

Chapter 1 - Web Server Management and Cluster Topology

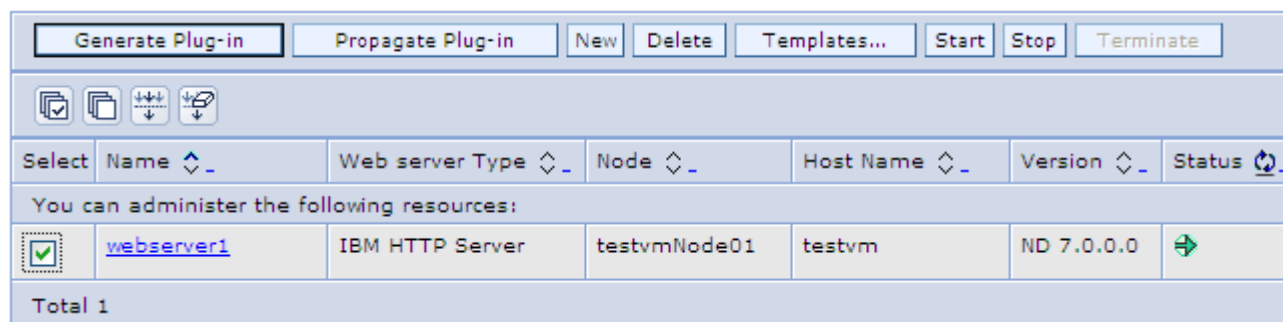
Objectives

At the end of this chapter, participants will be able to understand:

- Web server management options provided by Network Deployment
- Clustered Application Servers
- Cluster creation and management
- Types of scaling
- Topology Considerations
- Common Topologies

1.1 Web Server Management

- Network Deployment has abilities to manage web servers
- The abilities depend on two factors
 - ◇ Is the web server an IBM HTTP Server?
 - ◇ Is there a Node Agent "local" to the web server machine?
- Using the IBM HTTP Server presents the widest range of abilities



The screenshot shows the Network Deployment administration console interface. At the top, there are several buttons: "Generate Plug-in", "Propagate Plug-in", "New", "Delete", "Templates...", "Start", "Stop", and "Terminate". Below these buttons are icons for selection, copy, and refresh. The main area contains a table with columns: "Select", "Name", "Web server Type", "Node", "Host Name", "Version", and "Status". The table lists one resource: "webserv1" of type "IBM HTTP Server" on node "testvmNode01" with host name "testvm" and version "ND 7.0.0.0". The status is indicated by a green arrow icon. At the bottom, it says "Total 1".

Select	Name	Web server Type	Node	Host Name	Version	Status
<input checked="" type="checkbox"/>	webserv1	IBM HTTP Server	testvmNode01	testvm	ND 7.0.0.0	

Total 1

Web Server Management

If you are using IBM HTTP Server (IHS) as the web server, you can perform basic management of it from admin console. IHS is essentially the Apache web server. So, you can also directly edit the httpd.conf file. The web server administrative activities in admin console is purely a convenience feature. You can view the status of and start stop a large number of web server without logging into different machines.

If you are not using IHS as the web server, you can not perform most of the administrative tasks. You must still create a web server in admin console. Because a web application must be targeted to a web server for the application to be added to the plugin-config.xml of that web server.

1.2 Administering IBM HTTP Server

- You can do these activities from admin console:
 - ◇ Start and stop remote web server
 - ◇ Automatically propagate plug-in configuration
 - ◇ Edit web server configuration file
 - ◇ View log files
 - ◇ View plug-in log
- These abilities utilize the IBM HTTP Administration process running on the remote machine
- The web server is defined in an "unmanaged" Node

Using IBM HTTP Server

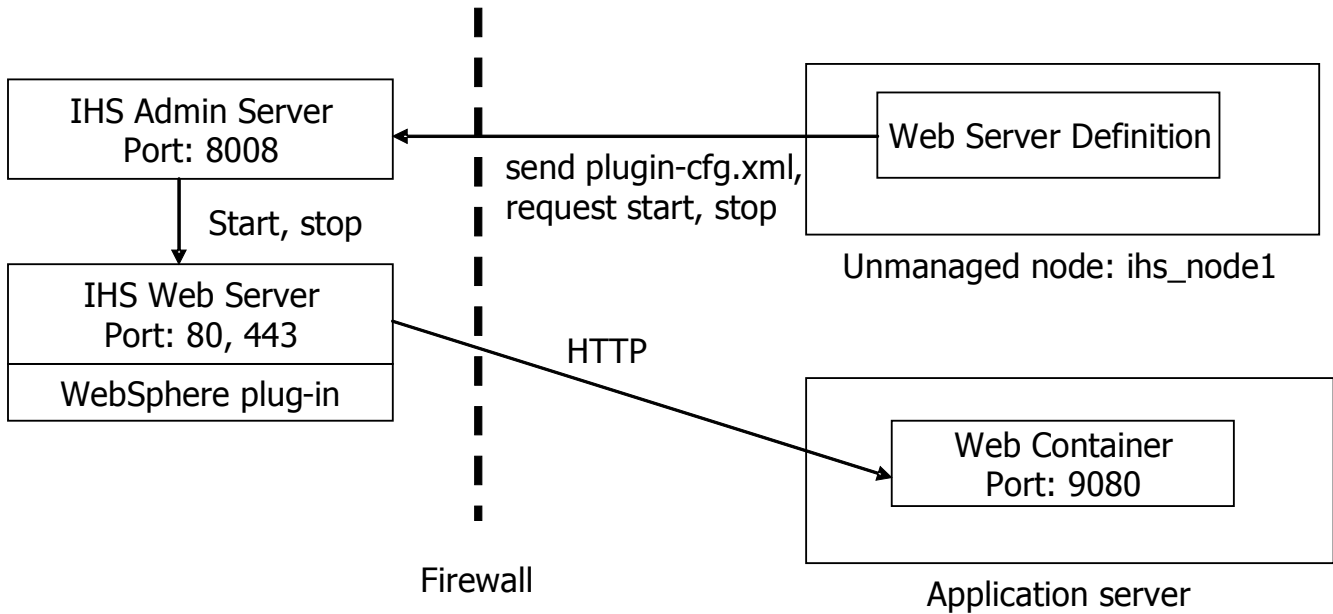
An "unmanaged" Node simply means there is no Node Agent. The Node is defined simply to provide a separate definition for the web server.

