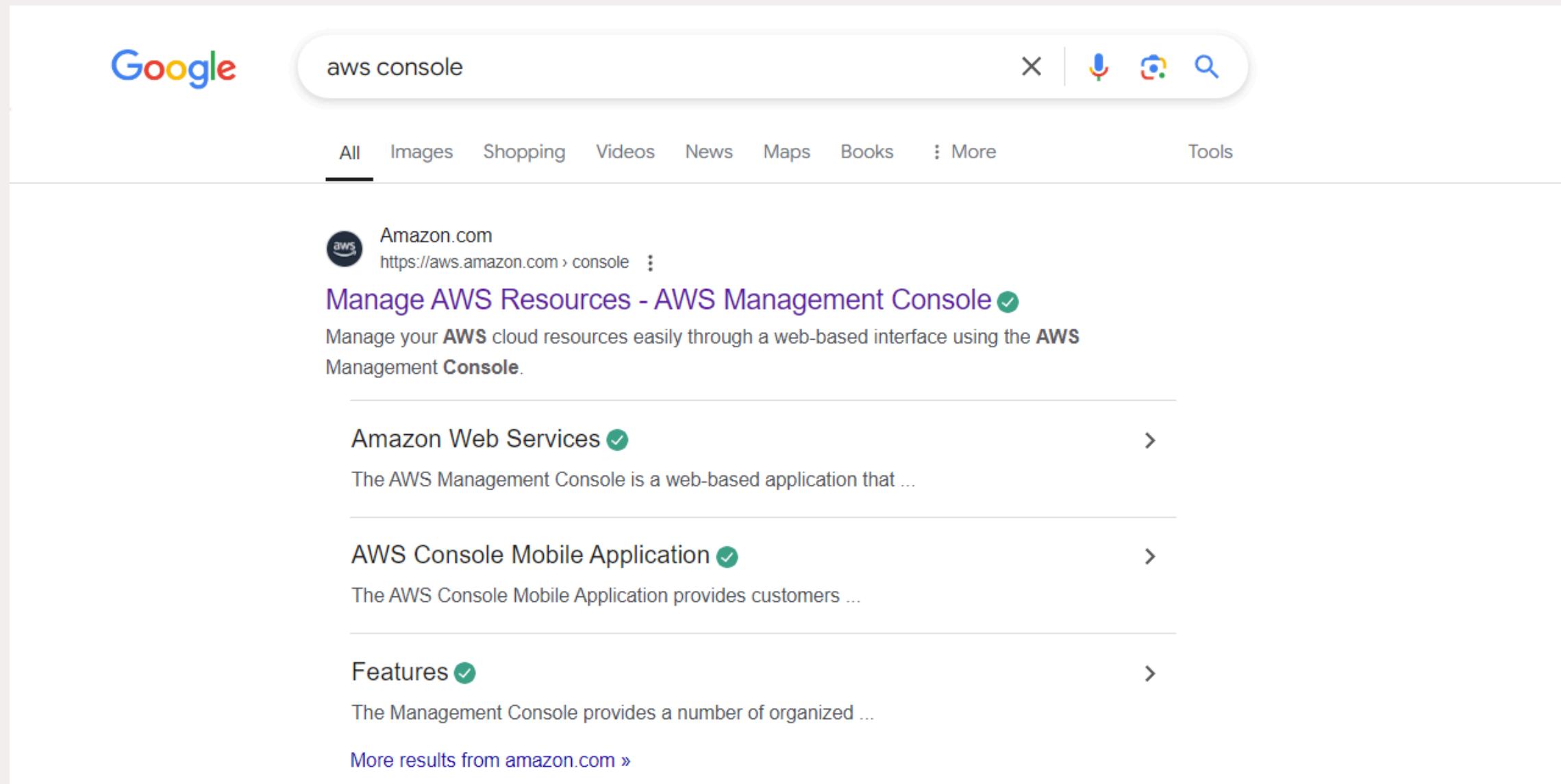


AWS Cloud Computing

Task 1 : Launching EC2 instances

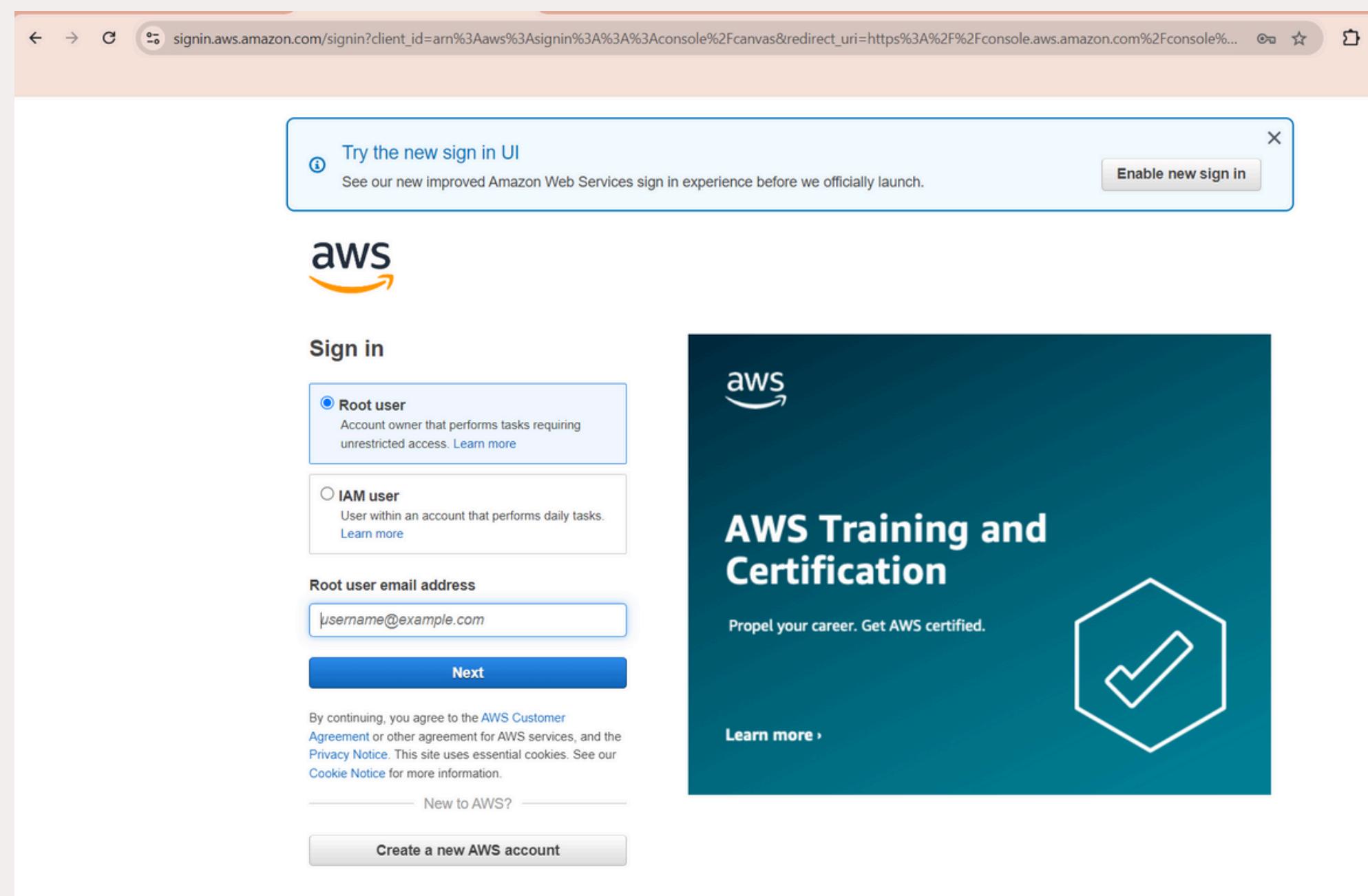
1. Search AWS Console on any browser



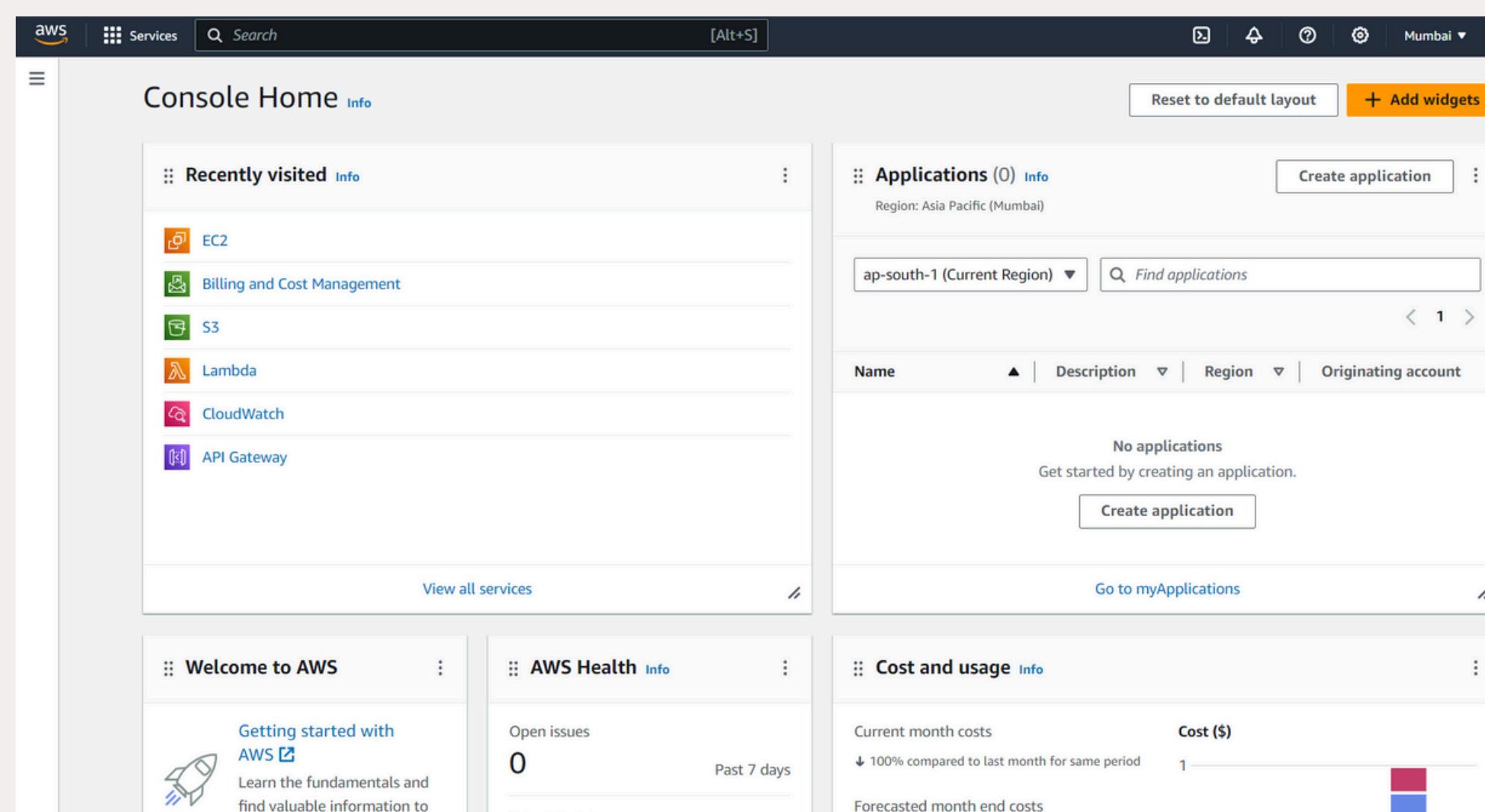
2. Create an AWS Account (If You Don't Have One)

A screenshot of the AWS Management Console homepage. The top navigation bar includes links for About AWS, Contact Us, Support, English, My Account, and Sign In to the Console. Below the navigation is a breadcrumb trail: Products > Management and Governance > Management Console. The main heading is "AWS Management Console" with the subtext "Everything you need to access and manage the AWS Cloud — in one web interface". A "Sign in" button is prominently displayed. At the bottom, there are four service cards: "AWS Certification", "Free AWS Training", "7 Reasons to get AWS Certified", and "AWS Training". The "AWS Training" card features a chatbot icon with a message bubble saying "Hi, I can connect you with an AWS representative or answer questions you have on AWS."

3.Sign In to the AWS Management Console



4.Navigate to EC2



5.Launch an EC2 Instance

The screenshot shows the AWS EC2 Dashboard in the Asia Pacific (Mumbai) Region. The left sidebar includes sections for Instances, Images, Elastic Block Store, and Network & Security. The main area displays EC2 resources: 0 running instances, 0 Auto Scaling Groups, 0 Capacity Reservations, 0 Dedicated Hosts, 0 Elastic IPs, 3 Instances, 1 Key pair, 0 Load balancers, 0 Placement groups, 4 Security groups, 0 Snapshots, and 2 Volumes. A prominent 'Launch instance' button is available. The right sidebar provides information about the EC2 Free Tier, service health, offer usage, and account attributes.

6.Configure Instance Details

The screenshot shows the 'Launch an instance' wizard. In the 'Name and tags' step, the name 'Web1' is entered. In the 'Application and OS Images (Amazon Machine Image)' step, the user is searching for an AMI. A tooltip on the right explains the EC2 Free Tier benefits. The summary step shows 1 instance being launched with the selected AMI, instance type t2.micro, and other configurations like a new security group and 1 volume (8 GiB). The final step shows the 'Launch instance' button.

