300

Oracle Public Cloud Workshop

Cloud Native Rapid JavaScript Development with Node.js

Contact: Dennis Foley dennis.foley@oracle.com

November 17, 2016

Introduction

This is the third of several labs that are part of the **Oracle Public Cloud Cloud Native Microservices** workshop. This workshop will walk you through the Software Development Lifecycle (SDLC) for a Cloud Native project that will create and use several Microservices.

In the previous lab (200), the Java Developer created several microservices that pull data from twitter and allow for dynamic filtering based on keywords. In this lab, you will assume the role of the front-end JavaScript developer who will create a web application that incorporates the data from those microservices. This node.js application will be developed in the Developer Cloud Service taking advantage of automated builds and deployments to the Application Container Cloud Service.

Please direct comments to: Dennis Foley (dennis.foley@oracle.com)

Objectives

- ☐ Access Developer Cloud Service
- ☐ Import Code from external Git Repository
- ☐ Import Project into Brackets
- ☐ Build and Deploy project using Developer Cloud Service and Oracle Application Container Cloud Service

Required Artifacts

- ☐ The following lab an Oracle Public Cloud account that will be supplied by your instructor. You will need to download and install latest version of Brackets text editor.

Outline

Introduction 2
Objectives 2
Required Artifacts 2
Outline 3

Create Initial Twitter Marketing UI Service 4
Explore Developer Cloud Service 4
Create Initial Git Repository 10
Create Default Build and Deployment Process 12
Verify default deployment of Twitter Marketing UI 20

Extend default application to Display Twitter Feed 24
Clone Project to Brackets Text Editor 25
Add Code to display Twitter Feed in Table Format 30

Create a new Branch and Commit Code 38
Create a Branch and Commit Code 38
Create Merge Request 42
Merge the Branch as Lisa Jones 44
Test the JETFrontEndAPP UI in the Cloud 49

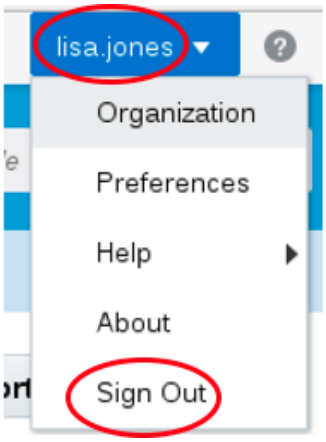
Appendix 1 – Installing Brackets and Git 52
Download and Install Git 52
Download and Install Brackets Text Editor 55
Start Brackets and Configure Git 57

Create Initial Twitter Marketing UI Service

Explore Developer Cloud Service

STEP 1: Login to your Oracle Cloud account

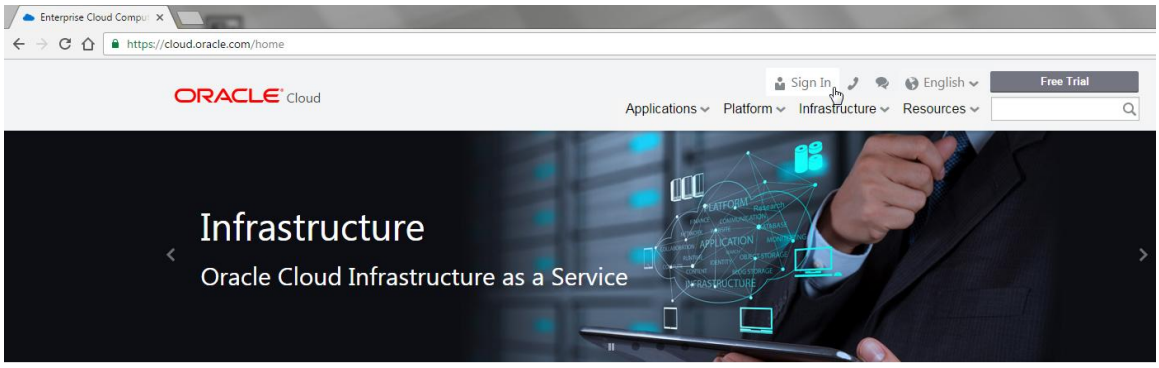
☐ If you just completed lab 200, or if you are still logged in as Lisa.Jones, you will need to first sign out before continuing this lab. Sign out by clicking on the user's name (lisa.jones) at the top right corner of the screen, then selecting **Sign Out** from the dropdown menu.



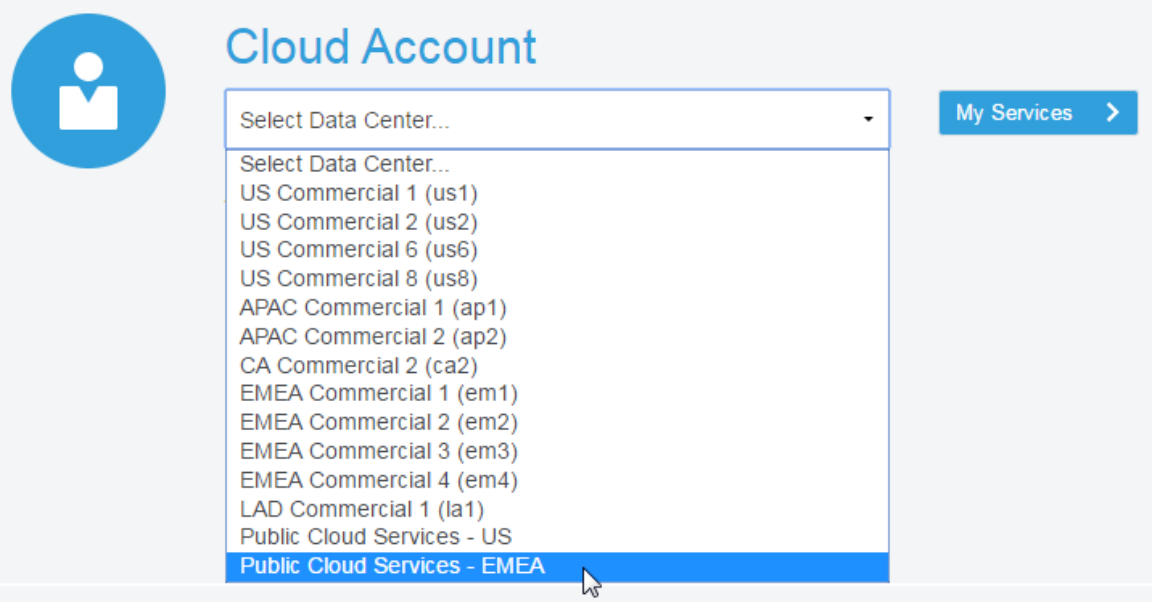
☐ Now we can login again. From any browser, go to the following URL:

```
https://cloud.oracle.com
```

☐ Click **Sign In** in the upper right hand corner of the browser

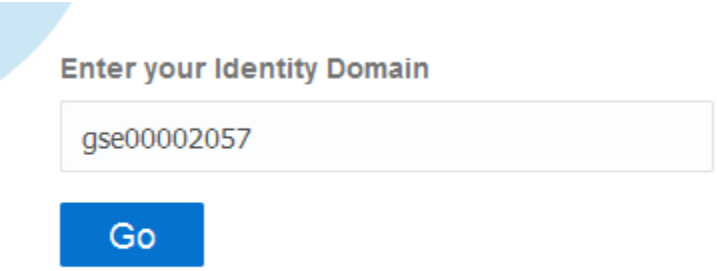


☐ **IMPORTANT** - Under My Services, ask your instructor which **Region** to select from the drop down list, and **click** on the **My Services** button.



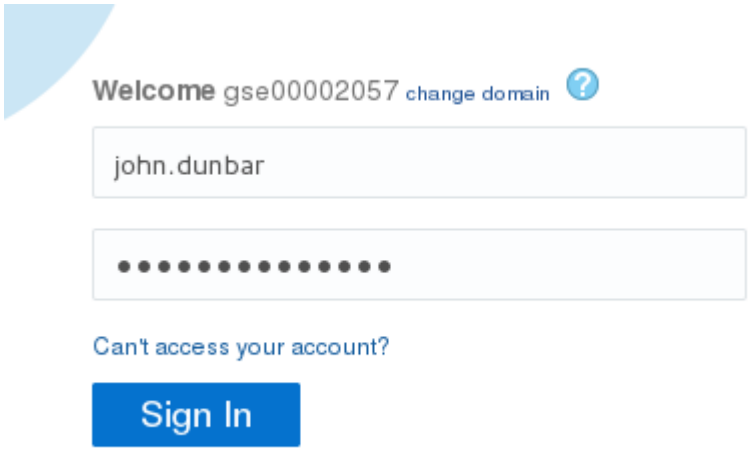
☐ Enter your identity domain and click **Go**

NOTE: the **Identity Domain, User Name** and **Password** values will be given to you from your instructor.

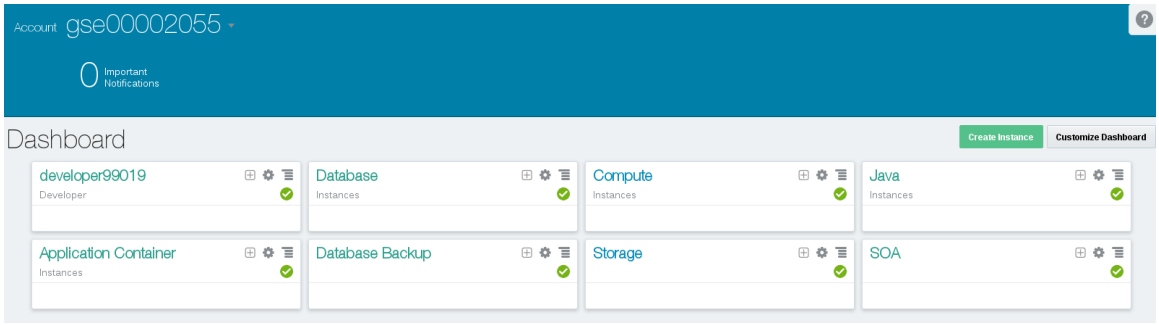


☐ Once your Identity Domain is set, enter your User Name and Password and click **Sign In**

NOTE: For this lab you will be acting as the JavaScript Developer **John Dunbar**. As with the previous lab, if you are not able to support multiple users, login as a supported user, and assume the "logical" identify of John Dunbar, the JavaScript Developer.



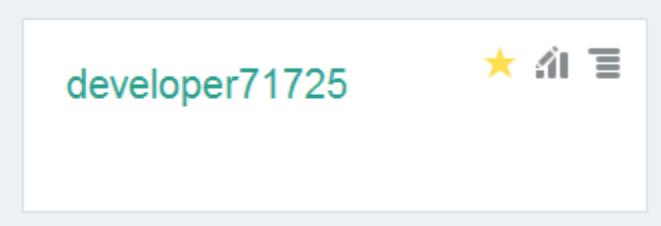
☐ You will be presented with a Dashboard displaying the various cloud services available to this account.



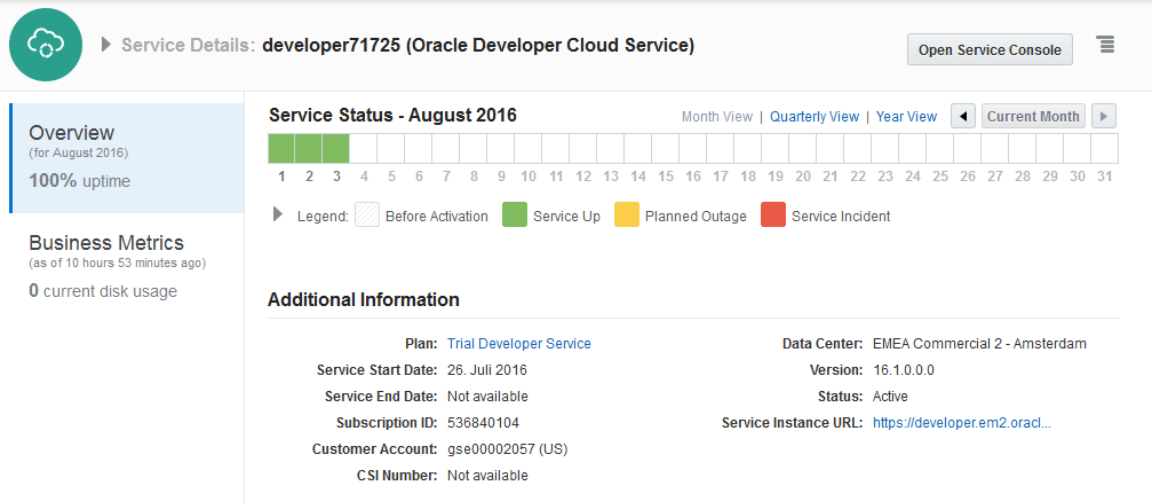
STEP 2: Login to Developer Cloud Service

Oracle Developer Cloud Service provides a complete development platform that streamlines team development processes and automates software delivery. The integrated platform includes issue tracking system, agile development dashboards, code versioning and code review platform, continuous integration and delivery automation, as well as team collaboration features such as wikis and live activity stream. With a rich web based dashboard and integration with popular development tools, Oracle Developer Cloud Service helps deliver better applications faster.

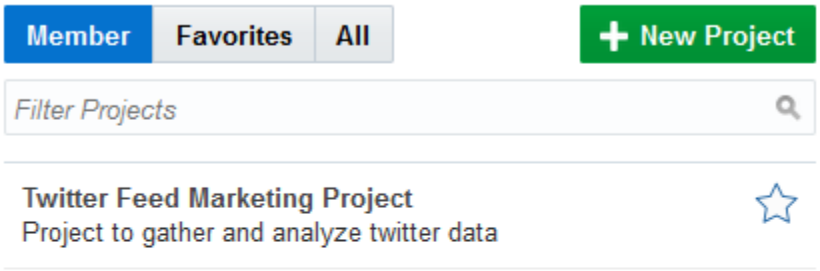
☐ From Cloud UI dashboard click on the **Developer** service. In our example the Developer Cloud Service is named **developer71725**.



☐ The Service Details page gives you a quick glance of the service status.

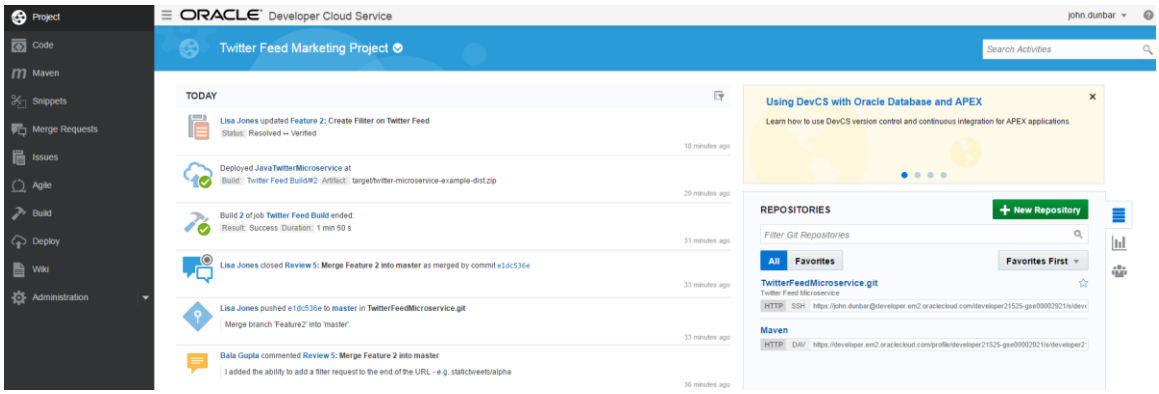


☐ Click **Open Service Console** for the Oracle Developer Cloud Service. The Service Console will list all projects that you are currently a member.

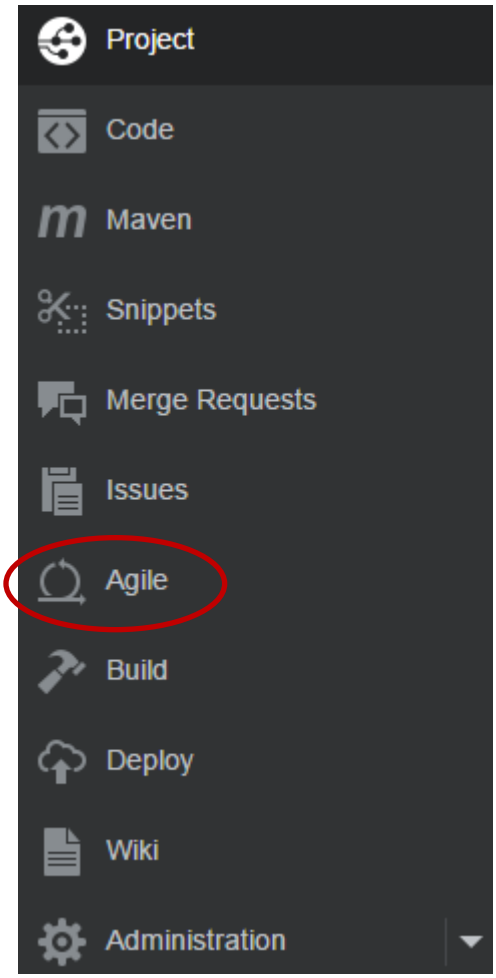


STEP 3: Review Agile Board

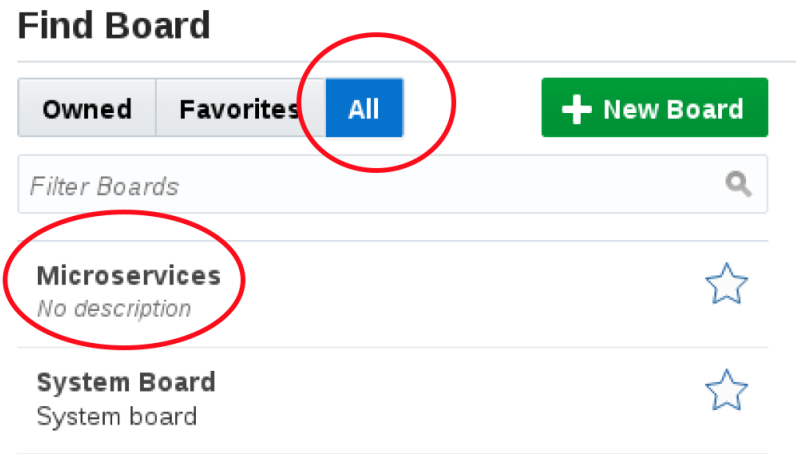
☐ Click **Twitter Feed Marketing Project** to access the project.



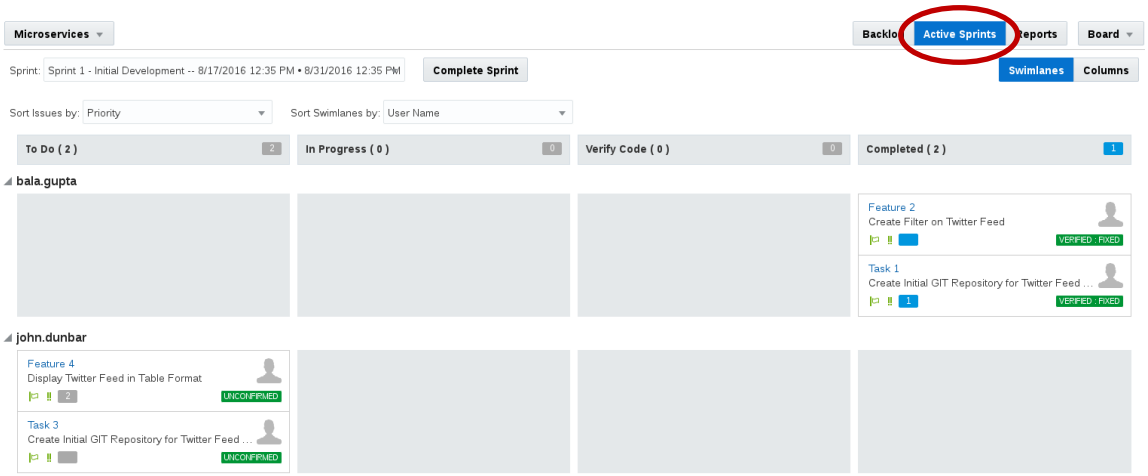
☐ Click on **Agile** in navigation panel.



☐ If the **Microservices** is not the default board, click on the current board's dropdown, select the filter **All**, and click on **Microservices**



☐ Click on the **Microservices** Board **Active Sprints**.

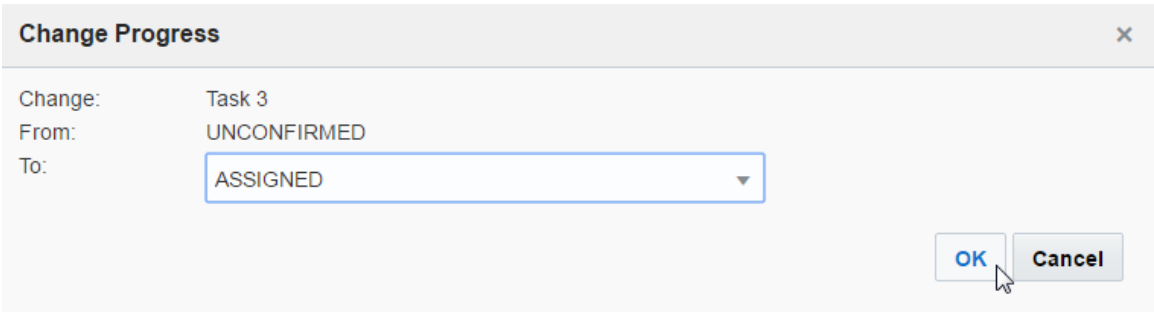


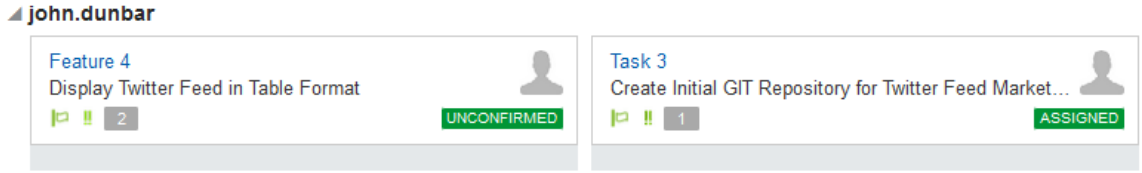
Create Initial Git Repository

STEP 4: Create Initial Git Repository

As in the previous lab, we could start coding this application from scratch at this point. However, one of our colleagues has already begun working on the shell for our web application outside of the Developer Cloud Service. We want to use his work as a starting point and extend it to incorporate our twitter microservices. To pull his code into the Developer Cloud Service, we will clone his external GIT repository. First let's update our agile board to show that we are working on this task:

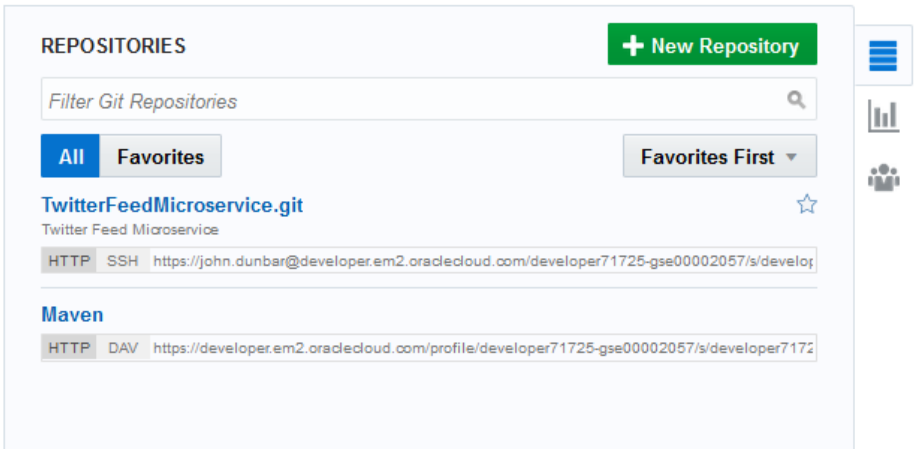
☐ Drag and drop **Task 3 - Create Initial GIT Repository for Twitter Marketing UI** into the **In Progress** swim-lane. Click **OK** on Change Progress popup.





