



300

Oracle 퍼블릭 클라우드 워크샵

Node.js 로 클라우드
네이티브한 신속한 자바
스크립트 개발

2016 년 12 월 22 일

도입

오라클 퍼블릭 클라우드 DevOps 클라우드 네이티브 마이크로 서비스 워크샵의 일부인 여러 Lab 중 세 번째 워크샵입니다. 이 워크샵에서는 여러 Microservices 를 만들고 사용하는 클라우드 네이티브 프로젝트에 대한 SDLC (Software Development Lifecycle)를 안내합니다.

이전 Lab(200)에서 Java 개발자는 트위터를 통해 데이터를 끌어 와 키워드를 기반으로 필터링 할 수 있는 동적 필터링을 기능을 추가했습니다. 이 Lab 에서는 웹 응용 프로그램의 데이터를 통합할 수 있는 웹 응용 프로그램을 생성하는 프론트 엔드 JavaScript 개발자 역할을 맡게 됩니다. 이 Node.js 애플리케이션은 자동화된 빌드를 통해 Developer Cloud Service 에서 개발되고 애플리케이션 컨테이너 클라우드 서비스에 배포됩니다.

목표

- ☐ Developer 클라우드 서비스에 접근
- ☐ 외부 Git Repository 의 코드 가져오기
- ☐ Brackets 에 Project 가져오기
- ☐ 개발자 클라우드 서비스 및 Oracle 애플리케이션 컨테이너 클라우드 서비스를 이용하여 프로젝트 build 및 배포(Deploy)

필수 요건

- ☐ 다음은 강사가 제공하는 Oracle Public 클라우드 계정입니다. 최신 버전의 Brackets 텍스트 편집기를 다운로드하여 설치해야 합니다.

개요

도입	2
목표	2
필수 요건	2
개요	3
초기 트위터 마케팅 UI 서비스 생성.....	4
Developer 클라우드 서비스 살펴보기	4
초기 Git Repository 만들기	10
기본 빌드 및 구축 프로세스 생성.....	13
Twitter Marketing UI 의 기본 배포 검증(verify)	21
기본 응용 프로그램을 확장하여 Twitter 피드 표시.....	26
Brackets 텍스트 편집기에서 Clone Project.....	27
Twitter Feed 를 표시하기 위해 코드를 추가.....	33
새 Branch and 및 Commit 코드 생성.....	41
새 Branch 와 코드 Commit.....	41
병합 요청 생성	46
Lisa Jones 로 Branch 를 Merge 하기	48
클라우드에서 JETFrontEndAPP UI 테스트	54
부록 1-Brackets 및 Git 을 설치하는 부속 문서.....	56
Git 다운로드 및 설치	56
Brackets 텍스트 편집기 다운로드 및 설치	59
Brackets 시작 및 Git 구성	61

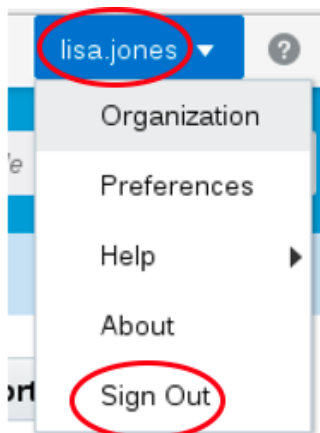
초기 트위터 마케팅 UI 서비스 생성

Developer 클라우드 서비스 살펴보기

참고: 여러 사용자(예. Lisa.Jones, Bala.Gupta, John.Dunbar)를 사용하여 이 랩을 실행하지 않는 경우 1 단계를 건너뛰고 2 단계로 이동합니다. 그러나 lab 200 을 완료한 상태인 사용자로 로그인 한 경우에도 개발자인 John Dunbar 로 로그인한 것으로 간주하셔야 합니다.

STEP 1: Oracle 클라우드 계정으로 로그인

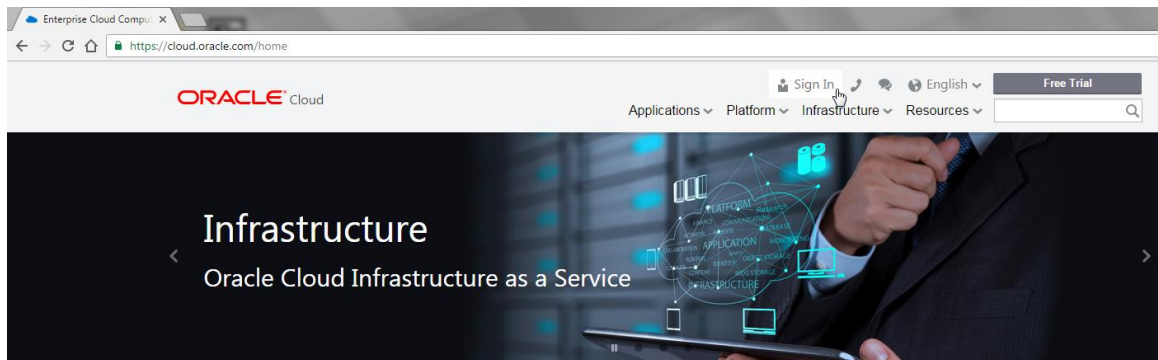
- lab 200 을 막 완성했다면, 여전히 리사 존스로 로그인 되어 있다면, 이 lab 을 계속 안내하기 전에 먼저 Sign out 을 해야 합니다. 사용자 이름(리사)을 클릭하여 로그아웃 하십시오. 화면 맨 위에 있는 오른쪽 상단 모서리에서 드롭 아웃 메뉴에서 Signout(로그아웃)을 선택합니다..



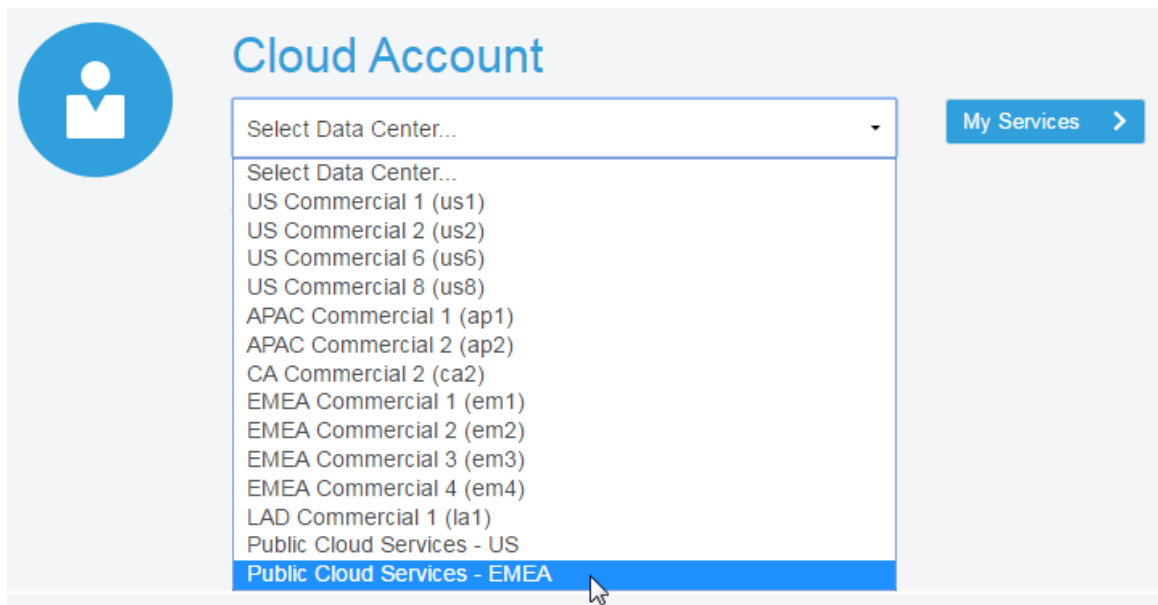
- 이제 다시 로그인할 수 있습니다. 브라우저에서 다음 URL 로 이동합니다.

<https://cloud.oracle.com>

□ 브라우저의 오른쪽 상단 모서리에 SignIn(로그인)을 클릭합니다.

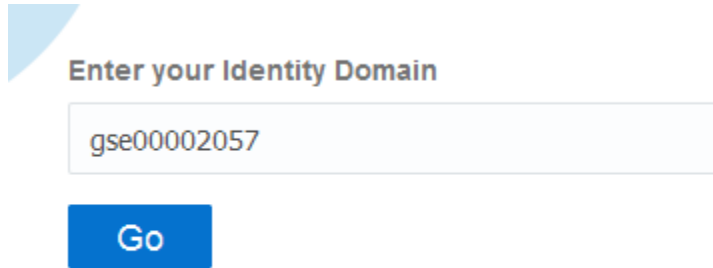


□ 중요-내 서비스에서 드롭 다운 목록에서 선택할 영역을 강사에게 요청하고 MyServices(내 서비스)버튼을 클릭합니다.



- **ID 도메인을 입력하고 Go(이동)를 클릭합니다.**

참고:강사에게 사용자 이름, 사용자 이름 및 암호 값이 부여됩니다.



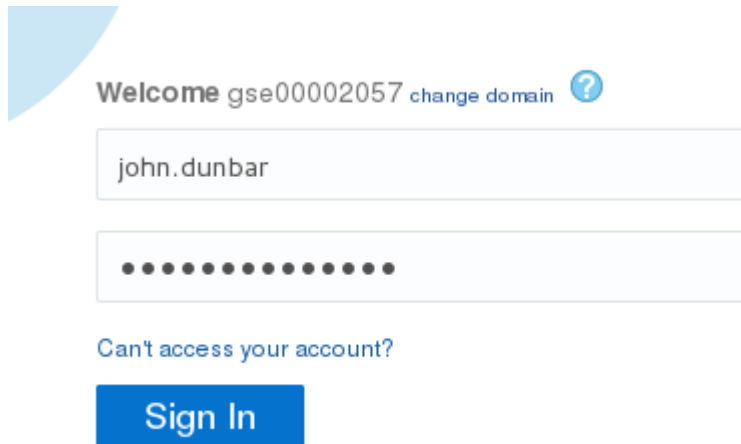
Enter your Identity Domain

gse00002057

Go

- **ID 도메인이 설정되면 사용자 이름과 암호를 입력하고 SignIn(로그인)을 클릭합니다.**

참고:이 lab에서는 자바 스크립트 개발자인 John Dunbar 역할을 수행합니다. 이전의 lab과 마찬가지로, 여러 사용자를 지원할 수 없는 경우, 지원되는 사용자로 로그인하고, 자바 스크립트 개발자라고 가정한다.



Welcome gse00002057 [change domain](#) ?

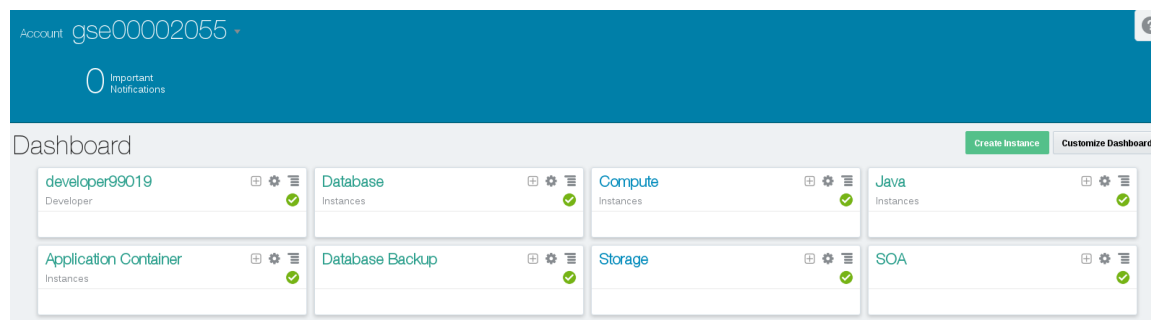
john.dunbar

.....

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

Sign In

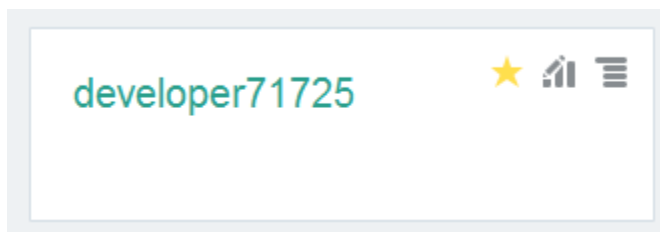
- 이 계정에는 다양한 클라우드 서비스를 제공하는 Dashboard가 표시됩니다.



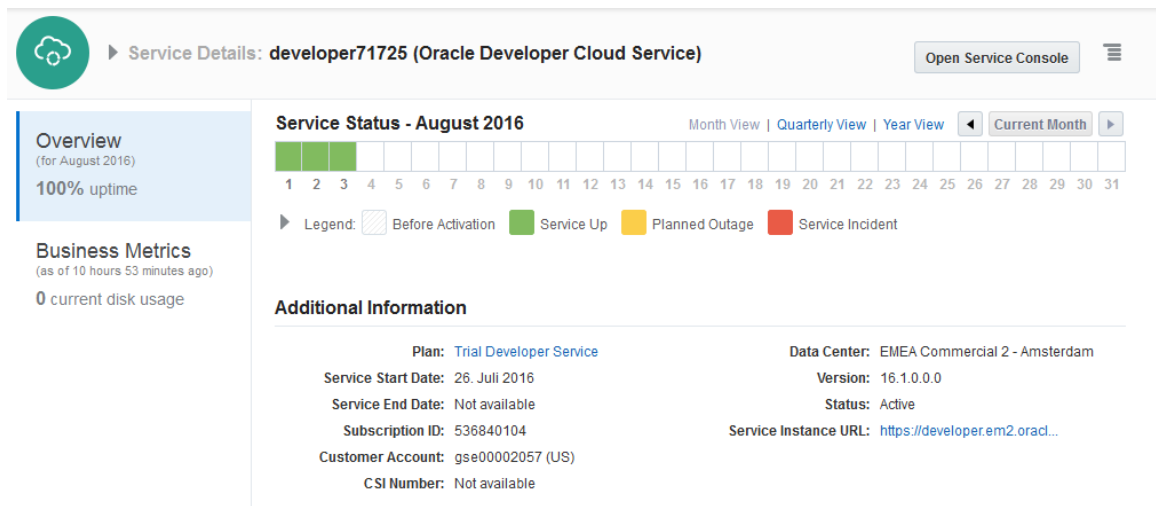
STEP 2: 개발자 클라우드 서비스에 로그인

Oracle 개발자 클라우드 서비스는 팀 개발 프로세스를 간소화하고 소프트웨어 제공을 자동화하는 완벽한 개발 플랫폼을 제공합니다. 통합 플랫폼에는 추적 시스템이 포함되어 있습니다. 민첩한 개발 대시 보드, 코드 버전 관리 및 코드 검토 플랫폼, 지속적 통합 및 공급, 자동화, 팀의 협업과 라이브 활동 스트림 등의 기능도 갖추고 있습니다. 풍부한 웹 기반 대시 보드와 인기 있는 개발 툴과의 통합을 통해 Oracle 개발자 클라우드 서비스는 보다 빠른 애플리케이션을 제공하는 데 도움이 됩니다.

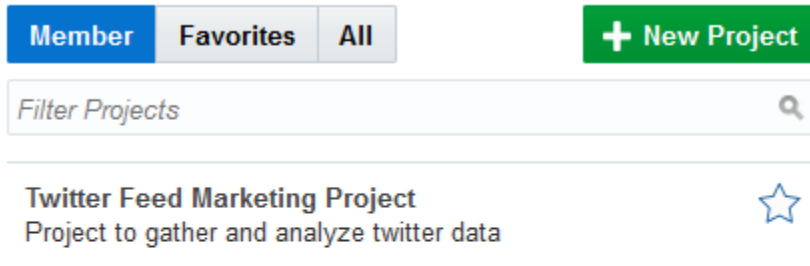
- ☐ CloudUI 대시 보드에서 개발자 서비스를 클릭합니다. 예를 들어 개발자 클라우드 서비스는 developer71725 라는 이름으로 하겠습니까.



- ☐ 서비스 세부 정보 페이지에서는 서비스 상태를 한눈에 파악할 수 있습니다.

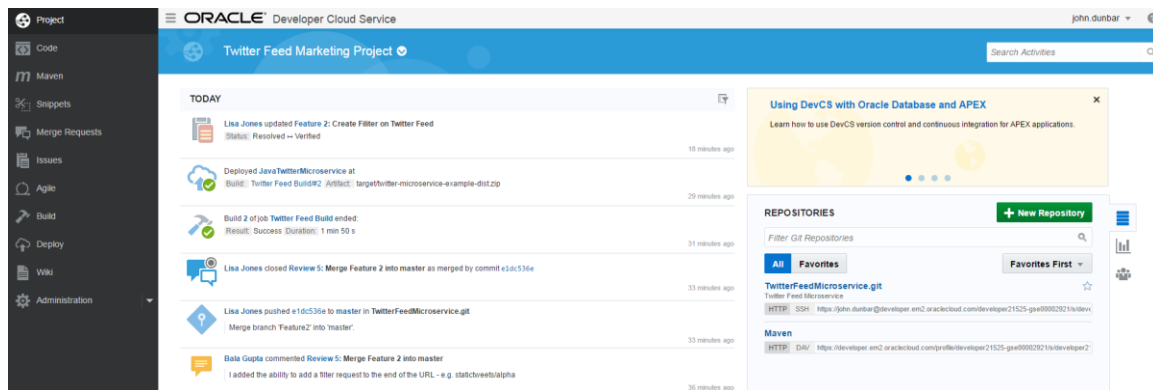


- Oracle 개발자 클라우드 서비스에 대한 **Open Service Console** 을 클릭합니다.
ServiceConsole 에서는 현재 귀하가 소속된 모든 프로젝트를 나열합니다.

