Launching AWS EC2 instance using awscli and python

Prerequisites

To run the ec2 commands in this topic, you need to:

Have AWS account with privileges
Set your IAM permissions to allow for Amazon EC2 access.

Install and configure the awscli (in powershell)

Setting Up the AWS EC2 Console:
- For accessing the AWS services from python code we first need to create a user and give him programmatic access using Amazon console.
- Launch IAM console
- Add user
- Then provide a username and give programmatic access to it and then click Next.
- Now provide the necessary permission related to the user, this user might belong to a group and their policies can be directly attached to the user, or we can copy the policies of an existing user, or we can directly attach an existing policy as we are going to provide here.

- Tag is optional we can skip it, review the permissions, and create the user finally. Download the CSV as this is the last time it is available for download, this file contains the access key ID and the secret key which will be used next in code.

- Create a key pair and a security group.

- Select an Amazon Machine Image (AMI) and note the AMI ID.
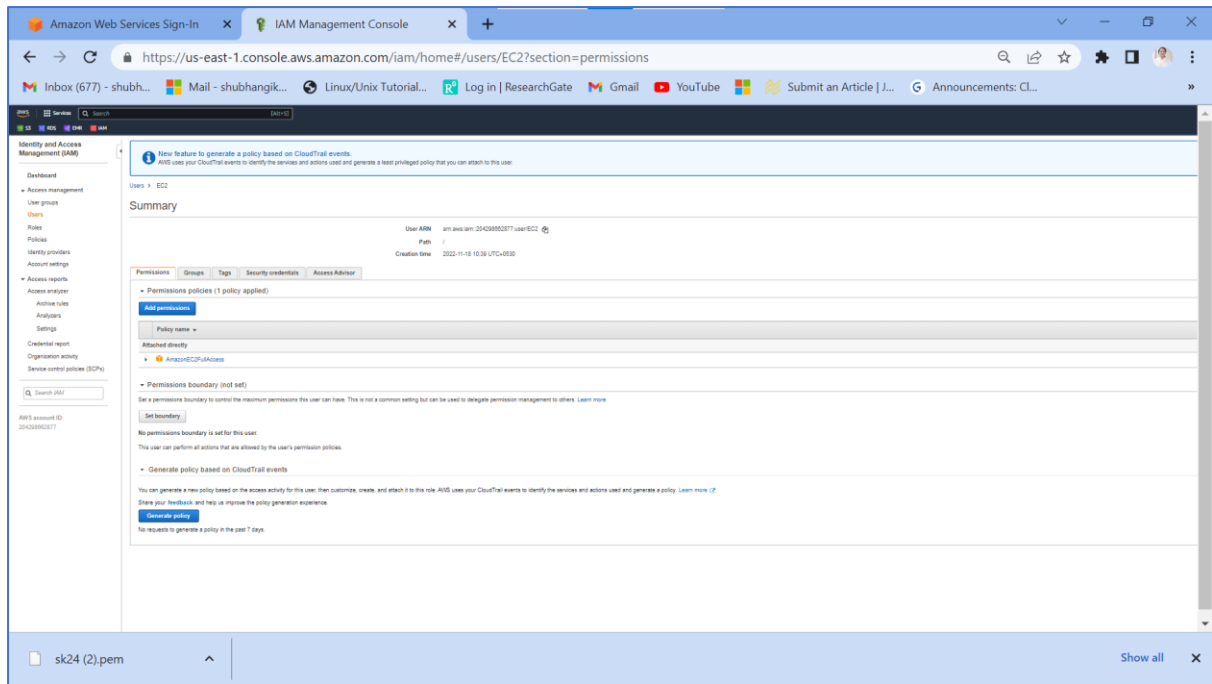
Now we are all set to launch our EC2 instance using python code. For making a connection with EC2 instance we use boto3's client API. The client API takes in following arguments to make a connection with AWS the service.

- **Service Name:** The service to which connection has to be established.
- **Region:** Amazon EC2 is hosted in multiple locations worldwide. Based on our need we can choose our region, we have taken Asia-pacific as our region('ap-south-1').
- **aws_access_key_id:** AWS Security Credentials. Paste the downloaded ID in the blank space.
- **aws_secret_access_key:** AWS Security Credentials. Paste the downloaded secret key in the blank space.

