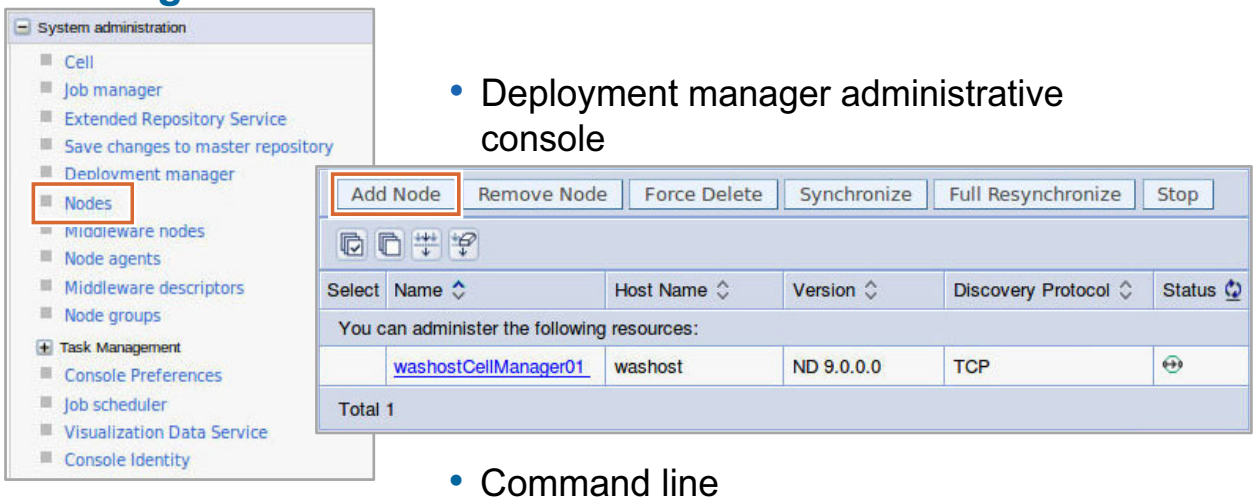
- `startNode` to start the node agent
- `syncNode` to synchronize a node

The graphic describes the process of adding a node to a cell with the following topology:

- An unfederated node that includes a stand-alone application server
- Following federation:
  - A cell encapsulating:
    - Deployment manager with connection to the node agent for the newly managed node
    - Node agent with connection to the application server of the node
    - Application server

IBM Training

Adding a node



- Deployment manager administrative console
- Command line

```
addNode dmgr_host [dmgr_port] [-profileName profilename]
[-conntype type] [-excludesecuritydomains true | false] [-includeapps]
[-startingport portnumber] [-portprops qualified_filename]
[-nodeagentshortname name] [-nodegroupname name]
[-includebuses] [-registerservice] [-serviceusername name]
[-servicepassword password] [-coregroupname name] [-noagent]
[-statusport 1231] [-quiet] [-nowait] [-logfile filename] [-replacelog]
[-trace] [-username uid] [-password pwd] [-localusername localuid]
[-localpassword localpwd] [-help]
```

Federating a cell

Figure 2-26. Adding a node

A node can be added to a cell by one of two methods:

- Deployment manager administrative console:
  - From the navigation tree, expand **System administration**. Click **Nodes > Add Node**.
- Command line:
  - The `addNode` command is used to add a node to a cell. The command syntax is:

```
addNode dmgr_host [dmgr_port] [-profileName profilename]
[-conntype type] [-excludesecuritydomains true | false] [-includeapps]
[-startingport portnumber]
[-portprops qualified_filename] [-nodeagentshortname name]
[-nodegroupname name] [-includebuses] [-registerservice]
[-serviceusername name] [-servicepassword password]
[-coregroupname name] [-noagent] [-statusport 1231] [-quiet] [-nowait] [-logfile
filename] [-replacelog] [-trace] [-username uid]
[-password pwd] [-localusername localuid]
[-localpassword localpwd] [-help]
```

  - Here is an example:

- Add profile `profile4` to the cell managed by profile `dmgr01`:

```
addNode dmgr01 8879 -profileName profile4
```

IBM Training

Managed versus unmanaged nodes

- Managed nodes
  - Use node agent or administrative agent to manage their servers
  - Application server process runs within the deployment manager cell
- Unmanaged nodes
  - Node agent or administrative agent does not manage its servers
  - A stand-alone application server is an unmanaged node
  - Commonly used for web servers

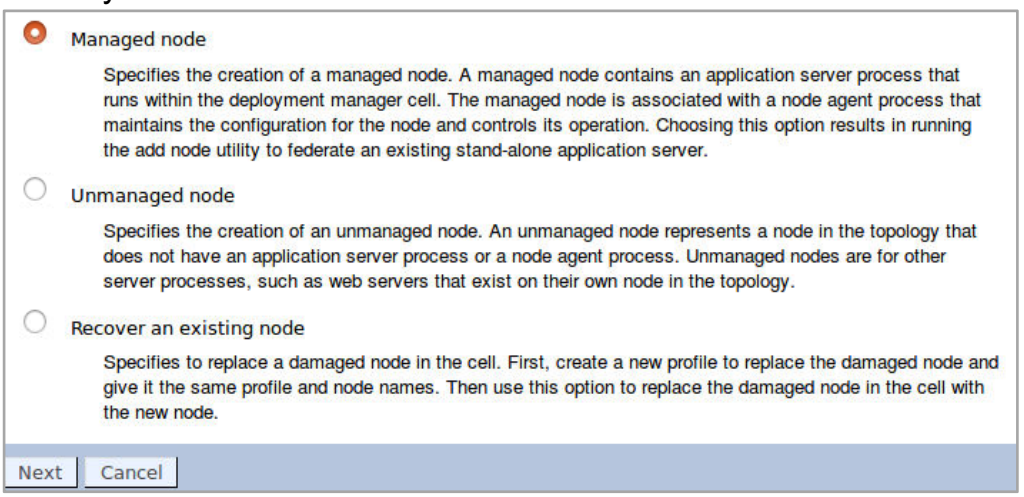


Figure 2-27. Managed versus unmanaged nodes

Nodes can be managed or unmanaged. Both application servers and supported web servers can be on unmanaged or managed nodes.

- **Managed nodes**
  - Node agent or administrative agent is used to manage its servers
  - Application server process runs within the deployment manager cell
- **Unmanaged nodes**
  - Node agent and administrative agent do not manage their servers
  - A stand-alone application server is an unmanaged node
  - Commonly used for web servers

IBM Training

Cell topology

- Cell topology can be viewed through the administrative console
  - From **System Administration > Cell > Local Topology**

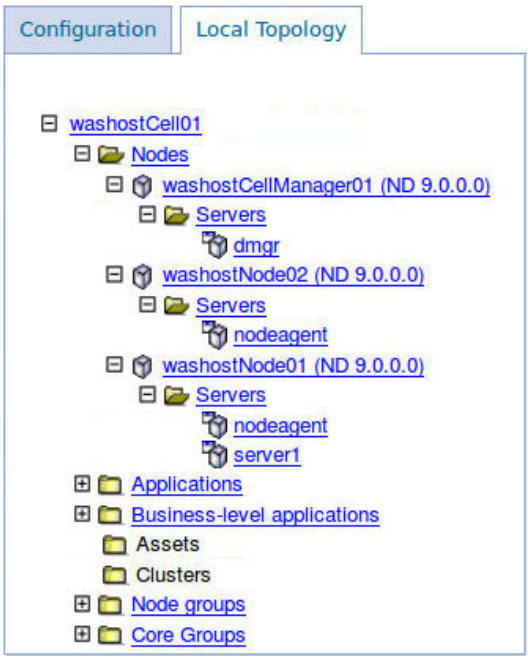


Figure 2-28. Cell topology

The deployment manager administrative console can be used to view the topology of a cell. The topology page is used to set the discovery protocol for an existing cell. A cell is a configuration concept, a way for an administrator to logically associate nodes according to whatever criteria makes sense in the administrator's organizational environment.

To view the cell topology, expand **System Administration** from the navigation tree. Click **Cell > Local Topology**.

IBM Training

Configuring synchronization

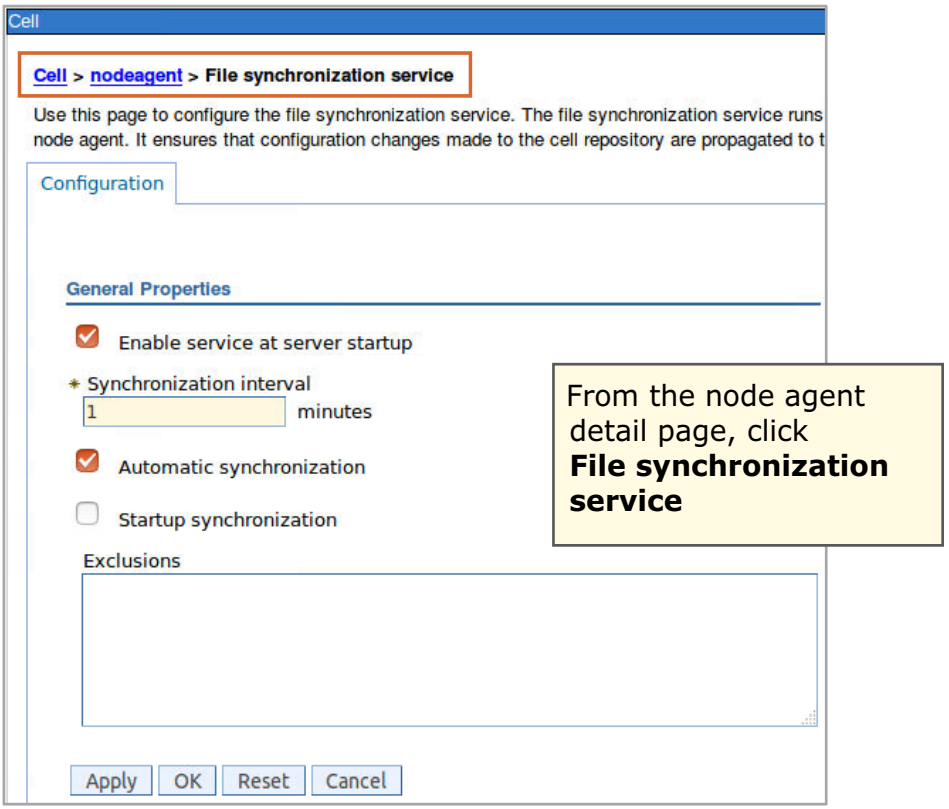


Figure 2-29. Configuring synchronization

The file synchronization service runs in the deployment manager and node agent. The service ensures that configuration changes made to the cell repository are propagated to the appropriate node repositories.

The deployment manager administrative console is used to configure the file synchronization service. Configuration is set for each node agent.

To configure file synchronization, expand **System administration** from the navigation tree. Click **Node agents** and click the appropriate node agent to set the configuration. Under **Additional Properties**, click **File synchronization service**.

The file synchronization page includes the following properties:

- **Enable service at server startup**
  Specifies whether the server attempts to start the file synchronization service. This setting does not cause a file synchronization operation to start. This setting is enabled by default.
- **Synchronization interval**
  Specifies the number of minutes that elapse between synchronizations. Increase the time interval to synchronize files less often. Decrease the time interval to synchronize files more often.

- **Automatic synchronization**
  Specifies whether to synchronize files automatically after a designated interval. When this setting is enabled, the node agent automatically contacts the deployment manager every synchronization interval to attempt to synchronize the configuration repository of the node with the master repository owned by the deployment manager.

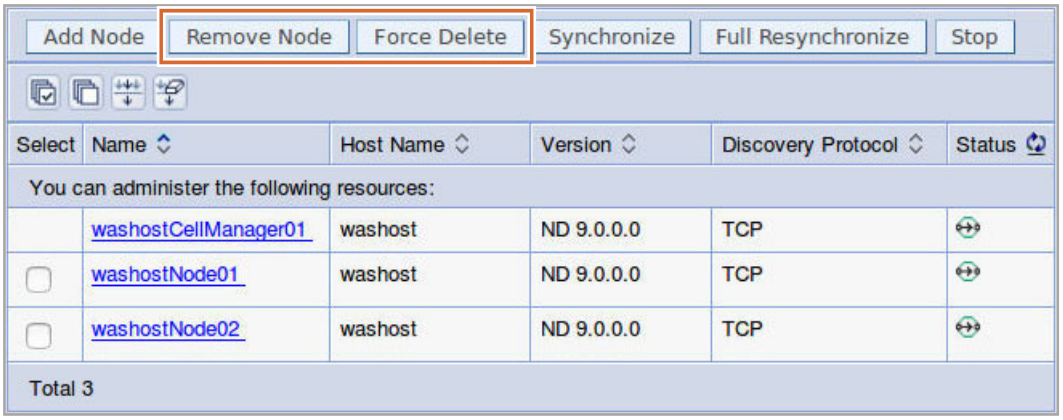
  If the automatic synchronization setting is enabled, the node agent attempts file synchronization when it establishes contact with the deployment manager. The node agent waits the synchronization interval before it attempts the next synchronization.
- **Startup synchronization**
  Specifies whether the node agent attempts to synchronize the node configuration with the latest configurations in the master repository before starting an application server.

  The default is to not synchronize files before starting an application server. Enabling the setting ensures that the node agent has the latest configuration but increases the amount of time it takes to start the application server.
- **Exclusions**
  Specifies files or patterns that should not be part of the synchronization of configuration data. Files in this list are not copied from the master configuration repository to the node, and are not deleted from the repository at the node.

IBM Training

Remove a node from a cell

- Use the `removeNode` command to remove a node from a cell
  - Restores stand-alone configuration of the node from a backup
  - The `removeNode` command is equivalent to using the **Remove Node** action
- Use the `cleanupNode` (Force Delete) command to force the removal of a node from a cell
  - Used to clean up a node that is defined in the cell configuration, but no longer exists
  - The `cleanupNode` command is equivalent to using the **Force Delete** action



Federating a cell

Figure 2-30. Remove a node from a cell

The `removeNode` command returns a node from a cell to a stand-alone application server installation. The `removeNode` command removes the node-specific configuration from the cell. When a node is removed from a cell, the profile reverts to the configuration it had before it was federated into a cell. Any applications or configuration changes that were made while it was part of a cell are lost.

- Restores stand-alone configuration of the node from a backup.
- The node can be removed through the deployment manager administrative console. Expand **System administration**. Click **Nodes**. Select the node to remove and click **Remove Node**.

The `cleanupNode` command cleans up a node configuration from the cell repository.

- The command is used to clean up a node that is defined in the cell configuration, but no longer exists.
- A node can be cleaned up through the deployment manager administrative console. Expand **System administration**. Click **Nodes**. Select the node to clean up and click **Force Delete**.

Synchronization

- **Synchronize**
  - Uses the normal synchronization optimization algorithm
  - Node and cell configuration might still be out of synchronization after operation

- **Full Resynchronize**
  - Clears all synchronization optimization settings
  - No mismatch between node and cell configuration

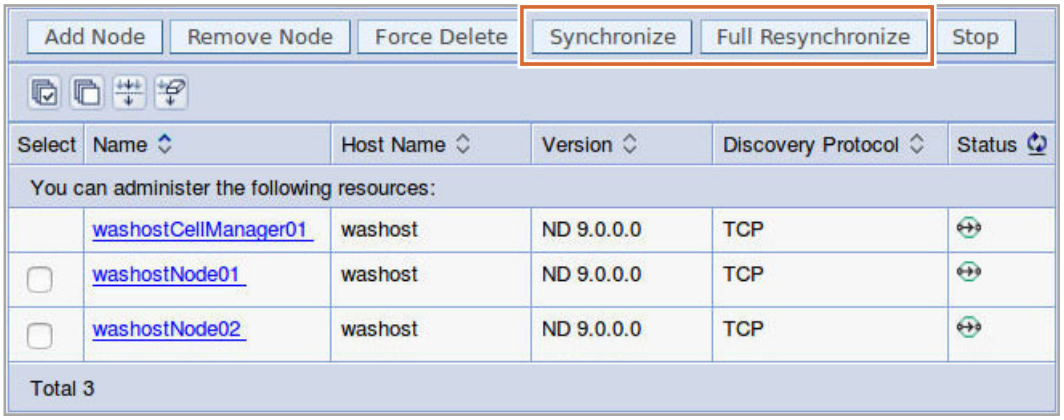


Figure 2-31. Synchronization

If you add a managed node or change a managed node configuration, synchronize the node configuration. Synchronization can be done by selecting the nodes and clicking **Synchronize** or **Full Resynchronize**.

Clicking either option sends a request to the node agent for that node to complete a configuration synchronization immediately, instead of waiting for the periodic synchronization to occur. This action is important if automatic configuration synchronization is disabled. It is also important if the synchronization interval is set to a long time, and a configuration change is made to the cell repository that must replicate to that node. Settings for automatic synchronization are on the File Synchronization Service page.

- **Synchronize**
  Synchronize requests that a node synchronization operation processes by using the normal synchronization optimization algorithm. This operation is fast, but might not fix problems from manual file edits that occur on the node. It is still possible for the node and cell configuration to be out of synchronization after this operation is done.
- **Full Resynchronize**
  Full Resynchronize clears all synchronization optimization settings and completes configuration synchronization anew, so there is no mismatch between node and cell configuration after this operation is done. This operation can take longer than the Synchronize operation.

- **syncNode command**
  The `syncNode` command forces a configuration synchronization to occur between the node and the deployment manager for the cell in which the node is configured.

  The node agent server runs a configuration synchronization service that maintains synchronization between the configuration the node and that of the master cell. If the node agent is unable to run because of a problem in the node configuration, you can use the `syncNode` command to complete a synchronization. When the node agent is not running, it can be used to force the node configuration back in sync with the cell configuration. If the node agent is running and you want to run the `syncNode` command, you must first stop the node agent.

IBM Training

Managing a web server: Adding a node to a cell

- Create an unmanaged node for defining remote web servers
  - From **System administration > Nodes > Add node**

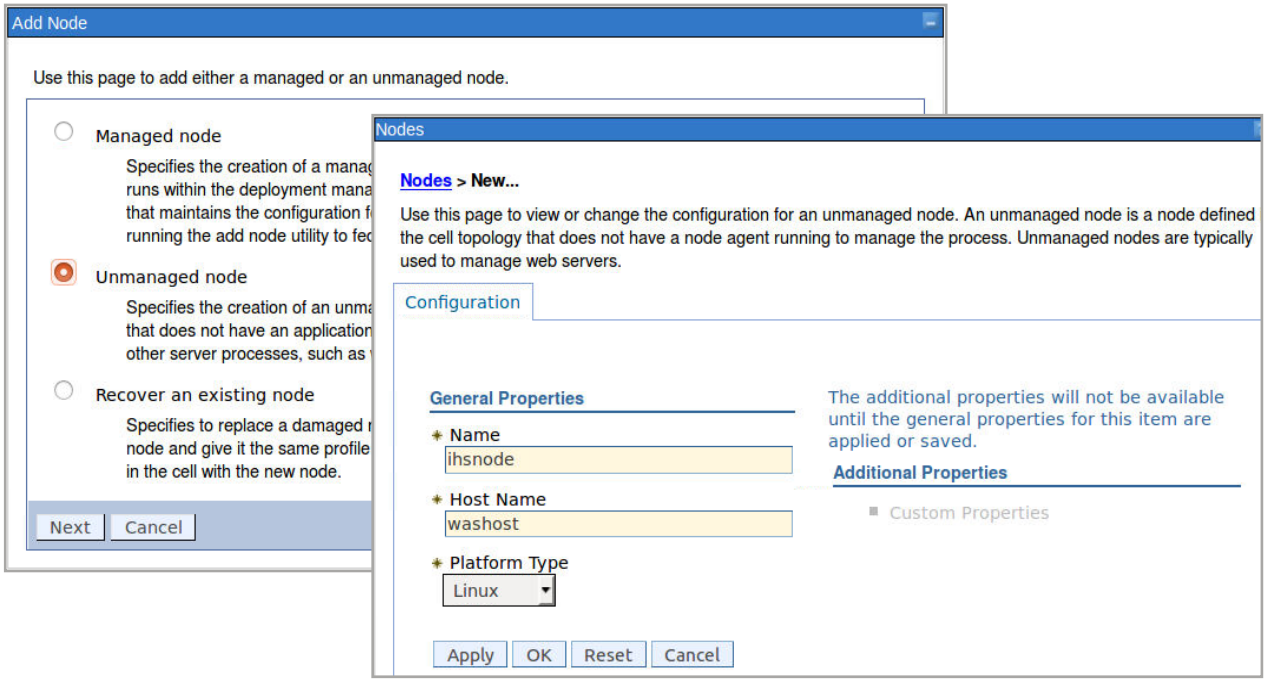


Figure 2-32. Managing a web server: Adding a node to a cell

An application server works with a web server to handle requests for dynamic content, such as servlets, from web applications. A web server uses a web server plug-in to establish and maintain persistent HTTP and HTTPS connections with an application server.

Before beginning work to manage the web server, make sure that the appropriate plug-in file is installed on your web server.

The first step to establish communication with a web server is to create a node to manage the web server. Within a cell topology, unmanaged nodes are typically used to manage web servers.

To create an unmanaged node:

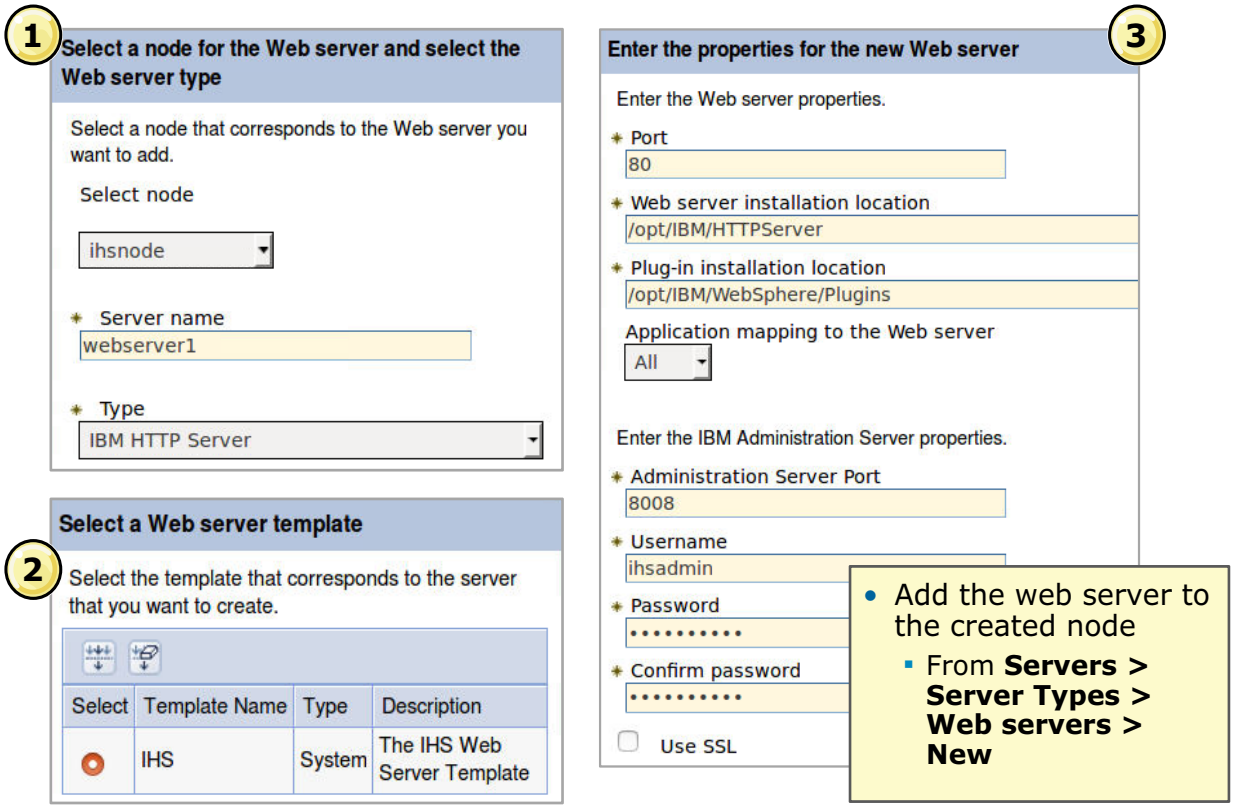
1. From the navigation tree, expand **System administration**. Click **Nodes > Add node**.
2. Click **Unmanaged node** and click **Next**.
3. From the new node page, provide the general properties for the unmanaged node:
   - **Name:**
     Specifies a logical name for the node. The name must be unique within the cell.
   - **Host name:**
     Specifies the host name of the unmanaged node that is added to the configuration.

- **Platform type:**
  Specifies the operating system on which the unmanaged node runs. Valid options are:
  - Windows
  - AIX
  - HP-UX
  - Solaris
  - Linux
  - OS/400
  - z/OS

4. Click **OK**.

IBM Training

Managing a web server: Add the web server



Federating a cell

Figure 2-33. Managing a web server: Add the web server

When a node is created that is used to manage the web server, a web server definition can be created. The deployment manager administrative console can be used to create the web server definition. The web server definition adds the web server to the newly created node.

To create a web server definition: from the navigation tree, expand **Servers > Server** Types. Click **Web servers > New**.

1. The first step is to identify the node and web server.
   - **Select node:**
     Identifies the node that manages the web server.
   - **Server name:**
     Server name is the logical name of the server. The name must be unique within the node.
   - **Type:**
     Type identifies the web server vendor type.
2. Select the web server template that you want to correspond to the web server you want to create. The template is based on the type of web server you chose previously.
3. Enter the properties for the new web server.

- **Port:**
  The port that the web server uses.
- **Web server installation location:**
  The location of the web server installation. It is required for IBM HTTP Server only.
- **Plug-in installation location:**
  The fully qualified path for the location of the plug-in configuration file.
- **Application mapping to the web server:**
  Web server application mapping. The options are:
  - **All**: All applications are automatically mapped to the web server.
  - **None**: No applications are automatically mapped to the web server.

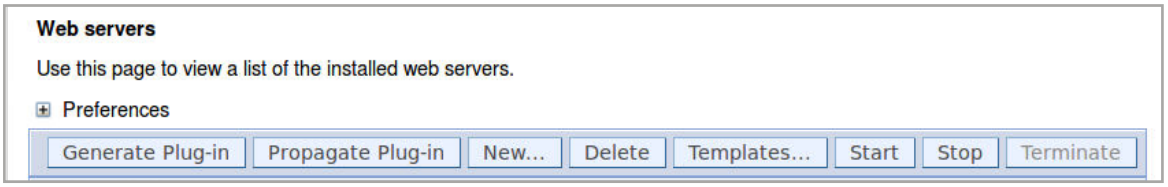
Enter the properties for the IBM Administration Server:

- **Administration Server Port**
- **Username**
- **Password**

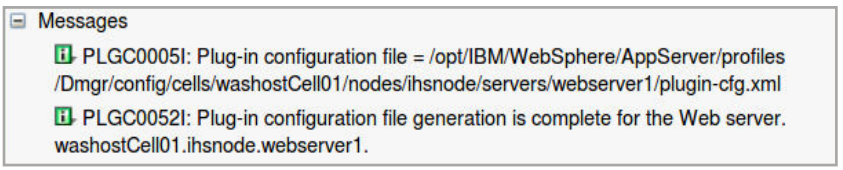
IBM Training

Managing a web server: Plug-in configuration file

- The plug-in configuration file contains routing for all applications that are mapped to the web server
- After changes that affect routing, regenerate and propagate the plug-in file to the web server
  - From **Servers > Server Types > Web servers**



1. Generate plug-in



2. Propagate plug-in

```
localuser@washost:/opt/IBM/WebSphere/AppServer/profiles/Dmgr/config/cells/washos
tCell01/nodes/ihsnode/servers/webserver1$ sudo cp plugin* /opt/IBM/WebSphere/Plu
gins/config/webserver1/
```

Federating a cell

Figure 2-34. Managing a web server: Plug-in configuration file

The plug-in configuration file contains routing for all applications that are mapped to the web server, passing HTTP requests from a web server to WebSphere Application Servers.

The plug-in is regenerated and propagated to the web server after changes that affect routing are made. The plug-in configuration file is automatically generated by default, whenever the web server environment changes, with a few exceptions. For example, the plug-in configuration file is regenerated whenever one of the following activities occurs:

- A new application is deployed on an associated application server.
- The web server definition is saved.
- An application is removed from an associated application server.
- A new virtual host is defined.

You can manually generate and propagate a plug-in configuration file for a web server from the deployment manager administrative console. From the navigation tree, expand **Servers > Server Types**. Click **Web servers**.

1. To generate the plug-in configuration file, select the web server whose plug-in configuration file must be generated. Click **Generate Plug-in**.

When the generation is complete, a message displays to indicate that the generation was successful. A message also indicates the directory location and name of the plug-in configuration file.

To propagate the plug-in configuration file, select the web server whose plug-in configuration file must be propagated. Click **Propagate Plug-in**.

When the propagation is complete, a message indicates the directory location and the name of the plug-in configuration file that is used for propagation. The message also indicates the directory location and name of the plug-in configuration file where the configuration is propagated to on the web server computer.

IBM Training

Unit summary

- Describe WebSphere Application Server cell concepts
- Describe and create the deployment manager profile
- Describe and create other profile types
- Describe custom profiles and automatic federation
- Describe the directories and configuration files for profiles
- Add a node by using commands or the administrative console
- Compare the deployment manager administrative console with the base administrative console
- Compare managed and unmanaged nodes
- Manage a web server by using the administrative console

Figure 2-35. Unit summary

Review questions

1. Which managed processes can be part of a cell?
   A. Deployment manager
   B. Node agent
   C. Load balancer
   D. Application server

2. Which profiles can be created by using the Profile Management Tool?
   A. Load balancer profile
   B. Custom profile
   C. Plug-in profile
   D. IBM HTTP Server profile

3. True or False: All application servers have a corresponding node agent.

Figure 2-36. Review questions

Write your answers here:

1.

2.

3.

IBM Training

Review answers

1. Which managed processes can be part of a cell?
   A. Deployment manager
   B. Node agent
   C. Load balancer
   D. Application server
   The answer is A, B, and D.
2. Which profiles can be created by using the Profile Management Tool?
   A. Load balancer profile
   B. Custom profile
   C. Plug-in profile
   D. IBM HTTP Server profile
   The answer is B.
3. True or False: All application servers have a corresponding node agent.
   The answer is False. A node can have multiple application servers.

Figure 2-37. Review answers

IBM Training

Exercise: Configuring the lab workstation

Figure 2-38. Exercise: Configuring the lab workstation

IBM Training

Exercise objectives

After completing this exercise, you should be able to:

- Configure the course lab workstation to start the exercises

Figure 2-39. Exercise objectives

IBM Training

Exercise: Creating a federated cell

Figure 2-40. Exercise: Creating a federated cell

IBM Training

Exercise objectives

After completing this exercise, you should be able to:

- Create a deployment manager profile
- Back up the deployment manager configuration
- Perform basic tasks by using the deployment manager administrative console
- Federate a node into the deployment manager cell
- Create a custom profile
- Create an unmanaged web server node
- Start and stop a web server by using the administrative console
- Map an application to a web server

Figure 2-41. Exercise objectives

Unit 3. Workload management

Estimated time

01:30

Overview

In this unit, you learn workload management principles and how WebSphere Application Server can be configured to participate in workload management.

