

Lab 2

Design-Time Governance

Lab Steps

- Step 1 : Review default workflow in JBDS
- Step 2 : Deploy default workflow
- Step 3 : Review DTGov UI
- Step 4 : View DTGov configuration
- Step 5 : Distribution management for application
- Step 6 : Create link for production
- Step 7 : Deploy application to S-RAMP
- Step 8 : Walk through workflow

Step 0

Start the FSW Server

Goals

- *Start an instance of Fuse Service Works*

FSW Server

TODO

1. Open a Terminal window
2. Navigate to the Server directory and start the server:

```
cd Servers/jboss-eap-6.1  
bin/standalone.sh
```

A terminal window titled 'lab10@summitlab:~/Servers/jboss-eap-6.1' showing the execution of 'bin/standalone.sh'. The output displays the JBoss Bootstrap Environment with variables like JBOSS_HOME and JAVA_OPTS. It then shows a series of INFO logs from the JBoss Modules, MSC, and AS components, indicating the startup of JBoss Red Hat Fuse Service Works and the deployment of several WAR files (switchyard-bpel-console-server, overlord-rtgov, overlord-rtgov-epn, and gadgets).

```
lab10@summitlab:~/Servers/jboss-eap-6.1  
File Edit View Search Terminal Help  
[lab10@summitlab ~]$ cd Servers/jboss-eap-6.1/  
[lab10@summitlab jboss-eap-6.1]$ bin/standalone.sh  
  
=====
```

JBoss Bootstrap Environment

JBOSS_HOME: /home/lab10/Servers/jboss-eap-6.1

JAVA: /home/lab10/JDKs/jdk1.7.0_51/bin/java

JAVA_OPTS: -server -XX:+UseCompressedOops -Xms1303m -Xmx1303m -XX:MaxPermSize=256m -Djava.net.preferIPv4Stack=true -Djboss.modules.system.pkgs=org.jboss.byteman -Djava.awt.headless=true

=====

```
18:35:36,459 INFO [org.jboss.modules] (main) JBoss Modules version 1.2.2.Final-redhat-1  
18:35:36,833 INFO [org.jboss.msc] (main) JBoss MSC version 1.0.4.GA-redhat-1  
18:35:36,882 INFO [org.jboss.as] (MSC service thread 1-6) JBAS015899: JBoss Red Hat JBoss Fuse Servi  
ce Works 6.0.0.GA-redhat-2 (AS 7.2.1.Final-redhat-10) starting  
18:35:40,717 INFO [org.switchyard] (ServerService Thread Pool -- 3) SwitchYard version 1.1.1-p5-redh  
at-1  
18:35:41,612 INFO [org.jboss.as.server.deployment.scanner] (DeploymentScanner-threads - 1) JBAS01500  
3: Found switchyard-bpel-console-server.war in deployment directory. To trigger deployment create a f  
ile called switchyard-bpel-console-server.war.dodeploy  
18:35:41,612 INFO [org.jboss.as.server.deployment.scanner] (DeploymentScanner-threads - 1) JBAS01500  
3: Found overlord-rtgov.war in deployment directory. To trigger deployment create a file called overl  
ord-rtgov.war.dodeploy  
18:35:41,613 INFO [org.jboss.as.server.deployment.scanner] (DeploymentScanner-threads - 1) JBAS01500  
3: Found overlord-rtgov-epn.war in deployment directory. To trigger deployment create a file called o  
verlord-rtgov-epn.war.dodeploy  
18:35:41,613 INFO [org.jboss.as.server.deployment.scanner] (DeploymentScanner-threads - 1) JBAS01500  
3: Found gadgets.war in deployment directory. To trigger deployment create a file called gadgets.war.
```

Step 1

Review DTGov Workflow

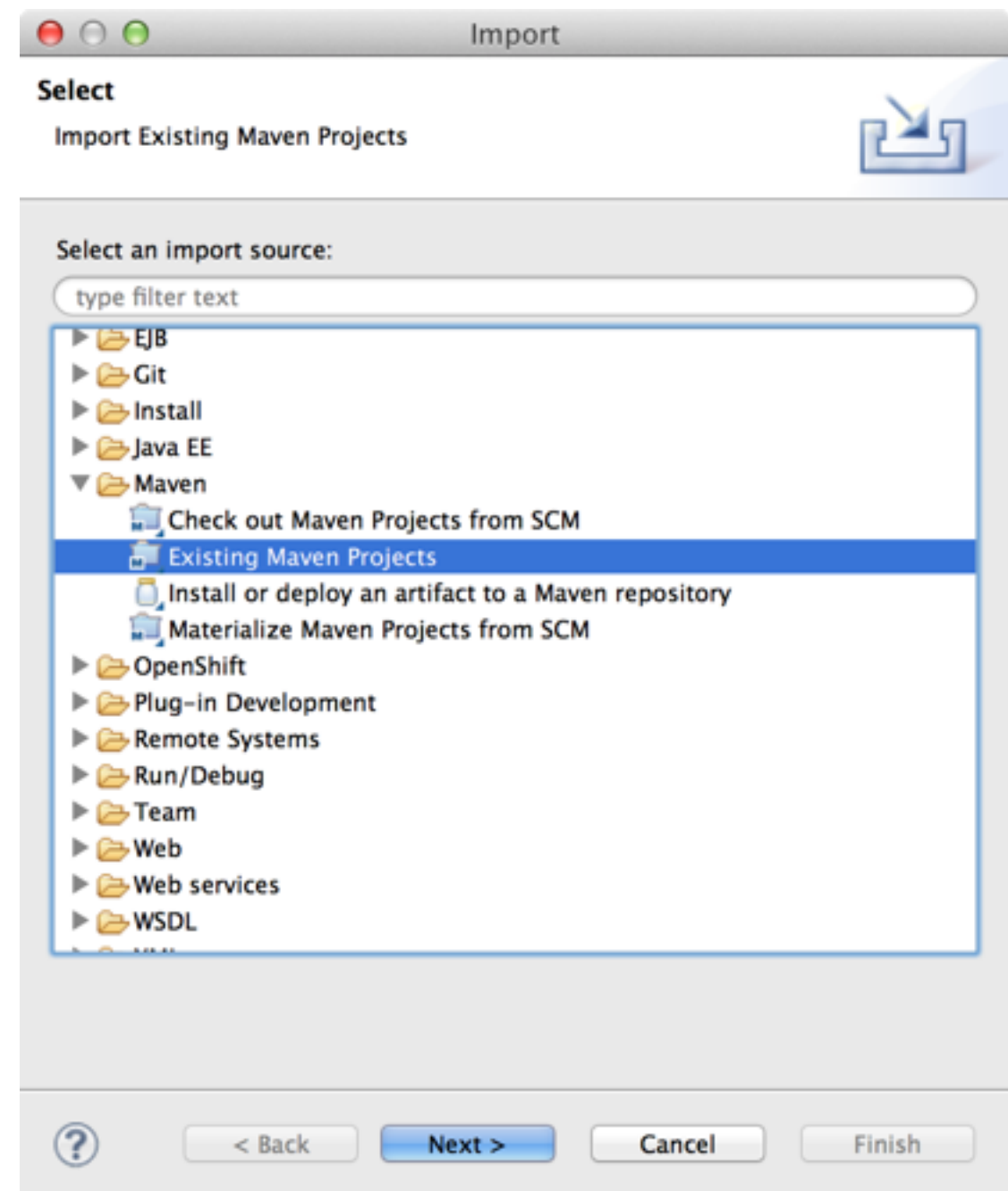
Goals

- *Import the default workflow project*
- *Review the SimpleReleaseProcess Workflow*
- *Review the Roles for the Process Task*

Importing the Workflow

TODO

1. File -> Import ... from the JBDS menu.
- Select Maven -> Existing Maven Projects
 - Click Next



Importing the Workflow

TODO

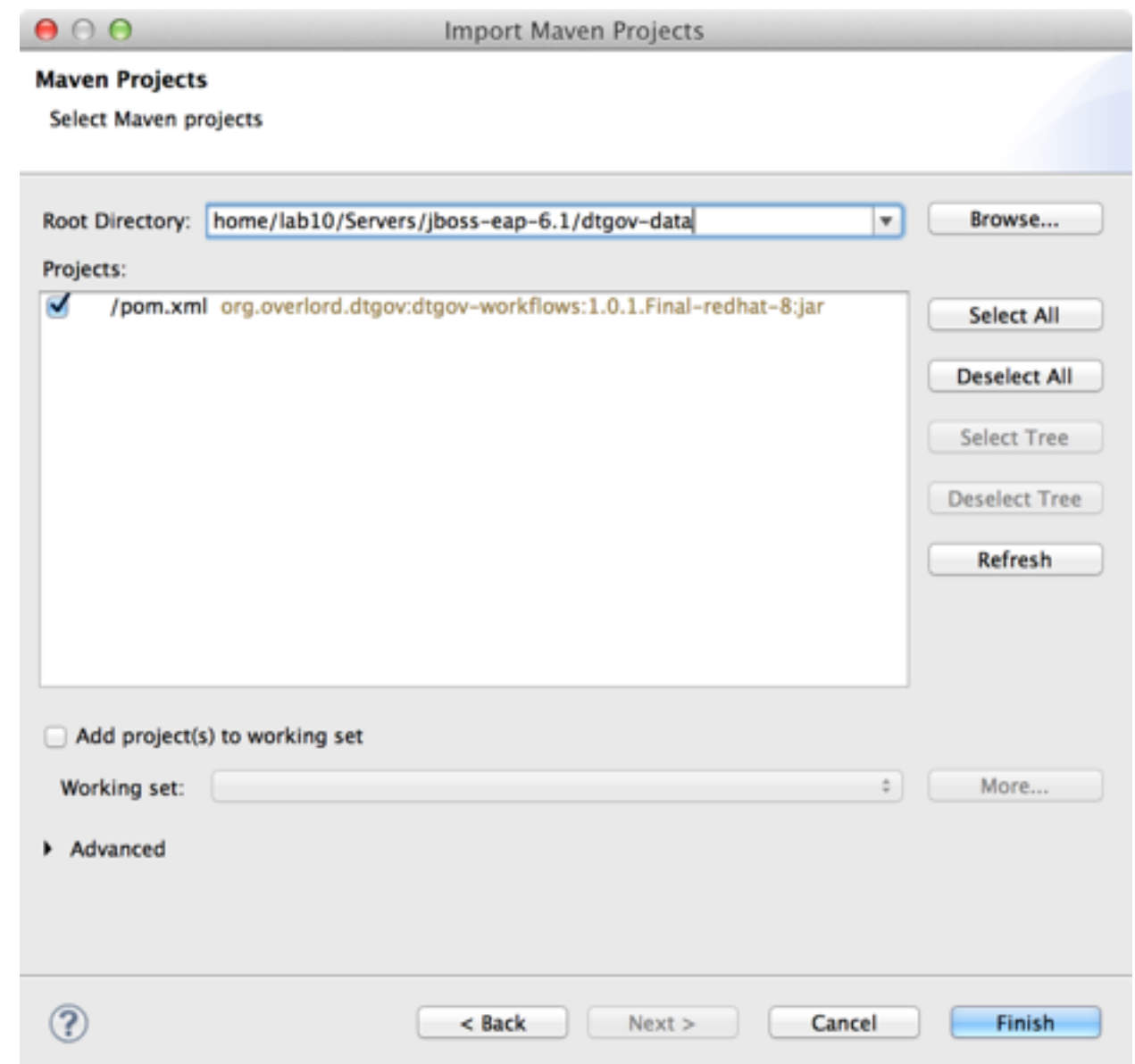
1. Click Browse ... and navigate to the location of lab1. For example:

`/home/lab10/Servers/jboss-eap-6.1/dtgov-data`

2. Make sure the pom.xml is checked for:

`org.overlord.dtgov:dtgov-workflows`

- Click Finish



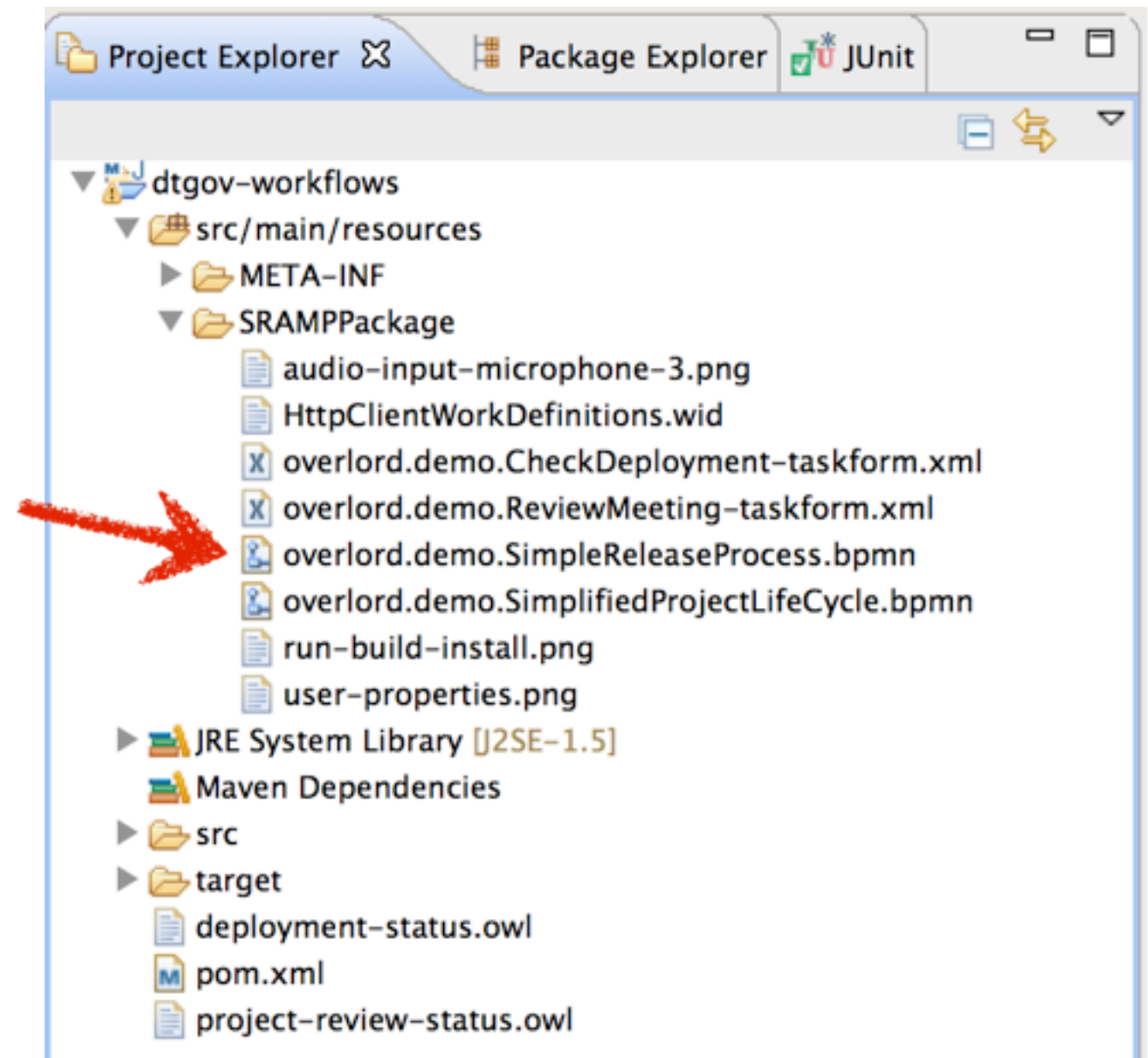
Default Release Workflow

FYI

- Workflow can be customized or new ones created
- BPMN2 based governance workflows
- Out of the box integration with jBPM-6

