



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

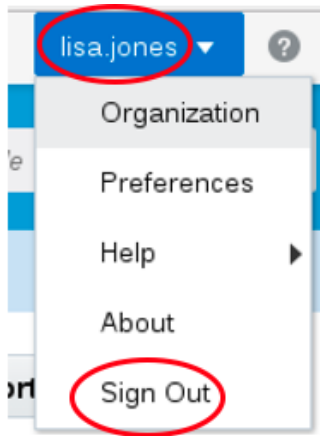
Introduction	2
Objectives	2
Required Artifacts.....	2
Outline.....	3
Create Initial Twitter Marketing UI Service	4
Explore Developer Cloud Service	4
Create Initial Git Repository.....	10
Create Default Build and Deployment Process	12
Verify default deployment of Twitter Marketing UI	20
Extend default 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

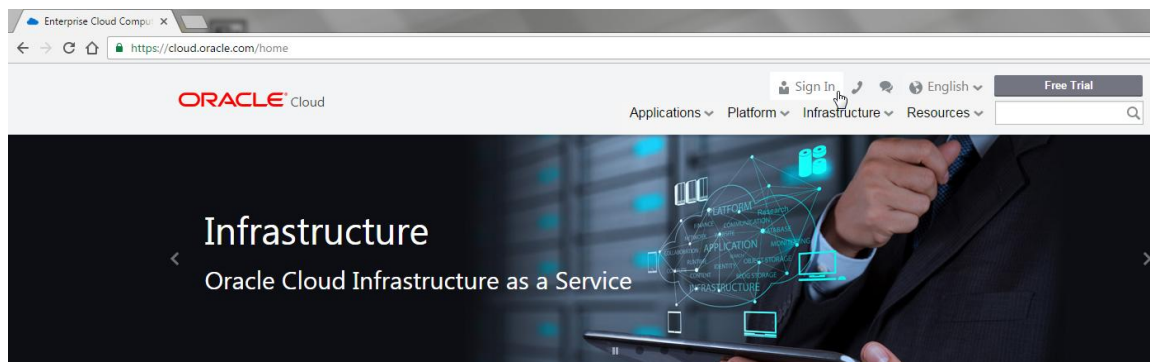
- ☐ If you just completed lab 200, or if you are still logged in as Lisa.Jones, you will need to first sign out before continuing this lab. Sign out by clicking on the user's name (lisa.jones) at the top right corner of the screen, then selecting **Sign Out** from the dropdown menu.



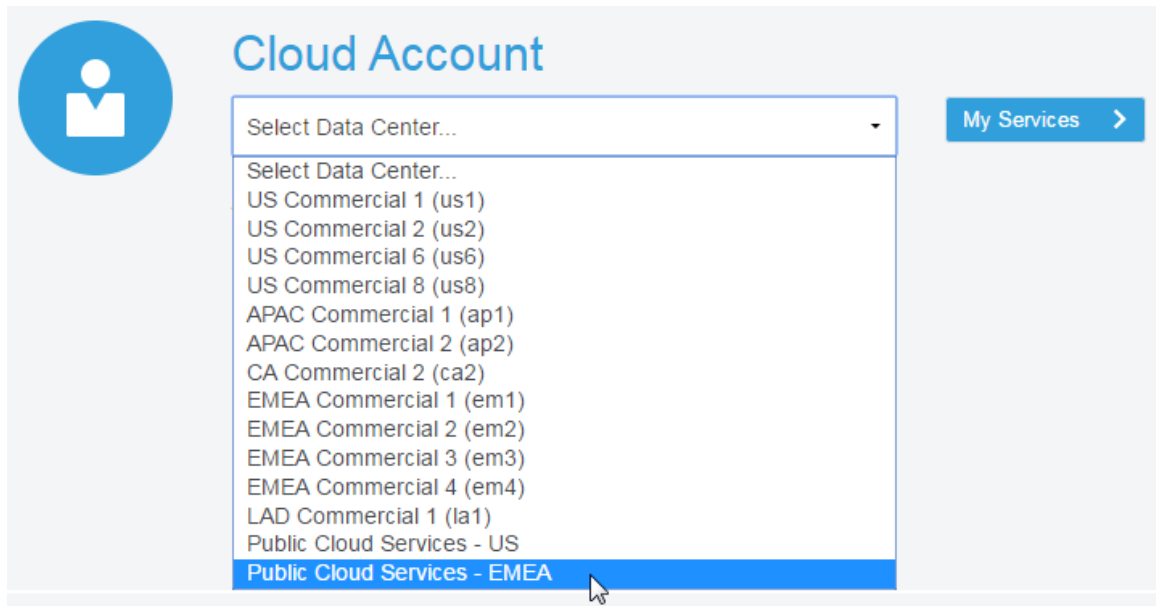
- ☐ Now we can login again. From any browser, go to the following URL:

<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.



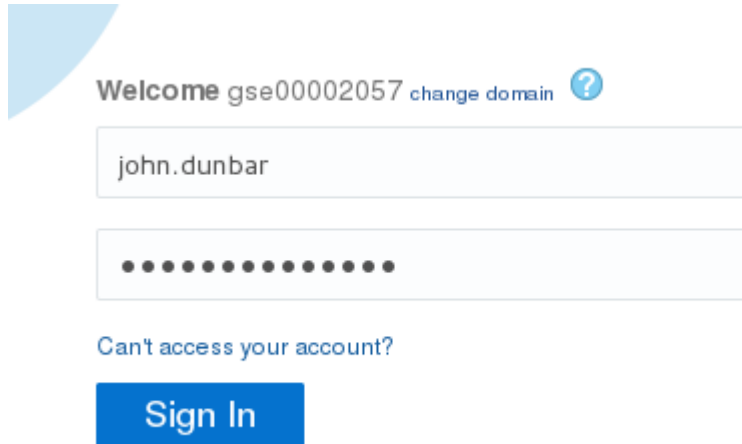
- ☐ Enter your identity domain and click **Go**

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

Enter your Identity Domain

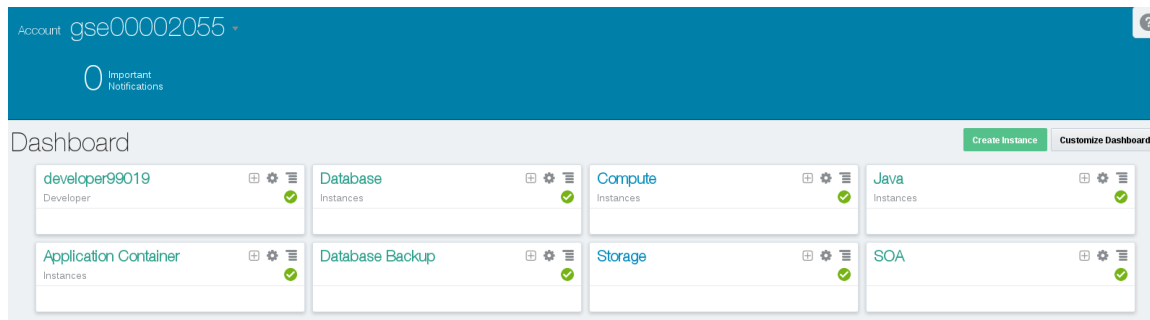
- ☐ Once your Identity Domain is set, enter your User Name and Password and click **Sign In**

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” identity of John Dunbar, the JavaScript Developer.



The login form features a light blue header with the text 'Welcome gse00002057' followed by a 'change domain' link and a help icon. Below this is a text input field containing 'john.dunbar'. Underneath is a password input field represented by a series of dots. A link 'Can't access your account?' is positioned below the password field. At the bottom is a prominent blue 'Sign In' button.

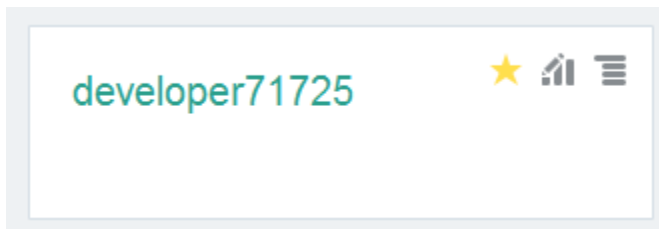
- ☐ You will be presented with a Dashboard displaying the various cloud services available to this account.



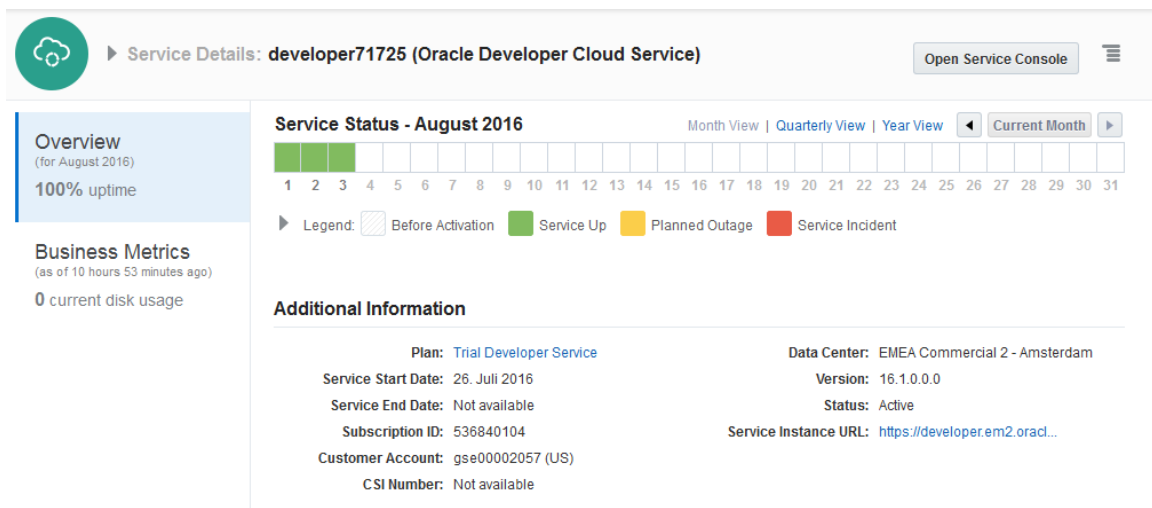
STEP 2: Login to Developer Cloud Service

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.

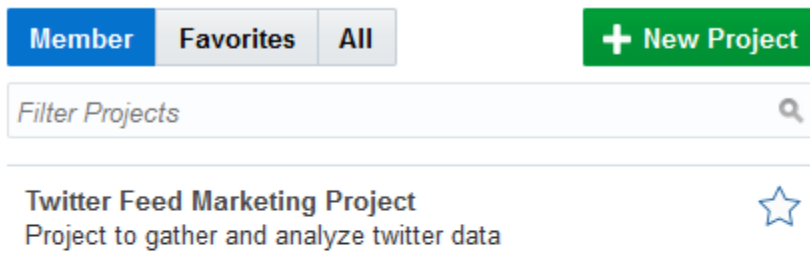
- ☐ From Cloud UI dashboard click on the **Developer** service. In our example the Developer Cloud Service is named **developer71725**.



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

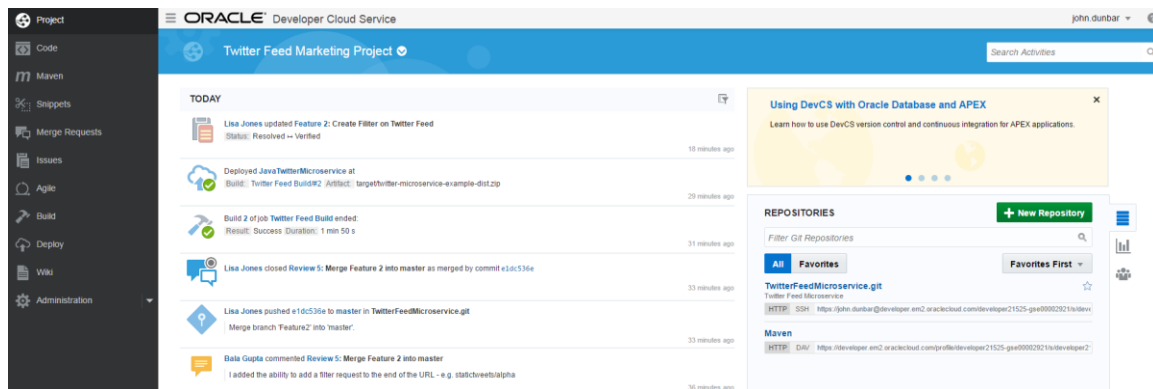


- ❑ Click **Open Service Console** for the Oracle Developer Cloud Service. The Service Console will list all projects that you are currently a member.

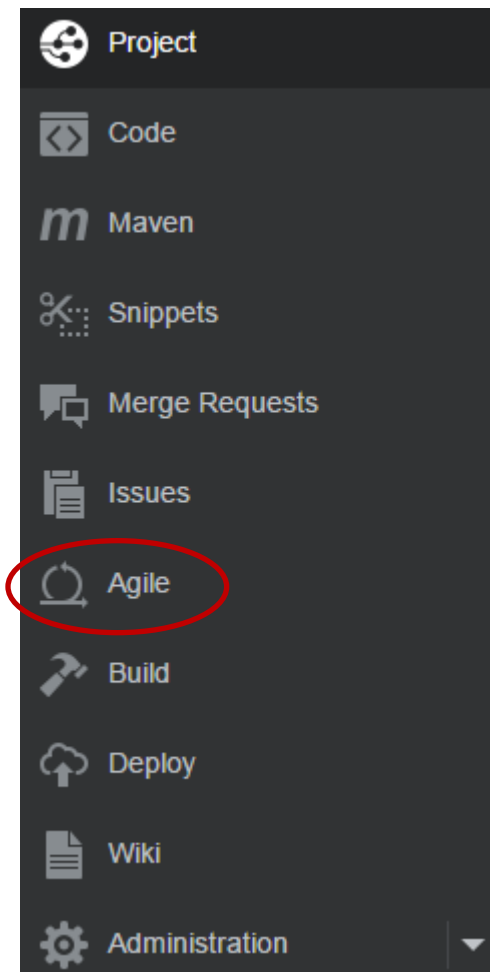


STEP 3: Review Agile Board

- ❑ Click **Twitter Feed Marketing Project** to access the project.

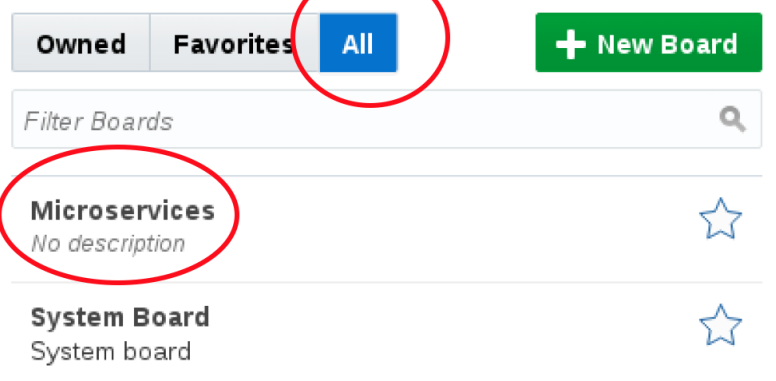


- ☐ Click on **Agile** in navigation panel.