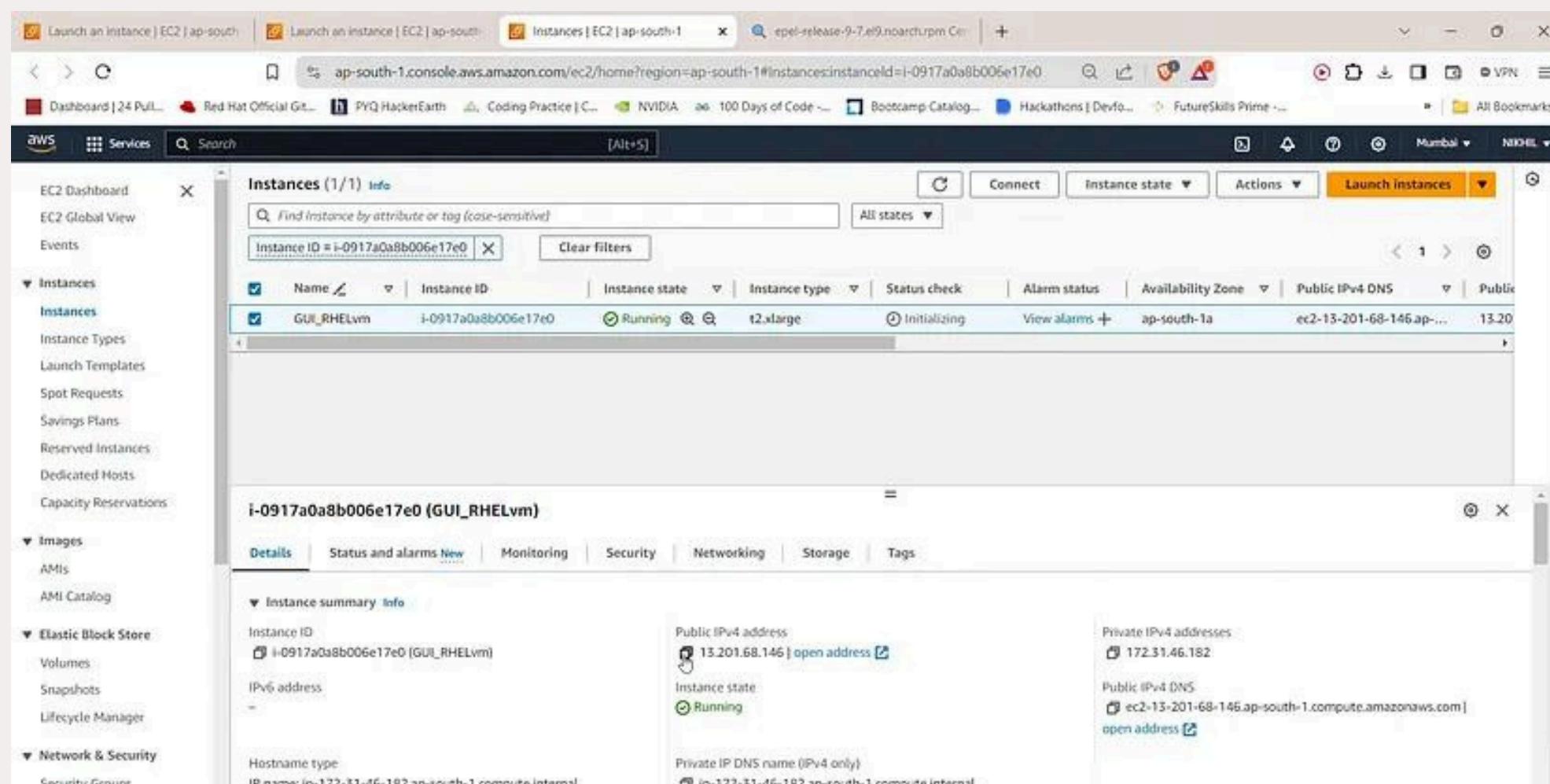
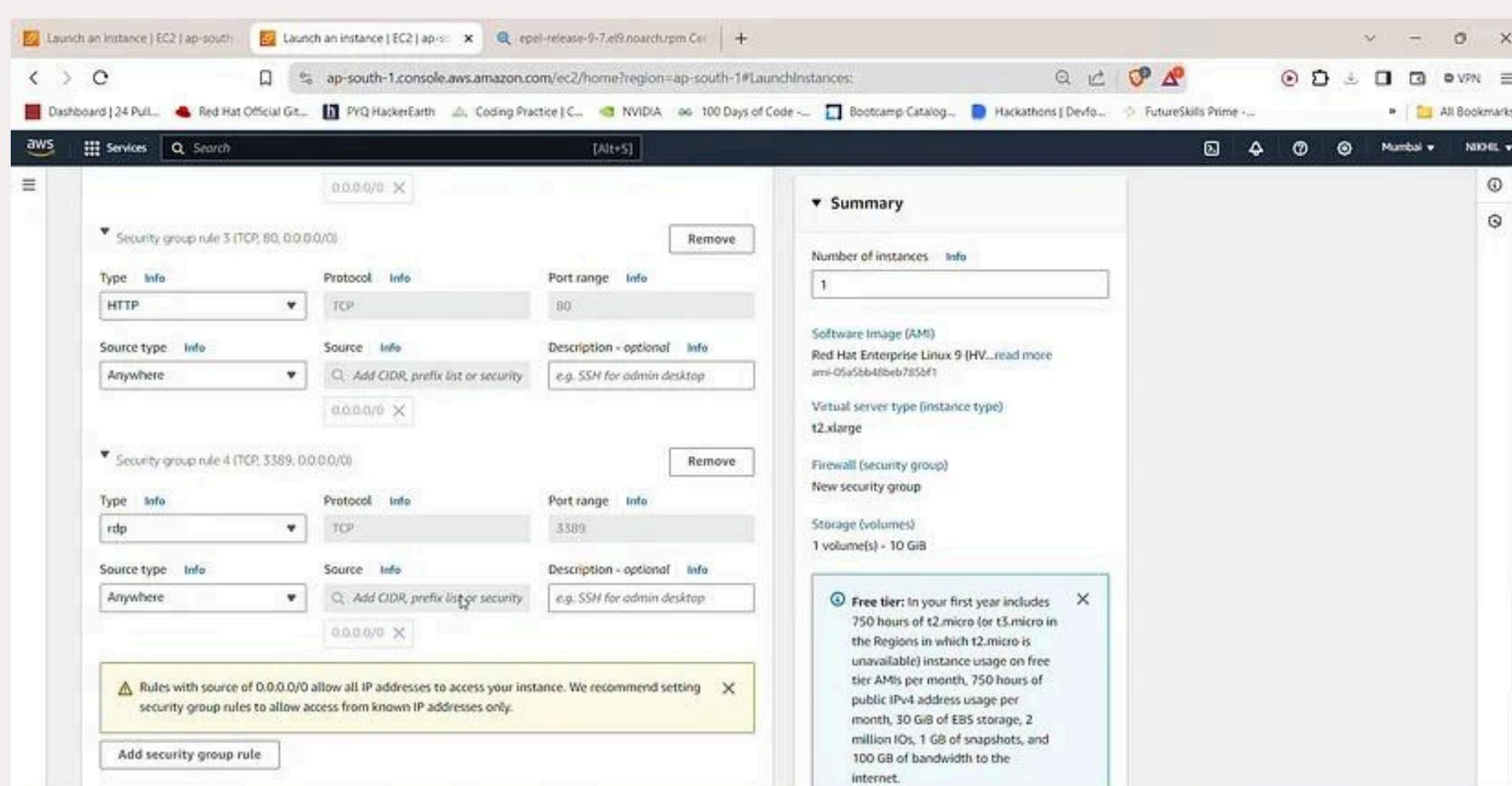
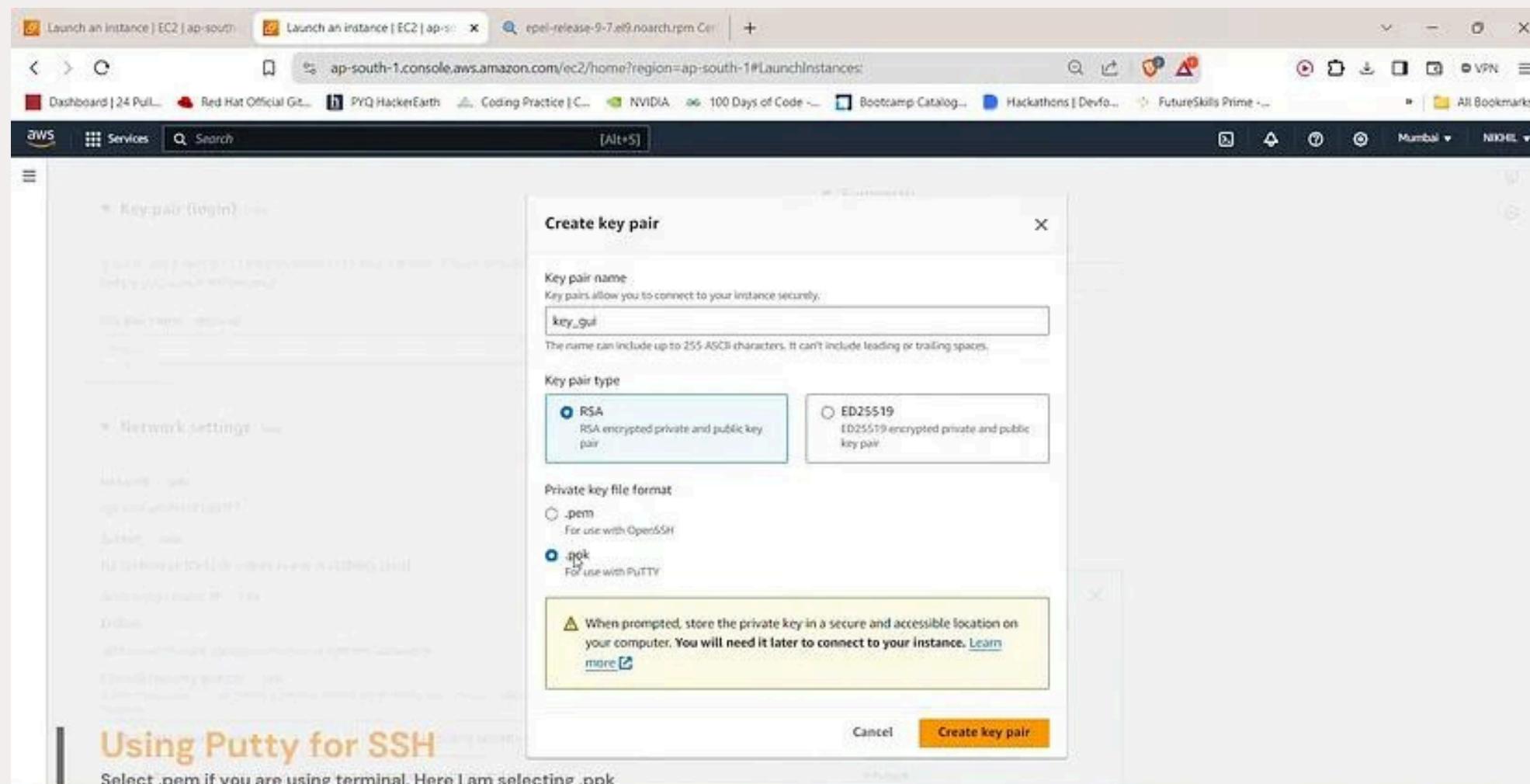
7.Launch Your Instance Access Your EC2 Instance

The screenshot shows the AWS EC2 Instances page. On the left, there's a sidebar with navigation links like EC2 Dashboard, EC2 Global View, Events, Instances (with sub-links for Instances, Instance Types, Launch Templates, Spot Requests, Savings Plans, Reserved Instances, Dedicated Hosts, Capacity, and Reservations), Images (AMIs, AMI Catalog), Elastic Block Store (Volumes, Snapshots, Lifecycle Manager), and Network & Security (Security Groups). The main content area has a search bar at the top. Below it, a table lists instances. One instance, "Web1" (ID: i-0f6a0e4bcd4733af), is selected and shown in detail below the table. The "Details" tab is active, displaying information such as Public IPv4 address (65.1.134.13), Instance state (Running), Hostname type (IP name: ip-172-31-6-201.ap-south-1.compute.internal), and Auto-assigned IP address (65.1.134.13 [Public IP]). Other tabs include Status and alarms, Monitoring, Security, Networking, Storage, and Tags.

Task 2 : RHEL GUI instance in cloud

Step 1: Launch a RHEL 9 Instance on AWS EC2

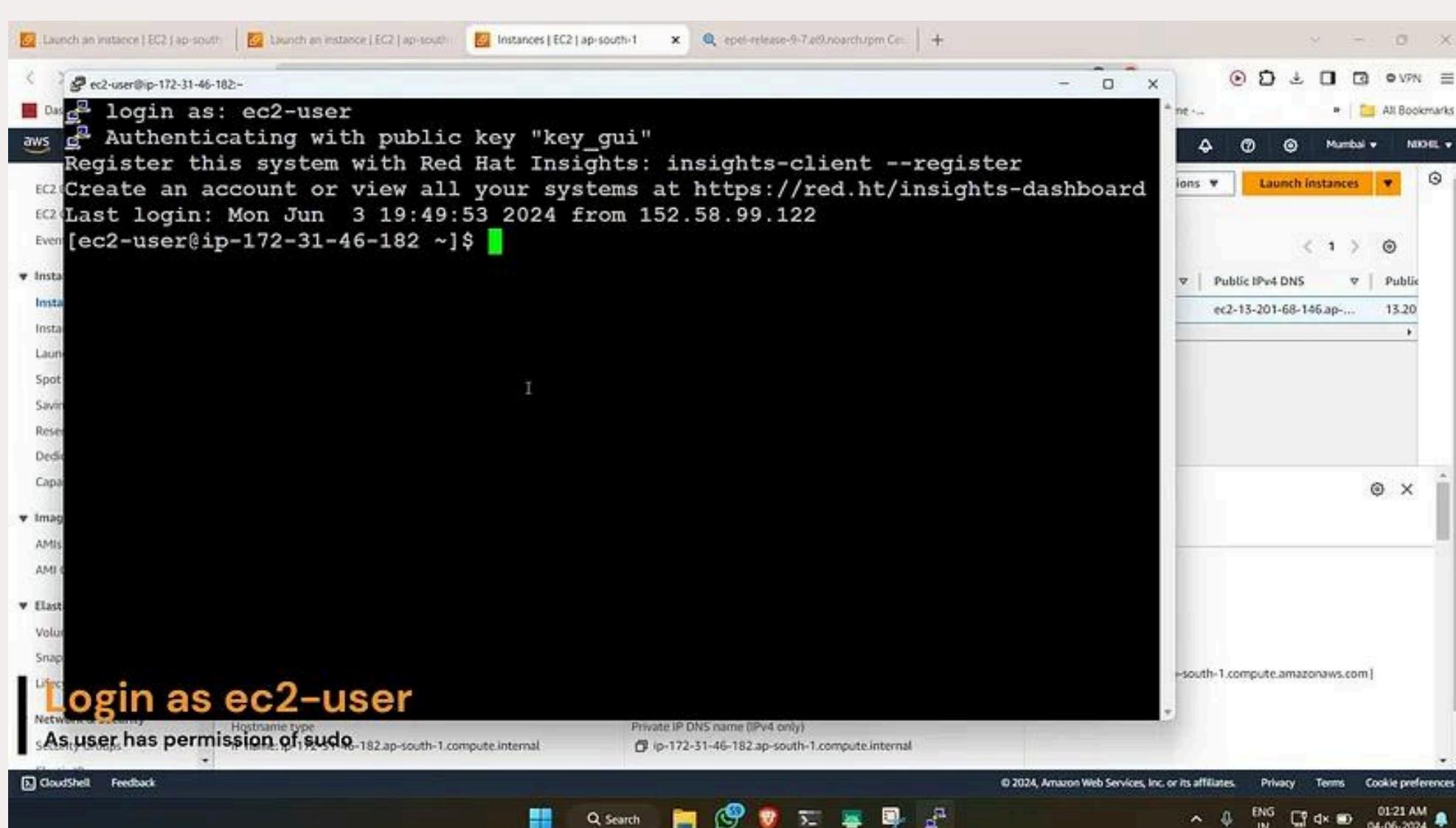
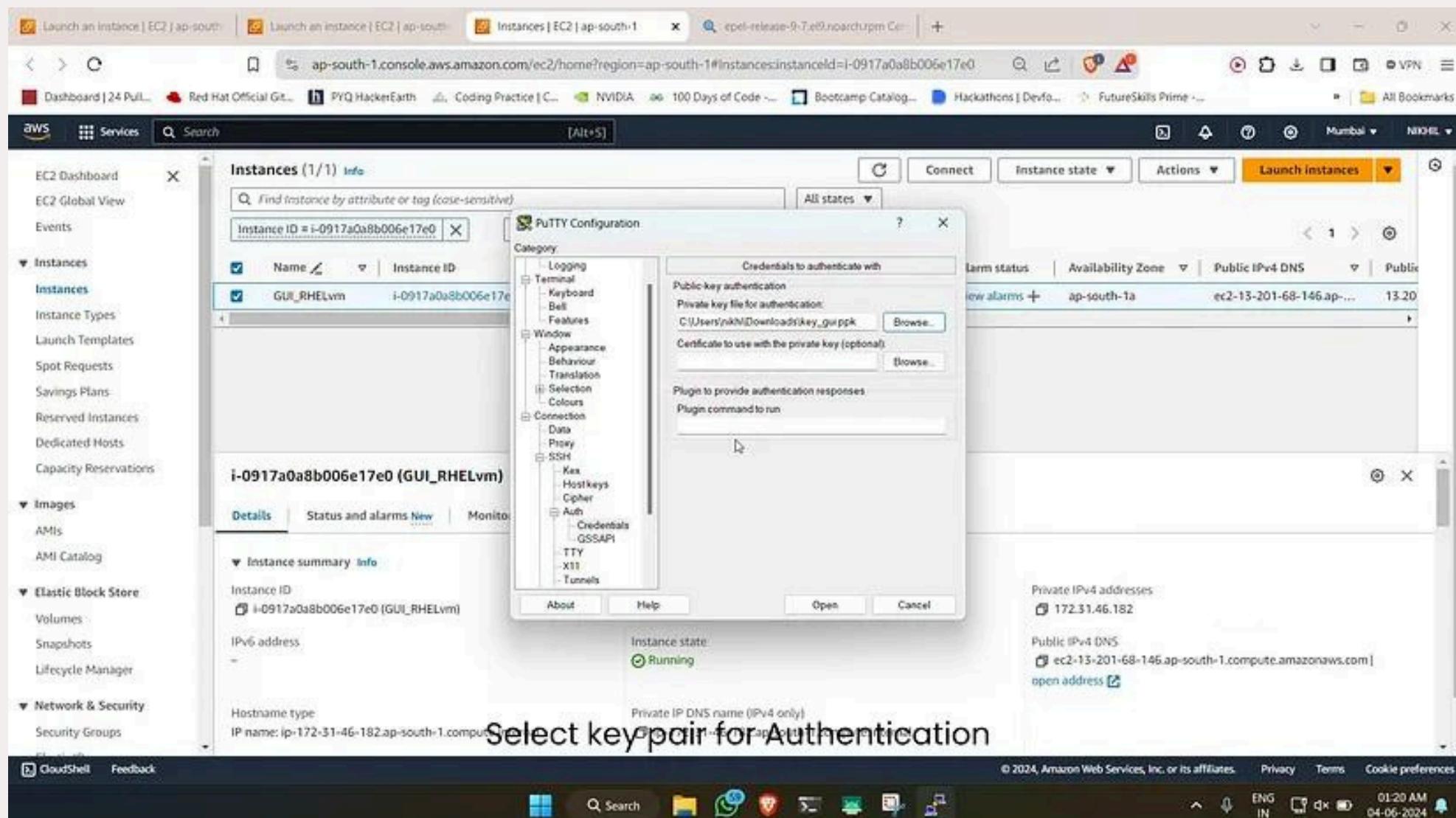
1. Log in to your AWS Management Console.
2. Navigate to EC2 Dashboard and click on Launch Instance.
3. Select RHEL 9 as the AMI.



Step 2: Connect to the Instance via SSH

If you are using PuTTY for SSH access, follow these steps else open Windows Terminal and type `ssh ec2-user@<ip-address-you-copied>`:

1. Open PuTTY and enter the public IP address of your EC2 instance.
2. In the SSH section, go to Auth and select the .ppk file (private key) you generated.
3. Connect to the instance by logging in as ec2-user.



