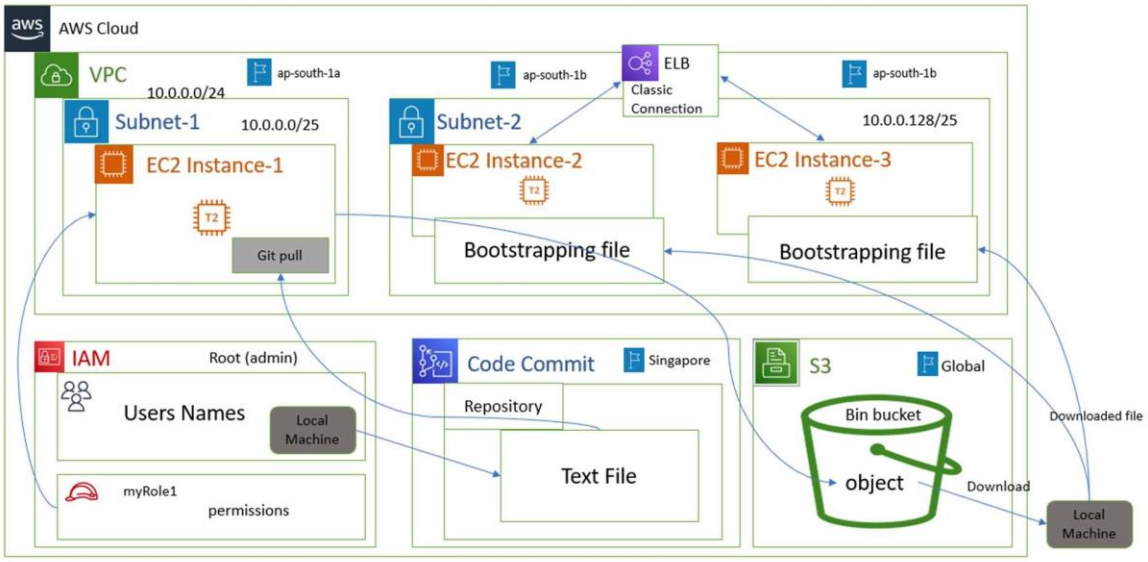
An Architecture of Elastic Load Balancer (ELB) using Bootstrapping Method In custom VPC - 7 tier.

AWS Services Used

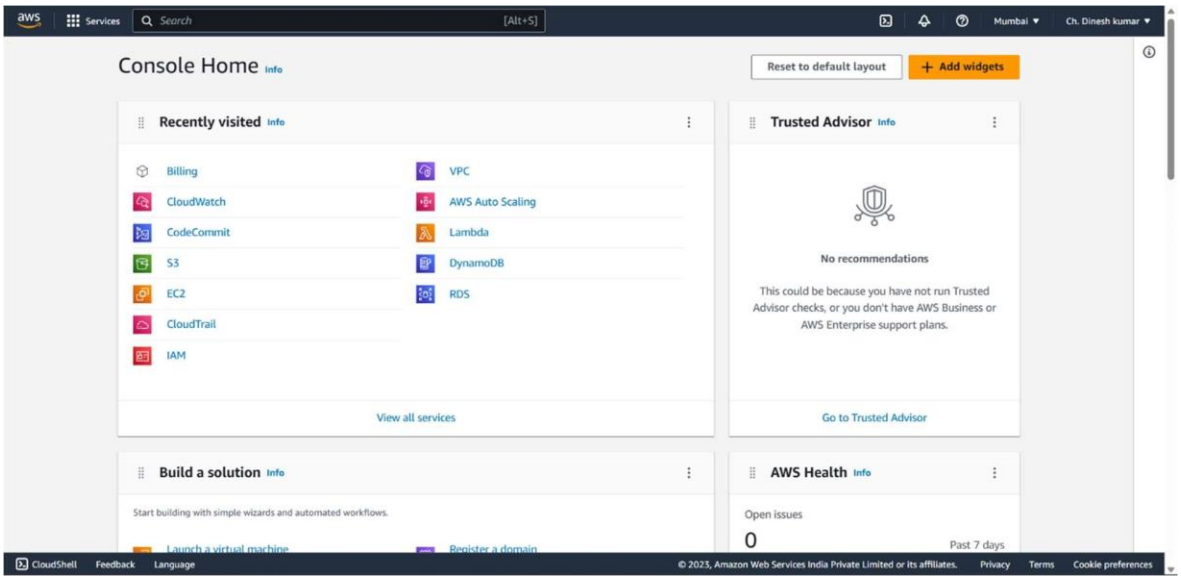
- Instance EC2
- Custom VPC
- Code Commit
- IAM User & Groups
- IAM Role
- S3 Buckets
- EC2 Load Balancer

Architecture



Implementation

Initially open the AWS console in the AWS web site



Create a User in IAM

To create a user, follow these steps:

- Open the AWS Management Console in your web browser.
- Navigate to the IAM service.
- In the left navigation pane, click on "Users" and then click on the "Add user" button.

User Details

- Provide a name for the user in the "Username" field, such as "vidyadhar". • Optionally, you can enable the checkbox for "Programmatic access" to allow the user to interact with AWS services programmatically through APIs.

- If necessary, enable the checkbox for "AWS Management Console access" to grant the user access to the AWS Management Console.

Password

- If you selected "AWS Management Console access" in the previous step, choose one of the following options to set the user's password:
- "Autogenerated password" will create a randomly generated password for the user.
- "Custom password" will allow you to set a specific password for the user. • "Require password reset" will prompt the user to change their password upon first login.

Permissions

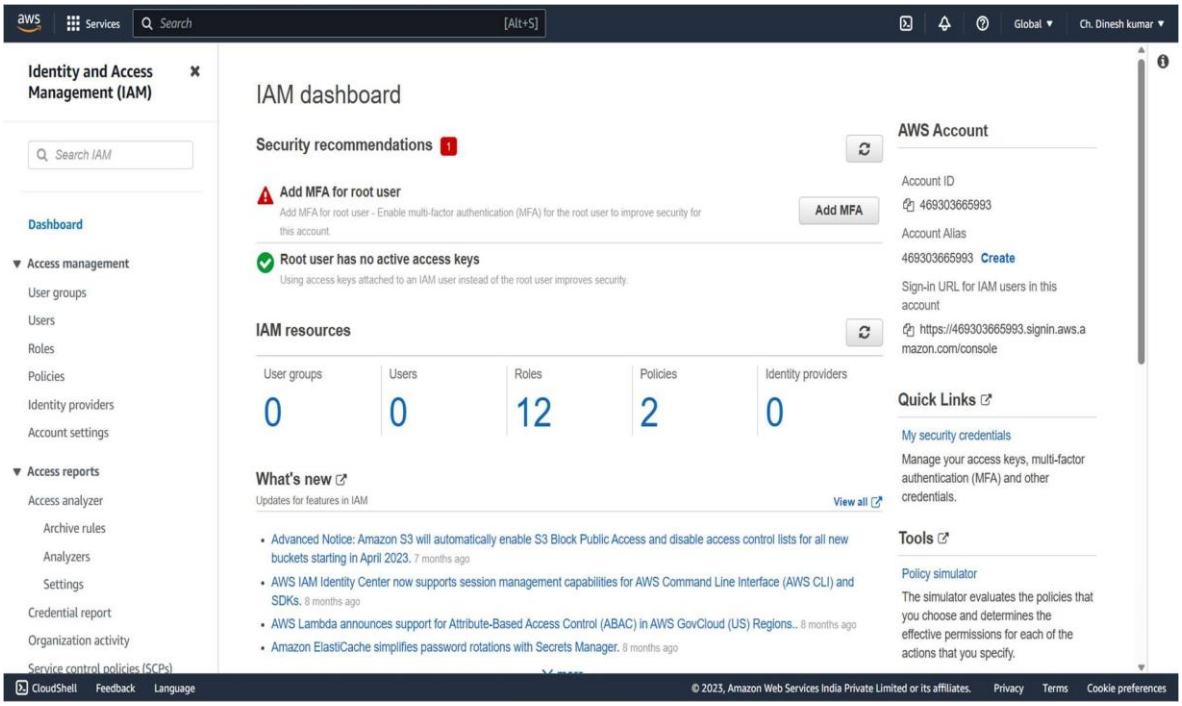
- In the "Set permissions" section, you can choose to attach existing policies to the user or create a custom policy. This will define the user's permissions and access to AWS resources.
- To attach existing policies, click on the "Add permissions" button, select the desired policies from the list, and click on the "Next: Tags" button.
- To create a custom policy, click on the "Create group" button and follow the steps to define and attach the policy to the user

Tags

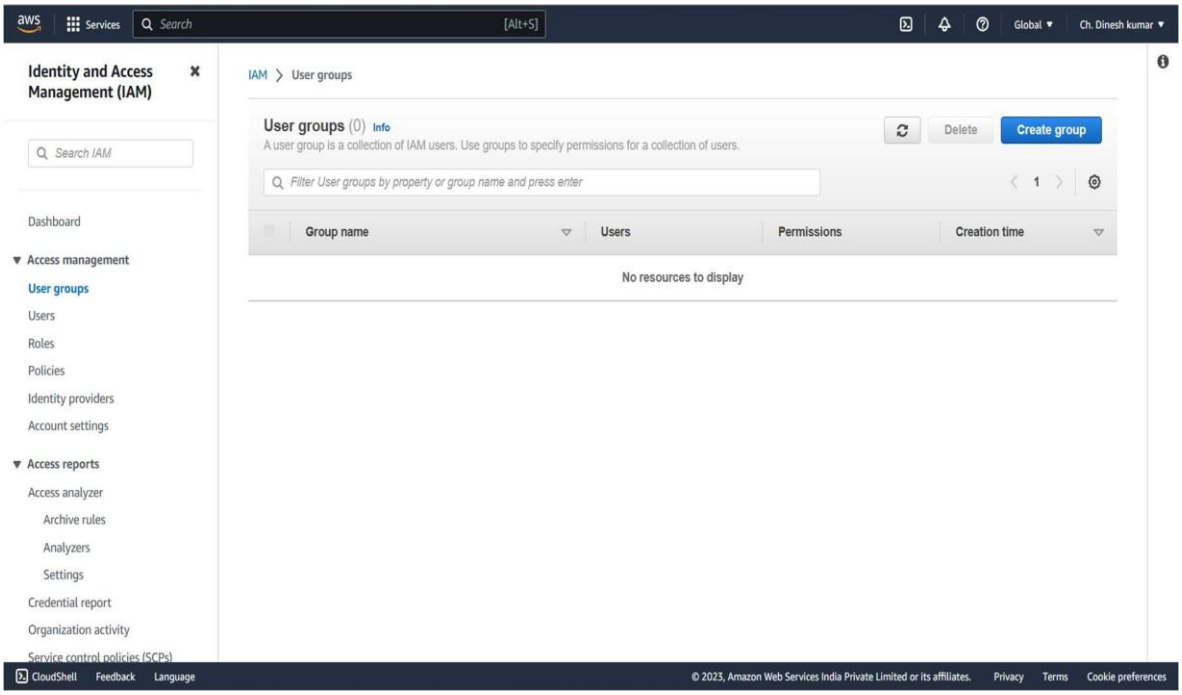
• Optionally, you can add tags to the user for better organization and management. Tags are key-value pairs that provide metadata about the user.

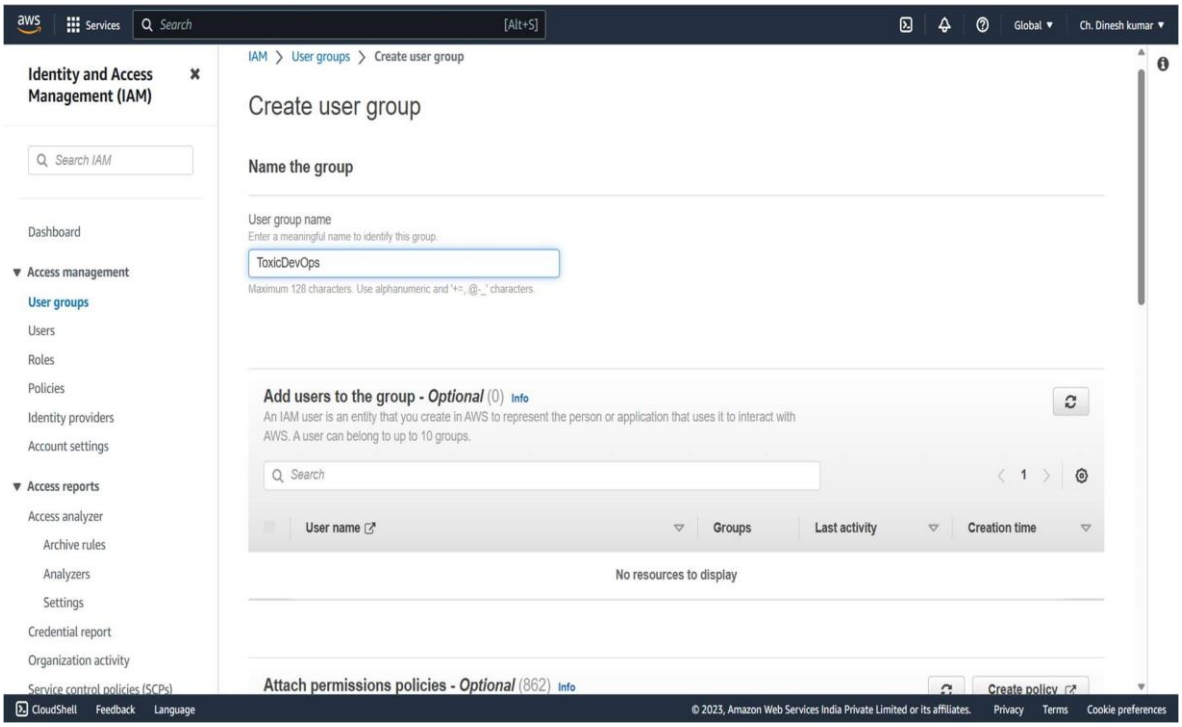
Review

• Review the user details, permissions, and tags to ensure they are correct.
• Click on the "Create user" button to create the user.
• The user "vidyadhar" will be created in IAM with the specified permissions and access settings. Make sure to securely manage and share the user's access credentials, such as access key and secret key, if programmatic access is enabled.
• Copy console sign in link another Browser.
• Enter proper credentials and login.

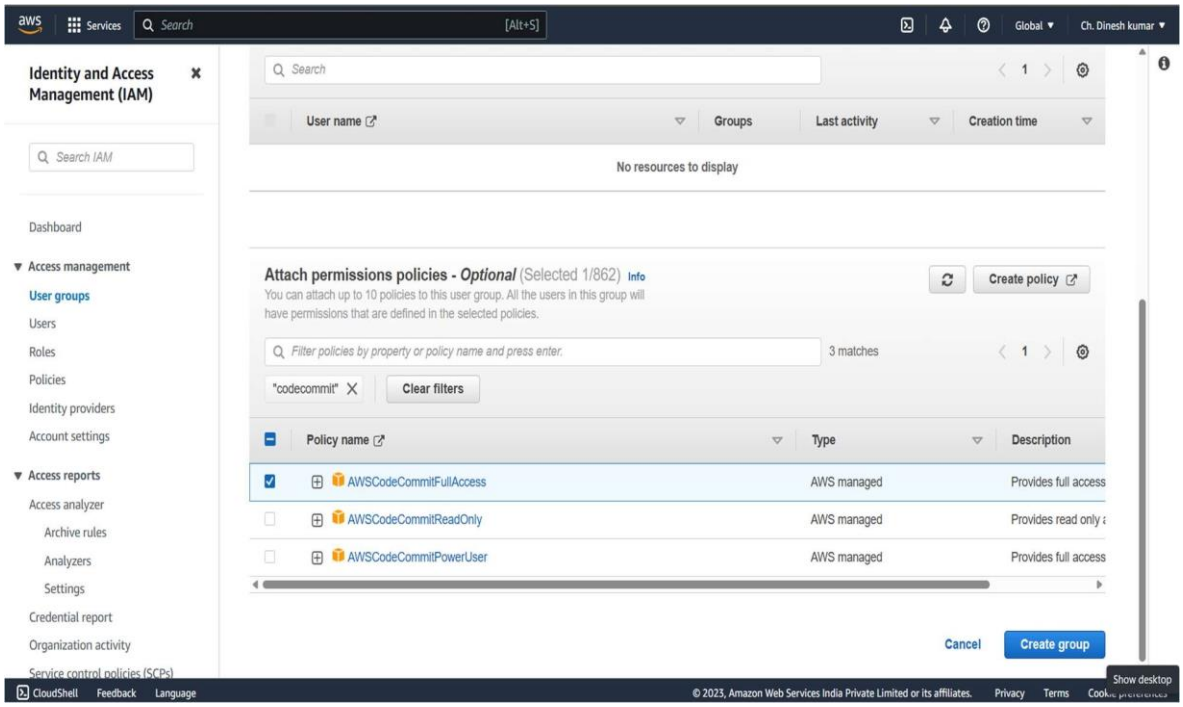


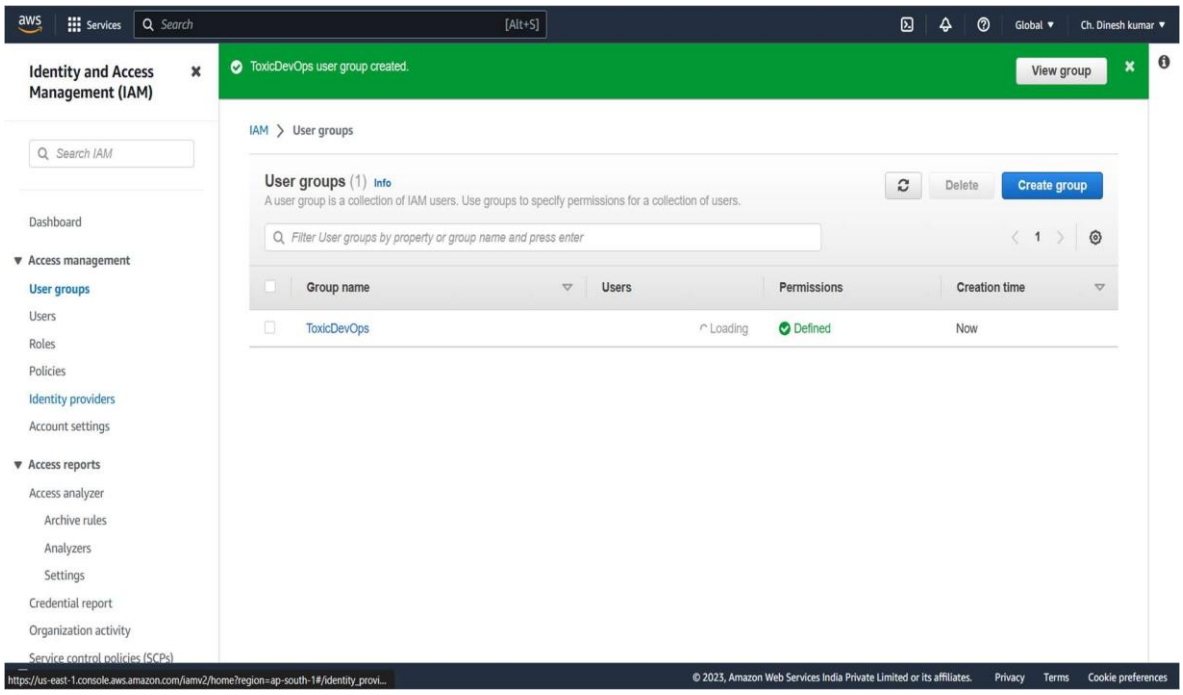
- go to user group and create a user group

