OpenShift Enterprise v3 Workshop

ABOUT THE WORKSHOP ENVIRONMENT

For the workshop, you have your own OpenShift Enterprise 3.1.1 instance running in EC2. The instance is designed to replicate both the server and the client of OpenShift in a single instance.

GETTING STARTED

Your access details and credentials are on the label below. You can access the environment using three different methods:

- VNC – this opens a desktop on the remote server, and provides all client tools and services. VNC is needed for some exercises. If you don't have vncviewer on your laptop, you can use the noVNC applet through your local browser.
- SSH – you can use SSH to run command line tools directly, requiring less bandwidth than VNC.
- Web – you can open your OpenShift web console and hosted applications using your local browser, also requiring less bandwidth than VNC.

- **IMPORTANT:** Please ensure that you can connect properly to the environment before proceeding.

Connection	**Where <IP> is the IP address on the label**
noVNC	Open **https://no-vnc.apps.<IP>.xip.io/** in your preferred browser
Local VNC client	$ **vncviewer <IP>:0**
SSH	$ **ssh demo@<IP>**
Web	Open **https://openshift.<IP>.xip.io:8443/** in your preferred browser

LAB EXERCISES

CONFIGURE SCREEN RESOLUTION

By default, the demo VM has a screen resolution of 1024x768. If your host supports it, you may want to increase this.

If you are connected via VNC, you may be able to increase the screen resolution simply by resizing the VNC window on the host.

If this does not work, as an alternative, try the following steps.

1. Navigate to *Applications* / *System Tools* / *Settings*.
2. Click *Displays*.
3. Click *Unknown Display*.
4. Select a suitable resolution from the *Resolution* drop-down menu.
5. Click *Apply*.
6. Click *Keep Changes*.
7. Close the *Displays* window.

BROWSE THE OPENSHIFT WEB USER INTERFACE

1. Double-click the *Firefox Web Browser* icon on the desktop. The OpenShift web user interface will open on the login page.
2. Enter your username (`demo`) and the password above, and click *Log In*.

LOG IN TO OPENSHIFT VIA THE CLI

1. Double-click the Terminal icon on the desktop.
2. Type `oc login` and press `ENTER` to login to OpenShift.
3. Enter your username (`demo`) and press `ENTER` .
4. Enter the password above and press `ENTER` .
5. Type `exit` and press `ENTER` to close the terminal window.

CREATE AN APPLICATION HOSTED ON OPENSHIFT FROM SOURCE CODE

These steps demonstrate how to import a simple application into OpenShift from a git source tree, and have OpenShift build and deploy the application. The steps demonstrate how easy it is to get started with OpenShift.

1. Click the *New Project* button.
2. Type a name for your project, e.g. `test`, in the *Name* box and click *Create*.
3. Type `nodejs:0.10` in the *Filter by keyword* box.
4. Click the *nodejs:0.10* builder image below.
5. Type a name for your application, e.g. `myapp`, in the *Name* box.
6. Click the *Try it* text to populate the *Git Repository URL* box with the sample repository `git://openshift.example.com/github.com/openshift/nodejs-ex.git`.
7. Click *Create*.
8. Click *Continue to overview*.
9. Wait for the example application to be cloned, built and deployed. It will take up to a minute or two for the build process to begin automatically and then complete.
10. `CTRL`-click the application URL (e.g. *http://myapp-test.apps.example.com*) to launch the application in a new tab.
11. Close the new tab.
12. Click the *OPENSHIFT ENTERPRISE* logo in the top left of the OpenShift web user interface to return to the *Projects* list.
13. Click the *Delete Project* icon next to the *test* project, then click *Delete this project*. **IMPORTANT**: do not delete the *demo*, *prod* or *xpaas* projects.

CREATE AN APPLICATION HOSTED ON OPENSHIFT FROM A PRE-DEFINED TEMPLATE

These steps demonstrate how OpenShift can build and deploy a (possibly multi-component) traditional or modern application from a pre-defined template. This might be done by a developer, tester or operator to quickly set up a dedicated application instance according to its template, using OpenShift's self-service portal.

1. **IMPORTANT**: in the *Projects* list, select the *demo* project. You must use the *demo* project at this point for the rest of the walk-through to proceed!
2. Click the *Add to Project* button.
3. Type `monster` in the *Filter by keyword* box.
4. Under *Instant Apps*, click *monster*.
5. Scroll down and click *Create*.
6. Click *Continue to overview*.
7. Wait for the example application to be cloned, built and deployed. It will take up to a minute or two for the build process to begin automatically and then complete.
8. Optional: to follow the build, click *View Log*. Later, click *Overview* to return to the overview pane. Alternatively use the command `oc logs --follow build/monster-1` in a terminal to follow the build process.
9. When the monster deployment shows as running, `CTRL`-click the application URL (*http://monster-demo.apps.example.com*) to launch the application in a new tab.

