Instructions to schedule turn On/Off an OCI instance on Windows

Introduction

In this step by step guide, we will have a look how we can leverage on OCI CLI to automate the turning On/Off of OCI instances, as well as the scaling of said instance. For this guide, we will be using Autonomous Data Warehouse as an example. Any other instances will work the same way. For the scheduling, we will be running it on a Windows machine.

Why want to turn On/Off an OCI instance?

The reason of having to automatically turn On/Off instances is because of the reason that you would want to save cost, as well as resources. Instances can be turned off when it is not in use in order to manage your OCI cost and be turned on again when it is needed.

With automation, users of the instances would not need to keep reaching out to OCI Admins to turn on or off the instances. Admins will also decrease their workload by setting up the automation.

Prerequisites:

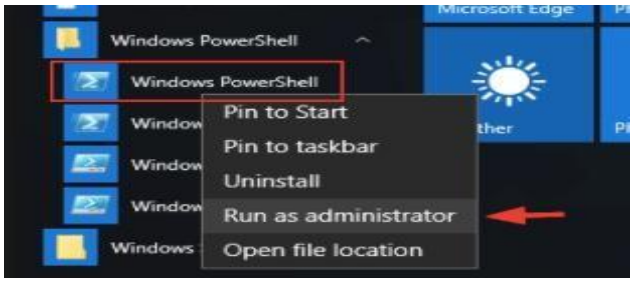
1. A Windows instance, either locally or on cloud
2. An OCI instance
3. Download the scripts provided. (There will be 2 scripts)

In this guide, you will learn how setup OCI CLI, how to use the tagging feature on OCI, as well as how to set up a Task Scheduler on Windows

Install OCI CLI Software on Windows

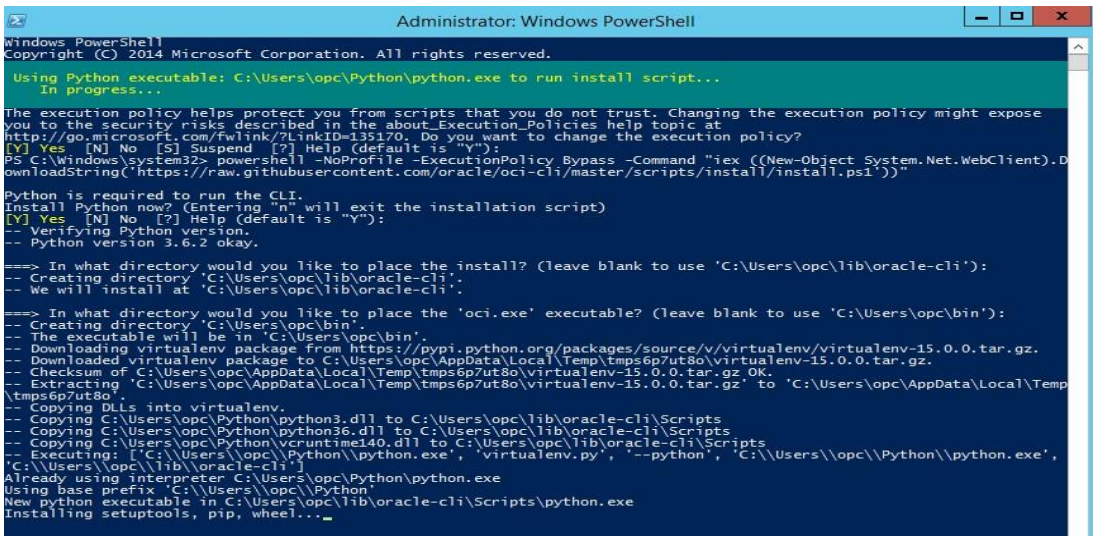
CLI is the utility to upload files into OCI object storage. This section documents the steps to install the CLI software. Following steps are the step by step method to install CLI on a Windows machine.

1. To run the install OCI CLI on Windows, you will need to access PowerShell. To access PowerShell all you need to do is follow these steps:
   Start menu -> typing “powershell” -> right click PowerShell to run it as Administrator.



