A picture containing diagram

Description automatically generated

Overview

The following resources have been created on your AWS account:

**VPC**

Pefoma-vpc – this is the Virtual Private Cloud created in the London region and consisting of the 10.0.0.0/24 CIDR ranges

**Subnets**

The Pefoma-vpc is segmented into smaller subnetworks that contain your resources.

Pefoma-Public-1 – this is the first public subnet and is located in EU-West-2b – this is split into the CIDR range of 10.0.0.0/28

Pefoma-Public-2– this is the second public subnet and is located in EU-West-2c

this is split into the CIDR range of 10.0.0.16/28

Pefoma-Private-Subnet-Ec2-1 – this is the first EC2 private subnet and is located in EU-West-2b – this is split into the CIDR range of 10.0.0.32/27

Pefoma-Private-Subnet-Ec2-2 – this is the second EC2 private subnet and is located in EU-West-2c – this is split into the CIDR range of 10.0.0.64/26

Pefoma-Private-Subnet-rds-1 – this is the first private RDS subnet and is located in EU-West-2b – this is split into the CIDR range of 10.0.0.32/27

Pefoma-Private-Subnet-rds-2 – this is the second RDS private subnet and is located in EU-West-2c – this is split into the CIDR range of 10.0.0.64/26

**EC2 Instances**

These are Elastic Compute instances. These will be in the application layer of your infrastructure.

Auto Scaling Group – There is an auto scaling group called Pefoma-ASG active across Pefoma-Private-Subnet-Ec2-1 & Pefoma-Private-Subnet-Ec2-2. This group controls the deployment and destruction of EC2 instances in these subnets.

The settings are currently at minimum of 1 instance, maximum of 5 instances, ideally 1 instance. These instances are deployed when a health check is failed twice within 300 seconds or when the CPU usage is at 40% or higher. Any instances that fail health checks will be removed by the ASG.

There is also a EC2 instance in the Pefoma-Public-1 subnet. This is called Pefoma Bastion and plays an important role in your infrastructure. This instance is how you connect to your internal infrastructure. The ASG instances are in a private subnet so they cannot be accessed from the outside. To allow you to connect into the private subnet you will first connect to the Pefoma Bastion instance and then use SSH key forwarding to connect into the internal network.

Please check the reference section for guidance on this.

**Elastic Load Balancer**

There is an Application Load Balancer that sits in your Public Subnets. This takes any http traffic and distributes it to your ASG EC2 instances. This is currently set at a round robin configuration, which means if 10 people visit the DNS end point, 5 would go to Private-Subnet-ec2-1 and 5 to Private-Subnet-ec2-2. However, it is random and not guaranteed to be a 50/50 split.

Also, I have lowered the ideal number of instances to 1 in the ASG, so the Load Balancer will not really help. However, when traffic increases, new instances are deployed in different subnets and this will allow traffic to be spread evenly amongst these.

The end point for the ALB is

You can create a DNS record to change this to whatever URL you require.

**Internet Gateway**

There is an Internet Gateway called Pefoma-igw. This allows your VPC to connect to the internet IF the traffic comes from the public subnet.

**Bastion Host**

There is a bastion host located at xxx. This is used to ssh into the main server.

**NAT Gateways**

There are two NAT gateways in your infrastructure. These are in the Public subnets 1 and 2. These take traffic from the private subnets and allow them to connect to the internet by going through the Internet Gateway without directly being exposed to the public subnet.

**RDS**

The main RDS server is called Pefoma-RDS-Database and located in EU West 2B. There is a read replica aptly named, ReadReplica located in EU West 2C.

**Security Groups**

The primary security groups are: Pefoma-Bastion (allows connection via ssh), Pefoma-ALG-SG, Pefoma-EC2-SG and Pefoma-RDS-SG-2b.

**Route Tables**

There are two route tables. Public and Private. Private route table allows internal communication and sends all other traffic to the NAT gateway. The Public route table allows internal communication and sends all outbound traffic (including that from the NAT gateway) to the Internet Gateway.

**How do we connect to the Bastion Host?**

The key, Pefoma-ASG-Key.pem is used to SSH to the Bastion host at ---

First we need to add and forward the SSH key.

