

# **AWS Enabled Employee Profile Management System**

## **Agenda**

- Lets create an application using a few services that have been discussed in this course.

## **Application**

- Employee profile of XYZ company – New employees input their information and upload photos. Existing employees can get their information.

## **AWS Setup**

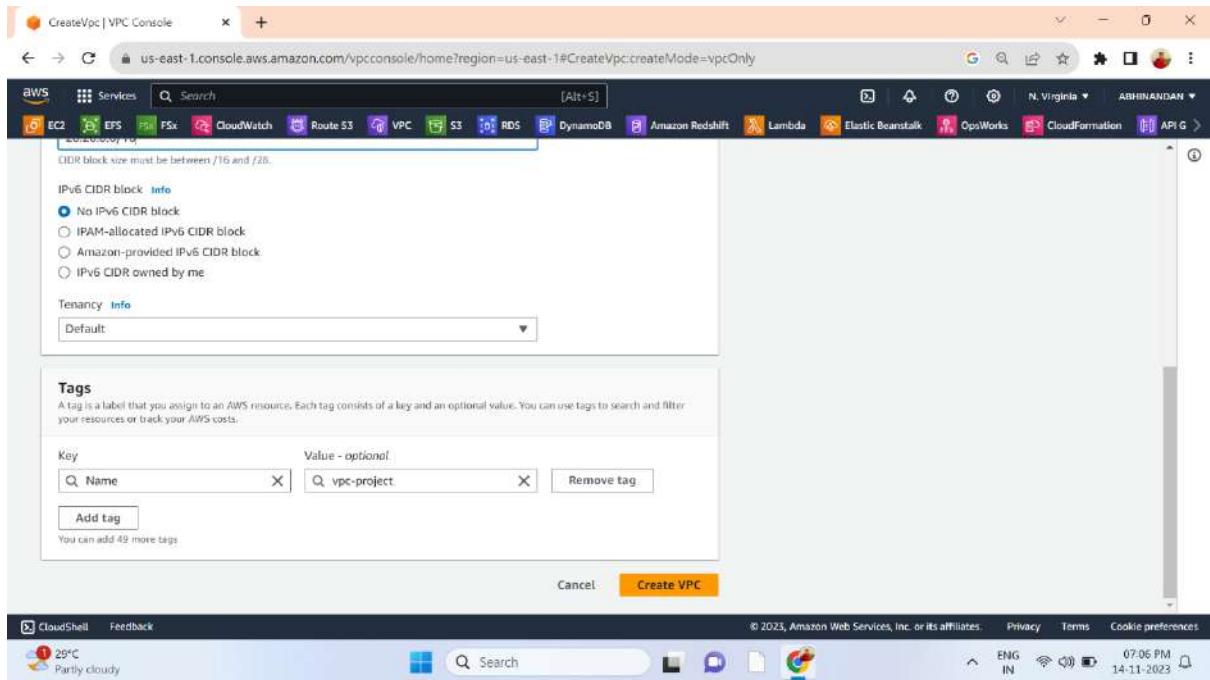
- Setup VPC for Load Balancer, Application EC2 instance and RDS Database - one private and two public subnets.
- Create Load balancer and Auto Scaling Group.
- Create RDS DB instance and DynamoDB table.
- S3 Bucket.
- Create instance profile that has to be attached to the EC2 instances being launched. Instance profile(IAM) should have permission to access RDS, DynamoDB and S3 bucket.

The screenshot shows the AWS VPC Console Home page. On the left, there's a sidebar with sections like 'Virtual private cloud' (Your VPCs, Subnets, Route tables, Internet gateways, Egress-only internet gateways, Carrier gateways, DHCP option sets, Elastic IPs, Managed prefix lists, Endpoints, Endpoint services, NAT gateways) and 'Create VPC' (Launch EC2 Instances). The main area displays 'Resources by Region' for the US East region, showing counts for VPCs (2), Subnets (17), Route Tables (3), Internet Gateways (2), NAT Gateways (0), and various other services like NAT Gateways, VPC Peering Connections, Network ACLs, Security Groups, and Customer Gateways. On the right, there are sections for 'Service Health', 'Settings' (Zones, Console Experiments), 'Additional Information' (VPC Documentation, All VPC Resources, Forums, Report an Issue), and 'AWS Network Manager'.

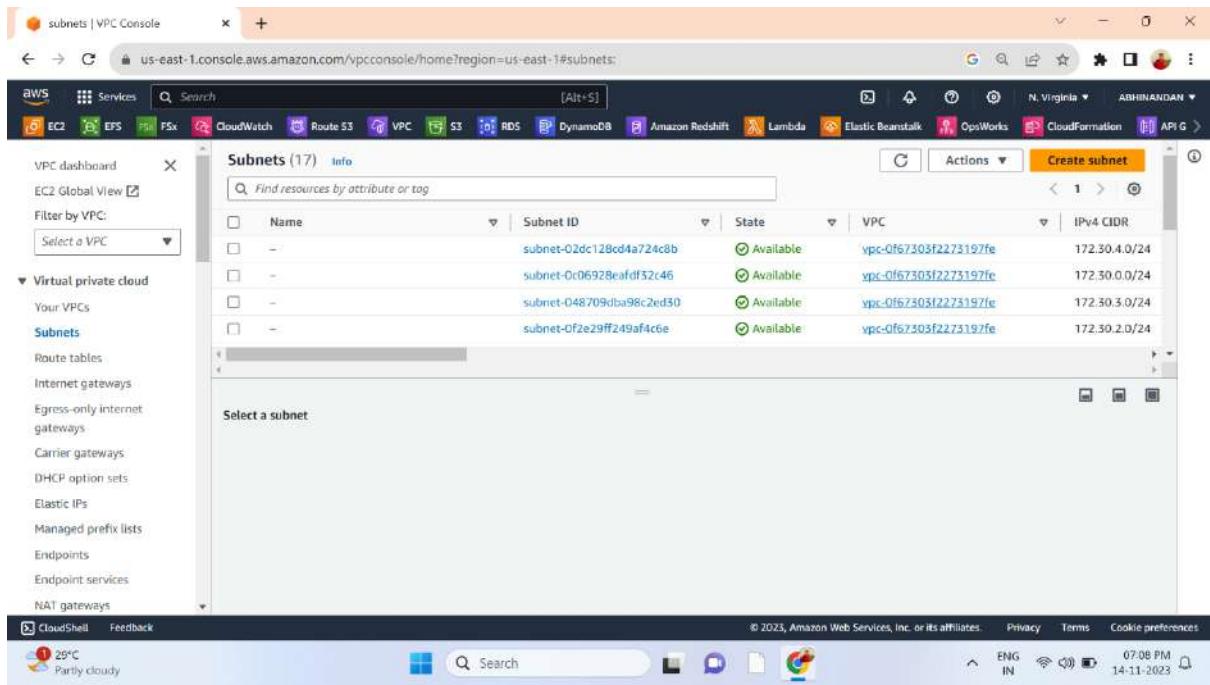
First, we will create a VPC. Click on Create VPC.

The screenshot shows the 'Create VPC' page. It starts with a brief description of what a VPC is. Below that is a 'VPC settings' section. Under 'Resources to create', the 'VPC only' option is selected. There's a note about creating a name tag. In the 'Name tag - optional' field, 'vpc-project' is typed. Under 'IPv4 CIDR block', the 'IPv4 CIDR manual input' option is selected, and the CIDR block '20.20.0.0/16' is specified. A note at the bottom says 'CIDR block size must be between /16 and /28.'

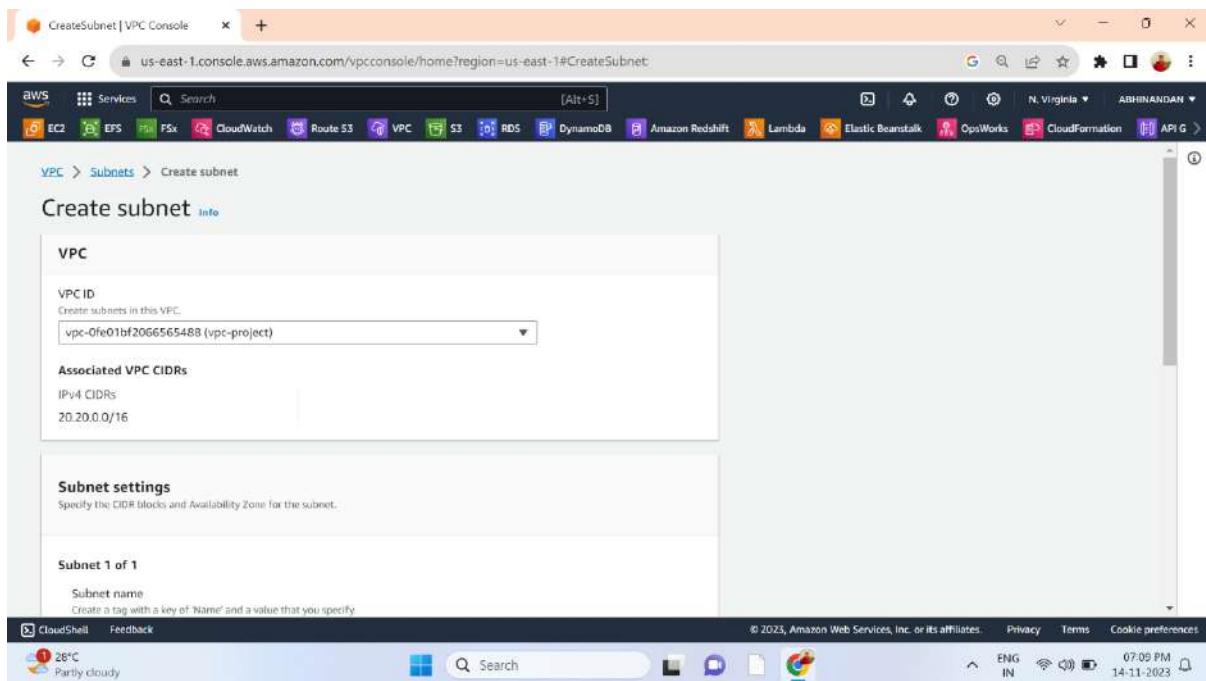
Choose option vpc only. give a name tag as vpc-project. IPV4 CIDR block-choose IPV4 CIDR manual input. IPV4 CIDR: 20.20.0.0/16



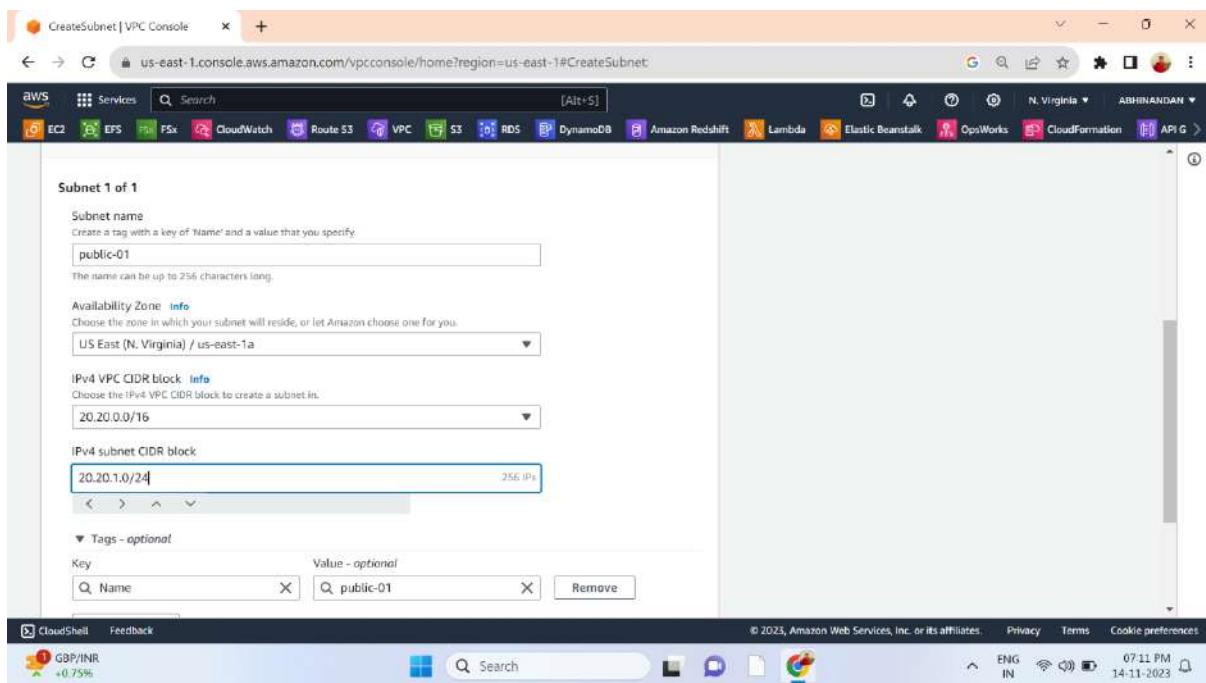
In the IPV6 CIDR block, choose No IPV4 CIDR block. Keep tenancy as the default. Click on Create VPC.



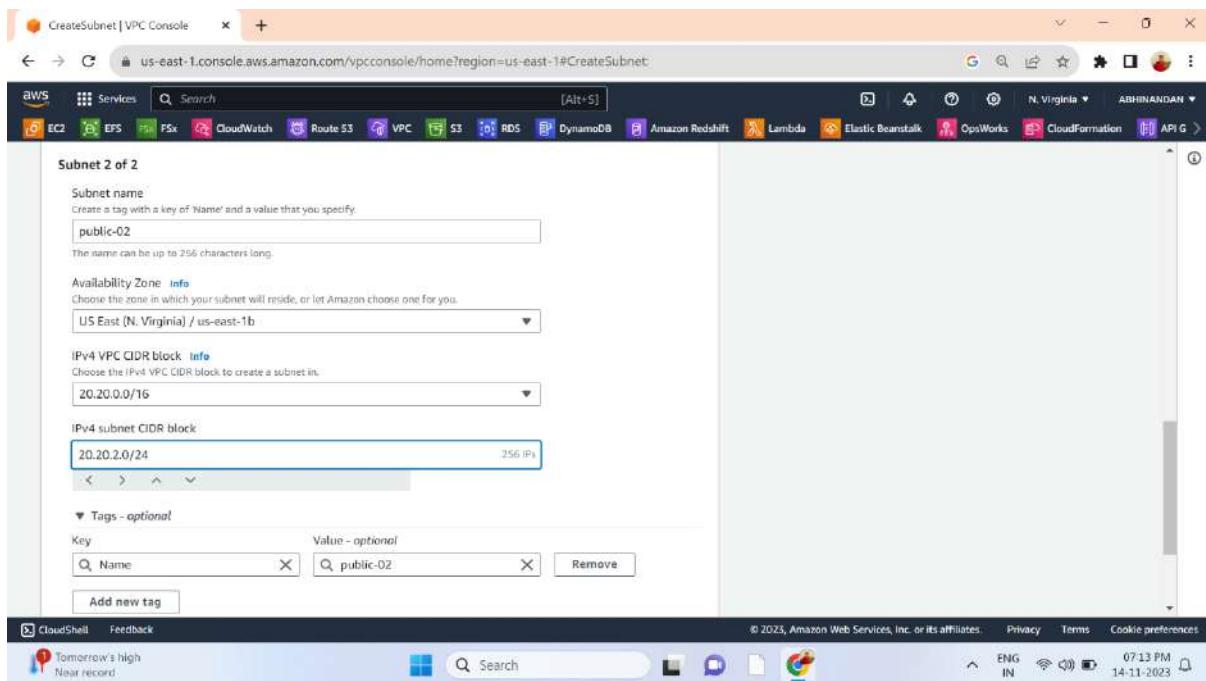
In the navigation bar, go to the subnets option. Click on Create Subnet.



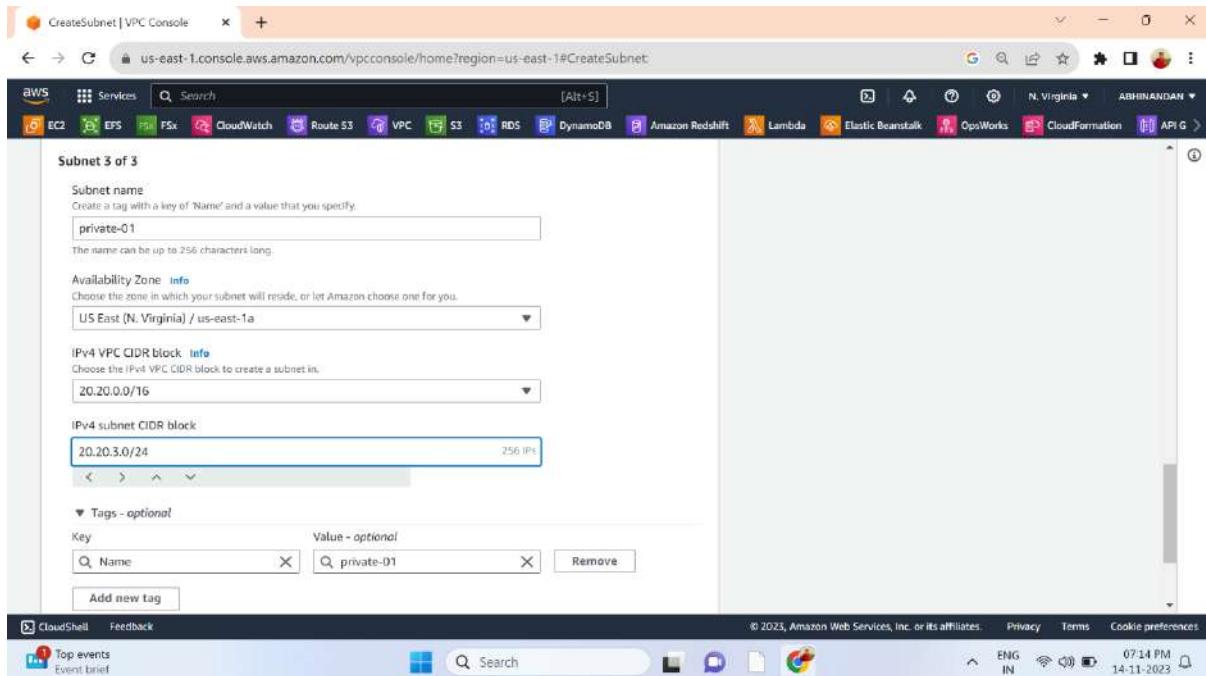
Choose our vpc that we named "vpc-project" in vpc - vpc id.



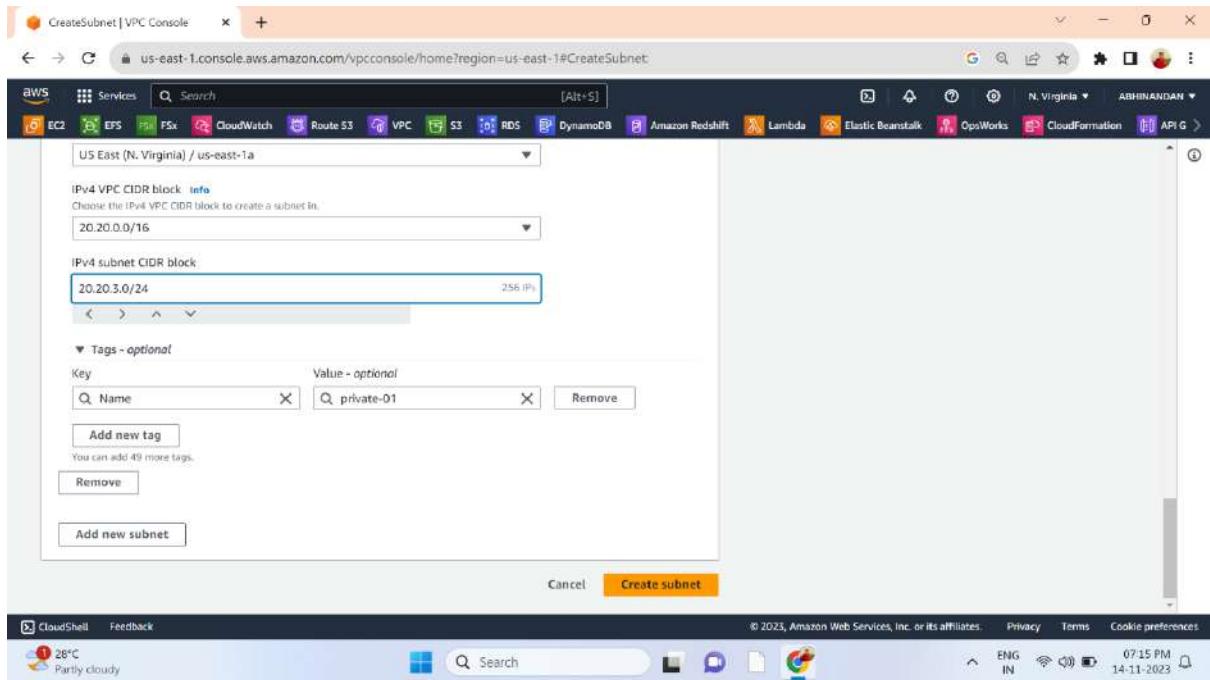
Give the subnet name "public-01". choose availability zone us-east (N.virginia)/us-east-1a. IPv4 CIDR block: 20.20.1.0/24



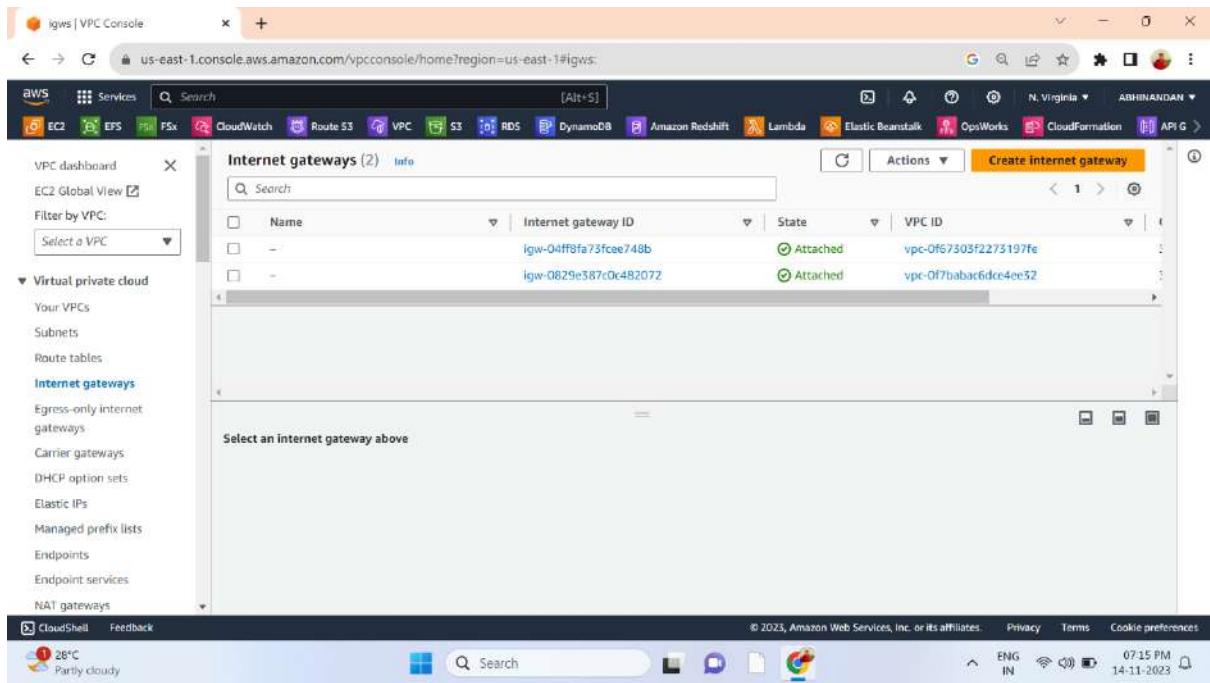
Here, actually, we are creating three subnets. Again, down-click on Create Subnet. subnet 2 of 2. subnet name "public-02" availability zone: choose us-east (N. Virginia)/us-east-1b. IPV4 CIDR block: 20.20.2.0/24



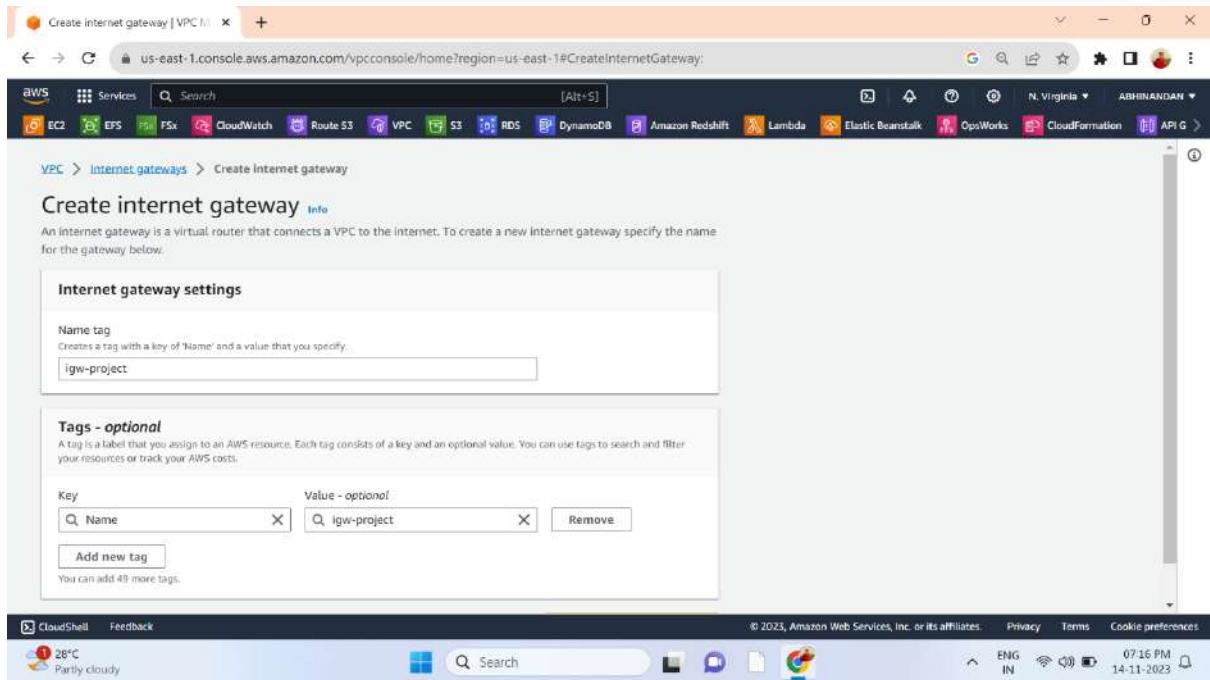
Again, click on Add New Subnet. subnet 3 of 3. subnet name: private; 01. availability zone: choose: us east (N. Virginia)/us-east-1a.



