# AWS

Before cloud computing every organisation has, they own assets (visual machine etc) which manages by visualization team by working with DCOPS team

# Cloud computing:

They are two types:

1. Private – it is for the organizations (big company)
2. Public - it is for the public

* Aws
* Azure
* Google cloud

# **What is cloud computing?**

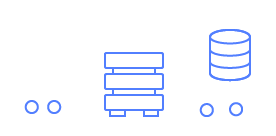
Cloud computing is the on-demand delivery of IT resources over the Internet with pay-as-you-go pricing. Instead of buying, owning, and maintaining physical data centers and servers, you can access technology services, such as computing power, storage, and databases, on an as-needed basis from a cloud provider like Amazon Web Services (AWS).

Benefits of cloud computing:

1. Agility
2. Elasticity
3. Cost saving
4. Deploying globally in minutes

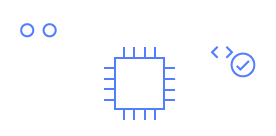
# **Types of cloud computing**

The three main types of cloud computing include Infrastructure as a Service, Platform as a Service, and Software as a Service. Each type of cloud computing provides different levels of control, flexibility, and management so that you can select the right set of services for your needs.



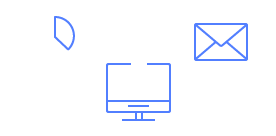
## **Infrastructure as a Service (IaaS)**

IaaS contains the basic building blocks for cloud IT. It typically provides access to networking features, computers (virtual or on dedicated hardware), and data storage space. IaaS gives you the highest level of flexibility and management control over your IT resources. It is most similar to the existing IT resources with which many IT departments and developers are familiar.



## **Platform as a Service (PaaS)**

PaaS removes the need for you to manage underlying infrastructure (usually hardware and operating systems), and allows you to focus on the deployment and management of your applications. This helps you be more efficient as you don’t need to worry about resource procurement, capacity planning, software maintenance, patching, or any of the other undifferentiated heavy lifting involved in running your application.



## **Software as a Service (SaaS)**

SaaS provides you with a complete product that is run and managed by the service provider. In most cases, people referring to SaaS are referring to end-user applications (such as web-based email). With a SaaS offering, you don’t have to think about how the service is maintained or how the underlying infrastructure is managed. You only need to think about how you will use that particular software.

# AWS Infrastructure

Region

Availability zones—inside many data centers.

<https://aws.amazon.com/about-aws/global-infrastructure/?p=ngi&loc=0>

at now 32 regions 102 availability zone

region inside( 2 or more availability zones inside (many data centers)))

AWS has over 200 fully featured services for a wide range of technologies, industries, and use cases.

In devops:

Compute , storage, database, transfer, network, developer tools, manage and governances, secops services

# EC2

## EC2 Features

EC2 provides web services API for provisioning, managing, and deprovisioning virtual servers inside amazon cloud.

➢ Ease In Scaling Up/Down

➢ Pay only for what you use

➢ Can be integrated into several other services

## Ec2 Pricing

* On Demand Pay per hour or seconds.
* Spot Bid your price for unused ec2 capacity.
* Reserved Reserve Capacity(1 or 3 yrs) for discounts.
* Dedicated Hosts Physical Server dedicated for you.

AMI: Amazon Machine Image (AMI) provides the information required to launch an instance, which is a virtual server in the cloud.

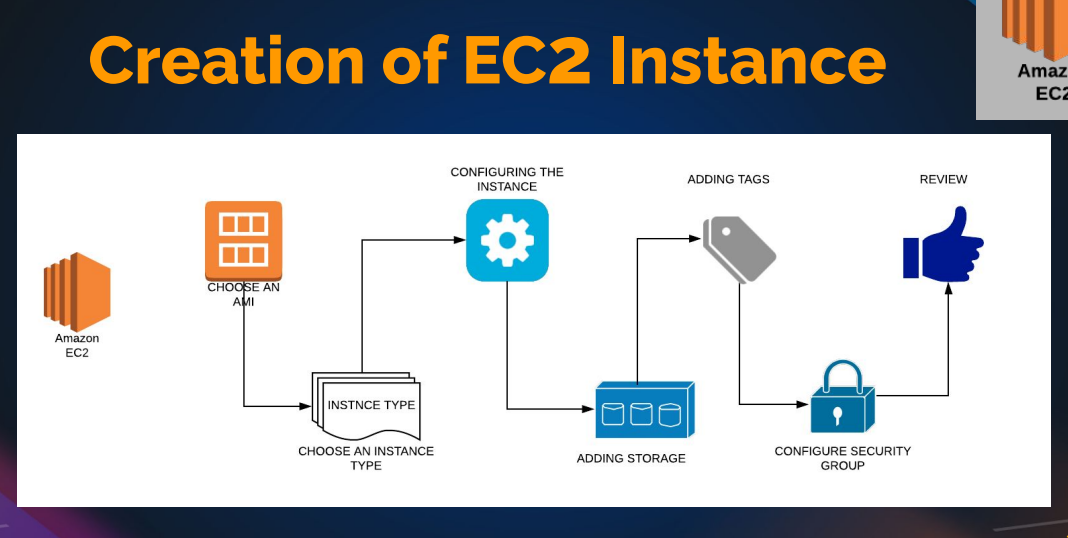
INSTACES TPYES: When you launch an instance, the instance type that you specify determines the hardware of the host computer used for your instance

EBS: amazon elastic bock store(EBS) Amazon EC2 provides you with flexible, cost effective, and easy-to-use data storage options for your instances.

Tag :Tag is a simple label consisting of a customer-defined key and an optional value that can make it easier to manage, search for, and filter resources.

security group: A security group acts as a virtual firewall that controls the traffic for one or more instances

public–key: Amazon EC2 uses public–key cryptography to encrypt and decrypt login information.



* Hand-on: go to ec2 instances and create a instances in free tiar and create key-pair(ssh, http) and create security group .

Do the proper Tags

Change/give the instances name and description

Add the security group

Check the network interface

* Do the provision step startbootstap and launch the website
* Launch website using both ssh and public Ip
* Create a elastic Ip and use it (it will get some cost)
* Action 🡪 change instances type ---> to increase /decrease the site of the instances
* Check the system log ----> in monitor & networking

To check the open ports : ss -tunlp | grep 80

8080 is Apache port number to assign it

# EC2 INSTANCES CRATION

Gathering information

1. Key pairs

2. Security Group

3. Instance Launch

# Gathering information

Toolplate website

2. OS a. Centos

3. Size => Ram, CPU, Network etc a. Min

4. Storage size a.

5 gigs for web server images

5. Project

6. Services/Apps Running a. SSH, Http, Mysql etc

7. Environment (Dev, QA, Staging, Prod) 8. Login User/ Owner

# Aws cli

IAM 🡪 user 🡪 add user 🡪 in security credentials 🡪Access key(cli)

## Some commands

Aws configure

Aws –version

Aws sts get-caller-identity

Aws ec2 describe-instances

Etc

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

to check commands

and follow the command line interface pdf in devops learning

# EBS -ELASTIC BLOCK STORAGE

* Block based storage

★ Runs ec2 OS, store data from db, file data, etc

★ Placed in specific AZ. Automatically replicated within the AZ to protect from failure.