How you will check your progress

- Review questions
- Lab exercises

References

WebSphere Application Server V9 Knowledge Center

https://www.ibm.com/support/knowledgecenter/SSEQTP_9.0.0/as_ditamaps/was900_welcome_base.html

IBM Training

Unit objectives

- Define workload management
- Create clusters and cluster members
- Compare clustered configurations
- Explain how weights are used in workload management
- Describe failover scenarios
- Describe the role of the HTTP plug-in in workload management
- Explain session management
- Configure distributed session management

Figure 3-1. Unit objectives

IBM Training

Topics

- Workload management concepts
- Clusters and cluster members
- Routing concepts and session affinity
- Failover
- Session persistence

Figure 3-2. Topics

3.1. Workload management concepts

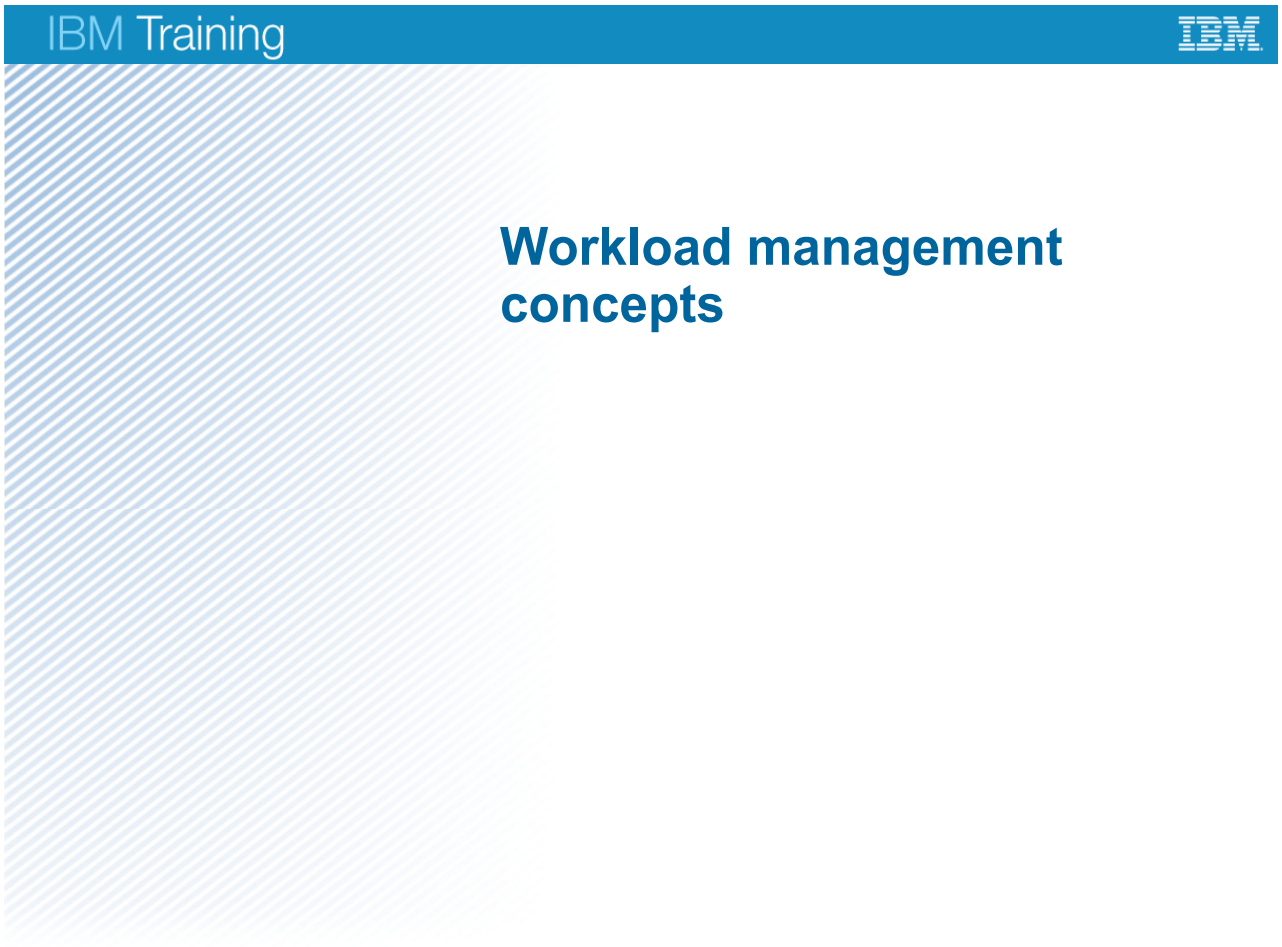


Figure 3-3. Workload management concepts

What is workload management (WLM)?

- Sharing requests across multiple application servers
- Configuration options that improve:
  - Performance: Improve response time for requests
  - Scalability: Grow capacity as the number of users increases
  - Load balancing: Allocate workload proportionately among available resources
  - Availability: If a server fails, applications are still available

Figure 3-4. What is workload management (WLM)?

There are numerous potential definitions for the term *workload management*. Within the context of the WebSphere Application Server, what is meant generically is to spread the work between different hosts. This feature can provide for better performance, scalability, load balancing, and availability.

IBM Training

What can be workload managed? (1 of 2)

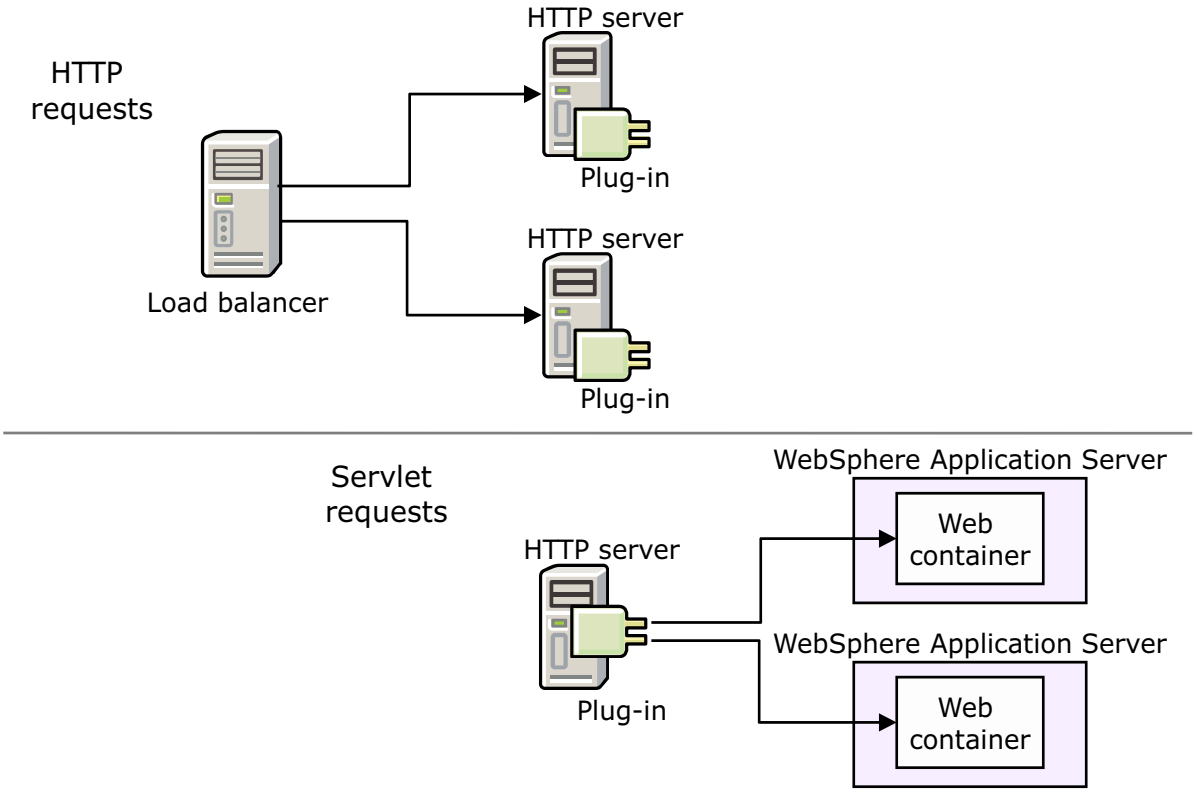


Figure 3-5. What can be workload managed? (1 of 2)

Within a standard application server topology, there are frequently three points where WLM occurs. The first point is where a load balancer spreads the work among numerous web servers. The second point is where the web server plug-in spreads work among numerous application servers, or more specifically, web containers.

What can be workload managed? (2 of 2)

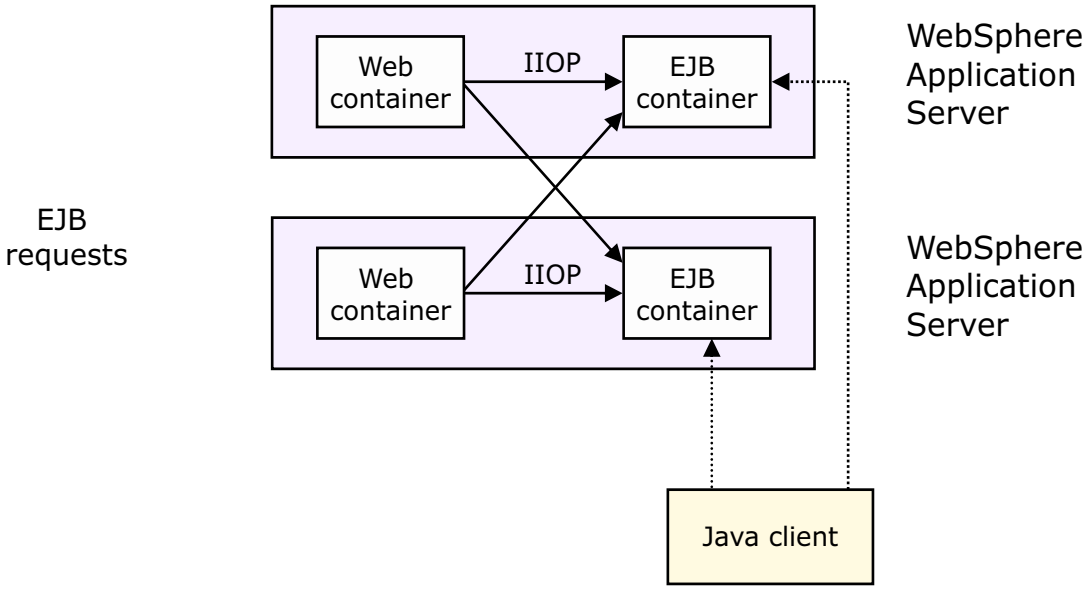


Figure 3-6. What can be workload managed? (2 of 2)

The third point where work can be spread is between the EJB clients and the EJB containers. This third point does not usually occur because the EJB client (in most cases, the web container) is typically in the same JVM as the EJB container. This topology forces all traffic to stay within the same JVM. The other cases are when a web container is placed in a different JVM and then the EJB container (not common), and when a stand-alone Java client is being used (again, not common).

3.2. Clusters and cluster members

IBM Training

Clusters and cluster members

Figure 3-7. Clusters and cluster members

IBM Training

Clusters

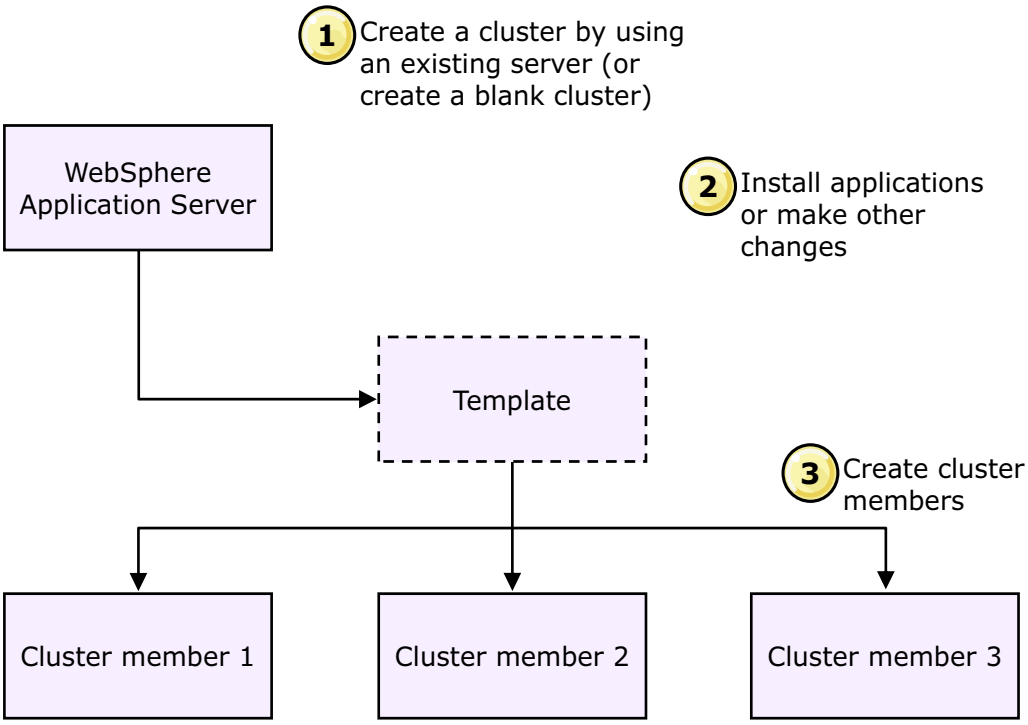


Figure 3-8. Clusters

Creating more throughput within a cell cannot be done by installing one application on multiple application servers. This situation can possibly create conflicts with the namespace and the plug-in. Clusters allow for the same application server to have multiple copies within a single cell. They provide a single point of management, automatic workload management, and failover.

IBM Training

Clusters and cluster members

- Clusters are a set of application servers that install the same applications, and are grouped logically for workload management
  - Applications that are installed to the cluster are automatically propagated to the cluster members
- Creation of a cluster
  - Can use an existing server to become the first cluster member
  - Other cluster members are created from templates
- Cluster members are similar to “clones” in previous WebSphere Application Server versions in that they:
  - Run the same applications
  - Share workload
  - Can be centrally administered

Figure 3-9. Clusters and cluster members

You can create a blank cluster and then install applications in it. It is also possible to have a cluster that is based on an existing application server. At any point, cluster members can be created within a cell. These cluster members are application servers, but they are associated with the cluster. At any point, then, more applications or configuration changes can be made to the cluster, and those changes are pushed out to all of the cluster members.

Configurations: Vertical scaling

- Might provide better performance with multiple processors
- Provides process level failover

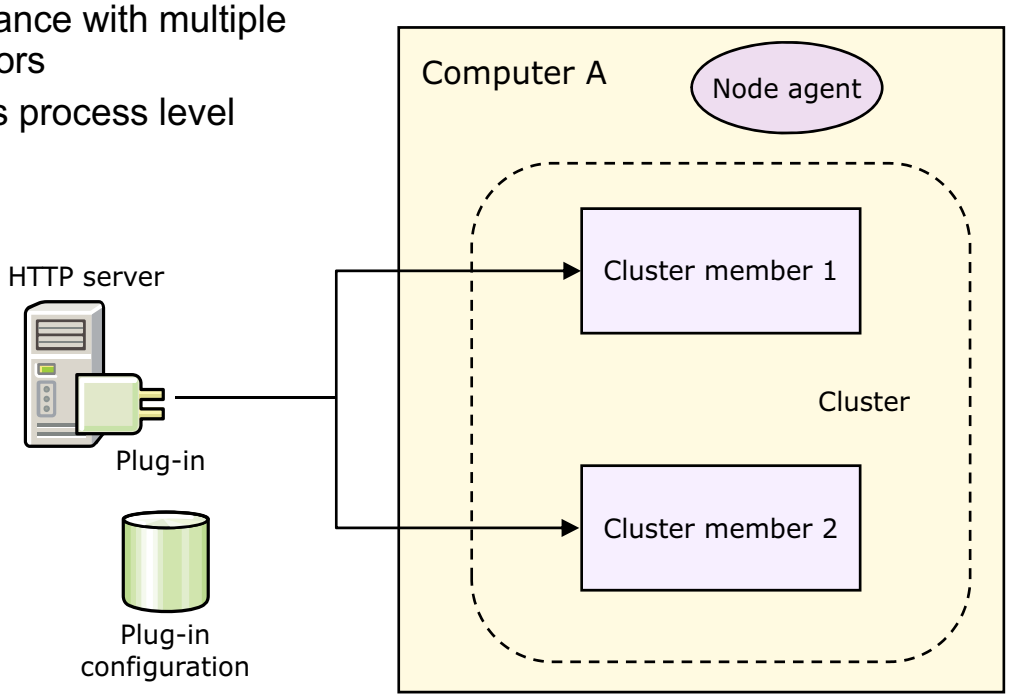


Figure 3-10. Configurations: Vertical scaling

Cluster members can be scaled vertically. You can put multiple cluster members on the same node (or computer). Although this practice might not increase throughput (assuming that one cluster member uses the computer fully), it does provide process level failover.

IBM Training

Configurations: Horizontal scaling

- Supports server failover

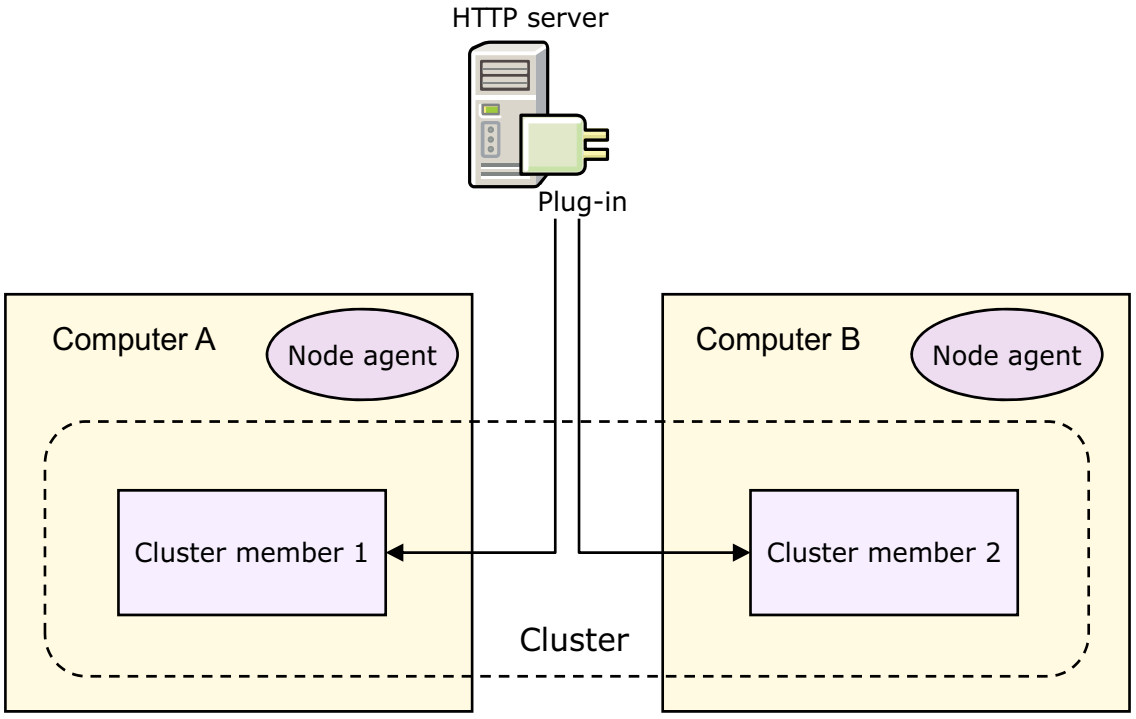


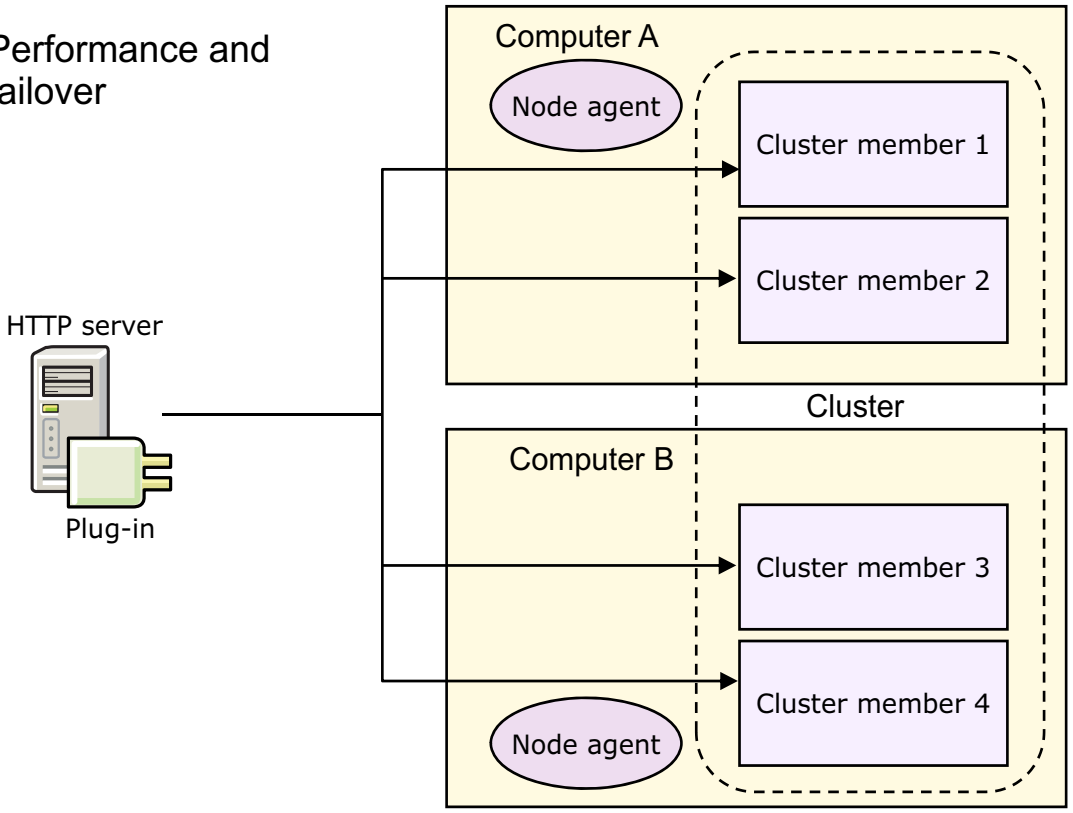
Figure 3-11. Configurations: Horizontal scaling

Cluster members can be scaled horizontally. You can put cluster members on different computers. This method not only increases throughput but also provides system level failover.

IBM Training

Configurations: Vertical and horizontal scaling

- Performance and failover



Workload management

Figure 3-12. Configurations: Vertical and horizontal scaling

Scaling both vertically and horizontally combines both system and process level failover.

Creating a cluster (1 of 4)

- In the console, select **Servers > Clusters > WebSphere application server clusters** and click **New**
- Enter cluster name
- Check options
  - Prefer local
  - Configure HTTP session memory-to-memory replication
- Click **Next**

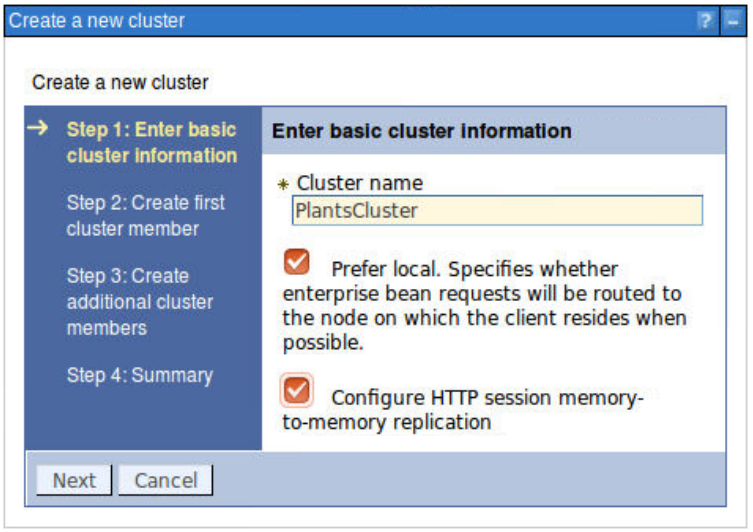


Figure 3-13. Creating a cluster (1 of 4)

This diagram shows the first step of the cluster creation wizard.

IBM Training

Creating a cluster (2 of 4)

- Create the first cluster member that is based on:
  - A server template
  - An existing server
  - Converting an existing server
  - An empty cluster
- Enter member name
- Select node
- Enter weight
- Click **Next**

Figure 3-14. Creating a cluster (2 of 4)

When creating a cluster, several options must be specified. These options include what to base the new cluster on (a template, an existing application server, converting an existing application server) and things like the cluster member name, the weight, the node, and other items.

IBM Training

Creating a cluster (3 of 4)

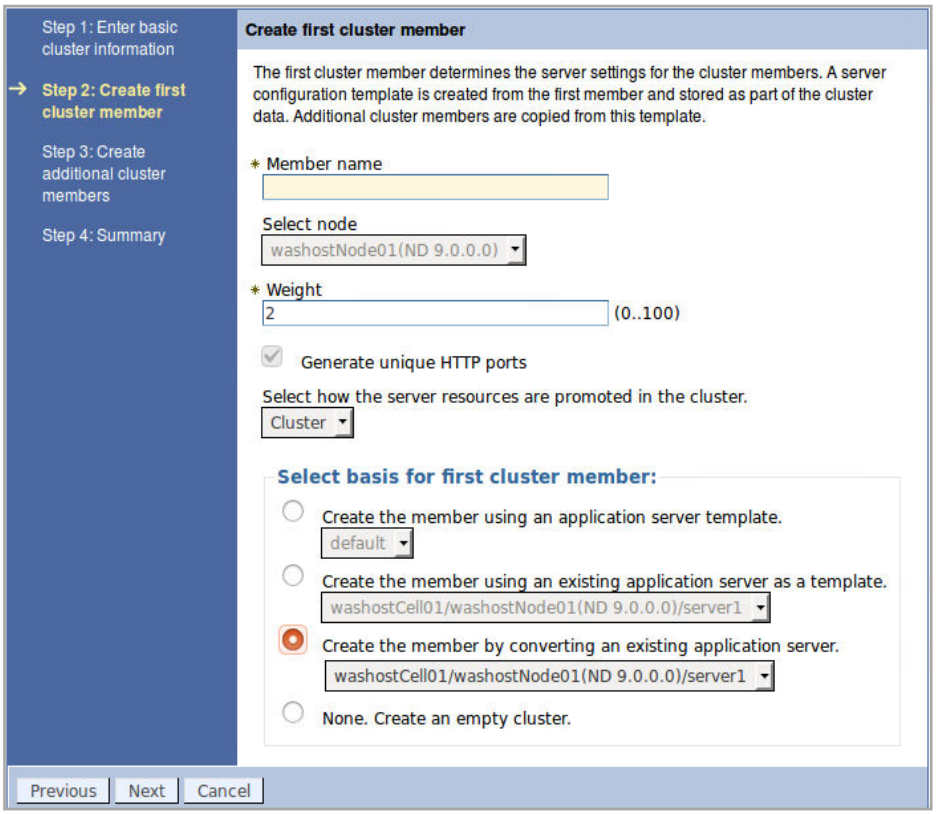


Figure 3-15. Creating a cluster (3 of 4)

In this example, the option to create the first cluster member that is based on an existing server is selected.

IBM Training

Creating a cluster (4 of 4)

1. Enter member name
2. Select node
3. Set weight
4. Check options:
   - Generate unique HTTP ports (default)
5. Click **Add Member**
6. Repeat to create other cluster members
7. Click **Next**
8. Click **Finish** on Summary screen

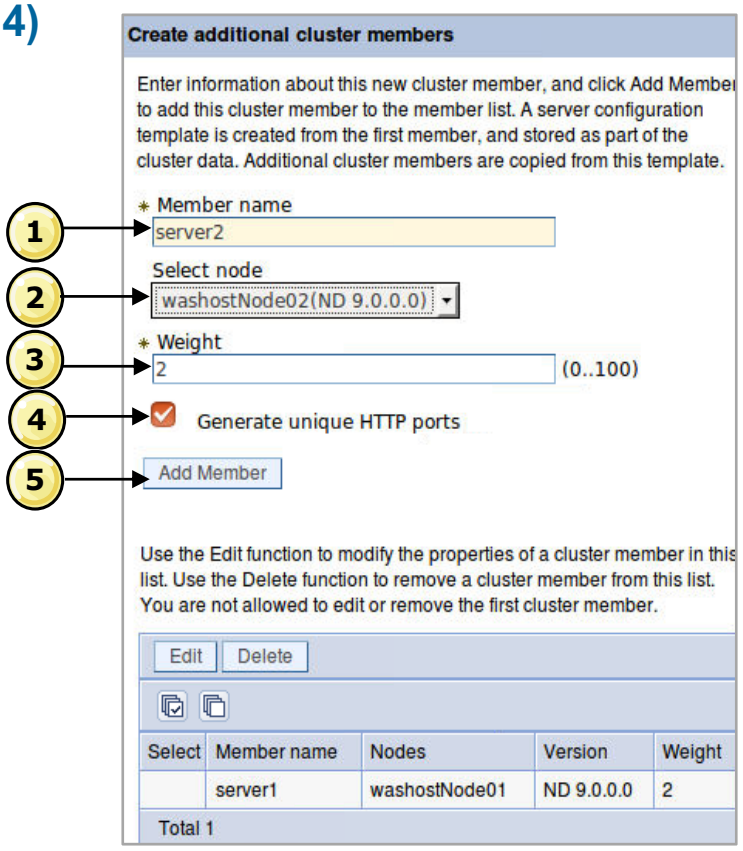


Figure 3-16. Creating a cluster (4 of 4)

After the first cluster member is created, you can create more members. This screen shows the options for creating more cluster members.

