

Task 1:

- Create a VPC with a public subnet
- Create a new phpmyadmin EC2
- Create a Web proxy for caching

What helped me build this

https://docs.aws.amazon.com/AWSCloudFormation/latest/UserGuide/AWS_EC2.html

<https://docs.aws.amazon.com/codebuild/latest/userguide/cloudformation-vpc-template.html>

Description: This template deploys a VPC, with a pair of public and private subnets spread across two Availability Zones. It deploys an internet gateway, with a default route on the public subnets. It deploys a pair of NAT gateways (one in each AZ), and default routes for them in the private subnets.

Parameters:

EnvironmentName:

Description: An environment name that is prefixed to resource names

Type: String

VpcCIDR:

Description: Please enter the IP range (CIDR notation) for this VPC

Type: String

Default: 192.168.0.0/16

PublicSubnet1CIDR:

Description: Please enter the IP range (CIDR notation) for the public subnet in the first Availability Zone

Type: String

Default: 192.168.0.0/18

PublicSubnet2CIDR:

Description: Please enter the IP range (CIDR notation) for the public subnet in the second Availability Zone

Type: String

Default: 192.168.64.0/18

PrivateSubnet1CIDR:

Description: Please enter the IP range (CIDR notation) for the private subnet in the first Availability Zone

Type: String

Default: 192.168.128.0/18

PrivateSubnet2CIDR:

Description: Please enter the IP range (CIDR notation) for the private subnet in the second Availability Zone

Type: String

Default: 192.168.192.0/18

Resources:

VPC:

Type: AWS::EC2::VPC

Properties:

CidrBlock: !Ref VpcCIDR

EnableDnsSupport: true

EnableDnsHostnames: true

Tags:

- Key: Name

Value: !Ref EnvironmentName

InternetGateway:

Type: AWS::EC2::InternetGateway

Properties:

Tags:

- Key: Name

Value: !Ref EnvironmentName

InternetGatewayAttachment:

Type: AWS::EC2::VPCGatewayAttachment

Properties:

InternetGatewayId: !Ref InternetGateway

VpcId: !Ref VPC

PublicSubnet1:

Type: AWS::EC2::Subnet

Properties:

VpcId: !Ref VPC

AvailabilityZone: !Select [0, !GetAZs "]

CidrBlock: !Ref PublicSubnet1CIDR

MapPublicIpOnLaunch: true

Tags:

- Key: Name

Value: !Sub \${EnvironmentName} Public Subnet (AZ1)

PublicSubnet2:

Type: AWS::EC2::Subnet

Properties:

VpcId: !Ref VPC

AvailabilityZone: !Select [1, !GetAZs "]

CidrBlock: !Ref PublicSubnet2CIDR

MapPublicIpOnLaunch: true

Tags:

- Key: Name

Value: !Sub \${EnvironmentName} Public Subnet (AZ2)

PrivateSubnet1:

Type: AWS::EC2::Subnet

Properties:

VpcId: !Ref VPC

AvailabilityZone: !Select [0, !GetAZs "]

CidrBlock: !Ref PrivateSubnet1CIDR

MapPublicIpOnLaunch: false

Tags:

- Key: Name

Value: !Sub \${EnvironmentName} Private Subnet (AZ1)

PrivateSubnet2:

Type: AWS::EC2::Subnet

Properties:

VpcId: !Ref VPC

AvailabilityZone: !Select [1, !GetAZs "]

CidrBlock: !Ref PrivateSubnet2CIDR

MapPublicIpOnLaunch: false

Tags:

- Key: Name

Value: !Sub \${EnvironmentName} Private Subnet (AZ2)

NatGateway1EIP:

Type: AWS::EC2::EIP

DependsOn: InternetGatewayAttachment

Properties:

Domain: vpc

NatGateway2EIP:

Type: AWS::EC2::EIP

DependsOn: InternetGatewayAttachment

Properties:

Domain: vpc

NatGateway1:

Type: AWS::EC2::NatGateway

Properties:

AllocationId: !GetAtt NatGateway1EIP.AllocationId

SubnetId: !Ref PublicSubnet1

NatGateway2:

Type: AWS::EC2::NatGateway

Properties:

AllocationId: !GetAtt NatGateway2EIP.AllocationId

SubnetId: !Ref PublicSubnet2

PublicRouteTable:

Type: AWS::EC2::RouteTable

Properties:

VpcId: !Ref VPC

Tags:

- Key: Name

Value: !Sub \${EnvironmentName} Public Routes

DefaultPublicRoute:

Type: AWS::EC2::Route

DependsOn: InternetGatewayAttachment

Properties:

RouteTableId: !Ref PublicRouteTable

DestinationCidrBlock: 0.0.0.0/0

GatewayId: !Ref InternetGateway

PublicSubnet1RouteTableAssociation:

Type: AWS::EC2::SubnetRouteTableAssociation

Properties:

RouteTableId: !Ref PublicRouteTable

SubnetId: !Ref PublicSubnet1

PublicSubnet2RouteTableAssociation:

Type: AWS::EC2::SubnetRouteTableAssociation

Properties:

RouteTableId: !Ref PublicRouteTable

SubnetId: !Ref PublicSubnet2

PrivateRouteTable1:

Type: AWS::EC2::RouteTable

Properties:

VpcId: !Ref VPC

Tags:

- Key: Name

- Value: !Sub \${EnvironmentName} Private Routes (AZ1)

DefaultPrivateRoute1:

Type: AWS::EC2::Route

Properties:

RouteTableId: !Ref PrivateRouteTable1

DestinationCidrBlock: 0.0.0.0/0

NatGatewayId: !Ref NatGateway1

PrivateSubnet1RouteTableAssociation:

Type: AWS::EC2::SubnetRouteTableAssociation

Properties:

RouteTableId: !Ref PrivateRouteTable1

SubnetId: !Ref PrivateSubnet1

PrivateRouteTable2:

Type: AWS::EC2::RouteTable

Properties:

VpcId: !Ref VPC

Tags:

- Key: Name

- Value: !Sub \${EnvironmentName} Private Routes (AZ2)

DefaultPrivateRoute2:

Type: AWS::EC2::Route

Properties:

RouteTableId: !Ref PrivateRouteTable2

DestinationCidrBlock: 0.0.0.0/0

NatGatewayId: !Ref NatGateway2

PrivateSubnet2RouteTableAssociation:

Type: AWS::EC2::SubnetRouteTableAssociation

Properties:

RouteTableId: !Ref PrivateRouteTable2

SubnetId: !Ref PrivateSubnet2

NoIngressSecurityGroup:

Type: AWS::EC2::SecurityGroup

Properties:

GroupName: "no-ingress-sg"

GroupDescription: "Security group with no ingress rule"

VpcId: !Ref VPC

Ec2instance1:

Type: AWS::EC2::Instance

Properties:

ImageId: ami-0629230e074c580f2

InstanceType: t2.micro

SubnetId: !Ref PublicSubnet1

KeyName: "Python"

SecurityGroupIds:

- !Ref BastionSecurityGroup

Tags:

- Key: "Name"

Value: "Bastion-Host"

BastionSecurityGroup:

Type: AWS::EC2::SecurityGroup

Properties:

GroupDescription: "Security group that allows SSH from anywhere"

GroupName: "Bastion"

SecurityGroupIngress:

- IpProtocol: tcp

FromPort: 22

ToPort: 22

CidrIp: 0.0.0.0/0

VpcId: !Ref VPC

Ec2instance2:

Type: AWS::EC2::Instance

Properties:

ImageId: ami-0629230e074c580f2

InstanceType: t2.micro

SubnetId: !Ref PublicSubnet1

KeyName: "Python"

SecurityGroupIds:

- !Ref BastionSecurityGroup

Tags:

- Key: "Name"

Value: "Nginx-Host"

NginxSecurityGroup:

Type: AWS::EC2::SecurityGroup

Properties:

GroupDescription: "Security group that allows SSH from bastion host only. It also allows client access on HTTP/HTTPS"

GroupName: "Nginx"

SecurityGroupIngress:

- IpProtocol: tcp

FromPort: 22

ToPort: 22

SourceSecurityGroupId:

Fn::GetAtt:

- BastionSecurityGroup

- GroupId

- IpProtocol: tcp

FromPort: 80

ToPort: 80

CidrIp: 0.0.0.0/0

- IpProtocol: tcp

FromPort: 443

ToPort: 443

CidrIp: 0.0.0.0/0

VpcId: !Ref VPC

Ec2instance3:

Type: AWS::EC2::Instance

Properties:

ImageId: ami-0629230e074c580f2

InstanceType: t2.micro

SubnetId: !Ref PublicSubnet1

KeyName: "Python"

SecurityGroupIds:

- !Ref BastionSecurityGroup

Tags:

- Key: "Name"

Value: "PhpMYAdmin"

PhpMyAdminSecurityGroup:

Type: AWS::EC2::SecurityGroup

Properties:

GroupDescription: "Security group that only allows SSH from the 1st instance (bastion host)

"

GroupName: "phpMyAdmin"

SecurityGroupIngress:

- IpProtocol: tcp

FromPort: 22

ToPort: 22
SourceSecurityGroupId:
Fn::GetAtt:
- BastionSecurityGroup
- GroupId
- IpProtocol: tcp
FromPort: 80
ToPort: 80
SourceSecurityGroupId:
Fn::GetAtt:
- NginxSecurityGroup
- GroupId
VpcId: !Ref VPC

Outputs:

VPC:

Description: A reference to the created VPC
Value: !Ref VPC

PublicSubnets:

Description: A list of the public subnets
Value: !Join [",", [!Ref PublicSubnet1, !Ref PublicSubnet2]]

PrivateSubnets:

Description: A list of the private subnets
Value: !Join [",", [!Ref PrivateSubnet1, !Ref PrivateSubnet2]]

PublicSubnet1:

Description: A reference to the public subnet in the 1st Availability Zone
Value: !Ref PublicSubnet1

PublicSubnet2:

Description: A reference to the public subnet in the 2nd Availability Zone
Value: !Ref PublicSubnet2

PrivateSubnet1:

Description: A reference to the private subnet in the 1st Availability Zone
Value: !Ref PrivateSubnet1

PrivateSubnet2:

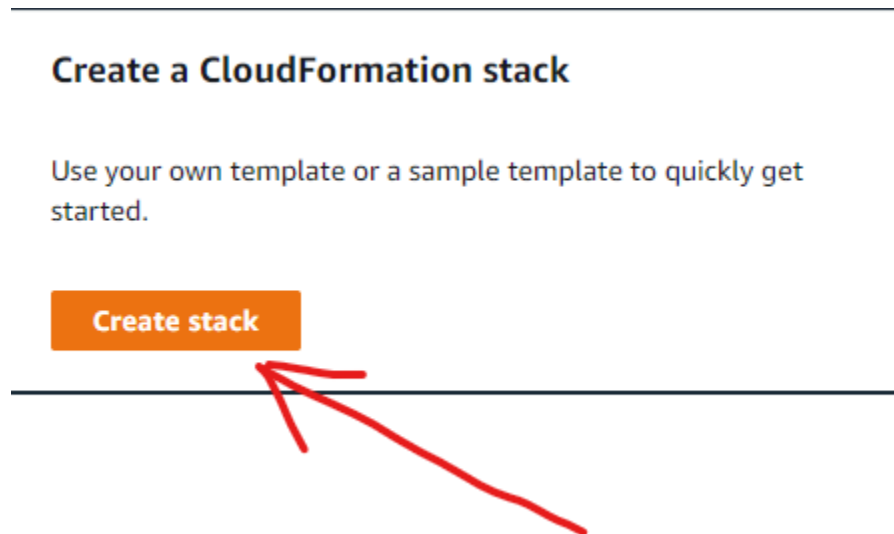
Description: A reference to the private subnet in the 2nd Availability Zone
Value: !Ref PrivateSubnet2

NoIngressSecurityGroup:

Description: Security group with no ingress rule

Value: !Ref NoIngressSecurityGroup

CreATE THE STACK IN CLOUD FORMATION



Create stack

Prerequisite - Prepare template


Prepare template
Every stack is based on a template. A template is a JSON or YAML file that contains configuration information about the AWS resources you want to include in the stack.

☒ Template is ready ☐ Use a sample template ☐ Create template in Designer

Specify template
A template is a JSON or YAML file that describes your stack's resources and properties.

Template source
Selecting a template generates an Amazon S3 URL where it will be stored.

☐ Amazon S3 URL ☒ Upload a template file


Upload a template file
Choose file  3-tier.yml
JSON or YAML formatted file

S3 URL: <https://s3.us-east-2.amazonaws.com/cf-templates-pwt6qx1dg629-us-east-2/2021343Zey-3-tier.yml> [View in Designer](#)

[Cancel](#) [Next](#)


Stack failure options

Behavior on provisioning failure

Specify the roll back behavior for a stack failure. [Learn more](#) 

- ☒ **Roll back all stack resources**
Roll back the stack to the last known stable state.
- ☐ **Preserve successfully provisioned resources**
Preserves the state of successfully provisioned resources, while rolling back failed resources to the last known stable state. Resources without a last known stable state will be deleted upon the next stack operation.


Advanced options

You can set additional options for your stack, like notification options and a stack policy. [Learn more](#) 

► Stack policy

Defines the resources that you want to protect from unintentional updates during a stack update.

► Rollback configuration

Specify alarms for CloudFormation to monitor when creating and updating the stack. If the operation breaches an alarm threshold, CloudFormation rolls it back. [Learn more](#) 

► Notification options

► Stack creation options

Cancel

Previous

Next



Specify stack details

Stack name

Stack name

Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

Parameters

Parameters are defined in your template and allow you to input custom values when you create or update a stack.

EnvironmentName

An environment name that is prefixed to resource names

PrivateSubnet1CIDR

Please enter the IP range (CIDR notation) for the private subnet in the first Availability Zone

PrivateSubnet2CIDR

Please enter the IP range (CIDR notation) for the private subnet in the second Availability Zone

PublicSubnet1CIDR

Please enter the IP range (CIDR notation) for the public subnet in the first Availability Zone

PublicSubnet2CIDR

Please enter the IP range (CIDR notation) for the public subnet in the second Availability Zone

VpcCIDR

Please enter the IP range (CIDR notation) for this VPC

Cancel

Previous

Next



Review→create stack

Rollback configuration

Monitoring time

-

CloudWatch alarm ARN

-

Notification options

No notification options

There are no notification options defined

Stack creation options

Timeout

-

Termination protection

Disabled

► Quick-create link

Cancel

Previous

Create change set

Create stack



3.21.43.17
18.222.110.36
192.168.180.20

```
ssh -i "Python.pem" ubuntu@3.21.43.17
Sudo nano linux.pem —>copy paste you key inside linux pem
sudo chmod 400 linux.pem
sudo ssh into Nginx and php using the new key
sudo ssh -i "linux.pem" ubuntu@192.168.162.62
sudo ssh -i "linux.pem" ubuntu@192.168.180.20
```

Task 2:

Create an AWS Application Load Balancer to connect to your reverse proxy.



Target Groups New

Basic configuration

Settings in this section cannot be changed after the target group is created.

