

Using WebSphere Virtual Enterprise V7 to Build Reliable and Resilient Infrastructures

Lab Exercise 2-3



Lab Number

Contents

| | | |
|--------------|--|-----------|
| LAB 1 | APPLICATION HEALTH MANAGEMENT LAB | 4 |
| | 1.1 PASSWORDS, SHORTCUTS AND STARTING THE IMAGE | 4 |
| | 1.2 APPLICATION HEALTH MANAGEMENT LAB..... | 7 |
| LAB 2 | APPLICATION EDITION MANAGEMENT LAB | 13 |
| LAB 3 | BBSON LAB (OPTIONAL) | 7 |

Lab 1 Application Health Management Lab

1.1 Passwords, shortcuts and starting the image

Password information

VMware Guest Image Authentication

User: Administrator

Password: passw0rd

WebSphere Console Authentication

User: username

Password: password

Naming conventions

File system shortcuts

Lab objectives

- How to use WebSphere Virtual Enterprise to monitor and manage application health without any loss in service
- How to use WebSphere Virtual Enterprise to change application editions without any loss of service
- How to configure WebSphere Virtual Enterprise to use either the high availability manager or bulletin board service overlay network for application infrastructure communications to enhance performance and scalability

Prerequisite knowledge

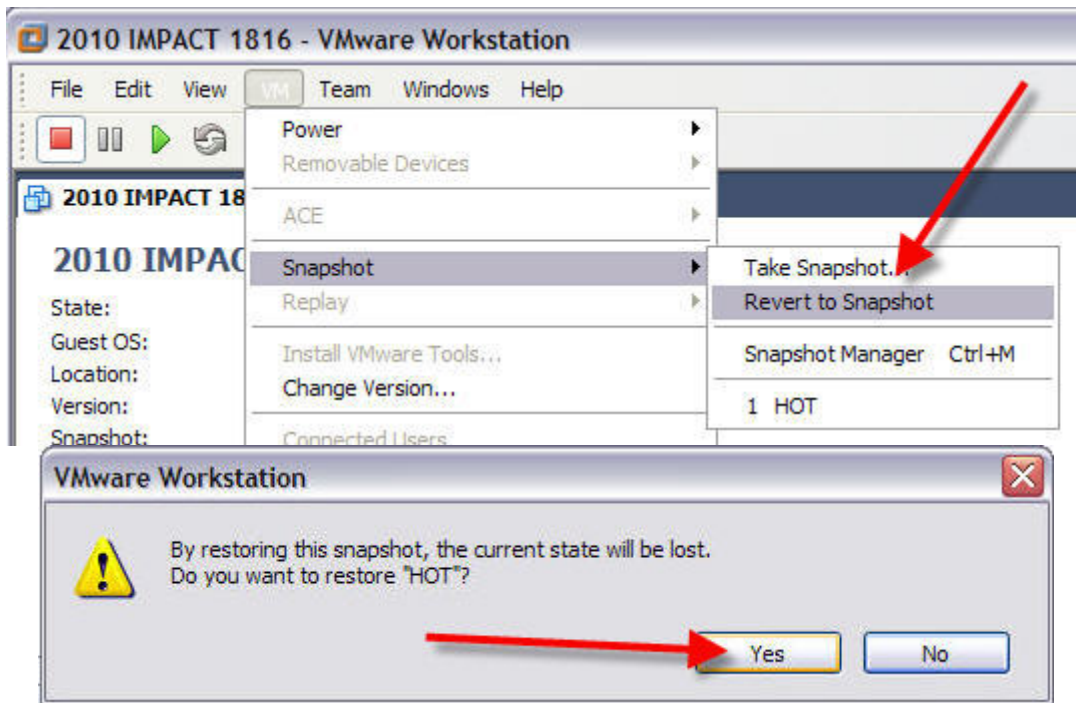
- Basic Microsoft Windows knowledge

- Basic VMware Workstation knowledge
- Basic IBM WebSphere Application Server knowledge

Getting Started

- This virtual machine requires a T60+ Thinkpad with 3GB+ RAM and 10GB HD available
- Open the virtual machine using VMware Workstation 6.5+ by double clicking on the file `base-image-2003-SE-server.vmx`
- Power on the virtual machine by selecting **VM -> Snapshot -> Revert to Snapshot, Yes**, then wait for the guest's CPU to settle as observed in the Task Manager

If you are asked if the image has been copied or moved, select **moved**



1.2 Application Health Management Lab

Health management is the ability of the system to take a policy-driven approach to monitoring the application server environment and taking action when certain predefined criteria are discovered. The health monitoring and management subsystem continuously monitors the operation of servers to detect functional degradation that is related to user application malfunctions. A health policy defines a set of conditions that are interpreted by WebSphere Virtual Enterprise as a degradation of server function. Health policies are a combination of a health condition to monitor, actions to take if the condition occurs, the deployment target to be monitored, and a reaction mode that defines whether the action occurs automatically or with operator intervention.

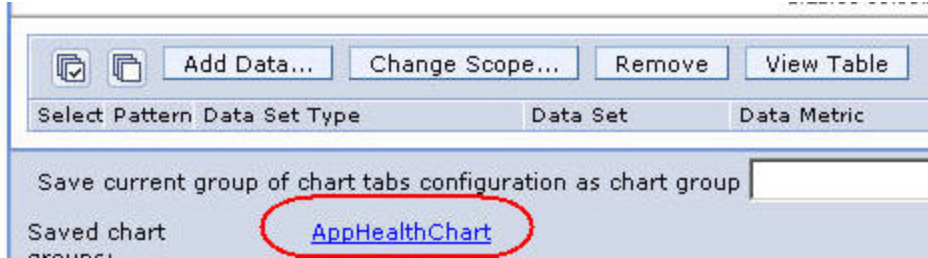
In this section of the lab an application has been deployed to a dynamic cluster. A health policy has been defined that has a health condition identified to be when 50% of the heap has been exceeded for more than 1 minute. When this condition is breached, WebSphere Virtual Enterprise will start additional application resources, route new work to these new resources, drain the troubled resource, and finally restart it, while maintaining continuous availability. You will drive a workload that continuously leaks memory, slowly reducing the amount of free heap, visualizing this using WebSphere Virtual Enterprise's charting capabilities.

