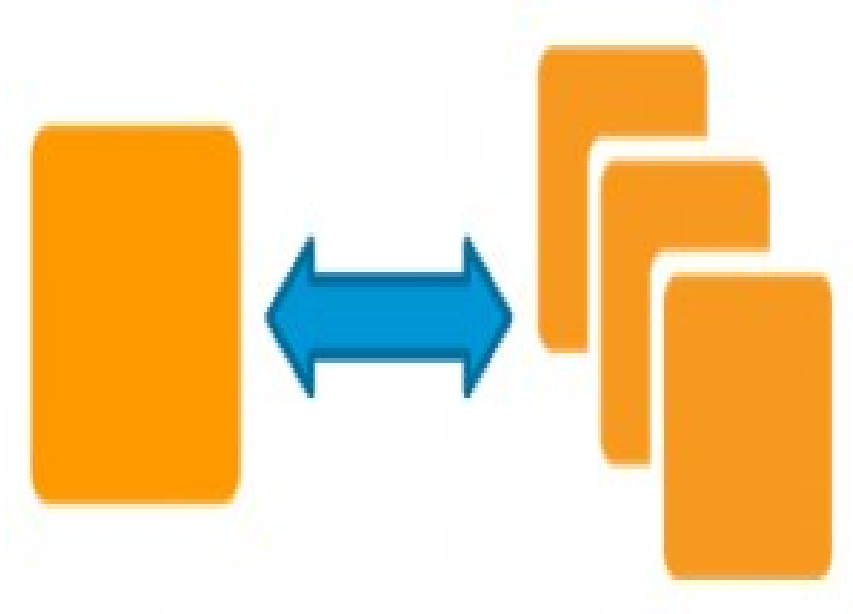
Elastic Compute Cloud (EC2)

What is EC2

Amazon Elastic Compute Cloud, EC2 is a web service from Amazon that provides **re-sizable** compute services in the cloud.

How are they re-sizable

- They are re-sizable because you can quickly **scale up** or **scale down** the number of server instances you are using if your computing requirements change.



What is EC2 ?

- Amazon EC2 is a web service that provides resizable compute capacity in the cloud.

- Amazon Ec2 reduces the time required to obtain and boot new server instances to minutes, allowing you to quickly scale capacity both up and down as your computing requirements change.

Instances

- An instance is a virtual server for running applications on Amazon's EC2. It can also be understood like a tiny part of a larger computer, a tiny part which has its own Hard drive, network connection, OS etc.

- But it is actually all virtual. You can have multiple "tiny" computers on a single physical machine, and all these tiny machines are called Instances.

EC2 Pricing Models

- **On Demand Instances**

  Allows you to pay a fixed rate by the hour with no commitment.

- **Reserved Instances**

  Provides you with the capacity reservation, and offer a significant discount on the hourly charge for an instance. Contract terms 1 or 3 years

- **Spot Instances**

  Enables you to bid whatever price you want for instance capacity, providing for even greater saving if your applications have flexible start and end times

- **Dedicated Hosts**

Physical EC2 server dedicated for your use. Dedicated host can help you to reduce costs by allowing you to use your existing server-bound software licenses.

EC2 Pricing Models – On Demand

On Demand Pricing is used for

- Users that want low cost and flexibility of Amazon EC2 without any up-front payment or long term commitment.
- Applications with short term , spiky or unpredictable work loads that cannot be interrupted.
- Applications being developed or testes on Amazon EC2 for the first time.

EC2 Pricing Models – Reserved

Reserved Pricing is used for

- Application with Steady state or predictable usage
- Applications that required reserved capacity
- Users able to make upfront payments to reduce their total computing cost even further

Reserved pricing types

Standard Reserved Instances: These offers up to 75% off on demand instances. The more you pay upfront and the longer contract, the greater the discount.

Convertible Reserved instances: These offers up to 54% off on demand capability to change the attributes of RI as long as the exchange results in the creation of RI of equal or greater value.

Scheduled Reserved instances: These are available to launch with in the time window you reserve. This option allows you to match your capacity reservation to a predictable recurring schedule that on requires **fraction of a day , week or a month**

EC2 Pricing Models – Spot Instances

Spot instances Pricing is used for

- Application that has flexible start and end times
- Application that are only feasible at very low compute prices
- Users with urgent compute needs for large amount of additional capacity

EC2 Pricing Models – Dedicated

Dedicated instances Pricing is used for

- Useful for regulatory requirements that may not support multi-tenant virtualization
- Great for licensing which does not support multi-tenancy or cloud deployments.
- Can be purchase on demand hourly.
- Can be purchase as a reservation for up to 70% of the on demand price

Let’s understand the types of EC2 Computing Instances:

Computing is a very broad term, the nature of your task decides what kind of computing you need

Therefore, AWS EC2 offers many types of instances which are few as follows:

- ***General Instances***
  - For applications that require a balance of performance and cost.
    - E.g email responding systems, where you need a prompt response as well as the it should be cost effective, since it doesn’t require much processing.
- ***Compute Instances***
  - For applications that require a lot of processing from the CPU.
    - E.g analysis of data from a stream of data, like Twitter stream
- ***Memory Instances***
  - For applications that are heavy in nature, therefore, require a lot of RAM.
    - E.g when your system needs a lot of applications running in the background i.e multitasking.
- ***Storage Instances***
  - For applications that are huge in size or have a data set that occupies a lot of space.
    - E.g When your application is of huge size.
- ***GPU Instances***
  - For applications that require some heavy graphics rendering.
    - E.g 3D modelling etc.

Instance Types

- **Now, every instance type has a set of instances which are optimized for different workloads:**
- General Instances
  - t2
  - m4
  - m3
- Compute Instances
  - c4
  - c3
- Memory Instances
  - r3
  - x1
- Storage Instances
  - i2
  - d2
- GPU Instances
    - G2 E.g 3D modelling etc.

Burstable Performance Instances

- *T2 and T3 instances* are burstable instances, meaning the CPU performs at a baseline, say 20% of its capability.

- When your application needs more than 20% of the performance of the CPU, the CPU enters into a burst mode giving higher performance for a limited amount of time, therefore work happens faster.

Elastic Block Storage (EBS)

- *Amazon* Elastic Block Storage provides persistent block storage volumes for use with Amazon EC2 instances in the AWS Cloud.

- Each Amazon EBS volume is automatically replicated within its availability zone to protect you from component failure, offering high availability and durability.