Choose a target type

☒ **Instances**

- Supports load balancing to instances within a specific VPC.

☐ **IP addresses**

- Supports load balancing to VPC and on-premises resources.
- Facilitates routing to multiple IP addresses and network interfaces on the same instance.
- Offers flexibility with microservice based architectures, simplifying inter-application communication.

☐ **Lambda function**

- Facilitates routing to a single Lambda function.
- Accessible to Application Load Balancers only.

☐ **Application Load Balancer**

- Offers the flexibility for a Network Load Balancer to accept and route TCP requests within a specific VPC.
- Facilitates using static IP addresses and PrivateLink with an Application Load Balancer.

Target group name

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Protocol

Port

HTTP



:

80

VPC

Select the VPC with the instances that you want to include in the target group.

ENV-3tier

vpc-0656ae7cad5009785

IPv4: 192.168.0.0/16



Protocol version

☒ **HTTP1**

Send requests to targets using HTTP/1.1. Supported when the request protocol is HTTP/1.1 or HTTP/2.

☐ **HTTP2**

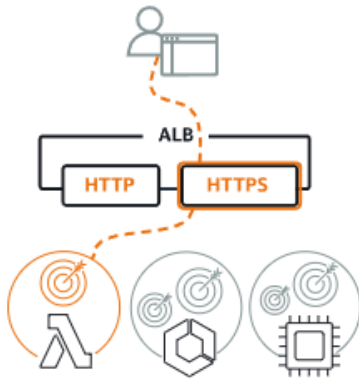
Send requests to targets using HTTP/2. Supported when the request protocol is HTTP/2 or gRPC, but gRPC-specific features are not available.

☐ **gRPC**

Send requests to targets using gRPC. Supported when the request protocol is gRPC.

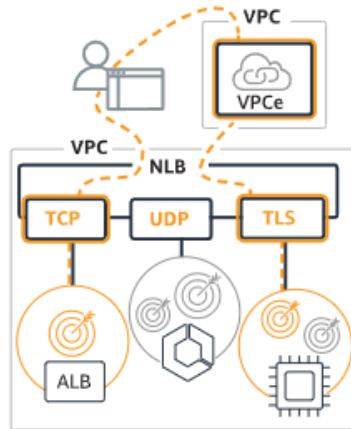
Load balancer types

Application Load Balancer [Info](#)



Choose an Application Load Balancer when you need a flexible feature set for your applications with HTTP and HTTPS traffic. Operating at the request level, Application Load Balancers provide advanced routing and visibility features targeted at application architectures, including microservices and containers.

Network Load Balancer [Info](#)



Choose a Network Load Balancer when you need ultra-high performance, TLS offloading at scale, centralized certificate deployment, support for UDP, and static IP addresses for your applications. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per second securely while maintaining ultra-low

Gateway Load Balancer [Info](#)



Choose a Gateway Load Balancer when you need to deploy and manage a fleet of third-party virtual appliances that support GENEVE. These appliances enable you to improve security, compliance, and policy controls.

Create

Basic configuration

Load balancer name

Name must be unique within your AWS account and cannot be changed after the load balancer is created.

Threetier

A maximum of 32 alphanumeric characters including hyphens are allowed, but the name must not begin or end with a hyphen.

Scheme [Info](#)

Scheme cannot be changed after the load balancer is created.

☒ Internet-facing

An internet-facing load balancer routes requests from clients over the internet to targets. Requires a public subnet. [Learn more](#)

☐ Internal

An internal load balancer routes requests from clients to targets using private IP addresses.

IP address type [Info](#)

Select the type of IP addresses that your subnets use.

☒ IPv4

Recommended for internal load balancers.

☐ Dualstack

Includes IPv4 and IPv6 addresses.

Network mapping [Info](#)

The load balancer routes traffic to targets in the selected subnets, and in accordance with your IP address settings.

VPC [Info](#)

Select the virtual private cloud (VPC) for your targets. Only VPCs with an internet gateway are enabled for selection. The selected VPC cannot be changed after the load balancer is created. To confirm the VPC for your targets, view your [target groups](#).

ENV-3tier

vpc-0656ae7cad5009785

IPv4: 192.168.0.0/16



Mappings [Info](#)

Select at least two Availability Zones and one subnet per zone. The load balancer routes traffic to targets in these Availability Zones only. Availability Zones that are not supported by the load balancer or the VPC are not available for selection. Subnets cannot be removed after the load balancer is created, but additional subnets can be added.

☒ us-east-2a

Subnet

subnet-08fba9639bdd91ef3

ENV-3tier Public Subnet (AZ1) ▼

IPv4 settings

Assigned by AWS

☒ us-east-2b

Subnet

subnet-03aef3f247887cc7d

ENV-3tier Public Subnet (AZ2) ▼

You just need a security group that take HTTP and HTTPS. So you could use Nginx` NGIUNIX

Security groups [Info](#)

A security group is a set of firewall rules that control the traffic to your load balancer.

Security groups

Select security groups



[Create new security group](#)

Nginx

sg-02e74a52e79588c20



VPC: vpc-0656ae7cad5009785

Listeners and routing [Info](#)

A listener is a process that checks for connection requests, using the protocol and port you configure. Traffic received by the listener is then routed per your specification. You can specify multiple rules and multiple certificates per listener after the load balancer is created.

▼ Listener HTTP:80

Remove

Protocol

HTTP

:

Port

80

1-65535

Default action [Info](#)

Forward to

Select a target group



[Create target group](#)

Add listener

Listeners and routing [Info](#)

A listener is a process that checks for connection requests, using the protocol and port you configure. Traffic received by the listener is then routed per your specification. You can specify multiple rules and multiple certificates per listener after the load balancer is created.

▼ Listener HTTP:80 Remove

Protocol

Port

Default action [Info](#)

HTTP ▼

:

80

Forward to

Threetier-TG

HTTP ▼

⌂

1-65535

Target type: Instance, IPv4


[Create target group](#) 🔗

Add listener

3.21.43.17
18.222.110.36
192.168.180.20

Task 3:

- Create a MySQL database on AWS by first creating a subnet group



Amazon Aurora

Amazon Aurora is a MySQL- and PostgreSQL-compatible enterprise-class database, starting at <\$1/day. Aurora supports up to 64TB of auto-scaling storage capacity, 6-way replication across three availability zones, and 15 low-latency read replicas. [Learn more](#)

Create database

Or, Restore Aurora DB cluster from S3

×

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](#)

☐ Amazon Aurora



☒ MySQL



☐ MariaDB



☐ PostgreSQL



☐ Oracle



☐ Microsoft SQL Server



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](#)

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.

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.

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

☐ 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 password [Info](#)

Keep default

DB instance class

DB instance class [Info](#)

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

1 vCPUs 1 GiB RAM Not EBS Optimized



☐ Include previous generation classes

Keep Default

Storage

Storage type [Info](#)

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

▼

Allocated storage

20

GiB

(Minimum: 20 GiB. Maximum: 16,384 GiB) Higher allocated storage [may improve](#) IOPS performance.

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 once the specified threshold is exceeded.

Maximum storage threshold [Info](#)

Charges will apply when your database autoscales to the specified threshold

1000

GiB

Minimum: 21 GiB. Maximum: 16,384 GiB

Pick

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

Subnet group [Info](#)

DB subnet group that defines which subnets and IP ranges the DB instance can use in the VPC you selected.

Create new DB Subnet Group ▼

Public access [Info](#)

☐ Yes

Amazon EC2 instances and devices outside the VPC can connect to your database. Choose one or more VPC security groups that specify which EC2 instances and devices inside the VPC can connect to the database.

☒ No

RDS will not assign a public IP address to the database. Only Amazon EC2 instances and devices inside the VPC can connect to your database.