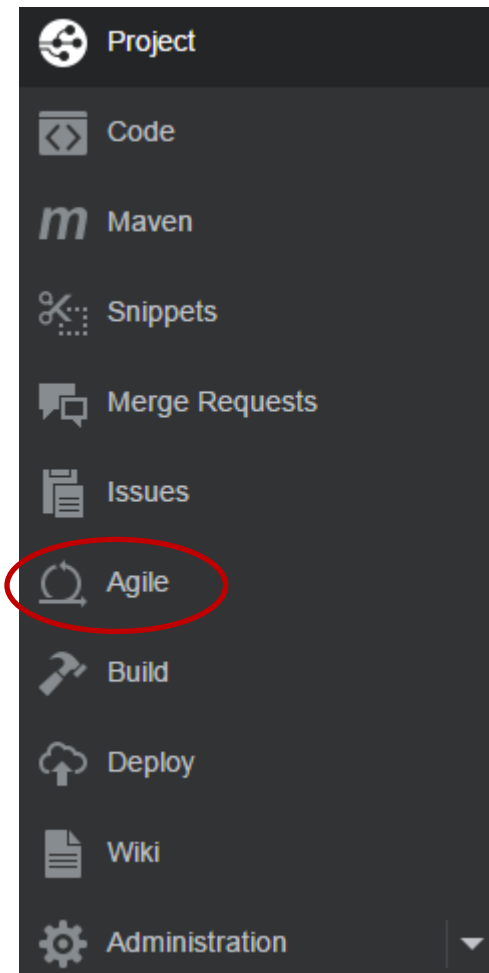


STEP 3: Agile Board 검토

- 프로젝트에 접근하려면 **Twitter Feed Marketing Project** 를 클릭하십시오.

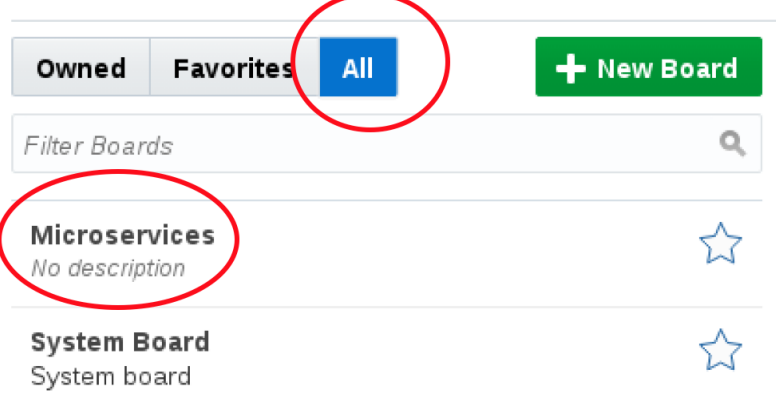


- ☐ 탐색 패널에서 **Agile** 를 클릭합니다.

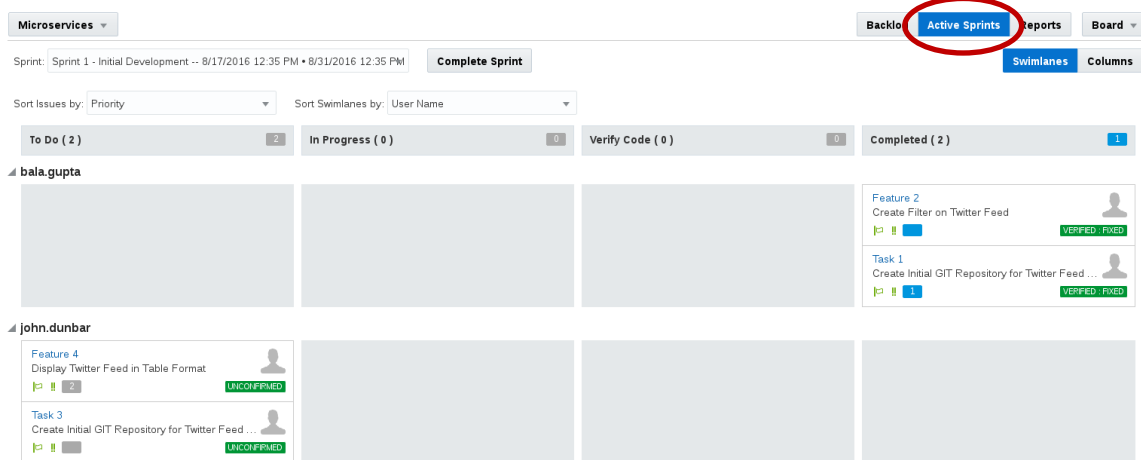


- ☐ **Microservices** 가 기본 보드가 아닌 경우 현재 보드의 드롭 다운을 클릭하고 **All** 필터를 선택한 다음 **Microservices** 를 클릭합니다.

Find Board



□ Microservices Board **Active Sprints** 를 클릭합니다.

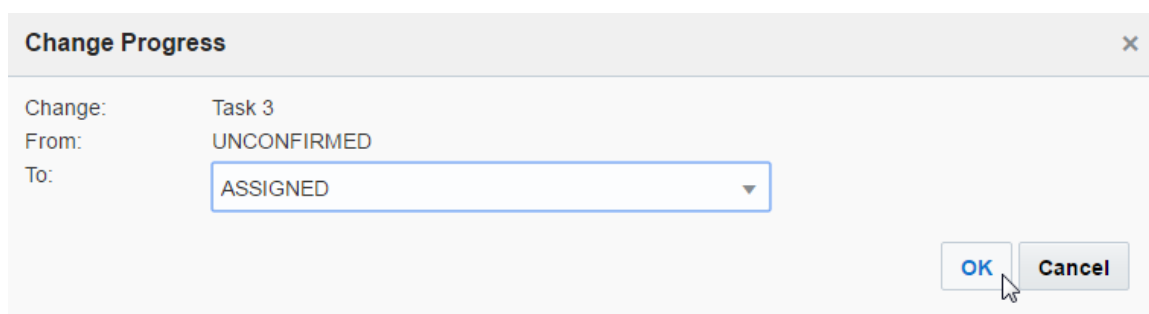


초기 Git Repository 만들기

STEP 4: 초기 Git Repository 만들기

이전 lab 에서처럼 이 시점에서 이 애플리케이션을 처음부터 다시 시작할 수 있습니다. 하지만 우리의 동료 중 한명은 이미 개발자 클라우드 서비스 외부에서 웹 애플리케이션을 위해 작업을 시작했습니다. 우리는 그의 작업을 시작점으로 삼고 우리의 트위터 microservices 를 통합하기 위해 그것을 확장하고자 한다. 그의 코드를 개발자 클라우드 서비스에 투입하기 위해, 우리는 그의 외부 GIT 저장소를 복제할 것입니다. 우선 이 작업을 진행하고 있다는 것을 보여 주기 위해 애자일 게시판을 업데이트하겠습니다.

□ **Task 3 - Create Initial GIT Repository for Twitter Marketing UI** 을 드래그 앤 드롭으로 **In Progress** 에 놓습니다. 진행 상황 확인 팝업에서 OK(확인)를 클릭합니다.



▲ john.dunbar

Feature 4 Display Twitter Feed in Table Format 2 UNCONFIRMED	Task 3 Create Initial GIT Repository for Twitter Feed Market... 1 ASSIGNED
--	--

- ☐ **Project** 를 클릭합니다.
- ☐ 새 Git 을 만들려면 **New Repository** 를 클릭하시오.

REPOSITORIES

+ New Repository

Filter Git Repositories

All Favorites

Favorites First

TwitterFeedMicroservice.git

Twitter Feed Microservice

HTTP

SSH

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

Maven

HTTP

DAV

https://developer.em2.oraclecloud.com/profile/developer71725-gse00002057/s/developer7172

- New Repository Wizard(새 마법사)마법사에서 다음 정보를 입력하고 Create(생성)을 클릭합니다.

이름= `TwitterMarketingUIMicroservice`

설명=트위터 마케팅 UI Microservice

초기 콘텐츠= `Import existing repository`

URL 입력:

`https://github.com/oraclenassolutionengineering/JETTwitterQuickStart.git`

New Repository [X]

Name

Description

Initial content

- ☐ Empty Repository
- ☐ Initialize repository with README file
- ☒ Import existing repository

► **Credentials** for non-public repos

Create **Cancel**

□ 이제 기존 저장소를 기준으로 새 GIT 저장소를 생성했습니다.

TwitterMarketingUIMicroservice.git		master	Files	Commits	Branches	Tags	Compare
Description	HTTP	SSH	Twitter Marketing UI Microservice				
doc_root	Add dynamic graph code Ed Shnekendorf				May 19, 2016 11:39 AM -0600		
target	Initial Commit ccrock				March 15, 2016 8:29 AM -0600		
gitignore	Initial commit ccrock				March 15, 2016 8:22 AM -0600		
manifest.json	Initial Commit ccrock				March 15, 2016 8:29 AM -0600		
package.json	update versions and fix description Ed Shnekendorf				May 19, 2016 11:46 AM -0600		
README.md	Update README.md Ed Shnekendorf				May 19, 2016 11:47 AM -0600		
server.js	Initial Commit ccrock				March 15, 2016 8:29 AM -0600		
README.md							
JET Hackathon Quickstart							
This project serves as the seed for Oracle's Microservices Hackathon							

기본 빌드 및 구축 프로세스 생성

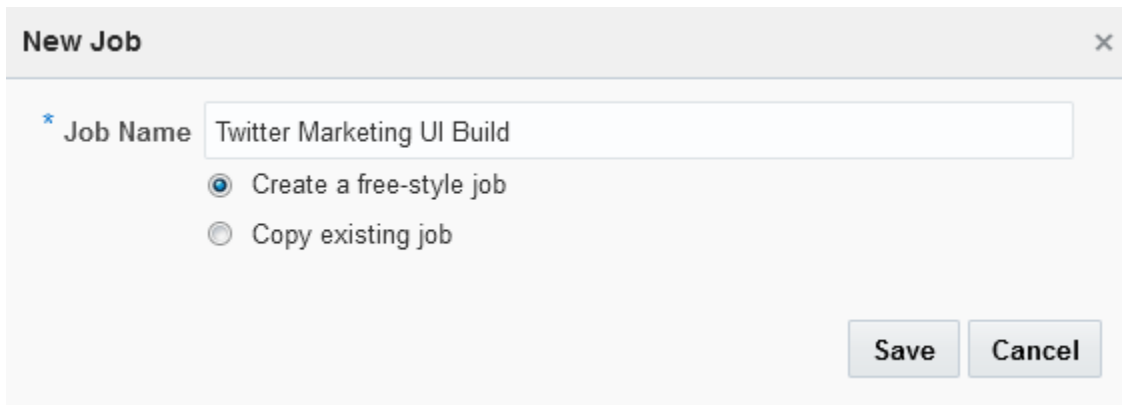
STEP 5: 기본 빌드 프로세스 생성

이제 관리되는 GIT 저장소에 소스 코드가 있으므로 마스터 branch 에 커밋 될 때마다 생성되는 빌드 프로세스를 생성해야 합니다. 이 섹션에서는 웹 스크립트 빌드 프로세스를 설정합니다.

□ 빌드 페이지에 액세스 하려면 **Build** 를 클릭하고 **New Job**(새 작업)을 클릭합니다.

Status	Weather	Job	Last Success
✓	☀	Twitter Feed Build	41 minutes ago

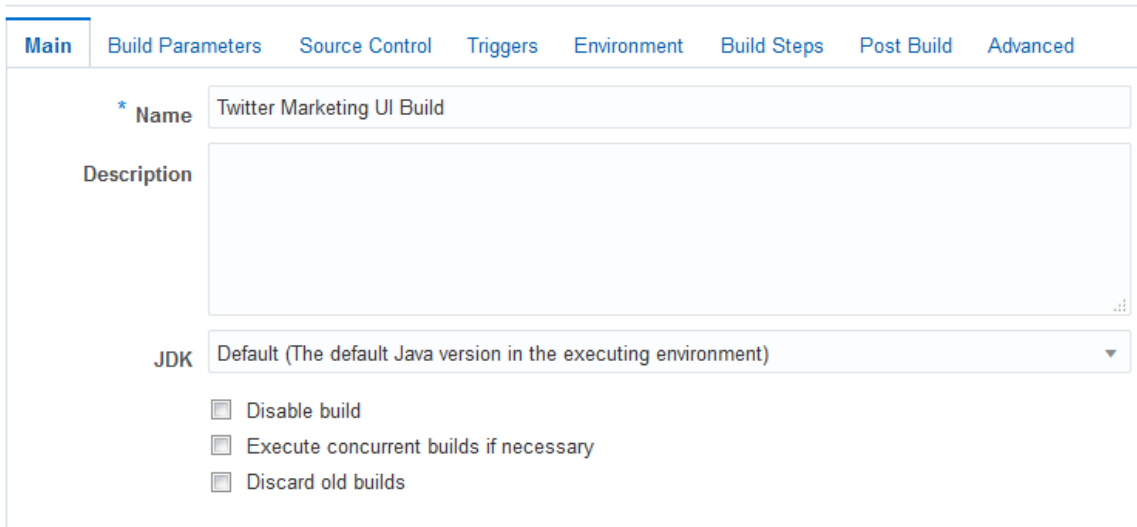
- New Job 팝업에서 작업 이름으로 **Twitter Marketing UI Build** 를 입력하고 Save(저장)을 클릭합니다.



The image shows 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 buttons: 'Create a free-style job' (which is selected) and 'Copy existing job'. At the bottom right are 'Save' and 'Cancel' buttons.

- 이제 작업 구성 화면에 배치됩니다.

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



The image shows the 'Configure build job' page for 'Twitter Marketing UI Build'. It has a tabbed interface with 'Main', 'Build Parameters', 'Source Control', 'Triggers', 'Environment', 'Build Steps', 'Post Build', and 'Advanced'. The 'Main' tab is active, showing fields for 'Name' (Twitter Marketing UI Build), 'Description' (empty), and 'JDK' (Default (The default Java version in the executing environment)). Below these are three checkboxes: 'Disable build', 'Execute concurrent builds if necessary', and 'Discard old builds', all of which are unchecked.

- **Source Control** 탭을 클릭합니다. **Git** 을 클릭하고 URL 드롭 다운에서 **TwitterMarketingUIMicroservice.git** 를 선택한다.

참고:Twitter 마케팅 UI 에 대한 Git 저장소를 선택해야 합니다.

[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**

- **Triggers** 탭을 클릭합니다. **SCM polling schedule** 를 근거로 선정하고, 스케줄을 추가합니다. `*/1*****`

참고: Schedule 에 표현식을 넣지 않아도 commit 이 되게 되면 자동으로 트리거가 됩니다.
 위 표현은 추가로 매 분마다 체크 하여 변경 사항을 확인합니다.

[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

□ Build 단계 탭을 클릭합니다. **Add Build Step** 를 클릭하고 **실행 셸** 을 선택합니다.

◀ [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

□ Command 에 **npm install** 을 넣으세요

◀ [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`

×

- ☐ **Post Build** 탭을 클릭합니다. **Archive the artifacts** 를 확인하고 Files to Archive 에 ****/*target/*** 를 입력하세요. 압축 형식에서 GZIP 를 확인합니다.

[< 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

☐ Enable auto validation for file masks

Excludes

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

Compression Type

- ☐ Save(저장)를 클릭하여 구성을 완료합니다. 빌드는 1 분 또는 2 분 이내에 자동으로 시작해야 합니다. 자동으로 시작되지 않으면 지금 Build Now 단추를 클릭합니다. 상태는 다음과 같이 변경됩니다.

[< Jobs Overview](#) | [Twitter Marketing UI Build](#)
Build Now

Description
No description available

Permalinks
[Last](#)

Notifications
On Off
CC Me

Build History

New Build
N/A

Status	Build	Time	Duration	Console
	#1	Just now	N/A	

Artifacts of Last Successful Build

Build Trend

Conso

Loading...

- 빌드가 시작되면 빌드가 완료되는 데 약 1~2 분 정도 소요됩니다. 배포 구성을 하기 위해서는 빌드 결과물이 필요하기 때문에 빌드가 완료될 때까지 기다리십시오.

< Jobs Overview | Twitter Marketing UI Build

Description

No description available

Permalinks

Last | Successful | Completed | Stable

Notifications

On Off

CC Me

Build History

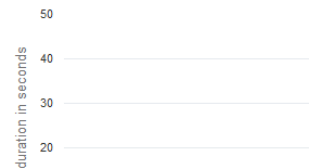
Status	Build	Time	Duration	Console
✓	#1	Just now	46 s 73 ms	

Artifacts of Last Successful Build

target

(all files in zip)

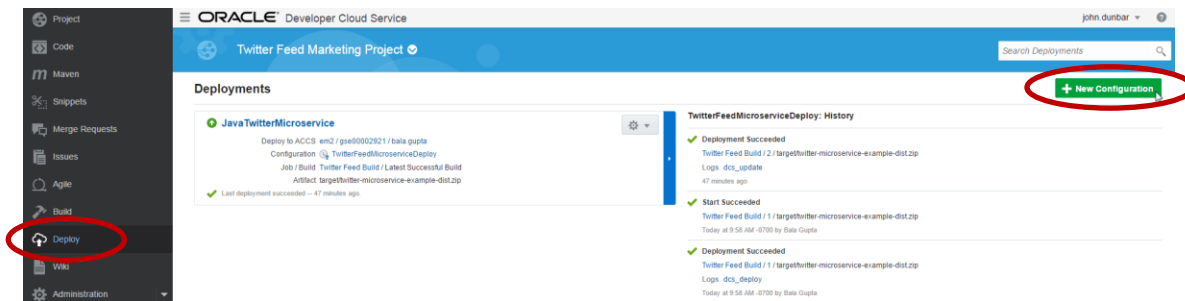
Build Trend



STEP 6: 기본 구축 프로세스 생성

이제 자동화된 빌드 프로세스를 구축하기 위해 구성을 설정할 예정입니다. Build 가 성공할 때 마다 애플리케이션 컨테이너 클라우드 서비스에서 실행되는 node.js 에 build 된 결과물이 올려지게 됩니다.