★ Snapshot is backup of a volume

## EBS Types

● General Purpose (SSD) ○ Most Work Loads

● Provisioned IOPS ○ Large Databases

● Throughput Optimized HD ○ Big Data & Data Warehouses

● Cold HDD ○ File Servers

● Magnetic ○ Backups & Archives

<https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/AmazonEBS.html>

or ebs volume types

## volumes

instances 🡪 storage 🡪 volume 🡪 click and rename the volume (both instances and volume should be in the same AZ (Availability zone)

exercise:

Size => Ram, CPU, Network etc

Min

4. Storage size

5 gigs for web server images

* Create a volume(make sure instances and volume should be same AZ) 🡪 attach the volume to the exiting once in the (actions 🡪 attach)

Partition:

Partition web images only :

Cd /var/www/html/images/

fdisk -l

fdisk /dev/xvdf

help -n

-a for partition

-1

+3g

Click enter

mkfs (double tab)

mkfs -ext4 /dev/xvdf **# formating**

mount & umount

take a back-up of images : mkdir /tmp/backup-imgs

mv /var/www/html/images/\* /tmp/backup-imgs/

mount /dev/xvdf /var/www/html/images/

**# mount filesystem path**

df -h (to check mounted or not

umount /var/www/html/images/

**# umount path**

**this is tempory mount**

**to do permanently**

* Vi /etc/fstab

Add the line

/dev/xvdf /var/www/html/images/ ext4 defaults 0 0

# **( 0 – no dump, 0 – file system sfc code)**

Mount -a

mv /tmp/backup-imgs/\* /var/www/html/images/

if still website is not working then use

vi /etc/selinux/cofig

* disable selinux

reboot the machine

short note

check

df -h (list to filesystem to mounted)

fdisk -l (list of disks)

mkfs -🡪 format for files(file type)

add the volume🡪 make new partition(fdisk xvdf) 🡪 mkfs (files format 🡪 do mount 🡪 check the mount status

🡪it temporary

* to make permanent 🡪 vi /etc/fstab 🡪 add the line dev/xvdf1 /var/www/html ext4 defaults 00

# **filesystem** path to mount **file-format defaults 0 0**

# Snapshot Backup & Restore

* Unmount partition

• Detach volume

• Create new volume from snapshot

• Attach the volume created from snapshot

• Mount it back

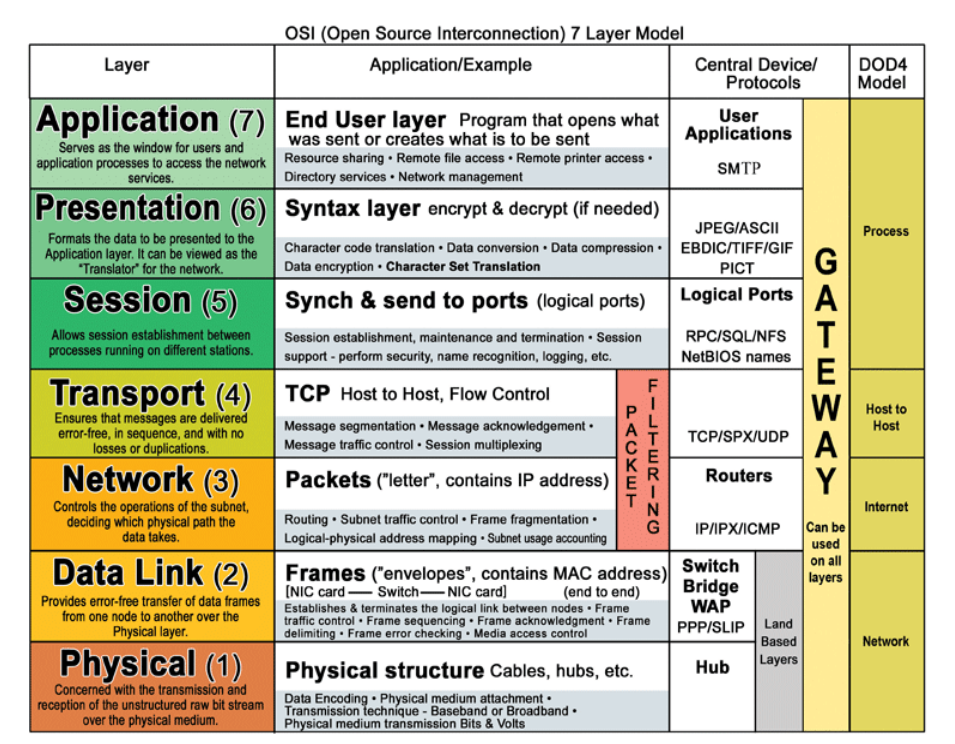
Snapshot :

Create a snapshot from exiting volume(like for backup&restore), if you remove the file or corrupt the file then we can replace the new volume by using snapshot

* Umount the exiting volume🡪 detach the volume🡪 create a new volume from snapshot(action)🡪attach the new volume 🡪 mount it again (mount -a)
* Snapshot can be used for exception , increase the volume , change the type of the volume , image , copy from one region to other region etc

If you need to delate the volume

* Detach the volume 🡪delate the volume



**Application(Layer 7)** This layer supports application and end-user processes. Communication partners are identified, quality of service is identified, user authentication and privacy are considered, and any constraints on data syntax are identified. Everything at this layer is application-specific. This layer provides application services for file transfers, e-mail, and other network software services.  
  
**Presentation(Layer 6)** This layer provides independence from differences in data representation (e.g., encryption) by translating from application to network format, and vice versa. This layer formats and encrypts data to be sent across a network, providing freedom from compatibility problems. It is sometimes called the syntax layer.  
  
**Session(Layer 5)** This layer establishes, manages and terminates connections between applications. The session layer sets up, coordinates, and terminates conversations, exchanges, and dialogues between the applications at each end. It deals with session and connection coordination.  
  
**Transport(Layer 4)**This layer provides transparent transfer of data between end systems, or hosts, and is responsible for end-to-end error recovery and flow control. It ensures complete data transfer.  
  
