

Module Assignment - A complete guide to AWS VPC and EC2: Building a reliable WordPress website architecture

Problem Statement:

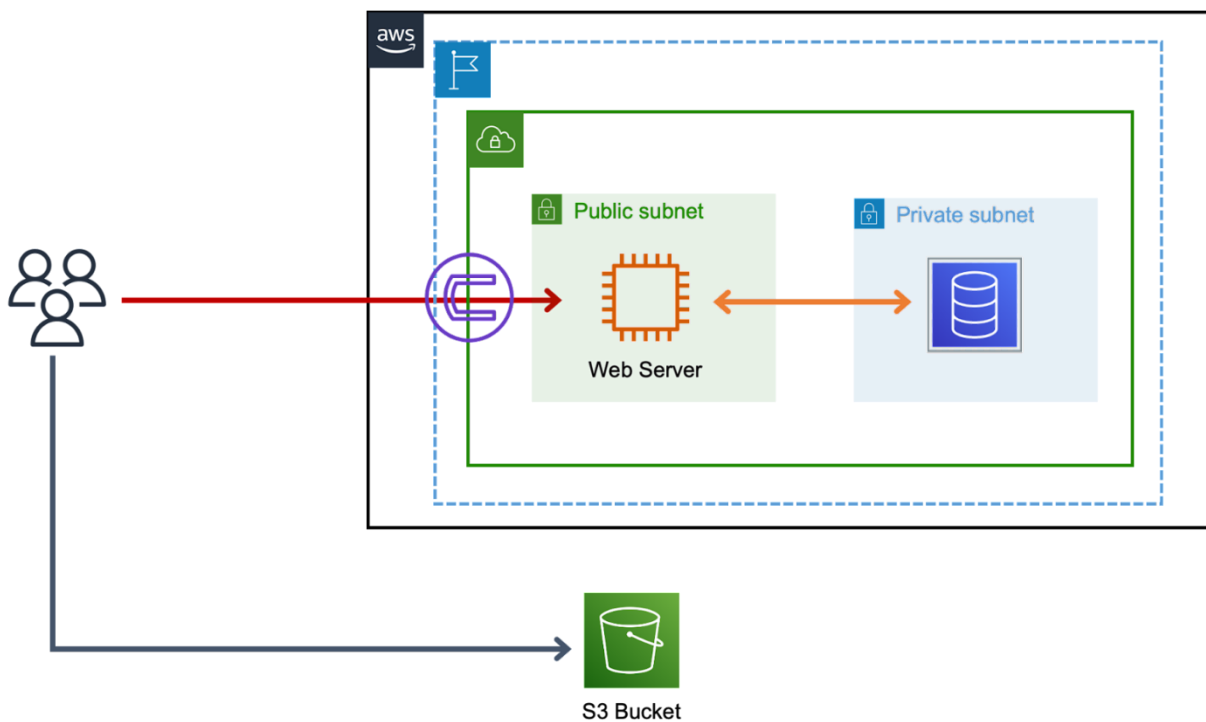
I am going to introduce an AWS architecture of an application. This is a complete application that you will be building in a step-by-step manner. With every module you complete, you will be adding a new service or resource to your architecture. But before we start, let's look at the prerequisites.

Prerequisites:

1. **An AWS account** with privileges to create IAM roles, AWS VPCs, EC2 instances, and RDS databases.

Problem:

In this lab, you will create the VPC inside your AWS account. As you know, before you create VPC, you need to select a region by considering factors like cost, compliance, latency, etcetera. You will also need two subnets inside your VPC. A public subnet to host your web application. For that, you will launch an EC2 Instance. Also, a private subnet to deploy your RDS MySQL database. Then, you will set up the connection between RDS and EC2, then install the WordPress website on the instance. Finally, you will start hosting a simple WordPress website on an EC2 instance and export the static assets into the S3 bucket.



Architecture:

- **Create a VPC inside your AWS account.**

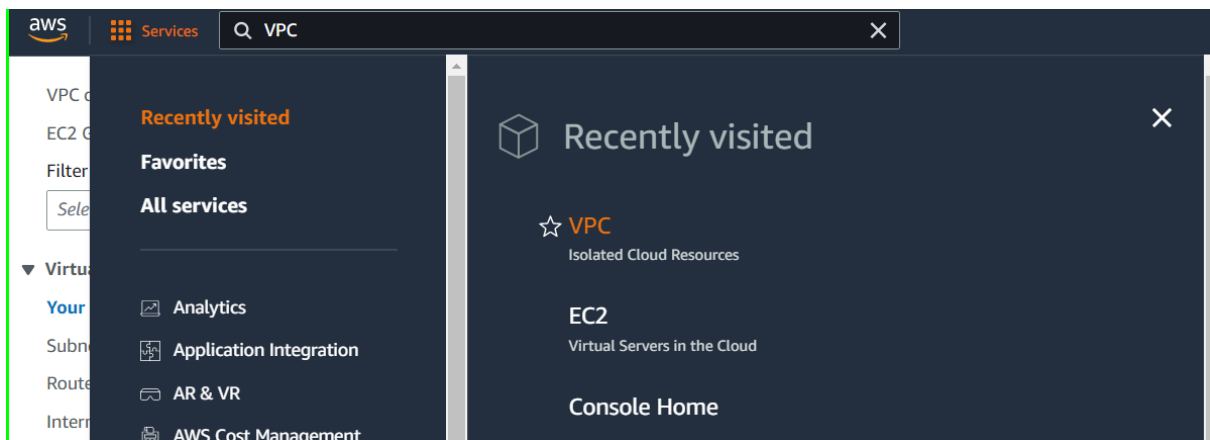
What is VPC??

VPC stands for Virtual Private Cloud, which is a virtual network infrastructure provided by Amazon Web Services (AWS). It enables you to create a logically isolated section of the AWS cloud where you can launch resources such as EC2 instances, RDS databases, and more.

A VPC allows you to have complete control over your network configuration, including IP address ranges, subnets, and routing tables. You can also create security groups and network access control lists (ACLs) to control traffic to and from your instances.

With a VPC, you can extend your data center into the cloud and connect to it securely using an IPsec VPN or AWS Direct Connect. Additionally, you can integrate your VPC with other AWS services like Amazon S3, AWS Lambda, and more.

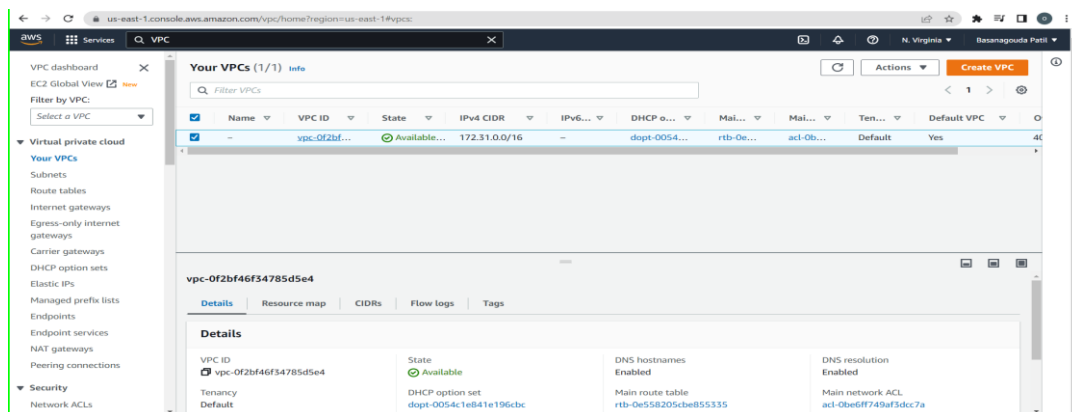
Step 1 – Open AWS account, search VPC in search bar and click on VPC



Before creating an own VPC, lets understand about default VPC

By default, when you create a new AWS account, a default VPC (Virtual Private Cloud) is created for you in each AWS region. The default VPC is a logically isolated virtual network within the AWS Cloud that you can use to launch your AWS resources, such as EC2 instances, RDS databases, and more.