Types- Elastic Block Storage (EBS)

- General Purpose (gp2, SSD)
- Provisioned IOPS (io1, SSD)
- Throughput Optimized Hard Disk Drive (st1) (Low Cost, frequently accessed)
- Cold Hard Disk Drive (sc1) (Low cost, less frequently accessed)
- Magnetic (standard) (Previous generation)

Types- Elastic Block Storage (EBS)

Solid-State Drives (SSD)			Hard disk Drives (HDD)		
Volume Type	**General Purpose SSD**	**Provisioned IOPS SSD**	**Throughput Optimized HDD**	**Cold HDD**	**EBS Magnetic**
Description	General purpose SSD volume that balances price and performance for a wide variety of transactional workloads	Highest-performance SSD volume designed for mission-critical applications	Low cost HDD volume designed for frequently accessed, throughput-intensive workloads	Lowest cost HDD volume designed for less frequently accessed workloads	Previous generation HDD
Use Cases	**Most Work Loads**	**Databases**	**Big Data & Data Wharehouses**	**File Servers**	**Workloads where data is infrequently accessed**
API Name	gp2	io1	st1	sc1	Standard
Volume Size	1 GiB - 16 TiB	4 GiB - 16 TiB	500 GiB - 16 TiB	500 GiB - 16 TiB	1 GiB-1 TiB
Max. IOPS**/ Volume	16,000	64,000	500	250	40-200

Types- Elastic Block Storage (EBS)

	Solid-State Drives (SSD)		Hard Disk Drives (HDD)	
Volume Type	General Purpose SSD (gp2)*	Provisioned IOPS SSD (io1)	Throughput Optimized HDD (st1)	Cold HDD (sc1)
Description	General purpose SSD volume that balances price and performance for a wide variety of workloads	Highest-performance SSD volume for mission-critical low-latency or high-throughput workloads	Low-cost HDD volume designed for frequently accessed, throughput-intensive workloads	Lowest cost HDD volume designed for less frequently accessed workloads
Use Cases	• Recommended for most workloads • System boot volumes • Virtual desktops • Low-latency interactive apps • Development and test environments	• Critical business applications that require sustained IOPS performance, or more than 16,000 IOPS or 250 MiB/s of throughput per volume • Large database workloads, such as: ◦ MongoDB ◦ Cassandra ◦ Microsoft SQL Server ◦ MySQL ◦ PostgreSQL ◦ Oracle	• Streaming workloads requiring consistent, fast throughput at a low price • Big data • Data warehouses • Log processing • Cannot be a boot volume	• Throughput-oriented storage for large volumes of data that is infrequently accessed • Scenarios where the lowest storage cost is important • Cannot be a boot volume
API Name	gp2	io1	st1	sc1
Volume Size	1 GiB - 16 TiB	4 GiB - 16 TiB	500 GiB - 16 TiB	500 GiB - 16 TiB
Max. IOPS/Volume**	16,000***	64,000****	500	250
Max. Throughput/Volume	250 MiB/s***	1,000 MiB/s†	500 MiB/s	250 MiB/s
Max. IOPS/Instance	80,000	80,000	80,000	80,000
Max. Throughput/Instance††	1,750 MiB/s	1,750 MiB/s	1,750 MiB/s	1,750 MiB/s
Dominant Performance Attribute	IOPS	IOPS	MiB/s	MiB/s

EBS-optimized Instances

- *C4, M4, and D2 instances*, are EBS optimized by default, EBS means Elastic Block Storage, which is a storage option provided by AWS in which the IOPS* rate is quite high.

- Therefore, when an EBS volume is attached to an optimized instance, single digit millisecond latencies can be achieved.

- *IOPS (Input/Output Operations Per Second, pronounced eye-ops) is a performance measurement used to characterize computer storage devices.

Instance Store Volumes

An **instance store** is a temporary **storage** type located on disks that are physically attached to a host machine.

Once Instance is rebooted your data is not lost.

If you stop and terminate, then data is lost.

Snapshots

- An **EBS snapshot** is a point-in-time copy of your Amazon **EBS** volume.
- Which is lazily copied to Amazon Simple Storage Service (Amazon S3).
- **EBS snapshots** are incremental copies of data. This means that only unique blocks of **EBS** volume data that have changed since the last **EBS snapshot** are stored in the next **EBS snapshot**

EBS vs Instance Store Volumes

All AMI are categorized as either backed by Amazon EBS or backed by instance store.

- **For EBS Volumes**: The root device for an instance launched from the AMI is an Amazon EBS volume created from an Amazon EBS snapshot.

- **For Instance Store Volumes:** The root device for an instance launched from the AMI is an instance store volume created from a template stored in Amazon S3.

Cluster Networking Instances

- *X1, M4, C4, C3, I2, G2 and D2 instances* support cluster networking. Instances launched into a common placement group are put in a logical group that provides high-bandwidth, low latency between all the instances in the group.

- A placement group is basically a logical cluster where some select EC2 instances which are a part of that group can utilize up to 10Gbps for single flow and 20Gbps for multi flow traffic in each direction.

- Instances which are not a part of that group are limited to 5 Gbps speed in multi flow traffic. Cluster Networking is ideal for high performance analytics system.

Cluster Placement Groups

Three Types of Placement Groups

Clustered Placement Group

- A Cluster placement group is a grouping of instances with in a **single availability zone.** Placement groups are recommended for applications that need low network latency, high network throughput or both.
- Only **certain instances types** can be launched in to a Clustered Placement Group.

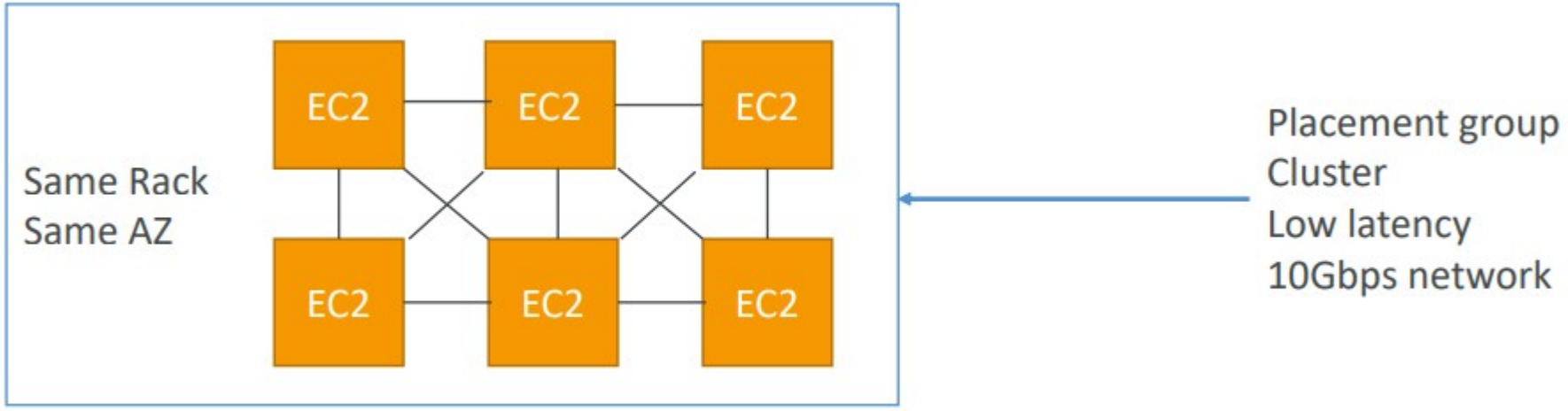
Spread Placement Group

- A spread placement group is a group of instances that are each placed on distinct underlying hardware.(Max 7 instances per group per AZ)
- Spread placement group are recommended for applications that have a small number of critical instances that should be kept separate from each other.

