IBM Software
WebSphere

WebSphere Lab Jam

Application Infrastructure

IBM Workload Deployer

Lab Exercise

Contents

THIS PAGE INTENTIONALLY LEFT BLANK

Connection Parameters Spreadsheet

Student Suffix	IWD User ID's	Password
ID's for Morning Lab Session 1		
01	user01	password
02	user02	password
03	user03	password
04	user04	password
05	user05	password
06	user06	password
07	user07	password
08	user08	password
09	user09	password
10	user10	password
11	user11	password
12	user12	password
13	user13	password
14	user14	password
15	user15	password
ID's for Afternoon Lab Session 2		
16	user16	password
17	user17	password
18	user18	password
19	user19	password
20	user20	password
21	user21	password
22	user22	password
23	user23	password
24	user24	password
25	user25	password
26	user26	password
27	user27	password
28	user28	password
29	user29	password
30	user30	password
IBM Workload Deployer:		**XX.XX.XX.XX**

Lab 1 Building and deploying a custom virtual system pattern

IBM Workload Deployer provides multiple deployment models to meet the needs of your application serving environments. In this exercise, you will learn about deploying virtual systems, or topologies. In virtual system patterns, you define the exact topology you want for your environment, based on the parts that are available in Hypervisor Edition images. For example, WebSphere Application Server Hypervisor Edition provides parts such as a deployment manager, HTTP server, and custom nodes that you can use to build a network deployment cluster.

Virtual system patterns offer rich customization capabilities, many of which you will use in this exercise:

- Defining the specific topology required for your environment, including the number, type, and product version of each component

- Using script packages to customize the topology components, including performing administrative tasks such as defining application servers and installing applications

- Using add-ons to customize virtual machines at the operating system layer; add-ons are special scripts that include hypervisor-level API calls that can perform advanced hardware-level customizations (for example, creating a NIC or a virtual disk)

In this exercise, you will use the graphical Pattern Editor to define your virtual system pattern (topology), and customize your topology with several script packages and add-ons. The script packages you use will work together to install a sample application into your network deployment cell. You will include two add-ons in your pattern – one to create an additional virtual disk on your deployment manager, and a custom add-on to create an auditor user account in the application server. You will learn how to deploy your pattern into the cloud, and then go through and verify the configuration of your topology – including validating the configuration performed by the add-ons in your pattern and accessing the installed sample application.

By codifying your topology's requirements in a virtual system pattern, you are able to quickly, repeatedly deploy instances of your topology into the cloud. Depending on the size of your topology and the speed of your systems and network, a new topology instance can be ready for your use in as little as a few minutes.

Prerequisites:

- IBM Workload Deployer lab files bundle, extracted on the workstation where you are running this lab; the exercise will refer to this directory as **<LAB_FILES>/VirtualSystemDeploymentLab**

- User-level credentials to an IBM Workload Deployer V3.0 device with an ESX cloud defined for deployments (minimal permissions of create / deploy patterns and create new catalog content)

    - Your appliance credentials will be of the form user<N>, where <N> is your user number – for example, if you are using ID user04, then your user number, <N>, is 4. Make a note of your user number because you will use it throughout the exercise

- Access to Firefox V3.0 or higher, to access the web console for IBM Workload Deployer

- 2GB of virtual memory, 2 virtual CPU, and 2 IP addresses for deployments

1.1 Build a virtual system pattern that uses script packages

IBM Workload Deployer provides a way to virtualize and standardize your topologies for use in a private cloud, by letting you codify your environments as virtual system patterns. In this section of the exercise, you will define several script packages, and then use the graphical Pattern Editor to define your middleware topology using the script packages that you built.

1.1.1 Define script packages

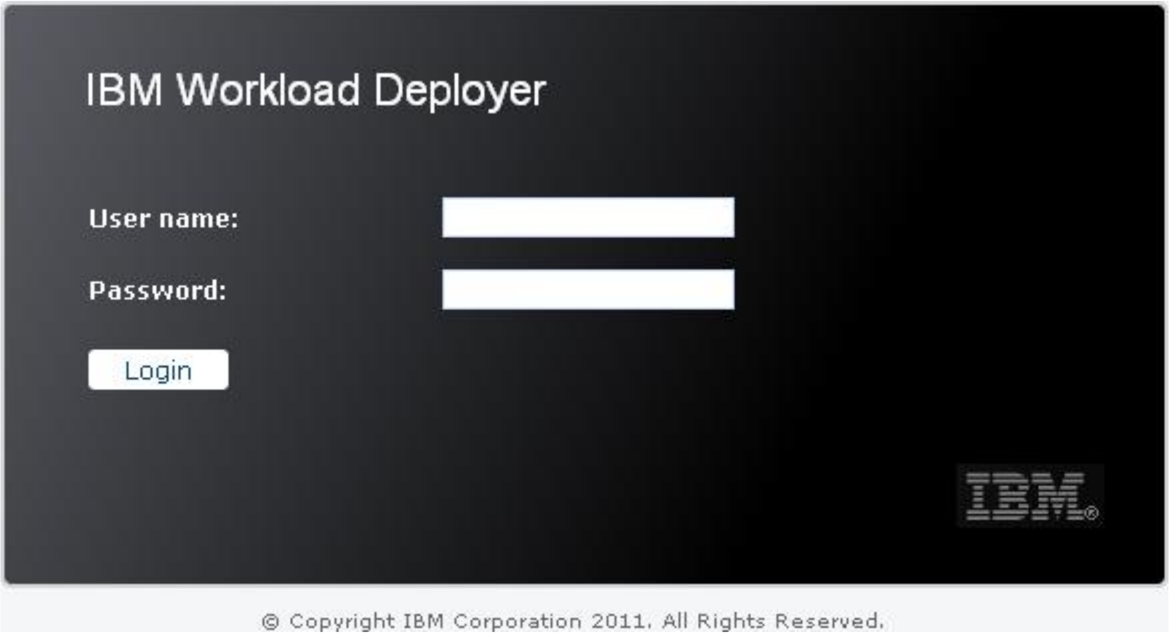
Script packages provide a rich customization framework for your virtual system patterns, allowing you to perform WebSphere administrative tasks, call into shell scripts, or invoke any other program required to customize your deployed system. In this exercise, you will define two script packages that work together to install a sample application in the small cell pattern that you will be creating.

- **Create Servers:** This script package defines an application server instance on all of the custom nodes in your cell; in this sample, there will just be one custom node in the virtual system pattern that you create

- **Install Application:** This script package installs a simple web application on the application server instance running on the custom node in your cell

__1. Log in to IBM Workload Deployer.

__a. Open a new window in your browser and point it to the IBM Workload Deployer at **https://<IWD_IP>/login/**



__b. Log in with the user-level credentials on your lab information sheet. Your **User name** will be of the form **user<N>**, where **<N>** is your user number. For example, if your ID is user99, then your <N> value is 9. Make a note of this number because you will use it in all of the names of your configuration objects on the appliance. Provide your **Password** (also available on your lab information sheet), then click **Login**.

__2. Define some script packages.

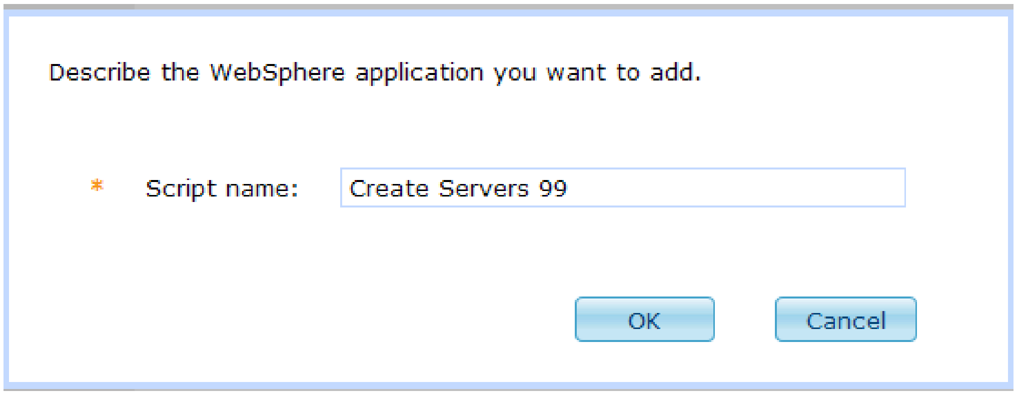
In this lab, you will use pre-built script packages that have been provided with the lab materials. Make sure that you have the lab files content available on your local workstation. Locate the folder **<LAB_FILES>/VirtualSystemDeploymentLab**. It contains two ZIP files (**appInstall.zip** and **serverCreate.zip**). You will use these to create your script packages.

__a. Define the **Create Servers** script package.

For the two script packages, you will upload a ZIP archive file. These script package archives contain a special JSON object, cbscript.json, that has the required configuration data for the script package. The JSON object is recognized by IBM Workload Deployer and is used to configure the script package. This means that you do not have to enter all of the parameters by hand; the available fields to configure the script package will be automatically populated based on the content of the cbscript.json file. The Create Servers script package, which you are defining in this step of the exercise, will create application server instances in the custom nodes of the cell that you will define in your pattern.

__i. In the console, if you are not already there, click on the **Catalog** tab located on the ribbon at the top of the page. From there choose **Script Packages**.

__ii. Click the add icon (✚). In the **Script name** field enter **Create Servers <N>**, where <N> is a replacement variable representing your user ID number. For example, if your user ID for the appliance is user99, your <N> value is 99 and you would call the script package Create Servers 99.



Describe the WebSphere application you want to add.

* Script name: Create Servers 99

__iii. Click the **OK** button. Now you have a script package configuration object defined and you can upload your script archive.

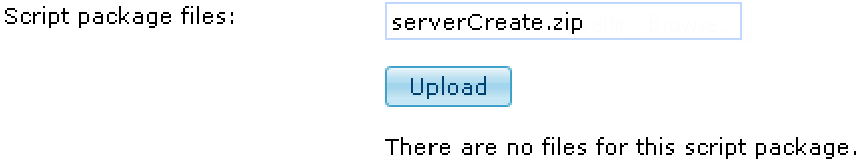
__b. Upload the script content.

For this script package, you will use the archive **serverCreate.zip** that contains the source and configuration information for the script package you are defining. There are two files in the archive:

- **cbscript.json** – a JSON object that is used to define the script package parameters for the script package.
- **serverCreate.jy** – a wsadmin script that runs from your deployment manager and creates an application server on each of the custom nodes in your cell.

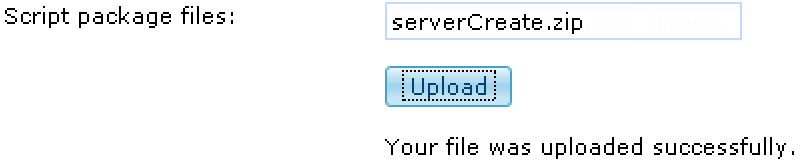
__i. Locate the **Script package files** section of the Create Servers <N> script package. Click inside the text field where it says **Browse…**.

__ii. This brings up a file dialog box. Browse to the location of the **<LAB_FILES>/ VirtualSystemDeploymentLab/serverCreate.zip** file and click **Open**.


Script package files: serverCreate.zip

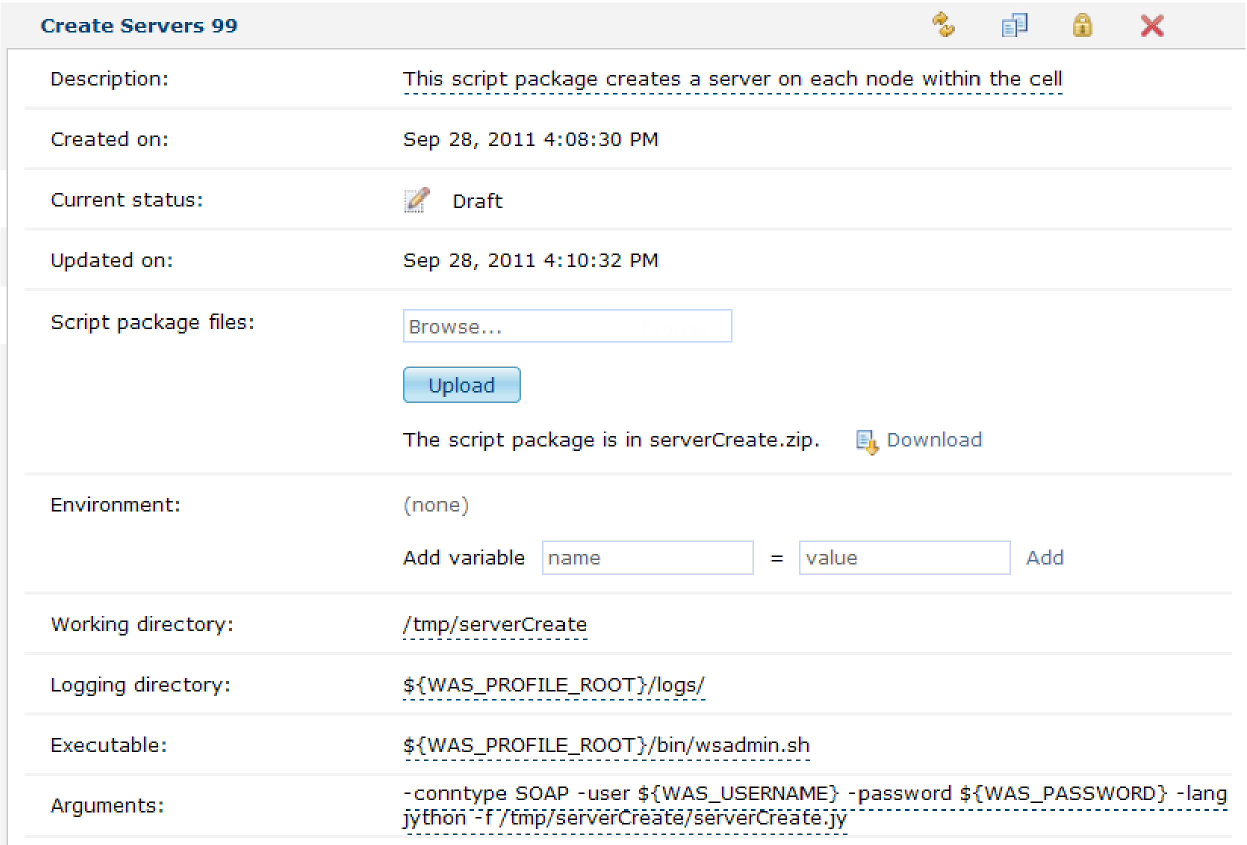
There are no files for this script package.

__iii. Click the **Upload** button. A spinning blue icon appears beside the button to indicate that the file is being uploaded to the appliance. When the upload has completed, the status message will change to **Your file was uploaded successfully**.


Script package files: serverCreate.zip

Your file was uploaded successfully.

__iv. In the upper right toolbar for the script package, click the **Refresh** icon (🔄). After refreshing, your display should contain the parameters as shown in the screen capture below. The configuration parameters for this script package were populated based on the content of the **cbscript.json** file in the **serverCreate.zip** archive.



Create Servers 99

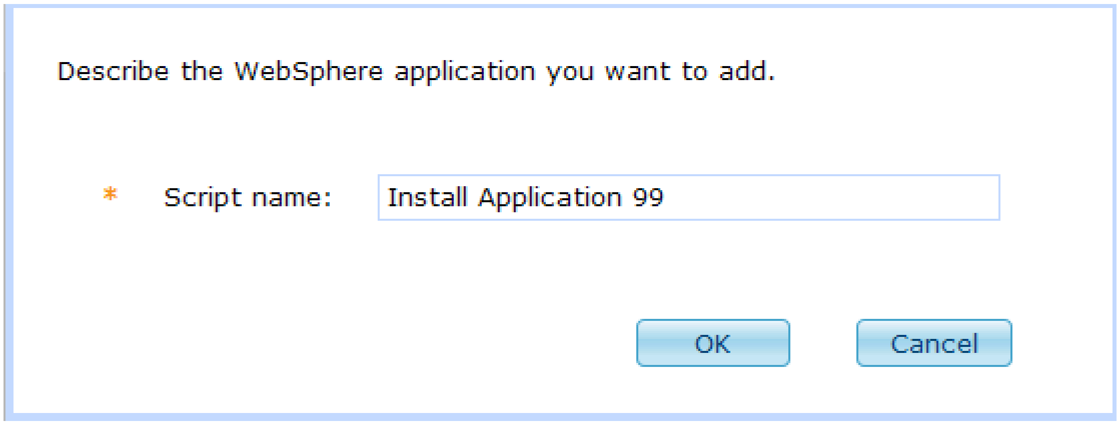
Description:	This script package creates a server on each node within the cell
Created on:	Sep 28, 2011 4:08:30 PM
Current status:	Draft
Updated on:	Sep 28, 2011 4:10:32 PM
Script package files:	Browse... Upload The script package is in serverCreate.zip. Download
Environment:	(none) Add variable name = value Add
Working directory:	/tmp/serverCreate
Logging directory:	${WAS_PROFILE_ROOT}/logs/
Executable:	${WAS_PROFILE_ROOT}/bin/wsadmin.sh
Arguments:	-conntype SOAP -user ${WAS_USERNAME} -password ${WAS_PASSWORD} -lang jython -f /tmp/serverCreate/serverCreate.jy

__c. Define the **Install Application** script package.

This script package will install a simple web application into your network deployment cell.

__i. In the console, if you are not already there, click on the **Catalog** tab located on the ribbon at the top of the page. From there choose **Script Packages**.

__ii. Click the add icon (✚). In the **Script name** field enter **Install Application <N>**, where <N> is a replacement variable representing your user ID number.



Describe the WebSphere application you want to add.

* Script name: Install Application 99

__iii. Click the **OK** button.

__d. Upload the script content.

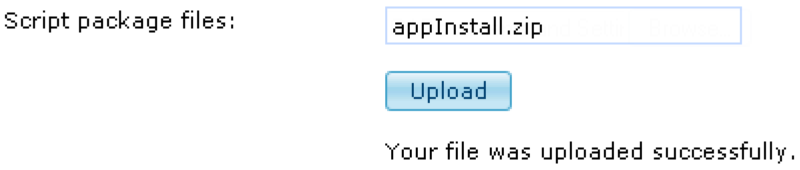
The **appInstall.zip** file contains the source for the script package you are defining. There are three files in the archive:

- **cbscript.json** – a JSON object that is used to define the script package parameters for the script package.
- **appInstall.jy** – a wsadmin script that runs from your deployment manager and installs the dgreeting.ear sample application onto the application server in your cell.
- **dgreeting.ear** – the sample application that the appInstall.jy script is going to install on your server.

__i. Locate the **Script package files** section for your **Install Application** script package. Click inside the text field where it says **Browse….**

__ii. This brings up a file dialog box. Browse to the location of the **<LAB_FILES>/ VirtualSystemDeploymentLab/appInstall.zip** file and click **Open**.

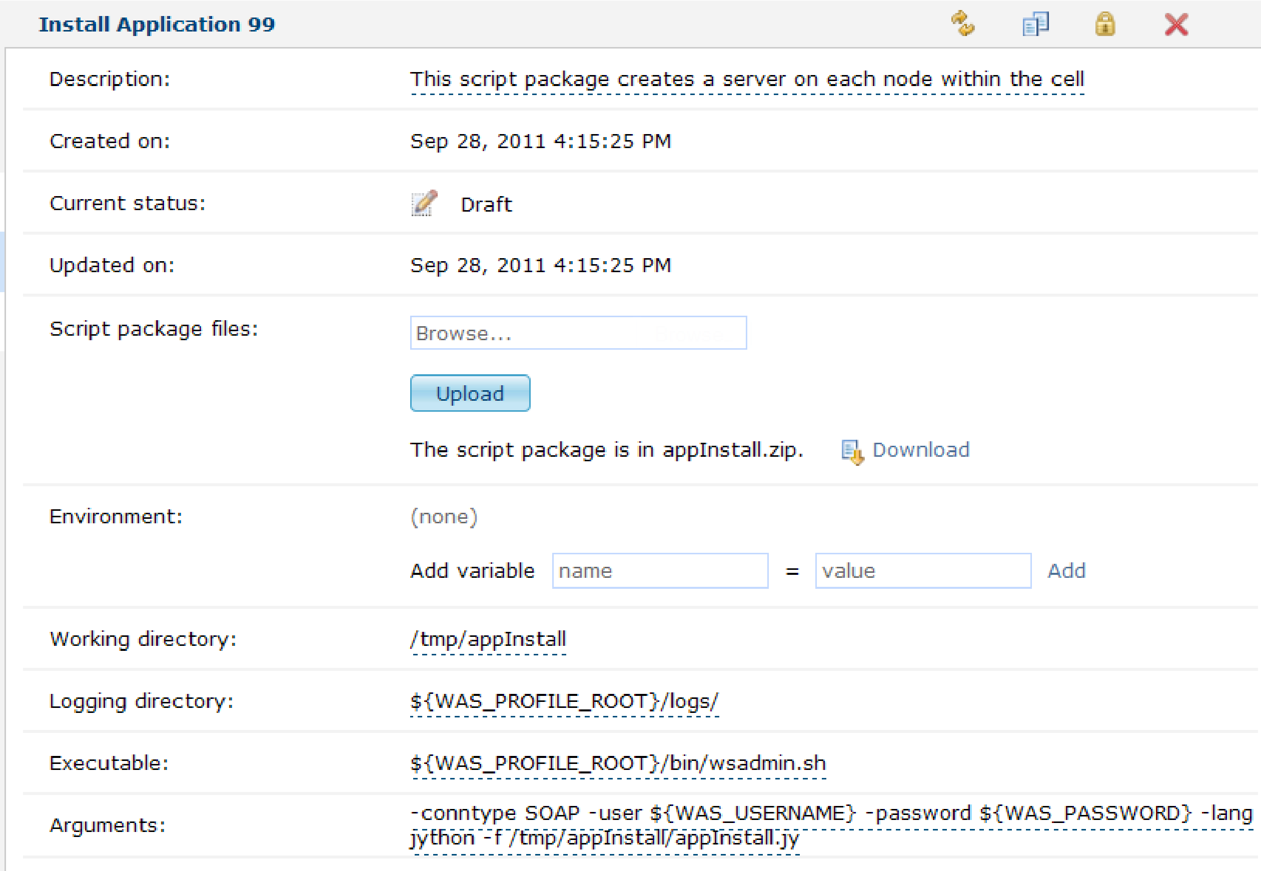
__iii. Click the **Upload** button. A spinning blue icon appears beside the button to indicate that the file is being uploaded to the appliance. When the upload has completed, the status message will change to **Your file was uploaded successfully**.



Script package files: appInstall.zip

Your file was uploaded successfully.

__iv. In the upper right toolbar for the script package, click the refresh icon (🔄). After refreshing, your display should contain the parameters as shown in the screen capture below:



Install Application 99

Description: This script package creates a server on each node within the cell

Created on: Sep 28, 2011 4:15:25 PM

Current status: Draft

Updated on: Sep 28, 2011 4:15:25 PM

Script package files: Browse...

Upload

The script package is in appInstall.zip. Download

Environment: (none)

Add variable name = value Add

Working directory: /tmp/appInstall

Logging directory: ${WAS_PROFILE_ROOT}/logs/

Executable: ${WAS_PROFILE_ROOT}/bin/wsadmin.sh

Arguments: -conntype SOAP -user ${WAS_USERNAME} -password ${WAS_PASSWORD} -lang jython -f /tmp/appInstall/appInstall.jy

__e. At this point, you have created all of your script packages, and you are ready to build a pattern that uses them.

1.1.2 Create a virtual system pattern that uses script packages

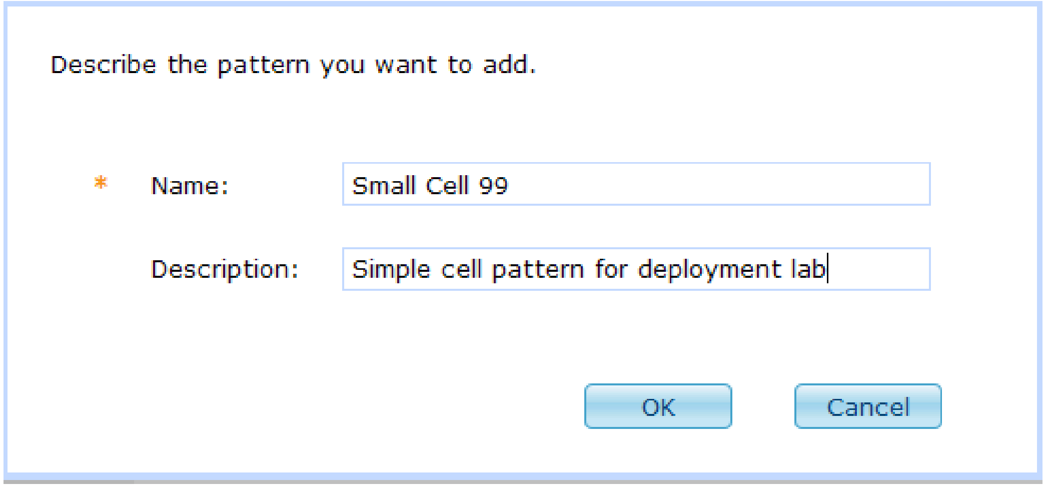
For this exercise, you will define a simple pattern for a network deployment cell that contains one deployment manager and one custom node. All of the script packages you defined in the previous section are designed to run on the deployment manager, so you will use the Pattern Editor to add these scripts to the deployment manager part in your pattern.

__1. Create a pattern for your network deployment cell.

__a. In the IBM Workload Deployer web console, navigate to **Patterns > Virtual Systems**.

__b. Click the **New** icon (➕) to create a new pattern.

__c. In the **Name** field, enter **Small Cell <N>**, where <N> is your user ID number. Optionally enter a **Description**, then click **OK**.

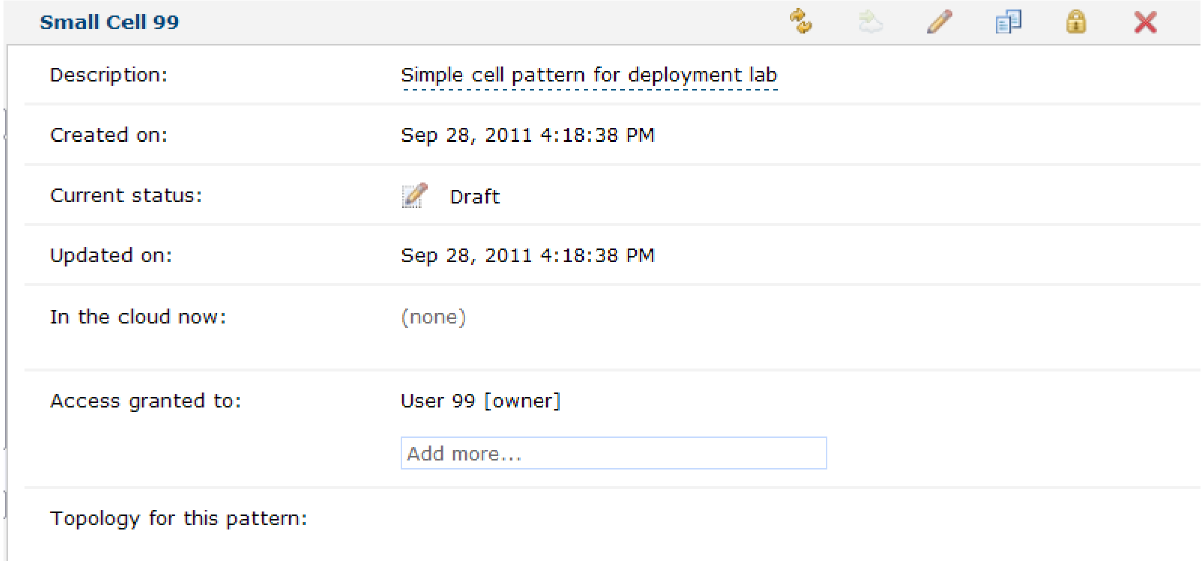


Describe the pattern you want to add.

* Name: Small Cell 99

Description: Simple cell pattern for deployment lab

__d. Initially, your new pattern will be empty. Click the **Edit** button (✎) in the upper right toolbar to open the Pattern Editor.



Small Cell 99

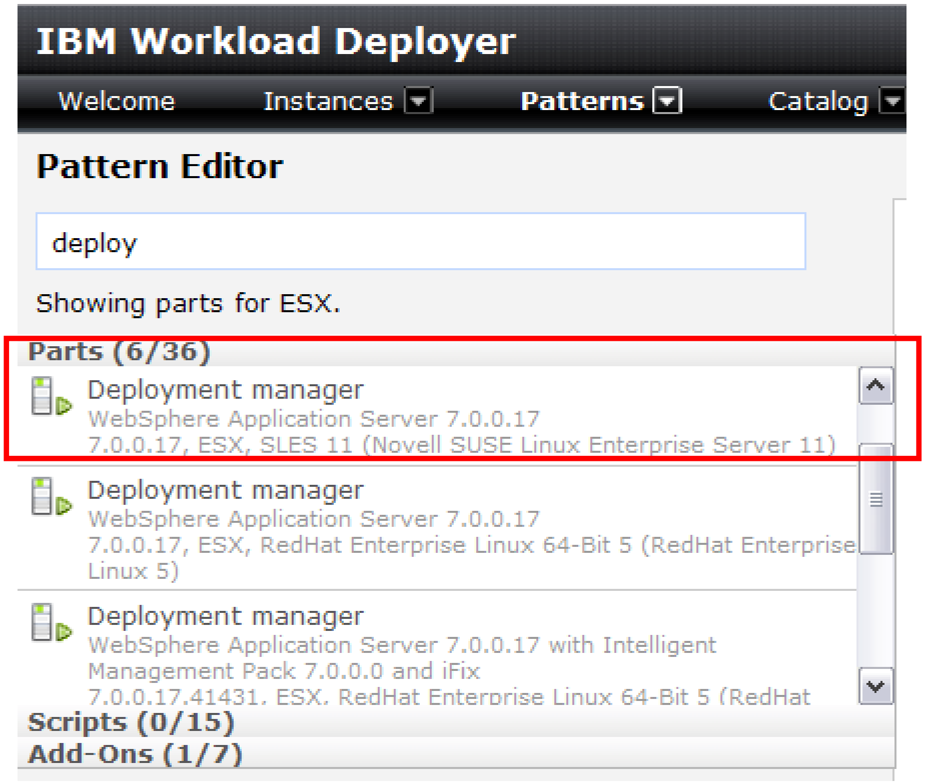
Description:	Simple cell pattern for deployment lab
Created on:	Sep 28, 2011 4:18:38 PM
Current status:	Draft
Updated on:	Sep 28, 2011 4:18:38 PM
In the cloud now:	(none)
Access granted to:	User 99 [owner] Add more...
Topology for this pattern:	

__2. Add parts to your pattern.

Your pattern will use a deployment manager and one custom node.

__a. In the **Pattern Editor**, locate the **Search** field in the upper left of the pattern palette. Click the text **Search…** and type **deployment** into the search field.

__b. Using the search field allows you to filter the list of parts shown. You will see multiple deployment manager parts in the palette. Locate the **Deployment manager** part for **WebSphere Application Server V7.0.0.17** for **ESX, SLES 11** (example shown below).



deploy

Showing parts for ESX.

