# **Oracle Cloud Day- Application Development Hands-On Workshop**

## **DevOps on Oracle Infra:**

## Lab Overview

In this lab you’ll learn how to leverage Oracle Developer Cloud Service to help a development team manage their agile development process and automate their DevOps cycle on Oracle Cloud infrastructure (Oracle Kubernetes Engine).

How to create demo environment with all needed components:   
(Click Ctrl+links below)

* [Setup Developer Cloud Service Instance](https://docs.oracle.com/en/cloud/paas/developer-cloud/csdcs/service-setup.html#GUID-8EE9FC19-70A0-4508-A6B1-FB8425C13A91)
* [Create OKE cluster](https://www.oracle.com/webfolder/technetwork/tutorials/obe/oci/oke-full/index.html)

# Managing Projects with Developer Cloud Service

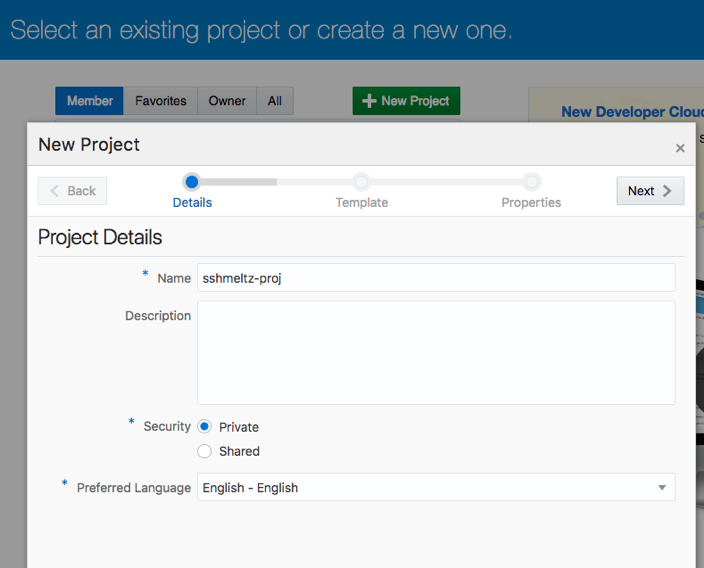
## Creating a DevCS Project

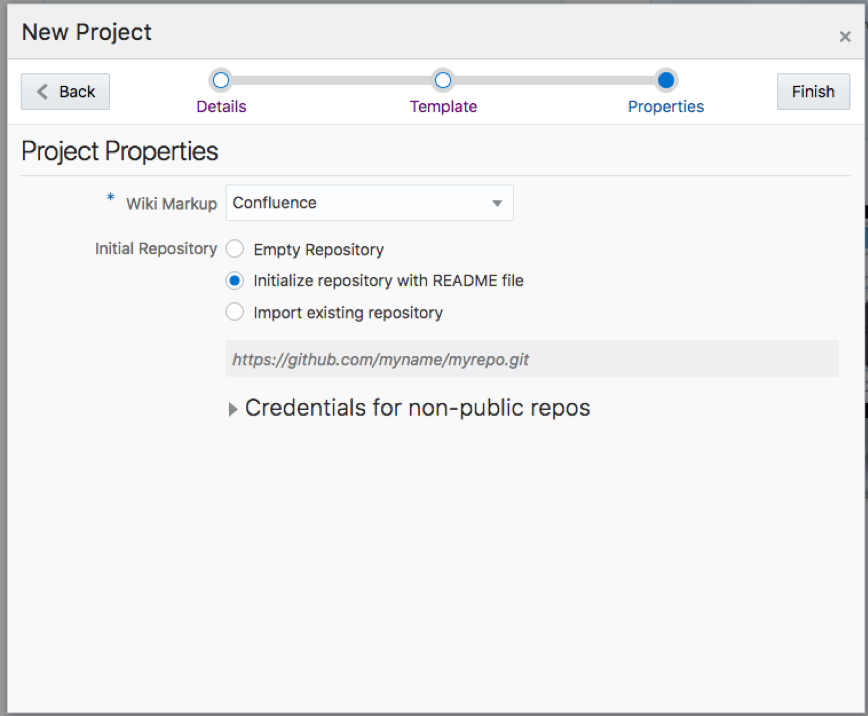
In this lab you’ll create a DevCS project for your team and learn about some of the basic tasks of a project administrator.