**Network(Layer 3)** This layer provides switching and routing technologies, creating logical paths, known as virtual circuits, for transmitting data from node to node. Routing and forwarding are functions of this layer, as well as addressing, internetworking, error handling, congestion control and packet sequencing.  
  
**Data Link(Layer 2)** At this layer, data packets are encoded and decoded into bits. It furnishes transmission protocol knowledge and management and handles errors in the physical layer, flow control and frame synchronization. The data link layer is divided into two sublayers: The Media Access Control (MAC) layer and the Logical Link Control (LLC) layer. The MAC sublayer controls how a computer on the network gains access to the data and permission to transmit it. The LLC layer controls frame synchronization, flow control and error checking.  
  
**Physical(Layer 1)** This layer conveys the bit stream - electrical impulse, light or radio signal -- through the network at the electrical and mechanical level. It provides the hardware means of sending and receiving data on a carrier, including defining cables, cards and physical aspects.

**OSI Layer Model for concentrators**

**Hubs/Repeaters** are found in the **Physical Layer**  
  
**Switches /Bridges/Wireless Access Point**are found in the **Data Link Layer**

**Multilayer Switch** are found in both the **Data Link Layer and Network Layer**  
  
**Routers** are found in the**Network Layer**  
  
**Gateway** are found in **All 7 of the OSI Layers**

**Brouter** are found in both the **Data Link** and **Network Layer**

|  |
| --- |
| [**OSI**](http://en.wikipedia.org/wiki/OSI_model)**OSI 7 Layer Model** |
| **7. Application Layer**- DHCP, DNS, FTP, HTTP, IMAP4, NNTP, POP3, SMTP, SNMP, SSH, TELNET and NTP[more)](http://en.wikipedia.org/wiki/Category:Application_layer_protocols) |
| **6.** **Presentation layer** – SSL, WEP, WPA, Kerberos, |
| **5. Session layer** – Logical Ports 21, 22, 23, 80 etc… |
| **4. Transport -**TCP, SPX and UDP[more)](http://en.wikipedia.org/wiki/Category:Transport_layer_protocols) |
| **3. Network -** IPv4, IPV6, IPX, OSPF, ICMP, IGMP and ARP[MP](http://en.wikipedia.org/wiki/Internet_Group_Management_Protocol) |
| **2. Data Link-**802.11 (WLAN), Wi-Fi, WiMAX, ATM, Ethernet, Token Ring, Frame Relay, PPTP, L2TP and ISDN[-](http://en.wikipedia.org/wiki/Wi-Fi)[ore)](http://en.wikipedia.org/wiki/Category:Link_protocols) |
| **1. Physical-**Hubs, Repeaters, Cables, Optical Fiber, SONET/SDN,Coaxial Cable, Twisted Pair Cable and Connectors [(more)](http://en.wikipedia.org/wiki/Category:Physical_layer_protocols) |

# Clusters

● Cluster of servers needs Endpoints

● Endpoints are usually of a Load Balancer

● Load Balancer balances incoming traffic to backend servers

## LOAD BALANCER Ports

* Frontend Port: Listens from the User Requests on this port AKA Listeners.
* e:g 80, 443, 25 etc
* Backend Ports: Services running on OS listening on this port
* e:g 80, 443, 8080 etc

ELASTIC LOAD BALANCER

● Elastic Load Balancing distributes incoming application or network traffic across multiple targets, such as Amazon EC2 instances, containers, and IP addresses, in multiple Availability Zones.

● Elastic Load Balancing supports three types of load balancers:

* Application Load Balancer
* Network Load Balancer
* Classic Load Balancer
* Gateway Load Balance

# CLASSIC LOAD BALANCER

● The Classic Load Balancer that routes traffic based on either application or network level information

● The Classic Load Balancer is ideal for simple load balancing of traffic across multiple EC2 instance

# APPLICATION LOAD BALANCER

● Application Load Balancer that routes traffic based on advanced application level information that includes the content of the request

## NETWORK LOAD BALANCER

● A Network Load Balancer functions at the fourth layer of the Open Systems Interconnection (OSI) model.

● It can handle millions of requests per second.

● Static IP

# Refer awssides in devops learning

# ELB HANDS-ON

Create a instances with amazon – linux with provision tooplate template website 🡪 create a AMI(from the instances)[ instances🡪action🡪image and templates🡪 create image]

**“snapshot we can create a volume”**

**“AMI we can create an instance” AMI equlent shapshot + matadata**

* We can launch the instances from the AMI

OR

We can create a launch template from template, we can create a same type of instances very quickly

ELB: importance’s

Now we have two instances with blog website, they cannot be access separately by user,

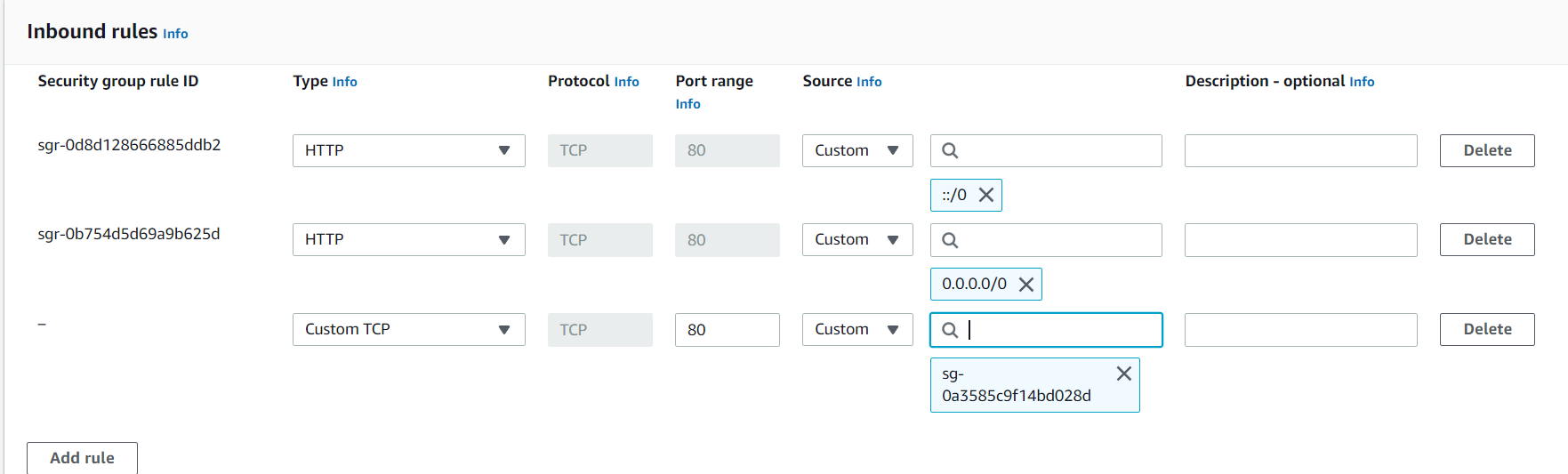
They can be accessing the website from the single endpoint that can be done by the load balances