Parts (6/36)

Deployment manager
WebSphere Application Server 7.0.0.17
7.0.0.17, ESX, SLES 11 (Novell SUSE Linux Enterprise Server 11)

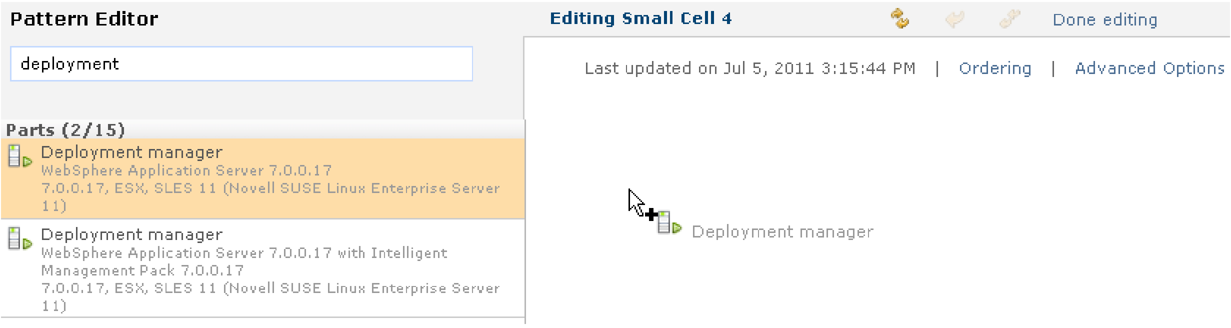
Deployment manager
WebSphere Application Server 7.0.0.17
7.0.0.17, ESX, RedHat Enterprise Linux 64-Bit 5 (RedHat Enterprise Linux 5)

Deployment manager
WebSphere Application Server 7.0.0.17 with Intelligent Management Pack 7.0.0.0 and iFix
7.0.0.17.41431, ESX, RedHat Enterprise Linux 64-Bit 5 (RedHat

Scripts (0/15)

Add-Ons (1/7)

__c. Click the **Deployment manager** part (WebSphere Application Server V7.0.0.17, ESX, SLES 11), drag it across to the right, then drop it on the pattern canvas.



deployment

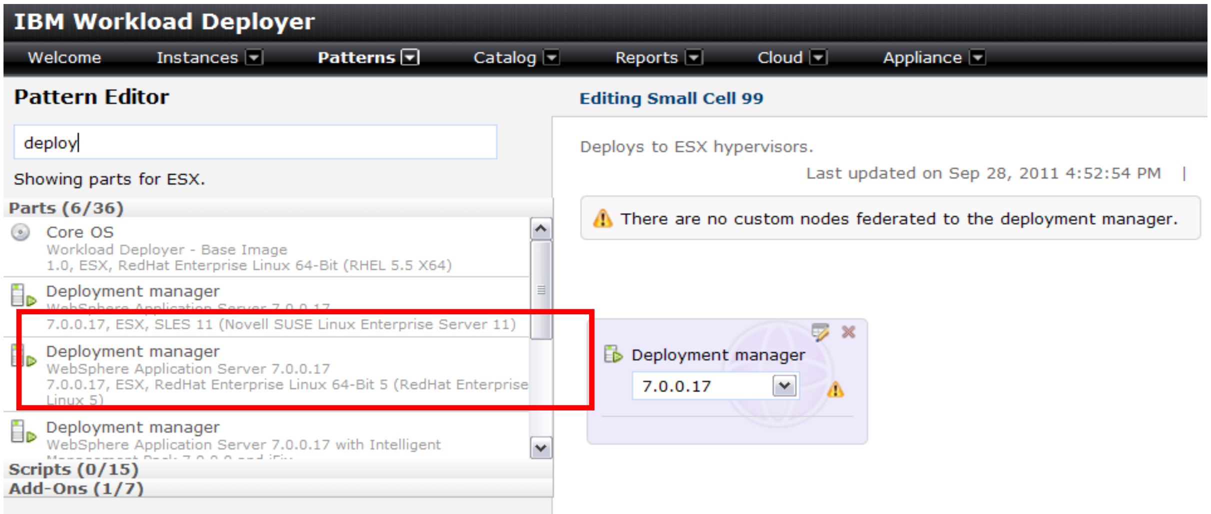
Last updated on Jul 5, 2011 3:15:44 PM

Parts (2/15)

Deployment manager
WebSphere Application Server 7.0.0.17
7.0.0.17, ESX, SLES 11 (Novell SUSE Linux Enterprise Server 11)

Deployment manager
WebSphere Application Server 7.0.0.17 with Intelligent Management Pack 7.0.0.17
7.0.0.17, ESX, SLES 11 (Novell SUSE Linux Enterprise Server 11)

__d. After a moment, the deployment manager part will be added to your pattern, and you will see it displayed in the pattern canvas with a warning message, letting you know that **There are no custom nodes federated to the deployment manager**. You will be adding a custom node in the next steps of the exercise.



Pattern Editor

deploy

Showing parts for ESX.

Parts (6/36)

Core OS
Workload Deployer - Base Image
1.0, ESX, RedHat Enterprise Linux 64-Bit (RHEL 5.5 X64)

Deployment manager
WebSphere Application Server 7.0.0.17
7.0.0.17, ESX, SLES 11 (Novell SUSE Linux Enterprise Server 11)

Deployment manager
WebSphere Application Server 7.0.0.17
7.0.0.17, ESX, RedHat Enterprise Linux 64-Bit 5 (RedHat Enterprise Linux 5)

Deployment manager
WebSphere Application Server 7.0.0.17 with Intelligent
Management Pack 7.0.0.0 and iFix

Scripts (0/15)
Add-Ons (1/7)

Editing Small Cell 99

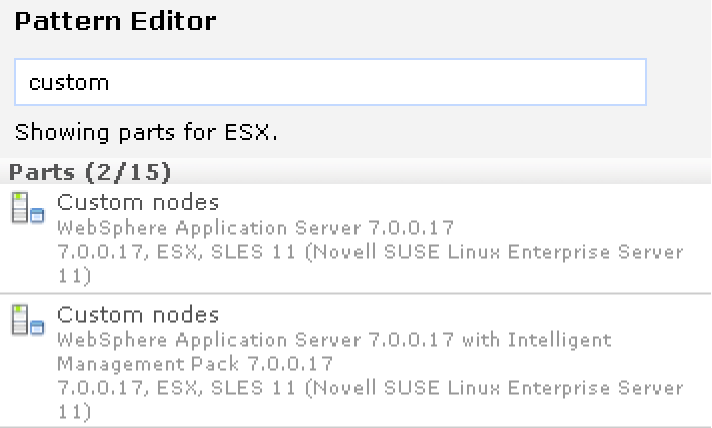
Deploys to ESX hypervisors.

Last updated on Sep 28, 2011 4:52:54 PM |

⚠ There are no custom nodes federated to the deployment manager.

Deployment manager
7.0.0.17 ⚠

__e. Back in the palette on the left, go back to the **Search…** field. Remove the text deployment and type **custom** into the search field. You will see several custom node parts listed in the palette.



Pattern Editor

custom

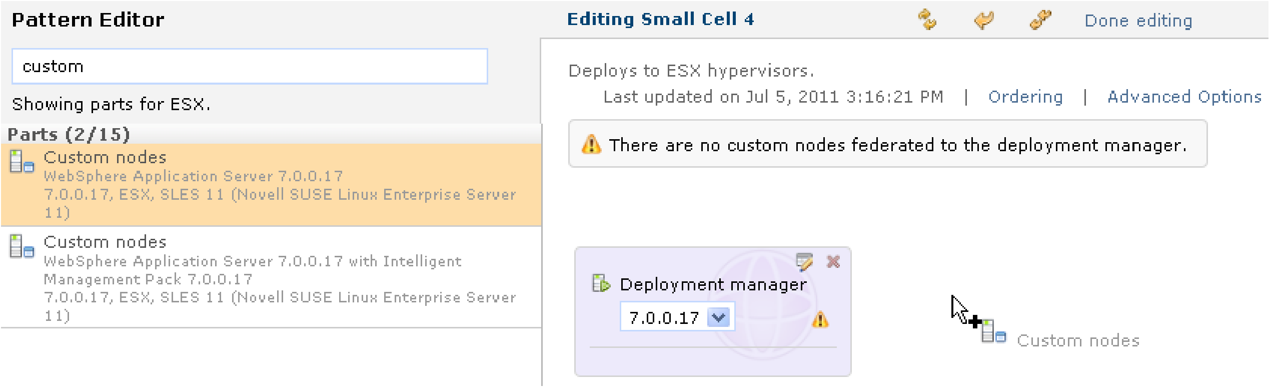
Showing parts for ESX.

Parts (2/15)

Custom nodes
WebSphere Application Server 7.0.0.17
7.0.0.17, ESX, SLES 11 (Novell SUSE Linux Enterprise Server 11)

Custom nodes
WebSphere Application Server 7.0.0.17 with Intelligent Management Pack 7.0.0.17
7.0.0.17, ESX, SLES 11 (Novell SUSE Linux Enterprise Server 11)

__f. Locate the **Custom nodes** part for **WebSphere Application Server V7.0.0.17** for **ESX, SLES 11**. Click that part, then drag it across to drop on the pattern canvas.



Pattern Editor

custom

Showing parts for ESX.

Parts (2/15)

Custom nodes
WebSphere Application Server 7.0.0.17
7.0.0.17, ESX, SLES 11 (Novell SUSE Linux Enterprise Server 11)

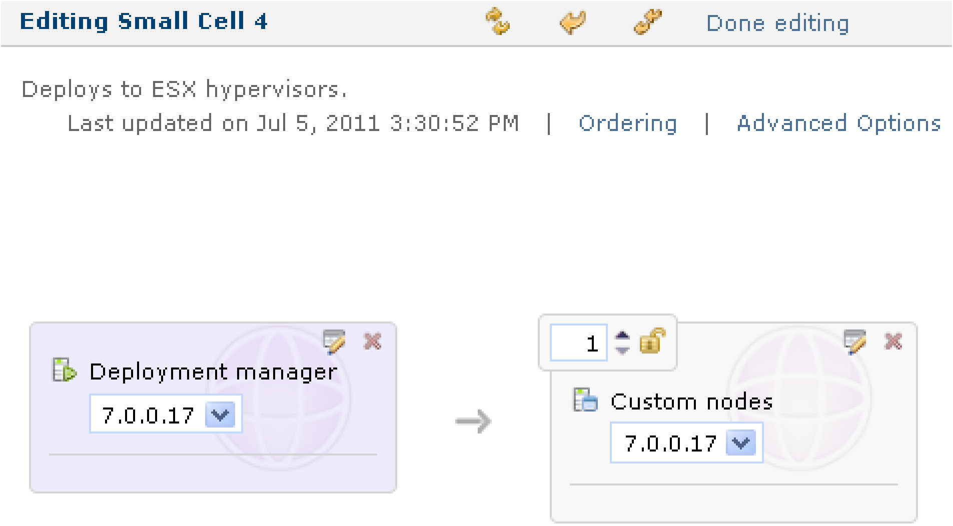
Custom nodes
WebSphere Application Server 7.0.0.17 with Intelligent Management Pack 7.0.0.17
7.0.0.17, ESX, SLES 11 (Novell SUSE Linux Enterprise Server 11)

Editing Small Cell 4

Deploys to ESX hypervisors.
Last updated on Jul 5, 2011 3:16:21 PM | Ordering | Advanced Options

There are no custom nodes federated to the deployment manager.

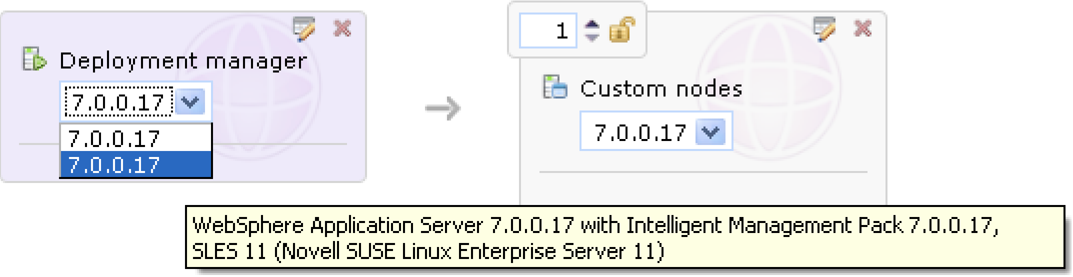
__g. After a moment, your custom node will be displayed in the canvas, with an arrow connecting the deployment manager and the node, indicating that this node will be federated into your deployment manager's cell.



Editing Small Cell 4

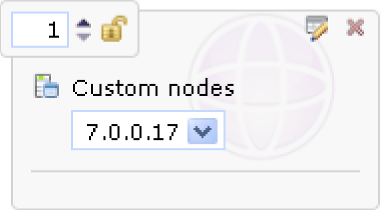
Deploys to ESX hypervisors.
Last updated on Jul 5, 2011 3:30:52 PM | Ordering | Advanced Options

__h. In the pattern canvas, you have the option to change some settings for your parts. For example, if you select the **version drop down menu** for the deployment manager, you will see that you can easily switch the source image for the part, just by selecting a different option from the menu (hover over the items in the drop down menu to view context help that gives a verbose description of the source image). In this case, you could switch this part to use the Intelligent Management Pack image. For this exercise, **do not change the image version** – keep the standard V7.0 image as the source (the first one in the list).



WebSphere Application Server 7.0.0.17 with Intelligent Management Pack 7.0.0.17, SLES 11 (Novell SUSE Linux Enterprise Server 11)

__i. On the custom node part, notice the **counter** in the upper left. If you want to have multiple custom nodes in your pattern, all you need to do is adjust the counter to the appropriate number of nodes (either using the click buttons, or by typing the desired number into the field). For this exercise, a single custom node is sufficient, so leave the counter set to **1**.

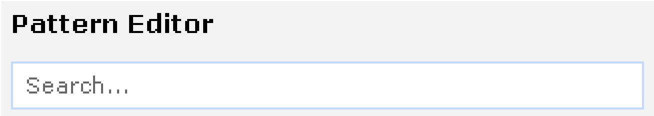


__j. Also notice the lock symbol next to the node counter. If you want to prohibit users from adding extra nodes to this virtual system post-deployment, you can click the lock to restrict cloning this node later. For this exercise, you do not need to lock the node count.

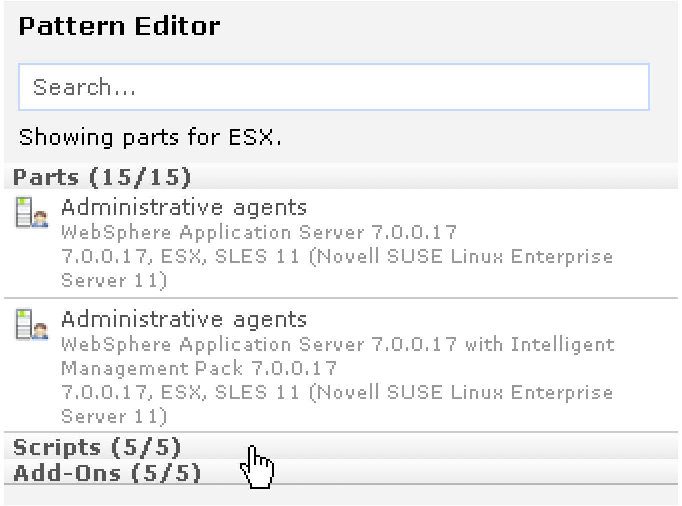
__k. Now you have defined your topology, you are ready to add your script packages to the pattern.

__3. Add script packages to your pattern.

__a. In the pattern editor, go back to **Search** field in the upper left and clear out the search filter text (currently set to “custom”).



__b. In the palette on the left side, locate the **Scripts** section and click the grey bar to expand it.



Pattern Editor

Search...

Showing parts for ESX.

Parts (15/15)

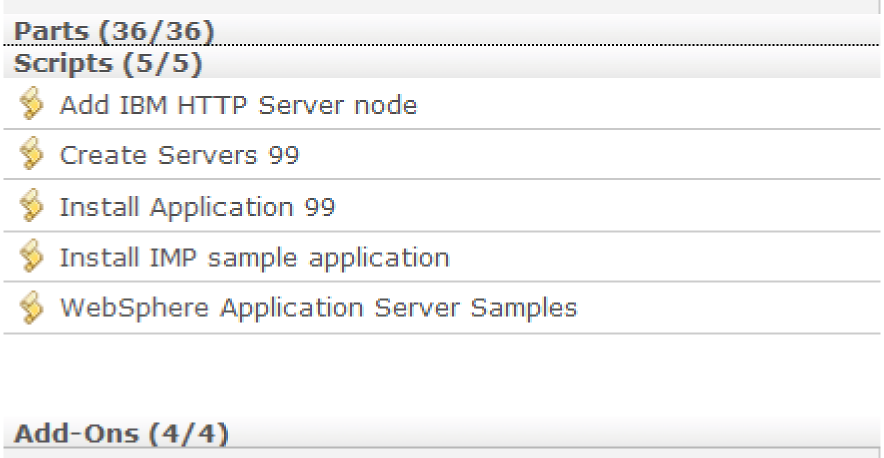
Administrative agents
WebSphere Application Server 7.0.0.17
7.0.0.17, ESX, SLES 11 (Novell SUSE Linux Enterprise Server 11)

Administrative agents
WebSphere Application Server 7.0.0.17 with Intelligent Management Pack 7.0.0.17
7.0.0.17, ESX, SLES 11 (Novell SUSE Linux Enterprise Server 11)

Scripts (5/5)

Add-Ons (5/5)

__c. In addition to the pre-loaded sample scripts, you should see your two script packages here in the list: **Create Servers <N>**, and **Install Application <N>.** These script packages have been designed to run on the deployment manager in your cell.



Parts (36/36)

Scripts (5/5)

Add IBM HTTP Server node

Create Servers 99

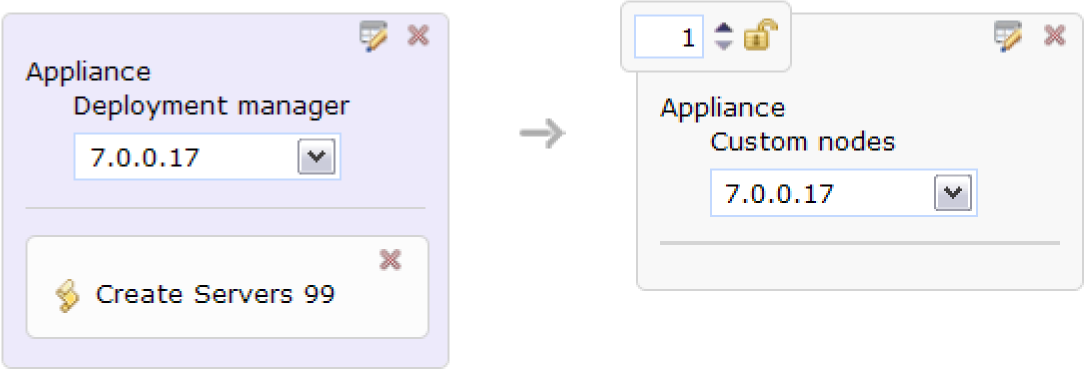
Install Application 99

Install IMP sample application

WebSphere Application Server Samples

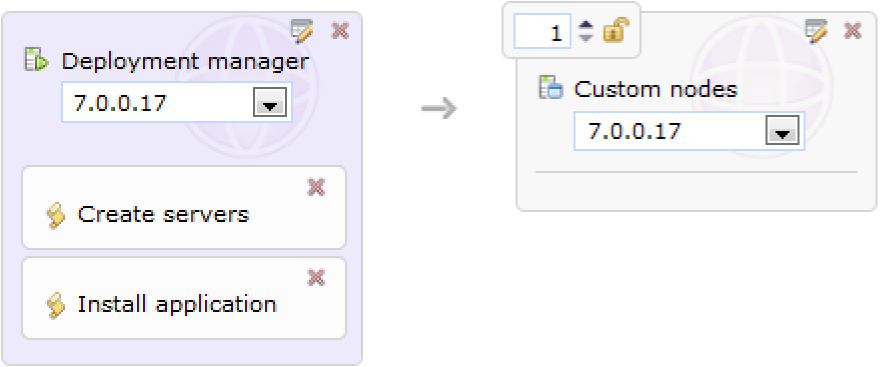
Add-Ons (4/4)

__d. Select the **Create Servers <N>** script package from the left, drag it across to the pattern canvas and drop it onto your **Deployment manager** part, At this point, your pattern should appear like this:



__e. Select the **Install Application <N>** from the palette at left, drag it across to the pattern canvas, and drop it on top of the **Deployment manager** part.

__f. You will briefly see a message indicating that the script is being added to the part. When the add operation is completed, the script package **Install Application <N>** will appear on the **Deployment manager** part, as shown:

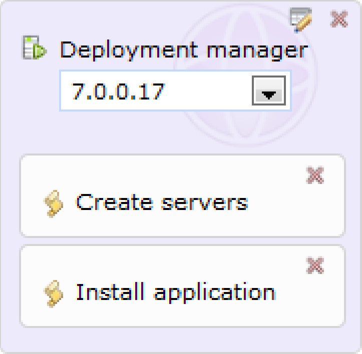


__g. Notice that the display order of the script packages in the pattern canvas. Make sure they are in the same order as shown above. If they are not you should remove them 'x' and read them in the correct order. There is an ordering function, but you will not be using it in this lab.

__4. Define configuration options for your pattern parts.

There are configuration options that are required in order to deploy your virtual system – for example, virtual hardware information for CPU and memory, and passwords for the user IDs created in each virtual machine. These options can be configured here in the pattern editor or at deployment time. You will configure some of your part settings here, in the pattern, and then these attributes will be included each time you deploy your pattern (you can also override these settings at deployment time, if you leave them unlocked).

__a. Back in the main pattern canvas, locate the **Properties** icon (✎) in the upper right of the **Deployment manager** part and select it.



Deployment manager

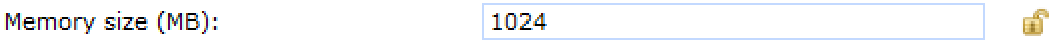
7.0.0.17

Create servers

Install application

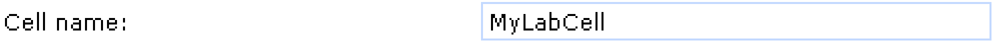
__b. In the properties menu, modify these values:

__i. In the **Memory size (MB)** field, use **1024**.



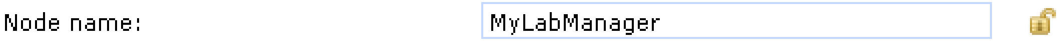
Memory size (MB): 1024

__ii. In the **Cell** name field, use **MyLabCell**.



Cell name: MyLabCell

__iii. In the **Node name** field, use **MyLabManager**.

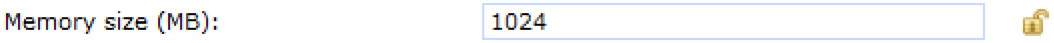


Node name: MyLabManager

__iv. Keep all of the other default values, and click **OK**.

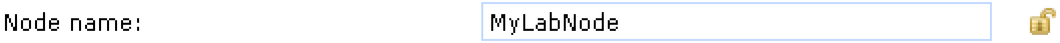
__c. Select the **Properties** icon (✎) for the **Custom node** part. In the properties menu, modify these values:

__i. In the **Memory size (MB)** field, use **1024**.



Memory size (MB): 1024

__ii. In the **Node** name field, use **MyLabNode**.



Node name: MyLabNode

__iii. Keep all of the other default values, and click **OK**.

__d. Now you have created a virtual system pattern that creates a small cell and uses several script packages. In the upper right of the editor click the text **Done editing** to save your changes and return to the pattern details page.

__e. Before you deploy your pattern, you will work through some advanced customizations using virtual machine add-ons.

1.2 Enhance a virtual system pattern with add-ons

Add-ons provide a way for you to do advanced virtual machine customization for your deployed environments. The appliance comes with three types of pre-loaded add-ons, for creating user accounts, NICs, and disks. Add-ons are able to offer operating system- and hardware-level configuration for your virtual machines because they include an internal layer that makes hypervisor-level API calls to perform some configuration actions. Add-ons, then, are very well-suited to creating new virtual hardware for a virtual machine, such as disks and network interfaces.

1.2.1 Create a custom virtual machine add-on

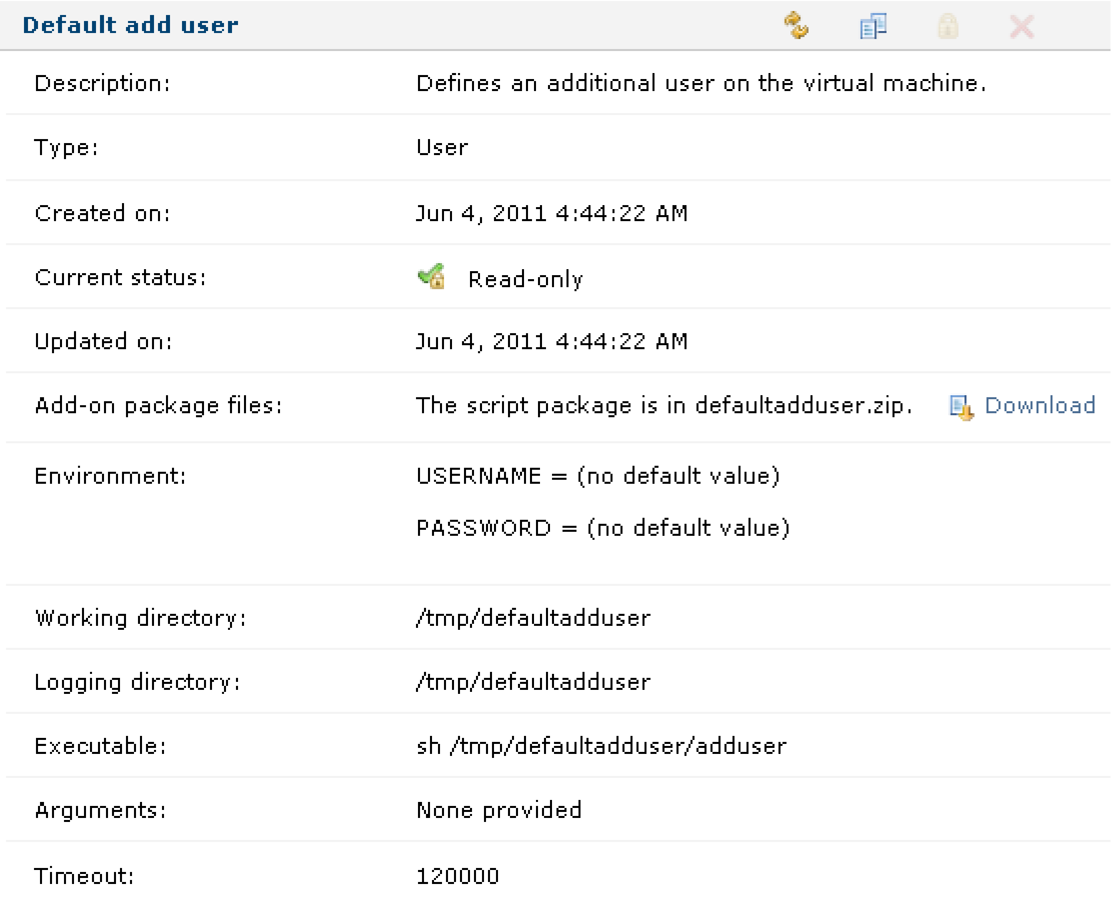
By default, the user add-on performs a simple adduser operation to create a new user account in the operating system of your VM. The default add-on exposes user name and password properties that you can define during deployment to create the new user. In this exercise, you will be customizing the default user add-on implementation and overriding its behavior with a custom script. Your new add-on will define a user account in the application server configuration, and then optionally give that user auditor privileges in the application server.

__1. Examine the configuration of the default user add-on.

__a. In the web console, navigate to **Catalog > Add-ons**.

__b. Select the **Default add user** add-on from the list at left. This add-on defines an operating system user account in your virtual machine.

__c. Review the add-on parameters. Notice that these are very similar to the properties you configured earlier when you were creating script packages.



Default add user

Description:	Defines an additional user on the virtual machine.
Type:	User
Created on:	Jun 4, 2011 4:44:22 AM
Current status:	Read-only
Updated on:	Jun 4, 2011 4:44:22 AM
Add-on package files:	The script package is in defaultadduser.zip. Download

Environment:	USERNAME = (no default value)
	PASSWORD = (no default value)

Working directory:	/tmp/defaultadduser
Logging directory:	/tmp/defaultadduser
Executable:	sh /tmp/defaultadduser/adduser
Arguments:	None provided
Timeout:	120000

__d. Notice that the **Add-on package files** field contains a link to **Download** the default implementation package for this add-on, called **defaultadduser.zip**. That implementation package contains two files:

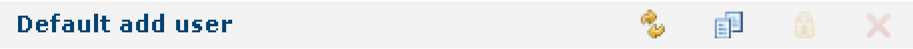
- **cbscript.json:** A configuration file that defines the add-on parameters (for example, add-on type, and script execution parameters) and environment variables for the script – in this case, a user ID and a password

- **adduser:** The script file that issues the adduser command to create an operating-system level user account and set its password

__e. In the next steps of the exercise, you will create your own user add-on and customize it with an implementation package, overriding the default add-on implementation.

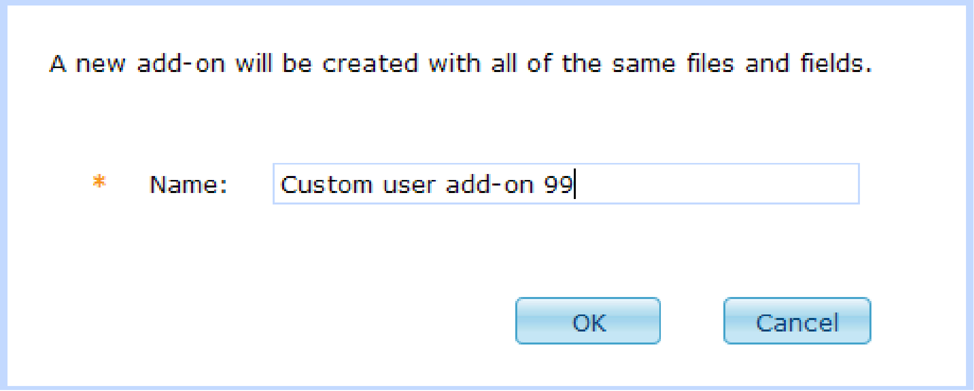
__2. Clone the existing user add-on.

__a. From the toolbar for the **Default add user** add-on, click the **Clone** button ().



Default add user

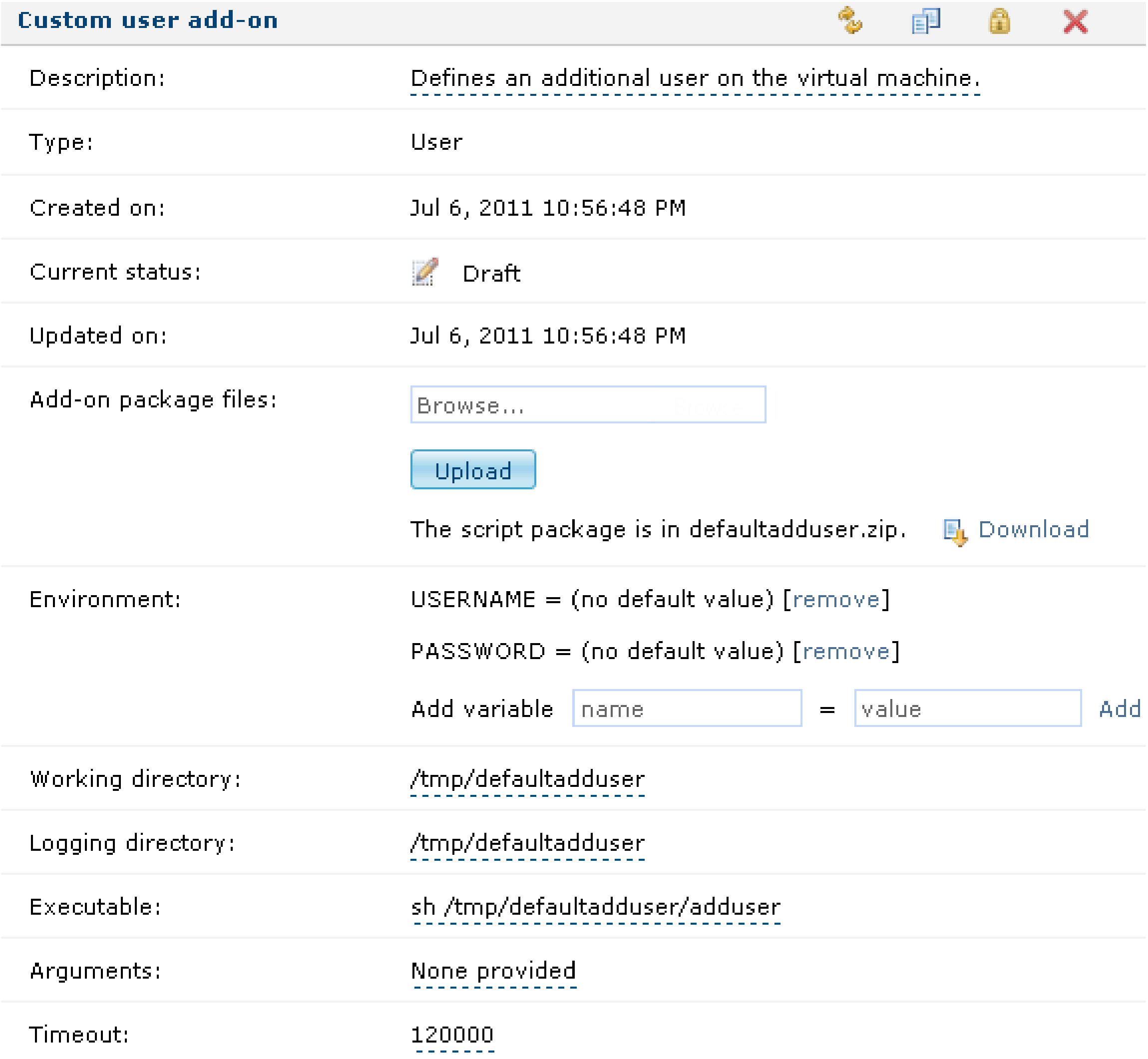
__b. In the **Name** field, use **Custom user add-on <N>**, where <N> is your user ID number, then click **OK**.