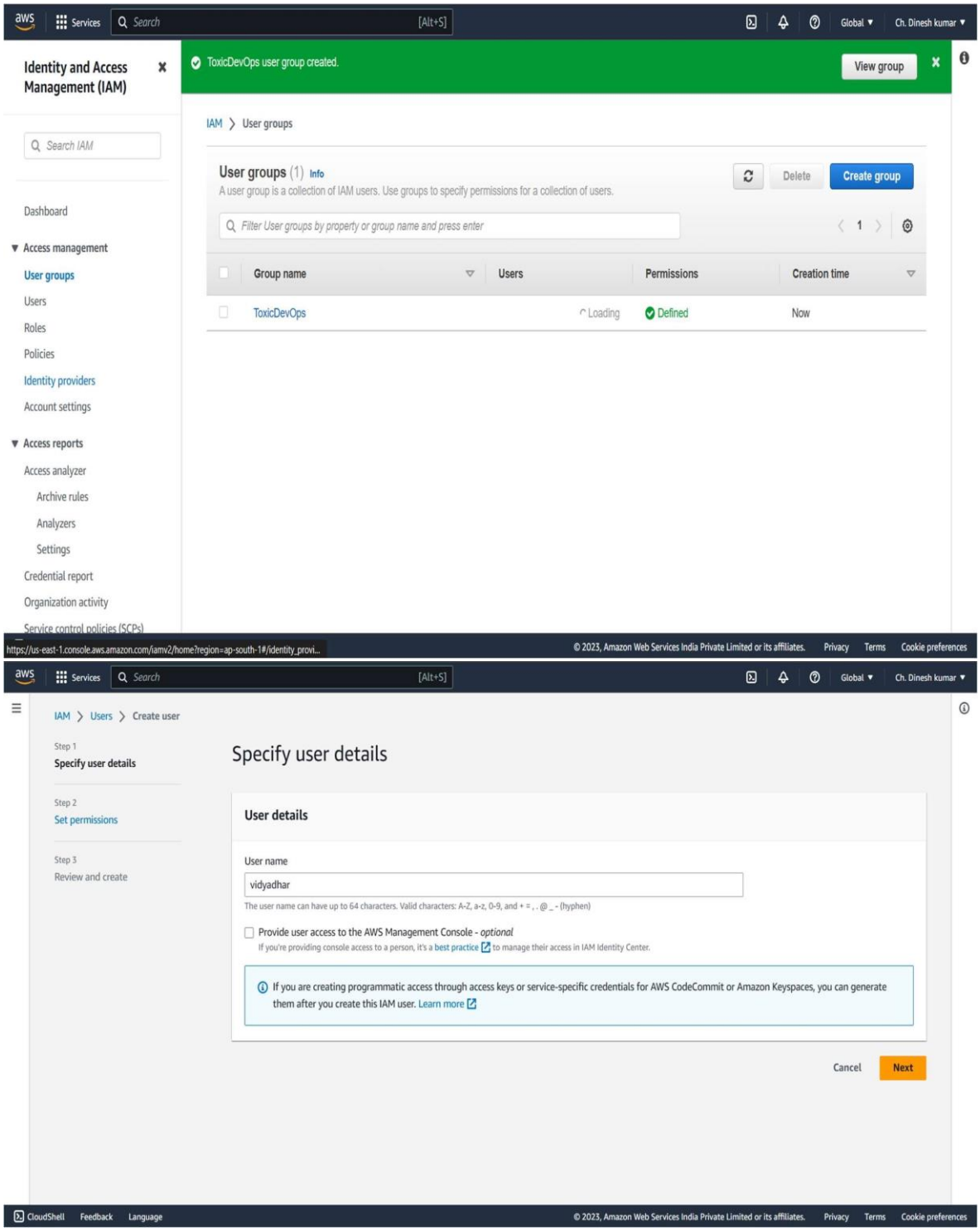




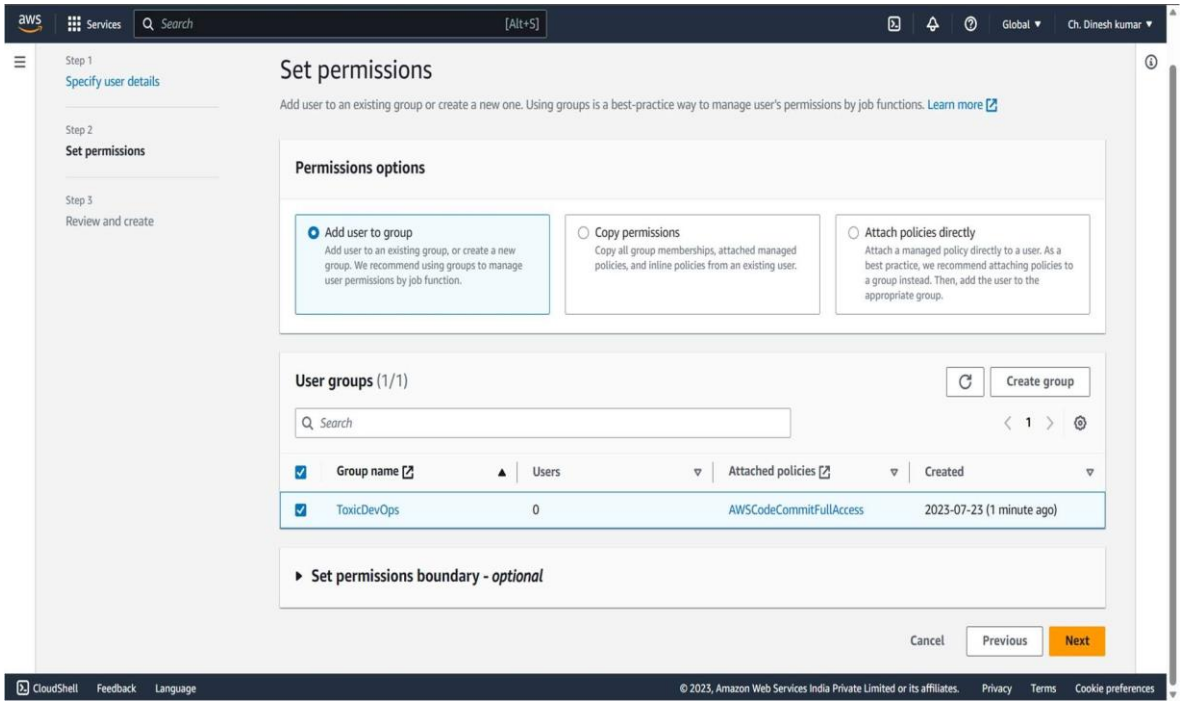
•Add AWSCodeCommitFullAccess to the user group

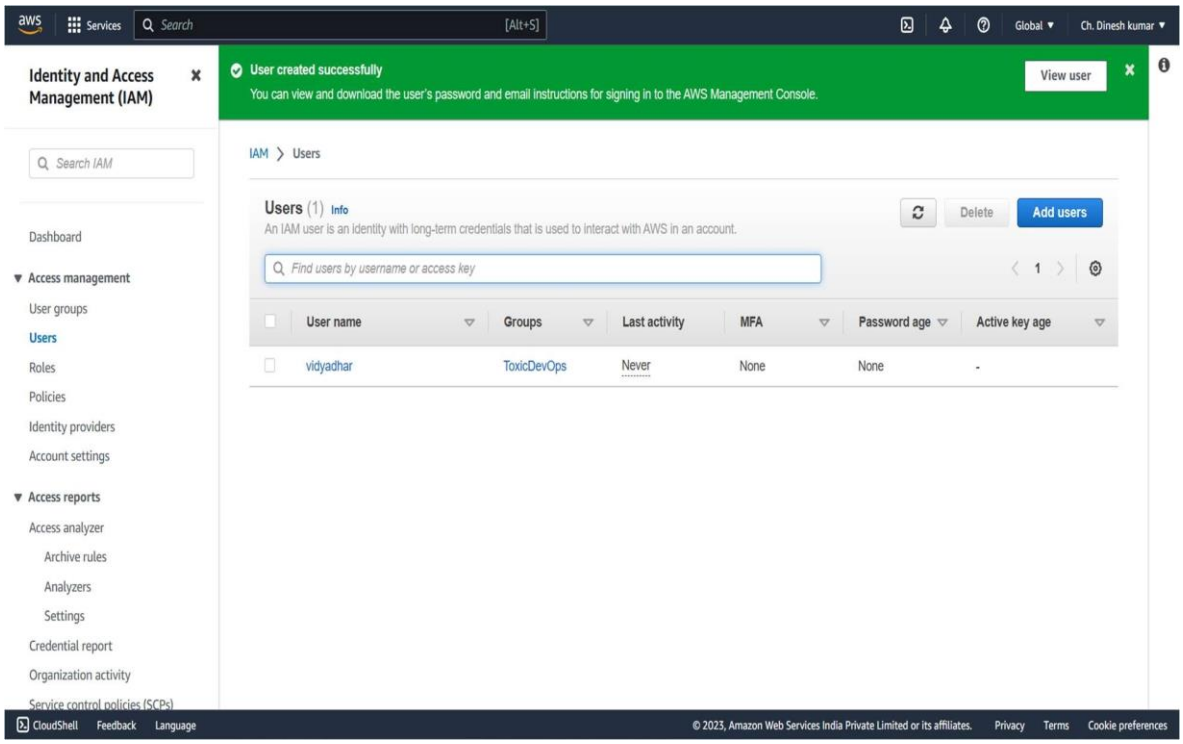

aws
Services
Search
[Alt+S]
Global
Ch. Dinesh kumar
Identity and Access Management (IAM)
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Access reports
Access analyzer
Archive rules
Analyzers
Settings
Credential report
Organization activity
Service control policies (SCPs)
User name
Groups
Last activity
Creation time
No resources to display
Attach permissions policies - Optional (Selected 1/862) Info
You can attach up to 10 policies to this user group. All the users in this group will have permissions that are defined in the selected policies.
Create policy
Filter policies by property or policy name and press enter.
3 matches
"codecommit"
Clear filters
Policy name
Type
Description
AWSCodeCommitFullAccess
AWS managed
Provides full access
AWSCodeCommitReadOnly
AWS managed
Provides read only
AWSCodeCommitPowerUser
AWS managed
Provides full access
Cancel
Create group
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Show desktop

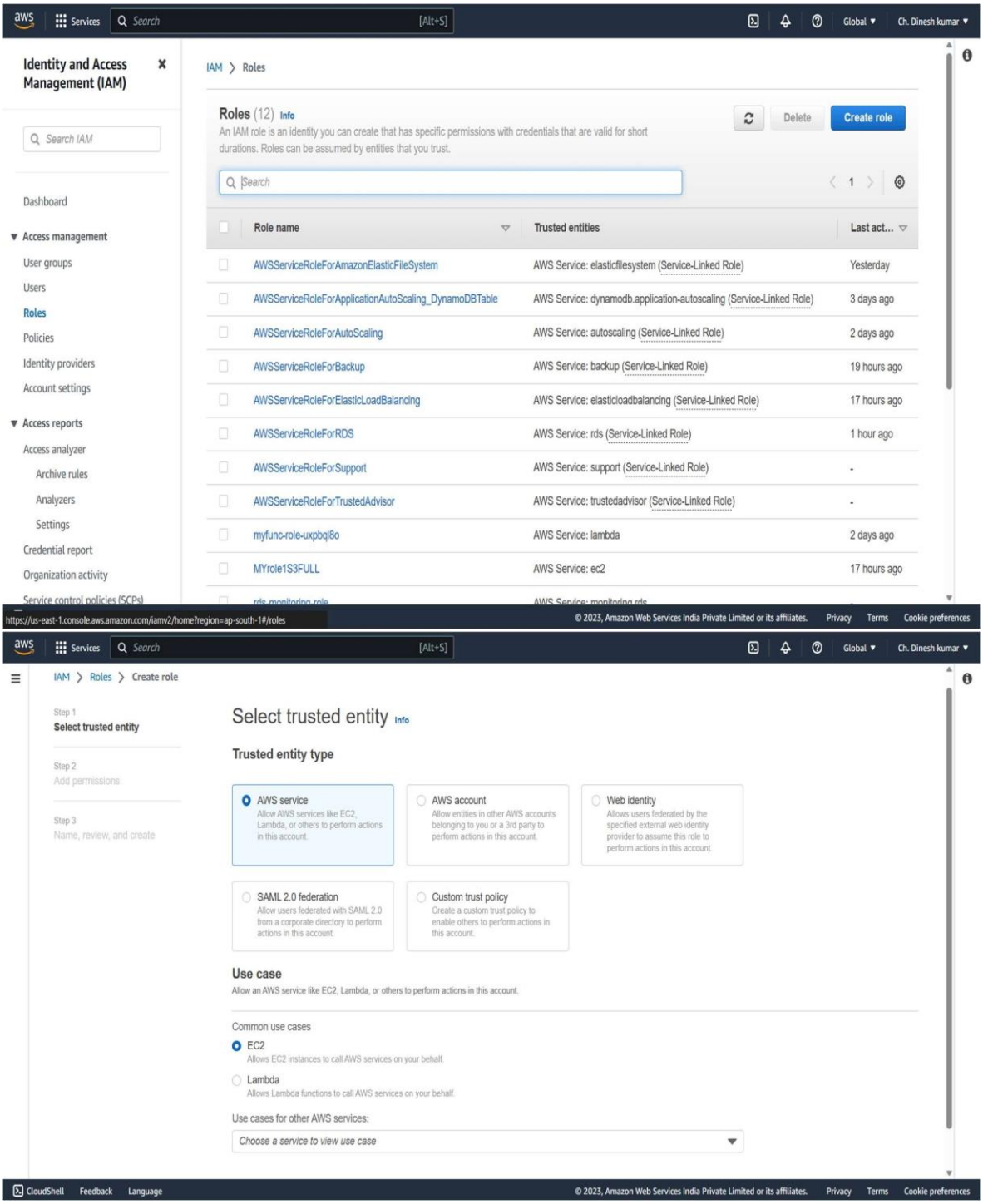

aws
Services
Search
[Alt+S]
Global
Ch. Dinesh kumar
Identity and Access Management (IAM)
ToxicDevOps user group created.
View group
IAM > User groups
User groups (1) Info
A user group is a collection of IAM users. Use groups to specify permissions for a collection of users.
Delete
Create group
Filter User groups by property or group name and press enter
Group name
Users
Permissions
Creation time
ToxicDevOps
Loading
Defined
Now
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Access reports
Access analyzer
Archive rules
Analyzers
Settings
Credential report
Organization activity
Service control policies (SCPs)
https://us-east-1.console.aws.amazon.com/iamv2/home?region=ap-south-1#/identity_provi...
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences



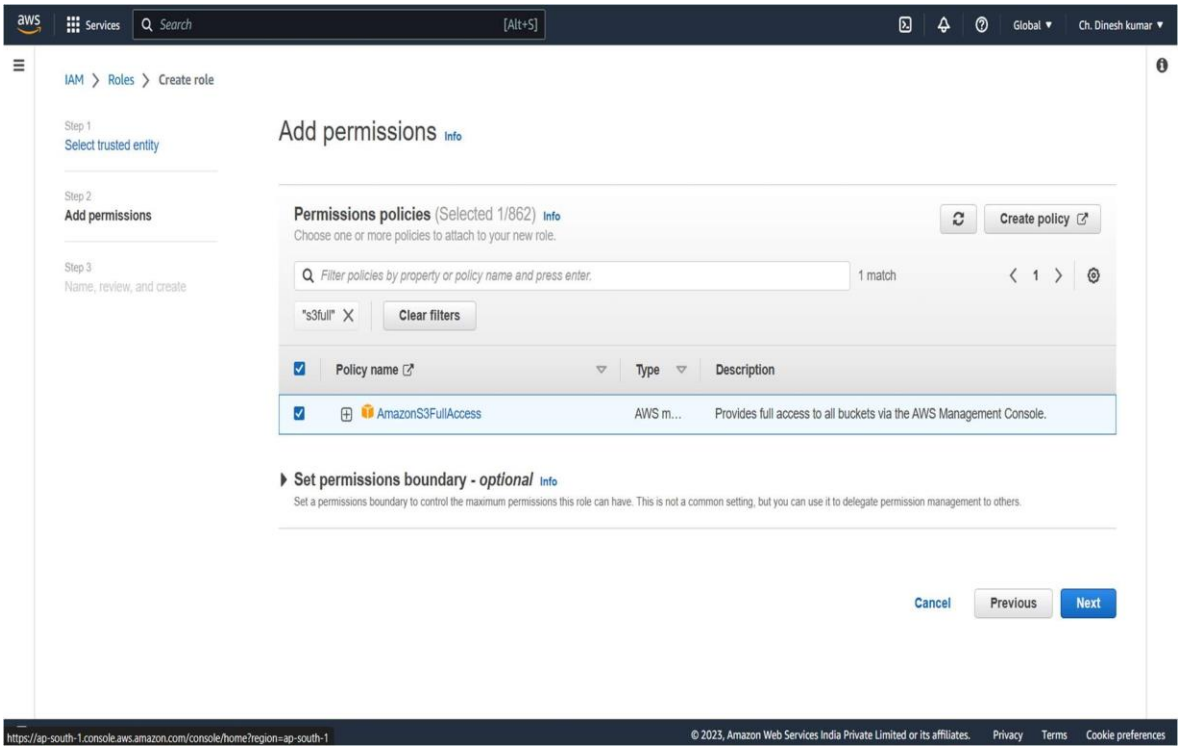
- create and add user in user group

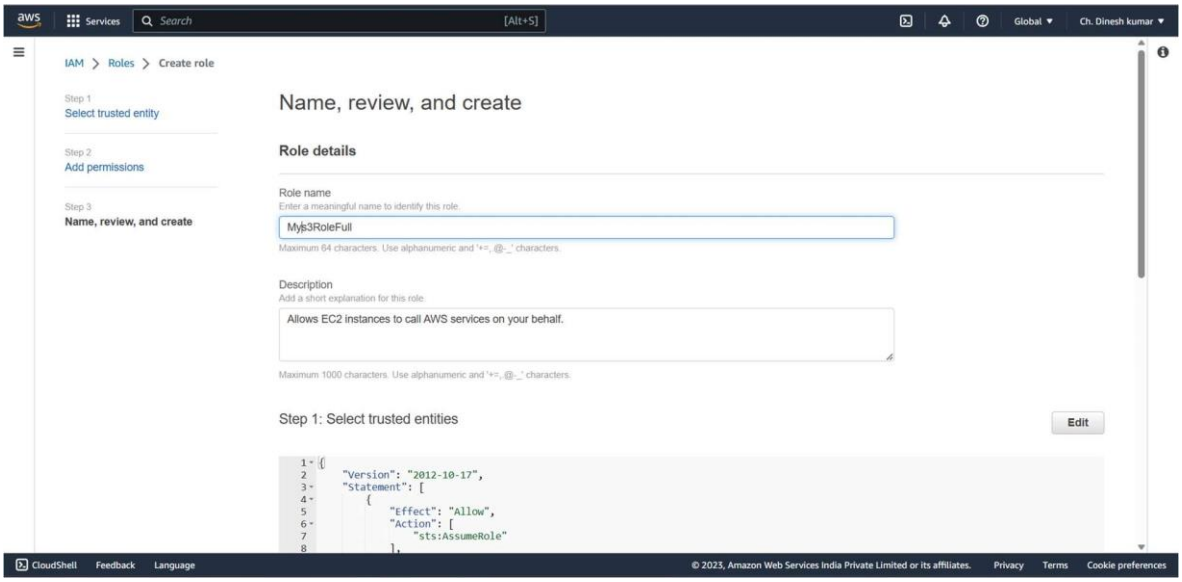

Services
Search
[Alt+S]
Global
Ch. Dinesh kumar
Step 1
Specify user details
Step 2
Set permissions
Step 3
Review and create
Set permissions
Add user to an existing group or create a new one. Using groups is a best-practice way to manage user's permissions by job functions. Learn more
Permissions options
Add user to group
Add user to an existing group, or create a new group. We recommend using groups to manage user permissions by job function.
Copy permissions
Copy all group memberships, attached managed policies, and inline policies from an existing user.
Attach policies directly
Attach a managed policy directly to a user. As a best practice, we recommend attaching policies to a group instead. Then, add the user to the appropriate group.
User groups (1/1)
Create group
Search
Group name
Users
Attached policies
Created
ToxicDevOps
0
AWSCodeCommitFullAccess
2023-07-23 (1 minute ago)
Set permissions boundary - optional
Cancel
Previous
Next
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences


Services
Search
[Alt+S]
Global
Ch. Dinesh kumar
Identity and Access Management (IAM)
User created successfully
You can view and download the user's password and email instructions for signing in to the AWS Management Console.
View user
Search IAM
Dashboard
Access management
User groups
Users
Roles
Policies
Identity providers
Account settings
Access reports
Access analyzer
Archive rules
Analyzers
Settings
Credential report
Organization activity
Service control policies (SCPs)
IAM > Users
Users (1) Info
An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.
Delete
Add users
Find users by username or access key
User name
Groups
Last activity
MFA
Password age
Active key age
vidyadhar
ToxicDevOps
Never
None
None
-
CloudShell
Feedback
Language
© 2023, Amazon Web Services India Private Limited or its affiliates.
Privacy
Terms
Cookie preferences

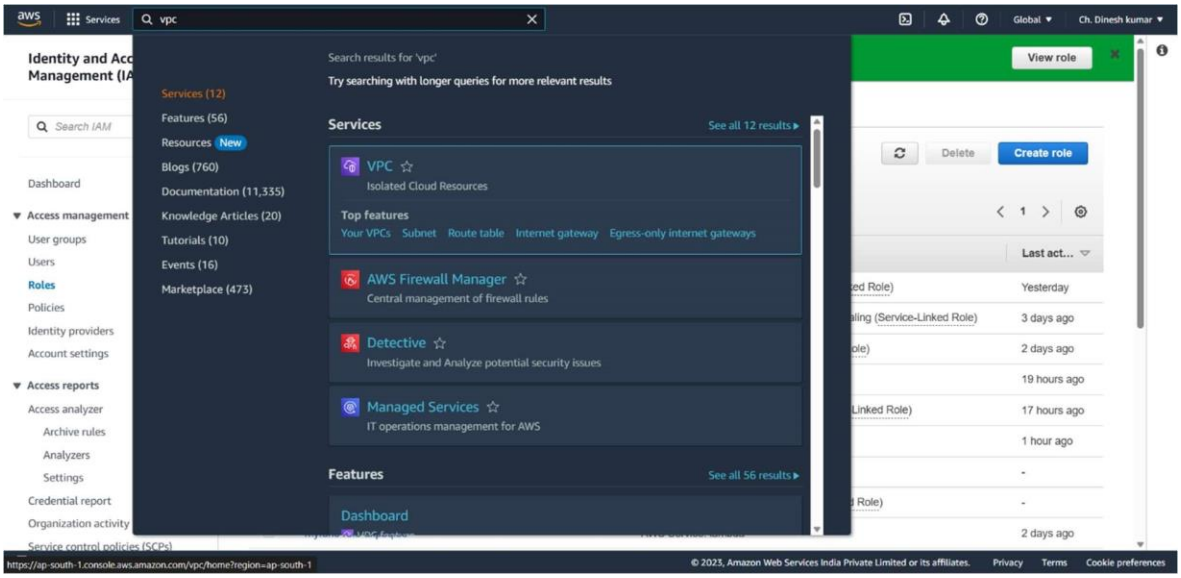


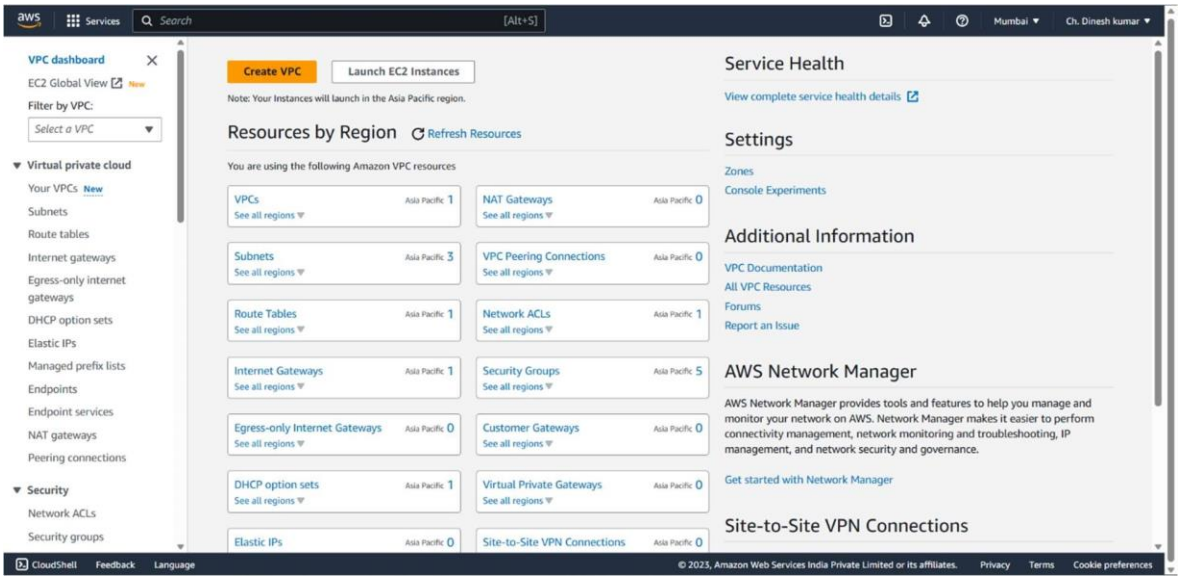
- Add AmazonS3FullAccess and AWSCodeCommitFullAccess to IAM Role





- Now again come to the AWS console and Navigate to the VPC.
- Select the VPC.

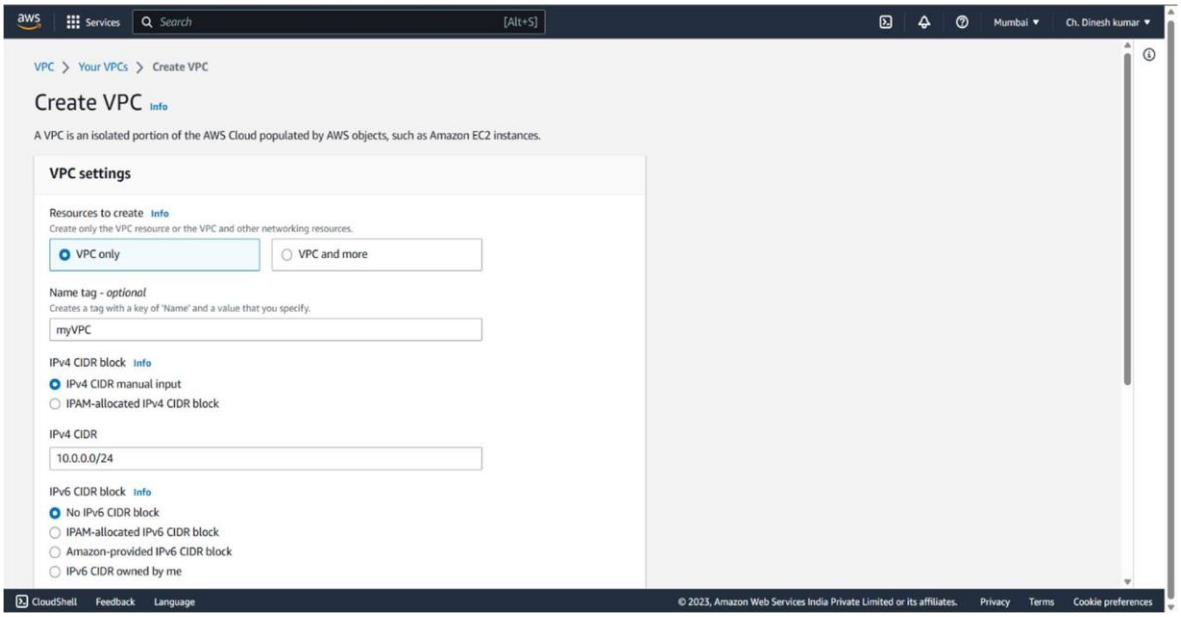




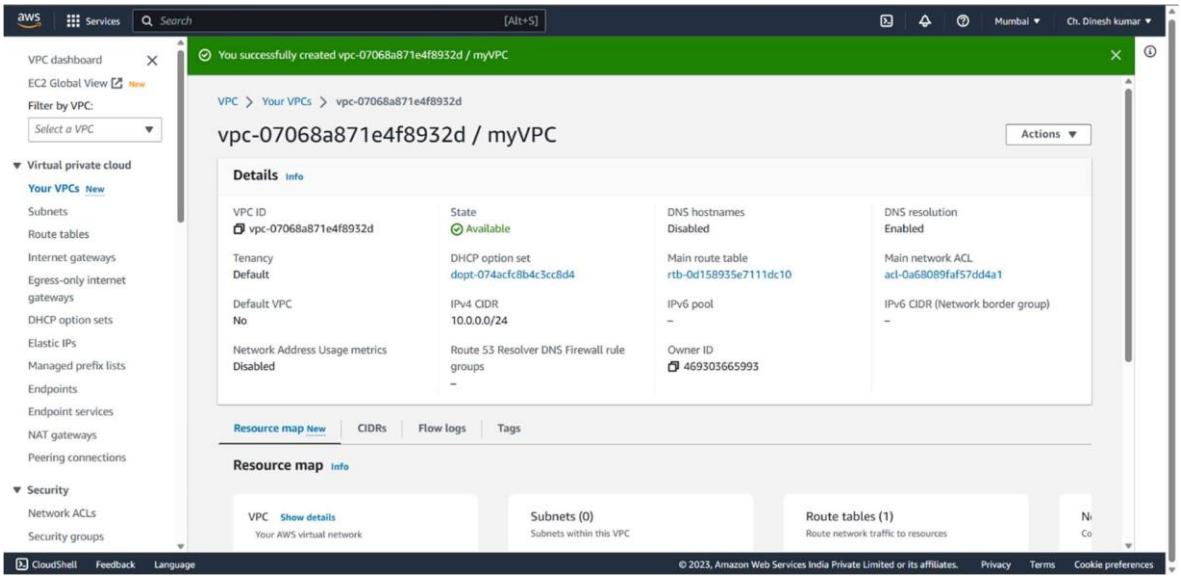
Create VPC

- VPC only
- Give the name to the VPC.
- IPv4 CIDR
- Select the Subnets

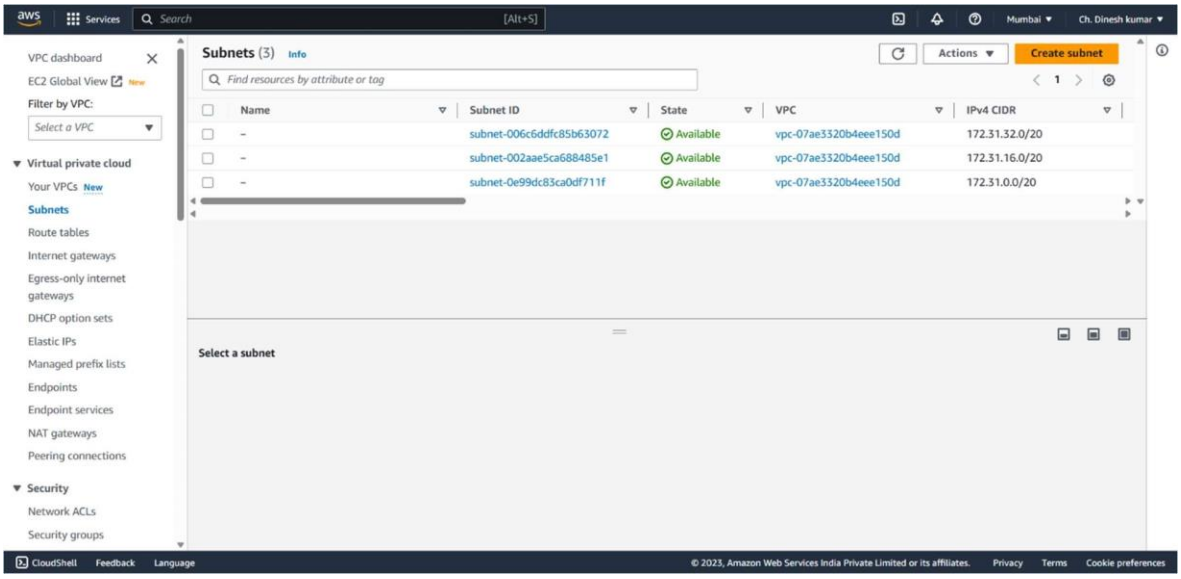
- Finally Create the VPC



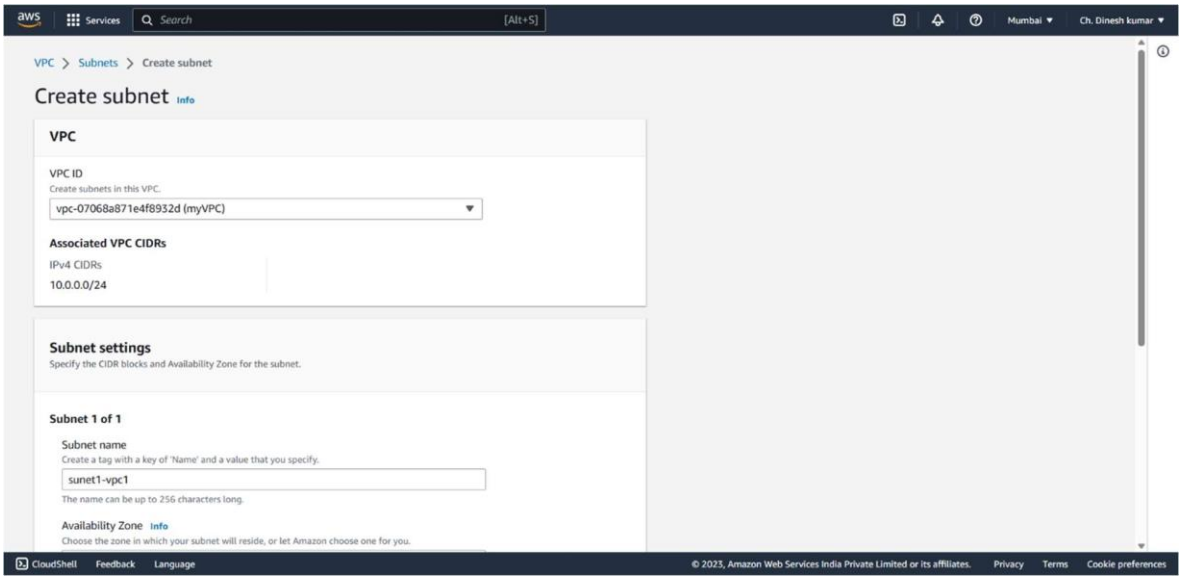
- If the VPC is create successfully it will pop up the Notification that YOU SUCCESSFULLY CREATED VPC.

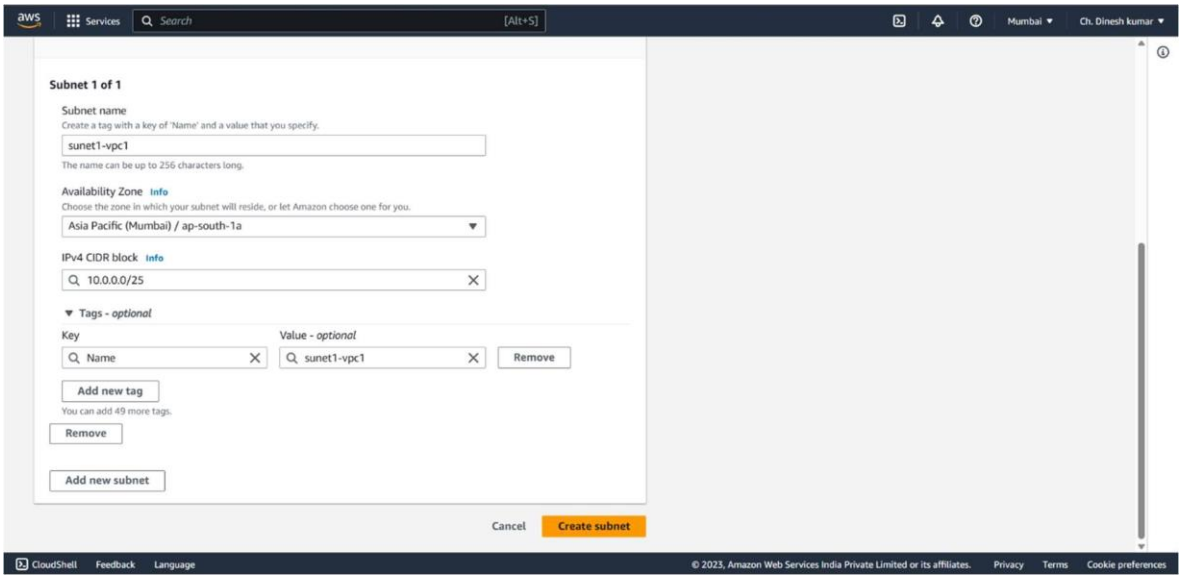


- After VPC creation now create the Subnets for your custom VPC.

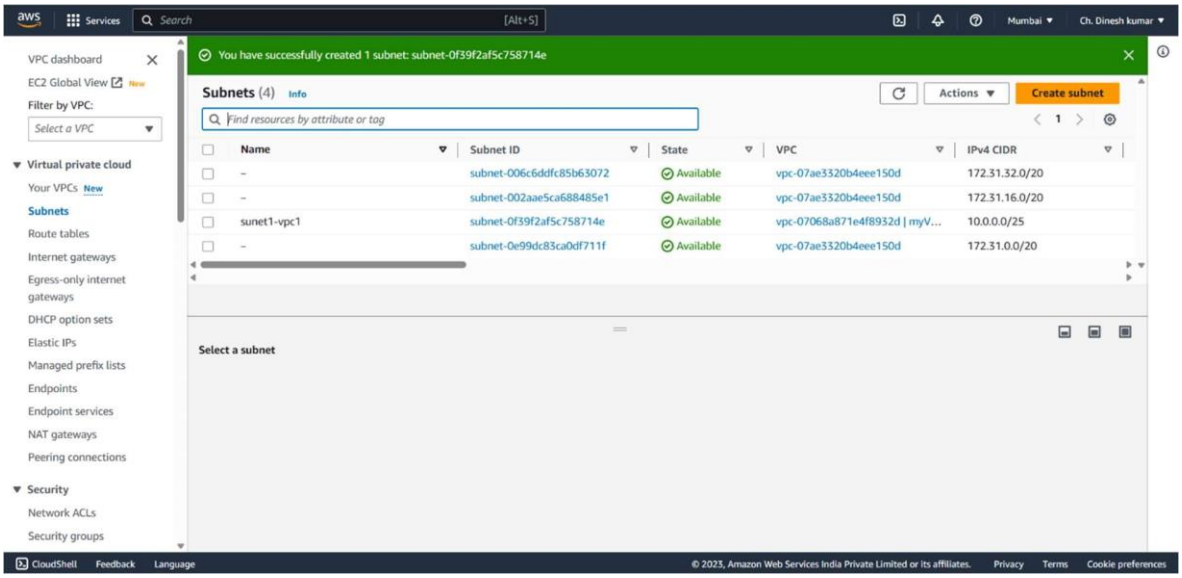


- For Creating subnets Their will some option choose that options and create the subnets.

