**First Scenario:**

* **Step One:**

We need to create a Database By using RDS Service And this is the Info for DB:

* DB Name: **databaseIb**
* Username: **admin**
* Password: **tareq482**
* VPC: **Example VPC**
* Security Group: **Example-DB**
* **Step Two:**

We need to check all Security Groups and the inbound rule for each one

So, there is a four Security Groups ...

1. Bastion-SG with SSH Type, TCP Protocol and Port Range 22.
2. Example-DB with MYSQL/Aurora, TCP Protocol and Port Range 3306.
3. Inventory-App with HTTP type, TCP Protocol and Port Range 80 & HTTPS type, TCP Protocol and Port Range 443, Then we will add new Inbound rule with SSH Type, TCP Protocol and 22 Port with Bastion-SG source.
4. ALBSG With HTTP type, TCP Protocol and Port Range 80 & HTTPS type, TCP Protocol and Port Range 443.

* **Step Three:**

We need to create a new key pair to use it for Pageant and Putty to connect with EC2 Instances through SSH, with vockey2 Name, we need to download it and we need to download the PPK file for labuser.

* **Step Four:**

We need to create a new private instance to secure the website, so we will use this info:

* AMI – Amazon Linux 2 AMI.
* Instance type- t2. micro.
* Name: Web server.
* VPC – Example VPC.
* Subnet – Private Subnet 1.
* Auto- assign Public Ip – Disable.
* IAM role – Inventory-App-Role.
* Name – ExampleApp.
* We select an existing security Group – after that select the “Inventor App” Security group.
* Select an Existing key – vockey2.
* **Step Five:**

Now we will go the System Manger service and from the Parameter Store will create four new Parameters:

1. **Name:** /example/endpoint **& Value:**  The endpoint of the RDS DB.
2. **Name:** /example/username **& Value:** admin.
3. **Name:** /example/password **& Value:** tareq482.
4. **Name: /**example/database & **Value**: databaseIb.

* **Step Six:**

We need to access to the Web Server Instance via Bastion-Sg Instance so we cannot do this because there is not any NAT Gateway to connect with the private instance so:

1. We will create NAT Gateway and when do this when have to choose the Allocate Elastic IP Address, Change the VPC to Example-DB and choose the Public 1 Subnet.
2. We will Edit the routes and add the 0.0.0.0/0 Destinationin Private Subnets Route Tables and choose the NAT that’s we created earlier like a target.

* **Step Seven:**
* Now we will Access to Web Server instance through Bastion-SG Instance by using Pageant & Putty:
* In Pageant we used the Vockey2 that we created before.
* And in Putty we will go to the session tab and enter the Public IPv4 address for the Bastion-SG Instance, then we will change the Seconds between keepalives from 0 to 30 from connection tab and from the Auth tab in SSH tab we choose the Allow agent forwarding button and browse the labuser that’s we downloaded.
* Now we will log in like **ec2-user** (after the above command you will see like this-> ec2-use@ip-<private IPv4 address of Bastion instance>)
* In command line we will use the follow command:
* SSH ec2-user@ <private IPv4 address of Web Server instance>.
* **Step Eight:**

Now we need to Install the Apache Web Server (httpd) and MySQL on Instance ExampleApp, on the command line enter the following command in order:

1. ping [www.google.com](http://www.google.com) to test the connection, click ctrl+c to stop it.
2. sudo su
3. yum install -y httpd MySQL
4. amazon-linux-extras install -y php7.2

* **Step Nine:**

Download the SQL dump file that contains sample data by this command:

wget <https://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/ILT-TF-200-ACACAD-20-EN/capstone-project/Countrydatadump.sql>

* **Step Ten:**

Download source code Web App by this command:

wget <https://aws-tc-largeobjects.s3.us-west-2.amazonaws.com/ILT-TF-200-ACACAD-20-EN/capstone-project/Example.zip>

* **Step Eleven:**

extract source Web App and copy in directory /var/www/html, use these commands in order:

* 1. unzip Example.zip -d /var/www/html/
  2. ls /var/www/html
* **Step Twelve:**

Enable and start service httpd, by using these commands in order:

* 1. systemctl enable httpd
  2. systemctl start httpd
  3. systemctl status httpd
* **Step Thirteen:**

Login to your database by using the following commands in order:

* 1. mysql -u admin -p --host <The Endpoint of RDS DB>
  2. Enter the password - tareq482
  3. show databases;
  4. exit;
* **Step Fourteen:**

Import data the database by using the following commands in order:

1. mysql -u admin -p --host < The Endpoint of RDS DB > databaseIb <Countrydatadump.sql.
2. Enter the password - tareq482
3. exit
4. exit

* **Step Fifteen:**

Now we need create a Target Group for Load Balancer that’s we will create forward, so now this is the info for the Target Group:

1. VPC- Example VPC
2. advanced health check setting:

* Healthy threshold – 2
* Interval – 10
* **Step Sixteen:**

Now we are ready to create Load Balancer, and this is the right info:

* Select Application Load balancer

1. Name - LB-Example
2. VPC- Example VPC
3. Select first region - us-east-1a
4. Select subnet - public subnet 1
5. Select second region - us-east-1b
6. Select subnet - public subnet 2
7. Change the security groups to:

* Select – ALBSG

1. Listeners and routing:

* Select - TG-Example

1. Click on create load balancer

* **Step Seventeen:**

In this step we will create Auto Scaling Group:

1. Name- ASG-Example
2. on lunch template:

* select - Example-LT

1. Network -VPC - Example VPC
2. Subnets:

* Select private subnet 1
* Select private subnet 2

1. Load balancing:

* click on attach an existing load balancer.
* select - TG-Example
* **Step Eighteen:**

Go to your Instances tab there are 2 more instances are launching wait for 2 to 3 min. to pass the checks.

* **Step Nineteen:**

Test the Web Application by using load balancer DNS in new tab by following these steps:

* + 1. Go to Load balancer you created earlier.
    2. Copy the DNS address of the load balancer and paste it in a new tab.