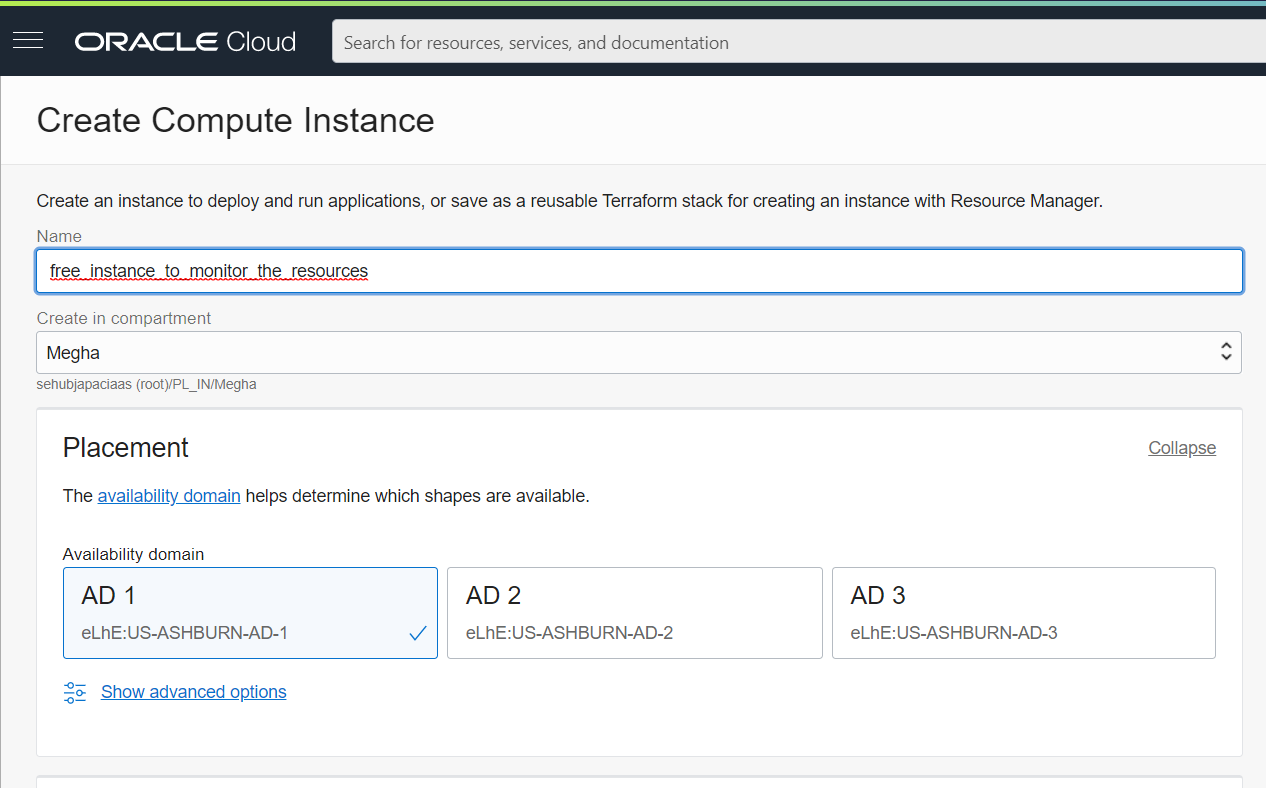
Auto start/stop of instances using python and cronjob

Many of the times we might need to manually shutdown the instances in multiple compartments within the tenancy. It can be a tedious job. What if there are 50 odd instances!! Or even more  
the task can be automated using python script which runs in a free instance available to all the tenancies!