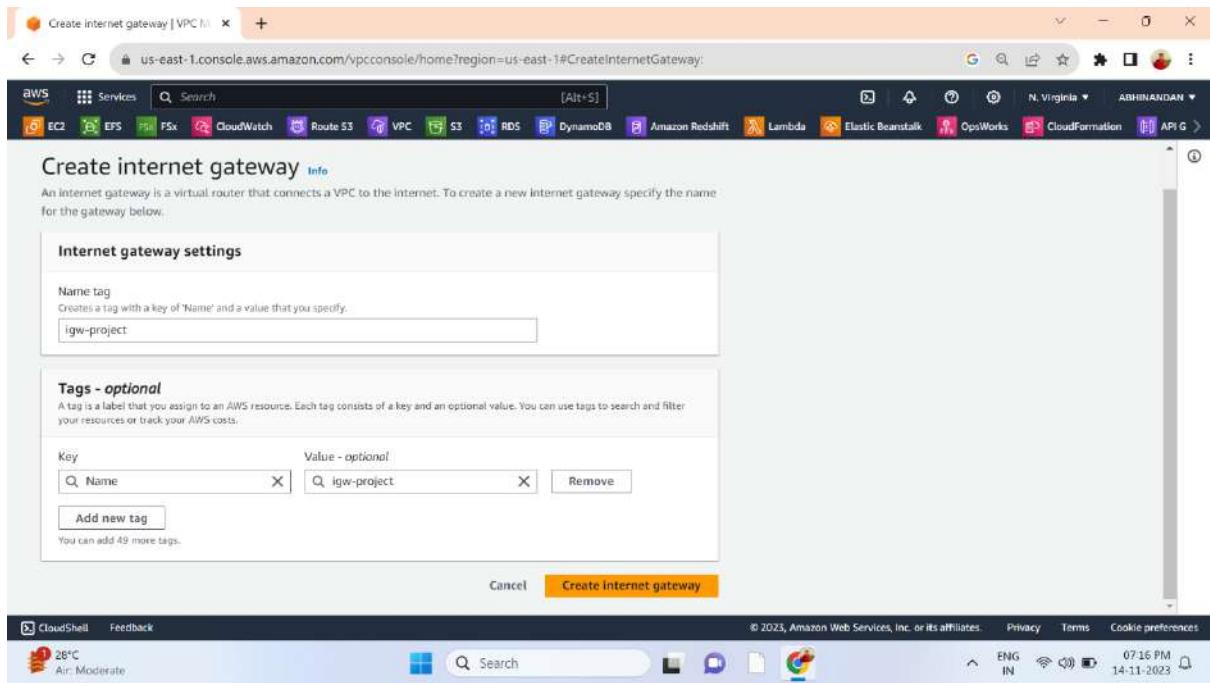
IPV4 CIDR block: 20.20.3.0/24 Click on Create Subnet.



In the navigation bar on the left side, go to the internet gateways option. Click on Create Internet Gateway.



Name tag – igw-project.



Click on Create Internet Gateway.

The screenshot shows the AWS VPC Console with the URL <https://us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#InternetGateway:internetGatewayId=igw-04bad9c29d9ac6e21>. A green banner at the top right says, "The following internet gateway was created: igw-04bad9c29d9ac6e21 - igw-project. You can now attach to a VPC to enable the VPC to communicate with the internet." Below this, the Internet gateway details are shown: Internet gateway ID is igw-04bad9c29d9ac6e21, State is Detached, VPC ID is -, and Owner is 364258757682. A table under "Tags" shows one tag: Name is igw-project. On the left sidebar, under "Virtual private cloud", "Internet gateways" is selected. At the bottom right, there is an "Actions" dropdown and a "Manage tags" button.

Attach to vpc is an option in the upper right corner. as soon as you create an internet gateway.  
Click on Attach to VPC.

The screenshot shows the "Attach Internet Gateway" dialog from the AWS VPC Console. The URL is <https://us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#AttachInternetGateway:internetGatewayId=igw-04bad9c29d9ac6e21...>. The dialog title is "Attach to VPC (igw-04bad9c29d9ac6e21)". It has a "VPC" section with the instruction "Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below." Below this is a "Available VPCs" section with the sub-instruction "Attach the internet gateway to this VPC." A search bar contains the text "vpc-0fc01bf2066565488". At the bottom are "Cancel" and "Attach internet gateway" buttons. The "Attach internet gateway" button is highlighted with a yellow background. The AWS navigation bar and system tray are visible at the bottom.

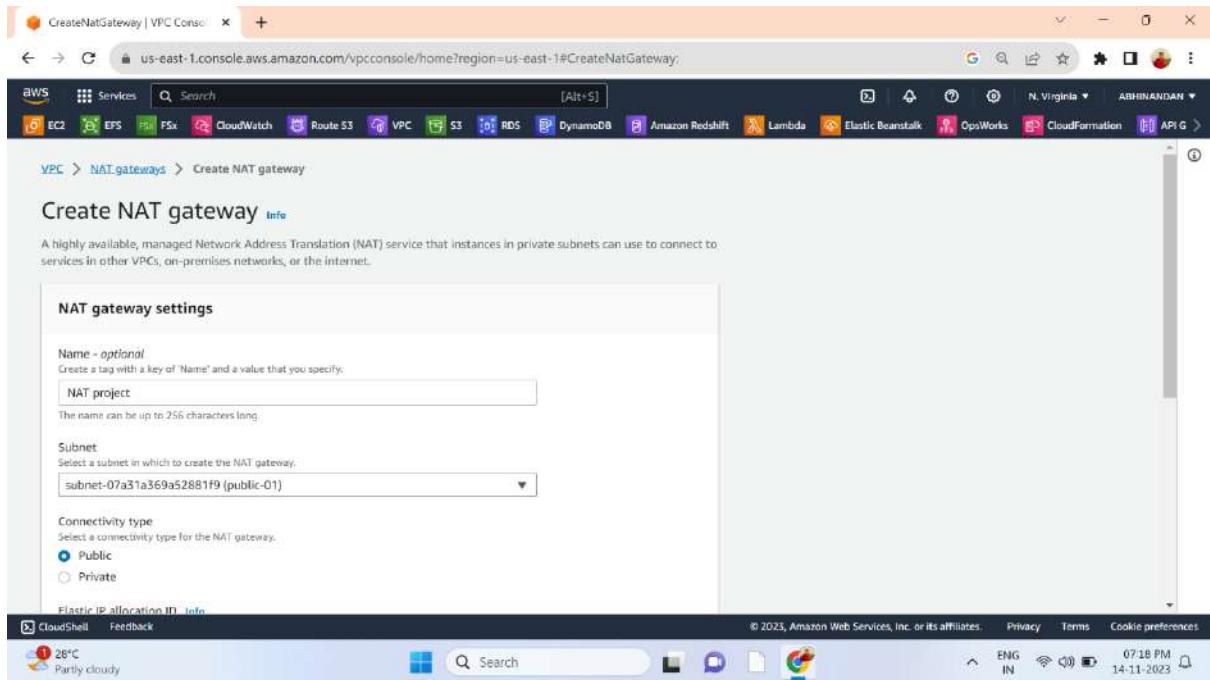
VPC is available VPC's: VPC project. Click on attach internet gateway.

The screenshot shows the AWS VPC Console with the URL <https://us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#InternetGateway:internetGatewayId=igw-04bad9c29d9ac6e21>. The main content area displays a green success message: "Internet gateway igw-04bad9c29d9ac6e21 successfully attached to vpc-0fe01bf2066565488". Below this, the navigation path is shown as VPC > Internet\_gateways > igw-04bad9c29d9ac6e21. The page title is "igw-04bad9c29d9ac6e21 / igw-project". On the left sidebar, under the "Virtual private cloud" section, "Internet gateways" is selected. The main details table shows the Internet gateway ID as "igw-04bad9c29d9ac6e21", State as "Attached", VPC ID as "vpc-0fe01bf2066565488 | vpc-project", and Owner as "364258757682". A "Tags" section shows a single tag named "Name" with the value "igw-project". The bottom status bar indicates the date and time as "14-11-2023 07:17 PM".

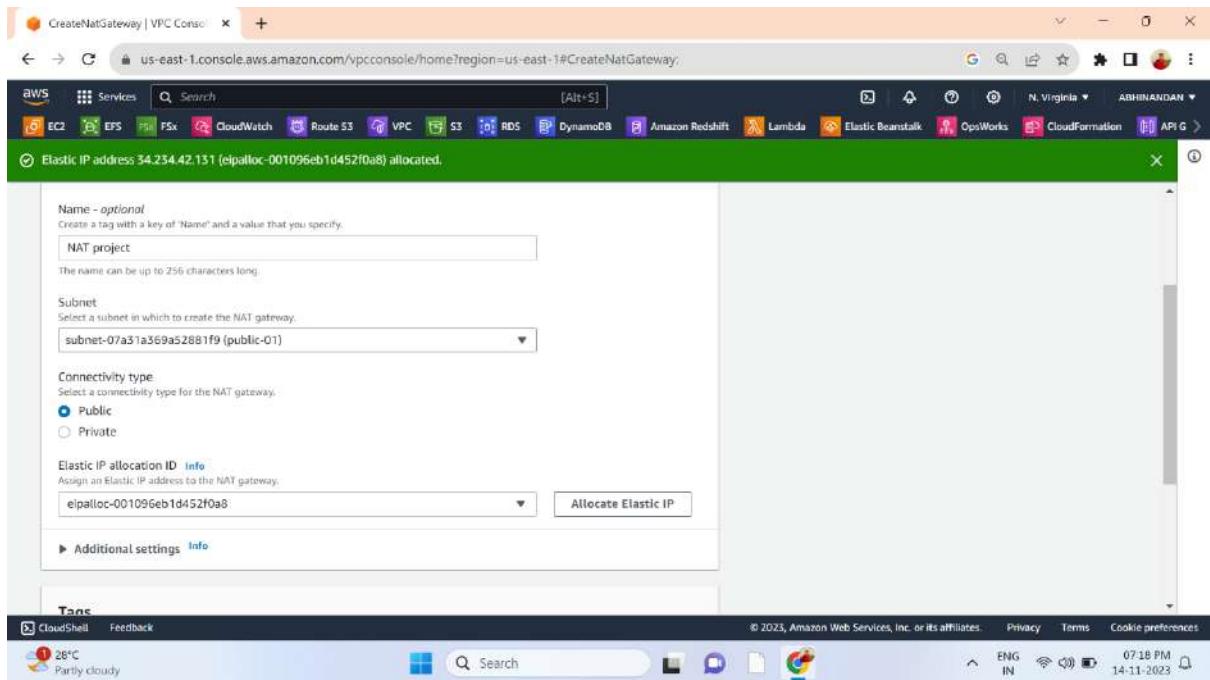
Now we successfully attached internet gateway to vpc.

The screenshot shows the AWS VPC Console with the URL <https://us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#NatGateways>. The main content area displays a table titled "NAT gateways" with a single row labeled "Select a NAT gateway". The table columns include Name, NAT gateway ID, Connectivity..., State, State message, and Primary public IP... The top navigation bar shows the "VPC" service selected. The left sidebar lists various VPC-related options, including "Internet gateways", "Egress-only Internet gateways", "Carrier gateways", "DHCP option sets", "Elastic IPs", "Managed prefix lists", "Endpoints", "Endpoint services", "NAT gateways", and "Peering connections". The bottom status bar indicates the date and time as "14-11-2023 07:17 PM".

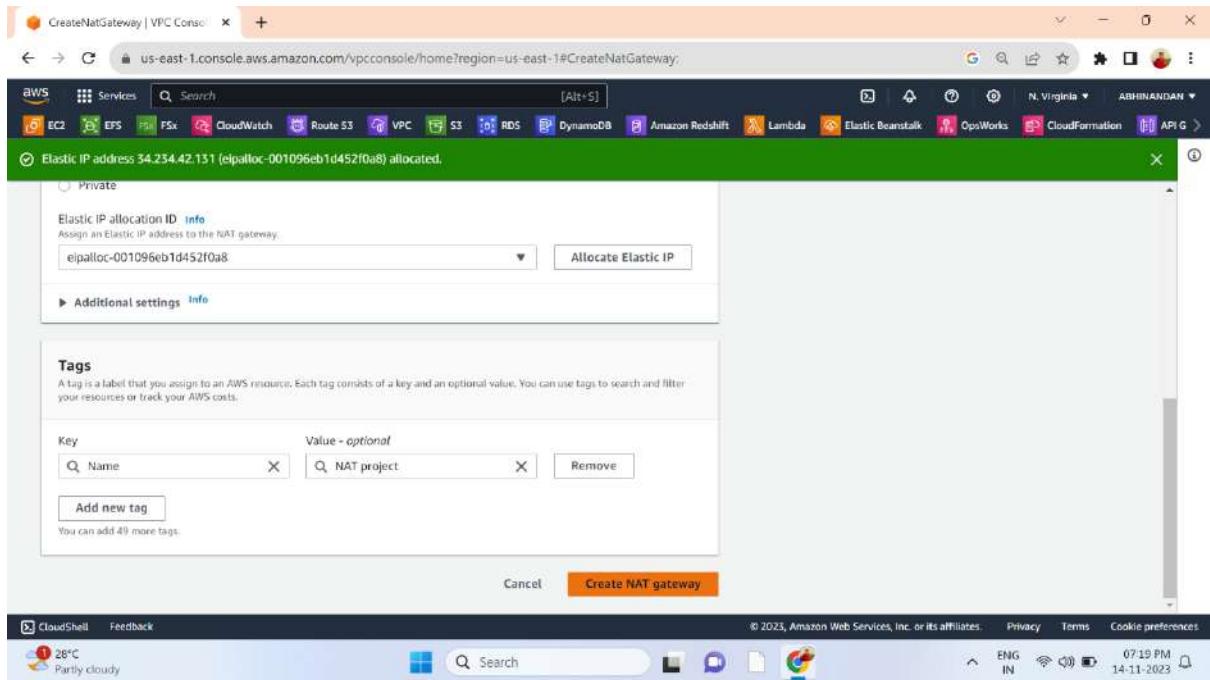
Navigate to the NAT gateways option in the navigation bar on the left side. Click on Create NAT Gateways.



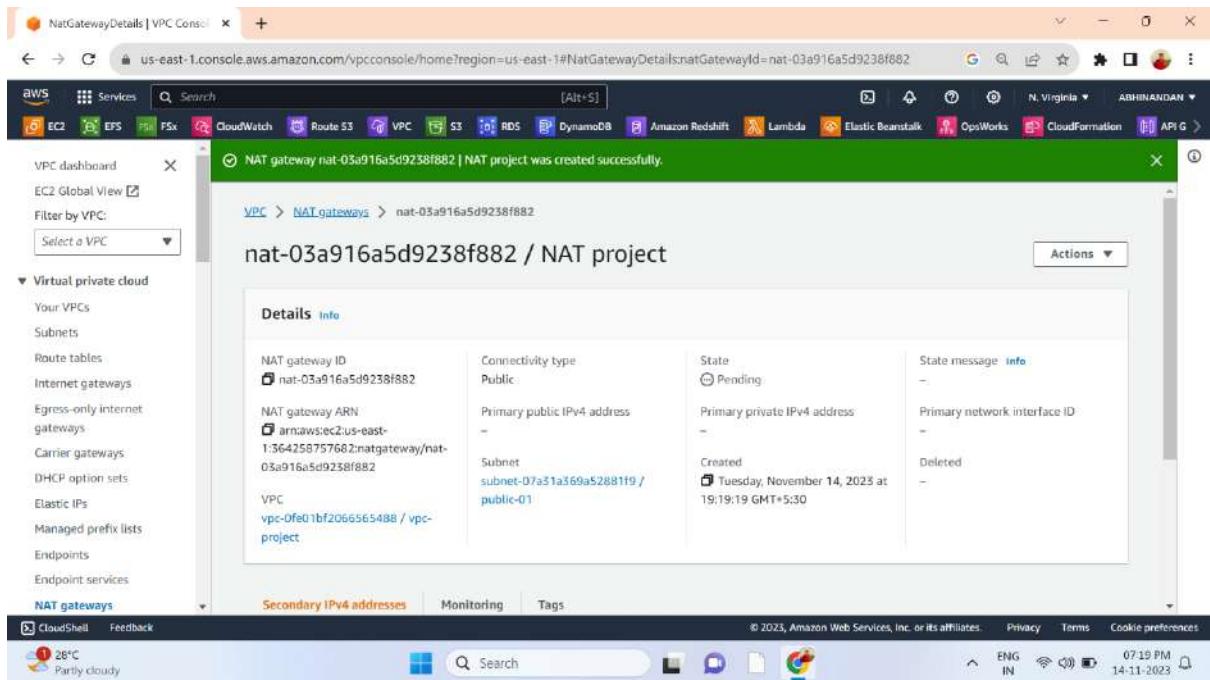
Name: NAT project; subnet: choose the public-01 subnet; connectivity type: public.



Allocate Elastic IP.



Click on Create NAT gateway.



NAT gateway created successfully.

Name	Route table ID	Explicit subnet associations	Edge associations	Main	VPC
-	rtb-0ab3c19b7d6d24859	-	-	Yes	vpc-0f6;
-	rtb-07016e4f1858c53ec	-	-	Yes	vpc-0f6;
RDS-Pvt-rt	rtb-0c60fd022eb39aa9	5 subnets	-	No	vpc-0f7;
-	rtb-0b1a7694f3ae9114c	-	-	Yes	vpc-0f7;

Go to the route tables option in the navigation bar on the left side. We will get the default route table as soon as we create a VPC.

Name	Route table ID	Explicit subnet associations	Edge associations	Main	VPC
-	rtb-0ab3c19b7d6d24859	-	-	Yes	vpc-0f6;
<input checked="" type="checkbox"/> public-RT	rtb-07016e4f1858c53ec	-	-	Yes	vpc-0f6;
-	rtb-0c60fd022eb39aa9	5 subnets	-	No	vpc-0f7;
-	rtb-0b1a7694f3ae9114c	-	-	Yes	vpc-0f7;

Modify the name "public RT." (defaultly created) Choose "public-RT".

**Edit subnet associations**

Change which subnets are associated with this route table.

Available subnets (2/3)					
Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID	
private-01	subnet-0b2cea82441f94ee2	20.20.3.0/24	-	Main (rtb-07016e4f1858c53ec / public)	
<input checked="" type="checkbox"/> public-02	subnet-03bdcb0d4cf21e10	20.20.2.0/24	-	Main (rtb-07016e4f1858c53ec / public)	
<input checked="" type="checkbox"/> public-01	subnet-07a31a369a52881f9	20.20.1.0/24	-	Main (rtb-07016e4f1858c53ec / public)	

**Selected subnets**

- subnet-07a31a369a52881f9 / public-01
- subnet-03bdcb0d4cf21e10 / public-02

**Buttons:** Cancel, Save associations

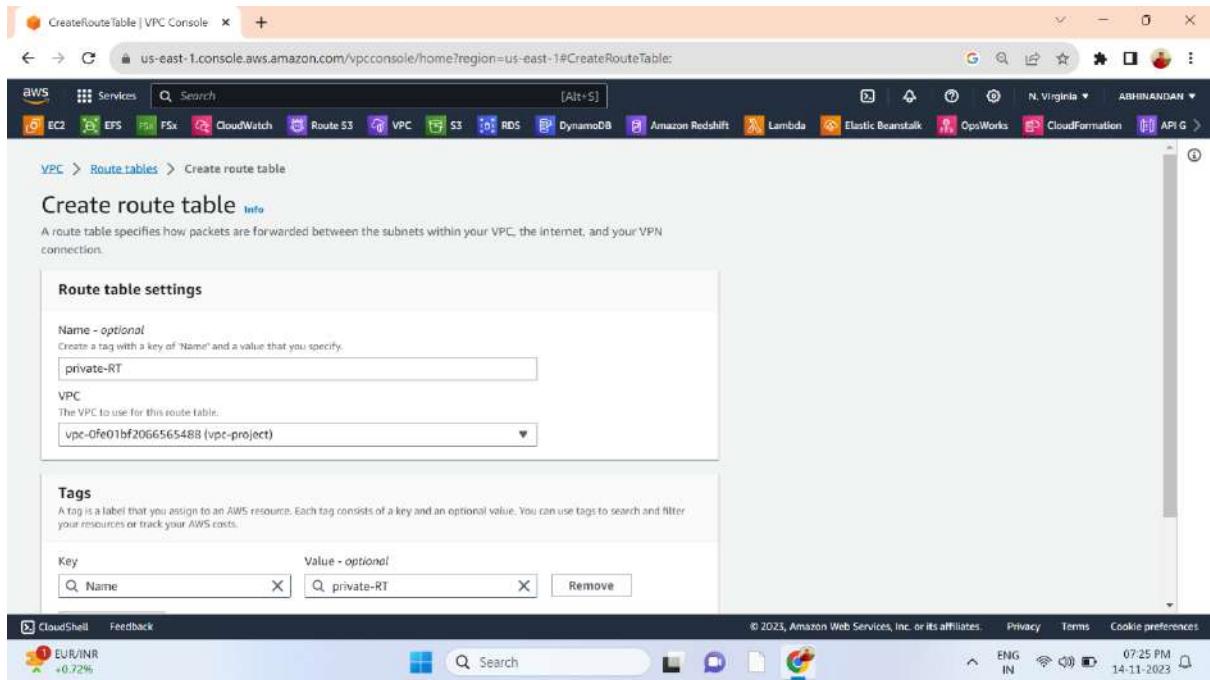
Choose subnet associations. Click on Edit Subnet Association. Choose the public-01, public-02 subnet over here.

**Edit routes**

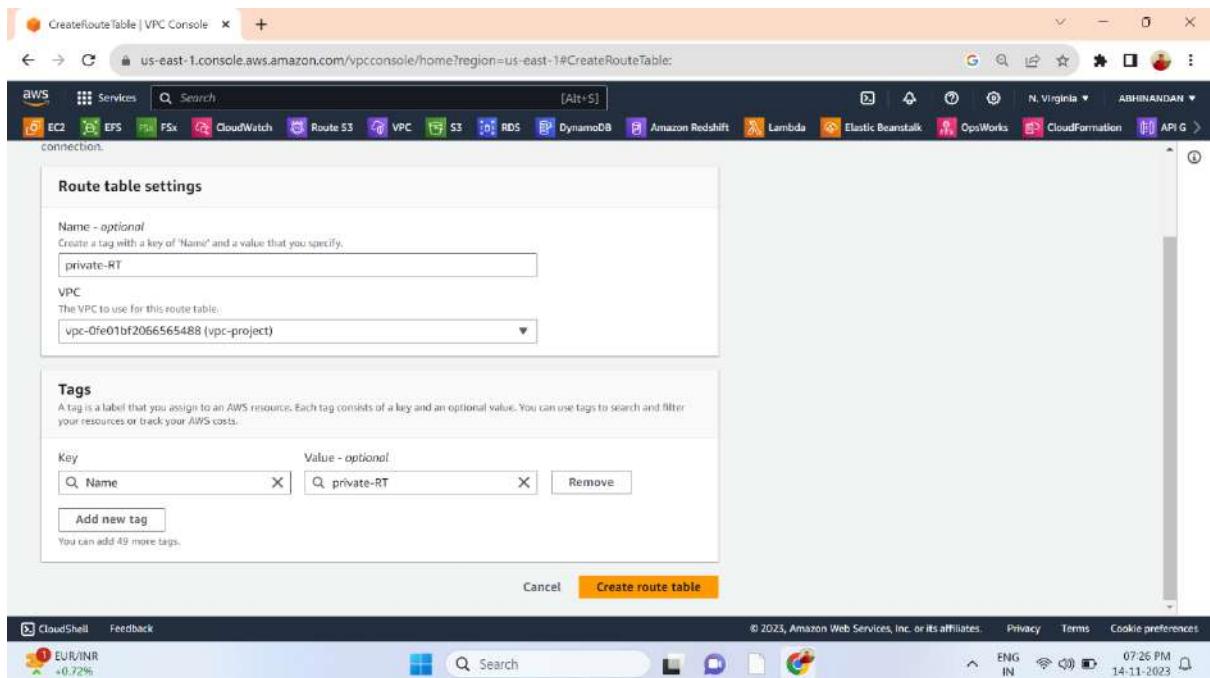
Destination	Target	Status	Propagated
20.20.0.0/16	local Q_ local	Active	No
Q_ 0.0.0.0/0	Internet Gateway Q_ igw-	-	No

**Buttons:** Add route, Cancel, Preview, Save changes

Click on Edit Routes. Click on Add Routes. choose 0.0.0.0/0 (Destination). choose target as an Internet gateway. and next choose our newly created internet gateways. Below, click on Save Changes.



Now we will create a private route table. In route table settings, give name, private RT, vpc, and choose our created vpc project.



Click on Create route table.

