

EC2

Creating a EC2 instance in proper way

- > Go to EC2 dashboard
- > First go to the key pair
- > Click on create key pair
- > Give it a name like (key-dev-nv)
- > Format can be pem or ppk
- > You can give it tag if you want
- > Click on create a key pair

- > Now go to the security group
- > click on create a security group
- > give it a proper name and description
- > click on add rules
- > give rules like ssh http and source my ip or anywhere
- > give a tag if you want and click create

- > now go to instance and click launch an instance
- > go to aws marketplace and select the AMI you need
- > then select instance type
- > click add storage
- > take the default and check delete on termination
- > fill some tags
- > now select the already created security group
- > review and launch now
- > choose the existing key
- > launch instance and connect with ssh

- > instance public ip is dynamic if you want to reserve an ip
- > go to elastic ip
- > click on allocate elastic ip address
- > create one
- > now go to actions and click associate elastic ip address and select the instance

EBS

- > now go to ec2 dashboard and go EBS
- > select the volumes and click create volume
- > choose the volume type and size
- > choose the availability zones same as your EC2 instance AZ
- > If you want you can encrypt the volume
- > give the tag
- > click on create
- > click on volume and click actions
- > select attach
- > choose the instance and attach to it

We only attached the volume to actually mount it you have to go to system and to do that go and see extras in linux command docs

EBS SNAPSHOTS

Snapshot is used to backup the volume or we can even use it move volume from one AZ to other AZ also used to change volume size ,type,zone and encryption

- > first create a volume and do all the needed things mentioned in aws extras
- > mkdir -p /var/lib/mysql
- > install mariadb (yum install mariadb-server -y)
- > start the service (systemctl start mariadb)
- > if you get any problem disable selinux and restart the system

Now we steps to do in aws

- > first go back to volume and select the volume
- > actions and click create a snapshot
- > give description and tag
- > click create
- > unmount the partition
- > Detach the volume
- > create new volume from snapshot
- > attach the volume created from the snapshot and mount it back

-> you can use snapshot to send for other aws user too by clicking on actions

-> you can make it public too

ELB

-> first create an instance while creating put the script so website will be auto created

-> if you launch the instance and paste the dns ip you can see website opened

-> now go to actions and select image and templates

-> choose create a image

-> give it a name and create it

-> an AMI and snapshot will be created

-> now you can automate creating instance instead of manually creating it

-> go to launch templates

-> click create a launch template and give it a name

-> go to my ami and select the ami

-> select the instance type

-> select the existing key

-> select the security group

-> give the resource tag for instance and volumes

-> create it

-> go to launch templates

-> actions and launch it

Go to load balance

We first creating a application load balance to do that we first need a target group

-> click create target group

-> you can change protocol if you need for now we keep it default

-> you can change health check

-> can change health check threshold and unhealthy threshold (ip/80 if not respond 2 times then it says unhealthy)

-> click next

-> select the 2 instance and click include as pending below

-> and create it

- > now go to create load balancer
- > select the application load balancer
- > give it a name
- > scheme select the internet (for public) other for private
- > select all zones (check all)
- > create a security group for it
- > select the created target group
- > select the listener and routing put the created target group

- > now if you put link of elb dns name you get 504 error (because targets are unhealthy can check that on target group)
- > now we have have edit the instance security group inbound rule
- > custom tcp 80 for elb-sg
- > now save it and go back to target group
- > you can now see targets become healthy
- > You can deregister or register the instances to the load balancer

- > you deregister a ami and delete the snapshot
- > you can also copy the ami and do some other things

CLoudWATCH

- > first go to ec2 instance and go to actions
- > select monitoring and there select the detailed monitoring
- > now go to cloud watch and metrics
- > ec2 and per instance metrics
- > select your instance cpu utilization if you can't find your instance
- > search with instance name or id

- > now we gonna make graph go up and down in our ec2 instance
- > go to your ec2 instance
- > yum install epel -release -y
- > yum install stress -y -> both are used to create a package to stress our vm
- > if both not work
- > sudo amazon-linux-extras install epel -y
- > now you install stress package