IBM Training

Installing enterprise applications to a cluster

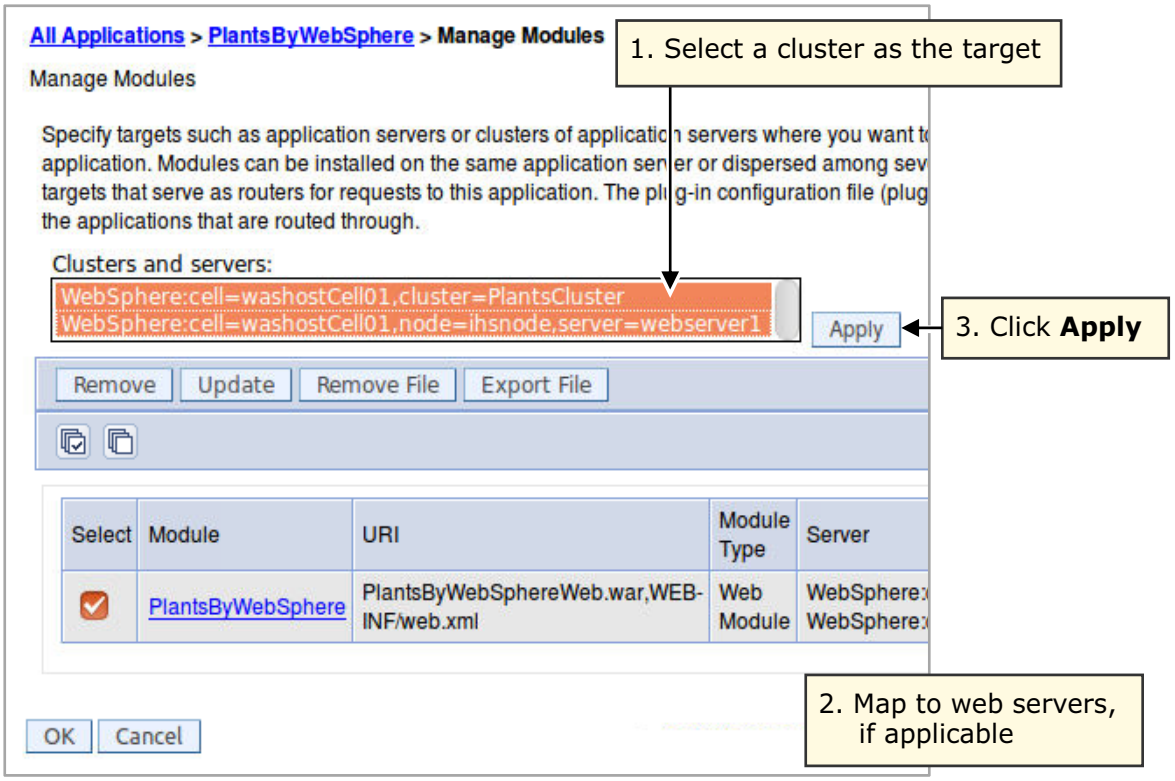


Figure 3-17. Installing enterprise applications to a cluster

This procedure follows the same steps as installing to a base server except:

1. You select a cluster as the target, rather than a server.
2. You possibly map to web servers.

IBM Training

Controlling a cluster

- Start the cluster
  - Start starts all servers together
  - Ripplestart starts servers one at a time
- Status
  - Started: All servers are started
  - Partial Start: Some servers are started
  - Stopped: All the servers are stopped
  - Partial Stop: Some servers are stopped

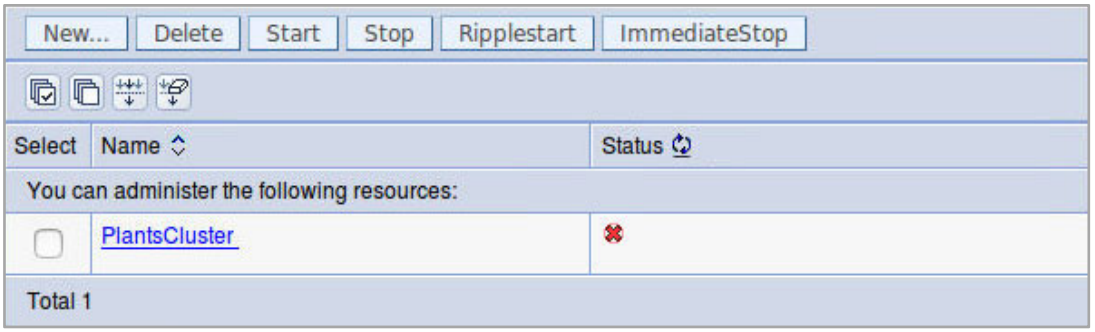


Figure 3-18. Controlling a cluster

If ripplestart is used on a running cluster, each application server in the cluster is stopped one at a time, and restarted.

The screen capture shows a partially stopped cluster.

Cluster members

- Clusters can also be started (or stopped) by merely starting all application servers that are members
- Cluster members are just application servers

Figure 3-19. Cluster members

Cluster members can be started individually by starting the application servers.

IBM Training

Modification of clusters

- Use the administrative console or wsadmin
- What can be changed in a cluster?
  - Cluster member settings: Weights, prefer local
  - Install or update applications
  - Application server settings: Cluster members are application servers and the normal application server settings can be modified
- After you modify the cluster:
  - Save the configuration and resynchronize
  - Regenerate the HTTP server plug-in and redistribute it if necessary

Figure 3-20. Modification of clusters

Individual cluster member settings include Weight and Prefer local. Applications can be installed or updated on a cluster. Changes to application server settings must be made to each cluster member individually.

3.3. Routing concepts and session affinity

IBM Training

Routing concepts and session affinity

Figure 3-21. Routing concepts and session affinity

IBM Training

Basic routing algorithms

Routing decision points

1. Load balancer
2. HTTP Server plug-in
3. WLM-aware EJB client
   - Includes web container, Java client, EJB

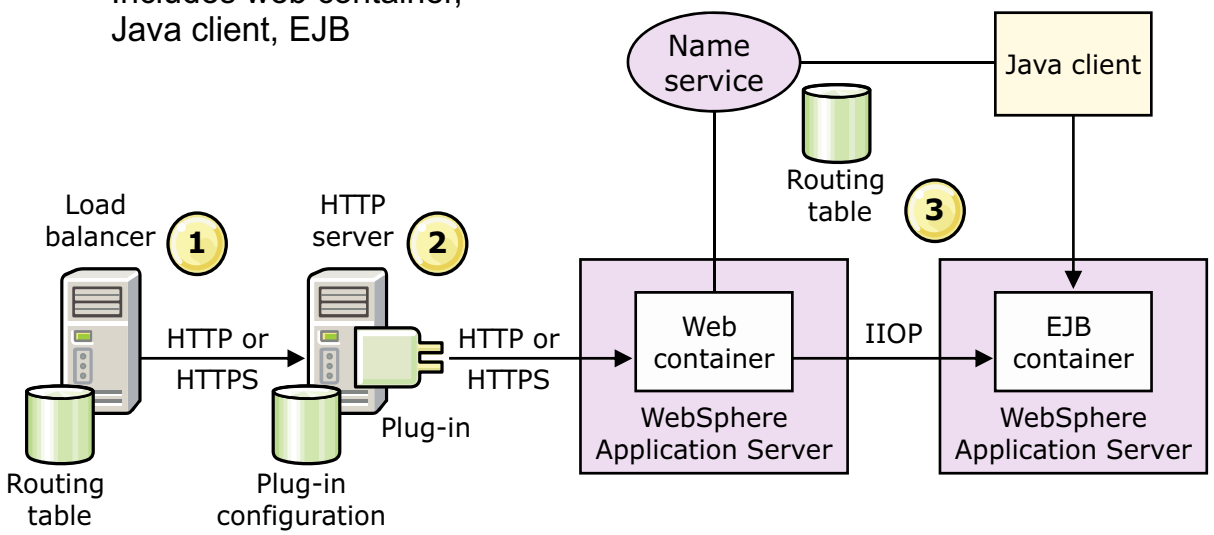


Figure 3-22. Basic routing algorithms

1. Load balancer:
   - The routing decision table is stored internally
   - Configured by using the Load Balancer tool
   - Multiple intelligent routing options
2. HTTP server plug-in:
   - The `plugin-cfg.xml` file defines routing
   - Configured by using administrative console, or wsadmin
   - Default is weighted round robin
3. WLM-aware EJB client:
   - Includes web container, Java client, EJB
   - Name service then supplies routing table
   - Configured by using administrative console, or wsadmin
   - Options:
     - Prefer local: yes or no

IBM Training

HTTP session management

- HTTP is a stateless protocol
- You can use sessions to maintain state information across a series of HTTP requests from the same client
  - For example, maintain shopping cart until check-out

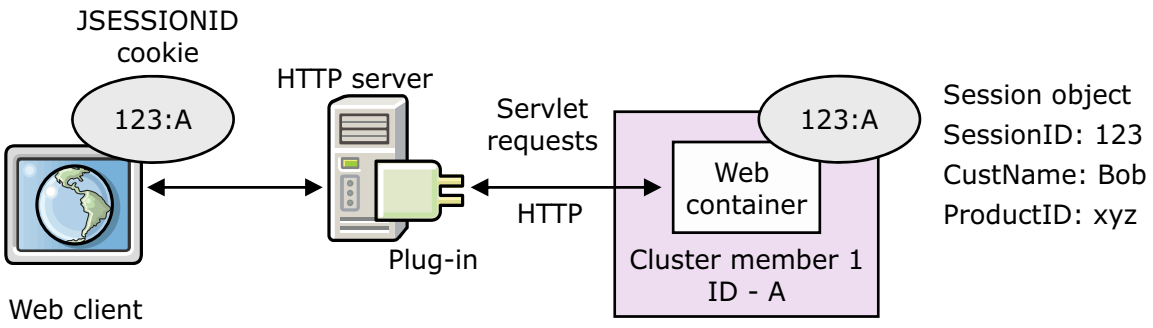


Figure 3-23. HTTP session management

HTTP is a stateless protocol.

You can use sessions to maintain state information across a series of HTTP requests from the same client (for example, maintaining your shopping cart until check out).

The Java servlet specification defines the session management process for web applications.

The session manager stores session information, and sends the client a unique clone ID and session ID through:

- A cookie in the HTTP header, or
- URL rewriting in a parameter on links or forms

IBM Training

Session affinity

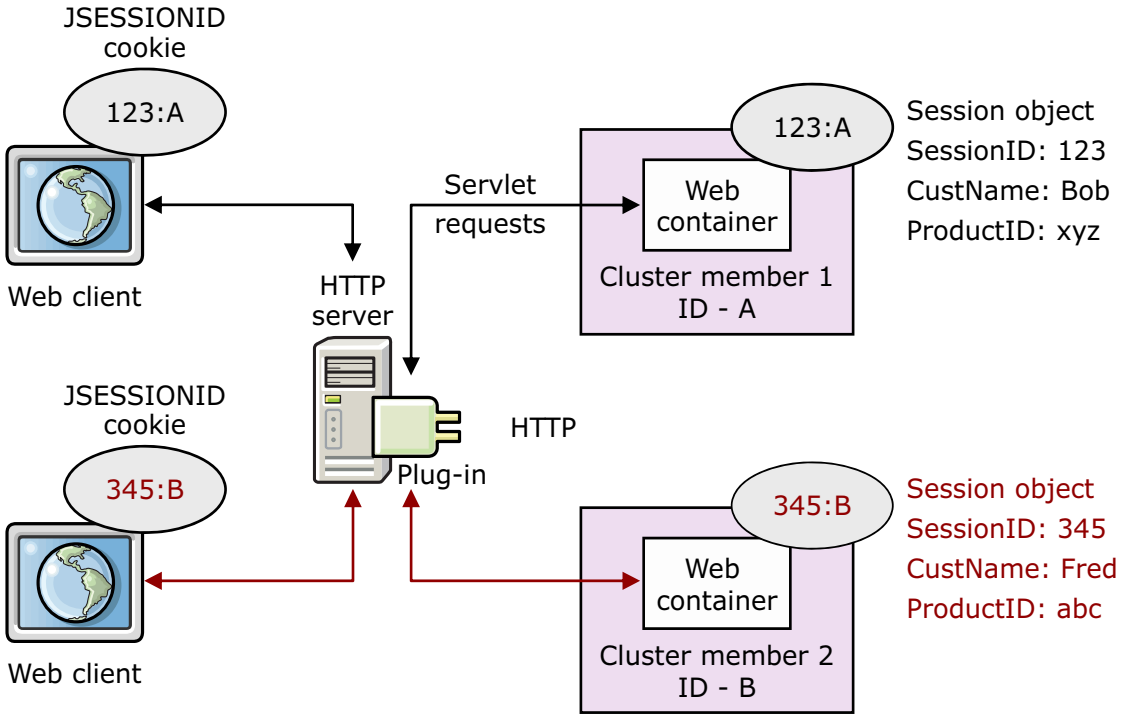


Figure 3-24. Session affinity

An application can retain state information for a user's session in memory. Other servers in the cluster might not have this information.

The HTTP server plug-in routes subsequent servlet requests consistently to the same application server after the session is created, by using a clone ID passed with the session ID in a cookie or URL.

JSESSIONID cookie

- The JSESSIONID cookie is used to help manage sessions
  - The plug-in uses the data to find which clone has affinity
  - The web container uses it to find the right http session object

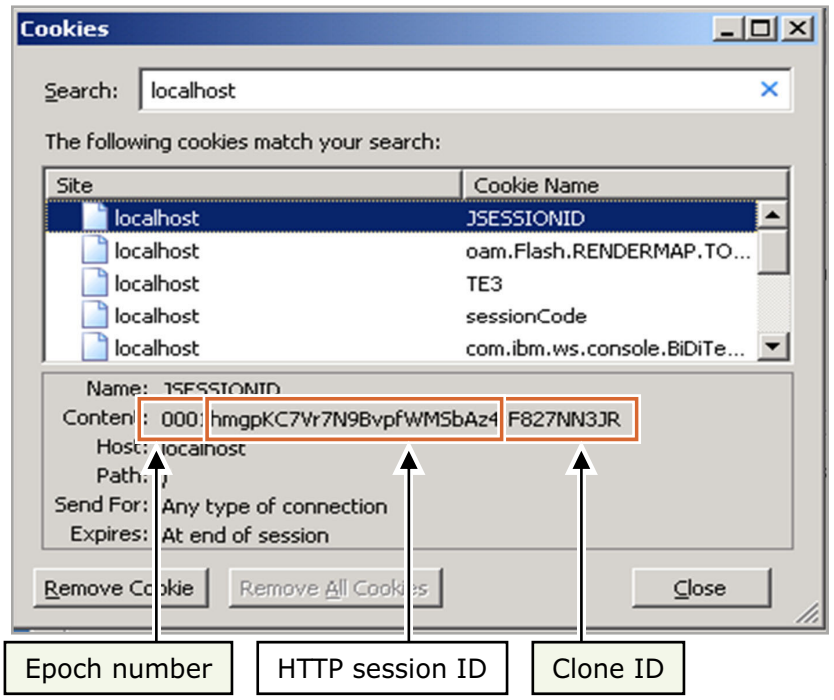


Figure 3-25. JSESSIONID cookie

The JSESSIONID cookie includes three main parts:

- The HTTP session ID
- The clone ID
- The epoch number

The clone ID allows the plug-ins to identify which cluster member (or clone) holds the session object. The HTTP session object allows the application server to find the object that is specific to that particular request. And finally, the epoch number allows the web container to make sure that the cached HTTP session object is not stale.

The format of the cookie can change in certain cases. For example, if memory-to-memory session replication is used, the format is entirely different. Also, if the session is failed over to another web container, there can be a list of clone IDs instead of just one.

IBM Training

WebSphere session affinity

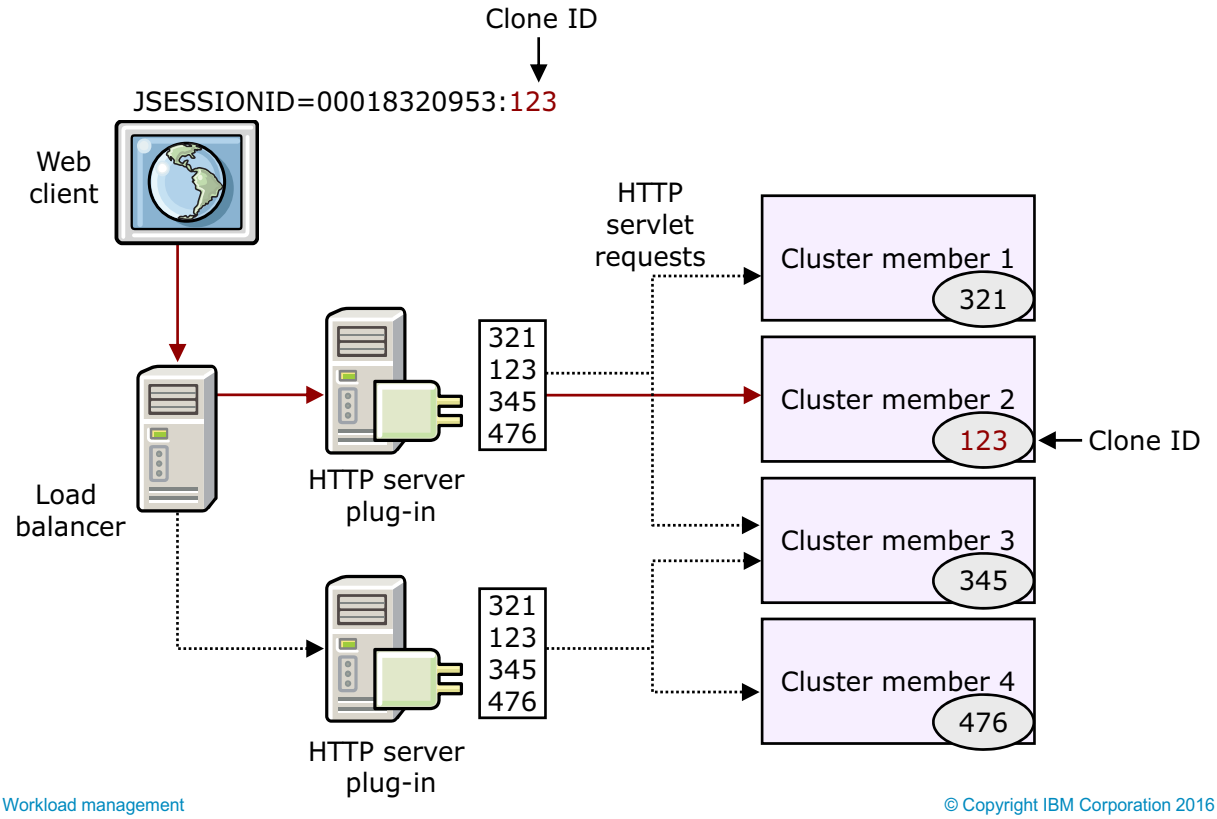


Figure 3-26. WebSphere session affinity

WebSphere session affinity has the following features:

- Session cookie is generated
- Uses clone ID to allow affinity on first hit
- Plug-ins know how to route to correct clone ID
- Can be turned off by removing clone IDs from plug-in file
- Format for the JSESSIONID cookie changes when using memory-to-memory replication

Plug-in

- Based on the data in the `plugin-cfg.xml` file, the plug-in is able to route sticky requests to the correct application server
  - If the application server is unavailable, the request is rerouted to the next application server
  - This new application server now has affinity
  - The session information that is held within the web container does not fail over unless session failover is configured
  - An unavailable application server is marked as down for a certain amount of time (default is 2 minutes)
  - After that time, the server is tried again
  - The `plugin-cfg.xml` file is checked for updates every 60 seconds so that new application servers are automatically brought into the active list

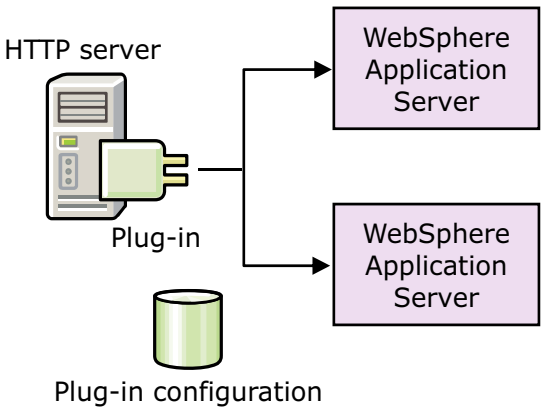


Figure 3-27. Plug-in

Based on the data in the `plugin-cfg.xml` file, the plug-in is able to route sticky requests to the correct application server. If the application server is unavailable, the request is rerouted to the next application server; this new application server now has affinity. The session information that is held within the web container does not fail over unless session failover is configured. An unavailable application server is marked as down for a certain amount of time (default is 2 minutes); after that time, the server is tried again. The `plugin-cfg.xml` file is checked for updates every 60 seconds; so new application servers are automatically brought into the active list.

IBM Training

Plugin-cfg.xml (1 of 3)

```
<?xml version="1.0" encoding="ISO-8859-1"?><!--HTTP server plugin config file for the webserver was7host01Cell01.ihsnode.webserver01 generated
on 2009.02.19 at 01:08:34 AM EST-->
<Config ASDisableNagle="false" AcceptAllContent="false" AppServerPortPreference="WebserverPort" ChunkedResponse="false"
FIPSEnable="false" IISDisableNagle="false" IISPluginPriority="High" IgnoreDNSFailures="false" RefreshInterval="60" ResponseChunkSize="64"
VHostMatchingCompat="false">
  <Log LogLevel="Error" Name="c:\Program Files\IBM\HTTPServer\Plugins\logs\webserver01\http_plugin.log"/>
  <Property Name="ESIEnable" Value="true"/>
  <Property Name="ESIMaxCacheSize" Value="1024"/>
  <Property Name="ESIInvalidationMonitor" Value="false"/>
  <Property Name="ESIEnableToPassCookies" Value="false"/>
  <Property Name="PluginInstallRoot" Value="c:\Program Files\IBM\HTTPServer\Plugins\"/>
  <VirtualHostGroup Name="default_host">
    <VirtualHost Name="*:9080"/>
    <VirtualHost Name="*:80"/>
    <VirtualHost Name="*:9443"/>
    <VirtualHost Name="*:5060"/>
    <VirtualHost Name="*:5061"/>
    <VirtualHost Name="*:443"/>
    <VirtualHost Name="*:9081"/>
    <VirtualHost Name="*:9444"/>
  </VirtualHostGroup>
  <ServerCluster CloneSeparatorChange="false" GetDWLMTable="false" IgnoreAffinityRequests="true" LoadBalance="Round Robin"
Name="TradeCluster" PostBufferSize="64" PostSizeLimit="-1" RemoveSpecialHeaders="true" RetryInterval="60">
    <Server CloneID="13u6hqmf8" onnectTimeout="0" ExtendedHandshake="false" LoadBalanceWeight="2" MaxConnections="-1"
Name="was7host01Node01_server1" ServerIOTimeout="0" WaitForContinue="false">
      <Transport Hostname="was7host01" Port="9080" Protocol="http"/>
      <Transport Hostname="was7host01" Port="9443" Protocol="https">
        <Property Name="keyring" Value="c:\Program Files\IBM\HTTPServer\Plugins\config\webserver01\plugin-key.kdb"/>
        <Property Name="stashfile" Value="c:\Program Files\IBM\HTTPServer\Plugins\config\webserver01\plugin-key.sth"/>
      </Transport>
    </Server>
```


Figure 3-28. Plugin-cfg.xml (1 of 3)

This slide shows the contents of a `plugin-cfg.xml` file where the CloneID is highlighted.

IBM Training

Plugin-cfg.xml (2 of 3)

```
    <Server CloneID="13u6hr5pv" ConnectTimeout="0" ExtendedHandshake="false" LoadBalanceWeight="2" MaxConnections="-1"
Name="was7host01Node02_server2" ServerIOTimeout="0" WaitForContinue="false">
      <Transport Hostname="was7host01" Port="9081" Protocol="http"/>
      <Transport Hostname="was7host01" Port="9444" Protocol="https">
        <Property Name="keyring" Value="c:\Program Files\IBM\HTTPServer\Plugins\config\webserver01\plugin-key.kdb"/>
        <Property Name="stashfile" Value="c:\Program Files\IBM\HTTPServer\Plugins\config\webserver01\plugin-key.sth"/>
      </Transport>
    </Server>
        <Property Name="stashfile" Value="c:\Program Files\IBM\HTTPServer\Plugins\config\webserver01\plugin-key.sth"/>
      </Transport>
    </Server>
    <PrimaryServers>
      <Server Name="was7host01Node01_server1"/>
      <Server Name="was7host01Node02_server2"/>
    </PrimaryServers>
  </ServerCluster>
  <UriGroup Name="default_host_TradeCluster_URIs">
    <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid" Name="/ivt/*"/>
    <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid" Name="/snoop/*"/>
    <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid" Name="/hello"/>
    <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid" Name="/hitcount"/>
    <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid" Name="*.jsp"/>
    <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid" Name="*.jsv"/>
    <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid" Name="*.jsw"/>
    <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid" Name="/j_security_check"/>
    <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid" Name="/ibm_security_logout"/>
    <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid" Name="/servlet/*"/>
    <Uri AffinityCookie="JSESSIONID" AffinityURLIdentifier="jsessionid" Name="/Trade/web/*"/>
  </UriGroup>
```


Figure 3-29. Plugin-cfg.xml (2 of 3)

IBM Training

Plugin-cfg.xml (3 of 3)

```
  <Route ServerCluster="TradeCluster" UriGroup="default_host_TradeCluster_URIs" VirtualHostGroup="default_host"/>
  <RequestMetrics armEnabled="false" loggingEnabled="false" rmEnabled="false" traceLevel="HOPS">
    <filters enable="false" type="URI">
      <filterValues enable="false" value="/snoop"/>
      <filterValues enable="false" value="/hitcount"/>
    </filters>
    <filters enable="false" type="SOURCE_IP">
      <filterValues enable="false" value="255.255.255.255"/>
      <filterValues enable="false" value="254.254.254.254"/>
    </filters>
    <filters enable="false" type="JMS">
      <filterValues enable="false" value="destination=aaa"/>
    </filters>
    <filters enable="false" type="WEB_SERVICES">
      <filterValues enable="false" value="wsdlPort=aaa:op=bbb:nameSpace=ccc"/>
    </filters>
  </RequestMetrics>
</Config>
```


Figure 3-30. Plugin-cfg.xml (3 of 3)

The slide shows the portion of the contents of a `plugin-cfg.xml` file that contains JSESSIONID cookie information.

IBM Training

Interpreting the plugin-cfg.xml file (1 of 4)

Find a UriGroup that has a regular expression that matches the request

```
<Config ...>
   ...
   <UriGroup Name="default_host_PlantsCluster_URIs">
      ...
      <Uri Name="/PlantsByWebSphere/*" .../>
   ...
   <Route UriGroup="default_host_PlantsCluster_URIs“
          ServerCluster="PlantsCluster"/>
   ...
   <ServerCluster Name="PlantsCluster" LoadBalance="Round Robin" ... >
      <Server CloneID="17shqbbrq" LoadBalanceWeight="2" ...>
         <Transport Hostname="washost" Port="9080" Protocol="http"/>
         <Transport Hostname="washost" Port="9443" Protocol="https">
            ...
      <Server CloneID="17shqbcf9" LoadBalanceWeight="2" ...>
         <Transport Hostname="washost" Port="9081" Protocol="http"/>
         <Transport Hostname="washost" Port="9445" Protocol="https">
```


Figure 3-31. Interpreting the plugin-cfg.xml file (1 of 4)

Application requests are mapped to a UriGroup by matching one of the UriGroup’s regular expressions.

IBM Training

Interpreting the plugin-cfg.xml file (2 of 4)

Find the Route element that maps to the UriGroup name

```
<Config ...>
   ...
   <UriGroup Name="default_host_PlantsCluster_URIs">
      ...
      <Uri Name="/PlantsByWebSphere/*" .../>
   ...
   <Route UriGroup="default_host_PlantsCluster_URIs"
          ServerCluster="PlantsCluster"/>
   ...
   <ServerCluster Name="PlantsCluster" LoadBalance="Round Robin" ... >
      <Server CloneID="17shqbbrq" LoadBalanceWeight="2" ...>
         <Transport Hostname="washost" Port="9080" Protocol="http"/>
         <Transport Hostname="washost" Port="9443" Protocol="https">
            ...
      <Server CloneID="17shqbcf9" LoadBalanceWeight="2" ...>
         <Transport Hostname="washost" Port="9081" Protocol="http"/>
         <Transport Hostname="washost" Port="9445" Protocol="https">
```


Figure 3-32. Interpreting the plugin-cfg.xml file (2 of 4)

All of the application requests that are mapped to a UriGroup are routed based on the information in a Route XML element that corresponds to the UriGroup.

IBM Training

Interpreting the plugin-cfg.xml file (3 of 4)

Find the ServerCluster element that maps to the Route's ServerCluster

```
<Config ...>
   ...
   <UriGroup Name="default_host_PlantsCluster_URIs">
      ...
      <Uri Name="/PlantsByWebSphere/*" .../>
   ...
   <Route UriGroup="default_host_PlantsCluster_URIs"
          ServerCluster="PlantsCluster"/>
   ...
   <ServerCluster Name="PlantsCluster" LoadBalance="Round Robin" >
      <Server CloneID="17shqbbrq" LoadBalanceWeight="2" ...>
         <Transport Hostname="washost" Port="9080" Protocol="http"/>
         <Transport Hostname="washost" Port="9443" Protocol="https">
            ...
      <Server CloneID="17shqbcf9" LoadBalanceWeight="2" ...>
         <Transport Hostname="washost" Port="9081" Protocol="http"/>
         <Transport Hostname="washost" Port="9445" Protocol="https">
```


Figure 3-33. Interpreting the plugin-cfg.xml file (3 of 4)

The Route XML element determines which ServerCluster element to use for selecting cluster members to forward the request to.

IBM Training

Interpreting the plugin-cfg.xml file (4 of 4)

Server selected based on algorithm, affinity, and protocol

```
...
   <UriGroup Name="default_host_PlantsCluster_URIs">
      ...
      <Uri Name="/PlantsByWebSphere/*" .../>
   </UriGroup>
   ...
   <Route UriGroup="default_host_PlantsCluster_URIs“
          ServerCluster="PlantsCluster"/>
   ...
   <ServerCluster Name="PlantsCluster" LoadBalance="Round Robin" …
      <Server CloneID="17shqbbrq" LoadBalanceWeight="2" ...>
         <Transport Hostname="washost" Port="9080" Protocol="http"/>
         <Transport Hostname="washost" Port="9443" Protocol="https">
            ...
      <Server CloneID="17shqbcf9" LoadBalanceWeight="2" ...>
         <Transport Hostname="washost" Port="9081" Protocol="http"/>
         <Transport Hostname="washost" Port="9444" Protocol="https">
```


Figure 3-34. Interpreting the plugin-cfg.xml file (4 of 4)

After a ServerCluster XML element is determined, a Server in the ServerCluster is chosen based on algorithm, affinity, and protocol.