A new add-on will be created with all of the same files and fields.

* Name: Custom user add-on 99

__c. Initially, the configuration data for your add-on will match the default add-on – except that now many of the configuration properties are editable, including the add-on package files.



Custom user add-on

Description:	Defines an additional user on the virtual machine.
Type:	User
Created on:	Jul 6, 2011 10:56:48 PM
Current status:	Draft
Updated on:	Jul 6, 2011 10:56:48 PM
Add-on package files:	Browse... Upload The script package is in defaultadduser.zip. Download

Environment:	USERNAME = (no default value) [remove]
	PASSWORD = (no default value) [remove]
	Add variable name = value Add

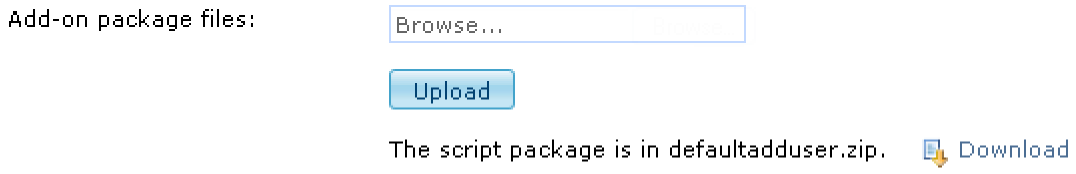
Working directory:	/tmp/defaultadduser
Logging directory:	/tmp/defaultadduser
Executable:	sh /tmp/defaultadduser/adduser
Arguments:	None provided
Timeout:	120000

__3. Upload a custom add-on implementation package.

An implementation package that you can use to customize this add-on has been pre-built and provided for you to use with this exercise. The implementation is based on the default user add-on, but it has been enhanced to perform additional application server configuration for the user being created, allowing you to optionally define the user in the application server as an auditor. There are three key differences in this enhanced add-on implementation from the default implementation:

- The **cbscript.json** file has been updated to include an additional key, called **WAS_AUDITOR**, for specifying whether to define the user as an application server auditor
- The **adduser** script now contains an additional call to a wsadmin script to perform the application server configuration for the application server user
- A file called **createuser.jy** has been added to the implementation package; this Jython script contains the wsadmin commands for creating the application server auditor user

__a. Locate the **Add-on package files** field for your cloned add-on, and click inside the field on the text **Browse….**

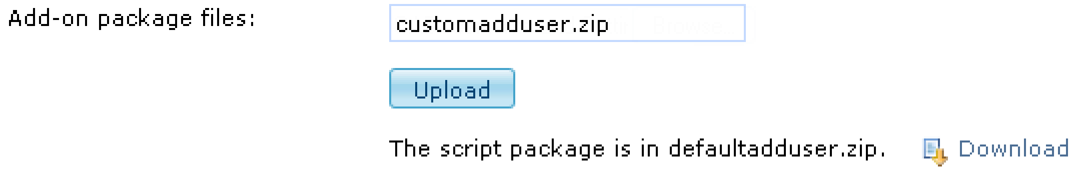


Add-on package files: Browse...

The script package is in defaultadduser.zip.

__b. In the file upload dialog, navigate to **<LAB_FILES>/VirtualSystemDeploymentLab**, then select **customadduser.zip** and click **Open**.

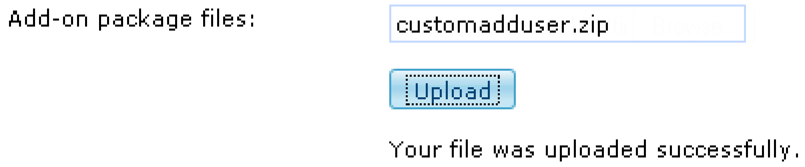
__c. Press the **Upload** button.



Add-on package files: customadduser.zip

The script package is in defaultadduser.zip.

__d. After a moment, the add-on display will update to indicate that your file was uploaded successfully.

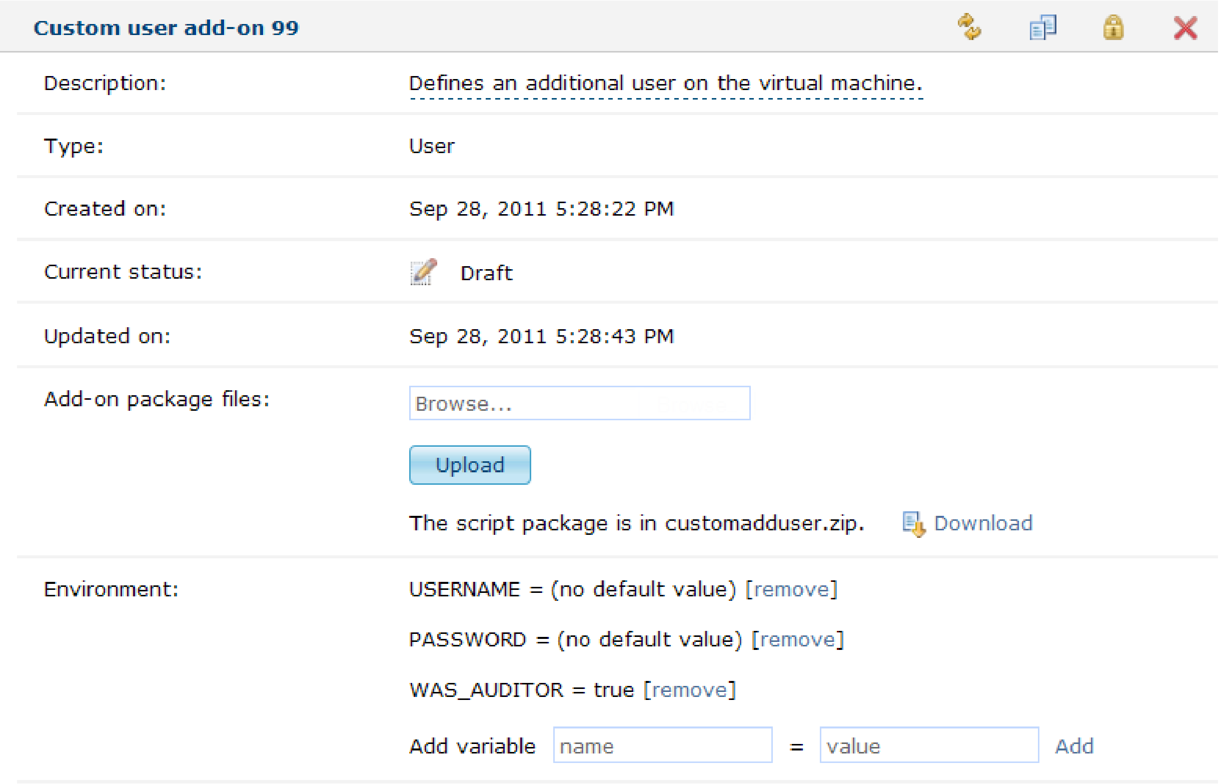


Add-on package files: customadduser.zip

Your file was uploaded successfully.

__e. In the add-on toolbar, click the **Refresh** button (🔄).

__f. At this point, the add-on details page will be automatically populated with the updated configuration data from the **cbscript.json** file in the **customadduser.zip** implementation package. Verify that your add-on display matches the screen capture shown below. Make a special note of the **WAS_AUDITOR** parameter in the **Environment** section. This is the flag (true or false) that your add-on will use to determine whether to define the user account in the application server as an auditor.



Custom user add-on 99

Description:	Defines an additional user on the virtual machine.
Type:	User
Created on:	Sep 28, 2011 5:28:22 PM
Current status:	✎ Draft
Updated on:	Sep 28, 2011 5:28:43 PM
Add-on package files:	Browse... Upload The script package is in customadduser.zip. Download

Environment:	USERNAME = (no default value) [remove]
	PASSWORD = (no default value) [remove]
	WAS_AUDITOR = true [remove]
	Add variable name = value Add

__g. Now your custom add-on is complete and ready to be used in a virtual system pattern.

1.2.2 Use virtual machine add-ons in your virtual system pattern

In this section, you will start with the Small Cell <N> pattern you created previously and enhance the deployment manager configuration with add-ons. You will add both your custom user add-on and the default disk add-on to the deployment manager. Using the Pattern Editor to incorporate add-ons into your pattern works just like you have already seen with script packages – simply drag the add-ons from the pattern palette onto the part on the canvas where you want to use the add-on.

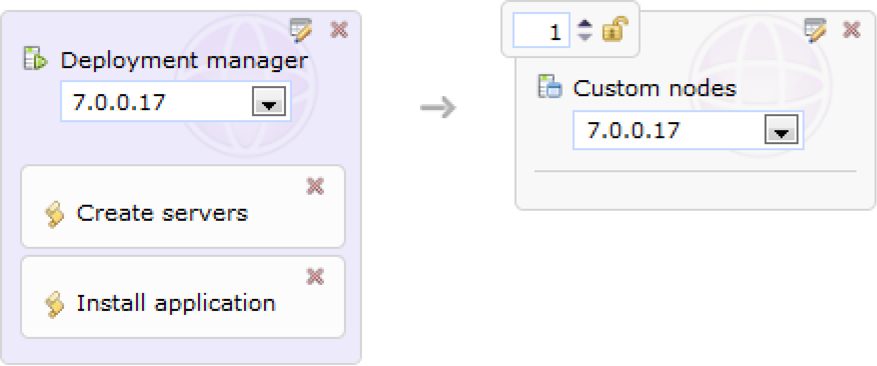
__1. Open the pattern editor for your pattern.

__a. Navigate to **Patterns > Virtual Systems** in the web console.

__b. Select your **Small Cell <N>** pattern from the list at left.

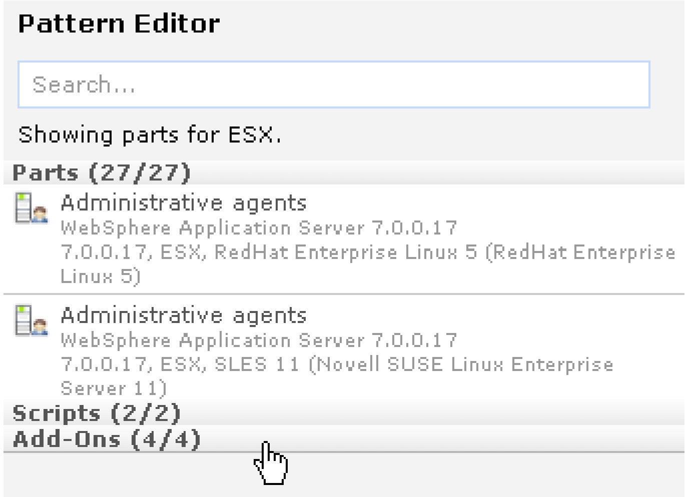
__c. Click the **Edit** button (✎) in the pattern toolbar to open the pattern editor.

__d. Your pattern should look just like you left it – a deployment manager with two script packages, and one custom node:



__2. Add some add-ons to the deployment manager part.

__a. Locate the **Add-Ons** section at the bottom of the palette and click the thin grey bar to expand the section.


Pattern Editor

Search...

Showing parts for ESX.

Parts (27/27)

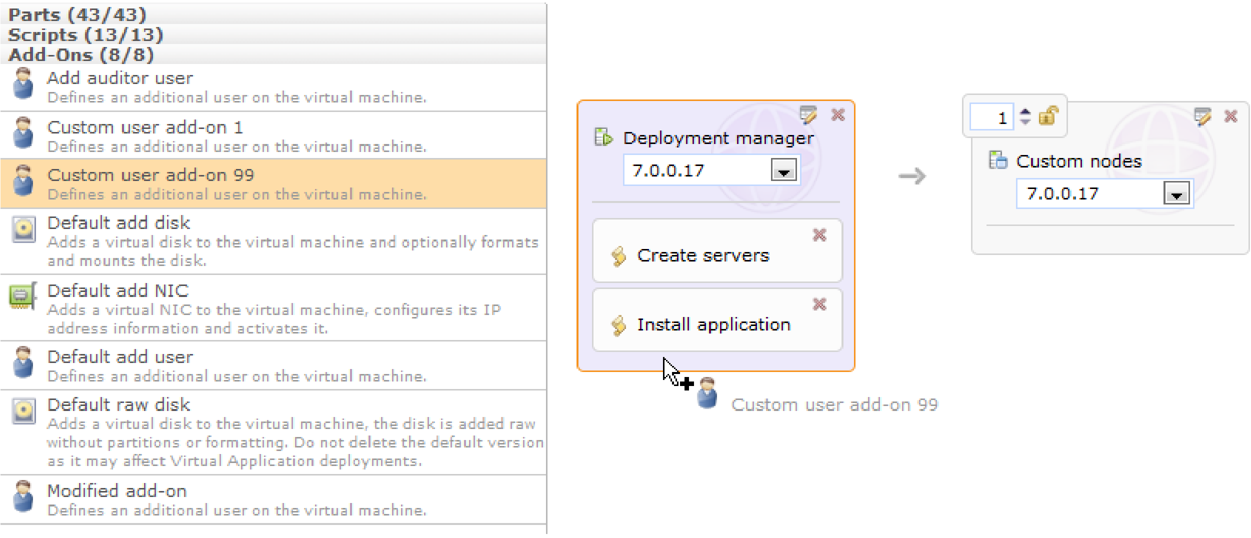
Administrative agents
WebSphere Application Server 7.0.0.17
7.0.0.17, ESX, RedHat Enterprise Linux 5 (RedHat Enterprise Linux 5)

Administrative agents
WebSphere Application Server 7.0.0.17
7.0.0.17, ESX, SLES 11 (Novell SUSE Linux Enterprise Server 11)

Scripts (2/2)

Add-Ons (4/4)

__b. In the list of **Add-Ons**, locate your **Custom user add-on <N>**. Drag that add-on from the palette and drop it on top of the **Deployment manager** part in your pattern canvas.



Parts (43/43)
Scripts (13/13)
Add-Ons (8/8)

Add auditor user
Defines an additional user on the virtual machine.

Custom user add-on 1
Defines an additional user on the virtual machine.

Custom user add-on 99
Defines an additional user on the virtual machine.

Default add disk
Adds a virtual disk to the virtual machine and optionally formats and mounts the disk.

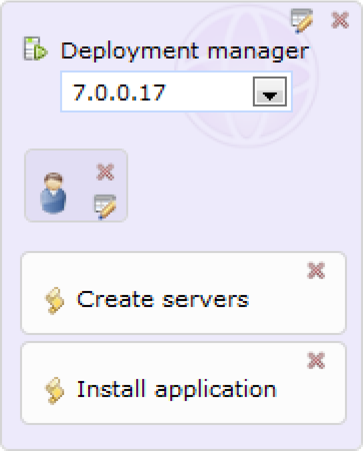
Default add NIC
Adds a virtual NIC to the virtual machine, configures its IP address information and activates it.

Default add user
Defines an additional user on the virtual machine.

Default raw disk
Adds a virtual disk to the virtual machine, the disk is added raw without partitions or formatting. Do not delete the default version as it may affect Virtual Application deployments.

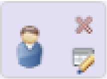
Modified add-on
Defines an additional user on the virtual machine.

__c. After a moment of processing, your add-on will be added to the deployment manager part, and your deployment manager will look like this:



__3. Look at the add-on properties for your add-on.

__a. The user add-on has configuration parameters, which you can tell by seeing the **Parameters** icon (📝) in the lower right of the add-on bubbles (as an example, the default NIC add-on, which is not being used in this exercise, does not have any user-configurable parameters). Click on the **Parameters** icon (📝) for the **Custom user add-on**.



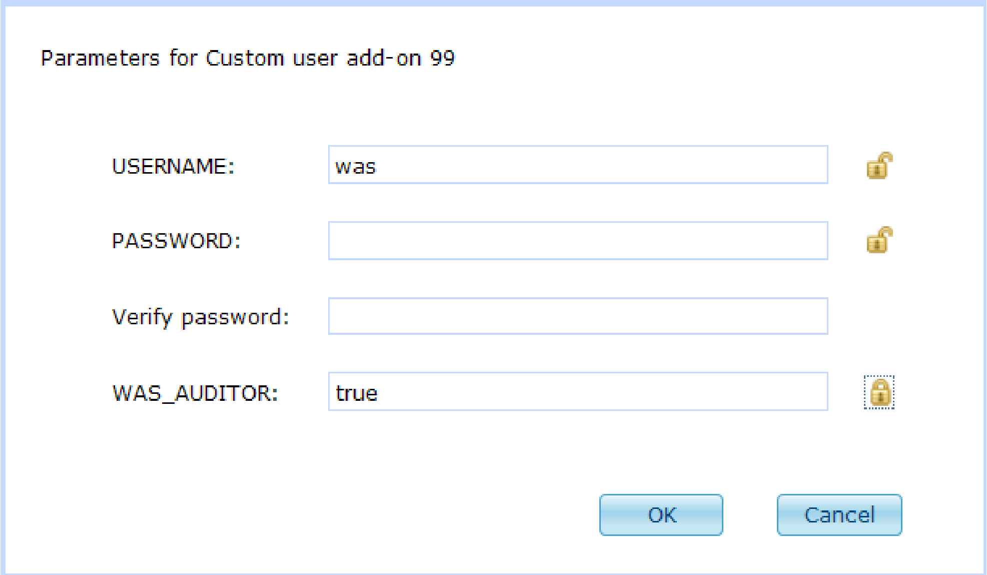
__b. In this case, you will see three parameters – the **USERNAME** and **PASSWORD**, which were part of the default add-on; and the new **WAS_AUDITOR** parameter, that you added as a part of your custom configuration. For this exercise, say that your company recommends that you set the auditor user name in the application server to the string **was**, and requires that an auditor user must always be created. You can configure and optionally lock those values into the pattern.

__i. In the **USERNAME** field, enter **was**. Since this is just a recommended configuration, it does not need to be locked into the pattern.

__ii. Since an auditor must be created, based on your corporate standard, you do not want pattern deployers to be able to change the WAS_AUDITOR field to false. It is set to true by default, so you just need to lock that value into the pattern.

__iii. Click the **open lock** icon (🔓) to the right of the **WAS_AUDITOR** field to lock it to **true**. After you click the lock icon, it will change to a **closed lock** (🔒), as shown below, indicating that the WAS_AUDITOR value is locked into the pattern and can't be changed at deployment time.

__iv. Click **OK** to close the dialog.



Parameters for Custom user add-on 99

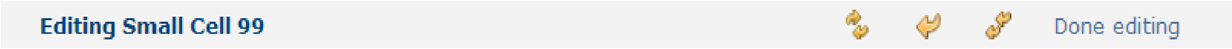
USERNAME: was 🔓

PASSWORD: 🔓

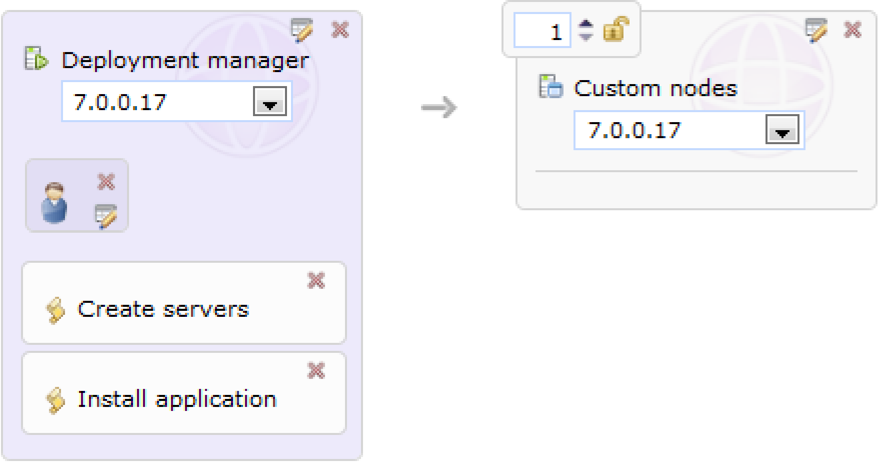
Verify password:

WAS_AUDITOR: true 🔒

__c. At this point, you are done defining your pattern and are ready to deploy it into the cloud. Click the **Done editing** link at the top of the pattern editor to close the Pattern Editor and save your changes.



__d. Back on the pattern details page, verify that your cell topology looks like this:



1.3 Deploy a virtual system pattern to the cloud

This section of the lab will guide you through the steps to deploy your virtual system pattern into the cloud. You have now created a new pattern in the pattern editor and customized it with script packages and add-ons. In the Pattern Editor, you provided several configuration data updates to your pattern parts and add-ons – for this deployment, the only values that are left to provide are the required passwords.

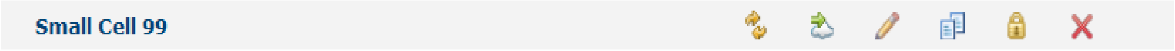
After deploying your virtual system, you will explore the virtual system and verify its configuration, then access the sample application that was deployed by your script packages.

1.3.1 Deploy your virtual system pattern

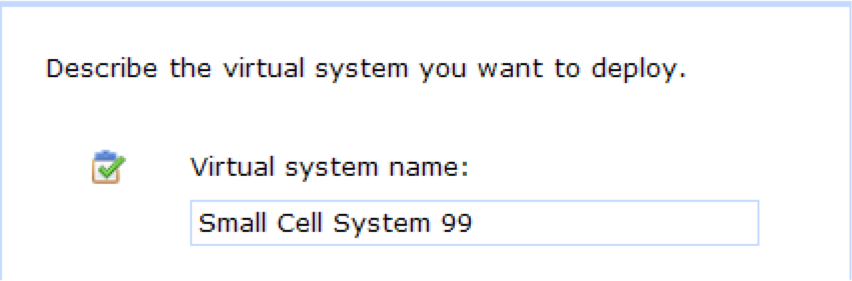
To deploy your pattern, you need to provide a name for the virtual system instances, identify the cloud group where the pattern will be deployed, and provide required configuration data for all of the pattern parts. In this case, the only configuration data left for your pattern parts will be passwords for all of the user accounts being created during deployment.

__1. Start the pattern deployment.

__a. If you are not already there, navigate to **Patterns > Virtual Systems** in the web console, then select **Small Cell <N>** from the list on the left. In the pattern toolbar in the right panel, select the **Deploy in the cloud…** button (☁).



__b. In the **Virtual system name** field, type **Small Cell System <N>**, where <N> is your user number.

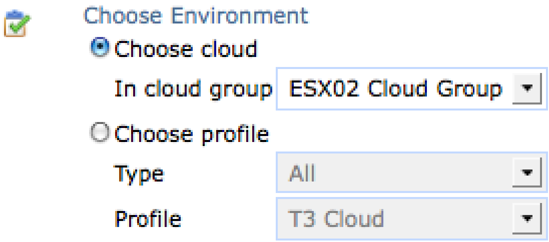


Describe the virtual system you want to deploy.

Virtual system name:

Small Cell System 99

__c. Select the text **Choose Environment** to expand the environment section. Verify that the **Choose cloud** radio button is selected. Choose the appropriate cloud group from the **In cloud group** menu (typically, the default cloud group). If you do not know which cloud group to choose, ask your instructor.



Choose Environment

● Choose cloud

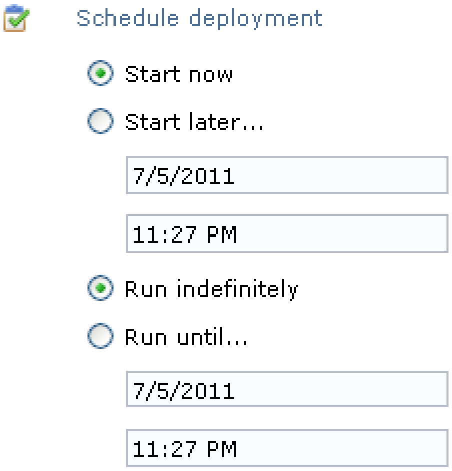
In cloud group: ESX02 Cloud Group

○ Choose profile

Type: All

Profile: T3 Cloud

__d. Keep the default values for the **Schedule deployment** section (which are to deploy the system immediately and keep it around indefinitely).



Schedule deployment

● Start now

○ Start later...

7/5/2011

11:27 PM

● Run indefinitely

○ Run until...

7/5/2011

11:27 PM

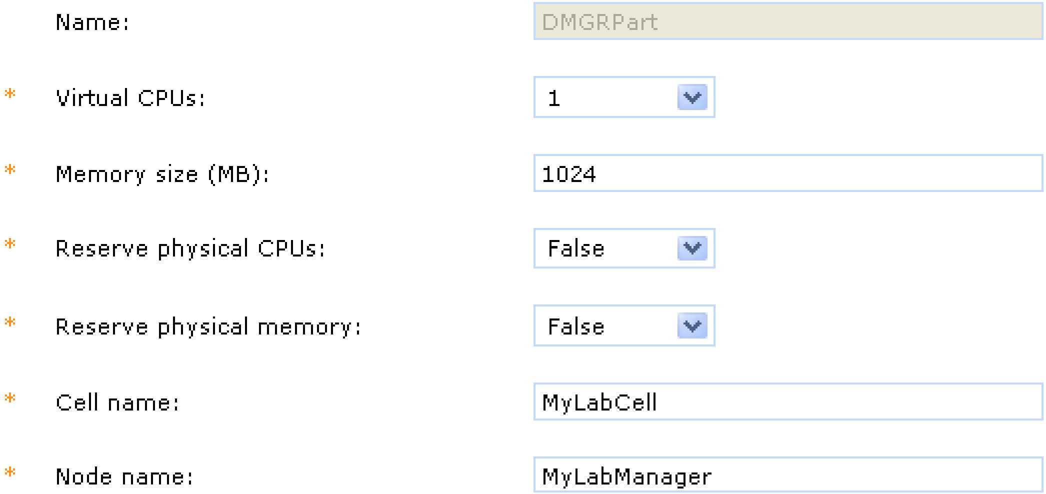
__e. Expand the **Configure virtual parts** section, then select the **Deployment manager** part. Provide these configuration values for the deployment manager:

__i. Leave the **Virtual CPUs** field set to **1**.

__ii. The **Memory size (MB)** field should already be set to **1024**, since you configured this value in the pattern part properties.

__iii. Leave the **Reserve physical CPUs** and **Reserve physical memory** fields set to **false**. In this case, by not reserving physical resources to map to your virtual machines, you will be able to make more efficient use of your cloud hardware.

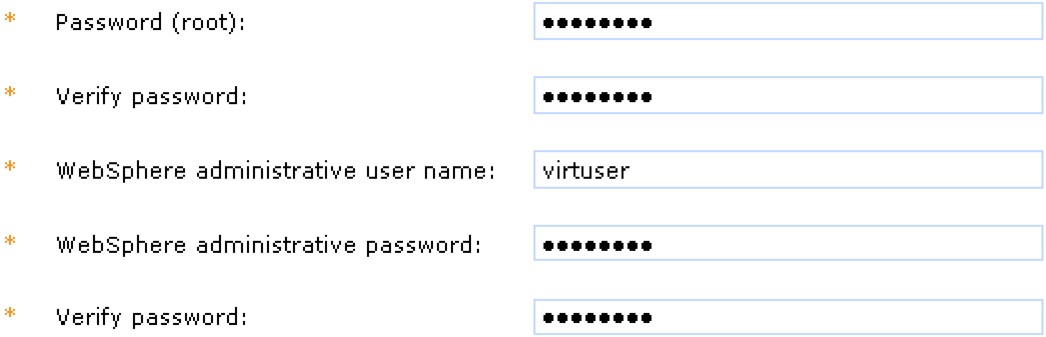
__iv. The **Cell name** should already be set to **MyLabCell**, and the **Node name** field should already be set to **MyLabManager**, just like you specified in the pattern part properties.



	Name:	DMGRPart
*	Virtual CPUs:	1
*	Memory size (MB):	1024
*	Reserve physical CPUs:	False
*	Reserve physical memory:	False
*	Cell name:	MyLabCell
*	Node name:	MyLabManager

__v. Scroll down in the part properties until you reach the **Password (root)** field. Set **Password (root)** and **Verify password** to **password**. This will set the operating system root user password.

__vi. Leave the **WebSphere administrative name** set to **virtuser**, and set the **WebSphere administrative password** and **Verify password** fields to **password**. This user account is built into the underlying image. This will create an operating system user named virtuser and also set the application server administrative ID to virtuser.

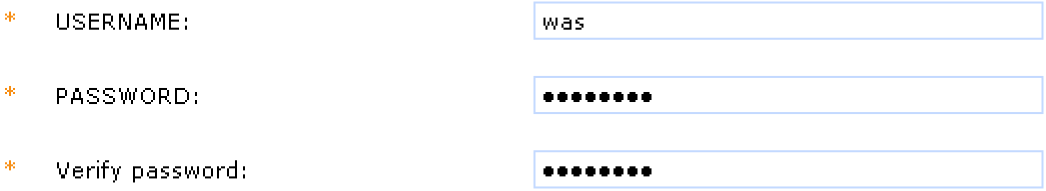


*	Password (root):	••••••••
*	Verify password:	••••••••
*	WebSphere administrative user name:	virtuser
*	WebSphere administrative password:	••••••••
*	Verify password:	••••••••

__vii. Scroll down to the bottom of the deployment manager part properties to locate the properties for your add-ons. Notice that the **USERNAME** field is set to **was**, just like you specified in the pattern configuration.

__viii. Set the **PASSWORD** and **Verify password** fields to **password** (this will be the password for the user **was**).

__ix. Notice that the **WAS_AUDITOR** field is not displayed for you to modify here. That is because you locked that parameter to true in the pattern editor, so it cannot be changed during deployment.



*	USERNAME:	was
*	PASSWORD:	●●●●●●●●
*	Verify password:	●●●●●●●●

__x. Select **OK** to close the deployment manager properties.

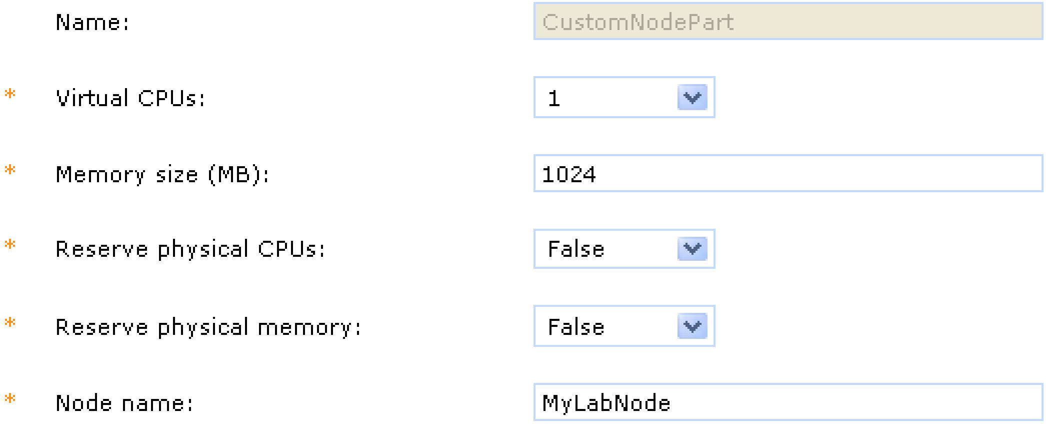
__f. Back in the virtual system deployment dialog, select **Custom nodes** and use these values to configure the properties for this pattern part:

__i. Leave the **Virtual CPUs** field set to **1**.

__ii. The **Memory size (MB)** field should already be set to **1024**, since you configured this value in the pattern part properties.

__iii. Leave the **Reserve physical CPUs** and **Reserve physical memory** fields set to **false**. In this case, by not reserving physical resources to map to your virtual machines, you will be able to make more efficient use of your cloud hardware.

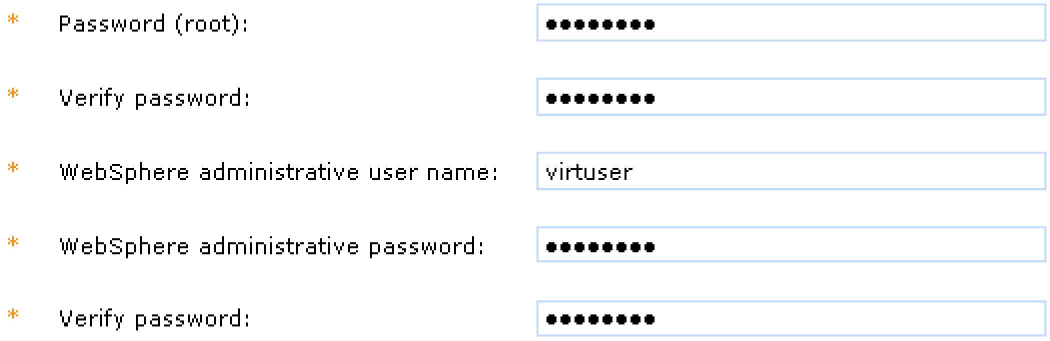
__iv. The **Node name** field should already be set to **MyLabNode**, just like you specified in the pattern part properties.



