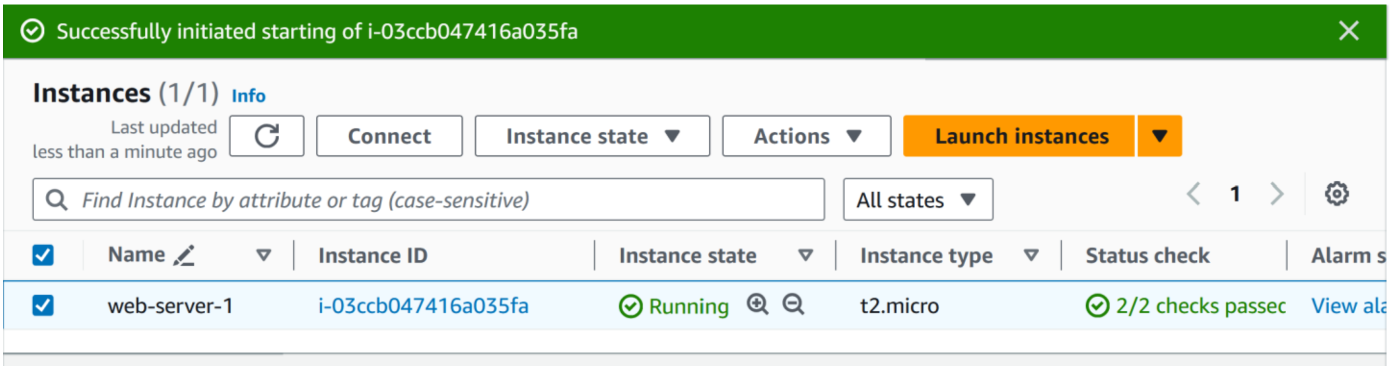
Name: Pranav Chaudhari

Task: Create a python script that automatically takes backups from the Apache webserver and save it in S3 bucket also find ways to automate it.

Step 1-- Launch EC2 instance: Goto AWS > EC2> launch a new instance, > choose AMI (I am using amazing Linux 2023)> ensure port 80 is enabled from everywhere as we are going to host a website on it and Apache webserver works on port 80.



Successfully initiated starting of i-03ccb047416a035fa

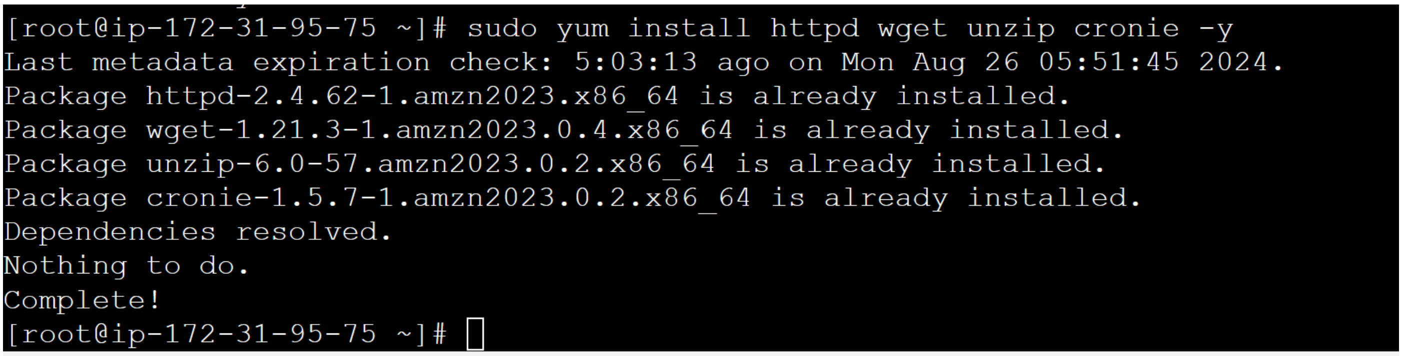
Instances (1/1) Info
Last updated less than a minute ago
Connect
Instance state ▼
Actions ▼
Launch instances
Find Instance by attribute or tag (case-sensitive)
All states ▼
1

☑	Name	Instance ID	Instance state	Instance type	Status check	Alarm s
☑	web-server-1	i-03ccb047416a035fa	Running	t2.micro	2/2 checks passec	View ala

Step 2 -- Install necessary packages and content on EC2: To simulate the website hosting and the logs are genuinely occurring from it, we are going to host a website and use the logs to put on s3 bucket later. For this operation we are going to need necessary packages and files. We will need python, boto3 module, crony, httpd, wget and unzip. Since it is Amazon linux, python comes pre-installed so skipping the installation of that.