VPC security group

Choose a VPC security group to allow access to your database. Ensure that the security group rules allow the appropriate incoming traffic.

☒ Choose existing

Choose existing VPC security groups

☐ Create new

Create new VPC security group

Existing VPC security groups

Choose VPC security groups ▼

Nginx X

Availability Zone [Info](#)

us-east-2a ▼

Pick your vpc – which is the three-tier

Virtual private cloud (VPC) [Info](#)

VPC that defines the virtual networking environment for this DB instance.

ThreeTierENV (vpc-06a85c8eed23024de) ▼

Only VPCs with a corresponding DB subnet group are listed.

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

Pick no public access and use nginx security group because you want to allow Http and https

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

Create new DB Subnet Group ▼

Public access [Info](#)

☐ Yes
Amazon EC2 instances and devices outside the VPC can connect to your database. Choose one or more VPC security groups that specify which EC2 instances and devices inside the VPC can connect to the database.

☒ No
RDS will not assign a public IP address to the database. Only Amazon EC2 instances and devices inside the VPC can connect to your database.

VPC security group
Choose a VPC security group to allow access to your database. Ensure that the security group rules allow the appropriate incoming traffic.

☒ Choose existing
Choose existing VPC security groups

☐ Create new
Create new VPC security group

Existing VPC security groups

Choose VPC security groups ▼

Nginx ✕

Availability Zone [Info](#)

us-east-2a ▼

► Additional configuration

Everything else default

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.

► Additional configuration

Database options, backup enabled, backtrack disabled, Enhanced Monitoring disabled, maintenance, CloudWatch Logs, delete protection disabled.

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 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](#).

Go below and pick create database, it will take you some time

► Additional configuration

Database options, encryption enabled, failover, backup enabled, backtrack disabled, Performance Insights enabled, Enhanced Monitoring enabled, maintenance, CloudWatch Logs, delete protection enabled.

-  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

Task 4:

SSH into your phpMyadmin. If you aren already connected go into bastion and then connect to phpmyadmin with the new pem key.

```
sudo apt-get update && sudo apt-get upgrade -y
```

Instal apache

```
sudo apt-get install apache2 -y
```

Install PHP and a module that will have php connect to my sql server

```
sudo apt install php libapache2-mod-php php-mysql -y
```

Now we have to make sure our web page is working so change directory to where apache host web pages

```
cd /var/www/html
```

Create a PHP file

```
sudo nano test.php
```

Paste this inside test.php

```
<?php phpinfo();
```

Install MySQL server

```
sudo apt install mysql-server -y
```

Run MYSQL installation

```
sudo mysql_secure_installation
```

Yes

1

Same passwords as the password for rds

Yes

<ENTERKEY>

<ENTERKEY>

<ENTERKEY>

<ENTERKEY>

Enter the following command inside the interactive shell

```
ubuntu@ip-192-168-180-20:/var/www/html$ show databases;  
  
Command 'show' not found, but can be installed with:  
  
sudo apt install mailutils-mh # version 1:3.7-2.1, or  
sudo apt install mmh          # version 0.4-2  
sudo apt install nmh          # version 1.7.1-6
```

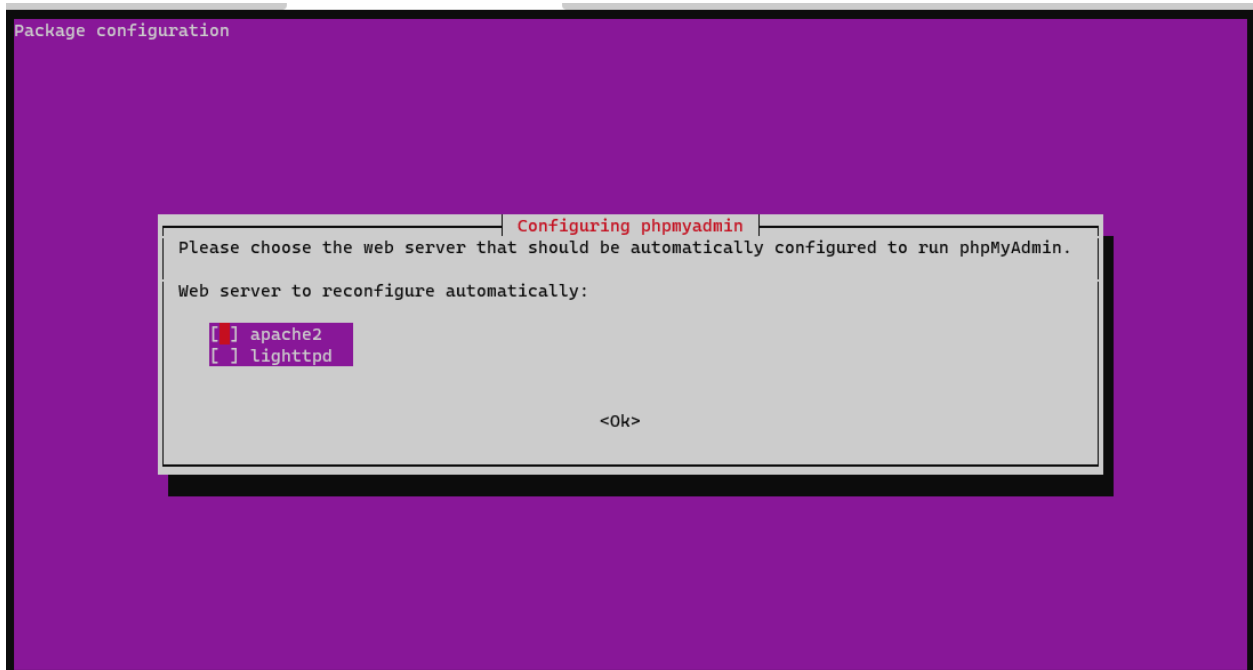
show databases;

