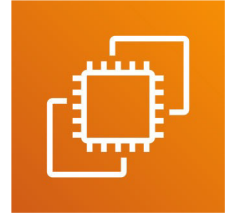


EC2

Section

Amazon EC2



- EC2 is one of the most popular of AWS' offerings
- EC2 = Elastic Compute Cloud = Infrastructure as a Service
- It mainly consists in the capability of :
 - Renting virtual machines (EC2)
 - Storing data on virtual drives (EBS)
 - Distributing load across machines (ELB)
 - Scaling the services using an auto-scaling group (ASG)
- Knowing EC2 is fundamental to understand how the Cloud works

EC2 sizing & configuration options

- Operating System (OS): Linux, Windows or Mac OS
- How much compute power & cores (CPU)
- How much random-access memory (RAM)
- How much storage space:
 - Network-attached (EBS & EFS)
 - hardware (EC2 Instance Store)
- Network card: speed of the card, Public IP address
- Firewall rules: security group
- Bootstrap script (configure at first launch): EC2 User Data

EC2 User Data

- It is possible to bootstrap our instances using an EC2 User data script.
- bootstrapping means launching commands when a machine starts
- That script is only run once at the instance first start
- EC2 user data is used to automate boot tasks such as:
 - Installing updates
 - Installing software
 - Downloading common files from the internet
 - Anything you can think of
- The EC2 User Data Script runs with the root user

EC2 Instance Types - Overview

- You can use different types of EC2 instances that are optimised for different use cases (<https://aws.amazon.com/ec2/instance-types/>)
- AWS has the following naming convention:

m5.2xlarge

- **m**: instance class
- **5**: generation (AWS improves them over time)
- **2xlarge**: size within the instance class

General Purpose

Compute Optimized

Memory Optimized

Accelerated Computing

Storage Optimized

Instance Features

Measuring Instance
Performance

EC2 Instance Types – General Purpose

- Great for a diversity of workloads such as web servers or code repositories
- Balance between:
 - Compute
 - Memory
 - Networking
- In the course, we will be using the t2.micro which is a General Purpose EC2 instance

General Purpose

General purpose instances provide a balance of compute, memory and networking resources, and can be used for a variety of diverse workloads. These instances are ideal for applications that use these resources in equal proportions such as web servers and code repositories.

| | | | | | | | | | | | |
|-----|-----|----|-----|----|-----|----|-----|-----|------|----|----|
| Mac | T4g | T3 | T3a | T2 | M6g | M5 | M5a | M5n | M5zn | M4 | A1 |
|-----|-----|----|-----|----|-----|----|-----|-----|------|----|----|

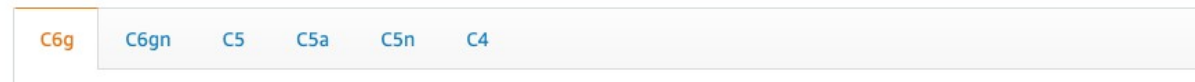
** this list will evolve over time, please check the AWS website for the latest information*

EC2 Instance Types – Compute Optimized

- Great for compute-intensive tasks that require high performance processors:
 - Batch processing workloads
 - Media transcoding
 - High performance web servers
 - High performance computing (HPC)
 - Scientific modelling & machine learning
 - Dedicated gaming servers

Compute Optimized

Compute Optimized instances are ideal for compute bound applications that benefit from high performance processors. Instances belonging to this family are well suited for batch processing workloads, media transcoding, high performance web servers, high performance computing (HPC), scientific modeling, dedicated gaming servers and ad server engines, machine learning inference and other compute intensive applications.



** this list will evolve over time, please check the AWS website for the latest information*

EC2 Instance Types – Memory Optimized

- Fast performance for workloads that process large data sets in memory
- Use cases:
 - High performance, relational/non-relational databases
 - Distributed web scale cache stores
 - In-memory databases optimized for BI (business intelligence)
 - Applications performing real-time processing of big unstructured data

Memory Optimized

Memory optimized instances are designed to deliver fast performance for workloads that process large data sets in memory.

| | | | | | | | | | |
|-----|----|-----|-----|-----|----|-----|----|-------------|-----|
| R6g | R5 | R5a | R5b | R5n | R4 | X1e | X1 | High Memory | z1d |
|-----|----|-----|-----|-----|----|-----|----|-------------|-----|

** this list will evolve over time, please check the AWS website for the latest information*

EC2 Instance Types – Storage Optimized

- Great for storage-intensive tasks that require high, sequential read and write access to large data sets on local storage
- Use cases:
 - High frequency online transaction processing (OLTP) systems
 - Relational & NoSQL databases
 - Cache for in-memory databases (for example, Redis)
 - Data warehousing applications
 - Distributed file systems

Storage Optimized

Storage optimized instances are designed for workloads that require high, sequential read and write access to very large data sets on local storage. They are optimized to deliver tens of thousands of low-latency, random I/O operations per second (IOPS) to applications.

| | | | | | |
|----|------|----|----|------|----|
| I3 | I3en | D2 | D3 | D3en | H1 |
|----|------|----|----|------|----|

** this list will evolve over time, please check the AWS website for the latest information*

EC2 Instance Types: example

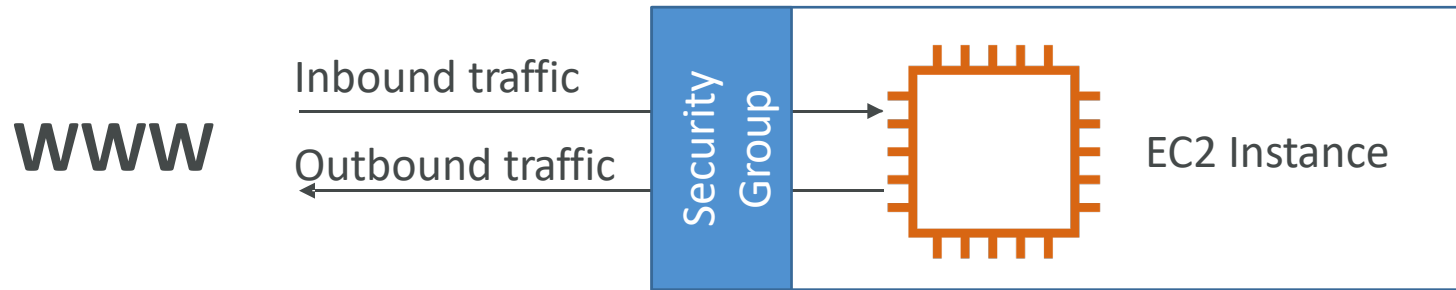
| Instance | vCPU | Mem (GiB) | Storage | Network Performance | EBS Bandwidth (Mbps) |
|-------------|------|--------------|------------------|------------------------|-------------------------|
| t2.micro | 1 | 1 | EBS-Only | Low to Moderate | |
| t2.xlarge | 4 | 16 | EBS-Only | Moderate | |
| c5d.4xlarge | 16 | 32 | 1 x 400 NVMe SSD | Up to 10 Gbps | 4,750 |
| r5.16xlarge | 64 | 512 | EBS Only | 20 Gbps | 13,600 |
| m5.8xlarge | 32 | 128 | EBS Only | 10 Gbps | 6,800 |

t2.micro is part of the AWS free tier (up to 750 hours per month)

Great website: <https://instances.vantage.sh>

Introduction to Security Groups

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



- Security groups only contain **Allow** rules
- Security groups rules can reference by IP or by security group

