Oracle Cloud Infrastructure – Operation Efficiency

Hands on Lab

ORACLE

Table of Contents

1. Disclaimer

The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

In the document, value of information such as user name, password etc. is shown as XXXX or YYYY etc. Also if actual value is shown consider it as an example value.

2. Introduction

Background

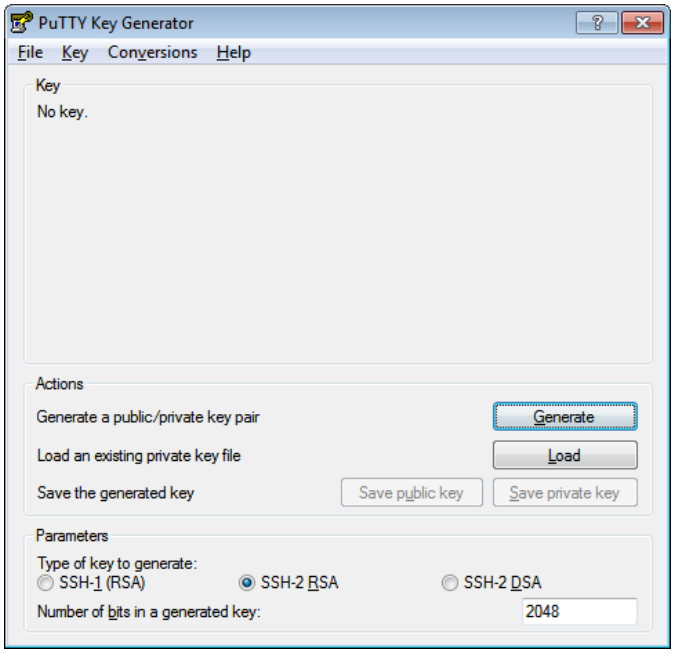
Oracle Cloud Infrastructure (OCI) is a set of complementary cloud services that enable you to build and run a wide range of applications and services in a highly available hosted environment. Oracle Cloud Infrastructure offers high-performance compute capabilities (as physical hardware instances) and storage capacity in a flexible overlay virtual network that is securely accessible from your on-premises network.

3. Pre-Requisites

- To perform the steps in this tutorial, you must have an active subscription to Oracle Cloud Infrastructure or a Free Trial Account.
- PuTTY and PuTTYGen installed

To generate an SSH key pair using the PuTTY Key Generator,

- Find puttygen.exe in the PuTTY folder on your computer, for example, C:\Program Files (x86)\PuTTY. Double-click puttygen.exe to open it. Or you may download it from here.



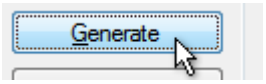
- Accept the default key type, **SSH-2 RSA**.

SSH-2 is the most recent version of the SSH protocol (and is incompatible with SSH-1). **RSA** and DSA are algorithms for computing digital signatures.

- Set the **Number of bits in a generated key** to 2048 bits, if it is not already set with that value.

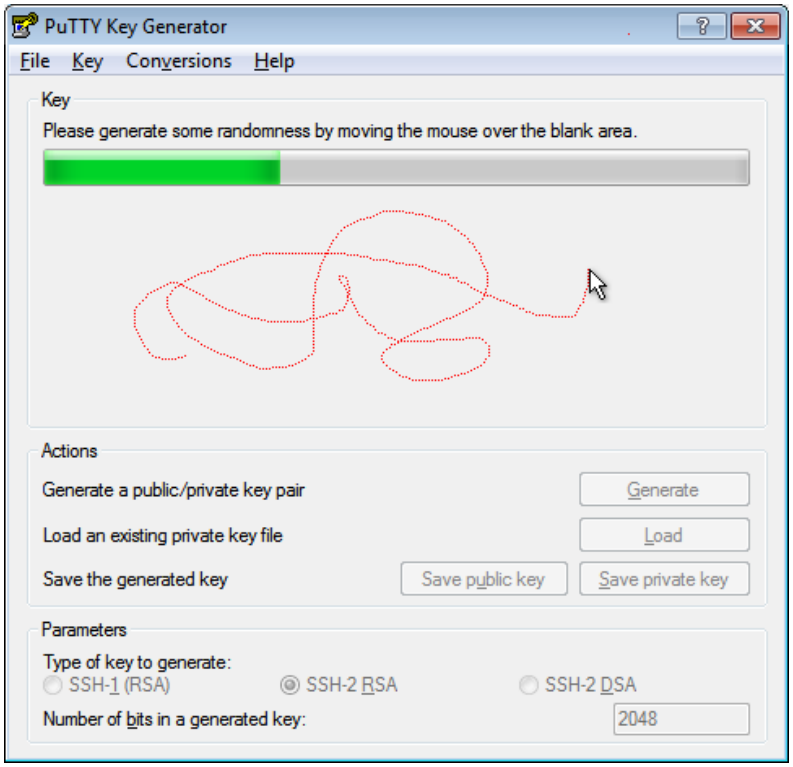
This sets the size of your key and thus the security level. A minimum of 2048 bits is recommended for SSH-2 RSA.

- Click Generate.

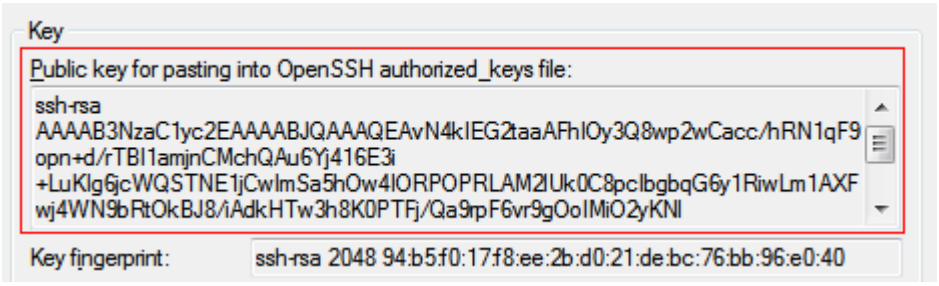


- Move your mouse around the blank area to generate randomness to the key.

Note: the dotted red line in the image below is for illustration purposes only. It does not appear in the generator pane as you move the mouse.

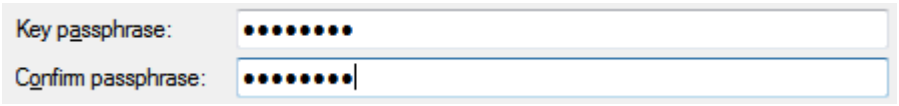


- The generated key appears under Public key for pasting into OpenSSH authorized_keys file.



- The key comment is the name of the key that you will use to identify it. You can keep the generated key comment or create your own.

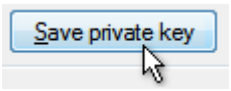
- If you want to password-protect your key, enter a **Key passphrase** and enter it again for **Confirm passphrase**. When you reload a saved private key, you will be asked for the passphrase, if one is set.
-



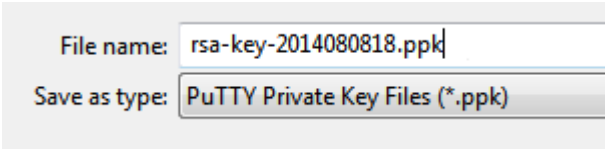
While a passphrase is not required, you should specify one as a security measure to protect the private key from unauthorized use.

There is no way to recover a passphrase if you forget it.

- Save the private key of the key pair. Depending how you work with the private key in the future, you may need one saved in the PuTTY PPK format and one saved in OpenSSH format. Let's do both.
    - To save the key in the PuTTY PPK format, click **Save private key** to save the private key of the key pair.

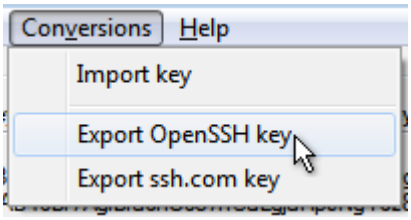


You can name it anything you want, although you may want to use the same name as you used for the key comment. The private key is saved in PuTTY's Private Key (PPK) format, which is a proprietary format that works only with the PuTTY toolset.

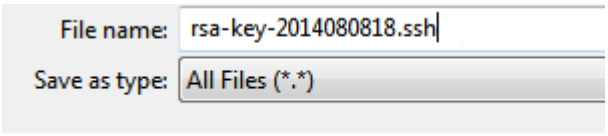


You can use this key whenever you use Putty to perform SSH actions.

- To save the key in OpenSSH format, open the **Conversions** menu and select **Export SSH key**. This will be the same key as above, just saved in a different format.

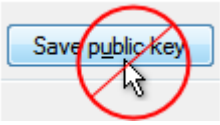


You can name it anything you want, but to keep track of your keys, you should give it the same name as the key you saved in PPK format in the previous step. You can also use any extension (or no extension), but let's use .ssh, to make it clear what format it is.



You can use this key whenever you use OpenSSH to perform SSH actions using ssh utitlities that support OpenSSH, for example when using Linux in a command shell.

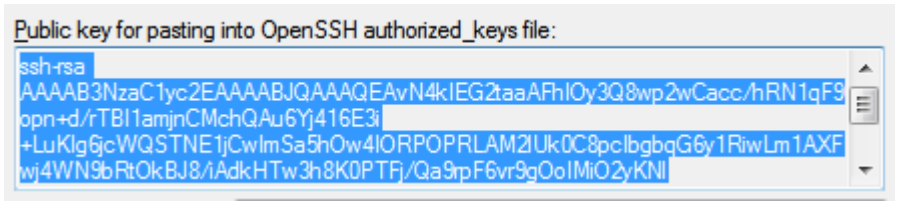
- Now you need to create the public key to be paired with the private key(s) you just created. However, clicking the **Save public key button** will create a public key that won't work with Oracle Cloud services in certain cases. So, for the purposes of this tutorial, there is no reason to save a public key using the **Save public key** button.



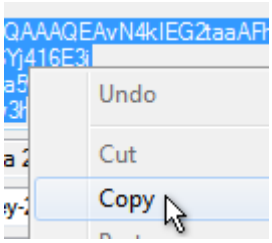
Instead, proceed as follows.

- In the PuTTY Key Generator, select all of the characters under Public key for pasting into OpenSSH authorized_keys file.

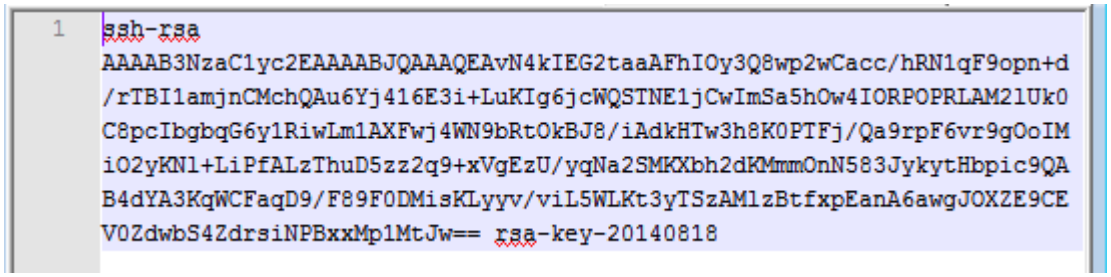
Make sure you select all the characters, not just the ones you can see in the narrow window. If a scroll bar is next to the characters, you aren't seeing all the characters.



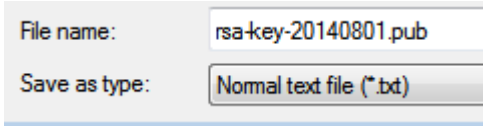
Right click somewhere in the selected text and select **Copy** from the menu.



- Open a text editor and paste the characters, just as you copied them. Start at the first character in the text editor, and do not insert any line breaks.



- Save the key as a text file, using the same root name as you used for the private key. Add a .pub extension. You can give it any extension you want, but .pub is a useful convention to indicate that this is a public key.



