

A PROJECT REPORT ON AWS ELASTIC LOAD BALANCING INSIDE OF A VIRTUAL PRIVATE CLOUD

Submitted by

DISHA GHOSH

**FUTURE INSTITUTE OF ENGINEERING AND
MANAGEMENT**

Under the guidance of

Mr. AMIT JAISWAL

Assistant Professor

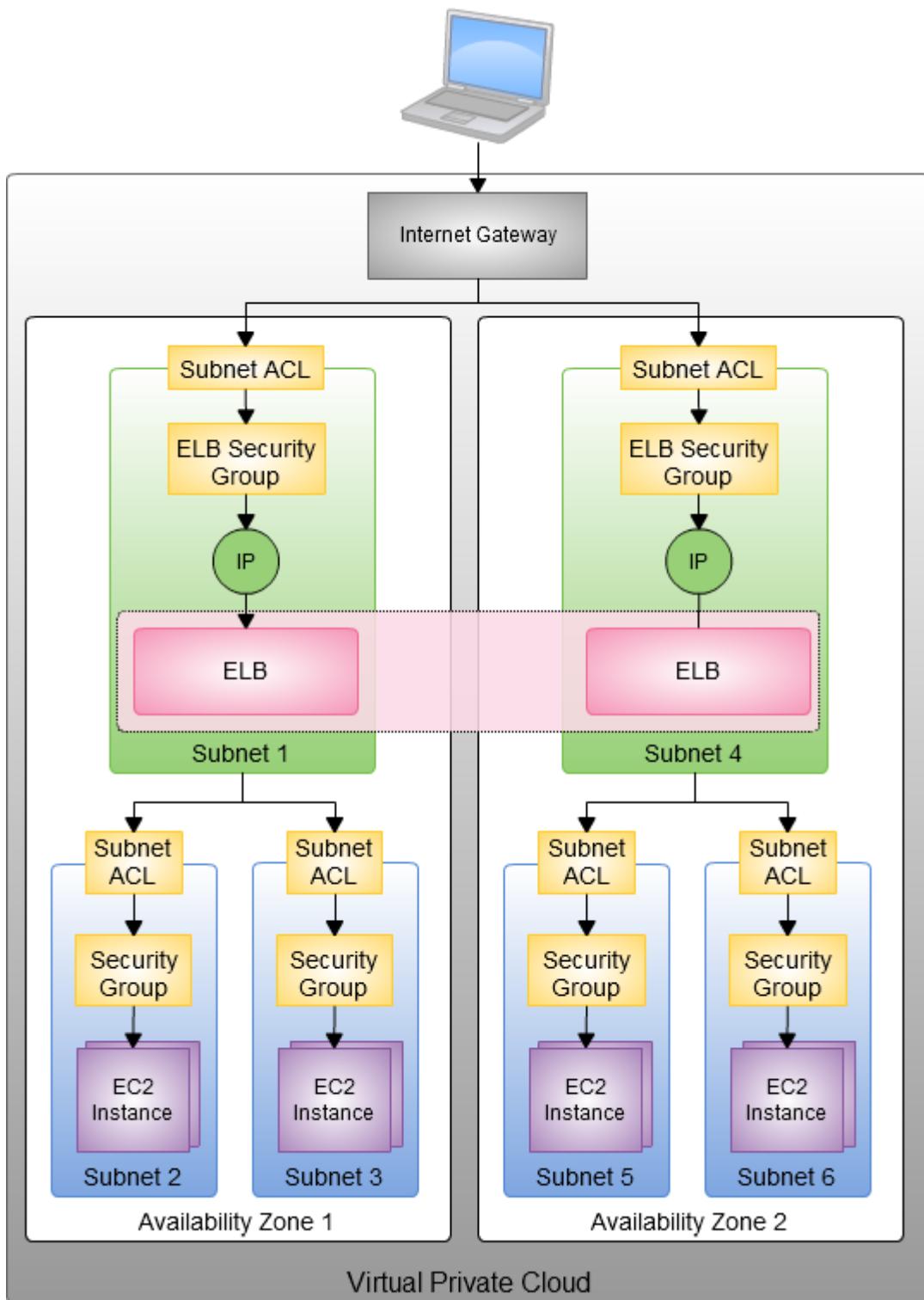
Dept. of Cloud System Administration

MTA, India



The popular AWS Elastic Load Balancing Feature is now available within the Virtual Private Cloud (VPC). Features such as SSL termination, health checks, sticky sessions and CloudWatch monitoring can be configured from the AWS Management Console, the command line, or through the Elastic Load Balancing APIs. When you provision an Elastic Load Balancer for your VPC, you can assign security groups to it. You can place ELBs into VPC subnets, and you can also use subnet ACLs (Access Control Lists). The EC2 instances that you register with the Elastic Load Balancer do not need to have public IP addresses. The combination of the Virtual Private Cloud, subnets, security groups, and access control lists gives you precise, fine-grained control over access to your Load Balancers and to the EC2 instances behind them and allows you to create a private load balancer.

Here's how it all fits together:



INTRODUCTION TO ELASTIC LOAD BALANCING AND VPC

Elastic Load Balancing- Elastic Load Balancing automatically distributes our incoming traffic across multiple targets, such as EC2 instances, containers, and IP addresses, in one or more Availability Zones. It monitors the health of its registered targets, and routes traffic only to the healthy targets. Elastic Load Balancing scales our load balancer capacity automatically in response to changes in incoming traffic. Elastic Load Balancing supports the following load balancers: **Application Load Balancers, Network Load Balancers, Gateway Load Balancers, and Classic Load Balancers.**

AWS offers a variety of different infrastructure and platform services. Virtual Private Cloud is among one of them.

Amazon Virtual Private Cloud (Amazon VPC)- Amazon Virtual Private Cloud (Amazon VPC) enables us to define a virtual network in our own logically isolated area within the AWS cloud, known as a virtual private cloud or VPC. We can create AWS resources, such as Amazon EC2 instances, into the subnets of your VPC. VPC closely resembles a traditional network that we might operate in our own data center, with the benefits of using scalable infrastructure from AWS. We can configure our VPC; we can select its IP address range, create subnets, and configure route tables, network gateways, and security settings. We can connect instances in our VPC to the internet or to our own data center.

➤ **Features:-**

The following features help us configure a VPC to provide the connectivity that our applications need:

- **Virtual private clouds (VPC)**

A VPC is a virtual network that closely resembles a traditional network that we would operate in our own data center. After we create a VPC, we can add subnets.

- **Subnets**

A subnet is a range of IP addresses in our VPC. A subnet must reside in a single Availability Zone. After we add subnets, we can deploy AWS resources in our VPC.

- **IP addressing**

We can assign IP addresses, both IPv4 and IPv6, to our VPCs and subnets. We can also bring our public IPv4 addresses and IPv6 GUA addresses to AWS and allocate them to resources in our VPC, such as EC2 instances, NAT gateways and Network Load Balancers.

- **Routing**

Using route tables to determine where network traffic from our subnet or gateway is directed.

- **Gateways and endpoints**

A gateway connects our VPC to another network. For example, we use an internet gateway to connect our VPC to the internet. We can use a VPC endpoint to connect to AWS services privately, without the use of an internet gateway or NAT device.

- **Peering connections**

Use a VPC peering connection to route traffic between the resources in two VPCs.

- **Traffic Mirroring**

Copy network traffic from network interfaces and send it to security and monitoring appliances for deep packet inspection.

- **Transit gateways**

Use a transit gateway, which acts as a central hub, to route traffic between our VPCs, VPN connections, and AWS Direct Connect connections.

- **VPC Flow Logs**

A flow log captures information about the IP traffic going to and from network interfaces in our VPC.

- **VPN connections**

We can connect our VPCs to on-premises networks using AWS Virtual Private Network (AWS VPN).

BENEFITS OF USING AWS LOAD BALANCER:

Elastic Load Balancing automatically distributes your incoming traffic across multiple targets, such as EC2 instances, containers, and IP addresses, in one or more Availability Zones. It monitors the health of its registered targets, and routes traffic only to the healthy targets. Elastic Load Balancing scales your load balancer as your incoming traffic changes over time. It can automatically scale to the vast majority of workloads.

AWS load balancing services play a pivotal role in ensuring the high availability and optimal performance of web applications.

By leveraging the robust features, security measures and scalability provided by the AWS organizations can achieve greater operational efficiency and deliver exceptional user experiences in the ever-evolving digital landscape.

ACCESSING ELASTIC LOAD BALANCING:

We can create, access, and manage our load balancers using any of the following interfaces:

- **AWS Management Console**— Provides a web interface that we can use to access Elastic Load Balancing.
- **AWS Command Line Interface (AWS CLI)** — Provides commands for a broad set of AWS services, including Elastic Load Balancing. The AWS CLI is supported on Windows, macOS, and Linux.
- **AWS SDKs** — Provide language-specific APIs and take care of many of the connection details, such as calculating signatures, handling request retries, and error handling.
- **Query API**— Provides low-level API actions that we call using HTTPS requests. Using the Query API is the most direct way to access Elastic Load Balancing.

BENEFITS OF USING VIRTUAL PRIVATE CLOUD:

Virtual private clouds offer a number of benefits for your business.

- **Security**—Since private clouds are built around one single business, the data storage, network, and hardware can be specifically designed around the security needs of your business. It cannot be accessed by any other client or company in the same building or data center.

- **Compliance**—Compliance is much easier to accomplish because each network and storage configuration is based around one single client. This makes it much easier to monitor their security and compliance needs.
- **Customization**—Network performance, storage performance, and hardware performance can be specifically customized into the client's private cloud.

ACCESSING AMAZON VPC:

We can create, access, and manage our VPCs using any of the following interfaces:

- AWS Management Console — Provides a web interface that we can use to access our VPCs.
- AWS Command Line Interface (AWS CLI) — Provides commands for a broad set of AWS services, including Amazon VPC, and is supported on Windows, Mac, and Linux.
- AWS SDKs — Provides language-specific APIs and takes care of many of the connection details, such as calculating signatures, handling request retries, and error handling.
- Query API — Provides low-level API actions that we call using HTTPS requests.

AMAZON VPC CONCEPTS:

Amazon VPC is the networking layer for Amazon EC2The following are the key concepts for VPCs:

- Virtual private cloud (VPC) — A virtual network dedicated to our AWS account.
- Subnet — A range of IP addresses in our VPC.
- Route table — A set of rules, called routes, that are used to determine where network traffic is directed.
- Internet gateway — A gateway that we attach to our VPC to enable communication between resources in our VPC and the internet.
- VPC endpoint — Enables to privately connect our VPC to supported AWS services and VPC endpoint services powered by Private Link without requiring an internet gateway, NAT device, VPN connection or AWS Direct Connect connection. Instances in our VPC do not require public IP addresses to communicate with resources in the service. Traffic between our VPC and the other service does not leave the Amazon network.