Weighted round robin

- The plug-in and EJB routing both use weighted round robin by default
  - The plug-in can be configured to use random instead
- Initial weights are given to each cluster member
  - The weights default to a value of 2
  - The maximum weight that is allowed in the console is 20
- An internal routing table is set up for each plug-in
  - Using a greatest common divisor (GCD), the plug-in attempts to reduce the weight values for each cluster member

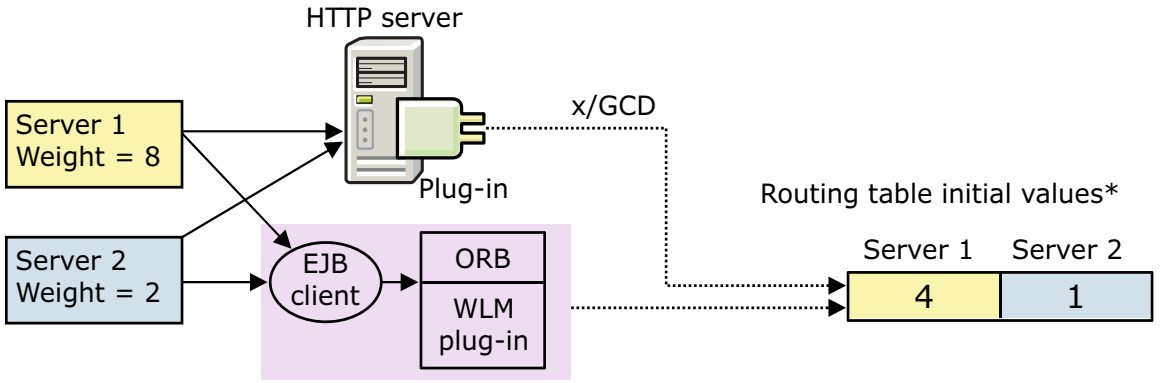


Figure 3-35. Weighted round robin

The default routing option for both the plug-in and the ORB is called weighted round robin. Each server is assigned a weight that is then reduced by dividing all values by the GCD (greatest common divisor). These new values are then inserted into a table (for both the plug-in and the ORB) that is used to track the distribution between the available servers.

IBM Training

Weighted routing example with no affinity

	Server 1	Server 2
Start:	4	1
Request 1	3	1
Request 2	3	0
Request 3	2	0
Request 4	1	0
Request 5	0	0
Reset:	4	1

As soon as all values <=0, a reset is done

Reset adds initial values of 4, 1

Example 1: Only new hits (no session affinity)

- As requests come into the plug-in, round robin is used to distribute the hits
- Each request decrements the value in the internal table
- As soon as a server has a count of zero, no new hits are sent to that server
- As soon as all servers have a value of zero, the values are all reset
- Results:
  - server1: 80%
  - server2: 20%

Figure 3-36. Weighted routing example with no affinity

This algorithm is used for both the plug-in and ORB WLM.

Weighted routing example with affinity

Example 2: Combination of hits

- Sticky hits are those hits that have affinity
- Property **IgnoreAffinityRequests (default = true)** causes the internal table to *not* decrement the count for sticky hits
- As requests come into the plug-in, weighted round robin is used
- Each non-sticky request decrements the value in the table
- Sticky hits are routed to the server for which they have affinity
- As soon as a server has a count of zero, no new hits are sent to that server
- As soon as all servers have a value of zero, the values are all reset
- Results:
  - server1: Around 80%
  - server2: Around 20%

	Server 1	Server 2
Start:	4	1
Request 1	3	1
Request 2*	3	1
Request 3	3	0
Request 4**	3	0
Request 5	2	0
Request 6	1	0
Request 7**	1	0
Request 8	0	0
Reset:	4	1

Forced hits due to sticky sessions:
Server 1: *
Server 2: **

Figure 3-37. Weighted routing example with affinity

This algorithm is used for both the plug-in and ORB WLM.

Weighted routing example with counting affinity

Example 3: Combination of hits

- Setting **IgnoreAffinityRequests** to false
- As requests come into the plug-in, weighted round robin is used
- All requests decrement the weight values in the table
  - Sticky hits are routed to the server for which they have affinity
  - Weighted round robin is used to route non-sticky hits
- As soon as a server has a count of zero, no new hits are sent to that server
- As soon as all servers have a value of zero or less, the values are all reset
  - A reset adds the modified server weights (4 and 1) to the servers repeatedly until all servers are greater than zero
- Results (can vary):
  - server1: Around 80%
  - server2: Around 20%

	Server 1	Server 2
Start:	4	1
Request 1	3	1
Request 2*	2	1
Request 3	2	0
Request 4**	2	-1
Request 5	1	-1
Request 6**	1	-2
Request 7	0	-2
Reset:	4	-1

Forced hits due to sticky sessions:
Server 1: *
Server 2: **

Figure 3-38. Weighted routing example with counting affinity

This algorithm is used for both the plug-in and ORB WLM.

Routing alternative: Random

- An alternative algorithm to weighted round robin is random
  - Sticky hits still go to the appropriate application server
  - New hits are randomly distributed

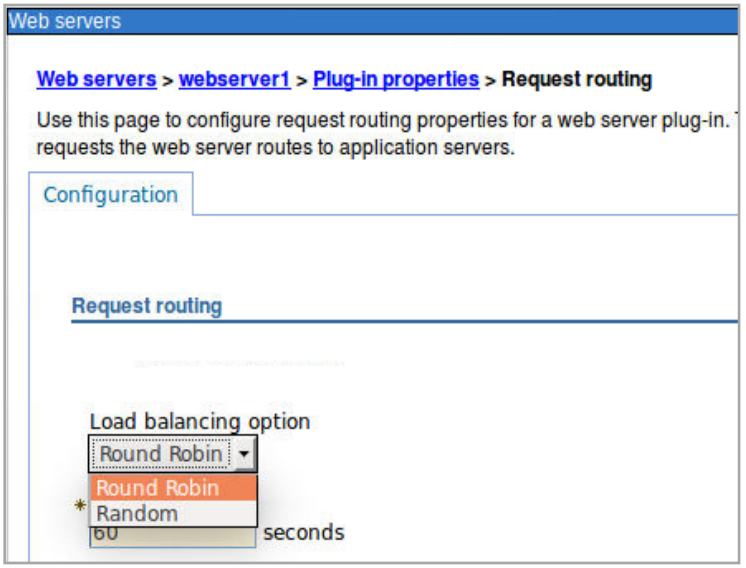


Figure 3-39. Routing alternative: Random

An alternative to the weighted round robin is random. This attribute can be set in the plug-in properties (as show in the diagram). Hits that have affinity are still routed to the appropriate server.

IBM Training

Using Intelligent Management

- The plug-in can be configured to retrieve dynamic weights rather than using static weights
- Information is added to the plugin-cfg.xml file about how to retrieve the dynamic weights
- Enabled by using:
  - **Servers > Server Types > Web servers > [web server name] > Additional Properties > Intelligent Management**
- More information in Intelligent Management Unit

Figure 3-40. Using Intelligent Management

The plug-in can be configured to use Intelligent Management features of WebSphere Application Server. When Intelligent Management is enabled, the plug-in communicates with management servers in the cell to obtain the current state of the application servers. The current state includes server weights that are calculated based on the current performance characteristics of the application servers. More detail on this topic is included in the unit on Intelligent Management.

3.4. Failover

IBM Training

Failover

Figure 3-41. Failover

IBM Training

Failover

If a failure occurs, what happens?

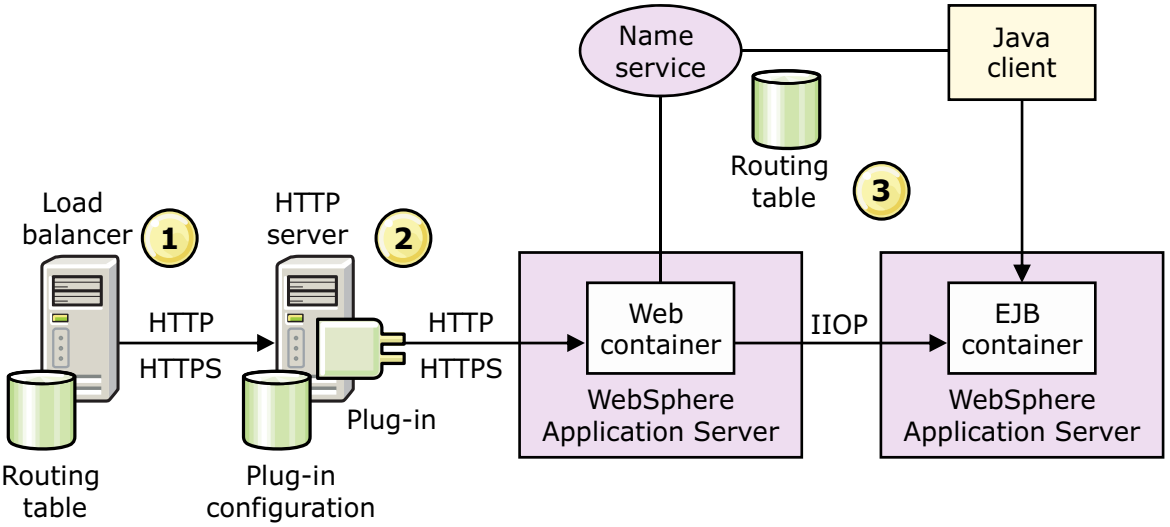


Figure 3-42. Failover

The next few slides cover what happens during failures at:

1. The load balancer
2. The web server
3. The web container

IBM Training

Edge Components failover

- Load balancer can be paired with a backup server
- Topology is active or standby
  - One computer does all the work
  - The other waits for a failure to begin handling routing

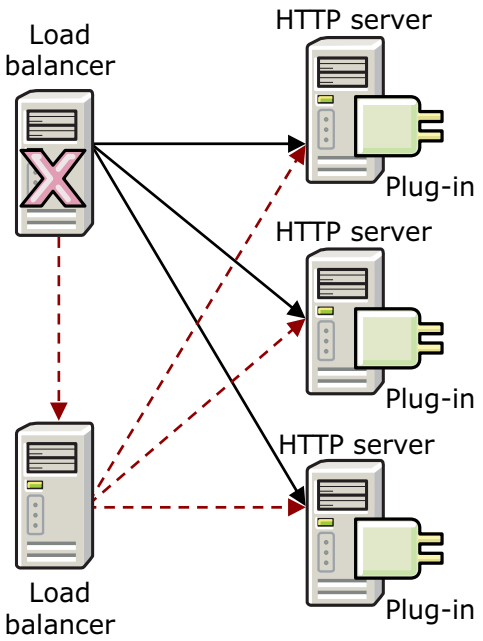


Figure 3-43. Edge Components failover

If a load balancer fails, a standby load balancer can take over. No session information is lost.

HTTP server failover

- Multiple HTTP servers provide coverage
- Load Balancer can route around a failed HTTP server
- All HTTP servers handle load before failover
- HTTP plug-in
  - Every plug-in knows about all web containers
  - Session key contains address of server
  - Sessions get properly routed

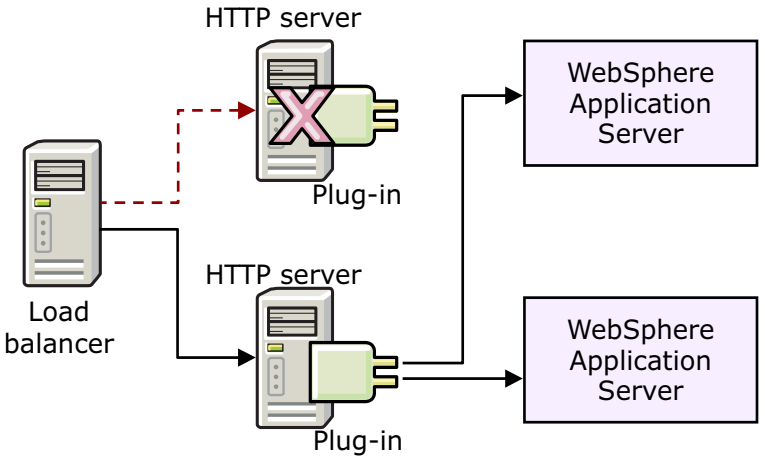


Figure 3-44. HTTP server failover

If a web server fails, any other web server can deal with the request since each plug-in is aware of all of the downstream application servers. If there is affinity to a specific application server, the request still ends up going to the correct application server (based on the JSESSIONID cookie) even if a web server is unavailable.

IBM Training

Web container failover

- HTTP server plug-in
  - Detects failure of web containers
  - Marks container as unavailable
  - Tries next cluster member in the cluster
- What about in-flight sessions?
  - Sessions can be persisted to database
  - Sessions can be replicated in memory

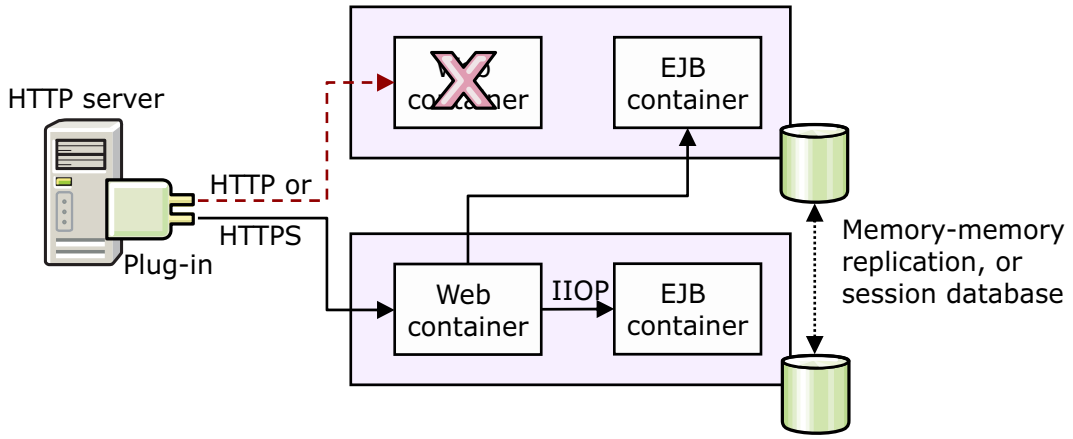


Figure 3-45. Web container failover

If a plug-in detects that a web container is unavailable, it marks that container as down and does not make more attempts to communicate with it for 2 minutes. After 2 minutes, it attempts to communicate with that server. If it is still unavailable, it is marked down for 2 minutes again, and the process continues.

This algorithm allows each plug-in to dynamically deal with web containers that either become unexpectedly unavailable or are taken offline. At the same time, when new cluster members are added and the `plugin-cfg.xml` file is generated and propagated, the plug-in automatically knows about the new containers and dynamically adds them to the pool.

EJB container failover

- Client code and ORB plug-in can route to next available cluster member
- Failure occurs before any work is initiated on the cluster member:
  - ORB automatically reroutes EJB request
  - If no other cluster member is available, throws NO_IMPLEMENT exception
- Failure occurs after EJB method call initiated work:
  - System exceptions are thrown
  - Client must determine appropriate recovery action
  - Reissue request or roll back transaction
  - If NO_IMPLEMENT exception is thrown, no recovery is possible

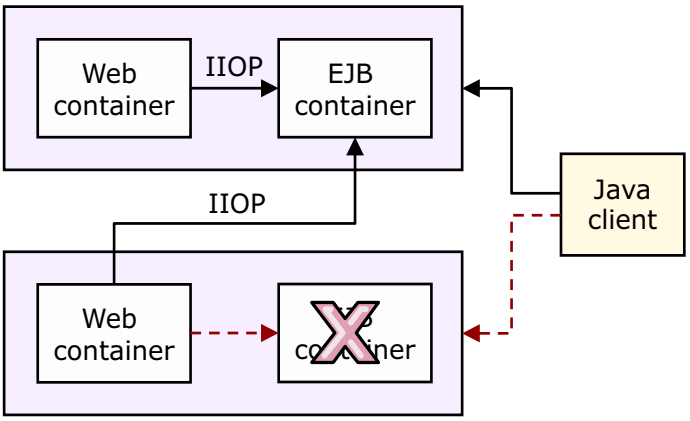


Figure 3-46. EJB container failover

When an EJB container fails, the failover depends on the initial type of failure. If the error was in reaching the EJB container, then WebSphere fails over to another possible EJB container. However, if the initial call was successful, but the response timed out, then the programmer must try again.

3.5. Session persistence

IBM Training

Session persistence

Figure 3-47. Session persistence

IBM Training

Session persistence

Session objects are cached in server memory by default
- Therefore, if the server fails, they are lost

Three methods to enable session persistence:
- Database
  - JDBC data source is used to persist session objects
  - DB2 included in package for session persistence
- Memory-to-memory replication
  - Data Replication Service (DRS) is used to copy sessions to another server
- eXtreme Scale servlet filter
  - Entitlement included with WebSphere Application Server

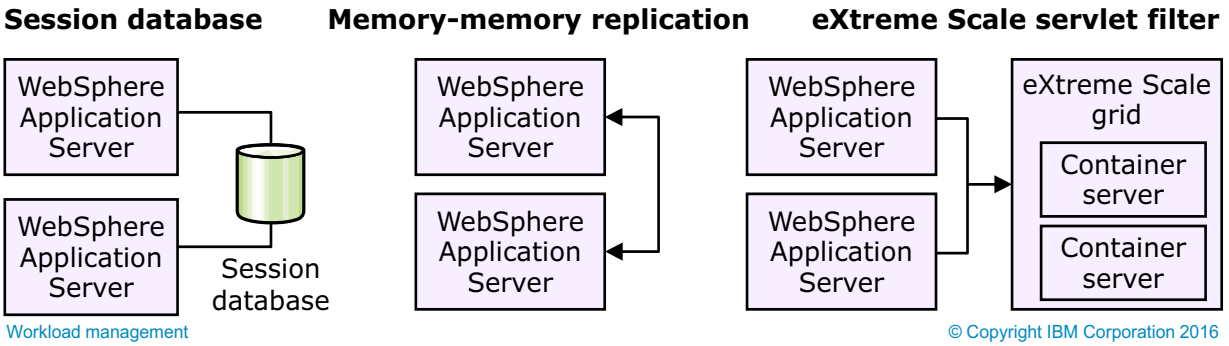


Figure 3-48. Session persistence

If your application cannot tolerate loss of session information, session persistence can be enabled. The three choices are database, memory-to-memory, and eXtreme Scale. The database approach has greater cost, but provides persistence to a database. Memory-to-memory can still fail if all copies of the session are lost. The eXtreme Scale solution can be more robust.

A number of settings can be used to affect the balance between failover and performance. This topic is described in upcoming slides.

IBM Training

Session configuration: Memory-to-memory

- Configuring memory-to-memory session replication
  - Found under **WebSphere application servers > <*servername*> > Session management > Distributed environment settings**
  - Select **Memory-to-memory replication**
  - Configure the replication domain

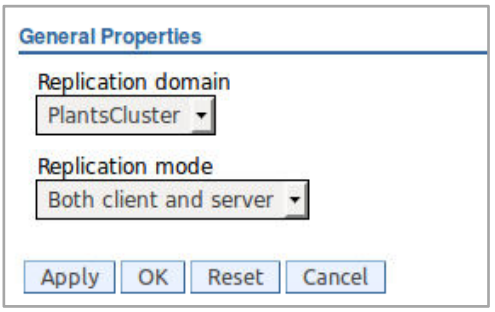


Figure 3-49. Session configuration: Memory-to-memory

Replication modes:

- Server mode: Stores copies of other WebSphere Application Server sessions and not to send out copies of any session that are created in that particular server
- Client mode: Broadcasts or sends out copies of the sessions it owns and not to receive backups of sessions from other servers
- Both mode: Simultaneously broadcasts or sends out copies of the sessions it owns and acts as a backup table for sessions that other WebSphere Application Server instances own

IBM Training

Session configuration: Replication domains

Creating replication domains:

- Created during cluster creation, or
- **Environment > Replication Domains > New**

To configure replication domains after creation:

- Through the administrative console, select the Replication domain under **Environment > Replication domains**
- Select the replication domain
- Select the number of replicas
  - Single replica
  - Entire domain
  - Specified

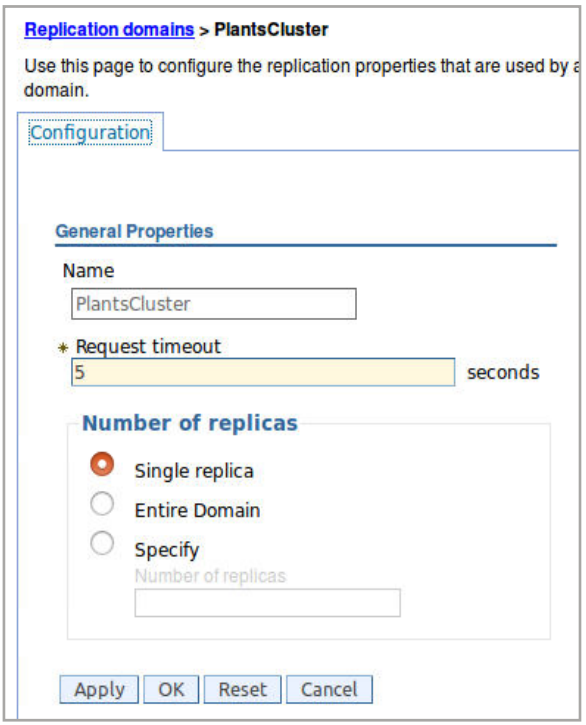


Figure 3-50. Session configuration: Replication domains

Single replica means that for every session object, one backup is stored on a different server for failover. It is also possible to back up the sessions on every member of the domain. This practice does not scale as well because every member must store every other member's session objects. However, it means that the chance for a successful failover is higher.

Depending on the needs of the application and the tolerance for failure, it is possible to tune the number of members that back up every session object.

IBM Training

Database persistence configuration

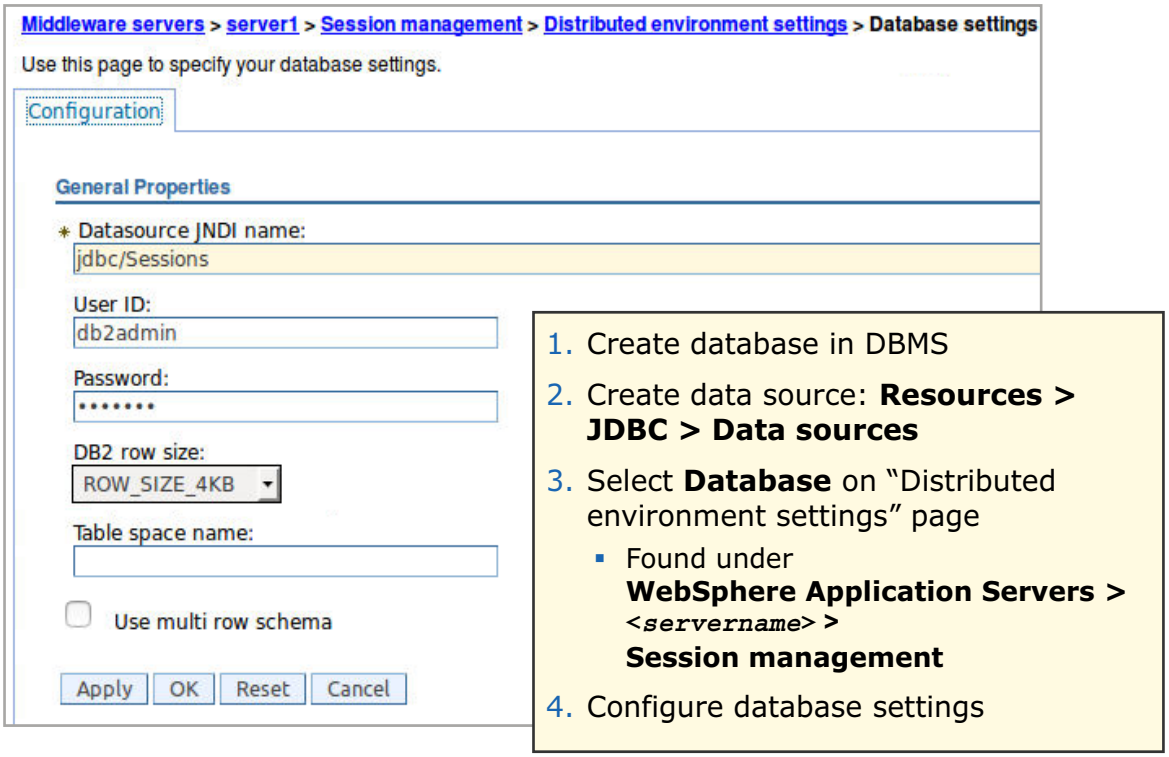


Figure 3-51. Database persistence configuration

Database persistence configuration requires specifying a data source. This picture shows the console database replication and data source configuration settings.

Tuning session persistence

- Session persistence can be tuned to favor performance or failover
- Accesses through **WebSphere Application Servers > *<servername>* > Session management > Distributed environment settings > Tuning parameter**

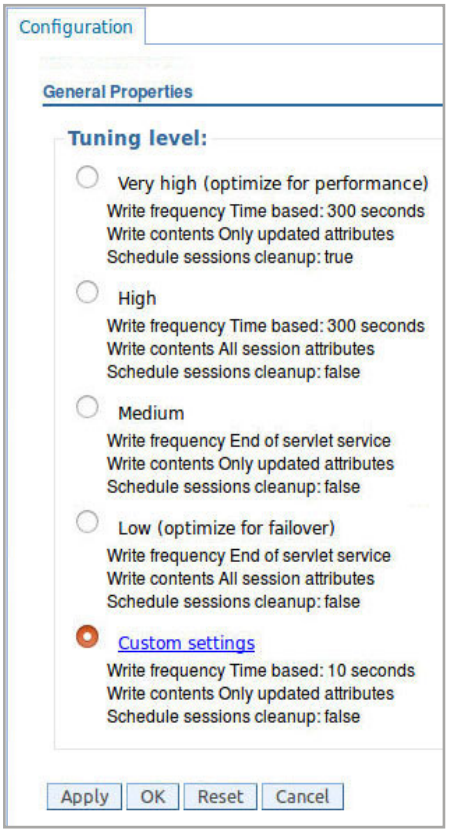


Figure 3-52. Tuning session persistence

Session persistence can be tuned so that it is more appropriate for failover or for performance. These settings allow administrators to choose if they want to have greater failover ability (at the price of performance) or have better performance (at the price of having failover).

eXtreme Scale persistence configuration

- Create `splicer.properties` file
  - Provides eXtreme Scale configuration details
- "Splice" HTTP servlet filter into a web application
  - Run script

```
"addObjectGridFilter.[bat|sh]
   <location of ear/war file>
   <location of splicer.properties file>"
```

- Deploy application

Figure 3-53. eXtreme Scale persistence configuration

To activate the HTTP servlet filter for WebSphere eXtreme Scale, you must "splice" the filter into your application. eXtreme Scale includes a script, addObjectGridFilter, to add the filter to your application. The script takes two parameters: the absolute path to the application to be spliced, and the location of a file that contains various configuration properties. The splicer utility modifies the application's deployment descriptor, `web.xml`, with parameters that control how the filter interacts with the WebSphere eXtreme Scale grid.

Follow the normal procedure to deploy the new application to your application server. The servlet filter is started as part of the application initialization.

IBM Training

Unit summary

- Define workload management
- Create clusters and cluster members
- Compare clustered configurations
- Explain how weights are used in workload management
- Describe failover scenarios
- Describe the role of the HTTP plug-in in workload management
- Explain session management
- Configure distributed session management

Figure 3-54. Unit summary

Review questions

1. A WebSphere cluster member is what type of process?
   A. An application server
   B. A web server
   C. An edge server
   D. A proxy server

2. The creation of a cluster can be based on which of the following choices?
   A. An application
   B. An application server
   C. An enterprise application
   D. An application manager

3. True or False: Having session affinity means that session information is not lost during failover.

Figure 3-55. Review questions

Write your answers here:

1.

2.

3.

Review answers

1. A WebSphere cluster member is what type of process?
   A. An application server
   B. A web server
   C. An edge server
   D. A proxy server
   The answer is A.
2. The creation of a cluster can be based on which of the following choices?
   A. An application
   B. An application server
   C. An enterprise application
   D. An application manager
   The answer is B.
3. True or False: Having session affinity means that session information is not lost during failover.
   The answer is False. Session information can be lost during failover with or without session affinity.

Figure 3-56. Review answers

IBM Training

Exercise: Clustering and workload management

Figure 3-57. Exercise: Clustering and workload management

IBM Training

Exercise objectives

After completing this exercise, you should be able to:

- Create a cluster and add cluster members
- Map modules to clusters and web servers
- Test load balancing and failover between two cluster members
- Configure a data replication domain for session management

Figure 3-58. Exercise objectives

Unit 4. WebSphere security: SSL

Estimated time

01:30

Overview

In this unit, you learn basic security concepts and architecture that apply to WebSphere Application Server. You learn how to configure administrative security, application security, and Secure Sockets Layer (SSL).

How you will check your progress

- Review questions
- Lab exercises

References

WebSphere Application Server V9 Knowledge Center

https://www.ibm.com/support/knowledgecenter/SSEQTP_9.0.0/as_ditamaps/was900_welcome_base.html

IBM Training

Unit objectives

- Describe the basic concepts of SSL
- Describe the purpose of certificates and certificate authorities
- Describe the role of SSL within a WebSphere cell

Figure 4-1. Unit objectives

IBM Training

Topics

- SSL basics
- Certificates and certificate authorities
- SSL within a WebSphere cell

Figure 4-2. Topics

4.1. SSL basics

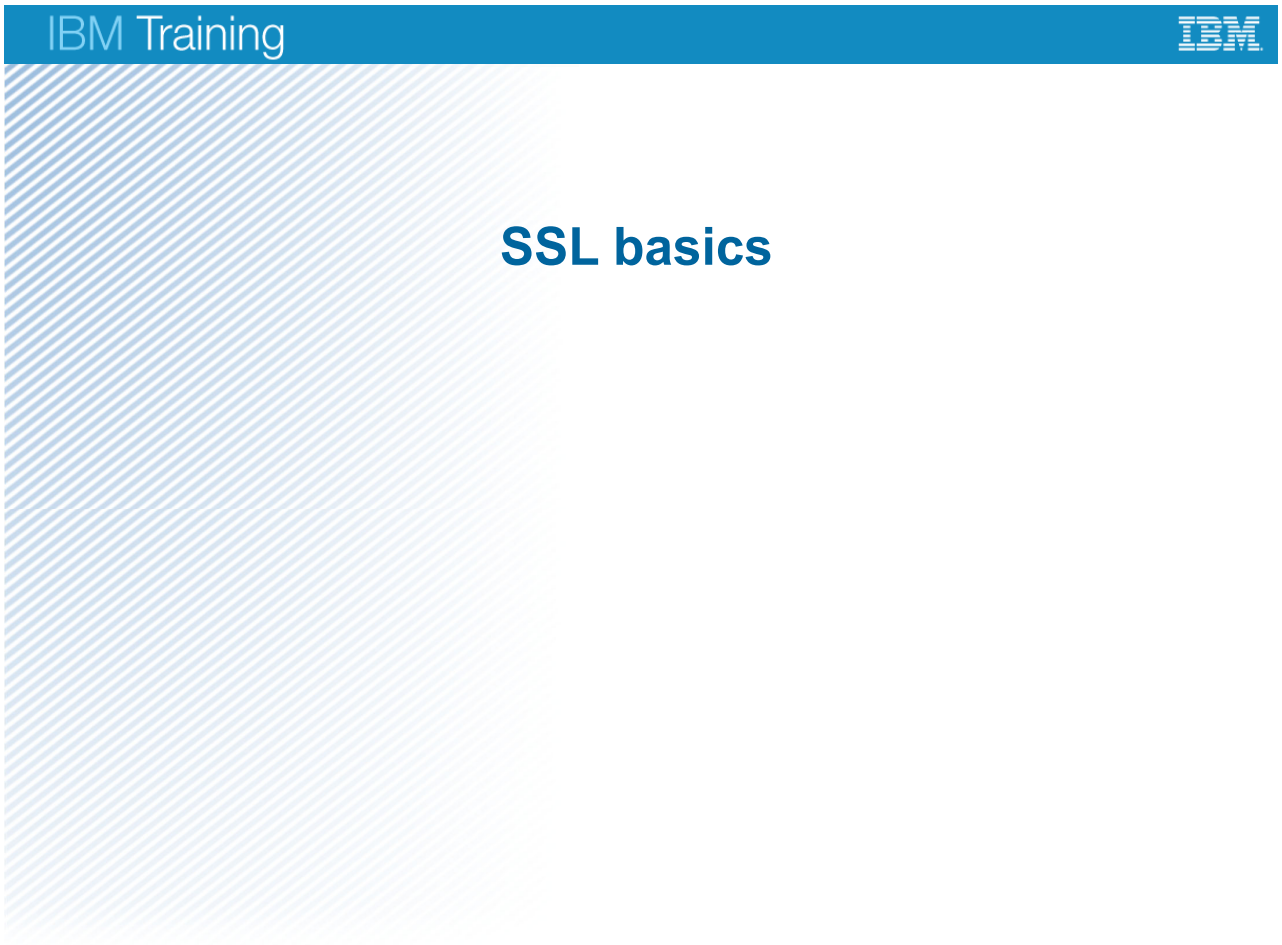


Figure 4-3. SSL basics

What is SSL?

