Oracle Code Day – Blackjack Hands-On Labs (HOL)

Table of Contents

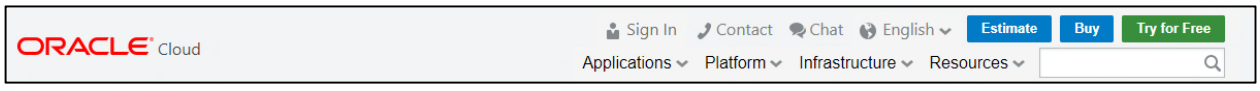
Required Items for HOL labs

- Cloud Account
- Oracle University environment simulating a local development environment with required software (Git, Maven, and NetBeans) already installed. It will also have the Blackjack application.

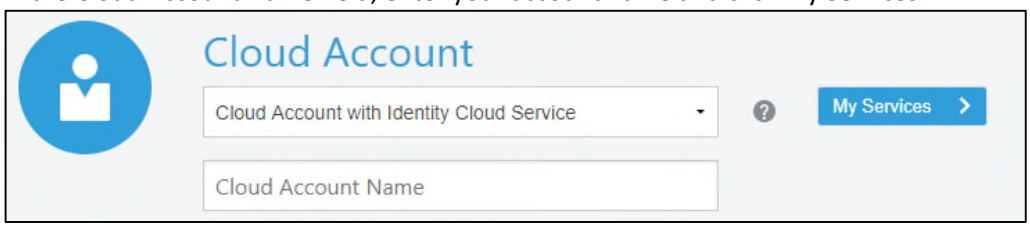
Trial Cloud Accounts

DO THIS ONLY IF USING A TRAIL CLOUD ACCOUNT

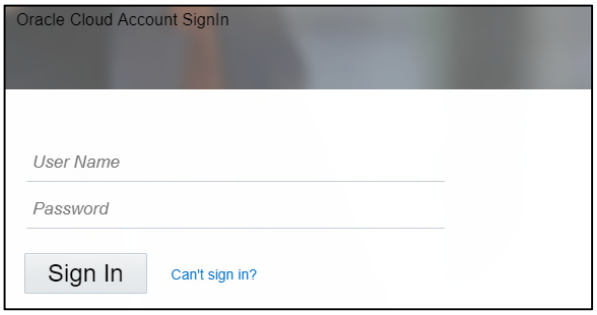
1. In the browser in your OU virtual desktop, go to cloud.oracle.com. Click Sign In.



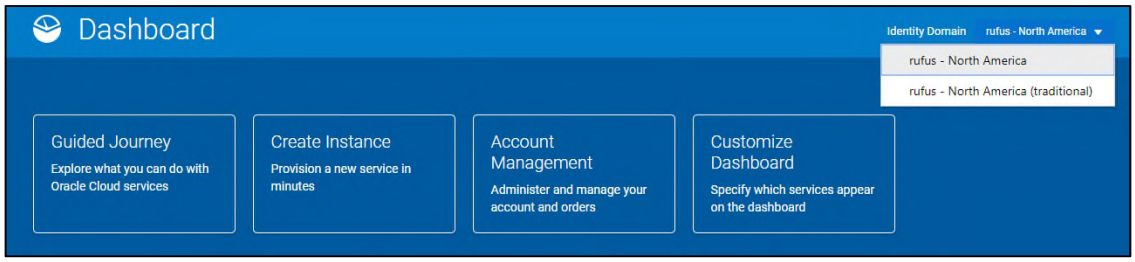
2. In the Cloud Account Name field, enter your account name and click My Services.



3. Enter in your cloud username and password.

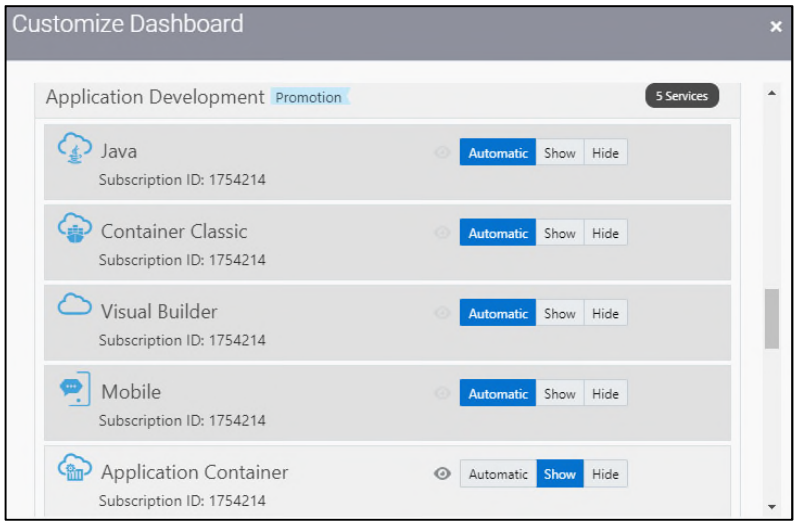


4. In the upper right corner, you will see a down arrow for your domain. Click it and you will see two cloud account domains, traditional cloud and identity cloud. Depending on your location, the account may show a different country.
   Your *traditional cloud* will have Developer Cloud Service, and your *identity cloud* will have Application Container Cloud Service.
   Select the identity cloud from the drop-down list.

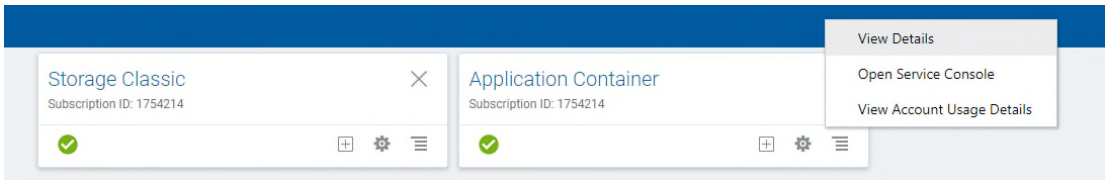


5. If you don't see Application Container displayed on the Dashboard, click Customize Dashboard.

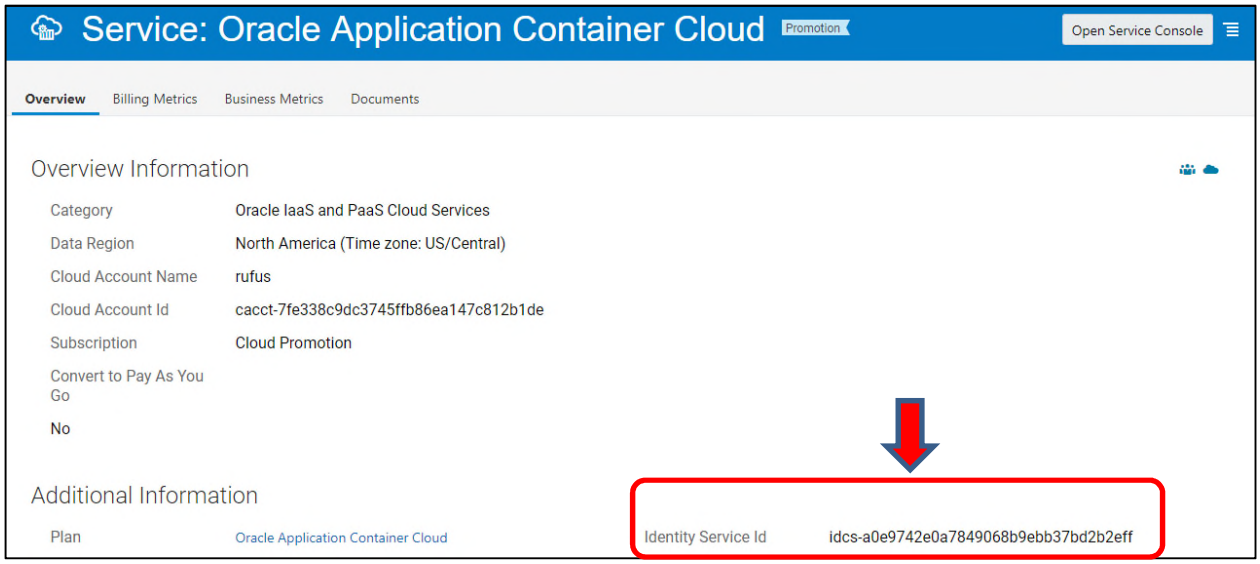
   a. Scroll to the Application Development section, make sure **Show** is highlighted for Application Container.



6. View the Application Container Cloud Service (CS) details: click the menu (four horizontal lines) and select View Details



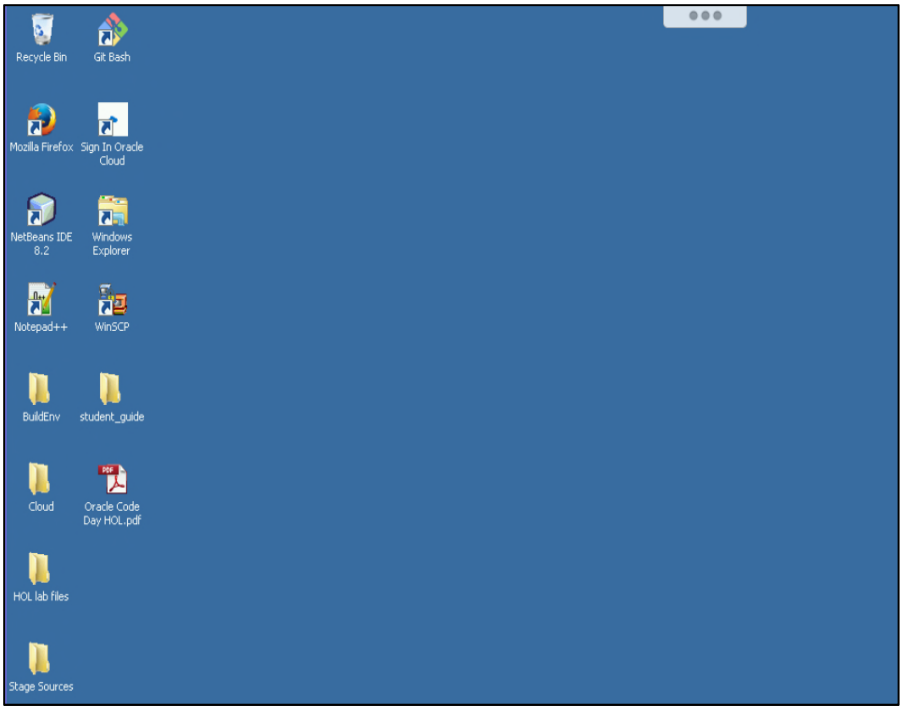
7. On the Overview page of Oracle Application Container Cloud Service, note the Identity Service ID. Copy the Identity Service ID to NotePad.



Test, Build and Deploy the Blackjack Web Service App

I. Deploying the Blackjack Application on a Local Server

1. Follow the instructions to log into your provided Oracle University class account.
   The virtual desktop looks like this:



2. Observe that on the desktop, there is a directory named **Cloud**. In the Cloud directory there is a **Blackjack.zip** file.
   The Blackjack.zip file has been unzipped and the extracted files are in the Blackjack directory.
   In the Blackjack directory are two sub-directories (Blackjack-part2 and html5-client), as follows:

   Blackjack-part2

   - Scr – The source for the Blackjack application has been created for you.
   - Target – Files created from a previous build of an application
   - Manifest.json – Configuration file for deploying to Oracle Application Container Cloud Service
   - Pom.xml – Defines plugs-in for Maven

   html5-client

   - cards folder – Images of the deck of cards
   - lib folder
   - index.html – An html file (HyperText Markup Language), it defines the structure and layout of a web page.
   - request.json – A json file, JSON (JavaScript Object Notation) use to store information of how you request your cards in Blackjack.

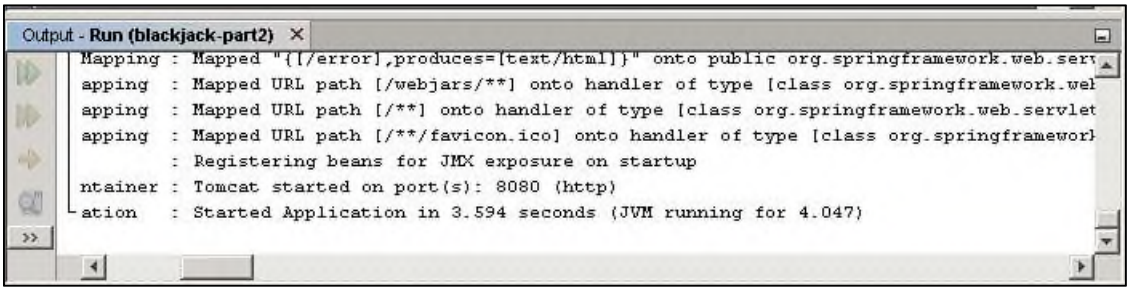
   The HTML-5 client application interacts with the Blackjack gaming application deployed on the Tomcat Server that is installed locally when NetBeans was installed on the VM.