Partition Placement Group

—spreads instances across many different partitions (which rely on different sets of racks) within an AZ. Scales to 100s of EC2 instances per group (Hadoop, Cassandra, Kafka)

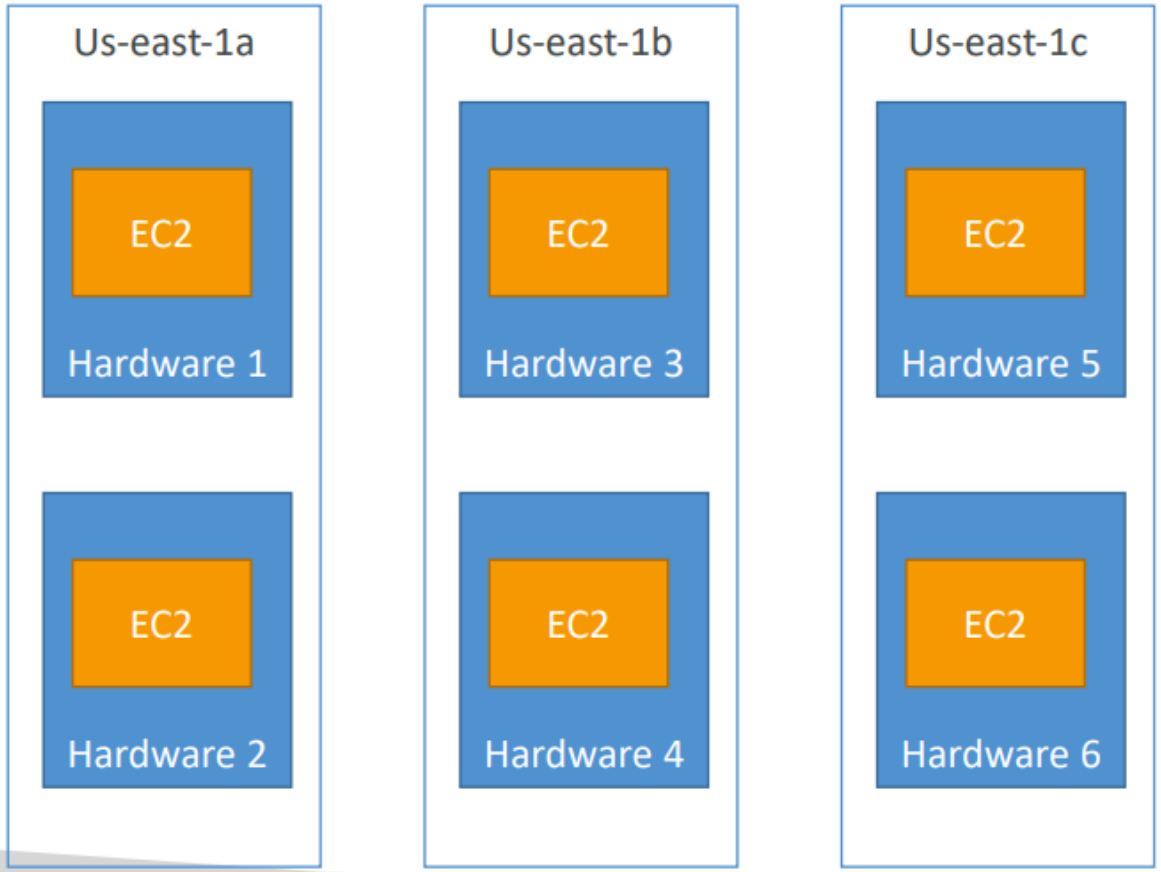
Cluster Placement Groups



Pros: Great network (10 Gbps bandwidth between instances)

Cons: If the rack fails, all instances fails at the same time

Spread Placement Groups



Pros:

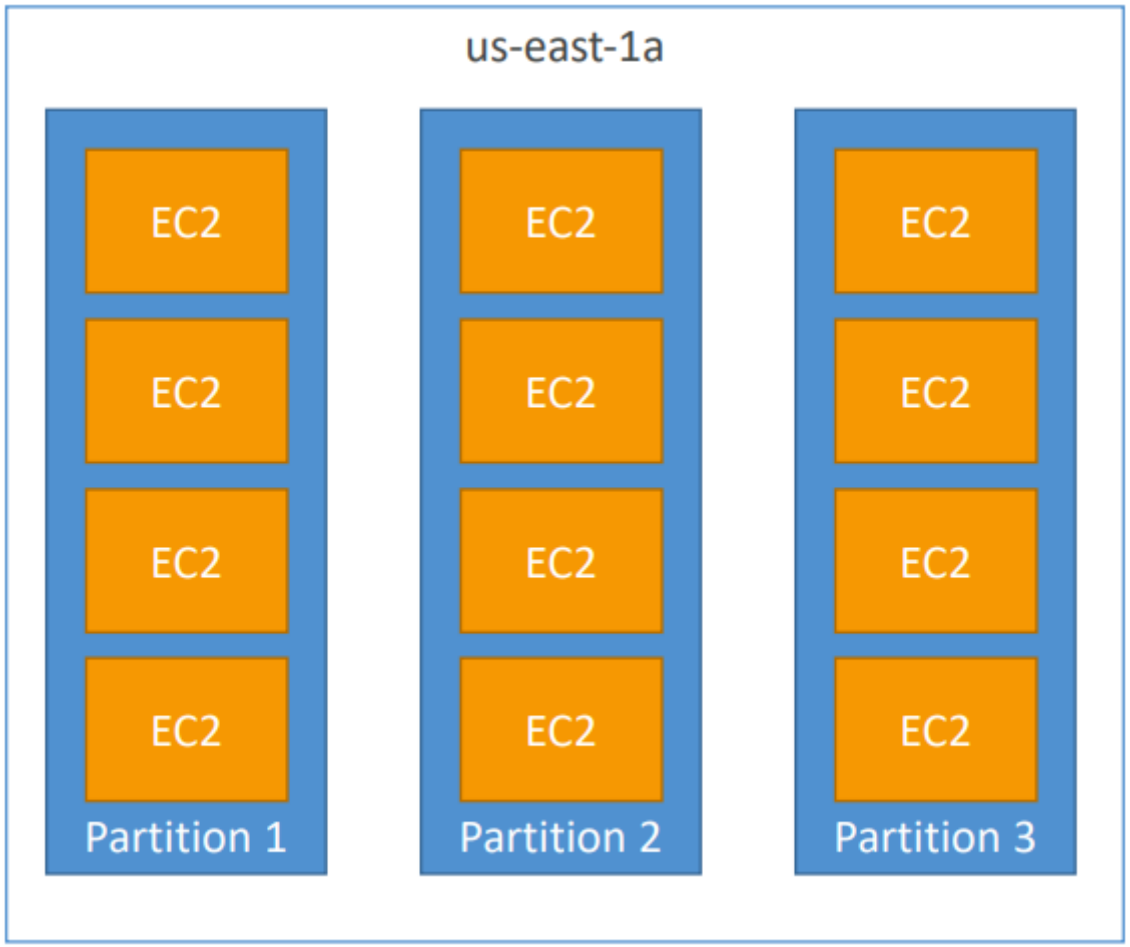
- Can span across Availability Zones (AZ)
- Reduced risk is simultaneous failure
- EC2 Instances are on different physical hardware

Cons:

- Limited to 7 instances per AZ per placement group

- **Use case:**
- Application that needs to maximize high availability
- Critical Applications where each instance must be isolated from failure from each other

Partition Placement Groups



- Up to 7 partitions per AZ
- Up to 100s of EC2 instances
- The instances in a partition do not share racks with the instances in the other partitions
- A partition failure can affect many EC2 but won't affect other partitions

- Use cases: HDFS, HBase, Cassandra, Kafka

Amazons Machine Image (AMI)

- AMIs are templates of OS and they provide the information needed to launch an instance.