Write down the names of your public and private keys, and note where they are saved. You will need the public key when creating service instances in, for example, Oracle Java Cloud Service and Oracle Database Cloud - Database as a Service. You will need the private key when trying to access a service instance's virtual machine via SSH.


Let the Fun Begin!

4. Signing in to the OCI Console

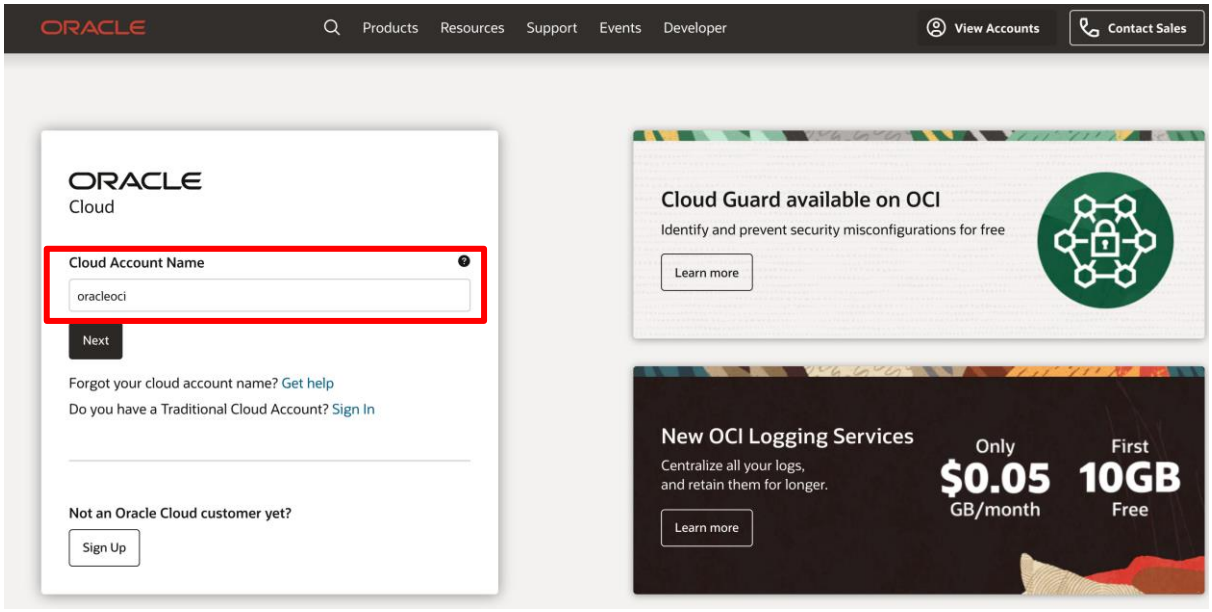
Oracle Cloud Infrastructure Identity and Access Management (IAM) Service lets you control who has access to your cloud resources. You control the types of access a group of users has and to which specific resources. The purpose of this lab is to give you an overview of the IAM Service components and an example scenario to help you understand how they work together.

Pre-requisites:

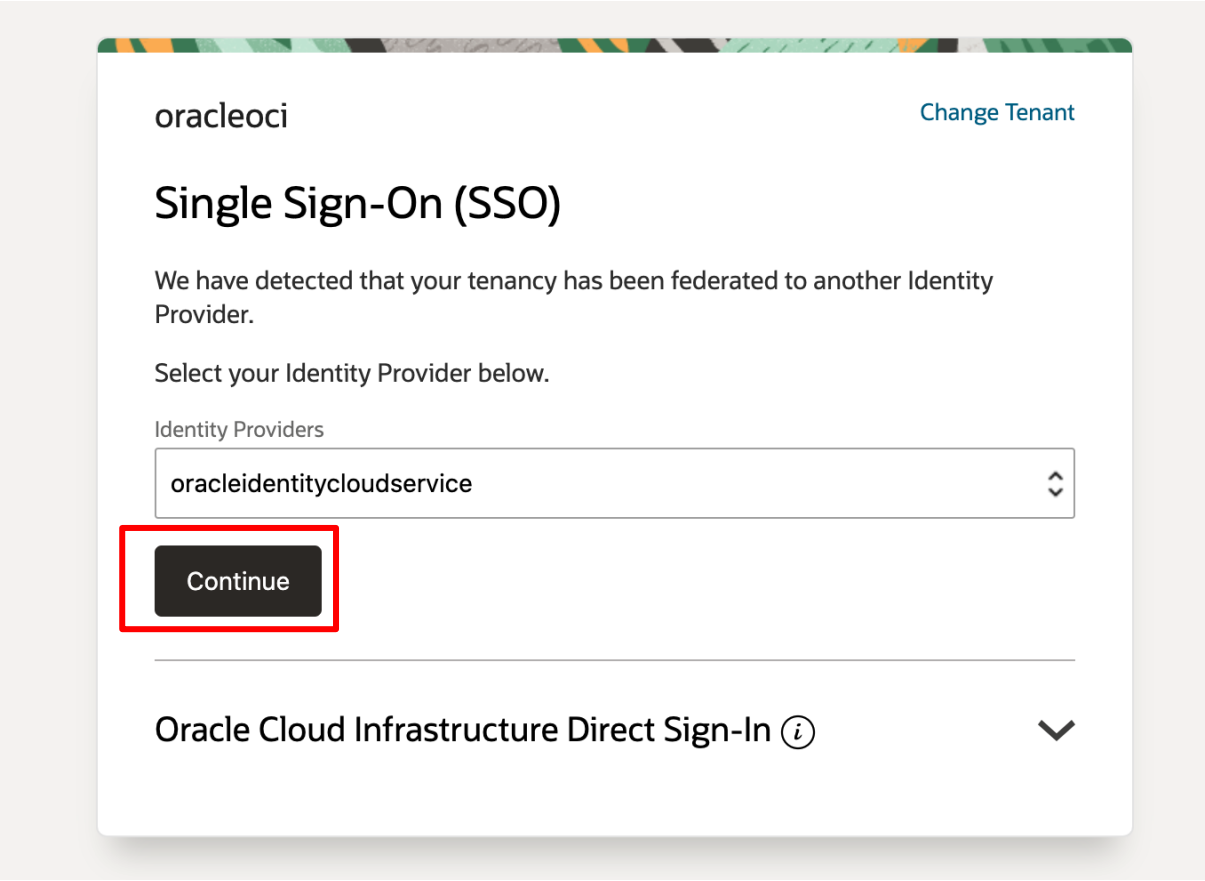
- Oracle Cloud Infrastructure account credentials (User, Password, and Tenancy)
- To sign in to the Console, you need the following:
    - Tenant, User name and Password
    - URL for the Console: https://console.eu-frankfurt-1.oraclecloud.com/
    - Oracle Cloud Infrastructure supports the latest versions of Google Chrome, Firefox and Internet Explorer 11

In this Lab, you will sign in to the Oracle Cloud Infrastructure console using your credentials.

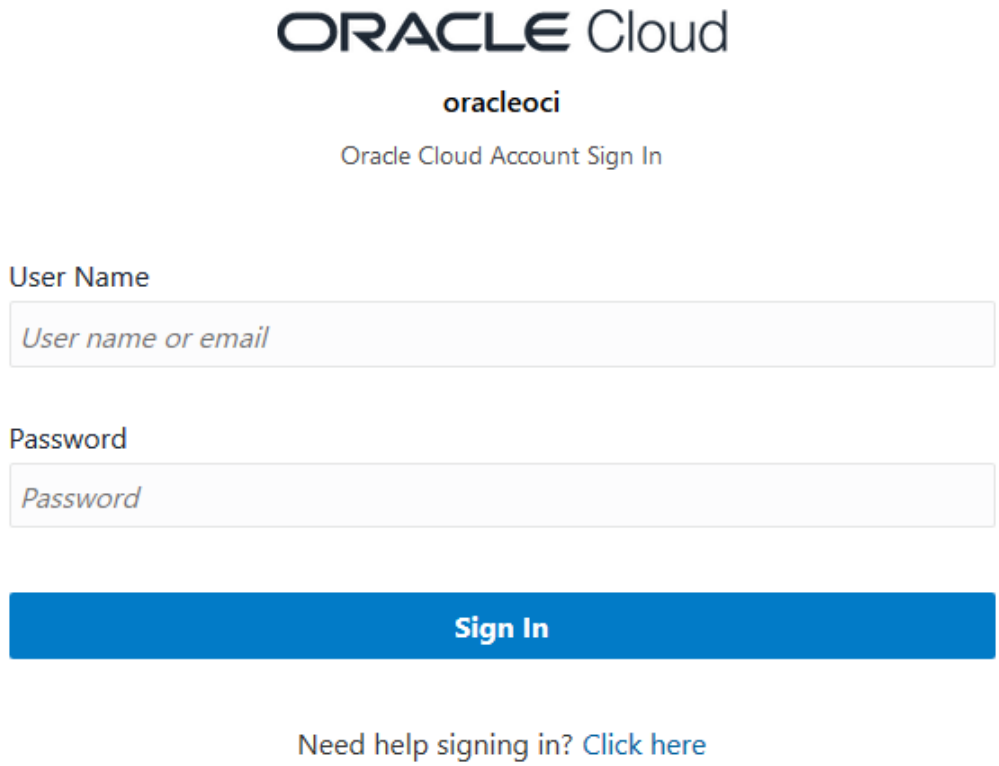
1. Open a supported browser and go to the Console URL: https://console.eu-frankfurt-1.oraclecloud.com/.
2. Enter your tenant name and click **Next**



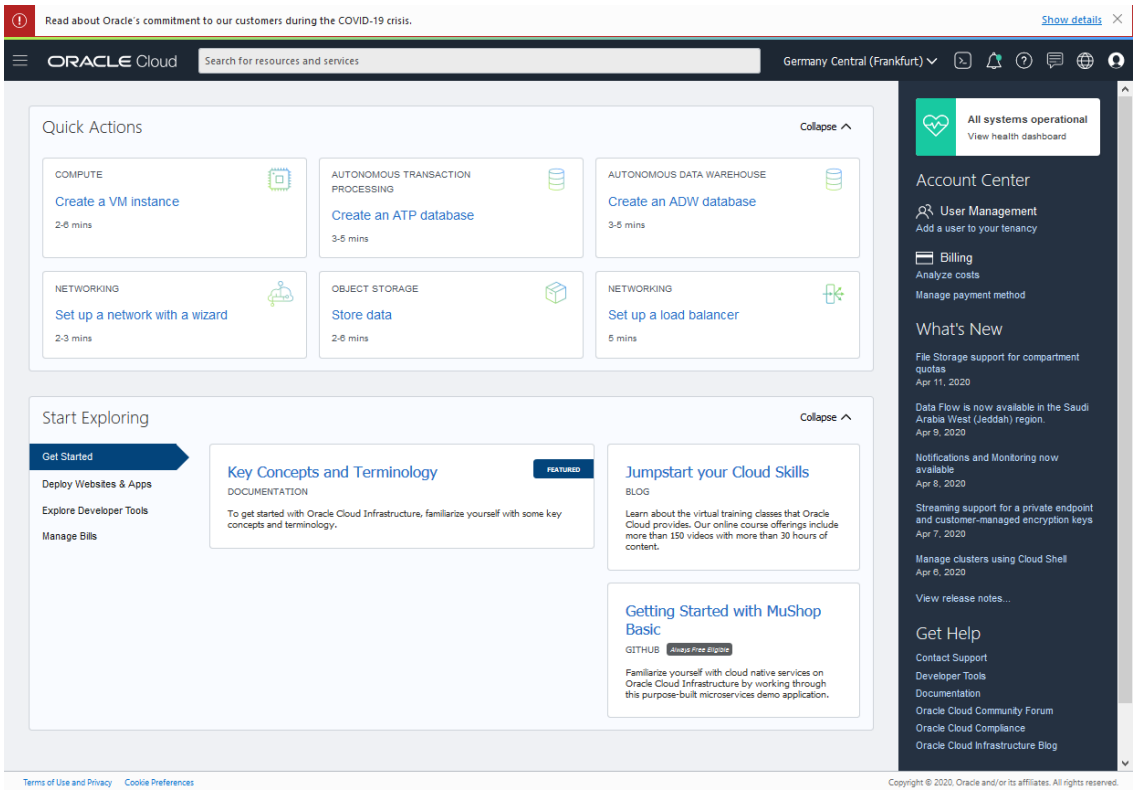
3. Oracle Cloud Infrastructure is integrated with Identity Cloud Services, you will see a screen validating your Identity Provider. Click **Continue**.