TODO

1. Double-click on the SimpleReleaseProcess BPMN 2 definition to open up the default workflow.



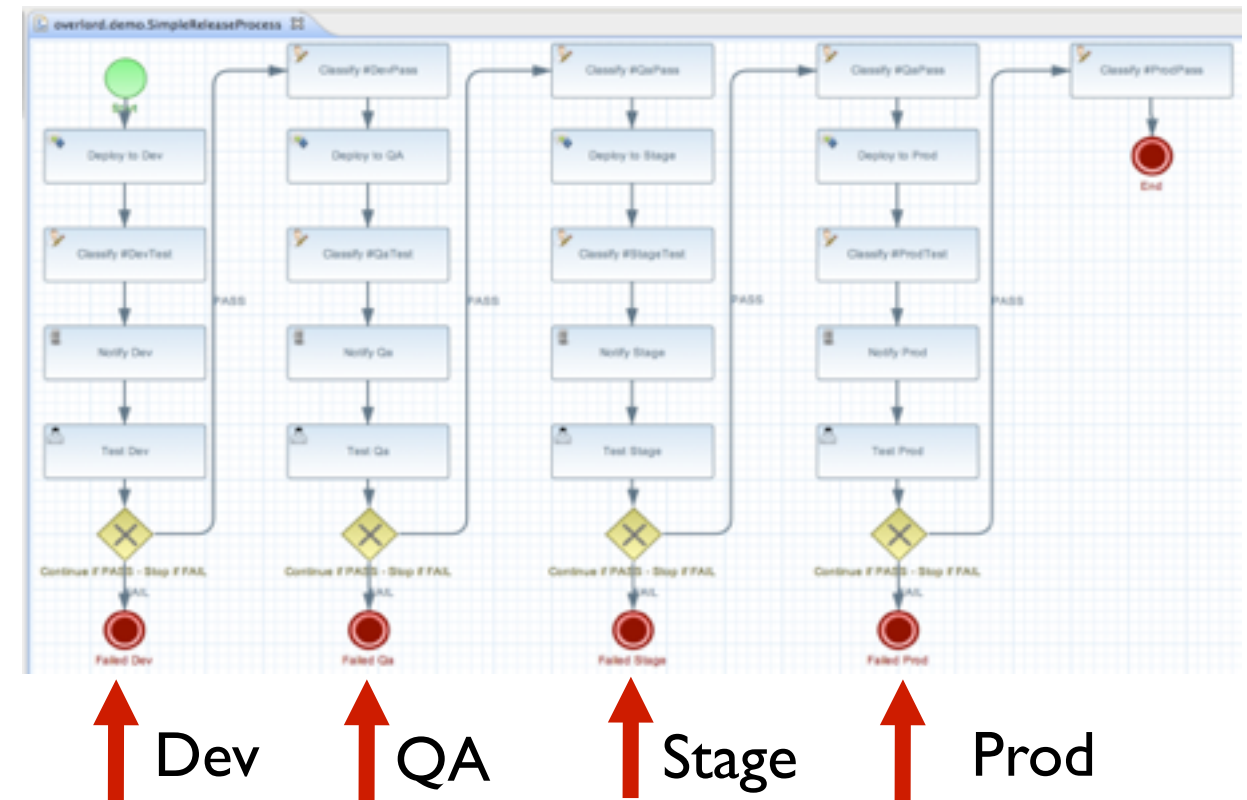
SimpleReleaseProcess

FYI

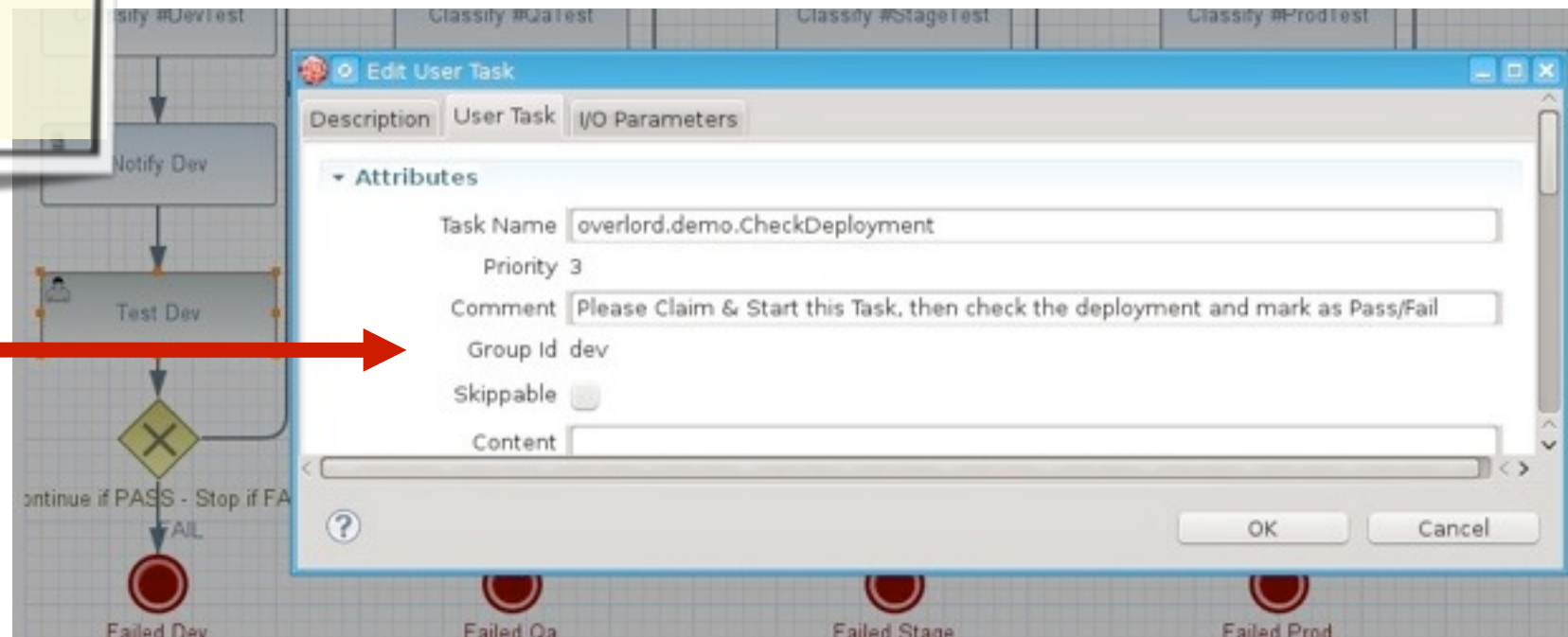
The SimpleReleaseProcess provides the capability to manage the lifecycle through the workflow tasks:

- Deploy to the X Environment
- Classify the artifact with the environment classification
- Notify the X group by sending out an email with a link to the X taskform
- Test X User task to claim, start and pass or fail the deployment

The Group Id determines the Role associated with the user task.



Role



Step 2

Deploy Default Workflow

Goals

- *Configure JBDS to deploy workflow*
- *Sign onto S-RAMP*
- *Check for deployment of workflow*

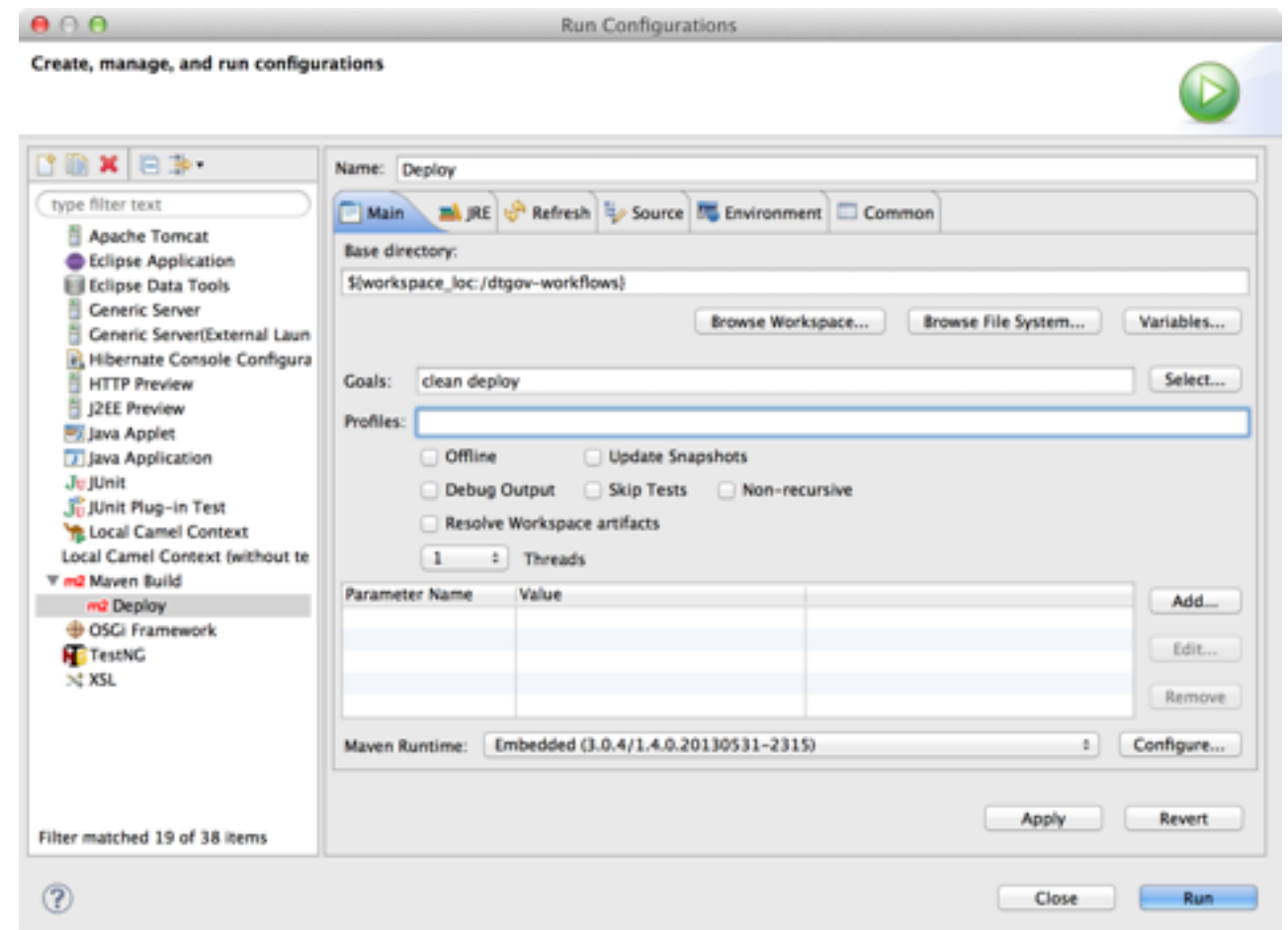
Configure Maven Deploy

FYI

The run configuration will do a maven clean and deploy the KieJarArchive jar to the FSW server. The deployment can also be done at the command line.

TODO

1. Right-click on the dtgov-workflows project in the Project Explorer
2. Select Run As -> Run Configurations
3. Right-click on Maven Build and select New
4. Enter the following values for the build:
Name : Deploy Workflow
Base directory : `${workspace_loc:/dtgov-workflows}`
Goals : clean deploy
5. Click Run



S-RAMP Console

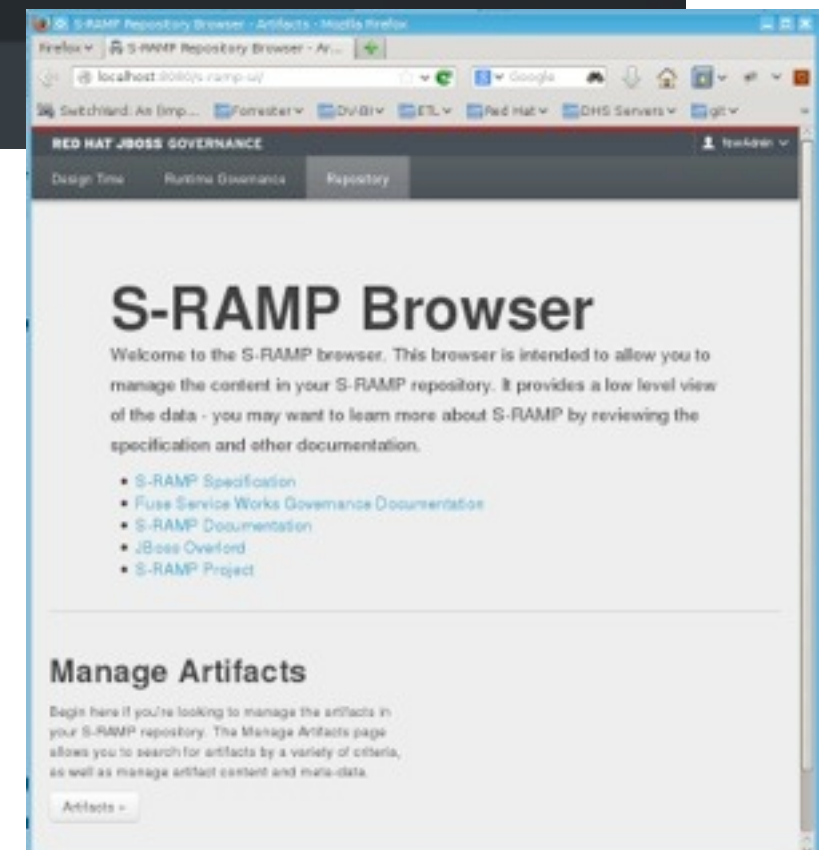
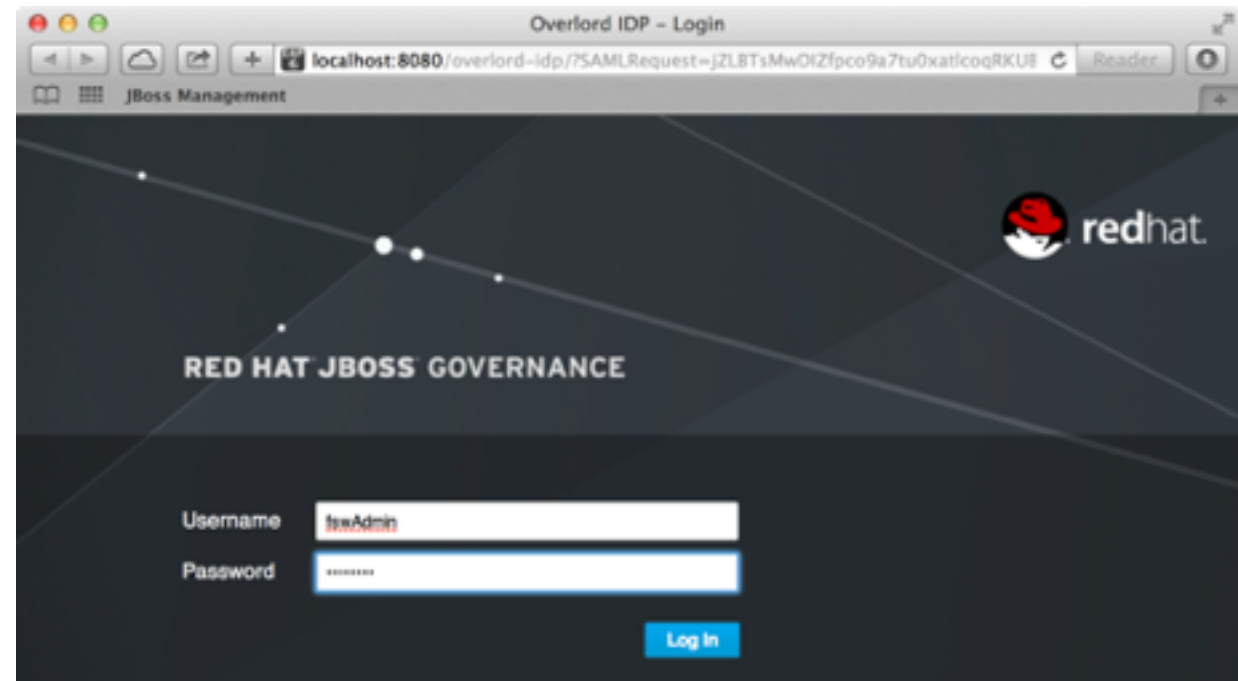
FYI

The Governance Console contains DTGov, RTGov and S-RAMP:

- <http://localhost:8080/s-ramp-ui>
- <http://localhost:8080/dtgov-ui>
- <http://localhost:8080/gadget-web>

TODO

1. Browse to <http://localhost:8080/s-ramp-ui>
2. Enter the username and password for Governance that was entered on installation
3. Click on the Manage Artifacts button



S-RAMP Console

FYI

The dtgov-workflows-1.0.1.Final-redhat-8.jar will contain the KieJarArchive derived type. Multiple workflows can be deployed and referenced by DTGov.

TODO

1. Find the default workflow in the artifact list

The screenshot shows the S-RAMP Console interface. The top navigation bar includes 'Design Time', 'Runtime Governance', and 'Repository'. The 'Repository' tab is active. Below the navigation bar, the breadcrumb 'S-RAMP Dashboard / Artifacts' is visible. On the left, the 'Core Properties' section contains filters for 'Type', 'Date Created', 'Date Last Modified', 'Created By', 'Last Modified By', and 'Origin'. A red arrow points from the 'Date Created' filter to the 'dtgov-workflows-1.0.1.Final-redhat-8.jar' entry in the artifact list. The main area displays a table of artifacts with columns: Name, Type, Derived, and Last Modified. The table shows 7 artifacts, with the first one being the target of the red arrow.

