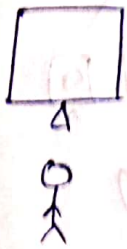
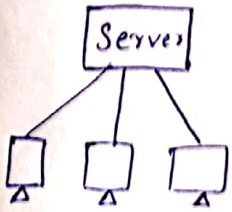
Computing Models

1. Desktop Computing

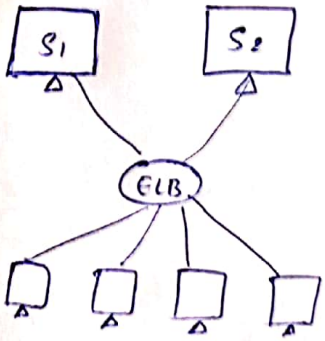


2. Client-Server Computing



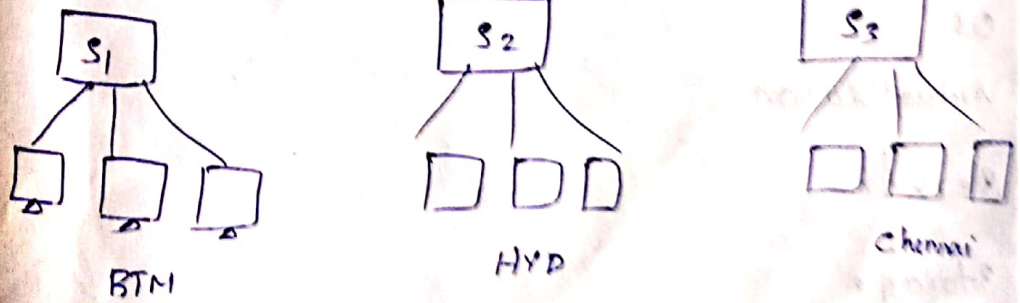
eg: Bank office

3. Cluster-Computing



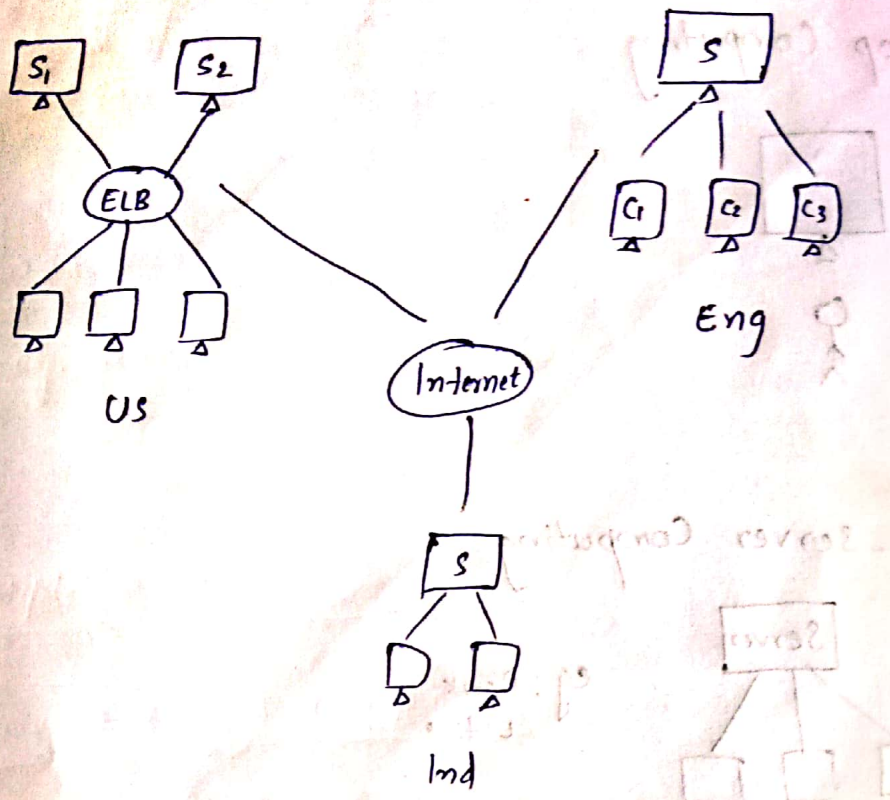
Elastic load balance

4. Grid. Computing



Cluster + Grid ⇒ Cloud Computing

5. Cloud - Computing



Cloud computing is the on demand delivery of ~~servers~~ IT resources through the Internet with pay-as-you go pricing.

Service Models

Applications

Data

Runtime

Middleware

OS

Virtualization

Server

Storage

Network

Iaas

Appn
Data
Runtime
Middleware
} you manage

OS
Virtual Server
Storage
N/w
} manage by provider

Paas

Appn
Data
} you manage

Runtime
Middleware
OS
Virtual Server
Storage
N/w
} manage by provider

Saas

Appn
Data
Runtime
Middleware
OS
Virtual Server
Storage
N/w
} manage by provider

on-premis

Appn
Data
Runtime
middleware
OS
Virtual Server
Storage
N/w
} you manage

Iaas → Infrastructure as a service

Paas → Platform as a service

Saas → Software as a service
↳ providing all the IT requirement

Iaas

eg: AWS, Azure, some public clients.

In Paas, developers are can create.

Paas

eg: google engine
Salesforce

Saas

eg: gmail

Deployment Models

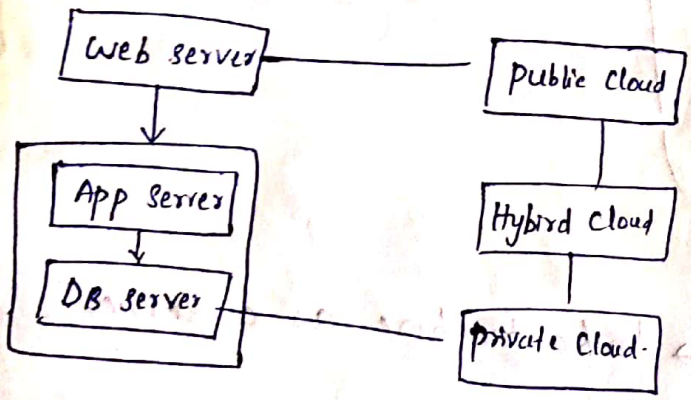
1. **private cloud Computing**

A company or organisation own servers or resources in his location.

2. **Public cloud** : Servers or resources sharing it with publ

Url: aadarcard.com

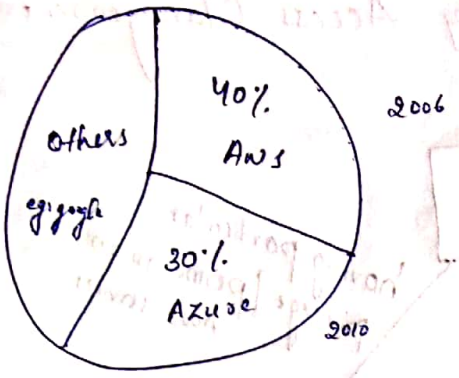
3. **Hybird Cloud**



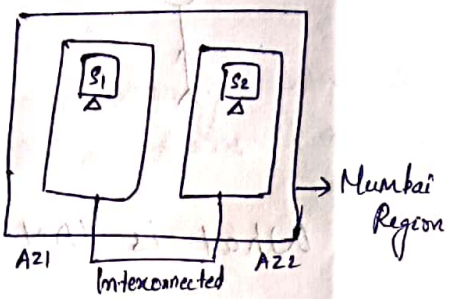
The combination of public and private cloud i called **hybird cloud**.

4. **Community cloud**

Sharing the resources or servers among the companies.



AWS



Region

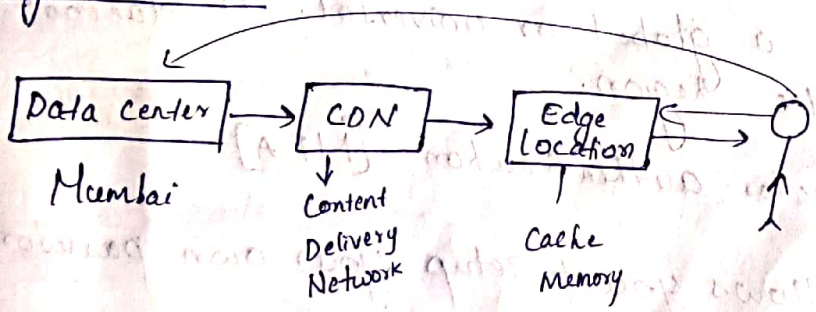
A Region is the geographical location where Amazon has infrastructure. The regions are designed to be independent to each other with separate powerforce, network connection and geographically location.

Availability Zone

We have one region in India ie; Mumbai.

AZ is a separate data center within a region Minimum two availability zones in a region.

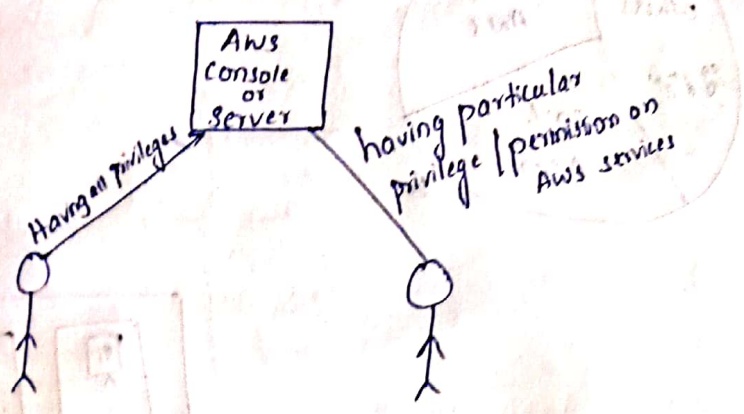
Edge location



Edge location is also one of the data centers but we cannot host any server in edge location. (we dont have permission) only used to store in cache memory.

31-01-19

IAM – Identity Access Management



What is IAM User?

IAM is a web service that helps you securely control access to AWS resources.

- IAM allows you to manage users and there level of access to the AWS console.

Features

1. Centralised control of your AWS account.
2. You can provide share access to your AWS account.
3. You can grand different permission to different users for different services.
4. IAM is a global or universal. It doesnot apply to specific region.
5. Multifactor authentication. [MFA]
6. IAM allows you to setup your own password rotation policy.

Components of IAM

- user
- group
- policies
- Roles

User

Using IAM you can create and manage AWS users. and use permissions to allow and deny there level of access to AWS services.

Root user

- Root user having all the privileges or permission on the system/console.
- Here AWS account console controlled by Root user who created AWS account with email id is the owner or root user to the console.

IAM user

- IAM user is a normal user just like in linux.
- Here we can create maximum 5000 users per console.
- Initially IAM user are normal user doesnot have any privilege on IAM services.

Steps to Create users

1. IAM
2. Select users
3. Add user

4. User name

5. Access type
   - ○ Programmatic access → cli, pyamatic
     API, CLI
   - ◉ AWS management console ⇒ username with password login

6. password
   - ○ Autogenerated
   - ◉ Custom password (own password)

* Require password reset at next sign-in
  ☐

If we want to reset the password click the ☐

6. Next permission

7. next tags

8. Review
   Create user

9. Close

⇒ user created.

we created the username & pwd, if we want to login with the user then *

How to login into user

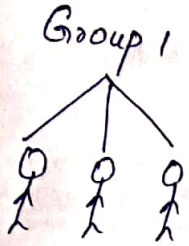
- Go to dashboard
- copy signin link
- open different browser (one)
- paste url
- ~~Typ~~ provide username & password
- sign-in

These users don't have permissions to create users groups

(Normal user login in the

Group

Group is a collection of IAM users to easy administrate the users.



- IAM users inside the group are having one set of permissions.
- we can create maximum 300 groups.
- An IAM user can be member of 10 groups.

Steps to create groups

1. Go for Groups
2. Create New group
3. Group Name

↳ Next step

Create group

⇒ group is created.

