Task 1

Steps:

1. Using CloudFormation, create a VPC with a public and a private subnet. Create a YAML file called base.yaml and paste the following:

```
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

  KeyName:
    Description: Name of an existing EC2 KeyPair to enable SSH access to the instance
    Type: AWS::EC2::KeyPair::KeyName

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:
```

```yaml
    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:
```

```yaml
 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

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

BastionEC2Instance:
  Type: AWS::EC2::Instance
  Properties:
    ImageId: ami-09e67e426f25ce0d7
    InstanceType: t2.micro
    SubnetId: !Ref PublicSubnet1
    KeyName: !Ref KeyName
    SecurityGroupIds:
      - !Ref BastionSecurityGroup
    Tags:
      - Key: "Name"
        Value: "Bastion"


NginxSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: "Security group that allows SSH from bastion host only and 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

NginxEC2Instance:
  Type: AWS::EC2::Instance
  Properties:
    ImageId: ami-09e67e426f25ce0d7
    InstanceType: t2.micro
    SubnetId: !Ref PrivateSubnet1
    KeyName: !Ref KeyName
    SecurityGroupIds:
      - !Ref NginxSecurityGroup
    Tags:
      - Key: "Name"
        Value: "Nginx"

phpMyAdminSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: "Security group that allows SSH from the bastion host only"
    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
```

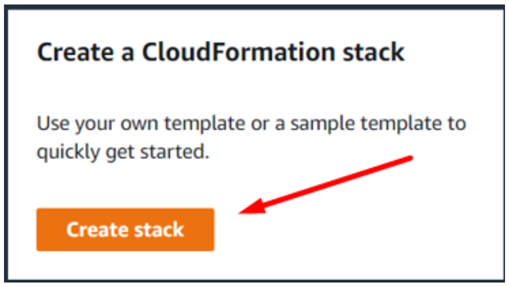
```
phpMyAdminEC2Instance:
  Type: AWS::EC2::Instance
  Properties:
    ImageId: ami-09e67e426f25ce0d7
    InstanceType: t2.micro
    SubnetId: !Ref PrivateSubnet1
    KeyName: !Ref KeyName
    SecurityGroupIds:
      - !Ref phpMyAdminSecurityGroup
    Tags:
      - Key: "Name"
        Value: "phpMyAdmin"

ThreeTierSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: "Security group that allows client access on HTTP/HTTPS for the Load Balancer"
    GroupName: "ThreeTier"
    SecurityGroupIngress:
      - 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

ThreeTierDBSecurityGroup:
  Type: AWS::EC2::SecurityGroup
  Properties:
    GroupDescription: "Security group for the RDS MySQL database that allows access from phpMyAdmin SG only"
    GroupName: "ThreeTierDB"
    SecurityGroupIngress:
      - IpProtocol: tcp
        FromPort: 3306
        ToPort: 3306
        SourceSecurityGroupId:
          Fn::GetAtt:
          - phpMyAdminSecurityGroup
          - 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

BastionSecurityGroup:
  Description: Security group with SSH from anywhere ingress rule
  Value: !Ref BastionSecurityGroup

NginxSecurityGroup:
  Description: Security group with SSH from anywhere ingress rule
  Value: !Ref NginxSecurityGroup

NginxSecurityGroup:
  Description: Security group that allows SSH from bastion host only and allows client access on HTTP/HTTPS
  Value: !Ref NginxSecurityGroup

phpMyAdminSecurityGroup:
  Description: Security group with SSH from only the bastion SG ingress rule
  Value: !Ref phpMyAdminSecurityGroup

ThreeTierSecurityGroup:
  Description: Security group that allows client access on HTTP/HTTPS for the Load Balancer
  Value: !Ref ThreeTierSecurityGroup

ThreeTierDBSecurityGroup:
```

Description: Security group for the RDS MySQL database that allows access from phpMyAdmin SG only

Value: !Ref ThreeTierDBSecurityGroup

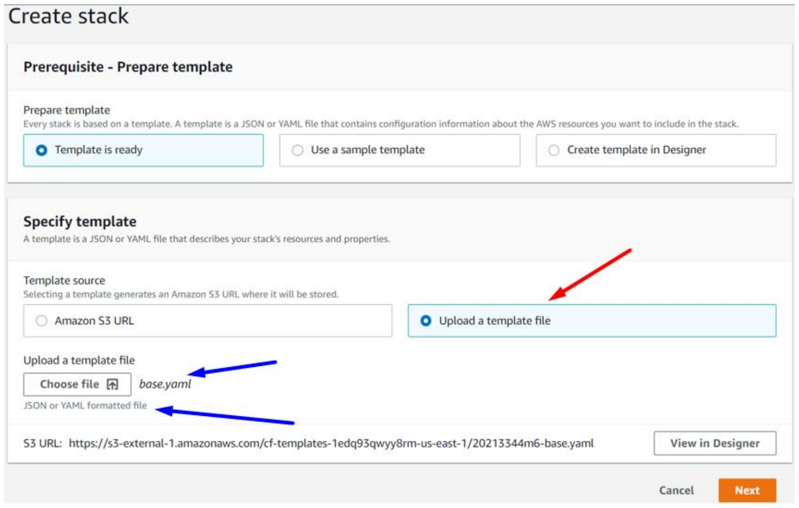
2. Once the file is created, save it and got to AWS CloudFormation to create a stack.



Create a CloudFormation stack

Use your own template or a sample template to quickly get started.

3. Create your stack and add the yaml file as shown below:



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.

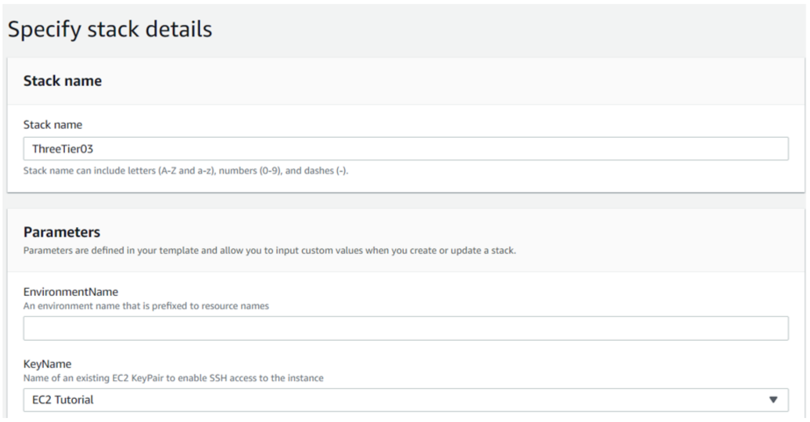
Template source
Selecting a template generates an Amazon S3 URL where it will be stored.

Amazon S3 URL [✓ Upload a template file]

Upload a template file
Choose file base.yaml
JSON or YAML formatted file

S3 URL: https://s3-external-1.amazonaws.com/cf-templates-1edq93qwyy8rm-us-east-1/20213344m6-base.yaml

4. Give the stack a name and include the Key Pair you'll use to SSH into the ec2 instances.



Specify stack details

Stack name

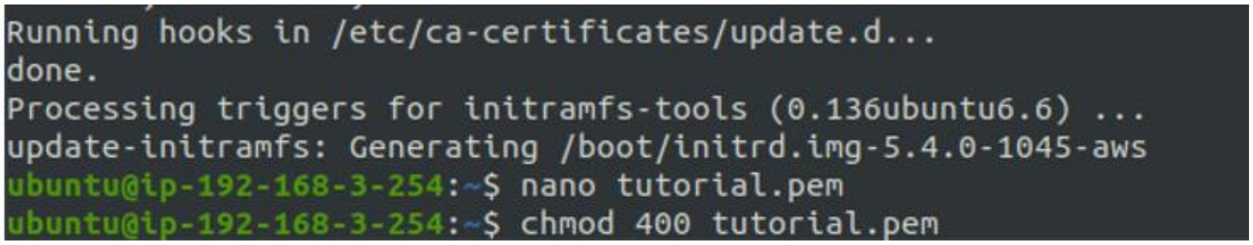
Stack name: ThreeTier03
Stack name can include letters (A-Z and a-z), numbers (0-9), and dashes (-).

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

KeyName
Name of an existing EC2 KeyPair to enable SSH access to the instance
EC2 Tutorial

5. Keep the rest of the default settings and create the stack. Once done SSH into the Bastion EC2 instance to make sure everything works correctly. Use the command ($ ssh -i key.pem ubuntu@publicIPv4). **NOTE:** use the **publicIPv4** of the Bastion EC2.
6. Once inside your instance use the command ($ sudo apt-get update && sudo apt-get upgrade -y).
7. Once the updating is complete use the command ($ nano tutorial.pem). Then copy and paste your private key pair so that you can SSH into the other instances. **NOTE:** the key.pem here can have any name.
8. After pasting the key value, save it and use the command ($ chmod 400 tutorial.pem) to change the permissions of the file.

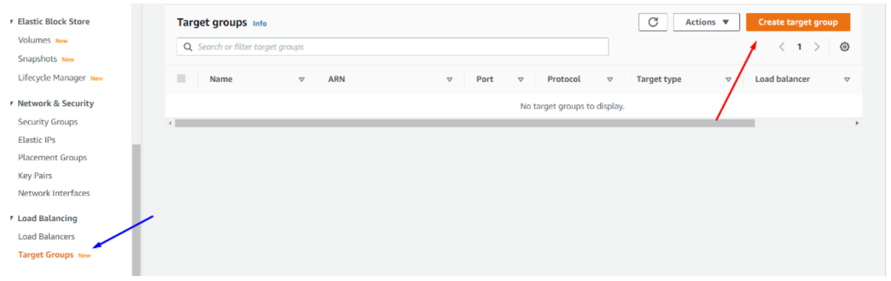


```
Running hooks in /etc/ca-certificates/update.d...
done.
Processing triggers for initramfs-tools (0.136ubuntu6.6) ...
update-initramfs: Generating /boot/initrd.img-5.4.0-1045-aws
ubuntu@ip-192-168-3-254:~$ nano tutorial.pem
ubuntu@ip-192-168-3-254:~$ chmod 400 tutorial.pem
```

9. Now SSH into the NGINX EC2 instance to make sure everything works. Use the command ($ ssh -i tutorial.pem ubuntu@privateIPv4). Once inside your instance use the command ($ sudo apt-get update && sudo apt-get upgrade -y). Once done type “exit”. **NOTE:** use the **privateIPv4** of the NGINX EC2.
10. Now SSH into the phpMyAdmin EC2 instance to make sure everything works. Use the command ($ ssh -i tutorial.pem ubuntu@privateIPv4). Once inside your instance use the command ($ sudo apt-get update && sudo apt-get upgrade -y). **NOTE:** use the **privateIPv4** of the phpMyAdmin EC2.

Task 2

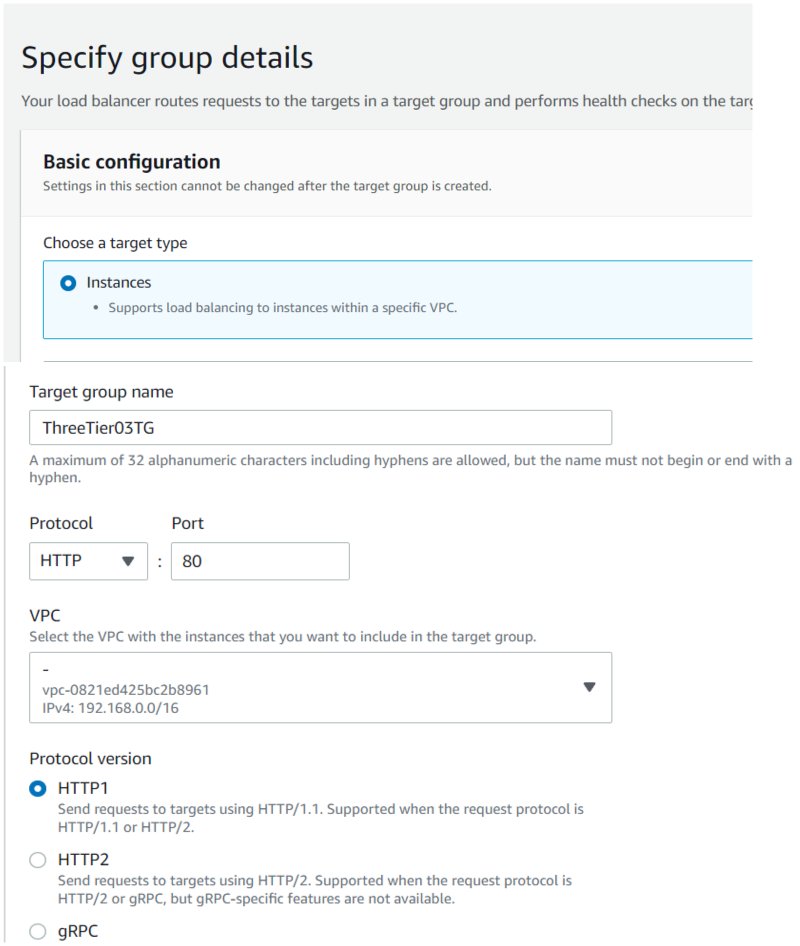
11. Now create an AWS Application Load Balancer (ALB) that will connect to the reverse proxy by first creating a Target Group



Name	ARN	Port	Protocol	Target type	Load balancer

No target groups to display.

12. Then choose the target type “instances” and then give it name and select the HTTP protocol. Also ensure that you choose the correct **VPC** (begins with 192).



Specify group details

Your load balancer routes requests to the targets in a target group and performs health checks on the targ

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.