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

Find Board



- ❑ Click on the **Microservices** Board **Active Sprints**.

The screenshot shows the Jira Microservices Board interface. The 'Active Sprints' tab is highlighted with a red circle. The board is for 'Sprint 1 - Initial Development' running from 8/17/2016 to 8/31/2016. It has four swimlanes: 'To Do (2)', 'In Progress (0)', 'Verify Code (0)', and 'Completed (2)'. Under 'bala.gupta', there are two tasks: 'Feature 2: Create Filter on Twitter Feed' (VERIFIED, FIXED) and 'Task 1: Create Initial GIT Repository for Twitter Feed...' (VERIFIED, FIXED). Under 'john.dunbar', there are two tasks: 'Feature 4: Display Twitter Feed in Table Format' (UNCONFIRMED) and 'Task 3: Create Initial GIT Repository for Twitter Feed...' (UNCONFIRMED).

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.

The 'Change Progress' dialog box is shown. It has a title bar with a close button. The 'Change:' field is 'Task 3'. The 'From:' field is 'UNCONFIRMED'. The 'To:' field is a dropdown menu currently showing 'ASSIGNED'. At the bottom right, there are 'OK' and 'Cancel' buttons. A mouse cursor is pointing at the 'OK' button.

The screenshot shows the Jira board after the task has been moved. Under 'john.dunbar', 'Feature 4' remains 'UNCONFIRMED'. 'Task 3' has been moved to the 'In Progress' swimlane and its status is now 'ASSIGNED'.

- ☐ Click on **Project**.
- ☐ Click on **New Repository** to create a new Git Repository



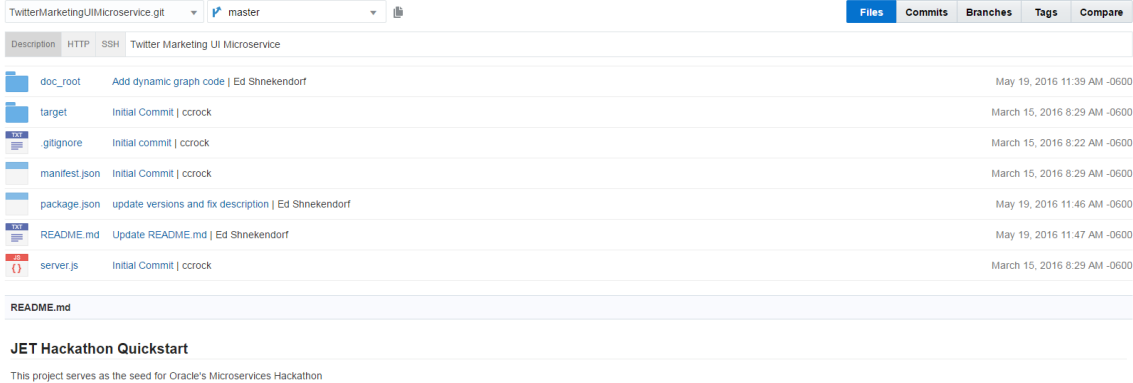
- ☐ In the New Repository wizard enter the following information and click **Create**.

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

The 'New Repository' wizard is shown with the following details:

- Name:** TwitterMarketingUIMicroservice
- Description:** Twitter Marketing UI Microservice
- Initial content:** ☒ Import existing repository
- URL:** `https://github.com/oraclenassolutionengineering/JETTwitterQuickStart.git`
- Credentials:** for non-public repos
- Buttons:** Create, Cancel

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



Create Default Build and Deployment Process

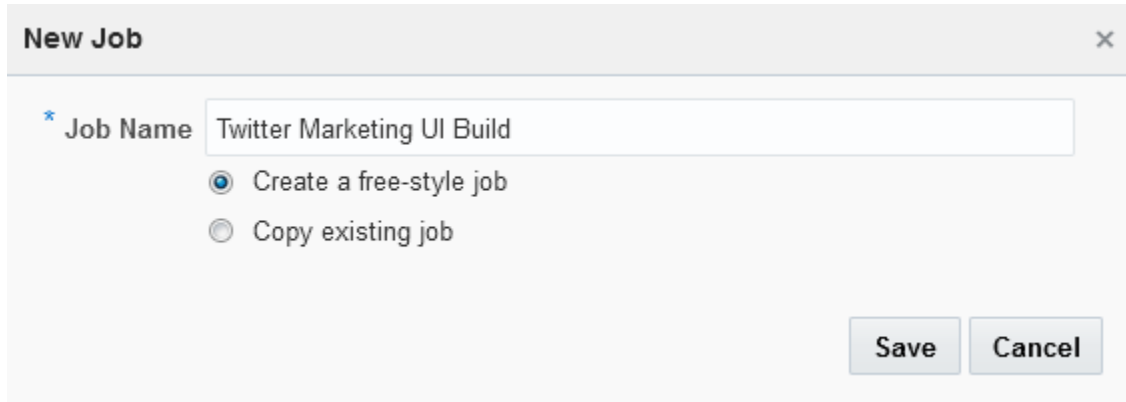
STEP 5: Create Default Build Process

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**.

A 'New Job' dialog box with a close button (X) in the top right corner. It contains a text input field for 'Job Name' with the value 'Twitter Marketing UI Build'. Below the input field are two radio button options: 'Create a free-style job' (selected) and 'Copy existing job'. At the bottom right are 'Save' and 'Cancel' buttons.

New Job [X]

* Job Name

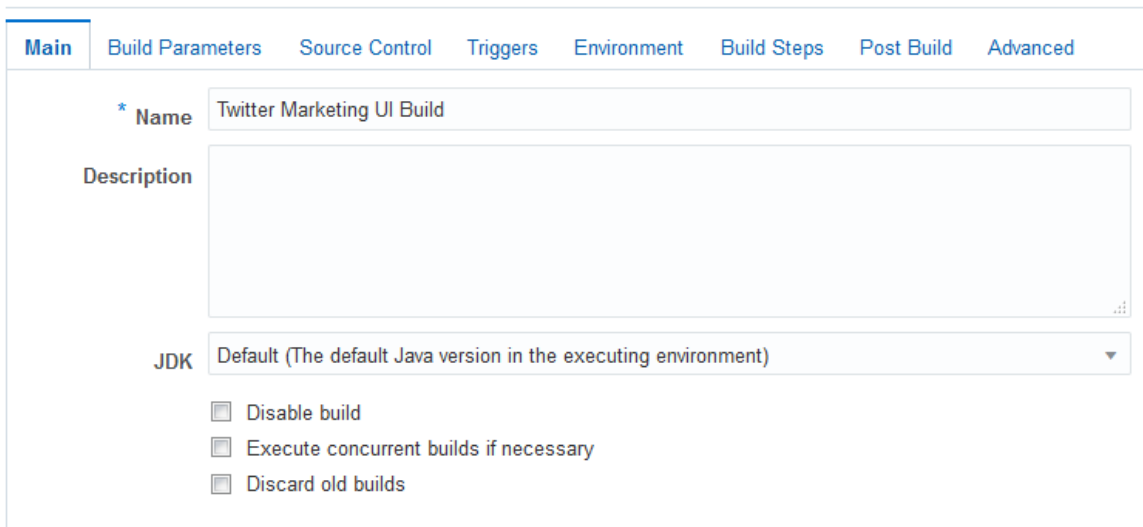
☒ Create a free-style job

☐ Copy existing job

Save Cancel

- ☐ You are now placed into the job configuration screen.

[< Jobs Overview](#) | [Twitter Marketing UI Build](#) | [Configure build job](#)

A job configuration screen with tabs: 'Main', 'Build Parameters', 'Source Control', 'Triggers', 'Environment', 'Build Steps', 'Post Build', and 'Advanced'. The 'Main' tab is active. It contains a 'Name' field with 'Twitter Marketing UI Build', a 'Description' text area, a 'JDK' dropdown menu set to 'Default (The default Java version in the executing environment)', and three checkboxes: 'Disable build', 'Execute concurrent builds if necessary', and 'Discard old builds'.

Main | Build Parameters | Source Control | Triggers | Environment | Build Steps | Post Build | Advanced

* Name

Description

JDK

☐ Disable build

☐ Execute concurrent builds if necessary

☐ Discard old builds

- ☐ 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.

[< Jobs Overview](#) | [Twitter Marketing UI Build](#) | **Configure build job**

Main Build Parameters **Source Control** Triggers Environment Build Steps Post Build Advanced

To integrate the Build System with Source Control, select an option below and then configure the required settings.

☐ None
☒ Git

Repositories Add

* Repository Select a Repository ×

TwitterFeedMicroservice.git

TwitterMarketingUIMicroservice.git

Branches Add

► **Advanced Git Settings**

- ☐ 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.

[< Jobs Overview](#) | [Twitter Marketing UI Build](#) | **Configure build job**

Main Build Parameters Source Control **Triggers** Environment Build Steps Post Build Advanced

☐ When these jobs are built
☐ Based on this schedule
☒ Based on SCM polling schedule

Schedule `*/1****`

Polling will automatically trigger on commits to the SCM. Specify a schedule only for any additional polling required.

☐ When Maven dependencies have been updated by Maven 3 integration

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

[Jobs Overview](#) | [Twitter Marketing UI Build](#) | **Configure build job**

Main Build Parameters Source Control Triggers Environment **Build Steps** Post Build Advanced

Configure Build Steps

Add Build Step ▼
Execute shell
Invoke Ant
Invoke Maven 2 (Legacy)
Invoke Maven 3
Invoke Gradle
Invoke NodeJS
Copy Artifacts

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

[Jobs Overview](#) | [Twitter Marketing UI Build](#) | **Configure build job**

Main Build Parameters Source Control Triggers Environment **Build Steps** Post Build Advanced

Configure Build Steps **Add Build Step** ▼

Execute shell

Command 1 `npm install` ×

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

[Jobs Overview](#) | [Twitter Marketing UI Build](#) | **Configure build job**

Main Build Parameters Source Control Triggers Environment Build Steps **Post Build** Advanced

☐ Aggregate downstream test results
☐ Build other jobs
☒ Archive the artifacts

* Files To Archive `**/target/**`

☐ Enable auto validation for file masks

Excludes

☐ Discard all but the last successful/stable artifact to save disk space

Compression Type `GZIP` ▼

- ☐ 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:

The screenshot shows the 'Twitter Marketing UI Build' configuration page. At the top right is a 'Build Now' button. Below it, the 'Description' section states 'No description available'. The 'Permalinks' section shows 'Last'. The 'Notifications' section has 'On' and 'Off' buttons, with 'On' selected, and a 'CC Me' button. The 'Build History' section shows a 'New Build' with a progress bar and 'N/A' status. Below this is a table with columns: Status, Build, Time, Duration, and Console. The table has one row with a sun icon, '#1', 'Just now', 'N/A', and a console icon. To the right, there is a 'Build Trend' section and a 'Console' icon.

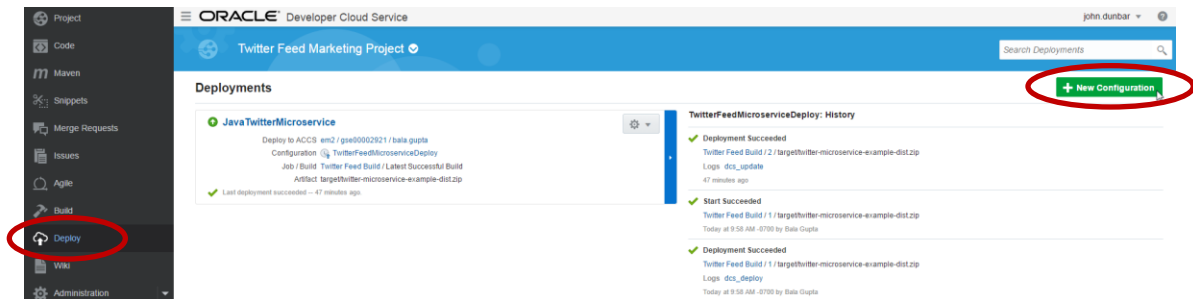
- ☐ Once the build begins, it should take about approximately 1 to 2 minutes for the build to complete. Wait for the build to complete before continuing on to the next step, as we need the build artifact to create the deployment configuration.

The screenshot shows the 'Twitter Marketing UI Build' configuration page after the build has completed. The 'Build Now' button is no longer visible. The 'Description' section still states 'No description available'. The 'Permalinks' section now shows 'Last | Successful | Completed | Stable'. The 'Notifications' section has 'On' and 'Off' buttons, with 'On' selected, and a 'CC Me' button. The 'Build History' section shows a 'New Build' with a progress bar and 'N/A' status. Below this is a table with columns: Status, Build, Time, Duration, and Console. The table has one row with a green checkmark, '#1', 'Just now', '46 s 73 ms', and a console icon. To the right, there is a 'Build Trend' section showing a graph of 'duration in seconds' (y-axis: 20, 30, 40, 50) and a 'Console' icon. The 'Artifacts of Last Successful Build' section shows a 'target' icon and '(all files in zip)'.

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
```

* Configuration Name

* Application Name

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

* Configuration Name

* Application Name

* Deployment Target **New**

Type ☒ On Demand ☐ Automatic

* Job

* Build

* Artifact

Java Cloud Service - SaaS Extension...

Java Cloud Service...

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>
```

Deploy to Application Container Cloud ×

* Data Center

EMEA Commercial 2 - em2

* Identity Domain

gse00002057

* Username

john.dunbar

* Password

••••••••••

✓ Successful

Use Connection

- ☐ 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.