The default VPC comes preconfigured with several default settings, including an Internet Gateway and a default subnet in each Availability Zone within the region. This means that you can launch your resources in the default VPC without having to worry about configuring networking settings.



Step 2 - Click on create VPC, select VPC only and enter IPv4.

Create VPC Info

A VPC is an isolated portion of the AWS Cloud populated by AWS objects, such as Amazon EC2 instances.

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.

Assg-VPC

IPv4 CIDR block Info
☒ IPv4 CIDR manual input
☐ IPAM-allocated IPv4 CIDR block

IPv4 CIDR
10.0.0.0/16

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

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 Assg-VPC Remove

Add new tag
You can add 49 more tags.

Cancel Create VPC

- A Main Route table will be created automatically but it has only local in target group.

Route tables (1/2) Info

Filter route tables

	Name	Route table ID	Explicit subn...	Edge a...	Main	VPC	Owner ID
<input type="checkbox"/>	-	rtb-0e558205cbe85...	-	-	Yes	vpc-0f2bf46f34785d5e4	404502750219
<input checked="" type="checkbox"/>	Assg-Ro...	rtb-0412572e426f7...	-	-	Yes	vpc-0a9516d951e5e1328 Assg-VPC	404502750219

rtb-0412572e426f7c46 / Assg-Route table

Details Routes Subnet associations Edge associations Route propagation Tags

Routes (1)

Filter routes Both

Destination	Target	Status	Propagated
10.0.0.0/16	local	Active	No

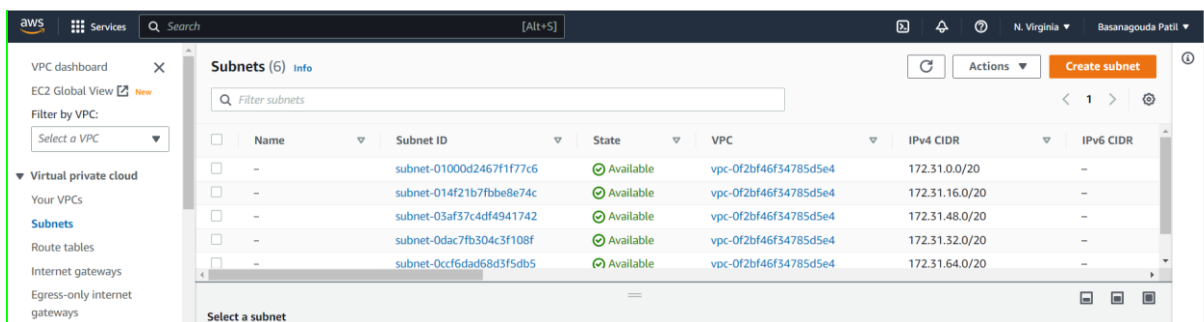
- **Create a Public Subnet in your Custom VPC**

What is Public Subnet?

A public subnet is a portion of a computer network that is accessible from the Internet, meaning it has a public IP address that can be reached by anyone on the Internet. In a public subnet, resources like web servers, email servers, and other services that need to be publicly accessible are typically deployed.

Public subnets are often used in cloud computing environments like Amazon Web Services (AWS), where they are paired with a private subnet to create a virtual private cloud (VPC). The

private subnet is used to host resources that should not be accessible from the Internet, such as databases, backend servers, and internal applications.



In above pic you can see subnets which are create by AWS when default VPC is created.

Step 1 – Click on Create subnet, select your custom VPC, Give name to subnet, select Availability zone according your convince and assign IPV4 CDIR block to this subnet.

A screenshot of the AWS 'Create subnet' wizard. The 'VPC' section shows 'VPC ID' as 'vpc-0a9516d951e5e1328 (Assg-VPC)' and 'Associated VPC CIDRs' as '10.0.0.0/16'. The 'Subnet settings' section shows 'Subnet name' as 'Assg-Public subnet', 'Availability Zone' as 'US East (N. Virginia) / us-east-1a', and 'IPv4 CIDR block' as '10.0.0.0/20'. There is a 'Tags - optional' section with a key 'Name' and value 'Assg-Public subnet'. At the bottom, there are 'Cancel' and 'Create subnet' buttons.

Step 2 – Make Auto-assign IP enable because as name suggest it is Public subnet.

VPC > Subnets > subnet-094f15fda78cf901d > Edit subnet settings

Edit subnet settings [Info](#)

Subnet

Subnet ID	Name
subnet-094f15fda78cf901d	Assg-Public subnet

Auto-assign IP settings [Info](#)

Enable the auto-assign IP settings to automatically request a public IPv4 or IPv6 address for a new network interface in this subnet.

☒ Enable auto-assign public IPv4 address [Info](#)

☐ Enable auto-assign customer-owned IPv4 address [Info](#)
Option disabled because no customer owned pools found.

Resource-based name (RBN) settings [Info](#)

Specify the hostname type for EC2 instances in this subnet and optional RBN DNS query settings.

☐ Enable resource name DNS A record on launch [Info](#)

☐ Enable resource name DNS AAAA record on launch [Info](#)

Hostname type [Info](#)

☐ Resource name

☒ IP name

DNS64 settings

Enable DNS64 to allow IPv6-only services in Amazon VPC to communicate with IPv4-only services and networks.

☐ Enable DNS64 [Info](#)

Cancel Save

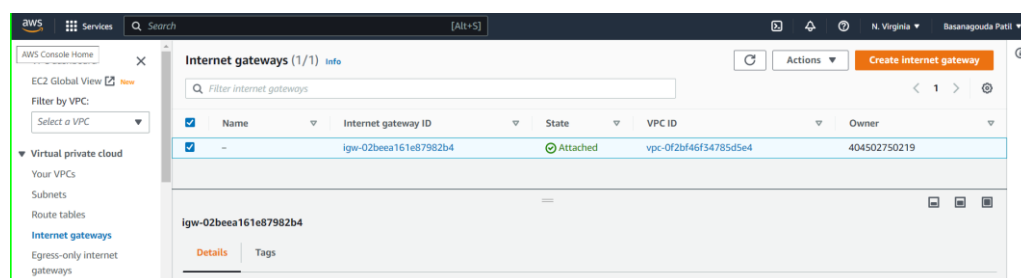
Step 3 – Create an Internet gateway to your Custom VPC

What is an Internet Gateway?

An Internet Gateway (IGW) is a horizontally scaled, redundant, and highly available component in Amazon Web Services (AWS) that allows communication between resources in a VPC and the internet.

An Internet Gateway is a virtual router that connects a VPC to the internet. It provides a target for traffic destined for the public internet from instances in the VPC and a source for traffic originating from the internet and intended for instances in the VPC.

It is an AWS-managed component that is attached to your VPC and It acts as a gateway between your VPC & the internet, basically the outside world.



In above pic you can see an internet gateway already create which is create by AWS when default are VPC is created.

