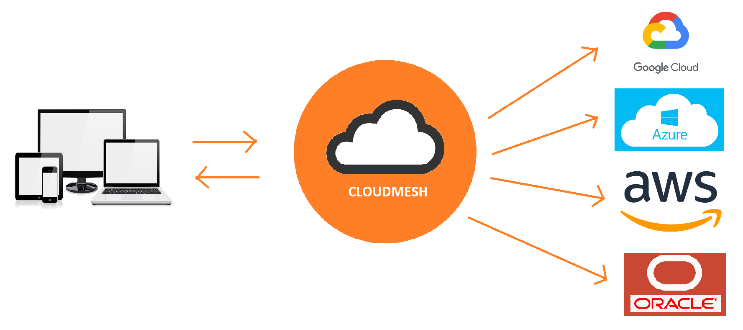
Oracle cloud integration with cloudmesh

Shivani Katukota   
*Intelligent Systems Engineering*  
*Indiana University*Bloomington, Indiana  
shkatu@iu.edu

*Abstract* - Cloudmesh is a multi-cloud project aimed at easy access to computing as well as storage resources. Currently, cloudmesh integrates four cloud service providers: AWS, Azure, Openstack and Google. The aim of this project is to integrate Oracle cloud to cloudmesh.

**

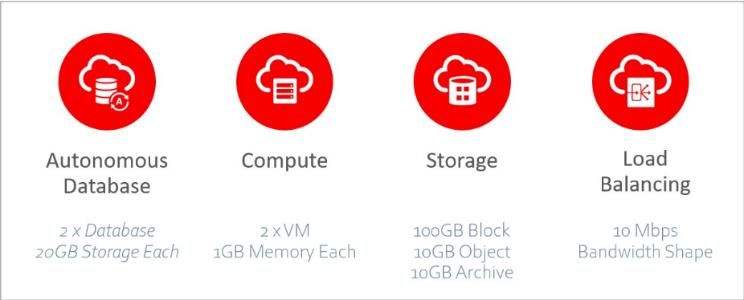
Keywords - Cloudmesh, cloud, oracle, free-tier, multi-cloud

# Introduction

A multi-cloud environment uses resources from multiple cloud providers. The resources can be computing resources, storage services, database services, or any of the several different services offered by the vendors. A multi-cloud environment gives companies the flexibility to use clouds for services that are better managed by them, avoids vendor lock-in, and is more reliable in the event of a disaster. Using multiple clouds also increases the storage capacity and computing power of the company's network.

Cloudmesh aims to bring multiple clouds to a single platform. Via a simple command-line tool, Cloudmesh aims to seamlessly integrate tasks such as the creation and utilization of virtual machines from different cloud providers. The user can switch clouds using a single variable. Hence, he does not need to know the process of VM creation on all the clouds.

On 16th September 2019, Oracle announced a free tier. This allows students and professionals to build, test, and deploy applications on the Oracle cloud and database for free for an unlimited time. The free-tier, though, has limits for various services. Two virtual machines and two block volumes, and 10 GB each of object and archive storage is the current limit for free cloud tier accounts. This project aims to integrate the Oracle cloud with Cloudmesh. This will include figuring out the python SDK for the Oracle cloud called OCI (Oracle Cloud Infrastructure). Using OCI, we will allow Cloudmesh to connect and use computing as well as storage resources on the Oracle cloud.

[](https://github.com/cloudmesh/cloudmesh-manual/blob/master/docs-source/source/accounts/images/oracle/free-tier-services.png)

*Figure: Free services on the Oracle cloud.*

# Advantages

1. Reliability- There is uncertainty about a single cloud's reliability. Accidents happen and having down-times can significantly affect business. Hence, organizations are moving towards multi-cloud environments for a reliable network.
2. Price- The deployments on the cloud are price-sensitive. Organizations are charged for what they use. If a cloud provider increases their prices by even a small margin, it can lead to a loss for large organizations that depend solely on a single cloud provider.
3. Data sovereignty- It is the idea that if data is stored in a digital form, then it is subjected to the laws of the country it is located in. The cloud providers have their data centers in various locations across the globe. Organizations are hence more comfortable with using data centers nearest to their location for sensitive data.
4. Vendor lock-in- It is a very tedious and expensive task to switch between cloud providers. It is hence more advisable to use the multi-cloud environment as then moving data from one cloud to another will not take significant effort.
5. Specific Services- Some cloud providers such Google provide specialized services for IoT and ML. Organizations that are using a different cloud provider but require such services will have to depend on different vendors.

# Current Trends

In a recent survey by Gartner, over 81% of the public cloud users said that they are working with two or more clouds [4]. The main reason they mentioned is to avoid vendor lock-in. In current times, the systems are built to be modular. It is straightforward for organizations to utilize different clouds for their modular systems. This decision also stems from the fact that organizations try to cut costs wherever possible and use resources that best fits their needs at the time. If for any reason, they would like to move to a different cloud, it should be easy. For this reason, Gartner also predicts that by 2021, over 75% of the mid-size and large organizations will move to multi-cloud or hybrid cloud strategies [5].

# Account Setup

The first step to start integrating the Oracle cloud with Cloudmesh is to set up an account on the Oracle cloud free-tier. The credentials of the Oracle cloud will then be added to cloudmesh.yaml, the config file used for Cloudmesh.