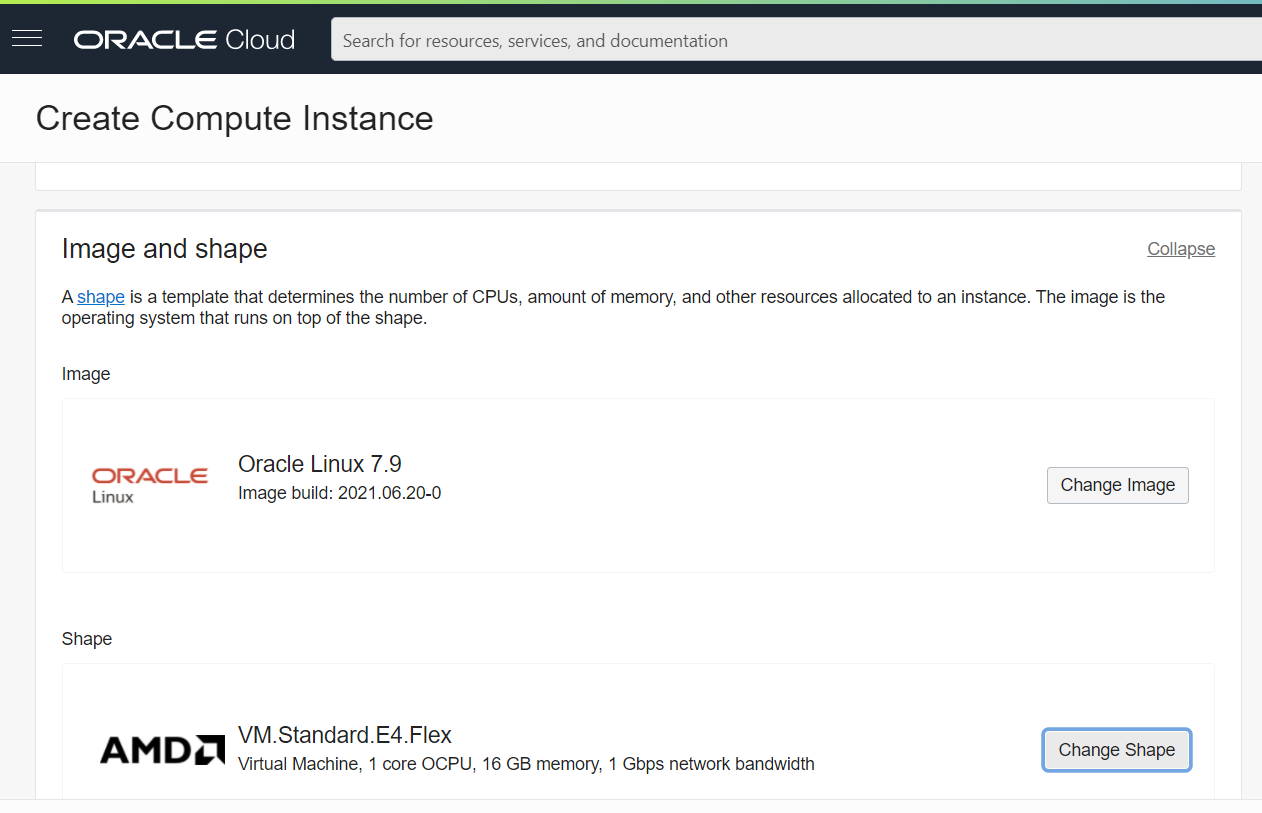
For this task you might need to deploy the code once in your instance and it will do the job until you have instances running in your compartment. No maintenance needed!!