Load balances: goto target group

[ **target group is basically a group of instances, with health . checks** (it checks the website health by loading again and again upto the value we given, if it all loaded good then it is health]

* Create a load balances (application load balances) with instances and create a security group and add the target group and check the option again launch it)

🡪Once load balance in active copy the DNS name and check the website is working or not. If not, check the security group (if you using different sg’s for the instance the pass the elb-sg in sg (sg- security group)



In target group(tg) we can register or deregister the tg 🡪(deregister is help to stop the traffic on maintance work)

# Cloud watch

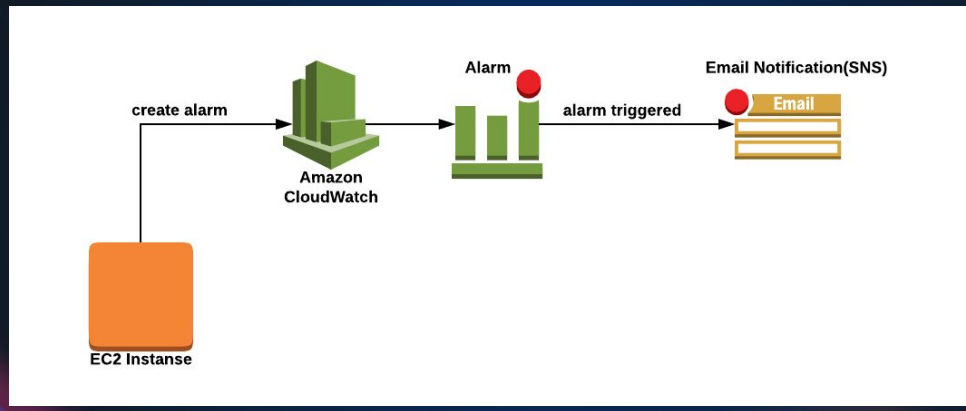
Cloud Watch - Monitor performance of AWS environment - standard infrastructure metrics.

● Metrics: AWS cloud watch allows. you to record metrics for services such as EBS, EC2, ELB, Route53 Health checks, RDS, Amazon S3, cloudfront etc etc…

● Events: AWS Events delivers a near real-time stream of system events that describe changes in Amazon Web Services (AWS) resources

● Logs: You can use Amazon CloudWatch Logs to monitor, store, and access your log files from Amazon Elastic Compute Cloud (Amazon EC2) instances, AWS CloudTrail, Route 53, and other sources

* Alarm monitors CloudWatch metrics for Instances.
* Simple Notification Service (Amazon SNS) is a web service that coordinates and manages the delivery or sending of messages to subscribing endpoints or clients.



In instance🡪monitor(can see the metrics logs)

Normally/default cloud watch will monitor the metrics every five minutes (but can customize the time but some cost will applicable)

Instances🡪monitor🡪managing &detail monitor🡪can change the time here

* Launch the instance🡪 install the stress🡪 check the monitor graft

Yum install stress -y

Stress

Stress -c 4 (to stress the cpu 4 proess)

Top (to check the all utilization)

Script

Script.sh

Sleep 60 & stress -c 4 -t 60 & Sleep 30 & stress -c 4 -t 60 &

Sleep 60 & stress -c 4 -t 500 & Sleep 60 & stress -c 4 -t 30 &

Nohup ./script.sh &

Cloudwatch🡪alarm🡪create alarm🡪select metrics🡪browse🡪ec2🡪 cpu-unitization🡪create topic🡪set alarm (name : warning | high cpu untilztion )

Some other alarm tools : promethus, nagios, icinga, zenos..etc

# EFS – ELASTIC FILE SYSTEM

https://aws.amazon.com/efs

EFS – is a shared storage on aws (shared file system over the network)

* Create a instance 🡪 create a security group (efs-blog-sg) 🡪 create a EFS file system 🡪 create a access point
* To mount the EFS file system using this link :
* <https://docs.aws.amazon.com/efs/latest/ug/wt1-test.html>

to install amzon-efs-ulity is easily in amazon linux

sudo yum install -y amazon-efs-utils

**#please check the doc for more details**

# Using the EFS mount helper to automatically re-mount EFS file systems

file-system-id:/ efs-mount-point efs \_netdev,noresvport,tls,iam,accesspoint=access-point-id 0 0

fs-47a77ccb /var/www/html/img efs \_netdev,noresvport,tls,iam,accesspoint=fsan-03f6334520365d3 0 0

* Take a backup of images

Mkdir /tmp/backup-img

Mv /var/www/html/img/\* /tmp/backup-img

Vi /etc/fstab

fs-47a77ccb /var/www/html/img efs \_netdev,noresvport,tls,iam,accesspoint=fsan-03f6334520365d3 0 0