	Name:	CustomNodePart
*	Virtual CPUs:	1
*	Memory size (MB):	1024
*	Reserve physical CPUs:	False
*	Reserve physical memory:	False
*	Node name:	MyLabNode

__v. Scroll down in the part properties until you reach the **Password (root)** field. Set **Password (root)** and **Verify password** to **password**. This will set the operating system root user password.

__vi. Leave the **WebSphere administrative name** set to **virtuser**, and set the **WebSphere administrative password** and **Verify password** fields to **password**.



* Password (root): ●●●●●●●●

* Verify password: ●●●●●●●●

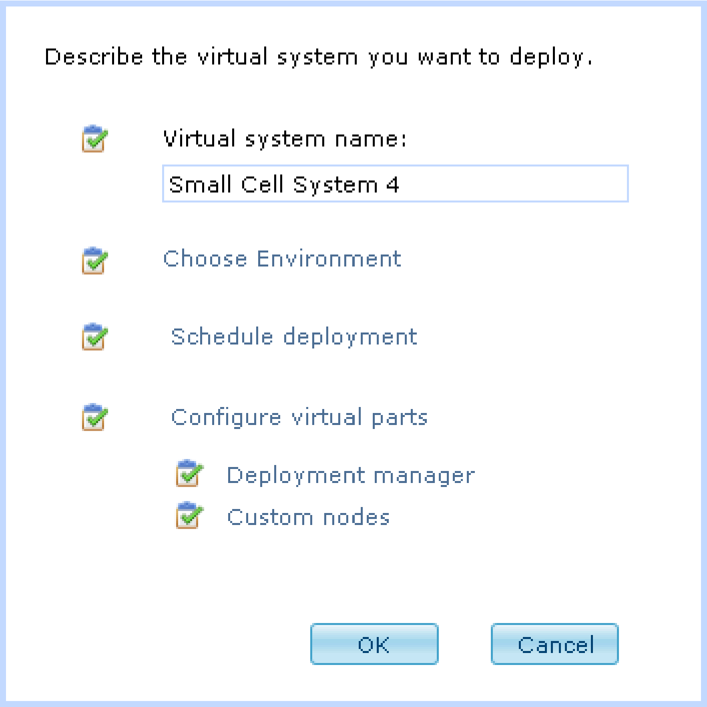
* WebSphere administrative user name: virtuser

* WebSphere administrative password: ●●●●●●●●

* Verify password: ●●●●●●●●

__vii. Select **OK** to close the custom node part properties.

__g. Back in the virtual system deployment dialog, all of the items in the panel show a green check status icon (✓). This means that all of the required deployment information has been provided, and you are ready to deploy.



Describe the virtual system you want to deploy.

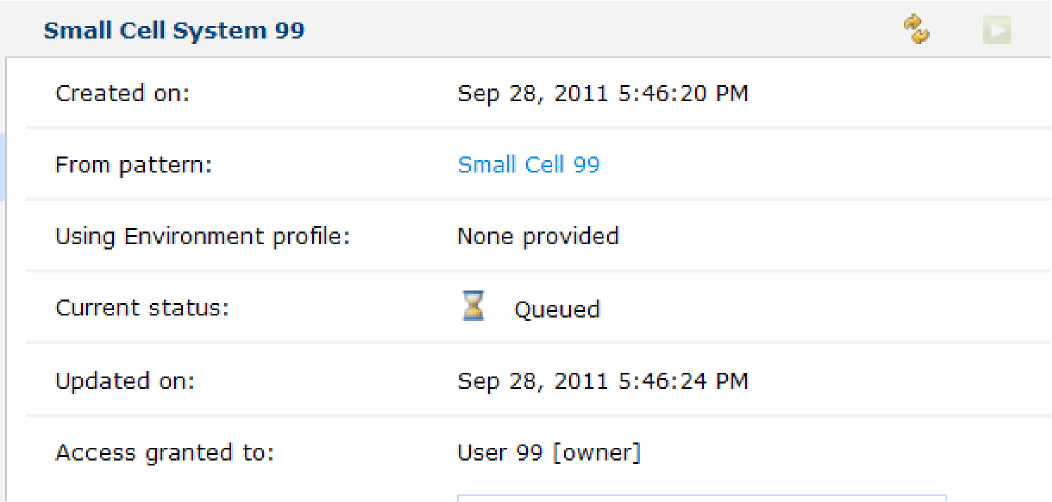
✓ Virtual system name: Small Cell System 4

__h. Click **OK** to begin the deployment process.

__2. Wait for your virtual system to finish deploying.

__a. You should be automatically redirected to the virtual system details page for your system. If not, you can also navigate to your virtual system by **selecting Instances > Virtual Systems**, then choosing **Small Cell System <N>** from the list at the left of the display.

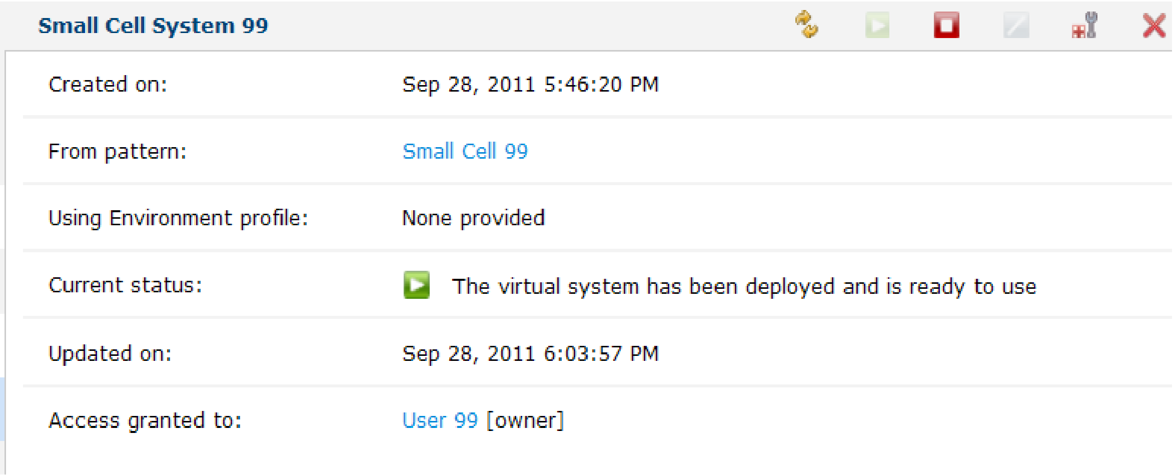
__b. Initially, your virtual system will have a status of **Queued**, indicating that the deployment has been added to the appliance queue to begin processing.



Small Cell System 99

Created on:	Sep 28, 2011 5:46:20 PM
From pattern:	Small Cell 99
Using Environment profile:	None provided
Current status:	⌛ Queued
Updated on:	Sep 28, 2011 5:46:24 PM
Access granted to:	User 99 [owner]

__c. Periodically click the **Refresh** button (🔄) to view updated status information for your virtual system. When deployment has finished, the current status will be updated to say that **The virtual system has been deployed and is ready to use**.



Small Cell System 99

Created on:	Sep 28, 2011 5:46:20 PM
From pattern:	Small Cell 99
Using Environment profile:	None provided
Current status:	▶ The virtual system has been deployed and is ready to use
Updated on:	Sep 28, 2011 6:03:57 PM
Access granted to:	User 99 [owner]

__d. Now that your virtual system is ready to use, you can proceed to the next section, in which you will verify the configuration of your virtual system.

1.3.2 Verify the virtual system configuration

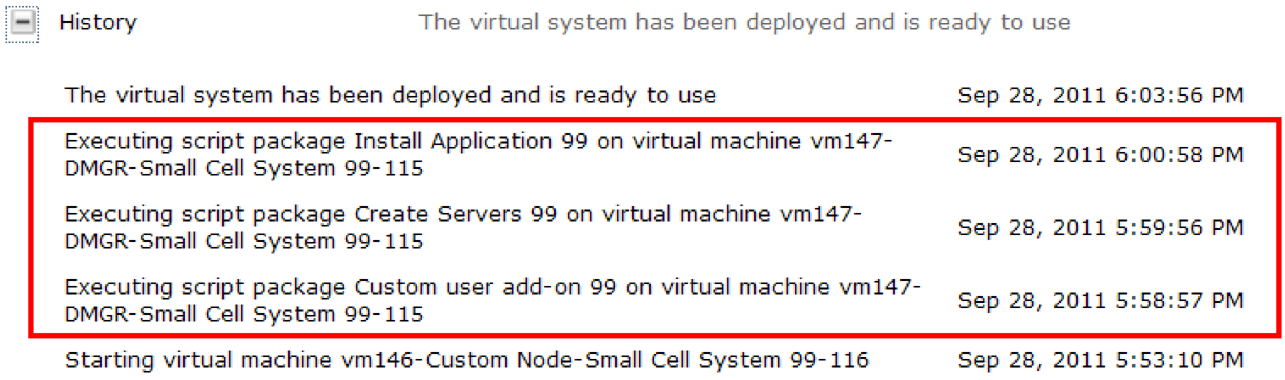
Now that your system is deployed, you can browse through the deployment results and the system itself to verify that your instance is deployed and configured correctly. In this section, you will browse through the IBM Workload Deployer console to view configuration status for your virtual system and access the virtual system directly, including through an SSH session, using the application server administrative console, and opening the deployed sample application.

__1. Explore the virtual system deployment results.

__a. Browse through the system history to see the script execution for the script packages and add-ons.

__i. From the virtual system page (if you are not there, go to **Instances > Virtual Systems**, and open up **Small Cell System <N>**), expand the **History** section.

__ii. In the history, you will see several messages about script package execution. Note that, in the history, events are recorded chronologically, with the first events at the bottom and the most recent events at the top. Add-on packages are always executed before user-supplied script packages, so you will notice that your add-on package (custom add user) is displayed **below** the script packages in your history. You should also see three messages about script package execution that reflect the order of scripts you defined – the create servers script comes first and then install application. This matches exactly the order you defined in your pattern.



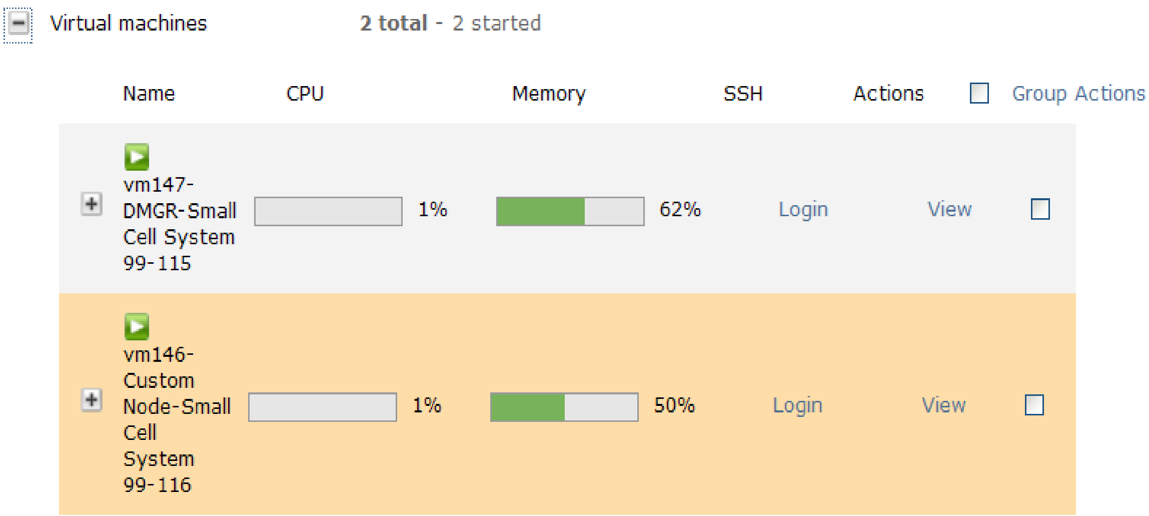
⊟ History The virtual system has been deployed and is ready to use

The virtual system has been deployed and is ready to use	Sep 28, 2011 6:03:56 PM
Executing script package Install Application 99 on virtual machine vm147-DMGR-Small Cell System 99-115	Sep 28, 2011 6:00:58 PM
Executing script package Create Servers 99 on virtual machine vm147-DMGR-Small Cell System 99-115	Sep 28, 2011 5:59:56 PM
Executing script package Custom user add-on 99 on virtual machine vm147-DMGR-Small Cell System 99-115	Sep 28, 2011 5:58:57 PM
Starting virtual machine vm146-Custom Node-Small Cell System 99-116	Sep 28, 2011 5:53:10 PM

__b. Verify that all of the scripts – for the add-ons and the script packages – ran successfully on the deployment manager.

__i. Scroll further down the virtual system display and click the twistie (⊞) to expand the **Virtual machines** section.

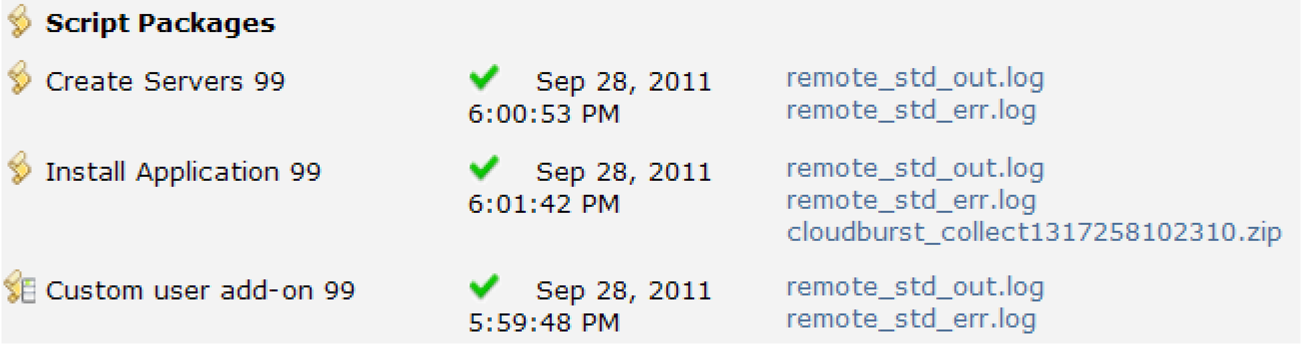
__ii. Next, expand the twistie (⊞) for the **DMGR** virtual machine (it should be the first virtual machine in the list).



⊟ Virtual machines **2 total** - 2 started

Name	CPU	Memory	SSH	Actions	□ Group Actions
⊞ vm147-DMGR-Small Cell System 99-115	1%	62%	Login	View	□
⊞ vm146-Custom Node-Small Cell System 99-116	1%	50%	Login	View	□

__iii. Scroll down in the virtual machine details until you locate the **Script Packages** section. You should see the standard application server configuration and mustgather scripts, one script for your add-on (**Custom user add-on <N>**), and the two script packages that were on the deployment manager – **Create Servers <N> and Install Application <N>** The add-on and script package execution results are listed together in alphabetical order. All of the scripts should show a green check mark as a status indicator (✓).



Script Packages

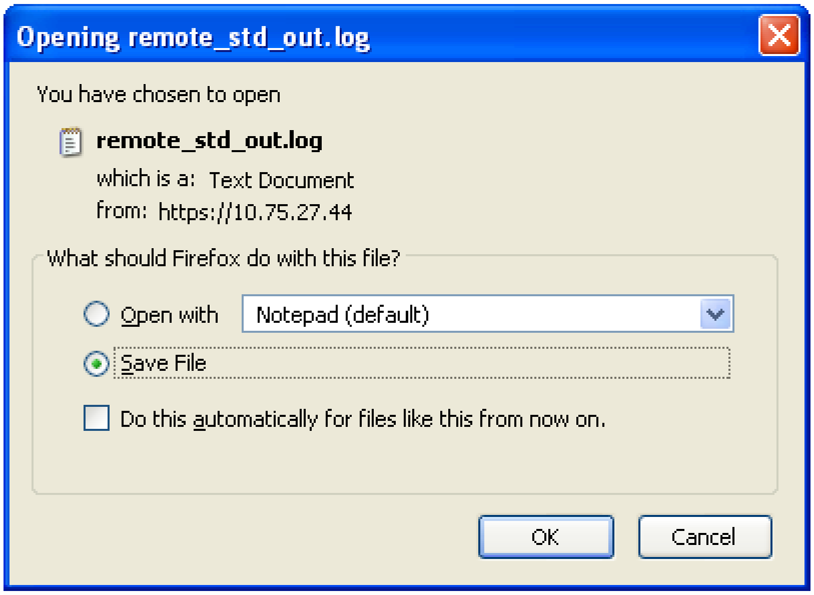
Create Servers 99	✓ Sep 28, 2011 6:00:53 PM	remote_std_out.log remote_std_err.log
Install Application 99	✓ Sep 28, 2011 6:01:42 PM	remote_std_out.log remote_std_err.log cloudburst_collect1317258102310.zip
Custom user add-on 99	✓ Sep 28, 2011 5:59:48 PM	remote_std_out.log remote_std_err.log

__iv. If any of your scripts are displayed with an error status (❗), you can gather logs and start analyzing the problem using the links to log files available in the web console, to the right of the script package results.

__c. Review the log file for the **Create Servers <N>** script package.

__i. To the right of each script package are links to the associated log files for the script. Locate **Create Servers <N>** in the **Script Packages** list, and select the link for the associated **remote_std_out.log** file.

__ii. When prompted, download the file to your local workstation.



Opening remote_std_out.log

You have chosen to open

remote_std_out.log

which is a: Text Document

from: https://10.75.27.44

What should Firefox do with this file?

○ Open with Notepad (default)

● Save File

□ Do this automatically for files like this from now on.

__iii. Open the **remote_std_out.log** file in a simple text editor, or list its content in a command prompt (for example, on Linux, navigate to the directory where you downloaded the file and run the command **cat remote_std_out.log**).

__iv. In the log file, you should see content similar to the following, indicating that this script package ran on the **deployment manager**, that the script found one node in the cell named **MyLabNode** (which is the name you specified in your pattern), and that an application server called **MyLabServer** was created on the node (this action happened during the Create Servers <N> script package).

```
WASX7209I: Connected to process "dmgr" on node MyLabManager_1
using SOAP connector;  The type of process is: DeploymentManager
parmlist:  [-serverType APPLICATION_SERVER -nodeName MyLabNode_3]
server name:
MyLabServer(cells/MyLabCell_1/nodes/MyLabNode_3/servers/MyLabServ
er|server.xml)
```

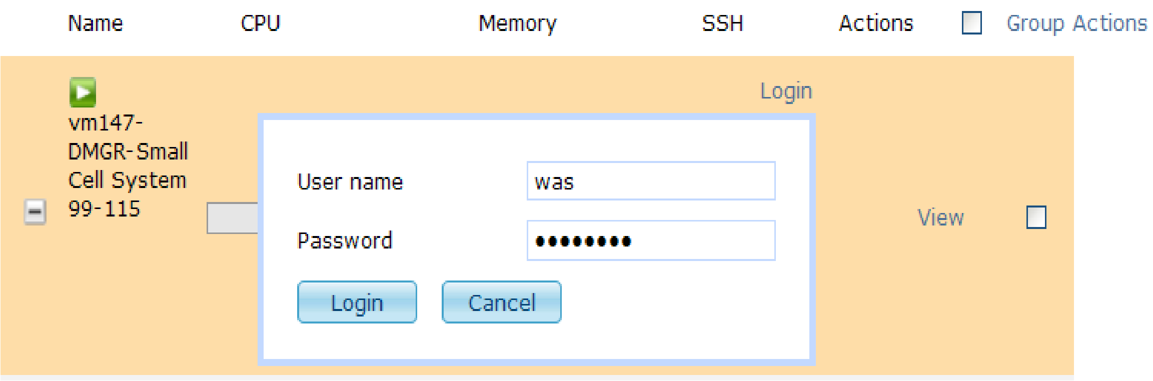
__2. Verify the configuration of the custom user add-on in the deployment manager virtual machine.

The user account you created with your custom add-on should exist in the operating system and the application server, and it should have auditor level authority in the application server.

__a. Open a new SSH session for your deployment manager VM, using the credentials created by the custom user add-on.

__i. Under the **SSH** column on the virtual machine details page for the deployment manager, click the **Login** link.

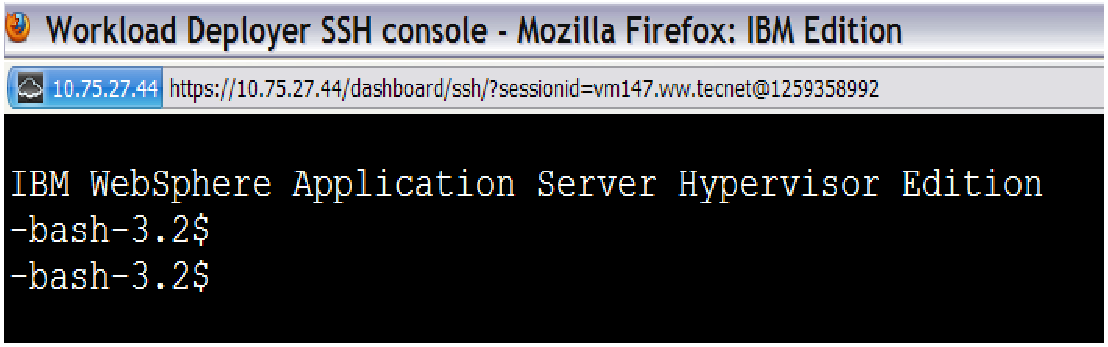
__ii. For the **User name**, use **was**. For the **Password**, use **password**. Click the **Login** button.



User name: was

Password: ••••••••

__iii. A new browser window should pop up that contains an SSH console for your virtual machine. If the new window does not open, check your browser's settings and make sure that pop-ups are allowed, then try to open the SSH session again.


Workload Deployer SSH console - Mozilla Firefox: IBM Edition
10.75.27.44

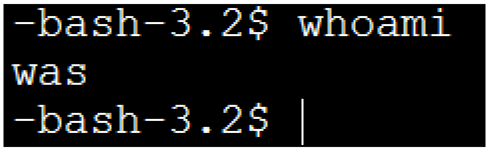
https://10.75.27.44/dashboard/ssh/?sessionid=vm147.ww.tecnet@1259358992

```
IBM WebSphere Application Server Hypervisor Edition
-bash-3.2$
-bash-3.2$
```

__iv. Once the SSH window opens, run the command **whoami** to verify that you are logged in with the **was** user account.

(1) **whoami**

__v. These are the results of the command:



```
-bash-3.2$ whoami
was
-bash-3.2$
```

__vi. You have verified that the operating system user account exists. Now you can close your SSH session.

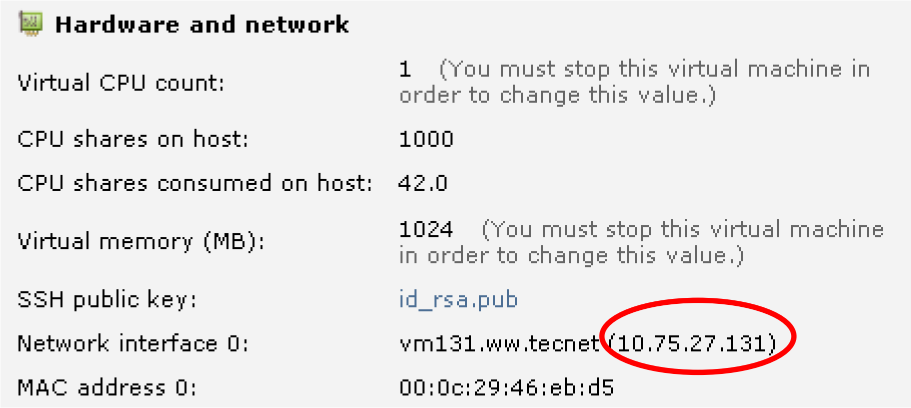
__3. Verify that the **was** user exists in the application server configuration.

You verified that the was user exists at the operating system level, but in your custom add-on, this user account should also have been defined at the application server level and configured with auditor privileges.

__a. Open the application server console and log in using the default administrative ID, **virtuser**.

__i. On the virtual machine details page for the deployment manager, scroll down to the **Hardware and network** section.

__ii. Locate the **Network interface 0** field and make a note of the IP address for the deployment manager, shown in parentheses after the host name. This exercise will refer to the IP address as **<DMGR_IP>**.



Hardware and network

Virtual CPU count:	1 (You must stop this virtual machine in order to change this value.)
CPU shares on host:	1000
CPU shares consumed on host:	42.0
Virtual memory (MB):	1024 (You must stop this virtual machine in order to change this value.)
SSH public key:	id_rsa.pub
Network interface 0:	vm131.ww.tecnet (10.75.27.131)
MAC address 0:	00:0c:29:46:eb:d5

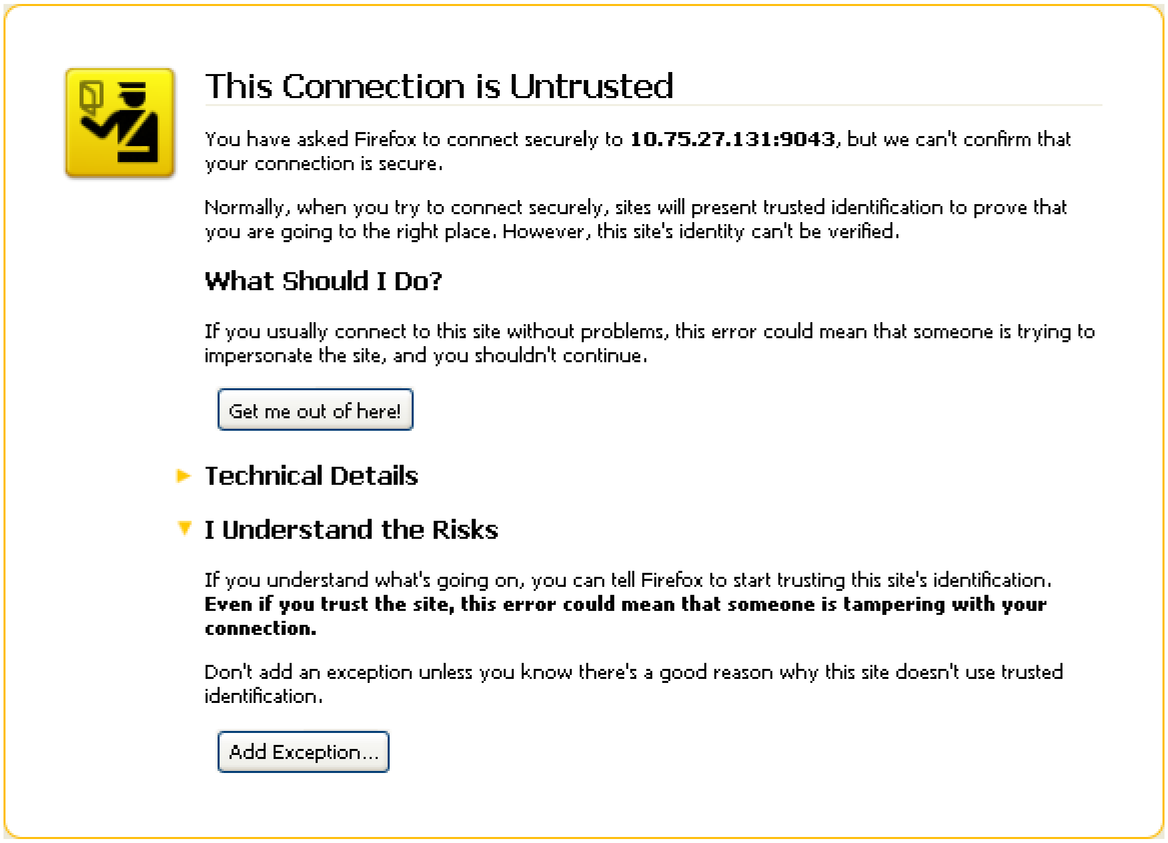
__iii. Open a new browser window and point it to the URL for the administrative console:

(1) **http://<DMGR_IP>:9060/ibm/console**

(2) Note that, if your workstation is configured to resolve hosts in the same DNS that your appliance is using, you would also be able to access the administrative console from the **Consoles** section of the virtual machine details page, using the **WebSphere link.**

__iv. When prompted, accept the security certificate for the connection.

(1) For example, in Firefox, expand the **I Understand the Risks** section, then click **Add Exception**.



This Connection is Untrusted

You have asked Firefox to connect securely to **10.75.27.131:9043**, but we can't confirm that your connection is secure.

Normally, when you try to connect securely, sites will present trusted identification to prove that you are going to the right place. However, this site's identity can't be verified.

What Should I Do?

If you usually connect to this site without problems, this error could mean that someone is trying to impersonate the site, and you shouldn't continue.

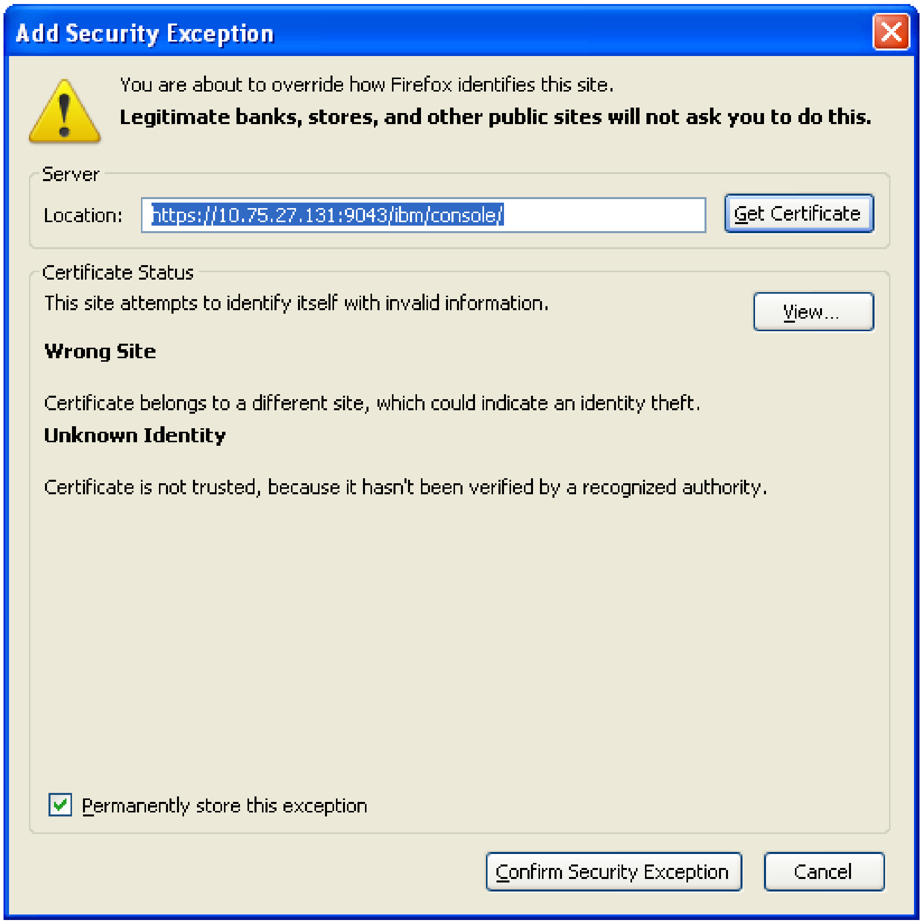
▸ Technical Details

▾ I Understand the Risks

If you understand what's going on, you can tell Firefox to start trusting this site's identification. **Even if you trust the site, this error could mean that someone is tampering with your connection.**

Don't add an exception unless you know there's a good reason why this site doesn't use trusted identification.

(2) In the pop-up window, choose to **Confirm Security Exception**.



Add Security Exception

You are about to override how Firefox identifies this site.

Legitimate banks, stores, and other public sites will not ask you to do this.

Server

Location: https://10.75.27.131:9043/ibm/console/

Certificate Status

This site attempts to identify itself with invalid information.

Wrong Site

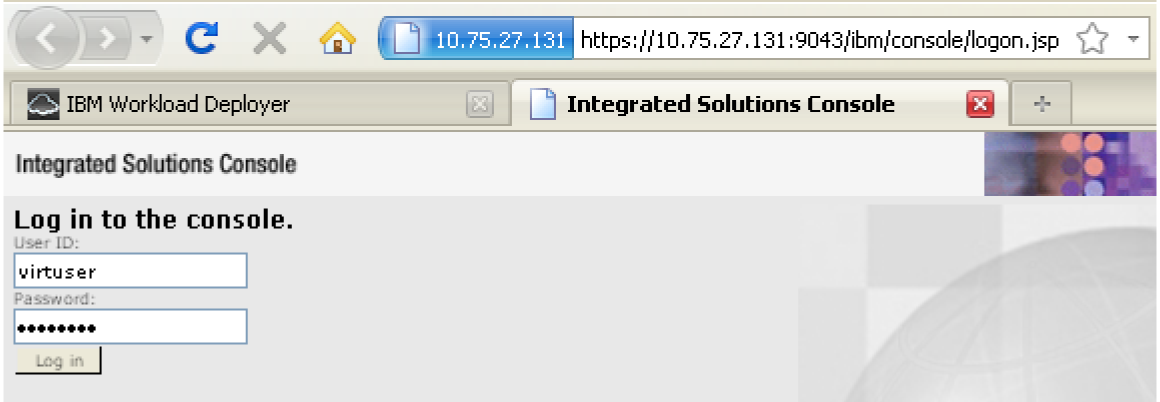
Certificate belongs to a different site, which could indicate an identity theft.