Target group name

ThreeTier03TG

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

Protocol: HTTP
Port: 80

VPC

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

-
vpc-0821ed425bc2b8961
IPv4: 192.168.0.0/16

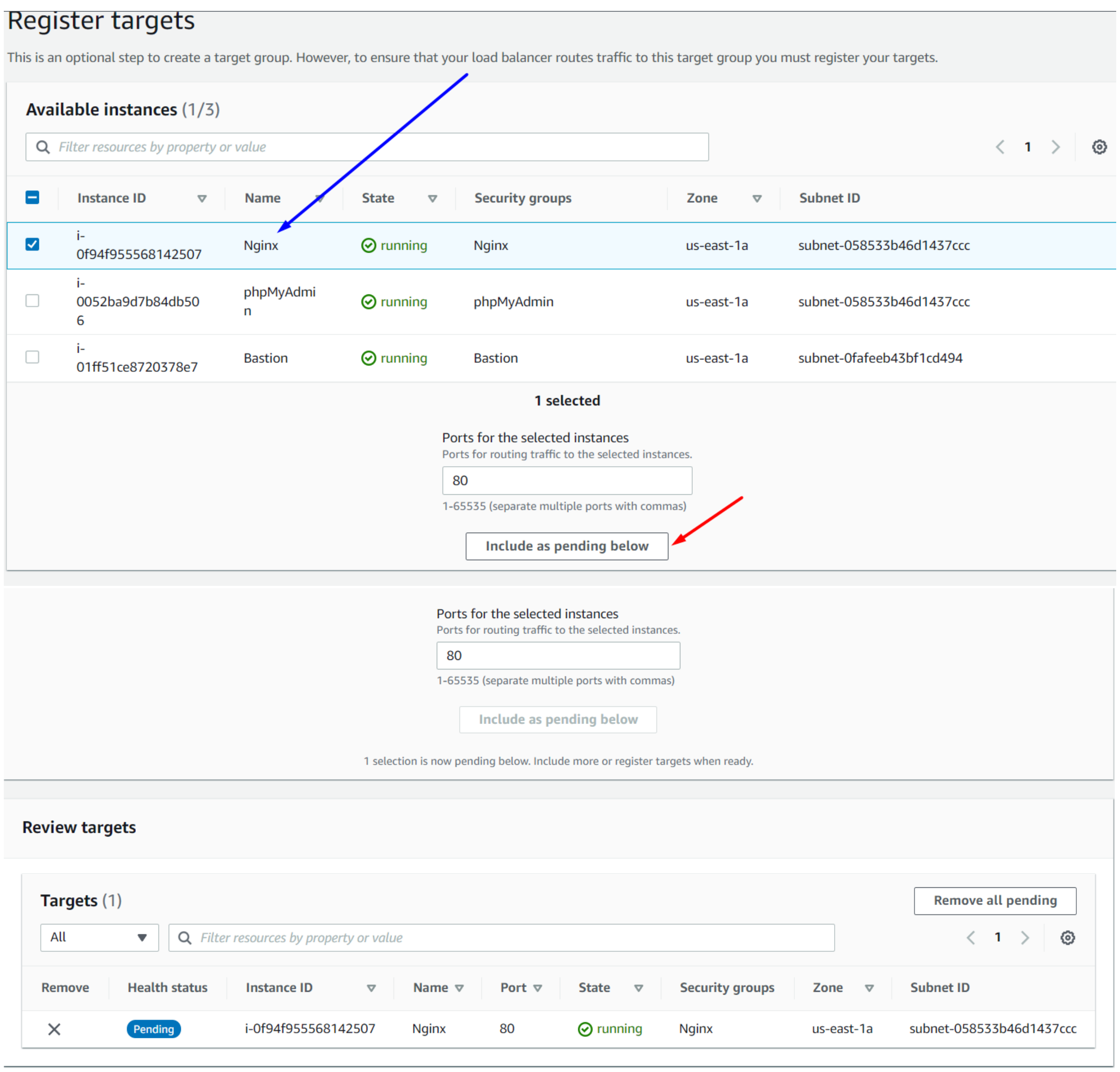
Protocol version

[✓ 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

13. After keep all the default settings and go to the next step. Then choose the EC2 instance we want to target. That will be **NGINX EC2** and will be our reverse proxy. After choosing it click “include as pending below”. After create the target group.



Register targets

This is an optional step to create a target group. However, to ensure that your load balancer routes traffic to this target group you must register your targets.

Available instances (1/3)

Filter resources by property or value

	Instance ID	Name	State	Security groups	Zone	Subnet ID
✓	i-0f94f955568142507	Nginx	running	Nginx	us-east-1a	subnet-058533b46d1437ccc
	i-0052ba9d7b84db506	phpMyAdmin	running	phpMyAdmin	us-east-1a	subnet-058533b46d1437ccc
	i-01ff51ce8720378e7	Bastion	running	Bastion	us-east-1a	subnet-0fafeeb43bf1cd494

1 selected

Ports for the selected instances
Ports for routing traffic to the selected instances.
80
1-65535 (separate multiple ports with commas)

Include as pending below

Ports for the selected instances
Ports for routing traffic to the selected instances.
80
1-65535 (separate multiple ports with commas)

Include as pending below

1 selection is now pending below. Include more or register targets when ready.

Review targets

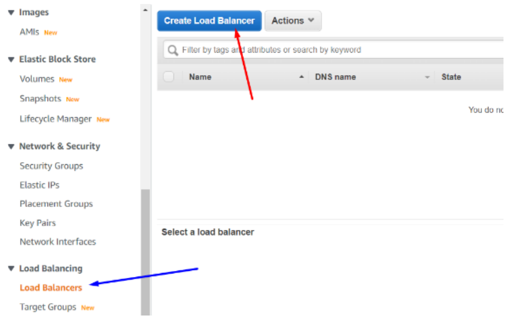
Targets (1)

Remove all pending

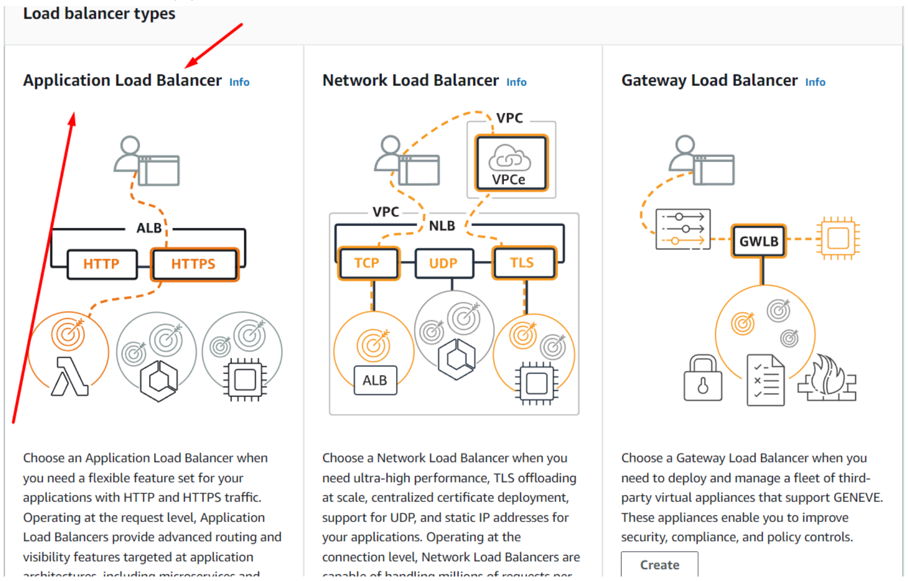
All — Filter resources by property or value

Remove	Health status	Instance ID	Name	Port	State	Security groups	Zone	Subnet ID
✕	Pending	i-0f94f955568142507	Nginx	80	running	Nginx	us-east-1a	subnet-058533b46d1437ccc

14. Now create and configure the load balancer.



15. After select Application Load Balancer


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

Network Load Balancer Info

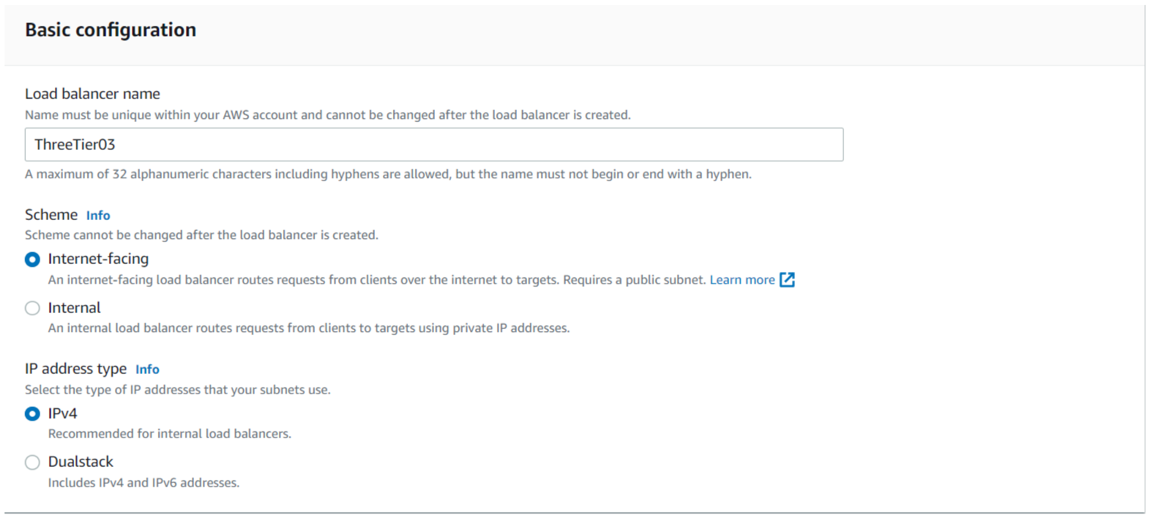
Choose a Network Load Balancer when you need ultra-high performance, TLS offloading at scale, centralized certificate deployment, support for UDP, and static IP addresses for your applications. Operating at the connection level, Network Load Balancers are capable of handling millions of requests per

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

16. Then configure it making sure it's given a suitable **name**


Basic configuration

Load balancer name

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

ThreeTier03

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.

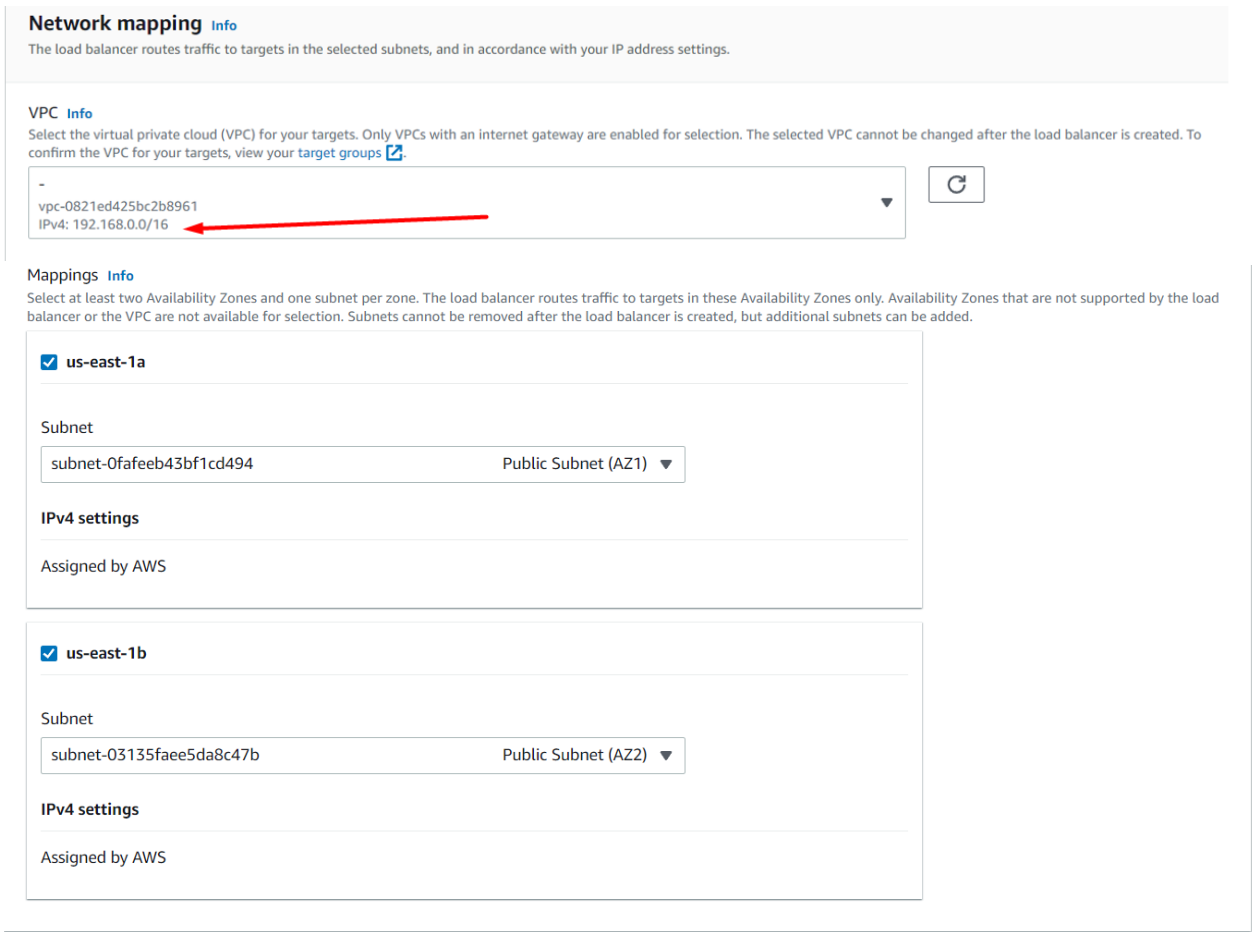
IP address type Info

Select the type of IP addresses that your subnets use.

