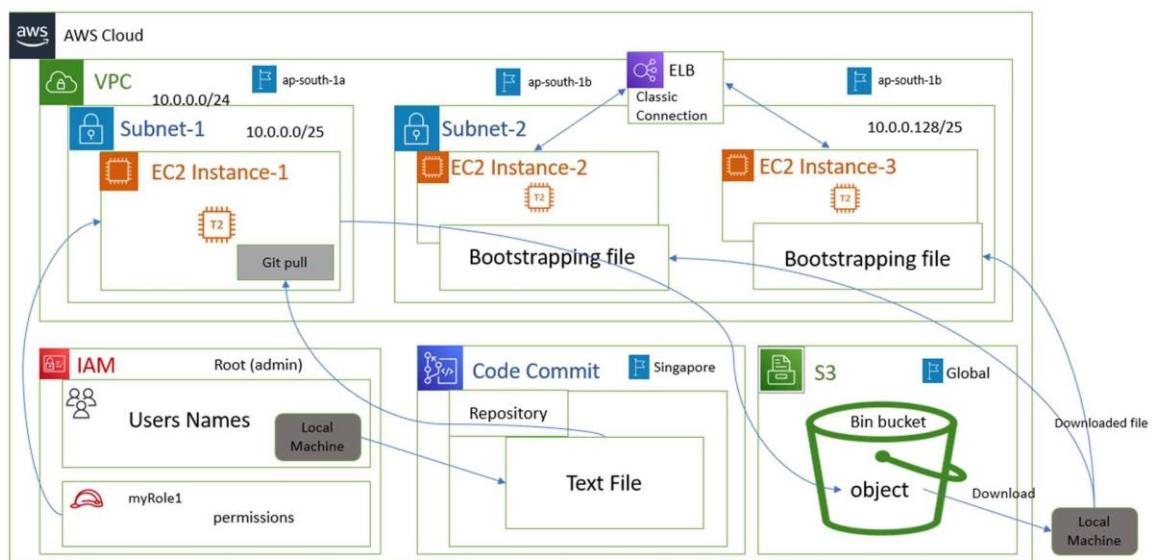


# An Architecture of Elastic Load Balancer (ELB) using Bootstrapping Method In custom VPC - 7 tier.

## AWS Services Used

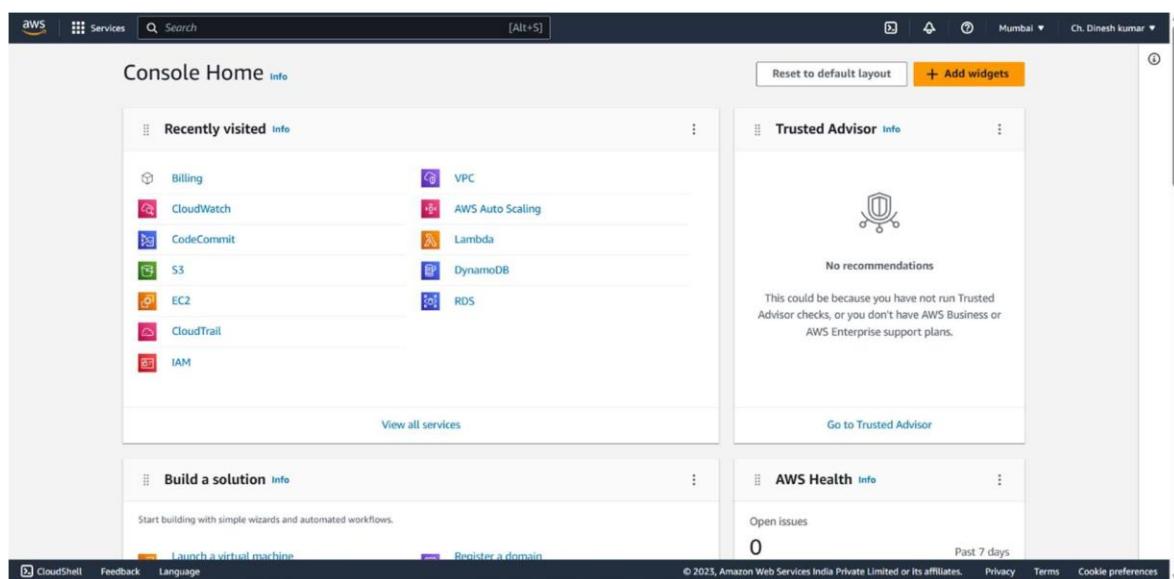
- Instance EC2
- Custom VPC
- Code Commit
- IAM User & Groups
- IAM Role
- S3 Buckets
- EC2 Load Balancer

## 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.

The screenshot shows the AWS Identity and Access Management (IAM) dashboard. On the left, a sidebar lists navigation options like 'Identity and Access Management (IAM)', 'Dashboard', 'Access management', 'Access reports', and 'What's new'. The main content area displays 'Security recommendations' with a red warning icon, 'Root user has no active access keys' (green checkmark), and 'IAM resources' (User groups: 0, Users: 0, Roles: 12, Policies: 2, Identity providers: 0). A vertical sidebar on the right contains 'AWS Account' details (Account ID: 469303665993, Account Alias: 469303665993, Sign-In URL: https://469303665993.sigin.aws.amazon.com/console), 'Quick Links' (My security credentials), and 'Tools' (Policy simulator).

IAM dashboard

Security recommendations !

⚠ Add MFA for root user  
Add MFA for root user - Enable multi-factor authentication (MFA) for the root user to improve security for this account.

✓ Root user has no active access keys  
Using access keys attached to an IAM user instead of the root user improves security.

**IAM resources**

User groups	Users	Roles	Policies	Identity providers
0	0	12	2	0

What's new !

Updates for features in IAM

View all !

- Advanced Notice: Amazon S3 will automatically enable S3 Block Public Access and disable access control lists for all new buckets starting in April 2023. 7 months ago
- AWS IAM Identity Center now supports session management capabilities for AWS Command Line Interface (AWS CLI) and SDKs. 8 months ago
- AWS Lambda announces support for Attribute-Based Access Control (ABAC) in AWS GovCloud (US) Regions.. 8 months ago
- Amazon ElastiCache simplifies password rotations with Secrets Manager. 8 months ago

AWS Account

Account ID: 469303665993

Account Alias: 469303665993 Create

Sign-In URL for IAM users in this account: <https://469303665993.sigin.aws.amazon.com/console>

Quick Links

My security credentials

Manage your access keys, multi-factor authentication (MFA) and other credentials.

Tools

Policy simulator

The simulator evaluates the policies that you choose and determines the effective permissions for each of the actions that you specify.

© 2023, Amazon Web Services India Private Limited or its affiliates. [Privacy](#) [Terms](#) [Cookie preferences](#)

- go to user group and create a user group

The screenshot shows the AWS Identity and Access Management (IAM) service interface. The left sidebar is titled "Identity and Access Management (IAM)" and includes sections for Dashboard, Access management (User groups, Users, Roles, Policies, Identity providers, Account settings), Access reports (Access analyzer, Archive rules, Analyzers, Settings, Credential report, Organization activity), and Service control policies (SCPs). The main content area is titled "User groups (0) Info" and contains a message stating "A user group is a collection of IAM users. Use groups to specify permissions for a collection of users." A search bar at the top of the content area says "Filter User groups by property or group name and press enter". Below the search bar is a table header with columns: Group name, Users, Permissions, and Creation time. The table body displays the message "No resources to display". At the bottom right of the content area are "Create group" and "Delete" buttons.

The screenshot shows the "Create user group" page within the AWS IAM service. The left sidebar is identical to the previous screenshot. The main content area is titled "Create user group". It has a section titled "Name the group" with a "User group name" input field containing "ToxicDevOps". Below this is a section titled "Add users to the group - Optional (0) Info" which states "An IAM user is an entity that you create in AWS to represent the person or application that uses it to interact with AWS. A user can belong to up to 10 groups." A search bar and a table with columns "User name" and "Groups" are shown, both currently displaying "No resources to display". At the bottom of the page is a section titled "Attach permissions policies - Optional (862) Info" with a "Create policy" button. The footer of the page includes links for CloudShell, Feedback, Language, and copyright information: "© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences".

- Add AWSCodeCommitFullAccess to the user group

AWS Services Search [Alt+S] Global Ch. Dinesh kumar

Identity and Access Management (IAM)

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.

