ASSIGNMENT-1-AWS

1. Create an EC2 instance in AWS:

1. Log in to your AWS console and navigate to the EC2 service.
2. Click on the "Launch Instance" button to begin the instance creation process.
3. Choose an Amazon Machine Image (AMI) to use for your instance. You can select one that comes pre-installed with screens, such as Ubuntu or Amazon Linux 2.
4. Select the instance type that you want to use, depending on your needs.
5. Configure your instance details, including the number of instances to launch, the VPC, the subnet, and any advanced settings you may need.
6. Add any storage devices that you need for your instance, such as an EBS volume.
7. Configure your instance security group to allow inbound traffic on port 22, which is used for SSH connections.
8. Review and launch your instance, and then connect to it using an SSH client such as PuTTY.
9. Once you have connected to your instance, you can install screens by running the following command:

arduino

Copy code

```
sudo apt-get install screen
```

Screens should now be installed and ready to use on your EC2 instance. You can create a new screen session by running the command:

Copy code

```
screen
```

And you can detach from a screen session by pressing the keys `CTRL+A`, followed by `d`. To re-attach to a screen session, use the command

Copy code

```
screen -r
```

That's it! You should now be able to use screens on your EC2 instance.

2.To Create Elastic Block Store:

Log in to your AWS console and navigate to the EC2 service.

1. Click on the "Volumes" link in the left-hand menu to access the EBS volumes dashboard.
2. Click on the "Create Volume" button to begin the volume creation process.
3. Specify the size of the volume, the availability zone, and any other advanced settings you need.
4. Click on the "Create" button to create the volume.
5. Once the volume has been created, you will need to attach it to an EC2 instance. To do this, select the volume in the dashboard and click on the "Actions" button, then choose "Attach Volume".
6. Choose the EC2 instance to which you want to attach the volume, and specify the device name for the volume.
7. Click on the "Attach" button to attach the volume to the instance.
8. Connect to the instance using an SSH client such as PuTTY.
9. Once you have connected to the instance, use the following command to format the volume:

bash

Copy code

```
sudo mkfs -t ext4 /dev/xvdf
```

Make sure to replace `/dev/xvdf` with the device name you specified when attaching the volume.

11. Create a mount point for the volume using the following command:

bash

Copy code

```
sudo mkdir /mnt/ebsvolume
```

12.Mount the volume to the mount point using the following command:

bash

Copy code

```
sudo mkdir /mnt/ebsvolume
```

13.Install screens by running the following command:

arduino

Copy code

```
sudo apt-get install screen
```

Screens should now be installed and ready to use on your EC2 instance, and you can use them with the EBS volume that you created.

3.Snap Screenshot Creation:

1. Connect to your EC2 instance using an SSH client such as PuTTY.
2. Use the following command to create a new screen session:

Copy code

```
Screen
```

3.Once you are in the screen session, run any commands or perform any actions that you want to include in the snapshot.

3. Detach from the screen session by pressing the keys `CTRL+A`, followed by `d`.

4. Use the following command to list the available screens:

bash

Copy code

```
screen -ls
```

This will display a list of screens, along with their IDs.

6. Take note of the ID of the screen session that you want to include in the snapshot.

7. Use the following command to take a snapshot of the instance:

Arduino

Copy code

```
aws ec2 create-image --instance-id <your-instance-id> --name <your-image-name> --description
"<your-image-description>" --block-device-mappings "DeviceName=/dev/xvda,Ebs={DeleteOnTermination=true}"
```

```
               "<your-image-description>"
"DeviceName=/dev/xvda,Ebs={DeleteOnTermination=true}"
```

Make sure to replace `<your-instance-id>` with the ID of your EC2 instance, `<your-image-name>` with the name you want to give the image, and `<your-image-description>` with a brief description of the image.

8. Wait for the snapshot creation process to complete. This may take several minutes, depending on the size of your instance and the amount of data to be included in the snapshot.

9. Once the snapshot is complete, you can view it in the EC2 dashboard under the "AMIs" section.

That's it! You have now taken a snapshot of your EC2 instance while using screens.