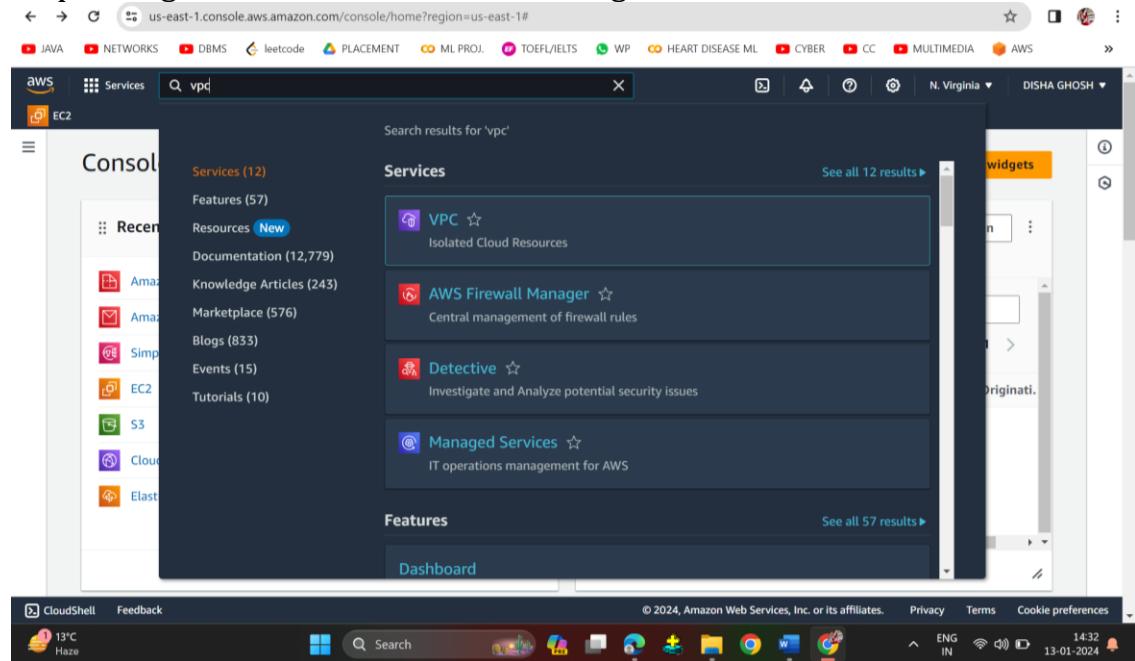
PREREQUISITES OF THE PROJECT:

The prerequisites for learning this project are as following:-

- Having basic knowledge of operating systems like Windows OS, Linux etc.
- As Visualization play a major role in AWS we need to have the understanding of it.
- Networking is an essential skill as all operations on cloud platform involves it.

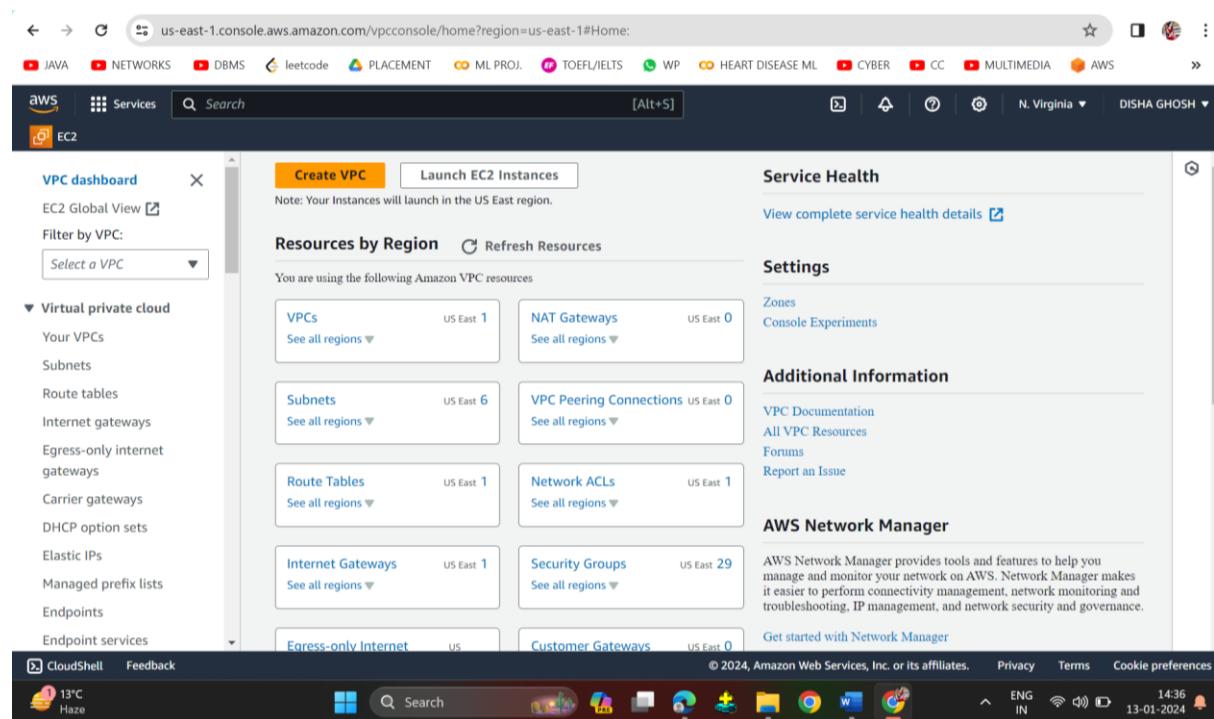
Setting up an Elastic Load Balancer inside a Virtual Private Cloud

Step-1: Login into AWS Console and navigate to the VPC as shown below:



The screenshot shows the AWS console interface. The search bar at the top contains the query 'vpc'. Below the search bar, the 'Services' section is expanded, showing 12 results. The first result is 'VPC' with the description 'Isolated Cloud Resources'. Other services listed include AWS Firewall Manager, Detective, and Managed Services. The 'Features' section shows 57 results, and a 'Dashboard' button is also visible. The bottom of the screen shows the standard AWS navigation bar with links for CloudShell, Feedback, and various AWS services like Java, Networks, DBMS, etc. The date and time are also displayed.

Step-2 After navigating to the AWS VPC click on create VPC.



The screenshot shows the AWS VPC console. On the left, there is a sidebar with 'VPC dashboard' and 'Virtual private cloud' sections. The 'Create VPC' button is highlighted in orange. The main content area displays 'Resources by Region' with sections for VPCs, Subnets, Route Tables, Internet Gateways, Egress-only Internet gateways, Carrier Gateways, DHCP option sets, Elastic IPs, Managed prefix lists, Endpoints, and Endpoint services. Each resource type has a 'See all regions' link. To the right, there are sections for 'Service Health', 'Settings' (with links to Zones and Console Experiments), 'Additional Information' (with links to VPC Documentation, All VPC Resources, Forums, and Report an Issue), and 'AWS Network Manager' (with a detailed description and 'Get started with Network Manager' link). The bottom of the screen shows the standard AWS navigation bar.

Step-3 Select VPC only, name tag will be: disha-vpc, IPv4 CIDR- 192.168.0.0/16 and click on create VPC.

The screenshot shows the 'Create VPC' configuration page. Under 'Resources to create', 'VPC only' is selected. The 'Name tag - optional' field contains 'disha-vpc'. The 'IPv4 CIDR block' section shows '192.168.0.0/16' entered. The status bar at the bottom indicates 'CloudShell Feedback' and the date '14-01-2024'.

The screenshot shows the 'Create VPC' configuration page. Under 'IPv6 CIDR block', 'No IPv6 CIDR block' is selected. The 'Tenancy' dropdown is set to 'Default'. In the 'Tags' section, a single tag 'Name: disha-vpc' is added. The status bar at the bottom indicates 'CloudShell Feedback' and the date '14-01-2024'.

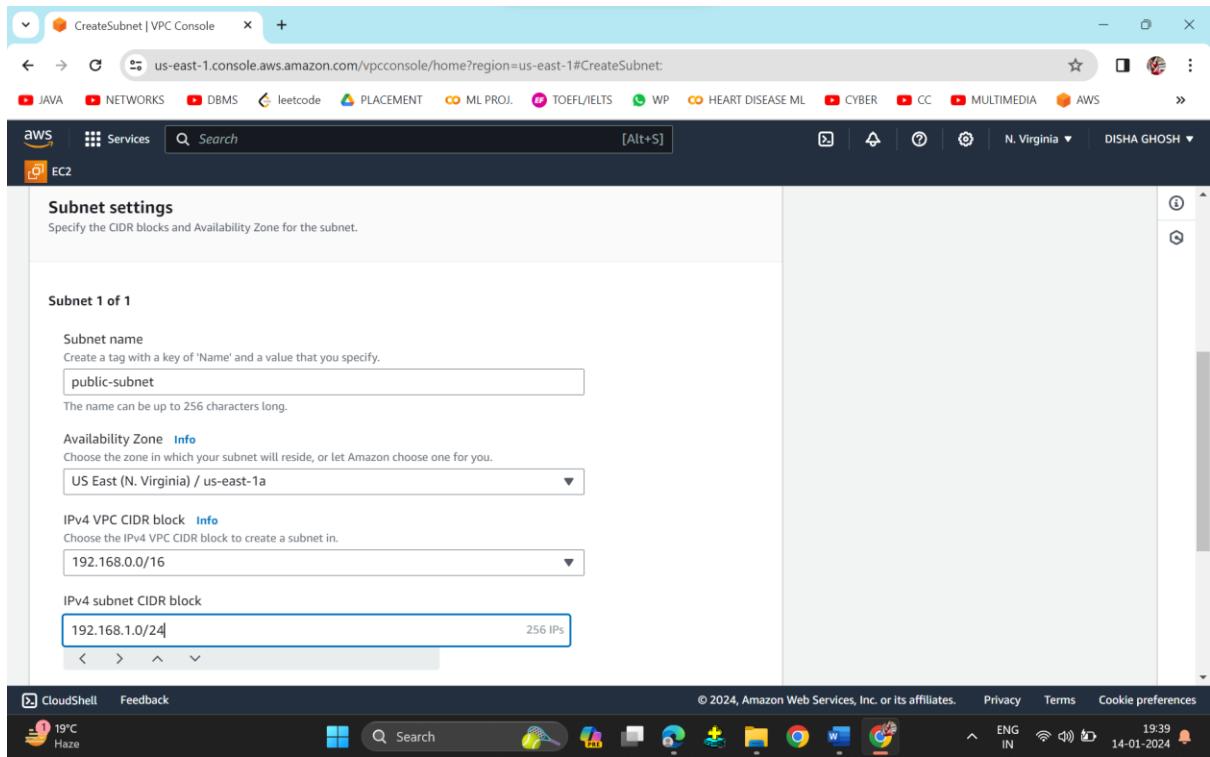
Step-4 Click on Subnets and create subnet.

