Java Cloud Service Demo Guide: Side-by-side deployment

Description

For Use By	Oracle Sales / Pre-Sales, Consulting, Product Managers, Solution Managers, etc
Internal/External Consumption	Internal

Target Audience	• Customer / Partner
	  ○ Architects,
	  ○ Development Leads

Demo Duration (Approximate)	Approximately 10 minutes, plus the time needed for Oracle Cloud to provision your service instance. (Provisioning is the process of allocating cloud resources for your service instance.)
Background Knowledge Required/Suggested	• Maven • WLS architecture • WLST • Git

Related Materials	• Oracle Public Cloud Services
	  ○ Java Cloud Services
	  ○ Database Cloud Services
	• Git or Downloaded Sources
	• Java Development Kit 1.8+
	• Apache Maven 3.2.5+

Feature Overview

Weblogic Server supports a feature called side-by-side deployment (or versioned deployment). This function is extremely useful when you need to deploy a new version of an application and still keep the old one up and running. So the running instances will still use the current version and all new instances will be able to invoke the new deployed version of the application.

As soon as all the sessions that are using the old version of the application are expired, Weblogic will recognize it and will deactivate the old version. So at this moment only the new deployed version is active and all new sessions will make us of it.

This is often referred to as Production Redeployment too.

Demo Overview

So how does this work?

You could either use shell scripting (WLST or weblogic.deployer) to deploy the application and specify the version number on deployment as one of the parameters or you can use the manifest.mf file. We'll be using the first one.

A viewlet of the same is available here as well.

We would be deploying V1 of an application, using it for some time. Then deploy V2 in Administration mode for the Admin / Developer to verify proper the functionality. Then V2 would be promoted. The sessions which are already logged in will be using the previous version (V1) of the application and the new sessions created will use the new version of application (V2). And once all the sessions that were using V1 are expired, it would finally be deactivated.

Requirements / Prerequisites

- Windows or Linux operating system
- Computer with 2 GB RAM
- Java Development Kit 1.8+
- Apache Maven 3.2.5
- Git
- A web browser is required for working with Oracle Cloud services. These are the minimum requirements:
  - Microsoft Internet Explorer 9 or 10 with Browser Mode and Document Mode set to IE9 or IE10
  - Mozilla Firefox 24 and later
  - Google Chrome 29 and later
  - Apple Safari 6 and later
- Roles and Credentials - You must have the following roles and credentials:
  - Roles and privileges described in Oracle Cloud User Roles and Privileges in Getting Started with Oracle Cloud
  - Log-in credentials for creating Oracle Java Cloud Service instances and for working with the associated database and storage services.
  - Java Administrator role (This role lets you create Oracle Java Cloud Service instances.)
  - For more information, see About Oracle Java Cloud Service Roles and User Accounts in Using Oracle Java Cloud Service.
- Existing Database and Java Cloud Services.

Tips

- Use FireFox for the demos.

Demo Steps

Setup & Dependencies

1. Install Java Development Kit and Apache Maven
2. Install Java Development Kit and Apache Maven
   The SSH connection requires advanced Unlimited Strength Java Cryptography Extension which is not enabled by default due to import control restrictions of some countries. Please download the necessary extension policy files (local_policy.jar, US_export_policy.jar) and copy it to your <java-home>/lib/security
   For detailed information see the instructions attached to policy files.
3. Follow the steps outlined in WInS - Demo Guides - Setup – Clone Git repository to get the necessary sources ready.
4. To create necessary (JCS and DBCS) environment follow the **WInS - Demo Guides - JCS Setup.pdf**
5. Environment variables and directories
   a. This document will refer to the local clone of WInS remote repository as **WINS_SOURCE_REPOSITORY**. Every steps where it is used replace to your location.

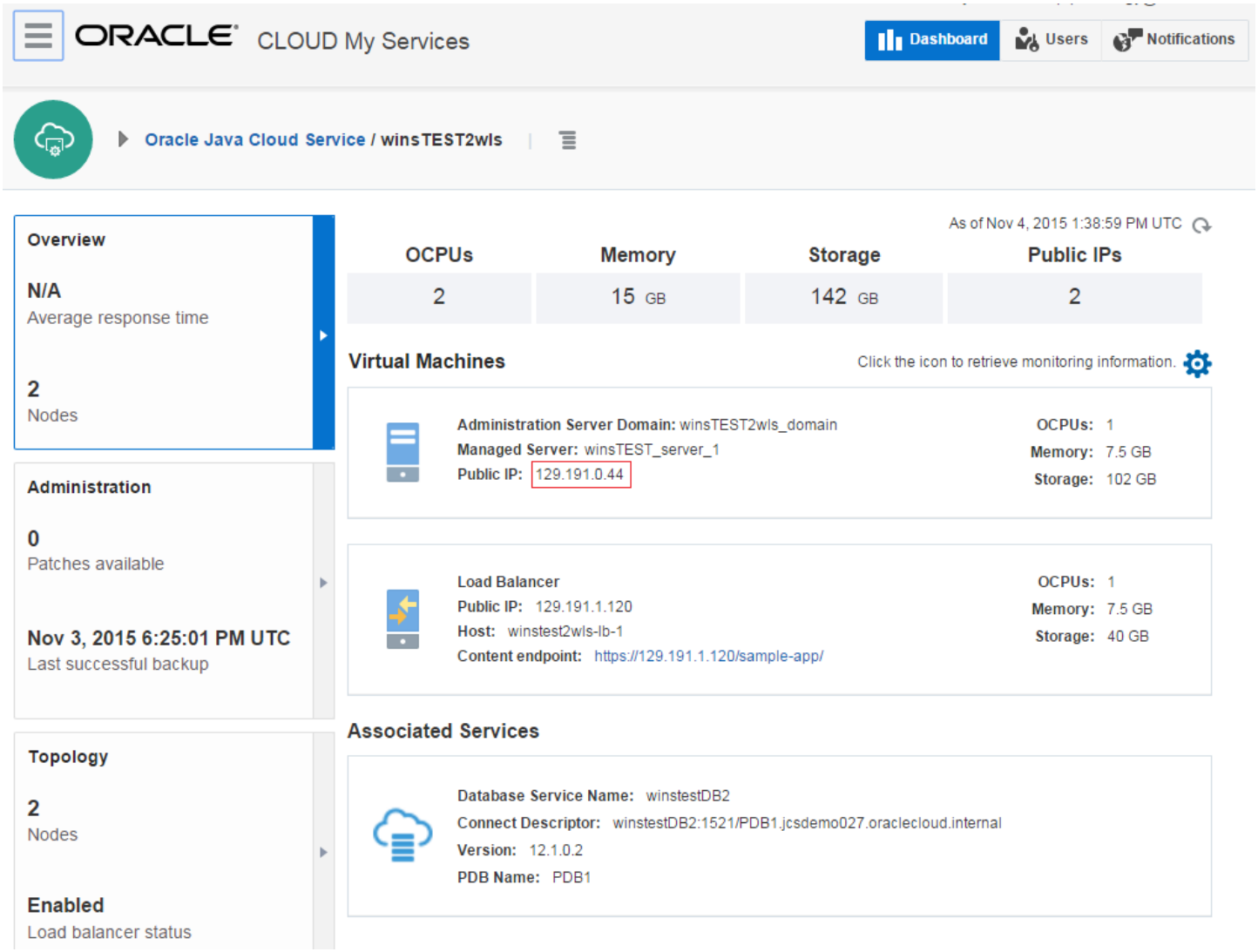
b. Set JAVA_HOME to point your jdk location. Example:
   Windows: `set JAVA_HOME=c:\jdk1.8.0_31`
   Linux: `export JAVA_HOME=/usr/java/jdk1.8.0_31`
c. Add Apache Maven to your PATH variable. In this case it is easier to execute maven. Example:
   Windows: `set PATH=%PATH%;c:\apache-maven-3.2.5\bin`
   Linux: `export PATH=$PATH:/usr/apache-maven-3.2.5/bin`

Executing the Side by Side Deployment demo

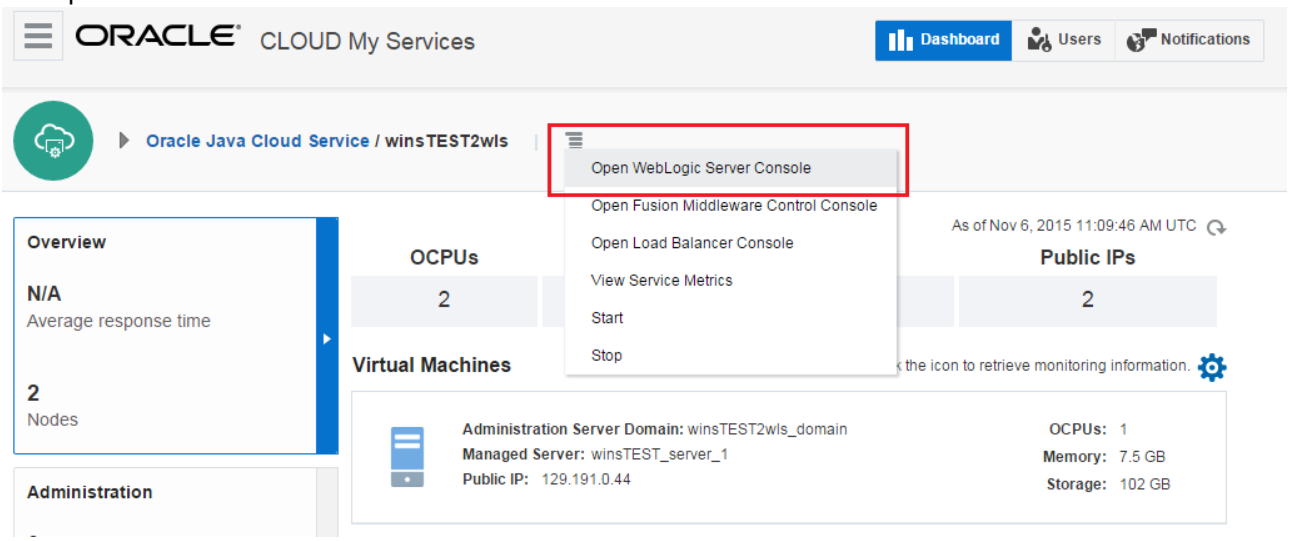
1) Get the public IP address of your Java Cloud Service Instance Administration Server Java. To obtain this information there are two options.
   a. Open terminal/command prompt and change directory to `WINS_SOURCE_REPOSITORY\cloud.demos`
      Execute the maven build with the following options:
      `mvn install -Dgoal=jcs-get-instance-details`
      The result output will show the IP address:

```
C:\Windows\System32\cmd.exe
c:\Java\git.repos\weblogic-innovation-seminars.cloud\cloud.demos>mvn install -DexecuteCloudUtil -Dgoal=jcs-get-instance-details
[INFO] Scanning for projects...
[INFO] ------------------------------------------------------------------------
[INFO] Reactor Build Order:
[INFO]
[INFO] wins-cloud
[INFO] cloud-api
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building wins-cloud 1.0.0-SNAPSHOT
[INFO] ------------------------------------------------------------------------
[INFO]
[INFO] --- maven-install-plugin:2.4:install (default-install) @ wins-cloud ---
[INFO] Installing c:\Java\git.repos\weblogic-innovation-seminars.cloud\cloud.demos\pom.xml to C:\Users\pnagy\.m2\repository\com\oracle\wins\cloud\wins-cloud\1.0
.0-SNAPSHOT\wins-cloud-1.0.0-SNAPSHOT.pom
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building cloud-api 1.0.0-SNAPSHOT
[INFO] ------------------------------------------------------------------------
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ cloud-common ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ cloud-common ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ cloud-common ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ cloud-common ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ cloud-common ---
[INFO] Tests are skipped.
[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ cloud-common ---
[INFO]
[INFO] --- maven-install-plugin:2.4:install (default-install) @ cloud-common ---
[INFO] Installing C:\Java\git.repos\weblogic-innovation-seminars.cloud\cloud.demos\common\target\cloud-common.jar to C:\Users\pnagy\.m2\repository\com\oracle\wi
ns\cloud\cloud-common\1.0.0-SNAPSHOT\cloud-common-1.0.0-SNAPSHOT.jar
[INFO] Installing C:\Java\git.repos\weblogic-innovation-seminars.cloud\cloud.demos\common\pom.xml to C:\Users\pnagy\.m2\repository\com\oracle\wins\cloud\cloud-c
ommon\1.0.0-SNAPSHOT\cloud-common-1.0.0-SNAPSHOT.pom
[INFO]
[INFO] --- maven-antrun-plugin:1.8:run (first) @ cloud-common ---
[INFO] Executing tasks

main:
     [java] Read all properties from file: environment.properties
     [java] Selected goal: jcs-get-instance-details
     [java] JCS get specific instance details----------------------------------------
     [java] Auth: {<any realm>@jaas.oraclecloud.com:443=[principal: peter.nagy@oracle.com]}
     [java] URI: http://jaas.oraclecloud.com/paas/service/jcs/api/v1.1/instances/jcsdemo027/winsTEST2wls
     [java] Executing request GET http://jaas.oraclecloud.com/paas/service/jcs/api/v1.1/instances/jcsdemo027/winsTEST2wls HTTP/1.1
     [java] Response: HTTP/1.1 200 OK
     [java] Output from Server ....
     [java]
     [java] {
     [java]   "service_name": "winsTEST2wls",
     [java]   "version": "12cR3",
     [java]   "wlsVersion": "12.1.3.0.4",
     [java]   "status": "Running",
     [java]   "error_status_desc": "",
     [java]   "compliance_status": "",
     [java]   "compliance_status_desc": "",
     [java]   "auto_update": "true",
     [java]   "description": "winstest instance through REST",
     [java]   "identity_domain": "jcsdemo027",
     [java]   "creation_time": "Mon Nov 2 5:55:53 UTC 2015",
     [java]   "last_modified_time": "Mon Nov 2 5:55:52 UTC 2015",
     [java]   "created_by": "peter.nagy@oracle.com",
     [java]   "service_uri": "https:\/\/jaas.oraclecloud.com\/paas\/service\/jcs\/api\/v1.1\/instances\/jcsdemo027\/winsTEST2wls",
     [java]   "domain_name": "winsTEST2wls_domain",
     [java]   "domain_mode": "PRODUCTION",
     [java]   "cluster_name": "winsTEST2wls_cluster",
     [java]   "num_nodes": 2,
     [java]   "level": "PAAS",
     [java]   "subscription_type": "HOURLY",
     [java]   "edition": "EE",
     [java]   "shape": "oc3",
     [java]   "otd_provisioned": "yes",
     [java]   "otd_shape": "oc3",
     [java]   "otd_storage_size": 40960,
     [java]   "service_components": [
     [java]     {
     [java]       "type": "OTD_JDK",
     [java]       "version": "1.7.0_85"
     [java]     },
     [java]     {
     [java]       "type": "JDK",
     [java]       "version": "1.7.0_85"
     [java]     },
     [java]     {
     [java]       "type": "OTD",
     [java]       "version": "11.1.1.9.0"
     [java]     },
     [java]     {
     [java]       "type": "WLS",
     [java]       "version": "12.1.3.0.4"
     [java]     }
     [java]   ],
     [java]   "creation_job_id": "527525",
     [java]   "deletion_job_id": 0,
     [java]   "db_info": "winstestDB2:1521\/PDB1.jcsdemo027.oraclecloud.internal",
     [java]   "db_service_name": "winstestDB2",
     [java]   "num_ip_reservations": 2,
     [java]   "wls_admin_url": "https:\/\/129.191.0.44:7002\/console",
     [java]   "fmw_control_url": "https:\/\/129.191.0.44:7002\/em",
     [java]   "otd_admin_url": "https:\/\/129.191.1.120:8989",
     [java]   "sample_app_url": "https:\/\/129.191.1.120\/sample-app\/",
     [java]   "content_url": "http:\/\/129.191.1.120",
     [java]   "secure_content_url": "https:\/\/129.191.1.120",
     [java]   "wls_deployment_channel_port": 9001,
     [java]   "psm_plugin_version": "15.4.1-0-1510120238"
     [java] }
     [java]
[INFO] Executed tasks
[INFO] ------------------------------------------------------------------------
[INFO] Reactor Summary:
[INFO]
[INFO] wins-cloud ......................................... SUCCESS [  0.265 s]
[INFO] cloud-api .......................................... SUCCESS [  5.658 s]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 6.020 s
[INFO] Finished at: 2015-11-04T14:29:58+01:00
[INFO] Final Memory: 11M/227M
[INFO] ------------------------------------------------------------------------
c:\Java\git.repos\weblogic-innovation-seminars.cloud\cloud.demos>
```