[✓ IPv4]
Recommended for internal load balancers.

Dualstack
Includes IPv4 and IPv6 addresses.

17. Then set up the **Network mapping** for the **VPC** (using the IPv$ of 192) and choosing the **public subnets** for the mapping.



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.

-
vpc-0821ed425bc2b8961
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-1a

Subnet: subnet-0fafeeb43bf1cd494 Public Subnet (AZ1)

IPv4 settings

Assigned by AWS

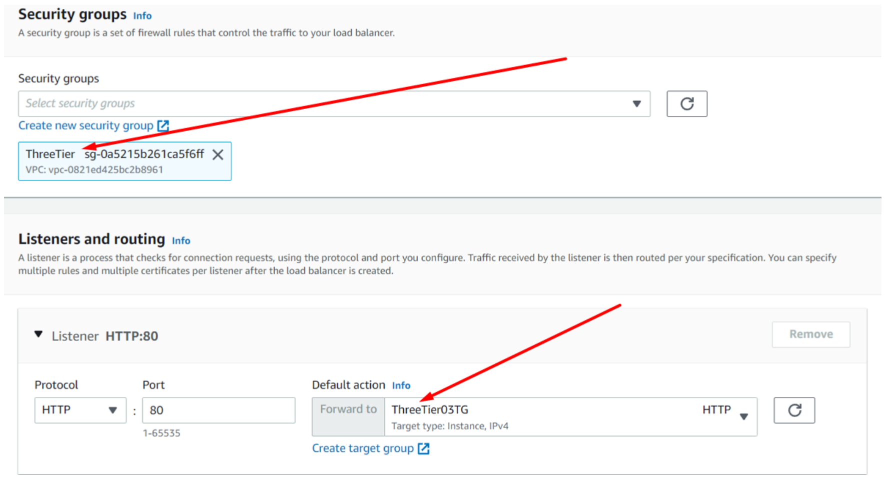
✓ us-east-1b

Subnet: subnet-03135faee5da8c47b Public Subnet (AZ2)

IPv4 settings

Assigned by AWS

18. Then use the **security group** created during CloudFormation that allows HTTP and HTTPS. In addition, for listeners forward the traffic to the **Target Group** created earlier and create the load balancer keeping the rest of the default settings.



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

ThreeTier sg-0a5215b261ca5f6ff
VPC: vpc-0821ed425bc2b8961

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.

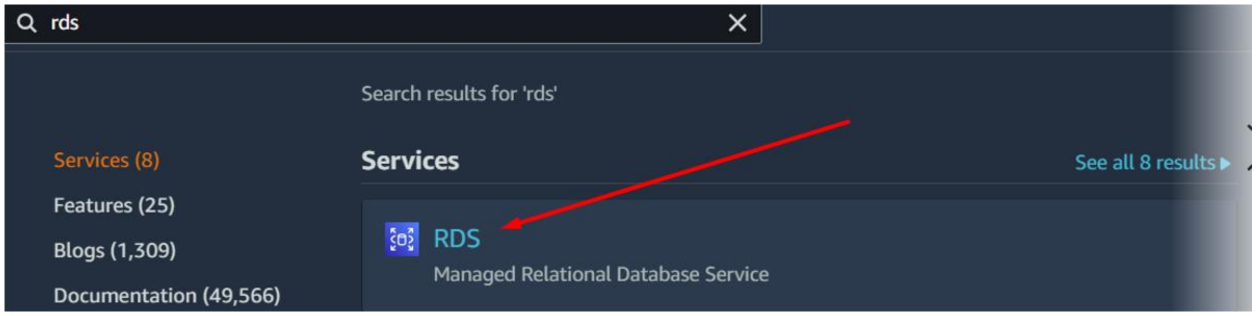
Listener HTTP:80

Protocol: HTTP
Port: 80 (1-65535)
Default action Info: Forward to ThreeTier03TG HTTP
Target type: Instance, IPv4

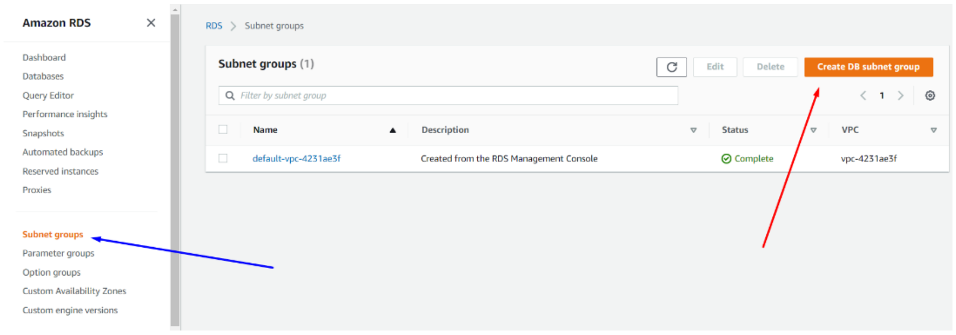
Create target group

Task 3

19. Now create a MYSQL database using AWS RDS.

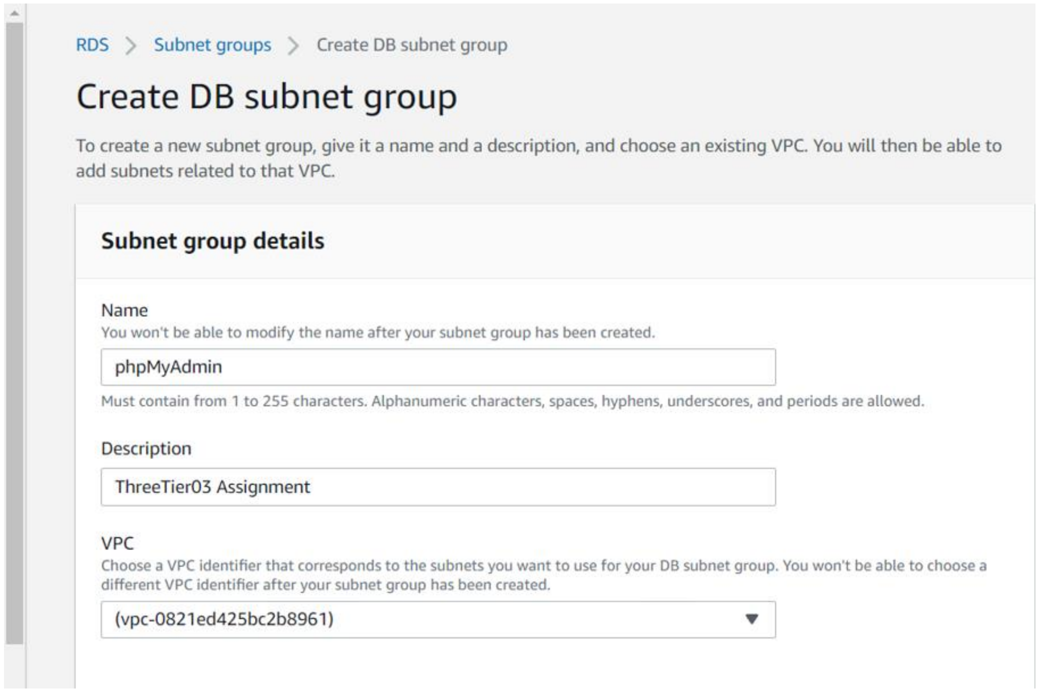


20. Now create a subnet group



Name	Description	Status	VPC
default-vpc-4231ae3f	Created from the RDS Management Console	Complete	vpc-4231ae3f

21. Configure the subnet giving it a **name**, description and choosing the **VPC** created.



RDS > Subnet groups > Create DB subnet group

Create DB subnet group

To create a new subnet group, give it a name and a description, and choose an existing VPC. You will then be able to add subnets related to that VPC.

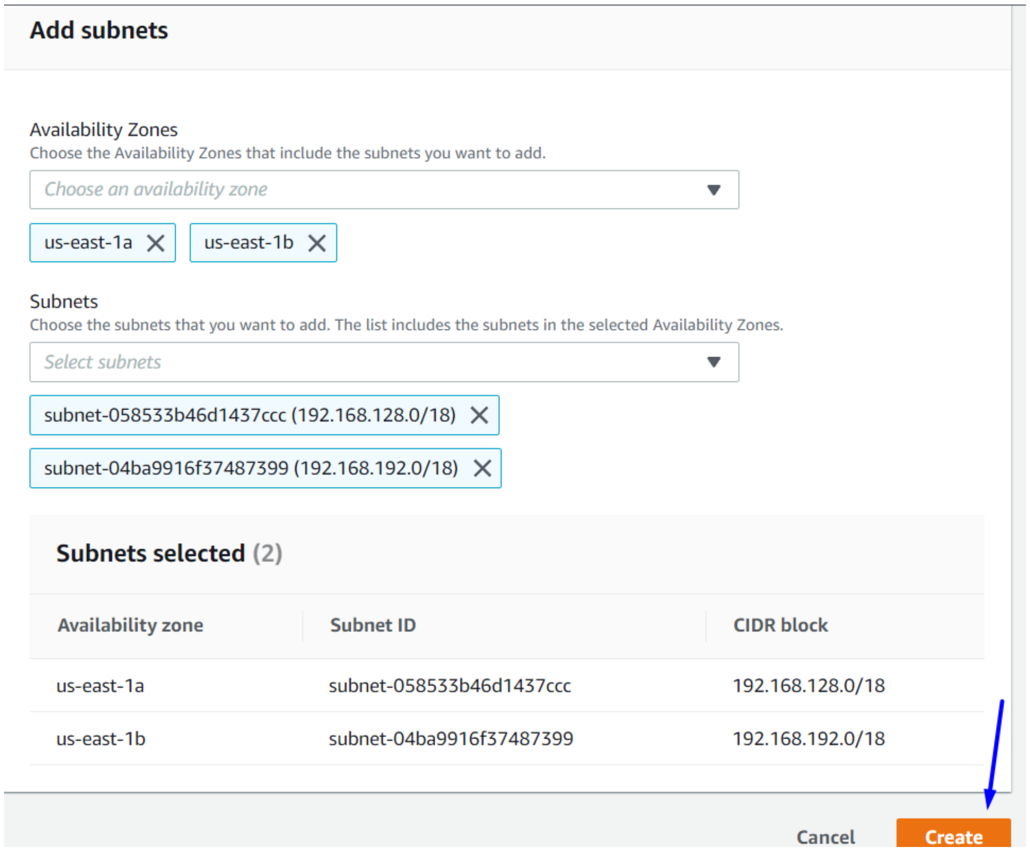
Subnet group details

Name
You won't be able to modify the name after your subnet group has been created.
phpMyAdmin
Must contain from 1 to 255 characters. Alphanumeric characters, spaces, hyphens, underscores, and periods are allowed.

Description
ThreeTier03 Assignment

VPC
Choose a VPC identifier that corresponds to the subnets you want to use for your DB subnet group. You won't be able to choose a different VPC identifier after your subnet group has been created.
(vpc-0821ed425bc2b8961)

22. Now add the subnets from the 2 **availability zones** and select the two **private** subnets.
NOTE: You can find the private subnet IP ranges inside AWS VPC service -> Subnet Association



Add subnets

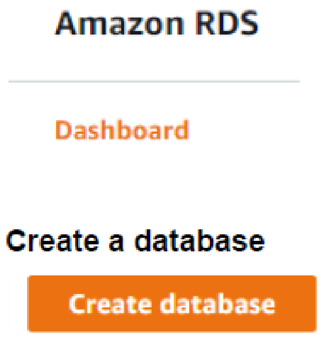
Availability Zones
Choose the Availability Zones that include the subnets you want to add.
Choose an availability zone
us-east-1a ✕ us-east-1b ✕

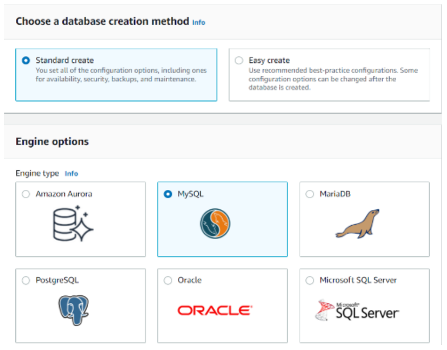
Subnets
Choose the subnets that you want to add. The list includes the subnets in the selected Availability Zones.
Select subnets
subnet-058533b46d1437ccc (192.168.128.0/18) ✕
subnet-04ba9916f37487399 (192.168.192.0/18) ✕

Subnets selected (2)

Availability zone	Subnet ID	CIDR block
us-east-1a	subnet-058533b46d1437ccc	192.168.128.0/18
us-east-1b	subnet-04ba9916f37487399	192.168.192.0/18

23. Now go back to the Dashboard to create the database.



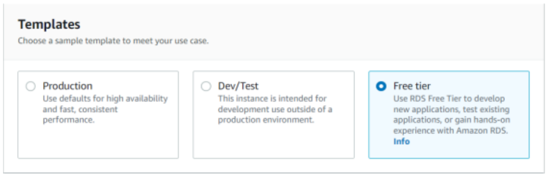


Choose a database creation method Info

[✓ 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

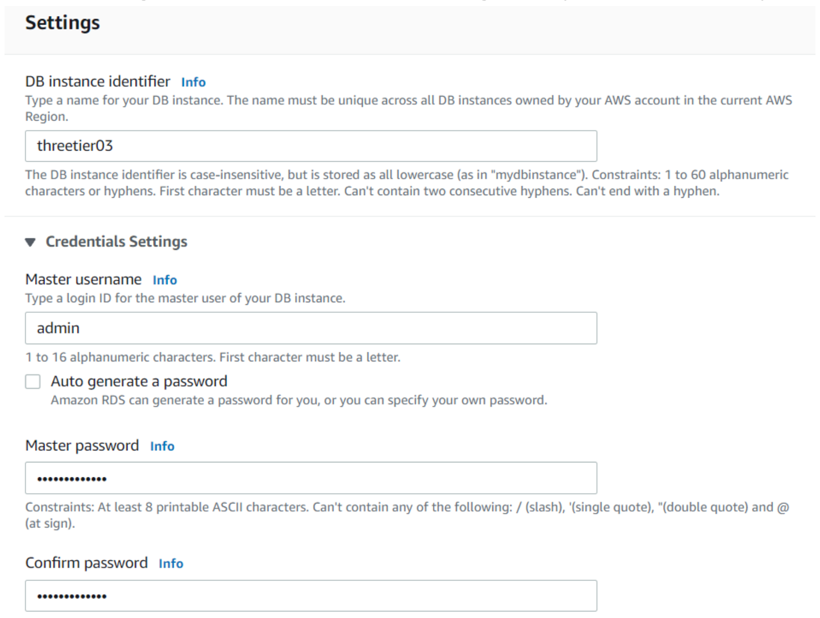
Engine type Info
Amazon Aurora [✓ MySQL] MariaDB
PostgreSQL Oracle Microsoft SQL Server



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

24. Now configure the database ensuring that you record the password in a safe place.



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.

threetier03

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.