Adding users into groups

1. Select the group
2. Add users to group
3. Select users
4. Click on Add users

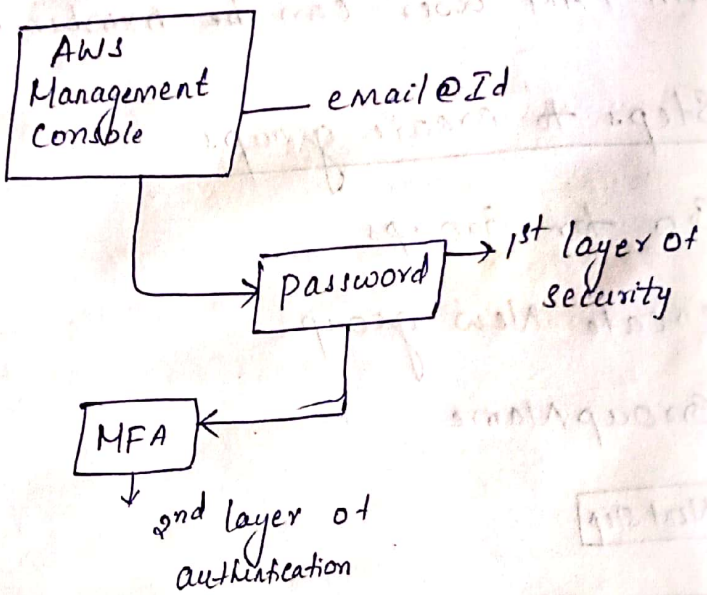
→ For checking ⇒ click on groups.

☐ group 1
☑ group 2

OR

1. click the box
2. Then group Actions
3. select Add users to Group
4. select users
5. Add users

* Activate MFA on your root account



1. Click on Activate MFA

Manage MFA

2. Multifactor authentication

Activate MFA

- ⊙ Virtual MFA device

Continue

4. Show QR code

3. Type two consecutive MFA codes

MFA code1 [ ]

MFA code2 [ ]

[Assign MFA]

Goto dashboard.

* Appy an IAM password policy

→ ~~click on apply~~ account

→ password rotation policy for IAM user

Steps

→ Click on account settings

→ Click particular options

→ [Apply password policy]

→ Go to dashboard.

5/02/19

How to delete user

1. Select users in IAM
2. Click particular user
3. [Delete user]
4. [Yes, delete]

5/02/19 * How to change root account password /email-id

1. click your account user name
2. ▾select my security credentials

click here

3. change the email / password
4. Done

* How to change normal user password

→ Dashboard

1. Click account settings

☑ Allow users to change their own password

Apply password policy

→ Login into normal user

→ select account

→ select my security credentials

Change password

→ current pwd : xx

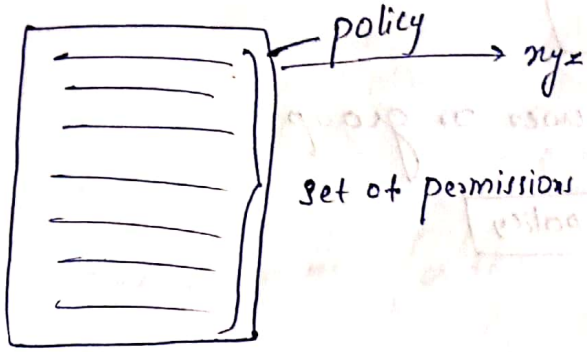
new pwd : xx

Change password

Total 7+ services in AWS

Policies

- A policy is a document that fully defines a set of permissions to access and manipulate AWS Resources.
- Permissions in the policy determine whether the request is allowed or denied.
- Policies are stored in AWS as JSON (Java script object Notation) script.
- policies are attached to the users, groups and roles.



- Depend upon policy users are classified as two types:

Admin 1. Administrative user

sub user 2. Power user

Administrative

- He has complete control over the console

power user

He has complete control over AWS Resources or console expects IAM.

- we can attach policy (permission) after creating user & groups or we can attach policy while creating users & groups.

* Attaching policies for ... groups:

→ click on policies

↳ Any particular policy

Q search s3

→ select any policy

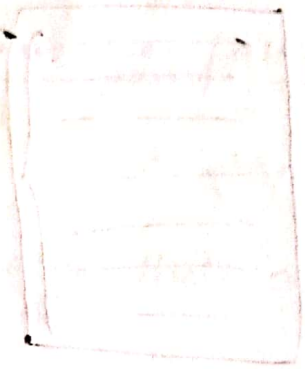
• Amazon s3 Full Access

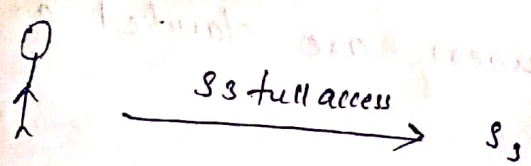
→ Policy actions

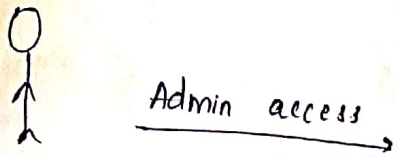
→ Attach

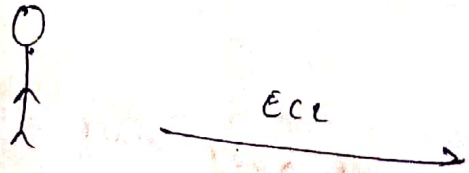
→ select user or group

→ Attach policy









* Attaching policies while creating user

Dashboard

→ Click on user

→ username: [ ]

→ Console access

→ Click on [Attach existing policies]

☐ PowerUser Access

→ select policy

→ [Next Tags]

→ key: name Value: poweruser / or skip

→ [Review]

→ [Create user]

Information about users

Credential Report

dashboard

→ Click on credential Report

→ [Download]

↳ In this files, we will get all the information about users.

2/6/19

Roles

IAM Role is a IAM entity that defines set of permissions for making AWS service requests.

- IAM Roles are associated with a trusted entities such as IAM users, application and AWS services. eg: EC_2
- we can create maximum 1000 Roles in AWS account.
- Each role can have upto 10 policies attached
- we have 3 types of roles

1. Amazon service Role
2. Cross Account Access Role
3. Identity provider Access

1. Amazon service Role

Granting permissions to applications running on EC2 instance.

Between 2 services, creating a role.

EC2 — AWS service
  |— service role
S3 — AWS service

Steps to create service role

IAM → dashboard
→ select Roles
→ Create Role

→ select [AWS service]

→ select EC2

→ [Next: Permissions]

→ Q search s3 services → select s3 Full Access

→ [Next tags]

→ [Review]

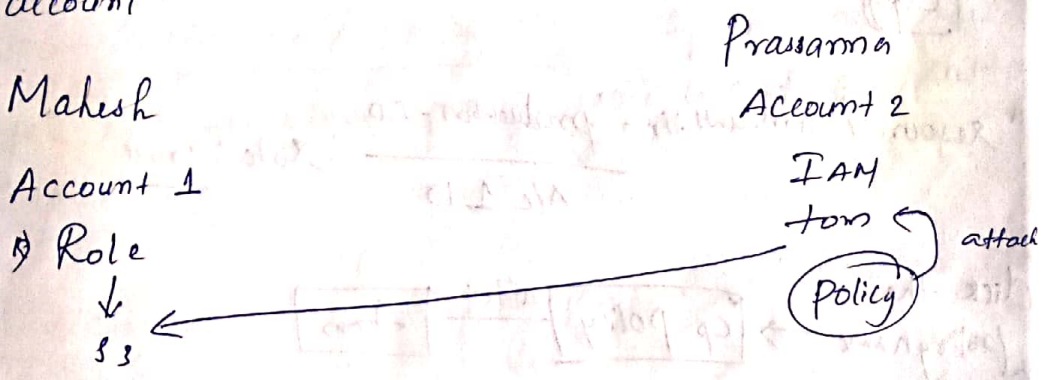
→ Role name :

→ Role description : Allows EC2 instance to call S3

→ [create Role]

2. Cross Account Access Role

Granting permissions to IAM user in another account.



1) Role
2) Cross A/c role
3) Account ID [A/c 2 ID]
4) Role name [Cross]
5) Create Role.

Step 2
6.

2nd Account

1) User

Create user

Username: [ ]

2) Policies

3) Create policy

Create policy

Select JSON

Remove old codes → Paste the JSON code

Search: To get the code:-

Open another tab

Type: Cross A/c Role

Tutorial: Delegate Access

6.

Copy

"Resource": "arn:aws:iam::production-ID / A/c 1 ID :role/cross

4) Click the Policyname → Cp policy → attach → tom

→ Login with sign-in link

→ go to Account

- Switch role

Switch role

A/c ID | Role cross

7/2/19

Instance Creating

EC2 or ECC → Elastic Computing Cloud

- Amazon EC2 is a web service that provides resizable compute capacity in the cloud.
- Amazon EC2 reduces the time required to obtain and boot new server instances to minutes.
- Amazon EC2 allowing you to quickly scale capacity both up and down as your computing requirement changes.

EC2 options

- **On Demand EC2**: allows you to pay a fixed rate by the hour or by the second with no commitement.
- **Reserved EC2**: provides you with a capacity reservation and it offers a significant discount compared to on demand EC2.
  - we have to reserve the instance minimum one year or maximum 3 years.
- **Spot Instance / spot EC2**: spot instance enabled you bid whatever price you want per instance capacity.

Dedicated host: physical EC2 server dedicated for your use.

- Dedicated host can help you reduce cost by allowing you to use your existing server ~~bound~~ bound software licenses.

EC2 Instance Types

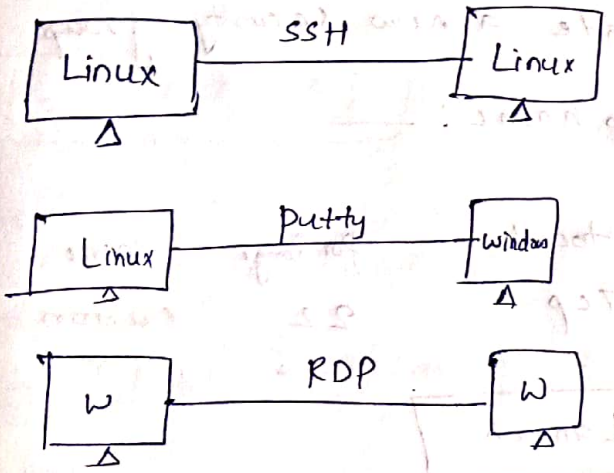
EC2 Instance Types

Family	speciality	use case
D2 ↓ Density	Dense storage	File servers OR data warehouse or Hadoop
R4 ↓ RAM	Memory optimized	Memory Intensive application Iops (Input Output per seconds)
M4 ↓ Main choice for general purpose	General purpose	Application servers
C4 ↓ Compute	Compute optimized	cpu Intensive
G2 ↓ Graphics	Graphics Intensive	Video encoding or 3D application streaming
I2 ↓ Iops	High speed storage	No sql database or data warehousing
F1 ↓ Field programming graphics	Field programming	Hardware acceleration for your code
T2 ↓ Cheap general purpose (Think T2) micro	Low cost, general purpose	Web servers or small databases
P2 ↓ Graphics pixels quality	Graphics speciality or general purpose	Machine learning bid coin mining
X1 ↓ extreme memory	Memory optimized	SAP or Apache Spark

Amazon EC2 Key pairs

- A Key pair consist of public key and private key.
- public key

  To encrypt password.
- The recipent use the private key to decrypt password.
- To login your instance you must create a key pair specify the name of the key pair when you launch the instance and provide the private key when you connected the instance (PPK file).
- Linux instance have no password and you use a key pair to login using ssh with windows.
- Maximum we can create 5000 key pairs.



Remote desktop protocol.