At the end of this lab section we will explore the environment using the WebSphere Console.

1. Using the open Firefox browser, log into the WebSphere Console using the values given at the start of this lab for user/password.
2. Navigate in the left-hand frame **Runtime Operations -> Reports**



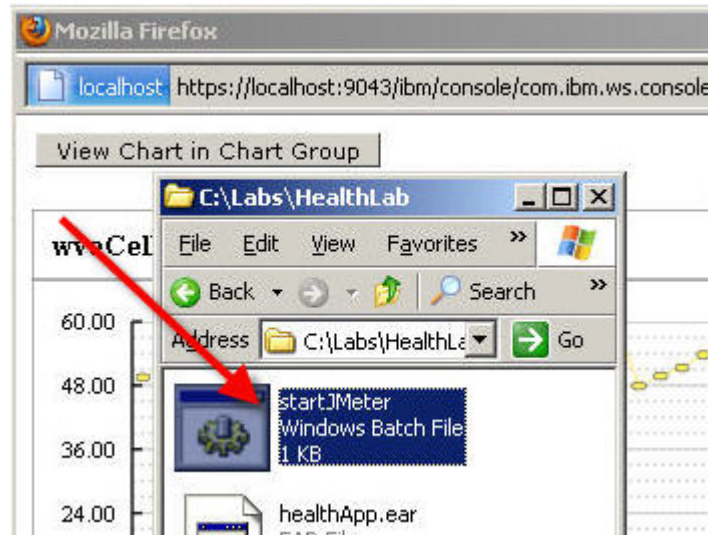
3. Select the saved chart **AppHealthChart** located towards the bottom of the central frame

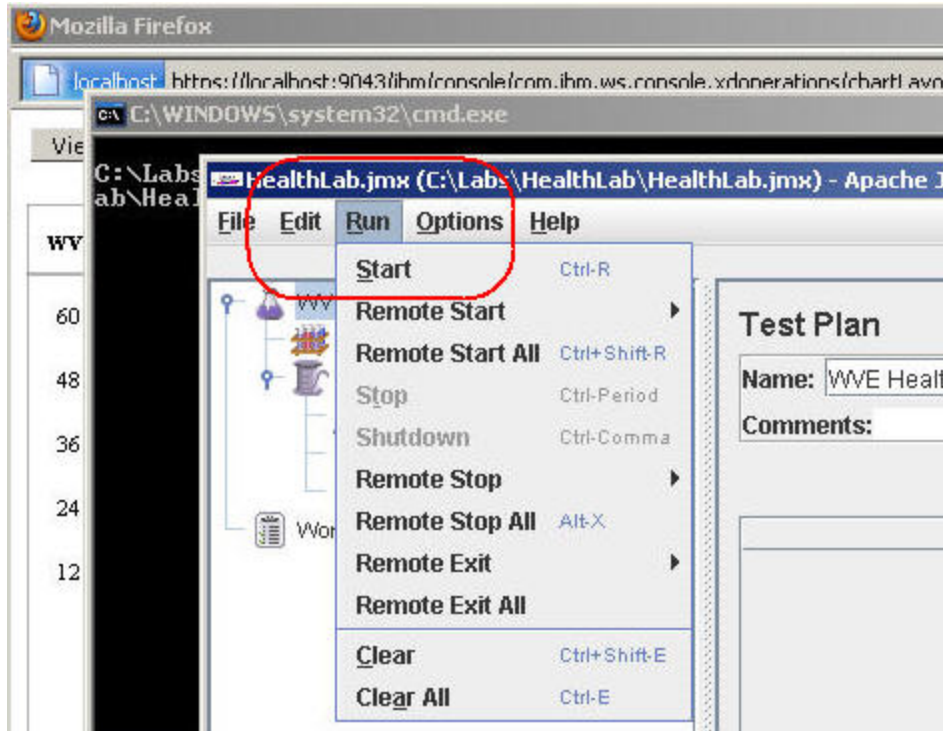


4. Open the chart in a new window by selecting the **View Chart in New Window** button



5. From the Windows Desktop open the **Labs** folder (it should be open), then the **HealthLab** folder
6. Double-click **startJMeter**, then from the JMeter menu bar select **Run -> Start**





7. Watch the browser window with the chart for the following to happen over ~10 minutes:

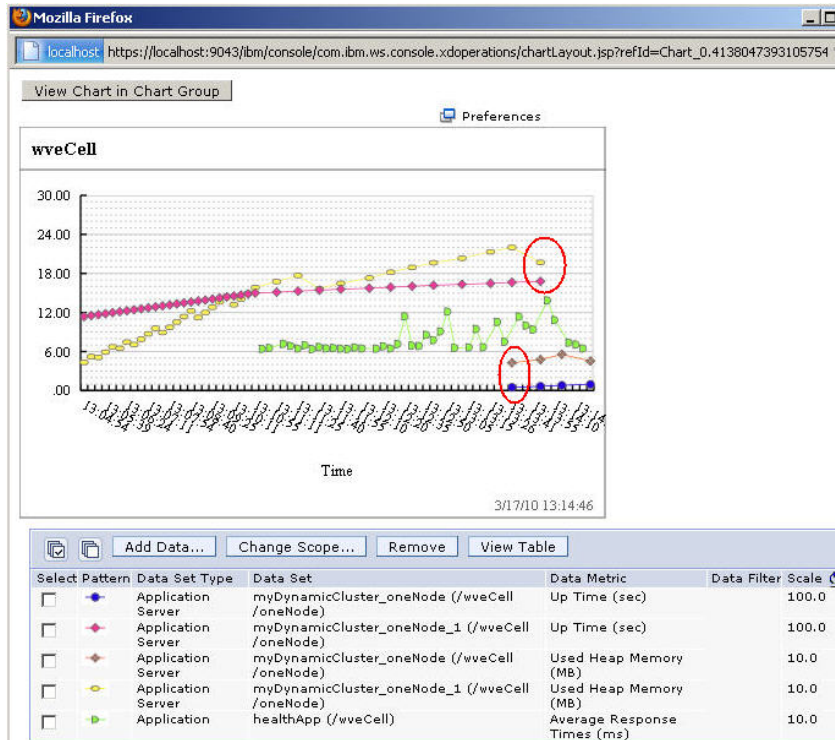
- a new application server instance being started
- the application server instance in ill health being stopped
- consistent service response times