3. Launch NetBeans using the NetBeans IDE icon on the desktop.
   You will see the blackjack project already loaded for you. You will do a **CLEAN and BUILD.** A **CLEAN and BUILD** will build your application for deployment. When you use the Clean and Build command, the IDE runs a build script that performs the following tasks:
   - Deletes any previously compiled files and other build outputs.
   - Recompiles the application and builds a JAR file containing the compiled files

4. Right-click the **backjack-part2** project and select **Clean and Build**.
   You may continue to the next step when you see "Build Success" in the Output window in NetBeans. You may have to scroll up a little to see the "Build Success" message.

5. Deploy the project on Tomcat Server
   a. Right click the **backjack-part2** project and select **Run**.
   b. Select **com.example.blackjack.rest.Application** from the **Available Main Classes** list, and click the **Select Main Class** button.

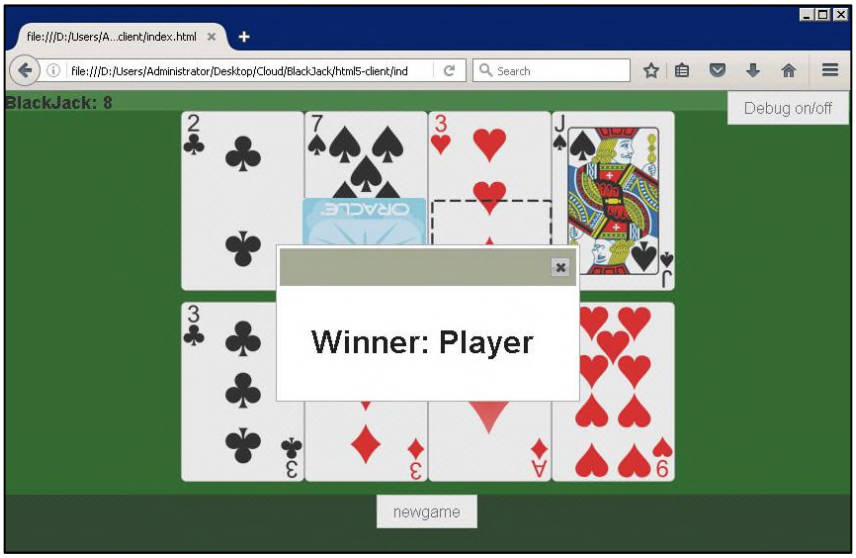
   You know your application has started and is running when you see a "*Started Application in <<seconds>> seconds (JVM running for 5.19)*" message in the Output window. You may need to scroll to the right In order to see the "Started Application" message.



II. Testing the Locally Deployed Blackjack Application

1. Run Blackjack application locally:
   a. Go to Desktop > Cloud > Blackjack > html5-client directory.
   b. Open the **index.html** file. (Double-click the file to open it in a browser.)
   c. Make sure that the first field, Service, is populated with <**http://127.0.0.1:8080**/>, enter a number between 1 and 9 in the second field, and then click **Connect**.

2. Once you connect to the Blackjack application gaming console, do the following:
   a. Click the Debug on/off button to view the Debug console.
   b. You can use the Hit and Stand buttons available in the user interface to play the game.



3. Close this HTML5 Client application once you are done.

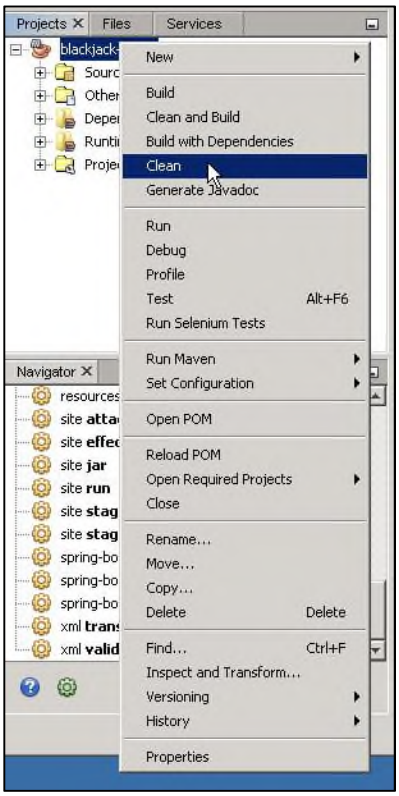
III. Generating Application Archive Files for the Blackjack Application

Oracle Application Container Cloud can deploy and run applications written in a number of popular programming languages, including Java Platform, Standard Edition (Java SE), and Node.js. Before deploying the application, you will compress the application in a ZIP or Gzipped Tar (TGZ) archive file, which includes the required configuration information.

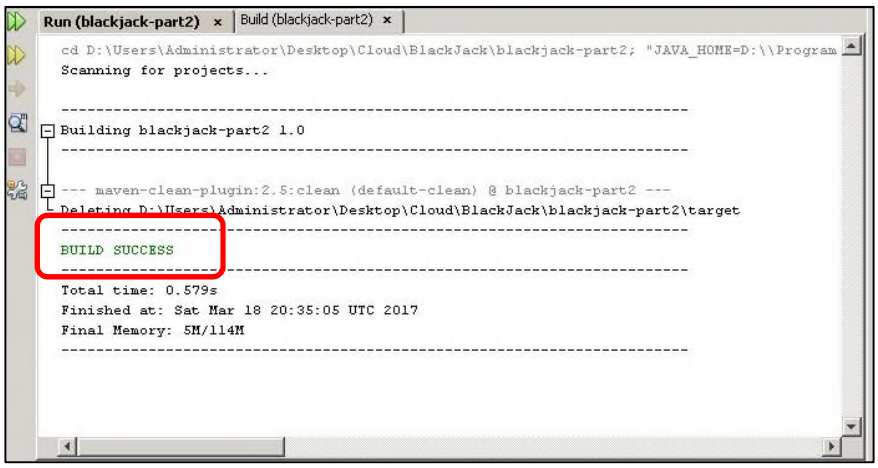
Note: You will use Oracle Application Container Cloud Service later in the lab.

1. Create an application archive:
   a. Open the blackjack-part2 application in NetBeans if it is not opened already.
   b. Right-click the project and click **Clean**.

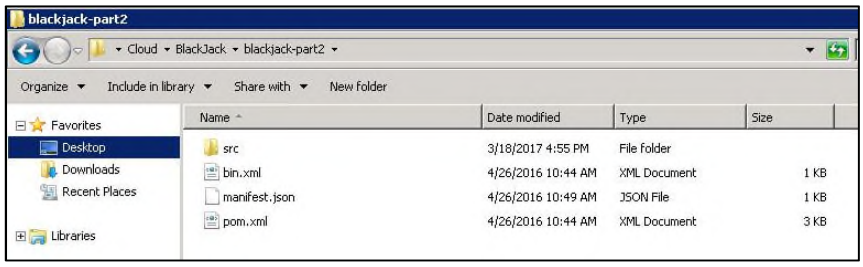
      A **Clean** will remove what you have created before.



c. When you see “Build Success” in the output window, the Clean is completed.



2. Open a graphical file explorer, navigate to **Desktop > Cloud > Blackjack > blackjack-part2**, and make a note of the directory structure and its contents.



Note: The manifest.json file is required for all applications deployed to Oracle Application Container Cloud Service. If this file is not present in the root directory of the.zip,.tar, or.tar.gz file, deployment will fail. At a minimum, this file specifies the major version of the runtime environment and the launch command.
Note also, in the **manifest.json** file**,** it shows a **java -jar Blackjack-part2-1.0.jar** file will be created as a result of a Build.

Contents of the manifest.json file:

```
{
  "runtime": {
    "majorVersion": "8"
  },
  "command": "java -jar blackjack-part2-1.0.jar",
  "release": {
    "version": "1.0",
    "build": "1",
    "commit": "1A2B345"
  },
  "notes": "Blackjack Web Service"
}  ```
```

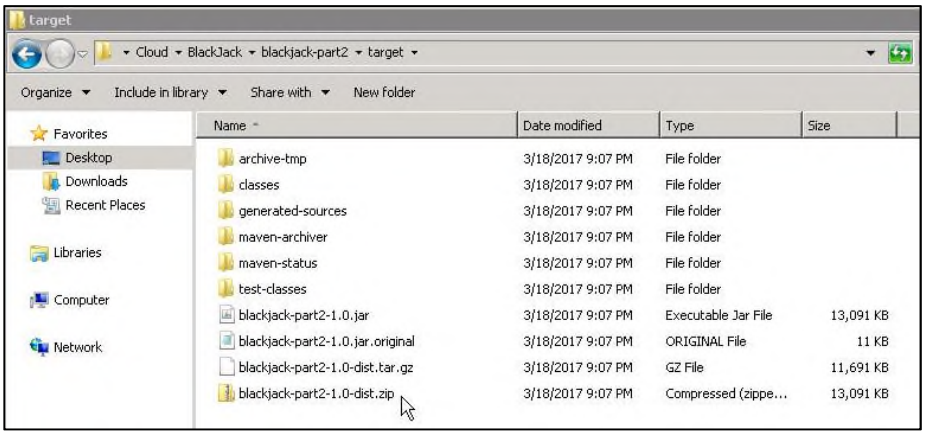
3. Switch to NetBeans, right-click the blackjack-part2 project, and click **Build**.

   A **Build** will compile only the source files that need to be compiled & links them together to form a new binary.

   When you see "Build Success" in the output window, the Build is completed

4. Switch to the file explorer and go to **Desktop > Cloud > Blackjack > blackjack-part2** directory and notice that a new directory named **target** is created.

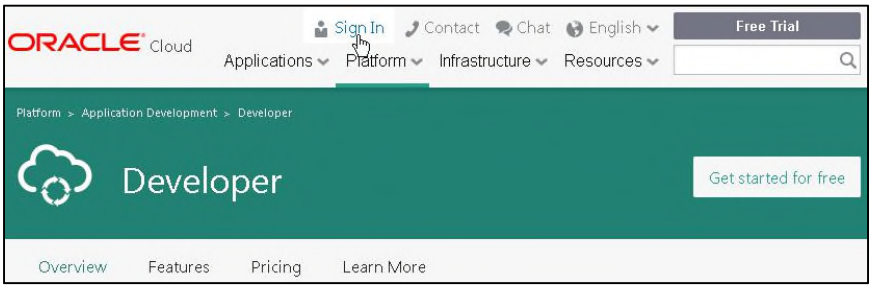
   Examine the target directory. You will notice that.zip and.tar.gz distribution files have been generated. These are application archive files you will use to deploy to OACCS.



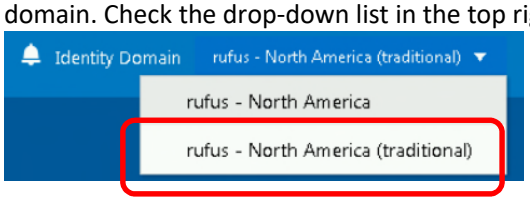
IV. Creating a New Project in Developer Cloud Service (DevCS)

Log in to your cloud account if you have not already done so.

1. **Log in to your Oracle Cloud Account**
   a. Go to the URL: https://cloud.oracle.com/
   b. Click Sign In.



Reminder: If you are using a Trail Cloud Account, make sure that you are using the *traditional* domain. Check the drop-down list in the top right corner.

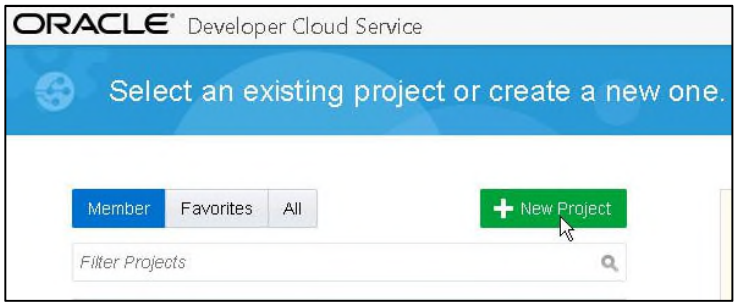


2. Create an empty Project in Developer Cloud Service
   a. On the **Developer Cloud Service** on the Dashboard, click the menu (gray horizontal lines).
   b. Select **Open Service Console**.

