**Lab 1 – IAM**

Create some IAM users  as per requirement.

If you login with your email id and password then it is your root account. And it is not recommended to use root account for performing daily tasks. There are your team members who are going to work on AWS along with you. And they need access to particular services on AWS platform.

Users who are going to work on project are like

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Emp. No. | Name | Role | Authorities | Emergency Contact no |
| 0001 | Raman | Project Admin | Full Admin Access | 1234567890 |
| 0002 | Natasha | DB admin | RDS full access | 1234567890 |
| 0003 | Navin | Storage admin | S3 bucket full access | 1234567890 |
| 0004 | Bhavesh | Storage admin | S3 bucket full access | 1234567890 |
| 0005 | Kamal | Accountant | Billing Access | 1234567890 |
| 0006 | Roni | Accountant | Billing Access | 1234567890 |
| 0007 | Rohan | Storage Security | Read Access for s3 | 1234567890 |

Assign proper tags for Identification of user. Activate MFA for Root account. Create strong password policy. Create alias for IAM login url.

**Lab 2 – VPC – Virtual Private Cloud**

You have users who are going to work on your project. Next task is to create an isolated network for your project. That thing can be done with VPC service.

In three tier architecture there is one web server and one database server. As per the requirement we need 2 networks in a single VPC. create 2 different subets

1. Webserver – 10.0.1.0/24
2. Database – 10.0..2.0/24

Subnet where you are going to configure your web server will be connected to the internet and another where you have **DB server should not connected to the internet**. You will also need a bastion server to connect your private VM’s.

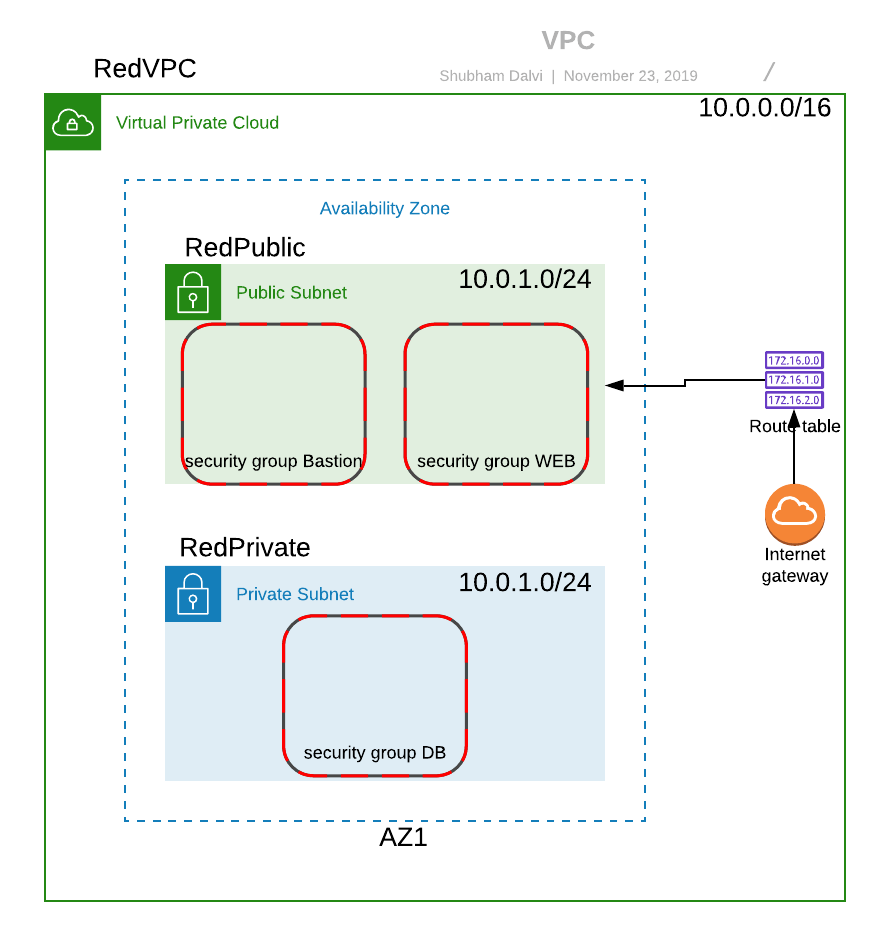
You will need an Internet Gateway to communicate outside your VPC, also a route table which will route traffic outside the VPC using Internet Gateway.

