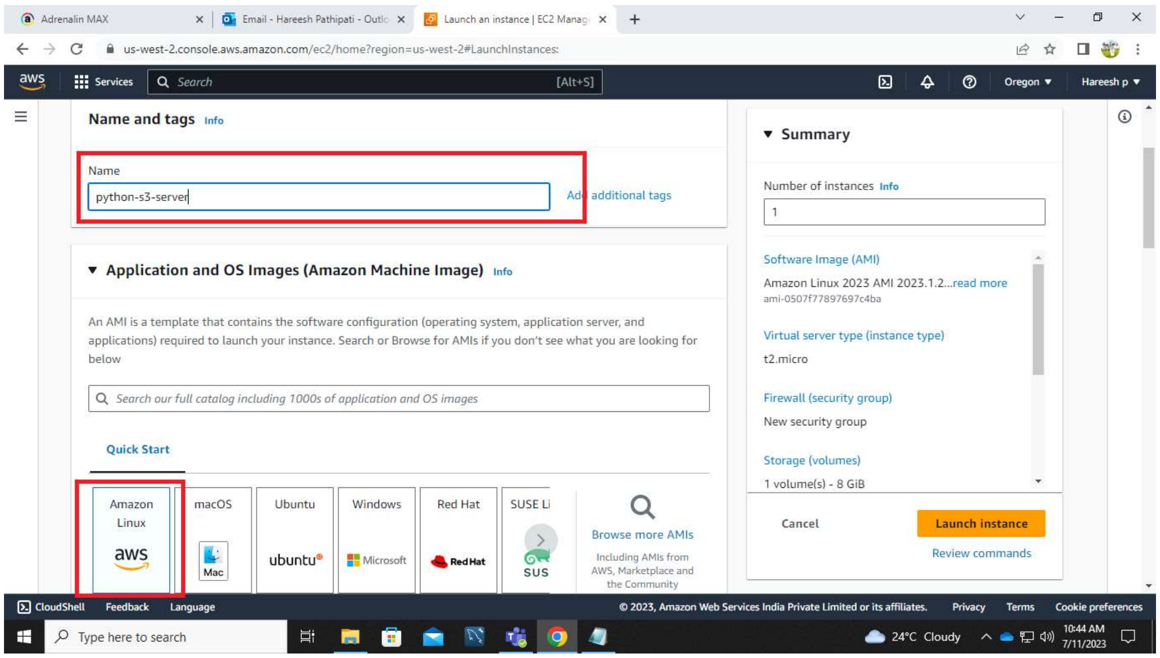
Amazon s3 examples using SDK for Python (Boto3)

Amazon Web Services (AWS) is the largest cloud computing platform. It offers over 200 resources that can help us with infrastructure to machine learning applications. Subsequently, Python is one of the most used programming languages right now. Therefore, AWS and Python are becoming one of the most demanding skills in the industry and combining AWS and Python is one of the best skills to acquire nowadays.

Python works well with AWS. There are readily available AWS Python Software Development Kits (SDKs), such as Boto3, that developers can use to interface with AWS services using Python programming language.

Python is widely used in web applications, software development and data science projects. This is mainly due to Python being an easy-to-understand language that can be used to code almost any application.

- **I am going to demonstrate how to use the Boto3 AWS SDK for Python to interact with Amazon S3 buckets and objects.**
- Login to AWS management console and Launch the instance
- Select the Amazon linux AMI



Name and tags

Name: python-s3-server

Application and OS Images (Amazon Machine Image)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Summary

Number of instances: 1

Software Image (AMI): Amazon Linux 2023 AMI 2023.1.2...read more ami-0507f77897697c4ba

Virtual server type (instance type): t2.micro

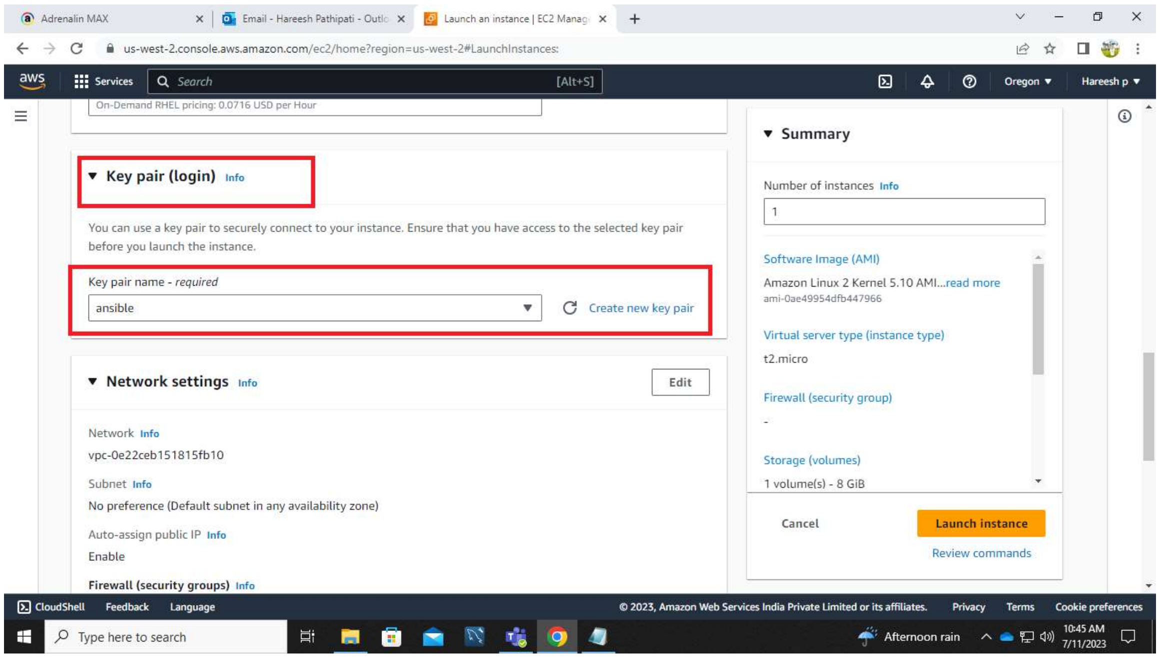
Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

- select the key pair ---A key pair, consisting of a public key and a private key, is a set of security credentials that you use to prove your identity when connecting to an Amazon

EC2.

- create a new key pair or existing old key pair



On-Demand RHEL pricing: 0.0716 USD per Hour

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - *required*: ansible

Network: vpc-0e22ceb151815fb10
Subnet: No preference (Default subnet in any availability zone)
Auto-assign public IP: Enable
Firewall (security groups)

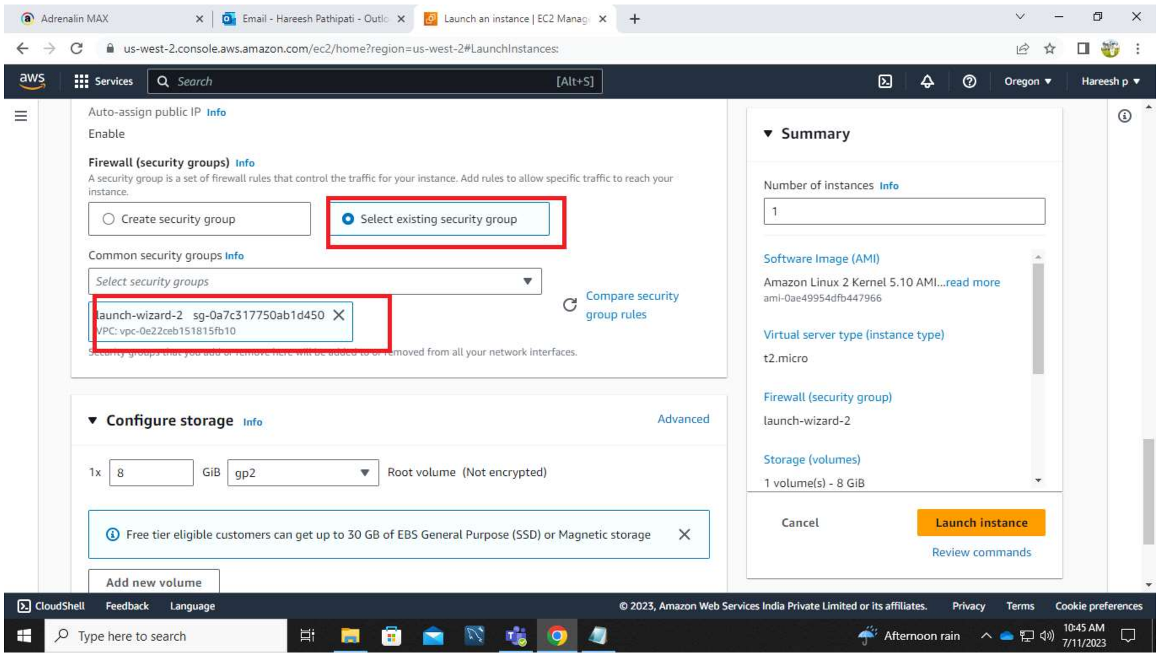
Number of instances: 1
Software Image (AMI): Amazon Linux 2 Kernel 5.10 AMI...read more ami-0ae49954dfb447966
Virtual server type (instance type): t2.micro
Firewall (security group): -
Storage (volumes): 1 volume(s) - 8 GiB

- select the security group --AWS Security Groups help you secure your cloud environment by controlling how traffic will be allowed into your EC2 machines. With Security Groups, you can ensure that all the traffic that flows at the instance level is only through your established ports and protocols.

- select the security group either existing security group or create new security group.

- select the number of instances

- Finally click on launch instance


Auto-assign public IP Info
Enable

Firewall (security groups) Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group [✓ Select existing security group]

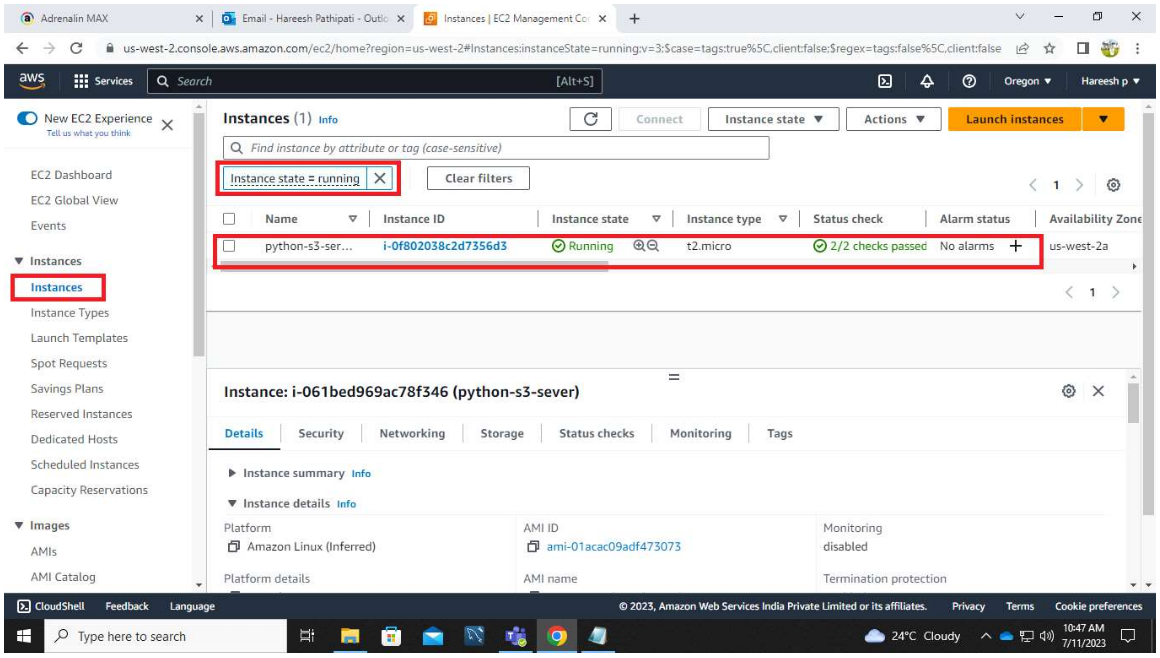
Common security groups Info
Select security groups
launch-wizard-2 sg-0a7c317750ab1d450
VPC: vpc-0e22ceb151815fb10

Configure storage Info
1x 8 GiB gp2 Root volume (Not encrypted)

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage

Summary
Number of instances: 1
Software Image (AMI): Amazon Linux 2 Kernel 5.10 AMI...read more ami-0ae49954dfb447966
Virtual server type (instance type): t2.micro
Firewall (security group): launch-wizard-2
Storage (volumes): 1 volume(s) - 8 GiB

