**OCI CLI Installation and VM Stop/Start Schedule Script**

**Hager El-Sehaimy Mohamed**

**November 2021**

**Disclaimer:** Below steps/details are written by Engineers that work in Oracle, it does not represent anything from Oracle.

**The Start**

1. Launch a Linux server for having OCI CLI installed in it and for this machine to be used as management box.

* Give it a name in the console for example: mgmt-linux
* Choose Ubuntu Machine with latest version and patch level ([Canonical-Ubuntu-20.04-Minimal-2021.06.14-0](https://console.eu-frankfurt-1.oraclecloud.com/compute/images/ocid1.image.oc1.eu-frankfurt-1.aaaaaaaakdtauwupkvi54552qmli3ozzj5zdwlhdfzcluphhyawzv7tqeu7q))
* Select 1 OCPU & 1 GB of memory (AMD E4 shape or any other shape)
* Place it in public subnet preferably or in private subnet in case you have VPN access
* Generate a public/private key pairs or use an existing one in order to be able to ssh to the machine using a private key

1. Once the machine is up, connect to it by ssh and username as **“Ubuntu”** and follow the below steps:

**$bash -c "$(curl -L https://raw.githubusercontent.com/oracle/oci-cli/master/scripts/install/install.sh)"**

**You will see below message:**

**===> In what directory would you like to place the install? (leave blank to use '/home/ubuntu/lib/oracle-cli'):**

**(Press Enter)**

**Then you will see below message:**

**===> In what directory would you like to place the 'oci' executable? (leave blank to use '/home/ubuntu/bin'):**

**(Press Enter)**

**Then you will see below message:**

**===> In what directory would you like to place the OCI scripts? (leave blank to use '/home/ubuntu/bin/oci-cli-scripts'):**

**(Press Enter)**

**Then you will see below message:**

**===> Currently supported optional packages are: ['db (will install cx\_Oracle)']**

**What optional CLI packages would you like to be installed (comma separated names; press enter if you don't need any optional packages)?:**

**(Press Enter)**

**Then you will see below message:**

**===> Modify profile to update your $PATH and enable shell/tab completion now? (Y/n):**

**(Press Y)**

**Then you will see below message:**

**===> Enter a path to an rc file to update (file will be created if it does not exist) (leave blank to use '/home/ubuntu/.bashrc'):**

**(Press Enter)**

**++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

**After that you will see below message as the OCI CLI has been installed successfully:**

**-- Backed up '/home/ubuntu/.bashrc' to '/home/ubuntu/.bashrc.backup'**

**-- Tab completion set up complete.**

**-- If tab completion is not activated, verify that '/home/ubuntu/.bashrc' is sourced by your shell.**

**--**

**-- \*\* Run `exec -l $SHELL` to restart your shell. \*\***

**--**

**-- Installation successful.**

**-- Run the CLI with /home/ubuntu/bin/oci --help**

**++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++**

1. Now we need to have an OCI user with right privileges to be able to execute API requests against your OCI tenancy. This user has to have an API key added in OCI console, please follow below steps to collect all information that will be required later on and to create api key pairs and add it to the user in OCI console. Note: We will assume that you will use the default admin OCI user **(non federated)** for simplicity however you can create a specific user to run specific tasks by using OCI CLI.

**While you are still connected to you Ubuntu server, execute below commands to generate API signing key for use with OCI CLI:**