### Creating a Project

1. Using your cloud user – login into your cloud account.
2. In my services dashboard locate Developer (if not visible click customize dashboard and make sure developer is in show mode).
3. Open the service console.
4. Create a new project
5. Choose the following options in the dialog

* Private project
* Initial Repository
* Confluence syntax for your wiki

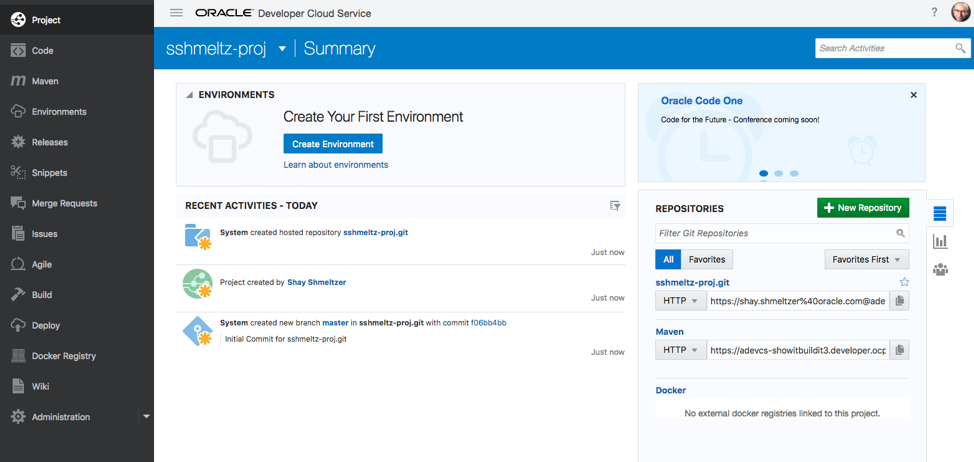
[](https://github.com/PrasannaShasthri/eca-draft/blob/master/tutorials/DevcsImages/Picture1.png)

[](https://github.com/PrasannaShasthri/eca-draft/blob/master/tutorials/DevcsImages/Picture2.png)

**GitHub repo to clone:**

<https://github.com/saiyam1814/oraclecloudday>

1. Click Finish to create your project. Once the project is created you’ll be taken into the project’s home page.

[](https://github.com/PrasannaShasthri/eca-draft/blob/master/tutorials/DevcsImages/Picture3.png)

1. Generate and upload the API signing key using the below link:

<https://docs.cloud.oracle.com/iaas/Content/API/Concepts/apisigningkey.htm#How>

1. Create the auth token for the user.

a. In the top-right corner of the Console, open the **User** menu (**User menu icon**) and then click **User Settings** to view the details.

b. On the **Auth Tokens** page, click **Generate Token**.

c. Enter a friendly description for the auth token. Avoid entering confidential information.

d. Click **Generate Token**. The new auth token is displayed.

e. Copy the auth token immediately to a secure location from where you can retrieve it later, because you won't see the auth token again in the Console.

f. Close the Generate Token dialog.

