Deploy two tier application with aws

Introduction: This project demonstrates the deployment of a two-tier web application on AWS using EC2, RDS, and networking components. The architecture consists of a public-facing web server and a private database layer, ensuring security and scalability.

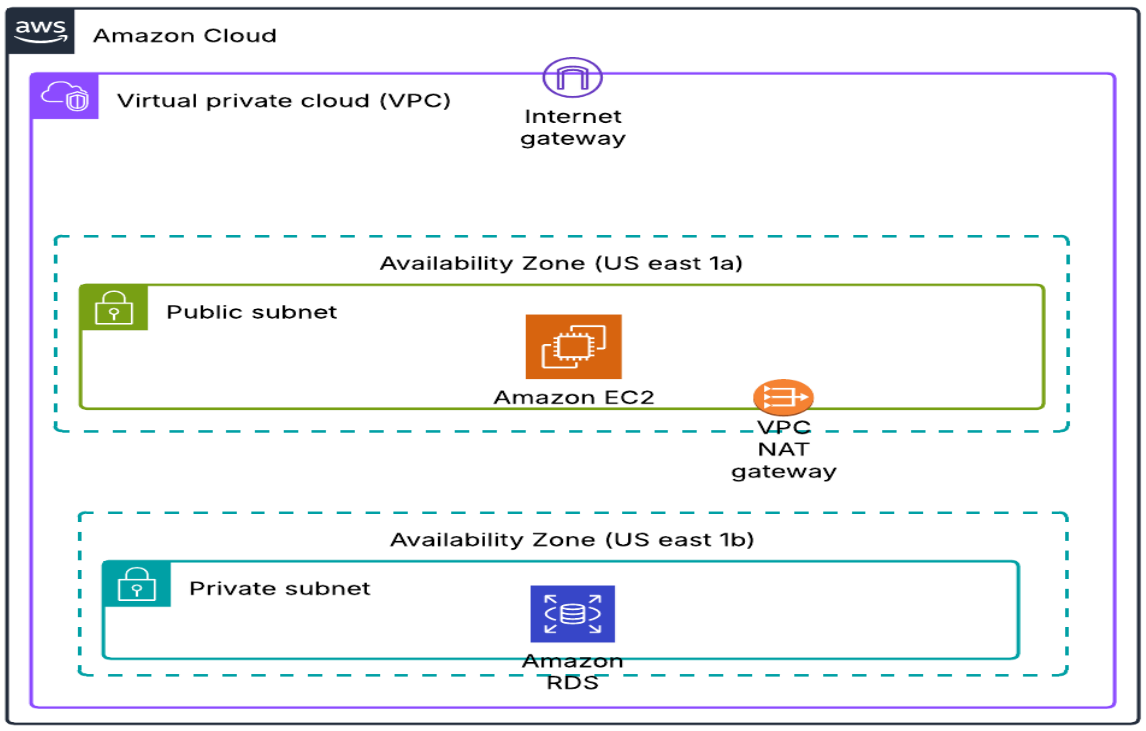
Objectives

- Set up a Virtual Private Cloud (VPC) with public and private subnets.
- Deploy a web server (EC2) in the public subnet.
- Configure a managed database (RDS) in the private subnet.
- Secure communication between EC2 and RDS.
- Deploy WordPress on the web server with a MariaDB database.

. AWS Services Used

- **Networking**: VPC, Subnets, Route Tables, Internet Gateway, NAT Gateway
- **Compute**: EC2 (Amazon Linux 2)
- **Database**: RDS (MariaDB)
- **Security**: Security Groups, SSH Access

Architecture Diagram



Implementation Steps:

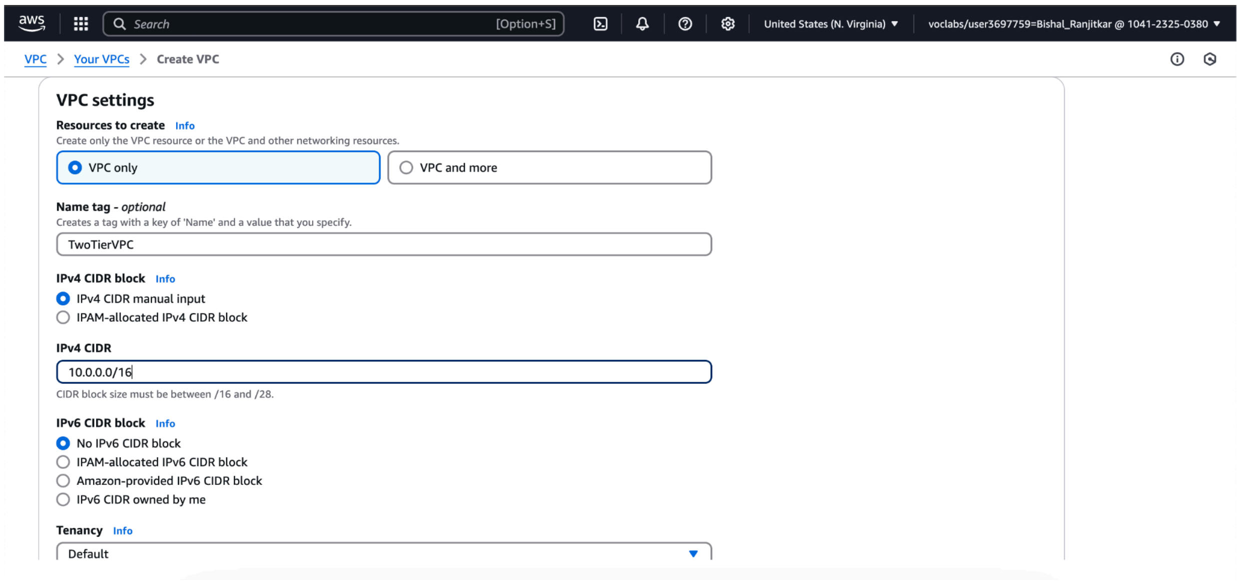
Step 1: VPC and Networking Setup\

Added two subnets:

- **Public Subnet** (10.0.1.0/24) for EC2.
- **Private Subnet** (10.0.2.0/24) for RDS.

Configured route tables:

- Public Route Table: Connected to an Internet Gateway.
- Private Route Table: Connected to a NAT Gateway for outbound internet access.


aws
Search [Option+S]
United States (N. Virginia) ▼
voclabs/user3697759=Bishal_Ranjitkar @ 1041-2325-0380 ▼
VPC > Your VPCs > Create VPC

VPC settings

Resources to create Info
Create only the VPC resource or the VPC and other networking resources.
[✓ VPC only] VPC and more

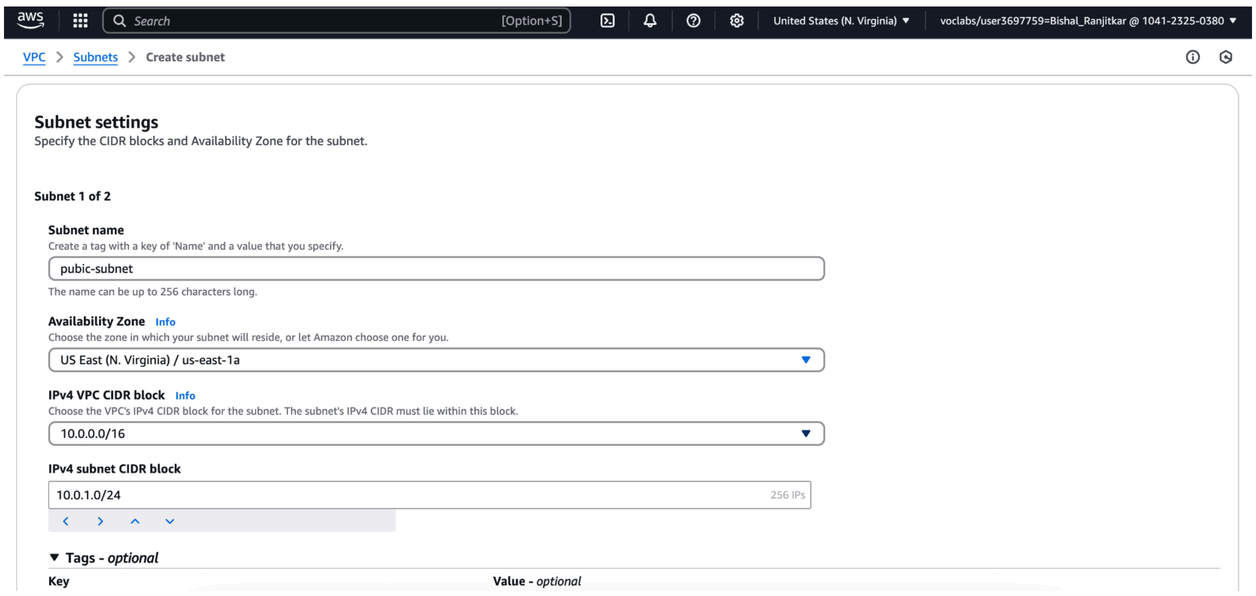
Name tag - *optional*
Creates a tag with a key of 'Name' and a value that you specify.
TwoTierVPC