* Configuration Name

TwitterMarketingUIDeploy

* Application Name

JETFrontEndApp

* Deployment Target

em2 / gse00002057 / john.dunbar

New ▾

ACCS Properties

Runtime

☐ Java
 ☒ Node

Subscription

☒ Hourly
 ☐ Monthly

Type

☐ On Demand
 ☒ Automatic
 ☒ Deploy stable builds only

* Job

Twitter Marketing UI Build

* Artifact

target/jet-quickstart-client-dist.zip

☐ Click **Save**

Save Cancel

* Configuration Name

TwitterMarketingUIDisplay

* Application Name

JETFrontEndApp

* Deployment Target

em2 / gse00002057 / john.dunbar

New ▾

ACCS Properties

Runtime

☐ Java
 ☒ Node

Subscription

☒ Hourly
 ☐ Monthly

Type

☐ On Demand
 ☒ Automatic
 ☒ Deploy stable builds only

* Job

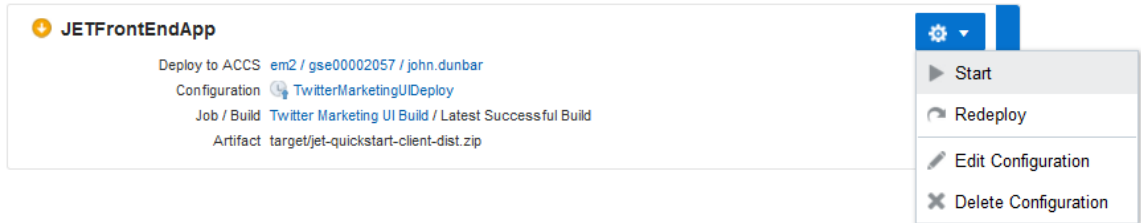
Twitter Marketing UI Build

* Artifact

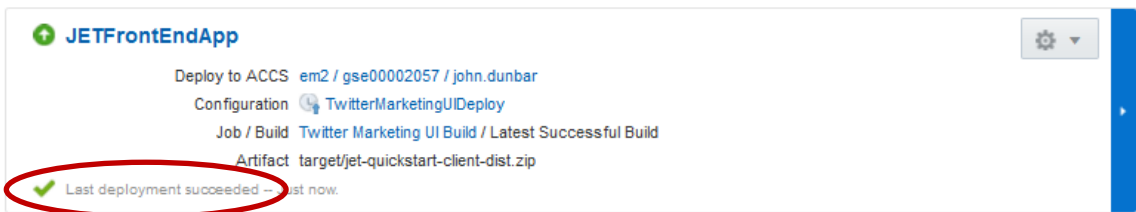
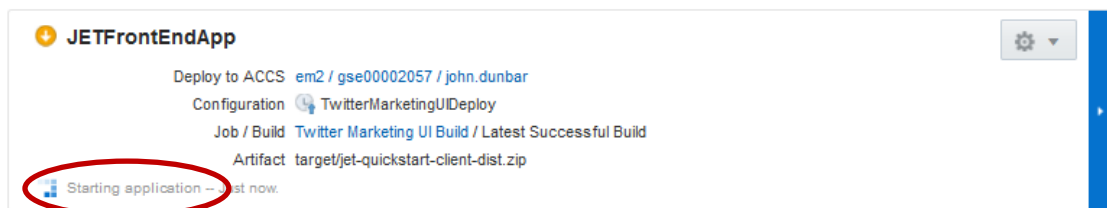
target/jet-quickstart-client-dist.zip

Revised: November 17, 2016
Copyright © 2012, Oracle Corporation

- ☐ Click drop down and select **Start**



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



Verify default deployment of Twitter Marketing UI

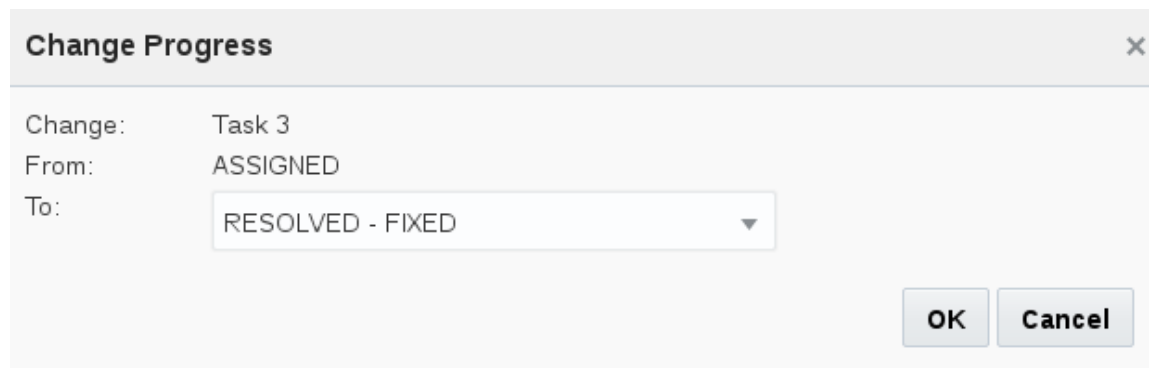
STEP 7: Change status to Verified

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**

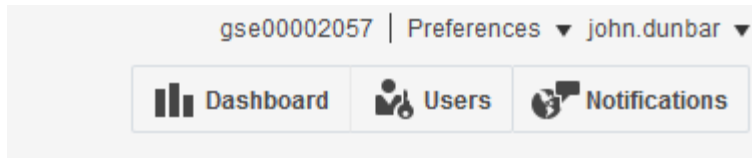



- ☐ The code is now ready for verification before moving to Completed

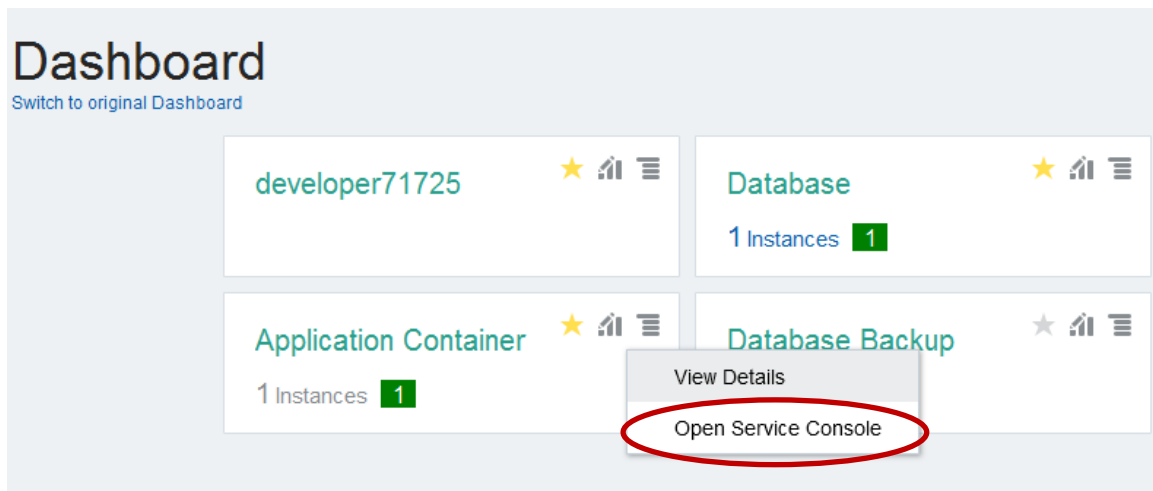


STEP 8: Login to Oracle Application Container Cloud Service

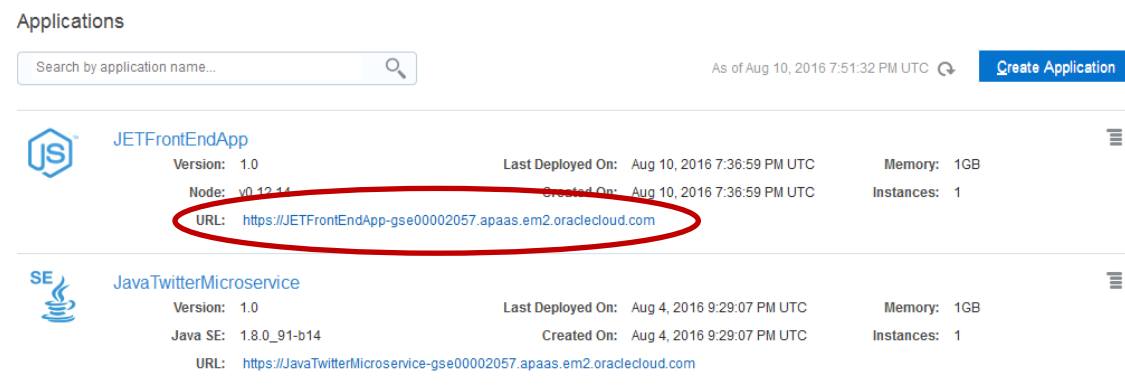
- Navigate back to the Oracle Public Cloud tab. Click **Dashboard** to return back to main Cloud Service Dashboard.



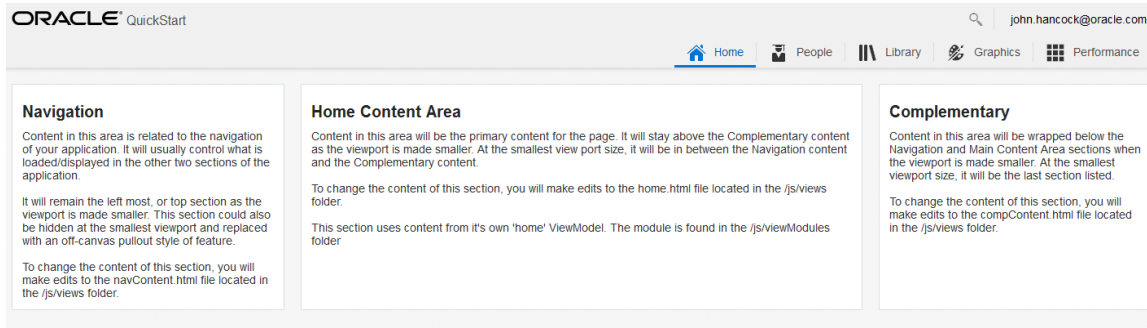
- On the Application Container Cloud Service (ACCS) click  and select **Open Service Console**



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



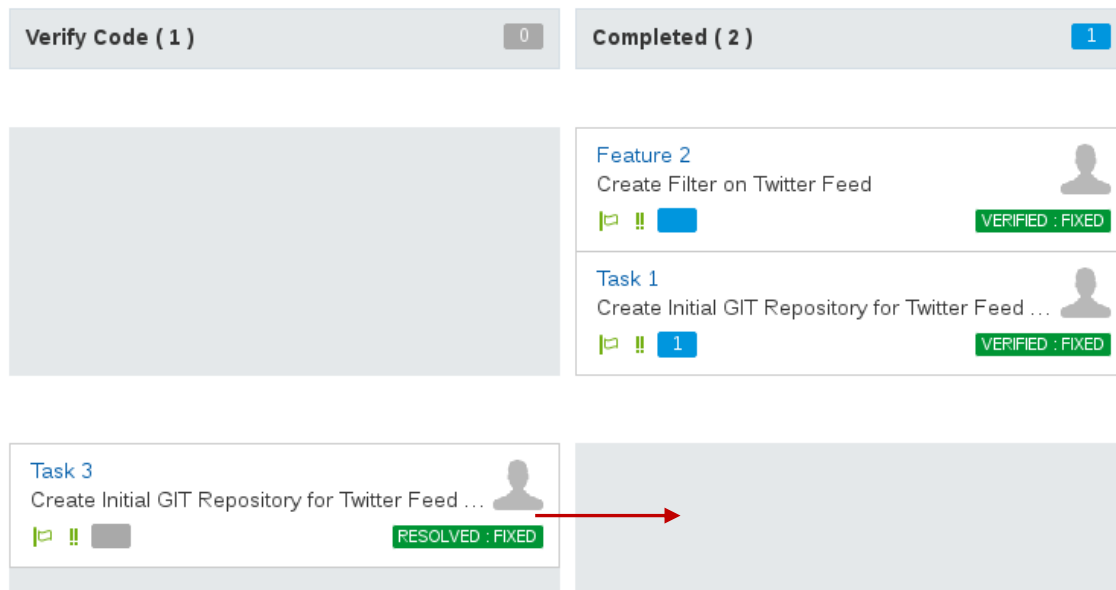
- ☐ Click on URL to bring up the application.



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**

Change Progress

Change: Task 3

From: RESOLVED - FIXED

To:

VERIFIED - FIXED

OK

Cancel

- ☐ Your Sprint should now look like the following:

Microservices

BacklogActive SprintsReportsBoard

Sprint: Sprint 1 - Initial Development -- 8/17/2016 12:35 PM • 8/31/2016 12:35 PMComplete Sprint

SwimlanesColumns

Sort Issues by: PrioritySort Swimlanes by: User Name

To Do (1)2

In Progress (0)0

Verify Code (0)0

Completed (3)1

Feature 4

Display Twitter Feed in Table Format

UNCOMPLETED

john.dunbar

Feature 2

Create Filter on Twitter Feed

VERIFIED - FIXED

bala.gupta

Task 1

Create Initial GIT Repository for Twitter Feed ...

VERIFIED - FIXED

bala.gupta

Task 3

Create Initial GIT Repository for Twitter Feed ...

VERIFIED - FIXED

bala.gupta

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.

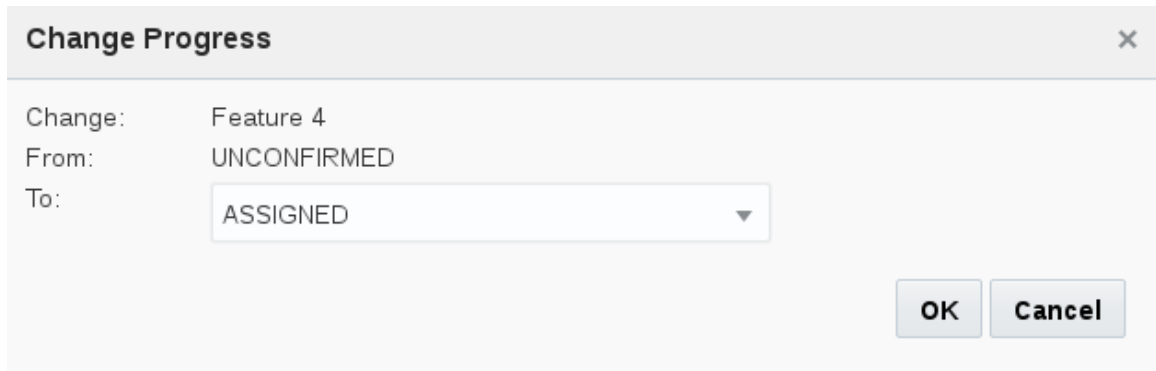
STEP 10: Move Task to In Progress

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**.

The screenshot displays the Agile Sprints interface. At the top, there are two columns: 'To Do (1)' with a count of 2, and 'In Progress (0)' with a count of 0. Below the columns, the user 'bala.gupta' is listed. Under 'bala.gupta', there are two large empty rectangular boxes. Below these, the user 'john.dunbar' is listed. Under 'john.dunbar', there is a task card for 'Feature 4' with the description 'Display Twitter Feed in Table Format'. The card includes a flag icon, an exclamation mark icon, a count of 2, and a green 'UNCONFIRMED' label. A red arrow points from the task card to the 'In Progress' column, indicating the move action.

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



A dialog box titled "Change Progress" with a close button (X) in the top right corner. It contains three labels: "Change:" with the value "Feature 4", "From:" with the value "UNCONFIRMED", and "To:" with a dropdown menu showing "ASSIGNED". At the bottom right, there are two buttons: "OK" and "Cancel".