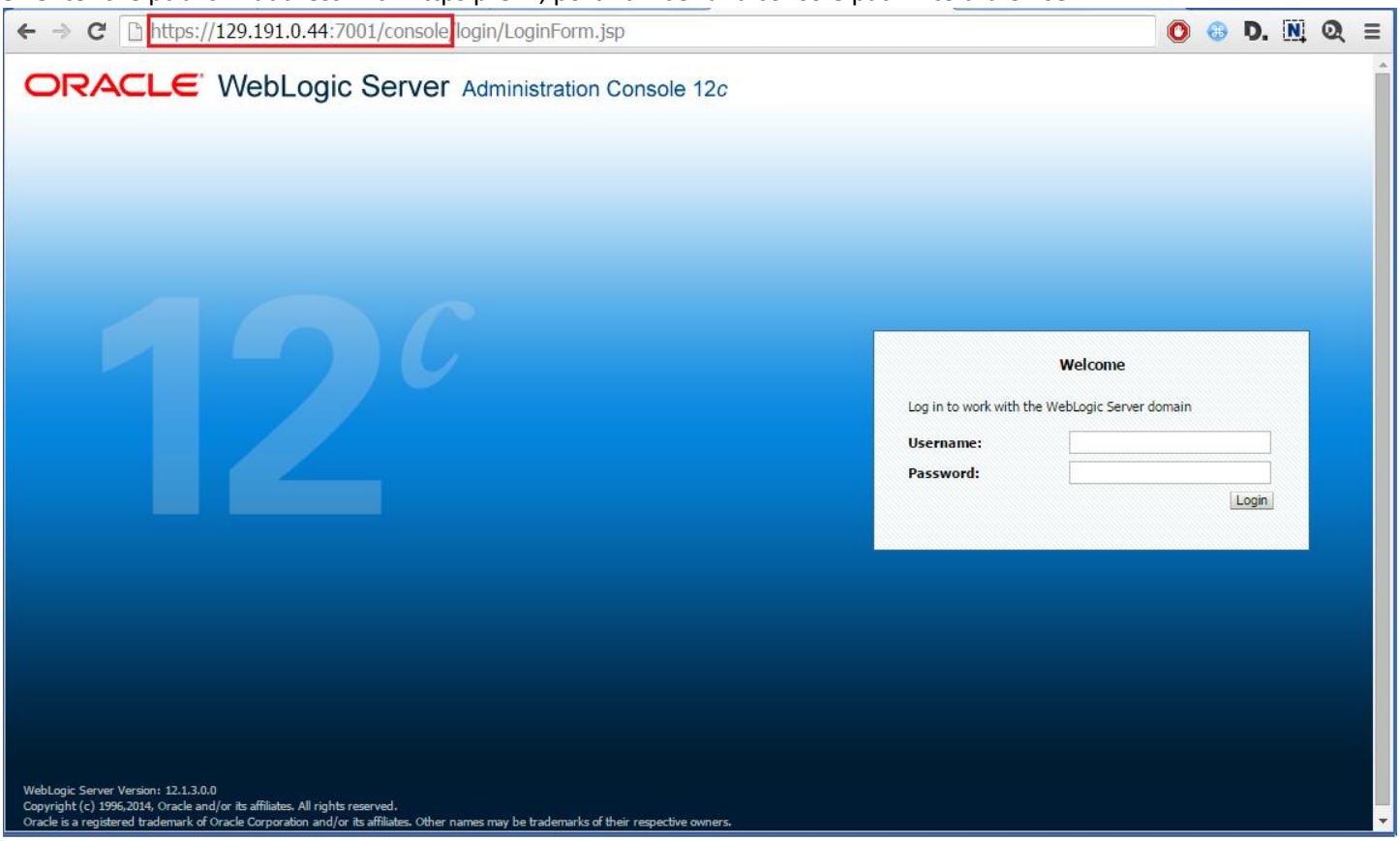
b. Or open the Java Cloud Service Console. Click on the desired JCS instance and you can see the public IP address.



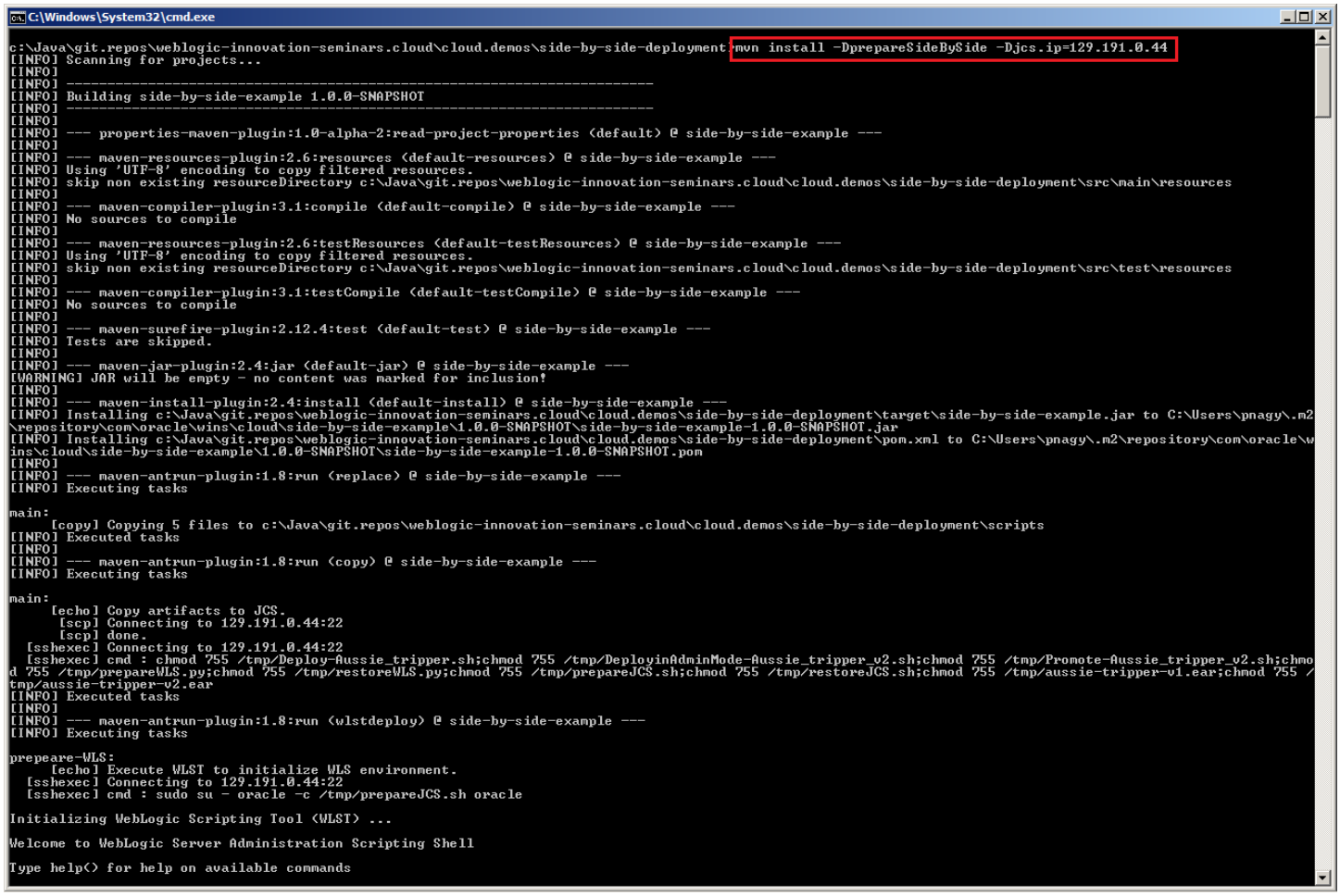
To check the current port availability –which will change below- open the Weblogic Admin Console. Click on the JCS console to open.



Or enter the public IP address with https prefix, port number and console path into a browser.



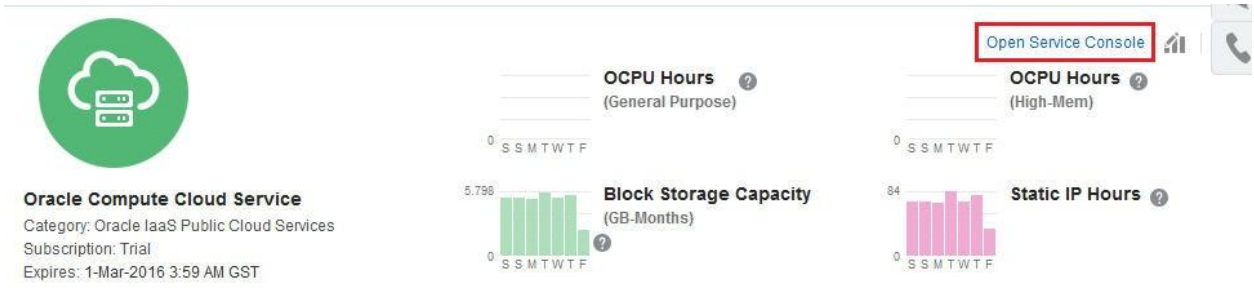
2) Setup and configure Java Cloud Service instance. Run the following maven build in the `WINS_SOURCE_REPOSITORY\cloud.demos\side-by-side-deployment` directory:
`mvn install -DprepareSideBySide -Djcs.ip=129.191.1.19`
The preparation includes:
   - Stopping Managed Servers to apply administration port settings
   - Enable administration port on 7200
   - Enable Tunneling for Admin server.