IPv4 CIDR block Info
[✓ IPv4 CIDR manual input]
IPAM-allocated IPv4 CIDR block

IPv4 CIDR
10.0.0.0/16
CIDR block size must be between /16 and /28.

IPv6 CIDR block Info
[✓ No IPv6 CIDR block]
IPAM-allocated IPv6 CIDR block
Amazon-provided IPv6 CIDR block
IPv6 CIDR owned by me

Tenancy Info
Default


aws
Search [Option+S]
United States (N. Virginia) ▼
voclabs/user3697759=Bishal_Ranjitkar @ 1041-2325-0380 ▼
VPC > Subnets > Create subnet

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

Subnet 1 of 2

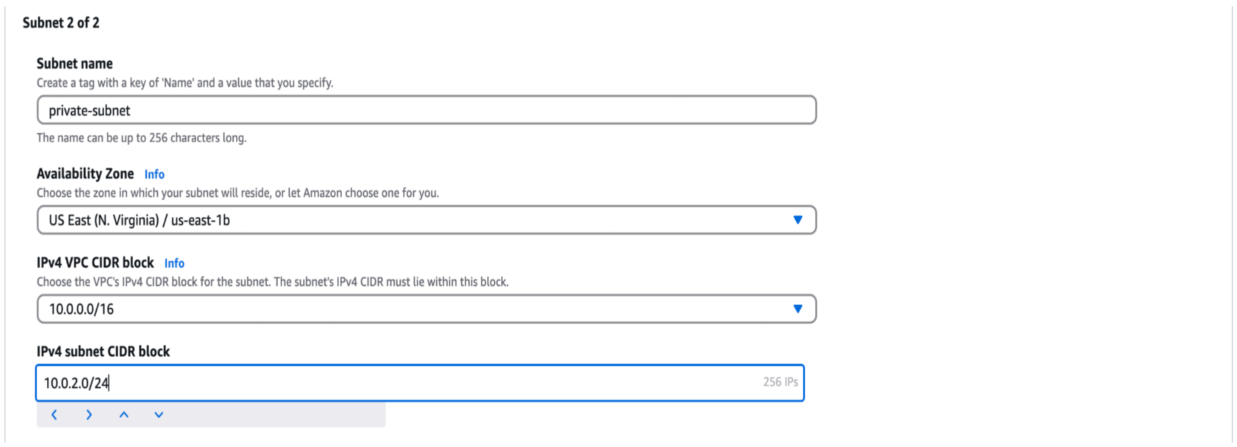
Subnet name
Create a tag with a key of 'Name' and a value that you specify.
pubic-subnet
The name can be up to 256 characters long.

Availability Zone Info
Choose the zone in which your subnet will reside, or let Amazon choose one for you.
US East (N. Virginia) / us-east-1a

IPv4 VPC CIDR block Info
Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.
10.0.0.0/16

IPv4 subnet CIDR block
10.0.1.0/24 256 IPs

▼ Tags - *optional*
Key Value - *optional*


Subnet 2 of 2

Subnet name

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

private-subnet

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.

US East (N. Virginia) / us-east-1b

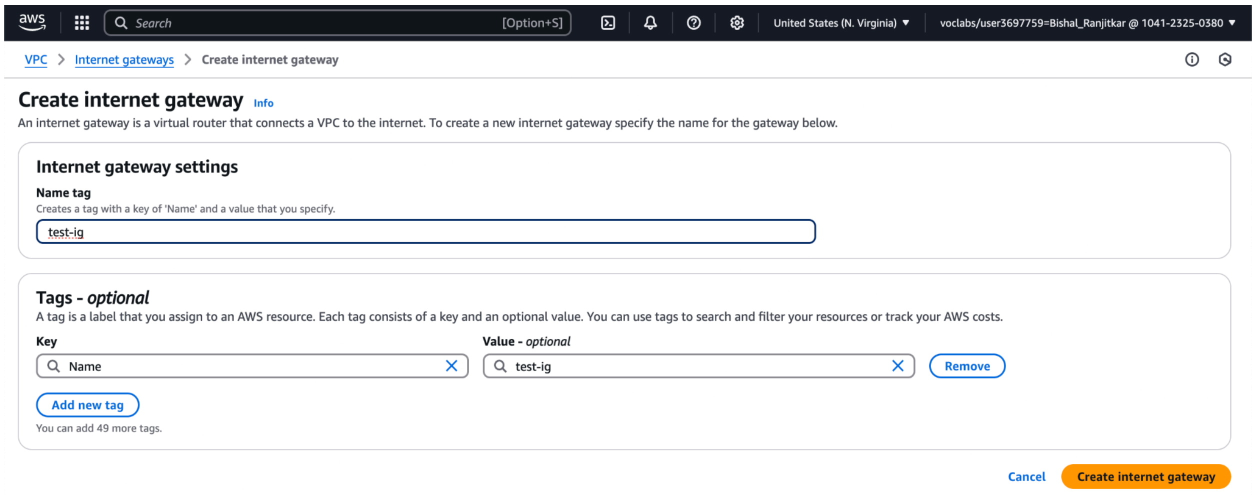
IPv4 VPC CIDR block Info

Choose the VPC's IPv4 CIDR block for the subnet. The subnet's IPv4 CIDR must lie within this block.

10.0.0.0/16

IPv4 subnet CIDR block

10.0.2.0/24 256 IPs


aws
Search [Option+S]
United States (N. Virginia)
voclabs/user3697759=Bishal_Ranjitkar @ 1041-2325-0380

VPC > Internet gateways > Create internet gateway

Create internet gateway Info

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

Internet gateway settings

Name tag

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

test-ig

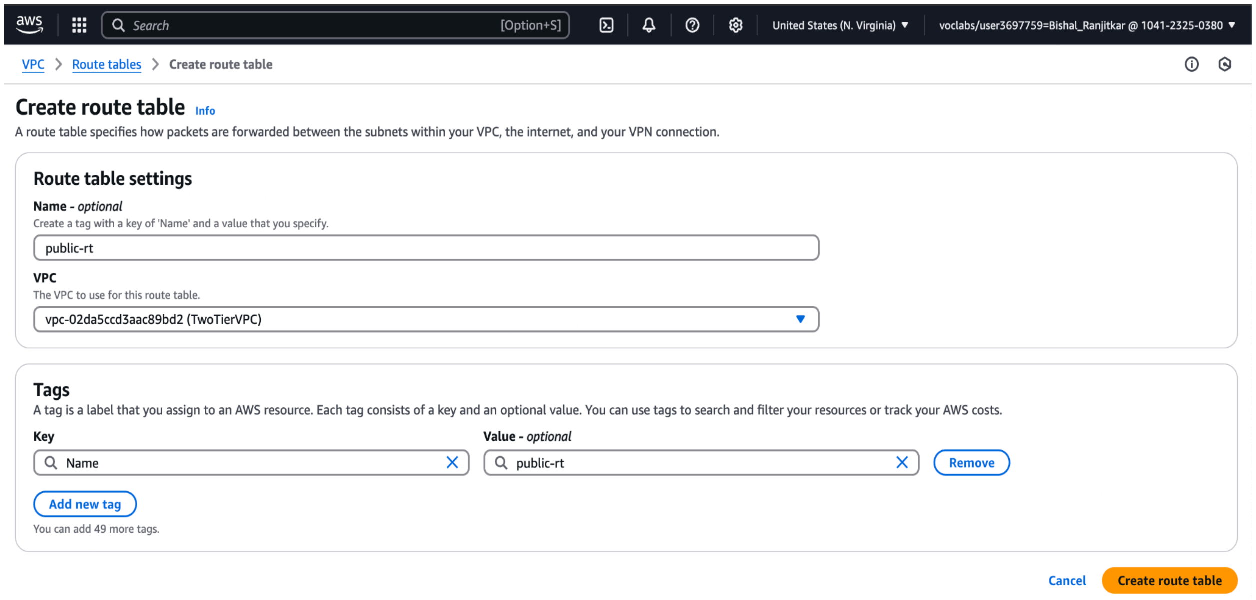
Tags - *optional*

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

Key	Value - *optional*
Name	test-ig

You can add 49 more tags.