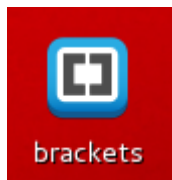
Clone Project to Brackets Text Editor

STEP 11: Start Brackets Text Editor

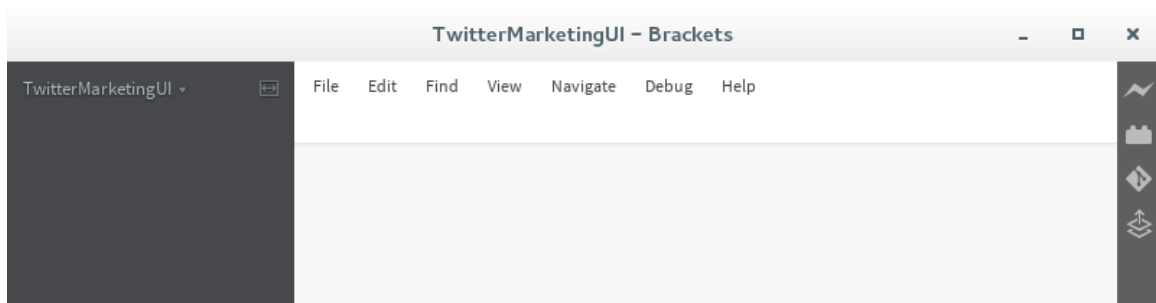
- ☐ Start **Brackets** text editor. How you start Brackets will depend on your OS. We have documented how to start Brackets from our OEL image.

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

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

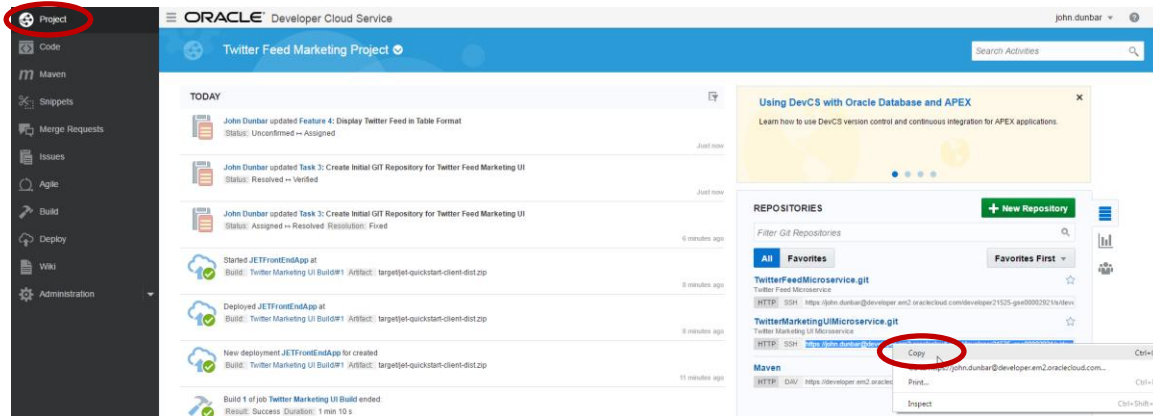



- ☐ Brackets should open with the **TwitterMarketingUI** folder already loaded.

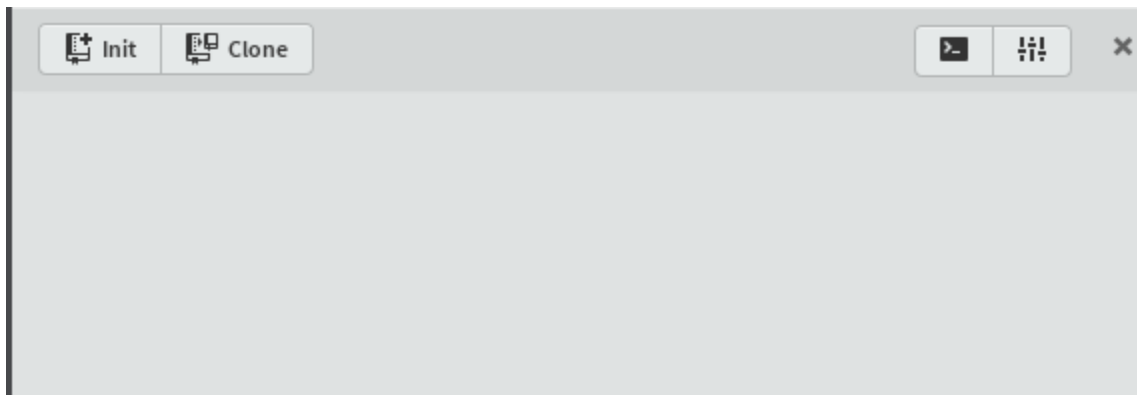


STEP 12: Copy GIT URL

- ☐ 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.

Clone repository

Enter Git URL of the repository you want to clone:

`https://john.dunbar@developer.em2.oraclecloud.com/developer71725-gse00002057/s/devel`

You don't need to fill out username/password if your credentials are managed elsewhere. Use this only when your operation keeps timing out.

Username:

Password:

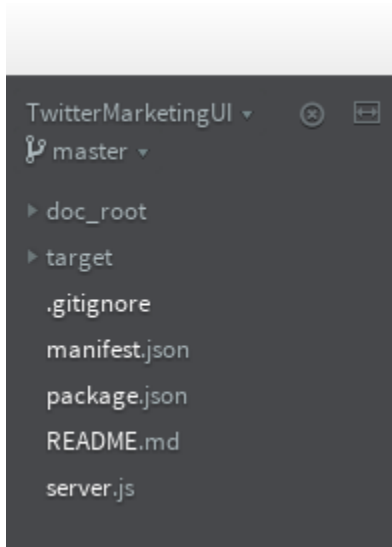
☒ Save credentials to remote url (in plain text)

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

Git operation in progress...

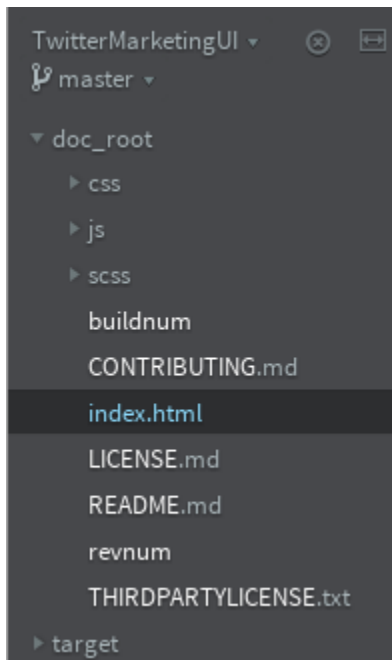
```
Running command: git clone
https://john.dunbar:fervent%404Spate@developer.em2.oraclecloud.com/developer71725-
gse00002057/s/developer71725-gse00002057_twitter-feed-marketing-
project_1165/scm/TwitterMarketingUIMicroservice.git . --progress
```


- ☐ You now have a local copy of the repository.

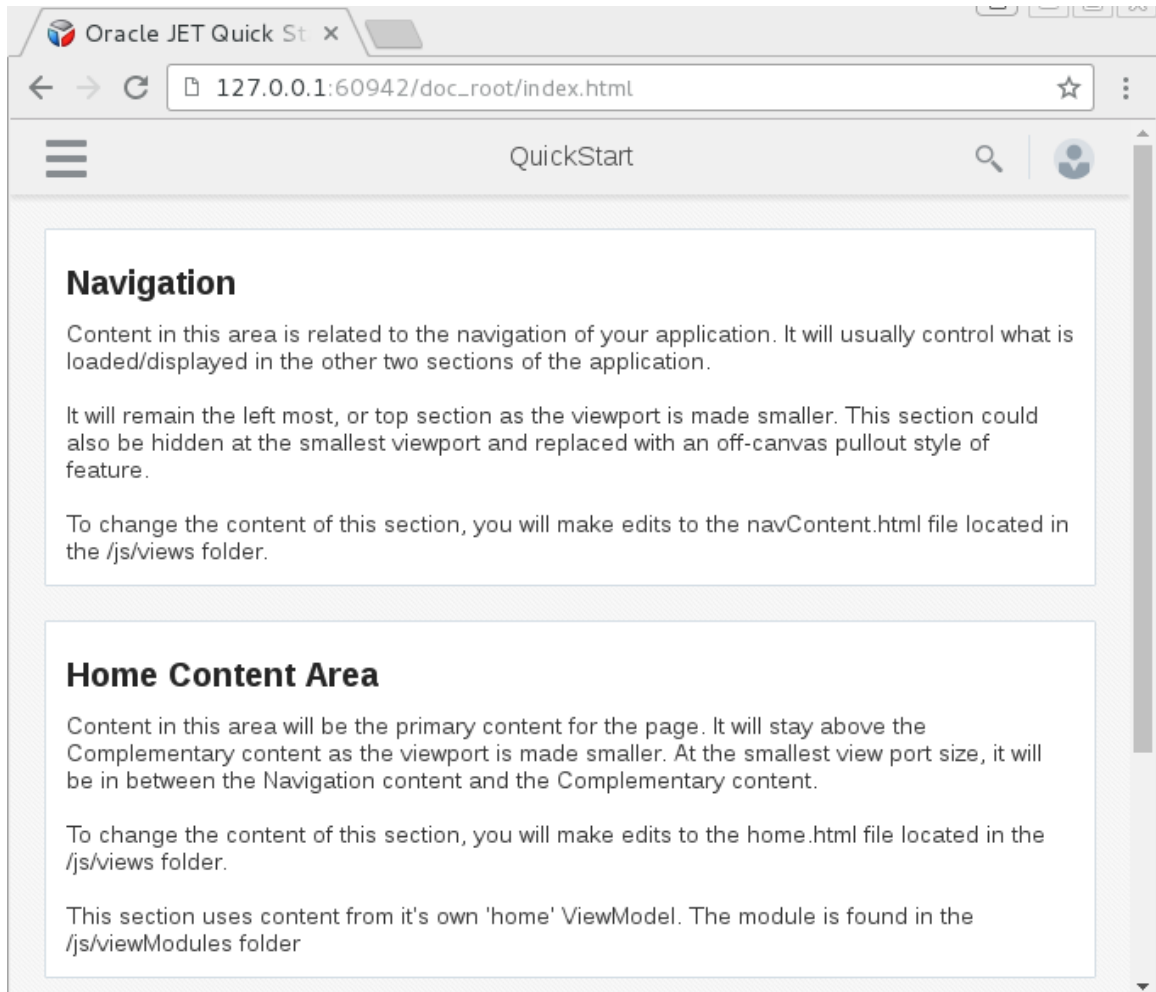


STEP 14: Run Live Preview.

- ☐ Before we make our code changes lets first run the code locally.
- ☐ Expand **doc_root** and select **index.html**



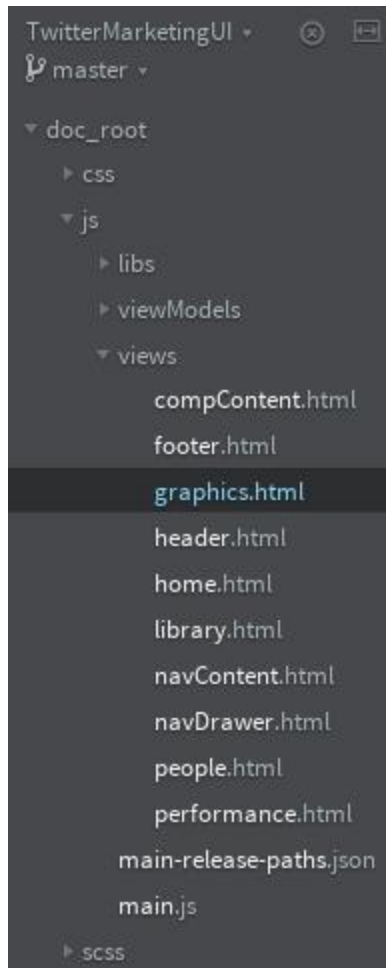
- ☐ On right hand panel, click  Live Preview. This will start your JavaScript application in a browser. Once you verify the application is working you can close the browser.



Add Code to display Twitter Feed in Table Format

STEP 15: Modify graphics.html

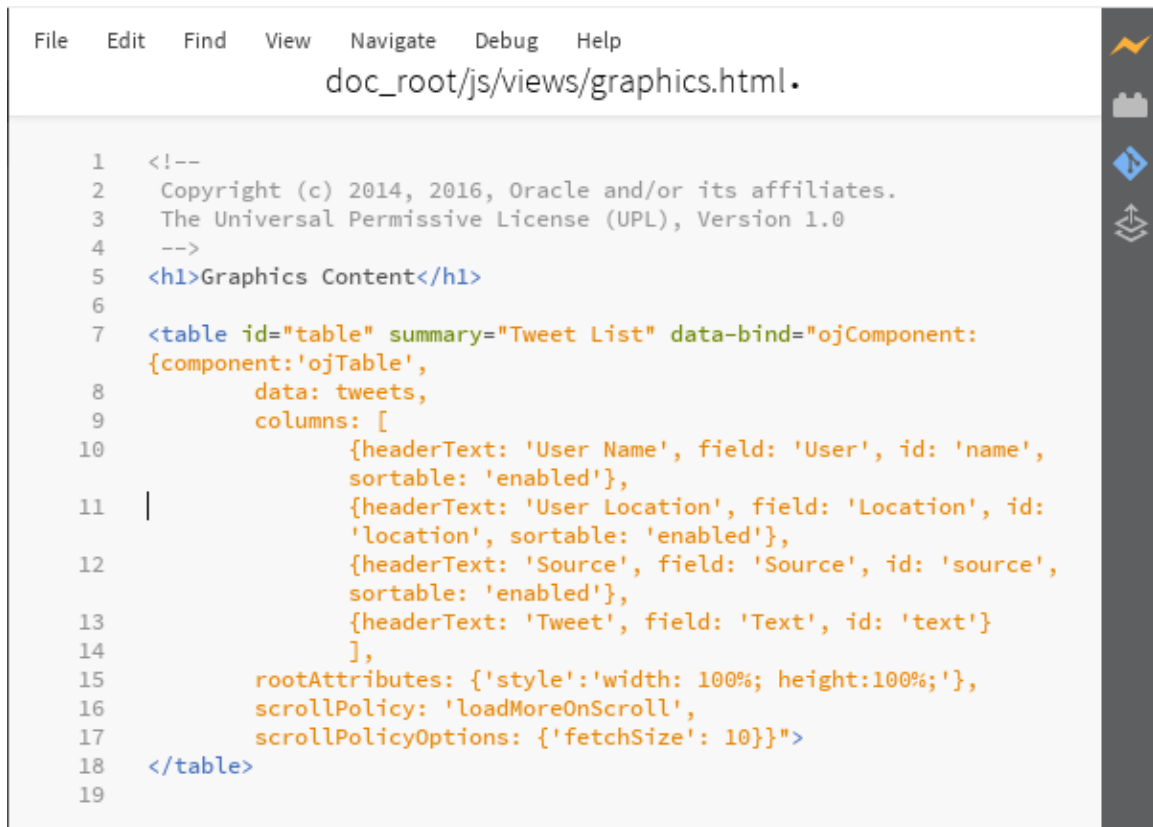
- ☐ Expand **doc_root** -> **js** -> **views** and click **graphics.html**.