C:\Windows\System32\cmd.exe
c:\Java\git.repos\weblogic-innovation-seminars.cloud\cloud.demos\side-by-side-deployment>mvn install -DprepareSideBySide -Djcs.ip=129.191.0.44
[INFO] Scanning for projects...
[INFO] Building side-by-side-example 1.0.0-SNAPSHOT
[INFO] --- properties-maven-plugin:1.0-alpha-2:read-project-properties (default) @ side-by-side-example ---
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ side-by-side-example ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory c:\Java\git.repos\weblogic-innovation-seminars.cloud\cloud.demos\side-by-side-deployment\src\main\resources
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ side-by-side-example ---
[INFO] No sources to compile
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ side-by-side-example ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] skip non existing resourceDirectory c:\Java\git.repos\weblogic-innovation-seminars.cloud\cloud.demos\side-by-side-deployment\src\test\resources
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ side-by-side-example ---
[INFO] No sources to compile
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ side-by-side-example ---
[INFO] Tests are skipped.
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ side-by-side-example ---
[WARNING] JAR will be empty - no content was marked for inclusion!
[INFO] --- maven-install-plugin:2.4:install (default-install) @ side-by-side-example ---
[INFO] Installing c:\Java\git.repos\weblogic-innovation-seminars.cloud\cloud.demos\side-by-side-deployment\target\side-by-side-example.jar to C:\Users\pnagy\.m2\repository\com\oracle\wins\cloud\side-by-side-example\1.0.0-SNAPSHOT\side-by-side-example-1.0.0-SNAPSHOT.jar
[INFO] Installing c:\Java\git.repos\weblogic-innovation-seminars.cloud\cloud.demos\side-by-side-deployment\pom.xml to C:\Users\pnagy\.m2\repository\com\oracle\wins\cloud\side-by-side-example\1.0.0-SNAPSHOT\side-by-side-example-1.0.0-SNAPSHOT.pom
[INFO] --- maven-antrun-plugin:1.8:run (replace) @ side-by-side-example ---
[INFO] Executing tasks
main:
[copy] Copying 5 files to c:\Java\git.repos\weblogic-innovation-seminars.cloud\cloud.demos\side-by-side-deployment\scripts
[INFO] Executed tasks
[INFO] --- maven-antrun-plugin:1.8:run (copy) @ side-by-side-example ---
[INFO] Executing tasks
main:
[echo] Copy artifacts to JCS.
[scp] Connecting to 129.191.0.44:22
[scp] done.
[sshexec] Connecting to 129.191.0.44:22
[sshexec] cmd : chmod 755 /tmp/Deploy-Aussie_tripper.sh;chmod 755 /tmp/DeployinAdminMode-Aussie_tripper_v2.sh;chmod 755 /tmp/Promote-Aussie_tripper_v2.sh;chmod 755 /tmp/prepareWLS.py;chmod 755 /tmp/restoreWLS.py;chmod 755 /tmp/prepareJCS.sh;chmod 755 /tmp/restoreJCS.sh;chmod 755 /tmp/aussie-tripper-v1.ear;chmod 755 /tmp/aussie-tripper-v2.ear
[INFO] Executed tasks
[INFO] --- maven-antrun-plugin:1.8:run (wlstdeploy) @ side-by-side-example ---
[INFO] Executing tasks
prepeare-WLS:
[echo] Execute WLST to initialize WLS environment.
[sshexec] Connecting to 129.191.0.44:22
[sshexec] cmd : sudo su - oracle -c /tmp/prepareJCS.sh oracle
Initializing WebLogic Scripting Tool (WLST) ...
Welcome to WebLogic Server Administration Scripting Shell
Type help() for help on available commands

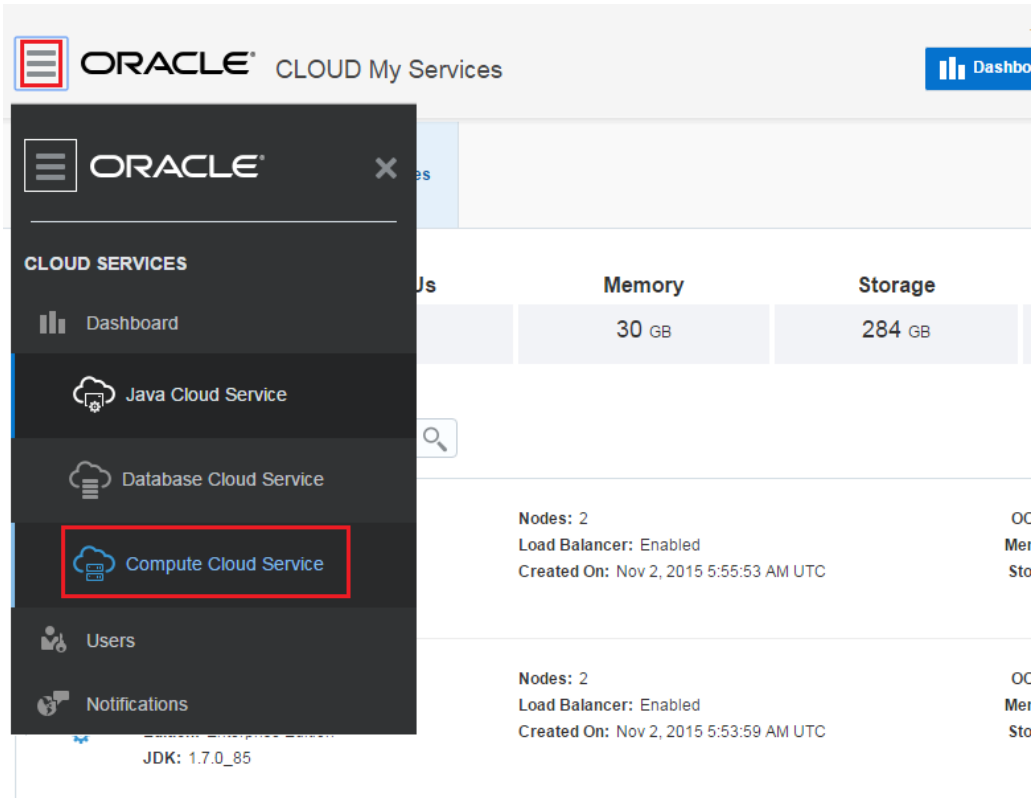
```
Connecting to t3://winsTEST2wls-wls-1.compute-jcsdemo027.oraclecloud.internal:7001 with userid weblogic ...
Successfully connected to Admin Server "winsTEST_adminserver" that belongs to domain "winsTEST2wls_domain".

Warning: An insecure protocol was used to connect to the
server. To ensure on-the-wire security, the SSL port or
Admin port should be used instead.

Get managed server list
Location changed to serverRuntime tree. This is a read-only tree with DomainMBean as the root.
For more help, use help('domainConfig')

Connecting to Node Manager ...
<Nov 6, 2015 10:13:58 AM UTC> <Info> <Security> <BEA-090905> <Disabling the CryptoJ JCE Provider self-integrity check for better startup performance. To enable
this check, specify -Dweblogic.security.allowCryptoJDefaultJCEVerification=true.>
<Nov 6, 2015 10:13:58 AM UTC> <Info> <Security> <BEA-090906> <Changing the default Random Number Generator in RSA CryptoJ from ECDRBG128 to FIPS186PRNG. To disa
ble this change, specify -Dweblogic.security.allowCryptoJDefaultPRNG=true.>
<Nov 6, 2015 10:13:58 AM UTC> <Info> <Security> <BEA-090909> <Using the configured custom SSL Hostname Verifier implementation: weblogic.security.utils.SSLWLSHo
stnameVerifier$NullHostnameVerifier.>
Successfully Connected to Node Manager.

RUNNING

winsTEST_adminserver status...RUNNING

RUNNING

winsTEST_server_1 status...RUNNING

RUNNING

winsTEST_server_1 is RUNNING
Stopping winsTEST_server_1
Killing server winsTEST_server_1 ...
Successfully killed server winsTEST_server_1

SHUTDOWN

Now winsTEST_server_1 status is SHUTDOWN
Successfully disconnected from Node Manager.
Location changed to edit tree. This is a writable tree with
DomainMBean as the root. To make changes you will need to start
an edit session via startEdit().

For more help, use help('edit')

Starting an edit session ...
Started edit session, please be sure to save and activate your
changes once you are done.
Saving all your changes ...
Saved all your changes successfully.
Activating all your changes, this may take a while ...
The edit lock associated with this edit session is released
once the activation is completed.
"
You will have to re-connect to the server due to some changes" "
you have made."
Disconnected from weblogic server: winsTEST_adminserver
[INFO] Executed tasks
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 21.287 s
[INFO] Finished at: 2015-11-06T11:14:02+01:00
[INFO] Final Memory: 17M/322M
[INFO] ------------------------------------------------------------------------
c:\Java\git.repos\weblogic-innovation-seminars.cloud\cloud.demos\side-by-side-deployment>_
```

Important: after the modification you cannot access to the Weblogic Admin Console through the known and preconfigured port 7002. When the administration port is enabled you can access through only that port using ssl connection. So move forward to enable external access to the newly defined administration port which is by default disabled.

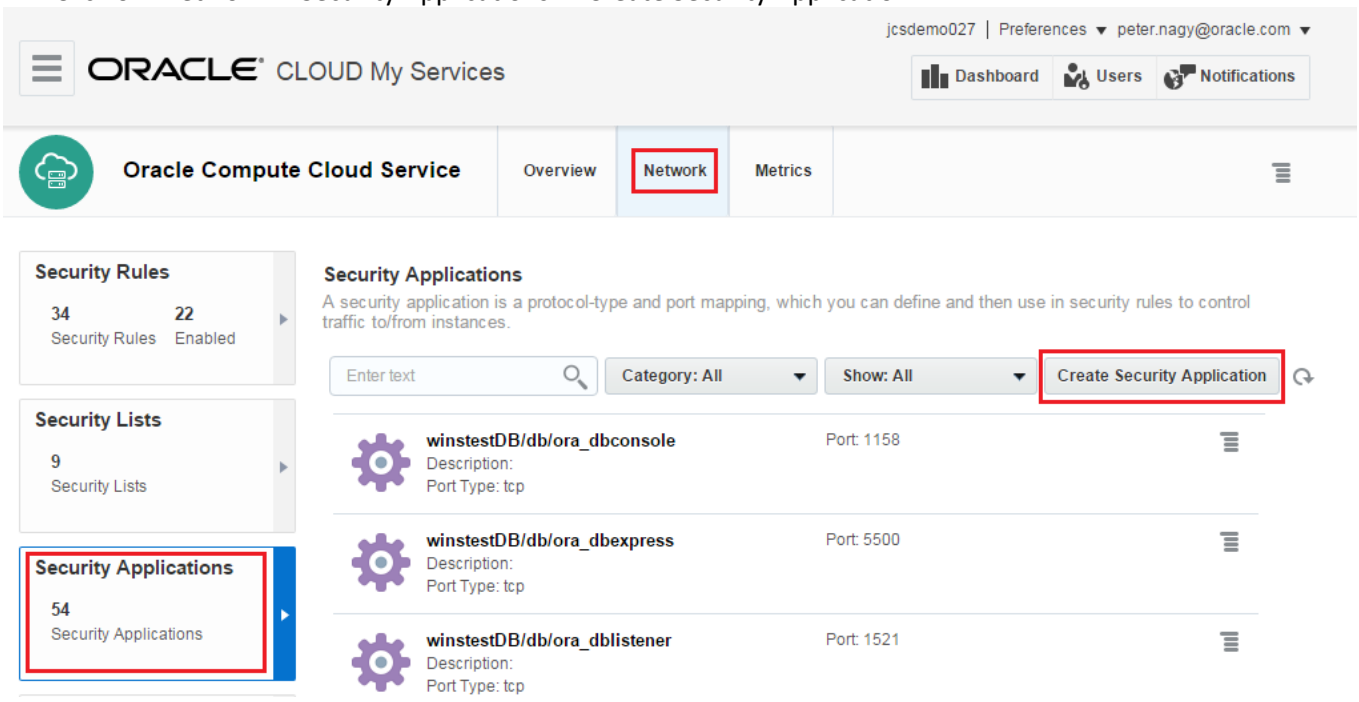
3) Now we need to enable a port on the JCS VM – so that external traffic can be channeled to the VM. Login to the Oracle Public Cloud and visit the Oracle Compute Cloud Service home page. From here click on 'Open Service Console'



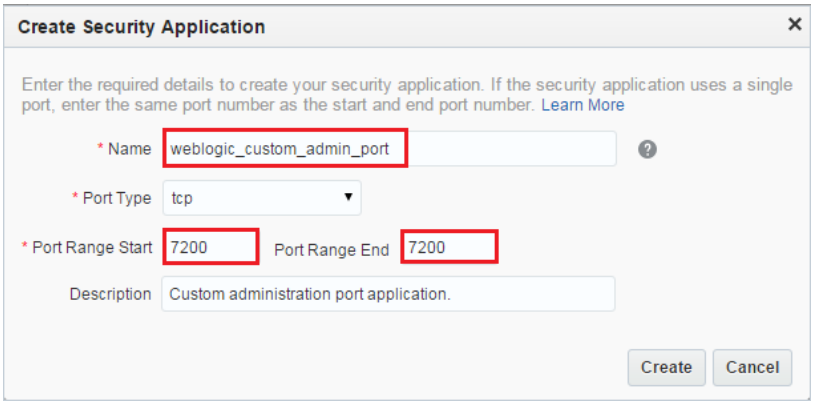
Or click on the console shortcut and select 'Compute Cloud Service'



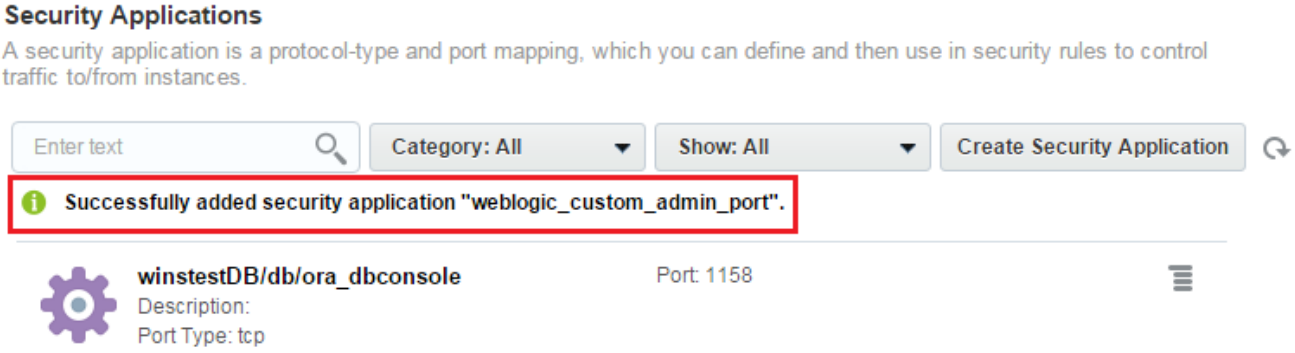
Click on 'Network -> Security Applications -> Create Security Application'



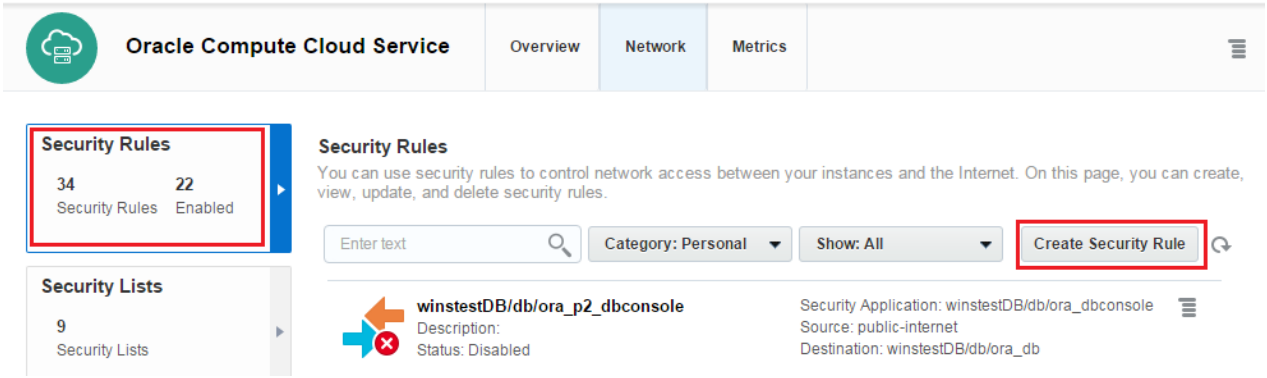
Now create a generic 'Security Application' which can enable tcp access through port 7200. Define the values as per mentioned in the image below:



Once the protocol is created – a success message is shown.