Cancel
Create internet gateway


aws
Search [Option+S]
United States (N. Virginia)
voclabs/user3697759=Bishal_Ranjitkar @ 1041-2325-0380

VPC > Route tables > Create 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.

public-rt

VPC

The VPC to use for this route table.

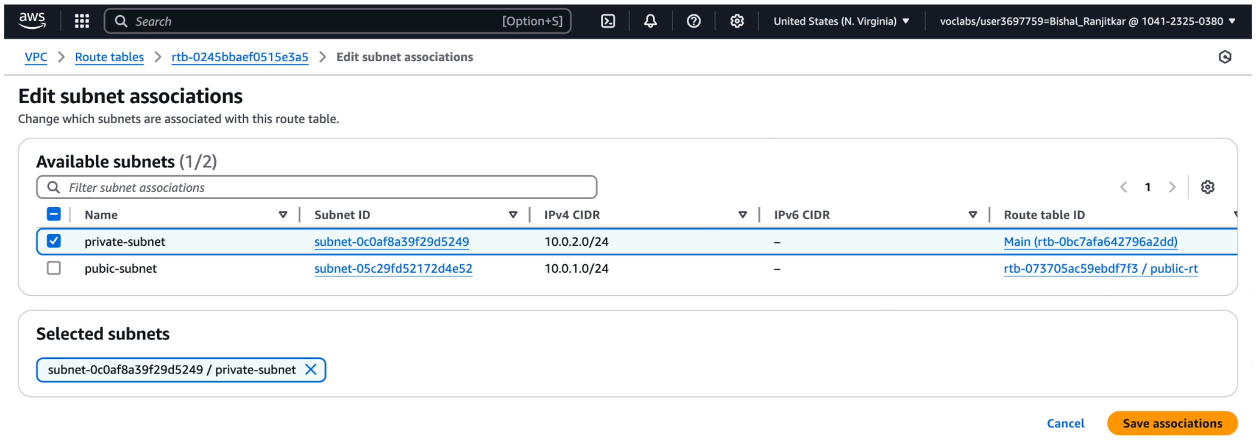
vpc-02da5ccd3aac89bd2 (TwoTierVPC)

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 - *optional*
Name	public-rt

You can add 49 more tags.


aws
Search [Option+S]
United States (N. Virginia)
voclabs/user3697759=Bishal_Ranjitkar @ 1041-2325-0380
VPC > Route tables > rtb-0245bbaef0515e3a5 > Edit subnet associations

Edit subnet associations

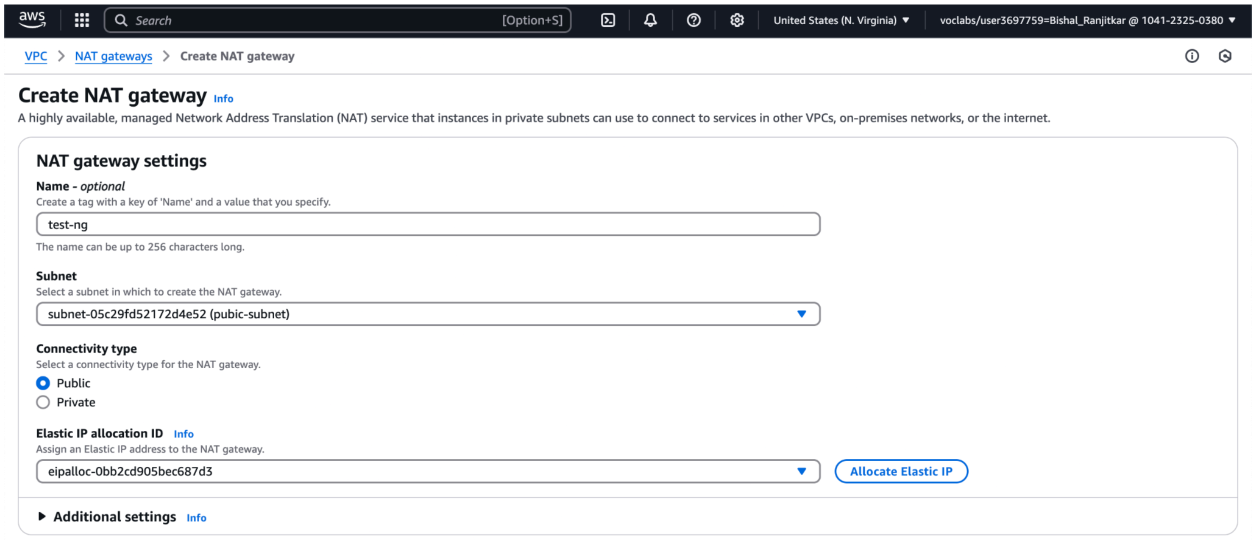
Change which subnets are associated with this route table.

Available subnets (1/2)

	Name	Subnet ID	IPv4 CIDR	IPv6 CIDR	Route table ID
✓	private-subnet	subnet-0c0af8a39f29d5249	10.0.2.0/24	–	Main (rtb-0bc7afa642796a2dd)
	pubic-subnet	subnet-05c29fd52172d4e52	10.0.1.0/24	–	rtb-073705ac59ebdf7f3 / public-rt

Selected subnets

subnet-0c0af8a39f29d5249 / private-subnet


aws
Search [Option+S]
United States (N. Virginia)
voclabs/user3697759=Bishal_Ranjitkar @ 1041-2325-0380
VPC > NAT gateways > Create NAT gateway

Create NAT gateway Info

A highly available, managed Network Address Translation (NAT) service that instances in private subnets can use to connect to services in other VPCs, on-premises networks, or the internet.

NAT gateway settings

Name - *optional*
Create a tag with a key of 'Name' and a value that you specify.
test-ng
The name can be up to 256 characters long.

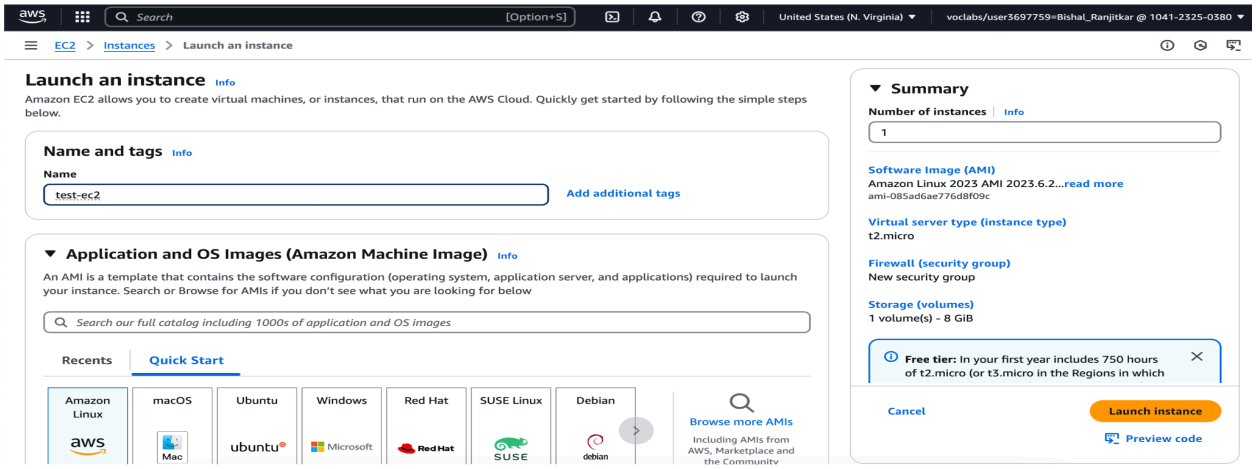
Subnet
Select a subnet in which to create the NAT gateway.
subnet-05c29fd52172d4e52 (pubic-subnet)