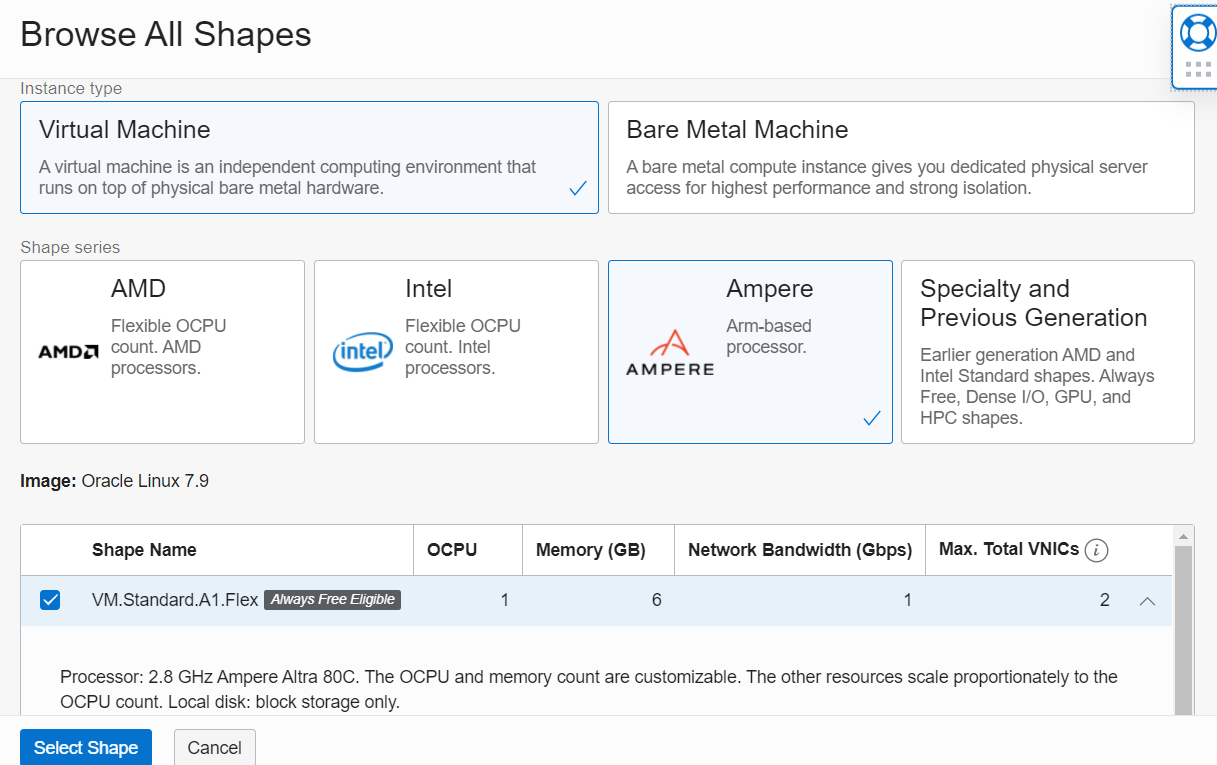
Here are the steps to follow:

1. Sign in to your OCI console.
2. Create a VCN with public subnet in the parent compartment where you want this to be implemented (If you have one you can use it)
3. Create a free ampere shape instance with the image of your choice. (I have used oracle linux image here)

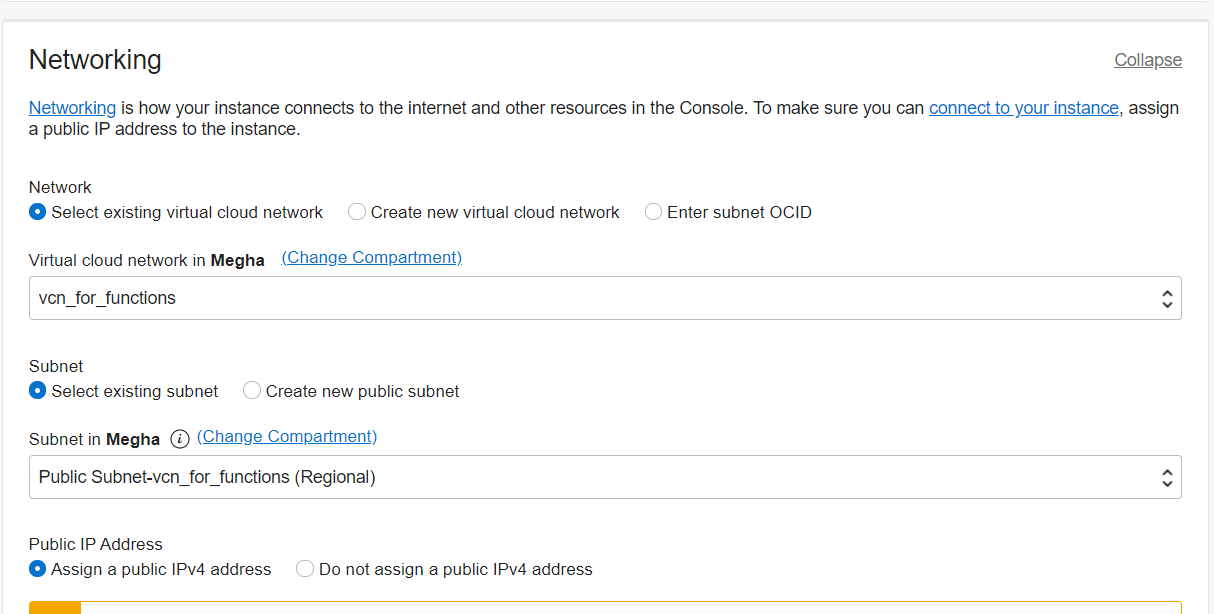


Chose the shape and image.