as shown in the following screenshot. The red ovals denote when the instances change state. Notice that even as a new instance is started the response times remain consistent, ensuring that the service level agreements can be met.

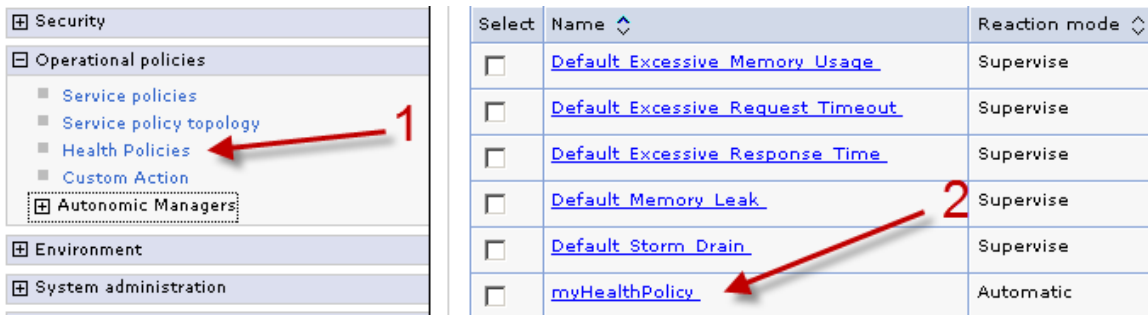
An explanation of the choice of metrics to chart:

- **Up Time** displays the operational state of an application server
- **Average Response Time** indicates the quality of service
- **Used Heap Memory** is the metric followed by the health policy

Notice that as the second instance is started the service times become more erratic. This is due to the fact that all components are running on the same physical resource (your laptop) and there is no strong infrastructure enforcement of component isolation (i.e., VMware Workstation), leading to resource contention. Still, service times remain reasonable and no user transaction is lost.



- When the health policy has run to completion, from the JMeter menu bar select **Run -> Stop, File -> Exit**, then close the browser window displaying the chart by selecting the 'X' in the upper right corner
- Let's explore how health management was configured to achieve this result. From the WebSphere Console, navigate to the health policy we have been watching: **Operational policies -> Health Policies**, select **myHealthPolicy**



The relevant details of our health policy are the size of the JVM heap to trigger a health condition and the time period over which the condition has been breached, the reaction mode that the health controller is in (automatic or supervised) for this condition, the resulting action(s) and the target of the health controller's action(s). In our case, if the JVM heap exceeds 50% of the maximum for 1 minute the health controller will automatically restart the offending application server in the dynamic cluster named myDynamicCluster. As this dynamic cluster's operational mode is set to Automatic, the restart is performed without any loss of service.

Health condition properties

* JVM heap size
50 %

* Offending time period
1 Minutes

Health management monitor reaction

Reaction mode
Automatic

Take the following actions when the health condition breaches

Add Action... Remove Action Move Up Move Down

| Select | Step | Action | Target server | Target node |
|--------------------------|------|----------------|---------------|--------------------------|
| <input type="checkbox"/> | 1 | Restart server | Sick server | Node hosting sick server |

Memberships

Filter by Select a member type

Available for membership

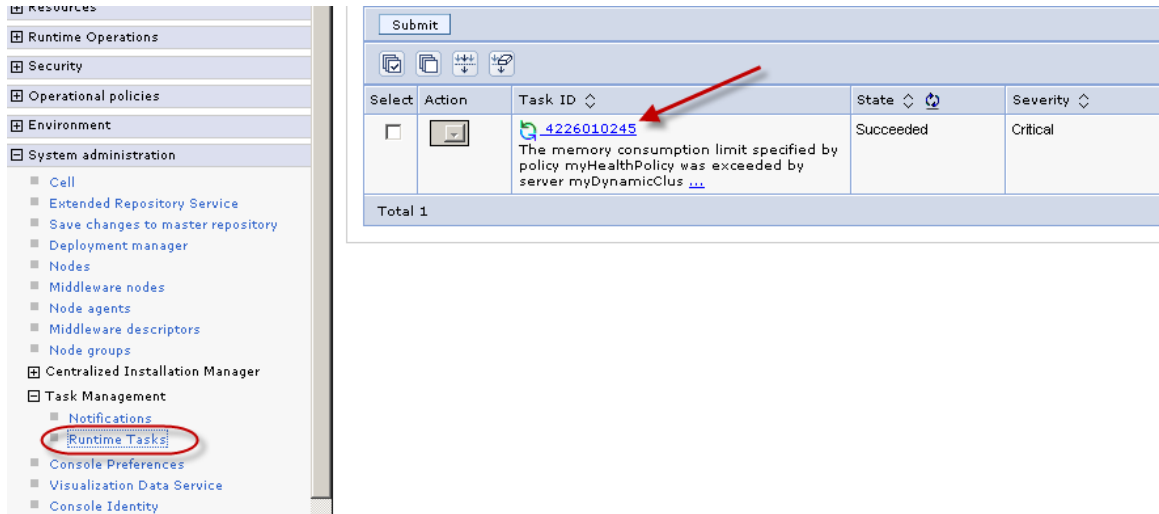
Members of **myHealthPolicy**:

myDynamicCluster (Dynamic clusters)