Unknown Identity

Certificate is not trusted, because it hasn't been verified by a recognized authority.

☑ Permanently store this exception

__b. Your pattern was automatically created with administrative security enabled, so you will be automatically re-directed to the secure administrative console. You will start by logging in with the administrative user for the application server. In the **User ID** field, use **virtuser**. In the **Password** field, use **password**. These are the credentials you provided for the WebSphere administrative user during pattern deployment. Click the **Log in** button.



10.75.27.131 https://10.75.27.131:9043/ibm/console/logon.jsp

Integrated Solutions Console

Log in to the console.

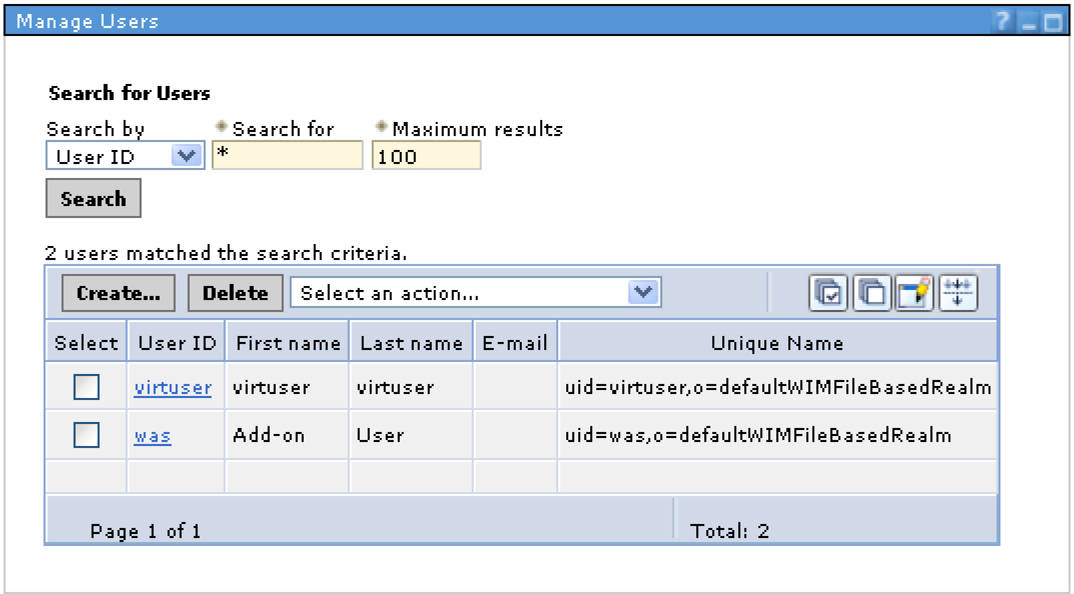
User ID: virtuser

Password: ••••••••

__c. Check that the user account **was** exists in the application server configuration.

__i. In the administrative console, navigate to **Users and Groups > Manage Users**.

__ii. On the **Manage Users** panel, click the **Search** button to list all of the users configured in the application server. You should see two users – **virtuser** and **was** – as shown below.



Search for Users

Search by: User ID

* Search for: *

* Maximum results: 100

2 users matched the search criteria.

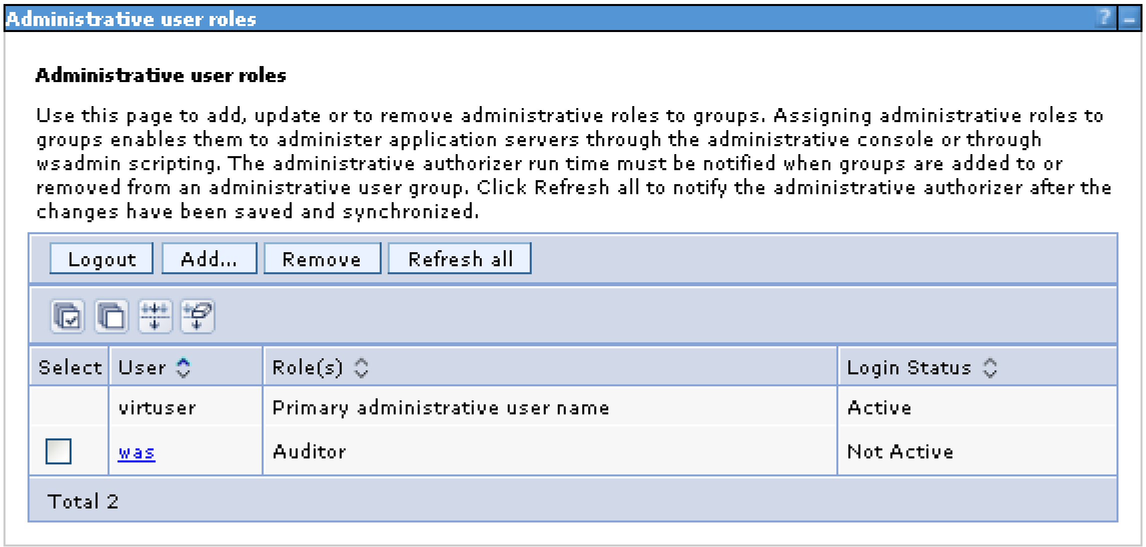
Select	User ID	First name	Last name	E-mail	Unique Name
□	virtuser	virtuser	virtuser		uid=virtuser,o=defaultWIMFileBasedRealm
□	was	Add-on	User		uid=was,o=defaultWIMFileBasedRealm

Page 1 of 1 Total: 2

__d. Verify that the user was has been given auditor authority in the application server.

__i. You should still be logged in to the administrative console as virtuser.

__ii. Navigate to **Users and Groups > Administrative user roles**. You will see two users listed – **virtuser**, the primary administrative account, and then **was**, which is listed as having the **Auditor** role.



Administrative user roles

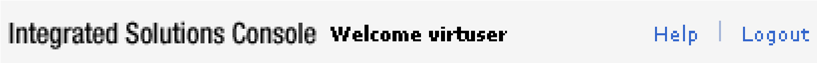
Use this page to add, update or to remove administrative roles to groups. Assigning administrative roles to groups enables them to administer application servers through the administrative console or through wsadmin scripting. The administrative authorizer run time must be notified when groups are added to or removed from an administrative user group. Click Refresh all to notify the administrative authorizer after the changes have been saved and synchronized.

Select	User	Role(s)	Login Status
	virtuser	Primary administrative user name	Active
□	was	Auditor	Not Active

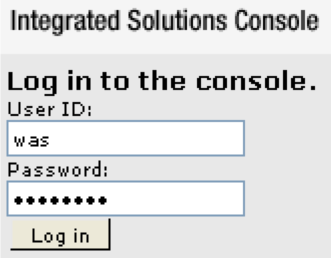
Total 2

__e. Finally, log in to the application server administrative console using the **was** account.

__i. Log out of the console as **virtuser** by clicking the **Logout** link near the top of the page.



__ii. Log in to the console again using **was** and **password**. Click **Log in**.



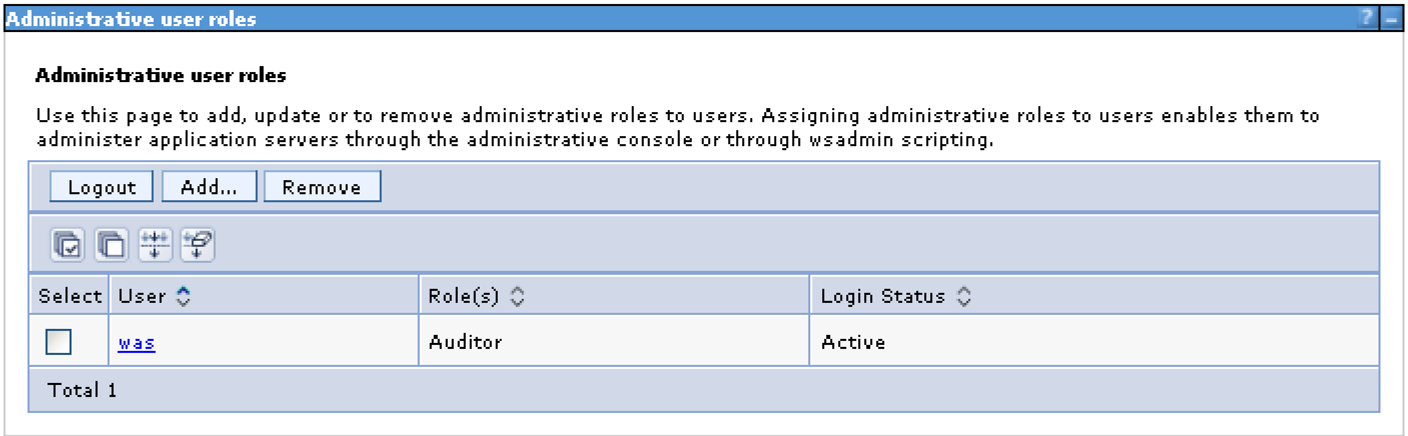
Log in to the console.

User ID: was

Password: •••••••••

__iii. Navigate back to the **Users and Groups > Administrative user roles** page.

__iv. Notice that you still see the **was** account listed here as an **Auditor**, but that the **virtuser** account is **not** shown in the table. That is because a user with the Auditor role does not have permission to see information about the administrative user account.



Administrative user roles

Use this page to add, update or to remove administrative roles to users. Assigning administrative roles to users enables them to administer application servers through the administrative console or through wsadmin scripting.

Select	User	Role(s)	Login Status
□	was	Auditor	Active
Total 1			

__v. At this point you have verified the custom user add-on configuration in your deployed virtual machine. All of your add-ons are deployed and working properly.

__f. Log out of the administrative console for the deployment manager, but keep the browser window open because you will use it in the next steps of the exercise.

__4. Access the sample application that was deployed with your script packages.

__a. Before you can access the application, you need to start the application server on which the application was installed. The application server was created using a script package, and the sample application was installed using a script package.

__b. Log in to the administrative console for the deployment manager using the **virtuser** ID.

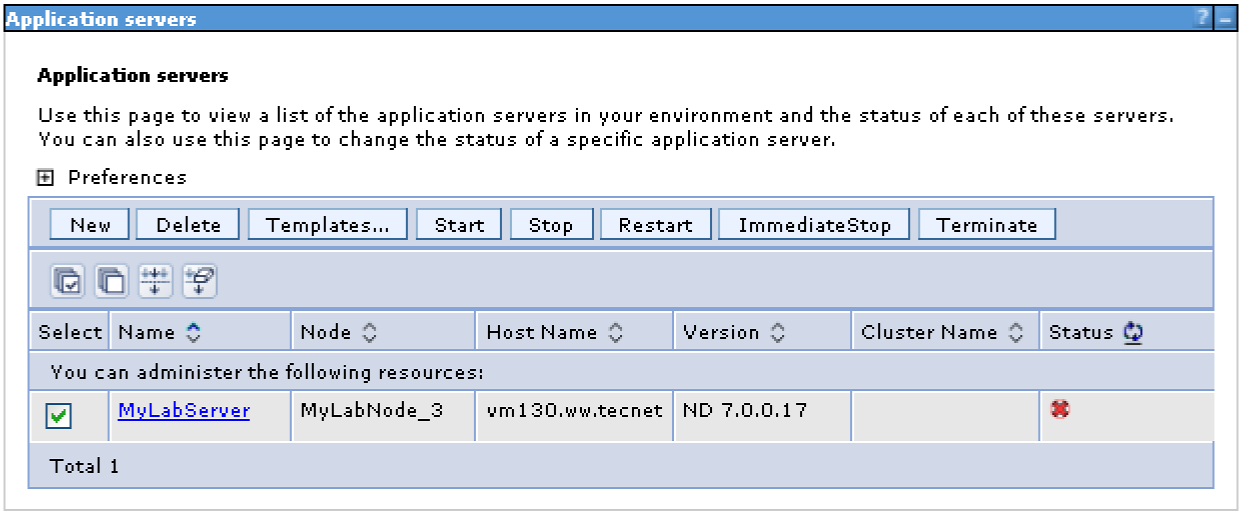
__i. If you do not still have the login page open for the administrative console, open a new browser window and point it to **http://<DMGR_IP>:9060/ibm/console**

__ii. Log in to the console using **virtuser** and **password**.

__c. Start the application server.

__i. Navigate to **Servers > Server Types > WebSphere application servers**.

__ii. Select the box next to **MyLabServer** and click the **Start** button.



Application servers

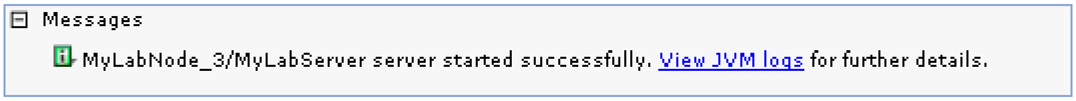
Application servers

Use this page to view a list of the application servers in your environment and the status of each of these servers. You can also use this page to change the status of a specific application server.

⊞ Preferences

Select	Name	Node	Host Name	Version	Cluster Name	Status
You can administer the following resources:						
☑	MyLabServer	MyLabNode_3	vm130.ww.tecnet	ND 7.0.0.17		✖
Total 1						

__d. After a moment, you will see a message at the top of the administrative console when the server has started.

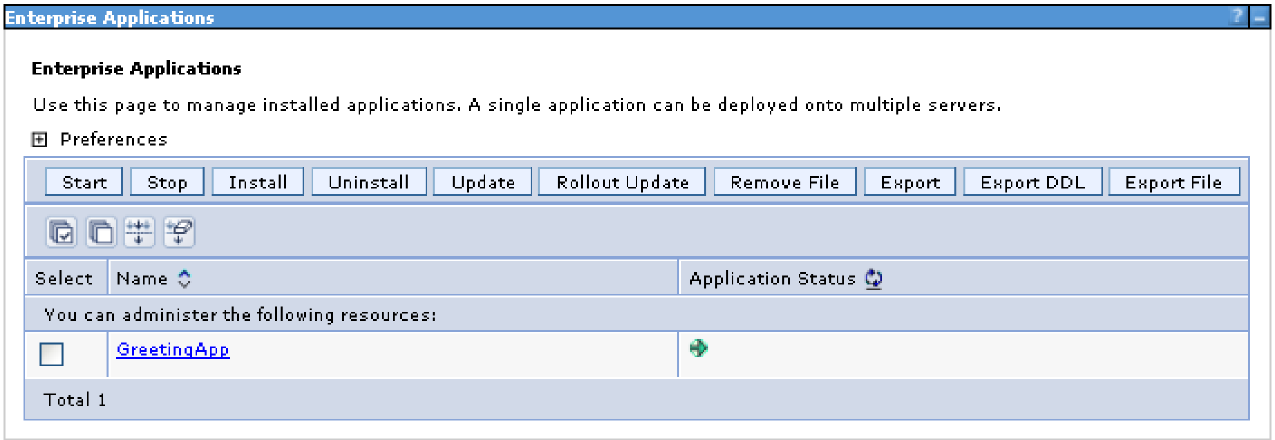


⊟ Messages

MyLabNode_3/MyLabServer server started successfully. View JVM logs for further details.

__e. In the left menu pane of the administrative console for the deployment manager, navigate to **Applications > Application Types > WebSphere enterprise applications**.

__f. Verify that **GreetingApp** is listed in **Started** state (➜). This is the sample application that was installed using your **Install Application <N>** script package.



Enterprise Applications

Use this page to manage installed applications. A single application can be deployed onto multiple servers.

Select	Name	Application Status
You can administer the following resources:		
□	GreetingApp	➜
Total 1		

__5. Now that the server is running and you have verified the sample application is started and available, you will access the sample application.

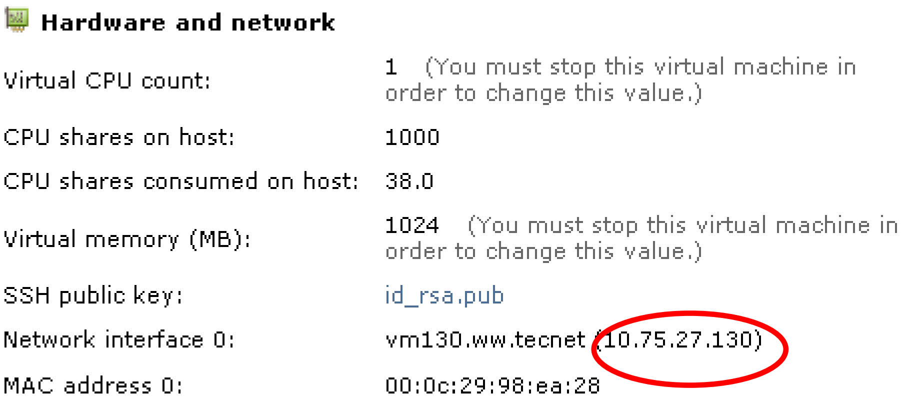
__a. Determine the host IP address for your custom node.

__i. Back in the IBM Workload Deployer web console, navigate to the instance page for your virtual system if you are not there already (**Instances > Virtual Systems > Small Cell System <N>**).

__ii. Expand the **Virtual machines** section, then expand the section for your **Custom node** virtual machine.

__iii. Scroll down through the virtual machine details and locate the **Hardware and network** section.

__iv. Locate the **Network interface 0** field and make a note of the **IP address** listed in parentheses after the host name. You will use this IP address (referred to as **<CUSTOM_NODE_IP>** below) to access the sample application.



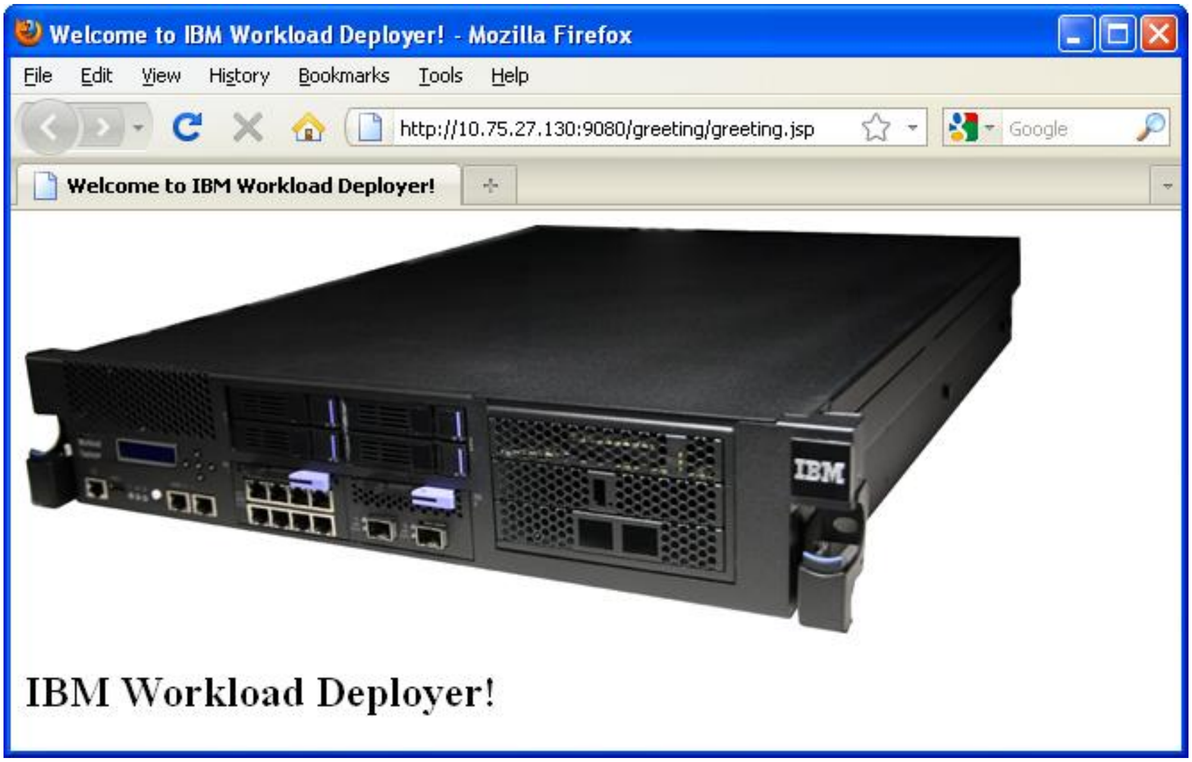
Hardware and network

Virtual CPU count:	1 (You must stop this virtual machine in order to change this value.)
CPU shares on host:	1000
CPU shares consumed on host:	38.0
Virtual memory (MB):	1024 (You must stop this virtual machine in order to change this value.)
SSH public key:	id_rsa.pub
Network interface 0:	vm130.ww.tecnet (10.75.27.130)
MAC address 0:	00:0c:29:98:ea:28

__b. Open a browser window and type the following URL, replacing <CUSTOM_NODE_IP> with the value you made a note of above.

__i. **http://<CUSTOM_NODE_IP>:9080/greeting/greeting.jsp**

__ii. You should see the following **IBM Workload Deployer!** message in your web browser.


Welcome to IBM Workload Deployer! - Mozilla Firefox
File
Edit
View
History
Bookmarks
Tools
Help
http://10.75.27.130:9080/greeting/greeting.jsp
Google
Welcome to IBM Workload Deployer!
IBM

IBM Workload Deployer!

__c. At this point, you have successfully created and deployed a virtual system pattern that uses script packages and add-ons to customize the topology, then verified the topology configuration and accessed the deployed sample application.

1.4 [Optional] Applying Maintenance

Virtual systems can require maintenance to the product code and the operating system. Rather than accessing each virtual machine, you can apply interim fixes and fix packs to your virtual system directly from IBM Workload Deployer. These fixes are downloaded from IBM and then are uploaded to the catalog. While a virtual system is started, you can apply the emergency fixes to all the virtual machines in the virtual system directly from the user interface.

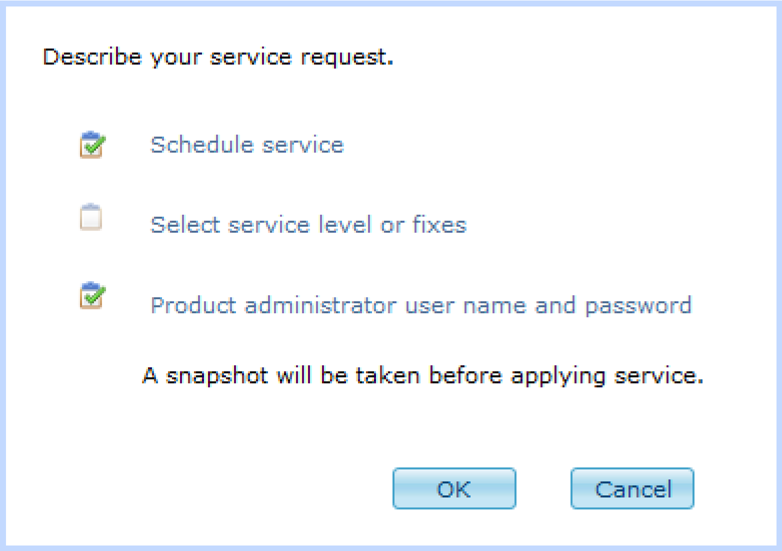
You will be applying a WebSphere Application Server interim fix (iFix) specific to version 7.0.0.17 that has been downloaded from IBM Fix Central and added to the catalog as an Emergency Fix.

__1. Use the Apply service mode to apply the emergency fix to your system.

__a. If you are not already on your virtual system instance page, navigate **to Instances > Virtual Systems > Small Cell System <N>**.

__b. Select the **Apply service** button ( ) from the instance toolbar

__c. From the service description panel, click **Select service level or fixes** to expand that section.



Describe your service request.

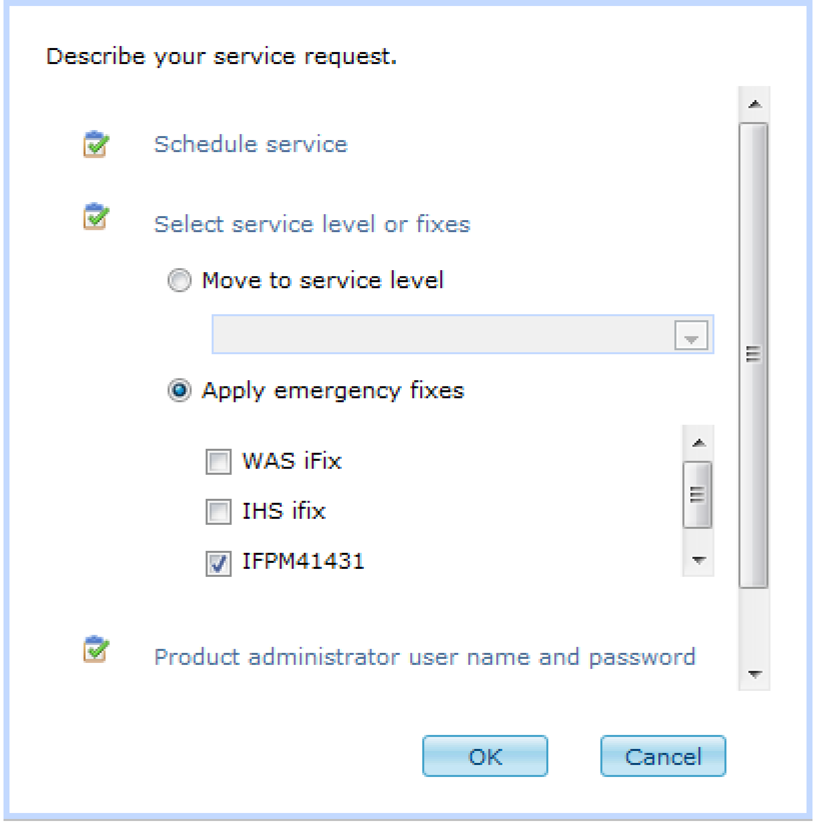
☑ Schedule service

□ Select service level or fixes

☑ Product administrator user name and password

A snapshot will be taken before applying service.

__d. Select the ratio button next to **Apply emergency fixes**, then check the box next to **IFPM41431**.



Describe your service request.

Schedule service

Select service level or fixes

○ Move to service level

● Apply emergency fixes

□ WAS iFix

□ IHS ifix

☑ IFPM41431

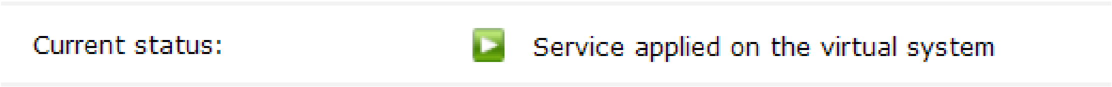
Product administrator user name and password

__e. In this example, you do not need to expand the **Product administrator user name and password** section. That is because, by default, the appliance will use its SSH keys to access the deployed virtual system – since you have not modified the WebSphere administrative user ID / password post-deployment, you do not need to provide any special credentials.

__f. Click **OK** to apply the fix.

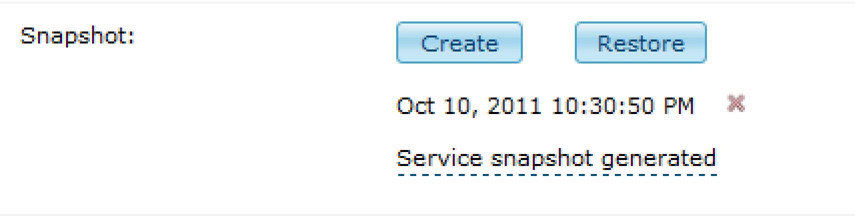
__2. Check that the fix was applied.

__a. Monitor the status of the virtual system. The virtual system will be stopped then restarted as maintenance is applied. Periodically refresh the virtual system status to view updated information. When the maintenance process is complete, the status of the virtual system will display Service applied on the virtual system.



Current status: ▶ Service applied on the virtual system

__b. Scroll down to the **Snapshot** section of the virtual system page. You will see that a snapshot was generated during the maintenance process. If you want to back out your fix, just restore from the previous snapshot (you do not need to do this now).

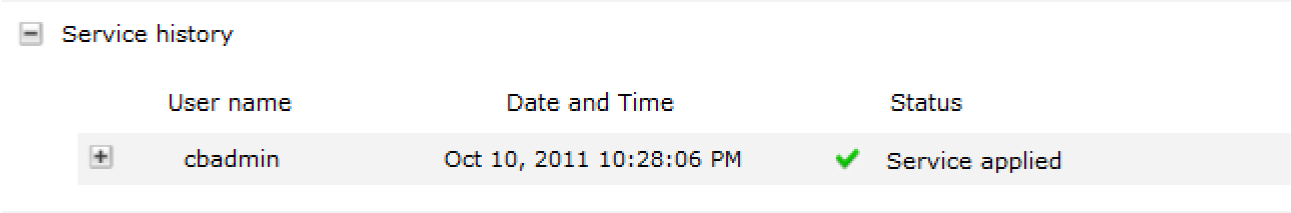


Snapshot:

Oct 10, 2011 10:30:50 PM ✗

Service snapshot generated

__c. Scroll down to the **Service history** section and expand it. You will see that it lists your fix IFPM41431 as having been applied to your virtual system.



⊟ Service history

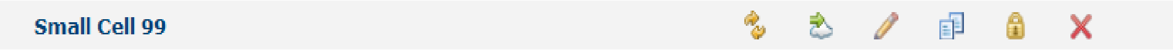
	User name	Date and Time	Status
⊞	cbadmin	Oct 10, 2011 10:28:06 PM	✓ Service applied

1.5 Clean-up system resources

__1. Now that you have finished validating your virtual system configuration, you should delete your environment from the cloud.

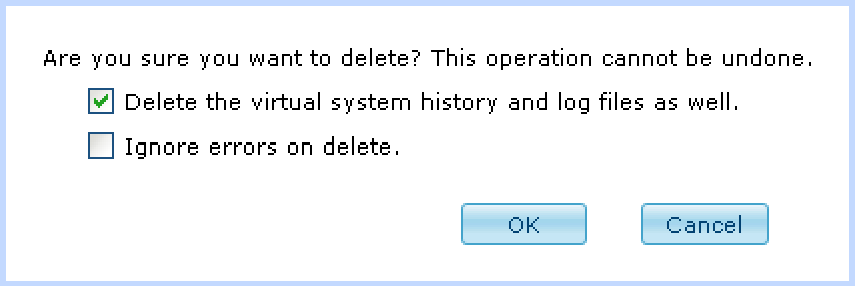
__a. In the IBM Workload Deployer web console, navigate back to your virtual system details page, if you are not there already (**Instances > Virtual Systems > Small Cell System <N>**).

__b. Click the **Delete** button (✗) in the virtual system toolbar.



Small Cell 99

__c. When prompted whether you are sure you want to delete the system, select **OK**.



Are you sure you want to delete? This operation cannot be undone.

☑ Delete the virtual system history and log files as well.

□ Ignore errors on delete.

__d. The appliance will cycle through and stop your virtual machines, then delete the virtual system. This process may take a few minutes. In a moment, refresh the browser display, and your virtual system, **Small Cell System <N>**, should no longer appear in the virtual system list.

__2. Delete your virtual system pattern.

__a. Navigate to **Patterns > Virtual Systems > Small Cell <N>**.

__b. Use the pattern toolbar to **Delete** (❌) the pattern.

__3. Delete your script packages.

__a. Navigate to **Catalog > Script packages**.

__b. Open your script package **Create Servers <N>,** and use the toolbar to **Delete** (❌) the script package.

__c. Open your script package **Install Application <N>,** and use the toolbar to **Delete** (❌) the script package.

1.6 Summary

In this exercise, you learned how to build, customize, and deploy virtual system patterns. You explored several types of customization for virtual systems, including:

- Explicitly defining your topology, including the number, type, and middleware version level of the components being used
- Using script packages to perform administrative tasks post-deployment – in this case, creating an application server instance inside your custom node and installing an enterprise application
- Performing advanced customizations on your virtual machines using add-ons
- Maintaining your deployed virtual system using an Emergency Fix from the IBM Workload Deployer catalog

After deploying your pattern into the cloud, you explored the virtual system configuration to verify that all of your script packages and add-ons executed correctly, and then you accessed your deployed sample application.

