Oracle Developer Cloud

# **Create a Cluster with Oracle Cloud Infrastructure Container Engine for Kubernetes.**

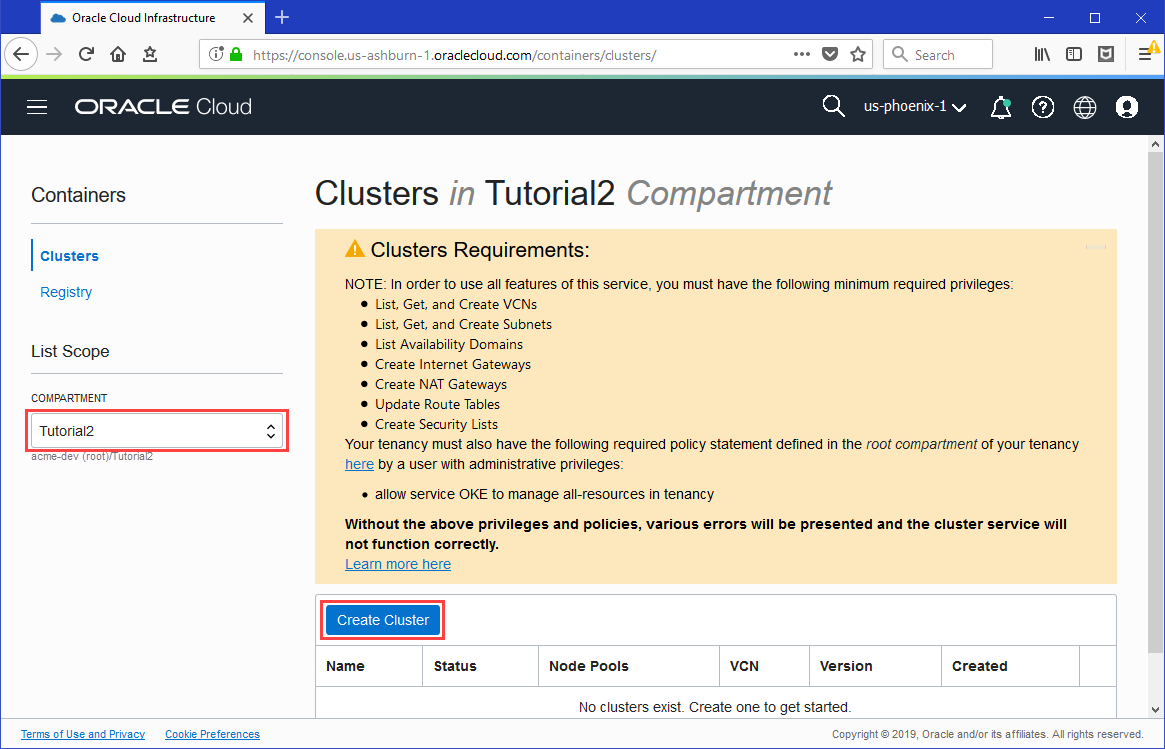
# Reference:

<https://www.oracle.com/webfolder/technetwork/tutorials/obe/oci/oke-full/index.html>

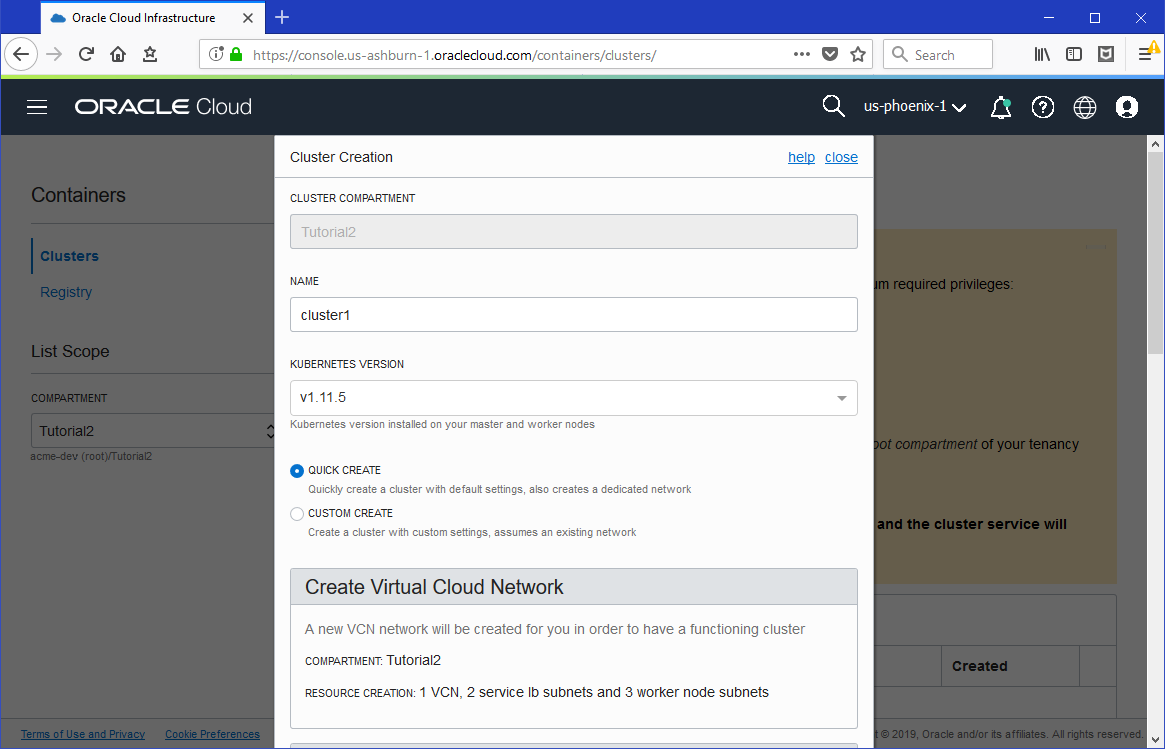
In this tutorial, you use default settings to define a new cluster. When you create the new cluster, new network resources for the cluster are created automatically, along with a node pool and worker nodes.

* Within the root compartment of your tenancy, a policy statement “Allow service OKE to manage all-resources in tenancy” must be defined to give Container Engine for Kubernetes access to resources in the tenancy.
* Within your tenancy, there must already be a compartment to contain the necessary network resources (VCN, subnets, internet gateway, NAT gateway, route table, security lists).
* To create and/or manage clusters, you must belong to the tenancy's “Administrators” group.

1. In the Console, open the navigation menu. Under **Solutions and Platform** go to **Developer Services** and click **Container Clusters**.
2. Choose a **Compartment** that you have permission to work in, and in which you want to create both the new cluster and the associated network resources.

