

Region & Availability Zones :AWS Resources are hosted in multiple locations world-wide, these locations are composed of regions and Availability Zones. Each region is a separate geographic area. Each region has multiple, isolated locations known as Availability Zones.

Amazon Elastic Compute Cloud (EC2):

=====

1) This Service is related to Server Configuration like

- a) " Type of Instance",
- b) " What type of OS we need to install",
- c) " Where we need to Launch the server"
- d) " Creating new volumes"
- e) " adding/removing volumes from server"
- f) " Taking back of the server (Snapshot)"
- g) " Creating Load Balancers"
- h) " Creating Auto Scalling" etc.

2) We can launch EC2 instances any region.

To Launch an Ec2 instance below resources are required:

- =====
- a) AMI
 - b) Security Group
 - c) Keypair
 - d) Type of Instance

Amazon Machine Images (AMIs) :

- =====
- a) An AMI is a template that contains a software configuration (operating system,application server and applications) that can run on Amazon's Ec2.AMIs can be used to launch an instance.
 - b) AMI is specific to the region.
 - c) Each AMI have its own unique ID.
 - d) We can Create our own AMIs.

Security Group:

- =====
- a) SG is a virtual firewall. It can be attached to the Ec2 instance.
 - b) SG configuration is having two parts
 - i) Inbound rules
 - ii)outbound Rules
 - b) What type of requests server allows, will be configured in Inbound Rules.
 - c) What type of requests server can send, will be configured in Outbound Rules.
 - d) In SG we can Configure only allow rules.
 - e) One security Group can be attached to Multiple Ec2 instances.
 - f) One Ec2 instance can have upto max of "5" Security Groups.

Keypair:

- =====
- a) To log in to your instance, you must create a key pair, specify the name of the

key pair when you launch the instance, and provide the private key when you connect to the instance.

b) On a Linux instance, the public key content is placed in an entry within "`~/.ssh/authorized_keys`". This is done at boot time and enables you to securely access your instance using the private key instead of a password.

c) Keypairs are region specific.

Type of Instance :

=====

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instance types comprise varying combinations of CPU, memory, storage, and networking capacity and give you the flexibility to choose the appropriate mix of resources for your applications.

Ex :

t2.nano,t2.micro,t2.small,t2.medium,t2.large,t2.xlarge,t2.2xlarge

Creating Security Group:

=====

- a) Goto EC2 Dashboard
- b) On the navigation bar select region for the Security Group
- c) Click SecurityGroup (NETWORK & SECURITY Sec) in the navigation pane
- d) click on Create Security Group.
- e) Specify a name and description for the security group.
- f) choose a VPC ID to create a security group for that VPC.
- g)start adding rules.

Creating Keypair:

=====

- a) Goto EC2 Dashboard
- b) On the navigation bar select region for the key pair
- c) Click Key Pairs in the navigation pane
- d) Click Create Key Pair
- e) Enter a name for the key pair in the Key Pair Name field of the dialog box and click Create
- f) The private key file, with .pem extension, will automatically be downloaded by the browser.

Creating an Ec2 Instance:

=====

- a) Goto EC2 Dashboard
- b) From the navigation bar select the region for the instance
- c) From the Amazon EC2 console dashboard, click Launch Instance
- d) Select the AMI (Ex: Ubuntu16)
- 5) Select the instance type (ex : t2.micro)
- 6) Click Next: Configure Instance Details
- 7) Click Next: Add Storage
- 8) Click Next: Add Tags
- 9) Click Next: Configure Security Group.

- 10) Click Review and Launch. Scroll down to review the AMI details, instance type, and security group information, and then click Launch.
- 11) At the prompt for a key pair: Select Choose an existing key pair/NewKeyPair
- 12) Click the check box to acknowledge that you have access to the private key.
- 13) Click Launch Instances.

Note :

- a) You can also attach existing SecurityGroup to ec2 instance.
- b) you can connect to linux ec2 instance by ssh protocol.

```
ssh -i "privatekeyname.pem" username@public_ip
```

Userdata (Bootstrap Scripts):

=====

- 1) Whatever the command we write in Userdata field, these commands get executed at the time server booting.
- 2) Userdata can be changed on existing servers only if it is in "Stopped" state.
- 3) We will use this option to start some services, to clear the existing log files.
- 4) First line of user data must be "Shebang" (#!/bin/bash), After that we can give in any valid os commands.