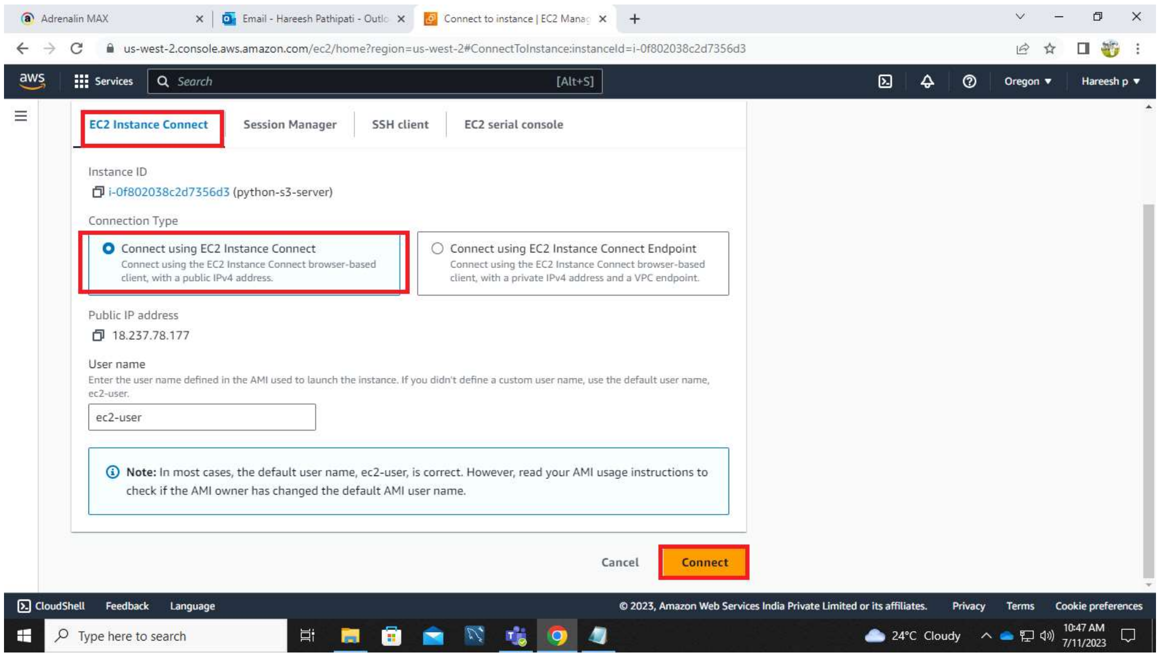
- Navigate to ec2 dashboard and verify the instance created or not


Instances (1) Info
Instance state = running

Name	Instance ID	Instance state	Instance type	Status check	Alarm status	Availability Zone
python-s3-ser...	i-0f802038c2d7356d3	Running	t2.micro	2/2 checks passed	No alarms	us-west-2a

Instance: i-061bed969ac78f346 (python-s3-sever)
Platform: Amazon Linux (Inferred)
AMI ID: ami-01acac09adf473073
Monitoring: disabled

- connect the instance using EC2 instance connect
- click on connect it will navigate into linux terminal



Instance ID
i-0f802038c2d7356d3 (python-s3-server)

Connection Type
Connect using EC2 Instance Connect
Connect using the EC2 Instance Connect browser-based client, with a public IPv4 address.
Connect using EC2 Instance Connect Endpoint
Connect using the EC2 Instance Connect browser-based client, with a private IPv4 address and a VPC endpoint.

Public IP address
18.237.78.177

User name
Enter the user name defined in the AMI used to launch the instance. If you didn't define a custom user name, use the default user name, ec2-user.
ec2-user

Note: In most cases, the default user name, ec2-user, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.

- when we connect to terminal Default we are ec2-user to become a root user we need to execute bellow command then only we will get root privileges.

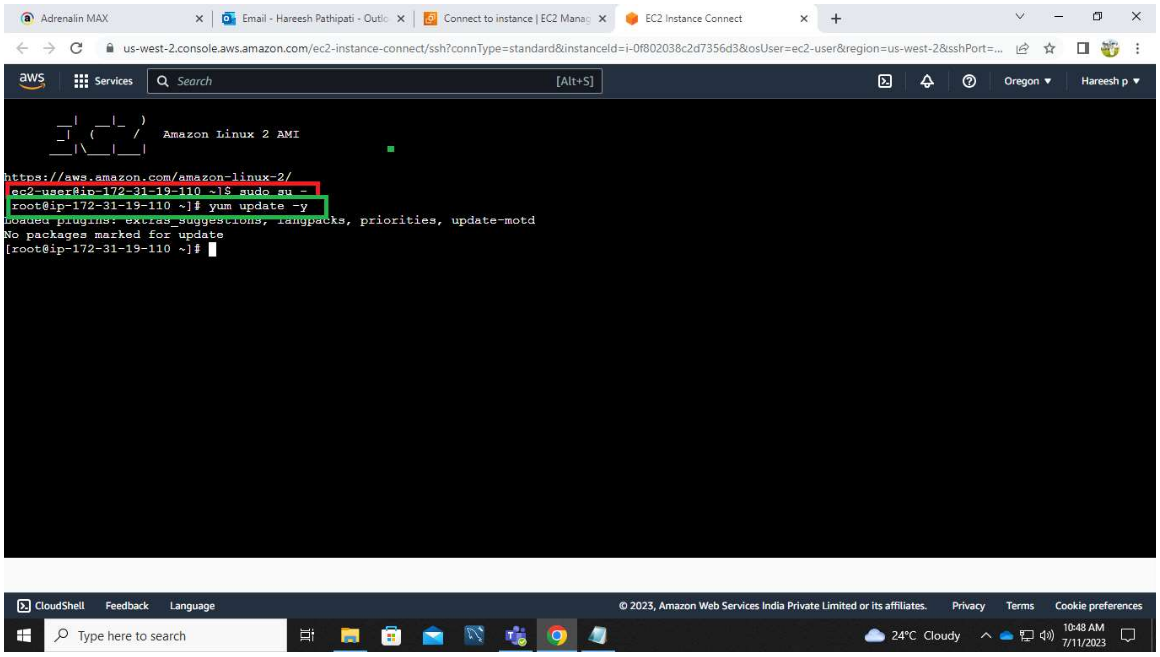
CMD --

```
sudo su -
```

CMD --

```
- yum update -y
```

- Start with running the system update command before installing anything on your Amazon Linux. For that just use the YUM command .It will not only install the latest available security update but also rebuild the package manager’s cache.



```
   __|  __|_  )
   _|  (     /   Amazon Linux 2 AMI
  ___|\___|___|

https://aws.amazon.com/amazon-linux-2/
[ec2-user@ip-172-31-19-110 ~]$ sudo su -
[root@ip-172-31-19-110 ~]# yum update -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
No packages marked for update
[root@ip-172-31-19-110 ~]#
```

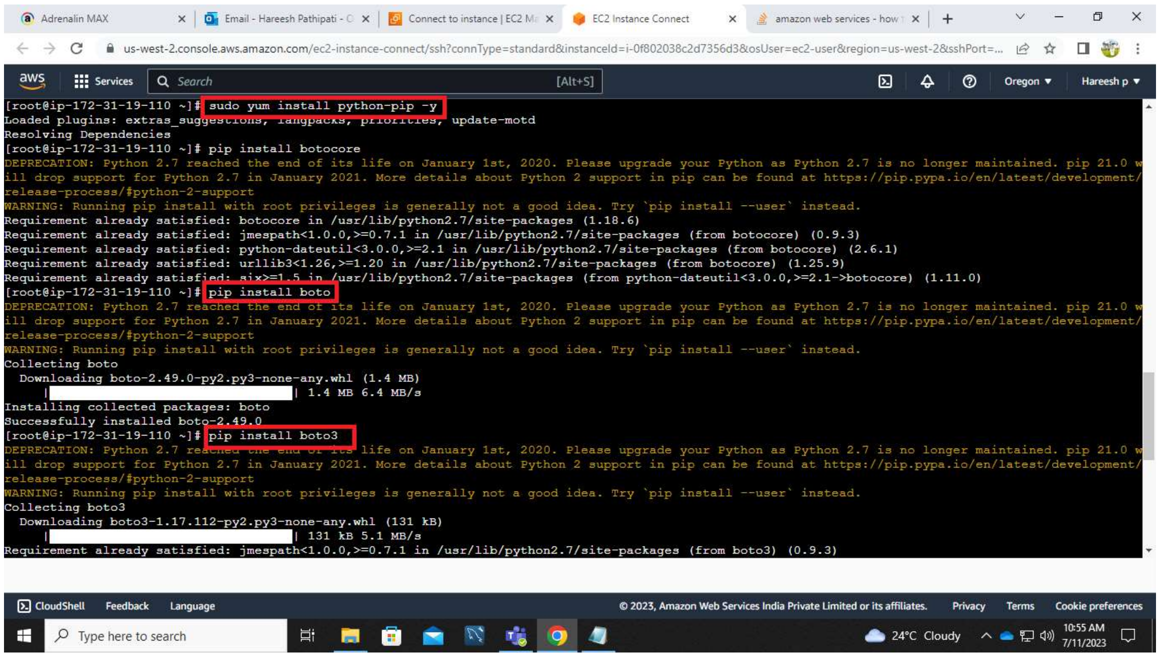
- **Installing Python 3 & PIP on Amazon Linux:**
- AWS Botocore is a foundation-level library that delivers the pieces for developers to build tools and applications for operating with AWS services.
- Boto3 is the AWS SDK for Python. It is simple to use and allows Python developers to write software applications that use and interface with AWS services.
- PIP is the package manager for Python.so to get the Boto3 we simply need to use PIP.

CMD --

sudo yum install python-pip -Y

CMD--

yum install python3-pip



```
[root@ip-172-31-19-110 ~]# sudo yum install python-pip -y
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
Resolving Dependencies
[root@ip-172-31-19-110 ~]# pip install botocore
DEPRECATION: Python 2.7 reached the end of its life on January 1st, 2020. Please upgrade your Python as Python 2.7 is no longer maintained. pip 21.0 w
ill drop support for Python 2.7 in January 2021. More details about Python 2 support in pip can be found at https://pip.pypa.io/en/latest/development/
release-process/#python-2-support
WARNING: Running pip install with root privileges is generally not a good idea. Try `pip install --user` instead.
Requirement already satisfied: botocore in /usr/lib/python2.7/site-packages (1.18.6)
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /usr/lib/python2.7/site-packages (from botocore) (0.9.3)
Requirement already satisfied: python-dateutil<3.0.0,>=2.1 in /usr/lib/python2.7/site-packages (from botocore) (2.6.1)
Requirement already satisfied: urllib3<1.26,>=1.20 in /usr/lib/python2.7/site-packages (from botocore) (1.25.9)
Requirement already satisfied: six>=1.5 in /usr/lib/python2.7/site-packages (from python-dateutil<3.0.0,>=2.1->botocore) (1.11.0)
[root@ip-172-31-19-110 ~]# pip install boto
DEPRECATION: Python 2.7 reached the end of its life on January 1st, 2020. Please upgrade your Python as Python 2.7 is no longer maintained. pip 21.0 w
ill drop support for Python 2.7 in January 2021. More details about Python 2 support in pip can be found at https://pip.pypa.io/en/latest/development/
release-process/#python-2-support
WARNING: Running pip install with root privileges is generally not a good idea. Try `pip install --user` instead.
Collecting boto
  Downloading boto-2.49.0-py2.py3-none-any.whl (1.4 MB)
     |████████████████████████████████| 1.4 MB 6.4 MB/s
Installing collected packages: boto
Successfully installed boto-2.49.0
[root@ip-172-31-19-110 ~]# pip install boto3
DEPRECATION: Python 2.7 reached the end of its life on January 1st, 2020. Please upgrade your Python as Python 2.7 is no longer maintained. pip 21.0 w
ill drop support for Python 2.7 in January 2021. More details about Python 2 support in pip can be found at https://pip.pypa.io/en/latest/development/
release-process/#python-2-support
WARNING: Running pip install with root privileges is generally not a good idea. Try `pip install --user` instead.
Collecting boto3
  Downloading boto3-1.17.112-py2.py3-none-any.whl (131 kB)
     |████████████████████████████████| 131 kB 5.1 MB/s
Requirement already satisfied: jmespath<1.0.0,>=0.7.1 in /usr/lib/python2.7/site-packages (from boto3) (0.9.3)
```

- However, if you don't want to install the library globally but instead in an isolated environment just for the application you are developing then create a Python3 environment.
- Python virtual environments give you the ability to isolate your Python development projects from your system installed Python and other Python environments.

