**Assignment - 1**

Create VPC peering between Two VPCs and migrate data between two Databases.

**Goals:**

1. VPC Peering will help you learn more about networking concepts and databases.

2. Database migration helps you how to handle large amounts of data

**Technologies Used:**

1. AWS Cloud

2. Shell Scripting

3. Debugging database issues

4. Peering VPC

5. AWS Database Service

6. Networking

**Steps:**

1. Create VPC Peering between two VPCs

2. Bastion Host [ec2 instance] needs to be created

3. Access the RDS Instance from the Bastion Machine

4. Migrate data from 10.x database to 12.x database

**Assignment - 2**

**Deploying a Static Website to AWS S3 with HTTPS using CloudFront.**

**Prerequisites.**

1. You should have a domain name already purchased to link with your Static Website.
2. Some Web content to display on your domain.

**By the end of this post, you will be able to:**

1. Create an S3 bucket and configure it for static website hosting.
2. Create a record in Route 53.
3. Create a CloudFront distribution and link it with your custom domain.
4. Create Certificates in AWS Certificate Manager.
5. Then Finally Link the CloudFront CDN, S3, Custom Domain, and SSL Certificate via Route 53 To securely access your webpage.

**I have created a simple index.html file.**

**index.html**

<html>

<body>

<h1> "Be A Better Version of Yourself".</h1>

<img src="image/image.jpg" alt="Be-a-Better-Version-of-Yourself">

</body>

</html>

**Assignment – 3**

**Auto Scaling**

* Auto Scaling helps you ensure that you have the correct number of EC2 instances available to handle the load for your application.
* You create collections of EC2 instances, called Auto Scaling groups.
* You can specify the minimum number of instances and the maximum number of instances in the Auto Scaling group, and Amazon EC2 Auto Scaling ensures that your group never goes above this size.

**steps involved to complete this task:**

1. Create a load balancer.
2. Create Launch configuration.
3. Create an Autoscaling group.
4. Create a Topic in SNS
5. Create an Alarm in CloudWatch
6. Add Policy in Auto Scaling

**Assignment - 4**

**IAM – It’s related to administration.**

Every company will have only one AWS Account. The owner of the account can create user accounts. We can give limited privileges to the user accounts.

**The task needs to do.**

1. User A ---> Should have EC2 full access.
2. User B ---> Should have S3 full access.
3. User C ---> Should have EC2 read only access.

**Assignment - 5**

**CloudWatch**

1. Create two EC2 machines.

Name it as --> i) Machine 1

ii) Machine 2

1. The monitoring tab, helps you get the CloudWatch metrics like

* CPU Utilization
* Disk read (Bytes)
* Disk write (Bytes)
* Network packets in
* Network packets out etc

1. These metrics will be updated every 5 min.

**Assignment - 6**

**Create the VPC using the below requirement and deploy the web server on the public subnet.**

1. Create --> VPC ( MyVPC ) - 10.0.0.0/16
2. Create two subnets --> subnet1 - 10.0.1.0/24, subnet2 - 10.0.2.0/24
3. Enable public IP to subnet1.
4. Create Internet Gateway attach to VPC -- MyIGW
5. Create Route table -- InternetRT
6. Attach the Route table to subnet1.
7. Attach Route table to Internet Gateway

Now, subnet1 is public.

Now, Let’s launch the webserver in the public subnet.

Additional Details -- User Data

#!/bin/bash

sudo su

yum update -y

yum install httpd -y

cd /var/www/html

echo "MyGoogle" > index.html

service httpd start

chkconfig httpd on

**Assignment - 7**

**Create the VPC using the below requirement and deploy the DB server in private, Webserver in the public subnet.**

VPC - Select the Mumbai region.

1. Create VPC ( MyVPC ) - 10.0.0.0/16
2. Create two subnets. --> subnet1 - 10.0.1.0/24 – WebSN, subnet1 - 10.0.2.0/24 --DbSN
3. Enable public IP to subnet1.
4. Create Internet Gateway attach to VPC -- MyIGW
5. Create Route table - InternetRT
6. Attach the Route table to subnet1.
7. Attach Route table to Internet Gateway--> Now, subnet1 is public.
8. Create NAT in the public subnet. --> So that we can have the internet to a private subnet

**Steps:**

* Let’s launch the Ec2 machine in a public subnet. --> Name: Webserver
* Create one database server in the private subnet. --> Name Tag: DbServer
* Creating Bastion/ Jump server in the public subnet.