- > stress -c 10 -t 30 -> first just press stress and enter to see options 10 is no of process C is cpu t is time 30 is seconds
- > now if check at metrics cpu utilization will be high
- > now we can set an alarm for this
- > now go to alarms
- > select create alarm and select metrics
- > select the instance and cpu utilization ,time,greater than certain percentage
- > click next and create an sns topic and set email
- > click next and set alarm name
- > click create

EFS

Elastic file system is used if you want a cluster of servers (ec2) to store data at same place we can use EFS also called as shared file system

- > First go to security group and create one with nfs rule and allow for instance SG
- > now go to EFS dashboard
- > click on create efs
- > give it name and click on customize
- > keep it general purpose and burst mode for now
- > click next now remove default sg and put the one we created for all
- > and click on next and next and create
- > now create a access point
- > file system as a efs
- > click create access point

Now we need to mount it first we need to do some commands on vm

If its amazon linux system

[Manually installing the Amazon EFS client - Amazon Elastic File System](#)

Else

[Manually installing the Amazon EFS client - Amazon Elastic File System](#)

Now go to this and based on what we selected we copy command since i went with the access point i have to select that

[Mounting file systems using the EFS mount helper - Amazon Elastic File System](#)

```
file-system-id efs-mount-point efs _netdev,noresvport,tls,accesspoint=access-point-id 0 0
```

It will be like above now make changes to it

```
fs-06b44d9a7b480cc7b efs-mount-point efs  
_netdev,noresvport,tls,accesspoint=fsap-0fdfd32ccc0967072 0 0
```

Now go to vm and move the files in /var/www/html/images directory files to /tmp/backup

vi /etc/fstab and paste the above text into this file and save it

Since we have to mount it to certain path make the path changes now paste it at fstab

```
fs-06b44d9a7b480cc7b /var/www/html/images efs  
_netdev,noresvport,tls,accesspoint=fsap-0fdfd32ccc0967072 0 0
```

mount -fav

Now can check df -h and move back files in backup to that directory

-> now go to ec2 instance and create a ami and delete the snapshot keep it for next session

AUTO SCALING

- > first go to the auto scaling option in ec2 dashboard and select launch configuration
- > click on create a launch configuration
- > give it a name and select the ami we created
- > choose the instance type (t2.micro)
- > we can enable detailed monitoring
- > you can go to advance and commands on userdata if you want
- > you have to select your pre created security group and key of instance

- > now go to ELB and select target group
- > give it name and keep everything default
- > you don't have to give instance yet auto scaling group will put it
- > click create

- > now go to load balancer click create
- > select application type
- > give name and check all zones
- > put a sg group if not created then create one and add target group
- > click create

- > click on creating auto scaling group
- > give it a name
- > you can select launch template or switch it to launch configuration
- > click next
- > select the zones you want to place your instance
- > click next
- > now select existing load balancer
- > click choose from your load balancer target group
- > check load balancer health check
- > you can also enable group metrics collection (to get all instances in group metrics)
- > group size set desired as 2 and min 2 and max 8
- > select the scaling policy (if you put none AS will just keep 2 instances running)
- > now select scaling policy name and metric type
- > give the metric(like cpu utilization) and target value (cpu usage)
- > if you don't want your instances to auto terminate
- > click next and you can add notification
- > now give the tags

-> click create

-> now if you want to make any changes to Auto scaling you will do that from auto scaling and go to created one and go and edit and update

-> if you want to make any changes to instance you will create a new launch configuration and update it in auto scaling

-> you can add policy for auto scaling group

-> go to the options and click add dynamic policies

-> select policy type = step scaling

-> give it a name and create a alarm select the instance id and set usage >60

-> create a alarm

-> now back to dynamic policies and add the alarm and now we can set action for alarm

-> in take action add no of instance should create when load is higher than 60 and can add multiple for 70 80

-> click create

-> now delete the default policy and same way add scale down policy

-> instead of add now change in action from add to remove

-> create it

-> in instance management next to policies you can set some actions for instances like set scale in protection

-> you can detach and attach an instance here to attach go to instance and actions->instance settings ->attach to auto scaling group

-> right next to this there will be instance refresh we use that we update or launch a new instance we select the percentage around 30%