Filter policies by property or policy name and press enter. 3 matches < 1 >

"codecommit" X Clear filters

Policy name	Type	Description
<input checked="" type="checkbox"/> AWSCodeCommitFullAccess	AWS managed	Provides full access
<input type="checkbox"/> AWSCodeCommitReadOnly	AWS managed	Provides read only access
<input type="checkbox"/> AWSCodeCommitPowerUser	AWS managed	Provides full access

Cancel Create group Show desktop

CloudShell Feedback Language © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

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.

Filter User groups by property or group name and press enter < 1 >

Group name	Users	Permissions	Creation time
ToxicDevOps	Loading	Defined	Now

https://us-east-1.console.aws.amazon.com/iamv2/home?region=ap-south-1#/identity\_group... © 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

Screenshot of the AWS IAM User Groups page showing a newly created group "ToxicDevOps".

The screenshot shows the AWS IAM interface with the following details:

- User Groups:** A single group named "ToxicDevOps" is listed. It has no users assigned and no permissions attached.
- Permissions:** The status is "Defined".
- Creation Time:** Now.

Below this, there is a "Specify user details" step for creating a new user named "vidyadhar". The "User name" field contains "vidyadhar". There is an optional checkbox for "Provide user access to the AWS Management Console - optional". A note indicates that if programmatic access is required, AWS CodeCommit or Amazon Keyspaces can generate access keys after the user is created.

- create and add user in user group

**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)**

Group name	Users	Attached policies	Created
ToxicDevOps	0	AWSCodeCommitFullAccess	2023-07-23 (1 minute ago)

**Set permissions boundary - optional**

Cancel Previous Next

**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.

**Users (1) Info**

An IAM user is an identity with long-term credentials that is used to interact with AWS in an account.

User name	Groups	Last activity	MFA	Password age	Active key age
vidyadhar	ToxicDevOps	Never	None	None	-

View user

Screenshot of the AWS IAM Roles page showing a list of 12 roles. The roles are listed in descending order of last activity. The first role is 'AWSRoleForAmazonElasticFileSystem'.

Role name	Trusted entities	Last act...
AWSRoleForAmazonElasticFileSystem	AWS Service: elasticfilesystem (Service-Linked Role)	Yesterday
AWSRoleForApplicationAutoScaling_DynamoDBTable	AWS Service: dynamodb.application-autoscaling (Service-Linked Role)	3 days ago
AWSRoleForAutoScaling	AWS Service: autoscaling (Service-Linked Role)	2 days ago
AWSRoleForBackup	AWS Service: backup (Service-Linked Role)	19 hours ago
AWSRoleForElasticLoadBalancing	AWS Service: elasticloadbalancing (Service-Linked Role)	17 hours ago
AWSRoleForRDS	AWS Service: rds (Service-Linked Role)	1 hour ago
AWSRoleForSupport	AWS Service: support (Service-Linked Role)	-
AWSRoleForTrustedAdvisor	AWS Service: trustedadvisor (Service-Linked Role)	-
myfunc-role-uxpbql8o	AWS Service: lambda	2 days ago
MYrole1S3FULL	AWS Service: ec2	17 hours ago
role_monitoring.role	AWS Service: monitoring.role	

Screenshot of the 'Create role' wizard, Step 1: Select trusted entity. It shows five options: AWS service (selected), AWS account, Web Identity, SAML 2.0 federation, and Custom trust policy.

Common use cases include EC2 (selected) and Lambda.

Use cases for other AWS services: Choose a service to view use case

- Add AmazonS3FullAccess and AWSCodeCommitFullAccess to IAM Role

**Add permissions**

**Permissions policies (Selected 1/862)**

"s3full" X Clear filters

Policy name	Type	Description
AmazonS3FullAccess	AWS m...	Provides full access to all buckets via the AWS Management Console.

**Set permissions boundary - optional**

Set a permissions boundary to control the maximum permissions this role can have. This is not a common setting, but you can use it to delegate permission management to others.

Cancel Previous Next

**Name, review, and create**

**Role details**

**Role name**  
Enter a meaningful name to identify this role.  
**MyS3RoleFull**

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

**Step 1: Select trusted entities**

```

1< {
2   "Version": "2012-10-17",
3   "Statement": [
4     {
5       "Effect": "Allow",
6       "Action": [
7         "sts:AssumeRole"
8       ],
9     }
10 }

```

CloudShell Feedback Language

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

The screenshot shows the AWS Identity and Access Management (IAM) console. A search bar at the top contains the query 'vpc'. Below the search bar, there's a sidebar with various navigation options like 'Identity and Access Management (IAM)', 'Dashboard', 'Access management', 'Access reports', and 'Service control policies (SCPs)'. The main content area displays search results for 'vpc' under 'Services' and 'Features'. Under 'Services', it lists 'VPC', 'AWS Firewall Manager', 'Detective', and 'Managed Services'. Under 'Features', it lists 'Dashboard'. On the right side, a modal window titled 'View role' is open, showing a list of roles with their last activity times.

The screenshot shows the AWS VPC dashboard. At the top, there are buttons for 'Create VPC' and 'Launch EC2 Instances'. Below this, a note says 'Note: Your Instances will launch in the Asia Pacific region.' The left sidebar has sections for 'Virtual private cloud' (including 'Your VPCs', 'Subnets', 'Route tables', etc.) and 'Security' (including 'Network ACLs', 'Security groups'). The main content area shows 'Resources by Region' with a grid of icons for VPCs, NAT Gateways, Subnets, VPC Peering Connections, Route Tables, Network ACLs, Internet Gateways, Security Groups, Egress-only Internet Gateways, Customer Gateways, DHCP option sets, Virtual Private Gateways, and Elastic IPs. To the right, there are sections for 'Service Health', 'Settings', 'Additional Information' (about AWS Network Manager), and 'Site-to-Site VPN Connections'.

## Create VPC

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

- Finally Create the VPC

A screenshot of the AWS VPC 'Create VPC' page. The 'VPC settings' section is visible, showing the following configuration:

- Resources to create:** VPC only (selected)
- Name tag - optional:** myVPC
- IPv4 CIDR block:** IPv4 CIDR manual input (selected)
  - 10.0.0.0/24
- IPv6 CIDR block:** No IPv6 CIDR block selected

The bottom of the page includes standard AWS navigation links: CloudShell, Feedback, Language, and footer links for Privacy, Terms, and Cookie preferences.

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

A screenshot of the AWS VPC dashboard showing the details of the newly created VPC. The main card displays the following information:

VPC ID	State	DNS hostnames	DNS resolution
vpc-07068a871e4f8932d	Available	Disabled	Enabled

Below the main card, there are sections for Resource map, Subnets (0), Route tables (1), and Tags.

The left sidebar shows the VPC dashboard navigation and lists for Subnets, Route tables, Internet gateways, Egress-only internet gateways, DHCP option sets, Elastic IPs, Managed prefix lists, Endpoints, Endpoint services, NAT gateways, and Peering connections. The security section lists Network ACLs and Security groups.

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

The screenshot shows the AWS VPC Subnets list page. On the left, there's a navigation sidebar with options like VPC dashboard, EC2 Global View, Filter by VPC, Virtual private cloud, Subnets, Security, and CloudShell. The main area displays a table titled "Subnets (3) Info" with columns: Name, Subnet ID, State, VPC, and IPv4 CIDR. The table lists three subnets: subnet-006c6ddfc85b63072, subnet-002aae5ca688485e1, and subnet-0e99dc83ca0df711f, all in the "Available" state. A "Create subnet" button is located at the top right of the table.

Name	Subnet ID	State	VPC	IPv4 CIDR
-	subnet-006c6ddfc85b63072	Available	vpc-07ae3320b4eee150d	172.31.32.0/20
-	subnet-002aae5ca688485e1	Available	vpc-07ae3320b4eee150d	172.31.16.0/20
-	subnet-0e99dc83ca0df711f	Available	vpc-07ae3320b4eee150d	172.31.0.0/20

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

The screenshot shows the "Create subnet" wizard step 1: VPC settings. It has two tabs: "VPC" (selected) and "Subnet settings". Under "VPC", the "VPC ID" dropdown is set to "vpc-07068a871e4f8932d (myVPC)". Under "Subnet settings", the "Subnet name" field contains "subnet1-vpc1". The "Availability Zone" dropdown is set to "Choose the zone in which your subnet will reside, or let Amazon choose one for you." The bottom of the screen shows standard AWS footer links.