Name	Type	Derived	Last Modified
dtgov-workflows-1.0.1.Final-redhat-8.jar	KieJarArchive		03/27/2014
dtgov-workflows-1.0.1.Final-redhat-8.pom	MavenPom		03/27/2014
kmodule.xml	KieXmlDocument		03/27/2014
overlord.demo.CheckDeployment-taskform.xml	XmlDocument		03/27/2014
overlord.demo.ReviewMeeting-taskform.xml	XmlDocument		03/27/2014
overlord.demo.SimpleReleaseProcess.bpmn	BpmnDocument		03/27/2014
overlord.demo.SimplifiedProjectLifeCycle.bpmn	BpmnDocument		03/27/2014

Step 3

Review DTGov UI

Goals

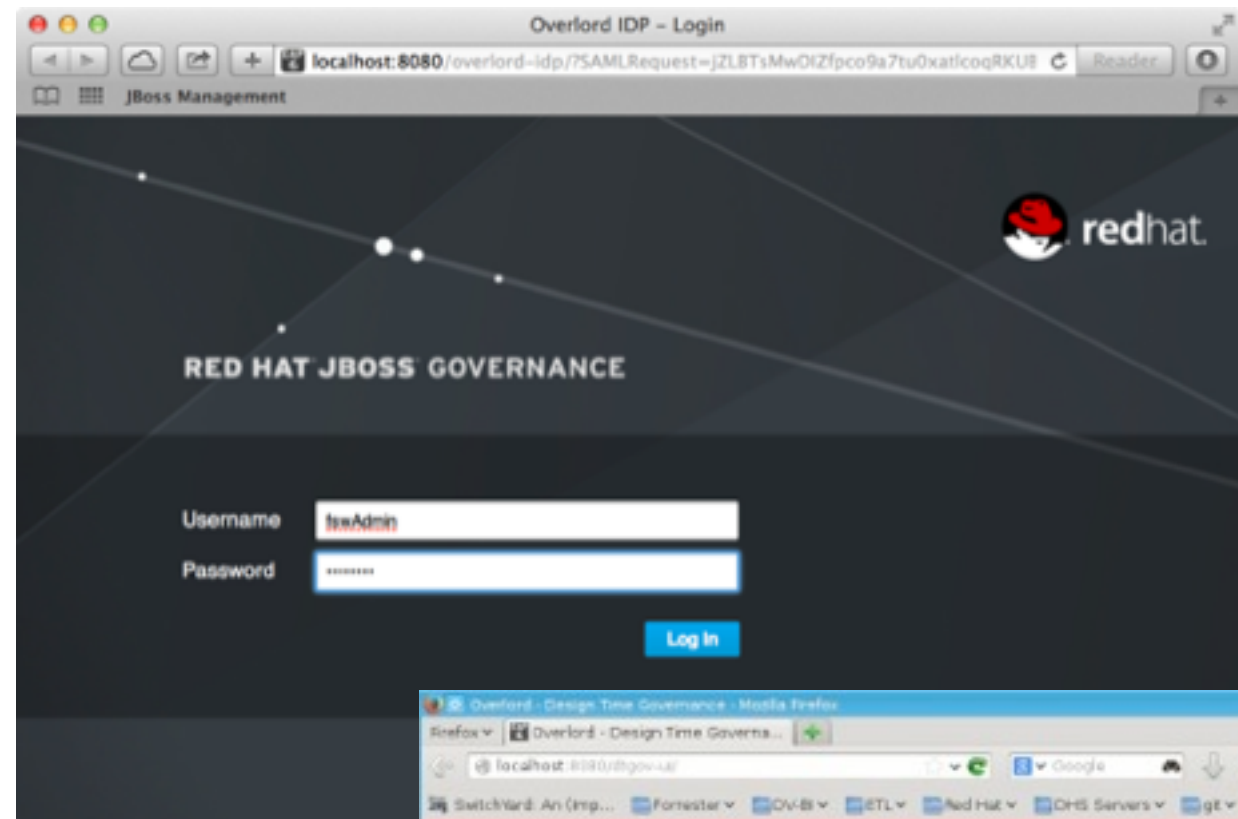
- *Explore the DTGov console*

DTGov Console

FYI

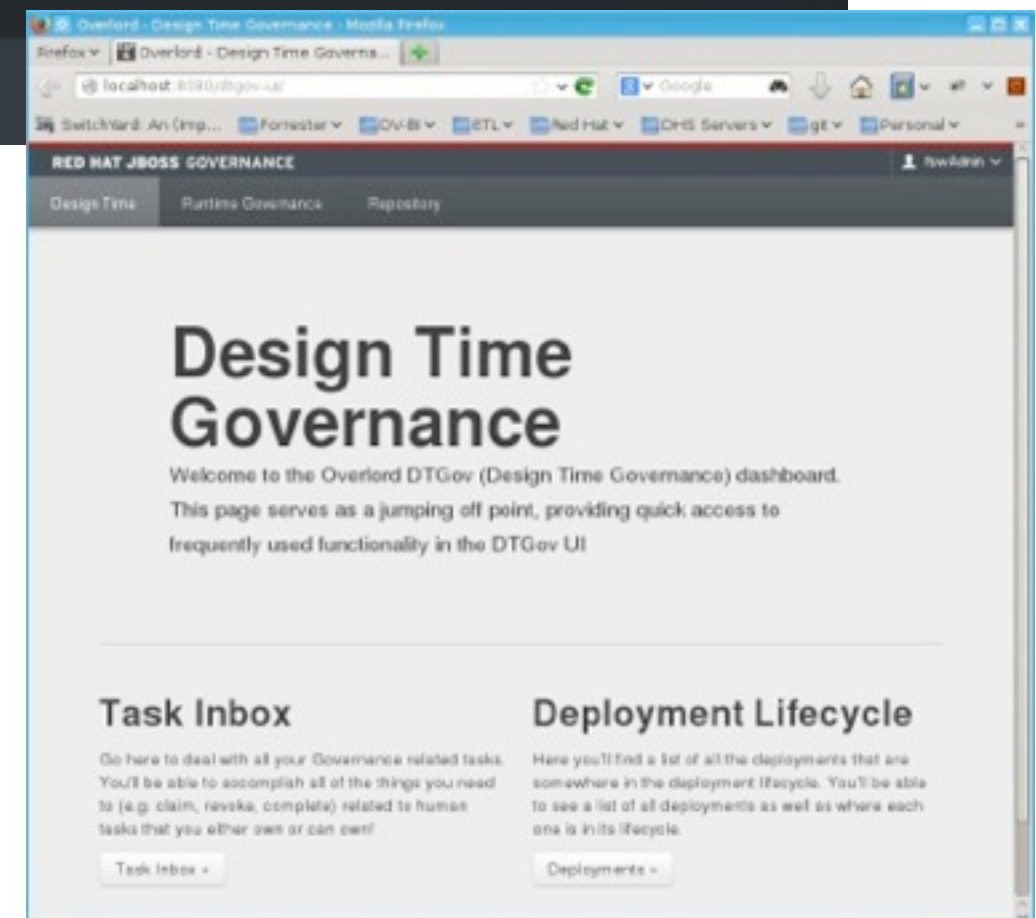
The Governance Console contains DTGov, RTGov and S-RAMP:

- <http://localhost:8080/s-ramp-ui>
- <http://localhost:8080/dtgov-ui>
- <http://localhost:8080/gadget-web>



TODO

1. Browse to <http://localhost:8080/dtgov-ui>
2. Login with credentials:
username : fswAdmin
password : redhat1!



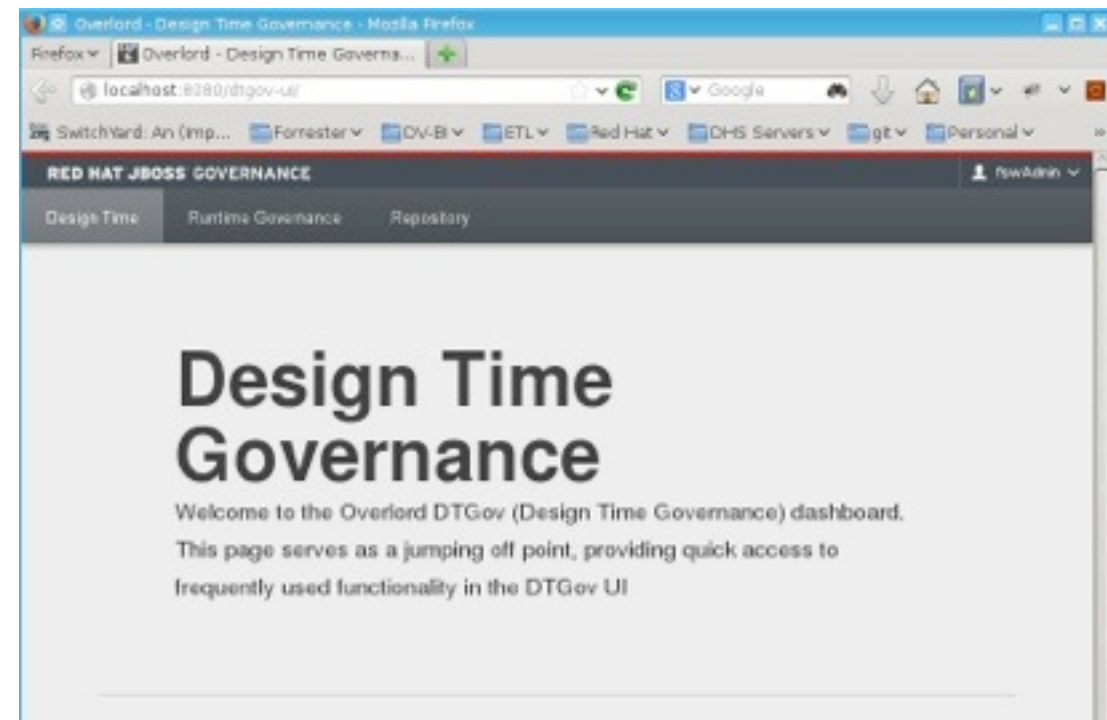
DTGov Task Inbox

FYI

The Task inbox gives the user their Governance related tasks. This allows the user to claim, revoke, complete, etc related to human tasks they own or can own.

TODO

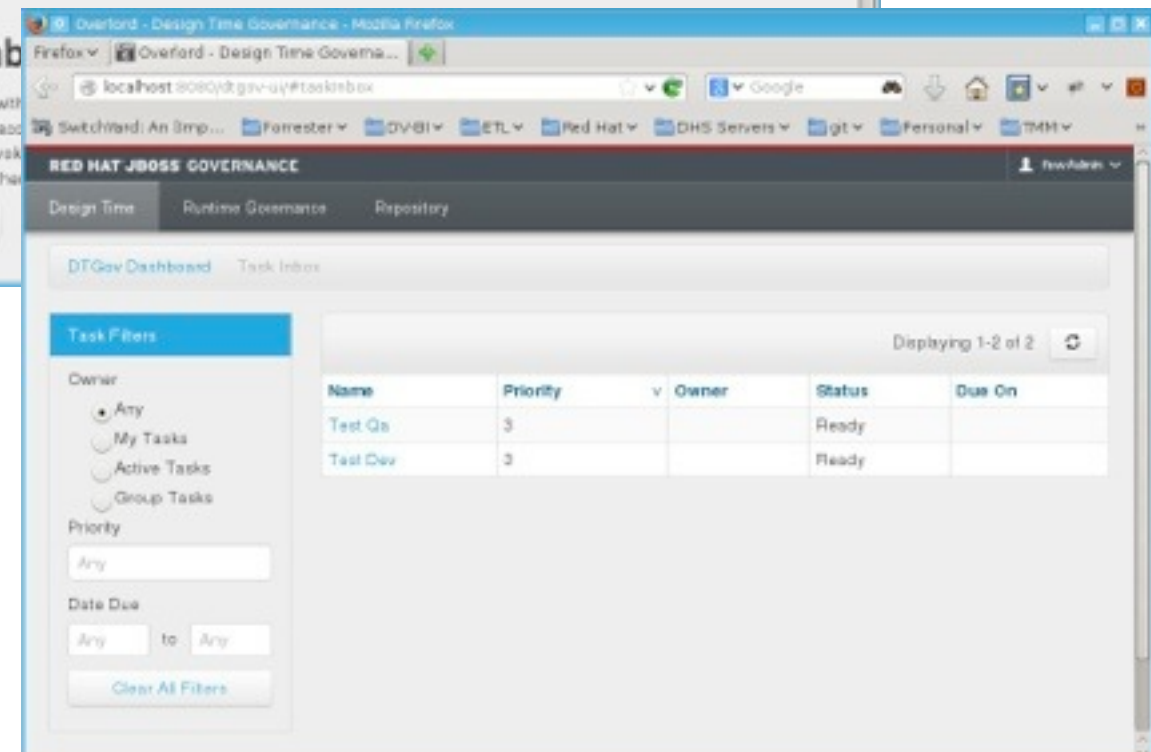
- 1. Click on the Task inbox button*
- 2. Your task list will be empty until a later step but this screen shot shows what the list will look like*



Task Inbox

Go here to deal with
You'll be able to add
to (e.g. claim, revoke
tasks that you either

[Task Inbox >](#)



DTGov Deployment Lifecycle

FYI

The Deployment Lifecycle list all the deployments and their status in the deployment lifecycle.

TODO

1. Click on the DTGov Dashboard link
2. Click on the Deployment Lifecycle button
3. Your deployment list will be empty until a later step but this screen shot shows what the list will look like

The screenshot shows the Red Hat JBoss Governance Overlord web application in a Mozilla Firefox browser. The address bar shows `localhost:8080/dtgov-ui/#deployments`. The page has a navigation bar with tabs for "Design Time", "Runtime Governance", and "Repository". Below this is a "DTGov Dashboard" section with a "Deployments" link. On the left, there are "Deployment Filters" for Type, Environment, and Date Initiated, each with a dropdown menu set to "Any". A "Search by name..." input field and an "Add Deployment" button are also present. A table displays two deployments:

Deployment	Type	Initiated On
switchyard-quickstart-demo-orders-1.1.1-p2-redhat-2-sources.jar	JavaArchive	02/05/2014
switchyard-quickstart-demo-orders-1.1.1-p2-redhat-2.war	SwitchYardApplication	02/05/2014

Below the table, there is a footer for "JBoss Overlord 2013". At the bottom of the page, there are two sections: "Task Inbox" and "Deployment Lifecycle". The "Task Inbox" section has a description: "Go here to deal with all your Governance related tasks. You'll be able to accomplish all of the things you need to (e.g. claim, revoke, complete) related to human tasks that you either own or can own!" and a "Task Inbox" button. The "Deployment Lifecycle" section has a description: "Here you'll find a list of all the deployments that are somewhere in the deployment lifecycle. You'll be able to see a list of all deployments as well as where each one is in its lifecycle." and a "Deployments" button.

Step 4

View DTGov Configuration

Goals

- *View the Application Roles*
- *View the DTGov Configuration*

Governance Roles

FYI

Roles file contains the appropriate role for each user that correspond to the groupid in the workflow. Notice the fswAdmin contains the dev,qa,stage,prod roles

TODO

1. Open Terminal
2. Change to the directory
`/home/lab10/Servers/jboss-eap-6.1/standalone/configuration`
3. Display the application-roles.properties using the concatenate File (cat) command

```
application-roles.properties x
1 # The following illustrates how an admin user could be defined.
2 #
3 #admin=PowerUser,BillingAdmin,
4 #guest=guest
5 fswAdmin=overlorduser,admin.sramp,dev,qa,stage,prod,manager,arch,ba
6 dtgovworkflows=overlorduser,admin.sramp
7
```

Governance Configuration

FYI

Shows the Governance queries to kick off the workflow. Notice the SimpleReleaseprocess for the SwitchYard Application. Also notice the deployment targets and the location of the deployment. In our case we are deploying to the server on the prod target.

TODO

1. Open Terminal
2. Change to the directory
`/home/lab10/Servers/jboss-eap-6.1/standalone/configuration`
3. Display the `dtgov.properties` using the concatenate File (`cat`) command

```
dtgov.properties
1 # WARNING this file allows Property Arrays. Property Arrays which are represented
2 # as a multiple occurrence of the same property. Most Property Editors will try
3 # to be 'helpful' and remove any property defined more than once.
4 #
5 #sramp.repo.url=http://localhost:8080/s-ramp-server
6 #governance.url=http://localhost:8080/dtgov
7 #dtgov.ui.url=http://localhost:8080/dtgov-ui
8
9 governance.query.interval=20000
10
11 # Governance queries
12 governance.queries=/s-ramp/ext/SwitchYardApplication|overlord.demo.SimpleReleaseProcess|DeploymentU
13 governance.queries=/s-ramp/ext/JavaArchive|overlord.demo.SimpleReleaseProcess|DeploymentUrl={govern
14 governance.queries=/s-ramp/ext/JavaWebApplication|overlord.demo.SimpleReleaseProcess|DeploymentUrl=
15 governance.queries=/s-ramp/ext/JavaEnterpriseApplication|overlord.demo.SimpleReleaseProcess|Deploym
16 governance.queries=/s-ramp/ext/ArtifactGrouping[xp2:matches(@name\, 'Project.*')]|overlord.demo.Sim
17
18 # Deployment targets
19 governance.targets= dev|http://www.jboss.org/overlord/deployment-status.owl#InDev|copy|/tmp/dev/jb
20 governance.targets= qa|http://www.jboss.org/overlord/deployment-status.owl#InQa|copy|/tmp/qa/jbos
21 governance.targets= stage|http://www.jboss.org/overlord/deployment-status.owl#InStage|copy|/tmp/stag
22 governance.targets= prod|http://www.jboss.org/overlord/deployment-status.owl#InProd|copy|/tmp/prod/
23
24 # Workflow package maven info
25 dtgov.workflows.group=org.overlord.dtgov
26 dtgov.workflows.name=dtgov-workflows
27 dtgov.workflows.package=SRAMPPackage
28 dtgov.workflows.version=1.0.1.Final-redhat-8
29 sramp.repo.user=dtgovworkflows
30 sramp.repo.password=${vault:VAULT::dtgov::dtgov-workflows.password::1}
31 governance.bpm.user=dtgovworkflows
32 governance.bpm.password=${vault:VAULT::dtgov::dtgov-workflows.password::1}
33 governance.user=dtgovworkflows
34 governance.password=${vault:VAULT::dtgov::dtgov-workflows.password::1}
```

