6th Nov

Cloud Computing :-

* Cloud computing is the on-demand delivery of compute power, database, storage, servers, N/w ing, software analytics and more resources through a cloud service platform via the internet with pay as you go prices.

* Companies offer these computing services are called cloud providers and typically charge of cloud computing services based on usage similar to how you are billed for water, electricity at home.

Cloud Service Models :-

→ IaaS (Infrastructure as a Service)

* You rent infrastrutre - server, storage, N/w, VMs, operating system from cloud provider on a pay-as-you-go basis.

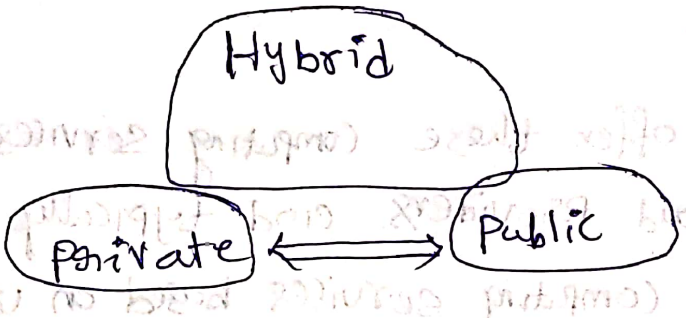
→ PAAS (Platform As a service)

* supply an on-demand environment for developing, testing, delivering or managing S/w applications.

* SAAS (Software as A Service) :-

→ It is a method of delivering s/w applications over the internet on demand and typically on subscriptions.

Cloud Deployment Models :-



Models :-

Public Cloud :-

* Computing resources are provided publicly based on a pay-per use model.
* Public clouds are owned and operated by a third party cloud service provider.

Private Cloud :-

* The cloud infrastructure is owned by an organization and hosted and operated initially.

* A Private cloud can be physically located on the company's on-site data center.

Hybrid Cloud:-

* Computing resources are provided by a composition of private and public clouds.

* Hybrid cloud gives businesses greater flexibility and more deployment options.

7th Nov. AWS (Amazon Web Service's)

AWS is a secured cloud service platform, offering compute powers, Database, storage, content delivery and other functionality to help business scale and growth.

Benfits :-

→ Easy to use. → flexible. → secure
→ Reliable. → Cost effective → Global leader.

Aws Global Infrastructure

- The AWS Cloud operates Availability zones (AZ's) with in the Geographic Region around the world.

Region:

- It is a physical location in the world where we have multiple AZ's.

Availability Zone :-

→ It consist of one or more discrete data center each with redundant power, n/w and connectivity based in seperate facilities.

→ The AZ's offer you the ability to operate AWS service (Remain below).

8/11

AWS Free Tier (12 Months) :-

* AWS is offering a free usage tier for new AWS customer

* Amazon EC2 instances with 750 Hrs usage of Linux and 750 Hrs usage of windows (t2 micro.)

* Amazon Light-sail 750 Hrs.
* " RDS 750 Hrs.
* " DynamoDB 25GB.
* AWS LAMBDA 1 Million.
* 30GB of amazon EBS in General Purpose (SSD) (or) Magnetic.
* 1Gb for snapshot storage.
* 5GB for amazon S3 standard storage.
* Expire 12 Months in signup.

Data Center :-

→ A data center is a facility the centralizes an organization IT relation and equipment as well as where it store manage and discreminates its data.

* aws.amazon.com. (Account (a/c) Creation)

In the ac.

* Create a new Aws a/c. then

Aws Services :-

Compute

* → EC2 (Elastic Compute Cloud)
  - → LightSaile.
  - → ECR (Elastic Container Register)
  - → ECS (" " service)
  - → EKS (" Kuberneticeg ")
  - → Lambda
* → Elastic Bean Stack.
  - →

Storage.

* → Instant Store.
* → EBS (Elastic Block storage)
* → S3 (simple storage service)
* → EFS (Elastic File System)
* → FSx (Third Party File system)
* → S3 glaciar.
* → storage gate way

Data Base :-

* → RDS (Relational DB service).
  → Dynamo DB.
  → Elastic cache.
  → Amazon Redshift.
  →

Networking & content Delivery :-

* → VPC (virtual Private Cloud)
  → Cloud front.
* → Route 53.
  → Direct Connect.
  →

Management & Governance :-

→ Aws organizations.
* → Cloud watch.
* → Aws auto scalling.
* → Cloud formation.
  → Cloud trade.
  → Systems manager.

Security, Identity & Complaint :-

* → IAM (Identity Access Management)
* → Directry service.

Application Integration :-

* → Simple notification service (SNS)
  → simple Queue service (SQS).
  → Simple email service (SES).

Amazon EC2 :-

→ Amazon Elastic Compute Cloud (Amazon EC2) provide scalable computing capacity in the amazon webservices (AWS) Cloud.

→ Using EC2 eliminates your need to invest in H/w upfront, so you can develop and deploy application faster.

→ Amazon EC2 provide scalable virtual server in the cloud.

→ The virtual servers can run different OS, but most commonly run a flavor of Linux and windows.

Amazon EC2 Features

* Virtual Compute Environment (Instances).
* Pre-Configured images (AMI's).
  → Amazon machine Images.
* Configuration of various CPU's, memory, CPU, N/w ing (Instance Types).

Imp * Storage volume for temporary data (Instance Store Volumes)

Imp * Persistant storage Volumes for data (EBS)
→ Elastic Block Storage.

* Multiple Physical locations (AZ's)
→ Availability Zone.

Imp * Firewall rules using Security Groups (SG's)

Imp * Secure login information for your instances (Key Pairs).

* Meta Data known as Tags.

Imp * Static IP-address for dynamic cloud computing (EIP).

* Logically isolated Networks (VPC)
Virtual Private Cloud.

EC2 Purchasing and Pricing

* on-demand instances:
  → Paid for by the second with no long-term commitment.
  → Low cost and flexible.

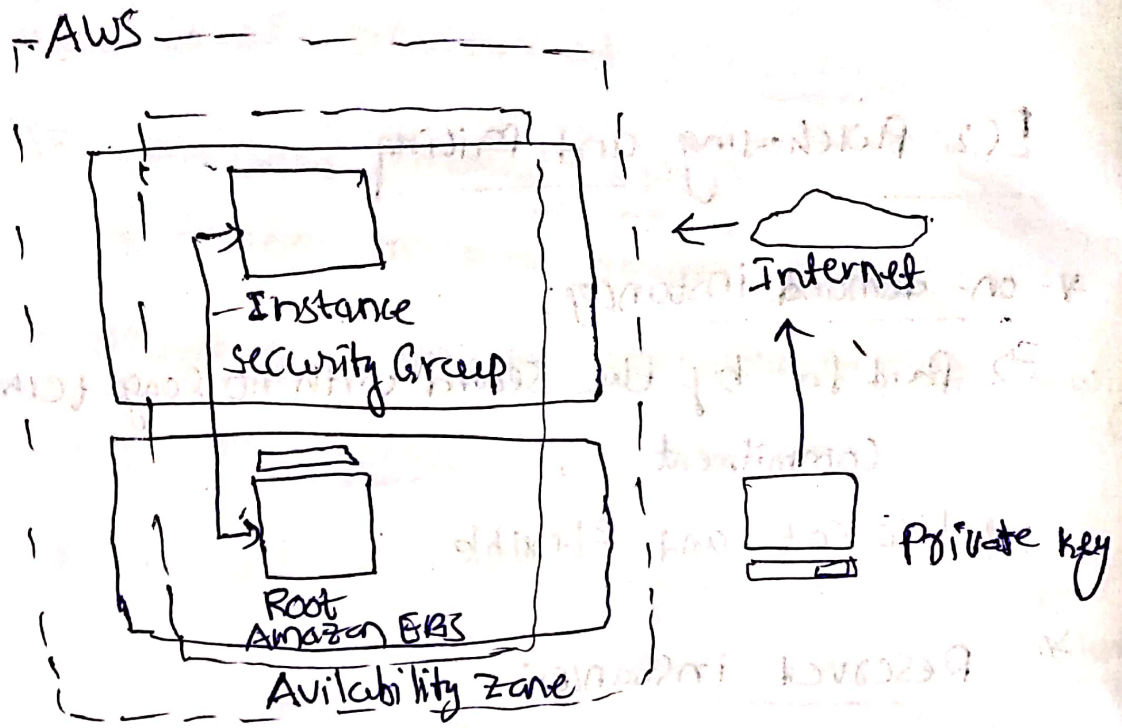
* Reserved instances:-
  → Lower per second pricing based on up front time commitment (or) payment.
  → Provides persecond discount upto 75%.

* Spot Instances:-
  → Per second based on bid, preemptible.
  → It fluctuate based on supply and demand of avilable unused EC2 capacity.

* Dedicated Hosts:-
  → Provide dedicated server for customer. No other customer instance run on dedicated server.
  → Pricing is per Hour.

Amazon EC2 Instances:

12/4



12/4

Launching windows instance

→ click on EC2

→ click on Launch instance.

→ select microsft windows server 2019.

→ Choose instance type.

→ select t2.micro (Free)

→ Click on Config. instance details.

9.3

→ Config instance details no. of instance 1,

→ Then click on add storage.

→ click Add Tags. Then click to add a name Tag.

→ Add Tags

key	value	instances	Volumes
Name	WIN2K19-SERVER	☑	
Development			

→ click configure security group.

security Group name : MYSG

Description: MYSG create.

Type	Protocal	Port Range	Source	Description
RDP	TCP	3380	Anywhere	Remote Desktop Protocal.
HTTP	TCP	80	Anywhere	

→ click Review & Launch. Then Launch option.

* Instance group.
* key Pairs.

→ metadata → collection of instance details.

* Imp.
→ Instance IP address

* Public IP address :-

A Public IP address is required if you want the EC2 instance to have direct communication with resources across the open internet.

* Private IP address :-

The Private IP address is used for internal communication b/w instance ~~within~~ with in the VPC.

Connecting windows Server :-

→ Go to local windows machine type RDP.
→ Copy Instance Public IP address and Paste in remote desktop Connection..
→ Username : Administrator.
Password :

→ select instance click on connect option.

→ click get Password.

→ choose file under download keypair. Decrypt P/w.

Stop / Reboot / Terminate Instance :-

→ select instance → Actions in the instance state terminate.

* Instance:

In amazon terms, it means a virtual machine/server which created by you in Aws data center (cloud).

13/11

Launching Linux instance

→ Go to EC2 instance.

→ select Redhat 8.

→ t2.micro free tire, then click Config. instance.

→ click on add storage.

→ click Add Tags.

→ In Add Tags:

Key	Value.
Name	RHEL8-Server
Development	Name1.

→ Config security Group.

SSH	TCP	22.
HTTP	TCP	80.

→ Then Review and Launch and keep launch option.

→ Existing keypair if required.

* Connecting EC2 Linux from Linux:

S:1
⇒ First Create Sample (our name) keypair.

Cat > sample.Pem (Command)

S:2
⇒ To change key Pair permission.

[root@server] # ls-l sample.Pem

[" "] # chmod 400 Sample.Pem

5:3

⇒ connecting Instance.

ssh -i (identity file) sample.pem

ec2-user @ Linux-IPaddress.

→ $ (EC2-User) ⇒ Developer

(root) ⇒ Administrator.

⇒ Exit command to Exit.

Connecting Linux instance from windows.

→ Download and install Putty and Puttygen s/w's.
Putty.org. (website name).

→ Download 64 bit s/w.

→ Puttygen.exe 64 bit s/w.

⇒ Puttygen s/w :-

It helps to save keypairs with required format.

→ The .Pem stands form Privacy enhanced mail.

→ .PPK stands for Putty private key.

⇒

→ converting .pem format to .ppk with help of putty gen.

→ Then open puttygen select load option choose sample file.

→ Save private key.

→ connecting instance,

* Open Putty copy instance IP address paste Hostname side.

* Expand SSH select authentication (Auth) then browse.

* In browse choose sample.ppk.

*⇒ Command : ec2-user

[ec2-user@IPaddress] $ sudo -i.

14/11

WEB SERVER

* A webserver is a n/w service that serves content (web pages) to a client over the web. webserver also known as HTTP (HyperTex Trans. protocol) servers

* The web server avilable in RHEL8 are:
  - → Apache HTTP server.
  - → nginx.

* An Apache open source server developed by the Apache s/w Foundation.

* Apache s/w is solid and stable server that has been around for years.

* SSL protocol safe and secure.

Imp Main Configuration Files

→ Packages : httpd, mod_ssl, openssl.

→ weblocations : /var/www/html.

→ Configuration Files : /etc/httpd/conf.d/ssl.conf/httpd.conf
/etc/httpd/conf.d

→ Default web page : /etc/httpd/conf.d/welcome.conf

→ Modules : /usr/lib64/httpd/modules.

→ Log Files location : /var/log/httpd
access-log & error-log.

→ Ports : http-80, HTTPS-443

→ service and Daemon : httpd

Notable changes :-

#service httpd graceful #apachectl graceful

service httpd configtest # apachectl configtest

→ Go to EC2 Launch Instance.

→ select OS as RHEL8.

→ Then click on Config inst. details.
No.of inst. 1.

→ Click on add storage. Capac. 10gb.

→ Add Tag. to enter Name.

→ Create New SG.

SSH 22

HTTP 80

HTTPS 443

→ Then click on Review and launch.

→ select key pair sample they launch instances.

→ Connecting server

sudo -i.
sudo su

→ Copy Public IP address

→ Paste ~~Host~~ name.

[root @ ip-] # cat /etc/redhat-release

[root @ ip-] # uname -r

[" "] # arch.

[" "] # cat /etc/release #. (complet OS details)

S:1 Installation and configuration

[" "] # yum update -y.

To update server command

[" "] # yum install httpd -y.

[" "] # httpd -V. (version)

[" "] # systemctl status httpd

[" "] # systemctl start httpd

[" "] # " enable "

[" "] # " " status "

→ By default Apache loads test Page
(/etc/httpd/conf.d/welcome.conf - file)

→ Go to webpage location.

[u] # cd /var/www/html/

→ [u] # ls

[u] # cat > index.html

```
<html>
<body bgcolor=red text=blue>
<marquee> <H1> Hello... </H1> </marquee>
</body>
</html>
```

Ctrl+d (Save file)

```
[root html] # ls
index.html
[u html] #.
```

Next verifying Log files :-

[root u] # cd /var/log/httpd/

[ ] # ls

~~access_log~~ error_log
access_log

[" "] # tail -f access_log.

[" "] # tail -f error-log. (server side)

15/4

Security Groups (SG):-

→ security Groups act as firewally that controlls ~~Profit~~ traffic for one (or) more instances.

→ AWS SG are Associated with EC2 instances and Provide security at the Protocol and port access level.

→ The rules of SG control the inbound traffic that's allow to reach the instances.

→ By default SG allow all outbound traffic.

→ SG rules are always permisive, that mean we can't create rules that deny access.

→ SG are statefull, that mean if you send a request from instance, the response traffic for that request is allowed to flow in regardless of inbound SG levels.

Creating Security Group :-

→ Go to EC2 dashboard, select SG

→ Then create SG.

Inbound

SSH	22
HTTP	80
HTTPS	443

→ Create rule

Key Pair :-

A key pair is a combination of public and private keys.

→ Amazon EC2 uses public key cryptography, to encrypt and decrypt log in information.

→ Public key cryptography enable to securly accesing instances using a private key instead of a password.

→ The public key is located in the instances uses home directory. (~/.ssh/authorized_keys)

authorized_keys.

→ creating a key pair

→ You can use amazon EC2 to create a key pair.

→ Alternatively, you use a third party tool and import the public key on amazon EC2.

Creating a key by puttygen.

→ open putty gen.

→ click generate

→ click save public key. (to save "mykey_pub")

→ click on save private key. (save as "mykey.pem")

→ Import public key from amazon account.

→ Go to ec2 dashboard, click on keypairs, then select import keypair.

Launching instance :-

→ Go to instances launch instance.

→ In step 6 SG settings select my SG.

Connect instance :-

Network Interface :-

→ An elastic N/w interface (refer to as a n/w interface) is a logical N/w ing component in a VPC that represents a virtual N/w card.

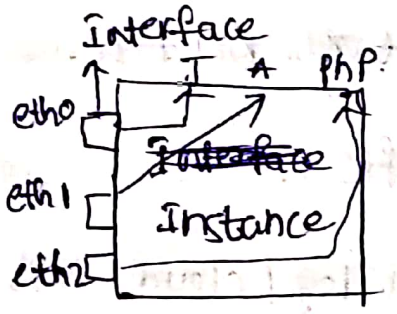
→ The term "elastic N/w interface" is sometimes shorted to "ENI".

→ All N/w interface have eni-xxx resource identifier.

→ A n/w interface can include the following attributes.

* A primary private IPv4 address.
* one elastic IP address (IPv4) per private IPv4 address.
* one public IPv4 address.
* one and more IPv6 address.
* one & more SG's.
* A source/destination check flags.
* A description.
* A MAC address.

private key



→ N/w interfaces

→ create N/w interface the discription is My-NIC.

→ Select "usest-1c".

→ select N/w interface then attach, select instance then attach.

* Command if config [# if config].

18/11

Imp 2nd TOPIC EC2 :-

Bootstrapping & User-data | Meta-data

Bootstrapping :- (Running commands on your linux instance at Launch)

* Refer to a self starting process (or) set of commands without external i/p's.

* With EC2 we can bootstrap the instance with custom commands. (such as installing packages, running instances updates and config. varies settings)

* Commands in text (or) file.

Bootstrap verific. :-

* /var/log/cloud-init.log.
* /var/log/cloud-init-output.log.
* Select instance --> instance settings --
  > Get system log.

→ Go to AWS A/C.
→ Click on EC2.. (Virtual server environment) create instance.
→ Launch instance.
→ select Redhat 8.
→ Choose instance type. (select t2.micro)
→ Then click on Config. instance details.

→ S:3:- Advance details,

```
#! #/bin/bash.
yum update -y
yum install httpd -y
systemctl start httpd.
system ctl enable httpd.
echo "welcome .." >> /var/www/html/index.html.
```

→ select SG, web security group.

→ click on launch instance.

→ copy IP address and open it in browser see on "welcome.. ".

User Data / Meta_data :-

→ Instance meta data is data about your instance that you can use to configure (or) manage the running instance.

→ EC2 instance can also include dynamic data, such as instance identity document that is generated when instance is launched.

Retrieving Instance Metadata :-

* $ http://169.254.169.254/latest/meta-data/
* $ curl http://169.254.169.254/latest/meta-data/
* $ GET http://169.254.169.254/latest/ " /.

Imp
* To verify system log information select Instance action, Instant settings, And choose Get system Log.

* verifying through command Line

→ connect instance.
→ use putty here.
→ verify log details

[root@ IP.add]# cat /var/log/cloud-init.log.

cat /var/log/cloud-init-output.log.

To verify User Data :-

→ Instance settings, view/change User data.

Imp NOTE :- To edit ~~your~~ instance your User data first ~~you~~ need stop instance.

Imp → How to User data thorugh command line.

[root @ n] # curl http://169.254.169.254/latest/User-Data.

~~To Ver~~

To verify Meta data:-

→ #curl http://169.254.169.254/latest/meta-data/.

To verify dynamic data:-

[root@]# curl http://169.254.169.254/latest/dynamic/

[root@]# " " " /instance-identity/.

19/11

Amazon Machine Images (AMI's):-

→ An Amazon machine image (AMI's) is a master image for creation of virtual servers (known as EC2 instance) in the Amazon webservice environment.