The screenshot shows the AWS VPC Console with a success message: "Route table rtb-02b5f76de3d5c0180 | private-RT was created successfully." The left sidebar shows the navigation path: VPC > Route tables > rtb-02b5f76de3d5c0180 / private-RT. The main details pane shows the route table ID as rtb-02b5f76de3d5c0180, which is Main and associated with the VPC vpc-0fe01bf2066565488. There is one route listed under Routes.

Now private route table created successfully.

The screenshot shows the "Edit subnet associations" dialog for the route table rtb-02b5f76de3d5c0180. It lists available subnets (private-01, public-02, public-01) and selected subnets (private-01). The selected subnet is highlighted in a box. At the bottom, there are "Cancel" and "Save associations" buttons.

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
private-01	subnet-0b2cea82444f94ee2	20.20.3.0/24	-	Main rtb-07016e4f1858c53ec / public
public-02	subnet-03bdcb0d4cf21e10	20.20.2.0/24	-	rtb-07016e4f1858c53ec / public-RT
public-01	subnet-07a31a369a52881f9	20.20.1.0/24	-	rtb-07016e4f1858c53ec / public-RT

Edit subnet associations; choose private-01. Below, click on Save Associations.

The screenshot shows the 'Edit routes' page in the AWS VPC console. A table lists routes:

Destination	Target	Status	Propagated
20.20.0.0/16	local	Active	No
0.0.0.0/0	NAT Gateway	-	No

Buttons at the bottom include 'Add route', 'Save changes', and 'Cancel'.

Then choose routes. Click on Edit Routes. Click on Add Route. destination: 0.0.0.0/0 target: choose NAT gateway; select our created NAT gateway. Below, click on Save Changes.

The screenshot shows the EC2 Instances page in the AWS console. The left sidebar is expanded, showing 'Instances' selected. The main content area displays:

No instances  
You do not have any instances in this region

Launch instances

Below this, a modal window titled 'Select an instance' is open.

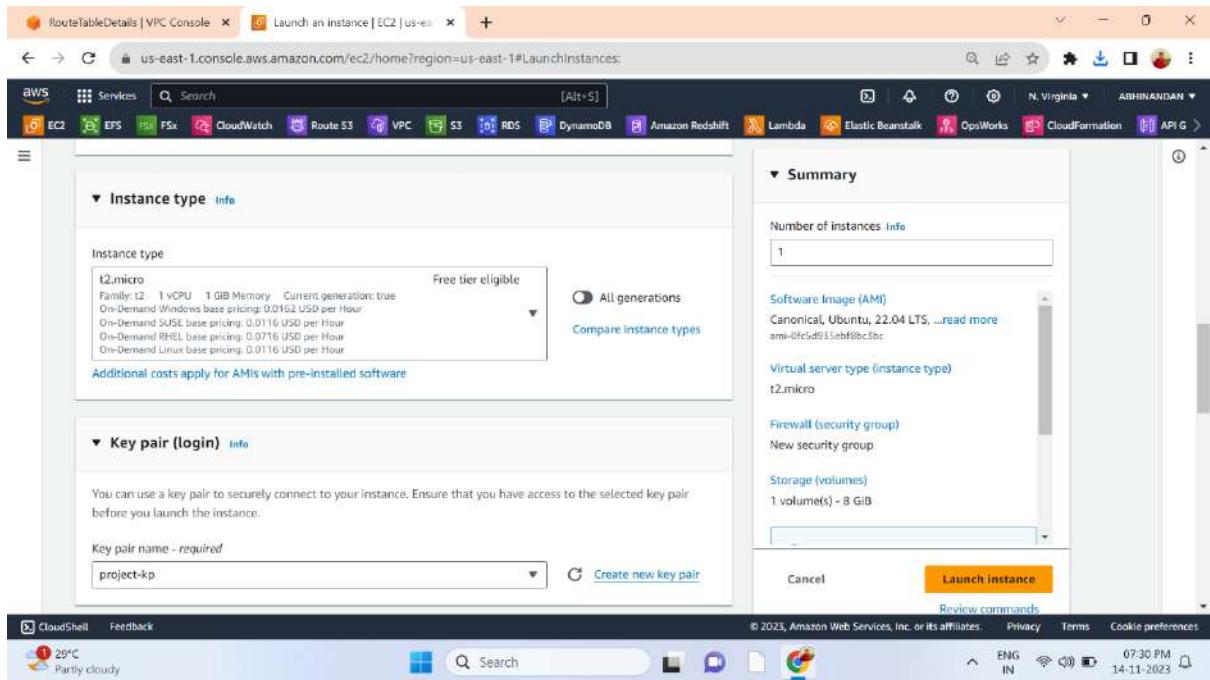
Let's create two instances. Click on launch instances.

The screenshot shows the 'Launch an instance' wizard in the AWS EC2 console. The current step is 'Name and tags'. The 'Name' field contains 'public-project'. The 'Software Image (AMI)' section shows 'Amazon Linux 2023 AMI 2023.2.2...'. The 'Virtual server type (instance type)' is set to 't2.micro'. The 'Storage (volumes)' section indicates '1 volume(s) - 8 GiB'. The 'Launch instance' button is highlighted in orange at the bottom right.

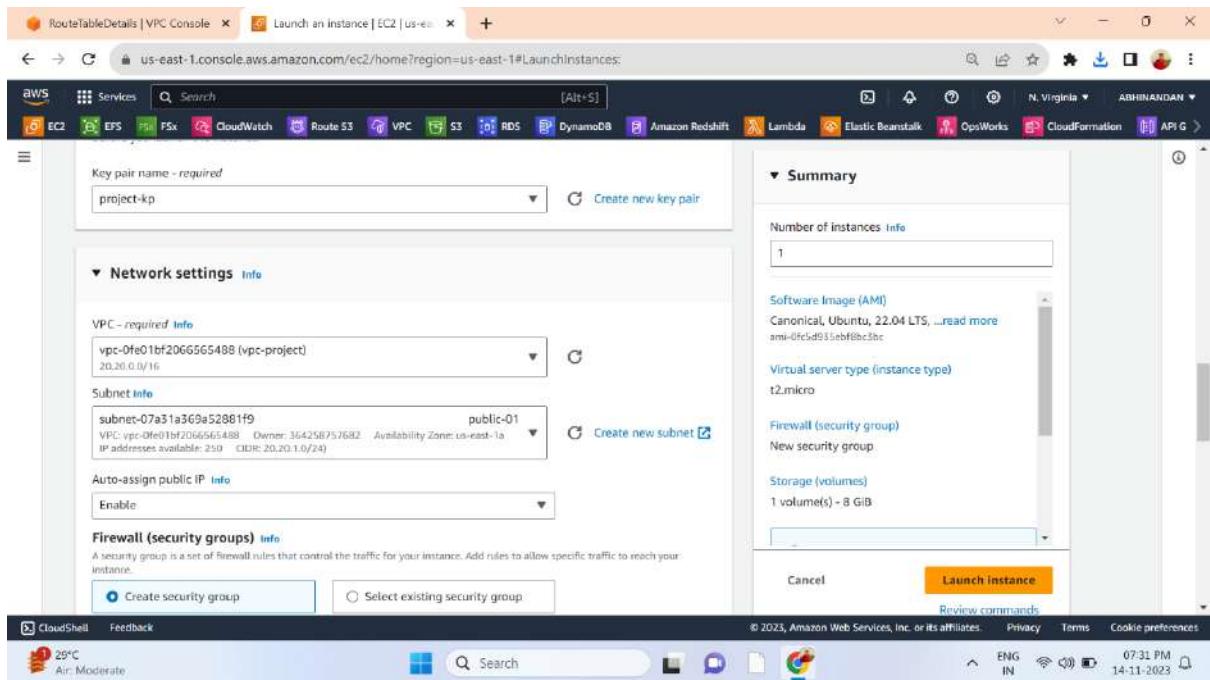
Give a Name – “public-project”.

The screenshot shows the 'Launch an instance' wizard in the AWS EC2 console. The current step is 'Application and OS Images (Amazon Machine Image)'. The 'Recent' tab is selected, showing options for Amazon Linux, macOS, Ubuntu, Windows, Red Hat, and SUSE. The 'Ubuntu' option is highlighted. The 'Summary' section shows 'Number of instances: 1'. The 'Software Image (AMI)' section shows 'Canonical, Ubuntu, 22.04 LTS, ...'. The 'Virtual server type (instance type)' is set to 't2.micro'. The 'Storage (volumes)' section indicates '1 volume(s) - 8 GiB'. The 'Launch instance' button is highlighted in orange at the bottom right.

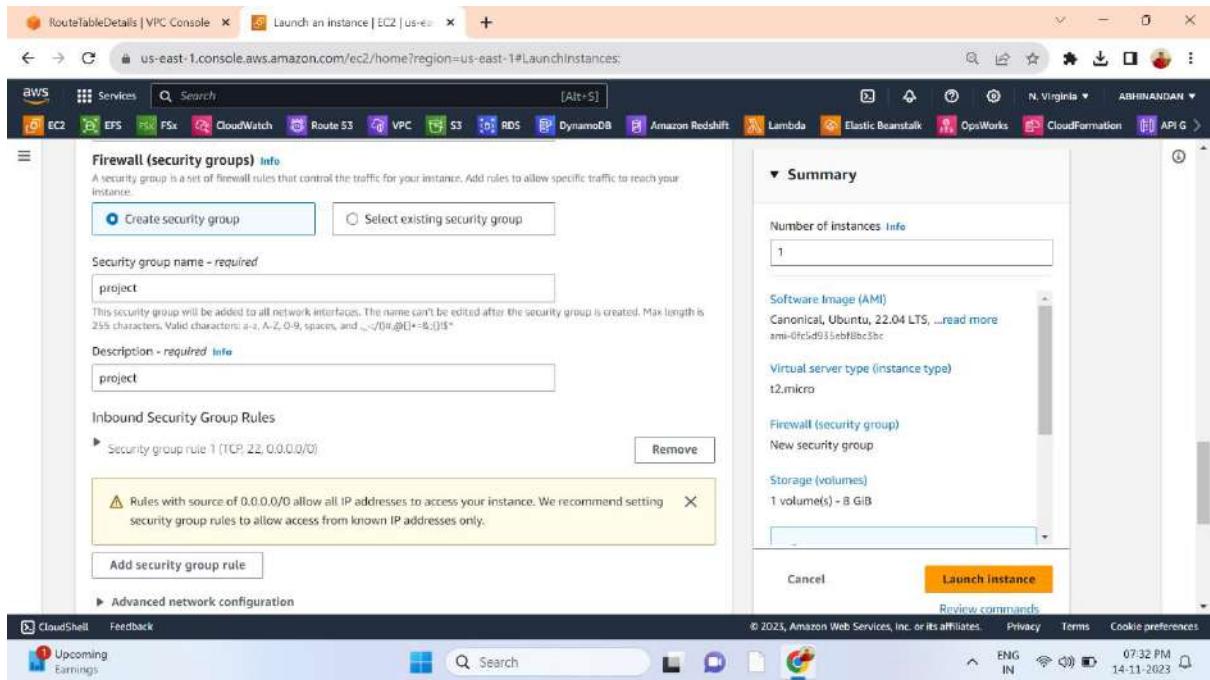
Choose AMI – Ubuntu.



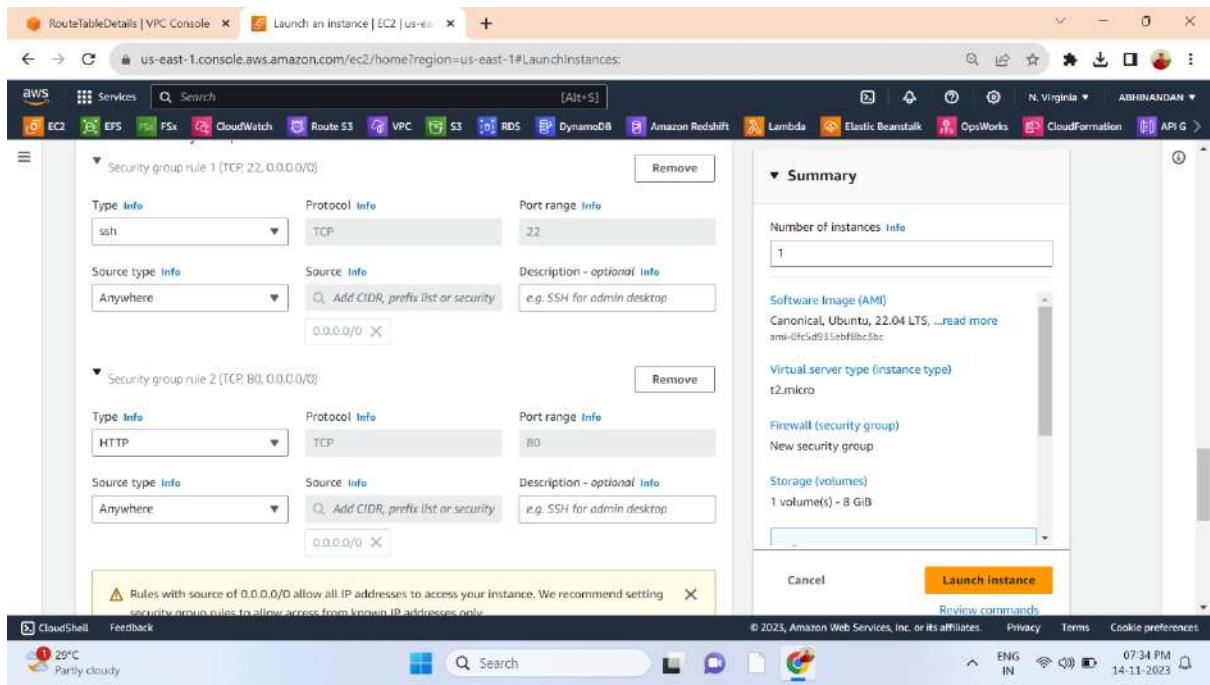
Choose or Create a key pair. Key pair should be (.pem)



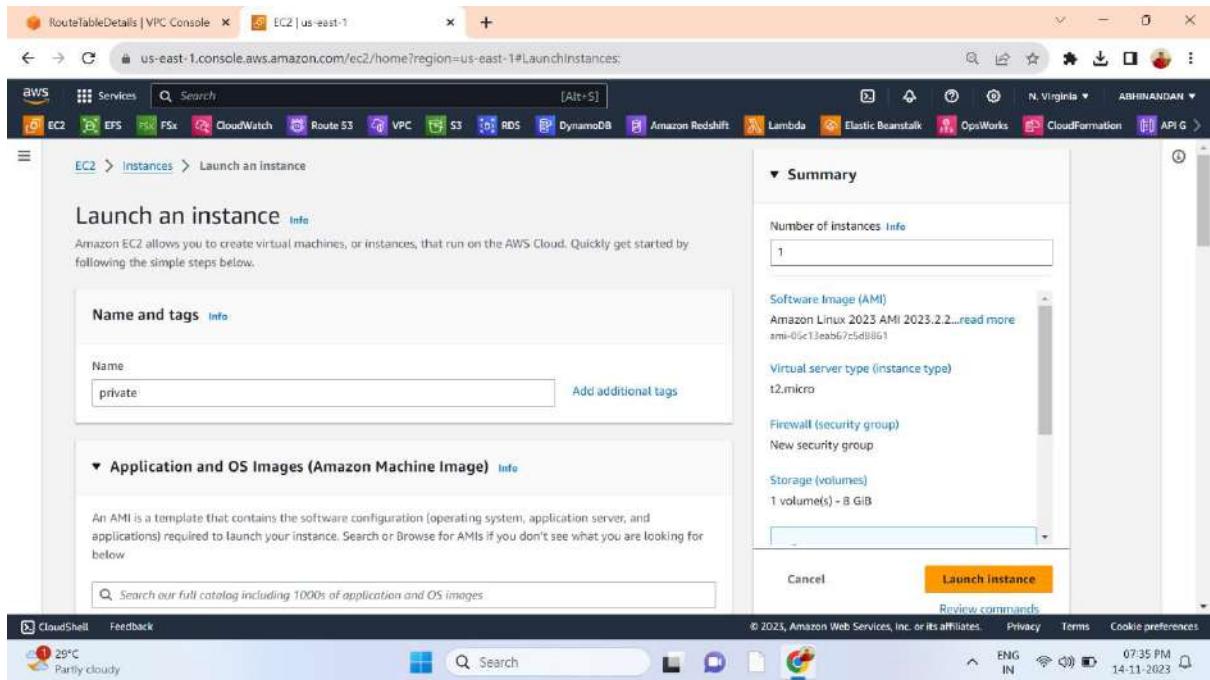
In network settings. click edit. Choose our newly created VPC named "VPC-project". Choose subnet "Public-01". Auto-assign a public IP; enable it.



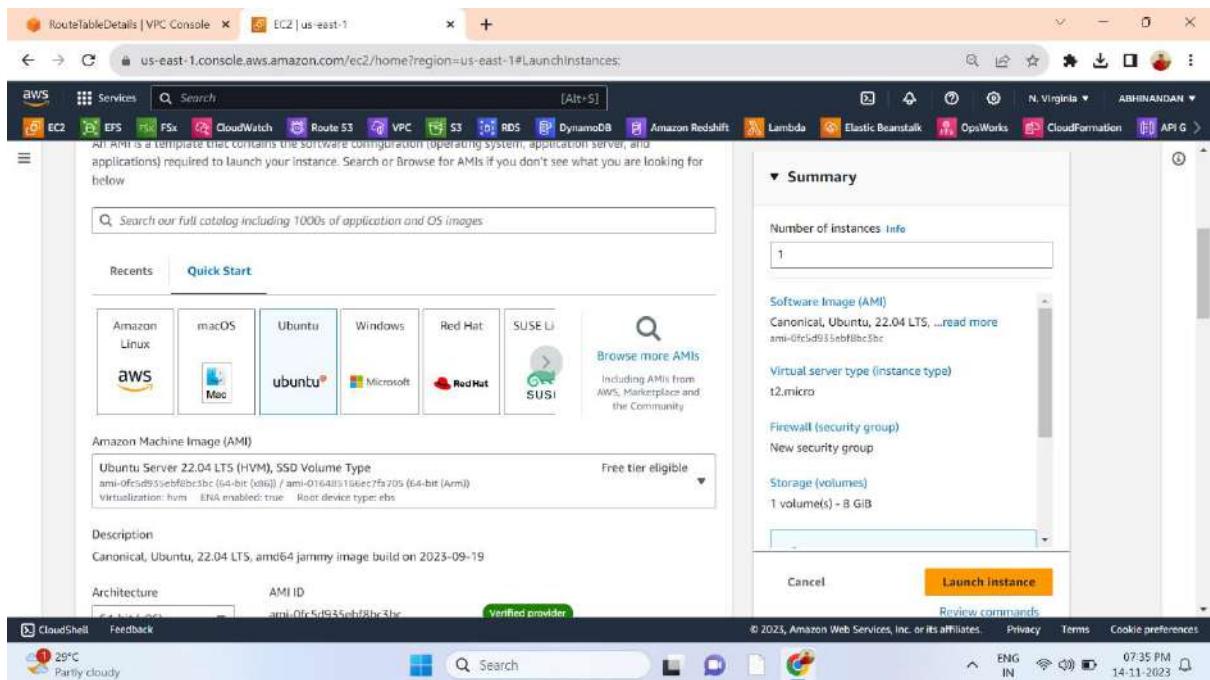
In firewall security groups, choose Create Security Group. security group name: project. description: project. Click on Add security group rule.



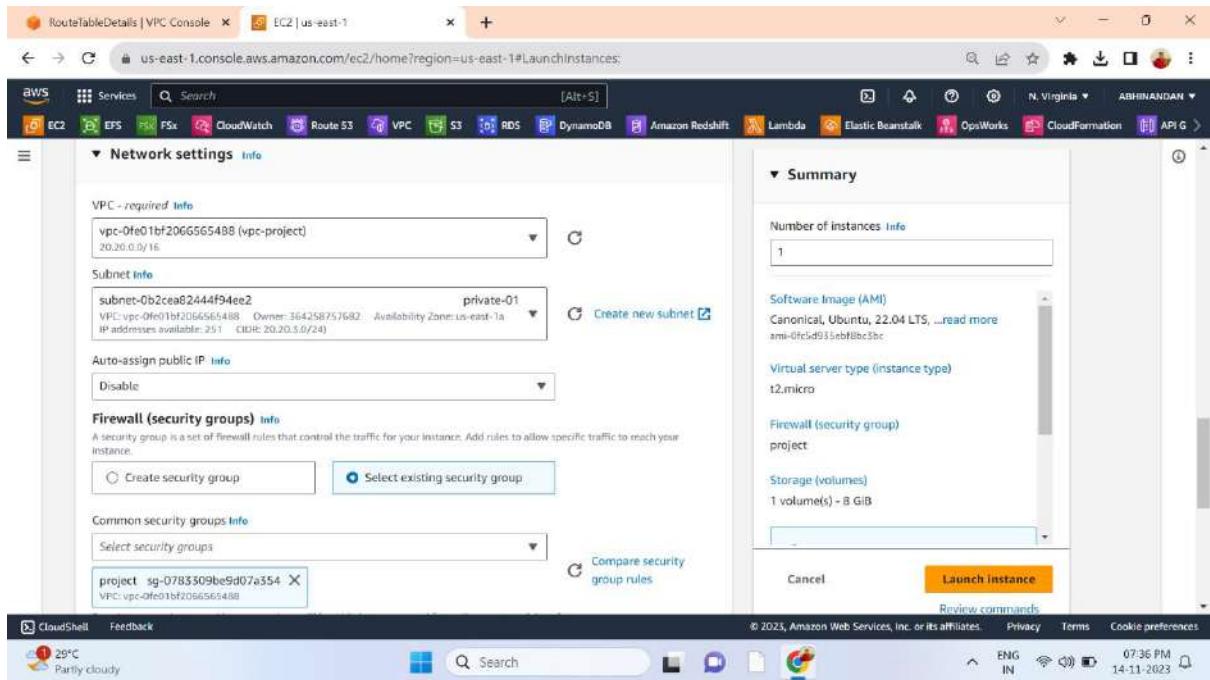
Type – HTTP. Source Type – Anywhere. Just Click on launch instance. (This is my public Instance)



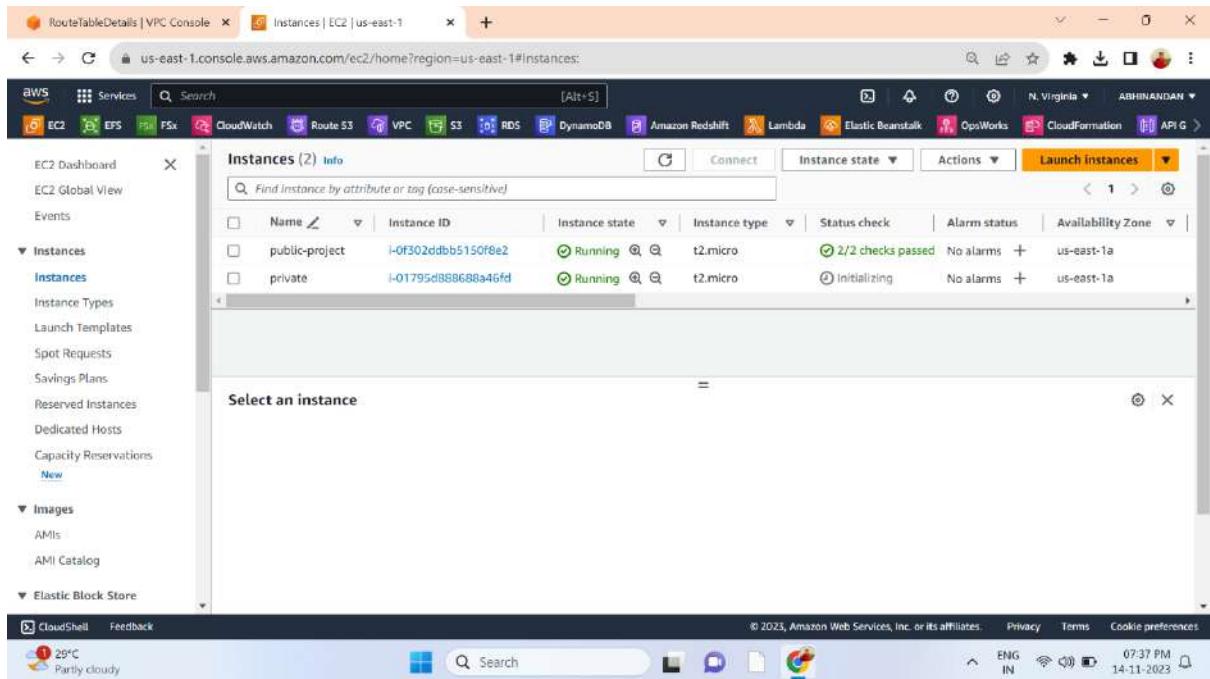
Now we will be creating a private EC2 instance. Give a name to an instance as "private".



Choose AMI – Ubuntu.



In network settings, click on edit and choose our own VPC that we created. subnet: choose private; 01. auto: assign a public IP; keep it disabled. In the firewall security group, choose an existing security group. It is named "project", Choose that. Then click on Launch instance.



Now we successfully launched an instances.

The screenshot shows the 'Create bucket' wizard in the AWS S3 console. The 'General configuration' tab is active, displaying fields for 'Bucket name' (set to 'abhig-project2') and 'AWS Region' (set to 'US East (N. Virginia) us-east-1'). Below these, there's a section for 'Copy settings from existing bucket - optional' with a 'Choose Bucket' button. The 'Object Ownership' tab is also visible below. The browser address bar shows the URL: `s3.console.aws.amazon.com/s3/bucket/create?region=us-east-1`. The top navigation bar includes links for EC2, S3, RDS, Lambda, and other AWS services.

Now we will create a S3 bucket. Name my bucket "myproject-abhig". AWS region: US east (N. Virginia).