Step 5

Distribution Management

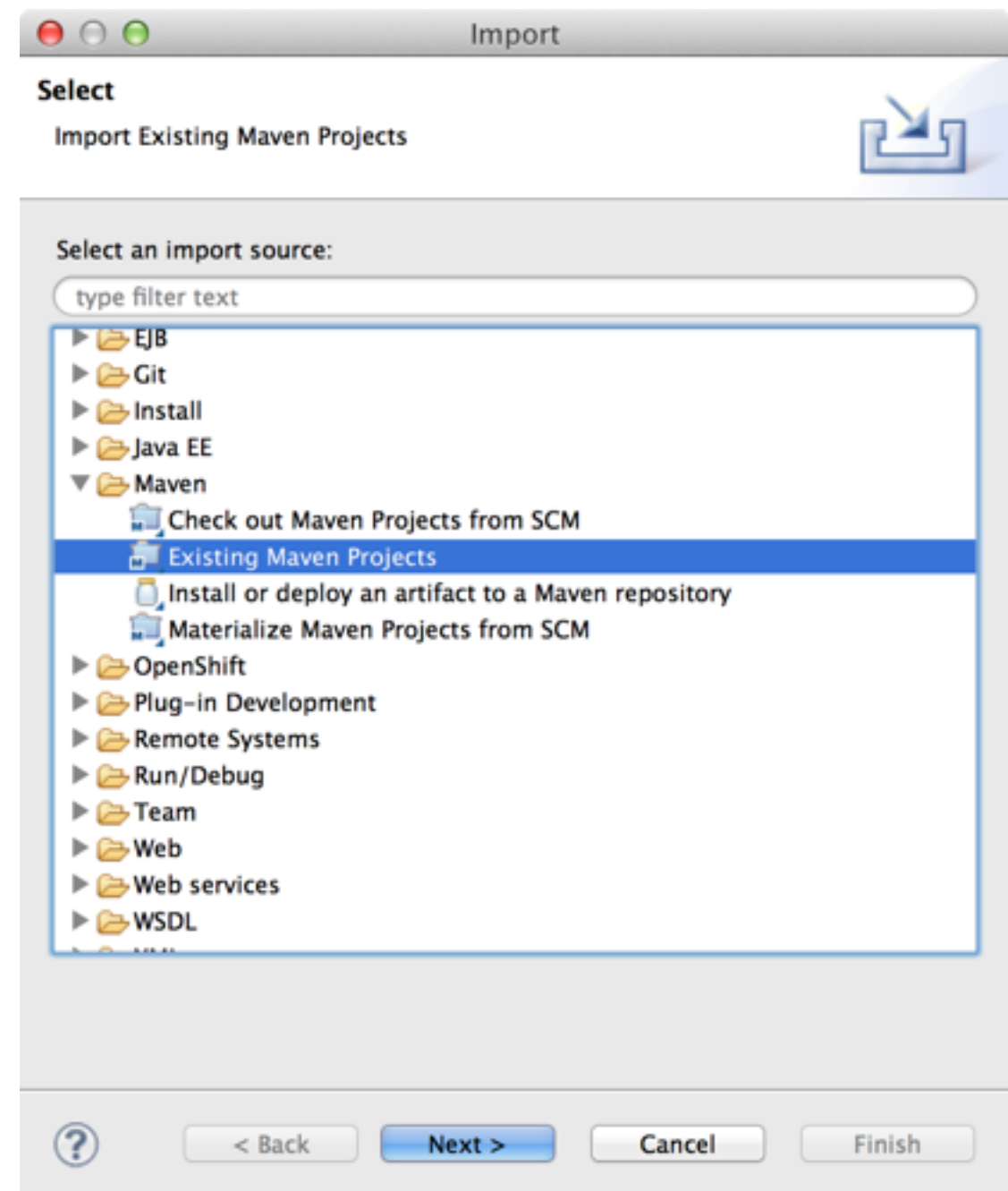
Goals

- *Import the Home Loan Application*
- *Review the POM for the Distribution Management settings*

Importing the Application

TODO

1. File -> Import ... from the JBDS menu.
- Select Maven -> Existing Maven Projects
 - Click Next



Importing the Application

TODO

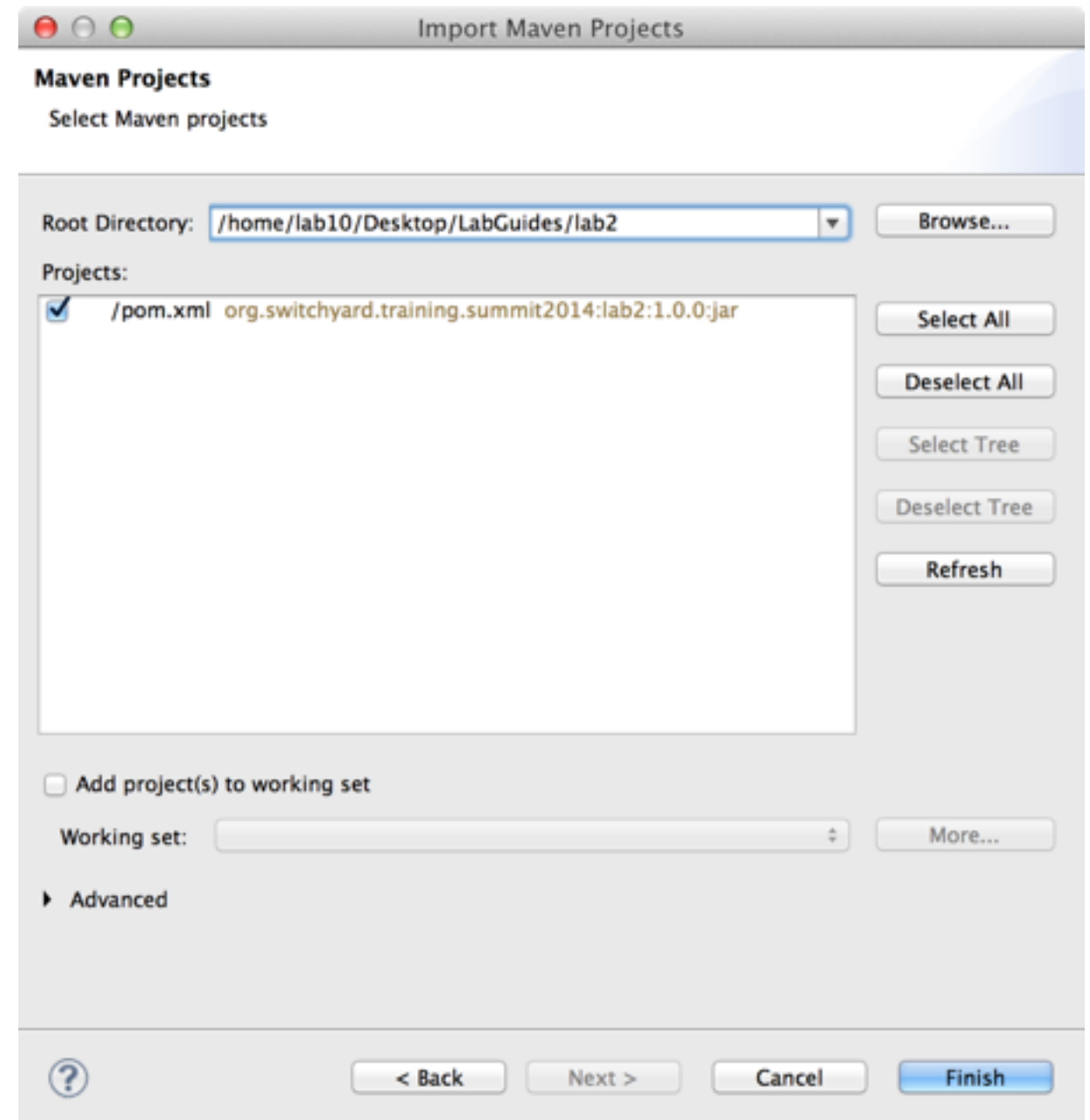
1. Click Browse ... and navigate to the location of lab1. For example:

`/home/lab10/Desktop/LabGuides/lab2`

2. Make sure the pom.xml is checked for:

`org.switchyard.training.summit2014:lab2`

- Click Finish



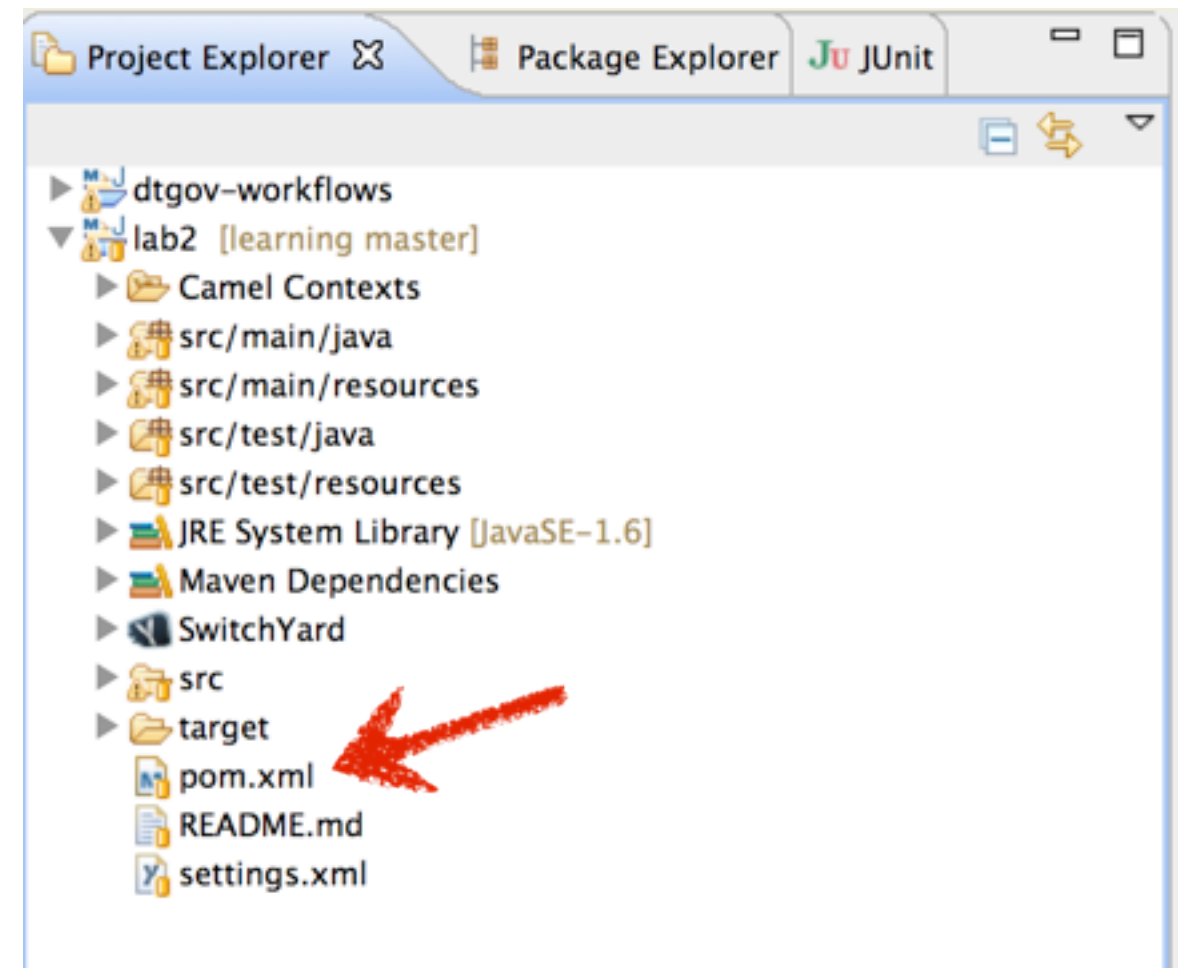
Configuring Distribution Management

FYI

The POM will contain the distribution management settings to connect to S-RAMP

TODO

1. Double-click on pom.xml in the Project Explorer to open it in the editor.



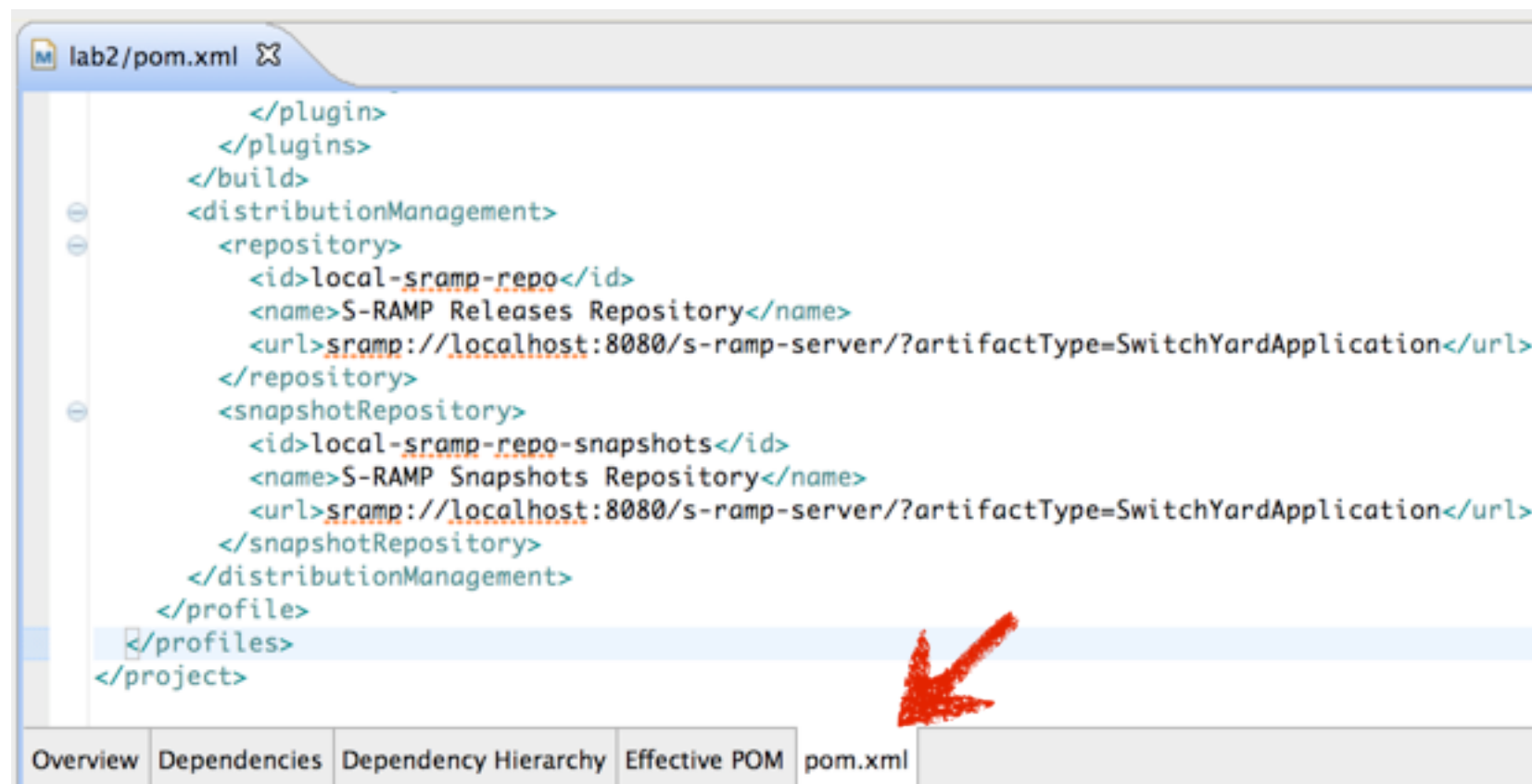
Configuring Distribution Management

TODO

1. Click on the 'pom.xml' tab to see the XML content of the file.

FYI

The distribution management element will contain the url element for the S-RAMP server and define the artifact type which will correspond to the workflow definition in the dtgov.properties file.



The screenshot shows an IDE window with a tab labeled 'lab2/pom.xml'. The XML content is displayed with syntax highlighting. The 'distributionManagement' section is expanded, showing two repositories: 'local-sramp-repo' and 'local-sramp-repo-snapshots'. Both have the same URL: 'sramp://localhost:8080/s-ramp-server/?artifactType=SwitchYardApplication'. A red arrow points to the 'pom.xml' tab in the bottom panel.

```
</plugin>
</plugins>
</build>
<distributionManagement>
  <repository>
    <id>local-sramp-repo</id>
    <name>S-RAMP Releases Repository</name>
    <url>sramp://localhost:8080/s-ramp-server/?artifactType=SwitchYardApplication</url>
  </repository>
  <snapshotRepository>
    <id>local-sramp-repo-snapshots</id>
    <name>S-RAMP Snapshots Repository</name>
    <url>sramp://localhost:8080/s-ramp-server/?artifactType=SwitchYardApplication</url>
  </snapshotRepository>
</distributionManagement>
</profile>
</profiles>
</project>
```

Overview Dependencies Dependency Hierarchy Effective POM pom.xml

Step 6

Create Production Link

Goals

- *Create the link for production to the server deployment directory*

Create Production Link

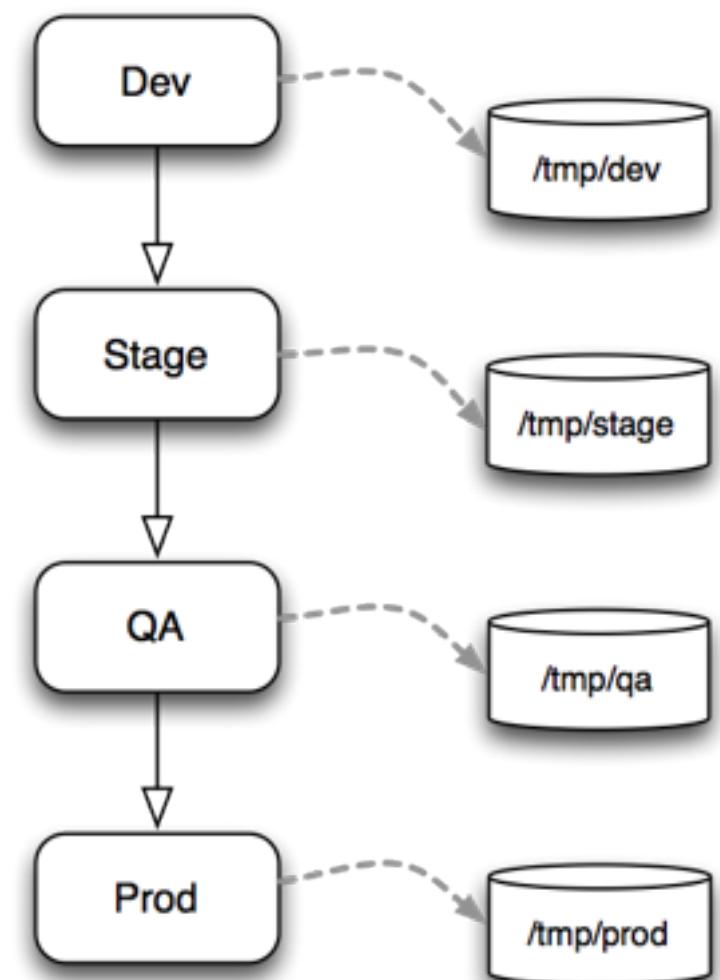
FYI

The governance targets for our default workflow is using the file copy deployment type. Linking the production target /tmp/prod to the server deployment directory will deploy the application to the server when the stage task is passed. In addition to file copy, applications can be deployed with JON, JBoss AS CLI and Maven deployment types.