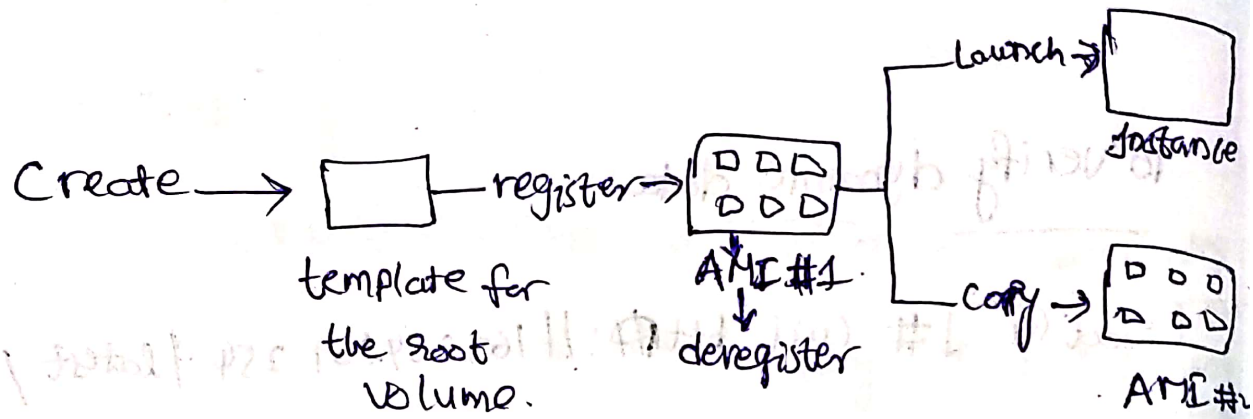
→ AMI's are used with auto scalling to quickly launch new servers ondemand, and to quickly recover from disaster.

→ An AMI includes.

* A template for the root volume for the instance.
* Launch permissions.

→ Pre configured Images (AMI's)

Imp. AMI Life cycle :-



Launch permission :-

Public :- The owner grants permission to all AWS A/C's.

Explicit → The owner grants permission to specific AWS A/C's.

Implicit → The owner has implicit launch permission for an AMI.

AMI's comes in three categories :-

Community AMI's

→ Free to use.

→ Generally, with these AMI's you are just selecting to OS u want.

AWS Market Place AMI's:-

→ Pay to use.

→ Buying, selling and sharing AMI's.

→ Generally, comes package with additional, licensed s/w.

My AMI's:-

→ AMI's that you create yourself.

Imp AMI Virtualization:

→ Virtualization is process of creating a s/w based, (or) virtual, representation of something, such as virtual app's., server, storage and N/w's.

→ Virtualization for EC2 is running the xen Hypervisor s/w.

Types

→ HVM AMI's (H/w virtual Machine)

* It provides the ability to run an O/S directly on top a virtual machine without any modification, as if it run on bare-metal H/w.

→ PV AMI's (Para Virtual)

* Guest can run on host hw that does not have explicit support for virtualization.

→ Create webserver.

→ Launch instance.

→ Amazon linux.

→ HTTP, SSH, 80, 22 port.

→ Linux server to install.

Connecting linux server :-

→ open Putty Connect server.

→ $yum update -y. (Update command).

→ Install Httpd.

$ yum install httpd -y.

→ # cd /var/www/html (Create webpage)
ls.

→ # cat > index.html.

<html code>

→ # systemctl start httpd.

→ # systemctl enable httpd.

→ # history -c. (Clear history)

Creating Image (AMI)

→ select instance, in the actions Image → create image.

→ Image name : web-server

→ Then create Image.

→ Launch instances based on image.

→ Launch instance, select My AMI, Choose webserver. Next continue steps.

→ TO change Image permission

→ select Image files, in 'actions'

→ 'Permissions' edit.

TO set selected permission (Explicit)

→ Select Private, add a/c number.

Imp Copying Image :-

→ 'Actions' → copy AMI.

→ Go to selected (mumbai) region and verify it.

→ Action → Deregister.

20-11

AWS command line Interface (CLI)

→ AWS command line interface (CLI) is undefined tool to manage your AWS services. with just one tool to download and configure, you can control multiple AWS services from the command line and automate them through scripts.

Pip install Python (PIP):-

→ It is recommanded method of installing aws CLI on Linux which is Python based tool helps in install, upgrade and remove Python Packages.

Requirements:-

→ Python

→ windows, Linux, mac OS and Unix.

CLI installing on windows

S.1

→ Search on aws cli windows.

→ Install the AWS CLI version 1 using the MSI installer.

→ command Prompt "aws -- version".

S.2 : Generate Access key :-

→ Before the AWS CLI can be configured to use, you need to create a user with the required permissions and download the access key. (Aws access key id and Aws secret key) for use in the AWS CLI.

→ In Aws a/c, click on a/c name, Below click on my security credentials.

→ select continue to security credentials.

→ Expand access and create new access key download it.

→ root key download access & secret key id.

→ command line "aws configure".

→

O/P formats:-

→ command line

cd .aws

ls.

cat .\config

cat .\credentials.

- JSON → Javascript object notation.
- Text
- Table.

→ aws help

Syntax:-

aws [option] <command> <subcommand>

→ Search ⇒ aws CLI ~~config~~ command Refere. [Parameters].

→ aws ec2 describe regions --output=text

--output=Table.

→ Creating bucket (s3 concept)

→ aws s3 mb s3://aws-avi. (Create bucket)

→ aws " rb " " " (remove ")

AWS CLI Linux (or) mac OS :-

→ Search aws cli Linux.

→ Download SI ~~Cw~~. , click bundle installer.

→ unzip file.

→ aws --version.

→ aws --configure.

21/11

IAM (Identity Access Management)

→ AWS IAM enables you to securly control access to AWS service and resources for your user and groups.

→ Using IAM, you can create and manage AWS users and groups, and use permissions to allow and deny their access to AWS resources.

→ IAM is global to all AWS regions, creating a user a/c will apply to all regions.

→ All users beside the root user, users must be created with proper permissions.

IAM Policies :-

→ A policy is a document that formly state one (or) more permissions.

→ By defaulit, an explicit deny always overrides and an Explicit allow.

→ IAM provide pre-built policy template to design to users and groups.

→ you can also create custom policy using the policy generator from the scratch.

~~→ Go IAM~~

Security Identity Compliants

→ Create new group. [dev-team].

→ Select Policy 'S3' full access and LAMBDA Policy.

→ Click on next and create group.

→ Create new group : [Prod-team].

→ Select VPC [Amazon VPC fullaccess]
Route53 full access.

→ Create group.

Users:-

→ Add User :- Uname : Sara
Uname : Max

Access type :
Aws console management.

Console Pwd :

Require Pass cd : ☑

→ go to Next step.

→ select dev-team, click on tage, review and create.

→ create Another 2 Users.

→ Add User uname : Ramu
: Ramesh.
: Ravi

→ Produtian team (Pro-team).

→ Go to tags, Review then create users.

— — — —

→ create new user, Add User.

Uname : Raju

Assign Pwd : own Passwd.

→ Attach existing Policy

Attach EC2-full access.

→ Click on Tag and review create User.

→ From the IAM dashboard Copy IAM User sign in link then paste it.

→ Sign in raju User access EC2 service only

others are not access (Access deny).

IAM Roles :- Imp

→ An IAM role is an IAM entity that defines a set of permissions for making AWS service requests.

→ IAM roles are not associated with a specific user and group.

→ Roles for EC2 instances. An EC2 instance can have only one role attached at a time.

→ Go to roles, click on roles.

→ create role, select EC2, Then click on Permission

→ EC2 full access, S3 full access

→ Role name : EC2-S3-FACCESS

→ Then create role

→ Create another role, setup VPC (Select VPC)

→ Allow site click go to next.

→ click on tag review, create role.

→ Go to EC2 launch instance select any one linux

→ S:2 ⇒ IAM role : EC2-S3-FACCESS

→ Remain same steps, launch instance.

→ To change existing role, select instance, In the action, Instant setting.

→ Then attach and replace IAM code.

Imp Multi factor Authentication :-

→ It adds an extra layer of protection on top of your uname and password.

→ when user signin to AWS website, they will be prompted for their uname and password.

→ you can enable ~~AWS~~ ~~a/c~~ MFA for you AWS a/c and for individual IAM users.

NOTE :-

→ Each identity has its own MFA configuration.

→ Go to IAM dashboard, click on activate MFA root a/c, manage MFA.

→ continue security credentials choose MFA

→ Activate MFA, select VFA virtual (1st option).

→ Virtual MFA device.

→ MFA deactivate :-

→ Click on MFA, manage MFA in action manage now Remove.

MFA for IAM Users

→ ~~Users~~ Click user, from security credentials

→ assign MFA device.,

→ click on manage setup MFA device.

22/11

Creating Policies

→ IAM policies.

→ Create policy.

→ Visual editor Choose service. Select S3.

→ under action select list and Read.

→ click on ~~servo~~ resource.

→ select all resource. Then review Policy

→ S3_List-Read only.

→ Then 'create' policy.

→ Filter Policy ⇒ Custom policy. select S3-List-Read.

→ policy action attach, click name Raju.

→ create new Policy for EC2

→ create Policy, choose a service

→ Type EC2, In Actions select Read, list, Tagging

→ select Resources, specific resource.

→ Review policy. EC2-List-Read-Tagging.

→ Then create policy.

Option2: creating Policy:-

→ create Policy, select import managing Policy.

→ verify JSON code, Review Policy

→ Policy name: S3Deny.

→ Then create Policy.

Create Policy another method :-

→ copy code Paste ~~hear~~ Here.

Imp

Cross Account Access :-

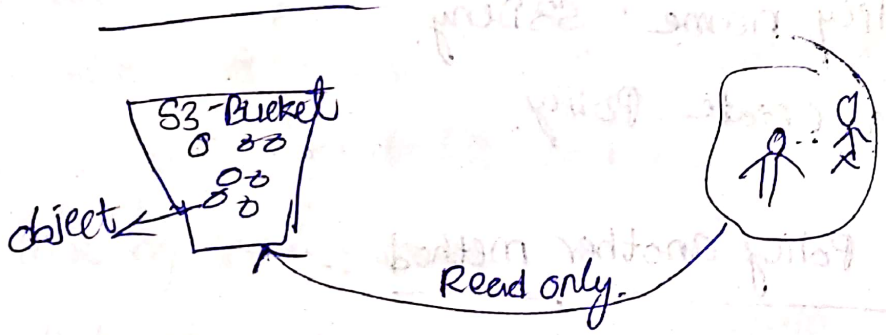
IAM security Token service (STS) :-

→ STS allow to temporary security ~~services~~ credentials that grant trusted user access to your AWS resources.

→ These temporary credentials are for short-term use and can be active for a few minutes to several hours.

→ once they expired they can no longer be used to access your AWS resources.

AWS A/C - I (Prod) AWS A/C - II (Dev)



S:1:- login 2 a/c in different brauser.

1 is Produc. 1 is Devp.

→ copy dev. a/c id, then in IAM create ~~Polity~~ Role in the ~~role~~ another aws a/c. Paste a/c id.

→ Click on Permissions, select S3-read only access.

→ Click on Tags then review.

→ Role name : S3 read ~~Another~~ for Dev.
Role disc : " "

→ Then create.

→ In a/c Create Bucket S3.

→ Create Bucket:

S:2:-

→ Go to dev. a/c create Polity, Go to IAM, select Policies.

→ Create Policy. and choose service, select 'STS'

→ under STS Expand write select Assume role.

→ Choose resources not all specific,

→ Add ARN.

→ copy produc a/c Role ARN and put here. (dev. a/c)

→ click on Review Policy, "S3-assume" role.
S3-assume-Production

→ Then create policy.

S3:-
→ Create IAM user and attach Policy 2 user
→ Dev. a/c. Add user : Raju.
→ Custom P.W: ——
→ click on Permission,
→ choose attach existing Policy directly.
→ S3 Assume Prod., click on Tag, create user (final step)
→ login IAM user.
→ In raju a/c Click 'switch role'
→ Copy raju Produc a/c. and Paste Dev. a/c. switch user.

Account settings :-
→ Under P.W Policy :-
A. P.W Policy is a set of rules that define. The type of P.W an IAM user can set.

→ click on set Password Policy, save changes.

25/11

AWS MONITORing, Audit & Performance

⇒ Simple Notification service (SNS) :-

→ It is a webservice that enable applications, end-users and devices to instantly send and recieve notification from the Cloud.

→ we able to use SNS to recieve notifications when events occur in our AWS environment.

→ SNS is integrated with many AWS services.

→ with Cloud watch and SNS, a fully environment monitering solution can be created that notify administrator of alerts, capacity issues, downtime, changes in environment and more.

SNS Components

TOPIC → The group of subscription that you send a message to.

Subscription → An end point that message is sent

Publisher → The entity that trigger the sending of message.

Endpoints → HTTP, HTTPS, EMAIL, SQS, LAMBDA, SMS, etc...

⇒ In Dashboard, Application service, Simple application service.

→ Go to topic, create topic.
Name: CPU Utilization.

Then create topic. (ARN → Amazon resource name).

→ 'Create subscription', Select 'Email'.
Endpoint name → gmail enter.
Then create subscription.

→ Go to gmail confirm subscription.

⇒ create another topic, name: Scalling.
Display name: scalling.
Then create topic.

→ Create subscription, select email,
enter mail Id.
Then create subscription.

⇒ create topic, topic name: Monthly Billing.
display name: Billing.
Then create topic → create subscription.

→ SMS, enter mobile number:

⇒ create topic., S3-Bucket notification
topic name: Bucket.
→ create subscription Email.

→ click on Publish message, Subject name: Text message
→ select 'String' type, [name]
→ Publish message.

<u>Text Messages</u>:-

Amazon SNS lets you send SMS text messages to any Phone number.

→ click on text messages, publish text message. write phone number, message body then Publish message.

Amazon Cloud TRAIL

→ Provide governance, compliance and audit for your AWS a/c.

→ Cloud trail is enable on your AWS a/c whenever you create it.

→ When activity occurs in your AWS a/c, that activity is recorded in a Cloud Trail event.

→ This event history simplifies security analysis, resource change tracking, and troubleshooting.

Trails :-

→ Deliver logs to an Amazon S3 bucket.

→ Cloud trail events can be processed by trail for free.

→ There is a charge for processing events

with additional trails.

Benfits :-

→ simplified complaience.

→ visibility into user and resource activity.

→ security analysis and troubleshooting.

Use Cases :-

→ Compliance aid.

→ security analysis.

→ operational issue and Trouble shooting.

Console.

⇒ Management & governance, → Cloud trail.

→ Go to S3 and Create Bucket : Aws-trail

Then Create Bucket.

→ Go to trails, Create Trails.

Trial name : MY_TRAIL.

All regions select "yes".

Read/write events "select all."

~~and~~ Data events record → S3 select.

Storage location select 'No', exist file.

Log file Prefix : Ctrail

SNS topic : S3 Bucket.

- Yes
- Yes
- NO

Then create trail.

26/11

Amazon Cloud watch

→ Amazon cloud watch is a monitoring service for AWS cloud services, resources and the applications you run on AWS.

→ It provides centralized logging and performance ~~meters~~ metrics for AWS resources.

→ Using cloud watch to collect and track metrics ~~for AWS resources~~, collect and monitoring log files, set alarams, and automatically react to change in your AWS resources.

→ Metric is a variable to monitor (CPU utilization N/w in.)

→ Cloud watch Alaram used "Triggers" in your AWS.

Monitoring Types

Basic :- Free to use.

→ Data is avilable automatically in 5-minutes Periods at no charge.

→ By default, your instance enabled with basic monitoring.

Detailed :- Pay as you go pricing

→ Data is avilable in 1-miny. period for an additional cost.

→ Aws free tier to allows us to have 10 detailed monitoring metrics.

→ Go to EC2, In the instance, launch instance.

→ select linux, click on config. inst. details.

In monitoring ☑ enable cloud watch detail monitoring

→ remain all steps common.

Cloud Watch DashBoards :-

→ Great way to setup dashboard for quick access to key metrics.

→ Dashboards are global, which includes graphs from differe. regions.

→ Go to cloud watch, Go to dashboards.

→ Create dashboard. name: "Billing team" create dashboard

→ Create dashboard name: "operation team", create dashboards.

Alarams :-

→ Alaram is used to triggers for notification to any metric.

→ most alarams states are: OK, INSUFFICIENT DATA, ALARAM.

→ select instance, choose monitoring,

→ create alaram.

☑ Send notific. to: [ ] topic

> 60 percent

→ select create alaram

N/w

1000 bytes , Select Create alaram.

→ create alaram for Billing.

→ under alaram select Billing.

→ create alaram, ~~Getting start~~

Billing alaram:

Exceed 10 USD

Then ~~Create~~ Treat miss good

Creat alaram.

Adding alarams to Dashboard :-

⇒ select alaram add to dash board, click add to dashboard.

Events :-

→ Schedule: Cron Jobs.

→ Event Pattern: Event rules to react to a service doing something (Ex: Code Pipeline State Changes)

→ Event created small JSON document to give inform. about the change.

*	*	*	*	*
[minutes]	[Hours]	[Days]	[months]	[weeks]
[0-59]	[0-23]	[1-31]	[1-12]	[0-6] (or) [1-7]

Commands to be executed.

0 & 7 → Sunday.

→ Ex:- 55 * * * * date

30 20 * * * ls

* * * * Cit Instance name.

→ cloud watch, Go to events, Rule.

→ In events get started, The fixed rate of field add target select Instance ID. Then Configur. details click.

→ Stop instance. Then create rule.

→ create another rule, schedule

→ select cron expression: 59 23 * * * *

→ click Target select Below., select ID:—

click Config. details.

27/11

CLOUD WATCH Logs :-

→ To collect logs from your EC2 instance and on-premists services into Cloud watch logs.

→ To send logs to Cloud watch, make sure IAM permissions are correct.

→ Logs can use filter ~~permissions~~ Expressions.

→ Dashboard, select Cloud watch.

→ Go to EC2, Launch one linux instance with Port 22, 80.

→ connect instance (go to Putty)

→ update server (# yum update -y)

→ install webPage (# install httpd -y)

cd /var/www/html.

cat > index.html

Log files intigration with cloud watch :-

#yum install awslogs -y.

vi /etc/awslogs/awslogs.conf

Go to last line (Ctrl+G), copy last lines and paste it.

→ Paste items change.

[var/log/httpd/access.log]

Apache ~~error~~ access log file. same as error log file.

→. # aws configure.

Access key Id

secret key Id

: US-east-1

cd. aws

ls

cat config

cat Credentials.

service awslogs restart.

chkconfig awslogs on

service awslogs status.

Status Checks :-

System status checks: (Things that outside of our control)

→ Loss of n/w connectivity.

→ Loss of system power.

→ S/w issue on physical host.

→ H/w " " "

* How to solv:- Generally stopping and restarting the instance will fix the issues.

→ This causing the instance to launch on a differe. physical H/w device.

Instance status checks: (sw issue that we do control)

→ Miss configu. nlwing (or) start-up configuration.

→ Exhausted memory.

→ Corrupted system.

→ Incompatible kernel.

~~Y~~ How to solv :- Generally, a reboot (or) Resolving file system configu. issue.