- Click on Create internet gateway, give name to it and create internet gateway

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.

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

You can add 49 more tags.

- Now, Attach this internet gate to your custom VPC

AWS Services Search [Alt+S] N. Virginia Basanagouda Patil

VPC dashboard EC2 Global View [New](#)

Filter by VPC: Select a VPC

Virtual private cloud

Your VPCs

Subnets

Route tables

Internet gateways

Egress-only internet

The following Internet gateway was created: igw-0e961d3e026f284b0 - Assg-internet gateway. You can now attach to a VPC to enable the VPC to communicate with the internet. [Attach to a VPC](#)

VPC > Internet gateways > igw-0e961d3e026f284b0

igw-0e961d3e026f284b0 / Assg-internet gateway

Details [Info](#)

Internet gateway ID	State	VPC ID	Owner
igw-0e961d3e026f284b0	Detached	-	404502750219

Actions

- Attach to VPC
- Detach from VPC
- Manage tags
- Delete

VPC > Internet gateways > Attach to VPC (igw-0e961d3e026f284b0)

Attach to VPC (igw-0e961d3e026f284b0) [Info](#)

VPC
Attach an internet gateway to a VPC to enable the VPC to communicate with the internet. Specify the VPC to attach below.

Available VPCs
Attach the internet gateway to this VPC.

[AWS Command Line Interface command](#)

AWS Services Search [Alt+S] N. Virginia Basanagouda Patil

VPC dashboard EC2 Global View [New](#)

Filter by VPC: Select a VPC

Virtual private cloud

Your VPCs

Subnets

Route tables

Internet gateways

Internet gateways (2) [Info](#)

Filter Internet gateways

	Name	Internet gateway ID	State	VPC ID	Owner
<input type="checkbox"/>	-	igw-02brea161e87982b4	Attached	vpc-0f2bf46f34785d5e4	404502750219
<input type="checkbox"/>	Assg-internet gate...	igw-0e961d3e026f284b0	Attached	vpc-0a9516d951e5e1328 Assg-VPC	404502750219

Select an internet gateway above

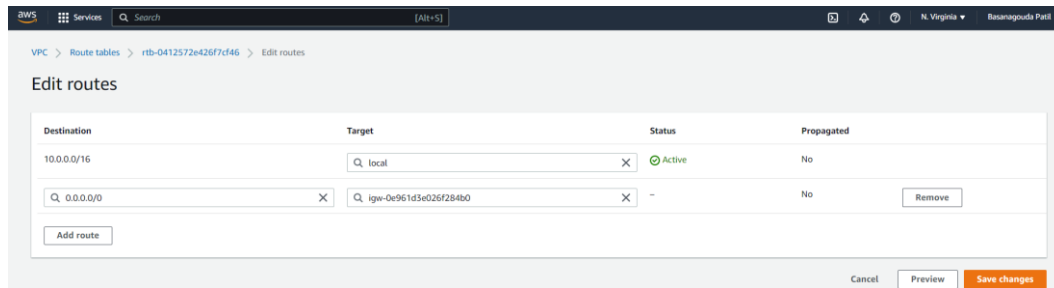
Step 4 – Add Internet way id to main route table

What is Route table ?

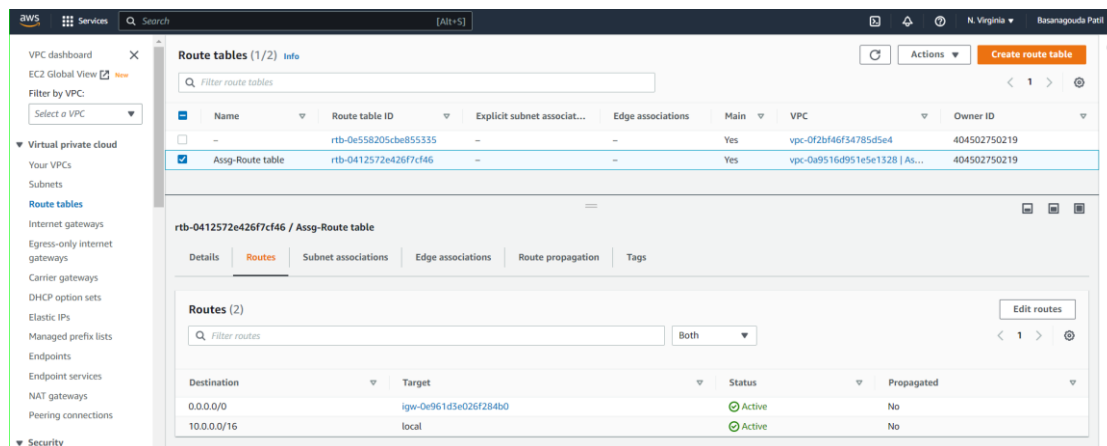
In Amazon Web Services (AWS), the main route table is the default route table that is created when you create a VPC. Every subnet that you create in the VPC is associated with this default route table unless you explicitly associate it with a custom route table.

The main route table contains rules that define how traffic is directed within the VPC. It is used to route traffic between subnets within the VPC and to control access to and from the internet. By default, the main route table has a route that sends all traffic to a local route for communication within the VPC.

You can add, modify or delete the routes in the main route table to control the flow of traffic within your VPC. For example, you can add a route that sends traffic to an Internet Gateway to allow resources in a public subnet to access the internet. You can also add a route that sends traffic to a NAT Gateway to allow resources in a private subnet to access the internet.



Select route table and go to edit routes in that add destination as 0.0.0.0/0 and past your internet gateway id in target and save the changes.



- Create a Ec2 instance in public subnet of custom VPC

Give name to that instance, select instance type, create a key pair and in network setting select your custom VPC and public subnet of it.

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

[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

Instance type [Info](#)

Instance type

t2.micro

Family: t2 1 vCPU 1 GB Memory

On-Demand Windows pricing: 0.0162 USD per Hour

On-Demand SUSE pricing: 0.0116 USD per Hour

On-Demand RHEL pricing: 0.0716 USD per Hour

On-Demand Linux pricing: 0.0116 USD per Hour

Free tier eligible

Compare instance types

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

[Create new key pair](#)

Network settings [Info](#)

VPC - required [Info](#)

[10.0.0.0/16](#)

Subnet [Info](#)

[Assg-Public subnet](#)

VPC: vpc-0a9516d951e5e1328 Owner: 404502750219
Availability Zone: us-east-1a IP addresses available: 4091 CIDR: 10.0.0.0/20

Auto-assign public IP [Info](#)

Firewall (security groups) [Info](#)

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

☐ Create security group
 ☒ Select existing security group

Summary

Number of instances [Info](#)

Software Image (AMI)

Amazon Linux 2 Kernel 5.10 AMI...[read more](#)
ami-0dfcb1ef8550277af

Virtual server type (instance type)

t2.micro

Firewall (security group)

default

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 AMIs per month, 30 GiB of EBS storage, 2 million I/Os, 1 GiB of snapshots, and 100 GB of bandwidth to the internet.

Cancel [Launch instance](#)

- **Create a Private Subnet in your Custom VPC**

In Amazon Web Services (AWS), a private subnet is a subnet within a VPC that does not have a direct route to the Internet, meaning it is not associated with an Internet Gateway. Instances within a private subnet can communicate with other instances within the same VPC or with other VPCs through a VPN connection or a VPC peering connection.