The screenshot shows the 'Object Ownership' tab of the 'Create bucket' wizard. It contains two options: 'ACLs disabled (recommended)' and 'ACLs enabled'. The 'ACLs enabled' option is selected, with a note explaining that objects can be owned by other AWS accounts and access can be controlled via ACLs. A warning message states: 'We recommend disabling ACLs, unless you need to control access for each object individually or to have the object writer own the data they upload. Using a bucket policy instead of ACLs to share data with users outside of your account simplifies permissions management and auditing.' Below this, the 'Object Ownership' section shows 'Bucket owner preferred' is selected, with a note that new objects are owned by the bucket owner if no specific ACL is specified. The 'Object writer' option is also available. The browser address bar shows the URL: `s3.console.aws.amazon.com/s3/bucket/create?region=us-east-1`. The top navigation bar includes links for EC2, S3, RDS, Lambda, and other AWS services.

In Object Ownership – ACLs enabled. ObjectOwnership – Bucket owner preffered.

**Block Public Access settings for this bucket**

Public access is granted to buckets and objects through access control lists (ACLs), bucket policies, access point policies, or all. In order to ensure that public access to this bucket and its objects is blocked, turn on Block all public access. These settings apply only to this bucket and its access points. AWS recommends that you turn on Block all public access, but before applying any of these settings, ensure that your applications will work correctly without public access. If you require some level of public access to this bucket or objects within, you can customize the individual settings below to suit your specific storage use cases. [Learn more](#)

**Block all public access**  
Turning this setting on is the same as turning on all four settings below. Each of the following settings are independent of one another.

- Block public access to buckets and objects granted through new access control lists (ACLs)**  
S3 will block public access permissions applied to newly added buckets or objects, and prevent the creation of new public access ACLs for existing buckets and objects. This setting doesn't change any existing permissions that allow public access to S3 resources using ACLs.
- Block public access to buckets and objects granted through any access control lists (ACLs)**  
S3 will ignore all ACLs that grant public access to buckets and objects.
- Block public access to buckets and objects granted through new public bucket or access point policies**  
S3 will block new bucket and access point policies that grant public access to buckets and objects. This setting doesn't change any existing policies that allow public access to S3 resources.
- Block public and cross-account access to buckets and objects through any public bucket or access point policies**  
S3 will ignore public and cross-account access for buckets or access points with policies that grant public access to buckets and objects.

**⚠ Turning off block all public access might result in this bucket and the objects within becoming public.**

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

CloudShell Feedback 27°C Partly cloudy Search ENG IN 07:39 PM 14-11-2023

Disable Block all public access.

**Bucket Versioning**

Versioning is a means of keeping multiple variants of an object in the same bucket. You can use versioning to preserve, retrieve, and restore

I acknowledge that the current settings might result in this bucket and the objects within becoming public.

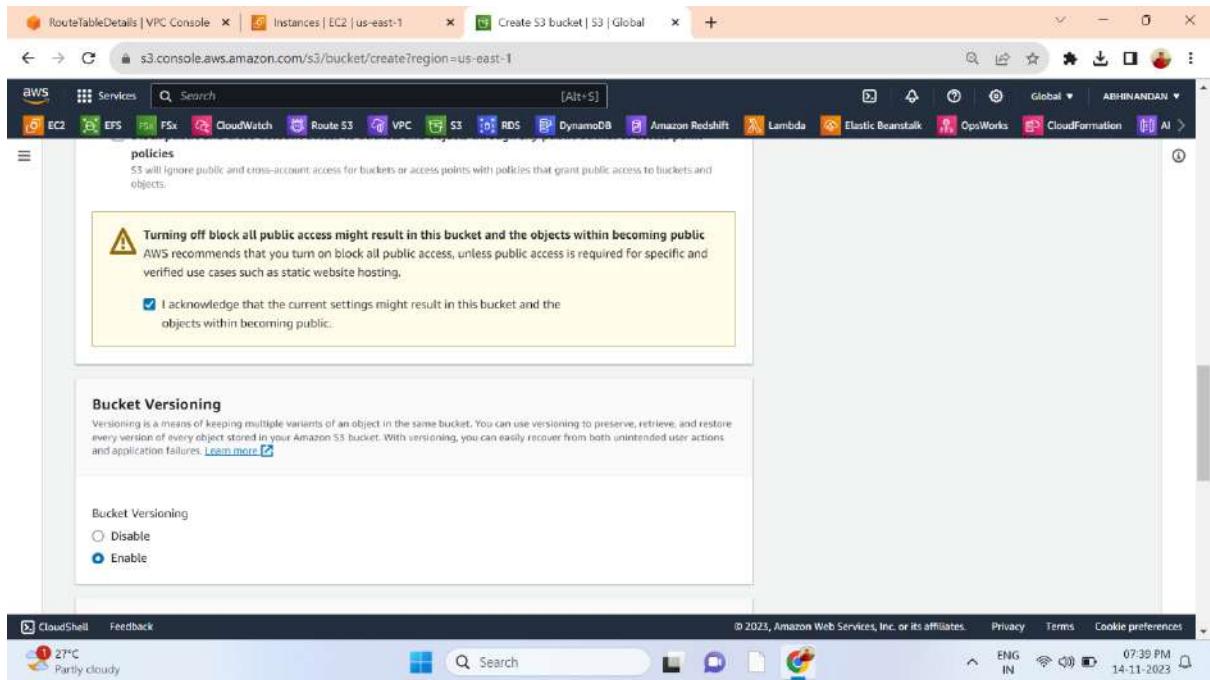
**⚠ Turning off block all public access might result in this bucket and the objects within becoming public.**

AWS recommends that you turn on block all public access, unless public access is required for specific and verified use cases such as static website hosting.

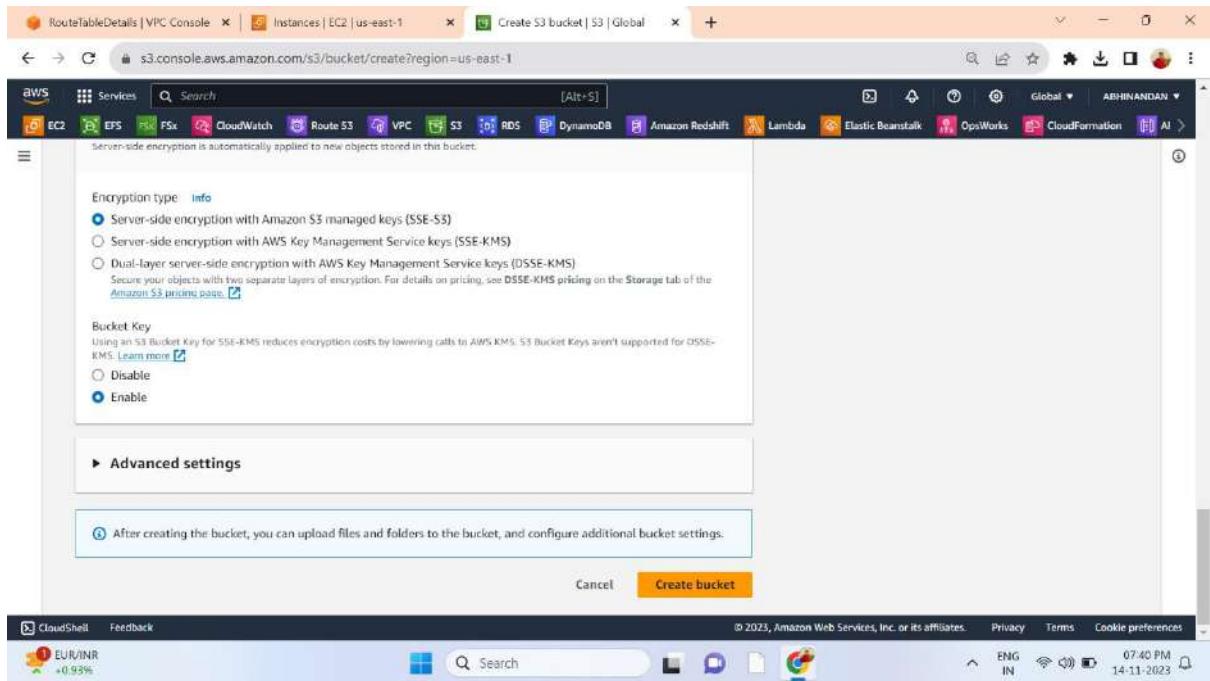
© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

CloudShell Feedback 27°C Partly cloudy Search ENG IN 07:39 PM 14-11-2023

Acknowledge it.



Enable Bucket Versioning.



Choose Encryption type as Server-side encryption with Amazon S3 managed keys (SSE-S3)  
Enable Bucket key.

The screenshot shows the AWS S3 console with a green success message at the top: "Successfully created bucket 'abhibg-project2'". Below it, a table lists three buckets: "abhibg-project2", "cf-templates-9xkkmq3wznzro-us-east-1", and "elast/cbeanstalk-us-east-1-364250257682".

Name	AWS Region	Access	Creation date
abhibg-project2	US East (N. Virginia) us-east-1	Objects can be public	November 14, 2023, 19:40:44 (UTC+05:30)
cf-templates-9xkkmq3wznzro-us-east-1	US East (N. Virginia) us-east-1	Bucket and objects not public	November 1, 2023, 21:02:29 (UTC+05:30)
elast/cbeanstalk-us-east-1-364250257682	US East (N. Virginia) us-east-1	Objects can be public	October 19, 2023, 08:33:55 (UTC+05:30)

Now we successfully Created a S3 bucket.

The screenshot shows the AWS S3 console for the bucket "abhibg-project2". The "Objects" tab is selected, showing a table with one row: "No objects". A message below the table says "You don't have any objects in this bucket." At the bottom, there is a "Upload" button.

Now we will add objects to our S3 bucket. Click on Upload.

The screenshot shows the AWS S3 console interface. At the top, there are four tabs: 'RouteTableDetails | VPC Console', 'Instances | EC2 | us-east-1', 'Upload objects - S3 bucket abhig-project2', and 'aws-code-main/EmpApp.py at main'. The current tab is 'Upload objects - S3 bucket abhig-project2?region=us-east-1'. The browser address bar shows 's3.console.aws.amazon.com/s3/upload/abhig-project2?region=us-east-1'. The AWS navigation bar includes services like EC2, EFS, FSx, CloudWatch, Route 53, VPC, S3, RDS, DynamoDB, Amazon Redshift, Lambda, Elastic Beanstalk, OpsWorks, CloudFormation, and AI.

The main content area has a large text input field with placeholder text: '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 (2 Total, 2.3 KB)'. It lists two files: 'error.html' (text/html, 798.0 B) and 'aboutus.html' (text/html, 1.5 KB). There are 'Remove', 'Add files', and 'Add folder' buttons above the table. A search bar labeled 'Find by name' is also present.

The 'Destination' section shows the URL 's3://abhig-project2'. The bottom of the screen shows a Windows taskbar with icons for CloudShell, Feedback, and a weather widget.

Here now we added error.html and aboutus.html page

The screenshot shows the AWS S3 console interface for the bucket 'abhig-project2'. The left sidebar has sections for 'Buckets', 'Storage Lens', and 'Feature spotlight'. The main area is titled 'abhig-project2' and shows the 'Objects' tab selected. The 'Objects (2)' section lists 'aboutus.html' and 'error.html'. Both files are of type 'html' and were last modified on November 14, 2023, at 19:59:03 (UTC+05:30). The 'Actions' column for 'aboutus.html' is open, showing options like 'Initiate restore', 'Query with S3 Select', 'Edit actions', 'Rename object', 'Edit storage class', 'Edit server-side encryption', 'Edit metadata', 'Edit tags', and 'Make public using ACL'. The bottom of the screen shows a Windows taskbar with icons for CloudShell, Feedback, and a weather widget.

Choose the aboutus.html file above, click on actions, and make it public using ACL.

The make public action enables public read access in the object access control list (ACL) settings. [Learn more](#)

⚠ When public read access is enabled and not blocked by Block Public Access settings, anyone in the world can access the specified objects.

**Specified objects**

Name	Type	Last modified	Size
aboutus.html	html	November 14, 2023, 19:59:03 (UTC+05:30)	1.5 KB

Cancel **Make public**

Now Click on Make public.

The make public action enables public read access in the object access control list (ACL) settings. [Learn more](#)

⚠ When public read access is enabled and not blocked by Block Public Access settings, anyone in the world can access the specified objects.

**Specified objects**

Name	Type	Last modified	Size
error.html	html	November 14, 2023, 19:59:01 (UTC+05:30)	798.0 B

Cancel **Make public**

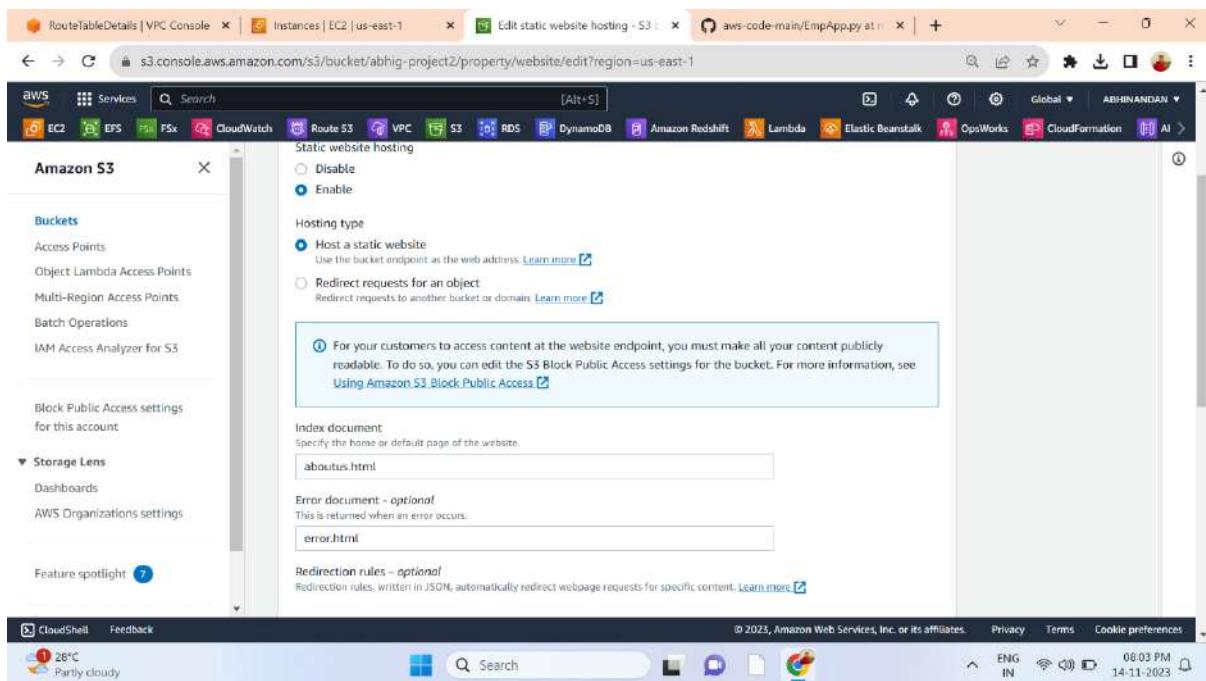
Now again Click on Make public.

The screenshot shows the AWS S3 console with the 'Properties' tab selected. On the left, there's a sidebar for 'Amazon S3' with options like Buckets, Requester pays, and Static website hosting. The main area displays the properties of the 'abhg-project2' bucket. Under 'Static website hosting', it says 'Disabled' and has an 'Edit' button. Other sections like 'Requester pays' also have 'Edit' buttons.

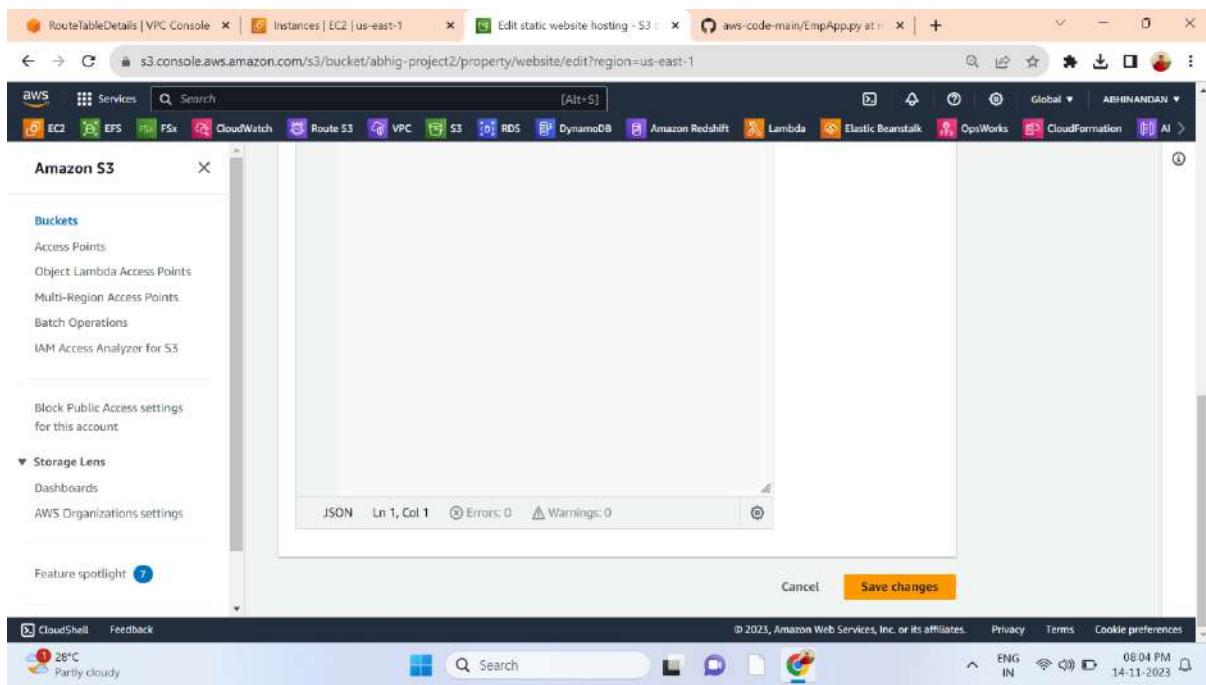
Here Click on Edit Static website hosting.

This screenshot shows the 'Edit static website hosting' configuration page for the 'abhg-project2' bucket. It includes fields for enabling or disabling static website hosting, selecting a hosting type (either 'Host a static website' or 'Redirect requests for an object'), and specifying an index document ('aboutus.html'). A note at the bottom explains that content must be publicly readable if using a static website host.

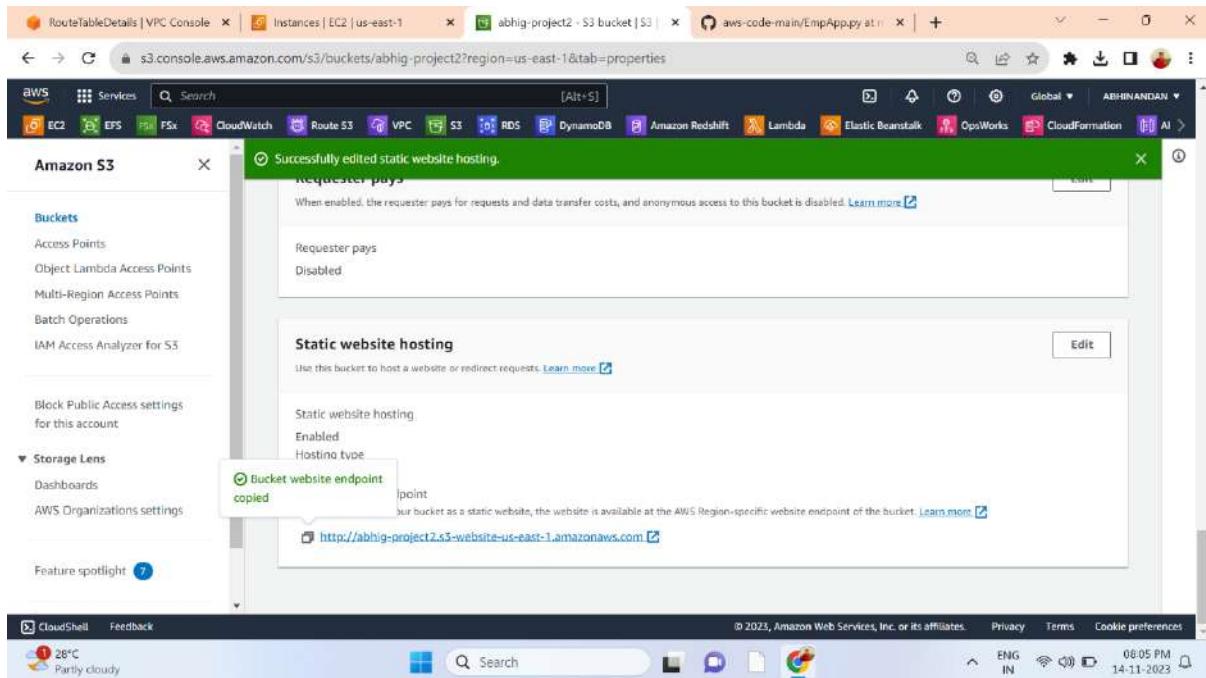
Enable static website hosting. Host a static website. In the index document, give an aboutus.html document.



In Error document give error.html document



Now let us make it static, Click On Save Changes.



In the static website hosting section, we will see a http link. Copy the http link.

Welcome to our website! [Your company or organization name] is dedicated to [briefly describe your mission or purpose]. We strive to [mention your key goals or values].

## Our Team

Our team consists of passionate individuals committed to [describe your team's expertise or focus]. We work together to [highlight how your team contributes to your mission].

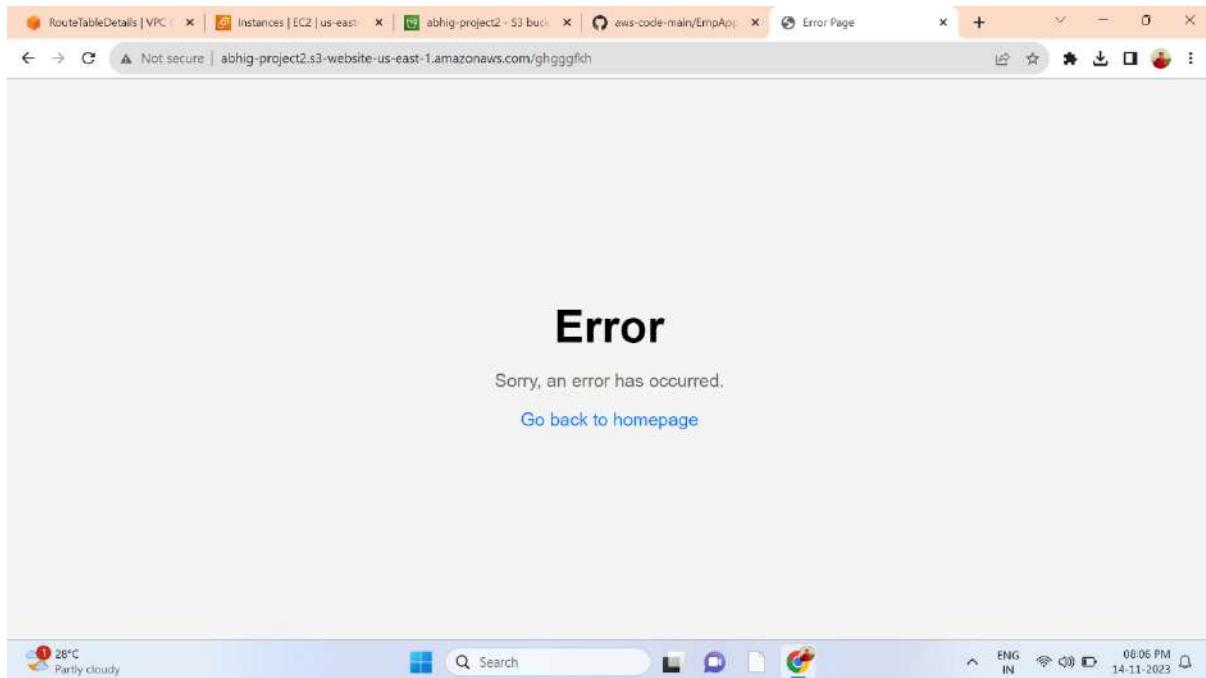
## Our Vision

At [Your company or organization name], our vision is to [explain your long-term goals or aspirations]. We aim to [mention how you plan to make an impact or contribute to a certain cause].

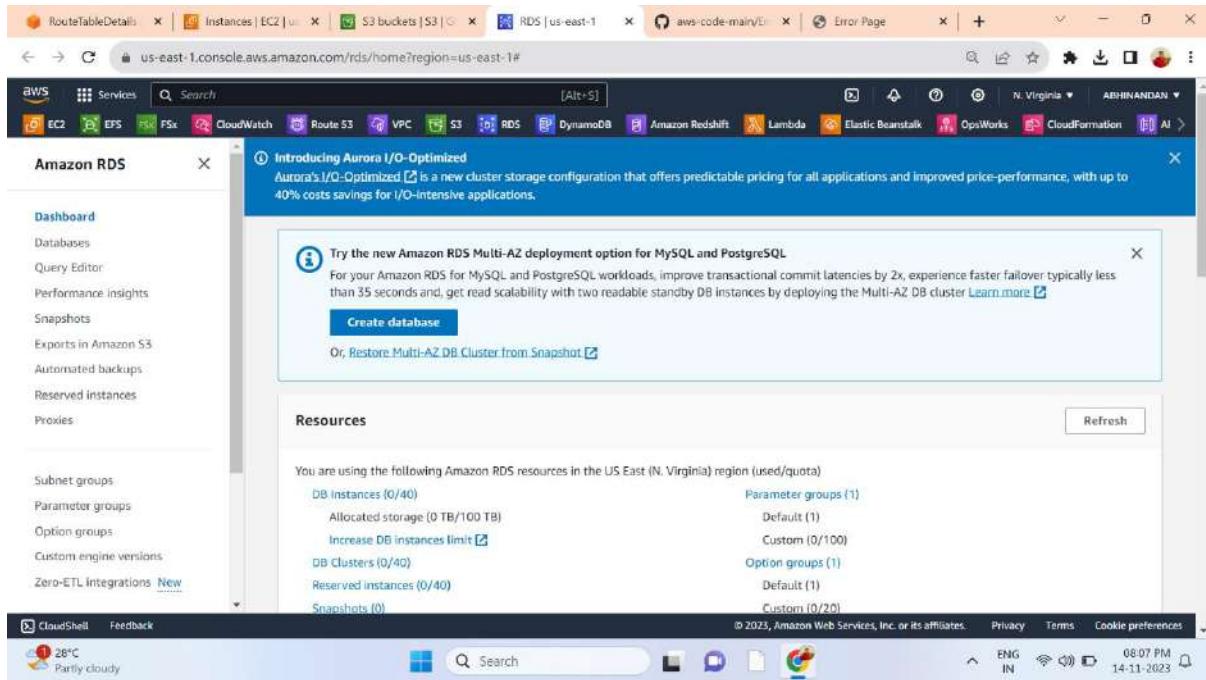
## Contact Us

If you have any questions or inquiries, feel free to contact us at [provide contact information]. We'd love to hear from you!