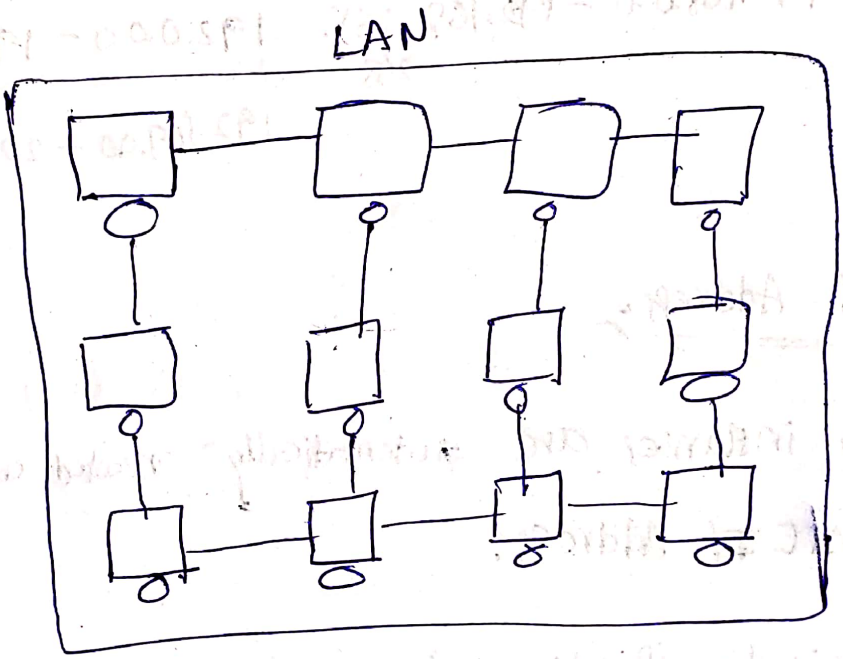
28/11

AWS Networking

→ connection b/w two (or) more device is called as Nw.

→ Sharing possible b/w interconnection of devices is called as Nw ing.

eg:- LAN, WAN (wire Area Nw).



IP-Address :-

→ An Internet Protocol (IP) address is a numerical label assigned to each device connected to a computer nw that use the internet protocol for communication.

→ The Internet Assigned number Authority (IANA) reserves the following IP address blocks.

Class	Private IP Address Range	Public IP Address Range
A	10.0.0.0 – 10.255.255.255	1.0.0.0 – 9.255.255.255 11.0.0.0 – 126.255.255.255
B	172.16.0.0 – 172.31.255.255	128.0.0.0 – 172.15.255.255 172.32.0.0 – 191.255.255.255
C	192.168.0.0 – 192.168.255.255	192.0.0.0 – 192.167.255.255 192.169.00 – 223.255.255.255

Private IP- Address :-

→ All EC2 instances are automatically created with a PRIVATE IP Address.

→ The private IP address is used for internal (inside VPC) communication instances.

Public IP- address :-

→ When creating an EC2 instance, you have the option to enable (or autosign) a public address.

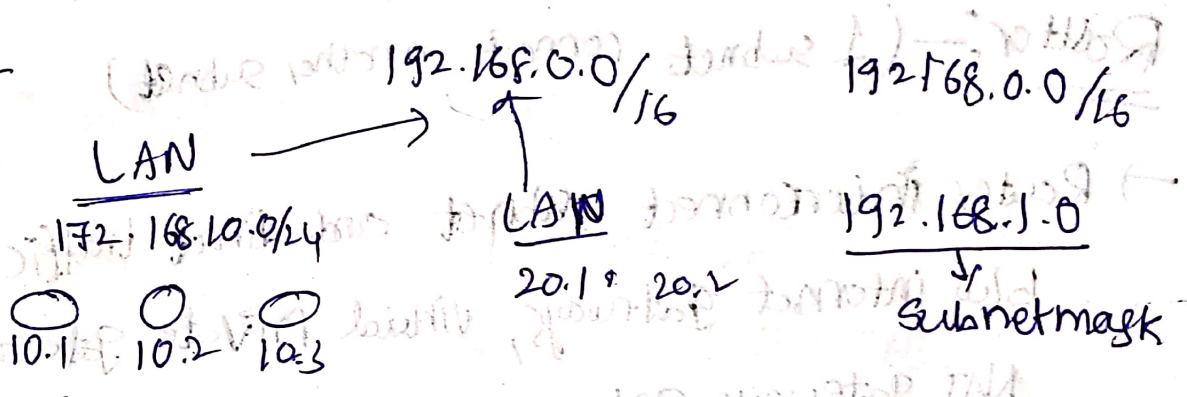
→ A public IP address is required if u want to the EC2 instance to have direct communication with resource across the open internet.

CIDR:- (collection of IP address)

→ Classless interdomain routing.

→ It is a set of internet protocol (IP) standards that is used to create unique identifier for nlw and individual devices.

eg:-



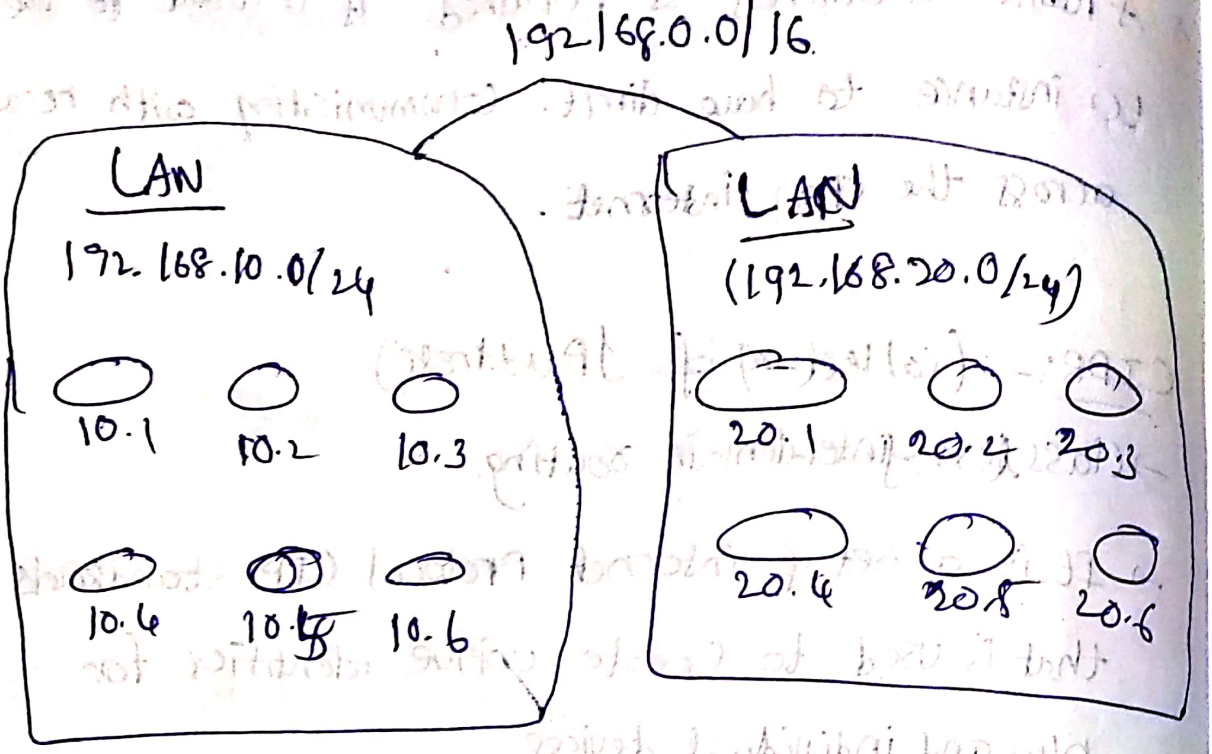
Subnet:- (Portion of the nlw)

→ It is seperate and identifiable portion an organization nlw, typically a local area nlw.

→ A subnet is range of IP address in your VPC.

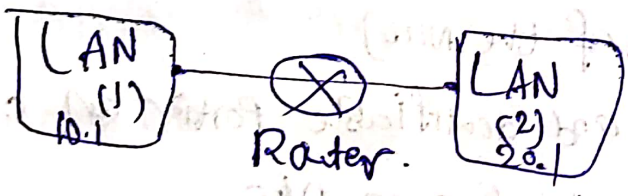
eg:- 192.168.10.0.

→ A subnet lives within a VPC and with a single Availability zone.



Router:- (1 subnet connect another subnet)

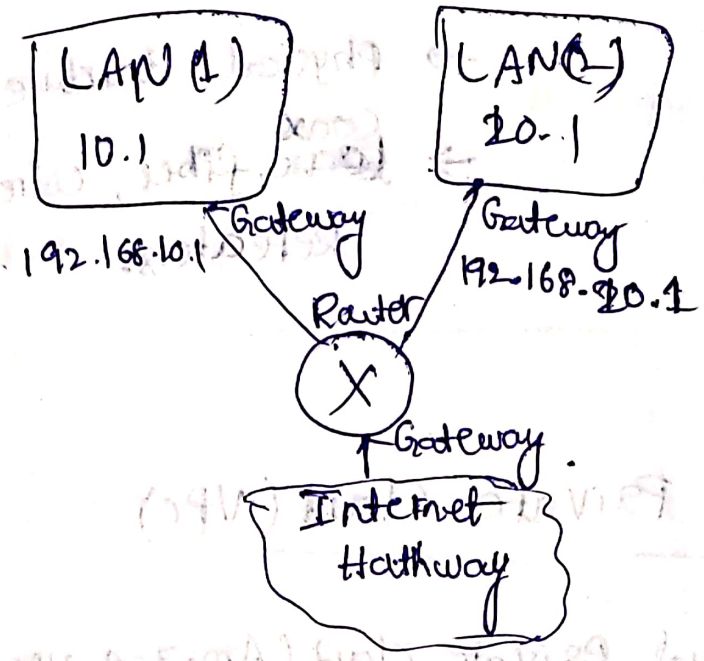
→ Router interconnect subnet and direct traffic b/w internet gateways, virtual private gateways, NAT gateways, and subnet.



Internet Gateway:-

→ An internet gateway is a n/w "node" that connect two differe. n/ws that use differ. protocols for communicating.

→ It allows Communic. b/w instances in your VPC and the Internet.



OSI MODEL (7 Layers)

Application	• End User layer • HTTP, FTP, IRC, SSH, DNS.
Presentation.	→ Syntax layer → SSL, SSH, IMAP, FTP, MPEG, JPEG.
Session	→ Synch & send to port. → API's, sockets, WinSock.
Transport	→ End to end connection → TCP, UDP.
Network	→ Packets → IP, ICMP, IPsec, IGMP

2. Data Link → Frames
→ Ethernet, PPP, switch, Bridge.

1 Physical → Physical structure.
→ Coax, fiber, wireless, Hubs, Repeaters.

9/12

Imp Virtual Private Cloud (VPC)

* Amazon virtual Private Cloud (Amazon VPC) enables to you launch AWS resources into a virtual N/w that you've defined.

* It is a virtual n/w dedicated to your AWS a/c. It is logically isolated from other virtual n/w in AWS cloud.

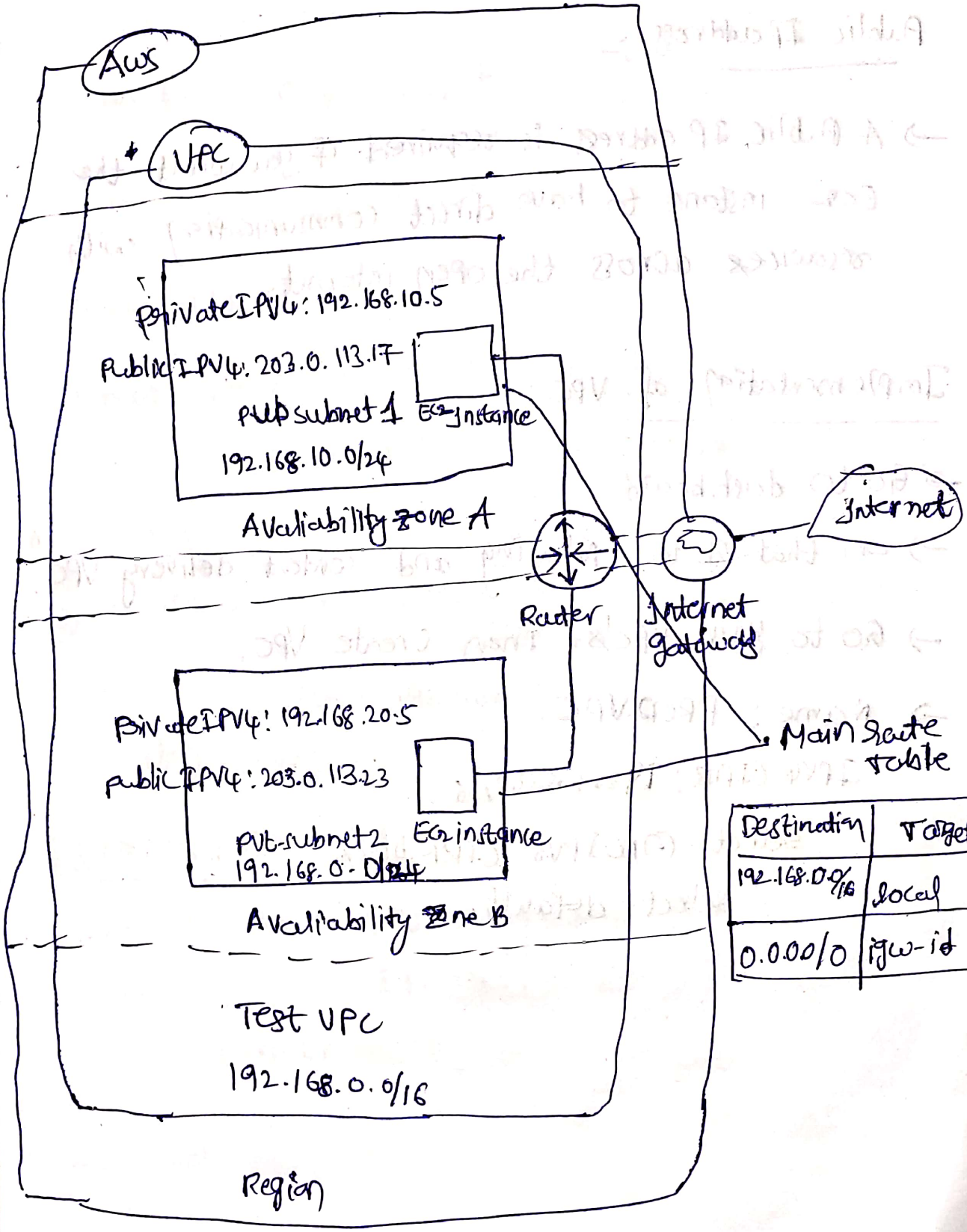
The Default VPC:-

→ The default VPC comes with Preconfigured setup in your AWS a/c.

→ It is differe. setup than a non-default VPC.

→ In the default VPC, All Subnet have a Route to the internet via route table and an attached to IGW.

→ Each instance in the default VPC has a Private and Public IP address.



Destination	Target
192.168.0.0/16	local
0.0.0.0/0	igw-id

IP- Address:-

Private IP-address :-

→ The private IP address is used for internal communic. b/w instances within the VPC.

Public IP address :-

→ A Public IP address is required if you want the EC2 instance to have direct communication with resources across the open internet.

Implementation of VPC :-

→ Go to dashboard.

→ on that go to "Nlwing and content delivery VPC".

→ Go to your VPC's. Then Create VPC.

→ Name : PROD.VPC.

IPV4 CIDR : 192.168.0.0/16

select ⊙NO IPV6 CIDR block.

select: default.

Creating Subnet:-

192.168.0.0/16 (CIDR)

192.168.10.0/24 → Pub1

192.168.15.0/24 → Pub2

192.168.20.0/24 → (Pvt1)

192.168.25.0/24 → (Pvt2)

Create public subnets:-

→ Creat subnet : name: Prod-Pub1
US east-1a

IPv4 CIDR Block: 192.168.10.0/24

Then create VPC.

→ create another: Prod-Pub2
US-east 1b

IPadd : 192.168.15.0/24.

Then create it.

Creating Private Subnets:-

→ create subnet name tag: Prod-Pvt 1
east-1c

IP-add: 192.168.20.0/24.

Then create it.

→ ~~tag~~ Prod-Pvt 2
east-1d

IP : 192.168.25.0/24.

Create a Route Tables :-

→ Create public Route Table

Name : Prod - Pub - RT

→ Public subnets attached to Public Route Table.

→ Select Public route Table, Subnet associations

Then choose "Edit Subnet Associations".

Select Prod - Pub 2

Prod - Pub 1 .

→ Create Private Route Table.

Create Route Table

Name-tag : Prod - PVt - RT .

→ PVt subnet associate Private subnet associations.

Edit subnet associations.

→ select Private subnet - 1

Subnet - 2 then Create/Select.

→ Select public route table, edit routes

Now ~~add~~ edit route → target IGW → Save IGW.

0.0.0.0/0

Creating Internet Gateways :- (IGW)

→ create internet Gateway
Name tag : Prod-IGW.

→ select Prod-IGW → Go to actions 'Attach to VPC'

→ on that select ~~Prod~~ VPC : Produc.
then attach it.

Enableing DNS and Hostnames :-

→ your VPC, select Produc. VPC → Go to Actions
Edit hostname : select enable ☑. then save.

3/12 VPC flow Logs :-

→ It capture information about the IP traffic going to and from n/w interface in your VPC.

→ Flow log data can be Published to Amazon cloud watch logs and Amazon S3.

→ Go to Cloud watch, under logs
'Actions'.
→ Create "log group". Lg name : Produc Flow-logs.
then create log role.

→ Go to VPC., select Prod-VPC, and choose flow logs, Then Create flow log.

Filter Type: Accept.

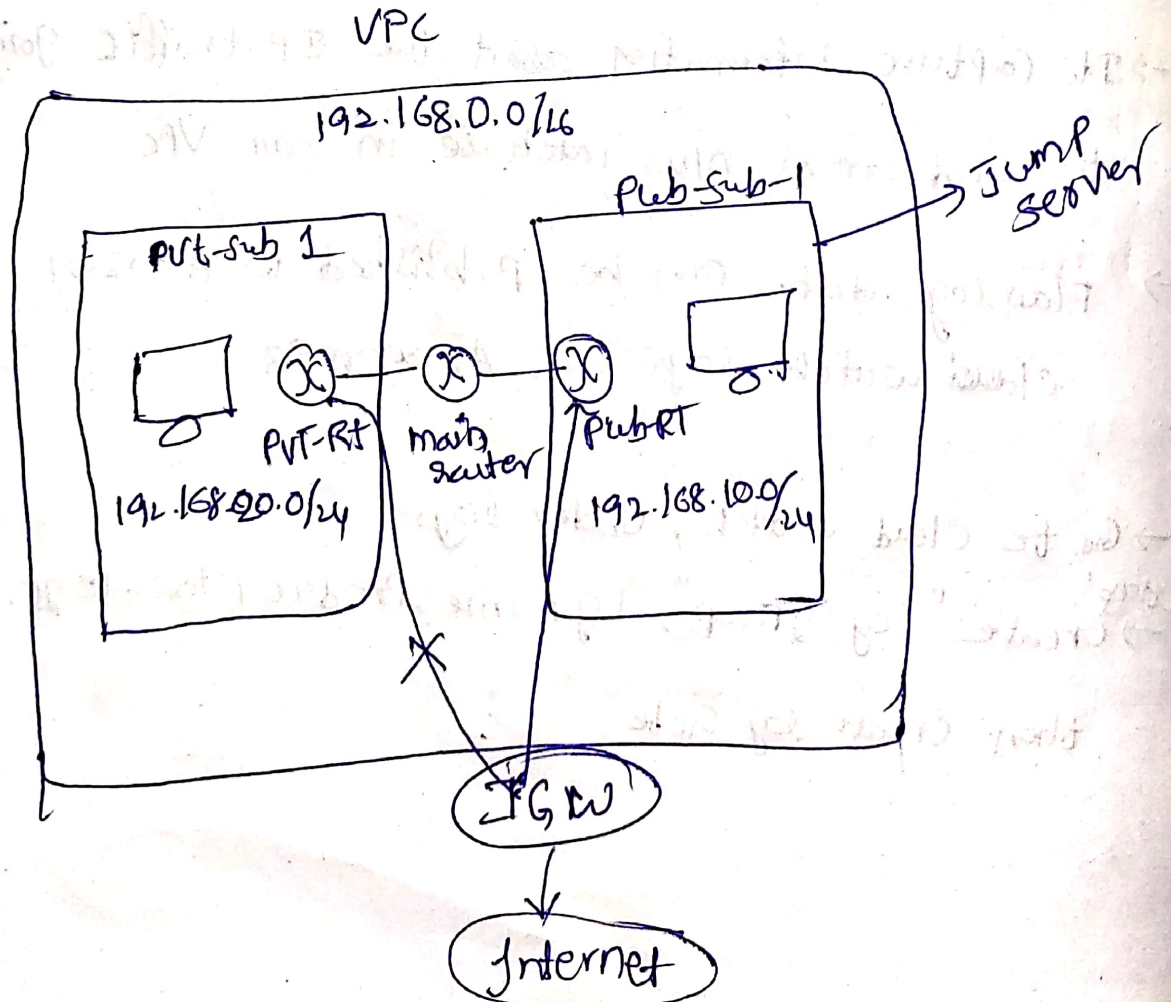
setup permissions.