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

Master password Info

•••••••••••••

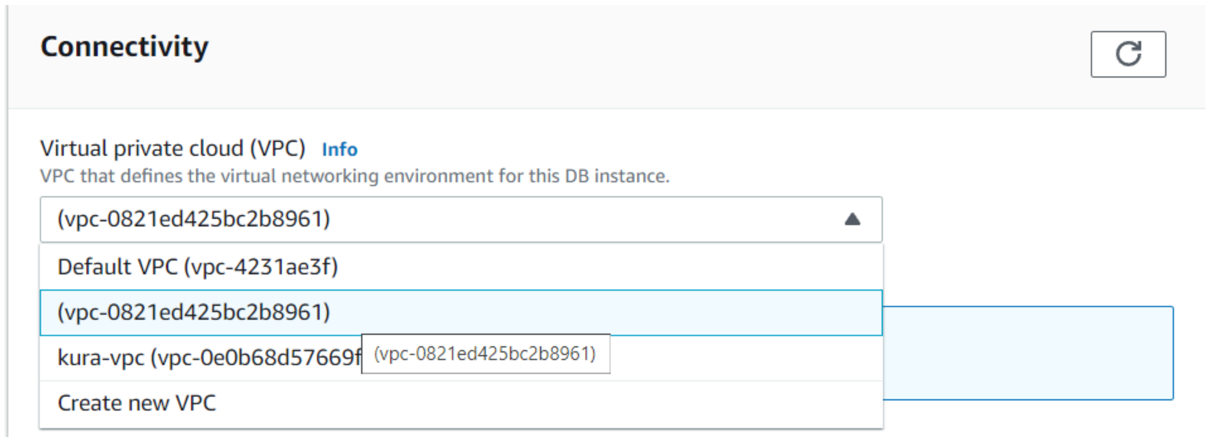
Constraints: At least 8 printable ASCII characters. Can't contain any of the following: / (slash), '(single quote), "(double quote) and @ (at sign).

Confirm password Info

•••••••••••••

Password - **threetier0003** (note should have named it **ThreeTier0003$** which is the password I used later when creating the database since I'll need to include a symbol).

25. Now choose the VPC you created.

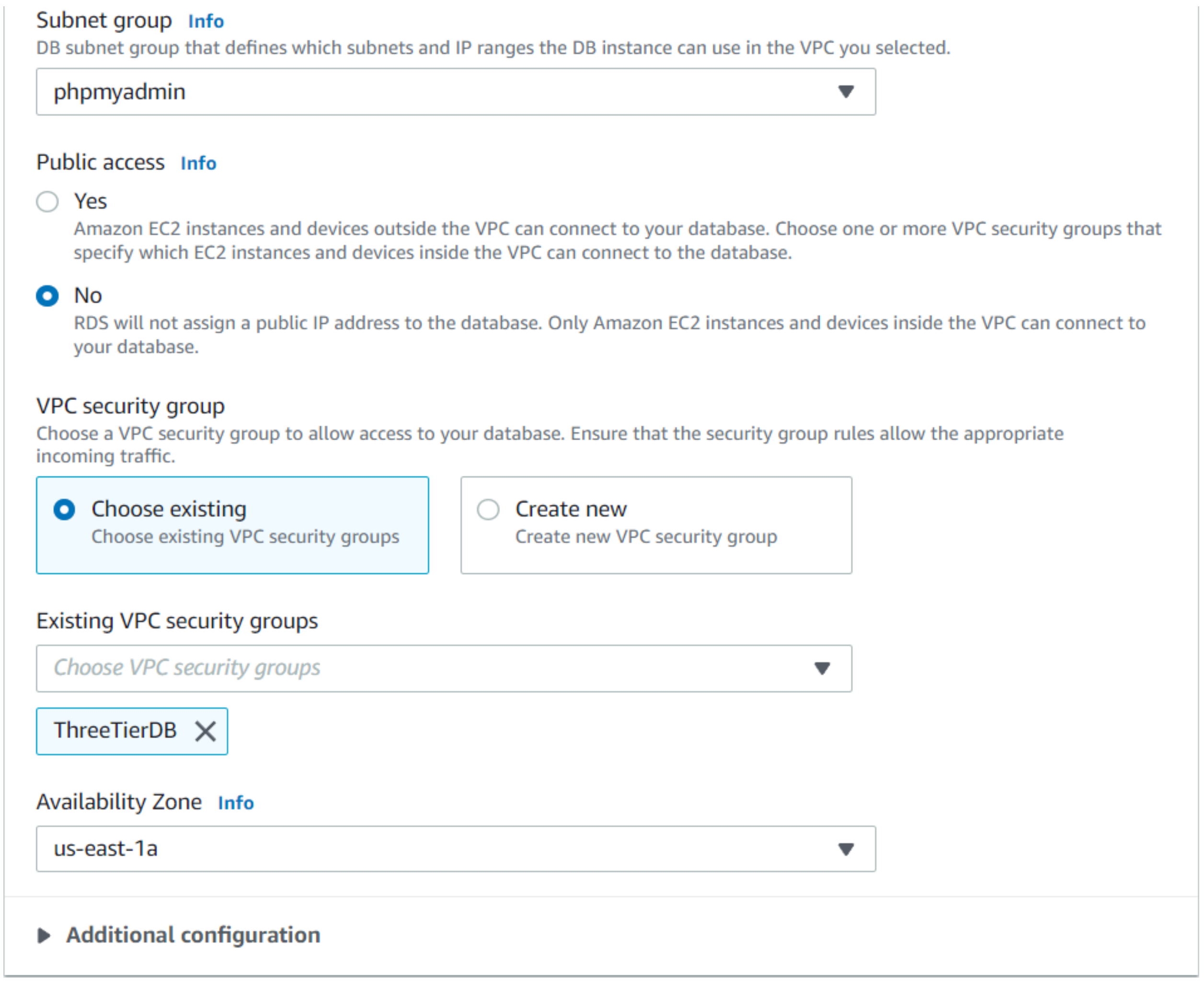


Connectivity

Virtual private cloud (VPC) Info
VPC that defines the virtual networking environment for this DB instance.

(vpc-0821ed425bc2b8961)

26. Now choose the **Subnet Group** created from earlier as well and the **VPC** security group and **availability zone**. Once done create the database.
NOTE: this will take a couple minutes to create.



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

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
ThreeTierDB

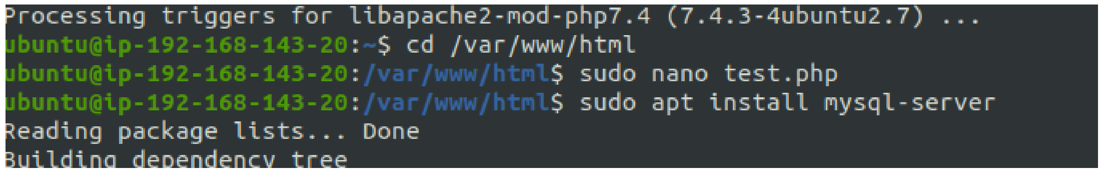
Availability Zone Info
us-east-1a

Additional configuration

Task 4

NOTE: time to set up phpMyAdmin on our EC2 and connect to our MySQL database.

27. Now SSH into the bastion EC2 then SSH into the phpMyAdmin EC2.
28. Then use the command ($ sudo apt-get update && sudo apt-get upgrade-y).
29. Then download Apache using the command ($ sudo apt-get install apache2-y).
30. After install PHP, so that php can connect to Apache and also connect to MySQL server. Use the command ($ sudo apt install php libapache2-mod-php php-mysql-y).
31. Now check to see that PHP is working by going to the directory where apache host web pages by using the command ($ cd /var/www/html).
32. Then create a PHP file using the command ($ sudo nano test.php) and paste into the file (<?php phpinfo();) and save it.
33. Now install MySQL server using the following command ($ sudo apt install mysql-server-y). Then use the command ($ sudo mysql_secure_installation) to install MySQL.



```
Processing triggers for libapache2-mod-php7.4 (7.4.3-4ubuntu2.7) ...
ubuntu@ip-192-168-143-20:~$ cd /var/www/html
ubuntu@ip-192-168-143-20:/var/www/html$ sudo nano test.php
ubuntu@ip-192-168-143-20:/var/www/html$ sudo apt install mysql-server
Reading package lists... Done
Building dependency tree
```

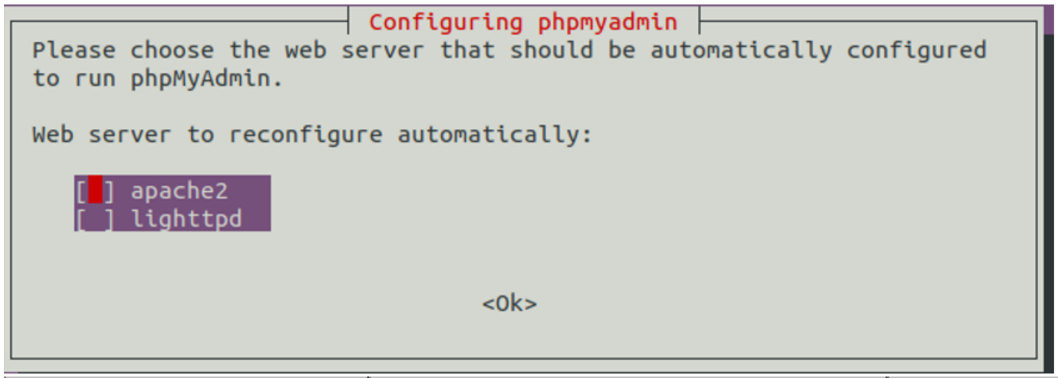
Note: Installation steps:

Y
1
Password for root user mysql: same as rds database **threetier0003**
Y
<ENTERKEY>
<ENTERKEY>
<ENTERKEY>
<ENTERKEY>

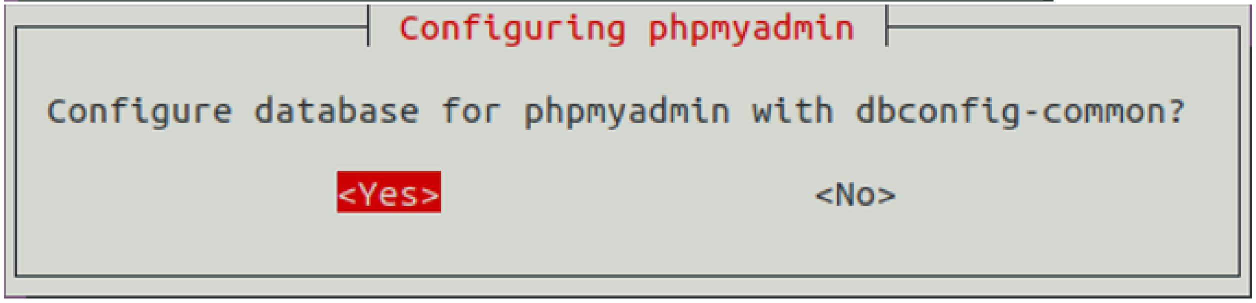
34. Enter into the interactive shell of mysql to check if installation was successful. Use the following commands:
sudo mysql
show databases;
exit

35. Now use the command ($ sudo apt install phpmyadmin php-mbstring php-zip php-gd php-json php-curl -y) to download other necessary packages.