Step 3: Set Up GUI on RHEL 9

Switch to the root user:

sudo su - root

2. Set the root password:

passwd

3. Register the system with Red Hat Insights:

insights-client --register

4. Update your system:

yum update -y

5. List available groups and install the GUI:

yum grouplist

yum group install "Server with GUI" -y

```
root@ip-172-31-46-182:~# login as: ec2-user
[ec2-user@ip-172-31-46-182 ~]$ sudo su
[root@ip-172-31-46-182 ec2-user]# passwd
Changing password for user root.
New password:
BAD PASSWORD: The password is shorter than 8 characters
Retype new password:
passwd: all authentication tokens updated successfully.
[root@ip-172-31-46-182 ec2-user]# yum update -y
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use subscription-manager to register.

Red Hat Enterprise Linux 9 for x86_64 - AppStream from RHUI (RPMs)      51 MB/s | 35 MB   00:00
```

```
root@ip-172-31-46-182:~# yum grouplist
sssd-client-2.9.4-6.el9_4.x86_64
sssd-common-2.9.4-6.el9_4.x86_64
sssd-kcm-2.9.4-6.el9_4.x86_64
subscription-manager-1.29.40-1.el9.x86_64

sudo-1.9.5p2-10.el9_3.x86_64
systemd-252-32.el9_4.x86_64
systemd-libs-252-32.el9_4.x86_64
systemd-pam-252-32.el9_4.x86_64
systemd-rpm-macros-252-32.el9_4.noarch
systemd-udev-252-32.el9_4.x86_64
tuned-2.22.1-1.el9.noarch
tzdata-2024a-1.el9.noarch
util-linux-2.37.4-18.el9.x86_64
util-linux-core-2.37.4-18.el9.x86_64
xfsprogs-6.3.0-1.el9.x86_64
yum-4.14.0-9.el9.noarch
yum-utils-4.3.0-13.el9.noarch

Installed:
freetype-2.10.4-9.el9.x86_64
grub2-tools-efi-1:2.06-77.el9.x86_64
harfbuzz-2.7.4-10.el9.x86_64
kernel-5.14.0-427.18.1.el9_4.x86_64
kernel-modules-5.14.0-427.18.1.el9_4.x86_64
libpng-2:1.6.37-12.el9.x86_64
openssl-fips-provider-3.0.7-2.el9.x86_64

graphite2-1.3.14-9.el9.x86_64
grub2-tools-extra-1:2.06-77.el9.x86_64
jq-1.6-16.el9.x86_64
kernel-core-5.14.0-427.18.1.el9_4.x86_64
kernel-modules-core-5.14.0-427.18.1.el9_4.x86_64
oniguruma-6.9.6-1.el9.5.x86_64

Complete!
[root@ip-172-31-46-182 ec2-user]#
```

Step 3: Set Up GUI on RHEL 9

Switch to the root user:

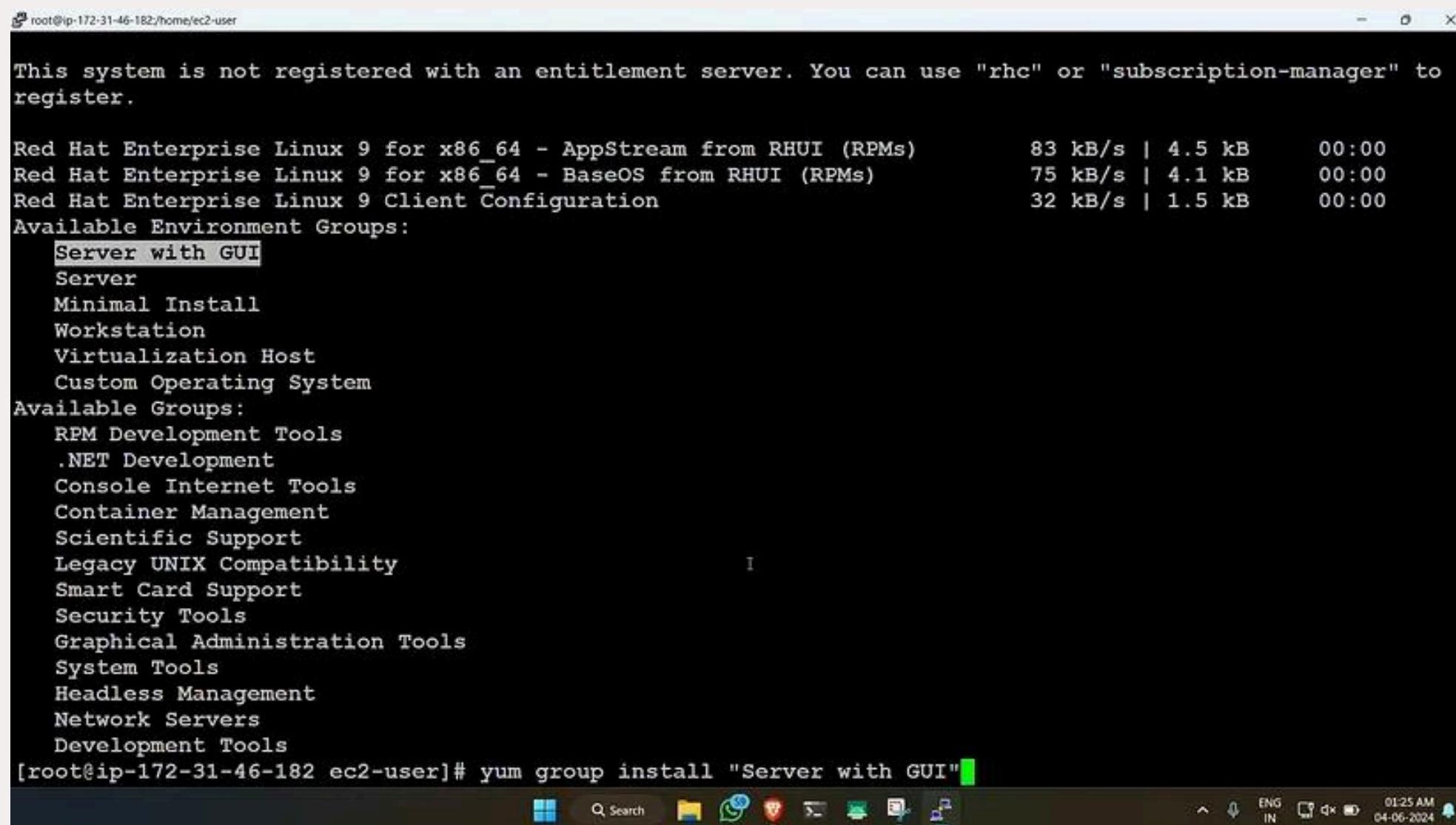
```
sudo su - root
```

2. Set the root password:

```
passwd
```

3. Register the system with Red Hat Insights:

```
insights-client --register
```



```
This system is not registered with an entitlement server. You can use "rhc" or "subscription-manager" to register.

Red Hat Enterprise Linux 9 for x86_64 - AppStream from RHUI (RPMs)      83 kB/s | 4.5 kB    00:00
Red Hat Enterprise Linux 9 for x86_64 - BaseOS from RHUI (RPMs)        75 kB/s | 4.1 kB    00:00
Red Hat Enterprise Linux 9 Client Configuration                      32 kB/s | 1.5 kB    00:00

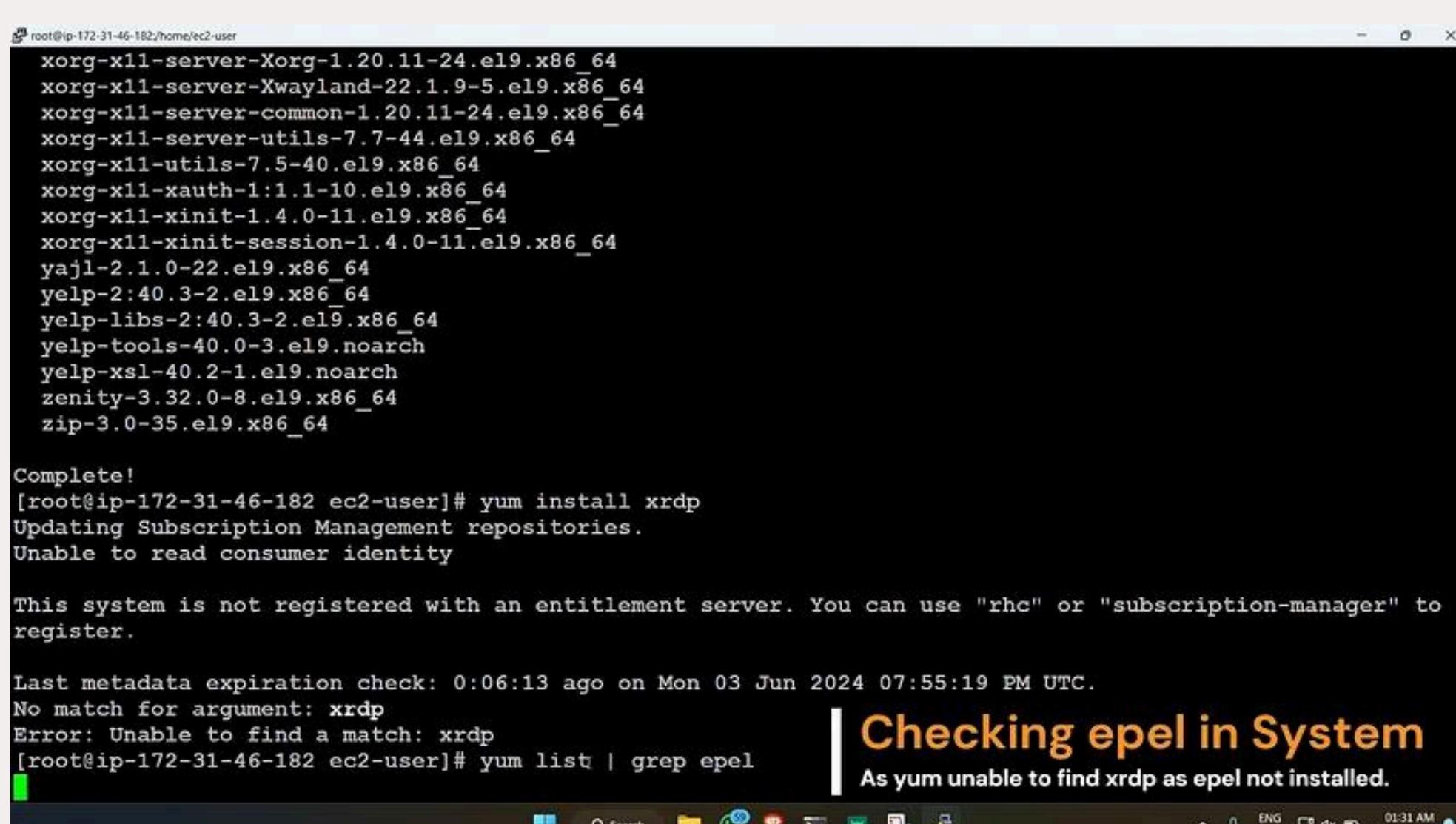
Available Environment Groups:
  Server with GUI
  Server
  Minimal Install
  Workstation
  Virtualization Host
  Custom Operating System

Available Groups:
  RPM Development Tools
  .NET Development
  Console Internet Tools
  Container Management
  Scientific Support
  Legacy UNIX Compatibility
  Smart Card Support
  Security Tools
  Graphical Administration Tools
  System Tools
  Headless Management
  Network Servers
  Development Tools

[root@ip-172-31-46-182 ec2-user]# yum group install "Server with GUI"
```

4. Update your system: `yum update -y`

5. List available groups and install the GUI:
`yum grouplist`
`yum group install "Server with GUI" -y`



```
xorg-x11-server-Xorg-1.20.11-24.el9.x86_64
xorg-x11-server-Xwayland-22.1.9-5.el9.x86_64
xorg-x11-server-common-1.20.11-24.el9.x86_64
xorg-x11-server-utils-7.7-44.el9.x86_64
xorg-x11-utils-7.5-40.el9.x86_64
xorg-x11-xauth-1.1.1-10.el9.x86_64
xorg-x11-xinit-1.4.0-11.el9.x86_64
xorg-x11-xinit-session-1.4.0-11.el9.x86_64
yajl-2.1.0-22.el9.x86_64
yelp-2:40.3-2.el9.x86_64
yelp-libs-2:40.3-2.el9.x86_64
yelp-tools-40.0-3.el9.noarch
yelp-xsl-40.2-1.el9.noarch
zenity-3.32.0-8.el9.x86_64
zip-3.0-35.el9.x86_64

Complete!
[root@ip-172-31-46-182 ec2-user]# yum install xrdp
Updating Subscription Management repositories.
Unable to read consumer identity

This system is not registered with an entitlement server. You can use "rhc" or "subscription-manager" to register.

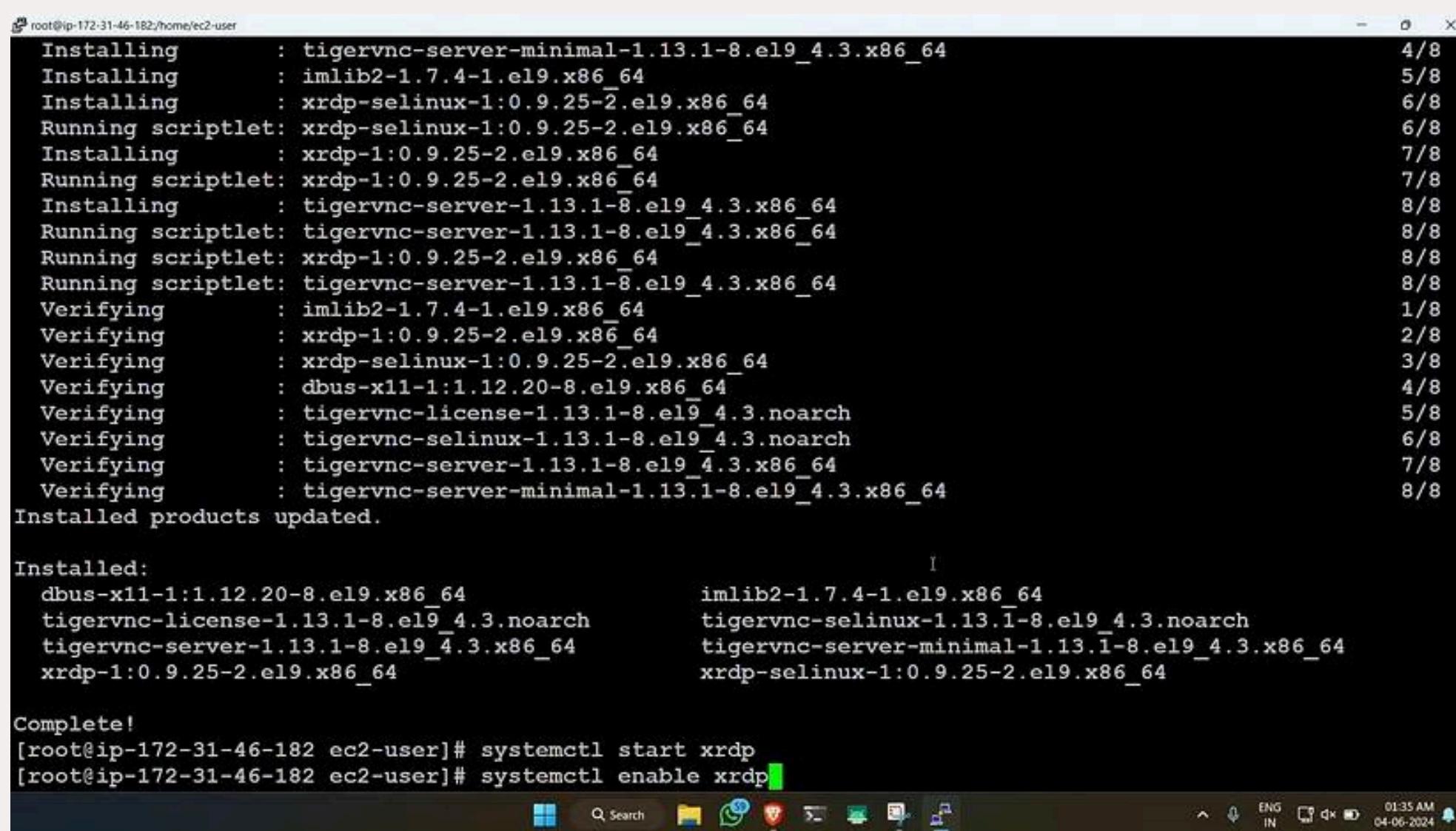
Last metadata expiration check: 0:06:13 ago on Mon 03 Jun 2024 07:55:19 PM UTC.
No match for argument: xrdp
Error: Unable to find a match: xrdp
[root@ip-172-31-46-182 ec2-user]# yum list | grep epel
```

Checking epel in System
As yum unable to find xrdp as epel not installed.