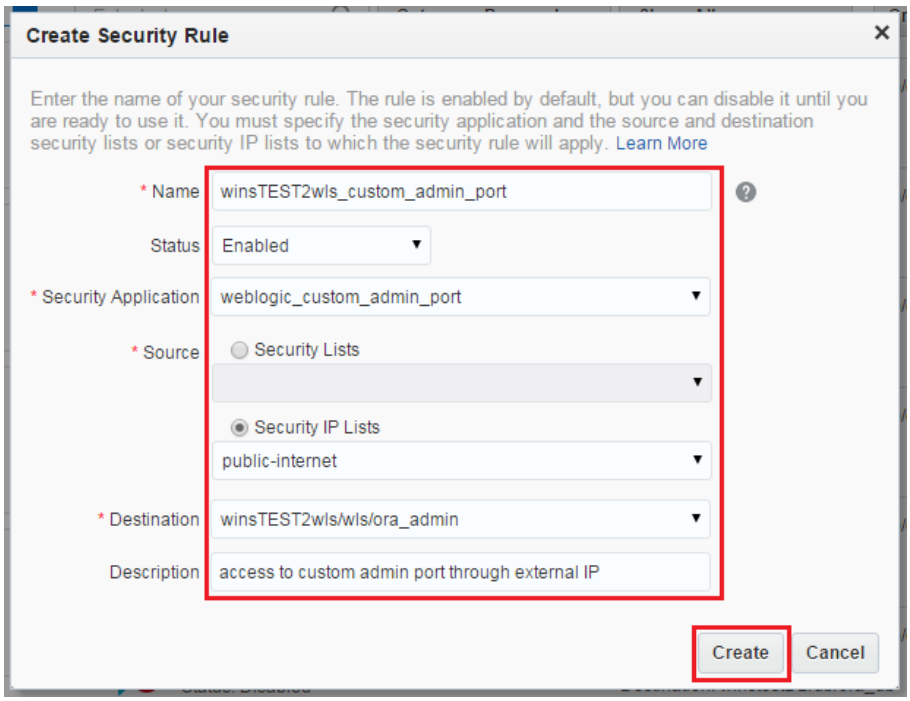


Now create the a new rule what will asssign the 'Security Application' to the desired instance. Click on 'Security Rules -> Create Security Rule'

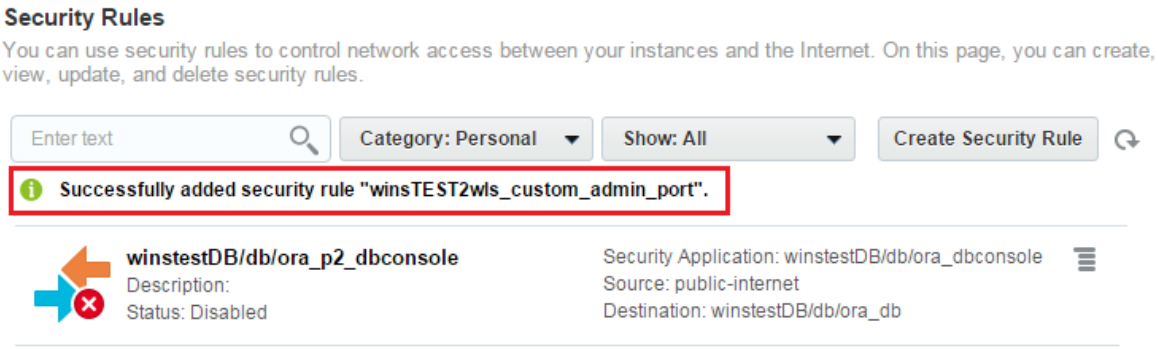


Define the values for the new rule:

- Name: choose what you want
- Status: **Enabled** (default)
- Security Application: select the previously created Security Application.
- Source: select **Security IP Lists** and the predefined **public-internet**
  Note: Here you can also create specific source address to restrict external access for more security.
- Destination: select the predefined **<JCS_INSTANCE>/wls/ora_admin**
- Description: choose what you want.

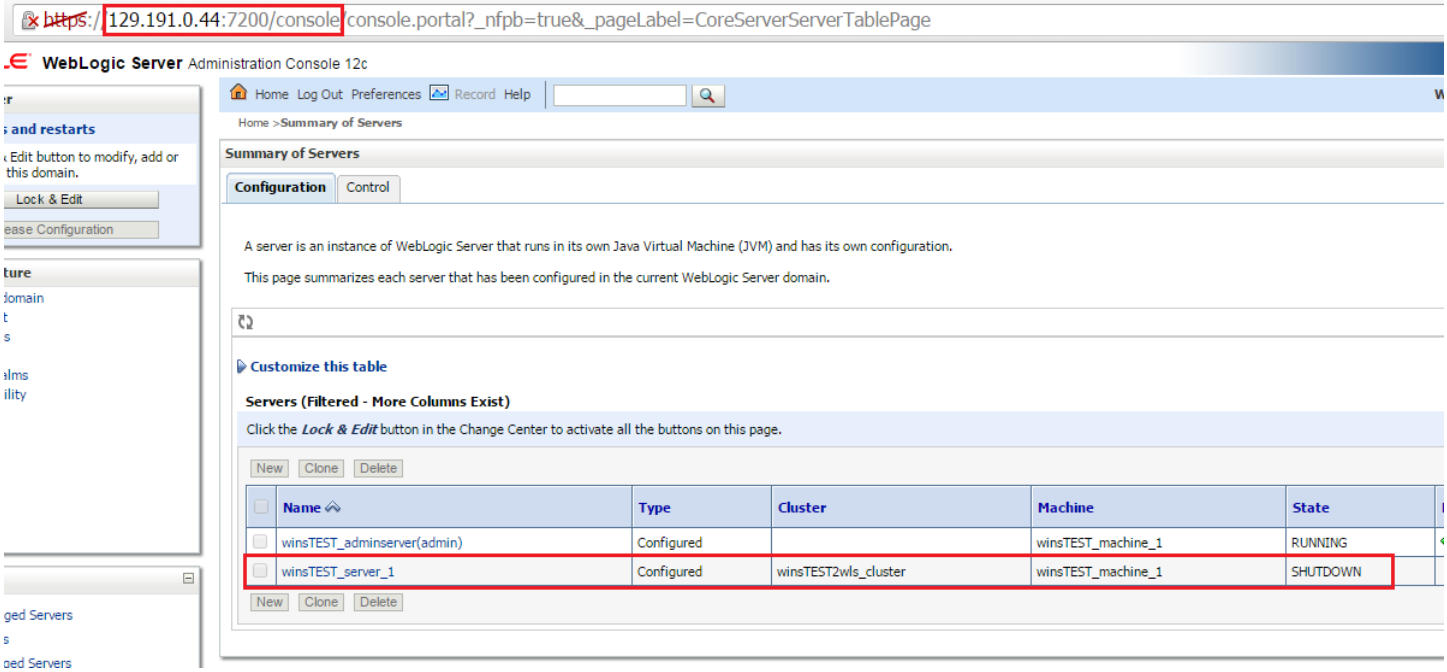


Once the Access Rule is created, a success message is shown.

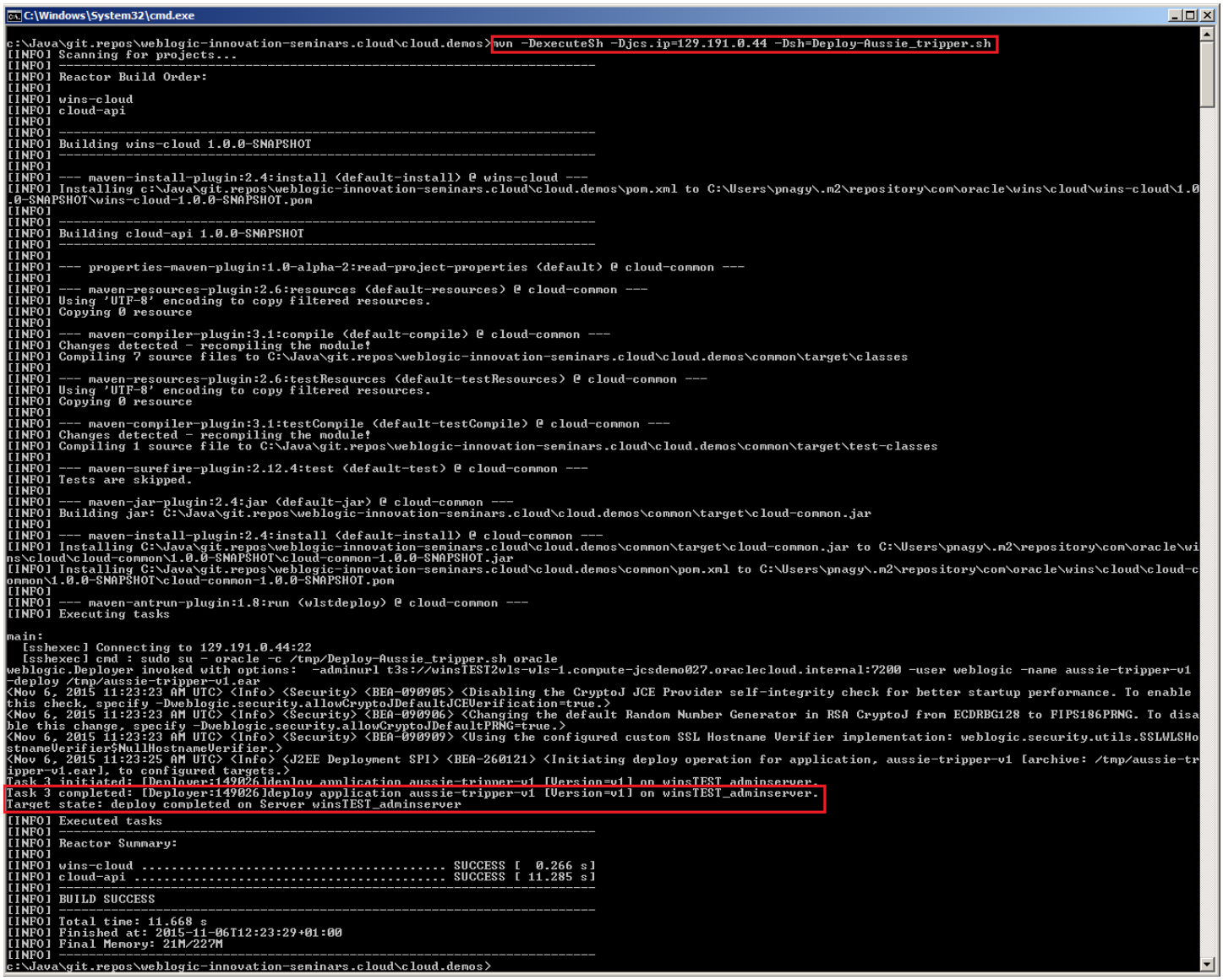


Now check the Weblogic Admin Console access through the new administration port. Don't try to open through JCS console because that will try through the 7002 port. You can check just for sure.