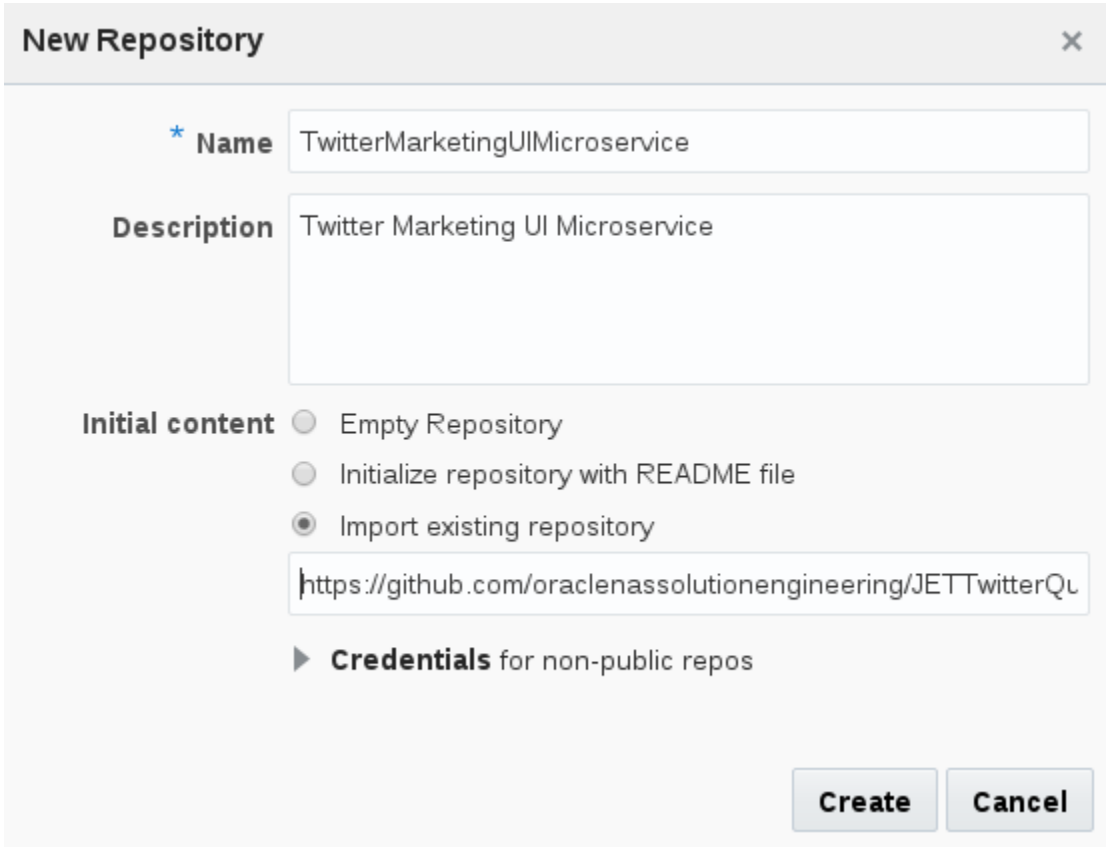
☐ Click on **Project**.

☐ Click on **New Repository** to create a new Git Repository

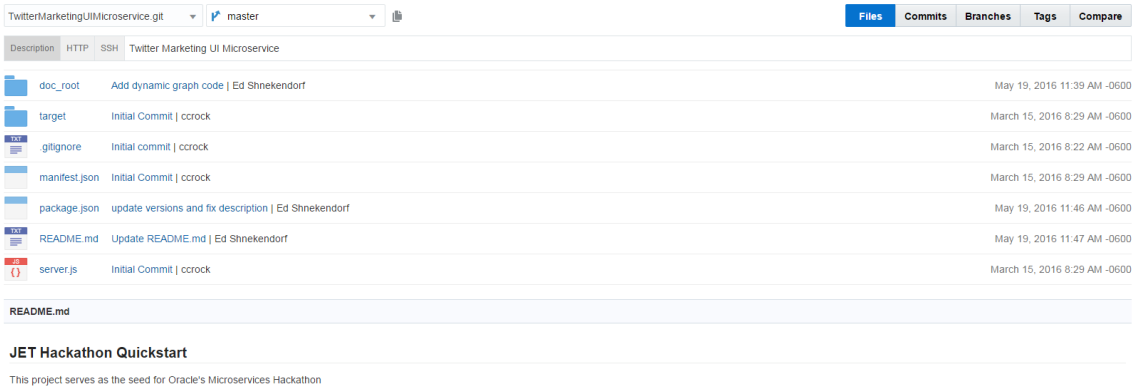


☐ In the New Repository wizard enter the following information and click **Create**.

```
Name = TwitterMarketingUIMicroservice
Description = Twitter Marketing UI Microservice
Initial content = Import existing repository
  and enter the URL:
https://github.com/oraclenassolutionengineering/JETTwitterQuickStart.git
```



☐ You have now created a new GIT repository based on an existing repository.



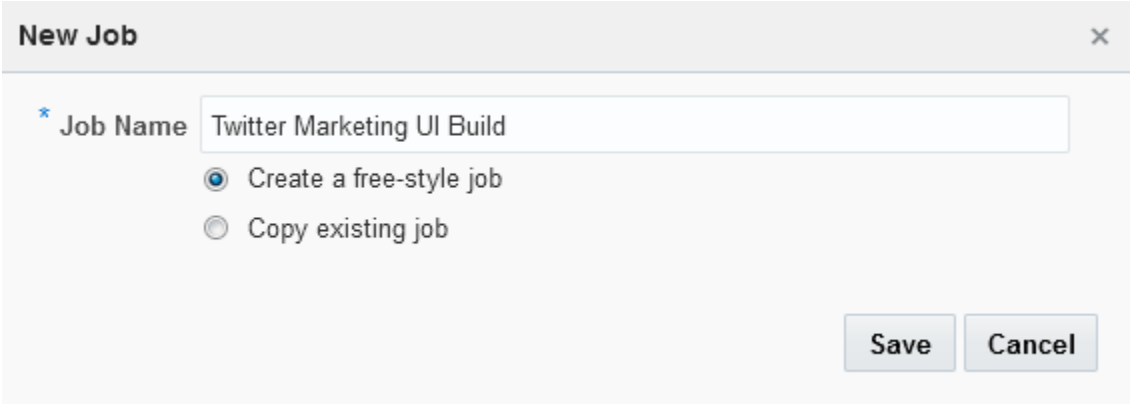
Create Default Build and Deployment Process

STEP 5: Create Default Build Process

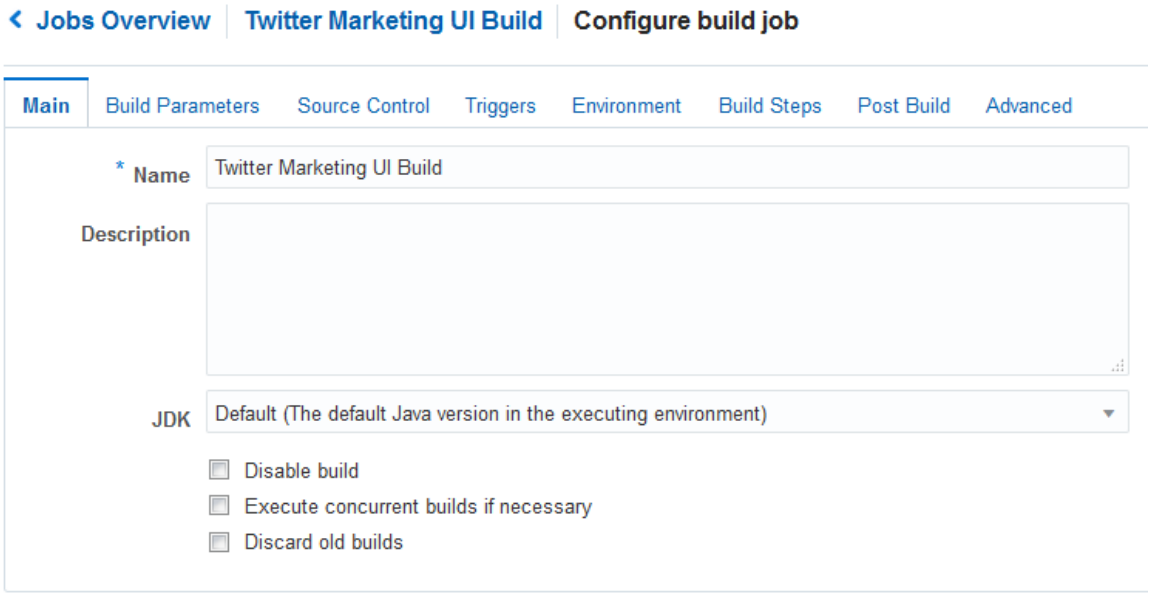
Now that we have the source code in our managed GIT repository, we need to create a build process that will be triggered whenever a commit is made to the master branch. We will set up a shell script build process in this section.

☐ Click **Build** to access the build page and click **New Job**.

☐ In the New Job popup enter **Twitter Marketing UI Build** for Job Name and click **Save**.

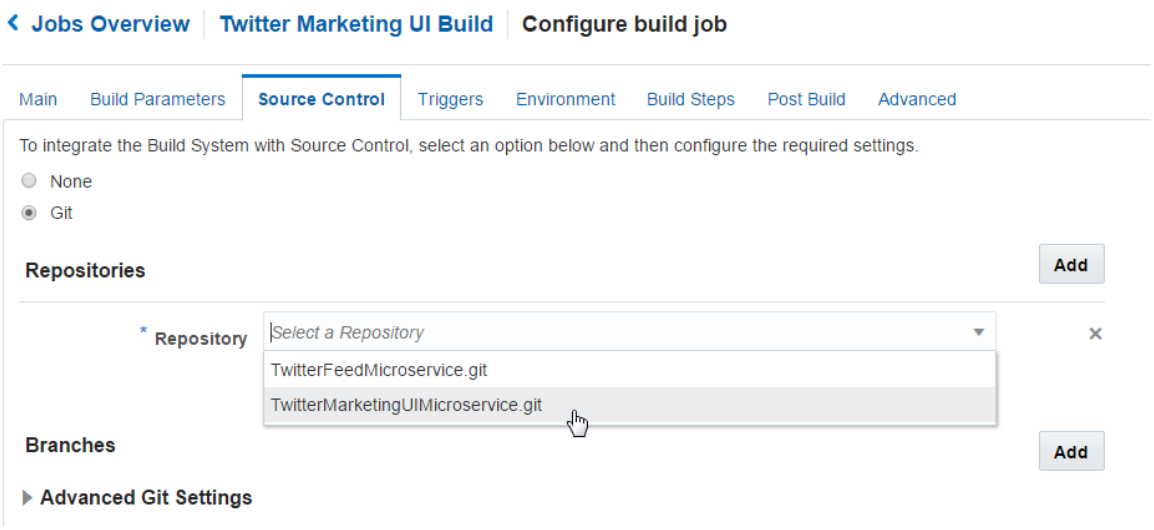


☐ You are now placed into the job configuration screen.



☐ Click the **Source Control** tab. Click **Git** and select **TwitterMarketingUIMicroservice.git** from the URL drop down.

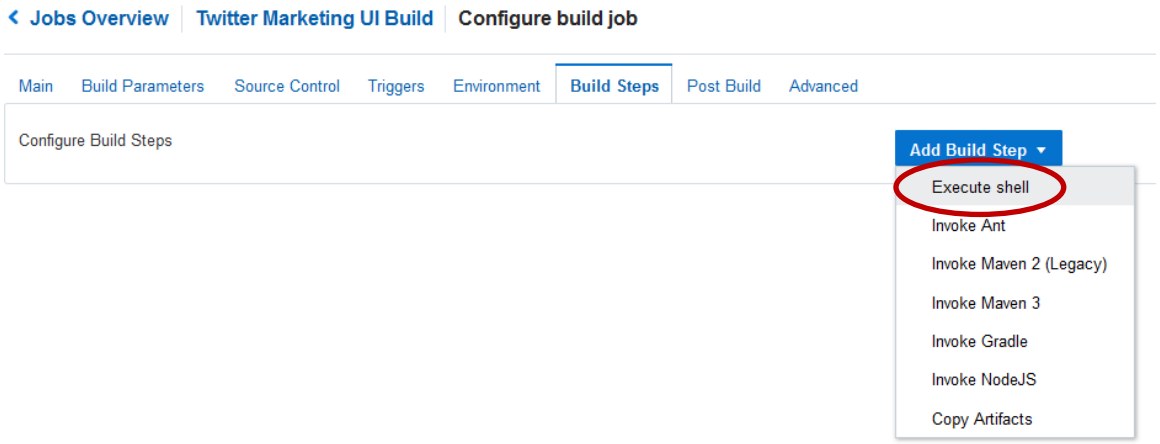
Note: Make sure you select the Git repository for the Twitter Marketing UI Microservice.



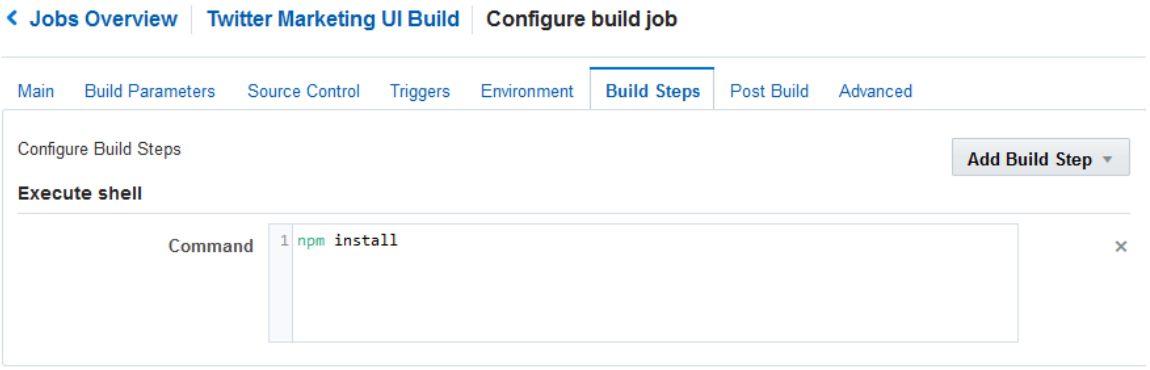
☐ Click the **Triggers** tab. Select Based **on SCM polling schedule,** and add a schedule: **`*/1****`**

Note: The above expression results in the repository being polled every minute to check for any changes. If there are changes, the build will trigger.

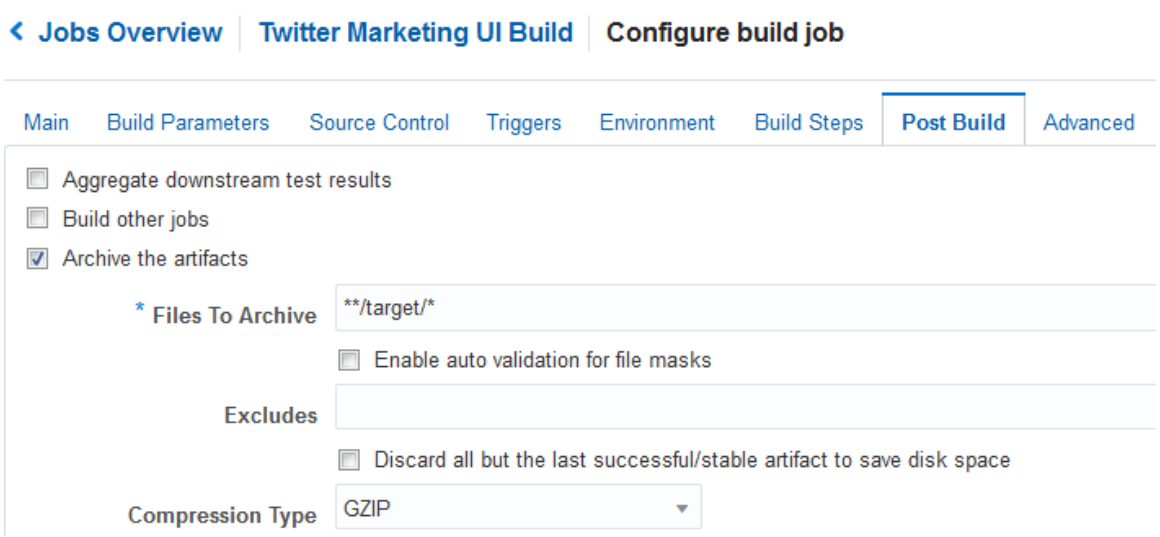
☐ Click the **Build Steps** tab. Click **Add Build Step,** and select **Execute shell**.



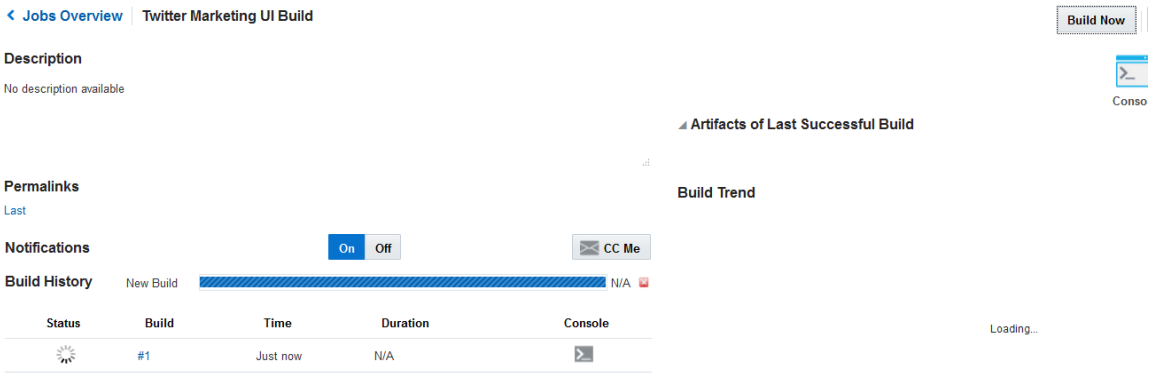
☐ For **Command** enter **npm install**



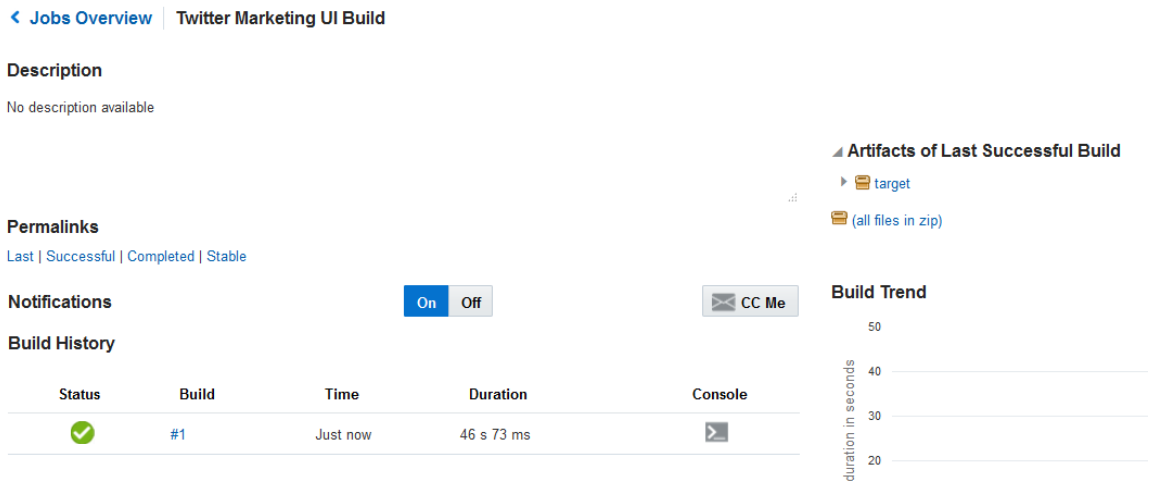
☐ Click the **Post Build** tab. Check **Archive the artifacts** and enter **`**/target/*`** for Files to Archive. Verify **GZIP** in the Compression Type.



☐ Click **Save** to complete the configuration. A build should start automatically within a minute or two. If it does not start automatically, click on the **Build Now** button. The status will change to the following:



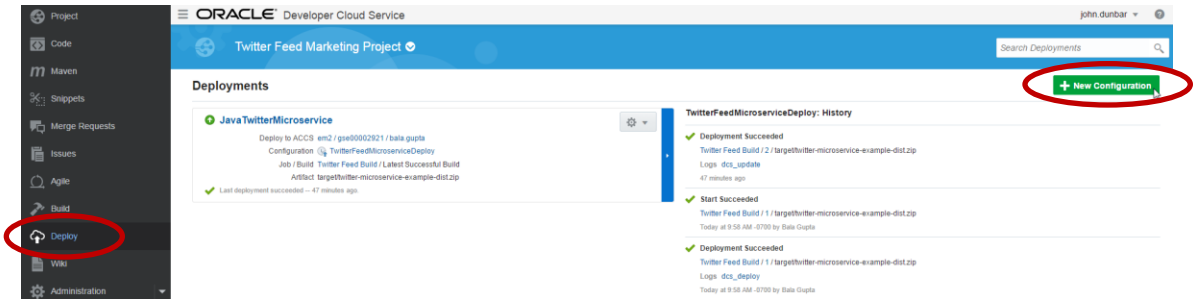
☐ Once the build begins, it should take about approximately 1 to 2 minutes for the build to complete. Wait for the build to complete before continuing on to the next step, as we need the build artifact to create the deployment configuration.



