AWS TRAINING NOTES- Part A

- Certified Solution Architect Associate (CSAA) Exam Roadmap: Practitioner (not necessary for next level) > **Associate level** > **Professional Certification**.
- 720 out of 1000 is required to pass the associate level certification exam.
- **Topic wise weightage for the new exam.**

Domain	% in the Examination
1: Design Resilient Architecture	**34%**
2. Define Performant Architecture	**24%**
3. Specify Secure Application Architecture	**26%**
4. Design Cost optimized Architecture	**10%**
5. Define Operation Excellent Architecture	**6%**

Contents

General Overview

Data center -> physical location where servers are hosted/located.

The components of Datacenter

- Network (*both physical and data security*)
- Security
- Server
- WAN
- Internet
- Storage *(Block Storage & Object Storage)*
- File Share

Multi-Tenancy: The process of hosting multiple client data/servers on a single piece of physical hardware is call multi-tenancy. Client's data/traffic should be separate from the other client (tenant); thus, security is an important aspect of multi-tenancy. AWS archive virtualization using Critix Xen hypervisor.

Different Types of Cloud offering

- Infrastructure as Service: Compute and storage alone.
- Platform as Service: Platform Service
- Database as a service: Database Service
- Software as a service: Complete software

Type of cloud:

- Private Cloud – on premises hosted virtualized server.
- Public Cloud – publicly hosted virtualized server.
- Hybrid Cloud – Mixed combination of Private and Public Cloud.

Benefit of Cloud Computing

- OPEX cost – no upfront cost.
- Reduce time to deploy
- Dev to Test environment made easier
- Elasticity – On Demand capacity
- Higher degree of automation can be archive.

Disadvantage of cloud computing are:

- Lack of visibility in areas that are maintained by service provider.

Top player: Cloud computing

Gartner's Magic Quadrant for Cloud Infrastructure as a Service, Worldwide June 2017.
Click here to embiggen

Region / Availability Zone:

- Geographically separated region where AWS is hosting their data center are called as **AWS Region.**
- Availability Zone: Are the data centers within an AWS region where servers are hosted.

AWS feature:

- Multi Tenancy
- Own Virtual private cloud (can configure your own security policy / firewall / network traffic router).
- Compute
- Database
- Storage

- Monitoring & logging (audit trail)
- Domain Name Service (DNS) – Route 53
- Security & Encryption
- Other services (bid data/machine learning/blockchain)

How to access AWS

- AWS console: web interface accessible from internet https://aws.amazon.com
- AWS SDK: Custom tailored APIs to be use with different programming platform like JAVA, php, python, node.js, .net, go, etc.
- AWS Command line interfaces: Command line interface to access AWS features.
- **AWS API**: All AWS services can be access through AWS API (Application Programing Interface).

Free Tier Account Limits

To be added.

Root Account

- Enable multifactor authentication.
- Don't use root account for day to work, as it has all access – compromising the password of a root can lead to a disaster. AWS suggest least privileges access policy, provide minimum access to perform the assigned task. Once new access is required, same can be provided on the fly.

IAM (identity and access management)

User – 5000 max users, allowed by the IAM users

Role – when application needs to use other services on your behalf, roles are use. Temporary access is provided using rotating key.

Groups – Logical grouping of the users.

Virtual Private Cloud (VPC)

- In each region default VPC created by default.
- VPC is region specific, they can span across multiple availability zone. VPC cannot span across region.
- One subnet can be hosted in a single availability zone.
- VPC can be per department / per environment.
- Components of VPCs
    - CIDER block of IP range can be selected from RFC1918 ranges
        - 10.0.0.0 – 10.255.255.255 (10/8) = 1,67,77,211 (very high)
        - 172.16.0.0 – 172.31.255.255 (172.16/12) = 10,48,571 (moderate)
        - 192.168.0.0 – 192.168.255.255 (19.168/16) = 65,531 (low)

          Formula to calculate number of IPs is **[(2) $^{32 - [\text{mask number}]}$ - 5 IP]**

          Each of the above mention IP range can be assigned within a private network; each address will be unique on that network but NOT outside the network.

- Through routing table, one subnet can interact with the other subnet. By default, routing table is attached to the VPC that allow routing between the connected subnet to route their request locally.
- Internet gateway – AWS managed service that connected instance in the subnet to interact with the internet ips.
- Security Group (last line of defense): Attached to EC2 instances.
- NACL (network access control list): Connected to the subnets.
- Virtual Private Gateway.
- IPV6 addressing: ALL Public address, there is nothing like private addresses like in case of IPv4 / AWS allocates the IPV6 addresses on its own.
- Implied router – connecting all subnet through routing table.
- Each subnet should be associated with ONLY one route table at any given time. IF no association is defined then the subnet is associated with the default route table.
- Each of the route table has a default entry to communicate with each subnet within the VPCs. This entry cannot be deleted or altered.
- 200 route table per VPC/ with maximum of 50 Routing entry per table.
- VPC can have any of RFC19818 IP address or any publicly routable.
- Maximum CIDER block is /16 which will have 2^{32-16} -5 = **65,531 address**
- Minimum CIDER block is /28 which will have 2^{32-28} -5 = **11 address**
- No overlapping IP addresses, it allowed in subnet attached to a VPC.
- VPC CIDER block can be expanded by adding secondary IP address. [NO need to plan the future extension, as it allows expansion].
- Limitation for adding additional CIDER block to the existing VPC
    - Common limitation of the CIDER block, as applicable
    - CIDER block must not be same or larger than any of the existing route table entry.
    - There are few restrictions on the RFC1918 uses for expansion.
- AWS reserves first 4 and last one (*total of 5 IPs are reserved by AWS per VPC*)
    - **First address reserved as its base IP address**.
    - **Second address is VPC router.**
    - **Third address is reserved for DNS routing.**
    - **Fourth one is reserved for future use.**
    - **Last address is for broadcasting, which is not allowed.**

      Example: for a range 10.0.0.0/25 = out of 32 bit – 25 bits is reserved for network and (32-25) = 7 bits can be used for hosts. Out of those available IP addresses
      The IPs that can't be use are
        - First IP **10.0.0.0**
        - Second IP **10.0.0.1**
        - Third IP **10.0.0.2**
        - Fourth IP **10.0.0.3**
        - Last IP 10.0.0.(2^7 -1) = **10.0.0.127**
- Internet gateway: ONLY one internet gateway per VPCs.
- Any public IP are not associated to the instance, instead in the Internet gateway (IGW) will have NAT (Network Address Translation) for that IP. It will translate public IP to Private IP.
- Routing table rules are executed from top to bottom.
    - First Rule is 172.31.0.0/13 – local (any IP that are from the subnet traffic is routed locally): **This route cannot be deleted or editable.**

    - Second rule is 0.0.0.0/0 – IGW (any other IP that are not from/to the above rule are routed to the IGW (internet).
- **Elastic IP address are charged even when they are NOT USE. They need to be de-attached from the AWS account to incurring charges.**
  [IMPORTANT] Elastic IPs are totally free, as long as they are being used by an instance. However, Amazon will charge you $0.01/hr for each EIP that you reserve and do not use. You will be charged if you ever remap an EIP more than 100 times in a month.
- In order to delete a resource, AWS ensure there is no dependencies.
- Security Groups:
    - There is virtual firewall, that are attached to the Elastic NIC cards.
    - Any traffic coming to the security group is inbound traffic (ingress traffic), Any traffic coming out of the security group is outbound traffic (egress traffic).
    - **Up-to 5 security group can be attached to an EC2 instances.**
    - In security group is ONLY allow RULE
    - Security groups are stateful.

Default VPC	Custom VPC
Created by default for every Region	Need to create manually.
By default - NACL - Security Group - Route table - Default CIDER IP addresses. - Internet Gateway connected to default Route table.	By default - NACL - Security Group - Route table - IP range user has to be selected. - No internet gateway connected by default.

- Creating VPC with wizard, there are four (4) options to chose from
    - VPC with single public subnet
    - VPC with public and private subnet
    - VPC with public and private subnet and hardware VPN access
    - VPC with ONLY private subnet and hardware VPN access.
- ~~VPC peering can happen only within a single region access different AWS account.~~
- Starting Nov 2017, AWS allow VPC peering across region. When peered the traffic across the VPCs across region goes through AWS backbone network NOT through the internet bandwidth.
- Inter region transferring of the data rate will be applicable.
- VPC is a one-to-one peering. Transitive Routing (a.k.a. edge-to-edge) routing is not allowed in VPC Peering.
- For VPC to peer successfully, the subnet CIDER block should not overlap with each other.
- VPC peering an AWS managed service, there is NO single point of failure.
- In order to have successful peering connection between two VPCs, one need to make sure the route table on the either side (both the VPCs) are altered and ensure that they have a route entry added that routes desired inbound/outbound requests to the peered VPC.
- VPC peering is an AWS managed service. There is no single point of failure. No need to create redundant.
- AWS does not allow to use NAT instance to be used through customer gateways / VPG (virtual private gateway) connection to allow access for internet access.

