Configuring a Cluster

At the end of this module, you will be able to:

- ✓ Prepare your environment for a cluster
- ✓ Create and configure a cluster
- ✓ Create and configure a proxy server

Road Map

1. **Preparing for a Cluster**
   - Cluster License
   - Cluster Architecture
   - Network and Security Topology
   - Machines
   - Names and Addresses
2. Configuring a Cluster
3. Configuring a Proxy Server

Preparing Your Environment

- Before you can configure a cluster, there are steps you need to take to prepare your environment.
  - Obtain a cluster license
  - Determine your cluster architecture
  - Understand your network and security topologies
  - Choose the machines for the cluster installation
  - Identify IP Addresses or DNS names, and port numbers for the server instances in the cluster

Cluster License

- Clustered WebLogic server instances must have a valid cluster license.
- To update your current license, use the UpdateLicense.cmd in the main BEA directory.
  - UpdateLicense <new_license_file>

```
<license
        component="Cluster"
        cpus="unvalued"
        expiration="never"
        ip="any"
        licensee="BEA Internal Development"
        serial="616351266349-1844896394531"
        type="SDK"
        units="5"
        signature="MC0CFQCQrk+Kbddfz3RHVH6uGfj"
/>
```

Cluster Architecture

- Will you be using a single tier or a multi-tier architecture?
- How are you going to do your load balancing?
  - Are you going to use basic WebLogic server load balancing?
  - Will you use a 3rd party load balancer?
- Are you going to use de-militarized zones with firewalls?

Network and Security Topology

- Will your cluster existing in a single LAN?
- Will your cluster span a MAN or a WAN?
- Depending on the network topology you choose, your security requirements will change.
  - Some network topologies can interfere with multicast communications
  - Avoid deploying server instances in a cluster across a firewall

Security Options for Cluster Architectures

- For proxy architectures you could have:
  - A single firewall between untrusted clients and the Web server layer
  - A firewall between the proxy layer and the cluster
- When using a load balancer, in addition to the security features provided with the load balancer, you may want to place a firewall between it and untrusted clients.
- When you use a single database supporting both internal and external data:
  - Place an additional firewall in front of the database server

Hardware

- You may set up a cluster on a single, non-multihomed computer for demonstration or development purposes.
  - This is not practical for production environments
- The machine cannot have a dynamically assigned IP address.
- There is no built-in limit for the number of server instances in a cluster.
  - The only limitation is your license
  - Large, multi-processor servers can host large clusters
  - The recommendation is one WebLogic Server instance for every two CPUs

Clustering on One Piece of Hardware

- To set up a cluster on one machine, the server is not required to be multi-homed; the servers in a cluster can use one IP address.
- Servers in a cluster can use a dedicated multicast port number for inter-server communication:
  - Required, if servers are using a single IP address
  - Useful for segmenting multicast traffic among specific NICs

Names and Addresses

- Location information will be needed for:
  - The administration server
  - Managed servers
  - Multicast location
- For a production environment, it is recommended that you use DNS names, as opposed to IP addresses.
  - Firewalls can cause IP address translation errors
- Each WebLogic server resource should have a unique name
- The multicast address should not be used for anything other than cluster communications.

Cluster Address

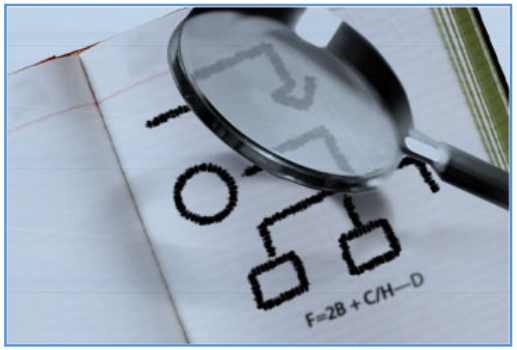
- The cluster address is used in entity and session beans to construct the host name portion of request URLs.
- You can explicitly define the address of a cluster.
  - The cluster address should be a DNS name that maps to the IP addresses or DNS names of each WebLogic server instance in the cluster
- You can also have WebLogic Server dynamically generate an address for each new request.
  - Minimizes configuration
  - Ensures accurate cluster address
- The dynamic cluster address is created in the form of:

```
listenaddress1:listenport1,listenaddress2:listenport2,listen
address3:listenport3
```

Section Review

In this section, we learned how to:

- ✓ Prepare the environment for a cluster



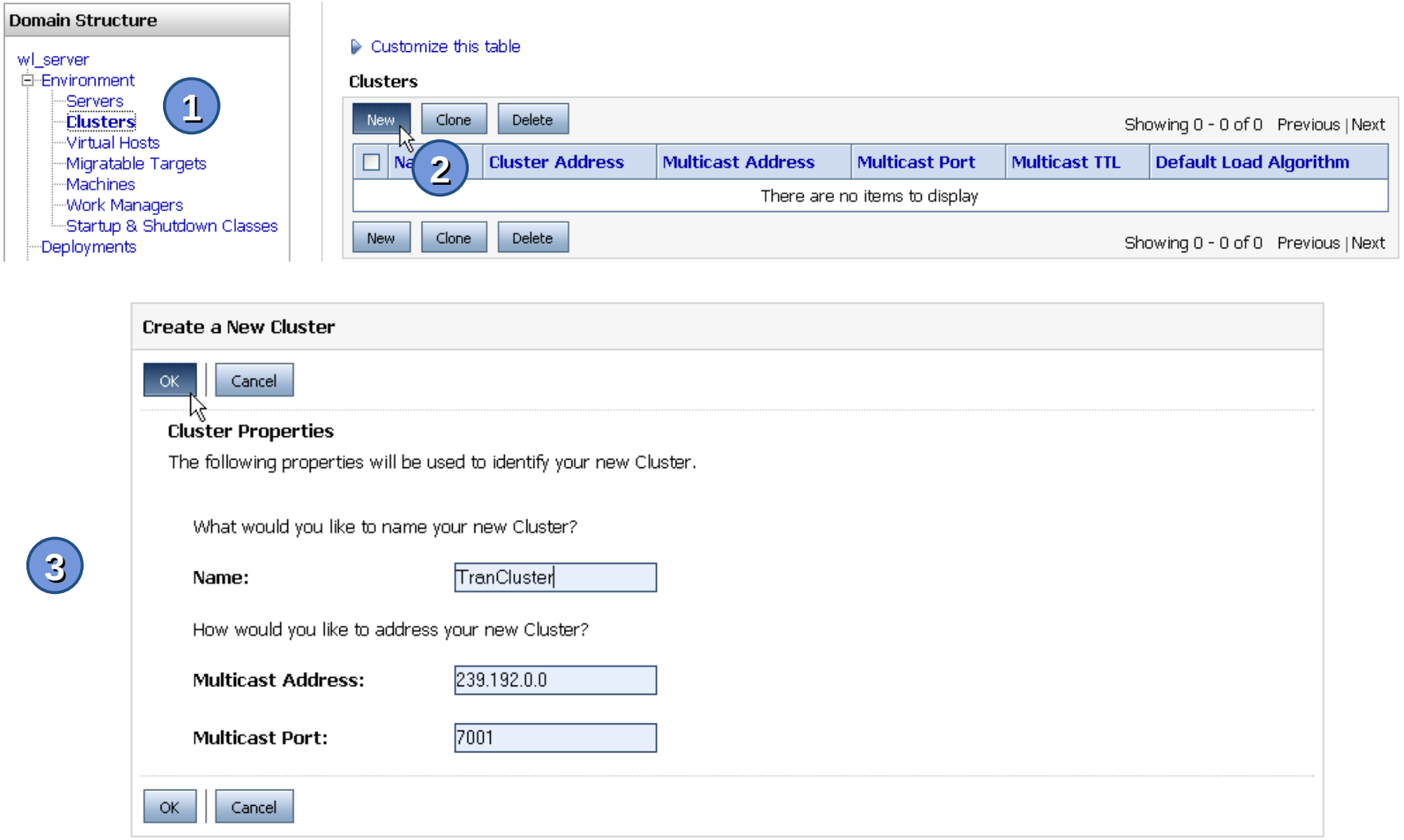
Road Map

1. Preparing for a Cluster
2. **Configuring a Cluster**
   - Administration Console
   - Configuration Wizard
   - WLST
   - Ant
3. Configuring a Proxy Server