Steps to launch Instance

- Select EC2 d
- Instance
- Lanuch Instance
- Select Red hat Free ~~Type~~ Tier

Select

→ General purpose t2 micro free tier

→ Next : configure Instance Details

→ No changes

Next : Add storage

→

Next : Add Tags

→

Next : configure security group

→ select: Create a new security group

Security group name : _____

Type	Protocol	Port Range	Source
SSH	TCP	22	Custom

Review and Launch

Launch

Select: ▾ Create a new key pair

Key pair name : lab1

Download key pair ↵ (One file will be in downloads)

Launch Instance

View Instance ✓

Name

Change the name: Server #1

How to login into instance (save private key)

- open puttykey
- load
- path: (where u downloaded)
- select pem file

  labl.pem → open → ok

- Save private key
- Yes
- save again

  Another name: name: labl → save

- close

Check the instance state

How to login into instance

- select Instance

  server 2

  Server2 IP: _____ copy public key IP

- open puttygen: paste IP addres
- + SSH
- ~~Authentication~~ Click on Auth → browse the ppk file (new saved file)

Select ppk file

lab1.ppk

open

open

Yes

login as:

Terminal

login as: ec2-user

$ sudo -i

login as root user

#

---

* How to terminate

- Select server
- actions
- ↓
- Instance state → Terminate.

11/2/19 * How to launch windows Instance

Select EC2 ↓

Instance

Launch Instance

- Select windows Free-tier

Select

- General purpose t2 micro

free-tier

Next: Configure Instance Details

Next: Add Storage

Next: Add Tags

Next: Configure Security group

→ Select: ⊙ ~~Existing~~ Create a new Security group

Security group name:

Type

RDP

Review and Launch

Lanuch

Select: ▾ select existing Key pair

Key pair name:

☑ accept

Lanuch Instance

View Instances

Name: Windows

* How to login windows from windows

① ~~copy password~~ Run

① Run
↓
mstsc

Paste public IP

Click on Another user

Connect.

② Select [Connect]

Click on [Get Password]

[Choose file]

[ ] (pem-file)

[Decrypt password]

↓

Copy password.

→ Login windows

Username: Administrator

Password: ***

[OK]

* How to login + Linux ~~machine~~/Instance from linux

→ Launch Instance

→ Download Keypair
(.pem file)

→ ~~Login~~

→ ~~Logon into~~ Instance

→ change the permission → [Connect]

chmod 400 xyz.pem

↳ ~~Change to login~~

and then login.

12/2/19

How to protect Instance termination Protection.

→ We can set while launching instance or we can modify set after launching instance.

→ While launching

↳ Configure Instance

→ Enable termination process ☑ Protect against accidental termination.

→ After launching

→ select Instance

Actions

↳ Instance setting

↳ change Termination Protection

↓

Yes, Enable (Make it to enable)

12/2/19 EBS - Elastic Block Storage

An EBS allows you to create storage volumes and attach them to EC2 instance.

- Once attached, we can create a file system on the top of the volume.
- Amazon EBS volumes are placed in a specific availability zone where they are automatically replicated to protect from the failure of a single component.

- You can mount multiple volumes to same machine but each volume can be attached to only one instance at a time.

Types of EBS volume

1. General purpose SSD (GP2) (solid state Drive)
2. Provisioned IOPS SSD (io1)
3. Cold HDD (sc1)
4. Throughput Optimized HDD (st1) (Hard disk)
5. Magnetic (standard)

1. General purpose SSD

- It balances both price and performance.
- Ratio of 3 IOPS for GB with upto 10k IOPS.

Provisioned

- Designed for Input ouput Intensive applications such as large Relational database or no sql database.
- use if we need more than 10,000 IOPS.
- It provides maximum 20,000 IOPS per volume

Magnetic

- It is a low cost compared to all EBS volumes ie; bootable.
- Magnetic volumes are ideal for overloads

where that data is accessed infrequently.

4. Throughput optimized

- This type of hard disk we are using in big data, datawarehousing, log processing.
- This is not bootable.

5. Cold HDD

- This is low cost for infrequently access workloads.
- This kind of Hard disk is using file server.

* How to create a new volume:

- EC2
  ↳ ~~click on~~ Elastic Block store
  ↓
  Click on volumes

→ Create Volume
→ Volume typed
→ Select anyone
→ Make sure that creating volume should be same as where the instance is present.
  ↳
→ Create volume
⇒ Volume created

Modifying the size of Root volume EBS

Note :- Don't dettach Root volume.

- select particular volume we want to modify
- Actions → modify volume
  ↓
  give specific size
  ↓
  Modify.
  ↓
  yes

Backup of hardisk.

Snapshots / Backup of EBS

- To make a volume available outside of availability zone, you can create a snapshot of volume and restore that snapshot to a new volume anywhere in the region.
- you can copy snapshots to other region and then restore to new volumes. making it easier to leverage multiple AWS regions for geographical expansion, data center migration and disaster recovery.

Steps to create snapshots :

→ click on volumes
→ select particular volume to take snapshot
→ click on Actions → create snapshot.
→ description: [ ]
→ [create snapshot]

→ Check the status in snapshot

* Restore from snapshot to new volume

→ ~~Note~~ create a new volume and make size equal to snapshot size or greater than that.
→ ~~while creating~~ (but we cannot reduce the size)
→ copy snapshot Id while creating paste it
→ Attach to new instance.

login
ec2-user
sudo -i
#blkid
#lsblk
#mkfs.ext4 /dev/xvdf
#mkdir /rose
go for mounting. Temporary mounting
#mount /dev/xvdf /rose
#mount -a
#df -h

Permanent mounting
#vi /etc/fstab
↳
last line
/dev/xvdf /rose ext4 defaults 0 0
:wq!

Mount -a

df -h

cd /rose

touch a1 a2

ls

a1 a2 lost+found

After completing mounting, we need to take a backup of that volume, go for volumes.

OR

13/2/19

* Restore from snapshot to new volume

→ Select snapshot

→ select particular snapshot

– go for actions

→ create volume

⇒ Volume Type:

Size:

Availability Zone:

→ Create Volume

⇒ Go Attach Volumes to another instance

⇒ Select the new (Newly created volume) volume

⇒ Actions → Attach volume

Instance: ...

Attach

→ Go for Instance → check whether it is copied or not

→ Copy the public IP and logon to the instance.

→ log in into that instance

→ check with lsblk and blkid

→ create mount point and mount it

Note : Don't assign file system again. If you assign that data will be formatted.

same as firewall in linux.

Login

```
# lsblk
# blkid
# mkdir /jerry
# mount /dev/xvdb /jerry
# cd /jerry
# ls
```

* Copying a snapshot to another region

→ select snapshot

→ Click on Actions

↓

→ Copy

Destination Region: Select the particular region

→ Copy

⇒ Then check with that destination region whether it is copied or not.

follow the steps: Restore from snapshot + new volume

→ If we want to check whether it is copied or not first create volume, then attach it, then create an instance, logon to the instance, mount it and then

lsblk

blkid

ls

same as firewall in linux.

Security groups

A security group is a virtual firewall that controls the traffic (services) for our instance we can add rules to allow specific traffic to reach our instance.

- you can create upto 500 security groups for each VPC (virtual private cloud).
- You can add upto 50 inbound and 50 outbound rules to each security groups.
- It we need to apply morethan ~~than~~ 50 you can associate upto 5 security groups.
- All inbound traffic is closed by default.
- All outbound traffic is allowed by default.
- You can have any number of EC2 instance with a security group.
- ~~One~~ one instance can have multiple security groups.
- You cannot block specific ip address using Security groups instead use network NACL (Network Access control List).

6. Configure security group.

Steps

→ Assign a security group : ⊙ Create a new security group

→ Security group name : sg1

→ Description : sg1 created 2019-02-13

→ Add rule

Type	Protocol	Port range	Source
SSH	TCP	22	Anywhere

→ Add how many rules we want

→ Review and Launch

⇒ After creating the instance, if we want to add rules again

→ Click the particular instance

→ then click on security groups : sg1

View inbound View outbound

→ edit and add the rules

→ Ok

IPs

Classes in IP

Class A : 1 - 126

B : 128 - 191

C : 192 - 223

D : 224 - 239

E : 240 - 254

Private IP

A : 10.0.0.0

B : 172.16.0.0
    ⋮
    31

C : 192.168..0.0
        ⋮
        255

~~Elastic~~

Note:

- When you stop an instance and then if you start the instance, we will get another IP.

Elastic IP Address

- Elastic IP Address is a static IPV_4 address designed for dynamic cloud computing.
- An elastic IP address is associated with your Aws account.
- with an elastic IP address you can mask the failure of an instance or software by rapidly remapping the address to ~~ad~~ another ~~address~~ instance in your account.
- Elastic IP address limited to 5 IP address to region.

~~Step~~

Steps for creating Elastic IP :

- Network & Security
- Select ~~Cel~~ Elastic IPs
- Click on [Allocate new address]
- [Allocate]

* If you want to attach particular instance

- Select ~~Instance~~ EIP
- Actions → Associate addresses
- Instance : Particular instance

Private IP : _____

[Associate]

Checking : Instance
Select Instance & check the IP

EBS

Note

We cannot mount one EBS volume to multiple instance at a time; Instead u EFS.

14/2/19

Elastic Network Interface

An elastic network interface is a virtual network interface that you can attach to an instance in a VPC.

- Network interfaces are available only for instances running in a VPC.
- A network interface can include the following attributes:

1. Public IPv4 address
2. One primary ~~private~~ Private IPv4 address
3. one or more secondary private IPv4 address
4. one elastic IPv4 address
5. one or more IPv6 address
6. one or more security groups
7. A MAC address

Steps to create

→ Network and security

→ Select Network Interface

→ Create ~~Also~~ Network Interface

Description: — —

Subnet : Select the available zone

Private IP : ——

Security group: Select the security group

Yes, Create

⇒ For attaching Network Interface to Instance

→ Click on Network Interface

→ Select Instance ID : _____

→ Attach

→ Then go to instance

→ check whether created or not [It will show two IPs address and eth0, eth1]

Install packages and we check with ip on another instance

AMI — Amazon Machine Image

- An AMI provides the information required to launch an instance which is virtual server in the cloud.

*- You must specify a source AMI, when we launch an instance.

- You can launch multiple instances from a single AMI, when you need multiple instance with the same configuration.

- An AMI includes the following :

(a) A template for the root volume for the instance.

eg: an operating sim, An application server / s/w or architecture

(b) AMI can copy to the same region or different region.

- 2 types of AMI

(i) Instance stored AMI (Temporary storage)

(ii) EBS (backup) AMI

* EBS is permanent storage

Steps to create AMI

- Select instance to take image
- Click on Actions
- Image → Create image

Imagename : myimg

Image description : myimg

Create image → To recheck :- Click on images
Click on AMIs.

* Launch instance using AMI

- Instance
- Click on launch instance
- Click on My AMI's (left side)
- Select particular backup.

Select

- Continue the same steps
- Check whether it is copied or not
  (using IP) both browsers and puttygen

* Copying images to another region

- Images
- Click on AMIs
- Click on Actions
- Click on Copy AMIs

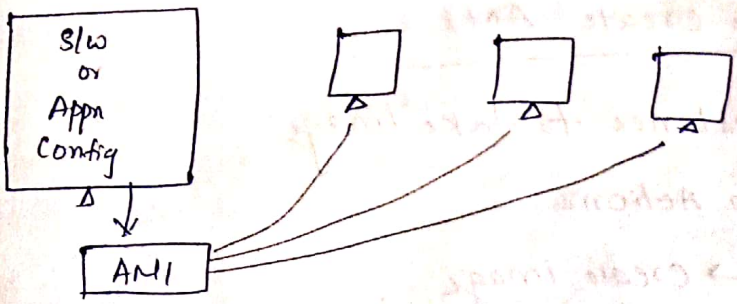
Designation Region: Mumbai

Copy AMI → Check :- launch instance → My AMI's → particular backup → Select and continue the same process

To rename the copied image
Click on
→ actions
→ deregister

Scanned by CamScanner

we can take the backups to any another regions.



Steps

#yum install httpd -y
systemctl start httpd
systemctl enable httpd
Vi /var/www/html/html.index

:wq!

cat /var/www/html/index.html

Boot strap ~~scripts~~ scripts

- Boot strap refers to a self starting process or set of commands without external input.
- with ~~easy to~~ EC2 we can bootstrap the instance during ~~the creating~~ launching instance with custom commands such installing s/w packages, running updates and configuring other various setting.