4. Enter your user name and password
   - **Username:** *instructor will provide username*
   - **Password:** *instructor will provide password*



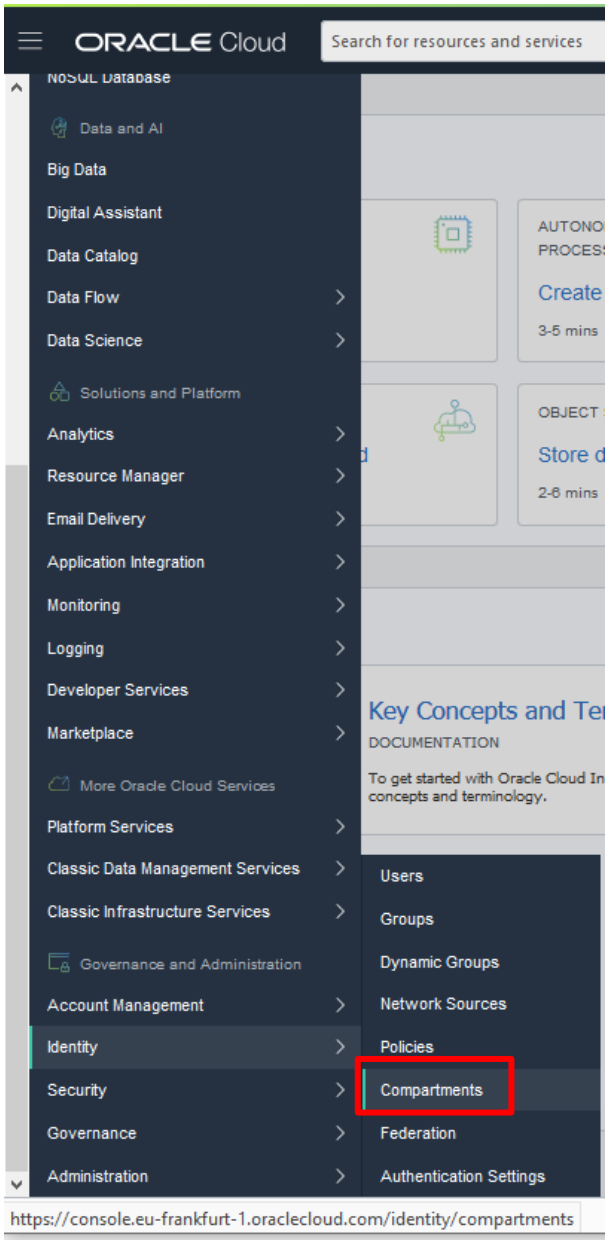
5. When you sign in to the Console, the dashboard is displayed.



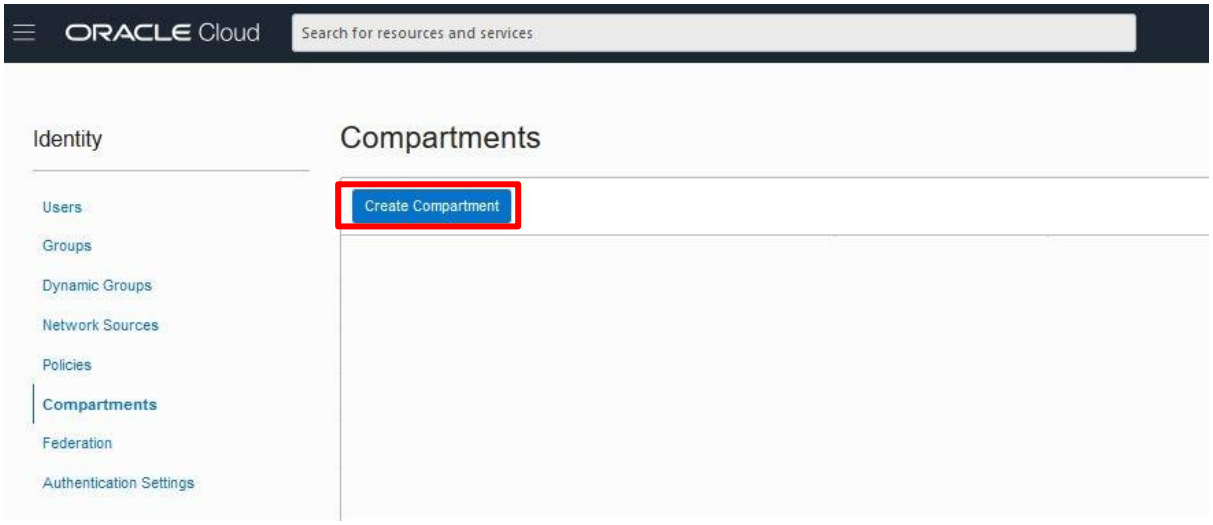
5. Creation of a compartment

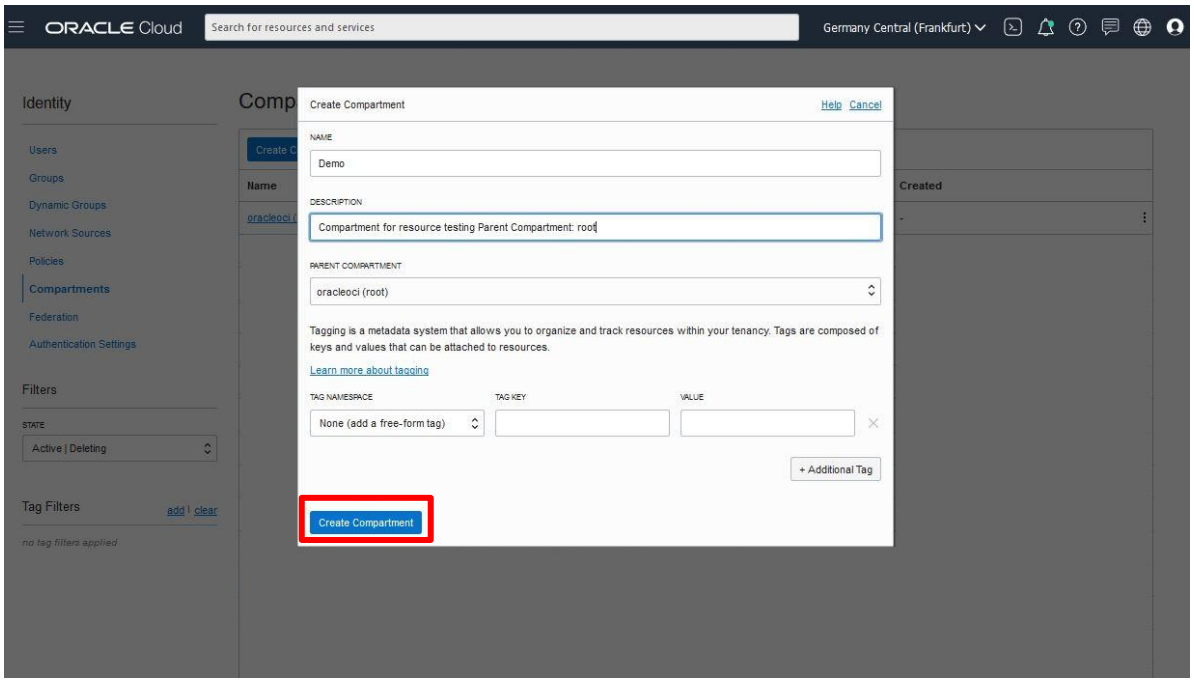
A **compartment** – is a collection of related resources that can be accessed only by certain groups that have been given permission by an administrator in your organization. When you first start working with Oracle Cloud Infrastructure, you need to think carefully about how you want to use compartments to organize and isolate your cloud resources. Compartments are fundamental to that process. Most resources can be moved between compartments.

In order to view and create a compartment in your tenancy, please navigate to **Main Menu**, **Governance and Administration** section, **Identity** and select **Compartments**.



► Click on "**Create Compartment**" and fill the information:
Name: *Demo*
Description: *Compartment for resource testing Parent Compartment: root*



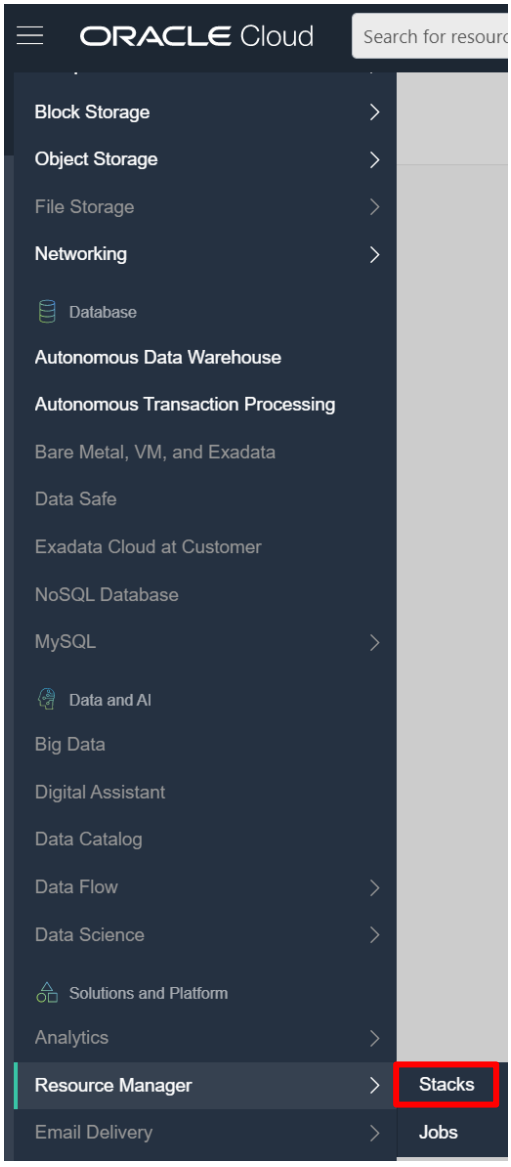


To learn more about compartments in OCI, please visit: https://docs.cloud.oracle.com/en-us/iaas/Content/Identity/Tasks/managingcompartments.htm

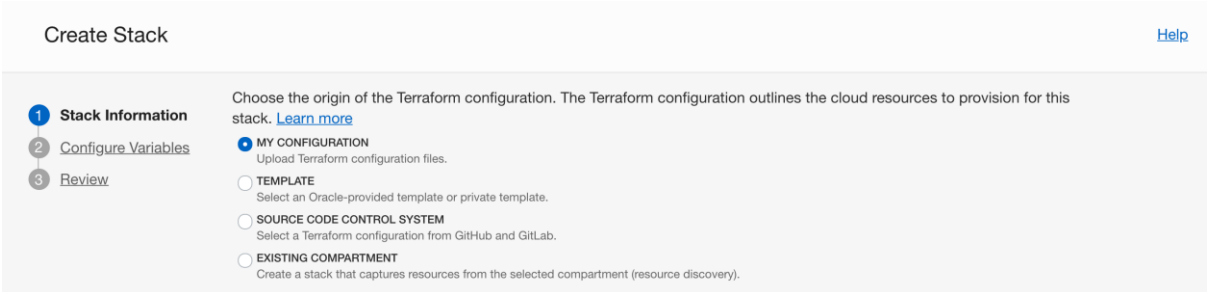
6. Deploy a sample application using Resource Manager