- Route propagation – When connected over the customer gateway and virtual private cloud, route table can update dynamically to make then aware of their counterpart's subnet routing information.
- Max of 10 VPN connection [**SOFT-LIMIT**] can be established to single VPN gateway. However, this limit can be extended by making a request to increase the limits.
- VPN CloudHub: Building on the AWS managed VPN and AWS Direct Connect options described previously, you can securely communicate from one site to another using the AWS VPN CloudHub. The AWS VPN CloudHub operates on a simple hub-and-spoke model that you can use with or without a VPC. Use this design if you have multiple branch offices and existing internet connections and would like to implement a convenient, potentially low cost hub-and-spoke model for primary or backup connectivity between these remote offices. AWS VPN CloudHub leverages an Amazon VPC virtual private gateway with multiple gateways, each using unique BGP autonomous system numbers (ASNs). Your gateways advertise the appropriate routes (BGP prefixes) over their VPN connections. These routing advertisements are received and readvertised to each BGP peer so that each site can send data to and receive data from the other sites. The remote network prefixes for each spoke must have unique ASNs, and the sites must not have overlapping IP ranges. Each site can also send and receive data from the VPC as if they were using a standard VPN connection.
  ANY subnet that is not advertise over the BGP will not be accessible on the other side. So NOT-ALL of the subnet gets exposed when connected over AWS cloud hub.
- **VPN – keywork fast/cost-effective/high latency**
- **Direct Connection Location: DX location**
- **VPC endpoints:**
- **VPC transitive routing :**
    - **Interface endpoints**
    - **Gateway endpoints: ONLY for S3 and DynomoDB**
- VPC Flow logs – that capture IP traffic pass through your VPC. This can be created at **VPC level**, **subnet level** or at **network interface level**. The logs can be store at **cloud watch logs** or in **S3 bucket**. It can capture IP traffic going-in or going-out of your VPC.
- DHCP Option Set:
    - On-premises DNS can be used for VPC environment resources.
    - AWS Route 53 service can't be use as DNS for on premises infrastructure. As for Route 53 the traffic need to coming from **within** or **for within** AWS network.
    - The Dynamic Host Configuration Protocol (DHCP) provides a standard for passing configuration information to hosts on a TCP/IP network. The options field of a DHCP message contains the configuration parameters. Some of those parameters are the domain name, domain name server, and the netbios-node-type.

Security Group (SG)

- Security group can be a source of a security group or can be destination of a security group.
- Security group are regional.
- Security group are implicitly denial.
- Default rule of the security group, all traffic from instance which has the same security group attached. As it has allow-all rule with inbound destination as security group of itself.
- Security groups are tied up to VPCs, it can't be shared.
- Any change in the security group is affected immediately.
- Instance guard by security group / subnet guard by NACL.

Default Security Group in Default VPC	Default Security Group in custom VPC
- All inbound traffic originated from the security group are allowed. - All outbound traffic is allowed.	- All inbound traffic is denying. - All outbound traffic is allowed.

Network Access List (NACL)

Security Group	NACL
Operate at instance level	Operate at subnet level
Stateful	Stateless
ONLY allow rules	Allow & Deny both rules can be configured
Evaluate all the rules first before deciding on if the network traffic is allowed or denied.	Each rule has a number – rules are evaluated from smallest to the highest.
Need to attach to instances.	Automatically attached to the subnet.
Last line of defense.	First line of defense.
Every time an instance needs to be communicated this comes into picture.	Explicit deny rules added to the NACL
	ONLY comes into picture when instances need to be communicated outside of the subnet.

NACL can block certain range of IP addresses from getting into your instances, as it has denied rules same cannot be done by security groups as they don't have deny rules. Also, its advisable to block the IP ranges at first line of defense.

Note: Router will translate instance private IP addresses into public IP address or elastic IP addresses.

Within a same VPC, route table will have default entry that will all communication within the instances of the VPC.

NAT instance & NAT gateway

Public Subnet instances that need to be connected to internet (outbound connectivity) should be pass their traffic through a NAT instance / NAT gateway. Each NAT gateway maintain a PAT that allow single PUBLIC IP to send back the reply to desired connected instance.

The NAT instance should be place at the public subnet for it communicate to the internet.

To protect the NAT instances security group will be attached. A NAT instance security group must allow:

- Traffic instance security group or the private subnet's security group as a source on port 80(HTTP) and 443(HTTPS).
- Traffic Outbound to 0.0.0.0/0 (internet) on Ports 80 and 443.
- Traffic inbound from the customer's own network on port 22(SSH) to administer the NAT instance.

For NAT instances Source-Destination check needs to be disable, by default this is enable. By default, all instance ONLY allows those traffic that are originated by their IP OR sends to their IP, but in case of NAT this this different as NAT instance acts a proxy, source-destination check needs to be disable.

NAT Instances	NAT gateway
Use needs to maintain the instance. AND will be charged for the use of the instance.	AWS maintained the instance, AND user needs to pay for hours of usages.
Uses needs to add appropriate security group to secure NAT instance	AWS takes care of the security.
Can work with public and elastic IP addresses	Works ONLY with elastic IP addresses

There are three type of IP address

Private IP address	Public IP address	Elastic IP address
Non publicly routable IP address.	These are AWS allocated publicly routable IP addresses.	These are static publicly routable IP addresses. They need to be allocated to an AWS account, up-to 5 elastic IP addresses can be allocated. When elastic IP address are in use, they are free, but if they are allocated and not in use they are chargeable.

Direct Connect (DX)

- Direct Connect is a NON internet based and provide high speed, low latency, and higher performance then internet VPN.
- VIF (Virtual InterFace) is basically an 802.1Q VLAN mapped from the customer router to Direct Connect router.
- Private VIF connect to private VPC and Public VIF connect to public VPC & public AWS service.
- Layer 2 connection cannot be established over Direct Connection.
- You cannot use NAT instance hosted on the VPC to route datacenter traffic over internet.

Elastic Cloud Compute (EC2)

- EC2 purchase option type = On-Demand/Spot Instance / Reserved Instance
- EC2 tenancy type = Shared / Dedicated host
-

- 99.95 % up-time means ~ 22min per month downtime.
- Two types of BLOCK store
    - **Elastic Block Store (EBS)**
        - Can be use as root volume
        - Persistence store
        - Network attached virtual drive
    - **Instance Store**
        - Can be use as root volume

- NON-Persistence store
- Hard drive at the virtual host where the instance is running.
- Max up-to 10 GB can be allocated per block store device

Types of Instances:

General Purpose	Compute Optimized	Memory Optimized	Graphic Compute Instance.
Balance CPU & memory.	More CPU then Memory.	More memory then computes.	Graphic optimized
Suitable for most of the purpose.	Compute & High-performance compute optimized.	Memory intensive app, DB and caching	Hugh performance and parallel computing
Example: T1/M3/M4	Example: C1/C2	Example: R3/R4	Example: G2

Storage Optimized
Very High I/O, low latency
I/O Intensive App, data warehousing, Hadoop
Example: I2/D2

Types of EBS:

General Purpose – SSD	Provision IOPS	Throughput Optimized HDD	Cold HDD
SSD backed	SSD backed	HDD backed	HDD backed
Best suited for transactional workload, small databases, DEV/Test environments, Low latency interactive app.	Mission Critical application, I/O intensive, SQL and No-SQL databases. Provide suitable IOPS performance and low latency.	Ideal for streaming, big data, log processing, data warehousing. This cannot be use as Root volume. Mostly use for frequently access, throughput intensive workloads. **Cannot be use as root volume.**	Ideal for low throughput requirement. Use of less frequently access workload. **Cannot be use as root volume.**
Sizes 1TiB – 16 TiB	Sizes 4TiB – 16 TiB	500 GiB to 16TiB	500 GiB to 16TiB
Max IOPS 10,000	Max IOPS 32,000		

Magnetic EBS (HDD)
HDD back
Low IOPS. Use for transactional workloads where performance is not dependent on the IOPS.
Use for infrequently access data
Size 1GiB to 1TiB

Note: in associate level exam, the questions will be more to test understanding of the different type and their use-cases rather than IOPS numbers. However, in case of professional exam, its important to know the IOPS value for each individual type as there might be some scenario base questions where one may have to chose answers based on the IOPS value also.