The screenshot shows the AWS VPC Subnet creation interface. The subnet name is set to 'sunet1-vpc1'. The availability zone is 'Asia Pacific (Mumbai) / ap-south-1'. The IPv4 CIDR block is '10.0.0.0/25'. A tag 'Name' is added with the value 'sunet1-vpc1'. The 'Create subnet' button is highlighted.

Name	Value
Name	sunet1-vpc1

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

The screenshot shows the AWS VPC Subnets list. A success message indicates 1 subnet was created. The table lists four subnets, including the newly created one 'sunet1-vpc1'.

Name	Subnet ID	State	VPC	IPv4 CIDR
-	subnet-006c6ddfc85b63072	Available	vpc-07ae3320b4eee150d	172.31.32.0/20
-	subnet-002aae5ca688485e1	Available	vpc-07ae3320b4eee150d	172.31.16.0/20
<b>sunet1-vpc1</b>	<b>subnet-0f39f2af5c758714e</b>	<b>Available</b>	<b>vpc-07068a871e4f8932d   myV...</b>	<b>10.0.0.0/25</b>
-	subnet-0e99dc83ca0df711f	Available	vpc-07ae3320b4eee150d	172.31.0.0/20

- Create the Internal Gateway.

Internet gateways (1) Info

Name	Internet gateway ID	State	VPC ID	Owner
-	igw-09b76ffcc1e5b9f9d	Attached	vpc-07ae3320b4eee150d	469303665993

Select an internet gateway above

Create internet gateway

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

Create internet gateway Info

An internet gateway is a virtual router that connects a VPC to the internet. To create a new internet gateway specify the name for the gateway below.

**Internet gateway settings**

**Name tag**  
Creates a tag with a key of 'Name' and a value that you specify.

**Tags - optional**  
A tag is a label that you assign to an AWS resource. Each tag consists of a key and an optional value. You can use tags to search and filter your resources or track your AWS costs.

Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="igw1-vpc1"/>

**Add new tag**  
You can add 49 more tags.

Create internet gateway

The screenshot shows the AWS VPC dashboard. On the left sidebar, under 'Virtual private cloud', 'Internet gateways' is selected. In the main content area, a message at the top states: 'The following internet gateway was created: igw-0450e0461a1d75a6e - igw1-vpc1. You can now attach to a VPC to enable the VPC to communicate with the internet.' Below this, the 'igw-0450e0461a1d75a6e / igw1-vpc1' page is displayed. The 'Details' tab is selected, showing the Internet gateway ID (igw-0450e0461a1d75a6e), State (Detached), VPC ID (-), and Owner (469303665993). A 'Tags' section shows one tag: Name = igw1-vpc1. There is also a 'Manage tags' button. At the bottom of the page, there is a 'Actions' dropdown menu.

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

The screenshot shows the 'Attach to VPC' dialog box. The URL in the address bar is 'Attach to VPC (igw-0450e0461a1d75a6e)'. The dialog has a 'VPC' section with the sub-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 instruction: 'Attach the internet gateway to this VPC.' A search bar labeled 'Select a VPC' contains the text 'vpc-07068a871e4f8932d - myVPC'. At the bottom of the dialog are two buttons: 'Cancel' and 'Attach internet gateway'.

Internet gateway igw-0450e0461a1d75a6e successfully attached to vpc-07068a871e4f8932d

VPC > Internet gateways > igw-0450e0461a1d75a6e / igw1-vpc1

Details		Info					
Internet gateway ID igw-0450e0461a1d75a6e	State Attached	VPC ID vpc-07068a871e4f8932d   myVPC	Owner 469303665993				
Tags							
<input type="text" value="Search tags"/> <table border="1"> <thead> <tr> <th>Key</th> <th>Value</th> </tr> </thead> <tbody> <tr> <td>Name</td> <td>igw1-vpc1</td> </tr> </tbody> </table>				Key	Value	Name	igw1-vpc1
Key	Value						
Name	igw1-vpc1						

- 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.

Route tables (1/2) Info

Name	Route table ID	Explicit subnet associations	Main	VPC
-	rtb-08556b21a8e485273	-	Yes	vpc-07ae3320b4eee150d
<input checked="" type="checkbox"/> rt1-vpc1	rtb-0d158935e7111dc10	-	Yes	vpc-07068a871e4f8932d

rtb-0d158935e7111dc10 / rt1-vpc1

Details | Routes | Subnet associations | Edge associations | Route propagation | Tags

You can now check network connectivity with Reachability Analyzer

Run Reachability Analyzer

Details

Route table ID	Main	Explicit subnet associations	Edge associations
rtb-0d158935e7111dc10	<input checked="" type="checkbox"/> Yes	-	-

- Add Routes to Route Table.

The screenshot shows the 'Edit routes' page for a specific route table. The table has four columns: Destination, Target, Status, and Propagated. There are two entries:

Destination	Target	Status	Propagated
10.0.0.0/24	Q local	Active	No
Q 0.0.0.0/0	Q igw-0450e0461a1d75a6e	-	No

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

The screenshot shows the 'Route tables' page for a specific route table. A green success message box is displayed at the top: 'Updated routes for rtb-0d158935e7111dc10 / rt1-vpc1 successfully'. Below it, the route table details are shown:

**rtb-0d158935e7111dc10 / rt1-vpc1**

You can now check network connectivity with Reachability Analyzer. Actions: Run Reachability Analyzer.

**Details**

Route table ID rtb-0d158935e7111dc10	Main Yes	Explicit subnet associations -	Edge associations -
VPC vpc-07068a871e4f8932d   myVPC	Owner ID 469303665993		

**Routes (2)**

Destination	Target	Status	Propagated
10.0.0.0/24	Q local	Active	No
Q 0.0.0.0/0	Q igw-0450e0461a1d75a6e	-	No

Buttons at the bottom include 'Edit routes', '< 1 >', and a refresh icon.

## EC2 creation

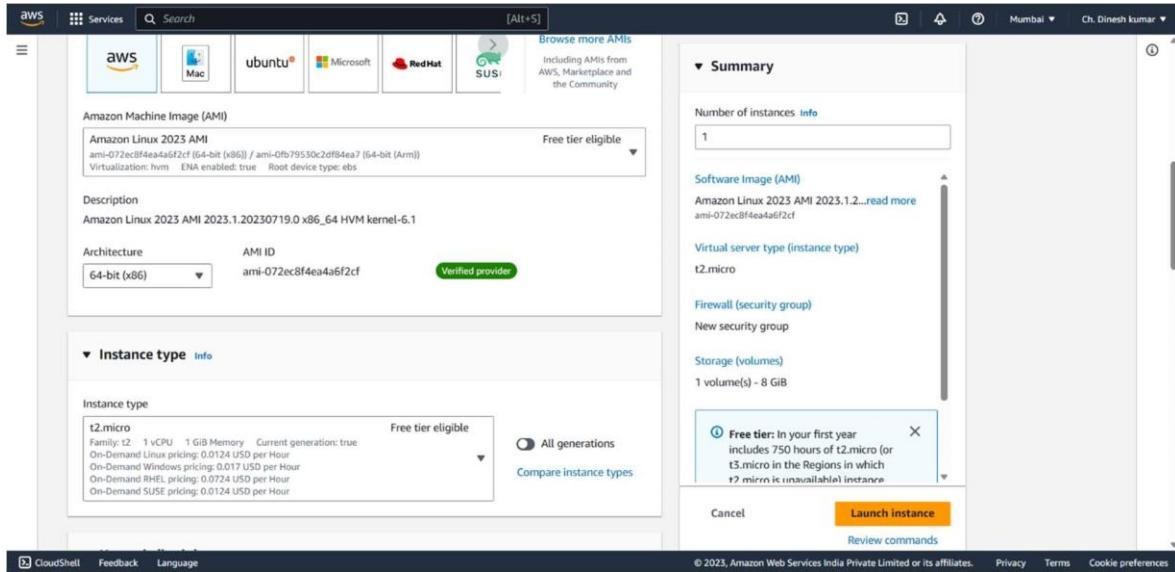
- Now again come back to the Console.
- Navigate to the EC2.
- Launch the EC2 Instance.

The screenshot shows the AWS CloudSearch interface. The search bar at the top contains the query 'ec2'. Below the search bar, there are two sections: 'Services' and 'Features'. The 'Services' section lists 'EC2' (Virtual Servers in the Cloud), 'EC2 Image Builder' (A managed service to automate build, customize and deploy OS images), 'Recycle Bin' (Protect resources from accidental deletion), and 'Amazon Inspector' (Continual vulnerability management at scale). The 'Features' section lists 'Dashboard' (EC2 feature) and 'AMIs'. On the right side of the screen, there is a table with one row, showing a single EC2 instance. The table includes columns for 'Creation time' and 'Last modified'. The 'Creation time' column shows '8 minutes ago'.