Download the MuShop zip file from github: https://github.com/oracle-quickstart/oci-cloudnative/releases/latest/download/mushop-basic-stack-latest.zip

Please navigate to **Main Menu**, **Resource Manager** section and select **Stacks.**

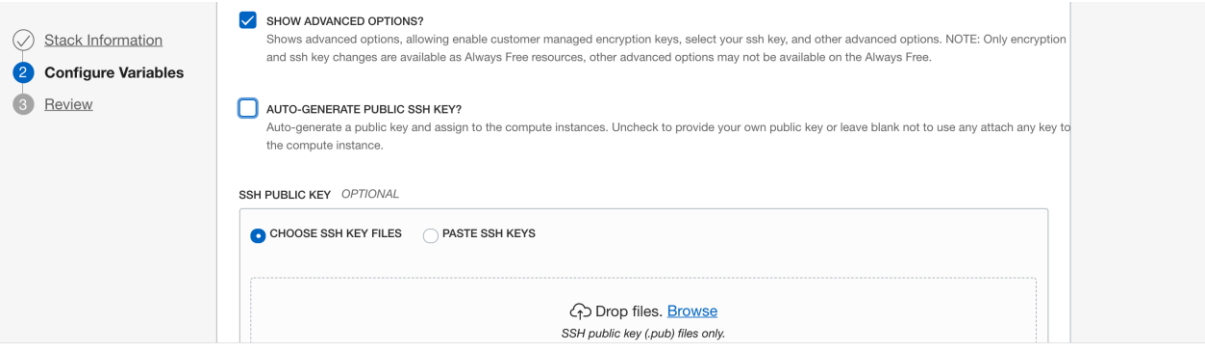


Choose the origin of the Terraform configuration: “My configuration”

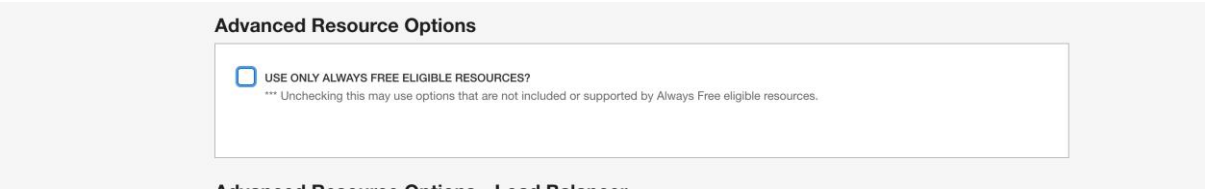


Drag and drop the .zip file into the **Stack Configuration** window, fill in the name as “MushopDemo”, description “Ecommerce Application Demo”, select “Demo” compartment and select “Next”.

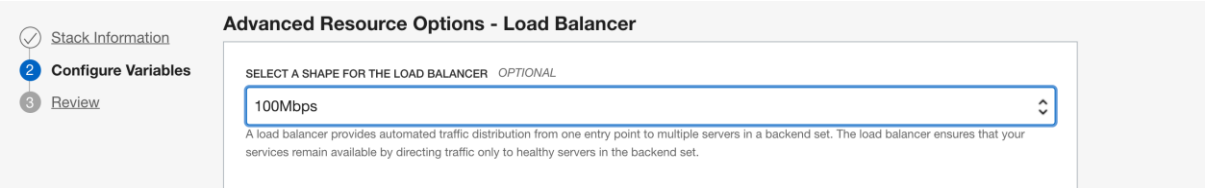
Select “Show advanced option?” an unmark “auto-generate public ssh key?” add the SSH Public key you have created and select Next:



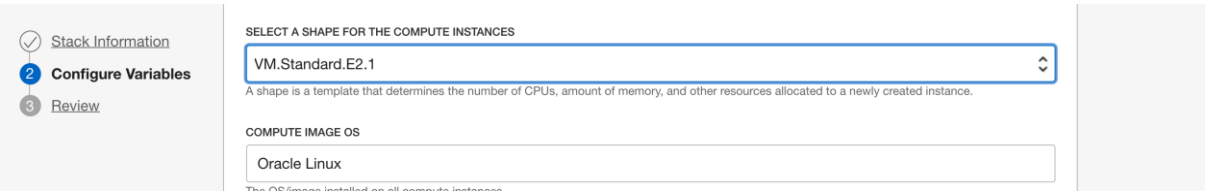
In **Advanced Resource Option** section, uncheck “use only always free eligible resources?” (this is due because some of “always free” resources are not available in all availability domains. The Mushop Terraform stack does not allow to select the availability domain).



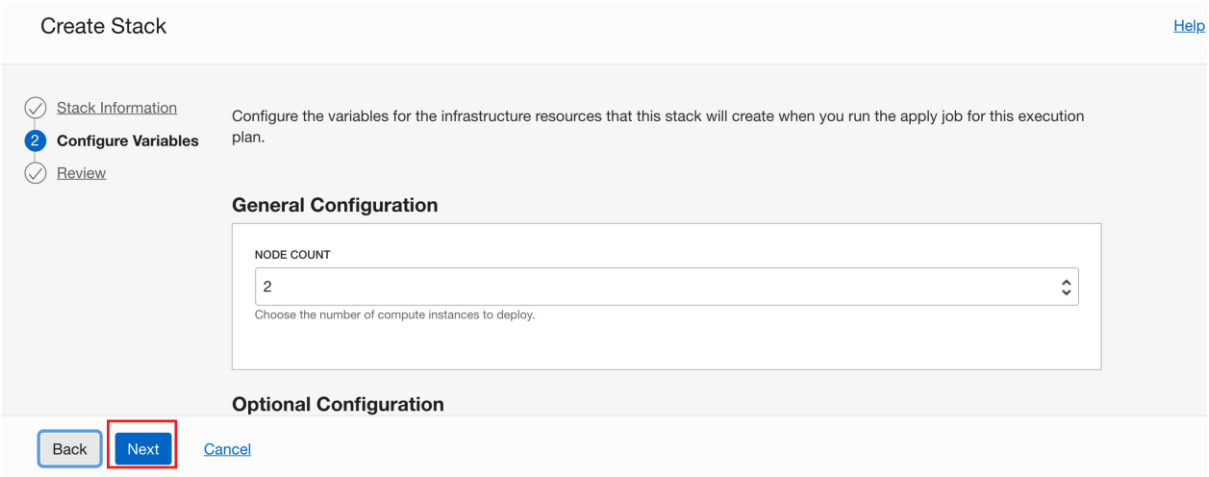
Change the Shape for the Load Balancer (100Mbps):



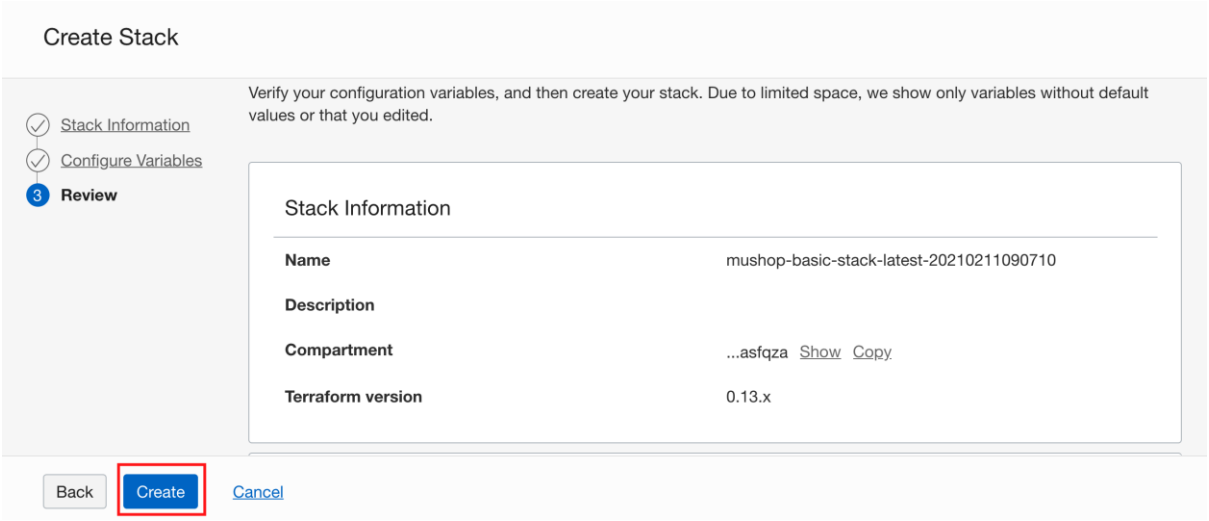
Change the Shape for the Compute Instances (VM.Standard.E2.1):



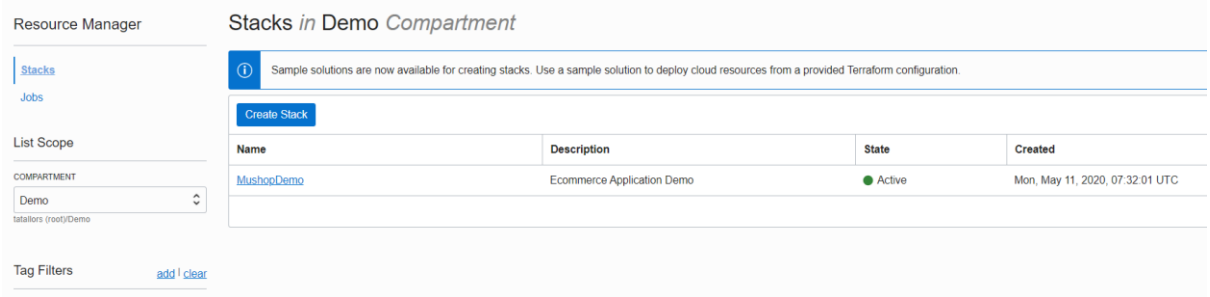
Select “NEXT:



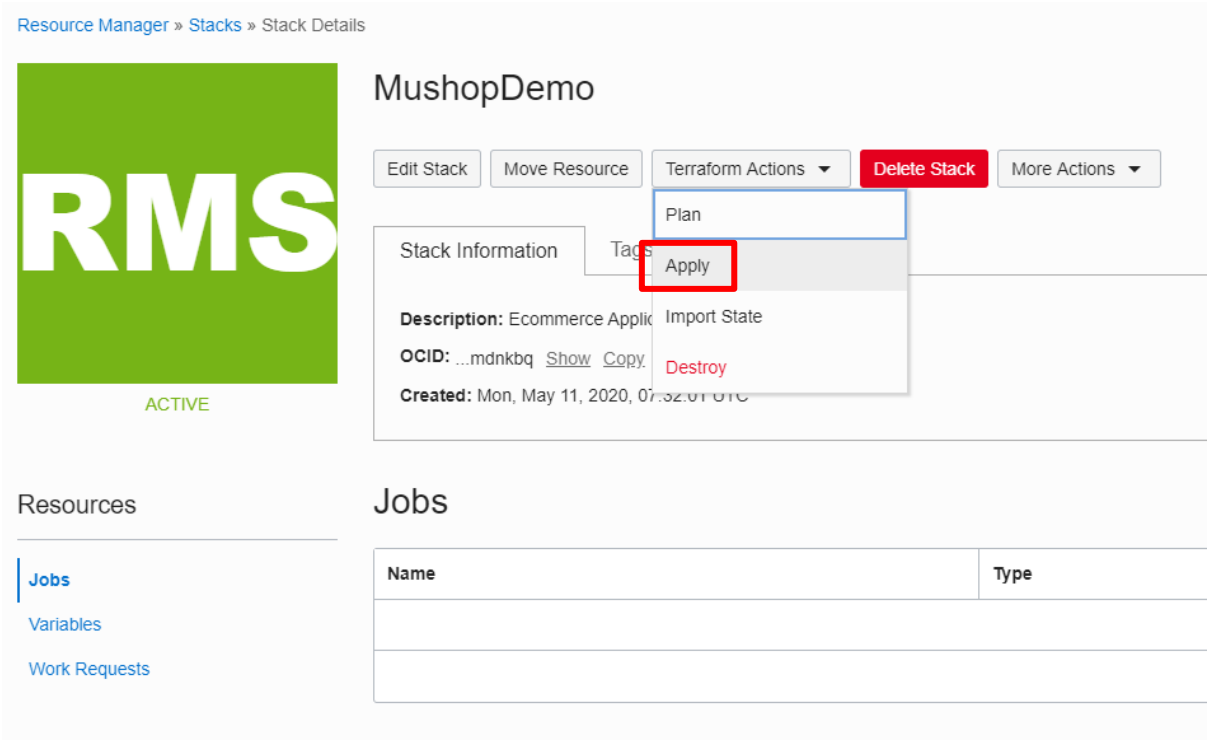
Review the configuration and select **Create**:



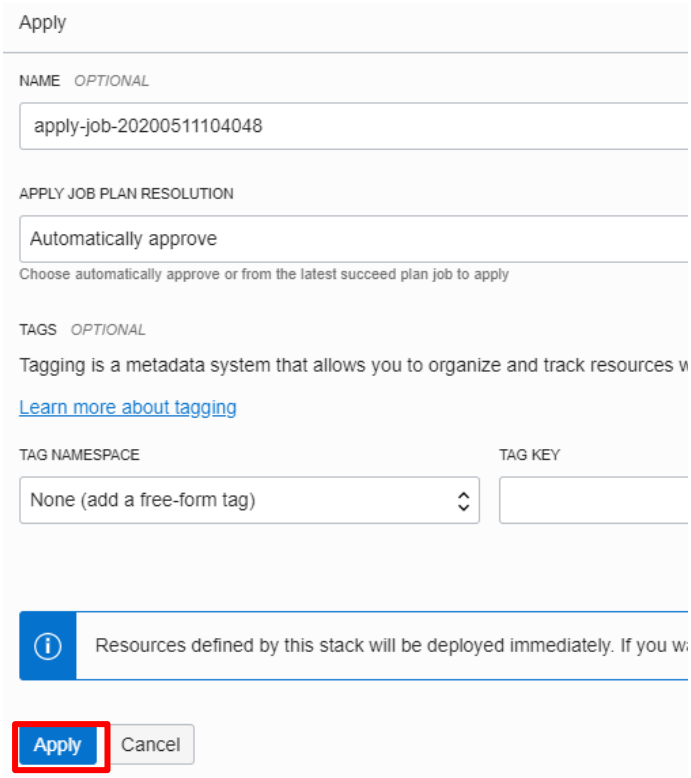
Wait for the stack to be created and select it by clicking on it:



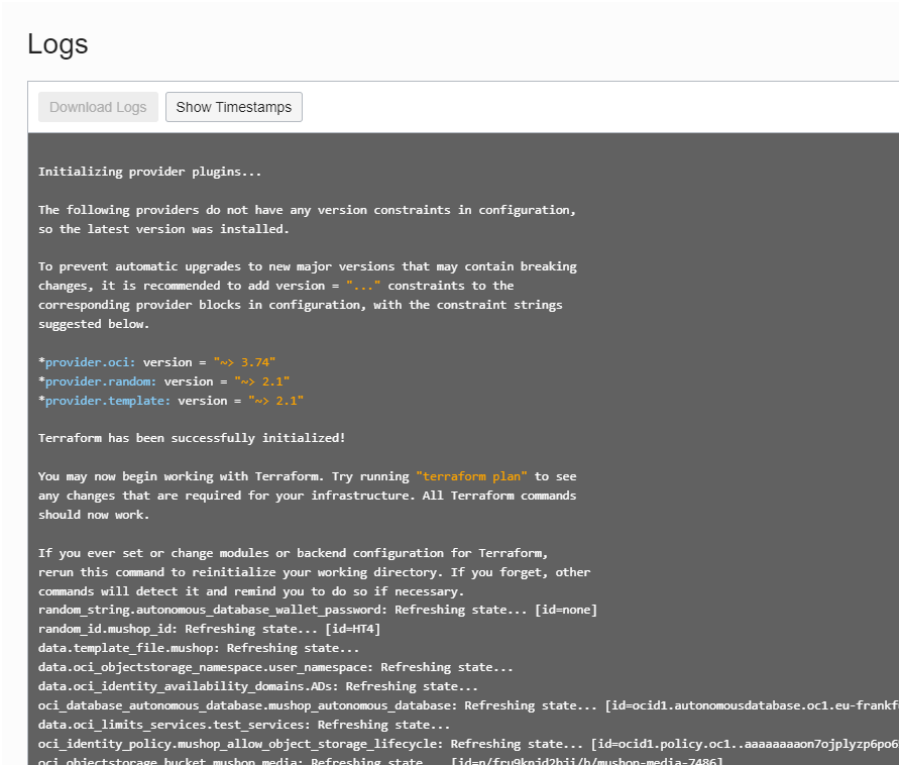
The terraform stack is loaded , so now it has to be deployed by executing Terraform actions. Select the “Terraform action” button and select “Apply”:



Review the configuration and select “Next”:



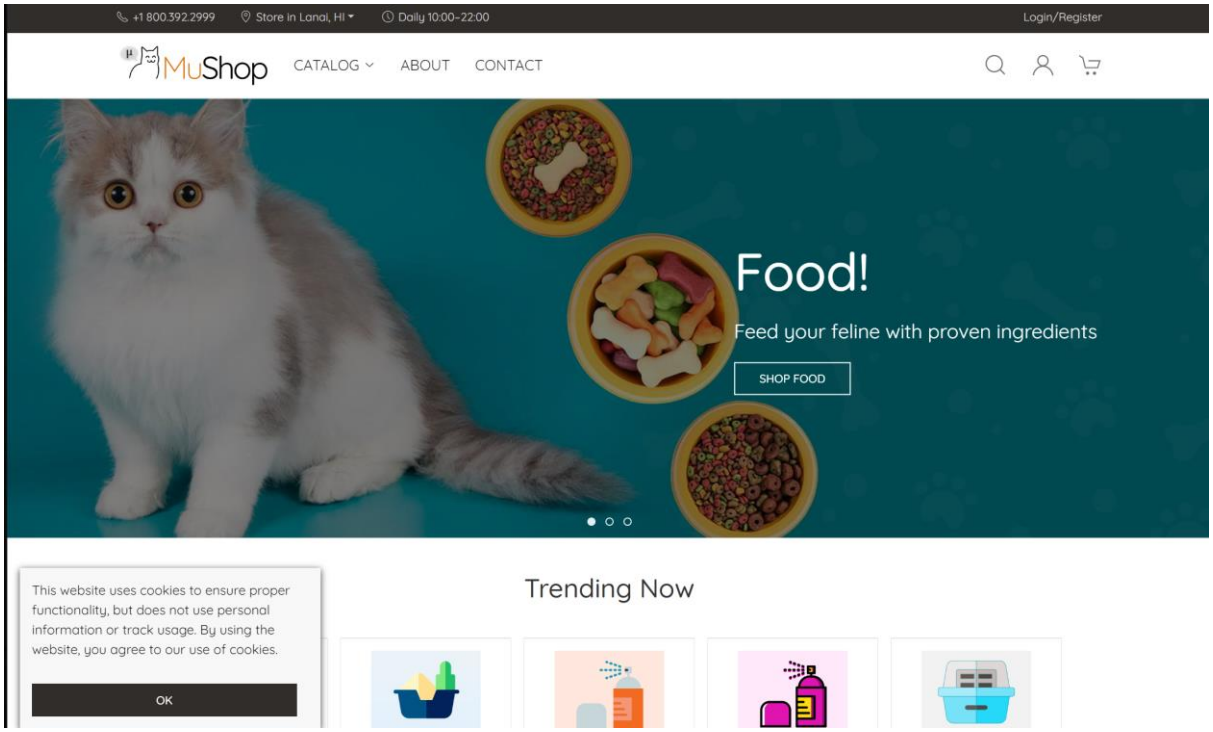
While you wait until the “Apply Job” is deployed and it’s status color changes to green scroll through the logs at the bottom of the page and notice the terraform code being executed and infrastructure being provisioned.



When the Apply job is finished you will observe the public IP of the application at the end of the log. Make a not of your specific IP :

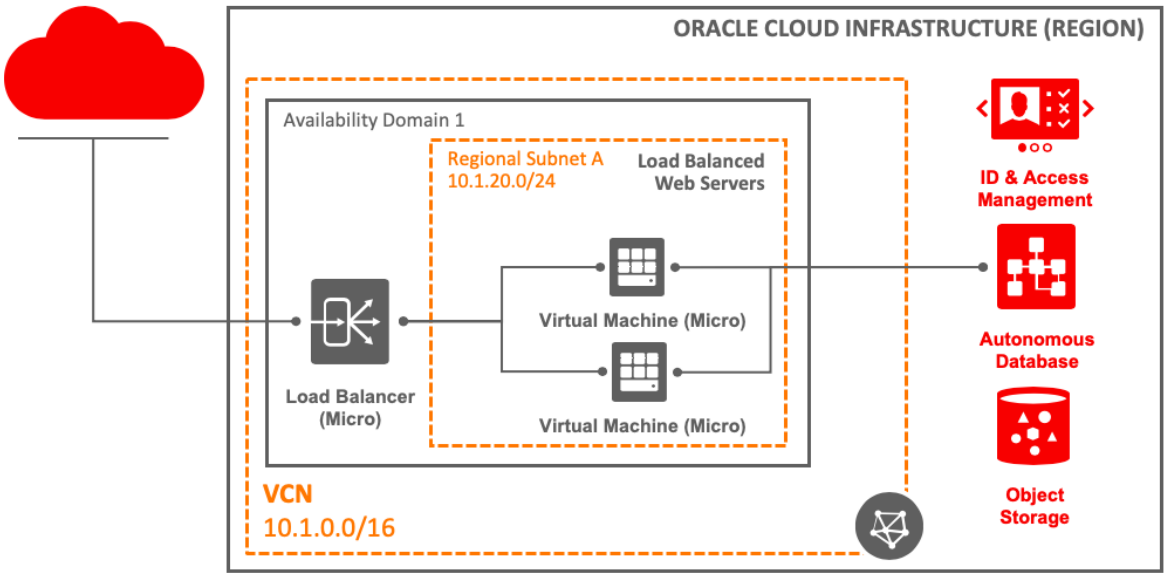
```
autonomous_database_password = 1mW<%w67C}vhUAps
comments = The application URL will be unavailable for a few minutes after provisioning, while the application is configured
dev = Made with ♥ by Oracle A-Team
lb_public_url = http://132.145.224.225
```

Open a browser windows and enter the **lb_public_url** IP address. You will notice the web application loading:



Congratulations, you have just deployed a fully functional Ecommerce application using Resource Manager!