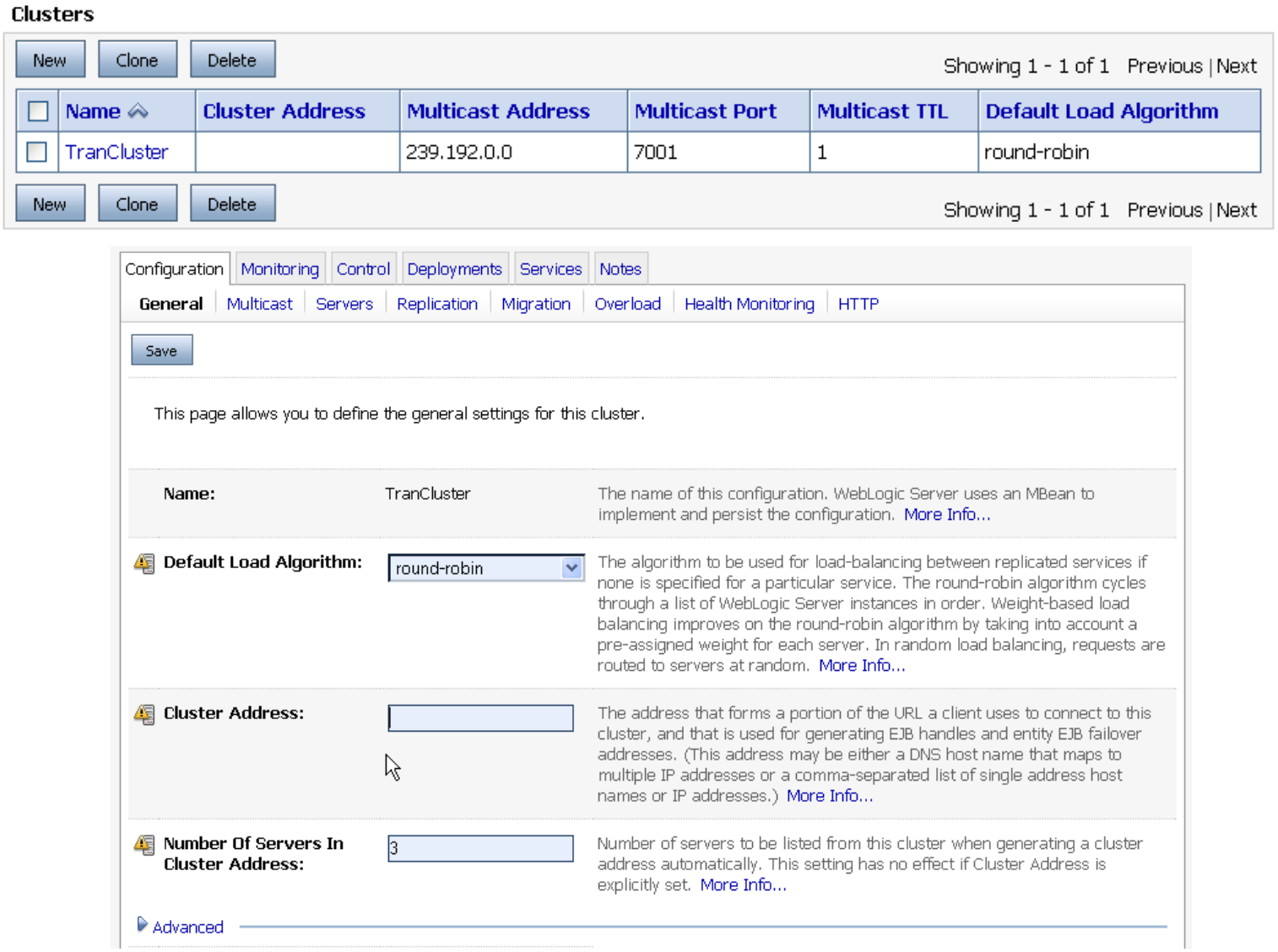
Configuration Options

- There are multiple ways to create and configure a WebLogic Server cluster:
  - Configuration Wizard
  - Administration Console
  - Ant
  - WLST

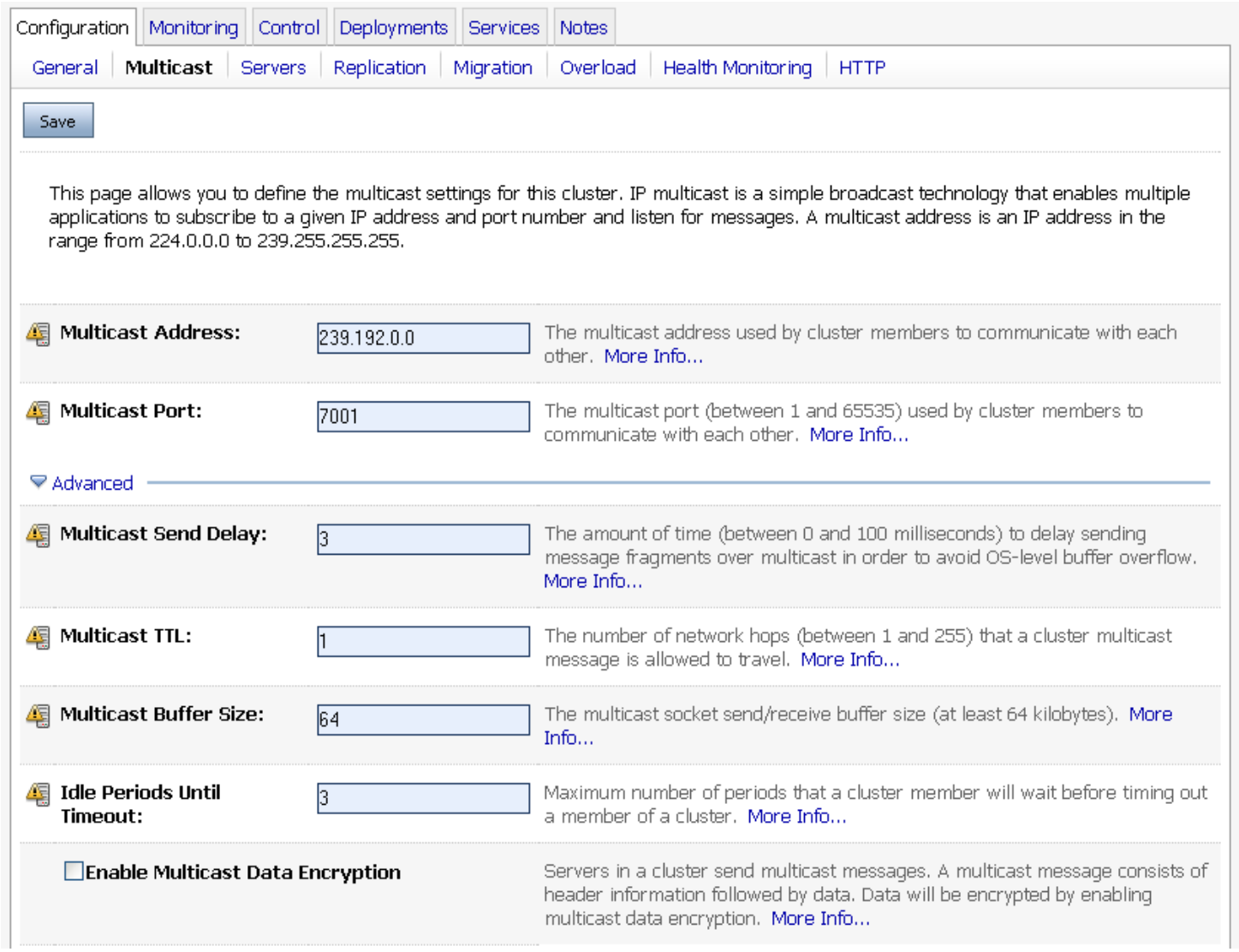
Creating a Cluster Using the Administration Console…



