ORACLE

Scaling Up and Down OAC Instance Automatically on a Linux VM based on a schedule

20.02.2024, Version 1

Scaling Up and Down OAC Instance Automatically on a Linux VM based on a schedule

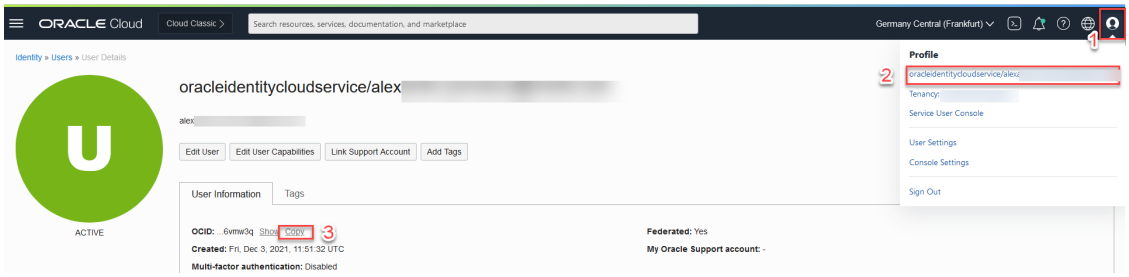
1. Install CLI on the machine. Proceed with default installation.

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

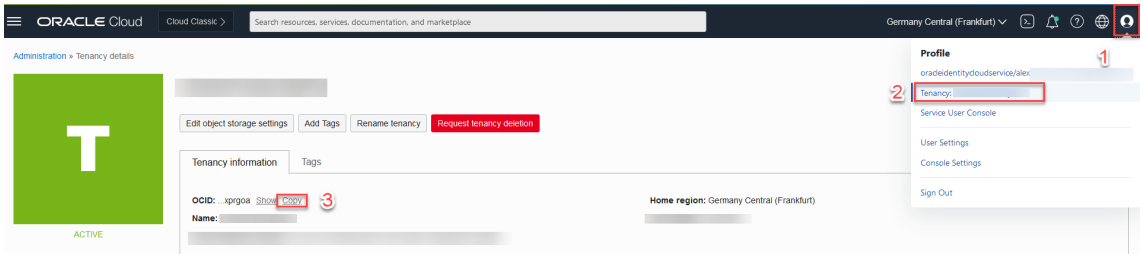
```
[opc@oci ~]$ bash -c "$(curl -L https://raw.githubusercontent.com/oracle/oci-cli/master/scripts/install/install.sh)"
```

2. For the next step, you will need **User OCID** and **Tenancy OCID**. For getting them, login the Oracle Cloud.

   2.1 For User OCID => On the Right side of the screen, click on the user logo and select the user.
   => Next to OCID click on Copy and paste it into a notepad.



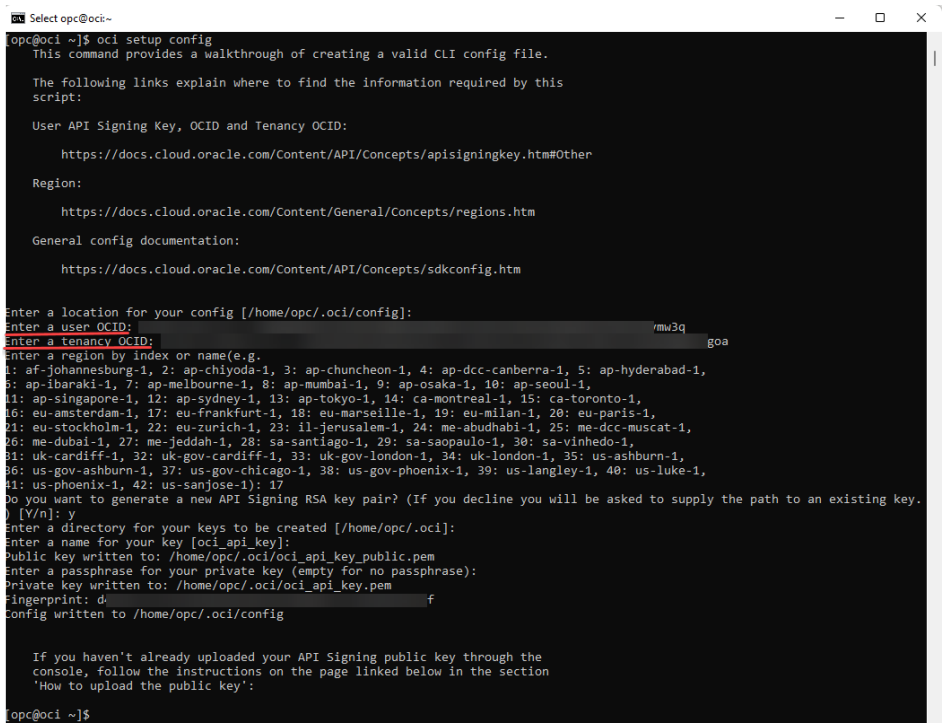
   2.2 For Tenancy OCID => On the Right side of the screen, click on the user logo and select the Tenancy => Next to OCID click on Copy and paste it into a notepad.