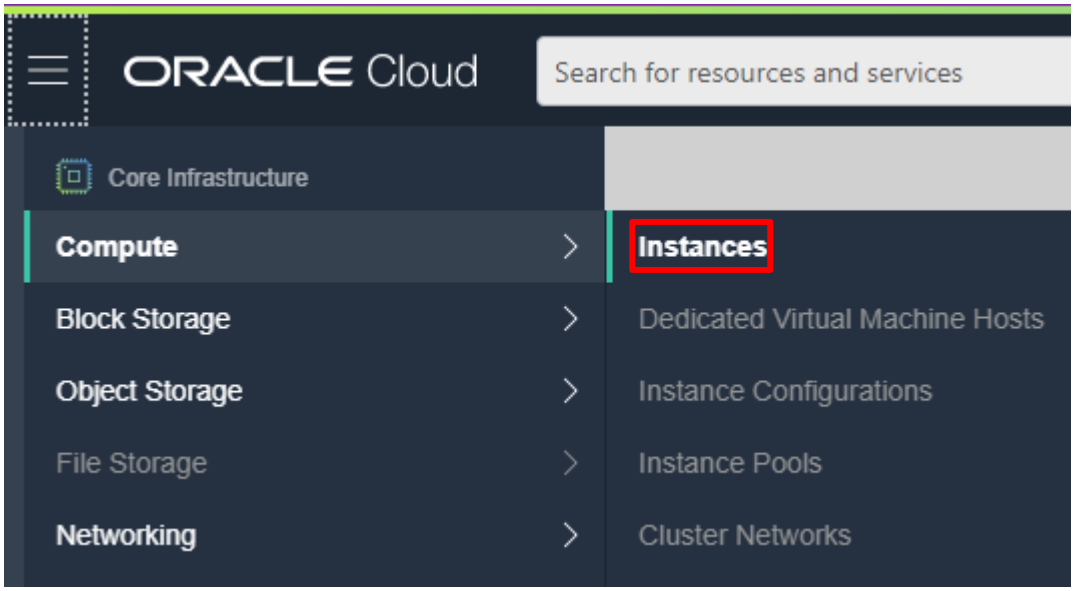
The application topology is as follows:



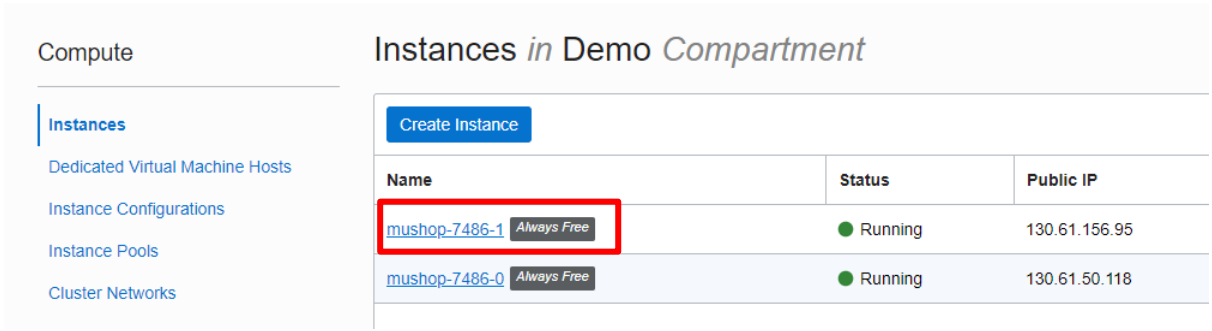
Browse through the OCI Console and review the infrastructure that has been provisioned.

7. Set an CPU load alarm

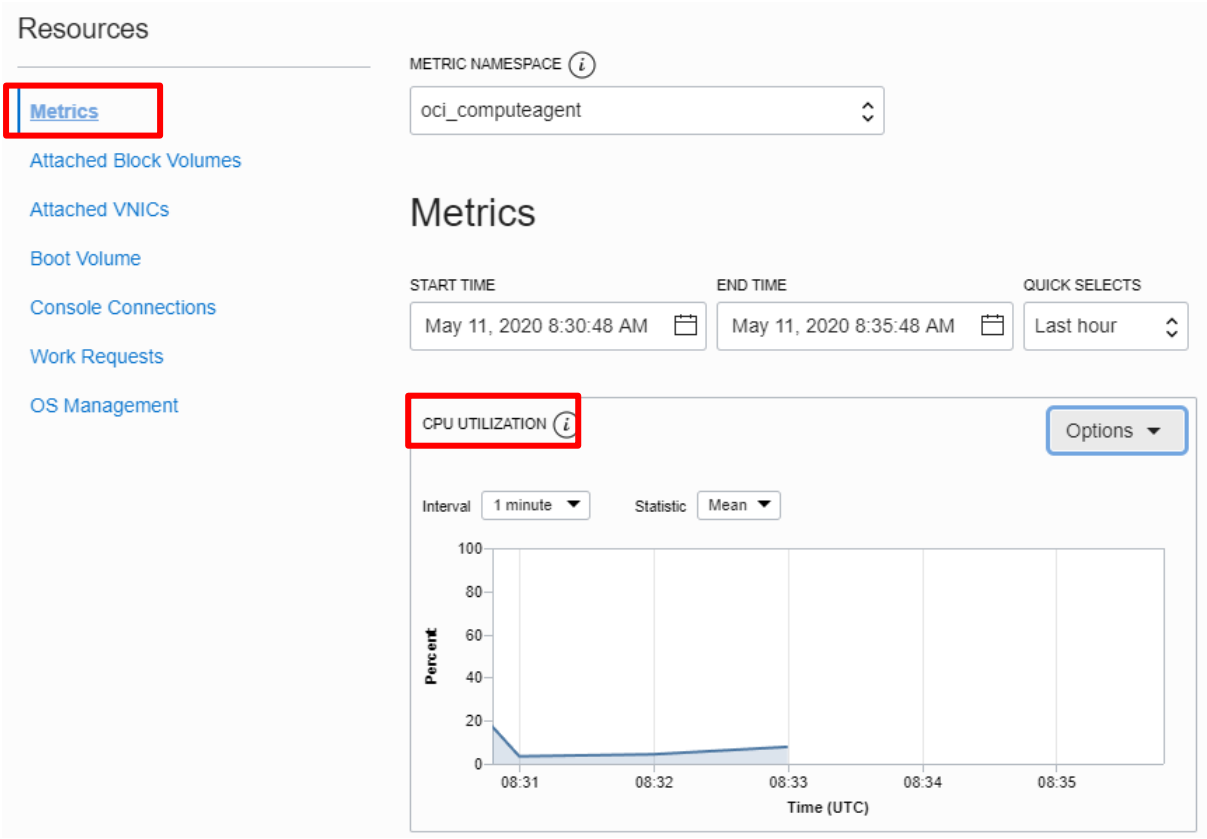
Navigate to **Main Menu**, **Compute**, **Instances**



Select the first of the two computer instances:



Scroll down until you reach the **Metrics** section and the **CPU Utilization** chart:



Select the **Options** button and select the **Create an Alarm on this Query** option.

You will now be redirected to the **Create Alarm** page:

Fill or select the values in the required fields as follows:

Alarm name: CPUAlarm

A;arm Severity: Critical

Alarm Body: High CPU Usage

Compartment: Demo

Metric Namespace: oci_computeragent

Metric Name: CpuUtilization

Interval: 1m

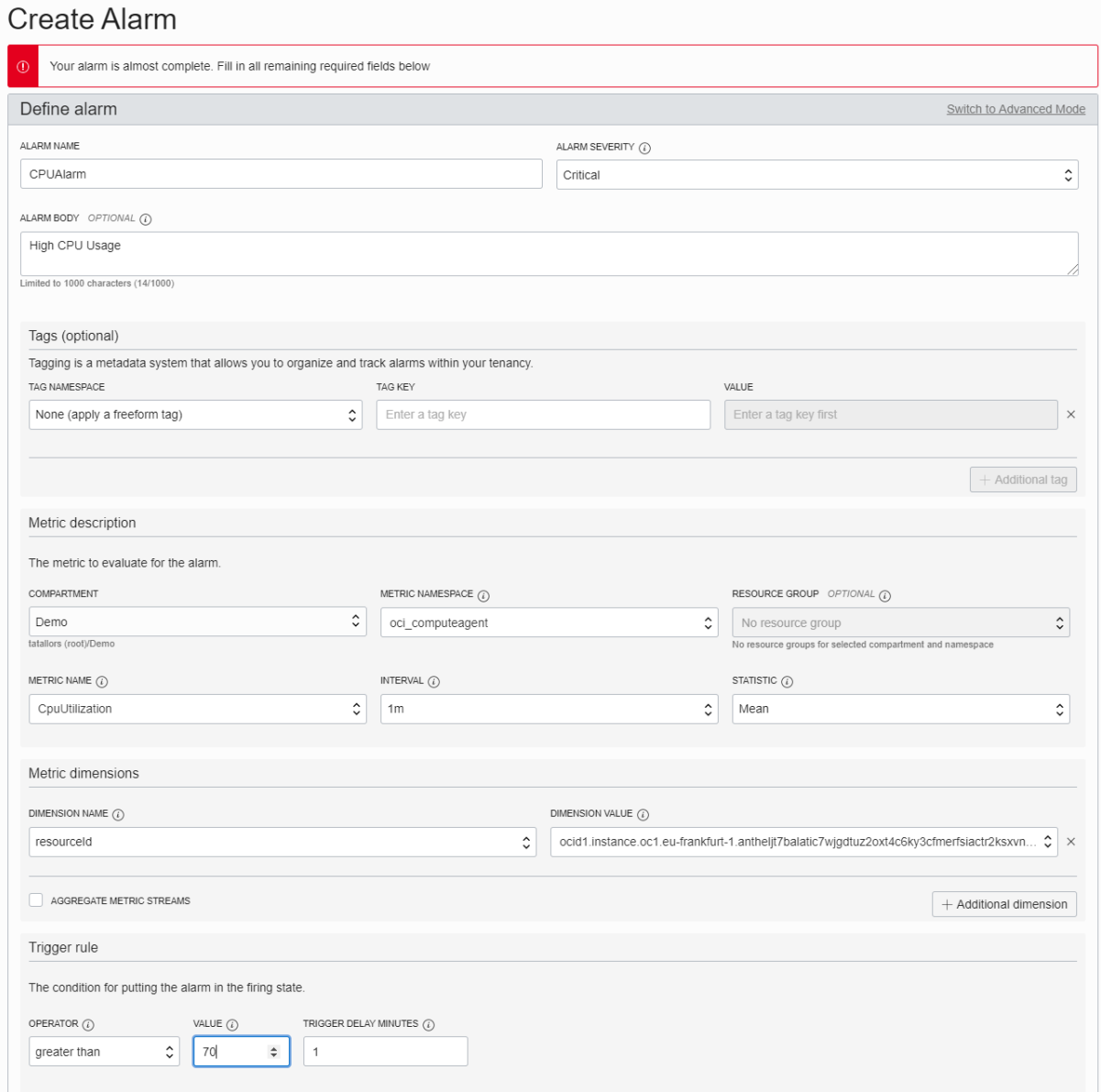
Statistic: mean

Dimension name: should be already populated

Dimension value: should be already populated

Trigger rule:

- Operator: greater than
- Value: 70
- Trigger delay minutes: 1



Keep scrolling down, now you have to add a notification:

Notifications

Destination service: Notification Service

Check the Repeat Notification check box

Notification frequency: 5 minutes

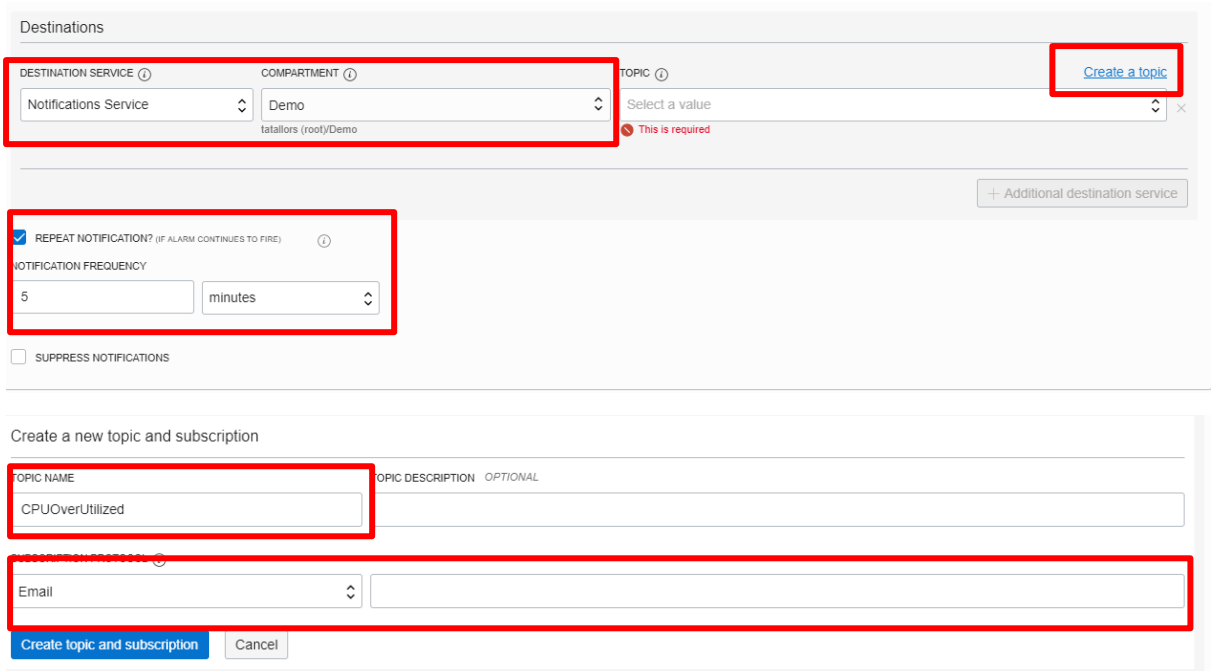
Select the **Create a Topic** button:

Topic Name:CPUOverUtilized

Subscription Protocol: email

Add your email address where to receive the notifications

Select **Create Topic and Subscription**

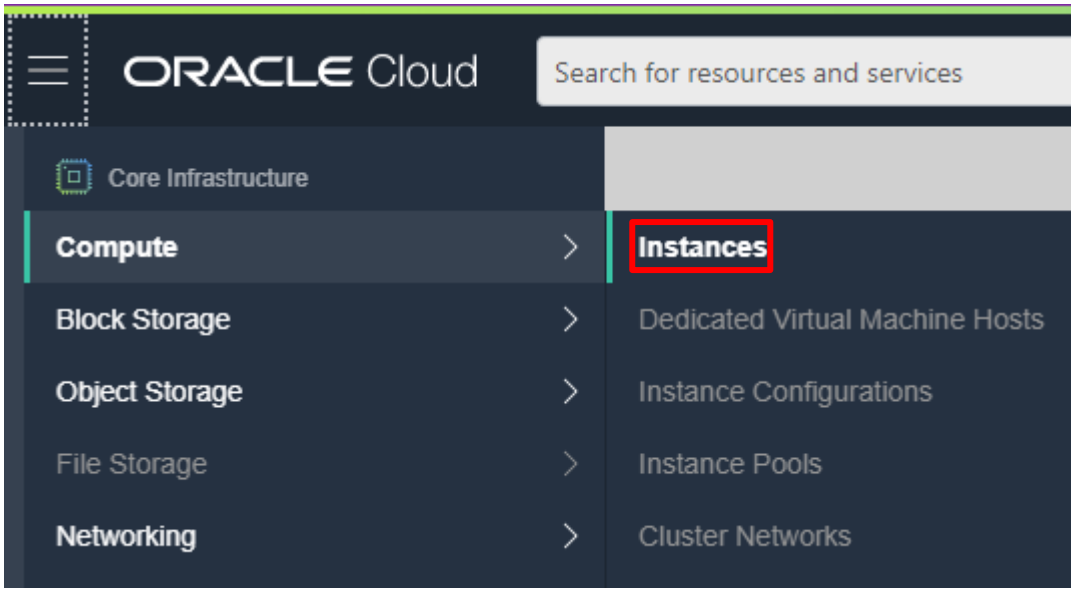


Select the **Save Alarm** button.

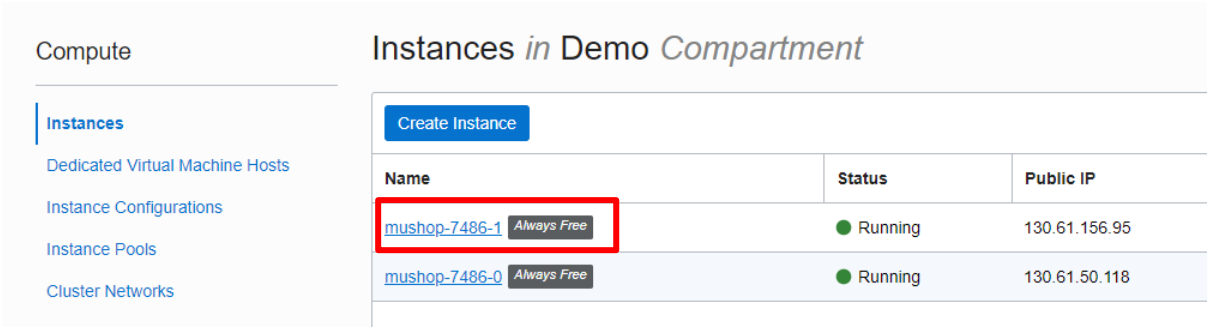
Now that you have created an alarm check your inbox and confirm the subscription for the alarm service, so you can receive notifications.

Now let's generate some load to trigger the alarms.

Navigate to **Main Menu**, **Compute**, **Instances**

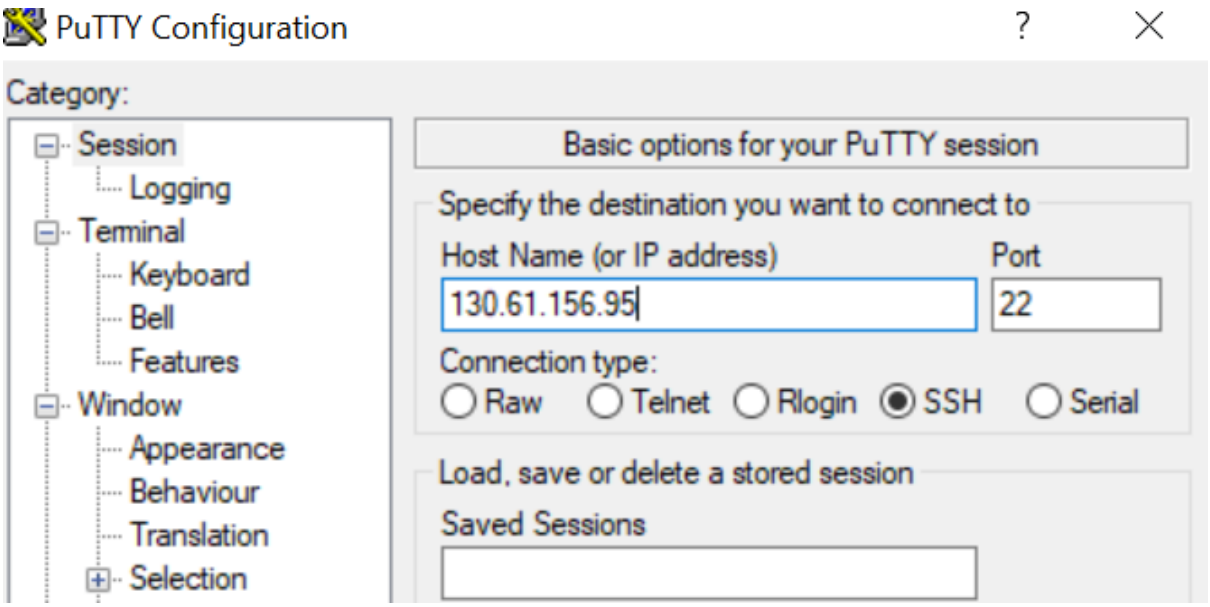


Select the first of the two computer instances:

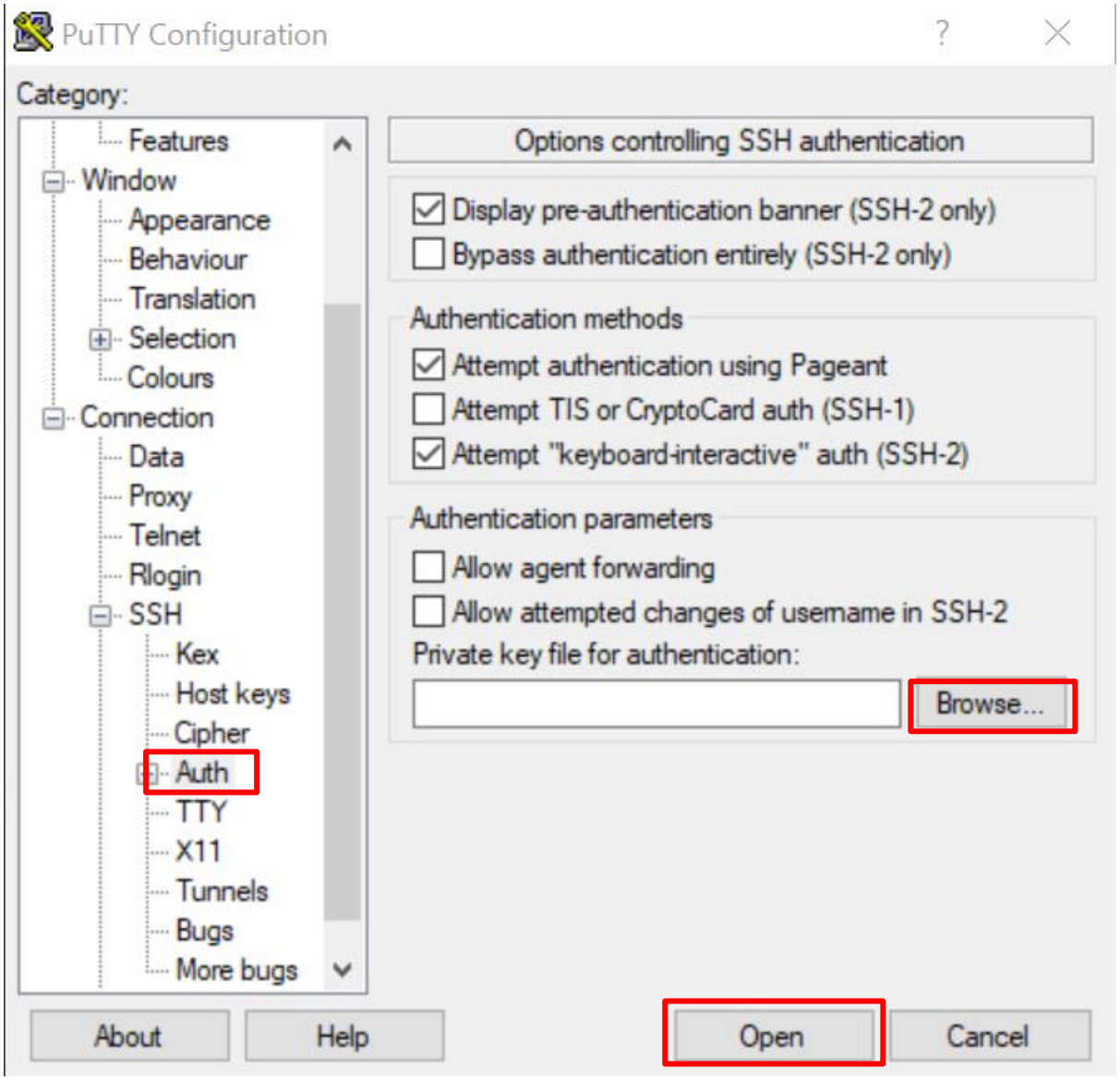


Make a note of the public IP of the instance and create a new connection using Putty.