- Block Storage Mapping – mapping data volumes (root & data) to the AMI.
- EC2 instance store mapping only shows mapped EBS volumes, when queried from AWS console. To view the instance store mapping one need to queried the instance meta-data information related to block device mapping.
- Instance configuration can be changed after the launch (for running instance) – EBS volume can be added or removed to/from the instance configuration however instance store cannot

be added after launch. The following three property of the EBS Root volume can be changed after instance is created

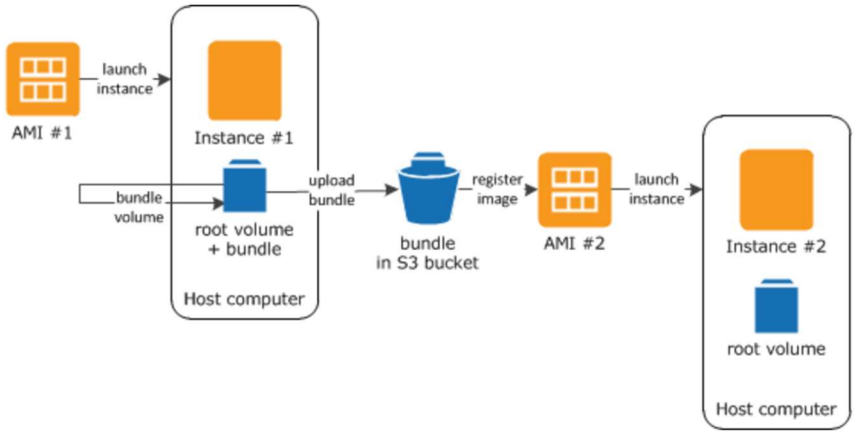
- Volume size – increase the volume size. **Volume size cannot be decreased.**
- Volume type – change from general purpose to provision IOPS
- Change Delete-On-Termination flag

More Info : https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/block-device-mapping-concepts.html

- Point in time snapshot can be created for the EBS volume
  - This can be done manually
  - This can be done using life cycle management.

Root Volume	EBS Volume
By default, delete-on-termination flag is selected.	delete-on-termination flag is NOT selected, by default. Use can select it while launching or at running state.
Cannot be encrypted.	Can be encrypted.

- Default basic monitoring option available with EC2 instances are :
  - CPU utilization
  - Disk Read (bytes)
  - Disk Read Operation
  - Disk Write
  - Disk Write Operation
  - Network in
  - Network out
  - Network Packets In
  - Network Packets Out
  - Status Check Failed – ANY (count)
  - Status Check Failed – INSTANCE (count)
  - Status Check Failed – SYSTEM (count)
  - Credit Usage (count)
  - Credit Balance (count)
- Steps to create an EC2 instances with Root volume encryption
  - Create an EC2 instance based out of an AMI
  - Create an AMI out of the EC2 instance.
  - Copy the AMI to your desired region (desired region can be the same region also).
  - While creating the instance – check encrypted checkbox.
- If Instance starts – pubic IPV4 address from the pool gets allocated.
- If Instance is terminated / stop – pubic IPV4 address are release back to the pool.
- If Instance is rebooted – pubic IPV4 address is retained.
- To create an AMI out of Instance store powered EC2 instances – one need to use AWS CLI interface. AWS console doesn't give direct access to create AMI like in case of EBS backed EC2 instances. More details can be found in the below link: https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/creating-an-ami-instance-store.html
  High level process overview of the Instance Store AMI creation is :



EBS Optimized EC2 instance:

- EBS volumes are not part of the same hardware where Instance is running, the are part of AWS infrastructure connected by AWS network. The EC2 instances optimized to take full potential of the supported IOPS are called as **EBS optimized EC2 instances**.
- They are designed to work with all type of EBS types – General Purpose, Provision IOPS, Throughput optimized and Magnetic HDD
- But not all ec2 instance family are EBS optimized, only instance that are **SR-I/OV (*Single Root – Input/output virtualization*).**
- In case of SR-I/OV (***Single Root – Input/output virtualization***) EC2 instances, the host's NIC (network interface card) is virtualized and share with the guest instances unlike other instances where host NIC is made available to the guest instances using the virtualization layer.
- SR-I/OV is support by R3/R4, X1, P2, C3/4, I2, M4, D2.
- **EBS enhanced EC2 instances can be enable for EBS backed EC2 instances as well as Instance Store backed EC2 instances.**
- EBS enhanced EC2 instances can be enabled for multi AZ setup.
- EBS enhanced EC2 instances are supported for
    - Instance type should support SR-I/OV
    - Instance type should be created from Hardware Virtual Machine (HVM) not supported by Paravirtualization.
    - It's not supported for EC2 classic. Instance should be launch in VPC.
    - There is NO extra cost in turning on the EBS optimized feature for EC2 instances (if supported).
- Placement Groups: Is a logical grouping (clustering) of the resources in a same AZ or in different AZ, with goal to reduce latency and high network throughput by determining how the instance needs to be placed on the host hardware. This can be achieved using one of the two strategy
    - Cluster: In cluster placement group setup all instances are placed in a single AZ (Availability Zone)
    - Spread: In spread placement group setup instances are placed across multiple AZ (Availability Zone)
- There is NO additional cost involved in placing instances in placement group.
- Placement Group Best practice:
    - Ensure all instance are SR-I/OV enabled.

    - Launch all instance together to guarantied allocation, if the instance capacity ran out, capacity error will be thrown when launching additional instances.
    - Try avoiding launching more than one instance type in a single placement group, to ensure capacity allocation.
    - Can create a placement group across VPC peering connection. Both the VPCs should be in the same region.
    - When the traffic is within instances, better to use cluster placement group.
    - If instance is rebooted, then need to re-allocate space in underline hardware. If the requested instance type capacity ran out then capacity error will be thrown.
    - To ensure critical application that has small number of instances, by placing then in spread placement group it ensures there is no single point of hardware failure. As the instance will be hosted in separate hardware.
    - Within instance group, instance can communicate using private IP as well as public IP.
- EC2 instance status check:
    - EC2 Service status check
    - EC2 instance status check
    - By default, EC2 instance status check will evaluated. The outcome of the EC2 instance status check is – PASS or FAIL
    - For EBS backed volume AWS automatically restart the instance, doing so the instance get hosted in a different underlining hardware which mostly resolves the problem.
- EC2 service monitoring
    - BY default, EC2 service sends basic monitoring metric to cloud watch for every 5 mins (basic monitoring).
    - A detailed monitoring for every 1 min can be enable, with additional cost.
    - Based on EC2 service metric CloudWatch alarm can be set, depending on the cloudWatch alarm – stop/restart/terminate action can be performed.
- Different state of EC2 instance state

  Pending → Running → Instance receives private as well as public IP addresses. → **Stop** AWS shutdown the instance.

    - When instance is re-booted its still consider as running state. There is additional hour added into billing, however stopping and starting an instance adds additional hour into billing.
    - Stopping a running instance, maintains instance ID & Root Volume (if its EBS backed). Instance store backed EC2 instance cannot be stop, they can be either rebooted or terminated.
    - For stopped EC2 instances, they are NOT charged however for the EBS volume they will be charged.
    - For stopped EC2 instance, root as well as data volume can be detached and reattached.
- What happed when an instance is stopped?
    - Instance perform shutdown.
    - Instance state change from running → stopping → stopped.
    - EBS volume remain attached to the instance and it incur charges for that.
    - EC2 instance doesn't incur any charges.
    - Any data stored in RAM or in instance store are Lost.

    - When the instance is restarted the instance may start in the same underline hardware server or a different hardware server.
    - Instance IPv4 private address and IPv6 private/public address are retained.
    - Instance IPv4 public address release back to AWS pool.
    - Instance retains its Elastic IP addresses.
- EC2 Instance Termination

  Running → Shutting down → Termination
- What happened when an EC2 instance is Termination?
    - It will NOT incur any cost.
    - Instance Store volume data is lost.
    - By default, EBS root(boot) volume is deleted and data(non-root) volume attached to the EC2 instance is retained.
    - Above default behavior can be changed, by turning on the flag – ***deleteOnTermination*** flag during launching of the instance, during instance is running or stopped.
- Enabling Termination Protection on EC2 instances prevents the EC2 from accidental termination this can be enable on both EBS backed and Instance Store backed instances.
- For instances with termination protection ON , if the instances need to terminated using cloudwatch alerts then it would fail. Workaround is instead of termination try shutting down the instances for such instances.
- The flowing are the possible reasons why an instance state may move from pending to termination when launched
    - If the instance store backed AMI used for launching EC2 instances is missing or part of it is missing.
    - Limit is reached for max number of EBS volume can be attached to EC2 instance for a particular account
    - EBS snapshot is corrupted.
- Instance Metadata
    - Data about the instance.
    - This can be found – **http://169.254.169.254/latest/metadata**
- User data is the script that can be pass at the time of launching. Its limited to 16 KB only, and can be change after stopping the instance. Stop instance → Instance → Action → Instance Setting → View/change user data.
- Migrating VM from/to from VMware, Microsoft Hyper V, XEN – VM Connector automate the migration of the VMware VM into AWS. More infor https://docs.aws.amazon.com/amp/latest/userguide/migrate-vms.html
- IAM Role: IAM role can be attached to EC2 instance which enable EC2 instances to perform task on behalf of the user without any need to store IAM username/password within EC2 instances.
- Banstion Host – jump box instance which can be used to administrator other EC2 instances. AWS call baston host for Lunix instance OR remote desktop for windows instance.