**$mkdir ~/.oci**

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

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

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

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

**(Output of above command will look like below)**

**-----BEGIN PUBLIC KEY-----**

**MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAvw7Da+fmHCCz/8QZ4FRd**

**MRtdSP+yp4SpGB8afKZrQnoBcxEpK6I/QHsZ794X/glNA6htEvqISSSkB/dk+oZK**

**fmofAaFE/7SUXKuUQO8mh8Qpr5D6ywS03OTLZ0gOJu4+IiaCIW+D+wGmP9a6Mekf**

**OKsc6vCiZo1XJYZedAEsZv4XqBtpwIMM3c9UIdcNuOQEi6VdynMnvSg5+wn4US7+**

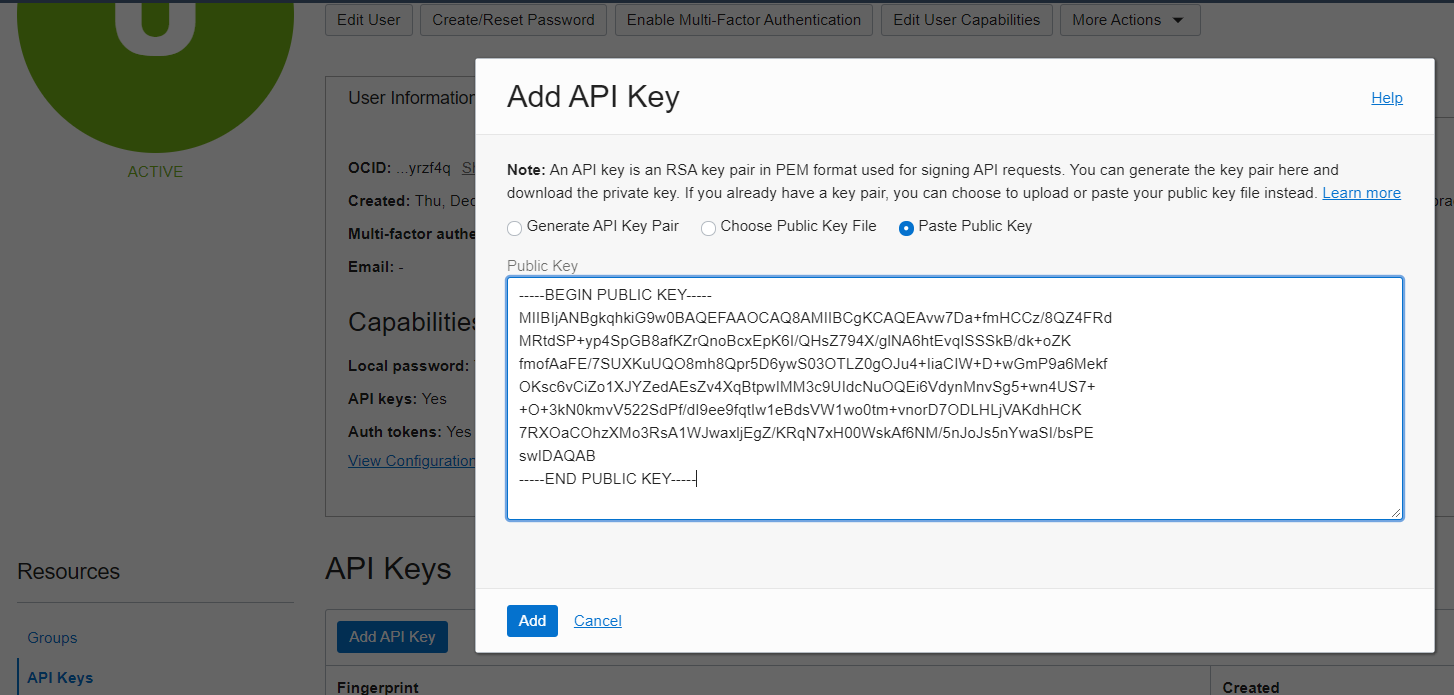
**+O+3kN0kmvV522SdPf/dI9ee9fqtIw1eBdsVW1wo0tm+vnorD7ODLHLjVAKdhHCK**

**7RXOaCOhzXMo3RsA1WJwaxljEgZ/KRqN7xH00WskAf6NM/5nJoJs5nYwaSI/bsPE**

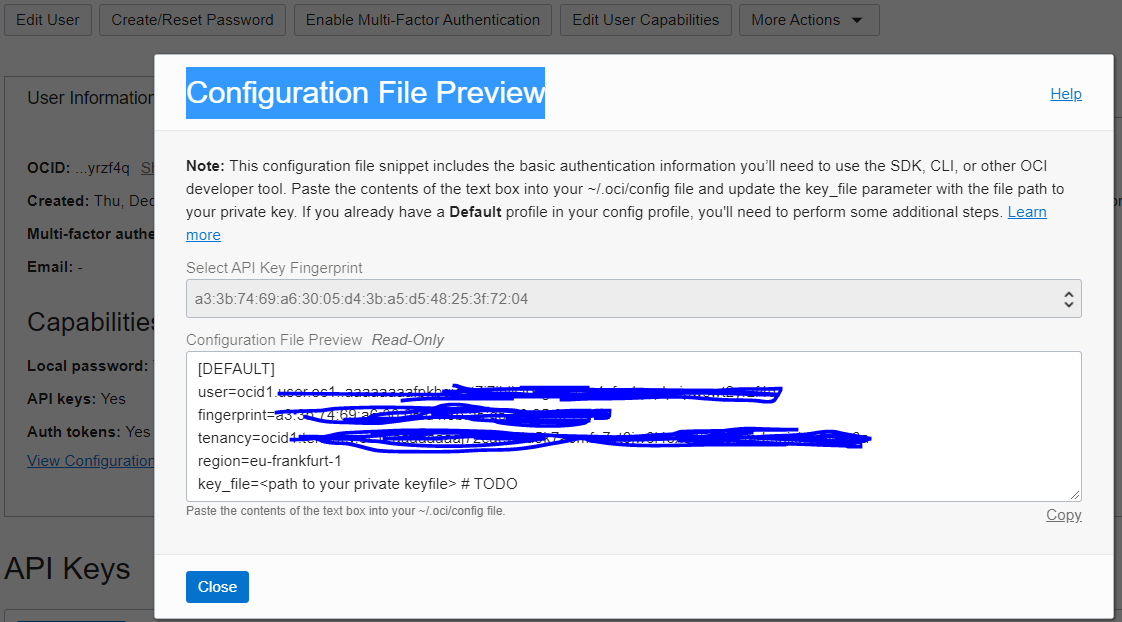
**swIDAQAB**

**-----END PUBLIC KEY-----**

**Copy above key and paste it in OCI console under you OCI user (non federated) as you see in below screenshot:**



* After you add the key, you see the configuration file preview as per below screenshot, please copy the content of OCIDs etc and paste it here.