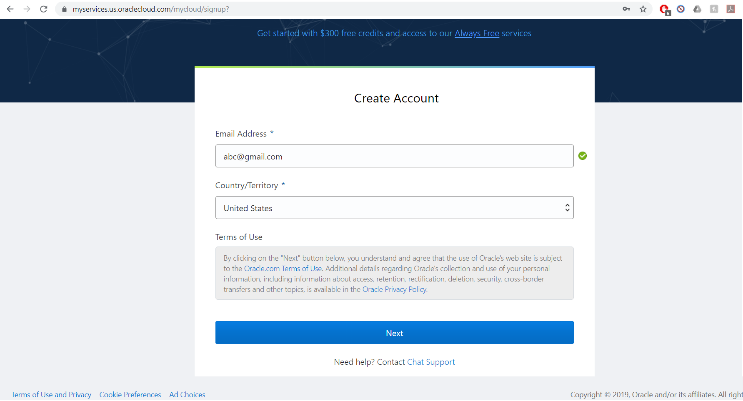
## Account Creation

Following steps are required for creation of a new oracle cloud account and set up the keys and values required for the Cloudmesh config file.

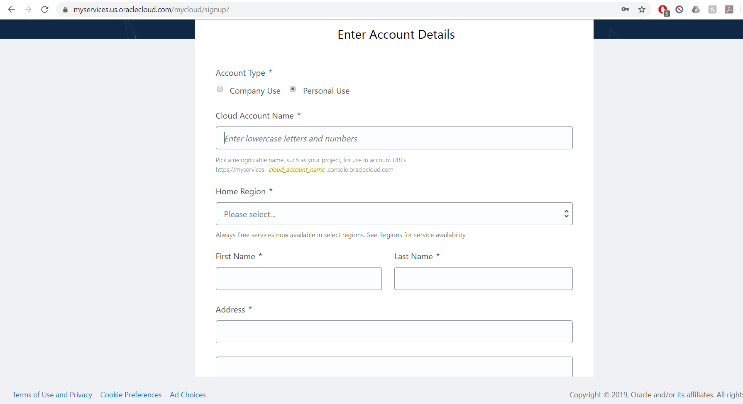
Step 1: Visit the Oracle Cloud website to sign up for a free account.

<https://myservices.us.oraclecloud.com/mycloud/signup?>

Step 2: Fill out the email address and click on Next.

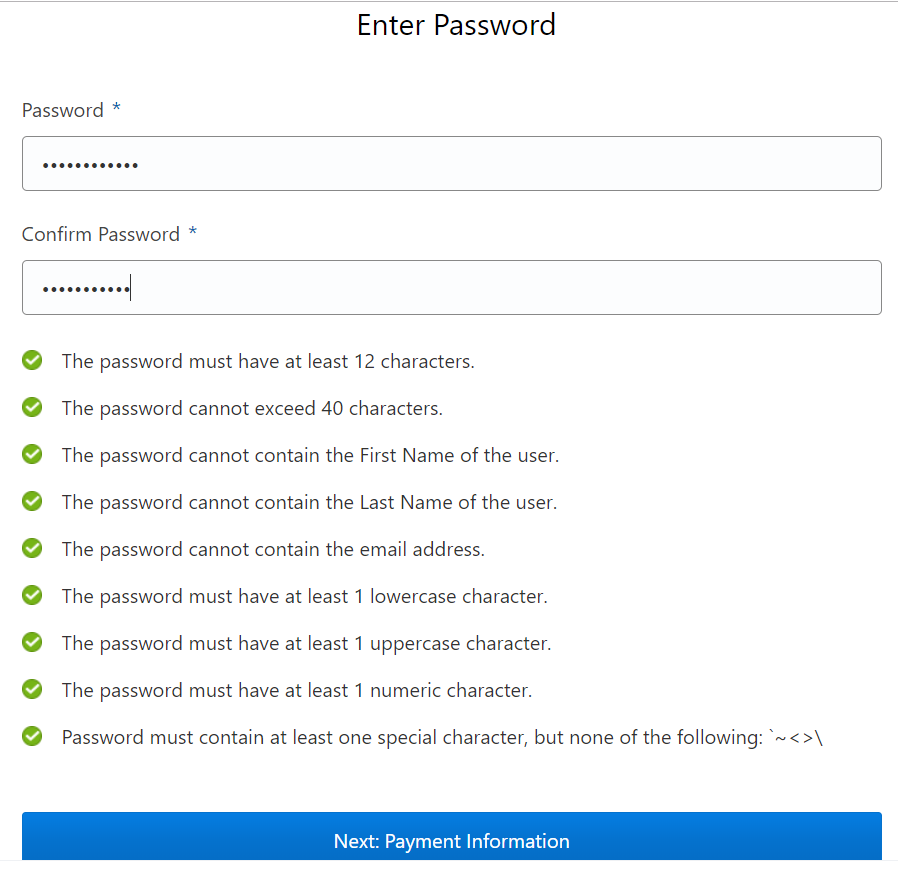
[](https://github.com/cloudmesh/cloudmesh-manual/blob/master/docs-source/source/accounts/images/oracle/create-new-account.png)

Step 3: This will be directed to a page to enter account details. Selected Account Type as Personal and filled out all the other fields and click Next.