Automatically configure WebSphere

When the plug-in is installed in the web server machine, a script is created in the <PLUGIN_ROOT>\bin folder. The name of the script is configureWEBSERVER_NAME.bat. Where, WEBSERVER_NAME is the name of the web server as it was entered during the plug-in installation. A script for UNIX is also created in <PLUGIN_ROOT>/bin/crossPlatformScripts called configureWEBSERVER_NAME.sh. Copy the appropriate script to <WAS_ROOT>/bin folder in the Deployment Manager machine.

This script loads a JACL script file called configureWebserverDefinition.jacl using the wsadmin command. Run the script. This will automatically configure an unmanaged node, a web server and deploy all existing web modules to the web server.

1.3 IBM HTTP Server Architecture



IBM HTTP Server Architecture

The diagram above shows how IBM HTTP Server (IHS) is integrated with WebSphere. In the web server machine, IHS and WebSphere web server plug-in for IHS is installed. In this machine, the web server is started. The web server administration server also needs to be started. Note that the administration server is only available from IHS and not from the default Apache distribution.

In the WebSphere side of things, an unmanaged node called `ihs_node1` is defined. Then a web server is defined in that node. These definitions contain enough information so that WebSphere can communicate with IHS admin server (such as host name, port number, administrative user ID and password).

The picture above can be misunderstood to imply there are two machines involved in running the web server. This is not true. The unmanaged node and web server definition on the right are simply logical definitions in the WebSphere configuration to represent the machine running the web server.

Using the web server definition in WebSphere, you can propagate `plugin-cfg.xml` to the web server, request the web server to be started and stopped. WebSphere communicates with the IHS admin server using HTTP over port 8008 to complete the task.

1.4 Non-IHS Web Server

- There are two options for using another web server besides IBM HTTP Server

- ◇ Install on a machine that has a "local" Node Agent
 - Able to start/stop web server
 - Plug-in configuration propagation
 - View plug-in log
- ◇ Install on an "unmanaged" Node
 - Plug-in configuration must be manually copied from Deployment Manager machine to web server machine
 - Web server only defined to configure associated plug-in

Non-IHS Web Server

If using a non-IBM web server outside of a firewall a Custom profile could be created on the web server machine. This will provide the Node Agent required for remote management but will not create any Application Servers.

1.5 Managing Multiple Web Servers

- Often an environment will have multiple web servers
- Multiple web servers with duplicate configuration is difficult to manage with the web server management abilities of WebSphere
 - ◇ To use the full range of web server management features you would need to configure each web server separately
 - This is true even if they share the same plug-in configuration
 - ◇ You could configure one web server as a template
 - You would then have to manually update the plug-in configuration on every web server machine and manually stop or start the web servers

Managing Multiple Web Servers

If you have multiple web servers that share the same plug-in configuration it would obviously be best to configure this once with one web server definition. Unfortunately, if you do this you can not use WebSphere to generate and propagate the plug-in configuration, even if the web servers are IBM HTTP Servers. In this situation it might be best to create a script that can automatically copy the plug-in

configuration to all web server machines once WebSphere generates and propagates the template configuration.

1.6 Cluster

- A cluster is a grouping of application servers
 - ◇ Each Application Server is a "cluster member"
 - ◇ The cluster members can belong to different nodes
 - ◇ Cluster members can be added or removed at any time
- An application can be deployed to a cluster instead of a stand alone application server
 - ◇ This effectively deploys the application to all application servers that are members of that cluster
 - ◇ This provides workload management – it does not matter which member handles a HTTP or EJB request
- Clusters can be created only if you are using the WebSphere Network Deployment software

Cluster

A cluster itself has few properties. It's main purpose is to create a group of application servers. The member application servers are not much different than a regular server. Except, all members of a single cluster must run the same set of applications.

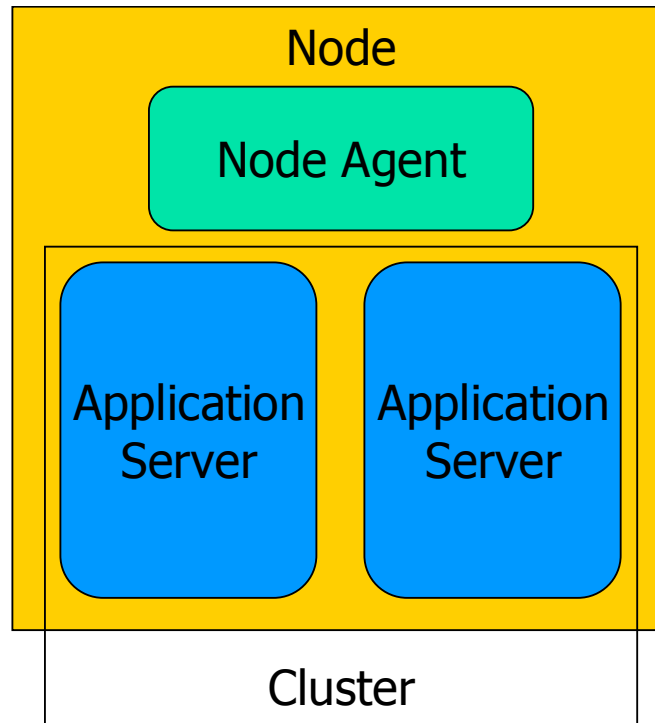
Application Servers that are part of a cluster, called Cluster members, still have individual configuration properties.

Note that an existing stand alone application server can not be added to the cluster as a member. Nor can a cluster member be removed from the cluster and become stand alone.