- SSL stands for Secure Sockets Layer
- SSL provides connection security through:
  - Communication privacy: The data on the connection can be encrypted
  - Communication integrity: The protocol includes a built-in integrity check
  - Authentication: The client knows who the server is
- Creates a VPN
  - Uses both symmetric and asymmetric key encryption

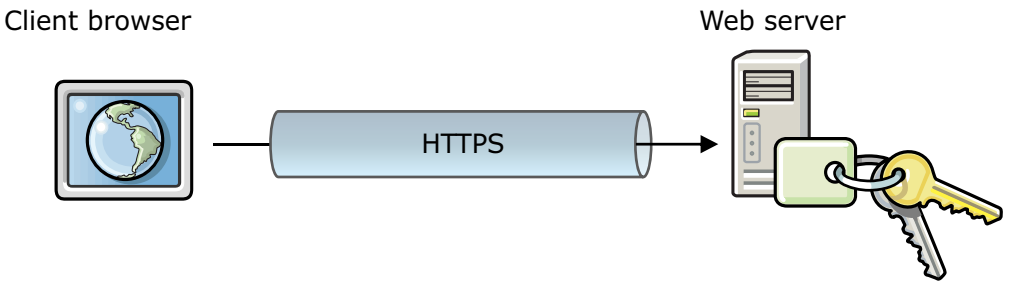


Figure 4-4. What is SSL?

Secure Sockets Layer (SSL) provides transport level security between two points, much like a VPN. It is often used to secure communications between a browser and a web server.

Symmetric key encryption

- Symmetric or secret key technology is a model in which two parties have a shared secret
- The same key is used for both encryption and decryption

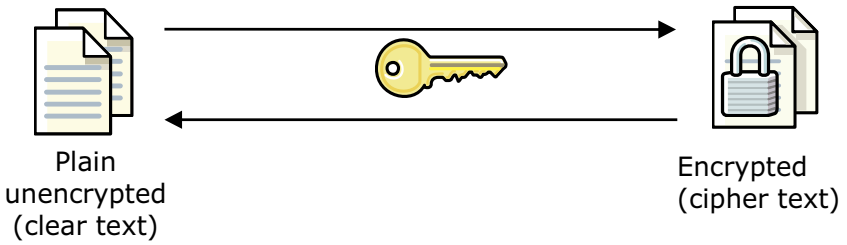


Figure 4-5. Symmetric key encryption

If a server has a public-private set, it can send out its public key (through a signing certificate; also known as a certificate) to client machines. Those client machines can then use that public key to encrypt messages that are destined for the server, which then only the server can decrypt. Unlike symmetric key encryption, this process does not require the client and server to have a shared secret.

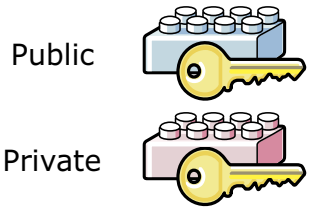
Since the client can validate the certificate of the server, there is one-way authentication. But the server has no way to authenticate the client. Nor can the server send the client secured messages.

IBM Training

Asymmetric key encryption

Public key cryptography

- Two keys that are cryptographically related:
  - Public key (can share with everyone)
  - Private key (must never be shared; possession is proof)
- Keys are asymmetric:
  - Given message is encrypted with one key and decrypted with another
  - Symmetric, secret key technology uses the same key for encryption and decryption



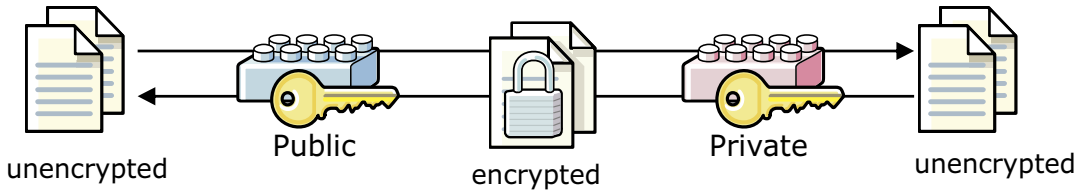


Figure 4-6. Asymmetric key encryption

If a server has a public-private set, it can send out its public key (through a signing certificate; also known as a certificate) to client machines. Those client machines can then use that public key to encrypt messages that are destined for the server, which then only the server can decrypt. Unlike symmetric key encryption, this process does not require the client and server to have a shared secret.

Since the client can validate the certificate of the server, there is one-way authentication. But the server has no way to authenticate the client. Nor can the server send the client secured messages.

How does SSL work?

SSL uses a combination of asymmetric and symmetric encryption to create a session between the client and server

- Asymmetric encryption is used to negotiate a session key (shared secret)
  - Asymmetric encryption is slow but does not require a shared secret
- Symmetric encryption is used to transfer data between the client and server
  - Symmetric encryption is fast but requires a shared secret

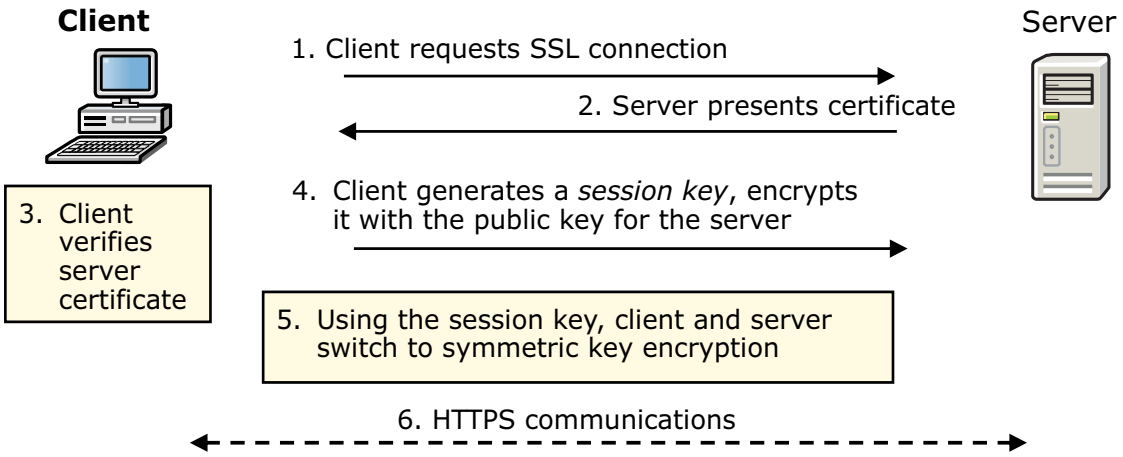


Figure 4-7. How does SSL work?

Because the client chooses its own session key, nobody else knows it. It can securely use the public key of the server to send that session key to the server. Now, nobody but the client and server know the session key. The session key is then used as a “shared secret” to switch to the much more efficient symmetric key encryption.

A certificate (or signing certificate) contains information about the server, including the public key of the server, and the certificate authority digitally signs it.

4.2. Certificates and certificate authorities

IBM Training

Certificates and certificate authorities

Figure 4-8. Certificates and certificate authorities

What is a certificate?

Simple answer:

- It is an electronic document that identifies you, and a third party vouches for both you and the certificate itself
- Examples:
  - Employee badge (vouched for by your employer)
  - Drivers license (vouched for by your state)
  - Passport (vouched for by your country)

More information:

- Includes information about you
- Includes public key
- A certificate authority digitally signs it

Figure 4-9. What is a certificate?

A digital certificate is an electronic document that identifies you, which a certificate authority then signs. This certificate means that the CA validates that you are who you say you are and is therefore vouching for you.

The certificate contains not just information about you, but also your public key. It is important to note that the public key is the counterpart to your private key (which is part of your personal certificate) which is kept secret.

IBM Training

Types of certificates

Different types of certificates:

- Certificate
  - Contains a public key that is signed
  - Contains information about the owner of the certificate
  - Contains certificate expiration date
- Personal certificate:
  - Typically meant as the certificate along with private key data
- Signer certificate:
  - The certificate that corresponds to the private key used to digitally sign another certificate

Figure 4-10. Types of certificates

The term “certificate” is sometimes confused since it is used in different ways. Generally speaking, a certificate contains just a public key and information about you.

A personal certificate is a certificate that contains the private key, which must be kept private and secret.

A signer certificate is just a regular certificate. In other words, it contains a public key and information about the owner. What specifically makes it a “signer certificate” is the fact that the corresponding personal certificate (private key) was used to sign a certificate. This case occurs for a CA or a chained certificate when some entity (usually a CA, or in the case of WebSphere, a cell root certificate) signs your certificate. Therefore, you need that signer certificate (which contains the signer’s public key) to validate SSL connections.

IBM Training

What is a certificate authority (CA)?

- An entity that signs public keys, thus creating certificates
  - A CA validates Kate's identity before it vouches for her (signing her certificate)
  - A special type of signer (a trusted signer)
- Signers of known CAs are built into browsers

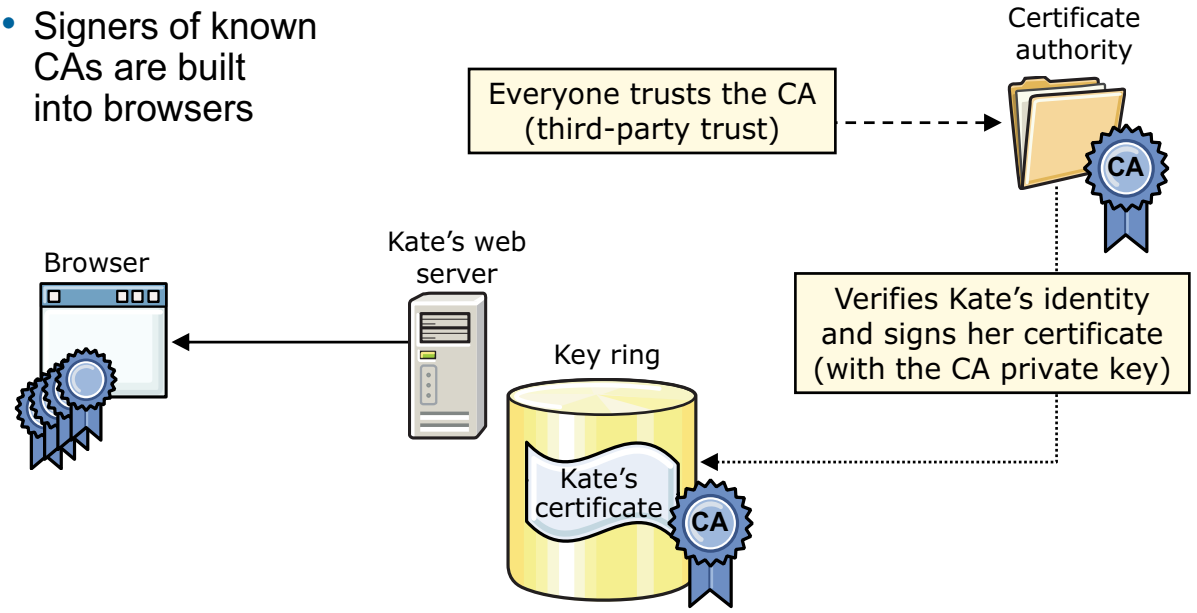


Figure 4-11. What is a certificate authority (CA)?

If a client is going to trust a certificate from a server, the client must be able to validate that certificate. This validation is possible because a certificate authority generates and signs certificates with its private key. That means that the browser can use the public key of the CA to verify the digital signature. The public keys from the standard CAs are built into browsers. If the browser does not have a copy of the public key of the CA, the user is prompted.

Public key infrastructure (PKI) is:

- Based on public-private key cryptography
- The whole infrastructure that makes public-key security work:
  - Certificate authority (CA)
  - Registration authority (RA)
  - PKI enabled applications
  - Directory (optional)

SSL: Putting it all together

The SSL handshake establishes:

- The identity of the server (based on trusting the CA)
  - The server provides a CA signed certificate
  - The server proves that it has the corresponding private key
  - Therefore, Joe trusts that Kate really is Kate
- Encrypted (with symmetric key) channel between the browser and server

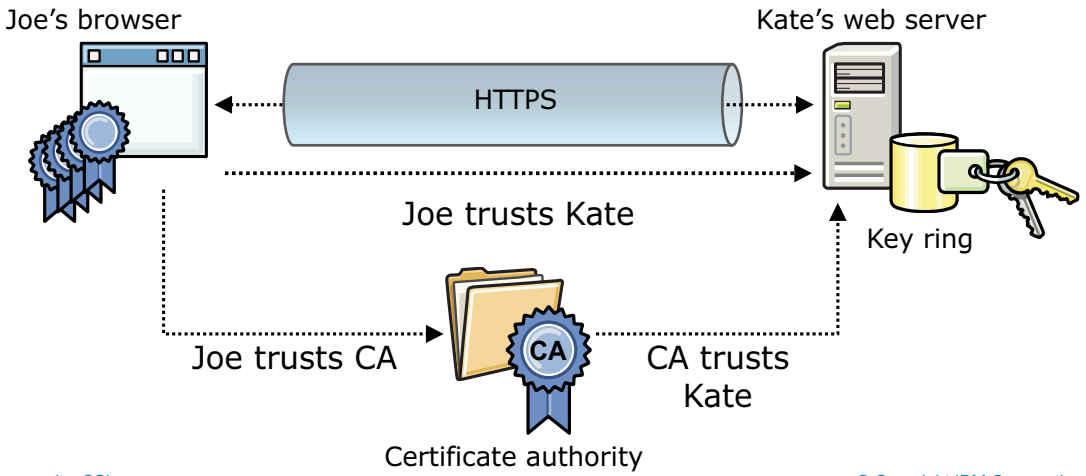


Figure 4-12. SSL: Putting it all together

Putting it together, Joe's browser is willing to trust Kate's server because Kate presented her certificate to Joe. The question then becomes how does Joe trust that the certificate really belongs to Kate? The answer is because the CA signed Kate's certificate, which means the CA verified that Kate was really Kate. Since Joe trusts the CA, he is therefore willing to trust Kate.

4.3. SSL within a WebSphere cell

IBM Training

SSL within a WebSphere cell

Figure 4-13. SSL within a WebSphere cell

SSL within WebSphere Application Server

- SSL can be used to secure network traffic for a number of links
  - From the client to the web server
  - From the plug-in to the application server
  - Other network links can also be secured (LDAP and others)
- The *administrative console* (or iKeyman) can be used to create and manage the necessary keys and keystores
  - Keystores contain digital certificates that are needed for SSL to establish secure communication between two points

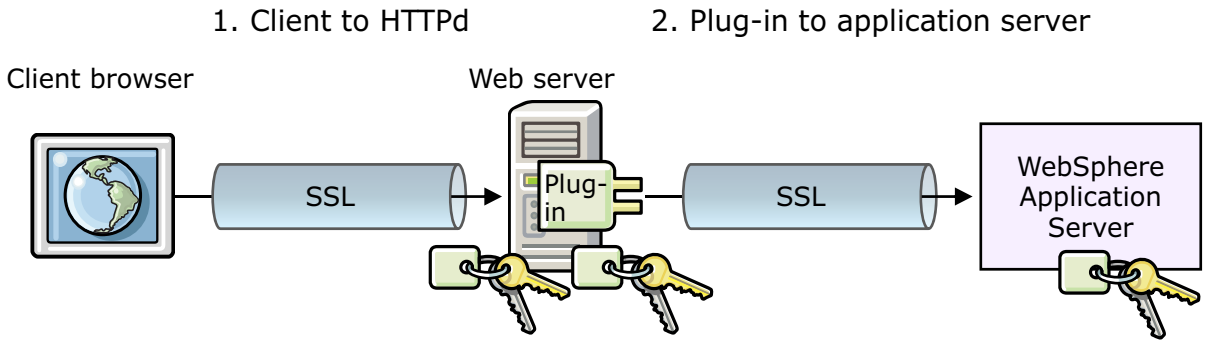


Figure 4-14. SSL within WebSphere Application Server

When thinking about SSL within a WebSphere cell, a number of different relationships must be taken into account. For example, consider the connection between the browser and the web server. That relationship is understood, and independent from the relationship between the plug-in and the application server. There are also the SSL connections within the actual cell (between the managed processes: deployment manager, node agents, application servers). Additionally, there might be SSL connections to an LDAP server, web services, external cell, and others.

IBM Training

WebSphere SSL management

- WebSphere automatically creates node certificates
  - The cell root certificate (called a “chained certificate”) signs the node certificates
  - Node certificates are stored in a node-specific keystore
- Cell-wide truststore includes cell root signer certificate
  - Therefore, each node can validate certificates that other nodes present
- An expiration manager automatically renews expiring keys (default behavior)
- The keystores and truststores are stored within the cell configuration
  - Therefore, they are distributed to the nodes through file synchronization

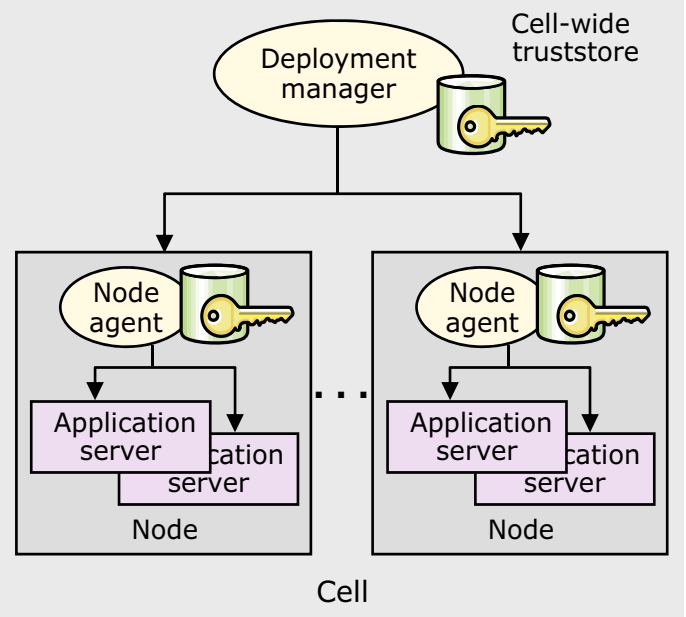


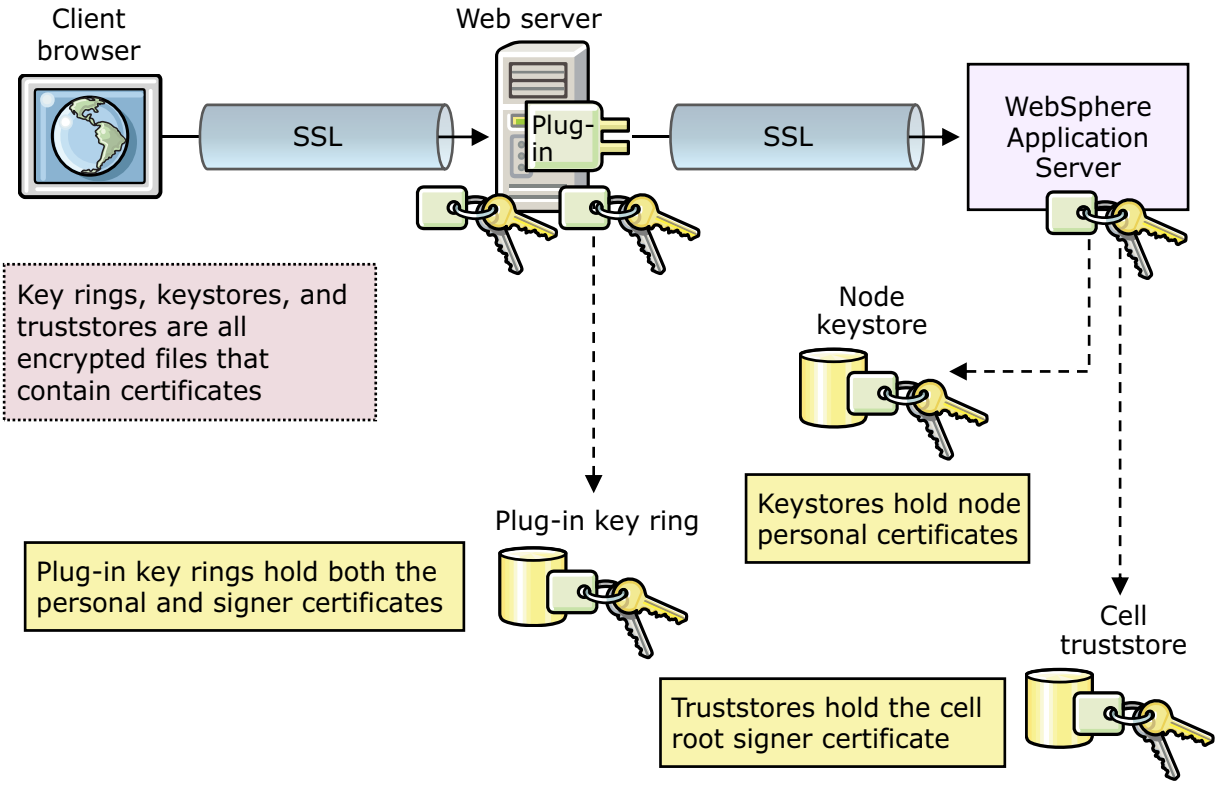
Figure 4-15. WebSphere SSL management

Each node within a cell gets a node personal certificate (1-year life span). The cell root certificate signs the node personal certificates (15-year life span). By default, all of the application servers on a node use the node certificates of that node. The node certificates are stored in node-specific keystores, and the cell root signer certificate is stored in the cell truststore. This storage method means that all nodes, through file synchronization, have access to the cell root signer.

This arrangement has several desirable side effects. First, all of the nodes can securely communicate with each other after validating themselves. Second, when a personal certificate is replaced (for example, when it expires), the other nodes still accept the new certificate. It is accepted because a known signer signed it (a new signer is not required to be distributed again, as it does when the certificate is self-signed).

IBM Training

What are key rings, keystores, and truststores?



WebSphere security: SSL

Figure 4-16. What are key rings, keystores, and truststores?

The plug-in uses a single keystore (or key ring) which is generated for it by WebSphere. This file contains both the personal certificate for the plug-in and the cell root signer.

The application servers (or nodes) use two different files. A keystore contains the node personal certificate, and the truststore holds the signer certificates that the node chooses to trust. By default, that means just the cell root signer certificate.

Node certificates

- Each node has a node certificate
  - The cell root certificate (a chained certificate) signs the node certificate
  - Therefore, the cell root signer is needed to validate the node certificate
  - The application servers, by default, all use the local node certificate

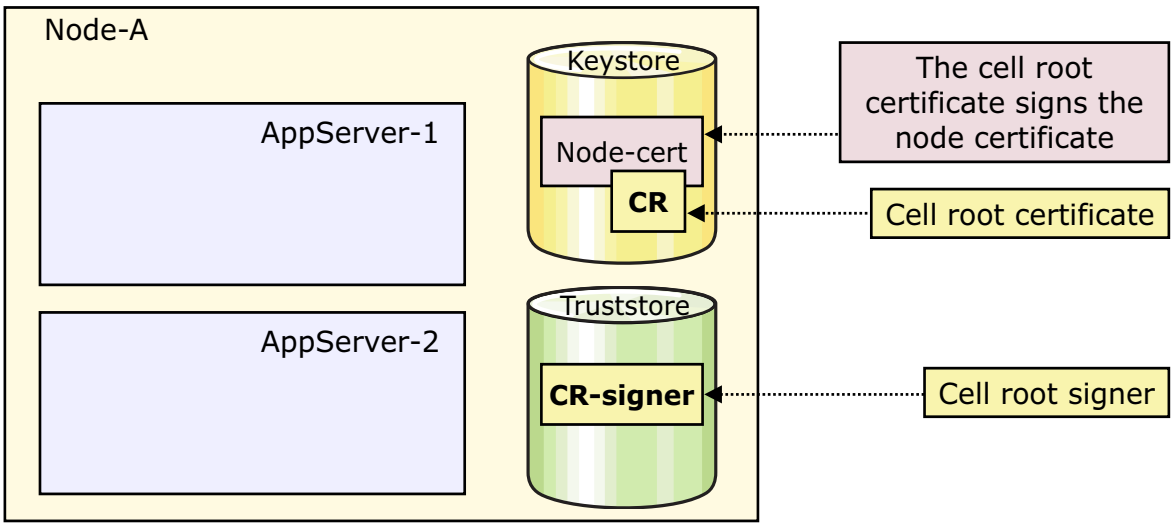


Figure 4-17. Node certificates

A keystore contains the node personal certificate (which the cell root certificate signs) and the truststore holds the signer certificates that the node chooses to trust. By default, that means just the cell root signer certificate.

It is stored in `.p12` files within the `config` directory of the profile:

`<profile-root>/config/<cell-name>/nodes/<node-name>/key.p12`
`<profile-root>/config/<cell-name>/trust.p12`

Cell default truststore

- The cell default truststore contains signers, include the cell root signer
  - It is synchronized to all the members of the cell
  - All the nodes use the common cell default truststore
  - Although node truststore files exist, they are not used by default

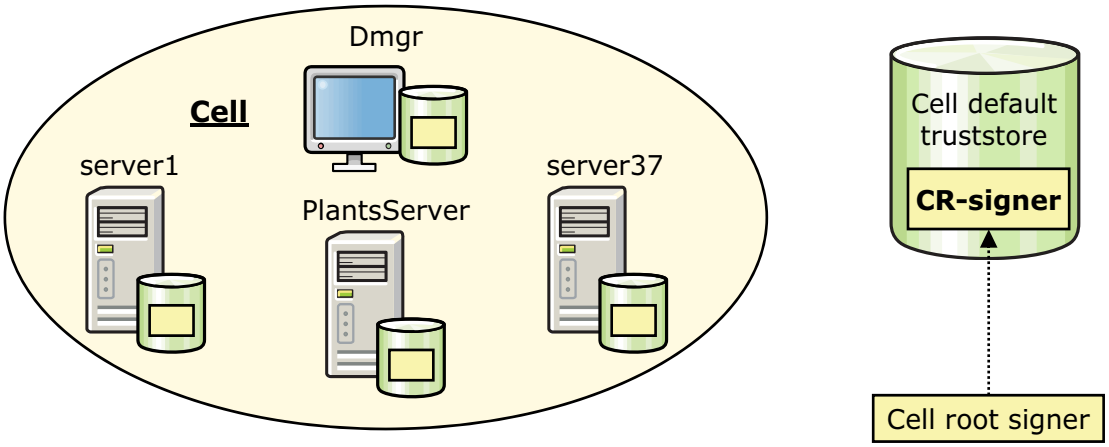


Figure 4-18. Cell default truststore

The cell default truststore contains the cell root signer certificate. To be able to validate potential SSL connections, each node needs the cell root signer. This file is made available to each of the nodes through standard file synchronization.

IBM Training

Managing WebSphere keystores

- Keystores and certificates for the cell, nodes, and plug-ins can be managed directly from the console
- Expiration management
- Keystores
- Trust files
- Certificates

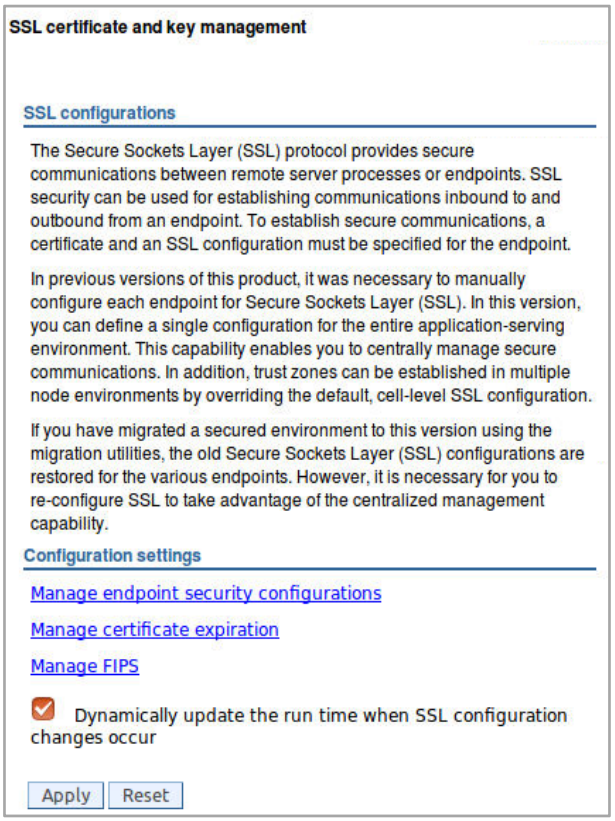
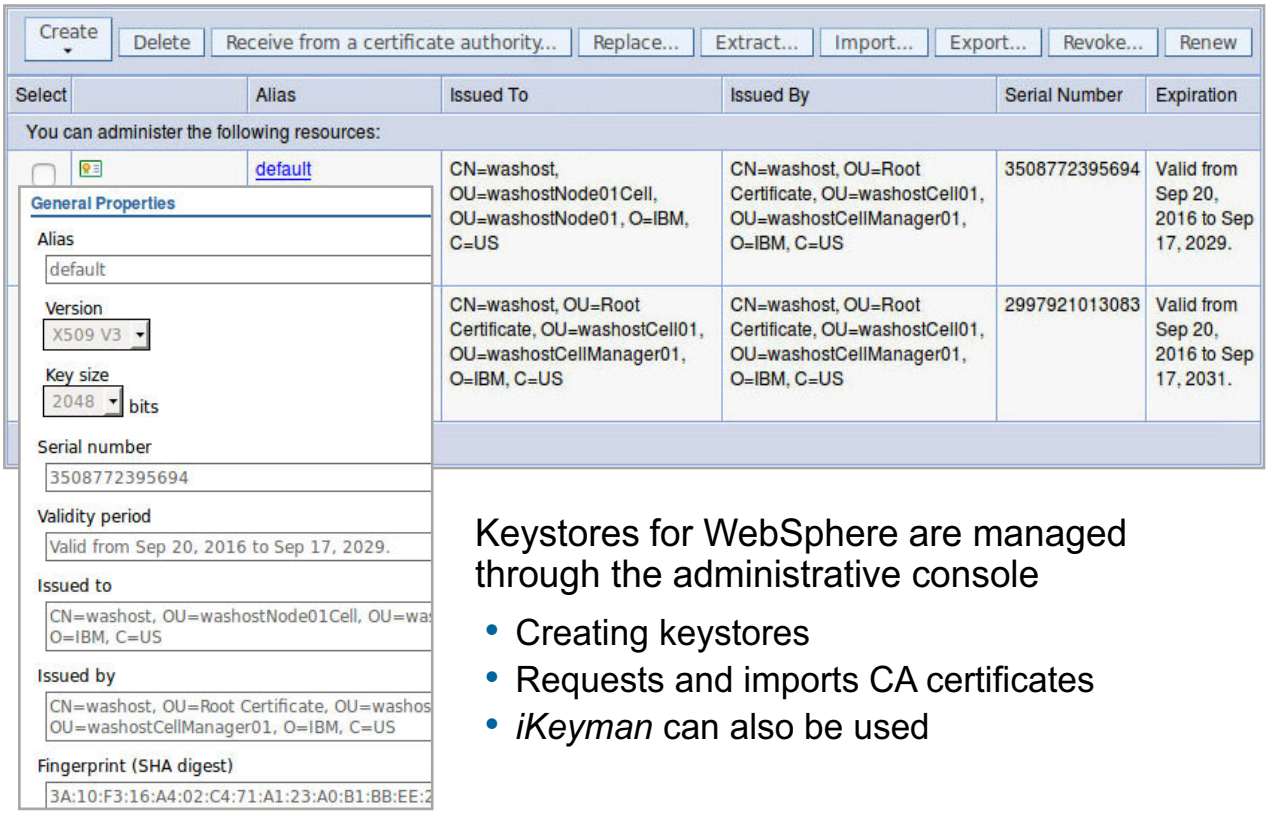


Figure 4-19. Managing WebSphere keystores

This screen shows the initial administrative console page for viewing and managing the SSL settings and configurations.

