**EC2**

EC2: - considered as VM let it make as a server.

Considered it as VM and a Server.

For storing – (Volumes) **EBS**

For operating - **ELB** (elastic load balancing)

If more users are using – **ASG** (Auto Scaling Group)

Each instance has 2 IPs: **Public & Private IPs**

* Public IP used to access in web-browser
* Private IP used to interact using UI / in PowerShell / cloud shell

**EC2 Instance types**

General purpose- load balancing (compute, network, storage). ---------t

Compute optimized- for high performance, Ex: Higher ROM --------------c

Memory optimized- memory capacity, High storage, Ex: High Ram -------r

Storage optimized- for faster read -write, I/O Operations Per Second. ----i

**EC2 Instances Purchasing options**

On demand

Reserved

Savings

Spot instances

Dedicated hosts

Dedicated instances

Capacity reservations

**Security Group** – Used to set firewall for inbound & outbound purpose to secure.

**IMP for exam**

Shared responsibility

EC2 Instance types

EC2 Instances Purchasing options

**EBS (Elastic Block Storage)**

Attach

If we connect EBS to EC2, even after terminating EBS it allows EC2 instance to persist data.

Data is stored in terms of blocks virtually.

They can be mounted to 1 instance at a time.

1 EBS can be detached from instance and attach to another instance.

1 EC2 can connect to a EBS only at once.

1 EC2 instance can connect to multiple EBS

1 EBS can connect to 1 EC2 only.

EBS is region & AZs specific.

EBS storage can be scalable as needed.

With each EC2 a Default volume (root volume), if ec2 gets deleted then its root volume also deleted not the actual EBS volume which we attached.

EBS archiving: To store Snapshots to archive we get up to 75% discount from AWS to store.

From archive to restore back it takes 24-72 hours.

We must set up a rule to retain accidental deletion of EBS vol.

Specify retain from recycle bin from 1day to 1 year

**Shared responsibility Model for Cloud Computing:**

|  |  |
| --- | --- |
| **AWS (of the cloud)** | **User / customer (in the cloud)** |
| S/W , H/W infra | Customer data, platform, apps |
| Compute, storage, network, database. | OS, N/W & firewall configuration. |
| Regions, AZs, Edge locs | Client-side data, server-side encrypt, networking traffic. |
|  | Analyze access permissions |

**Shared responsibility Model for IAM:**

|  |  |
| --- | --- |
| **AWS** | **User / customer** |
| Infra | Users, groups, roles, policies, monitor |
| Configuration | MFA |
| Compliance validation | IAM tools |
|  | Analyze access permissions |

**Shared responsibility Model for EC2 storage:**

|  |  |
| --- | --- |
| **AWS** | **User / customer** |
| Infra | Setting up back up / restore procedures |
| **Ensuring data is secured** | **Data encryption** |

**AMI (Amazon Machine Image)**

**How can we access AMIs :**

Public AMIs – provided by AWS cloud providers. (see search button Market place)

Your own AMI (created & managed by us)

AWS marketplace

Region AMI- copy of AMI from existing region to another region.

While creating AMI it also creates EBS snapshots.

**EC2 Image builder**

Automatically creating VMs / container images, in such a way when a trigger occur another instance should pop up.

Using EC2 Image builder we can maintain, validate, automate, test EC2 AMIs

Based on Trigger (packages, date),

Free service – we do not pay for this service, just pay for the resource.

Resources can spin up automatically by scheduling time (weekly / time).

EC2 image builder is used to **create VMs** based on triggers, then to **create AMIs**, then to copy to another instance to **test that AMIs**.

**EC2 instance store**

EBS volume are network drives with limited performance, if we need high performance hard-disk we will use **EC2 instance store**

**EC2 instance store PROs:**

Beter input output performance

Good for buffer cache & scratch data

Back up and replication are user responsibility

**EC2 instance store CONs:**

EC2 store loose there storage if they are stopped

If hardware fails there is risk of data loss

EC2 Instance store i series

**When creating an instance what default options are selected note that**

**EFS(Elastic File System)**

Mount

EFS(Elastic File System) Used to share data between 2 instances

We can create multiple instances

We can use 1 EFS in multiple AZs

Highly available reliable costlier (follows pay as you go model).