TODO

1. Open up a terminal session
2. Create a symbolic link for the production directory using the commands below (copy and paste recommended) :

```
mkdir -p /tmp/prod/jbossas7/standalone/deployments
ln -sv /tmp/prod/jbossas7/standalone/deployments /home/lab10/Servers/jboss-eap-6.1/standalone/deployments
```



Step 7

Deploy to S-RAMP

Goals

- *Deploy to S-RAMP*

Deploy App to S-RAMP

FYI

The run configuration will do a maven clean and deploy the KieJarArchive jar to the FSW server. The deployment can also be done at the command line.

TODO

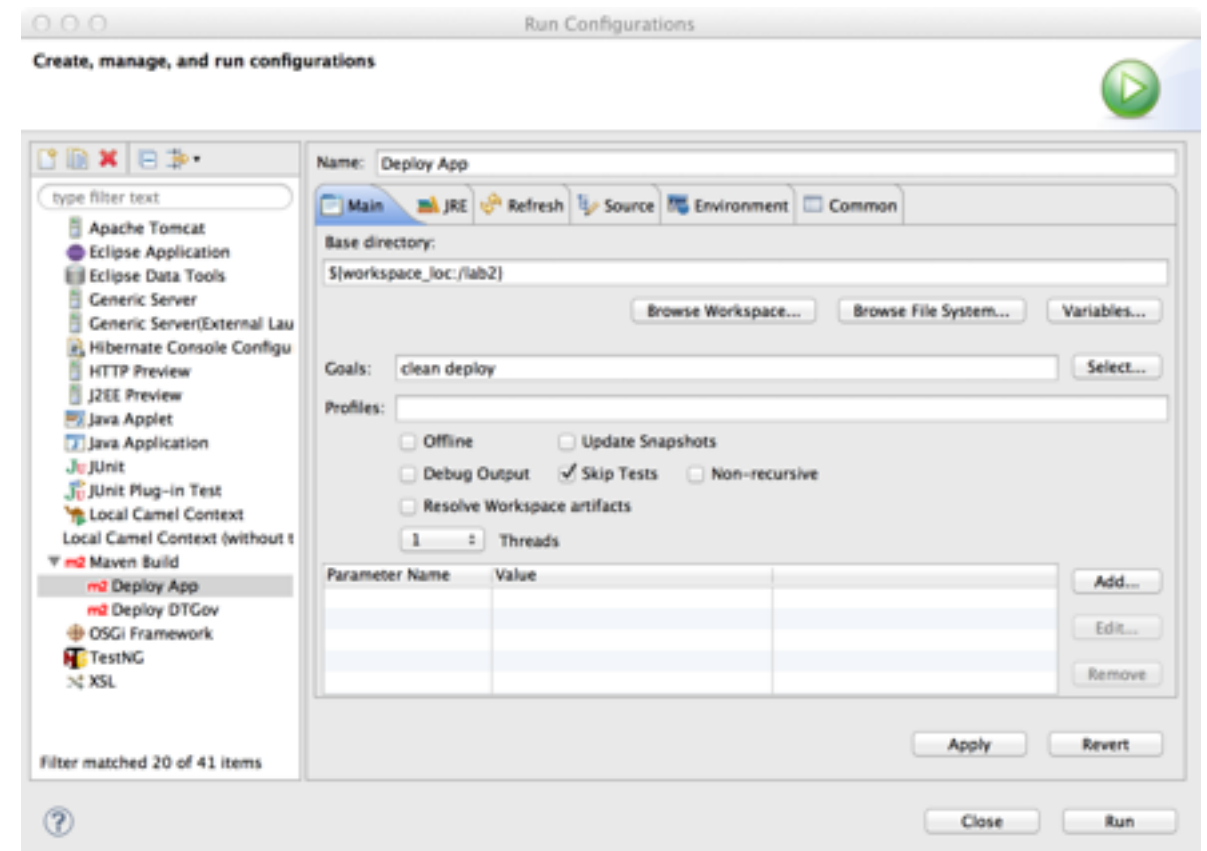
1. Right-click on the lab2 project in the Project Explorer
2. Select Run As -> Run Configurations
3. Right-click on Maven Build and select New
4. Enter the following values for the build:

Name : Deploy App

Base directory : \${workspace_loc:/lab2}

Goals : clean deploy

5. Click Run



Configuring Distribution Management

TODO

1. Confirm that the deployment was successful in the build output.

```
[INFO] --- maven-jar-plugin:2.3.2:jar (default-jar) @ lab2 ---
[INFO] Building jar: /home/kpeeples/mystuff/Summit2014/lab2/target/lab2-1.0.0.jar
[INFO] --- maven-install-plugin:2.3.1:install (default-install) @ lab2 ---
[INFO] Installing /home/kpeeples/mystuff/Summit2014/lab2/target/lab2-1.0.0.jar to /home/kpeeples/.m2/repository/org/switchyard/training/summit2014/lab2/1.0.0/lab2-1.0.0.jar
[INFO] Installing /home/kpeeples/mystuff/Summit2014/lab2/pom.xml to /home/kpeeples/.m2/repository/org/switchyard/training/summit2014/lab2/1.0.0/lab2-1.0.0.pom
[INFO] --- maven-deploy-plugin:2.7:deploy (default-deploy) @ lab2 ---
Uploading: sramp://localhost:8080/s-ramp-server/?artifactType=SwitchYardApplication/org/switchyard/training/summit2014/lab2/1.0.0/lab2-1.0.0.jar
[INFO] Uploading s-ramp artifact: org/switchyard/training/summit2014/lab2/1.0.0/lab2-1.0.0.jar
[INFO] Uploading s-ramp artifact: org/switchyard/training/summit2014/lab2/1.0.0/lab2-1.0.0.jar.shal
[INFO] Storing hash value as s-ramp property: lab2-1.0.0.jar.shal
[INFO] Uploading s-ramp artifact: org/switchyard/training/summit2014/lab2/1.0.0/lab2-1.0.0.jar.md5
[INFO] Storing hash value as s-ramp property: lab2-1.0.0.jar.md5
Uploaded: sramp://localhost:8080/s-ramp-server/?artifactType=SwitchYardApplication/org/switchyard/training/summit2014/lab2/1.0.0/lab2-1.0.0.jar (0 B at 0.0 KB/sec)
Uploading: sramp://localhost:8080/s-ramp-server/?artifactType=SwitchYardApplication/org/switchyard/training/summit2014/lab2/1.0.0/lab2-1.0.0.pom
[INFO] Uploading s-ramp artifact: org/switchyard/training/summit2014/lab2/1.0.0/lab2-1.0.0.pom
[INFO] Uploading s-ramp artifact: org/switchyard/training/summit2014/lab2/1.0.0/lab2-1.0.0.pom.shal
[INFO] Storing hash value as s-ramp property: lab2-1.0.0.pom.shal
[INFO] Uploading s-ramp artifact: org/switchyard/training/summit2014/lab2/1.0.0/lab2-1.0.0.pom.md5
[INFO] Storing hash value as s-ramp property: lab2-1.0.0.pom.md5
Uploaded: sramp://localhost:8080/s-ramp-server/?artifactType=SwitchYardApplication/org/switchyard/training/summit2014/lab2/1.0.0/lab2-1.0.0.pom (0 B at 0.0 KB/sec)
Downloading: sramp://localhost:8080/s-ramp-server/?artifactType=SwitchYardApplication/org/switchyard/training/summit2014/lab2/maven-metadata.xml
Downloaded: sramp://localhost:8080/s-ramp-server/?artifactType=SwitchYardApplication/org/switchyard/training/summit2014/lab2/maven-metadata.xml (344 B at 0.8 KB/sec)
Uploading: sramp://localhost:8080/s-ramp-server/?artifactType=SwitchYardApplication/org/switchyard/training/summit2014/lab2/maven-metadata.xml
[INFO] Uploading s-ramp artifact: org/switchyard/training/summit2014/lab2/maven-metadata.xml
[INFO] Skipping unsupported artifact: org/switchyard/training/summit2014/lab2/maven-metadata.xml
[INFO] Uploading s-ramp artifact: org/switchyard/training/summit2014/lab2/maven-metadata.xml.shal
[INFO] Skipping unsupported artifact: org/switchyard/training/summit2014/lab2/maven-metadata.xml.shal
[INFO] Uploading s-ramp artifact: org/switchyard/training/summit2014/lab2/maven-metadata.xml.md5
[INFO] Skipping unsupported artifact: org/switchyard/training/summit2014/lab2/maven-metadata.xml.md5
Uploaded: sramp://localhost:8080/s-ramp-server/?artifactType=SwitchYardApplication/org/switchyard/training/summit2014/lab2/maven-metadata.xml (345 B at 336.9 KB/sec)
[INFO] .....
[INFO] BUILD SUCCESS
[INFO] .....
[INFO] Total time: 1:14.093s
[INFO] Finished at: Thu Mar 27 04:00:12 EDT 2014
[INFO] Final Memory: 99M/594M
[INFO] .....
```

Configuring Distribution Management

TODO

1. Confirm in the server console output that the governance workflow has been initiated.

```
04:00:05,197 INFO [org.overlord.sramp.governance.QueryExecutor] (EJB default - 9) Starting workflow overlord.demo.SimpleReleaseProcess for artifact 6e2b6d16-fb76-46ab-aa03-d442bc940f77
04:00:05,276 INFO [org.overlord.dtgov.bpm.util.HttpClientWorkItemHandler] (EJB default - 9) Calling POST TO: http://localhost:8080/dtgov/rest/deploy/dev/6e2b6d16-fb76-46ab-aa03-d442bc940f77
04:00:05,416 INFO [org.overlord.sramp.governance.services.DeploymentResource] (http-localhost.localdomain/127.0.0.1:8080-28) Creating deploy directory /tmp/dev/jbossas7/standalone/deployments
04:00:06,968 INFO [org.overlord.dtgov.bpm.util.HttpClientWorkItemHandler] (EJB default - 9) reply={status=success, target=COPY:/tmp/dev/jbossas7/standalone/deployments/lab2-1.0.0.jar}
04:00:06,985 INFO [org.overlord.dtgov.bpm.util.HttpClientWorkItemHandler] (EJB default - 9) Calling PUT TO: http://localhost:8080/dtgov/rest/update/classification/http%3A%2F%2Fwww.jboss.org%2Foverlord%2Fdeployment-status.owl%23DevTest/6e2b6d16-fb76-46ab-aa03-d442bc940f77
04:00:07,972 INFO [org.overlord.dtgov.bpm.util.HttpClientWorkItemHandler] (EJB default - 9) reply={artifactName=lab2-1.0.0.jar, artifactCreatedBy=fswAdmin, status=success}
04:00:07,981 INFO [org.overlord.dtgov.bpm.util.HttpClientWorkItemHandler] (EJB default - 9) Calling POST TO: http://localhost:8080/dtgov/rest/notify/email/dev/deployed/dev/6e2b6d16-fb76-46ab-aa03-d442bc940f77
04:00:08,760 INFO [org.overlord.dtgov.bpm.util.HttpClientWorkItemHandler] (EJB default - 9) reply={status=success}
04:00:08,873 INFO [org.overlord.dtgov.bpm.ejb.ProcessBean$Proxy$_$$_WeldSubclass] (EJB default - 9) Process started ... : processInstanceId = 1
```


Step 8

Execute Release Workflow

Goals

- *Pass the deployment in Dev*
- *Pass the deployment in QA*
- *Pass the deployment in Stage*
- *Verify the application is deployed*
- *Pass the deployment in Prod*
- *Review Deployment History and Contents*

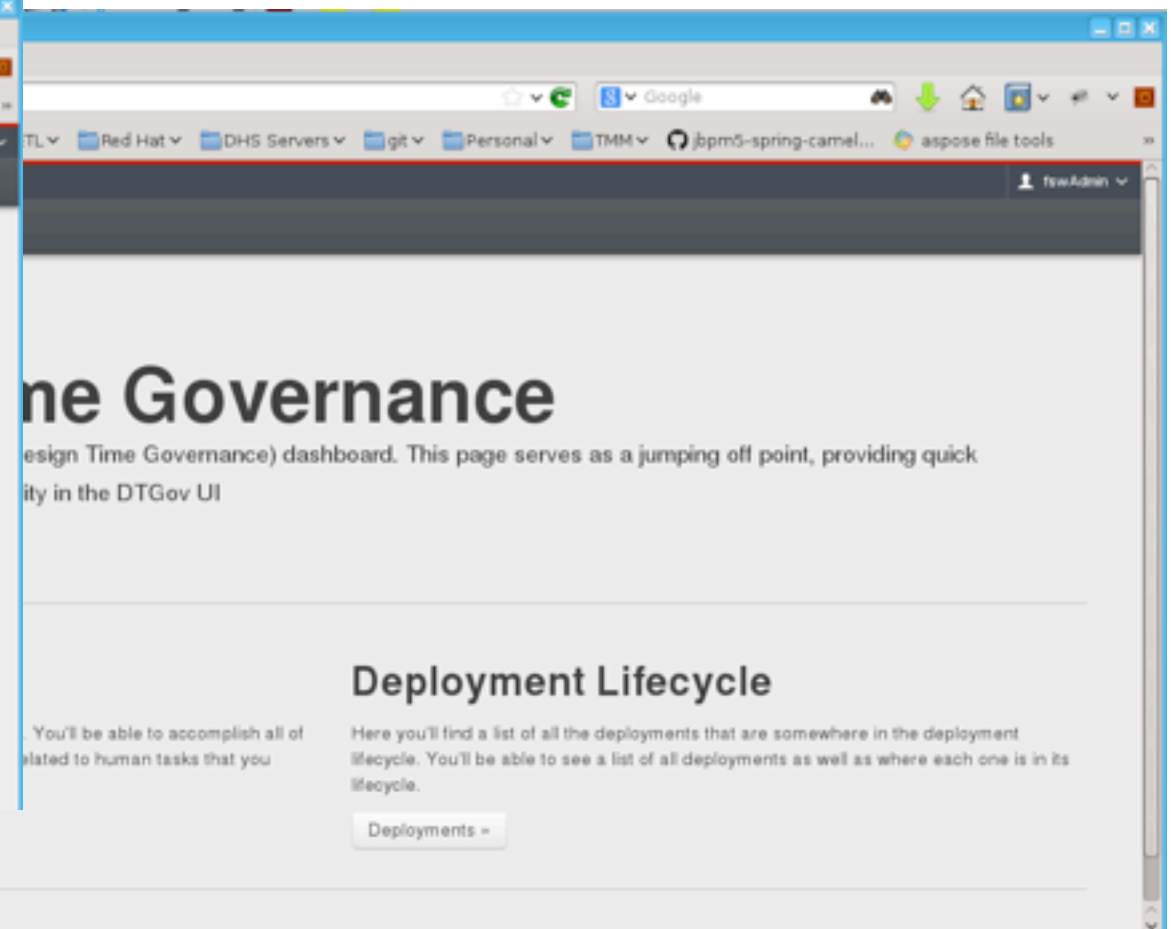
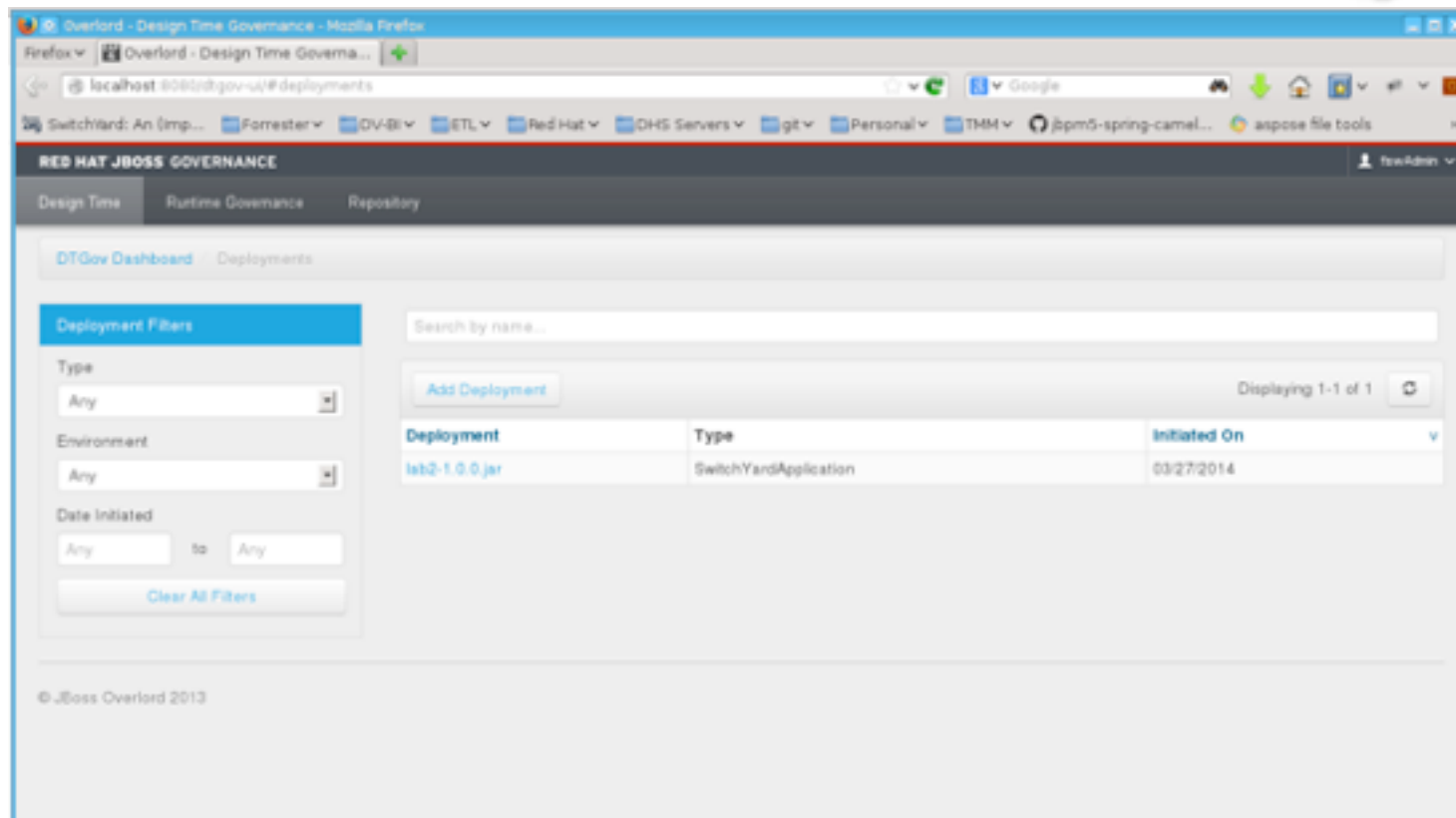
DTGov Deployments

FYI

The lab2 Switchyard Application will show in the deployment list. This allows the user to view deployment in go further into history and the individual artifacts by clicking on the deployment.