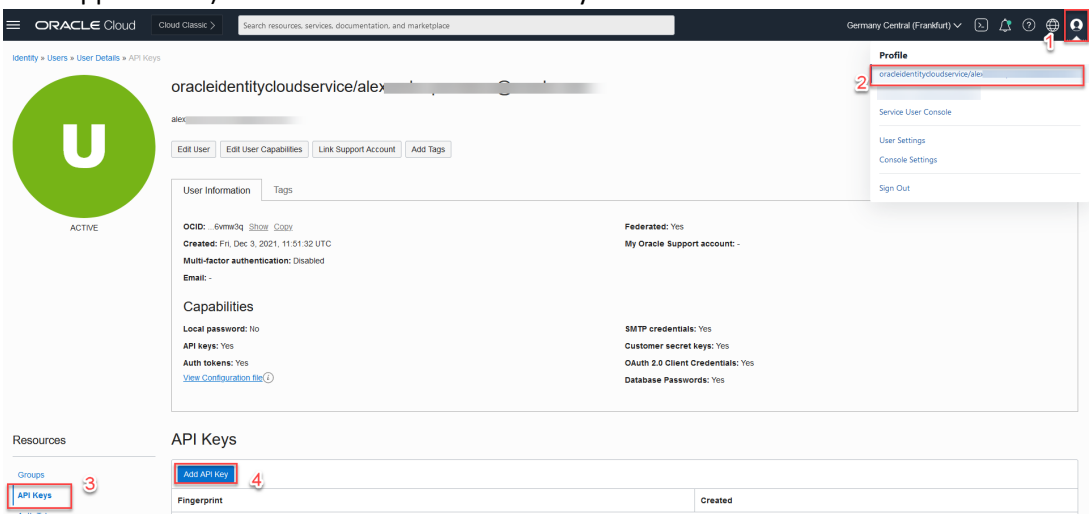
   2.3 The end result should look like the picture below, for faster action on the next point.

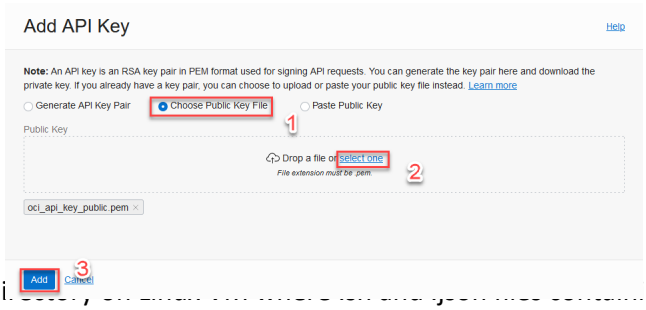
```
1  UserOCID:                                                    rmw3q
2  TenancyOCID:                                                 goa
```

3. After cli installation, you will have to proceed with **setup config**. You will need **User OCID** and **Tenancy OCID**
   oci setup config



4. Once done, you have to download the generated public key (using scp command or any other SFTP applications) and add it to the Users API keys.





5. Create a new di[illegible]ng the script will be stored. I named it “script”

```
[opc@oci ~]$ mkdir /home/opc/script
```

6. Access that directory.

```
[opc@oci ~]$ cd /home/opc/script
[opc@oci script]$
```

7. Enter command from below to create a json file called 2_ocpu.json

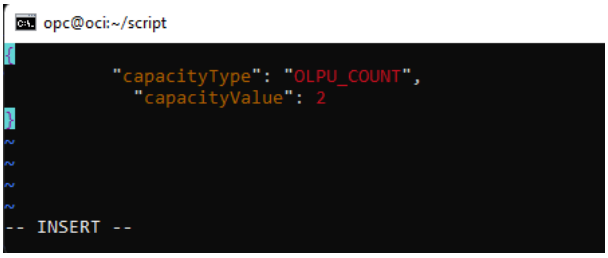
   vi 2_ocpu.json

```
[opc@oci script]$ vi 2_ocpu.json
```