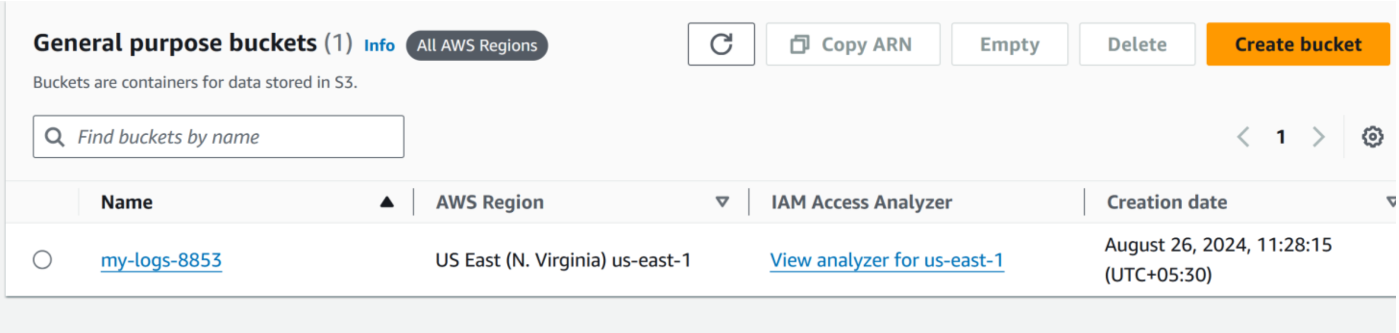
Httpd: using for webserver apache.
Wget: using to download website content.
Cronie: using to manage the corn jobs to schedule tasks.
Pip: it is a package manager for python to install the modules.
Boto3: it is a library an official AWS SDK for Python, to interact with AWS resources.
Tooplate.com: provides free html templates, using these templates to simulate our website.

```
sudo yum install httpd wget unzip cronie -y
sudo systemctl start httpd
sudo systemctl enable httpd
sudo systemctl start crond
sudo systemctl enable crond
sudo yum install python3-pip -y
pip install boto3
#download theme
sudo wget https://www.tooplate.com/zip-templates/2132_clean_work.zip
#unzip the file
sudo unzip 2132_clean_work.zip
sudo cp -r 2132_clean_work/* /var/www/html/
sudo systemctl restart httpd
```



```
[root@ip-172-31-95-75 ~]# sudo yum install httpd wget unzip cronie -y
Last metadata expiration check: 5:03:13 ago on Mon Aug 26 05:51:45 2024.
Package httpd-2.4.62-1.amzn2023.x86_64 is already installed.
Package wget-1.21.3-1.amzn2023.0.4.x86_64 is already installed.
Package unzip-6.0-57.amzn2023.0.2.x86_64 is already installed.
Package cronie-1.5.7-1.amzn2023.0.2.x86_64 is already installed.
Dependencies resolved.
Nothing to do.
Complete!
[root@ip-172-31-95-75 ~]# 
```

Step 3 -- Create S3 bucket: As it is going to be one time thingy, we are going to need to create a S3 bucket manually as later we are going to use it to store the logs.



General purpose buckets (1)

