AWS CLI & BOTO3 SDK FOR PYTHON

ACCENTURE

In this Manual we are going to discuss about the Management of AWS using the command line interface sitting on top of Boto3 SDK for python and this will allows you to manage your services in AWS using command line in windows and terminal in Linux, we will proceed with the steps in configuring the AWS CLI, previously we need to download the package and setting up path are difficult. But the current process is simple. Depends on the use case we can choose any of the below path

- AWS Tools for Windows PowerShell
- AWS SDK for Java
- AWS SDK for .NET
- AWS SDK for JavaScript
- AWS SDK for Ruby
- **AWS SDK for Python (Boto3)**
- AWS SDK for PHP
- AWS SDK for Go
- AWS Toolkit for Eclipse
- AWS Toolkit for Visual Studio
- AWS Mobile SDK for iOS
- AWS Mobile SDK for Android

In our first step we will proceed with installation of CLI tool for windows.

Use the below link to download the CLI for Windows 64 bit -> https://s3.amazonaws.com/aws-cli/AWSCLI64.msi

Use the below link to download the CLI for Windows 32 bit -> https://s3.amazonaws.com/aws-cli/AWSCLI32.msi

Once the installation is completed in a typical manner of windows software installation, please follow the steps for the configuration.

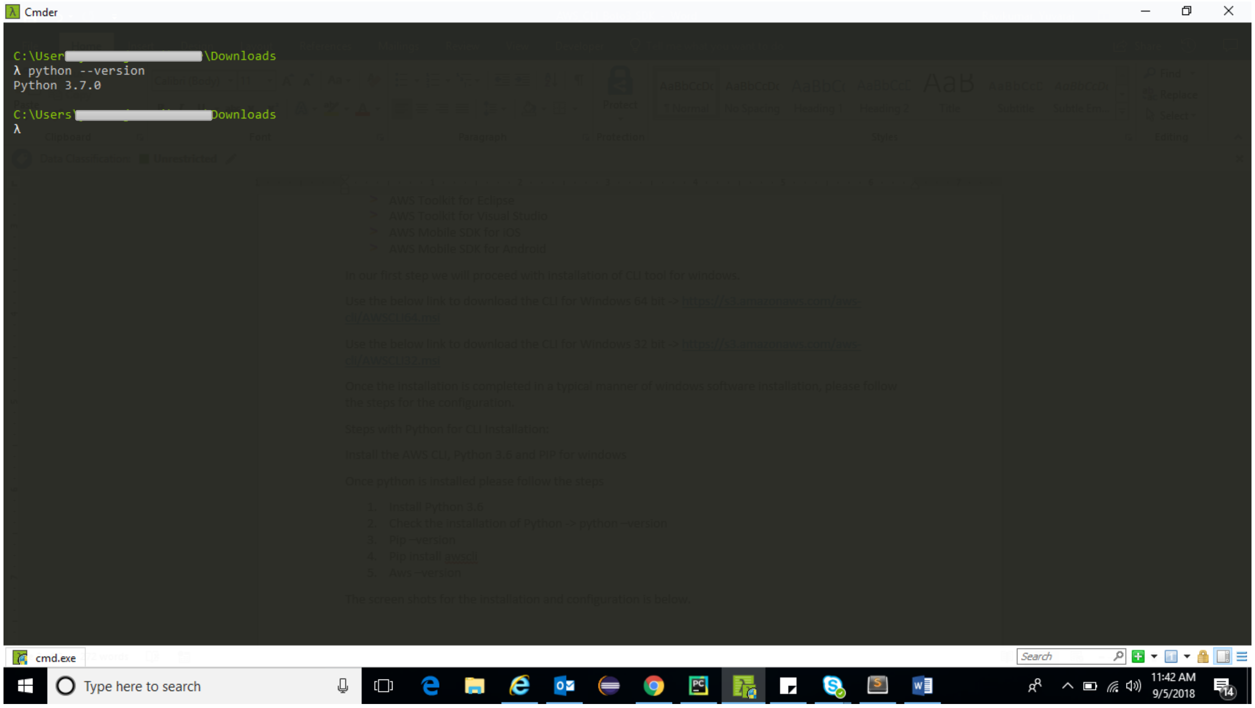
Steps with Python for CLI Installation:

Install the AWS CLI, Python 3.6 and PIP for windows

Once python is installed please follow the steps

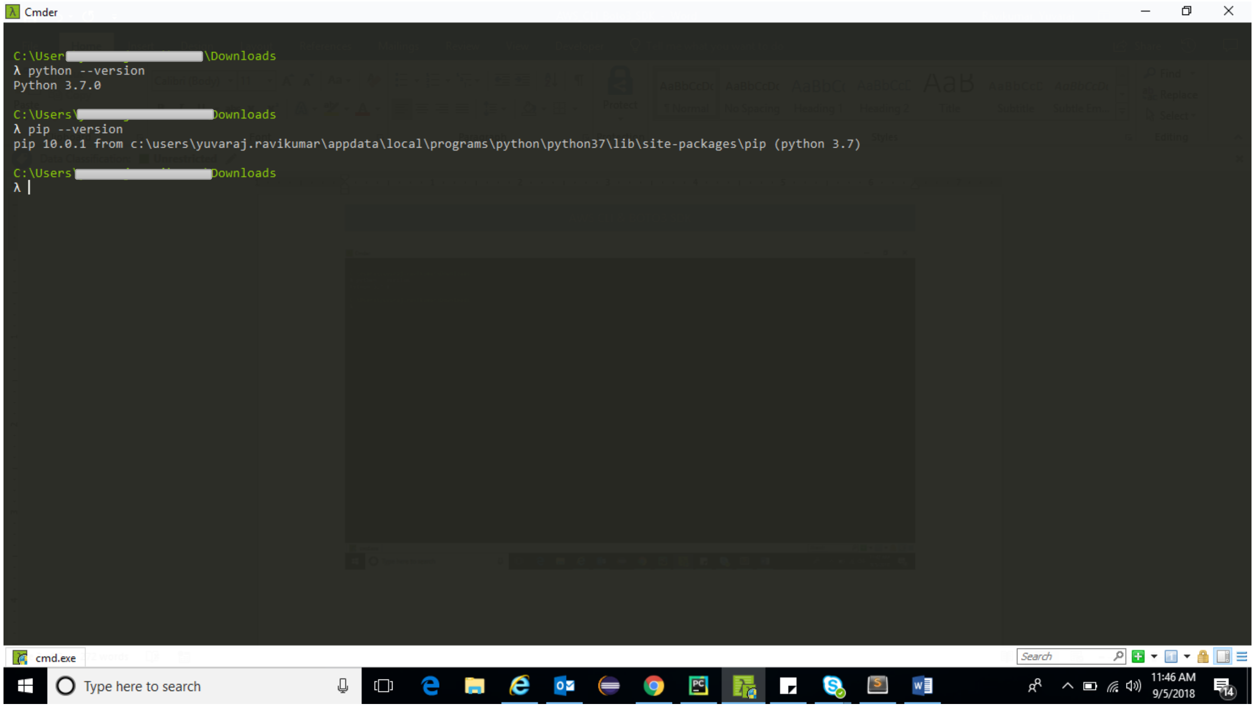
1. Install Python 3.6
2. Check the installation of Python -> python –version
3. Pip –version
4. Pip install awscli
5. Aws –version

The screen shots for the installation and configuration is below.



