Devops-Milestone1

1. Create instance from AMI in same as well as in different region.

Create instance and launch in terminal add http-80 in inbound rules:

```
yum install httpd -y
cd /var/www/html
echo “This is Mansi’s server” > index.html
systemctl start httpd
systemctl enable httpd
ip a s
curl http://localhost
curl http://ip-address-of-server
yum install vsftpd -y
yum install cfs-utils -y
yum install nfs-utils -y
cd /tmp
touch mansi.txt{1..5}
```

Copy public ip from aws and run on browser. right click on instance id and create AMI after creation launch instance from AMI
After instance creation launch it in terminal and check if rpmquery is already installed in it or not. Copy public ip from aws and run on browser.
Similary go to ami > actions >copy ami >add-region
Go to that region > ami >launch instance from ami >check httpd is installed or not > Copy public ip from aws and run on browser

2. EBS Disk partition, mounting, Increasing and Decreasing volume size

Create instance
Create Volume > gp2 >5GiB>100/3000 > create >name:data-disk
Select volume > attach volume to the instance created > /dev/sdb
Open instance in terminal:

```
lsblk
df -h
mkfs.<tab><tab>
mkfs.ext4 /dev/xvdb
mkdir /data
mount /dev/xvdb /data
blkid > copy uuid of xvdb and paste it below
vim /etc/fstab  { UUID=.....  /data  /ext4   defaults 0 0 }
df -h
cd  /data
touch mansi.txt
mount -a
```

Reboot the instance and check if data is mounted or not.
cd /data
ls
df -h

(Increase volume)
Go to volume >actions >Modify volume >change size to 10 GiB
resize2fs /dev/xvdb
df -h (size will be increased)

(Decrease volume)
We cannot decrease volume directly by modifying volume size. We need to make a new volume of desired decreased size and move the contents of the old volume to the new volume and delete the old volume.

Create new volume with 4 GiB > attach volume > /dev/sdc
lsblk
df -h
mkfs.ext4 /dev/xvdc
mkdir mansi-data
mount /dev/xvdc /mansi-data
cd /data
mv * /mansi-data/
blkid
vim /etc/fstab (paste uuid of xvdc and erase uuid line of xvdb)
umount /data
Reboot and check

(modify root volume i.e. /dev/xvda)
Modify volume from aws
df -h
xfs_growfs -d /dev/xvda1
growpart /dev/xvda 1
xfs_growfs -d /dev/xvda1
df -h

3. Creating Snapshots and copying it to another region

Create instance
Create Volume > gp2 >5GiB>100/3000 > create >name:data-disk
Select volume > attach volume to the instance created > /dev/sdb
Open instance in terminal:

lsblk
df -h
mkfs.<tab><tab>
mkfs.ext4 /dev/xvdb
mkdir /data
mount /dev/xvdb /data
blkid > copy uuid of xvdb and paste it below
vim /etc/fstab { UUID=..... /data /ext4 defaults 0 0 }
df -h
cd /data
touch mansi.txt{1..5}

Go to volume >action> create snapshot
Go to snapshot > action >copy snapshot > add desire region
Go to the selected region> create instance, create volume from the copied snapshot > attach it to the instance > /dev/sdb
Open instance in terminal
mkdir /snap-data
mount /dev/xvdb /data
cd /snap-data
ls (The contents of the web-data directory should be visible here)

4. Configure NFS and create replication to use it in another region

Create 3 instances Linux, ubuntu, redhat and add nfs-2049 inbound rule
Create file system in EFS, add security group in network tab, disable file protection
Attach > mount via ip > copy the sudo mount command
Open all the instances in terminal :
Linux-
rpmquery nfs-utils
mkdir /nfs-data
sudo mount ..... /nfs-data/ (paste it here)
df -h
cd /nfs-data
touch linux.txt{1..5}
Ubuntu:
apt update
apt install nfs-common
mkdir /efs-data
sudo mount ..... /efs-data/ (paste it here)
df -h

cd /efs-data
touch ubuntu.txt{1..5}

Redhat

yum install nfs-utils -y
mkdir /afs-data
sudo mount ..... /afs-data/ (paste it here)
df -h
cd /afs-data
touch redhat.txt{1..5}

Create an instance and file system in new region where you want to replicate the data – read only
In old region create replication
In new region attach fs to instance and check if the data is visible or not.

5. S3 bucket

Browser: AWS
Search s3 Bucket
Bucket name
ACL enable
unclick block public access > acknowledge checkmark
create bucket
add and upload file /folder
Go in file
click url and open in new tab : you will not be able to see
then go to file
and edit -> perimission -> everyone -> read -> save
click url and open in new tab : you will be able to see

Launch an EC2 instance

In Terminal:

1.yum install automake fuse fuse-devel gcc-c++ git libcurl-devel libxml2-devel make openssl-devel
2 git clone https://github.com/s3fs-fuse/s3fs-fuse.git
3 ls -la
4 cd s3fs-fuse
5 ./autogen.sh
6 ./configure --prefix=/usr --with-openssl
7 make

```
  8  sudo make install
  9  which s3fs
```