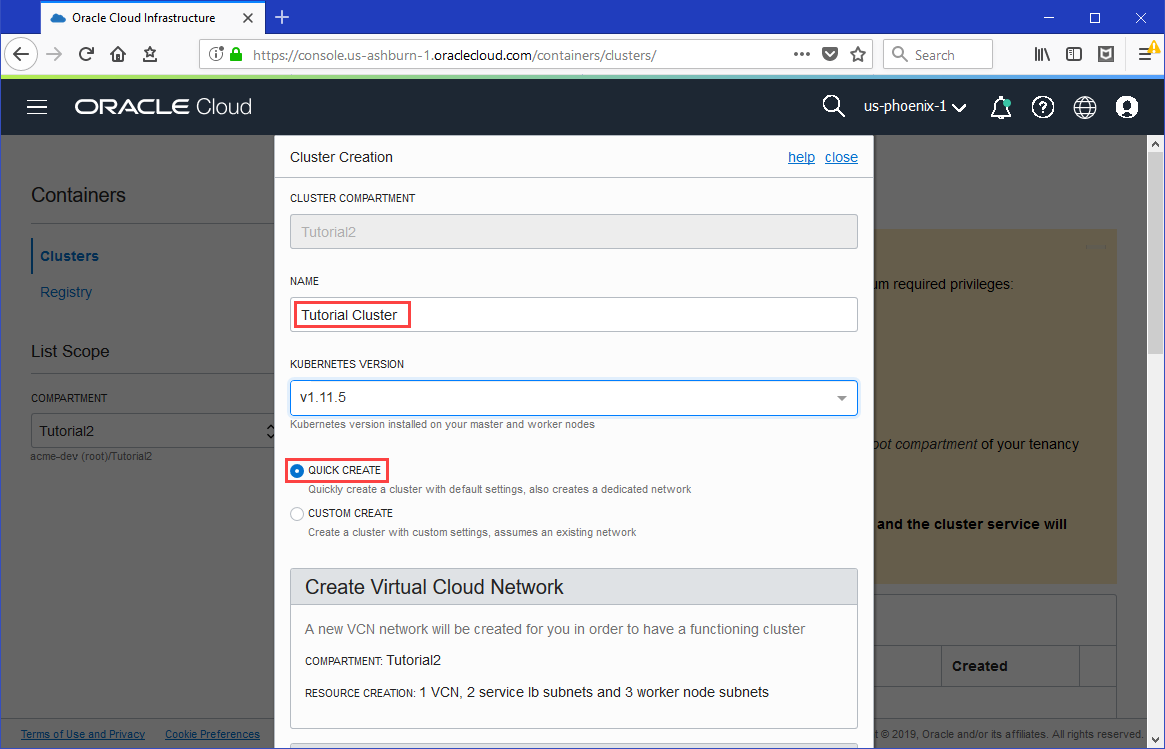


On the **Clusters** page, click **Create Cluster**. The examples shown in this tutorial assume you give the new cluster the name Tutorial Cluster.



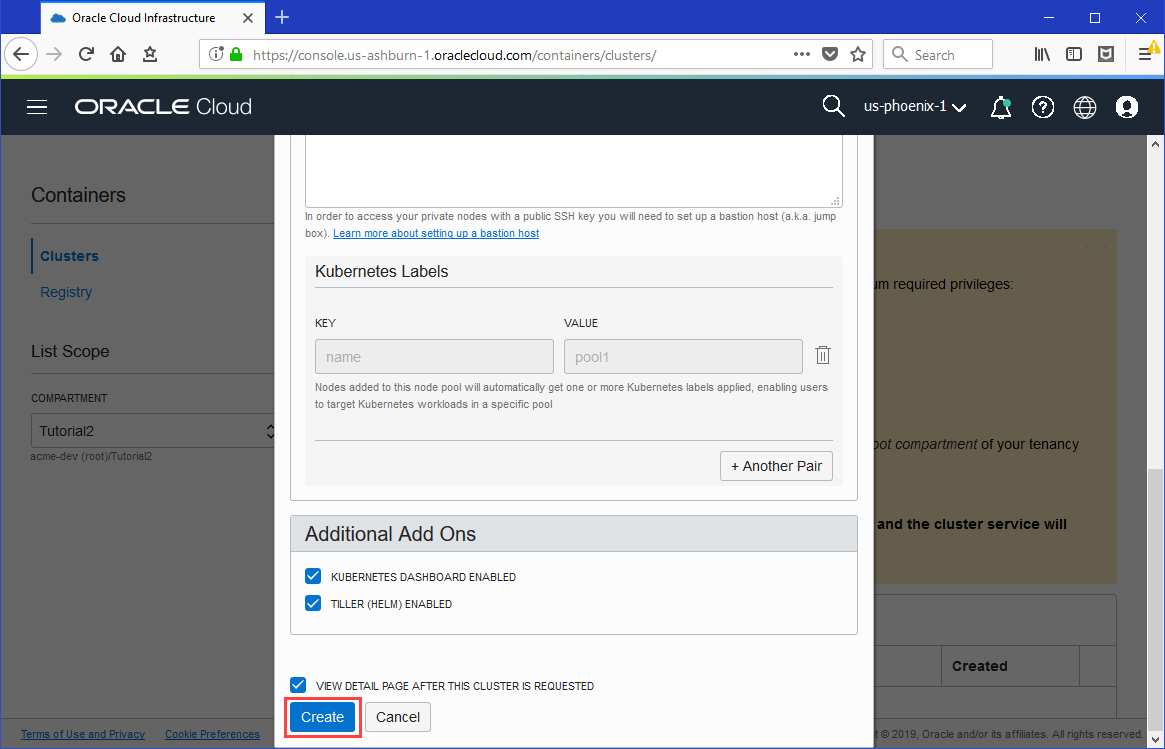
In the **Cluster Creation** dialog, change the placeholder value in the **Name** field

Select the **Quick Create** option to indicate you want to create a new cluster with default settings, along with new network resources for the new cluster.

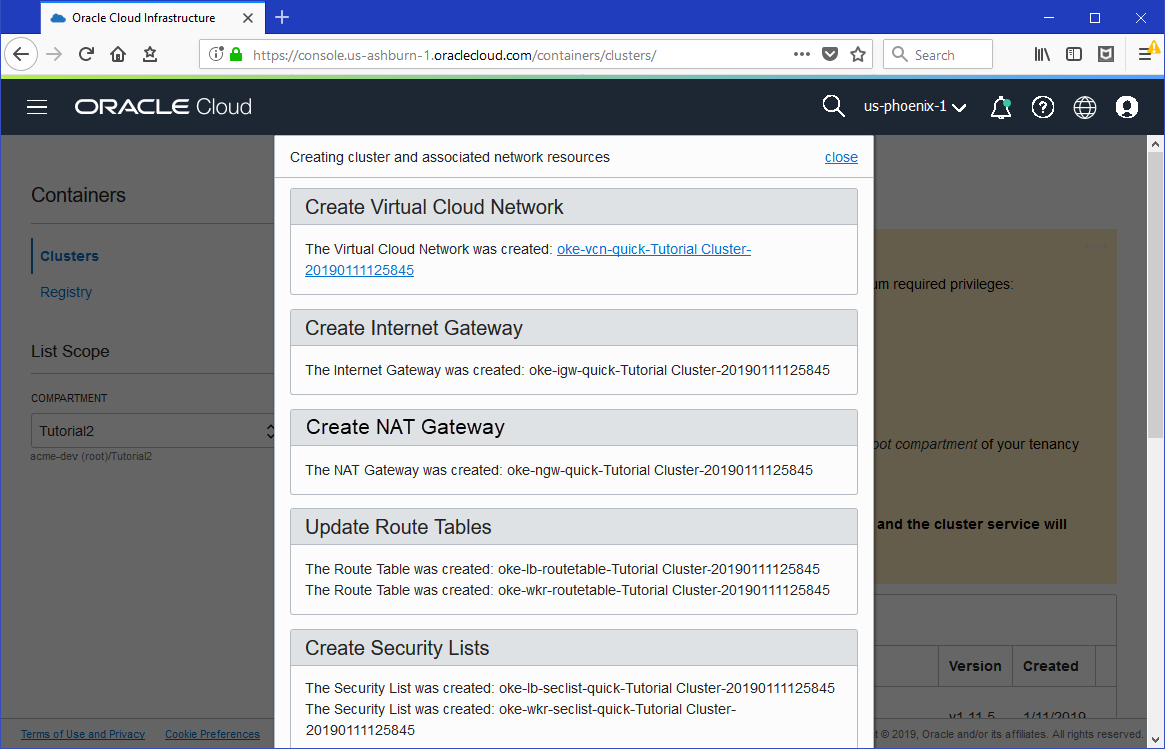


Select to create a public regional subnet to host worker nodes (along with the public regional subnet to host load balancers).