Steps

- EC2's
- Lanuch Instance

- Configure Instance
- Click on Advanced Details

```
#!/bin/bash
yum install httpd -y
systemctl start httpd
systemctl enable httpd
echo "Bangalore" > /var/www/html/index.html
systemctl Restart httpd
```

- Continue with the steps to create the Instance
- Check with IP address in another browser

18/2/19

Elastic Load Balancer

Elastic Load balancer distributes incoming ~~tra~~ network traffic into multiple targets such as EC2 instance, Containers and IP address in multiple availability Zones.

- You can add or Remove instance from your load balancer as your needs change.
- You can configure health checks which are used to monitor the health of computer Resources. So that the load balancer can send request only to the healthy ones.

Types of Load balancers

1. Application Load balancer
2. Network Load balancer
3. Classic Load balancer

1. Application Load Balancer

- Path based routine
- Route traffic to multiple server
- Route traffic to different ports and same EC2 instance
- It support HTTP, HTTPS and web sockets.
- It works OSI layer [7] (Application layer)

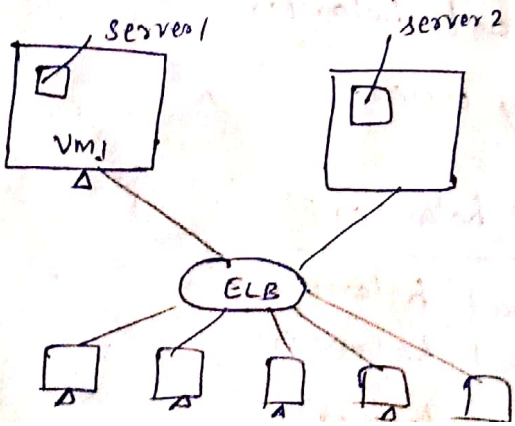
2. Network

- Choose a Network Load balancer when you need ultra high performance and static IP address.
- Network Load balancer are capable of handling millions of requests per second while maintaining ultra-low latency.

3. Classic

- Simple Load balancing across EC2 instance
- It supports HTTP, HTTPS, TCP and SSL.
- It works on OSI layer 4 & 7. [Transport layer and Application layer].

Steps to create



Steps to create Application

- Load balancing
- click on target groups
- click on [create target group]

Target group name : tg1
protocol : HTTP

- click on [Create]

⇒ Target group created.

→ select target group
- click on targets (down)
- [edit]
- Select the instances (add the instances into groups) (how many we want)
- ~~Save~~
- click on added to registered
- [Save]

→ click on load balancer
- [create load balancer]
- select Application load balancer
- [create]

Name : elb1

Availability zones
⇒ click on all availability zone
- Click on [Next: Configure security settings]

→ Create 2 no. Instances with coding ..

```
#!/bin/bash
yum install httpd -y
systemctl start httpd
systemctl enable httpd
echo "server1">/var/www/html/index.html
systemctl restart httpd
```

→ After launching ins 2 instances
→ create target groups.

- Next: configure Routing
- Click on 7
  Target group : Existing target group
  Name : tg1
- Next: Register Targets
- Review
- Check whether all availability zones added or not.
- Create
- Copy DNS name and open the browser.

2. Steps to create Network load balancer

→ Click on target groups
  Target group name : tg2
  protocol : TCP

Create the instances

→ Same steps as application ⇒ But we need to select network and load balance

3. Steps to create classic

Here no need to create targets directly we can create load balancer

→ Click on load balancer
→ Create Load balancer
- Click on Classic Load balancer
  Create
  Load balancer name : elb3
  Load Balancer Protocol: Add select HTTP, HTTPS, TCP → Click

→ Next: Assign security groups

→ Assign a security group: select the security group

→ Next: configure security settings

→ Next: configure health check

→ Next: Add EC2 instance

→ select the instances

→ Next: Add tags

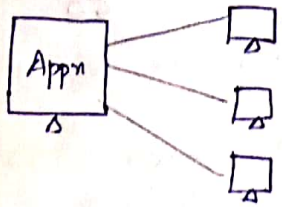
→ Review and create

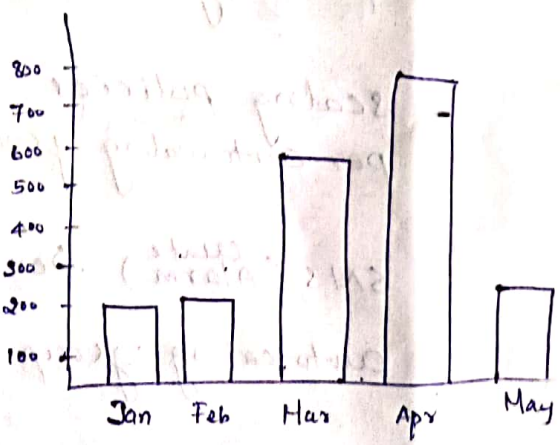
→ copy the DNS name and check

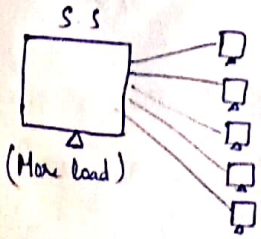
11/2/19 Auto scaling

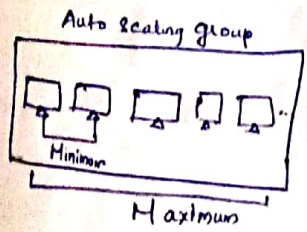
Auto scaling is amazon hosted service for automatically launching and terminating EC2 instance.

if the load is more, automatically launching EC2 instance









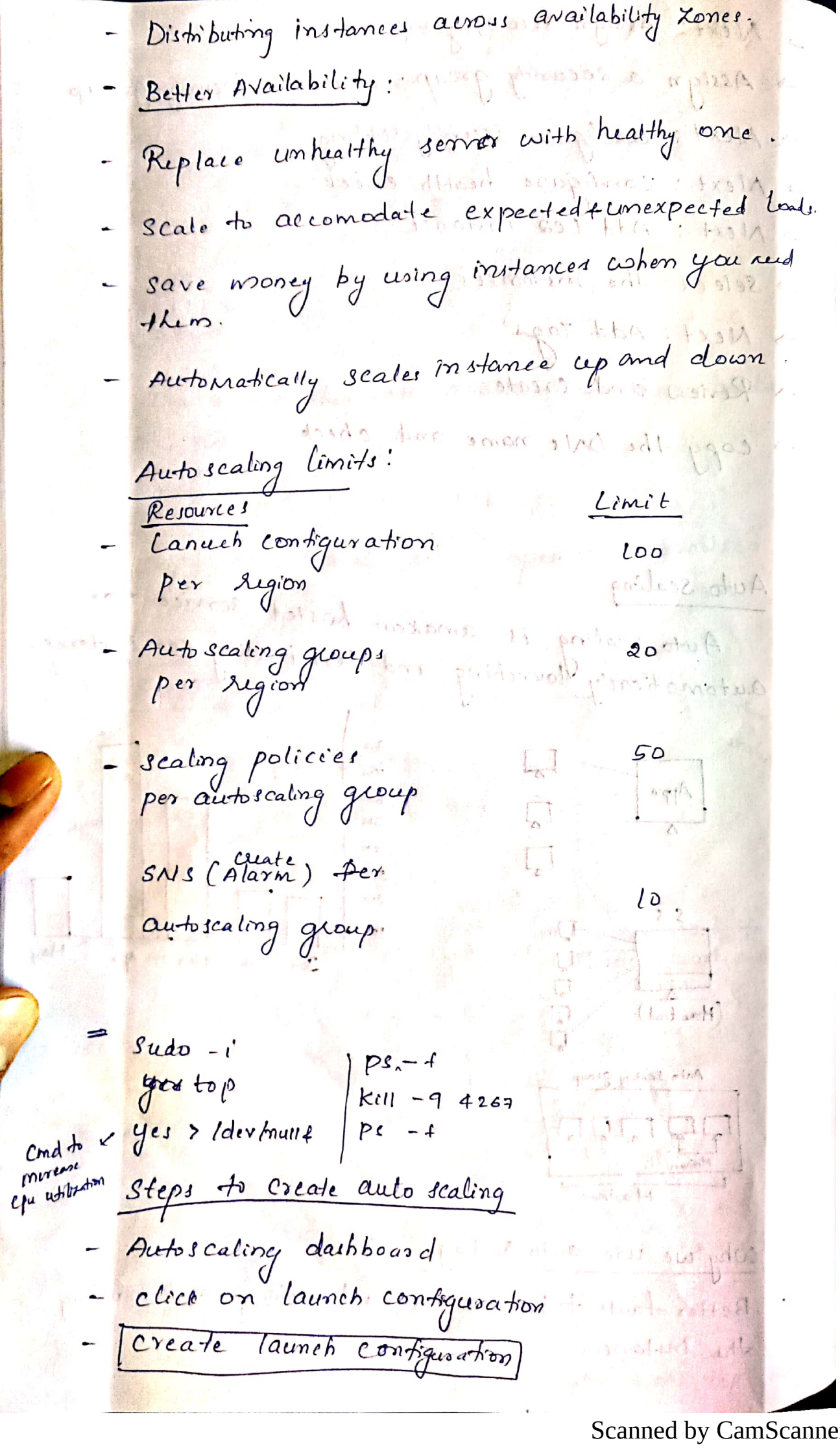
- Why we use auto scaling

- Better fault tolerence means launching or terminating the instances.

- Distributing instances across availability zones.
- Better Availability:
- Replace unhealthy server with healthy one.
- Scale to accomodate expected & unexpected loads.
- save money by using instances when you need them.
- Automatically scales instance up and down.

Autoscaling limits:

Resources	Limit
- Launch configuration per region	100
- Autoscaling groups per region	20
- scaling policies per autoscaling group	50
- SNS (Create Alarm) per autoscaling group	10



= Sudo -i
yes top
yes > /dev/null &

ps -f
kill -9 4267
ps -f

Cmd to increase cpu utilization

Steps to create auto scaling

- Autoscaling dashboard
- click on launch configuration
- Create launch configuration

- select particular operating system [Linux function]
- Next : configure details
- Create Lanuch configuration Name
  Name : lc
- Next : Add storage
- Next : configure security group
- select the security group
- Next Review
- create launch configuration
- select the key
- create

going to terminating
Symbol shows

⇒ Click on autoscaling group in dashboard
- Create Autoscaling group
- ⊙ use an existing launch configuration
- select launch configuration
- Next step
- create Autoscaling group
  Group name : scalegroup
  Group size : start with [1] instances
  Network : default
  subnet : select all availability zones
- Next : configure scaling policies
- ⊙ Click on use scaling policies to adjust the capacity of this group.
  scale betw [1] and [5] instances

Scroll down and click the link

- Click on scale the Auto scaling group using step & simple scaling policies

- Increase group size
  - Add a new alarm
  - create topic
  - send a notification (user name & mail id:)

  Is: [>=] [60] percent

  set the timing

  - create alarm

- Take the action: Add [1] instances

- Decrease group size

  send a notification:

  - Is [<=] [30] percent

  - set the timing [1 minute]

  - create alarm

- Take the action: Remove [1] instances

- Next: configure notification
- Next: Add tags → Review
- create auto scaling group

↳ Check in instances
copy the IP address and
to ~~putty~~ puttygen

Note

After login into Instance

top

ps -f

yes > /dev/null &

To cl
the sy

Scanned by CamScanner

20/2/19

After mounting. if we want to increase the size of the hard disk (xvda). first we need to ~~modify~~ go to volume, then ~~modify~~ go to actions, modify the size.

After that Logon to the instance (terminal) check.

```
# lsblk
# blkid
# growpart ^ /dev/xvda ^ 1
# blkid
# xfs_growfs ^ /dev/xvda ^ 1
# lsblk
```

To check the the system

In linux

*→ After mounting, it was if we want to increase the size of the harddisk: # lv extend ^ -L ^ +100M ^ /dev/vg/lv.

*→ After mounting, if we want to increase the size

↓

```
# lvextend --resizefs -l +50 /dev/vgname /vname
                                              /sun
# lvextend --resizefs -l 100 " "
```

To reduce

```
# lvreduce --resizefs -l -25 " "
# lvreduce --resizefs -l -75 " "
```

and logon

Note

xfs not support lv reduce direct cmd

```
# umount /sun
# e2fsck -f /dev/vgname /vname
```

#resize2fs /dev/vgname/vname

lvreduce -l 75 /dev/myvg/mylv

mount -a

21/2/2019

Global

S3 — Simple Storage Service

This is the first service, created in 2006.