IBM Training

Creating keystores and certificates



Keystores for WebSphere are managed through the administrative console

- Creating keystores
- Requests and imports CA certificates
- *iKeyman* can also be used

Figure 4-20. Creating keystores and certificates

This screen shows the administrative console view of a node certificate. It shows the initial information about not just the node certificate, but also the signer certificate (which in this case would be the cell root signer). If the link for the certificate is clicked, more details are displayed.

IBM Training

What is a chained certificate?

- A chained certificate is merely a certificate that another certificate signs
- The cell root certificate (sometimes called a mini-CA) signs the node certificate

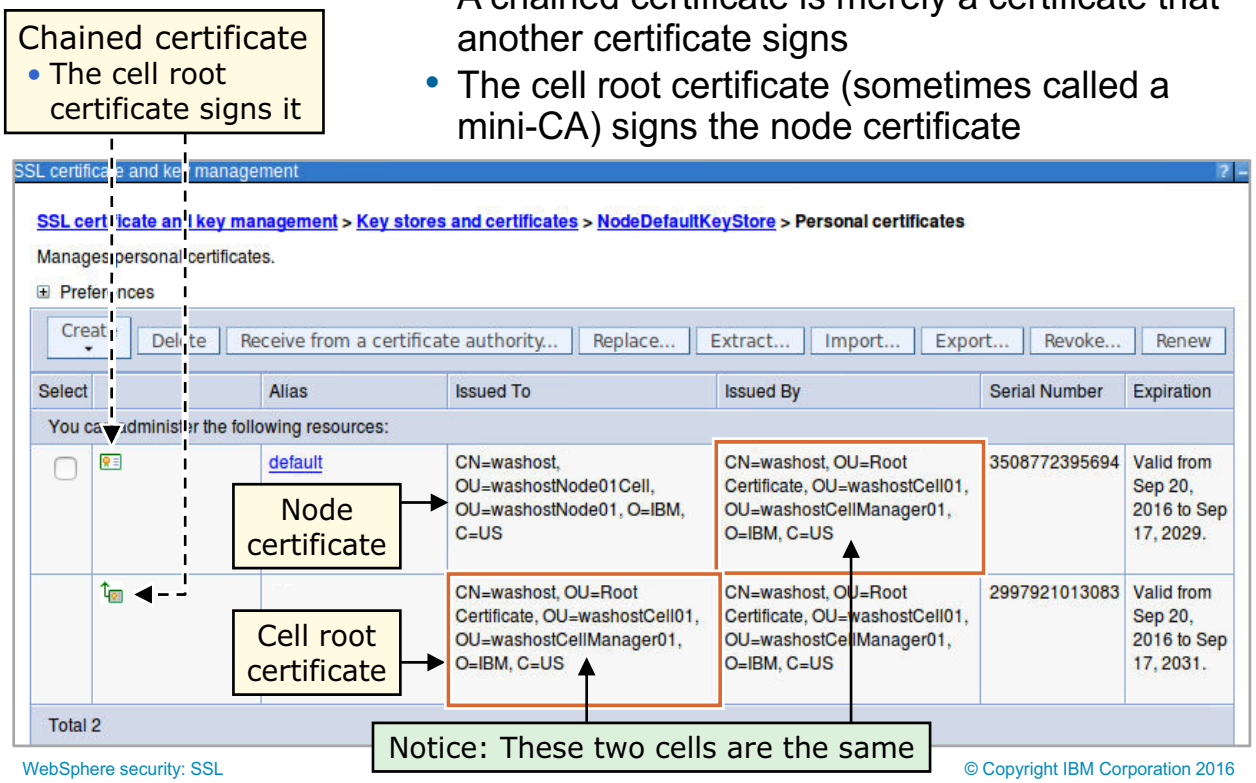


Figure 4-21. What is a chained certificate?

A chained certificate means that the cell root signer signs the node certificate. It is demonstrated in this screen capture by noting that the “Issued By” information for the node certificate matches the “Issued To” information for the signer. It is also worth noting that the “Issued To” and “Issued By” fields of the signer match each other, which indicates that the signing certificate is a self-signed certificate.

Some refer to the cell root signing certificate as a mini-CA since it signs all of the node certificates within a cell. Logically, it is what a CA does on a larger level.

IBM Training

Expiration manager scheduling

- The expiration manager can be run manually or through a schedule
- Running manually can be useful since you actively monitor the log file and thus generate a list of certificates that are going to expire soon

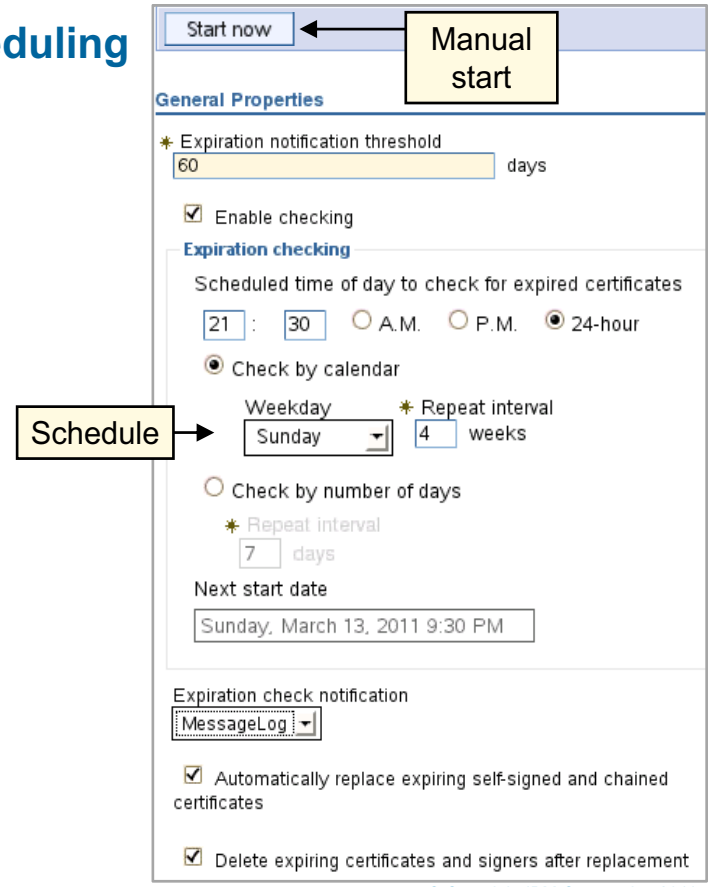
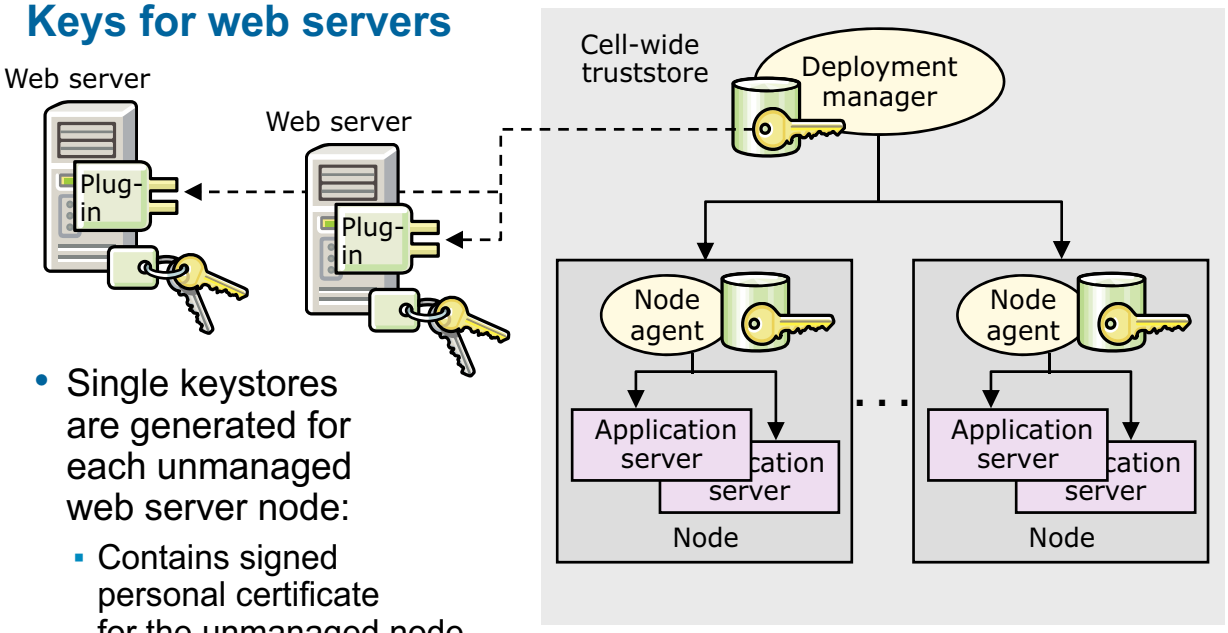


Figure 4-22. Expiration manager scheduling

WebSphere is able to automatically manage the expiration of its certificates. By default, the expiration management thread runs every four weeks (on Sundays at 21:30). If the certificates are close to expiring, they are renewed. Notification thresholds allow the administrator to also receive warnings before certificates are renewed.

IBM Training

Keys for web servers



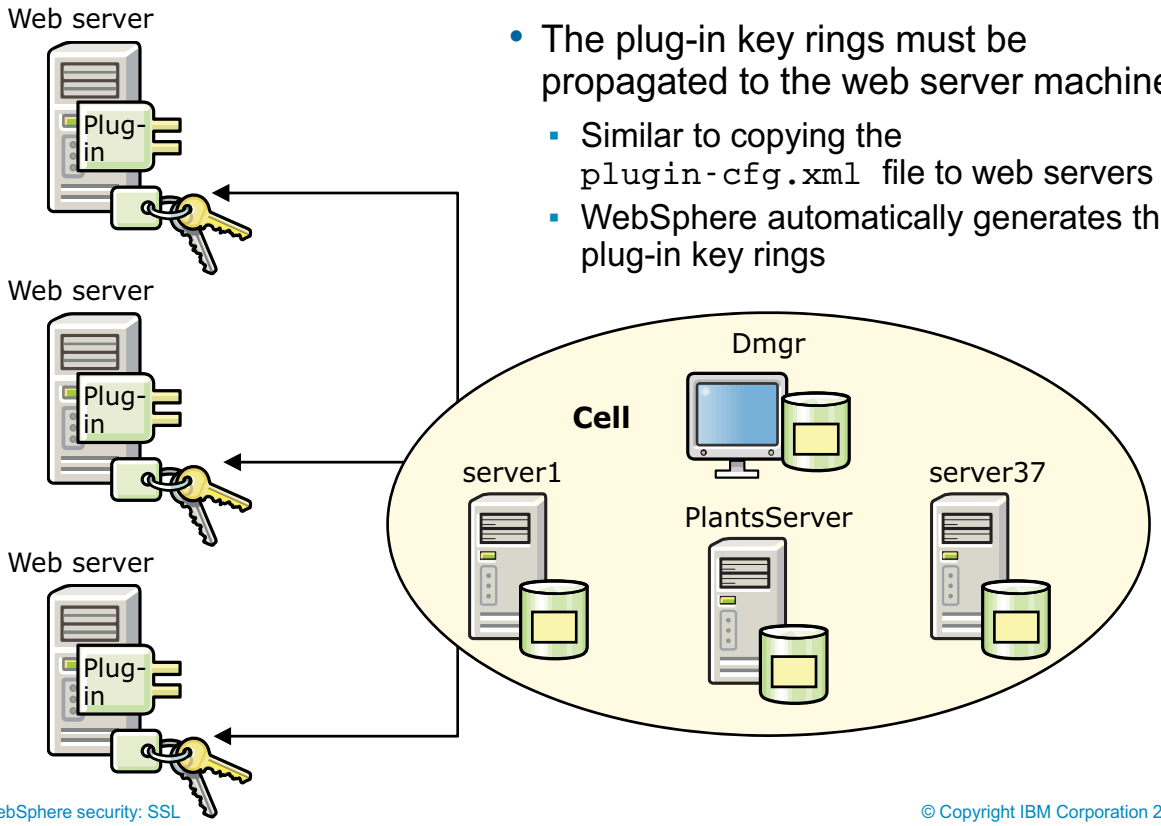
- Single keystores are generated for each unmanaged web server node:
  - Contains signed personal certificate for the unmanaged node (which the cell root certificate signs)
  - Includes the cell root signer certificate, allowing the plug-ins to communicate with the nodes securely
- Important: These key rings must be distributed to the web servers

Figure 4-23. Keys for web servers

Since the plug-in keystores include the cell root signer, the plug-in is able to complete an SSL handshake with any of the nodes since the cell root certificate signed all of their certificates.

IBM Training

Web server plug-in keystores propagation



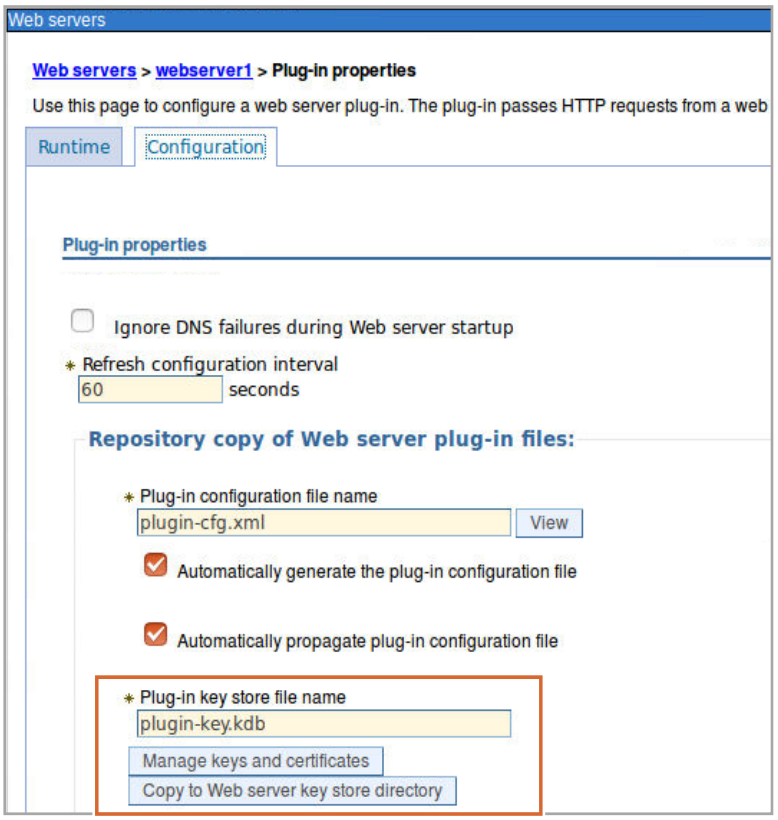
- The plug-in key rings must be propagated to the web server machines
  - Similar to copying the `plugin-cfg.xml` file to web servers
  - WebSphere automatically generates the plug-in key rings

Figure 4-24. Web server plug-in keystores propagation

Propagation of the plug-in keystores is typically done manually since file synchronization is not available.

IBM Training

IBM HTTP Server key ring propagation



- Web server keystores are automatically generated
  - Can be managed from the administrative console
- The keystore for IBM HTTP Server servers can be remotely propagated

Figure 4-25. IBM HTTP Server key ring propagation

WebSphere automatically generates keystores for defined web server plug-ins. These keystores then must be copied to the plug-ins. These keystores can be managed through the administrative console. With IBM HTTP Server, it is also possible to configure WebSphere to propagate the keystore that WebSphere generates to the web server plug-in.

IBM Training

Unit summary

- Describe the basic concepts of SSL
- Describe the purpose of certificates and certificate authorities
- Describe the role of SSL within a WebSphere cell

Figure 4-26. Unit summary

Review questions

1. Which type of encryption does SSL perform?
   A. Symmetric key encryption
   B. Asymmetric key encryption
   C. Both symmetric and asymmetric key encryption
   D. Neither
2. Which type of certificate does WebSphere use for the nodes?
   A. Self-signed certificate
   B. Personal certificate
   C. Signed certificate
   D. Chained certificate

Figure 4-27. Review questions

Write your answers here:

1.

2.

Review answers

1. Which type of encryption does SSL perform?
   A. Symmetric key encryption
   B. Asymmetric key encryption
   C. Both symmetric and asymmetric key encryption
   D. Neither

   The answer is C.

2. Which type of certificate does WebSphere use for the nodes?
   A. Self-signed certificate
   B. Personal certificate
   C. Signed certificate
   D. Chained certificate

   The answer is D.

Figure 4-28. Review answers

IBM Training

Exercise: Configuring SSL for WebSphere

Figure 4-29. Exercise: Configuring SSL for WebSphere

IBM Training

Exercise objectives

After completing this exercise, you should be able to:

- Define the certificate life span of a profile
- Use the administrative console to find and view certificates within the cell
- Configure and run the certificate expiration service
- Propagate the generated plug-in keystore out to the plug-in
- Create a keystore for a web server
- Generate a self-signed key
- Configure IBM HTTP Server to load and use HTTPS

Figure 4-30. Exercise objectives

Unit 5. Overview of Intelligent Management

Estimated time

01:00

Overview

This unit introduces the features of Intelligent Management.

How you will check your progress

- Review questions

References

https://www.ibm.com/support/knowledgecenter/SSEQTP_9.0.0/as_ditamaps/was900_welcome_base.html

IBM Training

Unit objectives

- Define Intelligent Management
- Describe virtualization and autonomic computing
- Define intelligent routing
- Describe dynamic workload management
- Describe the health management features of Intelligent Management
- Describe the application edition management features of Intelligent Management
- Describe the performance management features of Intelligent Management

Figure 5-1. Unit objectives

IBM Training

Topics

- Overview of Intelligent Management
- Intelligent Management components
- Health management
- Application edition management
- Performance Management
- Deployment manager high availability

Figure 5-2. Topics

5.1. Overview of Intelligent Management

IBM Training

Overview of Intelligent Management

Figure 5-3. Overview of Intelligent Management

IBM Training

Intelligent Management

- Intelligent management provides application server virtualization, resource management, and advanced operations
- Used to enhance operations efficiency by
  - Managing available resources to meet the demands of high-volume transactional workloads
  - Managing large scale, continuously available application server environments
- Provides application infrastructure virtualization
  - Separates applications from the physical infrastructure on which they are hosted
  - Requests are intelligently routed to respond to the most critical applications and users
- Dynamic operations allow an application environment to scale as required by virtualizing WebSphere resources

Overview of Intelligent Management

Figure 5-4. Intelligent Management

Intelligent Management provides a virtualized infrastructure that redefines the traditional concepts of Java Platform, Enterprise Edition (Java EE) resources and applications and their relationships with one another. This application infrastructure virtualization facilitates the ability of the product to automate operations in an optimal manner, increasing the quality of service. By introducing an automated operating environment with workload management, you can reduce total cost of ownership by using less hardware to do more work.

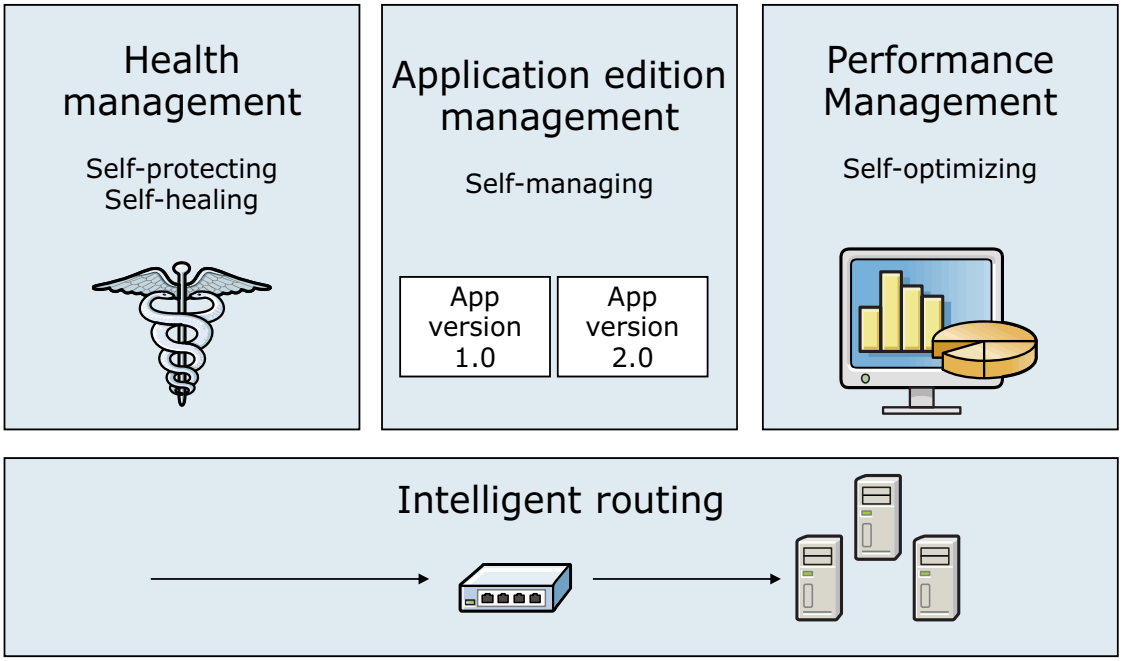
The dynamic operations environment consists of autonomic managers whose purpose is to use defined business goals to maximize utilization. Dynamic operations allow an application environment to scale as required by the virtualization of WebSphere resources and the use of a goals-directed infrastructure. Therefore, you can increase the speed at which your environment adapts to the business requirements. Using the dynamic operations features of WebSphere Application Server, you can change the way a typical WebSphere environment is configured to one that has the following features:

- Improves the utilization of available resources such as processor and memory
- Classifies and monitors the workload
- Provides a business-centric view of the workload and its performance

- Uses business guidelines that the organization specifies to respond in real time to changes in the workload mix (without human intervention if you choose)

IBM Training

Intelligent Management



Overview of Intelligent Management

Figure 5-5. Intelligent Management

Overview of intelligent routing capabilities:

- A routing tier that is aware of what is happening on the application server tier and reacts; business-critical applications are given priority
- Automatic routing without needing to update configuration files
- A highly scalable routing tier
- Ease of management
- Flexible policy-based routing to control if, when, and where requests are routed
- A highly available deployment manager

Overview of health management capabilities:

- Automatically detect and handle application health problems without requiring administrator time, expertise, or intervention
- Intelligently handle health issues in a way that maintains continuous availability
- Configure applications differently if you want to
- Approve automatic actions before the actions are taken

- Set policies that are based on both high-level metrics and low-level metrics to detect problems before you notice that something is wrong

Overview of application edition management capabilities:

- Ability to run multiple versions of your applications concurrently
- You do not incur any downtime while updating your applications
- Verify that a new version of your application works in production before sending real traffic to it
- Reduce your infrastructure costs and decrease potential outages
- Easily update your operating system or WebSphere without incurring downtime

Overview of performance management capabilities:

- Associate service policies with your applications and have WebSphere efficiently manage to achieve these goals
- Decrease administrative work that is required to monitor and diagnose performance issues
- Minimize the number of JVMs and virtual machines that are running to reduce processing that lightly used or idle JVMs or virtual machines incur
- Protect your middleware infrastructure against overload

IBM Training

History

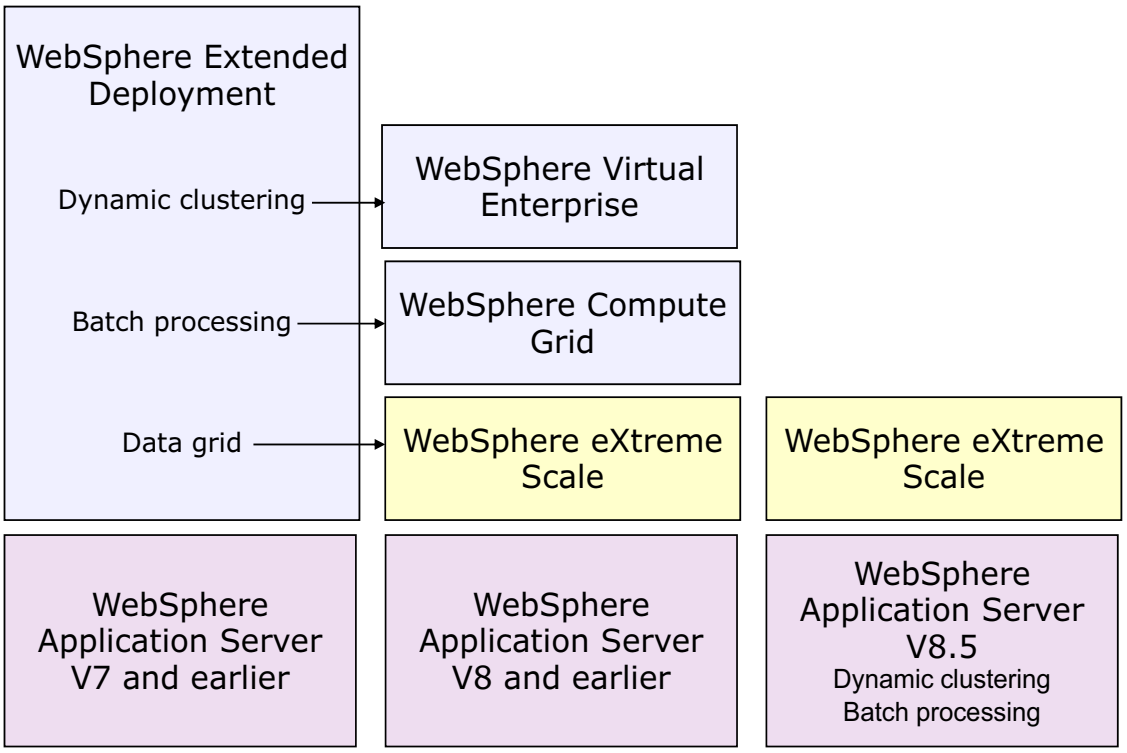


Figure 5-6. History

The biggest news under the theme of application resiliency is probably the integration of the features from WebSphere Virtual Enterprise into WebSphere Application Server Network Deployment. This merge allows a single WebSphere Application Server Network Deployment installation to deliver the traditional Network Deployment functions, and also WebSphere Virtual Enterprise functions. The former WebSphere Virtual Enterprise functions now in Network Deployment V8.5 are characterized as Intelligent Management features and encompass:

- Intelligent routing
- Application edition management
- Dynamic clustering
- Health management

5.2. Intelligent Management components

IBM Training

Intelligent Management components

Figure 5-7. Intelligent Management components

IBM Training

Intelligent Management components

- Dynamic clusters
  - Cluster members are dynamically created, started, and stopped
- Service policies
  - Define the business goals for application requests
- Autonomic managers and services
  - Provide information and take actions to implement intelligent management functions
- Intelligent routers
  - Support health management, application edition management, and Performance Management features
  - Two implementations
    - On demand router (ODR)
    - WebSphere plug-in

Figure 5-8. Intelligent Management components

Dynamic clusters allow the amount of resources available for application requests to be adjusted dynamically.

Service policies allow business goals to be used for decision making in how requests are handled.

Autonomic managers provide information and act to implement intelligent management functions.

Intelligent routers intelligently dispatch incoming requests to the application server tier.

IBM Training

Dynamic clusters

- A dynamic cluster is a cluster of servers where the number of active cluster members can change dynamically
- The number of cluster members available is based upon a node membership policy
  - Cluster members are created or deleted if a node that matches a membership policy is added to or removed from the cell
- The number of cluster members that are started is based upon current application demand and service policies
  - Cluster member weights and workload management are used to balance workload of cluster members
- Cluster member definitions are automatically updated when the server template associated with the dynamic cluster is updated

Figure 5-9. Dynamic clusters

A dynamic cluster is a server cluster that enables application server virtualization.

Members of a dynamic cluster are:

- Automatically created based on a membership policy
- Automatically updated by using a server template
- Automatically started and stopped based on current demand, available resources, and service policies

These features allow the application environment to dynamically expand and contract; it depends on the amount of workload that must be handled at any time.

IBM Training

Dynamic cluster settings

- Minimum number of cluster instances: Select to have one or more servers started always or stop all servers in times of inactivity
- Maximum number of cluster instances: Limit the number of servers that can start
- Vertical stacking of instances on a node: If you want to allow more than one server instance to be started on the same node
- Isolation requirements: Indicate whether a cluster member can run on the same node as cluster members from a different dynamic cluster

Figure 5-10. Dynamic cluster settings

A dynamic cluster is a virtual cluster of application servers that hosts an application. These application servers are on groups of nodes that are indicated by using the cluster membership policy. The membership policy is compared against the nodes in your cell, and servers are created for the dynamic cluster by using nodes that match the policy. When new nodes are added to your environment, they are added automatically to the dynamic cluster if they match the defined membership policy.