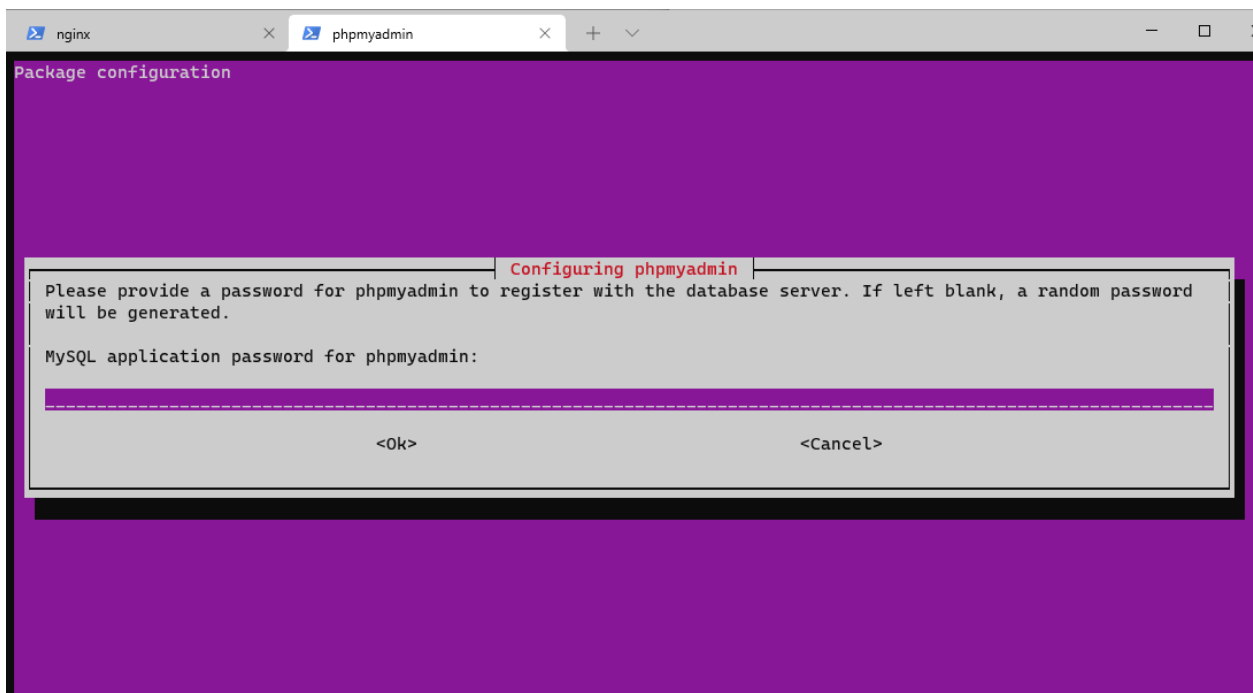
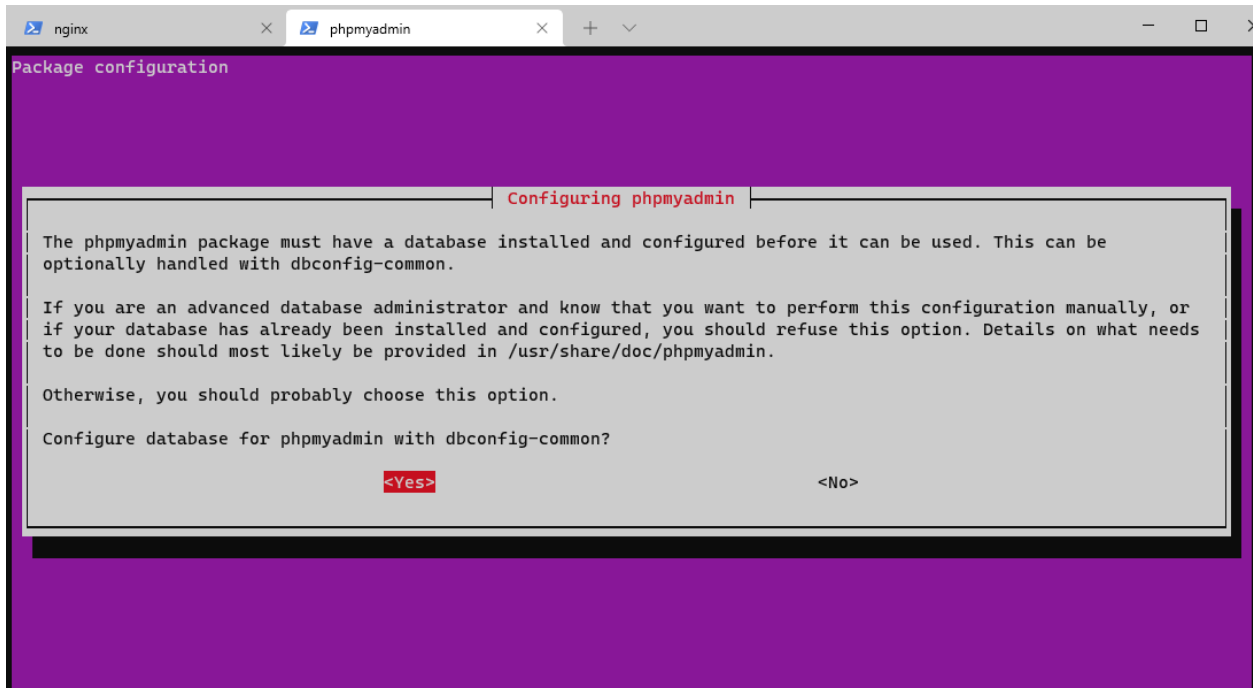
Logout using the command:exit

```
ubuntu@ip-192-168-180-20:/var/www/html$ exit  
logout  
There are stopped jobs.
```

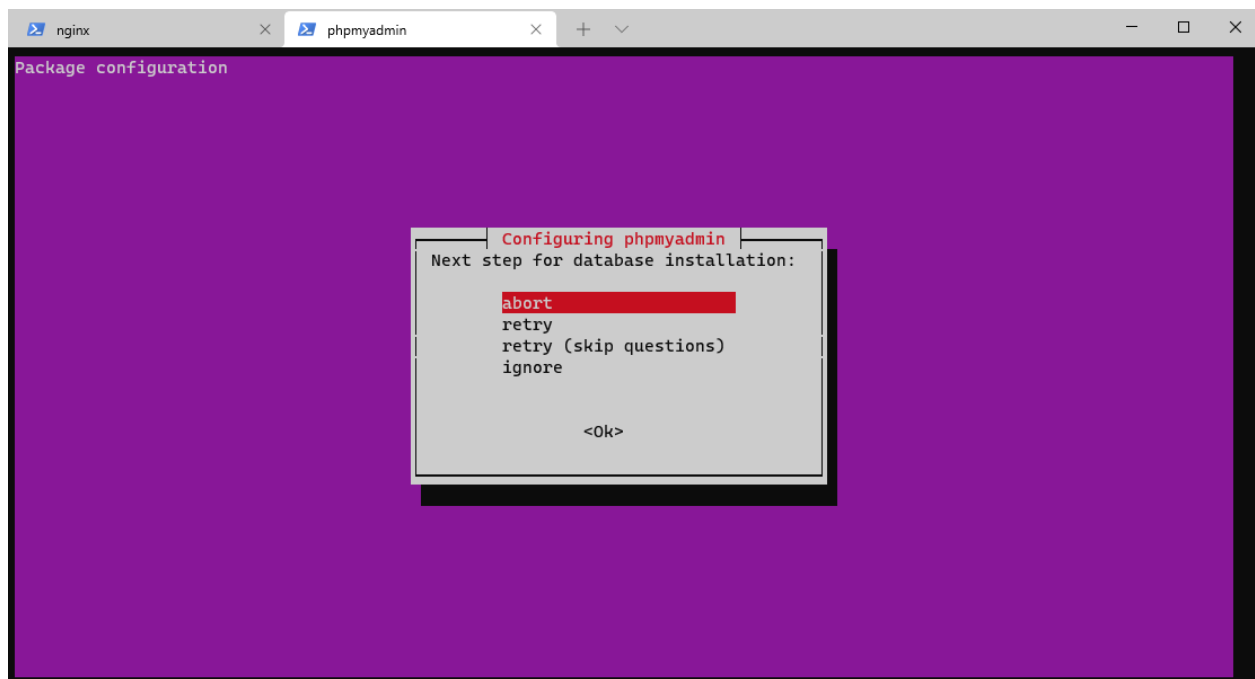
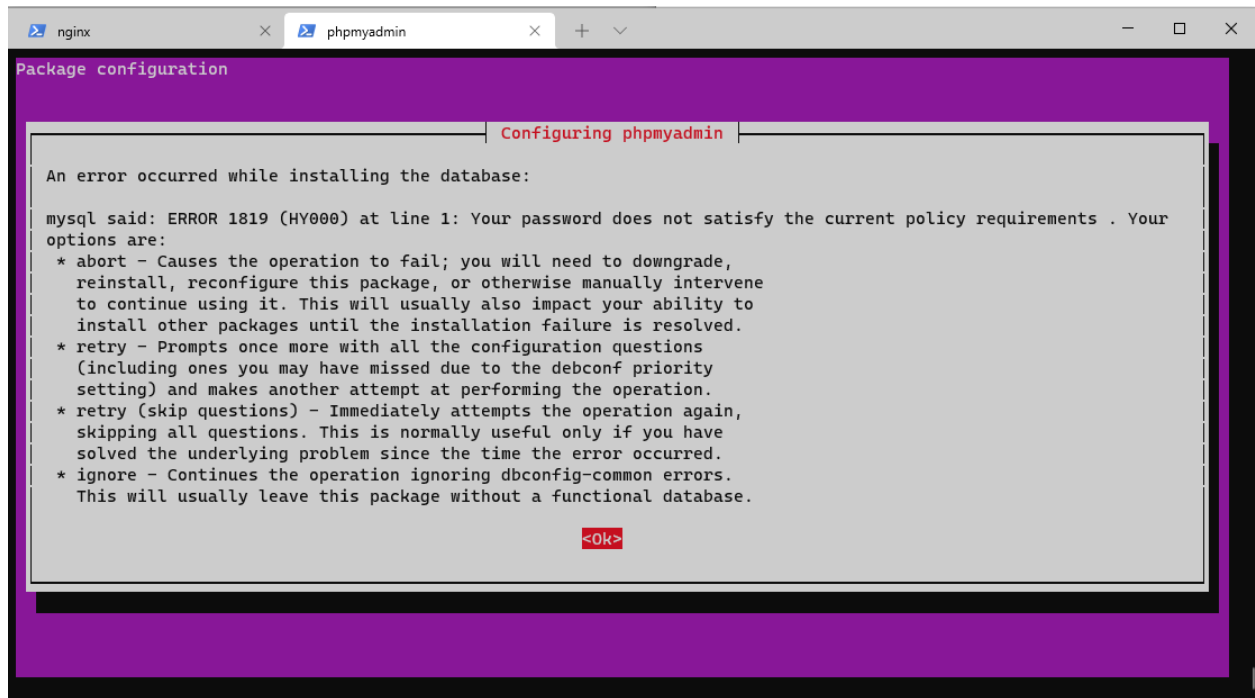
Download packages for phpmyadmin

