Contents

Scenario Index 1

Monoliths to Microservices: App Transformation Hands-on Workshop 1

Intro 1

Prerequisites 2

Source code for exercises 2

References to `/root`, ~ and `$HOME` 2

Additional OpenShift Configuration 2

Scenario Index

- Scenario 1 - Moving existing apps to the cloud
- Scenario 2 - A Developer Introduction to OpenShift
- Scenario 3 - Transforming an existing monolith (Part 1)
- Scenario 4 - Transforming an existing monolith (Part 2)
- Scenario 5 - Building Reactive Microservices
- Scenario 6 - Prevent and detect issues in a distributed system

Monoliths to Microservices: App Transformation Hands-on Workshop

These documents contains a complete set of instructions for running the workshop, split into different *scenarios*. You can use these documents as a companion as you progress through the scenarios, but keep in mind that some of the links in this document may not work as they will be specific to your online environment. You will be expected to substitute your own values for the following URLs:

- **$OPENSHIFT_MASTER** - When you see this variable, replace it with the value of your own OpenShift master hostname/port, such as `https://master.openshift.com:8443` (be sure to include the port!).
- **$ROUTE_{S}UFFIX * * -Whenyouseethisvariable, replaceitwiththevalueofyourownOpenShiftdefaultroutingsuffi$
  $//coolstore-dev.$ROUTE_SUFFIX`then you would replace this with`http://coolstore-dev.apps.mycompany.comas
  `your OpenShift cluster uses a default routing suffix of`apps.mycompany.com‘.

Intro

As modern application requirements become more complex, it’s apparent that one runtime, one framework, or one architectural style is no longer a feasible strategy. Organizations must figure out how to manage the complexity of distributed app development with diverse technologies, a lack of skilled resources, and siloed processes.

In this hands-on workshop you’ll learn about:

- Migrating an existing legacy Java™ EE app to Red Hat JBoss Enterprise Application Platform on OpenShift.
- Using modern frameworks like Spring Boot, Wildfly Swarm, Eclipse Vert.x, and Node.js to implement microservices and replace monolithic functionality.
- Developing and deploying using Red Hat OpenShift Container Platform, Red Hat OpenShift Application Runtimes, and DevOps processes.
- The benefits and challenges with microservices, including use cases for reactive microservices.
- Preventing and detecting issues in a distributed system.
- API gateways and microservices.

Prerequisites

You can complete this workshop using either a Linux-based system such as RHEL or CentOS, or Windows 7 or later. Windows users should be aware of the Prequisites and notes for Windows Users to understand how to execute commands and refer to environment variables.

To complete this exercise using your own environment, you'll need to install the following software:

- Red Hat Container Development Kit - this includes OpenShift and the `oc` command line utility. Follow the instructions for installing it and be sure you have access to the `oc` command and can login (e.g. `oc login -u developer -p developer`). You can also use other versions of OpenShift version 3.7 or later, including OpenShift Origin.

  NOTE: When using your own OpenShift cluster, you must run additional setup outlined in the **Additional OpenShift Configuration** section.

- JBoss EAP 7.1.0 GA - save the `.zip` file to your `${HOME}` directory (it will be used in the first scenario)
- OpenJDK 1.8 or later with its `bin` directory on your `$PATH`
  - Run `java -version` and it should be `1.8.x` or later
- Maven 3.3.9 or later
  - Run `mvn --version` and it should be `3.3.9` or later
- Git 2.7.2 or later
  - Run `git --version` and it should be `2.7.2` or later
- Various common Linux developer utilities (`unzip`, `curl`, `tree`, etc)
- A text editor or IDE for editing code
- Red Hat Application Migration Toolkit 4.0.0.Beta4. Download and unzip into the `${HOME}/rhamt-cli-4.0.0.Beta4` directory (you can install it elsewhere, but you'll need to specify the path when running the tool during the first lab).
- A copy of the source code for exercises at `${HOME}/projects` (see below)
  - You can open the project(s) in your favorite IDE for editing as needed.

Source code for exercises

The source code required for the exercises in this guide is in a *Git* repository on `github.com`. To use this source code, you must *clone* the repository into your `$HOME/projects` directory.

For example, the following command will clone the source code into the appropriate directory for use with the guide:

```
git clone https://github.com/RedHat-Middleware-Workshops/modernize-apps-labs $HOME/projects
```

There is also a *solution* branch in this repository, which refers to the source code of the solution for each exercise. To access the solution, you can use `git checkout solution`. Be sure to switch back to the `master` branch when done viewing the solution using `git checkout master`.

References to `/root`, ~ and `$HOME`

In various places throughout the exercises, commands are run using paths that refer to `/root` and ~ and `$HOME` - this directory is the user's home directory and will be different, depending on your OS and referred to as `%USER_HOME%` on Windows. You'll need to take care to replace the paths appropriately if you've installed the tools and source code in somewhere other than `${HOME}` and be sure to replace `/root/`, `$HOME` or ~ with `%USER_HOME%` as needed on Windows.

Additional OpenShift Configuration

This lab makes use of OpenShift features and Linux container images available on the Red Hat Container Catalog and Docker Hub. To install and configure OpenShift to use them, run the following commands:

```
oc login -u system:admin # login as cluster admin

# Add admin privileges for admin and developer
oc adm policy add-role-to-user system:image-puller system:anonymous
oc adm policy add-cluster-role-to-user cluster-admin admin
oc adm policy add-cluster-role-to-user sudoer developer
```

```
# Import jenkins images and re-tag for 3.7
oc import-image jenkins:v3.7 --from='registry.access.redhat.com/openshift3/jenkins-2-rhel7:v3.7' --c
oc export template jenkins-persistent -n openshift -o json | sed 's/jenkins:latest/jenkins:v3.7/g' |
oc export template jenkins-ephemeral -n openshift -o json | sed 's/jenkins:latest/jenkins:v3.7/g' |

# import Monolith templates and JBoss Imagestreams
oc create -n openshift -f https://raw.githubusercontent.com/openshift/openshift-ansible/release-3.9/
oc create -n openshift -f https://raw.githubusercontent.com/RedHat-Middleware-Workshops/modernize-ap
oc create -n openshift -f https://raw.githubusercontent.com/RedHat-Middleware-Workshops/modernize-ap

# Disable namespace ownership for router
oc env dc/router ROUTER_DISABLE_NAMESPACE_OWNERSHIP_CHECK=true -n default

echo "Importing images"
for is in {"registry.access.redhat.com/jboss-eap-7/eap70-openshift","registry.access.redhat.com/rhsc
do
  oc import-image $is --all --confirm --as=system:admin
done
```