AWS S3

- > go to s3 and click create bucket and give it a name
- > uncheck block public access if you want it to be public for now we not doing that
- > enable versioning if you want to have backup of data
- > enable encrypting if you want
- > now click on create a bucket
- > click on bucket that you have created
- > check out objects,properties and permissions etc
- > now go to object click upload and select file
- > while uploading file you can specify the storage class
- > and can even encrypt
- > click create
- > we can open it directly but by clicking open but we cant open with url because it's not public
- > to do that go back to permissions and uncheck the public access
- > now go to object ownership and enable acl
- > now go to object and actions and can make it public
- > now it works

You can use s3 for store and access from anywhere or to make other use or to host static website

- > download a static website from tooplate and unzip and drag and drop files in s3 bucket
- > now select all files at once and actions and enable public
- > now go down in properties and enable static hosting
- > give index.html and error.html and save it
- > now you will get a link at static hosting
- > When u click on version u will see old one and new one
- > you can download old one and reupload to override
- > you can delete the current version and old one will be back
- > now go management and go to lifecycle-policies
- > click create and give it a name
- > select life cycle rule action like expire etc
- > set storage class and time in days to change to that class

- > now select expire as how many days after you want this file to be deleted
- > give prefix and create
- > you can also replicate data to save your data into other regions s3 buckets
- > this can be used for disaster recovery

RDS

- > first go to the RDS
 - > click create a database
 - > you can choose easy or standard option we going with standard
 - > select the engine option (mysql in this case)
 - > go to templates and select the free tier just to be safe (selected dev/test)
 - > give the database name
 - > click auto generate password
 - > db instance size is burstable one and t3.micro
 - > set storage settings size for now is 20gb
 - > availability and durability and select the select create a standby instance
 - > can make it public access or private access
 - > you can select the default sg or create a new one for vpc
 - > you can click advance configuration and set a port number if you want
 - > again click on advanced configuration and name the database
 - > you can enable backup and enable replica
 - > you can stream logs to cloudwatch so select them all (logs)
 - > click create a database
- > now create a EC2 instance in same region within same vpc
- > copy the rds username and password and copy the rds endpoint in a notepad
 - > now ssh into the instance
 - > sudo -i
 - > apt update
 - > apt install mysql-client -y
 - > telnet endpoint 3306 (to see if it is connected or not)
 - > it will not connect because the security group of rds is not allowing the instance to connect
 - > now go to rds sg and change the my ip to instance sg or put ec2 private ip/32

Now mysql -h mysql-db01.cm29dj1cr1dw.us-east-1.rds.amazonaws.com -u admin
-pRjVDMkxcCUY93Ak4wQz2

- > rds you can see status backing up
- > now go to rds snapshots and systems to see backed up snapshot which can be used to restore our rds
- > You can go to rds instance and you can do some stuff like create a read replica ,migrate database ,take snapshot etc
- > you can also modify rds
- > you can go to parameter group and create one and attach it by modifying the existing rds

ROUTE 53

From vprofile lift and shift project

- > take note of instances private ips
- > now go to route 53
- > click create a hosted zone
- > give domain name like vprofile.in
- > select private zone
- > select the region (NV)
- > put default vpc and create
- > now go to create records
- > select simple record give blog name as db01 etc
- > also give ips and you can add more ips by clicking add record

- >aws s3 mb s3://vprofile-arti
- > aws s3 cp filename s3://vprofile-arti
- > aws s3 ls s3://vprofile-arti -> to see items in s3 bucket
- > aws s3 cp s3://vprofile-arti/vprofile-v2.war /tmp/vprofile-v2.war

When creating a auto scaling group select an IAM role in launch configuration

IAM ROLE

- > to connect the s3 bucket to our tomcat ec2 instance
 - > go to iam roles
 - > click create role
 - > choose ec2
 - > permissions set s3 full access
 - > create a role
 - > go to ec2 instance and choose tomcat instance and go to actions
 - > security > modify the role
 - > add it and save
-
- > cd /var/lib/tomcat
 - apt install awscli -y -> to install awscli and access services (if have role)
 - telnet db01.vprofile.in 3306 -> used to see if they are connected to other instances
-
- > when we use https in ELB we use our arm certificate

AWS Pricing calculator

- > you can go to there and estimate the price of the service
- > go to <https://calculator.aws/#/> and click create estimate
- > You can search for whatever service you want and you can calculator the monthly price