2. Enable the RemoteSigned execution policy in PowerShell, and press “Y” to apply the change

3. Run the installer script in PowerShell, and Press “Y” to install Python when prompted.

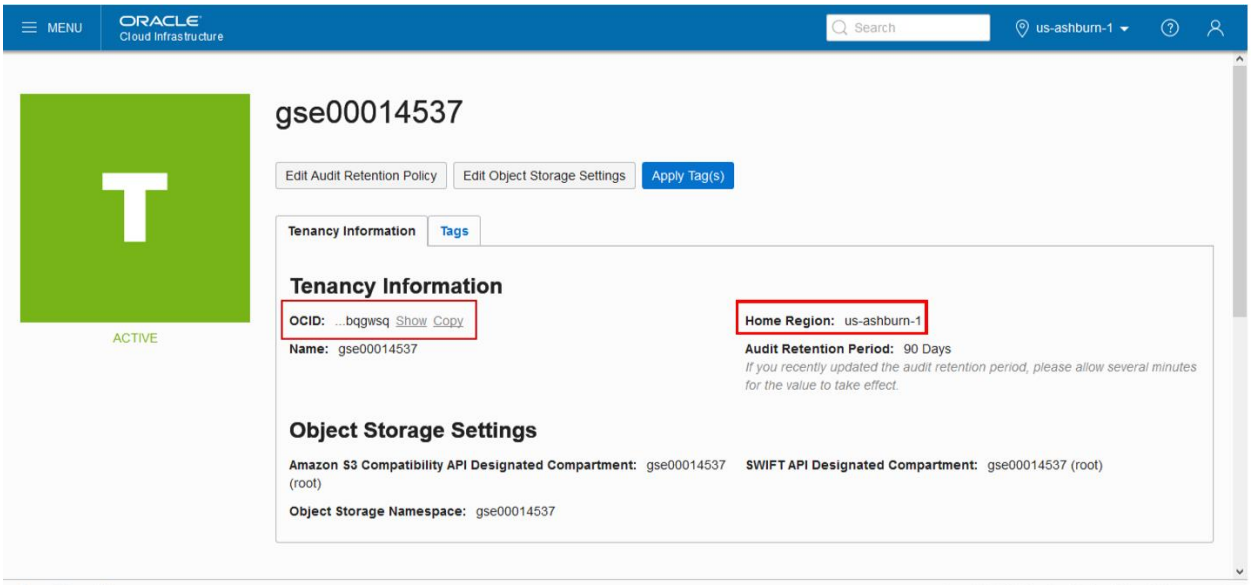


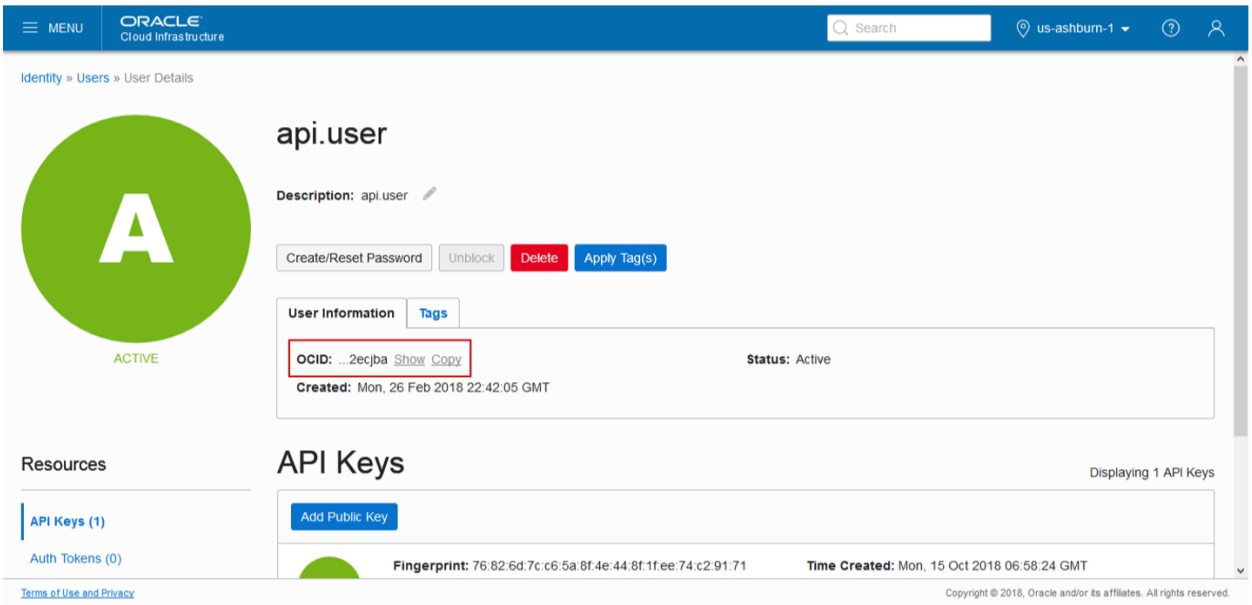
OCI-CLI Configuration

CLI OCI setup will generate the oci_api_key.pem and oci_api_key_public.pem based on the OCID information collected from the Cloud Portal page.