1.7 Vertical Scaling

- Multiple Application Servers in the same machine
- Provides better performance
 - ◇ This type of scaling can take advantage of machines with multiple CPUs and larger amounts of memory

- ◇ For smaller machines, configure fewer numbers of servers per machine



Vertical Scaling

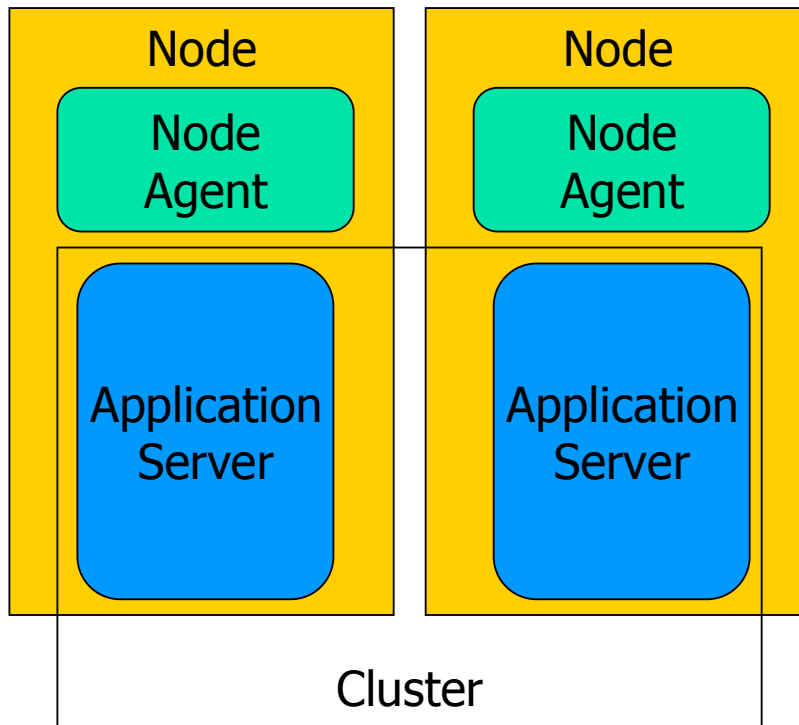
Even though it is possible to run more than one Node on a machine it is not typically done. The reason is that every Node must have a Node Agent even if there is another Node on the same machine. This provides overhead of running multiple Node Agents which wastes resources.

For a Java process, which an Application Server is, more memory is not always better. For machines with large amounts of memory, multiple servers may make more efficient use of that memory rather than allocating all of the memory to one server process. Managing too much memory is more difficult for a Java process to do.

Even moderately sized machines can typically handle more than one Application Server.

1.8 Horizontal Scaling

- Multiple Application Servers on different machines
- Provides for machine failover
 - ◇ Even if one machine goes down the application is available on other machines

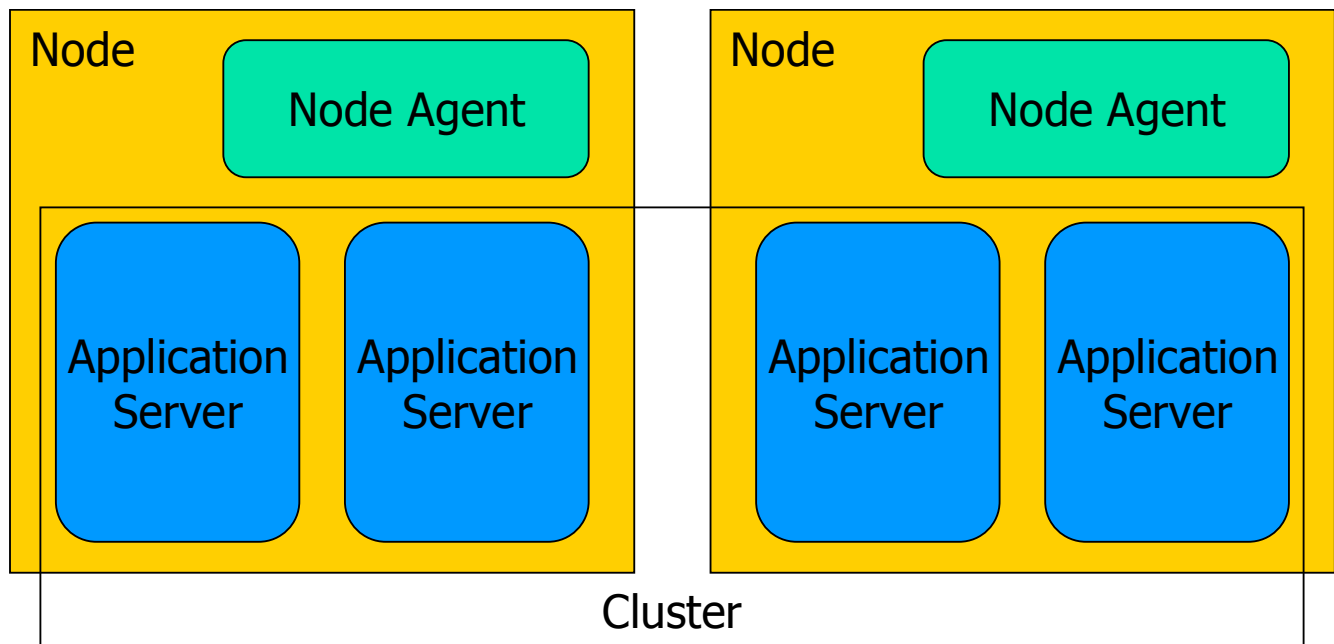


Horizontal Scaling

Since it is typical to have only one Node per machine this implies the Application Servers are running on different machines since they are in different Nodes.

1.9 "Mixed" Scaling

- Both types of scaling can be used for performance and failover benefits



"Mixed" Scaling

Using the different scaling options any type of topology can be created based on available hardware and performance requirements.