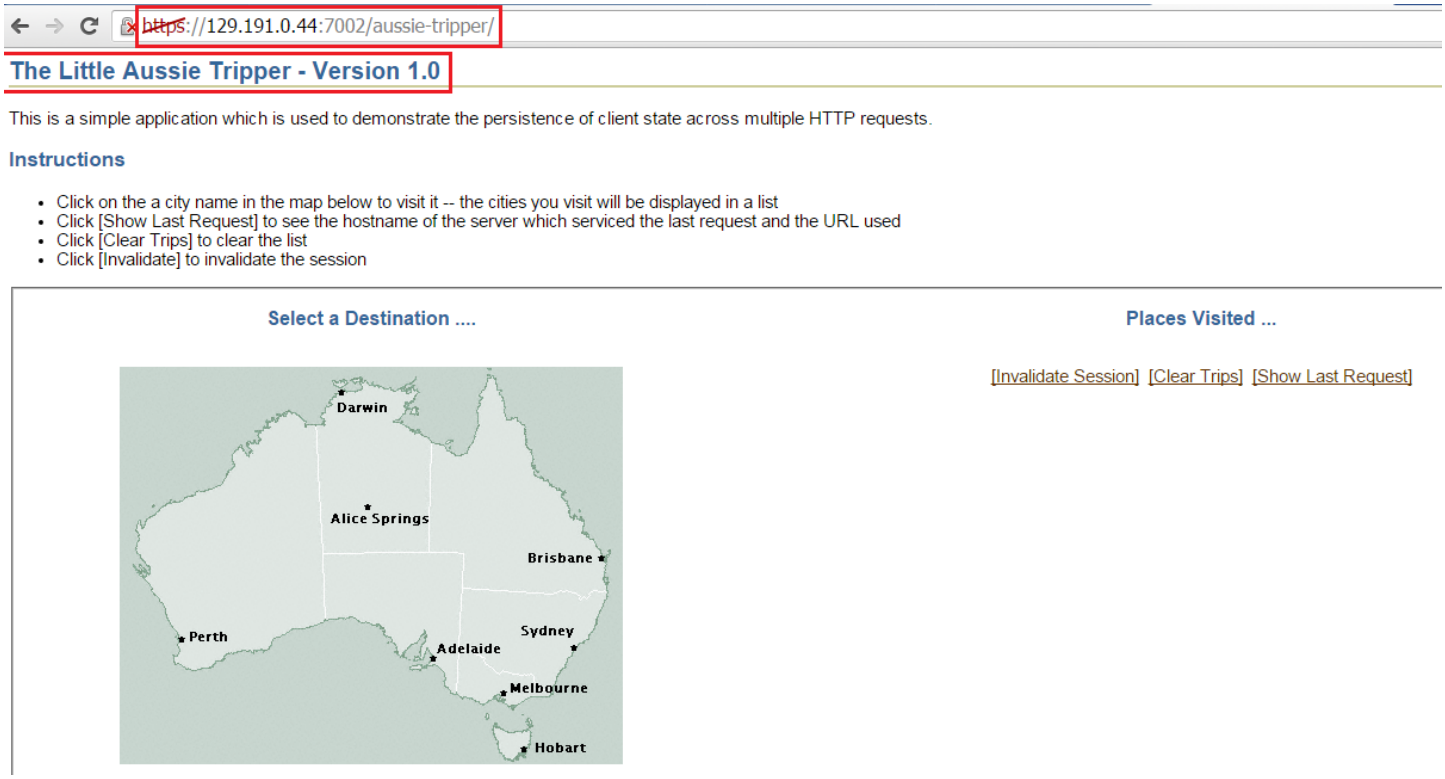
So change the port to 7200 in console url, accept the risks, add the exception and proceed to be able to access the WebLogic Server Admin Console. Optionally you can check the managed server's status. It should be shutdown.



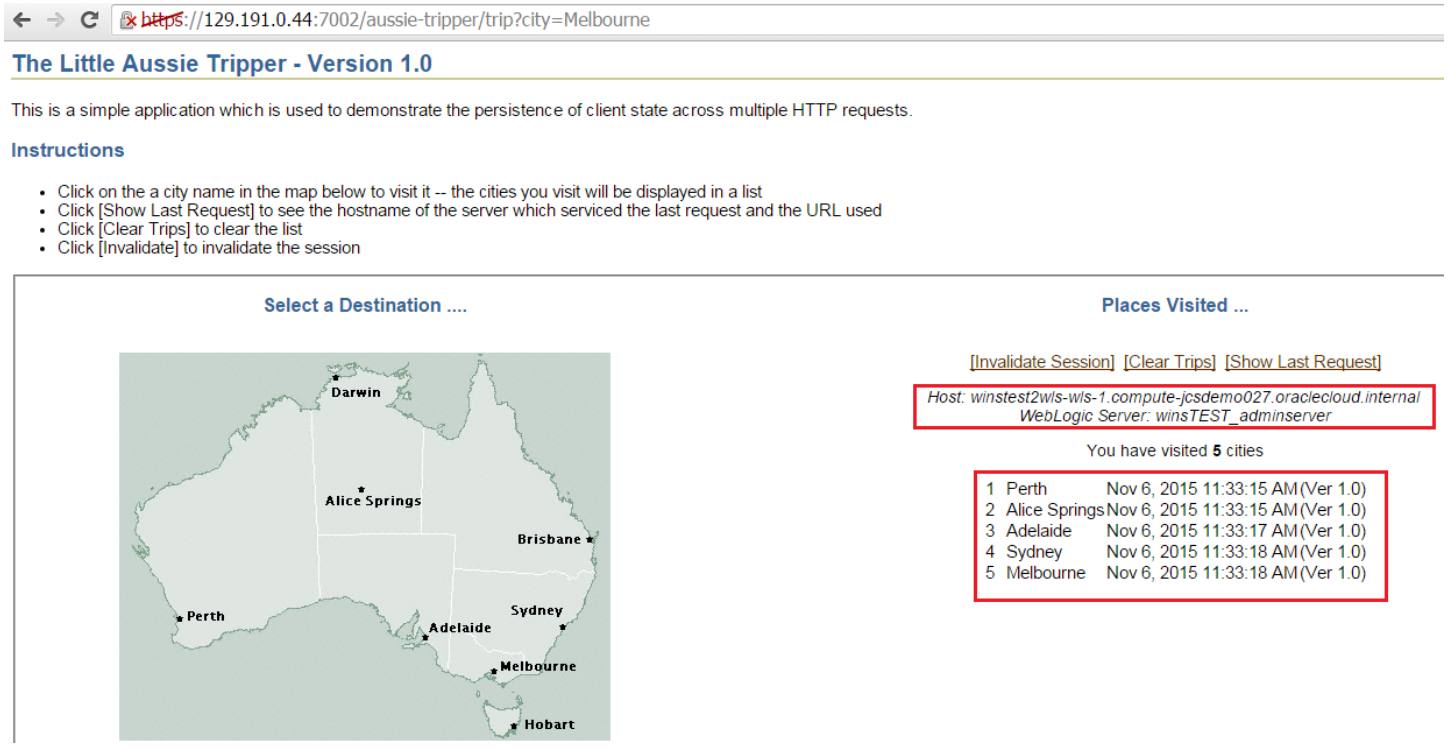
4) Now is the time to deploy the application. Execute the following maven build in the `WINS_SOURCE_REPOSITORY\cloud.demos` directory:
`mvn -DexecuteSh -Djcs.ip=129.191.0.44 -Dsh=Deploy-Aussie_tripper.sh`
Note: use your instance's public IP address and don't forget to change the directory one level up. The result should be the following:



Access the application using the following url : https://<ip address of JCS VM>:7002/aussie-tripper/



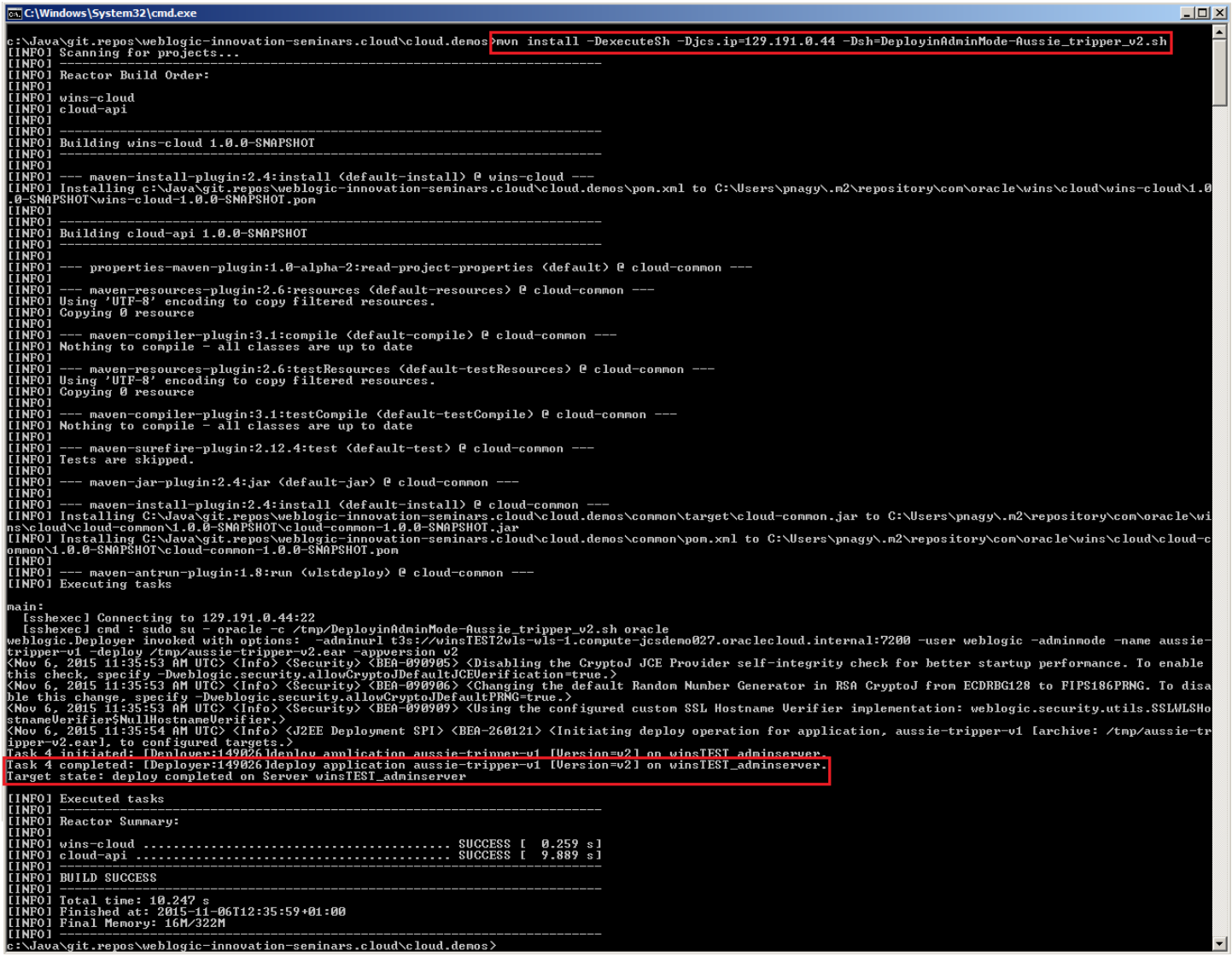
Click on a couple of places and the same would be reflected on the screen.



5) Now, we will deploy a more newer version of the application. Execute the following maven build in the `WINS_SOURCE_REPOSITORY\cloud.demos` directory:

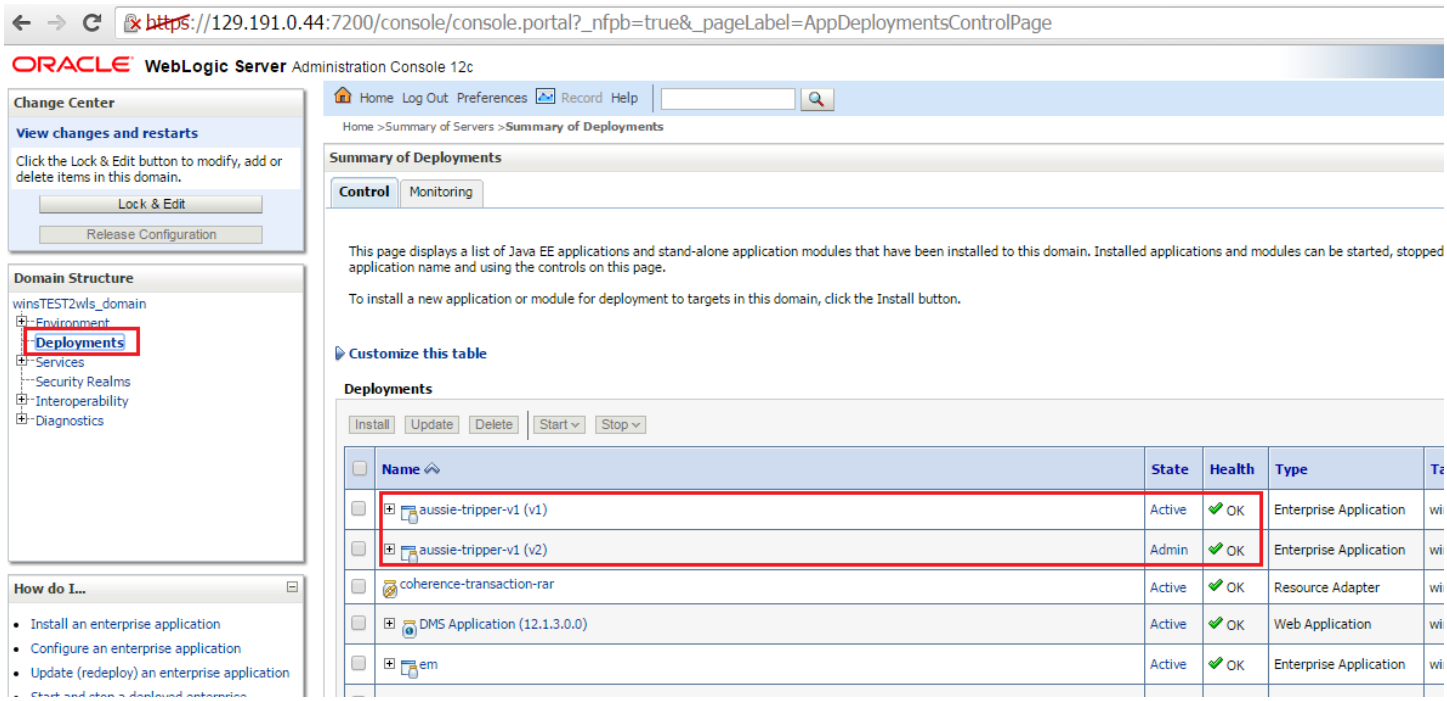
```
mvn install -DexecuteSh -Djcs.ip=129.191.0.44 -Dsh=DeployinAdminMode-Aussie_tripper_v2.sh
```