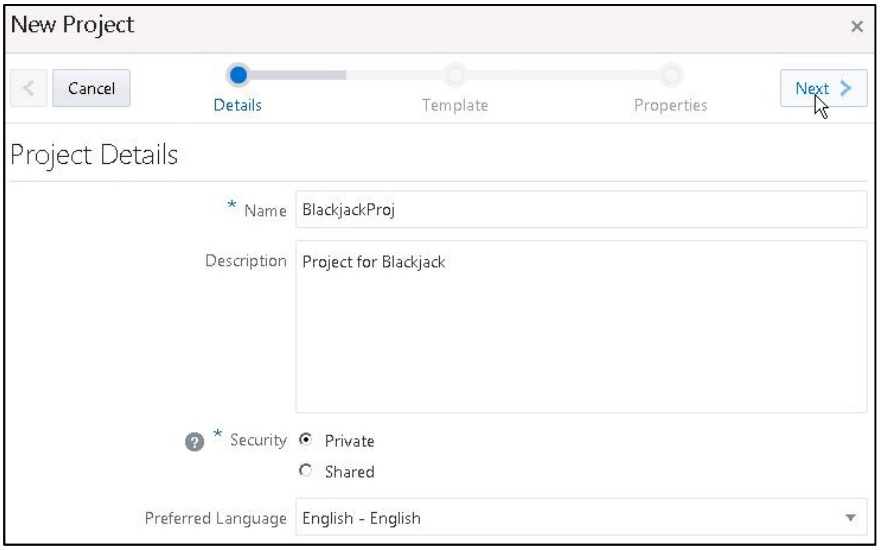
   Developer88387 is the name of the Service instance in DevCS; it will be different for every user.

   Note: If using a Trail Cloud account, you will see a window prompting you to log into the traditional cloud account. Enter in your domain, then your username and password.

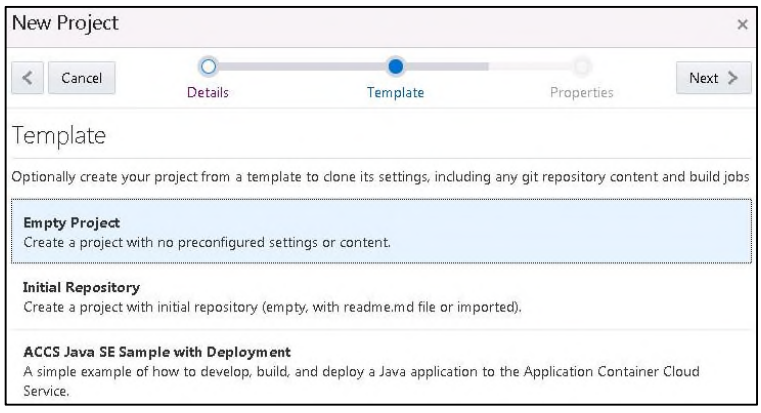
3. On the Developer Cloud Service page, click **New Project**



4. Enter the **Project Name and Description** as shown in the following screenshot and click **Next**



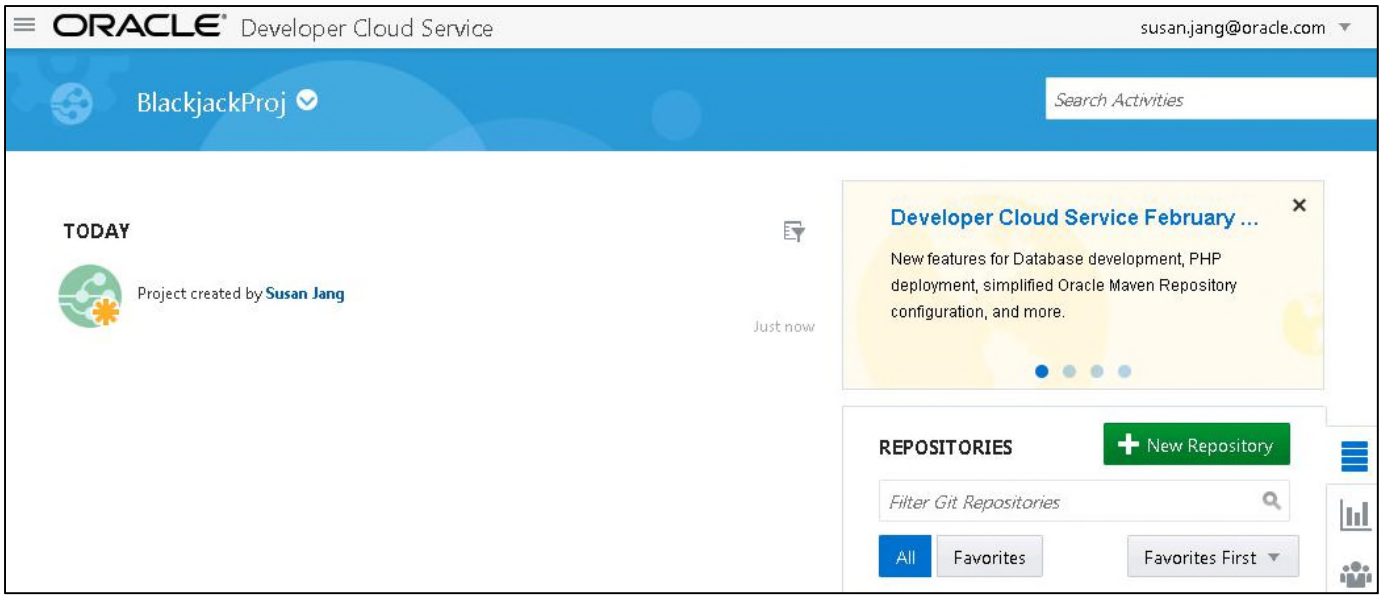
5. Click the **Empty Project** template and **Nex**t.



6. Select **MARKDOWN** from the Wiki Markup drop-down list and click **Finish**.

   Markdown is a plain-text formatting, and a software tool, written in Perl, that converts the plain text formatting to HTML. Provisioning the BlackjackProj project may take several minutes. When provisioning is completed, there will be a check next to all the modules.

7. The home screen of the BlackjackProj will be displayed.



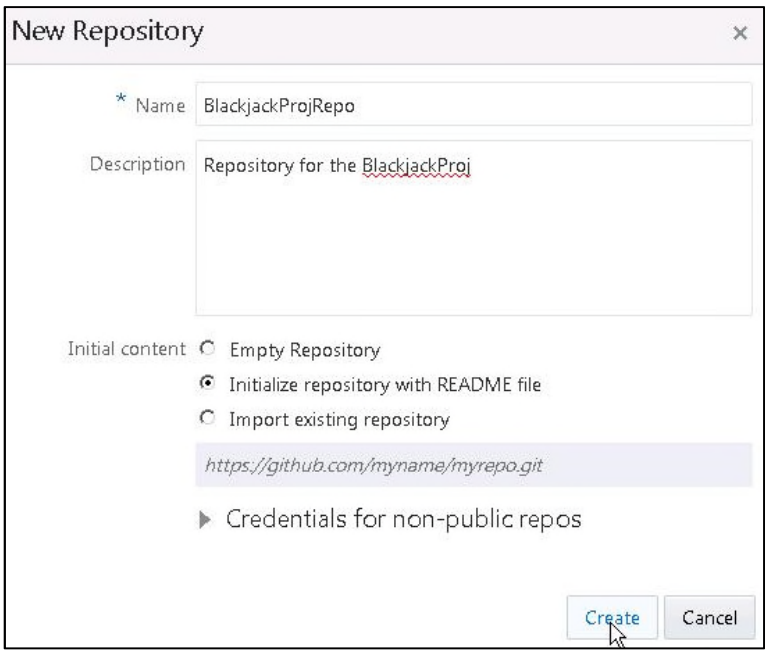
V. *Creating a GIT Repository in Developer Cloud Service*

Oracle Developer Cloud Service uses GIT source control management to save and manage your application source code files.

Create an empty GIT repository in Developer Cloud Service.

1. On the home screen of the BlackjackProj, click the **New Repository** button in the REPOSITORIES section.

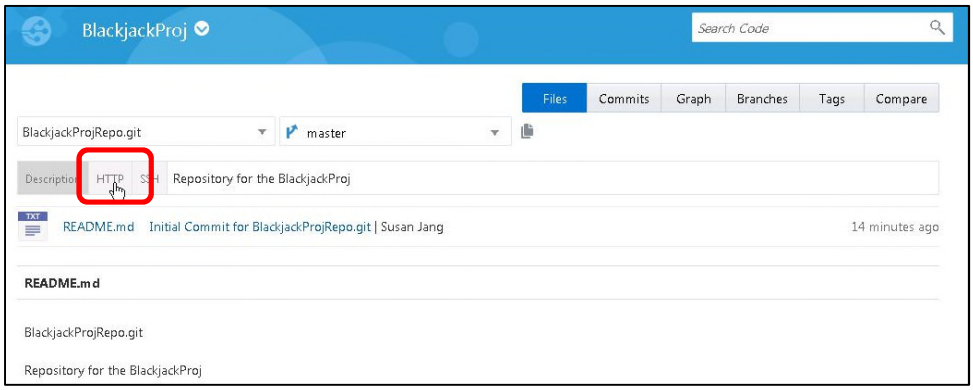
2. In the New Repository window, enter the repository **name** and a **description** as shown in the following screenshot and click **Create**.



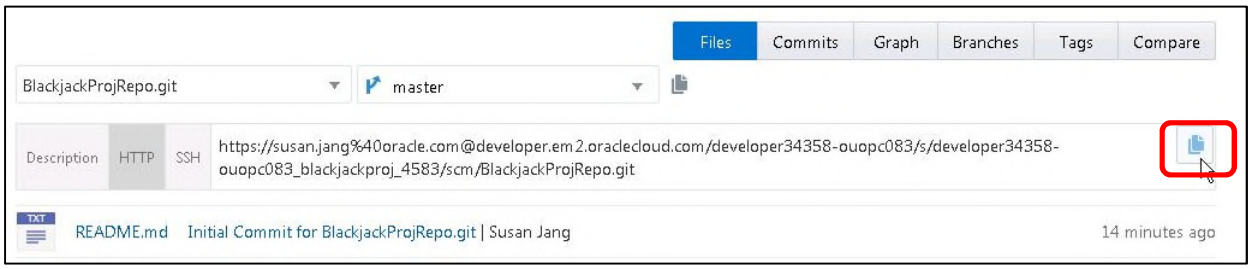
3. It may take a few minutes to create a repository.

   You will use https to connect your local GIT repository on your assigned OU classroom machine to the GIT repository in Developer Cloud Service

4. To get the URL to access the GIT Repository in Developer Cloud Service**,** click the **HTTP tab** in the Blackjack-Part2Repo home page.



5. When you click the HTTP tab, it will then display the HHTP. Click the icon on the far right to **copy the URL to Clipboard**.



   Here is an example of what will be in Clipboard.

```
https://susan.jang%40oracle.com@developer.em2.oraclecloud.com/developer34358-
ouopc083/s/developer34358-ouopc083_blackjackproj_4583/scm/BlackjackProjRepo.git
```

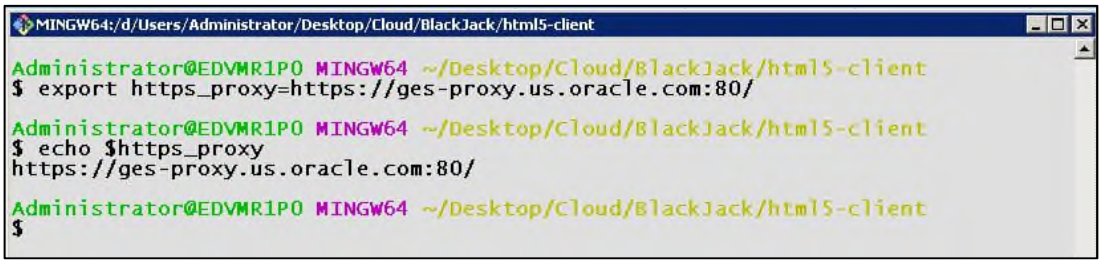
VI. Cloning a GIT Repository

You will clone the BlackjackProj project to a GIT repository on Developer Cloud Service. Your goal is to clone the BlackjackProj project on your local machine to the GIT repository on Developer Cloud Service. But first, you need to clone the empty GIT repository **from** Developer Cloud Service **to** the local machine, the VM.

1. Launch Git Bash from the icon on the Desktop.

   Note: Git needs to connect to the internet so to get to Developer Cloud Service. As you are connecting to internet on an Oracle University proxy server, you need to provide Git the proxy address of the proxy server. HTTP Proxy is basically a web address you type in to your company's proxy server so you can access the internet.