Implementation:

- Python3

```python
#Python Program for creating a connection
import boto3

ec2 = boto3.client('ec2',
                   'us-east-1',
                   aws_access_key_id='',
                   aws_secret_access_key='')


#This function will describe all the instances
#with their current state
response = ec2.describe_instances()
print(response)
```

Creating an instance:

The **ec2.run_instances** launches the specified number of instances using an AMI for which you have permissions. It provides a variety of launch configurations, but we can launch instances with few of the following arguments.

- **InstanceType:** The instance type that you specify determines the hardware of the host computer used for your instance. Each instance type offers different compute, memory, and storage capabilities and are grouped in instance families based on these capabilities. However, AWS provides "t2.micro" as free in the Free Tier limit.
- **MaxCount:** The maximum number of instances to launch. If MaxCount > available instances in target Availability Zone, then it launches the maximum number of Instances greater than MinCount.
- **MinCount:** The minimum number of instances to launch. If available instances in target Availability Zone < MinCount, then no instances are launched.
- **ImageId:** The ID of the AMI used to launch the instance. For our case we have chosen Ubuntu Server 18.04 LTS (HVM), SSD Volume Type (ami-02d55cb47e83a99a0).

Implementation:

- Python3

```python
#Python Program for creating a connection

import boto3


#Function for connecting to EC2

ec2 = boto3.client('ec2',
                   'us-east-1',
                   aws_access_key_id='',
                   aws_secret_access_key='')


#Function for running instances

conn = ec2.run_instances(InstanceType="t2.micro",
                         MaxCount=1,
                         MinCount=1,
                         ImageId="ami-02d55cb47e83a99a0")

print(conn)
```

Commands in windows powdershell:

Run:

pip install boto3 and **pip install awscli** separately

OR

C:\Users\Shubhangi>pip install awscli boto3

Collecting awscli

  Downloading awscli-1.27.12-py3-none-any.whl (3.9 MB)

  |████████████████████████████████| 3.9 MB 6.4 MB/s

Requirement already satisfied: boto3 in
c:\users\shubhangi\appdata\local\programs\python\python310\lib\site-packages (1.26.12)

Collecting colorama<0.4.5,>=0.2.5

  Downloading colorama-0.4.4-py2.py3-none-any.whl (16 kB)

Requirement already satisfied: s3transfer<0.7.0,>=0.6.0 in
c:\users\shubhangi\appdata\local\programs\python\python310\lib\site-packages (from awscli)
(0.6.0)

Requirement already satisfied: botocore==1.29.12 in
c:\users\shubhangi\appdata\local\programs\python\python310\lib\site-packages (from awscli)
(1.29.12)

Collecting rsa<4.8,>=3.1.2

  Downloading rsa-4.7.2-py3-none-any.whl (34 kB)

Collecting PyYAML<5.5,>=3.10

  Downloading PyYAML-5.4.1.tar.gz (175 kB)

  |████████████████████████████████| 175 kB 6.4 MB/s

  Installing build dependencies ... done

  Getting requirements to build wheel ... done

   Preparing wheel metadata ... done

Collecting docutils<0.17,>=0.10

  Downloading docutils-0.16-py2.py3-none-any.whl (548 kB)

  |████████████████████████████████| 548 kB ...

Requirement already satisfied: urllib3<1.27,>=1.25.4 in
c:\users\shubhangi\appdata\local\programs\python\python310\lib\site-packages (from
botocore==1.29.12->awscli) (1.26.12)

Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in c:\users\shubhangi\appdata\local\programs\python\python310\lib\site-packages (from botocore==1.29.12->awscli) (2.8.2)
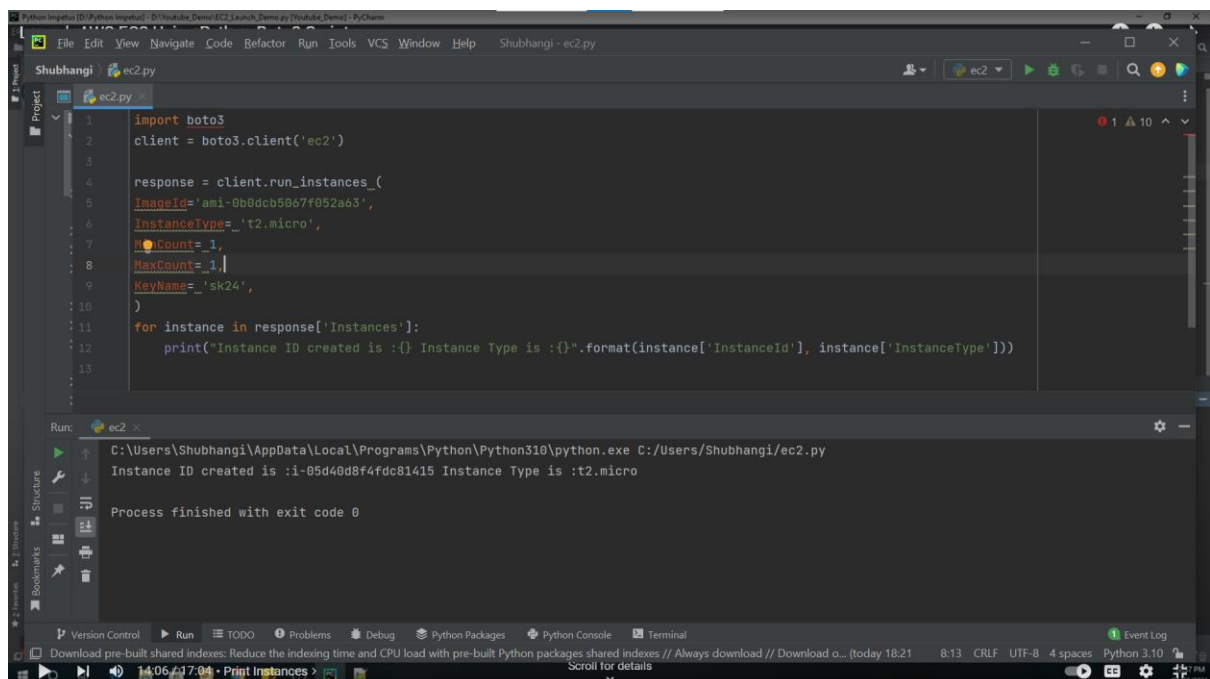
Requirement already satisfied: jmespath<2.0.0,>=0.7.1 in c:\users\shubhangi\appdata\local\programs\python\python310\lib\site-packages (from botocore==1.29.12->awscli) (1.0.1)

Requirement already satisfied: six>=1.5 in c:\users\shubhangi\appdata\local\programs\python\python310\lib\site-packages (from python-dateutil<3.0.0,>=2.1->botocore==1.29.12->awscli) (1.16.0)

Collecting pyasn1>=0.1.3

  Downloading pyasn1-0.4.8-py2.py3-none-any.whl (77 kB)

     |███████████████████████████████| 77 kB 5.1 MB/s

Building wheels for collected packages: PyYAML

  Building wheel for PyYAML (PEP 517) ... done

  Created wheel for PyYAML: filename=PyYAML-5.4.1-cp310-cp310-win_amd64.whl size=45670 sha256=57c62a4304ee6377587d8993c81aae0973bec4d907455bc569dba64441a7efa3

  Stored in directory: c:\users\shubhangi\appdata\local\pip\cache\wheels\c7\0d\22\696ee92245ad710f506eee79bb05c740d8abccd3ecdb778683

Successfully built PyYAML

Installing collected packages: pyasn1, rsa, PyYAML, docutils, colorama, awscli

Successfully installed PyYAML-5.4.1 awscli-1.27.12 colorama-0.4.4 docutils-0.16 pyasn1-0.4.8 rsa-4.7.2

WARNING: You are using pip version 21.2.4; however, version 22.3.1 is available.

You should consider upgrading via the 'C:\Users\Shubhangi\AppData\Local\Programs\Python\Python310\python.exe -m pip install --upgrade pip' command.

C:\Users\Shubhangi>aws configure

AWS Access Key ID [****************UWQQ]:

AWS Secret Access Key [****************MTHh]:

Default region name [us-east-1]:

Default output format [None]:

```
import boto3
client = boto3.client('ec2')

response = client.run_instances (
ImageId='ami-0b0dcb5067f052a63',
InstanceType= 't2.micro',
MinCount= 1,
MaxCount= 1,
KeyName= 'sk24',
)
for instance in response['Instances']:
    print("Instance ID created is :{} Instance Type is
:{}".format(instance['InstanceId'], instance['InstanceType']))
```