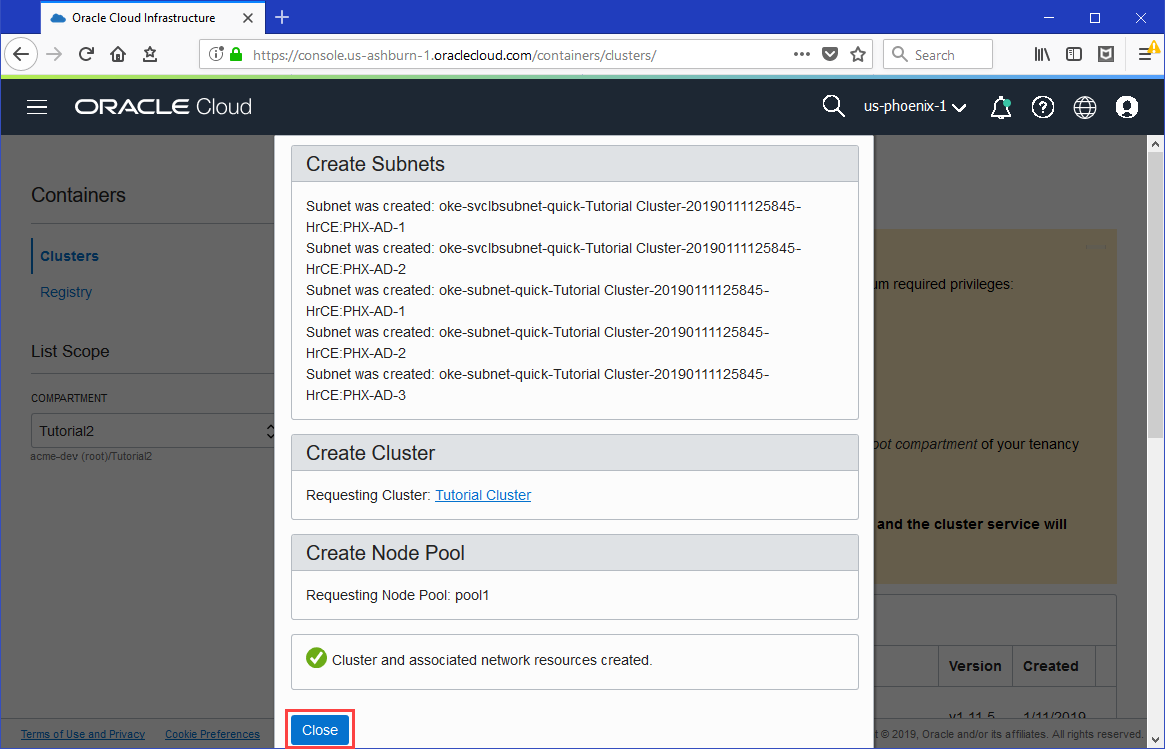
Click **Create** to create the new network resources and the new cluster.



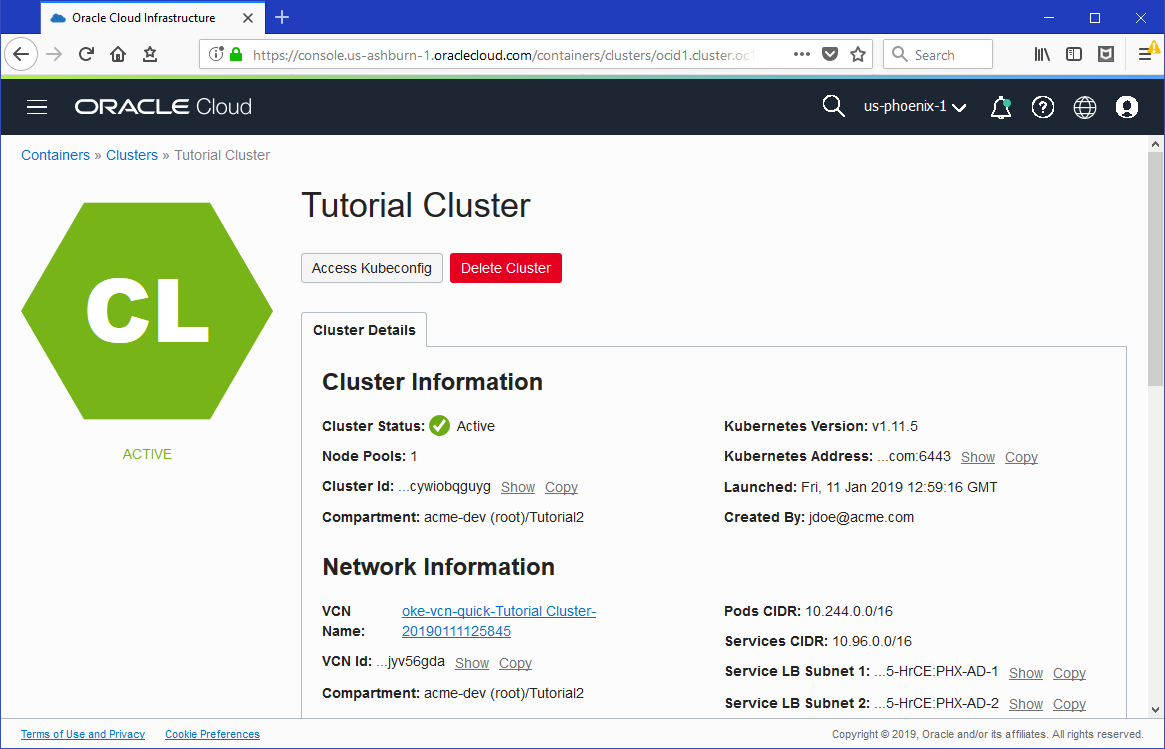
You see the different network resources being created for you.



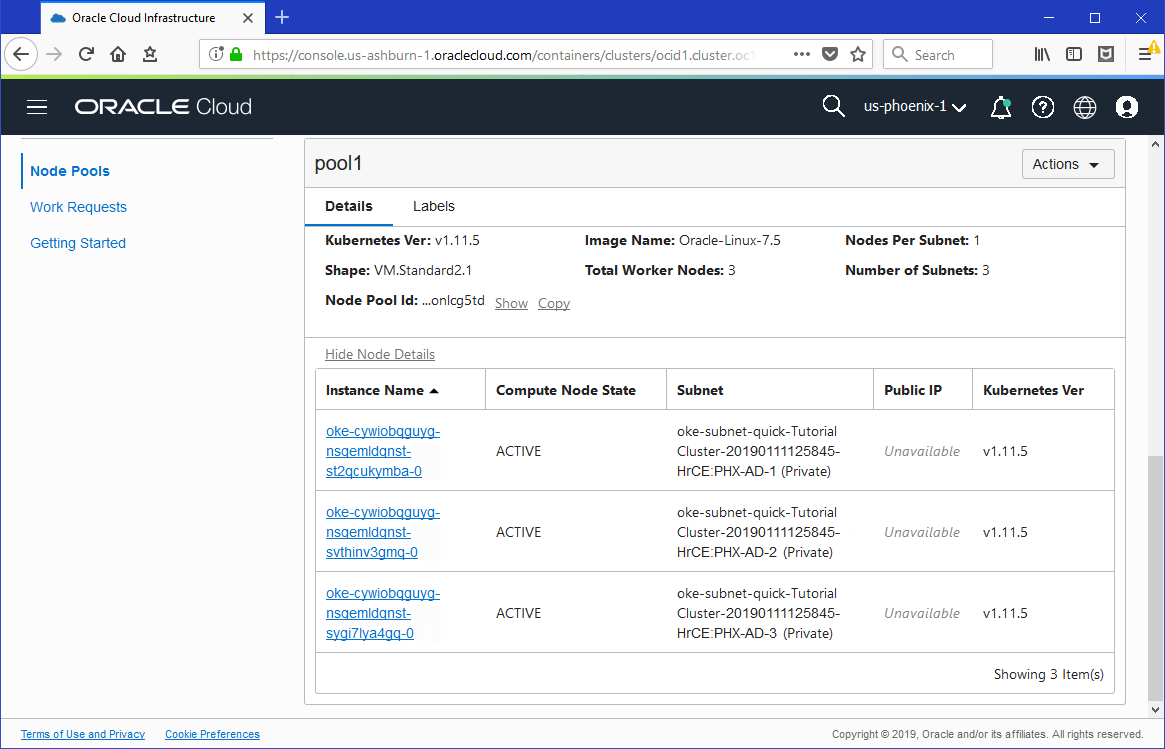
Click **Close** to return to the Console.