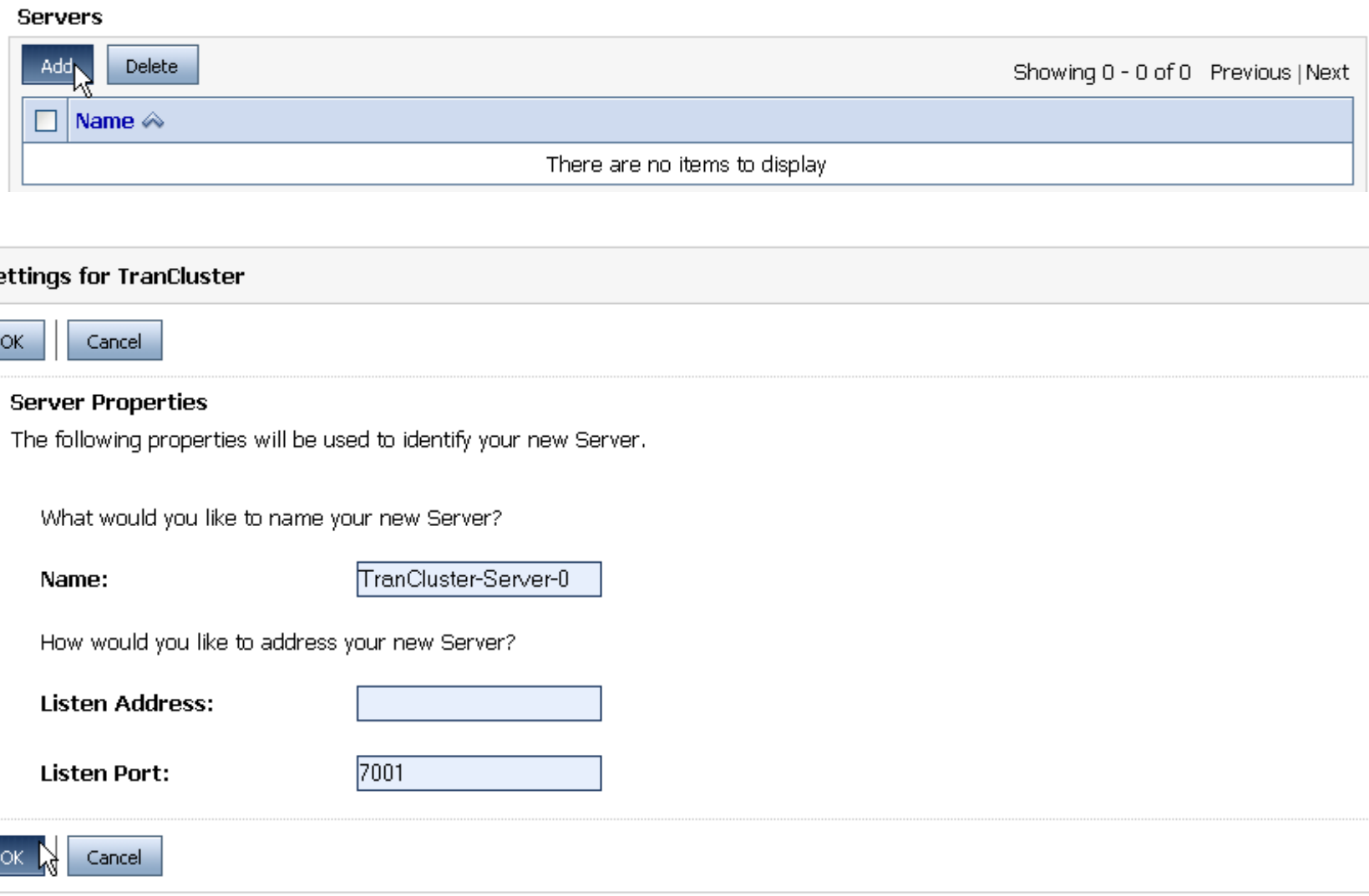
...Creating a Cluster Using the Administration Console



