



Dhirubhai Ambani **University** Technology

Formerly DA-IICT

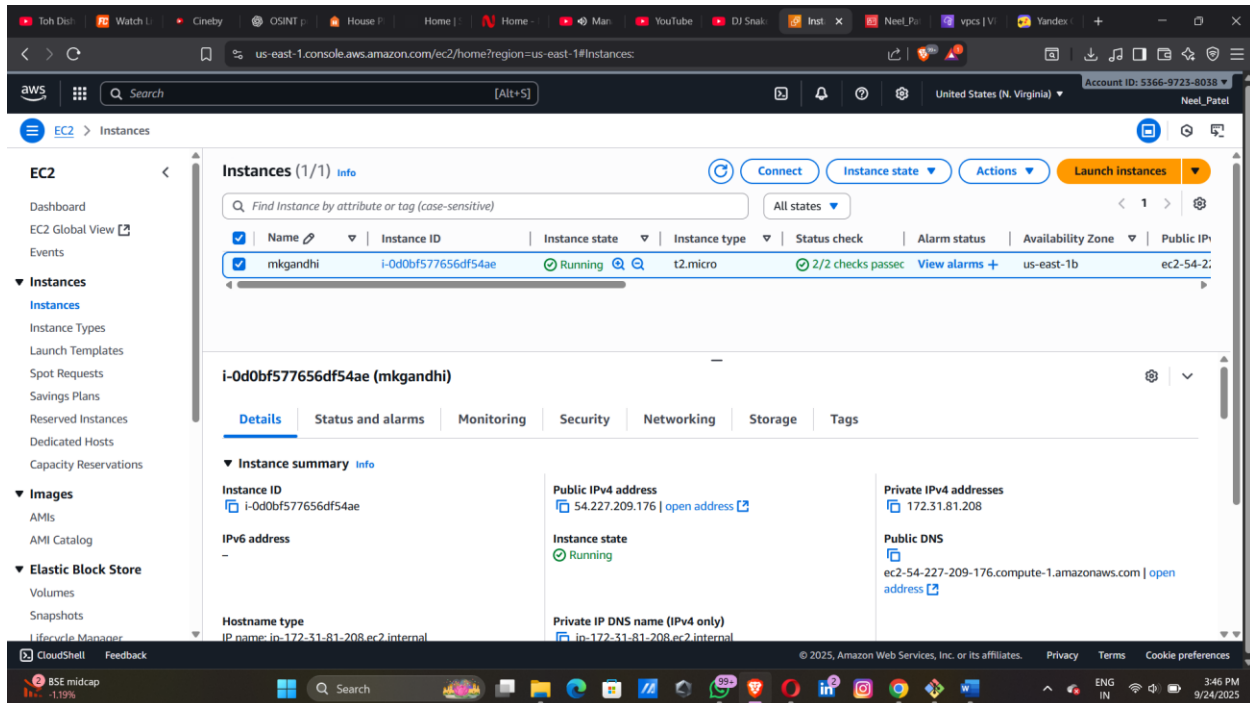
IT457 Cloud Computing

Assignment-EC2

Group 5

Name	Student ID
Sneh Joshi	202201048
Sahil Chaudhari	202201171
Kshitij Patel	202201232
Neel Patel	202201494

Instance



```
neelp@GeorgeBush MINGW64 ~/OneDrive/Desktop/Academic/Neel/Sem 7/IT457 Cloud Comp
uting/Ec2 Lab (master)
$ chmod 400 "mkgandhi.pem"

neelp@GeorgeBush MINGW64 ~/OneDrive/Desktop/Academic/Neel/Sem 7/IT457 Cloud Computing/Ec2 Lab (master)
$ ssh -i "mkgandhi.pem" ec2-user@ec2-54-227-209-176.compute-1.amazonaws.com

#_
~|##### Amazon Linux 2023
~|#####\
~|###|
~|\#/
~|V~'-'> https://aws.amazon.com/linux/amazon-linux-2023
~|. . /
~|/_/'/_/_/_/
[ec2-user@ip-172-31-81-208 ~]$
```

```

root@ip-172-31-81-208:/home/ec2-user# cd /home/ec2-user
root@ip-172-31-81-208:/home/ec2-user# ssh -i "/c/Users/Academic/OneDrive/Desktop/AmazonLinux/ssh-key.pem" ec2-user@ec2-54-227-209-176.compute-1.amazonaws.com
Warning: Permanently added 'ec2-54-227-209-176.compute-1.amazonaws.com' (ssh) to the list of known hosts.
Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

[ec2-user@ip-172-31-81-208 ~]$ sudo su
[root@ip-172-31-81-208 ec2-user]# wget https://s3.amazonaws.com/mountpoint-s3-release/latest/x86_64/mount-s3.rpm
--2025-09-24 09:48:03-- https://s3.amazonaws.com/mountpoint-s3-release/latest/x86_64/mount-s3.rpm
Resolving s3.amazonaws.com (s3.amazonaws.com)... 52.216.215.24, 52.217.82.238, 52.217.94.38, ...
Connecting to s3.amazonaws.com (s3.amazonaws.com) [52.216.215.24]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 11086952 (11M) [binary/octet-stream]
Saving to: 'mount-s3.rpm'

mount-s3.rpm                               100%[=====] 10.57M --KB/s in 0.09s

2025-09-24 09:48:04 (120 MB/s) - 'mount-s3.rpm' saved [11086952/11086952]

[root@ip-172-31-81-208 ec2-user]# yum install ./mount-s3.rpm
amazon linux 2023 Kernel Livepatch repository                                173 kb/s | 23 kb | 00:00
Dependencies resolved:

Package                               Architecture      Version              Repository              Size
Installing:
mount-s3                               x86_64            1.20.0-1             @commandline            11 M
Installing dependencies:
fuse                                   x86_64            2.9.9-13.amzn2023.0.2  amazonlinux             80 k
fuse-common                           x86_64            3.10.4-1.amzn2023.0.2  amazonlinux             8.5 k

Transaction Summary
Install 3 Packages

Total size: 11 M
Total download size: 88 k
Installed size: 50 M
Is this ok [y/N]: y
Downloading Packages:
(1/2): fuse-common-3.10.4-1.amzn2023.0.2.x86_64.rpm                242 kb/s | 8.5 kb | 00:00
(2/2): fuse-2.9.9-13.amzn2023.0.2.x86_64.rpm                      1.9 MB/s | 80 kb | 00:00
Total                                                                911 kb/s | 88 kb | 00:00
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction:
  Preparing                :
  Installing               : fuse-common-3.10.4-1.amzn2023.0.2.x86_64          1/1
                          : fuse-2.9.9-13.amzn2023.0.2.x86_64              1/3
  Installing               : fuse-2.9.9-13.amzn2023.0.2.x86_64              2/3