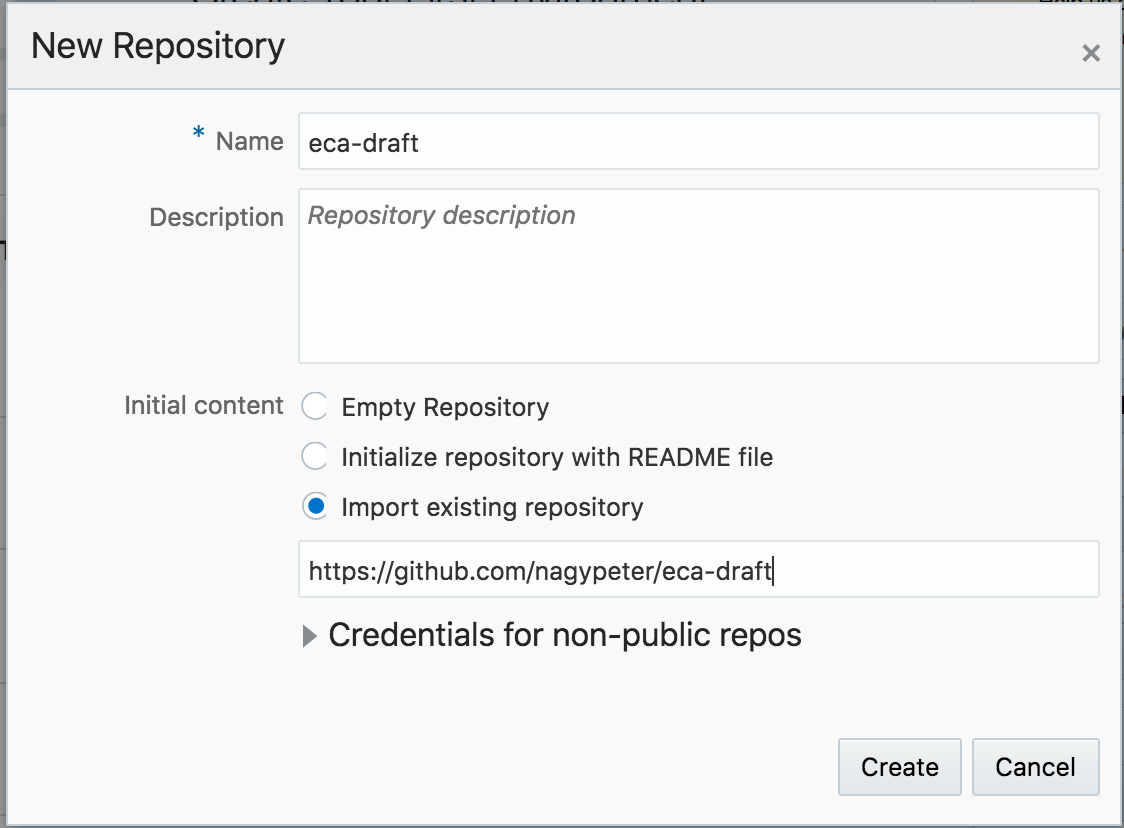
# Adding a Git Repository

A DevCS Project can have multiple git repositories. We are going to add a new repostiroy copying the content of an existing repository on GitHub.

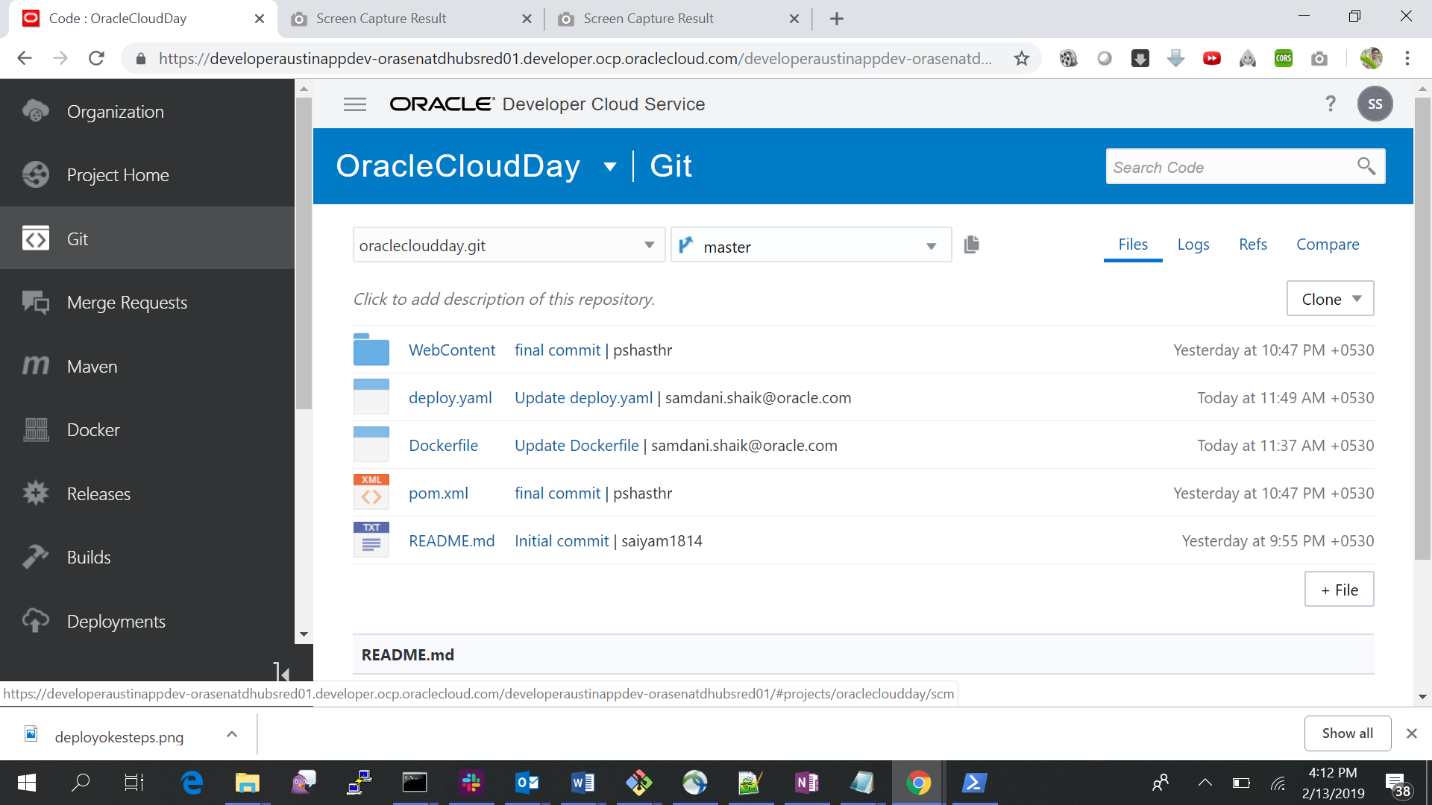
1. In the left menu choose Administration and then the Repositories section
2. Click to add a New Hosted Repository
3. Name the repository and choose to import existing repository –