36. When a prompt comes up complete it as follows:
First, select apache2

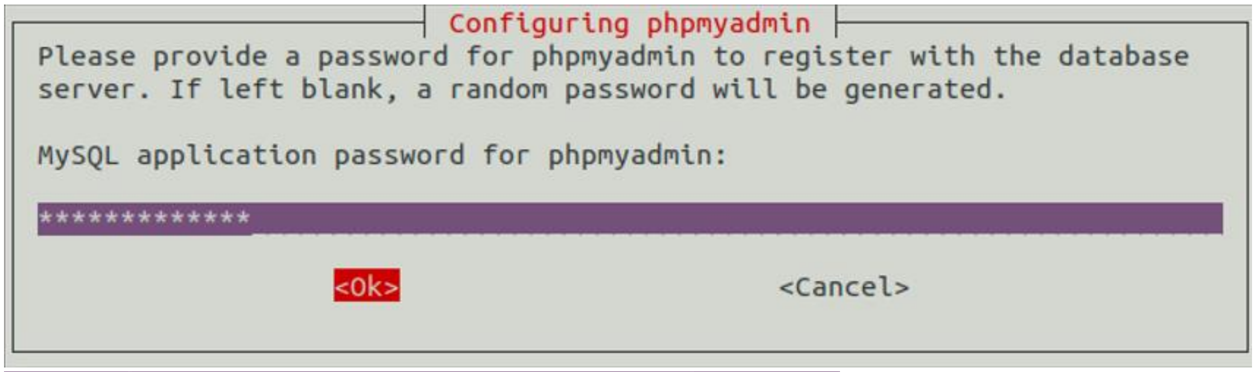


```
Configuring phpmyadmin
Please choose the web server that should be automatically configured
to run phpMyAdmin.

Web server to reconfigure automatically:

    [ ] apache2
    [ ] lighttpd

                          <Ok>
```

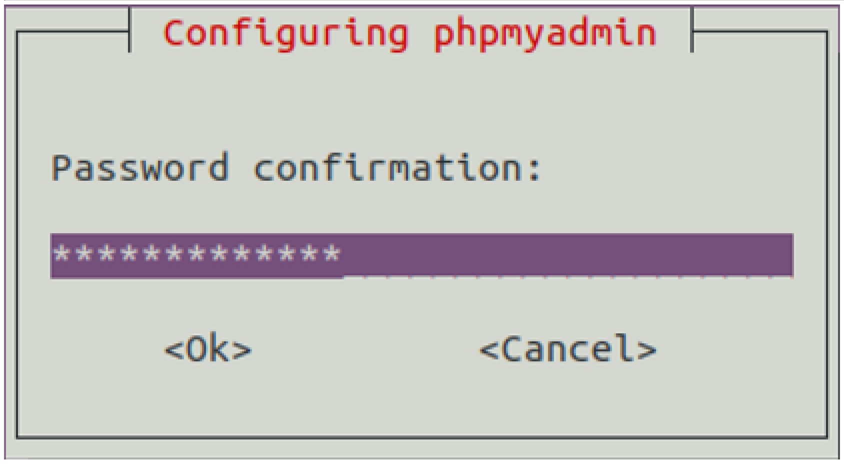


```
Configuring phpmyadmin

Configure database for phpmyadmin with dbconfig-common?

          <Yes>                    <No>
```

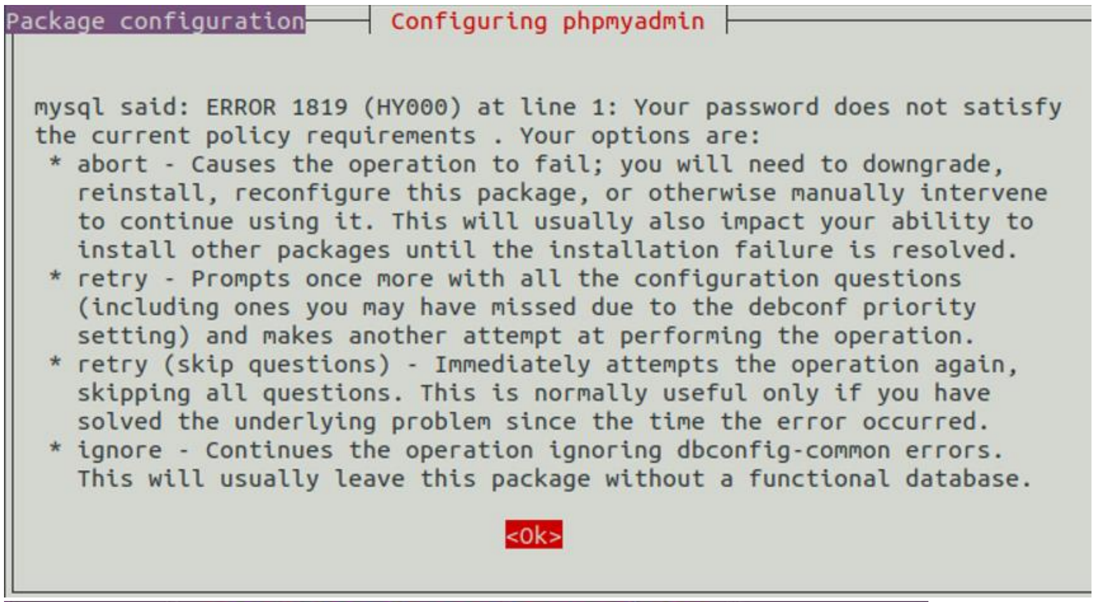
When the next prompt comes, we will have to enter a password. We can use **threetier0003** from the database we created.



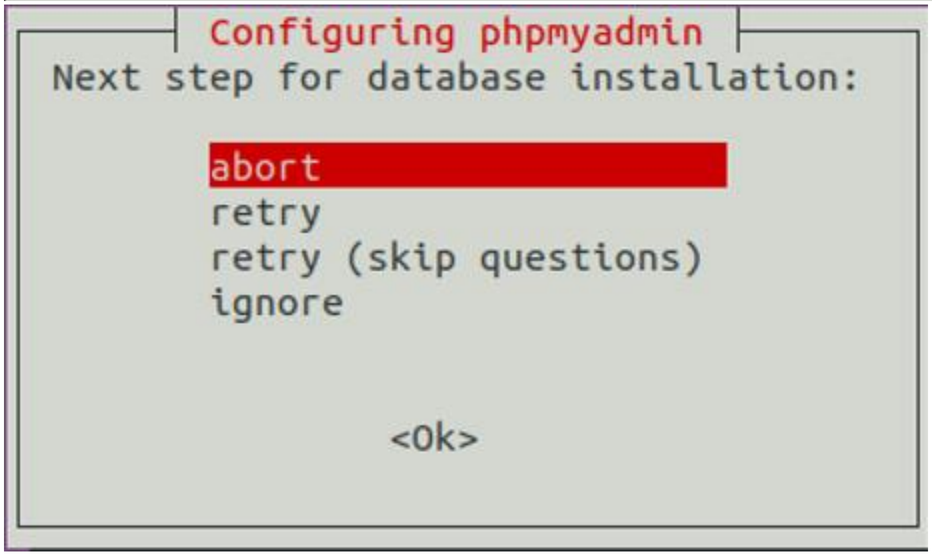
```
Configuring phpmyadmin
Please provide a password for phpmyadmin to register with the database
server. If left blank, a random password will be generated.

MySQL application password for phpmyadmin:

************

          <Ok>                    <Cancel>
```



```
Configuring phpmyadmin

Password confirmation:

*************

     <Ok>          <Cancel>
```

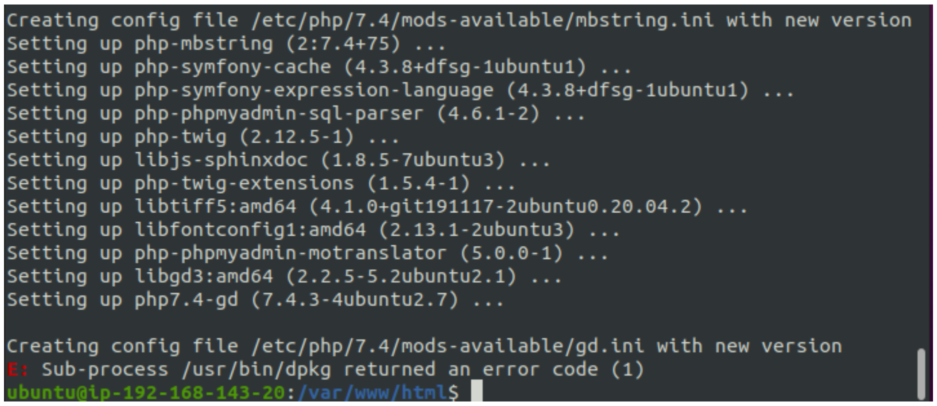


```
Package configuration ──┤ Configuring phpmyadmin ├──

 mysql said: ERROR 1819 (HY000) at line 1: Your password does not satisfy
 the current policy requirements . Your options are:
  * abort - Causes the operation to fail; you will need to downgrade,
    reinstall, reconfigure this package, or otherwise manually intervene
    to continue using it. This will usually also impact your ability to
    install other packages until the installation failure is resolved.
  * retry - Prompts once more with all the configuration questions
    (including ones you may have missed due to the debconf priority
    setting) and makes another attempt at performing the operation.
  * retry (skip questions) - Immediately attempts the operation again,
    skipping all questions. This is normally useful only if you have
    solved the underlying problem since the time the error occurred.
  * ignore - Continues the operation ignoring dbconfig-common errors.
    This will usually leave this package without a functional database.

                                  <Ok>
```



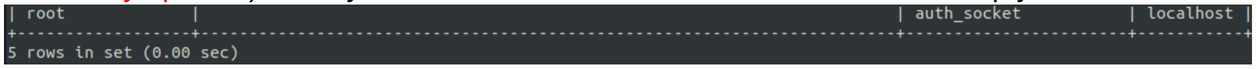
```
┤ Configuring phpmyadmin ├
 Next step for database installation:

          abort
          retry
          retry (skip questions)
          ignore

                  <Ok>
```

NOTE: received this error here and although I'm unsure why I was able to complete everything.



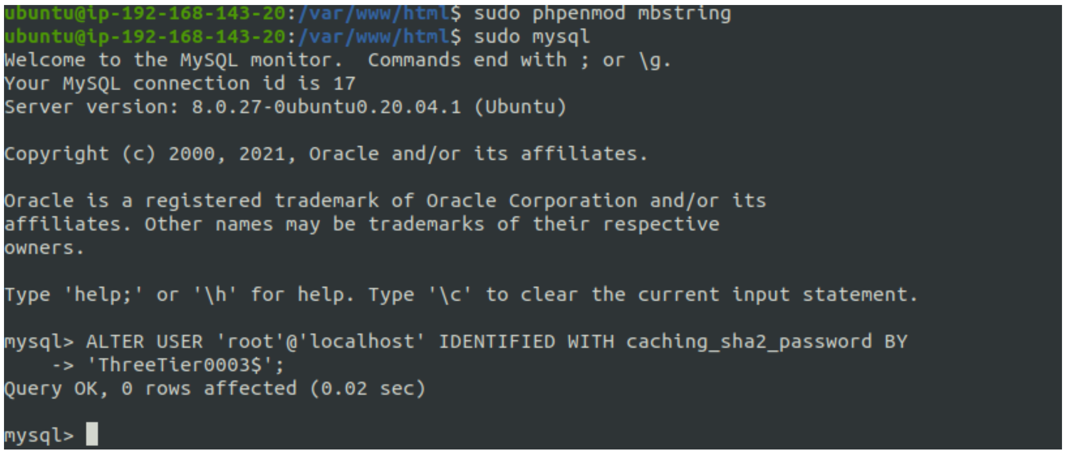
```
Creating config file /etc/php/7.4/mods-available/mbstring.ini with new version
Setting up php-mbstring (2:7.4+75) ...
Setting up php-symfony-cache (4.3.8+dfsg-1ubuntu1) ...
Setting up php-symfony-expression-language (4.3.8+dfsg-1ubuntu1) ...
Setting up php-phpmyadmin-sql-parser (4.6.1-2) ...
Setting up php-twig (2.12.5-1) ...
Setting up libjs-sphinxdoc (1.8.5-7ubuntu3) ...
Setting up php-twig-extensions (1.5.4-1) ...
Setting up libtiff5:amd64 (4.1.0+git191117-2ubuntu0.20.04.2) ...
Setting up libfontconfig1:amd64 (2.13.1-2ubuntu3) ...
Setting up php-phpmyadmin-motranslator (5.0.0-1) ...
Setting up libgd3:amd64 (2.2.5-5.2ubuntu2.1) ...
Setting up php7.4-gd (7.4.3-4ubuntu2.7) ...

Creating config file /etc/php/7.4/mods-available/gd.ini with new version
E: Sub-process /usr/bin/dpkg returned an error code (1)
ubuntu@ip-192-168-143-20:/var/www/html$
```

37. Now log back into mysql using the command ($ sudo mysql).
38. Then paste the following command (> SELECT user,authentication_string,plugin,host FROM mysql.user;). Here you should see that inside the table the root is empty.