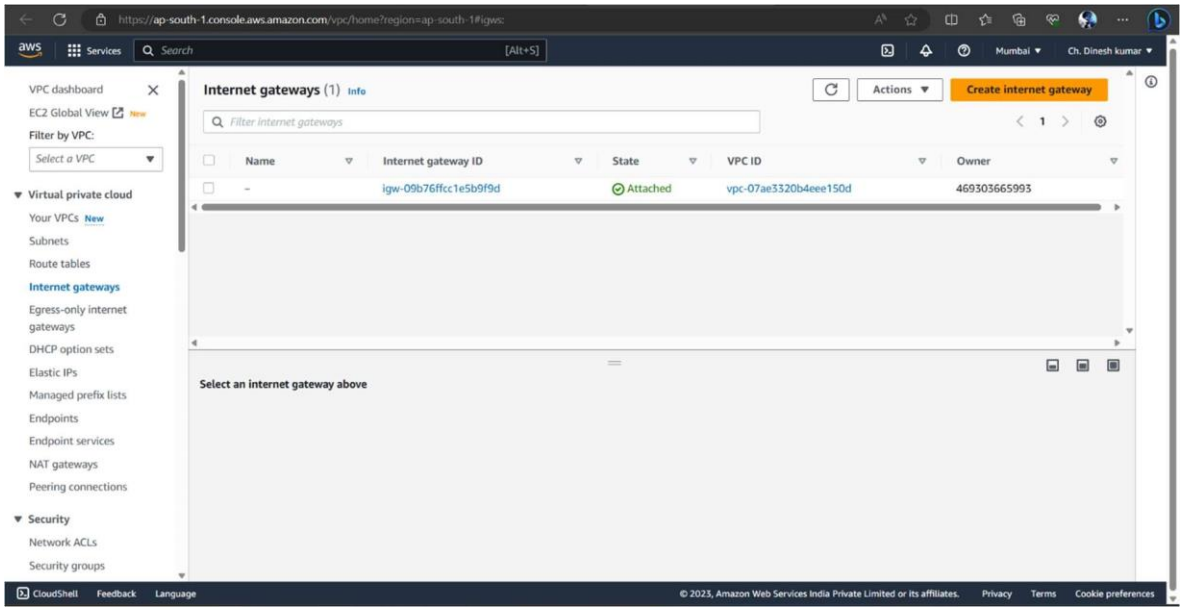




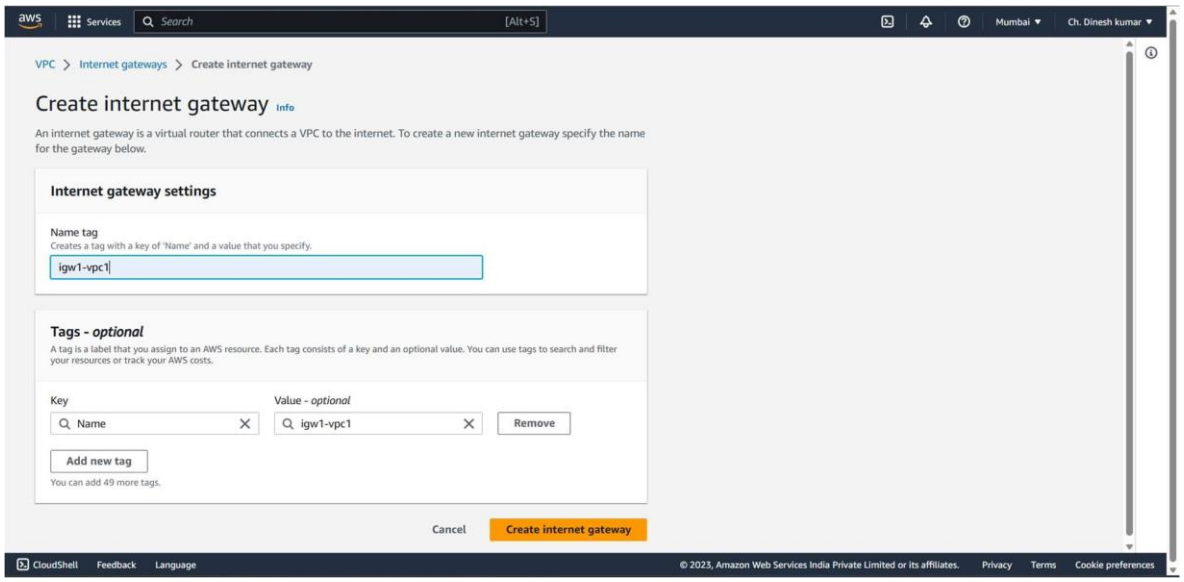
- Then Assign name to the created subnet.
- There three default subnets in VPC.

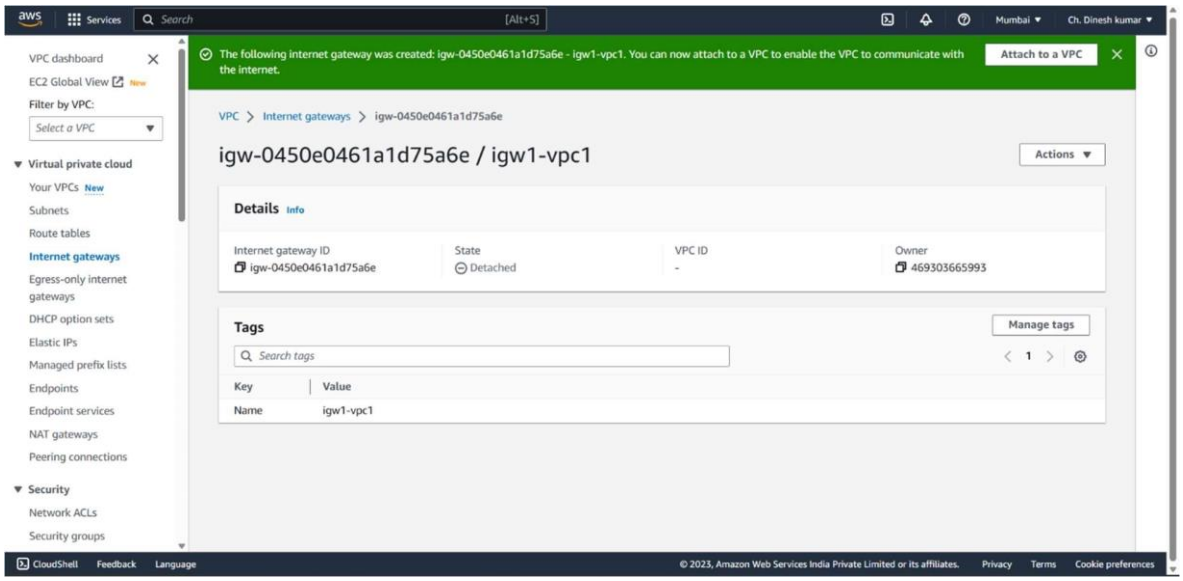


- Create the Internal Gateway.

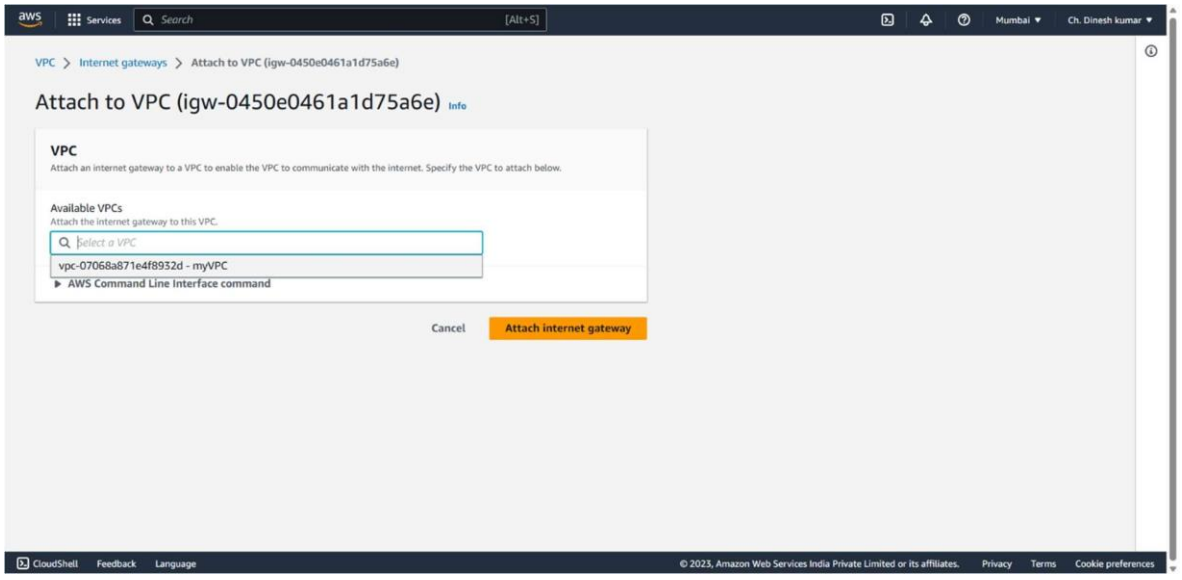


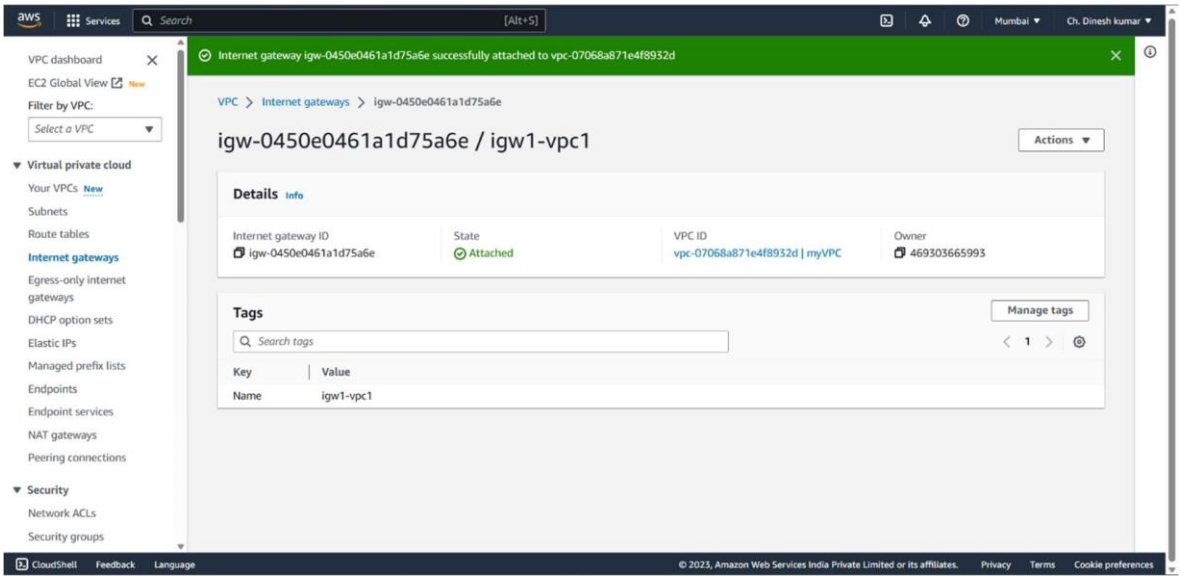
- Give Name tag to the Internal gateway.
- Select Create Internal Gateway.



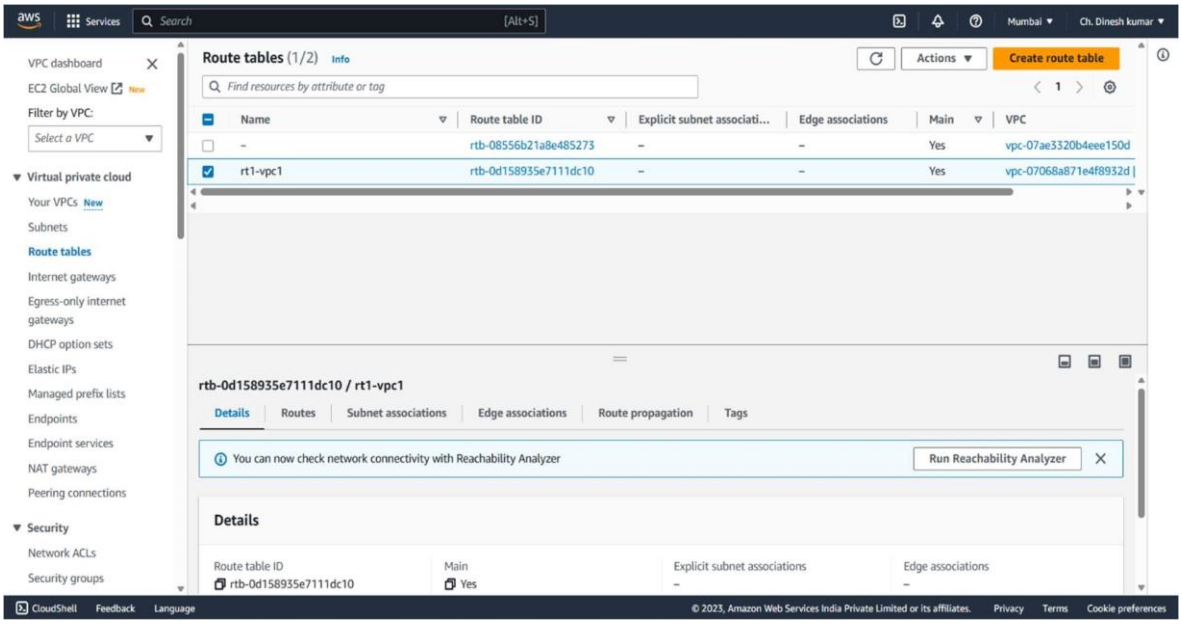


- Attach the created VPC to the Internal Gateway.
- Select ATTACH INTERNAL GATEWAY.

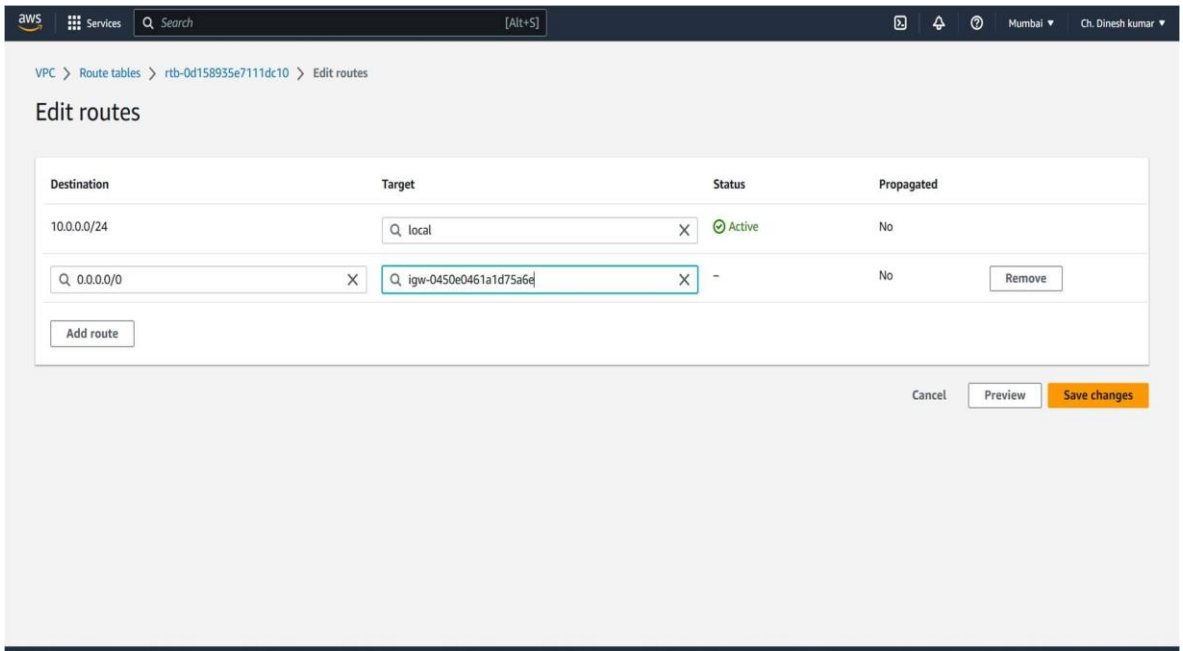


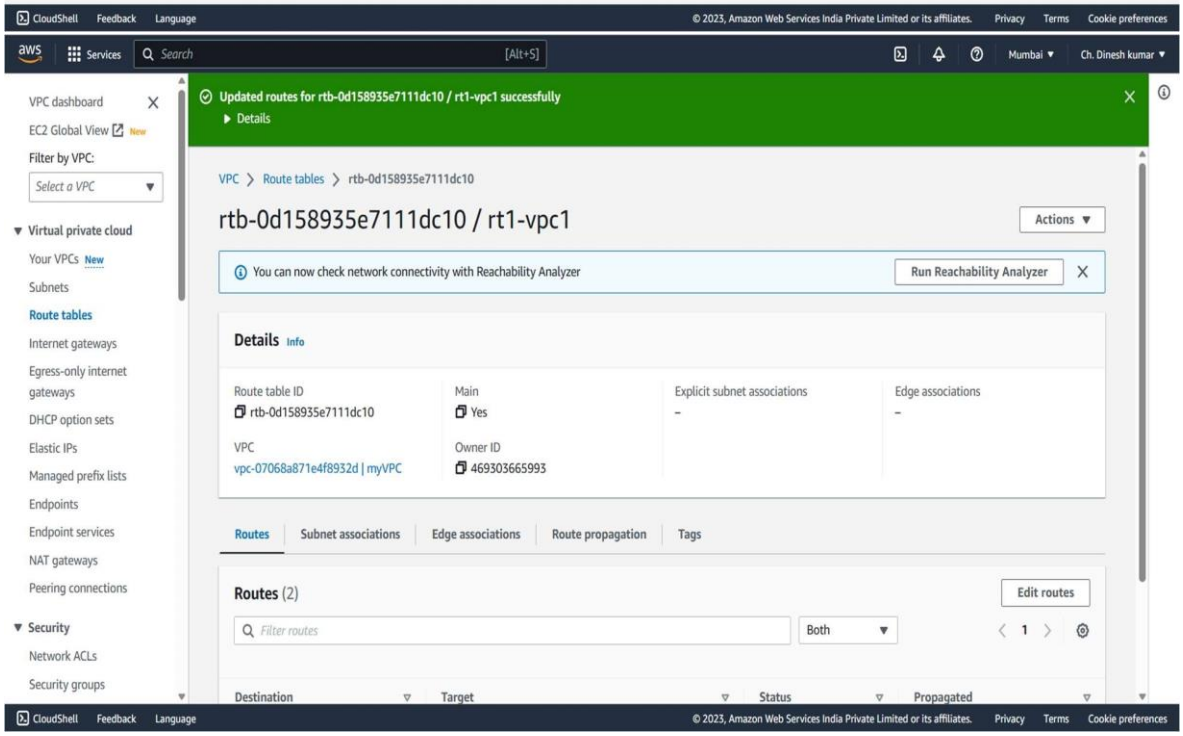


- The above picture shows the Attachment of VPC to Internal Gateway.
- Now Go to Route tables.
- Routes are automatically created when the VPC is created.