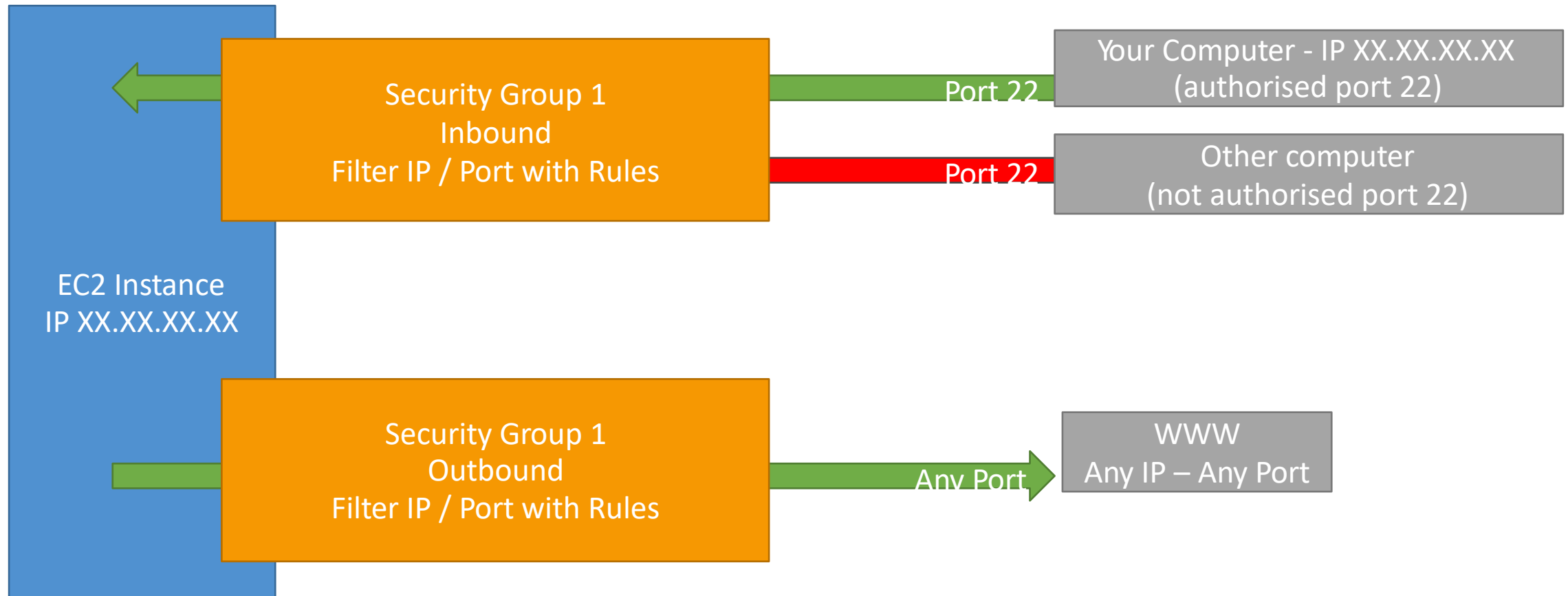
Security Groups

Deeper Dive

- Security groups are acting as a “firewall” on EC2 instances
- They regulate:
 - Access to Ports
 - Authorised IP ranges – IPv4 and IPv6
 - Control of inbound network (from other to the instance)
 - Control of outbound network (from the instance to other)

| 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 Diagram

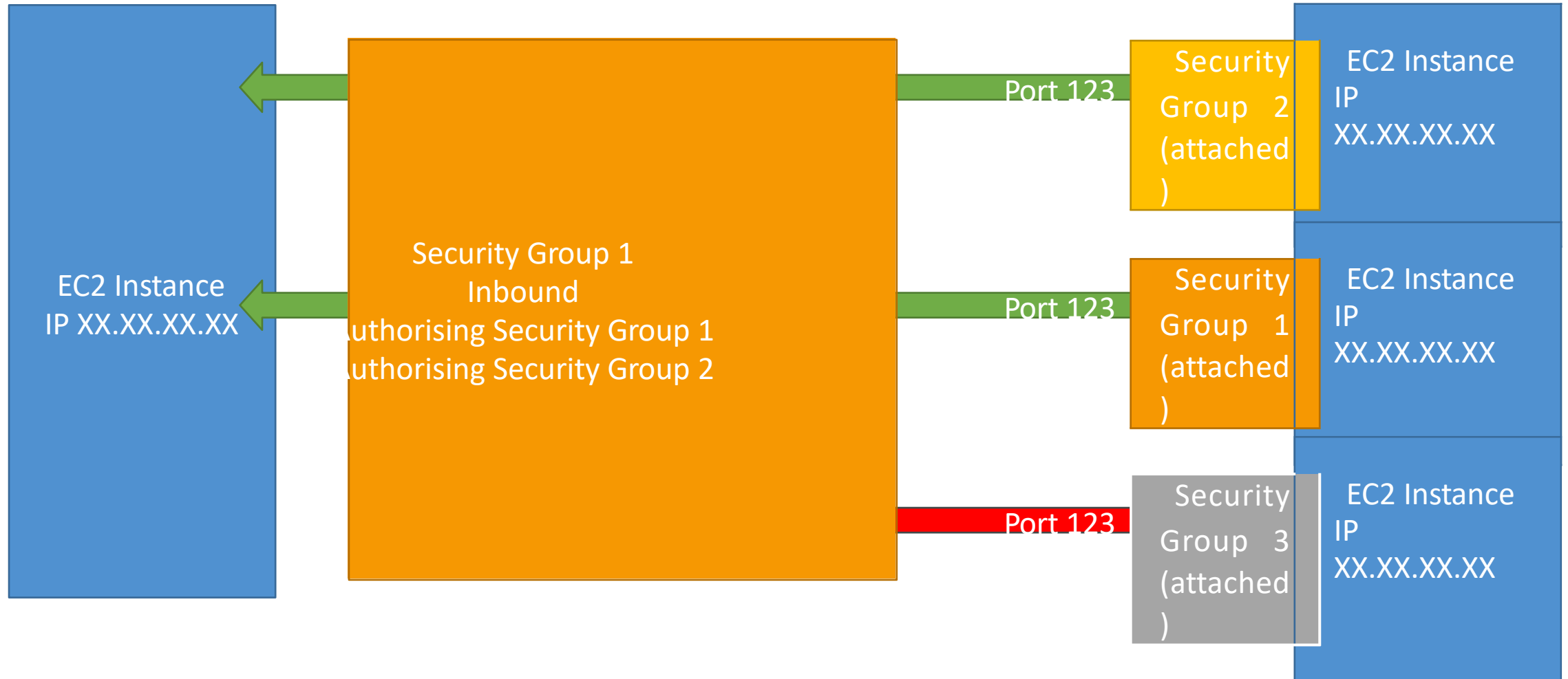


Security Groups

Good to know

- 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 **authorised** by default

Referencing other security groups Diagram



Classic Ports to know

- 22 = SSH (Secure Shell) - log into a Linux instance
- 21 = FTP (File Transfer Protocol) – upload files into a file share
- 22 = SFTP (Secure File Transfer Protocol) – upload files using SSH
- 80 = HTTP – access unsecured websites
- 443 = HTTPS – access secured websites
- 3389 = RDP (Remote Desktop Protocol) – log into a Windows instance

SSH Summary Table

| | SSH | Putty | EC2 Instance Connect |
|---------------|-----|-------|----------------------|
| Mac | ✓ | | ✓ |
| Linux | ✓ | | ✓ |
| Windows < 10 | | ✓ | ✓ |
| Windows >= 10 | ✓ | ✓ | ✓ |

EC2 Instances Purchasing Options

- On-Demand Instances – short workload, predictable pricing, pay by second
- Reserved (1 & 3 years)
 - Reserved Instances – long workloads
 - Convertible Reserved Instances – long workloads with flexible instances
- Savings Plans (1 & 3 years) – commitment to an amount of usage, long workload
- Spot Instances – short workloads, cheap, can lose instances (less reliable)
- Dedicated Hosts – book an entire physical server, control instance placement
- Dedicated Instances – no other customers will share your hardware
- Capacity Reservations – reserve capacity in a specific AZ for any duration

EC2 On Demand

- Pay for what you use:
 - Linux or Windows - billing per second, after the first minute
 - All other operating systems - billing per hour
- Has the highest cost but no upfront payment
- No long-term commitment
- Recommended for short-term and un-interrupted workloads, where you can't predict how the application will behave

EC2 Reserved Instances

- Up to 72% discount compared to On-demand
- You reserve a specific instance attributes (Instance Type, Region, Tenancy, OS)
- Reservation Period – 1 year (+discount) or 3 years (+++discount)
- Payment Options – No Upfront (+), Partial Upfront (++), All Upfront (+++)
- Reserved Instance's Scope – Regional or Zonal (reserve capacity in an AZ)
- Recommended for steady-state usage applications (think database)
- You can buy and sell in the Reserved Instance Marketplace
- Convertible Reserved Instance
 - Can change the EC2 instance type, instance family, OS, scope and tenancy
 - Up to 66% discount

Note: the % discounts are different from the video as AWS change them over time – the exact numbers are not needed for the exam. This is just for illustrative purposes 😊

EC2 Savings Plans

- Get a discount based on long-term usage (up to 72% - same as RIs)
- Commit to a certain type of usage (\$10/hour for 1 or 3 years)
- Usage beyond EC2 Savings Plans is billed at the On-Demand price
- Locked to a specific instance family & AWS region (e.g., M5 in us-east-1)
- Flexible across:
 - Instance Size (e.g., m5.xlarge, m5.2xlarge)
 - OS (e.g., Linux, Windows)
 - Tenancy (Host, Dedicated, Default)