COMMIT A SOURCE CODE CHANGE TO AN APPLICATION HOSTED ON OPENSHIFT

These steps demonstrate how a developer could make changes to their application source code and quickly see those changes replicated in their own application instance running on OpenShift.

1. Make a small change to the source code.
   1. Double-click the *Red Hat JBoss Developer Studio 9.0.0.GA* icon on the desktop.
   2. Click the X on the *JBoss Central* tab to close the tab.
   3. Press `CTRL + SHIFT + R` to open the Open Resource window, and type `home.html` to select *monster/src/main/webapp/resources/templates/desktop/home.html*. Click *Open*.
   4. Make a sample change to the text which is visible on the Ticket Monster home page, then click *File/Save*.
2. Commit and push the source code.
   1. Note: `CTRL + #` is an Eclipse shortcut for the following two steps.
   2. Right-click > *monster [monster master]* in the *Project* window and select *Team/Commit...*.
   3. Enter a git commit message (e.g. "minor change"), ensure *src/main/webapp/resources/templates/desktop/home.html* is ticked, then click *Commit and Push*.
   4. A message should be shown including the text "Triggering OSE3 build...". Click *OK*.
3. See the change in the application.
   1. Return to Firefox. OpenShift will clone, built and deploy a new version of the Ticket Monster application. It will be a faster process than the previous build.
   2. `CTRL` -click the application URL (*http://monster-demo.apps.example.com*), and the change you made should soon be visible running on OpenShift.

DEBUG AN APPLICATION HOSTED ON OPENSHIFT

These steps demonstrate how a developer could debug the source code of an application running on OpenShift from their IDE.

1. Set a breakpoint.
   1. Return to JBoss Developer Studio.
   2. Press `CTRL + SHIFT + R` to open the Open Resource window, and type `BookingService.java` to select *monster/src/main/java/org/jboss/examples/ticketmonster/rest/BookingService.java*. Click *Open*.
   3. Press `CTRL + L` , enter `124` and click `OK` or just scroll down to line 124, (`public Response createBook` -`ing…`).
   4. Right-click in the grey bar to the left of the text and select *Toggle Breakpoint* to set a breakpoint on the function.
2. Connect to the application debugger port.
   1. Click the *OpenShift Explorer* tab in the bottom right pane.
   2. Expand the *demo* connection.
   3. The *Sign in to OpenShift* window will appear. Enter your password and click *Finish*.
   4. Expand to find the pod hosting the Ticket Monster application. This will be under *demo / Pods*, will have a name of the form `monster-[1-9]-[a-z0-9]{5}` and will be in state `Running` .
   5. Right-click the pod and select *Port Forwarding...*. The *Port forwarding* window should show a list of ports including *debug* (*8787*). Click *Start All*.
   6. Click *OK*.

   Note: port forwarding can also be achieved using the command `oc port-forward` , e.g. `oc port-forward $POD 8787` .

3. Start debugging.
    1. Return to JBoss Developer Studio.
    2. Click *Run/Debug Configurations...*.
    3. Double-click *Remote Java Application* in the list on the left.
    4. Change the port number on the right from `8000` to `8787` and click *Debug*.
4. Trigger the breakpoint.
    1. Return to Firefox and the Ticket Monster application.
    2. Use the application to buy a ticket.
    3. When you click *Checkout*, the breakpoint will be triggered. You will be taken back to JBoss Developer Studio, and it will ask to open the Debug perspective. Click *Yes*.
5. Examine the state of the application.
    1. In the *Variables* window at the top right of the screen, you will be able to see details of the booking you just entered.
    2. Expand the variable *bookingRequest* and right-click *email*. Select *Change Value...* to change the application state in OpenShift.
    3. Enter a different, but valid, value in the box, and click *OK*.
    4. Click *Run/Resume* to resume flow of the application.
    5. Return to Firefox and click *Bookings*. You should see that you successfully modified the e-mail address of your booking through the debugger.

    Note: there is currently a known bug in Ticket Monster whereby if the REST call triggering the debugger takes too long to return, a second REST call is erroneously issued by the browser. This manifests in the breakpoint being hit twice and two bookings being made. If this occurs, note that this is a bug in Ticket Monster, not in OpenShift! To see the modified e-mail address in Ticket Monster, click the *Bookings* tab.
6. Return to the *JBoss* perspective in JBoss Developer Studio. Find the monster pod in the *OpenShift Explorer* pane, right-click the pod and select *Port Forwarding...*, click *Stop All* and click *OK*.

ROLLBACK AN APPLICATION COMPONENT

These steps demonstrate how a developer could rollback an unwanted change to an application component running on OpenShift.