Typically, resources that require a high degree of security, such as databases, backend servers, or internal applications, are deployed in private subnets. Since they are not directly accessible from the Internet, they are less vulnerable to attacks from the public Internet. Instead, they can only be accessed by authorized users or resources within the same VPC.

Step 1 – click on create a subnet then Select your custom VPC, give name to subnet, select availability zone and assign a IPV4 CIDR block to this subnet.

aws Services Search [Alt+S]

VPC > Subnets > Create subnet

Create subnet [Info](#)

VPC

VPC ID
Create subnets in this VPC.

vpc-0a9516d951e5e1328 (Assg-VPC) ▼

Associated VPC CIDRs

IPv4 CIDRs

10.0.0.0/16

Subnet settings

Specify the CIDR blocks and Availability Zone for the subnet.

Subnet 1 of 1

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

Assg -Private subnet

The name can be up to 255 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 CIDR block [Info](#)

10.0.16.0/20 X

► Tags - optional

Remove

Add new subnet

Cancel Create subnet

Note –

- Here, Auto-assign IP will be disabled because as name suggest it is Private subnet.
- Internet gateway is not required as it is private subnet.

Step 2 – Create a Route table and keep only local target in it.

AWS Console Home > 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.

Assg-route table for pvt subnet

VPC
The VPC to use for this route table.

vpc-0a9516d951e5e1328 (Assg-VPC) ▼

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.

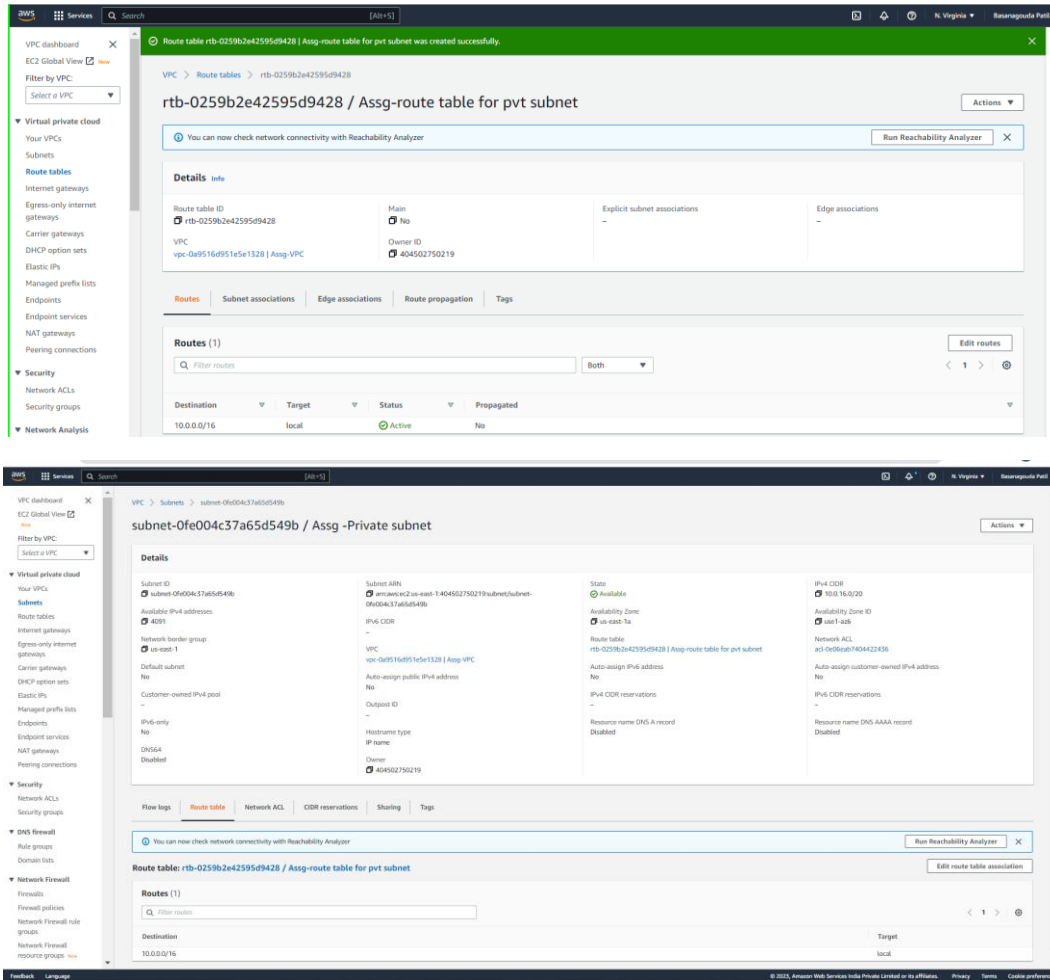
Key Value - optional

Q Name X Q Assg-route table for pvt subnet X Remove

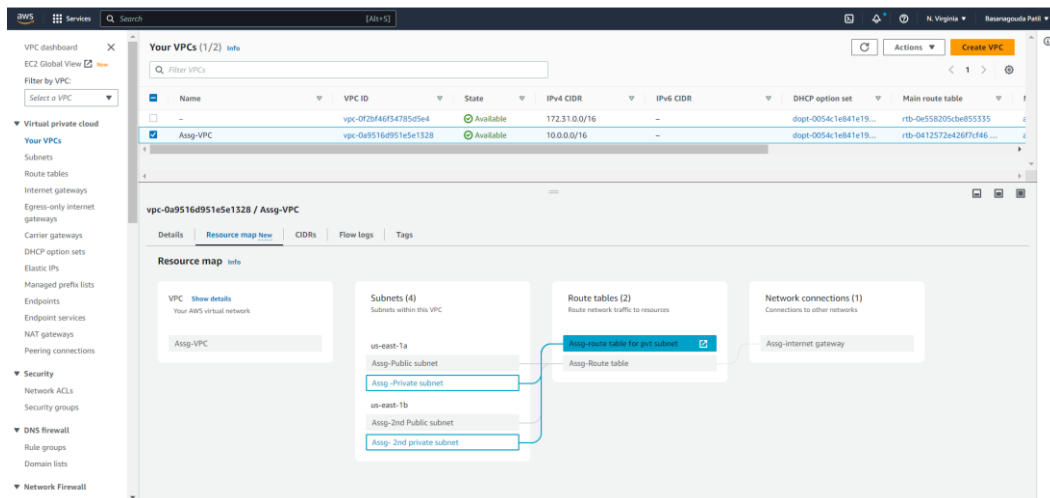
Add new tag

You can add 49 more tags.

Cancel Create route table



In below image you can see resource map, Public subnets are attached to Route table and Internet gateway, where Private subnets are attached to only Route table without Internet gateway access.