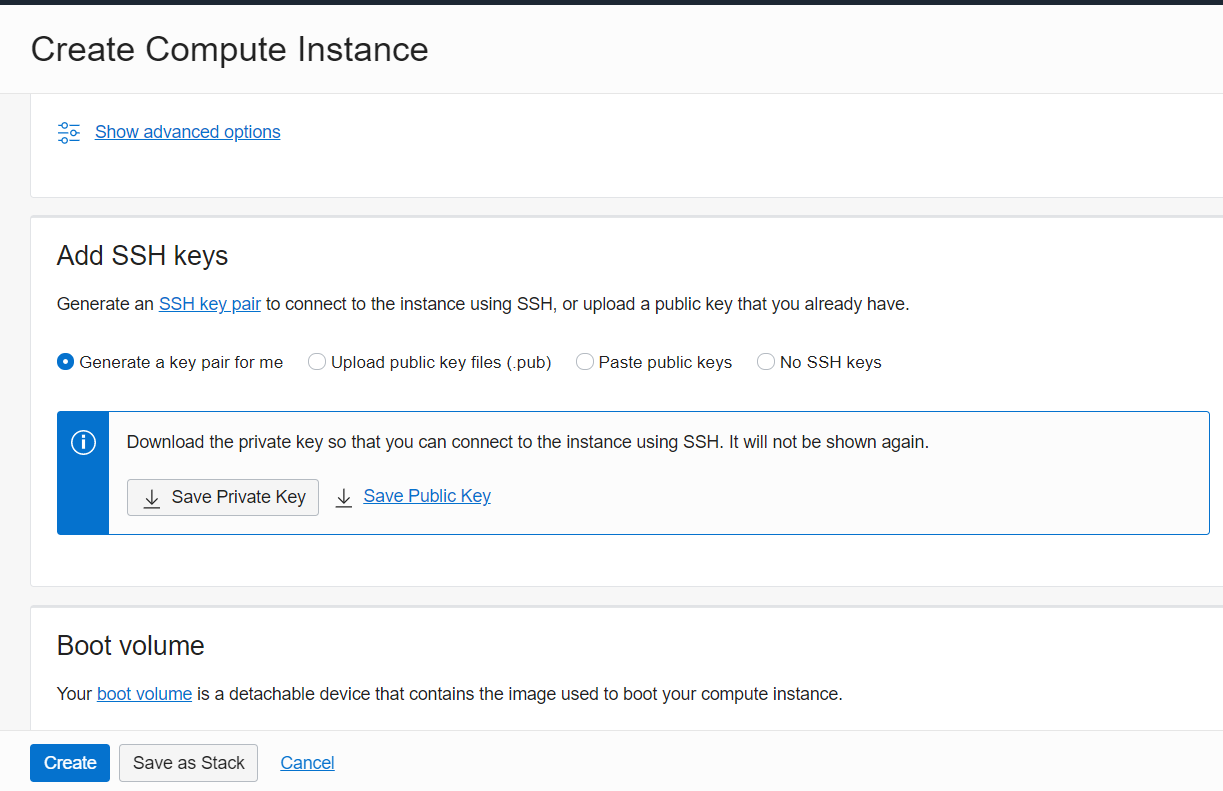




Chose the VCN and the subnet where you want to create this.



Add the SSH keys and click on create button. (for information on how to create the SSH keys click here.)



1. Once the instance is un and running. Log into the instance (I am using PUTTY in this case)

Install the necessary packages required in the instance.