Python Code to launch an EC2 Instance:

import boto3

client = boto3.client('ec2')

response = client.run_instances (

ImageId='ami-0b0dcb5067f052a63',

InstanceType= 't2.micro',

MinCount= 1,

MaxCount= 1,

KeyName= 'sk24',

)

for instance in response['Instances']:

   print("Instance ID created is :{} Instance Type is :{}".format(instance['InstanceId'], instance['InstanceType']))
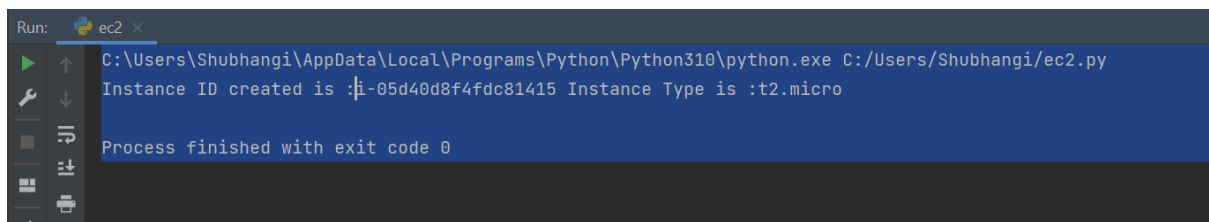
Output:

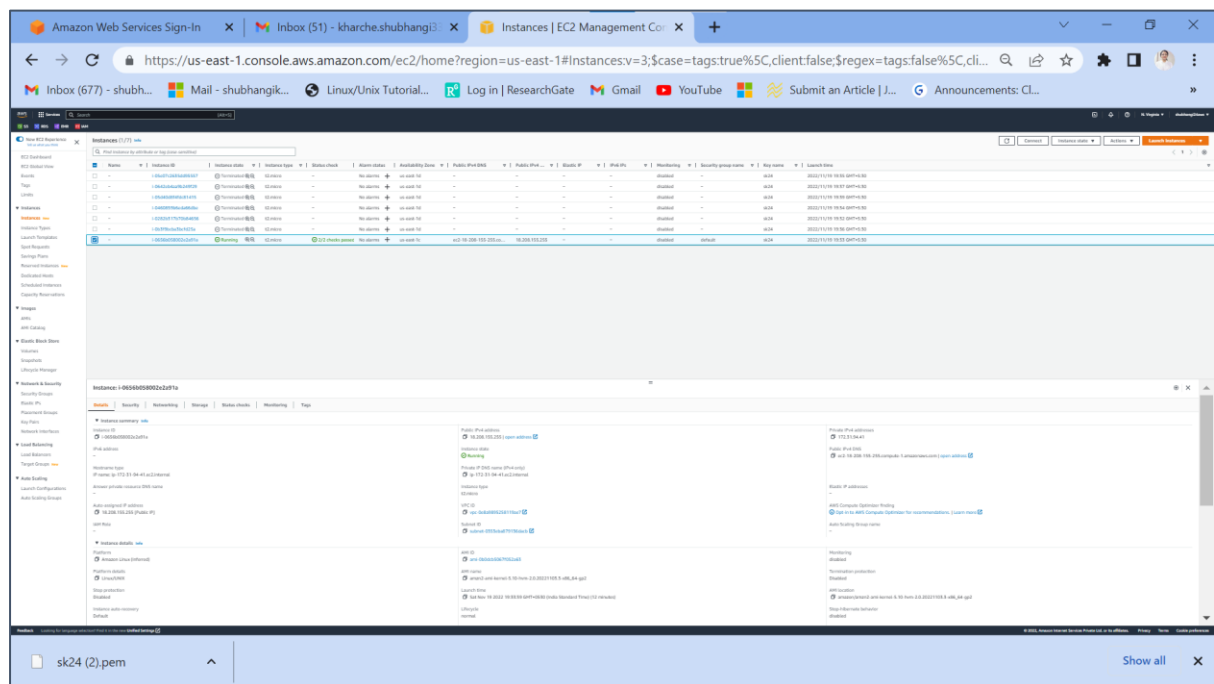C:\Users\Shubhangi\AppData\Local\Programs\Python\Python310\python.exe C:/Users/Shubhangi/ec2.py

Instance ID created is :i-05d40d8f4fdc81415 Instance Type is :t2.micro

Process finished with exit code 0

EC2 instance launched successfully