```

```

C:\Users\user> aws s3 cp --recursive . s3://my-bucket/
Total download size: 88 kB
Installed size: 50 W
Is this ok [y/N]: y
Downloading Packages:
(1/2): fuse-common-3.10.4-1.amzn2023.0.2.x86_64.rpm
(2/2): fuse-2.9.9-13.amzn2023.0.2.x86_64.rpm

Total                                                    242 kB/s |  8.5 kB  00:00
1.9 MB/s | 80 kB  00:00
-----
Running transaction check
Transaction check succeeded.
Running transaction test
Transaction test succeeded.
Running transaction
  Preparing                : fuse-common-3.10.4-1.amzn2023.0.2.x86_64                1/1
  Installing               : fuse-common-3.10.4-1.amzn2023.0.2.x86_64                1/3
  Installing               : fuse-2.9.9-13.amzn2023.0.2.x86_64                    2/3
  Installing               : mount-s3-1.20.0-1.x86_64                             3/3
  Running scriptlet        : mount-s3-1.20.0-1.x86_64                             3/3
  Verifying                : fuse-2.9.9-13.amzn2023.0.2.x86_64                    1/3
  Verifying                : fuse-common-3.10.4-1.amzn2023.0.2.x86_64              2/3
  Verifying                : mount-s3-1.20.0-1.x86_64                             3/3
Installed:
fuse-2.9.9-13.amzn2023.0.2.x86_64
fuse-common-3.10.4-1.amzn2023.0.2.x86_64
mount-s3-1.20.0-1.x86_64

Complete!
[root@ip-172-31-81-208 ec2-user]# ls
mount-s3.rpm
[root@ip-172-31-81-208 ec2-user]# aws s3 ls
Unable to locate credentials. You can configure credentials by running "aws configure".
[root@ip-172-31-81-208 ec2-user]# mkdir mountfolder
[root@ip-172-31-81-208 ec2-user]# ls
mount-s3.rpm  mountfolder
[root@ip-172-31-81-208 ec2-user]# mount-s3 mkgandhi mountfolder/
Error: Failed to create S3 client
Caused by:
  0: initial listobjectsV2 failed for bucket mkgandhi in region us-east-1
  1: Client error
  2: No signing credentials available, see CRT debug logs
Error: Failed to create mount process
[root@ip-172-31-81-208 ec2-user]# aws configure
AWS Access key ID [None]: AKIA2SM5I1P43ENW
AWS Secret Access Key [None]: AC
[root@ip-172-31-81-208 ec2-user]# aws configure
AWS Access key ID [None]: AKIA2SM5I1P43ENW20X
AWS Secret Access Key [None]: rRiBkvNOTcdRABY9JwtpvC1q9A5sesi2VCBtJT
Default region name [None]: Global
Default output format [None]:
[root@ip-172-31-81-208 ec2-user]# mount-s3 mkgandhi mountfolder/
bucket mkgandhi is mounted at mountfolder/
[root@ip-172-31-81-208 ec2-user]# ls
mount-s3.rpm  mountfolder
[root@ip-172-31-81-208 ec2-user]# cd mountfolder/
[root@ip-172-31-81-208 mountfolder]# ls
Assignment_1.docx
[root@ip-172-31-81-208 mountfolder]# |

```

Case Study: Mounting S3 on EC2 in a Real Startup

1. **How does mounting S3 on EC2 simplify your app's workflow compared to manually moving files back and forth?**

Mounting S3 makes the bucket appear like a local directory on EC2, so the app can read/write photos directly without extra upload/download scripts. This removes manual file transfer steps, reduces code complexity, and speeds up the workflow.

2. **If suddenly 10,000 users upload images in a single day, what issues might you face with this setup? (Think speed, cost, or performance.)**

Large uploads may cause network bottlenecks, high S3 request costs, and increased EC2 processing load. The instance may slow down or run out of compute capacity, leading to performance issues for users.

3. **What would you plan next to make the system more scalable and production-ready?**

I would introduce **auto-scaling EC2 instances**, use **S3 event triggers with AWS Lambda** for processing, and adopt a **queue system (SQS/Kinesis)** to handle spikes in uploads. Adding a load balancer and caching layer would further improve scalability and performance.