- 배포를 클릭하여 배포 페이지에 접근해서 New Configuration(새 구성)을 클릭합니다.



- 다음 데이터를 입력합니다.

구성 이름= TwitterMarketingUIDeploy

응용 프로그램 이름= JETFrontEndApp

* Configuration Name

* Application Name

- Deployment Target 에서 **New** 를 선택하고 **Application Container Cloud**(애플리케이션 컨테이너 클라우드)를 선택합니다.

* 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...

- 다음 데이터를 입력하고 **TestConnection**(테스트 연결)을 클릭합니다. 성공적으로 연결을 클릭하면 연결 사용

데이터 센터= `EMEA Commercial 2 - em2`

**`EMEACommercial2` 가 아닌 경우 적절한 데이터 센터 (해당하는 경우)

ID 도메인=<ID 도메인>>

사용자 이름=사용자 이름

Password=< 비밀 번호>

Deploy to Application Container Cloud ✕

* Data Center

* Identity Domain

* Username

* Password

✓ Successful Use Connection

- ACCS 속성을 런타임 **Node** 와 **Hourly** 으로 설정합니다. **Automatic** 을 클릭합니다. **Twitter Marketing UI Build** Job 을 선택하고, 배포할 결과물로 **target/jet-quickstart-client-dist.zip** 를 선택합니다.

* Configuration Name

* Application Name

* Deployment Target New ▾

ACCS Properties

Runtime ☒ Java ☒ Node

Subscription ☒ Hourly ☐ Monthly

Type ☐ On Demand ☒ Automatic

☒ Deploy stable builds only

* Job

* Artifact

- 저장을 클릭하십시오.

Save Cancel

* Configuration Name

* Application Name

* Deployment Target New ▾

ACCS Properties

Runtime ☐ Java ☒ Node

Subscription ☒ Hourly ☐ Monthly

Type ☐ On Demand ☒ Automatic

☒ Deploy stable builds only

* Job

* Artifact

- 드롭 다운 메뉴를 클릭하고 **시작** 을 선택합니다.

JETFrontEndApp

Deploy to ACCS [em2 / gse00002057 / john.dunbar](#)

Configuration [TwitterMarketingUIDeploy](#)

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

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

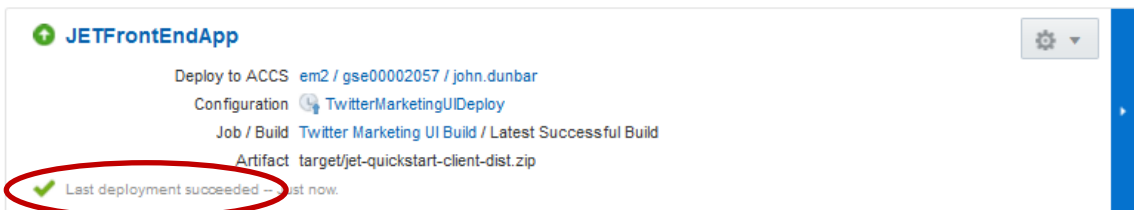
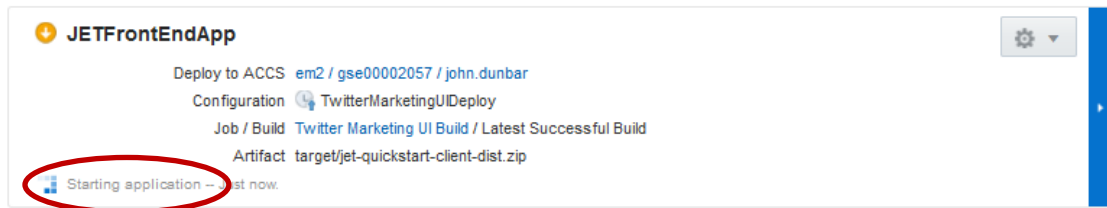
Start

Redeploy

Edit Configuration

Delete Configuration

- 메시지가 **Starting application** 에서 **Last deployment succeeded** 로 될때까지 기다립니다.

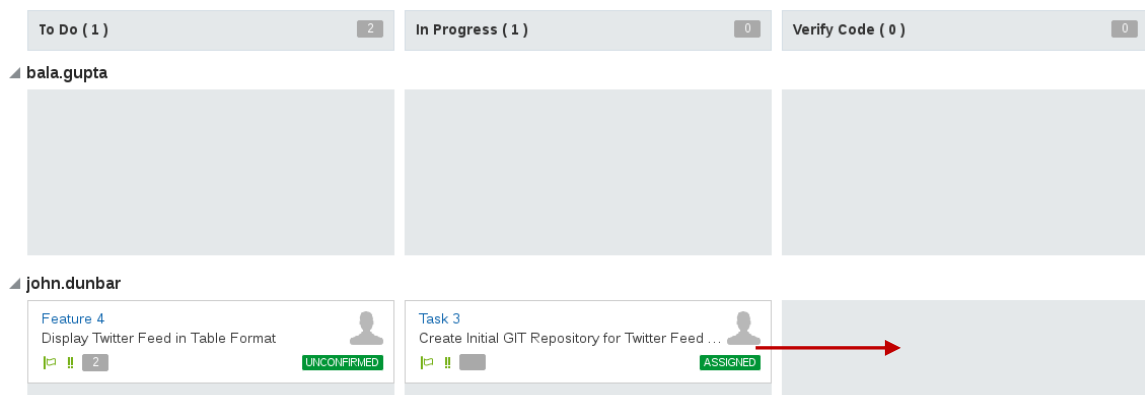


Twitter Marketing UI 의 기본 배포 검증(verify)

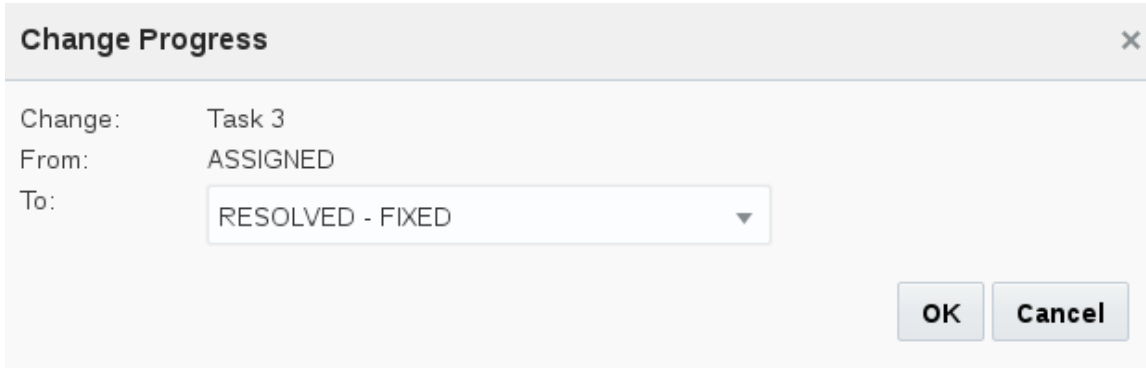
STEP 7: 검증(Verified) 상태로 전환

이제 build 된 결과물을 애플리케이션 컨테이너 클라우드 서비스에 성공적으로 배포했기 때문에, agile 보드에 상태를 반영할 것입니다. 다음 검증(verify)과정은 매우 간단하지만, 새 기능을 직접 검증하기 전에 과제를 "Verify Code"컬럼으로 이동시키도록 할 예정입니다.

- Agile 를 클릭하고 Active Sprints 를 클릭합니다. **In Progress** 에서 작업 3 을 끌어다 **Verify Code** 컬럼에 Drag and Drop 으로 놓습니다.



- ☐ 진행 상황 변경 팝업 화면에서 **OK(확인)**를 클릭합니다.



Change Progress [X]

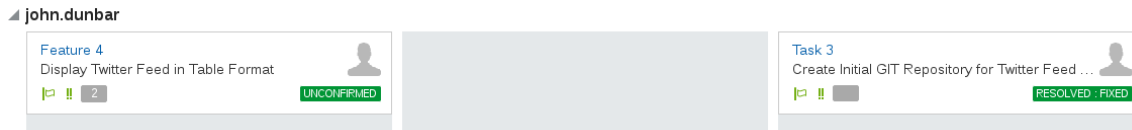
Change: Task 3

From: ASSIGNED

To: RESOLVED - FIXED ▼

OK Cancel

- ☐ 이제 완료되기 전에 확인을 위해 코드를 확인할 수 있습니다.



john.dunbar

Feature 4
Display Twitter Feed in Table Format

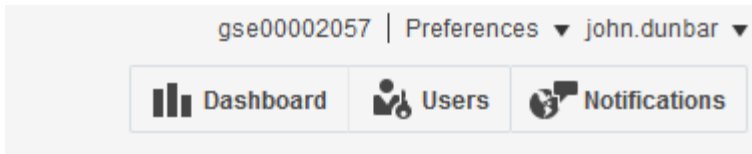
UNCONFIRMED

Task 3
Create Initial GIT Repository for Twitter Feed ...

RESOLVED - FIXED

STEP 8: Oracle 애플리케이션 컨테이너 클라우드 서비스에 로그인

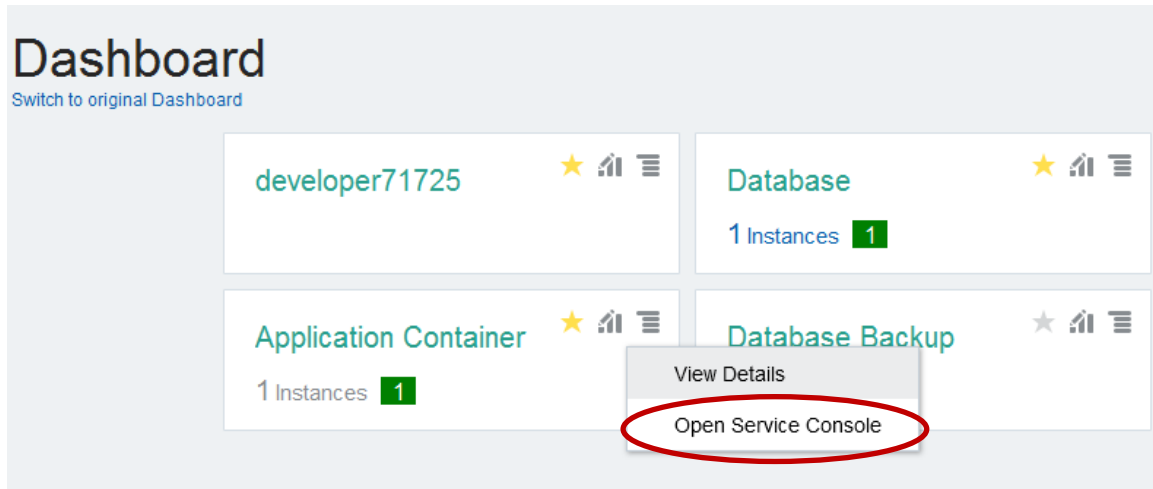
- ☐ Oracle Public 클라우드 탭으로 다시 이동합니다. 대시 보드를 클릭하여 메인 클라우드 서비스 대시 보드로 돌아갑니다.



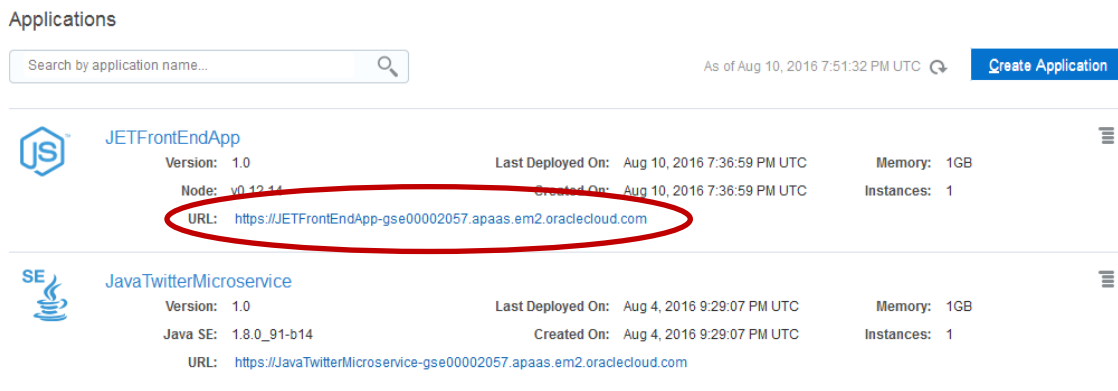
gse00002057 | Preferences ▼ john.dunbar ▼

Dashboard Users Notifications

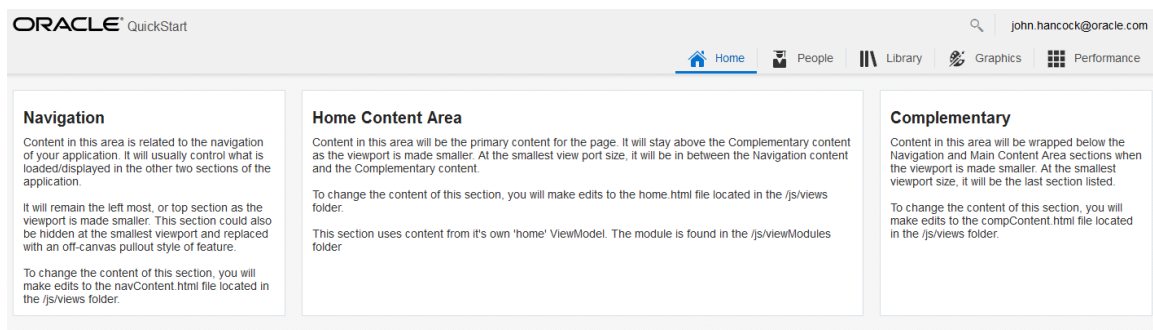
-  Application(애플리케이션)컨테이너에서 **Open Service Console** 를 선택합니다.



- ACCS Service Console 에서는 새로 생성된 **JETFrontEndApp** 애플리케이션을 비롯한 모든 배포된 애플리케이션을 볼 수 있습니다.



- URL 을 클릭하여 애플리케이션을 가져옵니다.



STEP 9: 작업 완료

우리는 현재 우리의 애플리케이션이 구축되었고 동작한다는 것을 확인했습니다. 이 부분을 완료하기 위해 우리는 스프린트에서 완성된 이슈를 완료로 표시하고자 합니다.

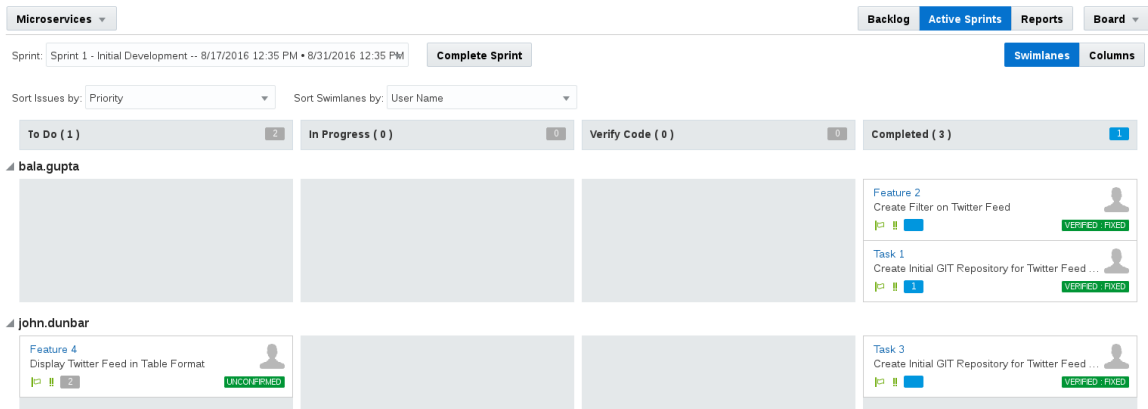
- ☐ 개발자 클라우드 서비스로 돌아가서, **Agile** 를 클릭하고 **Active Sprints** 를 클릭합니다.
- ☐ **Verify Code** 에서 **작업 3** 를 Drag and Drop 해서 **Completed** 로 옮깁니다.

The screenshot shows the Agile Sprints interface. At the top, there are two columns: 'Verify Code (1)' with a count of 0, and 'Completed (2)' with a count of 1. Below these columns, there are two task cards. The first card is 'Feature 2' with the title 'Create Filter on Twitter Feed' and a status of 'VERIFIED : FIXED'. The second card is 'Task 1' with the title 'Create Initial GIT Repository for Twitter Feed ...' and a status of 'VERIFIED : FIXED'. A third card, 'Task 3' with the title 'Create Initial GIT Repository for Twitter Feed ...', is shown below the first two. This card has a status of 'RESOLVED : FIXED'. A red arrow points from this card to the 'Completed' column, indicating the action of moving the task to the completed state.

- ☐ 진행 중인 팝업 창에서 **OK(확인)**를 클릭합니다.