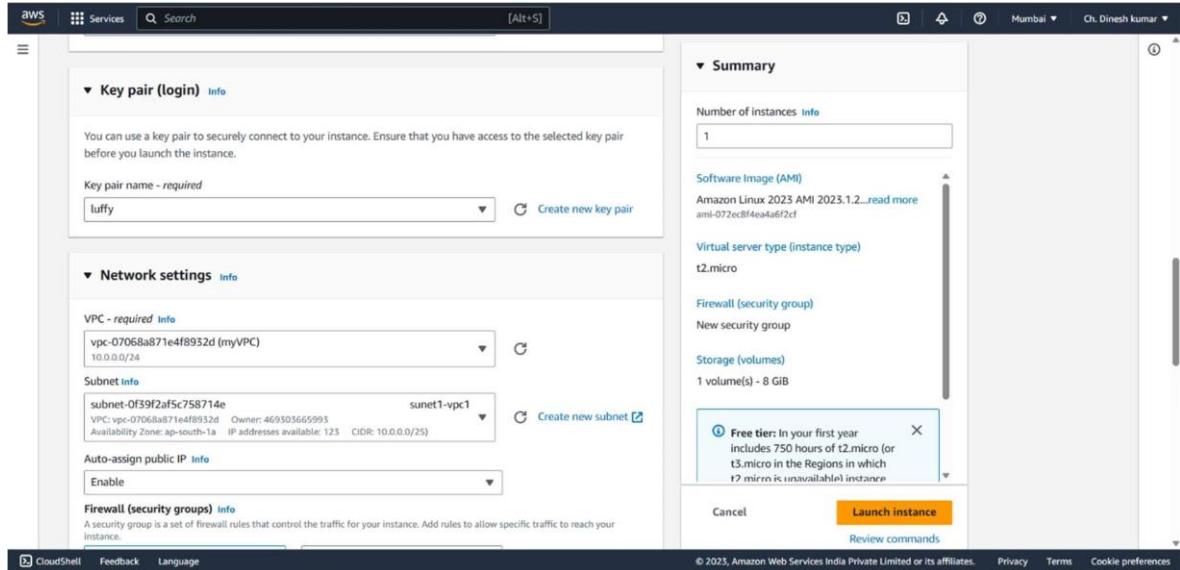
- Give name to the Instance.

The screenshot shows the 'Launch an instance' wizard in the AWS EC2 Instances section. The first step, 'Name and tags', has a 'Name' field containing 'vm1'. The second step, 'Application and OS Images (Amazon Machine Image)', has a search bar with the placeholder 'Search our full catalog including 1000s of application and OS images'. On the right side, the 'Summary' panel shows the configuration: 'Number of instances' set to 1, 'Software Image (AMI)' set to 'Amazon Linux 2023 AMI 2023.1.2...', 'Virtual server type (instance type)' set to 't2.micro', and 'Storage (volumes)' showing '1 volume(s) - 8 GiB'. A tooltip for the 'Free tier' is visible, stating: 'Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which t2.micro is unavailable) instance'. At the bottom, there are 'Cancel' and 'Launch instance' buttons, along with a 'Review commands' link.

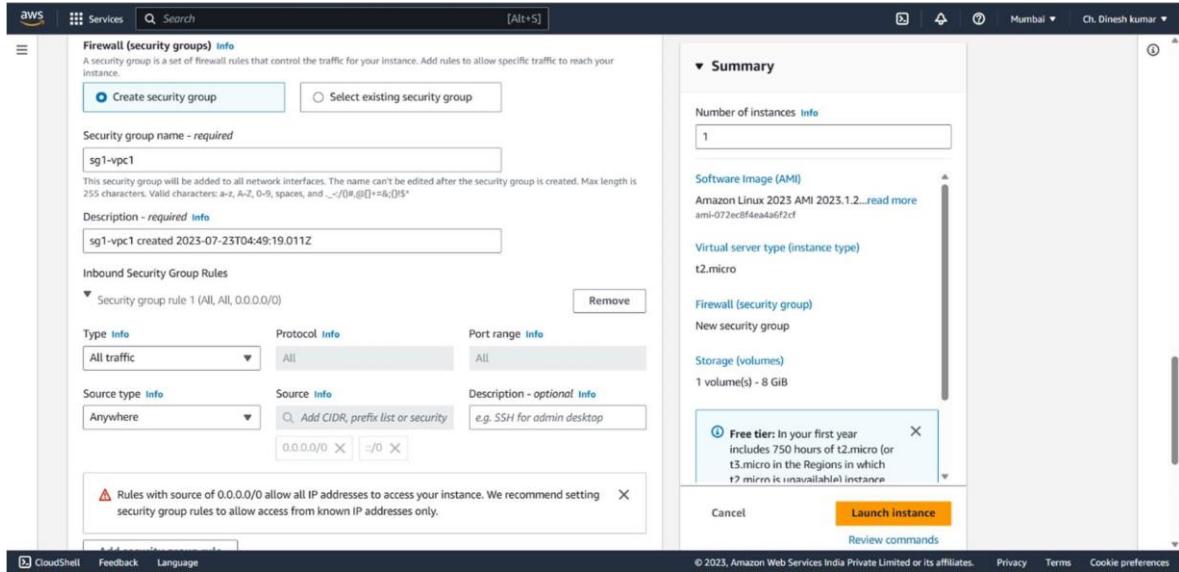
- We choose “Amazon Linux 2 AMI(HVM) – Kernel 5.10.SSD Volume Type in AMI.



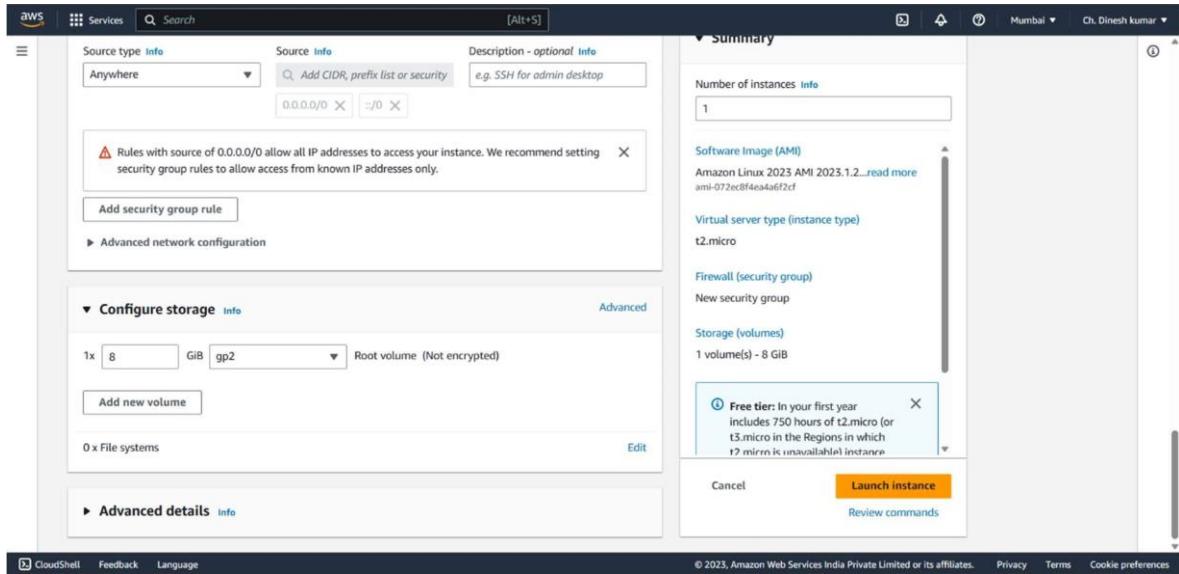
- Give created key pair, we given our key pair as “luffy”.



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



- Configure Storage add 1x 8 and GiB as gp2 and launch instance.