Note: use your instance's public IP address. The result should be the following:

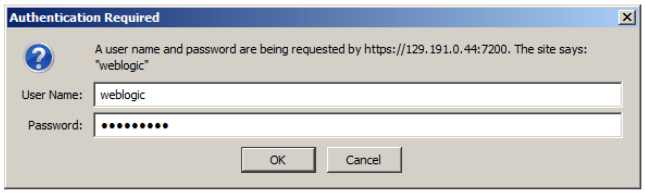


```
C:\Windows\System32\cmd.exe

c:\Java\git.repos\weblogic-innovation-seminars.cloud\cloud.demos>mvn install -DexecuteSh -Djcs.ip=129.191.0.44 -Dsh=DeployinAdminMode-Aussie_tripper_v2.sh
[INFO] Scanning for projects...
[INFO] ------------------------------------------------------------------------
[INFO] Reactor Build Order:
[INFO]
[INFO] wins-cloud
[INFO] cloud-api
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building wins-cloud 1.0.0-SNAPSHOT
[INFO] ------------------------------------------------------------------------
[INFO]
[INFO] --- maven-install-plugin:2.4:install (default-install) @ wins-cloud ---
[INFO] Installing c:\Java\git.repos\weblogic-innovation-seminars.cloud\cloud.demos\pom.xml to C:\Users\pnagy\.m2\repository\com\oracle\wins\cloud\wins-cloud\1.0
.0-SNAPSHOT\wins-cloud-1.0.0-SNAPSHOT.pom
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building cloud-api 1.0.0-SNAPSHOT
[INFO] ------------------------------------------------------------------------
[INFO]
[INFO] --- properties-maven-plugin:1.0-alpha-2:read-project-properties (default) @ cloud-common ---
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ cloud-common ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ cloud-common ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ cloud-common ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ cloud-common ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ cloud-common ---
[INFO] Tests are skipped.
[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ cloud-common ---
[INFO]
[INFO] --- maven-install-plugin:2.4:install (default-install) @ cloud-common ---
[INFO] Installing C:\Java\git.repos\weblogic-innovation-seminars.cloud\cloud.demos\common\target\cloud-common.jar to C:\Users\pnagy\.m2\repository\com\oracle\wi
ns\cloud\cloud-common\1.0.0-SNAPSHOT\cloud-common-1.0.0-SNAPSHOT.jar
[INFO] Installing C:\Java\git.repos\weblogic-innovation-seminars.cloud\cloud.demos\common\pom.xml to C:\Users\pnagy\.m2\repository\com\oracle\wins\cloud\cloud-c
ommon\1.0.0-SNAPSHOT\cloud-common-1.0.0-SNAPSHOT.pom
[INFO]
[INFO] --- maven-antrun-plugin:1.8:run (wlstdeploy) @ cloud-common ---
[INFO] Executing tasks

main:
  [sshexec] Connecting to 129.191.0.44:22
  [sshexec] cmd : sudo su - oracle -c /tmp/DeployinAdminMode-Aussie_tripper_v2.sh oracle
weblogic.Deployer invoked with options:  -adminurl t3s://winsTEST2wls-wls-1.compute-jcsdemo027.oraclecloud.internal:7200 -user weblogic -adminmode -name aussie-
tripper-v1 -deploy /tmp/aussie-tripper-v2.ear -appversion v2
<Nov 6, 2015 11:35:53 AM UTC> <Info> <Security> <BEA-090905> <Disabling the CryptoJ JCE Provider self-integrity check for better startup performance. To enable
this check, specify -Dweblogic.security.allowCryptoJDefaultJCEVerification=true.>
<Nov 6, 2015 11:35:53 AM UTC> <Info> <Security> <BEA-090906> <Changing the default Random Number Generator in RSA CryptoJ from ECDRBG128 to FIPS186PRNG. To disa
ble this change, specify -Dweblogic.security.allowCryptoJDefaultPRNG=true.>
<Nov 6, 2015 11:35:53 AM UTC> <Info> <Security> <BEA-090909> <Using the configured custom SSL Hostname Verifier implementation: weblogic.security.utils.SSLWLSHo
stnameVerifier$NullHostnameVerifier.>
<Nov 6, 2015 11:35:54 AM UTC> <Info> <J2EE Deployment SPI> <BEA-260121> <Initiating deploy operation for application, aussie-tripper-v1 [archive: /tmp/aussie-tr
ipper-v2.ear], to configured targets.>
Task 4 initiated: [Deployer:149026]deploy application aussie-tripper-v1 [Version=v2] on winsTEST_adminserver.
Task 4 completed: [Deployer:149026]deploy application aussie-tripper-v1 [Version=v2] on winsTEST_adminserver.
Target state: deploy completed on Server winsTEST_adminserver

[INFO] Executed tasks
[INFO] ------------------------------------------------------------------------
[INFO] Reactor Summary:
[INFO]
[INFO] wins-cloud ......................................... SUCCESS [  0.259 s]
[INFO] cloud-api .......................................... SUCCESS [  9.889 s]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 10.247 s
[INFO] Finished at: 2015-11-06T12:35:59+01:00
[INFO] Final Memory: 16M/322M
[INFO] ------------------------------------------------------------------------
c:\Java\git.repos\weblogic-innovation-seminars.cloud\cloud.demos>
```

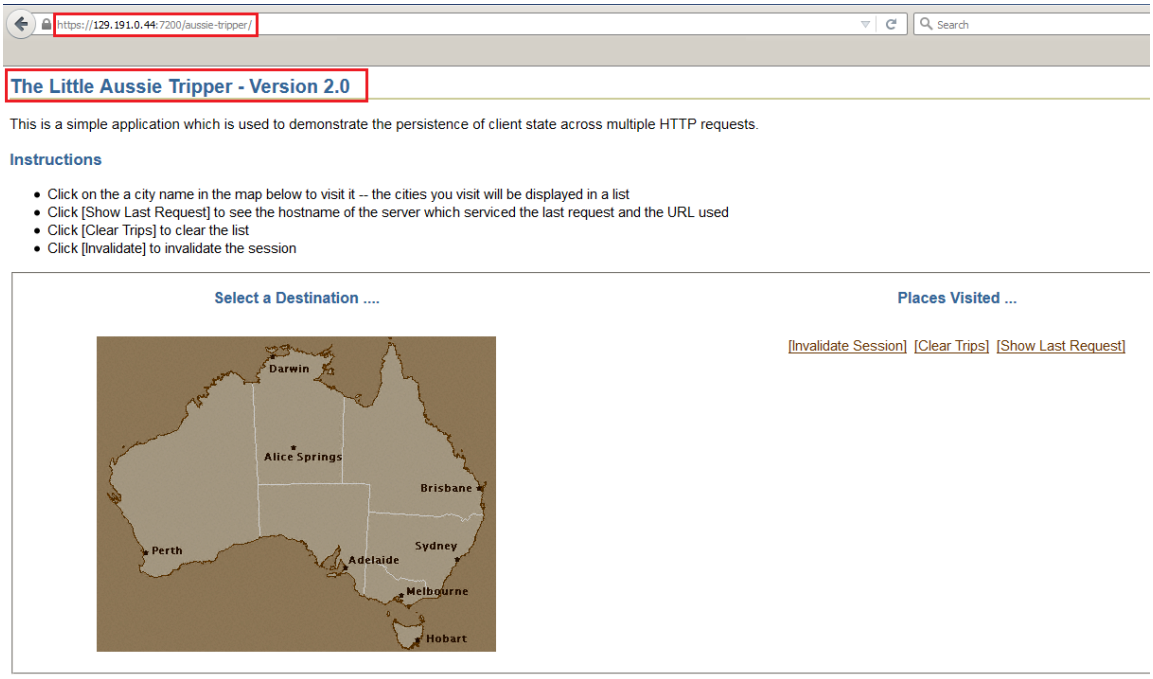
Go back to the WebLogic Server admin console and notice that there are 2 versions of the application deployed. And both are in Active state.



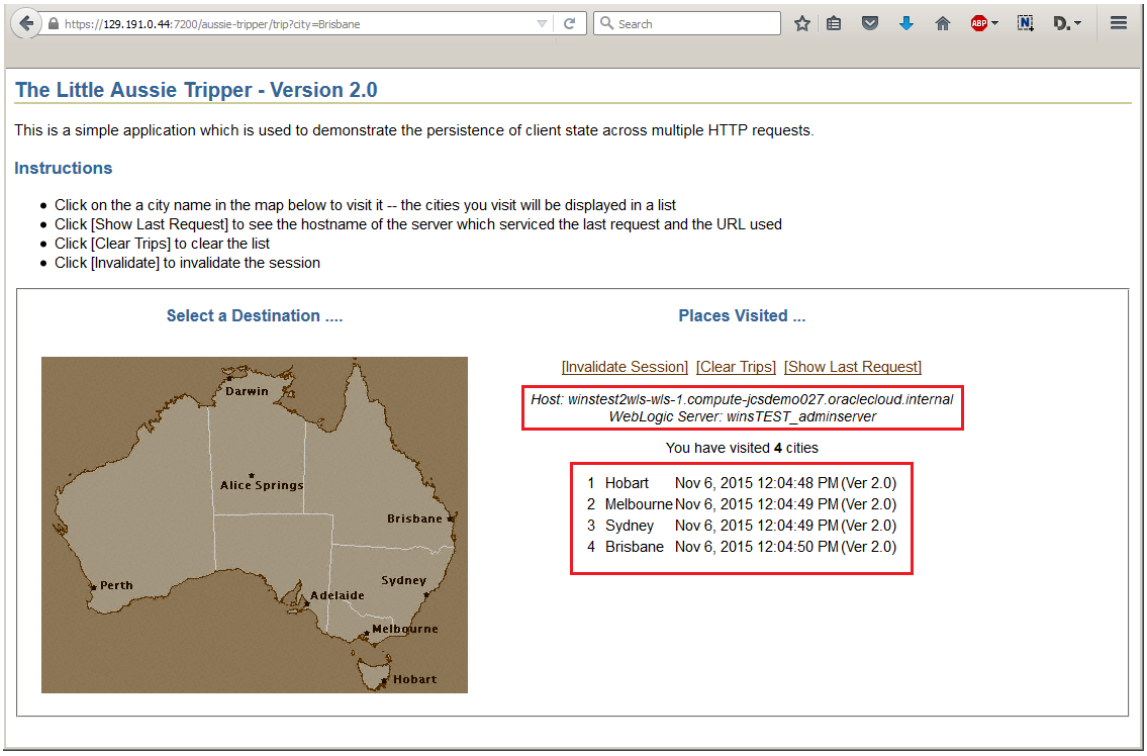
Launch a different browser and access the Version 2 of the application using the url : https://<ip address of the JCS VM>:7200/aussie-tripper. Login using the user name password which are used to login to the admin console
It can be useful to use a different browser to avoid any caching issues.



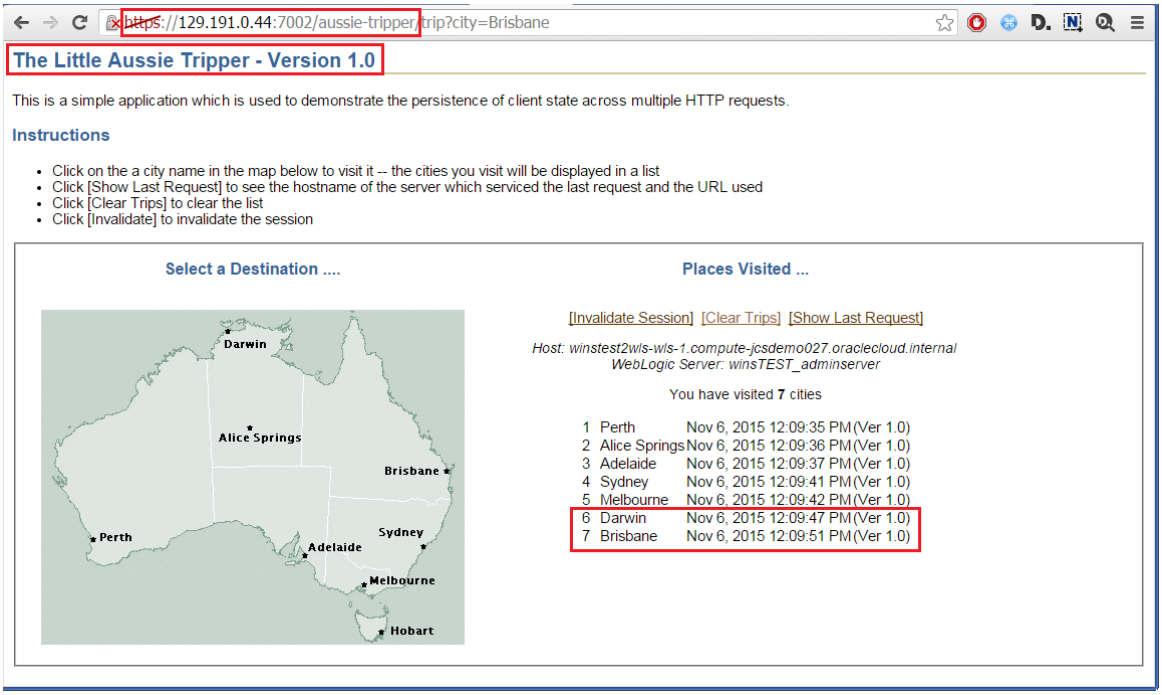
This should bring up the version 2 of the application.



Click on a few places of the map and see that it is being reflected on the screen.