The screenshot shows a 'Change Progress' dialog box. It has a title bar with a close button (X). The dialog contains the following fields: 'Change:' with the value 'Task 3', 'From:' with the value 'RESOLVED - FIXED', and 'To:' with a dropdown menu showing 'VERIFIED - FIXED'. At the bottom right, there are two buttons: 'OK' and 'Cancel'.

□ 이제 스프린트는 다음과 같아야 한다.



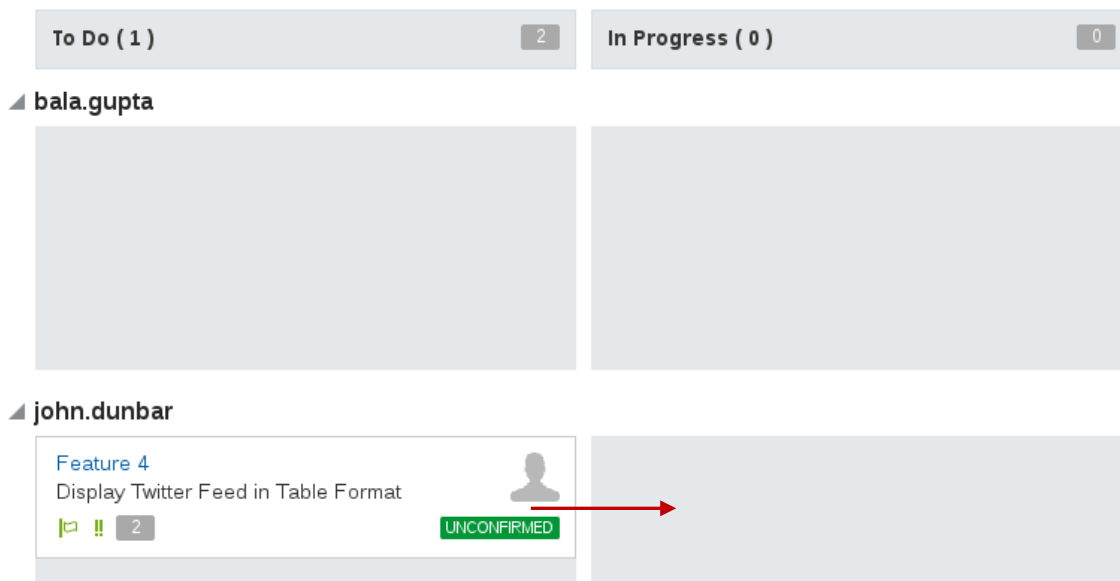
기본 응용 프로그램을 확장하여 Twitter 피드 표시

이제 기본 애플리케이션을 사용하여 이 애플리케이션을 확장하여 트위터 피드를 화면에 추가하고자 합니다. 이 과제를 위해 우리는 Brackets 텍스트 편집기를 사용하도록 하겠습니다. 개발자 클라우드 서비스에서 코드를 받아서 수정사항을 추가할 것입니다. 새로운 코드가 배포되면 코드를 검토하여 build 하고 배포하기 전에 코드 검토를 진행할 것입니다.

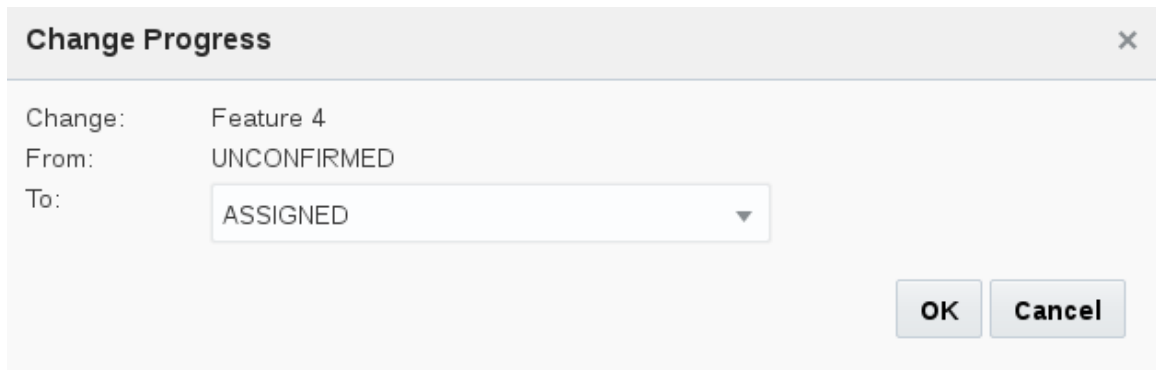
STEP 10: Task 를 '작업 중' 상태(In Progress)로 이동

이 lab 의 이 부분을 시작하기 위해 우리는 스프린트의 진행 상황에 대한 이슈를 진행하고자 합니다.

- ☐ 개발자 클라우드 서비스로 돌아가려면 **Agile** 을 클릭하고 **Active Sprints** 를 클릭합니다.
- ☐ **Feature 4** 를 Drag and Drop 해서 **To Do** 에서 **In Progress** 로 옮깁니다.



- 진행 중인 팝업 창에서 **OK(확인)**를 클릭합니다.



A dialog box titled "Change Progress" with a close button (X) in the top right corner. It contains the following text:

Change: Feature 4
 From: UNCONFIRMED
 To: ASSIGNED

At the bottom right, there are two buttons: "OK" and "Cancel".

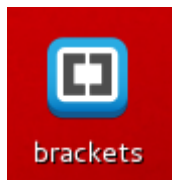
Brackets 텍스트 편집기에서 Clone Project

STEP 11: Brackets 텍스트 편집기 시작

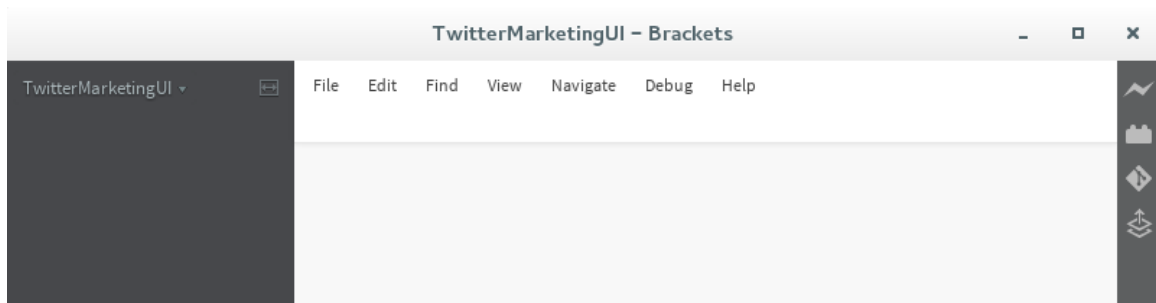
- Brackets 텍스트 편집기를 시작합니다. 어떻게 Brackets 을 시작할 것인지에 따라 달라질 수 있습니다. OEL 이미지에서 Brackets 을 어떻게 시작하는지에 대한 부분을 이 문서에 포함시켰습니다.

참고: Brackets 이 설치되지 않은 경우 부록에서 확인하실 수 있습니다.

- 바탕 화면 아이콘 **Brackets** 에 마우스 오른쪽 단추로 클릭하고 **Open(열기)**을 선택합니다.

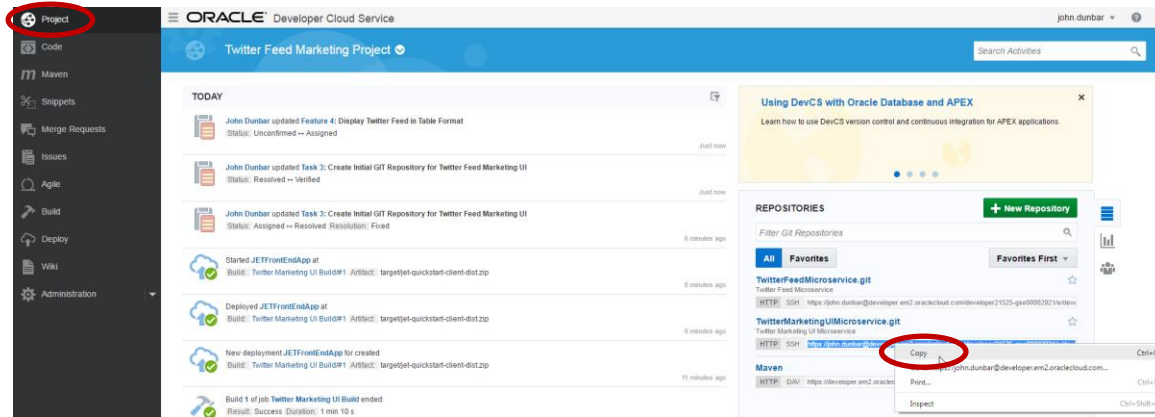


- **TwitterMarketingUI** 폴더가 이미 로드된 상태에서 열려야 합니다.




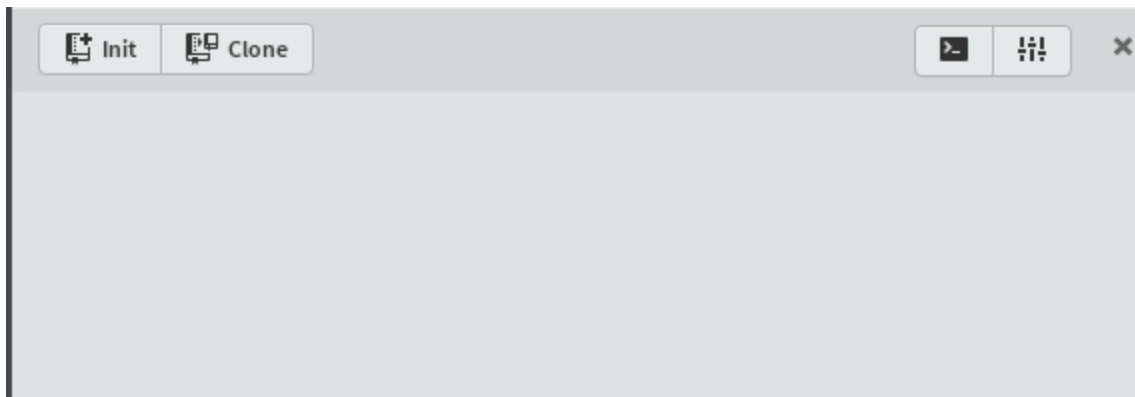
STEP 12: GIT URL 복사

- 개발자 클라우드 서비스로 돌아가려면 **프로젝트**를 클릭하십시오. 오른쪽에 있는 **TwitterMarketingUIMarketingMicroservice.git** URL 을 선택합니다. 오른쪽 클릭 및 **Copy(복사)**선택



STEP 13: GIT Repository 복제

- Brackets 편집기의 오른쪽에 있는  **GIT** 아이콘을 선택합니다.



- 개발자 클라우드 서비스에서 캡처한 Git URL 에서 **Clone**(복제) 및 붙여 넣기 클릭합니다. 사용자 이름은 자동으로 입력됩니다. 암호를 입력하고 Save(저장)을 클릭합니다. 완료되면 OK(확인)를 클릭하여 복제 프로세스를 시작합니다.

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.

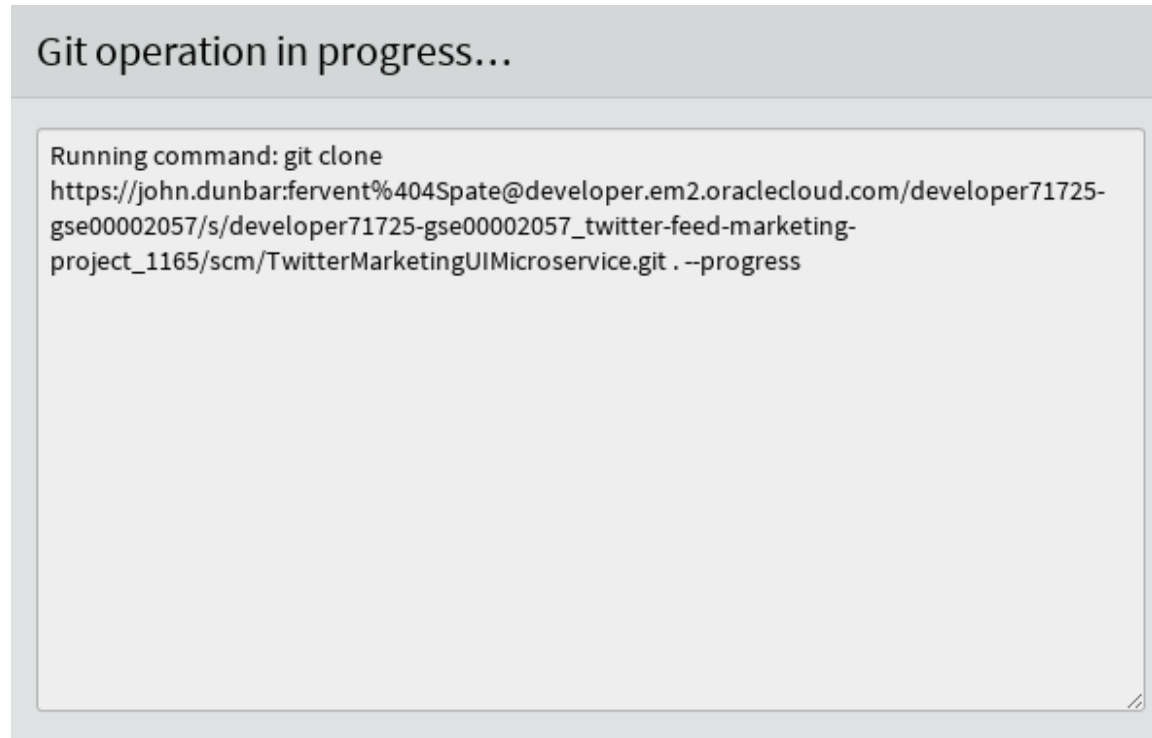
Username:

Password:

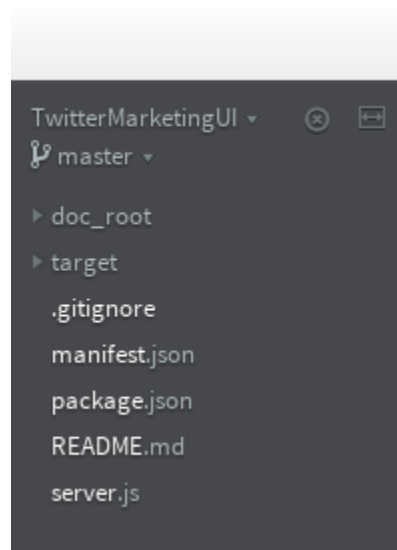
☒ Save credentials to remote url (in plain text)

CancelOK

- ☐ 복제가 실행되는 동안 진행 상황이 표시됩니다.

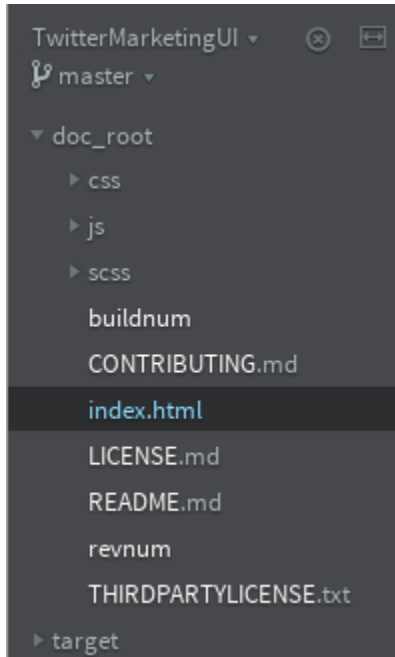



- ☐ 이제 해당 저장소의 로컬 복사본이 생성되었습니다.

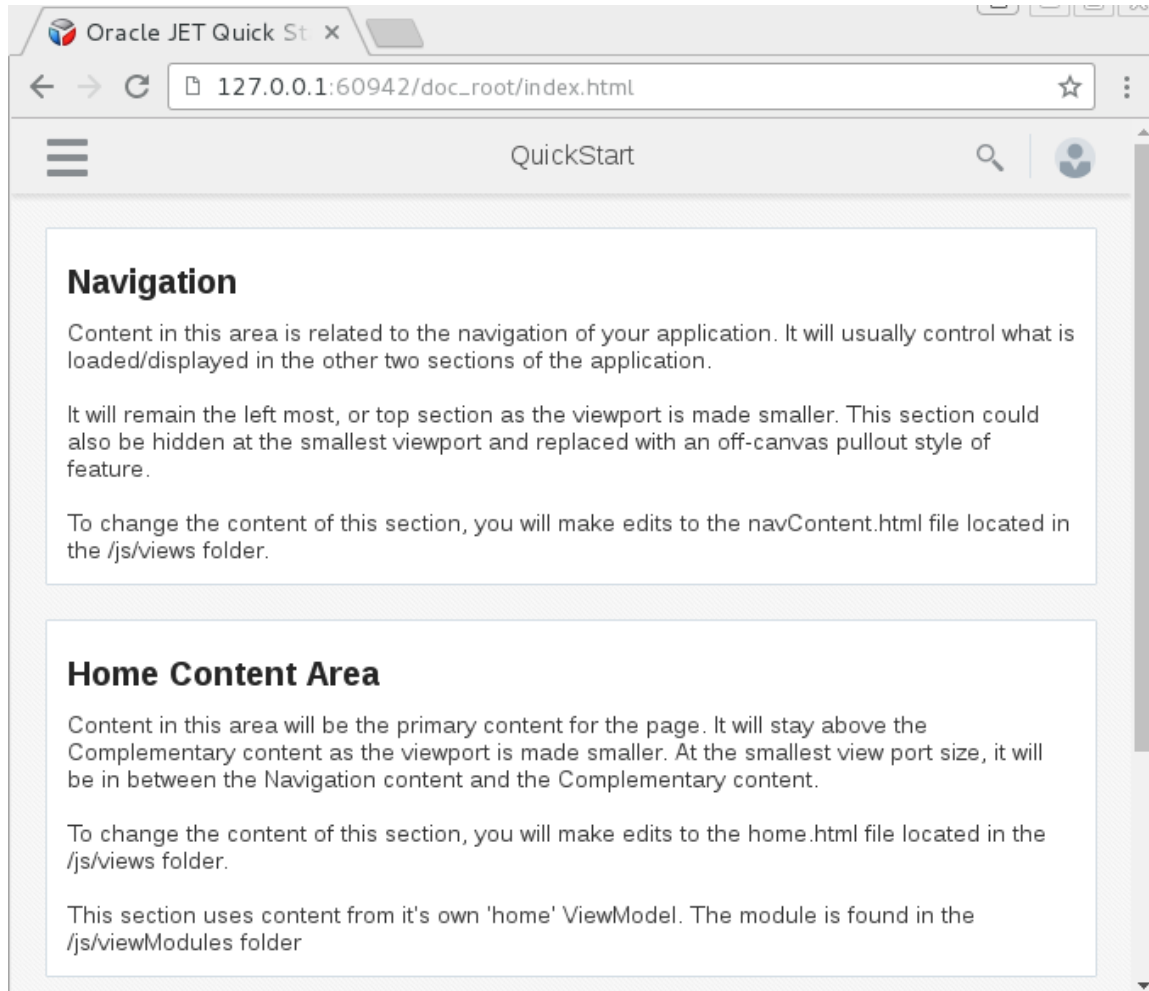


STEP 14: 라이브 미리 보기를 실행합니다.