Create 03 security group for

1. web Server,
2. DB Server and
3. Bastion Host.

And manage inbound traffic for security.

Finally your architecture look like below.

[](https://i1.wp.com/itstechschool.com/wp-content/uploads/2019/12/Lab-2-VPC-Virtual-Private-Cloud.png?ssl=1)

**Lab 3 – EC2**

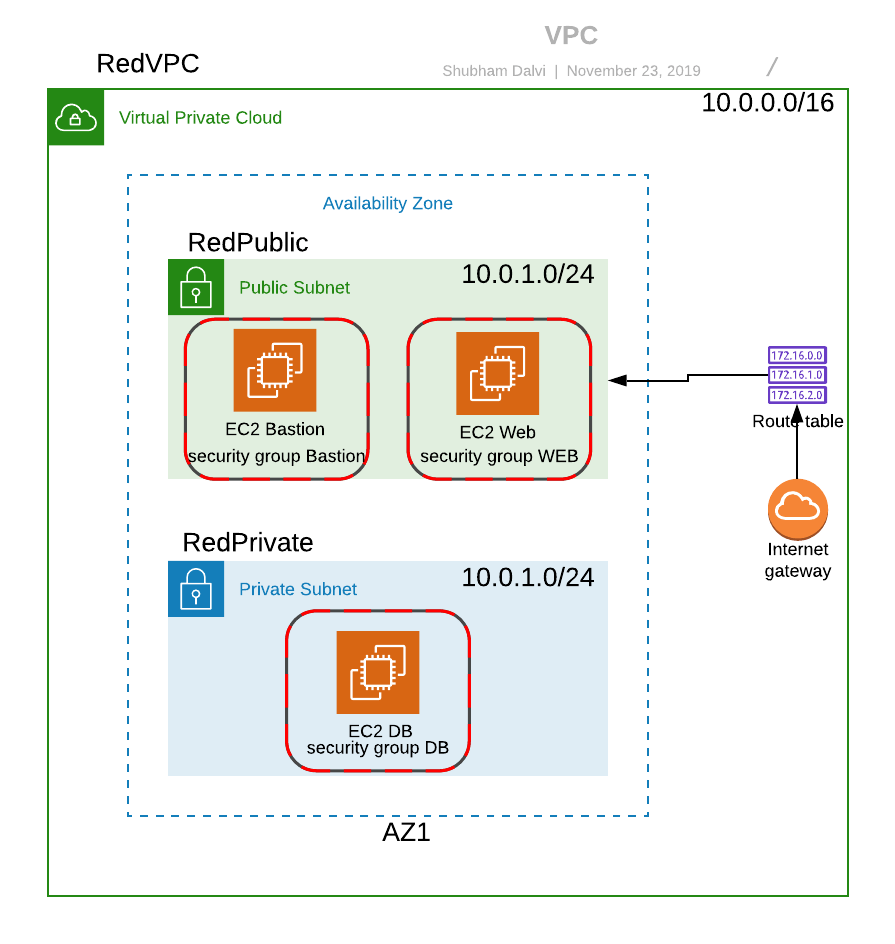
Create 3 VM

            1 Windows vm which works like a bastion host.

1. Linux VM for web server.
2. Linux VM for DB server.

Try to connect each machine one by one. And also check network traffic flow.

            Architecture will be like.

[](https://i1.wp.com/itstechschool.com/wp-content/uploads/2019/12/Lab-3-EC2.png?ssl=1)

**Lab 4- NAT**

Connect your DB vm and try to install mysql packages. Internet is not available for downloading packages that’s why you will need nat gateway in scenario.

1. Create a Nat Gateway
2. Create a route table
3. Associate route table with private subnet.
4. Add an entry for nat routing.