After pasting an http link in the web browser. We will see the About Us page.



If you pass any randomly generated data in the web browser, you will see the error page.



Now we will be creating an RDS database. Go to RDS in the AWS management console. go to the databases option in the navigation bar.

The screenshot shows the AWS RDS console in the N. Virginia region. The left sidebar includes options like Dashboard, Databases (which is selected), Query Editor, Performance insights, Snapshots, Exports in Amazon S3, Automated backups, Reserved instances, Proxies, Subnet groups, Parameter groups, Option groups, Custom engine versions, and Zero-ETL Integrations. A banner at the top says "Introducing Aurora I/O-Optimized". The main content area shows a table titled "Databases (0)" with columns for DB identifier, Status, Role, Engine, Region & AZ, Size, Actions, CPU, and Current activity. A message box suggests creating a Blue/Green deployment. The status bar at the bottom indicates it's 08:09 PM on 14-11-2023.

Click on Create a Database.

The screenshot shows the "Create database" wizard. It starts with a choice between "Standard create" and "Easy create". "Standard create" is selected. Below this, under "Engine options", "MySQL" is selected. A sidebar on the right provides detailed information about MySQL, stating it's the most popular open source database and listing its features: supports database sizes up to 64 TiB, general purpose, memory optimized, and burstable performance instance classes, automated backup and point-in-time recovery, and up to 15 read replicas per instance. The status bar at the bottom indicates it's 08:11 PM on 14-11-2023.

Choose an option standard create. engine type: MYSQL.

The screenshot shows the AWS RDS console with the URL [us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#launch-dbinstance](https://us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#launch-dbinstance). The interface includes a top navigation bar with tabs for RouteTableDetails, Instances, S3 buckets, RDS, aws-code-mail, and Error Page. Below the navigation is a service menu with EC2, EBS, FSx, CloudWatch, Route 53, VPC, S3, RDS, DynamoDB, Amazon Redshift, Lambda, Elastic Beanstalk, OpsWorks, CloudFormation, and AI.

The main content area displays options for launching a new DB instance. It shows two radio buttons: "Show versions that support the Multi-AZ DB cluster" (selected) and "Show versions that support the Amazon RDS Optimized Writes". A dropdown menu for "Engine Version" is set to MySQL 8.0.33. Below this is a "Templates" section with three options: "Production" (selected), "Dev/Test", and "Free tier". The "Free tier" option is described as being intended for development, testing, or proof-of-concept use. To the right of the templates is a sidebar with information about MySQL, including its popularity, features like automated backup and point-in-time recovery, and supported instance classes (General Purpose, Memory Optimized, and Burstable Performance).

Tempaltes – Free tier.

The screenshot shows the AWS RDS console with the URL [us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#launch-dbinstance](https://us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#launch-dbinstance). The interface is similar to the previous screenshot, with the same service menu and navigation bar.

The main content area now shows the configuration of a new DB instance. It includes fields for "Master username" (set to "admin") and "Master password" (set to "admin123"). There are also checkboxes for "Manage master credentials in AWS Secrets Manager" and "Auto generate a password". A note indicates that if master credentials are managed in Secrets Manager, some RDS features won't be supported. The right sidebar continues to provide information about MySQL's features and capabilities.

The master username is "admin" and the master password is "admin123".

The screenshot shows the AWS RDS console for launching a new DB instance. In the 'Allocated storage' field, '20 GiB' is selected. A note states: 'After you modify the storage for a DB instance, the status of the DB instance will be in storage-optimization. Your instance will remain available as the storage-optimization operation completes.' Below this, under 'Storage autoscaling', there is a checkbox for 'Enable storage autoscaling' which is unchecked. On the right side, there is a detailed description of MySQL and a bulleted list of its features.

MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

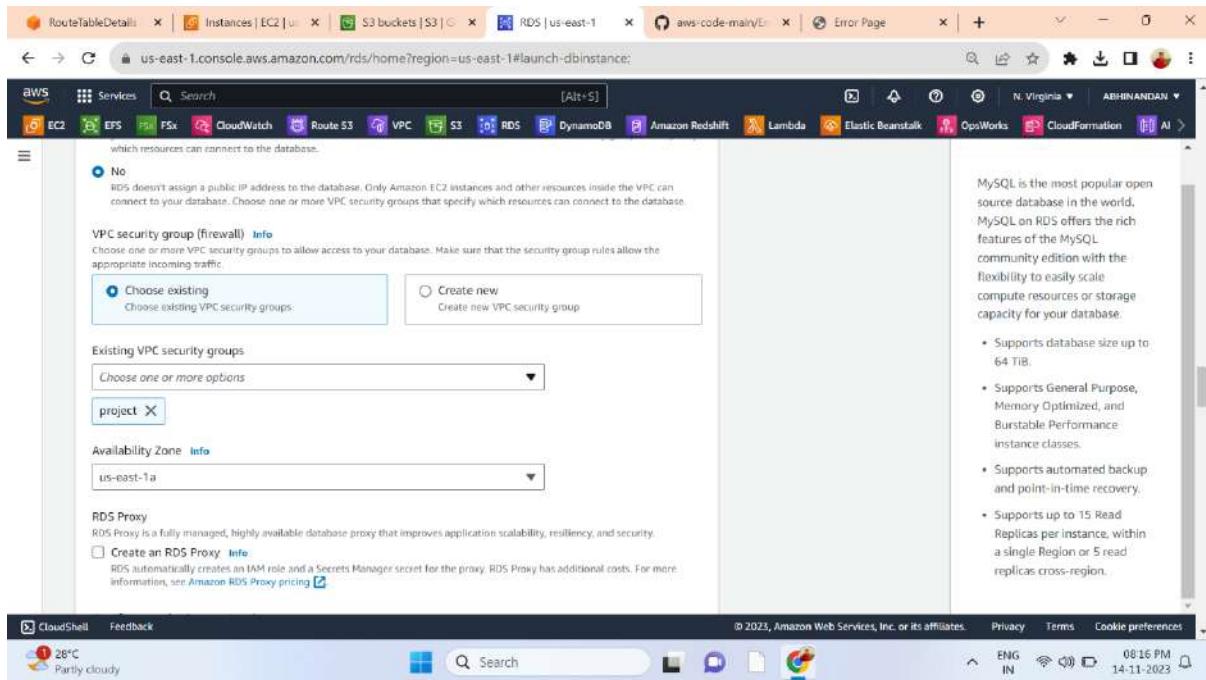
Disable storage autoscaling.

The screenshot shows the 'Connectivity' section of the RDS launch wizard. Under 'Compute resource', the 'Don't connect to an EC2 compute resource' option is selected. Under 'Virtual private cloud (VPC)', the 'vpc-project (vpc-0fe01bf2066565488)' VPC is chosen. A note states: 'After a database is created, you can't change its VPC.' Under 'DB subnet group', the 'vpc-project' subnet group is selected. The right side of the screen displays the MySQL database details and its features.

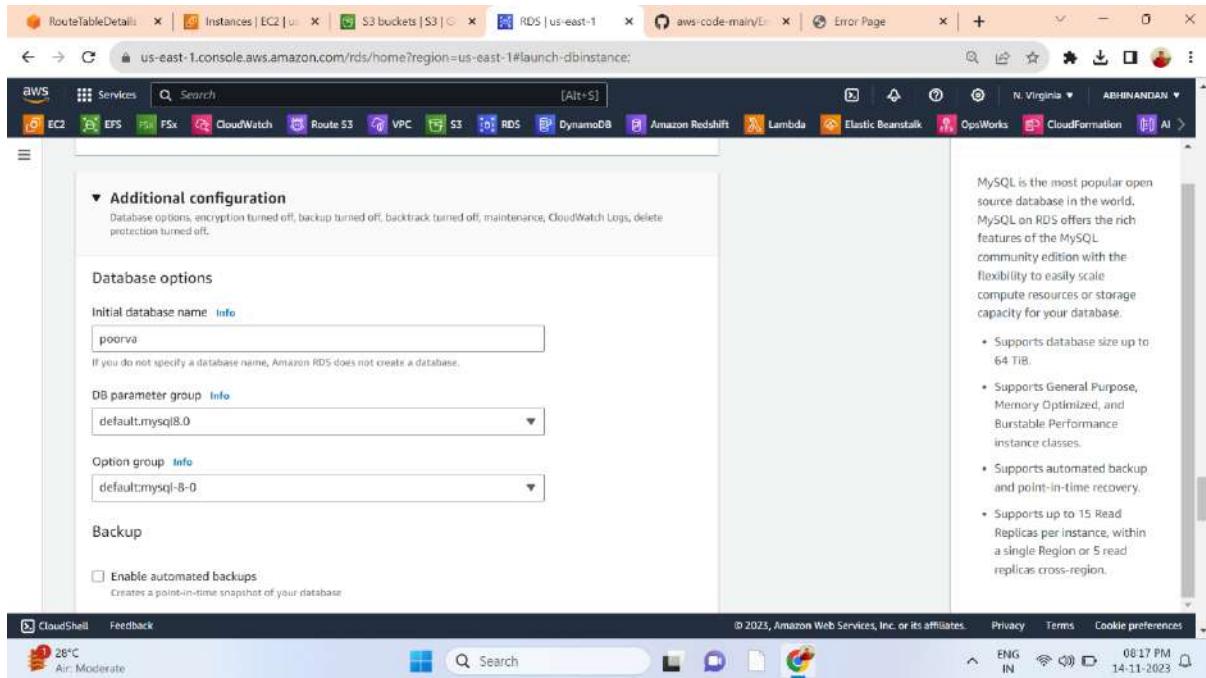
MySQL is the most popular open source database in the world. MySQL on RDS offers the rich features of the MySQL community edition with the flexibility to easily scale compute resources or storage capacity for your database.

- Supports database size up to 64 TiB.
- Supports General Purpose, Memory Optimized, and Burstable Performance instance classes.
- Supports automated backup and point-in-time recovery.
- Supports up to 15 Read Replicas per instance, within a single Region or 5 read replicas cross-region.

In the VPC section, choose the VPC that we created for our project. Name: VPC-project.



choose our existing VPC security groups. Choose Availability Zone – us-east-1a.



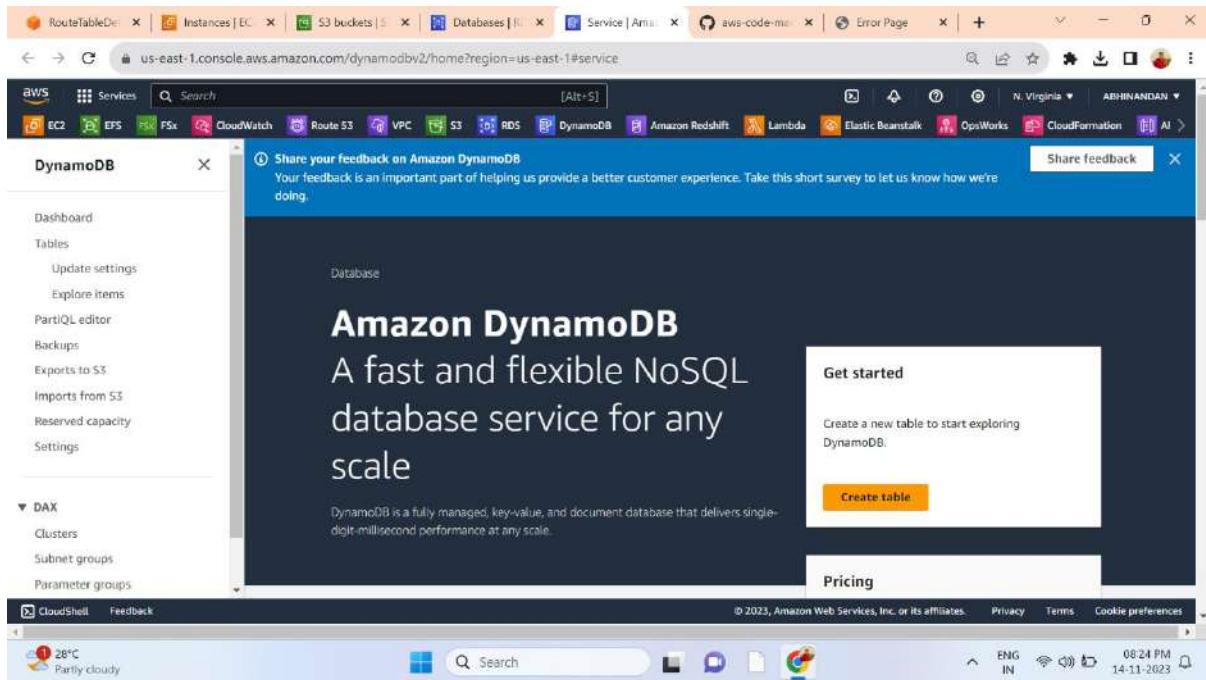
Click on additional configuration. database options; initial database name; poorva. Disable Backup.

The screenshot shows the AWS RDS console with the URL [us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#launch-dbinstance](https://us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#launch-dbinstance). The page is titled 'Create database'. It includes sections for 'Estimated monthly costs' (describing the Free Tier), a note about third-party rights, and a summary of MySQL features. At the bottom, there are 'Cancel' and 'Create database' buttons.

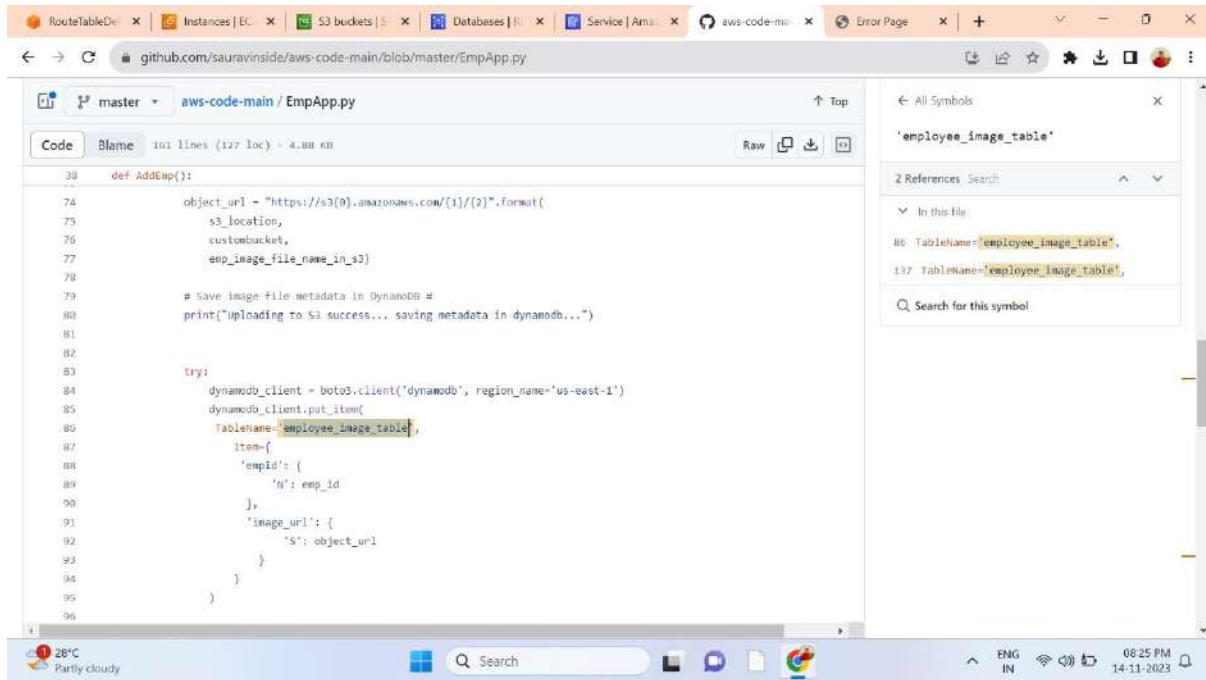
Now Click on Create database.

The screenshot shows the AWS RDS console with the URL [us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#databases](https://us-east-1.console.aws.amazon.com/rds/home?region=us-east-1#databases). A green success message box says 'Successfully created database database-1'. Below it, the 'Databases' table shows one entry: 'database-1' (Status: Available, Instance Type: db.t3.micro). The table has columns for DB identifier, Status, Role, Engine, Region & AZ, Size, Actions, CPU, and GPU.

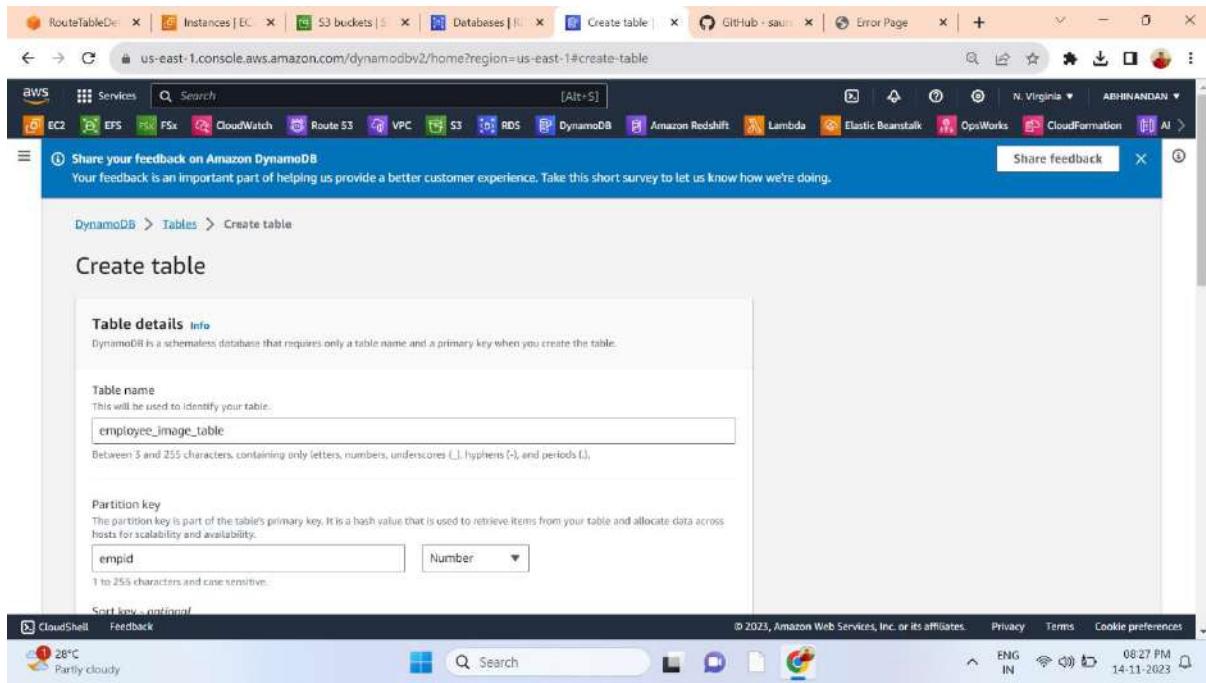
Now we Successfully Created a RDS database.



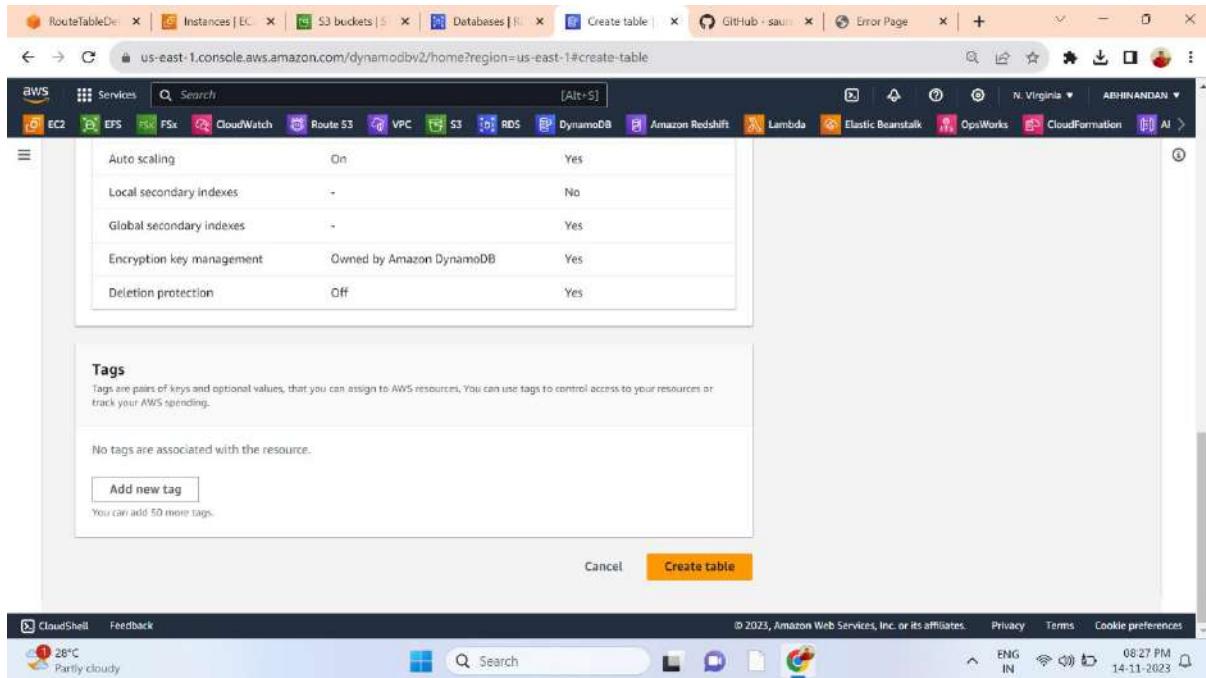
Now let's create a Dynamo DB in the AWS management console. Click On Create table.



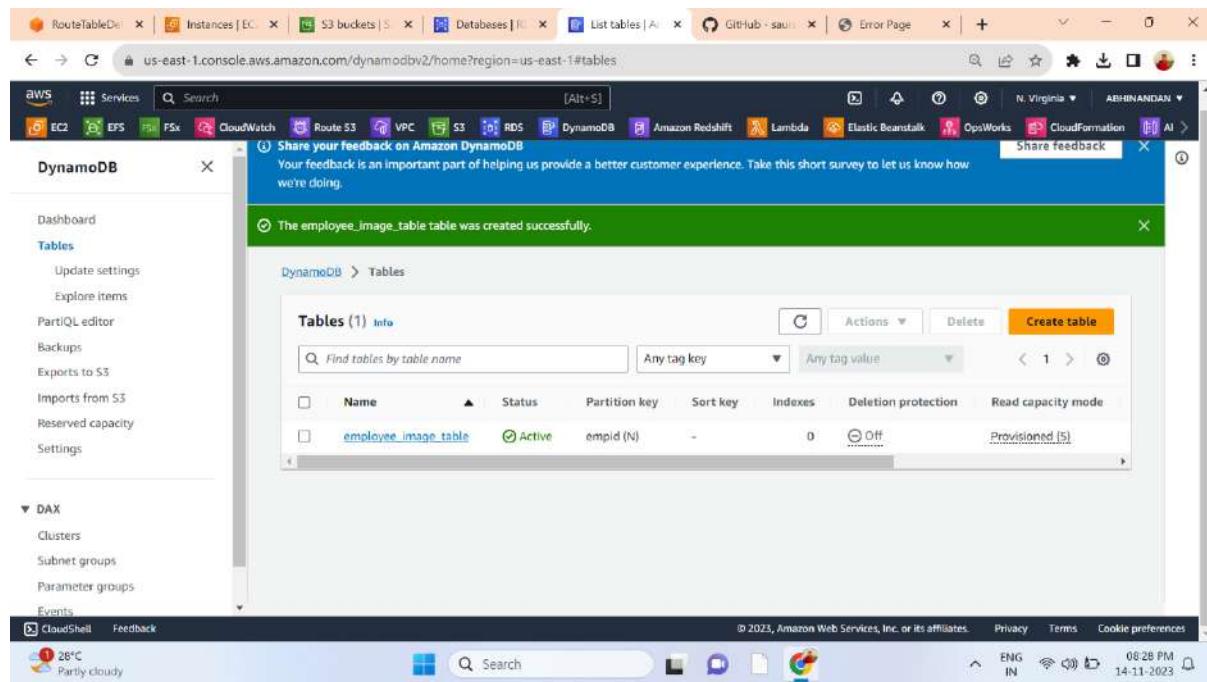
Now open the github link that is provided by the Intellipaat instructor and open the code. and copy the table name from the code.