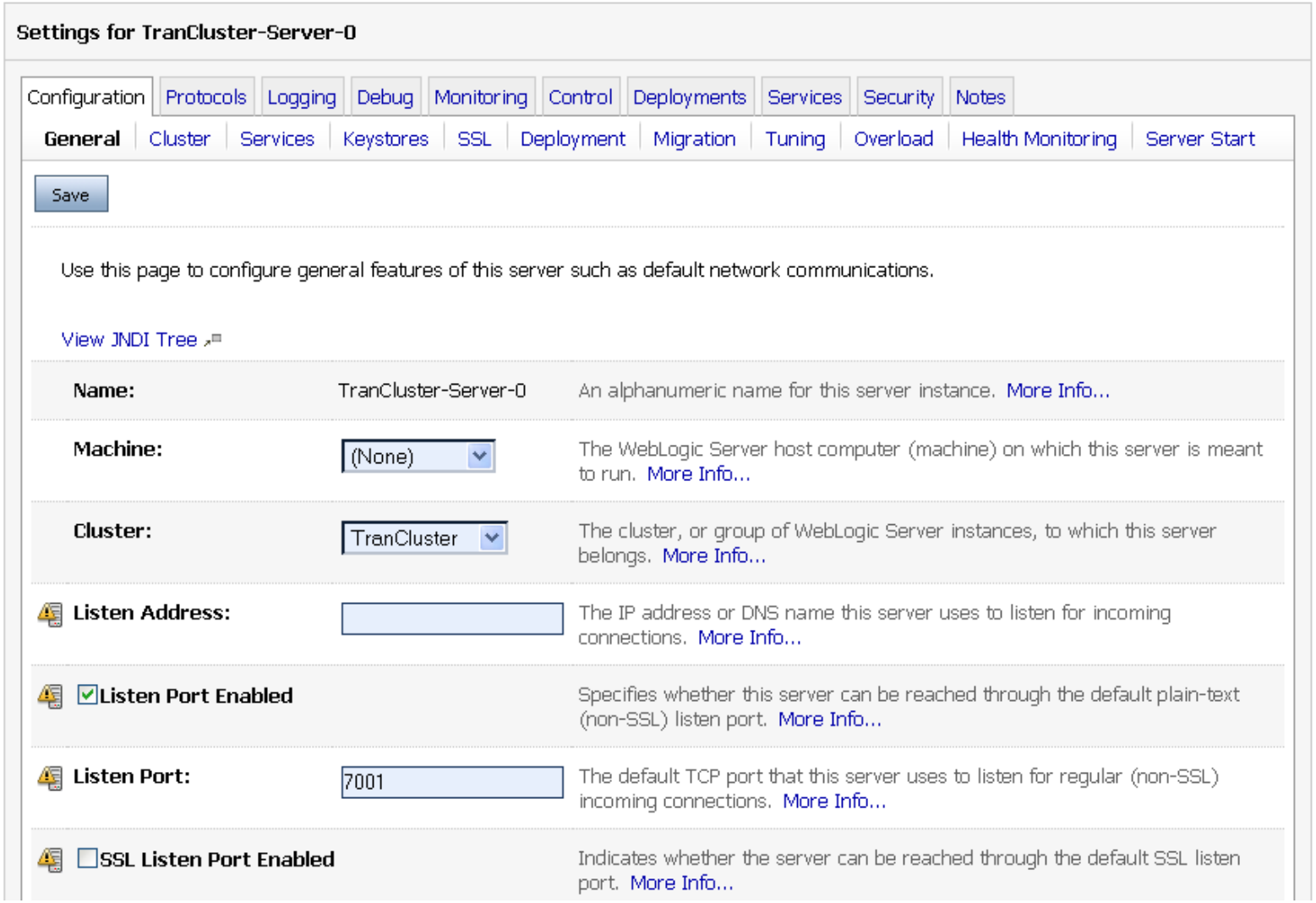
Configuring Multicast



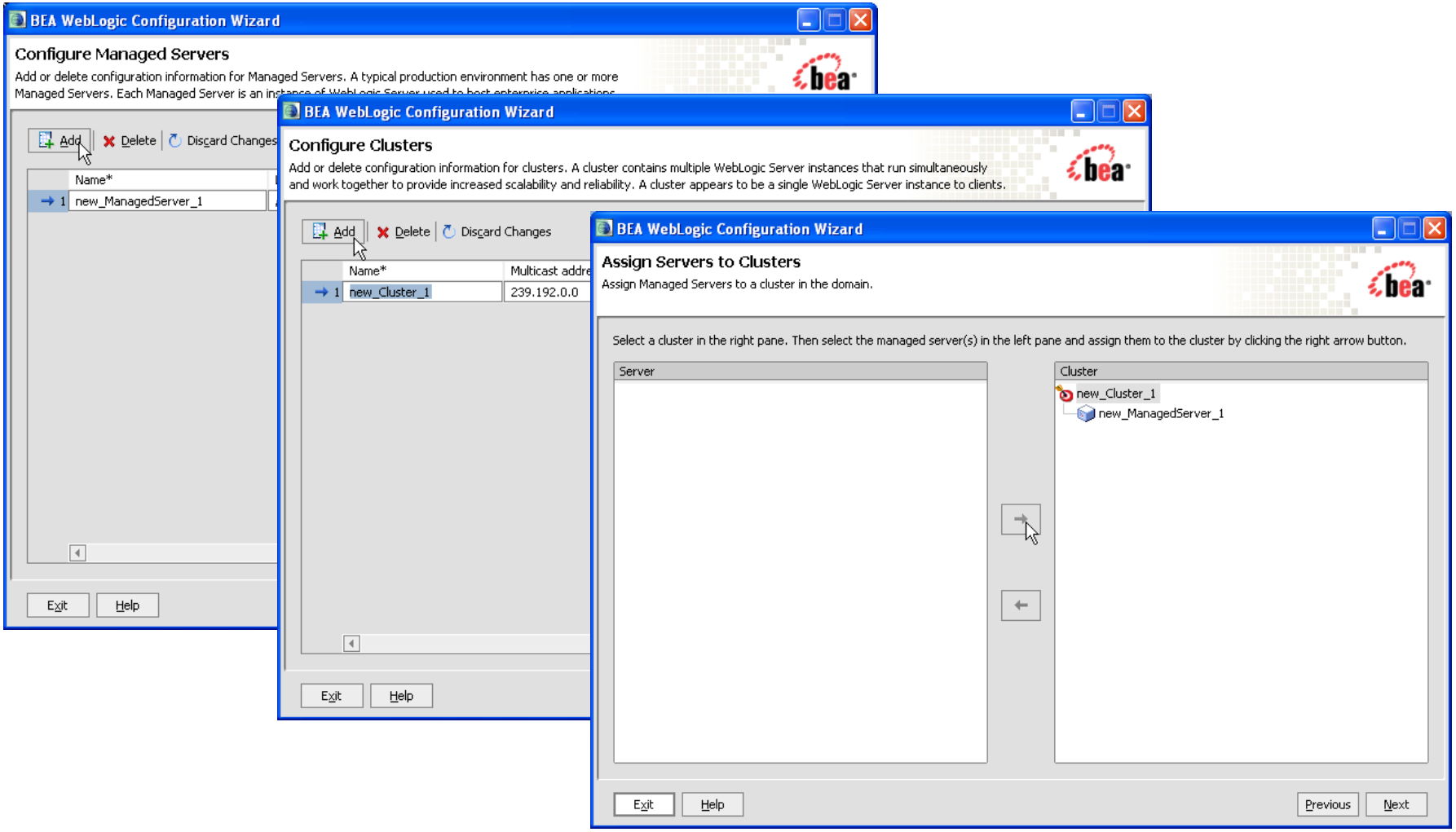
Adding Servers…



...Adding Servers



Creating a Cluster with the Configuration Wizard



Using the Cluster MBean

- The Cluster MBean is used to create a cluster using Ant or command-line tools.
- Configuring the cluster from the command line requires the combined use of Cluster and Server MBeans.
- To create new clusters within a domain, use:

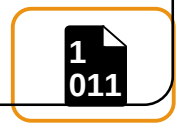
`weblogic.management.configuration.ClusterMBean`