Add >>

<< Remove

10. Let's explore how health management effected this result. From the WebSphere Console, navigate to the health policy we have monitored: **System administration -> Task Management**, select **Runtime Tasks**. Listed chronologically are the tasks emitted by the system. Find the one that resulted from the health condition breach, and note that the task explanation indicates the cause "The memory consumption limit..." Select this **Task ID** for details of the action(s) taken



The task detail include the explanation, the target of the actions, and the action plan enacted. In our case, the action plan started a second cluster instance before stopping the instance in ill health, ensuring service availability.

Runtime Tasks

[Runtime Tasks](#) > **Task Targets**

Situation description

The memory consumption limit specified by policy myHealthPolicy was exceeded by server myD node oneNode. The limit is 50 % and the current heap size is 61 % of the maximum of 262144

Show additional task detail information

Explore the data used to diagnose the situation

| Target Object | Target Type | Severity |
|----------------------------|-------------|----------|
| myDynamicCluster_oneNode_1 | server | Critical |

Action plan to resolve the situation

The action plan expires at 3/2/11 10:45:33.

Step 1 : Start server myDynamicCluster_oneNode on node oneNode.

Step 2 : Stop server myDynamicCluster_oneNode_1 on node oneNode.

Extra credit: check that indeed that myDynamicCluster has an operational mode of Automatic; what is the JVM heap size in MB at which the health controller is triggered?

This concludes section one of the WebSphere Virtual Enterprise lab.

Lab 2 Application Edition Management Lab

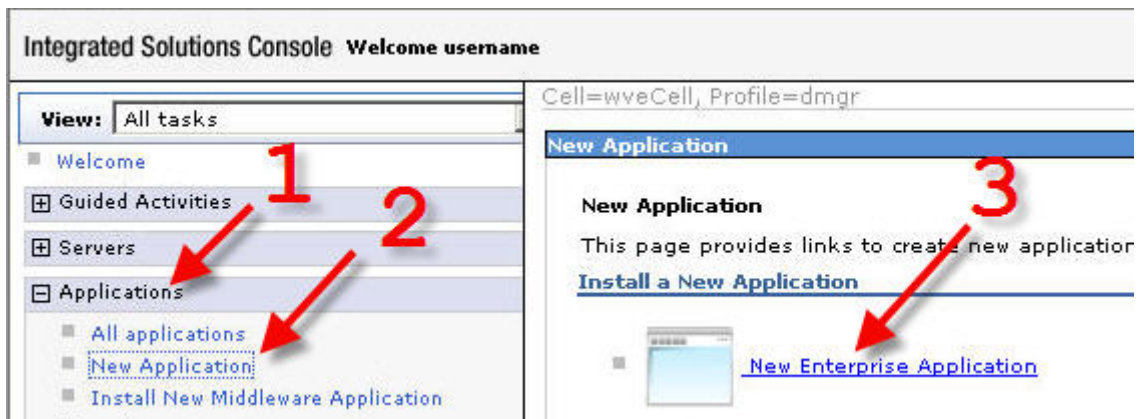
WebSphere Virtual Enterprise contains an application edition manager that provides advanced application management capabilities.

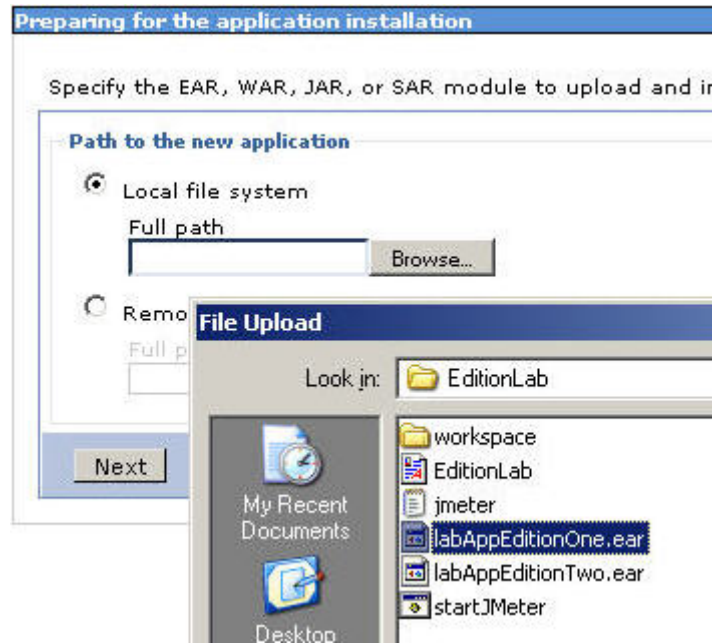
These capabilities address five functional requirements:

- Application versioning capability.
- Interruption-free application rollout.
- Ability to back-out an application update.
- A validation mode to test and verify an application before rolling it out to production.
- Ability to host concurrent versions of an application to support needs of more complicated rollout strategies, such as piloting and branch upgrade.