Step 4: Install and Configure Remote Desktop Services

1. Install xrdp and tigervnc-server. If the installation fails, you'll need to enable the EPEL repository:

```
yum install xrdp tigervnc-server -y
```



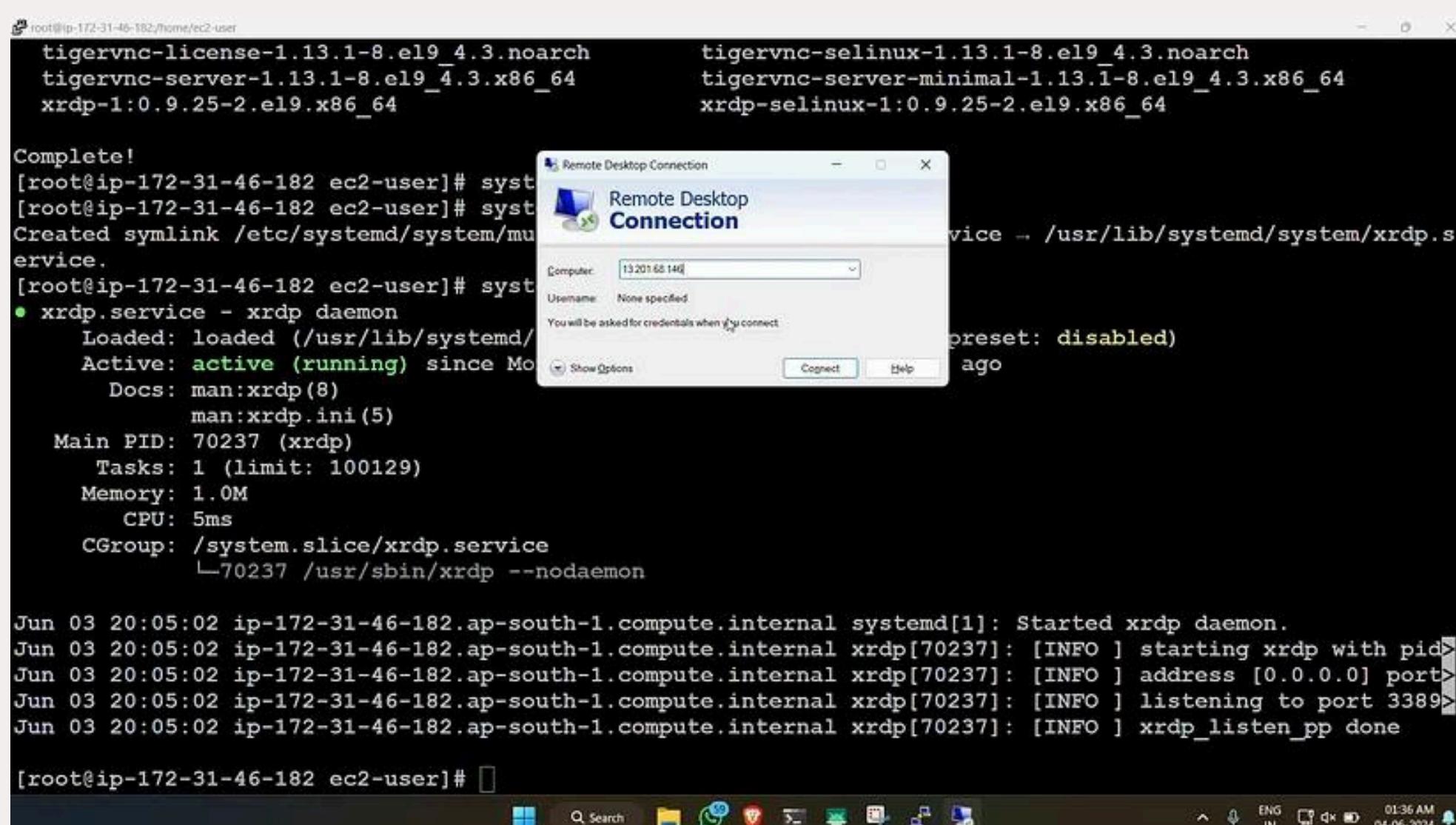
```
root@ip-172-31-46-182:~# yum install xrdp tigervnc-server -y
Installing : tigervnc-server-minimal-1.13.1-8.el9_4.3.x86_64 4/8
Installing : imlib2-1.7.4-1.el9.x86_64 5/8
Installing : xrdp-selinux-1:0.9.25-2.el9.x86_64 6/8
Running scriptlet: xrdp-selinux-1:0.9.25-2.el9.x86_64 6/8
Installing : xrdp-1:0.9.25-2.el9.x86_64 7/8
Running scriptlet: xrdp-1:0.9.25-2.el9.x86_64 7/8
Installing : tigervnc-server-1.13.1-8.el9_4.3.x86_64 8/8
Running scriptlet: tigervnc-server-1.13.1-8.el9_4.3.x86_64 8/8
Running scriptlet: xrdp-1:0.9.25-2.el9.x86_64 8/8
Running scriptlet: tigervnc-server-1.13.1-8.el9_4.3.x86_64 8/8
Verifying : imlib2-1.7.4-1.el9.x86_64 1/8
Verifying : xrdp-1:0.9.25-2.el9.x86_64 2/8
Verifying : xrdp-selinux-1:0.9.25-2.el9.x86_64 3/8
Verifying : dbus-x11-1:1.12.20-8.el9.x86_64 4/8
Verifying : tigervnc-license-1.13.1-8.el9_4.3.noarch 5/8
Verifying : tigervnc-selinux-1.13.1-8.el9_4.3.noarch 6/8
Verifying : tigervnc-server-1.13.1-8.el9_4.3.x86_64 7/8
Verifying : tigervnc-server-minimal-1.13.1-8.el9_4.3.x86_64 8/8
Installed products updated.

Installed:
  dbus-x11-1:1.12.20-8.el9.x86_64           imlib2-1.7.4-1.el9.x86_64
  tigervnc-license-1.13.1-8.el9_4.3.noarch    tigervnc-selinux-1.13.1-8.el9_4.3.noarch
  tigervnc-server-1.13.1-8.el9_4.3.x86_64     tigervnc-server-minimal-1.13.1-8.el9_4.3.x86_64
  xrdp-1:0.9.25-2.el9.x86_64                 xrdp-selinux-1:0.9.25-2.el9.x86_64

Complete!
[root@ip-172-31-46-182 ec2-user]# systemctl start xrdp
[root@ip-172-31-46-182 ec2-user]# systemctl enable xrdp
```

2. If you encounter issues just check for EPEL with following command :

```
yum list | grep epel
```



```
root@ip-172-31-46-182:~# yum list | grep epel
tigervnc-license-1.13.1-8.el9_4.3.noarch      tigervnc-selinux-1.13.1-8.el9_4.3.noarch
tigervnc-server-1.13.1-8.el9_4.3.x86_64       tigervnc-server-minimal-1.13.1-8.el9_4.3.x86_64
xrdp-1:0.9.25-2.el9.x86_64                   xrdp-selinux-1:0.9.25-2.el9.x86_64

Complete!
[root@ip-172-31-46-182 ec2-user]# systemctl start xrdp
[root@ip-172-31-46-182 ec2-user]# systemctl enable xrdp
Created symlink /etc/systemd/system/xdp.service → /usr/lib/systemd/system/xrdp.service.
[root@ip-172-31-46-182 ec2-user]# systemctl status xrdp.service
● xrdp.service - xrdp daemon
   Loaded: loaded (/usr/lib/systemd/system/xrdp.service)
   Active: active (running) since Mon Jun  3 20:05:02 2024
     Docs: man:xrdp(8)
           man:xrdp.ini(5)
     Main PID: 70237 (xrdp)
        Tasks: 1 (limit: 100129)
       Memory: 1.0M
          CPU: 5ms
         CGroup: /system.slice/xrdp.service
                  └─70237 /usr/sbin/xrdp --nodaemon

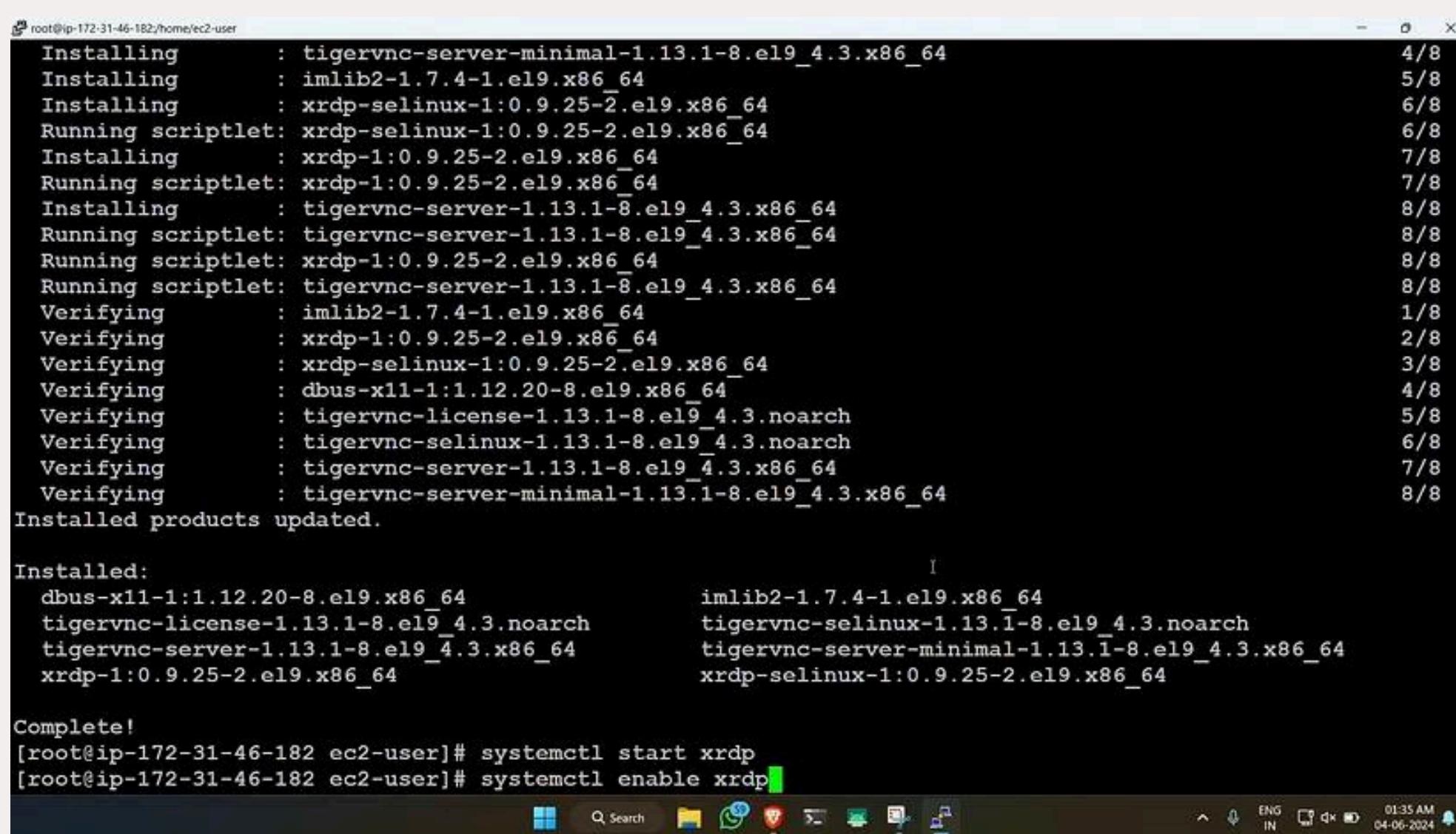
Jun  3 20:05:02 ip-172-31-46-182.ap-south-1.compute.internal systemd[1]: Started xrdp daemon.
Jun  3 20:05:02 ip-172-31-46-182.ap-south-1.compute.internal xrdp[70237]: [INFO ] starting xrdp with pid 70237
Jun  3 20:05:02 ip-172-31-46-182.ap-south-1.compute.internal xrdp[70237]: [INFO ] address [0.0.0.0] port 3389
Jun  3 20:05:02 ip-172-31-46-182.ap-south-1.compute.internal xrdp[70237]: [INFO ] listening to port 3389
Jun  3 20:05:02 ip-172-31-46-182.ap-south-1.compute.internal xrdp[70237]: [INFO ] xrdp_listen_pp done

[root@ip-172-31-46-182 ec2-user]#
```

Step 4: Install and Configure Remote Desktop Services

1. Install xrdp and tigervnc-server. If the installation fails, you'll need to enable the EPEL repository:

```
yum install xrdp tigervnc-server -y
```



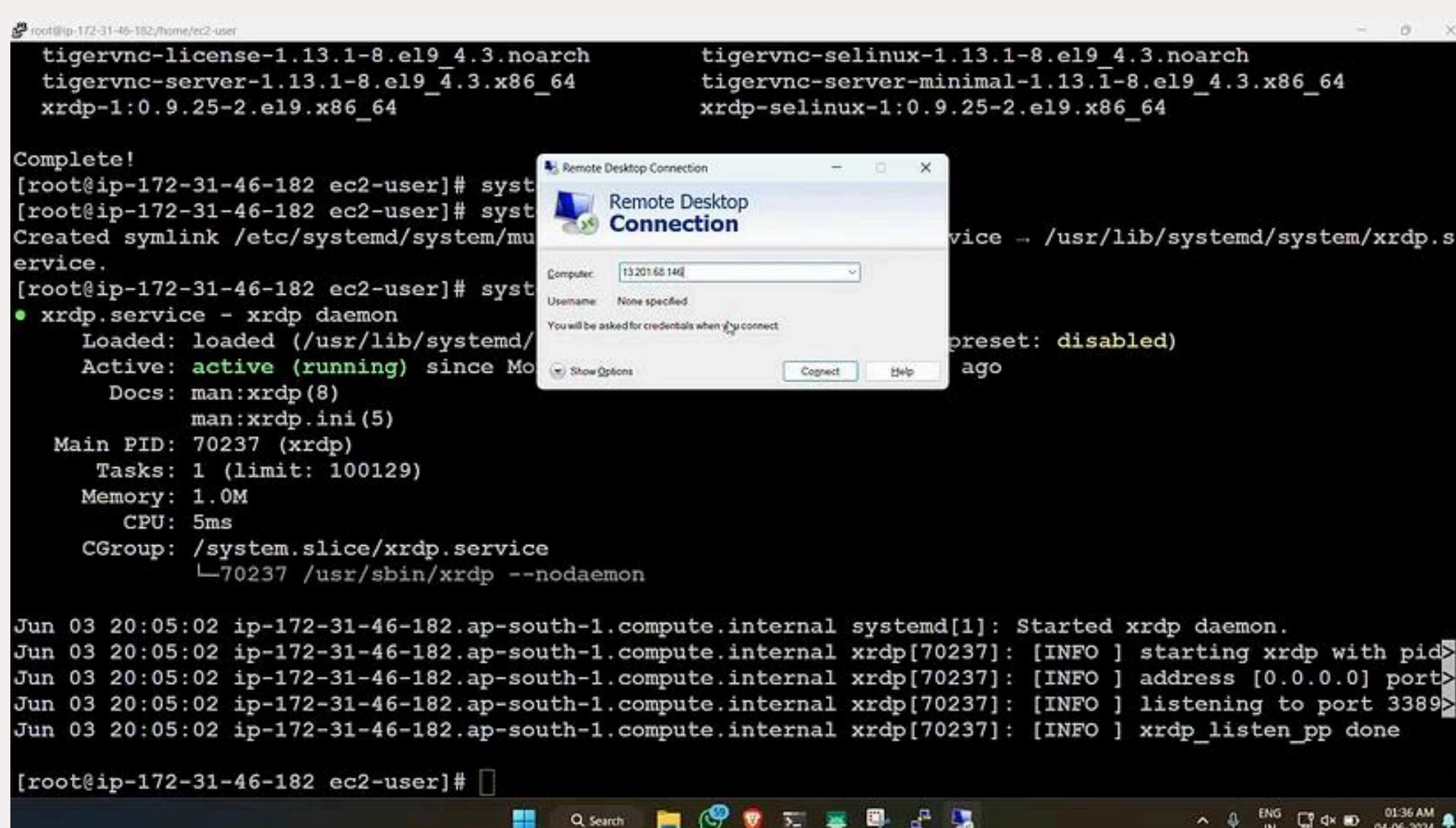
```
root@ip-172-31-46-182:~# yum install xrdp tigervnc-server -y
Installing : tigervnc-server-minimal-1.13.1-8.el9_4.3.x86_64 4/8
Installing : imlib2-1.7.4-1.el9.x86_64 5/8
Installing : xrdp-selinux-1:0.9.25-2.el9.x86_64 6/8
Running scriptlet: xrdp-selinux-1:0.9.25-2.el9.x86_64 6/8
Installing : xrdp-1:0.9.25-2.el9.x86_64 7/8
Running scriptlet: xrdp-1:0.9.25-2.el9.x86_64 7/8
Installing : tigervnc-server-1.13.1-8.el9_4.3.x86_64 8/8
Running scriptlet: tigervnc-server-1.13.1-8.el9_4.3.x86_64 8/8
Running scriptlet: xrdp-1:0.9.25-2.el9.x86_64 8/8
Running scriptlet: tigervnc-server-1.13.1-8.el9_4.3.x86_64 8/8
Verifying : imlib2-1.7.4-1.el9.x86_64 1/8
Verifying : xrdp-1:0.9.25-2.el9.x86_64 2/8
Verifying : xrdp-selinux-1:0.9.25-2.el9.x86_64 3/8
Verifying : dbus-x11-1:1.12.20-8.el9.x86_64 4/8
Verifying : tigervnc-license-1.13.1-8.el9_4.3.noarch 5/8
Verifying : tigervnc-selinux-1.13.1-8.el9_4.3.noarch 6/8
Verifying : tigervnc-server-1.13.1-8.el9_4.3.x86_64 7/8
Verifying : tigervnc-server-minimal-1.13.1-8.el9_4.3.x86_64 8/8
Installed products updated.

Installed:
  dbus-x11-1:1.12.20-8.el9.x86_64           imlib2-1.7.4-1.el9.x86_64
  tigervnc-license-1.13.1-8.el9_4.3.noarch    tigervnc-selinux-1.13.1-8.el9_4.3.noarch
  tigervnc-server-1.13.1-8.el9_4.3.x86_64     tigervnc-server-minimal-1.13.1-8.el9_4.3.x86_64
  xrdp-1:0.9.25-2.el9.x86_64                 xrdp-selinux-1:0.9.25-2.el9.x86_64

Complete!
[root@ip-172-31-46-182 ec2-user]# systemctl start xrdp
[root@ip-172-31-46-182 ec2-user]# systemctl enable xrdp
```