1. Collect information from OCI User & Tenancy

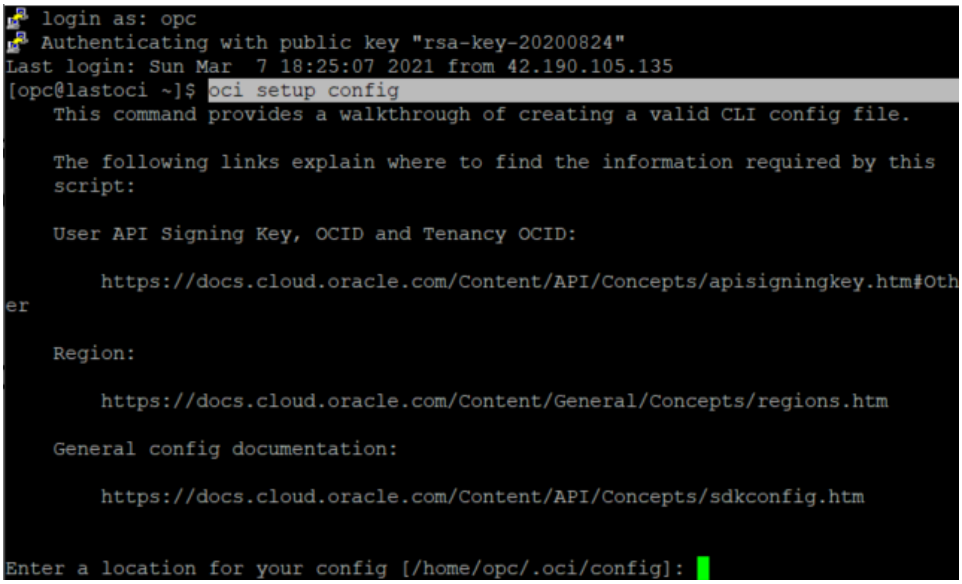
First you will be required to get the User OCID and Tenancy OCID as well as Tenancy Region from Oracle Cloud Infrastructure.





2. Run the setup tools to get the CLI ready to access OCI.

First step in on your PowerShell as Administrator type in the command oci setup config to run the setup process.



To note some fields needed to be entered when running the script:

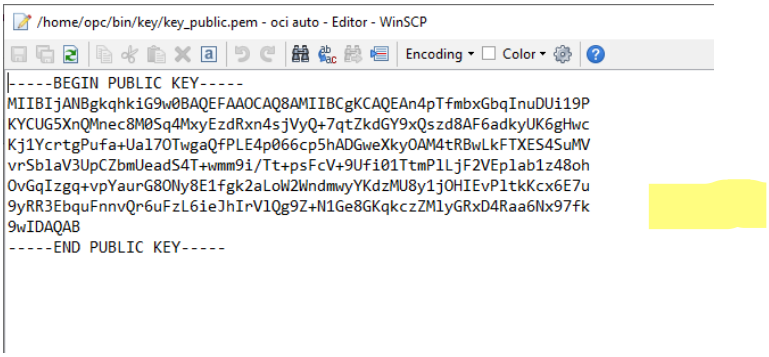
- Enter a location for your config [/home/opc/.oci/config]? (leave blank to use /home/opc/.oci/config)
- Enter a user OCID (Use your User OCID)
- Enter a tenancy OCID? (Use your tenancy OCID)
- Enter a Region (Use the region: ap-sydney-1) < (The region will depend on where your OCI tenancy is located.)
- Do you want to generate a new RSA key pair (Type Y to create a new key pair that will be stored in the machine. This key will be used to add a fingerprint for your user to access OCI Instances)
- Enter a directory for your keys to be created: (Type in any name you want for the key folder)
- Enter a name for your key: (Type in any name you want your keys to be.)
- Enter a passphrase for your private key (Here you can define a password for your private key. If not required, just keep it blank and hit enter.)

3. Add the public key to the User in OCI

Here we will be adding the created public key on our previous step to our User in OCI, allowing the script to be run through the user.