use this URL: <https://github.com/saiyam1814/oraclecloudday>

Click Create.

[](https://github.com/PrasannaShasthri/eca-draft/blob/master/tutorials/DevcsImages/Picture7.png)

1. In the left menu choose the Code section and you'll see all the code you need for the next labs in your project.
2. You can click on any of the files to see the content and even edit it directly from your browser.
3. From the web interface you can also branch the code when needed.



Replace the <DistributionManagement> tag contents in pom.xml with your specific Maven configuration mentioned in the Maven section of the DevCS project.

### Create build job to package service Build-OKE in container and push to container registry

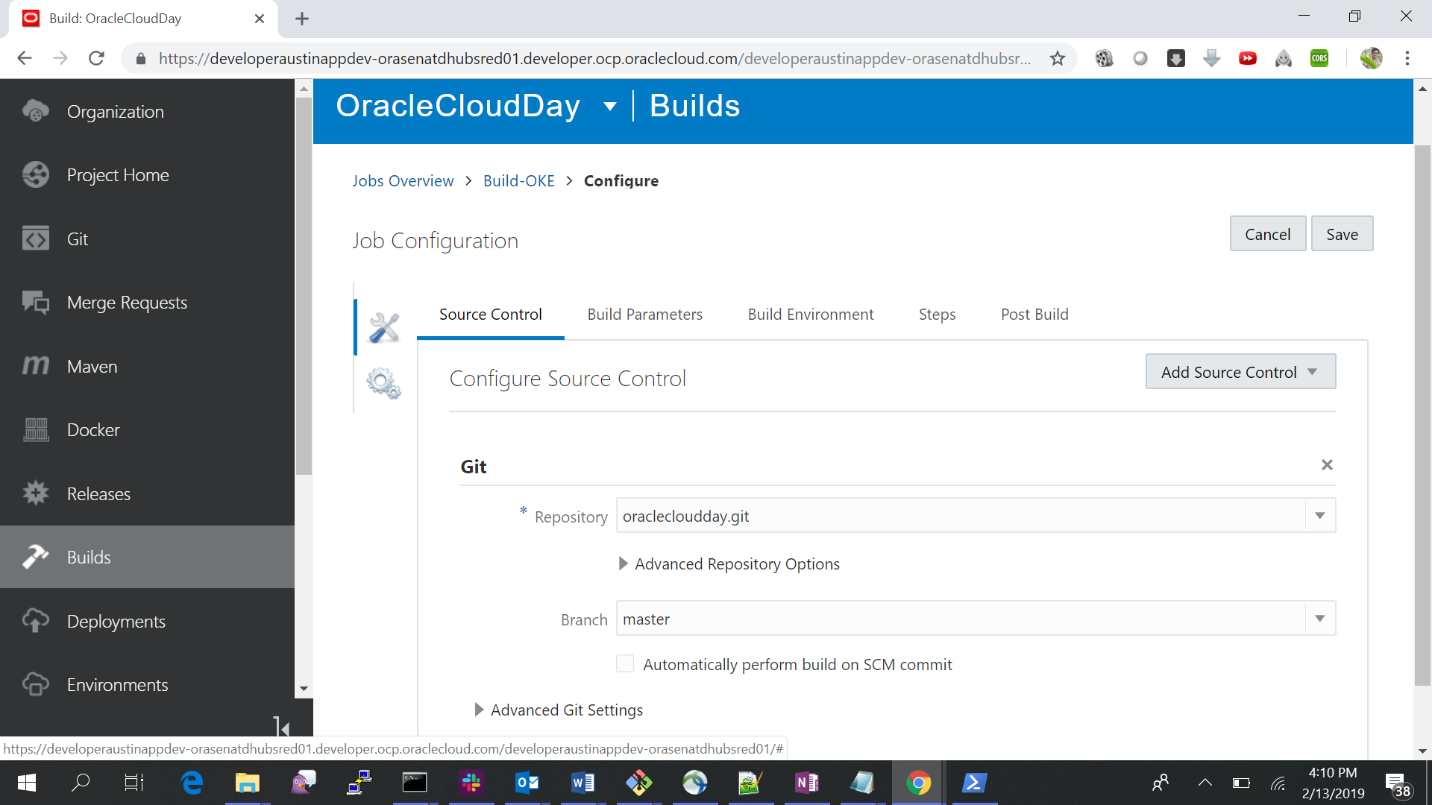
Open your DevCS project and from the left navbar choose Build. Click +New Job to create a build job.