Note : You will find "User data" field, In instance creation Process, "Configure Instance Details" step is having section called "AdvancedDetailed".

Change/add userdata commands on existing ec2 instances: Instance needs to be Stopped

=====

before changing userdata on existing ec2 instance.

- 1) Goto EC2 Dashboard
- 2) Select The ec2 instance
- 3) Action --> Instance Settings --> "View/Change user data"

Terminate Ec2 instance:

=====

- 1) Goto EC2 Dashboard
- 2) Select The ec2 instance
- 3) Action --> Instance State --> "Terminate"

Stop Ec2 instance: If instance is in running state we can stop the instance

=====

- 1) Goto EC2 Dashboard
- 2) Select The ec2 instance
- 3) Action --> Instance State --> "Stop"

Note: When we stop the instance public ip will be removed from the instance. when we start the instance new public ip will be assigned to ec2 instance.

Reboot Ec2 instance: If instance is in running state we can "Reboot" the instance

=====

- 1) Goto EC2 DashBoard
- 2) Select The ec2 instance
- 3) Action --> Instance State --> "Reboot"

Note: When we reboot the instance, public ip will not be changed.

Practice:

=====

- 1) Launch an ec2 instance in Mumbai region
- 2) Launch an ec2 instance in Mumbai region "ap-south-1a" AZ.
- 3) Create a security group and configure inbound "ssh" & "http" rules
- 4) Attach "step3" security group to "step2" ec2 instance.
- 5) Stop the "step2" ec2 instance and check the public ip address.
- 6) start the Ec2 instance after step5 execution, check the public ip address.
- 7) Create new keypair
- 8) Launch an ec2 instance using step3 "SecurityGroup" & step7 keypair
- 9) Install apache2 in Ubuntu Ec2 instance at the time of Launch.

Elastic IP(EIP):

=====

It is a dedicated public IP address which is reserved for your account, It can be assigned any ec2 instance in you account. After attaching it to ec2 instance, Even if we stop the ec2 instance this public will not be released from instance , It remain attached with EC2 instance.

Create EIP (Reserv EIP):

- 1) Goto EC2 dashboard
- 2) Goto NETWORK & SECURITY sections
- 3) Click On "Elastic IPs"
- 4) Click on "Allocate New Address"

Assigning Elastic IP to Ec2 Instances

- 1) Goto EC2 dashboard
- 2) Goto NETWORK & SECURITY sections
- 3) Click On "Elastic IPs"
- 4) Click On " ACTIONS" --> "Associate Address"
- 5) Select the Ec2 instance

Realese Elastic IP From Ec2 AWS account:

=====

step1: Realese the ELASTIC ip from Instances

- a) Goto EC2 dashboard
- b) Goto NETWORK & SECURITY sections
- c) Click On "Elastic IPs"
- d) Click On " ACTIONS" --> "Disassociate Adress"

step 2: Realese the ELASTIC ip from Account

- a) Goto EC2 dashboard

- b) Goto NETWORK & SECURITY sections
- c) Click On "Elastic IPs"
- d) Select Elastic Ip
- e) Click On " ACTIONS" --> "Release Address"

Note : only five EIP can be reserved for account.

Practice:

=====

- 1) Create EIP and attach it to any existing EC2 instance.
- 2) Stop the EC2 instance and check public IP is released or not from the Instance.
- 3) Release the EIP from EC2 instance
- 4) Release the EIP from your account.