Role: 'Flow log Roles.', Then allow.

Then create it.

Auto assign Public IP Address:-

→ Select Pub-subnet-1, Action, 'modify IP settings'. then enable it, after save it.



→ IGW not connected Pvt-Rtr

Launching instance on public subnet1:

→ Go to EC2, under instance, Launch instance

→ select any linux,

3) config instance.

N/W: prod-VPC

subnet: Pub-sub1.

→ 6): Prod-SG
Prod-SG

SSH	22
HTTP	80
HTTPS	443
RDP	3389

→ click on Review and launch.

→ connect instance, use Putty.

Launching instance on Private subnet1:-

→ Launch instance, select linux.

→ subnet: subnet Prod-Pvt1.

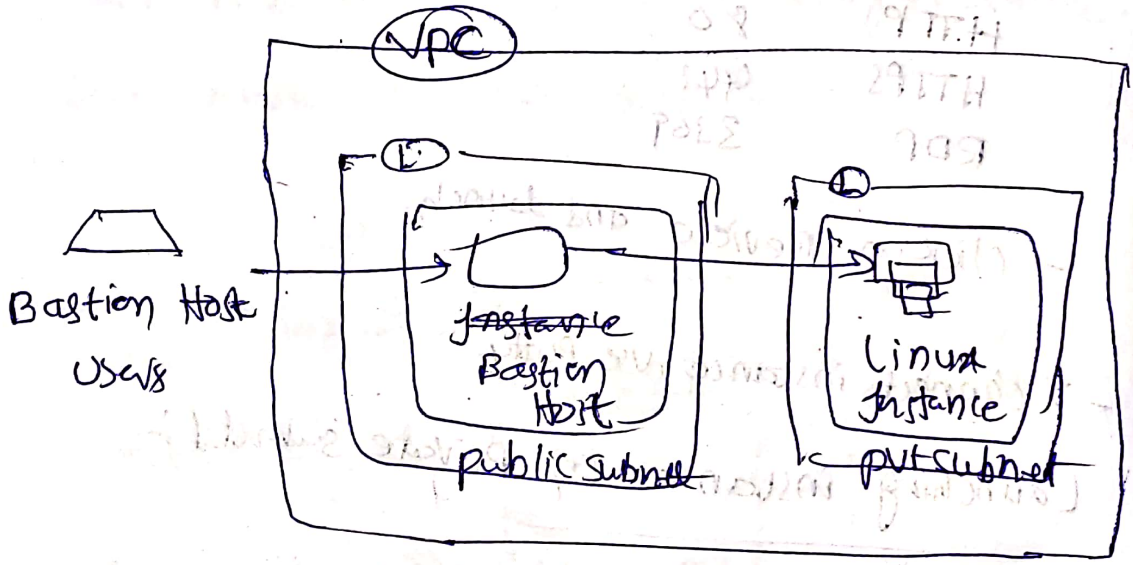
Auto assign: Disable

BASTION HOST

→ Bastion Host are instances that sit within your subnet and are typically accessed using SSH (or) RDP.

→ once remote connectivity has been established with the bastion host, it then act as a 'Jump' server within your VPC.

→ It essentially act as a bridge to your private instance via the internet.



Create a key Pair:-

→ Create key Pair Under Public instance.

```
$ cat > Sample.pem
$ ls -l sample.pem
$ chmod 400 Sample.pem
$ ssh -i sample.pem ec2-user@ pvt IP address.
```

NAT Gateway :- (Public Subnet)

→ A Highly available, managed N/W Address Translation (NAT) service for your resources in a private subnet to access the internet.

→ Go to VPC, NAT Gateways.

→ create NAT gateway.

→ select public-subnet1,

→ then create New EIP., Then create NAT gateway.

→ NAT gateway attached to private subnet.

→ Go to Route table, select Prodve. Private RT.

→ Routes choose, Now edit routes.

→ Add route 0.0.0.0/0, NAT gateway

Imp

→ NAT gateway enable then black hole created.

NAT instance :-

→ AMI

'~~sea~~search: nat

community AMI: ami select.

→ Prod-VPC

Prod-Pub1

Autoassign: enable.

→ Go to route table, select Prod-Pvt-RT, Routes edit route.

→ 0.0.0.0/0 , Instance

Desti. Target.

4/12

Elastic IP address

Imp

→ Static IP-address for Dynamic Cloud Computing.

→ An EIP is Static IPv4 address designed for dynamic cloud computing.

→ Attach an EIP to Instance will Place its default IP address for as long as it is attached.

→ Create an EC2 instance on Public subnet.

→ Creating elastic IP address :-

→ Under VPC, elastic IP's.

→ Allocate new address

grap IP addry ⊙ amazon Pool.

Then close it.

EIP attach to instance:

→ select EIP address, Acting 'associate address'.

→ Resource type : ⊙ Instance

Then associate.

→ To deassociate EIP, Action deassociate address.

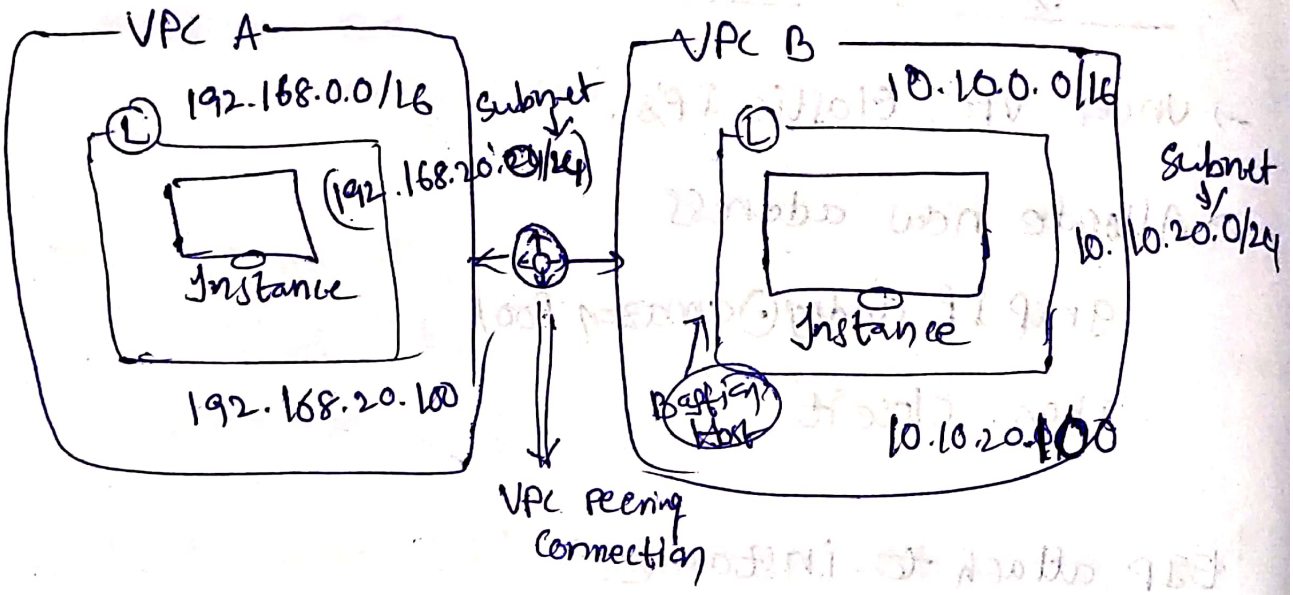
Then Release address.

Imp

Peering Connection

→ A VPC Peering connection is a nlwing connection b/w 2 VPCs that enables u to route traffic b/w them using Private IPV4 address (or) IPV6 address.

→ A VPC Peer connecting facilitate to transfer data.



→ NOTE:-

Create another VPC.

→ Go to Peering Connection Create Peering Connection

: Peering connect tag: Prod-Dev-Peering

→ Select another Peering

: Dev

Create Peering Connection.

→ Select Peering Connect, Action select Accept.

In the Route Tables select Prod Private Route table. Under Route section edit Routes.

10.0.0.0/16 Peering Connection.

→ Select DEV- Prod - RT Address

192.168.0.0/16 Produ.dev Peering

Then save Peering.

→ Launch another instance on Production Private

~~Subnet~~ Instance.

Produ- PVT

Disable

→ Launch another instance on Develop Private

~~Subnet~~ instance

DEV- VPC

PVt-Sub VPC

Disable.

→ Connect Public instance (Bastion Host).

Peering go to,

After Connecting DNS Resolved, Action & select Peering Connection "edit DNS settings"

→ Accept DNS resolution ☑, Then save it.

~~→ Rebooter~~ # cat > sample.pem

ls -l

chmod 400 sample.pem

ssh -i sample.pem ec2user@IP

Clean up Peering Connection

→ select peering connection 'Actions', Delete VPC Peering connection.

ICMP enable :- Ping communication.

→ Go to security group select Produ. SG

Choose Inbound rules edit rules then add rule

select : All ICMP-IPv4 Save.

→ Then go to Instance & Ping 192.168.0.0.212 :

5/12

End Points

→ A VPC endpoints enables u to privately connect your VPC through supported aws services and VPC endpoint services. Powered by Private link without requiring an internet gateway, NAT device, VPN connection (or) Aws direct connect connections.

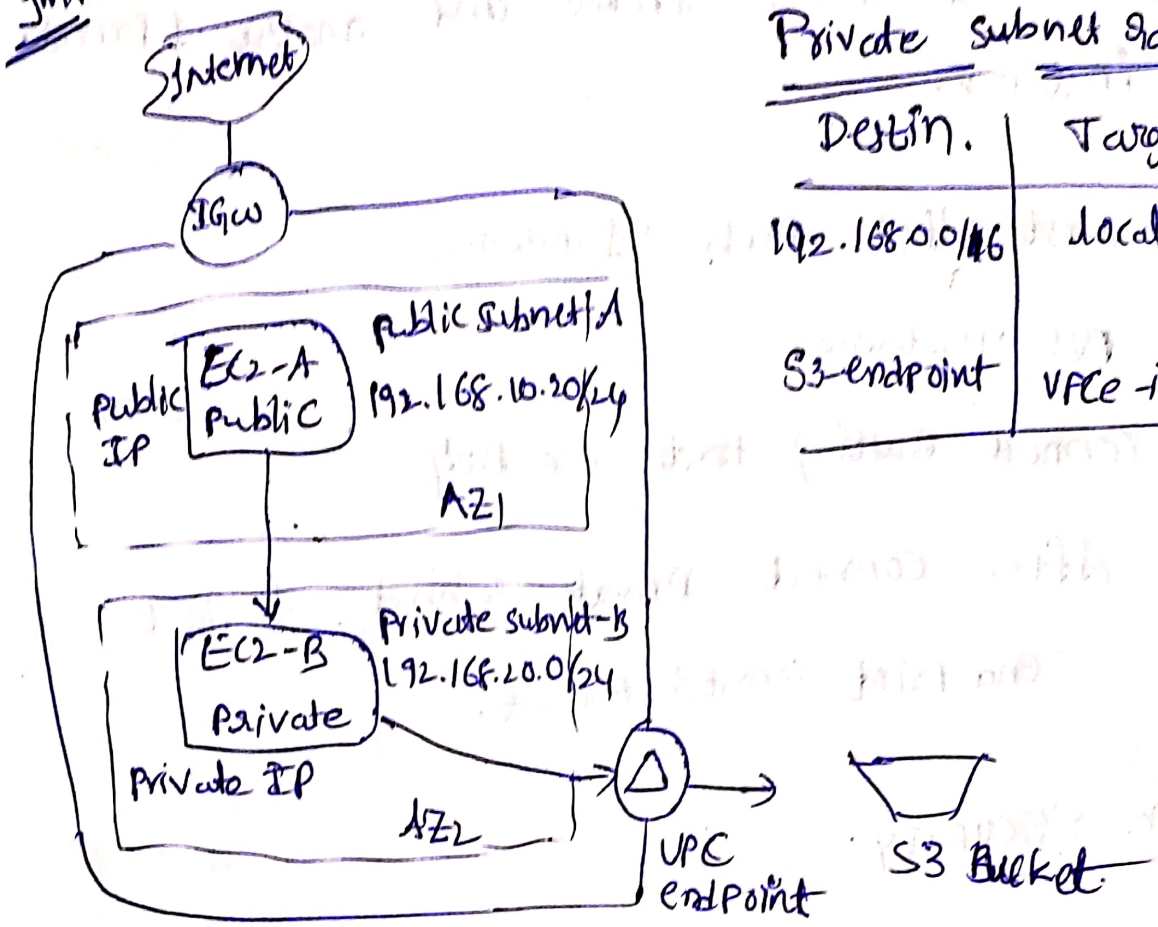
→ End Points are virtual devices.

→ Allow private access to S3 and DynamoDB.

→ Gateway type end points.

→ Interface type endpoints.

Imp



Private subnet Route Table

Destin.	Target
192.168.0.0/16	local
S3-endpoint	VPCe-id

→ In VPC, go to End Point, select end Point

Create end Point.

→ select amazon S3

→ VPC: PROD VPC.

→ Select 2 subnets PVt1, PVt2

→ Policy type full access

→ After create endpoint

→ Go to Route Tables select Produc-Private Rout table
edit Route Table.

→ Go to ~~S3~~ S3, Create Bucket

→ Launch 2 instances 1 Public and another 1 Private instance.

→ Bastion Host Pub. → 1 instance.

→ Pvt → instances

→ Connect Bastion Host use Putty.

→ After connect Private subnet and go to end point connect S3 Bucket.

VPC SECURITY:

Security Groups:- (SG)

→ SG are acts as firewally controlling traffic at instance level.

→ SG ~~allow~~ support only allow rules.

→ SG are "Stateful".

→ Go to security groups create security group.

SG Name : MY-SG

VPC : Prod-VPC then create it.

→ select MY-SG, Inbound rules, SSH. custom 192.168.10.0/24

Network Access control list (NACL's) :-

→ It is an optional layer of security for your VPC that act as firewall for controlling traffic in and out bound one (or) more subnets.

→ ACL operate on N/w/ subnet level.

→ By default, all IN and OUT traffic deny until you add rules.

→ ACL's are stateless.

⇒ Create New ACL :-

→ In the In Bound rule edit inbound rule

10 SSH(22) DENY.

→ out Bound rule edit

10 SSH(22) DENY.

Then save it.

→ In the Inbound rule, edit Inbound rule

100 SSH(22) ALLOW.

→ outbound rule edit it.

100 SSH(22) ALLOW.

eg:-

→ In outbound rule custom TCP rule Port range 49152-65535

#netstat -Pantl. (Find port number) ALLOW. ✓

Create a new ACL :-

: Prod-NACL

: Prod VPC. create it.

6/12

VPN CONNECTION

Virtual Private Network:

→ By default, instance that you launch into an Amazon VPC can't communicate with your own (remote) N/W.

→ You can connect your VPC to remote N/W's by using an VPN connection.

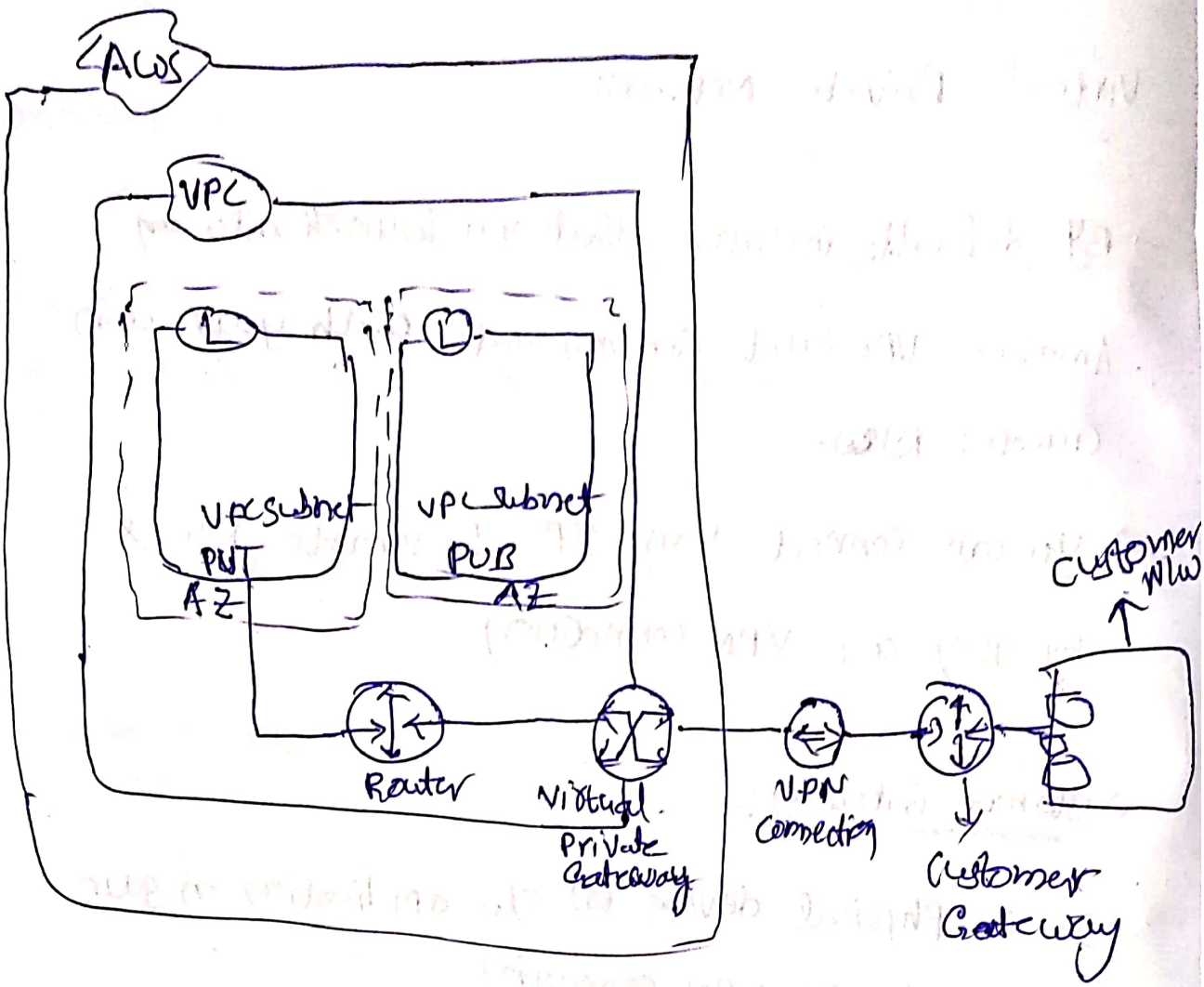
Customer Gateway:-

→ It is physical device or s/w application on your side of the VPN connection.

