# AWS Account Creation:

**STEP** 1:

<https://aws.amazon.com/free>

Email Address: root

Password and Confirm password: Strong Password

Account Name:

**STEP** 2:

Contact Information

--> Personal Account

--> Professional / Company Account

Address

**STEP** 3:

Payment Information

--> Visa / Master Card / AMEX (CC/DC)

--> 2 INR / 1 $ (Refunds back within 72 Hrs)

**STEP** 4:

Verify your Identity

--> Email / Phone Number

--> Enter captcha and Phone Number

--> Answer the IVR call, and Enter the 4 Digit pin generated on your screen.

**STEP** 5:

Choose the support plan : Getting assistance from AWS

1. Account and Billing Related : Free for All support plans.. 24x7 Support..

2. Service Limit increase : Free for All support plans.. 24x7 Support..

# IAM User Creation:

**TASK**: Create your AWS Account.

**TASK** 1: Enable MFA on your root account.

**TASK** 2: Create an IAM user with S3FullAccess and Test his level of access on account.

**TASK** 3: Create an IAM User with "Administrator Access" and verify his access on account including Billing dashboard.

**TASK** 4: Provide Billing access to TASK 3 IAM User. (use same user for all future sessions)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Policy

**TASK**: Create an IAM User with AdministratorAccess, and Add S3Deny and verify his level of access on s3 platform and other services.

**TASK** 2: Add TASK1 user to multiple groups, and use the policysimulator to understand the policy escalations.

**TASK** 3: Create a user with "AdministratorAccess", Enable billing access for this user, and use same user throughout our course.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# S3: Simple Storage Service:

**TASK**: User1, Need access only one bucket1. If user1 trying to access any other buckets he should end up with an error.

**TASK**2: Configure events on s3 bucket.

**TASK**3: Read S3 storage comparison chart and FAQs on s3.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**TASK**: Explore with S3Browser

**TASK**: User1, Need access only one bucket1. If user1 trying to access any other buckets he should end up with an error.

user 1 --> user1bucket --> Give a specific URL to user 1 to access his own bucket

user 2 --> user2bucket

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Introduction to Linux

**TASK**: Install httpd and delive the webpage using 8080 port number.

**TASK**: Configure http to https redirection using virtual host.

**TASK**: Install Nginix and deliver a webpage...

**TASK**: Create a volume and Associtae to Linux instance and make it available always.

**TASK** 2: Extend the **TASK** 1 volume with 1 more GB.

-----------------------------------------------------------------------------------------------------------------------------------

# EC2: Elastic Compute Cloud:

**TASK**: Launch a Windows ec2 instance with a keypair. Get connected to instance with the password.

Change the password at OS level, and signout.. Login back to your ec2 instance using Keypair generated password. try to login with newly setup password.

**TASK** 2: Create a user in Windows server and provide him "Local Admin Rights" and "Remote connection permissions”.

Connect to your ec2 instance as newly created user.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

D:2/1/2020

# Security Groups:

**TASK**: Open required ports to ping your ec2 instance.

**TASK** 2: Install IIS and deliver your website using public ip your ec2 instance.

**TASK** 3: Associate multiple ENI to an ec2 instance.. Launch a new ec2 instance, and try to access **TASK** 2 outputs using Both the private IP addresses.

**TASK** 4: Limit the **TASK** 3 output to only newly launched ec2 instance public Ip address using security groups.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

D: 2/3/2020

# Root Device Types: Storage options for our ec2 instances.

**DLM :** Data Lifecycle Manager : We can schedule Instance / volume snapshots automatically.

Need to define the appropriate TAGS.

**TASK** : Configure DLM to create a snapshot for every 2 Hours for a 1gb volume.

**TASK** 2 : Launch multiple instance in multiple AZs.. And perform attach volumes.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Linux EC2 Instance:

**TASK** 1: How to recover your Windows AMI ec2 instance, if you change default password and unable to retive the password.

(You still have the associated Keypair in you local laptop and AWS)

**TASK** 2: How to recover your Windows AMI ec2 instance, if you change default password and unable to retive the password.

(You have deleted the keypair from your local laptop and aws)

<https://docs.aws.amazon.com/AWSEC2/latest/WindowsGuide/ResettingAdminPassword_EC2Config.html>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

D: 2/5/2020

# ELB: ELASTIC LOAD BALANCER

# EFS: Elastic File System:

**TASK**: Configure EFS and mount Filesystem while launching ec2 instance to desired path.

**TASK** 2: Launch multiple ec2 instances, in multiple AZs. Make it as a webserver and mount EFS to Public Document Root path. And verify, both instances are delivering same webpage also modifications. Perform Permanent mount.

**TASK** 3: Configure Application ELB and run **TASK**2 instances on top of ELB.

Create a new instance make it as a webserver, create a New Target Group (8080), and deliver new instance via a newTG with 8080 port.

**TASK** 4: Launch and configure an ec2 instance, and create a new TG.

configure Weighted Routing on ELB with all the three Target groups.