2. Set https proxy in Git:
   `export https_proxy=https://ges-proxy.us.oracle.com:80/`

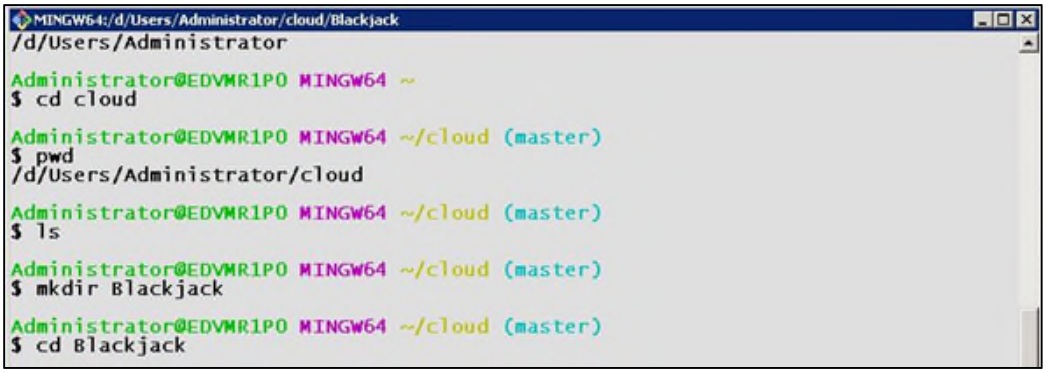


3. In Git, go to the cloud directory:
   `cd /d/Users/Administrator/Cloud`

4. Create a new directory for the Blackjack repository:
   `mkdir Blackjack`

5. Change to the Blackjack directory:
   `cd Blackjack`

6. Confirm you are in the Blackjack directory – cloud/Blackjack. Execute:
   `pwd`
   You should see - /d/Users/Administrator/**Cloud/Blackjack**



7. Get the http URL from the previous lab, “Creating a GIT Repository in Developer Cloud Service” step 5 (on page 12).

8. Clone the DevCS repository to your local git repository. Execute **`git clone`** `<the http URL from DevCS>` as follows:
   At the system prompt, type `git clone`, then right-click and paste the URL from Developer Cloud Service. For example:
   ```
   git clone
   https://susan.jang%40oracle.com@developer.em2.oraclecloud.com/developer34358-ouopc083/s/developer34358-ouopc083_blackjackproj_4583/scm/BlackjackProjRepo.git
   ```

```
Administrator@EDVMR1P0 MINGW64 ~/cloud/Blackjack (master)
$ pwd
/d/Users/Administrator/cloud/Blackjack

Administrator@EDVMR1P0 MINGW64 ~/cloud/Blackjack (master)
$ git clone https://susan.jang%40oracle.com@developer.em2.oraclecloud.com/develo
per34358-ouopc083/s/developer34358-ouopc083_blackjackproj_4583/scm/BlackjackProj
Repo.git
Cloning into 'BlackjackProjRepo'...
remote: Counting objects: 3, done
remote: Finding sources: 100% (3/3)
remote: Getting sizes: 100% (2/2)
remote: Compressing objects: 100% (92/92)
remote: Total 3 (delta 0), reused 0 (delta 0)
Unpacking objects: 100% (3/3), done.
```

   Note: You will be prompted for your **Cloud account password**. If you are not prompted for user name and password and the command fails with a 403 error, then provide the password explicitly in the GIT repository URL.

9. Notice that a **new directory named BlackjackProjRepo** was created in the cloud/Blackjack directory.

10. Change to the BlackjackProjRepo directory:
    `cd BlackjackProjRepo`

    In the next step, you will copy the blackjack-part2 directory and all its files (Scr – Source code for Blackjack, Manifest.json – Configuration file for deploying to OACCS, and Pom.xml – plugs in for Maven) into the local git repository. Recall that this directory was created when blackjack.zip was extracted.

```
Administrator@EDVMR1P0 MINGW64 ~/cloud/Blackjack (master)
$ pwd
/d/Users/Administrator/cloud/Blackjack

Administrator@EDVMR1P0 MINGW64 ~/cloud/Blackjack (master)
$ ls
BlackjackProjRepo/

Administrator@EDVMR1P0 MINGW64 ~/cloud/Blackjack (master)
$ cd BlackjackProjRepo/

Administrator@EDVMR1P0 MINGW64 ~/cloud/Blackjack/BlackjackProjRepo (master)
$ pwd
/d/Users/Administrator/cloud/Blackjack/BlackjackProjRepo

Administrator@EDVMR1P0 MINGW64 ~/cloud/Blackjack/BlackjackProjRepo (master)
$ ls
README.md

Administrator@EDVMR1P0 MINGW64 ~/cloud/Blackjack/BlackjackProjRepo (master)
$
```

11. Copy and paste the **blackjack-part2** project directory (or folder) and all its files from the cloud/Blackjack directory to the BlackjackProjRepo directory. Use the following command, which has a space and a period at the end:

```
cp -r /d/Users/Administrator/Desktop/Cloud/Blackjack/blackjack-part2 .
```

(Don't forget the space and the period at the end).

12. Note the contents of directory **Blackjack-Part2Repo**. You will see that directory **blackjack-part2** is created. Note the contents of that directory.
cd to directory **blackjack-part2.** Note the contents of that directory.

```
Administrator@EDVMR1P0 MINGW64 ~/cloud/Blackjack/BlackjackProjRepo (master)
$ pwd
/d/Users/Administrator/cloud/Blackjack/BlackjackProjRepo

Administrator@EDVMR1P0 MINGW64 ~/cloud/Blackjack/BlackjackProjRepo (master)
$ ls
README.md

Administrator@EDVMR1P0 MINGW64 ~/cloud/Blackjack/BlackjackProjRepo (master)
$ cp -r /d/Users/Administrator/Desktop/Cloud/BlackJack/blackjack-part2 .

Administrator@EDVMR1P0 MINGW64 ~/cloud/Blackjack/BlackjackProjRepo (master)
$ ls
blackjack-part2/  README.md

Administrator@EDVMR1P0 MINGW64 ~/cloud/Blackjack/BlackjackProjRepo (master)
$ cd blackjack-part2/

Administrator@EDVMR1P0 MINGW64 ~/cloud/Blackjack/BlackjackProjRepo/blackjack-par
t2 (master)
$ ls
bin.xml  manifest.json  pom.xml  src/  target/
```

13. Add the source files to GIT from project root directory. Use the following command, which has a space and a period at the end:
`git add .`
(Don't forget the space and the period at the end).

```
Administrator@EDVMR1P0 MINGW64 ~/cloud/Blackjack/BlackjackProjRepo/blackjack-par
t2 (master)
$ git add .
warning: LF will be replaced by CRLF in blackjack-part2/bin.xml.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in blackjack-part2/manifest.json.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in blackjack-part2/pom.xml.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in blackjack-part2/src/main/java/com/exampl
e/blackjack/rest/Application.java.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in blackjack-part2/src/main/java/com/exampl
e/blackjack/rest/GameController.java.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in blackjack-part2/target/maven-status/mave
n-compiler-plugin/compile/default-compile/createdFiles.lst.
The file will have its original line endings in your working directory.
warning: LF will be replaced by CRLF in blackjack-part2/target/maven-status/mave
n-compiler-plugin/compile/default-compile/inputFiles.lst.
The file will have its original line endings in your working directory.
```

14. Commit the changes of the additional source files to the repository:
    `git commit –m “Committing changes to BlackjackProjRepo repository”`

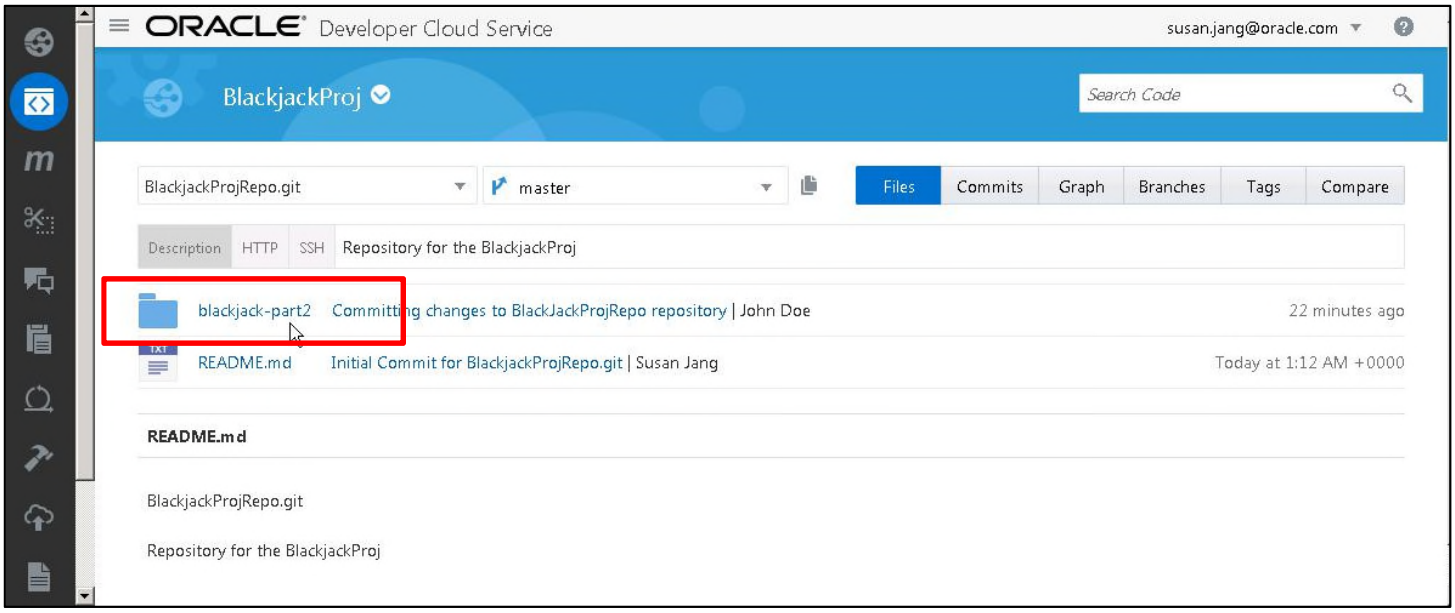
```
Administrator@EDVMR1P0 MINGW64 ~/cloud/Blackjack/BlackjackProjRepo/blackjack-par
t2 (master)
$ git commit -m "Committing changes to BlackJackProjRepo repository"
[master e1c0d05] Committing changes to BlackJackProjRepo repository
 Committer: John Doe <Administrator@edvmr1p0.us.oracle.com>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly:

    git config --global user.name "Your Name"
    git config --global user.email you@example.com

After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

 26 files changed, 503 insertions(+)
 create mode 100644 blackjack-part2/bin.xml
 create mode 100644 blackjack-part2/manifest.json
 create mode 100644 blackjack-part2/pom.xml
 create mode 100644 blackjack-part2/src/main/java/com/example/blackjack/game/Car
d.java
```

15. Push the files to the repository on Developer Cloud Service:
    `git push origin master`
    Wait until all the files are pushed to the repository.
    Note: You will be prompt for your **Cloud account password**.

```
Administrator@EDVMR1P0 MINGW64 ~/cloud/Blackjack/BlackjackProjRepo/blackjack-par
t2 (master)
$ git push origin master
Counting objects: 49, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (37/37), done.
Writing objects: 100% (49/49), 22.86 MiB | 316.00 KiB/s, done.
Total 49 (delta 3), reused 0 (delta 0)
remote: Resolving deltas: 100% (3/3)
remote: Updating references: 100% (1/1)
To https://developer.em2.oraclecloud.com/developer34358-ouopc083/s/developer3435
8-ouopc083_blackjackproj_4583/scm/BlackjackProjRepo.git
   9868be0..e1c0d05  master -> master
```

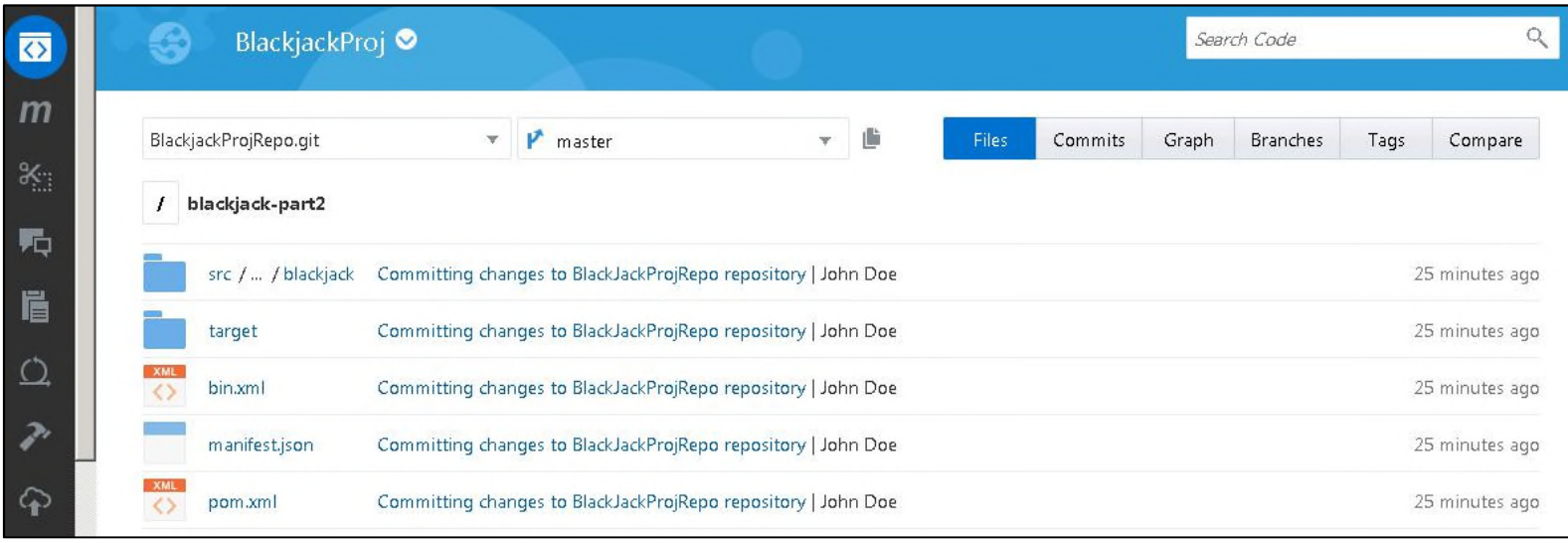
16. Switch to Developer Cloud Service to verify that the files pushed to the repository.