Just like google drive

- S3 provides with secure, durable, highly scalble object storage.

Amazon S3 is easy to use with the simple web service interface to store and retrieve any amount of data from anywhere on the web.

- S3 components:
  1. Bucket
  2. object
  3. Region

Bucket

- Root level folders you create in S3 or refered as a bucket.
- Any sub folder you create in a bucket is refered as a folder.

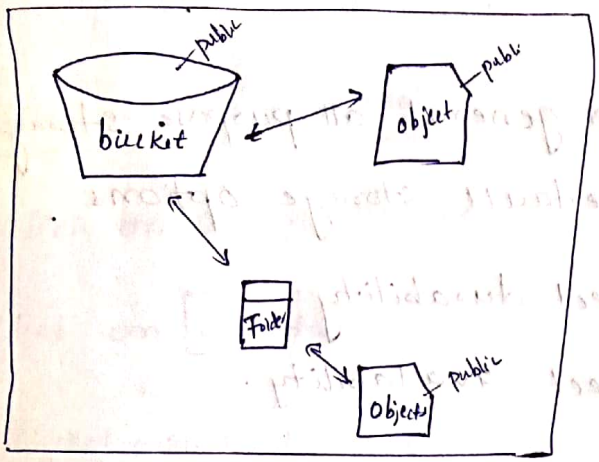
Object

- Files stored in a bucket or in a folder is refered as objects

Region

- When you create a bucket we must select a specific region for bucket.
- This means that any data that you upload to S3 will be physically stored on that region.

Structure of S3



S3 bucket rules

- Bucket names must be unique across all of AWS. (among the region).
- Bucket name must be 3 to 63 characters in length.
- Names can only contain lower case letters, numbers and hypens (-)
- You can store unlimited data in S3.
- one object maximum size is 5TB

(Types) ## Storage classes

- Standard
- Reduced Redundency storage (RRS)

- Infrequent storage (S3-IA)
- Glacier

- Each storage class has different storage cost, object availability, object durability and frequency of access.

1. Standard

- Designed for general all purpose storage
- It is the default storage options.
- 99.9% object durability
- 99.9% object availability.
- It is most expensive storage class.

Infrequent storage

- Designed for object that you do not acce frequently, but it must be immediately available when you access.
- 99.9% object durability and object availa
- less expensive than standard or RRS.

Reduced Re

- Designed for non-critical, reproducable obje
- 99.9% durability and availability.
- It is less expensive than standard storage cla

* Glacier

- Designed for long term archive purpose.
- It takes 3-5 hrs to restore from glacier (It won't available immediately)
- 99.9 % object ~~ava~~ durability.
- It is very low cost compared to all.

Steps to create a bucket

- click on s3 in storage
- Click on [create bucket]

  Bucket name : plastic6

  Region: specific region

- Create

Upload objects to bucket

- Select bucket
- click on upload
- Add files
  ↳ select file → open
- Click on [upload]
- We need to make it as public (if we want to see the file), if we didn't make the bucket and the object as public (we can't see the contents inside the object).

Make bucket as a public

- click on bucket
- click on permissions
- Edit
- unclick all boxes ☐
- Save
- Type: confirm
- Confirm
- success
- click on bucket
- click on object into public
- Make public
- success
- Then click object url

⇒ Now we can see the content inside the folder

Within a bucket create a folder

- click on bucket
- Create folder

Name *

[ ]

Save

folder created.

- Inside folder upload files and make it as public.

- ~~Create~~ Giving S3 permission to IAM users

- create some users → tom & jerry
- Attach S3 Read only permission – tom
- Attach S3 full access permission – jerry
- Login using sign link
- go to S3
- check permission
- S3 Read only means tom can view / download objects.

→ S3 full access → Jerry can ~~can~~ view / download / upload objects.

folder

25/2/19

Properties of S3

- Versioning
- Server access logging
- Static website hosting
- Object-level logging
- Default encryption
- Object lock
- Tags

- Transfer acceleration
- Events
- Requester Pays

Management

1) Life cycle
2) cross region replication

* Versioning

- Versioning means keeping multiple variance of an object of the same bucket.
- You can use versioning to preserve, retrive and restore every version of every object stored in s3 bucket.
- with versioning we can easily recover with our object from the buckets.
- Once we enable ~~bucket~~ Version in a bucket it can never return to an unversion or disable state.
- You can however suspend ~~on the~~ versioning on the bucket.

* Steps to enable versioning

- click on particular bucket
- click on properties
- click on versioning

- Click on enable versioning
- Save
- Again select the bucket
- upload new file with more content
- Make the object as public
- click on latest version (on the top of the properties)
- copy the link and check.

* In this, we will get only new version but we won't get old version.

* Backup

- select the object
- Actions → delete
- Delete.
- Click on show
- Delete marker
- Actions
- delete
- Then check in Bucket main page, deleted file restored.

*
- If it is in enable mode, then only we can get backup.
- if it is suspend mode, we won't get backup (it will delete)

* Logging

- In order to track requests for access to your bucket, you can enable access logging.
- Each access log record details about bucket name, request name, request action, respond code and error code any if there (it will display)
- Access log information can be useful in security and access audit.
- Logging is region specific.

Steps to enable server access logging.

- Select particular (source) bucket
- Go for properties
- Click on server access logging
  - click on Enable logging
    - Target bucket [source / ~~same~~ bucket or different)
      - Any name
    - Target prefix
      - log (Give any name)
    - Save
- After enabling, go to source bucket
- Upload some files
- make it as public

* Logging

We can create logging in same bucket or other bucket also

- Inorder to track requests for access to your bucket, you can enable ~~to~~ access logging.
- Each access log record details about bucket name, request name, request action, respond code and error code any if there (it will display)
- Access log information can be useful in security and access audit.
- Logging is region specific.

Steps to enable server access logging.

- Select particular bucket (source)
- Go for properties
- Click on server access logging
  - ↳ click on ⊙ Enable logging
    - Target bucket (source ~~same~~ bucket or different)
      - Any name
    - Target prefix
      - Log (Give any name)
    - Save
- After enabling, go to source bucket
- Upload some files
- Make it as public

Static Website Hosting

- You can host a static website on S3 bucket
- To host your static website you configure an amazon S3 bucket on website hosting and then upload your website content to the bucket.

Steps

- Click on the bucket
- properties
- Click on static website hosting
- Click on ⊙ use this bucket to host a website

Index document

index.html ← hosting page

* save the file as .html format.

Error document

error.html ← error page

- then save
- Upload these two files in the bucket and Make it as public
- Click on bucket
- properties
- Click on static website hosting
- Click on the link

Endpoint: http://

* If we give the correct url, it will shows the contents.

* if the url is wrong, it will go the error page.

26/2/19

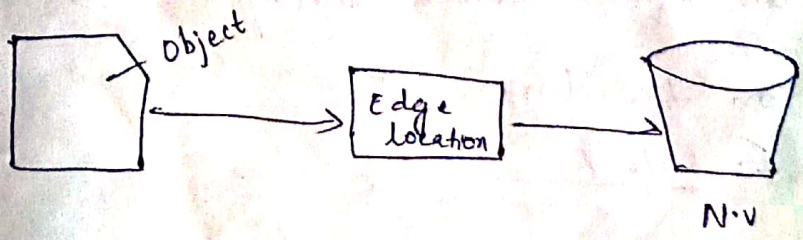
upload data very fastly

Transfer Acceleration

- we are using transfer acceleration feature for uploading objects very fast to bucket (region).
- we are uploading objects to nearest end points. that edge location will send object to specific bucket.

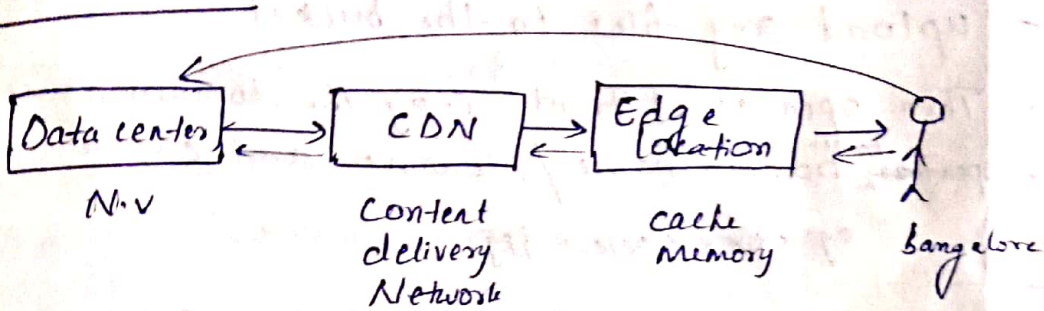
Steps to enable transfer acceleration

- Click on bucket
- select properties
- click on transfer acceleration
- ⊙ Enabled [Make it enable, to upload to send the object very fastly]
- [Save]
- If you want to compare
- After enabling it, upload one object to bucket make it as public



Cloud front

download the object very fastly.



Before starting we need to create a bucket and upload any files.

Steps

- Services
- Networking
- Click on cloud ~~Print~~ front
- Click on Create distribution
- web

  eg: html, css, php and graphic files.

Click ⇒ Get started

RTMP

eg: streaming media (live channel)

Get started

- Origin settings

  Origin Domain Name — Your bucket name

  Origin path — If any folder are present inside the bucket, we need to mention here — folder name

  Click on create distribution

- Copy the domain name & check, ~~it~~ the distribution in the browser. ~~like it~~

- upload any files in the S3 bucket
- Then open cloud front, copy the domain name.
- open the browser, Domain name / file name and check.
  eg: xxxx / Desert.jpg

---

* - Cloud front is for distributing (Read or downloading) files are object & effeciently.
- Transfer acceleration is for transfering (uploading) files are object efficiently.

* Default encryption

- Automatically encrypt objects when stored in Amazon S3.

Step

- Go to IAM & create 2 users
- Give S3 full permissions to users
- Go to encryption keys in the dashboard
- Get started
- Create key and give alias name
- Alias (Key Name): Rose
  Description: My key
- select users twice to allow that users to access bucket
- Click on Finish.

- Go to s3 service
- select particular bucket
- click on properties
  - ↳ select default encryption
  - ↳ select AWS-KMS
  - select a key [▲]
- upload a new object and make it as public
- Go to IAM
- selet sign-in link in dashboard and logon in another tab.
- Select s3 and open bucket.
- select new object (what we uploaded)
- Try to open, download or upload, we will get error.
- logout.
- Then logon with another user
- Try to open, download or upload

---

tom
↓ s3 full access

Key → (upload or download) ↑↓ possible

ram
↓ s3 full access

↑↓ Not possible

if we remove the keys from user, then the user can't have the access.

⇒ Management in S3 [if we want to delete a bucket automatically, we can assign the days]

- ~~In S3~~ Steps
- click on S3
- click on particular bucket
- click on Management
- Li Lifecycle
- click on Add lifecycle rule
- → Enter a rule name : [ ]

Add ~~prefix~~

→ Add filter to limit scope to prefix / tags

[Type to add Prefix /tag filter] , [Next]

- Storage class transition

☑ current version ☐ previous version

For current version of object - Click on + Add transaction

Object creation : Days after creation.

select a transaction : select days.