Connectivity type
Select a connectivity type for the NAT gateway.
[✓ Public] Private

Elastic IP allocation ID Info
Assign an Elastic IP address to the NAT gateway.
eipalloc-0bb2cd905bec687d3

Step 2: Launching EC2 and Installing LAMP

- Deployed an EC2 instance in the public subnet.
- Connected via SSH and installed LAMP stack:


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: test-ec2

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

Summary

Number of instances: 1

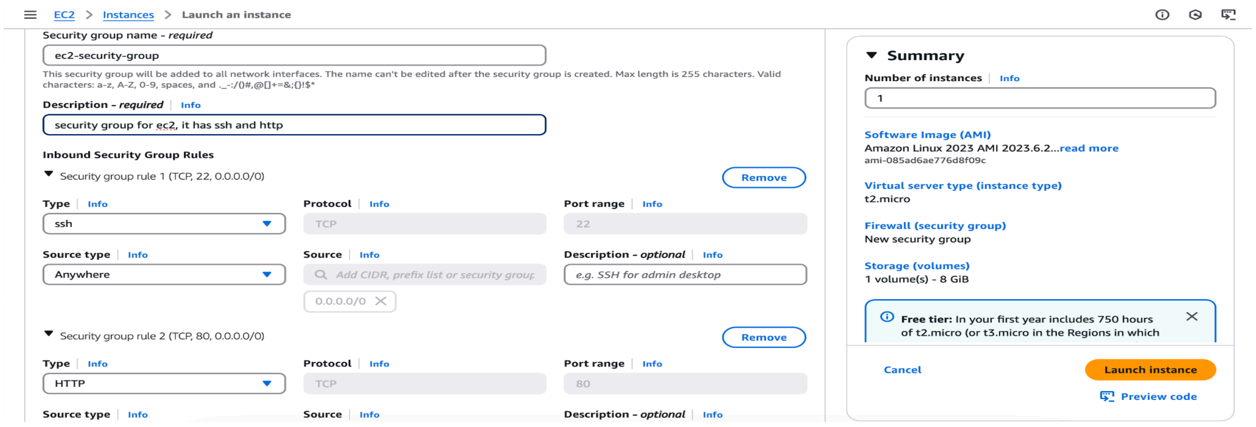
Software Image (AMI): Amazon Linux 2023 AMI 2023.6.2...read more
ami-085ad6ae776d8f09c

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


EC2 > Instances > Launch an instance

Security group name - *required*: ec2-security-group

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*: security group for ec2, it has ssh and http

Inbound Security Group Rules

Security group rule 1 (TCP, 22, 0.0.0.0/0)

Type	Protocol	Port range
ssh	TCP	22

Source type	Source	Description - *optional*
Anywhere	0.0.0.0/0	e.g. SSH for admin desktop

Security group rule 2 (TCP, 80, 0.0.0.0/0)

Type	Protocol	Port range
HTTP	TCP	80

Summary

Number of instances: 1

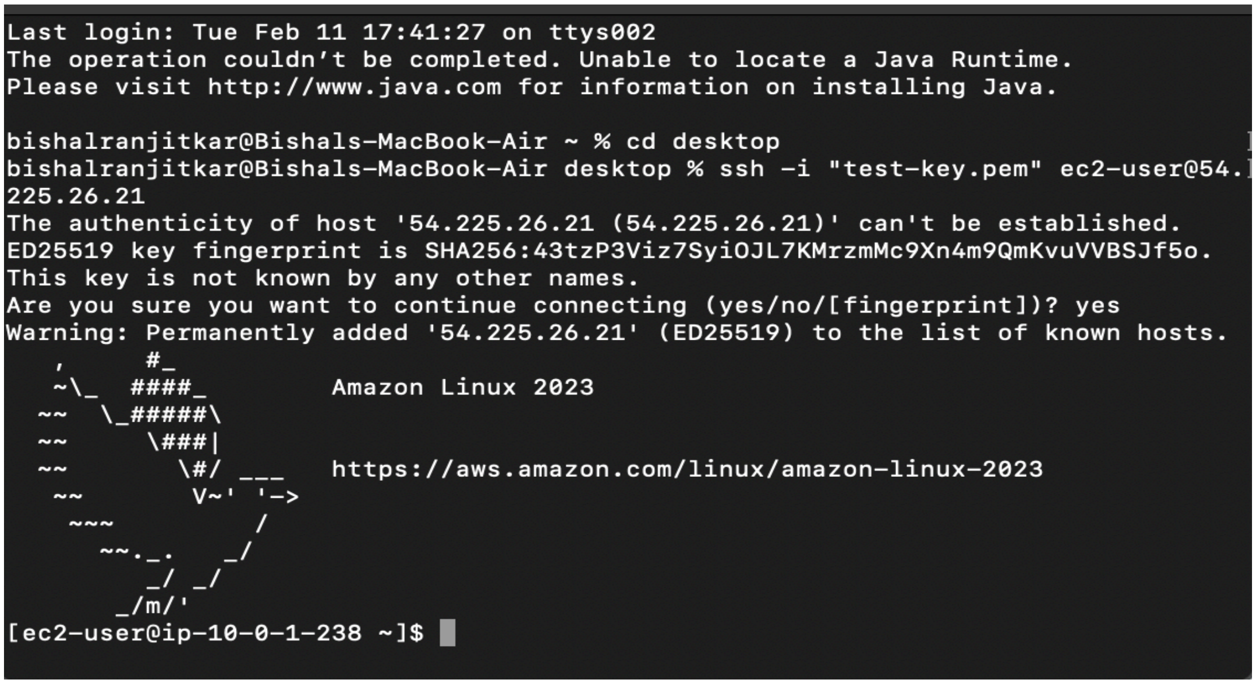
Software Image (AMI): Amazon Linux 2023 AMI 2023.6.2...read more
ami-085ad6ae776d8f09c

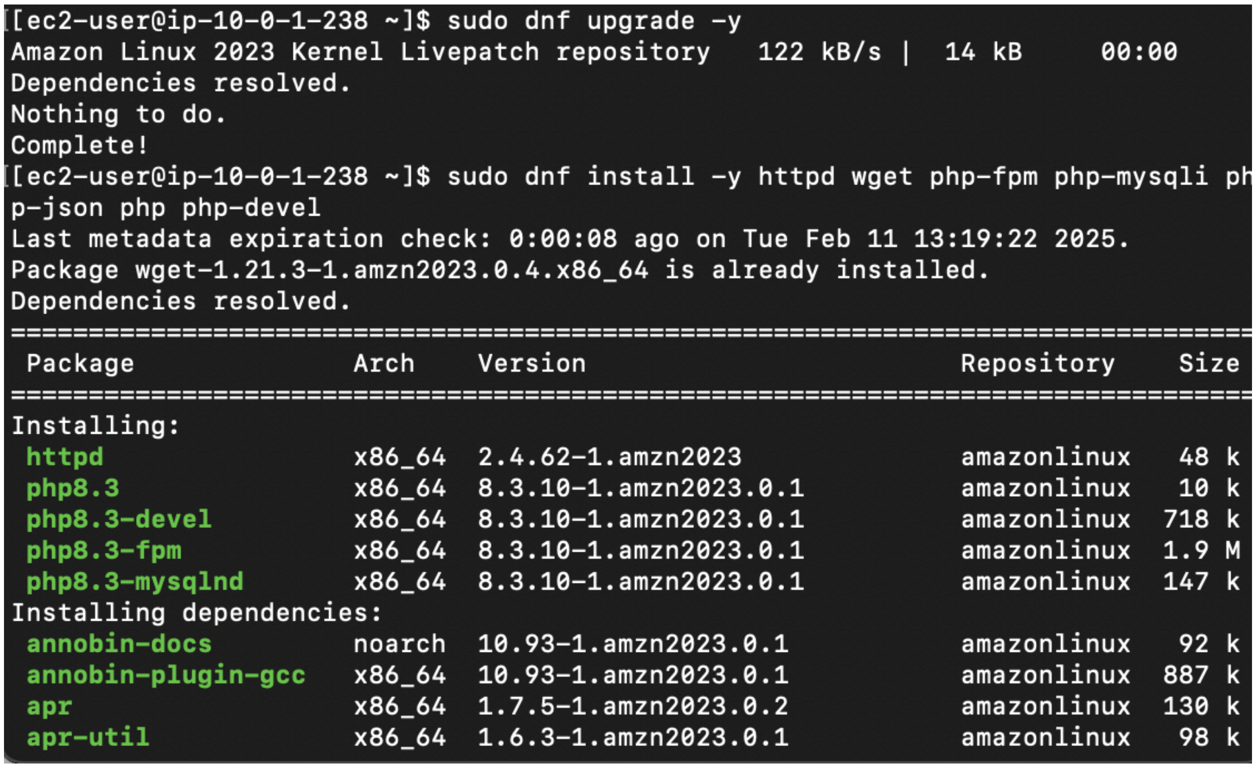
Virtual server type (instance type): t2.micro