17. In the **BlackjackProj** home page, click **Code** in the left panel. You will see **BlackjackProjRepo.git.** Click **blackjack-part2**.



18. Notice that Blackjackproj project directory has been pushed to repository on Developer Cloud Service. Click it and see its contents.

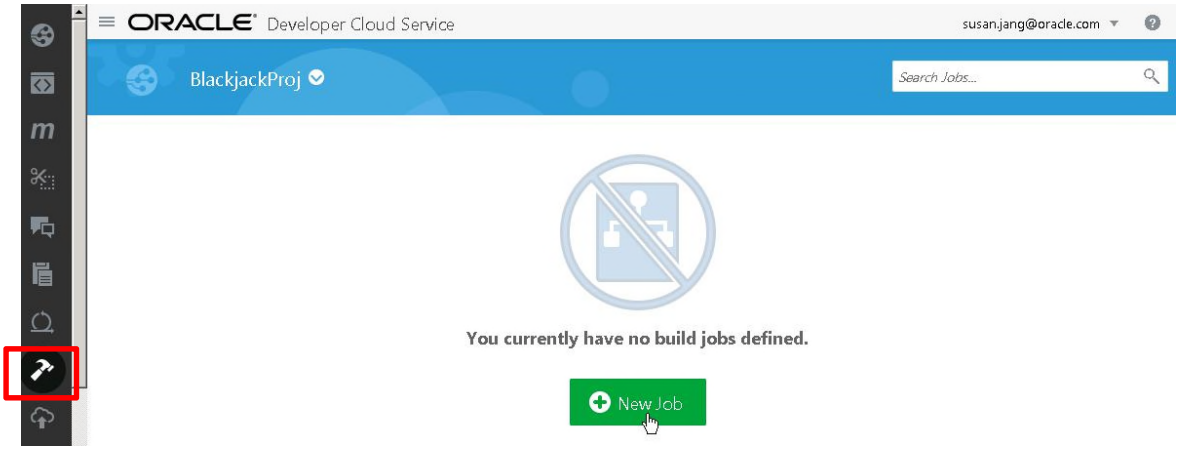
    The contents on the local server is in the local repository.
    The contents on the local repository is now in the git repository in Developer Cloud Service.



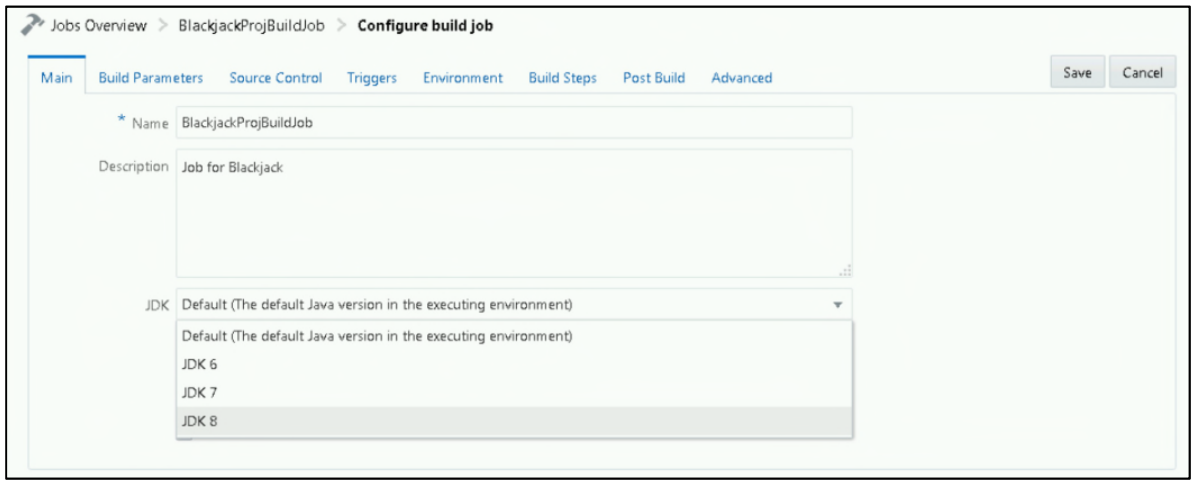
VII. Building a Project on Developer Cloud Service

Build the BlackjackProj project in Developer Cloud Service.

1. In the left navigation panel, click **Build** and then click **New Job**. (To bring up the left panel if it is not already displayed, click on the three horizontal gray lines next to Oracle Developer Cloud Services.



2. In the New Job window, enter **BlackjackProjBuildJob** in the job name field. **Create a free-style job** should be selected. Click **Save**.
3. On the Configure build Job page, do the following:
   a. In the Main tab, enter the following values:
      i. Enter a description (**Job for Blackjack Proj Build**)

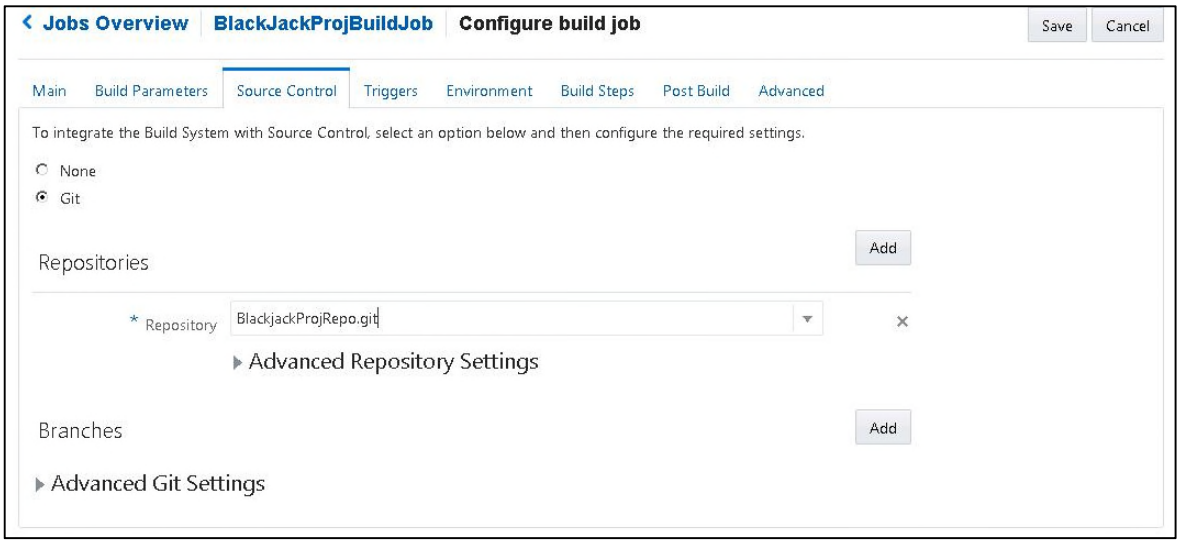


      ii. In the JDK drop-down, set the JDK to **JDK 8**.

b. Click the **Source Control** tab, and then do the following:

   i. Select **Git** as your repository.

   ii. For Repositories, in the drop-down, select **BlackjackProjRepo.git**.



c. Click the **Triggers** tab, and do the following:

   i. Check **Based on SCM polling schedule**.

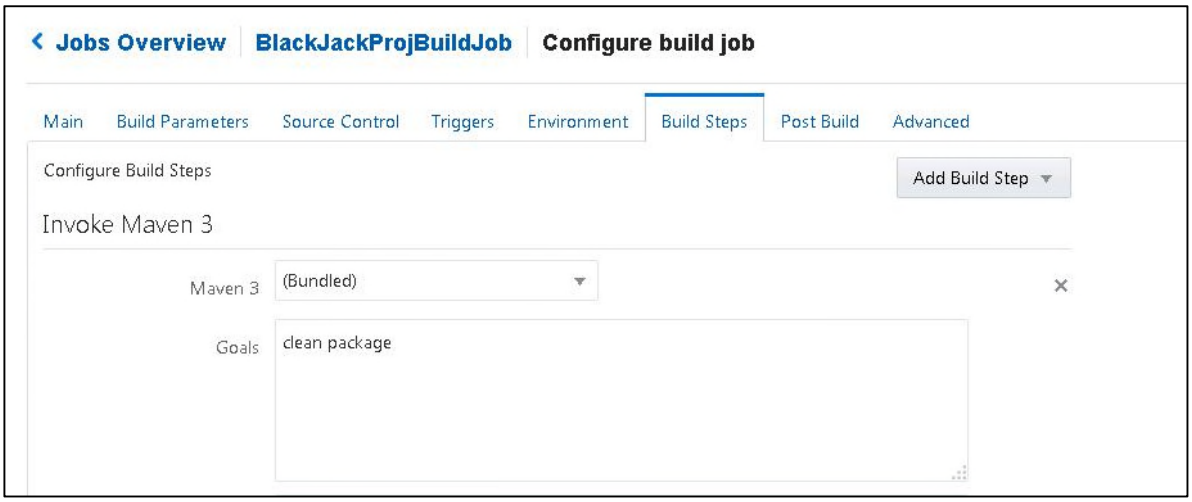
   Note: This setting will enable auto-deploy feature of DevCS. After the application is modified, if changes are committed to GIT repository on DevCS, it will automatically pick up the updated code and deploy it.

d. Click the **Build Steps** tab, and do the following:

   i. Click **Add Build Step** and select **Invoke Maven 3**.

   ii. Set the Goals to: **clean package**.

   iii. Set the POM File location to: **blackjack-part2/pom.xml**



e. Click the **Post Build** tab, and do the following:

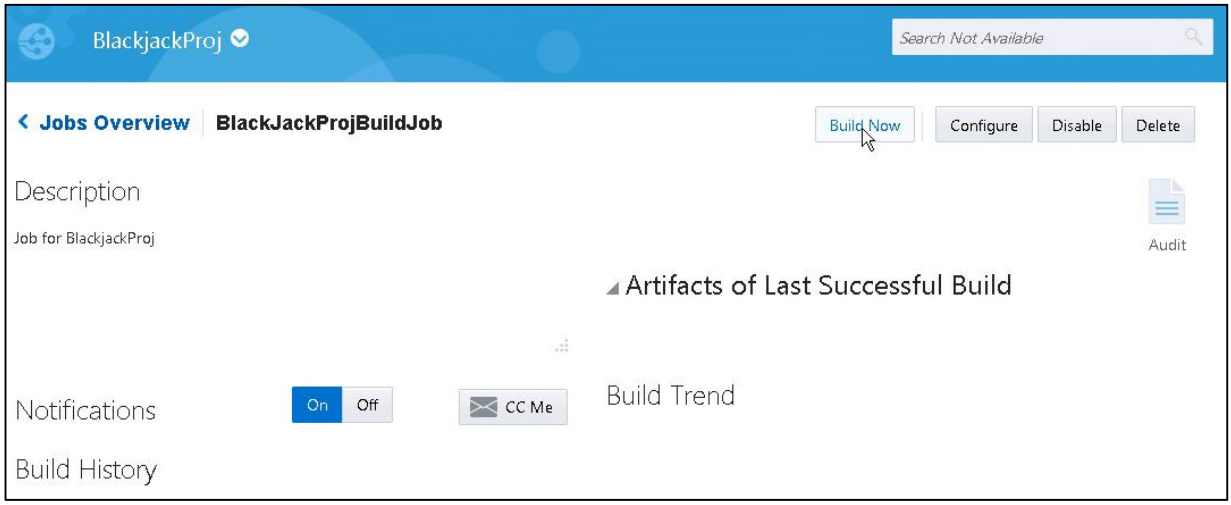
i. Check **Archive the artifacts**.