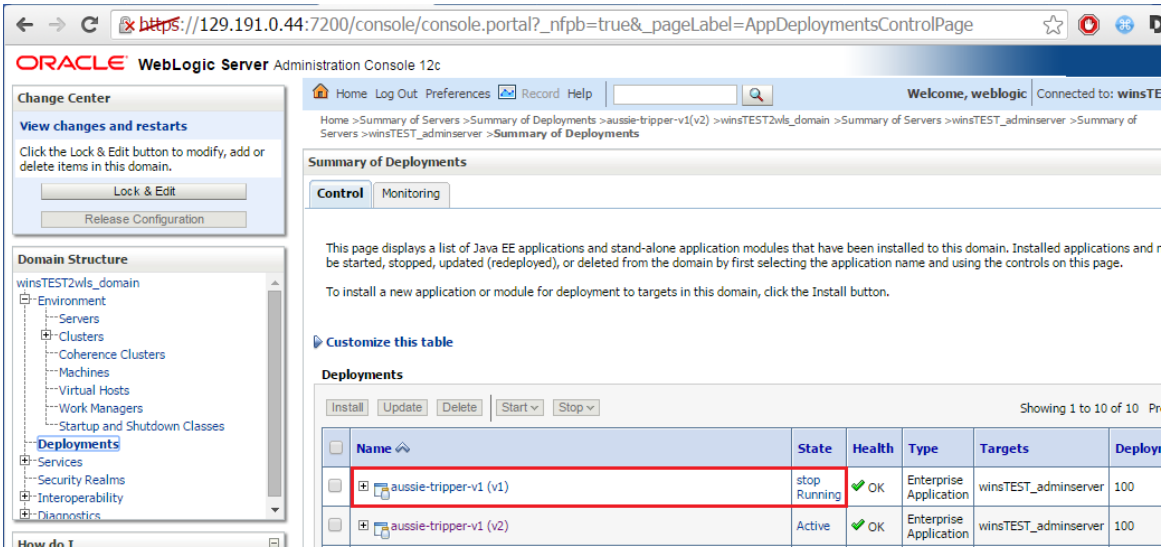
Go back to the other browser, where Version 1 was running and click on some more places on the map. You will notice that the state of the application of Version 1 was alive and more places are being continued to be clicked.



6) Now, you have to promote Version 2 of the application to go into production (change it from admin mode). Execute the following maven build in the `WINS_SOURCE_REPOSITORY\cloud.demos` directory:
`mvn install -DexecuteSh -Djcs.ip=129.191.0.44 -Dsh=Promote-Aussie_tripper_v2.sh`
Note: use your instance's public IP address. The result should be the following:

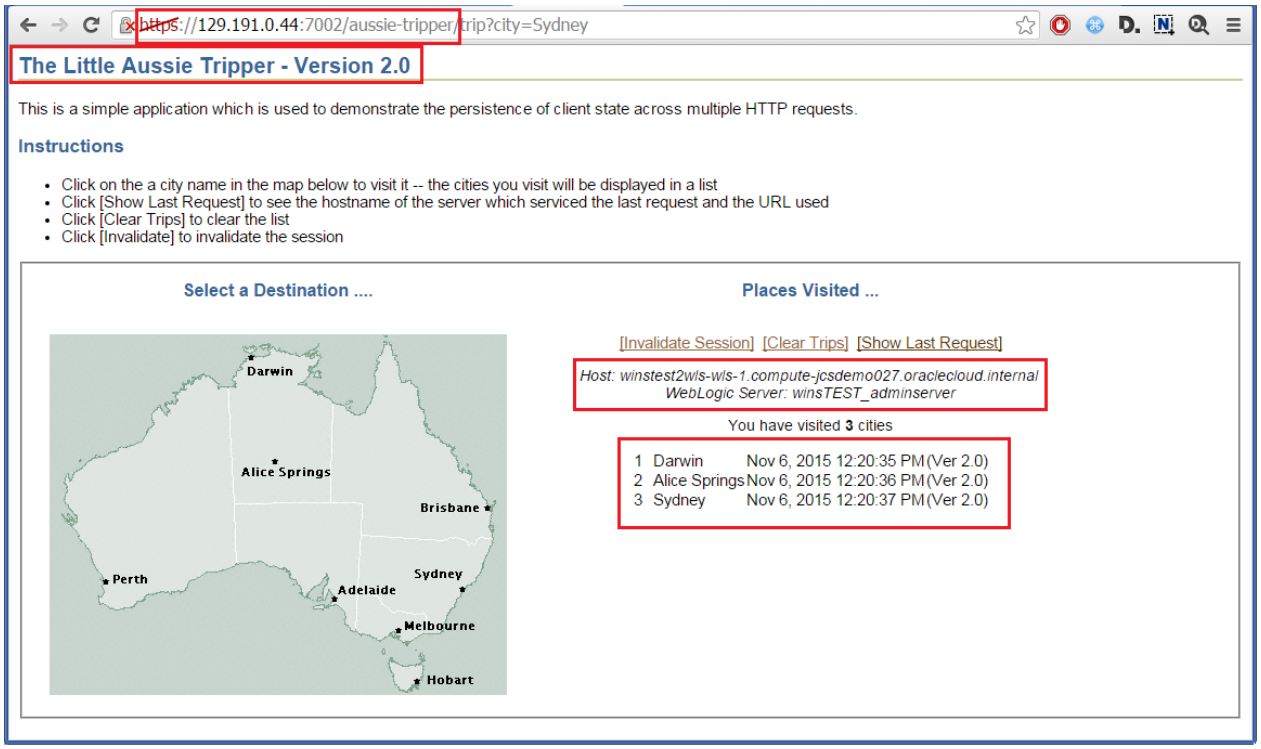
```
C:\Windows\System32\cmd.exe
c:\Java\git.repos\weblogic-innovation-seminars.cloud\cloud.demos>mvn install -DexecuteSh -Djcs.ip=129.191.0.44 -Dsh=Promote-Aussie_tripper_v2.sh
[INFO] Scanning for projects...
[INFO] ------------------------------------------------------------------------
[INFO] Reactor Build Order:
[INFO]
[INFO] wins-cloud
[INFO] cloud-api
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building wins-cloud 1.0.0-SNAPSHOT
[INFO] ------------------------------------------------------------------------
[INFO]
[INFO] --- maven-install-plugin:2.4:install (default-install) @ wins-cloud ---
[INFO] Installing c:\Java\git.repos\weblogic-innovation-seminars.cloud\cloud.demos\pom.xml to C:\Users\pnagy\.m2\repository\com\oracle\wins\cloud\wins-cloud\1.0
.0-SNAPSHOT\wins-cloud-1.0.0-SNAPSHOT.pom
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] Building cloud-api 1.0.0-SNAPSHOT
[INFO] ------------------------------------------------------------------------
[INFO]
[INFO] --- properties-maven-plugin:1.0-alpha-2:read-project-properties (default) @ cloud-common ---
[INFO]
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ cloud-common ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ cloud-common ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ cloud-common ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 0 resource
[INFO]
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ cloud-common ---
[INFO] Nothing to compile - all classes are up to date
[INFO]
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ cloud-common ---
[INFO] Tests are skipped.
[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ cloud-common ---
[INFO]
[INFO] --- maven-install-plugin:2.4:install (default-install) @ cloud-common ---
[INFO] Installing C:\Java\git.repos\weblogic-innovation-seminars.cloud\cloud.demos\common\target\cloud-common.jar to C:\Users\pnagy\.m2\repository\com\oracle\wi
ns\cloud\cloud-common\1.0.0-SNAPSHOT\cloud-common-1.0.0-SNAPSHOT.jar
[INFO] Installing C:\Java\git.repos\weblogic-innovation-seminars.cloud\cloud.demos\common\pom.xml to C:\Users\pnagy\.m2\repository\com\oracle\wins\cloud\cloud-c
ommon\1.0.0-SNAPSHOT\cloud-common-1.0.0-SNAPSHOT.pom
[INFO]
[INFO] --- maven-antrun-plugin:1.8:run (wlstdeploy) @ cloud-common ---
[INFO] Executing tasks

main:
  [sshexec] Connecting to 129.191.0.44:22
  [sshexec] cmd : sudo su - oracle -c /tmp/Promote-Aussie_tripper_v2.sh oracle
weblogic.Deployer invoked with options:  -adminurl t3s://winsTEST2wls-wls-1.compute-jcsdemo027.oraclecloud.internal:7200 -user weblogic -start -name aussie-trip
per-v1 -appversion v2
<Nov 6, 2015 12:13:05 PM UTC> <Info> <Security> <BEA-090905> <Disabling the CryptoJ JCE Provider self-integrity check for better startup performance. To enable
this check, specify -Dweblogic.security.allowCryptoJDefaultJCEVerification=true.>
<Nov 6, 2015 12:13:05 PM UTC> <Info> <Security> <BEA-090906> <Changing the default Random Number Generator in RSA CryptoJ from ECDRBG128 to FIPS186PRNG. To disa
ble this change, specify -Dweblogic.security.allowCryptoJDefaultPRNG=true.>
<Nov 6, 2015 12:13:05 PM UTC> <Info> <Security> <BEA-090909> <Using the configured custom SSL Hostname Verifier implementation: weblogic.security.utils.SSLWLSHo
stnameVerifier$NullHostnameVerifier.>
<Nov 6, 2015 12:13:07 PM UTC> <Info> <J2EE Deployment SPI> <BEA-260121> <Initiating start operation for application, aussie-tripper-v1 [archive: null], to confi
gured targets.>
Task 5 initiated: [Deployer:149026]start application aussie-tripper-v1 [Version=v2] on winsTEST_adminserver.
Task 5 completed: [Deployer:149026]start application aussie-tripper-v1 [Version=v2] on winsTEST_adminserver.
Target state: start completed on Server winsTEST_adminserver

[INFO] Executed tasks
[INFO] ------------------------------------------------------------------------
[INFO] Reactor Summary:
[INFO]
[INFO] wins-cloud ......................................... SUCCESS [  0.263 s]
[INFO] cloud-api .......................................... SUCCESS [  9.086 s]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time: 9.462 s
[INFO] Finished at: 2015-11-06T13:13:10+01:00
[INFO] Final Memory: 16M/322M
[INFO] ------------------------------------------------------------------------
c:\Java\git.repos\weblogic-innovation-seminars.cloud\cloud.demos>_
```

The application is deployed. Go to the WebLogic Server admin console and refresh the deployments screen. Version 1 is not in Stop Running state while Version 2 is in Active state.

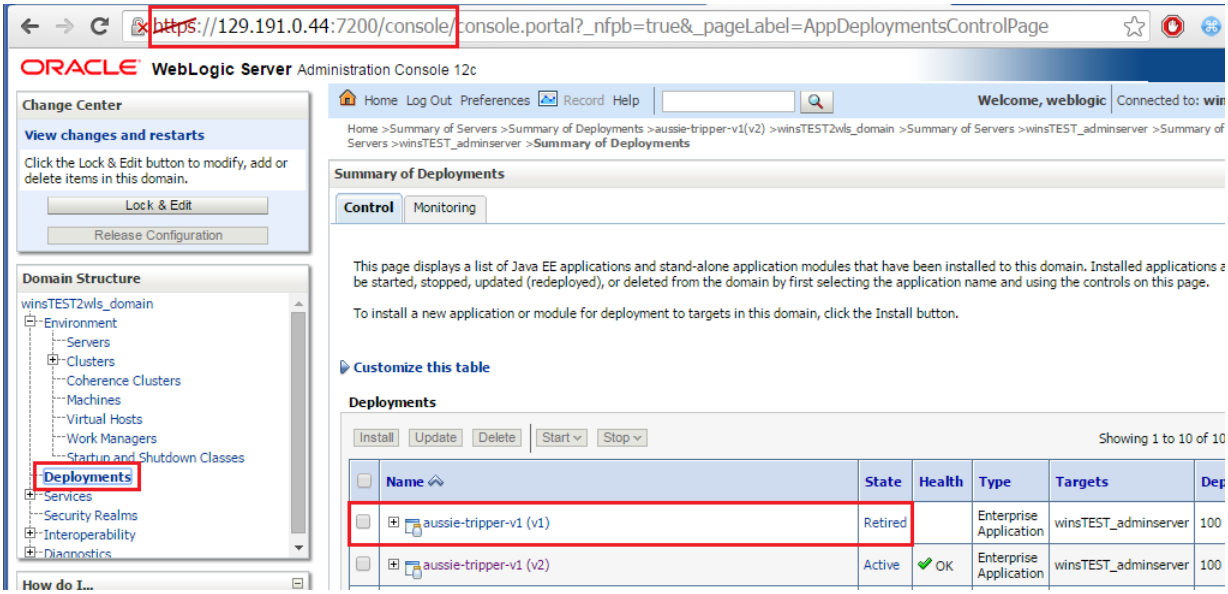


Go the browser where V1 was running and click on 'Invalidate Session'. A message will be shown that the session has ended. Click on the link to access the application again.

This time when the page loads it is version 2 of the application. You can click on various places on the map and the same will be reflected on the screen.