Virtual Private Gateway:

→ It is the VPN Concentrator on the ~~amazon~~ amazon side of the site-to-site VPN Connection.

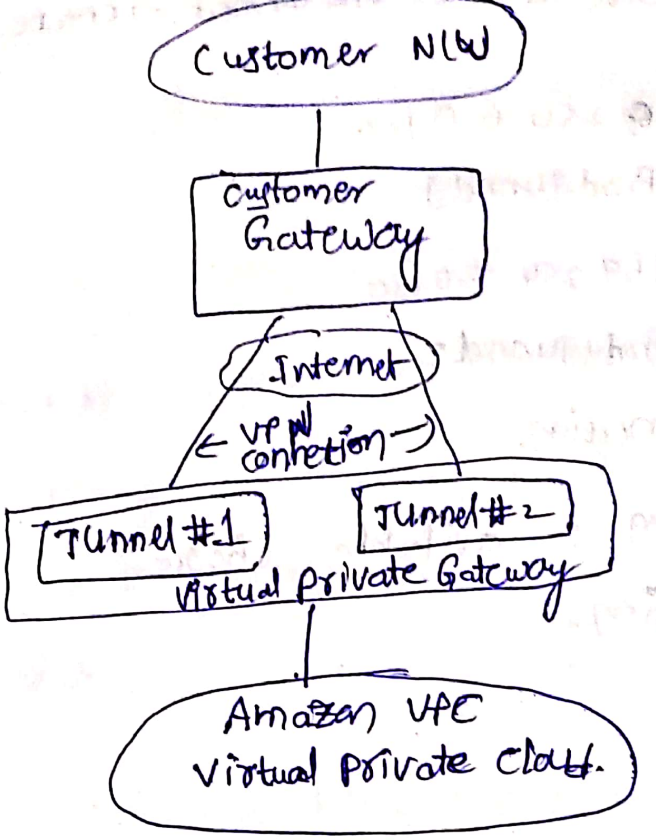
Single VPN connection :-



VPN Tunnel

→ VPN connection consist of two tunnels to provide increase availability for the VPC service.

→ If there's a device failure within AWS, your VPN connection automatically fails over the second tunnel so that your access isn't interrupted.



→ Go to Customer Gateway, Name: Prod-CGW.

→ Create Virtual Private Gateway.

Name: Prod-VPG

⊙ Amazon ASN. Create gateway.

→ Action Attach to VPC, Then Produc. VPC.

→ Creating VPN Connection.

tag: Prod-VPN

VPGateway: Select.

Routing Option: ⊙ Static

54.125.5.20

~~192.168.0.0/24~~

Go to google search IP address

~~SA~~ site-to-site IP address.

→ VPGateway: Nametage: Prod VPG → ASN Amazon

Action → Attach VPC → Produc VPC.

Site to site VPN connect → create →

Tunnel options:

inside range : 169.254.6.0/30

: ProdTunnel1

range : 169.254.7.0/30

: ProdTunnel2.

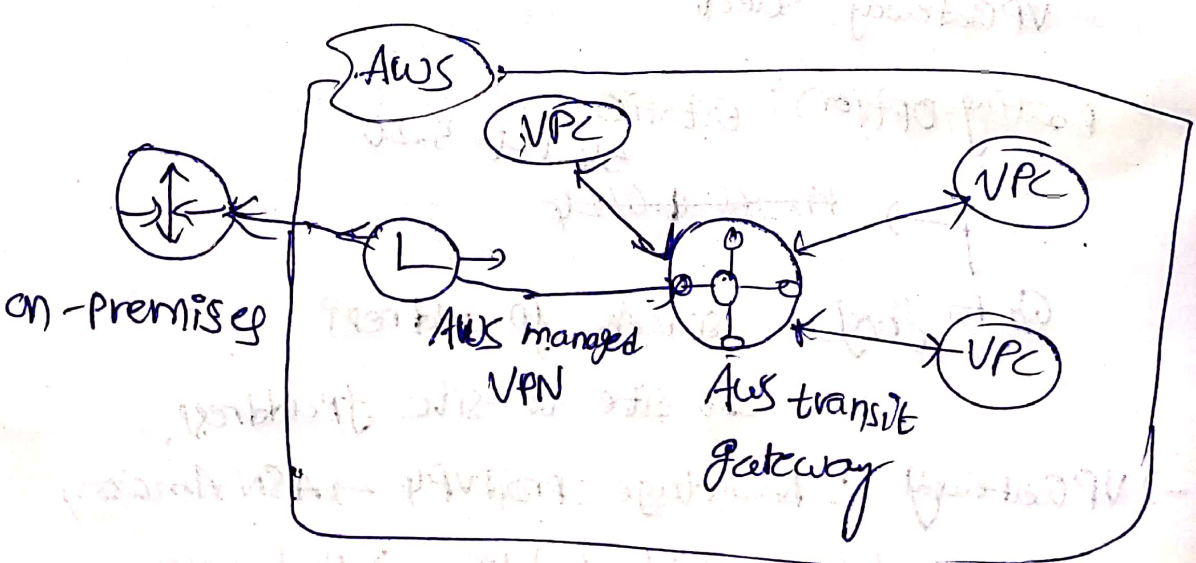
Create VPN connection.

→ once VPN connection is available choose download configuration.

Transit Gateway:

→ A transit gateway is a transit hub that you can use to interconnect your virtual Private Clouds (VPC) and on-premises N/w to single gateway.

→ Transit gateway act as a hub that controls how traffic is routed among all the connected n/w which act like spokes.



10/12

AWS CLOUD STORAGE

Block Storage :-

→ A block device is a storage device that data moves in sequence of bytes (or) bits which are called blocks. These device support random access and generally use buffered I/O.

→ Example block devices are Harddisks, CD-ROM drives and flash drives. A block device is physically attached a computer (or) access randomly.

→ Raw storage volume are created in block-level storage device and then the server-based operating system connect to the volumes and use them as individual hard drives.

→ Best option to use for database storage, virtual machine filesystem (VMFS) volumes and more.

EC2 Instance Store:-

→ Instance stores volumes are virtual devices whose underlying h/w is physically attached to the host computer that is running the instance.

→ Instance store volumes are considered Ephemeral data, meaning the data on the volumes only exist for the duration of the life of instance.

→ Once instance is "stopped" or "shutdown" the data is erased.

(Instance RAM EBS ROM)

→ Go to EC2, Launch instance.

→ select community AMI's. , select instance store.

Elastic Block Store (EBS)

→ EBS Provide Persistent block storage volumes for use with EC2 instance in the Aws Cloud.

→ Each ~~EC2~~ Amazon EBS Volume is automatically replicated with in AZ.

→ EBS volumes are highly avilable and reliable Storage volumes that can be attached to any running (or) Stopped instance that is in the same AZ.

→ An EBS Volume can be attached to only one instance at a time, but multiple Volumes can be attached to a single instance.

→ EBS volumes have benfit of being ~~backed~~ backed up into a snapshot - which can later be restored into a new EBS volume.

EBS Performance :-

→ EBS volume measure input/output operation in IOPS.

→ Aws measures IOPS in 256 KB chuckss.

→ For ex: 512 kbps operation would count as 2 IOPS.

EBS Volume Types

Solid State Drive (SSD):-

SSD is the best operation of IOPS-intensive Use Cases such as transtional workloads, database & boot volumes.

General Purpose SSD (gp2):

→ Use for Dev/Test environment and smaller DB instances.

→ Volume size of 1Gib to 16Tib.

Provisioned IOPS SSD (io1):

→ Used for mission critical applications, large database workloads.

→ volume size of 4Gib to 16Tib.

Hard Disk Drive (HDD):-

→ HDD is the best operation for throughput-intensive use cases like storage, MapReduce and log processing.

Throughput Optimized HDD (st1):

→ Low Cost HDD Volumes designed for frequently accessed, through-intensive workloads. For eg: Big data, warehouses, Log Processing.

→ The Volume size of 500Gib to 16Tib.

Cold HDD (sc1):-

→ lowest cost HDD Volumes designed for less frequently accessed workloads.

→ The volume size of 500Gib to 16Tib.

Magnetic (Standard):-

→ It is used for workloads where performance is not an important (or) data is an infrequently accessed.

→ Low storage cost.

→ Volume size of Min 1Gib max 1Tib.

11/12

→ Go to EC2, Launch instance, select any linux

→ Step 4: Add storage.

EBC 3 Magnetic standard.

→ Connect instance, Go to putty.

lsblk → To check volumes, fdisk -l.

mkfs.ext4 /dev/xvdb → ~~Partition~~ format device.

create a direct mount

~~mount~~ mkdir /aws-data

mount /dev/xvdb /aws-data/

df -h

cd /aws-data/

touch abc xyz

ls.

cd.

Modify Volume (Resize Volume) :-

select volume, 'Actions', modify volume.

→ Volume Type: General Purpose.

size : 4gb

Then modify volume.

lsblk. (Change volume size)

Creating new volume :-

under Volumes, Create Volume.

→ Volume type : Magnetic standard

Size : 1Gb

AZ : US-east-1C.

Then create volume.

Volume attach to instance.

select volume, Attach Volume.

select instance

device : /xdv /sdc.

lsblk.

fstab → file Persistent

Detaching Volume:

Before detaching 1st unmount file system.

df -h

umount -f /aws-data/.

Go to aws dashboard., Volume, Action & 'detach' Volume.

EBS snapshot :-

→ A EBS snapshot is a point-in-time backup of your EBS volumes that are stored in S3.

→ Snapshot occur asynchronously which mean that snapshot process can work while your EC2 instance is running. An in-progress snapshot is ~~required~~ not affected by ongoing read and write to the volume.

→ Snapshot is a good ~~recovery~~ solution for a disaster-recovery of your EBS volume.

→ snapshot are incremental in nature. That mean it store the changes since the most recent snapshot, thus reducing costs.

Create snapshot :-

Select Volume, Actions, Create snapshot.
name: my snap then create snapshot.

→ we can't read directly snapshot data.

→ Snapshot converted to volume,

volume attached to instance and mount it.

→ Go to Actions snapshot, create volume.

→ select volume, Action, Attach volume.

select instance :

: lsblk

Fast snapshot recover :-

snapshot select Actions manage fast snapshot

Usecase, save.

image → pre configu image

→ If snapshot contain any root volume backup

better to make image.

Automatic snapshot :- (Produ. envir.)

→ Go to Cloudwatch, Cloud watch events,

S:1 Create rule

schedule

Fixed rate.

Target.

Volume ID :

S2:- Config detail.

name: autosnap volume.

Data Life Cycle Manager (DLM)

→ DLM for EBS snapshots provide a simple, automated way to backup data stored on Amazon EBS volumes.

→ you can define backup and retention schedules for EBS snapshot by creating life cycle policies based on tags.

→ Life cycle manager, create Life cycle manager

Desori : 1 week

Retain : 7 (or) 1 month.

12/12

EFS (Elastic File system)

→ An EFS provides scalable file storage for EC2 instance and on-Premises resources.

→ EFS is a regional service where the data is replicated across multiple AZ's in your AWS region.

→ You can attach multiple EC2 instance to EFS as a centralized file storage at a time.

→ EFS is growing and shrinking automatically as u add and remove files.

→ You pay only for the storage used your file system.

→ EFS supports the Nlw file system (NFSV4) protocol.

NOTE:- Microsoft windows based EC2 instance are not supported.

EFS Benfits:-

→ Dynamic elasticity.
→ Shared file storage.
→ Seamless integration.
→ Automatically scales.
→ Scalable performance.
→ Low Cost.
→ Secure.
→ Data transfer & backup.

Prerequisites :-

→ create a SG:

ssh -22, nfs -2049.

→ EC2 instance.

→ nfs - utilities Package.

[6KB] → Initially

~~EFS~~

Go to AWS → Go to EFS.

→ Create SG ~~EFS~~

SSH 22

NFS 2049 Then create.

→ Go to EFS, create file system.

VPC : Prod-VPC

→ Enable mount target.

Useast-1a EFS SG

Useast-1e.. EFS SG.

→ Add tags

My-IMP data

→ select days

→ choose throughput.

→ Choose performance. General Purpose.

→ Next step Click.

Launch instance:-

→ select : PROD-VPC

Subnet : Useast-Pub-sub-1.

→ select EFS-SG, Launch EC2.

→ connect instance (open Putty)

yum update.

yum install nfs-utils -y → install package.

rpcinfo → check Package.

Create a directory for mount EFS file system.

mkdir /efs-data

mount DNS name paste /efs-data.

df -h.

cd /efs-data/

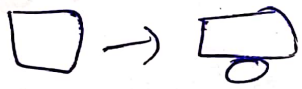
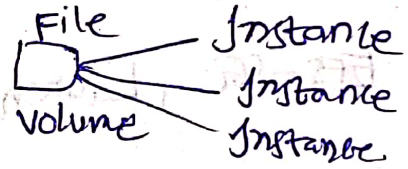
touch abc

ls

Manage file system access.

Select file system, Action/ manage file system access.

Imp → Difference b/w EBS and EFS

EBS	EFS
1. Block storage	File storage.
2. For EC2 instance	For EC2 (Linux)
3.  one volume → one instance	
4. Size not automatic	Size automatic ~~Expand~~ & EZ (unlimited)
5. Not Central Storage	Centralized storage.
6. No Connect anyport (SSH)	NFS - mandatory (2049)
7. For instance APP, OS	Doesn't support instance, formating, Deleting.

13/12

Amazon S3 (simple storage service)

Simple storage service (S3):-

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

→ whenever you see word durable, always think about S3

→ objects are synced across all AZ's for extremly high avilable and durability (99.999%)

→ objects are stored in buckets.

S3 bucket:-

→ Buckets are main storage container for objects.

→ Each bucket must have unique name across ALL of AWS.

→ There is unlimited storage.

→ 100 bucket per a/c.

→ S3 bucket in region that makes sense to its purpose.

* serving content to customer.
* Sharing Data with EC2

S3 Objects :-

→ objects are static files contain meta-data information

→ Each object must be assigned a storage type, which determine object availability, durability and cost.

→ all object private by default.

→ objects can:

* Be small as 0bytes and as large as 5Tb.
* Be made publicly avilable via a URL.
* Have multiple versions.
* Automatically switch to different storage class or deleted.

S3 Security:-

→ All Buckets created in PRIVATE mode.

→ Bucket access control by.

* Bucket Policies
* Access control Lists.

→ object access controlled by ACL.

→ Bucket Activity Tracking:

* Logs (Placed in same (or) differe. blocks)

* Events (send notification to person (or) other AWS service)

Objects Encryption:-

Server-side Encryption:

→ SSE-S3: S3 Managed keys - check box implemented, object level keys, master key rotation.

→ SSE-KMS: use of envelope to retrieve private key, Key under customer control.

Client-side encryption:-

→ SSE-C: Customer-managed keys given to S3 to encrypt objects.

→ Encryption/decryption handle at client end, S3 see only encrypted objects.

→ In Aws A/c, Go to S3.

→ Create Bucket : satya-aws

Click on
→ Bucket name, permissions → Block permission option edit.

① Uncheck: ☐ Block all public access.

→ Create Bucket : satya-devops.

Region : mumbai. click on next.

→ default options.

→ uncheck options ①. click next.

Create Bucket.

Uploading objects :-

→ ~~click~~ open bucket, select 'upload' Add files

→ then upload.

To make object public :-

click on object, make public

then access object url.

(or)

→ select object permissions under public access select everyone. Check & uncheck

Then save it.

Storage Classes :-

→ Standard ⇒ Frequent access data.

→ Intelligent Tiering ⇒ Long lived data with changing (or) unknown access patterns.

→ Standard-IA ⇒ Long lived, infrequently access data.

→ onezone-IA ⇒ " " , non critical data

→ Glacier ⇒ Archieve data with retrieval times ranging from minutes to hours.

→ Glacier Deep Archieve ⇒ Archieve data that rarely, if u need to access with retrival times in hours.

→ Reduce Redundency ⇒ Frequently not accessed.

⇒ Go to Bucket, Upload, add files.

⇒

14/12

Bucket Properties

S3 versioning :-

-) To keep multiple versions of an object in one bucket.

-) once enabled, can only be suspended, not disabled.

-) Integrate with Lifecycle management.

-) Required for Cross Region Replication.

⇒ Click on bucket, properties click versioning.

⊙ Enable versioning then save it.

S3 server access logging :-

-) server access logging provide details record for the request that made to a bucket.

-) server access logs are useful for many applications

-) click on server access logging.

⊙ Enable logging.

Target bucket.

Target prefix.

Then save it.

Imp

S3 Static website Hosting :-

→ S3 can be host entire static website.

→ requiring no virtual machine/instance.

→ Amazon Route 53 can also map human readable domain names to static web hosting buckets, which are ideal for DNS failover solution.

→ Click on static website Hosting, ~~use~~ select, use these bucket host a website.

index document
" . html
error document
" . html

→ Create a bucket, Bucket should domain.
~~kimano~~.in. Create bucket.
Cloudhelps

→ To make a bucket Public.
Go to bucket permission, Click edit option.
Uncheck Public access than save it.

→ select bucket policy.
google: S3 bucket policy.
Copy content
Paste it
Change last line "Cloudhelps.in"

S3 object level logging:-

→ Cloud Trail is the AWS API auditing service, object-level logging allows you to incorporate S3 object access to your central auiding and logging in Cloudtrial.

→ click on object level logging.

Select Trail option.

Read write, then create.

Default encryption :- (server side encryption)

→ objects are automatically encrypt when stored a Amazon S3.

S3 Transfer Acceleration :-

→ Amazon S3 transfer Acceleration enables fast, and secure transfer of files over long distance b/w your client and S3 bucket.

→ Offload object transfer (in/out) from bucket region location to AWS edge location.

→ Enable it.

S3 Event :-

→ Receive notifications when specific events occur in your bucket.

→ click on events, Add notification.

→ select event send to topic, and select bucket name.

Object properties :-

Storage class :-

→ TO change existing store of object.

→ click on storage class, select differe. storage class then save it.

Encryption :-

→ use encryption option to protect your data while intransit. And at ~~reset~~ rest.

→ click encryption, AES-256 then save it.

Meta data :-

→ Assign optional meta data to the object as a name value pair.

Bucket Management :-

Imp

S3 Life cycle Policies :-

→ An object life cycle Policy is a set of rules that automate the migration of an object storage class to a different storage class, based on specific time intervals.

→ Need versioning.

→ 30 days mini. time interval.

→ Open Bucket, select management, under life cycle Add life cycle rule.

Enter rule name:

Storage class transition:

Click on add transition

then select transition type.

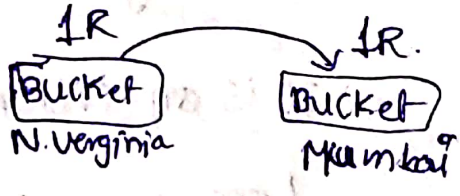
→ Expire time limit;

→ Expiry delete data, then save it.

S3 Cross region replication :-

→ one side only data

→ Automatic and asynchronous copying of objects across bucket in different AWS regions.



→ Requires versioning.

⇒ Create 2 buckets in different regions.

→ Both bucket enable versioning.

→ Satya-aws bucket open, management, Replication then add rule.

→ Replication Rule.

⊙ Entire bucket satya-aws.

→ destination, satya-devops.