1.10 Heterogeneous Scaling

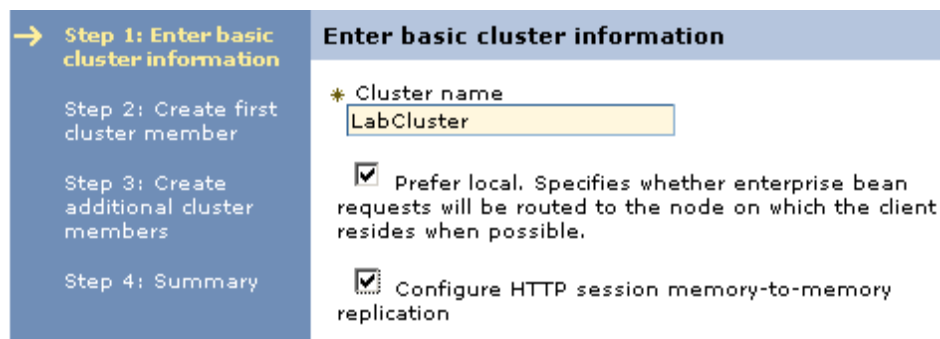
- WebSphere allows horizontal scaling across multiple, different platforms
 - ◇ That is, the nodes in a cell can be configured in Windows, Linux, Solaris, AIX and HP/UX
- A WebSphere v7 cell can also include nodes defined using WebSphere 5.x or 6.x
 - ◇ This helps gradual migration of a WebSphere system to v7
 - ◇ The Deployment Manager must be at the highest fixpack level
- This feature is also used by many companies to run the Deployment Manager in a cheaper Windows box
 - ◇ The nodes that run the application servers are configured in more expensive servers such as AIX or Solaris

Heterogeneous Scaling

The Deployment Manager server does not experience much user load. It is mainly used to administer the cell. As a result, the Deployment Manager can be running on cheaper hardware.

1.11 Creating a Cluster

- A new cluster is
 - ◇ Named
 - ◇ Configured with an initial list of cluster members
- A server template is used for the initial configuration of cluster members
- HTTP session memory-to-memory replication
 - ◇ This is important for sharing application data and can simplify future configuration



Creating a Cluster

When created, a cluster can configure the initial configuration of all cluster members in three ways:

1. Use an existing server as a cluster member. Every other cluster member will be configured like the existing server.
2. Use an existing server as a template but not as a cluster member. Cluster members will be configured like the existing server but the server will retain its independence from the cluster.
3. Use a default server template.

The Replication Domain is important for applications that use "sessions" to store user data. This will be discussed in more detail in the next lecture. For now, know that choosing this option when creating a cluster can make future configuration much easier.

The server weight is used as follows:

- Proportion of requests that are sent to a server = (weight of the server) / (sum of the weight of all member servers of a cluster)
- One option that should always be selected is "Generate Unique Http Ports". This will prevent port conflicts from servers running on the same machine.

1.12 Cluster Member Options

- Cluster members are added to Nodes
- Options for each cluster member
 - ◇ Weight
 - Proportion of requests that are sent to a server = (weight of the server) / (sum of the weight of all cluster members)
 - ◇ Generate Unique HTTP Ports
 - This will prevent port conflicts from servers running on the same machine

The screenshot shows a web-based configuration wizard titled "Create a new cluster". The current step is "Step 2: Create first cluster member". The interface includes a sidebar with navigation steps: "Step 1: Enter basic cluster information", "Step 2: Create first cluster member" (highlighted), "Step 3: Create additional cluster members", and "Step 4: Summary". The main content area contains the following fields and options:

- Member name:** A text input field containing "server2".
- Select node:** A dropdown menu showing "anil2003Node02(ND 6.1.0.0)".
- Weight:** A text input field containing "2", with a range "(0..20)" indicated to the right.
- Generate unique HTTP ports:** A checked checkbox.
- Select basis for first cluster member:** A section with three radio button options:
 - Create the member using an application server template. default (with a dropdown menu).
 - Create the member using an existing application server as a template. (with a dropdown menu containing "anil2003Cell01/anil2003Node02(ND 6.1.0.0)/server1").
 - Create the member by converting an existing application server. (with a dropdown menu containing "anil2003Cell01/anil2003Node02(ND 6.1.0.0)/server1").
 - None. Create an empty cluster.

At the bottom of the wizard are three buttons: "Previous", "Next", and "Cancel".

Cluster Member Options

In the screen shown above you create the initial list of cluster members.

When created, a cluster can configure the initial configuration of all cluster members in the following ways:

1. Use an existing server as a cluster member. Every other cluster member will be configured like the existing server.
2. Use an existing server as a template but not as a cluster member. Cluster members will be configured like the existing server but the server will retain its independence from the cluster.
3. Use a default server template.
4. Create an empty cluster

1.13 Cluster Member Options

- Additional cluster members can be added after the cluster has been created in the previous step
- Additional members can be added by entering the server names in various nodes and clicking on the Add Member button

Create a new cluster

Step 1: Enter basic cluster information
 Step 2: Create first cluster member
 → Step 3: Create additional cluster members
 Step 4: Summary

Create additional cluster members

Enter information about this new cluster member, and click Add Member to add this cluster member to the member list. A server configuration template is created from the first member and stored as part of the cluster data. Additional cluster members are copied from this template.

* Member name:

Select node:

* Weight: (0..20)

Generate unique HTTP ports

Use the Edit function to edit the properties of a cluster member that is already included in this list. Use the Delete function to remove a cluster member from this list. You are not allowed to edit or remove the first cluster member or an already existing cluster member.

Select	Member name	Nodes	Version	Weight
<input type="checkbox"/>	server2	anil2003Node02	ND 6.1.0.0	2

1.14 Cluster Member Options

- Administrators can view their actions in a summary window before committing changes and proceeding with creation of the cluster

Step 1: Enter basic cluster information
 Step 2: Create first cluster member
 Step 3: Create additional cluster members
 → Step 4: Summary

Summary

Summary of actions:

Options	Values
Cluster Name	LabCluster
Core Group	DefaultCoreGroup
Node group	DefaultNodeGroup
Prefer local	true
Configure HTTP session memory-to-memory replication	true
Server name	server2
Node	anil2003Node02(ND 6.1.0.0)
Weight	2
Clone Template	default
Clone Type	default
Generate unique HTTP ports	true

1.15 Managing Clusters

- Start/Stop a Cluster
 - ◇ Starts or stops all cluster members simultaneously
- Ripplestart a Cluster
 - ◇ Stops then starts each cluster member one at a time
 - ◇ Best option to ensure the cluster is available during the restart
- Add/Remove cluster members
 - ◇ Changes the Application Servers that are members of the cluster
- Rollout Update (available on list of applications)
 - ◇ This option updates a clustered application on each Node that has cluster members in sequence

Managing Clusters

After starting or stopping a cluster it is generally shown in a "partially" started or stopped state. This indicates that not all cluster members have finished starting or stopping.