ii. Set **Files To Archive** to: **blackjack-part2/target/blackjack-part2-1.0.jar, blackjack-part2/target/blackjack-part2-1.0-dist.zip**

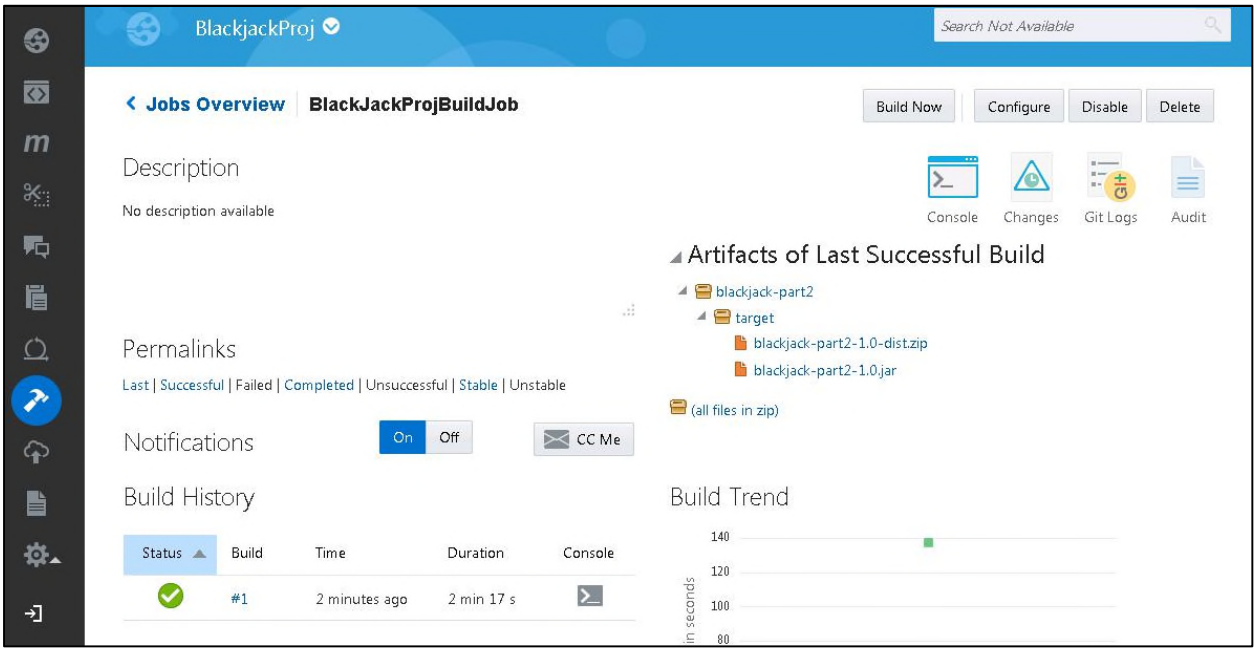
iii. Set Compression Type to **NONE**.

iv. Click **Save**.

4. Click **Build Now** to start the job to begin the application in the Cloud.



If the build was successful, you'll see **two files: blackjack-part2/target/blackjack-part2-1.0.jar** and **blackjack-part2/target/blackjack-part2-1.0-dist.zip** in the **Artifacts of Last Successful Build** section. Give it a few minutes. If you look under the Build History session, you can see it as it is being built. Once the build is completed you will see a green check under Status.



If the build fails, then go back to check the build job configuration, or click **Git Logs** to see more information about the error.

Great work!

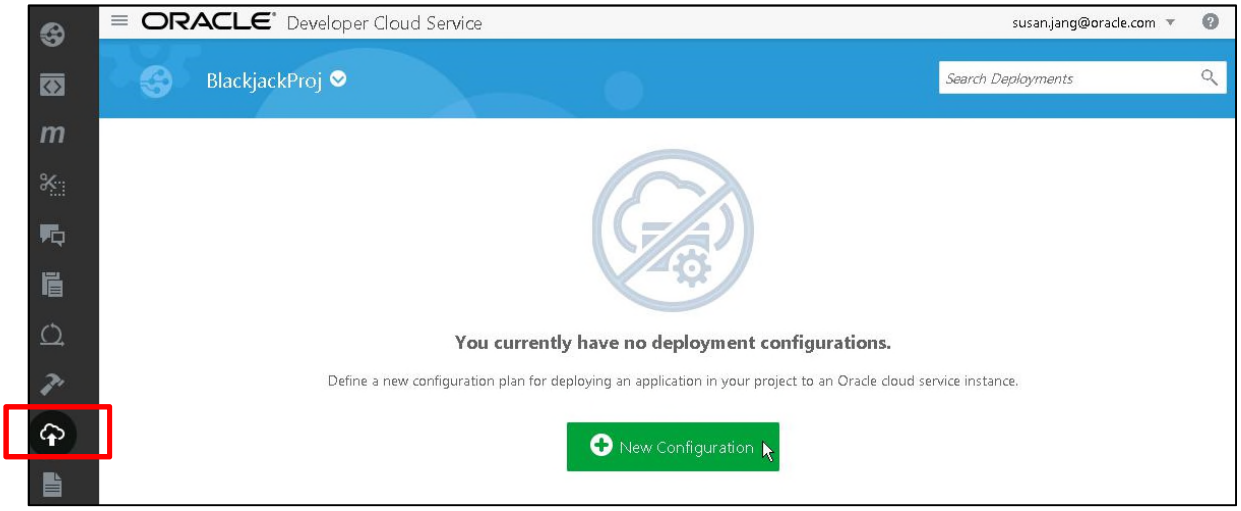
You took the application; the Blackjack game console and deployed on the local server, and now build it into the cloud with Oracle Developer Cloud Service.

Next, you will take what you built in the Cloud with Developer Cloud Service (DevCS) and deploy the Blackjack application to Oracle Application Container Cloud Service (OACCS)

VIII. Deploying a Project to OACCS from DevCS

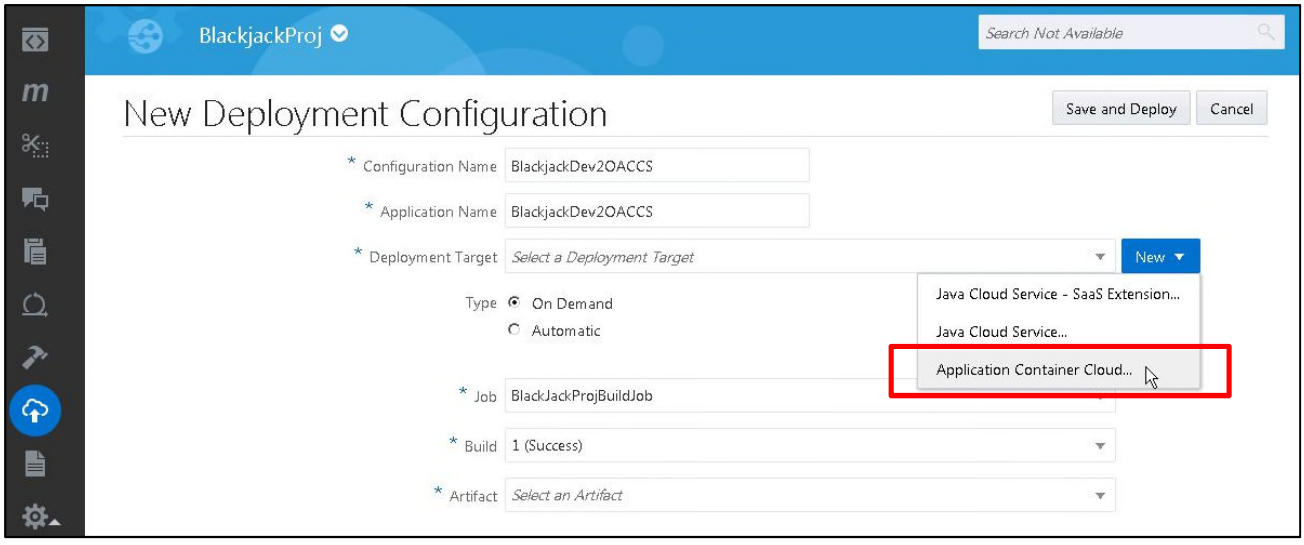
Deploy BlackjackProj to Application Container Cloud Service from Developer Cloud Service.

1. On the BlackjackProj home page, in the left navigation pane, click **Deploy** and then **New Configuration.**



2. In the **New Deployment Configuration** page, enter the **Configuration Name** and **Application Name**.

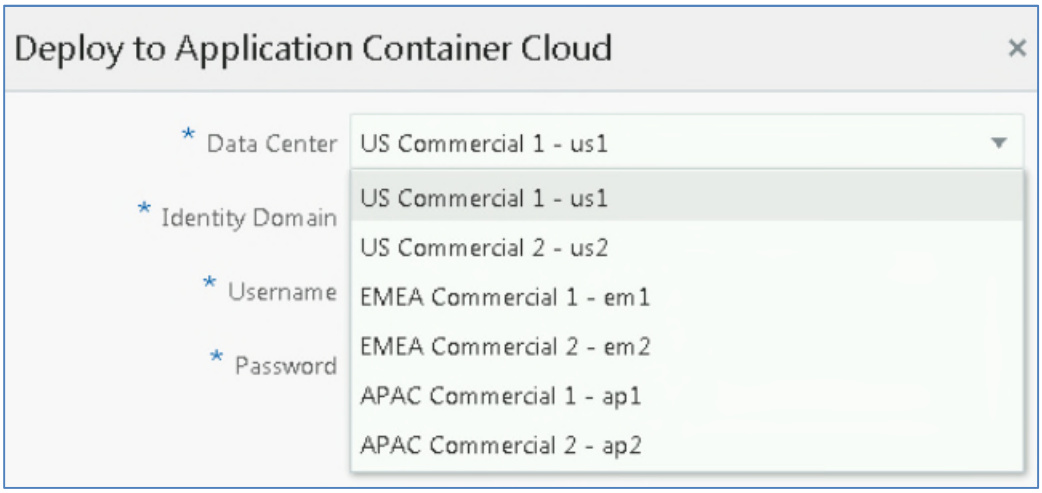
3. In the **Deployment Target** field, click **New** and select **Application Container Cloud.**



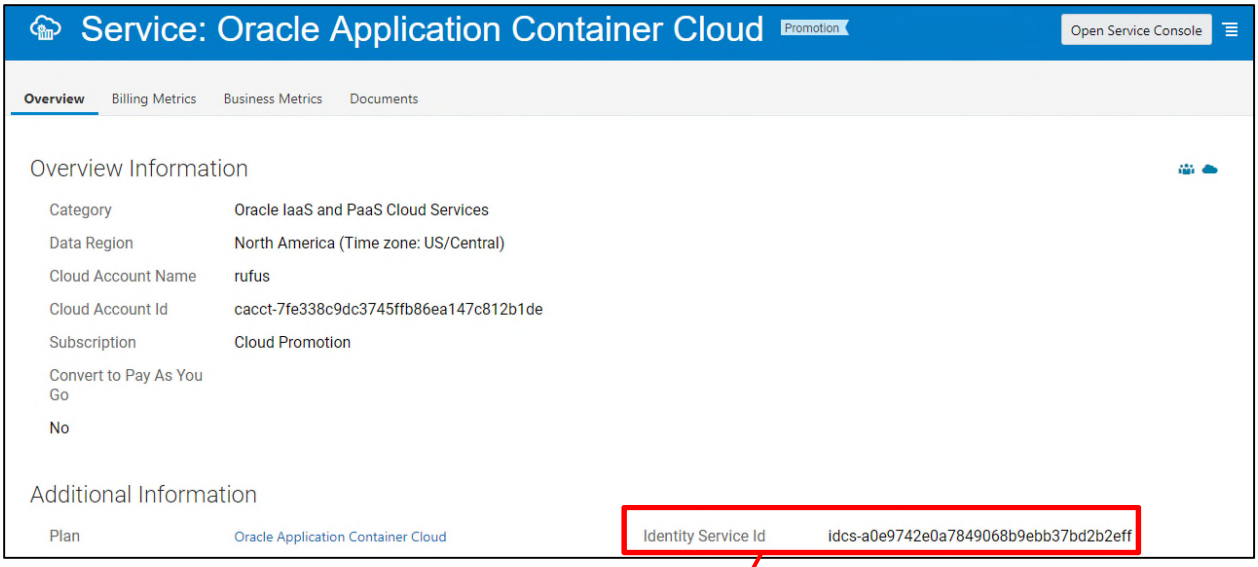
4. In the **Deploy to Application Container Cloud** window do the following:

   a. Enter (Data Center, Identity Domain, Username, Password) with the credentials you received.