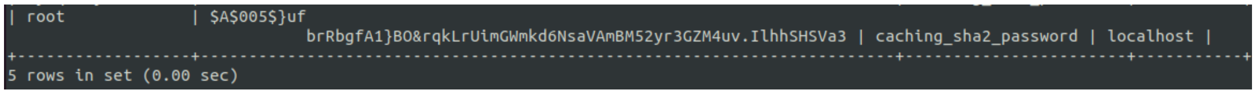
```
| root             |                                                                          | auth_socket           | localhost |
+------------------+--------------------------------------------------------------------------+-----------------------+-----------+
5 rows in set (0.00 sec)
```

39. Then run the command (> UNINSTALL COMPONENT "file://component_validate_password";) inside the MySQL interactive shell.
40. Followed by the command (> INSTALL COMPONENT "file://component_validate_password";) and then leave the shell by typing "**exit**".
41. Then install another package using the command ($ sudo phpenmod mbstring).
42. After return to MySQL using the command ($ sudo mysql).
43. Then add a password which will be stored into the root localhost field using the command (> ALTER USER 'root'@'localhost' IDENTIFIED WITH caching_sha2_password BY ' ThreeTier0003$;). **NOTE:** here is where the password changes from what I used earlier.



```
ubuntu@ip-192-168-143-20:/var/www/html$ sudo phpenmod mbstring
ubuntu@ip-192-168-143-20:/var/www/html$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 17
Server version: 8.0.27-0ubuntu0.20.04.1 (Ubuntu)

Copyright (c) 2000, 2021, Oracle and/or its affiliates.

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

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

mysql> ALTER USER 'root'@'localhost' IDENTIFIED WITH caching_sha2_password BY
    -> 'ThreeTier0003$';
Query OK, 0 rows affected (0.02 sec)

mysql> 
```

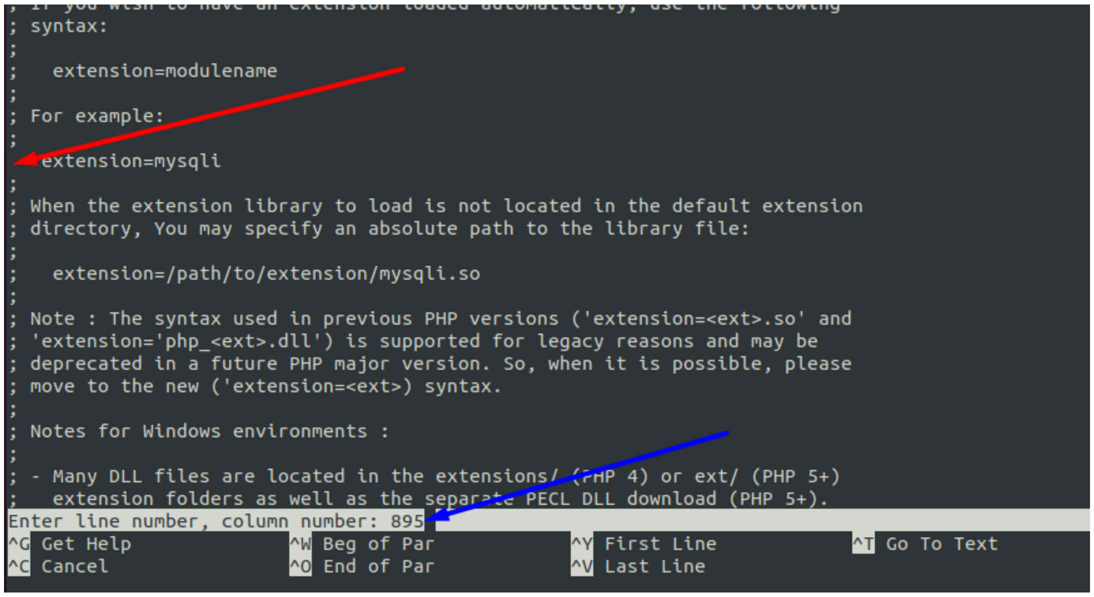
44. Then check to see if the changes were made using the command (> SELECT user,authentication_string,plugin,host FROM mysql.user;) and then leave the shell by typing "**exit**".



```
| root             | $A$005$}uf
                          brRbgfA1}BO&rqkLrUimGWmkd6NsaVAmBM52yr3GZM4uv.IlhhSHSVa3 | caching_sha2_password | localhost |
+------------------+--------------------------------------------------------------------------+-----------------------+-----------+
5 rows in set (0.00 sec)
```

45. Then change the directory to apache2 using the command ($ cd /etc/php/7.4/apache2/).
46. Then edit the file using the command ($ sudo nano php.ini).

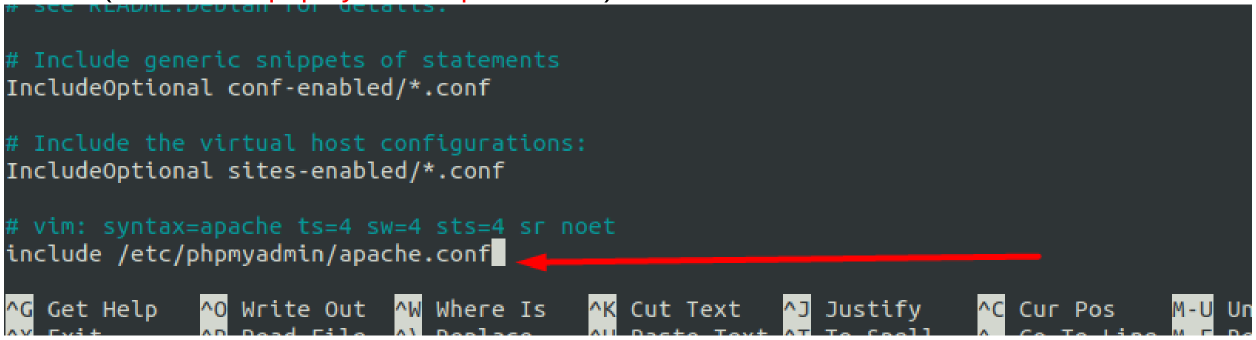
47. Inside Nano select ALT + G. This will allow us to go to a line. Go to line 895 and remove the “**semicolon ;**”.



```
; syntax:
;
;   extension=modulename
;
; 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 :
;
; - Many DLL files are located in the extensions/ (PHP 4) or ext/ (PHP 5+)
;   extension folders as well as the separate PECL DLL download (PHP 5+).
Enter line number, column number: 895
^G Get Help              ^W Beg of Par            ^Y First Line            ^T Go To Text
^C Cancel                ^O End of Par            ^V Last Line
```

Save and exit the file using CTRL + O followed by CTRL + X

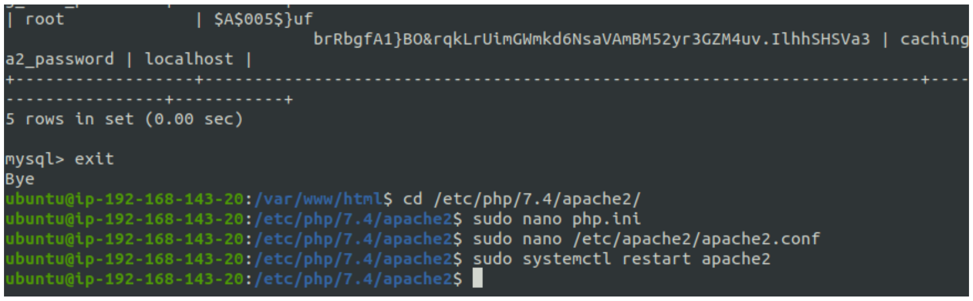
48. Now edit the apache2 config file using the command ($ sudo nano /etc/apache2/apache2.conf). Then scroll to the bottom and add the following at the bottom (include /etc/phpmyadmin/apache.conf).



```
# 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

^G Get Help    ^O Write Out   ^W Where Is    ^K Cut Text    ^J Justify     ^C Cur Pos     M-U Un
```

Save and exit the file using CTRL + O followed by CTRL + X

49. Then use the command ($ sudo systemctl restart apache2) to restart apache2



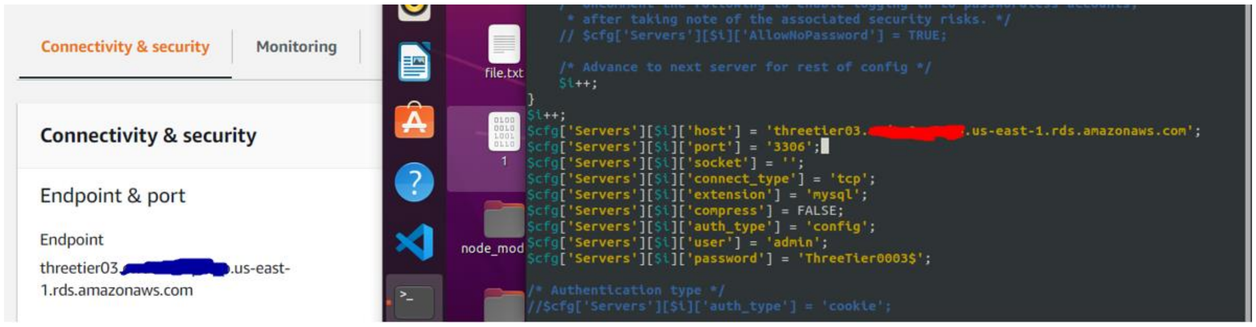
```
| root              | $A$005$}uf
                                brRbgfA1}BO&rqkLrUimGWmkd6NsaVAmBM52yr3GZM4uv.IlhhSHSVa3 | caching
a2_password | localhost |
+-----------------+-----------------------------------------------------------------------+----
----------------+-----------+
5 rows in set (0.00 sec)

mysql> exit
Bye
ubuntu@ip-192-168-143-20:/var/www/html$ cd /etc/php/7.4/apache2/
ubuntu@ip-192-168-143-20:/etc/php/7.4/apache2$ sudo nano php.ini
ubuntu@ip-192-168-143-20:/etc/php/7.4/apache2$ sudo nano /etc/apache2/apache2.conf
ubuntu@ip-192-168-143-20:/etc/php/7.4/apache2$ sudo systemctl restart apache2
ubuntu@ip-192-168-143-20:/etc/php/7.4/apache2$
```

50. Now we need to connect our MySQL database hosted on AWS to our phpMyAdmin. So edit the config file using the command ($ sudo nano /etc/phpmyadmin/config.inc.php).
51. Inside nano select ALT + G. This will allow us to go to a line. Go to line 102 and paste thenfollowing below:

```
$i++;
$cfg['Servers'][$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__';
```

NOTE:
Ensure you put in your necessary details in the lines that read **__FILL_IN_DETAILS__.**

- Host is the endpoint URL found on the AWS RDS database we created
- Enter the username and password in the user and password line of the code



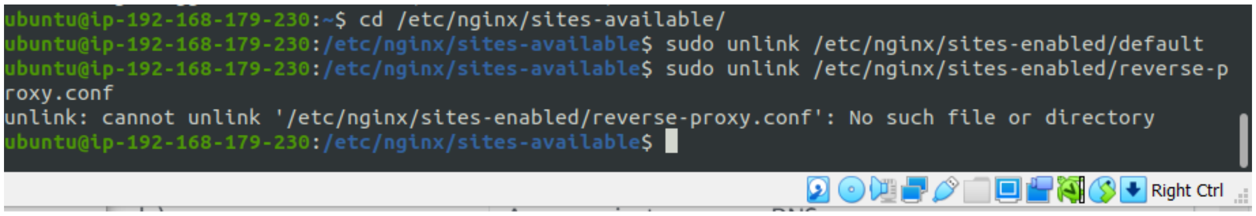
threetier03.[redacted].us-east-1.rds.amazonaws.com

```
     * 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'] = 'threetier03.[redacted].us-east-1.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'] = 'ThreeTier0003$';

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

52. Now test the connection to the new database b using the command ($ curl localhost:80/phpmyadmin/) to log into your AWS RDS.
53. Now restart your nginx using the command ($ sudo systemctl restart apache2) and exit out of the phpMyAdmin EC2 by typing "**exit**".
54. Now connect with your NGNIX EC2 instance by SSHing into the Bastion EC2 and then ssh into the NGINX EC2.
55. Then use the command ($ sudo apt-get update && sudo apt-get upgrade -y).
56. Once complete install NGINX using the command ($ sudo apt-get install nginx -y).

57. Change directories to Sites available. N.B. Sites-available are conf files that tell NGINX where to look for. Use the command ($ cd /etc/nginx/sites-available/).
58. Now unlink the default sites-enabled file using the commands ($ sudo unlink /etc/nginx/sites-enabled/default) followed by ($ sudo unlink /etc/nginx/sites-enabled/reverse-proxy.conf).

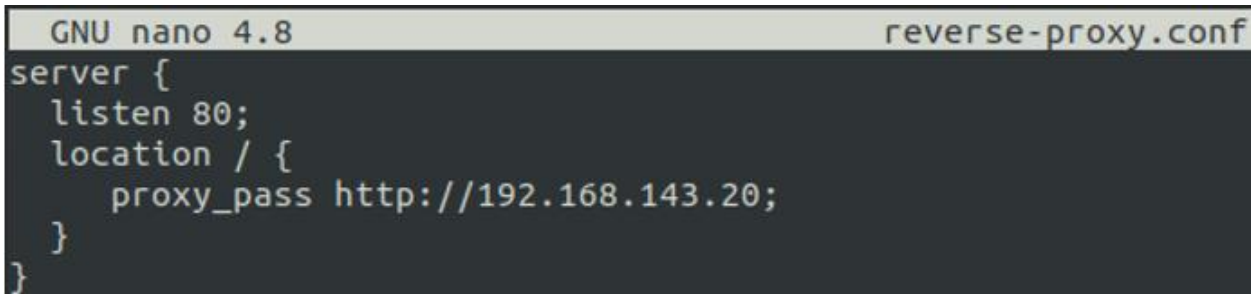
    NOTE: Unlinking will say there is no file, therefore we need to create a configuration file for the reverse proxy.



```
ubuntu@ip-192-168-179-230:~$ cd /etc/nginx/sites-available/
ubuntu@ip-192-168-179-230:/etc/nginx/sites-available$ sudo unlink /etc/nginx/sites-enabled/default
ubuntu@ip-192-168-179-230:/etc/nginx/sites-available$ sudo unlink /etc/nginx/sites-enabled/reverse-p
roxy.conf
unlink: cannot unlink '/etc/nginx/sites-enabled/reverse-proxy.conf': No such file or directory
ubuntu@ip-192-168-179-230:/etc/nginx/sites-available$ 
```

59. Use the command ($ sudo nano reverse-proxy.conf) and paste the following ensuring that the **proxy_pass IP** is the **phpMyAdmin private IPv4**.