**EFSIA (EFS Infrequent Access)**

It is where the infrequent data is stored

It costs up to 92% lower price than EFS.

|  |  |
| --- | --- |
| **AWS** | **User / customer** |
| Infrastructure | Setting up back up / restore / snapshot procedures |
| Replication of EBS volume, Trails of EFS | Data encryption |
| Ensuring their employees do not access the data |  |

**Amazon FSx ()**

Same for file storage but from third party

1. FSx for lustre

“Lustre = Linux + cluster”

Used in high performance systems. Ex: weather forecast

Big data analytics

Scalable & speed access

Interaction in millions iops .

1. FSx for windows file server

* Supports SMB (Server Message Block) protocol & Windows NDFS protocol
* We can access from on-premises / from AWS
* It is integrated with Microsoft active Directory
* Fully managed scalable windows native shared file system.
* Built on windows file server.

1. FSx for NeTApp

**Elastic load balancing & auto scaling groups**

**Load Balancing**

Kinds of Load Balancing:

1. Application LB

Controls HTTP & HTTPS traffic

Also handle domain names, static DNS

1. Network LB

Controls TCP & UDP incomings traffic

High performance & millions requests per second.

Recognizes static IPs

1. Gateway LB

Data share through IP packets

Routes the traffic to firewall to manage instance

1. Classic LB

**Auto Scaling Groups (ASG)**

It is a logical collection of several amazon ec2 instances used for management & scaling purposes.

Types of ASG:

1. Manual Scaling: we’ll set manually to ASG
2. Dynamic Scaling: response to On-demand

Step scaling: Based on trigger

Target tracking scaling: based on avg CPU usage, we’ll set ROI

Scheduled scaling: known usage pattern, we set a time at which I need to scale.

1. Predictive Scaling: using ML to predict to scaling groups resources in future

**''''''Activity ''''''**

**Connect two EC2 instances and verify in AWS itself**

**Bring 5 differences b/w EBS & EFS**

**EFS set up, Static IP and document on that**

**By doing Start and stop an EC2**

**An instance can attach to how many EBS volume? Get number**

''''''Activity done''''''

In EBS recycle bin concept

In EBS archive concept

Connect 1 EC2 to a EBS

how to get snapshot of EBS

# **To access destination from source we need to copy and paste source public key to destinations authorized\_keys**

**Amazon S3 bucket**

In **S3,** whatever we store is stored in terms of **objects.**

Buckets & also data are region specific

Usually, S3 is infinitely storage.

Use cases of S3 bucket:

Storage

Disaster Recovery

Backup & Recovery

Data archive

Hybrid cloud storage (combination of on-premises & cloud)

Media hosting

Data leaks & big data analytics

**Globally the S3 bucket name must be unique After deleting it can be used**

**S3 bucket name criteria**

* No uppercase
* No lowercase
* 3-63 characters limit
* Not an IP
* Starts with a number / lower case letter
* Must not start with prefix (xn--)
* Must not end with -S3alias

Each object in S3 bucket is identified using unique object id:

**Naming convention:**

**Key -> S3://<bucket\_name>/<folder1><folder2>/<file\_name>**

Object id consists of = prefix + file

Key = S3://<bucket\_name>/<folder2>

S3://<bucket\_name>/<file\_name>

**Max Object Size 5TB = 500Gb**

**Max size for a file is 5Gb**

While uploading a file 5Gb is max obj size

If the file uploading is above 5Tb we must divide the file

**ACL (Access Control List)**

Communication b/w User & S3-Bucket is through **API calls.**

**We need to set policies for user using same json format**

**Object Access Control List (OACL)**

It controls Access to individual objects stored in S3.

**Bucket Access Control List (BACL)**

Entire bucket can access objects in S3.