sudo apt install phpmyadmin php-mbstring php-zip php-gd php-json php-curl -y





Password is same as rds
Reconfirm password



Log back into mysql
sudo mysql

Paste the following inside the MySQL interactive shell
SELECT user,authentication_string,plugin,host FROM mysql.user;

Root empty

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

mysql> SELECT user,authentication_string,plugin,host FROM mysql.user;
+-----+-----+-----+-----+
| user          | authentication_string | plugin          | host |
+-----+-----+-----+-----+
| debian-sys-maint | $A$005$&$?0FDtj0:L0zS60F00 .sInWxYxz3ZKmrU7rFqxAhPI9u/hVdCthpmrR9vH8oiA | caching_sha2_password | localhost |
| mysql.infoschema | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED | caching_sha2_password | localhost |
| mysql.session   | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED | caching_sha2_password | localhost |
| mysql.sys       | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED | caching_sha2_password | localhost |
| root           |                        | auth_socket     | localhost |
+-----+-----+-----+-----+
5 rows in set (0.01 sec)

mysql>
```

Run the following command inside the MySQL interactive shell
UNINSTALL COMPONENT "file://component_validate_password";

```
mysql> UNINSTALL COMPONENT "file://component_validate_password";
Query OK, 0 rows affected (0.00 sec)
```

We can then run the next command inside the MySQL interactive shell
INSTALL COMPONENT "file://component_validate_password";

Exit mysql
Exit

Install the following packages

sudo phpenmod mbstring

Go back into the MySQL

sudo mysql

We can use the following command which will use a hashing algorithm to encrypt our password and store it into the root localhost field.

```
ALTER USER 'root'@'localhost' IDENTIFIED WITH caching_sha2_password BY 'Bkl.com1384';
```

We can check if the changes were made using the following command

```
SELECT user,authentication_string,plugin,host FROM mysql.user;
```

```
nginx x phpmyadmin x + - x
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> ALTER USER 'root'@'localhost' IDENTIFIED WITH caching_sha2_password BY [REDACTED];
Query OK, 0 rows affected (0.01 sec)

mysql> SELECT user,authentication_string,plugin,host FROM mysql.user;
+-----+-----+-----+-----+
| user          | authentication_string                                     | plugin          | host          |
+-----+-----+-----+-----+
| debian-sys-maint | $A$005$&?0FDTjD:LDzS6DFDD .sInWxYxz3ZKmrU7rFqxAhPI9u/hVdCthpmrR9vH8oiA | caching_sha2_password | localhost |
| mysql.infoschema | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED | caching_sha2_password | localhost |
| mysql.session    | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED | caching_sha2_password | localhost |
| mysql.sys        | $A$005$THISISACOMBINATIONOFINVALIDSALTANDPASSWORDTHATMUSTNEVERBRBEUSED | caching_sha2_password | localhost |
| root            | $A$005$y,Pg#AG00000}                                     | caching_sha2_password | localhost |
| 'WXt#5T49G5wtDKI7iZwL0KvS2FdpXGNUrNBCa7iKrGgNvvX9' | caching_sha2_password | localhost |
+-----+-----+-----+-----+
5 rows in set (0.01 sec)
```

Exit the MySQL interactive shell
Exit

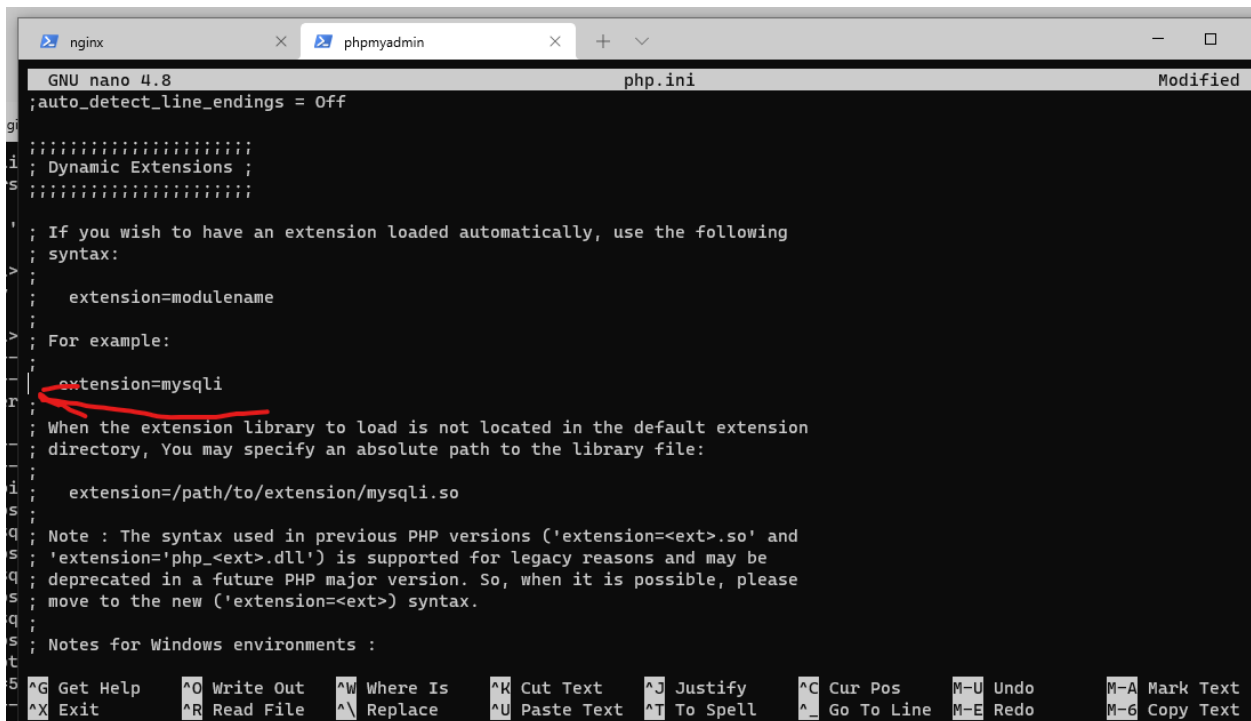
Change directory to the following

```
cd /etc/php/7.4/apache2/
```

We will need to edit a file...