- ☐ 코드를 변경하기 전에 먼저 코드를 로컬로 실행하십시오.
- ☐ **doc_root** 를 확장하고 **index.html** 를 선택합니다.



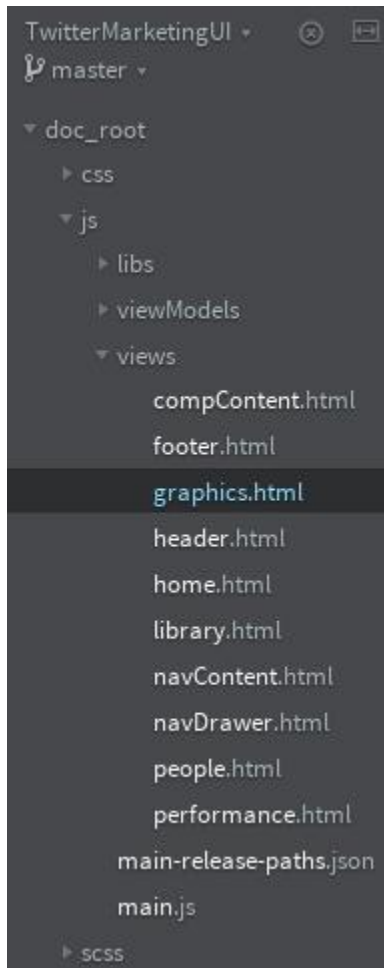
- 오른쪽 패널에서 라이브 미리 보기 를 클릭합니다. 브라우저에서 브라우저 응용 프로그램을 시작합니다. 응용 프로그램이 동작하는 것을 확인하면 브라우저를 닫을 수 있습니다.



Twitter Feed 를 표시하기 위해 코드를 추가

STEP 15: Graphics.html 수정

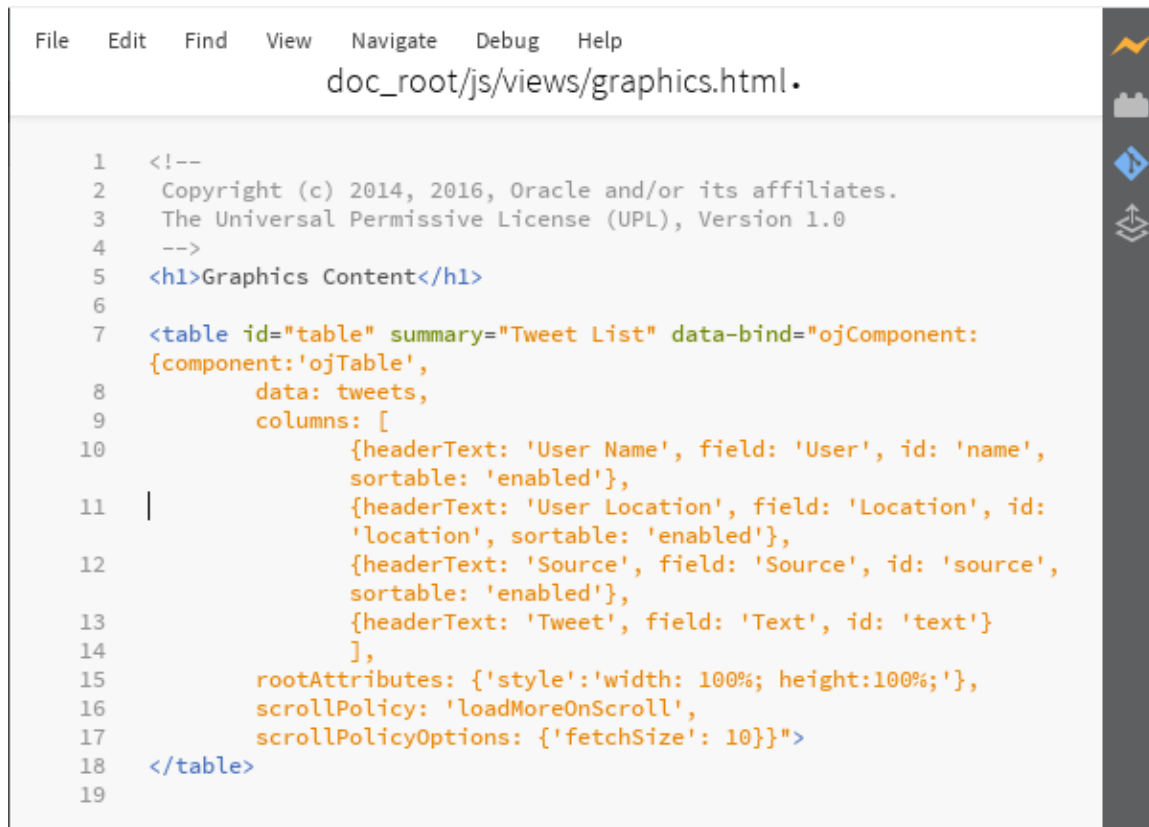
□ doc_root 확장-> js ->views 를 클릭하고 graphics.html.를 클릭합니다.



□ 배포해드린 Lab 파일 중 files 폴더에 있는 graphics.html 파일을 붙여 넣으시기 바랍니다.

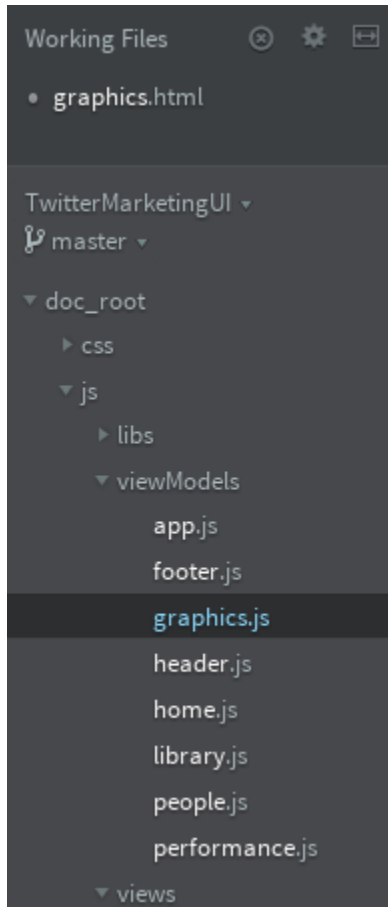
```
<h1>Graphics Content</h1>

<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**.



□ 배포해드린 Lab 파일 중 files 폴더에 있는 graphics.js 파일을 붙여 넣으시기 바랍니다.

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

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

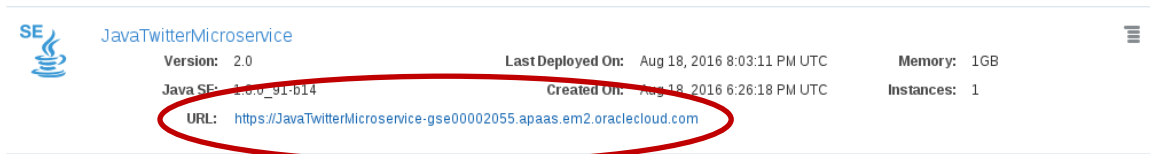
    function mainContentViewModel() {

        // 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;
});
```

- 브라우저로 돌아가 Application Container(애플리케이션 컨테이너)클라우드 서비스 콘솔로 이동합니다. lab 200 에서 만든 **JavaTwitterMicroservice** URL 을 복사해 주세요.



- root 변수를 기존 URL 을 이 URL 로 대체합니다.

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

□ 완성된 그래픽은 아래 이미지와 같은 모양이어야 합니다.

doc_root/js/viewModels/graphics.js.

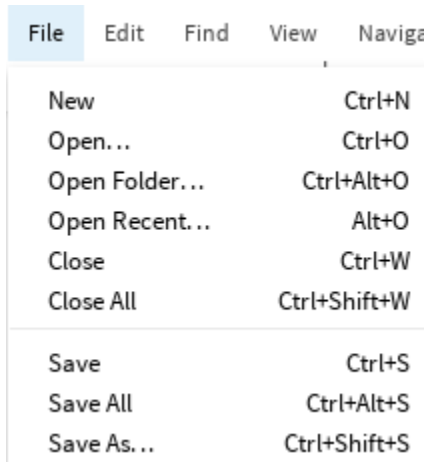
```

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         $.ajax({



```

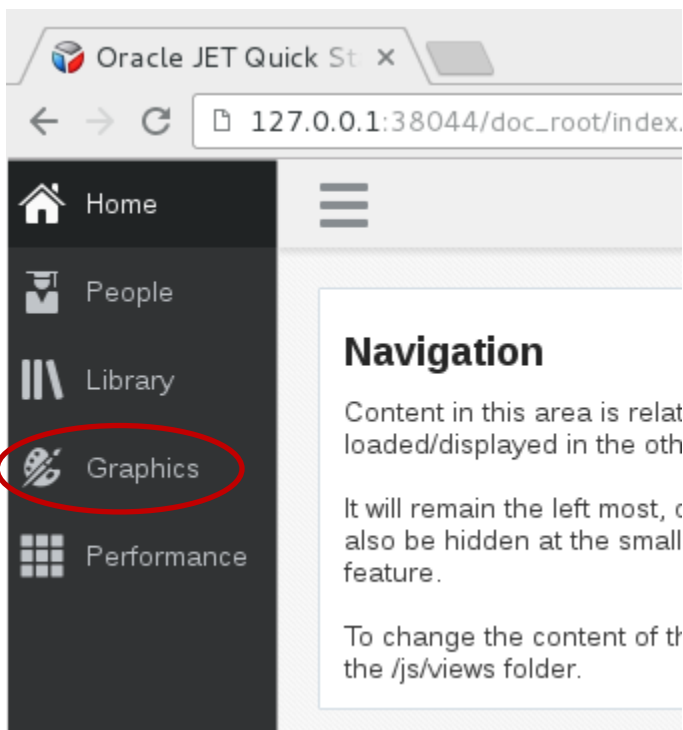
Line 17, Column 91 — 53 Lines INS JavaScript ▼ ✓ Spaces: 4

- ☐ 파일을 모두 저장하려면 **File → Save All(모두 저장)**을 클릭합니다.



STEP 17: 새 변경 사항을 테스트하십시오.

- ☐  라이브 Preview 를 클릭하여 새 변경 내용을 테스트합니다.
- ☐  클릭하고 그래픽을 선택합니다.



□ 이제 그래픽 섹션에서 모든 트위터 공급(feed) 데이터를 볼 수 있습니다.

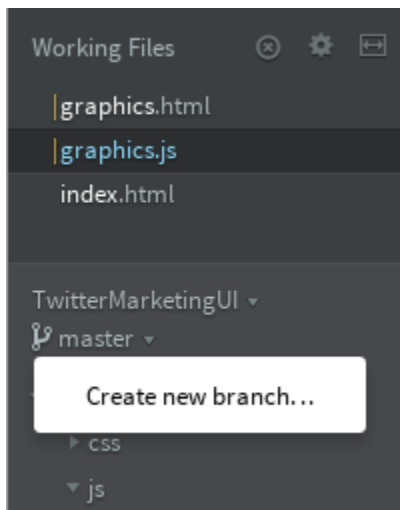
Graphics Content			
User Name	User Location	Source	Tweet
Nick Courtney	Portsmouth, UK	Facebook	The V
LizQuenlurve	Republic of the Philippines	TweetDeck	#Dolci
Jug		Twitter for iPhone	@Coc
Amethyst		Twitter for Android	RT @
Sharon ♥JB	Florida, USA	Twitter for Android	@kidr
Katie	England, United Kingdom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

새 Branch 및 Commit 코드 생성

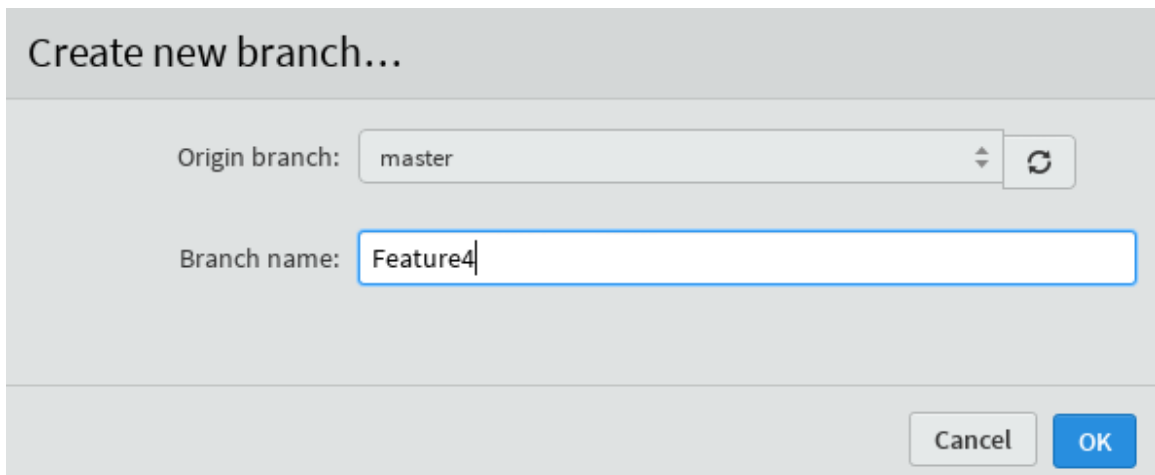
새 Branch 와 코드 Commit


STEP 18: 새 Branch 와 Commit 코드 생성

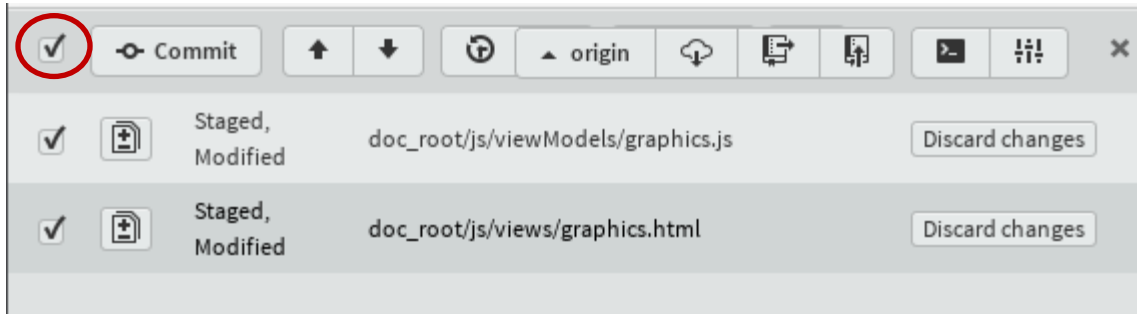
- 우선 이 기능에 대한 모든 변경 사항을 확인하기 위해 새로운 Branch 를 생성해야 합니다. 좌측 탐색 패널에서 **master** 를 선택하고 **Create New branch** 를 클릭합니다.



