AWS MINOR PROJECT

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 permissions t this user and navigate to diff services with this user acc. Check if u can work with any other services except ec2.
* Provide the admin privileges to this user.

LAM-2 : billing alarm

* Setup the billing alarm for your acc to get a notification whenever you cross the billing threshold.

LAB-3 : S3 bucket

Step 1

* Create an s3 bucket make sure to give it a unique name
* Upload some test files/folder in the bucket
* 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 2

* Enable versioning of the object
* Create a text file and upload the updated version to the bucket
* Check the content of the current file and the previous version of the txt file

Step 3

* Delete the text file from the bucket
* Check the browser the deleted file from the versioning.

Lab 4 : EC2 INSTANCE

* Launch the ec-2 instance of the type t2.micro with ubuntu os(free tier)
* Allow the required port in the security grps so that you can access it from outside
* Access this instance from your machine using putty/other ssh softwares.

Lab 5 : SEQURITY GROUPS

* Create a new SG and name it as “mynewsg”
* Check the default rules in this SG
* 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 no(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 the this volume it 8gb and check it in the instance
* Extend the size of this of this volume inside the linux machine
* Take a screen shot of this volume and delete it
* Create a new volume with the snapshot and attach it to the server just like the step1.

LAB 7 : AMI s

* Create an AMI of your running instance.

LAB 8 : LAOD BALANCERS

* Create two ec2 instaces and install nginx server on one machine and apache on server 2
* Access both the severs over browser and check if their web pageis visible
* Create one load balancers(app/classics) and attach both the instances with the load balancers
* Allow only port 80 in LB security groups and also make the SG of your instances to receive request from LB only on port 80
* Access the load balncers link over the browser and hit it a couple of times. Check if the webpages (nginx/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 as
* 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.

GIT VERSION CONTROL SYSTEM BY USING AMAZON WEB SERVER

LAB 1 : (CREATING EC2 INSTANCE)

Login to aws console

Create a server with amazon linux

Connect the server only with putty/gitbash

LAB2 (create repo in the local machine)

Create a folder on my local machine

Initialize this folder using git init command

a🡪git init <folder name>

go inside this folder and run git status command to check the status

created some empty files using touch command

a 🡪 touch test

again, run git status to see the changes you can notice that the file is available but not tracked by git

run git add<filename> to stage this change (git will tracking this file) you may check it using git status once again

now commit our changes by running git commit - “added test files”

run git status once again and it will show you that the working tree is clean.

LAB 3 : CREATEING REPO IN REMOTE LOCATION – GITHUB

In remote location – git hub

Create a new repo by clicking on new button

a . provide repo name

b. select whether it is provide or public repo

c. initialilize the repo by adding A README.md file

click on the create repo and done

LAB 4 working with remote REPO

Take the remote repo to your local

Machine make the changes

Push the changes to remote

1. Pick the clone url of the repo from the github repo
2. Go to the local machine and clone this repo using git clone command
3. Git clone <repo url that you will get by clicking on code button in github
4. you will be asked a provide username and password
5. here you will get one error for password –based authentication depreciation

5. now we need to create the personal access token to work with this repo

a. At the to right corner click on the user icon

b . go to the settings

1. Devaloper settings
2. PAT
3. Click on the generate token
4. Provide a note
5. Expirartion date and scope
6. Click on the generate token
7. Copy this token and keep it safe

Clone the repo again in your local machine and this time provide the token in the place of password

Once cloned go to the repo folder and add some samples files we can us touch command to create empty files

1. Touch f1 f2

Stage these changes by running git add f1 f2

Commit these changes by running git commit –m <any changes>

Git push 🡪it will ask you for username and password provide thr username and PAT in the place of password which we created

Go to the remote repo and see you will be able to find your new changes

LAB 5 PUSHING A LOCALLY CREATED REPO TO GITHUB

Create on repo in our local machine and inialize it locally

If not done then do lab 2

Create one remote repo with the same name as local repo in github and do not inialize it

Come to your local machine and run the following commands from inside your local repo

1. Git branch –M(to change the
2. Git remote add origin <url>
3. GIT PUSH –u origin main

LAB 6 (CREATING A NEW BRANCH FROM YOUR MAIN BRANCH)

Go to the repo at the place of main and click on the branch dropdown

Type the name of your new branch that you want to create and click on the create button

A new branch will created for u

Make some changes in this branch directly from the console

Here you will see that your changes are only applied to your new branch but not to the main branch

LAB 7 PULL ALL THE BRANCHES IN YOUR LOCAL MACHINE

Go to your local repo machine where you have the copy of your remote branch

Run the command ”git branch –a” to list down all the branches

1. The branch which starts with remote/🡪are the remote branches
2. The branch without remotes 🡪they are available on your local copy ad well

Checkout to the feature branch or the branch or the branch that you created

1. Git checkout <branch name>

Make sure that you are on the new branch by running git status or git branch command

Make some changes in this branch such as adding the files “touch file3 file 4 “

Git status

Git add <filename>

Git commit –m “<msg>”

Git push

Go to your github portal once again

Here see that the new changes are only available in your feature branch but not in the maun branch

LAB 8 MERGE OUR FEATURE BRANCH WITH MAIN BRANCH

Go to your github repo

Check the changes in your feature branch

Go to the pull request tab and click on create pull request

Put your main branch and the feature branch in the request blocks

Click on the create pull request and it will ask for a comment just click again on pull request

Go to pull request tab once again and click on the pull request available there

Click on review changes and then merge

Once done you can delete this feature branch if required or u can simply ignore it

Go to your code in main branch and see the changes are now visible here

LAB 9 (GO TO LOCAL MACHINE)

Go to your local machine where you have the copy of your remote repo

Checkout to the main branch

Git checkout<branch name>

Now run the command “git pull” to all the new changes such as branches from remote location

Here see that the new changes are only available in main branch.