EC2 Spot Instances



- Can get a discount of up to 90% compared to On-demand
- Instances that you can “lose” at any point of time if your max price is less than the current spot price
- The MOST cost-efficient instances in AWS
- Useful for workloads that are resilient to failure
 - Batch jobs
 - Data analysis
 - Image processing
 - Any distributed workloads
 - Workloads with a flexible start and end time
- Not suitable for critical jobs or databases

EC2 Dedicated Hosts

- A physical server with EC2 instance capacity fully dedicated to your use
- Allows you address compliance requirements and use your existing server- bound software licenses (per-socket, per-core, per—VM software licenses)
- Purchasing Options:
 - On-demand – pay per second for active Dedicated Host
 - Reserved - 1 or 3 years (No Upfront, Partial Upfront, All Upfront)
- The most expensive option
- Useful for software that have complicated licensing model (BYOL – Bring Your Own License)
- Or for companies that have strong regulatory or compliance needs

EC2 Dedicated Instances

- Instances run on hardware that's dedicated to you
- May share hardware with other instances in same account
- No control over instance placement (can move hardware after Stop / Start)

| Characteristic | Dedicated Instances | Dedicated Hosts |
|--|---------------------|-----------------|
| Enables the use of dedicated physical servers | X | X |
| Per instance billing (subject to a \$2 per region fee) | X | |
| Per host billing | | X |
| Visibility of sockets, cores, host ID | | X |
| Affinity between a host and instance | | X |
| Targeted instance placement | | X |
| Automatic instance placement | X | X |
| Add capacity using an allocation request | | X |

EC2 Capacity Reservations

- Reserve On-Demand instances capacity in a specific AZ for any duration
- You always have access to EC2 capacity when you need it
- No time commitment (create/cancel anytime), no billing discounts
- Combine with Regional Reserved Instances and Savings Plans to benefit from billing discounts
- You're charged at On-Demand rate whether you run instances or not
- Suitable for short-term, uninterrupted workloads that needs to be in a specific AZ

Which purchasing option is right for me?



- **On demand:** coming and staying in resort whenever we like, we pay the full price
- **Reserved:** like planning ahead and if we plan to stay for a long time, we may get a good discount.
- **Savings Plans:** pay a certain amount per hour for certain period and stay in any room type (e.g., King, Suite, Sea View, ...)
- **Spot instances:** the hotel allows people to bid for the empty rooms and the highest bidder keeps the rooms. You can get kicked out at any time
- **Dedicated Hosts:** We book an entire building of the resort
- **Capacity Reservations:** you book a room for a period with full price even you don't stay in it

Price Comparison

Example – m4.large – us-east-1

| Price Type | Price (per hour) |
|---|--|
| On-Demand | \$0.10 |
| Spot Instance (Spot Price) | \$0.038 - \$0.039 (up to 61% off) |
| Reserved Instance (1 year) | \$0.062 (No Upfront) - \$0.058 (All Upfront) |
| Reserved Instance (3 years) | \$0.043 (No Upfront) - \$0.037 (All Upfront) |
| EC2 Savings Plan (1 year) | \$0.062 (No Upfront) - \$0.058 (All Upfront) |
| Reserved Convertible Instance (1 year) | \$0.071 (No Upfront) - \$0.066 (All Upfront) |
| Dedicated Host | On-Demand Price |
| Dedicated Host Reservation | Up to 70% off |
| Capacity Reservations | On-Demand Price |

Shared Responsibility Model for EC2

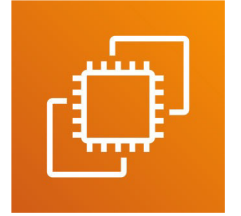


- Infrastructure (global network security)
- Isolation on physical hosts
- Replacing faulty hardware
- Compliance validation



- Security Groups rules
- Operating-system patches and updates
- Software and utilities installed on the EC2 instance
- IAM Roles assigned to EC2 & IAM user access management
- Data security on your instance

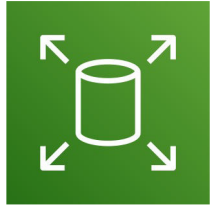
EC2 Section – Summary



- EC2 Instance: AMI (OS) + Instance Size (CPU + RAM) + Storage + security groups + EC2 User Data
- Security Groups: Firewall attached to the EC2 instance
- EC2 User Data: Script launched at the first start of an instance
- SSH: start a terminal into our EC2 Instances (port 22)
- EC2 Instance Role: link to IAM roles
- Purchasing Options: On-Demand, Spot, Reserved (Standard + Convertible + Scheduled), Dedicated Host, Dedicated Instance

EC2 Instance Storage Section

What's an EBS Volume?



- An EBS (Elastic Block Store) Volume is a network drive you can attach to your instances while they run
 - It allows your instances to persist data, even after their termination
 - They can only be mounted to one instance at a time (at the CCP level)
 - They are bound to a specific availability zone
-
- Analogy: Think of them as a “network USB stick”
 - Free tier: 30 GB of free EBS storage of type General Purpose (SSD) or Magnetic per month

EBS Volume

- It's a network drive (i.e. not a physical drive)
 - It uses the network to communicate the instance, which means there might be a bit of latency
 - It can be detached from an EC2 instance and attached to another one quickly
- It's locked to an Availability Zone (AZ)
 - An EBS Volume in us-east-1a cannot be attached to us-east-1b
 - To move a volume across, you first need to snapshot it
- Have a provisioned capacity (size in GBs, and IOPS)
 - You get billed for all the provisioned capacity
 - You can increase the capacity of the drive over time

EBS Volume Types

General Purpose SSD:

- Use for dev/test environments and smaller DB instances
- Performance of 3 IOPS/GB of storage size (burstable with baseline performance)
- Volume size of 1GB to 16TB
- Considerations when using T2 instances with SSD root volumes (burstable vs. baseline performance)

Provisioned IOPS SSD:

- Used for mission critical applications that require sustained IOPS performance
- Large database workloads
- Volume size of 4GB to 16TB
- Performs at provisioned level and can provision up to 32,000 IOPS per volume

Throughput Optimized HDD and Cold HDD:

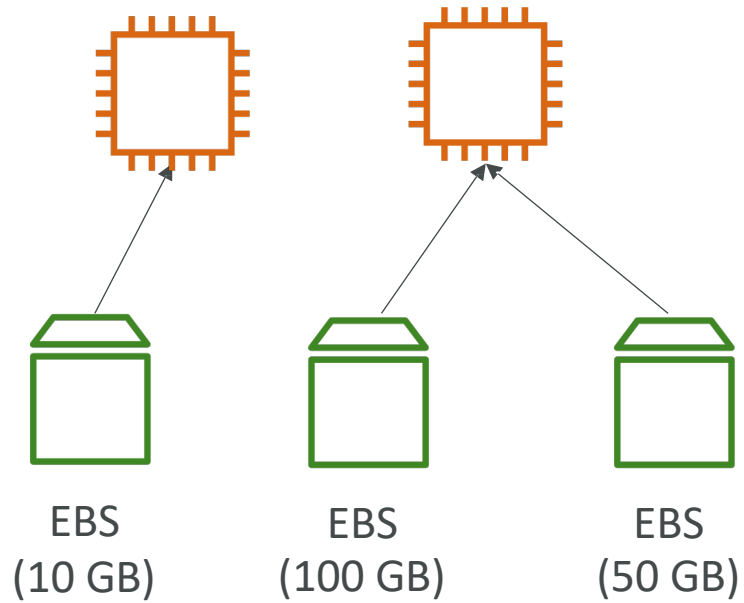
- Cheaper than SSD options, also less performant
- Cold HDD - Designed for less-frequent access
- Volume size of 500GB - 16 TB
- Cannot be a boot volume

EBS Magnetic (Previous Generation):

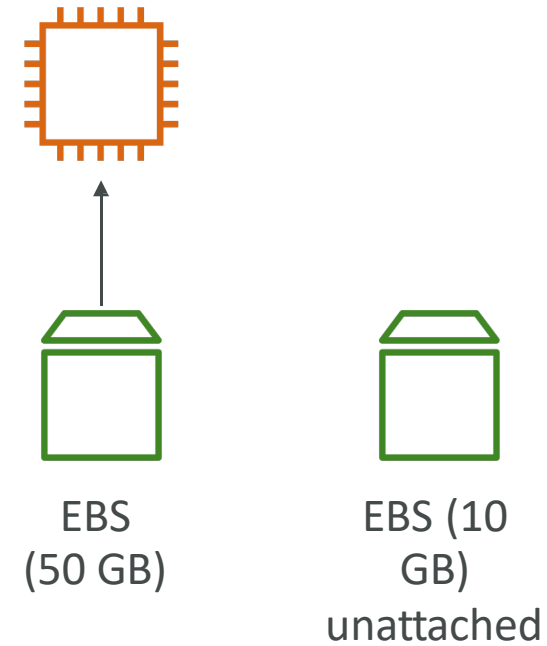
- Low storage cost
- Used for workloads where performance is not important or data is infrequently accessed
- Volume size of Min 1GB Max 1 TB