→ IAM role :

Rule name : content, save it

⟹ Satya-aws → Source, satya-devops → destination.

S3 Glacier :-

→ Amazon Glacier is a storage service optimized for infrequently used data ~~for~~ (or) Cold data..

→ It is an extremly low cost. for data archieving and Long-term backup.

→ Glacier has various types to Retrival times.

* Standard : 3 to 5 Hours
* Buck : 5 to 12 Hours
* Expedited : 1 to 5 min (for small files)

→ To reduce costs, you can compress the files into a zip (or) tar format before uploading them to Glacier.

→ mini-storage 90 days.

→ simple object can sent to Glacier using life cycle management policies.

Vaults :-

→ Glacier uses "vaults" as container to store archive.

→ An archive can be data such as photo, video, document etc..

Archives :-

→ An Archives means TAR (or) ZIP files.

→ Data is stored in Amazon Glacier in "Archives".

→ An archive can be any data such as Photo, video etc.

→ A single archive can be large as 40TB.

→ Store an unlimited no. of archives.

→ Each archive is assigned a unique archive ID.

Benefits :-

→ Low cost
→ Secure
→ Durable
→ simple
→ Flexible
→ Integrated.

Use Cases :-

→ Media Asset Archiving.
→ Healthcare information Archiving.
→ Digital Preservation.
→ Scientific Data Storage.
→ Magnetic Tape Replacement.

→ Click on Amazon S3, Create Vault.

Vault name: my-Backup. Next, submit.

google: S3 glacier tools.

Download fast glacier, then install fast glacier.

→ Add new a/c section.

Access Key ID, and secret Access Key then add new a/c.

16/12

STORAGE GATEWAY

→ AWS Storage gateway is a hybrid storage service that enables your on-premises applications to seamlessly use AWS Cloud storage.

→ Use this service for backup and archiving, disaster recovery, cloud bursting, storage tiering and migration.

→ The gateway connect to

Storage Gateway Types

File Gateway:-

→ Storage files are objects in amazon S3, with a local cache for low latency access to your most recently used data.

→ Reduce on-prem storage.

Volume Gateway :-

→ Provides a block storage in S3 without in time backups as EBS snapshots.

→ Volume types:

Cached Volumes :-

→ Volumes held in S3 and freq. accessed volume held on-prem.

Stored volumes:-

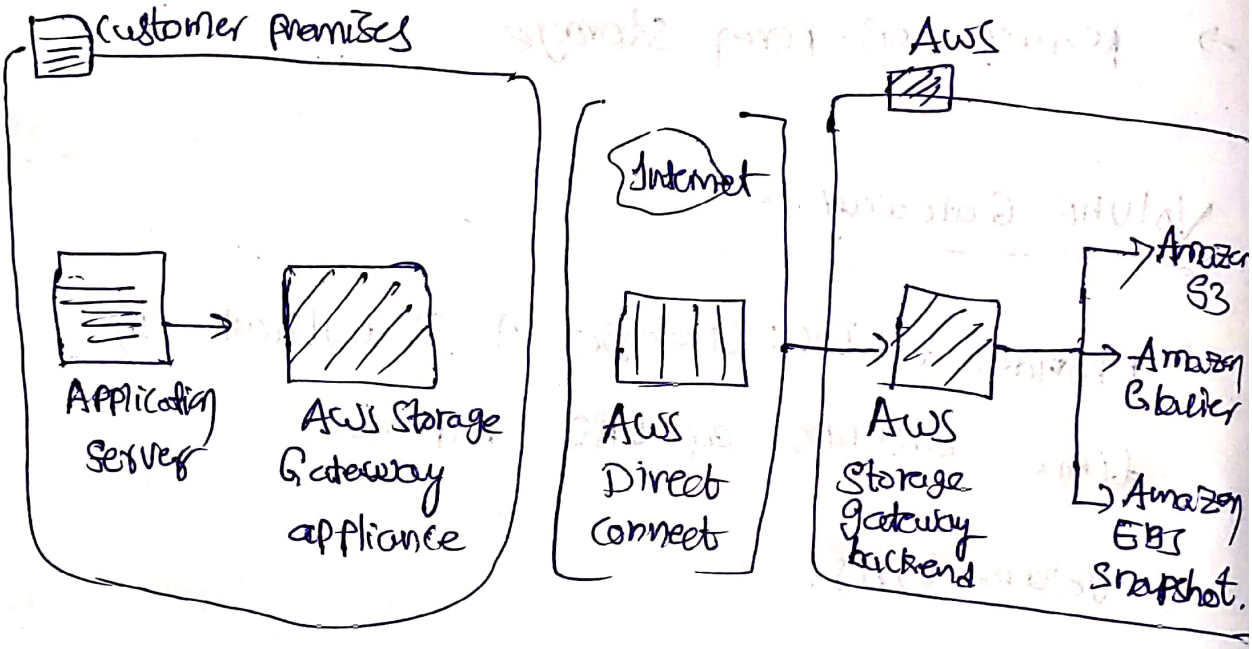
→ Volumes held on-prem and backup to S3.

Tape Gateway :-

→ Used for archive backup.

→ Supports popular backup applications from NetBackup, Veeam ... etc.

Storage Gateway Architecture



→ Go to amazon storage gateway.

→ Select file gateway,

→ service end point is select Public.

EC2 Amazon

→ If you collect storage EC2 instance.

→ m4.xlarge type is require

→ And storage 80gb and Port no. 2049 for NFS.

→ Go to Esxe server, connect exe server through ~~vsphere~~ client.

→ select file, Deploy OV template.

→ Browse file select template.

→ Set name: AWS appliance storage gateway.

→ select 'Thin' Browser Select 'Thin'.

→ change VM settings, right click on VM, change 4gb ram, 2CPU.

→ admin.
Password.

→ Gateway time zone select.

→ my file Gateway, Activate Gateway.

→ Create S3 bucket.

→ Go for S3, Set bucket name: Gateway Backup.

→ uncheck permission Buckets.

→ Go to AWS Storage Gateway

→ Create file Share,

Amazon S3 bucket name: Gateway Backup.

→ Then select NFS, click Next.

→ Storage class for new object : Standard.
Then next.

→ Next create file share.

→ Go to local linux machine, create a directory for mounting.

```
#mkdir /mysqw-backup.
#   (1).
```

/mysqw-backup.

```
# df -h
# cd /mysqw-backup.
# touch abc xyz.
```

→ Fileshare details. (1)

17/12

Data Base

Relational Database Service (RDS)

→ Amazon Relational Database service (RDS) is a webservice that makes it easier to set up, operate, and scale a relational database in the Cloud.

→ It provides cost-efficient and resizable capacity while automating time-consuming administration tasks such as h/w provisioning, database setup, patching and backups.

→ Doesnot allow access to underlying OS (fully managed)

→ you can enable multi-AZ deployment for backup and high avilable solutions.

→ Relational database are db that organize stored data into tables.

RDS example:-

→ Database → Tables → Records → Fields.

→ SQL interface: Structured Query language.

→ Extended functionality: Stored Procedure, triggers.

RDS Example :-

ID	Name	Age	Salary } → Fields
1	Ravi	34	20,000 } → Row
2	Alex	25	35,000
3	Raju	45	70,000

A single record in a table.

Column : A set of value of a specific type.

RDS Supports Databases :-

→ MySQL : Opensource

→ Maria DB : "

→ Postgre SQL : Opensource.

→ SQL Server : Commercial.

→ Oracle : Commercial.

→ Aurora : MySQL and Postgre SQL compatible

RDS Features :-

→ Automatic installation.

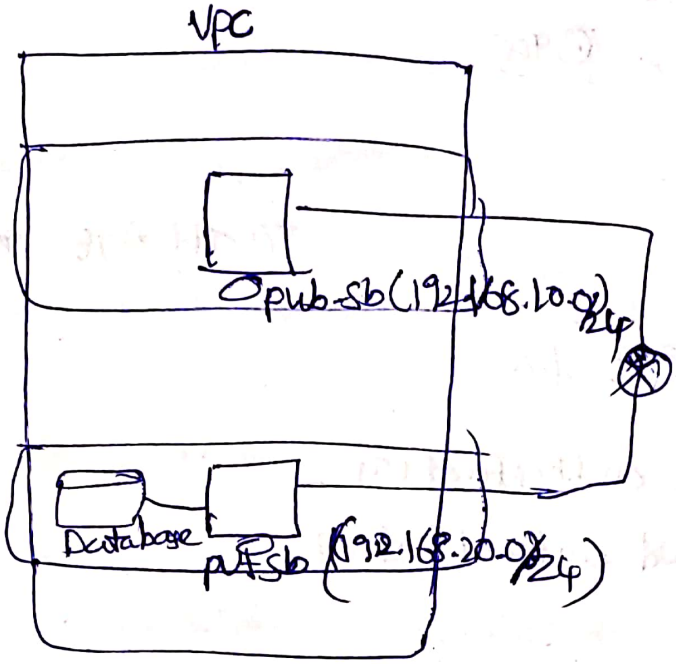
→ Automated backup / recovery.

→ Automatic Upgrade | Patches

→ " log management.

→ In-place scalling.

NOTE:- Database are only sitting Pvt. Sub net. in Custom VPC.



→ In RDS create database, select Standard Create.

→ Engine option, select MySQL.

→ Version : MYSQL 8.016.

→ template free tier.

→ Settings : MYSQL db instance.

→ Credential settings: Username : raju
Pwd : raj4123

→ DB instance type : db.t2.micro instance type.

→ Storage: select General Purpose-SSD
Storage : 20 GB.

→ Storage autoscalling : 1000 GB

→ connectivity : under PROD-VPC.

→ Create new DB subnet.
⊙ yes.

→ Prod-SG.

To add port number 3306
MySql Port no

→ AZ : us-east-1a.

→ Database authentication
P-wd authentication.

→ D/B option
Satya-db
default 8.0

→ Backup
automatic snapshot.

→ retention Period : 3days

→ Monitoring
☑ enable Enhanced monitoring
Log export
☑ Error log

* Free tier ISOH RDS
* 20GB general Purpose SSD
* 20GB of automated backup storage.

→ Add Port number

MYSQL/Aurora 3306 Anywhere. Prod-SG

18/12

Connecting database from windows :-

→ Install mysqul workbench 8.0CE in client.

→ MYSQL Connection ⊕ create

Name : Test

Hostname : end Point name as Hostname.

Uname : RAJU.

Connecting database from linux :-

→ Install mysql client

yum install mysql -client -y.

mysql -h mysqldb-instance.endPoint -P 3306

-u Raju -P.

Enter Password :

RDS Backups

Automated Backups :-

→ Allow point-in-time recovery within retention period 1 to 35 days.

→ Full daily snapshots and log storage.

→ Restoration :

* Both backups and snapshots can be restored.
* Create new RDS instance with new endpoint.

→ click the database, Actions, Take snapshot.

name : mysnap. → Take snapshot.

→ Restoring snapshot.

select snapshot., Action → last one.

Imp

Read Replicas (and backups)

→ Increase performance by providing additional read-only copies of database.

→ Aws automatically performs asynchronous replication.

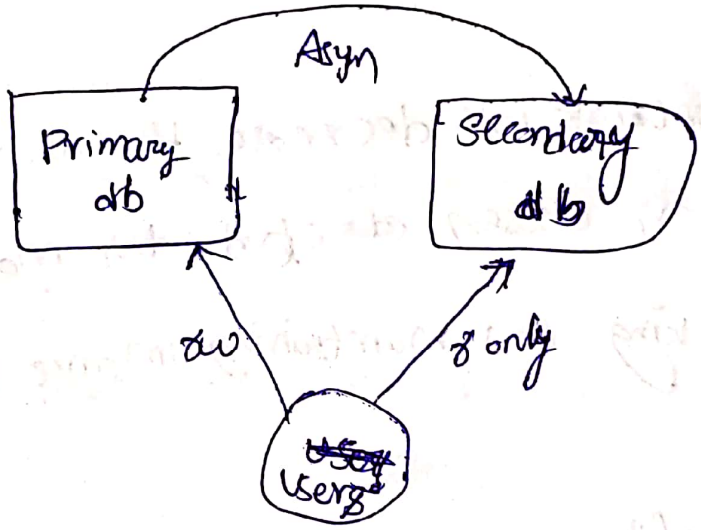
→ You can create, and multiple read replicas for a primary database.

→ you can monitor replication log using Cloud watch.

→ support MySQL, Maria DB, PostgreSQL.

→ upto 5 read replica instanles.

→ Read replicas can residep in other regiony.



→ ~~In Action~~
Click on database "Actiang", create read replica

→ Select instance type then, now select regiony.

<u>Modifying database</u> :-

19/12

Jmp

EC2 AUTO SCALING

→ Autoscalling helps you maintain application avilability and allows you to scale your amazon EC2 capacity up (or) down automatically according to condition you define.

→ Autoscalling is increase (or) decrease the no. of instances based on chosen cloudwatch metrics.

→ Automatic tracking and maintaining instance pool.

Features and Benfits :-

→ Main your Amazon EC2 instance avilability.

→ Automatically scale your Amazon EC2 Fleet.

⇒ Go to EC2, click on Launch configuration.

→ Config. details.

Name: Autoscalling

☑ Enable cloud watch

User data - (•) As text

```
#!/bin/bash.
yum update -y
yum install httpd -y.
```

→ Create launch config.

→ click on create an autoscalling group using launch config.

→ My auto scale group : Name group

→ Nw : defaut VPC

→ Subnet : add subnet 1.

go to next.

→ select, use scalling policy to adjust the capacity of the group.

→ click on, Scale the auto scalling group.

Under Increase group size, click add new alarm.

Create alarm

☑ sent notification : gmail.com

[75] percent.

Create Alarm.

Decrease group size

Add new alarm

Create alarm.
Avg CPU utilization.
15 <= 15 percent.

Name of alarm : Autoscale group.

Create alarm.

Take action : Remove ☑

Click on config. notification.

→ Create autoscalling group.

launch.

→ Config. Tags value
Scalling.

Review create.

→ Go to gmail verify notification's..

→ Connect instance, use Putty.

Imp

ELASTIC LOAD BALENCER (ELB)

→ Elastic load balancing automatically distributes incoming application traffic across multiple Amazon EC2 instances. This increase the fault tolerance of your applications.

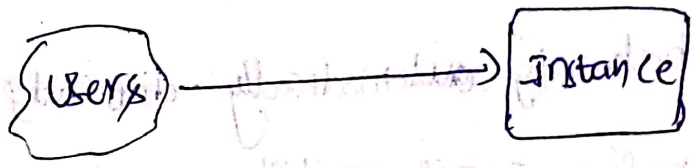
→ An ELB load balance traffic ~~multiple to~~ multiple EC2 instances located across multiple AZ's.

→ ELB should be paired with autoscalling to enhance high avilability and fault tolerance, and allow for automated scalability and elasticity.

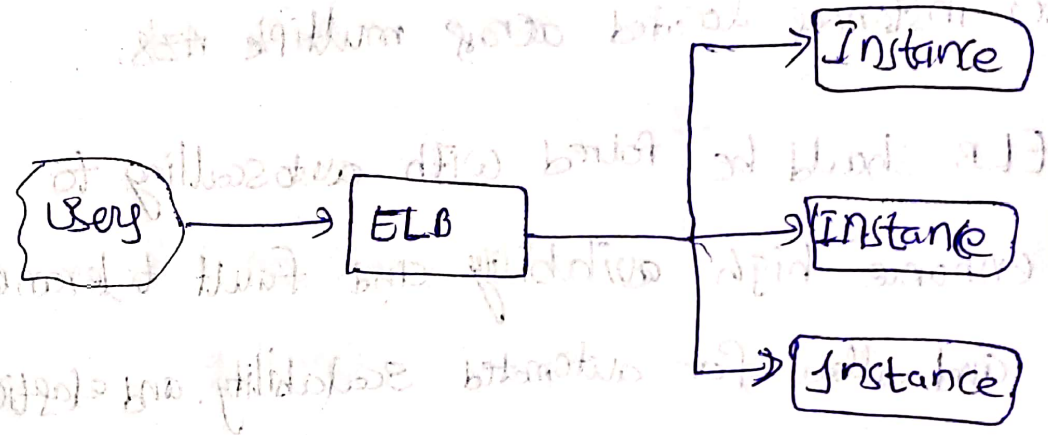
→ An ELB has its own DNS record set that allows for direct access from the open internet access.

→ ELB's automatically stop serving traffic to an instance that becomes unhealthy.

AWS application infrastructure with a single EC2 instan



Application AWS infrac. with ELB and multiple EC2 instances:



Health Checks :-

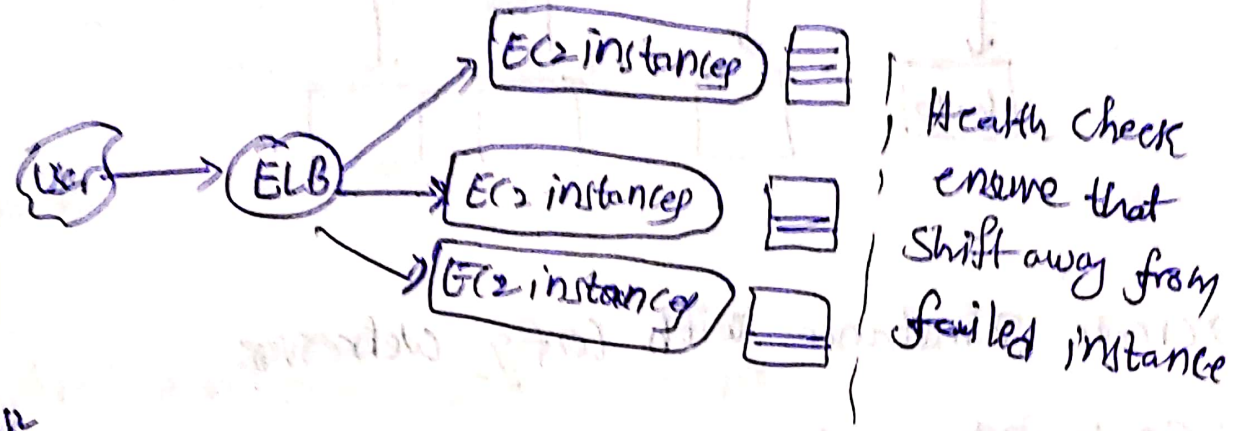
→ Health checks are crucial for load balancer.

→ They enable the load balancer to know if instances it forwards traffic to are ~~traffic~~ avilable to reply to requests.

→ The health check is done on a port and route (/health is common).

→ If the response is not 200 (OK), then instance is healthy..

Health Checks



20/12

ELB types (Products):-

→ ELB supporting following types of load balancer:

→ Classic Load Balancer (CLB)

→ Application " " (ALB)

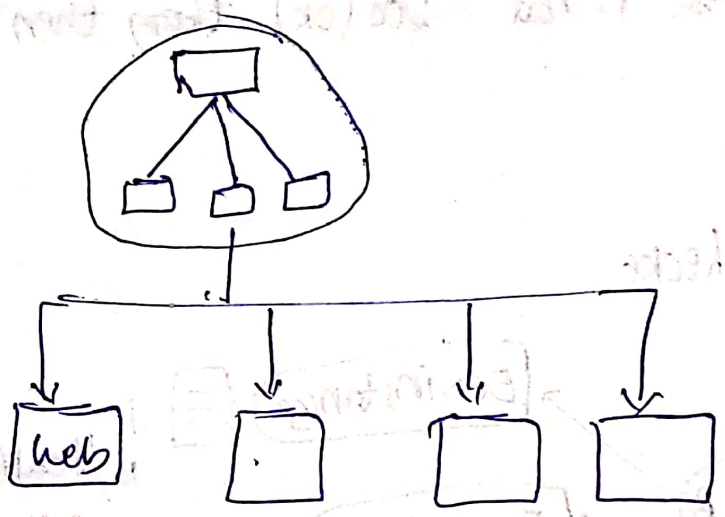
→ Network load Balancer (NLB)

Classic Load Balancer (CLB) :-

→ A CLB is designed for simple balancing of traffic to multiple EC2 instances.