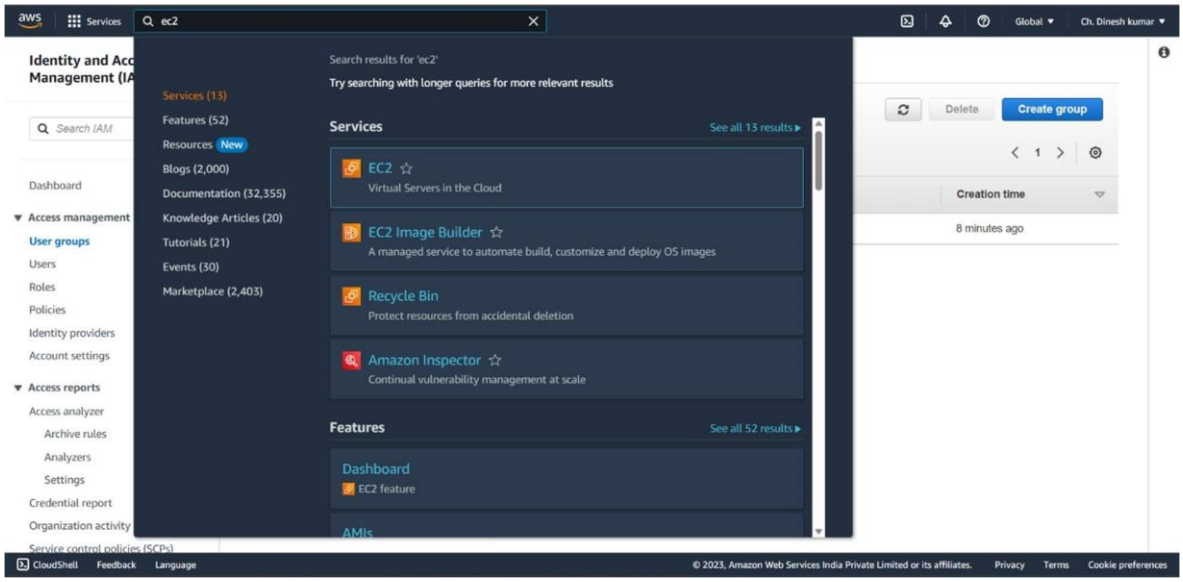
- Add Routes to Route Table.



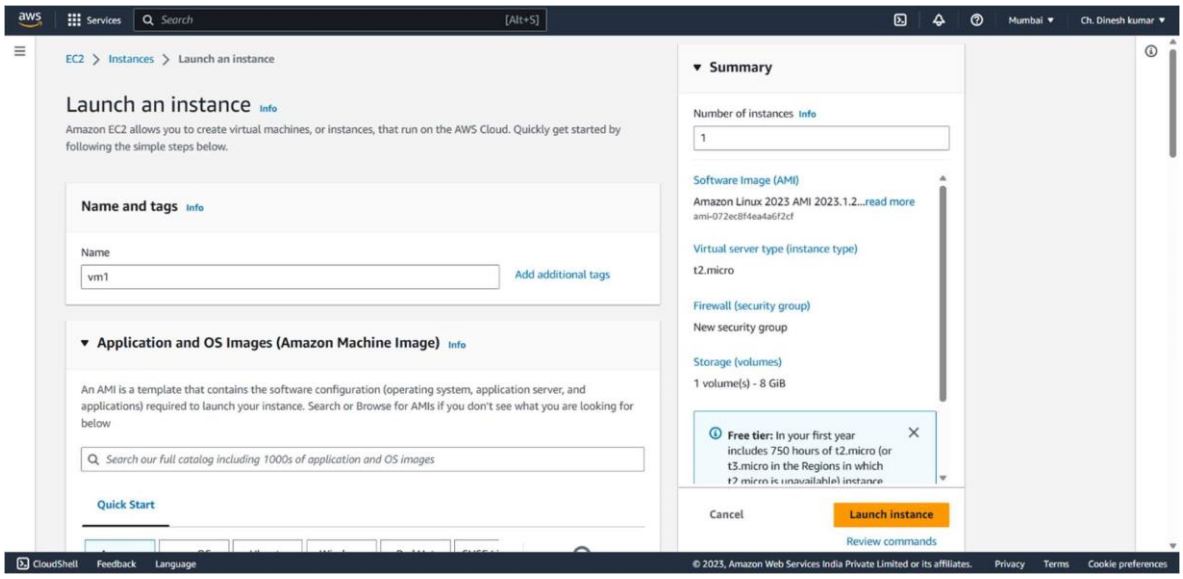


EC2 creation

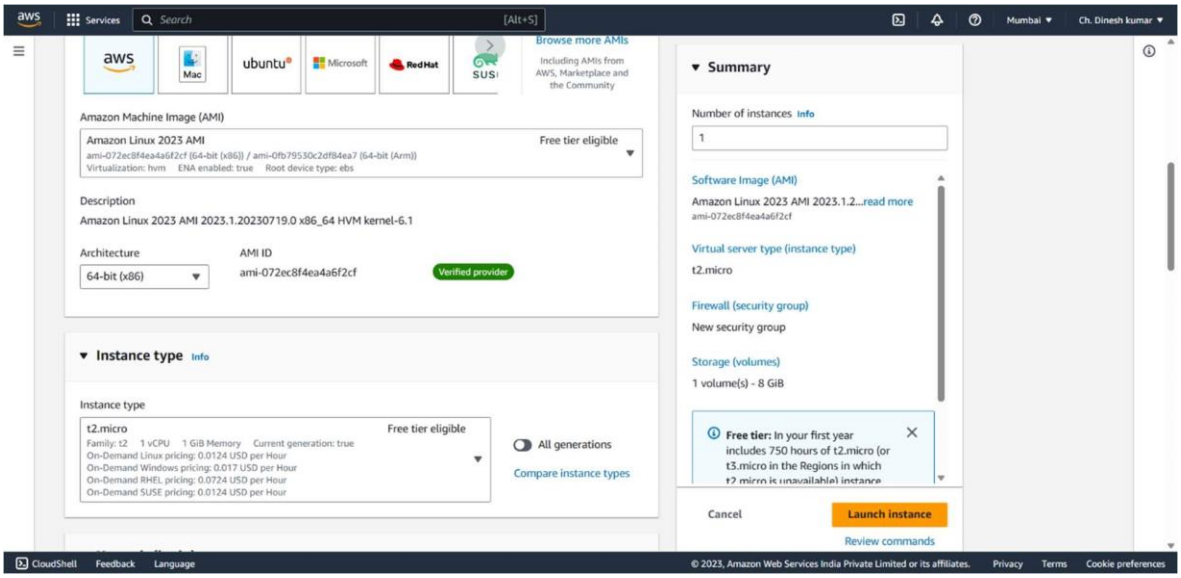
- Now again come back to the Console.
- Navigate to the EC2.
- Launch the EC2 Instance.



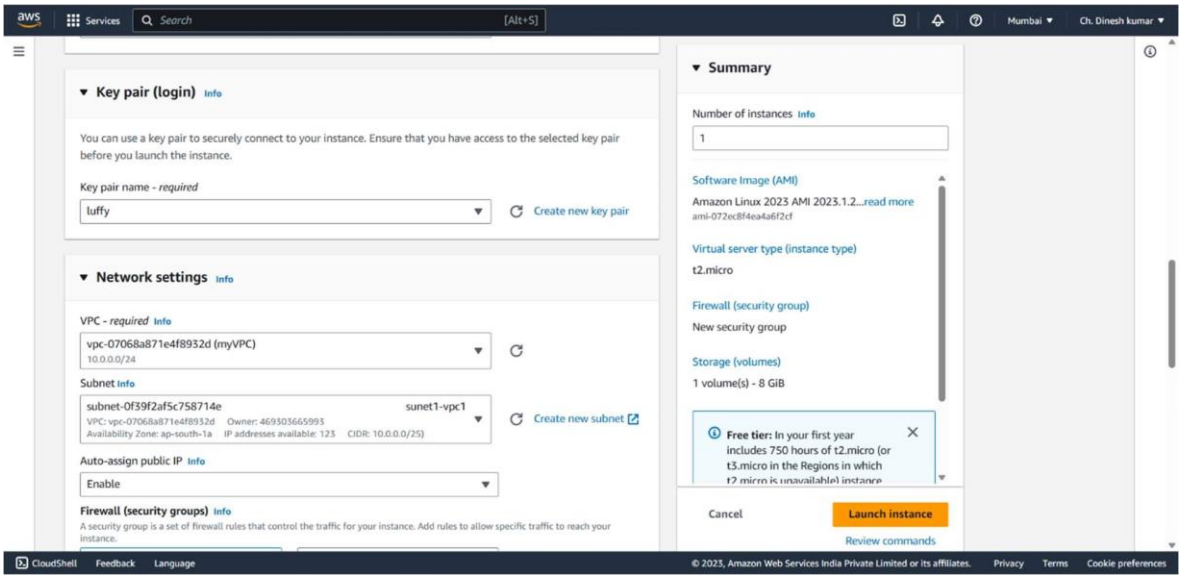
- Give name to the Instance.



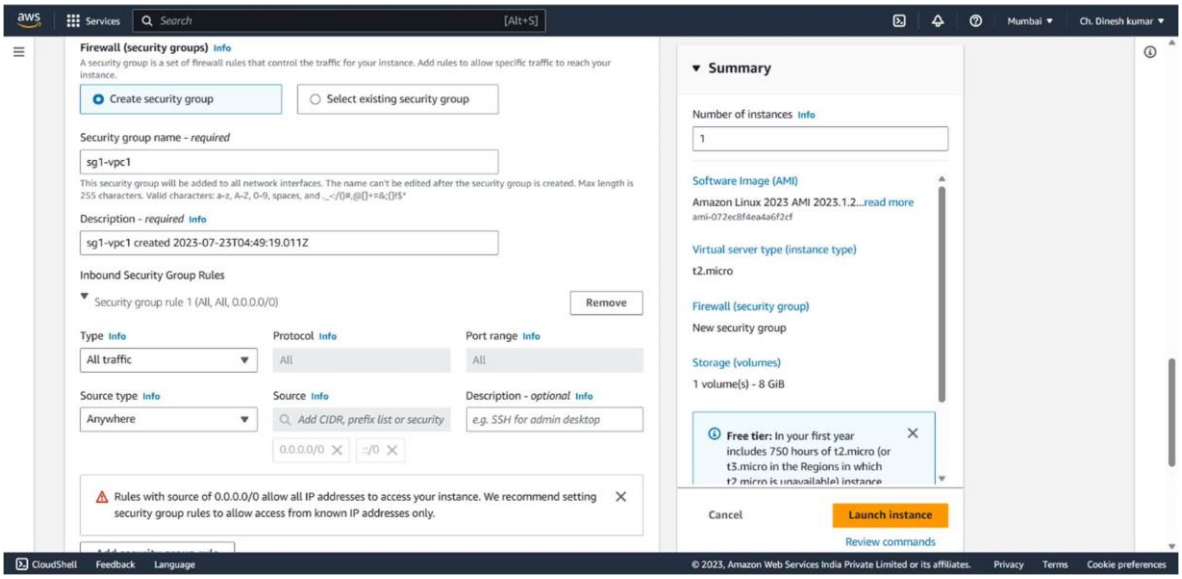
- We choose “Amazon Linux 2 AMI(HVM) – Kernel 5.10.SSD Volume Type in AMI.



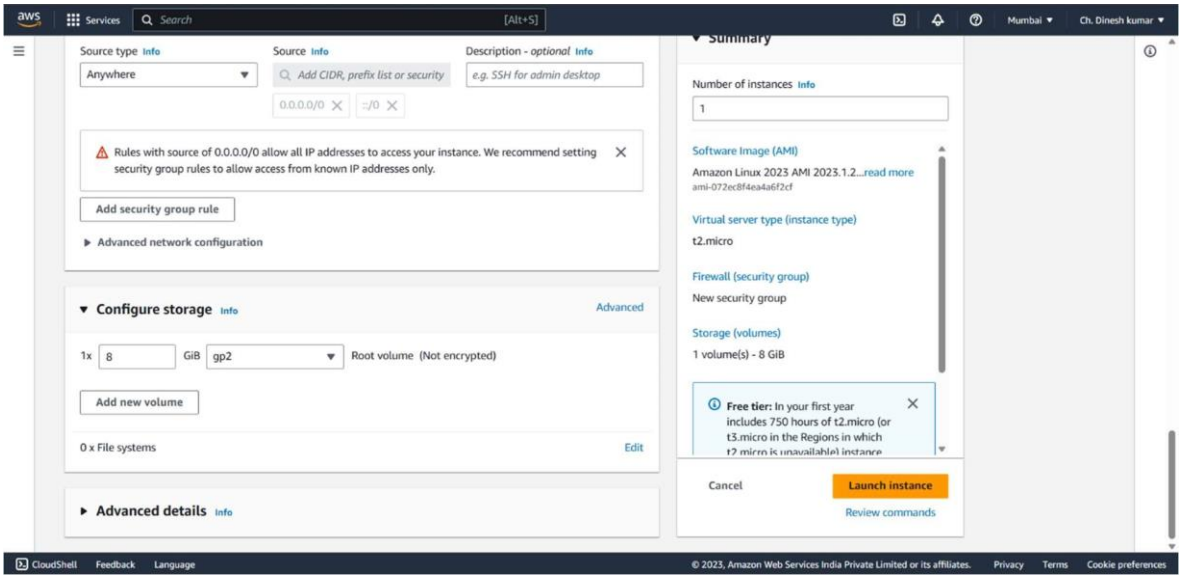
- Give created key pair, we given our key pair as “luffy”.

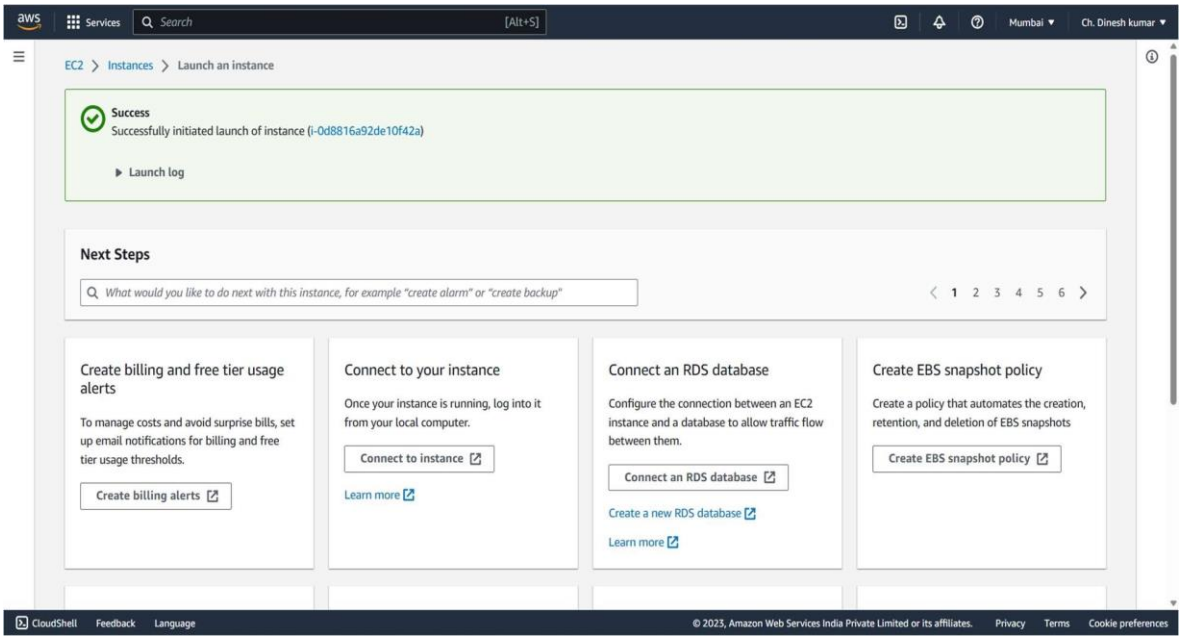


- Add “Type: All traffic”, “source type: Anywhere” in inbound security Group rules.

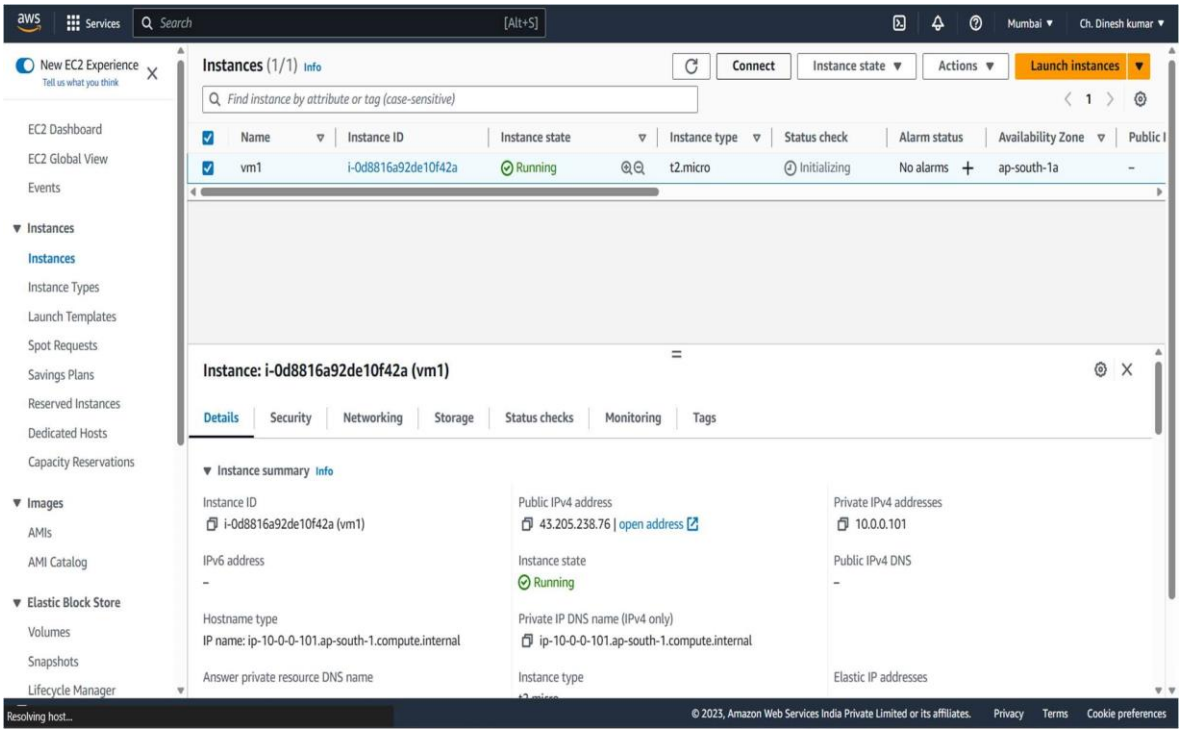


- Configure Storage add 1x 8 and GiB as gip2 and launch instance.