You would add or remove cluster members to change the resources assigned to applications deployed to the cluster. This can be used to help adjust to changing demand for various applications. New cluster members can be assigned to run on a Node that has just been added into a Cell.

New cluster members would use the same server template that was applied to the original cluster creation.

Rollout Update performs the following steps:

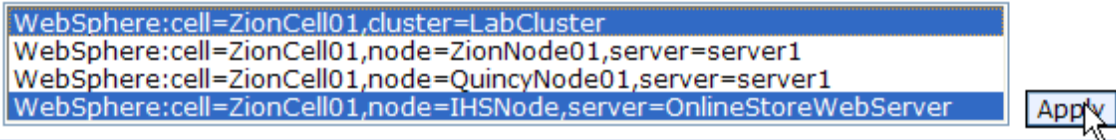
1. Saves the updated application configuration.
2. Stops all of the cluster members on one node.
3. Updates the application on the node by synchronizing the configuration.
4. Restarts the stopped cluster members.
5. Repeats steps 2 through 4 for all of the nodes that have cluster members.

1.16 Mapping Applications to Clusters

- Applications are mapped to the entire cluster

- ◇ It does not matter how many cluster members are in the cluster
- Applications mapped to a cluster will be automatically distributed to all Nodes that have cluster members
- ◇ This is done automatically by the file synchronization service

Clusters and Servers:

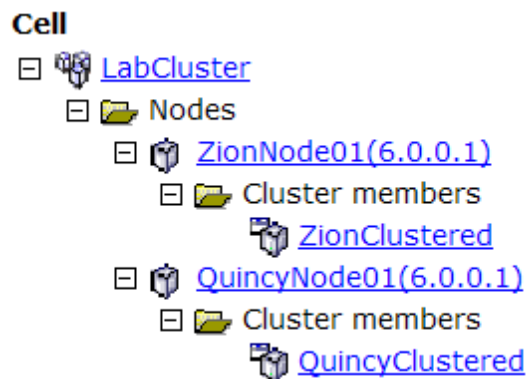


Mapping Applications to Clusters

The entries in the list shown above are clusters, servers that are not part of a cluster, and web servers. Besides mapping an application to a cluster it is critical to also map it to a web server as this is what will provide workload management as will be seen in the next chapter.

1.17 Cluster Topology Screen

- Displays the structure of clusters in the cell
- View expands and collapses like folders on a file system
- Nodes are only listed if they have cluster members
- Servers are only listed if they are cluster members
- You can click a link to view properties



1.18 Topology

- Topology stands for the hardware and software layout in a system
 - ◇ Decision revolves around how many machines you need and what each machine will run
- Determines the level of fault tolerance and load sharing
- Software that figures in a topology design are:
 - ◇ Web server
 - ◇ Application server
 - ◇ Node Agent and Deployment Manager
 - ◇ Database server

1.19 Factors Affecting Topology

- Security – Locating the web server behind one fire wall (in a DMZ) and the application server behind a second firewall adds security. The second firewall can have more stringent filters such as source IP addresses
- Load sharing – If you run the same application in more than one application server, you will most likely see better performance under heavy stress. In WebSphere you can locate these servers in:
 - ◇ A single machine – vertical scaling
 - ◇ In multiple machines – horizontal scaling
- Web server load sharing – You can set up a IP dispatcher machine that will route incoming HTTP requests to multiple web servers

Factors Affecting Topology

Web server load sharing is somewhat special. There has to be a single machine with an IP address that maps to the host name used by the web site users. Unless you use IP dispatcher, you must run a single web server machine. The load on a single web server machine may be too much. More importantly, a single web server machine does not provide any redundancy.

1.20 Factors Affecting Topology

- On demand scaling – You should be able to add extra web server and application server machines with minimal administrative changes to the existing system
- Process separation – You should be able to run multiple unrelated applications in separate JVMs. If one JVM goes down, it should not affect the other applications
- Failover – If you run the same application in multiple application servers, you get fault tolerance
- Minimum service interruption – Horizontal scaling can minimize downtime during scheduled maintenance

Factors Affecting Topology

Process separation

WebSphere gives you the option to run more than one unrelated application in the same application server. This is a good idea if you want to keep the resource usage low. If you run them in separate application servers, you will need more horsepower, but, you will get fault isolation.

Maintenance

If you can foresee a list of scheduled maintenance operations, you should be able to design a topology that minimizes system down time. For example, you can build a cluster of application servers. If you have copied an updated set of JAR files, you can do a ripple restart of the servers. System would stop and start one member server at a time and make sure that at least one server is available.

1.21 Coexistence Scenarios

- You can create a Deployment Manager profile on the same machine as other profiles
 - ◇ No need to dedicate a separate machine
- You can run multiple Deployment Managers in the same machine
 - ◇ Each Deployment Manager will be a separate profile
 - ◇ Each will manage a different cell
 - ◇ Cells will share the same ND installation

- You can define more than one node in a single machine
 - ◇ Will require you to run multiple node agents, one per node
 - ◇ Nodes can share the same WebSphere base installation

Coexistence Scenarios

Coexistence is useful when you are running multiple test environments in the same group of machines. For example, different versions of the same software can be installed in separate nodes or cells. They can be independently started and stopped.

WebSphere profiles make configuring coexistence easy. If all profiles are part of the same installation WebSphere can help avoid port conflicts automatically.