Fill out the following:

* Job Name: Build-OKE
* Description: build and store service
* Create New: yes
* Software Template: template available for the workshop

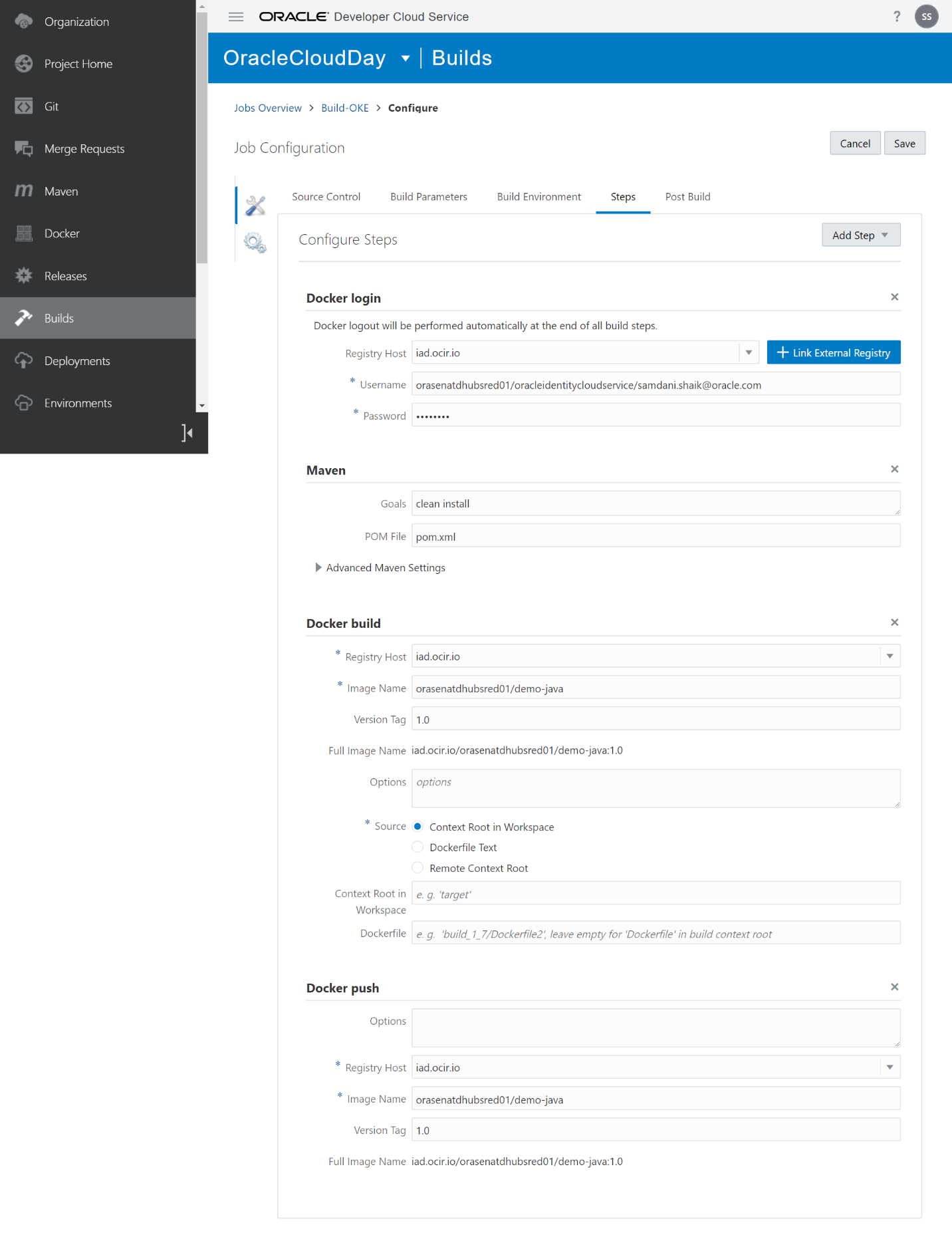
Click Create Job.

