GIT- VERSION CONTROL SYSTEM (VCS) BY USING AMAZON WEB SERVICES

LAB-1

* Setup the MFA for root user
* create a new user with console access and check its default permissions
* assign only ec2 full permission to this user and navigate to different services with this user account check if you can wok with only other service except EC2.
* Provide the administrative privileges to this user.

LAB-2

BILLING ALARM:

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

LAB-3

S3 BUCKETS

STEP-1:

* Create an s3 bucket, make sure to give it a unique name.
* Upload same test files/folder in the bucket.
* Try to access the file over the browser using its URL.it should give error.
* Check the permission of the file and the bucket.
* Make the object public and access it again over the browser.

STEP-2

* Enable versioning of the bucket.
* create a text file and the upload in the bucket.
* update the test file and upload the update version to the bucket.
* check the content of current filed and the pervious of test file.

STEP-3:

* delete the test file from the bucket.
* check and recover the deleted file from the versioning.

LAB-4

* Launch one ec2 instance type of t2.micro with ubuntu OS(free tire 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 softwares.

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:

* 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 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.

LAB-7

AMIs:

* Create an AMIs of your running instance.

LAB-8

LOAD BALANCE:

* Create two EC2 instance and install nginx server on one machine and apache server on server2.
* Access both the server 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(ngnix/apache are visible alternatively)

LAB-9

ASG AND LT:

* Create one launch template with ubuntu server.
* Create an auto scaling group and attach the above created launch template with it.
* Keep the size of instance 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.

LAB-1:(CREATING EC2 INSTANCE)

1.Login to AWS console.

2. Create a server with Amazon Linux 2 AMI/RHEL/Ubuntu.

3. Connect the server only with putti/gitbash.

LAB-2(CREATE REPO IN LOCAL MACHINE)

1. Create a folder on my local machine.
2. Initialize this folder using git init command

git init <folder name>

1. Go inside this folder and run git status to check the status
2. Created some empty file using touch command.

Touch test

1. Again ,run git status to see the changes, you can notice that file is available but not tracked by git.
2. Run git add <filename> to stage this change (git will start tracking this file) you may check it using git status once again.
3. Now commit our change by running git commit -m “address test file”
4. Run git status once again and it will show you that working tree is clean.