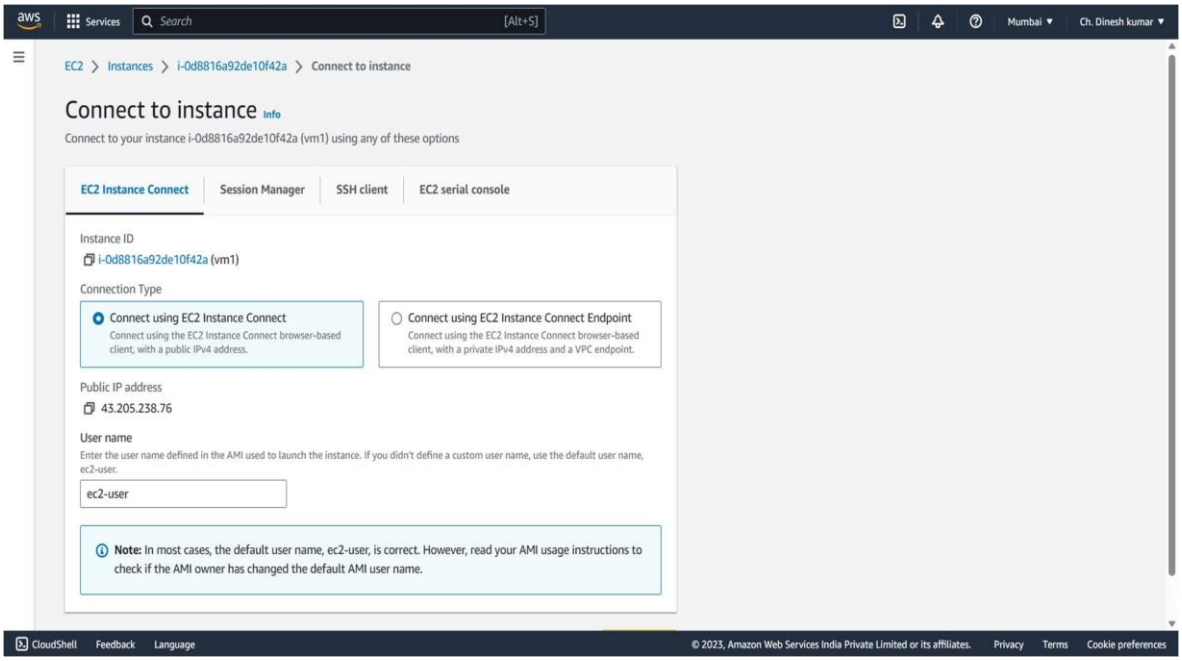




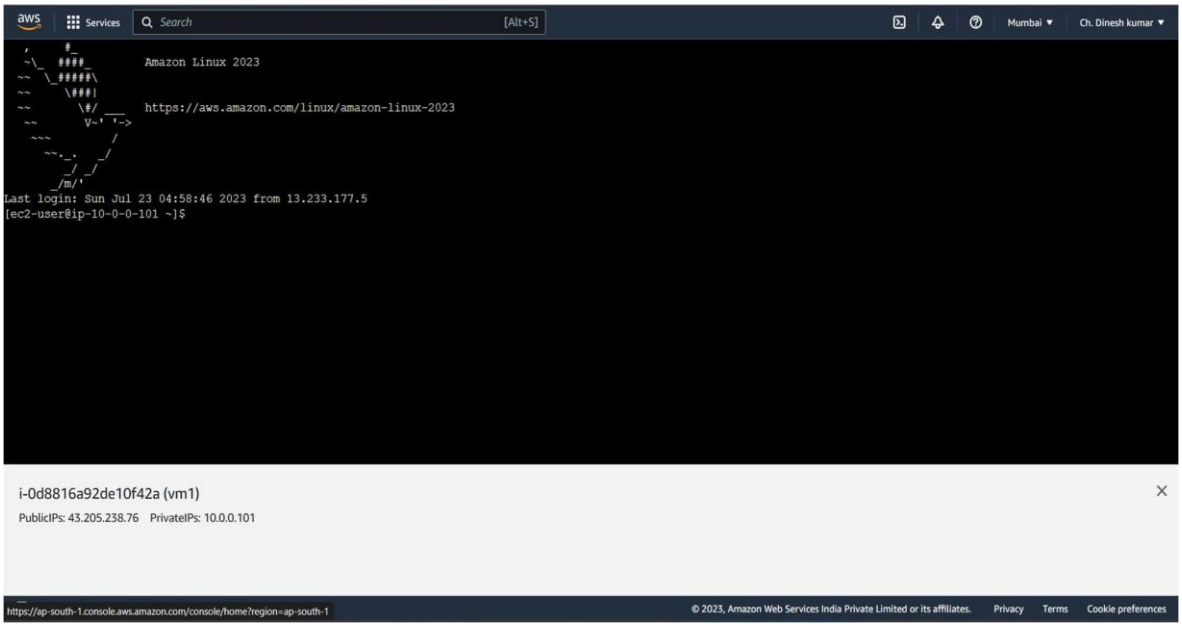
- Instance is successfully created.
- Then it will display as in the below picture.



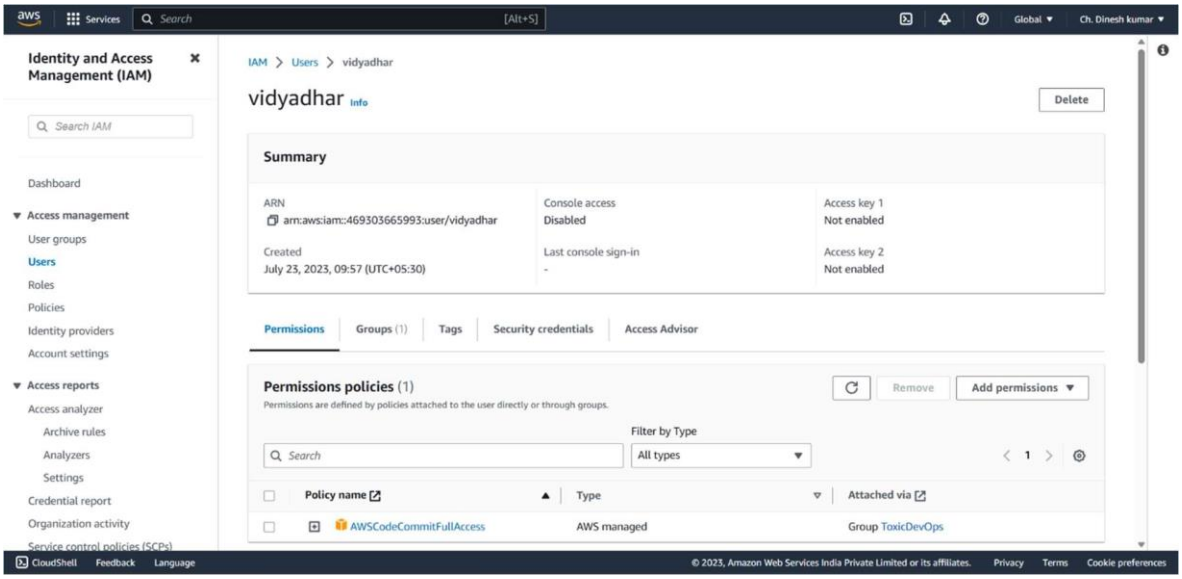
- Now Select the instance and connect the instance to the Console.



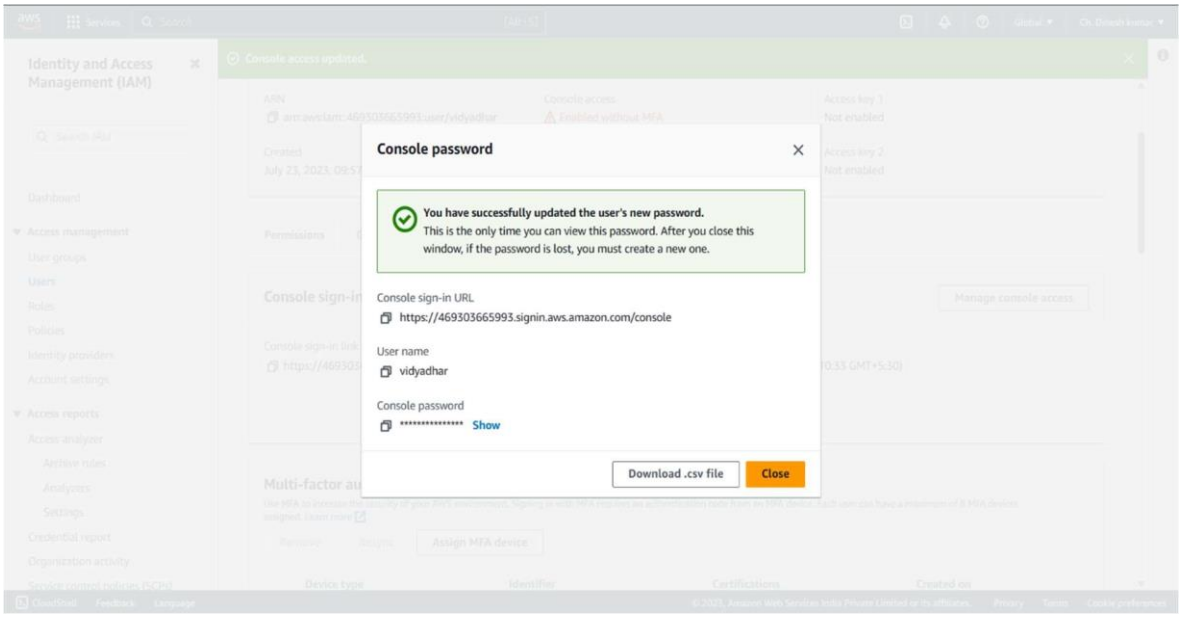
- If the instance is connected successfully then it will display the console as below picture.



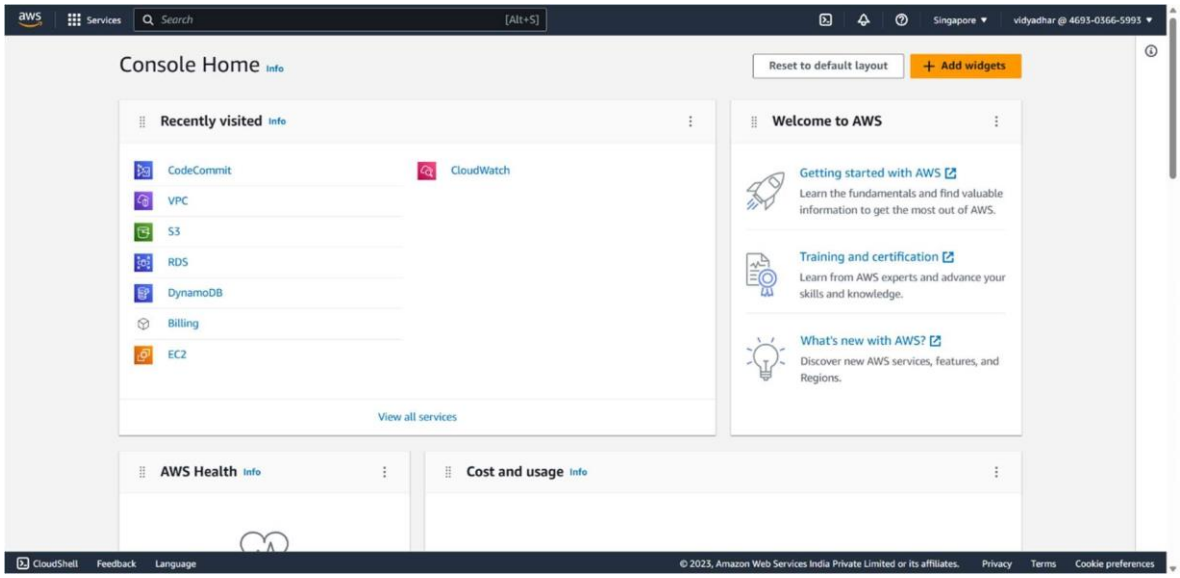
- Now again come back to the console and Navigate to the IAM.



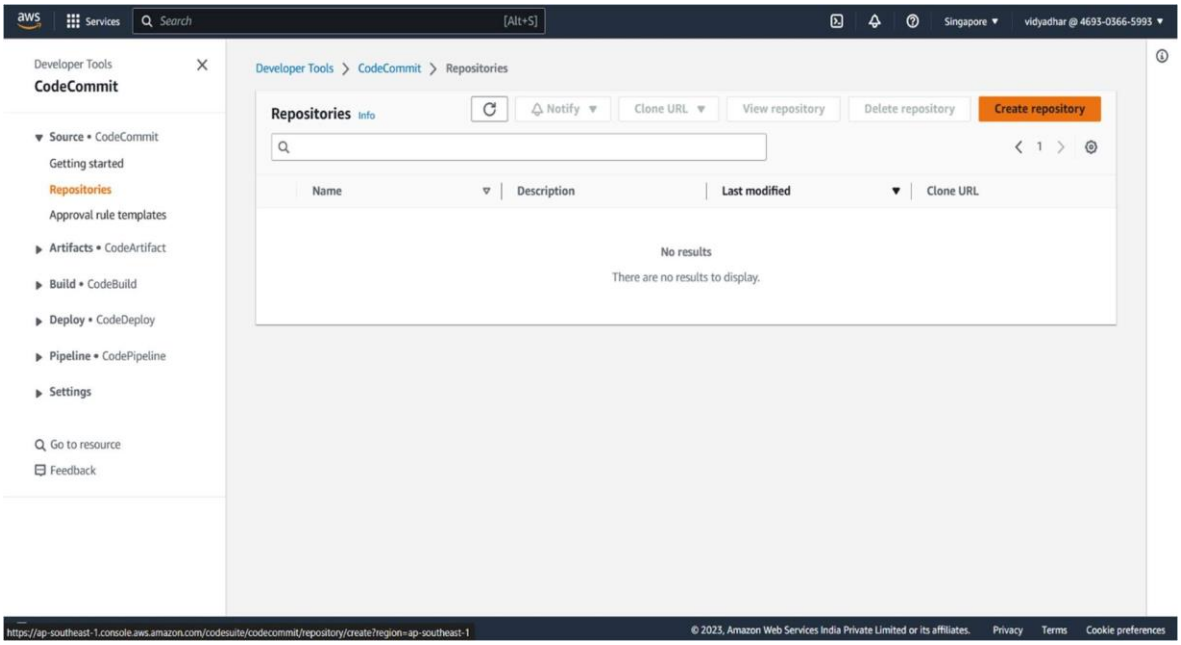
- Now go to the Security Credentials.
- Go to the console settings.
- It will provide the Console password and Download the Credentials.



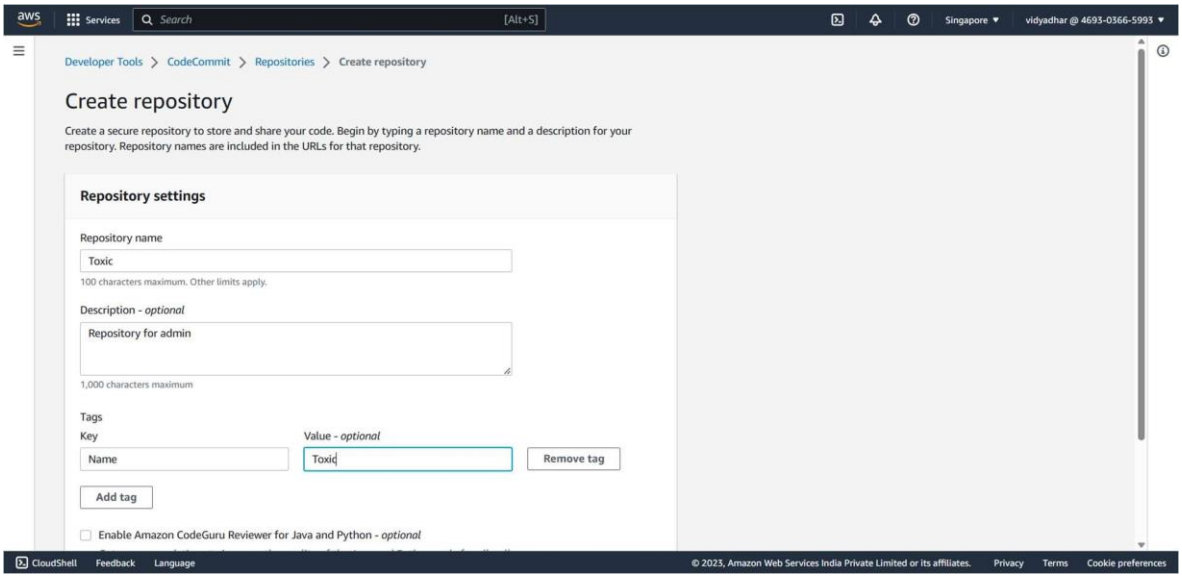
- Now login the IAM user account in another desktop.
- Navigate to the CodeCommit.



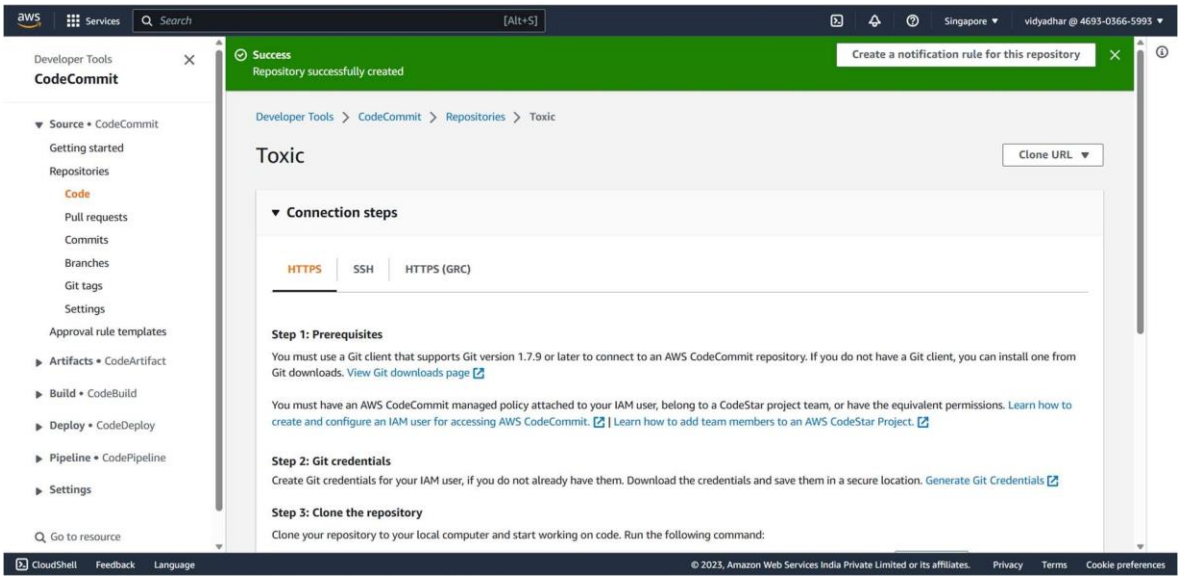
- After navigating to the Code commit and the home page of the code commit will as display in the below picture.



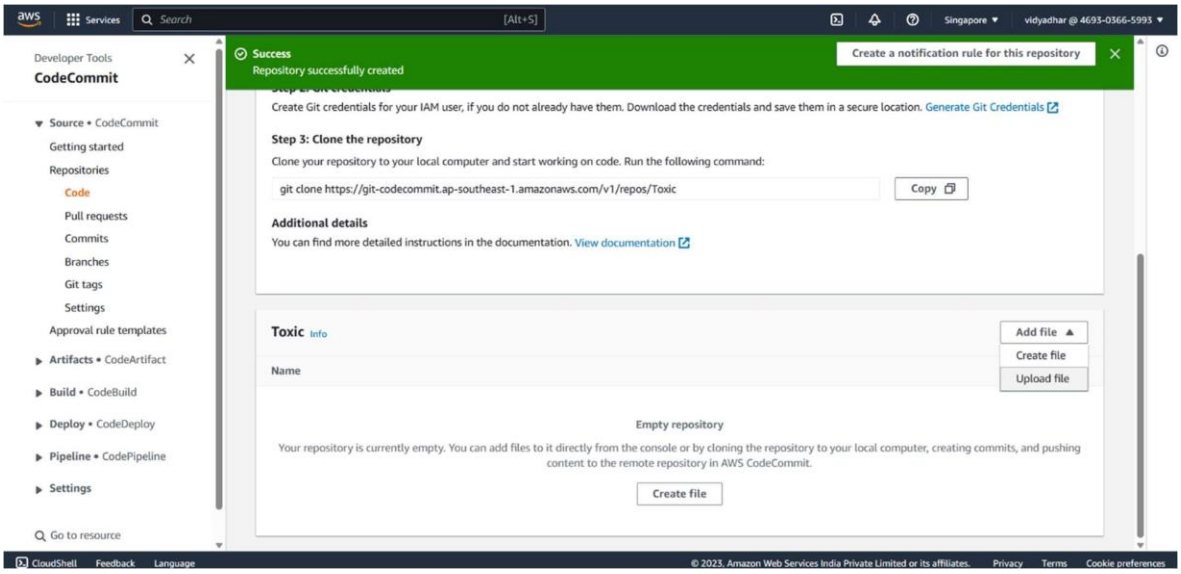
- Create the Repository.