```
server {
   listen 80;
   location / {
      proxy_pass http://192.168.143.20;
   }
}
```



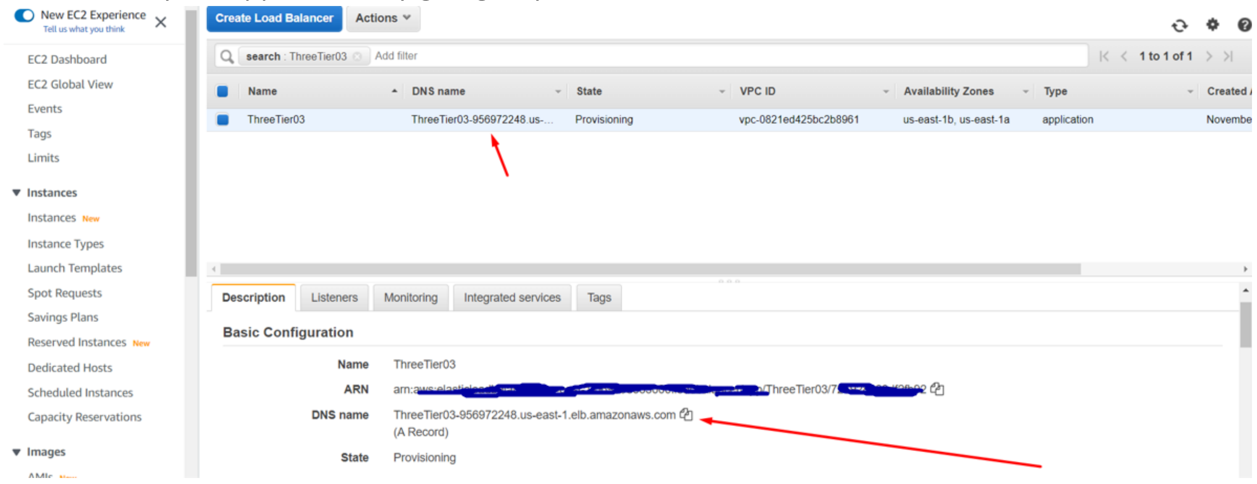
```
  GNU nano 4.8                     reverse-proxy.conf
server {
  listen 80;
  location / {
     proxy_pass http://192.168.143.20;
  }
}
```

Save and exit the file using CTRL + O followed by CTRL + X

60. Then use the command ($ ls /etc/nginx/sites-enabled/) to check if the directory is empty.
61. Now use the command ($ sudo ln -s /etc/nginx/sites-available/reverse-proxy.conf /etc/nginx/sites-enabled/reverse-proxy.conf) to link the reverse-proxy to sites so that Apache can read and use it.
62. Then restart NGINX using the command ($ sudo systemctl restart nginx).

Task 5

63. Now access your application by going to your load balancer on AWS

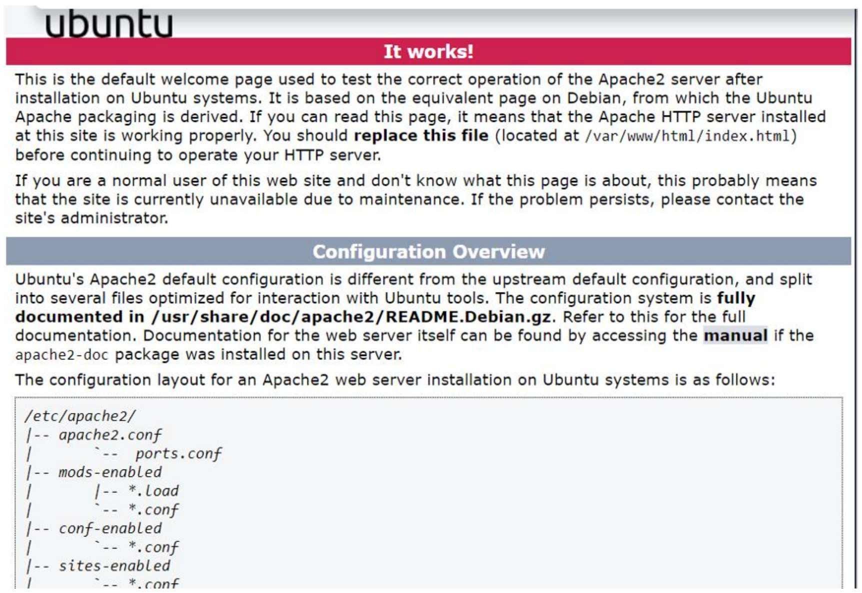


Name	DNS name	State	VPC ID	Availability Zones	Type	Created
ThreeTier03	ThreeTier03-956972248.us-...	Provisioning	vpc-0821ed425bc2b8961	us-east-1b, us-east-1a	application	Novembe

Basic Configuration

Name	ThreeTier03
ARN	arn:[redacted]/ThreeTier03/[redacted]
DNS name	ThreeTier03-956972248.us-east-1.elb.amazonaws.com (A Record)
State	Provisioning

64. Select the application by pasting your DNS name into your browser.



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.

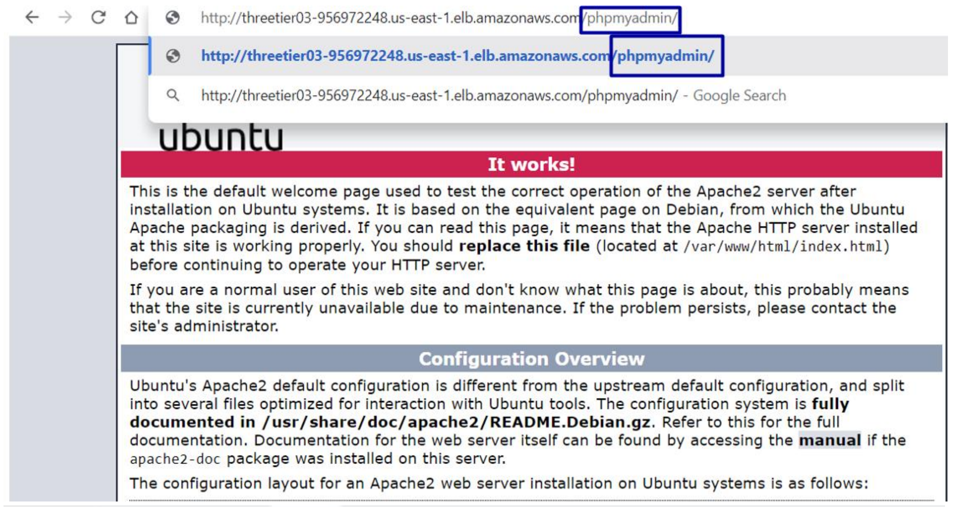
Configuration Overview

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
```

65. To access our phpMyAdmin application, we will have to put a route in the URL. The format will be http://url/phpmyadmin/.



http://threetier03-956972248.us-east-1.elb.amazonaws.com/phpmyadmin/

http://threetier03-956972248.us-east-1.elb.amazonaws.com/phpmyadmin/

http://threetier03-956972248.us-east-1.elb.amazonaws.com/phpmyadmin/ - Google Search

ubuntu

It works!

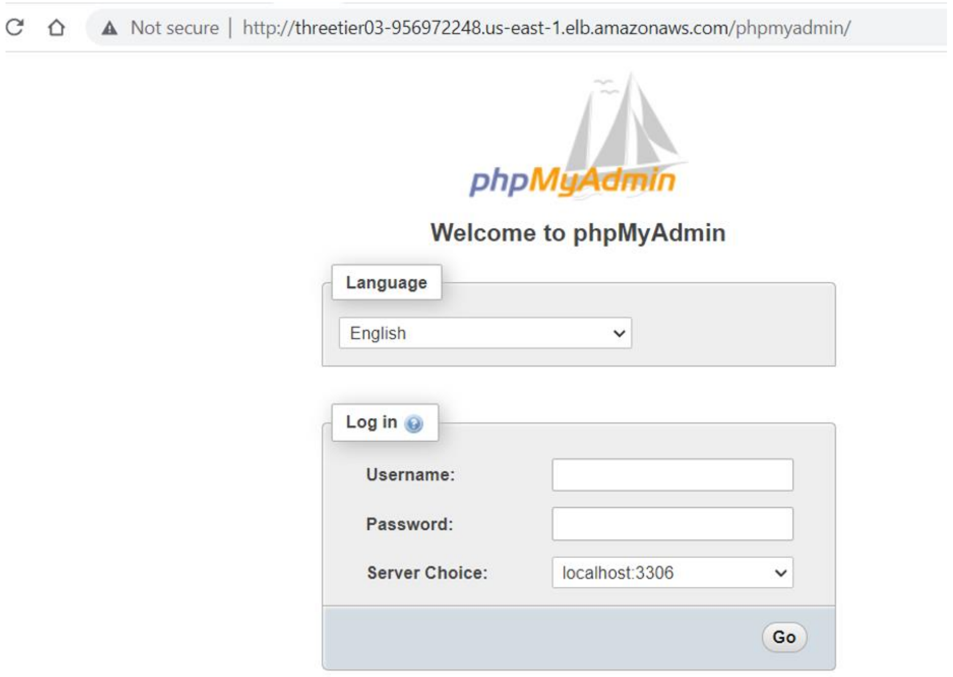
This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at 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:



Not secure | http://threetier03-956972248.us-east-1.elb.amazonaws.com/phpmyadmin/

Welcome to phpMyAdmin

Language

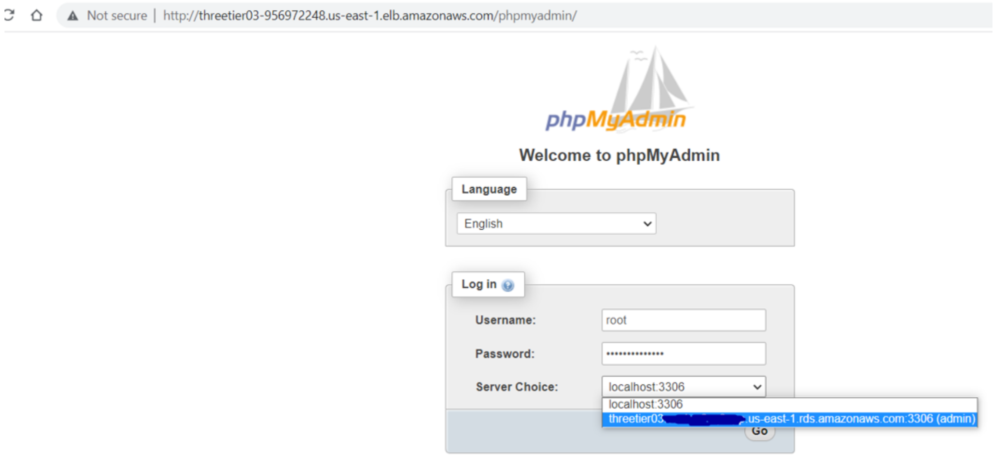
English

Log in

Username:	
Password:	
Server Choice:	localhost:3306

Go

66. You will be able to see the RDS has been configured to the server you choose



Not secure | http://threetier03-956972248.us-east-1.elb.amazonaws.com/phpmyadmin/

Welcome to phpMyAdmin

Language

English

Log in

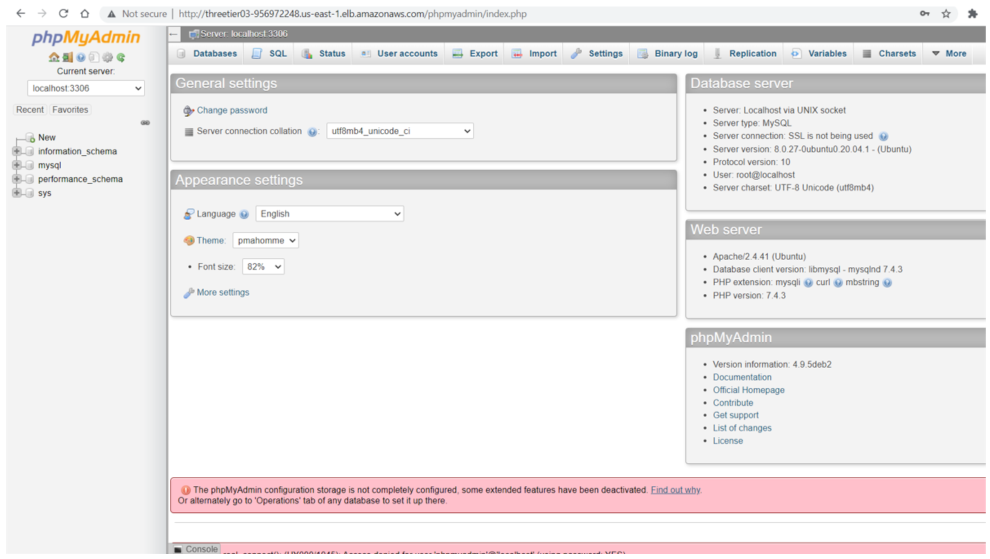
Username: root

Password: ••••••••••••••

Server Choice: localhost:3306

localhost:3306

threetier03.[redacted].us-east-1.rds.amazonaws.com:3306 (admin)



Not secure | http://threetier03-956972248.us-east-1.elb.amazonaws.com/phpmyadmin/index.php

General settings

- Change password
- Server connection collation: utf8mb4_unicode_ci

Appearance settings

- Language: English
- Theme: pmahomme
- Font size: 82%
- More settings

Database server

- Server: Localhost via UNIX socket
- Server type: MySQL
- Server connection: SSL is not being used
- Server version: 8.0.27-0ubuntu0.20.04.1 - (Ubuntu)
- Protocol version: 10
- User: root@localhost
- Server charset: UTF-8 Unicode (utf8mb4)