Enter the public IP in the Host Name (or IP address) text box



Also navigate to **Connection, SSH, Auth** using the **Category:** tree , select the Browse button and add the private key corresponding to the public key you have used when creating the Resource Manager stack and select Open and log in as the “opc” user:



Now let's install the stress application. Run the following two commands:

```
sudo yum install -y stress
```

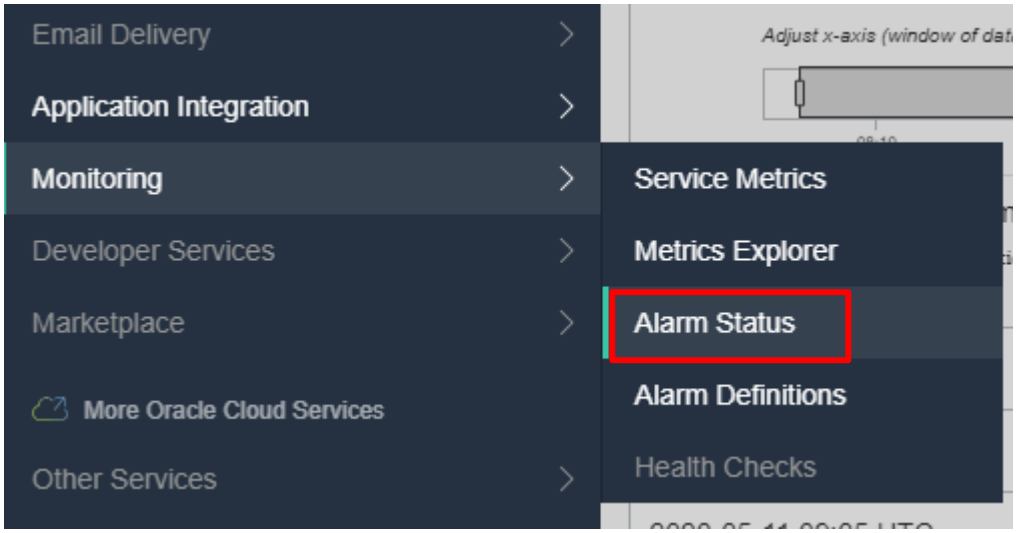
Once the installation process is finished, load the CPU by running the following command:

```
stress -c 10 -t 300
```

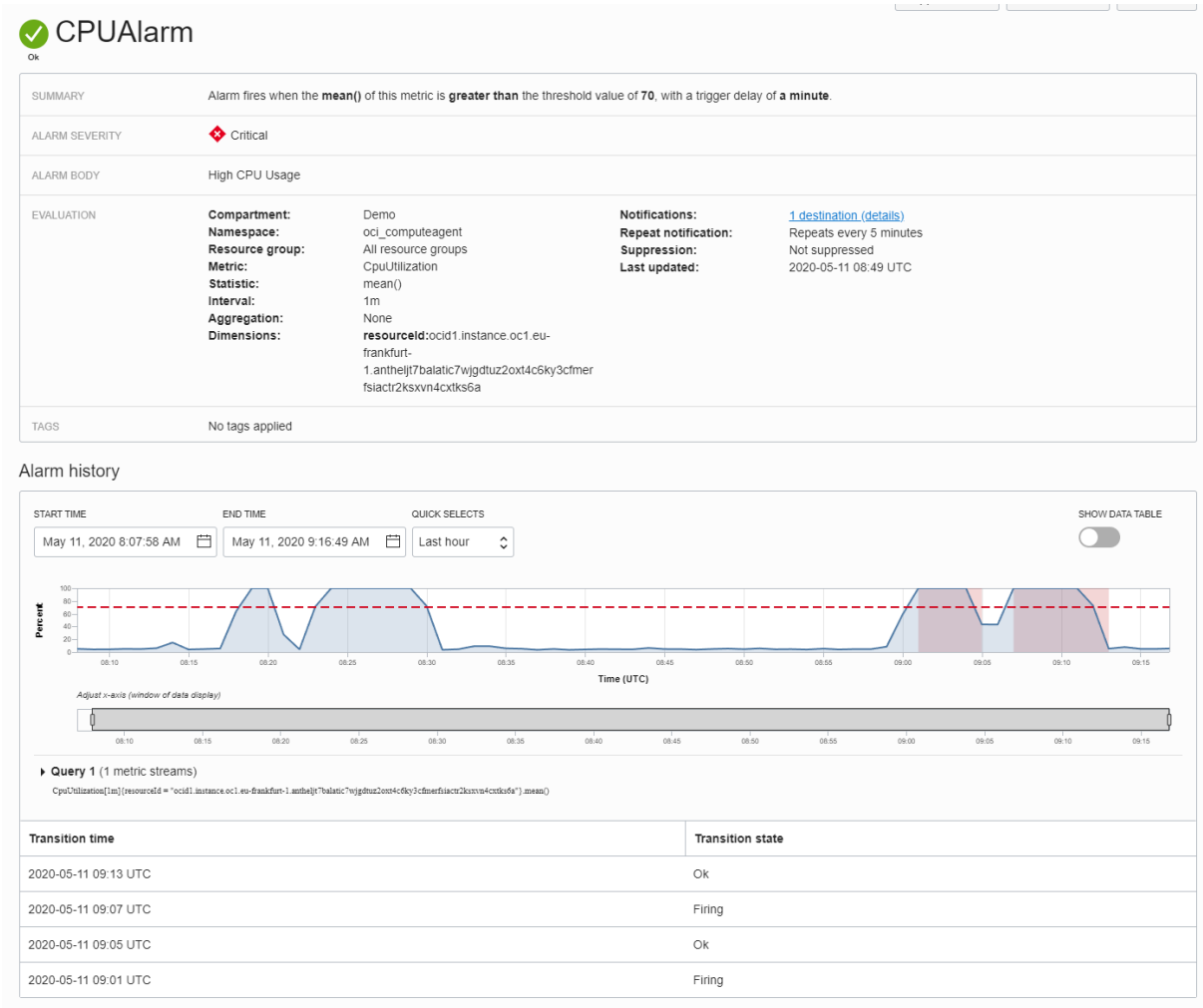
This will overload the CPU for 5 minutes.

You should check your inbox and verify that the alarm notifications have arrived.

Navigate to **Main Menu, Monitoring, Alarm Status** and check the status of your current alarm:

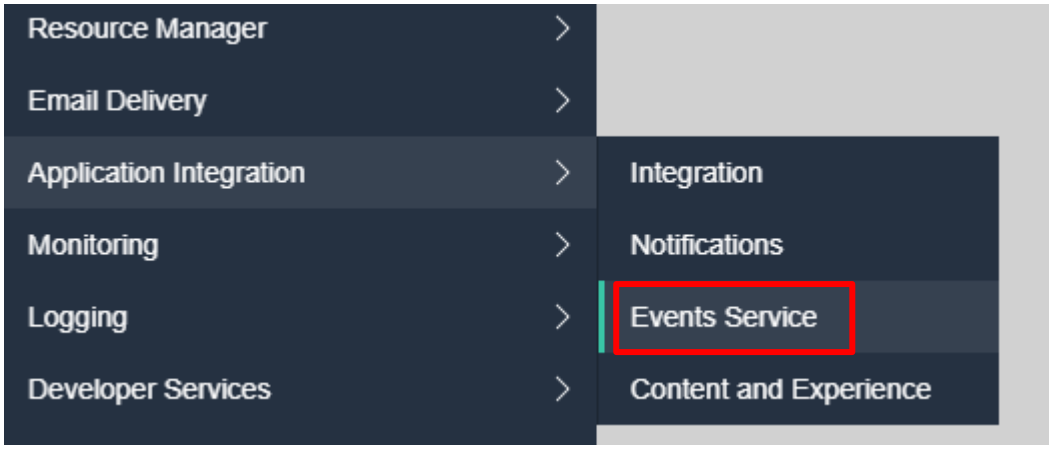


Select the alarm you have created and verify the current and historical state:



8. Set an Instance Termination event and notification

In order to create an event you need to navigate to Main Menu, Application Integration, Events Service



And select the Create Rule button:

Select the following:

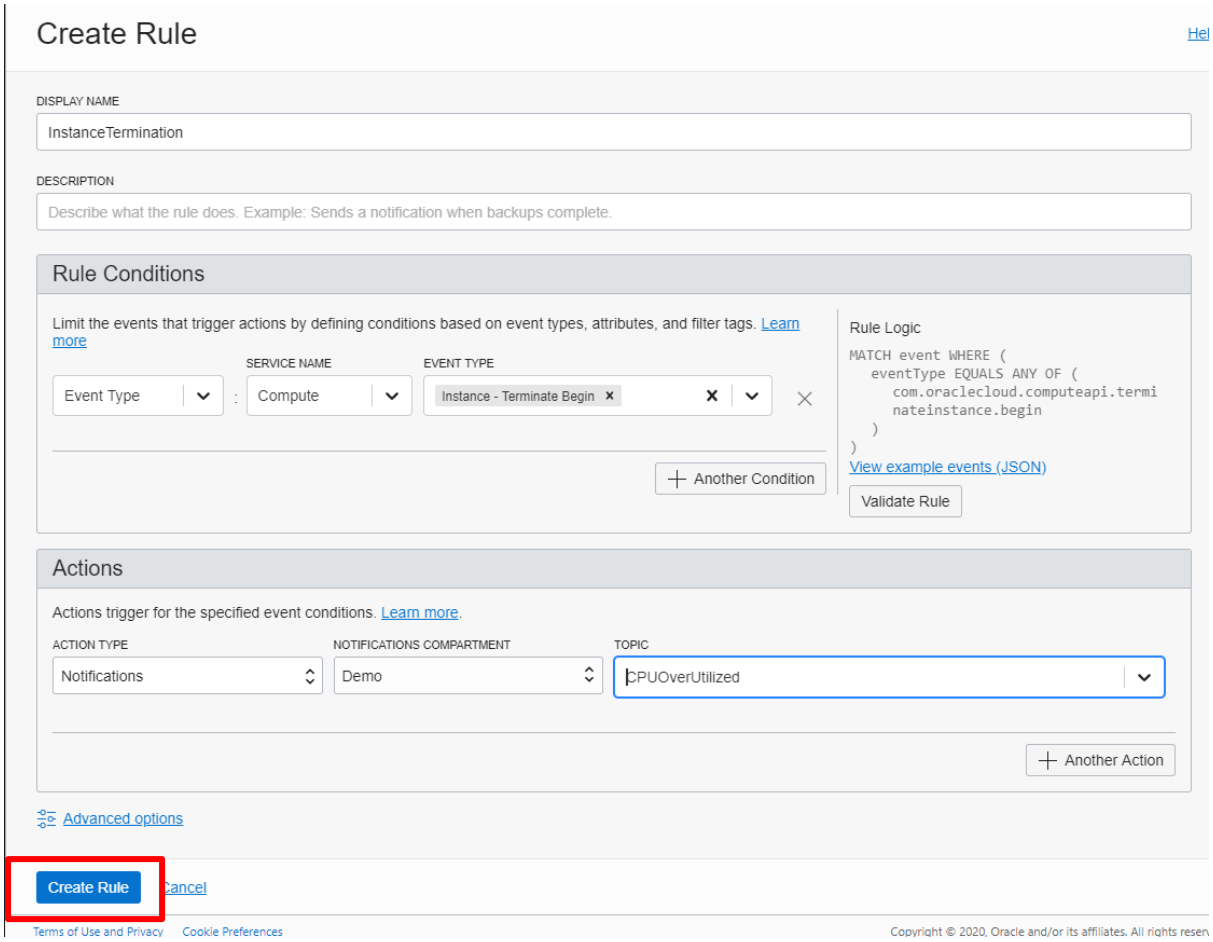
DisplayName: InstanceTermination

Sevice Name: Compute

Event Type: Instance – Terminate Begin

Action Type: Notifications

Topic: CPUOverUtilized



Once the event is created, proceed and terminate one of the compute instances you have provisioned and check your inbox for the notification.