- Give the Repository name.
- Tags
- Give key name and value.

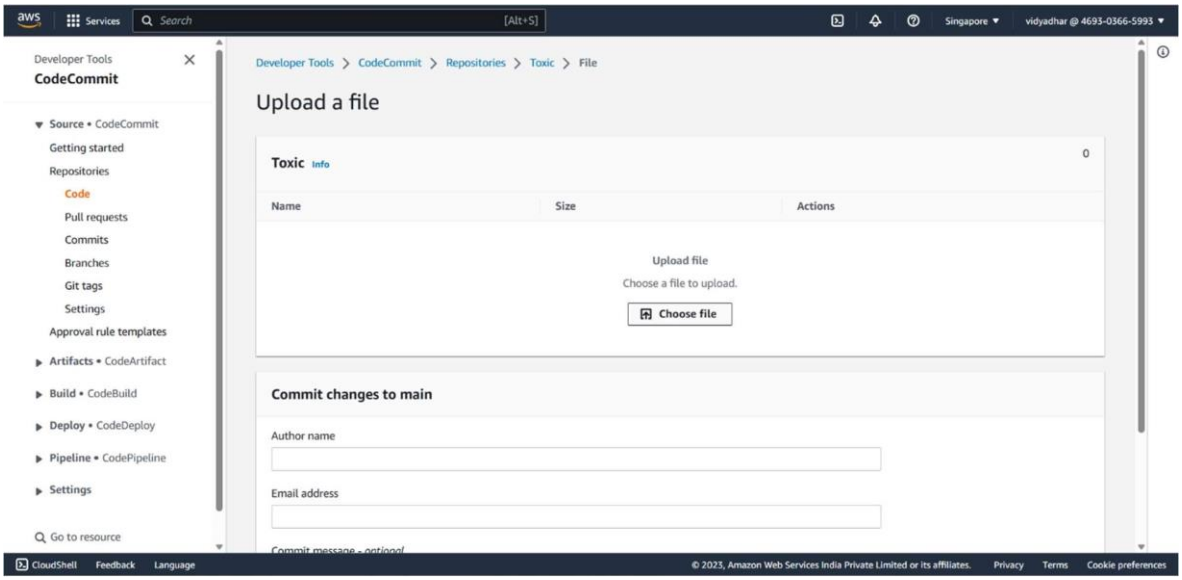


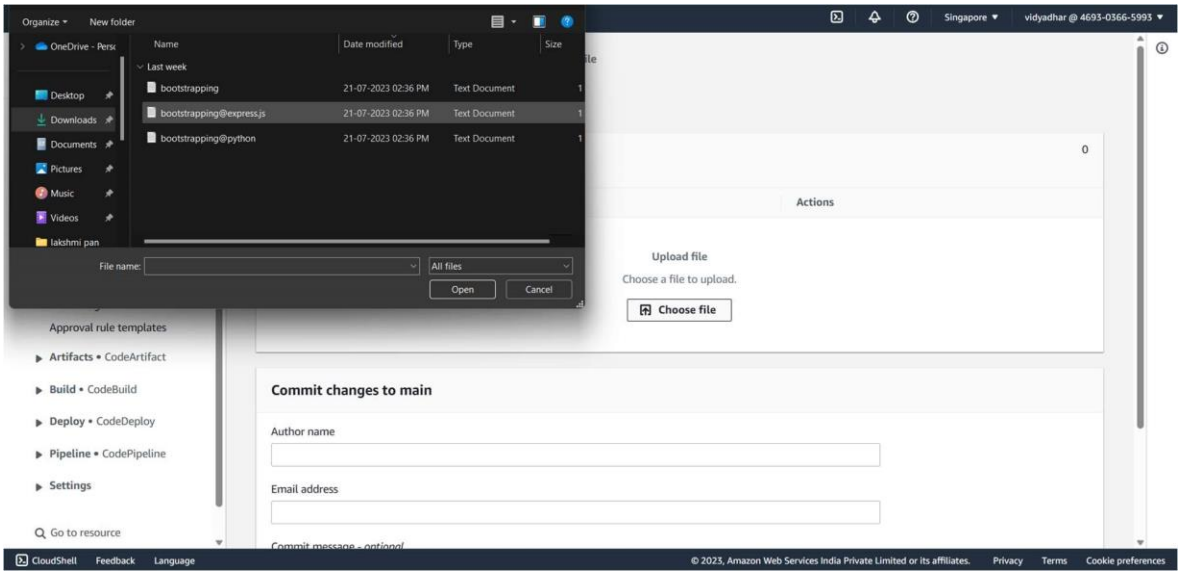
- After selecting the create section.
- Now connection steps are required.
- Scroll down and there will be create or upload File.



- Then choose upload file.

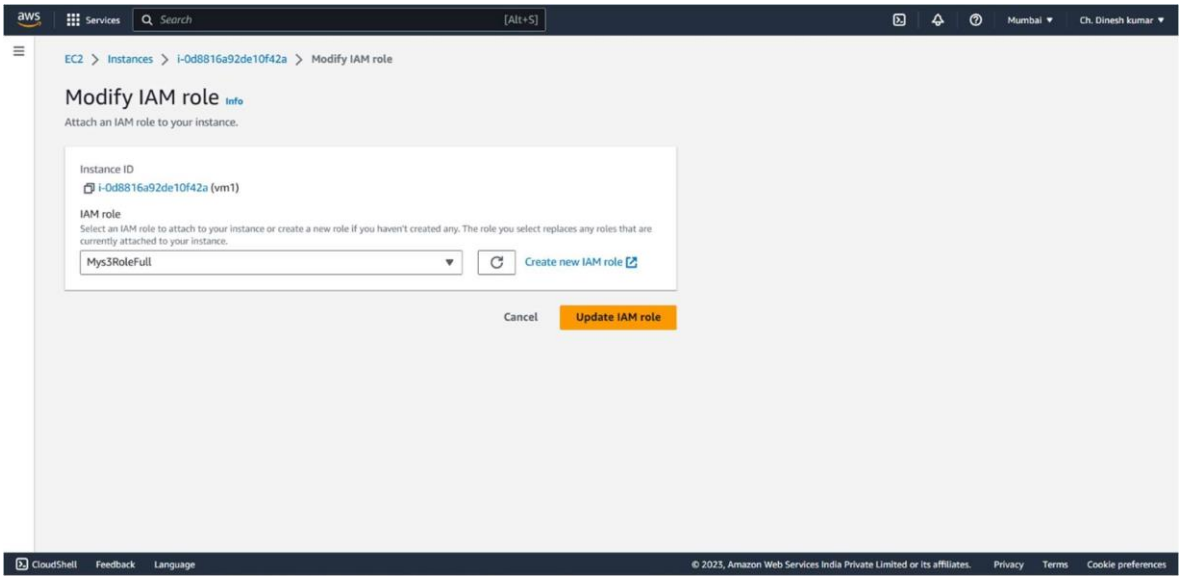
- Then the file will be browsed from our local file manager.



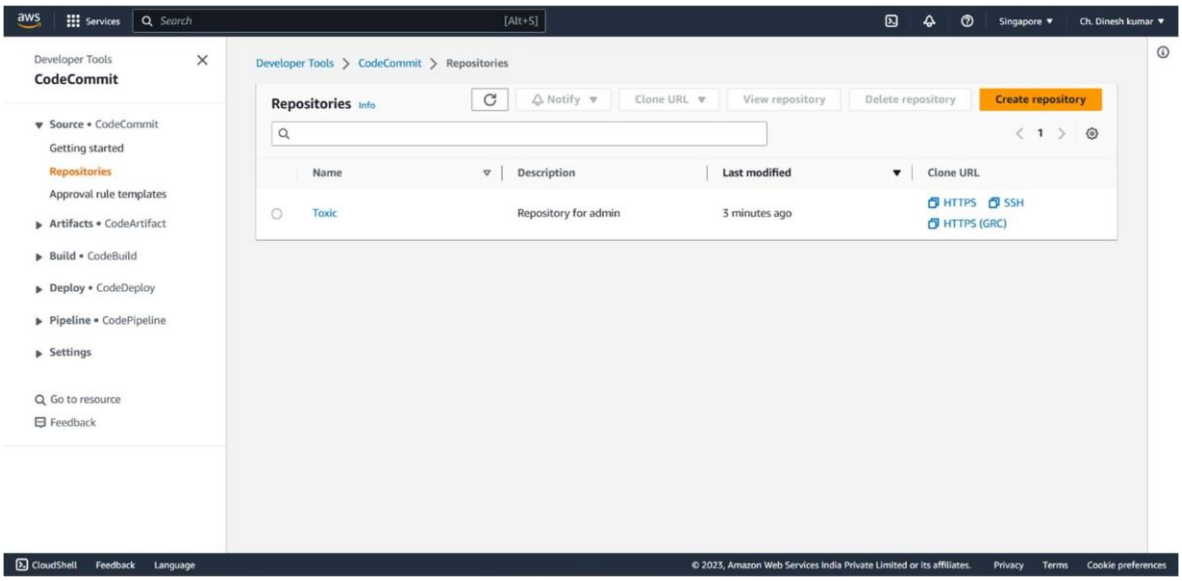


- Finally, the file will be uploaded.

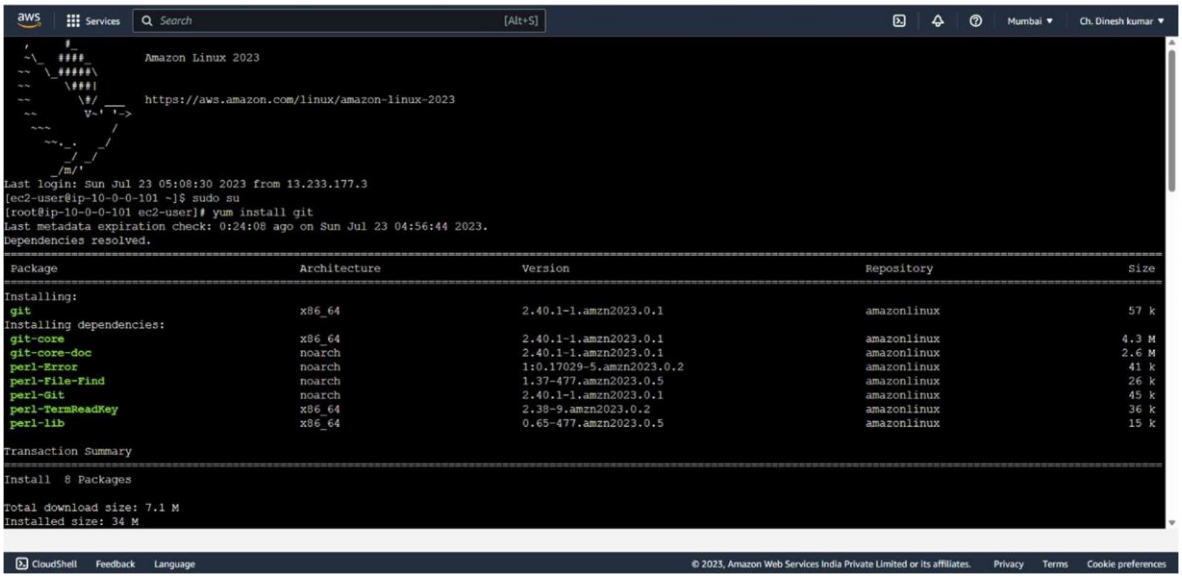
- Now go to root user
- Go to instance page.
- Select first instance and click actions. □ select Modify IAM role in actions.



- After that modify the IAM role from the Actions.



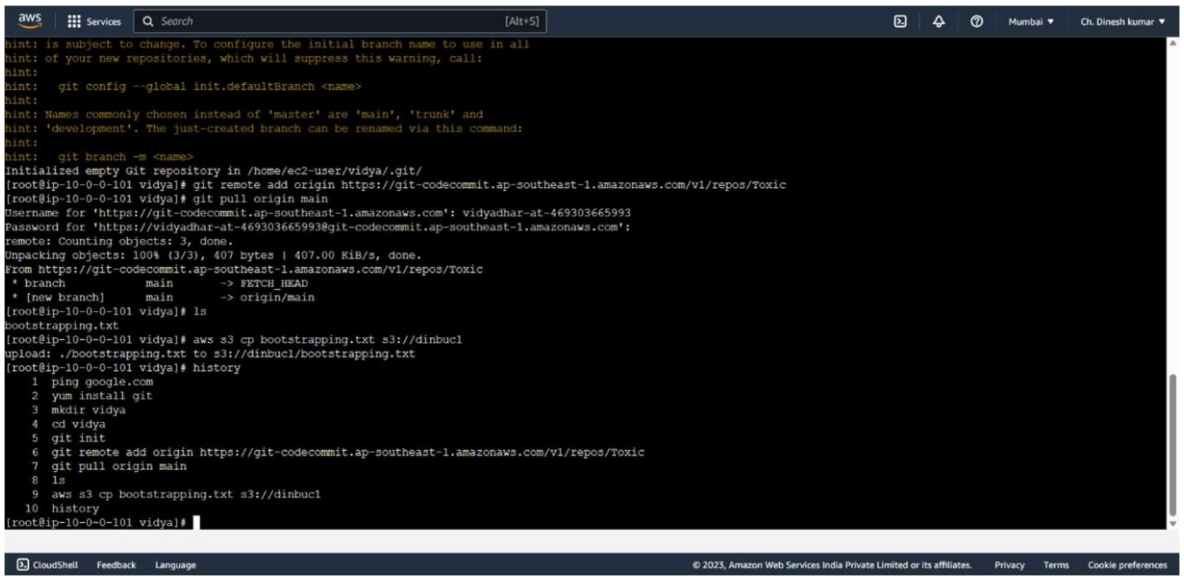
- Now open the console and install the git.
- After git installation.
- Copy the HTTPS link. Paste it in the console.



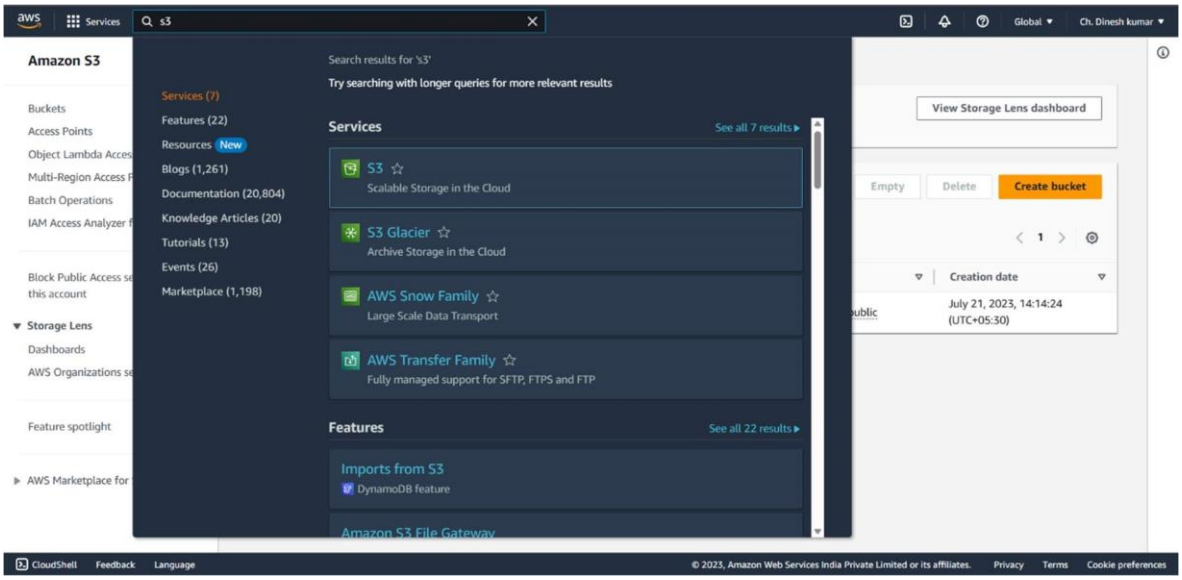
Follow the following commands in the console.

- Sudo su
- Yum install git -y
- Which git

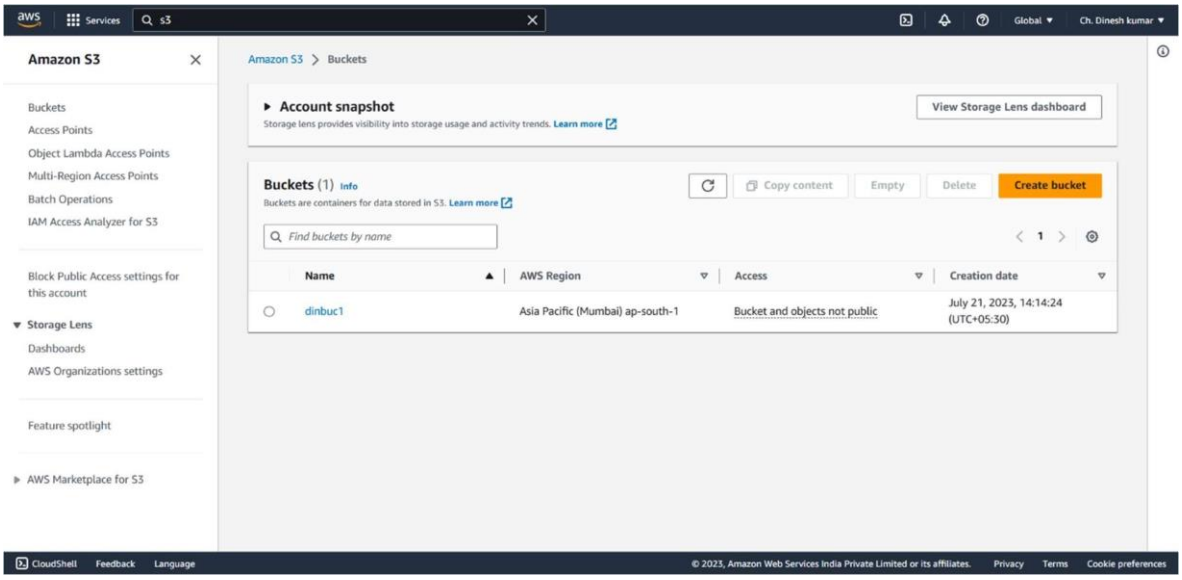
- mkdir name
- cd name
- git init
- git remote add origin (paste the HTTPS link here...)
- git pull origin main
- Here enter the Username present in console security credentials.
- Now enter the password present in console security credentials.
- ls
- Then the file will come from code commit to the instance using git



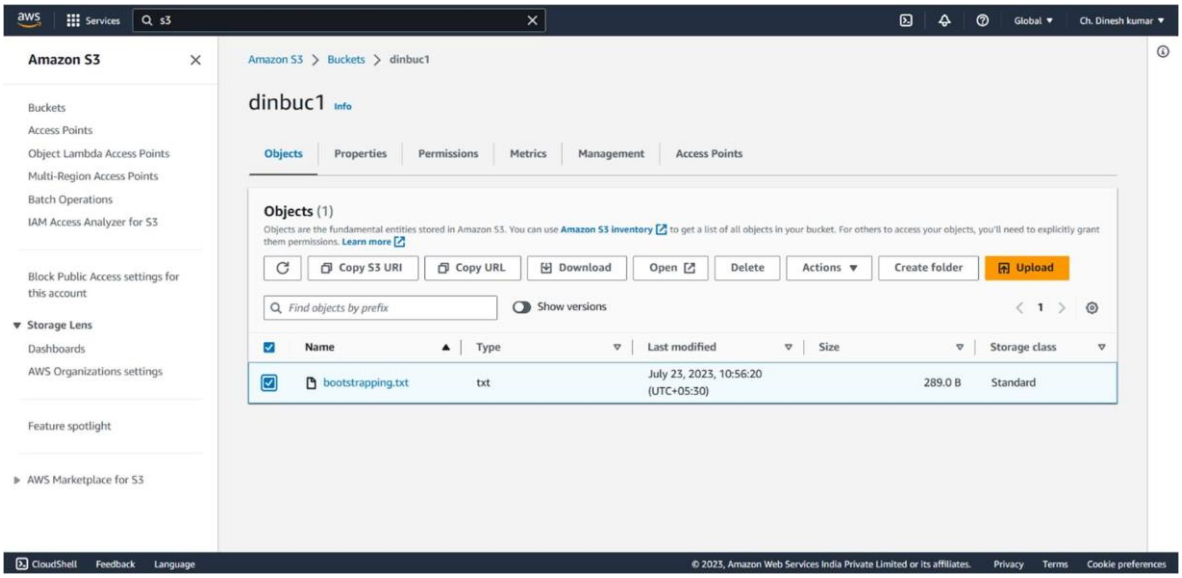
- Now come to the AWS console.
- Navigate the S3.