Firewall (security group): New security group

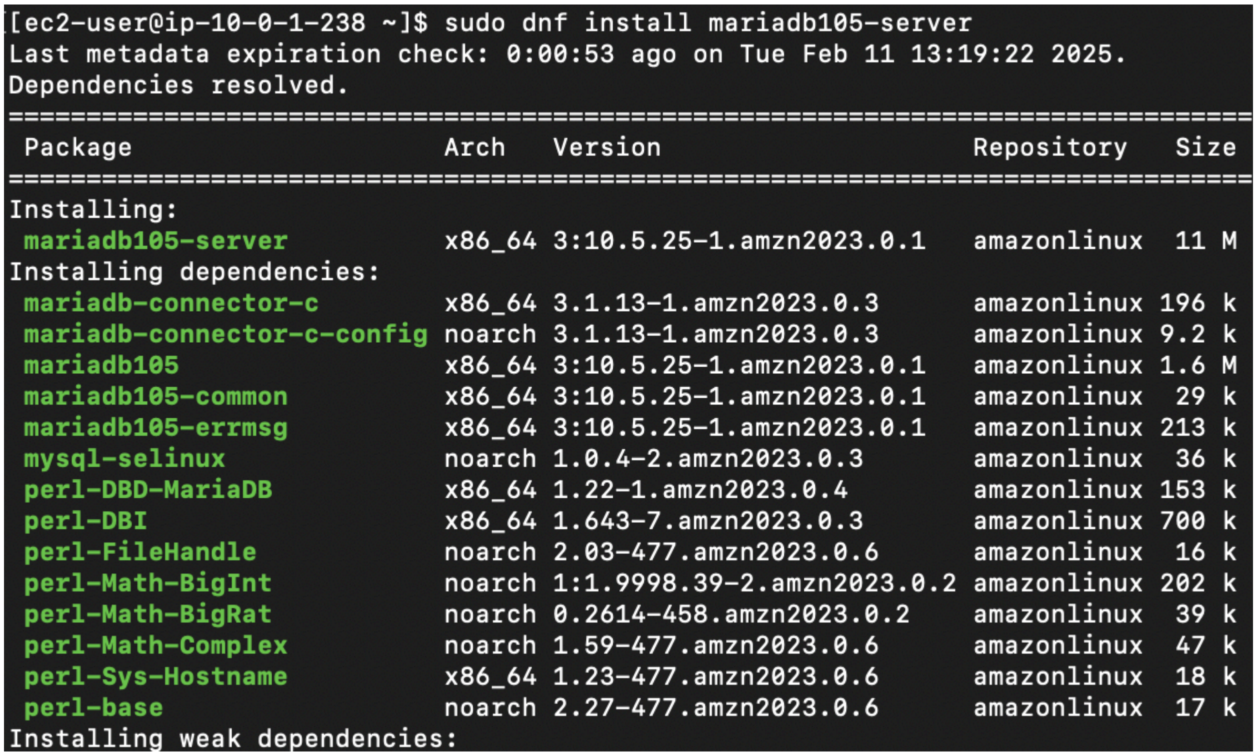
Storage (volumes): 1 volume(s) - 8 GiB

Free tier: In your first year includes 750 hours of t2.micro (or t3.micro in the Regions in which


```
Last login: Tue Feb 11 17:41:27 on ttys002
The operation couldn’t be completed. Unable to locate a Java Runtime.
Please visit http://www.java.com for information on installing Java.

bishalranjitkar@Bishals-MacBook-Air ~ % cd desktop
bishalranjitkar@Bishals-MacBook-Air desktop % ssh -i "test-key.pem" ec2-user@54.
225.26.21
The authenticity of host '54.225.26.21 (54.225.26.21)' can't be established.
ED25519 key fingerprint is SHA256:43tzP3Viz7SyiOJL7KMrzmMc9Xn4m9QmKvuVVBSJf5o.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '54.225.26.21' (ED25519) to the list of known hosts.
   ,     #_
   ~\_  ####_        Amazon Linux 2023
  ~~  \_#####\
  ~~     \###|
  ~~       \#/ ___   https://aws.amazon.com/linux/amazon-linux-2023
   ~~       V~' '->
    ~~~         /
      ~~._.   _/
         _/ _/
       _/m/'
[ec2-user@ip-10-0-1-238 ~]$ 
```