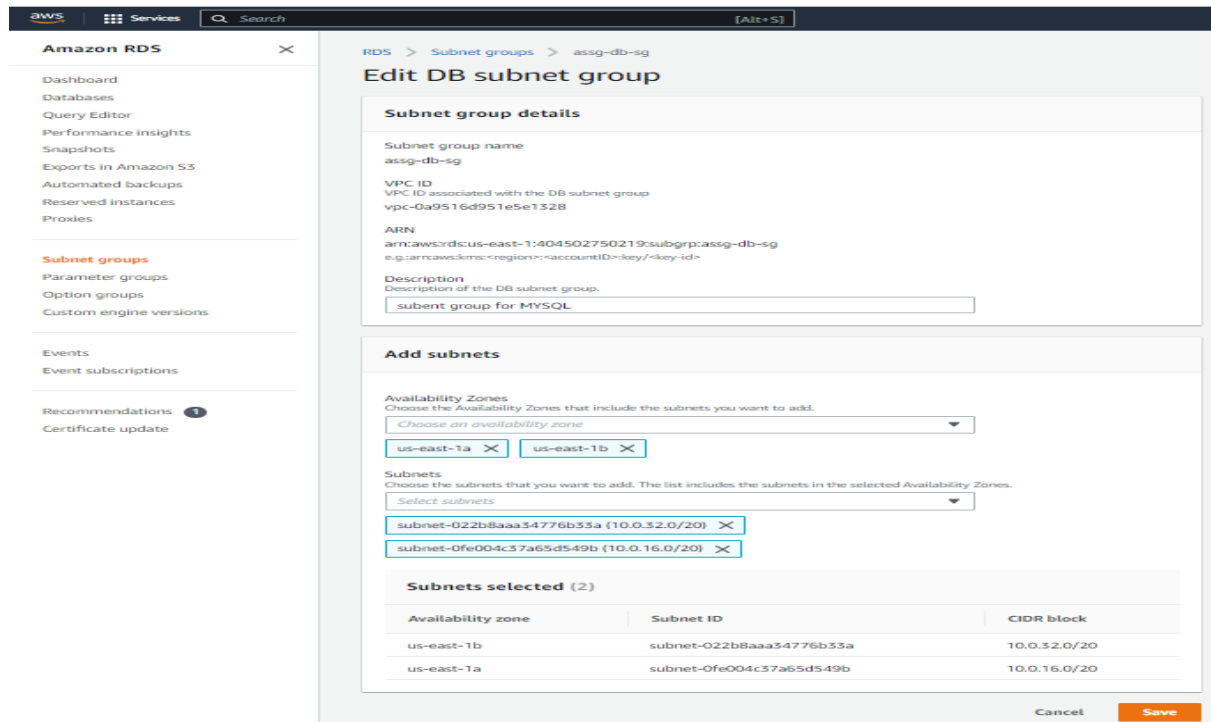
- **Create an RDS MySQL database in Private subnet of custom VPC.**

RDS - RDS (Relational Database Service) is a web service provided by AWS (Amazon Web Services) that makes it easy to set up, operate, and scale a relational database in the cloud. RDS supports multiple database engines, including MySQL, PostgreSQL, Oracle, and SQL Server.

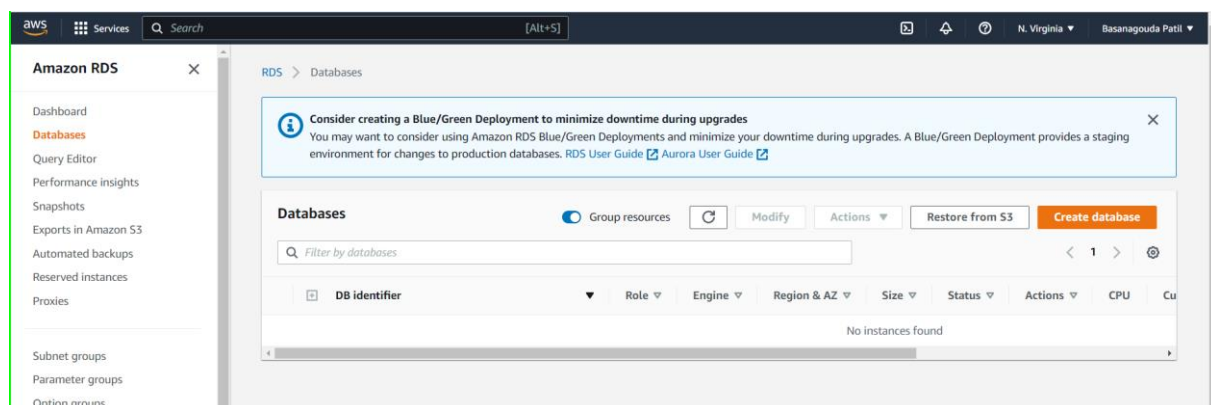
RDS allows users to manage their databases in a cloud-based environment without the need for extensive hardware or software resources. It provides automated backups, software patching, and scalable storage, making it an efficient and cost-effective solution for database management.

When using RDS in AWS, users can select from several instance types, which determine the computing and memory capacity of the database instance. RDS also supports read replicas, which allow users to create multiple copies of their database for read-intensive workloads.

Step 1 - Before Creating RDS, First create a subnet group consisting of only private subnets in RDS.



Step 2 – Create database by click on create database.



Step 3 – Select Standard create, choose mysql and version, and I select free tier template

RDS > Create database

Create database


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 

☐ MariaDB 

☐ PostgreSQL 

☐ Oracle 

☐ Microsoft SQL Server 

Edition

☒ MySQL Community

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

Edition

☒ MySQL Community

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

► Show filters

Engine Version

MySQL 8.0.28 ▼

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. [Info](#)

Step 4 – In setting type a name for your database, type master username and type of password.

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

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.

[Info](#) 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](#)

.....

Step 5 – In Instance configuration select Db instance class and select storage type and size.

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

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

Step 6 – In connectivity choose your VPC and Public access NO.

aws Services Search [Alt+S]

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.

assg-db-sg

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

assg-rds-mysql-sg

Availability Zone [Info](#)

No preference

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 Proxy 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 port [Info](#)
TCP/IP port that the database will use for application connections.

3306

Step 7 – Select Database authentication as password authentication, and click on click on database

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.

Monitoring

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 off, backtrace 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.

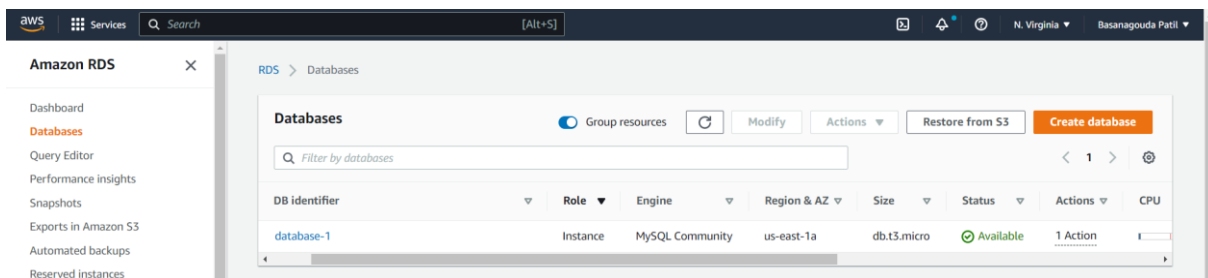
[Learn more about AWS Free Tier.](#)

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](#).

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.