You can select your AMI based on

- Region
- OS
- Architecture (32 or 64 bit)
- Storage for the Root device
  - Instance Store(Ephemeral Storage)
  - EBS backed volume

Amazons Machine Image (AMI)

- As we saw, AWS comes with base images such as:

Ubuntu, Fedora, RedHat, Windows,Etc…

- AMIs can be built for Linux or Windows machines
- Custom AMI?
- Pre-installed packages needed
- AMI are built for a specific AWS region (!)

Public AMIs

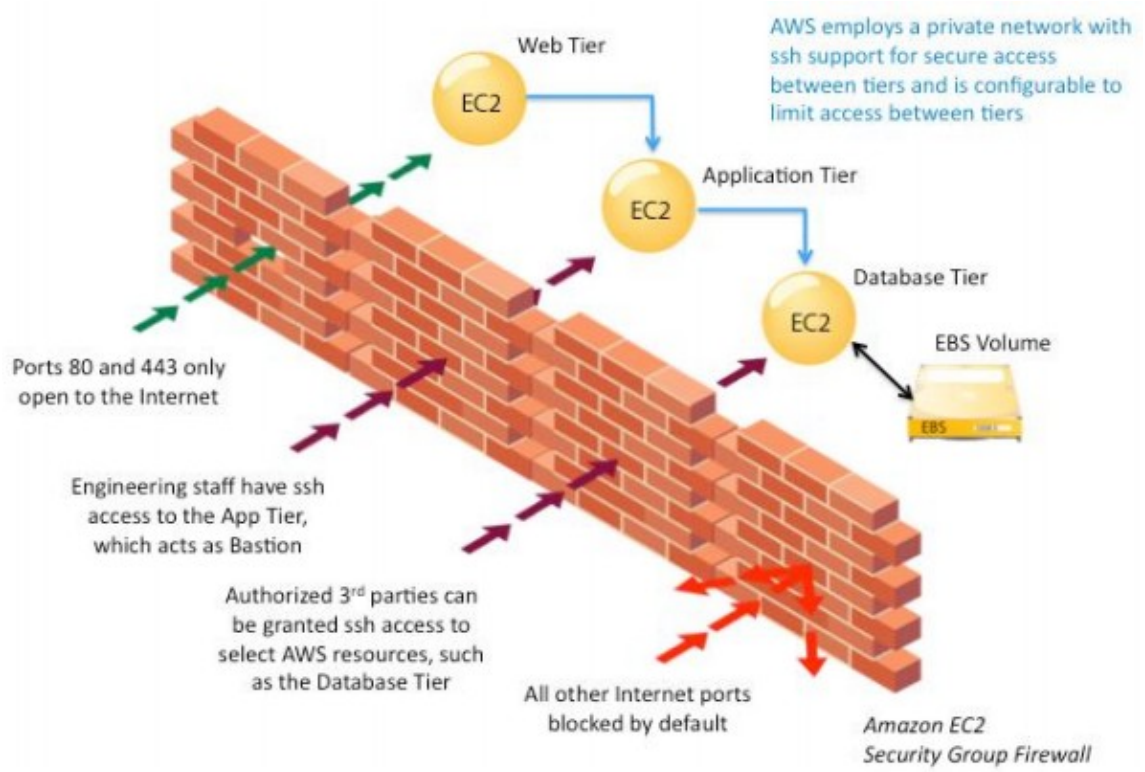
- You can leverage AMIs from other people
- You can also pay for other people's AMI by the hour
- These people have optimised the software
- The machine is easy to run and configure
- You basically rent "expertise" from the AMI creator
- AMI can be found and published on the Amazon Marketplace

AMIs Storage

- Your AMI take space and they live in Amazon S3
- Amazon S3 is a durable, cheap and resilient storage where most of your backups will live (but you won't see them in the S3 console)
- By default, your AMIs are private, and locked for your account / region
- You can also make your AMIs public and share them with other AWS accounts or sell them on the AMI Marketplace

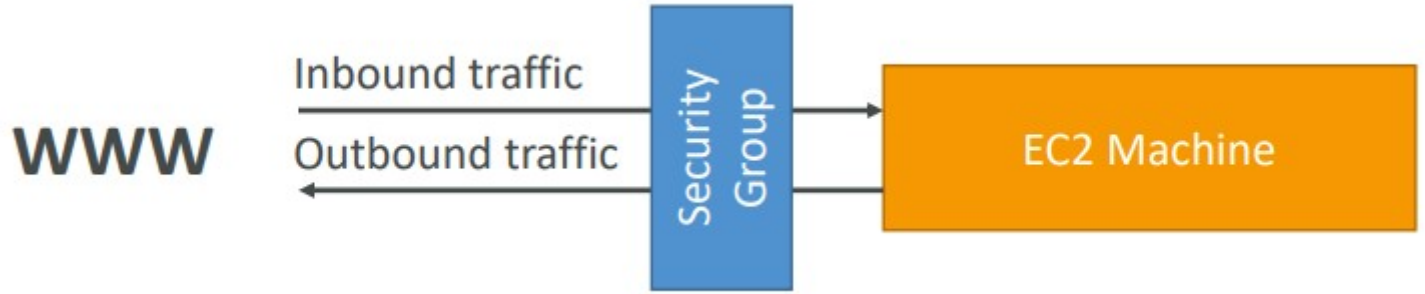
Security Groups

- A security group acts as a firewall to control inbound and outbound traffic.
- Each security group has rules according to which the traffic is governed.