❑ 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>
```

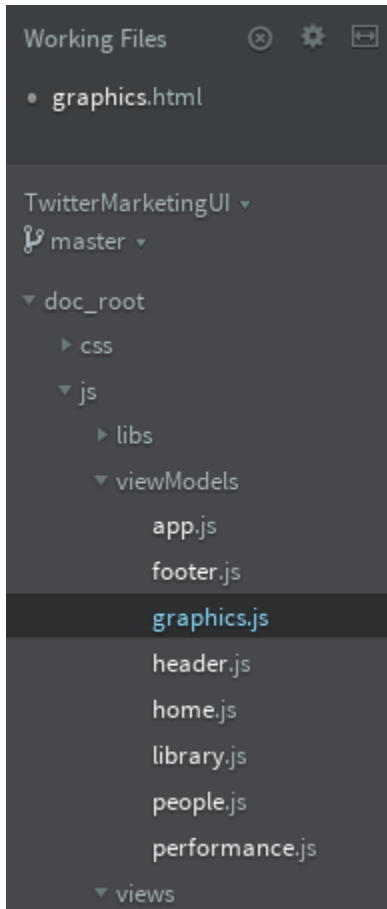


```
File Edit Find View Navigate Debug Help
doc_root/js/views/graphics.html.

1  <!--
2  Copyright (c) 2014, 2016, Oracle and/or its affiliates.
3  The Universal Permissive License (UPL), Version 1.0
4  -->
5  <h1>Graphics Content</h1>
6
7  <table id="table" summary="Tweet List" data-bind="ojComponent:
  {component:'ojTable',
8    data: tweets,
9    columns: [
10      {headerText: 'User Name', field: 'User', id: 'name',
      sortable: 'enabled'},
11      {headerText: 'User Location', field: 'Location', id:
      'location', sortable: 'enabled'},
12      {headerText: 'Source', field: 'Source', id: 'source',
      sortable: 'enabled'},
13      {headerText: 'Tweet', field: 'Text', id: 'text'}
14    ],
15    rootAttributes: {'style':'width: 100%; height:100%;'},
16    scrollPolicy: 'loadMoreOnScroll',
17    scrollPolicyOptions: {'fetchSize': 10}}">
18  </table>
19
```

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() {

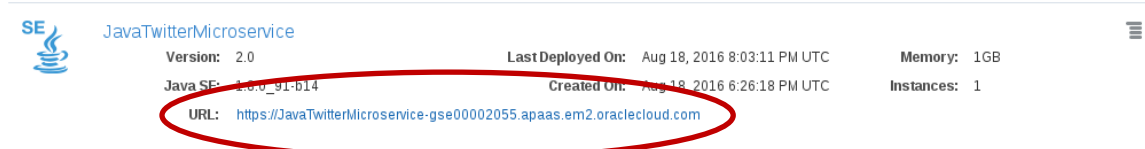
        // change this root variable to point to YOUR environment
        var root = 'https://javatwittermicroservice-
metcsgse00210.apaas.em2.oraclecloud.com/',
            self = this,
            uri = 'statictweets/',
            prettySource = function (source) {
                return source.substring(source.indexOf('>')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

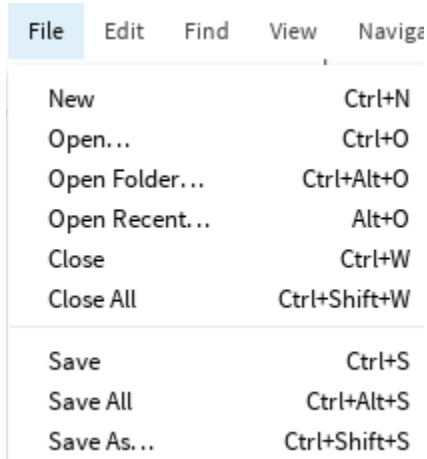
doc_root/js/viewModels/graphics.js.





```
1  /**
2   * Copyright (c) 2014, 2016, Oracle and/or its affiliates.
3   * The Universal Permissive License (UPL), Version 1.0
4   */
5  /**
6   * Your viewModel code goes here
7   */
8
9  /*global $, define, console*/
10 /*jslint sloppy:true*/
11
12 define(['ojs/ojcore', 'knockout', 'ojs/ojtable'], function (oj, ko)
13 {
14     function mainContentViewModel() {
15
16         // change this root variable to point to YOUR environment
17         var root = 'https://javatwittermicroservice-
18             gse00002055.apaas.em2.oraclecloud.com/',
19             self = this,
20             uri = 'statictweets/',
21             prettySource = function (source) {
22                 return source.substring(source.indexOf('>') + 1,
23                     source.lastIndexOf('<'));
24             },
25             url = root + uri;
26
27         self.items = ko.observableArray([]);
28         self.tweets = new oj.ArrayTableDataSource(self.items, {
29             idAttribute: 'Id'
30         });
31
32         $.ajax({
```

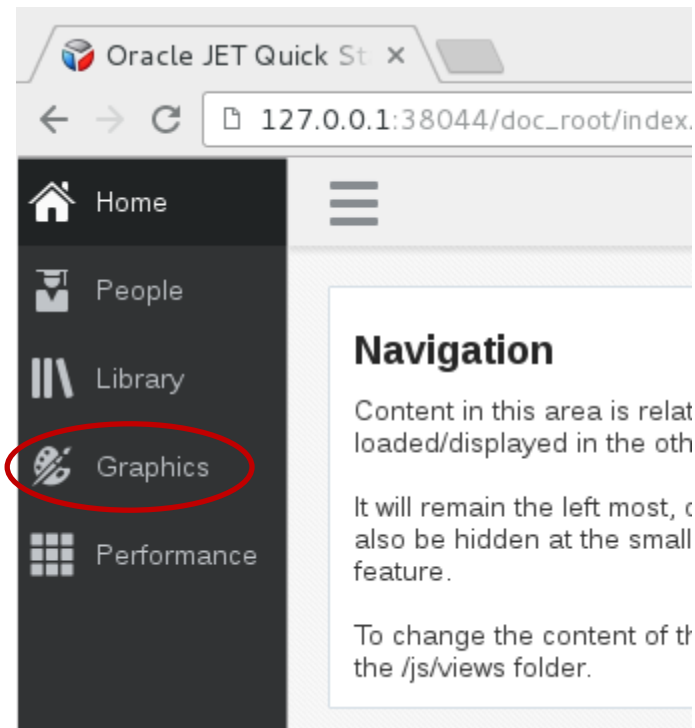
Line 17, Column 91 — 53 Lines INS JavaScript ✓ Spaces: 4

- ☐ 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:

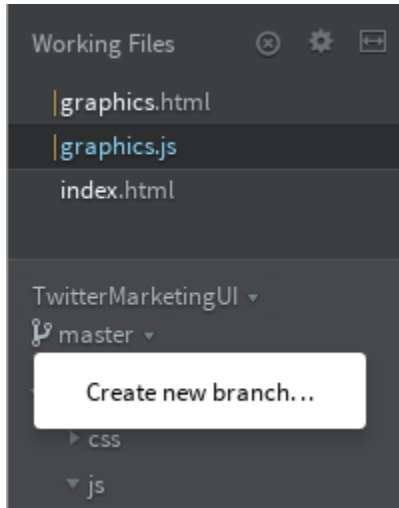
Graphics Content			
User Name	User Location	Source	Tweet
Nick Courtney	Portsmouth, UK	Facebook	The V
LizQuenlurve	Republic of the Philippines	TweetDeck	#Dolci
Jug		Twitter for iPhone	@Coc
Amethyst		Twitter for Android	RT @
Sharon ♥JB	Florida, USA	Twitter for Android	@kidr
Katie	England, United Kingdom	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

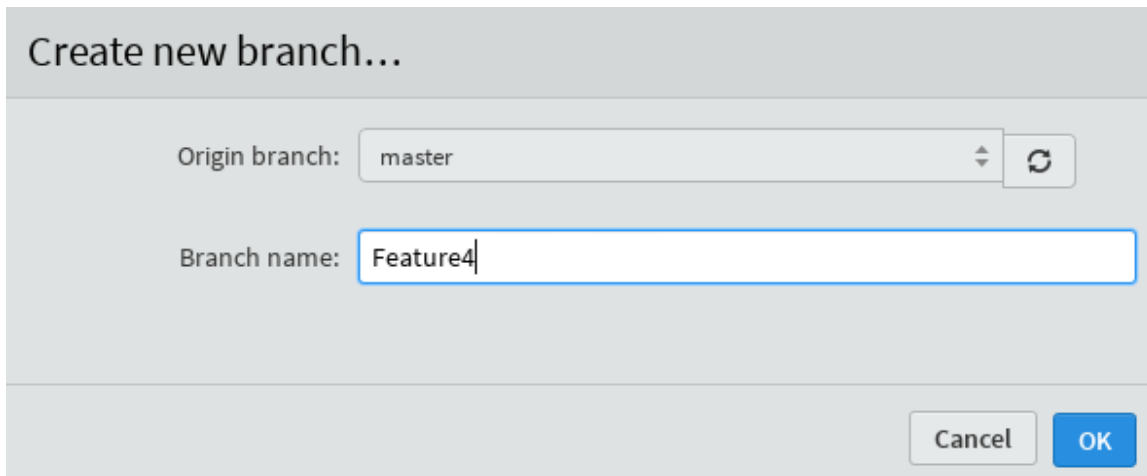
Create a Branch and Commit Code


STEP 18: Create a new Branch and Commit Code

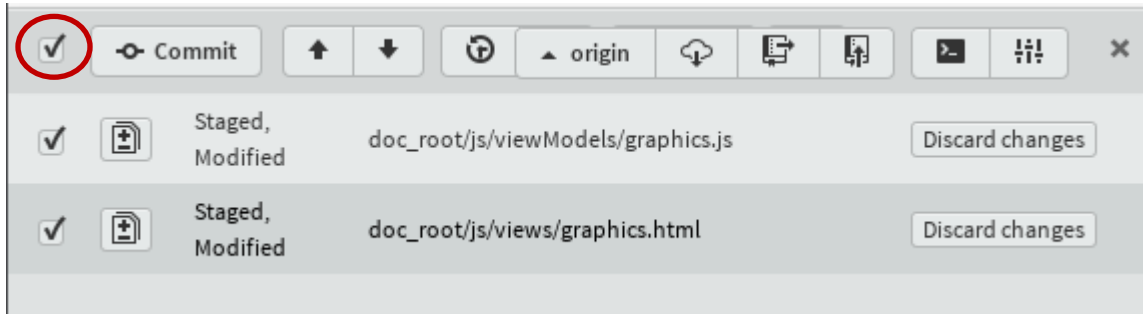
- ☐ First we need to create a new branch to check in all of our changes for this feature. In the left hand navigation panel, select **master** and click **Create new branch**.



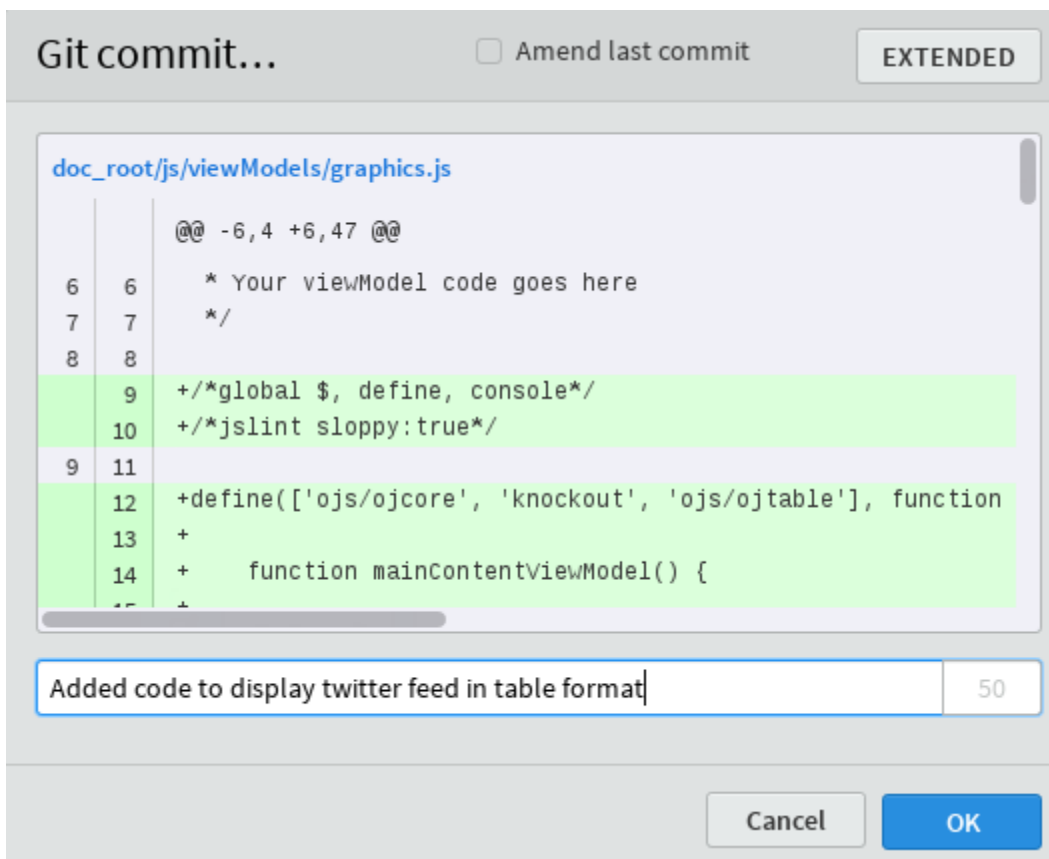
- ☐ In popup window, enter **Feature4** for branch name and click **OK**.




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

Push to remote — origin

Current tracking branch: none - "Feature4" branch will be created on remote

Action: ☒ Push to current tracking branch
☐ Push to another branch

☒ Default push
☐ Forced push
☐ Delete remote branch

☐ Send tags

You don't need to fill out username/password if your credentials are managed elsewhere. Use this only when your operation keeps timing out.

Username:

Password:

☐ Save credentials to remote url (in plain text)

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

Git Push response

Successfully pushed new ref

Info:

Remote url - https://john.dunbar:***@developer.em2.oraclecloud.com/developer99019-gse00002055/s/developer99019-gse00002055_twitter-feed-marketing-project_1284/scm/TwitterMarketingUIMicroservice.git