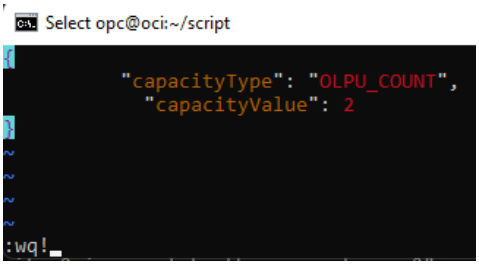
8. Press i to start typing and insert the code from below.

```
{
  "capacityType": "OLPU_COUNT",
  "capacityValue": 2
}
```

"capacityType": "OLPU_COUNT" – means that you would like to make scaling based on OCPU
***(use “CapacityType”: “USER_COUNT” – to scale based on users)
"capacityValue": 2 – means that you would like to make a scale to 2 OCPUs

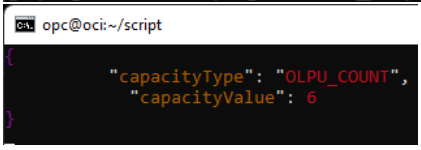


Once pasted, press ESC button and type “:wq!” to write the changes and quit.



9. Repeat step 7 and 8 to create a new file 6_ocpu.json that will increase the OCPUs up to 6.

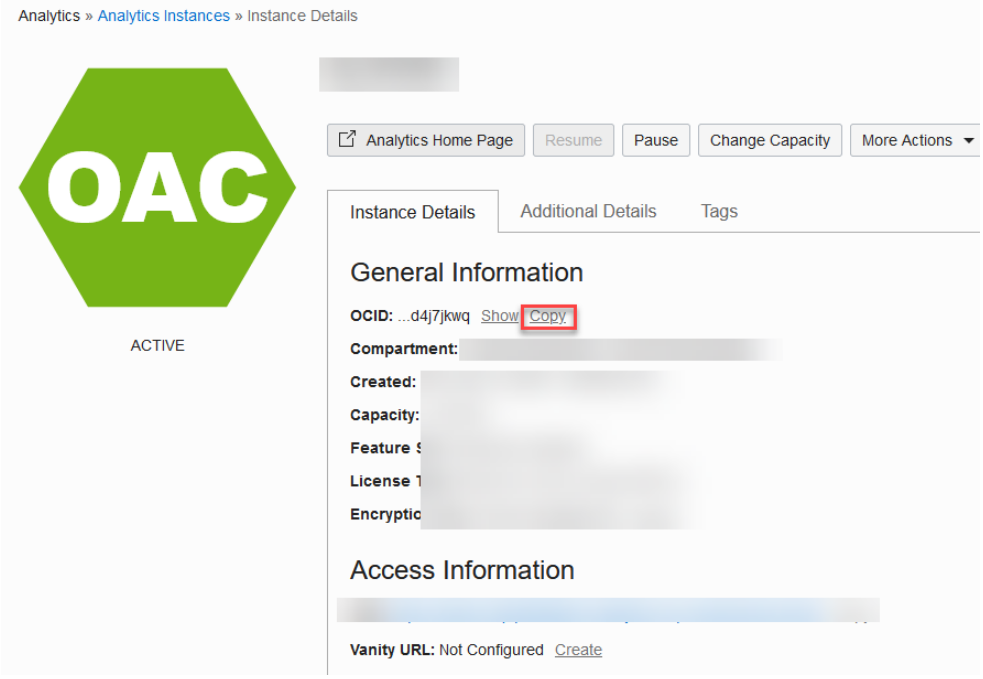
```
[opc@oci script]$ vi 6_ocpu.json
```




```
[opc@oci script]$ ls
2_ocpu.json  6_ocpu.json
```

10. For the next step you will need the OCID of your OAC instance.

    Navigate to Oracle Cloud => Burger Menu => Analytics & AI => Analytics Cloud
    Select the instance that you would like to scale.
    Click on Copy next to OCID.