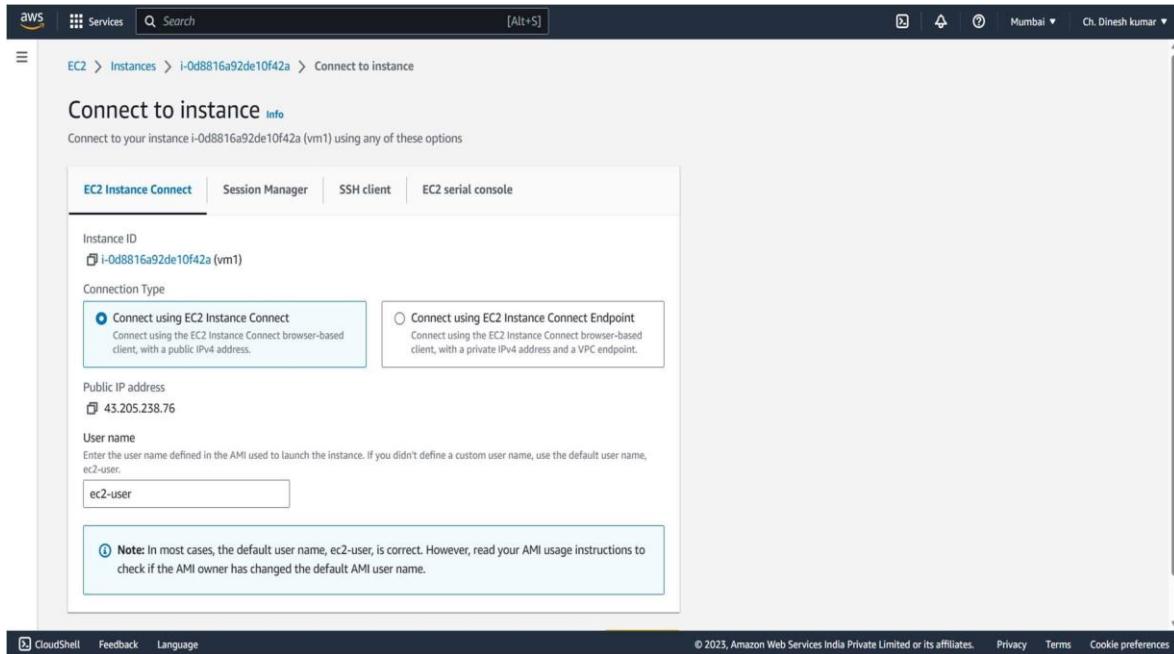


The screenshot shows the AWS EC2 Instances launch success page. At the top, there's a success message: "Successfully initiated launch of instance (i-0d8816a92de10f42a)". Below it is a "Launch log" button. The main area is titled "Next Steps" with a search bar. It includes four cards: "Create billing and free tier usage alerts", "Connect to your instance", "Connect an RDS database", and "Create EBS snapshot policy". Each card has a "Create" button and a "Learn more" link. At the bottom, there are links for CloudShell, Feedback, Language, and a footer with copyright information.

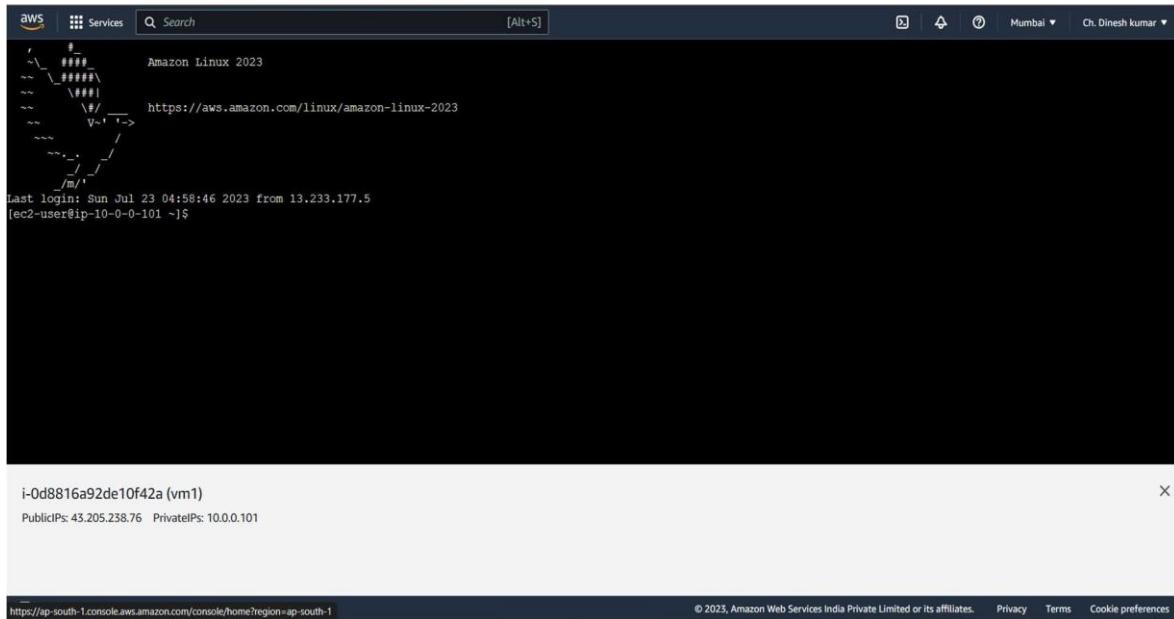
- Instance is successfully created.
- Then it will display as in the below picture.

The screenshot shows the AWS EC2 Instances list page. On the left, there's a sidebar with navigation links like EC2 Dashboard, EC2 Global View, Events, Instances (selected), Images, AMIs, AMI Catalog, Elastic Block Store, Volumes, Snapshots, and Lifecycle Manager. The main area shows a table with one row for "vm1". The table columns include Name, Instance ID, Instance state, Instance type, Status check, Alarm status, Availability Zone, and Public IP. The instance "vm1" is listed with the Instance ID "i-0d8816a92de10f42a", State "Running", Type "t2.micro", Status "Initializing", No alarms, and Availability Zone "ap-south-1". Below the table, there's a detailed view for "Instance: i-0d8816a92de10f42a (vm1)" with tabs for Details, Security, Networking, Storage, Status checks, Monitoring, and Tags. The Details tab shows an "Instance summary" with fields for Instance ID, Public IPv4 address, Private IPv4 addresses, IPv6 address, Instance state, Private IP DNS name, Public IP DNS name, Answer private resource DNS name, Instance type, and Elastic IP addresses.

- 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.



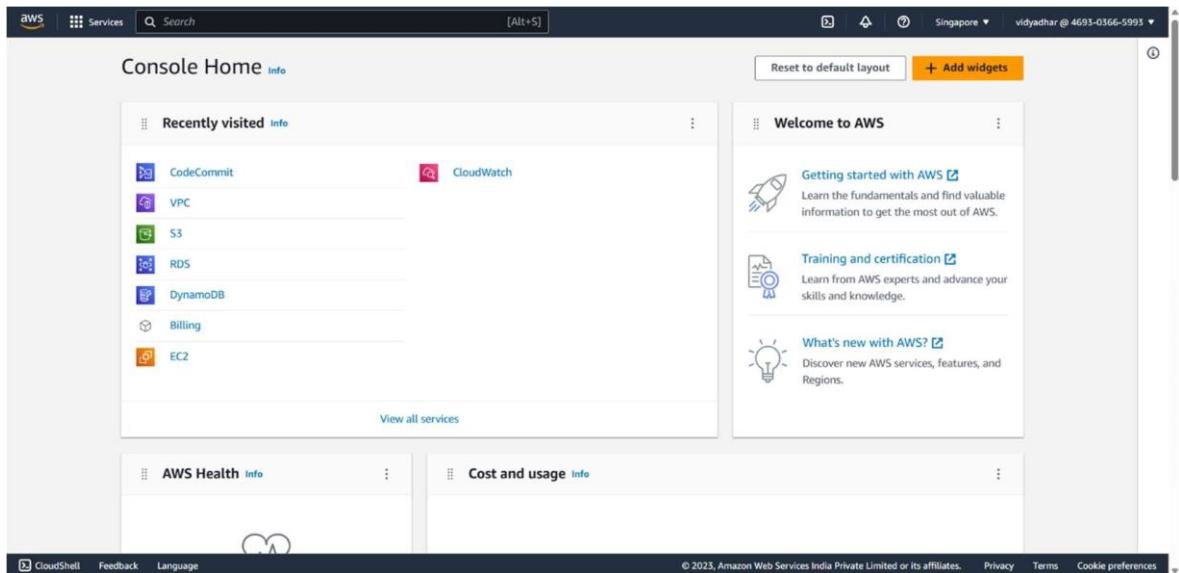
- Now again come back to the console and Navigate to the IAM.