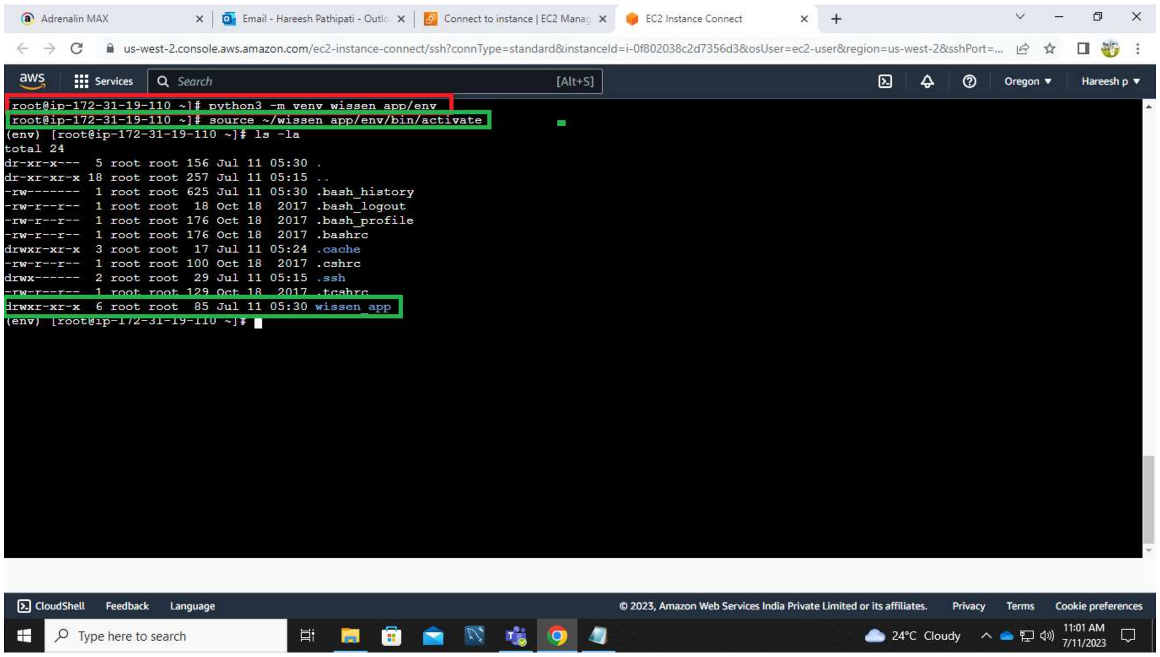
CMD--

python3 -m venv wissen_app/env

- wissen_app is the directory for which we are creating an environment.
- To activate the created environment for your current bash session, use:

CMD--

source ~/wissen_app/env/bin/activate

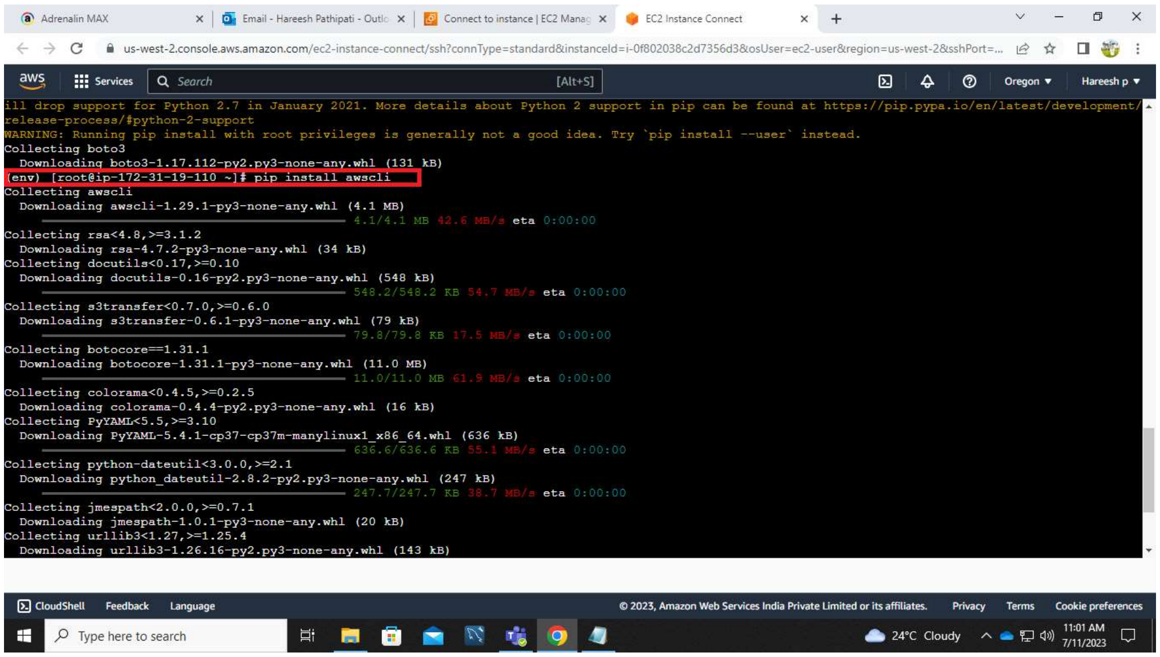


```
[root@ip-172-31-19-110 ~]# python3 -m venv wissen_app/env
[root@ip-172-31-19-110 ~]# source ~/wissen_app/env/bin/activate
(env) [root@ip-172-31-19-110 ~]# ls -la
total 24
dr-xr-x---  5 root root 156 Jul 11 05:30 .
dr-xr-xr-x 18 root root 257 Jul 11 05:15 ..
-rw-------  1 root root 625 Jul 11 05:30 .bash_history
-rw-r--r--  1 root root  18 Oct 18  2017 .bash_logout
-rw-r--r--  1 root root 176 Oct 18  2017 .bash_profile
-rw-r--r--  1 root root 176 Oct 18  2017 .bashrc
drwxr-xr-x  3 root root  17 Jul 11 05:24 .cache
-rw-r--r--  1 root root 100 Oct 18  2017 .cshrc
drwx------  2 root root  29 Jul 11 05:15 .ssh
-rw-r--r--  1 root root 129 Oct 18  2017 .tcshrc
drwxr-xr-x  6 root root  85 Jul 11 05:30 wissen_app
(env) [root@ip-172-31-19-110 ~]#
```

- Install awscli
- The AWS Command Line Interface (AWS CLI) is a unified tool to manage your AWS services. With just one tool to download and configure, you can control multiple AWS services from the command line and automate them through scripts.

CMD--

pip install awscli



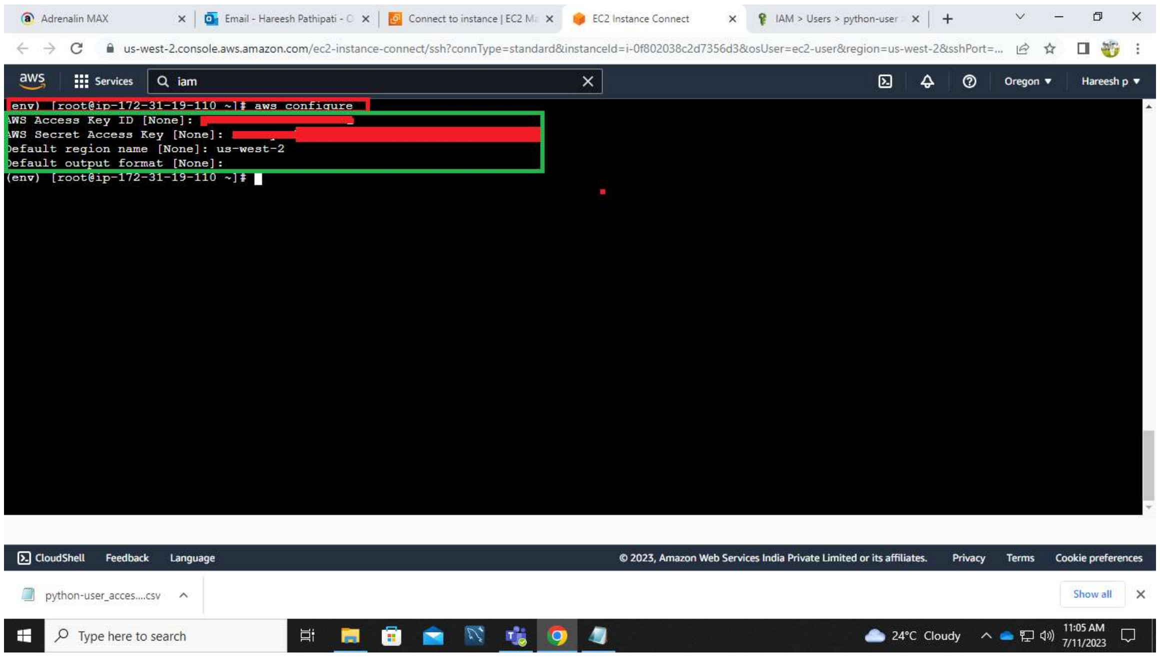
```
ill drop support for Python 2.7 in January 2021. More details about Python 2 support in pip can be found at https://pip.pypa.io/en/latest/development/
release-process/#python-2-support
WARNING: Running pip install with root privileges is generally not a good idea. Try `pip install --user` instead.
Collecting boto3
  Downloading boto3-1.17.112-py2.py3-none-any.whl (131 kB)
(env) [root@ip-172-31-19-110 ~]# pip install awscli
Collecting awscli
  Downloading awscli-1.29.1-py3-none-any.whl (4.1 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 4.1/4.1 MB 42.6 MB/s eta 0:00:00
Collecting rsa<4.8,>=3.1.2
  Downloading rsa-4.7.2-py3-none-any.whl (34 kB)
Collecting docutils<0.17,>=0.10
  Downloading docutils-0.16-py2.py3-none-any.whl (548 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 548.2/548.2 KB 54.7 MB/s eta 0:00:00
Collecting s3transfer<0.7.0,>=0.6.0
  Downloading s3transfer-0.6.1-py3-none-any.whl (79 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 79.8/79.8 KB 17.5 MB/s eta 0:00:00
Collecting botocore==1.31.1
  Downloading botocore-1.31.1-py3-none-any.whl (11.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 11.0/11.0 MB 61.9 MB/s eta 0:00:00
Collecting colorama<0.4.5,>=0.2.5
  Downloading colorama-0.4.4-py2.py3-none-any.whl (16 kB)
Collecting PyYAML<5.5,>=3.10
  Downloading PyYAML-5.4.1-cp37-cp37m-manylinux1_x86_64.whl (636 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 636.6/636.6 KB 55.1 MB/s eta 0:00:00
Collecting python-dateutil<3.0.0,>=2.1
  Downloading python_dateutil-2.8.2-py2.py3-none-any.whl (247 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 247.7/247.7 KB 38.7 MB/s eta 0:00:00
Collecting jmespath<2.0.0,>=0.7.1
  Downloading jmespath-1.0.1-py3-none-any.whl (20 kB)
Collecting urllib3<1.27,>=1.25.4
  Downloading urllib3-1.26.16-py2.py3-none-any.whl (143 kB)
```

- Here we need to interact with AWS CLI for that we use below command
- To configure the credentials, use the command aws configure and include the credentials of the user created in the previous

CMD--

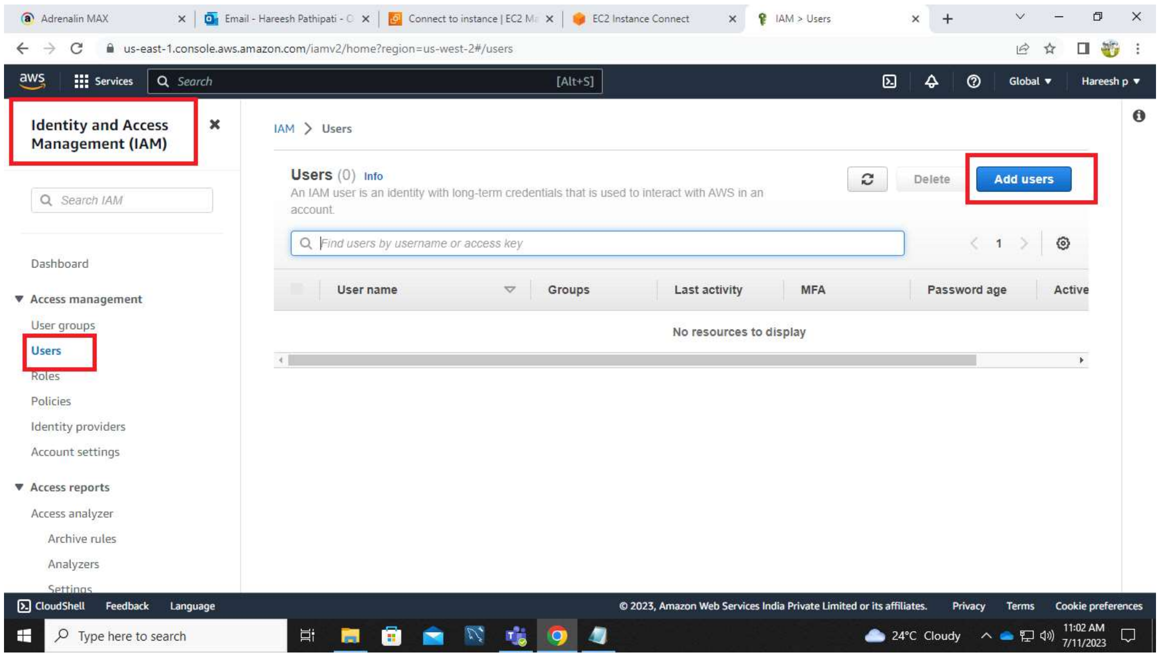
aws configure

- The credentials and config file are updated when you run the command aws configure.



```
env) [root@ip-172-31-19-110 ~]# aws configure
AWS Access Key ID [None]: [redacted]
AWS Secret Access Key [None]: [redacted]
Default region name [None]: us-west-2
Default output format [None]:
(env) [root@ip-172-31-19-110 ~]#
```

- To access AWS, you will need to sign up for an AWS account. Access keys consist of an access key ID and secret access key, which are used to sign programmatic requests that you make to AWS. If you don't have access keys, you can create them by using the IAM console by following steps.
- Navigate to IAM dash board and select the users
- select the Add users and add the user name
- click on next