```
sudo nano php.ini
```

Inside Nano select ALT + G. This will allow us to go to a line. Go to line 895 and remove the semicolon ;



```
GNU nano 4.8 php.ini Modified
;auto_detect_line_endings = Off

;
; Dynamic Extensions ;
;
; If you wish to have an extension loaded automatically, use the following
; syntax:
;
; extension=module_name
;
; For example:
;
; extension=mysqli
;
; When the extension library to load is not located in the default extension
; directory, You may specify an absolute path to the library file:
;
; extension=/path/to/extension/mysqli.so
;
; Note : The syntax used in previous PHP versions ('extension=<ext>.so' and
; 'extension='php_<ext>.dll') is supported for legacy reasons and may be
; deprecated in a future PHP major version. So, when it is possible, please
; move to the new ('extension=<ext>') syntax.
;
; Notes for Windows environments :
;
^G Get Help  ^O Write Out  ^W Where Is  ^K Cut Text   ^J Justify    ^C Cur Pos   M-U Undo     M-A Mark Text
^X Exit      ^R Read File  ^\ Replace   ^U Paste Text ^T To Spell   ^_ Go To Line M-E Redo     M-6 Copy Text
```

Save the file and exit it

```
CTRL + O
```

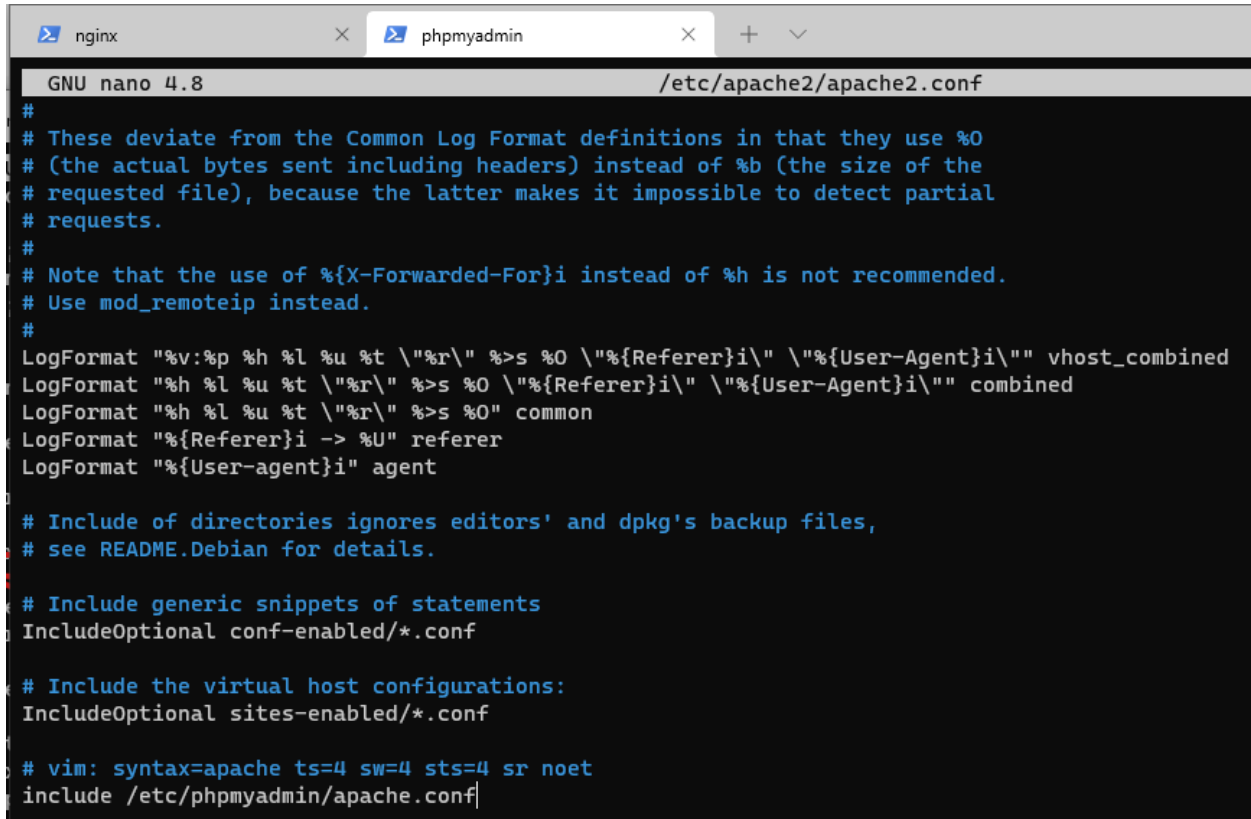
```
CTRL + X
```

We will edit the apache2 config file

```
sudo nano /etc/apache2/apache2.conf
```

Scroll all the way to the bottom and paste the following

```
include /etc/phpmyadmin/apache.conf
```



```
GNU nano 4.8 /etc/apache2/apache2.conf
#
# These deviate from the Common Log Format definitions in that they use %O
# (the actual bytes sent including headers) instead of %b (the size of the
# requested file), because the latter makes it impossible to detect partial
# requests.
#
# Note that the use of %{X-Forwarded-For}i instead of %h is not recommended.
# Use mod_remoteip instead.
#
LogFormat "%v:%p %h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\"" vhost_combined
LogFormat "%h %l %u %t \"%r\" %>s %O \"%{Referer}i\" \"%{User-Agent}i\"" combined
LogFormat "%h %l %u %t \"%r\" %>s %O" common
LogFormat "%{Referer}i -> %U" referer
LogFormat "%{User-agent}i" agent