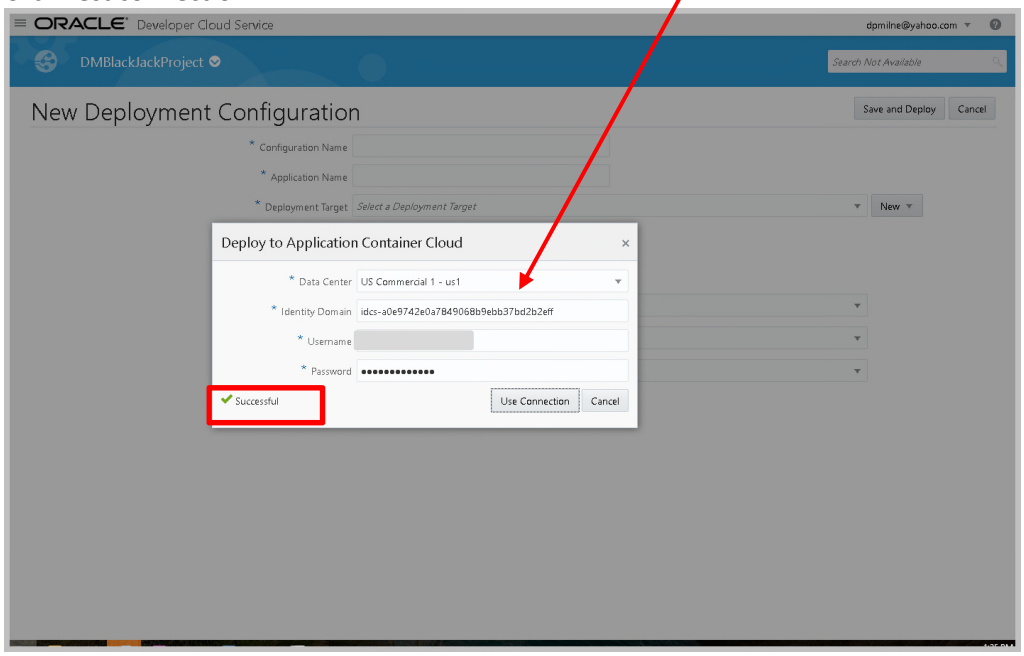
   - *If you were provided a Cloud account by your instructor*, use those credentials.
   - *If you are using a Trial Cloud account*, try the data centers in your region and then confirm successful connection.



   - *If you are using a Trial account*, go back to the Trial Cloud Accounts section, step 7, page 3 where you noted down the Identity Service ID.



b. Click **Test connection**.

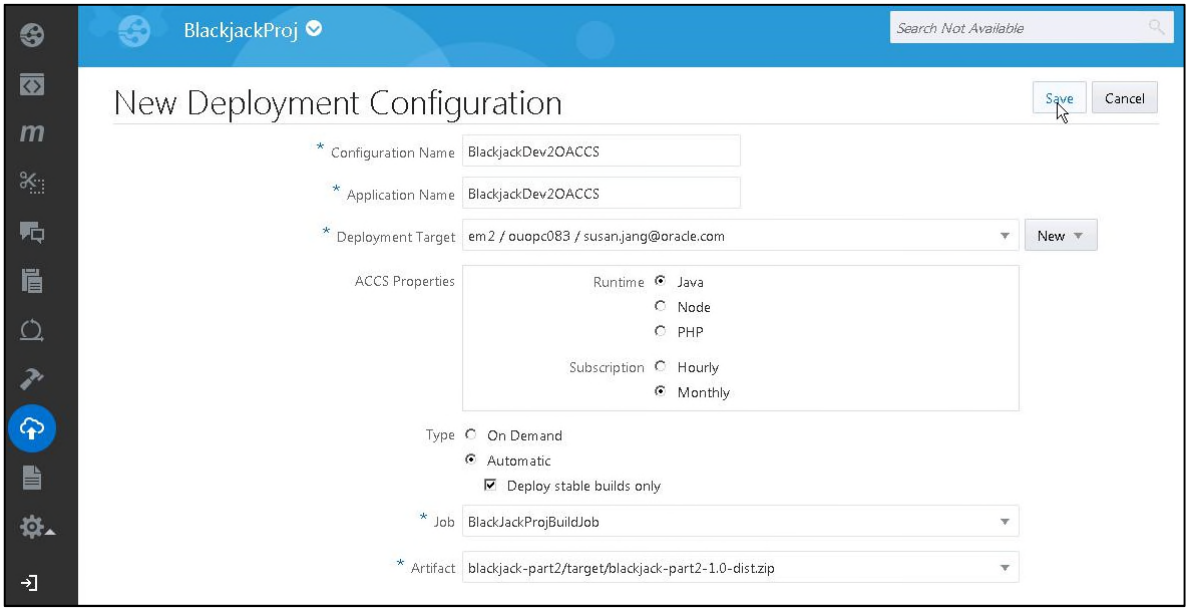


c. When you see the **Successful** message, click **Use Connection**

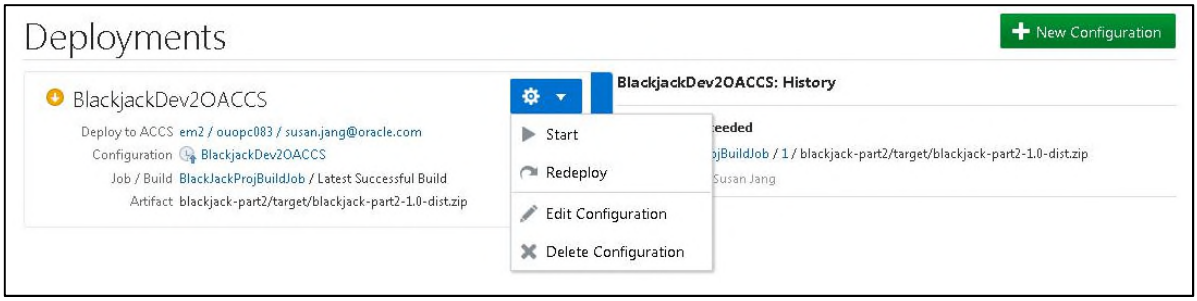
d. Enter the ACCS Properties:

- Runtime: **Java**
- Subscription:
  - *If you are using a Cloud account provided by your instructor, use* **Monthly**.
  - *If you are using a Trial Cloud account, use* **Hourly**.
- Type: **Automatic** (Select **Deploy stable build only**)
- Job: **BlackjackProjBuildJob**
- Artifact: **blackjack-part2/target/blackjack-part2-1.0-dist.zip**

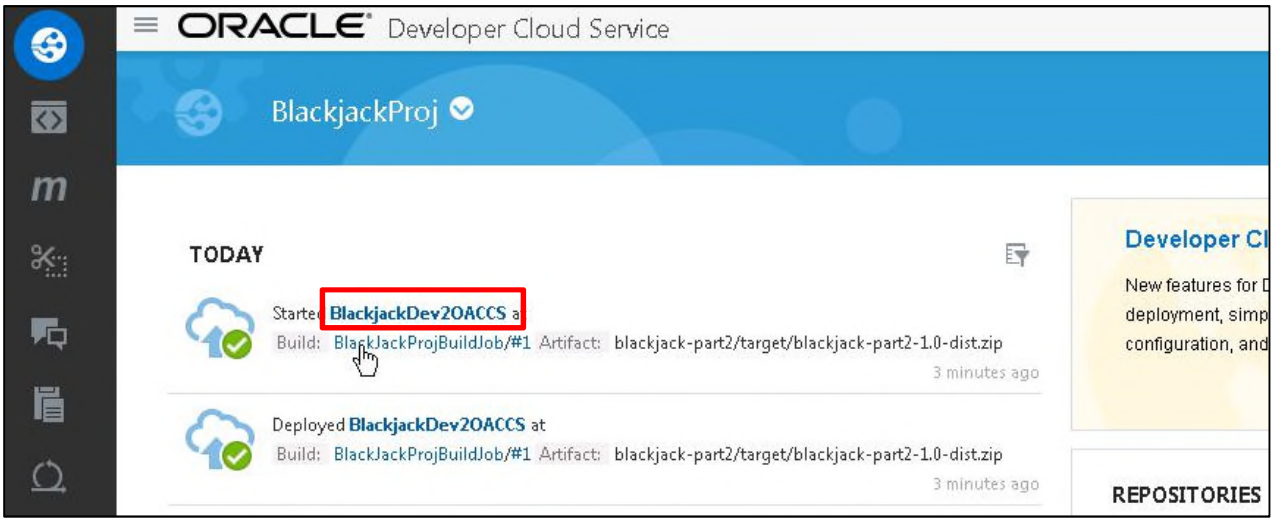
e. Click **Save.**



5. Click the gear icon  and then select **Start** to deploy the application to Oracle Application Container Cloud Service.



6. After the successful deployment, right-click the **BlackjackProj** project name and **copy the URL**.



IX. Testing the Blackjack Application Deployed to OACCS from DevCS

An HTML-5 client application is provided to test the functionality of the deployed Blackjack application.

Use the following instructions to test the Blackjack application.

1. Open a graphical file explorer and navigate to the **Cloud > Blackjack > html5-client** directory.
2. Open the **index.html** file with a browser.
   (file:///D:/Users/Administrator/Desktop/Cloud/Blackjack/html5-client/index.html)
   a. Make sure that the first field, **Service**, is populated with the URL you copied in step 6 of the previous exercise.
      For example, https://blackjackdev2oaccs-ouopc083.apaas.em2.oraclecloud.com/
      Make sure you include the forward slash at the end of the URL.
   b. Enter a number between 1 and 9 in the second field, and then click **Connect**.

3. Once you connect to the gaming console, click the **Debug on/off** button to view the Debug console.

   You can use the Hit and Stand buttons available in the user interface to play the game like before.

Congratulations! You have deployed your application to the Oracle Cloud.

Now you can access your Blackjack gaming console from anywhere you have internet access!

You can continue with the Hands-On Lab session and deploy the Blackjack application directly on the Application Cloud Service.

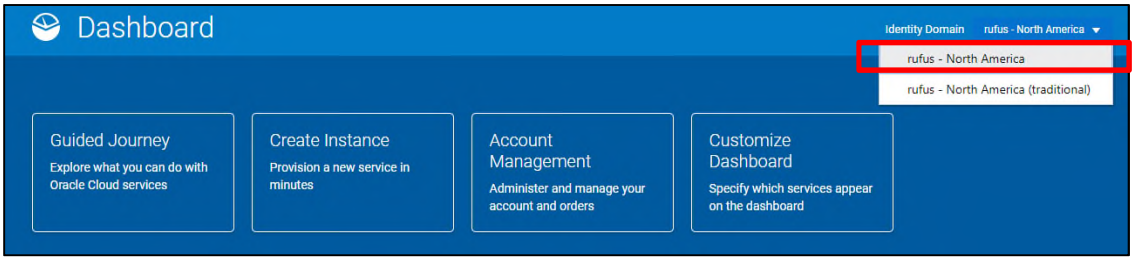
Deploy the Blackjack WebService App Directly to Oracle Application Container Cloud

Overview
This lab guides you through deploying the Blackjack application archive that you created in the previous lab to the Oracle Application Container Service on the cloud.