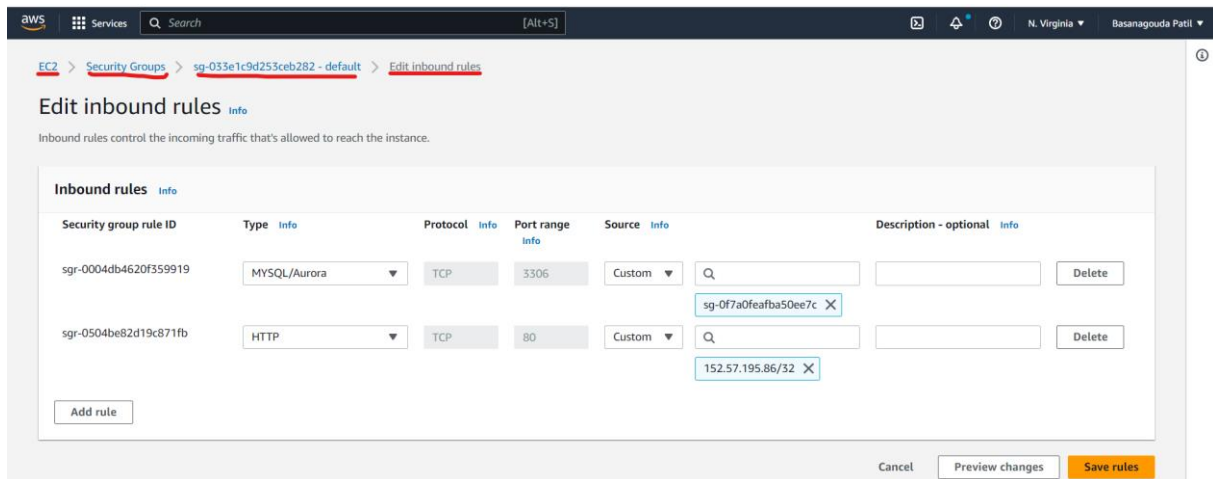
Cancel

Create database

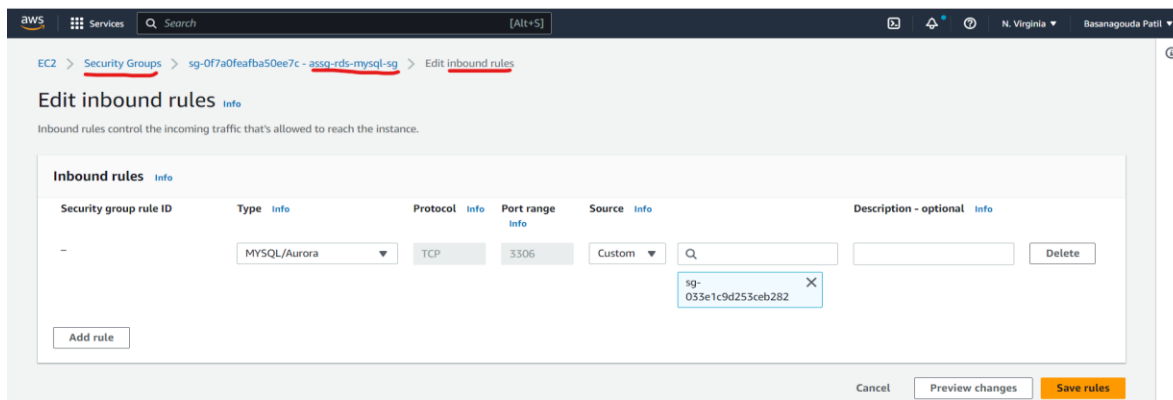


Step 8 - Now, we will continue further and modify security group of RDS and EC2 instance. So, we need EC2 instance. So please go ahead and create EC2 instance and all missing AWS resources which we created so far.

- select public-instance-sg,
- Click Edit inbound rules button
- Click Add rule, For Type, select MYSQL/Aurora,
- For Source, select custom and find the assg-rds-mysql-sg , and click Save rules
- Click Add rule, For Type, select HTTP,
- For Source, select My IP, and click Save rules

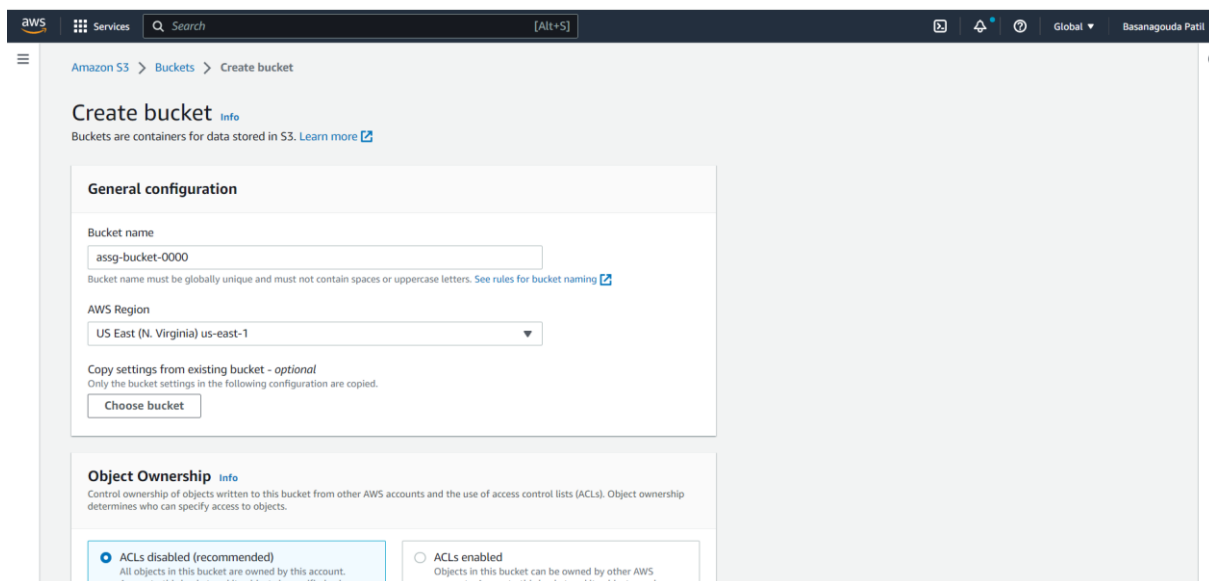


- select db-sg ,
- Click Edit inbound rules button
- Delete existing rule
- Click Add rule, For Type, select MYSQL/Aurora,
- For Source, select custom and find the public-instance-sg , and click Save rules