STEP 6: Create Default Deployment Process

Now that we have an automated build process, we will set up a deployment configuration that will push our build artifacts to a node.js environment running on the Application Container Cloud Service whenever a successful build occurs.

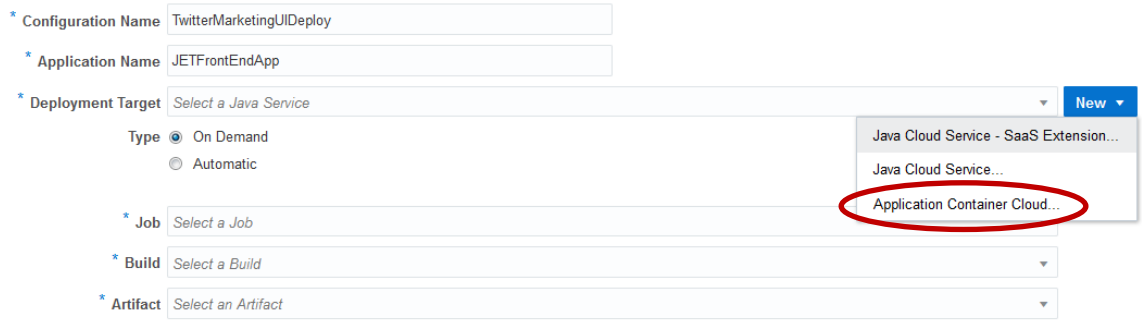
☐ Click Deploy to access the Deployment page and click **New Configuration**.



☐ Enter the following data:

```
Configuration Name = TwitterMarketingUIDeploy
Application Name = JETFrontEndApp
```

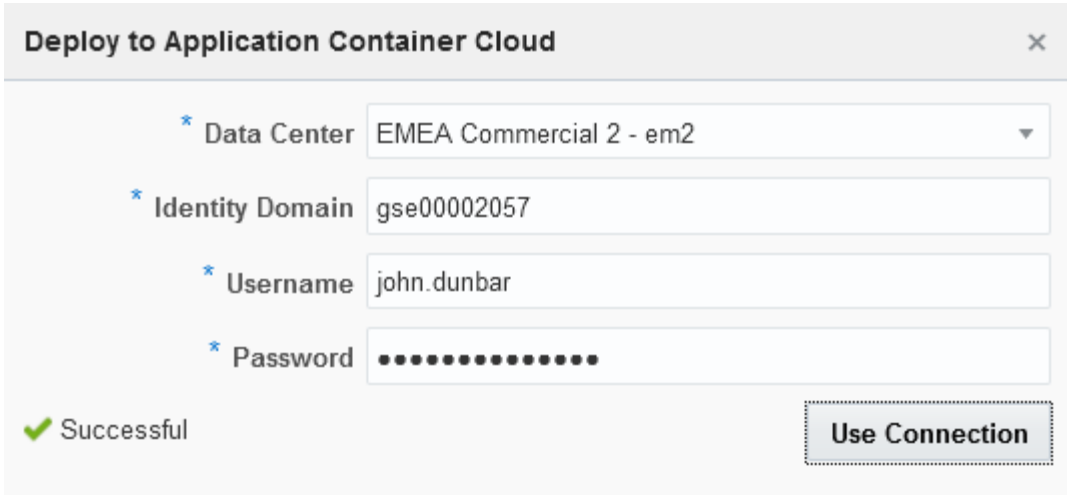
☐ Right of Deployment Target click **New** and select **Application Container Cloud**



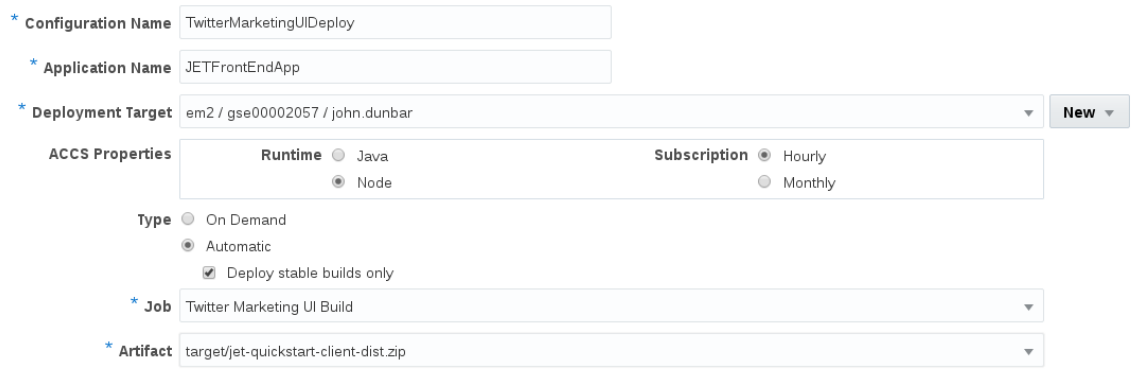
☐ Enter the following data and click **Test Connection**. If Successful click **Use Connection**

```
Data Center = EMEA Commercial 2 - em2
  ** or your appropriate Data Center, if not EMEA Commercial 2
Identity Domain = <You Identity Domain>
Username = john.dunbar
```

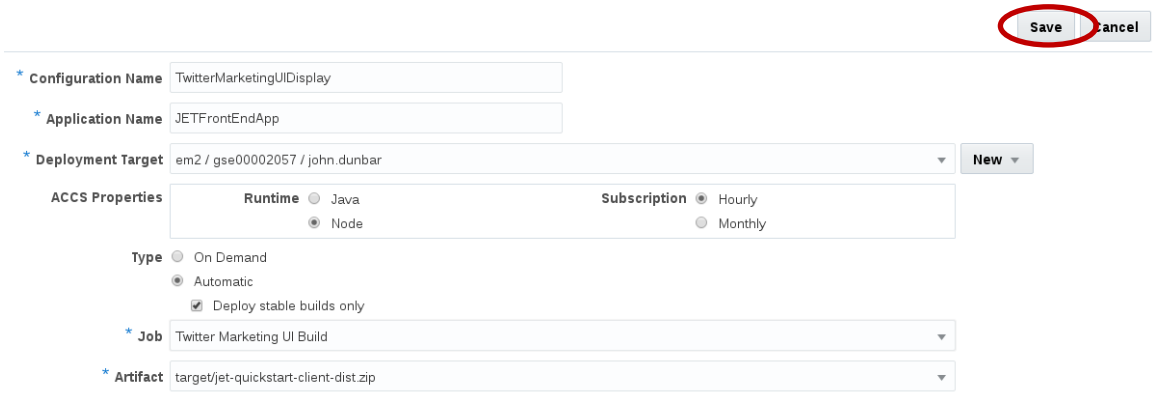
```
Password = <Supplied Password>
```



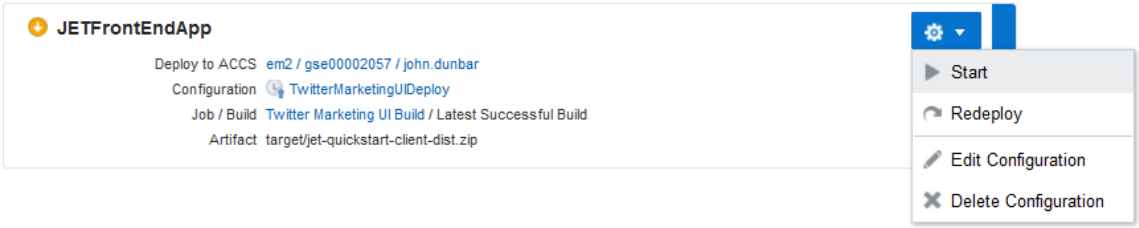
☐ Set ACCS Properties to Runtime **Node** and Subscription **Hourly**. Click Type **Automatic**. Select Job **Twitter Marketing UI Build** and select **target/jet-quickstart-client-dist.zip** for Artifact.



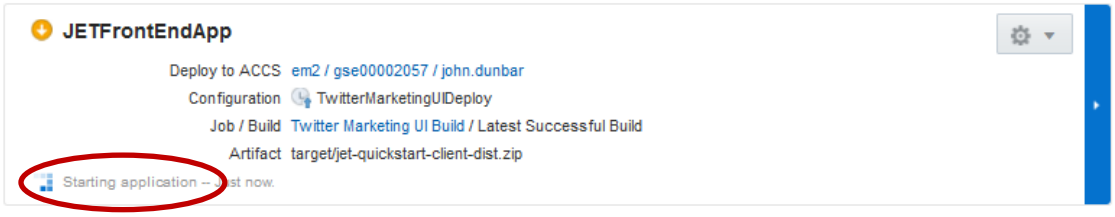
☐ Click **Save**

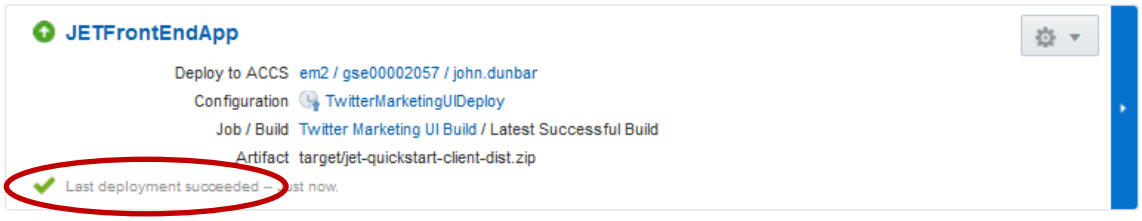


☐ Click drop down and select **Start**



☐ Wait until the message **Starting application** changes to **Last deployment succeeded**



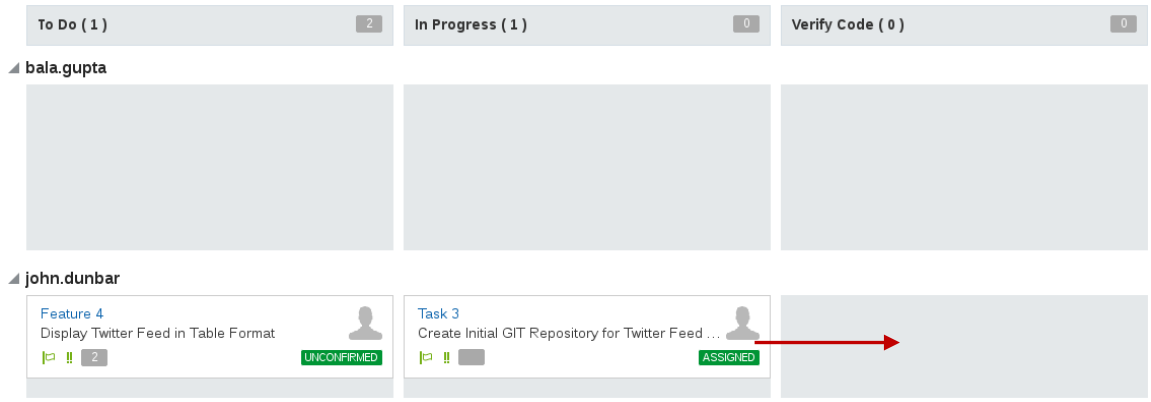


Verify default deployment of Twitter Marketing UI

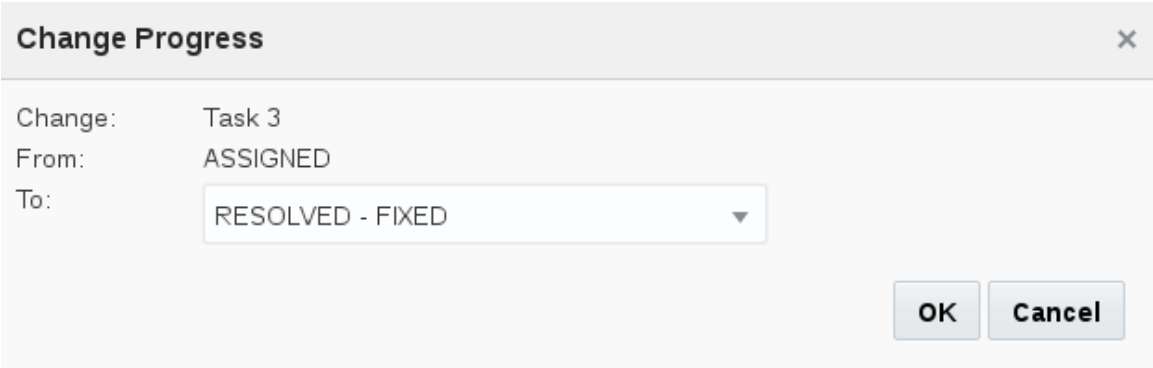
STEP 7: Change status to Verified

Now that we have successfully deployed the build artifact to the Application Container Cloud Service, we will update our agile board to reflect that status. Although the complexity of the next task (verification) is quite simple, we will still move the task to the "Verify Code" column before manually verifying the new functionality.

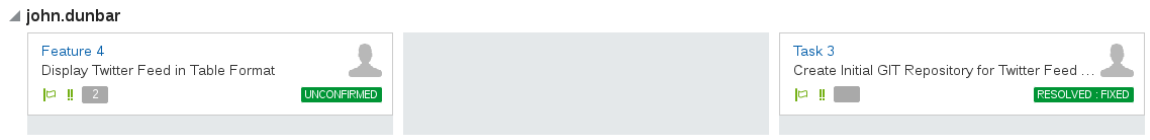
☐ Click on **Agile**, followed by clicking **Active Sprints**. Drag and drop **Task 3** from **In Progress** to the **Verify Code** column.



☐ In the Change Progress popup, click on **OK**

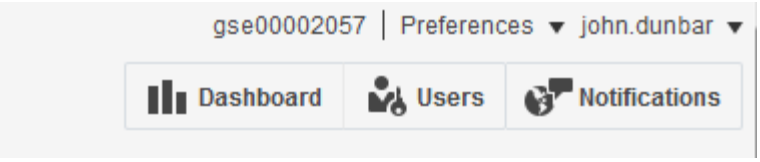


☐ The code is now ready for verification before moving to Completed

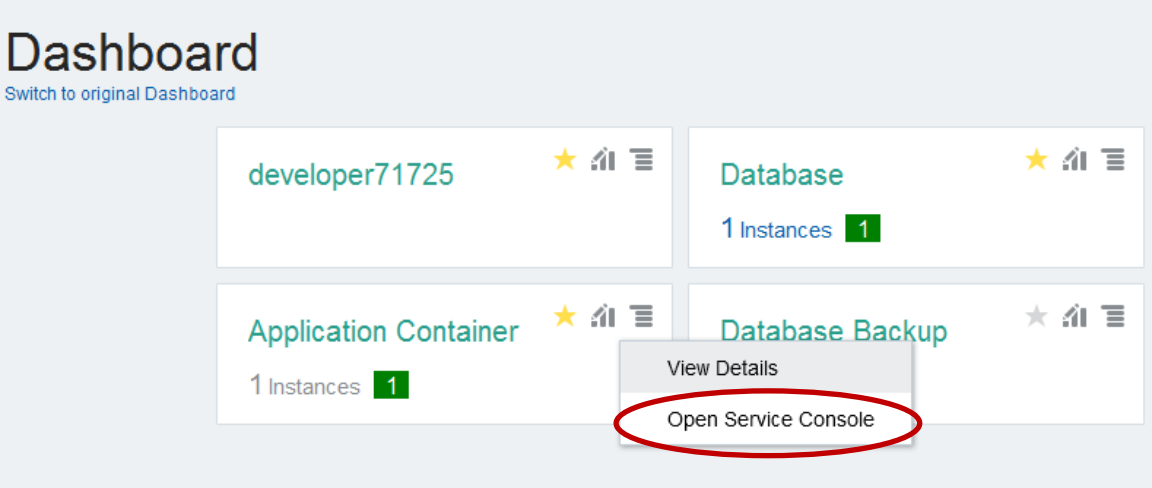


STEP 8: Login to Oracle Application Container Cloud Service

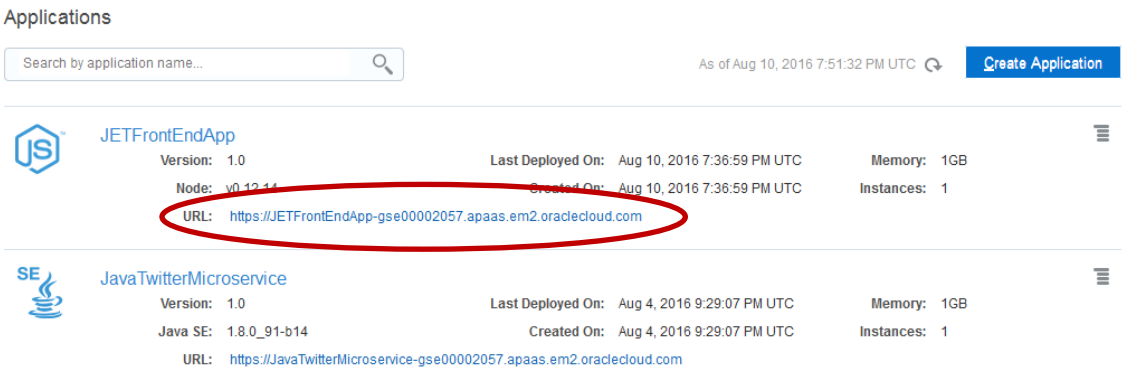
☐ Navigate back to the Oracle Public Cloud tab. Click **Dashboard** to return back to main Cloud Service Dashboard.



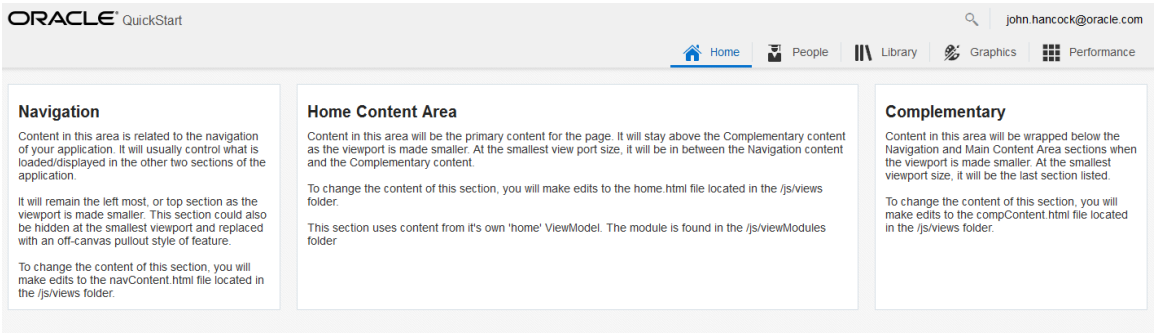
☐ On the Application Container Cloud Service (ACCS) click  and select **Open Service Console**



☐ On the ACCS Service Console you can view all the deployed applications including our newly create **JETFrontEndApp**.



☐ Click on URL to bring up the application.

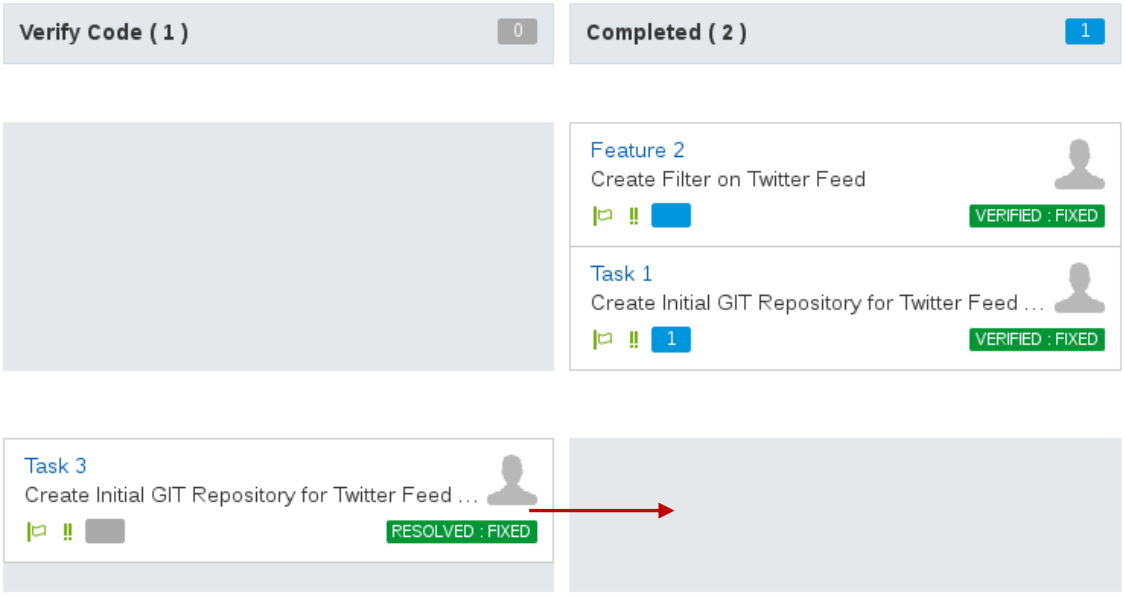


STEP 9: Complete Task

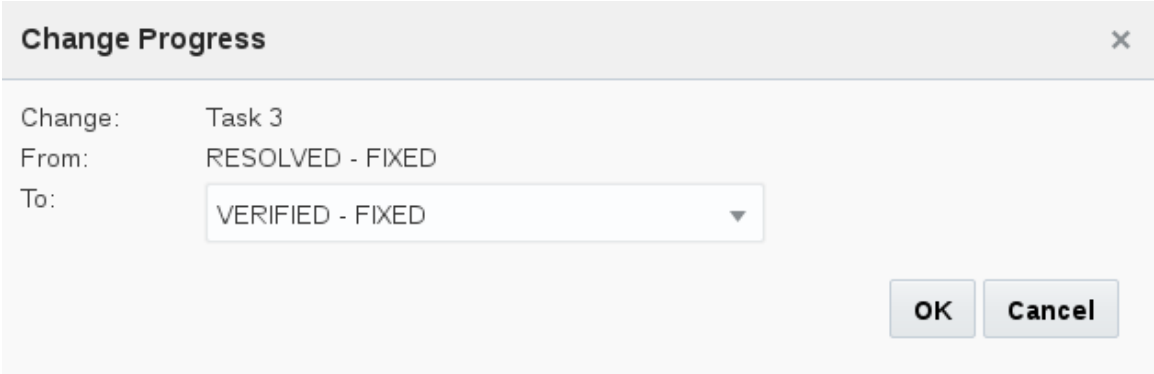
We have now verified that our application has been deployed and is functional. To finish up this part of the lab we will want to mark the Issue as completed in our Sprint.

☐ Back in the Developer Cloud Service, click **Agile,** followed by clicking **Active Sprints**.