The screenshot shows the AWS Identity and Access Management (IAM) service. On the left, the navigation pane includes 'Identity and Access Management (IAM)', 'Dashboard', 'Access management' (with 'User groups', 'Users', 'Roles', 'Policies', 'Identity providers', 'Account settings'), 'Access reports' (with 'Access analyzer', 'Archive rules', 'Analyzers', 'Settings', 'Credential report', 'Organization activity'), and 'Service control policies (SCPs)'. At the bottom of the sidebar are 'CloudShell', 'Feedback', and 'Language' buttons. The main content area is titled 'vidyadhar' and shows the 'Summary' tab. It displays the ARN (arn:aws:iam::469303665993:user/vidyadhar), Console access status (Disabled), and two Access keys (key 1: Not enabled, key 2: Not enabled). Below the summary is a 'Permissions' tab showing one policy attached: 'AWSCodeCommitFullAccess' (Type: AWS managed, Attached via Group: ToxicDevOps). The footer contains copyright information and links for 'Privacy', 'Terms', and 'Cookie preferences'.

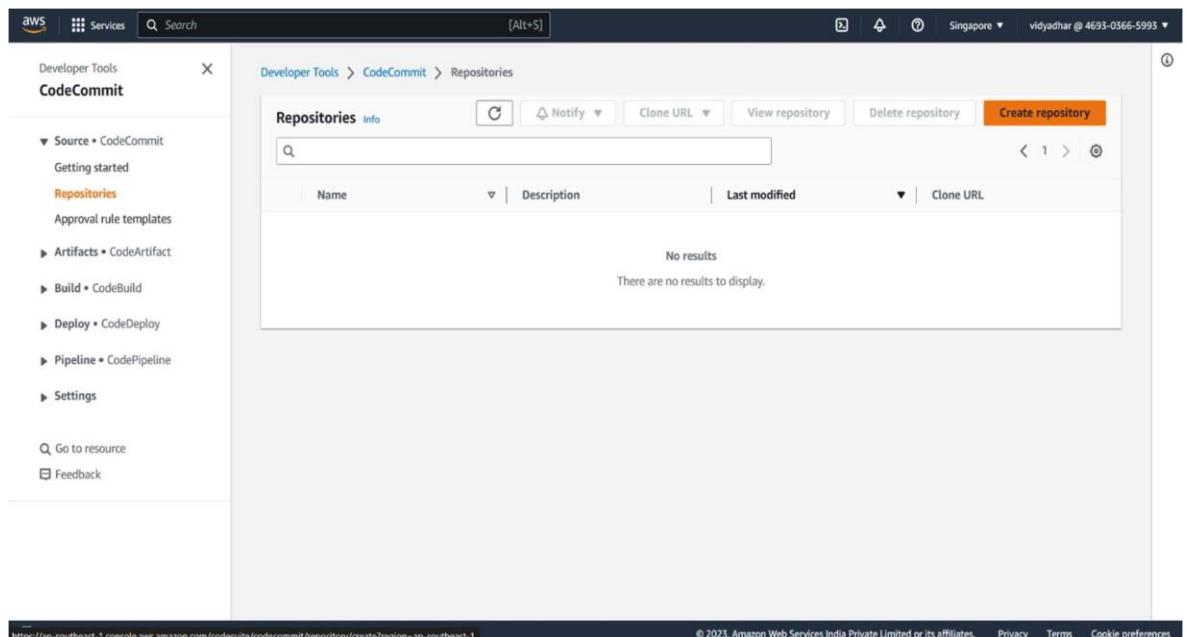
- Now go to the Security Credentials.
- Go to the console settings.
- It will provide the Console password and Download the Credentials.

This screenshot shows the same IAM user details page as above, but with a green banner at the top stating 'Console access updated.' The 'Console password' section is highlighted, showing a success message: 'You have successfully updated the user's new password. This is the only time you can view this password. After you close this window, if the password is lost, you must create a new one.' It also lists the 'Console sign-in URL' (https://469303665993.sigin.aws.amazon.com/console), 'User name' (vidyadhar), and 'Console password' (redacted). Buttons for 'Download .csv file' and 'Close' are visible. The right side of the screen shows the 'Access key 1' and 'Access key 2' sections, and a 'Manage console access' button. The bottom of the page includes a 'Multi-factor auth' section and a 'Created on' timestamp (10:33 GMT+5:30).

- 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.

Create repository

Create a secure repository to store and share your code. Begin by typing a repository name and a description for your repository. Repository names are included in the URLs for that repository.

**Repository settings**

Repository name: Toxic

Description - optional: Repository for admin

Tags:

Key	Value - optional
Name	Toxic

Add tag

Enable Amazon CodeGuru Reviewer for Java and Python - optional

CloudShell Feedback Language

© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

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

Success

Repository successfully created

Developer Tools > CodeCommit > Repositories > Toxic

Toxic

Clone URL

**Connection steps**

HTTPS | SSH | HTTPS (GRC)

**Step 1: Prerequisites**  
You must use a Git client that supports Git version 1.7.9 or later to connect to an AWS CodeCommit repository. If you do not have a Git client, you can install one from Git downloads. [View Git downloads page](#)

You must have an AWS CodeCommit managed policy attached to your IAM user, belong to a CodeStar project team, or have the equivalent permissions. Learn how to create and configure an IAM user for accessing AWS CodeCommit. [Learn how to add team members to an AWS CodeStar Project](#)

**Step 2: Git credentials**  
Create Git credentials for your IAM user, if you do not already have them. Download the credentials and save them in a secure location. [Generate Git Credentials](#)

**Step 3: Clone the repository**  
Clone your repository to your local computer and start working on code. Run the following command:

CloudShell Feedback Language

© 2023, Amazon Web Services India Private Limited or its affiliates. Privacy Terms Cookie preferences

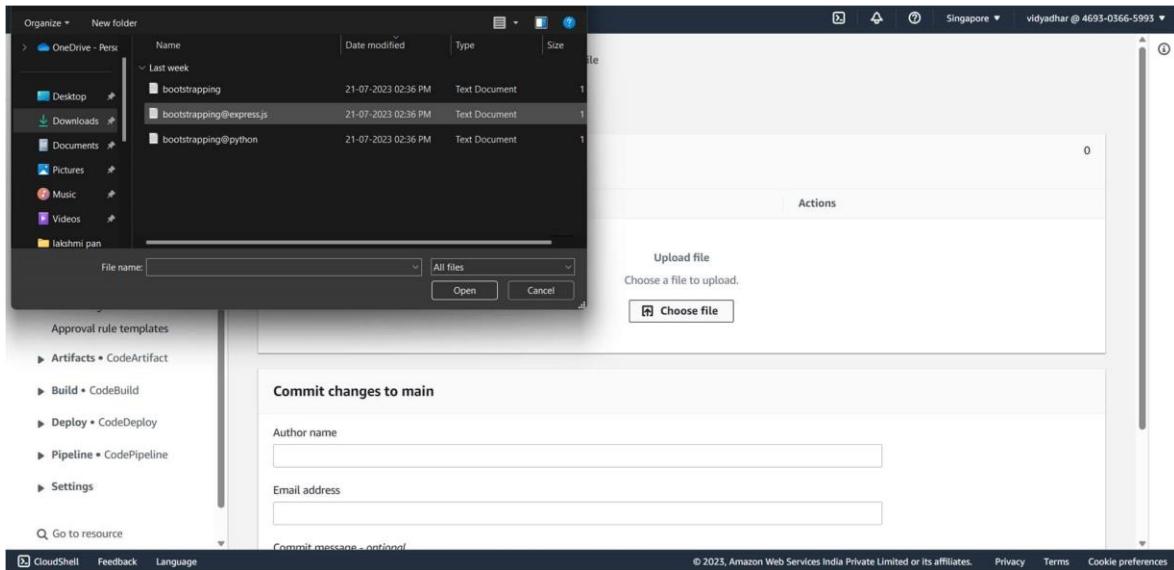
- After selecting the create section.
- Now connection steps are required.
- Scroll down and there will be create or upload File.

The screenshot shows the AWS CodeCommit interface. A green success banner at the top right states "Repository successfully created". Below it, a "Step 3: Clone the repository" section provides instructions to clone the repository using the command "git clone https://git-codecommit.ap-southeast-1.amazonaws.com/v1/repos/Toxic". A "Copy" button is available to copy this URL. An "Additional details" section links to documentation. On the left sidebar, under "Code", there are options for Pull requests, Commits, Branches, Git tags, and Settings. The main content area shows a repository named "Toxic" with an "Empty repository" message and a "Create file" button.