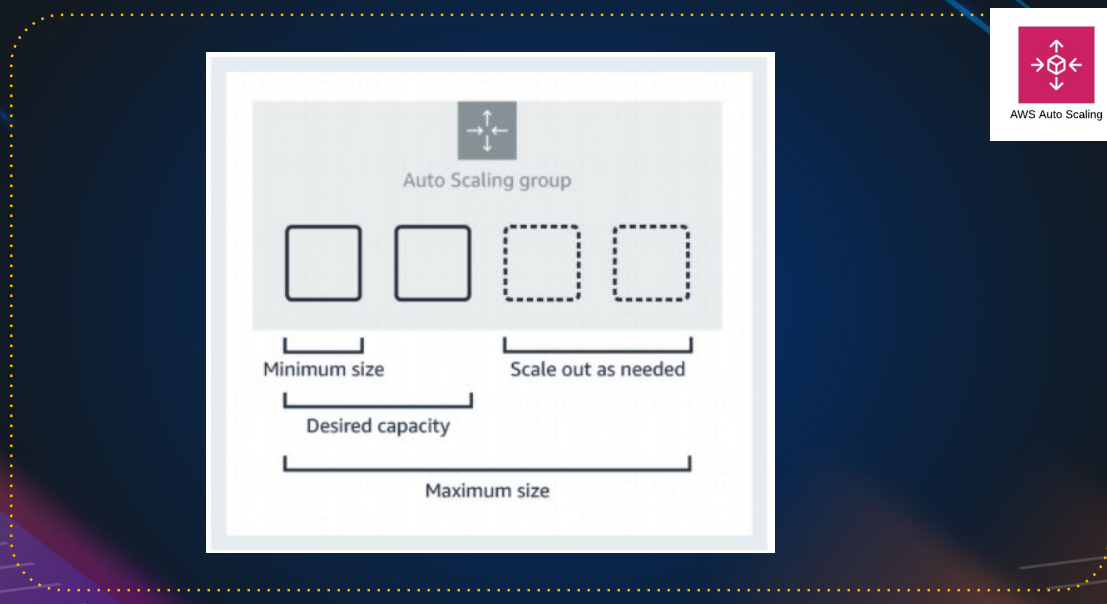
mount -fav

Mv /tmp/backup-img/\* /var/www/html/img/

Df -h

# AUTO SCALING

* Auto Scaling is a service that automatically monitors and adjusts compute resources to maintain performance for applications hosted in the AWS.
* Alarm monitors CloudWatch metrics for Instances
* A launch configuration/Template is an instance configuration template that an Auto Scaling group uses to launch EC2 instances
* Scaling policy is used to increase and decrease the number of running instances in the group dynamically to meet changing conditions.

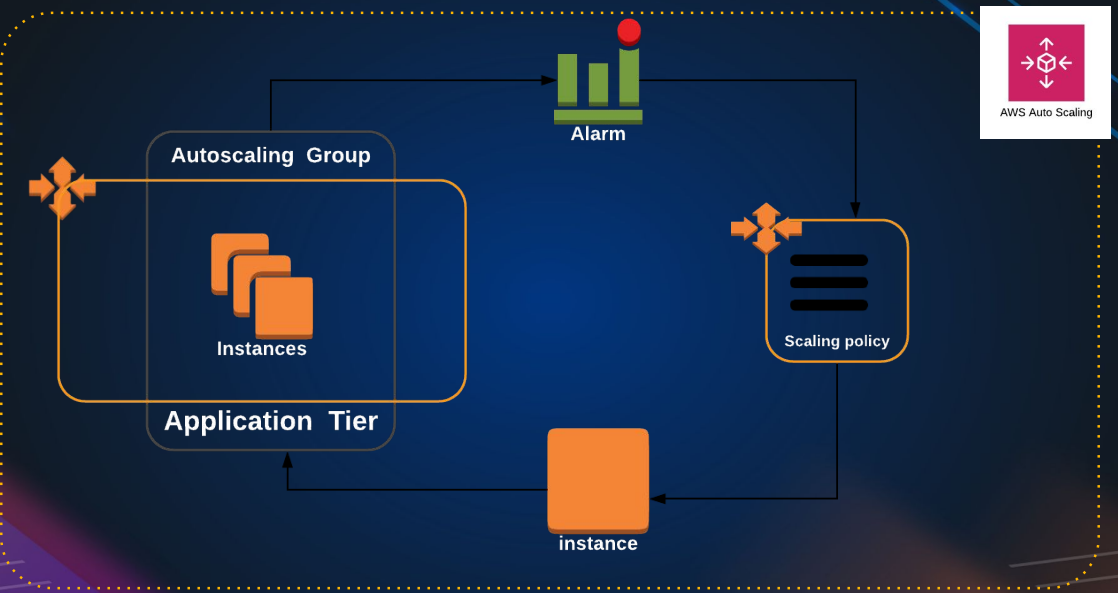


Example:

Minimum size: 1 instance #we can’t remove it

Desired capacity: 2 instances #it will create 2 instances on auto scaling

Maximum size: 4 instances #ASG can create max of 4 instances



Hands-on: create a lunch template🡪 set the target group(if it empty also no-issues ASG will assign automatically)🡪create a ASG🡪

Launch template: we can edit

Launch configurations: we can’t edit it

it

In Health check : ec2 health check by default it will do hardware health check and vm health check

ELB health check “ target group checks every 30 seconds, if instance is unhealth it will declare as unhealth, means basically it checking the port number or process whether it’s up or not.

If instance is health, target group will declare as health. If tg declare is not health then ASG remove the unhealth instance and create a new instance automatically.

Note:

If you update manually in instance, then after sometime instance get deleted and it will launch instance with OLD AMI, so always create a new launch template and edit the launch template in ASG and update they

ASG🡪 select the ASG🡪 details🡪edit the launch template and update the new launch template over there 🡪 start instances refresh in **instances refresh**

Important

ASG is dynamic instances, make sure they do not store any information.

The storage should be out of this easy instance, like on EFS OR NFS, somewhere out

# S3- Simple Storage Service

(it is like a google drive & drop box but much more feature)

Amazon Simple Storage Service (Amazon S3) is storage for the internet. You can use Amazon S3 to store and retrieve any amount of data at any time, from anywhere on the web.