The screenshot shows the AWS VPC Subnets page. On the left, there's a sidebar with options like 'Virtual private cloud' and 'Subnets'. The main area displays a table titled 'Subnets (6) Info' with columns for Name, Subnet ID, State, and VPC. Each row lists a subnet name, its unique ID, its availability status, and the specific VPC it belongs to. A prominent orange button at the top right says 'Create subnet'.

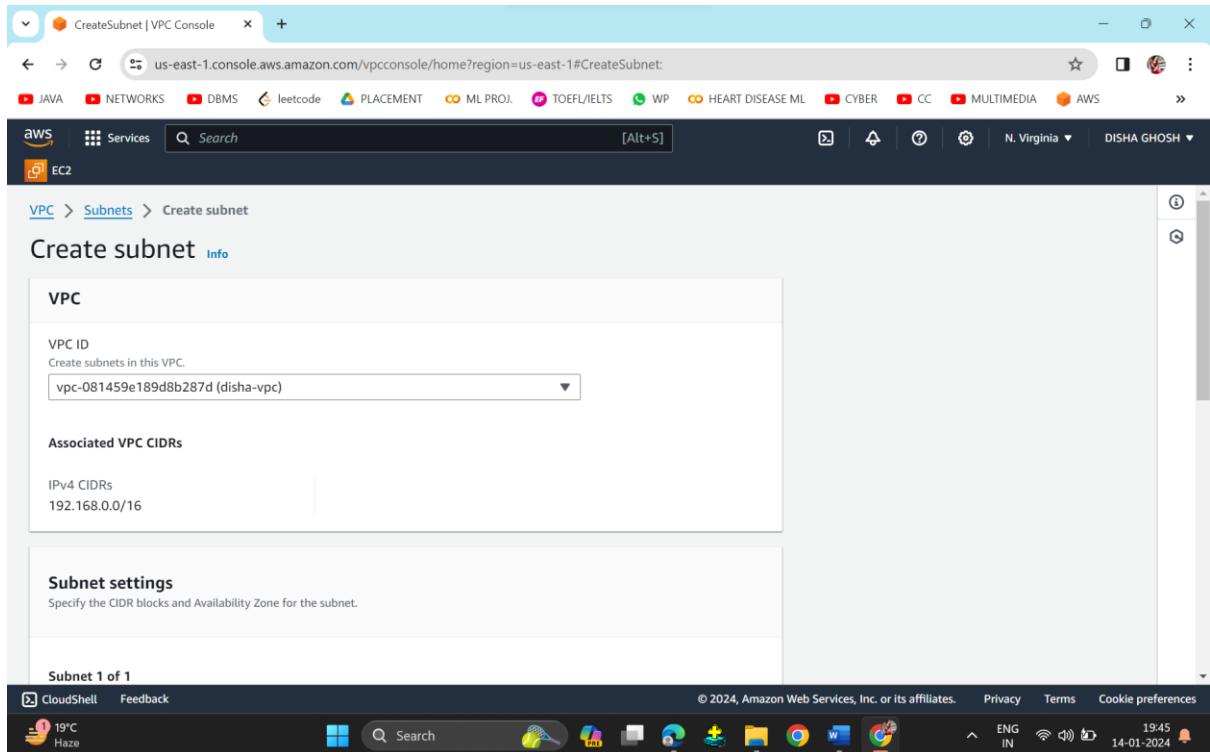
Name	Subnet ID	State	VPC
-	subnet-00a1b0a6ff68ba214	Available	vpc-054e78d65b6391585
-	subnet-0590139b6a66f3674	Available	vpc-054e78d65b6391585
-	subnet-0a417060d45acbd5	Available	vpc-054e78d65b6391585
-	subnet-02f1e57d54e8a9717	Available	vpc-054e78d65b6391585
-	subnet-02a4f734ba9f7db53	Available	vpc-054e78d65b6391585

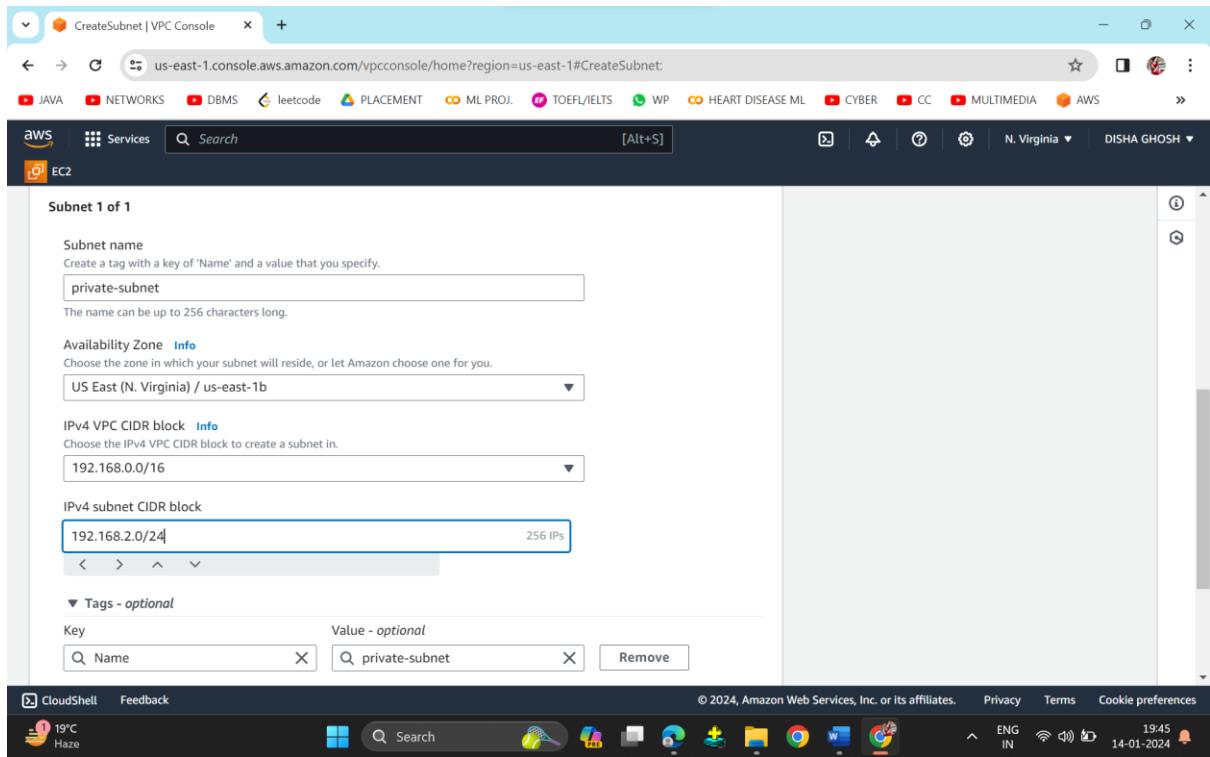
Step-5 Select VPC ID → subnet name: public-subnet → availability zone → US East(N.Virginia)/us-east-1a → CIDR block: 192.168.1.0/24 → create subnet

The screenshot shows the 'Create subnet' wizard. The first step, 'VPC', is completed with the VPC ID set to 'vpc-081459e189d8b287d (disha-vpc)'. The next step, 'Subnet settings', is currently active. It asks for the CIDR block, which is set to '192.168.0.0/16'. Below this, there's a section for 'Subnet 1 of 1' where users can specify the subnet name and choose an availability zone. The 'Availability Zone' dropdown is set to 'us-east-1a'.



Step-6 Again, click on create subnet → select VPC ID → subnet name: private-subnet → availability zone: US East (N.Virginia)/us-east-1b → CIDR Block: 192.168.2.0/24 → create subnet.





Two Subnets: Public and Private Subnets are available.

Name	Subnet ID	State	VPC
-	subnet-U2t1e5/d54e8a9/1/	Available	vpc-054e78d65b6391585
public-subnet	subnet-0f95063f498f38190	Available	vpc-081459e189d8b287d
-	subnet-02a4f734ba9f7db53	Available	vpc-054e78d65b6391585
-	subnet-0566e7bcf68a45fb07	Available	vpc-054e78d65b6391585
private-subnet	subnet-0ccb99c7761d8b4ae	Available	vpc-081459e189d8b287d

Step-7 Login to Internet Gateways and click on create internet gateway

The screenshot shows the AWS VPC Console with the URL us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#igws. The left sidebar is expanded to show the 'Virtual private cloud' section, with 'Internet gateways' selected. The main area displays a table titled 'Internet gateways (1)'. The table has columns for Name, Internet gateway ID, State, and VPC ID. One entry is listed: 'igw-07b394536aa1144bd' (State: Attached, VPC ID: vpc-054e78d65b63915). A large orange button at the top right of the table area says 'Create internet gateway'.

Step-8 Write my-igw on name tag and click on create internet gateway.

The screenshot shows the 'Create internet gateway' wizard with the URL us-east-1.console.aws.amazon.com/vpcconsole/home?region=us-east-1#CreateInternetGateway. The 'Internet gateway settings' step is active. It has two sections: 'Name tag' and 'Tags - optional'. In the 'Name tag' section, the tag key is 'Name' and the value is 'my-igw'. In the 'Tags - optional' section, there is one tag: 'Name' with value 'my-igw'. At the bottom right of the wizard is an orange button labeled 'Create internet gateway'.

Step-9 my-igw is created and attached.

The screenshot shows the AWS VPC Internet Gateways console. The left sidebar is collapsed. The main area displays a table titled "Internet gateways (1/2) Info". The table has columns: Name, Internet gateway ID, State, and VPC ID. There are two entries:

Name	Internet gateway ID	State	VPC ID
my-igw	igw-00f1d685daf63f38a	Attached	vpc-081459e189d8b287d
-	igw-07b394536aa1144bd	Attached	vpc-054e78d65b63915

Below the table, a modal window is open for the internet gateway with ID "igw-00f1d685daf63f38a", titled "igw-00f1d685daf63f38a / my-igw". The "Details" tab is selected, showing the following information:

- Internet gateway ID: igw-00f1d685daf63f38a
- State: Attached
- VPC ID: vpc-081459e189d8b287d
- Owner: 954976550727 (disha-vpc)

Step-10 Click on Route Table → Edit the name and write Public-Routing
→ create Route Table → Name: Private-Routing → VPC: select the disha rule → create Route Table.