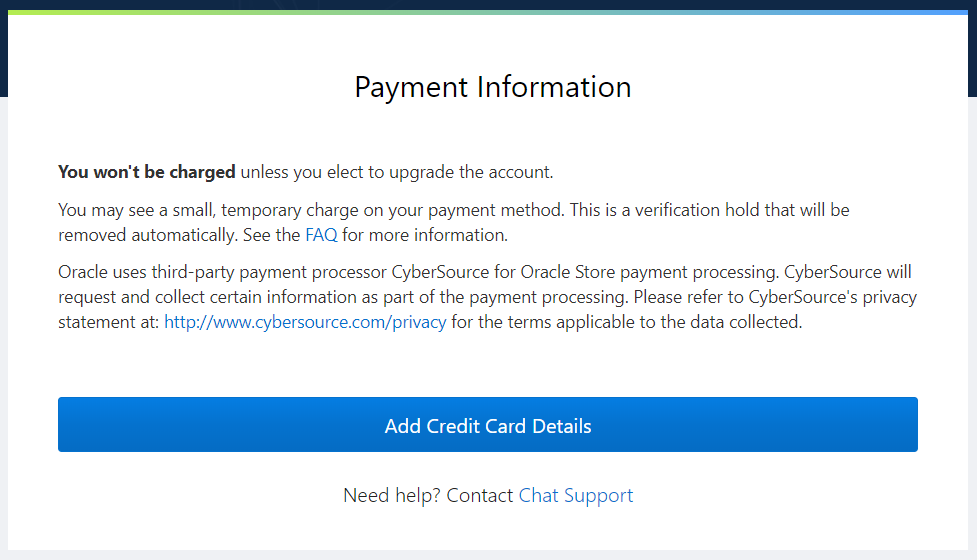
[](https://github.com/cloudmesh/cloudmesh-manual/blob/master/docs-source/source/accounts/images/oracle/account-details.png)

Step 4: Verify the mobile number using the code that will be messaged.

Step 5: Next, a password needs to be selected and click 'Next'.

[](https://github.com/cloudmesh/cloudmesh-manual/blob/master/docs-source/source/accounts/images/oracle/password.png)

Step 6: Fill out your payment information and proceed.

[](https://github.com/cloudmesh/cloudmesh-manual/blob/master/docs-source/source/accounts/images/oracle/payment-information.png)

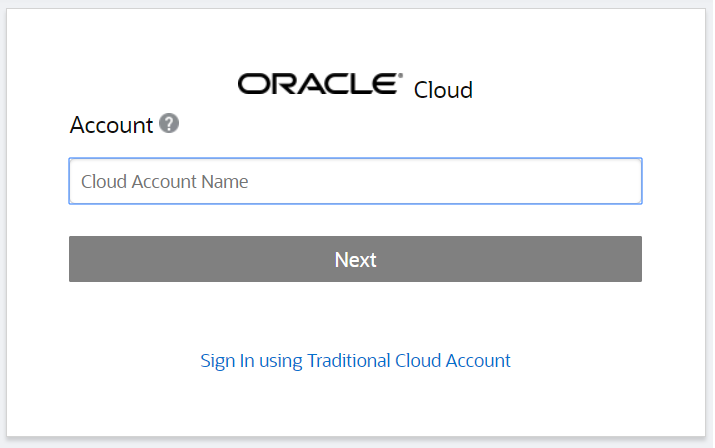
After following these steps, Oracle creates a new free account with $300 credit for 30 days. After 30 days one can still use the 'Always Free' resources. Unless one upgrades, the account will always be free.

[Trial](https://github.com/cloudmesh/cloudmesh-manual/blob/master/docs-source/source/accounts/images/oracle/trial.png)

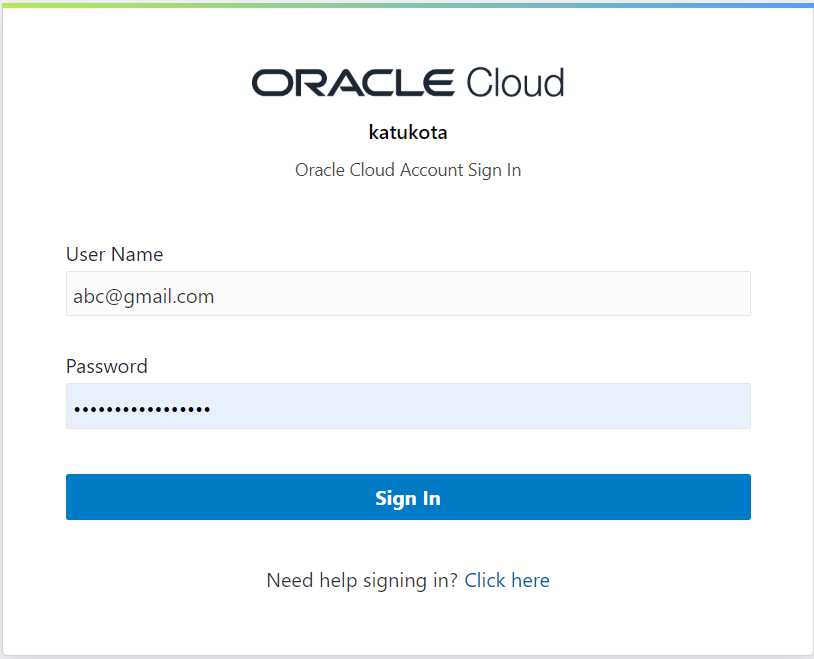
## Account Login

To login into an account, following steps are required.