Lab 2 Building a dynamic virtual system with the Intelligent Management Pack

The objective of this lab is to demonstrate some features of the Intelligent Management Pack. The Intelligent Management Pack provides advanced capabilities for dynamic management of your virtual system patterns, including dynamic clustering and policy-based management for request routing and health management. This lab will guide you through the process of selecting and customizing a virtual system pattern that uses Intelligent Management Pack features, setting advanced options for that pattern, then deploying that pattern to the cloud. Once deployed, you will verify the configuration of your pattern – a dynamic cluster – and see the health management capabilities of the Intelligent Management Pack in action.

At the end of this lab you should be able to:

- Locate the default Intelligent Management Pack patterns
- Clone and customize a virtual system pattern
- Set advanced options for intelligent management
- Deploy an Intelligent Management Pack pattern
- Verify deployment

For each student, this lab exercise requires:

- IBM Workload Deployer, firmware version 3.0.0.2+; the exercise will refer to the IP address / host name of this appliance using **<IWD_IP>**
- The virtual image **WebSphere Application Server 7.0.0.17, ESX, RedHat Enterprise Linux 64-Bit 5** in the appliance catalog; cloned and with the Intelligent Management Pack enabled – the final image should be cached on the hypervisor prior to any students running the lab
- Virtual system patterns **Advanced cluster and Advanced cluster (development)** available on the appliance
- At least 6 GB RAM and up to 40 GB storage in the hypervisor resource pool
- At least 4 IPs be available in the IP resource pool
- A user ID with access to a cloud group for deployment, and at least the following permissions: deploy patterns into the cloud, create patterns
    - Your appliance credentials will be of the form user<N>, where <N> is your user number – for example, if you are using ID user04, then your user number, <N>, is 4. Make a note of your user number because you will use it throughout the exercise.

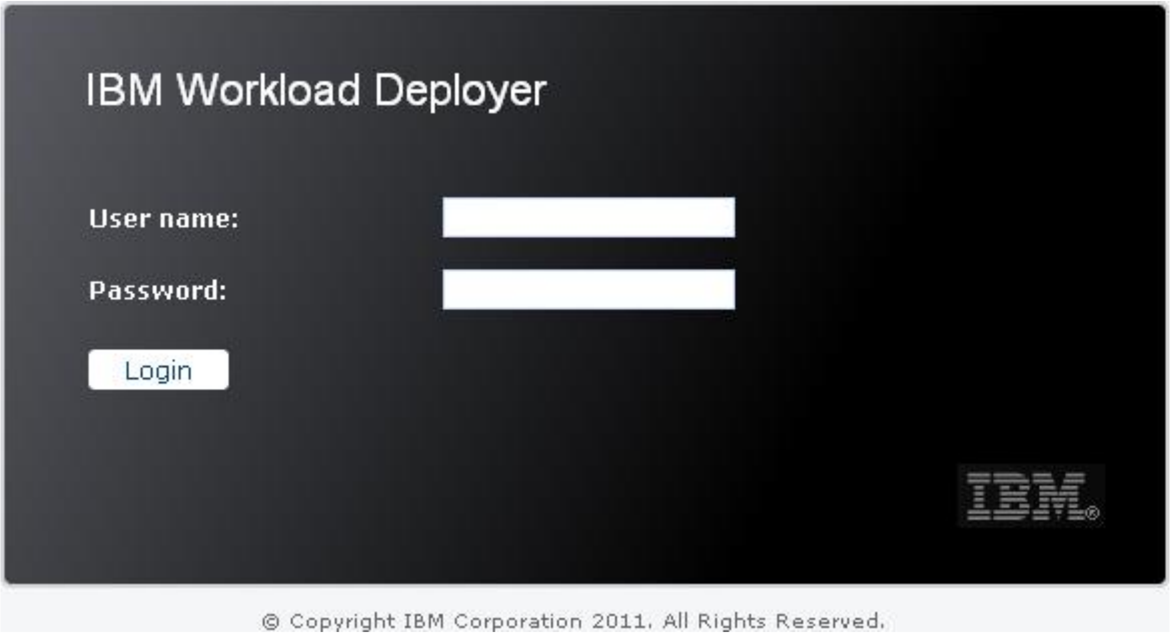
2.1 Create an enhanced virtual system pattern

This section of the lab will guide you through the process of creating your own custom pattern that uses some advanced capabilities of the Intelligent Management Pack.

2.1.1 Locate the pre-loaded Intelligent Management Pack patterns

__1. Log in to IBM Workload Deployer.

__a. Open a new window in your browser and point it to the IBM Workload Deployer at **https://<IWD_IP>/login/**

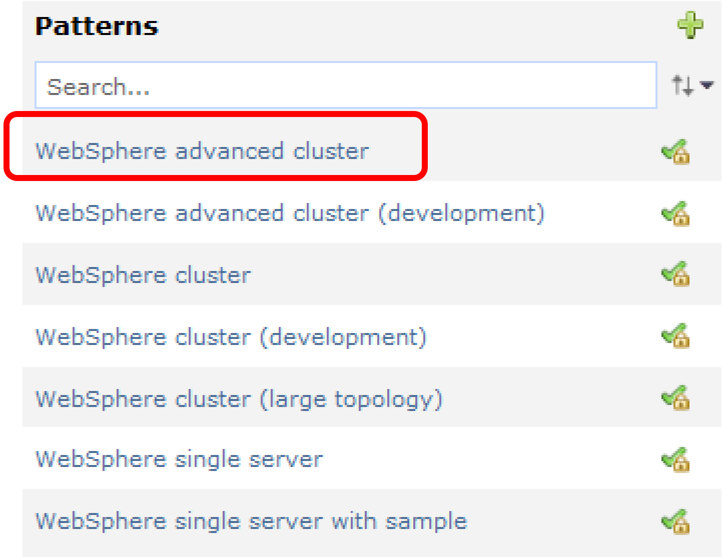


__b. Log in with the user-level credentials on your lab information sheet. Your **User name** will be of the form **user<N>**, where **<N>** is your user number. For example, if your ID is user04, then your <N> value is 4. Make a note of this number because you will use it in all of the names of your configuration objects on the appliance. Provide your **Password** (also available on your lab information sheet), then click **Login**.

__2. Explore the pre-loaded Intelligent Management Pack patterns.

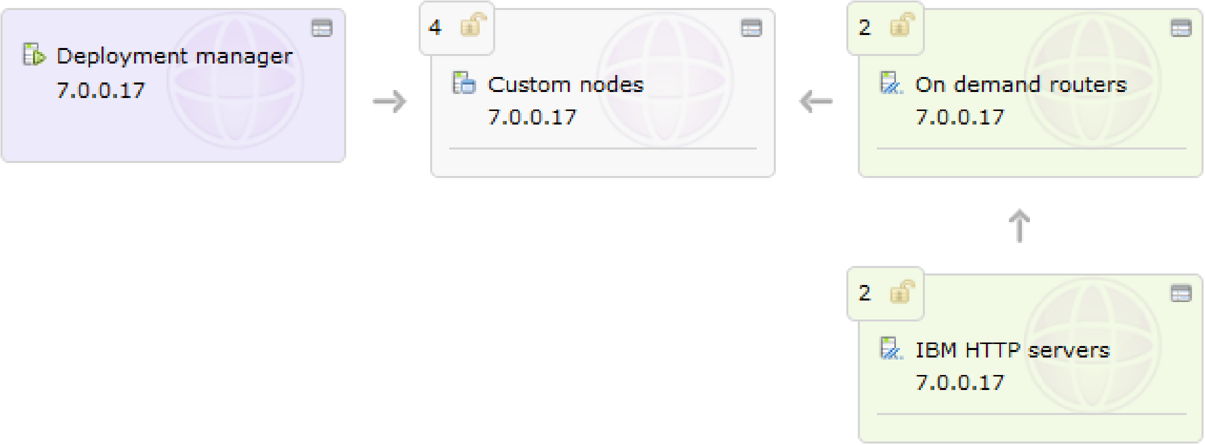
__a. In the console, navigate to the **Patterns > Virtual Systems** section, located on the ribbon at the top of the page.

__b. There are two advanced cluster patterns available; these patterns come pre-loaded on IBM Workload Deployer. The advanced cluster patterns make use of Intelligent Management Pack features, including the on demand router part. Start by selecting the **WebSphere advanced cluster** pattern from the list of available patterns.

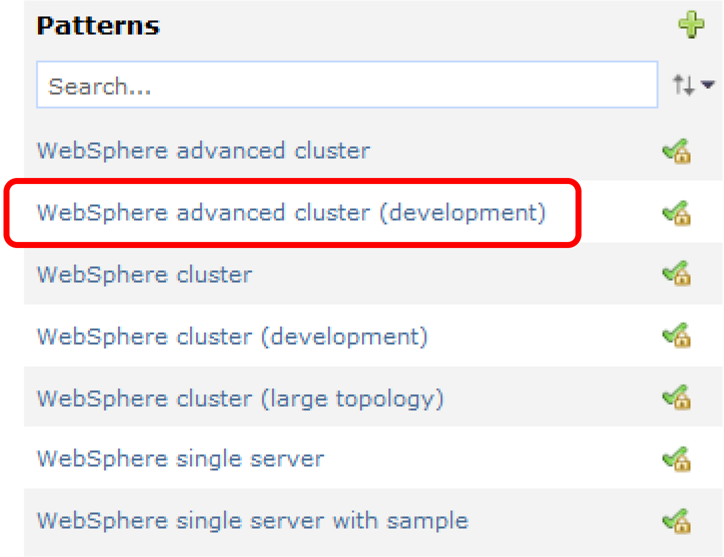


WebSphere advanced cluster
WebSphere advanced cluster (development)
WebSphere cluster
WebSphere cluster (development)
WebSphere cluster (large topology)
WebSphere single server
WebSphere single server with sample

__c. The **WebSphere advanced cluster** pattern is designed to be used in your production environment. The web tier (represented by the IBM HTTP servers parts) forwards incoming requests to the ODR tier (represented by the on demand routers parts), which then controls traffic flow back to the application server tier (represented by the custom node and deployment manager parts). This single pattern contains 9 virtual machines, so it will use 9 IP addresses. Each virtual machine has a default virtual memory allocation of 2GB, for a total of 18GB of memory for the entire deployed pattern. The memory configuration can be modified to a minimum of 9GB for the entire pattern (1GB memory per part). This screen capture shows the topology that is represented by the **WebSphere advanced cluster** pattern:



__d. Next, navigate to the **WebSphere advanced cluster (development)** pattern.



Patterns

Search...

WebSphere advanced cluster

WebSphere advanced cluster (development)

WebSphere cluster

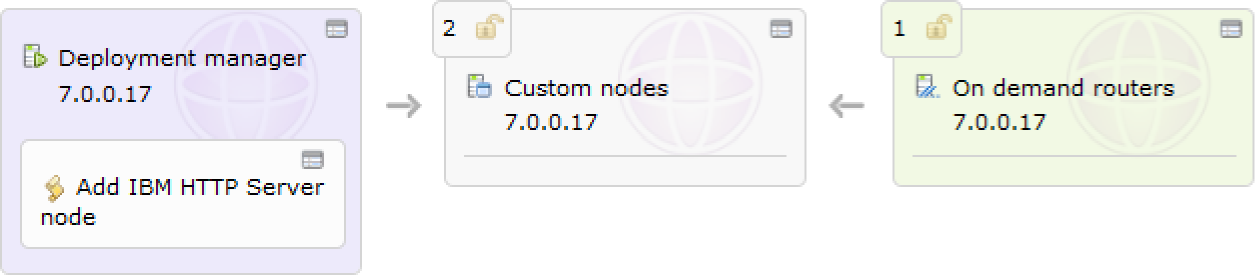
WebSphere cluster (development)

WebSphere cluster (large topology)

WebSphere single server

WebSphere single server with sample

__e. This pattern is geared toward your development and test environments (not production environments, like the previous pattern you looked at). One key difference between this pattern and the previous advanced pattern is that here, you do not have any IBM HTTP server parts (that means that there is no separate web tier, but an optional web server instance configured on the deployment manager part using a script package). In addition, the development pattern contains fewer nodes, only 2 custom nodes and 1 on demand router, giving it a much smaller footprint. This screen capture shows the topology of the **WebSphere advanced cluster (development)** pattern:



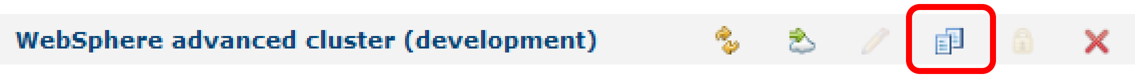
__f. The Intelligent Management Pack provides several other advanced features, such as dynamic clustering, health policies, and overload protection. You will learn how to enable these features in the next section of the exercise.

2.1.2 Clone and customize an Intelligent Management Pack pattern

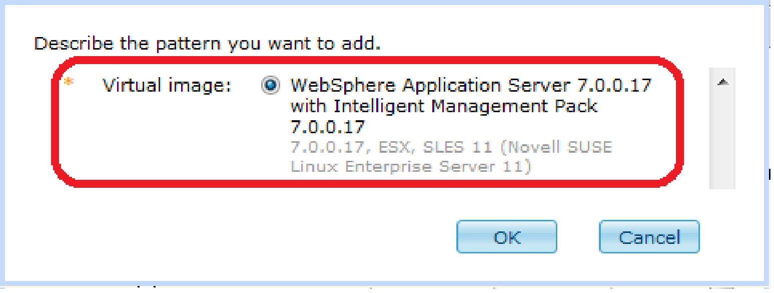
__1. Clone an existing Intelligent Management Pack pattern.

__a. If you are not already there, open the **WebSphere advanced cluster (development)** pattern by navigating to **Patterns > Virtual Systems**, then selecting the **WebSphere advanced cluster (development)** pattern to open the pattern details in the right panel.

__b. In the upper right of the panel, there is toolbar of action buttons for this pattern. Click the **Clone** button (⧉) to create a copy of this pattern.



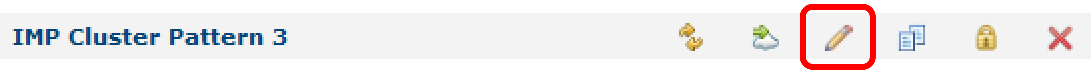
__c. In the **Name** field for your new pattern, type **IMP Cluster Pattern <N>,** where <N> is your user ID number. For example, if your user account is user03, then your <N> value is 3, and you would use the name IMP Cluster Pattern 3, as shown. Then, from the **Virtual image** list, select the radio button next to **WebSphere Application Server 7.0.0.17 with Intelligent Management Pack 7.0.0.0**, with the sub-heading **7.0.0.17, ESX, RedHat Enterprise Linux 64-bit 5 (RedHat Enterprise Linux 5)**. You might need to scroll through the virtual image list to find the exact image you are looking for. Click the **OK** button.


Describe the pattern you want to add.

* Virtual image: ● WebSphere Application Server 7.0.0.17 with Intelligent Management Pack 7.0.0.17
7.0.0.17, ESX, SLES 11 (Novell SUSE Linux Enterprise Server 11)

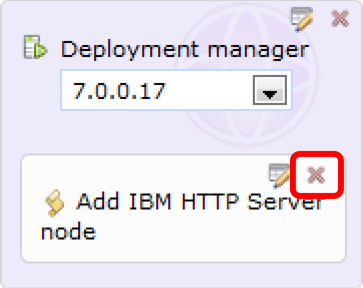
__2. Update the configuration for your pattern parts.

__a. After cloning the pattern, your new pattern **IMP Cluster Pattern <N>** will automatically be selected in the IBM Workload Deployer console. From the action toolbar for your pattern, click the **Edit** button (✎) to open the Pattern Editor.

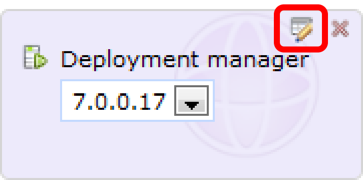


__b. In this scenario, you will be using a simple dynamic cluster with one deployment manager, one custom node, and one on demand router. No separate IBM HTTP server instance is necessary.

__c. Start by removing the IBM HTTP server script package from the **deployment manager** part. In the upper right of the **Add IBM HTTP Server node** script package box, click the **Delete** icon (✖) to remove the script package.



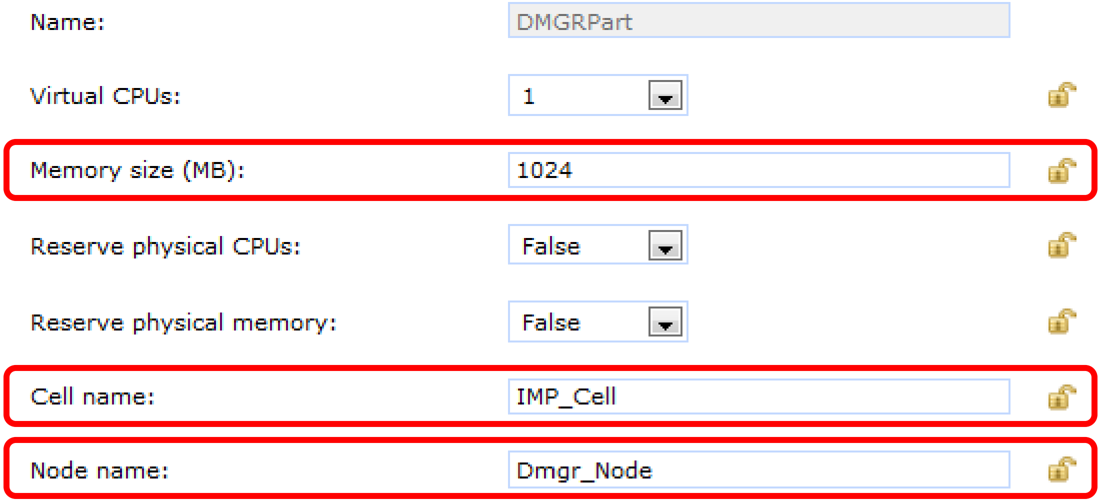
__d. Now, you need to update the properties for the parts in your pattern. Select the **Properties** button (✎) in the upper right corner of the **Deployment manager** part.



Deployment manager

7.0.0.17

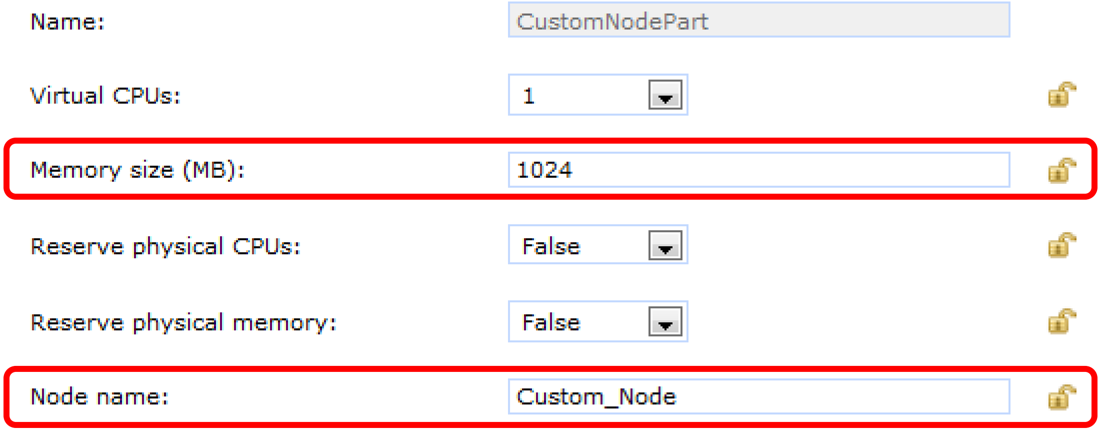
__e. In the **Properties for part Deployment manager** panel, update the field **Memory size (MB)** to **1024**. This change will help save resources in the cloud for the purposes of this class; in your own deployment environment scaling down the memory would not be necessary. Also update the **Cell name** field to **IMP_Cell** and the **Node name** filed to **Dmgr_Node**. ***Make sure that you use the exact case and spelling for the cell and node names!***



Name:	DMGRPart
Virtual CPUs:	1
Memory size (MB):	1024
Reserve physical CPUs:	False
Reserve physical memory:	False
Cell name:	IMP_Cell
Node name:	Dmgr_Node

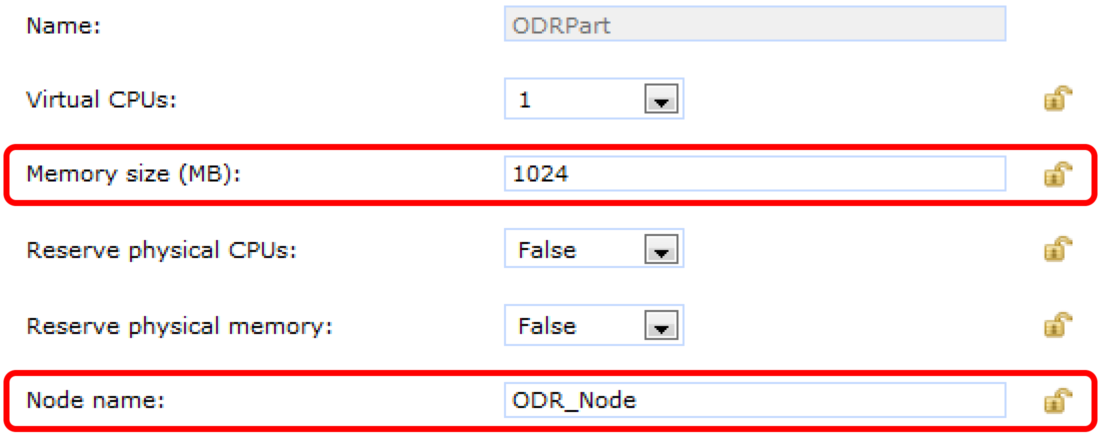
__f. Click **OK** in the **Properties** panel to save your changes to the deployment manager.

__g. Select the **Properties** button (✎) to open the **Properties panel for the Custom node** part. Here, change the **Memory size (MB)** field to **1024**, and update the **Node name** field to **Custom_Node**. Then click **OK** to save your changes.



Name:	CustomNodePart
Virtual CPUs:	1
Memory size (MB):	1024
Reserve physical CPUs:	False
Reserve physical memory:	False
Node name:	Custom_Node

__h. Next, open the **Properties** panel for the **On demand router** part. Again, change the **Memory size (MB)** field to **1024**. Set the **Node name** for this part to **ODR_Node**. Click **OK** to save your changes.

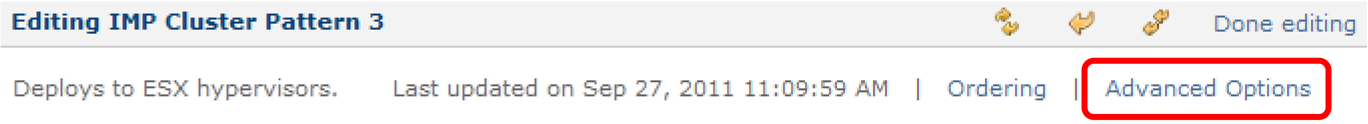


Name:	ODRPart
Virtual CPUs:	1
Memory size (MB):	1024
Reserve physical CPUs:	False
Reserve physical memory:	False
Node name:	ODR_Node

__i. Now you have finished configuring the properties for your pattern parts. Next, you will enable several Intelligent Management Pack features using the advanced options for your pattern.

__3. Configure advanced options for your pattern.

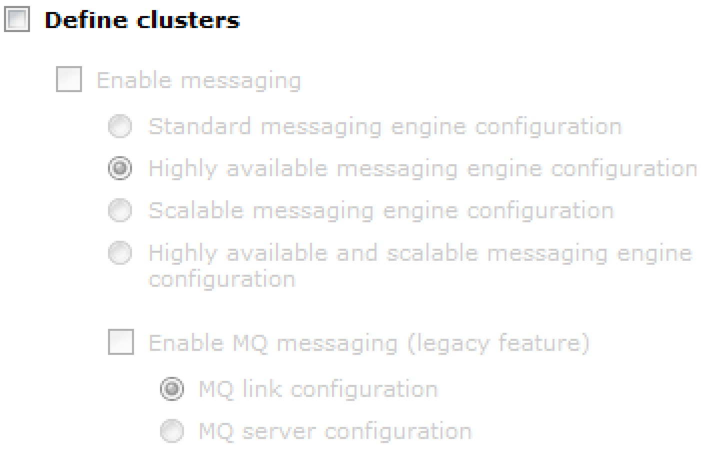
__a. Special options for enabling Intelligent Management Pack capabilities – such as dynamic clustering and health policies – are available in the pattern editor in the **Advanced Options** section. In the upper right of the pattern editor, click the link for Advanced Options to open a new panel for configuring additional settings for your pattern.



Editing IMP Cluster Pattern 3

Deploys to ESX hypervisors. Last updated on Sep 27, 2011 11:09:59 AM | Ordering | Advanced Options

__b. For your pattern, some static clustering options will be selected by default. In your pattern, you will be using dynamic clustering, which is a feature of the Intelligent Management Pack – so you need to turn the static clustering off. In the **Define clusters** section, deselect the box next to **Enable messaging**. Then, deselect the box next to **Define clusters** (in that order).



□ **Define clusters**

□ Enable messaging

○ Standard messaging engine configuration

● Highly available messaging engine configuration

○ Scalable messaging engine configuration

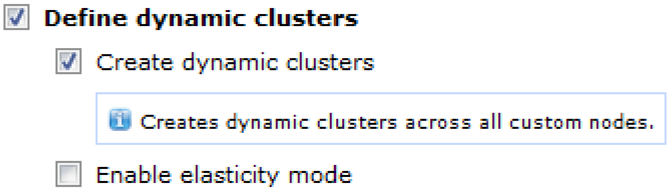
○ Highly available and scalable messaging engine configuration

□ Enable MQ messaging (legacy feature)

● MQ link configuration

○ MQ server configuration

__c. Now that you have disabled static clustering, you will turn on dynamic clustering and enable some health management options. Scroll down in the Advanced Options panel until you find the **Define dynamic clusters** section. Select the checkbox next **to Define dynamic clusters**, then select the **Create dynamic clusters** box. By selecting these options, the appliance will add a special script package called **Dynamic cluster configuration** to your on demand router part.



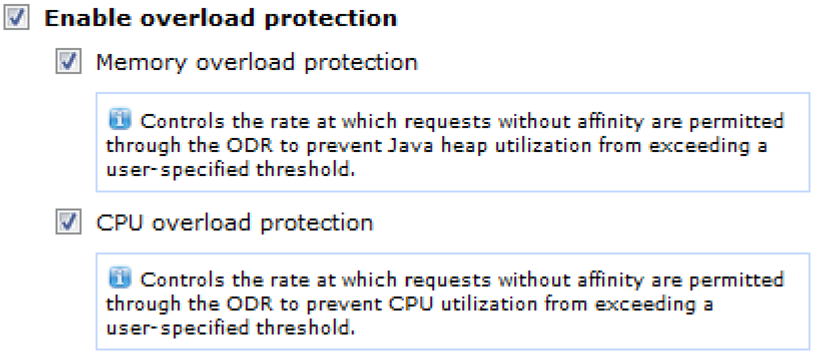
☑ **Define dynamic clusters**

☑ Create dynamic clusters

Creates dynamic clusters across all custom nodes.

□ Enable elasticity mode

__d. Next, select the high-level box for **Enable overload protection** and then select both sub-options – **Memory overload protection** and **CPU overload protection**. For these options, two script packages will be added to your on demand router part, **Heap overload protection configuration** and **CPU overload protection configuration**, respectively.



☑ **Enable overload protection**

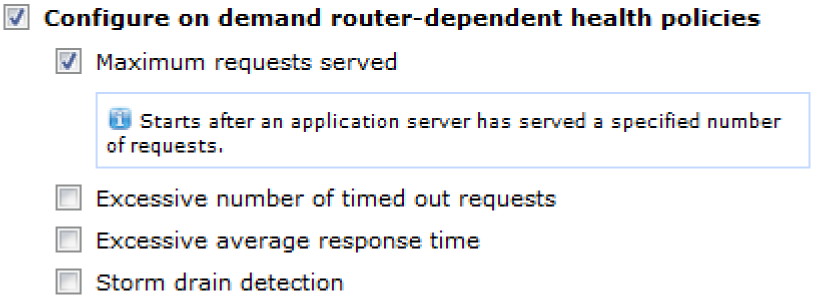
☑ Memory overload protection

Controls the rate at which requests without affinity are permitted through the ODR to prevent Java heap utilization from exceeding a user-specified threshold.

☑ CPU overload protection

Controls the rate at which requests without affinity are permitted through the ODR to prevent CPU utilization from exceeding a user-specified threshold.

__e. Lastly, select the option to **Configure on demand router-dependent health policies**. In this section, select the option for **Maximum requests served**. For this option, a script package called **Maximum requests policy configuration** will be added to your on demand router part.



☑ **Configure on demand router-dependent health policies**

☑ Maximum requests served

Starts after an application server has served a specified number of requests.

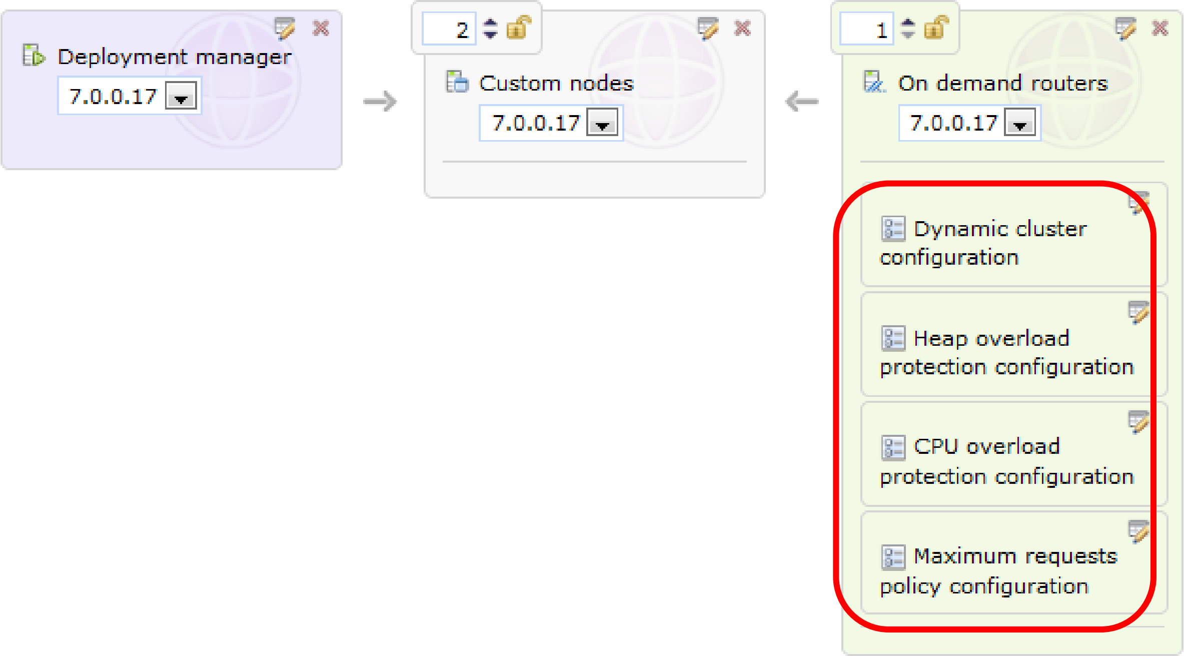
□ Excessive number of timed out requests

□ Excessive average response time

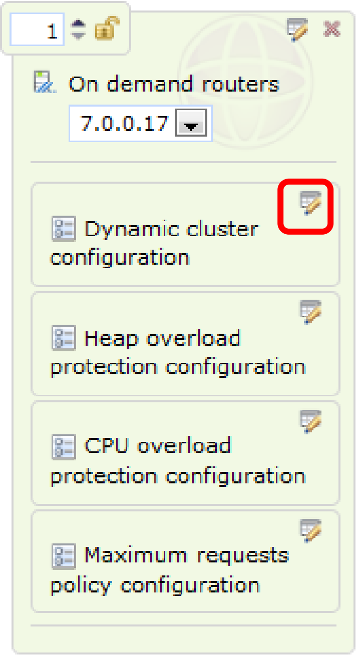
□ Storm drain detection

__f. Click **OK** to close the advanced options panel.

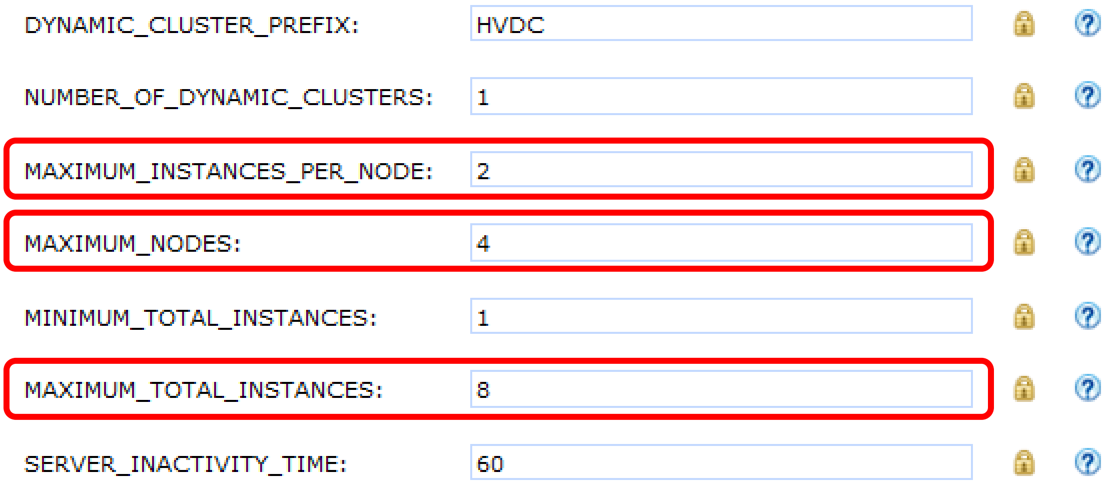
__g. In your pattern, you should see that the four script packages described above have been added to your on demand router part. You will use these script packages to fine-tune the configuration of your cluster and its health policies. Verify that your high-level pattern configuration is the same as this screen capture. This exercise includes a sampling of the advanced options that are available; in your environment you would choose the policies that are necessary to define the health management practices that are important to you.