The build job configuration opens. On the first Source Control tab click Add Source Control and select Git. From the Repository dropdown list select the source code repository.



Define the build steps in “Steps”. Add your first step by clicking on Add multiple Step buttons in the following order.

1. Docker login.
2. Maven
3. Docker Build
4. Docker Push



Click Save and Build the job. It will show the success log.

### Create deploy job Deploy-OKE, to deploy the dockerized app to the OKE cluster.

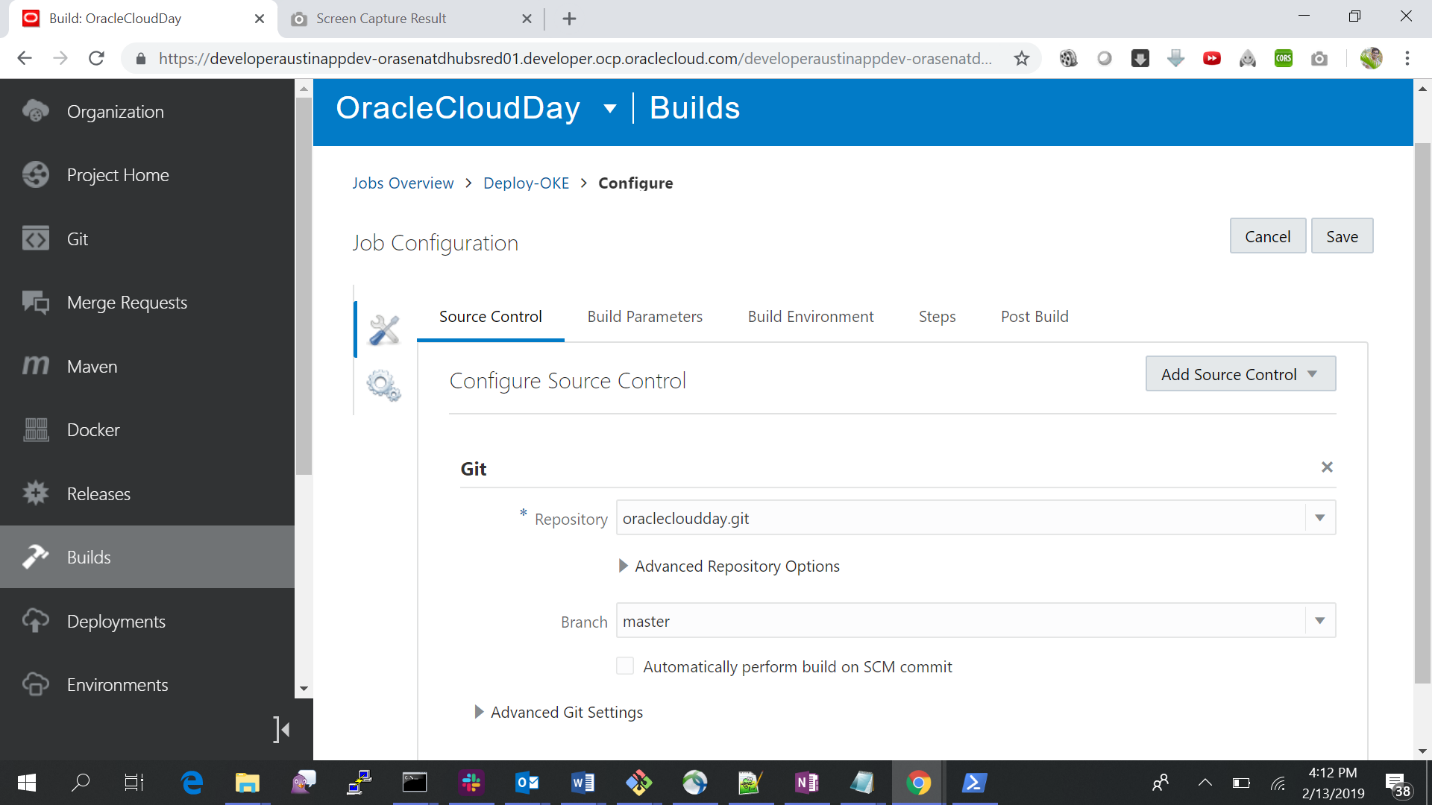
Open your DevCS project and from the left navbar choose Build. Click +New Job to create a build job.

