Table of contents

AWS CLI TOOL AND BOTO3

- Install `python3` using `yum`

```
whereis python
sudo yum install python3 -y
```

Using Python pip

- **pip** is the package installer for Python. You can use pip to install packages from the `Python Package Index`.
- To check Python Version use `python3 --version`
- To validate if a python package is present without going into Python Shell, use:

```
python3 -c "import boto3"
sudo python3 -m pip install boto3
```

--

- Using the `pip` command, install the AWS CLI and Boto3:

```
sudo pip3 install boto3 -U
python3 -c "import boto3"
```

```
[ec2-user@ip-172-31-20-228 ~]$ python3 -c "import boto3"
Traceback (most recent call last):
  File "<string>", line 1, in <module>
ModuleNotFoundError: No module named 'boto3'
[ec2-user@ip-172-31-20-228 ~]$ whereis pip
pip: /usr/bin/pip3.7
[ec2-user@ip-172-31-20-228 ~]$ sudo pip3 install boto3
WARNING: Running pip install with root privileges is generally not a good idea. Try `pip3 install --user` instead.
Collecting boto3
  Downloading boto3-1.20.41-py3-none-any.whl (131 kB)
     |████████████████████████████████| 131 kB 19.5 MB/s
Collecting botocore<1.24.0,>=1.23.41
  Downloading botocore-1.23.41-py3-none-any.whl (8.5 MB)
     |████████████████████████████████| 8.5 MB 34.0 MB/s
Collecting s3transfer<0.6.0,>=0.5.0
  Downloading s3transfer-0.5.0-py3-none-any.whl (79 kB)
     |████████████████████████████████| 79 kB 13.7 MB/s
Collecting jmespath<1.0.0,>=0.7.1
  Downloading jmespath-0.10.0-py2.py3-none-any.whl (24 kB)
Collecting urllib3<1.27,>=1.25.4
  Downloading urllib3-1.26.8-py2.py3-none-any.whl (138 kB)
     |████████████████████████████████| 138 kB 49.2 MB/s
Collecting python-dateutil<3.0.0,>=2.1
  Downloading python_dateutil-2.8.2-py2.py3-none-any.whl (247 kB)
     |████████████████████████████████| 247 kB 40.1 MB/s
Collecting six>=1.5
  Downloading six-1.16.0-py2.py3-none-any.whl (11 kB)
Installing collected packages: urllib3, six, python-dateutil, jmespath, botocore, s3transfer, boto3
Successfully installed boto3-1.20.41 botocore-1.23.41 jmespath-0.10.0 python-dateutil-2.8.2 s3transfer-0.5.0 six-1.16.0 urllib3-1.26.8
[ec2-user@ip-172-31-20-228 ~]$ python3 -c "import boto3"
[ec2-user@ip-172-31-20-228 ~]$
```

--

- If there is no error, package is present. If there is any error for above command, revisit the installation of boto3 package using `pip`

Permissions to access the Resources in AWS Account

EC2 Instance Execution

- Go to `IAM > Create Role for EC2 service > Assign Service Specific Policies to this Role (e.g AmazonEC2FullAccess , AmazonS3FullAccess, AmazonRDSFullAccess)` > `Attach this Role to EC2 Instance`

--

Local Machine Execution

- Navigate to `IAM > Add User > Select Programmatic Access > Attach Service Specific Permissions to this User (e.g AmazonEC2FullAccess , AmazonS3FullAccess, AmazonRDSFullAccess)`
- Now that we have a user and credentials, we can finally configure the scripting environment with the AWS CLI tool
- Open any command line utility : For Windows : `Open Git Bash`

```
aws configure
```