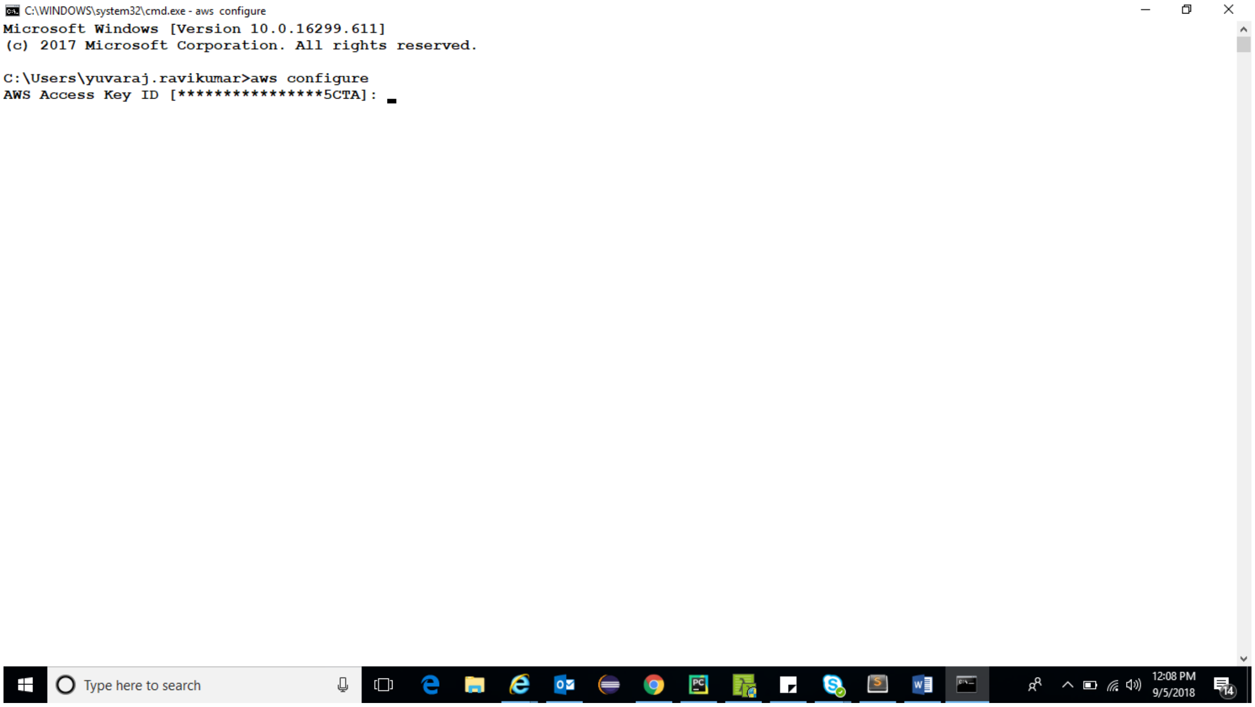
```
C:\User[redacted]\Downloads
λ python --version
Python 3.7.0

C:\Users[redacted]Downloads
λ
```

Check for the pip installation



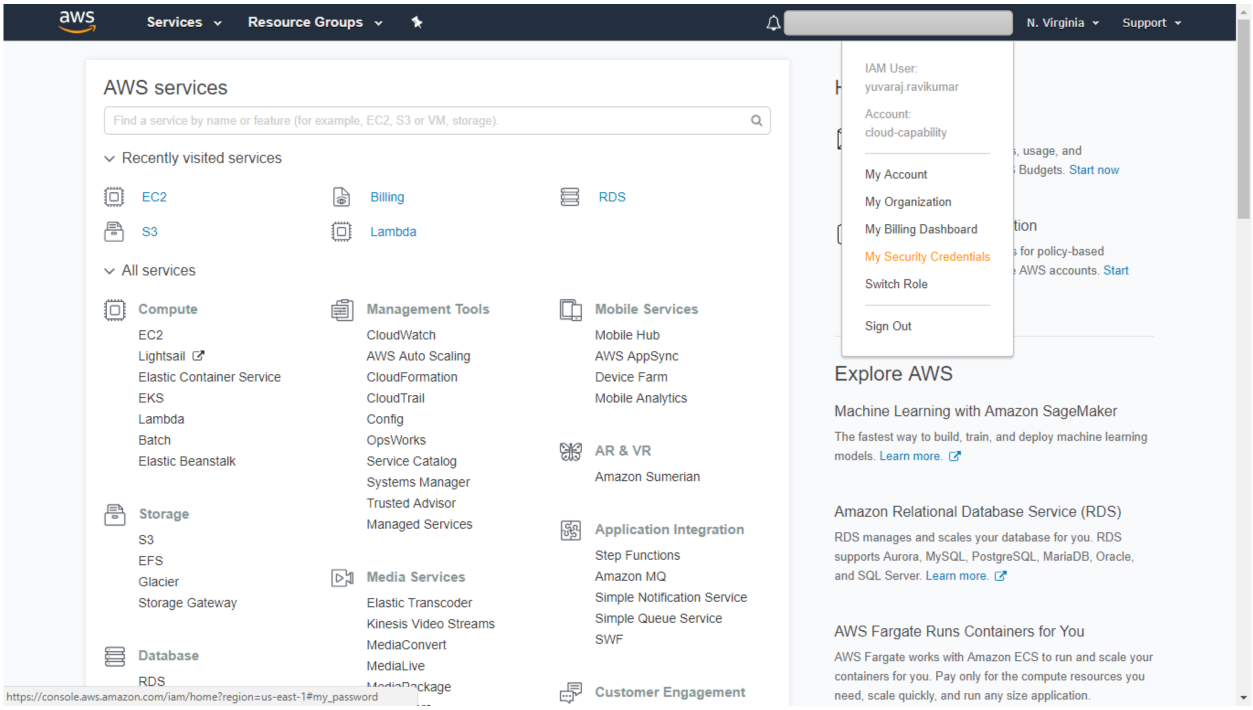
```
C:\User[redacted]\Downloads
λ python --version
Python 3.7.0

C:\Users[redacted]Downloads
λ pip --version
pip 10.0.1 from c:\users\yuvaraj.ravikumar\appdata\local\programs\python\python37\lib\site-packages\pip (python 3.7)

C:\Users[redacted]Downloads
λ |
```

Configure aws cli with access key and secret key



```
Microsoft Windows [Version 10.0.16299.611]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\yuvaraj.ravikumar>aws configure
AWS Access Key ID [****************5CTA]: _
```

Please create your access key and secret key from the management console.



Machine Learning with Amazon SageMaker

The fastest way to build, train, and deploy machine learning models. Learn more.