Elastic Block Storage (EBS):

=====

In AWS there are 3 types of storages

- a) EBS Volume (Elastic Block Storage)
- b) EFS (Elastic File System)
- c) S3 (Simple Storage Service)

- 1) EBS is primary storage unit for EC2 instance, Only in EBS we can install OS and softwares.
- 2) Its block based storage, Other two storages (EFS,S3) are object based storages.
- 3) One EBS volume we can attach to one EC2 instance at a time
- 4) We can attach EBS volume to EC2 instance, only if Instance & volume in same AZ .
- 5) EBS,S3 service is available in all regions, EFS is not available in all regions.
- 6) EBS Volumes (Disk) performance is measured by using IOPS(Input Output per second).
- 7) There are Five types of EBS volumes
 - a) GP2
 - b) IO1
 - c) SC1
 - d) ST1
 - e) Magnetic

Note : Above Five EBS volumes supports various rate of IOPS, And high IOPS have, high pricing.

- 8) Out of five EBS volume only below types can be root volumes
 - a) GP2
 - b) IO1
 - c) Magnetic

9) Root Volume: Root volume means where we install the OS. At the time of EC2 instance launch this volume is created.

Attaching Additional EBS volume to Ubuntu EC2 Instance.

=====

- 1) To attach EBS volume to EC2 instance ,Instance and volume must be in same AZ.
- 2) Create a volume in same AZ as EC2 instance
 - a) Goto EC2 dashboard
 - b) Goto "Elastic Block Storage"
 - c) Click on Volume

- d) Click on Create Volume.
- e) Select the AZ.
- 3) Attach volume to EC2 instance
 - a) Goto EC2 dash board
 - b) Goto "Elastic Block Storage"
 - c) select the Volume
 - d) Click on "Action" --> attach Volume
 - e) Select the ec2 instance.
- 4) Check the volume is attached or not from cmd prompt.
 - a) Login into EC2 instance
 - b) Run below cmd
"lsblk"
- 5) Create a file system
mkfs -t "ext4" <volume_name>
ex : mkfs -t ext4 /dev/xvdf
Note : Linux uses "ext4" file system.
- 6) Create mount dir.
mkdir <dir_name>
mkdir test
- 7) Map mount dir to volume
mount <volume_name> <mount_dir>
ex : mount /dev/xvdf test

Note : we able to know the file system is created or not
by running "file -s <volume_name>" cmd.
ex: file -s /dev/xvdf

Detach EBS Volume from Ubuntu EC2 Instance:

- =====
- 1) unmount the volume from cmd prompt
umount <volume_name>
ex : umount /dev/xvdf test
 - 2) Detach it from ec2 instance
 - a) Goto EC2 dash board
 - b) Goto "Elastic Block Storage"
 - c) select the Volume
 - d) Click on "Action" --> detach Volume

Note :
a) If volume state is "inuse" means it is attached to EC2 instance.
b) If volume state is "available" means it is not attached to EC2 instance.

Practice:

- =====
- 1)
 - a) Create an ubuntu EC2 Instance in mumbai region "ap-south-1a" AZ.
 - b) Create "gp2" volume in mumbai region "ap-south-1a" AZ.
 - c) Attach it to the step1 ec2 instance.
 - d) Mount it to Ubuntu instance and create some files in newly added volume.

e) Unmount & detach it from the instance

- 2)
 - a) Create new ubuntu ec2 instance in mumbai region "ap-south-1a" AZ.
 - b) attach & mount "Practice 1" volume to it.
 - c) check the files which you have created in "Practice 1 (step 4)" .

Server Upgradation (ex : T2 Micro To T2 Large)

=====

Assumptions:

=====

T2. Micro instance is running & instance is having
os users, other software are installed in it (ex Apache2,Java).

Steps:

- 1) Launch new "T2 Large" Instance
- 2) Stop existing "T2 Micro" Instance
- 3) Detach "root" volume from "T2 micro" Instance
- 4) Stop "T2 Large" Instance (Step1)
- 5) Detach "root" volume from "T2 Large" Instance
- 6) Attach t2micro instance root volume (step3) to
t2 Large instance (Root volume device type value must be "/dev/sda1")

2nd Method:

- =====
- 1) Goto Ec2 dashboard
 - 2) Select the EC2 instance
 - 3) Goto "Actions" --> "Instance settings" --> "Change Instance Type."

