

Installing boto-3:

```
PS C:\Users\srina> python --version
Python 3.11.4
PS C:\Users\srina> pip
```

Usage:

```
pip <command> [options]
```

Commands:

install	Install packages.
download	Download packages.
uninstall	Uninstall packages.
freeze	Output installed packages in requirements format.
inspect	Inspect the python environment.
list	List installed packages.
show	Show information about installed packages.
check	Verify installed packages have compatible dependencies.
config	Manage local and global configuration.
search	Search PyPI for packages.
cache	Inspect and manage pip's wheel cache.
index	Inspect information available from package indexes.
wheel	Build wheels from your requirements.
hash	Compute hashes of package archives.
completion	A helper command used for command completion.
debug	Show information useful for debugging.
help	Show help for commands.

D:\DE\DataEngineering\re
Notepad++

```
the spelling of the name, or if a path was included, verify that the path is correct and try again.
At line:2 char:1
+ sudo apt upgrade
+ ~~~~~
+ CategoryInfo          : ObjectNotFound: (sudo:String) [], CommandNotFoundException
+ FullyQualifiedErrorId : CommandNotFoundException

PS C:\Users\srina> pip install boto3
Collecting boto3
  Downloading boto3-1.26.157-py3-none-any.whl (135 kB)
----- 135.9/135.9 kB 2.7 MB/s eta 0:00:00
Collecting botocore<1.30.0,>=1.29.157 (from boto3)
  Downloading botocore-1.29.157-py3-none-any.whl (10.9 MB)
----- 10.9/10.9 MB 46.9 MB/s eta 0:00:00
Collecting jmespath<2.0.0,>=0.7.1 (from boto3)
  Downloading jmespath-1.0.1-py3-none-any.whl (20 kB)
Collecting s3transfer<0.7.0,>=0.6.0 (from boto3)
  Downloading s3transfer-0.6.1-py3-none-any.whl (79 kB)
----- 79.8/79.8 kB 4.3 MB/s eta 0:00:00
Collecting python-dateutil<3.0.0,>=2.1 (from botocore<1.30.0,>=1.29.157->boto3)
  Downloading python_dateutil-2.8.2-py2.py3-none-any.whl (247 kB)
----- 247.7/247.7 kB ? eta 0:00:00
Collecting urllib3<1.27,>=1.25.4 (from botocore<1.30.0,>=1.29.157->boto3)
  Downloading urllib3-1.26.16-py2.py3-none-any.whl (143 kB)
----- 143.1/143.1 kB ? eta 0:00:00
Collecting six>=1.5 (from python-dateutil<3.0.0,>=2.1->botocore<1.30.0,>=1.29.157->boto3)
  Downloading six-1.16.0-py2.py3-none-any.whl (11 kB)
Installing collected packages: urllib3, six, jmespath, python-dateutil, botocore, s3transfer, boto3
Successfully installed boto3-1.26.157 botocore-1.29.157 jmespath-1.0.1 python-dateutil-2.8.2 s3transfer-0.6.1 six-1.16.0
urllib3-1.26.16
```

Task – 5

Aws – boto3

Launching EC2 using python code :

```
D:\DE\DataEngineering\LaunchEC2.py - Notepad++
File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?
main.py test_buckets.py read & write from s3.py launchEC2.py
1 import boto3
2
3 # AWS credentials and EC2 instance details
4 aws_access_key_id = 'AKIA36KU44PYLR6ZF2PU'
5 aws_secret_access_key = 'N8sGwCnrUilcLJ89SfXrgMTIxxdrzj8YM2arAegF'
6 instance_type = 't2.micro'
7 ami_id = 'ami-0e820afa569e84cc1'
8 #security_group_ids = ['YOUR_SECURITY_GROUP_ID'] # List of security group IDs
9 key_pair_name = 'test'
10 #subnet_id = 'YOUR_SUBNET_ID'
11
12 # Create a session using your AWS credentials
13 session = boto3.Session(
14     aws_access_key_id=aws_access_key_id,
15     aws_secret_access_key=aws_secret_access_key
16 )
17
18 # Create an EC2 client using the session
19 ec2_client = session.client('ec2')
20
21 # Launch the EC2 instance
22 response = ec2_client.run_instances(
23     ImageId=ami_id,
24     InstanceType=instance_type,
25     MinCount=1,
26     MaxCount=1,
27     KeyName=key_pair_name,
28     # SecurityGroupIds=security_group_ids,
29     # SubnetId=subnet_id
30 )
31
32 # Get the instance ID of the launched instance
33 instance_id = response['Instances'][0]['InstanceId']
34
35 print(f"EC2 instance with ID '{instance_id}' launched.")
```