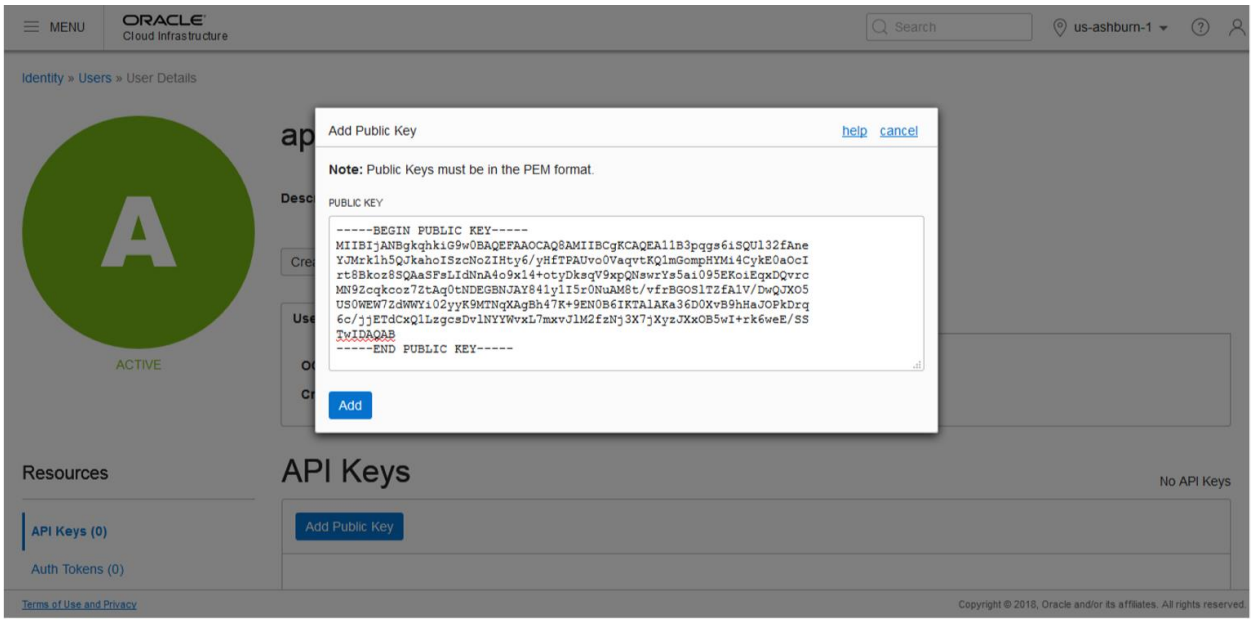
3.1 First open the location where you have your created public key.

Open the folder in which you have stored your key. Next double click on the public.pem file to open it in an editor. Copy the text within the file.



3.2 Copy the public key and add it to your OCI user

Follow this step in OCI to add the public key. MENU -> Identity -> Users -> User -> API Keys -> Add Public Key.



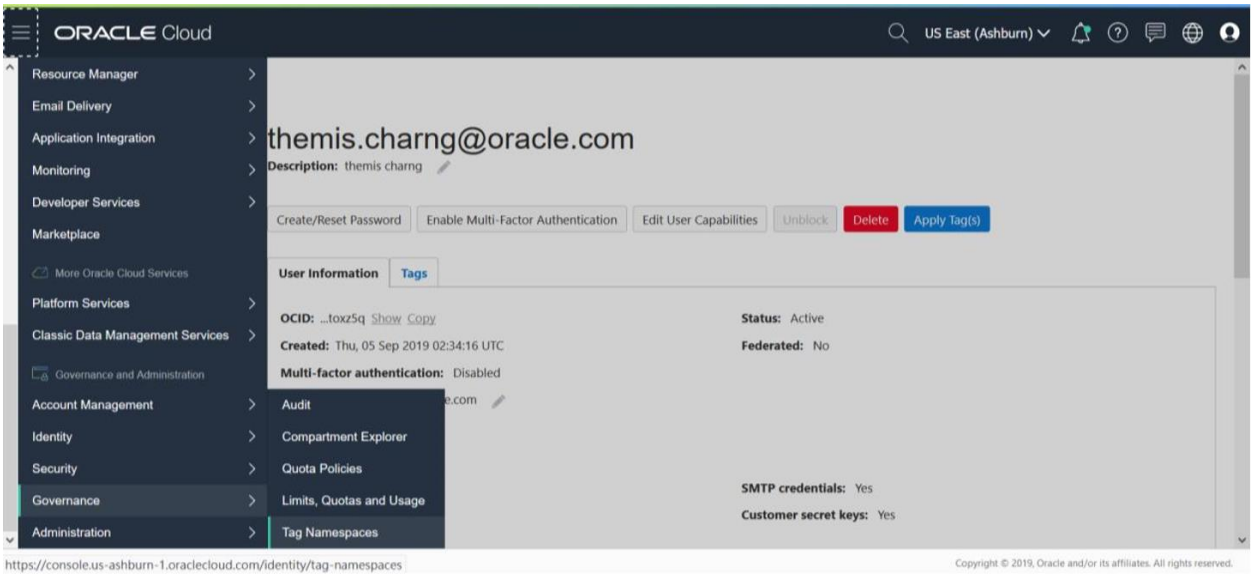
Click on Add to complete the step.