1. Install EPEL Repository:

sudo yum install epel-release  
(if the above commant returns no packages found error try the following commands)  
sudo yum install python3

sudo yum install python3-pip

to check if pip installed or not:

pip –version (this will return the version of pip)  
pip list (provides the list of the packages installed though pip command)  
if you want to upgrade pip:  
pip install --upgrade pip

install other utilities:

sudo python3 -m pip install -U setuptools

install OCI- CLI:

sudo pip install oci-cli

to check the OCI cli installed or not:

oci –version

install pandas package:

sudo pip install pandas

if you are getting setup error, follow: <https://stackoverflow.com/questions/35991403/pip-install-unroll-python-setup-py-egg-info-failed-with-error-code-1>  
<https://appuals.com/command-python-setup-py-egg_info/#:~:text=Fix%3A%20'Command%20%E2%80%9Cpython%20setup,code%201'%20When%20Installing%20Python&text=The%20error%20code%201%20is,to%20be%20installed%20or%20updated>.

Now, download the script and place it within the instance.

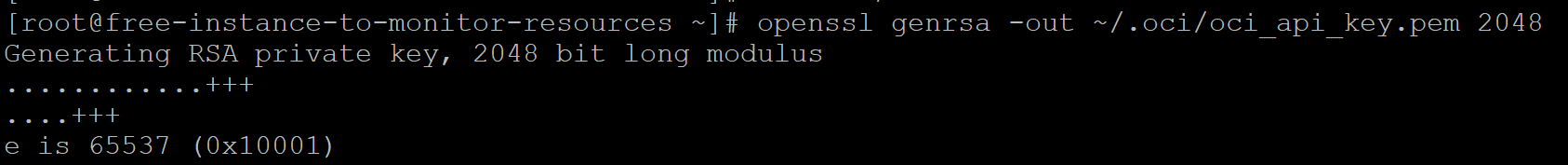
Get OCIDs needed for the configuration file. (tenancy, user, and compartment OCID etc, from console)

Create the API keys needed for configuration.

mkdir ~/.oci

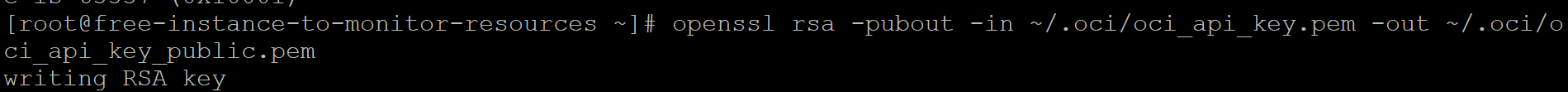


openssl genrsa -out ~/.oci/oci\_api\_key.pem 2048

****

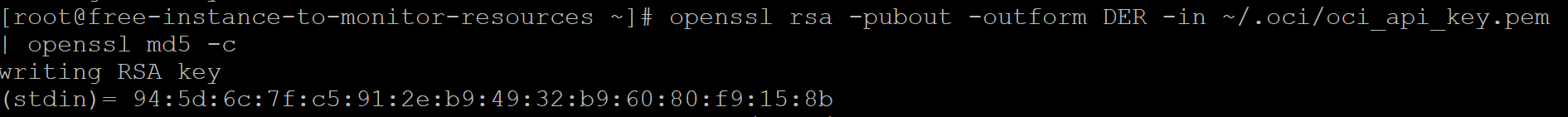
Finally, we generate the public key:

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

****

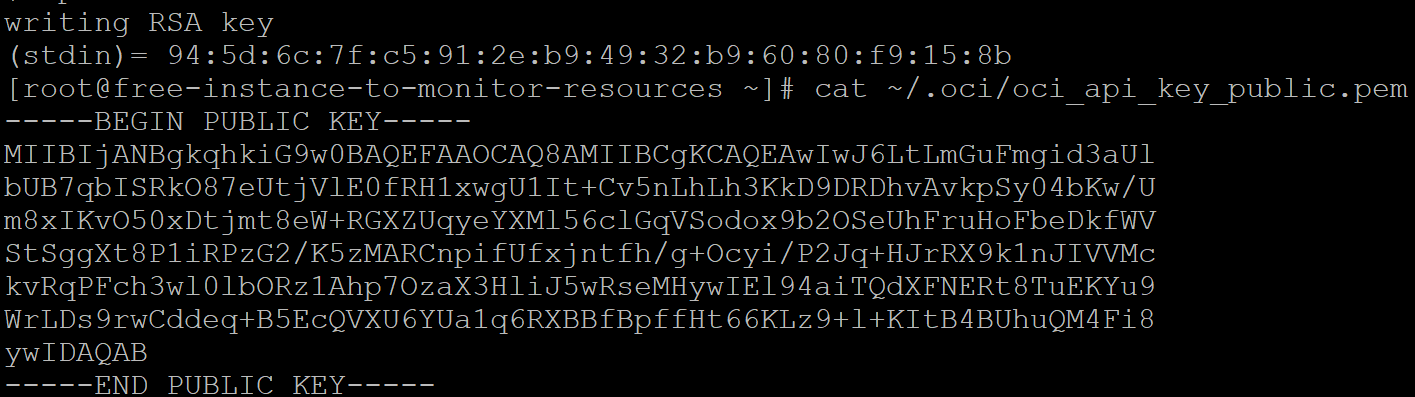
We can run the following command to view the fingerprint. The fingerprint of a key is a unique sequence of letters and numbers used to identify the key. Similar to a fingerprint of two people never being the same.