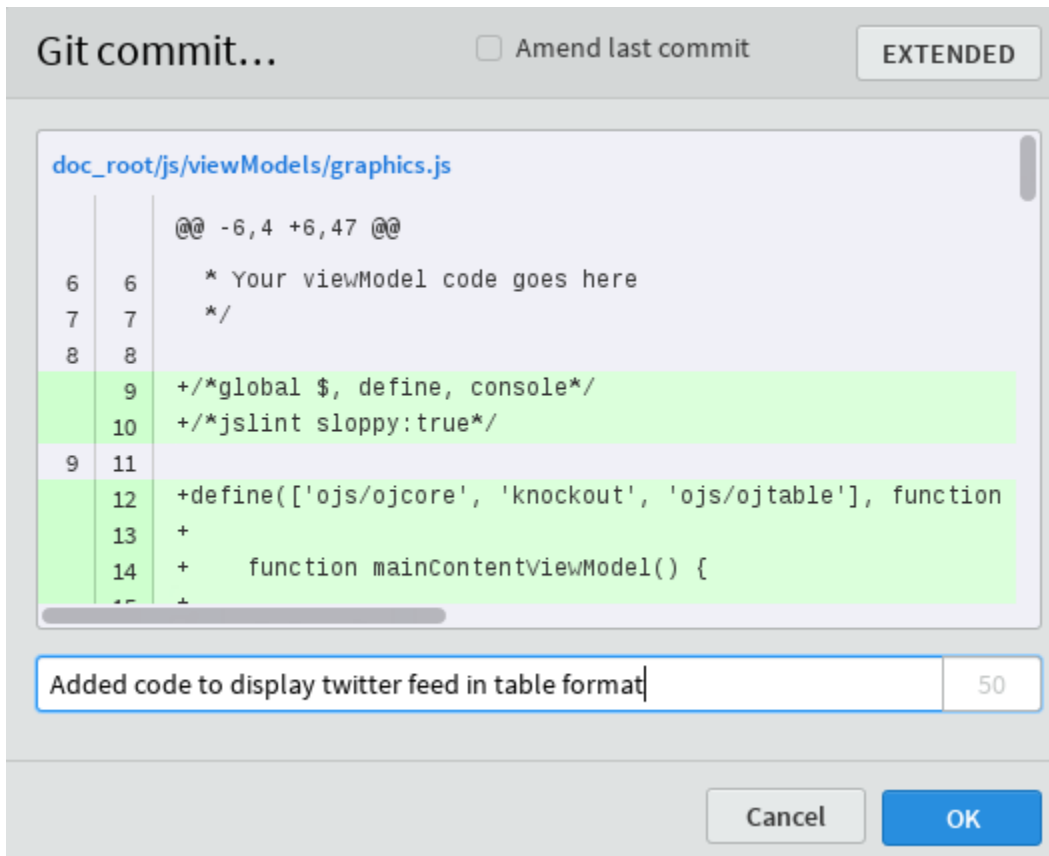
- 팝업 창에서 Branch 이름으로 **Feature4** 을 입력하고 OK(확인)를 클릭합니다.




- ☐  Git 아이콘을 클릭하시오. 수정된 모든 파일을 선택하려면 **Commit** 옆에 있는 상자를 확인하십시오.



- ☐ Commit 을 클릭합니다. 팝업 창에서 Added 코드를 입력하여 표 형식을 표시하고 확인을 클릭합니다.



- ☐  (1) Git Push 아이콘을 클릭하시오.
- ☐ 팝업 창에서 모든 기본 값을 그대로 두고 OK(확인)를 클릭합니다.

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)

CancelOK

☐ 일단 Git 이 완료되면 OK 를 클릭하시오.

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

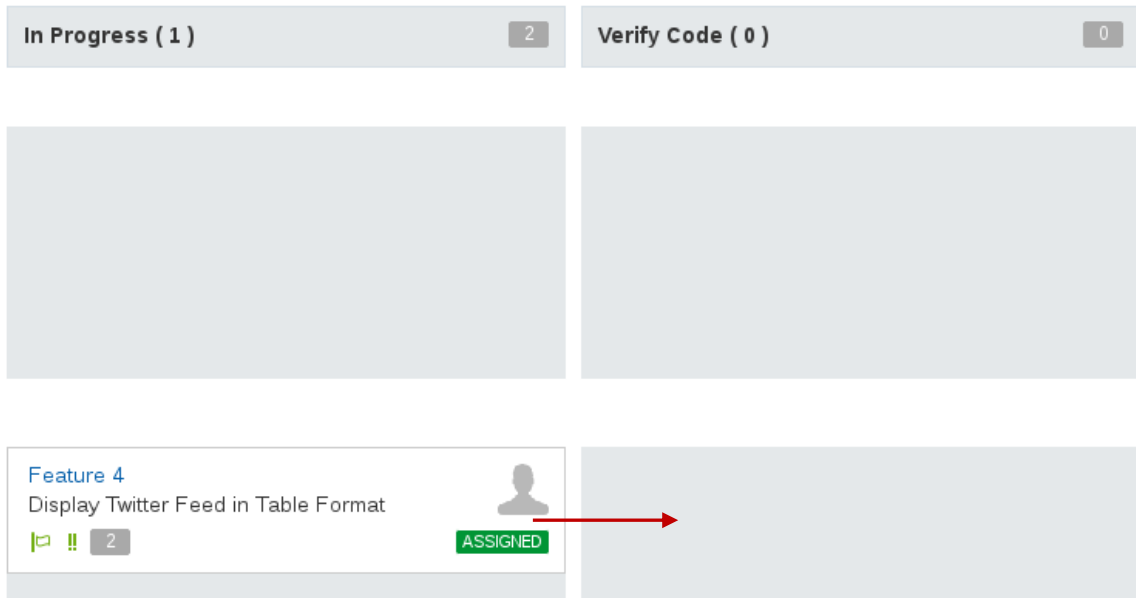
Summary - [new branch]

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

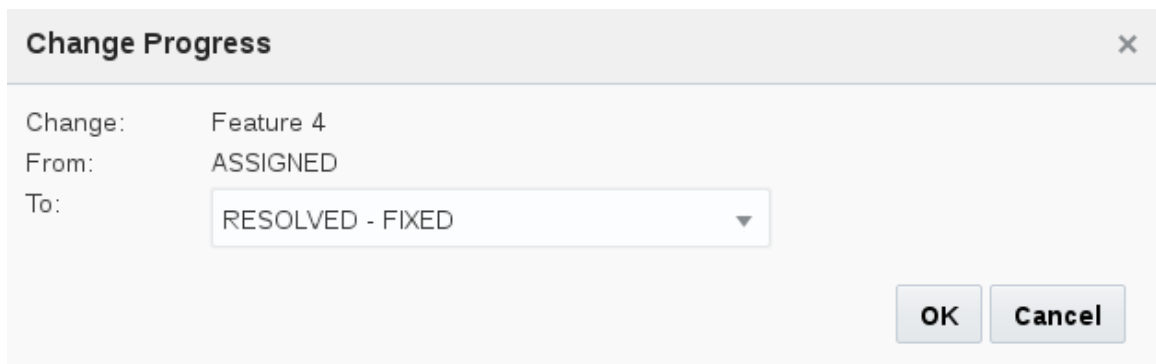
OK

STEP 19: Display Twitter Feed 작업을 완료합니다.

- ☐ 개발자 클라우드 서비스 창으로 돌아가려면 애자일(Agile)을 클릭하고 **Active Sprints** 를 클릭합니다.
- ☐ **Feature 4** 를 **In Progress** 에서 **Verify Code** 로 Drag and Drop 해서 이동합니다.



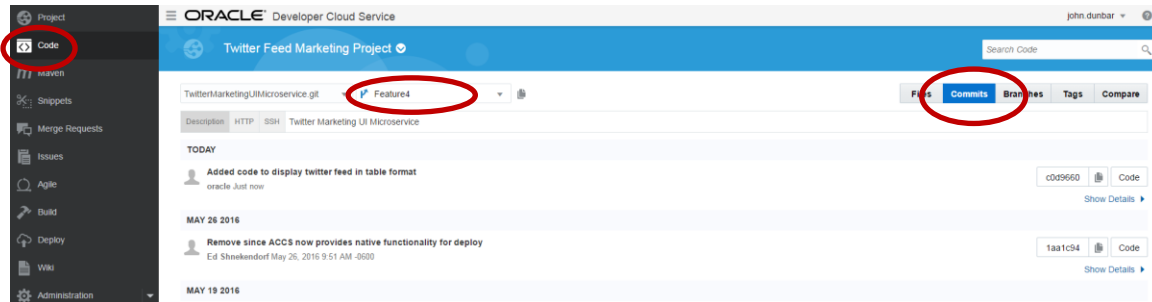
- ☐ Progress(진행)팝업 창에서 OK(확인)를 클릭합니다.



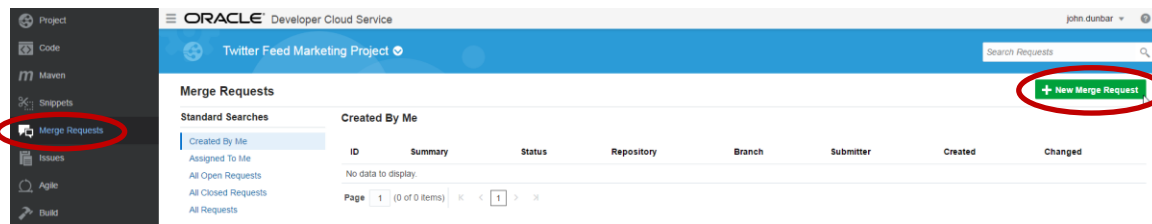
병합 요청 생성

STEP 20: 스프린트 상태 검토 및 병합 요청 생성

- **Code** 탭을 클릭하고 **Feature4** Branch 를 선택한 다음 하위 탭을 클릭합니다. 자 이제 최근 commit 된 것을 볼 수 있는데, Brackets 에서 commit 된 내용들을 확인할 수 있습니다.



- John Dunbar 가 테이블 형식으로 twitter feed 를 표시하는 작업을 완료했으므로, 병합 요청(Merge Request)이 존에 의해 작성되어 리사 존즈에 할당될 수 있습니다. 병합 요청(Merge Request)을 클릭한 다음 **New Request** 버튼을 클릭합니다..



□ 새 병합 요청에 다음 정보를 입력하고 **Next**(다음)을 클릭합니다.

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

New Merge Request

Progress: Branch (selected), Details, Description

Repository: TwitterMarketingUIMicroservice.git

Target Branch: master

Review Branch: Feature4

oracle Today
c0d9660 Added code to display twitter feed in table format

Next >

□ **Details** 에 다음 정보를 입력하고 **Create** 를 누릅니다.

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

New Merge Request

Progress: Branch, Details (selected), Description

Linked Issues: Search and Link Issues

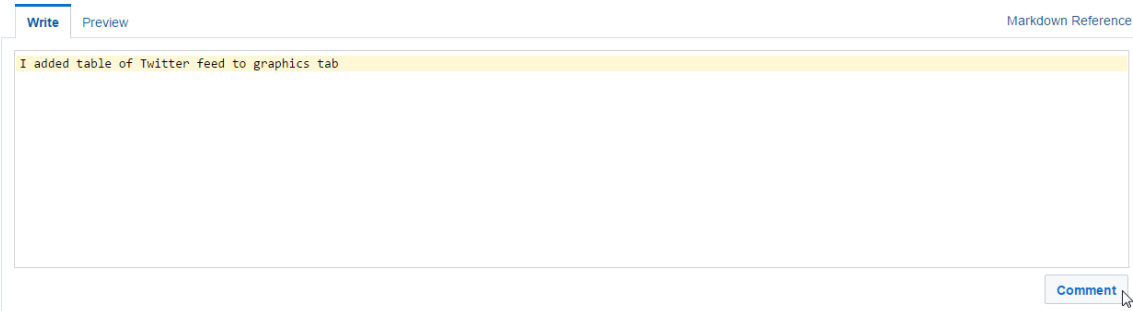
Linked Builds: Search and Link Build Jobs

Summary: Merge Feature 4 into master

Reviewers: Lisa Jones x |

+ Create Cancel

- ☐ Write 박스에서 다음 설명을 입력한 다음 Comment 버튼을 클릭하여 "그래픽 탭에 Twitter 피드를 추가합니다." 라고 입력합니다.



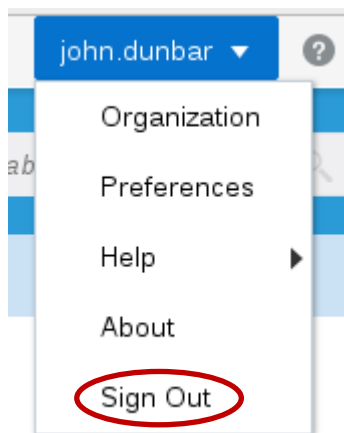
Lisa Jones 로 Branch 를 Merge 하기

다음 단계에서 "리자"는 "존"에 의해 만들어진 branch 를 master 에 병합을 할 것입니다.

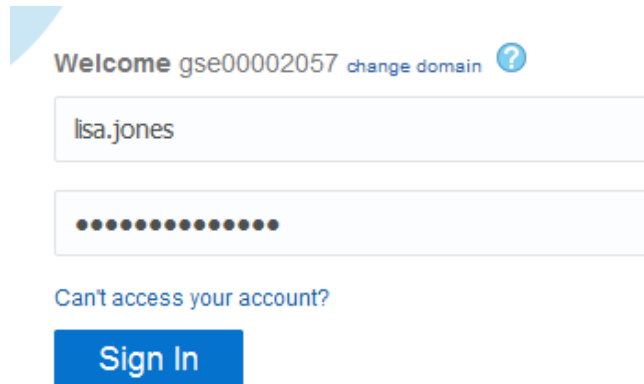
참고: 단일 사용자 환경을 사용하는 경우 다음 단계를 건너뛰고 "병합 요청"이라는 제목으로 다음 단계를 진행합니다.

STEP 21: john Dunbar 으로 Sign Out 하고 Lisa Jones 로 사인을 해 주세요

- ☐ 화면 오른쪽 상단 모서리에 있는 john.dunbar 드롭 다운 목록을 클릭합니다. SignOut(로그아웃)을 선택합니다.

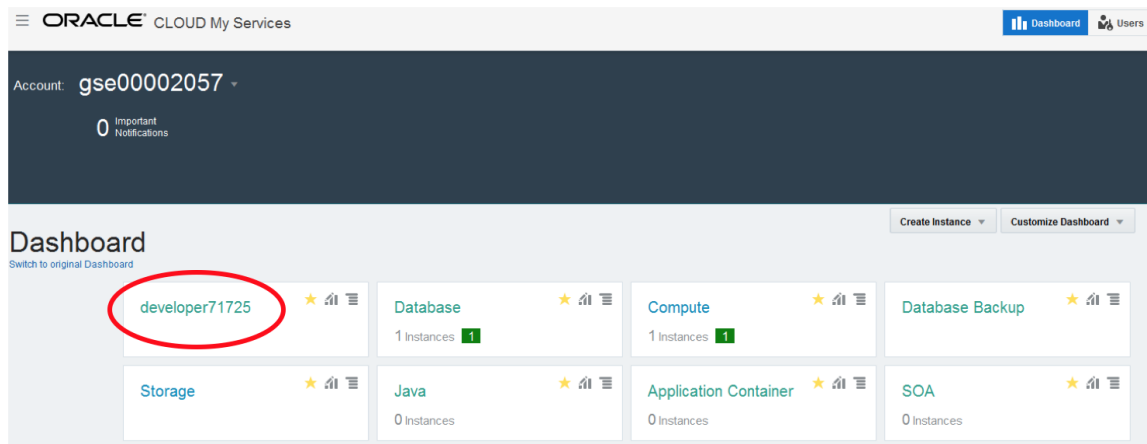


- 이전에 문서화된 단계에 따라 URL 로 이동합니다. <http://cloud.oracle.com> 을 참조하십시오. 창 오른쪽 상단 모서리에 있는 Sign In(로그인)을 클릭합니다. 올바른 데이터 센터를 선택하고 내 서비스 버튼을 클릭하고 올바른 ID 도메인을 입력하고 Go(이동)를 클릭합니다..
- 사용자 이름을 입력하고 올바른 암호를 입력합니다. Sign In(로그인)을 클릭합니다.



Oracle Cloud Sign In screen. It shows a welcome message for user 'gse00002057' with a 'change domain' link and a help icon. Below is a text input field containing 'lisa.jones' and a password input field with 12 dots. A link 'Can't access your account?' is present. At the bottom is a blue 'Sign In' button.

- 대시 보드가 표시되면 개발자 클라우드 서비스를 클릭합니다.



- 개발자 클라우드 서비스 대시 보드에서 **Open Service Console** 버튼을 클릭합니다.

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>

- 트위터 피드 마케팅 프로젝트를 선택하세요

Member Favorites Owner All + New Project

Filter Projects

Twitter Feed Marketing Project
Project to gather and analyze twitter data

STEP 22: 병합 요청

- **Merge Requests** 병합 요청을 클릭합니다. **Assigned to Me**(할당된 위치)를 선택합니다. 검색이 완료되면 **Merge Feature 4 into master** 를 클릭하여 할당된 요청에 따라 선택합니다.

Project Code Maven Snippets **Merge Requests** Issues Agile Build

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)

- 요청이 로드되면 **Changed Files** 탭을 선택합니다. "리사"는 이제 Branch 의 Approving, Rejecting 또는 Merging 의 작업을 하기 전에, 변경 사항을 검토하고 코멘트를 작성하고 더 많은 정보를 요청할 기회를 갖게 될 것입니다.