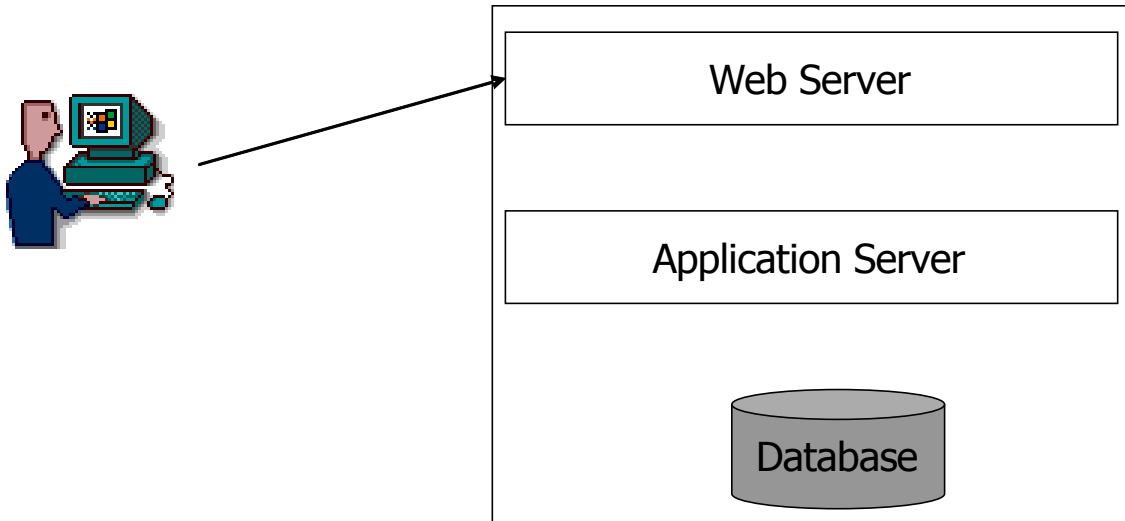
1.22 Common Topologies

- All in one
- Vertical scaling
- Web and database server separation
- Horizontal scaling
- Web server horizontal scaling

Common Topologies

Each of the above topologies, which will be discussed in more detail next, address different factors mentioned previously.

1.23 All in One



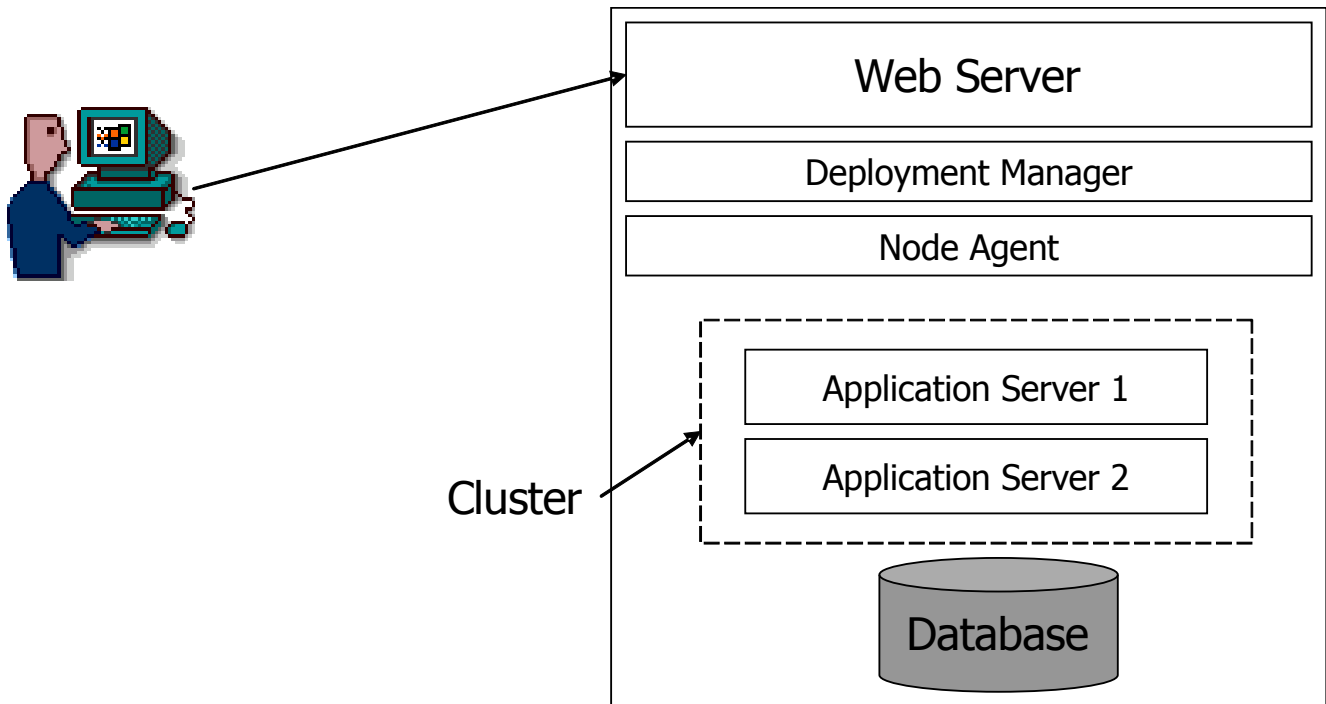
All in One

In this topology, the web server, application server and database server run in a single machine. This is the simplest topology and quite common in:

- Small web sites.
- Developer's workstation.
- Tester's machine.

You need to install just the WebSphere Base edition to run the application server.