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



We need to cat the public key file and copy the key output, as we need this to upload the API key to Oracle Cloud Infrastructure.

cat ~/.oci/oci\_api\_key\_public.pem

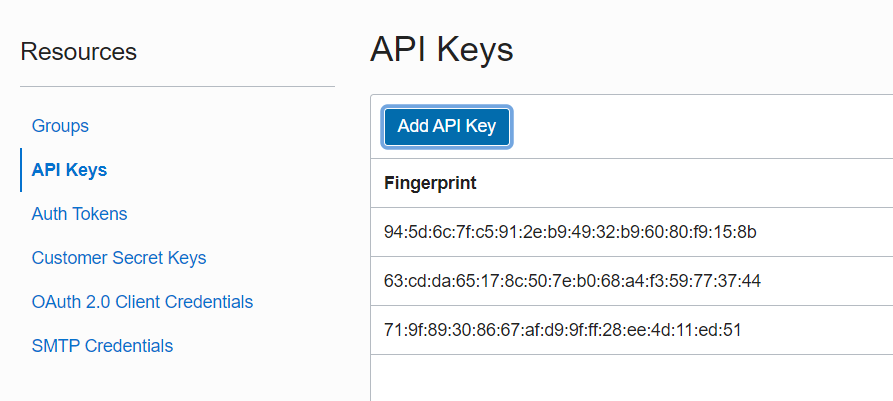


## Upload the Public Key

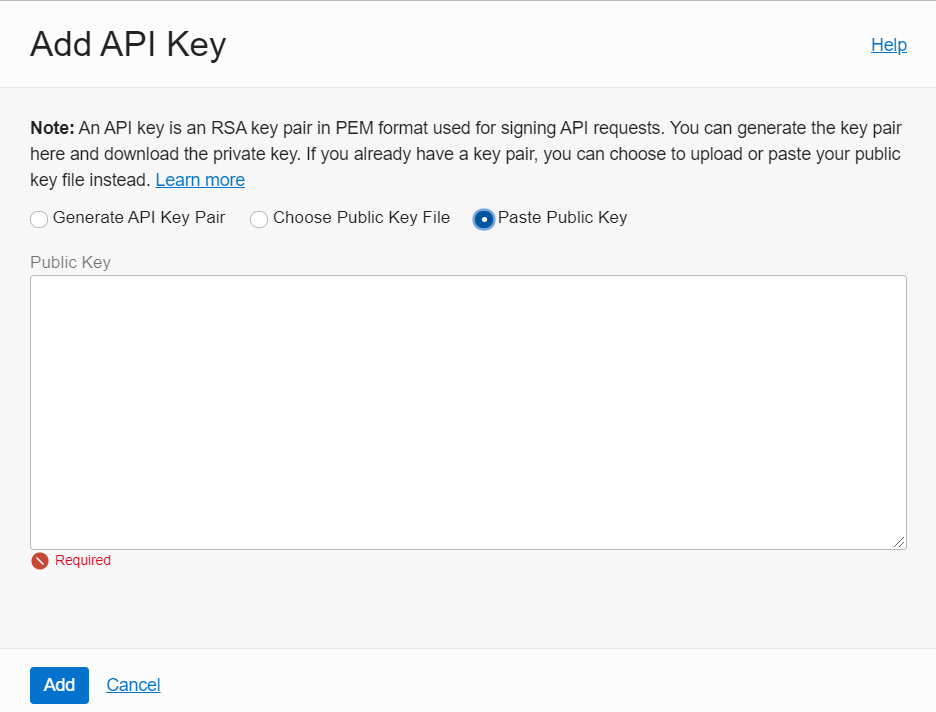
We now need to upload the public key created in the previous step to the Oracle Cloud Infrastructure. Log into Oracle Cloud Infrastructure and follow the steps detailed here. Once you upload the key, you will see your fingerprint displayed. You can check using the command referenced above. It is possible to create and upload a maximum of three keys.