Fill out the following:

* Job Name: Deploy-OKE
* Description: deploy service
* Create New: yes
* Software Template: template available for the workshop

Click Create Job.

The build job configuration opens. On the first Source Control tab click Add Source Control and select Git. From the Repository dropdown list select the source code repository.



Define the build steps in “Steps”. Add your first step by clicking on Add multiple Step buttons in the following order.

1. OCIcli

**2. Unix Shell :**

mkdir -p $HOME/.kube

export ENDPOINT="containerengine.us-ashburn-1.oraclecloud.com"

oci ce cluster create-kubeconfig --cluster-id <CLUSTER-ID> --file $HOME/.kube/config --region us-ashburn-1

kubectl get nodes

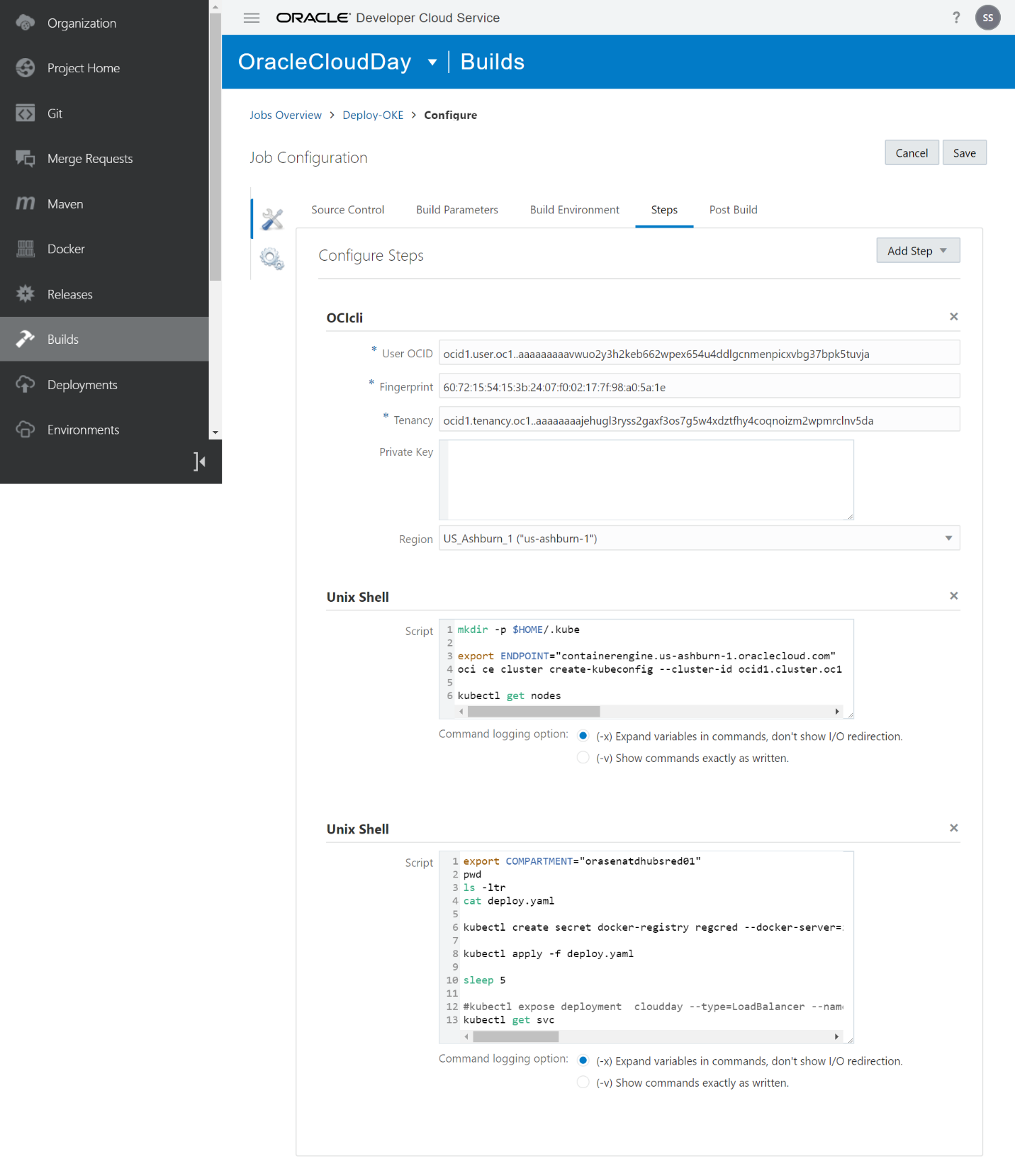
**3. Unix Shell :**

kubectl create secret docker-registry regcred --docker-server=iad.ocir.io --docker-username=<tenancy\_name>/<username> --docker-password='<auth-token>' --docker-email=<Email-ID> || echo 'secret exists'

