Quick installation guide for Watchdog VM with Python Script

Date: 07/06/2019
Authors:
Stephan Alluchon
Pierre Lepaul

Here are all the steps to install a new VM on OCI with python scripts.

1/ Create a VM instance on MGMT SUBNET (same Subnet). You can use a shape 2.1 with Oracle Linux 7.x

2/ Install Python 3.6 in a virtual environment

Follow: https://yum.oracle.com/oracle-linux-python.html
So you will have to add those commands:

```
sudo yum install -y oracle-epel-release-el7 oracle-release-el7
sudo yum install -y oracle-epel-release-el7
sudo yum install -y python36
python3.6 -m venv py36env
```

Install screen utility in order to have virtual screen in Linux

```
sudo yum install screen
```

3/ Go to new Virtual environment

```
source py36env/bin/activate
```

4/ Upgrade pip

```
(py36env) [opc@watchdog ~]$ pip install -U pip
```

5/Install OCI python SDK and extra Python module

```
(py36env) [opc@watchdog ~]$ pip install oci
(py36env) [opc@watchdog ~]$ pip install flask
(py36env) [opc@watchdog ~]$ pip install waitress
```

pip install httpsig_cffi requests six

6/ create .oci drirectory

```
(py36env) [opc@watchdog ~]$ mkdir ~/.oci
```

7/ create OCI config file and edit it with an 'UCM' profile. UCM profile is here an example that should be used in the python script:

```
(py36env) [opc@watchdog ~]$ nano ~/.oci/config
```

```
[UCM]
key_file=/home/opc/privateKey
```

```
user=ocid1.user.oc1..aaaaaaaafzzv6w52dc2pkdkpp2ixbr
a
fingerprint=e1:4f:7f:e7:b5: 92:bb:ae:3d
tenancy=ocid1.tenancy.oc1..aaaaaumwyvjedslpsdb2d2xe2kp2q
region=eu-frankfurt-1
```

8/ Install private KEY used for API user in /home/opc/privateKey

`You can use nano` /home/opc/privateKey and copy your key

9/ create /home/opc/oci_value_IP_address.json file which contains IP address of PAN (PaloAlto Network instances) and IP address of considered VNICs

10/ copy from GITHUB file test_json_ip_address_SDK.py in directory /home/opc
Don't forget to update the line of the script with yourprofile file name
config = from_file(profile_name="**yourprofile**")

Copy from GITHUB file updatePrivateIPfromIPwithSDK_waitress.py
in directory /home/opc
Don't forget to update the line of the script with yourprofile file name
config = from_file(profile_name="yourprofile")

11/ test your JSON file and OCI config file using test_json_ip_address_SDK.py

`(py36env) [opc@watchdog ~]$ python ./` test_json_ip_address_SDK.py

You should have status code 200 for each IP Address. Example:

```
Requesting:10.103.0.50
>status code: 200
>ocid1.vnic.oc1.eu-frankfurt-
1.abtheljs3jaznkckifs4iikhmomflwncphvplaxl5wulf65nppziho5uo23a
>ocid1.privateip.oc1.eu-frankfurt-
1.aaaaaaaanukpmuwtjrxacxaxxgpycj65nkxcrbvm2kf5gag6t65z4cgbx64q
finish!
```

12/ Launch python script in a new screen

Type in your linux shell:
`(py36env) [opc@watchdog ~]$ screen`

you will have a new screen session

Launch python script
`(py36env) [opc@watchdog ~]$ python ./updatePrivateIPfromIPwithSDK_waitress.py`

If all is ok you should have the following message:

```
No error on getting all OCID!
Starting Flask server
Serving on http://0.0.0.0:5000
```

You can detach remote screen: CTRL A + D

13/ test your script with a simple curl command

Check on OCI configuration if secondary IP are on PAN1 or PAN2

Type in your shell if all secondary on PAN2:
(py36env) [opc@watchdog ~]$ curl http://127.0.0.1:5000/PrimaryIsVM1

As a result you should verify that:

1/ On OCI, all the secondary are on the PAN1 VNICs

2/ you should see on your linux session, the following HTTP answer:

```
            <html>
            <body>
                <h1>
                    DATE: Fri, 07 Jun 2019 10:30:44 GMT <br>
                    <blockquote>HERE ARE ALL THE MODIFICATIONS:
 <br> 10.0.20.50 is now associated with 10.0.20.10
 <br> 10.0.30.50 is now associated with 10.0.30.10
 <br> 10.103.0.50 is now associated with 10.103.0.10
 <br> </blockquote> <br>
                    Private IP updated!
                </h1>
            </body>
            </html>
```

14/ You can come back to your detached screen:

```
(py36env) [opc@watchdog ~]$ screen -r
You will also the same kind of logs.
```