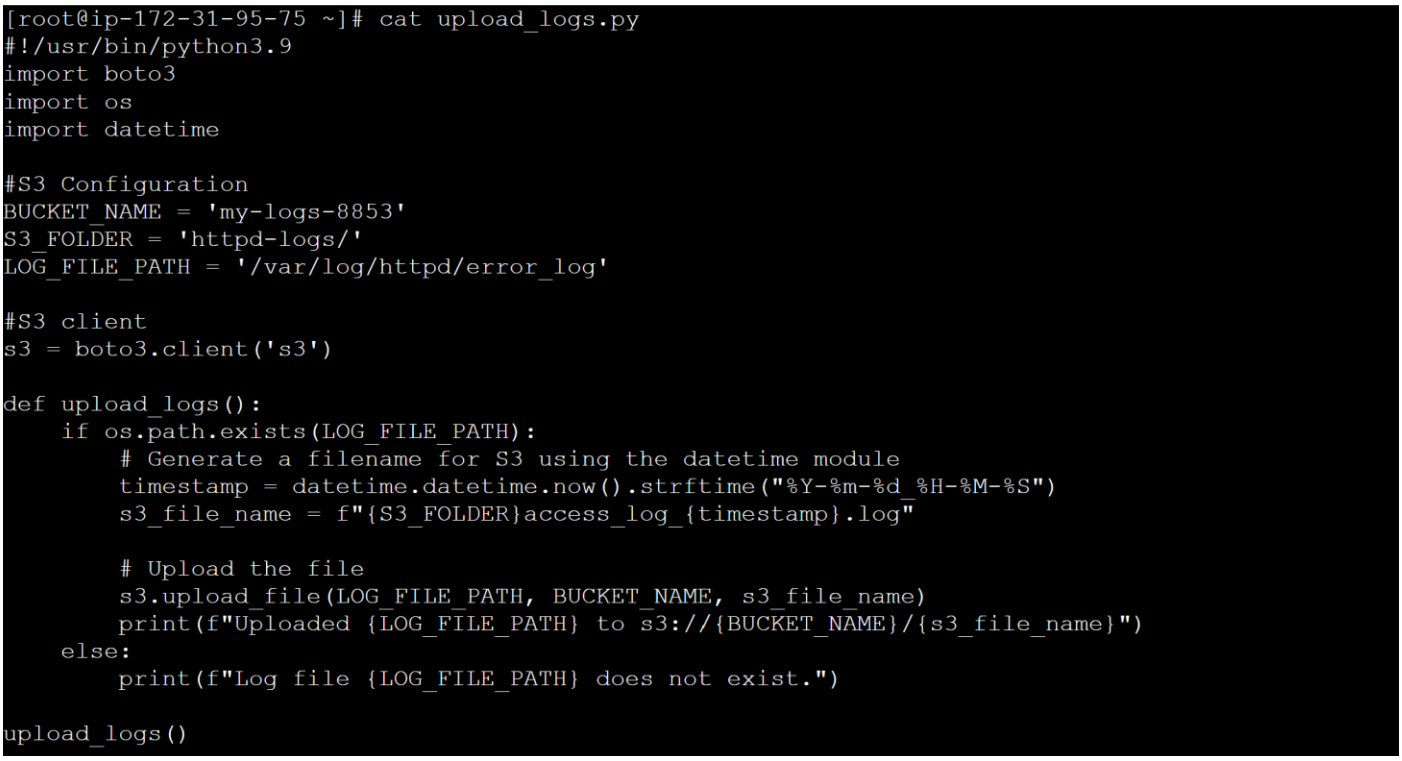
Buckets are containers for data stored in S3.

Name	AWS Region	IAM Access Analyzer	Creation date
○ my-logs-8853	US East (N. Virginia) us-east-1	View analyzer for us-east-1	August 26, 2024, 11:28:15 (UTC+05:30)

AWS > S3 > create bucket name.

Step 4 -- Create python script: Create the script in python and give executable permission to it.

Full script here: https://github.com/techdecipher/tasks/blob/main/upload_logs.py



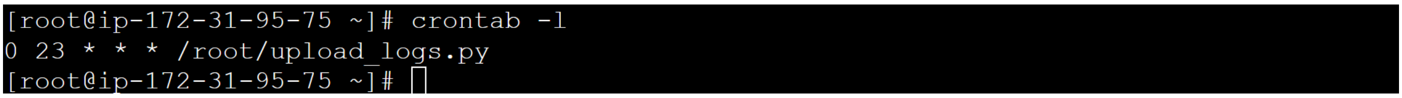
```
[root@ip-172-31-95-75 ~]# cat upload_logs.py
#!/usr/bin/python3.9
import boto3
import os
import datetime

#S3 Configuration
BUCKET_NAME = 'my-logs-8853'
S3_FOLDER = 'httpd-logs/'
LOG_FILE_PATH = '/var/log/httpd/error_log'

#S3 client
s3 = boto3.client('s3')

def upload_logs():
    if os.path.exists(LOG_FILE_PATH):
        # Generate a filename for S3 using the datetime module
        timestamp = datetime.datetime.now().strftime("%Y-%m-%d_%H-%M-%S")
        s3_file_name = f"{S3_FOLDER}access_log_{timestamp}.log"

        # Upload the file
        s3.upload_file(LOG_FILE_PATH, BUCKET_NAME, s3_file_name)
        print(f"Uploaded {LOG_FILE_PATH} to s3://{BUCKET_NAME}/{s3_file_name}")
    else:
        print(f"Log file {LOG_FILE_PATH} does not exist.")

upload_logs()
```