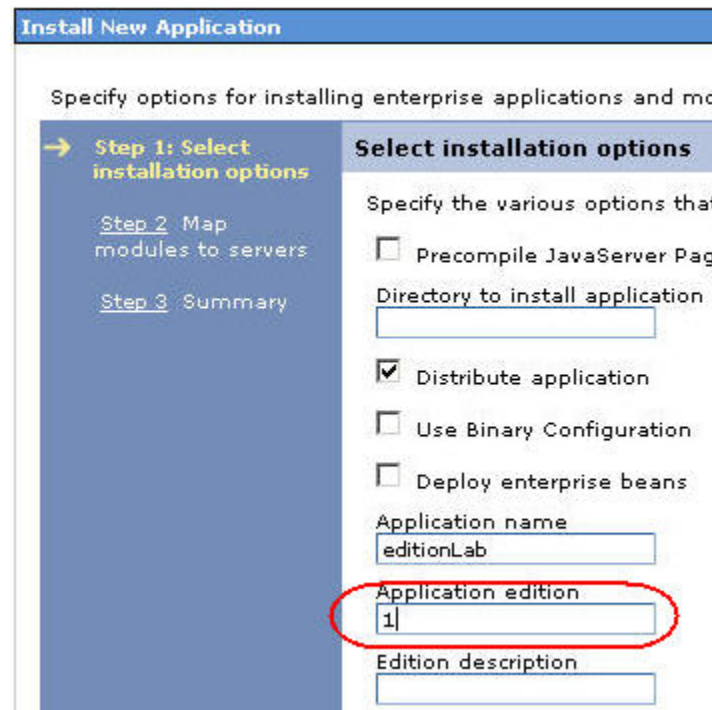
In this section of the lab you will deploy two editions of an application. While driving a constant workload, you will change application edition using two rollout strategies, group and atomic, while maintaining continuous availability.

1. In the WebSphere Console, select **Applications -> New Application**
2. Select **New Enterprise Application**, then **Browse** to C:\Labs\EditionLab and **Open** labAppEditionOne.ear, select **Next**, then **Next**

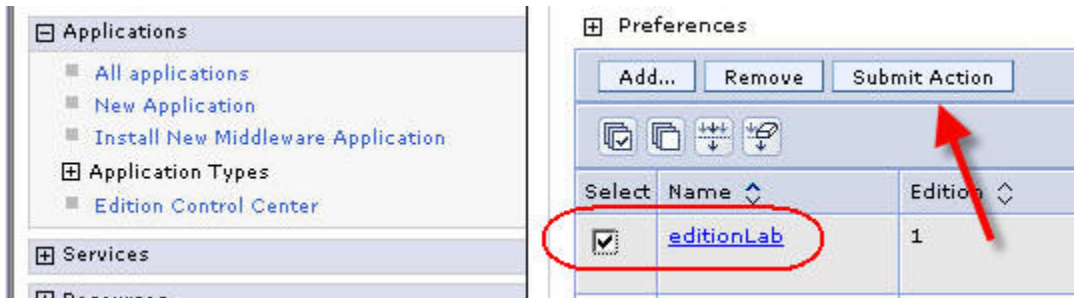




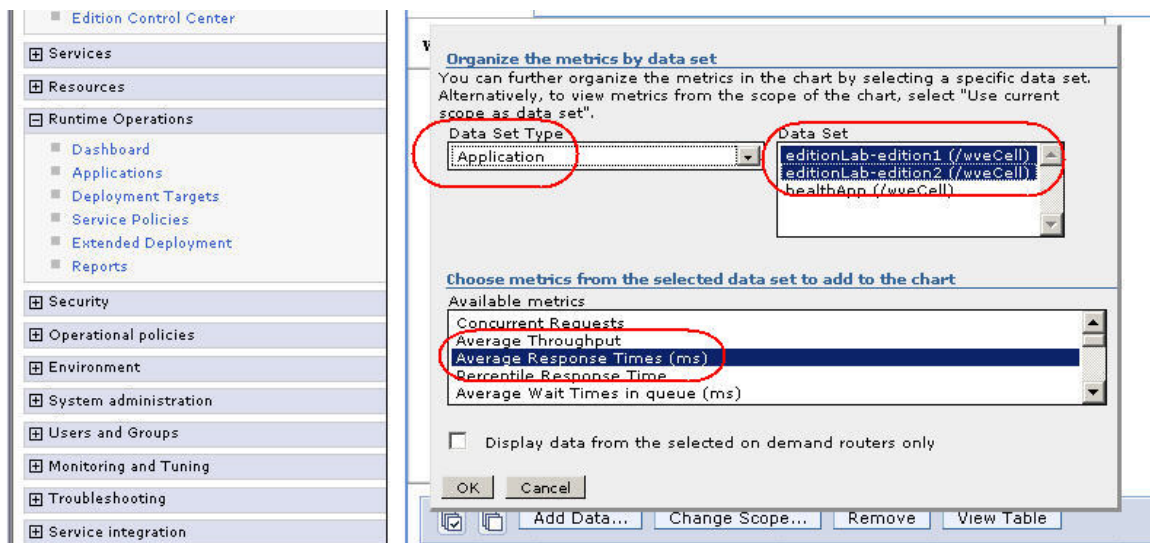
3. Enter '1' for **Application edition**, then select **Next, Next, Finish, Save**



4. Select **Applications -> New Application -> New Enterprise Application**, then **Browse** to C:\Labs\EditionLab and **Open** labAppEditionTwo.ear, select **Next**, then **Next**
5. Enter '2' for **Application edition**, then select **Next, Next, Finish, Save**
6. Start application edition 1. Select **Applications -> All applications**, select the checkbox for edition Lab Edition 1, then select **Submit Action**