Deployment manager
7.0.0.17
2
Custom nodes
7.0.0.17
1
On demand routers
7.0.0.17
Dynamic cluster configuration
Heap overload protection configuration
CPU overload protection configuration
Maximum requests policy configuration

__h. The last part of configuring your advanced options is setting the parameters on each of the script packages. Start by selecting the **Parameters** button (✎) on the **Dynamic cluster configuration** script package to open its configuration options.


1
On demand routers
7.0.0.17
Dynamic cluster configuration
Heap overload protection configuration
CPU overload protection configuration
Maximum requests policy configuration

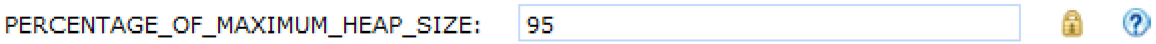
__i. In the parameters panel, you will configure your cluster to allow a maximum of two application server instances per custom node, a maximum of four custom node instances in your dynamic cluster, for a total of eight maximum instances overall. To do this, set the **MAXIMUM_INSTANCES_PER_NODE** field to **2**, the **MAXIMUM_NODES** field to **4**, and the **MAXIMUM_TOTAL_INSTANCES** field to **8**. Also, make a note of the MINIMUM_TOTAL_INSTANCES field, which is set to 1 by default. Keep this default setting. This means that a single application server instance will be started in your virtual system at deployment time.



DYNAMIC_CLUSTER_PREFIX:	HVDC
NUMBER_OF_DYNAMIC_CLUSTERS:	1
MAXIMUM_INSTANCES_PER_NODE:	2
MAXIMUM_NODES:	4
MINIMUM_TOTAL_INSTANCES:	1
MAXIMUM_TOTAL_INSTANCES:	8
SERVER_INACTIVITY_TIME:	60

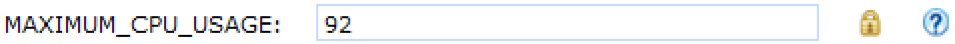
__j. Click **OK** to save the script parameters.

__k. Next, open the parameters panel for the script package Heap overload protection configuration. Here, say that you want to set your policy to take corrective action when your heap usage to 95% of the available heap space. So set the **PERCENTAGE_OF_MAXIMUM_HEAP_SIZE** to **95**. Click **OK** to save the configuration.



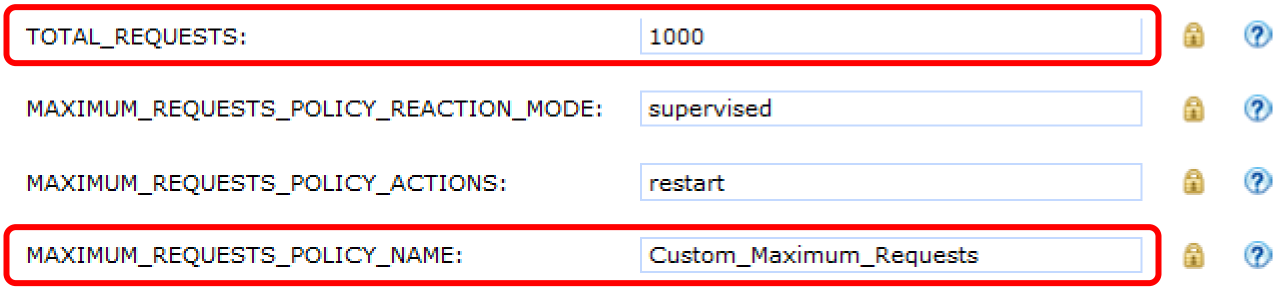
PERCENTAGE_OF_MAXIMUM_HEAP_SIZE: 95

__l. Open the parameters panel for the script package CPU overload protection configuration script package. For this example, say that you want to take corrective action when your CPU utilization goes over 92%. In this case, set the **MAXIMUM_CPU_USAGE** field to **92**, then click **OK**.



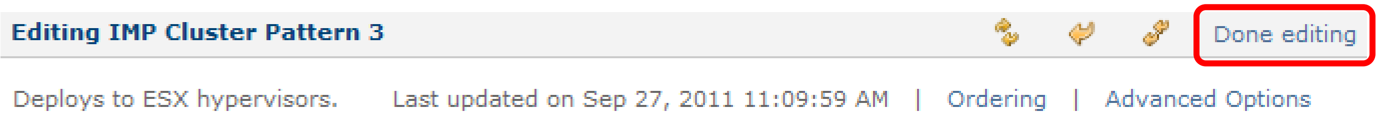
MAXIMUM_CPU_USAGE: 92

__m. Your last script package to configure is Maximum requests policy configuration, so open its parameters. Here, say that you want to set the request limit to 1000 – an artificially low number that would be useful for validating the basic function of the health policy. To accomplish this, set **TOTAL_REQUESTS** to **1000**. Since you modified the default configuration, also update the **MAXIMUM_REQUESTS_POLICY_NAME** field to **Custom_Maximum_Requests**. Keep the default options for doing a supervised restart of the server instance. Click **OK** to save your configuration updates.



TOTAL_REQUESTS:	1000
MAXIMUM_REQUESTS_POLICY_REACTION_MODE:	supervised
MAXIMUM_REQUESTS_POLICY_ACTIONS:	restart
MAXIMUM_REQUESTS_POLICY_NAME:	Custom_Maximum_Requests

__n. Now you have finished configuring your virtual system pattern. Click the **Done editing** link in the upper right of the pattern editor to save all of your changes.



Editing IMP Cluster Pattern 3

Deploys to ESX hypervisors. Last updated on Sep 27, 2011 11:09:59 AM | Ordering | Advanced Options

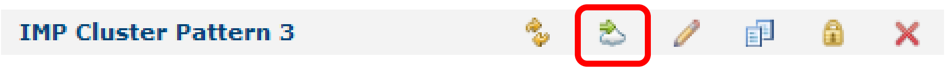
2.2 Instantiate an enhanced pattern

2.2.1 Deploy the custom Intelligent Management Pack pattern

__1. Deploy your pattern into the cloud.

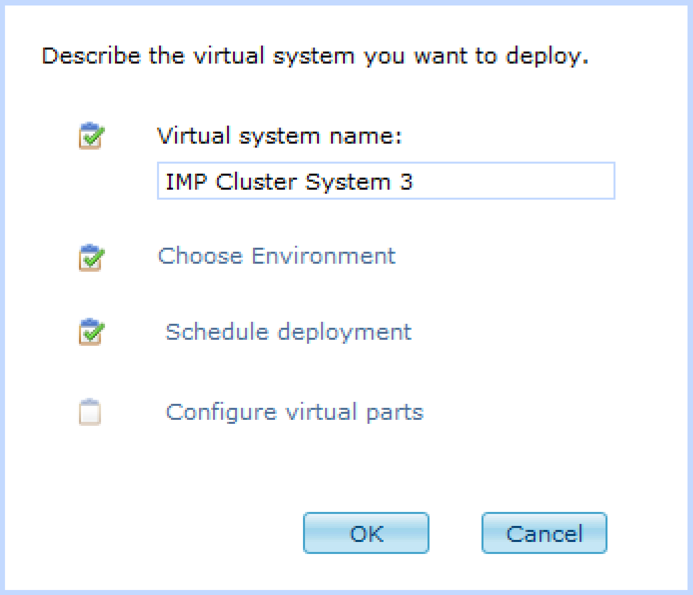
__a. To review, you just created a custom pattern with one deployment manager, two custom nodes, and an on demand router. You used the pattern advanced options to enable dynamic clustering and a set of health policies. Now you are ready to deploy that pattern.

__b. If you are not already there, go to the details page for your pattern (**Patterns > Virtual Systems > IMP Cluster Pattern <N>**). In the pattern toolbar, select the **Deploy in the cloud…** button ( ) to open the deployment dialog.



IMP Cluster Pattern 3

__c. In the **Virtual system name** field, use **IMP Cluster System <N>**, where <N> is your user number. Keep the default options for the **Choose environment** and **Schedule deployment** sections.



Describe the virtual system you want to deploy.

✓ Virtual system name:

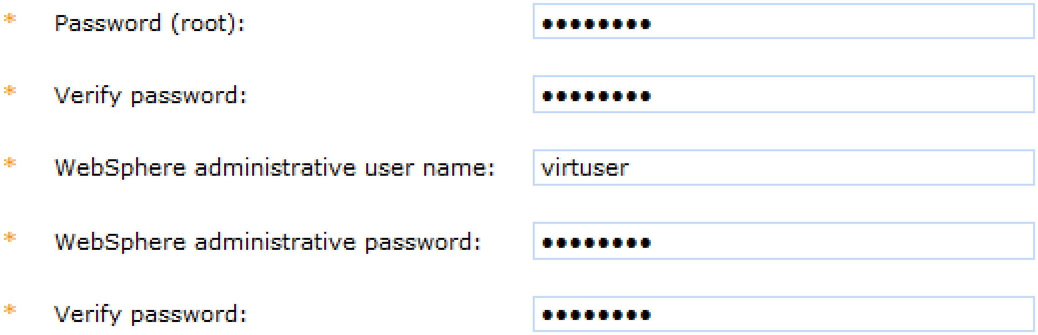
IMP Cluster System 3

✓ Choose Environment

✓ Schedule deployment

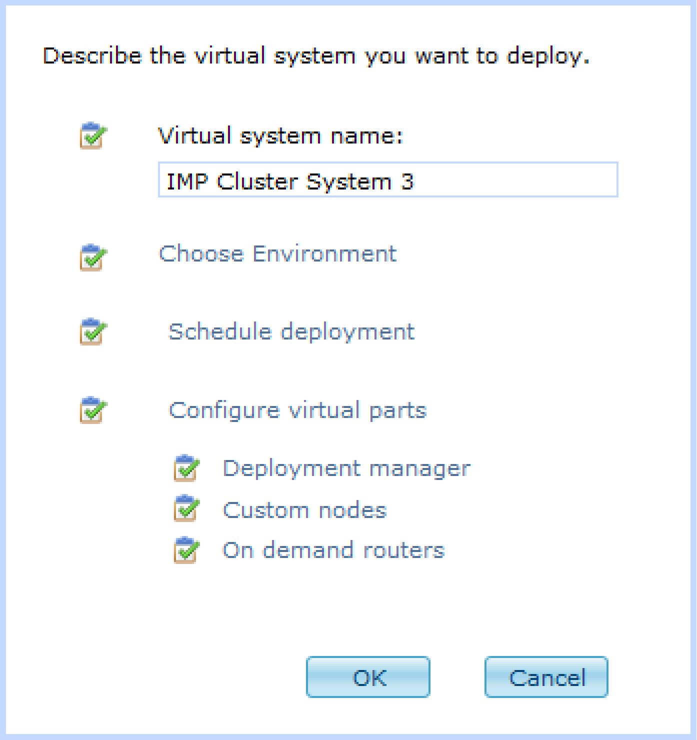
□ Configure virtual parts

__d. Click **Configure virtual parts** to expand the parts section, then choose **Deployment manager** to open the deployment manager configuration dialog. Here, all you need to do is provide a password for the root user account and the WebSphere administrative user account. For **Password (root)** and **Verify password** use **password**. Also, for **WebSphere administrative password** and **Verify password**, use **password**. Click **OK**.



* Password (root):	••••••••
* Verify password:	••••••••
* WebSphere administrative user name:	virtuser
* WebSphere administrative password:	••••••••
* Verify password:	••••••••

__e. Next, open the **Custom node** part properties and set the **root** and **WebSphere administrative** passwords to **password**. Click **OK**. Do the same thing for the on demand router part. After filling in all of your passwords, all of the sections of the deployment dialog will show a green check mark status (✓). This means that you are ready to start the deployment.



Describe the virtual system you want to deploy.

✓ Virtual system name:

IMP Cluster System 3

✓ Choose Environment

✓ Schedule deployment

✓ Configure virtual parts

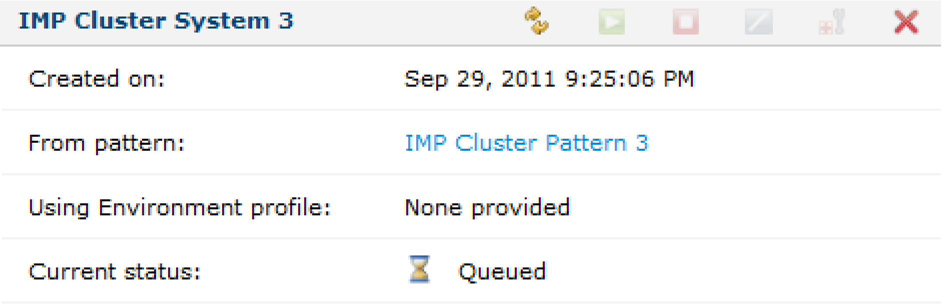
✓ Deployment manager

✓ Custom nodes

✓ On demand routers

__f. Click **OK** to start the deployment process.

__g. You will be redirected to the virtual system instance page for the instance that is currently deploying (you can also get here by navigating to **Instances > Virtual Systems > IMP Cluster System <N>**). Initially, the status of your instance will be displayed as **Queued**. This means that the appliance is getting ready to start instantiating your virtual system.



IMP Cluster System 3

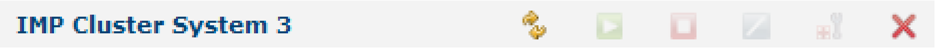
Created on: Sep 29, 2011 9:25:06 PM

From pattern: IMP Cluster Pattern 3

Using Environment profile: None provided

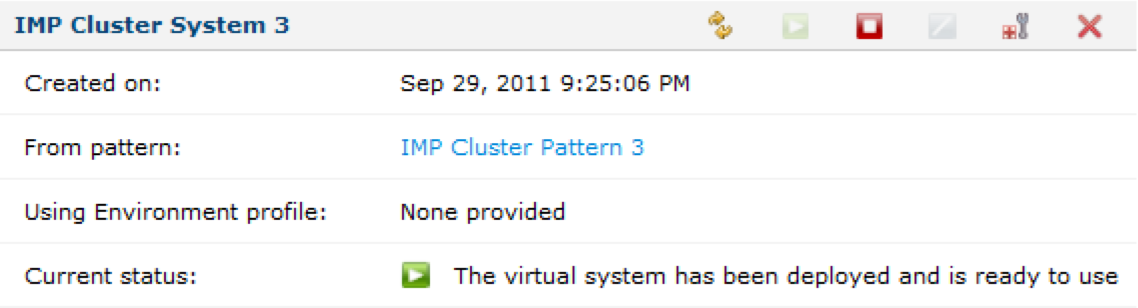
Current status: ⌛ Queued

__h. Your pattern will take **at least 30 minutes** to deploy. Periodically click the **Refresh** button (⟳) in the virtual system toolbar to view updated status for your system.



IMP Cluster System 3

__i. When your virtual system status shows **Started** (▶) and is displaying a message saying that **The virtual system has been deployed and is ready to use**.

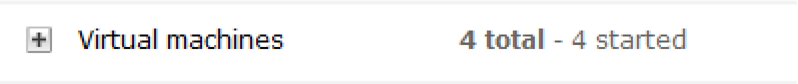


IMP Cluster System 3

Created on:	Sep 29, 2011 9:25:06 PM
From pattern:	IMP Cluster Pattern 3
Using Environment profile:	None provided
Current status:	▶ The virtual system has been deployed and is ready to use

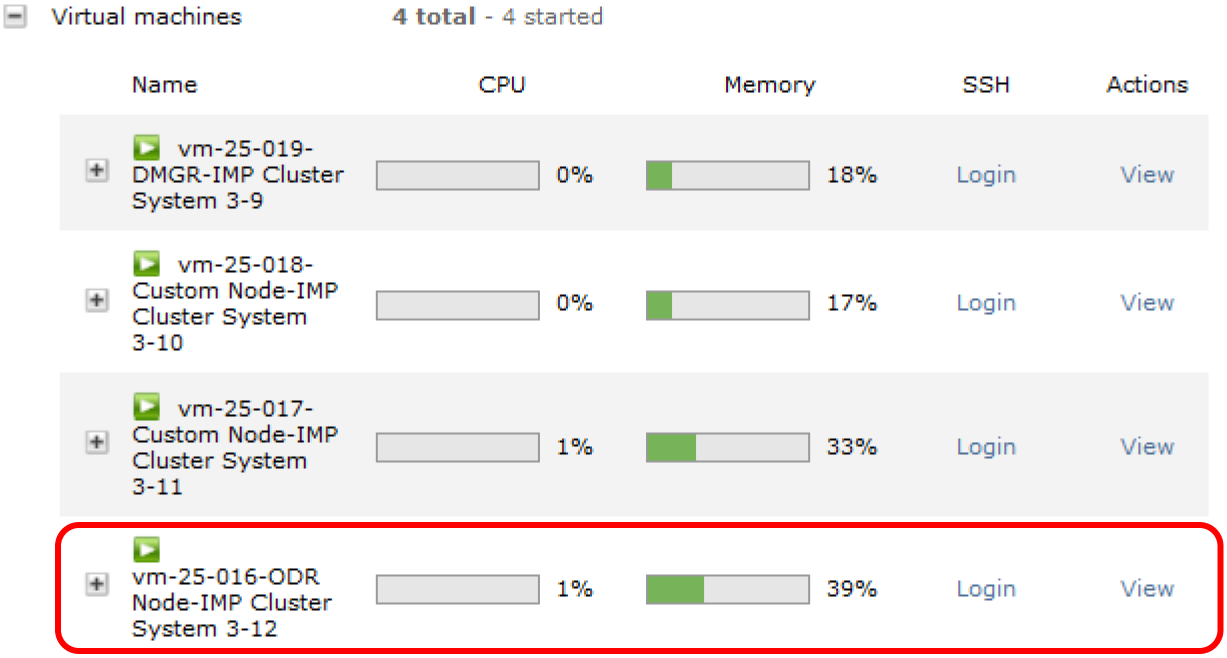
__2. Verify the basic configuration of your virtual system.

__a. You should already be on the Instance page for your virtual system **IMP Cluster System <N>** (if you are not, you can get there from **Instances > Virtual Systems**). Click the twistie (⊞) to expand the **Virtual machines** section.



⊞ Virtual machines **4 total** - 4 started

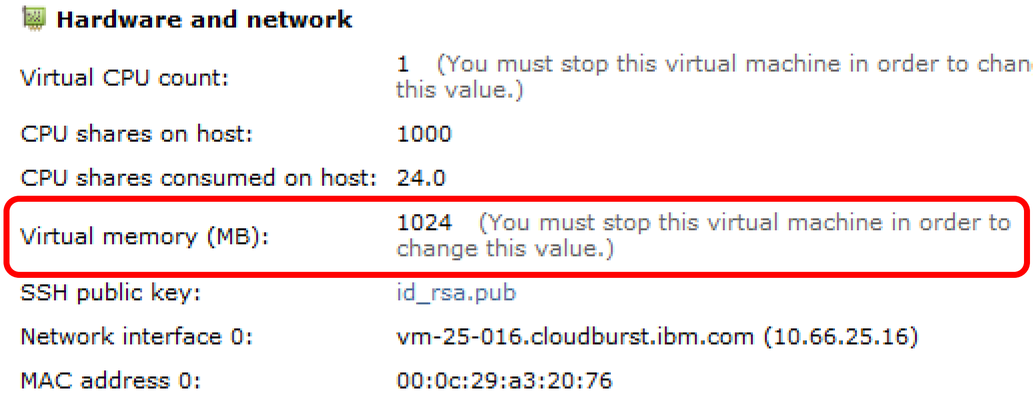
__b. There are four virtual machines in your virtual system – one deployment manager (listed first), two custom nodes, and one on demand router (listed last). Click the twistie (⊞) to expand the **ODR** virtual machine.



⊟ Virtual machines **4 total** - 4 started

	Name	CPU	Memory	SSH	Actions
⊞	▶ vm-25-019-DMGR-IMP Cluster System 3-9	0%	18%	Login	View
⊞	▶ vm-25-018-Custom Node-IMP Cluster System 3-10	0%	17%	Login	View
⊞	▶ vm-25-017-Custom Node-IMP Cluster System 3-11	1%	33%	Login	View
⊞	▶ vm-25-016-ODR Node-IMP Cluster System 3-12	1%	39%	Login	View

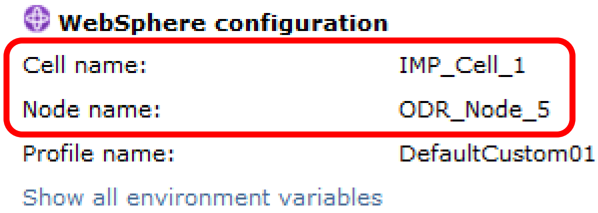
__c. In the **Hardware and network** section, verify that **Virtual memory (MB)** is set to **1024**. This is the value that you configured in the part properties on the on demand router:



Hardware and network

Virtual CPU count:	1 (You must stop this virtual machine in order to change this value.)
CPU shares on host:	1000
CPU shares consumed on host:	24.0
Virtual memory (MB):	1024 (You must stop this virtual machine in order to change this value.)
SSH public key:	id_rsa.pub
Network interface 0:	vm-25-016.cloudburst.ibm.com (10.66.25.16)
MAC address 0:	00:0c:29:a3:20:76

__d. In the **WebSphere configuration** section, verify that the cell name starts with **IMP_Cell**, and that the Node name starts with **ODR_Node**. These are the values that you specified in your pattern parts.

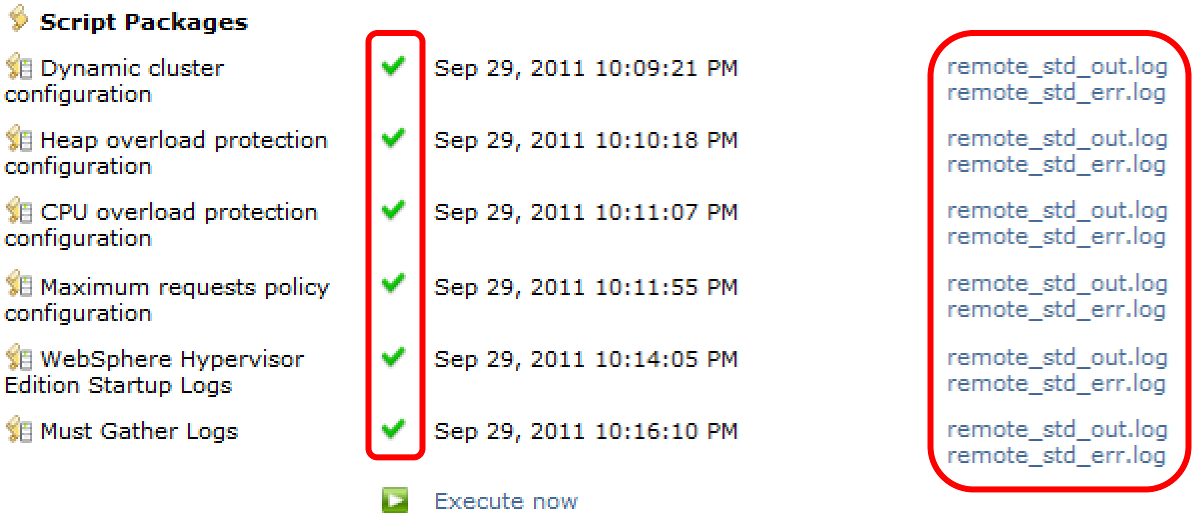


WebSphere configuration

Cell name:	IMP_Cell_1
Node name:	ODR_Node_5
Profile name:	DefaultCustom01

Show all environment variables

__e. Finally, in the **Script packages** section, you should see a success status indicator (✓) next to all of the script packages listed. These script package set up your dynamic cluster and the health policies in your deployed instance. If any of the script packages is failed, you can use the script package logs to investigate the failure.



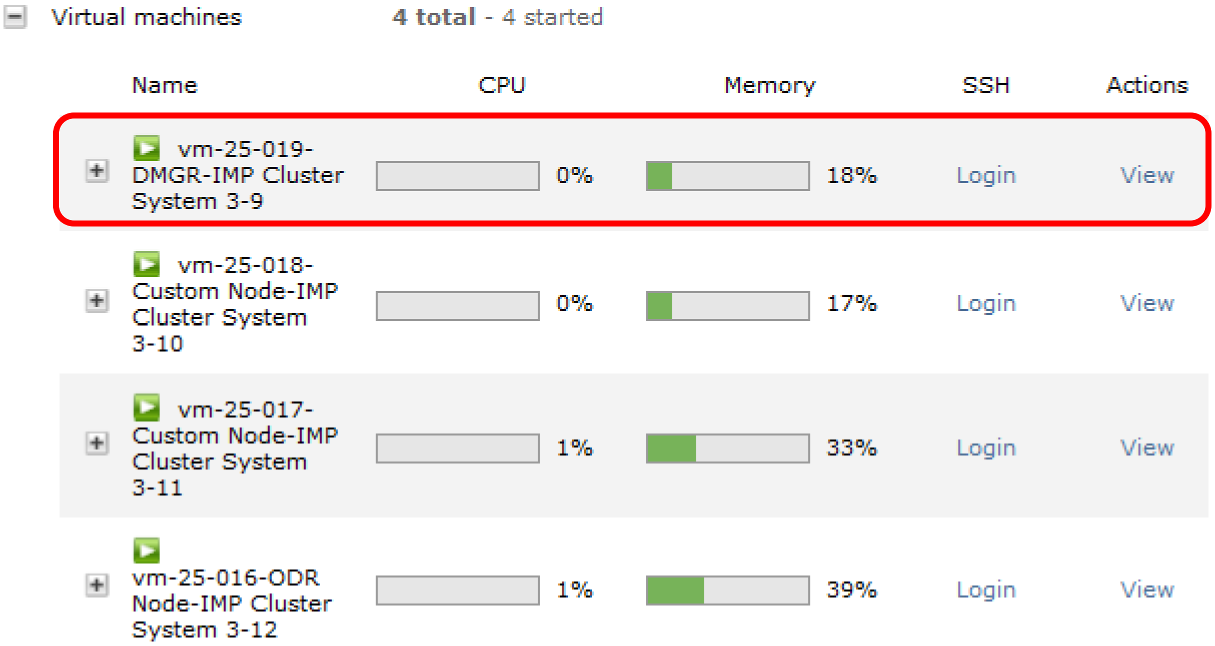
Script Packages

Dynamic cluster configuration	✓	Sep 29, 2011 10:09:21 PM	remote_std_out.log remote_std_err.log
Heap overload protection configuration	✓	Sep 29, 2011 10:10:18 PM	remote_std_out.log remote_std_err.log
CPU overload protection configuration	✓	Sep 29, 2011 10:11:07 PM	remote_std_out.log remote_std_err.log
Maximum requests policy configuration	✓	Sep 29, 2011 10:11:55 PM	remote_std_out.log remote_std_err.log
WebSphere Hypervisor Edition Startup Logs	✓	Sep 29, 2011 10:14:05 PM	remote_std_out.log remote_std_err.log
Must Gather Logs	✓	Sep 29, 2011 10:16:10 PM	remote_std_out.log remote_std_err.log

2.2.2 Verify the dynamic cluster configuration

__1. Access the administrative console.

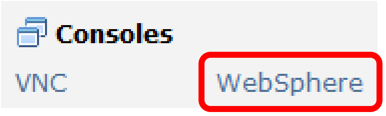
__a. From the list of virtual machines, click the twistie (⊞) to expand the details for the **DMGR** virtual machine (this should be the first one in the list).



⊟ Virtual machines **4 total** - 4 started

	Name	CPU		Memory		SSH	Actions
⊞	▶ vm-25-019-DMGR-IMP Cluster System 3-9		0%	🟩	18%	Login	View
⊞	▶ vm-25-018-Custom Node-IMP Cluster System 3-10		0%	🟩	17%	Login	View
⊞	▶ vm-25-017-Custom Node-IMP Cluster System 3-11		1%	🟩	33%	Login	View
⊞	▶ vm-25-016-ODR Node-IMP Cluster System 3-12		1%	🟩	39%	Login	View

__b. Scroll down to the bottom of the virtual machine details and locate the **Consoles** section. Click the **WebSphere** link to open the administrative console for the deployment manager.

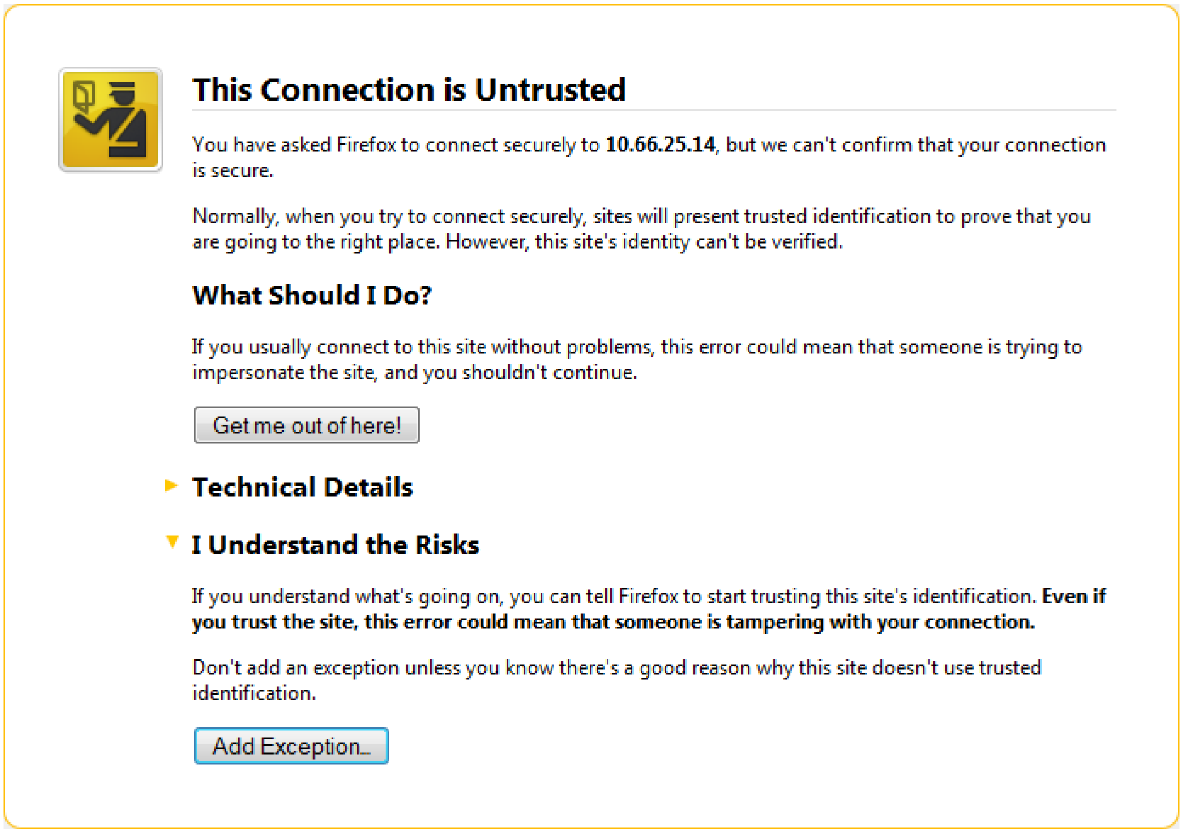


Consoles

VNC WebSphere

__c. The administrative console will open in a new tab in your browser. The browser will warn you that you are trying to make an untrusted connection, or that the site's security certificate has a problem. You need to define a security exception to proceed. If you see a server connection error (and not a security error) this usually means that there is a problem with the DNS configuration on your local workstation. In this case, you can access the administrative console using the IP address of your deployment manager virtual machine (located in the virtual machine details, in the Hardware and network section under Network interface 0). To access the console using the IP address, use a URL of the format http://<DMGR_IP>:9060/ibm/console, replacing <DMGR_IP> with your IP address. This will redirect you to the secure console interface.