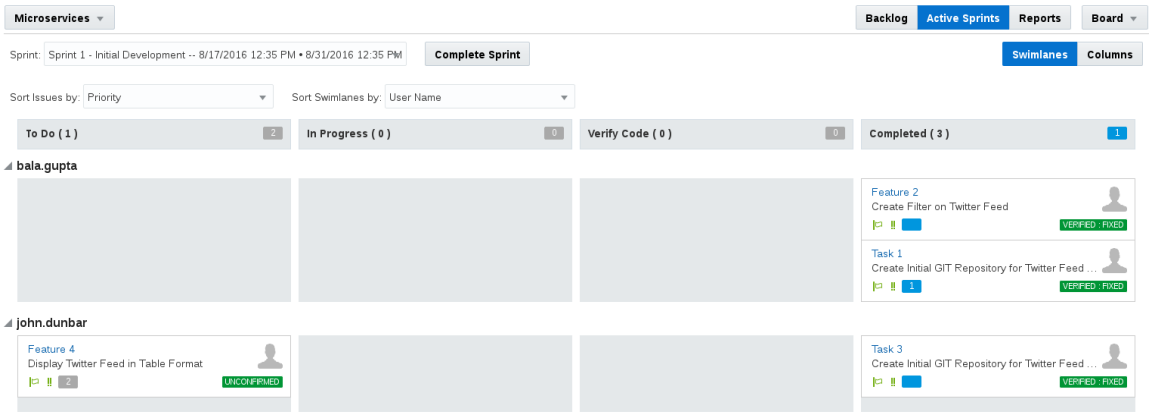
☐ Drag and drop **Task 3** from **Verify Code** to **Completed**.



☐ In the Change Progress popup click **OK**



☐ Your Sprint should now look like the following:



Extend default application to Display Twitter Feed

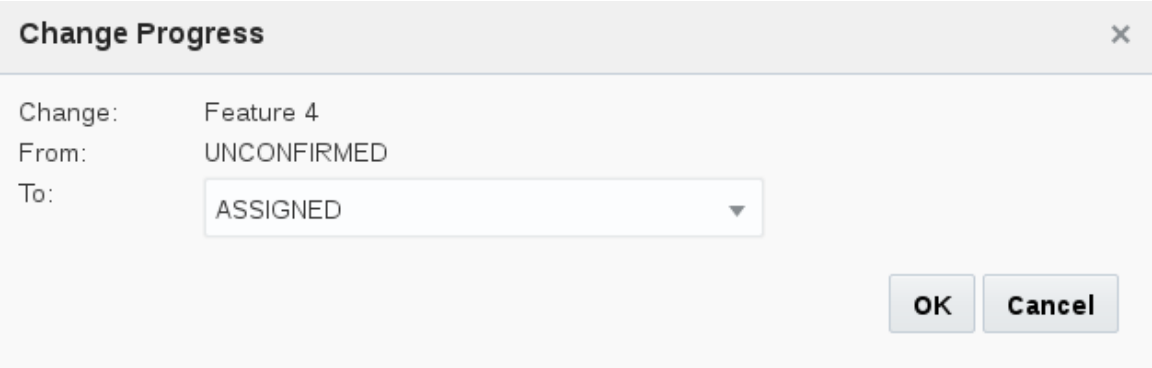
Now that we have our default application we want to extend this application to add the display of the twitter feed. For this task we will use Brackets text editor to pull down the code from Developer Cloud Service and add in our modifications. Once the new code is ready for deployment we will check the code in on a branch so that it can go through a code review prior to build and deployment.

STEP 10: Move Task to In Progress

To start this part of the lab we will want to mark the Issue as In Progress in our Sprint.

- ☐ Back in the Developer Cloud Service, click **Agile,** followed by clicking **Active Sprints**.
- ☐ Drag and drop **Feature 4** from **To Do** to **In Progress**.

☐ In the Change Progress popup click **OK**



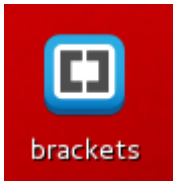
Clone Project to Brackets Text Editor

STEP 11: Start Brackets Text Editor

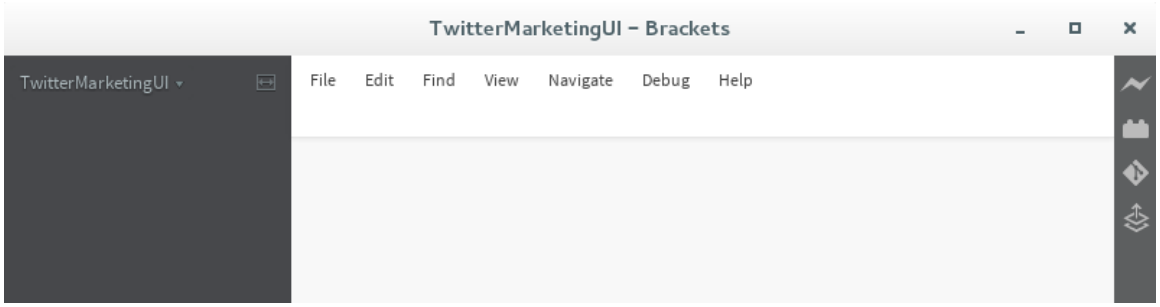
☐ Start **Brackets** text editor. How you start Brackets will depend on your OS. We have documented how to start Brackets from our OEL image.

Note: If you do not have Brackets installed please follow the appendix.

☐ Right click **Brackets** desktop icon and select **Open**

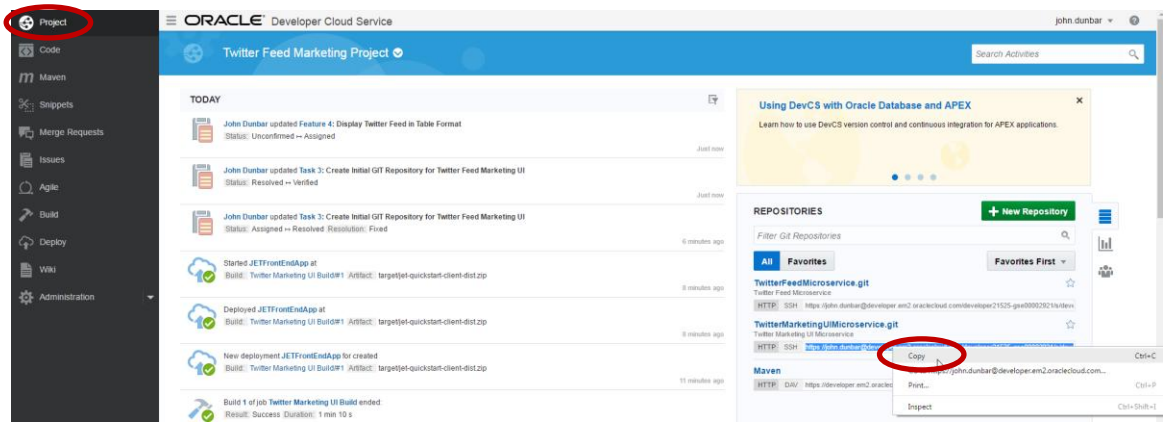


☐ Brackets should open with the **TwitterMarketingUI** folder already loaded.



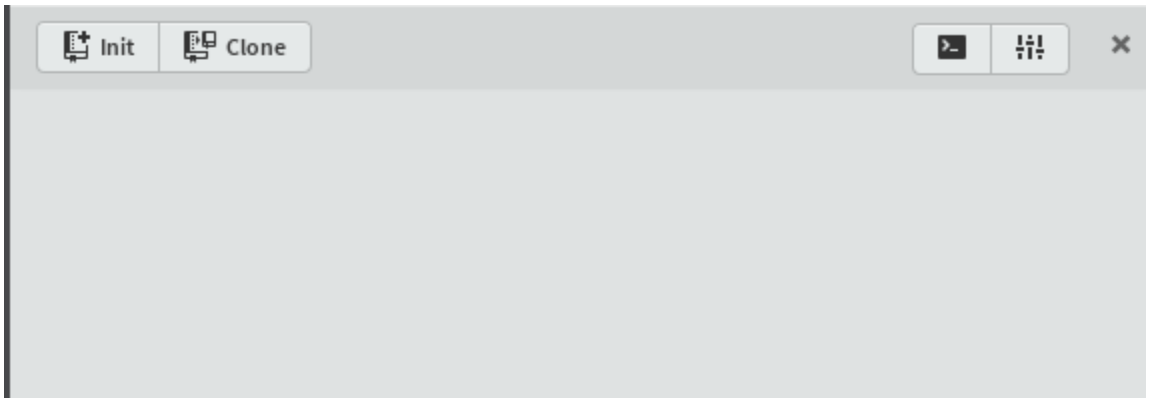
STEP 12: Copy GIT URL

☐ Back in Developer Cloud Service, click on **Project**. On right side, select the URL for **TwitterMarketingUIMicroservice.git**. Right click and select **Copy**

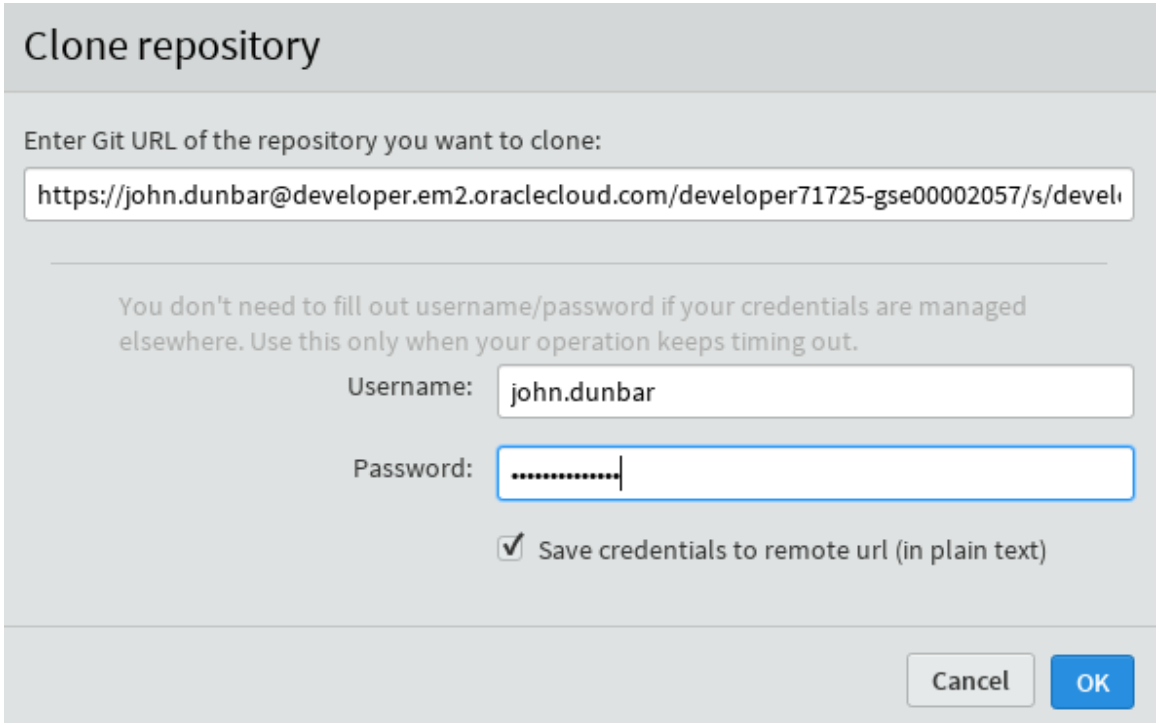


STEP 13: Clone GIT Repository

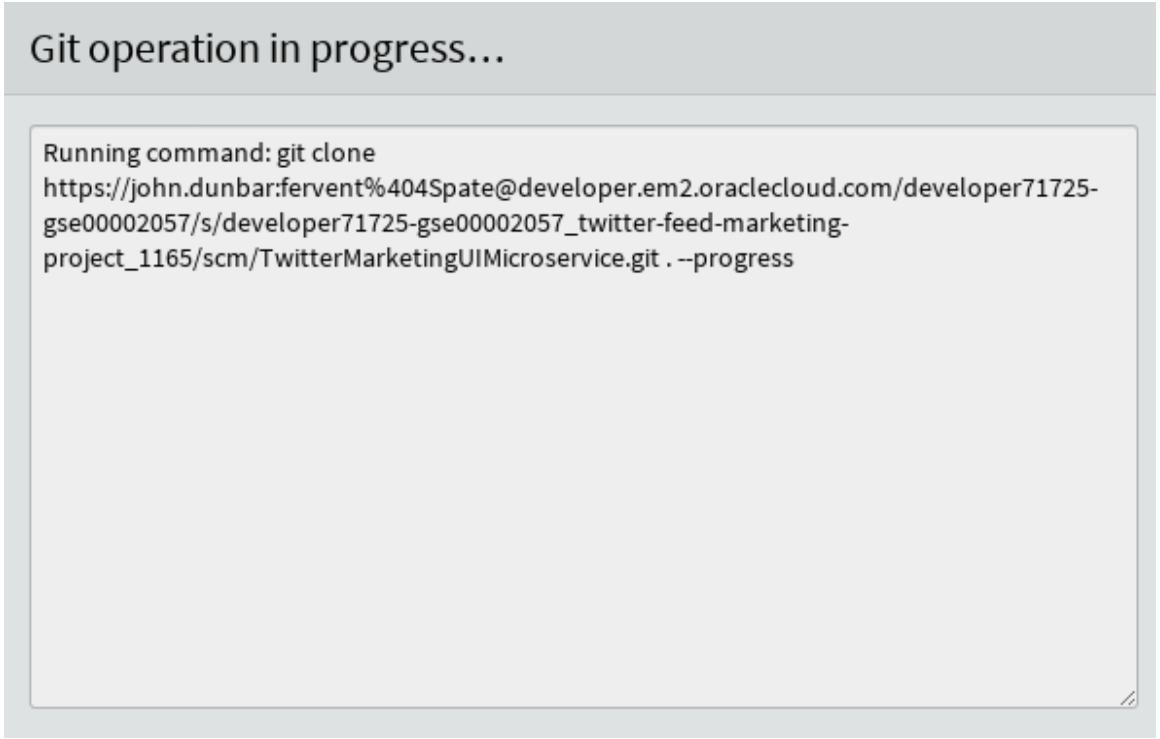
☐ Back in the Brackets editor, on the right hand side, click  **GIT** icon.



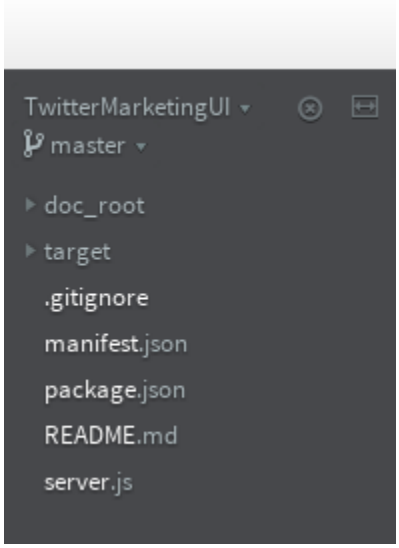
☐ Click **Clone** and paste in Git URL that you captured from Developer Cloud Service. Username should be populated automatically. Enter your **Password** and click **Save credentials**. Once completed click **OK** to start the cloning process.



☐ While the clone is running a dialog box will show you the progress.



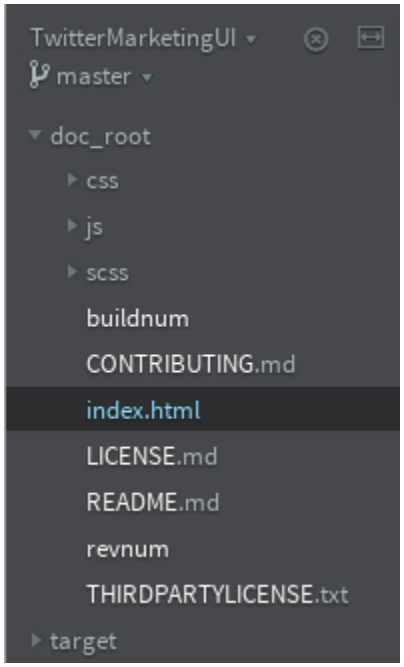
☐ You now have a local copy of the repository.



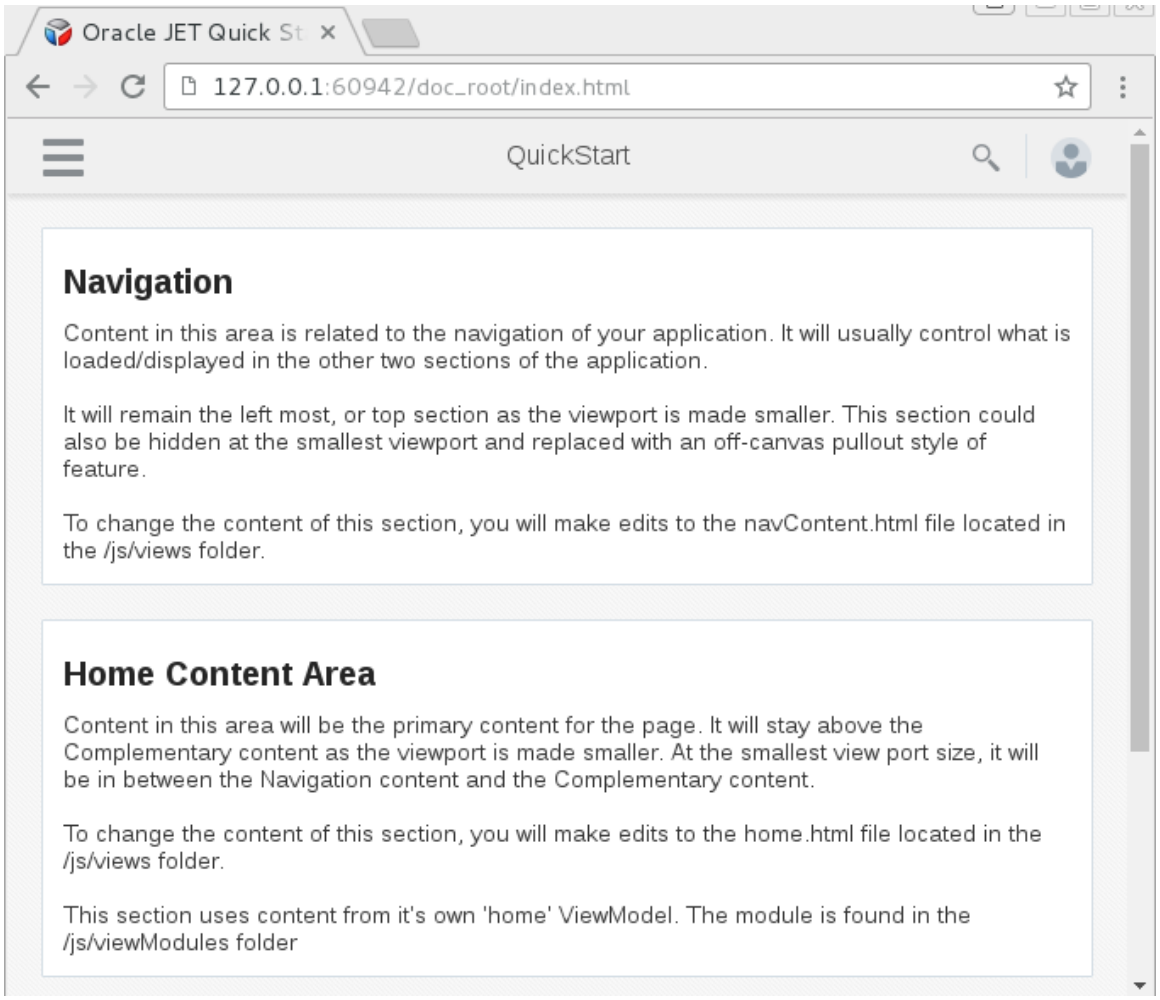
STEP 14: Run Live Preview.

☐ Before we make our code changes lets first run the code locally.

☐ Expand **doc_root** and select **index.html**



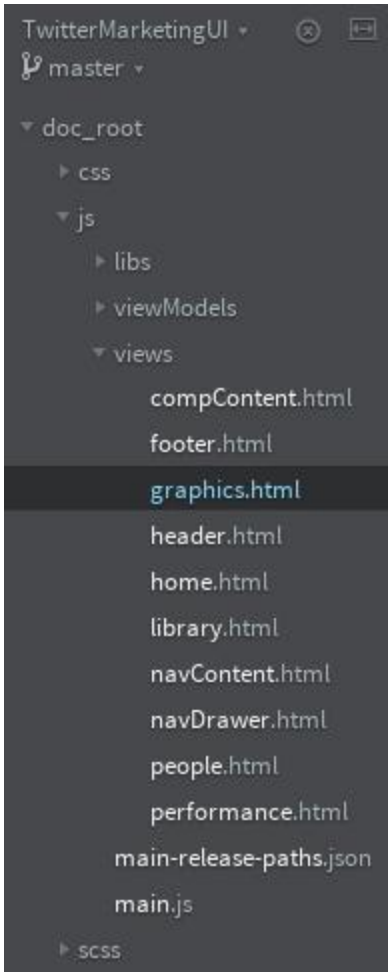
☐ On right hand panel, click  Live Preview. This will start your JavaScript application in a browser. Once you verify the application is working you can close the browser.



Add Code to display Twitter Feed in Table Format

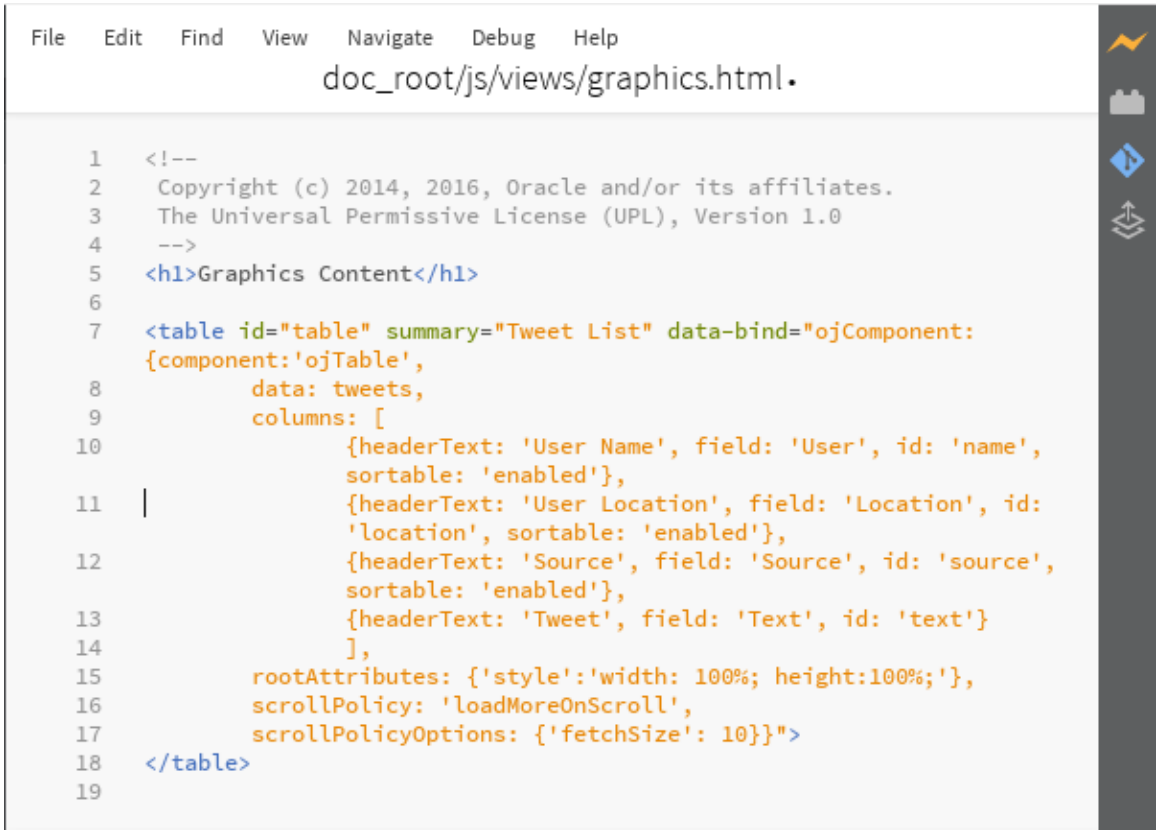
STEP 15: Modify graphics.html

☐ Expand **doc_root -> js -> views** and click **graphics.html**.