Security Groups

- Security Groups are the fundamental of network security in AWS
- They control how traffic is allowed into or out of our EC2 Machines



Type	Protocol	Port Range	Source	Description
HTTP	TCP	80	0.0.0.0/0	test http page
SSH	TCP	22	122.149.196.85/32	
Custom TCP Rule	TCP	4567	0.0.0.0/0	java app

Security Groups

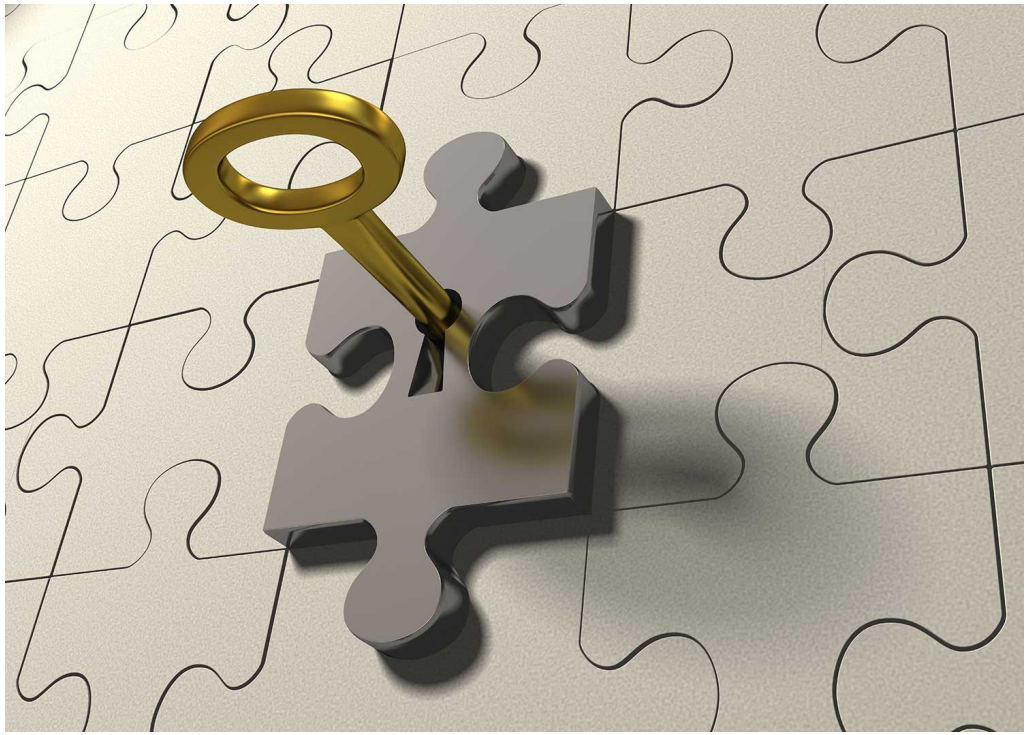
- Can be attached to multiple instances
- Locked down to a region / VPC combination
- Does live “outside” the EC2 – if traffic is blocked the EC2 instance won’t see it
- It’s good to maintain one separate security group for SSH access
- If your application is not accessible (time out), then it’s a security group issue
- If your application gives a “connection refused“ error, then it’s an application error or it’s not launched
- All inbound traffic is blocked by default
- All outbound traffic is allowed by default

Exam Tips – Security Group

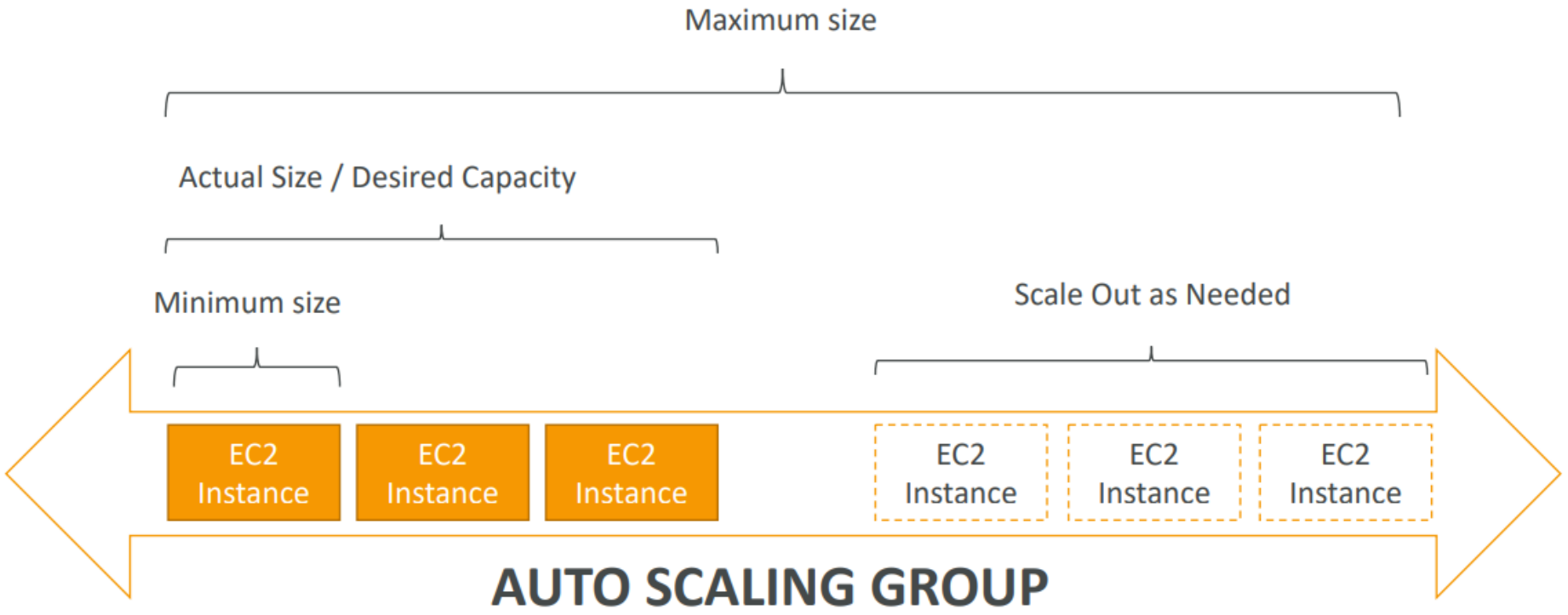
- **All in** bound traffic is blocked by default
- All outbound traffic is allowed
- Changes to the security Groups take effect immediately
- You can have any number of EC2 instances within a security group
- You can have multiple security groups attached to the EC2 instances
- Security Groups are STATEFULL
- If you create an inbound rule to allow traffic in, that traffic is automatically allowed back again that is state full
- You cannot block specific IP addresses using security groups, instead use NACL
- You can specify allow rules not deny rules
- NACL are STATELESS

Key Pair

- Amazon EC2 uses public–key cryptography to encrypt and decrypt login information. Public–key cryptography uses a public key to encrypt a piece of data, such as a password, then the recipient uses the private key to decrypt the data. The public and private keys are known as a *key pair*.