11. Enter the code from below to identify the full path of the oci that was installed at step 1.

    which oci

```
[opc@oci script]$ which oci
~/bin/oci
```

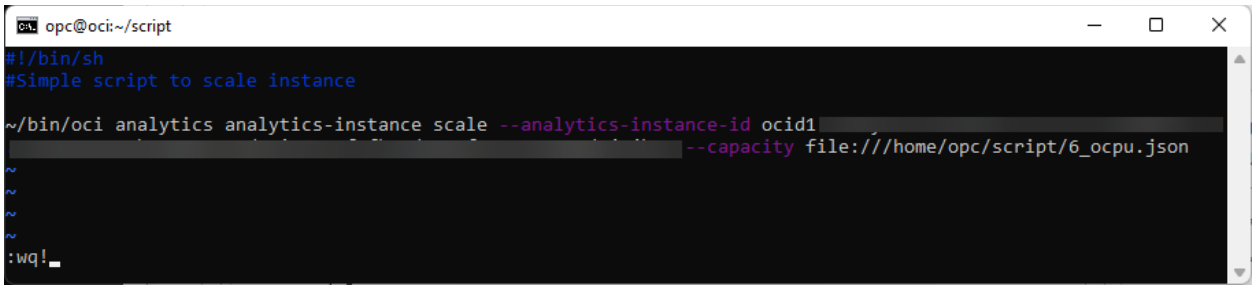
12. In the same directory, create a new file called **scale_up.sh** and enter the code from below. (be ready to paste OAC OCID and the oci path)

    #!/bin/sh
    #Simple script to scale instance

    ~/bin/oci analytics analytics-instance scale --analytics-instance-id ocid1.analyticsinstance.oc1.eu-frankfurt-1.aaaaaaaaq76hxp6mmaz3md45i44v32fsfkcpdqaogftwwr2aa4pgd4j7jkwq --capacity file:///home/opc/script/6_ocpu.json

    The code is:
    oci analytics analytics-instance scale --analytics-instance-id <instance OCID> --capacity file:///<json file path>

    ~/bin/oci – is the path where oci was installed. When cron opens a shell, it will not inherit the environment variables (like PATH) that you use from the command prompt. You need to include the complete, explicit path to the oci command in your script. You can find this from your command prompt with the which command



13. Create one more file called **scale_dowh.sh** and indicate the file path for 2_ocpu.json
14. Let's make files: **scale_up.sh** and **scale_down.sh** executable.

    chmod +x scale_down.sh
    chmod +x scale_up.sh

```
[opc@oci script]$ chmod +x scale_down.sh
[opc@oci script]$ chmod +x scale_up.sh
```

15. As a result, you will have 2 .sh files and 2 .json files in the script directory. Type a quick ls -la command to view the files and access.

    ls -la

```
[opc@oci script]$ ls -la
total 16
drwxrwxr-x. 2 opc opc  84 Jun 15 10:50 .
drwx------. 8 opc opc 174 Jun 15 10:43 ..
-rw-rw-r--. 1 opc opc  61 Jun 15 10:27 2_ocpu.json
-rw-rw-r--. 1 opc opc  62 Jun 15 10:30 6_ocpu.json
-rwxrwxr-x. 1 opc opc 263 Jun 15 10:43 scale_down.sh
-rwxrwxr-x. 1 opc opc 263 Jun 15 10:42 scale_up.sh
[opc@oci script]$
```

16. As final step, let's use crontab command to make a scheduler. Enter the command below to open the cron editor.

    crontab -e

```
[opc@oci script]$ crontab -e
```

17. The condition is that from Mon to Fri, from 9am to 5 pm to have 6 OCPUs and the rest to be 2 OCPUs. (use i to start typing and :wq! to save and exit)

    *** NOTE *** - it will run according the VMs clock. Check first with # date command the VMs timezone.

    0 9 * * 1-5 sh /home/opc/script/scale_up.sh >> /home/opc/script/scale_log.log 2>&1
    0 17 * * * sh /home/opc/script/scale_down.sh >> /home/opc/script/scale_log.log 2>&1

    Example: 0 9 * * 1-5:
    0 - represents minute

9 - represents hour
First * - represents day
Second * - represents month
1-5 - represent weekday
At 09:00 on every day-of-week from Monday through Friday will run the script scale_up.sh to scale up to 6 ocpus and will scale down everyday at 5pm (17:00).

18. Check the file scale_log.log to make sure that everything works fine.

    cat /home/opc/script/scale_log.log

```
[opc@oci /]$ cat /home/opc/script/scale_log.log
{
  "etag": "1925202593a9a73e8761b2a242502c82b1cbd5697f4d4d5edbab168fbc508519",
  "opc-work-request-id": "ocid1.coreservicesworkrequest.oc1.eu-frankfurt-1.aaaaaaaazmbic7elqdrg5ma35usbcvgyqgk3i
2bed4a6awzlakygaw3bphoa"
}
```