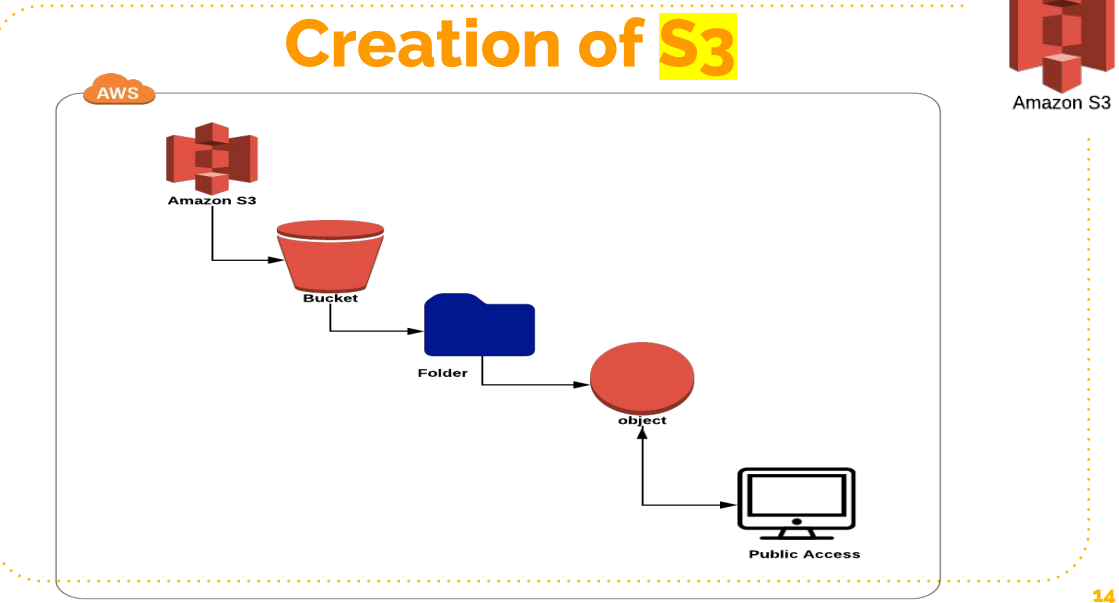
## S3 Basics

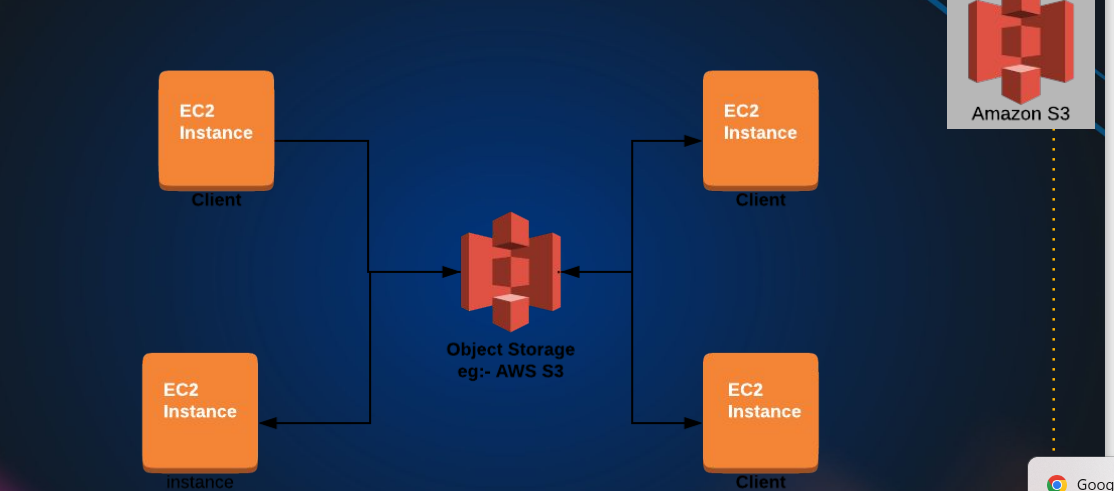
* It is Object-Based Storage
* Data is replicated across multiple facilities
* Unlimited Storage
* Amazon S3 stores data as objects within buckets
* Bucket name has to be unique

#Because you will be getting end point to access it and then end point will have the name of the bucket, so it has to be unique in the internet

Bucket: A bucket is a logical unit of storage in Amazon Web Services (AWS).

Object: Object Storage is a computer data storage architecture that manages data as Objects.





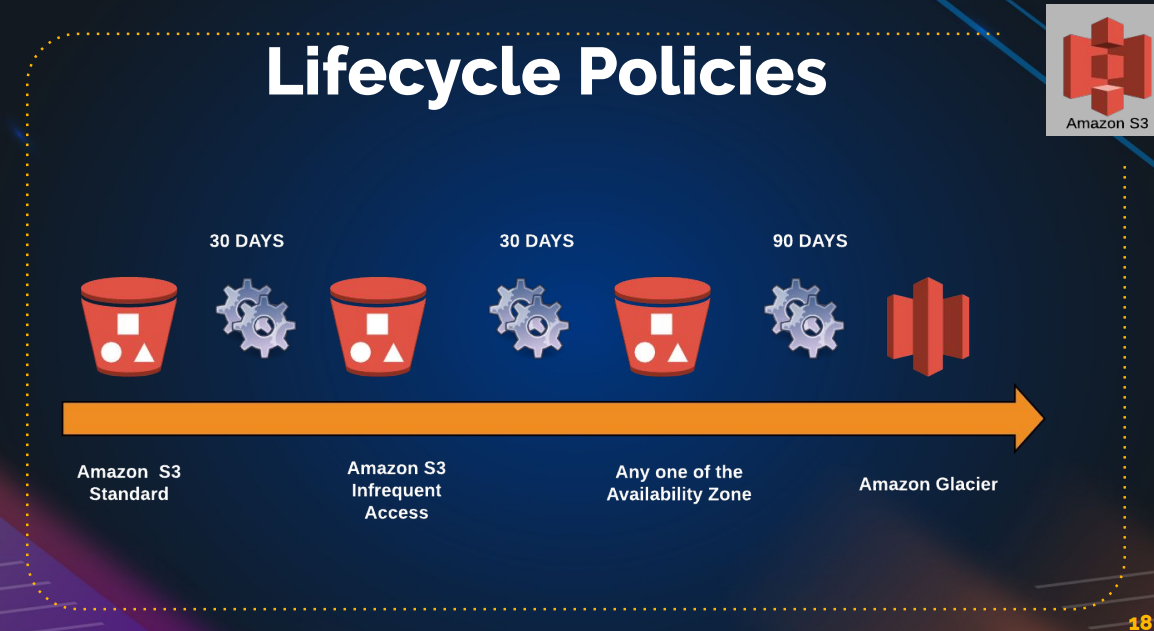
It seems like EFS through this architectural design, but in EFS we mount the filesystem at the operating system level.so we get a folder at the operating system layer, level, where we store the data

But S3 we are going programmatically access through our application.

We can also mount s3 bucket to a folder but that comes though a different driver S3FS

# S3 Storage Classes

1. S3 Standard: general-purpose storage of frequently accessed data. Fast access & object replication in multi-AZ.
2. S3 IA– Infrequent Access: Long-lived, but less frequently accessed data. Slow access, object replication in multi-AZ.
3. S3 One Zone-IA is for data that is accessed less frequently, but requires rapid access when needed. Slow access, no object replication.
4. S3 Intelligent Tiering Automatically moves data to most cost effective tier.
5. S3 Glacier: Low Cost Storage class for data Archiving.
6. S3 Glacier Deep Archive Lowest cost storage, retrieval time of 12 Hrs.



**This to save the cost** ,

we are applying the lifecycle policy, usually on log’s archive and you can also set expiry, also you can set after sometime(1year or two years) delete this data.

# S3 Charges

* Storage
* Requests
* Tiers
* Data Transfer
* Region Replication

Some of the feature’s like **bucket versioning**: if you enable this data is overwrite or deleted , then is present there only then we can revive deleted once. But if we delete more then more storage is required

S3 object 🡪 delete the file(index.html)🡪 it will not delete(because versioning is enble) 🡪if you delete permanent the click the enable-icon on the left and select deleted files do permanent delete to remove the files

Use case: upload the date and access to from anywhere by making public. We can host to public

Create a S3 bucket 🡪 upload the data(code, files and artifacts etc) 🡪unblock the public serves 🡪 goto object🡪 make it public (to files/object) 🡪 use the object URL to access the object/name from anywhere

Exercise:

Launch the website from tooplate.com using s3 bucket

Move/upload all the files to S3 bucket 🡪 select all files and make as public 🡪 check the block permission if in block state 🡪 properties under bottom enable the static website hosting 🡪 get the end point URL and check the website is working or not.