TODO

1. Open your Browser and go to <http://localhost:8080/dtgov-ui>
2. View the lab2 deployment by clicking on the Deployments button.



DTGov Tasks

FYI

The available tasks for the user is displayed in the task inbox. In our example we should have a Test Dev task which is the first human task that is defined for a Switchyard Application.

TODO

1. Return to the Design Time dashboard
2. Click on the Task Inbox Button

The image displays two screenshots of the Red Hat JBoss Governance Overlord DTGov interface. The left screenshot shows the 'Design Time Governance' dashboard with a 'Task Inbox' button. The right screenshot shows the 'Task Inbox' view with a table of tasks, including 'Test Dev'.

Design Time Governance Dashboard (Left Screenshot):

- Header: RED HAT JBOSS GOVERNANCE
- Navigation: Design Time, Runtime Governance, Repository
- Section: Design Time Governance
- Text: Welcome to the Overlord DTGov (Design Time Governance) dashboard. access to frequently used functionality in the DTGov UI
- Section: Task Inbox
- Text: Go here to deal with all your Governance related tasks. You'll be able to accomplish all of the things you need to (e.g. claim, revoke, complete) related to human tasks that you either own or can own!
- Button: Task Inbox

Task Inbox View (Right Screenshot):

- Header: RED HAT JBOSS GOVERNANCE
- Navigation: Design Time, Runtime Governance, Repository
- Breadcrumbs: DTGov Dashboard > Task Inbox
- Task Filters:

 - Owner: ☒ Any, ☐ My Tasks, ☐ Active Tasks, ☐ Group Tasks
 - Priority:
 - Date Due: to
 - Button: Clear All Filters

- Table:

Name	Priority	Owner	Status	Due On
Test Dev	3		Ready	

- Text: Displaying 1-1 of 1

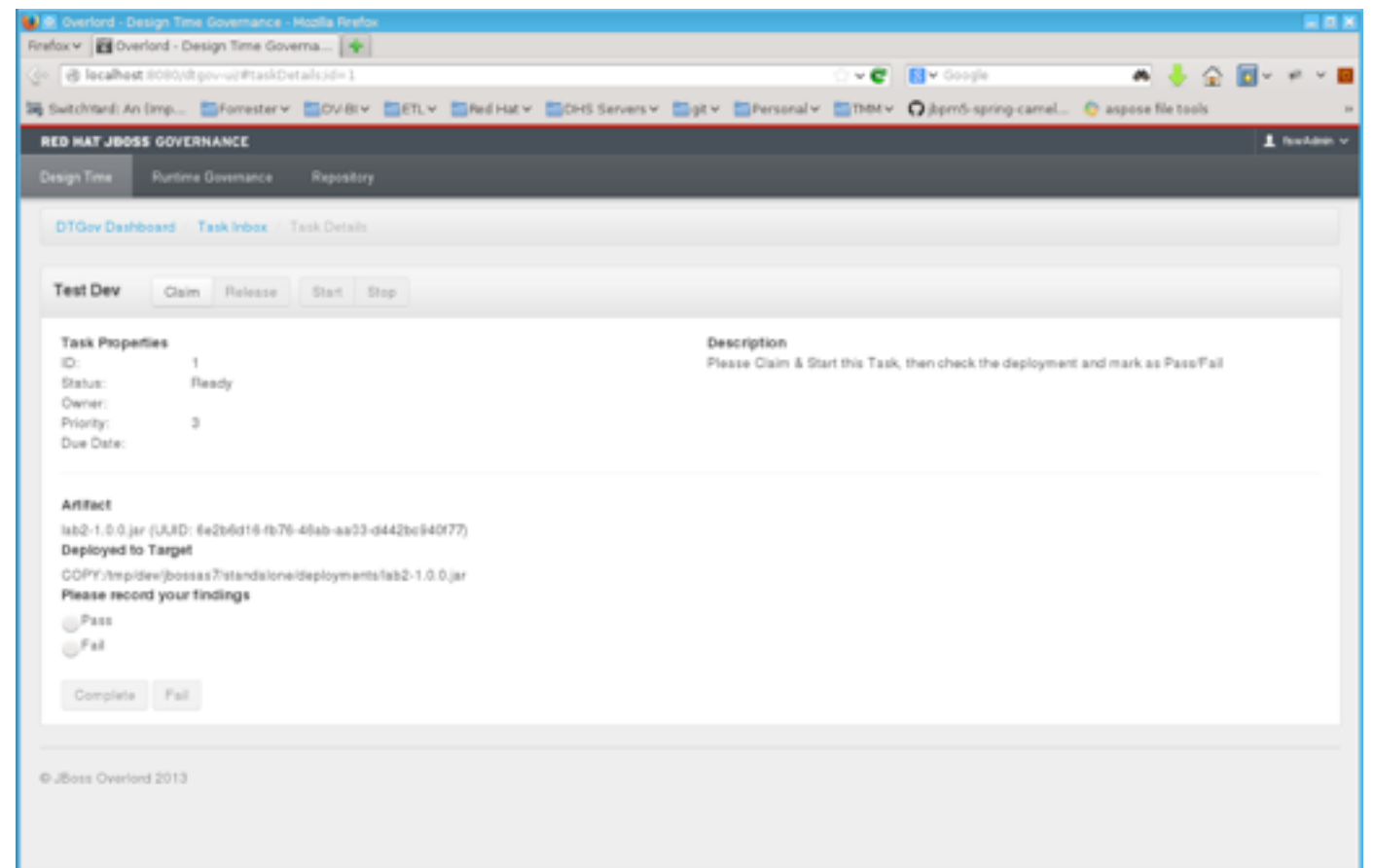
Claiming a Task

FYI

Remember our workflow for the human task for each environment – Claim, Start, Pass/Fail and Complete.

TODO

1. Claim the task by clicking on the Claim button.



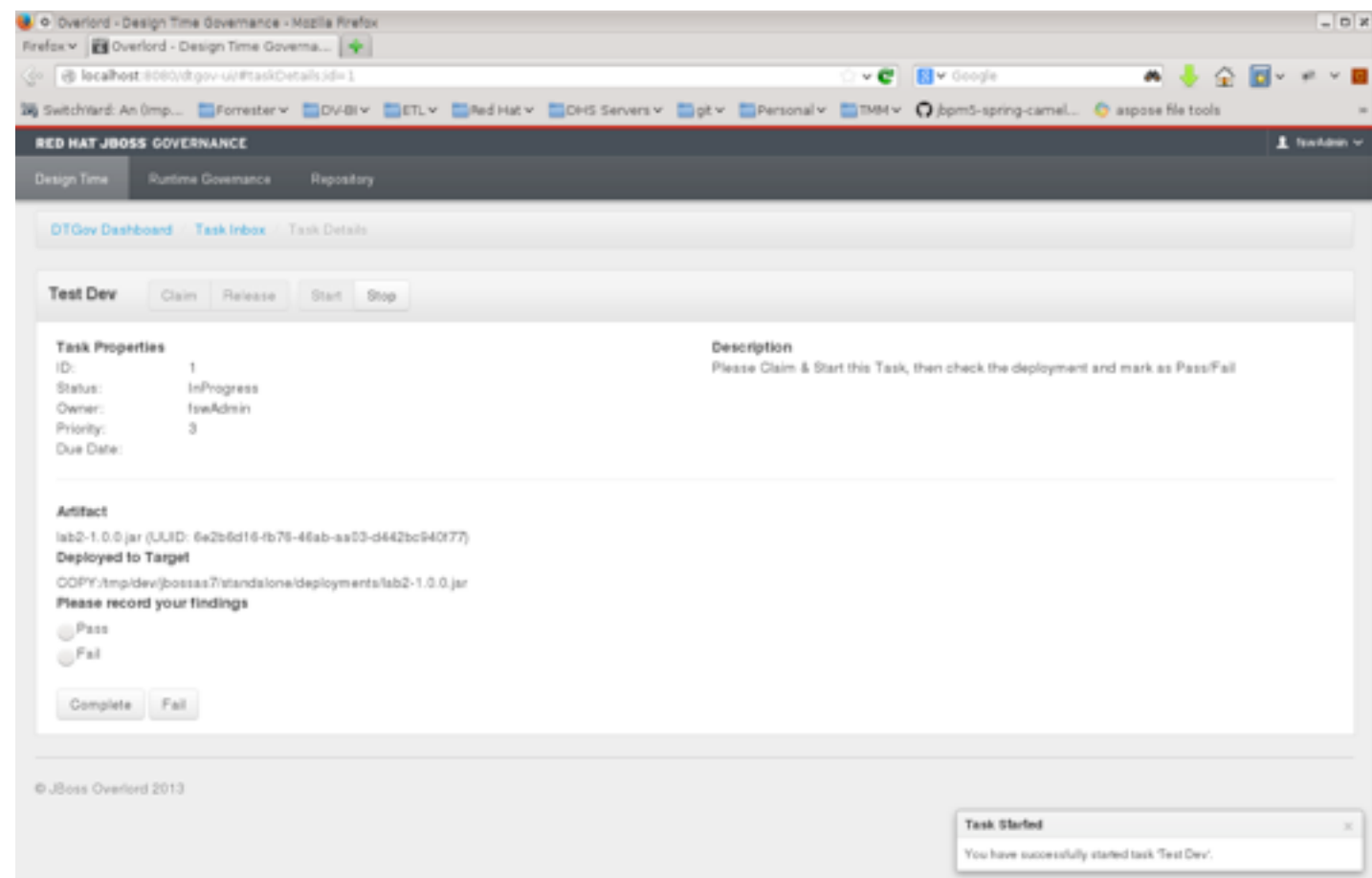
Starting a Task

FYI

We have to start our task before we can Pass or fail it.

TODO

1. Start the task by clicking on the Start button.



Completing a Task

FYI

We will pass the task so the deployment will go to the next environment for testing.

TODO

1. Click on the Pass radio button.
2. Complete the task by clicking on the Complete button.

The screenshot shows the 'Red Hat JBoss Governance' web application. The 'Task Details' page for 'Test Dev' is displayed. The task is in 'InProgress' status, owned by 'fooAdmin', with a priority of 3 and a due date. The description instructs the user to 'Please Claim & Start this Task, then check the deployment and mark as Pass/Fail'. An artifact 'lsb2-1.0.0.jar' is listed, deployed to a target. The user is prompted to 'Please record your findings' and can select 'Pass' (selected) or 'Fail'. 'Complete' and 'Fail' buttons are at the bottom. A 'Completing Task' dialog box is visible in the bottom right corner, showing 'Completing task Test Dev', please wait...

Overlord - Design Time Governance - Mozilla Firefox
localhost:8080/dt-gov-us/#taskDetails.id=1

RED HAT JOSS GOVERNANCE

Design Time Runtime Governance Repository

DTGov Dashboard Task Inbox Task Details

Test Dev Claim Release Start Stop

Task Properties

ID:	1
Status:	InProgress
Owner:	fooAdmin
Priority:	3
Due Date:	

Description

Please Claim & Start this Task, then check the deployment and mark as Pass/Fail

Artifact

lsb2-1.0.0.jar (UUID: 6e2b6d16-4b76-46ab-aa03-d442bc940f77)

Deployed to Target

COPY: /tmp/dev/jbossas7/standalone/deployments/lsb2-1.0.0.jar

Please record your findings

☒ Pass
☐ Fail

Complete Fail

© JBoss Overlord 2013

Completing Task
Completing task Test Dev, please wait...

Complete QA Task

FYI

We want to move the deployment through QA – Claim, Start, Pass/Fail and Complete.

TODO

1. Return to the Task Inbox.
2. Repeat the steps you followed for Dev

The screenshot shows the 'RED HAT JOSS GOVERNANCE' interface in a web browser. The 'Task Inbox' tab is active. On the left, there are 'Task Filters' for Owner (Any, My Tasks, Active Tasks, Group Tasks), Priority (Any), and Date Due (Any to Any). A 'Clear All Filters' button is at the bottom of the filter section. The main area displays a table with one task: 'Test Gs' with a priority of 3 and a status of 'Ready'. The footer indicates '© JBoss Overlord 2013'.

Name	Priority	Owner	Status	Due On
Test Gs	3		Ready	

Complete QA Task

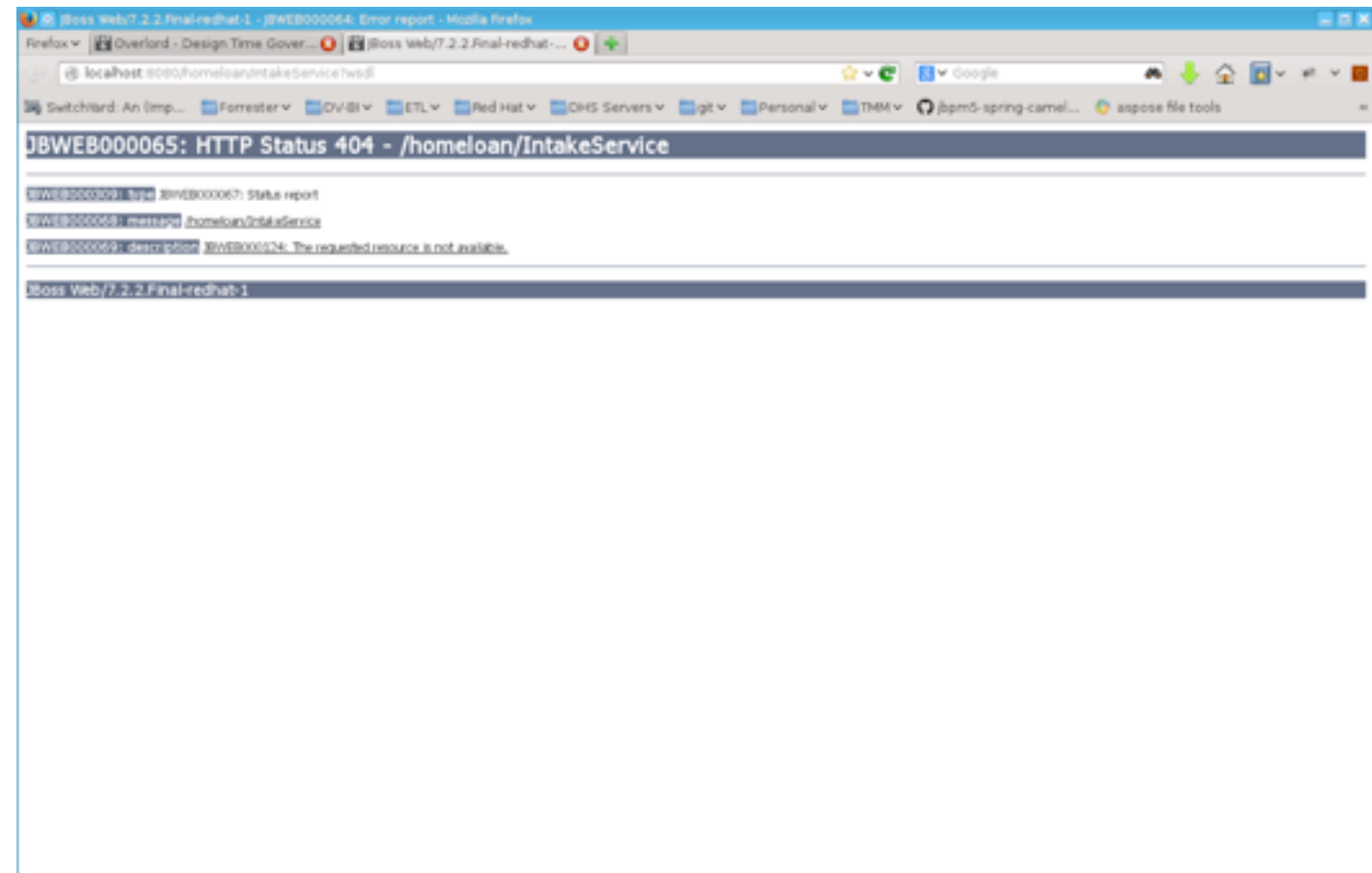
FYI

We only setup the link for the production environment so the deployment won't be available on our server until after it passes staging.