Snapshot:

- =====
- 1) Snapshot is a point in time backup of EBS volume.
 - 2) Snapshot is at region level.
 - 3) Based on snapshot we can create new EBS volumes
 - 4) Snapshot is stored in S3 bucket.
 - 5) snapshot size must be >= volume size.
 - 6) We can copy snapshot to other regions also
 - 7) we cannot attach snapshot to EC2 instance, To access
snapshot data ,we need to create volume.

Creating Snapshot:

- =====
- a) Goto EC2 dash board
 - b) Goto "Elastic Block Storage"
 - c) Click on Volume
 - d) select the Volume which you want to take the backup.
 - e) Click on "Actions" --> "Create Snapshot"
 - f) Created snapshot will be in Snapshot section.
 - i) Goto EC2 dash board
 - ii) Goto "Elastic Block Storage"
 - iii) Click on Snapshots

Retrieve Data from snapshot/ Create volume from snapshot:

=====

we cannot attach snapshot to EC2 instance, To access snapshot data ,we need to create volume and need to attach it to EC2 instance.

Steps:

- i) Goto EC2 dash board
- ii) Goto "Elastic Block Storage"
- iii) Click on Snapshots
- iv) Select the snapshot
- v) Click on "Action" --> Create Volume
- vi) Created volume will be in Volume section
- vii) Attach the created volume to any ec2 instance.

Practice:

- =====
- 1)
 - a) Take the backup of any existing EBS volume
 - b) Create a new volume from snapshot
 - c) attach & mount it to any ec2 instance.
 - d) Check the data is exists or not.

Q1) How to attach one AZ volume to Other AZ Ec2 instance.

Q2) How to attach One regions AZ volume to Other Region AZ instance.

2) Delete the snapshot which you have created in step 1.

Creating Custom/Own AMIs:

=====

AMI: AMI is used to boot the servers, We can create our own AMIs.

Creating custom AMIs

- =====
- 1) Launch any ec2 instance
 - 2) Install required softwares (ex apache2,java,etc) and security patches.
 - a) Login into ubuntu EC2 instance
 - b) Install apache2

```
sudo apt-get update
```

```
sudo apt-get install apache2 -y
```
 - 3) Based on instance we need to create Image from GUI
 - a) Goto ec2 dash board
 - b) select the instance
 - c) click "Actions" --> "Image" --> "Create Image"
 - 4) When we create AMI ,It creates Snapshot.
 - 5) We can see All Custom AMI in "My AMI" section.
 - 6) We can launch new instances based on this AMI (New instace will have apache2 software).

How to delete Custom AMI ?

- =====
- 1) First we need to deregister the AMI
 - a) Goto ec2 dash board
 - b) goto "IMAGES" section
 - c) Click on AMI
 - d) select the AMI.
 - e) click on "Actions --> Deregister AMI"
- 2) Next we need to delete the snapshot
 - a) Goto ec2 dash board
 - b) goto "ELASTIC BLOCK STORE" section
 - c) click on "Snapshot"
 - d) Select the snapshot
 - e) click on "Action" --> Delete

=====*****=====

=====

Elastic Load Balancer(ELB) :

=====

Elastic Load Balancer automatically distributes incoming application traffic across multiple targets, such as EC2 instances.

It can handle the varying load of your application traffic in a single Availability Zone or across multiple Availability Zones.

There are 3 types ELBs in AWS:

=====

- 1) Classic ELB
- 2) Application ELB
- 3) Network ELB

ELB Listener:

Front End Listener:

=====

An ELB Listener, is the process that checks for connection request Listeners check for traffic from "clients to the ELB"

Backend Listener:

=====

Listeners are configured with ports to check for traffic from ELB to EC2

ELB Features:

=====

- a) It may take sometime for the registration of ec2 instance under ELB to complete
- b) Elb also monitor the health of its registered instances and ensures that it routes traffic only to healthy instances
 - > A health instance shows as "In-service " Under ELB
- c) When ELB detects instance is unhealthy it stop routing traffic to the instance
 - > A unhealth instance shows as "outof-service " Under ELB