AutoScaling



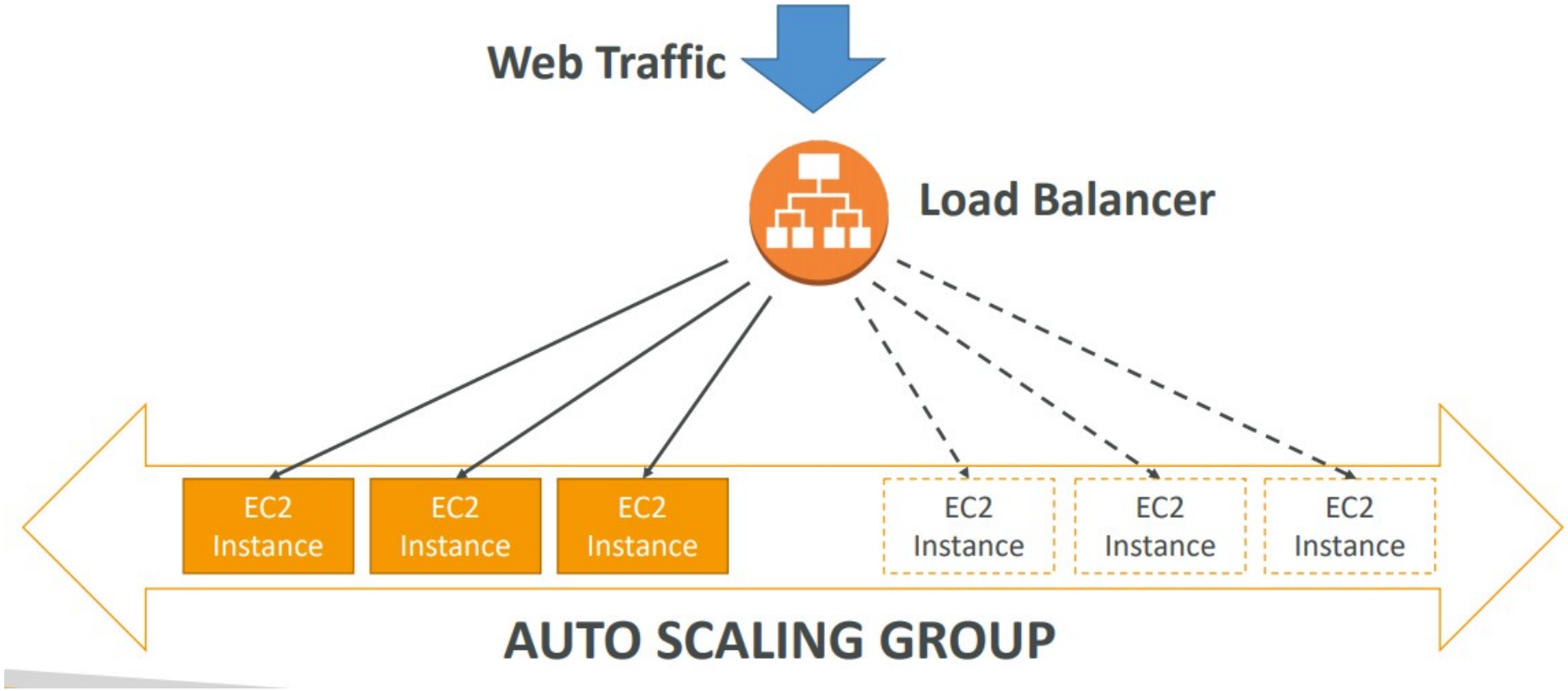
AutoScaling

- In real-life, the load on your websites and application can change
- • In the cloud, you can create and get rid of servers very quickly

The goal of an Auto Scaling Group (ASG) is to:

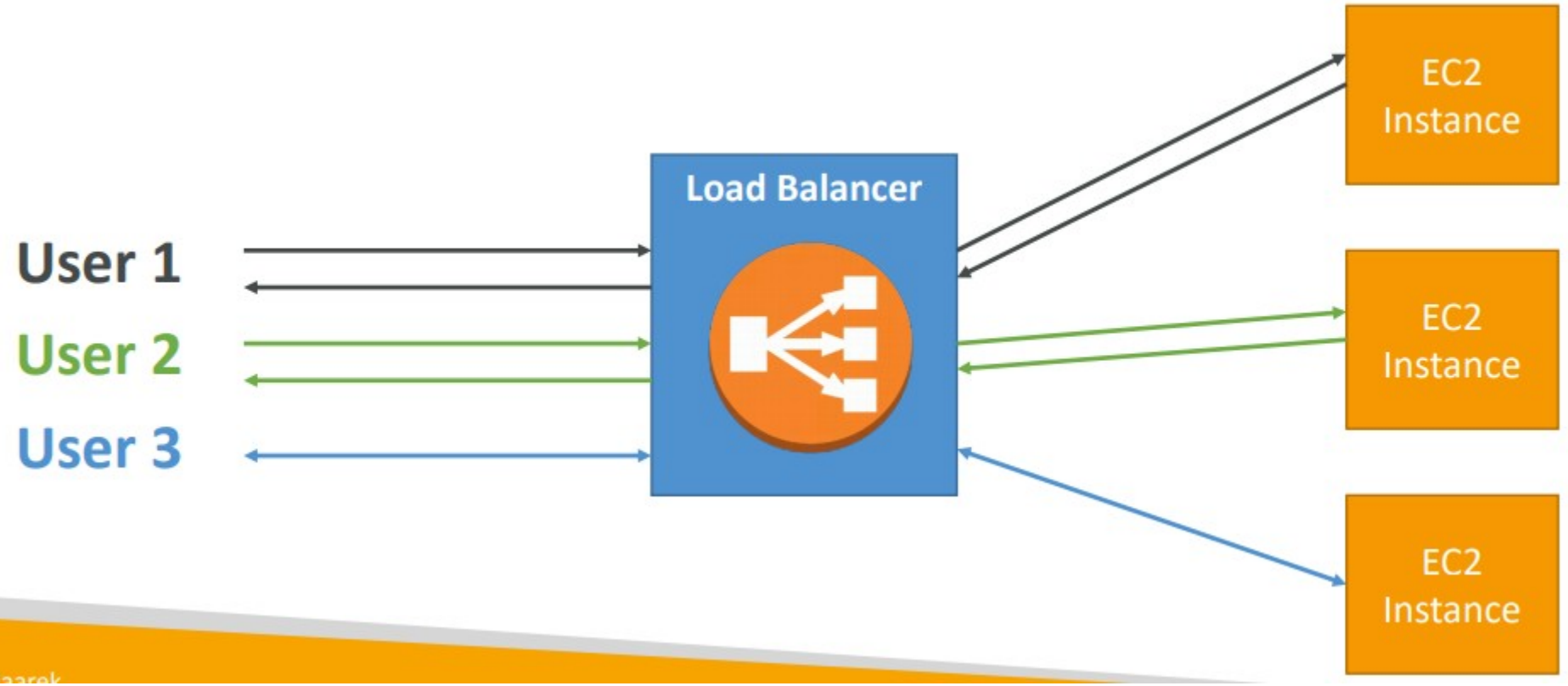
- • Scale out (add EC2 instances) to match an increased load
- • Scale in (remove EC2 instances) to match a decreased load
- • Ensure we have a minimum and a maximum number of machines running
- • Automatically Register new instances to a load balancer

AutoScaling



Elastic Load Balancer

Load balancers are servers that forward internet traffic to multiple servers (EC2 Instances) downstream.