Step 1: Visit the website <https://www.oracle.com/cloud/sign-in.html?>. Enter the cloud account name used while registration and click Next.

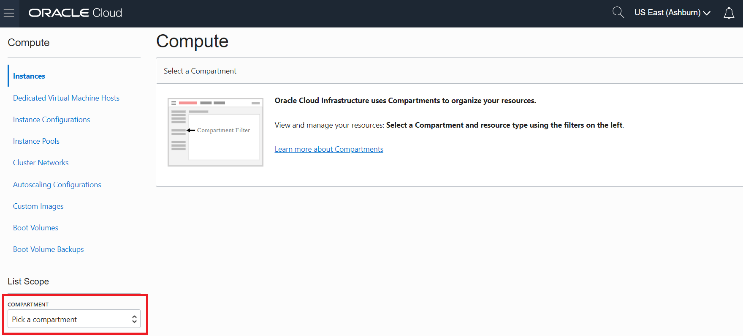
[](https://github.com/cloudmesh/cloudmesh-manual/blob/master/docs-source/source/accounts/images/oracle/login.png)

Step 2: Enter the email address and password you used for registration, and click on Sign In.

[](https://github.com/cloudmesh/cloudmesh-manual/blob/master/docs-source/source/accounts/images/oracle/login-next.png)

Now you've successfully signed into the Oracle Cloud.

You can click on the menu on the left to explore different oracle services. When you try to access compute or storage services, you will first need to select a compartment. A compartment is a logical container to organize you resources. A default compartment with the same name as your cloud name is created. It is easiest to select that compartment.

[](https://github.com/cloudmesh/cloudmesh-manual/blob/master/docs-source/source/accounts/images/oracle/compartment.png)

## Oracle API

To develop or run code related to oracle, its API called OCI (Oracle Cloud Infrastructure) needs to be installed. To do this, following command is run using the command line.

$ pip install oci

## Cloudmesh Config File

Cloudmesh config file (cloudmesh.yaml) will have the following entries for compute and storage respectively.

### Compute Entry

cloudmesh:

...

compute:

oracle:

cm:

active: true

heading: ORACLE

host: cloud.oracle.com

label: oracle

kind: oracle

version: TBD

service: compute

default:

image: Oracle-Linux-7.7-2019.11.12-0

size: VM.Standard.E2.1

key: TBD

credentials:

user : TBD

fingerprint : TBD

key\_file : ~/.oci/oci\_api\_key.pem

pass\_phrase : TBD

tenancy : TBD

compartment\_id : TBD

region : us-ashburn-1

### Storage Entry

cloudmesh:

...

storage:

cm:

active: true

heading: Oracle

host: cloud.oracle.com

label: oracle

kind: oracle

version: TBD

service: storage

default:

directory: TBD

bucket: home

credentials:

user : TBD

fingerprint : TBD

key\_file : ~/.oci/oci\_api\_key.pem

pass\_phrase : TBD

tenancy : TBD

compartment\_id : TBD

region : us-ashburn-1

## Required Keys and OCIDs

The keys marked as TBD in the config file needs to be filled in using the values from the Oracle account. To do this, following steps are required.

### Create API Signing Key

Following commands are run using command line.

Step 1: To store the credentials, a directory called .oci is created.

$ mkdir ~/.oci

Step 2: Generate the key using the command:

$ openssl genrsa -out ~/.oci/oci\_api\_key.pem -aes128 -passout stdin 2048

Step 3: You will then be prompted for a passphrase. Select a passphrase and hit Enter.

Step 4: To ensure that only you can read the key, run the following command:

$ chmod go-rwx ~/.oci/oci\_api\_key.pem

Step 5: Now, generate the public key and hit enter. You will again be prompted for the passphrase. Please enter the passphrase created while creating the private key and hit Enter.

$ openssl rsa -pubout -in ~/.oci/oci\_api\_key.pem \

-out ~/.oci/oci\_api\_key\_public.pem \

-passin stdin`

Step 6: Copy the contents of the public key to the clipboard as you will require this.