Creating a Cluster with WLST

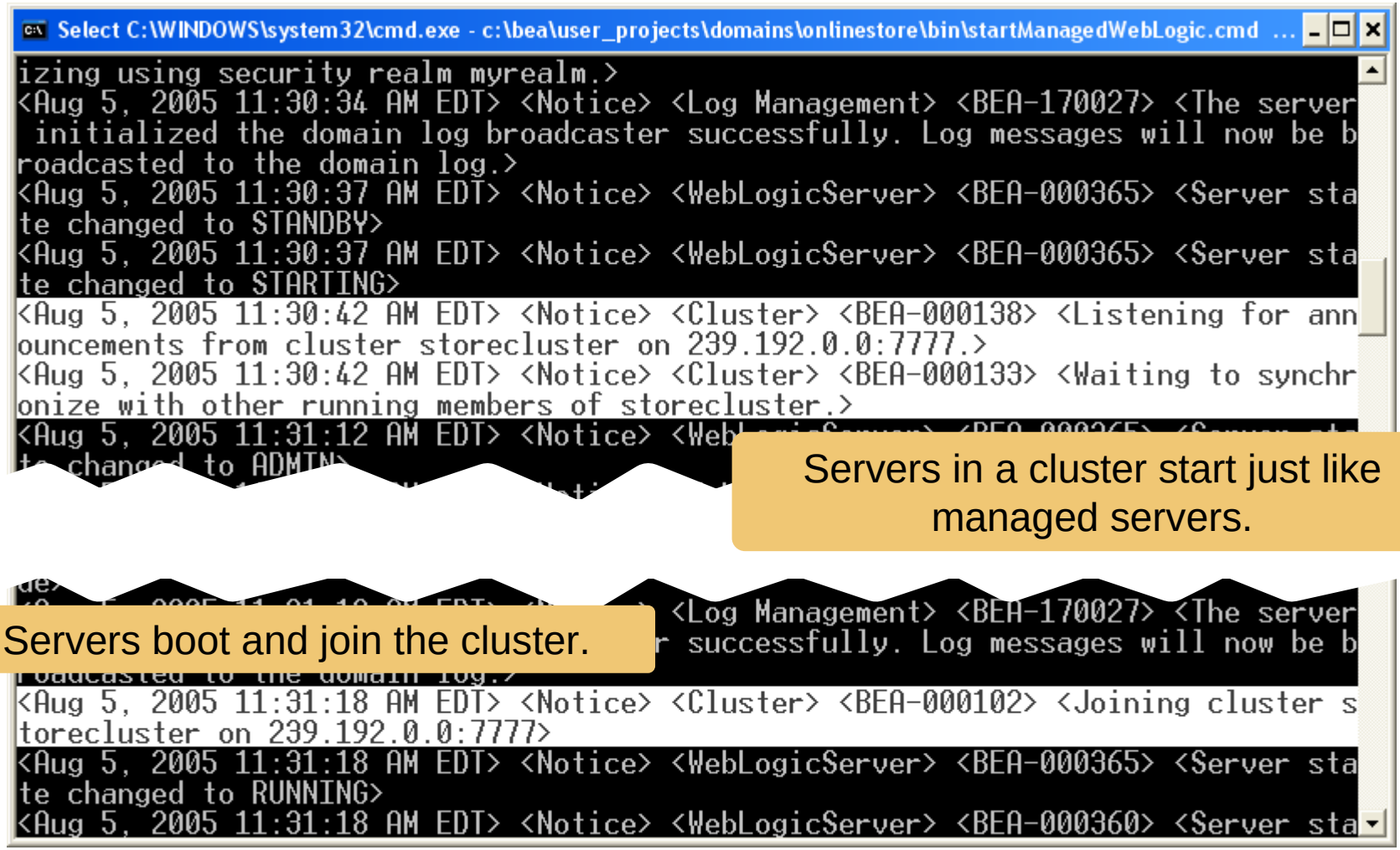
```
from java.util import *
from javax.management import *
connect('system','weblogic','t3://localhost:7001')
clusters = {"dizzyCluster"}
ms1 = {'dizzy1':7003,'dizzy2':7005,'dizzy3':7007}
clustHM = HashMap()
serverName = mbs.getServerName()
cd('Servers')
cd(serverName)
for c in clusters:
      clu = create(c,'Cluster')
      clustHM.put(c,clu)
cd('..\..')
clus1 = clustHM.get(clusters[0])
for m, lp in ms1.items():
      managedServer = create(m1,'Server')
      managedServer.setListenPort(lp)
      managedServer.setCluster(clus1)
disconnect()
quit()
```

Creating a Cluster with Ant

```
<wlconfig url="t3://localhost:7001" username="system"
     password="weblogic">
<create type="Cluster" name="dizzyCluster">
 <set attribute="MulticastAddress" value="234.0.0.1"/>
 <set attribute="MulticastPort" value="7070"/>
 <set attribute="ClusterAddress“
     value="127.0.0.1,127.0.0.1,127.0.0.1"/>
 <set attribute="DefaultLoadAlgorithm" value="round-robin"/>
</create>
<set attribute="Cluster"
     value=“dizzyworld:Name=dizzyCluster,Type=Cluster"
     mbean=" dizzyworld:Name=dizzy1,Type=Server"/>
<set attribute="Cluster"
     value="dizzyworld :Name=dizzyCluster,Type=Cluster"
     mbean="development:Name=dizzy2,Type=Server"/>
<set attribute="Cluster"
     value="dizzyworld:Name=dizzyCluster,Type=Cluster"
     mbean="development:Name=dizzy3,Type=Server"/>
```



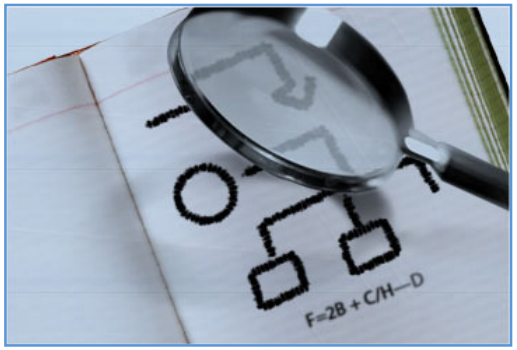
Starting Servers In a Cluster



Section Review

In this section, we learned how to:

- ✓ Configure a cluster in the administration console
- ✓ Create a cluster using the configuration wizard
- ✓ Configure a cluster using WLST
- ✓ Configure a cluster using Ant



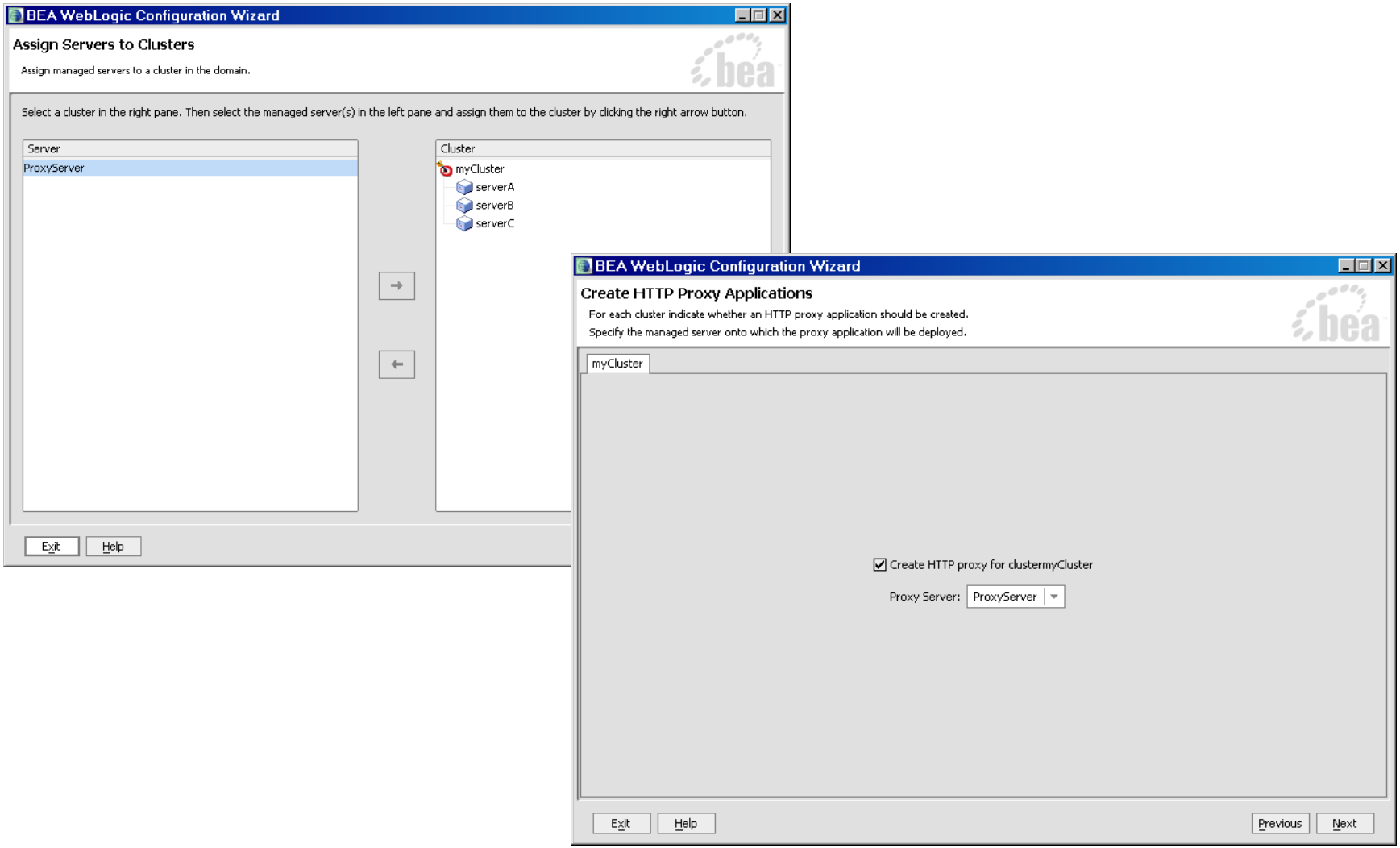
Road Map

1. Preparing for a Cluster
2. Configuring a Cluster
3. **Configuring a Proxy Server**
   - Configuration Wizard
   - HTTPClusterServlet
   - Apache

WebLogic Proxy Servers

- The WebLogic `HTTPClusterServlet` runs within a web application deployed on a WebLogic server.
  - The servlet proxies requests to other servers in a cluster
  - It should run on a separate, non-clustered managed server
- A WebLogic proxy server can be created initially using the Configuration Wizard.
  - You can also manually set up the web application with the HTTPClusterServlet and deploy it on the managed server

Create the WebLogic Proxy Server with the Configuration Wizard



Creating the WebLogic Proxy Server Manually

- The `HttpClusterServlet` is specified in the `web.xml` file of the default Web Application on the proxy server.
- This file must reside in the `\WEB-INF` directory of the web application directory.
- The proxy servlet needs to be defined as the default web application for the managed server.
  - This is defined in the `weblogic.xml` deployment descriptor located in the `\WEB-INF` directory of the web application

Configuring the HttpClusterServlet

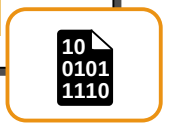
- The HttpClusterServlet is specified in the web.xml file of the default Web application on the proxy server.