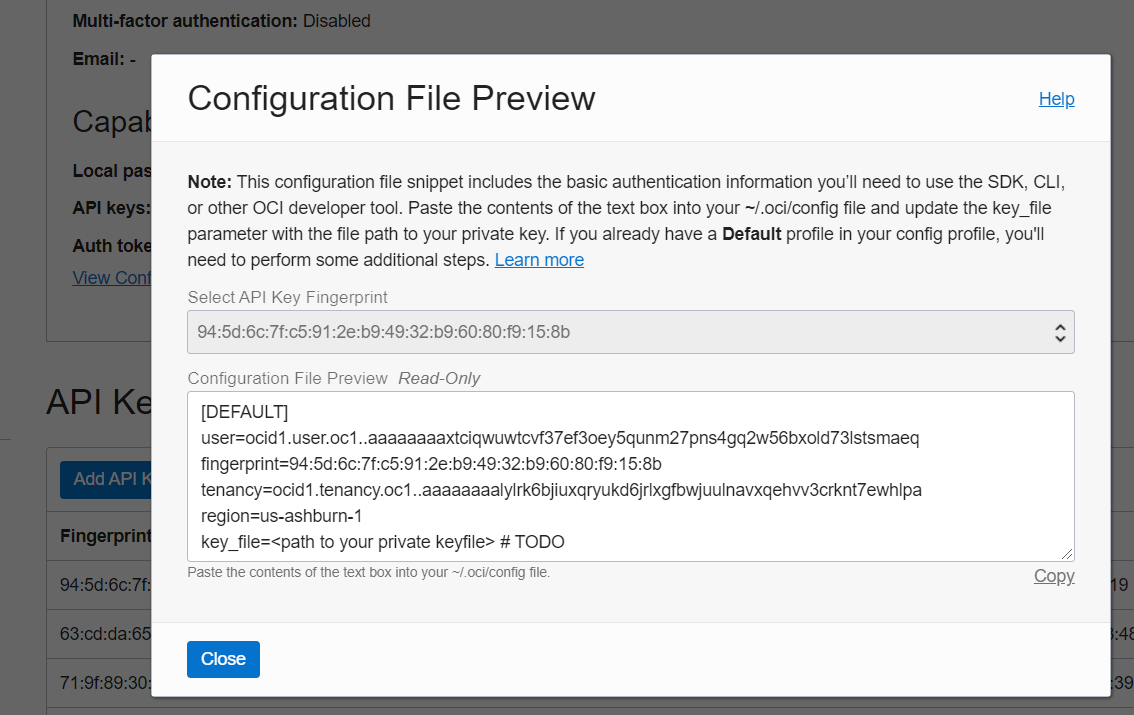
To upload the key go to the OCI console. Identity-> Users-> click on your user details

Under Resources select API key and click on add.



Add the API key and paste the content which you copied earlier.

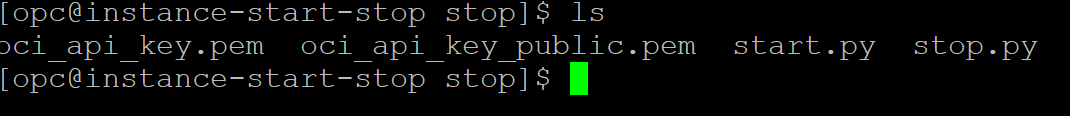


once you upload the key you will be able to see the fingerprint and configuration file which includes OCIDs.  