- d) When ELB service detects ec2 instance is back to healthy it resumes traffic routing to that instance
- e) By default elb uses ping HTTP (port 80) for health checks
- f) registered instance must respond within time period, otherwise it will be considers as unhealthy
- g) ELB Health Checks to EC2 Instance:
 - i) Health check interval : Period of time between health checks
 - ii)Unhealthy Threshold : No of consecutive failed health checks that should occur before the instance is declared unhealthy.
 - iii)Healthy Threshold : No of consecutive success health checks that must occur before the instance is declared healthy

Other Imp Parameters of ELB:

=====

- a) Cross-Zone Load Balancing: If this feature is disabled ELB sends traffic evenly between the AZs , without consideration to the number of registered instances in each AZ.
 - b)Sticky Sessions:With the stickiness feature, you can configure the load balancer to bind usersessions to specific application instances. All requests coming from the user during the session will be sent to the same application instance.
- ELB supports two Mechanism to provide session stickiness:
- i) load balancer-generated HTTP cookies: which allow browser-based session lifetimes,
 - ii) application-generated HTTP cookies: which allow application-specific session lifetimes.
- c) Connection Draining : Connection Draining allows existing requests to complete before the load balancer shifts traffic away from a unhealthy back-end instance.
 - d) ELB Idle Timeout : it is the number of seconds a connection can be idle before the load balancer closes the connection.

Practice:

=====

- 1) Create two ec2 instances in different AZs (allow ssh & Http requests in SG)
- 2) Install Apache2 webserver on above two instances
- 3) Apache run on port number 80.
- 4) Successful apache2 installation will open default apache2 webpage
(copy paste the "public ip " in the browser.)
- 5) Default apache2 page location
/var/www/html/index.html
- 6) Create a classic loadbalncer:
 - a) Goto Ec2 Dash Board
 - b)on the navigation pane, under LOAD BALANCING, choose Load Balancers.
 - c)Choose Create Load Balancer.
 - d)Select "Classic Load Balancer", choose Create.
 - i)Provide Required details like "Load Balancer name"

- ii) Click on "Next: Assign Security group"
- iii) Choose "create new security group:
and add http port requests
- iv) Click on "Configure Security Settings"
- v) Click on "Next : Configure Health Checks"
- vi) Click on "Next: Add EC2 Instances"
- v) Select the instances which you want to add.
- vi) click on Next
- vii) click on "Review and create".

7) Copy paste the DNS name into browser.

Classic ELB Practice Assignments:

Q1) Create new "classic" ELB in Sydney Region, Consider below

- a) Allow ELB to serve only "ap-southeast-2a" & "ap-southeast-2b" AZs
- b) Launch one ec2 instance in each AZ i.e ("ap-southeast-2a" & "ap-southeast-2b")
- c) Install tomcat7 in both EC2 instances
- d) add "8080" inbound request entry in EC2 instances SecurityGroup.
- e) Register both EC2 instances with ELB with proper configuration.

Note : In this case "ELB" Instance port must be "8080".

Q2) Create An ELB to run on port number 9090

Q3) Remove "ap-southeast-2b" AZ from Q1 ELB

Q4) Create an Classic ELB and enable "stickiness".

- a) Launch 2 ec2 instances and install apache2
- b) Change apache2 default page content in one ec2 instance.
- c) copy paste elb DNS name in browser multiple tab and check the output.

Application ELB:

Based on application context we can distribute the load between the application related servers

ex : Prepaid request need to be landed on prepaid servers.
postpaid request need to be landed on postpaid servers

Target Groups: Group the servers based on Application context. This process we call it as a Target groups.

Steps to Create Application ELB:

- 1) Launch four ec2 instances, Install apache2
 - a) Create Prepaid application in 2 servers
 - b) Create postpaid application in other 2 servers
- 2) Group the servers based on Application context. This process we call it as a Target groups.
- 3) Create Application ELB
- 4) Attach Target Groups to Application ELB
- 5) Define the traffic rules to send the request different target groups.