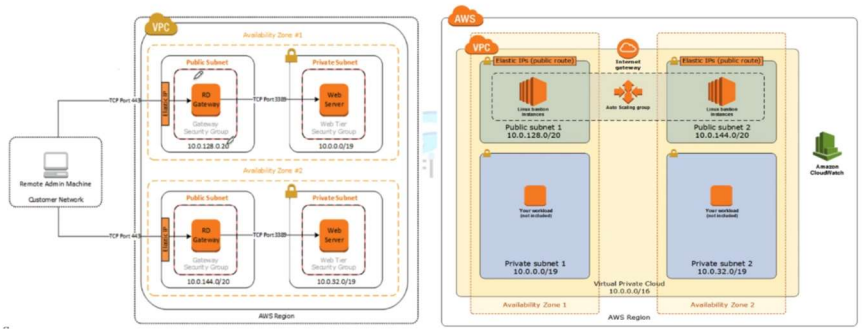


Figure 1 Bastion Host HA architecture

- EC 2 purchasing option
    - On demand instance [highest price]
    - Reserved instance [paying for long term capacity for specific availability zone /region]
    - Spot Pricing

Reserved instance – are of two types availability zone scope OR region scope

- Availability Zone scope reserved instance guaranteed capacity on that particular availability zone.
- Availability Zone scope reserved instance can be sold in marketspaces, region scope reserved instance cannot be sold in marketspaces.
- Following things can be changed for the reserved instances
    - Change the Availability Zone for Availability Zone scope reserved instance.
    - Availability Zone scope reserved instance to region scope reserved instances.
    - Change the instance type in the same family
    - One need to send request to AWS to make modification to Reserved Instances.
- Elastic Network Interface (ENI)
    - By default, eth0 primary ENI card will be added to all EC2 instance
    - During launch eth1 can be attached to the EC2 instance
    - After launch, more ENI can be attached to EC2 instance based on the instance type family.
        - Attaching ENI when the instance is on **running** state is called **HOT ATTACH**.
        - Attaching ENI when the instance is on **stopped** state is called **WARM ATTACH.**
        - Attaching ENI when the instance during launching state is called **COLD ATTACH.**
    - If additional ENI is attached to the EC2 instances during launch, then AWS will NOT allocate IPv4 public IP addresses to any of the ENIs, one need to attach elastic IP addresses manually. [*related to scenario base questions*]
    - When instance is terminated, default ENI eth0 is terminated by default however if other ENI are attached to the instance those are not terminated by default. This behaviour can be changed as required. Even ENI create through CLI are not terminated.
    - Each of the Elastic Network Interface can have
        - One Description.
        - One Primary IPv4 address.
        - One or Many Secondary IPv4 addresses .

        - One IPv4 public addresses assigned automatically.
        - Max Upto 5 security groups
        - One MAC address
        - Source Destination check flag
    - The benefit of havening multiple IPv4 address are
        - It allows hosting multiple website in a single server, with each IP addresses mapped to a single SSL certificate.
        - It allows network and security appliance to be use in VPC.
        - Routing internal traffic to a standby EC2 instance on an event of primary instance failure. This can be done by remapping of the secondary IPv4 address to a secondary (standby) instances. [When EC2 instance has secondary IPv4 private addresses, it allows reassigning of the secondary IPv4 address to new EC2 instance so that on an event of a failure, the traffic routing from elastic IP address to the secondary private IPv4 address get automatically routed to the new instance].

          NOTE:

          # Secondary IPv4 addresses should also be from the same subnet of the network interface or the EC2 instance.

          # Each private IPv4 address can be mapped to a single Elastic IP address (one-to-one relationship).

          # Once the primary IPv4 private assigned to a elastic IP address, is de-attached from the EC2 instance then Elastic IP address is also gets de-attached.

          # One cannot de-attached eth0 (primary subnet) from the instance.

          # Private IPv4, Secondary IPv4, IPv6, Elastic IP address all belongs to the same network interface even when the are de-attached and re-attached to a different EC2 instance.

          # When attaching Network Interface to another EC2 instance – the second instance should be in the same region and should be in the same Availability Zone. Within Availability Zone, they can be in a different subnet.

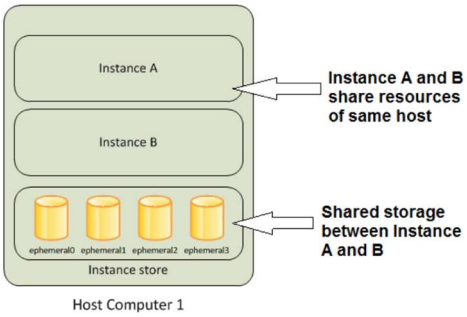
          # ENI are AZ specific
    - Source Destination Check fag [**question scope**]
        - By default, for all EC2 instances source destination check is enabled which restricts all traffic that leave the instance should be sourced from within.
        - For NAT instance, which acts as a proxy for the private subnet instance (instances that does not have direct connectivity to the internet) traffic that leave the instance is NOT from within the instance but from another instance. Thus, its important to disable the source destination check for NAT instances.

Elastic Block Store (EBS)

- There are two broader types of storage available on internet
    - Block store – this is primarily used for database, instance storage, for installing application etc.
    - Object store – for storing picture, songs, datafiles, logs files etc.
- Block store has two types

- **Elastic Block Store (EBS)** is a network storage that is attached to any EC2 instance over AWS backbone infrastructure.
    - EBS backed EC2 instance, means the root volume of the EC2 instance is hosted on an ESB.
    - EBS backed EC2 instance can be stopped, restarted, terminated.
    - When stopped the data stored in the EBS volume is NOT lost.
    - By default, EBS backed EC2 instance have deleteOnTermination flag = ON, which deletes the EBS volume when its associated EC2 instance is terminated, however deleteOnTermination flag can be turn off if required to prevent the data on the EBS to persist when the EC2 instance is terminated.

- **Instance Store** – This is the part of the host where EC2 instance is running. Once the instance is terminated the data stored in the instance store is lost.



Host Computer 1

    - Instance Store backed EC2 instance, means the root volume of the EC instance is hosted on an instance store.
    - Instance Store backed EC2 instance cannot be stopped, they can ONLY be restarted or terminated.
    - When EC2 instance backed by instance store is rebooted the data store on it will NOT be lost.
    - If the instance is terminated the data store on the instance store volume is LOST.
    - Instance Store AMI are stored in S3 instances

- EBS volume are **replicated on multiple server in a single availability zone**, to prevent single point of failure on an event of any AWS component failure.
- **Note: EBS backed EC2 instance can also have instance store which means – the root volume of the instance is on EBS and additional drive(s) which is/are hosted on instance store is attached to the EC2 instance. When such instance is stopped the data store on the instance store drives will be lost.**
- Instance Store are also called as **Ephemeral Storage**.
- Instance Store block store has higher IOPS then the its equivalent EBS block store. This is mainly because the instance store drive is on the same physical host where the instance is running whereas the EBS drives are on the NAS storage drive which is connected to the EC2 instance.
- EBS Snapshot
    - This are point-in-time copy of the EBS which can be used to recreate another image of the EBS.
    - EBS snapshot are store in S3 Bucket – however they cannot be access directly ONLY way to access them is through EC2 APIs.