2. If you encounter issues just check for EPEL with following command :

```
yum list | grep epel
```



```
root@ip-172-31-46-182:~# yum list | grep epel
tigervnc-license-1.13.1-8.el9_4.3.noarch      tigervnc-selinux-1.13.1-8.el9_4.3.noarch
tigervnc-server-1.13.1-8.el9_4.3.x86_64       tigervnc-server-minimal-1.13.1-8.el9_4.3.x86_64
xrdp-1:0.9.25-2.el9.x86_64                   xrdp-selinux-1:0.9.25-2.el9.x86_64

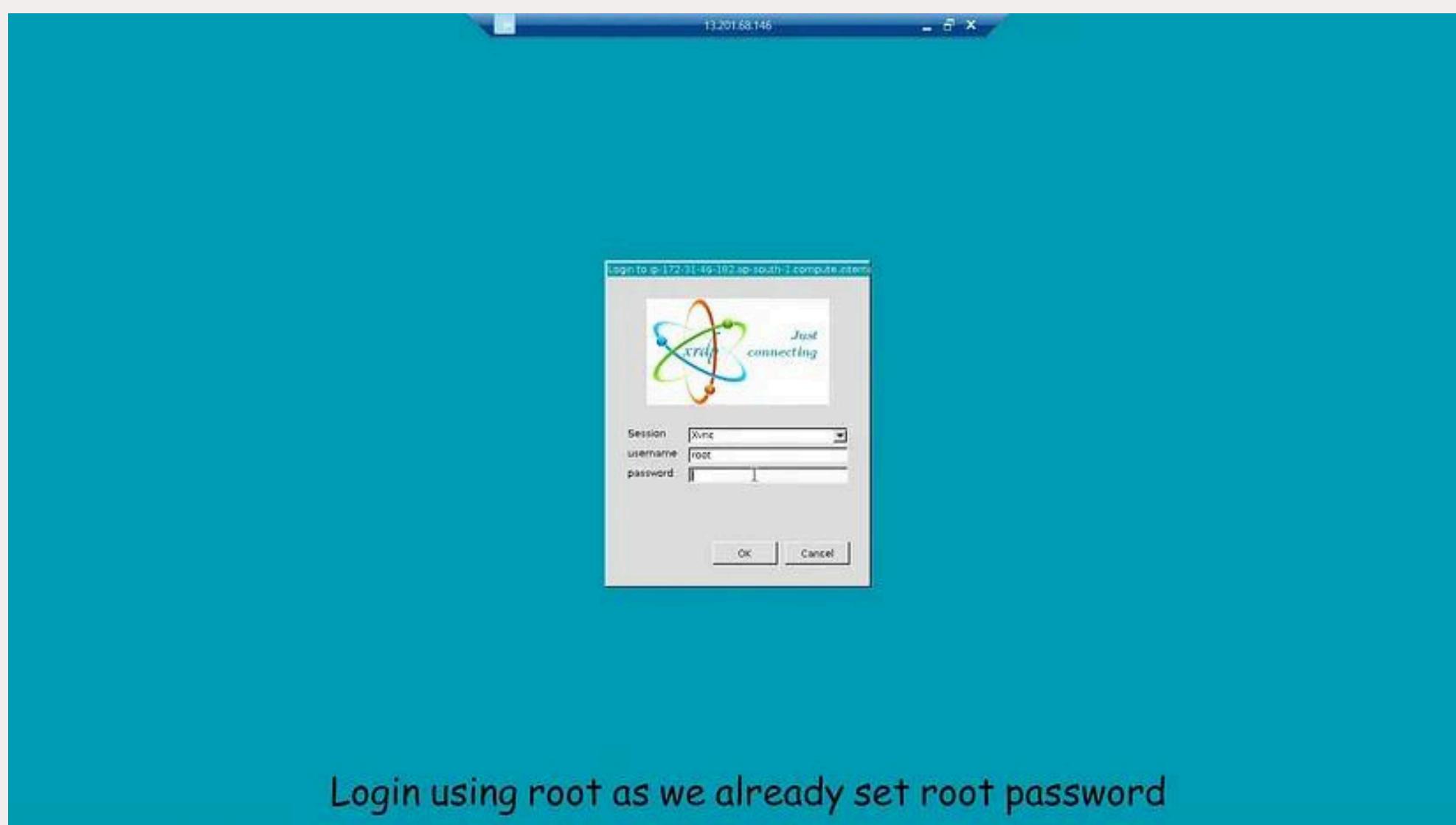
Complete!
[root@ip-172-31-46-182 ec2-user]# systemctl status xrdp
[root@ip-172-31-46-182 ec2-user]# systemctl status xrdp
Created symlink /etc/systemd/system/multi-user.target.wants/xrdp.service → /usr/lib/systemd/system/xrdp.service.
[root@ip-172-31-46-182 ec2-user]# systemctl status xrdp
● xrdp.service - xrdp daemon
   Loaded: loaded (/usr/lib/systemd/system/xrdp.service)
   Active: active (running) since Mon Jun  3 20:05:02 2024
     Docs: man:xrdp(8)
           man:xrdp.ini(5)
     Main PID: 70237 (xrdp)
        Tasks: 1 (limit: 100129)
       Memory: 1.0M
          CPU: 5ms
         CGroup: /system.slice/xrdp.service
                  └─70237 /usr/sbin/xrdp --nodaemon

Jun  3 20:05:02 ip-172-31-46-182.ap-south-1.compute.internal systemd[1]: Started xrdp daemon.
Jun  3 20:05:02 ip-172-31-46-182.ap-south-1.compute.internal xrdp[70237]: [INFO ] starting xrdp with pid 70237
Jun  3 20:05:02 ip-172-31-46-182.ap-south-1.compute.internal xrdp[70237]: [INFO ] address [0.0.0.0] port 3389
Jun  3 20:05:02 ip-172-31-46-182.ap-south-1.compute.internal xrdp[70237]: [INFO ] listening to port 3389
Jun  3 20:05:02 ip-172-31-46-182.ap-south-1.compute.internal xrdp[70237]: [INFO ] xrdp_listen_pp done

[root@ip-172-31-46-182 ec2-user]#
```

Step 5: Connect Using Remote Desktop Connection

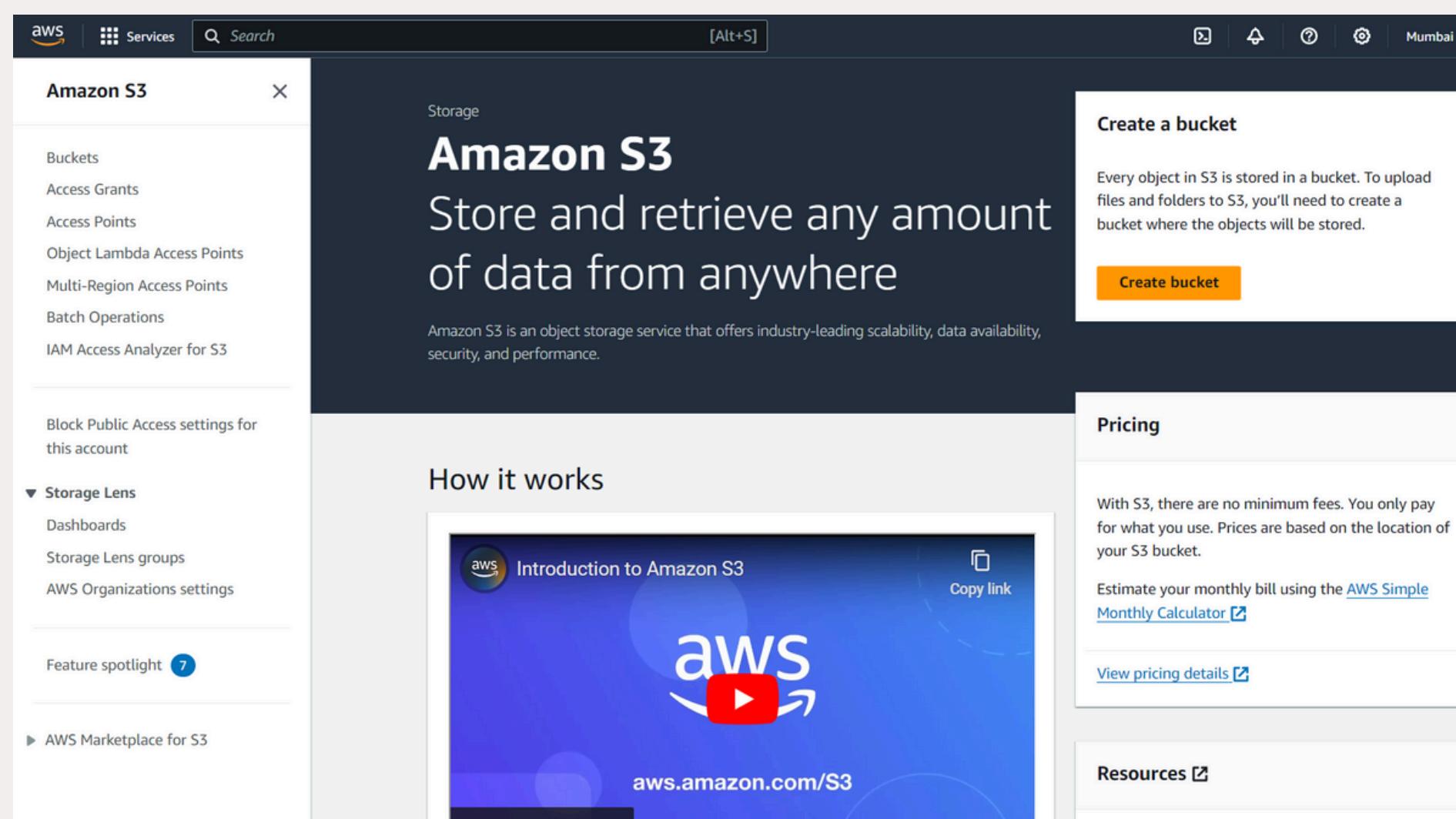
1. Open the Remote Desktop Connection application on your Windows machine.
2. Enter the public IP address of your EC2 instance.



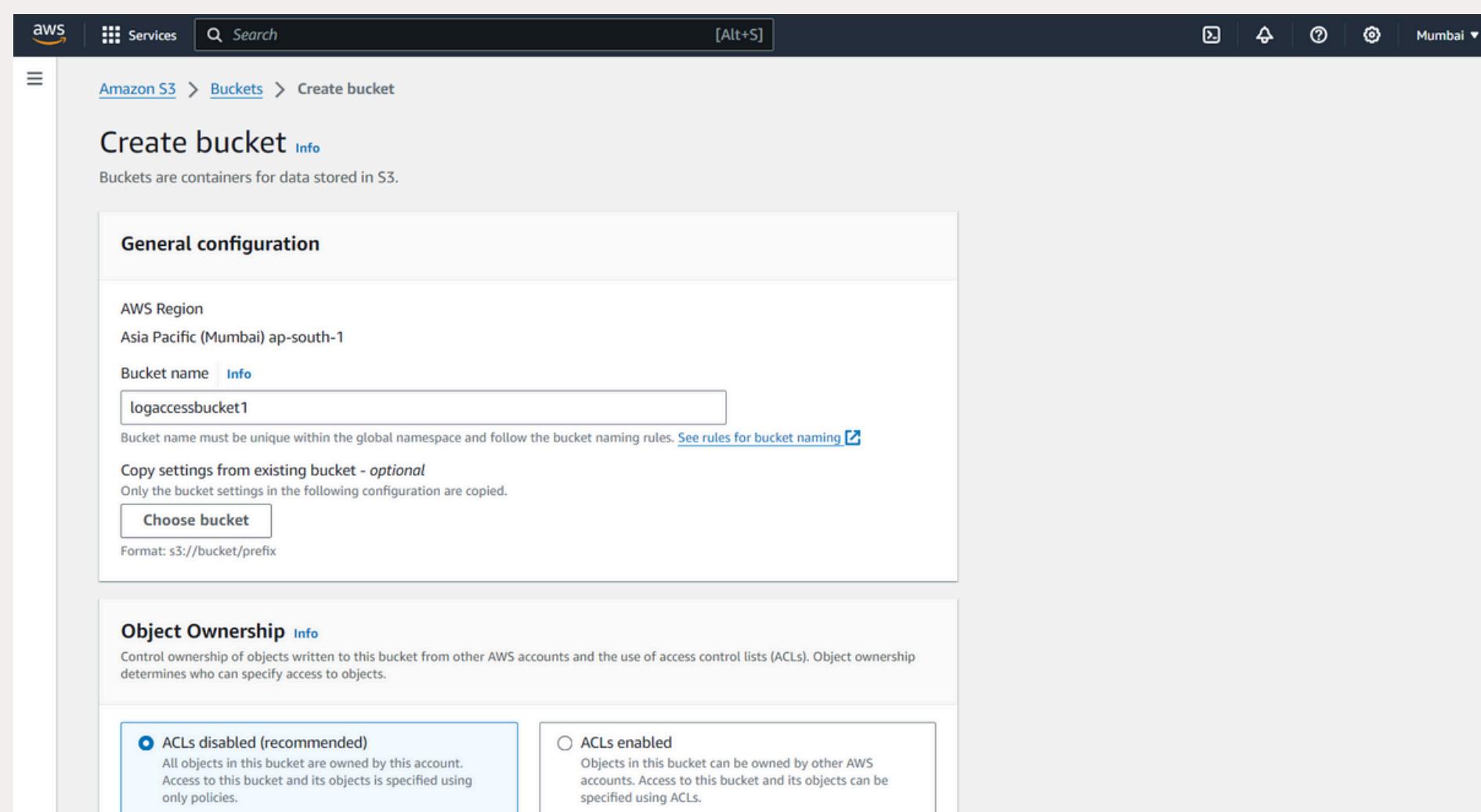
You should now be inside the GUI on your AWS EC2 instance running RHEL 9.

Task 3 :Access log from the cloud

1.To access logs from your S3 instance in AWS



2.Create the bucket in s3 with proper name



3.Optimizes the setting and information that given in the “object Ownership” part

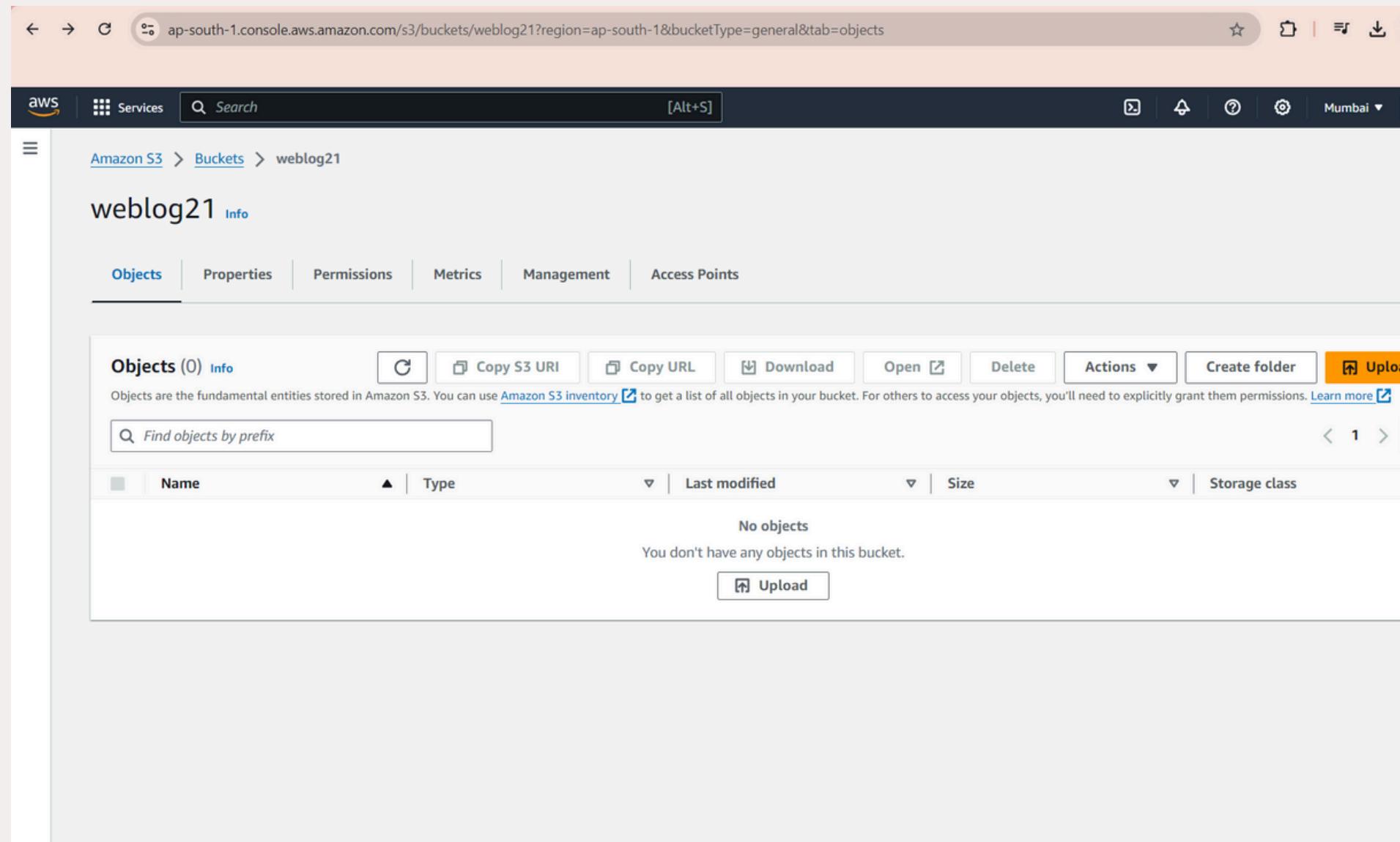
The screenshot shows the 'Create bucket' page in the AWS S3 console. In the 'General configuration' section, the 'Bucket name' field is set to 'weblog2'. Under 'Object Ownership', the 'ACLs disabled (recommended)' option is selected, indicating that all objects in the bucket are owned by the account.

