# Question 1

## Step 1 - Creating Instance

- Choose AMI - Amazon Linux AMI 2018.03.0 (HVM), SSD Volume Type
- Choose Instance Type - t2 micro
- Configure Instance Details - fill in the additional details to auto install apache in your instance
    - Use the script -

      ```
      #!/bin/bash
      yum update -y
      amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2
      yum install -y httpd mariadb-server
      systemctl start httpd
      systemctl enable httpd
      usermod -a -G apache ec2-user
      chown -R ec2-user:apache /var/www
      chmod 2775 /var/www
      find /var/www -type d -exec chmod 2775 {} \;
      find /var/www -type f -exec chmod 0664 {} \;
      echo "<h1>Hello World!</h1>" > /var/www/html/hello.html
      ```
- Add Storage
- Add Tags
- Configure Security Group - Using existing
- Review and Launch
- Select type of Key for that instance

## Step 2 - Deploy file on server

- Connect to your instance using - ssh -i "krutik-assessment.pem" ec2-user@ec2-3-94-101-54.compute-1.amazonaws.com
- Elevate privileges - sudo su
- Start apache service - service httpd start
- Navigate - cd /var/www/html
- Add index.html - nano index.html

## Step 3 - Create Auto Scaling Group From Instance

- From actions select - Attach Auto Scaling Group
- Use an existing or Create a new Auto Scaling Group by providing the name

## Step 4 - Create Load Balancer

- Now Create a new Load Balancer - Application Load Balancer
- Configure Load Balancer

- Configure Security Settings
- Configure Security Groups
- Configure Routing
- Register Targets
- Review and Launch

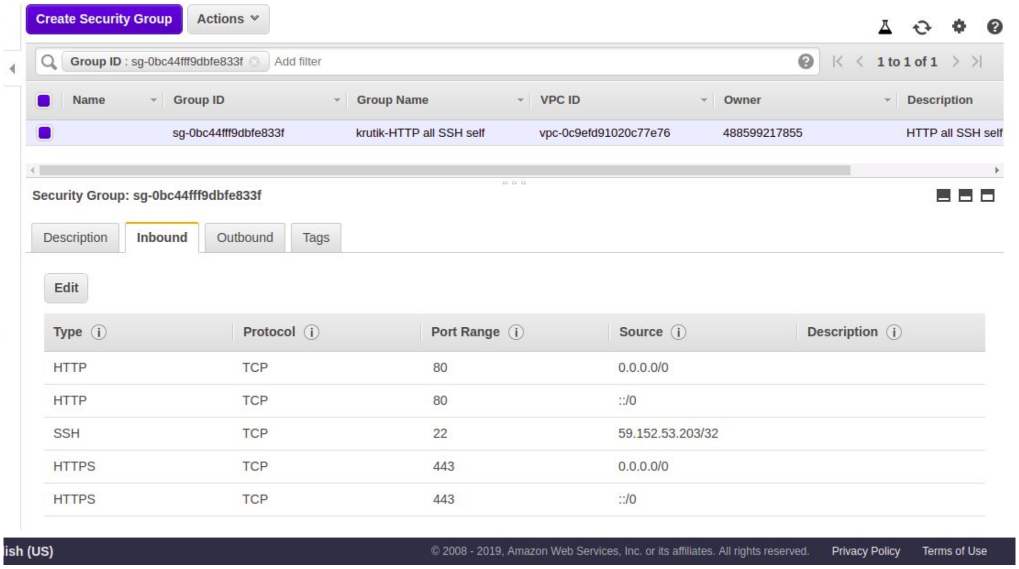
## Step 5 - Assign Target in Auto Scaling Group

- Now add the target (load balancer) in the Auto Scaling Group to which the instance is attached.

## SCRIPT

```
#!/bin/bash
yum update -y
amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2
yum install -y httpd mariadb-server
systemctl start httpd
systemctl enable httpd
usermod -a -G apache ec2-user
chown -R ec2-user:apache /var/www
chmod 2775 /var/www
find /var/www -type d -exec chmod 2775 {} \;
find /var/www -type f -exec chmod 0664 {} \;
echo "<h1>Hello World!</h1>" > /var/www/html/hello.html
```

## Security Group Rules



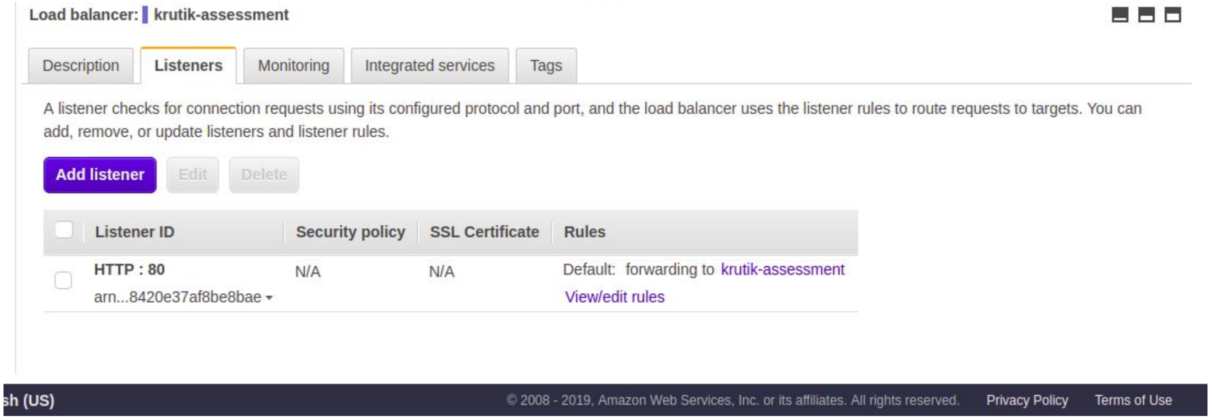
Group ID : sg-0bc44fff9dbfe833f

| Name | Group ID | Group Name | VPC ID | Owner | Description |
|---|---|---|---|---|---|
| | sg-0bc44fff9dbfe833f | krutik-HTTP all SSH self | vpc-0c9efd91020c77e76 | 488599217855 | HTTP all SSH self |

Security Group: sg-0bc44fff9dbfe833f

| Type | Protocol | Port Range | Source | Description |
|---|---|---|---|---|
| HTTP | TCP | 80 | 0.0.0.0/0 | |
| HTTP | TCP | 80 | ::/0 | |
| SSH | TCP | 22 | 59.152.53.203/32 | |
| HTTPS | TCP | 443 | 0.0.0.0/0 | |
| HTTPS | TCP | 443 | ::/0 | |

# Listener Configuration



**Load balancer:** **krutik-assessment**

A listener checks for connection requests using its configured protocol and port, and the load balancer uses the listener rules to route requests to targets. You can add, remove, or update listeners and listener rules.

| ☐ | Listener ID | Security policy | SSL Certificate | Rules |
|---|---|---|---|---|
| ☐ | **HTTP : 80**<br>arn...8420e37af8be8bae ▾ | N/A | N/A | Default: forwarding to krutik-assessment<br>View/edit rules |