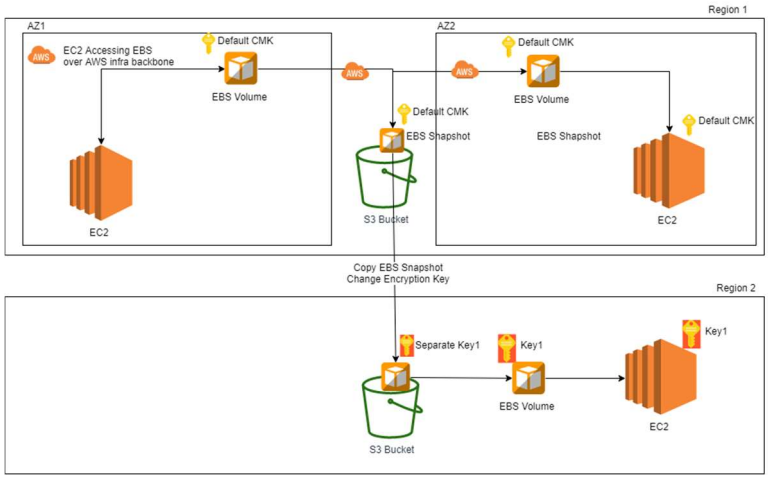
- Instance Store backed AMI snapshot are stored in user defined S3 bucket BUT EBS backed AMI snapshots are stored in S3 which cannot be access directly. ONLY through EC2-APIs these can be access.
- Max allowed limit, 5000 EBS volume & 10000 EBS snapshot per account.
- Characteristic of EBS Snapshot
  - They are region specific – while EBS are AZ specific snapshot are region specific. All AZ of that particular region can access the snapshot.
  - To migrate a EBS volume from one Availability Zone to another Availability Zone, create a snapshot of the volume and then create an another EBS volume in the desired AZ from that volume.
  - **[Important Exam Question]** While restoring the snapshot, the size of the new EBS volume should be same or larger than the original snapshot.
  - While EC2 instance is being accessed, EBS snapshot can be created for the NON-ROOT volume. However, any data cached by the OS or in memory will NOT be written into EBS snapshot. The snapshot will not be 100% consistent.
  - Phase of snapshots
    Snapshot request received → Snapshot **Pending** status [till the time all the data are written into S3 bucket the status remains in pending state] → Once all the data is copied successfully the status will change to **Complete**.
  - Best Practice for taking snapshot: Stop the EC2 instance then take the snapshot of the ROOT volume. For NON-ROOT volume, detached the snapshot from EC2 instance to ensure no read-write operation on the EBS volume. Once the EBS snapshot is triggered, the EBS volume can be attached back to the EC2 instance. [while the EBS snapshot is in pending state, it can be re-mount the instance].
  - Incremental Snapshot – ONLY changes that are added post EBS snapshot will be stored as incremental snapshot. [Cost effective]. Single full copy of the snapshot is need; rest can be incremental snapshot to be cost effective.
  - EBS snapshot charges
    - Cost incur due for the data transfer from the EBS volume.
    - Cost of S3 storage.
    - No charges for creating snapshot.
- EBS Encryption
  - EBS encryption is supported in all EBS volume types and for all EC2 instance families.
  - Snapshot of the encrypted volume is also encrypted.
  - Following are the ways to encrypt EBS volumes (EBS encryption at rest)
    - Use encrypted EBS volume.
    - Use 3rd party tool to encrypt EBS volume
    - Use OS level encryption, using plugins and drivers.
    - Encrypt data at the application level before storing it to a disk (EBS).
    - Use Encrypted file-system on top of the EBS volume.
  - Encrypted EBS volume are to be use just like how one use non-encrypted volume. EBS Encryption is handle transparently.
  - Single EC2 instance can be attached to both encrypted and non-encrypted EBS volumes at a same time. E.g. root volume is non-encrypted, while other data volumes (non-root volumes) are encrypted.

- There is NO direct way to change a sate of the EBS volume from un-encrypted EBS volume to encrypted EBS volume, the following are the in-direct ways to archive the same.
    - Attach a new encrypted EBS volume to the EC2 instance and transfer data from un-encrypted volume to the encrypted volume. Once all data are transferred successfully de-attached the un-encrypted EBS volume.
    - Create a snapshot of the un-encrypted volume, then copy the newly EBS snapshot to another snapshot while doing check "encryption" option the resulted snapshot will be encrypted. Create an EBS volume from the new encrypted snapshot.
      [EBS volume created from an encrypted snapshot is always encrypted. EBS volume created from an unencrypted snapshot is always unencrypted.]
- Root volume encryption: There is no direct way to change the encryption state of an EBS volume. However, AWS offers the are following in-direct way to encrypt the root volume.
    - Create an EC2 instance from un-encrypted volume attached to it.
    - Create an AMI from the above created EC2 instances.
    - Copy the AMI to another location, while doing so check the encrypted option.
    - The newly copied AMI will be encrypted.
    - Create a new EC2 instance from the encrypted AMI, the new instance will have the root volume as encrypted.

      While creating the snapshot of a root volume, make sure EC2 instance is stop.

- Encryption Key:
    - When encrypting the first EBS volume, AWS generates a default CMK (Default Customer Managed Key). Any encrypted snapshot created from the first encrypted volume or the any image created from that snapshot will be encrypted using the same default CMK. Default CMK will be managed by AWS Key Management Service (KMS).
    - Any newly created encrypted volume after first encrypted volume will have their own unique/separate AES256 bit encryption key. Which will be use in encryption of volume, its snapshot or any other volume created from the snapshot.
    - In order to change the encryption key, one need to create a copy of the snapshot in the process encryption key can be changed.
- To use an EBS volume outside AZ, one need to create an EBS snapshot. EBS snapshot are region specific all AZ in the region will have access to the EBS snapshot to create EBS volume from the snapshot.
- To move/migrate an EBS volume from one region to another region, create an EBS snapshot of the EBS volume and copy the snapshot to the desired region and create a new EBS volume from the copied snapshot.

  EBS volume are AZ specific, EBS snapshot are region specific.



- One cannot share a snapshot with other AWS account encrypted with default CMK, as the default CMK are assigned to a customer and cannot be exported from the AWS account. [One can share encrypted snapshot with the other AWS account, when encrypted with different key other than default CMS key.]

- Sharing snapshot:
  - **Unencrypted snapshot** can be share with all AWS accounts by making them as public.
  - Encrypted snapshot cannot be shared with all AWS accounts – there is no option to make encrypted snapshot public.
  - To share an encrypted snapshot with other AWS account one need to
    a) Ensure the snapshot is encrypted using non-default CMK. Default CMK cannot be share with other AWS account.
    b) Provide **Cross-Account-Permission** in order to provide access to the CMK with which the snapshot is encrypted.
    c) Mark the snapshot as private and provide the AWS account ID with whom the snapshot needs to be shared.
    d) The target AWS account with whom the snapshot is shared needs to create a copy of the snapshot. ONLY from the copy snapshot they can re-create an EBS volume in their account. **[Recommended Best Practice]:** When copy a encrypted snapshot from other AWS account, its always advisable to change the encryption key so that copy snapshot and its EBS volume are NOT dependent on the another account CMK (customer manage key). So that in future if **Cross-Account-Permission** is revoked, the snapshot and the EBS volumes are still remain accessible**.**
- Copying snapshot is used for:
  - Moving a snapshot from one region to another region.
  - Changing the encryption key of the snapshot.
  - Copy import/export service, AWS marketplace, AWS storage gateway.
- Use case for snapshot copying
  - Geographical expansion – extending service offering from one region to regions.

- For Disaster recover setup – setting up remote location for disaster recovery.
- Data retention and backup.
- Changing region – moving service offering from one region to another region.
- For encrypting an unencrypted snapshot.
- For changing encryption key.

▪ Other key pointers:

✓ While copying snapshot the tag associated with the original snapshot does not get copied.

✓ While copying snapshot from a different AWS account, if the account does not have **cross-account-permission** then the copy process will "**fail silently**".

✓ At a time up to 5 snapshot copy can be initiated.

✓ While the copy is in progress EBS volume cannot be created out of the new snapshot.

✓ When a CMK is deleted a retention period of 7 days is enforced to ensure the key is not use elsewhere. Once a key is deleted anything encrypted using that key will be unusable.

o Creating AMI

▪ Benefit of creating on the AMI

a) Standardization/Baselining of the EC2 images across organization.

b) Reduce manual effort of creating an EC2 image from basic AMI, OS patching, software installation, software configuration etc.

