

# **Mini Project – 2**

---

## **Amazon Web Services**

# LAB-1 IAM Hands-On



Setup the MFA for root user.



Create a new user with console access and check its default permissions.



Assign Only EC2 full permissions to this user and navigate to different services with this user account, check if you can work with any other service except EC2.



Provide the Administrative privileges to this user.

## *LAB-2*

# Billing Alarm

Setup the billing Alarm for your account to get a notification whenever you cross the billing threshold.

## LAB-3

# S3 Bucket

- Create an S3 bucket, make sure to give it a unique name.
- Upload Some test files/folder in the bucket.
- Try to access the file over the browser using its URL. It should give error.
- Check the permissions of the file and the bucket.
- Make the object public and access it again over the browser.

### STEP 1:

- Enable versioning of the bucket.
- Create a text file and upload it in the bucket.
- Update the text file and upload the updated version to the bucket.
- Check the content of current file and the previous version of text file.

### STEP 2:

- Delete the text file from the bucket.
- Check and recover the deleted file from the versioning

## *LAB-4*

# EC2 Instance

- Launch one EC2 instance of type t2.micro with ubuntu OS (free tier eligible).
- Allow the required port in the Security Groups so that you can access it from outside.
- Access this instance from your local machine using putty/ other ssh software's.

## *LAB-5*

# Security Group

- Create a new security group and name it as “mynewSG”
- Check the default rules in this security group.
- Allow inbound port 80 and 22 from your IP address. Select a range (IP/28) so that any changes in your dynamic IP will not impact the rule.
- Attach this new SG to your existing EC2 Instance and try to access the server now (SSH).



# LAB-6

## Volumes and Snapshots

- Create one 5GB volume and attach it with the running EC2 Instance.
- Make sure this volume is available to use in Instance, take a reference from below link.
- Put some data in this volume like some testing files.
- Increase the size of this volume to 8GB and check it in the Instance.
- Extend the size of this volume inside the Linux machine.
- Take a snapshot of this volume and delete it.
- Create a new volume with the snapshot and attach it to the server just like the step 1.

### *Reference:*

1. <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ebs-using-volumes.html>
2. <https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/recognize-expanded-volume-linux.html#extend-file-system>

## *LAB-7* AMIs

- Create an AMI of your Running Instance



# *LAB-8*

## Load Balancers

- Create two EC2 instances and Install nginx server on one machine and apache server on server2.
- Access both the servers over browser and check if their web page is visible.
- Create one load balancer (application/Classic) and attach both the instances with the load balancer.
- Allow only port 80 in LB security group and also make the SG of your instances to receive request from load balancer only on port 80.
- Access the load balancer link over the browser and hit it a couple of times. Check if both the webpages (nginx/apache are visible alternatively)

## *LAB-9* ASG and LC

- Create one launch configuration with ubuntu server.
- Create an auto scaling group and attach the above created launch configuration with it
- Keep the size of instances as Min=1 Max=3
- Try to change the max capacity and see the new instance should get created.

# *LAB-10*

## RDS

- Provision an RDS instance.
- Open MySQL Port in the connected SG.
- Access this RDS from your EC2 instance.