[DEFAULT]

user=ocid1.user.oc1..aaaaxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

fingerprint=a3:3b:etc

tenancy=ocid1.tenancy.xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx

region=eu-frankfurt-1

key\_file=/home/ubuntu/.oci/oci\_api\_key.pem

**Execute below command for testing:**

**$oci os ns get**

1. Setting up config file for OCI CLI, run below commands: (ignore it if you did point 3 )

**$oci setup config**

**You will see below message:**

**===> Enter a location for your config [/home/ubuntu/.oci/config]:**

**(Press Enter)**

**Then you will see below message:**

**===> Enter User OCID**

**(Copy Value from above)**

**Then you will see below message:**

**===> Enter a tenancy OCID**

**(Copy Value from above)**

**Then you will see below message:**

**===> Enter a region by index or name**

**(Copy Value from above)**

**Then you will see below message:**

**===>Do you want to generate a new API Signing RSA key pair? (If you decline you will be asked to supply the path to an existing key.) [Y/n]:**

**(Press n then enter)**

**Then you will see below message:**

**===>Enter the location of your API Signing private key file:**

**(Copy value from above)**

Ex:

/home/ubuntu/.oci/oci\_api\_key.pem

**Execute below command for testing:**

**$oci os ns get**

Power-On and Power-Off Script.

**-Install pip by running below command:**

**$sudo apt-get install python3-pip**

**-Install OCI Python SDK by running below command:**

**$pip3 install oci**

**-Create two scripts, one for power-on and one for power-ff, first let us create a directory for the script then create two files script:**

**-While you are inside ubuntu user home, run below commands:**

**$mkdir scripts**

**$cd scripts**

**Create a txt file that contains OCID you need to manage by the scripts and paste each OCID in separate line as below**

**$ vi OCID.txt**

OCID1  
OCID2  
OCID3

**Create power-off script as below**

**$vi power-off**

**-Insert below details:**

import oci  
import re  
  
config = oci.config.from\_file("file path") #leave blank if it’s default config file path. Blank example is below in the power-on script  
  
OCIDs\_file = open('OCID.txt', 'r')  
OCIDs = OCIDs\_file.readlines()  
  
  
for OCID in OCIDs:  
 OCID = OCID.strip()  
 if re.search("dbnode", OCID):  
 database\_client = oci.database.DatabaseClient(config)  
  
 # Send the request to service, some parameters are not required, see API  
 # doc for more info  
 db\_node\_action\_response = database\_client.db\_node\_action(db\_node\_id=OCID, action="STOP")  
 elif re.search("instance", OCID):  
 core\_client = oci.core.ComputeClient(config)  
 instance\_action\_response = core\_client.instance\_action(instance\_id=OCID, action="SOFTSTOP")  
  
# Send the request to service, some parameters are not required, see API  
# doc for more info  
  
# os.system("oci db node start ocid1.dbnode.oc1.me-jeddah-1.anvgkljrqvob4kyaqtp6ivivuc2ocbmnpeydfzqisls5qvdbnq4ix7qy7jgq")

**-Exit the editor and save the file.**

**-Create the other script file**

**$vi power-on**

**-instert below code in the file:**

import oci  
import re  
  
config = oci.config.from\_file()  
  
OCIDs\_file = open('OCID.txt', 'r')  
OCIDs = OCIDs\_file.readlines()  
  
  
for OCID in OCIDs:  
 OCID = OCID.strip()  
 if re.search("dbnode", OCID):  
 database\_client = oci.database.DatabaseClient(config)  
  
 # Send the request to service, some parameters are not required, see API  
 # doc for more info  
 db\_node\_action\_response = database\_client.db\_node\_action(db\_node\_id=OCID, action="START")  
 elif re.search("instance", OCID):  
 core\_client = oci.core.ComputeClient(config)  
 instance\_action\_response = core\_client.instance\_action(instance\_id=OCID, action="START")  
  
# Send the request to service, some parameters are not required, see API  
# doc for more info  
  
# os.system("oci db node start ocid1.dbnode.oc1.me-jeddah-1.anvgkljrqvob4kyaqtp6ivivuc2ocbmnpeydfzqisls5qvdbnq4ix7qy7jgq")

**-Save the changes and exit.**

1. Finally, install cron and add the python scripts commands to your cron tab according to your timing schedule needs:

**$sudo apt install cron**

**$sudo systemctl enable cron**

**$sudo apt-get vim**

**$crontab –e**

* Refer below link for more details about cron:

<https://www.cyberciti.biz/faq/how-do-i-add-jobs-to-cron-under-linux-or-unix-oses/>