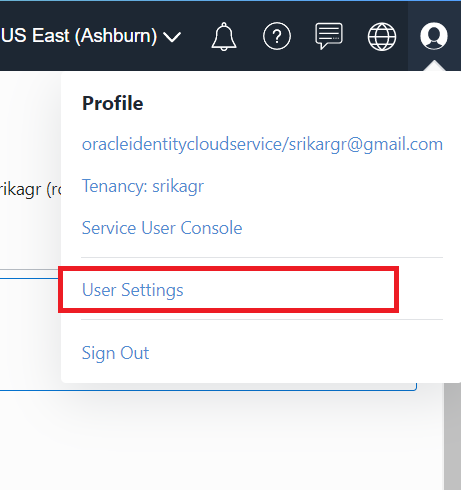
Step 7: Add the passphrase to your config file cloudmesh.yaml under oracle section.

cms register oracle [--dir=~/.oci]

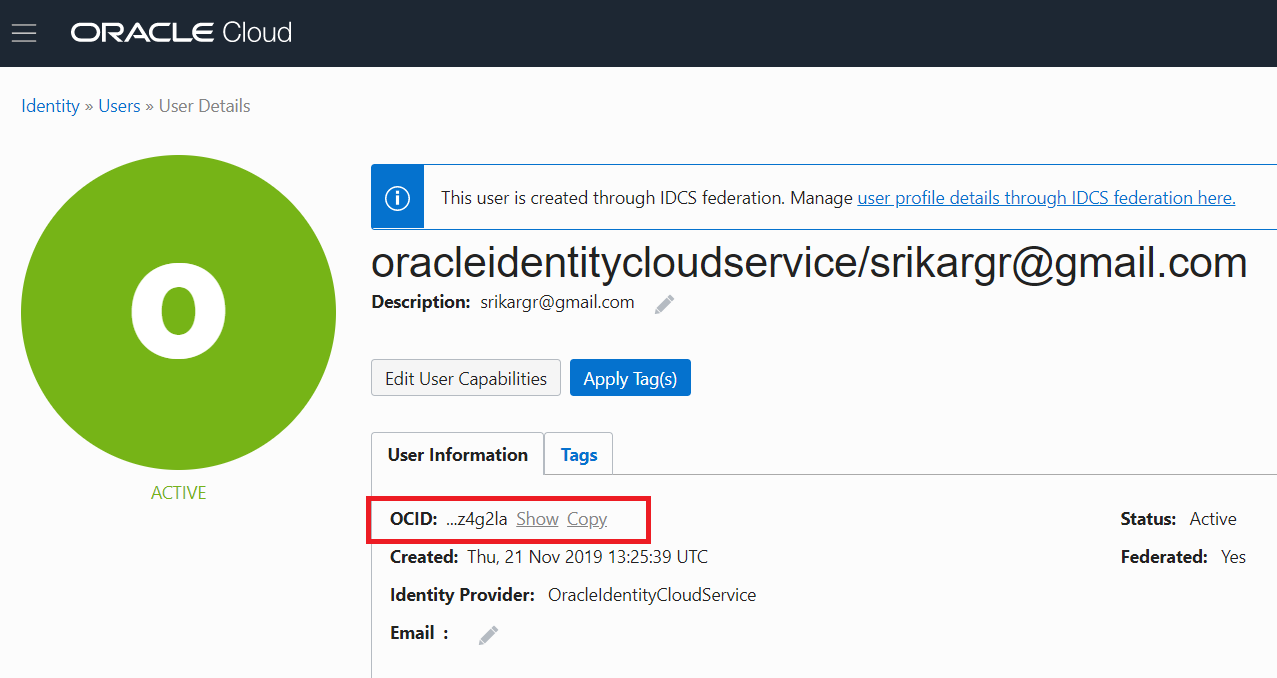
### User

The value for user in the config file is the OCID of the user of the oracle cloud account. You can find this using the following steps:

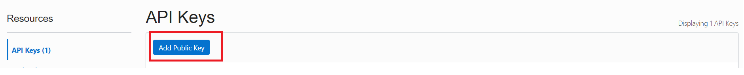
Step 1: Click on the profile menu on the right and click on User Settings.

[](https://github.com/cloudmesh/cloudmesh-manual/blob/master/docs-source/source/accounts/images/oracle/user-settings.png)

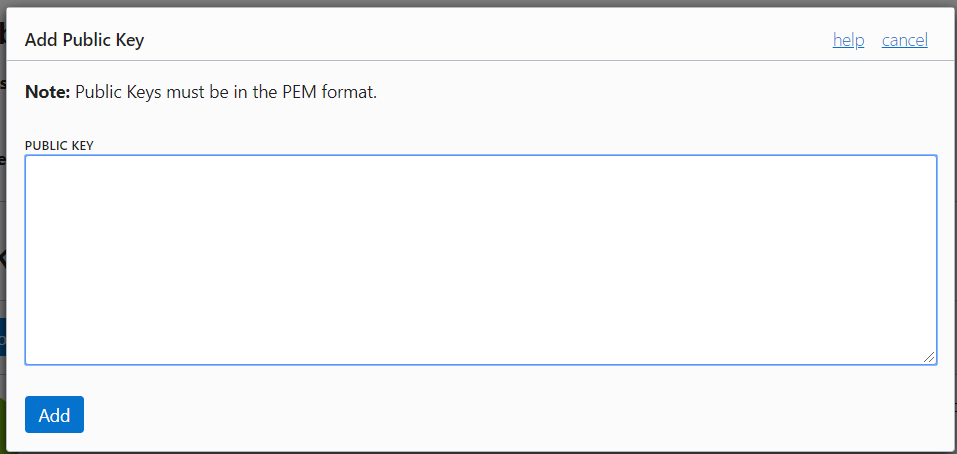
Step 2: The user OCID can be found under User Information. Copy and paste it to the config file.