TODO

1. In a new browser tab visit the SOAP endpoint URL to confirm the endpoint has not been deployed yet:

<http://localhost:8080/homeloan/IntakeService?wsdl>



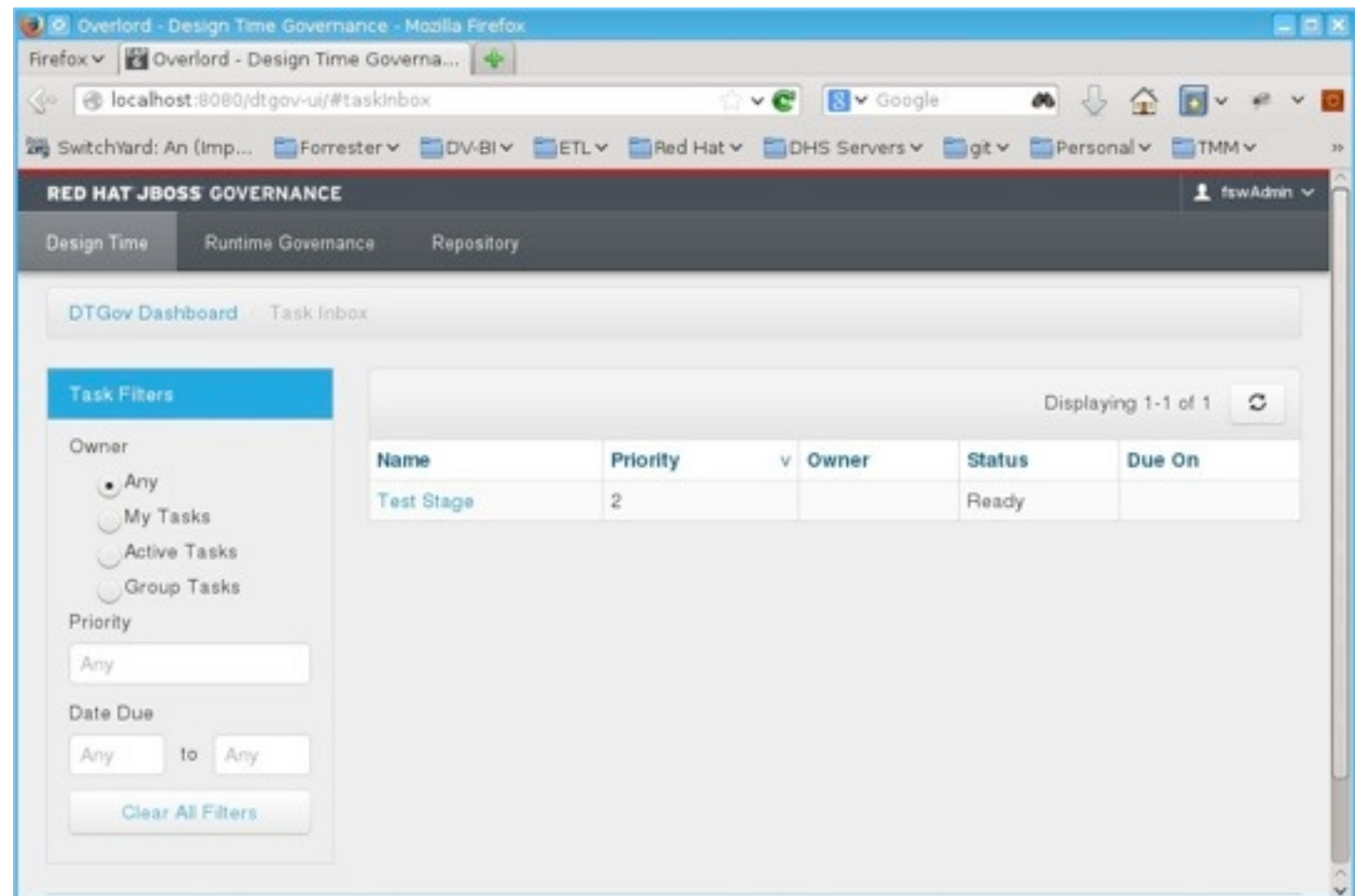
Complete Stage Task

FYI

We want to move the deployment through Staging – Claim, Start, Pass/Fail and Complete.

TODO

1. Return to the Task Inbox.
2. Repeat the steps you followed for QA



The screenshot shows the 'RED HAT JBOSS GOVERNANCE' interface in a Mozilla Firefox browser. The 'Design Time' tab is selected. The 'Task Filters' section on the left includes options for Owner (Any, My Tasks, Active Tasks, Group Tasks), Priority (Any), and Date Due (Any to Any). The 'Task Inbox' table displays one task:

Name	Priority	Owner	Status	Due On
Test Stage	2		Ready	

The interface also shows 'Displaying 1-1 of 1' and a 'Clear All Filters' button.

Complete Stage Task

FYI

We created our link for our production target to our FSW server deployment folder. Normally a Design Time Server will be different than the run time server but we have one server instance for our example.

TODO

1. Check the console log to verify the application was deployed.

```
04:19:22,814 INFO [org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandler] Ohttp://localhost:8080/dgwp/rest/update/classification/http3ba3c2f=2www.jboss.org#PowerLord#2Pdp  
[org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandler] Calling PUT To: http://localhost:8080/dgwp/rest/update/classification/http3ba3c2f=2www.jboss.org#PowerLord#2Pdp  
04:19:23,293 INFO [org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandler] reply={artifactName=lab2-1.0.0.jar, artifactCreatedBy=fwAdmin, status=success)  
04:19:23,243 INFO [org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandler] Calling POST To: http://localhost:8080/dgwp/rest/deploy/prod/6e2bd16-fb76-45ab-aad3-444bc34cf77  
04:19:23,814 INFO [org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandler] reply={status=success, targetCDPR=/tmp/prod/jbossas/standalone/deployments/Lab2-1.0.0.jar}  
04:19:23,819 INFO [org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandler] Calling PUT To: http://localhost:8080/dgwp/rest/update/classification/http3ba3c2f=2www.jboss.org#PowerLord#2Pdp  
[org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandler] Calling PUT To: http://localhost:8080/dgwp/rest/update/classification/http3ba3c2f=2www.jboss.org#PowerLord#2Pdp  
04:19:24,227 INFO [org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandler] reply={artifactName=lab2-1.0.0.jar, artifactCreatedBy=fwAdmin, status=success)  
04:19:24,230 INFO [org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandler] Calling POST To: http://localhost:8080/dgwp/rest/notify/wml/prod/deployed/prod/6e2bd16-fb76-45ab-aad3-444bc34  
04:19:24,439 INFO [org.jboss.as.server.deployment] (MSC service thread 1-7) JBAS015679: Starting deployment of 'Lab2-1.0.0.jar' (runtime-name: 'Lab2-1.0.0.jar')  
04:19:24,466 INFO [org.springframework.web.servlet.mvc.annotation.AnnotationMethodHandler] reply={status=success)  
04:19:24,467 INFO [org.jboss.weld.deployer] (MSC service thread 1-4) JBAS005002: Processing weld deployment Lab2-1.0.0.jar  
04:19:24,501 INFO [org.jboss.weld.deployer] (MSC service thread 1-8) JBAS005005: Starting services for CDI deployment: Lab2-1.0.0.jar  
04:19:24,502 INFO [org.switchyard] (MSC service thread 1-8) Deploying Switchyard application 'Lab2-1.0.0.jar'  
04:19:24,506 INFO [org.jboss.weld.deployer] (MSC service thread 1-8) JBAS005006: Starting weld service for deployment Lab2-1.0.0.jar  
04:19:24,584 INFO [org.switchyard] (MSC service thread 1-2) Starting Switchyard service  
04:19:24,586 INFO [org.apache.camel.management.ManagementStrategyFactory] (MSC service thread 1-2) JMX enabled.  
04:19:24,589 INFO [org.switchyard.common.camel.SwitchyardCamelContext] (MSC service thread 1-2) Apache Camel 1.1.1-p5-redhat-1 (CamelContext: camel-4) is starting  
04:19:24,603 INFO [org.apache.camel.impl.converter.DefaultTypeConverter] (MSC service thread 1-2) Loaded 182 type converters  
04:19:24,613 INFO [org.apache.camel.management.DefaultManagementLifecycleStrategy] (MSC service thread 1-2) StatisticsLevel at All so enabling load performance statistics  
04:19:24,617 INFO [org.switchyard.common.camel.SwitchyardCamelContext] (MSC service thread 1-2) Total @ routes, of which 0 is started.  
04:19:24,617 INFO [org.switchyard.common.camel.SwitchyardCamelContext] (MSC service thread 1-2) Apache Camel 1.1.1-p5-redhat-1 (CamelContext: camel-4) started in 0.026 seconds  
04:19:24,662 INFO [org.switchyard.common.camel.SwitchyardCamelContext] (MSC service thread 1-2) Route: direct((urn:jboss:lan:1.0)/PreQualificationProcess/CreditService started and consuming from: Endpoint[direct://A/Burn/homeLoan:1.0/PreQualificationProcess/CreditService]  
04:19:24,957 INFO [org.switchyard.common.camel.SwitchyardCamelContext] (MSC service thread 1-2) Route: direct((urn:jboss:lan:1.0)/PreQualificationProcess/LoanEvaluationService started and consuming from: Endpoint[direct://A/Burn/homeLoan:1.0/PreQualificationProcess/LoanEvaluationService]  
04:19:25,006 WARN [org.jboss.ruleflow.core.validation.RuleFlowProcessValidator] (MSC service thread 1-2) Process variable Parameter uses ObjectDataType for default type (java.lang) which could cause problems with setting variables, use d  
edicated type instead  
04:19:25,006 INFO [org.jboss.ruleflow.core.validation.RuleFlowProcessValidator] (MSC service thread 1-2) Process variable Result uses ObjectDataType for default type (java.lang) which could cause problems with setting variables, use des  
tinated type instead  
04:19:25,090 INFO [org.drools.compiler.kie.builder.impl.KieRepositoryImpl] (MSC service thread 1-2) KieModule was added: MemoryKieModule[ ReleasedId=org.switchyard.bpmapp:3]  
04:19:25,179 INFO [org.switchyard.common.camel.SwitchyardCamelContext] (MSC service thread 1-2) Route: direct((urn:jboss:lan:1.0)/IntakeCustomerLookup started and consuming from: Endpoint[direct://A/Burn/homeLoan:1.0/IntakeCustomerLo  
up]  
04:19:25,257 INFO [org.switchyard.common.camel.SwitchyardCamelContext] (MSC service thread 1-2) Route: direct((urn:jboss:lan:1.0)/IntakePrequalificationService started and consuming from: Endpoint[direct://A/Burn/homeLoan:1.0/IntakePr  
eQualificationService]  
04:19:25,296 INFO [org.switchyard.common.camel.SwitchyardCamelContext] (MSC service thread 1-2) Route: route2 started and consuming from: Endpoint[switchyard://IntakeService]  
04:19:25,426 INFO [org.drools.compiler.kie.builder.impl.KieRepositoryImpl] (MSC service thread 1-2) KieModule was added: MemoryKieModule[ ReleasedId=org.switchyard.bpmapp:4]  
04:19:25,530 INFO [org.switchyard.common.camel.SwitchyardCamelContext] (MSC service thread 1-2) Route: direct((urn:jboss:lan:1.0)/DataSOP started and consuming from: Endpoint[direct://A/Burn/homeLoan:1.0/DataSOP])  
04:19:25,548 INFO [org.switchyard] (MSC service thread 1-2) Addressing {enabled = false, required = false}  
04:19:25,549 INFO [org.switchyard] (MSC service thread 1-2) MTOM {enabled = false, threshold = 0}  
04:19:25,550 INFO [org.jboss.ws.cxf.metadata] (MSC service thread 1-2) JWSD020406: Adding service endpoint metadata: id=IntakeService  
address=http://localhost:8080/homeLoan/IntakeService  
wsElement=org.switchyard.component.soa.endpoint.BaseWebService  
serviceName=(urn:jboss:lan:1.0)/IntakeService  
portName=(urn:jboss:lan:1.0)/IntakeServicePort  
annotationswellLocation=null  
wellLocatsionovarrdevsrc/content/Lab2-1.0.0.jar/IntakeService.wsdl  
xmlns=ns1=xsi:type=t  
xsd:schemaLocation=  
http://schemas.xml.org/ws/2003/05/addressing mode=URI  
04:19:25,575 INFO [org.apache.cxf.service.factory.ReflectionServiceFactoryBean] (MSC service thread 1-2) Creating Service {urn:jboss:lan:1.0}/IntakeService from WSDL: xsr:/content/Lab2-1.0.0.jar/IntakeService.wsdl  
04:19:25,616 INFO [org.apache.cxf.endpoint.ServerImpl] (MSC service thread 1-2) Setting the server's publish address to be http://localhost:8080/homeLoan/IntakeService  
04:19:25,622 INFO [org.jboss.ws.cxf.deployment] (MSC service thread 1-2) JWSD020407: WSDL published to file:/home/gjpeles/rxstuf/jboss-tsx14/standalone/data/wsdl/homeLoan-deployment/IntakeService.wsdl  
04:19:25,627 INFO [org.jboss.ws.webservices] (MSC service thread 1-4) JBAS0205030: Starting service {jboss.ws.port:component-link  
04:19:25,628 INFO [org.jboss.ws.webservices] (MSC service thread 1-4) JBAS0205030: Starting service {jboss.ws.endpoint:'homeLoan-deployment'/IntakeService  
04:19:25,630 INFO [org.jboss.ws.common.management] (MSC service thread 1-4) JWSD022050: Endpoint registered: {jboss.ws.component:'homeLoan-deployment'/IntakeService  
04:19:25,783 INFO [org.switchyard.common.camel.SwitchyardCamelContext] (MSC service thread 1-2) Route: direct((urn:jboss:lan:1.0)/IntakeJMS started and consuming from: Endpoint[direct://A/Burn/homeLoan:1.0/IntakeJMS])  
04:19:25,836 INFO [org.switchyard.common.camel.SwitchyardCamelContext] (MSC service thread 1-2) Route: vicamelBindingModel(IntakeJMS)[JWSD020509 started and consuming from: Endpoint[jms://queue/LanIntakeConnectionFactory?connection  
factory=IntakeFactory]  
04:19:25,942 INFO [org.jboss.as.server] (DeploymentScanner-threads - 1) JBAS016099: Deployed 'Lab2-1.0.0.jar' (runtime-name: 'Lab2-1.0.0.jar')
```

Complete Stage Task

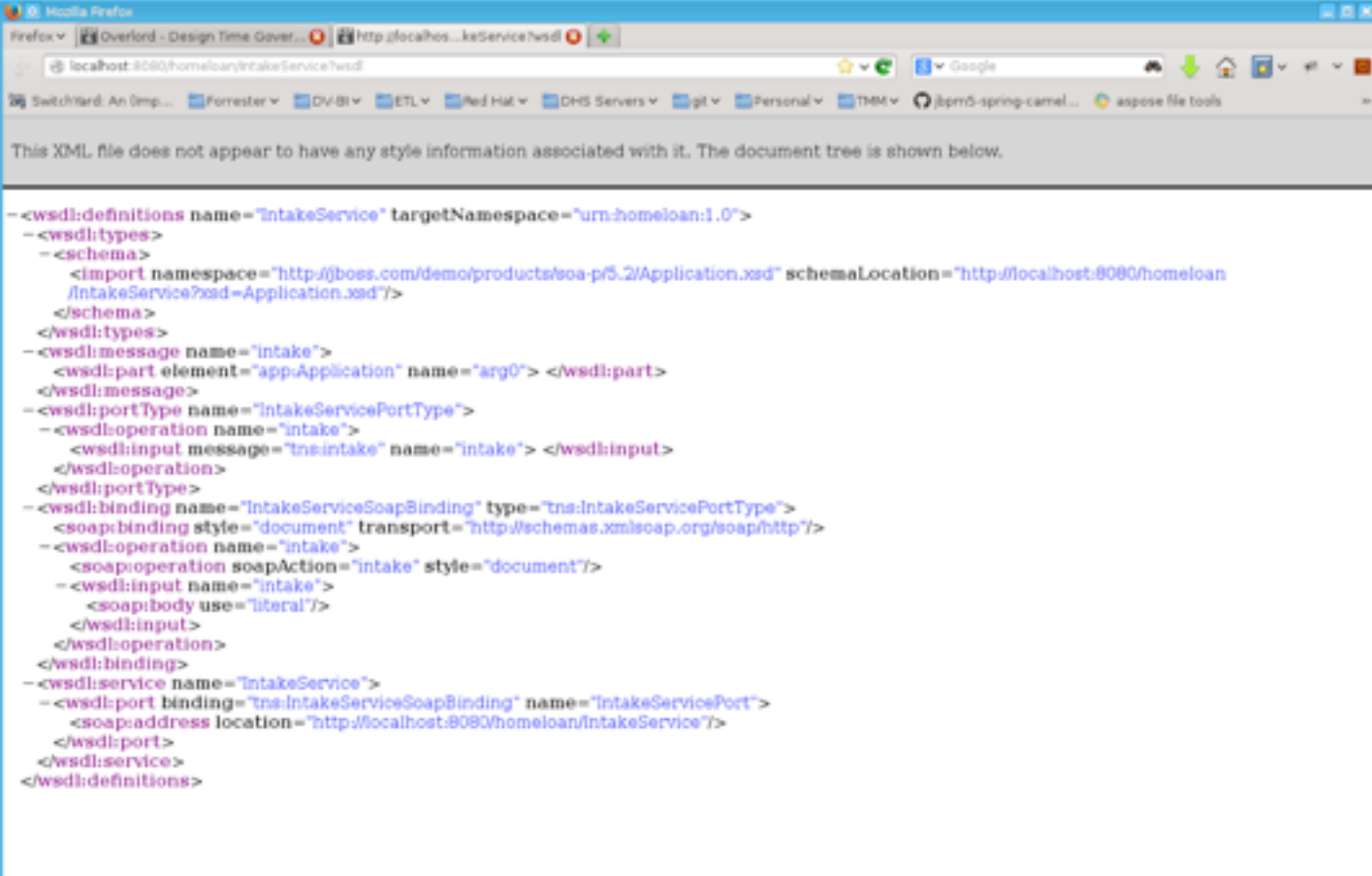
FYI

Our Home Loan Application is now available to test and complete our prod task.