EBS Volume - Example

US-EAST-1A



US-EAST-1B



EBS – Delete on Termination attribute

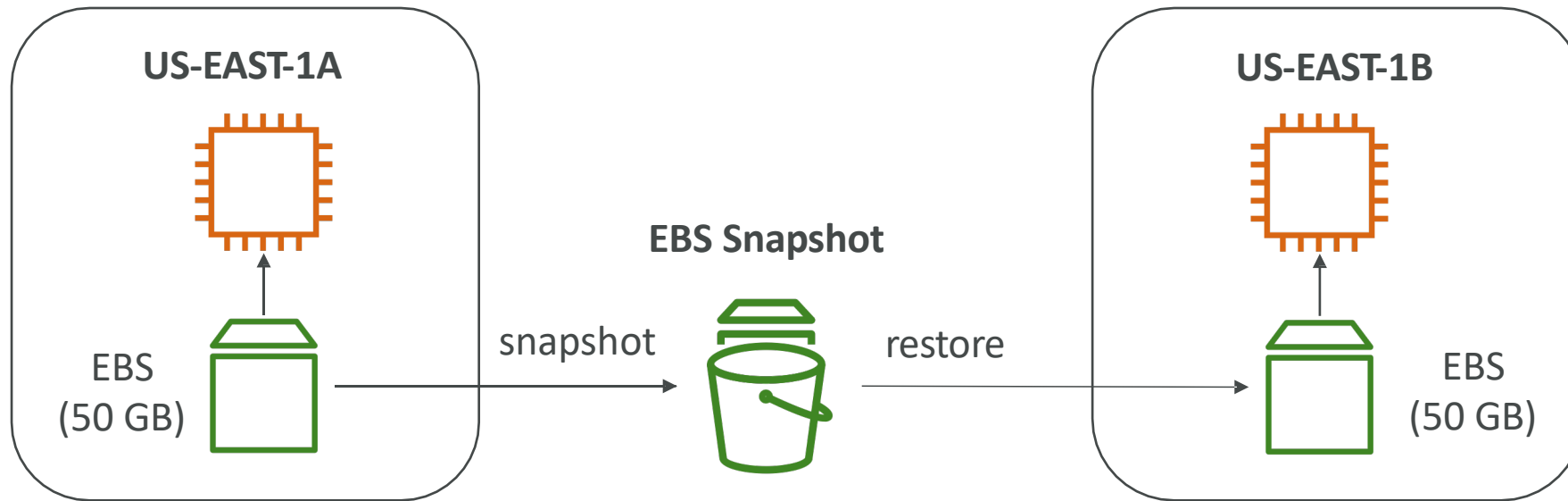
| Volume Type ⓘ | Device ⓘ | Snapshot ⓘ | Size (GiB) ⓘ | Volume Type ⓘ | IOPS ⓘ | Throughput (MB/s) ⓘ | Delete on Termination ⓘ | Encryption ⓘ |
|---------------|------------|------------------------|--------------|-----------------------------|------------|---------------------|-------------------------------------|-----------------|
| Root | /dev/xvda | snap-09f18f682fd23a1b1 | 8 | General Purpose SSD (gp2) ▼ | 100 / 3000 | N/A | <input checked="" type="checkbox"/> | Not Encrypted ▼ |
| EBS ▼ | /dev/sdb ▼ | Search (case-insensit | 8 | General Purpose SSD (gp2) ▼ | 100 / 3000 | N/A | <input type="checkbox"/> | Not Encrypted ▼ |

Add New Volume

- Controls the EBS behaviour when an EC2 instance terminates
 - By default, the root EBS volume is deleted (attribute enabled)
 - By default, any other attached EBS volume is not deleted (attribute disabled)
- This can be controlled by the AWS console / AWS CLI
- Use case: preserve root volume when instance is terminated

EBS Snapshots

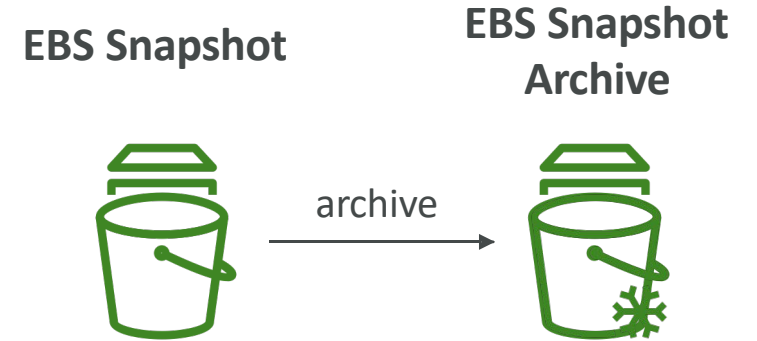
- Make a backup (snapshot) of your EBS volume at a point in time
- Not necessary to detach volume to do snapshot, but recommended
- Can copy snapshots across AZ or Region



EBS Snapshots Features

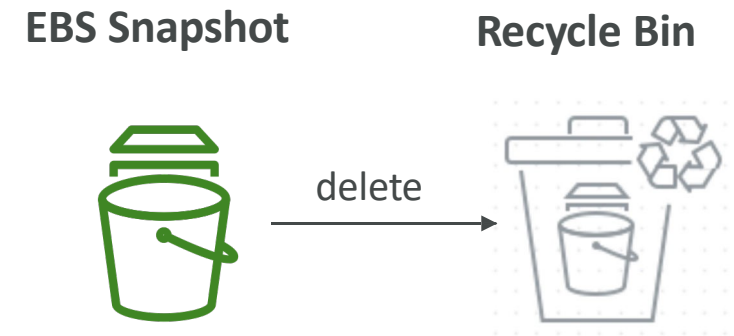
- EBS Snapshot Archive

- Move a Snapshot to an "archive tier" that is 75% cheaper
- Takes within 24 to 72 hours for restoring the archive



- Recycle Bin for EBS Snapshots

- Setup rules to retain deleted snapshots so you can recover them after an accidental deletion
- Specify retention (from 1 day to 1 year)



Mounting EBS Volume

- Launch a ec2 instance and create an EBS volume in the same az and attach it to instance
- The volumes attached to instance1 can be verified it by executing “lsblk” command, The volume is attached but it’s not mounted [df -TH]
- Mount the volume to instance
 - Format the disk with xfs file system : `mkfs -t ext4 /dev/xvdf`
 - Create a directory in root: `mkdir /mnt/mydisk`
 - Mount the disk: `mount /dev/xvdf /mnt/mydisk`
 - you can verify that disk is mounted by running [df -TH] command.
- Unmount the disk
 - Unmount Volume: `mount /mnt/mydisk`
 - Detach the volume from ec2 instance.
 - delete the volume

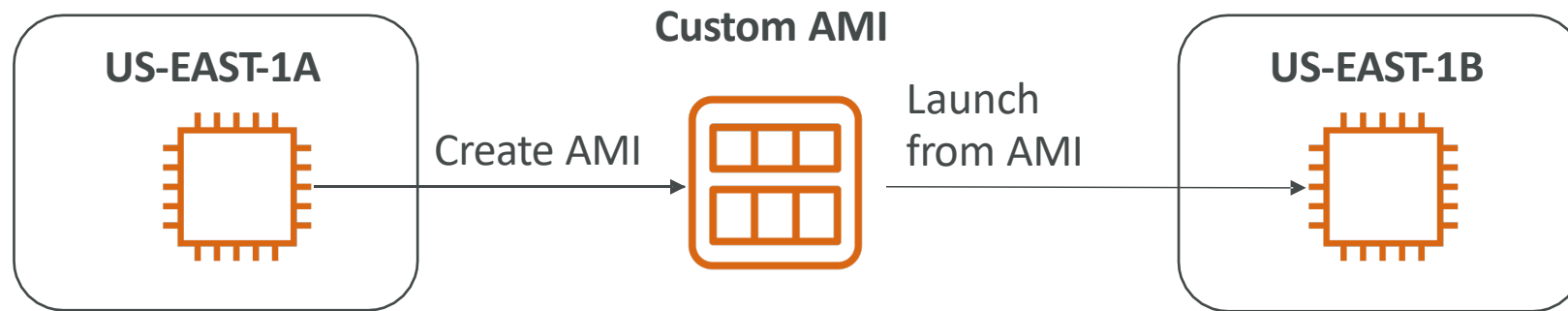
AMI Overview



- AMI = Amazon Machine Image
- AMI are a customization of an EC2 instance
 - You add your own software, configuration, operating system, monitoring...
 - Faster boot / configuration time because all your software is pre-packaged
- AMI are built for a specific region (and can be copied across regions)
- You can launch EC2 instances from:
 - A Public AMI: AWS provided
 - Your own AMI: you make and maintain them yourself
 - An AWS Marketplace AMI: an AMI someone else made (and potentially sells)