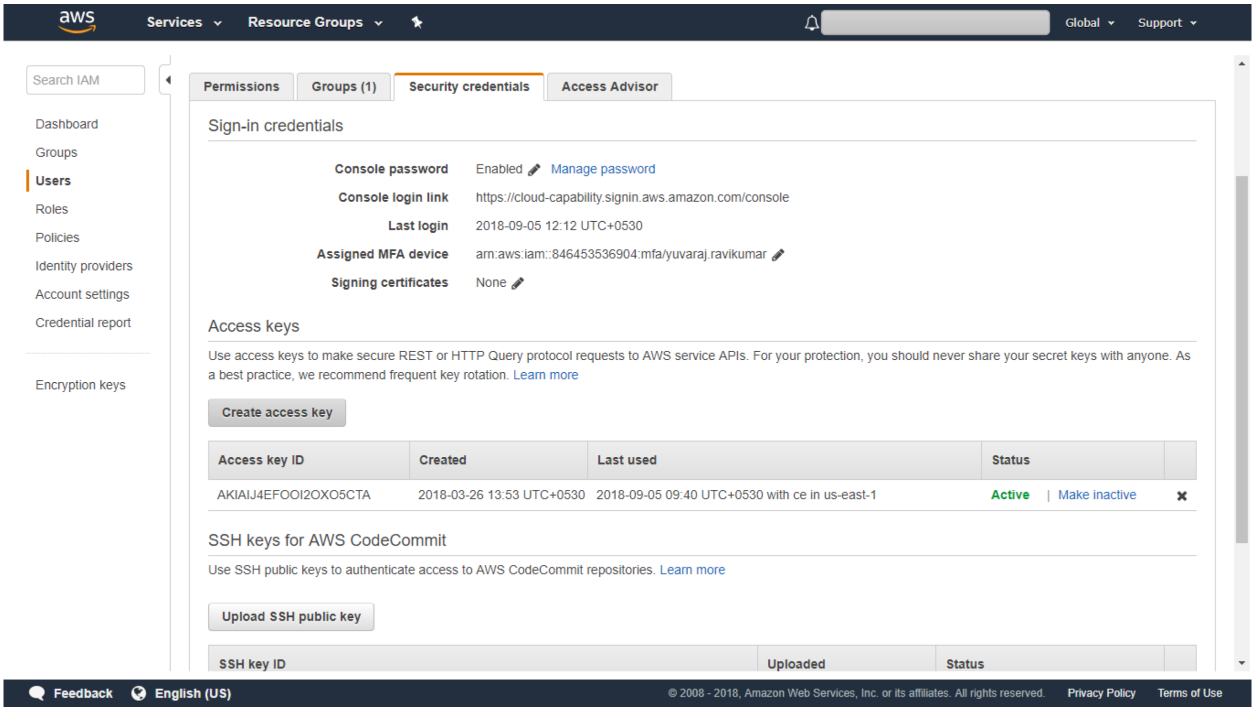
Amazon Relational Database Service (RDS)

RDS manages and scales your database for you. RDS supports Aurora, MySQL, PostgreSQL, MariaDB, Oracle, and SQL Server. Learn more.

AWS Fargate Runs Containers for You

AWS Fargate works with Amazon ECS to run and scale your containers for you. Pay only for the compute resources you need, scale quickly, and run any size application.

Security credentials -> Users -> Create Access Key



Sign-in credentials

Console password	Enabled Manage password
Console login link	https://cloud-capability.signin.aws.amazon.com/console
Last login	2018-09-05 12:12 UTC+0530
Assigned MFA device	arn:aws:iam::846453536904:mfa/yuvaraj.ravikumar
Signing certificates	None

Access keys

Use access keys to make secure REST or HTTP Query protocol requests to AWS service APIs. For your protection, you should never share your secret keys with anyone. As a best practice, we recommend frequent key rotation. Learn more

Access key ID	Created	Last used	Status
AKIAIJ4EFOOI2OXO5CTA	2018-03-26 13:53 UTC+0530	2018-09-05 09:40 UTC+0530 with ce in us-east-1	Active | Make inactive

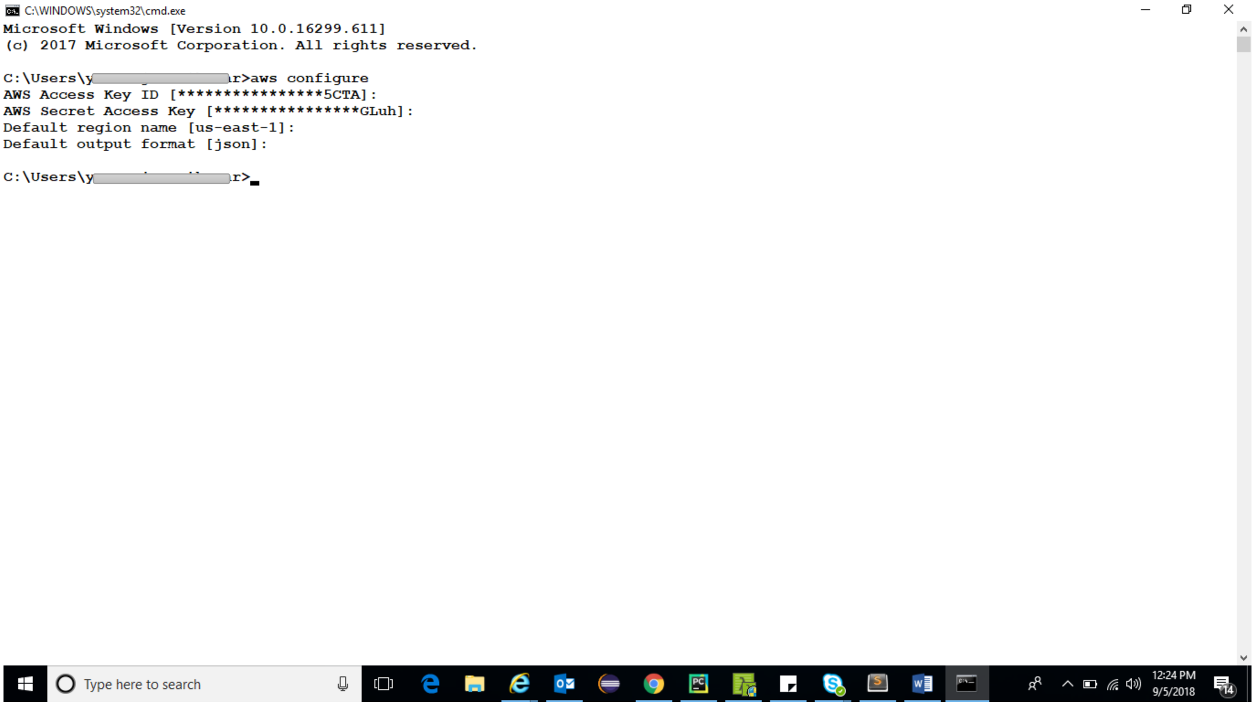
SSH keys for AWS CodeCommit

Use SSH public keys to authenticate access to AWS CodeCommit repositories. Learn more

SSH key ID	Uploaded	Status

© 2008 - 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.