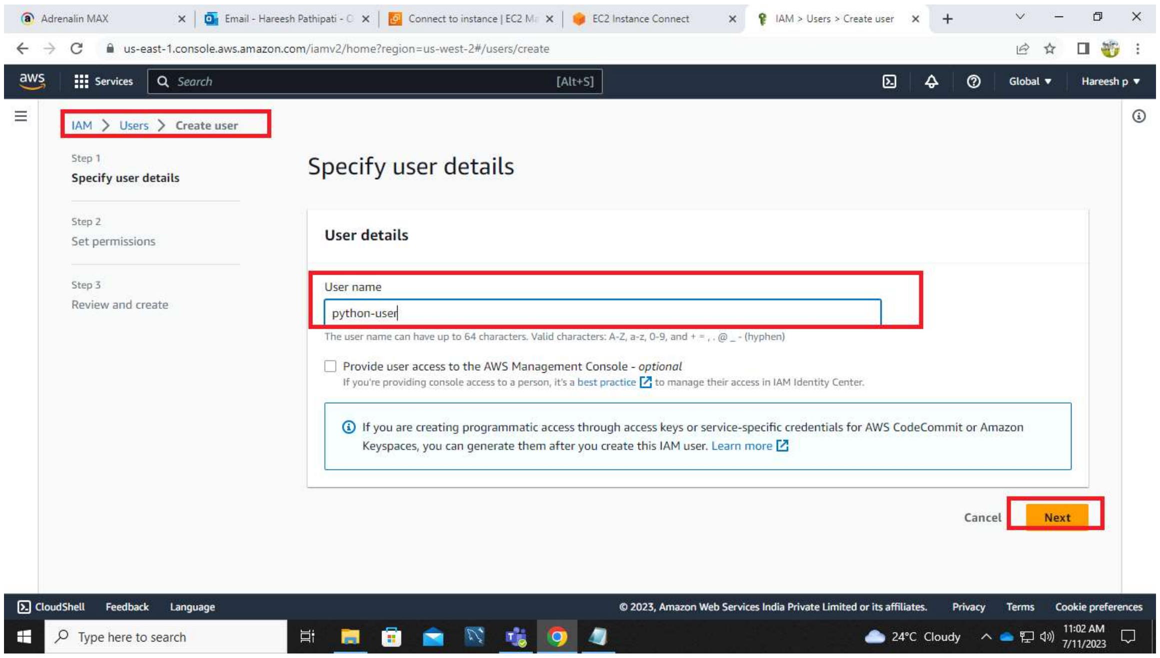
us-east-1.console.aws.amazon.com/iamv2/home?region=us-west-2#/users

Users (0) Info

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

User name	Groups	Last activity	MFA	Password age	Active

No resources to display



us-east-1.console.aws.amazon.com/iamv2/home?region=us-west-2#/users/create

Specify user details

User details

User name: python-user

The user name can have up to 64 characters. Valid characters: A-Z, a-z, 0-9, and + = , . @ _ - (hyphen)

☐ Provide user access to the AWS Management Console - *optional*
If you're providing console access to a person, it's a best practice to manage their access in IAM Identity Center.

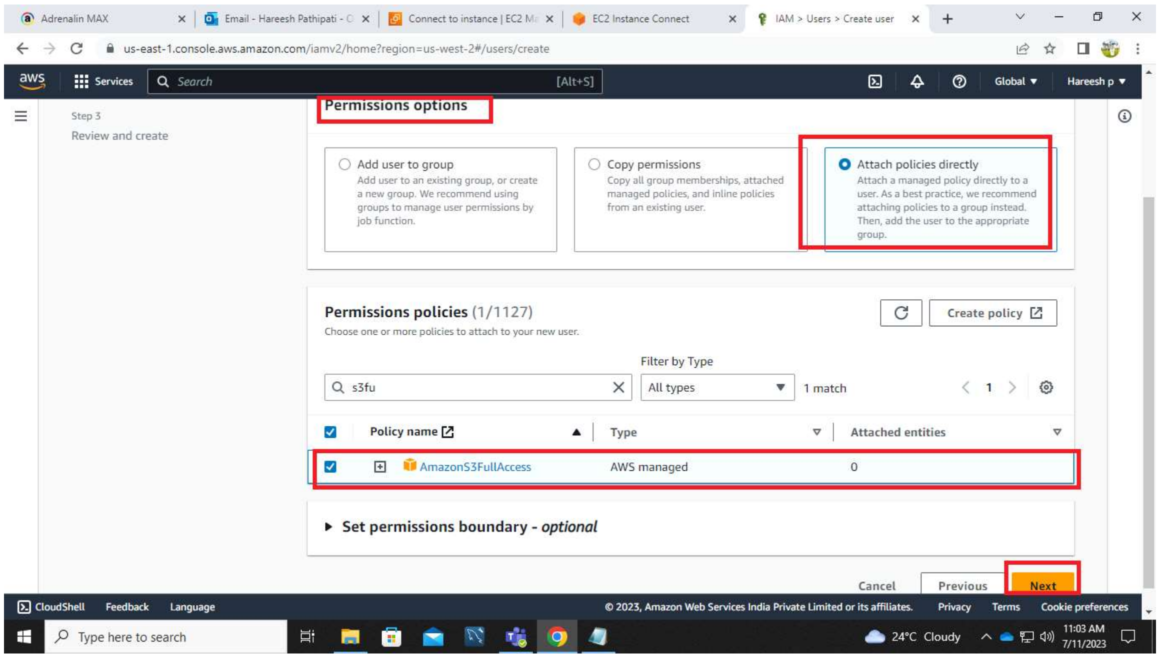
If you are creating programmatic access through access keys or service-specific credentials for AWS CodeCommit or Amazon Keyspaces, you can generate them after you create this IAM user. Learn more

- select the permissions to user
- Here we selected Attach policies directly
- Here we selected Amazons3fullAccess

NOTE-- we can attach all types of policies but our requirement here is s3 bucket creation.

thats why here we added Amazons3fullAccess.

- click on create user



Step 3
Review and create

Permissions options

- ○ Add user to group
  Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.
- ○ Copy permissions
  Copy all group memberships, attached managed policies, and inline policies from an existing user.
- ● Attach policies directly
  Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.

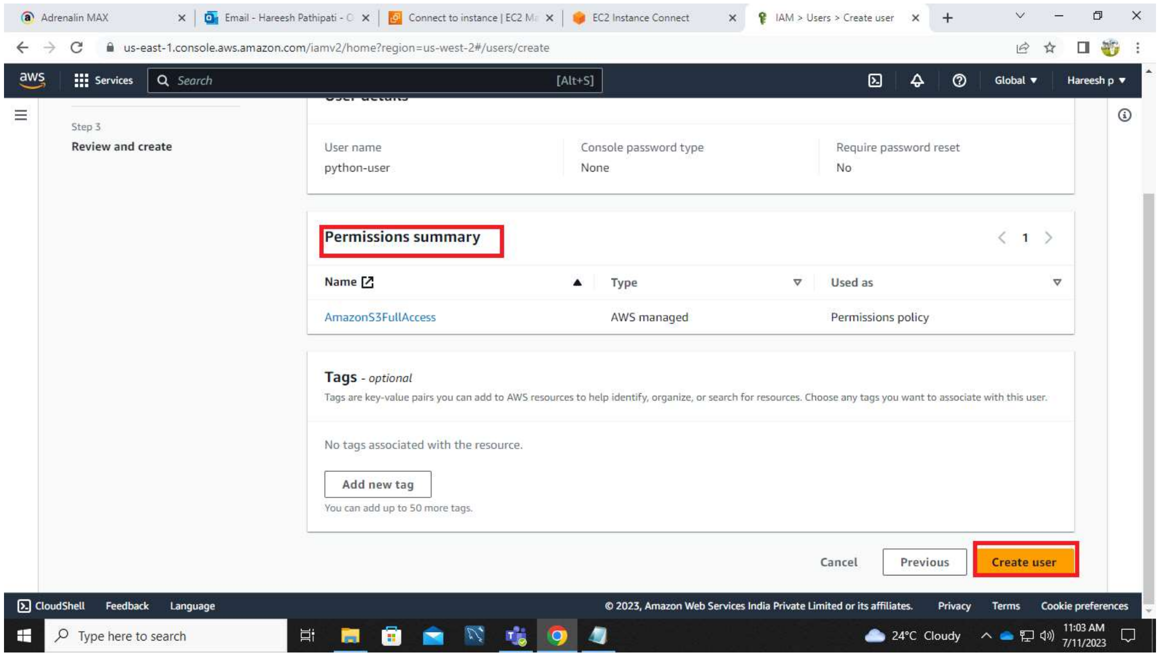
Permissions policies (1/1127)
Choose one or more policies to attach to your new user.

Search: s3fu — Filter by Type: All types — 1 match

	Policy name	Type	Attached entities
✓	AmazonS3FullAccess	AWS managed	0

▶ Set permissions boundary - *optional*

Cancel | Previous | Next



Step 3
Review and create

User name	Console password type	Require password reset
python-user	None	No

Permissions summary

Name	Type	Used as
AmazonS3FullAccess	AWS managed	Permissions policy

Tags - *optional*
Tags are key-value pairs you can add to AWS resources to help identify, organize, or search for resources. Choose any tags you want to associate with this user.

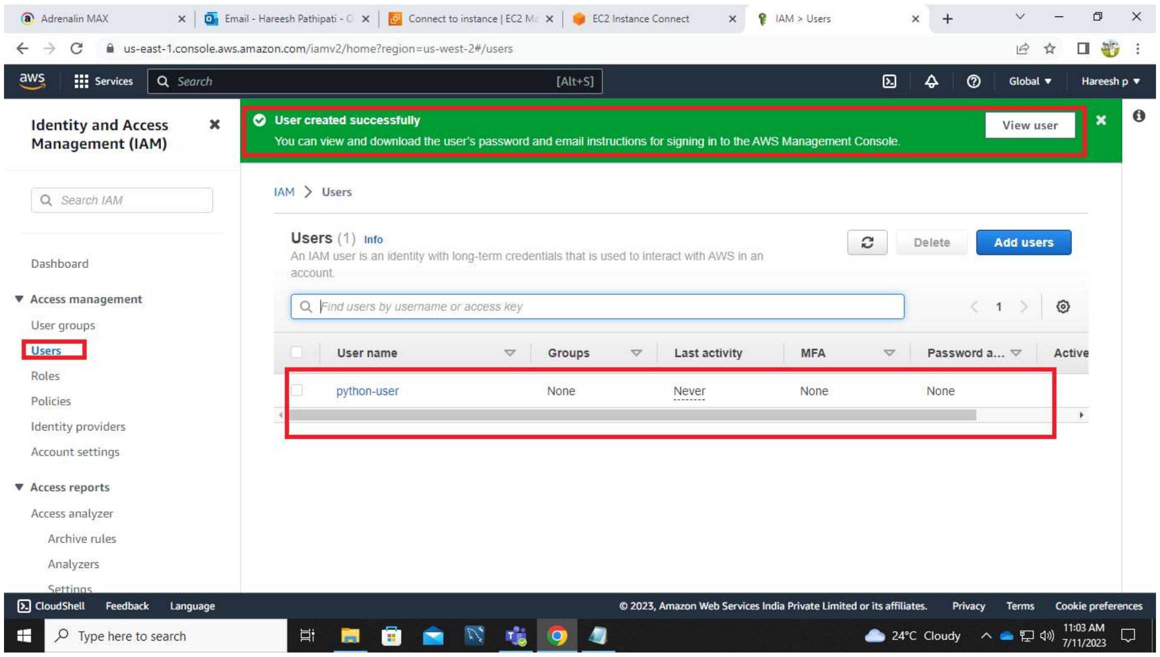
No tags associated with the resource.

Add new tag

You can add up to 50 more tags.

Cancel | Previous | Create user

- user created successfully

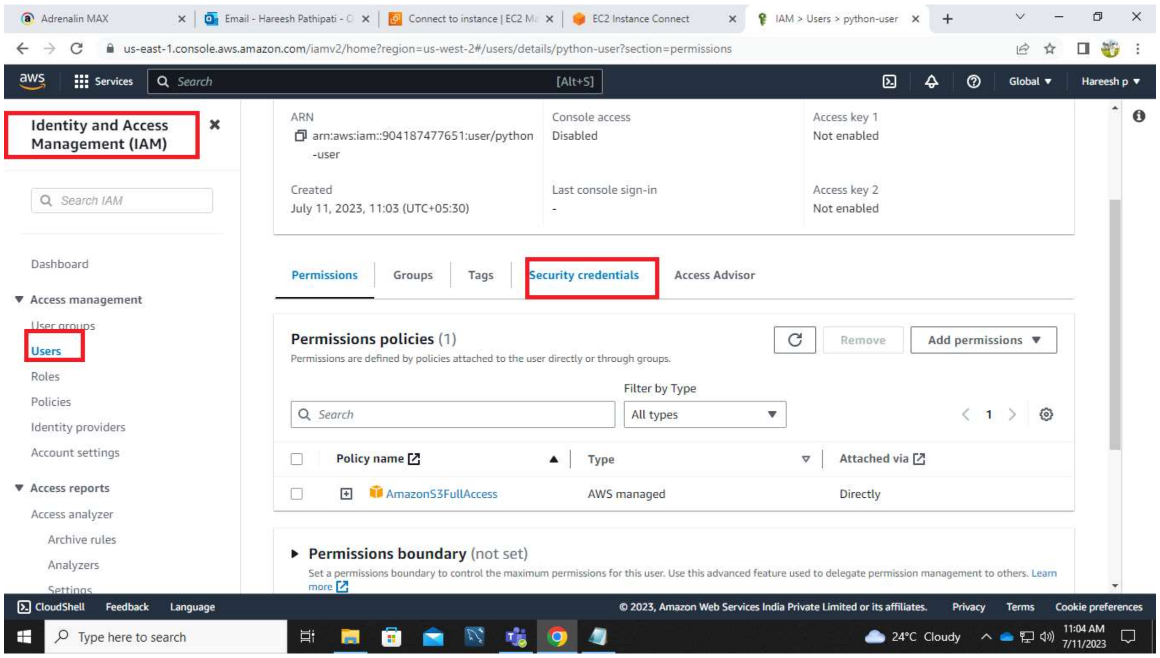


User created successfully
You can view and download the user's password and email instructions for signing in to the AWS Management Console.