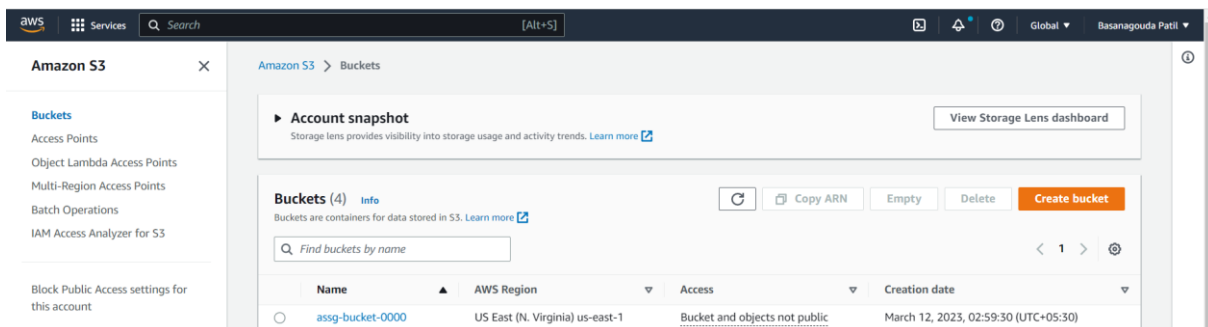
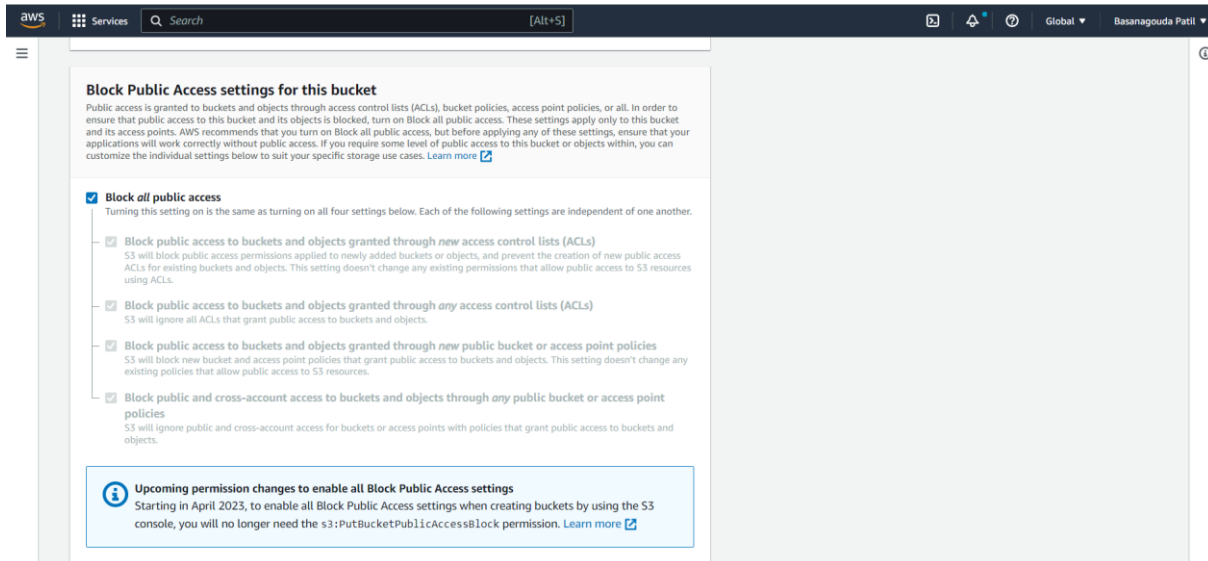


- **Create an S3 bucket with unique name**

Step 1 – Search S3 in search bar and create an S3 bucket with the “assg-bucket-**<random-number>**” name.



Step 2 – Block Public Access to this bucket and click on create bucket.



- **Set Up the Wordpress Environment**

Step 1 - Select the Public instance and click Connect,

- In Connect to instance page, click SSH client on menu and you can find the connection command in Example.
- Windows User: follow the step in this deck to connect EC2 instance by Putty
- MacOS User: open a terminal on your computer, navigate to the folder where your.pem key file is stored, and paste the connection command.


```

ec2-user@ip-10-0-5-142:~
login as: ec2-user
Authenticating with public key "linux 5/12"
Last login: Sun Mar 12 19:18:27 2023 from ec2-18-206-107-28.compute-1.amazonaws.com

  _ | _ | _ )
  _ | ( _ | /   Amazon Linux 2 AMI
  _ | \ _ | _ |

https://aws.amazon.com/amazon-linux-2/
16 package(s) needed for security, out of 23 available
Run "sudo yum update" to apply all updates.
[ec2-user@ip-10-0-5-142 ~]$

```

Step 2 - After connection successes, run the following command to install LAMP stack (Linux, Apache, Mariadb and php) on your instance

- sudo yum install httpd -y
- sudo service httpd start
- sudo service httpd status

- sudo yum install mariadb-server -y
- sudo service mariadb start
- sudo service mariadb status

- sudo amazon-linux-extras install php8.0 -y
- sudo service php-fpm start
- sudo service php-fpm status

- sudo service httpd restart
- sudo service mariadb restart
- sudo service php-fpm restart

Here, I'm installing LAMP using shell script and we have to give execute permission to that LAMP file.

```

GNU nano 2.9.8                                     LAMP.sh
echo"Hello ,Lets start LAMP installation"

sleep 2

echo "lets Install A from LAMP, A- Apache"
Sleep 2
sudo yum install httpd -y
sudo service httpd start
sudo service httpd status

echo "lets Install M from LAMP, M- Mariadb"
sleep 2
sudo yum install mariadb-server -y
sudo service mariadb start
sudo service mariadb status

echo "lets Install P from LAMP, P- php"
sleep 2
sudo amazon-linux-extras install php8.1 -y
sudo service php-fpm start
sudo service php-fpm status

echo"After installing all services, we have to restart them onces"
sleep 2
sudo service httpd restart
sudo service mariadb restart
sudo service php-fpm restart

[ec2-user@ip-10-0-5-142 ~]$ sudo nano LAMP.sh
[ec2-user@ip-10-0-5-142 ~]$ sudo chmod 777 LAMP.sh
[ec2-user@ip-10-0-5-142 ~]$ ./LAMP.sh
./LAMP.sh: line 1: echoHello ,Lets start LAMP installation: command not found
lets Install A from LAMP, A- Apache
./LAMP.sh: line 6: Sleep: command not found
Loaded plugins: extras_suggestions, langpacks, priorities, update-motd
amzn2-core
amzn2extra-docker
amzn2extra-kernel-5.10
Resolving Dependencies
--> Running transaction check
--> Package httpd.x86_64 0:2.4.55-1.amzn2 will be installed
--> Processing Dependency: httpd-tools = 2.4.55-1.amzn2 for package: httpd-2.4.55-1.amzn2.x86_64
--> Processing Dependency: httpdfilesystem = 2.4.55-1.amzn2 for package: httpd-2.4.55-1.amzn2.x86_64
--> Processing Dependency: system-logos-httpd for package: httpd-2.4.55-1.amzn2.x86_64
--> Processing Dependency: mod_http2 for package: httpd-2.4.55-1.amzn2.x86_64
--> Processing Dependency: httpdfilesystem for package: httpd-2.4.55-1.amzn2.x86_64

```

Step 3 - Set the environment variable of MySQL in your computer, replace `<your-endpoint>` to the endpoint which can be found in [RDS console](#)→ your database.

- `sudo mysql -h <your-endpoint> -u <username> -p <password>`
- create a database called `database1`

```
[ec2-user@ip-10-0-5-142 wordpress]$ sudo mysql -h database-1.cbtmpsiri9grk.us-east-1.rds.amazonaws.com -u admin -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MySQL connection id is 448
Server version: 8.0.28 Source distribution

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

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

MySQL [(none)]> create database database1;
Query OK, 1 row affected (0.00 sec)

MySQL [(none)]> show databases;
+-----+
| Database |
+-----+
| database1 |
| information_schema |
| mysql |
| performance_schema |
| sys |
+-----+
5 rows in set (0.00 sec)

MySQL [(none)]> exit
Bye
[ec2-user@ip-10-0-5-142 wordpress]$
```

Step 4 – Go to html using path `cd /var/www/html` then, download the WordPress module and unzip it