[▼] [days]

Click on [Next]

- Configure expiration

☑ current version ☐ previous versions

☑ ~~Expire~~ expire current version of object

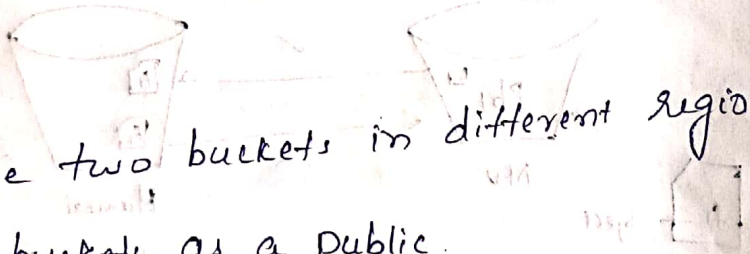
After [select days (1-...)] days from object creation

- Next
- [Save]

27/2/19

Cross region replication

- cross region replication is a bucket level feature that enables automatically asynchronous copying of objects across buckets in different AWS regions.
- The object replicas in the destination buckets are exact replica of the object in the source bucket.
- They have the same key name and same metadata.
- Existing objects of source bucket will not be copy on the destination bucket.
- The source and destination bucket must be enabled versioning.
- The source and destination bucket must be in different AWS region.



steps:

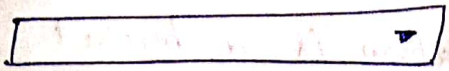
- create two buckets in different region
- Make buckets as a public.
- Enable both bucket versioning
- Click on source bucket
- click on management
- Click on replication
- Click on Add Rule
- ⊙ Entire bucket

Next

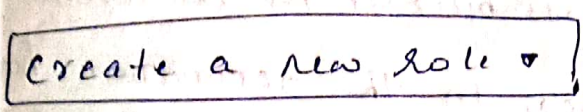
* if we want to send existing files, ~~before~~ means before we need to add them by procedure. (scp)

Scanned by CamScanner

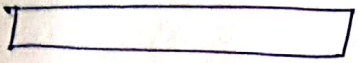
- Destination bucket



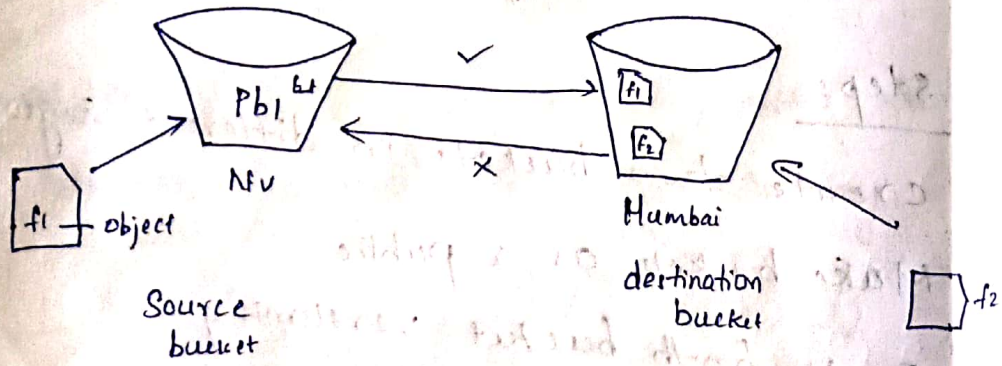
- Select Next



- Role Name



- Next
- Go to source bucket
- Upload one file and make it as public
- Check in destination bucket whether it copied or not (source file)



- In source bucket, if we upload any files, it will available in destination bucket.
- But, in destination bucket, if we upload any files, it wont go to source bucket.
- ~~At~~ a In this bucket, we can create only one destination bucket, but we can't create more than one destination bucket.

* If we enable ~~object~~ logging information from object level, will get complete log

Cloud Trail

- AWS cloud trail is an AWS service that helps
- Cloud trial is a new feature, it will record 90 days logs by default in your console or account.
- AWS cloud trial is an AWS service that helps you enable governance, compliance and operational and risk auditing of your AWS account.
- Actions taken by a user, role or an AWS service are recorded as events in cloud Trail.
- Cloud Trail is enabled on your AWS account when you create it. When activity occurs in your AWS account, that activity is recorded in a cloud trail event.
- Events include actions taken in the AWS Management console, AWS command line interface and AWS SDKs and APIs.

* By default the logs will stored for 90 days. This will visible on cloud trail dashboard
Else Logs if we want, we can store in bucket.

Steps to create a cloud Trail

- Management & Governance
- Cloud on Cloud Trail
- on dashboard click on trail
- create trail
- Trail Name [ ]
- Apply trail to all Regions

Cloud Trail

- AWS cloud trail is new AWS service that helps
- Cloud trial is a new feature, it will record 90 days logs by default in your console or account
- AWS cloud trial is an AWS service that helps you enable governance, compliance and operational and risk auditing of your AWS account.
- Actions taken by a user, role or an AWS service are recorded as events in cloud Trail.
- Cloud Trail is enabled on your AWS account when you create it. When activity occurs in your AWS account, that activity is recorded in a cloud trail event.
- Events include actions taken in the AWS Management console, AWS command line interface and AWS SDKs and APIs.

Steps to create a cloud Trail

- Management & Governance
- Cloud on Cloud Trail
- on dashboard, click on trail
- create trail
- Trail Name [ ]
- Apply trail to all Regions ⊙ Yes

- Read / write ⊙ All

- Storage location

  Create a new bucket ○ Yes ⊙ No

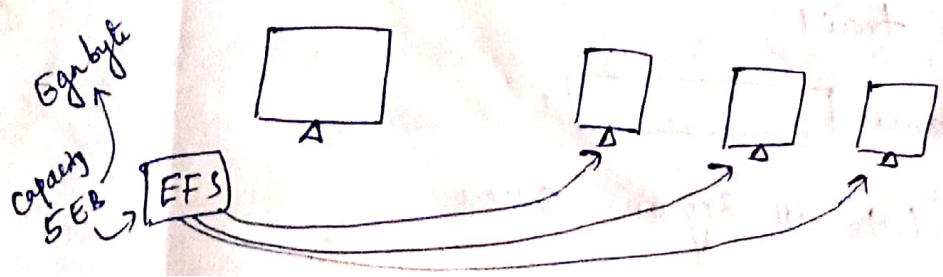
  S3 bucket [▾] (Select any existing bucket)

  [Create]

- Check in Cloud Trail
- Upload any files in the bucket
- Then check in the cloud trail → It will show all the log information. [year, month, date]

EFS — Elastic File System

- Amazon EFS provides simple, scalable file storage for use with Amazon EC2.
- With EFS, storage capacity is elastic, growing or shrinking automatically as you add and remove files.
- Multiple EC2 instance can access an EFS at the same time.
- With Amazon EFS you can pay only for the storage used by your file system.



EFS – NAS (Network as a Storage)

EBS – SAN (Storage as a Network)

Steps to create EFS

- Storage
- Click on EFS
- Click on create file system
- ~~Vpc~~ select Vpc default.

  VPC ▭

  VPC
  Virtual private cloud
- Create mount targets.

  Availability zone | Subnet | IP | security groups.

  ☐ | | | mysg (default)

  ☐ Select all zones
- Next
- Key | Value

  Name | ~~Foo~~ EFS
- Next
- Create File System → It shows Available
- Go to EC2
- launch multiple instances
- ~~Creat~~ after creating Instances
- Logon Into Instance

  ↓

  EC2-user

  Sudo -i

yum install nfs-utils -y

mkdir /sun

- Go to EFS
- Click on amazon EC2 Mount instructions (from box)
- Using the NFS Client

  Copy the mount :

  Sudo mount -t

-

copy path /^ /sun (my mount point)

- Logon to other instances
- Follow the same steps.
- #MKdir /moon

  # mount it

  # cd /moon

  # touch f1 } Create a file

  # ls.

  f1

- Agaon go to 1st instance logon
- Check the file created or not (what we create in 2nd instmce logon)

  # cd /sun

  # ls

  f1

from local vpc)

EBS, S3, EFS

EBS – Block storage

S3 – object storage

EFS – Network file system as storage

28/8/19

Service role between EC2 to S3

EC2 service

|—— service role (mysrole)

S3 service

Steps to create service role betw EC2

- Go to IAM
- click on role
- Select service role
- click on EC2
- Next
- Give permission - S3 & ~~EC2~~
- Next
- Assign role name : mysrole
- Create role

2. EC2 launch & logon

- launch on instance
  - ⇒ Amazon linux free tier [here all the packages installed]
  - ⇒ select IAM role [MyRole]
- launch instance
- logon to instance through IP address
  - ↓

```
# EC2-user
# sudo -i
# aws s3 help
# aws s3 ls [It will show the bucket created in s3]
# ls -a [to see hiding files]
```

To create new bucket

```
# aws s3 mb s3://Bucket name   (mb → make bucket)
# aws s3 ls
```

To remove bucket

```
# aws s3 rb s3://Bucketname   (rb → remove bucket)
```

AWS CLI

- Create user with programmatic access (CLI mode)
  (with ec2 full permission)
- While creating it will provide access key and Secret Key id. Copy the ids and save in somewhere else download the file.
- Go to windows cmd [or create an amazon linux instance and do
- Install the packages in that]

Install aws cli

Install python

Install PIP's

aws --version

python --version

aws configure → AWS Access Key ID: ...
AWS Secret Access Key: ...
Default region name: ...
Default output format: table

ls -a

here one hidden file will get created (.aws)

cd .aws

.aws] # ls

Config credentials

.aws] # cat config

[default]

Output = j → table

.aws] # cat credentials [here the user information stored]

AWS CLI

Create a new user with programmatic access. policy (permission): EC2 full permission

- Create user with programmatic access (CLI mode) (with EC2 full permission)
- While creating it will provide access key and secret key id. copy the ids and save in somewhere. orelse the download the file.
- Go to windows cmd. [Or create an amazon linux instance and do in that]
- Install the packages

package (install it)

```
# Install aws cli
# Install python
# Install Pip 3

# aws --version

# Python --version
# aws configure  → AWS Access Key ID : xxx
                   Aws Secret Access Key : xxx
                   Default Region name : xxx
                   Default output format : table
# ls -a
```

here one hidden fill will get created (.aws)

created (s3)

```
# cd .aws

.aws]# ls

Config  credentials

.aws]# cat config

[default]

Output = s → table

.aws]# cat credentials [here the user information stored]
```

[default]

aws_access_key-id

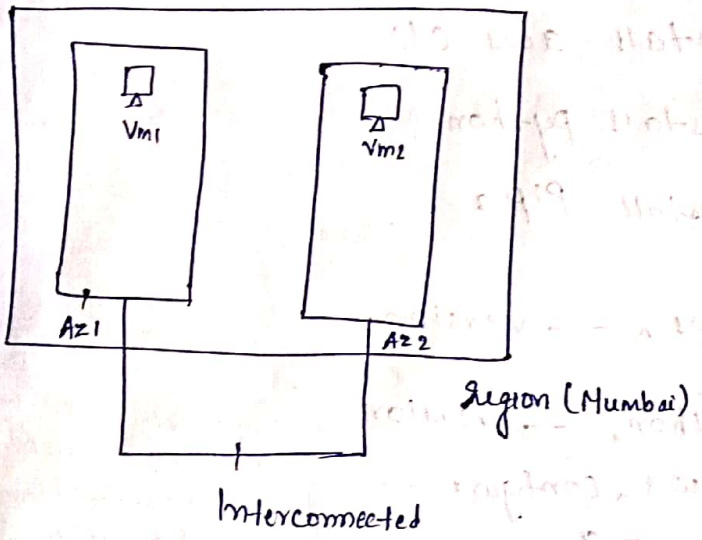
aws_secret_access_key

Basics networks

4/03/19

Networking and content Delivery

Vpc – Virtual Private cloud



Class B

172.25.0.0 /16

255.255.0.0

0 0

⋮ ⋮

254 254

Host per n/w

254 * 254 (machines)

172.25.0.0 /24

255.255.255.0

0

⋮

254

Host per n/w

254

CIDR — Classless Inter-Domain Routing

If you give 32 bit → You will get only 1 machine.

For eg: If we (an admin) a company and we want to connect 150 machines. Instead of wasting all the IPs we can calculate the system numbers by using the CIDR Method.

⇒ 8 . 8 . 8 . 8 / 25

255 . 255 . 255 . 00000000

$2^7 + 2^6 + 2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0$

128 + 64 + 32 + 16 + 8 + 4 + 2 + 1 = 255.

↑ 25 ↑ 26 ↑ 27 ↑ 28 ↑ 29 ↑ 30 ↑ 31 ↑ 32

So total ⇒ 127 machines we can connect.

⇒ 8 . 8 . 8 . 8 / 17

255 . 255 . 0 : 254 . 0 : 254

00000000

0 : 127

128 + 64 + 32 + 16 + 8 + 2 + 1

0 : 127

⇒ 127 * 254 ⇒ 32,254 Machines we can Connects.