4.We just Create the two buckets “weblog21” to store the data for client and “logaccessbucket1” to balance load of bucket 21.

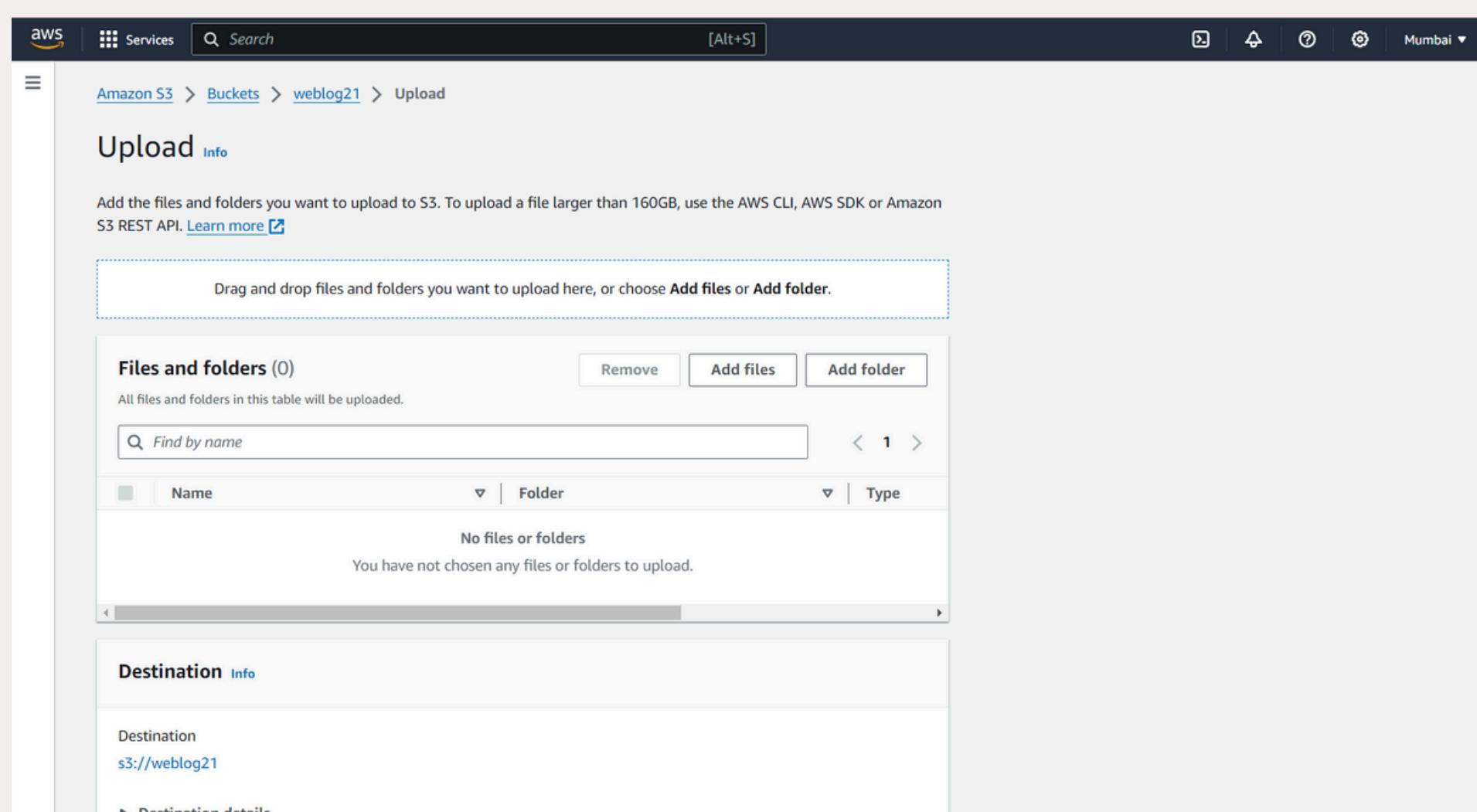
The screenshot shows the 'Buckets' page in the AWS S3 console. It lists two buckets: 'logaccessbucket1' and 'weblog21'. Both buckets were created in the 'Asia Pacific (Mumbai) ap-south-1' region on August 20, 2024, at 17:23:40 (UTC+05:30). The 'General purpose buckets' tab is selected.

Name	AWS Region	IAM Access Analyzer	Creation date
logaccessbucket1	Asia Pacific (Mumbai) ap-south-1	View analyzer for ap-south-1	August 20, 2024, 17:23:40 (UTC+05:30)
weblog21	Asia Pacific (Mumbai) ap-south-1	View analyzer for ap-south-1	August 20, 2024, 17:24:31 (UTC+05:30)

5.In the “ weblog21 ” we add some of the client data to check our service run or not



6.Add the file or folder



7. Set The permissions to access all the files by client

The screenshot shows the AWS S3 Bucket Properties page for a bucket named 'weblog21'. The top navigation bar includes the AWS logo, a search bar, and links for Services, Search, [Alt+S], and Mumbai. Below the navigation is a breadcrumb trail: Amazon S3 > Buckets > weblog21. The main content area has tabs for Objects, Properties (which is selected), Permissions, Metrics, Management, and Access Points. Under the Properties tab, there's a 'Bucket overview' section with details: AWS Region (Asia Pacific (Mumbai) ap-south-1), Amazon Resource Name (ARN) (arn:aws:s3:::weblog21), and Creation date (August 20, 2024, 17:24:31 (UTC+05:30)). Below this is a 'Bucket Versioning' section, which is currently disabled. There's also a note about Multi-factor authentication (MFA) delete, which is also disabled. At the bottom of the properties page is a 'Tags (0)' section.

8. Allow all the terms in server access logging

The screenshot shows the 'Server access logging' settings page. It features a 'Server access logging' section with a note that log requests can be checked via CloudWatch. The status is currently set to 'Disabled'. There is an 'Edit' button in the top right corner of this section.

9. Enable the server access logging setting for client

Amazon S3 > Buckets > weblog21 > Edit server access logging

Edit server access logging Info

The screenshot shows the 'Edit server access logging' dialog box. It contains a 'Server access logging' section with a note about log requests and a 'Learn more' link. Below this is a 'Server access logging' field with two radio button options: 'Disable' (which is selected) and 'Enable'. At the bottom right of the dialog are 'Cancel' and 'Save changes' buttons.

10.After all the Permissions we need to give access of “logaccessbucket1” for load balance.

Destination: s3://logaccessbucket1

Format: s3://<bucket>/<optional-prefix-with-path>

Destination Region: Asia Pacific (Mumbai) ap-south-1

Destination bucket name: logaccessbucket1

Destination prefix: -

11.All the traffic is managed by “logaccessbucket1” Bucket

Objects (2) Info

Name	Type	Last modified	Size	Storage class
2024-08-20-12-14-08-8993FBFC2B3A523	-	August 20, 2024, 17:44:09 (UTC+05:30)	633.0 B	Standard
2024-08-20-12-14-41-DF118CE9B463F554	-	August 20, 2024, 17:44:42 (UTC+05:30)	689.0 B	Standard

12.All the bucket is ready to work

Properties | Permissions | Versions

Object overview

Owner: d5cb7ed9e93d3a12dd6457e77dd590005308071eac1ad34c7c1037771f4b986d	S3 URI: s3://weblog21/webpage.html
AWS Region: Asia Pacific (Mumbai) ap-south-1	Amazon Resource Name (ARN): arn:aws:s3:::weblog21/webpage.html
Last modified: August 20, 2024, 17:25:50 (UTC+05:30)	Entity tag (Etag): Oed3741d07ebd0c6e41ba272c9a8d11f
Size: 1.6 KB	Object URL: https://weblog21.s3.ap-south-1.amazonaws.com/webpage.html
Type: html	
Key: webpage.html	

Login

Username:

Password:

Login

Task 4 : Create an event driven architecture such that when an audio file is uploaded in s3 it automatically gets converted to text using aws transcribe service.

The screenshot shows the Amazon Transcribe service homepage. The left sidebar contains navigation links: Get started, Real-time transcription, Transcription jobs, Custom language model, Custom vocabulary, Vocabulary filtering, Amazon Transcribe Call Analytics (Post-call Analytics, Category Management), and Amazon Transcribe Medical (Transcription jobs, Custom vocabulary). The main content area features the title "Amazon Transcribe Automatic Speech Recognition" and a sub-section "Machine learning". It includes a call-to-action button "Create a transcript" and a "Getting started" link. Below this, there's a section titled "Sample use cases" with two cards: "Call analytics" and "Media content search and monetization". A "Pricing (US)" section on the right provides information about pay-as-you-go pricing. The top right corner shows the location "Mumbai".

The screenshot shows the "Transcription jobs" page within the Amazon Transcribe service. The left sidebar is identical to the homepage. The main content area displays a table header for "Transcription jobs (0) Info" with columns for Name, Status, Language, Language settings, Model type, Model name, Created, and Expires. A search bar "Find job names" and a status filter "Status: All" are at the top of the table. Below the table, a message says "Empty resources No resources to display" and a "Create job" button is visible. The top right corner shows the location "Mumbai".

1. Access AWS Management Console

The screenshot shows the AWS Management Console with the Amazon S3 service selected. The left sidebar lists various S3 management options like Buckets, Access Grants, and Storage Lens. The main content area displays the Amazon S3 homepage with a large heading 'Amazon S3' and subtext 'Store and retrieve any amount of data from anywhere'. It includes a video thumbnail titled 'Introduction to Amazon S3' and a 'Create a bucket' button. To the right, there are sections for 'Pricing' and 'Resources'.

2. Set Up Amazon S3 Bucket:

The screenshot shows the 'Create bucket' wizard. The first step, 'General configuration', is completed. It shows the AWS Region set to 'Asia Pacific (Mumbai) ap-south-1' and the Bucket name set to 'audio1'. The 'Bucket name' field has a note: 'Bucket name must be unique within the global namespace and follow the bucket naming rules. See rules for bucket naming.' Below this is a 'Copy settings from existing bucket - optional' section with a 'Choose bucket' button. The second step, 'Object Ownership', is shown with two options: 'ACLs disabled (recommended)' (selected) and 'ACLs enabled'. Both options have detailed descriptions below them.

3.Bucket Create successfully

The screenshot shows the AWS S3 Buckets page. At the top, a green banner displays the message "Successfully created bucket 'audiodata1232'". Below the banner, there's an "Account snapshot - updated every 24 hours" section and a "View Storage Lens dashboard" button. The main area shows two tabs: "General purpose buckets" (selected) and "Directory buckets". Under "General purpose buckets", there's a table with one row for "audiodata1232". The table columns include Name, AWS Region, IAM Access Analyzer, and Creation date. The bucket details are as follows:

Name	AWS Region	IAM Access Analyzer	Creation date
audiodata1232	Asia Pacific (Mumbai) ap-south-1	View analyzer for ap-south-1	August 20, 2024, 17:57:16 (UTC+05:30)

4.Upload the all the recording that we have to convert into text.

The screenshot shows the AWS S3 Bucket details page for "audiodata1232". The top navigation bar includes "Amazon S3 > Buckets > audiodata1232". The main content area has tabs for "Objects", "Properties", "Permissions", "Metrics", "Management", and "Access Points". The "Objects" tab is selected, showing a table with one row: "No objects". A note below the table states, "You don't have any objects in this bucket." There is a prominent "Upload" button at the bottom of the table.

5.Upload audio file

The screenshot shows the AWS S3 Upload page for "audiodata1232". The top navigation bar includes "Amazon S3 > Buckets > audiodata1232 > Upload". The main content area has tabs for "Upload" (selected), "Properties", "Permissions", "Metrics", "Management", and "Access Points". The "Upload" tab contains instructions: "Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. [Learn more](#)". Below this is a large dashed blue box labeled "Drag and drop files and folders you want to upload here, or choose Add files or Add folder." A "Files and folders (0)" table is shown, with a note: "All files and folders in this table will be uploaded." The table has columns for "Name" and "Type". A note below the table says, "No files or folders" and "You have not chosen any files or folders to upload." The "Destination" section shows the destination as "s3://audiodata1232" and includes a "Destination details" link.

6.Successfully Uploaded

The screenshot shows the AWS S3 'Upload' interface. At the top, it says 'Amazon S3 > Buckets > audiodata1232 > Upload'. Below that is a section titled 'Upload' with a 'Info' link. A note says: 'Add the files and folders you want to upload to S3. To upload a file larger than 160GB, use the AWS CLI, AWS SDK or Amazon S3 REST API. [Learn more](#)'.

A large dashed box area is labeled 'Drag and drop files and folders you want to upload here, or choose Add files or Add folder.'

Below this is a table titled 'Files and folders (1 Total, 2.1 MB)'. It lists one item: 'WhatsApp Audio 2024-08-15 at 19.01.27_3...' with a size of '2.1 MB' and a type of 'audio/mpeg'. There are 'Remove', 'Add files', and 'Add folder' buttons above the table.

Search bar: 'Find by name'

Table headers: 'Name' and 'Type'.

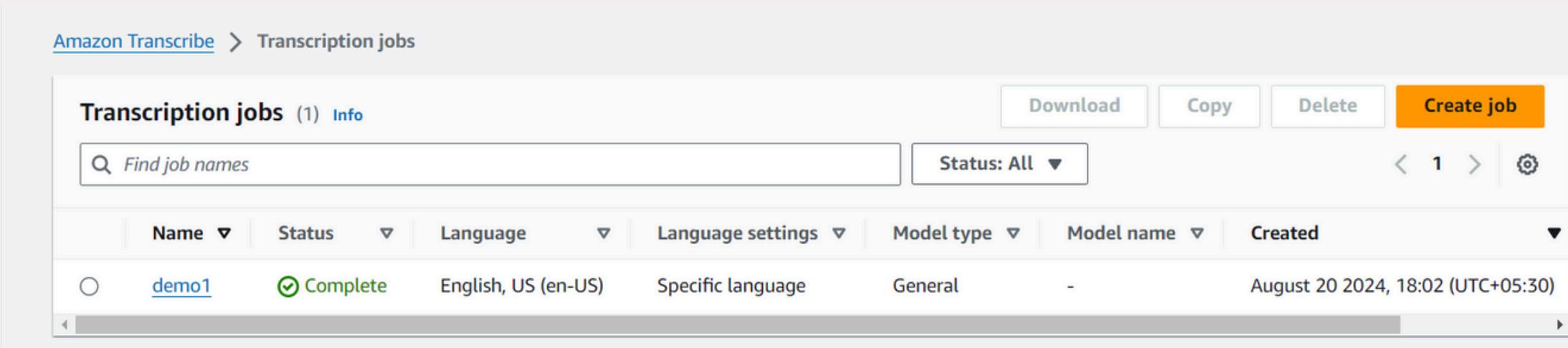
Destination section: 'Destination' set to 's3://audiodata1232'. 'Destination details' section notes: 'Bucket settings that impact new objects stored in the specified destination.'

7.Upload s3 bucket url into aws transcribe service

The screenshot shows the 'Create transcription job' interface in the AWS Transcribe service. The left sidebar has links for 'Get started', 'Real-time transcription', 'Transcription jobs', 'Custom language model', 'Custom vocabulary', 'Vocabulary filtering', 'Amazon Transcribe Call Analytics', 'Post-call Analytics', 'Category Management', 'Amazon Transcribe Medical', 'Transcription jobs', and 'Custom vocabulary'. The main panel shows 'Step 1 Specify job details' and 'Step 2 Configure job - optional'. Under Step 1, the 'Job settings' section includes a 'Name' field set to 'demo1'. The 'Language settings' section has a 'Specific language' radio button selected, with a note: 'If you know the language spoken in your source audio, choose this option to get the most accurate results. The options available for additional processing vary between languages.' Other options include 'Automatic language identification' and 'Automatic multiple languages identification'. The 'Language' dropdown is set to 'English, US (en-US)'. The 'Model type' dropdown is set to 'Choose the type of model to use for the transcription job.'