Local branch - refs/heads/Feature4

Remote branch - refs/heads/Feature4

Summary - [new branch]

Status - Branch Feature4 set up to track remote branch Feature4 from origin.

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 Progress (1) 2

Verify Code (0) 0

Feature 4

Display Twitter Feed in Table Format

  2

ASSIGNED

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

Change Progress

Change:

Feature 4

From:

ASSIGNED

To:

RESOLVED - FIXED

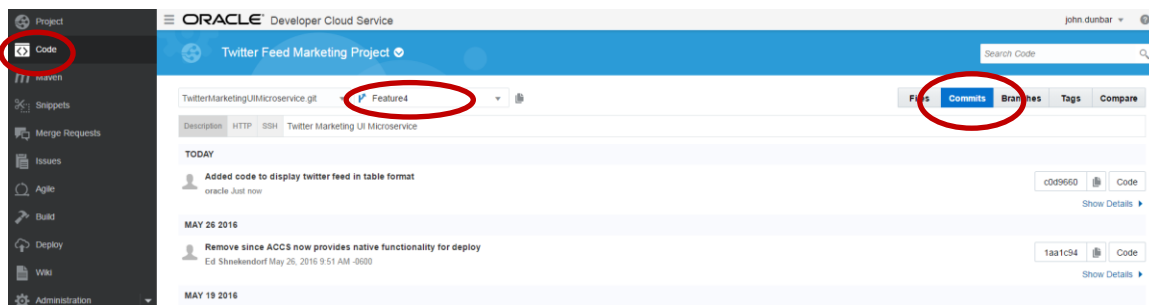
OK

Cancel

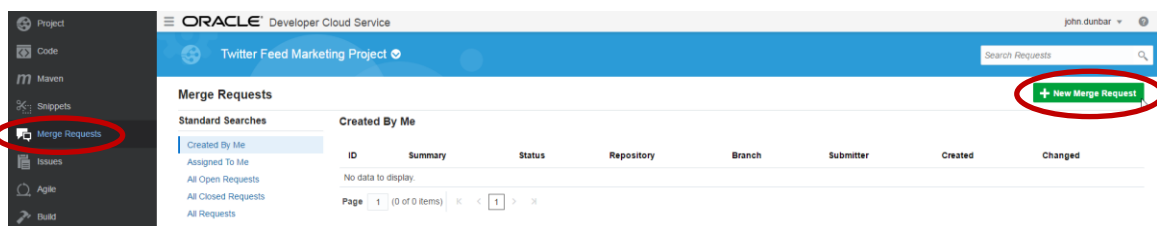
Create Merge Request

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`

The screenshot shows the 'New Merge Request' dialog with the 'Branch' tab selected. The progress bar has three steps: 'Branch' (active), 'Details', and 'Description'. The 'Next' button is highlighted. The form fields are as follows:

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

Below the fields, there is a commit preview showing a user icon, the text 'oracle Today', and a commit message 'c0d9660 Added code to display twitter feed in table format'.

- ❑ Enter the following information into **Details** and click **Create**

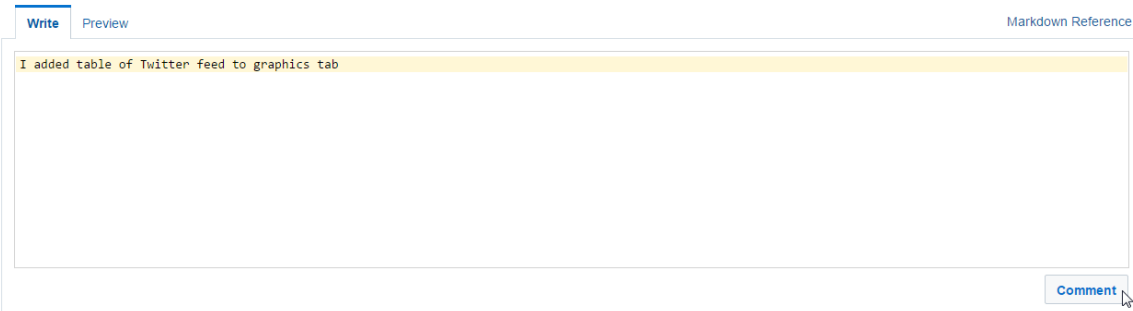
Summary: `Merge Feature 4 into master`
Reviewers: `Lisa Jones` (or current user in non-multi user env)

The screenshot shows the 'New Merge Request' dialog with the 'Details' tab selected. The progress bar has three steps: 'Branch', 'Details' (active), and 'Description'. The 'Next' button is disabled. The form fields are as follows:

- Linked Issues: `Search and Link Issues`
- Linked Builds: `Search and Link Build Jobs`
- Summary: `Merge Feature 4 into master`
- * Reviewers: `Lisa Jones` (with a close button 'X')

At the bottom right, there are two buttons: a green '+ Create' button and a grey 'Cancel' button.

- ☐ 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"



The screenshot shows a web interface for writing a comment. At the top, there are two tabs: 'Write' (selected) and 'Preview'. To the right of the tabs is a link for 'Markdown Reference'. Below the tabs is a large text area containing the text 'I added table of Twitter feed to graphics tab'. At the bottom right of the text area is a button labeled 'Comment'.

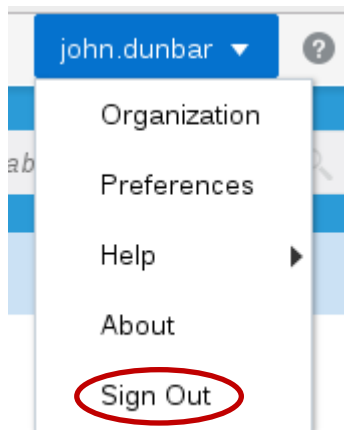
Merge the Branch as Lisa Jones

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

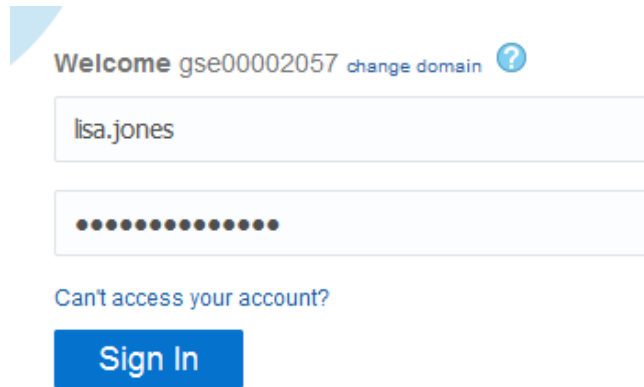
NOTE: If you are using a single user environment, you will skip the next step, and go to the following step titled: "Merge Requests"

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**.



Welcome gse00002057 [change domain](#) [?](#)

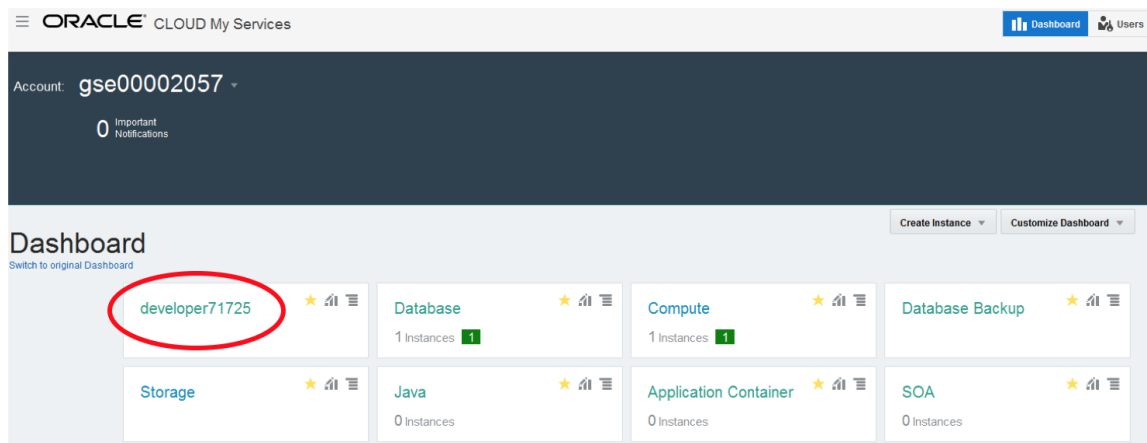
lisa.jones

.....

[Can't access your account?](#)

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

Service Details: developer71725 (Oracle Developer Cloud Service)

Open Service Console

Overview
(for August 2016)
100% uptime

Business Metrics
(as of 10 hours 53 minutes ago)
0 current disk usage

Service Status - August 2016
Month View | Quarterly View | Year View | Current Month

Legend: Before Activation, Service Up, Planned Outage, Service Incident

Additional Information

Plan: Trial Developer Service
Service Start Date: 26. Juli 2016
Service End Date: Not available
Subscription ID: 536840104
Customer Account: gse00002057 (US)
CSI Number: Not available

Data Center: EMEA Commercial 2 - Amsterdam
Version: 16.1.0.0.0
Status: Active
Service Instance URL: <https://developer.em2.oraclecloud.com>

- ☐ Select the **Twitter Feed Marketing Project**

Member Favorites Owner All

+ New Project

Filter Projects

Twitter Feed Marketing Project
Project to gather and analyze twitter data

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.

ORACLE Developer Cloud Service

Twitter Feed Marketing Project

Merge Requests

Standard Searches: Created By Me, Assigned To Me, All Open Requests, All Closed Requests, All Requests



Assigned To Me

ID	Summary	Status
6	Merge Feature 4 into master	OPEN

Page 1 of 1 (1 of 1 items)

- ☐ 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.

< | Merge Feature 4 into master 

OPEN John Dunbar wants to merge 1+ commits to  master from  Feature4

Conversation Commits (1+) **Changed Files (2)** Linked Issues

+43 graphics.js doc_root/js/viewModels

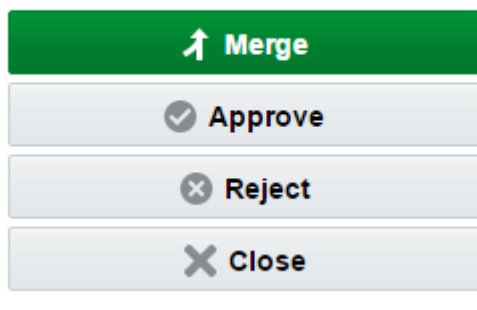
+43 graphics.js doc_root/js/viewModels