ADW and OAC configuration

To control what to power on/off, you need to create a predefined tag called Schedule. If you want to localize this, that is possible in the script. For the predefined tag, you need entries for the days of the week, weekdays, weekends and anyday. The tags names are case sensitive.

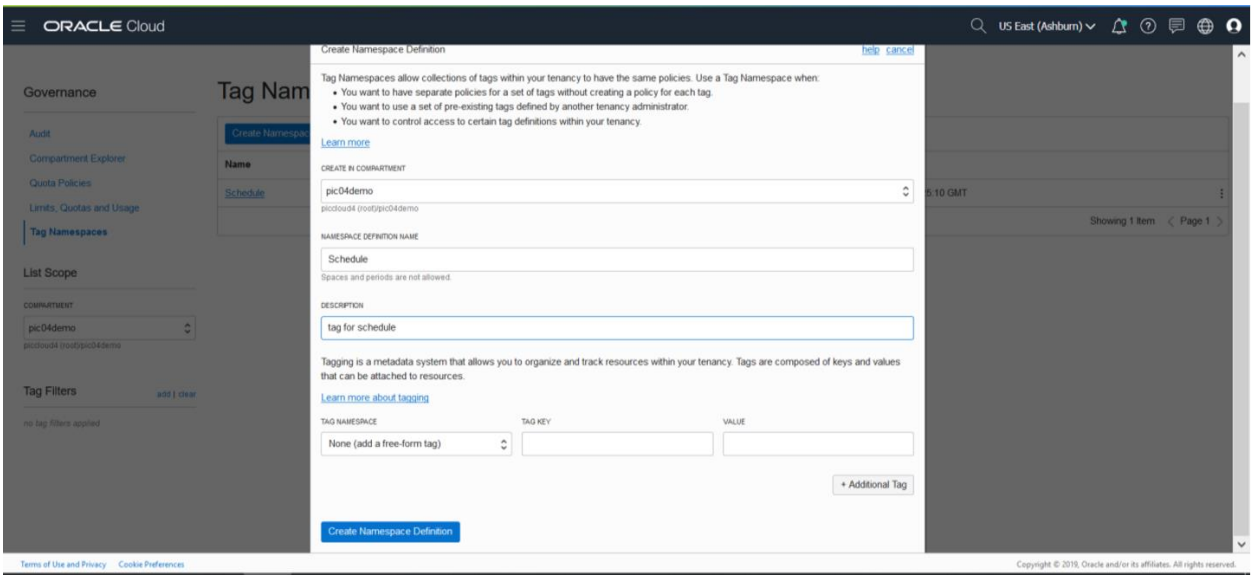
1. Create the Tag Namespace

This section guides you on how to create a tag, so it can be added and used on multiple instances.



Create Namespace Definition -> Choose Compartment -> Fill in Namespace Definition Name -> Fill in Description

- Namespace Definition Name: "Schedule"
- Description: "(File in description you want to specify)"