Please use the Access Key and Secret Key for configuring the CLI

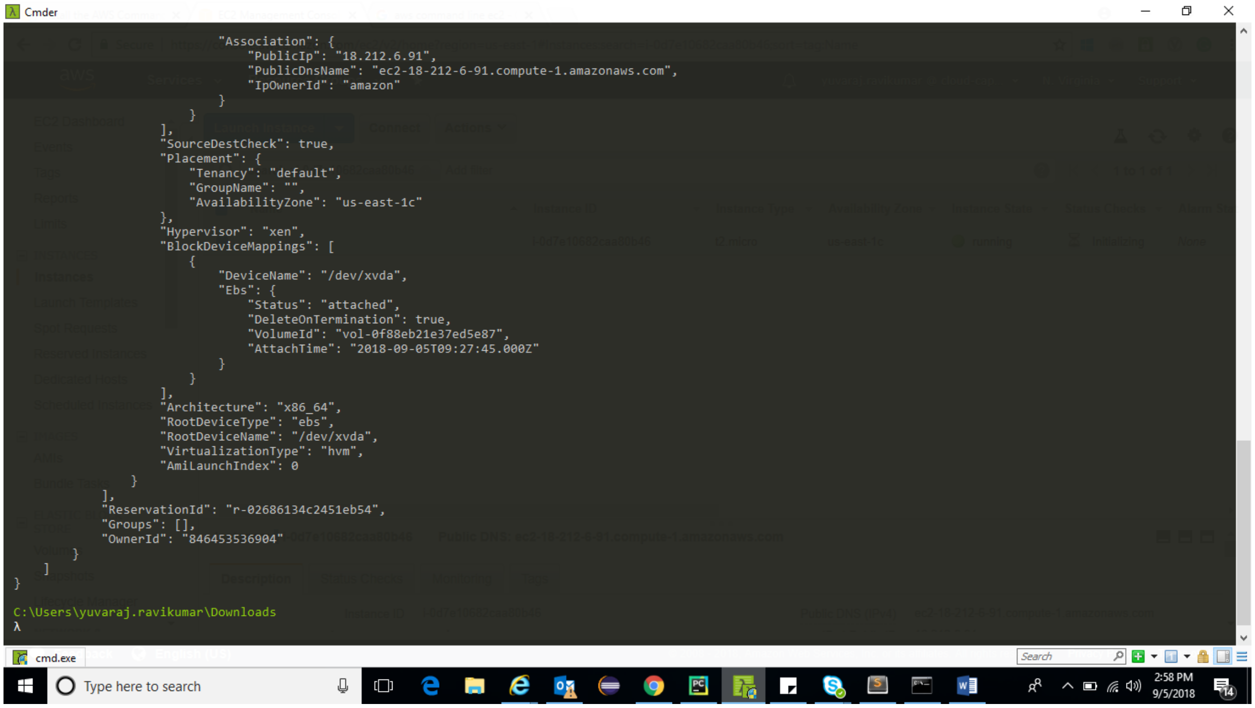


```
Microsoft Windows [Version 10.0.16299.611]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\y[redacted]r>aws configure
AWS Access Key ID [****************5CTA]:
AWS Secret Access Key [****************GLuh]:
Default region name [us-east-1]:
Default output format [json]:

C:\Users\y[redacted]r>_
```

Once the access and secret key is given and configured then we can start management using the CLI commands as documented from the below link.

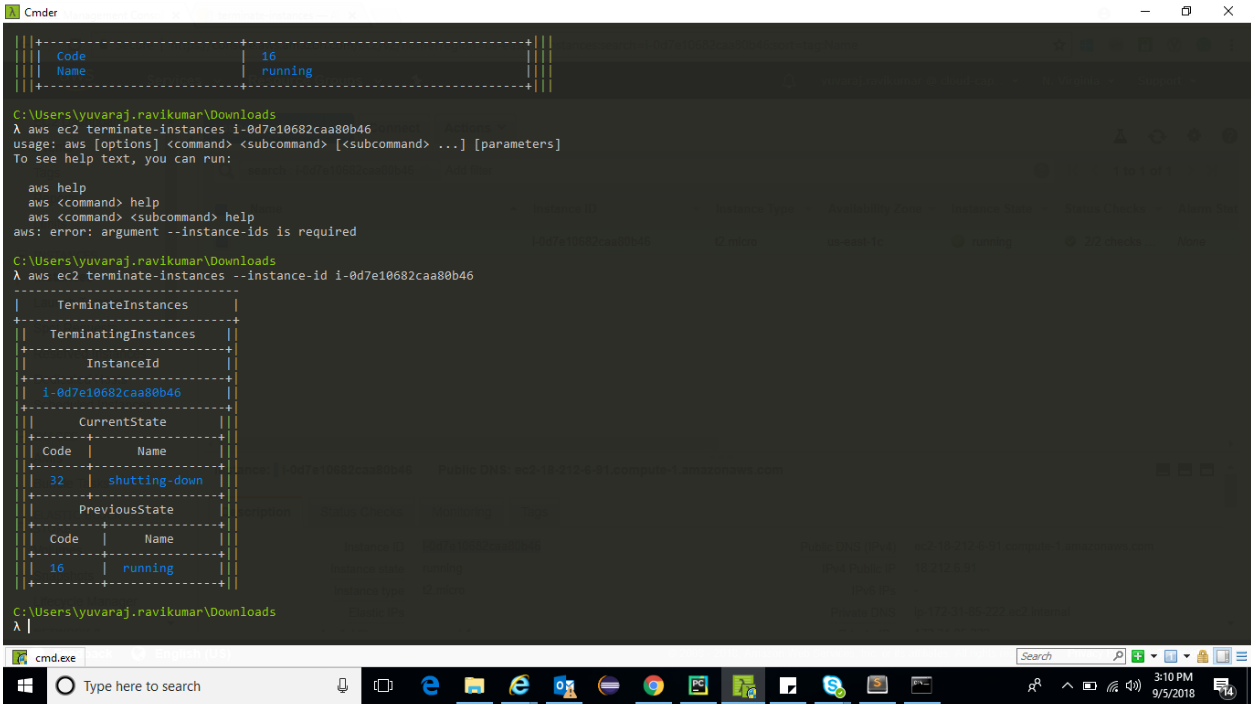
http://docs.aws.amazon.com/cli/latest/reference/

EC2 – Describe Instance command



```
                        "Association": {
                            "PublicIp": "18.212.6.91",
                            "PublicDnsName": "ec2-18-212-6-91.compute-1.amazonaws.com",
                            "IpOwnerId": "amazon"
                        }
                    }
                ],
                "SourceDestCheck": true,
                "Placement": {
                    "Tenancy": "default",
                    "GroupName": "",
                    "AvailabilityZone": "us-east-1c"
                },
                "Hypervisor": "xen",
                "BlockDeviceMappings": [
                    {
                        "DeviceName": "/dev/xvda",
                        "Ebs": {
                            "Status": "attached",
                            "DeleteOnTermination": true,
                            "VolumeId": "vol-0f88eb21e37ed5e87",
                            "AttachTime": "2018-09-05T09:27:45.000Z"
                        }
                    }
                ],
                "Architecture": "x86_64",
                "RootDeviceType": "ebs",
                "RootDeviceName": "/dev/xvda",
                "VirtualizationType": "hvm",
                "AmiLaunchIndex": 0
            }
        ],
        "ReservationId": "r-02686134c2451eb54",
        "Groups": [],
        "OwnerId": "846453536904"
    }
  ]
}

C:\Users\yuvaraj.ravikumar\Downloads
λ
```

In a similar way you can use the commands from the above link for any of the AWS services.



```
|||+-------------------------+-------------------------------------+|||
||||  Code                   |  16                                 ||||
||||  Name                   |  running                            ||||
|||+-------------------------+-------------------------------------+|||

C:\Users\yuvaraj.ravikumar\Downloads
λ aws ec2 terminate-instances i-0d7e10682caa80b46
usage: aws [options] <command> <subcommand> [<subcommand> ...] [parameters]
To see help text, you can run:

  aws help
  aws <command> help
  aws <command> <subcommand> help
aws: error: argument --instance-ids is required

C:\Users\yuvaraj.ravikumar\Downloads
λ aws ec2 terminate-instances --instance-id i-0d7e10682caa80b46
-------------------------------
|     TerminateInstances      |
+-----------------------------+
||    TerminatingInstances   ||
|+---------------------------+|
||         InstanceId        ||
|+---------------------------+|
||  i-0d7e10682caa80b46      ||
|+---------------------------+|
|||       CurrentState      |||
||+-------+-----------------+||
||| Code  |      Name       |||
||+-------+-----------------+||
|||  32   |  shutting-down  |||
||+-------+-----------------+||
|||       PreviousState     |||
||+---------+---------------+||
|||  Code   |     Name      |||
||+---------+---------------+||
|||  16     |   running     |||
||+---------+---------------+||

C:\Users\yuvaraj.ravikumar\Downloads
λ |
```

I have terminated the instance created from command line.