---------------------------------------------------------------------------------------------------------------------------

D: 10/2/2020

# ASG: Auto Scaling Group:

**TASK**: Configure ASG with EFS...

**TASK** 2: Test the Scheduled Scaling on ASG.

**TASK** 3: Read the comparison chart for ELB.

--------------------------------------------------------------------------------------------------------------------------

**Task 1:** Launch an ec2 instance using Programmatic Access. later terminate using cli.

* step1. Goto this link <https://aws.amazon.com/cli/> and download the windows/mac installer.
* step2. open the IAM console and create a new user by access type "programatic access"
* step3. now open cmd and type aws to verify. type "aws configure"

AWS Access Key ID [None]: AKIA3Y4GBCTGSBBUR7YD

AWS Secret Access Key [None]: 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

"to get access key ID and secret access key = goto IAM console and select the user and goto security credentials tab and create access key then copy access key ID and secret access key. "

Default region name [None]: ap-south-1

Default output format [None]: table/Jason/text

* step4. To launch an instance, follow the commands.

aws ec2 run-instances --image-id ami-0217a85e28e625474 --count 1 --instance-type t2.micro --key-name pk --security-group-ids sg-0a48ab66360dd3e22 --subnet-id subnet-40f4c028

* step5. to terminate the instance: -

aws ec2 terminate-instances --instance-ids "your instance id"

--AMI ID

--Subnet-id

--Security-group-ids

--keypair

**Task 2:** Create an index.html and upload it to s3 bucket.

While launching an ec2 instance, make it webserver and get the object from s3 to Document Root path using Userdata. Q

Create a role and associate

#!/bin/bash

yum install httpd -y

service httpd start

chkconfig httpd on

aws s3 cp s3://bucketname/index.html /var/www/html/ --recursive

**Task 3:** Create a task using Task Scheduler to sync data between your s3 bucket and local path.

**Task 4:** Perform Cross Account access using Switch roles feature.

**Task 5:** Add S3 plugin to your LightSail WordPress website.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

D: 2/12/2020

# Elastic Beanstalk environment

**Task:** Reset the Windows instance local administrator password using Run Command

**Task 2:** Install Putty using run command

**Task 3:** Perform windows Critical Updated using Run Command

**Task 4:** Install apache/nginx using run command on Amazon Linux

**VPC: Virtual Private Cloud:**

**Step 1:** Create a VPC (CustomVPC) (CIDR: 192.168.0.0/16) (65531 Usable IPs) (Max /16)

**Step 2:** Create Subnets under CustomVPC (Supports /28 to /16)

ap-south-1a: Public SUbnet : 192.168.1.0/24 (251 Usable IPs)

sp-south-1b: Private Subnet : 192.168.2.0/24 (251 Usable IPs)

**Step 3:** Create an IGW and Associate with CustomVPC

**Step 4:** Created a Route Table under CustomVPC.

--> Added a route 0.0.0.0/0 --> Internet Gateway

--> Subnet Association: ap-south-1a (Public Subnet)

(Whatever the subnets not associated in above step, Default member of Main Route table and Main route table doesn't have any IGW, so no internet)

**Step 5** :(Optional): Enable auto assign public settings on VPC.

|  |
| --- |
|  |
|  |  |
|  |  |

--------------------------------------------------------------------------------------------------------------------------------------

DATE 18/2/2020

# VPC PEERING

**Task:** Launch an ec2 instance in Custom VPC Private Subnet and make it as a webserver.

Create an ELB in Custom VPC public Subnet and Deliver the web[page from Custom VPC Private Subnet instances.

**Task 2:** Apply NACLs to deny traffic to one network.

**Task 3;** Enable VPC Flowlogs to an s3 bucket and observe the logs related to blocked requests/ allowed requests.

**Task 4:** Design the AWS Architectural Diagram With

--> One Public Subnet

--> Two Private Subnets

--> NAT and NACLs

# RDS: Relational Database Service

**TASK:**

#!/bin/bash

sudo yum -y update

yum install httpd php-mysql -y

amazon-linux-extras install -y php7.3

cd /var/www/html

wget https://wordpress.org/latest.tar.gz

tar -xzvf latest.tar.gz

cp -rv wordpress/\* /var/www/html/

rm -rfv wordpress

rm -rfv latest.tar.gz

chmod -R 755 wp-content

chown -R apache:apache wp-content

service httpd start

chkconfig httpd on

#!/bin/bash

sudo yum -y update

yum install httpd -y

service httpd start

chkconfig httpd on

echo "<h1>This is a test page</h1>"

# Certificate Manager:

# CloudFront:

**Task 1:** Launch a WordPress website and Deliver the website using ELB, configure SSL using ACM, and deliver the WordPress with https. (Perform application level https redirection using virtual host)

Ref: --https://www.youtube.com/watch?v=93ekvcpzs1o

**Task 2:** Deliver a ELB Website via CloudFront distribution. Use the WordPress template.

# AWS inspector, directory devices, WAF, AWS shield, application services, SNS, SES, SQS, Standard queue:

**Task:** Migration Services: Migrate an Amazon Linux from Mumbai region to NV Region using Cloud endure.

**Task 2:** https://aws.amazon.com/getting-started/projects/deploy-drupal-with-amazon-rds/

https://aws.amazon.com/getting-started/projects/deploy-wordpress-with-amazon-rds/