AutoScalling:

=====

- 1) Allows your AWS ec2 to grow or shrink depending on your workload
- 2) AS Ensures that you have the right number of AWS ec2 instances
- 3) Help you to save by cutting down the number of Ec2 instances when not need and scalling out to add more instances only when it required
- 4) AS always try to even distribution of the ec2 instances between the AZs
- 5) AS can span Multiple AZs within the Same region. Hence it can be used to create Fault Tolerant design.
- 6) No additional cost for launching AS group, you pay for what you use of EC2 instances.
- 7) AS can grow or shrink your EC2 instances base, according to your needs. Hence can play an Important role cost management.
- 8) It will work with AWS ELB, Cloud watch.
- 9) AS can Not span across multiple Regions
- 10) You can determine which subnets will AS group use to launch new instances in each AZ.
- 11) If AS fails to launch instances in an AZ (For AZ failure or capacity unavailability ..etc) It will try in the other AZs defined for this AS group until it succeeds.

AS Scalling policy Actions:

=====

- a) Scale Out : is the process of in which more EC2 instances are launched by scalling policy
- b) scale- In: is the process of in which in which EC2 instances are terminated by scaling policy

There are 2 Auto Scalling components:

=====

- 1) Launch configuration : it is the configuration template we need to defines parameters like Instance type, AMI, keypair, SG.

Note : Once template is created we cannot Edit it.

- 2) Auto Scalling Group (ASG): Here we define Scalling activity process related parameters like.
 - a) What type of scalling Activity Manual/Event based
 - b) Scalling Activity parameters like "max", "min", "desire"
 - c) AutoScalling helth check types

Note :

- a) ASG manage's ec2 instances based on defined AutoScaling policy parameters.
- b) ASG have 3 imp parameters
 - i) Min : The Min size of the ASgroup
 - ii) Max : The maximum size of the ASgroup
 - iii) Desire: The number of EC2 instances that should be running in the group. This number must be greater than or equal to the minimum size of the group and less than or equal to the maximum size of the group. If you do not specify a desired capacity, the default is the minimum size of the group

AS Rebalancing:

=====

- a) If AS finds the the number EC2 instances are not evenly distributed across AZ, It will initiate Rebalancing Activity.
- b) The target of this activity would be to reach an even distribution of instances between AZs.
- c) AS does this by launching new Ec2 instances in the AZs that have less Ec2 instances first, then terminating EC2 instances from the AZs that have more number of EC2 instances.
- d) In rebalancing activity AS temporarily launch instances more than what you have define in ASG "Max" value .
- e) The extra number of instances is less than or equal to 10% of max capacity value.

What can cause an Imbalance of EC2 instances Between AZs.

=====

- a) selected AZ is not having enough capacity instance type, family instances.
- b) adding removing AZs
- c) Manually requesting terminating of EC2 instances from your ASG.
- d) An AZ that did not have enough capacity, now has enough capacity and it is one of your ASG AZs.

Q) What will happen if the AS group was at or near max capacity when AZ rebalancing kicks in?

Type of AutoScaling:

=====

- a) Manual Scaling :
 - i) Maintains a current number of instance all the time
 - ii) Manually changes ASG's min/desired/max, attach/detach instances
- b) Scheduled based scaling: Used for Predictable load change.
 - i) you need to configure scheduled action for scale out at a specified date/time and to required capacity
 - ii) You can configure this action to happen once or at a recurring schedule.
 - iii) schedule action must have a unique date/time

iv) you can edit a schedule action after you create.

c) Event based scaling : Scaling in response to an event

Practice:

=====

Q1) Create Launch configuration in "Tokyo Region" with below parameters

- a) t2 micro instance type
- b) ubuntu 16 ami
- c) use any existing keypair
- d) use any existing SecurityGroup

Q2)

- a) Create an Autoscaling group (ASG) with 2 ec2 instance using Q1 Launch configuration.
- b) Once the Instances are launched, edit Q2(a) ASG to increase the max size parameters value to 4 and save it.
- c) Check any changes in number of instances launched by ASG.
- d) edit Q2(a) ASG to increase the "desired" size parameters value to 3 and save it.
- e) Check any changes in number of instances launched by ASG.