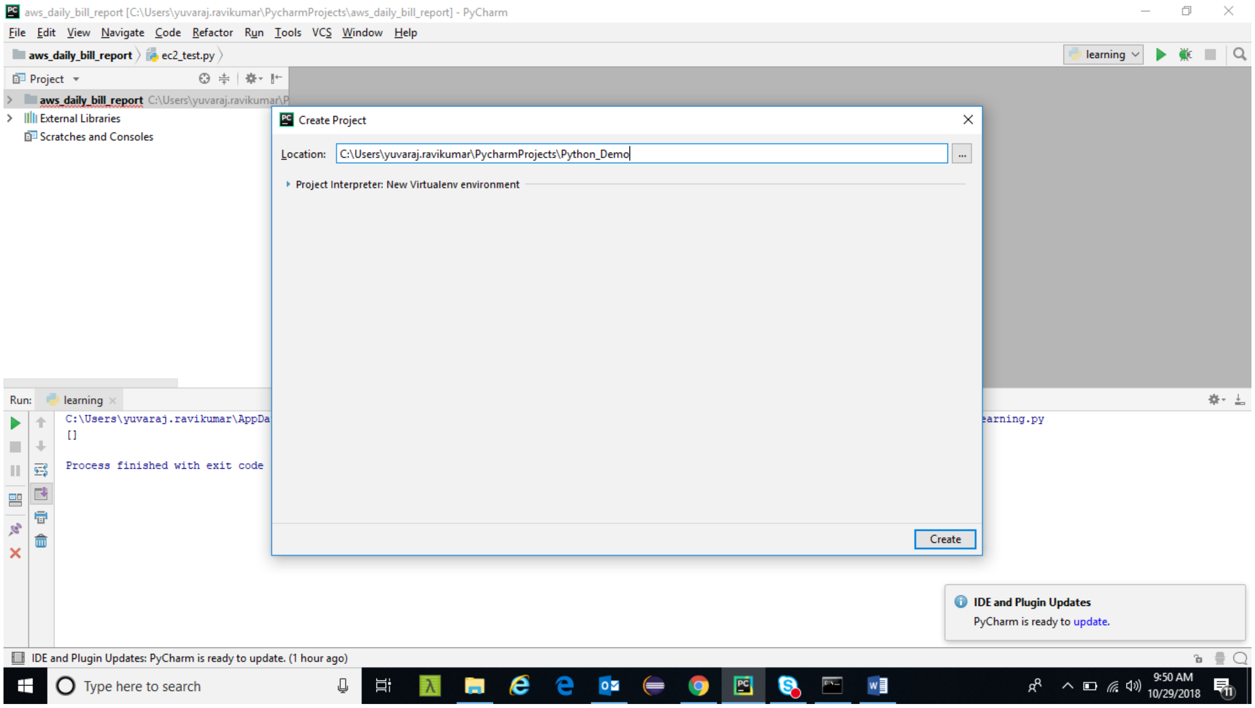
Now we can proceed with the Boto3 SDK for Management and Automation in Aws.

We need to install any one IDE for Python, here am installing PyCharm and make sure python is installed already and environment path is set.

Once PyCharm is installed with the above configurations and setup, proceed for creating the new project in python.

While working with Boto3 SDK please have the documentation alongside for reference.

Boto3 Documentation Link → https://boto3.amazonaws.com/v1/documentation/api/latest/index.html



aws_daily_bill_report [C:\Users\yuvaraj.ravikumar\PycharmProjects\aws_daily_bill_report] - PyCharm

Create Project

Location: C:\Users\yuvaraj.ravikumar\PycharmProjects\Python_Demo

Project Interpreter: New Virtualenv environment

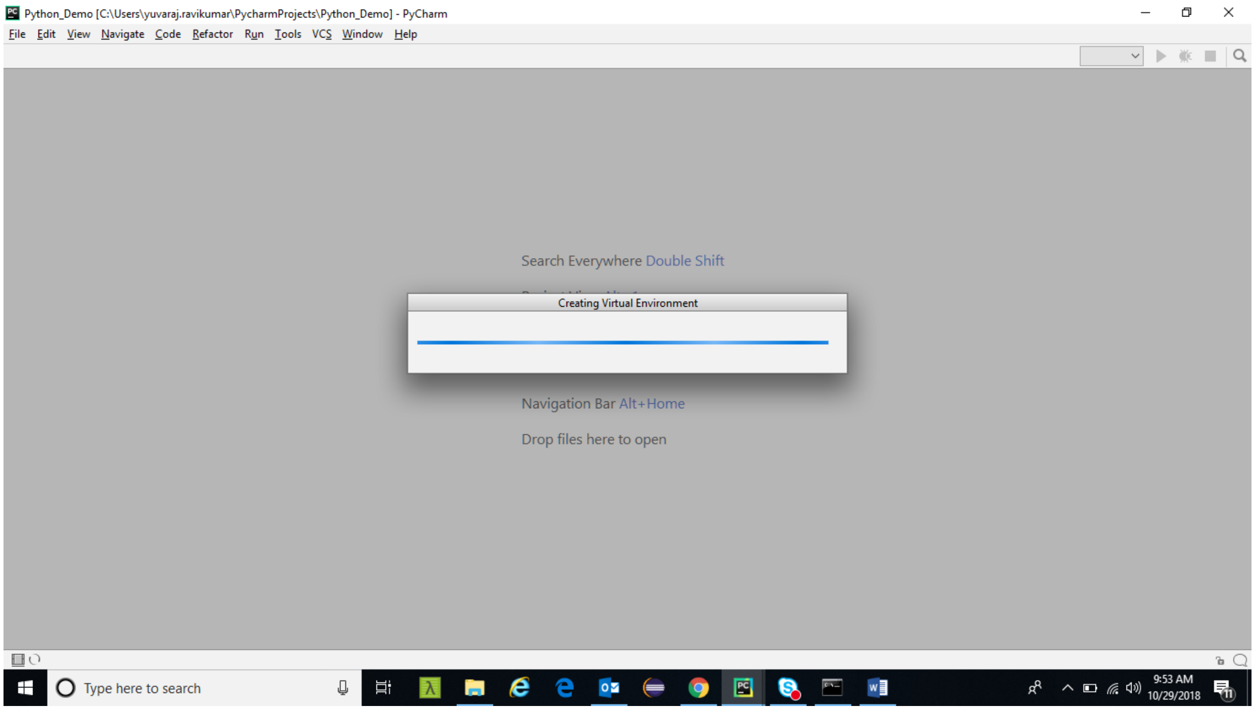
```
C:\Users\yuvaraj.ravikumar\AppDa
[]

Process finished with exit code
```

IDE and Plugin Updates

PyCharm is ready to update.