Mitchell-MackbookPro:Downloads andrew$ ssh-add -L Pefoma-ASG-Key.pem

Mitchell-MackbookPro:Downloads andrew$ ssh -A ec2-user@xxxxx

Last login: Mon Nov 30 16:48:01 2020 from 90.255.107.92

\_\_| \_\_|\_ )

\_| ( / Amazon Linux 2 AMI

\_\_\_|\\_\_\_|\_\_\_|

https://aws.amazon.com/amazon-linux-2/

[ec2-user@ip-10-0-0-13 ~]$

We can then connect to the internal EC2 instances by using the relevant IP’s.

[ec2-user@ip-10-0-0-13 ~]$ ssh ubuntu@10.0.0.39

Last login: Mon Nov 30 17:08:13 2020 from 10.0.0.13

\_\_| \_\_|\_ )

\_| ( / Amazon Linux 2 AMI

\_\_\_|\\_\_\_|\_\_\_|

https://aws.amazon.com/amazon-linux-2/

[ec2-user@ip-10-0-0-107 ~]$

Please visit the following URL to learn more. <https://digitalcloud.training/ssh-into-ec2-in-private-subnet/>

This may also be useful: <https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/USER_ConnectToInstance.html>

**Why do we need to use a bastion host?**

A bastion host is a server whose purpose is to provide access to a private network from an external network, such as the Internet. Because of its exposure to potential attack, a bastion host must minimize the chances of penetration. For example, you can use a bastion host to mitigate the risk of allowing SSH connections from an external network to the Linux instances launched in a private subnet of your Amazon Virtual Private Cloud (VPC).

**How do I change the Auto Scaling Settings?**

Visit the following URL and click edit in the group details. <https://eu-west-2.console.aws.amazon.com/ec2autoscaling/home?region=eu-west-2#/details/Pefoma-ASG?view=details>

Graphical user interface, text, application

Description automatically generated

How do we add the application to the Ec2 instances?

The ASG uses an AMI image to create the new instances. To update this, connect to the EC2 instance and install all of your required software and assets. Once this is done go into EC2 console, choose the EC2 instance and select actions, images and templates, create image.

Graphical user interface, text, application

Description automatically generated

Then go into the ASG and choose to edit the Launch Template. Click create new launch template and follow the instructions. Use the AMI as the image.

RDS

There is no public access to RDS. The only way to connect is via an EC2 instance in the private subnet. These instances have the Mysql client installed and can connect to the RDS endpoint.

Use the bastion host to connect to an internal instance and then amend the database as required.

Backup Settings

In terms of the EC2 instances, the ASG will automatically remove and add instances as required. Since this uses an AMI to do this (and a lot of your assets are hosted externally) this is essentially the backup. If an EC2 fails two health checks, it will be terminated and replaced.

RDS – RDS is set to a 7 day backup retention policy. Amazon RDS creates and saves automated backups of your DB instance during the backup window of your DB instance. RDS creates a storage volume snapshot of your DB instance, backing up the entire DB instance and not just individual databases. RDS saves the automated backups of your DB instance according to the backup retention period that you specify. If necessary, you can recover your database to any point in time during the backup retention period.

RDS Disaster Recovery

In addition to using Read Replicas to reduce the load on your source DB instance, you can also use Read Replicas to implement a DR solution for your production DB environment. If the source DB instance fails, you can promote your Read Replica to a standalone source server.

If a disaster occurs, you can create a new DB instance by restoring from a DB snapshot. When you restore the DB instance, you choose the name of the DB snapshot from which you want to restore. Then, you provide a name for the new DB instance that is created.

Recommendations

• Enable enhanced RDS monitoring. My account did not have access to do this.

In the navigation pane, choose Roles.

Choose Create role.

Choose the AWS service tab, and then choose RDS from the list of services.

Choose RDS - Enhanced Monitoring, and then choose Next: Permissions.

• Consider a cross region approach to your infrastructure.

You currently have a highly available infrastructure that should withstand most problems. However, as all resources are located in the same Region (across three separate datacentres) if there was a Region-wide issue (unlikely) this would be problematic.

• Enable Guard Watch to monitor any naughty traffic and react accordingly.

• Adjust your infrastructure vertical scaling based on the traffic after launch.

• Implement a CI/CD pipeline.

It would be possible to add a CI/CD pipeline to the existing architecture. One option may be a Jenkins server that sits in the private subnet. This would take application code from your developer commits. Run automated testing, and if it passes schedule it for roll out.

• Create a staging/dev environment within your VPC

On a similar note, creating a smaller, DEV or staging environment would be great to implement with the CI/CD pipeline. Allowing you to thoroughly test and share your live code before moving it to production.