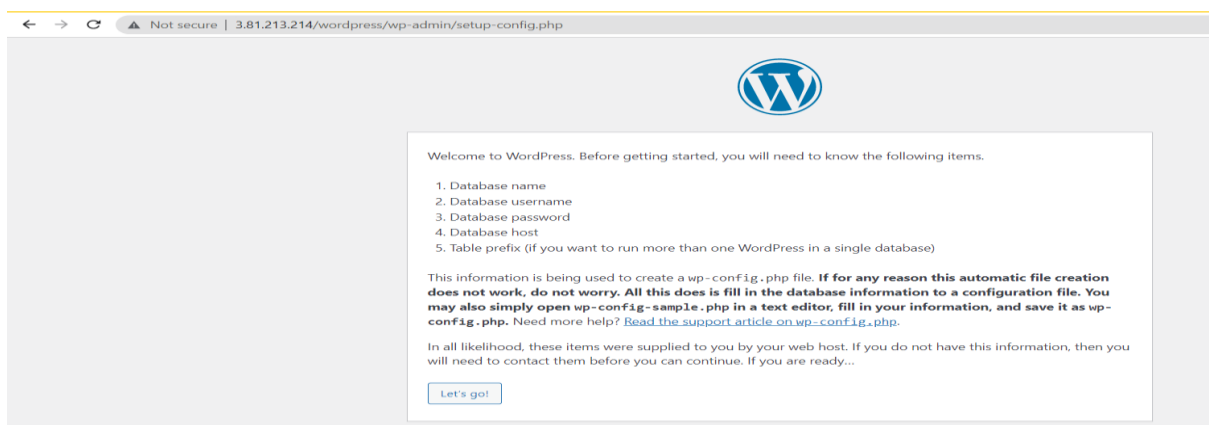
- `wget https://wordpress.org/latest.tar.gz`
- `tar -xzf latest.tar.gz`

```
[ec2-user@ip-10-0-5-142 ~]$ cd /var/www/html/
[ec2-user@ip-10-0-5-142 html]$ sudo wget https://wordpress.org/latest.tar.gz
--2023-03-13 07:49:46-- 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: 22751086 (22M) [application/octet-stream]
Saving to: 'latest.tar.gz'

100%[=====] 22,751,086 38.3MB/s in 0.6s

2023-03-13 07:49:46 (38.3 MB/s) - 'latest.tar.gz' saved [22751086/22751086]

[ec2-user@ip-10-0-5-142 html]$ ls
latest.tar.gz
[ec2-user@ip-10-0-5-142 html]$ sudo tar -xzf latest.tar.gz
[ec2-user@ip-10-0-5-142 html]$ ls
latest.tar.gz  wordpress
[ec2-user@ip-10-0-5-142 html]$
```



Step 5 - Move into WordPress folder and backup the default config file

- `cd wordpress`
- `cp wp-config-sample.php wp-config.php`

```

[ec2-user@ip-10-0-5-142 html]$ ls
latest.tar.gz  wordpress
[ec2-user@ip-10-0-5-142 html]$ cd wordpress/
[ec2-user@ip-10-0-5-142 wordpress]$ ls
index.php  readme.html  wp-admin  wp-comments-post.php  wp-content  wp-includes  wp-load.php  wp-mail.php  wp-signup.php  xmlrpc.php
license.txt  wp-activate.php  wp-blog-header.php  wp-config-sample.php  wp-cron.php  wp-links-opml.php  wp-login.php  wp-settings.php  wp-trackback.php
[ec2-user@ip-10-0-5-142 wordpress]$ sudo cp wp-config-sample.php wp-config.php
[ec2-user@ip-10-0-5-142 wordpress]$ ls
index.php  readme.html  wp-admin  wp-comments-post.php  wp-config-sample.php  wp-cron.php  wp-links-opml.php  wp-login.php  wp-settings.php  wp-trackback.php
license.txt  wp-activate.php  wp-blog-header.php  wp-config.php  wp-content  wp-includes  wp-load.php  wp-mail.php  wp-signup.php  xmlrpc.php
[ec2-user@ip-10-0-5-142 wordpress]$

```

Step 6 -After that use nano to edit the wp-config.php file

- nano wp-config.php

Step 7 - Modify the following script into the correct value:

DB_NAME: 'wordpress' • DB_USER: 'wordpress' • DB_PASSWORD: 'wordpress-pass' • DB_HOST: your RDS endpoint and save it.

```

GNU nano 2.9.8                                wp-config.php
<?php
/**
 * The base configuration for WordPress
 *
 * The wp-config.php creation script uses this file during the installation.
 * You don't have to use the web site, you can copy this file to "wp-config.php"
 * and fill in the values.
 *
 * This file contains the following configurations:
 *
 * * Database settings
 * * Secret keys
 * * Database table prefix
 * * ABSPATH
 *
 * @link https://wordpress.org/support/article/editing-wp-config-php/
 *
 * @package WordPress
 */

/** Database settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define( 'DB_NAME', 'database1' );

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

/** Database password */
define( 'DB_PASSWORD', 'omkar1111' );

/** Database hostname */
define( 'DB_HOST', 'database-1.cbtsiri9grk.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', '' );

/**#@+
 * Authentication unique keys and salts.
 *
 * Change these to different unique phrases! You can generate these using
 * the {@link https://api.wordpress.org/secret-key/1.1/salt/ WordPress.org secret-key service}.
 *
 * Read 96 lines (Converted from DOS format)
 */


```

Step 8 - Select the EC2 instance, and find the Public IPv4 DNS in Details below and paste it on your browser, then you will see the setup page of WordPress.

In setup page, enter your own value in the Site Title, Username, Password and Your Email, then click Install WordPress

← → 🔍 ⚙️ ☆ 📱 ⌵

Not secure | 3.81.213.214/wordpress/wp-admin/install.php



Welcome

Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.

Information needed

Please provide the following information. Do not worry, you can always change these settings later.

Site Title

Username

Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.

Password [Hide](#)

Strong

Important: You will need this password to log in. Please store it in a secure location.

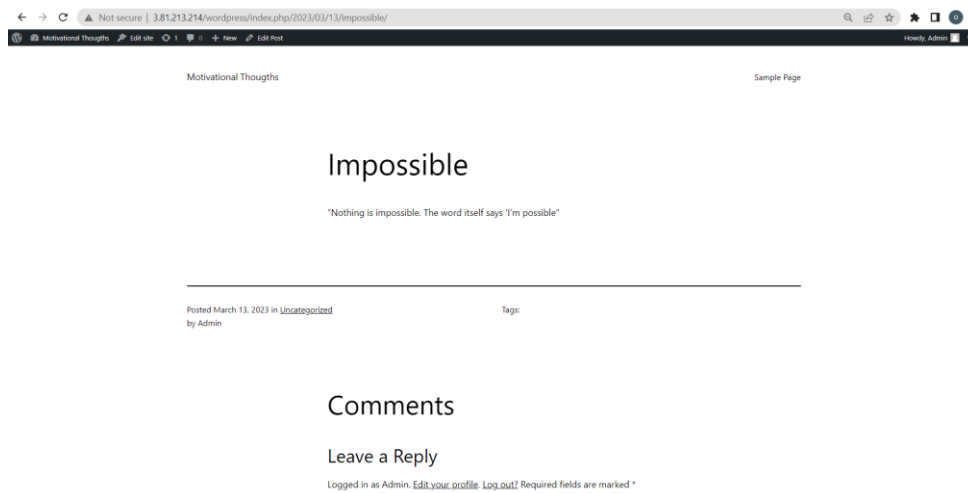
Your Email

Double-check your email address before continuing.

Search engine visibility ☐ Discourage search engines from indexing this site
It is up to search engines to honor this request.

[Install WordPress](#)

- After few seconds, it will be redirected to Login page,
- Enter your username and password, and click Login button, you will see the admin page,
- You can use Admin dashboard enhance your blog.
- Now, you can view your blog in <your ec2 domain>



Congratulations, you have successfully hosted a simple WordPress website on EC2 instance and configured all required AWS services.

In case you have any confusion, you can drop message here.

That's all for this assignment.