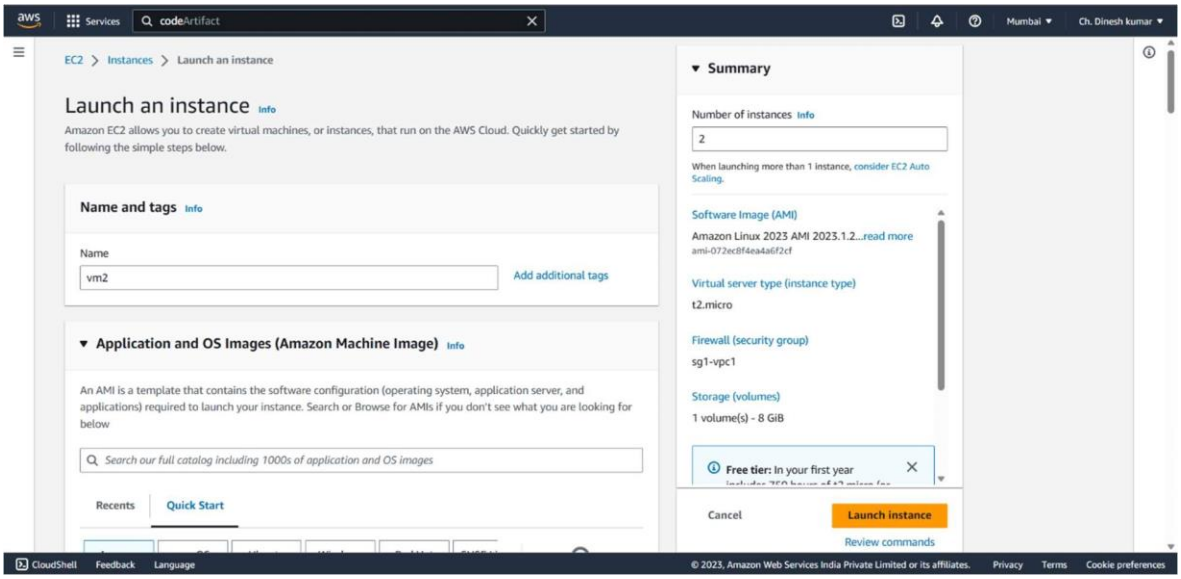
- Create the bucket in the S3.



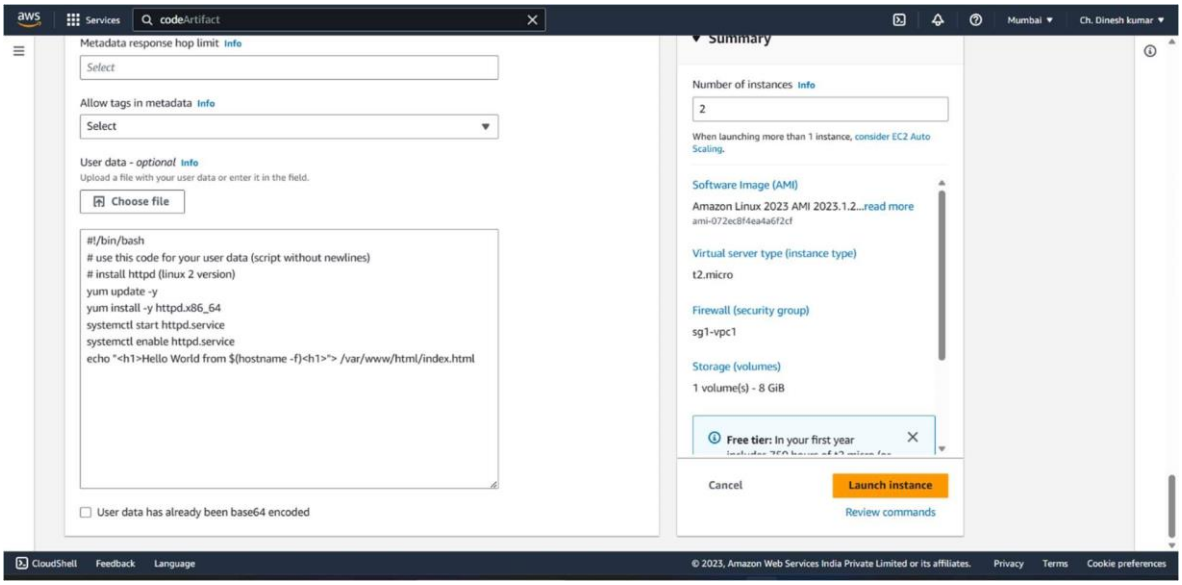
- In the bucket add object as the file that is taken from the code commit.



- Now Download that file to the local machine.

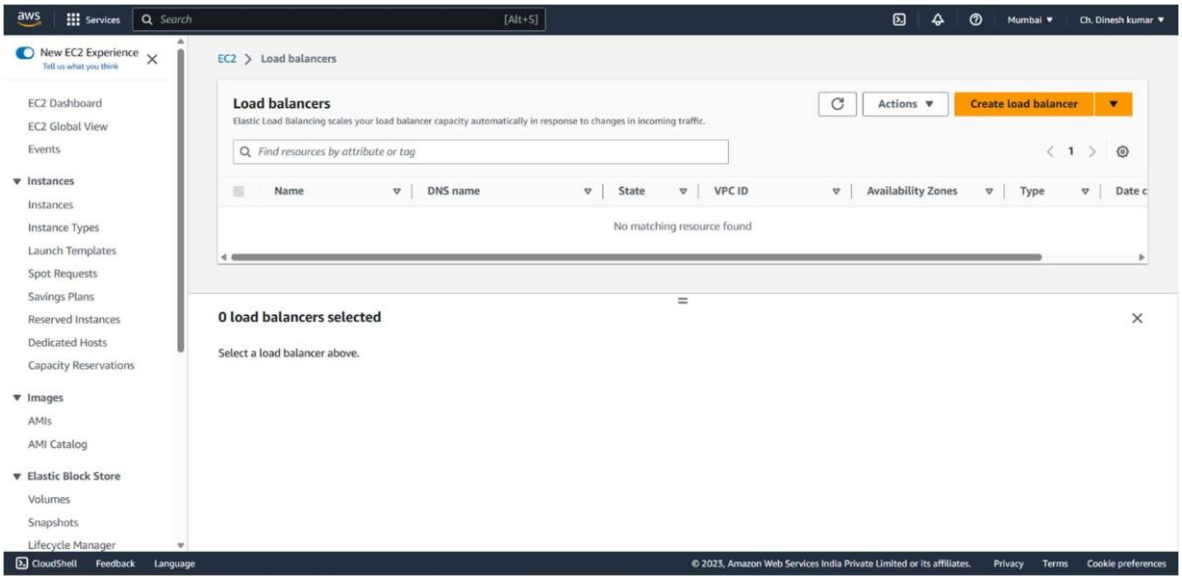


- Now again come to the AWS console. □ Navigate to the EC2 Instance.



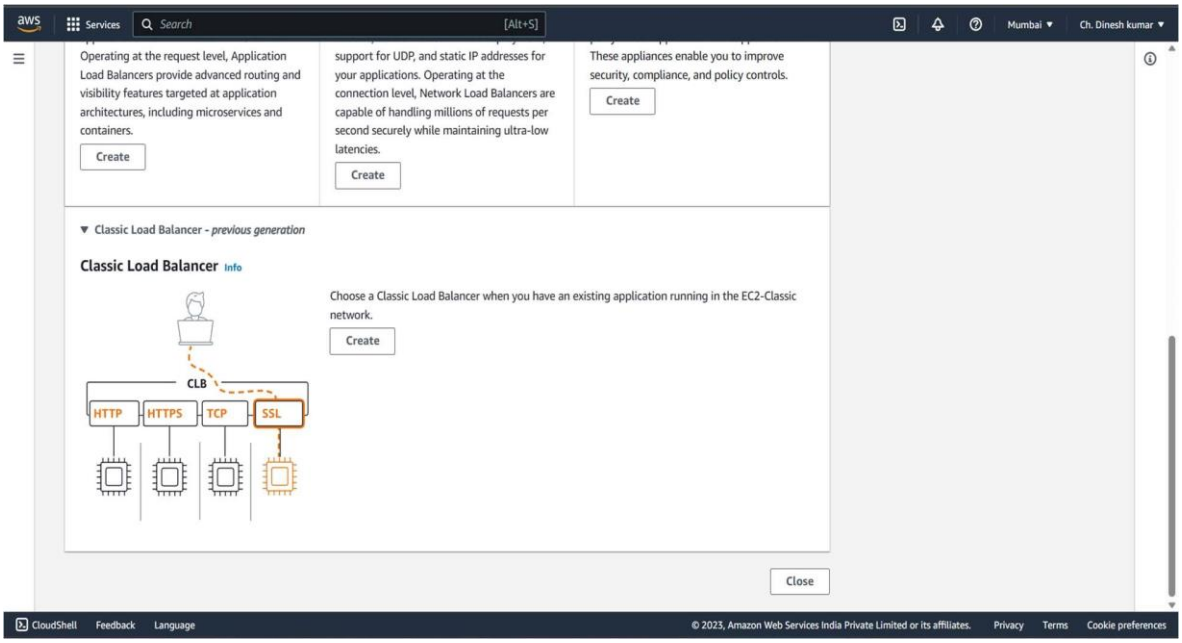
Here follow the following steps

- Create the two instances as created before.
- At the end in Additional Configuration.
- There will option to choose the file.
- Here choose the files that are downloaded to local machine before.
- After that launch the instance.
- Finally, the two instances will be created.
- Name that instance as instance1 and instance2.
- Now go to Elastic Load Balancer

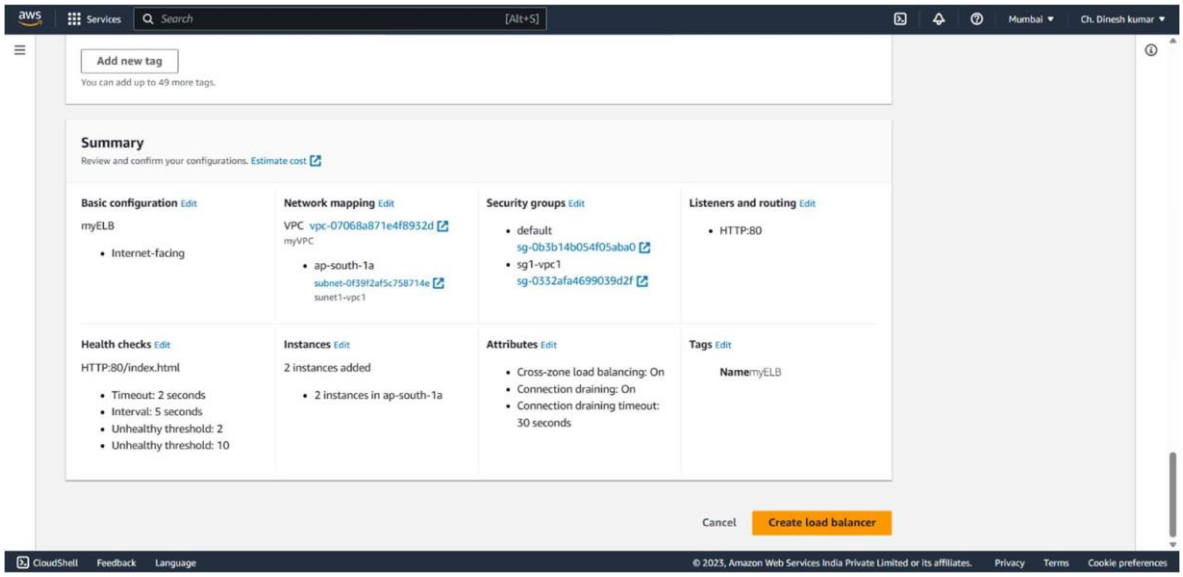


Load Balancer

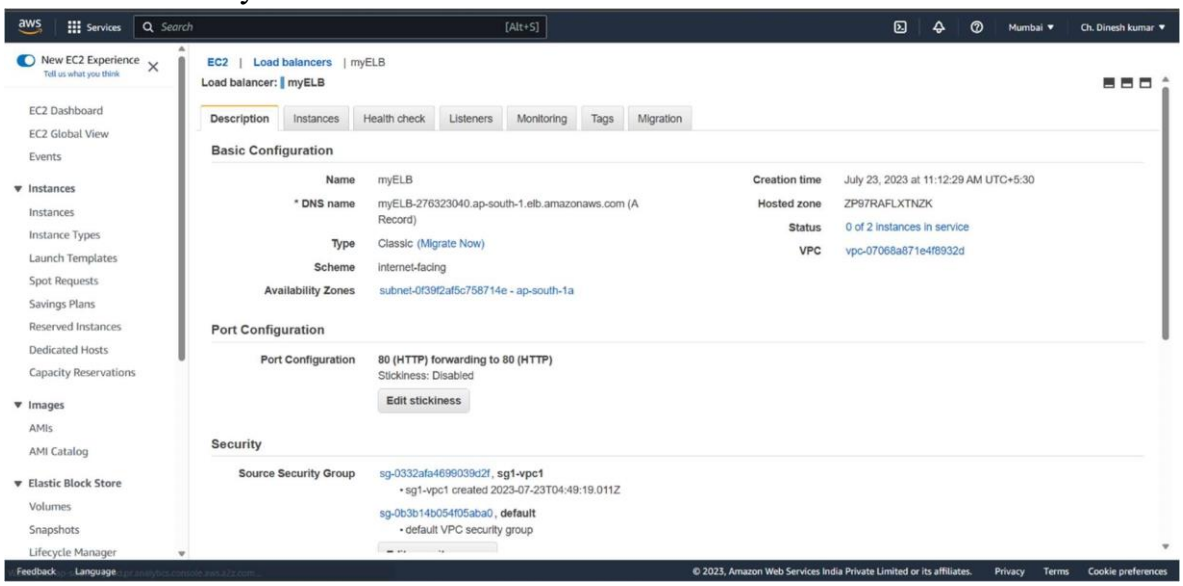
- Open the Load Balancer in the new tab.
- Create the load balancer.
- Select CLASSIC LOAD BALANCER.



- While creating the load balancer add the instance to the load balancer and add availability zone of that instance.

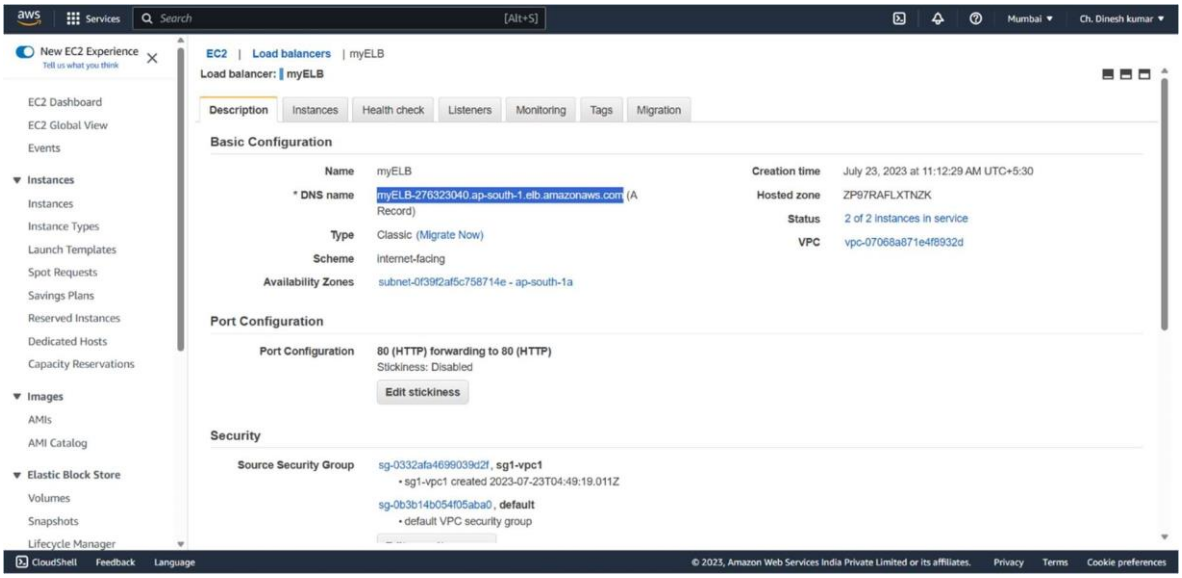


- Give security group as Default And instance security group.
- Then finally launch load balancer.



- After the load balancer is created the above page will display.
- In that select the Description.
- In that there will be DNS name.
- Before this check whether the instances are in Inservice or not.

- If there are Out of service wait for few minutes until it changes to the Inservice mode.



- Now copy the DNS name.
- And paste the DNS name in the new tab.
- Then it will print the content present in those files that we attached to the instances.
- When we paste the DNS name in the new tab then the first output will display as shown in the first photo.

Hello World from ip-10-0-0-114.ap-south-1.compute.internal

- When we refresh the page then the output will change.
- The server link will be changed.
- We attach two different files to the instances.
- First file output is above mentioned.
- The second file is present below.
- When we refresh the page, it will continuously change the output.

Conclusion

In conclusion, implementing an Elastic Load Balancer (ELB) using bootstrapping within a custom Virtual Private Cloud (VPC) offers a highly scalable and resilient solution for managing incoming traffic to your applications. By utilizing bootstrapping techniques, you can automate the provisioning and configuration of instances, making it easier to maintain a consistent and up-to-date environment.

The use of a custom VPC provides greater control over network settings, enabling you to design a secure and isolated environment for your applications. With ELB distributing incoming traffic across multiple instances, the workload is evenly balanced, ensuring optimal performance and minimizing the risk of overload in any single instance.

Overall, leveraging an Elastic Load Balancer in conjunction with bootstrapping within a custom VPC empowers you to build a robust and flexible architecture, capable of meeting the demands of modern applications and delivering a seamless experience to users.