AMI Process (from an EC2 instance)

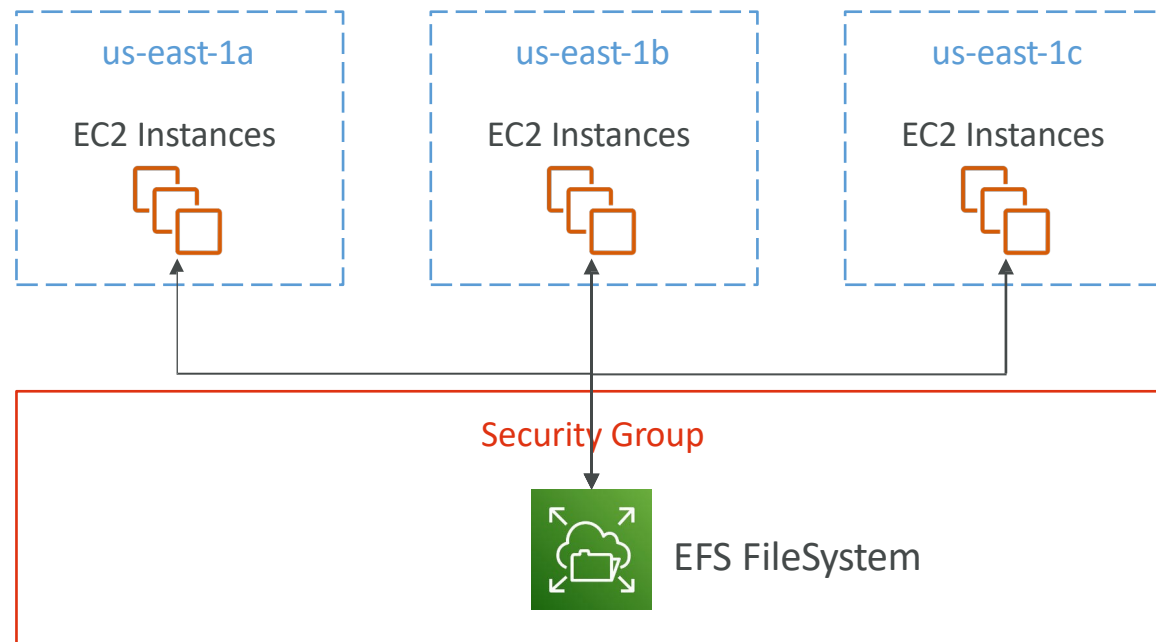
- Start an EC2 instance and customize it
- Stop the instance (for data integrity)
- Build an AMI – this will also create EBS snapshots
- Launch instances from other AMIs



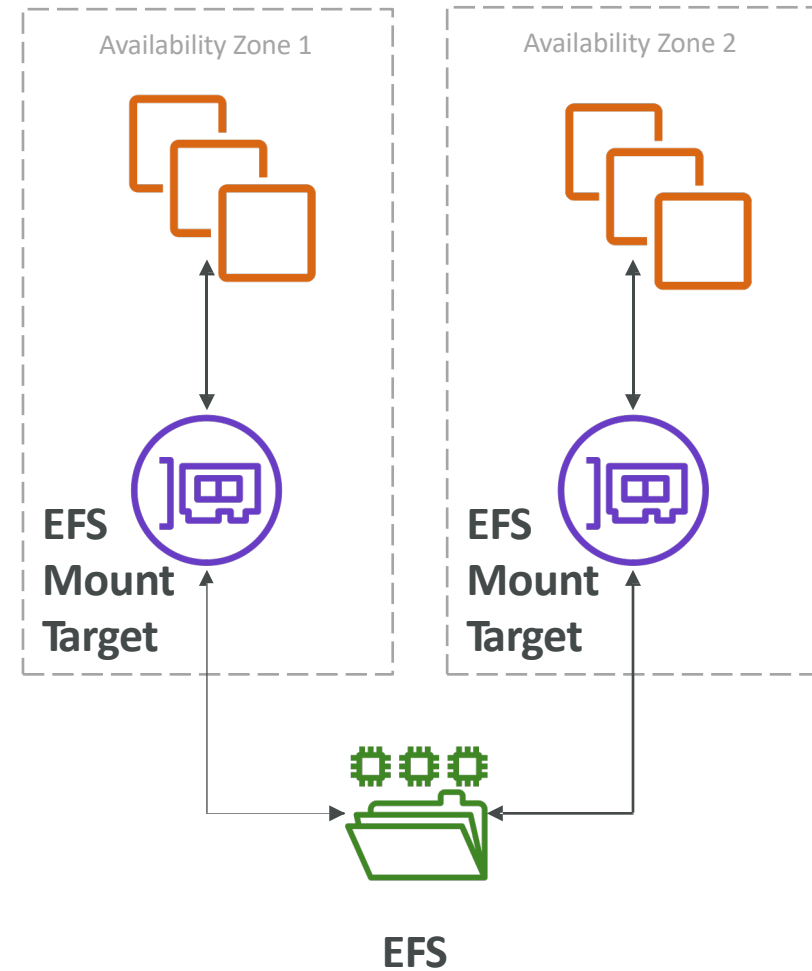
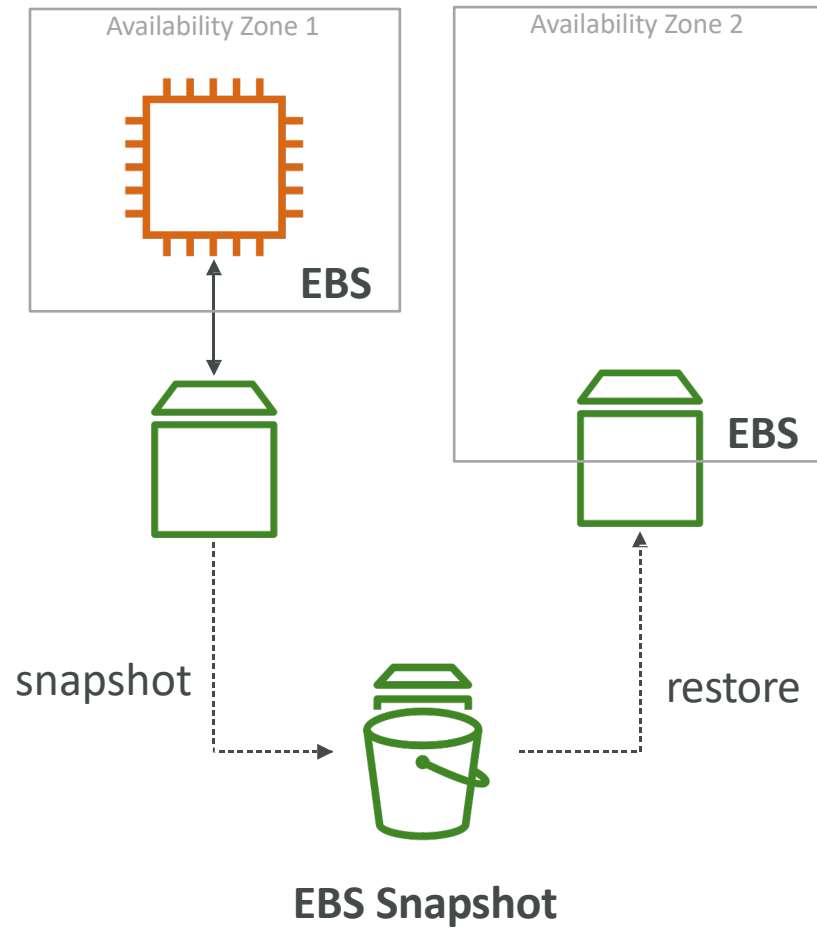
EFS – Elastic File System



- Managed NFS (network file system) that can be mounted on 100s of EC2
- EFS works with Linux EC2 instances in multi-AZ
- Highly available, scalable, expensive (3x gp2), pay per use, no capacity planning