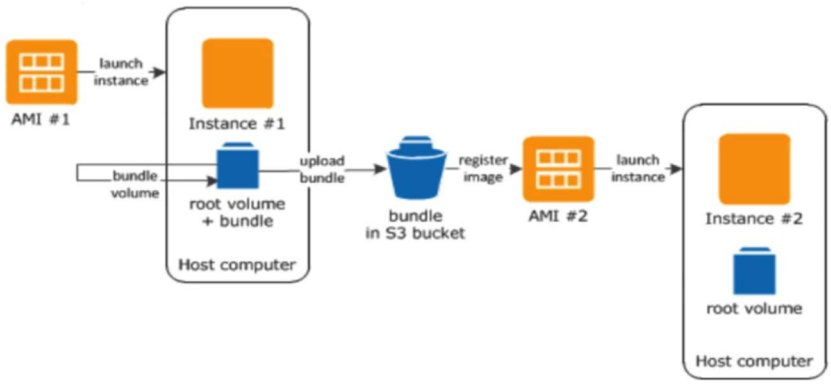
▪ Steps to create Instance store backed AMI (root device as instance store backed)

a) Create an EC2 instance from AWS Instance store backed AMI

b) Update the EC2 instance with required patching, software installation and software configuration.

c) Create an AMI from the EC2 instance, for instance store back EC2 instance the AMI copy will be stored in customer define S3 bucket.

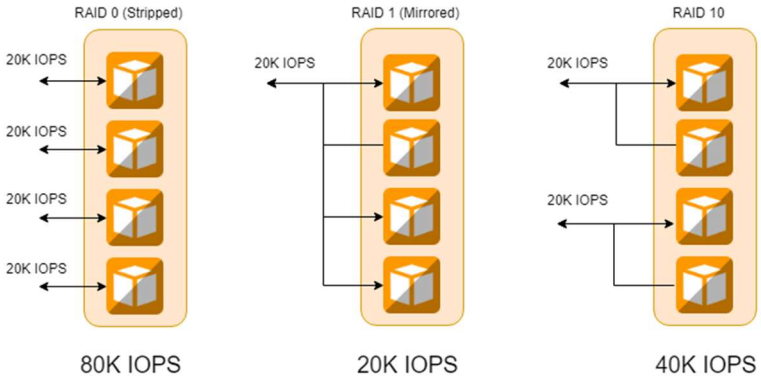
d) Instance store backed AMI needs to be manually register so that further EC2 can be created from the AMI instance.
Since the instance store backed AMI are stored in the S3 bucket, S3 storage charges will be applicable, until AMI is deregistered and all objects related to that AMI are deleted from the S3 bucket.



e) Once an Instance store backed AMI is de-registered, the EC2 instances are created prior to the de-registering of the AMI instance will NOT be impacted.

- Steps to create EBS store backed AMI (root device as EBS store backed)
  a) In-order to create an AMI from the EBS backed instances, one need to first freeze I/O for that instance by detaching the EBS volume from the instance or by stopping the EC2 instance. EBS snapshots are point in time snapshot, to create 100% consistent snapshot one need to freeze I/O. In doing so one need to keep in mind that I there are any data stored in the instance store volume attached to the EC2 instance that data will be lost and will not be available on the AMI.
  b) When creating AMI from an EBS backed EC2 instances, AWS will automatically register the AMI unlike in case of Instance store backed AMI user needs to manually register the AMI.
  c) In case of EBS backed AMIs, user don't need to specify the S3 bucket where the AMI or its EBS snapshot are stored. AWS does this on behalf of the user.
  d) In order to delete the snapshot of the EBS backed AMI, one need to first de-register the AMI, then delete the EBS snapshots of the AMI. Deregistering of the AMI will NOT automatically delete the EBS snapshots attached to the AMI. User needs to delete the EBS snapshots manually post deregistering of the AMI.

- Configure RAID Array
  - Combining multiple EBS volume to create a single EBS volume for better I/OPS and larger diskspace can be achieve through RIAD Array.
  - EC2 IOPS is dependent on NIC and the connected EBS's IOPS. By combining multiple EBS volume in a RAID Array higher IOPS can be achieved, which leverage multiple NIC card together to transfer data at higher IOPS. RAID is supported at the OS level, EBS volume supports any type RAID Array.
  - RAID Array IOPS is dependent upon EC2 instance supported IOPS, and individual EBS volume supported IOPS.
  - Its NOT advisable to use RAID array as ROOT volume of the EC2 instance.
  - Different Type of RAID array are
    - Stripped (RAID 0) [**Higher IOPS, NO Redundancy**] – where multiple volumes of EBS are connected in parallel to improve IOPS. The resulting IOPS is the sum of the all connected EBS volumes IOPS. However, there are NO redundancy build on the stripped RAID, single disk fails entire RAID Array fails.
    - Mirrored (RAID 1) [**No IOPS improvement, Greater Redundancy**], where multiple EBS connected in and ARRAY and data is written to all the EBS volumes in parallel.
    - RAID 10 [**Improve IOPS with redundancy**] combination of both RAID 1 and RAID 0 which takes benefit of Stripping along with mirroring.



Elastic Load Balancer

Elastic Load Balancer is an AWS service that helps to balance network traffic coming from internet to EC2 instance. There are two types of Elastic load balancer – ***Classic Load Balancer*** and ***Application Load Balancer*** (Layer 7). From the associate exam prospective – its classic load balancer that is in scope.

Elastic load balancer caters to the following traffic protocols - HTTP/HTTPS/SSL/TCP IP. NOT all traffic coming to the instance not necessarily needs to pass through the elastic load balancer.

ELB Network Load balancer support – TCP/SSL – Layer 4 protocol and ELB Application Load Balancer supports HTTP/HTTPS – layer 7 protocols.

ELB Listener connected to the backend servers – they also monitor connection request to ensure that the connection to the backend server or services are healthy.

ELB are charged hourly basis for the service they provided, once the status change to in-service the ELB charging starts. In order to avoid any charges incur from ELB one need to delete ELB however deleting ELB does not deletes the backend EC2 instances or the EBS volumes.

ELB route the traffic to the primary Ethernet address (eth0).

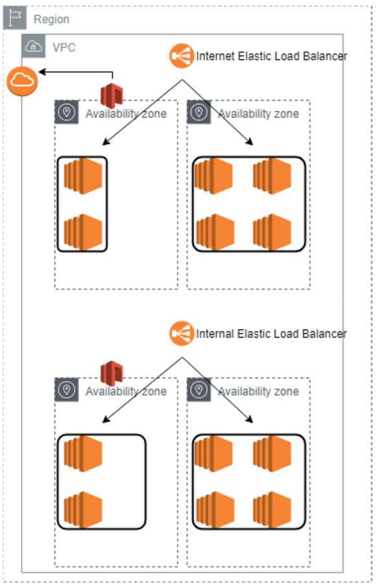
ELB health check – EBL monitor the health of the register instances to which it configured to route its request. ELB continuously monitor the register instance health till the time they are healthy its routes the request to its backend request, once health checks fail, ELB stops forwarding request to that instances. Healthy instances are represented as “in-service” instances, un-healthy instances are represented as “out-of-service” instances.

ELP configuration	Description	Default value	Configure Range
Response Timeout	Time by when the ELB should received HTTP 200 status for the health check	5 Sec	Can be configure any time between 2 sec to 60 sec.
Health Check Intervals	Intervals between two probs	30 sec	Can be configure any values between 5sec to 300 sec.
Unhealthy Threshold	Number of consecutive failed probing after which an instance will be	2	Can be configure any value between 2- 10

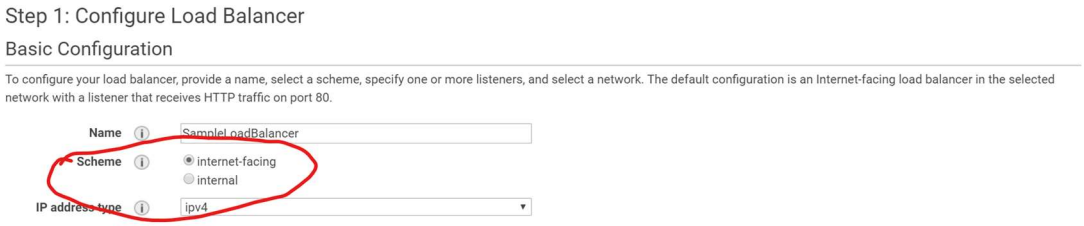
	marked as "out-of-service"		
Healthy Threshold	Number of consecutive failed probing after which an instance will be marked as "in-service"	2	Can be configure any value between 2- 10

Cross Zone Load Balancing: **By default**, this feature **is disable**, thus ELB by default send equal load to all the zone register to it. Once Cross Zone Load Balancing is enabled ELB will consider the number of register healthy EC2 instance available in each availability zone before routing its request to EC2 instances.

Elastic Load balancer is a region-specific service. Its ONLY operates in a single region.