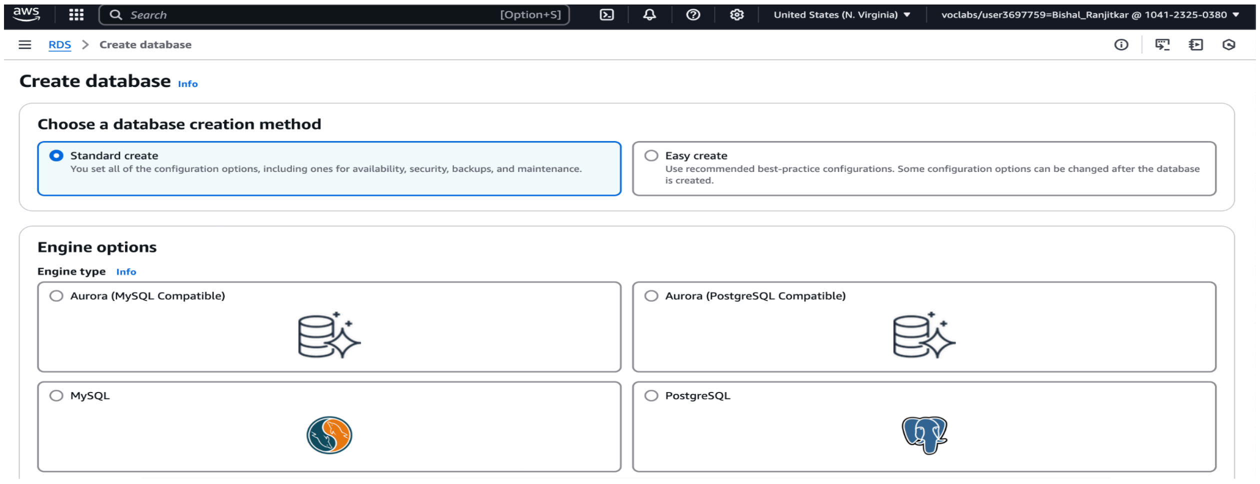
```
[ec2-user@ip-10-0-1-238 ~]$ sudo dnf upgrade -y
Amazon Linux 2023 Kernel Livepatch repository   122 kB/s |  14 kB     00:00
Dependencies resolved.
Nothing to do.
Complete!
[ec2-user@ip-10-0-1-238 ~]$ sudo dnf install -y httpd wget php-fpm php-mysqli ph
p-json php php-devel
Last metadata expiration check: 0:00:08 ago on Tue Feb 11 13:19:22 2025.
Package wget-1.21.3-1.amzn2023.0.4.x86_64 is already installed.
Dependencies resolved.
================================================================================
 Package                Arch     Version                      Repository    Size
================================================================================
Installing:
 httpd                  x86_64   2.4.62-1.amzn2023            amazonlinux   48 k
 php8.3                 x86_64   8.3.10-1.amzn2023.0.1        amazonlinux   10 k
 php8.3-devel           x86_64   8.3.10-1.amzn2023.0.1        amazonlinux  718 k
 php8.3-fpm             x86_64   8.3.10-1.amzn2023.0.1        amazonlinux  1.9 M
 php8.3-mysqlnd         x86_64   8.3.10-1.amzn2023.0.1        amazonlinux  147 k
Installing dependencies:
 annobin-docs           noarch   10.93-1.amzn2023.0.1         amazonlinux   92 k
 annobin-plugin-gcc     x86_64   10.93-1.amzn2023.0.1         amazonlinux  887 k
 apr                    x86_64   1.7.5-1.amzn2023.0.2         amazonlinux  130 k
 apr-util               x86_64   1.6.3-1.amzn2023.0.1         amazonlinux   98 k
```



```
[ec2-user@ip-10-0-1-238 ~]$ sudo dnf install mariadb105-server
Last metadata expiration check: 0:00:53 ago on Tue Feb 11 13:19:22 2025.
Dependencies resolved.
================================================================================
 Package                    Arch   Version                    Repository   Size
================================================================================
Installing:
 mariadb105-server          x86_64 3:10.5.25-1.amzn2023.0.1   amazonlinux  11 M
Installing dependencies:
 mariadb-connector-c        x86_64 3.1.13-1.amzn2023.0.3      amazonlinux 196 k
 mariadb-connector-c-config noarch 3.1.13-1.amzn2023.0.3      amazonlinux 9.2 k
 mariadb105                 x86_64 3:10.5.25-1.amzn2023.0.1   amazonlinux 1.6 M
 mariadb105-common          x86_64 3:10.5.25-1.amzn2023.0.1   amazonlinux  29 k
 mariadb105-errmsg          x86_64 3:10.5.25-1.amzn2023.0.1   amazonlinux 213 k
 mysql-selinux              noarch 1.0.4-2.amzn2023.0.3       amazonlinux  36 k
 perl-DBD-MariaDB           x86_64 1.22-1.amzn2023.0.4        amazonlinux 153 k
 perl-DBI                   x86_64 1.643-7.amzn2023.0.3       amazonlinux 700 k
 perl-FileHandle            noarch 2.03-477.amzn2023.0.6      amazonlinux  16 k
 perl-Math-BigInt           noarch 1:1.9998.39-2.amzn2023.0.2 amazonlinux 202 k
 perl-Math-BigRat           noarch 0.2614-458.amzn2023.0.2    amazonlinux  39 k
 perl-Math-Complex          noarch 1.59-477.amzn2023.0.6      amazonlinux  47 k
 perl-Sys-Hostname          x86_64 1.23-477.amzn2023.0.6      amazonlinux  18 k
 perl-base                  noarch 2.27-477.amzn2023.0.6      amazonlinux  17 k
Installing weak dependencies:
```

Step 3: Setting Up RDS (MariaDB)

- Created an RDS instance (MariaDB) in the private subnet.
- Configured security group rules to allow EC2 to access RDS on port 3306.



United States (N. Virginia) | voclabs/user3697759=Bishal_Ranjitkar @ 1041-2325-0380

RDS > Create database

Create database Info

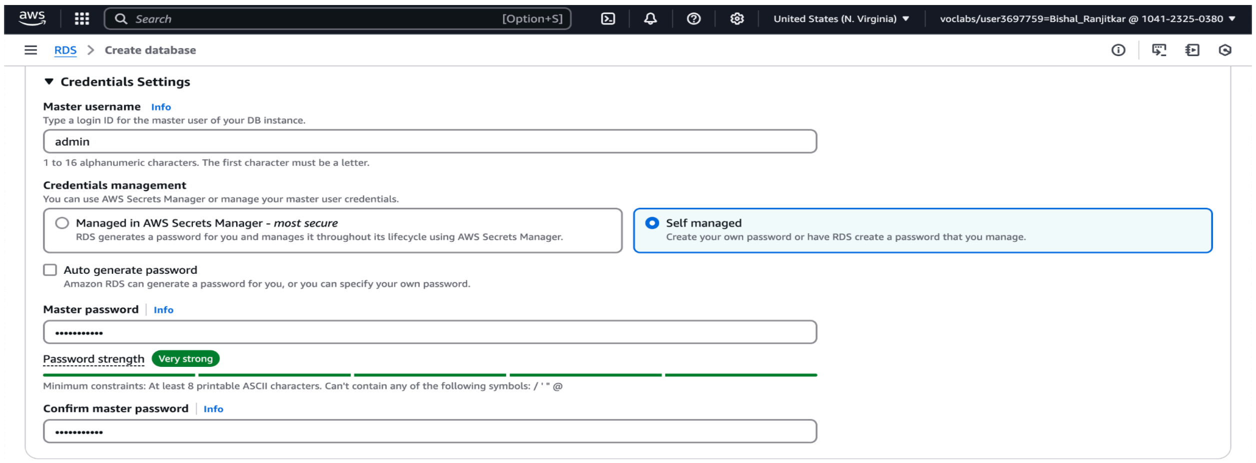
Choose a database creation method

- [✓ 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
- PostgreSQL



RDS > Create database

Credentials Settings

Master username Info
Type a login ID for the master user of your DB instance.
admin
1 to 16 alphanumeric characters. The first character must be a letter.

Credentials management
You can use AWS Secrets Manager or manage your master user credentials.

- Managed in AWS Secrets Manager - *most secure*: RDS generates a password for you and manages it throughout its lifecycle using AWS Secrets Manager.
- [✓ Self managed] Create your own password or have RDS create a password that you manage.

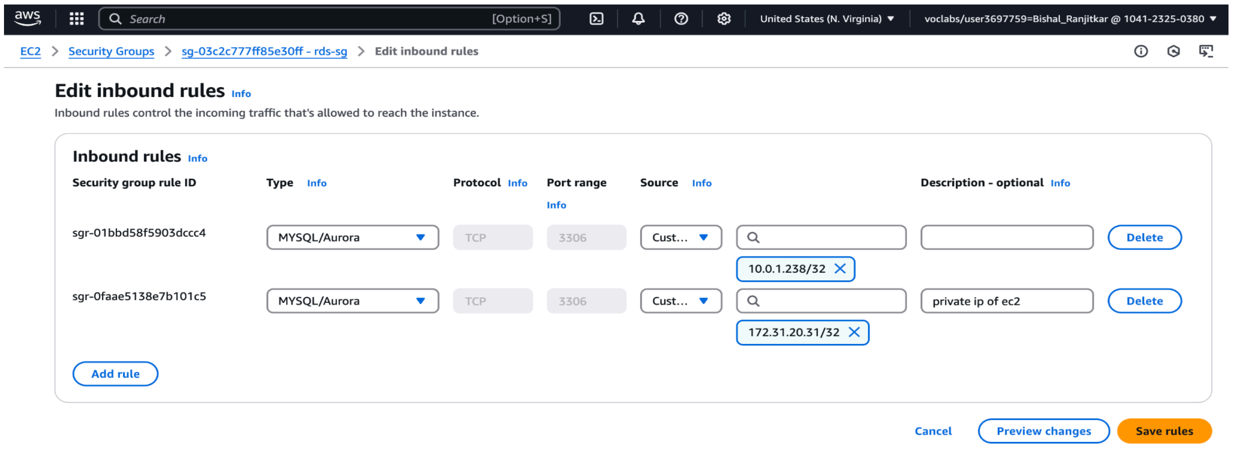
☐ Auto generate password
Amazon RDS can generate a password for you, or you can specify your own password.

Master password Info
•••••••••••

Password strength: Very strong

Minimum constraints: At least 8 printable ASCII characters. Can't contain any of the following symbols: / ' " @

Confirm master password Info
•••••••••••



EC2 > Security Groups > sg-03c2c777ff85e30ff - rds-sg > Edit inbound rules

Edit inbound rules Info

Inbound rules control the incoming traffic that's allowed to reach the instance.

Inbound rules Info

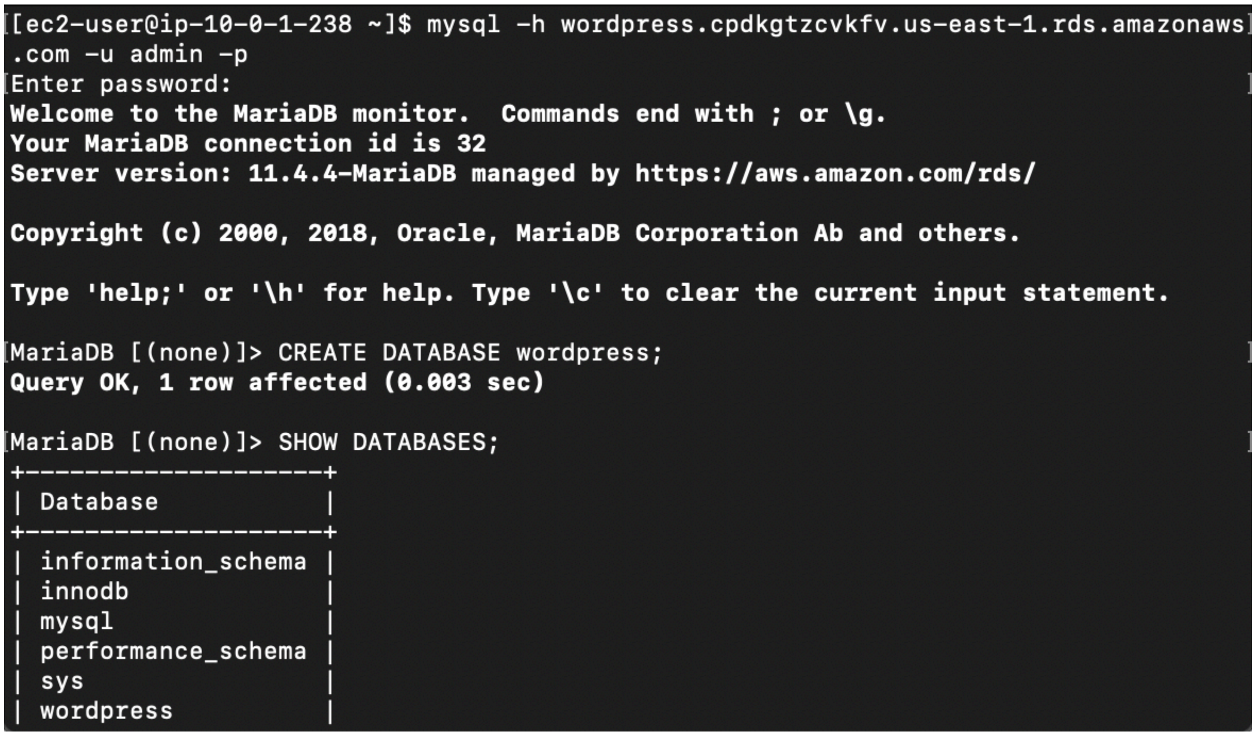
Security group rule ID	Type	Protocol	Port range	Source		Description - optional
sgr-01bbd58f5903dccc4	MYSQL/Aurora	TCP	3306	Cust...	10.0.1.238/32	
sgr-0faae5138e7b101c5	MYSQL/Aurora	TCP	3306	Cust...	172.31.20.31/32	private ip of ec2