# Lifecycle roles

[Amazon S3](https://s3.console.aws.amazon.com/s3/get-started?region=us-east-1) 🡪[Buckets](https://s3.console.aws.amazon.com/s3/buckets?region=us-east-1) 🡪[blog-storage-s3](https://s3.console.aws.amazon.com/s3/buckets/blog-storage-s3) 🡪[Lifecycle configuration](https://s3.console.aws.amazon.com/s3/management/blog-storage-s3/lifecycle?region=us-east-1) 🡪 **Create lifecycle rule**

**Set the roles for storage type to cost “refer the lifecycle policies**

## Replication

[Amazon S3](https://s3.console.aws.amazon.com/s3/get-started?region=us-east-1) 🡪 [Buckets](https://s3.console.aws.amazon.com/s3/buckets?region=us-east-1) 🡪[blog-storage-s3](https://s3.console.aws.amazon.com/s3/buckets/blog-storage-s3) 🡪[Replication rules](https://s3.console.aws.amazon.com/s3/management/blog-storage-s3/replication?region=us-east-1) 🡪 C**reate replication rule**

**“” UPTO 5GB S3 under free tair “”**

# Relational Database (RDB)

## DB Administration

* Installs
* Patching
* Monitoring
* Performance Tuning
* Backups
* Scaling
* Security
* Hardware upgrades
* Storage Management

## RDS

❖ Amazon Relational Database Service is a distributed relational database service.

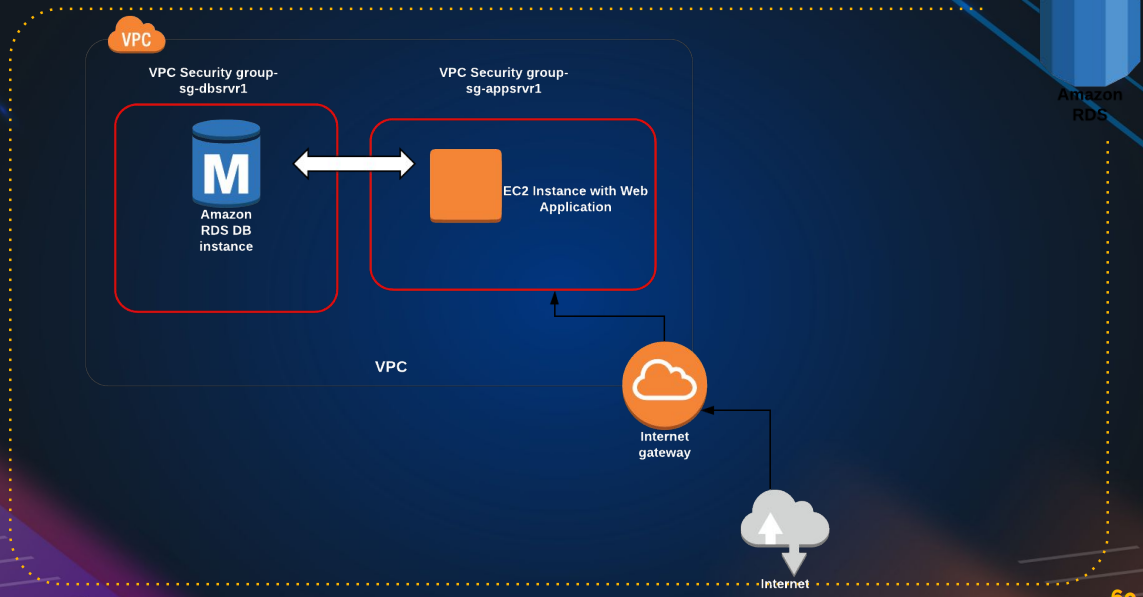
❖ High Availability Multi-AZ Deployments.

❖ Effortless Scaling.

❖ Read Replicas for performance

“RDS support various database engines like MySQL, PostgreSQL, Oracle etc. if you using MySQL or PostgreSQL then amazon suggest to go with “amazon Aurora” because

* amazon Aurora is 5 time faster than MySQL.
* amazon Aurora is 3 time faster than PostgreSQL.
* Cheater than compare to MySQL, PostgreSQL.
* It is serverless and support 64 TB.



RDS 🡪create a db (eg: mysql)🡪create instance EC2🡪login to instances 🡪make sure SG have proper roles🡪 connect the db

Sudo yum install mariadb-server -y

If it not found the search for available server

sudo yum/dnf search mariadb

and install it

to connect the server: mysql -h YourDatabaseEndpoint -u YourUsername -p current password

“””need to check for password reset”””

RDS cost’s more accounting to usage. Use properly

# Aws lift & shift project

🡪Refer the pdf in the devops learning

Vproject vagrant project shift to aws server

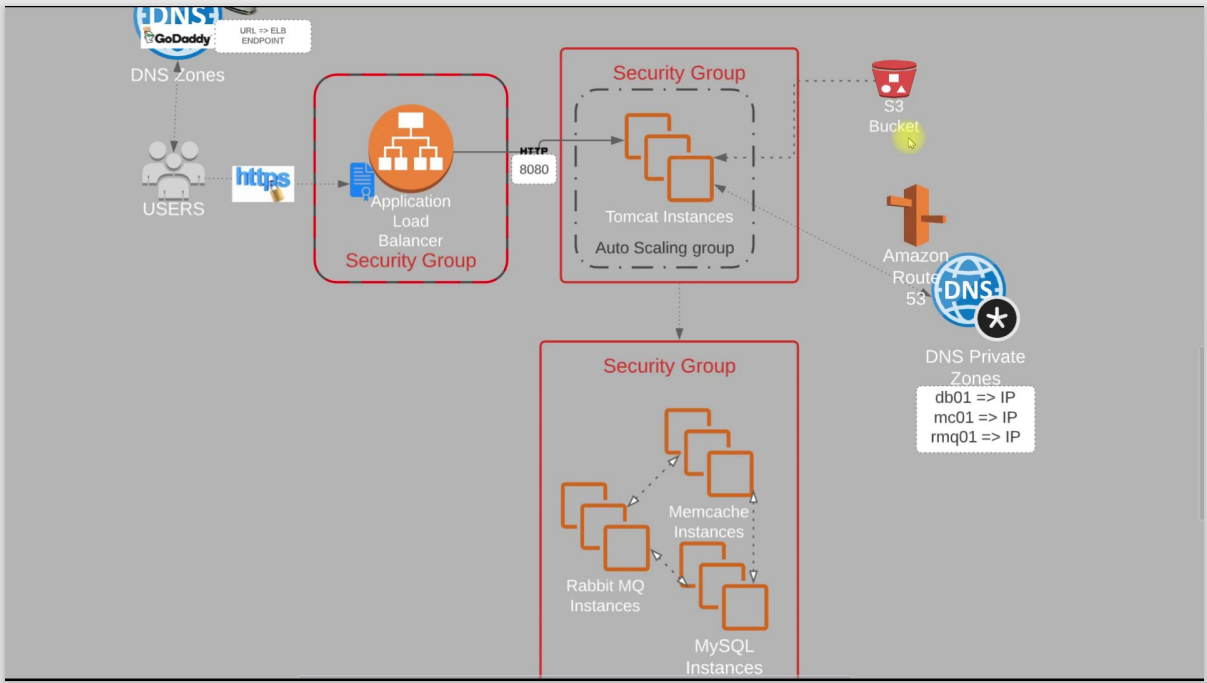
Vagrant:

User 🡪Nginx(load balancer🡪single end point) 🡪 Apache tomcat server(httpd servers)🡪 rabbit mq(message service)🡪Memcached(temporary che service) 🡪 database (mysql)

Vs

Aws

User 🡪 load balancer 🡪 instance (tomcat, RabbitMQ, Memcached,mysql instances)🡪key pairs 🡪 security group🡪s3/EFS🡪Rout 53



Key pair : bproject-KP.pem

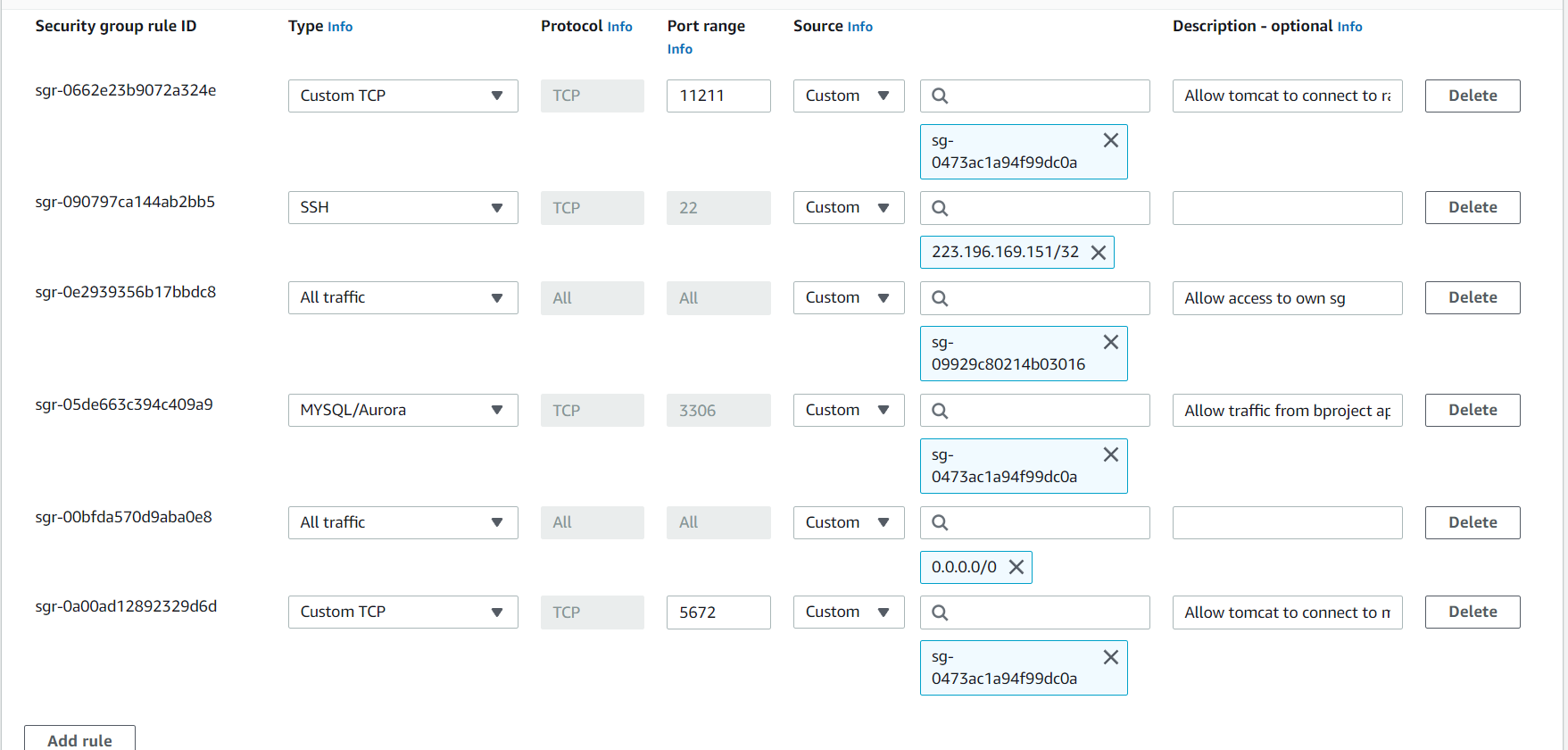
Security group :

ELB: inbound roles http & https to 8080 port for 0.0.0.0/0 & ::/0

APP: inbound roles 8080 port / allow the elb-sg to app & port 22

BACKEND: here we are using three services : mysql, Memcached and rabbitmq

Allow every port app-sg 3306 for sql , 5672 for rabbitmq and memecached 11211 and allow same sg for internal data transfer.



Then create ec2 for all 4 services mysql,rabbitmq,memecacahed & tomcat7(ubuntu)

By provision bash script in git repo aws lift and shift

“”default tomcat application can be they in this provision is /var/lib/tomcat9” we find webfiles here and inside ROOT(default tomcat page) will be there

<https://github.com/hkhcoder/vprofile-project/tree/main>

* Check the service active are not
* We can check the retrieve data for ec2 by curl <http://169.254.169.254/latest/user-data>
* We can check status by systemctl or ss -tunlp | grep “given the service port number”

Normally we use /etc/hosts to ip addr in vagrant but in ROUTE 53 service in aws

https://github.com/devopshydclub

# ROUTE 53

Domain name service (DNS)