The new cluster is shown on the **Cluster Details** page. When it has been created, the new cluster has a status of Active.

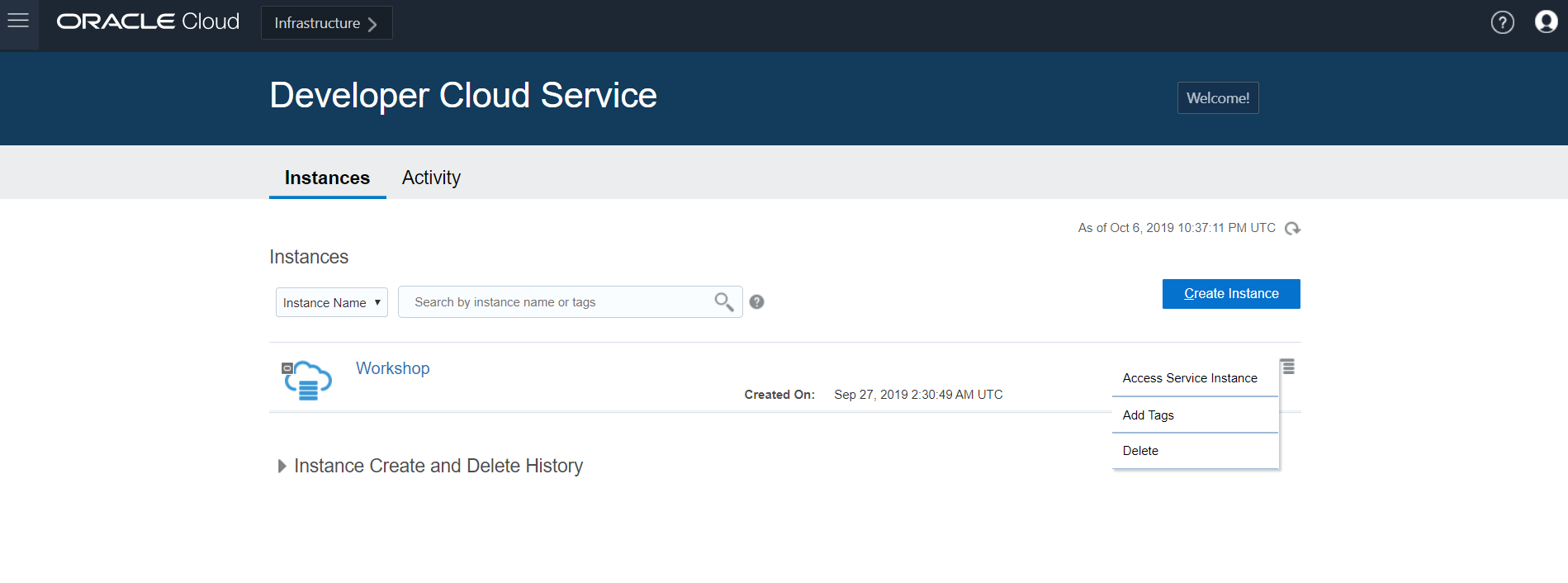


Scroll down to see details of the new node pool that has been created, along with details of the new worker nodes (compute instances).



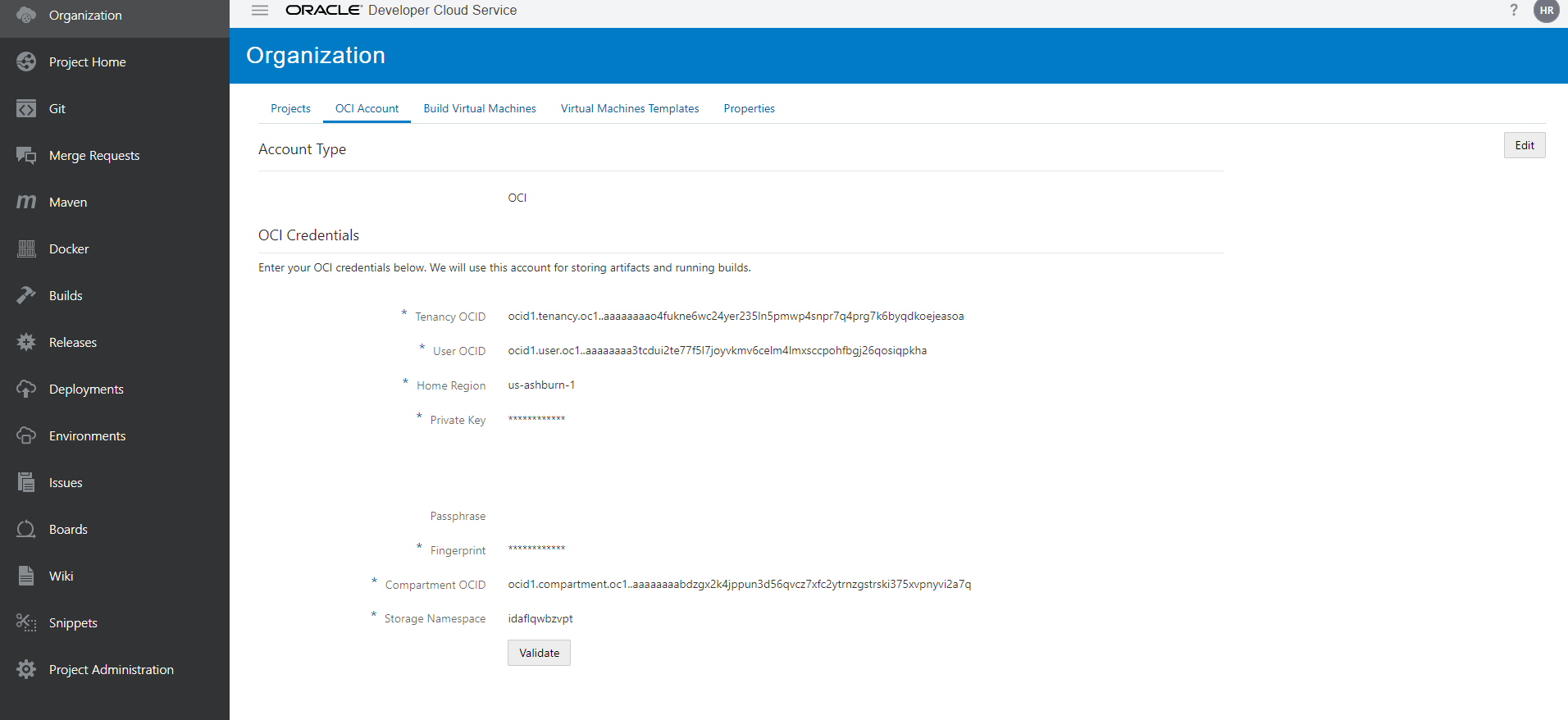
# II **Oracle Developer Cloud setup for your service instance.**

1. In the Console, open the navigation menu. Under **More oracle cloud service** go to **Platform Services** and click **Developer.**
2. **Create Instance**
3. Once the instance active, click **Access Service Instance**.