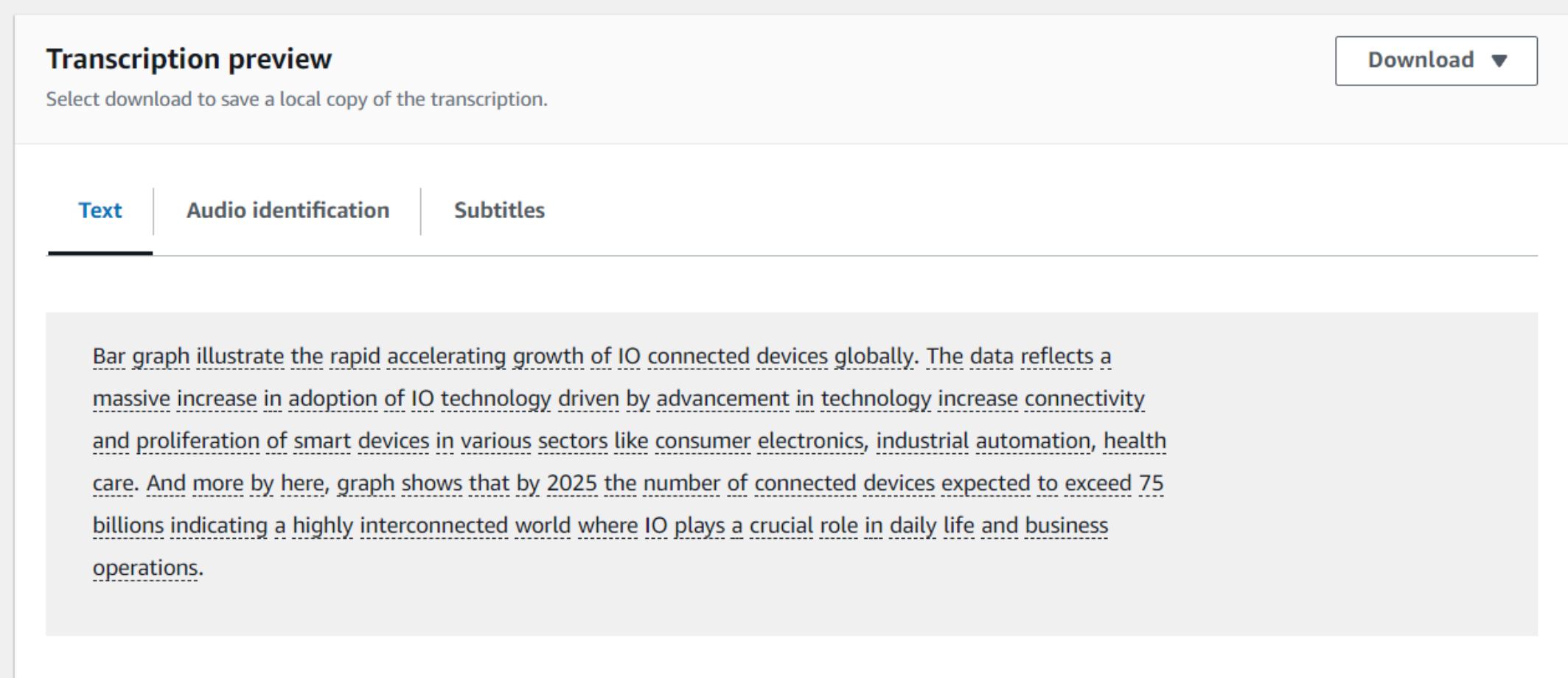
The screenshot shows the 'Input data' configuration screen. It has a 'Input file location on S3' section with a note: 'Specify an input audio or video file in Amazon S3.' An 'S3 URI' input field contains the value 's3://audiodata1232/WhatsApp Audio 2024-08-15 at 19.01.27_36b3!'. There are 'View' and 'Browse S3' buttons next to the input field. Below this, a note says 'Format: s3://bucket/prefix/file.extension.' and 'Valid file formats: MP3, MP4, WAV, FLAC, AMR, OGG, and WebM.'

8.Aws transcribe service is connect to s3 bucket



The screenshot shows the 'Transcription jobs' page in the Amazon Transcribe console. At the top, there's a search bar labeled 'Find job names' and a dropdown menu set to 'Status: All'. Below the header are several buttons: 'Download', 'Copy', 'Delete', and a prominent orange 'Create job' button. A navigation bar with arrows and a gear icon is at the top right. The main area is a table with columns: Name, Status, Language, Language settings, Model type, Model name, and Created. One row is visible, showing a job named 'demo1' in 'Complete' status, using 'English, US (en-US)' language, 'Specific language' settings, 'General' model type, and created on 'August 20 2024, 18:02 (UTC+05:30)'.

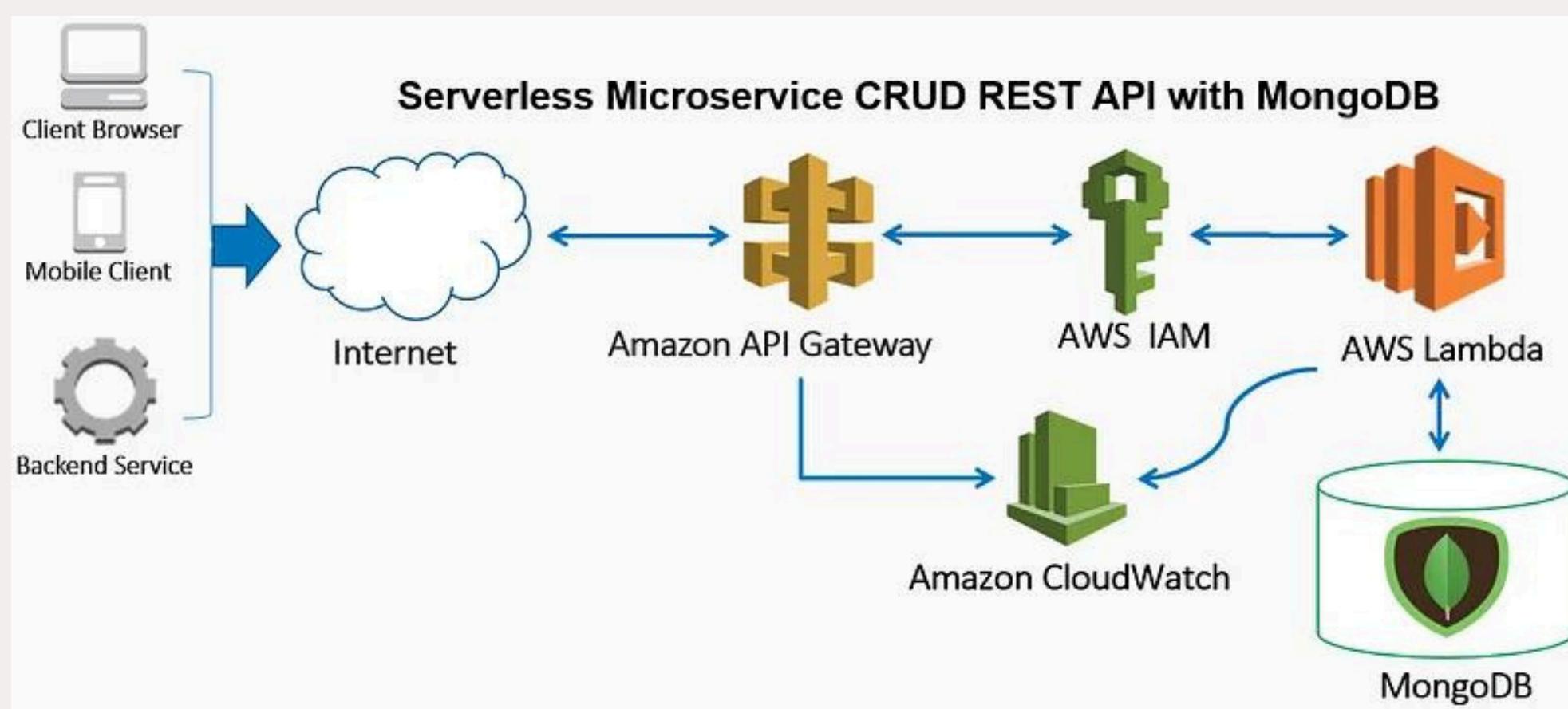
9.Let our Audio file converted into text



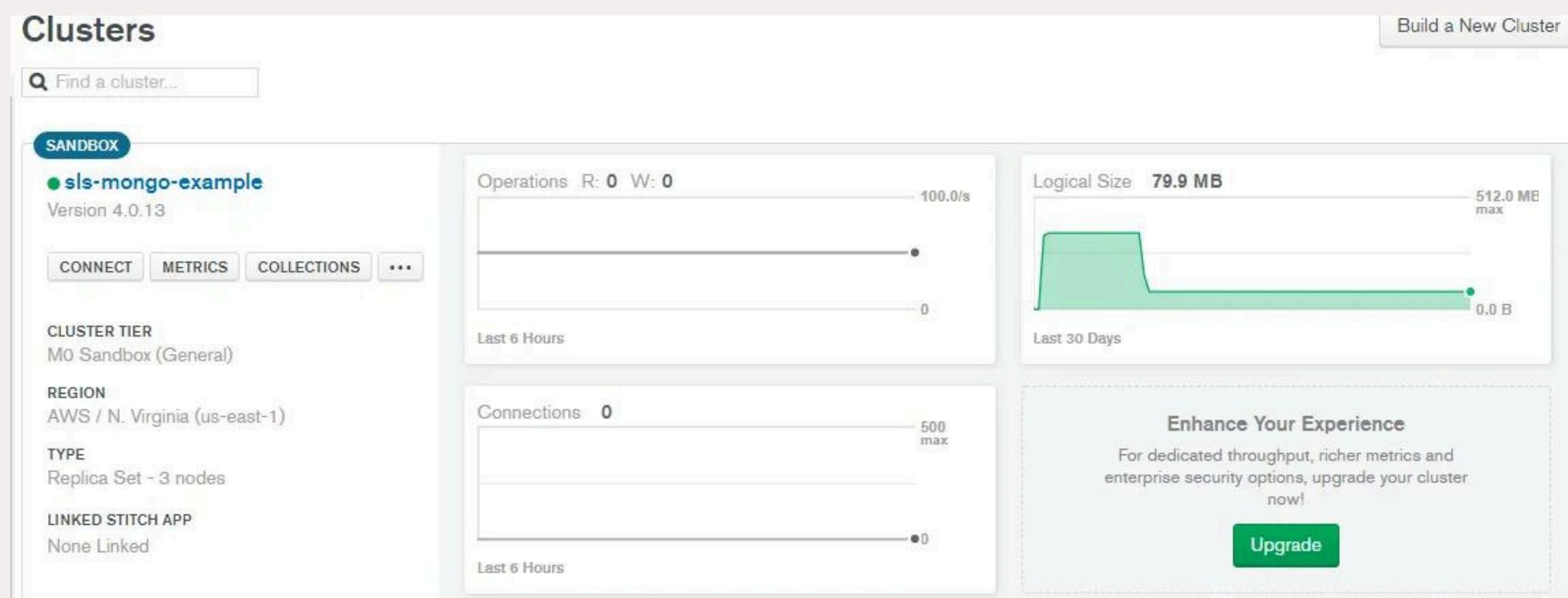
The screenshot shows the 'Transcription preview' page for the 'demo1' job. At the top, there's a 'Download' button. Below it, a note says 'Select download to save a local copy of the transcription.' There are three tabs: 'Text' (which is selected), 'Audio identification', and 'Subtitles'. The 'Text' tab displays a block of text: 'Bar graph illustrate the rapid accelerating growth of IO connected devices globally. The data reflects a massive increase in adoption of IO technology driven by advancement in technology increase connectivity and proliferation of smart devices in various sectors like consumer electronics, industrial automation, health care. And more by here, graph shows that by 2025 the number of connected devices expected to exceed 75 billions indicating a highly interconnected world where IO plays a crucial role in daily life and business operations.'

Task 5 : Connect python to mongodb service of aws using lambda

In this post we will build a Serverless Microservice that exposes create, read, update, delete (CRUD) operations[I have given sample code for read] on a fully managed MongoDB NoSQL database.

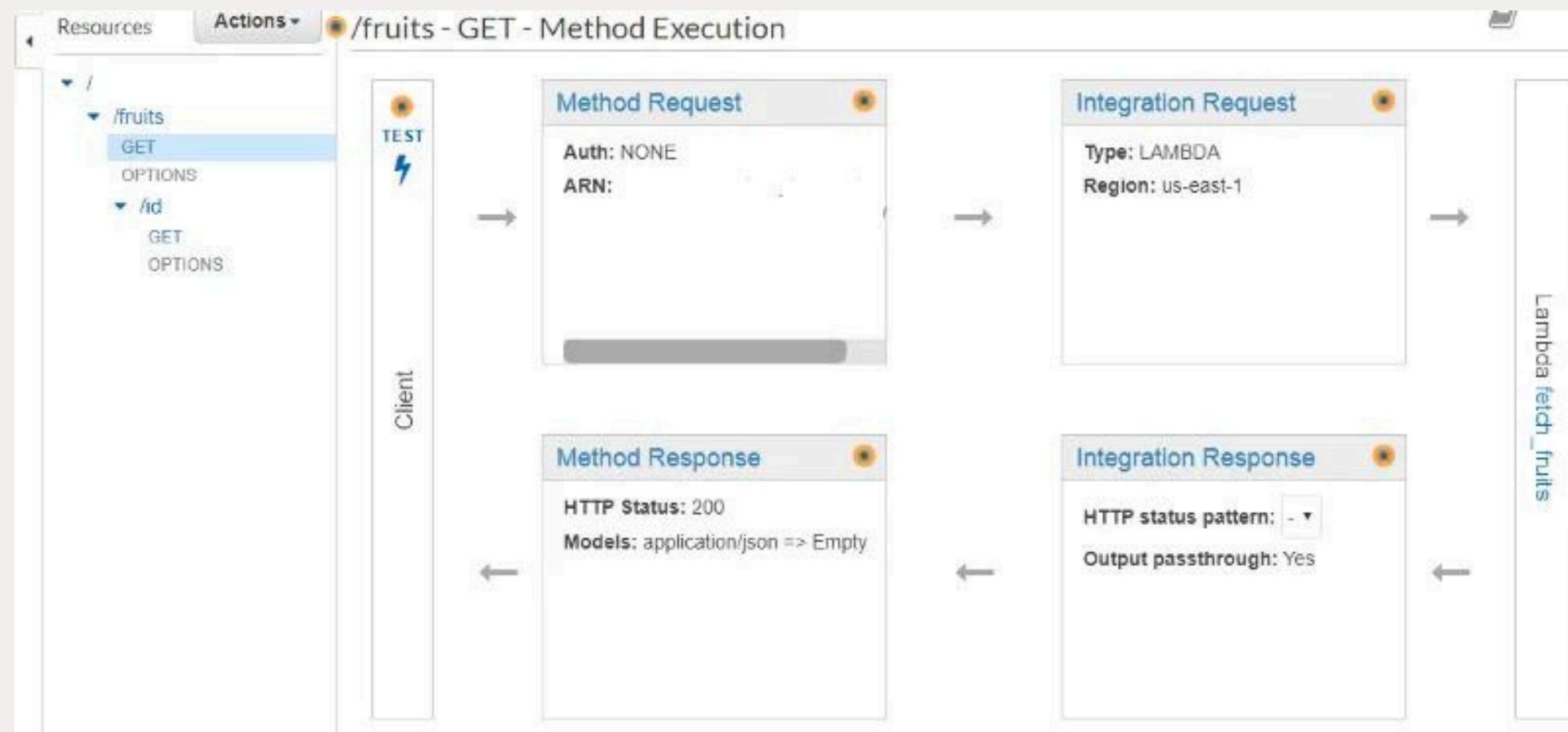


MongoDB Atlas is what we will be using as it has a free tier with no credit card, and is a fully managed service.

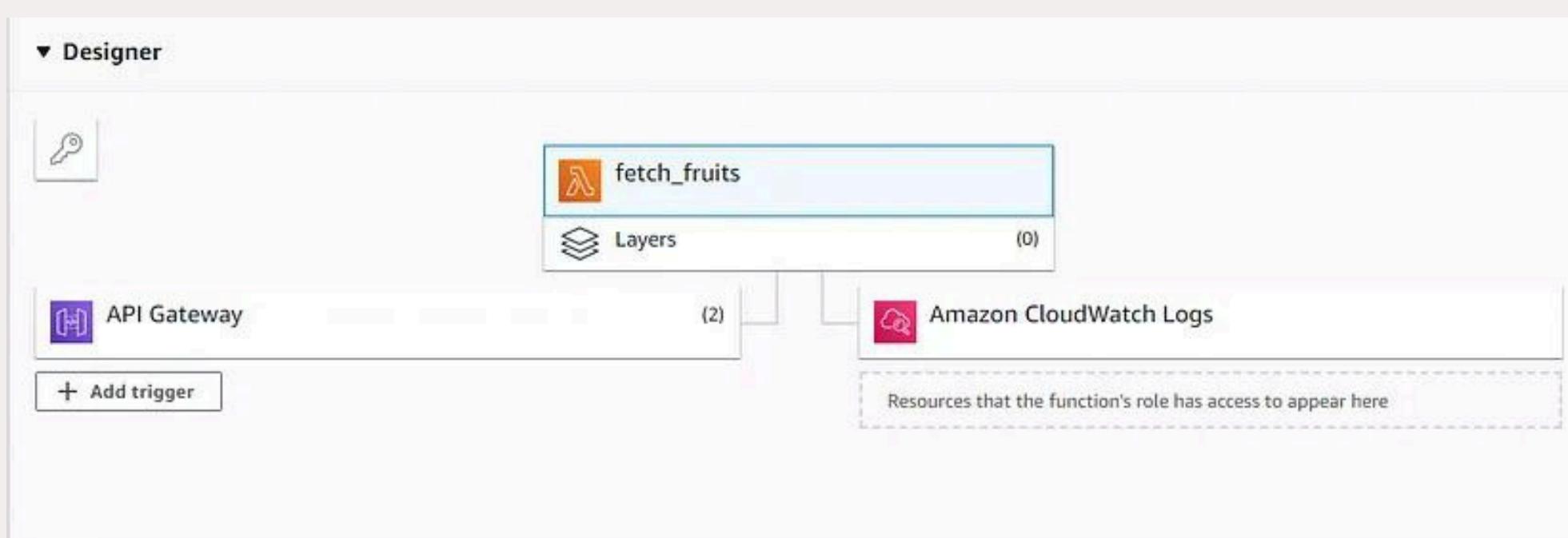


Database Creation

1. Open the Data Explorer. From the Atlas Clusters view, click your cluster's Collections button.
2. Click Load a Sample Database or create your new Db.



With this our DB is ready. Let's now move on to create our lambda function and API Gateway in AWS console. But read before that read through security recommendation for connection type we have chosen.