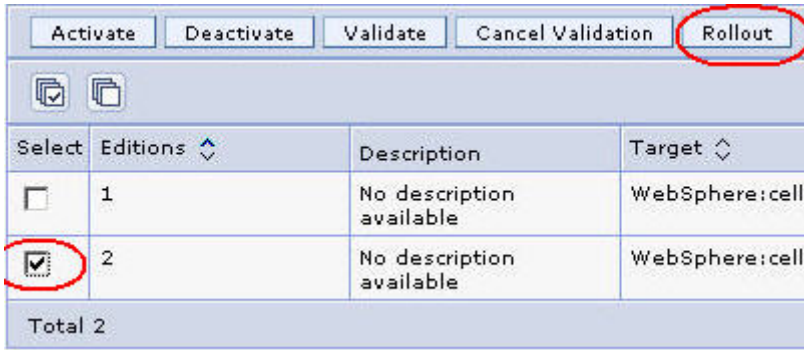
7. Select **Runtime Operations** -> **Reports**, select **Open a New Chart Tab, Add Data...**
8. For **Data Set Type** choose **Application**, choose editionLab-edition 1 and editionLab-edition2 from the **Data Set**, **Average Response Times** from the available metrics, **OK**, then **View Chart in New Window**



9. From the Windows Desktop navigate to the **EditionLab** folder, then double-click **startJMeter**
10. From the JMeter menu bar select **Run -> Start**

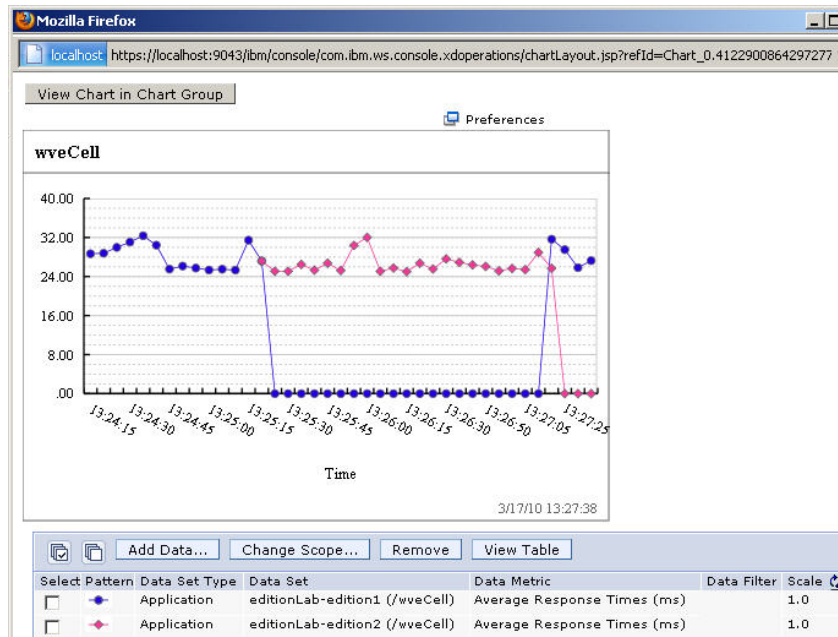
When using the group rollout strategy, the administrator defines the number of servers in the deployment target. That is, how many clustered application servers the application edition manager should update at the same time. Naturally, servers that are changed from one edition to another are offline and not processing application requests during the transition. In this lab there are only 2 servers due to resource constraints.

11. In the WebSphere Console select **Applications** -> **Edition Control Center** -> **editionLab**
12. Select the checkbox for edition 2, then **Rollout, OK**; in this window you can watch the step-by-step progress of the rollout, and in the chart window for a smooth change of application editions as shown in the screenshot shown in step 14



Atomic is a rollout strategy whereby all user requests are served by the same application version. After it is online, the active new edition completely replaces the old edition. For a server cluster, this is done by releasing (rolling out) the new edition to half the cluster at a time. The previous edition is served until the first half of the cluster is available to serve the next edition. At that time, the old edition is taken offline and the remaining half of the cluster is released.

13. In the WebSphere Console select **Applications -> Edition Control Center -> editionLab**
14. Select the checkbox for edition 1, then **Rollout**, change to **Atomic, OK**; watch the two browser windows for a smooth change of application editions as shown in the following screenshot



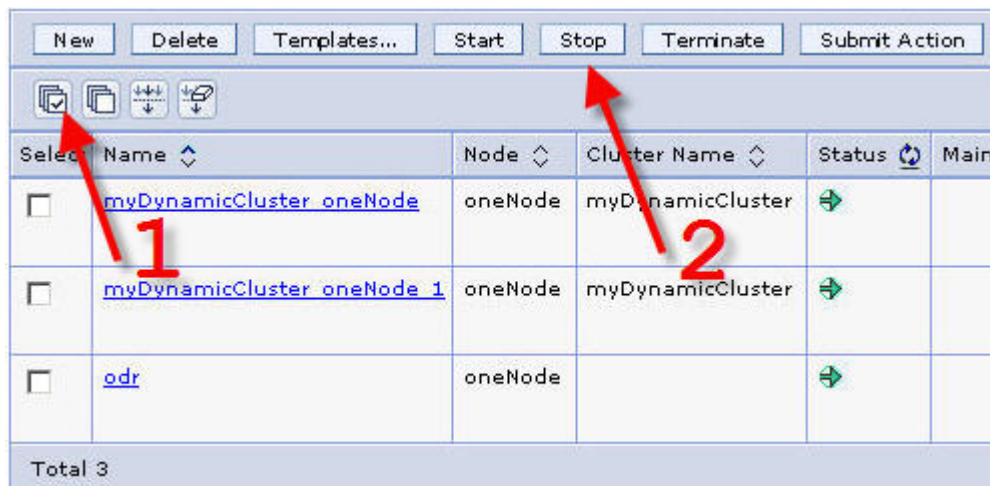
15. When finished, from the JMeter menu bar select **Run -> Stop, File -> Exit**, then close the browser window displaying the chart by selecting the 'X' in the upper right corner

Lab 3 BBSON Lab (optional)

WebSphere Virtual Enterprise offers a bulletin board service overlay network (BBSON) that is independent of the WebSphere Application Server Network Deployment high availability manager. The advantage of enabling BBSON is that it alleviates the dependency on the WebSphere Application Server Network Deployment high availability manager. The other benefit of using BBSON is that it removes the time-intensive requirement of constructing and managing core groups and bridges.