1. **Configure OCI Account - by referring the URL below.**

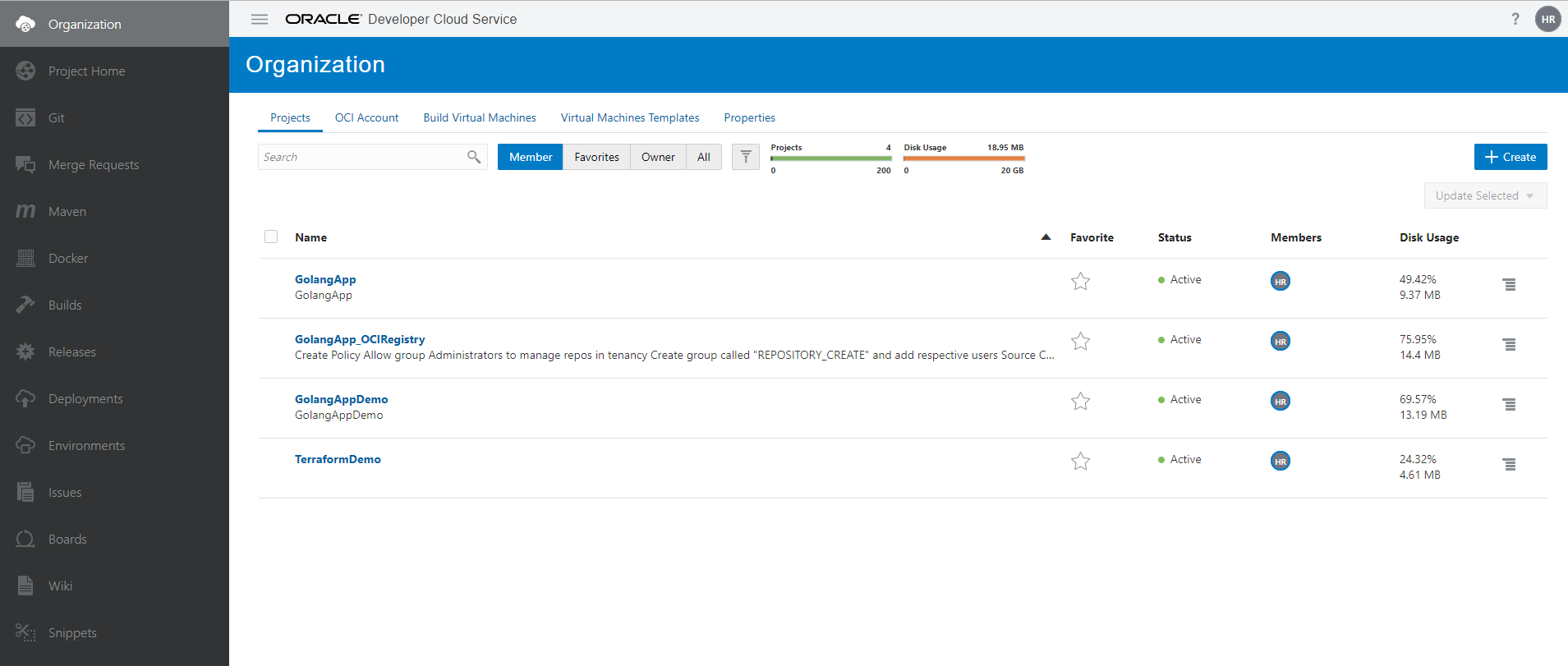
<https://blogs.oracle.com/developers/setting-up-oci-compute-and-storage-for-builds-on-oracle-developer-cloud>



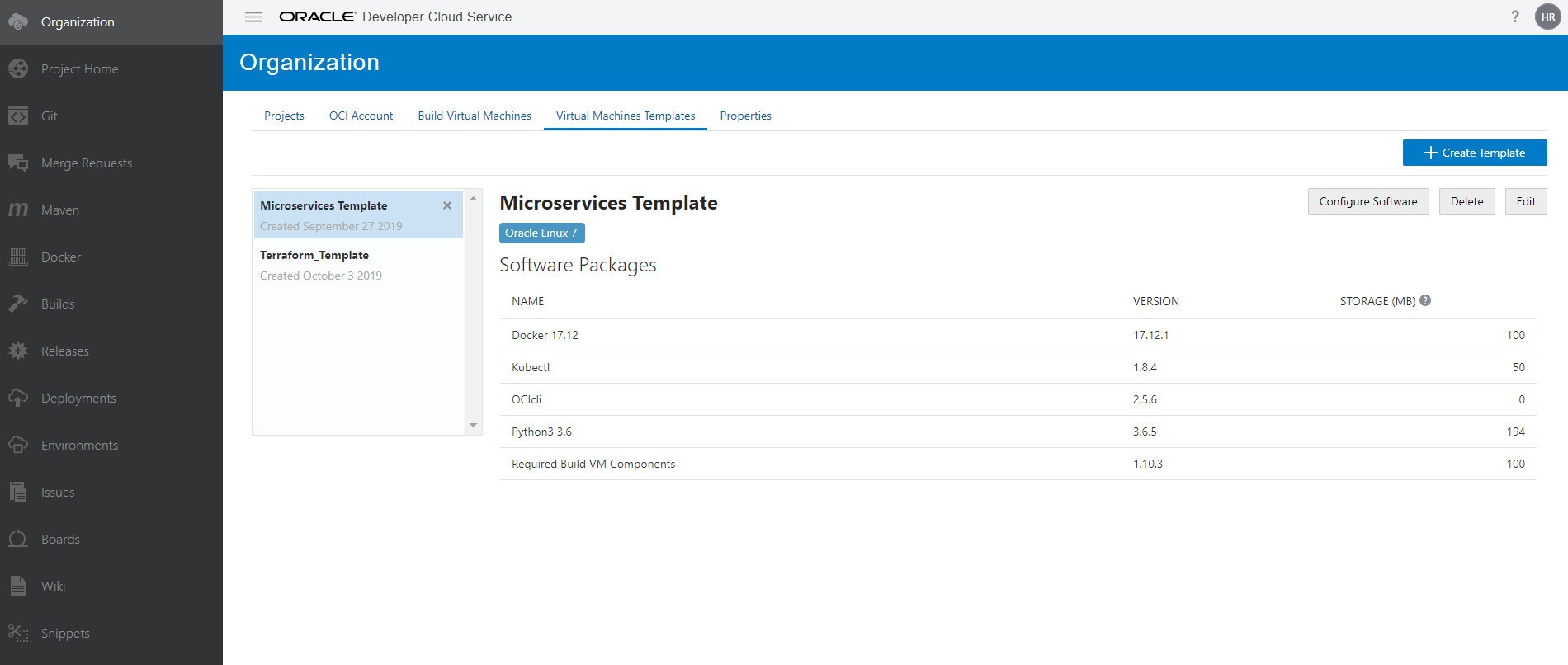
1. **Create and Configure Build VM Templates and Build VMs**

You’ll need to create and configure the Build VM template and Build VM with the required software, which will be used to execute the build job.