IDE and Plugin Updates: PyCharm is ready to update. (1 hour ago)

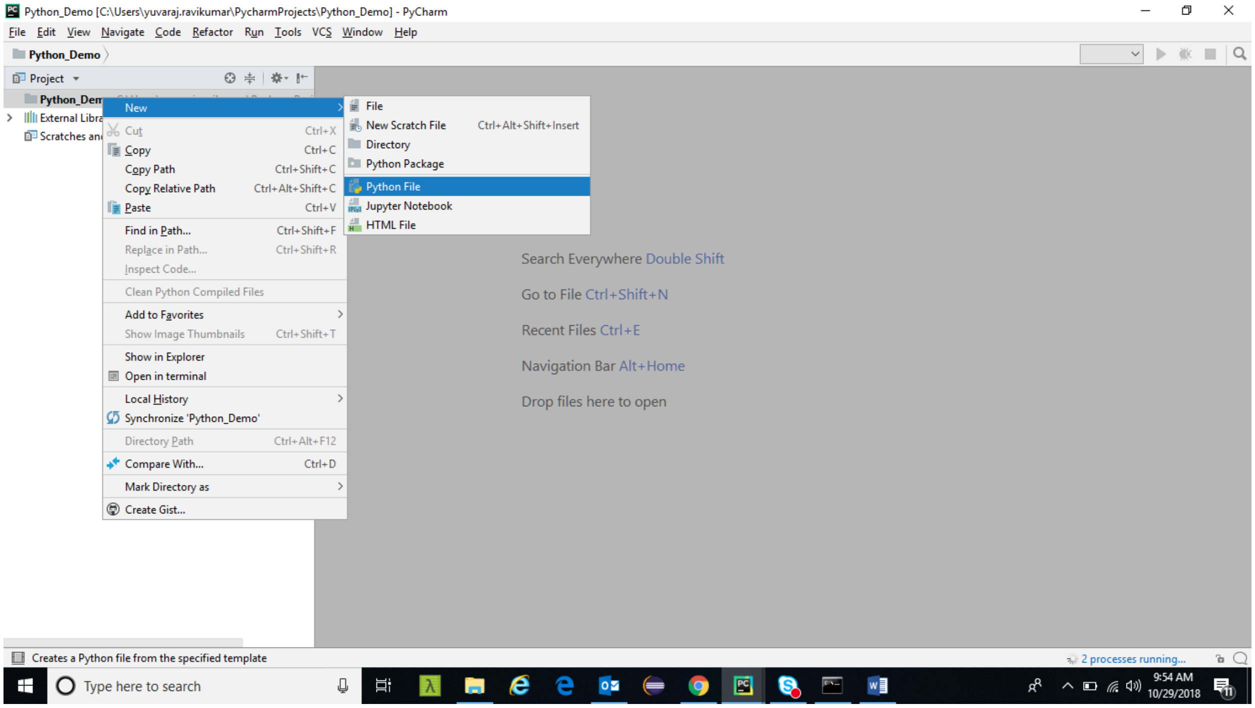
Create a New Project and proceed with the sample template below for creating EC2 Instance with reference to Boto3 Documentation.



Python_Demo [C:\Users\yuvaraj.ravikumar\PycharmProjects\Python_Demo] - PyCharm

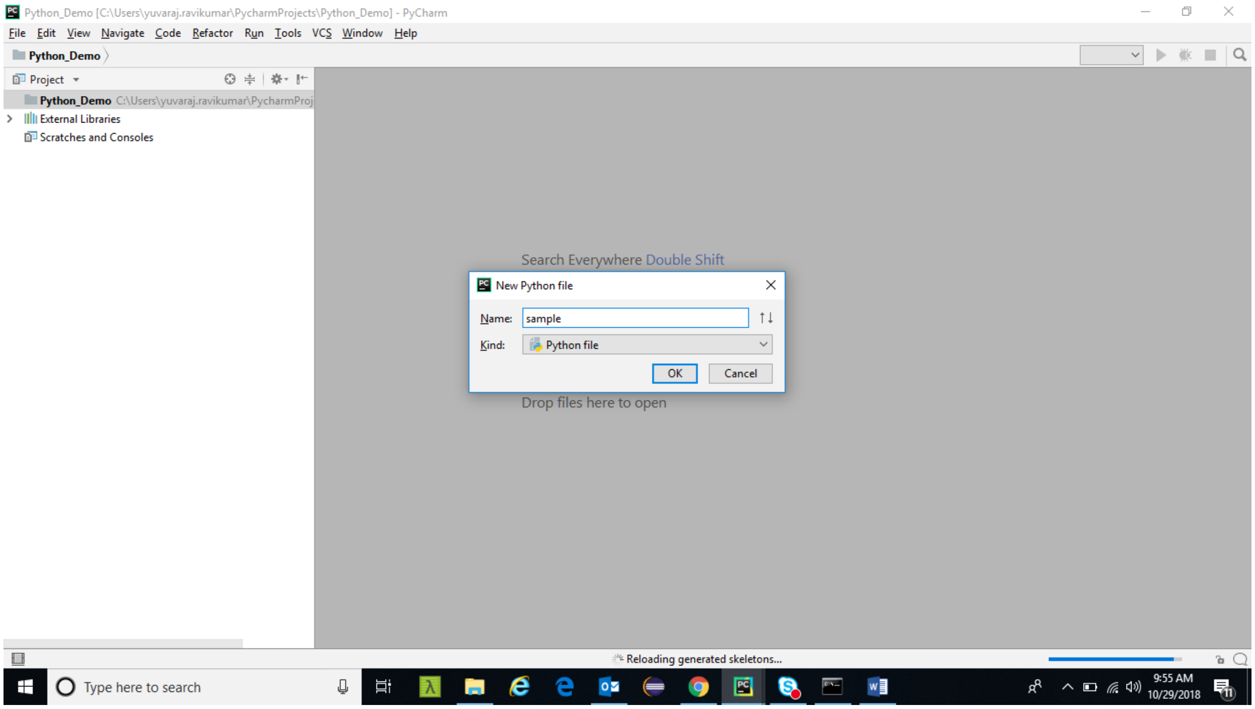
Creating Virtual Environment

File Edit View Navigate Code Refactor Run Tools VCS Window Help
Search Everywhere Double Shift
Navigation Bar Alt+Home
Drop files here to open
Type here to search
9:53 AM
10/29/2018



Python_Demo [C:\Users\yuvaraj.ravikumar\PycharmProjects\Python_Demo] - PyCharm

Creates a Python file from the specified template

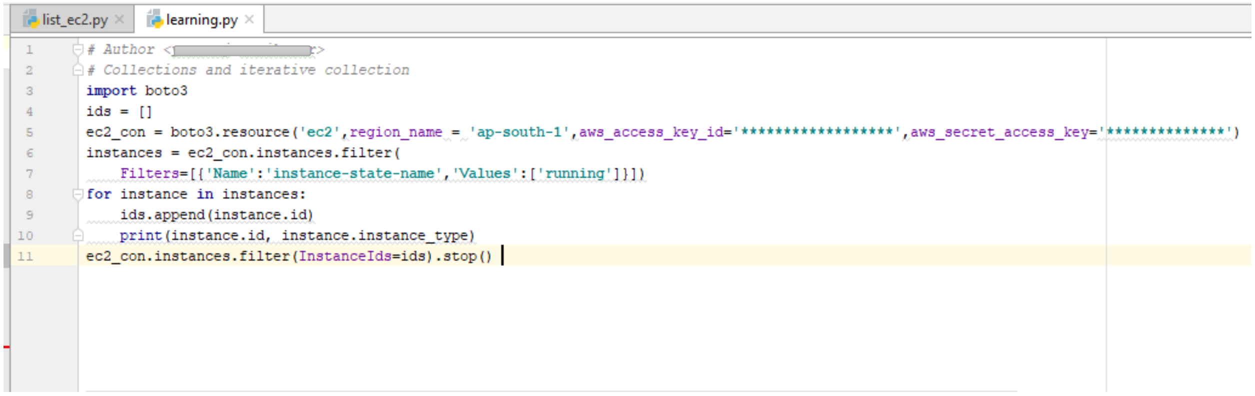

Python_Demo [C:\Users\yuvaraj.ravikumar\PycharmProjects\Python_Demo] - PyCharm
File Edit View Navigate Code Refactor Run Tools VCS Window Help
Python_Demo
Project
Python_Demo C:\Users\yuvaraj.ravikumar\PycharmProj
External Libraries
Scratches and Consoles
Search Everywhere Double Shift
Drop files here to open
Reloading generated skeletons...
Type here to search
9:55 AM
10/29/2018