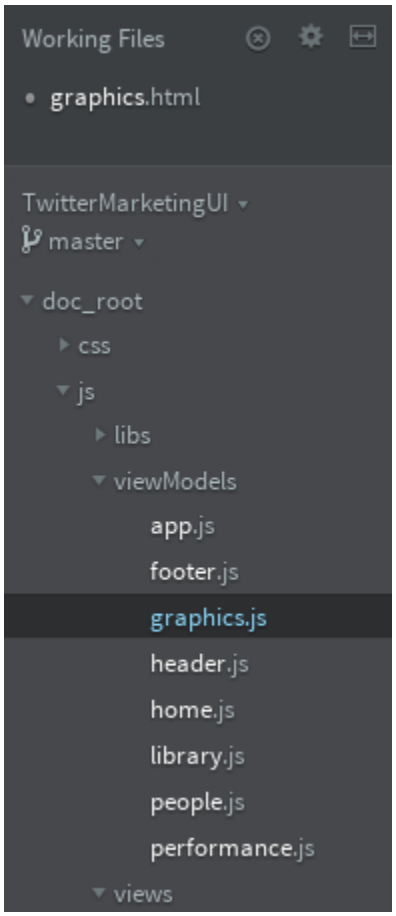
☐ Replace the existing code with the code block below:

```
<h1>Graphics Content</h1>

<table id="table" summary="Tweet List" data-
bind="ojComponent:{component:'ojTable',
        data: tweets,
        columns: [
               {headerText: 'User Name', field: 'User', id:
'name', sortable: 'enabled'},
               {headerText: 'User Location', field: 'Location',
id: 'location', sortable: 'enabled'},
               {headerText: 'Source', field: 'Source', id:
'source', sortable: 'enabled'},
               {headerText: 'Tweet', field: 'Text', id: 'text'}
               ],
        rootAttributes: {'style':'width: 100%; height:100%;'},
        scrollPolicy: 'loadMoreOnScroll',
        scrollPolicyOptions: {'fetchSize': 10}}">
</table>
```



STEP 16: Modify graphics.js

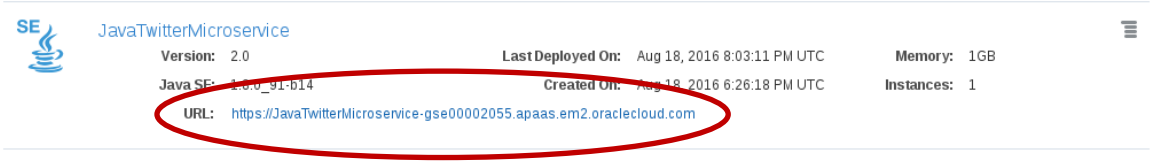
☐ Expand **doc_root -> js -> viewModels** and click **graphics.js**.



☐ Add the code block below to the bottom on the **graphics.js** file:

```
/*global $, define, console*/
/*jslint sloppy:true*/

define(['ojs/ojcore', 'knockout', 'ojs/ojtable'], function (oj,
ko) {

    function mainContentViewModel() {

        // change this root variable to point to YOUR environment
        var root = 'https://javatwittermicroservice-
metcsgse00210.apaas.em2.oraclecloud.com/',
            self = this,
            uri = 'statictweets/',
            prettySource = function (source) {
                return source.substring(source.indexOf('>') + 1,
source.lastIndexOf('<'));
            },
            url = root + uri;

        self.items = ko.observableArray([]);
        self.tweets = new oj.ArrayTableDataSource(self.items, {
            idAttribute: 'Id'
        });

        $.ajax({
            url: url,
            method: 'GET'
        }).success(function (result) {
            console.log(result.tweets);
            var items = self.items();
            ko.utils.arrayForEach(result.tweets, function (value)
{
                // make sure this is a creation tweet
                if (!!value.user) {
                    items.push({
                        Id: value.id,
                        Location: value.user.location,
                        Text: value.text,
                        Source: prettySource(value.source),
                        User: value.user.name
                    });
                }
            });
            self.items.valueHasMutated();
        });
    }
    return mainContentViewModel;
});
```

☐ Back in the browser; navigate back to the Application Container Cloud Service service console. Copy URL for **JavaTwitterMicroservice** that was created in Lab 200.



☐ Replace existing URL with your URL for the **root** variable. You must **append** a '**/**' to the **end of the URL**.

```
function mainContentViewModel() {

    // change this root variable to point to YOUR environment
    var root = 'https://javatwittermicroservice-
    gse00002055.apaas.em2.oraclecloud.com/',
```

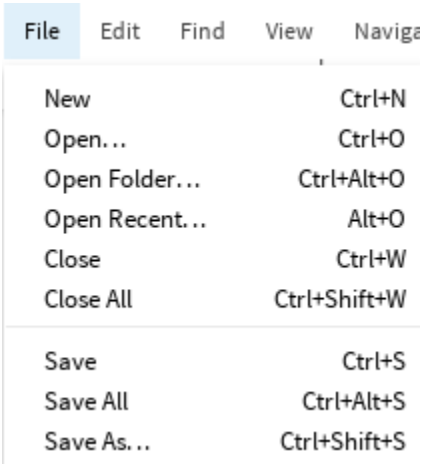
☐ Completed graphics should look something like the image below:

doc_root/js/viewModels/graphics.js

```
/**
 * Copyright (c) 2014, 2016, Oracle and/or its affiliates.
 * The Universal Permissive License (UPL), Version 1.0
 */
/*
 * Your viewModel code goes here
 */

/*global $, define, console*/
/*jslint sloppy:true*/

define(['ojs/ojcore', 'knockout', 'ojs/ojtable'], function (oj, ko)
{

    function mainContentViewModel() {

        // change this root variable to point to YOUR environment
        var root = 'https://javatwittermicroservice-
        gse00002055.apaas.em2.oraclecloud.com/',
            self = this,
            uri = 'statictweets/',
            prettySource = function (source) {
                return source.substring(source.indexOf('>') + 1,
                source.lastIndexOf('<'));
            },
            url = root + uri;

        self.items = ko.observableArray([]);
        self.tweets = new oj.ArrayTableDataSource(self.items, {
            idAttribute: 'Id'
        });

        $.ajax({
```

Line 17, Column 91 — 53 Lines | INS | JavaScript | Spaces: 4

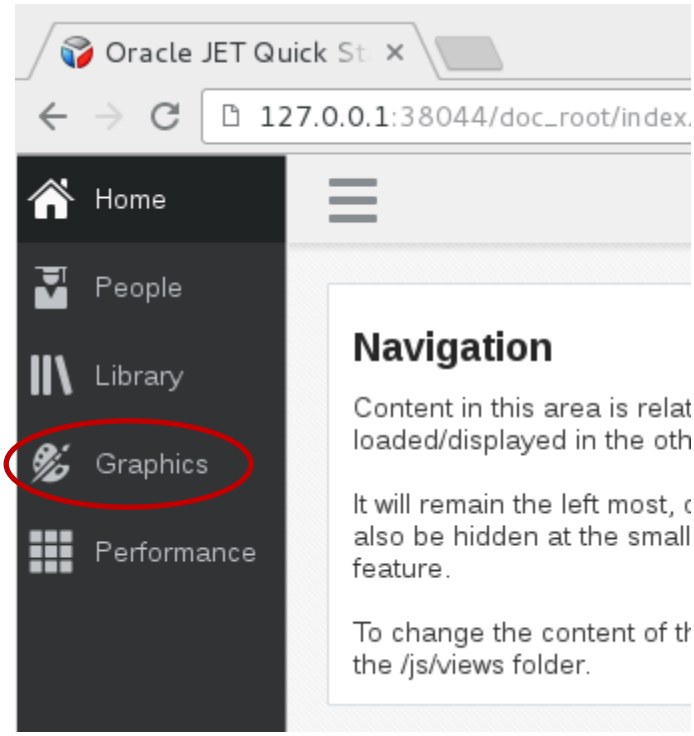
☐ Save all files by clicking **File -> Save All**



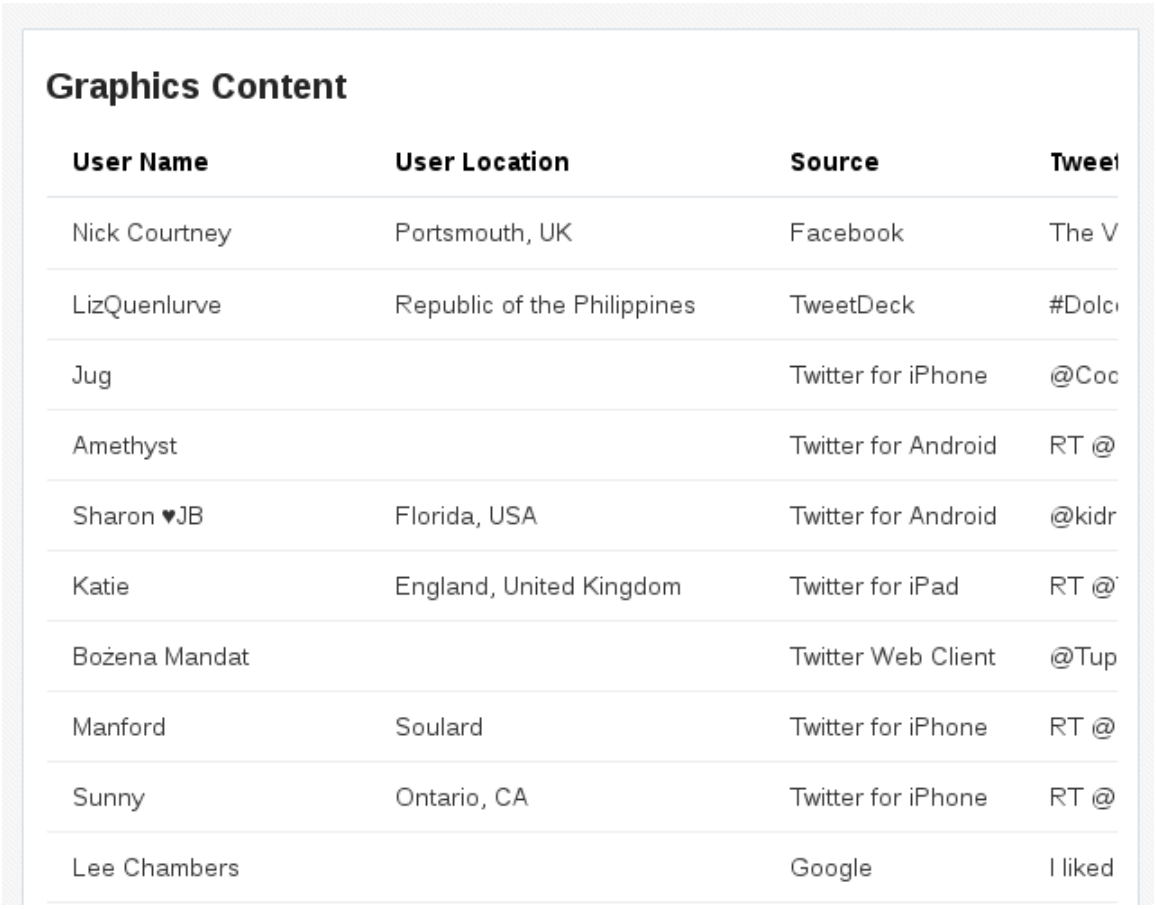
STEP 17: Test new changes

☐ Click  Live Preview to test out the new changes.

☐ Click  and select **Graphics**



☐ In the graphics sections you can now see all the twitter feed data:



Graphics Content

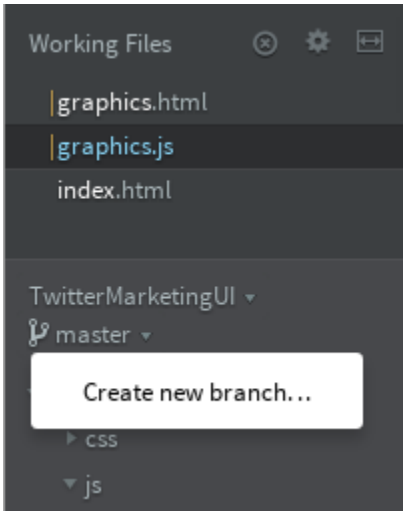
User Name	User Location	Source	Tweet
Nick Courtney	Portsmouth, UK	Facebook	The V
LizQuenlurve	Republic of the Philippines	TweetDeck	#Dolc
Jug		Twitter for iPhone	@Coc
Amethyst		Twitter for Android	RT @
Sharon ♥JB	Florida, USA	Twitter for Android	@kidr
Katie	England, United Kingdom	Twitter for iPad	RT @
Bożena Mandat		Twitter Web Client	@Tup
Manford	Soulard	Twitter for iPhone	RT @
Sunny	Ontario, CA	Twitter for iPhone	RT @
Lee Chambers		Google	I liked

Create a new Branch and Commit Code

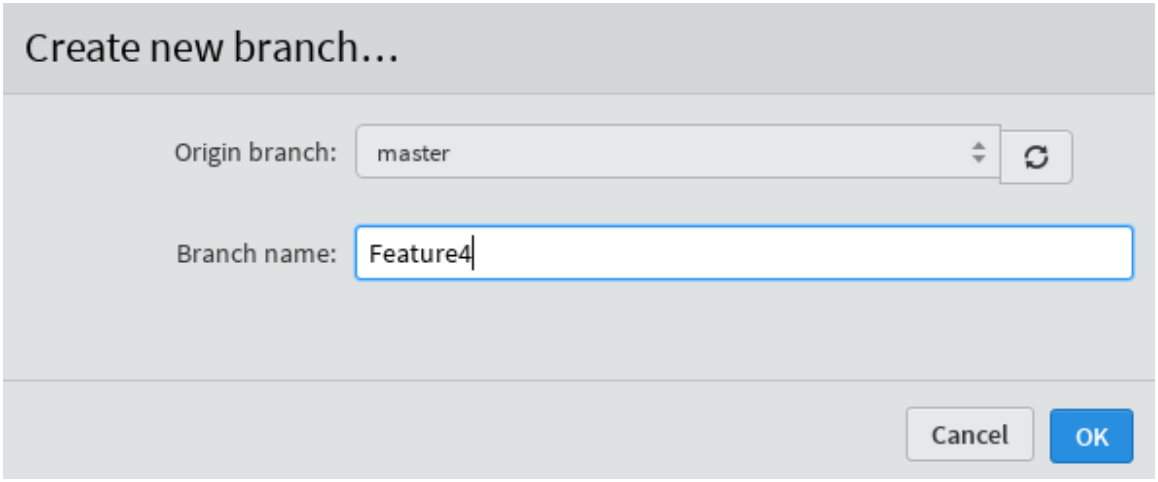
Create a Branch and Commit Code

STEP 18: Create a new Branch and Commit Code

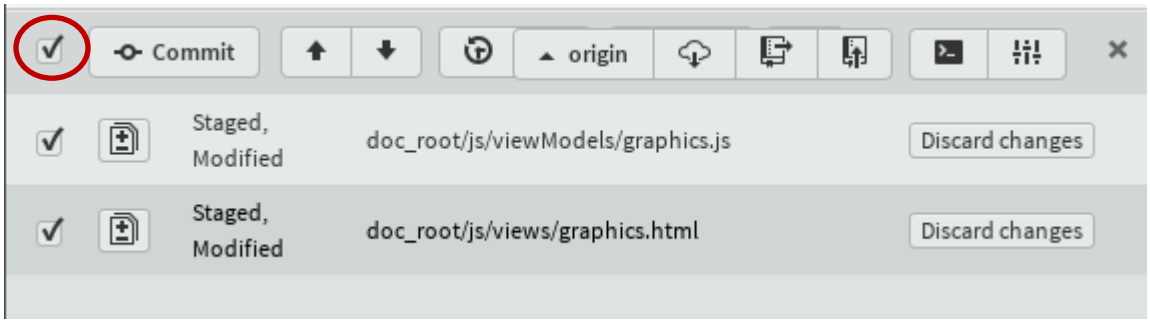
☐ First we need to create a new branch to check in all of our changes for this feature. In the left hand navigation panel, select **master** and click **Create new branch**.



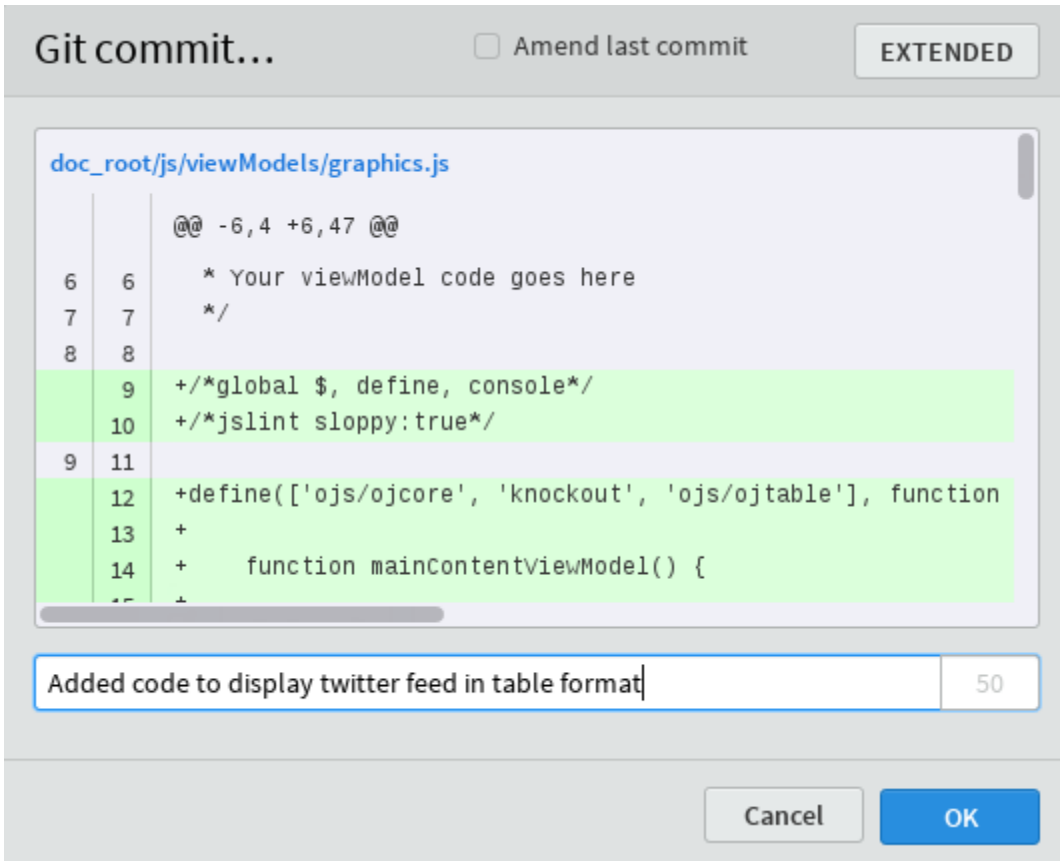
☐ In popup window, enter **Feature4** for branch name and click **OK**.



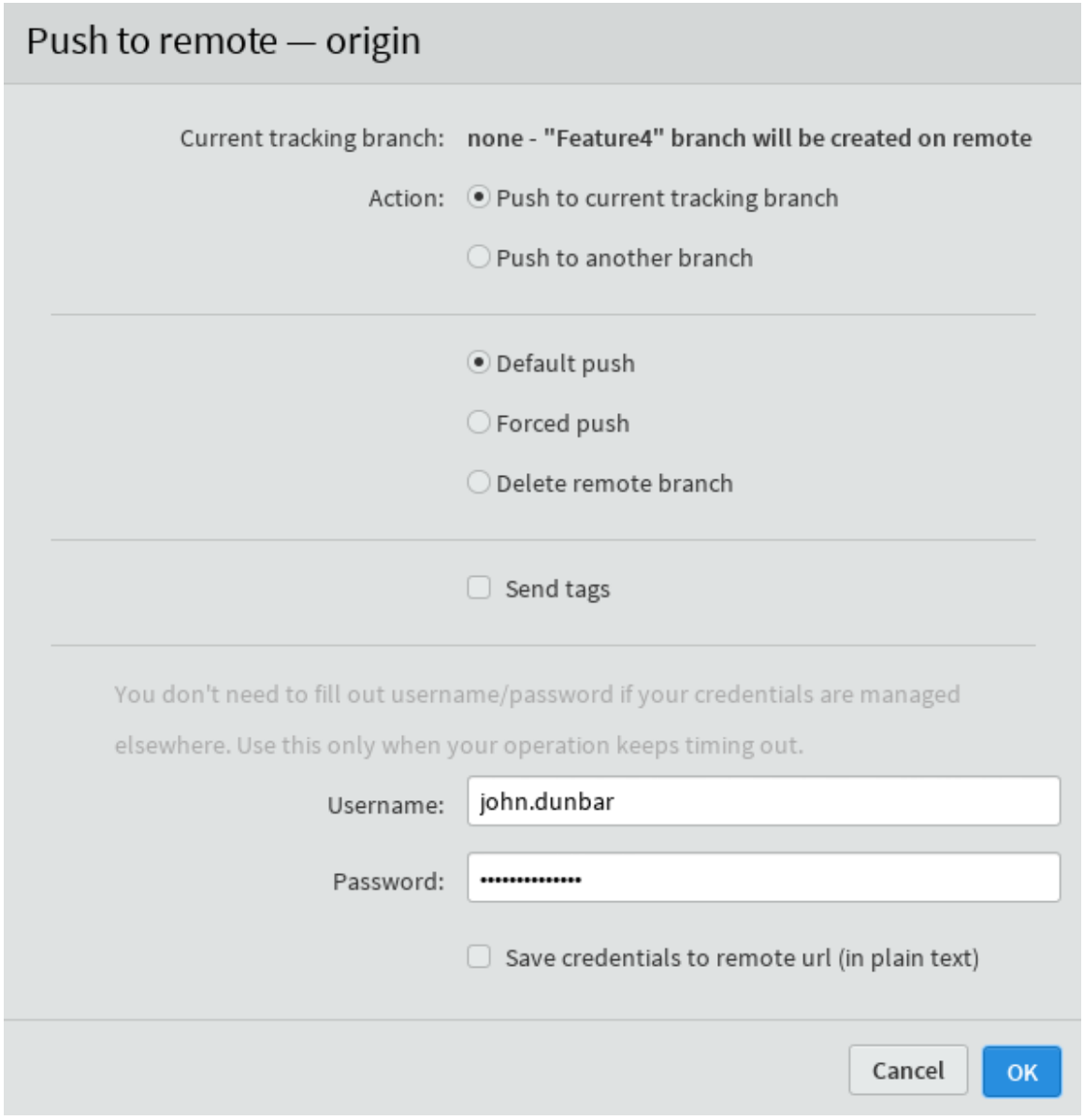
☐ Click  Git icon. Check the box next to **Commit** to select all modified files.