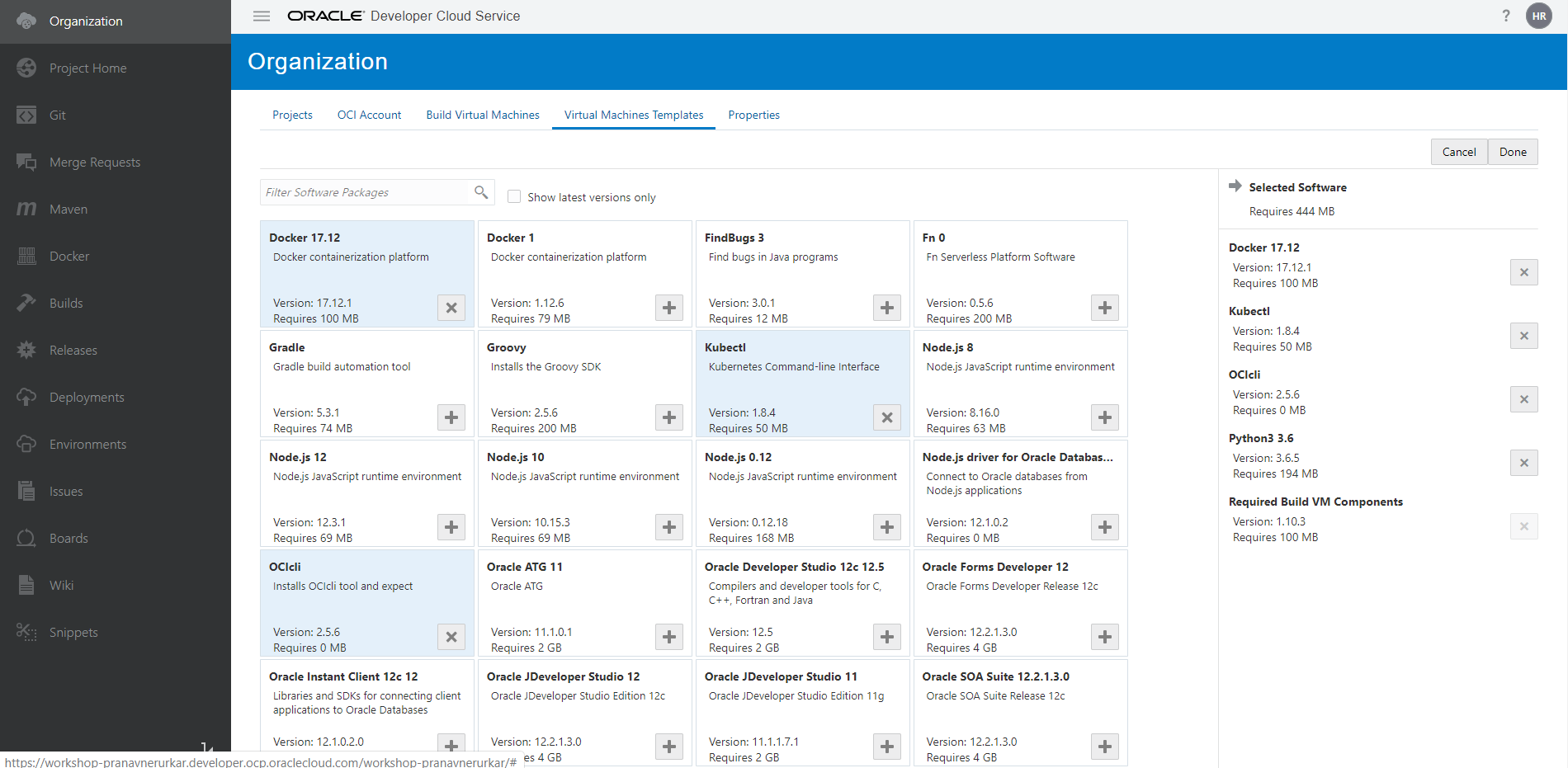
Select Organization and Click **Virtual machine templates** then **Create Template**. In the dialog that pops up, enter a template name, select “Oracle Linux 7” for the platform, then click the **Create** button.



1. After the template has been created, click **Configure Software**



Select **Kubectl** and **OCIcli** (you’ll be asked to add Python3 3.6, as well) from the list of software bundles available for configuration, then click **+** to add these software bundles to the template.

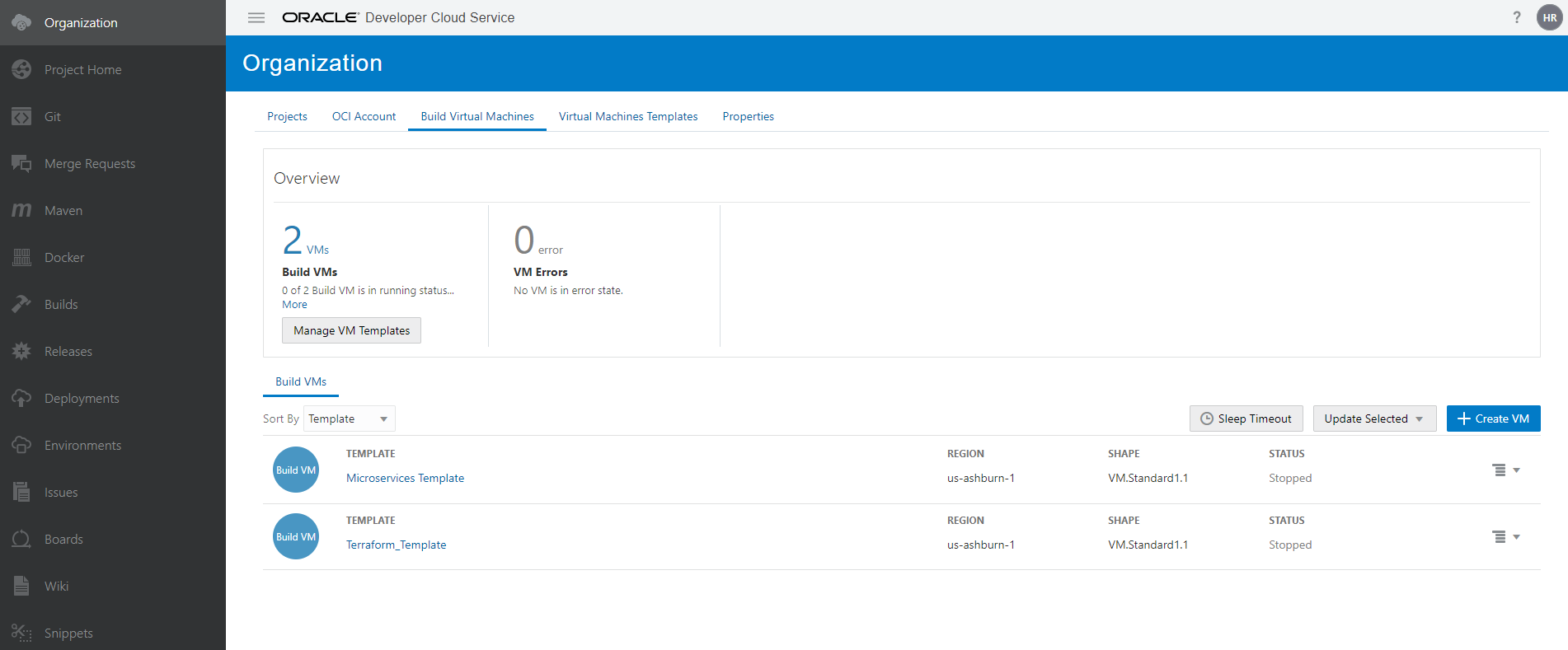


Click the **Done**button to complete the software configuration for that Build VM template.

**Additional Reference to** Build VM template,

<https://docs.oracle.com/en/cloud/paas/developer-cloud/csdcs/manage-project-jobs-and-builds.html#GUID-D27A1C86-27F2-4DBD-A60B-1E0095E3EDD1>

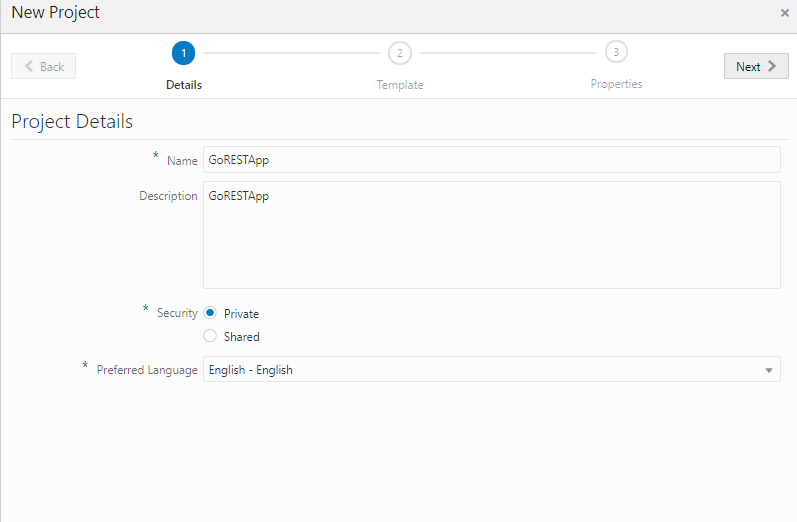
1. From **Organization** select **Build Virtual Machines** and click **+Create VM** in the dialog that pops up, enter the number of VMs you want to create, select the VM Template, Region and VM.