Add rule

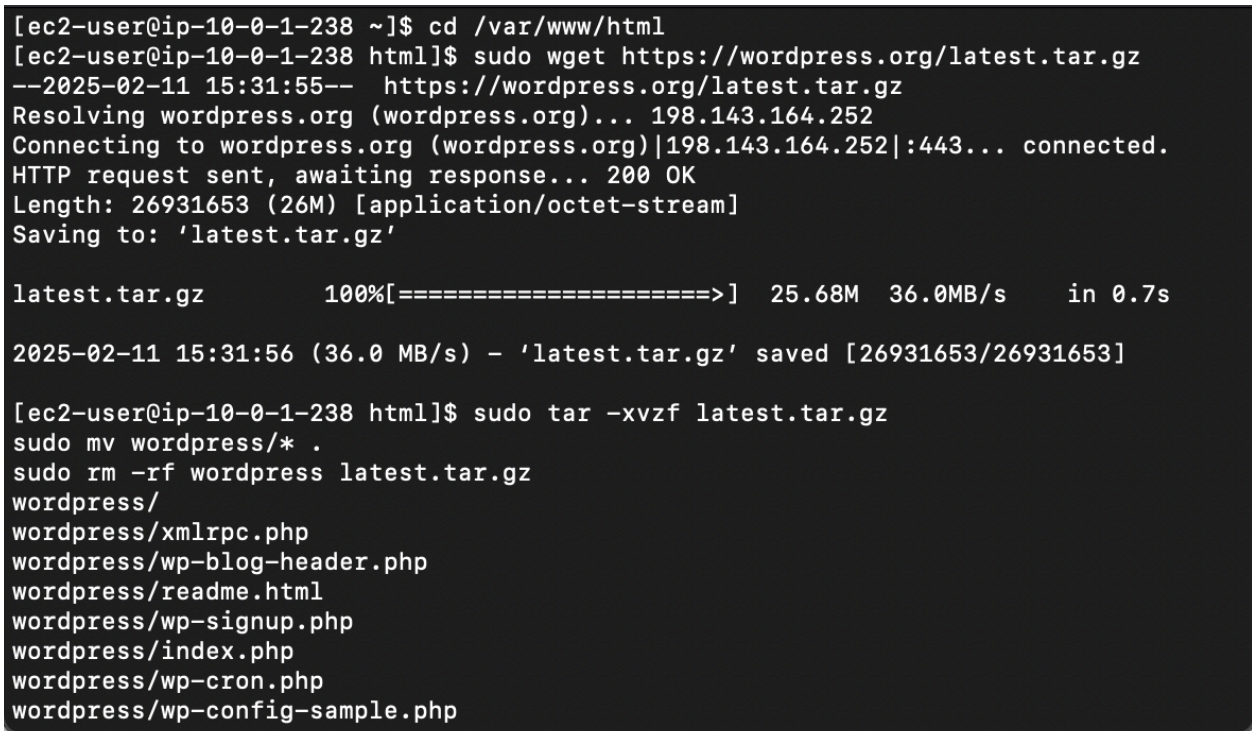
Cancel | Preview changes | Save rules

Step 4: Connecting EC2 and RDS

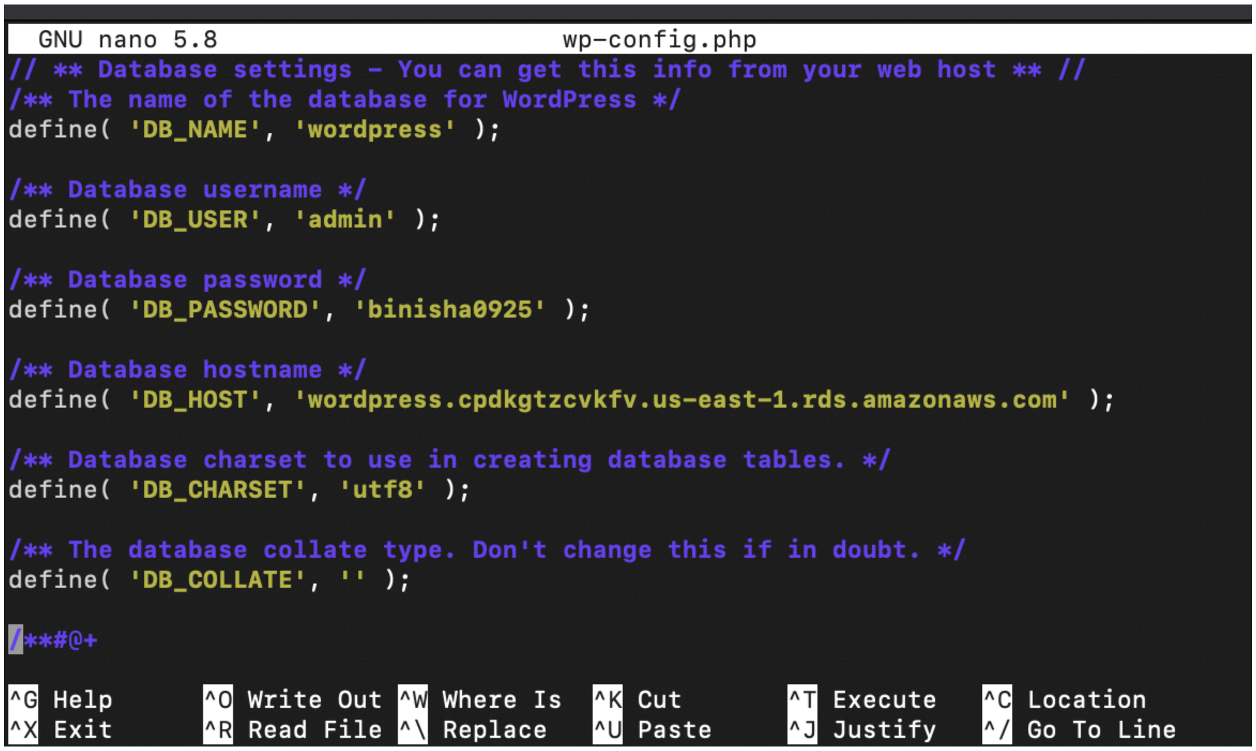
- Updated WordPress **wp-config.php** file to use RDS endpoint:



```
[ec2-user@ip-10-0-1-238 ~]$ mysql -h wordpress.cpdkgtzcvkfv.us-east-1.rds.amazonaws
.com -u admin -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 32
Server version: 11.4.4-MariaDB managed by https://aws.amazon.com/rds/

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)]> CREATE DATABASE wordpress;
Query OK, 1 row affected (0.003 sec)

MariaDB [(none)]> SHOW DATABASES;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| innodb             |
| mysql              |
| performance_schema |
| sys                |
| wordpress          |
```



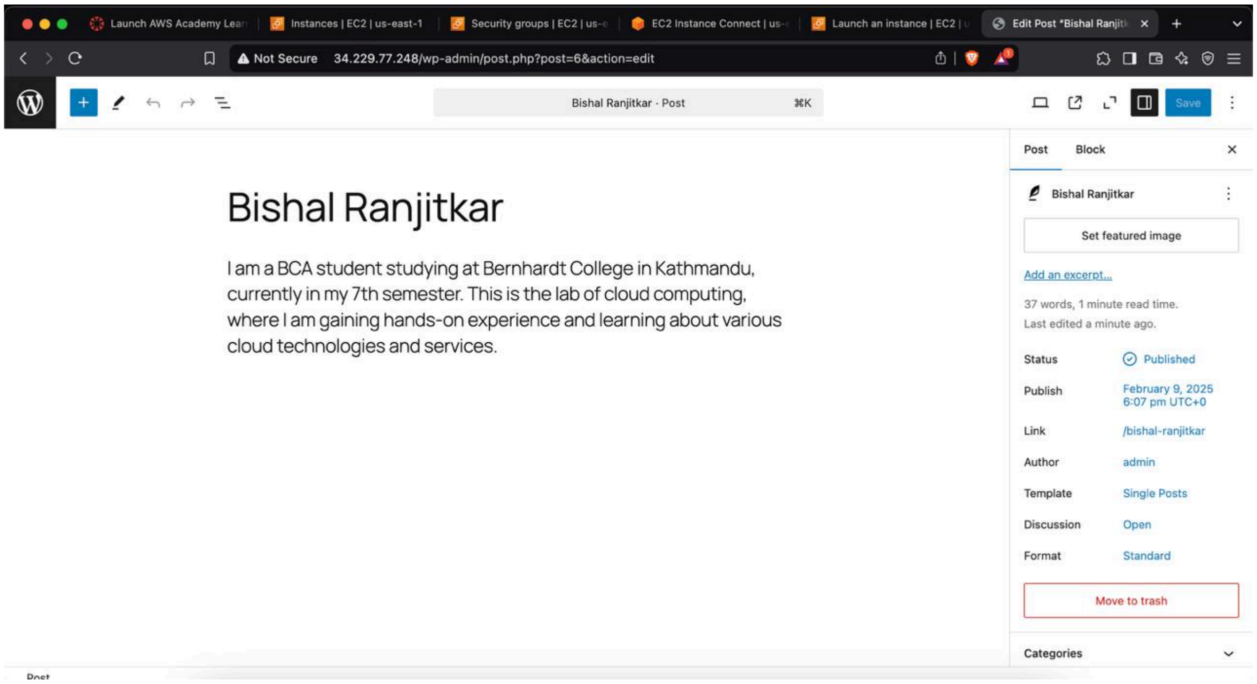
```
[ec2-user@ip-10-0-1-238 ~]$ cd /var/www/html
[ec2-user@ip-10-0-1-238 html]$ sudo wget https://wordpress.org/latest.tar.gz
--2025-02-11 15:31:55--  https://wordpress.org/latest.tar.gz
Resolving wordpress.org (wordpress.org)... 198.143.164.252
Connecting to wordpress.org (wordpress.org)|198.143.164.252|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 26931653 (26M) [application/octet-stream]
Saving to: ‘latest.tar.gz’

latest.tar.gz        100%[====================>]  25.68M  36.0MB/s    in 0.7s

2025-02-11 15:31:56 (36.0 MB/s) - ‘latest.tar.gz’ saved [26931653/26931653]

[ec2-user@ip-10-0-1-238 html]$ sudo tar -xvzf latest.tar.gz
sudo mv wordpress/* .
sudo rm -rf wordpress latest.tar.gz
wordpress/
wordpress/xmlrpc.php
wordpress/wp-blog-header.php
wordpress/readme.html
wordpress/wp-signup.php
wordpress/index.php
wordpress/wp-cron.php
wordpress/wp-config-sample.php
```



```
  GNU nano 5.8                          wp-config.php
// ** Database settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'wordpress' );

/** Database username */
define( 'DB_USER', 'admin' );

/** Database password */
define( 'DB_PASSWORD', 'binisha0925' );

/** Database hostname */
define( 'DB_HOST', 'wordpress.cpdkgtzcvkfv.us-east-1.rds.amazonaws.com' );

/** Database charset to use in creating database tables. */
define( 'DB_CHARSET', 'utf8' );

/** The database collate type. Don't change this if in doubt. */
define( 'DB_COLLATE', '' );

/**#@+

^G Help       ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^/ Go To Line
```

Step 5: Deploying WordPress

- Downloaded and configured WordPress on EC2.
- Verified the setup by accessing the public IP of EC2 via a browser.



Not Secure 34.229.77.248/wp-admin/post.php?post=6&action=edit

Bishal Ranjitkar · Post

Bishal Ranjitkar

I am a BCA student studying at Bernhardt College in Kathmandu, currently in my 7th semester. This is the lab of cloud computing, where I am gaining hands-on experience and learning about various cloud technologies and services.

Post | Block

Bishal Ranjitkar

Set featured image

Add an excerpt...

37 words, 1 minute read time.
Last edited a minute ago.

Status	Published
Publish	February 9, 2025 6:07 pm UTC+0
Link	/bishal-ranjitkar
Author	admin
Template	Single Posts
Discussion	Open
Format	Standard

Move to trash

Categories

Launch AWS Academy Lear
Instances | EC2 | us-east-1
Security groups | EC2 | us-e
EC2 Instance Connect | us-
Launch an instance | EC2 |
Edit Post "Bishal Ranjitk
⌘K
Save
Post

7. Conclusion

This project successfully deployed a two-tier application using AWS services. The setup ensures separation of the application and database layers while maintaining security and scalability.