There are two types of ELBs – Internet facing and internal ELBs



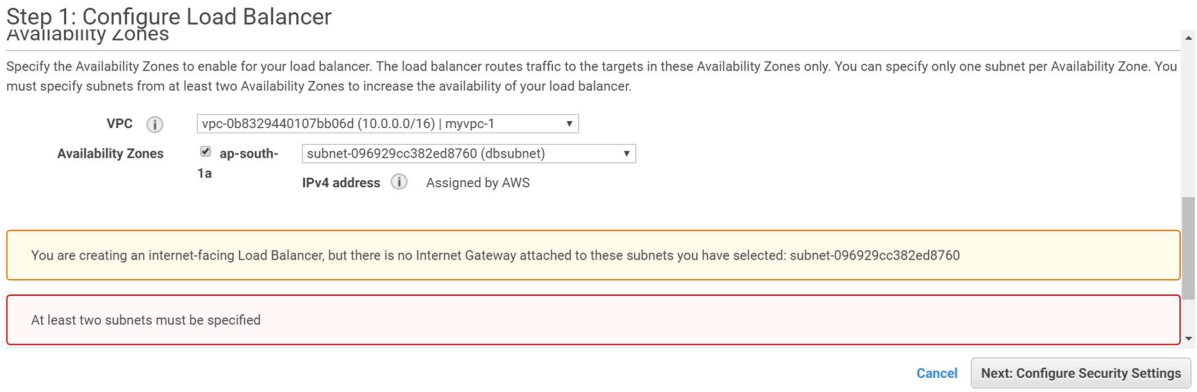
Internet facing.

Internet facing ELB are the one that are connected to the IGW and have public IP addresses. This can be reach from internet. Internet facing EBL forwards the incoming request from internet to EC2 **private IP addresses**.

Its need one public subnet in each of the availability zone to route ELB internet traffic to the EC2 instance.

Internal facing ELB.

Internal ELBs will have private IP addresses, and its route incoming VPC traffic to the IPv4 private address of the EC2 instance. If the instance has more than one IP address (or NIC) then ELP will route the internet traffic ONLY to eth0 NIC configured IP addresses.



ELB will always need to have a **security group** attached to it which allow communication from/to the client and the backend EC2 instances. Since ELB are define in a subnet its important that inbound and outbound connected to/from ELB are not blocked by NACL.

ELB-Security Group Rules (internet facing)				
Type	**Protocol**	**Range**	**Source**	**Description**
Custom TCP	TCP	80	0.0.0.0.0/0 (open to internet)	Allow all inbound transaction to EBL from internet on 80 port where ELB is configure to listen.
Custom TCP	TCP	80 (health check port)	EC2-Security Group	The reply sends back from the EC2 instance health check port needs to allowed to enter.

ELB-Security Group Rules (internal)				
Type	**Protocol**	**Range**	**Source**	**Description**
Custom TCP	TCP	80	VPC CIDER block address	Allow all inbound transaction to EBL from any VPC address on 80 port where ELB is configure to listen.
Custom TCP	TCP	80 (health check port)	EC2-Security Group	The reply sends back from the EC2 instance health check port needs to allowed to enter.

Since ELB forward the inbound request to the EC2, its not possible for the EC2 instance to know who the actual sender is. In order to know the original sender ELB support two feature.

Layer 4 (SSL & TCP protocol)	**Layer 6 (HTTP and HTTPS protocol)**
By enabling proxy protocol feature of the ELB, one can configure the ELB to carry forward the original connection details request like original user source IP, source port, etc…	HTTP header X-FORWARD will be use to store the original connection details request like original user source IP, source port, etc. In order to know the original user details backend instance should be configure to read X-FORWARD header details.

EC2-Security Group Rules (Internet/ internal)				
Type	**Protocol**	**Range**	**Source**	**Description**
Custom TCP	TCP	80	ELB-Security Group	Allow inbound transaction from ELB security group towards port 80.
Outbound Rule				
Type	**Protocol**	**Range**	**Destination**	**Description**
Custom TCP	TCP	Ephemeral-port range (1-65535)	ELB-Security Group	EC2 reply going back to the EBL. Since ELB node can be listening to any available ports - Ephemeral-port range needs to be mentioned

ELB Encryption : ELB can forward the encrypted request as is to the backend instance to achieve end-to-end encryption, also it can be configure to offload the SSL encrytion from the backend instances, in that can the SSL connection will be terminated at the ELB and message is decipher and forward to the backend server over SSL or non-SSL connection as configured.

ELB Connection Draining: When an instance is removed from the

ELB Access Logs: by default, the access logs are disable for ELB. Once they are enabled, they can be configured to send access log to user define S3 bucket which will be in the same region as that of the ELB. Storage cost will be applicable.

Sticky Session (also called as Server Affinity): By turning on the server affinity, ELB binds all the client session/request to a specific EC2 instances ONLY. IT required SSL termination on the ELB. The duration of the Sticky session is determined by the expiration defined in the cookie.

- If application can maintain its own cookies, then EBL can be configured to maintain the expiration defined in the application generated cookies.
- If application don't support cookies, ELB can be configure to maintain its own cookies. EBL will generate its own cookies and pass the same to the user along with its cookies. Till cookies don't expires ELB will continue to pass the request to same backend server which has server the request before, once the cookies expire ELB can forward the request to any of its configured EC2 instances.

Pro & Corns of using Sticky session:

Pro: Client session will be maintained; they don't need to authenticate again.

Corn: As the client session will be maintained in a single backend EC2 instances, if the backend EC2 instances fails the session information is lost when ELB forwards the request to another backed EC2 instance.

SSL Session Negotiation within ELB:

- Http on SSL = Https
- For front end SSL negotiation there are two possibilities
    - Define custom policy (SSL Protocols | SSL Ciphers | Server Order Preference)
    - Use pre-define policies
- For backend SSL negotiation there is on option available
    - Use pre-define policies
- SSL protocol supported are TSL 1.0, TSL 1.1 TSL 1.2 and SSL 3.0 (TSL1.3 and SSL 2.0 are not supported yet) Note: ACM (AWS certificate manager) uses RSA cipher for encryption, if one is planning to use ACM for X.509 certificates then RSA Cipher should be included)

- Server Order Preference will define weather server or the client will have more preference – if enable then, then the first match on the ELB cipher list with the client list will be use. For pre-define polices this property is enabled.
- Single X.509 certificate can be loaded into ELB. In there are multiple sites then create separate EBL for each of those websites and upload their corresponding certificates.
- ELB does not support client-side authentication (two-way authentication) with HTTPS, workaround for this is to configure ELB to work with TCP protocol instead of HTTPS and enable proxy protocol and let the backend instance (EC2 instance) read the proxy headers and perform all the task related to client-side authentication. *[For this one need to ensure that EC2 instance should be able to read proxy headers and have access to server certificates & ELB does not support sticky session].*
- **Connection Draining** (disable by default) – AWS will wait for the in-flight instances to complete, before AWS marking the backed instance as un-healthy. When an instance is in process of deregistering with connection drain feature turn on the instance appear as "*In Service: Instance deregistration in progress.*". NO new connection will be created by the ELB during that time. Default value is 300 sec max time 3600 sec.
- **ELB monitoring:**
    - **AWS CloudWatch**: (Enable by default) ELB service will send ELB metric every one min to CloudWatch when ELB is having request passing through it during ideal time it wouldn't send any metrics to AWS CloudWatch. AWS CloudWatch metric can be use to trigger SNS notification in case threshold is reach.
    - **Access Logs**: (Disable by default): By enabling the access logs one can capture connection details like requester IP/Port, request time, request type etc., which can be stored in S3 bucket. There is no additional cost involved for enabling access logs – for storing access log information in S3 one need to pay for the storage.
    - **CloudTrail** (Disable by default): All API call made to the ELB are sent to CloudTrail. By enabling CloudTrail, all API call made to ELB can be capture. CloudTrail log can be routed to S3 bucket, by doing so S3 storage charges will be incur.
- **ELB connection time-out (ideal connection time out):** When setting an ELB connection time out, one need to consider application session timeout. If the application session time out is less than the ELB connection time-out then the backend end instance will force disconnect the connection, this may result marking the instance as un-healthy as ELB connections are getting timeout. Best practice to match the application timeout with the ELB timeout.
- **EBL Scaling:** AWS manage ELB, it takes care of the scaling of the ELB node. ELB needs 2-7 min depending upon traffic to add ELB nodes. During the time its scaling, it will NOT queue the incoming request instead return HTTP 503 error. ELB can scale if a) traffic is increasing in 50% steps in every 5 min OR b) load is increasing linearly.