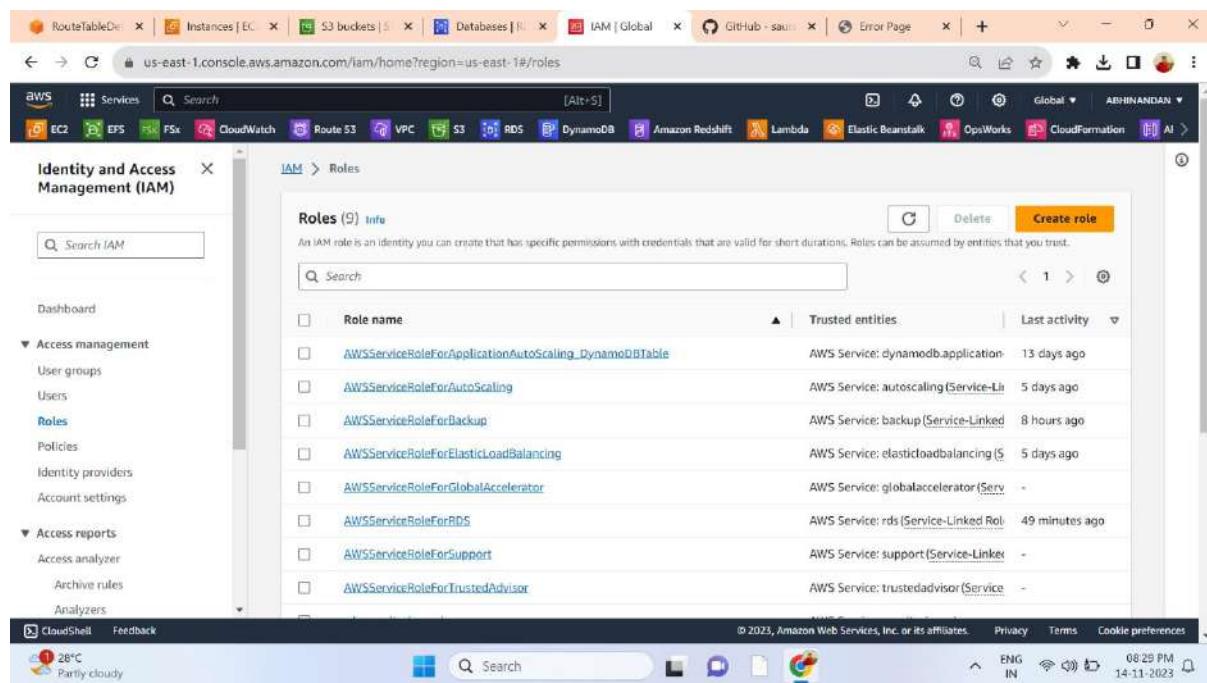
and paste it in the table details. partition key, empid, and number.



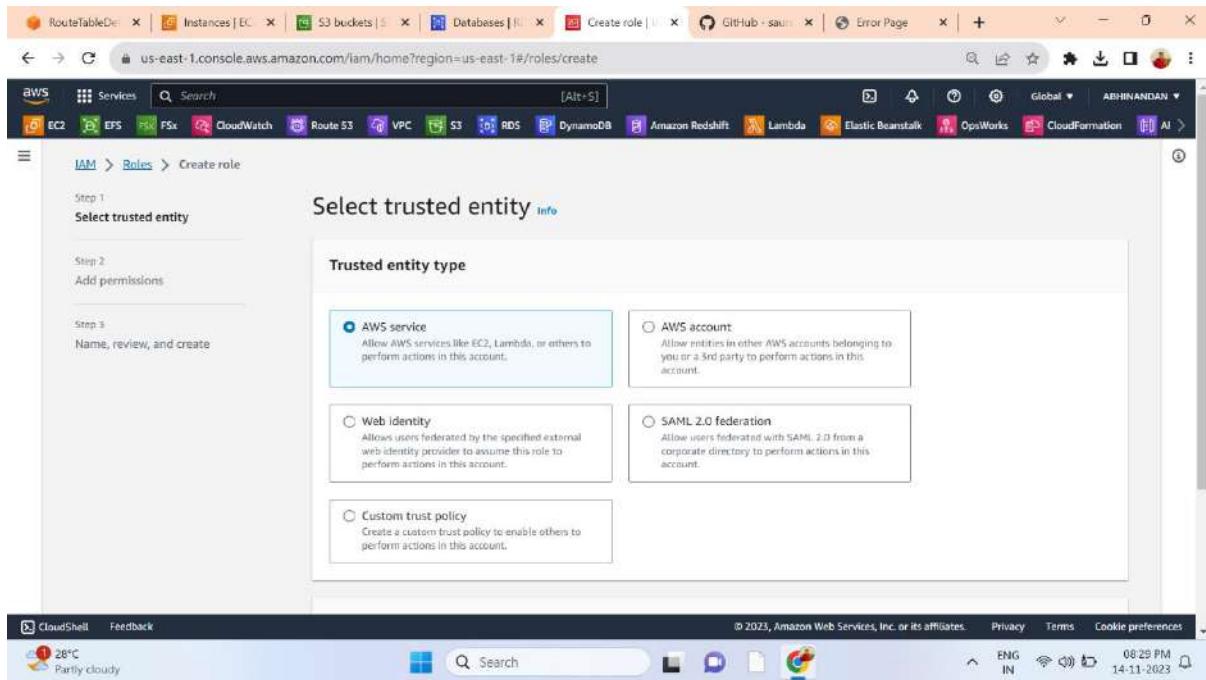
Click on Create Table.



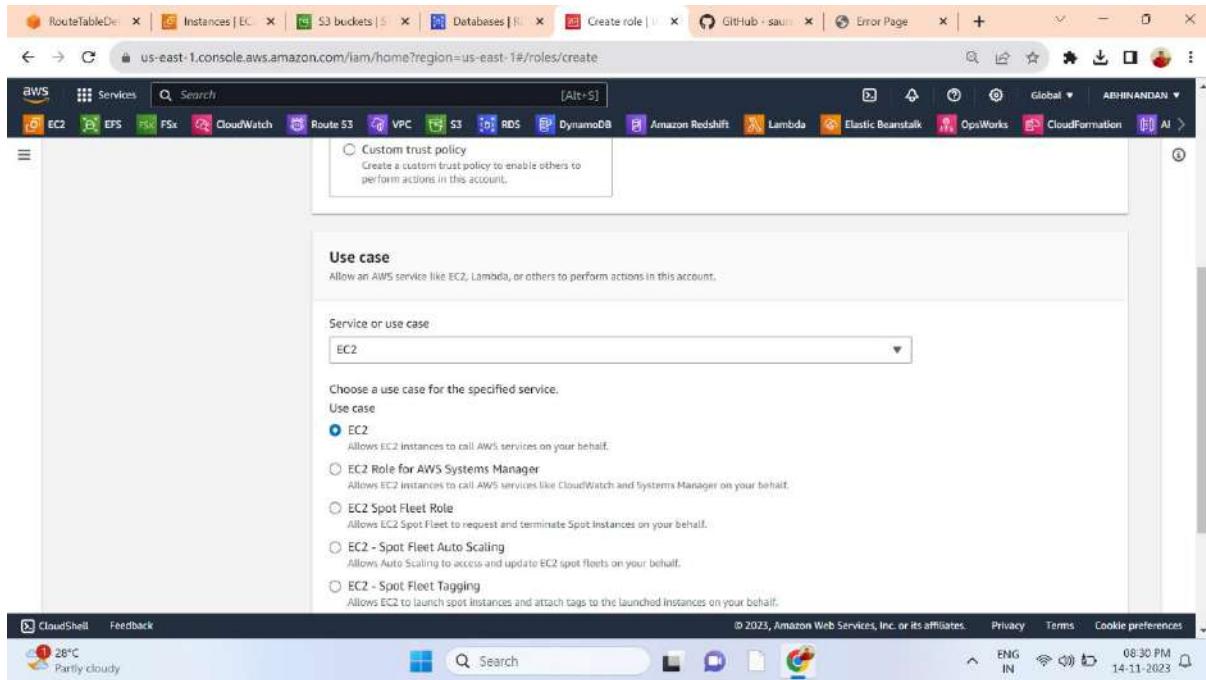
Now we have successfully created a Dynamo DB table.



Now we will create an IAM role. open IAM in the AWS management console. In the navigation bar, select roles. Click on Create a Role.



choose trusted entity type as an AWS service.



In Use Case, choose the EC2 service.

Step 1  
Select trusted entity

Step 2  
Add permissions

Step 3  
Name, review, and create

Name, review, and create

Role details

Role name  
Enter a meaningful name to identify this role.  
myproject-role

Description  
Add a short explanation for this role.  
Allows EC2 instances to call AWS services on your behalf.

Step 1: Select trusted entities

Trust policy

Now give a Role name – myproject-role.

Step 2: Add permissions

Permissions policy summary

Policy name	Type	Attached as
AmazonDynamoDBFullAccess	AWS managed	Permissions policy
AmazonRDSFullAccess	AWS managed	Permissions policy
AmazonS3FullAccess	AWS managed	Permissions policy

Step 3: Add tags

Create role

Here are the service accesses: AmazonS3fullAccess, AmazonRDSfullAccess, and AmazonDynamoDBfullAccess. Now Click on Create role.

The screenshot shows the AWS EC2 Instances console. There are two instances listed: 'public-project' (Running, t2.micro) and 'private' (Running, t2.micro). The 'private' instance has a context menu open under the 'Actions' dropdown. The 'Security' option is highlighted in the menu.

Now we came back to our instance console. Now I will select a private instance. above, go to actions go to security. go to modify the IAM role.

The screenshot shows the 'Modify IAM role' dialog box. The 'Instance ID' field is set to 'i-01795d888688a46fd (private)'. The 'IAM role' dropdown is set to 'myproject-role'. At the bottom, there are 'Cancel' and 'Update IAM role' buttons.

Select myproject-role and click on Update IAM role.

RouteTableDe x Load balance: x S3 buckets | x Databases | x Roles | IAM | x GitHub - saurabh | x Error Page | +

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#LoadBalancers:

Services Search [Alt+S]

N. Virginia ABHINANDAN

EC2 EPS FSx CloudWatch Route 53 VPC S3 RDS DynamoDB Amazon Redshift Lambda Elastic Beanstalk OpsWorks CloudFormation API G

AMI Catalog

Elastic Block Store

Volumes Snapshots Lifecycle Manager

Network & Security

Security Groups Elastic IPs Placement Groups Key Pairs Network Interfaces

Load Balancing

Load Balancers Target Groups

Auto Scaling

Auto Scaling Groups

CloudShell Feedback

28°C Partly cloudy

Search

Actions Create load balancer

No resources to display

0 load balancers selected

Select a load balancer above.

Click on the load balancer. Click on Create a Load Balancer.

RouteTableDe x Compare and x S3 buckets | x Databases | x Roles | IAM | x GitHub - saurabh | x Error Page | +

us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#SelectCreateELBWizard:

Services Search [Alt+S]

N. Virginia ABHINANDAN

EC2 EPS FSx CloudWatch Route 53 VPC S3 RDS DynamoDB Amazon Redshift Lambda Elastic Beanstalk OpsWorks CloudFormation API G

Application Load Balancer info

Choose an Application Load Balancer when you need a flexible feature set for your applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application

Network Load Balancer info

Choose a Network Load Balancer when you need ultra-high performance, TLS offloading at scale, centralized certificate deployment, support for UDP, and static IP addresses for your applications. Operating at the connection level, Network Load

Gateway Load Balancer info

Choose a Gateway Load Balancer when you need to deploy and manage a fleet of third-party virtual appliances that support GENEVE. These appliances enable you to improve security, compliance, and policy controls.

Create

CloudShell Feedback

28°C Partly cloudy

Search

Click on Create an Application Load Balancer.

The screenshot shows the AWS CloudWatch Metrics console with a list of metrics. The metrics listed are:

- RouteTableDeletion
- Create application
- S3 buckets
- Databases
- Roles | IAM
- Github - saurabh
- Error Page

The browser tab is titled "us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateALBWizard". The AWS navigation bar at the top includes services like EC2, EBS, FSx, CloudWatch, Route 53, VPC, S3, RDS, DynamoDB, Amazon Redshift, Lambda, Elastic Beanstalk, OpsWorks, CloudFormation, and API G.

Give load balancer name, project, LB.

The screenshot shows the AWS CloudWatch Metrics console with a list of metrics. The metrics listed are:

- RouteTableDeletion
- Create application
- S3 buckets
- Databases
- Roles | IAM
- Github - saurabh
- Error Page
- Start Count

The browser tab is titled "us-east-1.console.aws.amazon.com/ec2/home?region=us-east-1#CreateALBWizard". The AWS navigation bar at the top includes services like EC2, EBS, FSx, CloudWatch, Route 53, VPC, S3, RDS, DynamoDB, Amazon Redshift, Lambda, Elastic Beanstalk, OpsWorks, CloudFormation, and API G.

In network mapping. Select our VPC named "VPC-project". Mappings: select AZ's us-east-1a, us-east-1b

The screenshot shows the AWS CloudFront Create ALB Wizard - Step 2: Set up security groups and listeners. The top navigation bar includes tabs for RouteTable, Create app, S3 bucket, Database, Roles, GitHub, Error Page, Start Count, and others. The main content area has two sections:

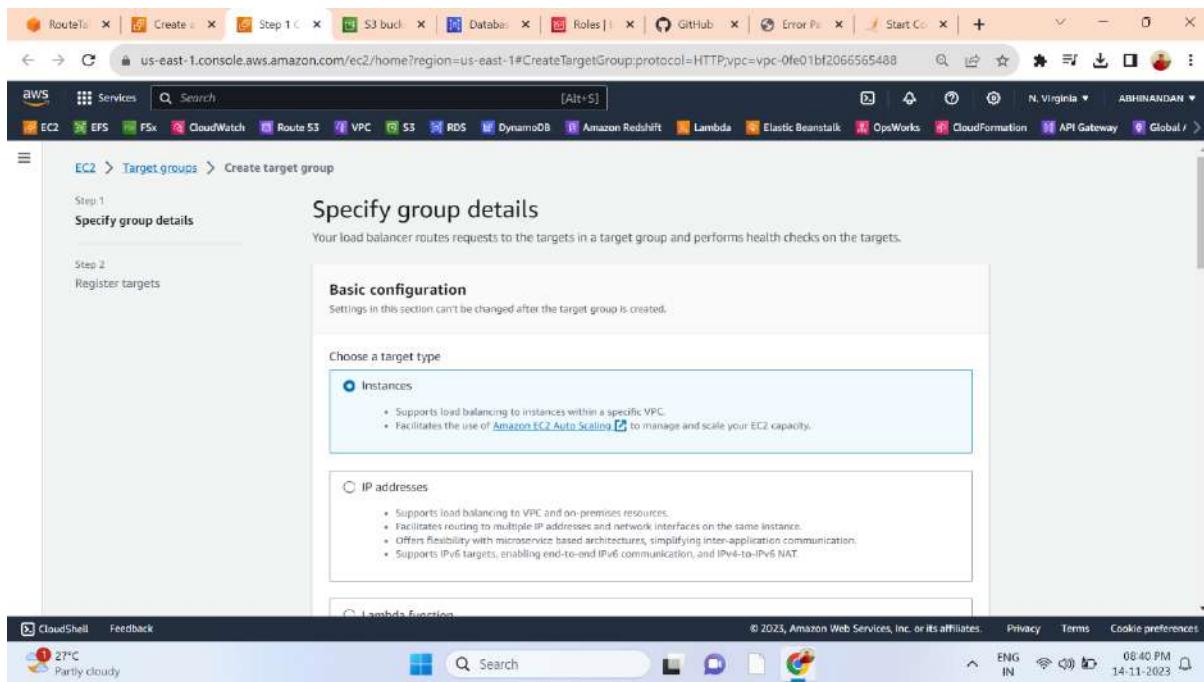
- Security groups**: A dropdown menu titled "Select up to 5 security groups" contains one item: "project".
- Listeners and routing**: A section titled "Listener: HTTP:80" shows a "Protocol" dropdown set to "HTTP" and a "Port" dropdown set to "80". The "Default action" dropdown is set to "Forward to Select a target group". A "Create target group" button is visible.

The bottom of the screen shows the Windows taskbar with icons for CloudShell, Feedback, Search, File Explorer, and Google Chrome. Weather information (28°C, Partly cloudy) and system status (ENG IN, 08:38 PM, 14-11-2023) are also present.

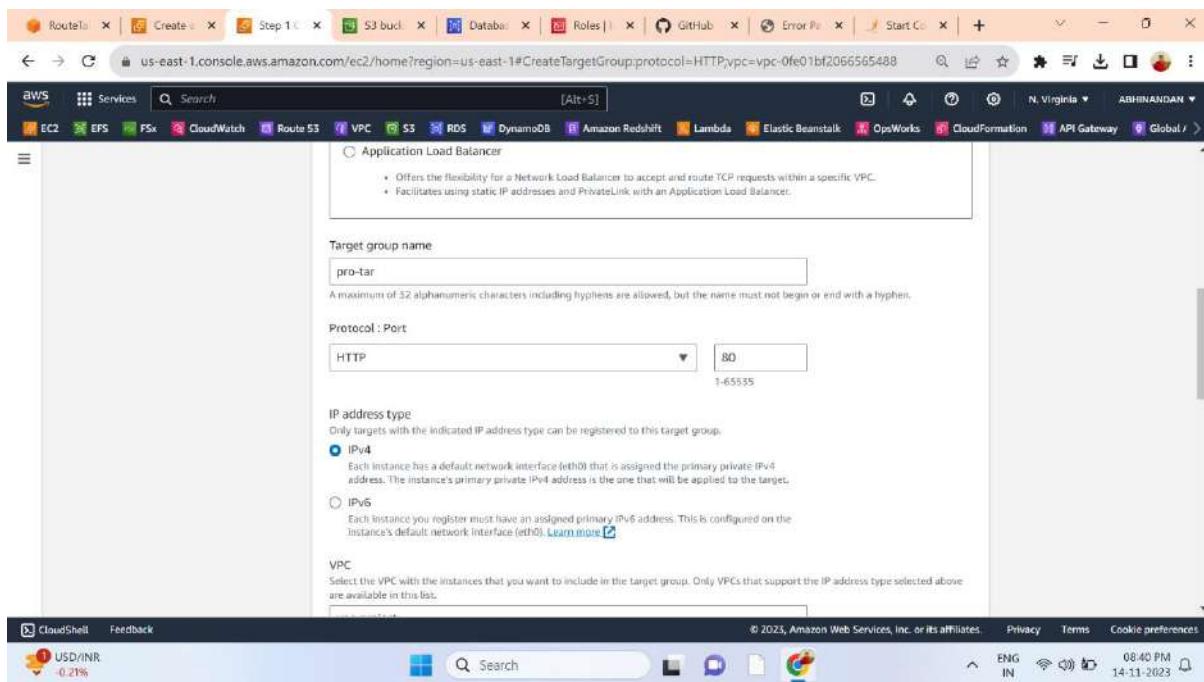
In security groups, select the security group that we created. The name is "project".

This screenshot is identical to the previous one, but the "Forward to" dropdown in the Listener section is now expanded, showing the "Select a target group" option. The rest of the interface and the system status at the bottom remain the same.

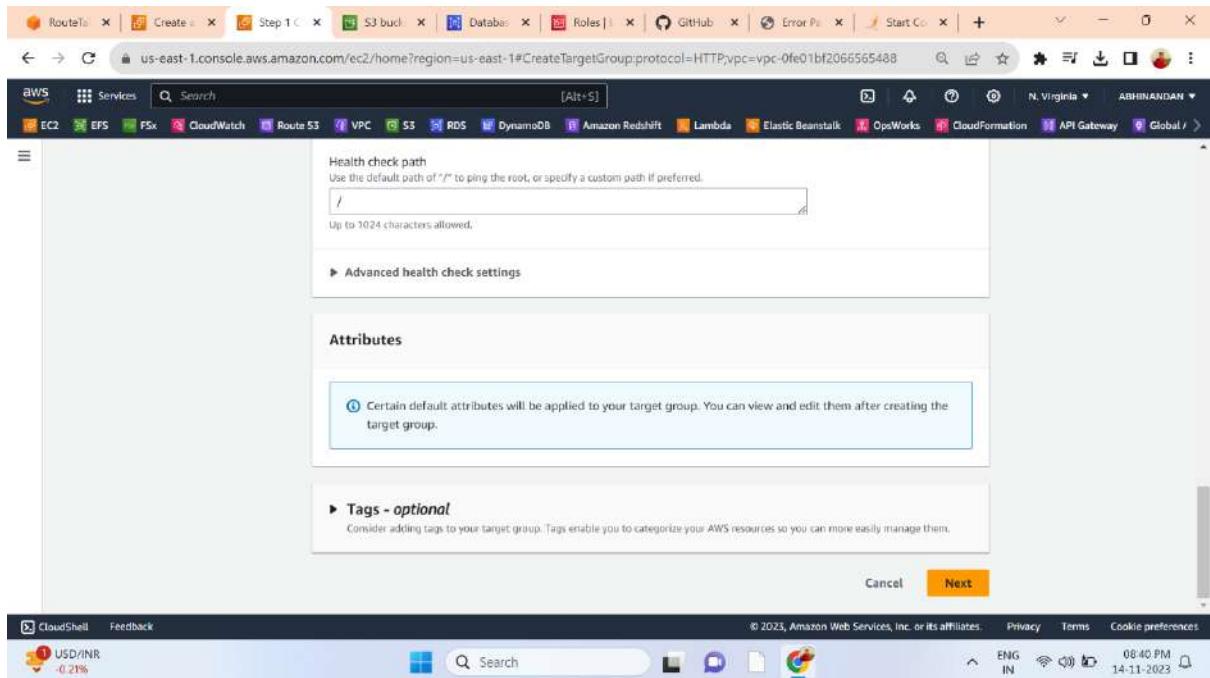
in listener and routing. go to create a target group option.



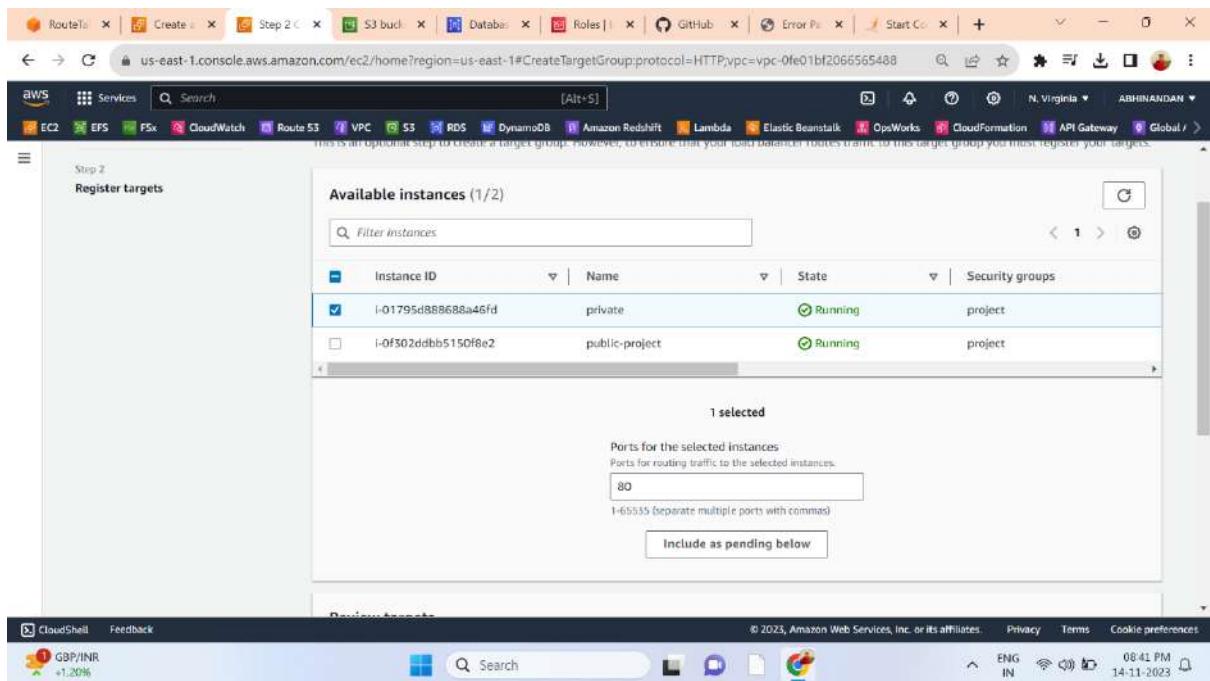
Choose the instances option only.



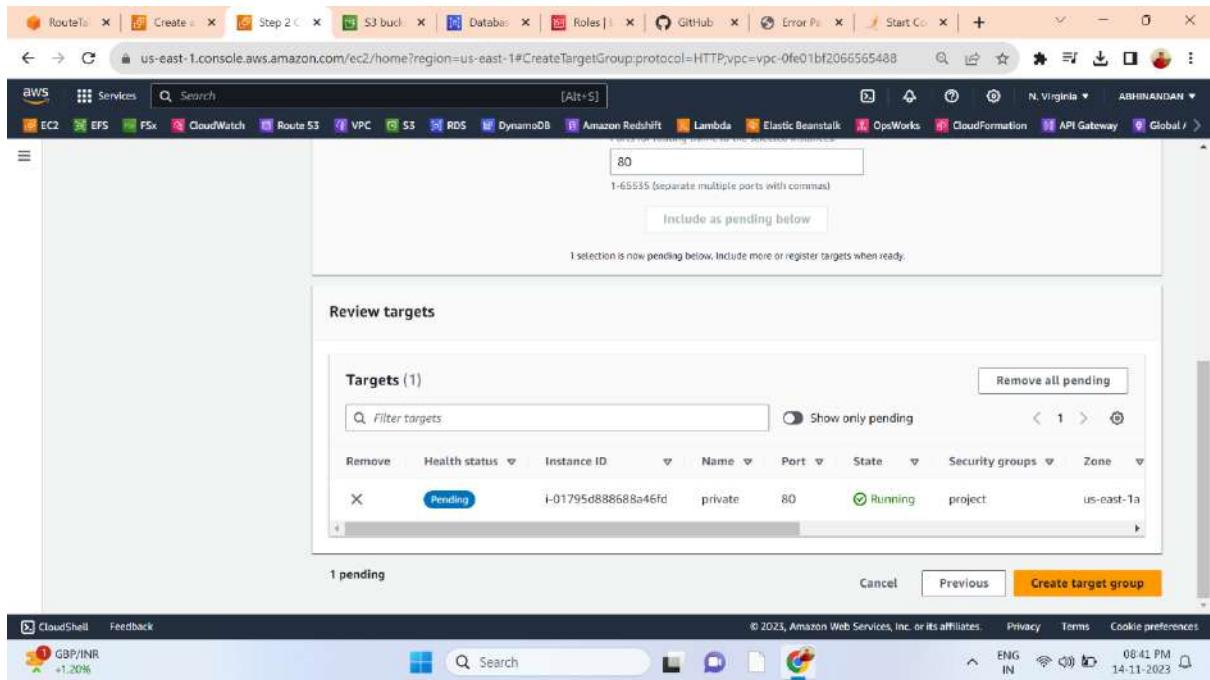
Give the target group name as "pro-tar".