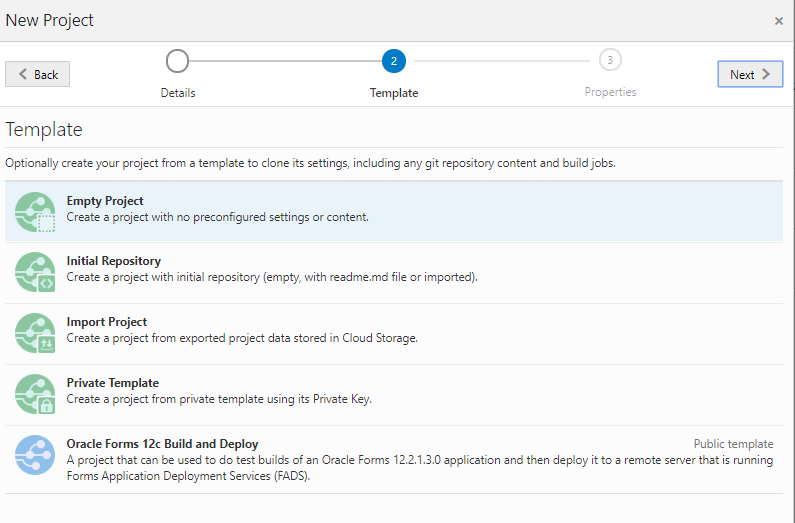
# **Steps to Build and Deploy a Golang Application Using Oracle Developer Cloud**

Create a project.

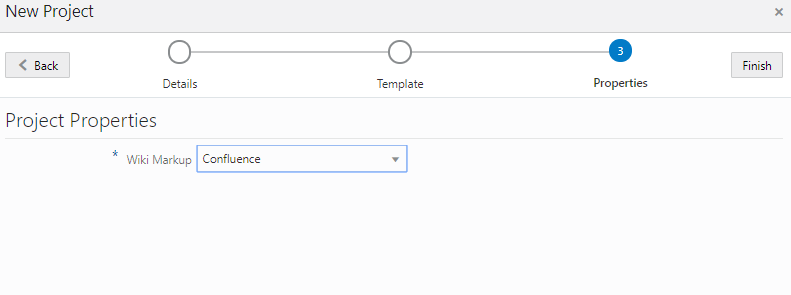
From **Organization** click +**create,** enter a **name** for the project andclick **next.**



From **Organization** click +**create,** enter a **name** andclick **next.**



Select **initial repository,** click next and Choose wiki Markup and finish



* + 1. **Configure the Build and Deployment Jobs**

# **Please refer this blog for Build and Deploy a Golang Application Using Oracle Developer Cloud**

<https://blogs.oracle.com/developers/build-and-deploy-a-golang-application-using-oracle-developer-cloud>

1. To create build job, navigate to the**Builds** page and click the **+Create Job** button **enter a name such as BuildGoRESTAppl and choose the Template that we created for this project and click create.**

**Click Configure: Jobs Overview > BuildGoRESTAppl > Configure**

In the **Git** tab, select **Git** from the **Add Git** dropdown, select **GoREST.git** as the Git repository and, for the branch, select **master**.

In the **Steps** tab, use the **Add Step** dropdown to add **Docker login**, **Docker build**, and **Docker push** steps. Refer the URL to add Docker steps.

1. To create another job for deployment Navigate to the**Builds** page and click the **+Create Job** button.

In the **New Job**dialog enter **DeployGoRESTAppl** for the Name, select the template with Kubectl, then click the **Create** button. This build job will deploy the Docker image built by the **BuildGoRESTAppl** build job to the Kubernetes cluster.

 To do this, in the **Git** tab, add **Git** from the dropdown, select **GoREST.git** as the Git repository and, for the branch, select **master**.

In the **Steps tab,**select**OCIcli**from the**Add Step**dropdown. Take a look at this

url <https://blogs.oracle.com/developers/oracle-cloud-infrastructure-cli-on-developer-cloud> to see how and where to get the values for the OCIcli configuration.

Then, select**Unix Shell**from the**Add Step**dropdown and, in the **Unix Shell** build step, enter the following script. Refer the URL to add steps and UNIX shell script.

1. In the **Create Pipeline** dialog, you can enter the **Name** as **GoApplPipeline.** Then click the **Create** button.
2. Drag and drop the **BuildGoRESTAppl** and **DeployGoRESTAppl** build jobs and then connect them.
3. Double click the link that connects the build jobs and select **Successful** as the **Result Condition**. Then click the **Apply** button.
4. Click the **Build** button, as shown, to run the build pipeline. The BuildGoRESTAppl build job will be executed first and, if it is successful, then the DeployGoRESTAppl build job that deploys the container on the Kubernetes cluster on Oracle Cloud will be executed next.

Enter the public IP address and port that you retrieved from the log, into the browser using the format shown in the following URL:

http://*<retrieved IP address>*:*<retrieved port>*/<your name>

**Additional Options,**

**Import external Github repository.**

**Create Oracle Cloud Infrastructure Registry.**

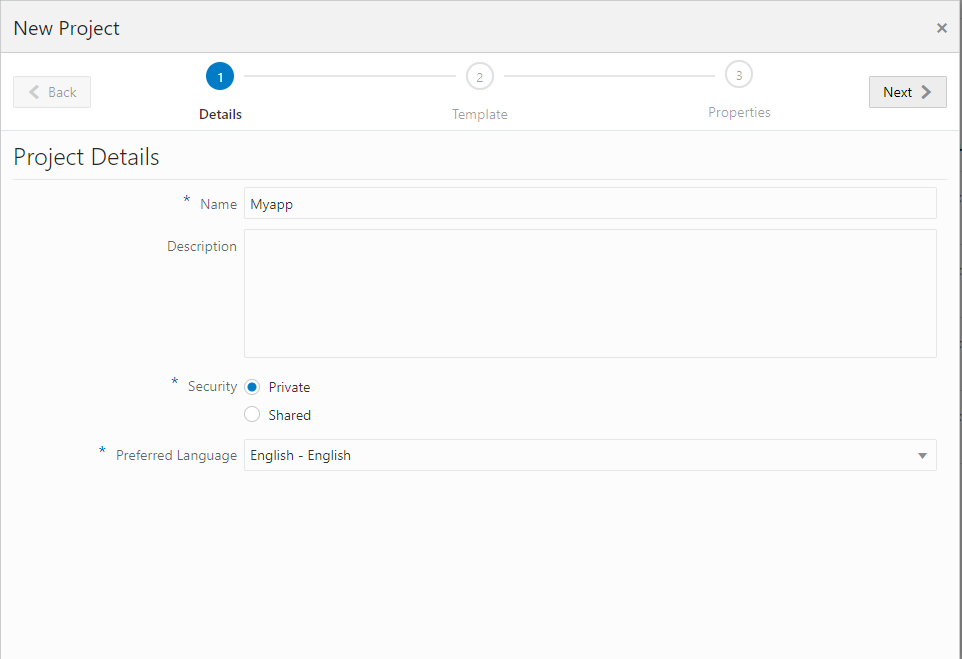
**Required Keys and OCIDs**

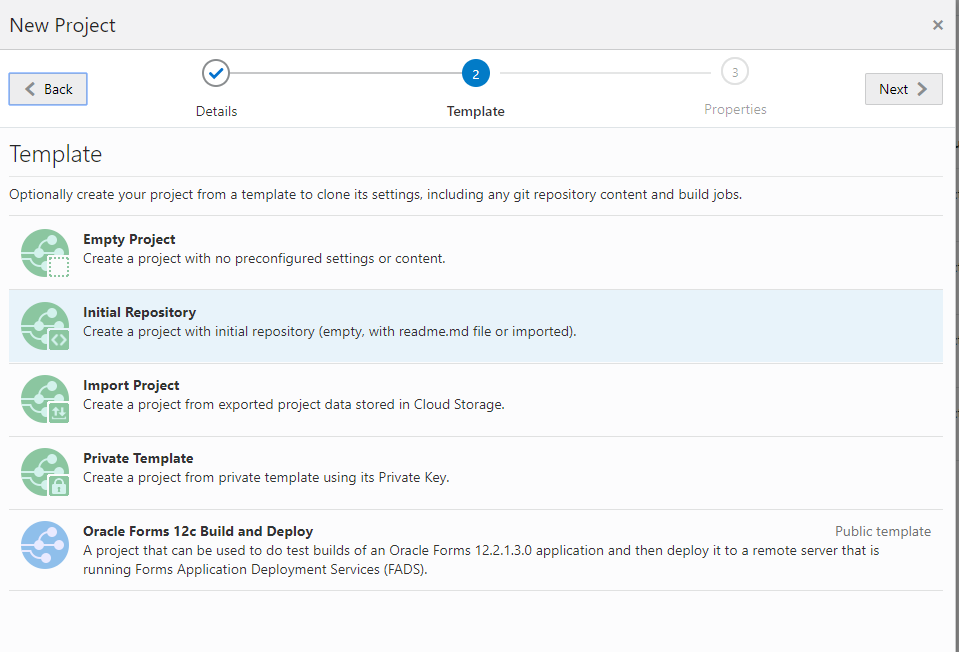
**Import from external Github repository**

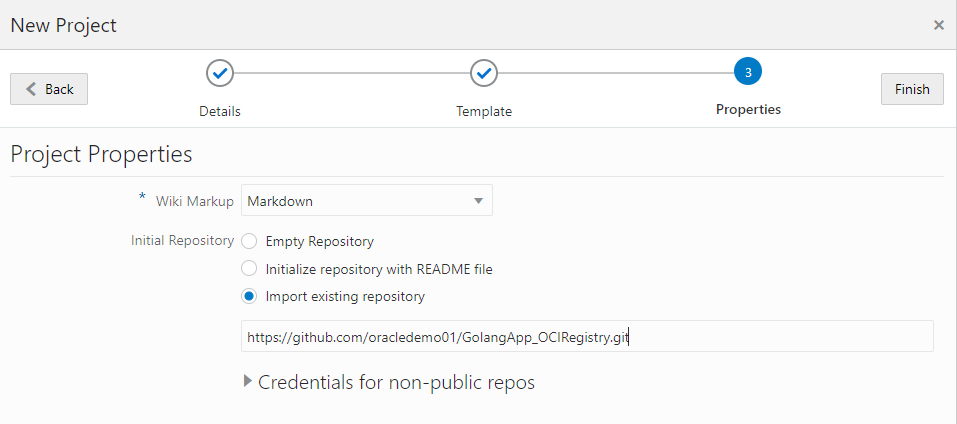
From **Organization** click +**create,** enter a **name** andclick **next.**

Select **initial repository and** click next

Choose wiki Markup and finish.







**Create Oracle Cloud Infrastructure Registry**

Reference:

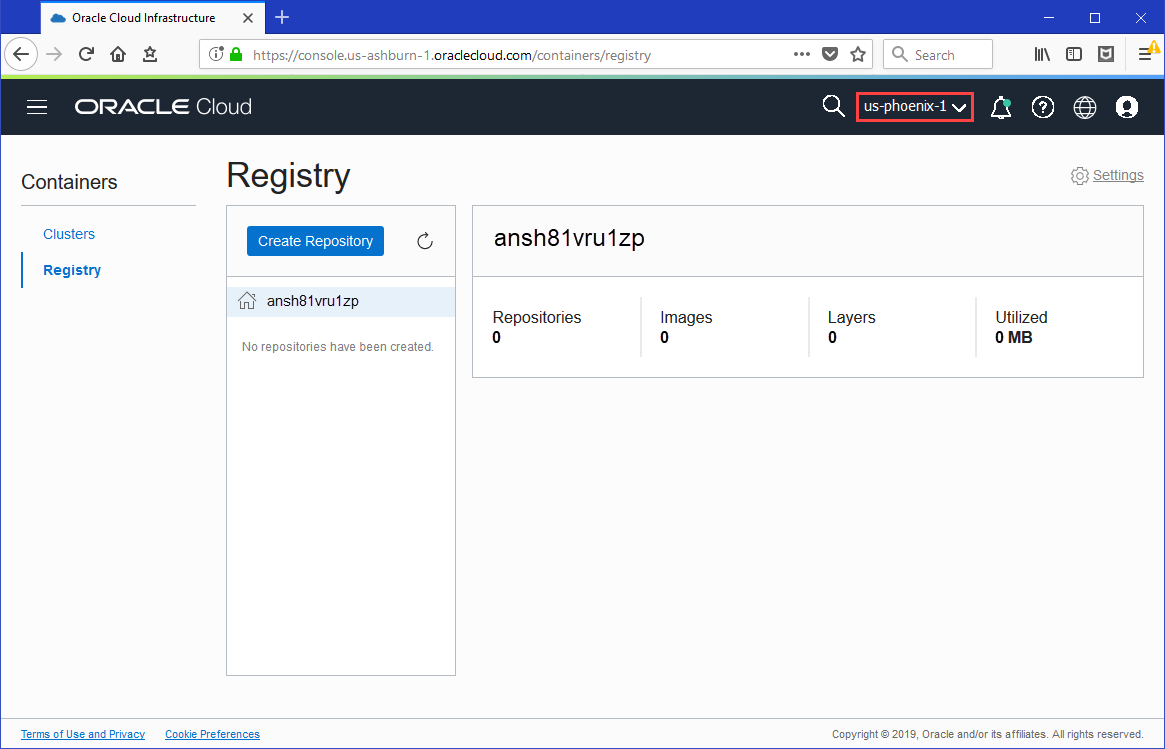
https://blogs.oracle.com/shay/automating-cicd-for-docker-with-oracle-cloud-infrastructure-registry-and-developer-cloud-service

Specify a tenancy in which you have the appropriate permissions to create repositories in Oracle Cloud Infrastructure Registry. You inherit these permissions in one of the following ways:

. By belonging to the tenancy's Administrators group

. By belonging to another group to which a policy grants the appropriate Oracle Cloud Infrastructure Registry permissions, including the REPOSITORY\_CREATE permission.

In the Console, open the navigation menu. Under **Solutions, Platform and Edge**, go to **Developer Services** and click **Registry**



# **Required Keys and OCIDs**

This key pair is **not** the SSH key that you use to access compute instances. See [Security Credentials](https://docs.cloud.oracle.com/iaas/Content/General/Concepts/credentials.htm).

Both the private key and public key must be in PEM format (not SSH-RSA format). The public key in PEM format looks something like this:

Reference:

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

Note: **Git for Windows provides a BASH emulation** used to run Git from the command line. \*NIX users should feel right at home, as the BASH emulation behaves just like the "git" command in LINUX and UNIX environments. URL: <https://gitforwindows.org/>

Generate the private and public key:

openssl genrsa -out oci\_api\_key.pem 2048

openssl rsa -pubout -in oci\_api\_key.pem -out oci\_api\_key\_public.pem

openssl rsa -pubout -outform DER -in ~/.oci/oci\_api\_key.pem | openssl md5 -c