Service policies

- Service policies specify how to classify an incoming request by using the request's attributes
  - Such as URI, client name, or HTTP headers
- Easily allows an administrator to specify the relative importance of applications and optionally a response time goal
- Service policies are used to define application service level goals
- Allow workloads to be classified, prioritized, and intelligently routed
- Enables application performance monitoring
- Resource adjustments are made if needed to consistently achieve service policies

Figure 5-11. Service policies

A service policy is a user-defined categorization that is assigned to potential work as an attribute that the application request flow manager (ARFM) reads. You can use a service policy to classify requests that are based on request attributes. These attributes include the Uniform Resource Identifier (URI), the client name and address, HTTP headers, query parameters, cookies, and time of day. By configuring service policies, you apply varying levels of importance to the actual work. You can use multiple service policies to deliver differentiated services to different categories of requests.

IBM Training

Autonomic managers and services

- Autonomic request flow manager
  - Classifies incoming requests and monitors performance of service classes
- Dynamic workload management controller
  - Dynamically adjusts server weights to even out and minimize response times in a cluster
- Application placement controller
  - Decides on how many dynamic cluster members are started, and on which nodes the cluster members are started
- Health controller
  - Monitors defined health policies in the environment and ensures that actions are taken to correct problems
- On demand configuration service
  - Maintains cell topology information and keeps the other controllers informed of the environment

Figure 5-12. Autonomic managers and services

With Intelligent Management, you can introduce autonomic capabilities into your infrastructure at your own pace. Autonomic capabilities are delivered in a set of components that are known as autonomic managers. Autonomic managers monitor performance and health statistics through a series of sensors; they optimize system performance and run traffic shaping.

- Dynamic workload controller

  The dynamic workload controller dynamically adjusts server weights to even out and minimize response times across the cluster. There is one dynamic workload controller per cluster. The dynamic workload controller maintains a list of active server instances for each dynamic cluster, and assigns each a routing weight according to observed performance trends. Requests are then routed to candidate server instances to balance workloads on the nodes within a dynamic cluster that is based on a weighted least outstanding requests algorithm.

- The application placement controller

  The application placement controller is responsible for the management of the location of an application within a node group. A single application placement controller exists in the cell and is hosted in the deployment manager or in a node agent process. The application placement controller starts and stops application server instances to manage HTTP, SIP, JMS, and IIOP traffic. The application placement controller can dynamically address periods of intense workflow that would otherwise require the manual intervention of a system administrator.

- The on-demand configuration manager

  The on-demand configuration manager maintains cell topology information and keeps the ARFM and other controllers aware of its environment. It tracks updates in cell topology and state, including the following changes:

  - Applications that are installed and removed
  - Servers started and stopped
  - Nodes that added and removed
  - Classification updates

  The on-demand configuration component allows the on demand router to sense its environment. The on demand router dynamically configures the routing rules at run time to allow the on demand router to accurately route traffic to those application servers.

IBM Training

Intelligent routers: The on demand router (ODR)

- A Java based HTTP and SIP Proxy built on the WebSphere run time
- Typically runs in a tier between the web servers and the application servers
- Displayed in the administrative console as a new server type
- Can be clustered, highly available, and scalable
- Uses on demand configuration service to retrieve routing information
- Can route to multiple cells with failover or load-balancing of application requests across cells

Figure 5-13. Intelligent routers: The on demand router (ODR)

The on demand router classifies incoming HTTP and SIP requests and then works with other Intelligent Management “decision makers” to route the workload in order. The on demand router ensures that the highest priority is given to business-critical applications. Requests are prioritized and routed based on administrator-defined rules, called service policies, which are used to specify application response time goals.

The on demand router handles the queuing and dispatching of requests according to operational policy. An on demand router can be defined and started before any service policies are defined. Operational policies can be defined before the appearance of the work to which they apply. However, if policies are not defined, the default policies handle the early work. The on demand router, similar to the web server plug-in for WebSphere Application Server, uses session affinity to route work requests. After a session is established on a server, later work requests for the same session go to the original server. This configuration maximizes cache usage and reduces queries to resources. The on demand router accepts incoming requests and distributes these requests to the system in an intelligent manner, reflecting configured business goals. This process depends on the characterization of requests so that the relative business importance of each request can be compared.

Intelligent routers: The WebSphere plug-in

- Intelligent routing function based on native code ODRLIB implementation
- Added to existing supported web servers
- Can be clustered, highly available, and scalable
- Uses a RESTful web service to retrieve routing information
- Can route to multiple cells – unique applications in each cell only
- DMZ-ready

Figure 5-14. Intelligent routers: The WebSphere plug-in

Rather than a Java based server, the WebSphere plug-in intelligent router function is added to an existing supported web server. The WebSphere plug-in intelligent router has most of the same capabilities as the ODR, although they are implemented differently. Some benefits of the plug-in intelligent router are that it is DMZ-ready, and it eliminates the extra processing tier and network hop that the ODR intelligent router introduces.

What is intelligent routing?

- A routing tier that is aware of what is happening on the application server tier
  - Knows which cluster members are currently started
  - Knows application server utilization, request performance, and other statistics
  - Understands service policies
  - Routes work to the application server that can do it best
  - Knows which servers are in maintenance mode (more later)
  - Can route to multiple application editions (more later)
  - Provides preference for higher priority requests (ODR only)
  - Provides processor and memory overload protection (ODR only)
- Integrates with dynamic clustering, health management, and application edition management

Figure 5-15. What is intelligent routing?

Intelligent routing *can* improve the quality of service by ensuring that priority is given to business-critical applications and users. Requests to applications are prioritized and routed based on administrator-defined rules. It is easiest to start a discussion of Intelligent Management by referring to the most visible component, the on demand router (ODR). The ODR is a specialized, Java based proxy server that classifies incoming requests, and then dispatches the requests across the application server environment.

5.3. Health management

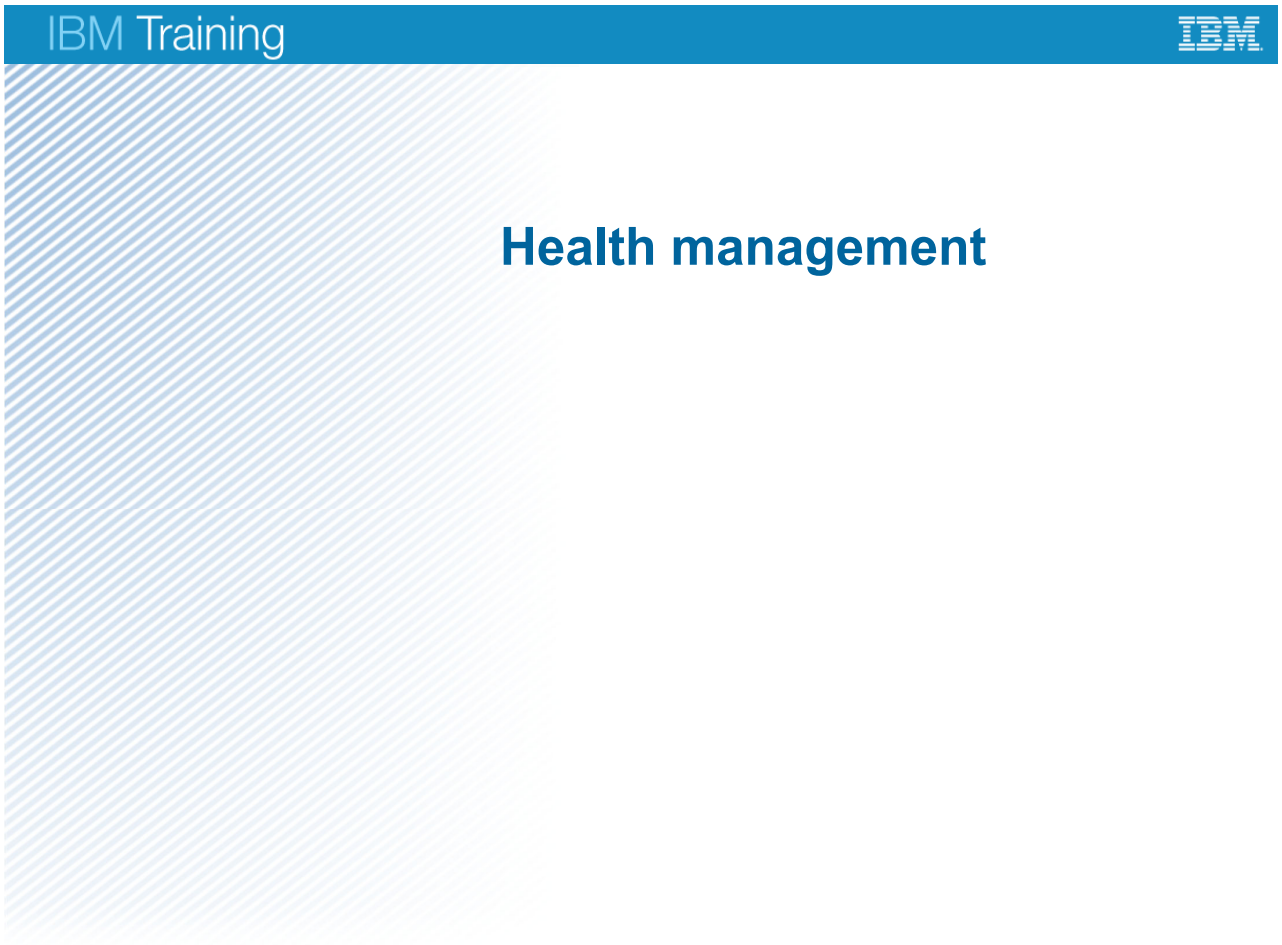


Figure 5-16. Health management

IBM Training

What is health management?

- A WebSphere environment can be monitored for various software health conditions
  - Age, work completed, memory usage, response time, and others
  - Excessive timeouts, storm drain detection
- Servers can have a custom sequence of steps that are run as a corrective or preventive action
  - Policy-driven autonomic system
- Health policies define monitored health conditions
- Customized health conditions and health actions can be defined

Figure 5-17. What is health management?

Intelligent Management provides a health management feature to monitor the status of your application servers to sense and respond to problem areas before an outage occurs. You can manage the health of your application environment with a policy-driven approach that enables specific actions to occur when monitored criteria are met. For example, when memory usage exceeds a percentage of the heap size for a specified time, health actions can run to correct the situation.

Health policies

- Health policies can be defined for common server health conditions
- A health policy defines a health condition, reaction, and targets
  - Condition: The *problem state* for which to look for
  - Reaction: The action to take when the condition is matched
  - Targets: The resources to monitor such as a single server, static or dynamic cluster, nodes or entire cell
- When a health policy condition is true, a corrective action runs automatically or requires approval
  - Notify administrator by sending email or SNMP trap
  - Capture diagnostics such as generate heap memory dump, Java core
  - Restart the application server in a way that prevents outages and service policy violations
- Each health policy can be in supervise or automatic mode

Figure 5-18. Health policies

WebSphere can monitor servers for common health problems and take corrective action. Various health conditions can be defined by using health policies. When a health policies violation is detected, an action plan can be put into effect automatically:

- Notify administrators (including email)
- Capture diagnostics information (Java thread or heap dump)
- Restart server

Application server restarts are done in a way to prevent outage and service policy violations. WebSphere provides the “First line of defense” for poor application health by mitigating common health problems and routing around unhealthy servers. IBM Tivoli Composite Application Manager for WebSphere extends WebSphere health management by adding in-depth application problem determination capabilities. To find out at a granular level what went wrong and how to fix it fast, IBM Tivoli Composite Application Manager gives support teams the diagnostic tools that they need. IBM Tivoli Composite Application Manager provides Rational and Eclipse developer and test tools with performance data captured in production, eliminating the need for attempting problem re-creation. IBM Tivoli Composite Application Manager integrates with the broader Tivoli Automation portfolio that enables customers to cost-effectively manage their IT infrastructure.

IBM Training

Viewing health conditions

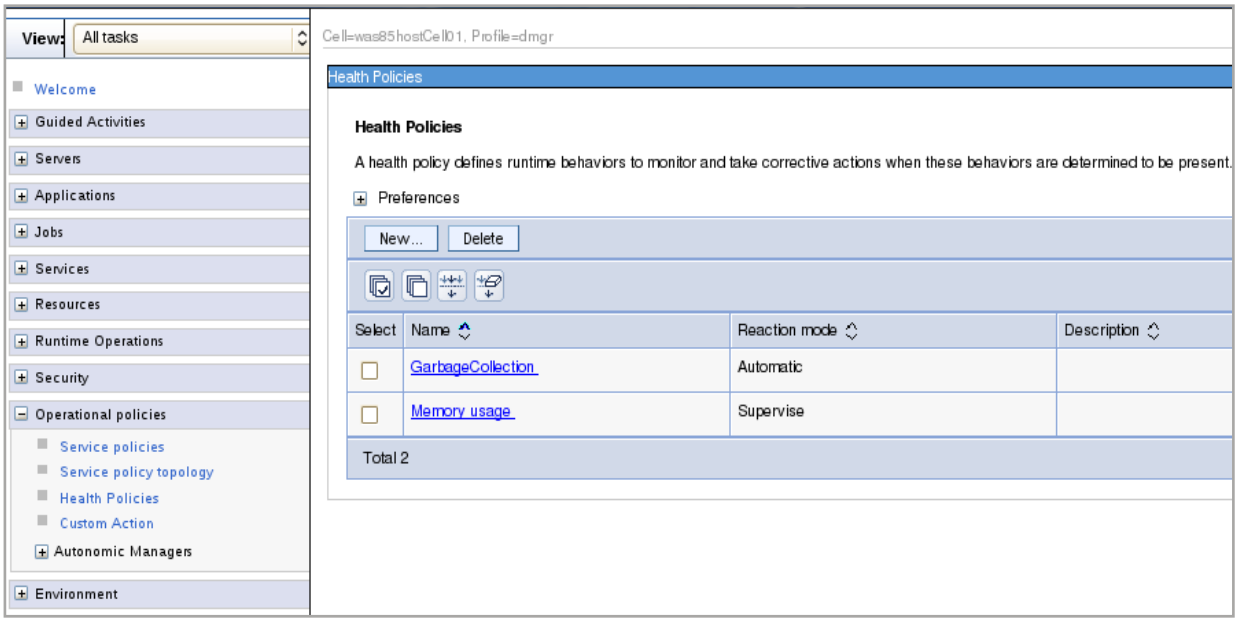


Figure 5-19. Viewing health conditions

You can use the administrative console to create health policies and view existing ones. Navigate to **Operational policies > Health Policies** to view any defined policies.

IBM Training

Predefined health conditions

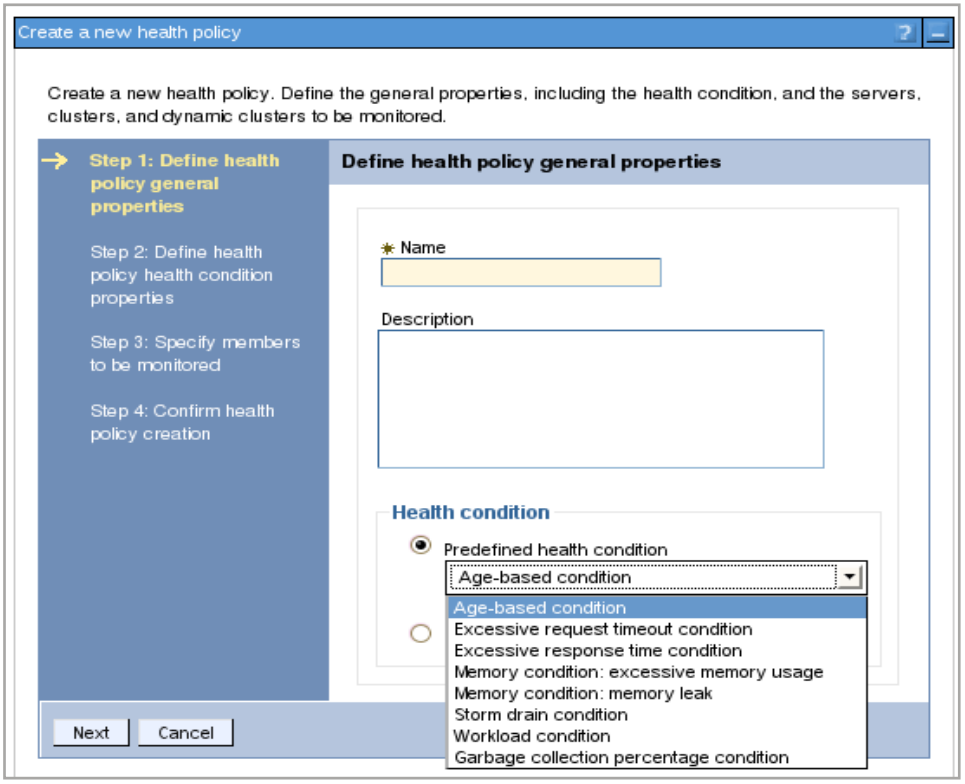


Figure 5-20. Predefined health conditions

Health conditions define the variables that you want to monitor in your environment. Several categories of health policy conditions exist. You can choose from the listing of predefined health conditions when creating a health policy.

Heath conditions

- Age-based: Amount of time the server is running
- Excessive conditions:
  - Excessive request timeout: Percentage of timed out requests
  - Excessive response time: Average response time
- Memory conditions:
  - Excessive memory usage: Percentage of maximum JVM heap size
  - Memory leak: JVM heap size after garbage collection
- Storm drain: Significant drop in response time
- Workload: Total number of requests
- Garbage collection: Percentage of time in garbage collection

Figure 5-21. Heath conditions

- Excessive garbage collection: Triggers when the Java virtual machine (JVM) spends more than a configured percentage of time when running garbage collections.
- Storm drain: Detects situations where requests are shifted toward a faulty cluster member that advertises low response times. This condition is triggered when there is a significant drop in the average response time. This drop must be measured at the on demand router, for a member of the cluster that is coupled with an increase in the dynamic weights for the cluster member.
- Workload: Triggers when the members that are associated with this policy serve a user-defined number of requests. You can use the workload condition on all server types.

Creating health conditions

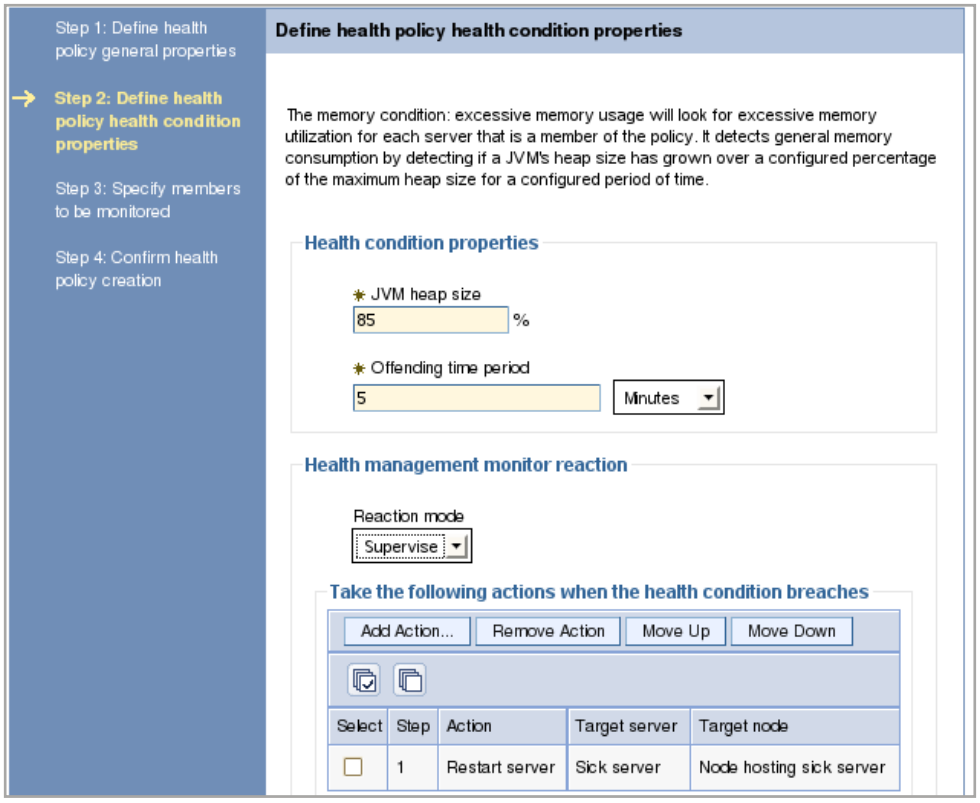


Figure 5-22. Creating health conditions

Actions can be taken automatically, or you can have them occur in supervised mode. Supervised mode requires an operator to confirm the action.

IBM Training

Predefined actions

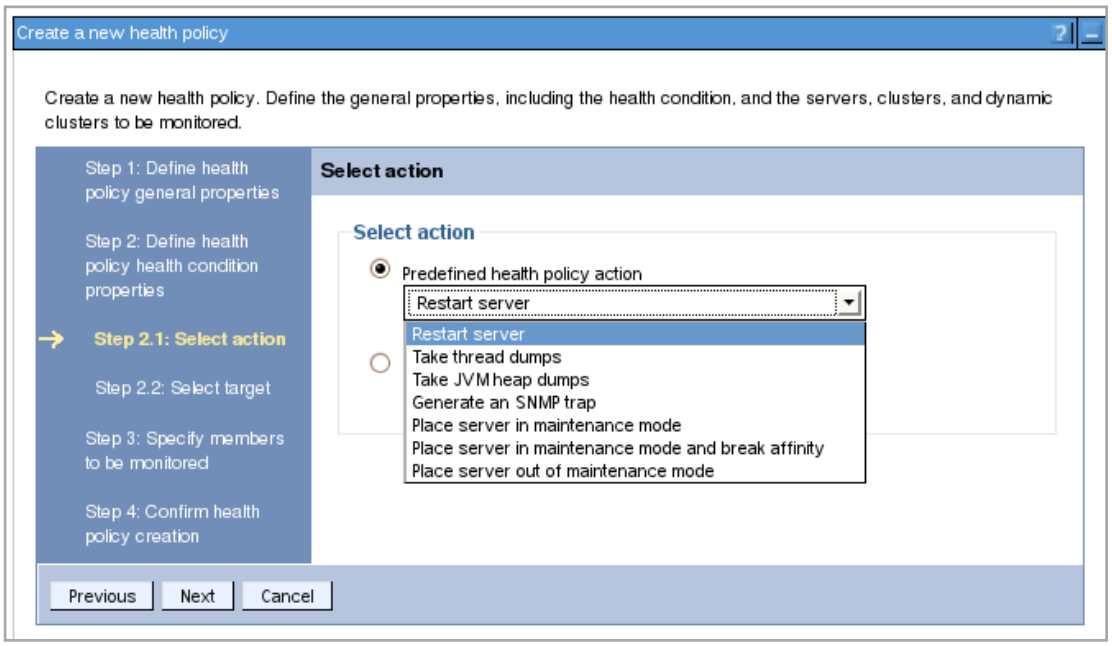


Figure 5-23. Predefined actions

When a health policy violation is detected, an action plan can be put into effect automatically. Actions to take when a monitored condition is detected are designed to bypass the problem and help in diagnosis. You can select the following predefined actions:

- Notifying an administrator
- Sending a Simple Network Management Protocol (SNMP) trap
- Restarting a server
- Putting a server into maintenance mode
- Generating Java cores or heap memory dumps for use in diagnosing the problem

Administering actions

Figure 5-24. Administering actions

If you chose the **Supervise** reaction mode, then you receive recommendations to improve your health conditions. These recommendations are displayed as runtime tasks that you can accept, deny, or close. To manage runtime tasks, click **System administration > Task management > Runtime tasks** in the administrative console. If you chose the **Automatic** reaction mode, actions to improve the health of your environment occur automatically.

IBM Training

Maintenance modes

- Allows you to update your environment without disrupting traffic to the production environment
- Servers or nodes are placed into maintenance mode which stops the routing from the intelligent routing tier
  - Application placement controller also excludes server or node from automatic application placement
  - Health controller uses the maintenance mode
- Node maintenance mode
  - Used to apply operating system fixes or provide WebSphere maintenance
  - Only traffic with affinity to servers on the node is routed to server
- Server maintenance mode
  - Perform server level problem determination
  - Modes to allow all traffic to the server, allow only traffic with affinity, or allow no traffic

Figure 5-25. Maintenance modes

Periodic product maintenance is important to ensure that your system environment works correctly, and to avoid trouble that known issues cause. At some point in time, you might have a problem with a server and find that diagnostic tests are necessary to troubleshoot a specific application server. These situations can lead to the disruption of client requests to servers in your environment.

Using the Intelligent Management feature, you can maintain the environment without disrupting traffic to the production environment. You can use it to administratively put a server or node in the cell into maintenance mode.

- Node maintenance mode:

  You can put a node into maintenance mode when you want to apply operating system fixes or do WebSphere maintenance. When a node is in maintenance mode, only traffic with affinity to servers on the node is routed to the server by the on demand router. You can set a maintenance immediate stop mode that immediately stops the servers on the node.

- Server maintenance mode:

  You can put a server into maintenance mode when you want to do server level problem determination. When an application server is placed into maintenance mode, you can indicate one of these modes:

  - Allow all traffic to the server

- Allow only traffic with affinity
- Allow no traffic during the maintenance period

Custom health conditions

- Enables you to create expressions that define what “unhealthy” means in your environment
- Custom expressions can be built which use metrics from:
  - The on demand router, URL return codes
  - PMI metrics, MBean operations, and attributes such as hung thread detection, connection pool exhaustion or slow down
  - And other metrics
- Complex Boolean expressions by using a mix of operands are supported (AND, OR, NOT)
- Provides flexibility by allowing the definition of custom actions that allow administrators to define an action plan to be carried out when the unhealthy situation detected

Figure 5-26. Custom health conditions

You can define custom conditions for your health policy if the predefined health conditions do not fit your needs. You define a custom condition as a subexpression that is tested against metrics in your environment. When you define a custom condition, consider the cost of collecting the data, analyzing the data, and if needed, enforcing the health policy. This cost can increase depending on the amount of traffic and the number of servers in your network. Analyze the performance of your custom health conditions before you use them in production.

For further information about creating custom conditions, go to the following website:

https://www.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.nd.doc/ae/cwve_hconditionsubex.html

5.4. Application edition management

IBM Training

Application edition management

Figure 5-27. Application edition management

IBM Training

What is Application edition management?

- Application versioning model that supports multiple deployments of the same application in the cell
  - The ability to upgrade applications without incurring outages or interruptions to users
- Concurrently run multiple editions of an application
  - Automatically route users to a specific application
- Includes an easy-to-use edition control center in administrative console, plus full scripting support
- Capabilities include:
  - Roll out policies to switch from one edition to another with no loss of service
  - Concurrent activation, where multiple editions can be concurrently active for an extended period
  - A validation mode to send selective traffic to verify correct operation

Overview of Intelligent Management

Figure 5-28. What is Application edition management?

Application edition management enables management of interruption-free production application deployments. Using this feature, you can validate a new edition of an application in your production environment without affecting users, and upgrade your applications without incurring user outages. You can also run multiple editions of a single application concurrently, directing different users to different editions, as the ODR maintains not only traditional application state (for example, HTTP session) affinity, but also application version affinity. The ability to queue requests is also employed with the Intelligent Management application edition function, if you want the following outcomes:

- An “atomic” application update that allows preprovisioning of a new application version
- An “atomic” update of all users from the old application version to the new application version

IBM Training

Terminology

- Application editions: Represents a unique instance of an application in the environment
  - Defined by the application name and an edition name
  - Might be a distinct build version
  - Might be the same build version with different deployment bindings (for example, resource references)
  - Might be both
- Edition name: Name of each edition of a particular application
- State: Identifies the status of the application edition
  - Active: Installed and available within a running application server
  - Inactive: Installed but not available
  - Validation: Used to selectively send traffic to an application server for testing or debugging purposes

Overview of Intelligent Management

Figure 5-29. Terminology

Application editions:

An application edition represents a unique instance of an application in the environment. An application edition encompasses both application versions and deployment bindings. An application edition is an application that is uniquely identified as the combination of an application name and an edition name.

Edition names and descriptions:

With the application edition manager, you can install multiple editions of the same application. Each edition is identified with an application edition name and description. The edition name is a field in which you can specify a value to uniquely identify one application edition from other editions of the same application. Create a version number scheme for naming editions that is meaningful in your environment. Multiple editions of the same application have the same application name but different edition names. When deploying an application, you can also specify an edition description next to the edition name, which gives you the ability to store more information.

Non-destructive update:

The existing application installation and update functions in Network Deployment are destructive. That is, they replace the old instance of the application with a new instance.

Installing an application edition is non-destructive. You can install any number of application editions and keep them in the system management repository.

IBM Training

Components

- Application edition manager
  - Interacts with the intelligent routers, dynamic workload manager, and application placement manager
- Application edition manager's edition control center
  - Provides control over the application update and rollout process, including edition activation across the application servers to which your application is deployed
  - Built into the administrative console

Figure 5-30. Components

The application edition manager ensures interruption-free production application deployments. Interruption-free deployment prevents loss of service when you install an application update in your environment.

The application edition manager provides an application versioning model that supports multiple deployments of the same application in the Intelligent Management cell. Each deployment has a unique edition name. The application edition manager allows you to select the edition to activate on an Intelligent Management cluster so that you can do a rollout of an application update or revert to a previous level.

The application edition manager is fully integrated with Intelligent Management, interacting with the on demand router (ODR), dynamic workload balancing, and the application placement manager. This integration ensures predictable application behavior when you apply application updates, and a smooth transition from one application edition to another while the system continues to manage your application performance goals. You can access application update processes with the administrative console, including edition activation across the application servers.

Rollout activation (1 of 2)

- Activates one edition in place of another edition of an application
- Soft or hard rollout
  - Soft rollout starts only the application
  - Hard rollout stops and starts the application server
- Atomic or group rollout
  - Atomic rollout guarantees that two editions do not service requests at the same time and queues requests
  - Group rollout might have two editions service requests at the same time; and it does not queue requests
- Drainage interval defines the maximum amount of time the application edition manager waits for sessions to expire before stopping an application server

Figure 5-31. Rollout activation (1 of 2)

Rollout activation activates one edition in place of another, ensuring an interruption-free update in the process. Thus, all application requests are serviced during the rollout, and none are lost. This process ensures continuous application operation from the perspective of the customers of that application. To do the rollout, the application edition manager carefully coordinates the activation of the edition and the routing of requests to the application.