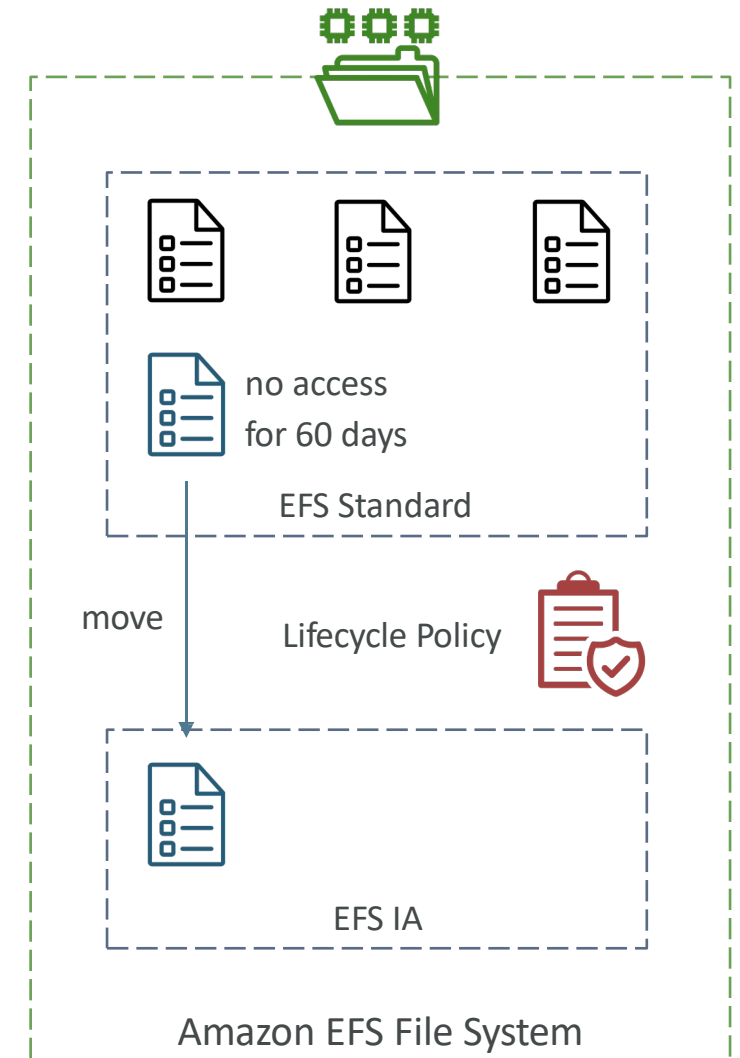


EBS vs EFS



EFS Infrequent Access (EFS-IA)

- Storage class that is cost-optimized for files not accessed every day
- Up to 92% lower cost compared to EFS Standard
- EFS will automatically move your files to EFS-IA based on the last time they were accessed
- Enable EFS-IA with a Lifecycle Policy
- Example: move files that are not accessed for 60 days to EFS-IA
- Transparent to the applications accessing EFS



EC2 Instance Storage - Summary

- EBS volumes:
 - network drives attached to one EC2 instance at a time
 - Mapped to an Availability Zones
 - Can use EBS Snapshots for backups / transferring EBS volumes across AZ
- AMI: create ready-to-use EC2 instances with our customizations
- EC2 Image Builder: automatically build, test and distribute AMIs
- EC2 Instance Store:
 - High performance hardware disk attached to our EC2 instance
 - Lost if our instance is stopped / terminated
- EFS: network file system, can be attached to 100s of instances in a region
- EFS-IA: cost-optimized storage class for infrequent accessed files

Elastic Load Balancing & Auto Scaling Groups Section

Scalability & High Availability

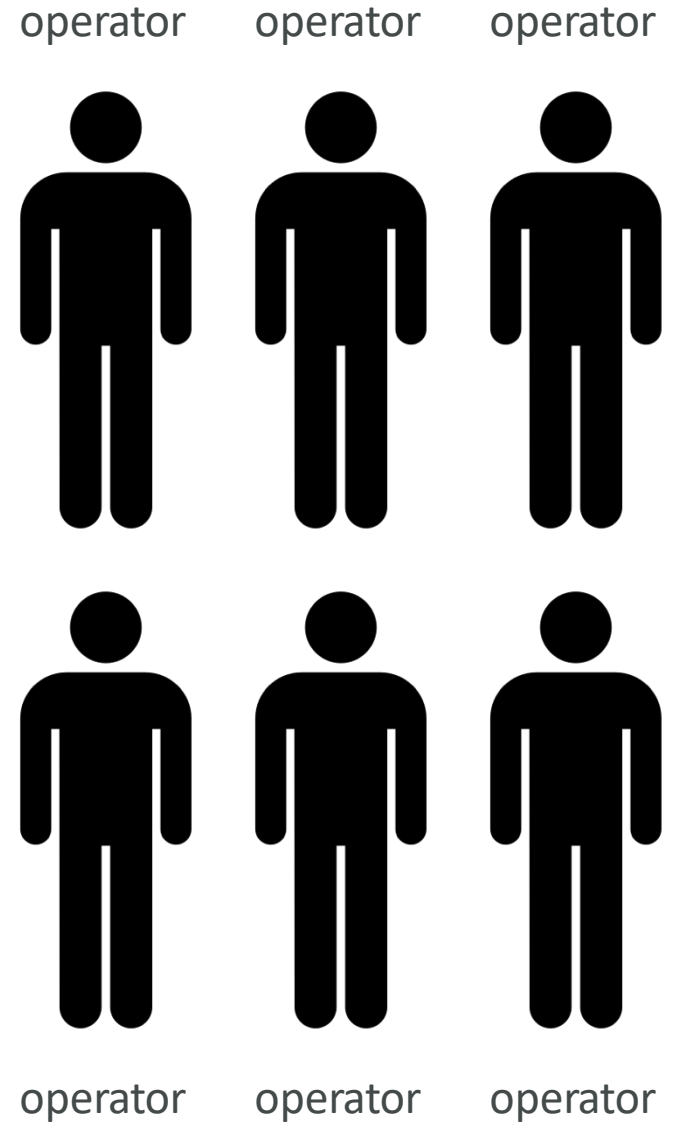
- Scalability means that an application / system can handle greater loads by adapting.
- There are two kinds of scalability:
 - Vertical Scalability
 - Horizontal Scalability (= elasticity)

Vertical Scalability

- Vertical Scalability means increasing the size of the instance
- For example, your application runs on a t2.micro
- Scaling that application vertically means running it on a t2.large
- Vertical scalability is very common for non distributed systems, such as a database.
- There's usually a limit to how much you can vertically scale (hardware limit)

Horizontal Scalability

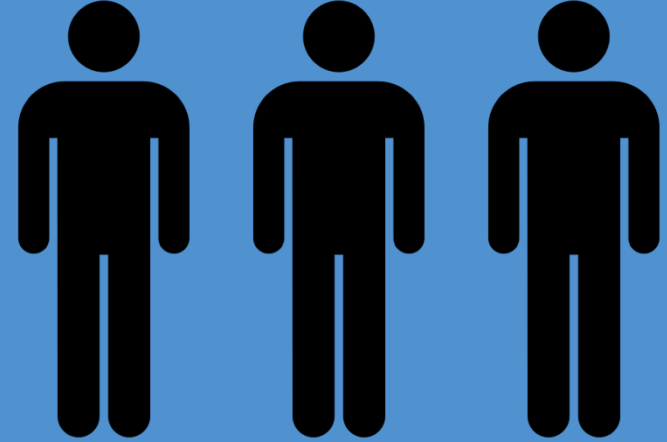
- Horizontal Scalability means increasing the number of instances / systems for your application
- Horizontal scaling implies distributed systems.
- This is very common for web applications / modern applications
- It's easy to horizontally scale thanks the cloud offerings such as Amazon EC2