The screenshot shows the AWS VPC Route Tables console. The left sidebar is collapsed. The main area displays a table titled "Route tables (1/3) Info". The table has columns: Name, Route table ID, Explicit subnet associations, and Edge associations. There are three entries:

Name	Route table ID	Explicit subnet associations	Edge associations
private-routing	rtb-0f29430bcb00b1894	-	-
-	rtb-04d97d855d35f6746	-	-
Public-Routing	rtb-08bac5060879af4ee	-	-

Below the table, a modal window is open for the route table with ID "rtb-08bac5060879af4ee", titled "rtb-08bac5060879af4ee / Public-Routing". The "Routes (1)" tab is selected, showing the following route:

Destination	Target	Status	Propagated
12.0.0.0/16	10.0.0.1	Active	Yes

Step-11 Select Public-Routing → go to route below → edit routes → add route → 0.0.0.0/0 → Target: Internet Gateway → save changes.

The screenshot shows the AWS VPC console interface. The top navigation bar includes links for JAVA, NETWORKS, DBMS, leetcode, PLACEMENT, ML PROJ., TOEFL/IELTS, WP, HEART DISEASE ML, CYBER, CC, MULTIMEDIA, and AWS. The user is in the N. Virginia region, logged in as DISHA GHOSH.

The main content area is titled "Edit routes". It displays a table with columns: Destination, Target, Status, and Propagated. A new route is being added:

Destination	Target	Status	Propagated
192.168.0.0/16	local	Active	No
0.0.0.0/0	Internet Gateway	-	No
	igw-00f1d685daf63f38a		Remote

Buttons at the bottom include "Add route", "Cancel", "Preview", and a prominent orange "Save changes" button.

The screenshot shows the AWS VPC console interface again. The top navigation bar is identical. The main content area shows a green success message: "Updated routes for rtb-08bac5060879af4ee / Public-Routing successfully". Below this, the "rtb-08bac5060879af4ee / Public-Routing" details page is displayed. The "Details" tab is selected, showing information like Route table ID (rtb-08bac5060879af4ee), Main (Yes), and Owner ID (954976550727). The "Routes" tab shows two routes: one to 0.0.0.0/0 targeting an Internet Gateway (igw-00f1d685daf63f38a).

Step-12 Select Public-Routing → go to subnet association below → click on public subnet → save association → private-routing → directly go to the subnet associations → private subnet → save associations.

VPC > Route tables > rtb-08bac5060879af4ee > Edit subnet associations

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (1/2)

Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
<input checked="" type="checkbox"/> public-subnet	subnet-0f95063f498f38190	192.168.1.0/24	-	Main (rtb-08bac5060879af4ee /)
<input type="checkbox"/> private-subnet	subnet-0ccb99c7761d8b4ae	192.168.2.0/24	-	Main (rtb-08bac5060879af4ee /)

Selected subnets

subnet-0f95063f498f38190 / public-subnet

Cancel **Save associations**

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

Route tables (1/3) Info

You have successfully updated subnet associations for rtb-0f29430bcb00b1894 / private-routing.

Name	Route table ID	Explicit subnet associati...	Edge associations
<input checked="" type="checkbox"/> private-routing	rtb-0f29430bcb00b1894	subnet-0ccb99c7761d8b...	-
<input type="checkbox"/> -	rtb-04d97d855d35f6746	-	-
<input type="checkbox"/> Public-Routing	rtb-08bac5060879af4ee	subnet-0f95063f498f381...	-

rtb-0f29430bcb00b1894 / private-routing

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

Details

Step-13 Login to EC2 Dashboard → Launch Instance → name: public-subnet → select Linux → MyCloud → go to network settings edit option and select vpc → check all the public and private subnet available → Auto-assign public IP: should be ENABLE → Launch instance.

The screenshot shows the 'Launch an instance' wizard. In the 'Name and tags' section, the name is set to 'public-subnet'. Under 'Application and OS Images (Amazon Machine Image)', the AMI is set to 'Amazon Linux 2023.3.2...'. The 'Virtual server type (instance type)' is 't2.micro'. The 'Firewall (security group)' is set to 'New security group'. Under 'Storage (volumes)', there is 1 volume(s) - 8 GB. At the bottom right, the 'Launch instance' button is highlighted.

The screenshot shows the 'Launch an instance' wizard with advanced network settings. In the 'Network settings' section, the VPC is selected as 'vpc-081459e189d8b287d (disha-vpc)'. The subnet is 'subnet-0f95063f498f38190' under 'public-subnet'. The 'Auto-assign public IP' checkbox is checked. The 'Firewall (security groups)' section indicates a new security group is being used. The 'Launch instance' button is visible at the bottom right.

Step-14 Again, click on Launch instance → name: private-subnet → select Linux → MyCloud → Edit on network settings → select vpc → check subnet, it should be Private → Auto-assign public IP: should be DISABLE → Launch instance.

The screenshot shows the 'Launch an instance' wizard. In the 'Name and tags' section, the name 'private-subnet' is entered. Under 'Application and OS Images (Amazon Machine Image)', the 'Software Image (AMI)' is set to 'Amazon Linux 2023 AMI 2023.3.2...'. The 'Virtual server type (instance type)' is 't2.micro'. The 'Firewall (security group)' is set to 'New security group'. The 'Storage (volumes)' section shows '1 volume(s) - 8 GiB'. At the bottom right, the 'Launch instance' button is highlighted in orange.

This screenshot shows the 'VPC - required' configuration step. The VPC dropdown is set to 'vpc-081459e189d8b287d (disha-vpc) 192.168.0.0/16'. In the 'Subnet' section, 'subnet-0ccb99c7761d8b4ae' is selected, which is part of the 'private-subnet' VPC. The 'Auto-assign public IP' dropdown is set to 'Disable'. Under 'Firewall (security groups)', the 'Create security group' option is selected. A new security group named 'launch-wizard-27' is being created. The 'Description' field contains the text 'launch-wizard-27 created 2024-01-15T04:27:56.980Z'. The right side of the screen shows the summary and launch buttons, identical to the previous screenshot.

Step-15 Go to the virtual machine and copy the public key address and open the command prompt.

The screenshot shows the AWS EC2 Instances page. The instance summary for i-0ba4829e76d745bf8 (public-subnet) is displayed. The Public IPv4 address is listed as 54.147.131.211, with a link to 'open address'. The instance state is Running. Other details include Private IP 192.168.1.24, Public IPv4 DNS ip-192-168-1-24.ec2.internal, Instance type t2.micro, VPC ID vpc-081459e189d8b287d, and Subnet ID. A tooltip indicates that the Public IPv4 address has been copied. The left sidebar shows navigation options like EC2 Dashboard, EC2 Global View, Events, Console-to-Code Preview, Instances, Images, AMIs, AMI Catalog, Elastic Block Store, Network & Security, and more.

Step-16 Write ssh -i "Downloads\MYCLOUD.pem" ec2-user@54.147.131.211

The screenshot shows a Microsoft Windows Command Prompt window. The user is running the command: ssh -i "Downloads\MYCLOUD.pem" ec2-user@54.147.131.211. The command prompt shows the usage of the ssh command and then the specific command being entered. The background shows the AWS EC2 Instances page with the same instance details as the previous screenshot.

Step-17 Virtual machine is created and formed.

The screenshot shows a Microsoft Windows Command Prompt window titled "ec2-user@ip-192-168-1-24~". The command entered is "ssh -i "Downloads\MYCLOUD.pem" ec2-user@54.147.131.211". The output indicates that the host's authenticity cannot be established, and the key fingerprint is SHA256:h9+ncR6mV4jYLkSBYrF0yq9Ml/F/D09kygK36RLxeeU. A warning message asks if the user wants to permanently add the host to the list of known hosts, to which the user responds "yes". The session then connects to the Amazon Linux 2023 instance at https://aws.amazon.com/linux/amazon-linux-2023. The command prompt ends with "[ec2-user@ip-192-168-1-24 ~]\$".

Step-18 Upload the given fwdphppage file.

The screenshot shows a Microsoft Windows Command Prompt window titled "Command Prompt". The user has already connected to the EC2 instance from Step 17. They run the command "scp -i "Downloads\MYCLOUD.pem" downloads\fwdphppage.zip ec2-user@54.147.131.211:/tmp". The output shows the file being transferred with a progress bar indicating 100% completion at 1977 bytes, 7.4KB/s, and 00:00. The command prompt ends with "C:\Users\Lenovo>".

```

Command Prompt
ED25519 key fingerprint is SHA256:h9+ncR6mV4jYLkSBYrFOyq9ML/F/Do9kygK36RLxeeU.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '54.147.131.211' (ED25519) to the list of known hosts.

#_
~\_ #####_      Amazon Linux 2023
~~ \#####\
~~ \###|
~~ \#/ ___ https://aws.amazon.com/linux/amazon-linux-2023
Event V~' '->
Co ~~~ /_
Pr ~~~ .-/
~/m'_

▶ Ins [ec2-user@ip-192-168-1-24 ~]$ exit
logout
▼ Im Connection to 54.147.131.211 closed.

AM C:\Users\Lenovo>scp -i "Downloads\MYCLOUD.pem" downloads
AM usage: scp [-346ABCpqrv] [-c cipher] [-F ssh_config] [-i identity_file]
AM           [-J destination] [-l limit] [-o ssh_option] [-P port]
AM           [-S program] source ... target
▼ Ela
▼ Net
C:\Users\Lenovo>
Sel
Elastic IPs IAM Role Subnet ID Auto Scaling Group name
CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
16°C Low visibility Search ENG IN 10:11 15-01-2024