[](https://github.com/cloudmesh/cloudmesh-manual/blob/master/docs-source/source/accounts/images/oracle/user.png)

Step 3: Go to the end of web page and click on Add Public Key under API Keys.

[](https://github.com/cloudmesh/cloudmesh-manual/blob/master/docs-source/source/accounts/images/oracle/public-key.png)

Step 4: Paste the contents of the public key ~/.oci/oci\_api\_key\_public.pem to the text-box in the pop-up and click Add.

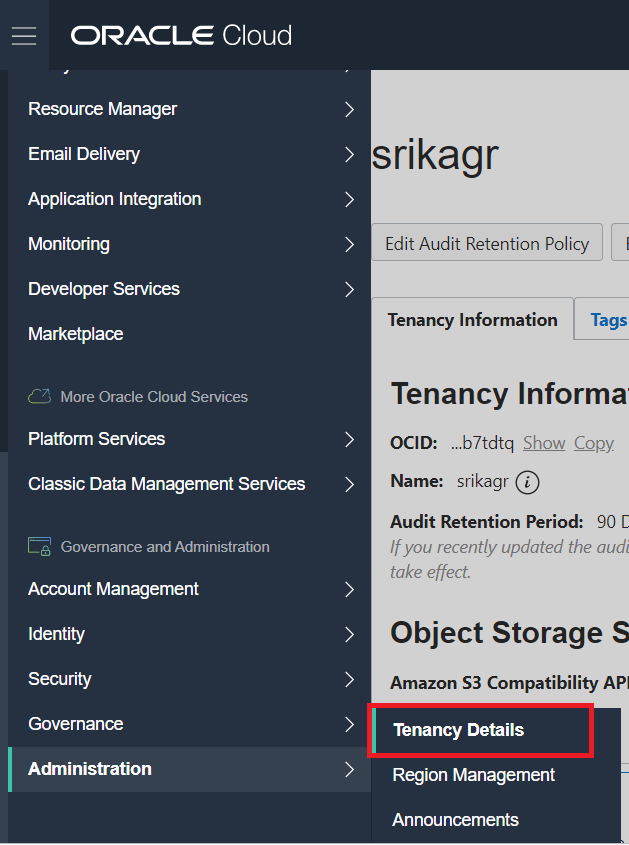
[](https://github.com/cloudmesh/cloudmesh-manual/blob/master/docs-source/source/accounts/images/oracle/add-public-key.png)

Step 5: A new key will be added to the API Keys. Copy the Fingerprint of the key and paste it in the config file.

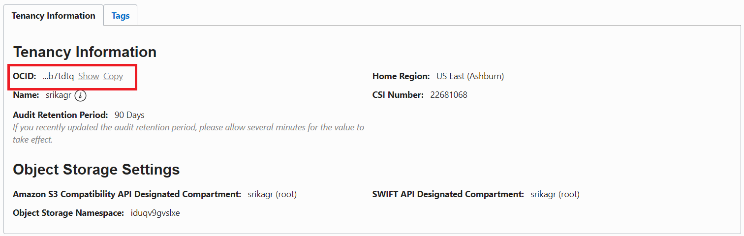
### Tenancy

The value for tenancy in the config file is the OCID of your tenancy account. You can find this using the following steps:

Step 1: Click on the navigation menu on the left, go to Administration and click on Tenancy Details.

[](https://github.com/cloudmesh/cloudmesh-manual/blob/master/docs-source/source/accounts/images/oracle/tenancy.png)

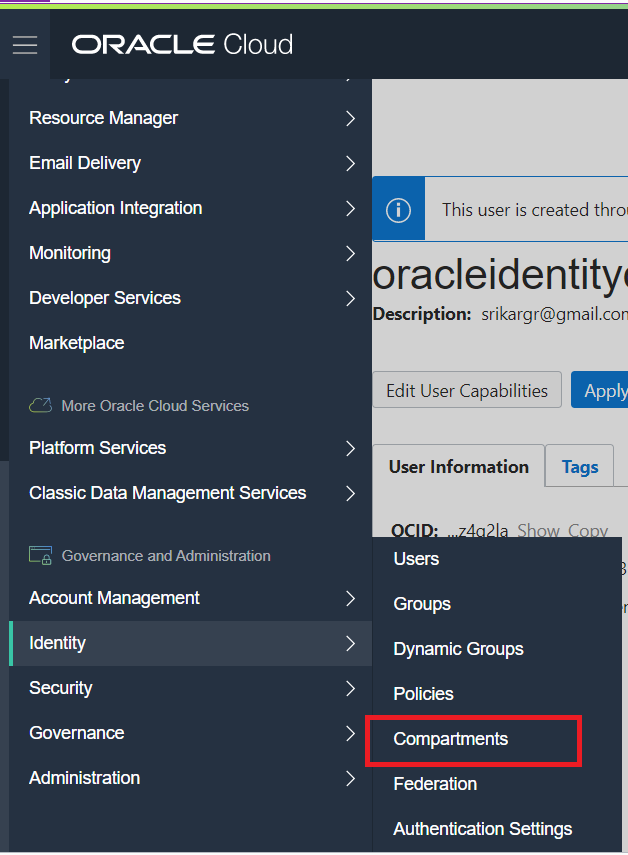
Step 2: The tenancy OCID can be found under Tenancy Information. Copy and paste it to the config file.