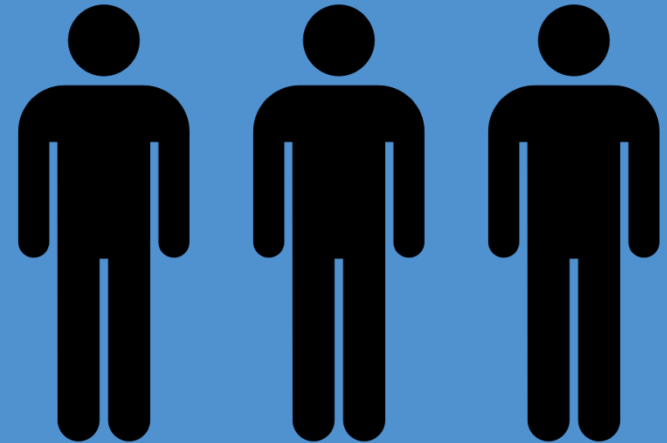
High Availability

- High Availability usually goes hand in hand with horizontal scaling
- High availability means running your application / system in at least 2 Availability Zones
- The goal of high availability is to survive a data center loss (disaster)

first building in New York



second building in San Francisco



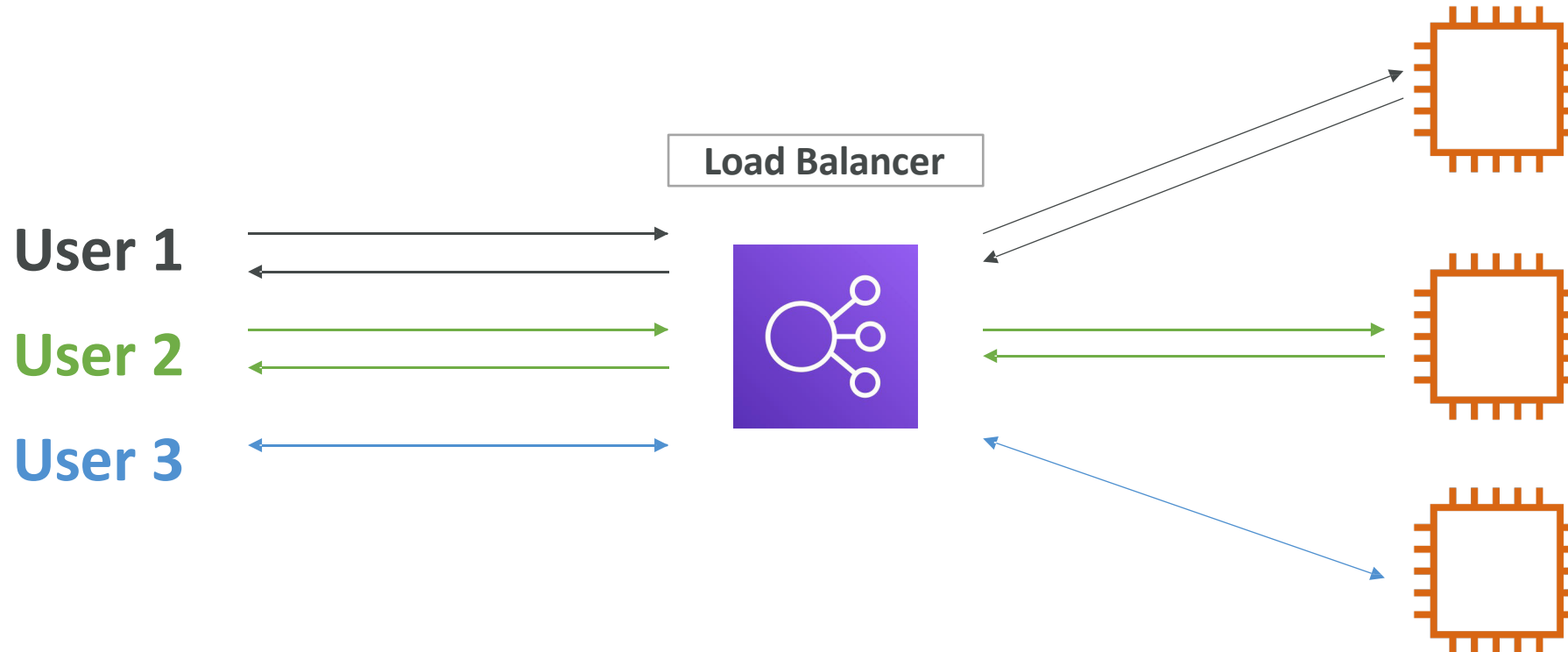
High Availability & Scalability For EC2

- Vertical Scaling: Increase instance size (= scale up / down)
 - From: t2.nano - 0.5G of RAM, 1 vCPU
 - To: u-12tb1.metal – 12.3 TB of RAM, 448 vCPUs
- Horizontal Scaling: Increase number of instances (= scale out / in)
 - Auto Scaling Group
 - Load Balancer
- High Availability: Run instances for the same application across multi AZ
 - Auto Scaling Group multi AZ
 - Load Balancer multi AZ

What is load balancing?



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



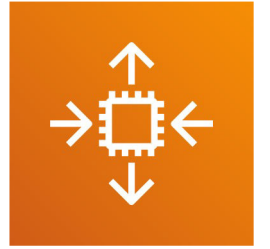
Why use a load balancer?

- 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
- High availability across zones

Why use an Elastic Load Balancer?

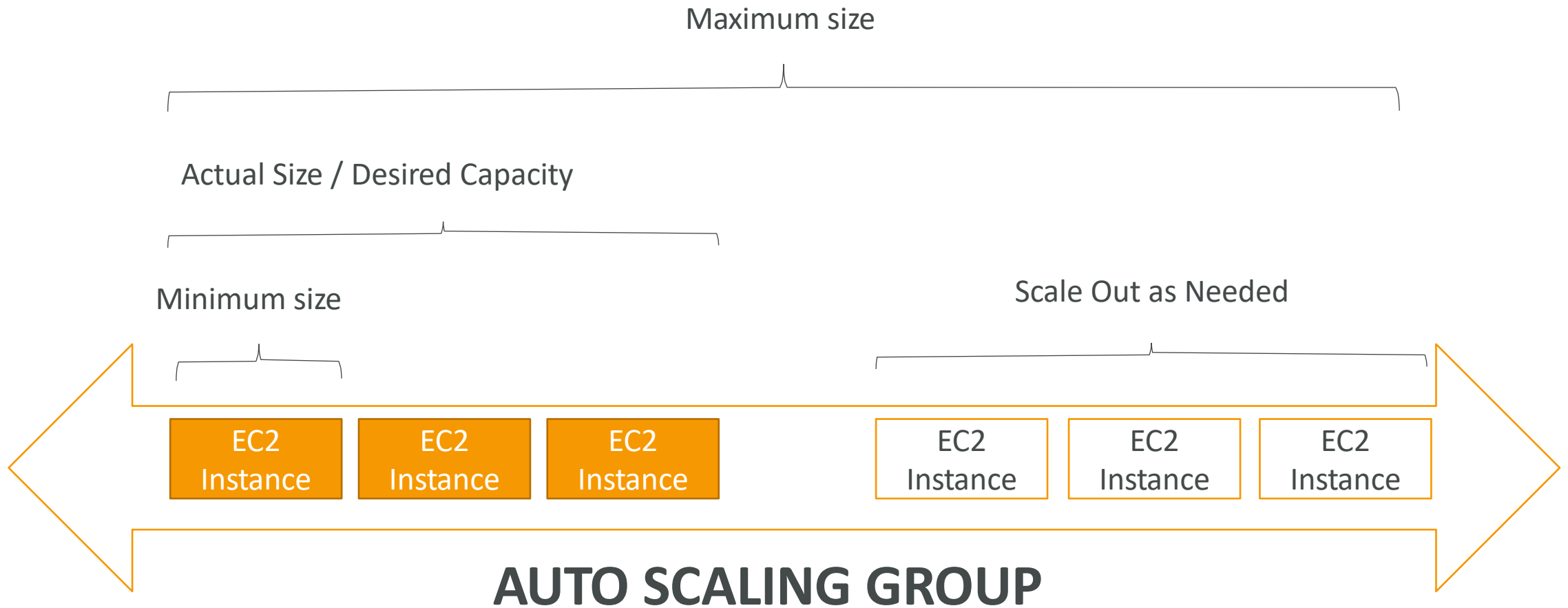
- An ELB (Elastic Load Balancer) is a managed load balancer
 - AWS guarantees that it will be working
 - AWS takes care of upgrades, maintenance, high availability
 - AWS provides only a few configuration knobs
- It costs less to setup your own load balancer but it will be a lot more effort on your end (maintenance, integrations)
- 3 kinds of load balancers offered by AWS:
 - Application Load Balancer (HTTP / HTTPS only) – Layer 7
 - Network Load Balancer (ultra-high performance, allows for TCP) – Layer 4
 - Classic Load Balancer (slowly retiring) – Layer 4 & 7

What's an Auto Scaling Group?