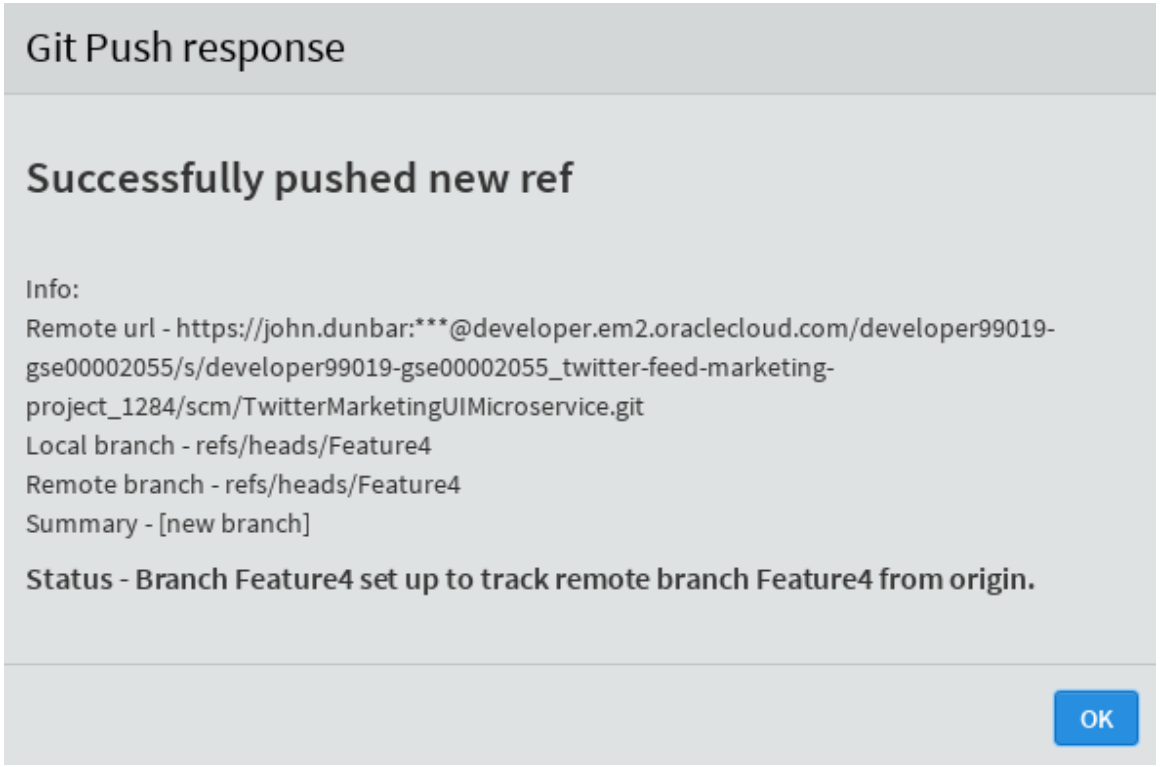
☐ Click **Commit**. In popup enter comment **Added code to display twitter feed in table format** and click **OK**. This will commit the changes to your local Git repository.



- ☐ Click  Git Push icon.
- ☐ In popup window leave all defaults and click **OK**



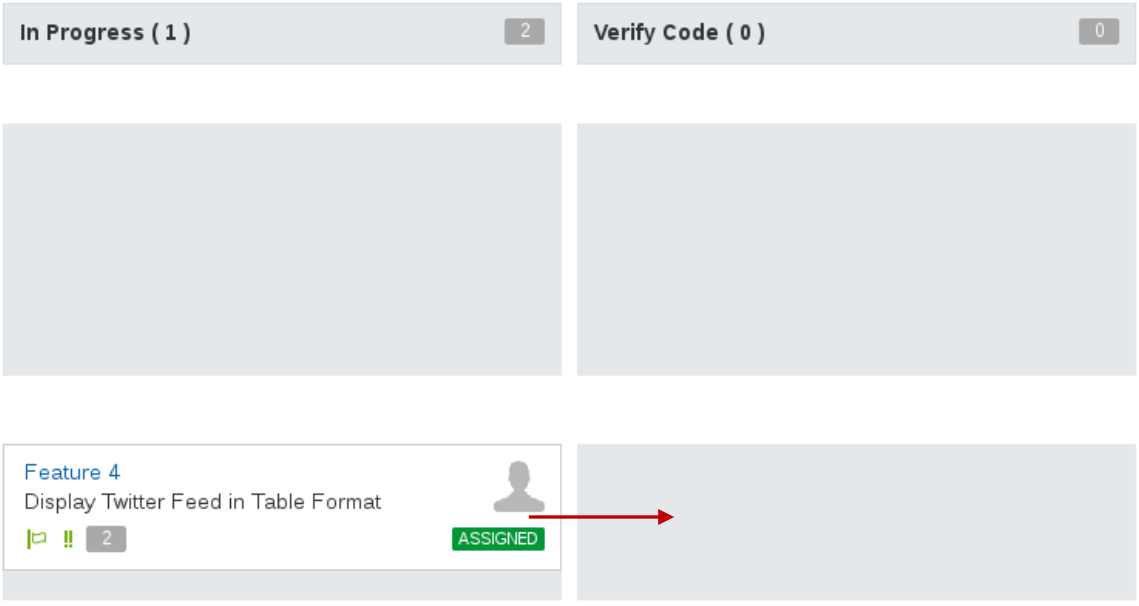
☐ Once Git Push completes click **OK**.



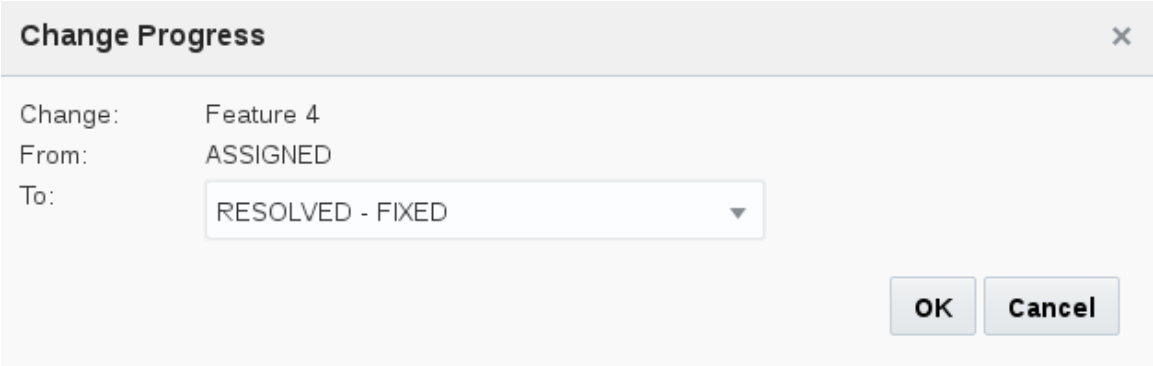
STEP 19: Complete the Display Twitter Feed Task

☐ Back in the Developer Cloud Service window, click **Agile,** followed by clicking **Active Sprints**.

☐ Drag and drop **Feature 4** from **In Progress** to **Verify Code**.



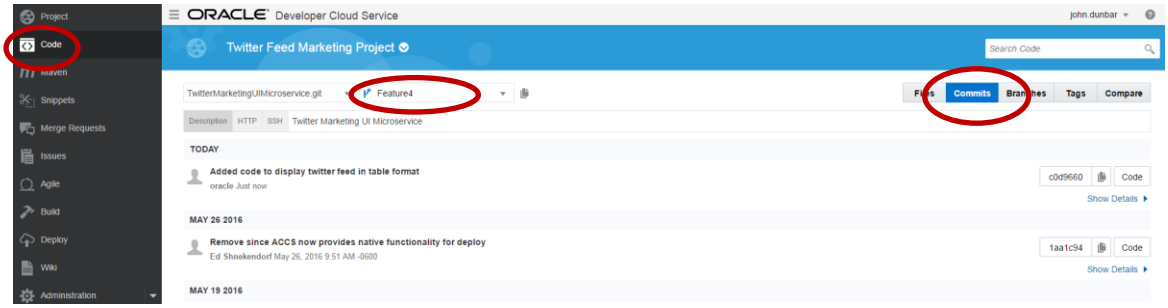
☐ In the Change Progress popup click **OK**.



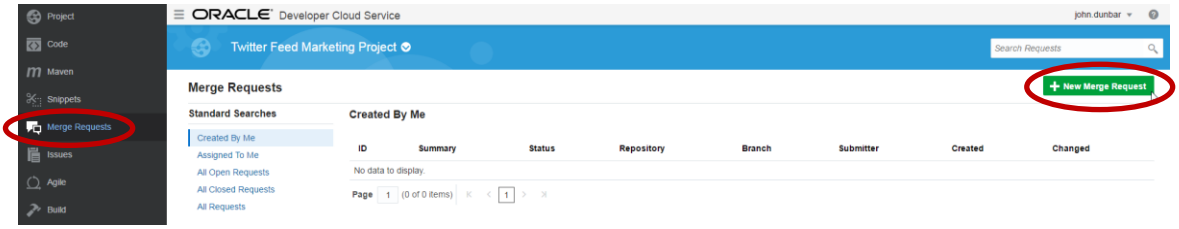
Create Merge Request

STEP 20: Review Sprint Status and create Merge Request

☐ Click on the **Code tab**, select the **Feature4** branch and then click on the **Commits** sub tab. Now view the recent commit that we made to branch from Brackets.

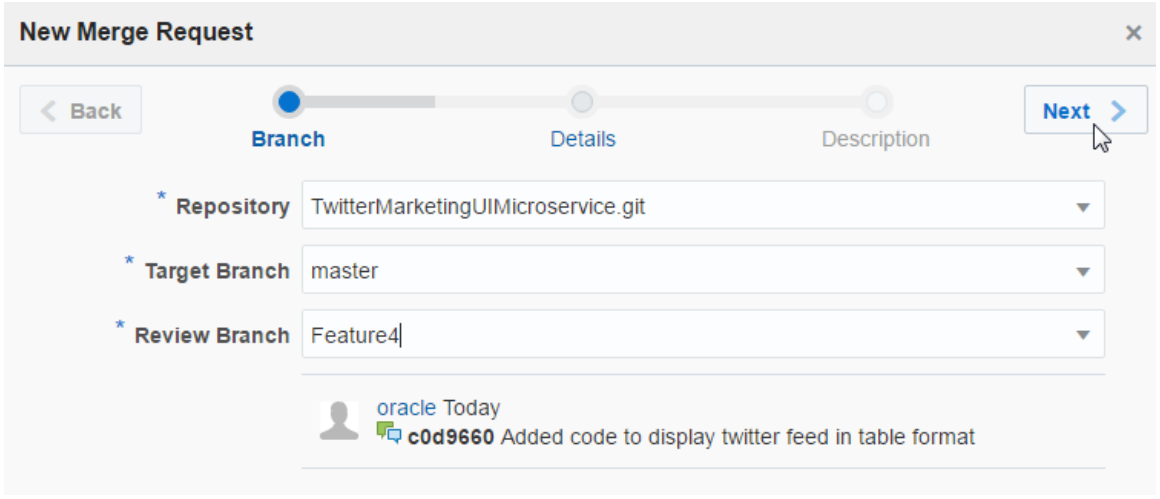


☐ Now that John Dunbar has completed the task of displaying twitter feed in table format, a Merge Request can be created by John and assigned to Lisa Jones. Click on **Merge Requests**, and then click on the **New Request button**.



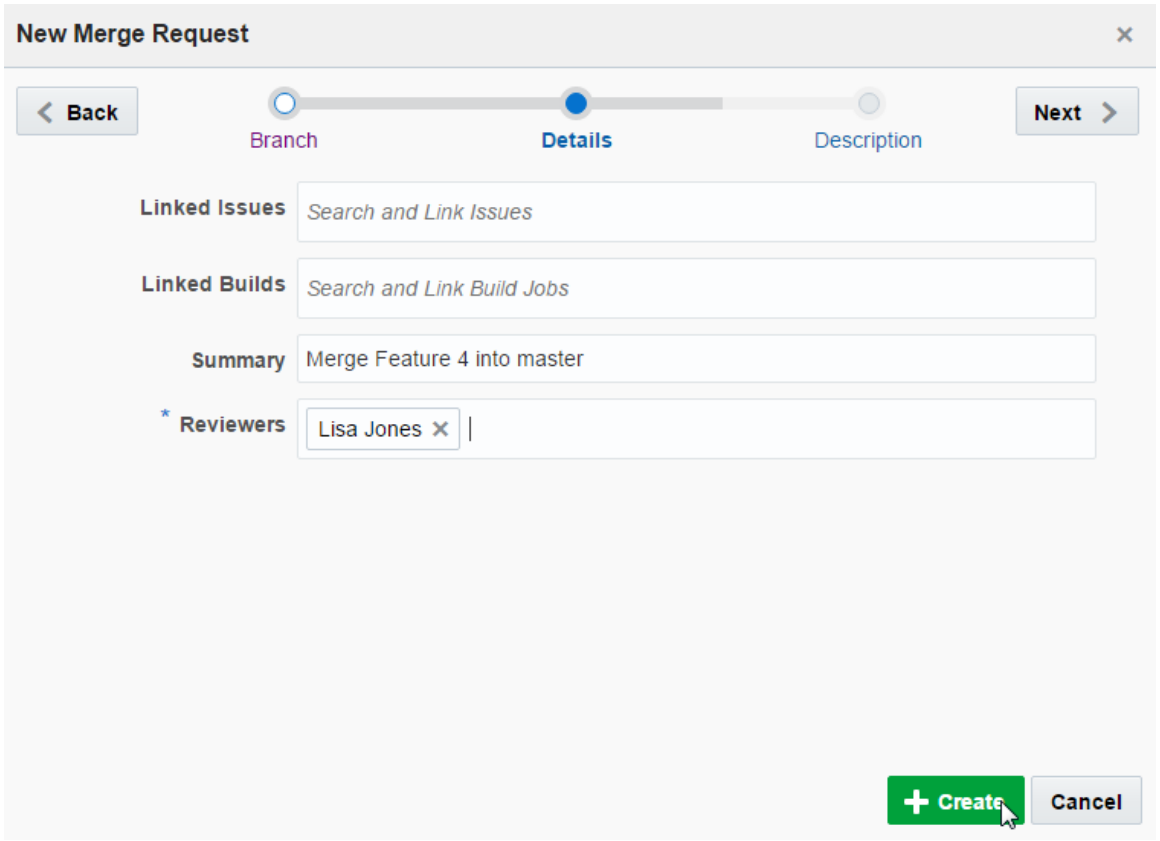
☐ Enter the following information into the New Merge Request and click **Next**

```
Repository:    TwitterMarketingUIMicroservice.git
Target Branch: master
Review Branch: Feature4
```

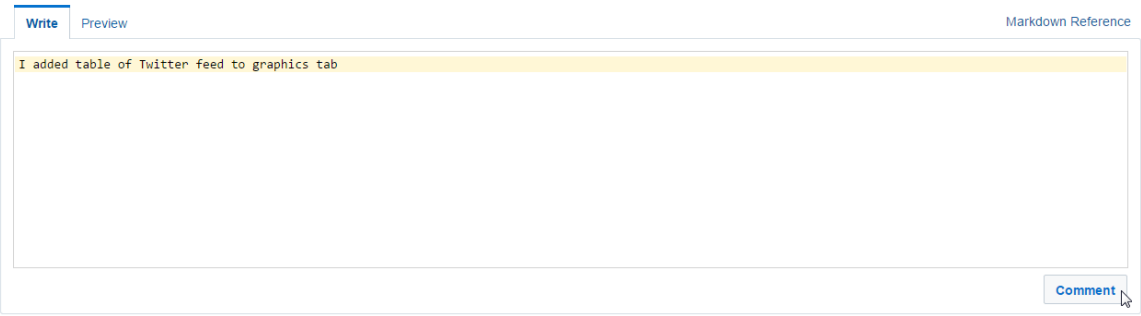


☐ Enter the following information into **Details** and click **Create**

```
Summary:       Merge Feature 4 into master
Reviewers:     Lisa Jones (or current user in non-multi user env)
```



☐ In the **Write** box, enter the following comment and then click on the **Comment** button to save: "`I added table of Twitter feed to graphics tab`"



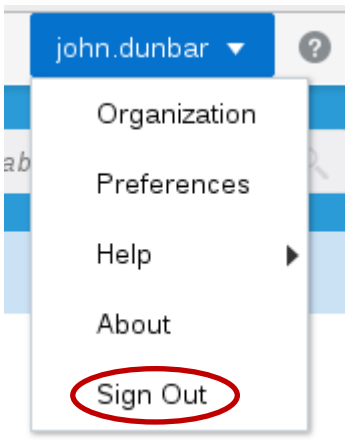
Merge the Branch as Lisa Jones

In the following steps "Lisa" will merge the branch created by "John" into the master.

NOTE: If you are using a single user environment, you will skip the next step, and go to the following step titled: "Merge Requests"

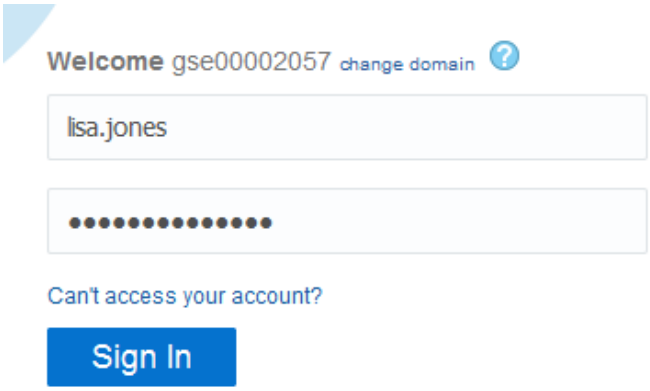
STEP 21: Sign Out as John Dunbar and Sign In as Lisa Jones

☐ Click on the **john.dunbar** dropdown located in the top right corner of the screen. Select **Sign Out**.

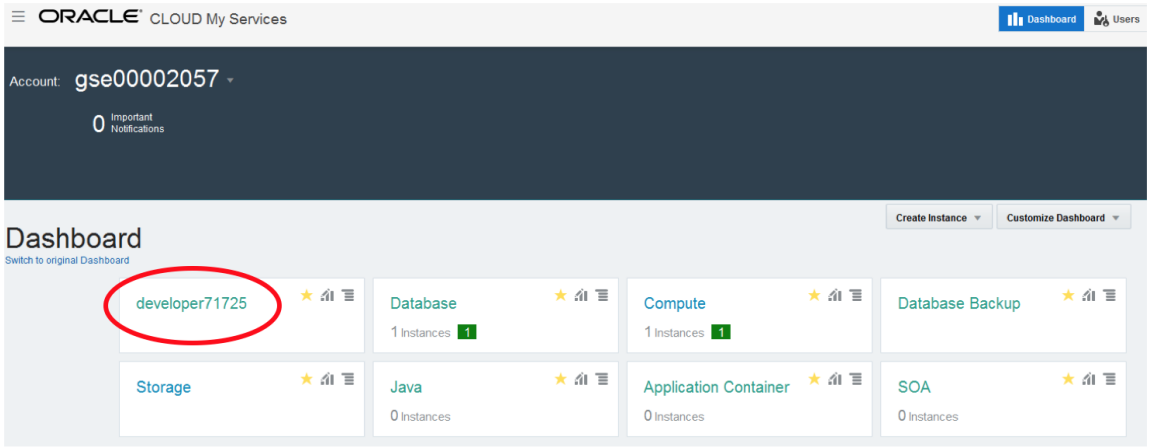


☐ Following the previously documented steps, go to the URL: **http://cloud.oracle.com**, click on **Sign In** found on the Top Right corner of the window. Select the correct **Data Center**, click on the **My Services** button, enter the correct **Identity Domain** and click on **Go**.

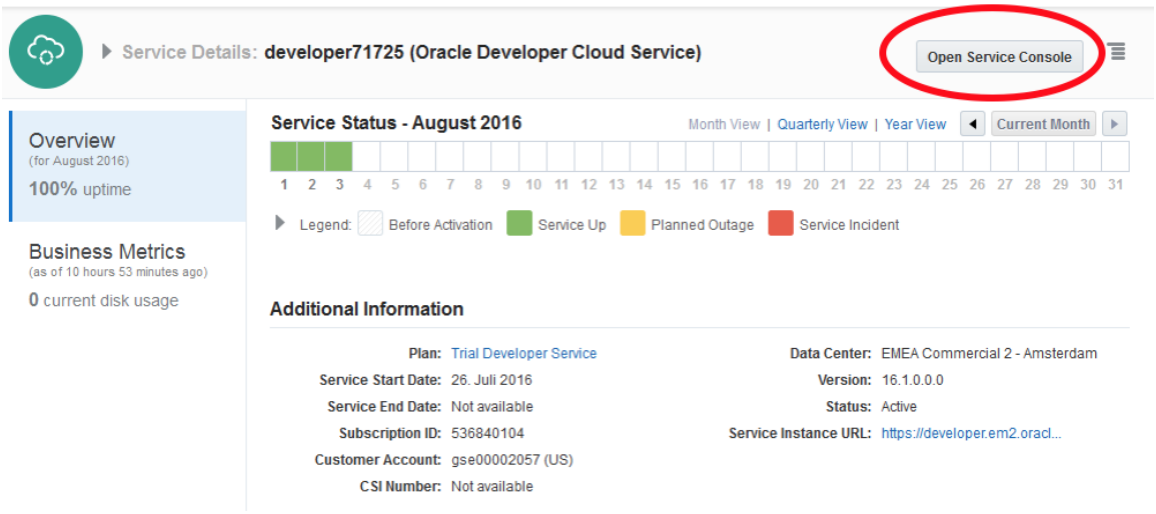
☐ Enter **lisa.jones** for the username, and enter the correct password. Click on **Sign In**.