```

...  ... @@ -6,4 +6,47 @@
6    6    * Your viewModel code goes here
7    7    */
8    8
9    9    /*global $, define, console*/
10   10   /*jslint sloppy:true*/
11
12   12   define(['ojs/ojcore', 'knockout', 'ojs/ojtable'], function (oj, ko) {
13
14       function mainContentViewModel() {
15
16           // change this root variable to point to YOUR environment
17           var root = 'https://javatwittermicroservice-gse00002055.apaas.em2.oraclecloud.com/',
18               self = this,
19               uri = 'statictweets/',
20               prettySource = function (source) {
21                   return source.substring(source.indexOf('>') + 1, source.lastIndexOf('<'));
22               },
23               url = root + uri;
24
25           self.items = ko.observableArray([]);
26           self.tweets = new oj.ArrayTableDataSource(self.items, {

```

- ☐ Click on the **Merge** button.



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

Merge

The review branch 'Feature4' will be merged into the target branch 'master'.

☐ Squash commits

☐ Delete branch

Commit Message

Merge branch 'Feature4' into 'master'.

Merge

Cancel

- ☐ 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

Jobs Overview

Build Queue

Twitter Marketing UI Build

18%

View Build History

+ New Job

Status	Weather	Job	Last Success
		Twitter Feed Build	Today at 6:04 PM -0400
		Twitter Marketing UI Build	Today at 9:14 PM -0400

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

Jobs Overview

Build Queue			
View Build History			
+ New Job			
Status	Weather	Job	Last Success
✓	☀	Twitter Feed Build	Today at 6:04 PM -0400
✓	☀	Twitter Marketing UI Build	Just now

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

JETFrontEndApp

Deploy to ACCS [em2 / gse00002055 / john.dunbar](#)

Configuration TwitterMarketingUIDeploy

Job / Build [Twitter Marketing UI Build](#) / Latest Successful Build

Artifact [target/jet-quickstart-client-dist.zip](#)

Deployment update in progress

JETFrontEndApp

Deploy to ACCS [em2 / gse00002055 / john.dunbar](#)

Configuration TwitterMarketingUIDeploy

Job / Build [Twitter Marketing UI Build](#) / Latest Successful Build

Artifact [target/jet-quickstart-client-dist.zip](#)

Last deployment succeeded -- Just now.

Test the JETFrontEndAPP UI in the Cloud

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

JETFrontEndApp

Deploy to ACCS [em2 / gse00002055 / john.dunbar](#)

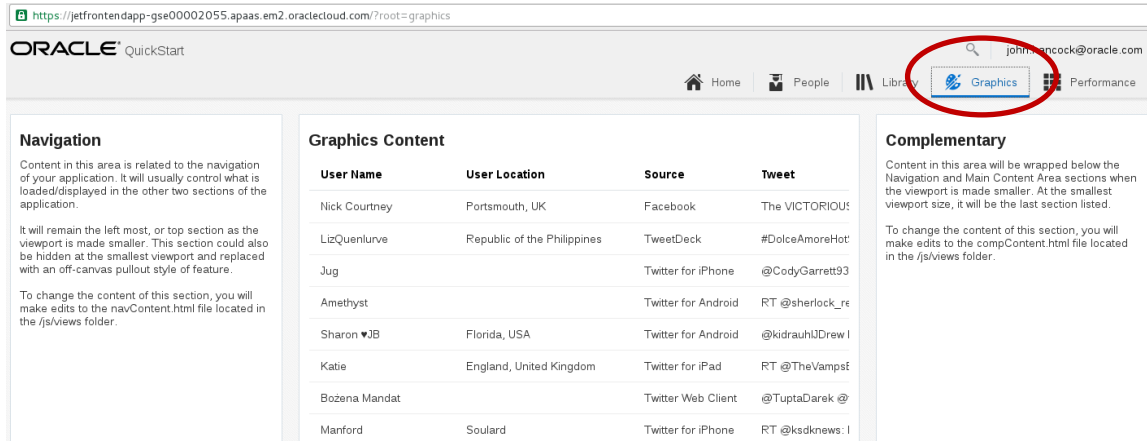
Configuration TwitterMarketingUIDeploy

Job / Build [Twitter Marketing UI Build](#) / Latest Successful Build

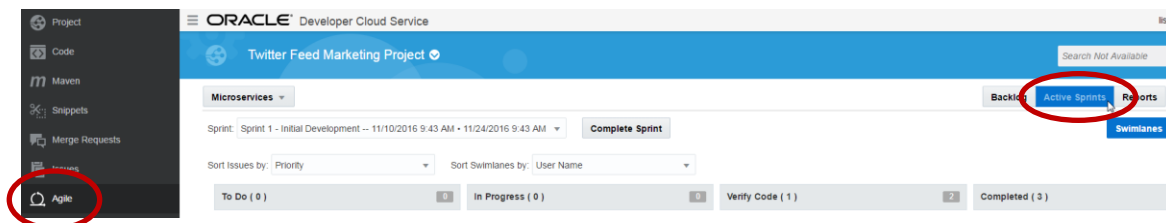
Artifact [target/jet-quickstart-client-dist.zip](#)

Last deployment succeeded -- Just now.

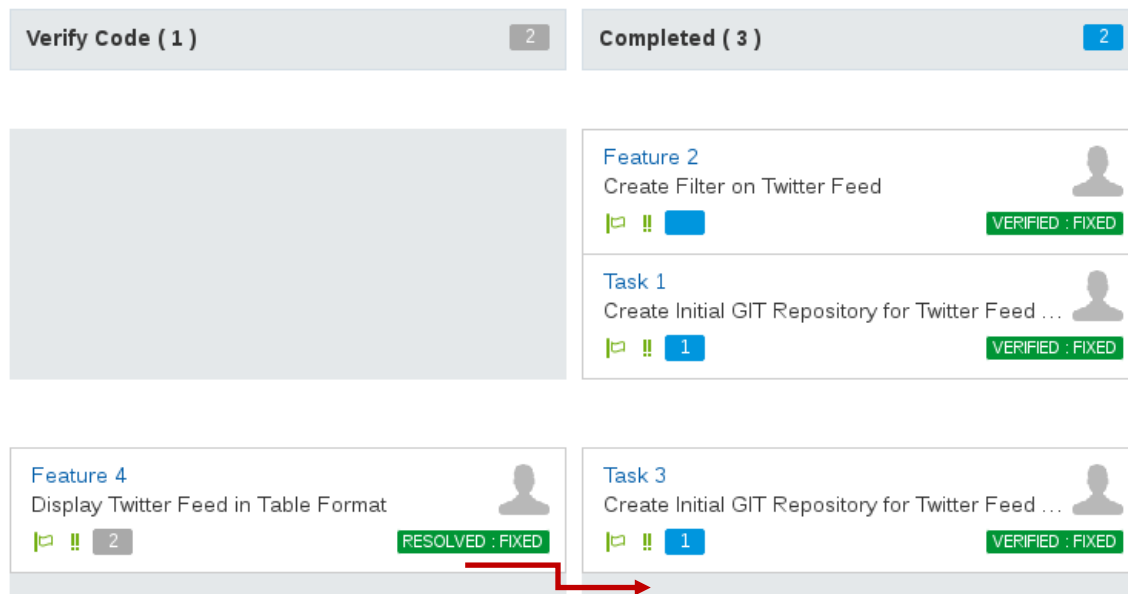
- ☐ When the new browser tab loads, click **Graphics** to display twitter feed data.



- ☐ 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**

Change Progress ×

Change:

Feature 4

From:

RESOLVED - FIXED

To:

VERIFIED - FIXED ▼

OK

Cancel

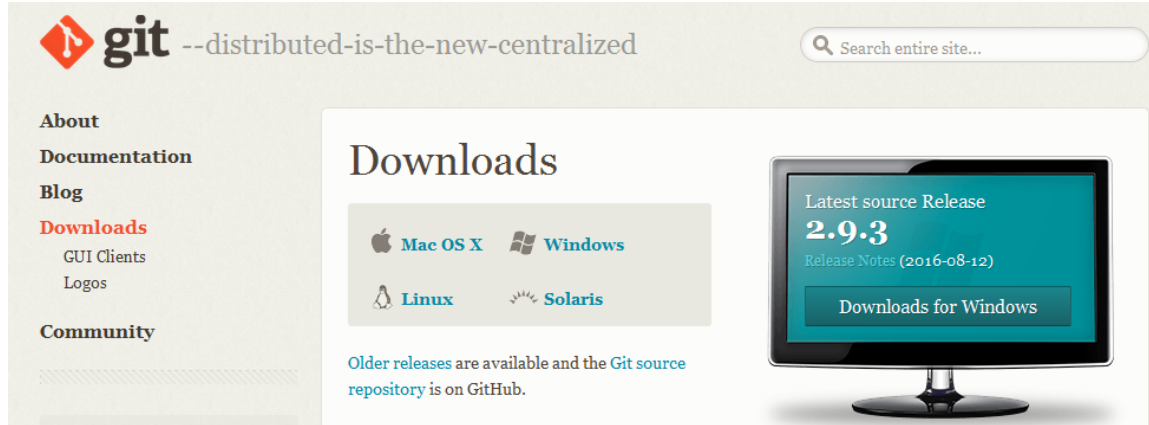
- ☐ You are now ready to move to the next lab.

Appendix 1 – Installing Brackets and Git

Download and Install Git

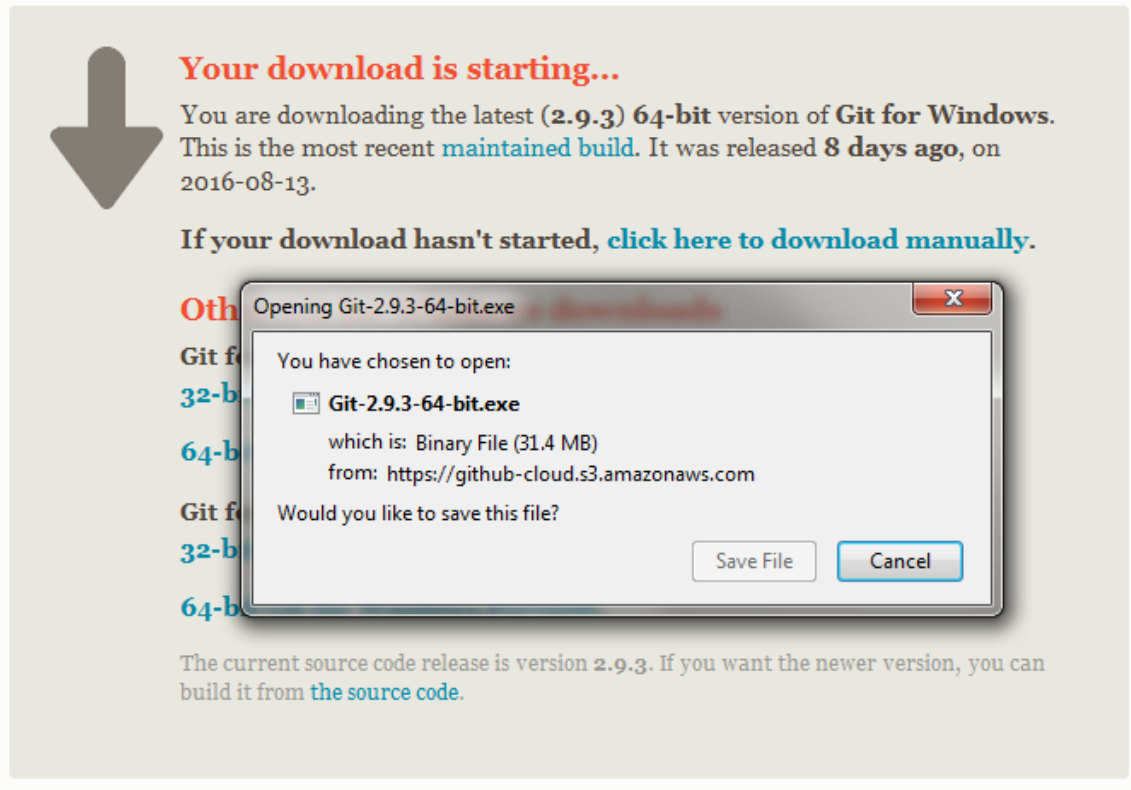
STEP 23: Download Git

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

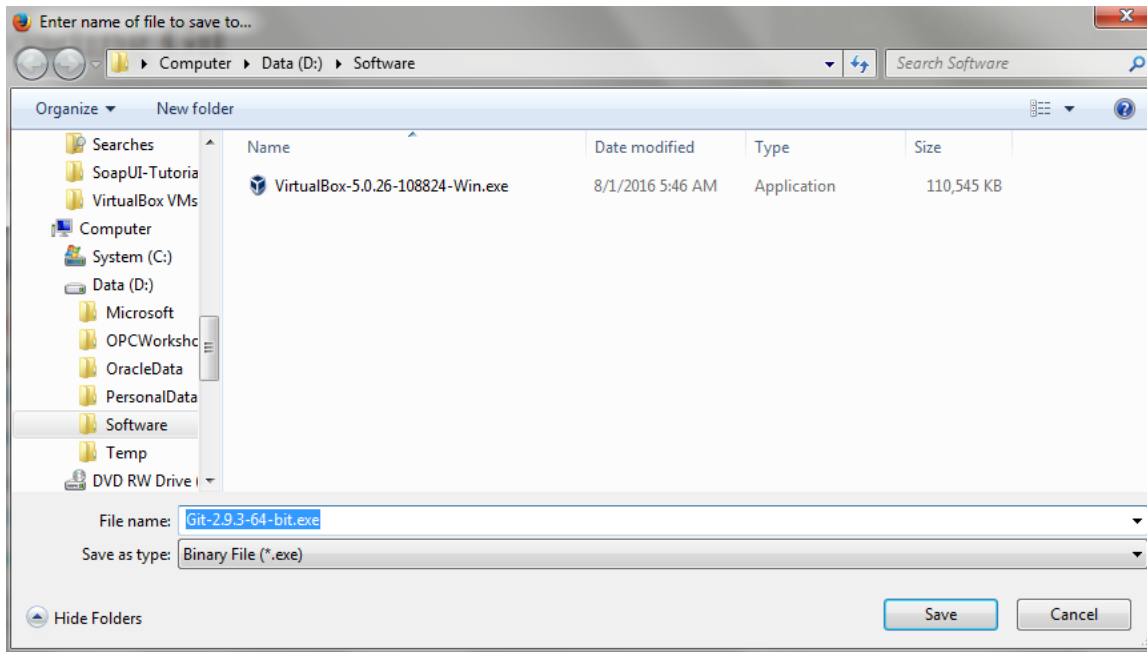