```

Step-19 The given fwdphppage.zip file is uploaded

```

root@ip-192-168-1-24:/tmp
fwdphppage.zip          100% 1977    7.4KB/s  00:00
C:\Users\Lenovo>scp -i "Downloads\MYCLOUD.pem" downloads\MYCLOUD.pem ec2-user@54.147.131.211:/tmp
MYCLOUD.pem             100% 1678    5.9KB/s  00:00

C:\Users\Lenovo>ssh -i "Downloads\MYCLOUD.pem" ec2-user@54.147.131.211
#_
~\_ #####_      Amazon Linux 2023
~~ \#####\
~~ \###|
~~ \#/ ___ https://aws.amazon.com/linux/amazon-linux-2023
Event V~' '->
Co ~~~ /_
Pr ~~~ .-/
~/m'_

▶ Ins [ec2-user@ip-192-168-1-24 ~]$ sudo bash
[ec2-user@ip-192-168-1-24 ~]$ sudo bash
[root@ip-192-168-1-24 ec2-user]# cd/tmp
bash: cd/tmp: No such file or directory
[root@ip-192-168-1-24 ec2-user]# cd /tmp
[root@ip-192-168-1-24 tmp]# ls
MYCLOUD.pem
fwdphppage.zip
systemd-private-aflc2e6afe684fcda8b280f552af2d34-chronyd.service-3XwDsn
systemd-private-aflc2e6afe684fcda8b280f552af2d34-dbus-broker.service-GQplHR
systemd-private-aflc2e6afe684fcda8b280f552af2d34-policy-routes@enX0.service-doOQeN
systemd-private-aflc2e6afe684fcda8b280f552af2d34-systemd-logind.service-StnjCy
[root@ip-192-168-1-24 tmp]# |
Sel
Elastic IPs Required
CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences
16°C Fog Search ENG IN 10:13 15-01-2024

```

```

root@ip-192-168-1-24:/tmp ~]$ sudo bash
[root@ip-192-168-1-24 ec2-user]# cd/tmp
bash: cd/tmp: No such file or directory
[root@ip-192-168-1-24 ec2-user]# cd /tmp
[root@ip-192-168-1-24 tmp]# ls
MYCLOUD.pem
fwdphppage.zip
systemd-private-a1c2e6afe684fcda8b280f552af2d34-chrony.service-3XwDsn
systemd-private-a1c2e6afe684fcda8b280f552af2d34-dbus-broker.service-GQplHR
systemd-private-a1c2e6afe684fcda8b280f552af2d34-policy-routes@enX0.service-doOqeN
systemd-private-a1c2e6afe684fcda8b280f552af2d34-systemd-logind.service-StnjCy
systemd-private-a1c2e6afe684fcda8b280f552af2d34-systemd-resolved.service-V5dGW9
[root@ip-192-168-1-24 tmp]# yum install httpd* php* -y
Last metadata expiration check: 0:16:05 ago on Mon Jan 15 04:28:22 2024.
Error:
Problem 1: conflicting requests
- package php8.2-common-8.2.9-1.amzn2023.0.3.x86_64 conflicts with php-common < 8.2.0 provided by php8.1-common-8.1.14-1.amzn2023.0.2.x86_64
- package php8.2-common-8.2.9-1.amzn2023.0.3.x86_64 conflicts with php-common < 8.2.0 provided by php8.1-common-8.1.22-1.amzn2023.0.1.x86_64
- package php8.1-common-8.1.22-1.amzn2023.0.1.x86_64 conflicts with php-common > 8.1.99 provided by php8.2-common-8.2.9-1.amzn2023.0.3.x86_64
- package php8.2-common-8.2.9-1.amzn2023.0.3.x86_64 conflicts with php-common < 8.2.0 provided by php8.1-common-8.1.21-1.amzn2023.0.1.x86_64
- package php8.1-common-8.1.21-1.amzn2023.0.1.x86_64 conflicts with php-common > 8.1.99 provided by php8.2-common-8.2.9-1.amzn2023.0.3.x86_64
- package php8.2-common-8.2.9-1.amzn2023.0.3.x86_64 conflicts with php-common < 8.2.0 provided by php8.1-common-8.1.23-1.amzn2023.0.1.x86_64

```

Page 24 of 24 1547 words English (India) Accessibility: Investigate Focus ENG IN 10:15 15-01-2024

```

root@ip-192-168-1-24:/var/www ~]$ php8.1-process-8.1.23-1.amzn2023.0.1.x86_64
php8.1-snmp-8.1.23-1.amzn2023.0.1.x86_64
php8.1-tidy-8.1.23-1.amzn2023.0.1.x86_64
pixman-0.40.0-3.amzn2023.0.3.x86_64
unixODBC-2.3.9-3.amzn2023.0.2.x86_64
xz-devel-5.2.5-9.amzn2023.0.2.x86_64
Skipped:
php8.2-8.2.9-1.amzn2023.0.3.x86_64
php8.2-cli-8.2.9-1.amzn2023.0.3.x86_64
php8.2-dba-8.2.9-1.amzn2023.0.3.x86_64
php8.2-devel-8.2.9-1.amzn2023.0.3.x86_64
php8.2-enchant-8.2.9-1.amzn2023.0.3.x86_64
php8.2-fpm-8.2.9-1.amzn2023.0.3.x86_64
php8.2-gmp-8.2.9-1.amzn2023.0.3.x86_64
php8.2-ldap-8.2.9-1.amzn2023.0.3.x86_64
php8.2-mysqlind-8.2.9-1.amzn2023.0.3.x86_64
php8.2-pdo-8.2.9-1.amzn2023.0.3.x86_64
php8.2-pspell-8.2.9-1.amzn2023.0.3.x86_64
php8.2-soap-8.2.9-1.amzn2023.0.3.x86_64
php8.2-tidy-8.2.9-1.amzn2023.0.3.x86_64
php8.2-zip-8.2.9-1.amzn2023.0.3.x86_64
Complete!
[root@ip-192-168-1-24 tmp]# cp fwdphppage.zip /var/www/html
[root@ip-192-168-1-24 tmp]# cd /var/www/html
bash: cd /var/www/html: No such file or directory
[root@ip-192-168-1-24 tmp]# cd /var/www/html
[root@ip-192-168-1-24 html]# ls
fwdphppage.zip
[root@ip-192-168-1-24 html]#

```

Downloads Microsoft Word Doc... 320 KB 19-12-2023 19:42 ENG IN 10:19 15-01-2024

```

root@ip-192-168-1-24:/var/www/html# unzip fwdphppage.zip
Archive:  fwdphppage.zip
  inflating: db_info.txt
  inflating: Auth.php
  inflating: index.php
[root@ip-192-168-1-24 html]# ls
Auth.php  db_info.txt  fwdphppage.zip  index.php
[root@ip-192-168-1-24 html]# service httpd start
Redirecting to /bin/systemctl start httpd.service
[root@ip-192-168-1-24 html]#

```

Page 25 of 26 1547 words English (India) Accessibility: Investigate

16°C Fog 10:23 15-01-2024

Step-20 Copy the public IP address in new tab and the zip file will open.

Employee Registration Form

Employee ID	<input type="text"/>
Employee Name	<input type="text"/>
Username	<input type="text"/>
Password	<input type="text"/>
Email ID	<input type="text"/>
Phone Number	<input type="text"/>
<input type="button" value="Register"/>	