[](https://github.com/cloudmesh/cloudmesh-manual/blob/master/docs-source/source/accounts/images/oracle/tenancy-details.png)

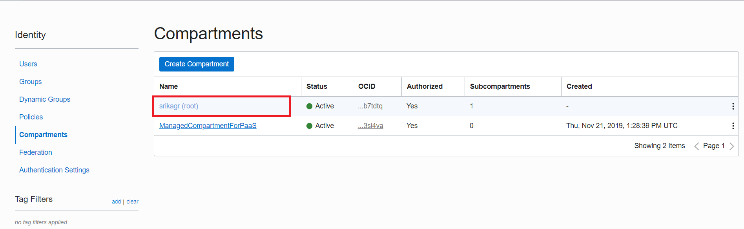
### Compartment

The value for compartment in the config file is the OCID of your selected compartment. You can find this using the following steps:

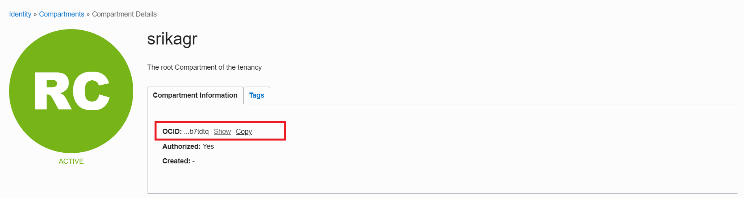
Step 1: Click on the navigation menu on the left, go to Identity and click on Compartments.

[](https://github.com/cloudmesh/cloudmesh-manual/blob/master/docs-source/source/accounts/images/oracle/identity.png)

Step 2: Select your root compartment from the compartment list.

[](https://github.com/cloudmesh/cloudmesh-manual/blob/master/docs-source/source/accounts/images/oracle/select-compartment.png)

Step 3: The compartment OCID can be found under Compartment Information. Copy and paste it to the config file.

[](https://github.com/cloudmesh/cloudmesh-manual/blob/master/docs-source/source/accounts/images/oracle/compartment-details.png)

# Results

## Compute Service

The first thing we need to make sure is that a private public ssh key pair has been set up in the default directory ~\.ssh\id\_rsa.pub. This key will be used to login into the virtual machine instances created by us.

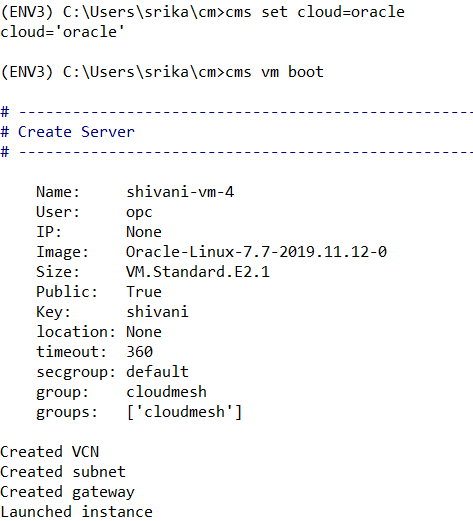
1. To set the cloud to oracle, use the command:

$ cms set cloud=oracle

1. To create a new instance on oracle cloud, use the command:

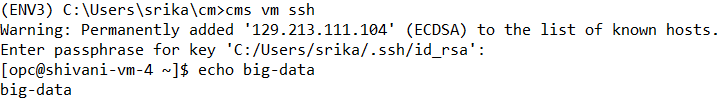
$ cms vm boot

This will create a new virtual machine on the cloud.



1. To login into the instance, use the command:

$ cms vm ssh

This command will login to the newly created virtual machine and one can now run commands on the VM.

1. To stop the instance, use the command:

$ cms vm stop 'vm-name'

1. To terminate the instance, use the command:

$ cms vm stop 'vm-name'

## Storage Service

The first time you try to access storage services, it wil be empty and any operation to list/get/delete will result in an error. Hence, to start first put a new file on the cloud. This will result in creation of a new bucket with the name specified in cloudmesh.yaml.

1. To upload a new file/directory to the cloud, use the command:

$ cms storage --storage=oracle put SOURCE DESTINATION

1. To download a file from the cloud, use the command:

$ cms storage --storage=oracle get SOURCE DESTINATION

1. To list all the files from the bucket/directory, use the command:

$ cms storage --storage=oracle list SOURCE

1. To delete a file/directory from the cloud, use the command:

$ cms storage --storage=oracle delete SOURCE

Note that if a directory is deleted, all the files inside the directory are also deleted.