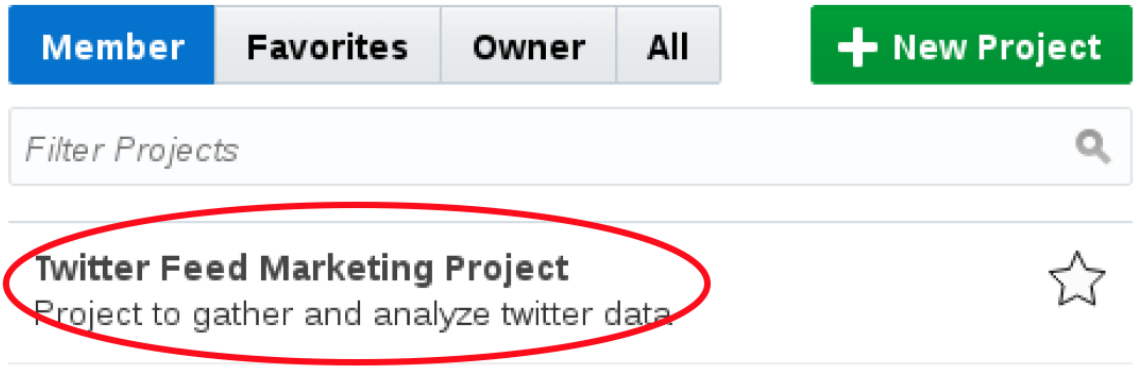
☐ When the Dashboard is displayed, click on the **Developer Cloud Service**.



☐ From the Developer Cloud Service Dashboard, click on the **Open Service Console** button



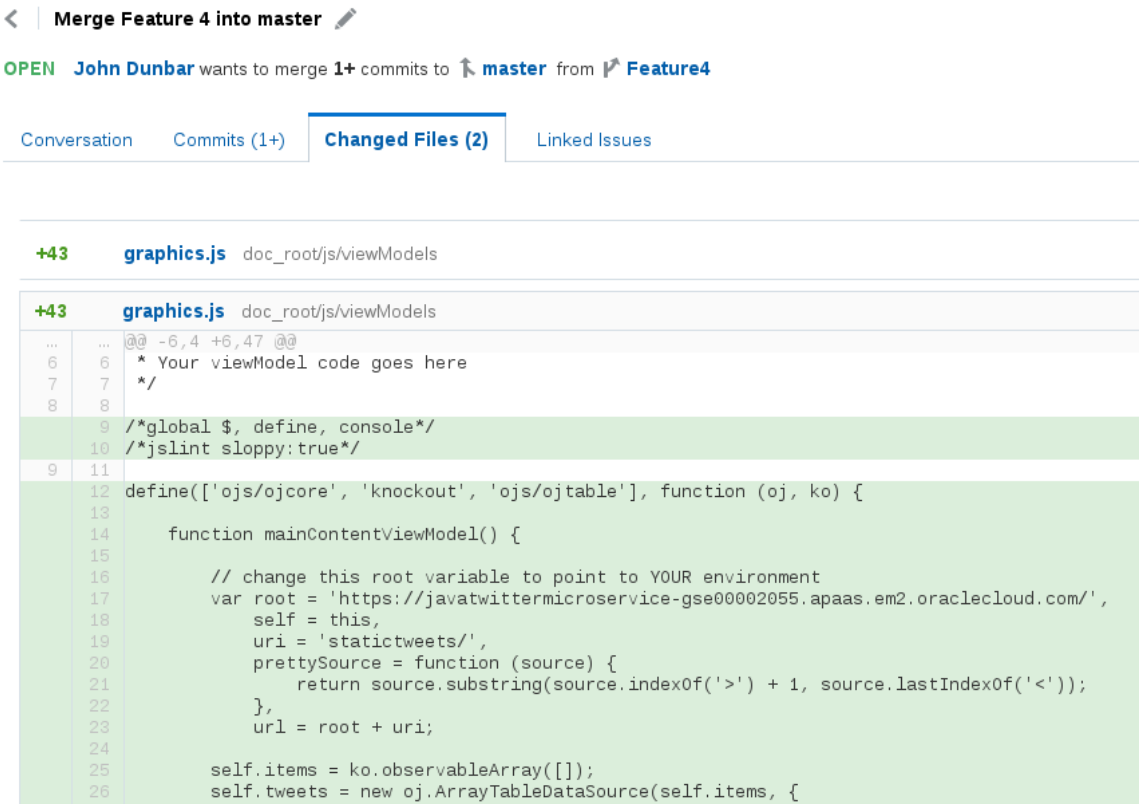
☐ Select the **Twitter Feed Marketing Project**



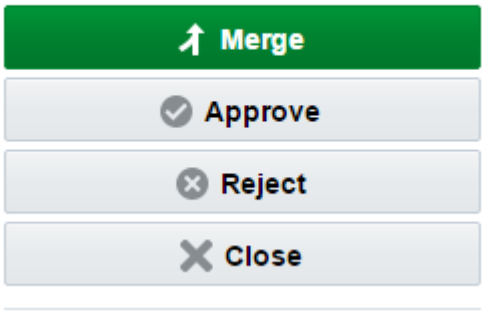
STEP 22: Merge Requests

☐ Click on **Merge Requests**. Select the **Assigned to Me** search. After the search completes, click on the **Merge Feature 4 into master** assigned request.

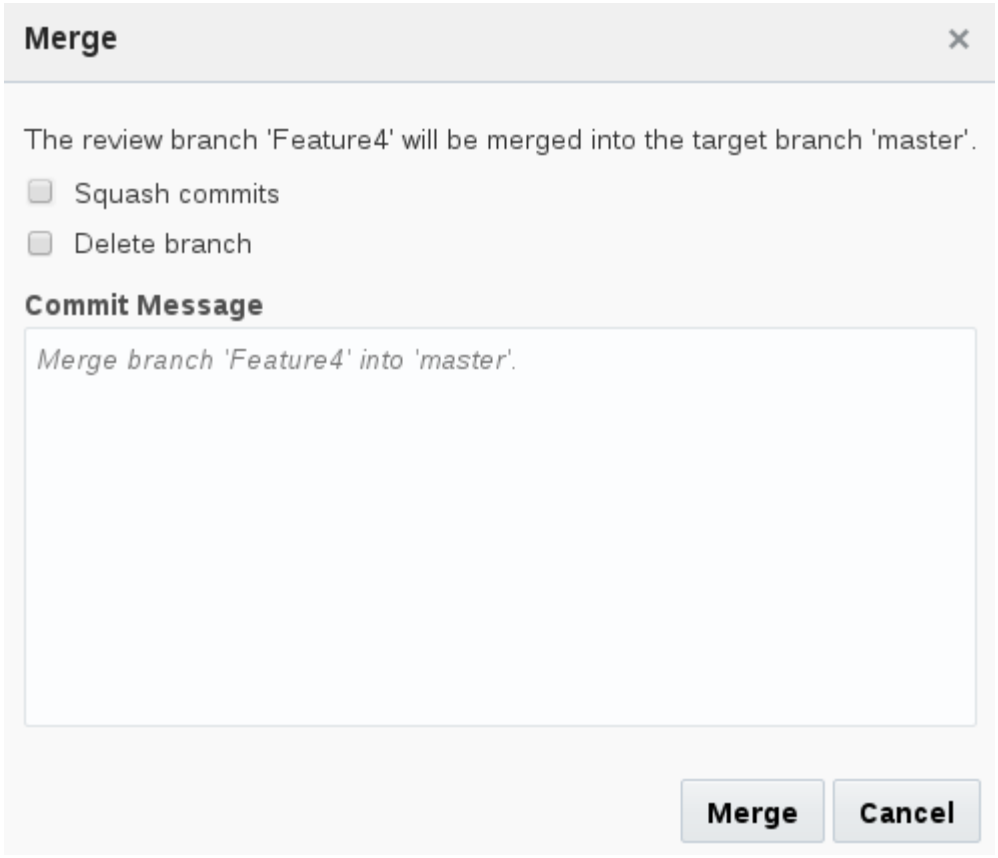
☐ Once the request has loaded, select the **Changed Files** tab. "Lisa" will now have the opportunity to review the changes in the branch, make comments, request more information, etc. before Approving, Rejecting or Merging the Branch.



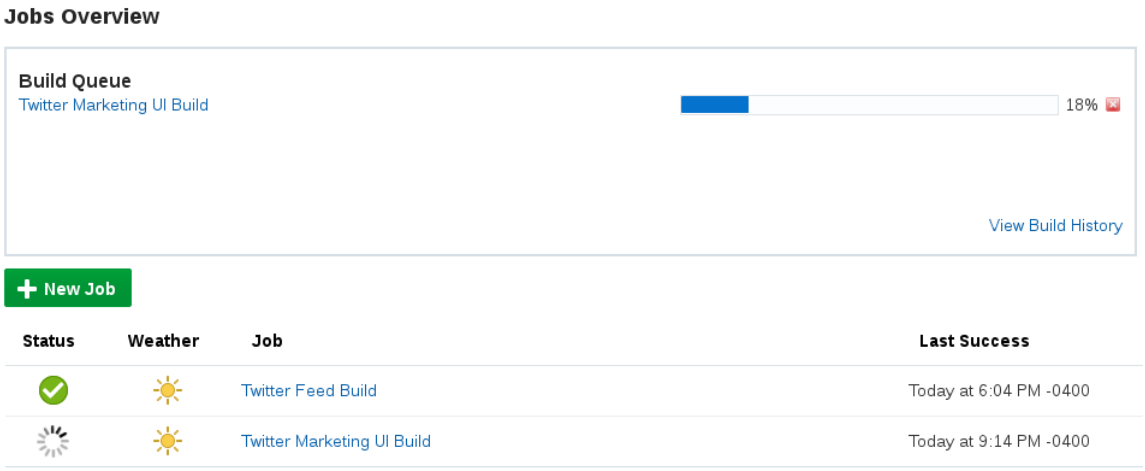
☐ Click on the **Merge** button.



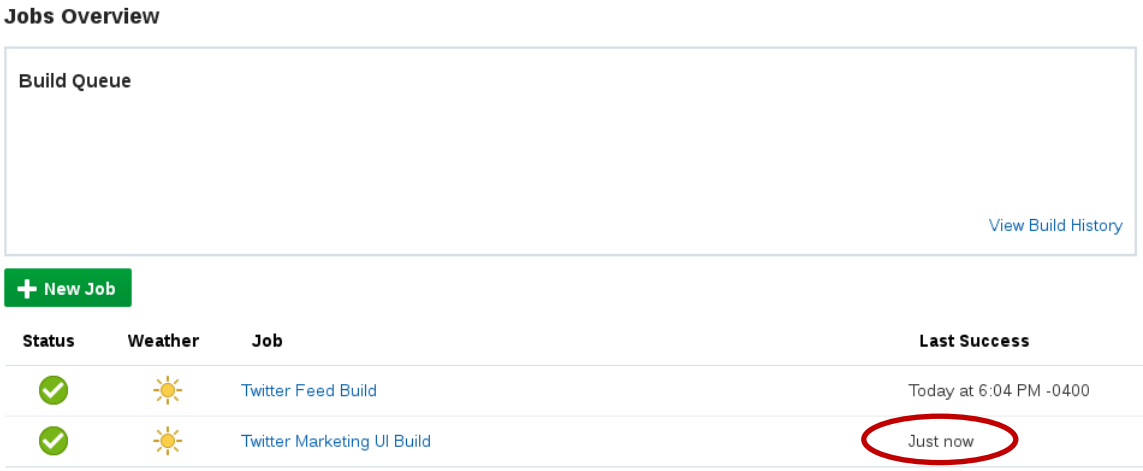
☐ Leave the defaults, and click on the **Merge** button in the confirmation dialog.



☐ Now that the code has been committed to the Developer Cloud Service repository, the build and deployment will automatically start. Click on **Build**, and you should see a **Twitter Marketing UI Build** in the Queue



☐ Wait a minute or two for the build to complete. The **Last Success** will be set to **Just Now** when the build completes.

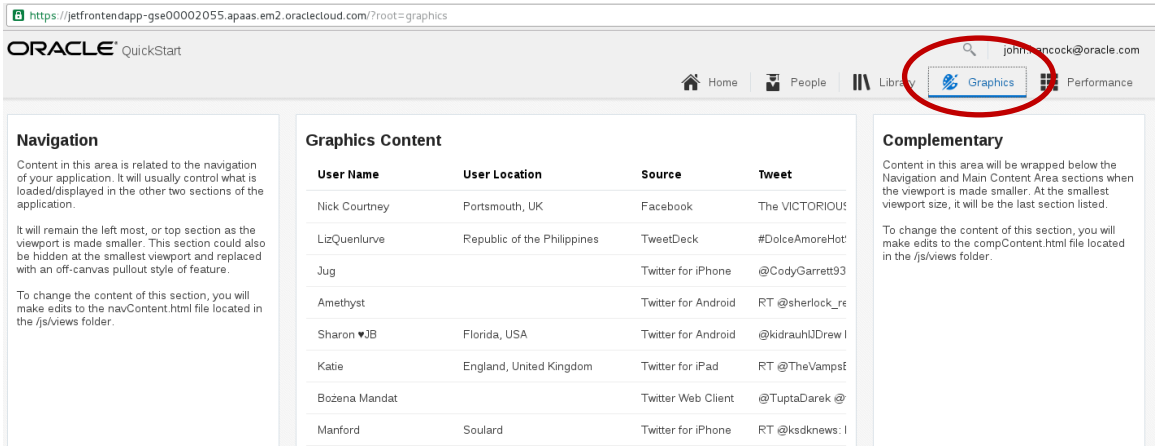


☐ Click on **Deploy**. Wait for the Deploy Status to change to **Deployment update in progress,** and then change to **Last deployment succeeded – Just now**.

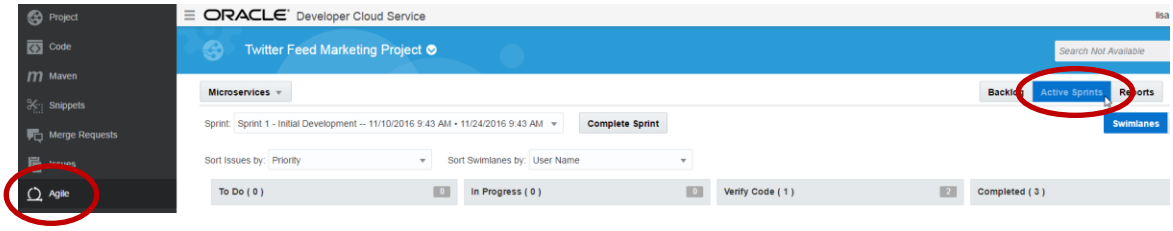
Test the JETFrontEndAPP UI in the Cloud

☐ Once the service has successfully deployed, click on the **JETFrontEndApp** link

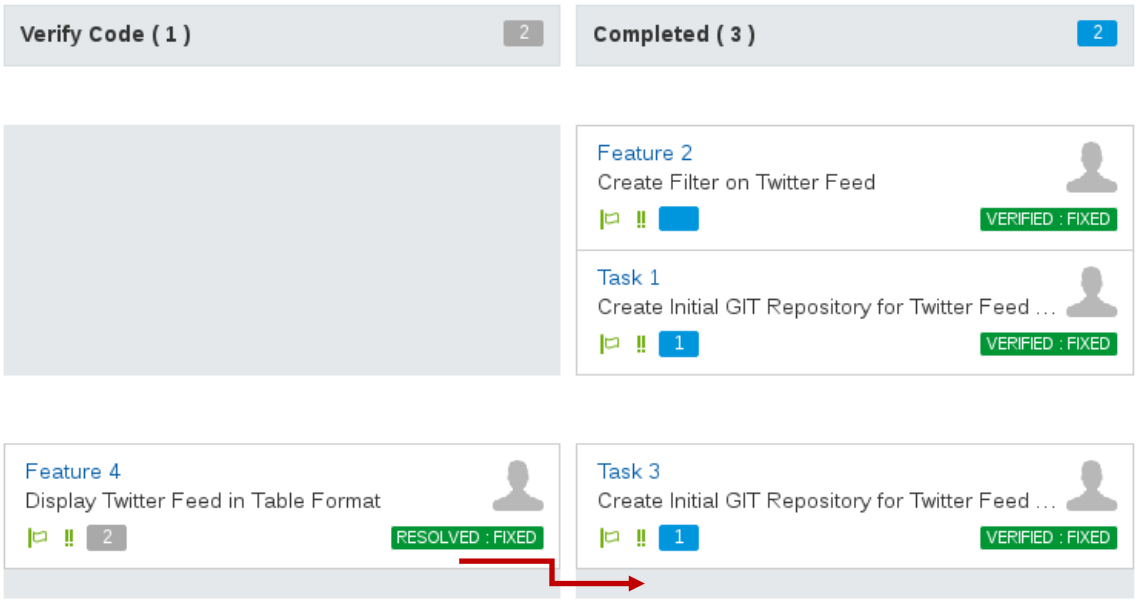
☐ When the new browser tab loads, click **Graphics** to display twitter feed data.



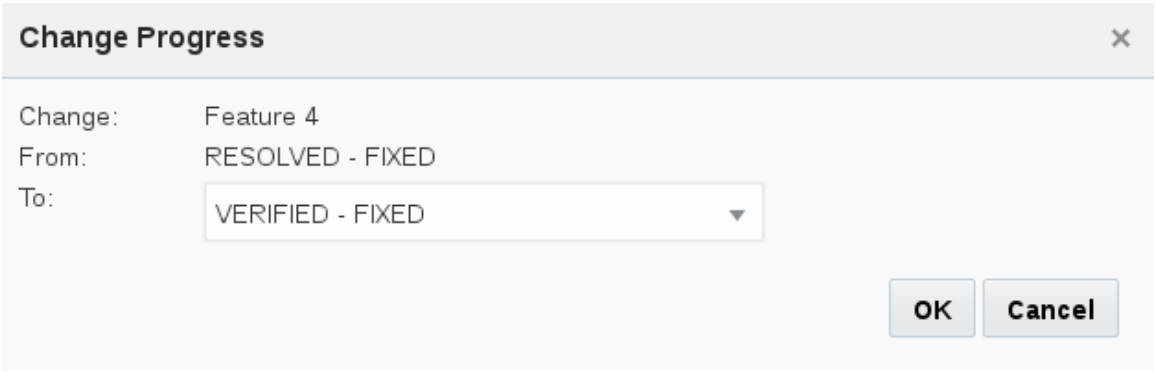
☐ To complete the Sprint Feature, click on **Agile** in the Twitter Feed Marketing Project Dashboard. Then click on the **Active Sprints** button.



☐ Complete the feature request by Dragging and Dropping **Feature 4** (Display Twitter Feed in Table Format) from the **Verify** Column to the **Completed** Column.



☐ Set the Status to **VERIFIED – FIXED** and click **OK**



☐ You are now ready to move to the next lab.

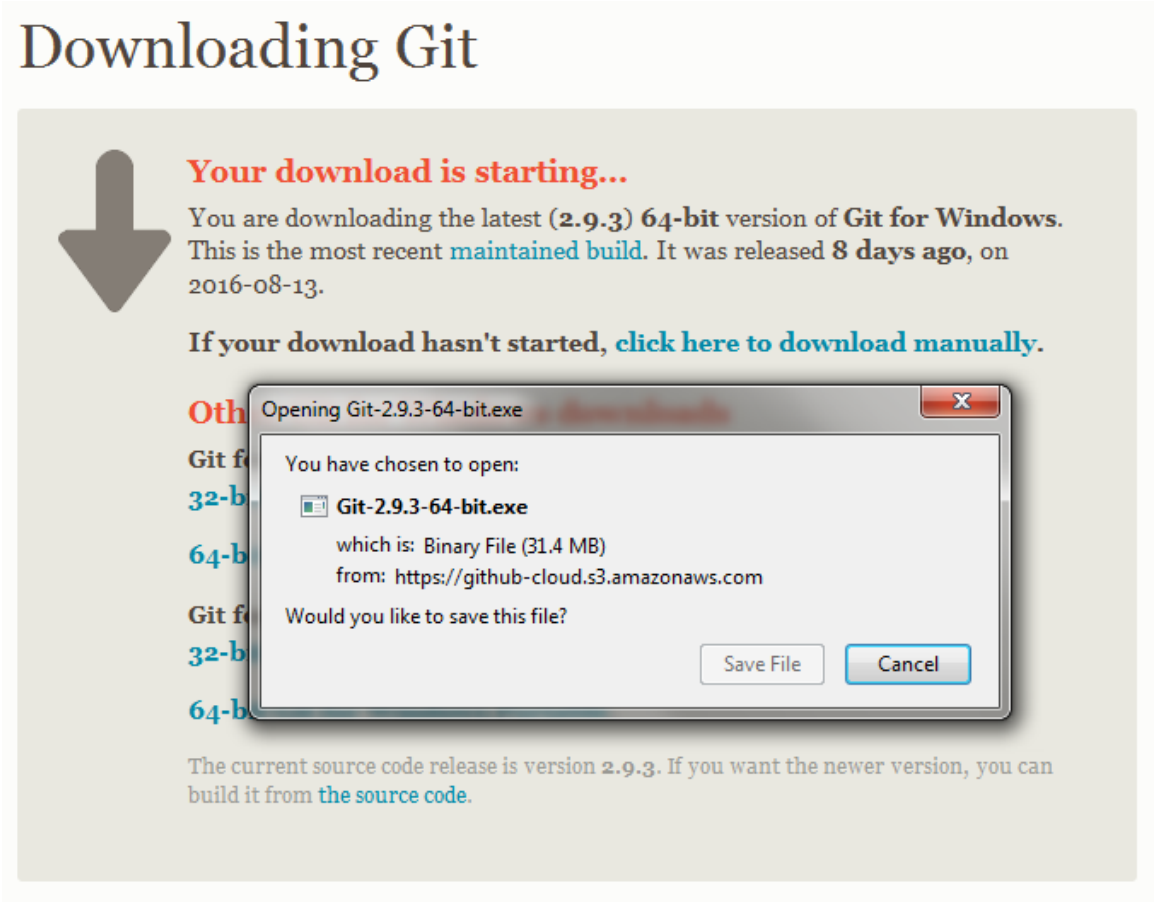
Appendix 1 – Installing Brackets and Git