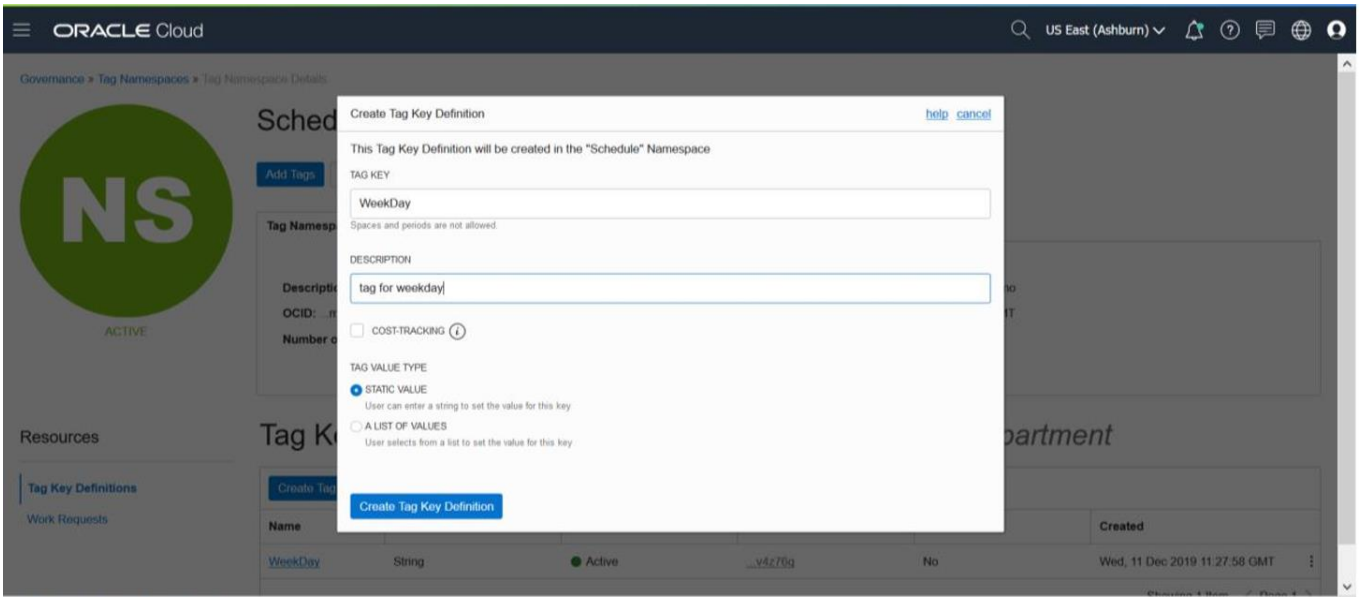


Create Tag Key Definition -> Fill in TAG KEY -> DESCRIPTION -> Choose TAG VALUE TYPE

▪ TAG KEY could be set as “WeekDay”, “Weekend”, “AnyDay” or a specific day e.g. Monday, Tuesday,…. A Weekend/Weekday tag overrules an AnyDay tag. A specific day of the week tag (ie. Monday) overrules all other tags in the scaling python script.

▪ DESCRIPTION should be filled in with words you want to specify.

▪ TAG VALUE TYPE should be chosen as STATIC VALUE for further configuration.

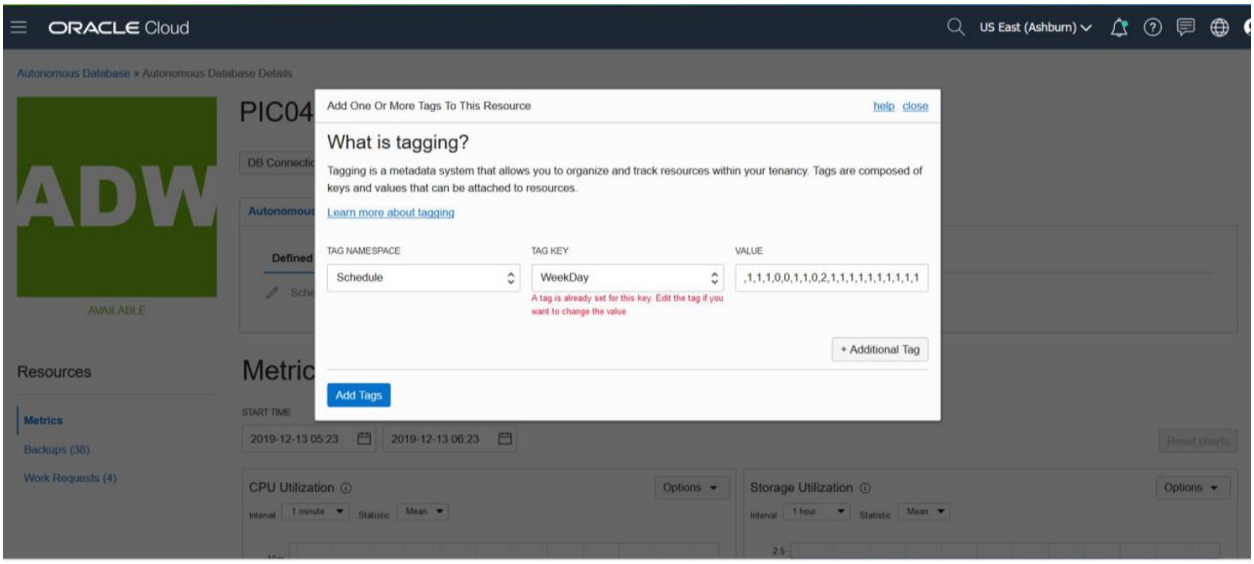


2. Add tag to ADW instances (Same method as for OAC)

A single resource can contain multiple tags. A Weekend/Weekday tag overrules an AnyDay tag. A specific day of the week tag (ie. Monday) overrules all other tags.

The value of the tag needs to contain 24 numbers (else it is ignored), seperated by commas. If the value is 0 it will power off the resource (if that is supported for that resource). Any number higher than 0 will re-scale the resource to that number. If the resource is powered off, it first will power-on the resource and then scale to the correct size.

Keep in mind the 24 numbers follows the 24 hour format, where the first number is 0000, second is 0100, and ends with 2300.



Prepare and Run the Script on Windows

Run python script to power on/ off and scale up/ down instances.

1. Setup environment for python script

In your Windows command prompt, run this command curl https://bootstrap.pypa.io/get-pip.py -o get-pip.py to download pip to within your Windows before we can run our scripts.