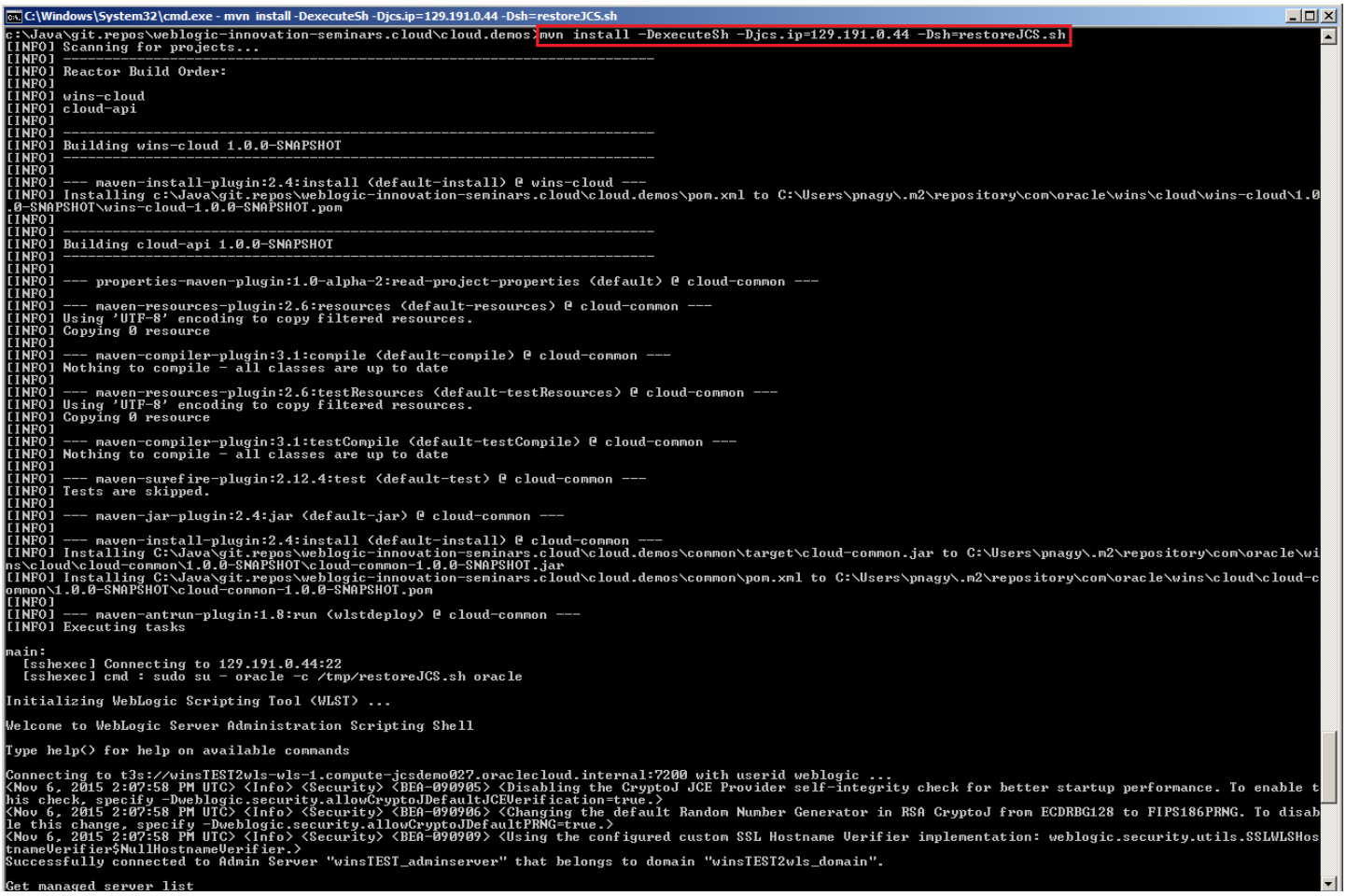


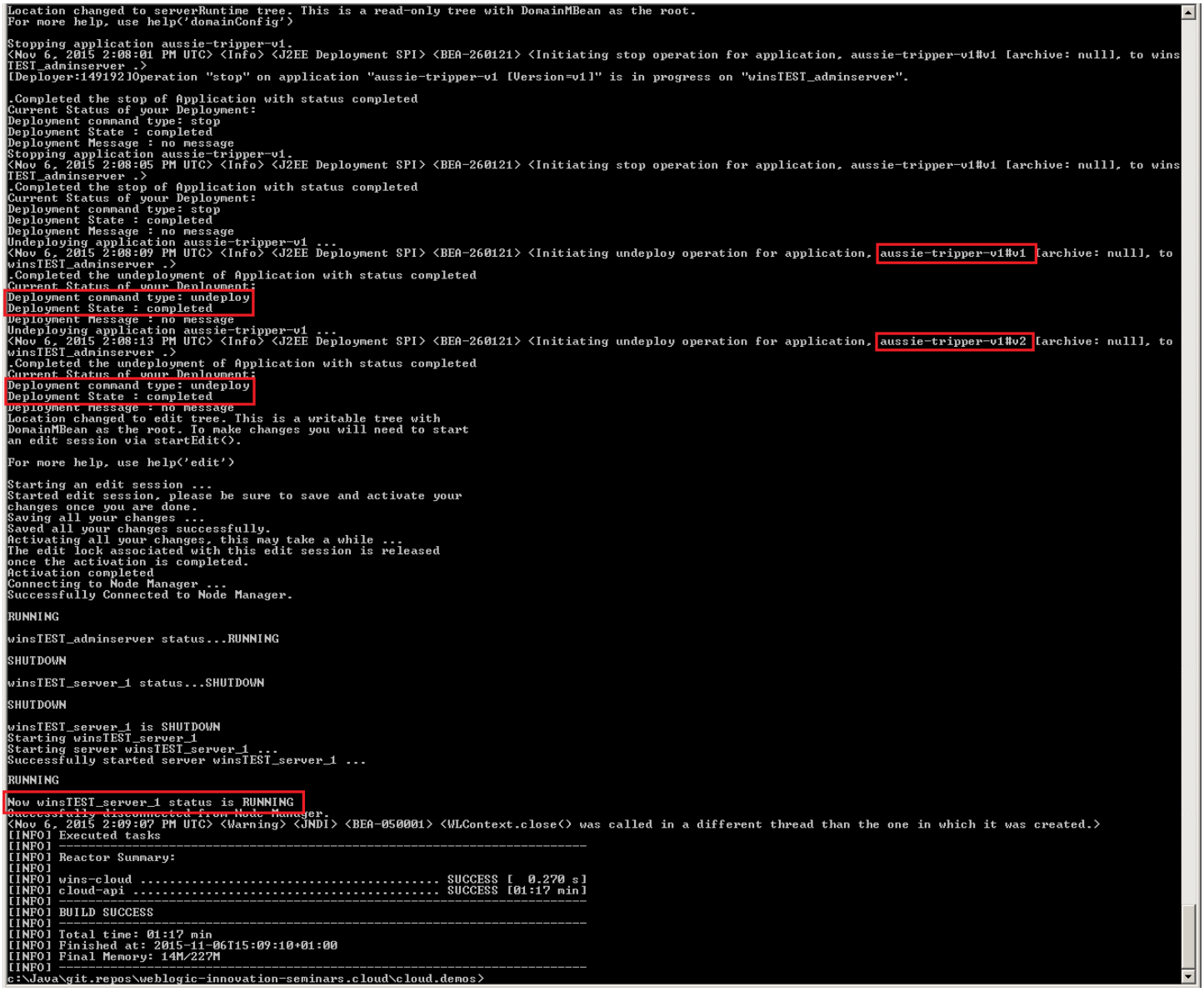
Close both application's browsers page/tab and login again to WebLogic Server Admin Console. Check out the Deployments page. This time you will notice that the Version 1 of the application is now in 'retired' state while Version 2 is in Active state.



Restore Java Cloud Service Instance

1. In order to roll back the changes on JCS execute the following maven build in the `WINS_SOURCE_REPOSITORY\cloud.demos` directory:
   `mvn install -DexecuteSh -Djcs.ip=129.191.0.44 -Dsh=-Dsh=restoreJCS.sh`
   Note: use your instance's public IP address. The result should be the following:


C:\Windows\System32\cmd.exe - mvn install -DexecuteSh -Djcs.ip=129.191.0.44 -Dsh=restoreJCS.sh
c:\Java\git.repos\weblogic-innovation-seminars.cloud\cloud.demos>mvn install -DexecuteSh -Djcs.ip=129.191.0.44 -Dsh=restoreJCS.sh
[INFO] Scanning for projects...
[INFO] Reactor Build Order:
[INFO] wins-cloud
[INFO] cloud-api
[INFO] Building wins-cloud 1.0.0-SNAPSHOT
[INFO] --- maven-install-plugin:2.4:install (default-install) @ wins-cloud ---
[INFO] Installing c:\Java\git.repos\weblogic-innovation-seminars.cloud\cloud.demos\pom.xml to C:\Users\pnagy\.m2\repository\com\oracle\wins\cloud\wins-cloud\1.0.0-SNAPSHOT\wins-cloud-1.0.0-SNAPSHOT.pom
[INFO] Building cloud-api 1.0.0-SNAPSHOT
[INFO] --- properties-maven-plugin:1.0-alpha-2:read-project-properties (default) @ cloud-common ---
[INFO] --- maven-resources-plugin:2.6:resources (default-resources) @ cloud-common ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 0 resource
[INFO] --- maven-compiler-plugin:3.1:compile (default-compile) @ cloud-common ---
[INFO] Nothing to compile - all classes are up to date
[INFO] --- maven-resources-plugin:2.6:testResources (default-testResources) @ cloud-common ---
[INFO] Using 'UTF-8' encoding to copy filtered resources.
[INFO] Copying 0 resource
[INFO] --- maven-compiler-plugin:3.1:testCompile (default-testCompile) @ cloud-common ---
[INFO] Nothing to compile - all classes are up to date
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ cloud-common ---
[INFO] Tests are skipped.
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ cloud-common ---
[INFO] --- maven-install-plugin:2.4:install (default-install) @ cloud-common ---
[INFO] Installing C:\Java\git.repos\weblogic-innovation-seminars.cloud\cloud.demos\common\target\cloud-common.jar to C:\Users\pnagy\.m2\repository\com\oracle\wins\cloud\cloud-common\1.0.0-SNAPSHOT\cloud-common-1.0.0-SNAPSHOT.jar
[INFO] Installing C:\Java\git.repos\weblogic-innovation-seminars.cloud\cloud.demos\common\pom.xml to C:\Users\pnagy\.m2\repository\com\oracle\wins\cloud\cloud-common\1.0.0-SNAPSHOT\cloud-common-1.0.0-SNAPSHOT.pom
[INFO] --- maven-antrun-plugin:1.8:run (wlstdeploy) @ cloud-common ---
[INFO] Executing tasks
main:
[sshexec] Connecting to 129.191.0.44:22
[sshexec] cmd : sudo su - oracle -c /tmp/restoreJCS.sh oracle
Initializing WebLogic Scripting Tool (WLST) ...
Welcome to WebLogic Server Administration Scripting Shell
Type help() for help on available commands
Connecting to t3s://winsTEST2wls-wls-1.compute-jcsdemo027.oraclecloud.internal:7200 with userid weblogic ...
<Nov 6, 2015 2:07:58 PM UTC> <Info> <Security> <BEA-090905> <Disabling the CryptoJ JCE Provider self-integrity check for better startup performance. To enable this check, specify -Dweblogic.security.allowCryptoJDefaultJCEVerification=true.>
<Nov 6, 2015 2:07:58 PM UTC> <Info> <Security> <BEA-090906> <Changing the default Random Number Generator in RSA CryptoJ from ECDRBG128 to FIPS186PRNG. To disable this change, specify -Dweblogic.security.allowCryptoJDefaultPRNG=true.>
<Nov 6, 2015 2:07:58 PM UTC> <Info> <Security> <BEA-090909> <Using the configured custom SSL Hostname Verifier implementation: weblogic.security.utils.SSLWLSHostnameVerifier$NullHostnameVerifier.>
Successfully connected to Admin Server "winsTEST_adminserver" that belongs to domain "winsTEST2wls_domain".
Get managed server list



Troubleshooting

If the script fails the following steps need to be complete:

- Undeploy all versions of Aussie-tripper application
- Disable Administration port (Domain -> Configuration -> General)
- Disable Tunneling (Adminserver -> Protocol -> General)

Note: once you have apply changes (Disable Administration port) the Admin console will no longer available on port 7200.

Typical Questions

- Side by Side deployment importance and steps
    - Releasing a new version of the product is not an easy task. After finishing development cycles and QA cycles, finally it is time to deploy the new version to the application server. Servers are shut down, the new version of the product is swapped in, and then pray there is no issue once everything comes up. If

you are lucky, only a few issues will show up. However if you are not in luck then you have to roll back the old applications, and it is even more painful than deploying the new version, because sometimes you cannot roll back all of the changes.
    - This is when it helps to have multiple versions on the application server and one could switch between each version without interrupting the system.
    - Side-by-side application deployment controls the process for deploying new versions of Web-based applications without the need to disrupt service. The new version of an application is deployed alongside existing version - WebLogic will gradually migrate the traffic. The older version is automatically un-deployed after all current clients complete their work. The Administrator explicitly un-deploys the older version, or a configured timeout is reached.
    - Rolling back the new version is simple: just stop the redeployment process if problems are detected in the newer application version.
    - For new applications, administrators can deploy an application in "administration mode," which makes it inaccessible to non-admin clients, in order to do sanity checks to ensure that the application is working as expected and then open it up to clients.

- List of the features of side-by-side deployment
    - Multiple application versions can coexist
    - Test versions before releasing to users
    - Roll back to previous versions
    - Automatic retirement: graceful, timeout, immediate
    - Creates version-aware application artifacts/resources
    - Reduces hardware, software, maintenance, and support costs

- Pre-requisite
    - Prerequisite to deploy the application using "Side By Side Deployment" mechanism is that the previously deployed application has to be versioned.
    - If previously deployed application is not versioned then the new application cannot be deployed using "Side By Side Deployment" mechanism. If attempted, it will throw weblogic.management.ManagementException: [Deployer:149081]You cannot deploy application, '[Your-App-Name]', with version '[App-Version]'. The application was previously deployed without version.

Wrap up / Overview

Sometimes, this is also referred to as Production Redeployment. This enables you to update and redeploy an application in a production environment without stopping the application or otherwise interrupting the application's availability to clients. Production redeployment saves you the trouble of scheduling application downtime, setting up redundant servers to host new application versions, manually managing client access to multiple application versions, and manually retiring older versions of an application.