Code to Connect Mongodb Server

```
"use strict";
const MongoClient = require('mongodb').MongoClient;
const MONGODB_URI = process.env.MONGODB_URI; // or Atlas connection string
let cachedDb = null;

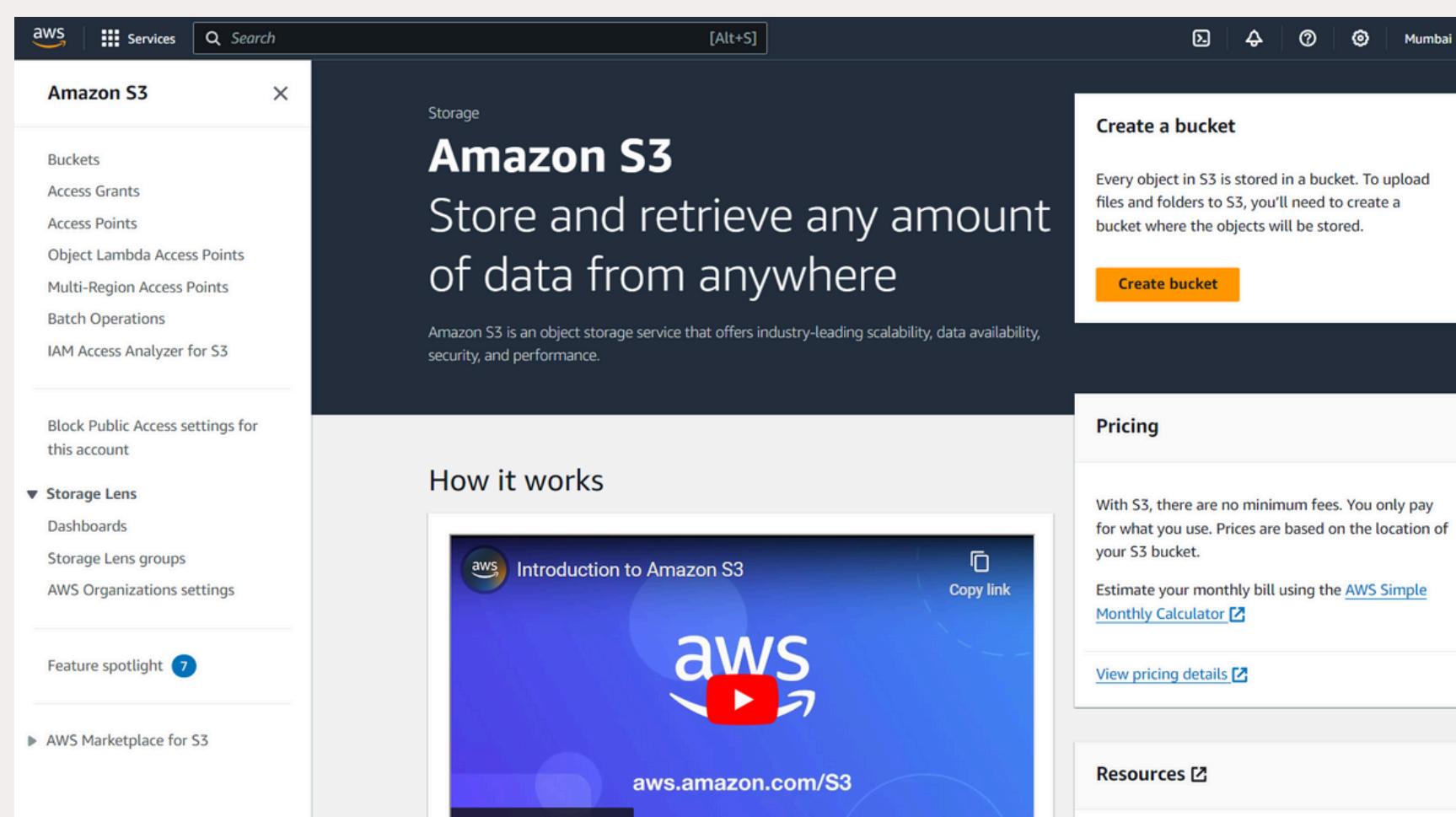
function connectToDatabase(uri) {
  console.log('=> connect to database');
  if (cachedDb) {
    console.log('=> using cached database instance');
    return Promise.resolve(cachedDb);
  }
  return MongoClient.connect(uri)
    .then(db => {
      cachedDb = db; //For mongo client before v3
      cachedDb = db.db("items"); //For mongo client v3,item is db i creted
      return cachedDb;
    });
}

function queryDatabase(db) {
  console.log('=> query database');
  return db.collection('fruits').find({}).toArray()
    .then(() => {
      return {
        statusCode: 200,
        body: 'success'
      };
    })
    .catch(err => {
      console.log('=> an error occurred: ', err);
      return {
        statusCode: 500,
        body: 'error'
      };
    });
}

module.exports.handler = (event, context, callback) => {
  context.callbackWaitsForEmptyEventLoop = false;
  console.log('event: ', event);
  connectToDatabase(MONGODB_URI)
    .then(db => queryDatabase(db))
    .then(result => {
      console.log('=> returning result: ', result);
      callback(null, result);
    })
    .catch(err => {
      console.log('=> an error occurred: ', err);
      callback(err);
    });
};
```

Task 6 : Uploading the any object to s3

1.To upload an object to an Amazon S3 bucket



2.Create the bucket in AWS S3 service

General configuration

AWS Region
Asia Pacific (Mumbai) ap-south-1

Bucket name [Info](#)
 Bucket name must be unique within the global namespace and follow the bucket naming rules. [See rules for bucket naming](#)

Copy settings from existing bucket - *optional*
Only the bucket settings in the following configuration are copied.

[Choose bucket](#)

Format: s3://bucket/prefix

3.Upload the data inside the bucket as required to client

The screenshot shows the AWS S3 console interface. At the top, there's a header with 'General purpose buckets (1) Info All AWS Regions'. Below the header is a search bar labeled 'Find buckets by name'. To the right of the search bar are buttons for 'Copy ARN', 'Empty', 'Delete', and 'Create bucket'. A table below the header lists one bucket: 'Name' (webobjects123), 'AWS Region' (Asia Pacific (Mumbai) ap-south-1), 'IAM Access Analyzer' (View analyzer for ap-south-1), and 'Creation date' (August 20, 2024, 18:57:01 (UTC+05:30)).

4.Upload the data inside the bucket as required to client

The screenshot shows the AWS S3 'Objects' list page for the 'webobjects123' bucket. At the top, there's a header with 'Objects (0) Info'. Below the header is a search bar labeled 'Find objects by prefix'. To the right of the search bar are buttons for 'Copy S3 URI', 'Copy URL', 'Download', 'Open', 'Delete', 'Actions', 'Create folder', and 'Upload'. A table below the header lists columns: 'Name', 'Type', 'Last modified', 'Size', and 'Storage class'. A message 'No objects' is displayed, followed by the sub-message 'You don't have any objects in this bucket.' A large 'Upload' button is centered at the bottom.

5.Upload Successfully

The screenshot shows the AWS S3 'Upload: status' page. At the top, a green banner displays a success message: 'Upload succeeded' and 'View details below.' Below the banner, the title 'Upload: status' is shown with a 'Close' button. A note states: 'The information below will no longer be available after you navigate away from this page.' The 'Summary' section shows the destination 's3://webobjects123' and the results: 'Succeeded' (1 file, 1.6 KB (100.00%)) and 'Failed' (0 files, 0 B (0%)). Below the summary, tabs for 'Files and folders' (selected) and 'Configuration' are visible. The 'Files and folders' section shows a table with one item: 'File Name' (webpage.htm...), 'Type' (text/html), 'Size' (1.6 KB), 'Status' (Succeeded), and 'Error' (-).

6. Publish the client work on server

Properties Permissions Versions

Object overview	
Owner	s3://webobjects123/webpage.html
d5cb7ed9e93d3a12dd6457e77dd590005308071eac1ad34c7c1037771f4b986d	
AWS Region	Amazon Resource Name (ARN)
Asia Pacific (Mumbai) ap-south-1	arn:aws:s3:::webobjects123/webpage.html
Last modified	Entity tag (Etag)
August 20, 2024, 18:57:56 (UTC+05:30)	0ed3741d07ebd0c6e41ba272c9a8d11f
Size	Object URL
1.6 KB	https://webobjects123.s3.ap-south-1.amazonaws.com/webpage.html
Type	
html	
Key	
webpage.html	

7. Running program

Login

Username

Password

Task 7 :Integrate Lambda with the powerful S3 service. Start by putting an object file in S3 that contains multiple email IDs.

Introduction:

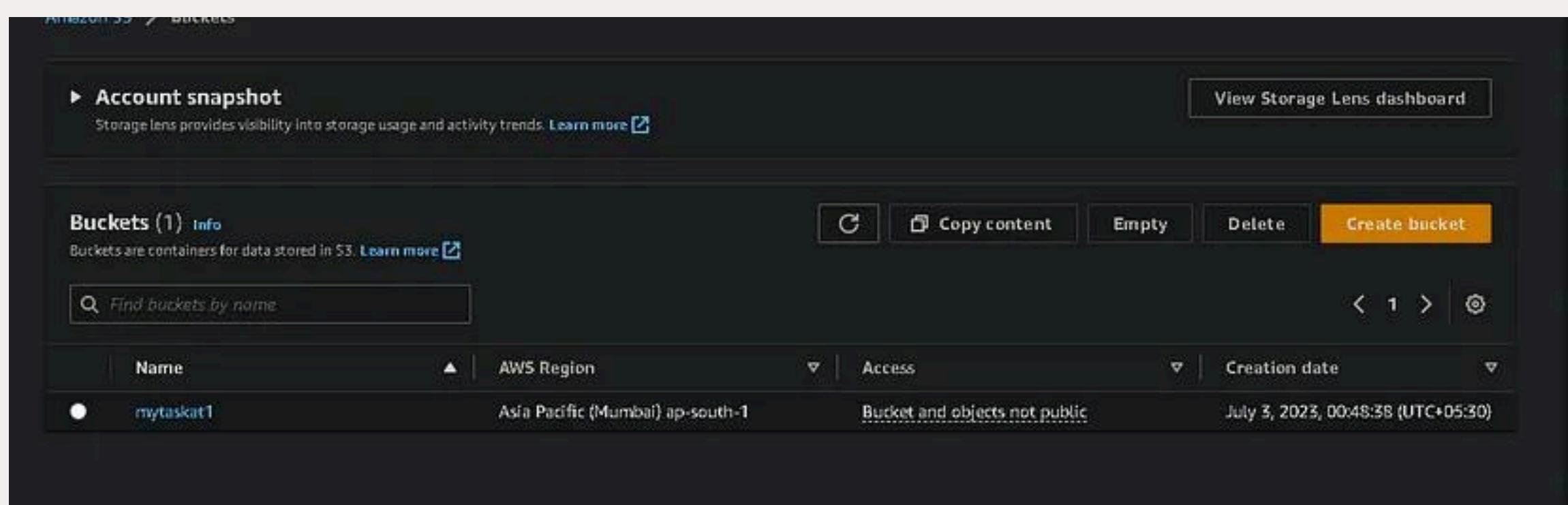
In this tutorial, we will explore how to integrate AWS Lambda with the powerful S3 service. We will start by uploading a file containing multiple email IDs to an S3 bucket. Then, utilizing the Boto3 library, we will retrieve these email IDs from the file. Finally, we will take it a step further by leveraging the Boto3 library to send an email using the Simple Email Service (SES) for each retrieved email ID. Let's get started!

Prerequisites: Before proceeding, make sure you have the following prerequisites:

1. An AWS account with access to AWS Lambda, S3, and SES.
2. Boto3 library installed in your Python environment.
3. Basic knowledge of AWS services, Python, and Lambda.

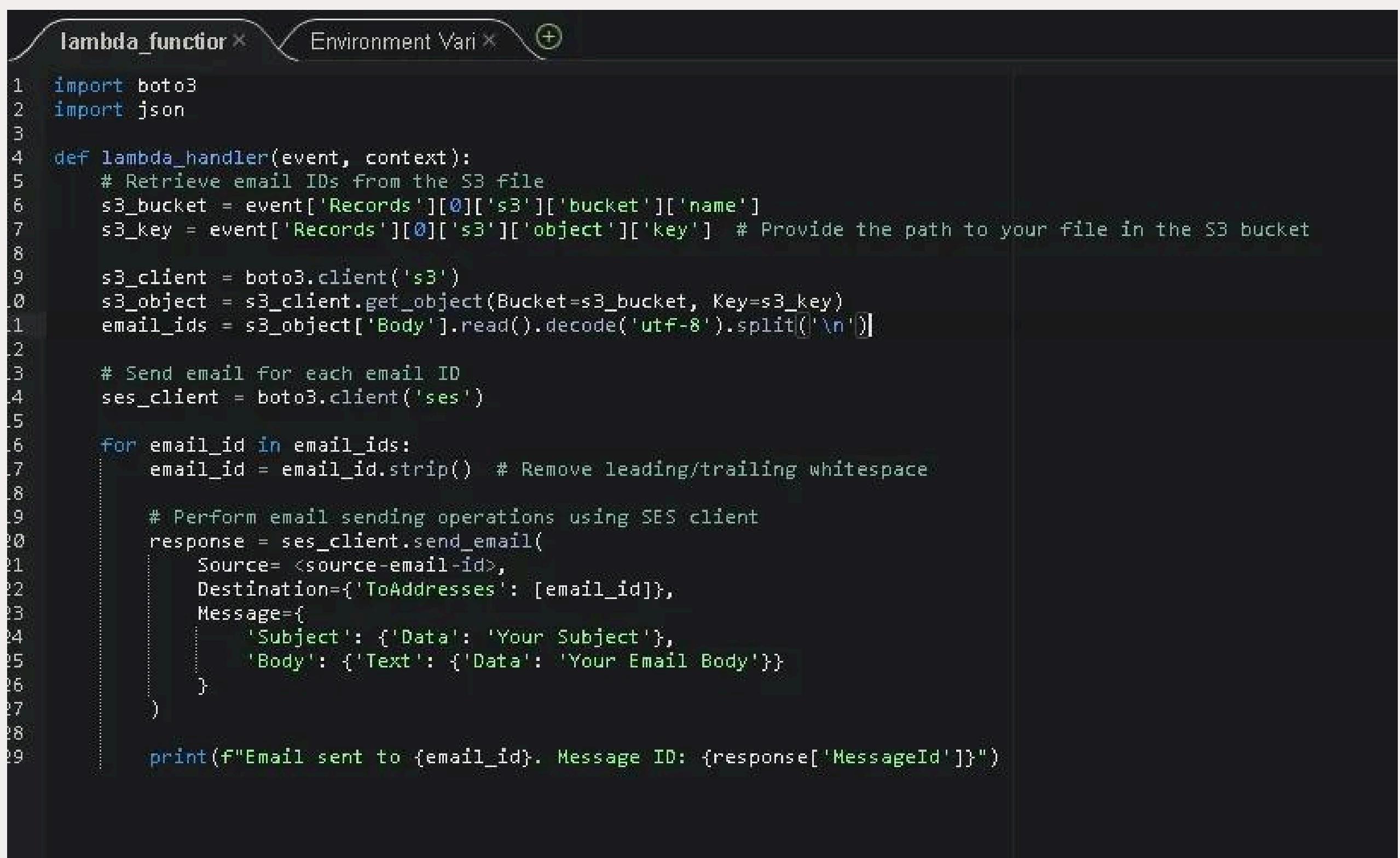
Step 1: Set up an S3 Bucket

1. Log in to your AWS Management Console and navigate to the S3 service.
2. Create a new bucket or choose an existing one to store your file.
3. Upload a file to the bucket containing multiple email IDs. Ensure the file format is supported (e.g., CSV, JSON).



Step 2: Create an AWS Lambda Function

1. Go to the AWS Lambda service in your AWS Management Console.
2. Click on “Create Function” and give your function a name and description.
3. Choose the runtime as “Python” and select an appropriate execution role with access to S3 and SES.
4. Once the function is created, you will be redirected to the function’s details page.



The screenshot shows the AWS Lambda function editor. The function name is "lambda_function". The code is written in Python and defines a lambda_handler function. This function retrieves email IDs from an S3 file and sends them as individual emails using the SES client. The code uses the boto3 library to interact with both S3 and SES services.

```
1 import boto3
2 import json
3
4 def lambda_handler(event, context):
5     # Retrieve email IDs from the S3 file
6     s3_bucket = event['Records'][0]['s3']['bucket']['name']
7     s3_key = event['Records'][0]['s3']['object']['key'] # Provide the path to your file in the S3 bucket
8
9     s3_client = boto3.client('s3')
10    s3_object = s3_client.get_object(Bucket=s3_bucket, Key=s3_key)
11    email_ids = s3_object['Body'].read().decode('utf-8').split('\n')
12
13    # Send email for each email ID
14    ses_client = boto3.client('ses')
15
16    for email_id in email_ids:
17        email_id = email_id.strip() # Remove leading/trailing whitespace
18
19        # Perform email sending operations using SES client
20        response = ses_client.send_email(
21            Source='<source-email-id>',
22            Destination={'ToAddresses': [email_id]},
23            Message={
24                'Subject': {'Data': 'Your Subject'},
25                'Body': {'Text': {'Data': 'Your Email Body'}}
26            }
27        )
28
29        print(f"Email sent to {email_id}. Message ID: {response['MessageId']}")
```

Step 3: Configure S3 Trigger

1. On the function details page, scroll down to the “Designer” section and click on “Add trigger.”
2. Select “S3” as the trigger type.
3. Configure the trigger settings, specifying the S3 bucket and event type (e.g., “Object Created”).
4. Save the trigger configuration.

Step 4: Retrieve Email IDs from the S3 File with the lambda function given above.

Step 5: Save the code and ensure the function is updated and send using Emails with SES and lambda function.