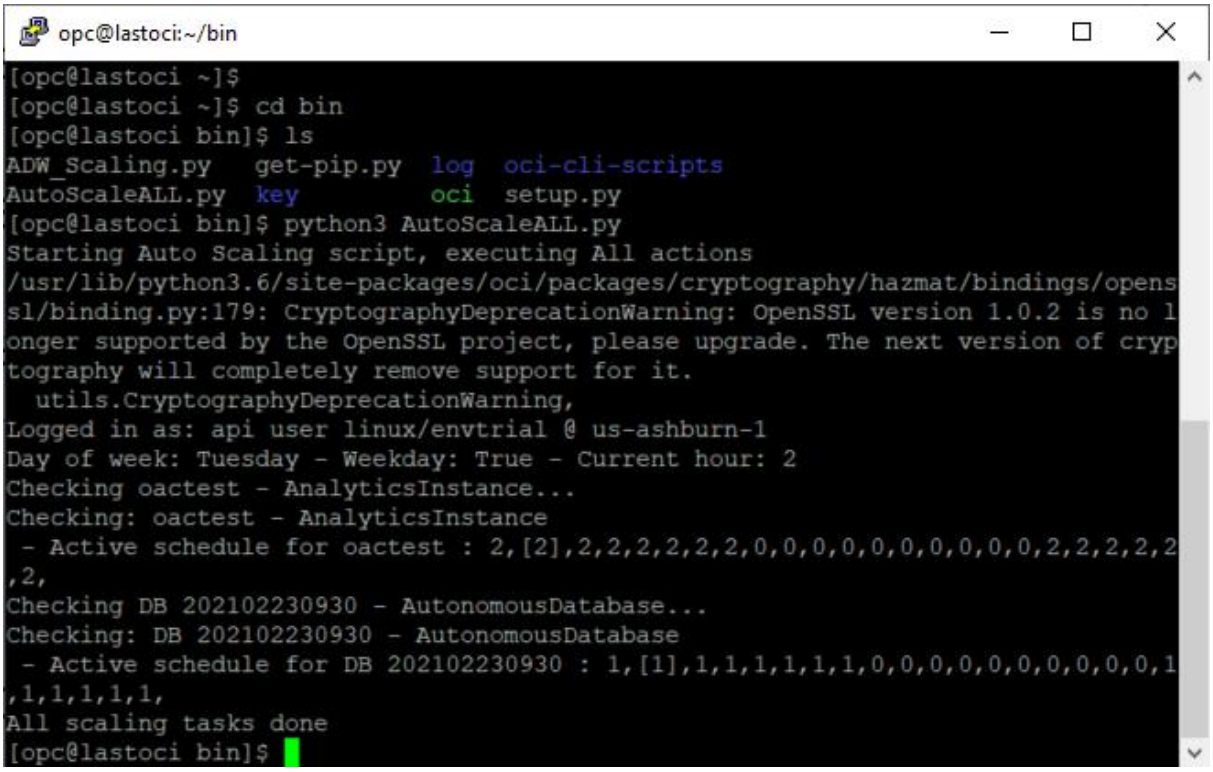
Next, we need to setup the pip. To do that, we just need to run the command python3 get-pip.py to start the setup.

Last step on the setup is to run the script setup.py in which you have added into your Linux machine's bin folder, through WinSCP. To run the setup script, just run python3 setup.py on your machine, in the location in which the file is stored.

2. Run the Auto Power On/Off Script

For running the script, ensure that the automation script is already added into the folder you want within Windows. Next, all you need to do is to run the command python3 AutoScaleALL.py. Once it is completed running, the instances ADW and OAC will then be turned on/off base on your tags on the instances.

If the tag is 0, the instance will be turned off. If it is 1 it will be turned on.