- Then choose upload file.

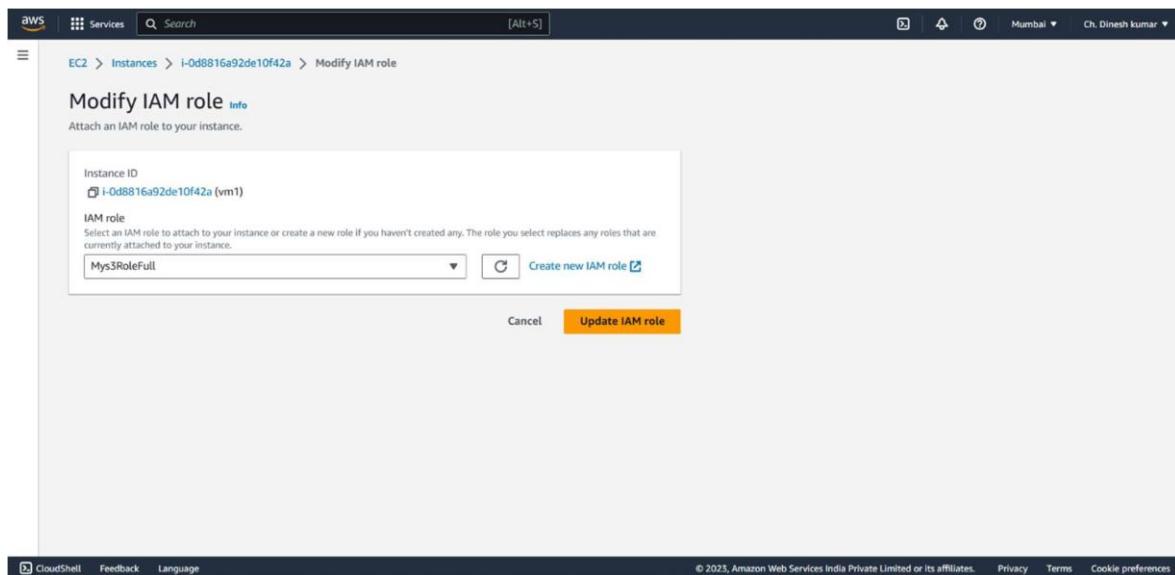
- Then the file will be browsed from our local file manager.

The screenshot shows the AWS CodeCommit interface for uploading files. The left sidebar shows the navigation path: Developer Tools > CodeCommit > Repositories > Toxic > File. The main area is titled "Upload a file" and shows a table for uploaded files with one entry: "Toxic" (Info), Size 0, Actions. Below this is a "Commit changes to main" section with fields for Author name and Email address, both currently empty. A "Choose file" button is available to select a file for upload. At the bottom, there is a "Commit message - optional" field.



- 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.

The screenshot shows the AWS CodeCommit interface. On the left, there's a navigation sidebar with 'Developer Tools' and 'CodeCommit' selected. Under 'CodeCommit', there are sections for 'Source', 'Artifacts', 'Build', 'Deploy', 'Pipeline', and 'Settings'. Below these are links for 'Go to resource' and 'Feedback'. The main content area is titled 'Repositories' and shows a single repository named 'Toxic'. The repository details are: Name: Toxic, Description: Repository for admin, Last modified: 3 minutes ago. There are Clone URL buttons for HTTPS, SSH, and HTTPS (GRC).

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

The screenshot shows an AWS CloudShell terminal window. The user is on an Amazon Linux 2023 instance. They run the command `sudo yum install git`. The output shows the package being installed along with its dependencies. The transaction summary indicates 8 packages were installed, totaling 7.1 M download size and 34 M installed size.

```

Amazon Linux 2023
https://aws.amazon.com/linux/amazon-linux-2023

Last login: Sun Jul 23 05:08:30 2023 from 13.233.177.3
[ec2-user@ip-10-0-0-101 ~]$ sudo su
[root@ip-10-0-0-101 ec2-user]# yum install git
Last metadata expiration check: 0:24:08 ago on Sun Jul 23 04:56:44 2023.
Dependencies resolved.

Transaction Summary
Install 8 Packages

Total download size: 7.1 M
Installed size: 34 M

```

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

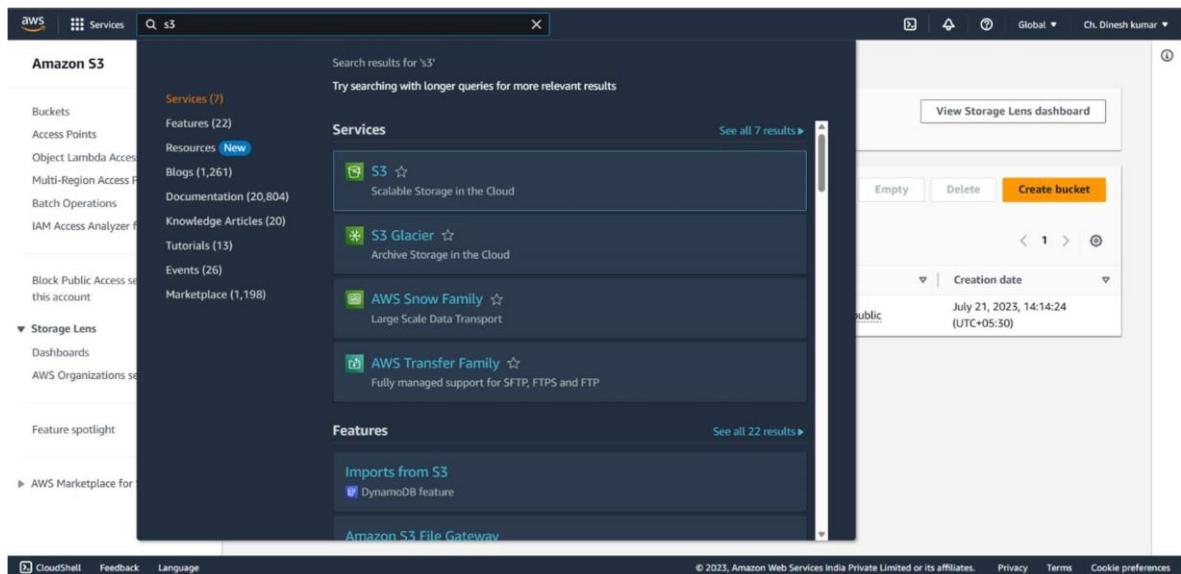
```

aws | Services | Search | [Alt+S]
hint: is subject to change. To configure the initial branch name to use in all
hint: of your new repositories, which will suppress this warning, call:
hint:
hint:   git config --global init.defaultBranch <name>
hint:
hint: Names commonly chosen instead of 'master' are 'main', 'trunk' and
hint: 'development'. The just-created branch can be renamed via this command:
hint:
hint:   git branch -m <name>
Initialized empty Git repository in /home/ec2-user/vidya/.git/
[root@ip-10-0-0-101 vidya]# git remote add origin https://git-codecommit.ap-southeast-1.amazonaws.com/v1/repos/Toxic
[root@ip-10-0-0-101 vidya]# git pull origin main
Username for 'https://git-codecommit.ap-southeast-1.amazonaws.com': vidyadhar-at-469303665993
Password for 'https://vidyadhar-at-469303665993@git-codecommit.ap-southeast-1.amazonaws.com':
remote: Counting objects: 3, done.
Unpacking objects: 100% (3/3), 407 bytes | 407.00 KiB/s, done.
From https://git-codecommit.ap-southeast-1.amazonaws.com/v1/repos/Toxic
 * [branch]      main    -> refs/heads/main
 * [new branch]   main    -> origin/main
[root@ip-10-0-0-101 vidya]# ls
bootstrapping.txt
[root@ip-10-0-0-101 vidya]# aws s3 cp bootstrapping.txt s3://dinbucl
upload: ./bootstrapping.txt to s3://dinbucl/bootstrapping.txt
[root@ip-10-0-0-101 vidya]# history
 1 ping google.com
 2 yum install git
 3 mkdir vidya
 4 cd vidya
 5 git init
 6 git remote add origin https://git-codecommit.ap-southeast-1.amazonaws.com/v1/repos/Toxic
 7 git pull origin main
 8 ls
 9 aws s3 cp bootstrapping.txt s3://dinbucl
10 history
[root@ip-10-0-0-101 vidya]#

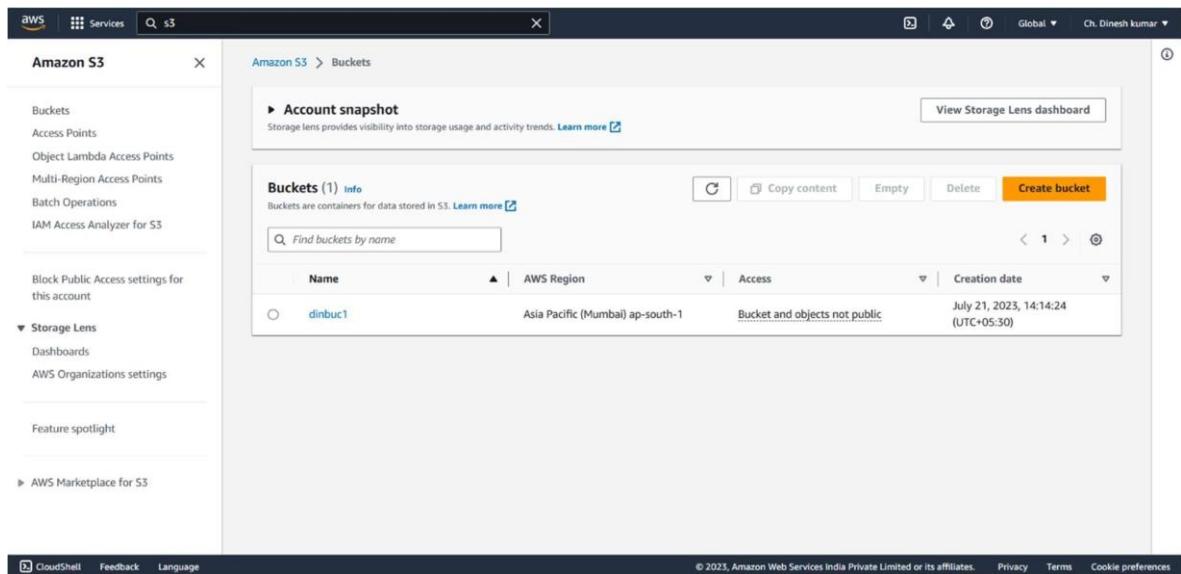
```