In Browser : AWS
Search IAM
create user -> username -> attach policy select (round option)
permission policy name -> search s3full -> next
create user -> .....(write name of your choice) click on the user >
create access key
application running on aws (tick)
acknowledge (tick) -> next
decription tag -> create access key
download .csv file
search IAM -> add MFA -> device name -(mobile-authentication)(or you can write anyname)

Go to Mobile:
open google authenticator app
scan QR code

In Browser:
Enter code 1(after 30 sec)
Enter code 2
Add MFA

In Terminal:

```
  10  touch /etc/passwd-s3fs
  11  vim /etc/passwd-s3fs
```

Open .csv file that you have downloaded before -> copy the first key in vim editor -> first key:second key

```
  12  chmod 640 /etc/passwd-s3fs
  13  mkdir /mys3bucket
  14  s3fs yourbucketname -o use_cache=/tmp -o allow_other -o uid=1001 -o mp_umask=002 -o multireq_max=5 /mys3bucket
  15  df -h
  16  cd /mys3bucket/
  17  ls -la
```

6. Protect your root account MFA

Go to IAM
2. Select Users
3. Choose the user
4. Go to their security credentials

5. Choose Virtual MFA device
6. # Open google authentication app in your phone
7. 1st MFA code will come , enter it
8. 2nd MFA code will come in 30 seconds or so enter it
9. Assign mfa
10. Copy summary & paste it on new tab
11. Enter user and passwd
12. MFA code will be asked for entering , go to authenticator where you will see a code on the screen m enter that code in mfa code bar .

8. VPC networks

a. Create Vpc > Vpc only

Name > vpc1
IPV4 CIDR > 10.0.0.0/16 > Create Vpc

b. Create Internet gateways and attach to vpc

Name > myigw1 > create
Action > attach to Vpc

c. Creation of subnets

Create two subnets
Select vpc1
Subnet name >public-subnet /private-subnet
Enter zone : prefer a
IPV4 subnet CIDR block : 10.0.0.0/24 / 10.0.1.0/24
Create subnet

d. Launch Instance

name: web-server
network settings:
select vpc1 network
select public-subnet
public ip : Enable
select key-pair and security groups
Add http, icmp in inbound rules
Create

e. Create route table

Name: public-rt
Select vpc1 > Create route table
Edit route > add >0.0.0.0/0 >Internet gateway > Select igw id > done
Edit Subnet association > select public-subnet >done

f. Open terminal login web-server and dev-server

SSH login web-server
sudo su –
yum install httpd -y
cd /var/www/html
echo “This is my server” > index.html
cd
systemctl start httpd

systemctl enable httpd

Check on the browser if the public ip of web-server is working...

g. Create dev-server instance in EC2

name : db-server

same as web-server : key and sg

select vpc1

select private-subnet

select disable

Create

- To login into private subnet- continue using web-server terminal

Check the name of your key.pem file and enter and copy the entire contents in the below vim file.

vim key.pem

chmod 400 key.pem

copy ssh command and login into private subnet instance

h. Create NAT Gateway

name: my-ngw

public-subnet

public

allocate elastic ip address

create

Create route table : private-rt

Edit route>add>nat gateway

Edit subnet association: private subnet

Try to Ping on db-server

Create one more vpc in another network

#peering connection

go to peering connection > create > old region to new region

my account

another region > choose new region to send

copy vpc id of the new region > create

check peering connection in new region > sctions > accept request

modify route tables > public rt > edit routes > add ip > peering connection > select id > create

go in the old region

modify route tables > public rt > edit routes > add ip > peering connection > select id > create

connect old region public instance instance in dev terminal >

ssh-keygen
cd .ssh
cat id_rsa.pub
cat authorized_keys

connect new region public instance to terminal
copy cat id_rsa.pub from old region and paste vim authorized keys in new region terminal
copy cat id_rsa.pub from new region and paste vim authorized keys in old region terminal

cat > white.txt in old region
scp white.txt root@ip:/

cat > black.txt in new region
scp black.txt root@ip:/

7. Cloud formation

Create EC2 from cloud formation:

1. Create a json file on notepad with the properties as mentioned and any other properties mentioned in the question,
use online json editor for identation.
2. AWSTemplateFormatVersion : '2010-09-09'
Resources:
(tab) EC2Instance:
(tab) Type: AWS::EC2::Instance
(tab)Properties:
(tab)InstanceType: t2.micro
(tab)ImageId: ami-0182f373e66f89c85
(tab)KeyName: testkey
to is get this ami id go on instances , click launch instance in quick start browse for more instances
you will get the info of the ami linux machine , ensure that it uefi preffered
#add only if required
InstanceSecurityGroup:
Type: AWS::EC2::SecurityGroup
Properties:
GroupDescription:Enable ssh access via port 22
SecurityGroupIngress:
IpProtocol: tcp
FromPort: 22
ToPort: 22
CidrIp: 0.0.0.0/0
3. Save this file with .json extension

4. Search Cloud formation in search bar
5. Create stack
6. Choose an existing file if template is provided in the question or create a json file whatever the question provides.
7. Upload the file (json file)
8. Next
9. Stack name > ecstack1
10. next
11. stack will get successfully created , go on instances and check if the instance is running.