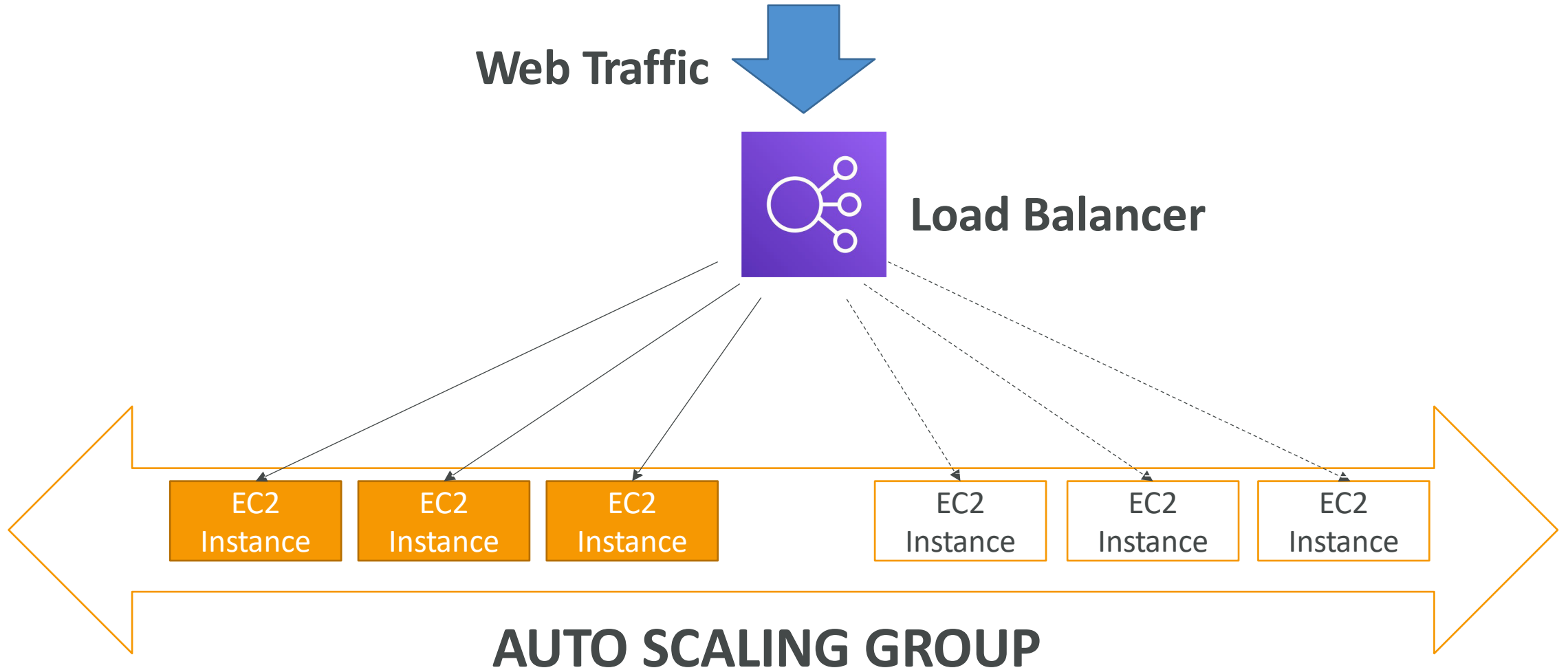


- In real-life, the load on your websites and application can change
- In the cloud, you can create and get rid of servers very quickly
- The goal of an Auto Scaling Group (ASG) is to:
 - Scale out (add EC2 instances) to match an increased load
 - Scale in (remove EC2 instances) to match a decreased load
 - Ensure we have a minimum and a maximum number of machines running
 - Automatically register new instances to a load balancer
 - Replace unhealthy instances
- Cost Savings: only run at an optimal capacity (principle of the cloud)

Auto Scaling Group in AWS



Auto Scaling Group in AWS With Load Balancer

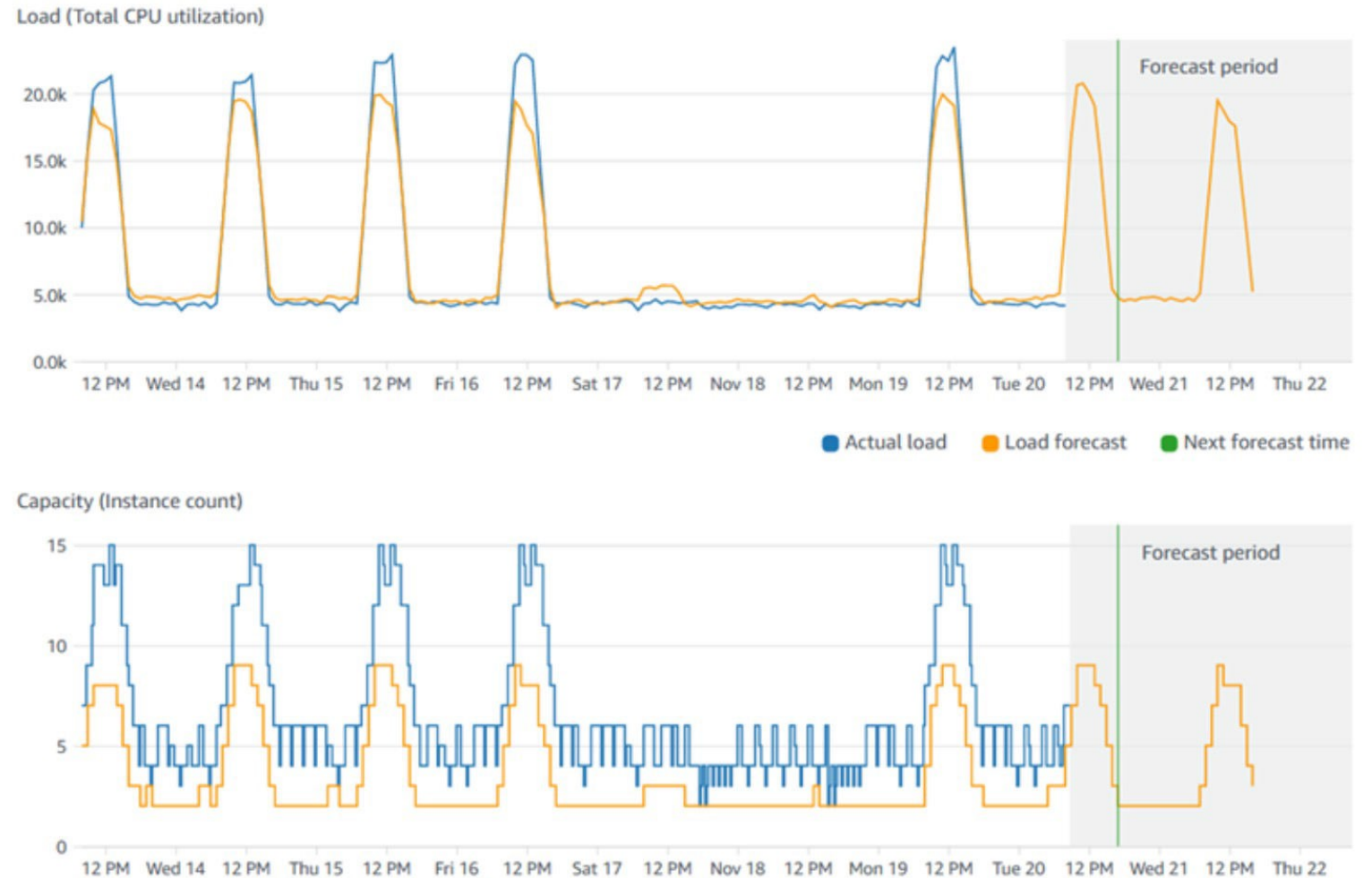


Auto Scaling Groups – Scaling Strategies

- Manual Scaling: Update the size of an ASG manually
- Dynamic Scaling: Respond to changing demand
 - Simple / Step Scaling
 - When a CloudWatch alarm is triggered (example CPU > 70%), then add 2 units
 - When a CloudWatch alarm is triggered (example CPU < 30%), then remove 1
 - Target Tracking Scaling
 - Example: I want the average ASG CPU to stay at around 40%
- Scheduled Scaling
 - Anticipate a scaling based on known usage patterns
 - Example: increase the min. capacity to 10 at 5 pm on Fridays

Auto Scaling Groups – Scaling Strategies

- Predictive Scaling
 - Uses Machine Learning to predict future traffic ahead of time
 - Automatically provisions the right number of EC2 instances in advance
- Useful when your load has predictable time-based patterns



ELB & ASG – Summary

- High Availability vs Scalability (vertical and horizontal) vs Elasticity vs Agility in the Cloud
- Elastic Load Balancers (ELB)
 - Distribute traffic across backend EC2 instances, can be Multi-AZ
 - Supports health checks
 - 3 types: Application LB (HTTP – L7), Network LB (TCP – L4), Classic LB (old)
- Auto Scaling Groups (ASG)
 - Implement Elasticity for your application, across multiple AZ
 - Scale EC2 instances based on the demand on your system, replace unhealthy
 - Integrated with the ELB