__i. To create a security exception in **Firefox**, for example, expand **the I Understand the Risks** section, and click the button to **Add Exception…**. When prompted, click through to **Confirm Security Exception** to access the administrative console.



This Connection is Untrusted

You have asked Firefox to connect securely to **10.66.25.14**, but we can't confirm that your connection is secure.

Normally, when you try to connect securely, sites will present trusted identification to prove that you are going to the right place. However, this site's identity can't be verified.

What Should I Do?

If you usually connect to this site without problems, this error could mean that someone is trying to impersonate the site, and you shouldn't continue.

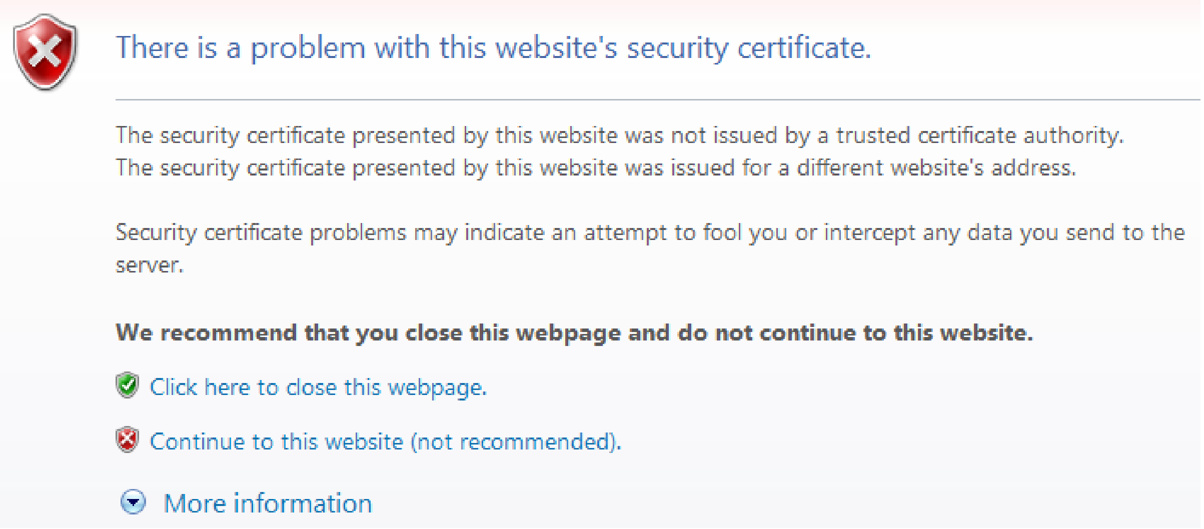
▶ Technical Details

▼ I Understand the Risks

If you understand what's going on, you can tell Firefox to start trusting this site's identification. **Even if you trust the site, this error could mean that someone is tampering with your connection.**

Don't add an exception unless you know there's a good reason why this site doesn't use trusted identification.

__ii. In **Internet Explorer**, click the link to **Continue to this website (not recommended)**. The browser will continue to display a certificate error, but you will be able to use the administrative console.



There is a problem with this website's security certificate.

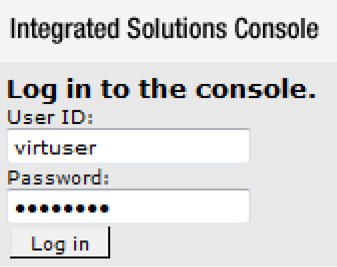
The security certificate presented by this website was not issued by a trusted certificate authority.
The security certificate presented by this website was issued for a different website's address.

Security certificate problems may indicate an attempt to fool you or intercept any data you send to the server.

We recommend that you close this webpage and do not continue to this website.

Click here to close this webpage.
Continue to this website (not recommended).
More information

__d. Log in to the administrative console using with user ID **virtuser** and password **password** (these are the values the you configured at pattern deployment time).



Integrated Solutions Console

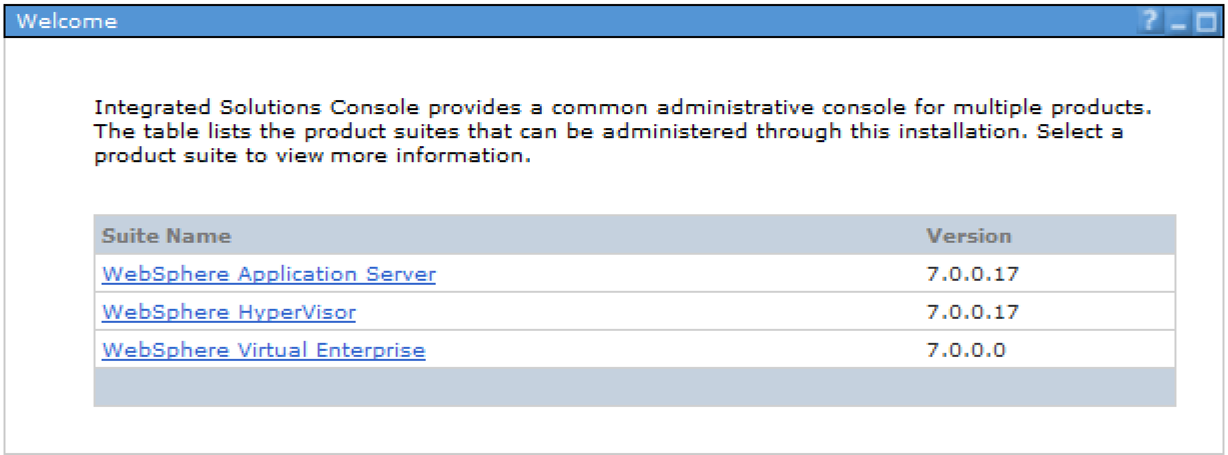
Log in to the console.

User ID: virtuser

Password: •••••••••

__2. Verify the basic configuration of your dynamic cluster.

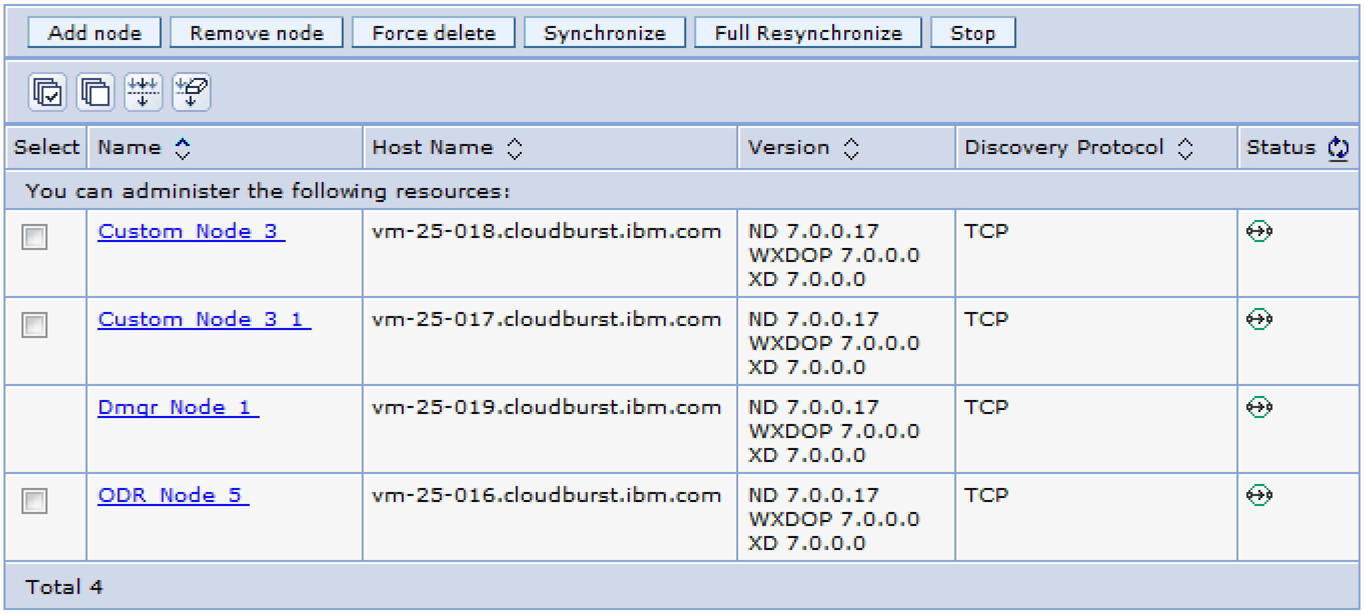
__a. The main page of the console shows the WebSphere Application Server and WebSphere Virtual Enterprise versions that you deployed:



Integrated Solutions Console provides a common administrative console for multiple products. The table lists the product suites that can be administered through this installation. Select a product suite to view more information.

Suite Name	Version
WebSphere Application Server	7.0.0.17
WebSphere HyperVisor	7.0.0.17
WebSphere Virtual Enterprise	7.0.0.0

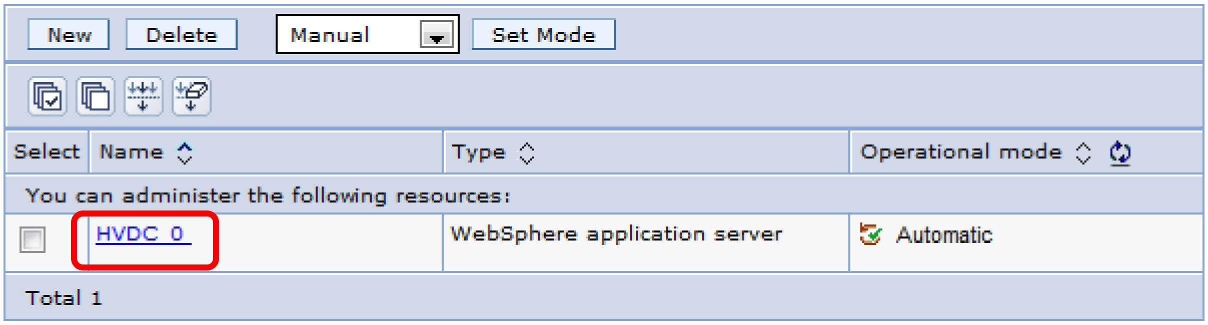
__b. To verify that your nodes were created and properly federated, navigate to **System administration > Nodes**. You should see four nodes – a deployment manager node, two custom nodes, and an on demand router node. Each node will contain the profile type in the node name, since this is what you specified in your virtual system pattern. Your list will look similar to the example below (some of the numbers in your node names could be different – that is ok). Each of the nodes should have the synchronized symbol (⇔) showing up in the **Status** column. The values in the **Host Name** column for each node correspond to the host names / IP addresses that are provided in the Virtual machine section of the IBM Workload Deployer virtual system that you deployed.



Select	Name	Host Name	Version	Discovery Protocol	Status
You can administer the following resources:					
□	Custom_Node_3	vm-25-018.cloudburst.ibm.com	ND 7.0.0.17 WXDOP 7.0.0.0 XD 7.0.0.0	TCP	⇔
□	Custom_Node_3_1	vm-25-017.cloudburst.ibm.com	ND 7.0.0.17 WXDOP 7.0.0.0 XD 7.0.0.0	TCP	⇔
	Dmgr_Node_1	vm-25-019.cloudburst.ibm.com	ND 7.0.0.17 WXDOP 7.0.0.0 XD 7.0.0.0	TCP	⇔
□	ODR_Node_5	vm-25-016.cloudburst.ibm.com	ND 7.0.0.17 WXDOP 7.0.0.0 XD 7.0.0.0	TCP	⇔
Total 4					

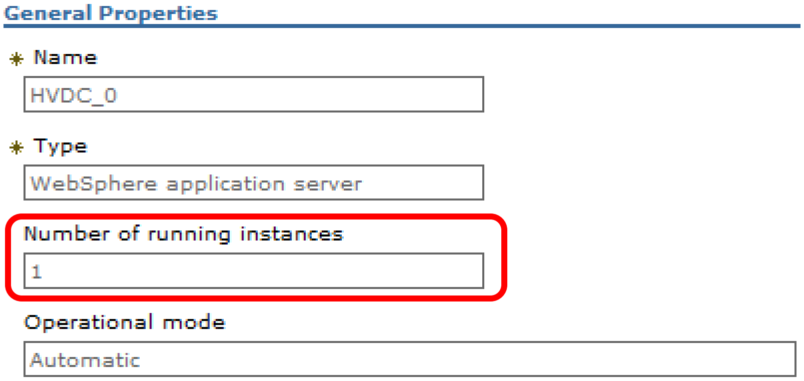
__c. **Do not proceed with the lab if you do not see 4 nodes deployed and synchronized.** Occasionally the nodes will be temporarily unsynchronized if you've taken some action to modify the configuration, but they will synchronize again usually within one or two minutes.

__d. To verify that your dynamic cluster was created correctly, navigate to **Servers > Clusters > Dynamic clusters**. Here, you should see one dynamic cluster listed, **HVDC_0** (recall that you specified HVDC as the dynamic cluster prefix in your virtual system pattern definition).



Select	Name	Type	Operational mode
You can administer the following resources:			
□	HVDC_0	WebSphere application server	✓ Automatic
Total 1			

__e. Select the text **HVDC_0** to view more configuration details of your dynamic cluster. In the **General Properties** section, verify that the **Number of running instances** is **1**. Because you have not applied any load to your dynamic cluster to cause it to spin up additional instances, the current number of running instances value corresponds to the **MINIMUM_TOTAL_INSTANCES** property that you configured in your virtual system pattern – which was 1.



General Properties

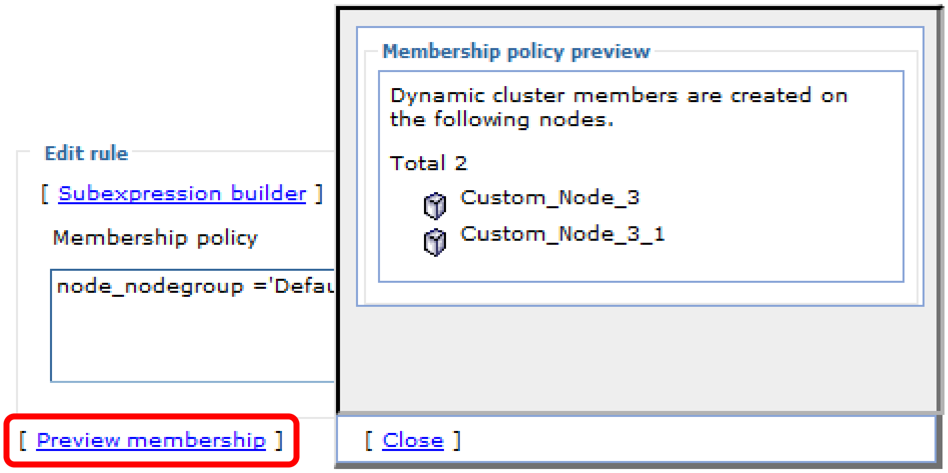
* Name
HVDC_0

* Type
WebSphere application server

Number of running instances
1

Operational mode
Automatic

__f. Further down in the **General Properties** section, locate the area labeled **Edit rule** – this section defines the membership policy for the nodes in your dynamic cluster. By default, this rule has been configured to include all of the custom nodes in the **DefaultCustomNodeGroup**. For your dynamic cluster, this should include both of the custom nodes that you deployed as a part of your virtual system pattern. Click the link to **Preview membership** to bring up a list of the dynamic clusters. You should see that both **Custom_Node_3** and **Custom_Node_3_1** are members of this cluster (your configuration might use a different number than 3).



Edit rule
[Subexpression builder]
Membership policy
node_nodegroup ='Defau
[Preview membership]

Membership policy preview

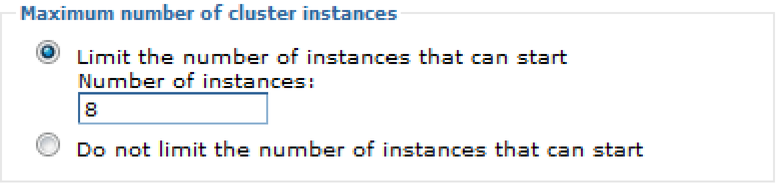
Dynamic cluster members are created on the following nodes.

Total 2
Custom_Node_3
Custom_Node_3_1

[Close]

__g. Click **Close** to close the **Membership policy preview** window.

__h. Further down in the dynamic cluster configuration page, locate the area labeled Maximum number of cluster instances. You should see that this field is set to **Limit the number of instances that can start**, with a maximum number of instances set to **8**. This corresponds to the value **MAXIMUM_TOTAL_INSTANCES** that you configured in your virtual system pattern.

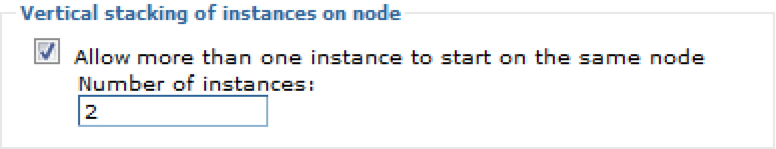


Maximum number of cluster instances

● Limit the number of instances that can start
Number of instances:
8

○ Do not limit the number of instances that can start

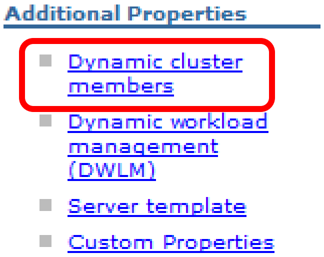
__i. Finally, in the **Vertical stacking of instances on node** section, verify that the option to **Allow more than one instance to start on the same node** is selected, and that the **Number of instances** is set to **3**. This corresponds to the **MAXIMUM_INSTANCES_PER_NODE** value that you configured in your virtual system pattern.



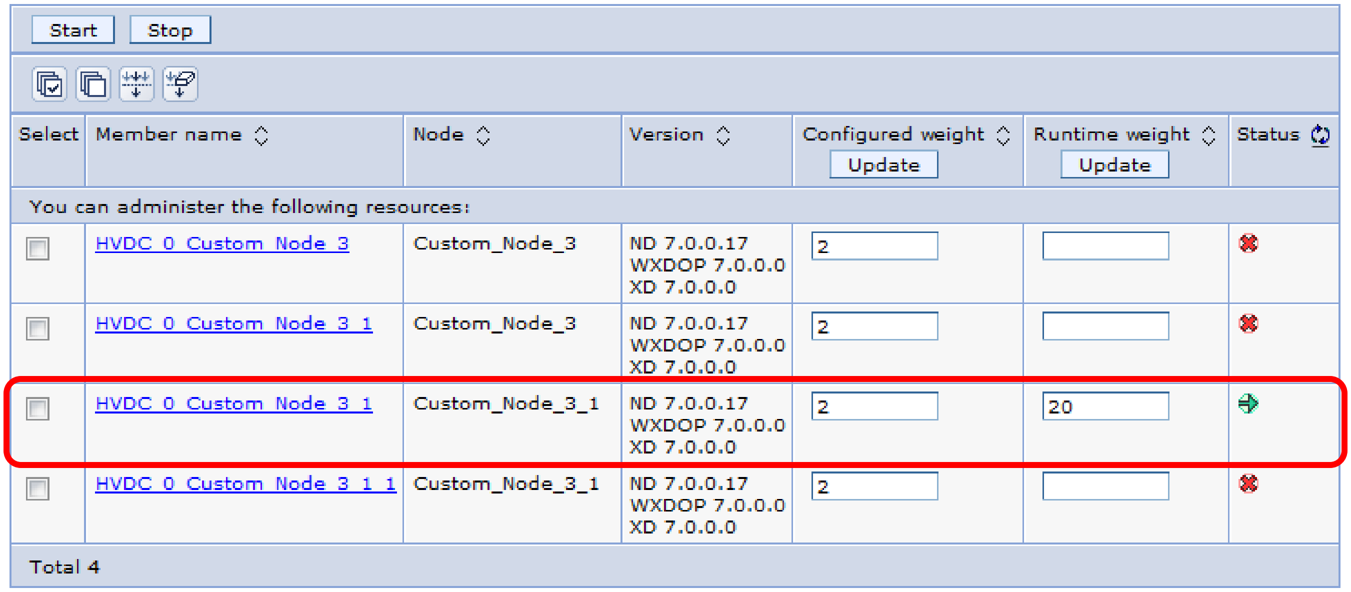
Vertical stacking of instances on node

☑ Allow more than one instance to start on the same node
Number of instances:
2

__j. Since you set the number of allowed vertical instances to 2, that means that 2 application server instances should be created in each of your custom nodes. Also, since one instance is currently running, that means that only 1 of your application server instances should be started. To verify this, locate the **Additional Properties** section in the upper right of your dynamic cluster configuration and click the link for **Dynamic cluster members**.

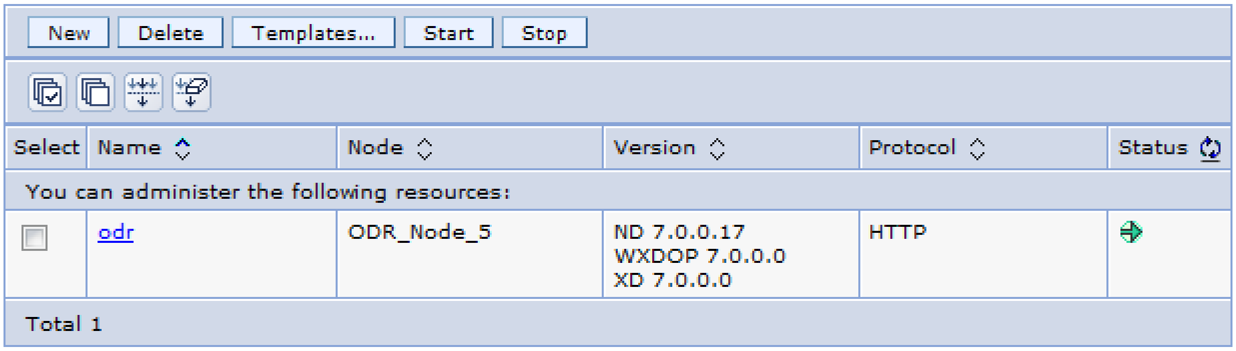


__k. You should see four instances in your dynamic cluster – two instances per custom node. At this time, only one of these instances should be started (➔). This is an example of the member list for your dynamic cluster, showing a single running instance:



Select	Member name ◇	Node ◇	Version ◇	Configured weight ◇ Update	Runtime weight ◇ Update	Status
You can administer the following resources:						
□	HVDC_0_Custom_Node_3	Custom_Node_3	ND 7.0.0.17 WXDOP 7.0.0.0 XD 7.0.0.0	2		✖
□	HVDC_0_Custom_Node_3_1	Custom_Node_3	ND 7.0.0.17 WXDOP 7.0.0.0 XD 7.0.0.0	2		✖
□	HVDC_0_Custom_Node_3_1	Custom_Node_3_1	ND 7.0.0.17 WXDOP 7.0.0.0 XD 7.0.0.0	2	20	➔
□	HVDC_0_Custom_Node_3_1_1	Custom_Node_3_1	ND 7.0.0.17 WXDOP 7.0.0.0 XD 7.0.0.0	2		✖
Total 4						

__l. Next, verify that the on demand router is also started. Navigate to **Servers > Server Types > On Demand Routers** and verify that your on demand router is showing a status of **Started** (➔).



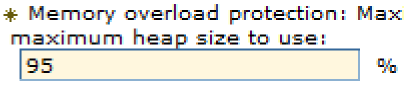
Select	Name ◇	Node ◇	Version ◇	Protocol ◇	Status
You can administer the following resources:					
□	odr	ODR_Node_5	ND 7.0.0.17 WXDOP 7.0.0.0 XD 7.0.0.0	HTTP	➔
Total 1					

__m. Now that you have verified that your dynamic cluster was structured correctly, you will check that your health policies were configured correctly.

__3. Verify the configuration of your overload protection and health policies.

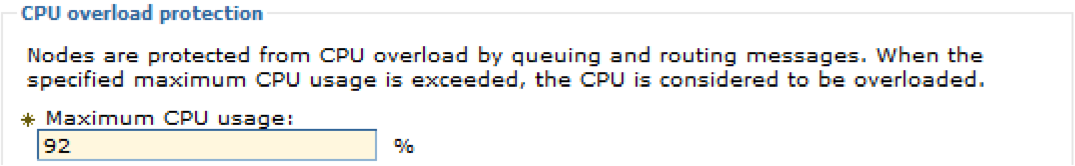
__a. You configured two types of overload protection in your dynamic cluster – memory overload, and CPU overload. Overload protection allows the on demand router to appropriately direct traffic in your dynamic cluster to avoid overloading instances in your cluster.

__b. Navigate to **Operational policies > Autonomic Managers > Automatic Request Flow Manager**. Locate the **Memory overload protection: Maximum percentage of the WebSphere Application Server maximum heap size to use** field, and verify that it is set to **95**. This corresponds to the value **PERCENTAGE_OF_MAXIMUM_HEAP_SIZE** that you configured in the **Heap overload protection configuration** script package in your virtual system pattern.



* Memory overload protection: Maximum percentage of the WebSphere Application Server maximum heap size to use: 95 %

__c. In the **CPU overload protection** section, verify that the **Maximum CPU usage field** is set to **95**. This corresponds to the value **MAXIMUM_CPU_USAGE** that you set on the **CPU overload protection configuration** script package in your pattern.

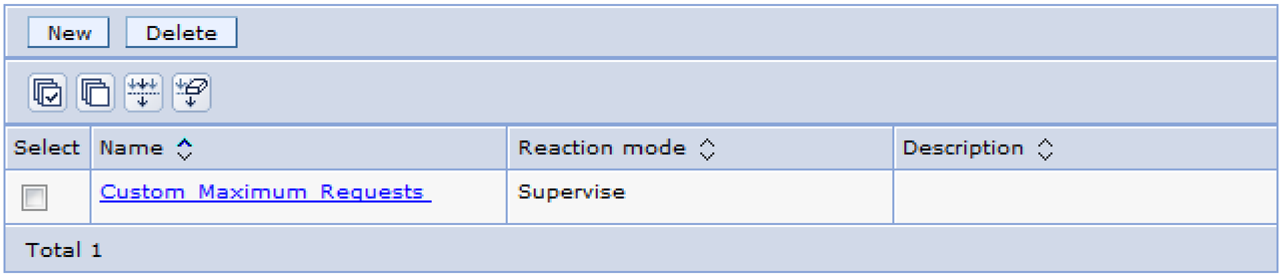


CPU overload protection

Nodes are protected from CPU overload by queuing and routing messages. When the specified maximum CPU usage is exceeded, the CPU is considered to be overloaded.

* Maximum CPU usage: 92 %

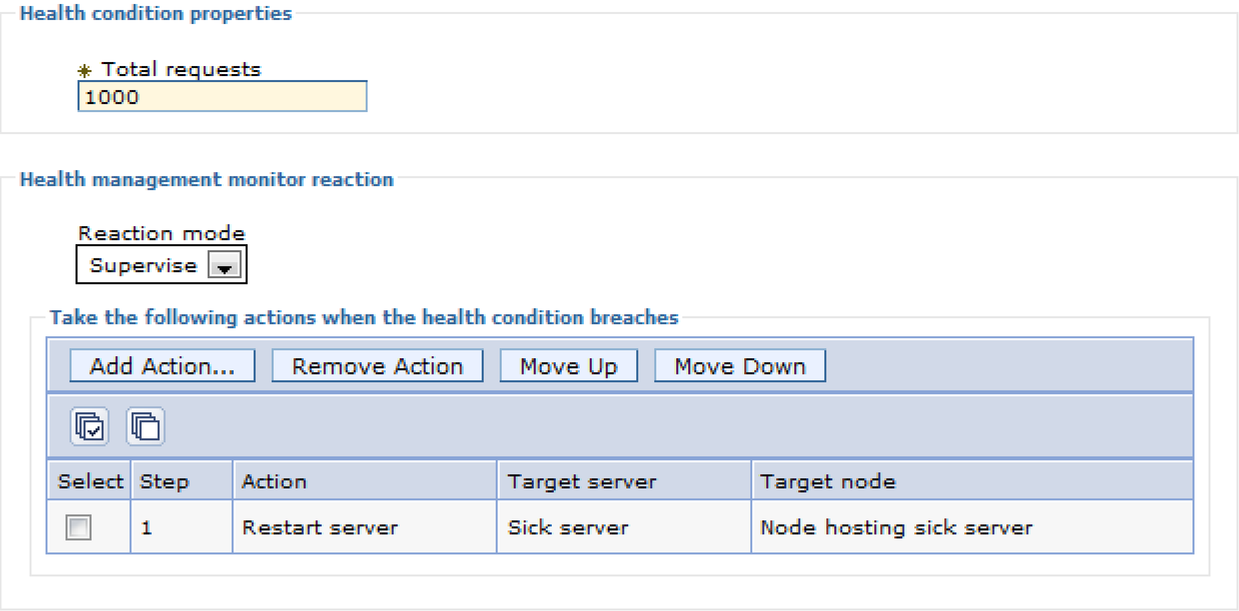
__d. You also configured a maximum requests policy in your dynamic cluster. To see the details of this policy, go to **Operational policies > Health Policies**. From the policy list, select Custom Maximum Requests.



Select	Name	Reaction mode	Description
□	Custom Maximum Requests	Supervise	

Total 1

__e. In the configuration for this health policy in your virtual system pattern, you set the maximum total requests to **1000** (this value was set artificially low, to make it easier to trigger the policy in a lab environment). When your instance receives over 1000 requests, you set your script package to take a **supervised** action to **restart** the server. Verify that these values were captured correctly in your health policy, as shown:



Health condition properties

* Total requests: 1000

Health management monitor reaction

Reaction mode: Supervise

Take the following actions when the health condition breaches

Select	Step	Action	Target server	Target node
□	1	Restart server	Sick server	Node hosting sick server

__f. At this point, you have verified that your dynamic cluster was configured properly, and that all of your policy definitions were defined correctly in your on demand router.

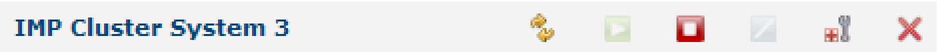
2.3 Clean up your configuration

__1. Delete your virtual system instance

__a. To save resources in the cloud, please clean up your deployed virtual system.

__b. In the IBM Workload Deployer console, go to your virtual system instance page.

__c. In the instance toolbar, click the **Delete** button (✗).



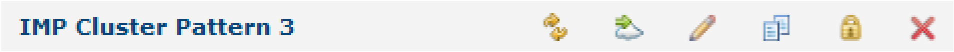
IMP Cluster System 3

__d. All of the virtual machines in your virtual system will be stopped, unregistered, and deleted. It will take several minutes for this operation to complete. You can periodically refresh the browser to see updated status, until the virtual system has been completely deleted.

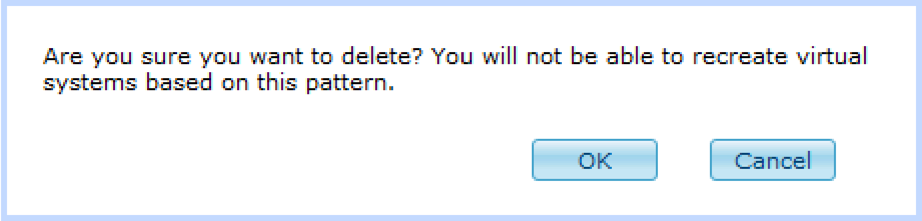
__2. Delete your virtual system pattern.

__a. Navigate to **Patterns > Virtual Systems**. Select **IMP Cluster Pattern <N>**.

__b. In the pattern toolbar, click the **Delete** button (✕).



__c. When prompted to accept the delete operation, click **OK**.


Are you sure you want to delete? You will not be able to recreate virtual systems based on this pattern.

__d. Now your cleanup is complete!

2.4 Summary

In this exercise, you learned how to clone an existing Intelligent Management Pack pattern, customize the structure of the pattern, and add dynamic cluster configuration and health policies using the pattern's advanced options. Then, you deployed your pattern into the cloud and verified that all of the advanced configuration operations ran successfully.

Appendix A. Notices

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

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

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

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106-0032, Japan

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

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

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

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

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

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

All statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only.

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

Appendix B. Trademarks and copyrights

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

IBM	AIX	CICS	ClearCase	ClearQuest	Cloudscape
Cube Views	DB2	developerWorks	DRDA	IMS	IMS/ESA
Informix	Lotus	Lotus Workflow	MQSeries	OmniFind	
Rational	Redbooks	Red Brick	RequisitePro	System i	
System z	*Tivoli*	*WebSphere*	*Workplace*	*System p*	

Adobe, Acrobat, Portable Document Format (PDF), and PostScript are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both. See Java Guidelines

Microsoft, Windows, Windows NT, and the Windows logo are registered trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.

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

ITIL is a registered trademark and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

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

NOTES

NOTES