Why we need Elastic Load Balancer

- An ELB (EC2 Load Balancer) is a managed load balancer
- AWS guarantees that it will be working
- AWS takes care of upgrades, maintenance, high availability
- AWS provides only a few configuration knobs

Types of Load Balancer

AWS has 3 kinds of Load Balancers

- Classic Load Balancer (v1 - old generation) - 2009
- Application Load Balancer (v2 - new generation) – 2016
- Network Load Balancer (v2 - new generation) - 2017

Overall, it is recommended to use the newer / v2 generation load balancers as they provide more features

You can setup internal (private) or external (public) ELBs

Application Load Balancer

Application load balancers (Layer 7) allow to do:

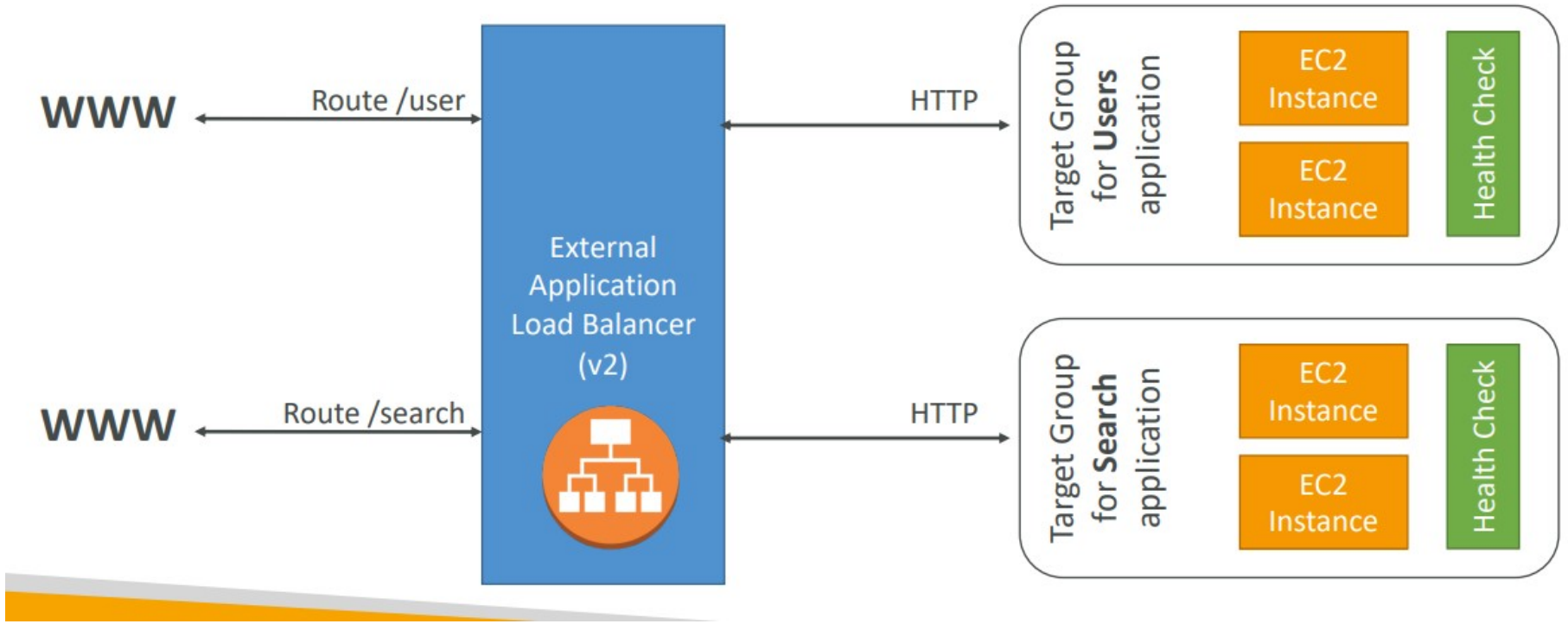
- Load balancing to multiple HTTP applications across machines (target groups)
- Load balancing to multiple applications on the same machine (ex: containers)
- Load balancing has path based and host based in URL
- Support routing based on hostname (users.example.com & payments.example.com)
- Support routing based on path (example.com/users & example.com/payments)

- Basically, they're awesome for micro services & container-based application (example: Docker & Amazon ECS)

- In comparison, we would need to create one Classic Load Balancer per application before. That was very expensive and inefficient.

Application Load Balancer works on HTTP and HTTPS

Application Load Balancer



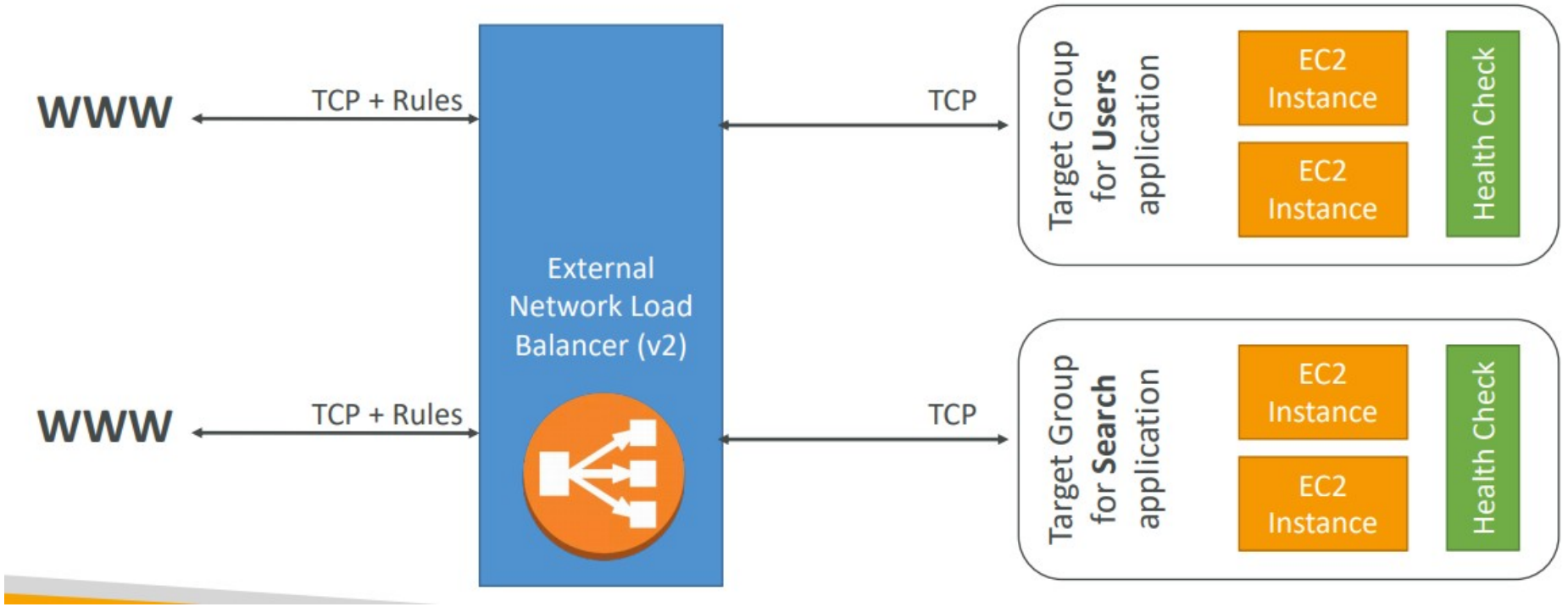
Network Load Balancer

Network load balancers (Layer 4) allow to do:

- Forward TCP traffic to your instances
- Handle millions of request per seconds
- Support for static IP or elastic IP
- Less latency ~100 ms (vs 400 ms for ALB)

Network Load Balancers are mostly used for extreme performance and should not be the default load balancer you choose

Network Load Balancer



Classic Load Balancer

Classic Load Balancer works on HTTP , HTTP and TCP but no routes(path and host based)

Why we need Elastic Load Balancer

***Elastic Load Balancer* (ELB)**

- Spread load across multiple downstream instances
- Expose a single point of access (DNS) to your application
- Seamlessly handle failures of downstream instances
- Do regular health checks to your instances
- Provide SSL termination (HTTPS) for your websites
- Enforce stickiness with cookies
- High availability across zones
- Separate public traffic from private traffic

Load Balancer – Health Checks

Health Checks are crucial for Load Balancers

- They enable the load balancer to know if instances it forwards traffic to are available to reply to requests

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

- If the response is not 200 (OK), then the instance is unhealthy

Load Balancers

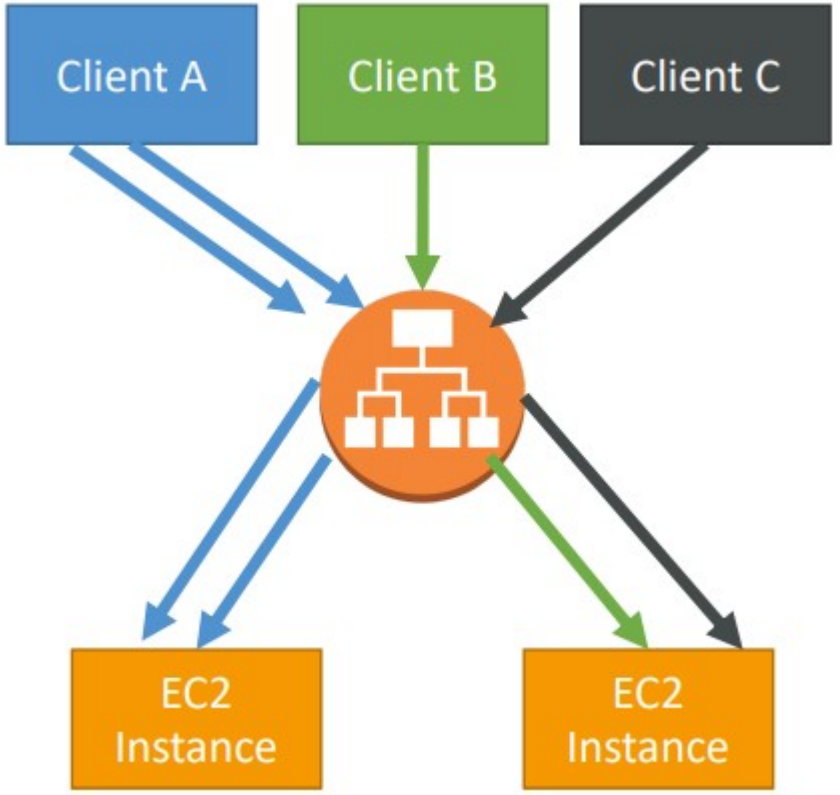
- Classic Load Balancers are Deprecated
- Application Load Balancers for HTTP / HTTPs
- Network Load Balancer for TCP
- CLB, ALB & NLB support SSL certificates and provide SSL termination
- All Load Balancers have health check capability
- ALB can route on based on hostname / path
- ALB is a great fit with ECS (Docker)

Any Load Balancer (CLB, ALB, NLB) has a static host name. Do not resolve and use underlying IP

Load Balancers Stickiness

It is possible to implement stickiness so that the same client is always redirected to the same instance behind a load balancer

- This works for Classic Load Balancers & Application Load Balancers



IP Addresses

Elastic IP address
Public IP address
Private IP Address

Elastic IP Addresses are static IP addresses which are associated with your AWS account, they can be used to mask the failure of an instance by automatically remapping your address to another working instance in your account

Instance metadata

Use to get information about the instance

- Curl http://169.254.169.254/latest/meta-data
- Curl http://169.254.169.254/latest/user-data

Let us Fire the EC2 instance now & Exam tip next, do it after demo

Exam Tips - EBS

- Volume exists on EBS. Think of EBS is an virtual hard disk.
- Snapshots exists on S3. Think snapshots as a photograph of the disk.
- Snapshots are point in time copies of volumes
- Snapshots are incremental - this means that only the blocks that have changed since your last snapshot are moved to S3.
- To create a snapshot for Amazon EBS volumes that serve as root devices, you should stop the instance before taking the snapshot.
- However you can take a snap while the instance is running
- You can create AMI from both volumes and snapshots
- You can change EBS volumes size on fly, including changing the size and storage type.
- Volumes will be the same availability zone as the EC2 instance

Exam Tips (EBS vs Instance Store)

- Instance store volumes are sometimes called as EMPHEMERAL STORAGE.
- Instance store volumes cannot be stopped. If the underlying host fails, you will loose your data.
- EBS backed instances can be stopped. You will not loose the data on the instance if it is stopped.
- You can reboot both, you will not loose data.
- By default both root volumes will be deleted on termination. However with EBS volumes you can tell AWS to keep the root volume device.

Exam Tips: Cluster Placement Groups

- A Cluster placement group cant span multiple Availability zones
- A spread placement group can span multiple AZs.
- The name you specify for a placement group must be unique within in your AWS account
- Only certain types of instances can be launched in a placement group(Compute Optimized, GPU, Memory Optimized, Storage Optimized).
- AWS recommended homogenous instances with in placement groups.
- You cant merge placement groups
- You cant move an existing instance into a placement group. You can create an AMI from your existing instance, and then launch a new instance from the AMI into a placement group.

Exam Tips - EBS

- To move an **EC2 volume from one AZ to another**, take a snapshot of it , create and AMI from the snapshot and then use the AMI to launch the EC2 instance in a new AZ.

- To move an **EC2 volume from one region to another** , take a snapshot of it, create an AMI from the snapshot and then copy the AMI from one region to the other. Then use the copied AMI to launch the new EC2 instance in the new region.

Exam Tips

- **Ter**mination Protection is turned off by default, you must turn it on
- On an EBS-backed instance, the default action is for the root EBS volumes to be deleted when the instance is terminated
- EBS root volumes of your default AMI cannot be encrypted. You can use third party tool such as bit locker to encrypt the root volume, or this can be done when creating AMIs in the AWS console or API
- Additional volumes can be encrypted

Encryption – Exam Tips

- By default, root volumes are not encrypted, but you can encrypt it. Additional volumes can be encrypted.
- Snapshots of encrypted volumes are encrypted automatically.
- Volumes restored from encrypted snapshots are encrypted automatically.
- You can share snapshots, but only if they are unencrypted.
- These snapshots can be shared with other AWS accounts or made public

How to make root volume encrypted.

- Create a snapshot of unencrypted root device volume
- Create a copy of the snapshot and select the encryption option
- Create an AMI from the encrypted snapshot
- Use the AMI to launch new encrypted instances