In this section of the lab you will enable BBSON, learning how to determine if it is enabled both from the WebSphere Console and from the WebSphere logs. To effect this change using the resources available, we will shut down components that are not directly involved.

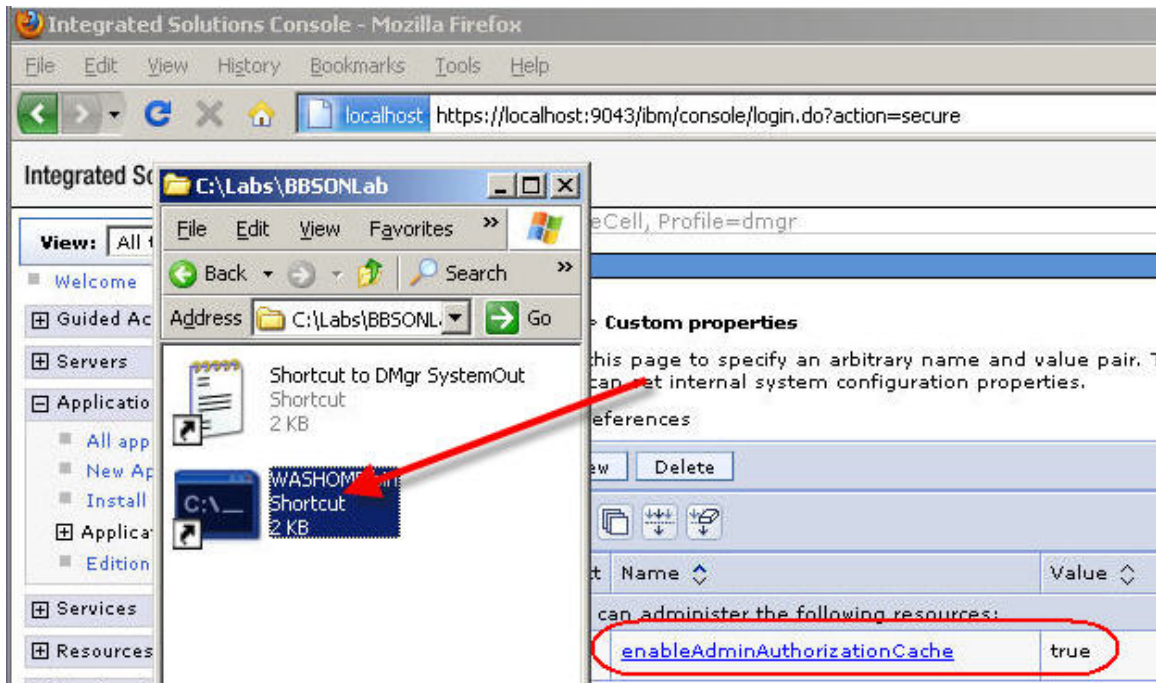
1. In the WebSphere Console navigate to **System administration** -> **Cell**, then under **Additional Properties** select **Custom Properties**: notice that only enableAdminAuthorizationCache is defined
2. Shut down unnecessary components: In the WebSphere Console navigate to **Servers** -> **All servers**, select the checkbox for **All Servers**, then **Stop**



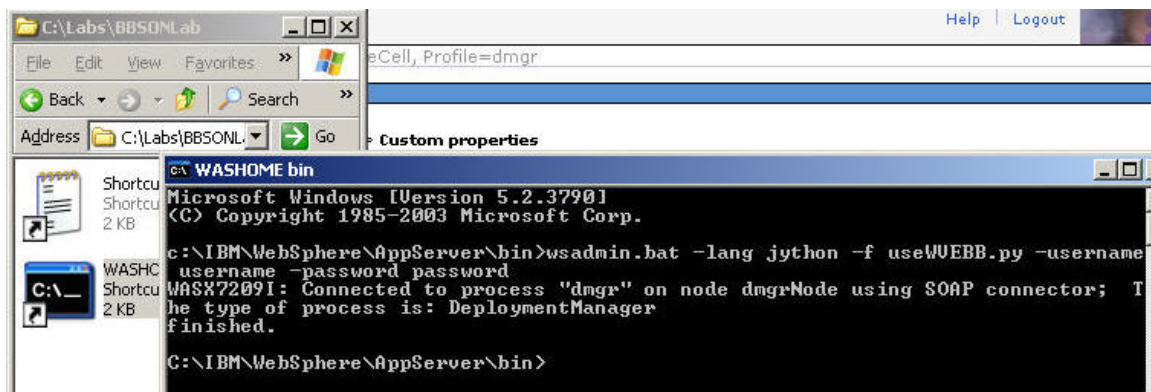
3. Navigate to **System administration** -> **Middleware nodes**, select the checkbox for **oneNode**, select the operational action **Stop agent**, then **Run**



4. From the Windows Desktop navigate to the **BBSONLab** folder, then double-click **WASHOME bin**



- Execute the command enclosed by double quotes on a single line using the values for user and password provided at the start of this lab “wsadmin.bat -lang jython -f useWVEBB.py -username <user> -password <password>” – then leave this DOS window open when complete as we will use it in a following step.