Create the Scheduler to run script when needed on Task Scheduler

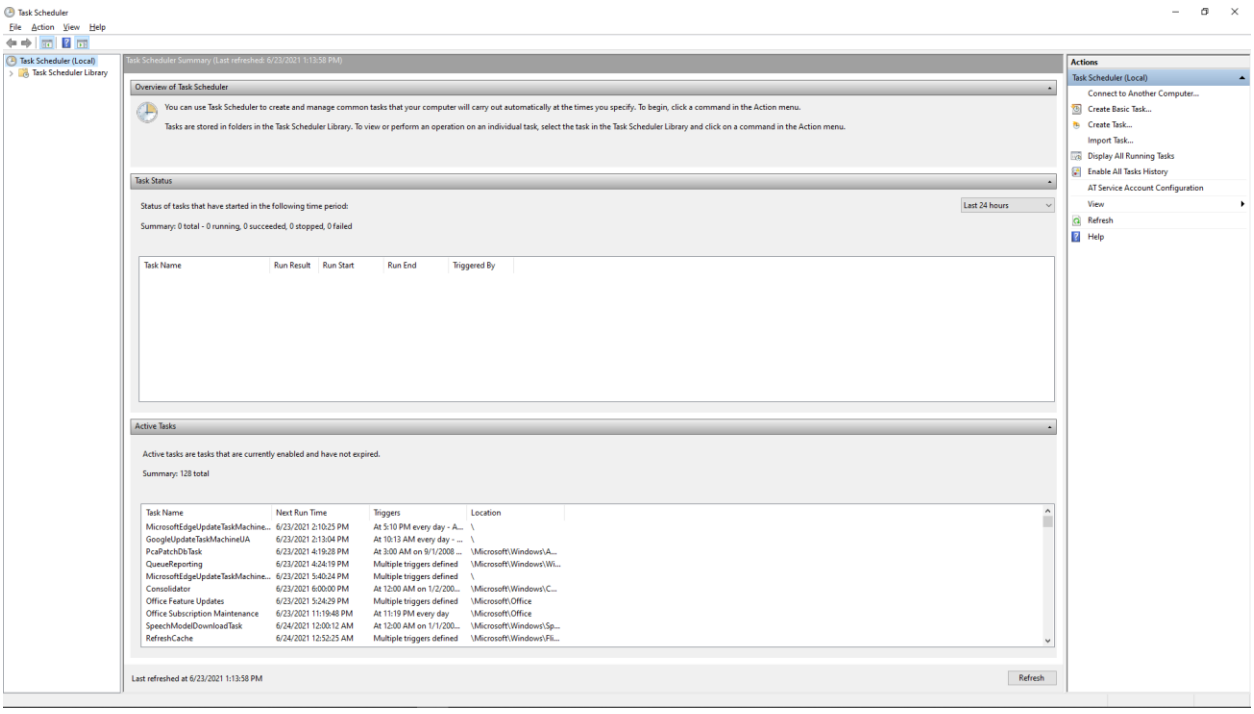
In Windows, schedule scaling or power on/ off with windows task scheduler in every hour by using Windows program called Task Scheduler

1. First, let's create a Bat File.

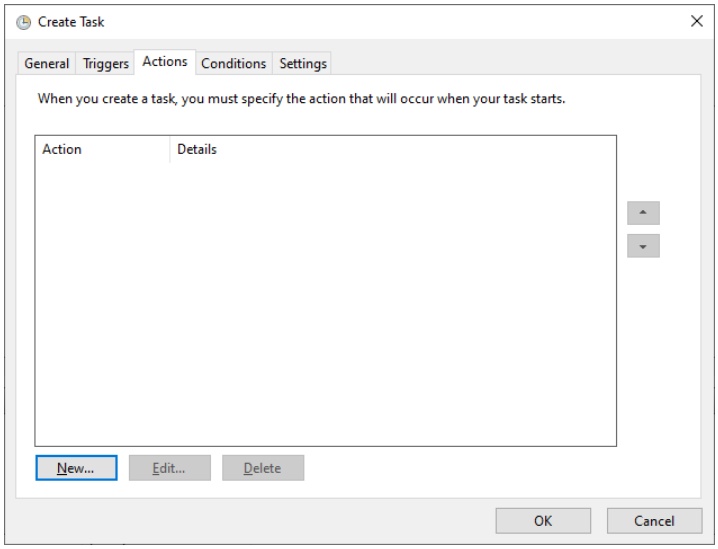
We will create a .bat file called "Scaling.bat" this file will be used to create the schedule in order to run the script

```
::set variables. Date and time should be adjusted based on your system format
set YYYY=%date:~0,4% set
MM=%date:~5,2% set
dd=%date:~8,2% set
HH=%time:~0,2% set
min=%time:~3,2% set
ss=%time:~6,2% :: python
log.bat cd
C:\xxxx\xxxx\xxxx
python ADW_Scaling.py > log_%YYYY%%MM%%dd%_%HH%%min%%ss%.txt 2>&1 ::pause
```

2. Execute Bat File with Windows Task Scheduler



In the task scheduler, just click on “Create Task” and go the the actions tab.



Click on “New” and browse the “scaling.bat” file in which we have created. With that and click on OK. There you have it! Your first automated script is ready to run on Windows!

Activity Complete

And that is all you need to do to completely implement the scripts on Windows as well as create a scheduler to run the script base on your requirement. Now you can easily scale as well as turn instances On/Off on your own accord. From the Admin's point, you will not need to again access the scripts or scheduling to change the requirement. The tags on OCI instance is the only thing needed to be changed.