Give executable permission.

```
sudo chmod +x upload_logs.py
```

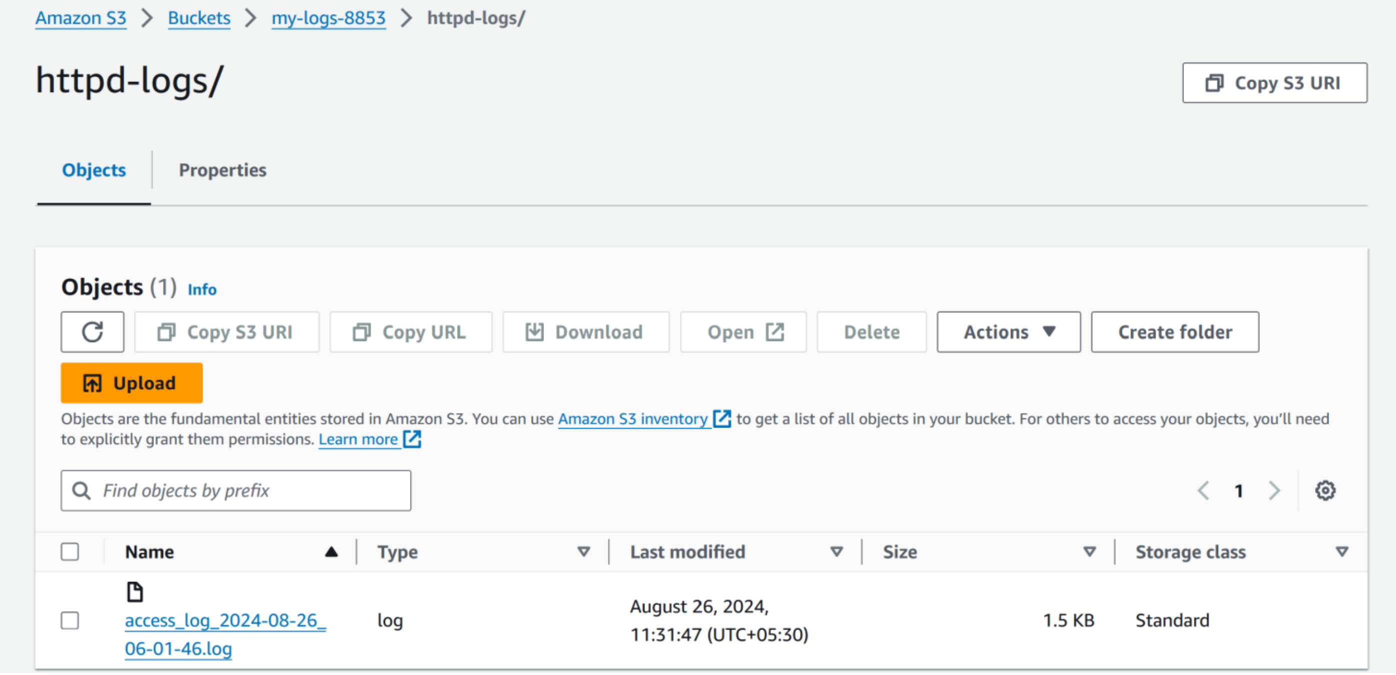
Step 5 -- setup CronJob: Its time we put the file in cron jobs so it runs on its own at certain time. The file will execute every day at 11PM.

```
Crontab -e
```



```
[root@ip-172-31-95-75 ~]# crontab -l
0 23 * * * /root/upload_logs.py
[root@ip-172-31-95-75 ~]# 
```

Output:



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

Find objects by prefix

□	Name	Type	Last modified	Size	Storage class
□	access_log_2024-08-26_06-01-46.log	log	August 26, 2024, 11:31:47 (UTC+05:30)	1.5 KB	Standard

This way we don't have to manually login to the app server or VM to review the logs, one can access s3 using AWS cli or manually and review the logs and see if everything is fine.