4.AMI Creation:AMAZON MECHINE IMAGE

Connect to your EC2 instance using an SSH client such as PuTTY.

1. Use the following command to create a new screen session:

Copy code

```
screen
```

3. Once you are in the screen session, run any commands or perform any actions that you want to include in the AMI.
4. Detach from the screen session by pressing the keys `CTRL+A`, followed by `d`.
5. Use the following command to list the available screens:

bash

Copy code

```
screen -ls
```

This will display a list of screens, along with their IDs.

6. Take note of the ID of the screen session that you want to include in the AMI.

7. Use the following command to create an AMI of the instance:

css

Copy code

```
aws ec2 create-image --instance-id <your-instance-id> --name <your-ami-name> --description
"<your-ami-description>" --no-reboot
```

`--instance-id` `--name`

`--description` `--no-reboot`

Make sure to replace `<your-instance-id>` with the ID of your EC2 instance, `<your-ami-name>` with the name you want to give the AMI, and `<your-ami-description>` with a brief description of the AMI.

8. Wait for the AMI creation process to complete. This may take several minutes, depending on the size of your instance and the amount of data to be included in the AMI.

9. Once the AMI is complete, you can view it in the EC2 dashboard under the "AMIs" section.

That's it! You have now created an AMI of your EC2 instance while using screens.

5.Load Balancer Creation:

1. Connect to your EC2 instance using an SSH client such as PuTTY.
2. Use the following command to create a new screen session:

Copy code

```
screen
```

3. Once you are in the screen session, run the following command to create a new load balancer:

css

Copy code

```
aws elbv2 create-load-balancer --name <your-lb-name> --subnets <your-subnet-ids> --security-groups <your-security-group-ids> --type application --scheme internet-facing
```

`--name` `--subnets`

`--security-groups` `--type` `--scheme`

Make sure to replace `<your-lb-name>` with the name you want to give the load balancer, `<your-subnet-ids>` with a comma-separated list of the subnet IDs where you want to deploy the load balancer, and `<your-security-group-ids>` with a comma-separated list of the security group IDs that you want to attach to the load balancer.

4. Wait for the load balancer creation process to complete. This may take several minutes, depending on the size of your instance and the amount of data to be included in the AMI.

5. Once the load balancer is created, you can create a target group by running the following command:

css

Copy code

```
aws elbv2 create-target-group --name <your-target-group-name> --protocol HTTP --port 80 --vpc-id <your-vpc-id>
```

`--name` `--protocol`

`--port 80 --vpc-id`

Make sure to replace `<your-target-group-name>` with the name you want to give the target group, `<your-vpc-id>` with the ID of the VPC where you want to deploy the target group, and specify the protocol and port that you want to use for your target group.

6. Once the target group is created, you can register your EC2 instances with the target group using the following command:

Csharp

```
aws elbv2 register-targets --target-group-arn <your-target-group-arn> --targets
Id=<your-instance-id>,Port=80
```

Copy code

```
aws elbv2 create-listener --load-balancer-arn <your-lb-arn> --protocol HTTP --port 80 --default-actions
Type=forward,TargetGroupArn=<your-target-group-arn>
```

`group` `group`

`80`

Make sure to replace `<your-target-group-arn>` with the ARN of the target group you created in step 5, and `<your-instance-id>` with the ID of the EC2 instance you want to register with the target group.

7. Once your instances are registered, you can create a listener using the following command:

css

Copy code

```
aws elbv2 create-listener --load-balancer-arn <your-lb-arn> --protocol HTTP --port 80 --default-actions
Type=forward,TargetGroupArn=<your-target-group-arn>
```

`--load-balancer-arn` `--protocol` `--port`
`80 --default-actions`

Make sure to replace `<your-lb-arn>` with the ARN of the load balancer you created in step 3, and `<your-target-group-arn>` with the ARN of the target group you created in step 5.

8. Wait for the listener creation process to complete. This may take several minutes.

9. Once the listener is created, you can test your load balancer by accessing it using the DNS name associated with it in the EC2 dashboard.

That's it! You have now created a load balancer in AWS while using screens.

END OF ASSIGNMENT