< 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

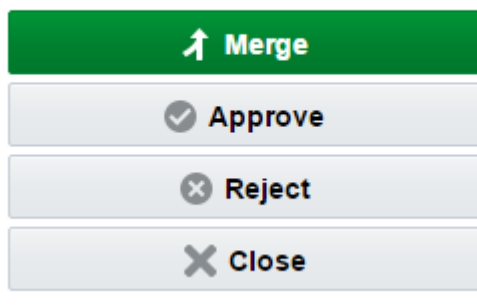
+43 graphics.js doc_root/js/viewModels

```

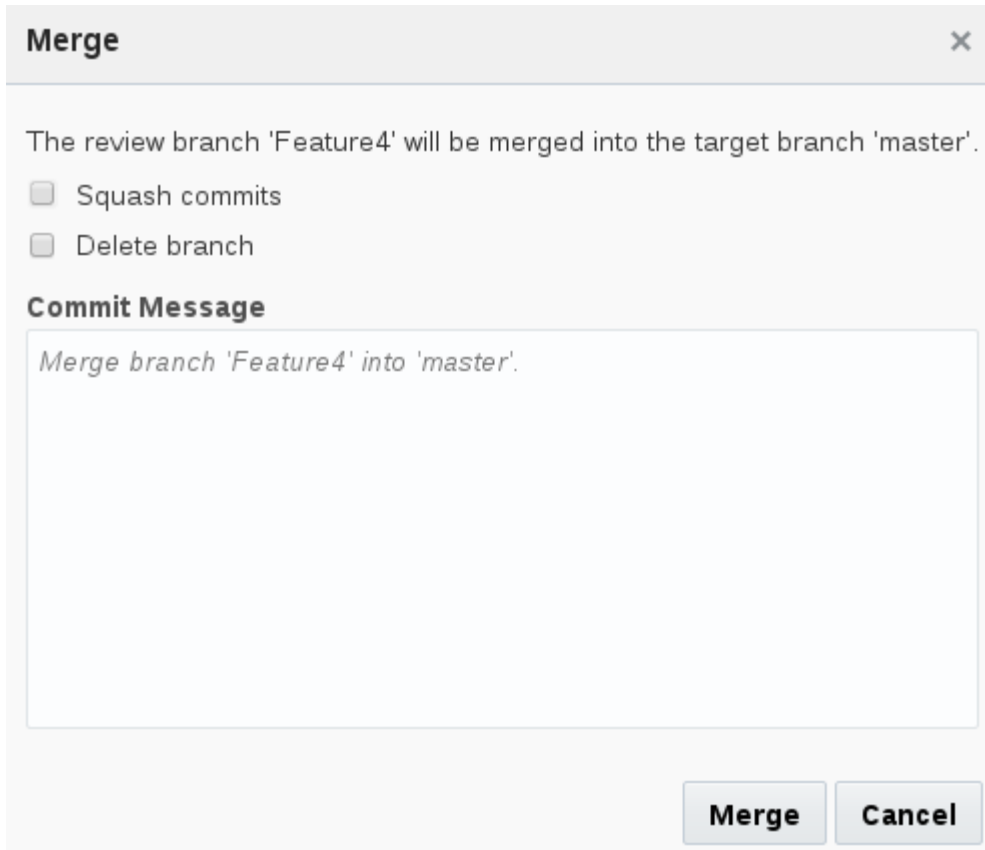
... 6 @@ -6,4 +6,47 @@
6 6 * Your viewModel code goes here
7 7 */
8 8
9 9 /*global $, define, console*/
10 10 /*jslint sloppy:true*/
11 11
12 12 define(['ojs/ojcore', 'knockout', 'ojs/ojtable'], function (oj, ko) {
13 13
14 14     function mainContentViewModel() {
15 15
16 16         // change this root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

```

- **Merge**(병합)버튼을 클릭합니다.



- 기본 값을 그대로 두고 확인 대화 상자에서 **Merge**(병합) 버튼을 클릭합니다.



Merge [X]

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

- 이제 코드가 개발자 클라우드 서비스 저장소에 커밋 되었으므로 build 와 배포가 자동으로 시작됩니다. 빌드를 클릭하면 큐에 **Twitter Marketing UI Build** 가 표시됩니다.

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

- 빌드가 완료될 때까지 1~2 분 정도 걸립니다. Build 가 완료되면 **Last Success** 가 **Just Now** 로 변경될 것입니다.

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

- **Deploy**(배포)를 클릭합니다. 배포 상태가 **Deployment update in progress** 로 변경될 때까지 기다린 후, **Last deployment succeeded – Just now** 로 바뀝니다.

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](#)

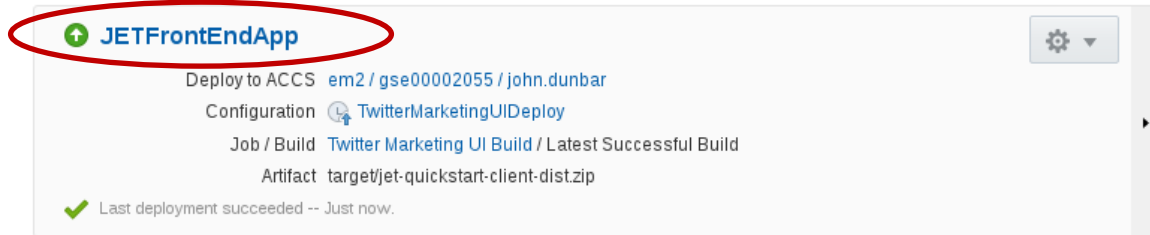
Job / Build [Twitter Marketing UI Build](#) / Latest Successful Build

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

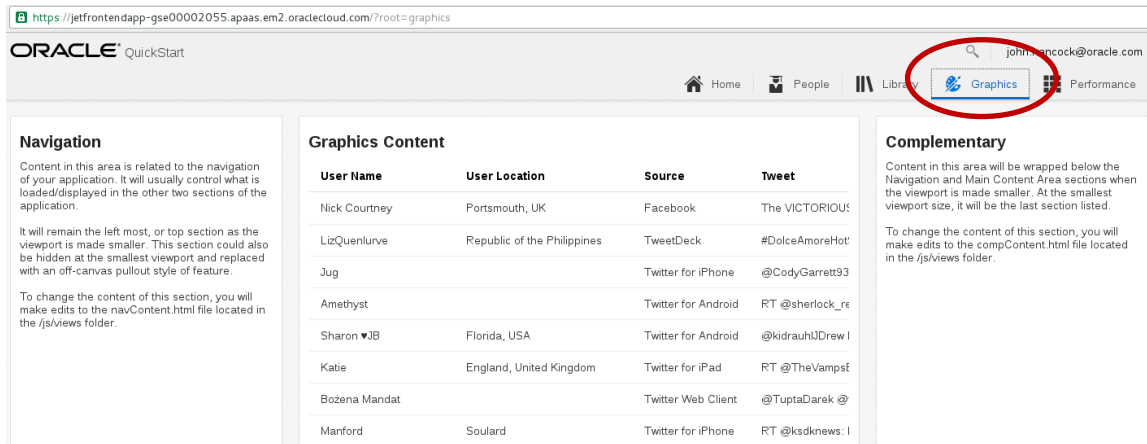
Last deployment succeeded -- Just now

클라우드에서 JETFrontEndAPP UI 테스트

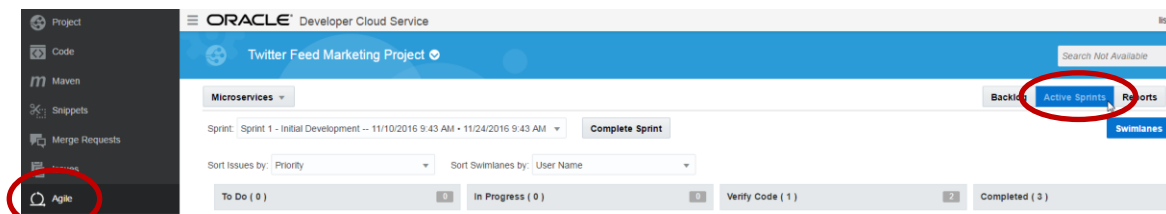
- 서비스가 성공적으로 전개되면 JETFrontEndApp 링크를 클릭합니다.



- 새 브라우저 탭이 로드되면 **Graphics** 을 클릭하여 트위터 피드 데이터를 표시합니다.



- 스프린트 기능을 완성하려면, Twitter 피드 마케팅 프로젝트 대시 보드에서 **Agile** 를 클릭하십시오. 그런 다음 Active Sprints 버튼을 클릭합니다.



- **Verify** 에서 **Complete** 로 Feature4 를 Drag and Drop 에서 완료시킵니다.

The screenshot shows a task management interface with two columns: 'Verify Code (1)' and 'Completed (3)'. Below these columns are several task cards. A red arrow indicates the movement of 'Feature 4' from the 'Verify Code' column to the 'Completed' column. The task cards include details such as the task name, description, and status (e.g., 'RESOLVED : FIXED' or 'VERIFIED : FIXED').

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

The 'Change Progress' dialog box displays the following information:

- Change:** Feature 4
- From:** RESOLVED - FIXED
- To:** VERIFIED - FIXED (selected from a dropdown menu)

Buttons: OK, Cancel

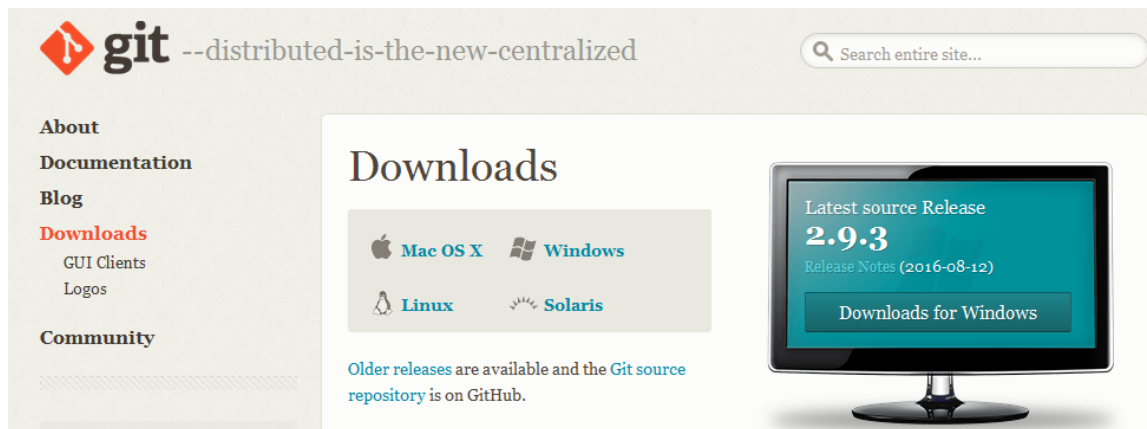
- 이제 다음 lab 로 이동할 준비가 되었습니다.

부록 1-Brackets 및 Git 을 설치하는 부속 문서

Git 다운로드 및 설치

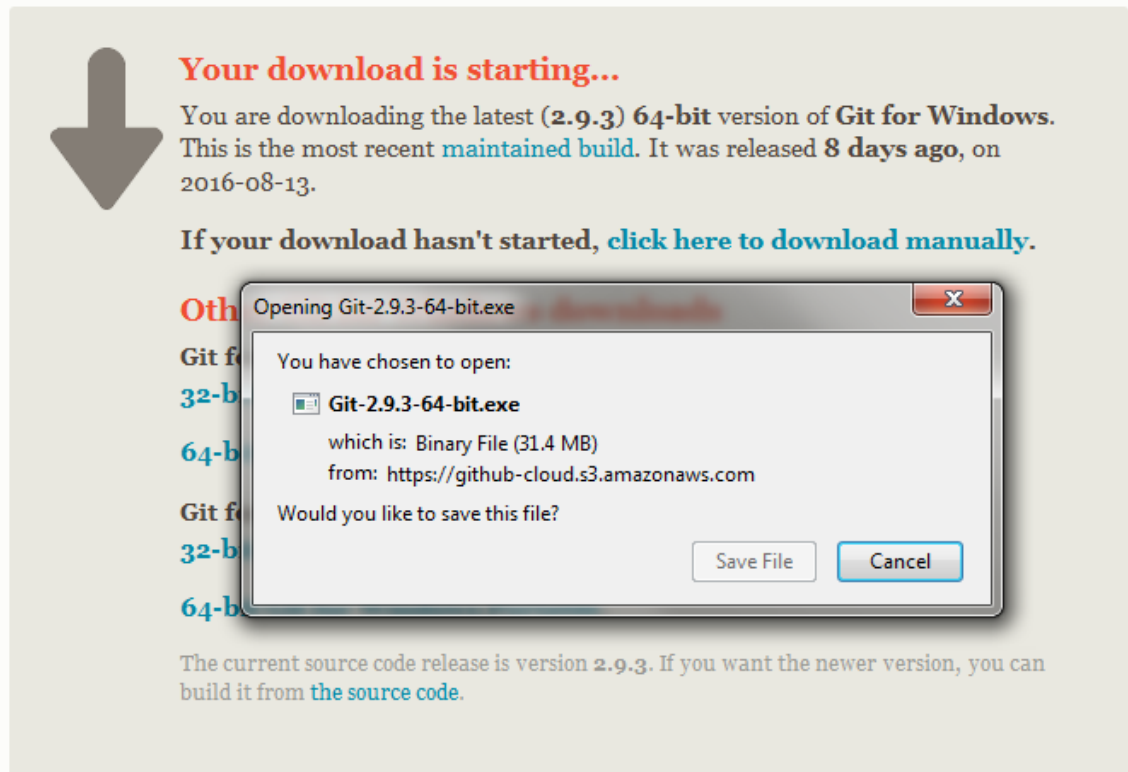
STEP 23: Git 다운로드

☐ 다음 URL 로 이동합니다. <https://git-scm.com/downloads>



- OS 를 선택합니다. 예를 들어 Windows 에서 설치하는 방법을 보여 드리겠습니다.
Windows 다운로드를 클릭하고 SaveFile(파일 저장)을 클릭합니다.

Downloading Git



Your download is starting...

You are downloading the latest **(2.9.3) 64-bit** version of **Git for Windows**. This is the most recent **maintained build**. It was released **8 days ago**, on 2016-08-13.

If your download hasn't started, [click here to download manually](#).


Other Downloads

Git for Windows
32-bit
64-bit
Git for Linux
32-bit
64-bit

The current source code release is version **2.9.3**. If you want the newer version, you can build it from [the source code](#).

Opening Git-2.9.3-64-bit.exe

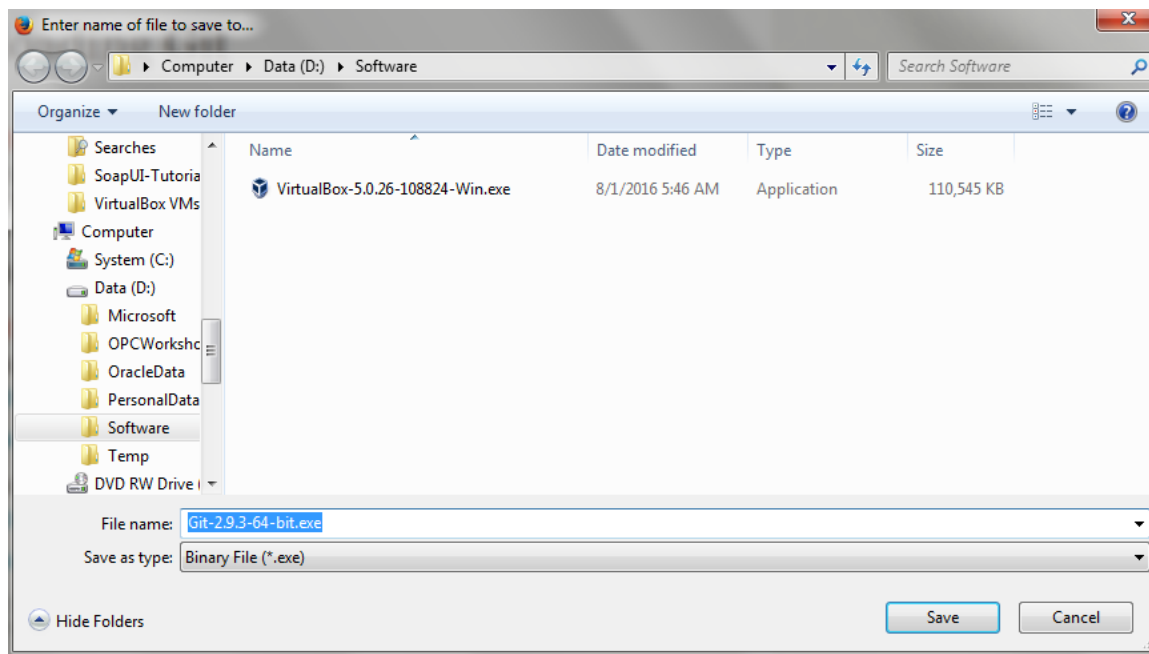
You have chosen to open:

 **Git-2.9.3-64-bit.exe**
which is: Binary File (31.4 MB)
from: <https://github-cloud.s3.amazonaws.com>

Would you like to save this file?

Save File Cancel

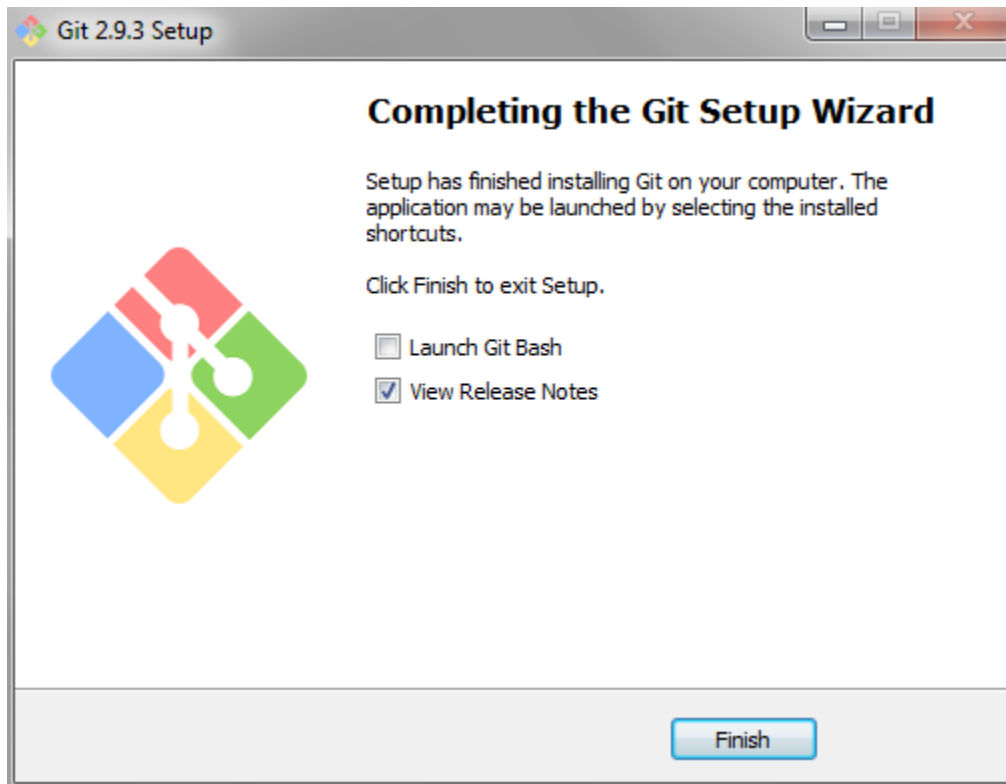
- ☐ 다운로드 위치를 선택하고 Save(저장)을 클릭합니다. 우리는 D:\Software 를 사용할 것이다.