CloudShell   Feedback   Language   © 2023, Amazon Web Services India Private Limited or its affiliates.   Privacy   Terms   Cookie preferences

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



- Create the bucket in the S3.



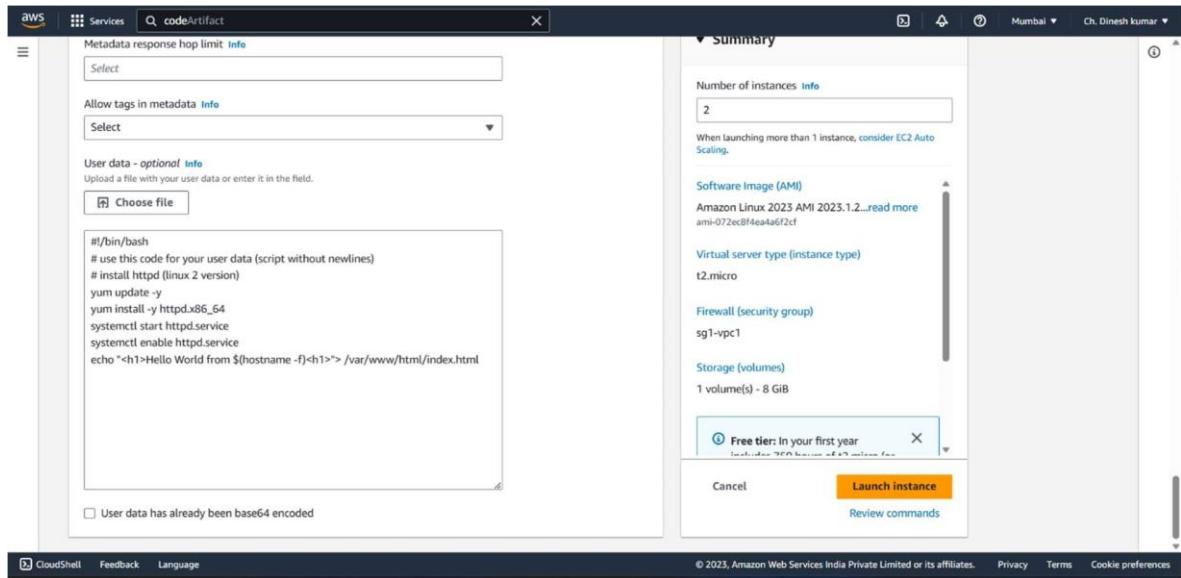
- In the bucket add object as the file that is taken from the code commit.

The screenshot shows the AWS S3 console interface. On the left, there's a sidebar with various options like Buckets, Access Points, Object Lambda Access Points, Multi-Region Access Points, Batch Operations, IAM Access Analyzer for S3, Storage Lens, Dashboards, AWS Organizations settings, Feature spotlight, and AWS Marketplace for S3. The main area shows a bucket named 'dinbuc1'. Below it, under 'Objects (1)', there's a table with one item: 'bootstrapping.txt'. The table includes columns for Name, Type, Last modified, Size, and Storage class. The file was last modified on July 23, 2023, at 10:56:20 (UTC+05:30), is a txt file, has a size of 289.0 B, and is stored in the Standard storage class.

- Now Download that file to the local machine.

The screenshot shows the AWS EC2 console. The user is launching a new instance. In the 'Name and tags' section, the name 'vm2' is entered. Under 'Application and OS Images (Amazon Machine Image)', the user has selected 'Amazon Linux 2023 AMI 2023.1.2...'. The 'Virtual server type (instance type)' is set to 't2.micro'. A message box indicates a 'Free tier: In your first year' offer. At the bottom right, there are 'Launch instance' and 'Review commands' buttons.

- 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

Screenshot of the AWS EC2 Load Balancers page. The left sidebar shows navigation options like EC2 Dashboard, Instances, Images, and Elastic Block Store. The main content area displays a table with columns for Name, DNS name, State, VPC ID, Availability Zones, Type, and Date created. A search bar at the top allows filtering by attribute or tag. A modal window titled "0 load balancers selected" with the message "Select a load balancer above." is open.

## Load Balancer

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

Screenshot of the AWS Classic Load Balancer creation wizard. The first step, "Classic Load Balancer - previous generation", shows a diagram where traffic from a client passes through a "CLB" (Classic Load Balancer) to four EC2 instances. The CLB supports protocols like HTTP, HTTPS, TCP, and SSL. A "Create" button is available for this step.

- 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 Inservice or not.

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

**EC2 | Load balancers | myELB**

**Description** Instances Health check Listeners Monitoring Tags Migration

**Basic Configuration**

Name	myELB	Creation time	July 23, 2023 at 11:12:29 AM UTC+5:30
* DNS name	myELB-27632040.ap-south-1.elb.amazonaws.com (A Record)	Hosted zone	ZPB7RAFLXTNZK
Type	Classic (Migrate Now)	Status	2 of 2 instances in service
Scheme	Internet-facing	VPC	vpc-07068a871e4f8932d
Availability Zones	subnet-0f39f2af5c758714e - ap-south-1a		

**Port Configuration**

Port Configuration	80 (HTTP) forwarding to 80 (HTTP) Stickiness: Disabled
	<a href="#">Edit stickiness</a>

**Security**

Source Security Group	sg-0332afa4699039d2f, sg1-vpc1 • sg1-vpc1 created 2023-07-23T04:49:19.011Z
	sg-0b5b14b054f05aba0, default • default VPC security group

- 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.



The screenshot shows a web browser window with the following details:

- Address Bar:** Not secure | myelb-276323040.ap-south-1.elb.amazonaws.com
- Title Bar:** Registration Page
- Form Fields:**
  - Firstname: [Input Field]
  - Middlename: [Input Field]
  - Lastname: [Input Field]
  - Course : [Course dropdown menu]
  - Gender :
    - Male
    - Female
    - Other
  - Phone : [+91] [Input Field]
  - Address:  
[Large text area input field]
  - Email: [Input Field]
  - Password: [Input Field]
  - Re-type password: [Input Field]
- Buttons:**
  - [Submit] button

## 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.