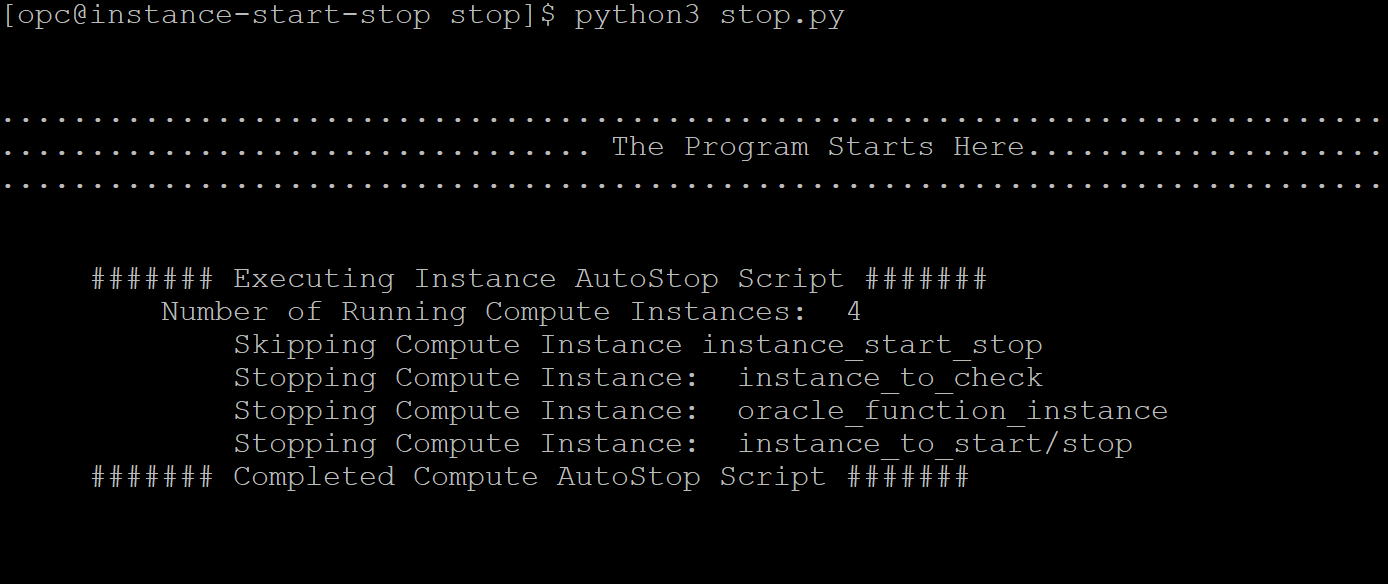
copy this to a notepad as you need to include this in your script later.

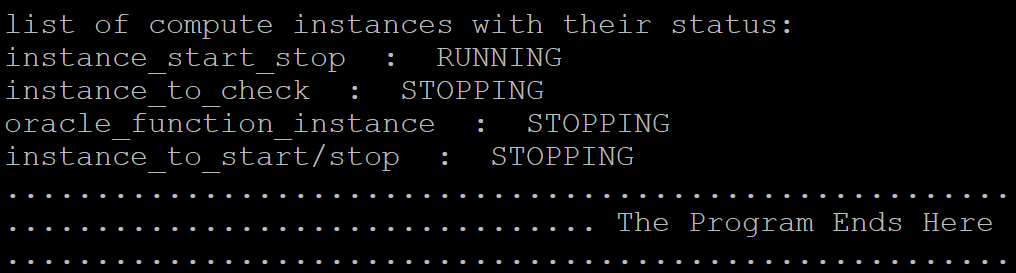
.

download the script and place it within the instance.

Here I have already placed it within a specific directory.  


Check if the script works  
run:   
python3 stop.py





You will be able to see the above output.  
similarly check if the start script runs fine.  
  
once the script is working, you can use the cron job to schedule it to execute automatically  
type the following commands:

crontab -e (opens it in the editor mode)

provide the time in UTC value for example see the below command.

30 06 \* \* \* cd /home/opc/stop && /usr/bin/python3.6 stop\_multiple.py >>test1.out  
for further information follow the below link:

<https://www.jcchouinard.com/python-automation-with-cron-on-mac/>

Hope now you are able to automatically manage starting and stopping of instances.