1. To search a file in a particular directory, use the command:

$ cms storage --storage=oracle search DIRECTORY FILENAME

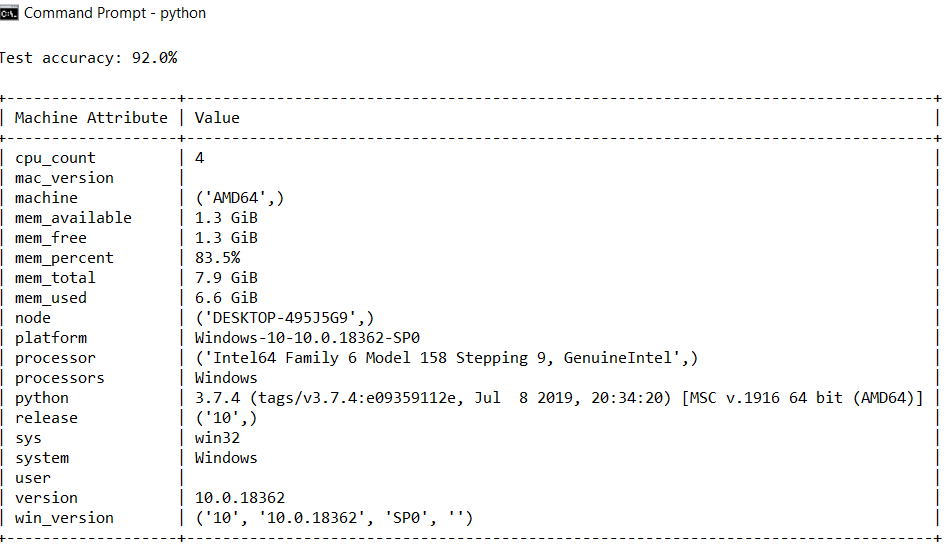
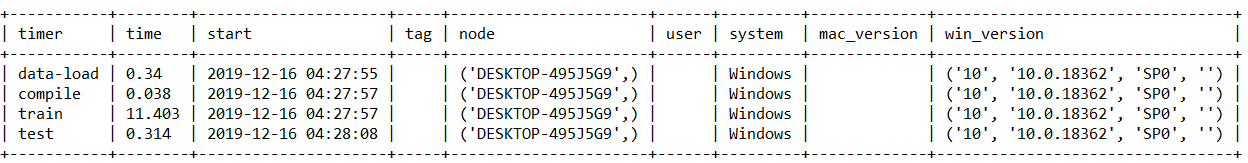
# Mnist Deep Learning Example

The MNIST database contains around 60,000 images of handwritten digits. This database is used as a standard for training various image processing deep learning systems. As a part of this project, a simple neural network with two dense layers and one activation layer was created. This code is saved on the Oracle cloud object storage. We download the file using the Oracle storage put command.

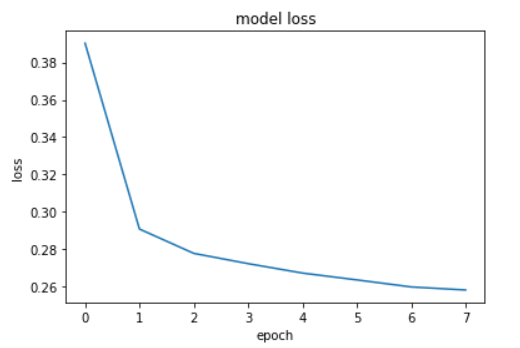
$ cms storage --storage=oracle get ‘big-data\mnist-deep-learning.py’ ‘mnist-deep-learning.py’

The downloaded file is then executed.

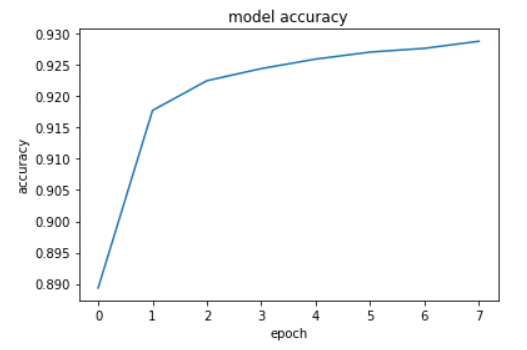
$ exec(open(‘mnist-deep-learning.py’.read())



*Figure: Benchmarks for MNIST example.*



*Figure: Model Loss for MNIST database.*

 *Figure: Model Accuracy for MNIST database.*

# Lessons Learnt

1. Cloud providers offer a lot of different services like computing resources and storage facilities for both structured and unstructured data.
2. Cloud services are ideal resources to store and analyze big data.
3. Learned to use the Python SDK to connect to cloud services.
4. Learned to use GitHub effectively.

# Links To Code Repository

1. Code Directory: <https://github.com/cloudmesh/cloudmesh-oracle>
2. Manual page: <https://github.com/cloudmesh/cloudmesh-manual/blob/master/docs-source/source/accounts/oracle.md>
3. Deep Learning Code: <https://github.com/cloudmesh-community/fa19-516-162/tree/master/big-data>
4. Benchmark: <https://github.com/cloudmesh/benchmark/blob/master/results/cloud-oracle-katukota.txt>

# References

1. Cloud Computing book, <https://laszewski.github.io/book/cloud/>
2. OCI Documentation, <https://oracle-cloud-infrastructure-python-sdk.readthedocs.io/en/latest/index.html>
3. Required Keys and OCIDs, <https://docs.cloud.oracle.com/iaas/Content/API/Concepts/apisigningkey.htm>
4. <https://www.gartner.com/smarterwithgartner/why-organizations-choose-a-multicloud-strategy/>
5. <https://www.gartner.com/en/conferences/apac/infrastructure-operations-cloud-india/featured-topics/cloud>