kubectl apply -f deploy.yaml

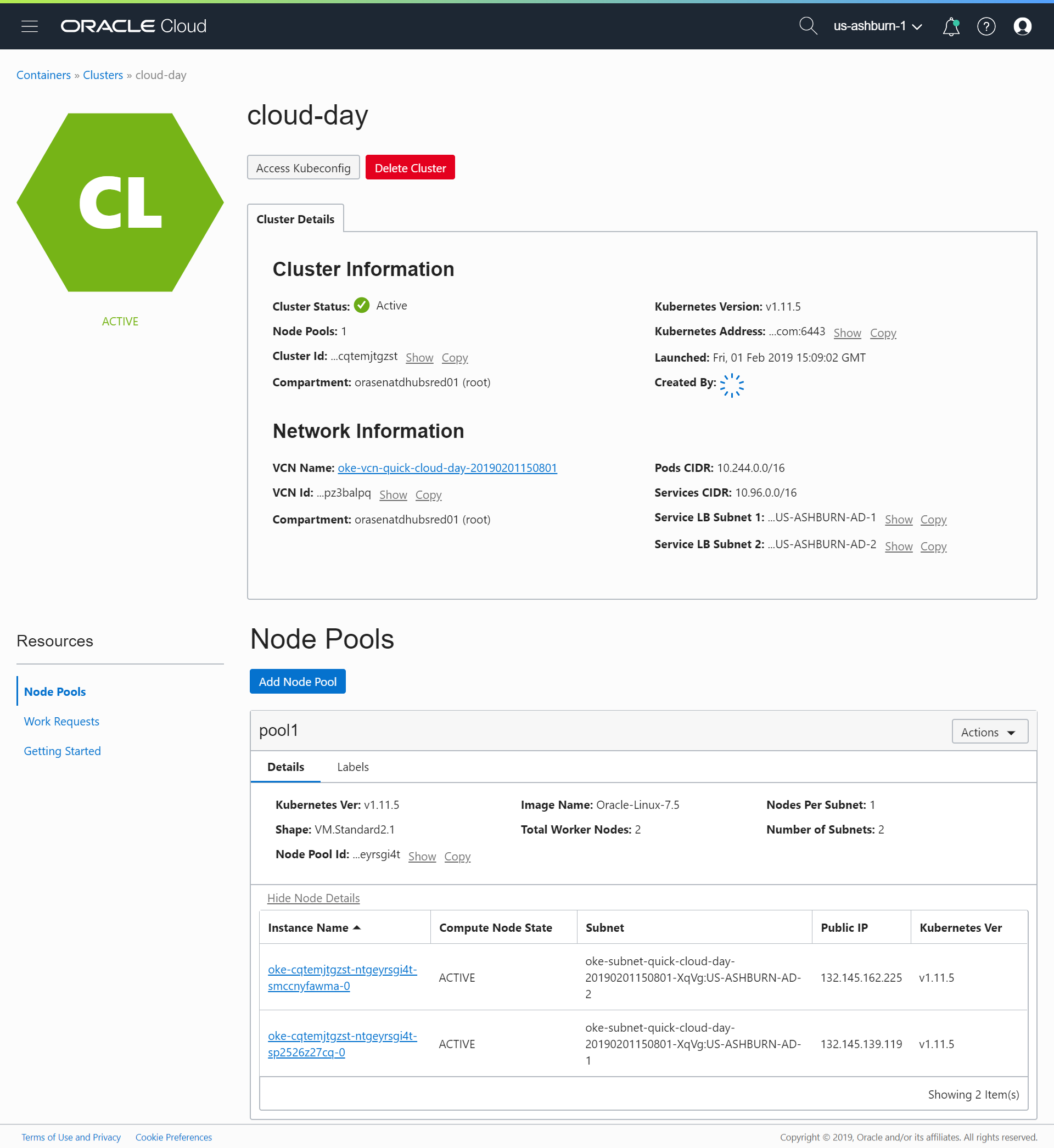
sleep 5

kubectl get svc

Click Save and Build the job. It will show the success log.

# OKE cluster and OCIR

The cluster details and node pools will be listed in the OCI console:



The OCIR image pushed through DevCS build job (demo-java) will be listed under Container Registry:

