

3-TIER WEB APPLICATION DEPLOYMENT IN CLOUD

3-Tier Architecture:

The 3-tier architecture is a software architecture pattern that is commonly used for building web applications. It is based on the separation of concerns principle, which means that different components of the application are separated into distinct layers, each with its own responsibilities.

Here are the three tiers of the architecture:

Presentation tier:

This is the topmost layer of the application, also known as the user interface layer. Its main responsibility is to interact with the user and display information in a user-friendly manner. This layer typically consists of a web server that handles HTTP requests and generates dynamic HTML pages.

Application tier:

This is the middle layer of the application, also known as the business logic layer. Its main responsibility is to handle business logic and application functionality. This layer typically consists of a web application server that handles application logic and communicates with the database layer.

Data tier:

This is the bottommost layer of the application, also known as the data storage layer. Its main responsibility is to store and retrieve data from the database. This layer typically consists of a database server that stores and retrieves data and provides a mechanism for managing data access.

The three tiers are interconnected and communicate with each other to provide a complete application. The presentation tier interacts with the application tier to retrieve data and display it to the user, while the application tier interacts with the data tier to retrieve and store data.

By separating the application into distinct tiers, the 3-tier architecture provides several benefits, such as scalability, maintainability, and flexibility. It also allows for easier modification and testing of individual components, without affecting the entire application.

Why do we use 3-tier architecture?

We use the three-tier architecture for several reasons, including:

Scalability:

By separating the application into three distinct layers, we can scale each layer independently based on its needs. For example, we can add more web servers to the presentation layer, more application servers to the application layer, or more database servers to the data storage layer to handle increased traffic or data.

Maintainability:

Separating the application into three distinct layers also makes it easier to maintain and upgrade the system. Each layer can be upgraded or replaced separately without affecting the other layers. This makes it easier to make changes to the system without causing disruptions or downtime.

Flexibility:

The three-tier architecture allows us to use different technologies in each layer. For example, we can use a different programming language or framework in the presentation layer than in the application layer. This gives us more flexibility in designing and implementing the system.

Security:

The three-tier architecture also enhances the security of the system. The presentation layer only has access to the application layer, and the application layer only has access to the data storage layer. This reduces the risk of security breaches and makes it easier to implement security measures in each layer.

Overall, the three-tier architecture provides several benefits that make it a popular choice for building complex software systems. It offers scalability, maintainability, flexibility, and security, making it an ideal architecture pattern for many applications.

Benefits of 3-tier architecture over traditional approach:

The main difference between a three-tier architecture and a typical architecture for deploying a web application is the separation of concerns into three distinct layers.

In a typical architecture for deploying a web application, all of the components are typically located on a single server. This can make the application less scalable and harder to maintain, as changes to one component can affect the others.

In contrast, in a three-tier architecture, the application is divided into three distinct layers: presentation, application, and data storage. Each layer is deployed separately, often on different servers or virtual machines. This allows for better scalability, as each layer can be scaled independently to meet demand. It also makes the application easier to maintain, as changes to one layer do not affect the others.

For example, in a typical architecture, a web server might host the web application, the application logic, and the database. This can make it difficult to scale the application, as all of the components are tightly coupled. In a three-tier architecture, the web server might host only the presentation layer, with the application and data storage layers hosted on separate servers or virtual machines. This allows for better scalability and flexibility.

Overall, the three-tier architecture provides a more modular and flexible approach to deploying web applications, making it easier to scale, maintain, and update the application.

Tools and Technologies used:

➤ Tools:

A three-tier web application typically consists of three layers: presentation layer, application layer, and database layer. The main tools used in each layer of a three-tier web application may vary depending on the specific technology stack and programming language used, but here are some common examples:

Presentation layer:

HTML, CSS, and JavaScript for building the user interface

Client-side frameworks like React, Angular, or Vue.js for building dynamic web applications

Web browsers such as Chrome, Firefox, Safari, or Internet Explorer

Web servers such as Apache, Nginx, or IIS for serving web content to clients

Application layer:

Server-side programming languages like Java, Python, Ruby, or PHP for processing user requests and generating responses

Web application frameworks like Spring, Django, Ruby on Rails, or Laravel for building web applications faster and more efficiently

Web servers like Tomcat, Jetty, or Glassfish for running web applications

Application servers like JBoss, WebLogic, or WebSphere for providing additional features such as clustering, load balancing, and failover

Database layer:

Relational database management systems (RDBMS) like MySQL, PostgreSQL, or Oracle for storing and retrieving data

NoSQL databases like MongoDB, Cassandra, or Couchbase for storing and retrieving unstructured data

Object-relational mapping (ORM) frameworks like Hibernate, SQLAlchemy, or ActiveRecord for mapping between database tables and application objects

Data access layers like Java Persistence API (JPA), Django ORM, or Ruby on Rails Active Record for abstracting the database access from the application code.

These are just some of the many tools that can be used in each layer of a three-tier web application. The specific tools used will depend on the requirements of the application, the skills of the development team, and the available resources.

➤ Technologies:

The technologies used in a 3-tier web application deployment in the cloud.

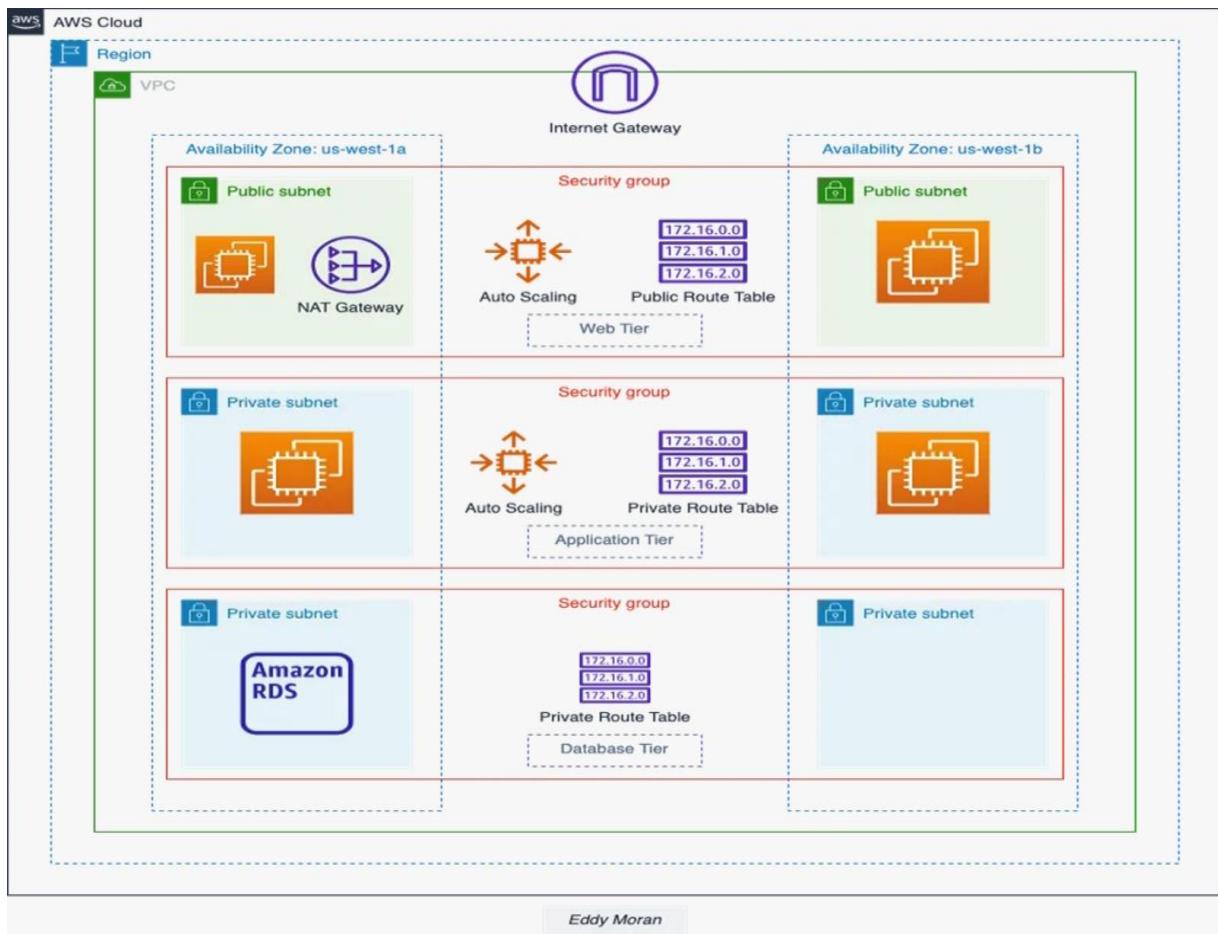
Front-end technologies: The front-end of a web application is what users see and interact with in their browsers. Common front-end technologies include HTML, CSS, and JavaScript, as well as various libraries and frameworks such as React, Angular, or Vue.js.

Back-end technologies: The back-end of a web application handles the processing of data and business logic. Common back-end technologies include programming languages such as Java, Python, or PHP, as well as frameworks like Spring, Flask, or Laravel.

Database technologies: The database is where an application stores its data. Common database technologies include relational databases such as MySQL or PostgreSQL, as well as NoSQL databases such as MongoDB or Cassandra.

Cloud infrastructure technologies: In order to deploy a 3-tier web application in the cloud, you would need to use various cloud infrastructure technologies, such as virtual machines, load balancers, and storage services. These technologies are typically provided by cloud providers like AWS, Azure, or GCP.

Architecture:



From this architecture we are having:

- **Virtual Private Cloud (VPC):**

A VPC is similar to a traditional network in terms of security and connectivity, but with the added benefits of AWS's scalability, reliability, and flexibility. It allows

you to create your own virtual network within the AWS cloud, with complete control over network configuration.

A VPC provides a completely isolated virtual network environment, allowing you to create and manage your own virtual network topology, including subnets, routing tables, and network gateways.

Key features in VPC:

1.Subnet

- A subnet is a range of IP addresses in your VPC.
- You can create one or more subnets in your VPC, and each subnet can be associated with a different availability zone.
- This allows you to create highly available and fault-tolerant applications.

2.Internet Gateway

- An internet gateway allows your instances in the VPC to communicate with the internet.
- You can use an internet gateway to route traffic to and from the internet.

3. NAT Gateway

- A Network Address Translation (NAT) gateway allows your instances in private subnets to access the internet.
- NAT gateway enables instances in the private subnet to communicate with the internet while still maintaining security.

4. Security Groups

- Security groups act as a firewall for your instances.
- You can use security groups to control traffic to and from your instances.
- Each security group can be associated with one or more instances.

5.Route Tables

- In networking, a route table is a data structure that maps the destination addresses of packets to the next hop or outgoing interface.
- It is used by routers to determine the best path for forwarding traffic between networks.

Step by step process:

❖ Creating VPC:

- Sign in to AWS Management Console.
- Go to VPC dashboard and click on create VPC. In VPC settings, create only the VPC resource.

- Create a tag with a key of “Name” and a value that you specify.
- Select IPv4 CIDR manual input.
- Give the IPv4 CIDR as 10.0.0.0/16. This IP address range includes all addresses from 10.0.0.0 to 10.255.255.255, and the subnet mask for this range is 255.255.0.0. The 10.0.0.0/16 subnet can accommodate up to 65,536 host addresses, making it suitable for larger networks.
- Click on create VPC.

The screenshot shows the 'Create VPC' wizard in the AWS Management Console. The current step is 'VPC settings'. The 'Resources to create' section has 'VPC only' selected. A 'Name tag' field contains 'proj_vpc'. Under 'IPv4 CIDR block', 'IPv4 CIDR manual input' is selected, and the CIDR block '10.0.0.0/16' is entered. The 'IPv6 CIDR block' section shows 'No IPv6 CIDR block' selected. The 'Tenancy' section shows 'Default'. In the 'Tags' section, a tag 'Name: proj_vpc' is added. At the bottom right are 'Cancel' and 'Create VPC' buttons.

❖ Creating subnet:

- Select the VPC that we created earlier to create a public subnet. In the subnet settings give the subnet name and choose ap-southeast-1a in Singapore region.
- Give the IPv4 CIDR as 10.0.1.0/24. Click on add subnet. Create 2 more private subnets one in same availability zone as public subnet and the other in another availability zone. Click on create subnets.

Create subnet [Info](#)

VPC

VPC ID
Create subnets in this VPC.

Associated VPC CIDRs
IPv4 CIDRs

Subnet settings
Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 3

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

The name can be up to 256 characters long.

Availability Zone [Info](#)
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

IPv4 CIDR block [Info](#)

[X](#)

Tags - optional

Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="public_sub1"/>
Remove	

[Add new tag](#)
You can add 49 more tags.

[Remove](#)

Subnet 2 of 3

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

The name can be up to 256 characters long.

Availability Zone [Info](#)
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

IPv4 CIDR block [Info](#)

[X](#)

Tags - optional

Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="private_sub1"/>
Remove	

Tags - optional

Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="private_sub2"/>
Remove	

[Add new tag](#)
You can add 49 more tags.

[Remove](#)

Subnet 3 of 3

Subnet name
Create a tag with a key of 'Name' and a value that you specify.

The name can be up to 256 characters long.

Availability Zone [Info](#)
Choose the zone in which your subnet will reside, or let Amazon choose one for you.

IPv4 CIDR block [Info](#)

[X](#)

Tags - optional

Key	Value - optional
<input type="text" value="Name"/>	<input type="text" value="private_sub2"/>
Remove	

[Add new tag](#)

The screenshot shows the AWS VPC dashboard with the Subnets section selected. A success message at the top states: "You have successfully created 3 subnets: subnet-0ac9ea3ffda311a1, subnet-07cb9ba615922aac0, subnet-061062e7ff823e1e8". Below this, a table lists the subnets:

	Subnet ID	Available	VPC	CIDR	
<input type="checkbox"/>	subnet-07cb9ba615922aac0	Available	vpc-05a17e3d4b6d1463a pr...	10.0.2.0/24	-
<input type="checkbox"/>	subnet-061062e7ff823e1e8	Available	vpc-05a17e3d4b6d1463a pr...	10.0.3.0/24	-
<input type="checkbox"/>	subnet-0ac9ea3ffda311a1	Available	vpc-05a17e3d4b6d1463a pr...	10.0.1.0/24	-

❖ Creating Internet Gateway (IGW):

- An internet gateway is a virtual router that connects a VPC to the internet.
- To create a new IGW specify the name for the gateway.
Click on create IGW.

The screenshot shows the "Create internet gateway" settings page. The "Name tag" field contains "proj_igw1". The "Tags - optional" section shows one tag named "proj_igw1". At the bottom, there are "Cancel" and "Create internet gateway" buttons.

- Click on Attach VPC and select the VPC that we have created earlier.

The screenshot shows the "Attach to VPC" page. The "Available VPCs" section has "vpc-05a17e3d4b6d1463a" selected. At the bottom, there are "Cancel" and "Attach internet gateway" buttons.

❖ Creating Network Address Translation (NAT) Gateway:

- A highly available, managed NAT service that instances in private subnets can use to connect to service in other VPC's , on premises networks, or the internet.
- In NAT gateway settings, give the name and select the public subnet.
- Allocate elastic IP address to the NAT gateway.

The screenshot shows the 'Create NAT gateway' wizard in the AWS VPC console. The first step, 'NAT gateway settings', is displayed. It includes fields for Name (optional), Subnet (selected as 'subnet-0ac9ea3ffdaef311a1 (public_sub1)'), Connectivity type (set to 'Public'), and an Elastic IP allocation ID ('eipalloc-0ce18beb335257a85'). An 'Allocate Elastic IP' button is present. The second step, 'Additional settings', is partially visible below it.

The screenshot shows the 'Additional settings' step of the wizard. It contains a 'Tags' section where a single tag 'Name' is added with value 'proj_nat1'. A note indicates up to 49 more tags can be added. At the bottom right is a 'Create NAT gateway' button.

❖ Creating Route Table

- A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.
- In route table settings, create a public route table by giving the name and select the VPC that was created earlier. Click on create route table.

Screenshot of the AWS VPC Route Tables creation interface:

Create route table [Info](#)

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Route table settings

Name - *optional*
Create a tag with a key of 'Name' and a value that you specify.

VPC
The VPC to use for this route table.

Tags
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 - <i>optional</i>
<input type="text" value="Name"/>	<input type="text" value="proj_public_rt"/>

[Add new tag](#)
You can add 49 more tags.

[Cancel](#) [Create route table](#)

- Click on create route table, In route table settings, create a private route table by giving the name and select the VPC that was created earlier. Click on create route table.

Screenshot of the AWS VPC Route Tables creation interface for a private route table:

Create route table [Info](#)

A route table specifies how packets are forwarded between the subnets within your VPC, the internet, and your VPN connection.

Route table settings

Name - *optional*
Create a tag with a key of 'Name' and a value that you specify.

VPC
The VPC to use for this route table.

Tags
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 - <i>optional</i>
<input type="text" value="Name"/>	<input type="text" value="proj_private_rt"/>

[Add new tag](#)
You can add 49 more tags.

[Cancel](#) [Create route table](#)

- Select the public route table and edit the routes by adding the internet gateway as target and destination is 0.0.0.0/0

Screenshot of the AWS VPC Route Tables edit routes interface:

Edit routes

Destination	Target	Status	Propagated
10.0.0.0/16	<input type="text" value="local"/> X	<input checked="" type="checkbox"/> Active	No
<input type="text" value="0.0.0.0"/> X	<input type="text" value="igw-0aa65075d0c7215b8"/> X	-	Remove

[Add route](#)

[Cancel](#) [Preview](#) [Save changes](#)

- Select the public route table and edit the subnet associations , add the public subnet. Click on save associations.

The screenshot shows the AWS VPC Route Tables - Edit subnet associations interface. At the top, there's a navigation bar with 'VPC > Route tables > rtb-09a472c59d4da5e3d > Edit subnet associations'. Below it, a title 'Edit subnet associations' and a subtitle 'Change which subnets are associated with this route table.' A table titled 'Available subnets (1/3)' lists three subnets: 'private_sub1', 'private_sub2', and 'public_sub1'. The 'public_sub1' row is selected, indicated by a checked checkbox. In the 'Selected subnets' section, the same subnet is listed with a blue border around it. At the bottom right are 'Cancel' and 'Save associations' buttons.

- Select the private route table ,and edit the routes , add the nat gateway to allow internet access.

The screenshot shows the AWS VPC Route Tables - Edit routes interface. At the top, there's a navigation bar with 'VPC > Route tables > rtb-0f8cc789b701197a6 > Edit routes'. Below it, a title 'Edit routes'. A table lists two routes: one for '10.0.0.0/16' target 'local' and another for '0.0.0.0/0' target 'nat-0676d1cd4c55b68b8'. Both routes are marked as 'Active'. At the bottom left is an 'Add route' button, and at the bottom right are 'Cancel', 'Preview', and 'Save changes' buttons.

- Select the private route table and go to subnet associations and associate the private subnets , click on save associations.

VPC > Route tables > rtb-Of8cc789b701197a6 > Edit subnet associations

Edit subnet associations

Change which subnets are associated with this route table.

Available subnets (2/3)					
Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID	
<input checked="" type="checkbox"/> private_sub1	subnet-07cb9ba615922aac0	10.0.2.0/24	-	Main (rtb-0143c4c51829c87b3)	
<input checked="" type="checkbox"/> private_sub2	subnet-061062e7ff823e1e8	10.0.3.0/24	-	Main (rtb-0143c4c51829c87b3)	
<input type="checkbox"/> public_sub1	subnet-0ac9ea3ffdfaf311a1	10.0.1.0/24	-	rtb-09a472c59d4da5e3d / proj_public_rt	

Selected subnets

- subnet-07cb9ba615922aac0 / private_sub1
- subnet-061062e7ff823e1e8 / private_sub2

Cancel **Save associations**

❖ Creating EC2 instances

- Go to the EC2 dashboard and create the instances.
- In the application tier , we have to launch the EC2 instance to deploy the web application.
- Create a public instance and give the name of as proj_public_server and select the ubuntu image.
- Create a new keypair.
- In the network settings , select the VPC that you have created and also select public subnet .
- Create the security group , the inbound rules should include the ssh to connect to the instance, http and https to request the query and get the response in web application.
- Create the instance.

EC2 > Instances > Launch an instance

Launch an instance Info

Amazon EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags Info

Name: proj_public_server Add additional tags

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Quick Start

Amazon Linux macOS Ubuntu Windows Red Hat S Review commands

Summary

Number of instances Info
1

Software Image (AMI)
Canonical, Ubuntu, 22.04 LTS, read more
ami-0a72af0f5d27b49ccb

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

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 usage on free tier

Cancel **Launch instance**

Application and OS Images (Amazon Machine Image) Info

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below.

Quick Start

Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type

Canonical, Ubuntu, 22.04 LTS, amd64 jammy image build on 2023-03-25

Description

Architecture: 64-bit (x86) | AMI ID: ami-072af05d27b49ccb | Verified provider

Summary

Number of instances: 1

Software Image (AMI): Canonical, Ubuntu, 22.04 LTS, ... read more

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Launch instance

(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 usage on free tier)

Create key pair

Key pairs allow you to connect to your instance securely.

Enter the name of the key pair below. When prompted, store the private key in a secure and accessible location on your computer. You will need it later to connect to your instance. [Learn more](#)

Key pair name: projKey

The name can include up to 255 ASCII characters. It can't include leading or trailing spaces.

Key pair type:

- RSA RSA encrypted private and public key pair
- ED25519 ED25519 encrypted private and public key pair (Not supported for Windows instances)

Private key file format:

- .pem For use with OpenSSH
- .ppk For use with PuTTY

Launch instance

Key pair (login) Info

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required: projKey

Network settings Info

VPC - required Info

vpc-05a17e3d4b6d1463a (proj_vpc) 10.0.0.0/16

Subnet Info

subnet-0ac9ea3ffad311a1 public_sub1

Auto-assign public IP Info

Enable

Firewall (security groups) Info

A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group Select existing security group

Summary

Number of instances: 1

Software Image (AMI): Canonical, Ubuntu, 22.04 LTS, ... read more

Virtual server type (instance type): t2.micro

Firewall (security group): New security group

Storage (volumes): 1 volume(s) - 8 GiB

Launch instance

(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 usage on free tier)

The screenshots show the AWS CloudFormation Launch Wizard interface. The left screenshot focuses on the 'Inbound security groups rules' section, where two rules are defined:

- Security group rule 1 (TCP, 22, 0.0.0.0/0)**: Type: ssh, Protocol: TCP, Port range: 22. Source type: Anywhere.
- Security group rule 2 (All, All, 0.0.0.0/0)**: Type: All traffic, Protocol: All, Port range: All. Source type: Anywhere.

A warning message is displayed: "⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only." The right screenshot shows the 'Summary' section with the following details:

- Number of instances: 1
- Software Image (AMI): Canonical, Ubuntu, 22.04 LTS
- Virtual server type (instance type): t2.micro
- Storage (volumes): 1 volume(s) - 8 GiB

A tooltip for the free tier is shown: "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 usage on free tier".

- Create another instance and make it private , because we should not give the public access to the Database that we would connect in further step.
- Give the name for the private instance and select ubuntu , select the existing key pair.
- In the network settings, select the VPC that you have created earlier and select the public subnet
- Create the new security group, it should include mysql inbound rule to communicate with database tier and include the ssh rule from the public server security group as source.Include the HTTP inbound rule to resend the response from database tier to the presentation tier.

Launch an instance

AWS EC2 allows you to create virtual machines, or instances, that run on the AWS Cloud. Quickly get started by following the simple steps below.

Name and tags

Name: proj_private_server

Software Image (AMI)

Canonical, Ubuntu, 22.04 LTS, ...read more
ami-0a72af05d27b49ccb

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

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 usage on free tier

Launch instance

Application and OS Images (Amazon Machine Image)

An AMI is a template that contains the software configuration (operating system, application server, and applications) required to launch your instance. Search or Browse for AMIs if you don't see what you are looking for below

Quick Start

Recent AMIs: Amazon Linux, macOS, Ubuntu, Windows, Red Hat

Amazon Machine Image (AMI)

Ubuntu Server 22.04 LTS (HVM), SSD Volume Type
Free tier eligible
ami-0a72af05d27b49ccb (64-bit (x86)) / ami-0dd28c803f9318319 (64-bit (Arm))
Virtualization: hvm ENA enabled: true Root device type: ebs

Description

Canonical, Ubuntu, 22.04 LTS, amd64 jammy image build on 2023-03-25

Architecture

64-bit (x86) AMI ID: ami-0a72af05d27b49ccb Verified provider

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 usage on free tier

Launch instance

Instance type

Instance type: t2.micro
Family: t2 - 1 vCPU - 1 GiB Memory
On-Demand Windows pricing: 0.0192 USD per Hour
On-Demand RHEL pricing: 0.0746 USD per Hour
On-Demand Linux pricing: 0.0146 USD per Hour
On-Demand SUSE pricing: 0.0146 USD per Hour

All generations

Key pair (login)

You can use a key pair to securely connect to your instance. Ensure that you have access to the selected key pair before you launch the instance.

Key pair name - required: projKey

Network settings

Edit

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 usage on free tier

Summary

Number of instances: 1

Software Image (AMI)

Canonical, Ubuntu, 22.04 LTS, ...read more
ami-0a72af05d27b49ccb

Virtual server type (instance type)

t2.micro

Firewall (security group)

New security group

Storage (volumes)

1 volume(s) - 8 GiB

Network settings

VPC - required info
vpc-05a17e3d4b6d1463a (proj_vpc)
10.0.0.0/16

Subnet Info
subnet-07cb9ba615922aac0
VPC: vpc-05a17e3d4b6d1463a Owner: 899064791528 Availability Zone: ap-southeast-1a IP addresses available: 251 CIDR: 10.0.2.0/24

Create new subnet

Auto-assign public IP Info
Disable

Firewall (security groups) Info
A security group is a set of firewall rules that control the traffic for your instance. Add rules to allow specific traffic to reach your instance.

Create security group
Select existing security group

Security group name - required
proj_private_sg

This security group will be added to all network interfaces. The name can't be edited after the security group is created. Max length is 255 characters. Valid characters: a-z, A-Z, 0-9, spaces, and _-/@[]+=&;!\$^

Description - required Info
proj_private_sg

Inbound security groups rules

Security group rule 1 (TCP, 22, sg-0c530a528ef15eeaca)

Remove

Summary

Number of instances Info
1

Software Image (AMI)
Canonical, Ubuntu, 22.04 LTS, ...read more
ami-0a72af05d27b49cc

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

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 usage on free tier

Cancel Launch instance Review commands

proj_private_sg

Inbound security groups rules

Security group rule 1 (TCP, 22, sg-0c530a528ef15eeaca)

Remove

Type Info	Protocol Info	Port range Info
ssh	TCP	22

Source type Info
Custom

Add CIDR, prefix list or security
e.g. SSH for admin desktop

sg-0c530a528ef15eeaca X

Security group rule 2 (TCP, 3306, 0.0.0.0/0)

Remove

Type Info	Protocol Info	Port range Info
MySQL/Aurora	TCP	3306

Source type Info
Anywhere

Add CIDR, prefix list or security
e.g. SSH for admin desktop

0.0.0.0/0 X

Security group rule 3 (TCP, 80, 0.0.0.0/0)

Remove

Type Info	Protocol Info	Port range Info
HTTP	TCP	80

Source type Info
Anywhere

Add CIDR, prefix list or security
e.g. SSH for admin desktop

0.0.0.0/0 X

Summary

Number of instances Info
1

Software Image (AMI)
Canonical, Ubuntu, 22.04 LTS, ...read more
ami-0a72af05d27b49cc

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

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 usage on free tier

Cancel Launch instance Review commands

proj_private_sg

Security group rule 3 (TCP, 80, 0.0.0.0/0)

Remove

Type Info	Protocol Info	Port range Info
HTTP	TCP	80

Source type Info
Anywhere

Add CIDR, prefix list or security
e.g. SSH for admin desktop

0.0.0.0/0 X

⚠ Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Add security group rule

Advanced network configuration

Configure storage Info Advanced

1x 8 GiB gp2 Root volume (Not encrypted)

Free tier eligible customers can get up to 20 GB of EBS General Purpose (SSD) or Magnetic storage

Summary

Number of instances Info
1

Software Image (AMI)
Canonical, Ubuntu, 22.04 LTS, ...read more
ami-0a72af05d27b49cc

Virtual server type (instance type)
t2.micro

Firewall (security group)
New security group

Storage (volumes)
1 volume(s) - 8 GiB

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 usage on free tier

Cancel Launch instance Review commands

The screenshot shows the AWS EC2 Instances Launch an instance page. At the top, there's a success message: "Successfully initiated launch of instance (i-0541d2bc3288bbc5d)". Below it is a "Launch log" table:

	Succeeded
Initializing requests	Succeeded
Creating security groups	Succeeded
Creating security group rules	Succeeded
Launch initiation	Succeeded

Below the log is a "Next Steps" section with several options:

- Create billing and free tier usage alerts
- Connect to your instance
- Connect an RDS database
- Create EBS snapshot policy

❖ Creating the Database

- Go to RDS
- Click on create database.
- Choose the standard create , that means you set all of the configurations options, including ones for availability , security, backups, and maintainance.
- Choose the MySQL engine option.
- Choose the free tier template to develop the application.
- In settings, give the name for the DB instance.
- Give the master username for the database and give the password to login to mysql database.

The screenshot shows the AWS RDS Create database page for MySQL. At the top, there's a message: "We listened to your feedback! Now, create a database with a single click using our pre-built configurations! Or choose your own configurations." Below it is a "Create database" section:

Choose a database creation method [Info](#)

Standard create You set all of the configuration options, including ones for availability, security, backups, and maintenance.

Easy create Use recommended best-practice configurations. Some configuration options can be changed after the database is created.

Engine options

Engine type [Info](#)

Aurora (MySQL Compatible)

Aurora (PostgreSQL Compatible)

MySQL

On the right side, there's a sidebar with information about MySQL:

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

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

Edition

- MySQL Community

Known issues/limitations
Review the Known issues/limitations [\[Link\]](#) to learn about potential compatibility issues with specific database versions.

Hide filters

Show versions that support the Multi-AZ DB cluster [Info](#)
Create A Multi-AZ DB cluster with one primary DB instance and two readable standby DB instances. Multi-AZ DB clusters provide up to 2x faster transaction commit latency and automatic failover in typically under 35 seconds.

Show versions that support the Amazon RDS Optimized Writes [Info](#)
Amazon RDS Optimized Writes improves write throughput by up to 2x at no additional cost.

Engine Version
MySQL 8.0.32

Templates
Choose a sample template to meet your use case.

- Production
Use defaults for high availability and fast, consistent performance.
- Dev/Test
This instance is intended for development use outside of a production environment.
- Free tier
Use RDS Free Tier to develop new applications, test existing applications, or gain hands-on experience with Amazon RDS.

MySQL

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

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

Availability and durability

Deployment options [Info](#)
The deployment options below are limited to those supported by the engine you selected above.

- Multi-AZ DB Cluster - new
Creates a DB cluster with a primary DB instance and two readable standby DB instances, with each DB instance in a different Availability Zone (AZ). Provides high availability, data redundancy and increases capacity to serve read workloads.
- Multi-AZ DB instance (not supported for Multi-AZ DB cluster snapshot)
Creates a primary DB instance and a standby DB instance in a different AZ. Provides high availability and data redundancy, but the standby DB instance doesn't support connections for read workloads.
- Single DB Instance (not supported for Multi-AZ DB cluster snapshot)
Creates a single DB instance with no standby DB instances.

Settings

DB instance identifier [Info](#)
Type a name for your DB instance. The name must be unique across all DB instances owned by your AWS account in the current AWS Region.
database-1

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

Credentials Settings

Master username [Info](#)
Type a login ID for the master user of your DB instance.
admin

MySQL

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

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

The DB instance identifier is case-insensitive, but is stored as all lowercase (as in "mydbinstance"). Constraints: 1 to 60 alphanumeric characters or hyphens. First character must be a letter. Can't contain two consecutive hyphens. Can't end with a hyphen.

Credentials Settings

Master username [Info](#)
Type a login ID for the master user of your DB instance.
admin

1 to 16 alphanumeric characters. First character must be a letter.

Manage master credentials in AWS Secrets Manager
Manage master user credentials in Secrets Manager. RDS can generate a password for you and manage it throughout its lifecycle.

If you manage the master user credentials in Secrets Manager, some RDS features aren't supported.
[Learn more](#)

Auto generate a password
Amazon RDS can generate a password for you, or you can specify your own password.

Master password [Info](#)

Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), ' (single quote), " (double quote) and @ (at sign).

Confirm master password [Info](#)

MySQL

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

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

Instance configuration

The DB instance configuration options below are limited to those supported by the engine that you selected above.

Amazon RDS Optimized Writes - new [Info](#)

Show instance classes that support Amazon RDS Optimized Writes

DB instance class [Info](#)

- Standard classes (includes m classes)
- Memory optimized classes (includes r and x classes)
- Burstable classes (includes t classes)**

db.t3.micro
2 vCPUs 1 GiB RAM Network: 2,085 Mbps

Include previous generation classes

Storage

Storage type [Info](#)

General Purpose SSD (gp2)
Baseline performance determined by volume size

MySQL

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

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

Include previous generation classes

Storage

Storage type [Info](#)

General Purpose SSD (gp2)
Baseline performance determined by volume size

Allocated storage [Info](#)

20 GiB
The minimum value is 20 GiB and the maximum value is 6,144 GiB

Storage autoscaling [Info](#)

Provides dynamic scaling support for your database's storage based on your application's needs.

Enable storage autoscaling
Enabling this feature will allow the storage to increase after the specified threshold is exceeded.

Maximum storage threshold [Info](#)

Charges will apply when your database autoscales to the specified threshold
1000 GiB
The minimum value is 22 GiB and the maximum value is 6,144 GiB

MySQL

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

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

- Choose the VPC and create the DB subnet group, this should allow only mysql from any source.

Connectivity [Info](#)

Compute resource
Choose whether to set up a connection to a compute resource for this database. Setting up a connection will automatically change connectivity settings so that the compute resource can connect to this database.

Don't connect to an EC2 compute resource
Don't set up a connection to a compute resource for this database. You can manually set up a connection to a compute resource later.

Connect to an EC2 compute resource
Set up a connection to an EC2 compute resource for this database.

Network type [Info](#)
To use dual-stack mode, make sure that you associate an IPv6 CIDR block with a subnet in the VPC you specify.

IPv4
Your resources can communicate only over the IPv4 addressing protocol.

Dual-stack mode
Your resources can communicate over IPv4, IPv6, or both.

Virtual private cloud (VPC) [Info](#)
Choose the VPC. The VPC defines the virtual networking environment for this DB instance.
proj_vpc (vpc-05a17e3d4b6d1465a)
Only VPCs with a corresponding DB subnet group are listed.

After a database is created, you can't change its VPC.

DB subnet group [Info](#)
Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB instance can use in the VPC that you selected.

Public access [Info](#)

MySQL

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

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

aws Services Search [Alt+S]

After a database is created, you can't change its VPC.

DB subnet group [Info](#)
Choose the DB subnet group. The DB subnet group defines which subnets and IP ranges the DB instance can use in the VPC that you selected.
[Create new DB Subnet Group](#)

Public access [Info](#)
 Yes
RDS assigns a public IP address to the database. Amazon EC2 instances and other resources outside of the VPC can connect to your database. Resources inside the VPC can also connect to the database. Choose one or more VPC security groups that specify which resources can connect to the database.
 No
RDS doesn't assign a public IP address to the database. Only Amazon EC2 instances and other resources inside the VPC can connect to your database. Choose one or more VPC security groups that specify which resources can connect to the database.

VPC security group (firewall) [Info](#)
Choose one or more VPC security groups to allow access to your database. Make sure that the security group rules allow the appropriate incoming traffic.
 Choose existing
Choose existing VPC security groups
 Create new
Create new VPC security group

New VPC security group name
proj_db_sg

Availability Zone [Info](#)
ap-southeast-1

RDS Proxy

MySQL

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

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

aws Services Search [Alt+S]

RDS Proxy
RDS Proxy is a fully managed, highly available database proxy that improves application scalability, resiliency, and security.
 [Create an RDS Proxy](#) [Info](#)
RDS automatically creates an IAM role and a Secrets Manager secret for the proxy. RDS Proxy has additional costs. For more information, see [Amazon RDS pricing](#).

Certificate authority - optional [Info](#)
Using a server certificate provides an extra layer of security by validating that the connection is being made to an Amazon database. It does so by checking the server certificate that is automatically installed on all databases that you provision.
rds-ca-2019 (default)
If you don't select a certificate authority, RDS chooses one for you.

Additional configuration

Database authentication

Database authentication options [Info](#)

Password authentication
Authenticates using database passwords.

Password and IAM database authentication
Authenticates using the database password and user credentials through AWS IAM users and roles.

Password and Kerberos authentication
Choose a directory in which you want to allow authorized users to authenticate with this DB instance using Kerberos Authentication.

MySQL

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

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

aws Services Search [Alt+S]

Monitoring
 [Enable Enhanced monitoring](#)
Enabling Enhanced monitoring metrics are useful when you want to see how different processes or threads use the CPU.

Additional configuration
Database options, encryption turned on, backup turned on, backtrack turned off, maintenance, CloudWatch Logs, delete protection turned off.

Estimated monthly costs

The Amazon RDS Free Tier is available to you for 12 months. Each calendar month, the free tier will allow you to use the Amazon RDS resources listed below for free:

- 750 hrs of Amazon RDS in a Single-AZ db.t2.micro, db.t3.micro or db.t4g.micro Instance.
- 20 GB of General Purpose Storage (SSD).
- 20 GB for automated backup storage and any user-initiated DB Snapshots.

[Learn more about AWS Free Tier](#).

When your free usage expires or if your application use exceeds the free usage tiers, you simply pay standard, pay-as-you-go service rates as described in the [Amazon RDS Pricing page](#).

MySQL

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

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

Note You are responsible for ensuring that you have all of the necessary rights for any third-party products or services that you use with AWS services.

Create database

The screenshot shows the AWS EC2 Security Groups interface for editing inbound rules. The URL is [EC2 > Security Groups > sg-000a072cc3c3960b4 - proj_db_sg > Edit inbound rules](#). The page title is "Edit inbound rules". A note says "Inbound rules control the incoming traffic that's allowed to reach the instance." Below is a table with one row:

Security group rule ID	Type	Protocol	Port range	Source	Description - optional
sgr-037f9a167156a1d20	MySQL/Aurora	TCP	3306	Anywhere	0.0.0.0/0

Buttons at the bottom include "Add rule", "Cancel", "Preview changes", and "Save rules".

This screenshot is identical to the one above, showing the same inbound rule configuration for the security group sg-000a072cc3c3960b4.

- Connect to the public instance from EC2 console.
- In the public instance, install the required software such as mysql and php.
- The commands used are

```

aws | Services | Search [Alt+S]
System load: 0.080078125 Processes: 96
Usage of /: 20.1% of 7.57GB Users logged in: 0
Memory usage: 20% IPv4 address for eth0: 10.0.1.46
Swap usage: 0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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

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

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

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

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

ubuntu@ip-10-0-1-46:~$ 

```

i-058749e47b8280025 (proj_public_server)
PublicIPs: 13.213.35.155 PrivateIPs: 10.0.1.46

\$sudo apt update

- "sudo apt update" is a command used in Linux-based operating systems to update the package lists from repositories.
- It requires administrative privileges to execute, hence the "sudo" (superuser do) prefix.
- Running this command ensures that the system has the latest information about available software updates and helps to keep the system secure and up to date.

```

ubuntu@ip-10-0-1-46:~$ sudo apt update
Hit:1 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy InRelease
Get:2 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:3 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-backports InRelease [108 kB]
Get:4 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 Packages [14.1 MB]
Get:5 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Get:6 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy/universe Translation-en [5652 kB]
Get:7 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy/universe amd64 c-n-f Metadata [286 kB]
Get:8 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 Packages [217 kB]
Get:9 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse Translation-en [112 kB]
Get:10 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy/multiverse amd64 c-n-f Metadata [8372 B]
Get:11 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [990 kB]
Get:12 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [210 kB]
Get:13 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 c-n-f Metadata [13.9 kB]
Get:14 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 Packages [743 kB]
Get:15 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted Translation-en [115 kB]
Get:16 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/restricted amd64 c-n-f Metadata [588 B]
Get:17 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [898 kB]
Get:18 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe Translation-en [180 kB]
Get:19 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/universe amd64 c-n-f Metadata [18.5 kB]
Get:20 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 Packages [24.1 kB]
Get:21 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse Translation-en [6312 B]
Get:22 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/multiverse amd64 c-n-f Metadata [444 B]
Get:23 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 Packages [40.6 kB]
Get:24 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main Translation-en [9800 B]
Get:25 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/main amd64 c-n-f Metadata [388 B]
Get:26 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-backports/restricted amd64 c-n-f Metadata [116 B]

ubuntu@ip-10-0-1-46:~$ 

```

i-058749e47b8280025 (proj_public_server)
PublicIPs: 13.213.35.155 PrivateIPs: 10.0.1.46

\$sudo apt install php libapache2-mod-php php-mysql

- The command "sudo apt install php libapache2-mod-php php-mysql" is used in Linux-based operating systems to install PHP programming language and its modules required for running PHP applications on Apache web server.

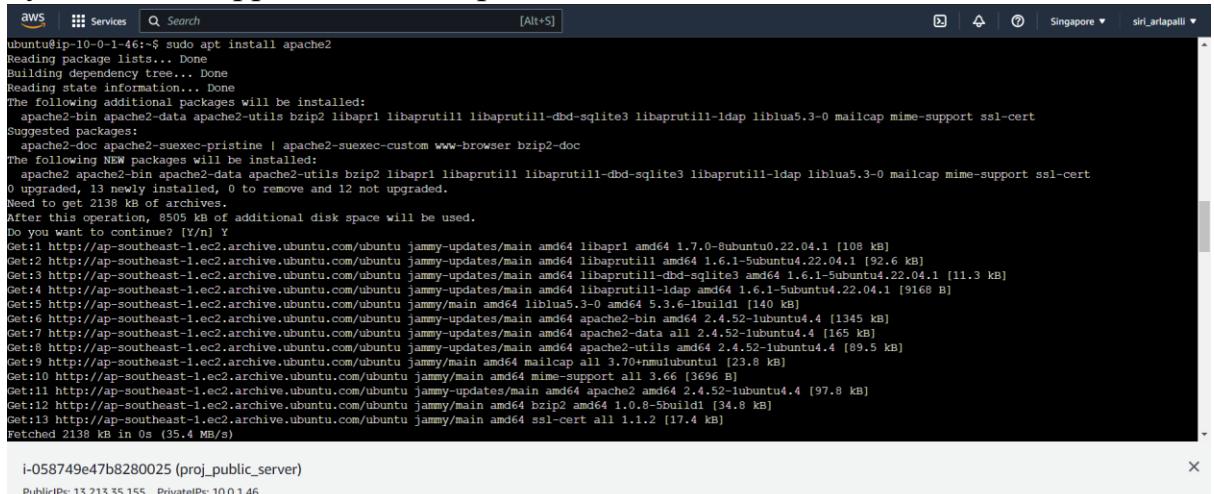
Here is what each package does:

php: Installs PHP programming language.

libapache2-mod-php: Installs the Apache module for PHP, which allows Apache to process PHP scripts.

php-mysql: Installs MySQL module for PHP, which allows PHP to interact with MySQL databases.

This command requires administrative privileges to execute, hence the "sudo" (superuser do) prefix. Running this command will download and install the required packages and their dependencies from the package repository. Once installed, you can start using PHP to develop and run dynamic web applications on Apache web server.



```
aws Services Search [Alt+S] Singapore siri_artapalli
ubuntu:~$ sudo apt install apache2
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apache2-bin apache2-data apache2-utils bzip2 libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.3-0 mailcap mime-support ssl-cert
Suggested packages:
  apache2-doc apache2-suexec-pristine | apache2-suexec-custom www-browser bzip2-doc
The following NEW packages will be installed:
  apache2 apache2-bin apache2-data apache2-utils bzip2 libapr1 libaprutil1 libaprutil1-dbd-sqlite3 libaprutil1-ldap liblua5.3-0 mailcap mime-support ssl-cert
0 upgraded, 13 newly installed, 0 to remove and 12 not upgraded.
Need to get 2138 kB of archives.
After this operation, 8505 kB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libapr1 amd64 1.7.0-8ubuntu0.22.04.1 [108 kB]
Get:2 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1 amd64 1.6.1-Subuntu4.22.04.1 [92.6 kB]
Get:3 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1-dbd-sqlite3 amd64 1.6.1-Subuntu4.22.04.1 [11.3 kB]
Get:4 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libaprutil1-ldap amd64 1.6.1-Subuntu4.22.04.1 [9168 kB]
Get:5 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 liblua5.3-0 amd64 5.3.6-1build1 [140 kB]
Get:6 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-bin amd64 2.4.52-ubuntu4.4 [1345 kB]
Get:7 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-data all 2.4.52-ubuntu4.4 [165 kB]
Get:8 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2-utils amd64 2.4.52-ubuntu4.4 [89.5 kB]
Get:9 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 mailcap all 3.70+mmu1ubuntu1 [23.8 kB]
Get:10 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 mime-support all 3.66 [3696 B]
Get:11 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 apache2 amd64 2.4.52-ubuntu4.4 [97.8 kB]
Get:12 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 bzip2 amd64 1.0.8-5build1 [34.8 kB]
Get:13 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 ssl-cert all 1.1.2 [17.4 kB]
Fetched 2138 kB in 0s (35.4 MB/s)

i-058749e47b8280025 (proj_public_server)
PublicIPs: 13.213.35.155 PrivateIPs: 10.0.1.46
```

\$sudo apt install mysql-client-core-8.0

The "sudo apt install mysql-client-core-8.0" command is used in Linux-based operating systems to install the MySQL client core package, which provides the necessary components for connecting to a MySQL server and executing SQL queries.

The package version "mysql-client-core-8.0" corresponds to the MySQL version 8.0. When you run this command with administrative privileges (i.e., using the "sudo" command), the package manager will download and install the MySQL client core package and its dependencies from the package repository.

After installation, you can use the "mysql" command in the terminal or command prompt to connect to a MySQL server and execute SQL queries. However, note that this command does not install the MySQL server itself, only the client package required for connecting to it.

```

ubuntu@ip-10-0-1-46:~$ sudo apt install php libapache2-mod-php php-mysql
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libapache2-mod-php8.1 php-common php8.1-cli php8.1-common php8.1-mysql php8.1-opcache php8.1-readline
Suggested packages:
  php-pear
The following NEW packages will be installed:
  libapache2-mod-php libapache2-mod-php8.1 php php-common php-mysql php8.1 php8.1-cli php8.1-common php8.1-mysql php8.1-opcache php8.1-readline
0 upgraded, 11 newly installed, 0 to remove and 12 not upgraded.
Need to get 9262 kB of archives.
After this operation, 21.8 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 php-common all 2:92ubuntu1 [12.4 kB]
Get:2 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 php8.1-common amd64 8.1.2-1ubuntu2.11 [1126 kB]
Get:3 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 php8.1-opcache amd64 8.1.2-1ubuntu2.11 [365 kB]
Get:4 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 php8.1-readline amd64 8.1.2-1ubuntu2.11 [13.5 kB]
Get:5 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 php8.1-cli amd64 8.1.2-1ubuntu2.11 [1833 kB]
Get:6 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libapache2-mod-php8.1 amd64 8.1.2-1ubuntu2.11 [1765 kB]
Get:7 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libapache2-mod-php all 2:8.1+92ubuntu1 [2898 B]
Get:8 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 php8.1 all 8.1.2-1ubuntu2.11 [9150 B]
Get:9 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 php all 2:8.1+92ubuntu1 [2756 B]
Get:10 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 php8.1-mysql amd64 8.1.2-1ubuntu2.11 [130 kB]
Get:11 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 php-mysql all 2:8.1+92ubuntu1 [1034 B]
Fetched 5262 kB in 12s (442 kB/s)
Selecting previously unselected package php-common.
(Reading database ... 64426 files and directories currently installed.)

```

i-058749e47b8280025 (proj_public_server)

\$mysql –version

The "mysql --version" command is used to check the version of MySQL database software installed on a system. When executed in a terminal or command prompt, it will display the version number of MySQL installed on the system.

```

ubuntu@ip-10-0-1-46:~$ mysql --version
Command 'mysql' not found, but can be installed with:
sudo apt install mysql-client-core-8.0 # version 8.0.32-0ubuntu0.22.04.2, or
sudo apt install mariadb-client-core-10.6 # version 1:10.6.12-0ubuntu0.22.04.1
ubuntu@ip-10-0-1-46:~$ C
ubuntu@ip-10-0-1-46:~$ sudo apt install mysql-client-core-8.0
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  mysql-client-core-8.0
0 upgraded, 1 newly installed, 0 to remove and 12 not upgraded.
Need to get 2677 kB of additional disk space will be used.
Get:1 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 mysql-client-core-8.0 amd64 8.0.32-0ubuntu0.22.04.2 [2677 kB]
Fetched 2677 kB in 0s (30.5 MB/s)
Selecting previously unselected package mysql-client-core-8.0.
(Reading database ... 64575 files and directories currently installed.)
Preparing to unpack .../mysql-client-core-8.0_8.0.32-0ubuntu0.22.04.2_amd64.deb ...
Unpacking mysql-client-core-8.0 (8.0.32-0ubuntu0.22.04.2) ...
Setting up mysql-client-core-8.0 (8.0.32-0ubuntu0.22.04.2) ...
Processing triggers for man-db (2.10.2-1) ...
Scanning processes...
Scanning linux images...

Running kernel seems to be up-to-date.

No services need to be restarted.

```

i-058749e47b8280025 (proj_public_server)

PublicIPs: 13.213.35.155 PrivateIPs: 10.0.1.46

\$sudo rm index.html

By default in var/www/html you have the index.html, for the front end web page creation remove the index.html using the rm command and create the file index.html using the command **sudo nano index.html**

```

ubuntu@ip-10-0-1-46:~$ mysql --version
mysql Ver 8.0.32-0ubuntu0.22.04.2 for Linux on x86_64 ((Ubuntu))
ubuntu@ip-10-0-1-46:~$ sudo su
root@ip-10-0-1-46:/home/ubuntu$ cd /
root@ip-10-0-1-46:/# cd var/www/html
root@ip-10-0-1-46:/var/www/html# ls
index.html
root@ip-10-0-1-46:/var/www/html# sudo rm index.html
root@ip-10-0-1-46:/var/www/html# sudo nano index.html

```

i-058749e47b8280025 (proj_public_server)

PublicIPs: 13.213.35.155 PrivateIPs: 10.0.1.46

```

root@ip-10-0-1-46:/var/www/html# sudo nano index.html
root@ip-10-0-1-46:/var/www/html# cat index.html
<!DOCTYPE html>
<html>
<head>
    <title>Login Page</title>
    <style>
        form{
            margin-top:3%;
            margin-bottom: 7%;
        }
        h1{
            margin-top: 10%;
        }
        div{
            border:2px solid black;
            width: 40%;
            margin-top: 15px;
            background-color: white;
        }
    </style>
</head>
<body bgcolor="#e6b3ff">
<center>
    <div class="row">
        <h1 style="color:purple">LOGIN</h1>
        <form action="authenticate.php" method="post">
            <label for="username"><b>Username:</b></label>

```

i-058749e47b8280025 (proj_public_server)
PublicIPs: 13.213.35.155 PrivateIPs: 10.0.1.46

Create another file authenticate.php for the back end .

```

root@ip-10-0-1-46:/var/www/html# sudo nano authenticate.php
i-058749e47b8280025 (proj_public_server)
PublicIPs: 13.213.35.155 PrivateIPs: 10.0.1.46

GNU nano 6.2
authenticate.php *
<?php
// Retrieve user input
$username = $_POST['username'];
$password = $_POST['password'];

// Connect to RDS
$servername = "database-1.cxycc96jpe13k.ap-southeast-1.rds.amazonaws.com";
$db_username = "admin";
$db_password = "Siri12345";
$dbname = "mydb";
$conn = new mysqli($servername, $db_username, $db_password, $dbname);

// Check connection
if ($conn->connect_error) {
    die("Connection failed: " . $conn->connect_error);
}

// Prepare SQL statement
$stmt = $conn->prepare("SELECT * FROM users WHERE username=? AND password=?");
$stmt->bind_param("ss", $username, $password);
$stmt->execute();
$result = $stmt->get_result();
// Check if user exists
if ($result->num_rows > 0) {

```

i-058749e47b8280025 (proj_public_server)
PublicIPs: 13.213.35.155 PrivateIPs: 10.0.1.46

```

root@ip-10-0-1-46:/var/www/html# sudo nano /etc/apache2/mods-enabled/dir.conf
root@ip-10-0-1-46:/var/www/html# i-058749e47b8280025 (proj_public_server)
PublicIPs: 13.213.35.155 PrivateIPs: 10.0.1.46

GNU nano 6.2
/etc/apache2/mods-enabled/dir.conf
<IfModule mod_dir.c>
    DirectoryIndex index.html index.cgi index.pl index.php index.xhtml index.htm
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet

```

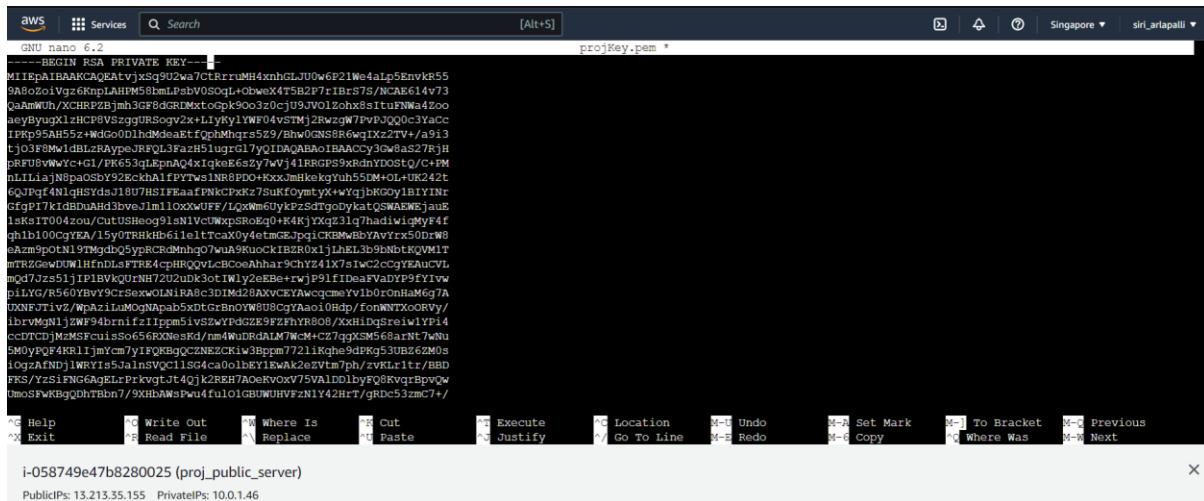
```

aws Services Search [Alt+S] Singapore siri_arlapalli
GNU nano 6.2
/etc/apache2/mods-enabled/dir.conf
<IfModule mod_dir.c>
    DirectoryIndex index.html index.cgi index.pl index.php index.xhtml index.htm
</IfModule>

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
^G Help      ^D Write Out   ^W Where Is   ^R Cut          ^I Execute   ^C Location   M-U Undo   M-Z Set Mark   M-J To Bracket   M-Q Previous
^X Exit      ^R Read File   ^\ Replace    ^T Paste         ^J Justify   ^Y Go To Line  M-E Redo   M-B Copy      M-Q Where Was    M-W Next

```

i-058749e47b8280025 (proj_public_server)
PublicIPs: 13.213.35.155 PrivateIPs: 10.0.1.46



```

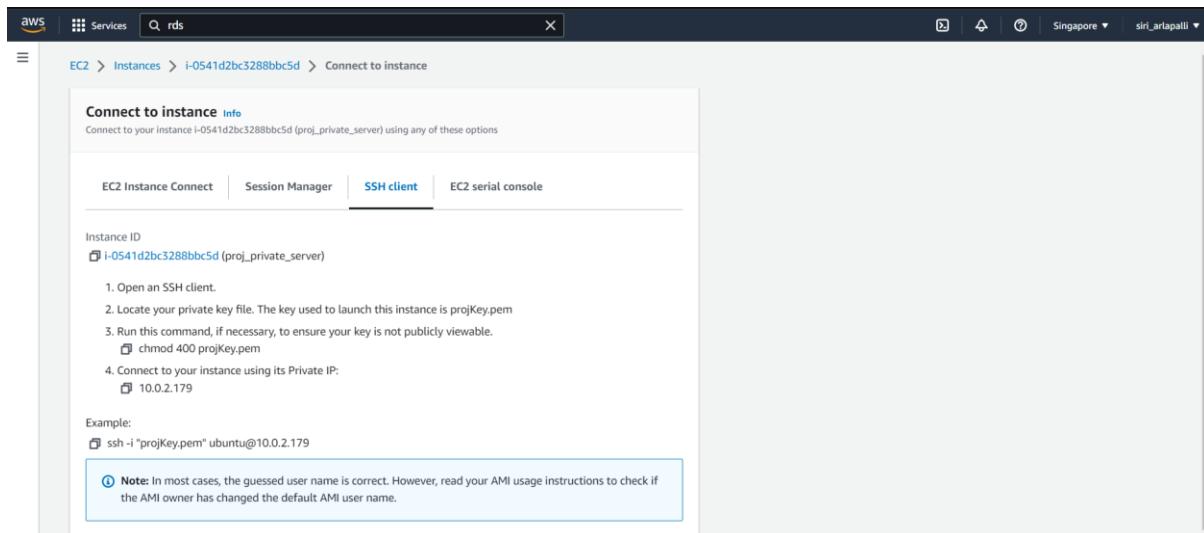
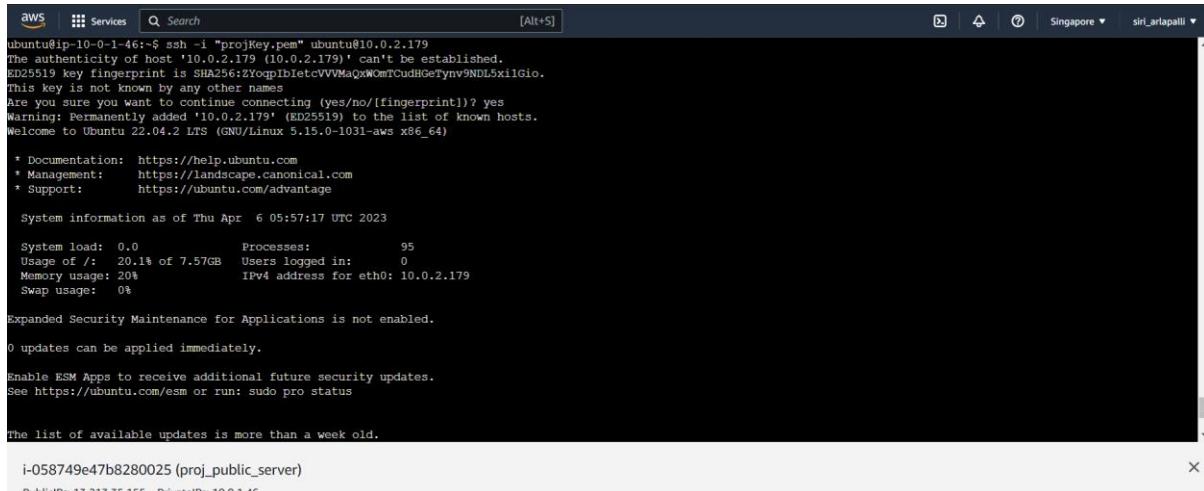
GNU nano 6.2
[Alt+S] projKey.pem *
-----BEGIN RSA PRIVATE KEY-----
MIIEPAIBAAKCAQEAtVixsq9U2wA7CTrruM4xnhGLJU0w6P21We4aLp5EnvKR55
9A8oZoiVgzGnpIaHP50mlnlabt0VO3Cqj+obwex475B2P71RbS75/NCAE614v73
QaAmWb/XCR12Bjhnh3C8dGDRmtxtccop9eo3z0cj9UJV0lZchx8sItuNWA4Zoo
aeYByugXzlICP8VSzgJU8Scopy2s+1ryKy1WF04vSTMj2rzgWPvP3Q0c3Yacc
1P8p95AH552+wDgoD1hdMeaRtfghMhqr5z9/bhw0CN58R6wqjXz-2TV+/a913
Lj03f8Mml8rlz8AyneIR8Q13FaH3luprC17yQ1DQA8A0tRAAACy3QwRa5278jH
pRFUfwWwCc-Cl/PK653qEPnQAx1gkra6s2y7wVj41RBGRGSx8dnYDOSTjC+PM
mLLLaJNBpaOsb92EckhAfDPYtw1NREPTO-KxxJmIkkyuH55M+OL+HK242t
$QJPmf4N1qlSYdsJ18U7HSIFaaafPNKCpxc7SukfOymtyw+YgjbjKCoYb1IY1Inr
GfpQ77K1d8DdAAR3bveJml1l0zoxwFF/1QxRm6uykPzSd7roDykatQSWE8EauE
1sKs1T004zou/cutUSHeeg91n1VCdwkxSh02q+H4KjYXjZjl7hadivigMyF4f
ghhb100CoyEA/1Sy/rHHRb61le1lTcx0Y4eImx8jpcjCKBWMbVAvYrx50Dw8
eAmd3pOrtN1.97ghby5yPCx8dlnqj7nh9muCkI802ijLhEL3b9NhLKQMVLT
mrxX9e0D0M1HfmlnsJXkA4-qmNqy01EBCoeeAhnrJnY241Xsiwc2CgjYEAucVIL
mQd7J2s51j1P1BVkQJUNH7J02uNk3oTwly2eEEr-rw}J91fideafVadYp9fYrw
p1yG/R560ByY9C9Sexw01NRA8-3D1M428AVxC8YAWcqmeyV1b0rOnHaM6g7A
UXNF7T1vz/WpZLzzLmQgApab5xDtGrBn0IW08C9yHaoiHdp/forNttx0RvY
ibrvkgN1j2WF94dmrlz1Ipjm51VsZwYDdcZ89ZfZHyR008/XHinxgrelwYPl4
ccFTCD|MzMSFcuiss0o56RNes8d|rn4wURGAJM7WCM-CZ/gp9SM568arNt7Wu
SM0yQj4GRU1jymcm7yIRqBgQ-ZNEZCCR1w3Bppm7211Kghe9dPRg53UB262M0s
iogZAFNDjWRV1s5a1nsVc11sgca001hEY18Wk2ezVm/ph/vtrLitr/BBD
FKS/jzSiFN6GagLrPrkvgtJt4q)k2EBf7AoKvoXv75VALddlyFQ8KvqrBpvQw
UmOsWfwBqQhtBbn?/3X1bAwspw4fu101GUWUHVF2N1Y42hrt/gRC53zmC7+/

```

i-058749e47b8280025 (proj_public_server)

PublicIPs: 13.213.35.155 PrivateIPs: 10.0.1.46

*from the public instance, connect to the private EC2 instance.

```

ubuntu@ip-10-0-1-46:~$ ssh -i "projKey.pem" ubuntu@10.0.2.179
The authenticity of host '10.0.2.179 (10.0.2.179)' can't be established.
ED25519 key fingerprint is SHA256:2YcpIBieltvVVMaqxwOmTCudHGe7ynv9NDL5x1Gio.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '10.0.2.179' (ED25519) to the list of known hosts.
Welcome to Ubuntu 22.04.2 LTS (GNU/Linux 5.15.0-1031-aws x86_64)

 * Documentation: https://help.ubuntu.com
 * Management: https://landscape.canonical.com
 * Support: https://ubuntu.com/advantage

System information as of Thu Apr  6 05:57:17 UTC 2023

System load: 0.0          Processes:           95
Usage of /: 20.1% of 7.57GB  Users logged in:      0
Memory usage: 20%          IPv4 address for eth0: 10.0.2.179
Swap usage:  0%

Expanded Security Maintenance for Applications is not enabled.

0 updates can be applied immediately.

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

The list of available updates is more than a week old.

i-058749e47b8280025 (proj_public_server)

```

PublicIPs: 13.213.35.155 PrivateIPs: 10.0.1.46

*To install the mysql server use the command

\$sudo apt-get install mysql-server

```

aws Services Search [Alt+S]
ubuntu@ip-10-0-2-179:~$ sudo apt-get install mysql-server
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libcgip-fast-perl libcgip-pm-perl libclone-perl libcode-locale-perl libevent-pthreads-2.1-7 libfcgi-bin libfcgi-perl libfcgi01db libhtml-parser-perl
  libhtml-tagset-perl libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl liblwp-mediatypes-perl libmecab2 libprotobuf-lite23 libtmeate-perl
  liburi-perl mecab-ipadic mecab-ipadic-utf8 mecab-utils mysql-client-8.0 mysql-client-core-8.0 mysql-common mysql-server-8.0 mysql-server-core-8.0
Suggested packages:
  libdata-dump-perl libipic-sharedcache-perl libbusiness-isbn-perl libwww-perl mailx tinyca
The following NEW packages will be installed:
  libcgip-fast-perl libclone-perl libcode-locale-perl libevent-pthreads-2.1-7 libfcgi-bin libfcgi-perl libfcgi01db libhtml-parser-perl
  libhtml-tagset-perl libhtml-template-perl libhttp-date-perl libhttp-message-perl libio-html-perl liblwp-mediatypes-perl libmecab2 libprotobuf-lite23 libtmeate-perl
  liburi-perl mecab-ipadic mecab-ipadic-utf8 mecab-utils mysql-client-8.0 mysql-client-core-8.0 mysql-common mysql-server-8.0 mysql-server-core-8.0
0 upgraded, 28 newly installed, 0 to remove and 0 not upgraded.
Need to get 29.5 MB of archives.
After this operation, 242 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:1 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 mysql-common all 5.8+1.0.8 [7212 B]
Get:2 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 mysql-client-core-8.0 amd64 8.0.32-0ubuntu0.22.04.2 [2677 kB]
Get:3 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 mysql-client-8.0 amd64 8.0.32-0ubuntu0.22.04.2 [22.7 kB]
Get:4 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 mysql-client-core-8.0 amd64 8.0.32-0ubuntu0.22.04.2 [7642 B]
Get:5 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libevent-pthreads-2.1-7 amd64 2.1.12-stable-1build3 [199 kB]
Get:6 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 libprotobuf-lite23 amd64 3.12.4-1ubuntu7.22.04.1 [209 kB]
Get:7 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 mysql-server-core-8.0 amd64 8.0.32-0ubuntu0.22.04.2 [17.5 kB]
Get:8 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy-updates/main amd64 mysql-server-8.0 amd64 8.0.32-0ubuntu0.22.04.2 [1427 kB]
Get:9 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 libhtml-tagset-perl all 3.20-4 [12.5 kB]
Get:10 http://ap-southeast-1.ec2.archive.ubuntu.com/ubuntu jammy/main amd64 liburi-perl all 5.10-1 [78.8 kB]

i-058749e47b8280025 (proj_public_server)
PublicIPs: 13.213.35.155 PrivateIPs: 10.0.1.46

```

*check the version of mysql using the command

\$mysql --version

```

aws Services Search [Alt+S]
No services need to be restarted.
No containers need to be restarted.
No user sessions are running outdated binaries.

No VM guests are running outdated hypervisor (qemu) binaries on this host.
ubuntu@ip-10-0-2-179:~$ sudo su
root@ip-10-0-2-179:/home/ubuntu# mysql --version
mysql Ver 8.0.32-0ubuntu0.22.04.2 for Linux on x86_64 ((Ubuntu))
root@ip-10-0-2-179:/home/ubuntu# systemctl start mysql
root@ip-10-0-2-179:/home/ubuntu# systemctl status mysql
● mysql.service - MySQL Community Server
   Loaded: loaded (/lib/systemd/system/mysql.service; enabled; vendor preset: enabled)
   Active: active (running) since Thu 2023-04-06 05:58:27 UTC; 1min 30s ago
     Process: 2071 ExecStartPre=/usr/share/mysql/mysql-systemd-start pre (code=exited, status=0/SUCCESS)
    Main PID: 2079 (mysqld)
      Status: "Server is operational"
     Tasks: 38 (limit: 1141)
    Memory: 361.4M
       CPU: 1.282s
      CGroup: /system.slice/mysql.service
              └─2079 /usr/sbin/mysqld

Apr 06 05:58:25 ip-10-0-2-179 systemd[1]: Starting MySQL Community Server...
Apr 06 05:58:27 ip-10-0-2-179 systemd[1]: Started MySQL Community Server.
root@ip-10-0-2-179:/home/ubuntu# 

i-058749e47b8280025 (proj_public_server)
PublicIPs: 13.213.35.155 PrivateIPs: 10.0.1.46

```

```

root@ip-10-0-2-179:/home/ubuntu# exit
exit

```

\$mysql -h database-end-point -P 3306 -u admin -p

This command is used to connect to a MySQL database server using the command line interface.

Here is the breakdown of the different parts of the command:

mysql: This is the command to start the MySQL client program.

-h: This option specifies the hostname or IP address of the MySQL server.

database-endpoint: This is the hostname or IP address of the MySQL server.

-P: This option specifies the port number to use for the connection. The default port for MySQL is 3306, which is why it is specified here.

3306: This is the port number for the MySQL server.

-u: This option specifies the username to use for the connection.

admin: This is the username for the MySQL server.

-p: This option prompts for a password to be entered. When you run the command, you will be prompted to enter the password for the specified user.

After running this command, if the username and password are correct, you will be logged in to the MySQL server and can start executing SQL commands.

mysql>create database mydb;

This command is used to create a new MySQL database with the name "mydb". Here is a breakdown of the command:

create database: This specifies that we want to create a new database.

mydb: This is the name of the database that we want to create.