Users (1) Info
An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

User name	Groups	Last activity	MFA	Password a...	Active
python-user	None	Never	None	None	

- we need to create access key and secret access key
- Select the security credentials option



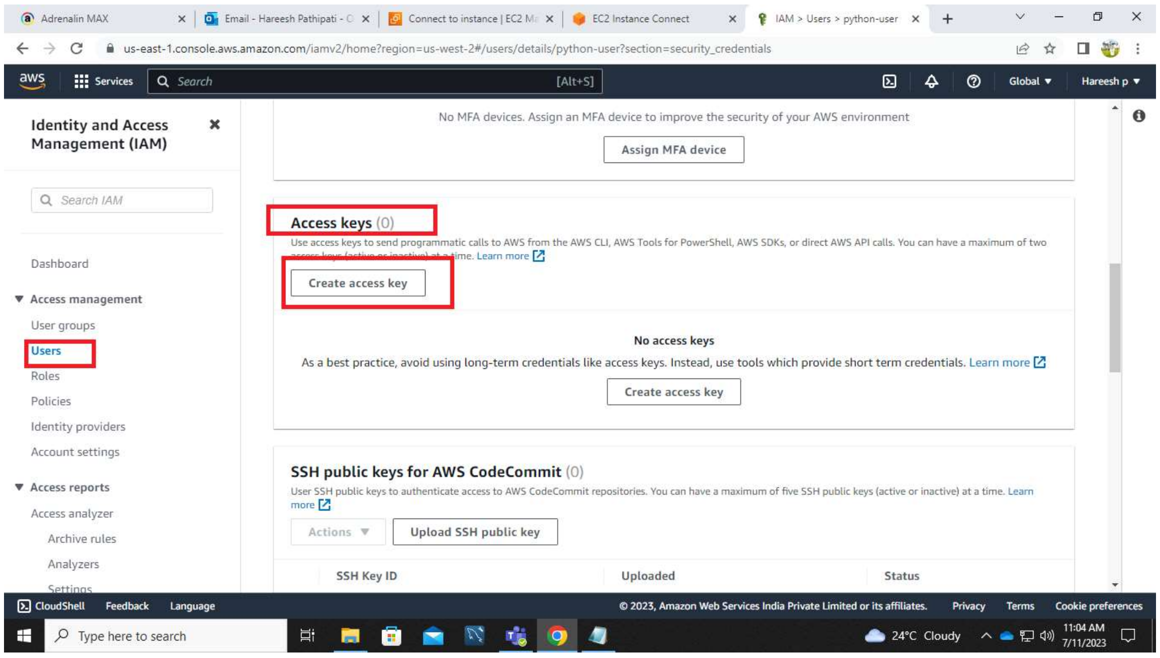
ARN: arn:aws:iam::904187477651:user/python-user
Console access: Disabled
Access key 1: Not enabled
Created: July 11, 2023, 11:03 (UTC+05:30)
Last console sign-in: -
Access key 2: Not enabled

Permissions policies (1)
Permissions are defined by policies attached to the user directly or through groups.
Filter by Type: All types

Policy name	Type	Attached via
AmazonS3FullAccess	AWS managed	Directly

Permissions boundary (not set)
Set a permissions boundary to control the maximum permissions for this user. Use this advanced feature used to delegate permission management to others. Learn more

- click on Accesskeys
- create Access key


No MFA devices. Assign an MFA device to improve the security of your AWS environment

Access keys (0)

Use access keys to send programmatic calls to AWS from the AWS CLI, AWS Tools for PowerShell, AWS SDKs, or direct AWS API calls. You can have a maximum of two access keys (active or inactive) at a time. Learn more

No access keys

As a best practice, avoid using long-term credentials like access keys. Instead, use tools which provide short term credentials. Learn more

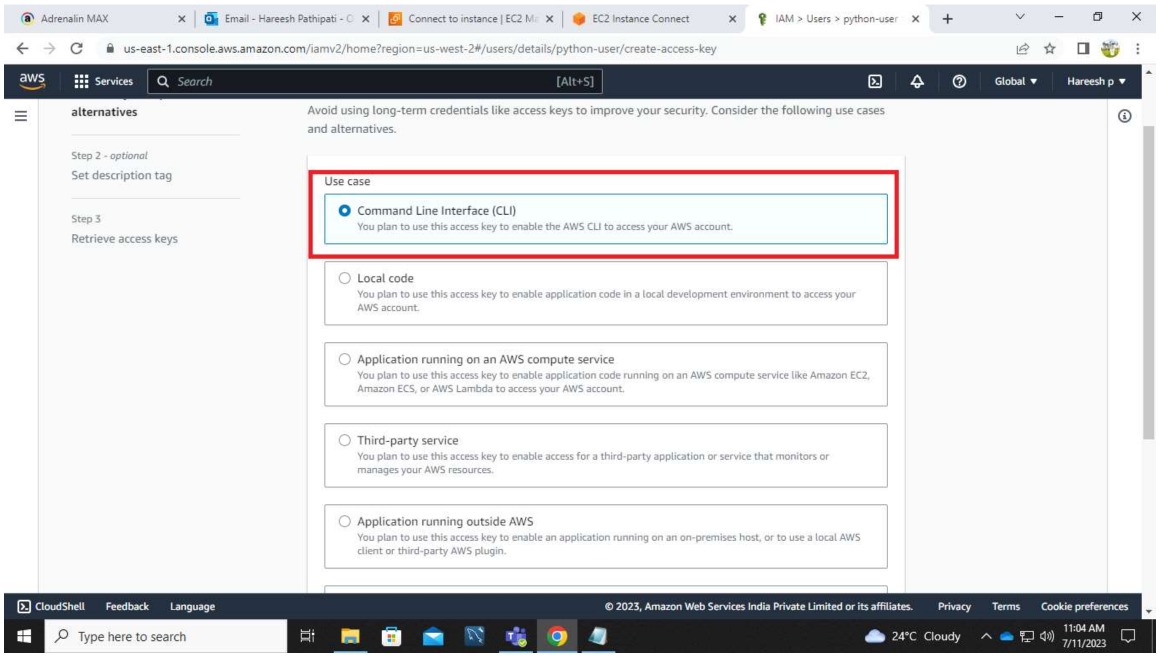
SSH public keys for AWS CodeCommit (0)

User SSH public keys to authenticate access to AWS CodeCommit repositories. You can have a maximum of five SSH public keys (active or inactive) at a time. Learn more

SSH Key ID	Uploaded	Status

© 2023, Amazon Web Services India Private Limited or its affiliates.

- Select the use
- Here we need CLI so click on command Line Interface

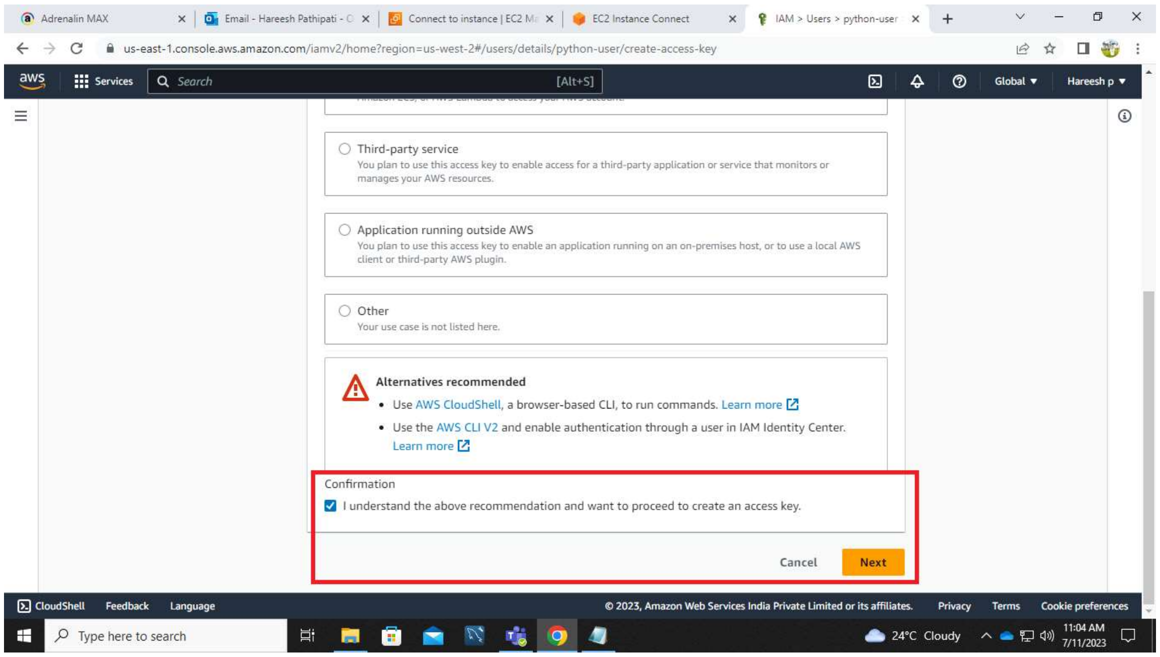

Avoid using long-term credentials like access keys to improve your security. Consider the following use cases and alternatives.

Use case

- [✓ Command Line Interface (CLI)] You plan to use this access key to enable the AWS CLI to access your AWS account.
- Local code — You plan to use this access key to enable application code in a local development environment to access your AWS account.
- Application running on an AWS compute service — You plan to use this access key to enable application code running on an AWS compute service like Amazon EC2, Amazon ECS, or AWS Lambda to access your AWS account.
- Third-party service — You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.
- Application running outside AWS — You plan to use this access key to enable an application running on an on-premises host, or to use a local AWS client or third-party AWS plugin.

© 2023, Amazon Web Services India Private Limited or its affiliates.

- For conformation tick the box and click on next


Third-party service
You plan to use this access key to enable access for a third-party application or service that monitors or manages your AWS resources.

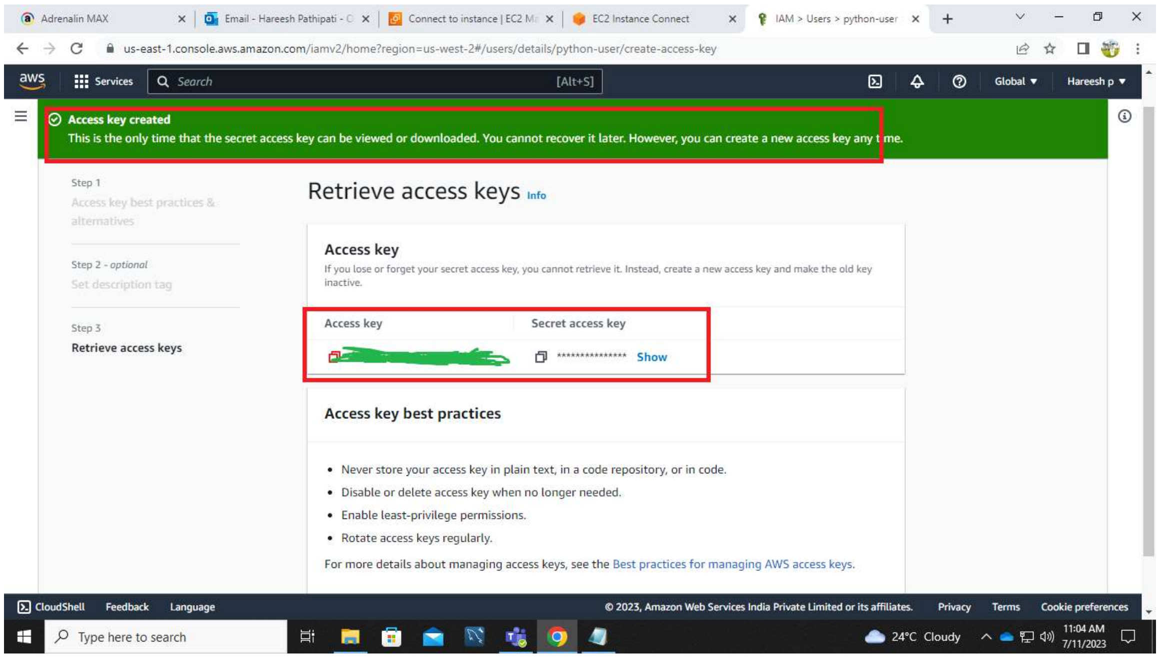
Application running outside AWS
You plan to use this access key to enable an application running on an on-premises host, or to use a local AWS client or third-party AWS plugin.

Other
Your use case is not listed here.

Alternatives recommended
- Use AWS CloudShell, a browser-based CLI, to run commands. Learn more
- Use the AWS CLI V2 and enable authentication through a user in IAM Identity Center. Learn more

Confirmation
☑ I understand the above recommendation and want to proceed to create an access key.

- so finally Access key created


Access key created
This is the only time that the secret access key can be viewed or downloaded. You cannot recover it later. However, you can create a new access key any time.

Retrieve access keys

Access key
If you lose or forget your secret access key, you cannot retrieve it. Instead, create a new access key and make the old key inactive.

Access key	Secret access key
[redacted]	*************** Show

Access key best practices
- Never store your access key in plain text, in a code repository, or in code.
- Disable or delete access key when no longer needed.
- Enable least-privilege permissions.
- Rotate access keys regularly.