← → ⌂ △ Not secure 54.147.131.211

JAVA NETWORKS DBMS leetcode PLACEMENT ML PROJ. TOEFL/IELTS WP HEART DISEASE ML CYBER CC MULTIMEDIA AWS

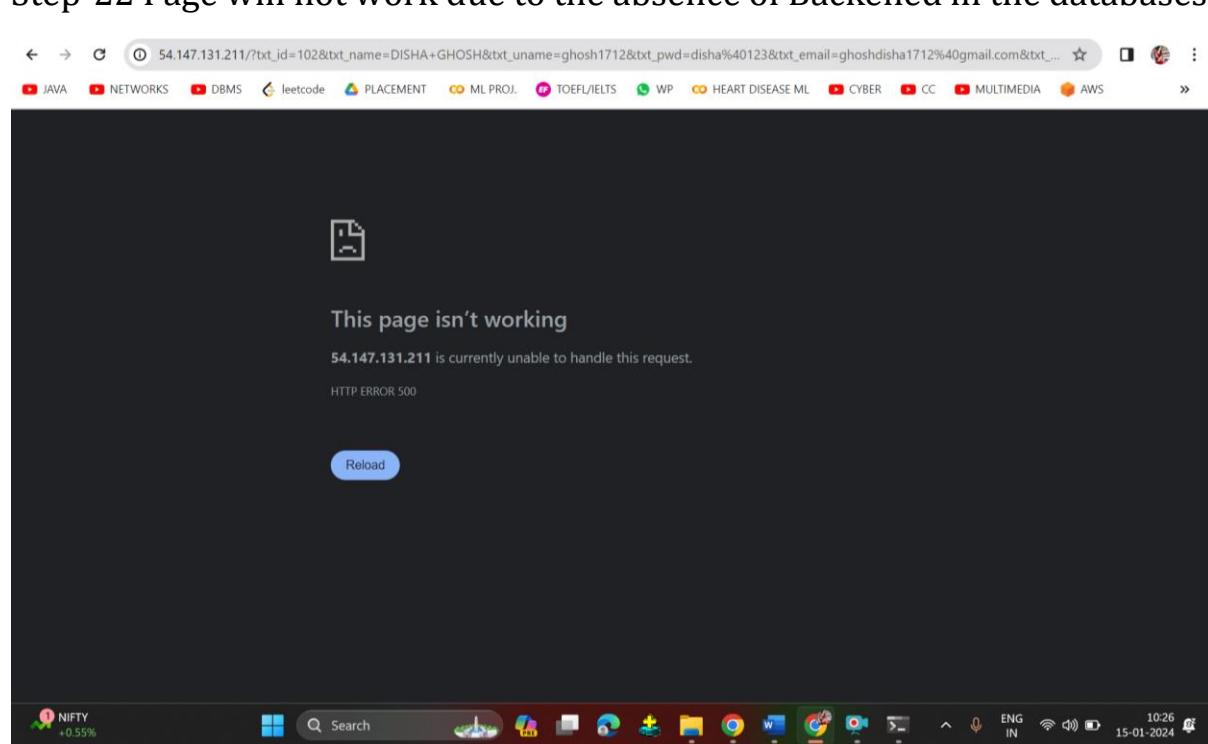
16°C Fog 10:22 15-01-2024

Step-21 After giving all the information → go to register

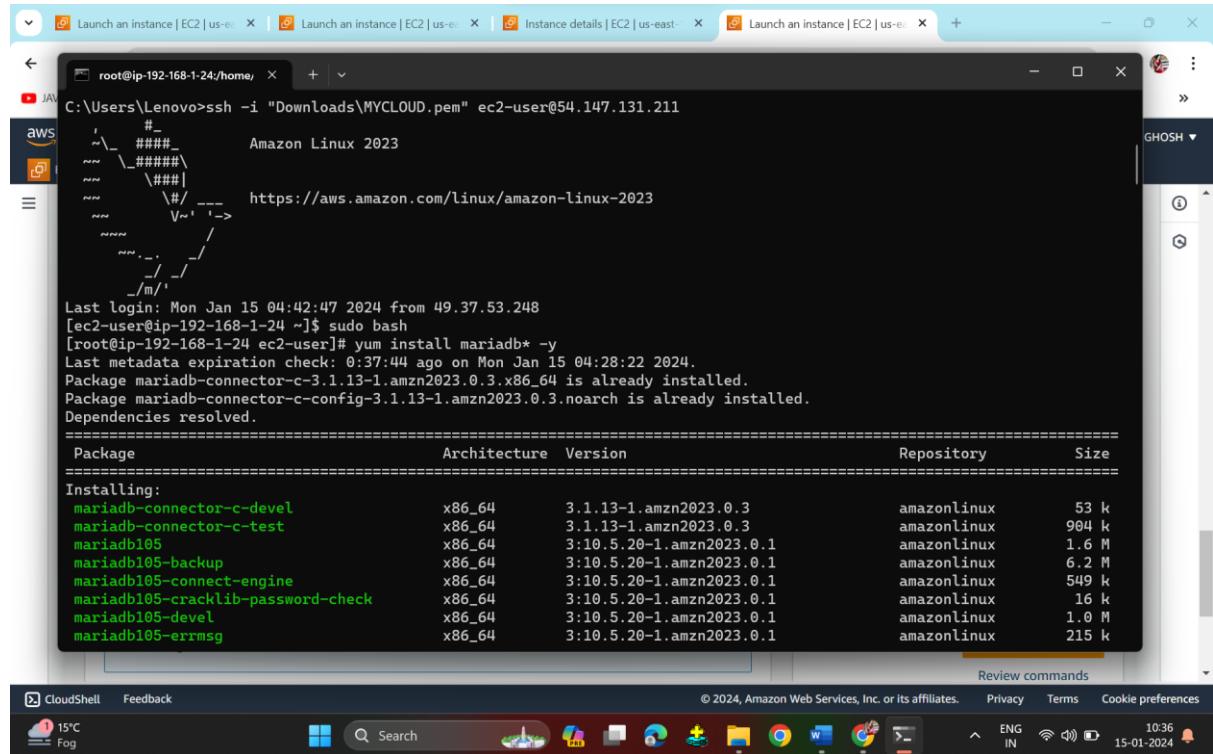
Employee Registration Form

Employee ID	102
Employee Name	DISHA GHOSH
Username	ghosh1712
Password	*****
Email ID	ghoshdisha1712@gmail.com
Phone Number	8918809965
<input type="button" value="Register"/>	

Step-22 Page will not work due to the absence of Backend in the databases.



Step-23 To build Backened in the databases, first we have to install mariadb.



The screenshot shows a CloudShell terminal window with four tabs open, all titled "Launch an instance | EC2 | us-east-1". The active tab displays a terminal session on an Amazon Linux 2023 instance. The user is installing MariaDB using the command:

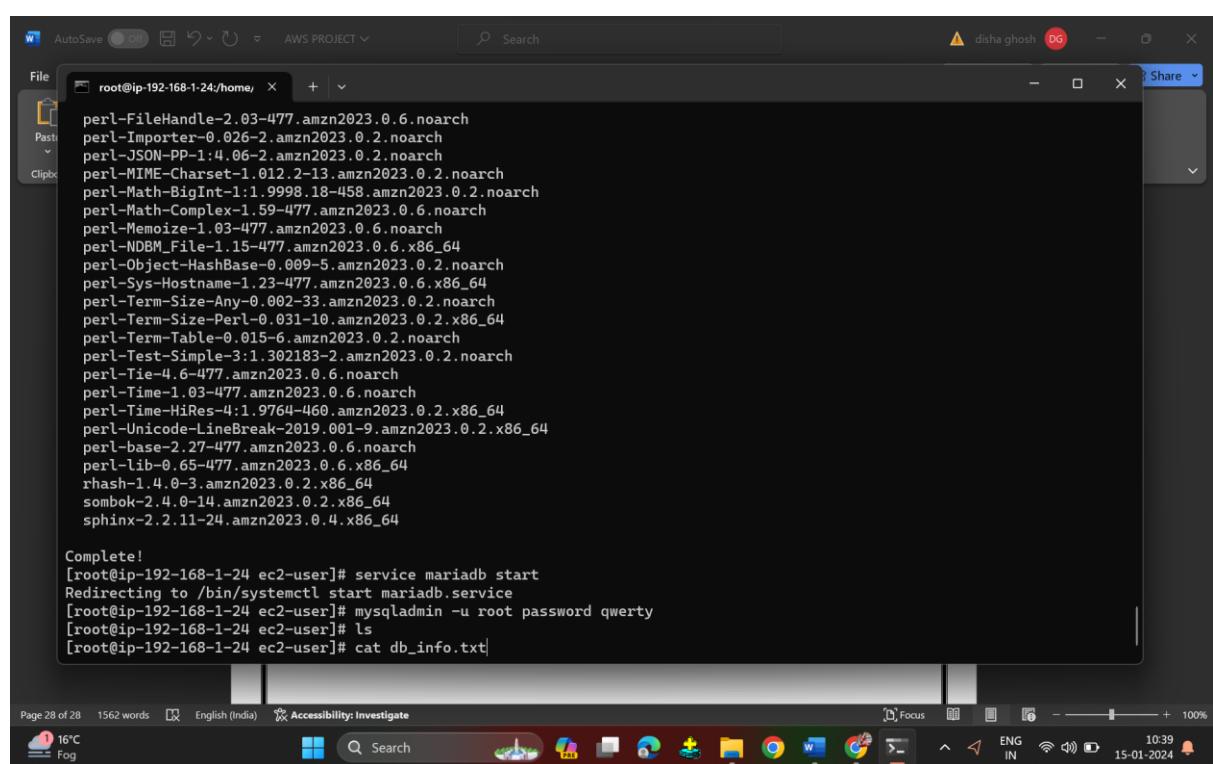
```
[root@ip-192-168-1-24 ~]# yum install mariadb* -y
```

The output shows the package is already installed:

```
Last metadata expiration check: 0:37:44 ago on Mon Jan 15 04:28:22 2024.  
Package mariadb-connector-c-3.1.13-1.amzn2023.0.3.x86_64 is already installed.  
Package mariadb-connector-c-config-3.1.13-1.amzn2023.0.3.noarch is already installed.  
Dependencies resolved.
```

A table of installed packages is shown:

Package	Architecture	Version	Repository	Size
mariadb-connector-c-devel	x86_64	3.1.13-1.amzn2023.0.3	amazonlinux	53 k
mariadb-connector-c-test	x86_64	3.1.13-1.amzn2023.0.3	amazonlinux	904 k
mariadb105	x86_64	3:10.5.20-1.amzn2023.0.1	amazonlinux	1.6 M
mariadb105-backup	x86_64	3:10.5.20-1.amzn2023.0.1	amazonlinux	6.2 M
mariadb105-connect-engine	x86_64	3:10.5.20-1.amzn2023.0.1	amazonlinux	549 k
mariadb105-cracklib-password-check	x86_64	3:10.5.20-1.amzn2023.0.1	amazonlinux	16 k
mariadb105-devel	x86_64	3:10.5.20-1.amzn2023.0.1	amazonlinux	1.0 M
mariadb105-errmsg	x86_64	3:10.5.20-1.amzn2023.0.1	amazonlinux	215 k



The screenshot shows a CloudShell terminal window with four tabs open, all titled "Launch an instance | EC2 | us-east-1". The active tab displays a terminal session on an Amazon Linux 2023 instance. The user is starting the MariaDB service and changing the root password:

```
[root@ip-192-168-1-24 ~]# service mariadb start  
Redirecting to /bin/systemctl start mariadb.service  
[root@ip-192-168-1-24 ~]# mysqladmin -u root password qwerty  
[root@ip-192-168-1-24 ~]# ls  
[root@ip-192-168-1-24 ~]# cat db_info.txt
```

The terminal window has a dark theme and includes standard Linux navigation keys like Esc, F1-F12, and arrow keys.

```
root@ip-192-168-1-24:/home/ + | v
perl-Term-Size-Perl-0.031-10.amzn2023.0.2.x86_64
perl-Term-Table-0.015-6.amzn2023.0.2.noarch
perl-Test-Simple-3.1.302183-2.amzn2023.0.2.noarch
perl-Tie-4.6-477.amzn2023.0.6.noarch
perl-Time-1.03-477.amzn2023.0.6.noarch
perl-Time-HiRes-4;1.9764-460.amzn2023.0.2.x86_64
perl-Unicode-LineBreak-2019.001-9.amzn2023.0.2.x86_64
perl-base-2.27-477.amzn2023.0.6.noarch
perl-lib-0.65-477.amzn2023.0.6.x86_64
rhash-1.4.0-3.amzn2023.0.2.x86_64
sombok-2.4.0-14.amzn2023.0.2.x86_64
sphinx-2.2.11-24.amzn2023.0.4.x86_64

Complete!
[root@ip-192-168-1-24 ec2-user]# service mariadb start
Redirecting to /bin/systemctl start mariadb.service
[root@ip-192-168-1-24 ec2-user]# mysqladmin -u root password qwerty
[root@ip-192-168-1-24 ec2-user]# ls
[root@ip-192-168-1-24 ec2-user]# cat db_info.txt
cat: db_info.txt: No such file or directory
[root@ip-192-168-1-24 ec2-user]# mysql -u root -pqwerty
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 4
Server version: 10.5.20-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]> |
```

Review commands

CloudShell Feedback © 2024, Amazon Web Services, Inc. or its affiliates. Privacy Terms Cookie preferences