  Alternative scaling option
    - **ELB pre warming:** When anticipating a large node, one can contact ELB make them aware of the anticipated load, so that AWS can add appropriate ELB nodes in advance to cater the increase in the traffic. This is known as ELB pre warning.
- Route53 maintains the list of ELB node that are added to the ELB DNS name, every time a new node is added/remove route53 update its list which is an authoritarian DNS for the ELB endpoint. Client will query the DNS and cache it locally for future interaction. While doing load testing one need to keep it mind of DNS Resolution else irrespective of having new node added to the ELB client will send its request to the cached ELB node address.

  Best practice while doing load testing for ELB frontend services
    - Use multiple test clients for load testing ELB
    - Use global test sites
    - In case, one test client is available for testing – then testing tool should be capable of DNS Resolution to ensure all request does not ends up to a single cached ELB node.

Auto Scaling

- AWS service that enable compute to grow or shrink based on the needs is called as Auto-scaling. This will have to build a fault torrent highly available system. **Auto scaling is a regional service**, it can't work across region. They also need to be in the same VPC.
- When need to add more compute - autoscaling: SCALE OUT.
- When need to reduce compute - autoscaling: SCALE IN.
- Different components of autoscaling
    - **Launch configuration** – EC2 template which will be use when scale in is required. It includes – instance family, instance type, volumes, volume type, AMI, Keypair Block devices, security groups.
        - **Launch configuration once created cannot be edited, every time it needs to be created from the scratch.**
        - Launch configuration cane be created from AWS CLI, AWS Console or from a running EC2 instances.
        - When creating a launch configuration from EC2 instances following things need to be keep in mind:
            a) The AMI use for creating the EC2 instance should be available. (its should not be deleted or unshared.)
            b) EC2 instance tags & any block store volume added to the EC2 instance after its launch will not be part of the EC2 instance.
            c) EC2 instance should belong to the same auto scaling group.
    - **Autoscaling group** – logical grouping of the autoscaling EC2 instances. This can be change even after its created.
        - Minimum Size – minimum number of instances to start with.
        - Desired Capacity – The ideal size
        - Maximum Size – maximum number of instances beyond which it can't grow.

      In auto scaling group, one can decide which subnet to be use for the for scaling in the instances.

      If the request instance family is not available on the targeted subnet then auto-scaling will try to create the same instance in another subnet which is define in the auto scaling group.

    - **Scaling Policy (Plan)**
        - ON-Demand/Dynamic scaling
        - Cyclic/Scheduled Scaling

      When load is known/predictable cyclic(scheduled) scaling policy are uses. When, scaling is needed to react an undesired scenario then use dynamic (on-demand) policy.

    - **Autoscaling rebalancing**: Autoscaling will try to evenly distribute the instances across all its Availability zone. In order to do this, it will first create instances in the AZ where it has less instances, once those instances are alive it terminates instances from the AZ where it has more instances. (*During autoscaling rebalancing,*

autoscaling can go 10% higher than the maximum size OR 1 EC2 instance which ever its higher).

Why balancing is required:

- If autoscaling group could not launch the desired instance in AZ due to unavailability of the instance capacity.
- Autoscaling was alter at later point of time and new AZ were added.
- Manually terminated instance of one AZ.
- Manually adding instance to one AZ.
- Spot instance were terminated due to increase in the market price.

- What is the difference between standby instance AND terminated/ de-attaching instances?

  When instance is in standby state – instance will still be charge as "in-service" state however they will not be part of autoscaling health-check or will have any active workload.

  The best practice to troubleshoot an EC2 instance which is a part of autoscaling group is to mark the instance as "standby" instance, troubleshoot the instance and then move it back as "in-service" instance.

  Note : During a very limited timeframe after auto scaling marked an instance for replacement and before instance is terminated by auto scaling group, one can use AWS CLI (command line interface) to set the instance health as healthy (as-set-instance-healthy)to avoid termination.
- In case of replacement – instance is terminated first and then replaced by a new healthy instance.
- In case of re-balancing – new instance is added to the desired AZ, once its healthy instance from over populated AZ will be terminated.
- One or more ELB can be attached to attached auto scaling group. Once attached, EC2 instances will be automatically attached to the Auto Scaling group define ELBs. Based on the policy define, instances will be added or removed from the ELB. ELB will be focal point of the attached auto scaling groups EC2 instance. If connection drained is defined for the ELB, the auto scaling group will honor it.
- Auto scaling can be defined to listen to both – EC2 Health check and ELB health checks.
- If **Health Check Grace Period** is defined in the auto scaling, auto scaling group will wait till the grace period is over.
- Auto scaling group for spot instances: *[one can't mix and match spot instance and on demand instances].*
- For spot instance, in order to change the bid price of a spot instance, new to create a new Launch Configuration OLD launch configuration can't be use.
- If auto scaling group fails to launch an EC2 spot instance in an AZ due to the market price more than that of the bid price, then it will create the spot instance in an AZ where market price is less than that of define bid price in the launch configuration. Once the targeted AZ price drops below bid price, auto scaling group will first create an instance on the targeted AZ and then rebalance the instances with the other associated AZ.
- SNS notification can be send to Emails auto scaling group IF
  - Instance is launched successfully
  - Instance is terminated successfully
  - Instance failed to launch

- Instance failed to terminate

Autoscaling group uses cloudWatch to send SNS notification under above mention events.

- Auto scaling group can be merged using CLI, in order to merge an auto scaling group to another – define the auto scaling in the other auto scaling group AZ (rezone the auto scaling group to span another AZ where the second auto scaling group is connected). Then delete the second auto scaling group.
- Autoscaling policy
  - Scale-Out
  - Scale-In
- There are three things to auto scaling policies

Minimum capacity	Desired capacity	Maximum capacity
Minimum capacity.	Idea capacity	Maximum allowed instance at any given point in time.

- Scaling Policy
  - **Manual Scaling -** Manually adding/removing EC2 instances to maintain a desired count of EC2 instances. This can be done by manually changing auto scaling group min/desired/max values, or by attaching/detaching instances manually.
  - **Cyclic (Schedule base) scaling [for predictable load change]** – When the load is predictable one can schedule to add more instances to cater the growing/shrinking load.
    a) Each of the schedule scaling policy should have unique date-time (unique start & end date). For example: Before black Friday sale.
  - **On-Demand (Event base scaling) [for unpredictable load change]**– when EC2 instances needs to be added base on an event (dynamic load) then use On-Demand policies.
    a) Alarm can be created on a single metrics. Cannot be created based on multiple metrics.
    b) Cool-down-timer (default value 300): A time gap after acting on a scaling activity to give sufficient time for the EC2 instances to be provision and come to in-service state.
    c) There are two types of on-demand scaling
      - *Simple Scaling* - adding capacity in single shot
      - *Step Scaling* – adding capacity in multiple steps. Incase of step scaling cool down period will not be applicable, instead warm up time is applicable: After the scaling action is trigger time gap before proccing a new alert.
  - **Single auto scaling group can have multiple scaling policies attached to it at a given time.**
  - **Auto scaling cannot change the capacity of the group above the maximum capacity OR below the minimum capacity.**
  - Default Scaling policy is **target scaling policy**: Base on one of the below metrics auto scaling will occur. Once set target is reached auto scaling policy will kick in.
    a) Application Load Balancer Request count per target

b) Average CPU utilization
c) Average Network in (bytes)
d) Average Network out (bytes)

One can disable scale-in feature to prevent instances from auto termination. Also, one can set warm up time, till warm up time counter is not ZERO another policy will not be triggered.

- Auto scaling monitoring: As a part of the launch configuration, one can define EC2 instance monitoring. If the Launch configuration is created from
    a) AWS console then by default basic monitoring is activated (which collects & send metrics to CloudWatch every 5 min).
    b) AWS CLI then by default detailed monitoring is activated (which collects & send metrics to CloudWatch every 1 min).
- Apart from collecting & monitoring EC2 metrics, auto scaling group can also send aggregated metrics of the auto scaling group (which also includes EC2 instances metrics) to CloudWatch. By default, auto scaling group aggregated metrics will be send every one min.
- To changing monitoring option in auto scaling group, a new launch configuration with the desired monitoring option needs to be recreated (as Launch Configuration can be edited, can only be copied or create fresh).
- The launch configuration monitoring setting AND the auto scaling group policy setting should match – if its basic monitoring then both needs to be basic monitoring/ if its detailed monitoring then both needs to be detailed monitoring.
    a) If EC2 instance monitoring set to basic in Launch configuration – set auto scaling alarm to 300 sec.
    b) If EC2 instance monitoring set to detailed in Launch configuration – set auto scaling alarm to 60 sec.

As such there will not be any error, however if EC2 instance is set as detailed monitoring and auto scaling is set as basic monitoring then 4 out of 5 EC2 metrics reading will not be used.