HttpClusterServlet Declaration:

```
<servlet>
  <servlet-name>HttpClusterServlet</servlet-name>
  <servlet-class>
    weblogic.servlet.proxy.HttpClusterServlet
  </servlet-class>

  … …
</servlet>
```

<web.xml>

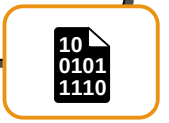


Specifying Initial Parameters

Declaring Initial Parameters:

```
<servlet>
  <servlet-name>HttpClusterServlet</servlet-name>
  <servlet-class>
    weblogic.servlet.proxy.HttpClusterServlet
  </servlet-class>
  <init-param>
     <param-name>WebLogicCluster</param-name>
     <param-value>
       serverA:7001:7002|serverB:7001:7002|serverC:7001:7002
     </param-value>
   </init-param>
   <init-param>
     <param-name>DebugConfigInfo</param-name>
     <param-value>ON</param-value>
   </init-param>


</servlet>
```



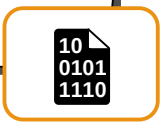
<web.xml>

Servlet Mapping

Configuring Servlet Mapping:

```
<servlet>
  <servlet-name>HttpClusterServlet</servlet-name>
  …
</servlet>
<servlet-mapping>
  <servlet-name>HttpClusterServlet</servlet-name>
  <url-pattern>/</url-pattern>
</servlet-mapping>

<servlet-mapping>
  <servlet-name>HttpClusterServlet</servlet-name>
  <url-pattern>*.jsp</url-pattern>
</servlet-mapping>
```



<web.xml>

HttpClusterServlet Initialization Parameters...

Parameter	Usage	Default Value
WebLogicCluster	(Required) A list of host names and port numbers of the servers to which requests are proxied.	(none)
secureProxy	ON/OFF ON enables SSL between HttpClusterServlet and the server it proxies to.	OFF
DebugConfigInfo	ON/OFF ON lets you query the HttpClusterServlet for debugging information.	OFF
ConnectTimeoutSecs	Maximum time in seconds that the servlet should attempt to connect to the WebLogic Server host.	0 (infinite timeout)
ConnectRetrySecs	Interval in seconds that the servlet will sleep between attempts to connect to a server instance.	5

...HttpClusterServlet Initialization Parameters

Parameter	Usage	Default Value
pathTrim	The string to trim from the beginning of the original URL.	none
trimExt	The file extension to trim from the end of the URL.	none
pathPrepend	The string to prefix to the original URL after pathTrim is trimmed.	none

3rd Party Proxy Servers

- If you are using a supported third-party web server, instead of a WebLogic web server, you will need to set up a proxy plug-in.
- The follow are the supported third-party web servers:
  - Netscape Enterprise Server
  - Apache
  - Microsoft Internet Information Server

Configure Apache Plug-in…

- Install the Apache HTTP Server plug-in as a module in your Apache HTTP Server installation.
- The module can be configured in two different ways:
  - As a dynamic shared object
  - As a statically linked module (1.3.x and higher)
- To configure the Dynamic Shared Object, select shared object type
  - mod_wl_20.so (Standard Apache Version 2.x)
  - mod_wl_ssl.so (Apache with SSL/EAPI)

...Configure Apache Plug-in...

- Edit the httpd.conf file and include WebLogic Server Modules
  - `LoadModule weblogic_module modules/mod_wl_20.so`
- Add an IfModule block to define the WebLogic Cluster Instance

```
<IfModule mod_weblogic.c>
    WebLogicCluster
  127.0.0.1:7003,127.0.0.1:7005,127.0.0.1:7
  007
</IfModule>
```

…Configure Apache Plug-in

- Add the proxy path

```
<Location /weblogic>
    SetHandler weblogic-handler
    WebLogicCluster
  127.0.0.1:7003,127.0.0.1:7005,127.0.0.1:7007
    DebugConfigInfo ON
    PathTrim /weblogic
</Location>
```

Apache Plug-in and SSL

- You can use the Apache Plug-in with SSL to guarantee the confidentiality and integrity of data.
- The Apache HTTP Server Plug-In does not use the transport protocol (http or https) specified in the HTTP request.
- Even though you can use two-way SSL authentication between the client and Apache, one-way authentication will be used between Apache and WebLogic.

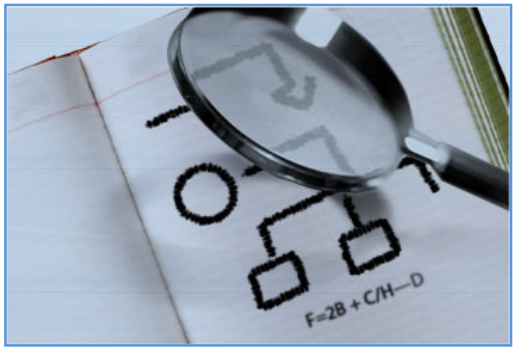
Configure Apache Plug-in SSL

- To enable SSL between Apache and WebLogic
  - Configure WebLogic for SSL
  - Configure WebLogic Server's SSL listen port
  - Set the WebLogicPort in httpd.conf to the WebLogic SSL listen port
  - Set the SecureProxy parameter in httpd.conf to ON
  - Install a trusted certificate authority file on Apache. You can use the DemoTrust.jks that comes with WebLogic.

Section Review

In this section, we learned how to:

- ✓ Set up the HTTPClusterServlet for proxying
- ✓ Set up Apache for proxying



Exercise

Create a Cluster

- For details on the exercise, refer to the Lab Guide.
- If questions arise, ask the instructor.
- The instructor will determine the stop time.

Exercise

Configure Proxy Servers

- For details on the exercise, refer to the Lab Guide.
- If questions arise, ask the instructor.
- The instructor will determine the stop time.

Module Review

In this module, we learned how to:

- ✓ Prepare your environment for clusters
- ✓ Configure a cluster using different tools
- ✓ Configure WebLogic server and 3rd party proxy servers