For more details about managing access keys, see the Best practices for managing AWS access keys.

- If you have the AWS CLI installed, then you can use the aws configure command to configure your credentials file:

CMD--

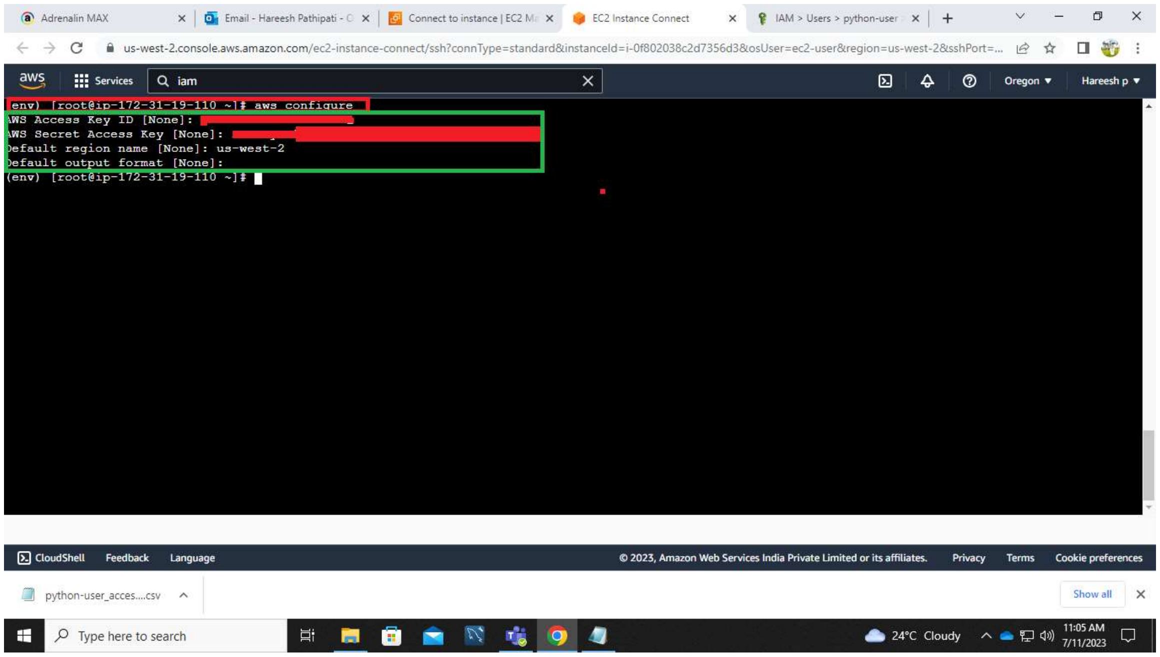
aws configure

- Now, switch to the Python command line and import the Boto3 library to start developing your application

CMD--

python

- You may also want to add a default region to the AWS configuration file, which is located by default



```
env) [root@ip-172-31-19-110 ~]# aws configure
AWS Access Key ID [None]: [redacted]
AWS Secret Access Key [None]: [redacted]
Default region name [None]: us-west-2
Default output format [None]:
(env) [root@ip-172-31-19-110 ~]#
```

Adrenalin MAX
Email - Hareesh Pathipati
Connect to instance | EC2
EC2 Instance Connect
IAM > Users > python-user
us-west-2.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0f802038c2d7356d3&osUser=ec2-user&region=us-west-2&sshPort=...
Services
iam
Oregon
Hareesh p
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences
python-user_acces....csv
Show all
Type here to search
24°C Cloudy
11:05 AM
7/11/2023

- **I am going to demonstrate how to use the Boto3 AWS SDK for Python to interact with Amazon S3 buckets and objects.**

- import keyword - to make code in one module available in another

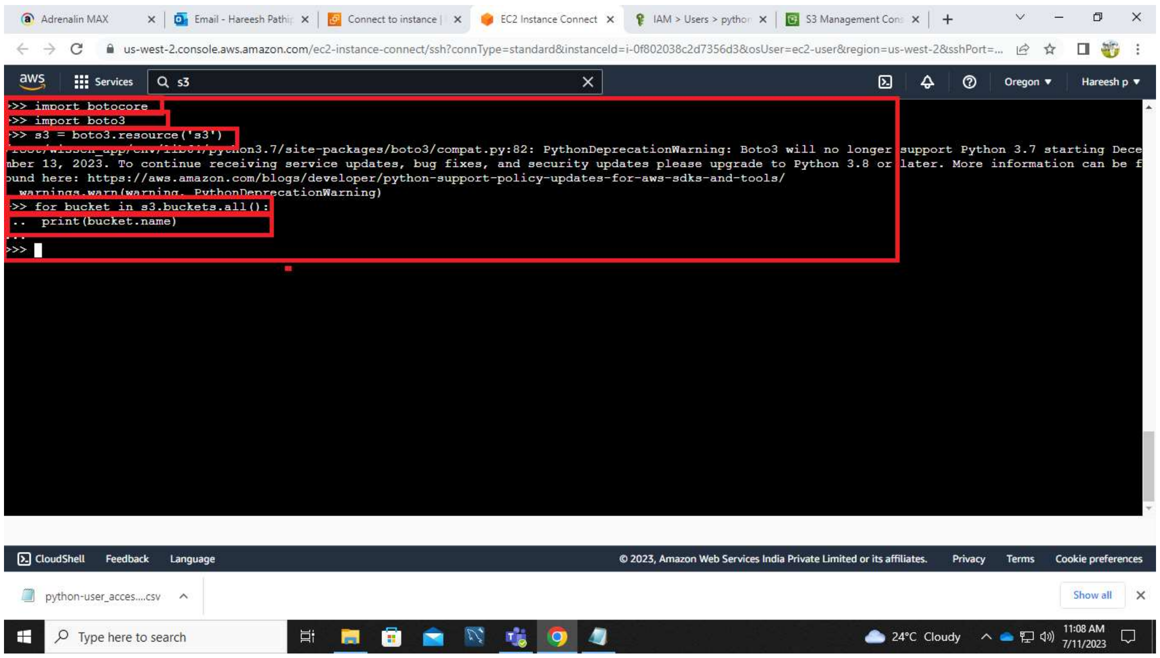
- To use Boto3, you must first import it and indicate which service or services you're going to use:

```
s3 = boto3.resource('s3')
```

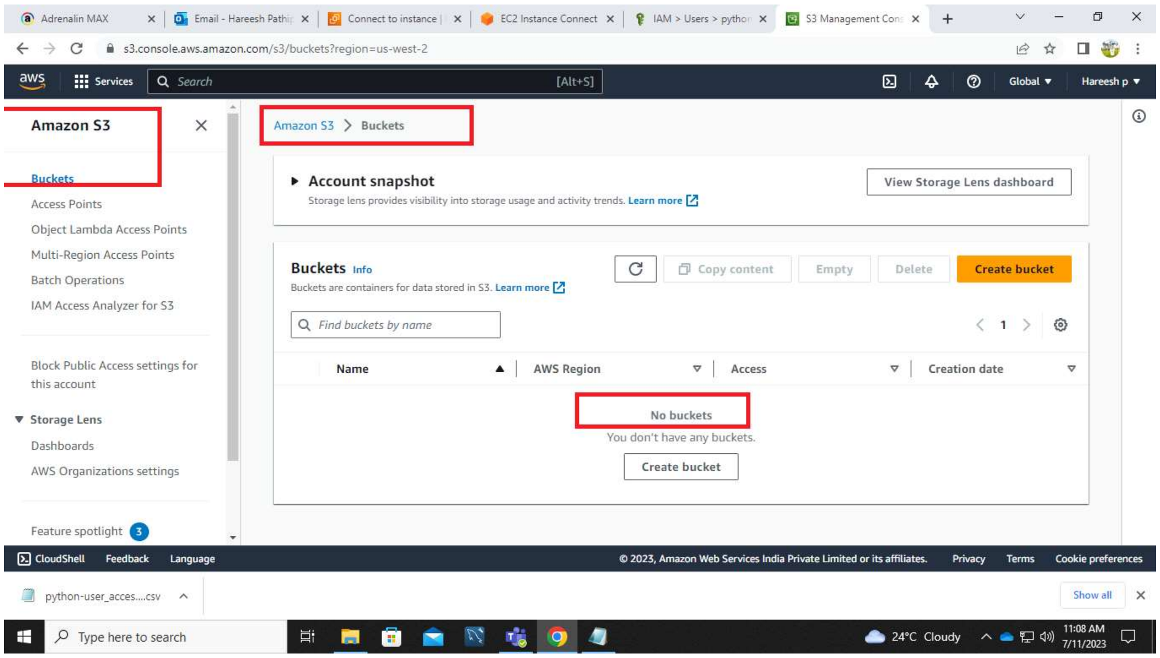
- Now that you have an s3 resource, you can make send requests to the service. The following code uses the buckets collection to print out all bucket names:

```
# Print out bucket names

for bucket in s3.buckets.all():

    print(bucket.name)
```

- Here we can see no buckets are created by seeing in CLI and AWS s3 dash board



```
>>> import botocore
>>> import boto3
>>> s3 = boto3.resource('s3')
/root/wissen_app/env/lib64/python3.7/site-packages/boto3/compat.py:82: PythonDeprecationWarning: Boto3 will no longer support Python 3.7 starting Dece
mber 13, 2023. To continue receiving service updates, bug fixes, and security updates please upgrade to Python 3.8 or later. More information can be f
ound here: https://aws.amazon.com/blogs/developer/python-support-policy-updates-for-aws-sdks-and-tools/
  warnings.warn(warning, PythonDeprecationWarning)
>>> for bucket in s3.buckets.all():
...   print(bucket.name)
...
>>> 
```



Amazon S3 > Buckets

Account snapshot
Storage lens provides visibility into storage usage and activity trends. Learn more

Buckets Info
Buckets are containers for data stored in S3. Learn more

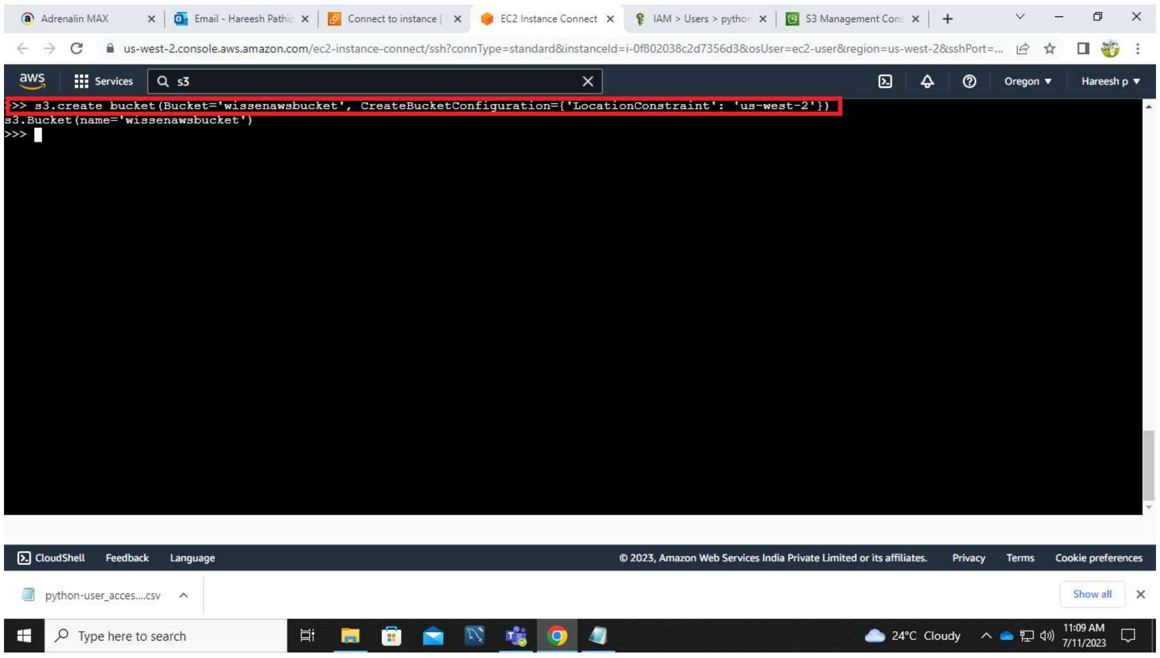
Name	AWS Region	Access	Creation date

No buckets
You don't have any buckets.