1. In Firefox, click *Browse*, then *Deployments*, then *monster*.
2. Scroll down, and you should see two deployments: *#2 (latest)* and *#1*. Click *#1*.
3. Click the small *Roll Back* button.
4. Tick the *deployment trigger* option on.
5. Click the small *Roll Back* button again.
6. A message should appear indicating *deployment #3 is rolling back monster to monster-1*.
7. Click *Overview* to return to the overview pane. Deployment #3 of the monster pod should be shown.
8. Navigate to the application front page, and press `CTRL + SHIFT + R` to refresh it. In due course you should find that the change you made earlier has been reverted.

Note: deployment rollbacks can also be achieved using the command `oc rollback`, e.g. `oc rollback monster-1`.

EXPLORE AN APPLICATION POD (LOGGING, METRICS, TERMINAL)

These steps demonstrate how a developer can explore the logs, metrics, filesystem and processes of an application component running inside a container on OpenShift.

1. Click *Browse*, then *Pods*, then click any pod hosting the Ticket monster application. This will be of the form `monster-[1-9]-[a-z0-9]{5}` and will be in state `Running`.
2. Click the *Metrics* tab. This will show CPU and Memory usage information from the pod.
3. Click the *Terminal* tab.
4. Type commands, e.g. `pwd`, `ls`, `id`, `exit`, etc., to explore the container.
5. Click the *Logs* tab. This will show the stdout/stderr of the container.
6. Click the *View Archive* button to navigate to an integrated Kibana instance which securely consolidates and allows searching of logs across multiple pods.
7. Use the browser history or click the home icon to return to OpenShift.

Note: terminal access to a container can also be achieved using the command `oc rsh $POD`.

PROMOTE AND ROLLBACK AN APPLICATION COMPONENT WITHOUT REBUILD

These steps demonstrate how an operator could promote an application component into a subsequent environment (e.g. production) for deployment/upgrade without rebuilding the component, as well as rolling back an unwanted change if necessary.

1. Prepare the *prod* project to host the application component.
    1. In Firefox, select the *prod* project.
    2. Click the *Add to Project* button.
    3. Type `monster-prod` in the *Filter by keyword* box.
    4. Under *Instant Apps*, click *monster-prod*.
    5. Scroll down and click *Create*.
    6. Click *Continue to overview*.
2. Tag the the oldest version of the application component in order to deploy it into the *prod* project.
    1. Double-click the Terminal icon on the desktop.
    2. Type `oc describe imagestream monster` and press `ENTER`.
    3. Choose the oldest of the built docker images (it'll look like *monster@sha256:[0-9a-f]{64}*). Call this $TAG. Ignore the leading `172.30.x.y:5000/demo/` text.
    4. Type `oc tag $TAG monster:prod` and press `ENTER`.
    5. Return to Firefox and wait for OpenShift to deploy the application into the prod project. Note that the application manifest has been set to deploy three application pods: after successfully deploying the first pod, OpenShift will scale it up to three pods.
    6. `CTRL`-click the application URL (*http://monster-prod.apps.example.com*) to launch the application in a new tab.
3. Repeat the process with a newer version of the application component. Watch the Firefox window to see OpenShift automatically carrying out rolling upgrade of the application component in the prod project.
4. Return to the terminal window and type `exit` and press `ENTER` to close the window.

SCALE AN APPLICATION COMPONENT

These steps demonstrate how an operator could manually scale up or down the number of instances of an application component running on OpenShift.

1. Click the *Scale up* (up) and *Scale down* (down) arrows in the *monster* deployment box. You should see the number of pods change.

Note: pod scaling can also be achieved using the command `oc scale`, e.g. `oc scale --replicas=2 dc monster`.

RESET THE DEMO AND PROD PROJECTS

1. Double-click the Terminal icon on the desktop.
2. Type `oc delete all --all -n prod` and press `ENTER` to reset the prod project.
3. Type `oc delete all --all -n demo` and press `ENTER` to reset the demo project.
4. Type `exit` and press `ENTER` to close the terminal window.

“XPAAS” CAPABILITIES

Further optional exercises which demonstrate OpenShift's “xPaaS” capabilities are available by double-clicking the “About this demo” on the VM desktop.

USE THE OPENSHIFT ("OC") COMMAND LINE TOOL AS THE SYSTEM:ADMIN USER

1. Double-click the Terminal icon on the desktop.
2. Type `sudo -i` and press `ENTER` to change to the root user.
3. Type oc commands such as `oc status` and `oc get pods --all-namespaces` as appropriate.
4. **IMPORTANT**: take care not to inadvertently delete objects in the *default*, *logging* or *openshift-infra* projects!
5. Type `logout` and press `ENTER` to change back to the demo user.
6. Type `exit` and press `ENTER` to close the terminal window.