16°C Fog 10:42 15-01-2024

Step-24 Show databases and create database employee

```
File Home In AutoSave Off AWS PROJECT v Search disha ghosh - X
+-----+
| employe |
| employee |
| information_schema |
| mysql |
| performance_schema |
+-----+
5 rows in set (0.000 sec)

MariaDB [(none)]> delete databases employe;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'databases employe' at line 1
MariaDB [(none)]> drop employe;
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near 'employe' at line 1
MariaDB [(none)]> drop database employe;
Query OK, 0 rows affected (0.001 sec)

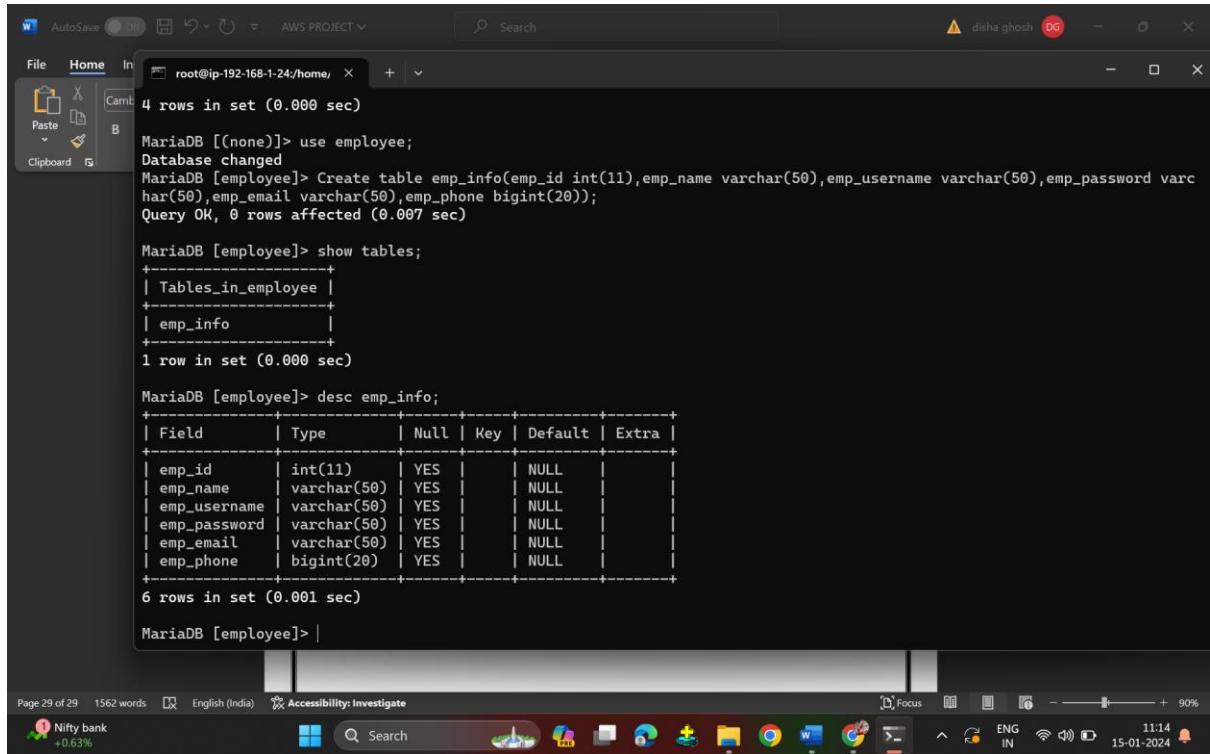
MariaDB [(none)]> show databases;
+-----+
| Database |
+-----+
| employe |
| information_schema |
| mysql |
| performance_schema |
+-----+
4 rows in set (0.000 sec)

MariaDB [(none)]> |
```

Page 29 of 29 1562 words English (India) Accessibility: Investigate Focus ENG IN 11:10 15-01-2024 90%

16°C Fog

Step-25 Database table is formed.



The screenshot shows a terminal window titled "root@ip-192-168-1-24:/home/" with the following command history:

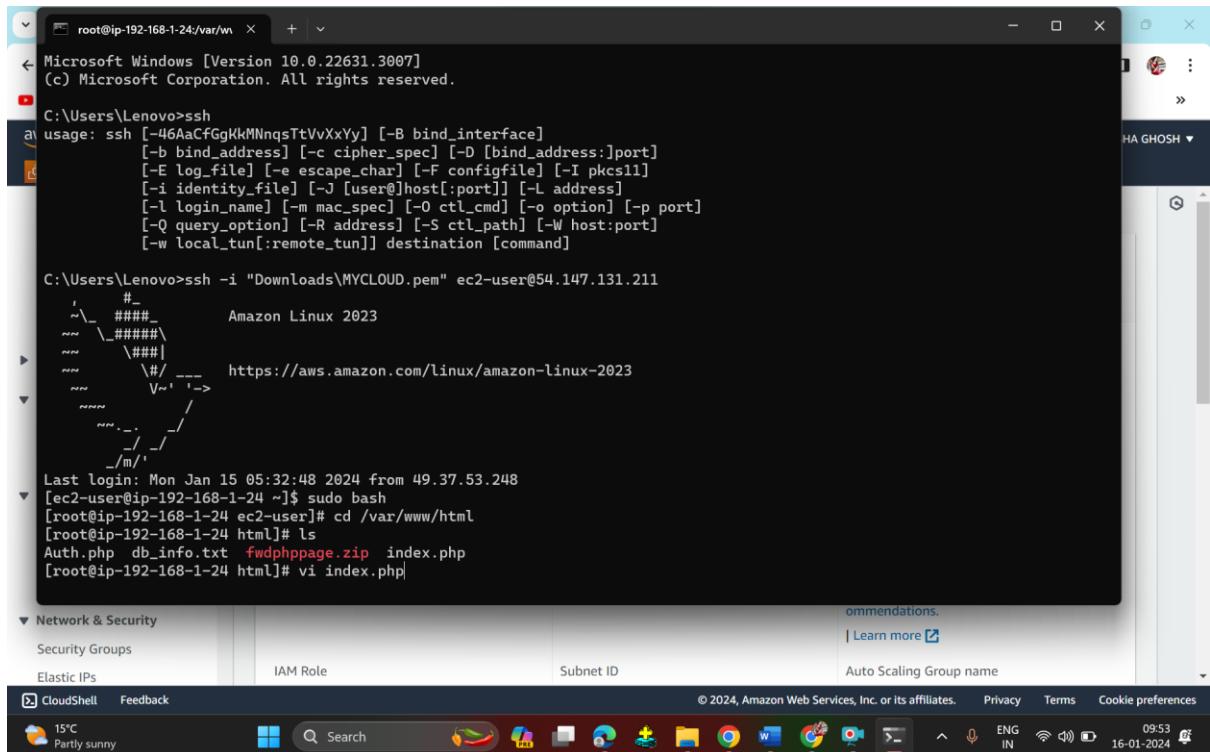
```
MariaDB [(none)]> use employee;
Database changed
MariaDB [employee]> Create table emp_info(emp_id int(11),emp_name varchar(50),emp_username varchar(50),emp_password varc
har(50),emp_email varchar(50),emp_phone bigint(20));
Query OK, 0 rows affected (0.007 sec)

MariaDB [employee]> show tables;
+-----+
| Tables_in_employee |
+-----+
| emp_info |
+-----+
1 row in set (0.000 sec)

MariaDB [employee]> desc emp_info;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| emp_id | int(11) | YES | | NULL |
| emp_name | varchar(50) | YES | | NULL |
| emp_username | varchar(50) | YES | | NULL |
| emp_password | varchar(50) | YES | | NULL |
| emp_email | varchar(50) | YES | | NULL |
| emp_phone | bigint(20) | YES | | NULL |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.001 sec)

MariaDB [employee]> |
```

Step-26 Check whether the zip file is available or not.



The screenshot shows a terminal window titled "root@ip-192-168-1-24:/var/www/html" with the following command history:

```
C:\Users\Lenovo>ssh -i "Downloads\MYCLOUD.pem" ec2-user@54.147.131.211
Last login: Mon Jan 15 05:32:48 2024 from 49.37.53.248
[ec2-user@ip-192-168-1-24 ~]$ sudo bash
[root@ip-192-168-1-24 ec2-user]# cd /var/www/html
[root@ip-192-168-1-24 html]# ls
Auth.php db_info.txt fudphppage.zip index.php
```

Step-27 Open the PHP file and press i for insert and do the needful changes, like- change the password and write “qwerty” and for saving and quitting the file press :wq

```

<?php
$msg="";
if(isset($_GET['reg']))
{
    $link= mysqli_connect("localhost","root","qwerty","employee");

    $qry="insert into emp_info values($_GET[txt_id],$_GET[txt_name'],$_GET[txt_uname'],$_GET[txt_pwd'],$_GET[txt_email'],$_GET[txt_no]);"

    mysqli_query($link,$qry);
    if(mysqli_affected_rows($link)>0)
    {
        $msg=<font color='green' size='5px'>Registration Done.....</font>;
    }
    else
    {
        $msg=<font color='red' size='5px'>Registration Fail.....</font>;
    }
    mysqli_close($link);
}
?>
<!DOCTYPE html>
<html>
    <head>
        <meta charset="UTF-8">
        <title></title>
        <script>
            temp=0;
            templ=0;
-- INSERT --

```

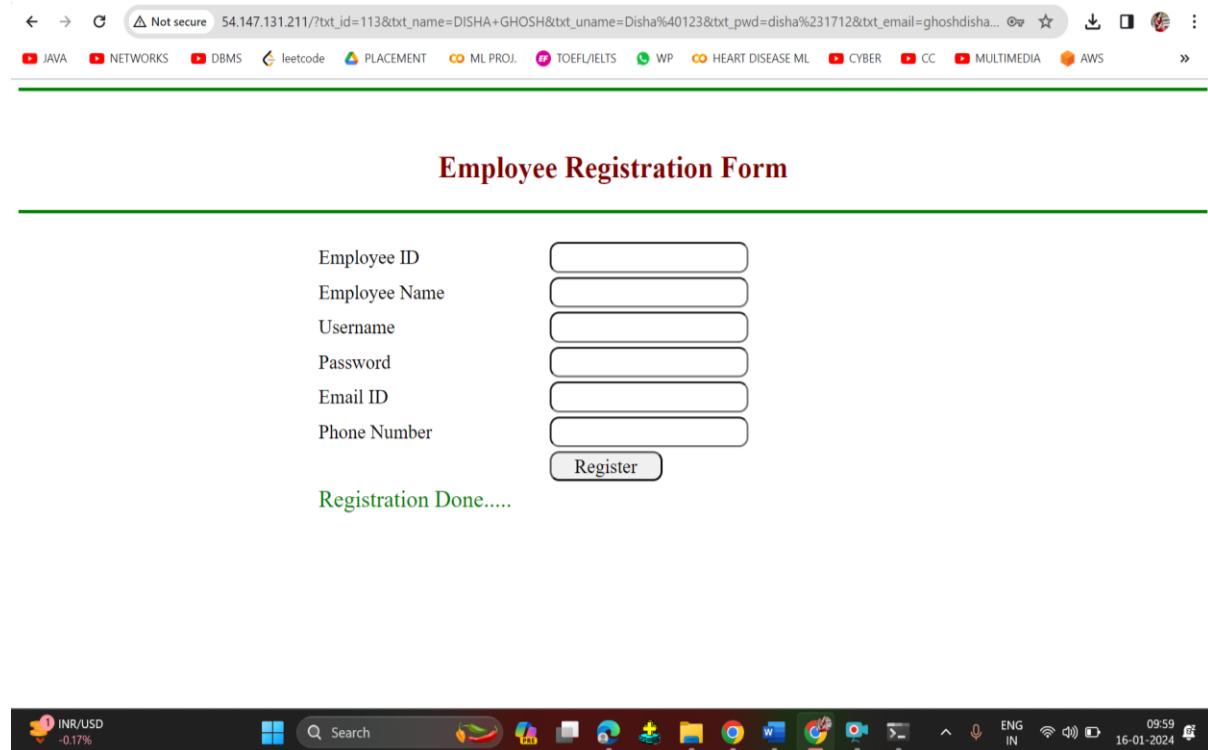
```

C:\Users\Lenovo>ssh
usage: ssh [-i46AaCfgKkMNqsTtVvXxYy] [-B bind_interface]
           [-b bind_address] [-c cipher_spec] [-D [bind_address]:port]
           [-E log_file] [-e escape_char] [-F configfile] [-I pkcs11]
           [-i identity_file] [-J [user@host[:port]] [-L address]
           [-l login_name] [-m mac_spec] [-O ctl_cmd] [-o option] [-p port]
           [-Q query_option] [-R address] [-S ctl_path] [-W host:port]
           [-w local_tun[:remote_tun]] destination [command]

C:\Users\Lenovo>ssh -i "Downloads\MYCLOUD.pem" ec2-user@54.147.131.211
#_
~\_\_ #####          Amazon Linux 2023
~~\_\_#####\_
~~ \###\_
~~  #/ -- https://aws.amazon.com/linux/amazon-linux-2023
~~ V\` ' ->
~~ / \
~~ . . / \
~~ / /
~~ /m/
Last login: Mon Jan 15 05:32:48 2024 from 49.37.53.248
[ec2-user@ip-192-168-1-24 ~]$ sudo bash
[root@ip-192-168-1-24 ec2-user]# cd /var/www/html
[root@ip-192-168-1-24 html]# ls
Auth.php db_info.txt fwpdphppage.zip index.php
[root@ip-192-168-1-24 html]# vi index.php
[root@ip-192-168-1-24 html]# vi Auth.php
[root@ip-192-168-1-24 html]# service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[root@ip-192-168-1-24 html]#