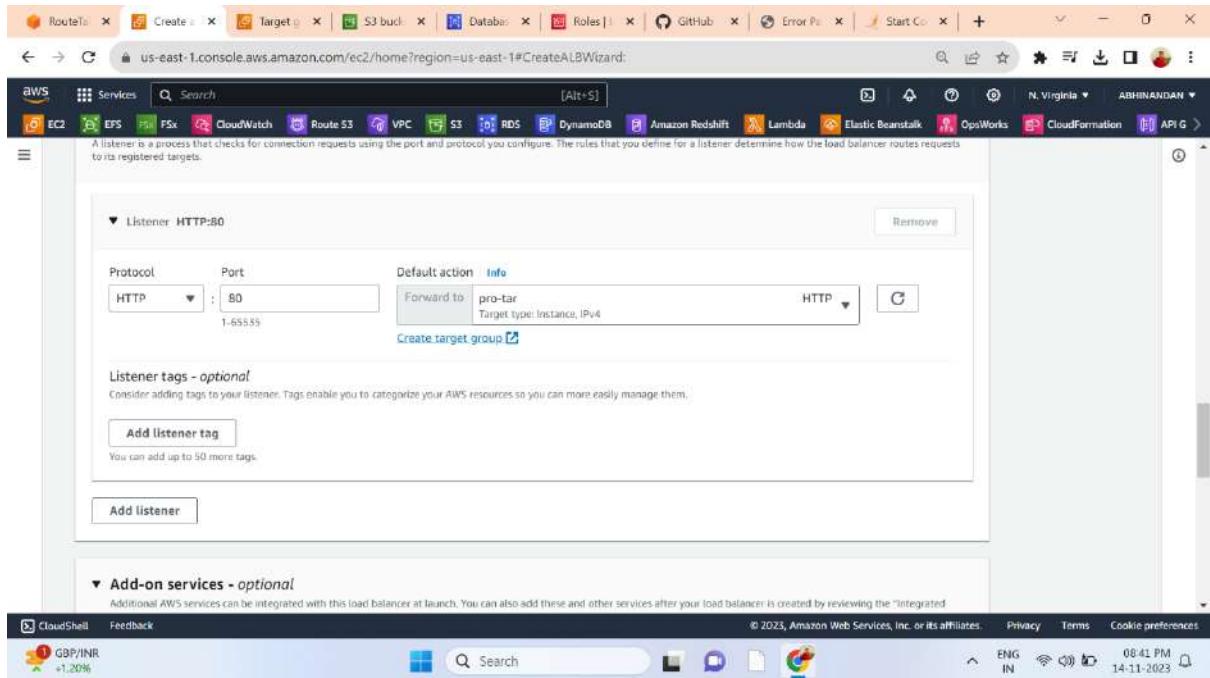
Click On Next.



In the available instances, choose a private instance only. Click on Include as Pending below.



Click on Create target group.



will go back to the load balancer tab. choose target-group name "pro-tar".

The screenshot shows the AWS CloudShell interface with a terminal window open. The command entered is:

```
aws elbv2 create-load-balancer --name project-LB --subnets subnet-03bd30d0fc4f21e10 subnet-0b2cera82444f94ee2 --security-groups sg-0783509be9d07a354 --vpc-id vpc-0fe01bf20665565480 --idle-timeout 60 --http-headers "Content-Type: application/json" --listen-ports 80 --load-balancer-type application
```

Click on Create load balancer.

The screenshot shows the AWS EC2 Load Balancers page. The table displays the following information for the 'project-LB' load balancer:

Name	DNS name	State	VPC ID	Availability Zones	Type
project-LB	project-LB-1613434902.us...	Active	vpc-0fe01bf206655654...	2 Availability Zones	application

Now Load balancer has been successfully.

The screenshot shows the AWS EC2 Instances console. On the left, a sidebar lists various EC2-related options like EC2 Dashboard, Instances, Images, and Elastic Block Store. The main area displays a table of instances. One instance, named 'public-project' with the ID i-0f302ddbb5150f8e2, is highlighted and expanded. This expanded view shows detailed information such as Public IPv4 address (34.226.195.154), Instance state (Running), and Private IPv4 addresses (20.0.1.123). At the bottom right of the expanded view, there are tabs for Details, Security, Networking, Storage, Status checks, Monitoring, and Tags.

Now we come back to our instance console. choose a public-project named instance.

The screenshot shows the AWS EC2 Instance Connect console. It displays connection details for the instance i-0f302ddbb5150f8e2. Under 'Connection Type', the 'Connect using EC2 Instance Connect' option is selected. Other fields include 'Public IP address' (34.226.195.154) and 'User name' (ubuntu). A note at the bottom states: 'Note: In most cases, the default user name, ubuntu, is correct. However, read your AMI usage instructions to check if the AMI owner has changed the default AMI user name.' At the bottom right are 'Cancel' and 'Connect' buttons.

Click on Connect.

```

Route 53 | Instances | EC2 Instances | Target groups | S3 buckets | Databases | Roles | IAM | GitHub | Error | Start Count | +
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0f302ddbb5150f8e2&osUser=ubuntu&r... | + | - | X

AWS Services Search [Alt+S]
EC2 | EPS | FSx | CloudWatch | Route 53 | VPC | S3 | RDS | DynamoDB | Amazon Redshift | Lambda | Elastic Beanstalk | OpsWorks | CloudFormation | API G |
Expanded Security Maintenance for Applications is not enabled.
0 updates can be applied immediately.

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-20-20-1-123:~$ i-0f302ddbb5150f8e2 (public-project)
PublicIPs: 34.226.195.154 PrivateIPs: 20.20.1.123

```

CloudShell Feedback © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences Top events Event brief ENG IN 09:00 PM 14-11-2023

Now we successfully connected a EC2 instance.

```

Instances | EC2 Instances | Target groups | S3 buckets | Databases | Roles | IAM | GitHub | Error | Start Count | +
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0f302ddbb5150f8e2&osUser=ubuntu&r... | + | - | X

AWS Services Search [Alt+S]
EC2 | EPS | FSx | CloudWatch | Route 53 | VPC | S3 | RDS | DynamoDB | Amazon Redshift | Lambda | Elastic Beanstalk | OpsWorks | CloudFormation | API G |
See https://ubuntu.com/esm or run: sudo pro status

The list of available updates is more than a week old.
To check for new updates run: sudo apt update

Last login: Tue Nov 14 15:30:11 2023 from 18.206.107.28
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

ubuntu@ip-20-20-1-123:~$ sudo apt update
Hit:1 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [109 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:5 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:6 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:7 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:8 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:9 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:10 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 kB]
Get:11 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1159 kB]
Get:12 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [247 kB]

i-0f302ddbb5150f8e2 (public-project)
PublicIPs: 34.226.195.154 PrivateIPs: 20.20.1.123

```

CloudShell Feedback © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences 27°C Partly cloudy ENG IN 09:03 PM 14-11-2023

Update EC2 machine command is : \$ sudo apt update

```

Get:26 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 c-n-f Metadata [116 B]
Get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [27.8 kB]
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [16.4 kB]
Get:29 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 c-n-f Metadata [644 B]
Get:30 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 c-n-f Metadata [116 B]
Get:31 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [947 kB]
Get:32 http://security.ubuntu.com/ubuntu jammy-security/main amd64 c-n-f Metadata [11.4 kB]
Get:33 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [1085 kB]
Get:34 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [1085 kB]
Get:35 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [176 kB]
Get:36 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 c-n-f Metadata [520 B]
Get:37 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [793 kB]
Get:38 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [146 kB]
Get:39 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [16.8 kB]
Get:40 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [36.5 kB]
Get:41 http://security.ubuntu.com/ubuntu jammy-security/multiverse Translation-en [7060 B]
Get:42 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 c-n-f Metadata [260 B]
Fetched 28.2 MB in 5s (5531 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
51 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-20-20-1-123:~$ 
```

i-0f302ddb5150f8e2 (public-project)  
PublicIPs: 54.226.195.154 PrivateIPs: 20.20.1.123

Now its Updated.

```

Get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [27.8 kB]
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [16.4 kB]
Get:29 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 c-n-f Metadata [644 B]
Get:30 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 c-n-f Metadata [116 B]
Get:31 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [947 kB]
Get:32 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [1085 kB]
Get:33 http://security.ubuntu.com/ubuntu jammy-security/main amd64 c-n-f Metadata [11.4 kB]
Get:34 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [1085 kB]
Get:35 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [176 kB]
Get:36 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 c-n-f Metadata [520 B]
Get:37 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [793 kB]
Get:38 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [146 kB]
Get:39 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [16.8 kB]
Get:40 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [36.5 kB]
Get:41 http://security.ubuntu.com/ubuntu jammy-security/multiverse Translation-en [7060 B]
Get:42 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 c-n-f Metadata [260 B]
Fetched 28.2 MB in 5s (5531 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
51 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-20-20-1-123:~$ sudo su
root@ip-20-20-1-123:/home/ubuntu# 
```

i-0f302ddb5150f8e2 (public-project)  
PublicIPs: 54.226.195.154 PrivateIPs: 20.20.1.123

Now connect, command is : \$ sudo su

```

Get:27 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 Packages [27.8 kB]
Get:28 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe Translation-en [16.4 kB]
Get:29 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/universe amd64 c-n-f Metadata [644 B]
Get:30 http://us-east-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/multiverse amd64 c-n-f Metadata [116 B]
Get:31 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [947 kB]
Get:32 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [107 kB]
Get:33 http://security.ubuntu.com/ubuntu jammy-security/main amd64 c-n-f Metadata [11.4 kB]
Get:34 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages [1085 kB]
Get:35 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-en [176 kB]
Get:36 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages [793 kB]
Get:37 http://security.ubuntu.com/ubuntu jammy-security/universe Translation-en [146 kB]
Get:38 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 c-n-f Metadata [16.8 kB]
Get:39 http://security.ubuntu.com/ubuntu jammy-security/multiverse amd64 Packages [36.5 kB]
Get:40 http://security.ubuntu.com/ubuntu jammy-security/multiverse Translation-en [7060 B]
Get:41 http://security.ubuntu.com/ubuntu jammy-security/multiverse Translation-en [260 B]
Fetched 28.2 MB in 5s (5531 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
51 packages can be upgraded. Run 'apt list --upgradable' to see them.
ubuntu@ip-20-20-1-123:~$ sudo su
root@ip-20-20-1-123:/home/ubuntu# nano key

```

i-0f302ddbb5150f8e2 (public-project)  
PublicIPs: 34.226.195.154 PrivateIPs: 20.20.1.123

Here we will create a file named “key” by using command nano key

```

Swap usage: 0%
* Ubuntu Pro delivers the most comprehensive open source security and
  compliance features.

https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

55 updates can be applied immediately.
35 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Tue Nov 14 15:32:15 2023 from 18.206.107.27
ubuntu@ip-20-20-1-123:~$ sudo su
root@ip-20-20-1-123:/home/ubuntu# nano key
root@ip-20-20-1-123:/home/ubuntu# ls
key
root@ip-20-20-1-123:/home/ubuntu#
```

i-0f302ddbb5150f8e2 (public-project)  
PublicIPs: 34.226.195.154 PrivateIPs: 20.20.1.123

Now we successfully copied a key pair content by copying it by notepad. Type ls command to view the created file “key”.

```
* Ubuntu Pro delivers the most comprehensive open source security and compliance features.

https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

55 updates can be applied immediately.
35 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Tue Nov 14 15:32:15 2023 from 18.206.107.27
ubuntu@ip-20-20-1-123:~$ sudo su
root@ip-20-20-1-123:/home/ubuntu# nano key
root@ip-20-20-1-123:/home/ubuntu# ls
key
root@ip-20-20-1-123:/home/ubuntu# chmod 400 key
root@ip-20-20-1-123:/home/ubuntu# ssh -i key ubuntu@20.20.3.146

i-0f302ddbb5150f8e2 (public-project)
PublicIPs: 34.226.195.154 PrivateIPs: 20.20.1.123
```

Now we will give permission for access by using command chmod 400 key.

```
* Ubuntu Pro delivers the most comprehensive open source security and compliance features.

https://ubuntu.com/aws/pro

Expanded Security Maintenance for Applications is not enabled.

55 updates can be applied immediately.
35 of these updates are standard security updates.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Tue Nov 14 15:32:15 2023 from 18.206.107.27
ubuntu@ip-20-20-1-123:~$ sudo su
root@ip-20-20-1-123:/home/ubuntu# nano key
root@ip-20-20-1-123:/home/ubuntu# ls
key
root@ip-20-20-1-123:/home/ubuntu# chmod 400 key
root@ip-20-20-1-123:/home/ubuntu# ssh -i key ubuntu@20.20.3.146

i-0f302ddbb5150f8e2 (public-project)
PublicIPs: 34.226.195.154 PrivateIPs: 20.20.1.123
```

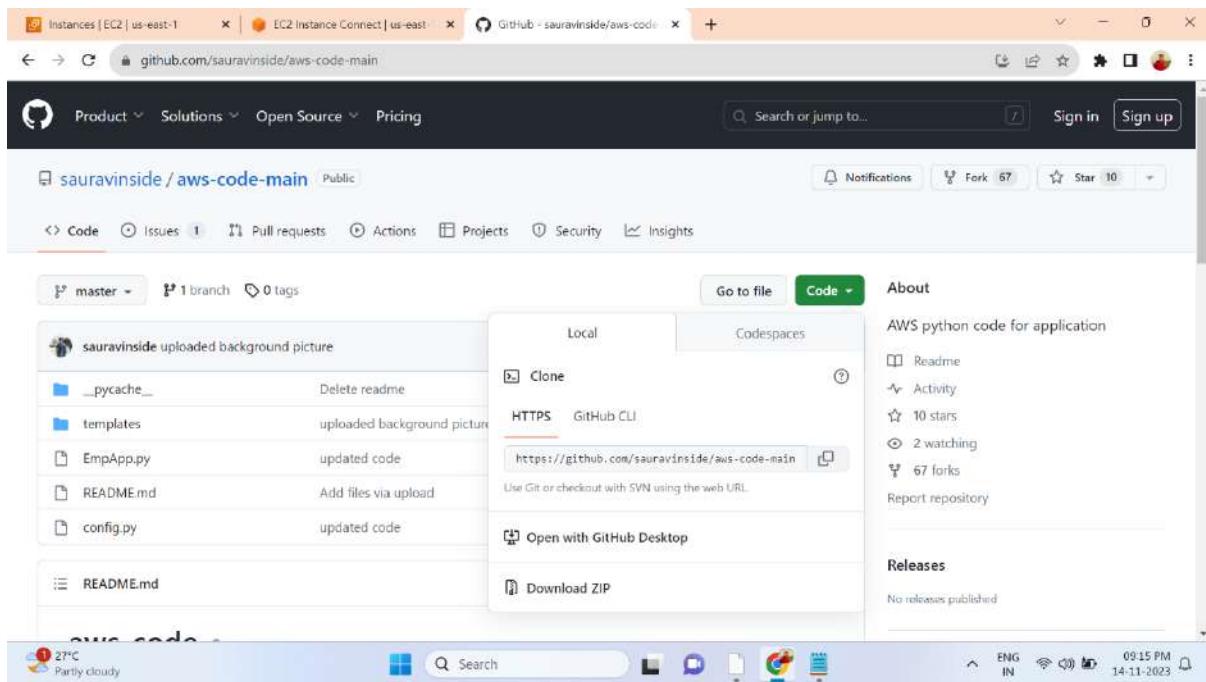
Now we will connect private instance using public instance, command is ssh -I key ubuntu@20.20.3.146

```
Instances [EC2 | us-east-1] x EC2 Instance Connect [us-east-1] x GitHub - sauravinside/aws-code... +  
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0f302ddbb5150f8e2&osUser=ubuntu&region=N. Virginia  
AWS Services Search [Alt+S]  
EC2 EFS FSx CloudWatch Route 53 VPC S3 RDS DynamoDB Amazon Redshift Lambda Elastic Beanstalk OpsWorks CloudFormation API G >  
http://ubuntu.com/aws/pro  
Expanded Security Maintenance for Applications is not enabled.  
55 updates can be applied immediately.  
35 of these updates are standard security updates.  
To see these additional updates run: apt list --upgradable  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
Last login: Tue Nov 14 15:32:15 2023 from 18.206.107.27  
ubuntu@ip-20-20-1-123:~$ sudo su  
root@ip-20-20-1-123:/home/ubuntu# nano key  
root@ip-20-20-1-123:/home/ubuntu# ls  
key  
root@ip-20-20-1-123:/home/ubuntu# chmod 400 key  
root@ip-20-20-1-123:/home/ubuntu# ssh -i key ubuntu@20.20.3.140  
The authenticity of host '20.20.3.140 (20.20.3.140)' can't be established.  
ED25519 key fingerprint is SHA256:rM4x/sFeZxf3Ma/gXf7y+HzIPupiYu2jvq3vCMv0Gx4.  
This key is not known by any other names  
Are you sure you want to continue connecting (yes/no/(fingerprint))?  
  
i-0f302ddbb5150f8e2 (public-project)  
Public IPs: 34.226.195.154 Private IPs: 20.20.1.123
```

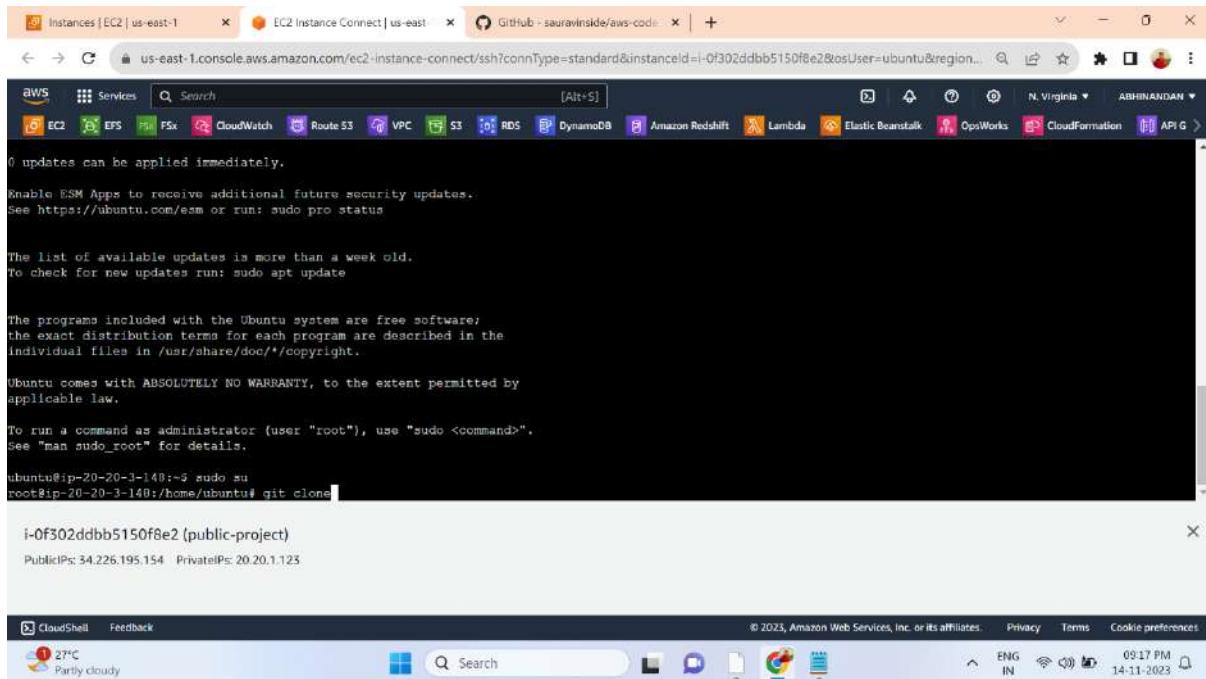
Now it ask for yes/no for connecting.

```
Instances [EC2 | us-east-1] x EC2 Instance Connect [us-east-1] x GitHub - sauravinside/aws-code... +  
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0f302ddbb5150f8e2&osUser=ubuntu&region=N. Virginia  
AWS Services Search [Alt+S]  
EC2 EFS FSx CloudWatch Route 53 VPC S3 RDS DynamoDB Amazon Redshift Lambda Elastic Beanstalk OpsWorks CloudFormation API G >  
http://ubuntu.com/aws/pro  
Expanded Security Maintenance for Applications is not enabled.  
0 updates can be applied immediately.  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/**/copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
ubuntu@ip-20-20-3-140:~$  
  
i-0f302ddbb5150f8e2 (public-project)  
Public IPs: 34.226.195.154 Private IPs: 20.20.1.123
```

Now we successfully connected our private instance.



Now we will copy the code link in github that is provided by intellipaat team.



Now connect to user. And give command git clone.

The terminal output shows the following steps:

```
0 updates can be applied immediately.  
Enable ESM Apps to receive additional future security updates.  
See https://ubuntu.com/esm or run: sudo pro status  
  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
ubuntu@ip-20-20-3-148:~$ sudo su  
root@ip-20-20-3-148:/home/ubuntu# git clone https://github.com/sauravinside/aws-code-main.git
```

Now paste the link that we copied from github code.

The terminal output shows the following steps:

```
individual files in /usr/share/doc/*copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
ubuntu@ip-20-20-3-148:~$ sudo su  
root@ip-20-20-3-148:/home/ubuntu# git clone https://github.com/sauravinside/aws-code-main.git  
Cloning into 'aws-code-main'...  
remote: Enumerating objects: 47, done.  
remote: Counting objects: 100% (47/47), done.  
remote: Compressing objects: 100% (39/39), done.  
remote: Total 47 (delta 16), reused 13 (delta 4), pack-reused 0  
Receiving objects: 100% (47/47), 651.63 KiB | 17.15 MiB/s, done.  
Resolving deltas: 100% (16/16), done.  
root@ip-20-20-3-148:/home/ubuntu# ls  
aws-code-main  
root@ip-20-20-3-148:/home/ubuntu#
```

i-0f302ddbb5150f8e2 (public-project)

Public IPs: 34.226.195.154 Private IPs: 20.20.1.123

Now we successfully added aws-code-main.

```
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
ubuntu@ip-20-20-3-148:~$ sudo su  
root@ip-20-20-3-148:/home/ubuntu# git clone https://github.com/sauravinside/aws-code-main.git  
Cloning into 'aws-code-main'...  
remote: Enumerating objects: 47, done.  
remote: Counting objects: 100% (47/47), done.  
remote: Compressing objects: 100% (39/39), done.  
remote: Total 47 (delta 16), reused 13 (delta 4), pack-reused 0  
Receiving objects: 100% (47/47), 651.63 KiB | 17.15 MiB/s, done.  
Resolving deltas: 100% (16/16), done.  
root@ip-20-20-3-148:/home/ubuntu# ls  
aws-code-main  
root@ip-20-20-3-148:/home/ubuntu# cd aws-code-main  
root@ip-20-20-3-148:/home/ubuntu/aws-code-main# ls  
EmpApp.py README.md __pycache__ config.py templates  
root@ip-20-20-3-148:/home/ubuntu/aws-code-main#
```

i-Of302ddbb5150f8e2 (public-project)  
PublicIPs: 34.226.195.154 PrivateIPs: 20.20.1.123

Now here we are changing directory to aws-code-main. And next list the files under aws-code-main file.

```
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
ubuntu@ip-20-20-3-148:~$ sudo su  
root@ip-20-20-3-148:/home/ubuntu# git clone https://github.com/sauravinside/aws-code-main.git  
Cloning into 'aws-code-main'...  
remote: Enumerating objects: 47, done.  
remote: Counting objects: 100% (47/47), done.  
remote: Compressing objects: 100% (39/39), done.  
remote: Total 47 (delta 16), reused 13 (delta 4), pack-reused 0  
Receiving objects: 100% (47/47), 651.63 KiB | 17.15 MiB/s, done.  
Resolving deltas: 100% (16/16), done.  
root@ip-20-20-3-148:/home/ubuntu# ls  
aws-code-main  
root@ip-20-20-3-148:/home/ubuntu# cd aws-code-main  
root@ip-20-20-3-148:/home/ubuntu/aws-code-main# ls  
EmpApp.py README.md __pycache__ config.py templates  
root@ip-20-20-3-148:/home/ubuntu/aws-code-main# cd  
root@ip-20-20-3-148:~#
```

i-Of302ddbb5150f8e2 (public-project)  
PublicIPs: 34.226.195.154 PrivateIPs: 20.20.1.123

Now again we will change the directory.

```
remote: Enumerating objects: 47, done.
remote: Counting objects: 100% (47/47), done.
remote: Compressing objects: 100% (39/39), done.
remote: Total 47 (delta 16), reused 13 (delta 4), pack-reused 0
Receiving objects: 100% (47/47), 651.63 KiB | 17.15 MiB/s, done.
Resolving deltas: 100% (16/16), done.
root@ip-20-20-3-148:/home/ubuntu# ls
aws-code-main
root@ip-20-20-3-148:/home/ubuntu# cd aws-code-main
root@ip-20-20-3-148:/home/ubuntu/aws-code-main# ls
Empapp.py README.md __pycache__ config.py templates
root@ip-20-20-3-148:/home/ubuntu/aws-code-main# cd
root@ip-20-20-3-148:# sudo apt-get install python3
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
python3 is already the newest version (3.10.6-1~22.04).
python3 set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@ip-20-20-3-148:#
```

i-0f302ddbb5150f8e2 (public-project)  
PublicIPs: 34.226.195.154 PrivateIPs: 20.20.1.123