- ☐ Windows 탐색기를 열고 Git executable 를 다운로드한 곳을 탐색합니다. Git 실행에 대한 두 번 클릭하여 설치 과정을 시작하십시오.

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

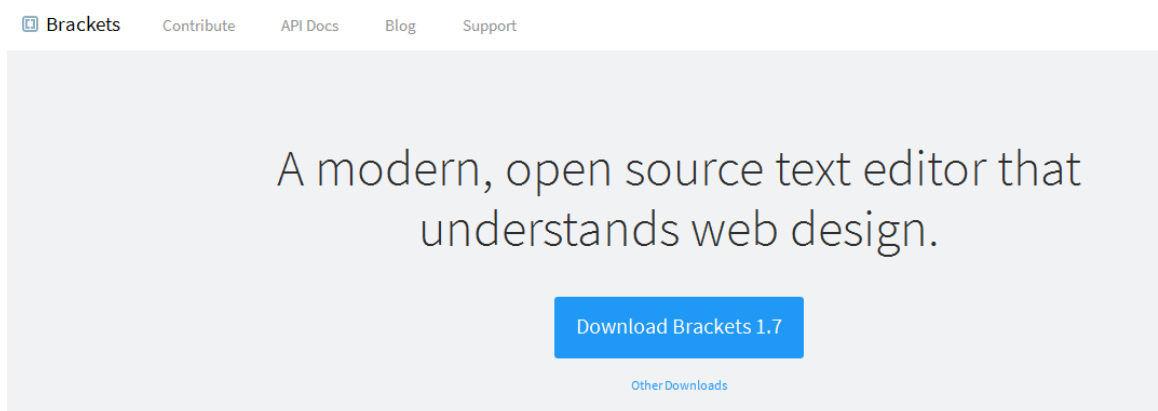
□ 설치 과정을 거칩니다.



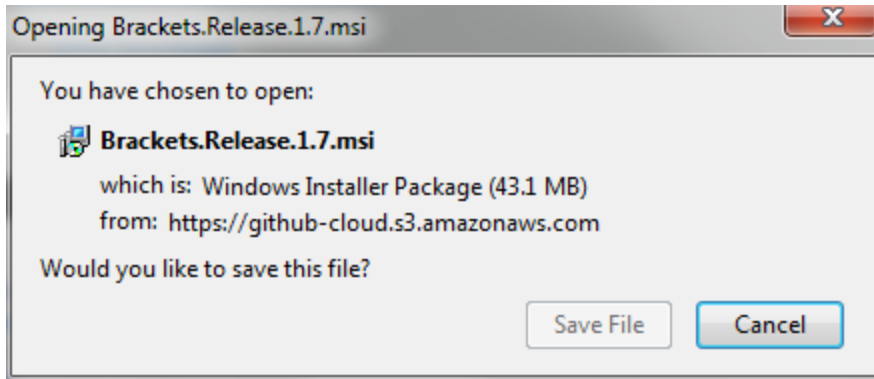
Brackets 텍스트 편집기 다운로드 및 설치

STEP 24: 다운로드 Brackets 텍스트 편집기

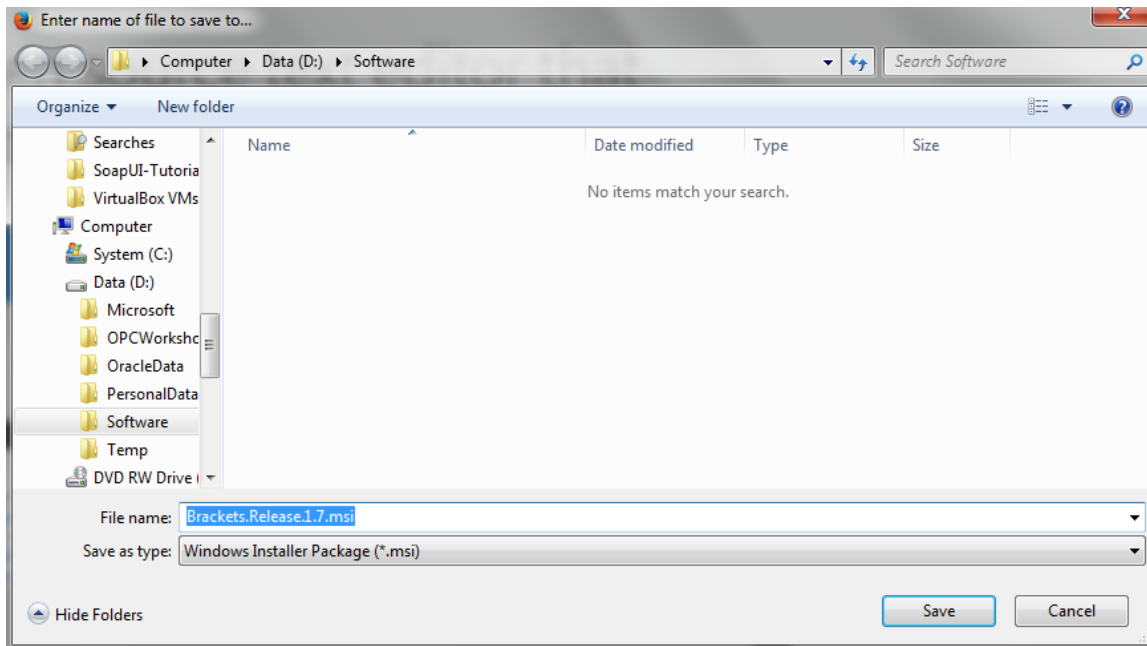
□ 다음 URL 로 이동합니다. <http://brackets.io>



- 다운로드 Brackets1.7 다운로드 클릭 후 파일 저장 클릭



- 다운로드 위치를 선택하고 Save(저장)을 클릭합니다. 우리는 D:\Software 를 사용할 것이다.



- Windows 탐색기를 열고 다운로드된 Brackets 을 탐색합니다. Git 실행에 대한 두 번 클릭하여 설치 과정을 시작하십시오.

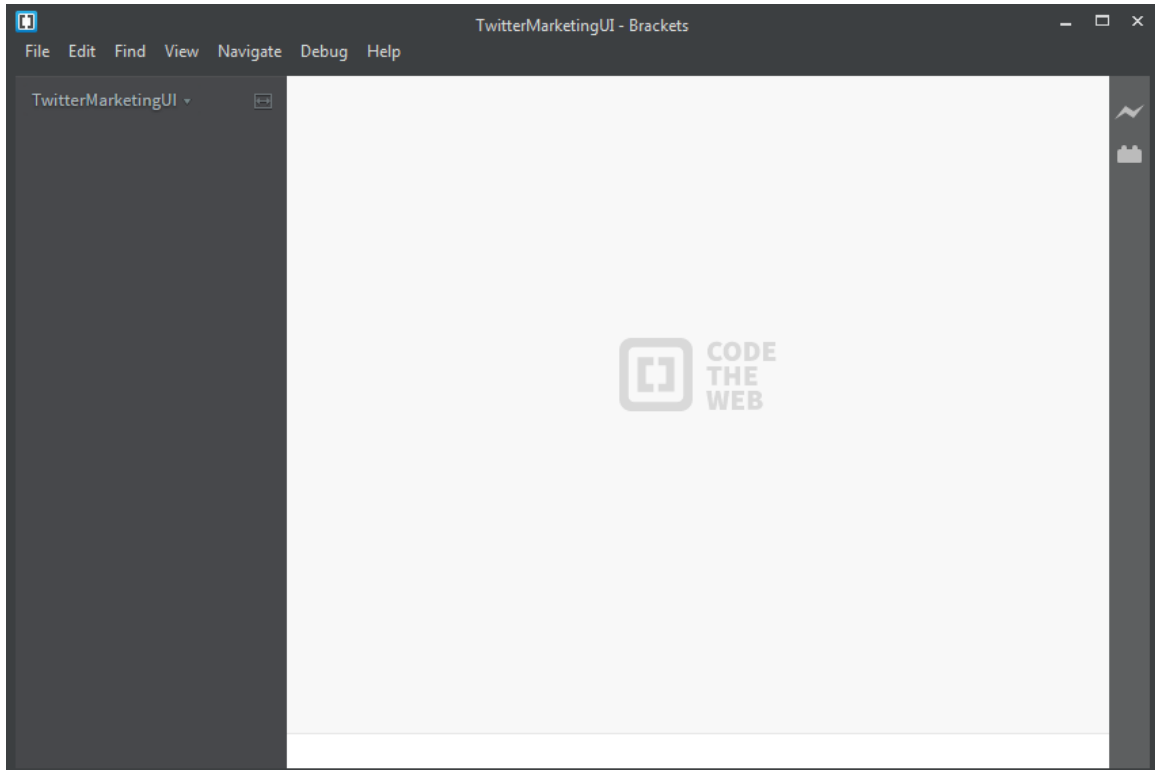
Name	Date modified	Type	Size
Brackets.Release.1.7.msi	8/21/2016 7:44 AM	Windows Installer ...	44,184 KB
Git-2.9.3-64-bit.exe	8/21/2016 7:06 AM	Application	32,120 KB
VirtualBox-5.0.26-108824-Win.exe	8/1/2016 5:46 AM	Application	110,545 KB

☐ 설치 과정을 거칩니다.

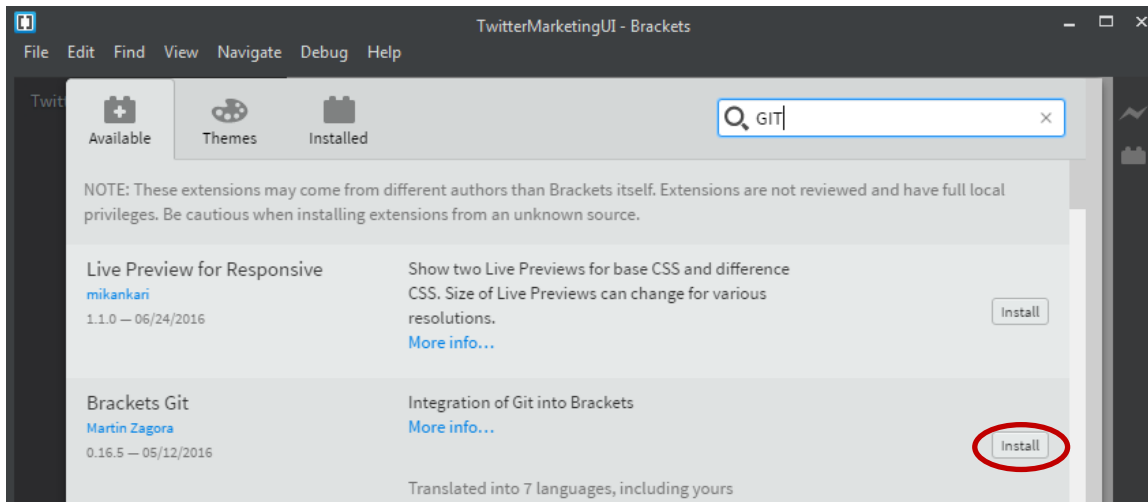
Brackets 시작 및 Git 구성

STEP 25: Brackets 시작 및 Git 구성

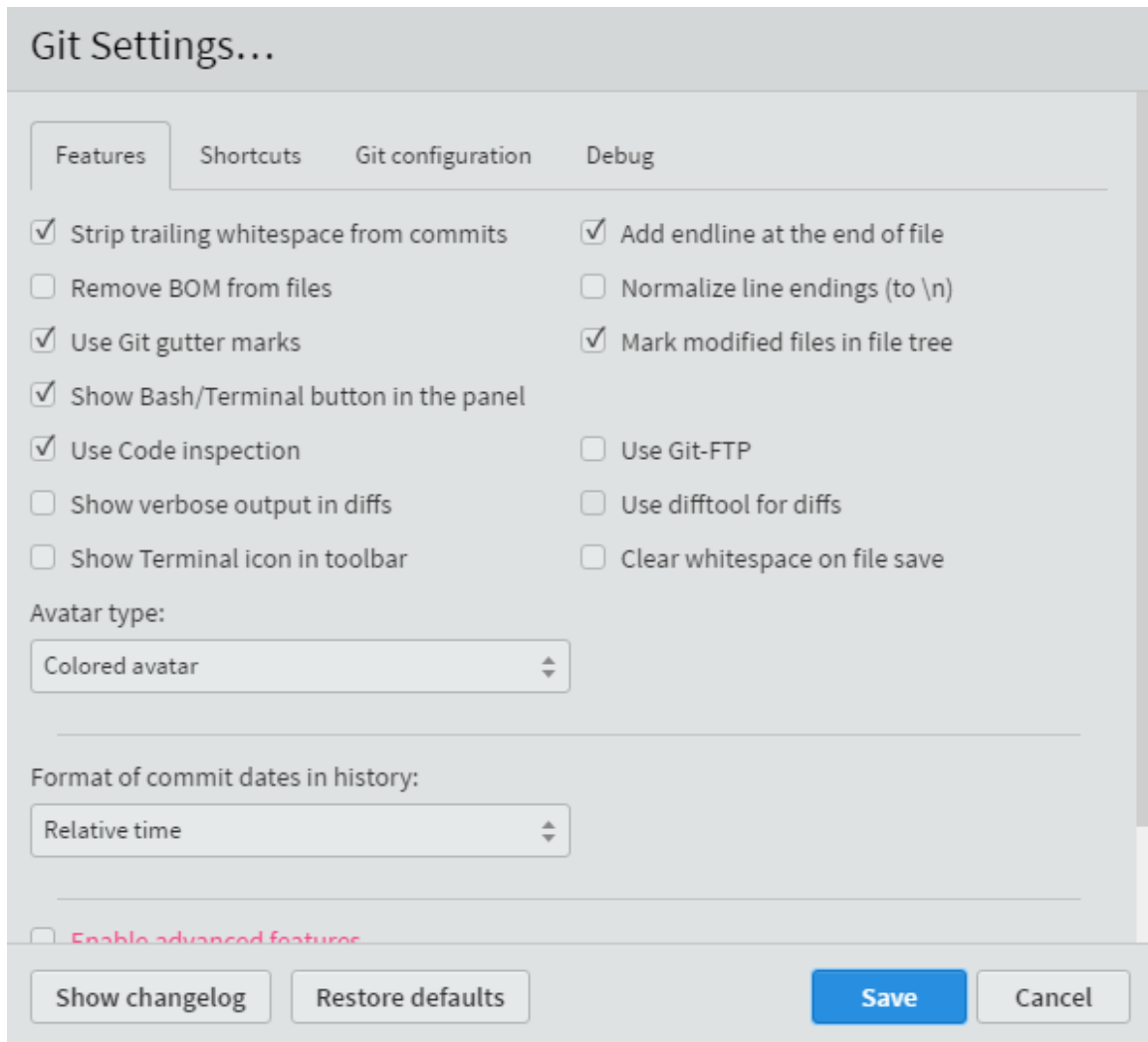
- ☐ **TwitterMarketingUI** 디렉토리를 생성합니다. **Windows** 탐색기에서 **TwitterMarketingUI** 탐색, 오른쪽 클릭 및 **Open as Brackets Project** 선택합니다.



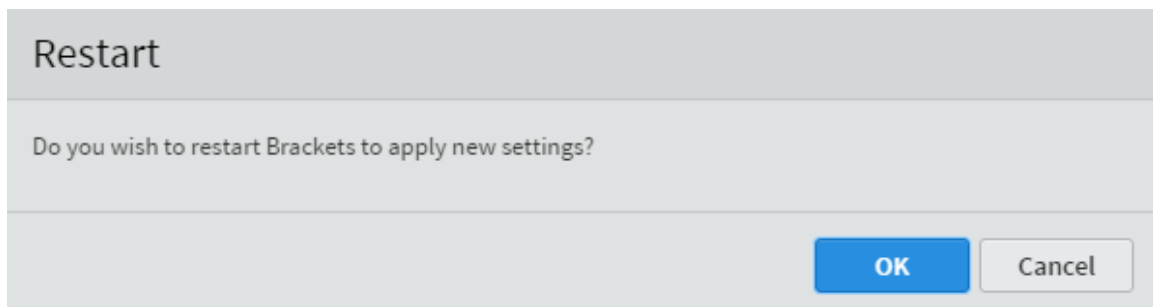
- ☐ **File-> Extension Manager** 를 선택합니다.
- ☐ 검색 창에 **GIT** 를 입력합니다. Brackets Git 를 위해 **Install** 클릭합니다.



- ☐ 설치를 완료한 후에는 기본 설정에 대한 기본 설정을 그대로 두고 Save(저장)을 클릭합니다.



- ☐ 확인을 클릭하여 Brackets 을 다시 시작합니다.



□ 이제 우측 패널에서 Git 아이콘이 보입니다.