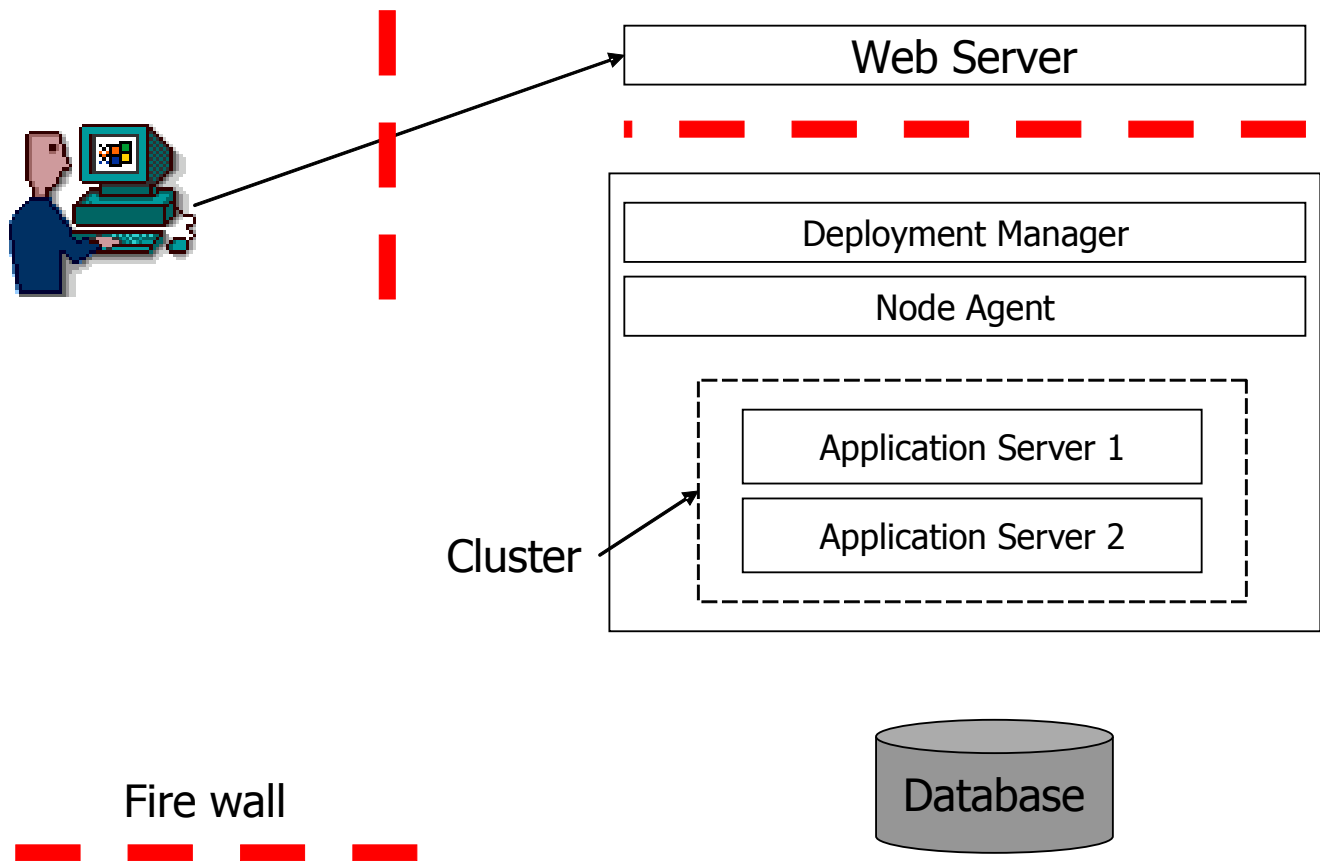
1.24 Vertical Scaling



Vertical Scaling

- We still have everything running in one machine. But, now we have the two application servers belong to a cluster. Now, they can execute the same application. The system will now offer EJB and Web container load balancing and fail over.
- The Network Deployment software is installed and configured with a Deployment Manager profile and a second profile that is added to the Network Deployment Cell. The Deployment Manager and Node Agent must both be running for administration.
- This topology is called vertical scaling because, the member application servers of the cluster belong to the same physical machine. This topology is well suited for powerful machines.
- The web server plug in can do load sharing between the application servers.