TODO

1. In a new browser tab visit the SOAP endpoint URL to confirm the endpoint is available:

<http://localhost:8080/homeloan/IntakeService?wsdl>



```
<?xml version='1.0'?>
<wsdl:definitions name="IntakeService" targetNamespace="urn:homeloan:1.0">
  <wsdl:types>
    <schema>
      <import namespace="http://boss.com/demo/products/soa-p/5.2/Application.xsd" schemaLocation="http://localhost:8080/homeloan/IntakeService?xsd=Application.xsd"/>
    </schema>
  </wsdl:types>
  <wsdl:message name="Intake">
    <wsdl:part element="app:Application" name="arg0"/>
  </wsdl:message>
  <wsdl:portType name="IntakeServicePortType">
    <wsdl:operation name="Intake">
      <wsdl:input message="tns:Intake" name="Intake"/>
    </wsdl:operation>
  </wsdl:portType>
  <wsdl:binding name="IntakeServiceSoapBinding" type="tns:IntakeServicePortType">
    <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http"/>
    <wsdl:operation name="Intake">
      <soap:operation soapAction="Intake" style="document"/>
      <wsdl:input name="Intake">
        <soap:body use="literal"/>
      </wsdl:input>
    </wsdl:operation>
  </wsdl:binding>
  <wsdl:service name="IntakeService">
    <wsdl:port binding="tns:IntakeServiceSoapBinding" name="IntakeServicePort">
      <soap:address location="http://localhost:8080/homeloan/IntakeService"/>
    </wsdl:port>
  </wsdl:service>
</wsdl:definitions>
```

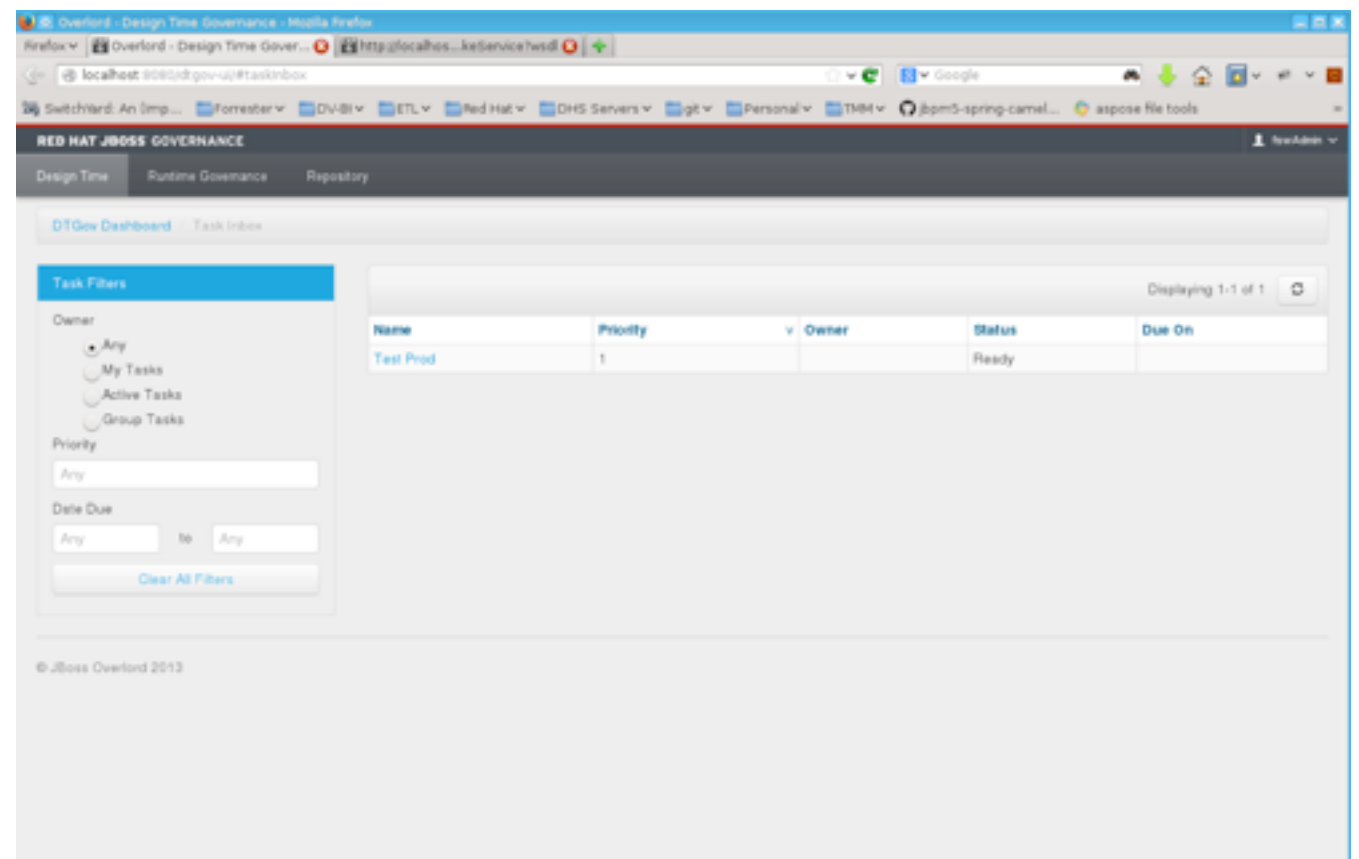
Complete Prod Task

FYI

We want to move the deployment through Production – Claim, Start, Pass/Fail and Complete.

TODO

1. Return to the Task Inbox.
2. Repeat the steps you followed for Stage



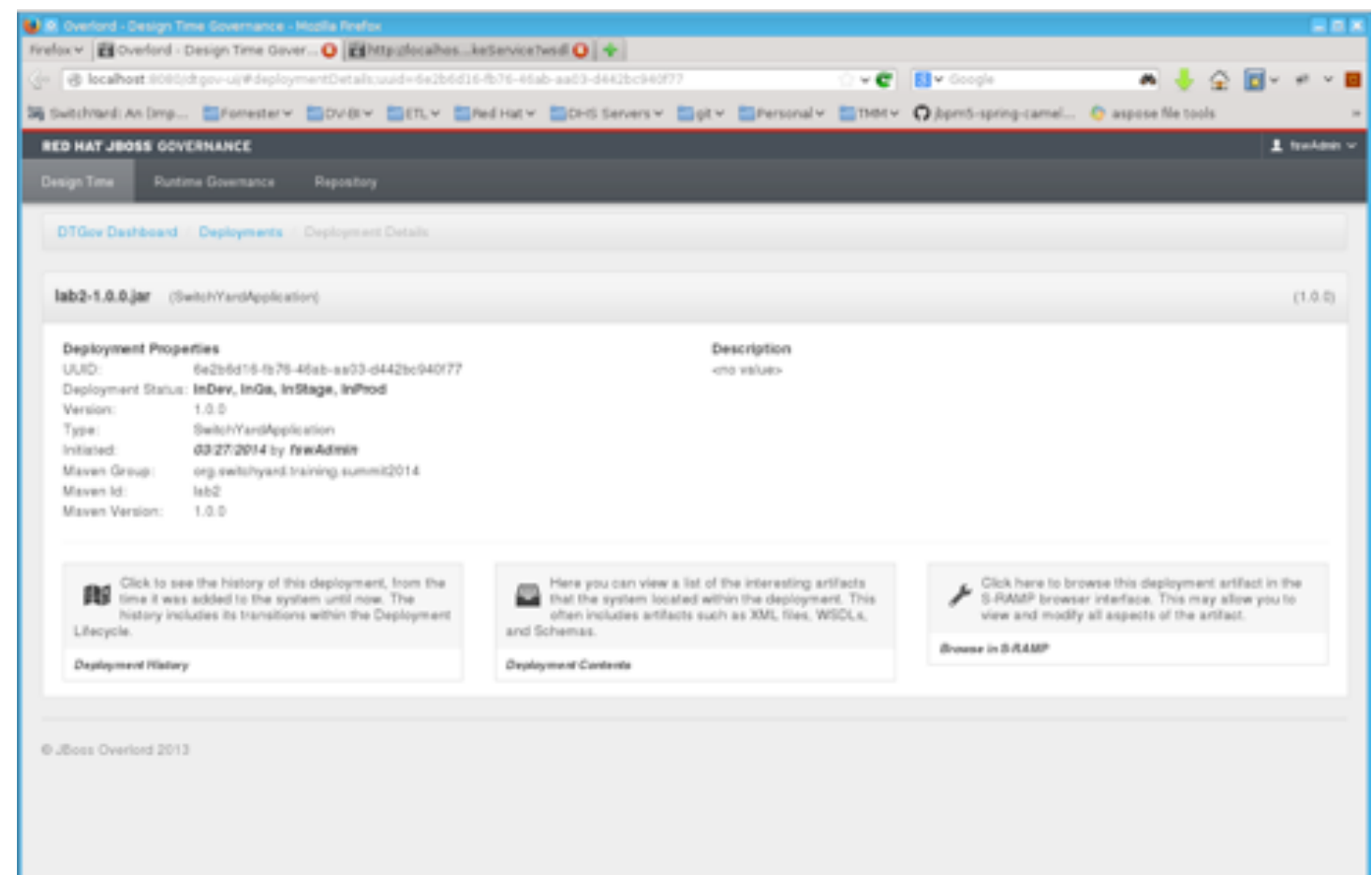
DTGov Deployment

FYI

You can view the deployment details, the deployment history and the deployment contents as well as browse S-RAMP.

TODO

1. Return to the DTGov Dashboard
2. Click on the Deployments button
3. Click on the lab2 deployment



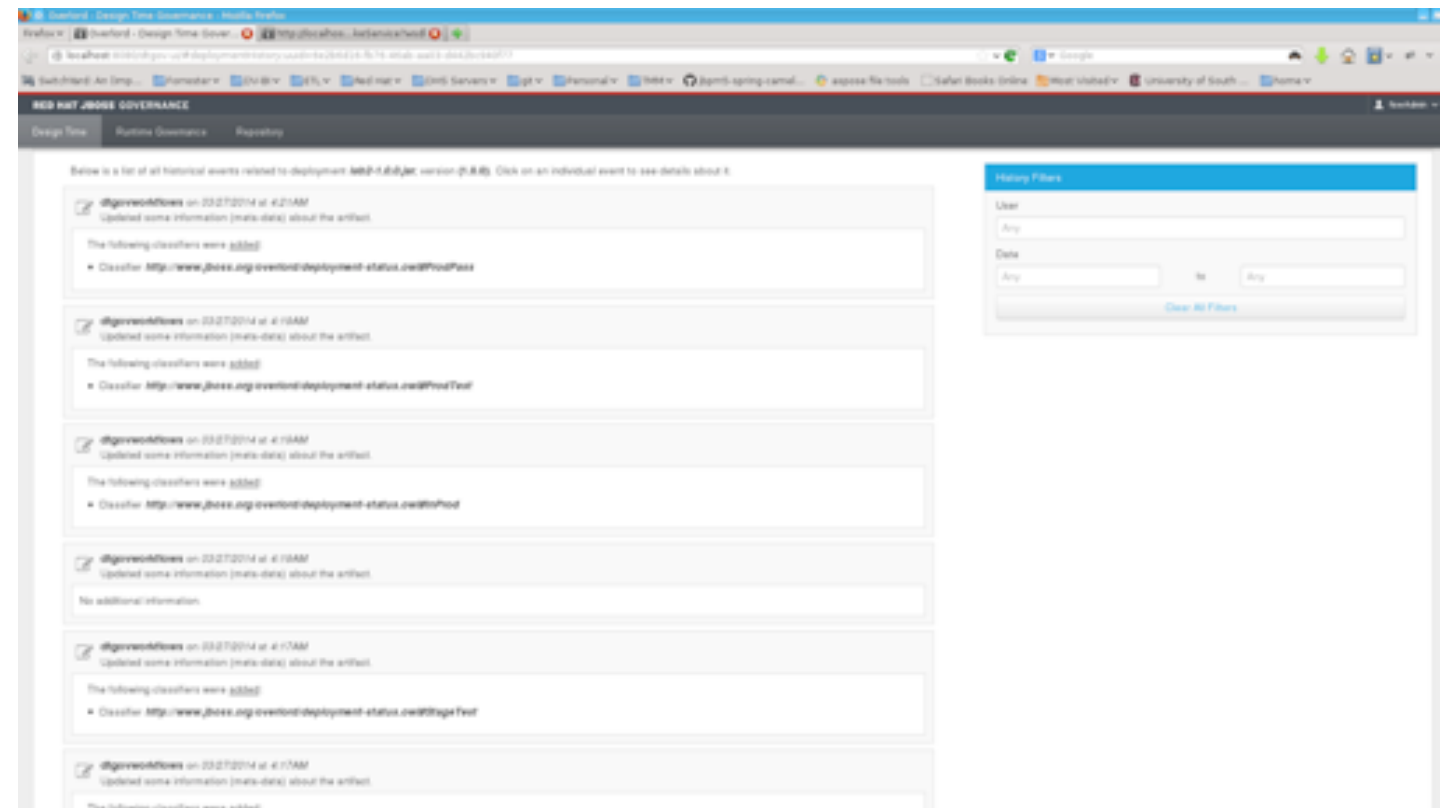
DTGov Deployment History

FYI

You can see the history of the deployment from the initial add until the current time. It includes the transitions within the deployment lifecycle.

TODO

1. Click on the Deployment History button



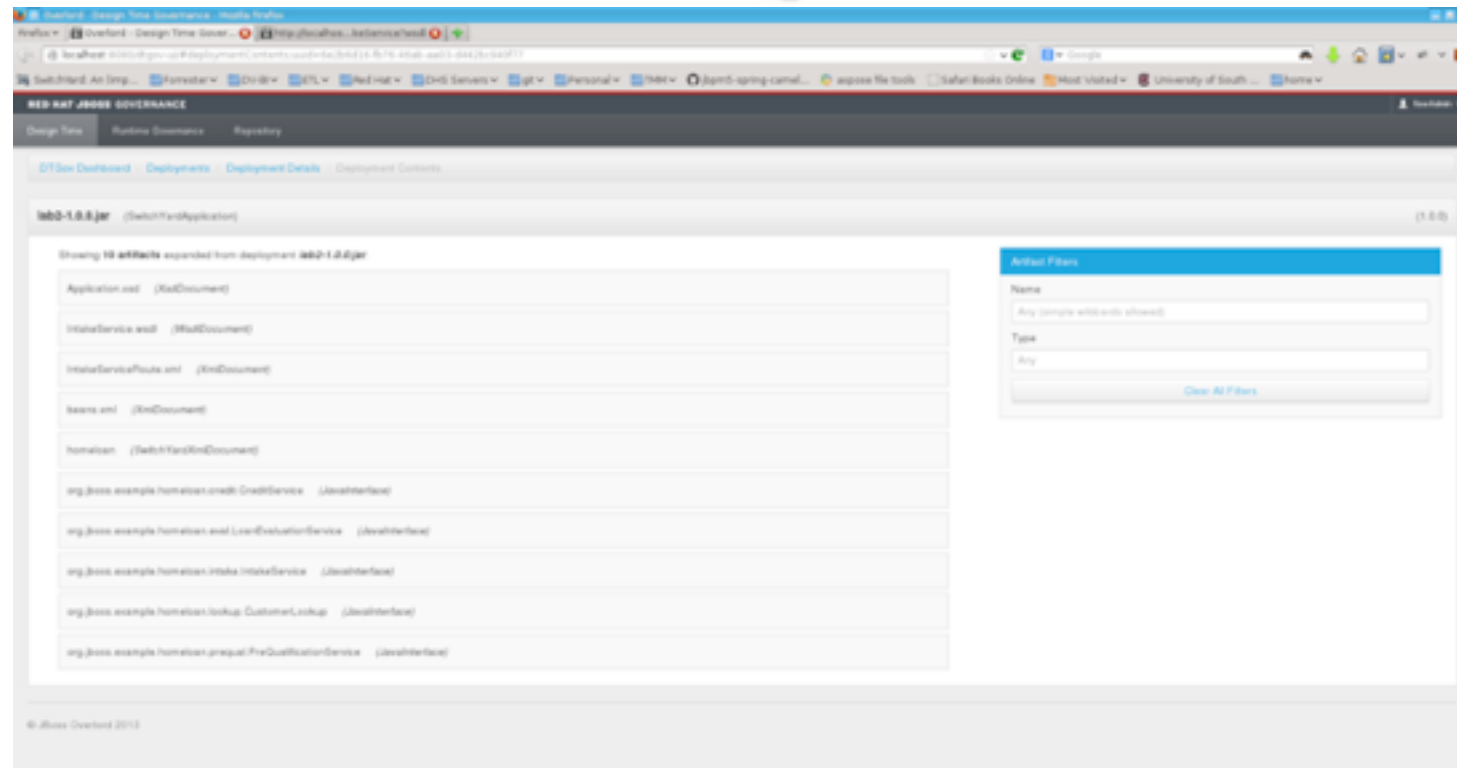
DTGov Deployment Contents

FYI

You can view the contents of the deployment and the artifact types. When clicking on an artifact the detail in S-RAMP will be shown.

TODO

1. Click on the deployment details link
2. Click on the deployment contents button



Lab 2 Complete!