Prerequisites
Completion of the previous labs

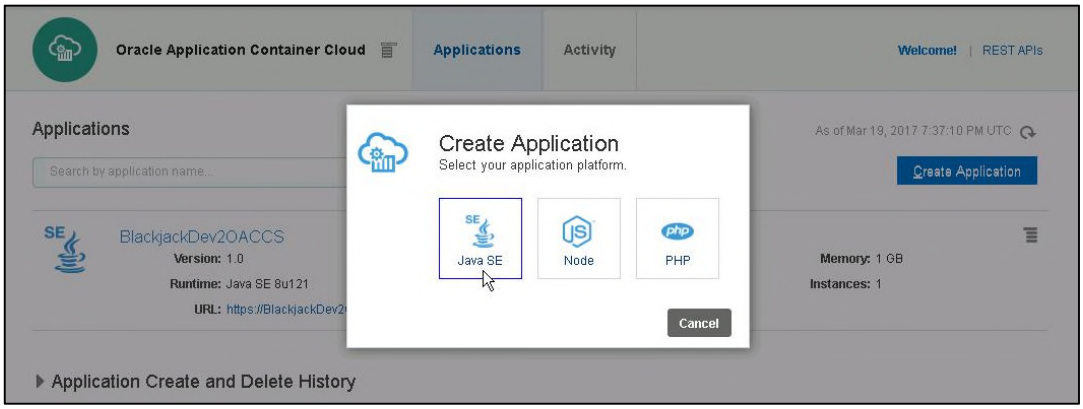
I. Deploy the Blackjack application to OACCS directly.

1. Log into your Cloud account, if you are not already logged in.
2. *If you are using a Trial Cloud account,* change to the Identity Domain (not the "traditional" domain). If not, skip to the next step.



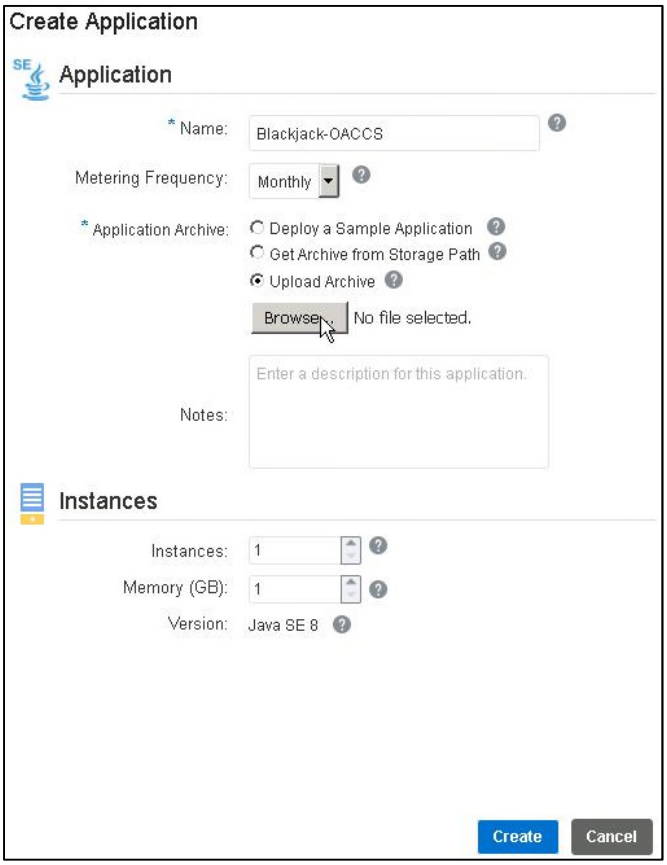
3. On the Dashboard, go to Application Container, click the menu button, and then click the Open Service Console button.

4. Click the **Create Application** button, it will open the Create Application window, and click the **Java SE** icon.

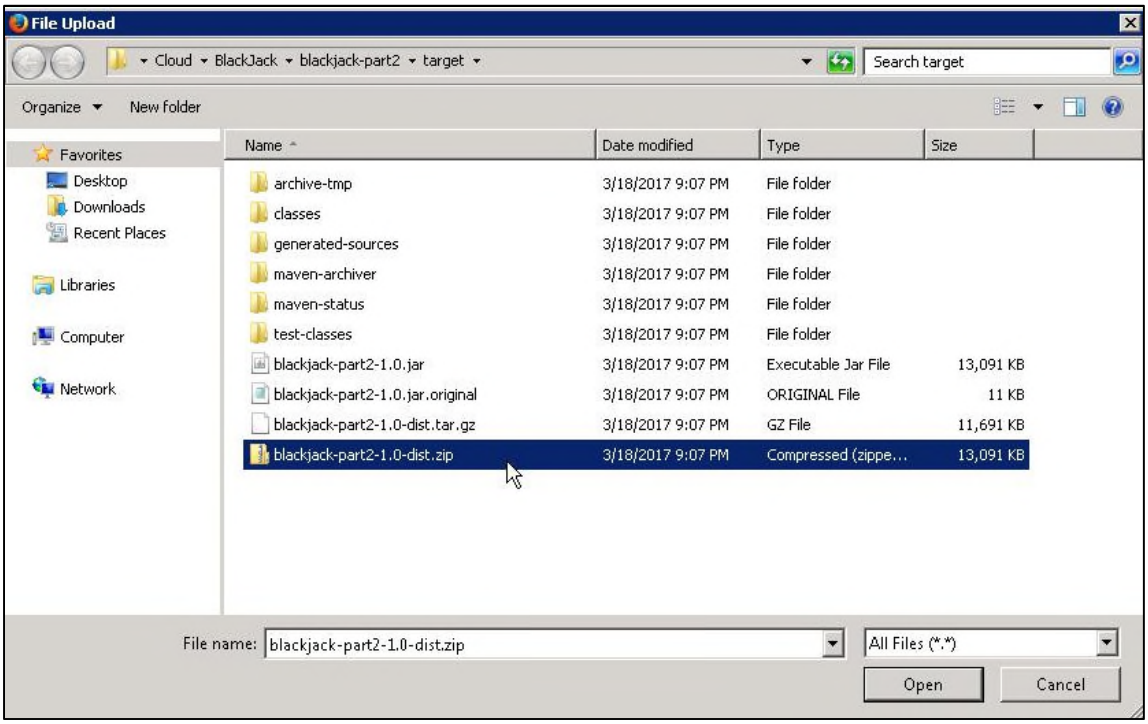


5. In the Create Application dialog box, do the following:
   a. Enter **BlackjackOACCS** for the application name.
   b. Select the default value (Hourly or Monthly) for the subscription type.
   c. For the Application Archive field, select **Upload Archive**.

d. Click **Browse**



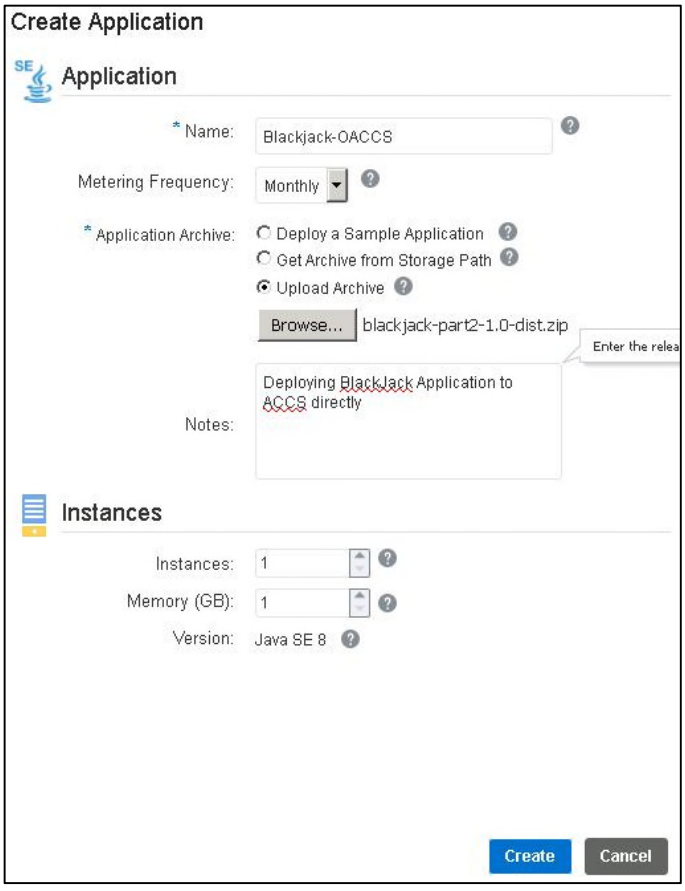
e. Select the **blackjack-part2-1.0-dist.zip** file from the **target directory**



f. Enter **Deploying Blackjack Application to ACCS directly** in the Notes field.

g. Provide the application name of **BlackjackOACCS. (**Some versions of the Cloud user interface do not permit the use of hyphens.)

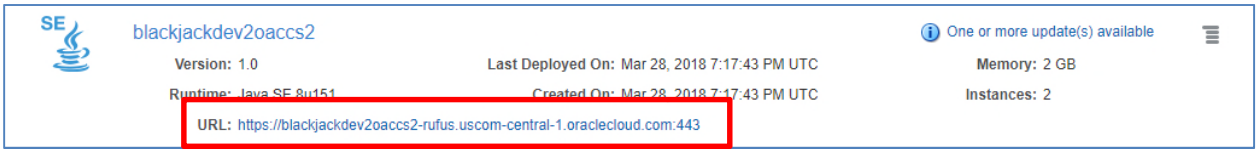
h. Set **Instances** and **Memory** to **1**.



i. The Create Application dialog box now shows
   i. The selected file.
   ii. Under Instance, review the number of instances and the memory size
j. Click **Create** to deploy your application to Oracle Application Container Cloud.

6. A status message should indicate that the Cloud Service is processing the archive. Refresh the browser window until you see that the process is complete.

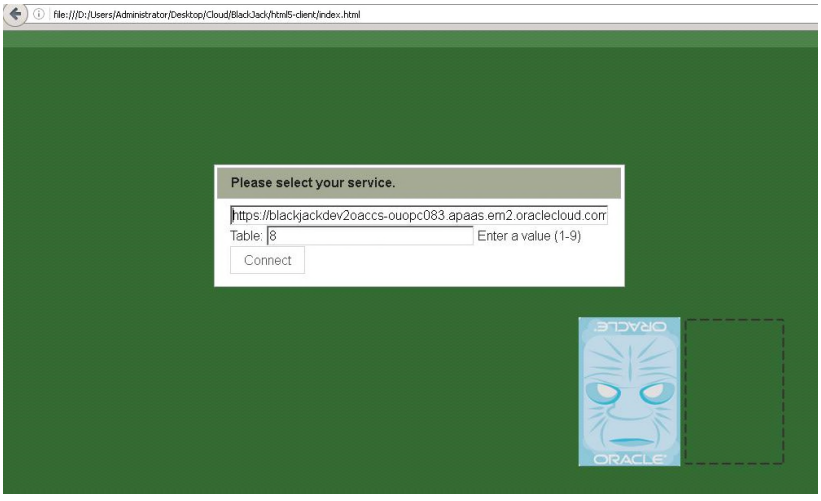
7. Once the application has been deployed successfully. You will not see the In Progress Activity section
   Copy the application URL and paste it in a notepad. We will need this URL for testing purposes.



II. Testing the Blackjack Application Deployed on OACCS

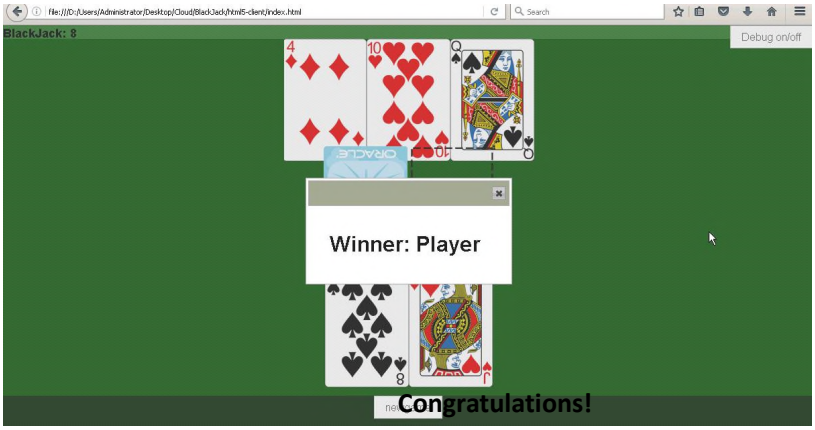
An HTML-5 client application has been supplied to test the functionality of the deployed Blackjack application.

1. Open a graphical file explorer and navigate to the **Cloud > Blackjack > html5-client** directory.
2. Open the **index.html** file with a browser. (Double-click it.)
3. Make sure that the first field, Service, is populated with the URL you copied in the previous exercise.
   For example, **https://blackjack-oaccs-ouopc083.apaas.em2.oraclecloud.com/**
   Make sure you include the forward slash at the end of the URL.
   Enter a number between 1 and 9 in the second field, and then click Connect.



4. Once you connect to the gaming console, click the Debug on/off button to view the Debug console.

You can use the Hit and Stand buttons available in the user interface to play the game.
This HTML5 Client application interacts with the Blackjack gaming application deployed on OACCS in the cloud.



You have successfully completed deploying Blackjack application to OACCS directly using its user interface and testing.