Amazon S3
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
Feature spotlight
View Storage Lens dashboard
Copy content
Empty
Delete
Create bucket
Find buckets by name
Create bucket

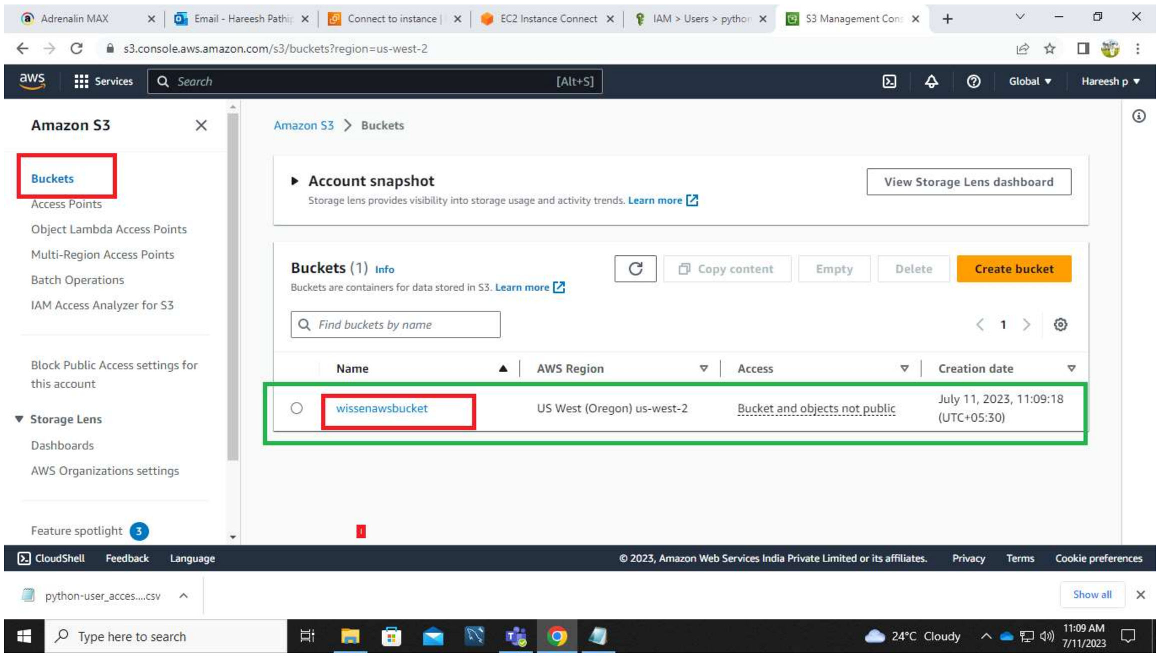
- **The following example creates a bucket. The request specifies an AWS region where to create the bucket.**

s3.create_bucket(Bucket='wissenawsbucket', CreateBucketConfiguration={'LocationConstraint': 'us-west-2'})



```
>>> s3.create_bucket(Bucket='wissenawsbucket', CreateBucketConfiguration={'LocationConstraint': 'us-west-2'})
s3.Bucket(name='wissenawsbucket')
>>>
```

- Navigate to s3 dash board and verify the bucket creation


Amazon S3
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
Feature spotlight
Amazon S3 > Buckets
Account snapshot
Storage lens provides visibility into storage usage and activity trends. Learn more
View Storage Lens dashboard
Buckets (1) Info
Buckets are containers for data stored in S3. Learn more
Copy content
Empty
Delete
Create bucket
Find buckets by name

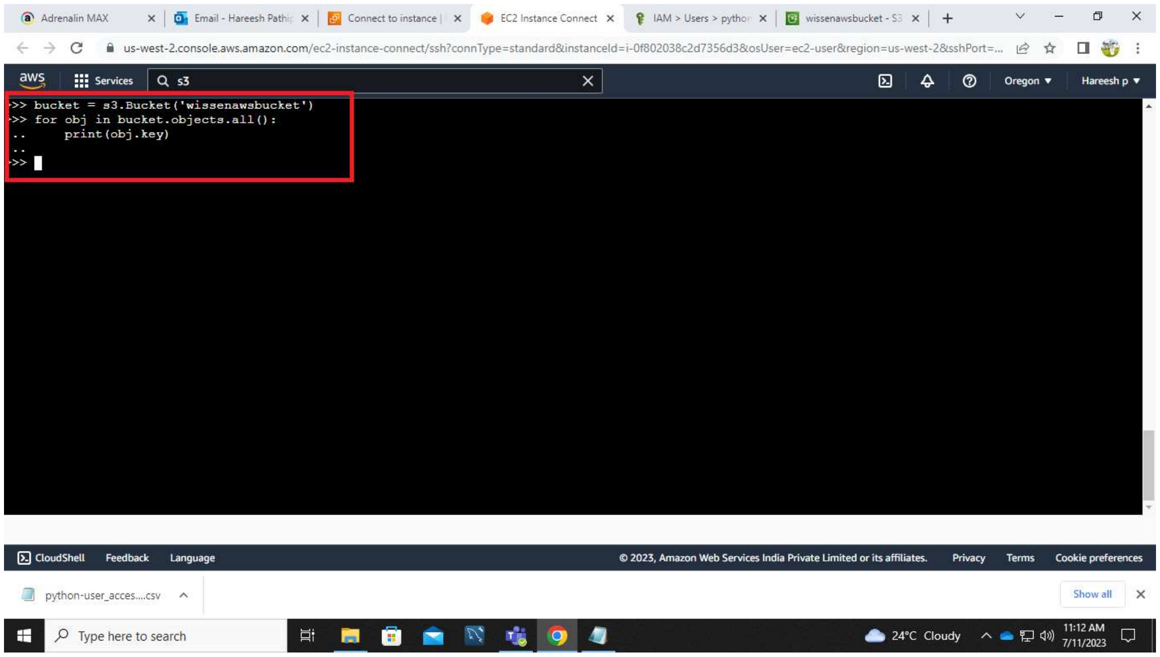
Name	AWS Region	Access	Creation date
wissenawsbucket	US West (Oregon) us-west-2	Bucket and objects not public	July 11, 2023, 11:09:18 (UTC+05:30)

- **The following example gives how many objects availabe inside the buckets**

bucket = s3.Bucket('wissenawsbucket')

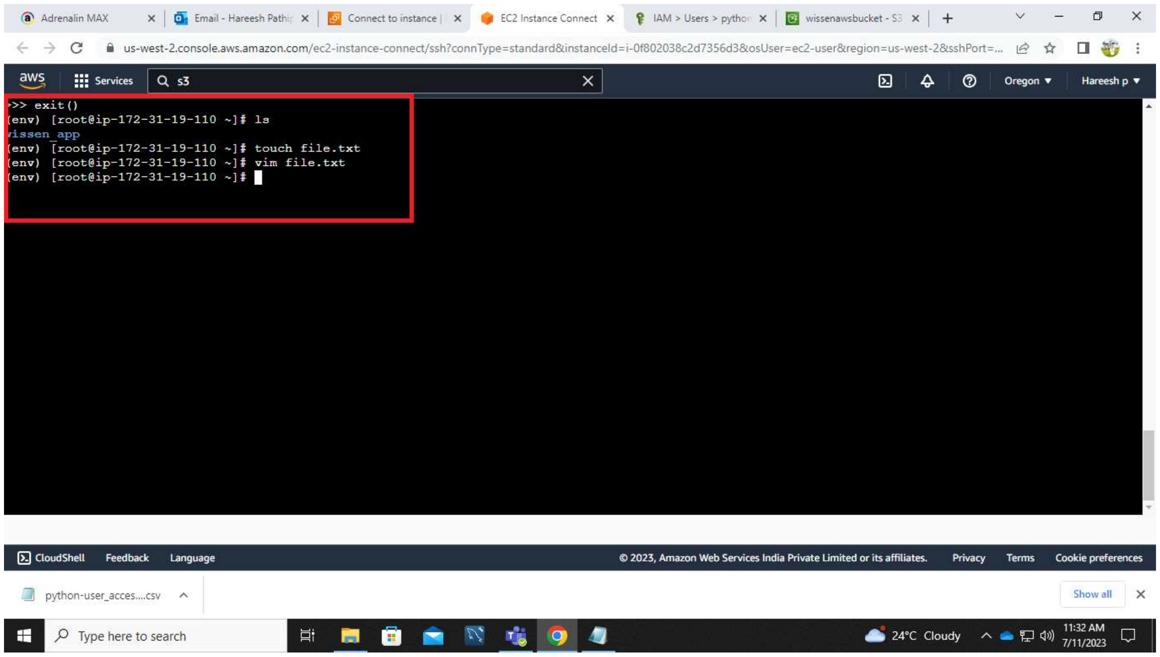
for obj in bucket.objects.all():

print(obj.key)



```
>>> bucket = s3.Bucket('wissenawsbucket')
>>> for obj in bucket.objects.all():
...     print(obj.key)
...
>>> 
```

- **To inserting file into s3 bucket through sdk**
- First we need to create file in local



```
>>> exit()
(env) [root@ip-172-31-19-110 ~]# ls
wissen_app
(env) [root@ip-172-31-19-110 ~]# touch file.txt
(env) [root@ip-172-31-19-110 ~]# vim file.txt
(env) [root@ip-172-31-19-110 ~]# 
```

- **inserting file into s3 bucket through sdk** import boto3

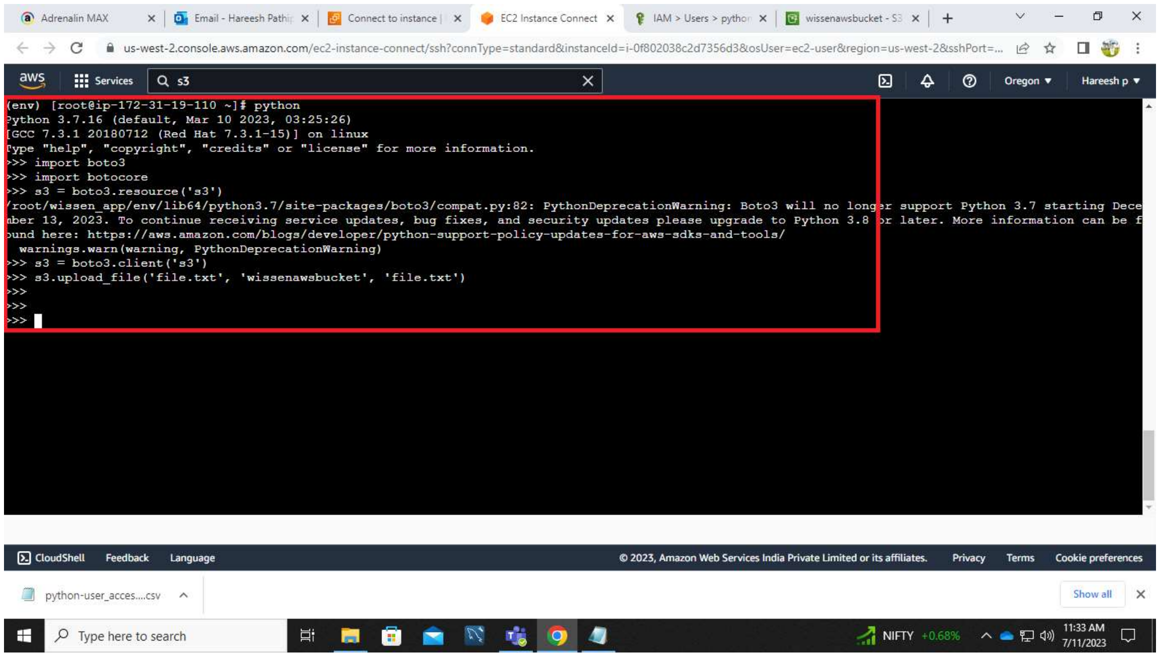
import botocore

s3 = boto3.resource('s3')

s3 = boto3.client('s3')

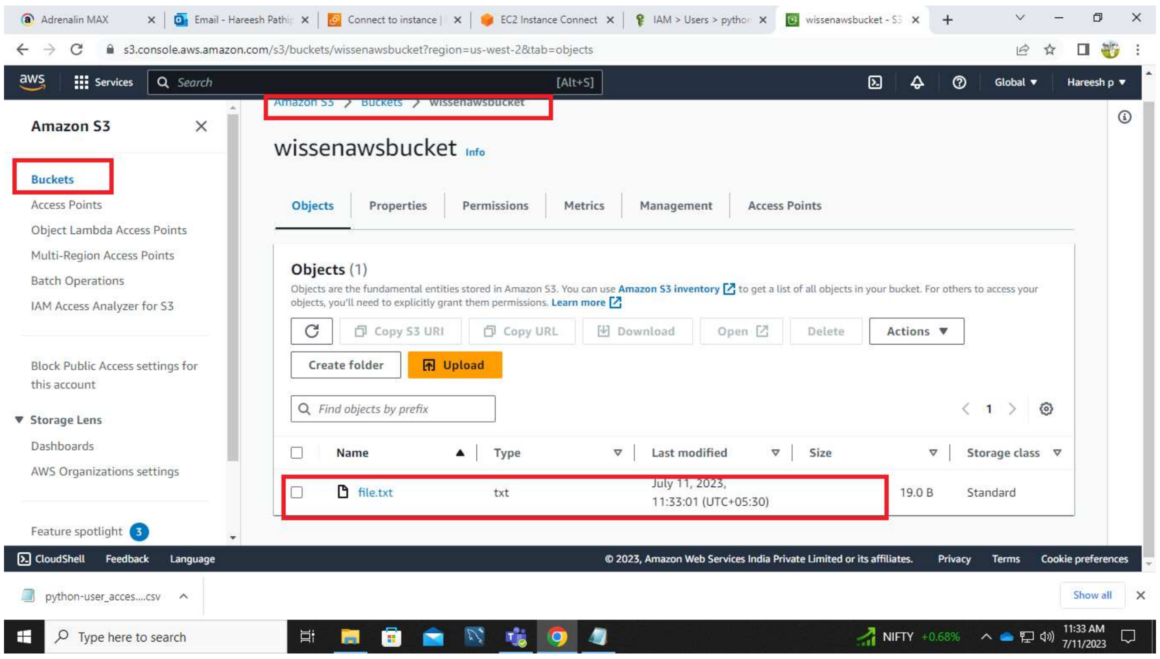
s3.upload_file('file.txt', 'wissenawsbucket', 'file.txt')