Now we will install python by using this command \$ sudo apt-get install python3.

```
Building dependency tree... Done
Reading state information... Done
python3 is already the newest version (3.10.6-1~22.04).
python3 set to manually installed.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
root@ip-20-20-3-148:# sudo apt-get install python3-flask
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  javascript-common libjs-jquery python3-itsdangerous python3-pyinotify python3-simplejson python3-werkzeug
Suggested packages:
  apache2 | lighttpd | httpd python3-asigref python3-dotenv python3-flask-doc python-pyinotify-doc ipython3 python-werkzeug-doc python3-lxml python3-watchdog
The following NEW packages will be installed:
  javascript-common libjs-jquery python3-flask python3-itsdangerous python3-pyinotify python3-simplejson python3-werkzeug
0 upgraded, 7 newly installed, 0 to remove and 0 not upgraded.
Need to get 683 kB of archives.
After this operation, 2491 kB of additional disk space will be used.
Do you want to continue? [Y/n] 
```

i-0f302ddbb5150f8e2 (public-project)  
PublicIPs: 34.226.195.154 PrivateIPs: 20.20.1.123

Now again we will install python3 flask, \$ sudo apt-get install python3-flask

```
Instances [EC2 | us-east-1] x EC2 Instance Connect [us-east-1] x GitHub - sauravinside/aws-code... x +  
← → ⌂ https://us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0f302ddbb5150f8e2&osUser=ubuntu&region=us-east-1  
Scanning processes...  
Scanning processes...  
Scanning linux images... [=====  
Scanning linux images... [======  
Running kernel seems to be up-to-date.  
No services need to be restarted.  
No containers need to be restarted.  
No user sessions are running outdated binaries.  
No VM guests are running outdated hypervisor (qemu) binaries on this host.  
root@ip-20-20-3-148:~# sudo apt-get install python3-boto3  
i-0f302ddbb5150f8e2 (public-project)  
PublicIPs: 34.226.195.154 PrivateIPs: 20.20.1.123
```

Now we will install python3-boto3, sudo apt-get install python3-boto3

```
Instances [EC2 | us-east-1] x EC2 Instance Connect [us-east-1] x GitHub - sauravinside/aws-code... x +| ChatGPT x  
← → ⌂ https://us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0f302ddbb5150f8e2&osUser=ubuntu&region=us-east-1  
python3-flask is already the newest version (2.0.1-2ubuntu1.1).  
0 upgraded, 0 newly installed, 0 to remove and 51 not upgraded.  
root@ip-20-20-3-148:~# sudo apt-get install python3-pymysql  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
python3-pymysql is already the newest version (1.0.2-1ubuntu1).  
0 upgraded, 0 newly installed, 0 to remove and 51 not upgraded.  
root@ip-20-20-3-148:~# sudo apt-get install python3-boto3  
Reading package lists... Done  
Building dependency tree... Done  
Reading state information... Done  
The following additional packages will be installed:  
  python3-botocore python3-dateutil python3-jmespath python3-s3transfer  
The following NEW packages will be installed:  
  python3-boto3 python3-botocore python3-dateutil python3-jmespath python3-s3transfer  
0 upgraded, 5 newly installed, 0 to remove and 51 not upgraded.  
Need to get 4728 kB of archives.  
After this operation, 65.8 MB of additional disk space will be used.  
Do you want to continue? [Y/n]   
i-0f302ddbb5150f8e2 (public-project)  
PublicIPs: 34.226.195.154 PrivateIPs: 20.20.1.123
```

Now it will ask do you want to continue, click yes.

```
Instances [EC2 | us-east-1] x EC2 Instance Connect [us-east] x Github - sauravinside/aws-code... x Start Course | Intellipaat x +  
us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0f302ddbb5150f8e2&osUser=ubuntu&region=us-east-1  
AWS Services Search [Alt+S] N. Virginia ABHINANDAN  
EC2 EFS FSx CloudWatch Route 53 VPC S3 RDS DynamoDB Amazon Redshift Lambda Elastic Beanstalk OpsWorks Cloud...  
root@ip-20-20-3-148:~# ls  
snap  
root@ip-20-20-3-148:~# exit  
exit  
root@ip-20-20-3-148:~# exit  
exit  
ubuntu@ip-20-20-3-148:~$ ls  
aws-code-main  
ubuntu@ip-20-20-3-148:~$ cd aws-code-main/  
ubuntu@ip-20-20-3-148:~/aws-code-main$ ls  
RmPApp.py README.md __pycache__ config.py templates  
ubuntu@ip-20-20-3-148:~/aws-code-main$ sudo nano config.py  
  
i-0f302ddbb5150f8e2 (public-project)  
PublicIPs: 34.226.195.154 PrivateIPs: 20.20.1.123  
CloudShell Feedback © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences  
USD/INR 0.27% 0.27% 09:41 PM 14.11.2023
```

Now we will configure aws-code-main.

Instances [EC2] x EC2 Instance Conn x RDS | us-east-1 x S3 buckets | S3 G x GitHub - sauravini x Start Course | Intel x +

us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0f302ddbb5150f8e2&osUser=ubuntu&region=us-east-1

aws Services Search [Alt+S]

N. Virginia ▾ ABHINANDAN ▾

EC2 EFS FSx CloudWatch Route 53 VPC S3 RDS DynamoDB Amazon Redshift Lambda Elastic Beanstalk OpsWorks Cloud

Expanded Security Maintenance for Applications is not enabled.

55 updates can be applied immediately.  
35 of these updates are standard security updates.  
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.  
See <https://ubuntu.com/esm> or run: sudo pro status

Last login: Tue Nov 14 15:45:06 2023 from 20.20.1.123  
ubuntu@ip-20-20-3-148:~\$ ls  
aws-code-main  
ubuntu@ip-20-20-3-148:~\$ cd aws-code-main/  
ubuntu@ip-20-20-3-148:~/aws-code-main\$ ls  
EmpApp.py README.md \_pyapache config.py templates  
ubuntu@ip-20-20-3-148:~/aws-code-main\$ sudo nano config.py  
ubuntu@ip-20-20-3-148:~/aws-code-main\$ cd  
ubuntu@ip-20-20-3-148:~\$ sudo mysql -h database-1.c9nlafdcoteq.us-east-1.rds.amazonaws.com -u admin -p

i-0f302ddbb5150f8e2 (public-project)

PublicIPs: 34.226.195.154 PrivateIPs: 20.20.1.123

<https://us-east-1.console.aws.amazon.com/redshiftv2/home?region=us-east-1>

© 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

1 25°C Mostly clear

Search

ENGLISH IN 14-Nov-2023 09:55 PM

Now we will copy the RDS database endpoint.

```
Setting up mysql-common (5.8+1.0.8) ...
update-alternatives: using /etc/mysql/my.cnf.fallback to provide /etc/mysql/my.cnf (my.cnf) in auto mode
Setting up mysql-client-core-8.0 (8.0.35-Ubuntu0.22.04.1) ...
Setting up mysql-client-8.0 (8.0.35-Ubuntu0.22.04.1) ...
Setting up mysql-client (8.0.35-Ubuntu0.22.04.1) ...
Processing triggers for man-db (2.10.2-1) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-20-20-3-148:~$ sudo mysql -h database-1.c9nlafdcoteq.us-east-1.rds.amazonaws.com -u admin -p
Enter password: [REDACTED]

i-Of302ddbb5150f8e2 (public-project)
PublicIPs: 34.226.195.154 PrivateIPs: 20.20.1.123
```

Now we will enter password.

```
No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-20-20-3-148:~$ sudo mysql -h database-1.c9nlafdcoteq.us-east-1.rds.amazonaws.com -u admin -p
Enter password: [REDACTED]

Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 41
Server version: 8.0.33 Source distribution

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> [REDACTED]

i-Of302ddbb5150f8e2 (public-project)
PublicIPs: 34.226.195.154 PrivateIPs: 20.20.1.123
```

Now we successfully connected to mysql.

Now we will show databases. Here we added our database named “poorva”.

Instances | EC2 | us-east-1 | EC2 Instance Conn | RDS | us-east-1 | S3 buckets | S3 | GitHub - sauravu... | Start Course | Intel... | +

us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0f302ddbb5150f8e2&osUser=ubuntu&region=us-east-1

aws Services Search [Alt+S]

EC2 EFS FSx CloudWatch Route 53 VPC S3 RDS DynamoDB Amazon Redshift Lambda Elastic Beanstalk OpsWorks CloudWatch Metrics

affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| poorva |
| sys |
+-----+
5 rows in set (0.01 sec)

mysql> use poorva;
Database changed
mysql>
```

i-0f302ddbb5150f8e2 (public-project)

PublicIPs: 34.226.195.154 PrivateIPs: 20.20.1.123

CloudShell Feedback © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

25°C Mostly clear ENG IN 09:57 PM 14-11-2023

Now we will use this database.

A screenshot of a Windows desktop environment. At the top, there are several browser tabs open, including 'Instances | EC2', 'EC2 Instance Connect', 'RDS | us-east-1', 'S3 buckets | S3', 'aws-code-main/RL', 'Start Course | Intel...', and others. Below the tabs, the taskbar shows icons for various AWS services like EC2, EFS, FSx, CloudWatch, Route 53, VPC, S3, RDS, DynamoDB, Amazon Redshift, Lambda, Elastic Beanstalk, OpsWorks, and CloudWatch Metrics. A MySQL terminal window is open in the foreground, showing the creation of a database 'poorva' and a table 'employee'. The table has five columns: emp\_id, first\_name, last\_name, primary\_skills, and location, all of type VARCHAR(20). The 'describe employee;' command shows the table structure with columns emp\_id, first\_name, last\_name, primary\_skills, and location.

```
mysql> use poorva;
Database changed
mysql> CREATE TABLE employee (emp_id VARCHAR(20), first_name VARCHAR(20),
-> last_name VARCHAR(20),primary_skills VARCHAR(20), location VARCHAR(20));
Query OK, 0 rows affected (0.04 sec)

mysql> describe employee;
+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+
| emp_id | varchar(20) | YES |   | NULL    |       |
| first_name | varchar(20) | YES |   | NULL    |       |
| last_name | varchar(20) | YES |   | NULL    |       |
| primary_skills | varchar(20) | YES |   | NULL    |       |
| location | varchar(20) | YES |   | NULL    |       |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>
```

i-0f302ddbb5150f8e2 (public-project)  
PublicIPs: 34.226.195.154 PrivateIPs: 20.20.1.123

Now here we will describe our database.

A screenshot of a Windows desktop environment, similar to the one above. It shows a MySQL terminal window with the same database and table creation commands as the previous screenshot. After creating the table, a new command is entered: 'insert into employee values(1,"Saurav","singh","AWS","BNG");'. This command inserts a single row into the 'employee' table with values 1, 'Saurav', 'singh', 'AWS', and 'BNG' respectively for the columns emp\_id, first\_name, last\_name, primary\_skills, and location.

```
mysql> use poorva;
Database changed
mysql> CREATE TABLE employee (emp_id VARCHAR(20), first_name VARCHAR(20),
-> last_name VARCHAR(20),primary_skills VARCHAR(20), location VARCHAR(20));
Query OK, 0 rows affected (0.04 sec)

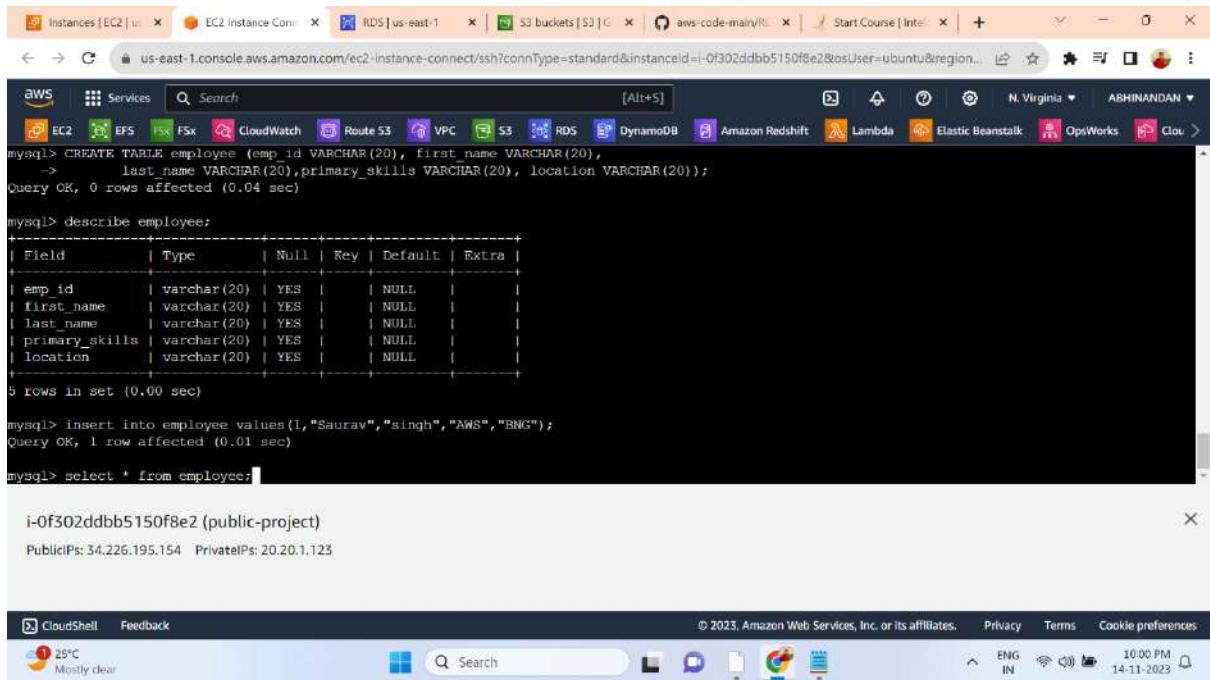
mysql> describe employee;
+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+
| emp_id | varchar(20) | YES |   | NULL    |       |
| first_name | varchar(20) | YES |   | NULL    |       |
| last_name | varchar(20) | YES |   | NULL    |       |
| primary_skills | varchar(20) | YES |   | NULL    |       |
| location | varchar(20) | YES |   | NULL    |       |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> insert into employee values(1,"Saurav","singh","AWS","BNG");

mysql>
```

i-0f302ddbb5150f8e2 (public-project)  
PublicIPs: 34.226.195.154 PrivateIPs: 20.20.1.123

Next here we will insert a data into our table.



AWS CloudShell interface showing a MySQL session. The session starts with creating a table named 'employee' with columns: emp\_id (VARCHAR(20)), first\_name (VARCHAR(20)), last\_name (VARCHAR(20)), primary\_skills (VARCHAR(20)), and location (VARCHAR(20)). It then describes the table, inserts a row with values (1, "Saurav", "singh", "AWS", "BNG"), and selects all rows from the employee table.

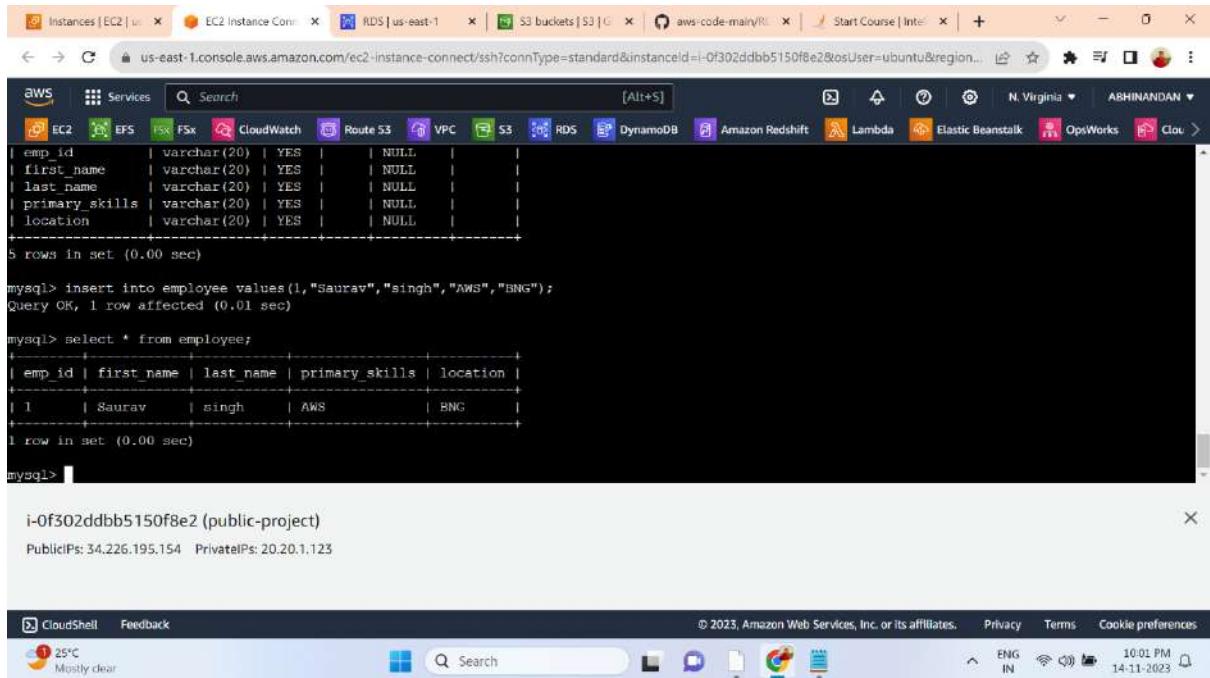
```
mysql> CREATE TABLE employee (emp_id VARCHAR(20), first_name VARCHAR(20),
-> last_name VARCHAR(20),primary_skills VARCHAR(20), location VARCHAR(20));
Query OK, 0 rows affected (0.04 sec)

mysql> describe employee;
+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| emp_id | varchar(20) | YES |   | NULL    |       |
| first_name | varchar(20) | YES |   | NULL    |       |
| last_name | varchar(20) | YES |   | NULL    |       |
| primary_skills | varchar(20) | YES |   | NULL    |       |
| location | varchar(20) | YES |   | NULL    |       |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> insert into employee values(1,"Saurav","singh","AWS","BNG");
Query OK, 1 row affected (0.01 sec)

mysql> select * from employee;
```

Now we inserted data into the table successfully.



AWS CloudShell interface showing a MySQL session. It lists the columns of the 'employee' table, inserts a row with values (1, "Saurav", "singh", "AWS", "BNG"), and then selects all rows from the employee table, showing the inserted data.

```
+-----+-----+-----+-----+-----+
| Field | Type  | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| emp_id | varchar(20) | YES |   | NULL    |       |
| first_name | varchar(20) | YES |   | NULL    |       |
| last_name | varchar(20) | YES |   | NULL    |       |
| primary_skills | varchar(20) | YES |   | NULL    |       |
| location | varchar(20) | YES |   | NULL    |       |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> insert into employee values(1,"Saurav","singh","AWS","BNG");
Query OK, 1 row affected (0.01 sec)

mysql> select * from employee;
+-----+-----+-----+-----+-----+
| emp_id | first_name | last_name | primary_skills | location |
+-----+-----+-----+-----+-----+
| 1      | Saurav     | singh     | AWS            | BNG       |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> 
```

Now we can see the inserted data here.

```
| last_name | varchar(20) | YES | NULL | |
| primary_skills | varchar(20) | YES | NULL | |
| location | varchar(20) | YES | NULL | |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> insert into employee values(1,"Saurav","singh","AWS","BNG");
Query OK, 1 row affected (0.01 sec)

mysql> select * from employee;
+-----+-----+-----+-----+
| emp_id | first_name | last_name | primary_skills | location |
+-----+-----+-----+-----+
| 1 | Saurav | singh | AWS | BNG |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> exit
Bye
ubuntu@ip-20-20-3-148:~$
```

i-0f302ddbb5150f8e2 (public-project)  
PublicIPs: 34.226.195.154 PrivateIPs: 20.20.1.123

Now we will exit from mysql.

```
| primary_skills | varchar(20) | YES | NULL | |
| location | varchar(20) | YES | NULL | |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> insert into employee values(1,"Saurav","singh","AWS","BNG");
Query OK, 1 row affected (0.01 sec)

mysql> select * from employee;
+-----+-----+-----+-----+
| emp_id | first_name | last_name | primary_skills | location |
+-----+-----+-----+-----+
| 1 | Saurav | singh | AWS | BNG |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql> exit
Bye
ubuntu@ip-20-20-3-148:~$ cd aws-code-main/
ubuntu@ip-20-20-3-148:~/aws-code-main$ sudo python3 EmpApp.py
```

i-0f302ddbb5150f8e2 (public-project)  
PublicIPs: 34.226.195.154 PrivateIPs: 20.20.1.123

Now we will run our python application to test it.

Now we will copy the DNS name.

Instances | EC2 Instances | RDS | S3 buckets | Load balancer | aws-codedeploy | Start Count | Add Employee | +

Not secure | project-lb-1613434902.us-east-1.elb.amazonaws.com

# EMPLOYEE DATABASE

GET EMPLOYEE INFORMATION

Employee ID:

First Name:

Last Name:

Primary Skills:

Location:

Image:  Choose File No file chosen

UPDATE DATABASE

And we paste DNS name in web browser and run it.

EMPLOYEE DATABASE

GET EMPLOYEE INFORMATION

Employee ID:  
1000

First Name:  
Abhinandan

Last Name:  
Gundim

Primary Skills:  
AWS

Location:  
Mysuru

Image: Choose File pes clg logo.jpg

UPDATE DATABASE

And we will add the required data in the blank space.

SAVE SUCCESSFUL

Following Employee has been added to the database

Abhinandan Gundim

GO BACK

And we successfully added data and we will get pop up message on browser.

Again we will run EC2 instance and we will check the inserted data is present here or not.

Instances | EC2 Instances | RDS | S3 buckets | Load balancer | aws-codedeploy | Start CloudWatch Metrics | 3.134.115.14 | +

← → ⌂ us-east-1.console.aws.amazon.com/ec2-instance-connect/ssh?connType=standard&instanceId=i-0f302ddbb5150f8e2&osUser=ubuntu&region=us-east-1

aws Services Search [Alt+S]

EC2 EFS FSx CloudWatch Route 53 VPC S3 RDS DynamoDB Amazon Redshift Lambda Elastic Beanstalk OpsWorks CloudWatch Metrics

No containers need to be restarted.

No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.

```
ubuntu@ip-20-20-1-123:~$ sudo mysql -h database-1.c9nlafdcotel.us-east-1.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 48
Server version: 8.0.33 Source distribution

Copyright (c) 2000, 2023, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> 
```

i-0f302ddbb5150f8e2 (public-project)

Public IPs: 34.226.195.154 Private IPs: 20.20.1.123

CloudShell Feedback © 2023, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Upcoming Earnings ENG IN 10:19 PM 14-11-2023

Now we runned mysql successfully.

The screenshot shows a terminal window within the AWS CloudShell interface. The user has run a MySQL command to select all rows from the 'employee' table. The output displays two rows of data: one for Saurav Singh (emp\_id 1) and one for Abhinandan Gundim (emp\_id 1000). The table structure includes columns for emp\_id, first\_name, last\_name, primary\_skills, and location.

```

mysql> use poorva;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from employee;
+----+-----+-----+-----+-----+
| emp_id | first_name | last_name | primary_skills | location |
+----+-----+-----+-----+-----+
| 1 | Saurav | Singh | AWS | BNG |
| 1000 | Abhinandan | Gundim | AWS | Mysuru |
+----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql>

```

i-0f302ddbb5150f8e2 (public-project)  
PublicIPs: 34.226.195.154 PrivateIPs: 20.20.1.123

Here the inserted data is successfully displayed here.

The screenshot shows the AWS S3 console for the 'abhiig-project2' bucket. The 'Objects' tab is selected, displaying three files: 'aboutus.html', 'emp-id-1000\_image\_file', and 'error.html'. The 'aboutus.html' file was uploaded on November 14, 2023, at 19:59:05 UTC+05:30, and has a size of 1.5 KB. The 'emp-id-1000\_image\_file' file was uploaded on November 14, 2023, at 22:14:28 UTC+05:30, and has a size of 18.8 KB. The 'error.html' file was uploaded on November 14, 2023, at 19:59:01 UTC+05:30, and has a size of 798.0 B.

Name	Type	Last modified	Size	Storage class
aboutus.html	html	November 14, 2023, 19:59:05 (UTC+05:30)	1.5 KB	Standard
emp-id-1000_image_file	-	November 14, 2023, 22:14:28 (UTC+05:30)	18.8 KB	Standard
error.html	html	November 14, 2023, 19:59:01 (UTC+05:30)	798.0 B	Standard

And in S3 also its displayed some inserted data.

The screenshot shows the AWS DynamoDB console with the table `employee_image_table`. A success message states "Completed. Read capacity units consumed: 0.5". The table contains one item with the following details:

empid (Number)	image_url
1000	<a href="https://s3.amazonaws.com/abhibg-project2/emp-id-1000_image_file">https://s3.amazonaws.com/abhibg-project2/emp-id-1000_image_file</a>

The data inserted in web browser we can see it in Dynamo DB also.

The screenshot shows the AWS DynamoDB console on the "Edit item" page for the `employee_image_table`. The item is being edited with the following attributes:

Attribute name	Value	Type
empid - Partition key	1000	Number
image_url	<a href="https://s3.amazonaws.com/abhibg-project2/emp-id-1000_image_file">https://s3.amazonaws.com/abhibg-project2/emp-id-1000_image_file</a>	String

Buttons at the bottom include **Cancel**, **Save**, and **Save and close**.