- ❑ Select your OS. In our example we will show how to install on Windows. Click **Windows** Download and click **Save File**

Downloading Git



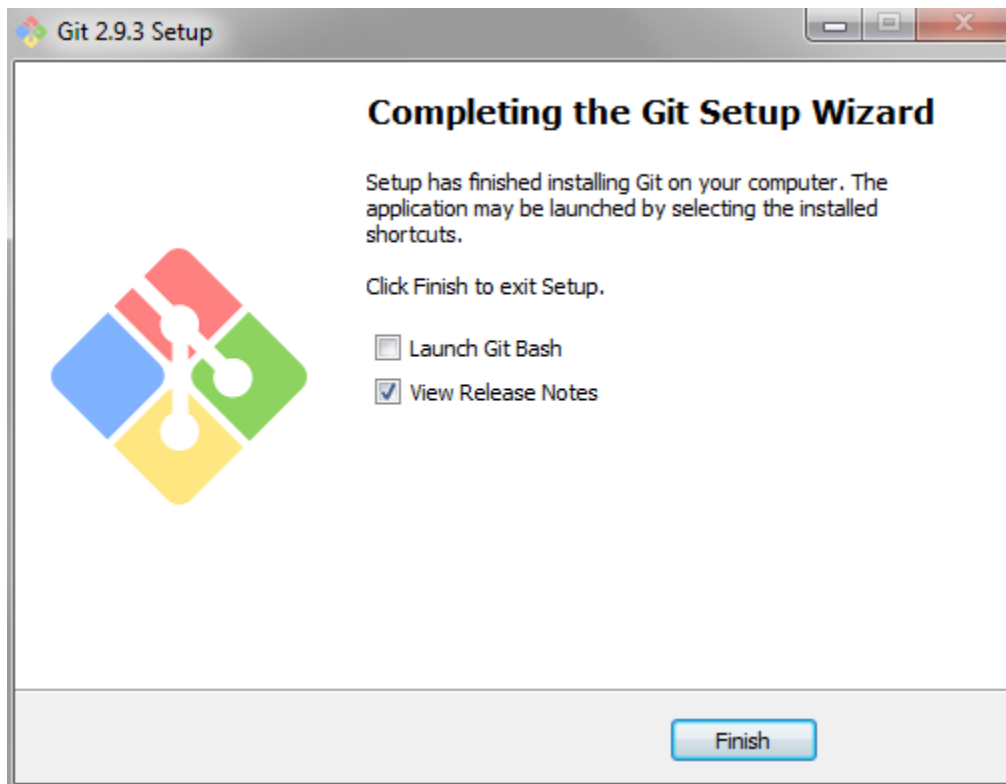
- ☐ Select your download location and click **Save**. We will use D:\Software



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

Name	Date modified	Type	Size
Git-2.9.3-64-bit.exe	8/21/2016 7:06 AM	Application	32,120 KB
VirtualBox-5.0.26-108824-Win.exe	8/1/2016 5:46 AM	Application	110,545 KB

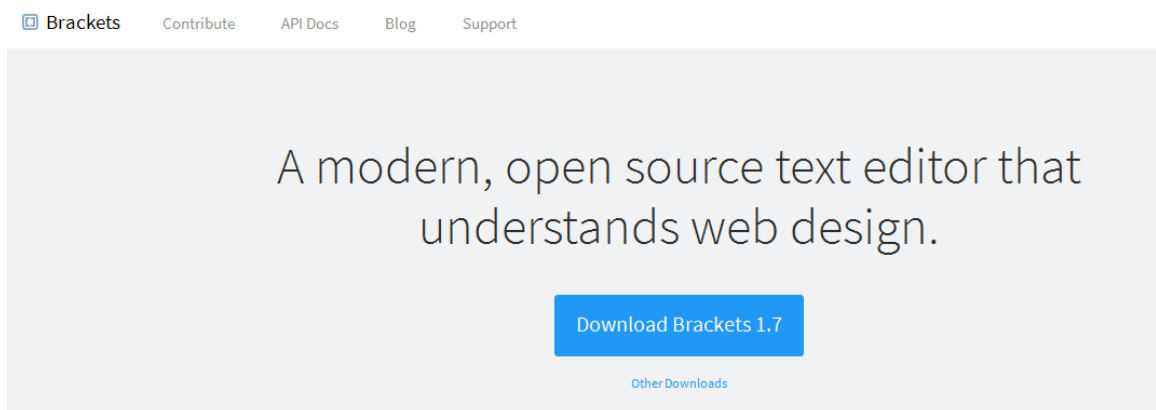
- ☐ Run through the installation process



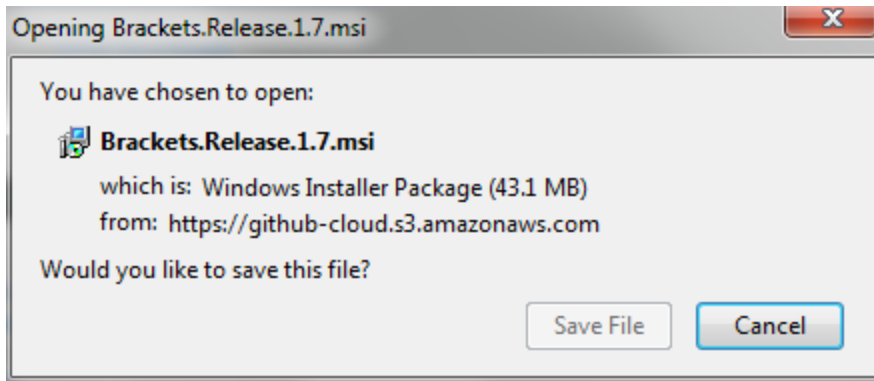
Download and Install Brackets Text Editor

STEP 24: Download Brackets Text Editor

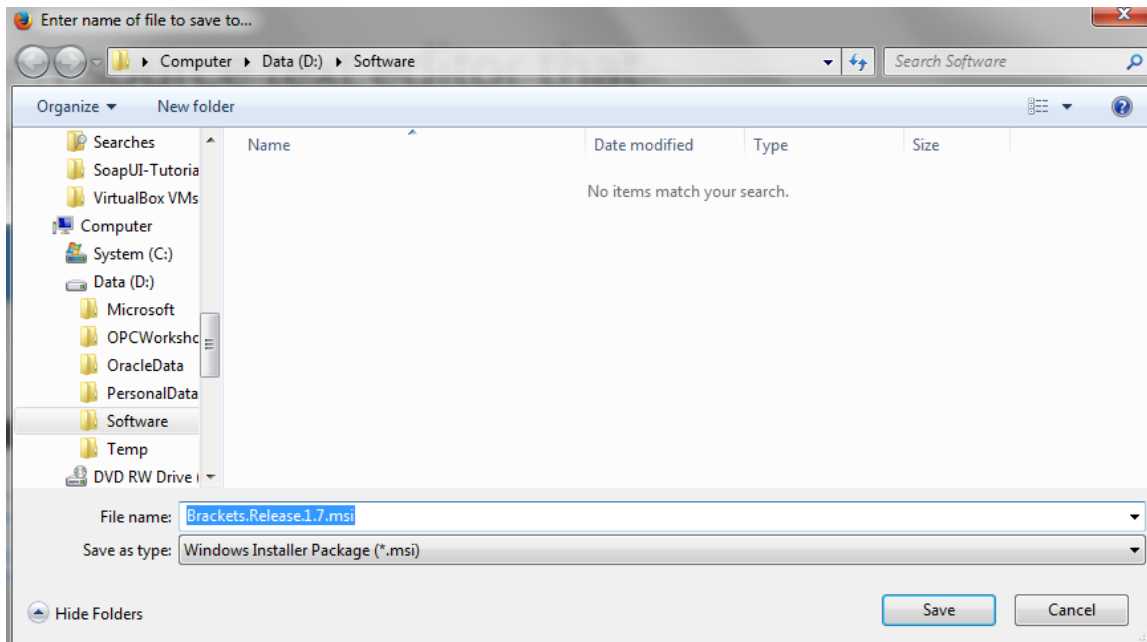
- ☐ Go to the following URL: <http://brackets.io>



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



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



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

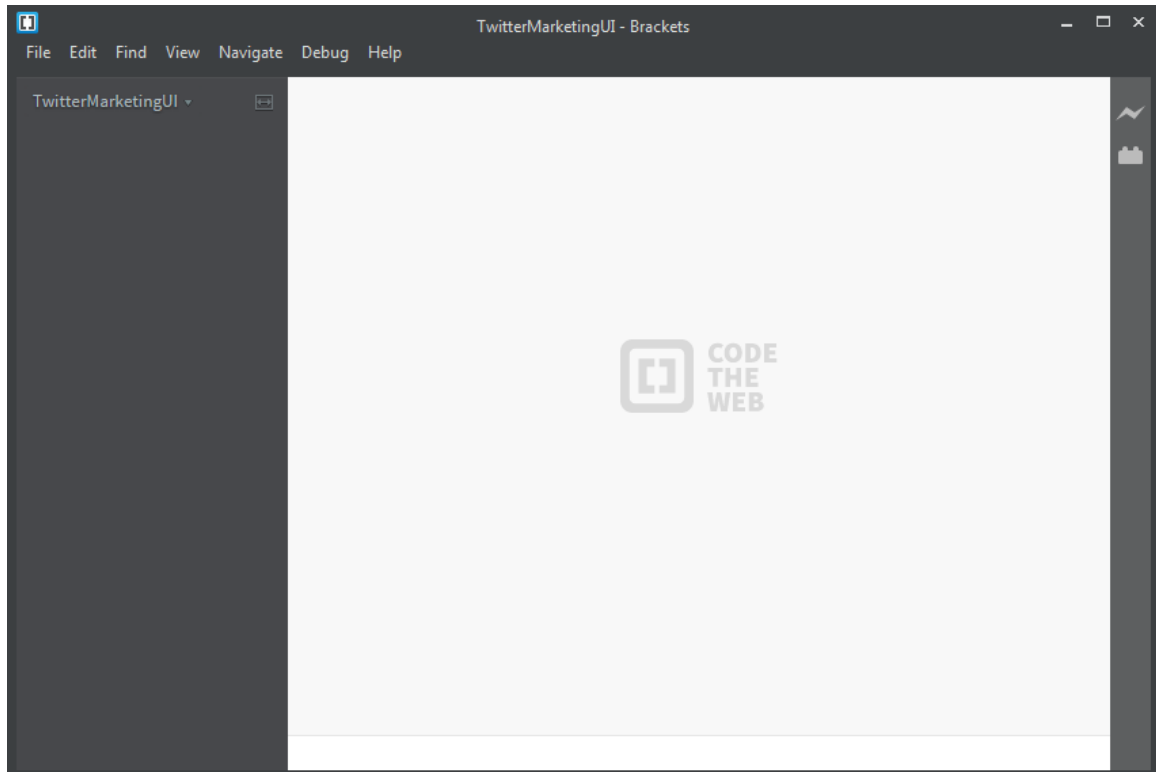
Name	Date modified	Type	Size
Brackets.Release.1.7.msi	8/21/2016 7:44 AM	Windows Installer ...	44,184 KB
Git-2.9.3-64-bit.exe	8/21/2016 7:06 AM	Application	32,120 KB
VirtualBox-5.0.26-108824-Win.exe	8/1/2016 5:46 AM	Application	110,545 KB

- ☐ Run through the installation process

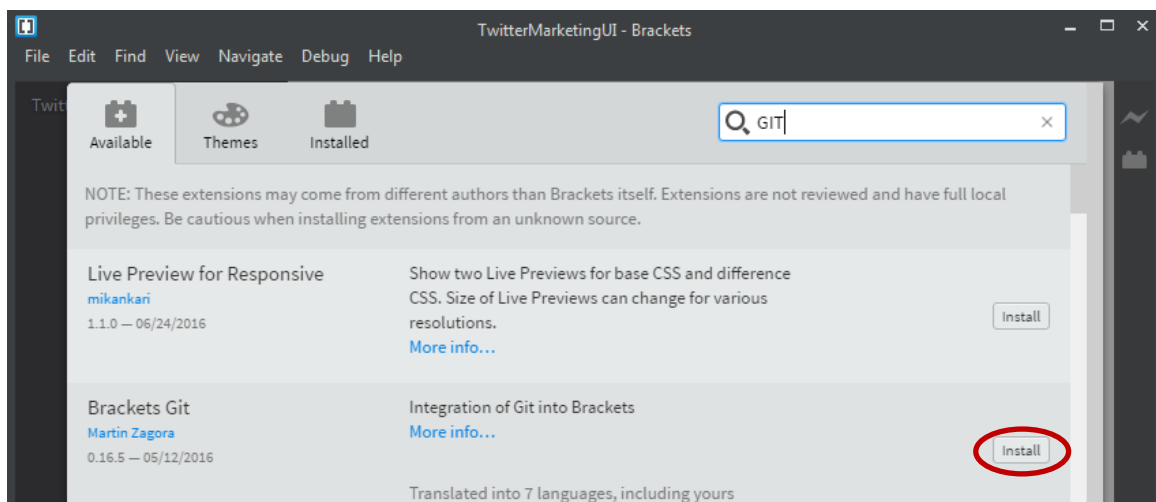
Start Brackets and Configure Git

STEP 25: Start Brackets and Configure Git

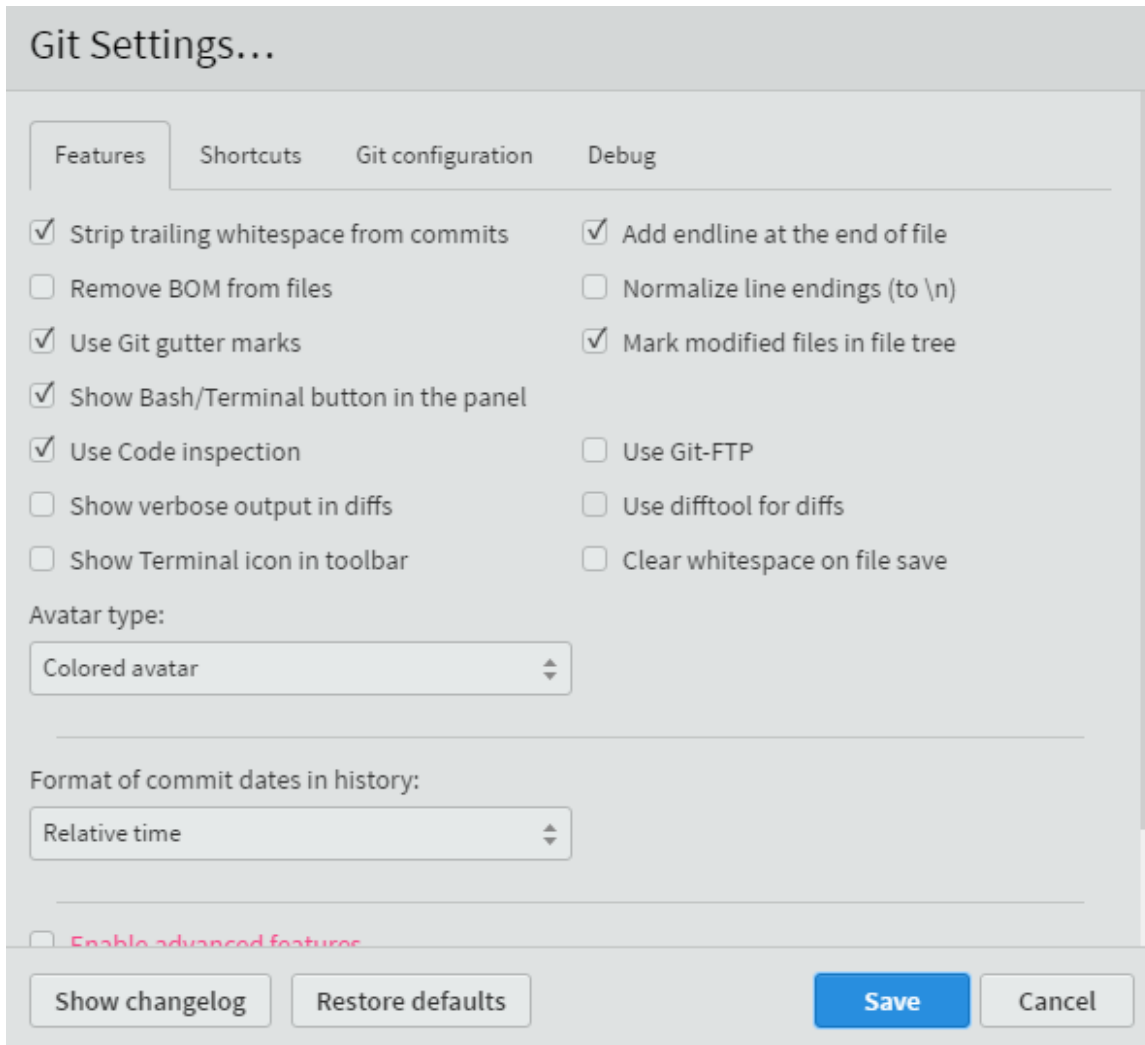
- ☐ Create directory **TwitterMarketingUI**. From Windows Explorer navigate to the directory **TwitterMarketingUI**, right click and select **Open as Brackets Project**



- ☐ 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**

The image shows the 'Git Settings...' dialog box with the 'Features' tab selected. It contains two columns of checkboxes for various Git features. The 'Avatar type' is set to 'Colored avatar' and the 'Format of commit dates in history' is set to 'Relative time'. At the bottom, there are buttons for 'Show changelog', 'Restore defaults', 'Save', and 'Cancel'.

Git Settings...

Features Shortcuts Git configuration Debug

<input checked="" type="checkbox"/> Strip trailing whitespace from commits	<input checked="" type="checkbox"/> Add endline at the end of file
<input type="checkbox"/> Remove BOM from files	<input type="checkbox"/> Normalize line endings (to \n)
<input checked="" type="checkbox"/> Use Git gutter marks	<input checked="" type="checkbox"/> Mark modified files in file tree
<input checked="" type="checkbox"/> Show Bash/Terminal button in the panel	
<input checked="" type="checkbox"/> Use Code inspection	<input type="checkbox"/> Use Git-FTP
<input type="checkbox"/> Show verbose output in diffs	<input type="checkbox"/> Use difftool for diffs
<input type="checkbox"/> Show Terminal icon in toolbar	<input type="checkbox"/> Clear whitespace on file save

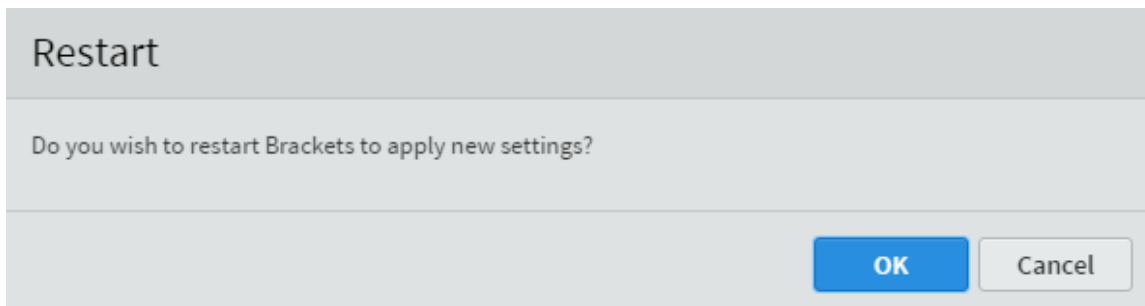
Avatar type:
Colored avatar

Format of commit dates in history:
Relative time

☐ Enable advanced features

Show changelog Restore defaults **Save** Cancel

- ☐ Click OK to restart Brackets

The image shows the 'Restart' dialog box. It asks the user if they wish to restart Brackets to apply new settings. At the bottom, there are buttons for 'OK' and 'Cancel'.

Restart

Do you wish to restart Brackets to apply new settings?

OK Cancel

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