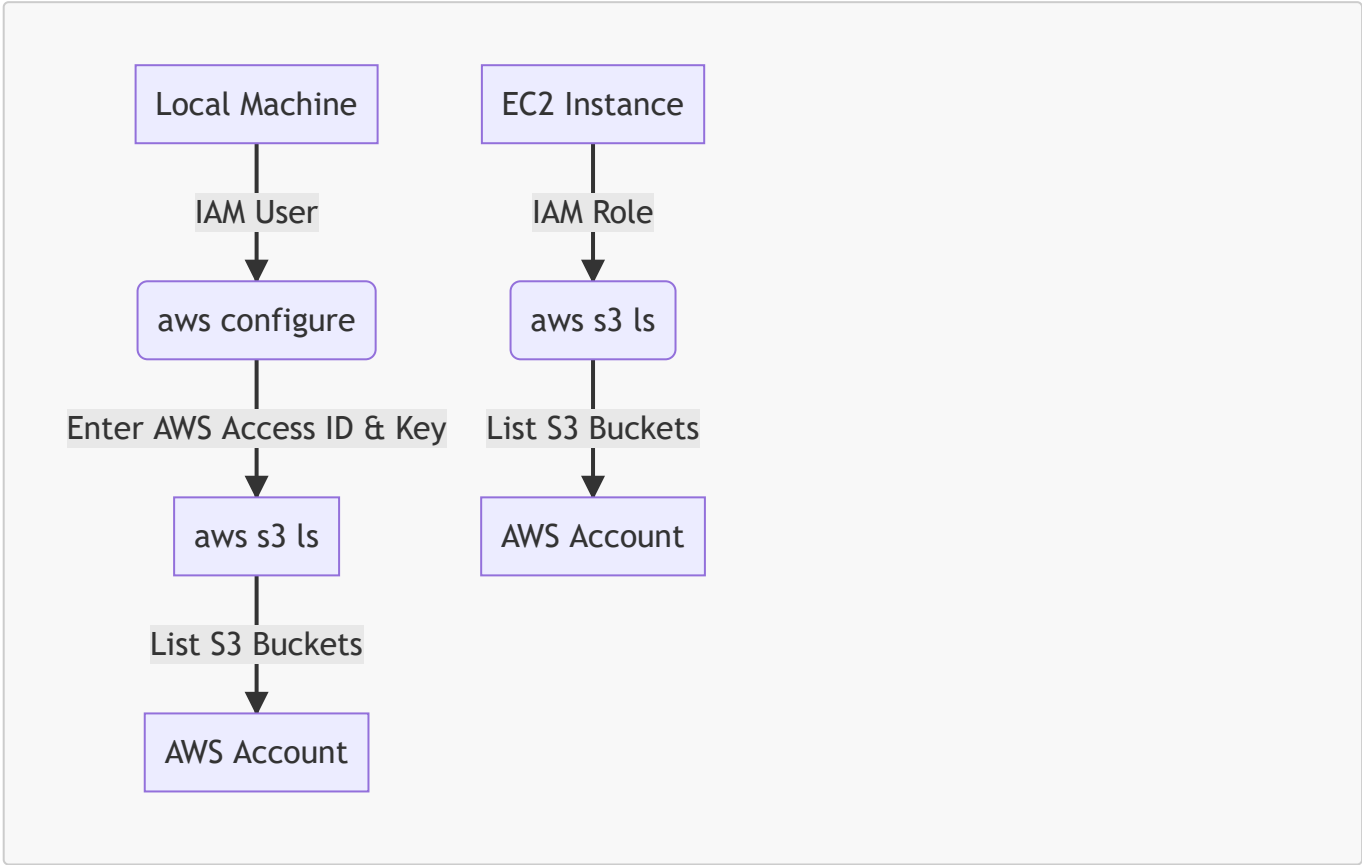
--

- Enter AWS access key ID, AWS secret access key for particular IAM User, default region name, and default output format.
- he region you enter will determine the location where any resources created by your script will be located.

- Now that our environment is all configured, lets test with aws cli tool.

```
$ aws ec2 describe-instances
```

--



- If you already have instances running, output of above command will be the details of those instances. If not, you should see an empty response.
- If there are any errors, validate previous steps particularly the access key ID/Secret access key and Policy attached to IAM User.

Python Code Execution with AWS Services

Python boto3 List EC2 Instances

- First, we'll import the `boto3` library. Using the library, we'll create a client object. This is like a handle to the EC2 console that we can use in our script to print the instance ID and state.

> Note : Save the below code as <filename.py> and execute using `python3 filename.py`

--

- List EC2 instances using python boto3

```
#!/usr/bin/env python
import boto3
#Create a client object connection with ec2 service
ec2 = boto3.client('ec2',region_name='us-west-2')

# Execute a function call to describe instances present in aws account
#
https://boto3.amazonaws.com/v1/documentation/api/1.9.42/reference/services/ec2.htm
l#EC2.Client.describe_instances
ec2_dict=ec2.describe_instances()
print("ec2_dict type is",type(ec2_dict))
print("ec2_dict is",ec2_dict)
```

--

- Get list of EC2 instances that are in 'stopped' state and start them.

```
import boto3
ec2 = boto3.client('ec2',region_name='us-west-2')

ec2_dict=ec2.describe_instances()
print("ec2_dict type is",type(ec2_dict))
print("ec2_dict is",ec2_dict)
reservations_list=ec2_dict['Reservations']
print(reservations_list)
print(type(reservations_list))
print(len(reservations_list))
# print("reservations_list is",reservations_list,
type(reservations_list),len(reservations_list))
print("----------------------------------------")
InstanceIdsList=[]
for instances in reservations_list:
    # print("instances",instances,type(instances))
    print("instance is of type",type(instances))
    instance_id=instances['Instances'][0]['InstanceId']
    instance_state=instances['Instances'][0]['State']['Name']
    print("instance_id is",instance_id)
    print("instance_state is",instance_state)
    if instance_state == 'stopped':
        print(instance_id ,"will be started")
        InstanceIdsList.append(instance_id)

# Check whether list is empty
if not InstanceIdsList:
    print("InstanceIdsList is empty, cannot perform start operation")
else:
    print("Starting all the instances with instance ids: ",InstanceIdsList)
    ec2.start_instances(InstanceIds=InstanceIdsList)
```

Python boto3 List S3 Buckets Information

```
import boto3
#Create a client object connection with ec2 service
s3 = boto3.client('s3')
# Execute a function call to get list of S3 buckets in aws account
bucket_dict=s3.list_buckets()
print("Type of bucket_dict is",type(bucket_dict))
bucket_list=bucket_dict['Buckets']
print("Type of bucket_list is",type(bucket_list))
print("Len of bucket_list is",len(bucket_list))
for bucket_info in bucket_list:
    print("Type of bucket_info is",type(bucket_info))
    print("Bucket Name is ",bucket_info['Name'])
```

💡 Tips 💡

- Do not remember every function/method names for each service in AWS, just search the documentation for the same.
- It is expected that you know how to use list/dictionary traversal in python.