- S3Client allows interacting with AWS S3's buckets and objects



```
(env) [root@ip-172-31-19-110 ~]# python
Python 3.7.16 (default, Mar 10 2023, 03:25:26)
[GCC 7.3.1 20180712 (Red Hat 7.3.1-15)] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> import boto3
>>> import botocore
>>> s3 = boto3.resource('s3')
/root/wissen_app/env/lib64/python3.7/site-packages/boto3/compat.py:82: PythonDeprecationWarning: Boto3 will no longer support Python 3.7 starting Dece
mber 13, 2023. To continue receiving service updates, bug fixes, and security updates please upgrade to Python 3.8 or later. More information can be f
ound here: https://aws.amazon.com/blogs/developer/python-support-policy-updates-for-aws-sdks-and-tools/
  warnings.warn(warning, PythonDeprecationWarning)
>>> s3 = boto3.client('s3')
>>> s3.upload_file('file.txt', 'wissenawsbucket', 'file.txt')
>>>
>>>
>>>
```

- Here we can see one file in side bucket

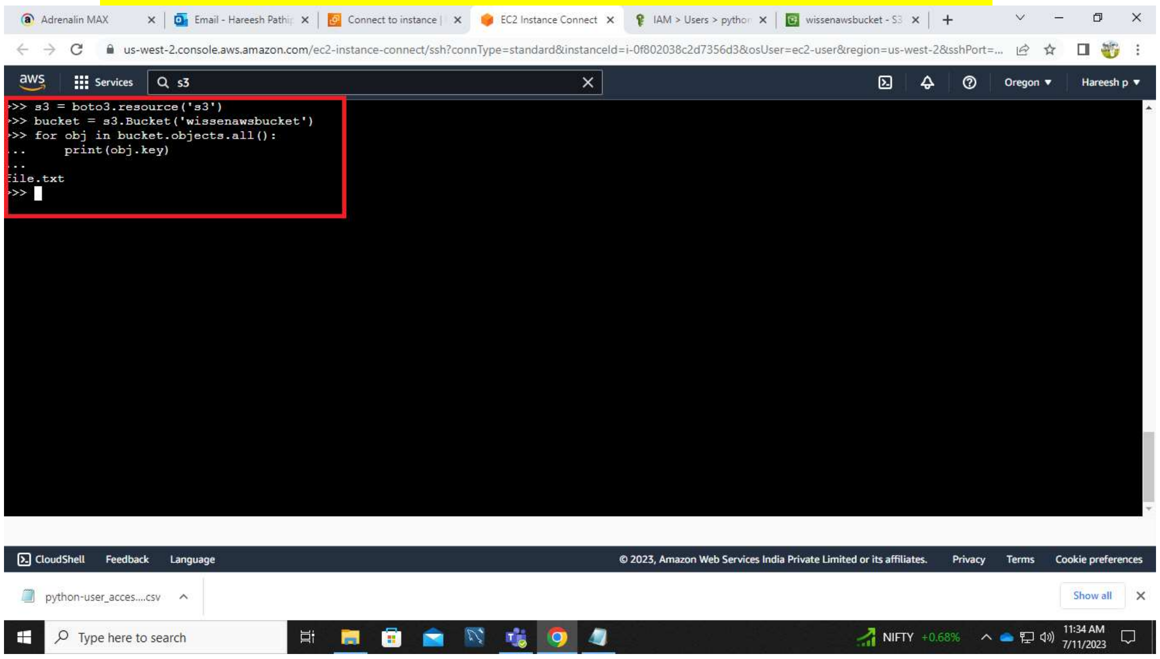


Objects (1)

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

Name	Type	Last modified	Size	Storage class
file.txt	txt	July 11, 2023, 11:33:01 (UTC+05:30)	19.0 B	Standard

- The following example gives how many objects availabe inside the buckets

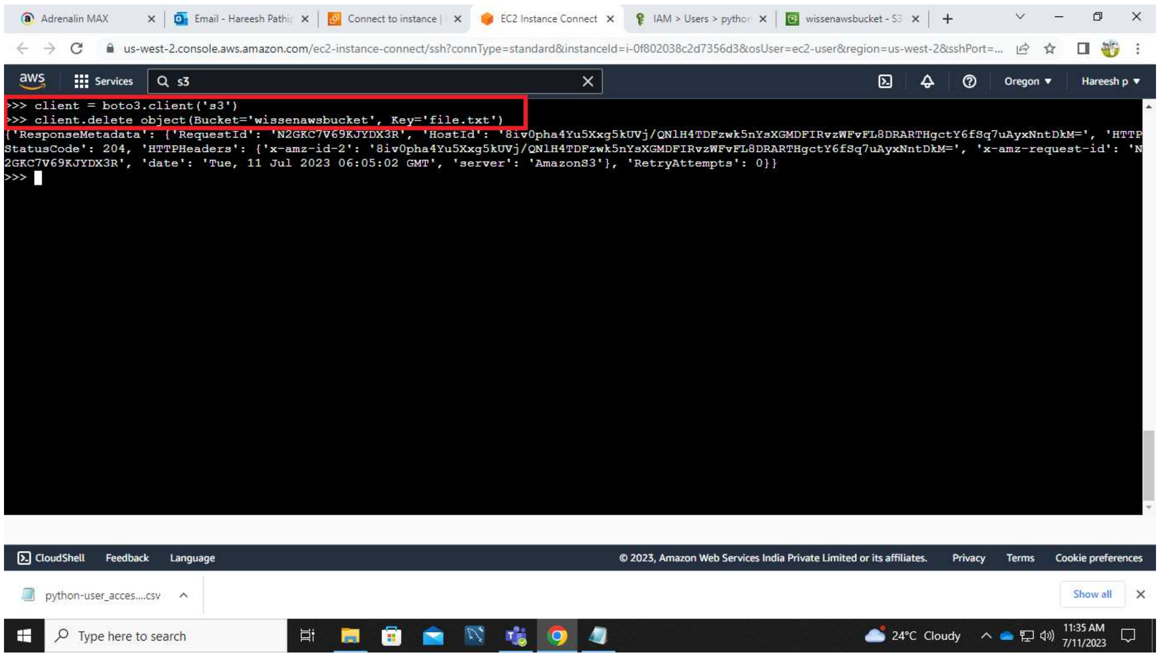


```
>> s3 = boto3.resource('s3')
>> bucket = s3.Bucket('wissenawsbucket')
>> for obj in bucket.objects.all():
..      print(obj.key)
..
ile.txt
>>
```

- The following example deletes an object from a non-versioned bucket.

client = boto3.client('s3')

client.delete_object(Bucket='wissenawsbucket', Key='file.txt')



```
>>> client = boto3.client('s3')
>>> client.delete object(Bucket='wissenawsbucket', Key='file.txt')
{'ResponseMetadata': {'RequestId': 'N2GKC7V69KJYDX3R', 'HostId': '8iv0pha4Yu5Xxg5kUVj/QNlH4TDFzwk5nYsXGMDFIRvzWFvFL8DRARTHgctY6fSq7uAyxNntDkM=', 'HTTP
StatusCode': 204, 'HTTPHeaders': {'x-amz-id-2': '8iv0pha4Yu5Xxg5kUVj/QNlH4TDFzwk5nYsXGMDFIRvzWFvFL8DRARTHgctY6fSq7uAyxNntDkM=', 'x-amz-request-id': 'N
2GKC7V69KJYDX3R', 'date': 'Tue, 11 Jul 2023 06:05:02 GMT', 'server': 'AmazonS3'}, 'RetryAttempts': 0}}
>>> 
```

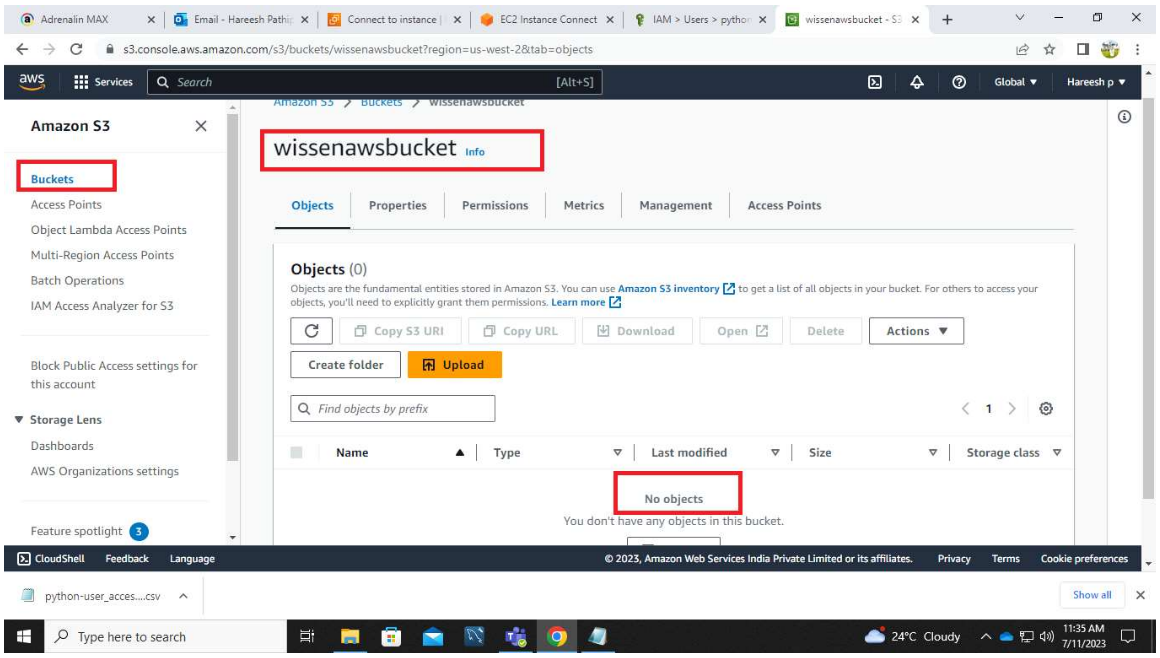
- Deletes the S3 bucket. All objects (including all object versions and delete markers) in the bucket must be deleted before the bucket itself can be deleted.

- The following example deletes the specified bucket.

client = boto3.client('s3')

response = client.delete_bucket(

Bucket='wissenawsbucket',)



s3.console.aws.amazon.com/s3/buckets/wissenawsbucket?region=us-west-2&tab=objects

Services
Search [Alt+S]
Global
Hareesh p
Amazon S3
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
Feature spotlight 3
Amazon S3 > Buckets > wissenawsbucket

wissenawsbucket Info

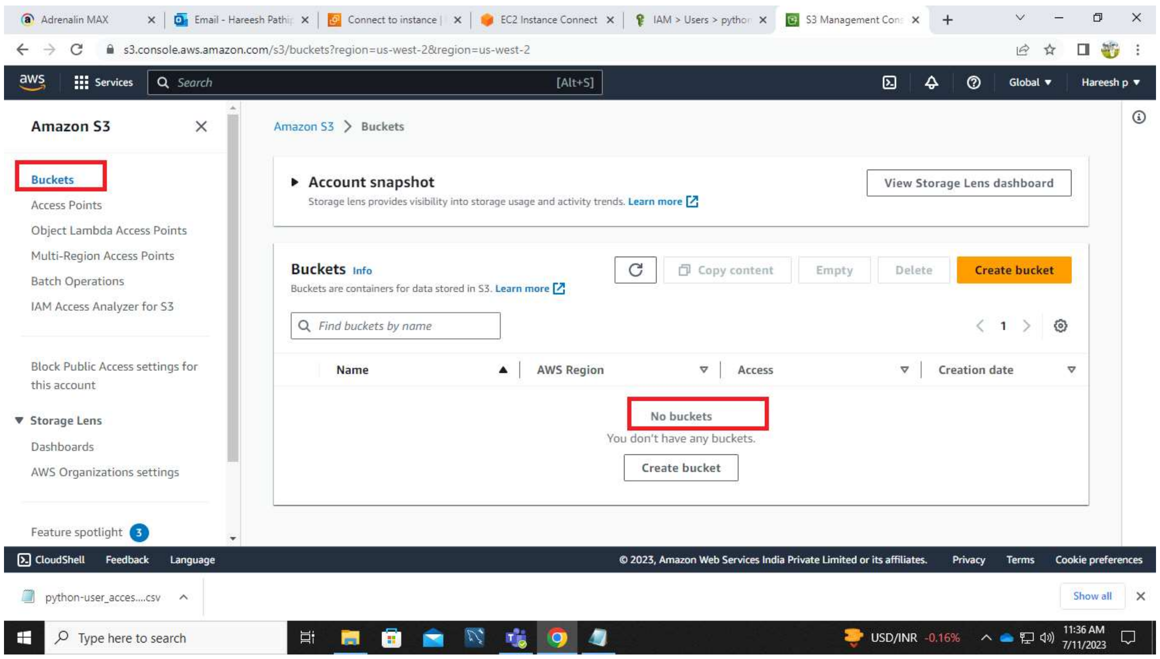
Objects (0)

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

Name	Type	Last modified	Size	Storage class

No objects

You don't have any objects in this bucket.



s3.console.aws.amazon.com/s3/buckets?region=us-west-2&region=us-west-2

Services
Search [Alt+S]
Global
Hareesh p
Amazon S3
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
Feature spotlight 3
Amazon S3 > Buckets

Account snapshot

Storage lens provides visibility into storage usage and activity trends. Learn more

Buckets Info

Buckets are containers for data stored in S3. Learn more

Name	AWS Region	Access	Creation date

No buckets

You don't have any buckets.