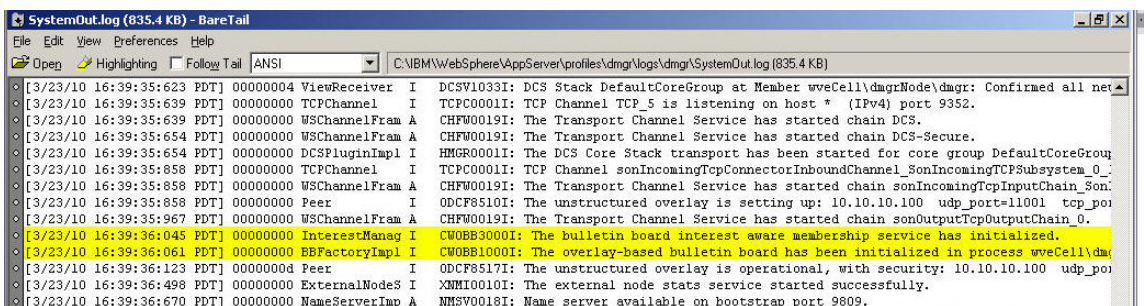
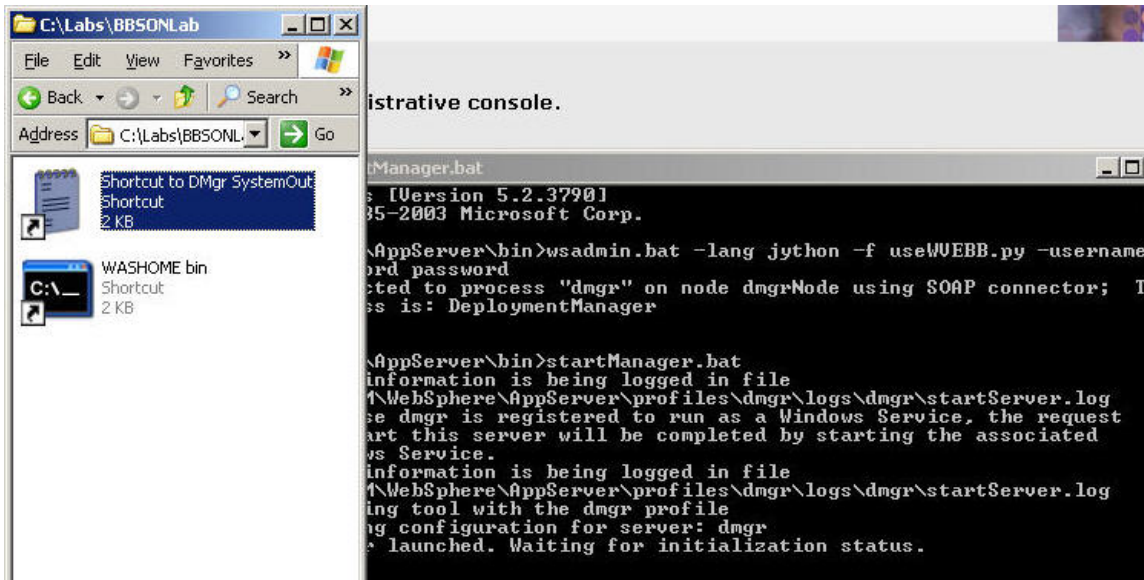


To effect this change the entire cell needs to be restarted. For this lab, only the deployment manager will be restarted to show how to establish if BBSON is active. The following steps demonstrate how to accomplish this from a DOS command window.

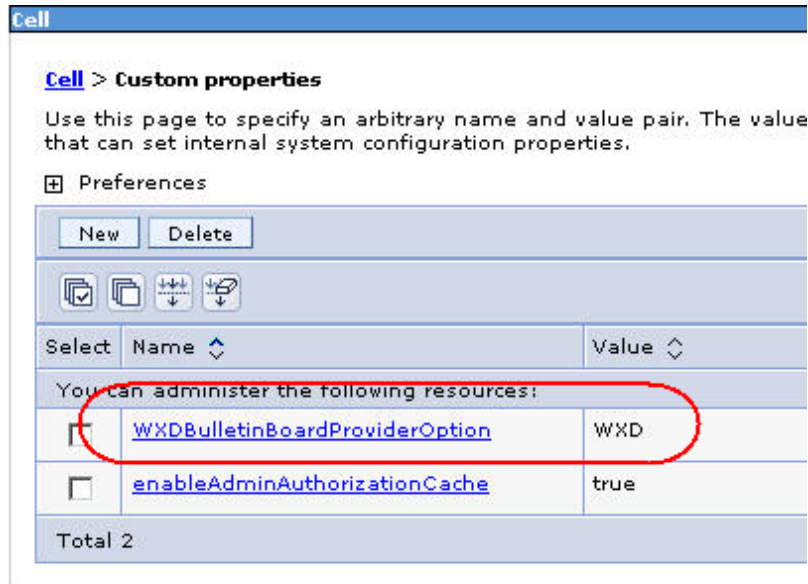
- In the DOS window execute the following command enclosed by double quotes and substituting the user and password provided at the beginning of this lab: “stopManager.bat -username <user> -password <password>” and wait until the deployment manager has stopped.
- In the DOS window execute the command enclosed by double quotes “startManager.bat” and wait until the deployment manager has started.

There are two ways to determine if BBSON has been enabled for the WebSphere Virtual Enterprise cell: first, messages are written to the deployment manager’s stdout log file; second, a new WebSphere cell environment variable is defined.

- From the Windows Desktop navigate to the **BBSONLab** folder, then double-click **Shortcut to DMgr SystemOut** - look for the message prefix **CWOBB** in yellow highlighting the change from using HAM to BBSON as shown in the following screenshots



- Log into the WebSphere Console (if you left the browser open, simply select Home) and navigate to **System administration -> Cell**, then select **Custom Properties**: notice that now **WXDBulletinBoardProviderOption** is defined, indicating that BBSON is enabled



10. When finished, shut down the virtual machine by selecting the red power off icon, then **File -> Exit** from the VMware Workstation menu bar.

Lab summary

- You have seen how to build use WebSphere Virtual Enterprise to change application editions without any loss of service
- You have seen how to build use WebSphere Virtual Enterprise to monitor and manage application health without any loss in service
- You have seen how to configure WebSphere Virtual Enterprise to use either the high availability manager or bulletin board service overlay network for application infrastructure communications