[Diagram

Description automatically generated](https://i0.wp.com/itstechschool.com/wp-content/uploads/2019/12/Lab-4-NAT.png?ssl=1)

**Lab 5 – VPC Peering**

Customer already have a application hosted on one of the vpc on aws. Customer wants to connect this two VPC with each other. They should communicate to each other with private network connection.

1. Create a VPC peering from one VPC
2. Accept connection request from another VPC
3. Add entry in route table

[Diagram

Description automatically generated](https://i2.wp.com/itstechschool.com/wp-content/uploads/2019/12/Lab-5-VPC-Peering.png?ssl=1)

**Lab 6 – High Availability**

For High availability it is important to host application on more than one data centers i.e. Availability Zones.

1. Create two web server in two different AZ.

     2. Create an application load balancer to balance a load between two web servers.

[A picture containing graphical user interface

Description automatically generated](https://i0.wp.com/itstechschool.com/wp-content/uploads/2019/12/Lab-6-High-Availability.png?ssl=1)

**Lab 7 – Scalability**

Configure your architecture for scaling out and scaling in web servers whenever CPU utilization of web server increases or decreases. Configure a load balancer to receive traffic from a load balancer.

As per customer’s requirement minimum, 2 web servers should be available on two different AZ. Restrict no VMS to extend more than 4.

Use CPU utilization as a matrix and 60% as a threshold value.

1. Create a launch configuration for web servers.
2. Create an auto-scaling group using a launch configuration.
3. Balance traffic on auto-scaling group using a load balancer.

[A picture containing box and whisker chart

Description automatically generated](https://i1.wp.com/itstechschool.com/wp-content/uploads/2019/12/Lab-7-Scalability.png?ssl=1)

**Lab 8 – Add-on Exercise.**

Application load balancers already have a public endpoint. Traffic coming for HTTP on web servers must be from a load balancer only. Remove public IP from the load balancer and make them private for improving security.

Configure traffic flow from the security group from as follows.

1. Create separate Security Group for the Web server, DB server, Bastion and Load Balancer.
2. DB instances allow 3306, 22 traffic only from web server Security Group and Bastion security Group respectively.
3. Web server allow 80, 22 port traffic only from Load balancer Security group and Bastion host security group respectively.
4. Make 80 port open for internet on load balancer security group.
5. Make 22 port open for internet on bastion security group.

[Graphical user interface, application, PowerPoint

Description automatically generated](https://i0.wp.com/itstechschool.com/wp-content/uploads/2019/12/Lab-8-Add-on-Exercise..png?ssl=1)

**Lab 9 – DNS**

            Purchase a domain name and create an alias between domain name and load balancer endpoint.

1. Purchase a domain name
2. Create a route 53 hosted zone and add an alias for redirecting traffic to load balancer.

[Graphical user interface, application

Description automatically generated](https://i2.wp.com/itstechschool.com/wp-content/uploads/2019/12/Lab-8-DNS.png?ssl=1)

**Lab 10 – Simple Storage Service S3**

1. Create an s3 bucket
2. Make versioning on
3. Host a static website on S3
4. Make cross-region replication on
5. Create a life cycle policy to move data from standard class IA after 30 days, IA to the glacier after 90 days, and expire after 360 days.

**Lab 11 – Elastic Block Storage**

1. Add an extra local EBS volume of 10 GB for storing application data.
2. Create a manual snapshot of EBS.
3. Add a life cycle rule to create an automatic snapshot.
4. Detach and Delete EBS.
5. Remove Lifecycle rule

**Lab 12 – Elastic File System**

The customer needs network storage for storing data of multiple web servers on a central location.

1. Create an EFS storage.
2. Mount EFS to linux EC2 machine.

**Lab 13 – Elastic File System**

1. Create a RDS instance with MySQL engine.
2. Login mysql database on linux EC2 instance.

**Lab 14 – Elastic Beanstalk**

1. Create an elastic beanstalk with python platform.
2. Upload your code.
3. Check endpoint and verify hosting application.