Rollout activation (2 of 2)

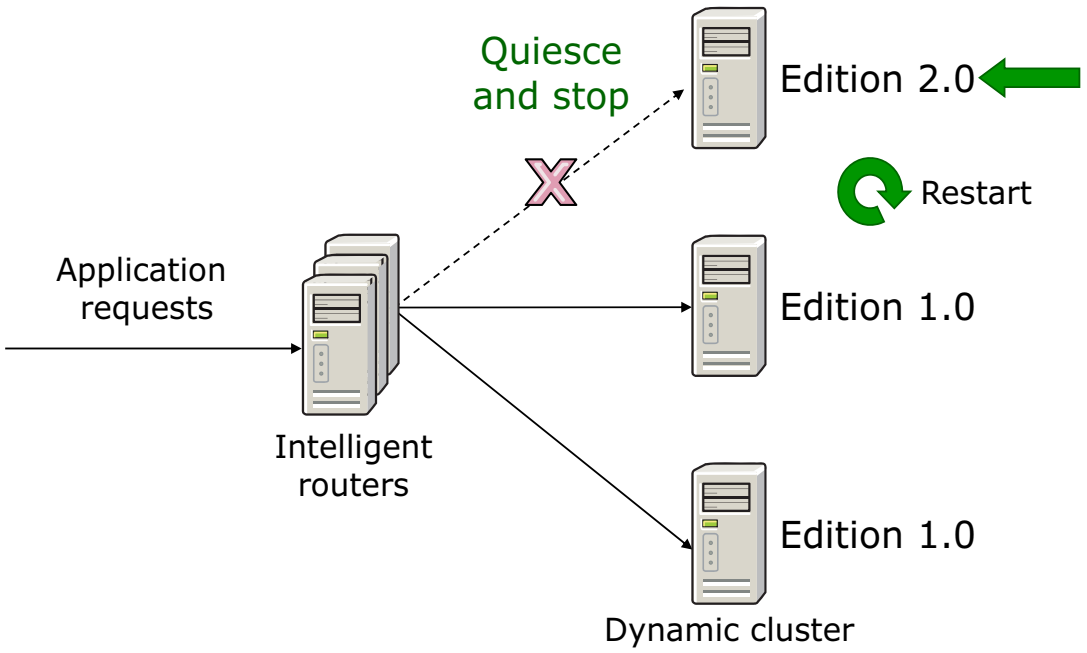


Figure 5-32. Rollout activation (2 of 2)

Replacement of one edition with another in a production environment requires certain discipline in the evolution of the application. Because edition replacement happens while application users are potentially accessing the previous application edition, the new edition must be compatible with earlier versions. Thus, the new edition cannot add or change any existing application interfaces, including essential behavior. New interfaces can be added. In addition, existing interfaces can be algorithmically corrected and, in some cases, even extended and remain compatible with existing application users.

This slide displays an example of a group rollout scenario. In the diagram, a dynamic cluster is created that consists of three servers. You first must divide the cluster into groups, which tells the application edition manager how many servers to update at the same time. If you do a rollout on a group, the servers in each group are upgraded to the new edition at the same time. Each server in the group is quiesced, stopped, and reset.

As the rollout is run in the diagram on the slide, one server in the cluster is moved from Edition 1.0 to Edition 2.0. During this time, the server does not receive user requests that are directed from the on demand router, and the server is stopped. All application requests are sent to the servers that are running Edition 1.0. After the server that is running Edition 2.0 is available, the on demand router directs application requests to that server. Any servers that are still running Edition 1.0 do not serve requests until the edition is updated to Edition 2.0.

Concurrent activation

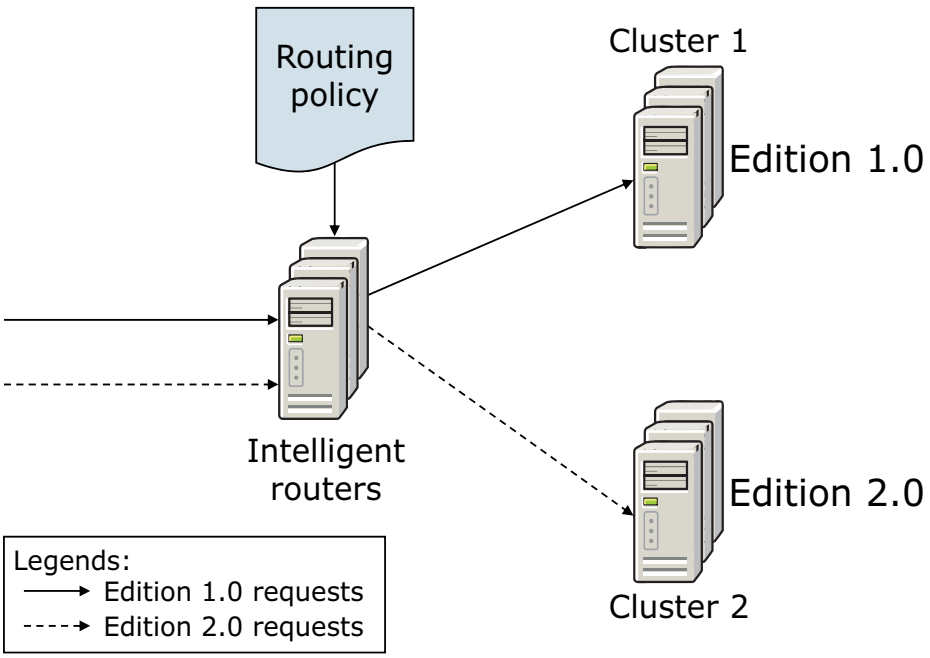


Figure 5-33. Concurrent activation

You can use concurrent activation to activate the same edition on different servers or clusters. To use multiple editions concurrently, you must distinguish user requests from one another so that the requests are sent to the application server that hosts the appropriate edition. For example, if you introduce a new edition of an application, you might want only a select group of users to test the edition.

IBM Training

Validation mode

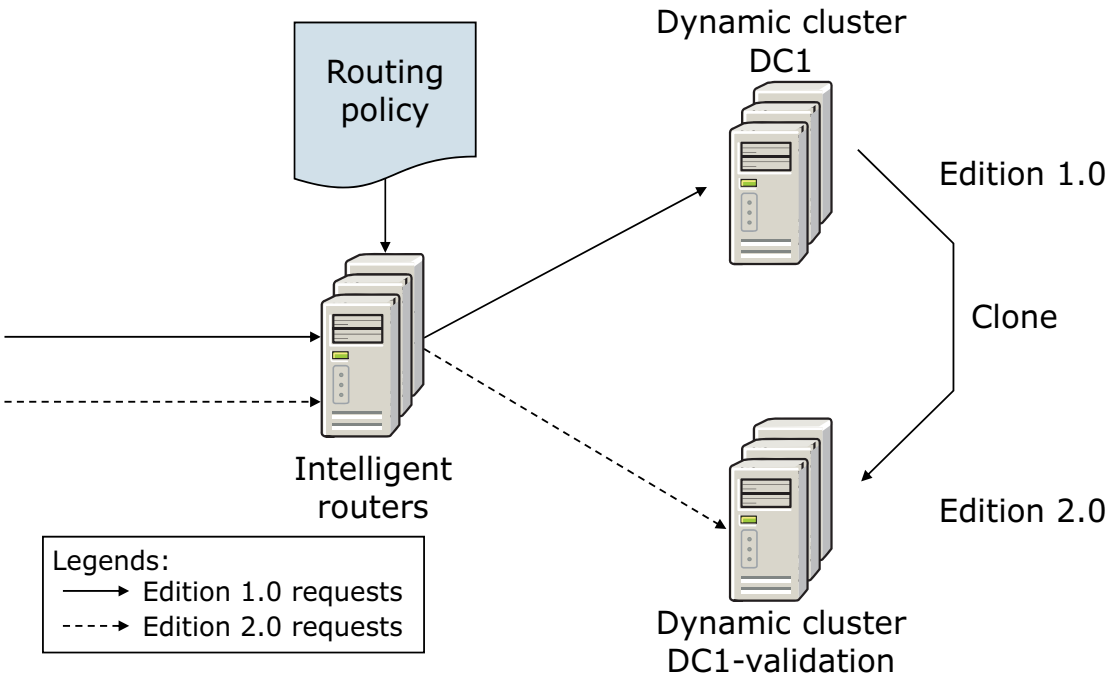


Figure 5-34. Validation mode

Validation activation is a special form of concurrent activation. It activates an edition on a clone of its original deployment target. The clone is created on activation of the edition. After the validation rollout to the original deployment target, the clone is removed automatically. This action allows you to do final preproduction testing of an application edition in the actual production environment with a selected set of users.

IBM Training

Edition control center (1 of 2)

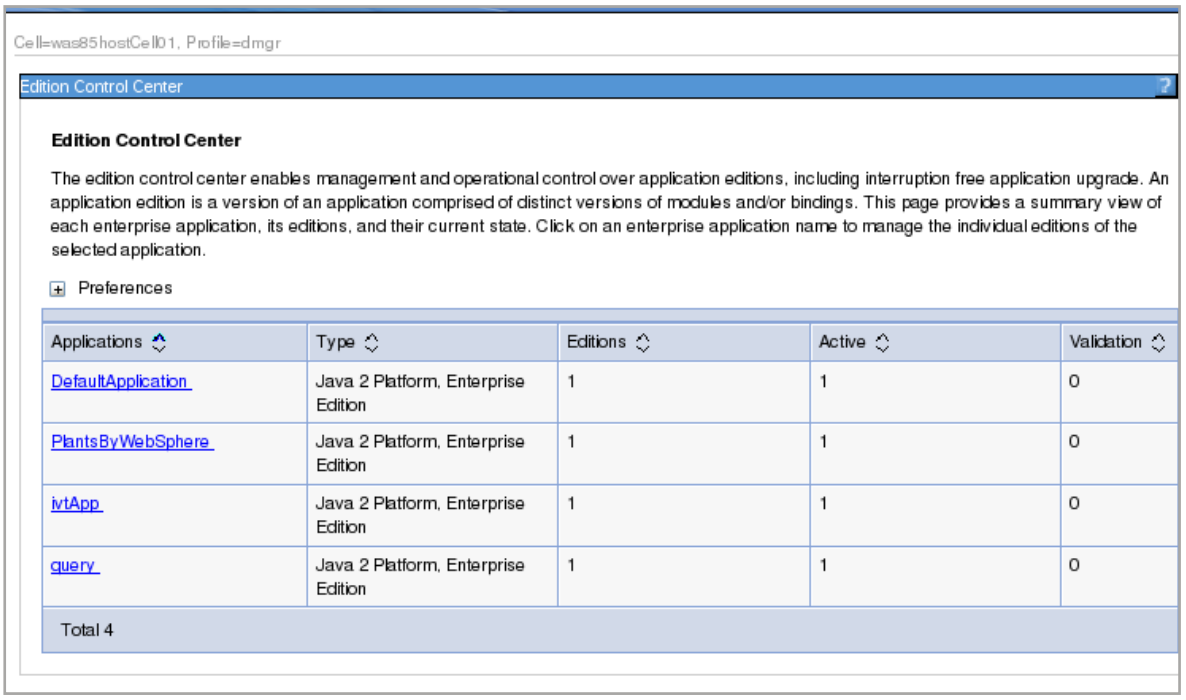


Figure 5-35. Edition control center (1 of 2)

To use the administrative console to validate the results, click **Applications > Edition Control Center > *application_name***.

IBM Training

Edition control center (2 of 2)

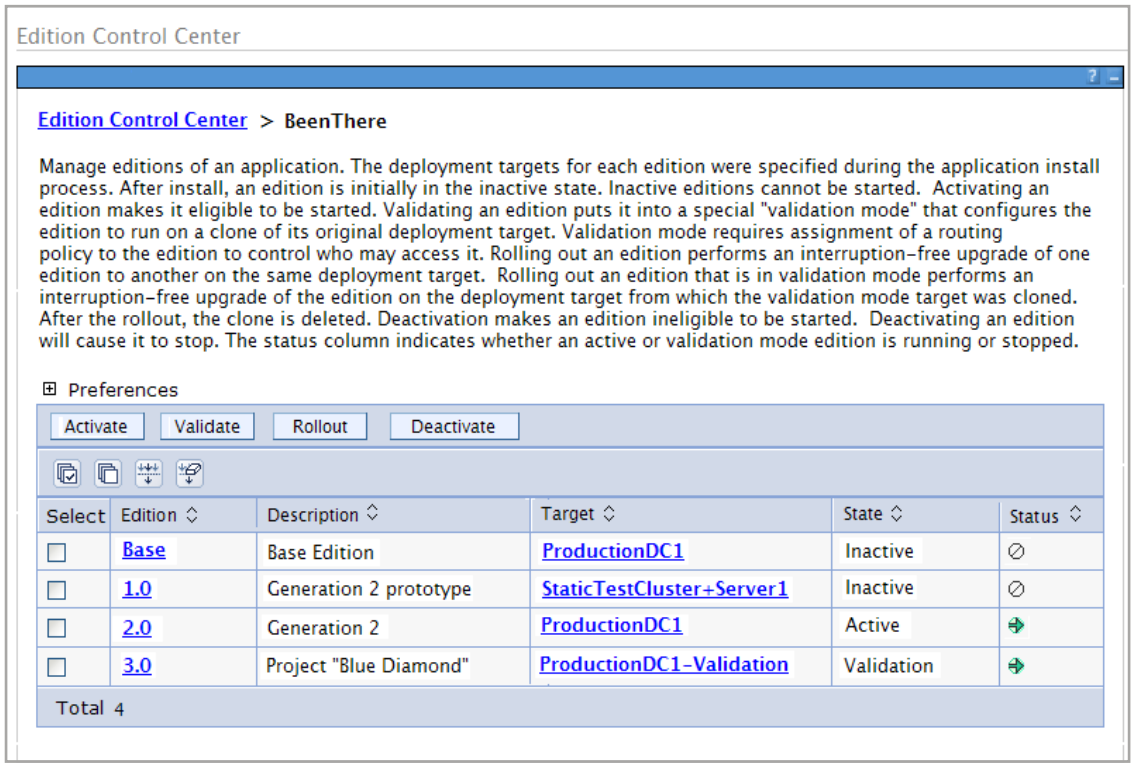


Figure 5-36. Edition control center (2 of 2)

This slide displays editions for the BeenThere application.

5.5. Performance Management

IBM Training

Performance Management

Figure 5-37. Performance Management

What is Performance Management?

- Provides a self-optimizing middleware infrastructure
- Ability to improve performance by using dynamic clustering and overload protection
- Dynamic clusters are used to scale up and scale down running cluster members to meet response time goals
- Overload protection limits the rate at which the on demand router forwards traffic to application servers
  - Prevents heap exhaustion, processor exhaustion, or both from occurring

Figure 5-38. What is Performance Management?

The performance management feature provides dynamic cluster capabilities and overload control. With dynamic clusters, you can automatically scale up and scale down the number of running cluster members as needed to meet response time goals for your users. You can use overload protection to limit the rate at which the on demand router forwards traffic to application servers. Doing so helps prevent heap exhaustion, processor exhaustion, or both from occurring.

5.6. Deployment manager high availability

IBM Training

Deployment manager high availability

Figure 5-39. Deployment manager high availability

IBM Training

Highly available deployment manager (1 of 3)

- WebSphere supports running multiple deployment managers for high availability when using the ODR intelligent router
- Multiple deployment managers can be configured
  - Active: The primary deployment manager that hosts the administrative functions
  - Standby: Waiting to take over if the active deployment manager fails
- One deployment manager is active, others run in standby mode until a failure is detected
- All deployment managers share master configuration repository and workspaces that are stored in a shared file system that supports fast lock recovery
- The on demand router routes traffic to the active deployment manager

Figure 5-40. Highly available deployment manager (1 of 3)

Although it is not required to have a deployment manager that is always running, you might require highly available administrative capability. This need occurs especially in environments that have a significant number of new application deployments or updates and server monitoring. Multiple instances of a deployment manager remove the single point of failure (SPOF) for cell administration, thus assuring the attainability of the administrative console, wsadmin, and scripting features to manage your environment. WebSphere Application Server provides a mechanism for cloning your existing deployment manager, thus achieving high availability, by employing redundant deployment managers with a hot-standby model and the use of a shared file system.

Highly available deployment manager (2 of 3)

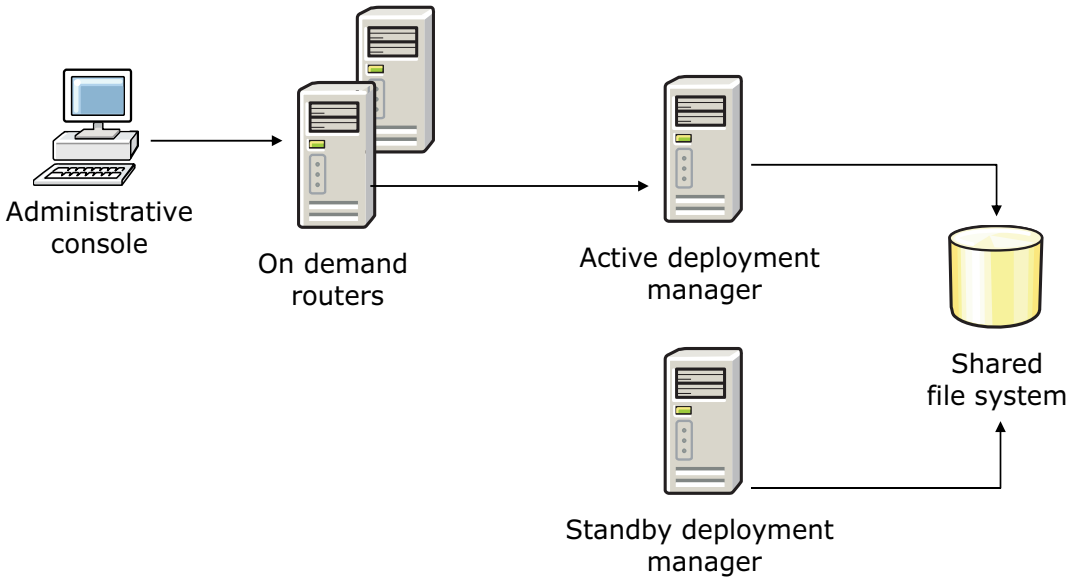


Figure 5-41. Highly available deployment manager (2 of 3)

In this paradigm, one of the deployment managers is elected as primary. As primary, it is considered an active deployment manager that is hosting the cell-wide endpoints for the administrative functions. Other deployment managers are considered backups; they are kept in standby mode and are available to take over the active role in case of failure or termination of the primary manager.

IBM Training

Highly available deployment manager (3 of 3)

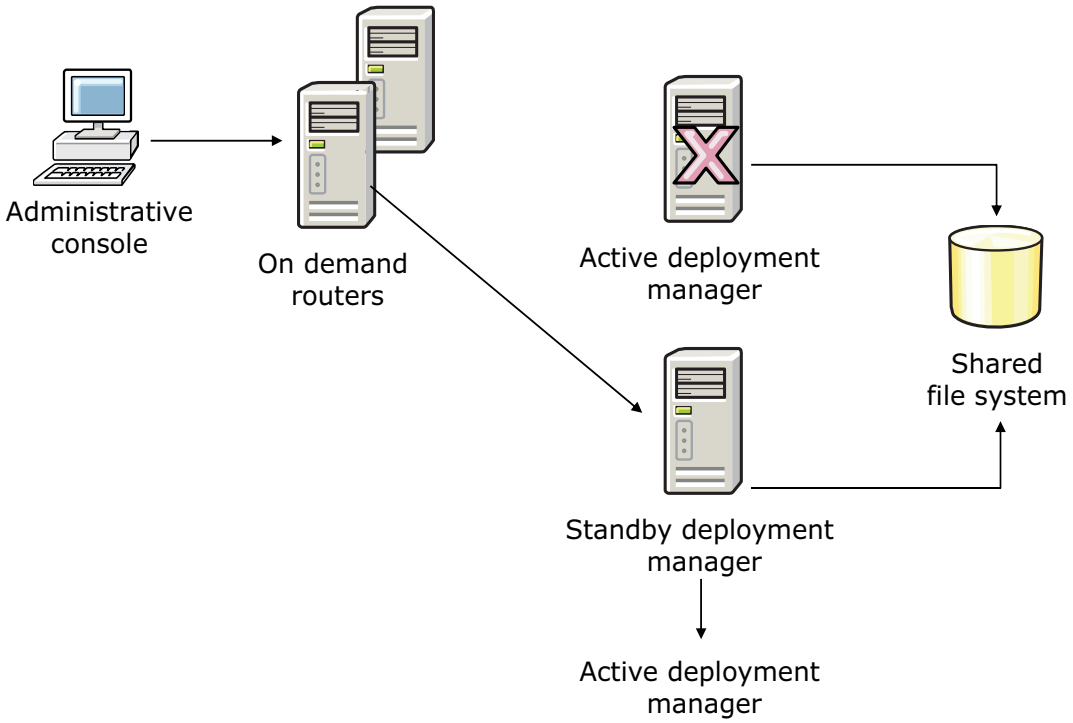


Figure 5-42. Highly available deployment manager (3 of 3)

A highly available deployment manager component runs in each deployment manager to control which deployment manager is elected as the active one.

IBM Training

Unit summary

- Define Intelligent Management
- Describe virtualization and autonomic computing
- Define intelligent routing
- Describe dynamic workload management
- Describe the health management features of Intelligent Management
- Describe the application edition management features of Intelligent Management
- Describe the performance management features of Intelligent Management

Figure 5-43. Unit summary

Review questions

1. True or False: WebSphere provides the ability for a hot standby for a deployment manager to be highly available.

2. Which intelligent management component maintains cell topology information and keeps the other controllers informed of the environment?
   A. Dynamic workload controller
   B. Autonomic request flow manager
   C. On demand configuration service

3. True or False: An on demand router can route requests within the cell only.

Figure 5-44. Review questions

Write your answers here:

1.

2.

3.

IBM Training

Review answers

1. True or False: WebSphere provides the ability for a hot standby for a deployment manager to be highly available.
   The answer is True. WebSphere provides the ability for a hot standby for a deployment manager to be highly available, but only when using the on demand router intelligent router.
2. Which intelligent management component maintains cell topology information and keeps the other controllers informed of the environment?
   A. Dynamic workload controller
   B. Autonomic request flow manager
   C. On demand configuration service
   The answer is C.
3. True or False: An on demand router can route requests within the cell only.
   The answer is False. An on demand router can route requests within the cell and can route to multiple cells.

Figure 5-45. Review answers

Unit 6. Course summary

Estimated time

00:15

Overview

This unit summarizes the course and provides information for future study.

IBM Training

Unit objectives

- Explain how the course met its learning objectives
- Access the IBM Training website
- Identify other IBM Training courses that are related to this topic
- Locate appropriate resources for further study

Figure 6-1. Unit objectives

IBM Training

Course objectives

- Describe the architectural concepts that are related to WebSphere Application Server Network Deployment
- Create a deployment manager instance
- Federate an application server to a cell
- Add a stand-alone application server to a WebSphere Application Server cell
- Cluster an application server within a WebSphere Application Server cell
- Configure WebSphere Application Server SSL security settings
- Deploy applications in clustered environments

Figure 6-2. Course objectives

IBM Training

Course objectives

- Describe the features of Intelligent Management

Figure 6-3. Course objectives

IBM Training

To learn more on the subject

- IBM Training website:
  www.ibm.com/training
- IBM Redbooks:
  www.redbooks.ibm.com
- IBM Knowledge Center for IBM WebSphere Application Server traditional V9:
  https://www.ibm.com/support/knowledgecenter/SSEQTP_9.0.0/as_ditamaps/was900_welcome_base.html
- IBM support:
  http://www.ibm.com/support/en-us/?lnk=hmmsu
- Product information:
  http://www.ibm.com/software/products/en/appserv-was

  https://www.ibm.com/common/ssi/ShowDoc.wss?docURL=/common/ssi/rep_ca/4/877/ENUSZP16-0344/index.html&lang=en&request_locale=en

Figure 6-4. To learn more on the subject

IBM Training

Enhance your learning with IBM resources

Keep your IBM Cloud skills up-to-date

- IBM offers resources for:
  - Product information
  - Training and certification
  - Documentation
  - Support
  - Technical information

- To learn more, see the IBM Cloud Education Resource Guide:
  - www.ibm.biz/CloudEduResources

Course summary

Figure 6-5. Enhance your learning with IBM resources

IBM Training

Unit summary

- Explain how the course met its learning objectives
- Access the IBM Training website
- Identify other IBM Training courses that are related to this topic
- Locate appropriate resources for further study

Figure 6-6. Unit summary

IBM Training

Course completion

You have completed this course:

WebSphere Application Server V9 Administration in a Federated Environment

Any questions?

Figure 6-7. Course completion

Appendix A. List of abbreviations

A

API	application programming interface
ARFM	application request flow manager

B

C

CA	certificate authority
CIM	Centralized installation manager
CN	common name

D

DMZ	demilitarized zone
DNS	Domain Name System
DRS	Data Replication Service

E

EAR	enterprise archive
EE	Enterprise Edition
EJB	Enterprise JavaBean
ESB	Enterprise service bus

F

FQDN	Fully qualified domain name
FTP	File transfer protocol

G

GCD	greatest common divisor
GUI	graphical user interface

H

HA	high availability or highly available
HTTP	Hypertext Transfer Protocol
HTTPD	HTTP Daemon

HTTPS	HTTP over SSL

I

IBM	International Business Machines Corporation
IE	Internet Explorer
IIM	IBM Installation Manager
IIOP	Internet Inter-ORB Protocol
I/O	input/output
IP	Internet Protocol

J

Java EE	Java Platform, Enterprise Edition
JCL	Java class library
JDBC	Java Database Connectivity
JDK	Java Development Kit
JMS	Java Message Service
JMX	Java Management Extensions
JNDI	Java Naming and Directory Interface
JPA	Java Persistence API
JSP	JavaServer Pages
JVM	Java virtual machine

K

L

LDAP	Lightweight Directory Access Protocol

M

MBean	Managed Bean (Managed Java object)

N

O

ODR	on demand router
ORB	Object Request Broke
OS	operating system

P

PKI	public key infrastructure
PME	programming model extensions
PMI	Performance Monitoring Infrastructure
PMR	problem management record
PMT	Program Management Tool
POJO	plain old Java object

Q

R

RA	Registration authority
REST	Representational State Transfer
RMI	Remote Method Invocation

S

SIP	Session Initiation Protocol
SNMP	Simple Network Management Protocol
SOAP	A lightweight, XML-based protocol for exchanging information in a decentralized, distributed environment. Usage note: SOAP is not an acronym; it is a word in itself (formerly an acronym for Simple Object Access Protocol)
SPOF	single point of failure
SSL	Secure Sockets Layer

T

U

URI	Uniform Resource Identifier
URL	Uniform Resource Locator

V

VPN	virtual private network

W

WCT	WebSphere Customization Toolbox
WLM	workload management

X

XML	Extensible Markup Language

Y

Z

z/OS	zSeries operating system

Appendix B. Resource guide

Completing this IBM Training course is a great first step in building your IBM Middleware skills. Beyond this course, IBM offers several resources to keep your Middleware skills on the cutting edge. Resources available to you range from product documentation to support websites and social media websites.

Training

- **IBM Training website**
  - Bookmark the IBM Training website for easy access to the full listing of IBM training curricula. The website also features training paths to help you select your next course and available certifications.
  - For more information, see: http://www.ibm.com/training
- **IBM Training News**
  - Review or subscribe to updates from IBM and its training partners.
  - For more information, see: http://www.ibm.com/blogs/ibm-training
- **IBM Certification**
  - Demonstrate your mastery of IBM Middleware to your employer or clients through IBM Professional Certification. Middleware certifications are available for developers, administrators, and business analysts.
  - For more information, see: http://www.ibm.com/certify
- **Training paths**
  - Find your next course easily with IBM training paths. Training paths provide a visual flow-chart style representation of training for many IBM products and roles, including developers and administrators.
  - For more information, see: http://www-304.ibm.com/jct03001c/services/learning/ites.wss/us/en?pageType=page&c=a0003096

Social media links

Connect with IBM Middleware Education and IBM Training, and learn about the latest courses, certifications, and special offers by seeing any of the following social media websites.

- **Twitter**
  - Receive concise updates from Middleware Education a few times each week.
  - Follow Middleware Education at: twitter.com/IBMCloudEdu

- **Facebook**:
  - Follow IBM Training on Facebook to keep in sync with the latest news and career trends, and to post questions or comments.
  - Find IBM Training at: facebook.com/ibmtraining
- **YouTube**:
  - See the IBM Training YouTube channel to learn about IBM training programs and courses.
  - Find IBM Training at: youtube.com/IBMTraining

Support

- **Middleware Support portal**
  - The Middleware Support website provides access to a portfolio of downloadable support tools, including troubleshooting utilities, product updates, drivers, and Authorized Program Analysis Reports (APARS). The Middleware Support website also provides links to online Middleware communities and forums for collaboratively solving issues. You can now customize the IBM Support website by adding or deleting portlets to show the most important information for the IBM products that you work with.
  - For more information, see: http://www.ibm.com/software/websphere/support
- **IBM Support Assistant**
  - The IBM Support Assistant is a local serviceability workbench that makes it easier and faster for you to resolve software product issues. It includes a desktop search component that searches multiple IBM and non-IBM locations concurrently and returns the results in a single window, all within IBM Support Assistant.
  - IBM Support Assistant includes a built-in capability to submit service requests; it automatically collects key problem information and transmits it directly to your IBM support representative.
  - For more information, see: http://www.ibm.com/software/support/isa
- **IBM Education Assistant**
  - IBM Education Assistant is a collection of multimedia modules that are designed to help you gain a basic understanding of IBM software products and use them more effectively. The presentations, demonstrations, and tutorials that are part of the IBM Education Assistant are an ideal refresher for what you learned in your IBM Training course.
  - For more information, see: http://www.ibm.com/software/info/education/assistant/

Middleware documentation and tips

- **IBM Redbooks**
  - The IBM International Technical Support Organization develops and publishes IBM Redbooks publications. IBM Redbooks are downloadable PDF files that describe

installation and implementation experiences, typical solution scenarios, and step-by-step “how-to” guidelines for many Middleware products. Often, Redbooks include sample code and other support materials available as downloads from the site.

  - For more information, see: http://www.ibm.com/redbooks

- **IBM documentation and libraries**
  - IBM Knowledge Centers and product libraries provide an online interface for finding technical information on a particular product, offering, or product solution. The IBM Knowledge Centers and libraries include various types of documentation, including white papers, podcasts, webcasts, release notes, evaluation guides, and other resources to help you plan, install, configure, use, tune, monitor, troubleshoot, and maintain Middleware products. The Knowledge Center and library are located conveniently in the left navigation on product web pages.
- **developerWorks**
  - IBM developerWorks is the web-based professional network and technical resource for millions of developers, IT professionals, and students worldwide. IBM developerWorks provides an extensive, easy-to-search technical library to help you get up to speed on the most critical technologies that affect your profession. Among its many resources, developerWorks includes how-to articles, tutorials, skill kits, trial code, demonstrations, and podcasts. In addition to the Middleware zone, developerWorks also includes content areas for Java, SOA, web services, and XML.
  - For more information, see: http://www.ibm.com/developerworks

Services

- IBM Software Services for Middleware are a team of highly skilled consultants with broad architectural knowledge, deep technical skills, expertise on suggested practices, and close ties with IBM research and development labs. The Middleware Services team offers skills transfer, implementation, migration, architecture, and design services, plus customized workshops. Through a worldwide network of services specialists, IBM Software Service for Middleware makes it easy for you to design, build, test, and deploy solutions, helping you to become an on-demand business.
- For more information, see: http://www.ibm.com/services/us/en/it-services/systems/middleware-services/

IBM Training