1.25 Server Separation



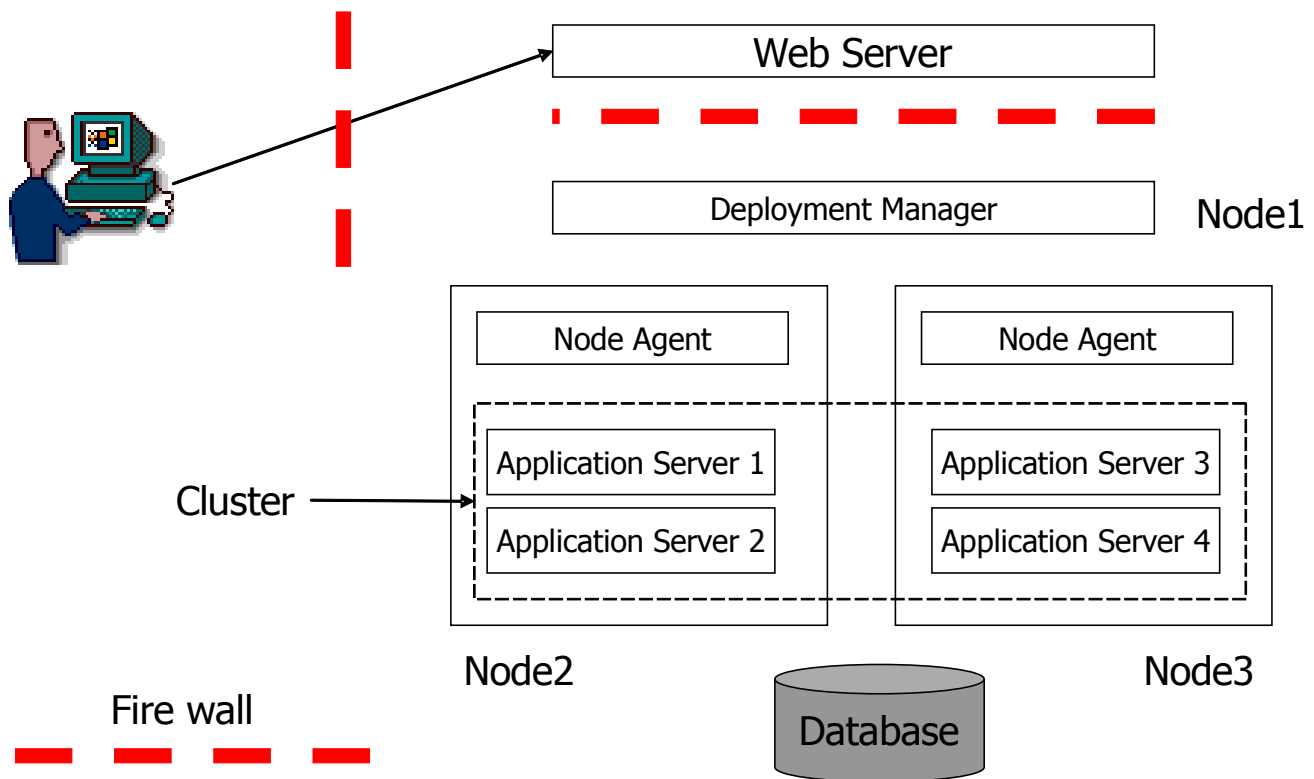
Server Separation

In this topology, the web server and/or the database server are separated from the application server machine. The web server machine is positioned between two firewalls in a demilitarized zone (DMZ). The first firewall does protocol based filtering and allows HTTP and HTTPS traffic only. The second firewall allows traffic from the web server's IP address only. The database or business logic is accessible from behind the second firewall only. Advantages of this topology are:

- Better security through multiple firewalls.
- You can locate static content such as images and HTML files in the web server machine. This reduces load on the application server machine.
- A dedicated database server machine will improve performance.

Keep in mind, that when you generate the plugin configuration file (`plugincfg.xml`), it is created in the machine running the Deployment Manager. You may need to manually copy the file to the web server machines, if the web server is something other than IBM HTTP Server v6.

1.26 "Mixed" Scaling

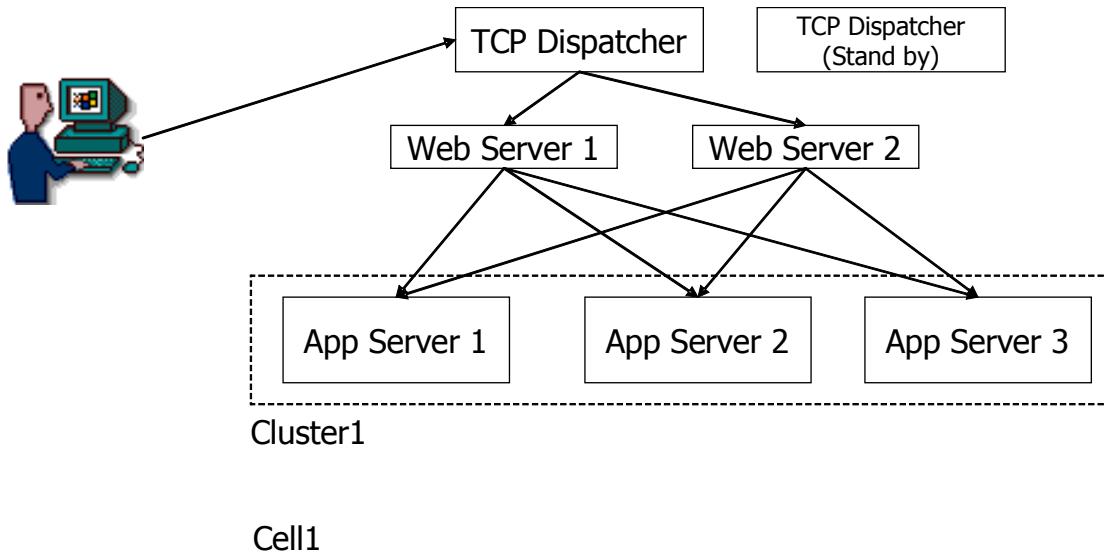


"Mixed" Scaling

The Deployment Manager may be running in a stand alone machine. It can also run in one of the application server machines. In this example, we have the Deployment Manager running in node1. The WebSphere Network Deployment software had to be installed in that machine. Node2 and Node3 are in two physical machines that have WebSphere base installed. Both nodes have joined the cell that is being managed by the Deployment Manager. Application Servers 1 and 2 are defined in Node2. Application Servers 3 and 4 are defined in Node3. A cluster is created in the cell that includes all four application servers. Since the cluster spans multiple machines and includes more than one server on each machine, we call the configuration "mixed" scaling. Main advantages are:

1. Provides better performance. You can add additional machines on demand. Expand the cluster further to increase the processor pool.
2. Provides hardware fail over.
3. Many types of maintenance can be done in one machine at a time. This greatly improves the availability of the application.

1.27 Web Server Horizontal Scaling



Web Server Horizontal Scaling

This topology allows us to run more than one web server. TCP dispatcher software routes incoming HTTP requests to the web servers. This distributes the load between several web server machines. It also offers web server redundancy. All web server machines have the same copy of the plugin-cfg.xml file. The web server plug in knows how to forward HTTP requests to the appropriate web container. Main advantages are:

1. Web server redundancy.
2. Better web server performance.

1.28 Web Server Management & Cluster Topology Questions

1. What does the ND Admin Console allow you to do when running "IHS" on a remote machine? What tool does it rely on to accomplish some of these features?
2. If you're using a non-IHS web server, what must the machine have running on it in order to be able start/stop the web server and propagate the plug-in configuration file?
3. What is a cluster?
4. What is the difference between horizontal, vertical, and mixed scaling?

5. What happens when you deploy an application to a cluster?

1.29 Web Server Management & Cluster Topology Answers

1. Edit web server configuration file, start/stop web server, automatically propagate plug-in configuration file to web server, view web server log files, and view plug-in log file. IBM HTTP Administration Service
2. Node Agent
3. A grouping of application servers that run the same set of applications and are configured the same way
4. Horizontal scaling has multiple application servers on different machines. Vertical scaling has multiple application servers on the same machine. Mixed scaling has both types of scaling
5. The application gets copied to every node containing a server that's a member of the cluster

1.30 Reference

- WAS System Management and Configuration Redbook
 - ◇ Chapter 8 - Managing Web Servers
 - ◇ Section 5.6 - Working with Clusters
- WAS Scalability and Performance Redbook
 - ◇ Chapter 2 - Infrastructure Planning and Design
 - ◇ Chapter 3 - Introduction to Topologies