→ CLB is best used when all instances contains the same data.

→ There are no granular routing "rules".



→ Create 2 instance with config. webserver.

→ Go to instances launch instance

→ Config. instances

: No. of instances : 2

→

→ Connect instance Test2, use Putty.

#Cd /var/www/html

Vi index.html

Vi test.html

#Creating load balancer :

Create load balancer : Classic load balancer.

LB Name : MY ELB

→ select existing SG.

→ Config. Health check

Ping. Path : /index.html

→ Click add EC2 instances.

Cross load avilable

cross region not avilable.

→ Review and create.

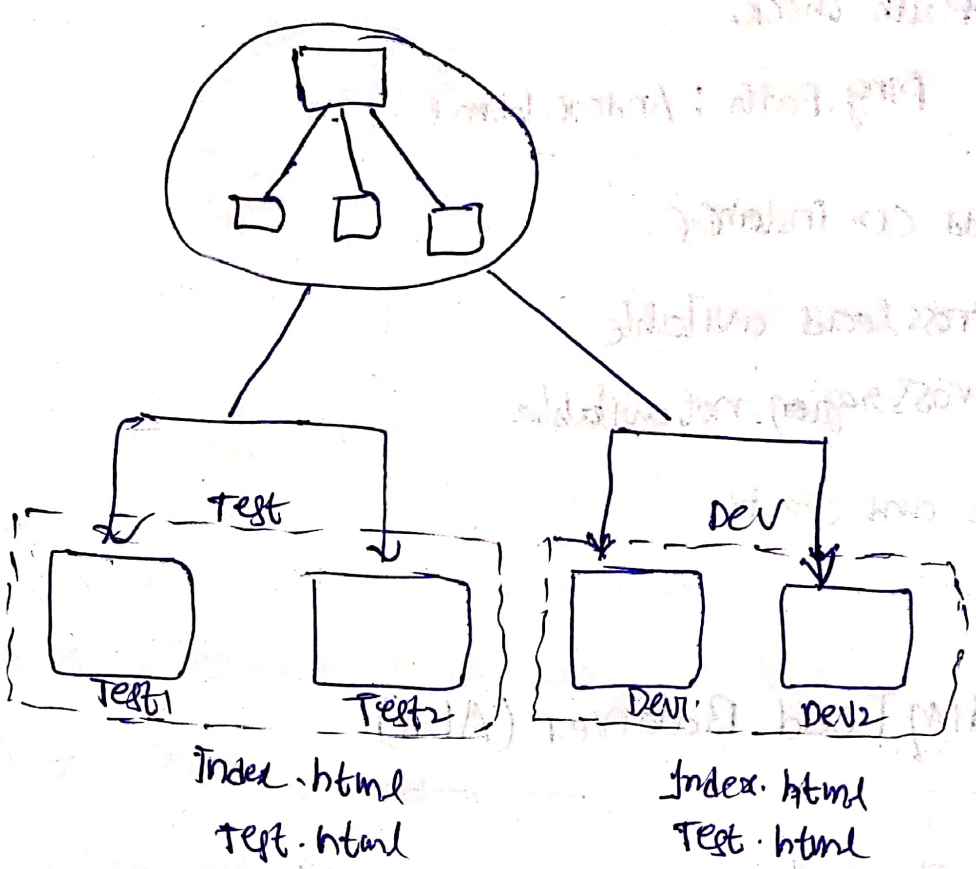
Application Load Balancer (ALB)

→ An "Application" ~~ALB~~ elastic load Balancer for complex balancing of traffic to multiple EC2 instance using "Content-based" rules.

→ * Host-based rule : Route traffic based on the host field of the HTTP header.

* Path-based rule : Route traffic based on the URL Path of HTTP header.

→ An application ELB also support ECs containers, HTTP, HTTP/2, Websockets, Access logs, sticky sessions, and web application firewall.



→ Launch 2 instances and install and config. webserver.

→ Create web pages, index.html, Test.html.

→ Create target group with name Test and add Test1 & Test2 instances.

→ Go to load balancer, Create group.

T.group : TEST

Protocol : HTTP

Path : /index.html.

→ select target group and choose target and edit.

Add to register and save it.

→ Create another 2 instances and install and Configu. webserver.

→ Create target group name.

Target name : DEV

Add dev1, dev2, Add to register and save it.

→ Creating load Balancer,

Go to Load balancer, create it.

select application load balancer.

Name : MYAPPELB

IP address : IPV4

click on config. settings.

→ select subnet click on next., select SG,

→ Click on Config. routing.

Target group : Existing target group

Name : DEV

click on next, Review & create it.

21/12

Hosting rules:-

- In the load balancer select MYAPPLB, under listening ports, View edit rules under edit rule ⊕ Add rule. Click on ⊕ insert rule.

if (All match) Then

Path: *dev* forward: Dev.

Add Condition if (All match) Then

Path: *test* Forward: Test

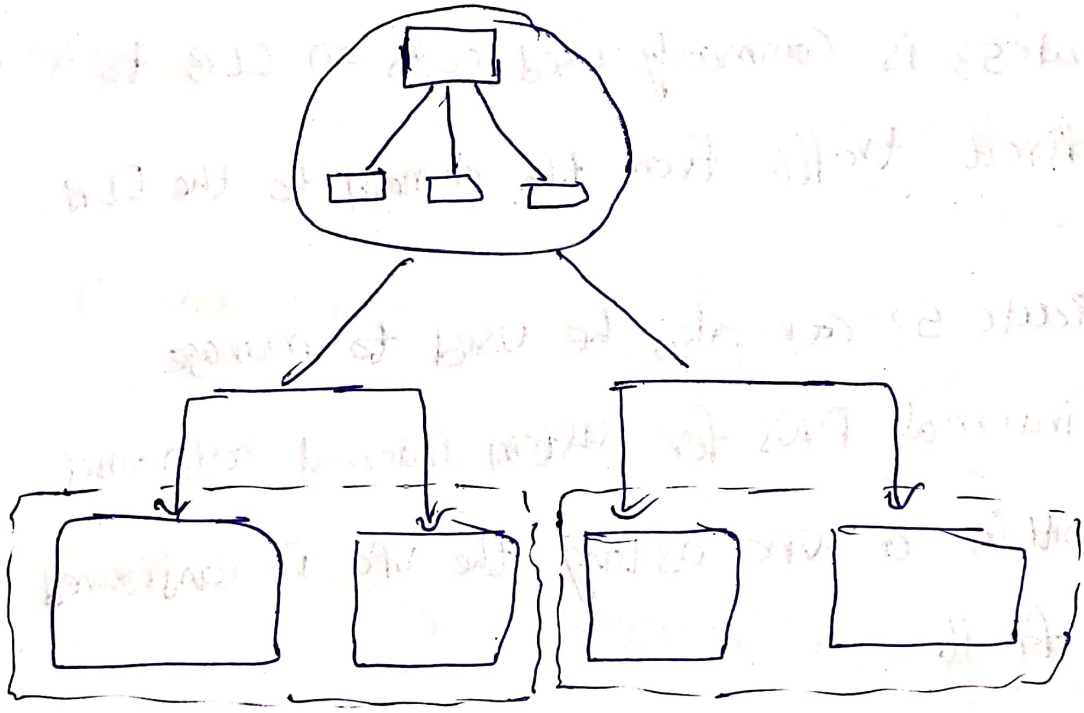
To Change Default target

→ under load balancer, select ☑ HTTP 80 and edit

1. Forward
   Test

Network Load Balancer (NLB)

- Forward TCP traffic to your instances.
- It can handle millions of request per second.
- Support for Static IP and Elastic IP.
- N/w Load Balancer are mostly used for extreme performance.
- Overall, the creation process is same as Application load Balancer.



Imp

Traffic Controlling

Amazon ROUTE 53

→ Route 53 is a domain Management service (DNS hosting solution) Provide by AWS.

→ Amazon Route 53 Provides highly available and Scalable Domain Name system (DNS), domain name registration, and health-checking web services.

→ Route53 is Commonly used with an ELB to direct traffic from the domain to the ELB.

→ Route 53 can also be used to manage internal DNS for custom internal hostnames within a VPC as long the VPC is configured for it.

Key Features :-

* Domain Registration.
* DNS Service.
* Traffic Management.
* Health checking.

Common Records Sets

→ A (Address Record) : Point a domain to an IPV4 address.

→ AAAA (IPV6 add. record) : " " a IPV6 address

→ CNAME (Canonical name record) : Translates one domain name to another.

→ MX (Mail exchange record) : Route Email

→ NS (nameserver record) : Nodes that hold information about given name.

→ PTR (Pointer Record) : Reverse DNS record (opposite to what a record does)

→ SOA (Start of Authority record) : manager of DNS. overall infor. about domain.

→ TTL (Time to Live) : storage time for cached records.

→ Register a public domain under Amazon, Godaddy any other.

→ Login yr domain Provider website and select nameservey and change it.

→ Enter my own name servers.

→ Go to Route53,

→ select DNS management.

→ Create hosted zone. Domain name:

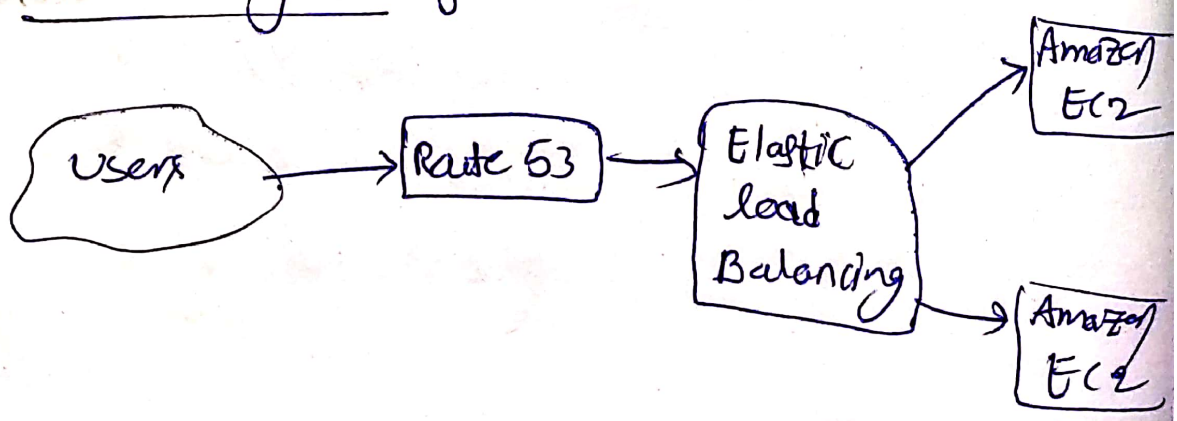
TYPE: Public hostzone

Then Create it.

→ Copy 4 name servey and Paste name servers (Domain name)

Creating Policies :-

Simple Routing Policy: Route all traffic control Point.



→ Create Record set.

Type : A-record

Alias : yes

Alias Target : MYAPP ELB.

→ Create routing Policy last step.

→ create health checks :-

Under health check.

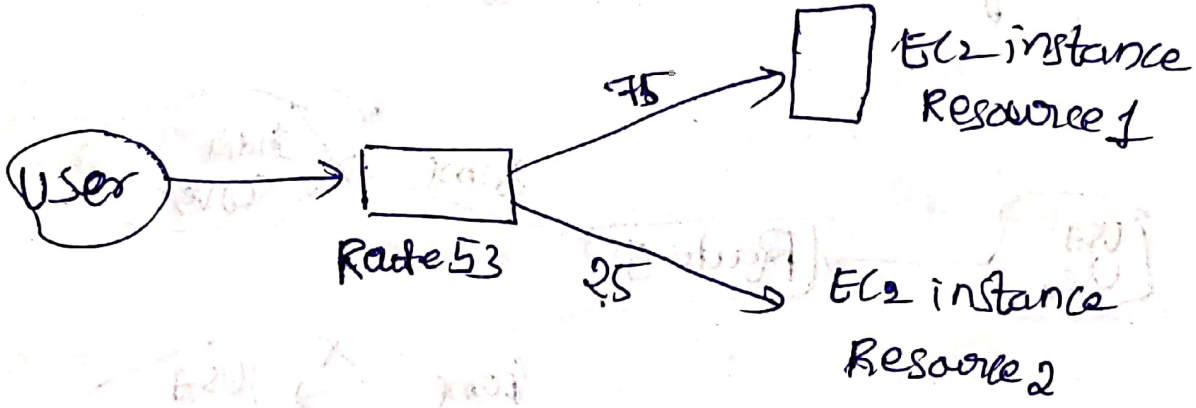
Name : my domain health.

Monitor endpoint : select Domain name

Domain name : Name

create health check.

Weighted :- Route traffic to multiple resources in proportions that u specify.



Create Recordset :-

TYPE : A - IPV4.

Alias : NO.

Value : Test1 IP add.

weight : 75.

Create another record set.

TYPE : IPV4 add.

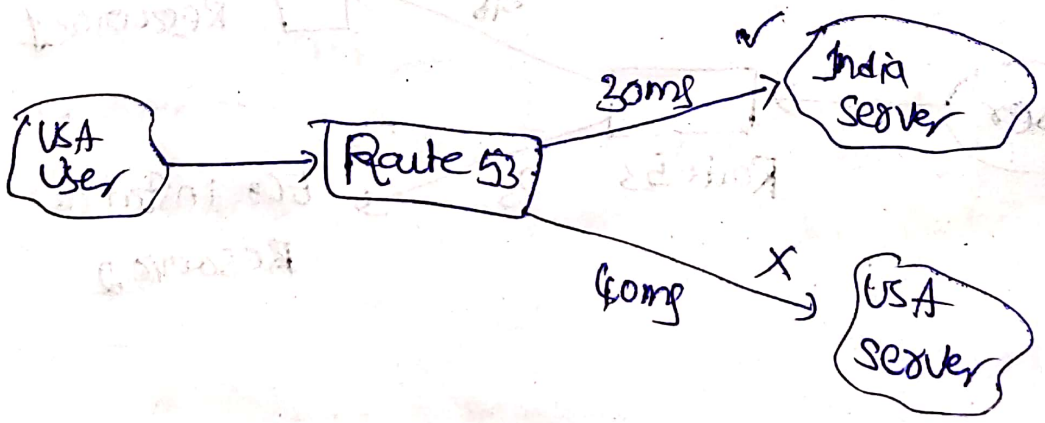
Alias : NO.

Value : Test2 IP add.

weight : 25

set ID : Test2

Latency : Route traffic to an endpoint based on user latency to various endpoints.



→ Creating 2 webserver in different regions.

→ Create record set:

Alias : NO

Routing policy : Latency.

Region: us east 1

SetID : US

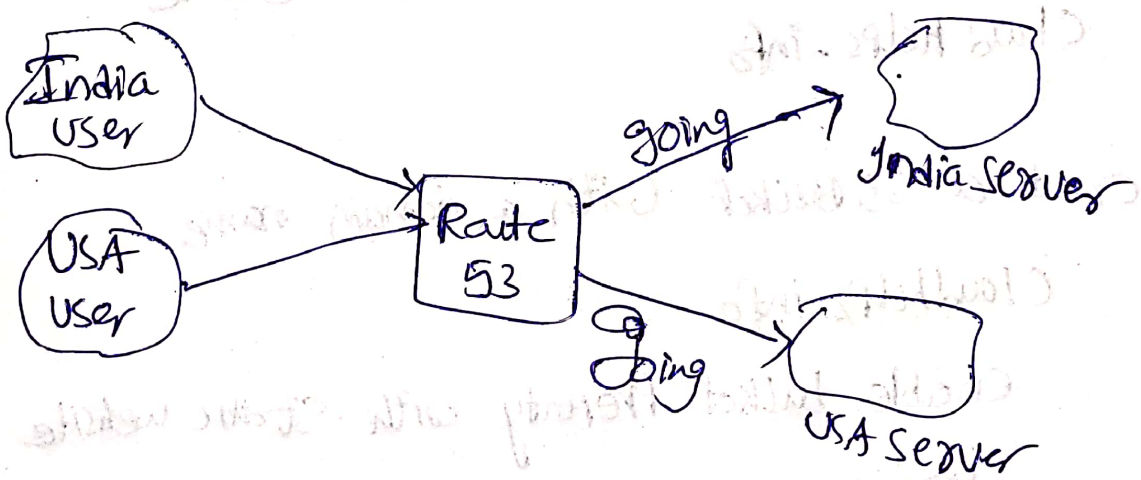
→ Create Recordset:-

Routing poliy : Latency.

Region : ap-southeast-1

SetID : MUM.

Geolocation:- Route traffic to an endpoint based on Geographical location of the user.



Create recordset.

Routing policy : Geolocation.

Location : Usverginia

Set ID : US

Create record set:-

Routing policy : Gelocation

Location : India

Failover:-

→ Route Traffic to a "secondary" endpoint if the "primary" is unavilable.

→ S3 for DNS failover.

→ Create a static website with bucketname Cloudhelps.info

⇒ Create a S3 Bucket with a domain name Cloudhelps.info.

enable bucket Property with Static website hosting.

→ Set up Bucket Policy Public.

→ Go to Route 53

Alias : Yes

Alias target : ~~S3 Bucket~~ MYAPPELB

Route Policy : failover

Evaluate target : Yes

: NO.

→ Create Record set.

Alias target : S3 Bucket

Route Policy : failover

Evaluate target : Yes

: NO

//

23/19

~~EBS~~ Elastic Beanstalk

→ AWS Elastic Beanstalk makes it even easier for developers to quickly deploy and manage applications in the AWS cloud.

→ It is a developer centric view of deploying an application on AWS.

→ Just the application code is the responsibility of the developer.

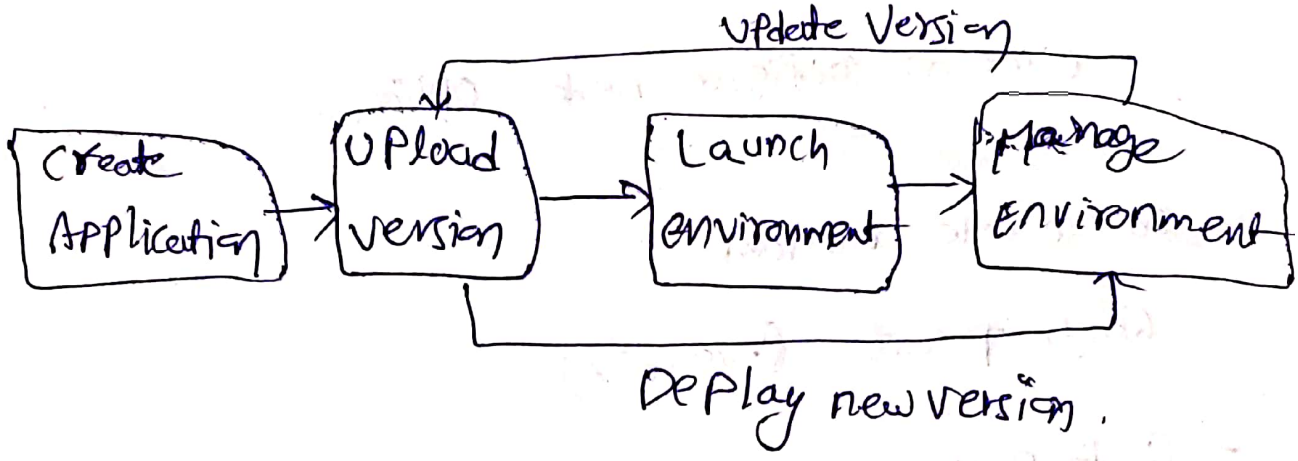
→ It use all components we've seen before EC2, ELB, ASG, VPC, RDS -- etc

→ Beanstalk is free but you pay for the underlying instances.