```

Step-28 Open a new tab and paste the public ip address and fill the registration form.



Step-29 An employee registration is completed in the web server.

```
[root@ip-192-168-1-24 ~]# vi Auth.php
[root@ip-192-168-1-24 ~]# service httpd restart
Redirecting to /bin/systemctl restart httpd.service
[root@ip-192-168-1-24 ~]# mysql -u root -pqwerty
bash: mysql: command not found
[root@ip-192-168-1-24 ~]# service mariadb restart
Redirecting to /bin/systemctl restart mariadb.service
[root@ip-192-168-1-24 ~]# mysql -u root -pqwerty
Welcome to the MariaDB monitor. Commands end with ; or \g.
Your MariaDB connection id is 6
Server version: 10.5.20-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

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

MariaDB [(none)]> use employee
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
MariaDB [employee]> Select * from emp_info;
+-----+-----+-----+-----+-----+
| emp_id | emp_name | emp_username | emp_password | emp_email      | emp_phone |
+-----+-----+-----+-----+-----+
|    113 | DISHA GHOSH | Disha@123 | disha#1712 | ghoshdisha1712@gmail.com | 8918809965 |
+-----+-----+-----+-----+-----+
1 row in set (0.000 sec)

MariaDB [employee]>
```

The terminal window shows the user performing several actions: editing an PHP file, restarting the httpd service, and then connecting to the MySQL database as root with the password "pqwerty". After connecting, the user selects the "employee" database and runs a SELECT query on the "emp_info" table. The output shows one row of data: emp_id 113, emp_name DISHA GHOSH, emp_username Disha@123, emp_password disha#1712, emp_email ghoshdisha1712@gmail.com, and emp_phone 8918809965. The time at the bottom right of the terminal window is 09:59.

Step-30 Here are some of the glimpses of employees who are registered in the web server

```
root@ip-192-168-1-24:/var/www/html - + v
Your MariaDB connection id is 6
Server version: 10.5.20-MariaDB MariaDB Server
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

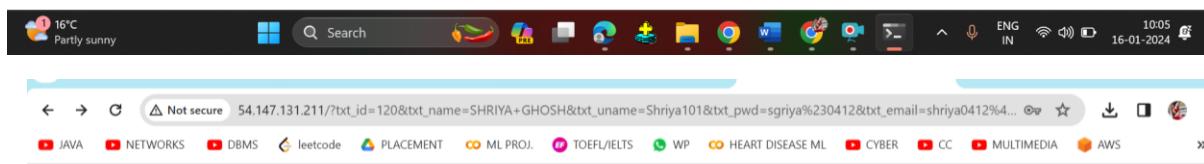
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> use employee
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
MariaDB [employee]> Select * from emp_info;
+-----+-----+-----+-----+-----+
| emp_id | emp_name | emp_username | emp_password | emp_email | emp_phone |
+-----+-----+-----+-----+-----+
| 113 | DISHA GHOSH | Disha@123 | disha#1712 | ghoshdisha1712@gmail.com | 8918809965 |
+-----+-----+-----+-----+-----+
1 row in set (0.000 sec)

MariaDB [employee]> Select * from emp_info;
+-----+-----+-----+-----+-----+
| emp_id | emp_name | emp_username | emp_password | emp_email | emp_phone |
+-----+-----+-----+-----+-----+
| 113 | DISHA GHOSH | Disha@123 | disha#1712 | ghoshdisha1712@gmail.com | 8918809965 |
| 120 | SHRIYA GHOSH | Shriya101 | sgriya#0412 | shriya0412@gmail.com | 9564241100 |
+-----+-----+-----+-----+-----+
2 rows in set (0.000 sec)

MariaDB [employee]>
```



Employee Registration Form

Employee ID	<input type="text" value="128"/>
Employee Name	<input type="text" value="CHANCHAL GHOSH"/>
Username	<input type="text" value="Chanchal\$190"/>
Password	<input type="password" value="*****"/>
Email ID	<input type="text" value="chanchal@789@gmail.com"/>
Phone Number	<input type="text" value="7892439990"/>

Registration Done.....



```

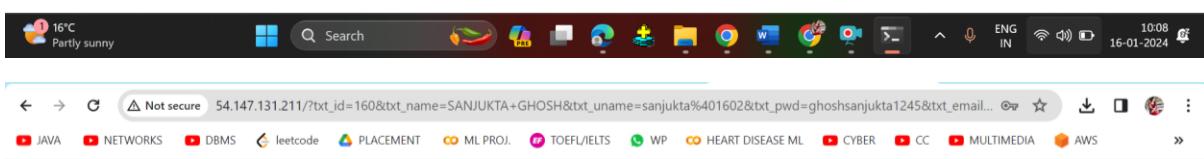
root@ip-192-168-1-24:/var/www/html - + 
Database changed
MariaDB [employee]> Select * from emp_info;
+-----+-----+-----+-----+-----+
| emp_id | emp_name | emp_username | emp_password | emp_email | emp_phone |
+-----+-----+-----+-----+-----+
| 113 | DISHA GHOSH | Disha@123 | disha#1712 | ghoshdisha1712@gmail.com | 8918809965 |
+-----+-----+-----+-----+-----+
1 row in set (0.000 sec)

MariaDB [employee]> Select * from emp_info;
+-----+-----+-----+-----+-----+
| emp_id | emp_name | emp_username | emp_password | emp_email | emp_phone |
+-----+-----+-----+-----+-----+
| 113 | DISHA GHOSH | Disha@123 | disha#1712 | ghoshdisha1712@gmail.com | 8918809965 |
| 120 | SHRIYA GHOSH | Shriya101 | sgriya#0412 | shriya0412@gmail.com | 9564241100 |
+-----+-----+-----+-----+-----+
2 rows in set (0.000 sec)

MariaDB [employee]> Select * from emp_info;
+-----+-----+-----+-----+-----+
| emp_id | emp_name | emp_username | emp_password | emp_email | emp_phone |
+-----+-----+-----+-----+-----+
| 113 | DISHA GHOSH | Disha@123 | disha#1712 | ghoshdisha1712@gmail.com | 8918809965 |
| 120 | SHRIYA GHOSH | Shriya101 | sgriya#0412 | shriya0412@gmail.com | 9564241100 |
| 128 | CHANCHAL GHOSH | Chanchal$190 | ghoshchanchal2602 | chanchal@789gmail.com | 7892439990 |
+-----+-----+-----+-----+-----+
3 rows in set (0.000 sec)

MariaDB [employee]> 

```



Employee Registration Form

Employee ID	<input type="text"/>
Employee Name	<input type="text"/>
Username	<input type="text"/>
Password	<input type="text"/>
Email ID	<input type="text"/>
Phone Number	<input type="text"/>
	<input type="button" value="Register"/>

Registration Done.....



```
W Command Prompt + v
File MariaDB [employee]> Select * from emp_info;
+-----+-----+-----+-----+-----+
| emp_id | emp_name | emp_username | emp_password | emp_email | emp_phone |
+-----+-----+-----+-----+-----+
| 113 | DISHA GHOSH | Disha@123 | disha#1712 | ghoshdisha1712@gmail.com | 8918809965 |
| 120 | SHRIYA GHOSH | Shriya101 | sgriya#0412 | shriya0412@gmail.com | 9564241100 |
| 128 | CHANCHAL GHOSH | Chanchal$190 | ghoschanchal2602 | chanchal@789gmail.com | 7892439990 |
+-----+-----+-----+-----+-----+
3 rows in set (0.000 sec)

MariaDB [employee]> Select * from emp_info;
+-----+-----+-----+-----+-----+
| emp_id | emp_name | emp_username | emp_password | emp_email | emp_phone |
+-----+-----+-----+-----+-----+
| 113 | DISHA GHOSH | Disha@123 | disha#1712 | ghoshdisha1712@gmail.com | 8918809965 |
| 120 | SHRIYA GHOSH | Shriya101 | sgriya#0412 | shriya0412@gmail.com | 9564241100 |
| 128 | CHANCHAL GHOSH | Chanchal$190 | ghoschanchal2602 | chanchal@789@gmail.com | 7892439990 |
| 160 | SANJUKTA GHOSH | sanjukta@1602 | ghoshsanjukta1245 | ghoshsanjukta2341@gmail.com | 9564209878 |
+-----+-----+-----+-----+-----+
4 rows in set (0.000 sec)

MariaDB [employee]>
Broadcast message from root@localhost (Tue 2024-01-16 04:42:41 UTC):

The system will power off now!

Connection to 54.147.131.211 closed by remote host.
Connection to 54.147.131.211 closed.

C:\Users\Lenovo>
```

Page 33 of 33 1667 words English (India) Accessibility: Investigate Focus

Breaking news Unfolding now

Search

Focus

ENG IN 11:08 16-01-2024