Download and Install Git

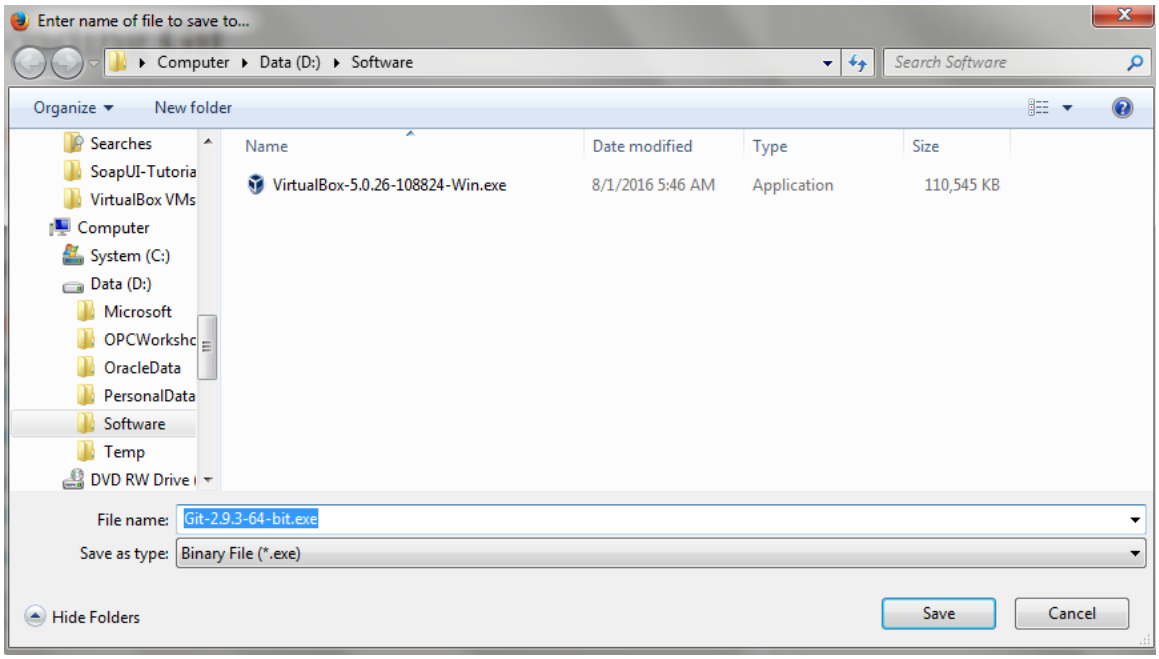
STEP 23: Download Git

☐ Go to the following URL: `https://git-scm.com/downloads`

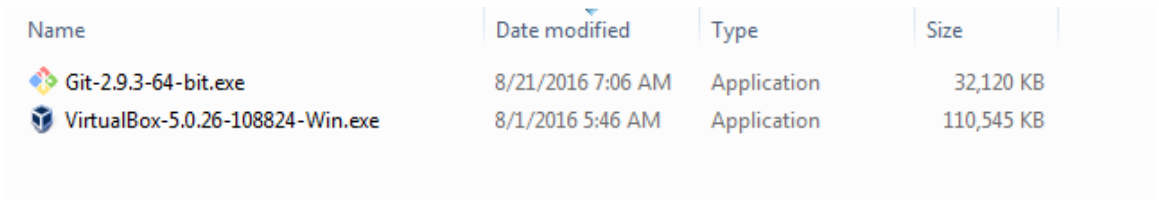
☐ Select you OS. In our example we will show how to install on Windows. Click **Windows** Download and click **Save File**



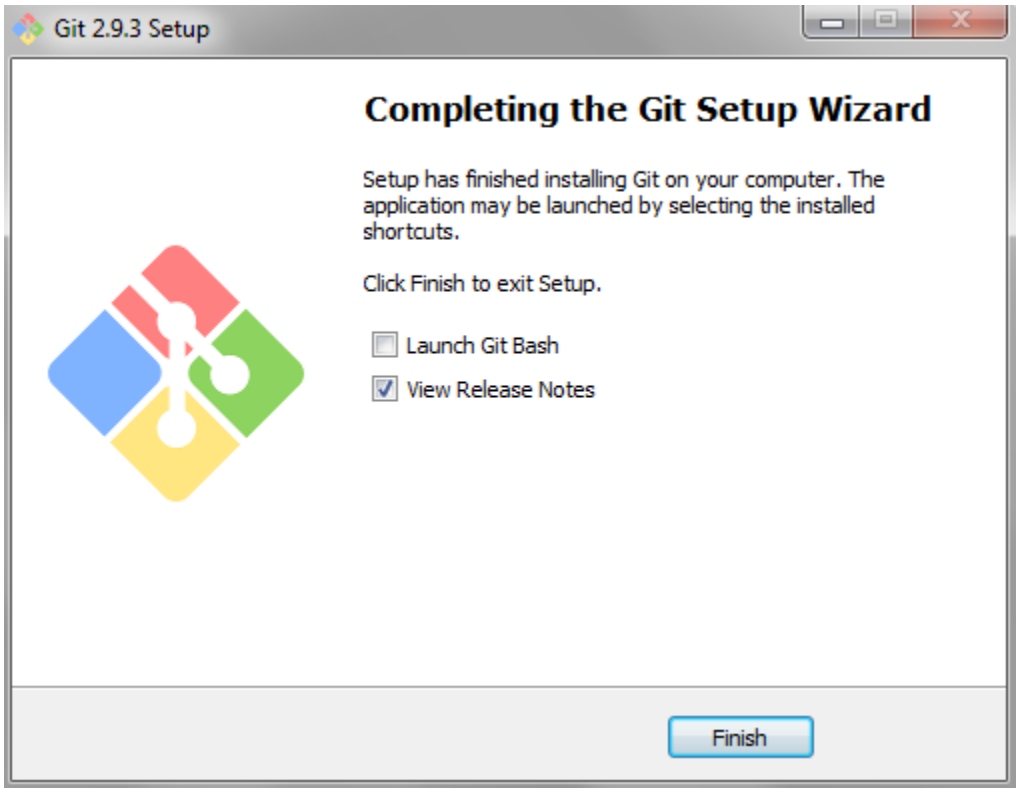
☐ Select you download location and click **Save**. We will use D:\Software



☐ Open Windows Explorer and navigate you where you downloaded the Git executable. Double click on Git executable to start install process.



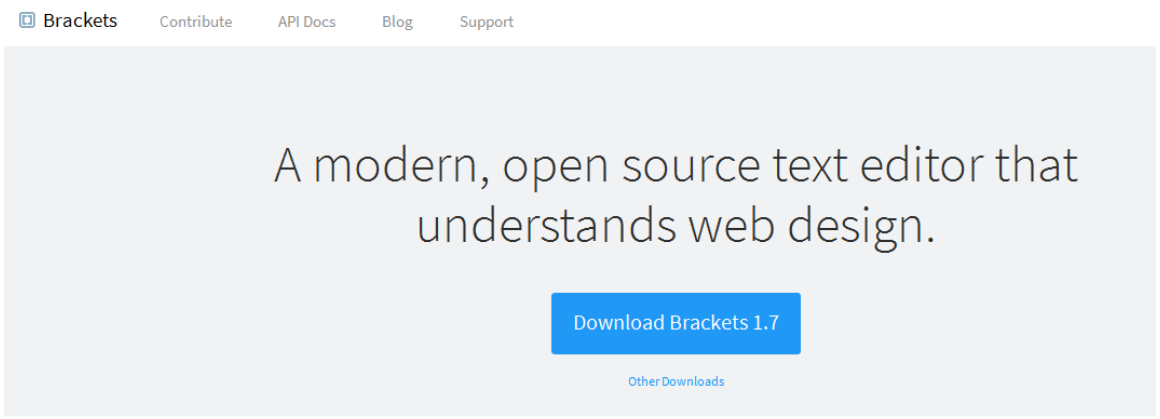
☐ Run through the installation process



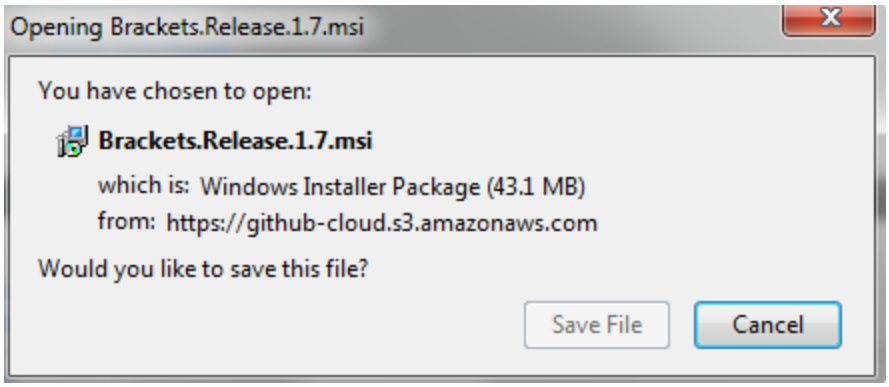
Download and Install Brackets Text Editor

STEP 24: Download Brackets Text Editor

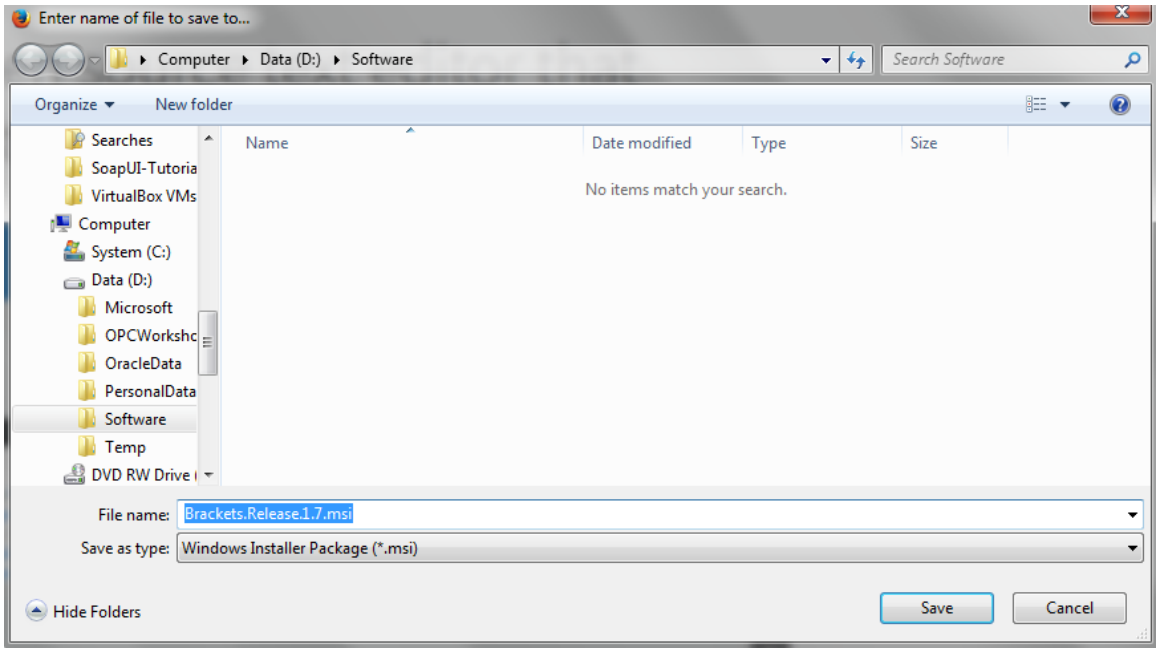
☐ Go to the following URL: **`http://brackets.io`**



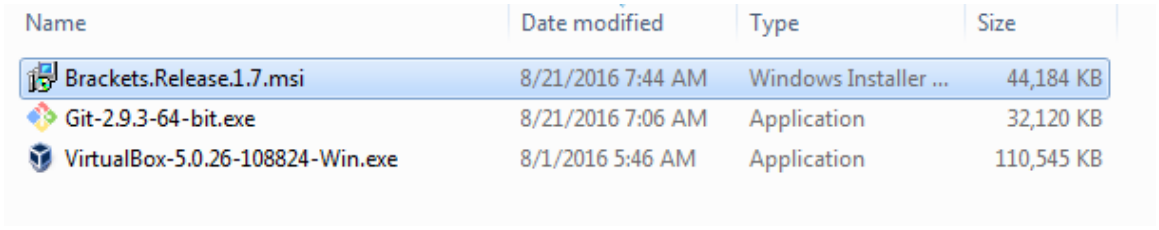
☐ Click **Download Brackets 1.7** then click **Save File**



☐ Select you download location and click **Save**. We will use D:\Software



☐ Open Windows Explorer and navigate you where you downloaded Brackets. Double click on Git executable to start install process.

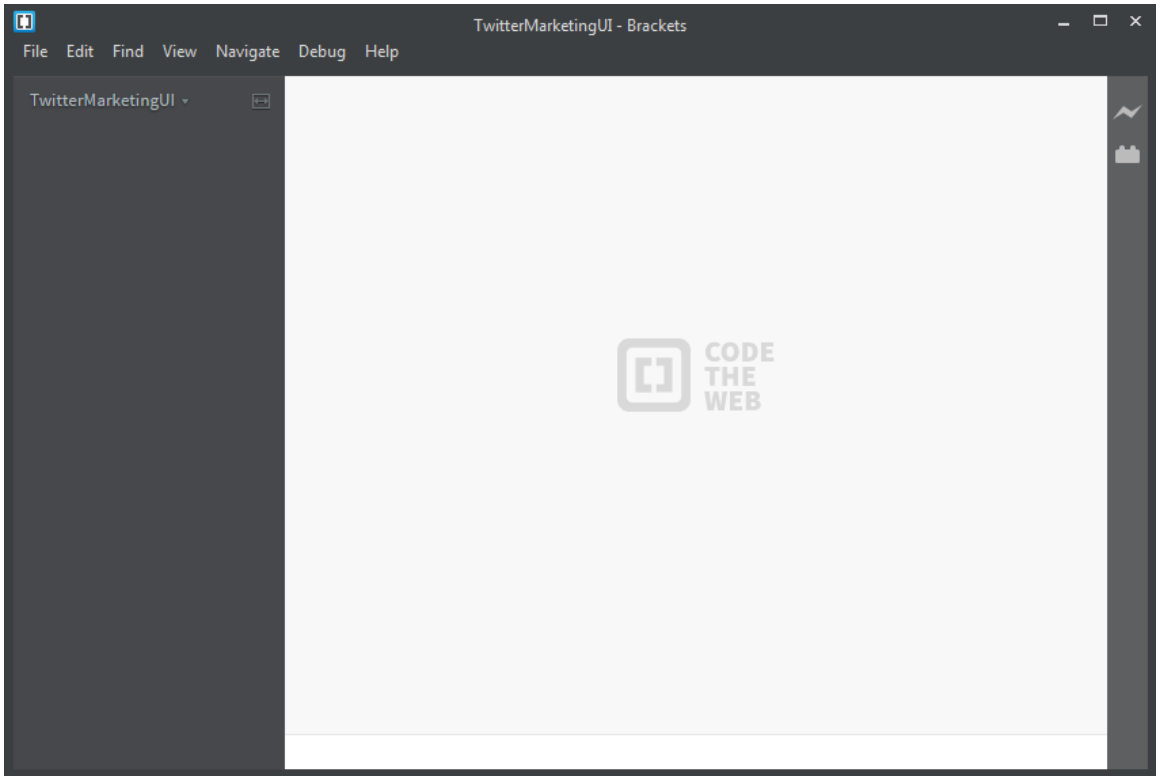


☐ Run through the installation process

Start Brackets and Configure Git

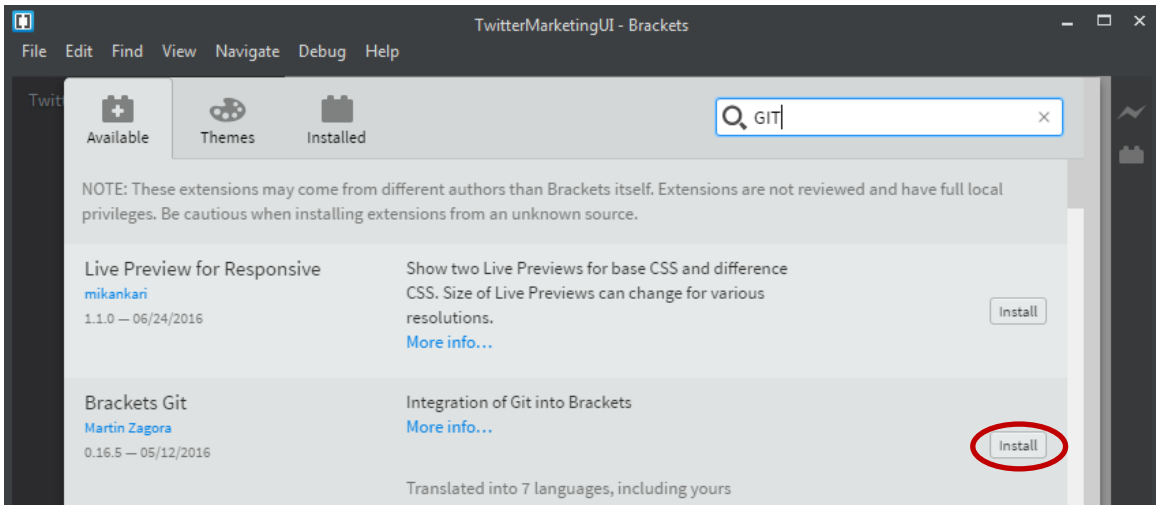
STEP 25: Start Brackets and Configure Git

☐ Create directory **TwitterMarketingUI**. From Windows Explorer navigate to the directory **TwitterMarketingUI**, right click and select **Open as Brackets Project**

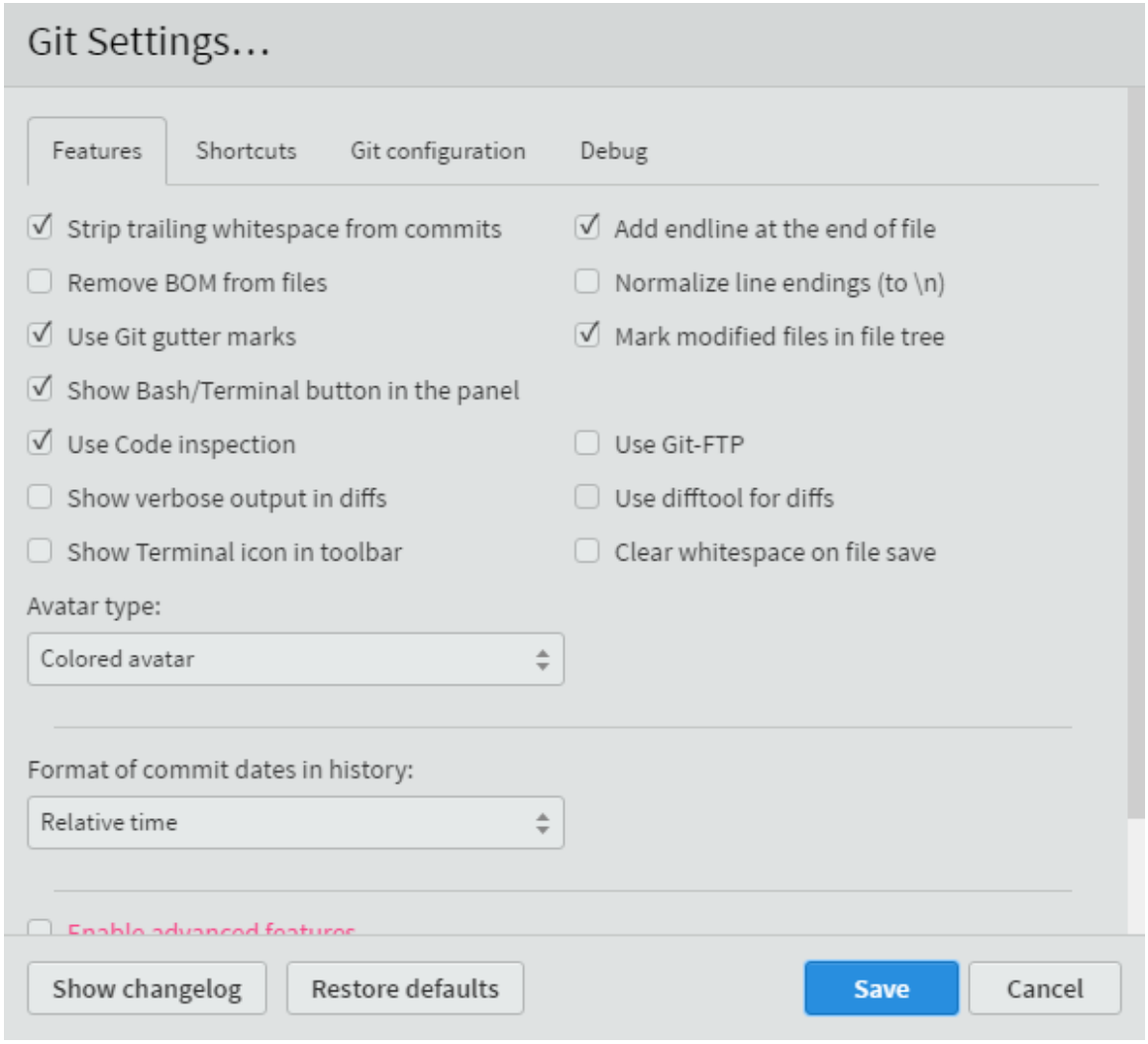


☐ Select **File -> Extension Manager...**

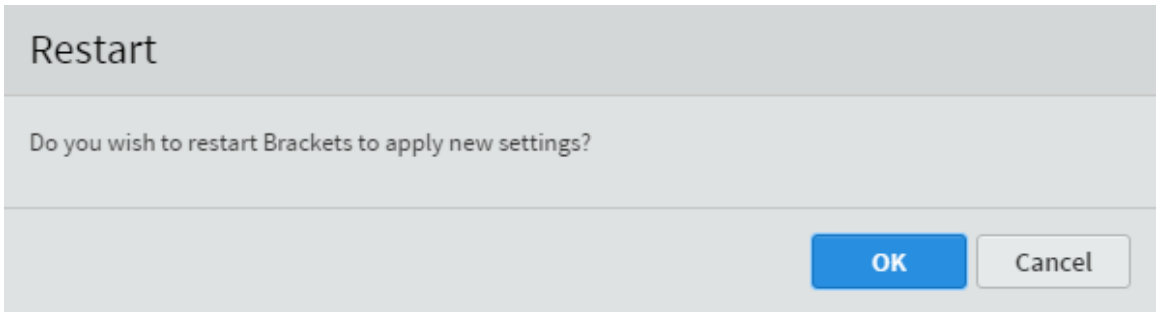
☐ In search window enter **GIT**. Click **Install** for **Brackets Git**



☐ After install completes, leave defaults for Git Settings and click **Save**



☐ Click OK to restart Brackets



☐ You will now see the Git icon on the right hand panel