Web server

- Apache/2.4.41 (Ubuntu)
- Database client version: libmysql - mysqlnd 7.4.3
- PHP extension: mysqli curl mbstring
- PHP version: 7.4.3

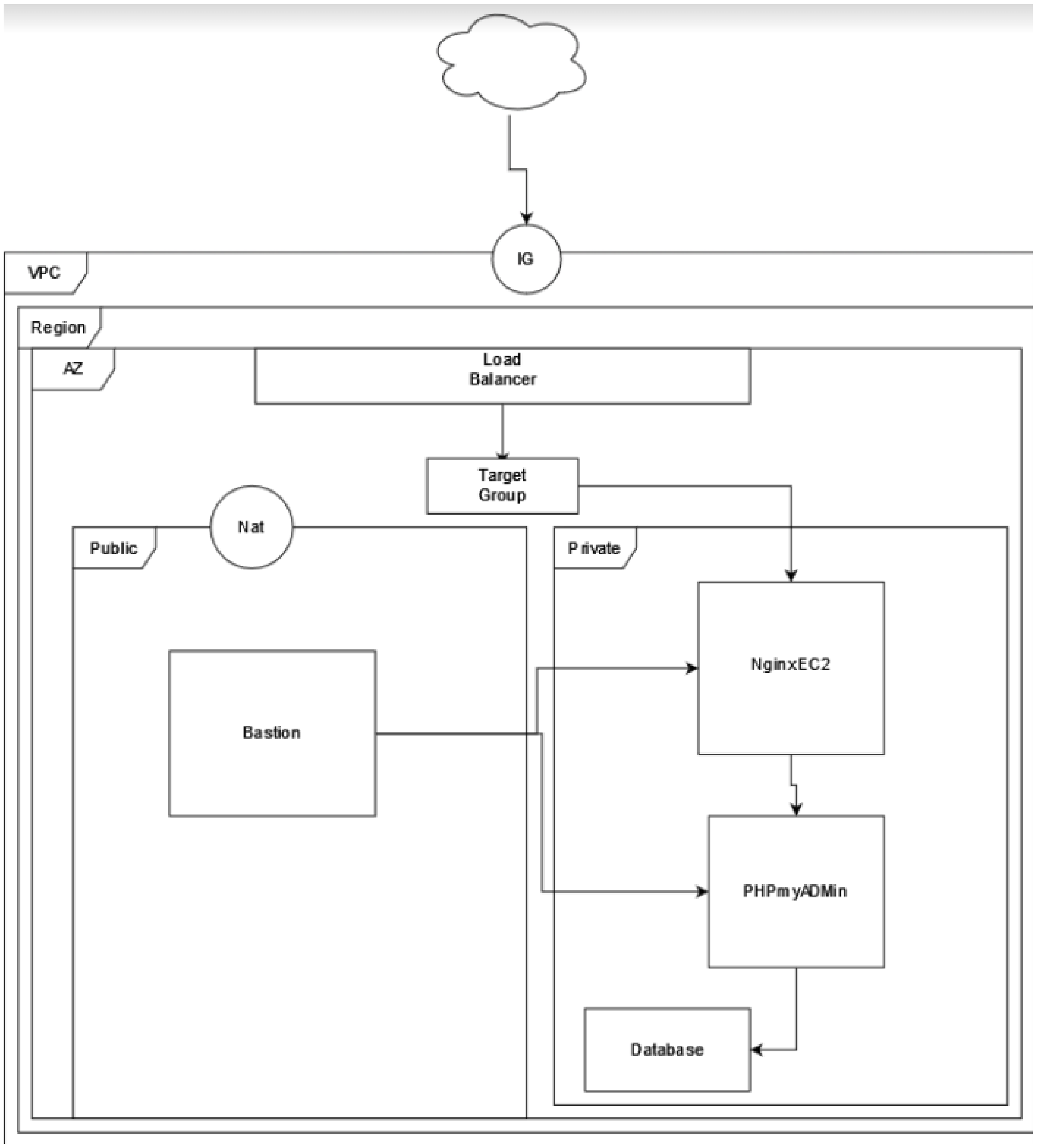
phpMyAdmin

- Version information: 4.9.5deb2
- Documentation
- Official Homepage
- Contribute
- Get support
- List of changes
- License

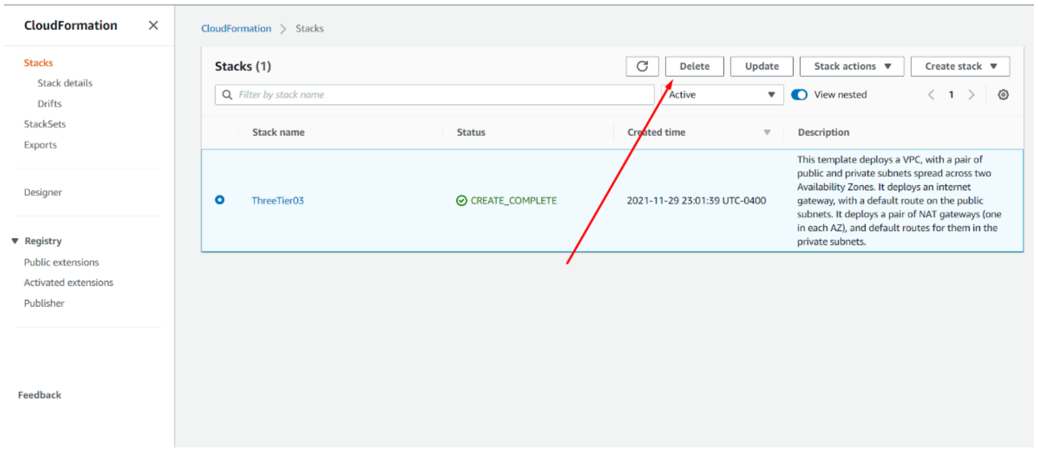
The phpMyAdmin configuration storage is not completely configured, some extended features have been deactivated. Find out why.
Or alternately go to 'Operations' tab of any database to set it up there.

Console real_connect(): (HY000/1045): Access denied for user 'phpmyadmin'@'localhost' (using password: YES)

Topology

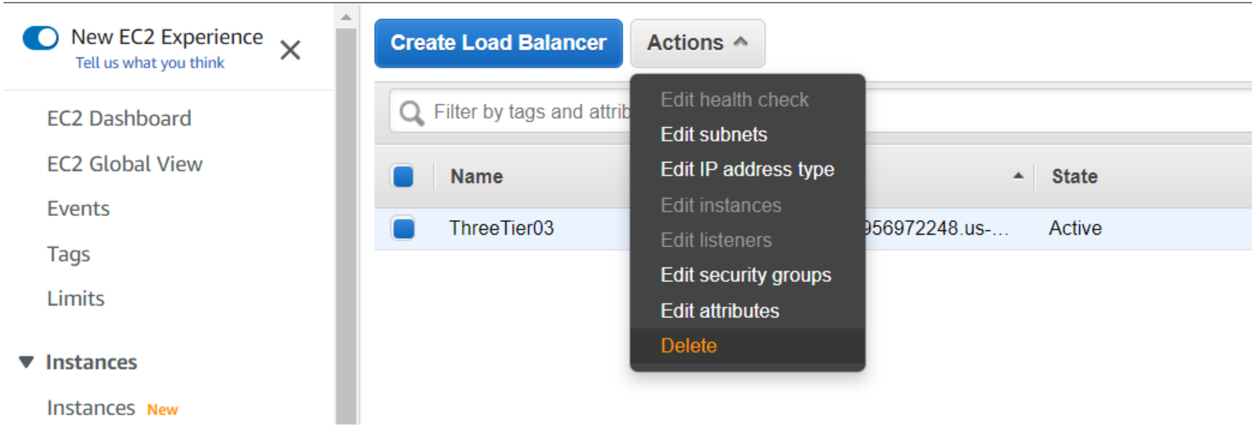

IG
VPC
Region
AZ
Load Balancer
Target Group
Nat
Public
Private
Bastion
NginxEC2
PHPmyADMin
Database

DELETE EVERYTHING

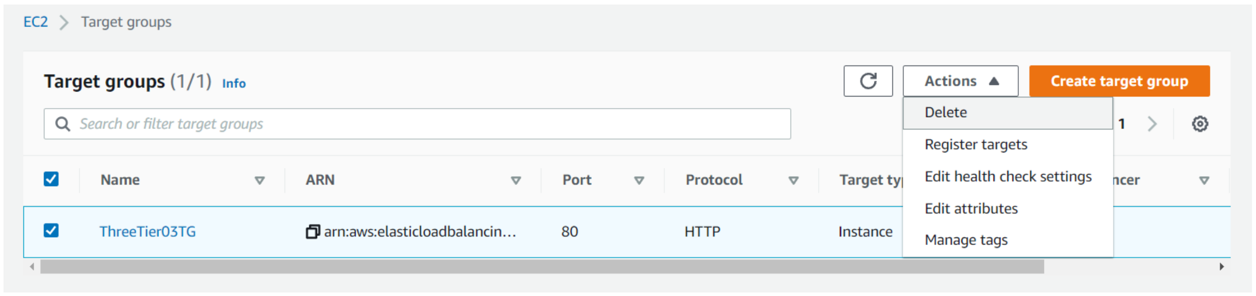


Stack name	Status	Created time	Description
ThreeTier03	CREATE_COMPLETE	2021-11-29 23:01:39 UTC-0400	This template deploys a VPC, with a pair of public and private subnets spread across two Availability Zones. It deploys an internet gateway, with a default route on the public subnets. It deploys a pair of NAT gateways (one in each AZ), and default routes for them in the private subnets.

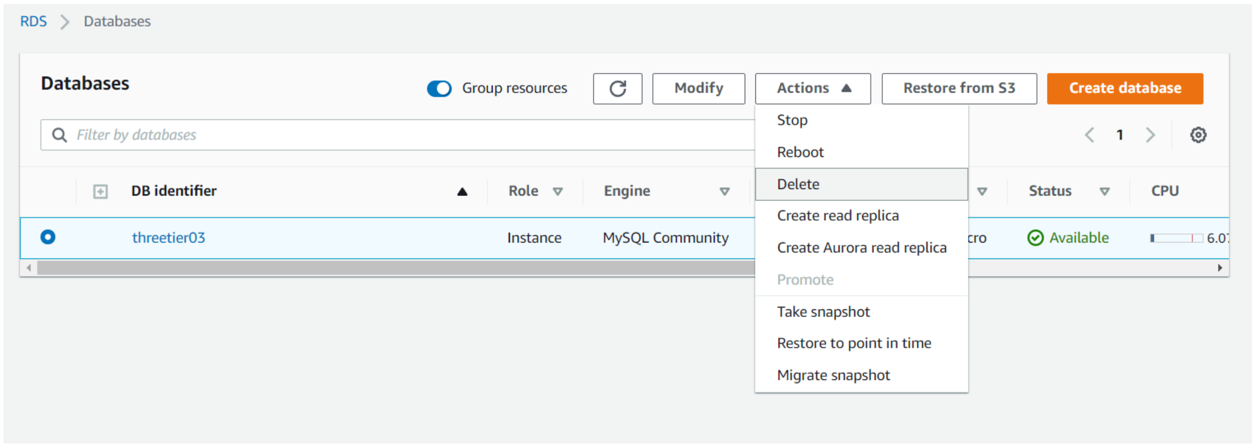
NOTE: this will take around 10 mins to fully delete. So can delete the other things you created in the mean time



Name		State
ThreeTier03	956972248.us-…	Active

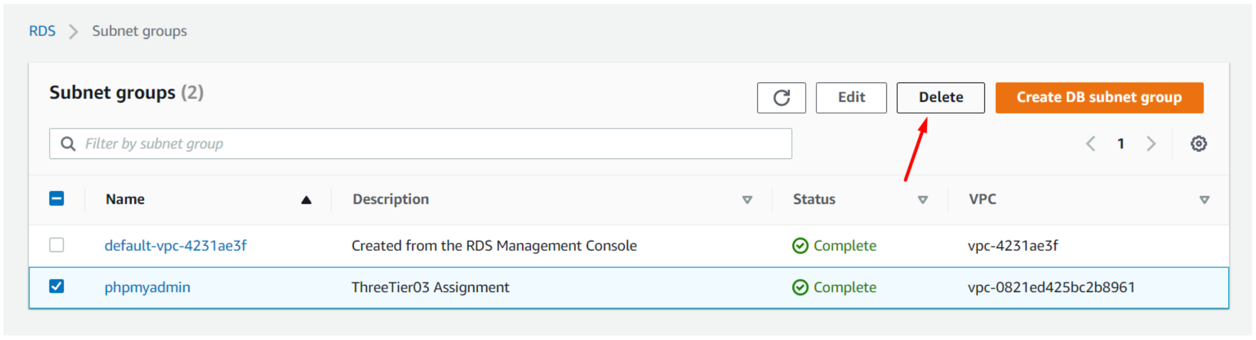


Name	ARN	Port	Protocol	Target ty	ncer
ThreeTier03TG	arn:aws:elasticloadbalancin…	80	HTTP	Instance	



DB identifier	Role	Engine		Status	CPU
threetier03	Instance	MySQL Community	cro	Available	6.0

NOTE: this will take some time to delete



Name	Description	Status	VPC
default-vpc-4231ae3f	Created from the RDS Management Console	Complete	vpc-4231ae3f
phpmyadmin	ThreeTier03 Assignment	Complete	vpc-0821ed425bc2b8961