# Include of directories ignores editors' and dpkg's backup files,
# see README.Debian for details.

# Include generic snippets of statements
IncludeOptional conf-enabled/*.conf

# Include the virtual host configurations:
IncludeOptional sites-enabled/*.conf

# vim: syntax=apache ts=4 sw=4 sts=4 sr noet
include /etc/phpmyadmin/apache.conf
```

Save the file and exit it

CTRL + O

CTRL + X

We will need to restart apache

sudo systemctl restart apache2

Edit the config file

sudo nano /etc/phpmyadmin/config.inc.php

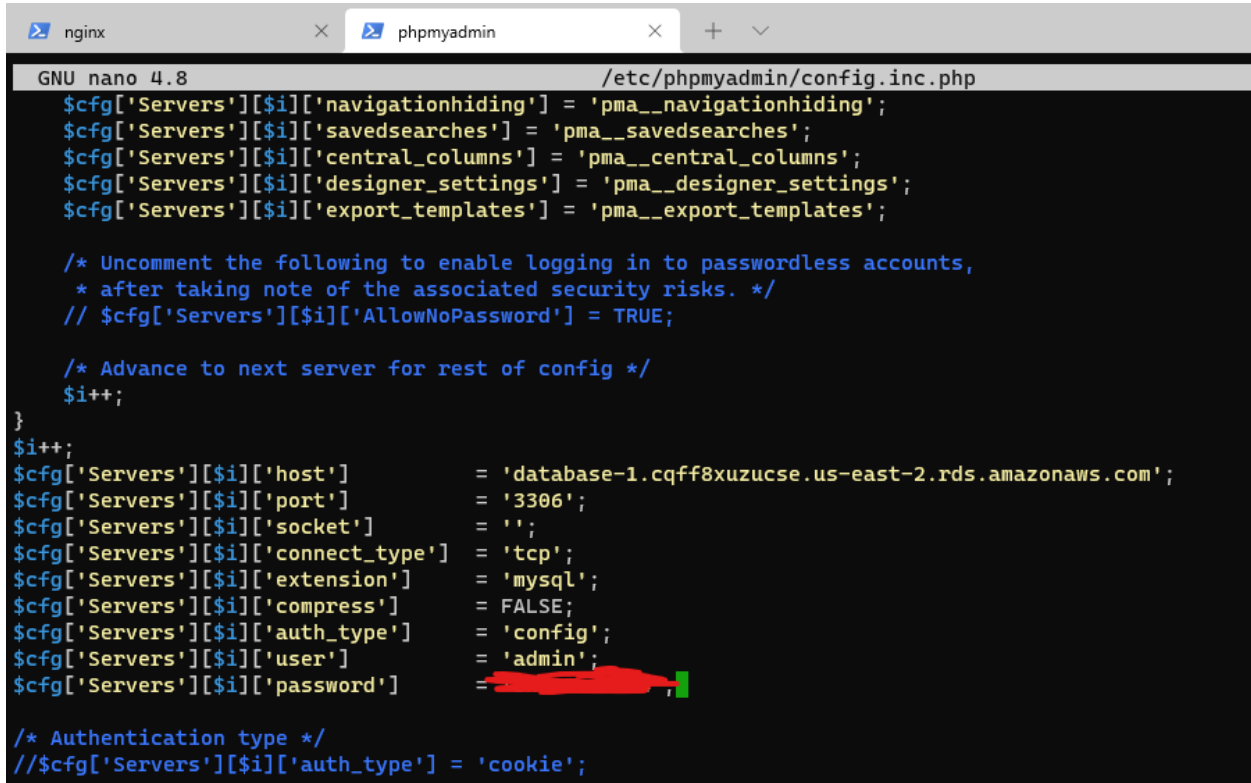
Inside nano select ALT + G. This will allow us to go to a line. Go to line 102 and paste the following below

```
$cfg['Servers'][$i]['host']      = '__FILL_IN_DETAILS__';
$cfg['Servers'][$i]['port']     = '3306';
$cfg['Servers'][$i]['socket']   = '';
$cfg['Servers'][$i]['connect_type'] = 'tcp';
```

```

$cfg['Servers'][$i]['extension']    = 'mysql';
$cfg['Servers'][$i]['compress']    = FALSE;
$cfg['Servers'][$i]['auth_type']   = 'config';
$cfg['Servers'][$i]['user']        = '__FILL_IN_DETAILS__';
$cfg['Servers'][$i]['password']    = '__FILL_IN_DETAILS__';

```



```

GNU nano 4.8 /etc/phpmyadmin/config.inc.php
$cfg['Servers'][$i]['navigationhiding'] = 'pma__navigationhiding';
$cfg['Servers'][$i]['savedsearches'] = 'pma__savedsearches';
$cfg['Servers'][$i]['central_columns'] = 'pma__central_columns';
$cfg['Servers'][$i]['designer_settings'] = 'pma__designer_settings';
$cfg['Servers'][$i]['export_templates'] = 'pma__export_templates';

/* Uncomment the following to enable logging in to passwordless accounts,
 * after taking note of the associated security risks. */
// $cfg['Servers'][$i]['AllowNoPassword'] = TRUE;

/* Advance to next server for rest of config */
$i++;
}
$i++;
$cfg['Servers'][$i]['host']            = 'database-1.cqff8xuzucse.us-east-2.rds.amazonaws.com';
$cfg['Servers'][$i]['port']            = '3306';
$cfg['Servers'][$i]['socket']          = '';
$cfg['Servers'][$i]['connect_type']    = 'tcp';
$cfg['Servers'][$i]['extension']       = 'mysql';
$cfg['Servers'][$i]['compress']        = FALSE;
$cfg['Servers'][$i]['auth_type']       = 'config';
$cfg['Servers'][$i]['user']            = 'admin';
$cfg['Servers'][$i]['password']        = 'REDACTED';

/* Authentication type */
// $cfg['Servers'][$i]['auth_type'] = 'cookie';

```

Crtl o vs certl x

We can now test our connection to the new database. We should be able to log into your AWS RDS

curl localhost:80/phpmyadmin/

Restart nginx

sudo systemctl restart apache2

We can now configure the NGINX Proxy. Exit the phpMyAdmin EC2

exit

Connect to your NGINX EC2 instance

ssh into the bastion EC2 -> ssh into the NGINX EC2

Run the following command

```
sudo apt-get update && sudo apt-get upgrade -y
```

Install NGINX

```
sudo apt-get install nginx -y
```

Change directories to Sites available

sites-available are conf files that tell NGINX where to look for.

```
cd /etc/nginx/sites-available/
```

We need to unlink the default sites-enabled file

```
sudo unlink /etc/nginx/sites-enabled/default  
sudo unlink /etc/nginx/sites-enabled/reverse-proxy.conf
```

Unlinking the reverse-proxy.conf will say there is no file. We need to create a configuration file for the reverse proxy

```
sudo nano reverse-proxy.conf
```

Paste the following inside the reverse-proxy configuration file (The proxy_pass IP is the phpMyAdmin private IPv4)

```
server {  
    listen 80;  
    location / {  
        proxy_pass http://192.168.180.20/;  
    }  
}
```

Check if the following directory is empty

```
ls /etc/nginx/sites-enabled/
```

We will link reverse-proxy to sites enabled so that apache can read it and use it. (ONE LINE COMMAND)

```
sudo ln -s /etc/nginx/sites-available/reverse-proxy.conf  
/etc/nginx/sites-enabled/reverse-proxy.conf
```

We will need to restart NGINX

```
sudo systemctl restart nginx
```

Now go to you load balancer

Create Load Balancer

Actions

search: Threetier

Add filter

Name	DNS name	State	VPC ID	Availability Zones	Type	Created At	Monitoring
Threetier	Threetier-1255670199 us-east-2.elb.amazonaws.com	Active	vpc-06a85c8eed23024de	us-east-2b, us-east-2a	application	December 9, 2021 at 7:48:0...	

Load balancer: Threetier

Description

Listeners

Monitoring

Integrated services

Tags

Basic Configuration

Name

Threetier

ARN

arn:aws:elasticloadbalancing:us-east-2:323867645900:loadbalancer/app/Threetier/889cbb9ad2758488

DNS name

Threetier-1255670199.us-east-2.elb.amazonaws.com

State

Active

Apache2 Ubuntu Default Page: It works!

Not secure | three-tier-847760348.us-east-2.elb.amazonaws.com

Gmail Microsoft Office Ho... Python Online Com... My Profile - Zoom NYC Instances | EC2 Ma... Maps kura-labs-org syllabus - Planner Mail - Bishajit Lodh... Github hom

Apache2 Ubuntu Default Page

ubuntu

It works!

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers `a2enmod`, `a2dismod`, `a2ensite`, `a2dissite`, and `a2enconf`, `a2disconf`. See their respective man pages for detailed information.
- The binary is called `apache2`. Due to the use of environment variables, in the default configuration, `apache2` needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`.

phpMyAdmin x Apache2 Ubuntu Default Page: It works!

Not secure | three-tier-847760348.us-east-2.elb.amazonaws.com/phpmyadmin/

Gmail Microsoft Office Ho... Python Online Com... My Profile - Zoom NYC Instances | EC2 Ma... Maps kura-labs-org syllabus - Planner Mail

phpMyAdmin

Welcome to phpMyAdmin

Language

English

Log in

Username: admin

Password:

Server Choice: database-1.cqff8xuzucse.1

Go