New Python file

Name: sample

Kind: Python file

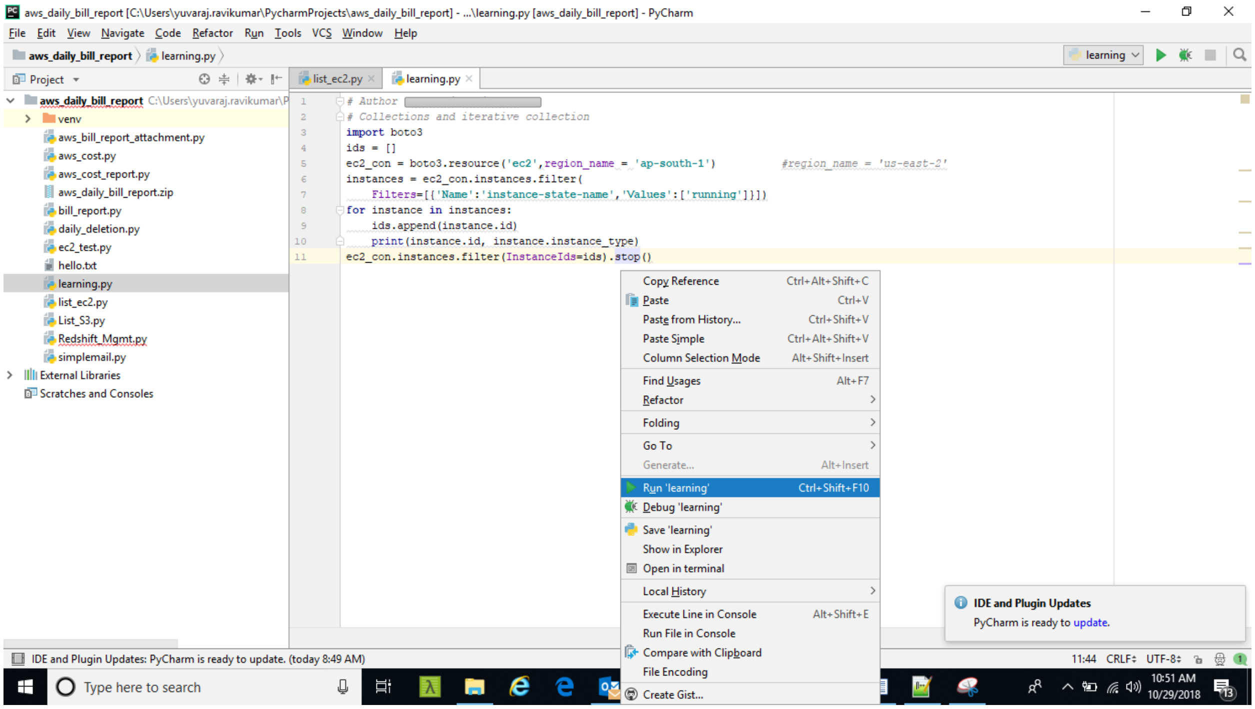
Please proceed with the below sample code to stop the instances running via python code using Boto3 SDK



```
# Author <[redacted]>
# Collections and iterative collection
import boto3
ids = []
ec2_con = boto3.resource('ec2',region_name = 'ap-south-1',aws_access_key_id='******************',aws_secret_access_key='**************')
instances = ec2_con.instances.filter(
    Filters=[{'Name':'instance-state-name','Values':['running']}])
for instance in instances:
    ids.append(instance.id)
    print(instance.id, instance.instance_type)
ec2_con.instances.filter(InstanceIds=ids).stop()
```

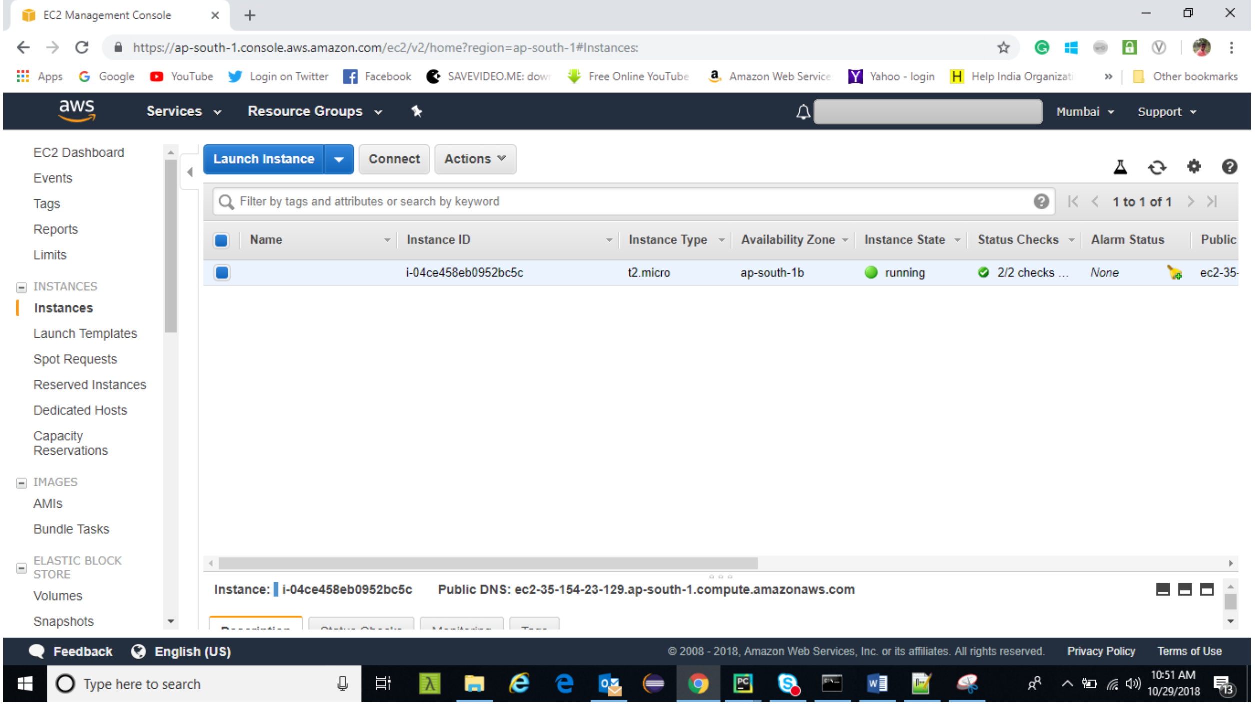
The above snipped of code will lists the available instances in the region and stop all the instances

Replace the access keys and secret keys with your access and secret keys respectively.



aws_daily_bill_report [C:\Users\yuvaraj.ravikumar\PycharmProjects\aws_daily_bill_report] - ...\learning.py [aws_daily_bill_report] - PyCharm

```
# Author [redacted]
# Collections and iterative collection
import boto3
ids = []
ec2_con = boto3.resource('ec2',region_name = 'ap-south-1')        #region_name = 'us-east-2'
instances = ec2_con.instances.filter(
    Filters=[{'Name':'instance-state-name','Values':['running']}])
for instance in instances:
    ids.append(instance.id)
    print(instance.id, instance.instance_type)
ec2_con.instances.filter(InstanceIds=ids).stop()
```