⇒ 8 . 8 . 8 . 8 / 18

255 . 255 . 0 : 254 . 0 : 254

00000000

0 : 63 . 0 : 254

128 + 64 + 32 + 16 + 8 + 4 + 2 + 1

0 : 127 . 0 : 63

⇒ 63 * 254 → 16,002

8 . 8 . 8 . 8 / 19

255 . 255 . 00000000
0
:
31 254

128 + 64 + 32 + 16 + 8 + 4 + 2 +

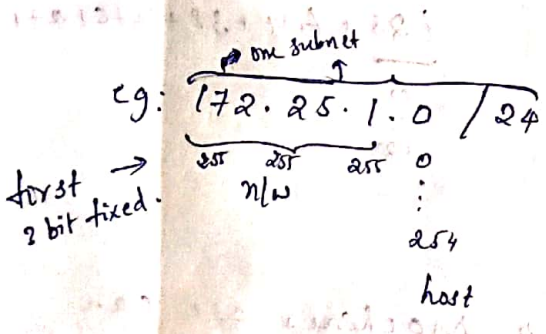
⇒ 31 * 254 ⇒ 7,874 Machines

5/8/19

Subnet

A subnetwork or subnet is a logical subdivision of an IP network.

- Computers that belongs to a subnet are addressed with an identical most significant bit-group in their IP addresses.



⇒ 255 machines we can connect.

172.25.1.0 /25 ⇒ 128 machines

Internet Gateway

- Internet Gateway is a network node that connects to different networks that use different protocols or rules for communicating.

eg: if we have wifi connection at home, your Internet gateway is the modem or router combination that your ISP (Internet service

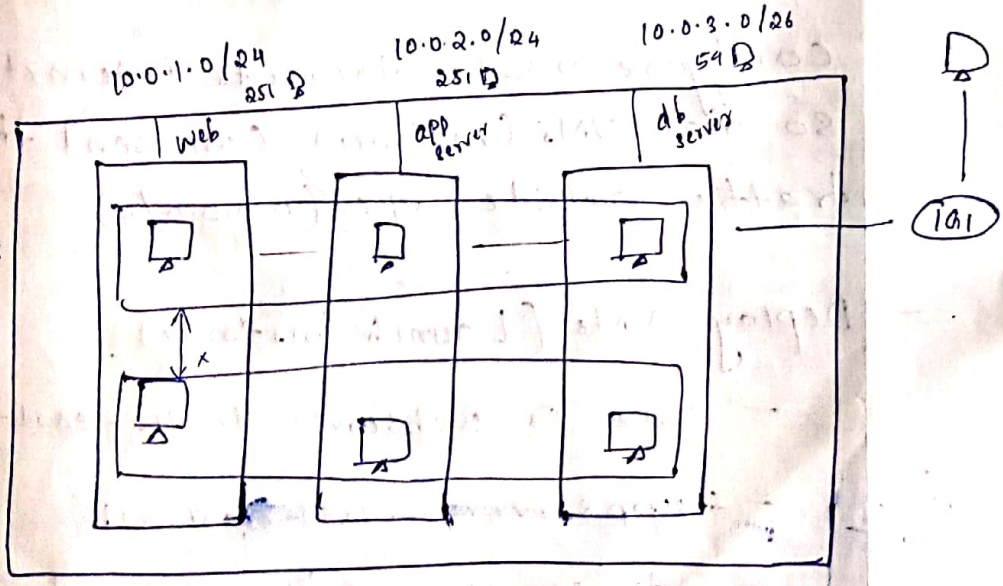
provider) provides so that you connect to the internet to their network.

Route Tables

A route table contains set of rules called routes that are used to determine where network traffic is diverted.

- Each subnet in your vpc must be associated with a route table.
- A subnet can only be associated with one route table at a time, but you can associate multiple subnet in the same route table.

Steps to create VPC



- Vpc is region dependent
- Vpc is not availability zone dependent.

Steps

- create VPC, name it as `MyVPC`
- set IP range 10.0.0.0/16

☑ Enable DNS hostname on VPC

- create subnets as following and enable auto assign IP.

webserver — 10.0.1.0/24 — us-east-1a

app server — 10.0.2.0/24 — us-east-1b

DB server — 10.0.3.0/26 — us-east-1c

Modify auto-assign IP settings

- create Internet gateway and attach to VPC

select route table

- use default route table and add subnets to it.
- configure route towards internet gateway so that VMs (instance) can send the traffic outside VPC (0.0.0.0/0).
- Deploy VMs (Launch Instance)
  - one is webserver in us-east-1a
  - app server — us-east-1b
  - db server — us-east-1c
- and check communication (ping)

(Same) steps for creating VPC

⇒ Networking & Content Delivery

- Click on VPC
- Click on your VPC's
- Click on create VPC

Name tag [MyVPC]

IPV4 CIDR block [10.0.0.0/16]

Tenancy [Default ▾]

- [Create]

⇒ For Enable DNS hostname on VPC

- Select VPC
- Actions → click on Edit DNS hostnames

DNS hostnames ☑ enable

[save]

⇒ Click on subnets (dashboard)

- Click on [create subnet]

webserver
10.0.1.0/24
us-east-1a

Name tag [web server]

VPC [Select myVPC] → what we created just now for VPC

Availability Zone [us-east-1a]

IPV4 CIDR block [10.0.1.0/24]

appserver
10.0.2.0/24
us-east-1b

[Create]

Dbserver
10.0.3.0/26
us-east-1c

- Create for appserver and Db server by using the same steps.

⇒ For auto-assign IP $\rightarrow$ Enable the IPC

- Click on subnet [Do the same for webserver, appserver, dbserver]
- Actions
- Click on Modify auto-assign IP settings

Auto assign IPV4 ☑ Enable auto-assign public IPV4 addresses

Save

⇒ Click on ~~one~~ Internet Gateways (dashboard)

- Click on create internet gateway

Name tag IG1

Create

When we create one VPC, by default one route table is created. That is private only.

⇒ For attaching Internet gateway to VPC

- Click on particular Internet gateway
- Actions
- Click on Attach to VPC

One VPC can connect only one Internet gateway.

VPC * myvpc — select the VPC

Attach

⇒ Click on route tables (dashboard)

- Use default route table and add subnet to it
- Select default route table, and rename it as First RT
- Click on subnet Associations
- Click on Edit subnet association

here we assigning the subnet as public and private

⇒ For auto-assign IP's → Enable the IPs

- Click on Subnet [Do the same for webserver, appserver, dbserver]
- Actions
- Click on Modify auto-assign IP settings

Auto-assign IPV4 ☑ Enable auto-assign public IPV4 addresses

Save

⇒ Click on Internet Gateways (dashboard)

- Click on Create internet gateway
- Name tag IG1

Create

⇒ For attaching Internet gateway to VPC

- Click on particular Internet gateway
- Actions
- Click on Attach to VPC
- VPC * myvpc — select the VPC

Attach

⇒ Click on Route tables (dashboard)

- Use default route table and add subnet to it
- Select default route table, and rename it as First RT
- Click on subnet Associations
- Click on Edit subnet associations

create by default one route table is created. That is private only.

One VPC can connect only one Internet gateway

here we assigning the subnet as public and private

public

No need to add IP address. default it will take the local.

- If we want to make it as public

- click on [Edit subnet association]

- Select the subnet which we want to make as public.

- [✓] subnet ... | appserver

  [✓] subnet ... | web server

[Save]

click on Routes

Click on [Edit Routes]

- Click on [Add route]

	Destination	Target
public	[0.0.0.0/0]	[Internet Gateway] Select the Internet gateway we created

[Save routes]

- if we want to make it as private

- follow the same steps, but we need to create another route table. so click on ① [create route table]

Name tag [second RT]

VPC [myvpc]

[create]

② Again go to Edit subnet association, select the subnet

③ In Routes

- Click on [Edit Routes]

- [Save routes]

No need to add IP address, default it will take the local.

- We need to launch the instances by using the created VPC and subnets. [EC2 instance]

- ~~we~~ Now here we need to launch

3 Instances with the subnets (Web server, appserver and db server).

Steps: Configure Instance Details

Network : myvpc — Select our vpc

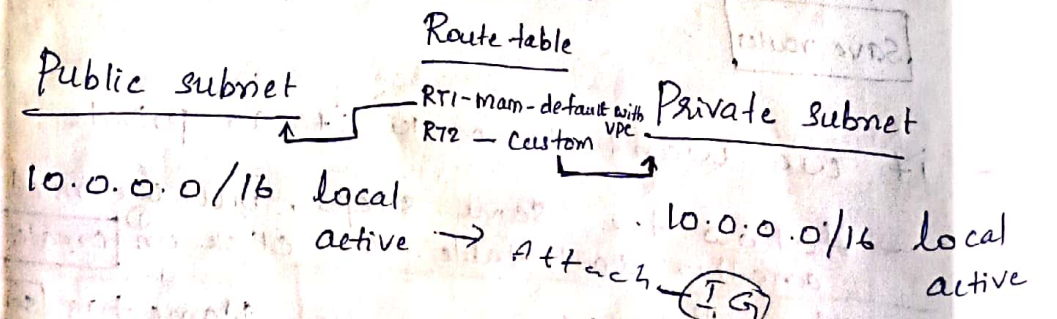
Subnet : Webserver/appserver/dbserver — Select particular subnets.

- After Launching the instance.
- ~~Long~~ logon to the instance and try to ping.

The subnets are in same vpc.

- If the subnet is public, we can logon to the instance and ping it.
- If the subnet is private, we can't logon to the instance.

Public and Private Subnet



add Route

0.0.0.0/0 -

now it will become into Public

6/8/19

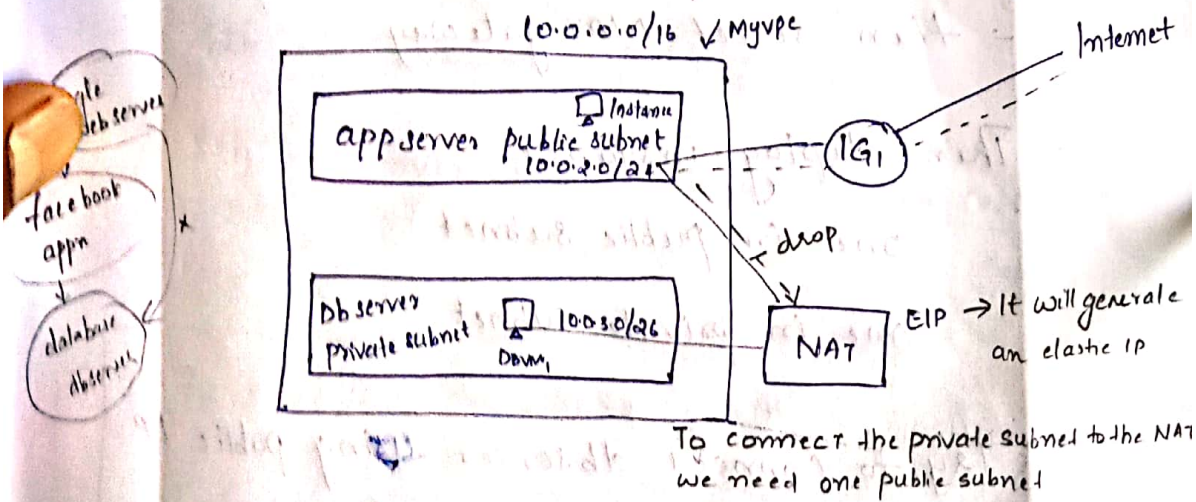
VPC NAT Gateway

Vpc NAT

- we can use NAT device to enable instances in a private subnet to connect to the Internet.

eg: For updating software, aws resources

- But it prevent the internet from initiating connections with the instances.
- A NAT device forwards traffic from the instance in the private subnet to the internet or other aws services.
- A NAT device contains an elastic IP. It supports only IPv_4 address.
- It doesnot support IPv_6 address



Steps to create NAT Gateway

- Create a vpc and enable hostname.
- Create subnet as following

appserver	10.0.2.0/24
db server	10.0.3.0/26

- And make it the appserver subnet as public and connect to internet ~~way~~. gateway (IGW)
  - Edit Routes → 0.0.0.0/0 Internet gateway
- Make the dbserver as a private subnet
  - Create one secondary route table and name it as second RT
- Create NAT gateway using public subnet
  - Click on create elastic IP
- And attach private subnet to NAT gateway (Using Route table)
  - Edit Routes → 0.0.0.0/0 ~~NAT~~
  - then select NAT gateway
- Then diploy instance
  one in public subnet
  one in private subnet
- Try to connect dbserver using public IP and it should fail. (private)
- Logon to VM1 (public subnet Instance) and from this connect to ~~database~~ dbserver (private)
  - Copy pem file from your laptop to ~~public subnet~~ laptop.