→ Three architecture models:

* single instance deployment : good for dev.
* LB+ASG : great for Production Pre-Production web applications.
* ASG only : great for non-web apps in Production.

→ workflow of Elastic Beanstalk



Beanstalk supported platform:-

Go	Java SE
Java with Tomcat	.NET for windows server with IIS
Node.Js	PHP
Python	Ruby
Packer Builder	Single container Docker
Multi container Docker	Pre Configured Docker.

If not supported, you can write your Custom platform (advanced)

→ Go Aws Console

→ Elastic Beanstalk. select.

→ Getstarted.

App name : Tomcat

Platform : Tomcat

Application Code : ⊙Sample Code.

click on configure node options.

Source code download.

Google Tomcat (.war file)

→ Go to Dashboard of Beanstalk, upload and deplay.

→ Access URL After deplay application.

→ Create new application.

Go to Elastic Beanstalk dashboard create new application.

App. name : Python.

: Python

Create.

Actions → Create environment.

Webserver environment.

24/12

Amazon Lightsale

→ Lightsale is the easiest way to get started with AWS for developers who just need virtual private servers.

→ Lightsale include a virtual machine, SSD-based storage, data-transfer, DNS Management, and static IP for a low, predictable price.

→ Lightsale is for developers. You don't have spend that much time installing s/w (or) frameworks.

→ use Amazon lightsale for:

* simple web applications.
* websites.
* Business s/w.
* Dev/Test environments.

Features:-

→ Lightsale virtual servers.
→ simplified load balancing.
→ Managed databases.
→ upgrade to EC2.
→ Access to AWS services.

→ Go to Aws console.

→ Click on lightsale, Go to instances.

→ Select instance, Create instance.

→ If u change ~~Aws~~ Instance location.

→ Select Platform.

Linux

→ Select blueprint.

LAMP (PHP)

→ Your Create Keypair

⊙ Default key.

→ ☑ Enable Autosnapshots. (depends on)

→ Choose instance 1st month free.

→ Identify instance.

LAMP_STACK.

→ Convert .pem file .PPK.

→ Connect instance.

$ Cat bitnami_application_Password.

$ Cat bitnami_Credentials

Connecting Lampstack.

→ Copy instance IP Address, Paste on Browser.

→ click on Access Phpmyadmin.

→ open Putty, Paste Public IP address load your Key Pair.

→ Auth, Tunnel : Source Port : 8888.
localhost:80
Click on Add it open.

SERVER LESS

→ Serverless is new Paradigm in which the developer don't have to manage servers anymore.

→ They just deplay code and deplay... functions.

→ Initially... serverless means function as a service (FAAS).

→ serverless way pioneered by Aws Lambda but ~~two~~ now also includes anything that's managed. "databases, Managing, storage, etc..."

→ Serverless doesn't mean there are no servers.

it means you just don't manage / provision / see them.

Traditional	Serverless
→ "Traditional" web hosting	"Serverless" web hosting.
Provision Capacity	Runs on-Demand
How much server capacity? CPU, RAM, Memory ⇓ Scalling (Pay too much or server issues)	Unlimited capacity only pay for code execution ⇓ Scales Automatically (Pay what u need)
Update OS & Sw	Runs on managed Aws infrastructure.
Prevent security issue by thraughfully update sw.	Code runs up-to-date and secure environment
Lots of overhead for SPA + API apps	
Great support for fullstack apps.	Limited support.

25/12

LABDA

→ Aws Lambda is a compute service that lets you run code without provisioning (or) managing servers.

→ Aws Lambda excute your code only when needed and scale automatically, from a few requests per day to thousand persecond.

→ you pay only for the compute time you consume - there is no charge when u code is not running.

→ youareresponsible for your code.

→ It runs your code on high-avilability compute infrastructure and performall the admin. of compute resources.

Languages support :-

→ Node.js (JS)

→ Python

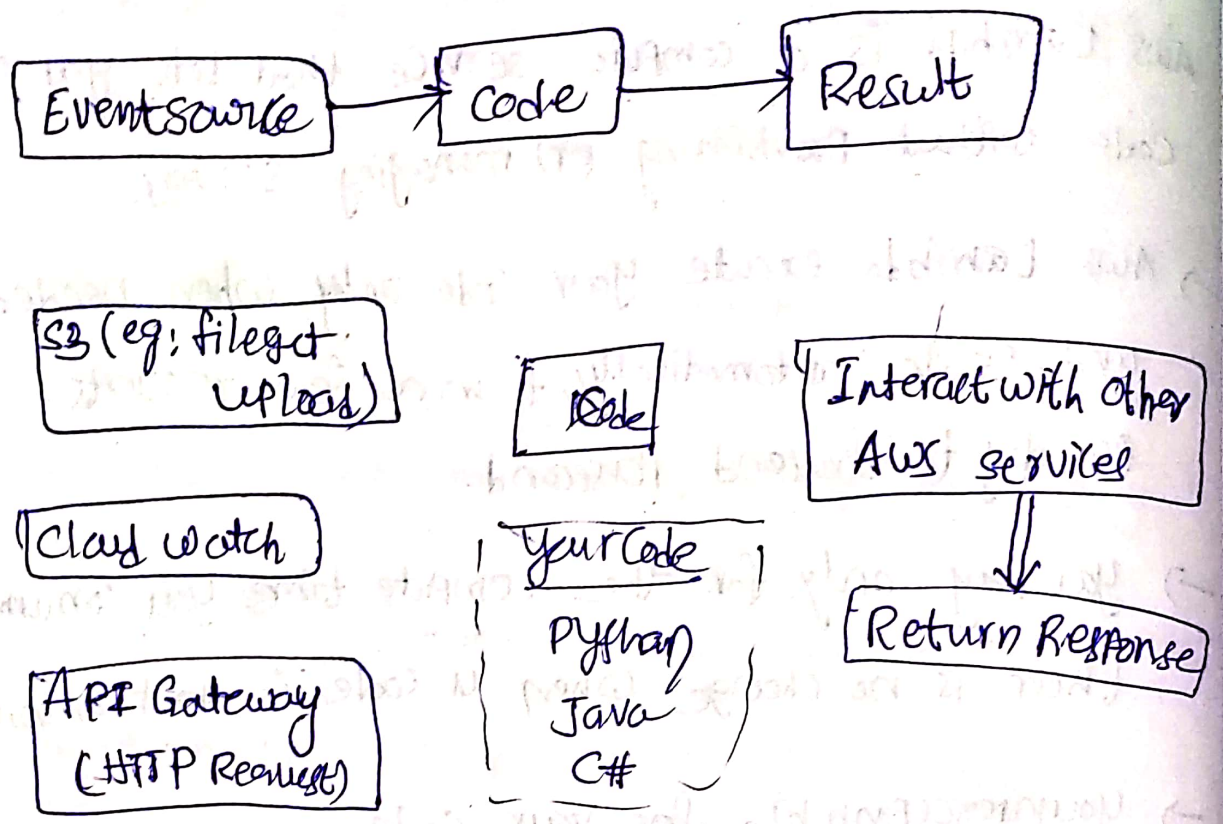
→ Java (Java 8 compatible)

→ C# (.NET)

→ Golang

→ C# / powershell

How it works



→ Go to AWS Console.

→ Select Lambda, Create function.

Function name : My first function.

Runtime : Python.

⊙ Create new role with basic Lambda function then Create function.

DYNAMO DB

→ Amazon Dynamo DB is fast and flexible NOSQL, database service for all applications that need consistent, single digit millisecond latency at any scale.

→ Fully managed, highly available with application across 3 AZ.

→ It is not a relational database.

→ It is fully managed cloud database and support both document and key-value store models.

→ Integrated with IAM security, authorization and administration.

→ Low cost and auto scalling capabilities.

DynamoDB Architecture :-

→ SSD storage.

→ Replicated in three facilities.

→ Support cross-region application.

→ Unlimited capacity.

→ User chose performance via read/write capacity units.

DynamoDB Components :-

-> Tables.

-> Items (Rows)

-> Attributes (Fields in Key/Value Pairs)

* Simple : First name : John
* Nested : Address : {

-> Number : 123

-> Street : Main

-> City : Anywhere.

Primary Key :-

-> Item identifier defined at table creation.

-> Partition key : Determines physical placement of item

-> Partition key + Sort key :- Places items on same storage.

Secondary Index :-

-> Defined on attribute.

-> Used for queries.

-> Can be create at any time.

→ Aws Console, Database → DynamoDB.

→ Create table.

Table name : ~~Employee~~.
Employee

~~Partion~~
Primary key : ENO Number

→ Backups, DynamoDB maintains continuing backup of your tables it supports point in time recovery.

→ select table Choose backups and create backup.

→ select backup, restore backups.

26 In

Cloud Formation

→ Cloud formation is an infrastructure as code.

→ It provides common language for you to describe and provision all the infrastructure resources in your cloud environment.

→ It allows you to ~~see~~ use text file to model and provision, in automated and secure manner, all the resources needed to your application across all region & A/C.

→ you can spend less time managing those resources, and more focusing on your applications.

→ Cloud formation can:

* Simplify infrastrumanagement.
* Quickly replicate your infrastructure.
* Easily control and track changes to your infrastructure.

Cloudformation Components

Template :-

→ A Template is JSON-Formatted text file that describe your AWS infrastructure.

→ Templates can include several major section. Such as format version, Description, Metadata, Parameters, Mapping, Condition, Resources, etc.

Stack :-

→ A stack is collection of AWS resources that you can manage as a single unit. In other words, you can create, delete, update a collection of resources by creating, deleting or

Updating tasks.

→ A stack is implementation of your template.

Prog

→ Install Notepad++.

→ Create S3 bucket.

→ Management & Governance, → Cloud formation.

→ Create stack. → Prepare Template.

⊙ Select upload a template.

→ Choose file, upload file click on next.

→ Stack name.

My first stack.. then click on next.

→ Click on next, create stack.

Updating stack:-

→ Select My first stack., click Check update, Use ~~Current~~ Replace template.

→ Upload a template file, choose file.

→ Update stack.

VPC Creation with help of cloud formation (VPC deployment)

→ Create VPC Template, Aws cloudformation Templ
Save .JSON file.

→ Create Stack, select Template is ready, upload Template file, Choose file select 'MyVPC' upload template file.

Stackname : MyVPC

→ Parameters:

Dev-VPC

10.10.20.0/24

10.10.25.0/24, Click on text.

→ Create stack.

Stack Set :-

→ A stackset is container for Aws cloudformation stacks that lets you provision stacks across Aws A/c's and regions by using a single Aws Cloudformation Template.

→ A stack is a set of regional resource. If you create a stack set in one region, you cannot see it (or) change it in other regions.

→ Click on stack sets, Create stack sets, upload template file, choose file. ~~upload~~ Click Next.

→ Stack name : MYVPC.

Environment : Test VPC., Click next.

→ Add AIC no. here.

specify region., No. of AIZ's 1

click on next.

27/12 Continue.

Cloud designer :-

To create without code.

Cloud Formerr

→ Cloud former is template creation beta tool that creates an AWS Cloudformation template for existing AWS resource in your AIC.

→ This means u can capture and deploy applications you already have running.

AWS ORGANIZATIONS

→ It is an A/c management service that enable you to consolidate multiple AWS Accounts into an organization that you can create and centrally manage.

→ As an administrator of an organization, you can create a/c's in your organization and invite existing Accounts to join the organization.

Features:

→ Manage & define your organization and accounts.

→ Control access and permissions.

→ Audit, monitor and secure environment for compliance.

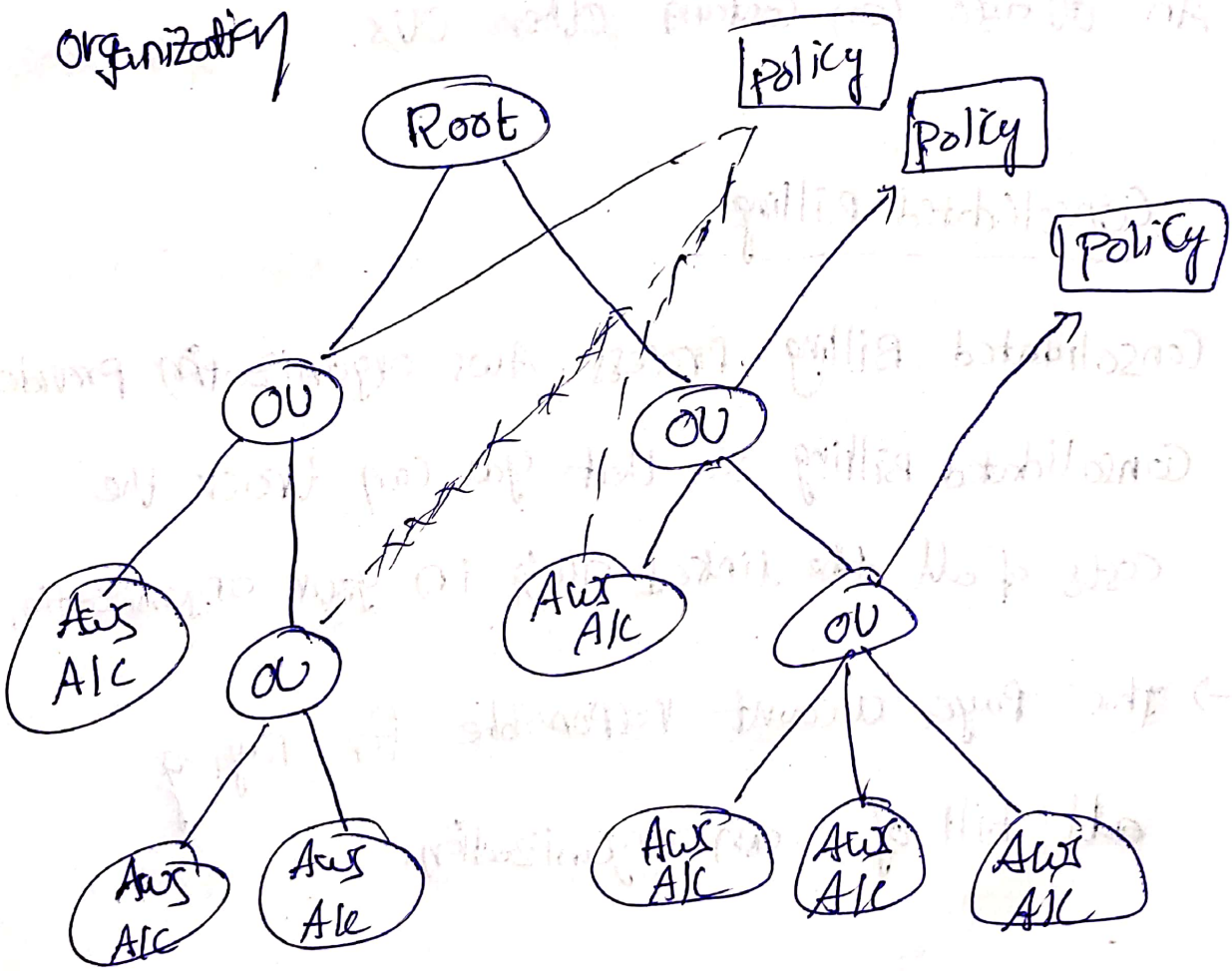
→ Share resource Across A/c's.

→ Centrally manage cost & billing.

Aws organization Terminology

→ The diagram show a basic organization to consist of accounts that are organized into organizational units (OU's under the root.

→ The organization also have several Policies that are attached to some of the OU's ~~or~~ (or) directly to accounts.



Organization :-

Organization is an entity that you create to consolidate your AWS Accounts.

Root:- The parent container for all the accounts for your organization.

Organization Unit (OU's):-

A Container for Accounts within a root.

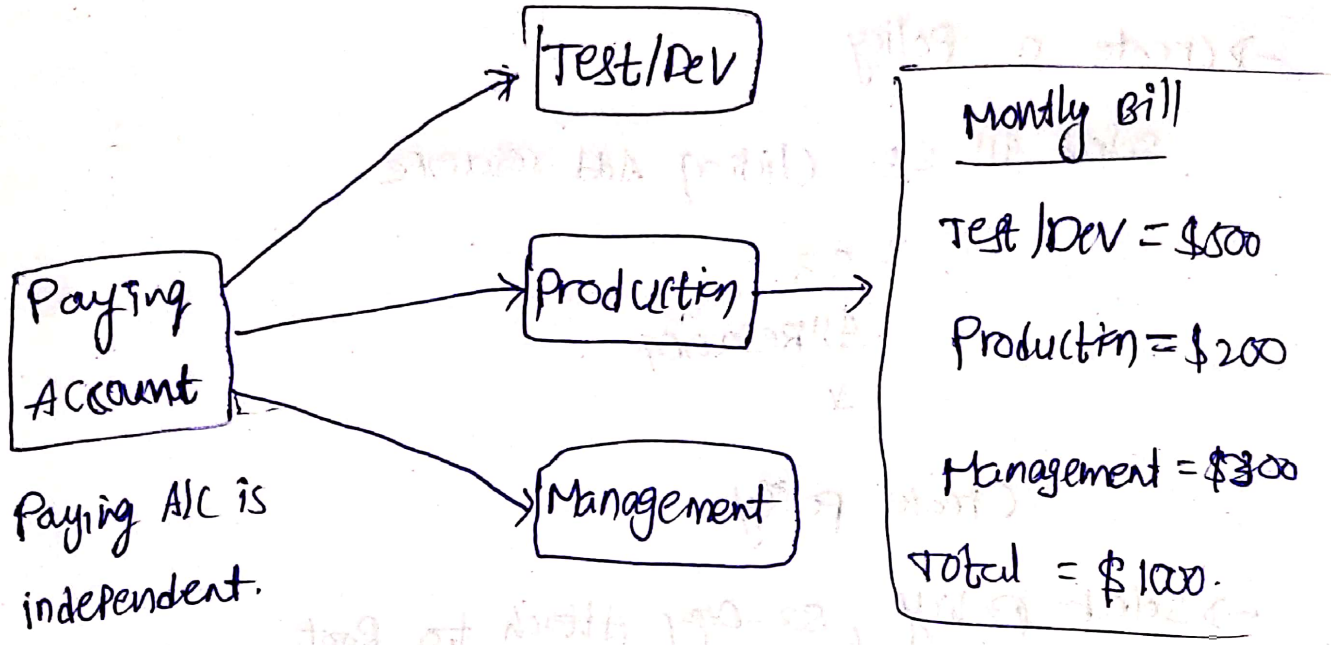
An OU also can contain other OU's.

Consolidated Billing :

→ Consolidated Billing Process. AWS organization provide Consolidated Billing. So that you can track the costs of all the linked a/c's in your organization

→ The Payer account responsible for paying all bill of an organization.

Linked Accounts



Paying A/C is independent.

cannot access resource of other a/cs.

All linked A/cs are independent

Currently, a limit of 20 linked a/c's for consolidated Billing.

AWS organization :-

→ Add A/c, invite A/c, Enter A/c ID.

→ Create organized A/c (OU) :-

New organization unit.

DEV-OU., Another create PROD-OU.

Adding A/c OU's :-

→ select A/c, move.

Policies :-

select service control Policies.

→ Create a Policy.

select All S3, Click on Add resource.

: S3
: All Resource
: *.

Create policy.

→ Select Policy, S3-Deny Attach to Root.