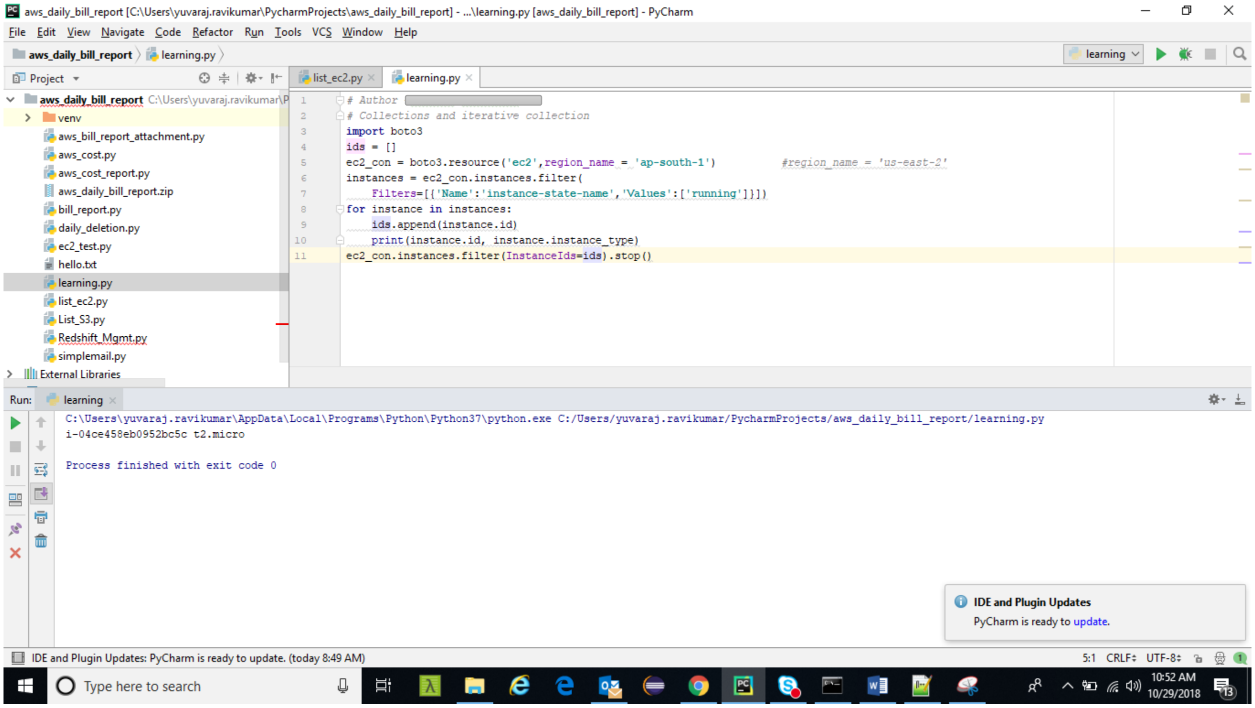
EC2 Management Console

https://ap-south-1.console.aws.amazon.com/ec2/v2/home?region=ap-south-1#Instances:

Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public
	i-04ce458eb0952bc5c	t2.micro	ap-south-1b	running	2/2 checks ...	None	ec2-35-

Instance: i-04ce458eb0952bc5c Public DNS: ec2-35-154-23-129.ap-south-1.compute.amazonaws.com

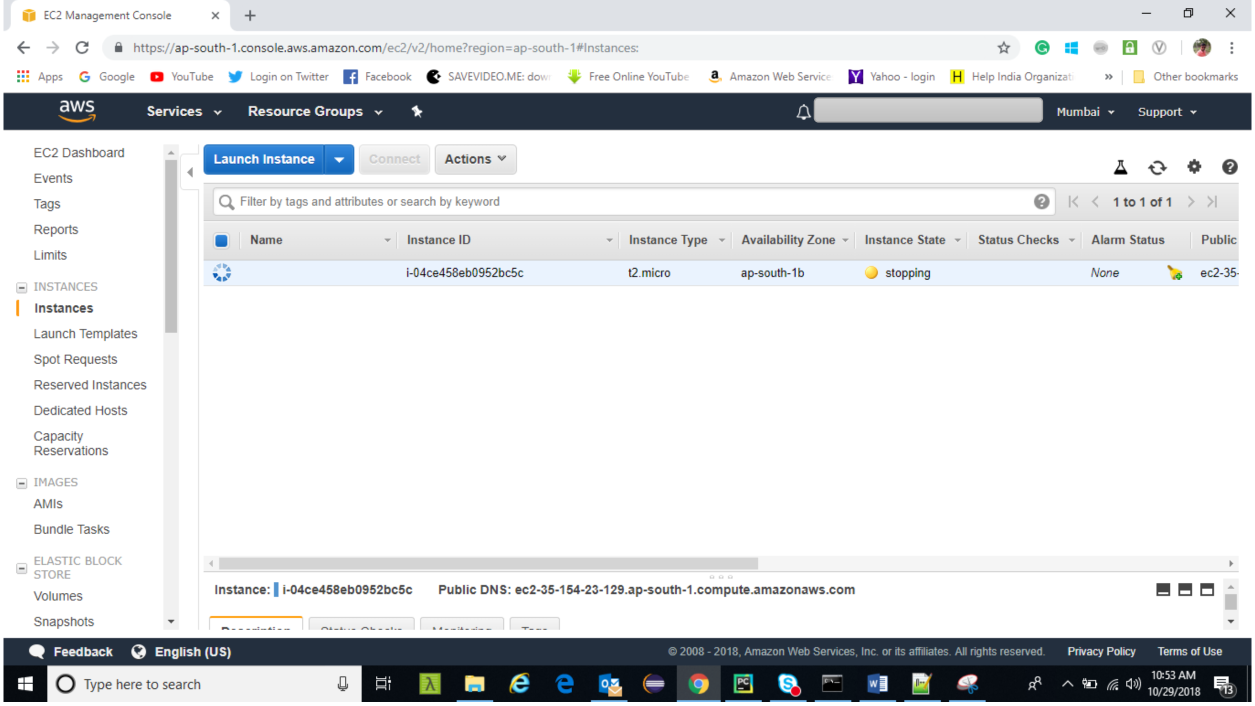
The above instance is running in the Mumbai region ap-south-1



```
# Author [redacted]
# Collections and iterative collection
import boto3
ids = []
ec2_con = boto3.resource('ec2',region_name = 'ap-south-1')          #region_name = 'us-east-2'
instances = ec2_con.instances.filter(
    Filters=[{'Name':'instance-state-name','Values':['running']}])
for instance in instances:
    ids.append(instance.id)
    print(instance.id, instance.instance_type)
ec2_con.instances.filter(InstanceIds=ids).stop()
```

```
C:\Users\yuvaraj.ravikumar\AppData\Local\Programs\Python\Python37\python.exe C:/Users/yuvaraj.ravikumar/PycharmProjects/aws_daily_bill_report/learning.py
i-04ce458eb0952bc5c t2.micro

Process finished with exit code 0
```

The instance is listed and stopped.



Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public
	i-04ce458eb0952bc5c	t2.micro	ap-south-1b	stopping		None	ec2-35-

Instance: i-04ce458eb0952bc5c   Public DNS: ec2-35-154-23-129.ap-south-1.compute.amazonaws.com

In a similar way we can do automation on AWS account with Python Boto3.

Hope you all uncovering the document accessible and useful.

If you have any demanding question which is mentioned in the document, please feel free to contact us.

Happy Learning

LKM