- change the permission to 400

chmod 400 pem file

- Then login ssh ec2-user@ public IP

- After login ping to outside.

ping www.google.com

- login as: ec2-user

sudo -i

Vi NVKey.pem

chmod 400 nvkey.pem

ls -l NVkey.pem

~~ssh ec2-user@ IP address or~~

~~ssh DNS~~

~~ssh -i "NVkey.pem" public DNS~~

ssh -i "nvkey.pem" ec2-user@ public DNS

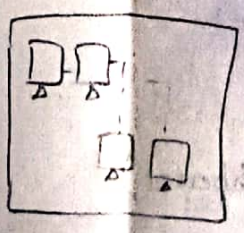
(copy)

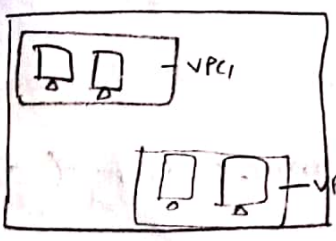
⇒ To see the commands for (ssh)

- click on the particular instance

- click on connect

VPC



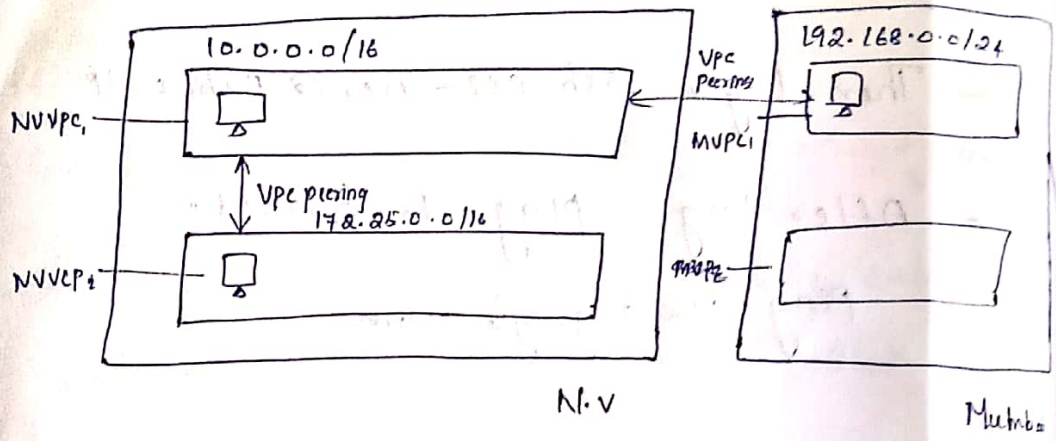


7/03/19

VPC Peering

Public IP we are using to connect the instance.

For pinging we are using private IP



Steps:

- Create 3 VPC, 2 in north virgina and one in ohio.
  - all VPC should have different IPs.
- Create ~~insta~~ one server in all 3 VPC.
- Ping between this servers using private IP and it must failed.

Peering connection

- Create a peer connection between NVVPC1 and NVVPC2 which is peering inside the same region.
- Change the route table on both VPC (VPC1 and VPC2) and check the connectivity from both the servers.

Steps to create peer connection

- VPC
- click on peering connections (dashboard)

- click on [create peering connection]

- Peering connection name tag [nvvpc1 to vpc2]

- Select a local VPC to peer with

VPC (Requester) [nvvpc1] (which VPC we want to send request)

- Select another VPC to peer with

Account ◉ My account
○ Another account

Region ◉ This region (us-east-1)
○ Another region

- VPC (Accepter) [nvvcp2]

- click on [create peering connection]

- After that select the peering connections

- click on Actions

- Accept request, [Yes, Accept]

- Then for go route table, click on nvvpc1 route table

- Click on Routes

- Edit Route

- Add Route

Destination	Target
nvvpc2 IP address [172.25.0.0/16]	[peering connection]

- : Do the same step for 2...

also...

- Then go to instance and ping it.

VPC peering between Two Regions

Region : North Virginia

- VPC
- click on peering connection
- follow the same steps :- nVVPC1 to ohioVPC
  - VPC (Requester) nVVPC1
  - Account ◉ My Account
    ○ Another (Another Person account)

Region ○ This Region
◉ Another My account another region

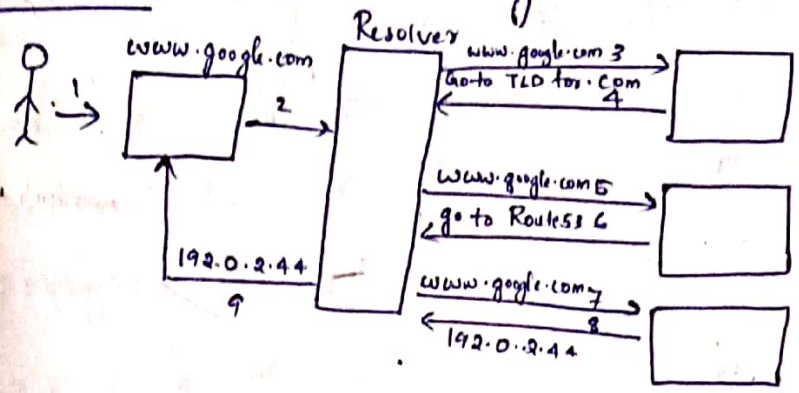
Go to ohio VPC, copy the VPC ID

- VPC (accepter) Copy the VPC ID of ohio account

- click on create peering connection
- Then go for ~~route table and make~~
- Then go for another region (ohio)
- Accept the request.
- Again go for north virgina
  - make the changes in route table
- Go for ohio
  - make the changes in route table
- Then go for instance and ping (ohio and n.v)

14/3/19

DNS – Domain Name System / server



If we search in our laptop 'www.g...com' the DNS will convert the domain name to IP address.

A – IPV4

AAAA – IPV6

In aws Route53 is converting the domain name to IP address

↳ Authorized server → .tk

port no

Tcp/udp – 53

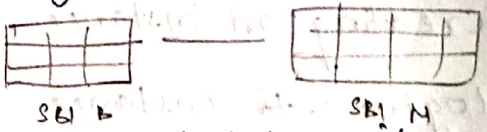
- Domain name system is converting domain name to IP address.
- DNS is a hierarchial distributed naming system for machines connected to a network.
- Route53 is a DNS service of AWS.
- Route 53 is highly available and scalable.
- Route53 is the port number of DNS. 53 we are indicating for TCP & UDP.
- Route53 is a global service
- Route 53 is pay per use.

12/8/19

Database

- Database is a collection of information that is organized so that it can be easily accessed, managed and updated.
- There are different kinds of databases
  (i) Relational database
  (ii) Non-Relational database
  (iii) data warehousing

1. Relational



- It is a collection of database items organized as a set of formally described tables from which data can be accessed or reassembled in many different ways without having two reorganized the database table.

Information are stored in the table format, is called relational database

- Amazon have a service called RDS (Relational database service).
- It including ~~since~~ 6 different dbs:
  (i) Mysql
  (ii) Mariadb
  (iii) Microsoft SQL
  (iv) Postgresql
  (v) Oracle
  (vi) Aurora

2. Non-Relational

- It is any database that doesnot follow the relational model, provided as by traditional relational database management systems.

eg: dynamodb

- Datawarehousing

- A datawarehouse exists as a layer on the top of another database.

eg: Redshift.

Steps

1. Creating an instance

- login into instance
- After login install package.

```
# yum install mariadb* -y
# systemctl restart mariadb
# systemctl enable mariadb
# mysql -u root
# show databases; → will get the default
```

2. Creating an RDS database

- click on RDS
- click on create database
- Select engine
  ⊙ mariadb ○ mysql ○ oracle
- click on next
- Click on ⊙ /dev/test-mariadb
- next
- free tier only
- db-t2-Micro

- db instance identifier [mydb]
- Master username [root]
- master user password [***]
- next Public accessibility ◉ No
- database name [dbrose]
- click on create database

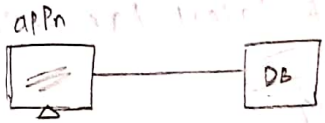
⇒ login to the instance

mysql -u root -p -h endpoint → Click on RDS, Click the mydb
show databases; Copy the endpoint
use dbrose;

title

13/3/19

Non-Relational DB



2-tier application

Dynamo DB

Step-1: -Create an instance (it should be Amazon Linux)
- login into the instance

yum install httpd -y
yum install restart httpd
yum install enable httpd
yum install php-mysql -y
yum install php -y
yum install mysql -y

Step2 (In browser)
- goto wordpress.org website
- WordPress – official site → This page will come

- click on get wordpress
- copy link location of download.tar.gz

Download.tar.gz → right click → copy link location

- come back to instance and run a command to download that above file.

```
# wget https://wordpress.org/latest.tar.gz (paste the link location)
# ls
   latest.tar.gz
# gunzip latest.tar.gz
# ls
   latest.tar
# mkdir /king
# tar -xf latest.tar -C /king   (-C: to specify particular file)
# cd /king
# ls
   wordpress
# cp -r wordpress/* /var/www/html/
# cd /var/www/html
# ls
   index.php   wp-includes ....
   ....
# chmod 755 -R wp-content
# chown apache:apache -R wp-content
# ls -ld wp-content
```

Step 3: (Creating an RDS database)

- create database
- select engine Mariadb

- next
- free tier only
- db.t2-micro
- db instance identifier → mydb
- Master user name → root
- Master user password → jesus123456
- next
- database name → dbrose
- click on create database

Step-4

- copy the public ip of instance and browse it browser
- click on lets go
  - → database name - dbrose
  - → master username - root
  - → user password - jesus123456
  - → DB host - copy <end point of RDS>
  - → Table prefix - wp_

Submit

- It will go to another page → wordpress > setup configuration
- Copy the content (copy the phpcode)
- goto instance (cmd page)

```
# cd /var/www/html
# vi wp-config.php
    paste the content
:wq!
```

systemctl restart httpd

→ Start the Installation

→ Again go back to the old page

- Page → wordpress setup configuration

-

- Click on Run the installation

- Another page:

Information needed

Site Title : Pragathi

Username : root

password : * * * * → here defaulty one encrypted password will generate. we need to copy it to notepad for first...

Your Email : roshlinarm@gmail.com

[Install word Press]

→ page ⇒ wordpress > Installation

Already Installed

[LogIn]

→ page ⇒ LogIn < pragathi — wordpress

Username or Email Adress

[roshlinarm@gmail.com]

password

[paste the password] open the notepad and copy the password

- page ⇒ Dashboard < pragathi — wordpress

systemctl restart httpd

→ Start the installation

→ Again go back to the old page

- Page → wordpress setup configuration

-

- Click on Run the installation

- Another page:

Information needed

Site Title : Pragathi

Username : root

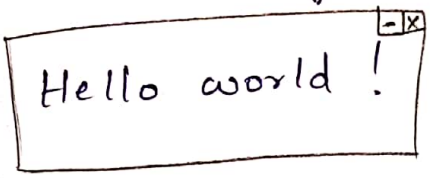
password : * * * * → Here defaulty one encrypted password will generate. we need to copy it to notepad for future purpose

Your Email : roshlinamm@gmail.com

Install word Press

Customize your site

→ page ⟹ Customize Pragathi – just another WordPress site



→ browse the Instance ip in browser
→ if we want to do some changes in that site
. login with above username and password

} if any changes, we want to make, then do this step.

14/3/19