```
D:\DE\DataEngineering>python "read & write from s3.py"
CSV to JSON conversion completed. JSON file 'sample_json.json' saved in bucket 'mytaskbucket111'.

D:\DE\DataEngineering>python launchEC2.py
EC2 instance with ID 'i-0b34631ada9e9b11e' launched.

D:\DE\DataEngineering>
```

reading csv file from s3 bucket and converting to json and writing back to the same bucket in a new file:

Task – 5

Aws – boto3

```
1 import boto3
2 import csv
3 import json
4
5 # AWS credentials and bucket names
6 aws_access_key_id = 'AKIA36KU44PYLR6ZF2PU'
7 aws_secret_access_key = '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'
8 source_bucket_name = 'mytaskbucket111'
9 source_file_name = 'csvjson.csv'
10 destination_bucket_name = 'mytaskbucket111'
11 destination_file_name = 'sample_json.json'
12
13 # Create a session using your AWS credentials
14 session = boto3.Session(
15     aws_access_key_id=aws_access_key_id,
16     aws_secret_access_key=aws_secret_access_key
17 )
18
19 # Create an S3 client using the session
20 s3_client = session.client('s3')
21
22 # Read the CSV file from the source S3 bucket
23 response = s3_client.get_object(Bucket=source_bucket_name, Key=source_file_name)
24 csv_data = response['Body'].read().decode('windows-1252')
25
26 # Convert CSV to JSON
27 csv_rows = csv_data.split('\n')
28 fieldnames = csv_rows[0].split(',')
29 json_data = []
30 for row in csv_rows[1:]:
31     if row:
32         row_data = row.split(',')
33         json_data.append(dict(zip(fieldnames, row_data)))
34
35 # Write the JSON data to the destination S3 bucket
```

D:\DE\DataEngineering\read & write from s3.py - Notepad++

File Edit Search View Encoding Language Settings Tools Macro Run Plugins Window ?

main.tf list_buckets.py read & write from s3.py launchEC2.py

```
31 if row:
32     row_data = row.split(',')
33     json_data.append(dict(zip(fieldnames, row_data)))
34
35 # Write the JSON data to the destination S3 bucket
36 json_content = json.dumps(json_data)
37 s3_client.put_object(Body=json_content, Bucket=destination_bucket_name, Key=destination_file_name)
38
39 print(f"CSV to JSON conversion completed. JSON file '{destination_file_name}' saved in bucket '{destination_bucket_name}'.")
40
```

Files that are in the S3 bucket after running the python code :

Amazon S3

X

Buckets

Access Points

Object Lambda Access Points

Multi-Region Access Points

Batch Operations

IAM Access Analyzer for S3

Block Public Access settings for this account

▼ Storage Lens

Dashboards

AWS Organizations settings

Objects

Properties

Permissions

Metrics

Management

Access Points

Objects (2)

Objects are the fundamental entities stored in Amazon S3. You can use [Amazon S3 inventory](#) to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. [Learn more](#)

↻

Copy S3 URI

Copy URL

Download

Open

Delete

Actions ▼

Create folder

Upload

Find objects by prefix

< 1 > ⚙

<input type="checkbox"/>	Name ▲	Type ▼	Last modified ▼	Size ▼	Storage class ▼
<input type="checkbox"/>	csvjson.csv	csv	June 20, 2023, 19:56:31 (UTC-04:00)	269.0 B	Standard
<input type="checkbox"/>	sample_json.json	json	June 20, 2023, 20:04:57 (UTC-04:00)	714.0 B	Standard