Q3) Delete Q2 ASG & and check ASG instances are deleted or not

Q4) Create an Autoscaling group (ASG) with 2 ec2 instance using Q1

Launch configuration. Consider below

- a) ensure that this ASG launch ec2 instance in "ap-southeast-2b" AZ only

Q5) Edit Q4 ASG and add "ap-southeast-2a" AZ and check how rebalancing is performed by ASG

Q6) Manually terminate any EC2 instance in Q4 ASG group and check the result

Q7) Manually Stop any EC2 instance in Q4 ASG group and check the result.

=====

AS-ELB Practice:

Q8) Create Launch configuration in "Tokyo Region" with below parameters

- a) t2 micro instance type
- b) ubuntu 16 ami
- c) use any existing keypair
- d) use any existing SecurityGroup (add Http inbound rule)
- e) Install apache2 using user data.

Q9)

a) Create an Autoscaling group (ASG) with 2 ec2 instance using Q8

Launch configuration.

b) Create New classic ELB .

c) EDIT Q9(a) ASG to attach ELB.

d) check ASG instances were automatically registered with Elb

e) Stop the apache2 service in one ec2 instance &
and check instance health status
in ELB & ASG

f) Start Apache2 service again in step(e) ec2 instance
and check instance health status in ELB & ASG

g) EDIT step(a) ASG and "Health check" type to ELB

i) Stop the apache2 service in one ec2 instance & and check instance health status
in ELB & ASG, This time ASG will replace the instance.

Imp Que:

=====

ELB is attached to the ASG, Instance state is showing "Outof service" In ELB
health checks, But in ASG Health checks it is showing "Inservice".

What might be the problem, Why there is a mismatch?

ASG Health Checks:

=====

a) AS clasifies its EC2 instances status as either Healty or Unhealty

b) By default ,AS uses EC2 status Checks only to determinethe health
status os an instance

c) When you have one or more ELBs defined with the AS group,
you can configure AS to use "both" the Health checks and the ELB
Health checks to determine the instances helath status.

Health Check Grace Period:

=====

a) It is the time AS waits from the time as instance comes into "Inservice"
before checking its health status.

b) A value of "zero" means no grace period and the
instance health is checked once it is Inservice.

c) Until the Grace Period timer expires, any unhealthy reported by EC2 status checkes,
or ELB attached to ASG,will not be acted upon.

d) After Grace Period expires,AS would consider an Instance unhealthy is any of
the following cases:

i) EC2 Status check reports to AS an instance status other than running.

ii)If ELB health check are configured to be used by ASG, If ELB reports the instance
status "Out-of-service"

iii) One source reports the instance as unhealthy is enough for AS to mark it for
replacement.

ASG - Unhealty Instance Termination:

=====

a) Once AS identifies an instance as unhealthy, it gets scheduled for termination.

b) Unlike AZ rebalancing, termination of unhealthy happens first, then AS attempts to launch new instance to replace the ones terminated

c) Elastic IP and EBS volumes get detached from the terminated instances, you need to manually attach them to new instances.

Attaching Existing EC2 instances to ASG

=====

a) Instance must be in running state

b) AMI used to launch the instance still exists

c) Instance is not part of another ASG

d) Instance is in same AZs of the AS group

e) If the existing EC2 instance under the AS group, plus the one to be added, exceeds the Max capacity of the ASG, the request will fail, EC2 instance won't be added.

Detaching Existing EC2 instances to ASG

=====

a) You can manually remove EC2 instances from ASG

b) You can then manage the detached instances independently, or attach it to another ASG.

c) When you detach the instance, you have the option to decrement the ASG desired capacity. If you do not, the ASG will launch another instance to replace the one detached.

ASG - Standby state:

=====

a) You can manually move an instance from ASG and put it in standby state

i) Instances in Standby are still managed by Auto Scaling

ii) Instances in Standby state are charged normal, In service instances

iii) AS does not perform Health Checks on instances in standby state.

Why we need to keep instance in Standby State?

=====

If you want to troubleshoot the instance or make changes in standby mode, without having the AS consider that as the instance being unhealthy.

Practice:

=====

Q10) Attach any existing ec2 instance to ASG.

Q11) Detach any existing AS group ec2 instance.

Q12) What happens if detach ec2 instance from ASG without changing "desired" parameter value.