After running this command, the new database "mydb" will be created in the MySQL server. Note that the user running the command must have sufficient privileges to create databases in the server.

mysql>use mydb;

This command is used to select a specific database in MySQL server and start using it for executing SQL commands. Here is a breakdown of the command:

use: This specifies that we want to select a specific database to use.

mydb: This is the name of the database that we want to select.

After running this command, any SQL commands that we execute will be executed in the context of the "mydb" database. If the database doesn't exist, an error will be returned.

mysql>CREATE TABLE users (
username VARCHAR(50) NOT NULL,
password VARCHAR(255) NOT NULL
);

This command is used to create a new table named "users" in the current database, with two columns: "username" and "password". Here is a breakdown of the command:

CREATE TABLE: This specifies that we want to create a new table.

users: This is the name of the table that we want to create.

username: This is the name of the first column in the table, which will hold usernames. It has a data type of VARCHAR(50), which means it can hold up to 50 characters of variable length.

NOT NULL: This specifies that the "username" column cannot be empty or NULL.

password: This is the name of the second column in the table, which will hold passwords. It has a data type of VARCHAR(255), which means it can hold up to 255 characters of variable length.

NOT NULL: This specifies that the "password" column cannot be empty or NULL.

After running this command, the new table "users" will be created in the current database, with the two specified columns. Note that there are other data types and column constraints that can be used when creating tables in MySQL, depending on the specific needs of your application.

mysql>INSERT INTO users (username, password) VALUES ('john', 'password123');

This command is used to insert a new row of data into the "users" table with the specified values for the "username" and "password" columns. Here is a breakdown of the command:

INSERT INTO: This specifies that we want to insert data into a table.

users: This is the name of the table that we want to insert data into.

(username, password): These are the names of the columns that we want to insert data into.

VALUES: This specifies that we want to specify the values to insert.

('john', 'password123'): These are the actual values that we want to insert into the "username" and "password" columns, respectively.

After running this command, a new row will be added to the "users" table with the specified values for the "username" and "password" columns. Note that if a column has a NOT NULL constraint and no value is specified, the insert statement will fail with an error.

The screenshot shows a terminal window with the AWS Services menu open at the top. The main area displays the MySQL command-line interface. The user creates a database named 'mydb' and then creates a table 'users' with two columns: 'username' (VARCHAR(50)) and 'password' (VARCHAR(255)). Finally, the user inserts a single record ('john', 'password123') into the 'users' table. The terminal also shows the MySQL prompt and various status messages.

```
aws | Services | Search [Alt+S]
ubuntu@ip-10-0-2-179:~$ mysql -h database-1.cxy96jpe13k.ap-southeast-1.rds.amazonaws.com -P 3306 -u admin -p
Enter password:
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 22
Server version: 8.0.32 Source distribution

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

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

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

mysql> create database mydb;
Query OK, 1 row affected (0.01 sec)

mysql> use mydb;
Database changed
mysql> CREATE TABLE users (
    >>     username VARCHAR(50) NOT NULL,
    >>     password VARCHAR(255) NOT NULL
    >> );
Query OK, 0 rows affected (0.02 sec)

mysql> INSERT INTO users (username, password) VALUES ('john', 'password123');
Query OK, 1 row affected (0.00 sec)

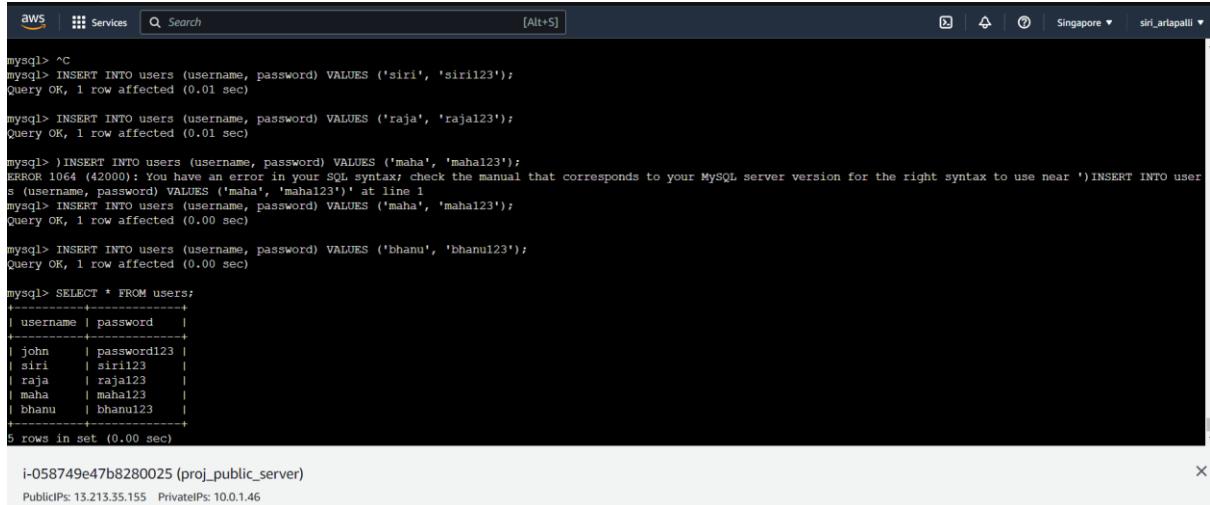
mysql> SELECT * FROM users;
+-----+-----+
| username | password |
+-----+-----+
| john     | password123 |
+-----+-----+
1 row in set (0.00 sec)

mysql>
```

i-058749e47b8280025 (proj_public_server)
PublicIPs: 13.213.35.155 PrivateIPs: 10.0.1.46

mysql>SELECT * from users;

"SELECT * FROM users" would retrieve all the data from the "users" table in a relational database. The asterisk (*) is a wildcard character that represents all columns in the table. This query would return all the rows and columns in the "users" table, showing information about each user stored in the table.



A screenshot of the AWS MySQL Workbench interface. The terminal window shows several SQL commands being run against a 'users' table. The first three insertions succeed, but the fourth one fails due to a syntax error ('ERROR 1064 (42000)'). The fifth insertion succeeds. Finally, a 'SELECT * FROM users' query is run, displaying the data in a tabular format:

username	password
john	password123
siri	siri123
raja	raja123
maha	maha123
bhanu	bhanu123

The message '5 rows in set (0.00 sec)' is displayed at the bottom. The status bar at the bottom right shows the instance ID 'i-058749e47b8280025 (proj_public_server)' and IP addresses 'PublicIPs: 13.213.35.155 PrivateIPs: 10.0.1.46'.

Reference Links:

<https://awstip.com/lets-create-3-tier-architecture-with-aws-6c836bddc023>

<https://medium.com/the-andela-way/designing-a-three-tier-architecture-in-aws-e5c24671f124>

<https://thornelabs.net/posts/deploy-a-three-tier-web-application-to-kubernetes/>

<https://www.c-sharpcorner.com/article/how-to-create-aws/>

<https://chat.openai.com/chat>

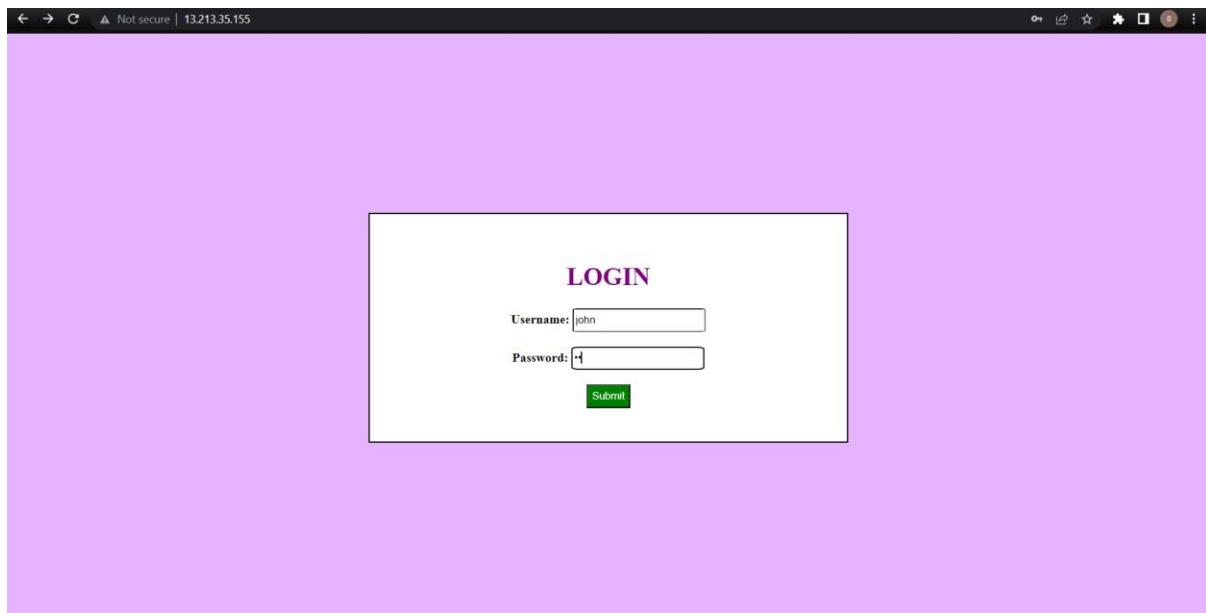
OUTCOMES:

A screenshot of a web browser window. The address bar shows the URL "13.213.35.155" and a "Not secure" warning. The main content area is a white rectangular box containing a "LOGIN" form. The form has the following fields:

- Username:
- Password:
- Submit button

A screenshot of a web browser window, identical to the one above, but with the "Username" field populated with the value "john". The "Password" field contains several dots, indicating masked input. The "Submit" button is present.

A screenshot of a web browser window. The address bar shows the URL "13.213.35.155/authenticate.php". The main content area displays the text "Welcome john!"



CONCLUSION

- We have successfully deployed a 3 tier web application in AWS cloud.
- Deploying a three-tier web application in the cloud offers numerous benefits and is a recommended approach for organizations looking to provide a scalable, available, and cost-effective solution to their users.