Amazon EC2 - Basics

Amazon EC2

- EC2 is one of the most popular of AWS’ offering
- 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

Hands-On

Launching an EC2 Instance running Linux

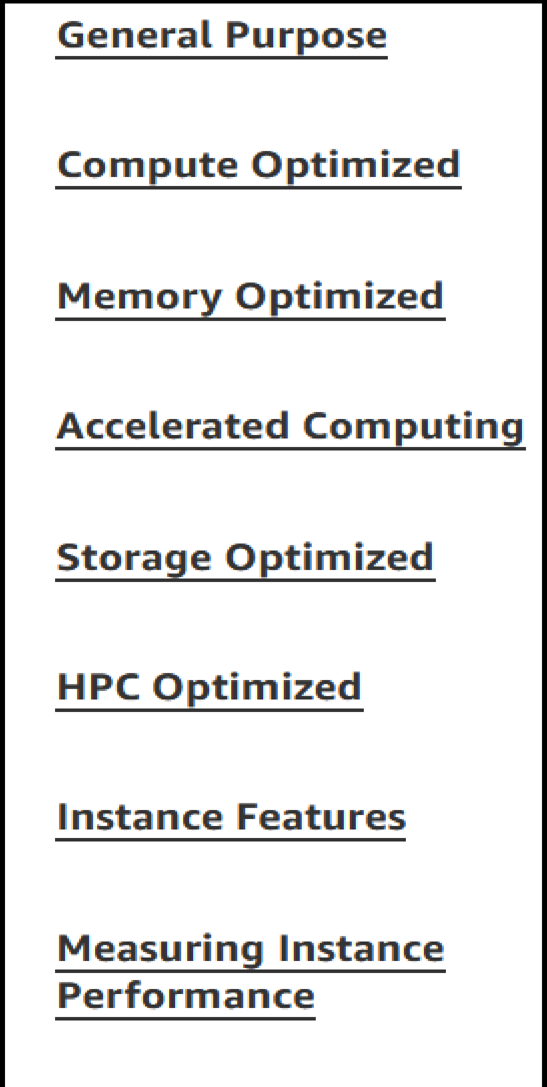
- We’ll be launching our first virtual server using the AWS Console
- We’ll get a first high-level approach to the various parameters
- We’ll see that our web server is launched using EC2 user data
- We’ll learn how to start / stop / terminate our instance.

EC2 Instance Types - Overview

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

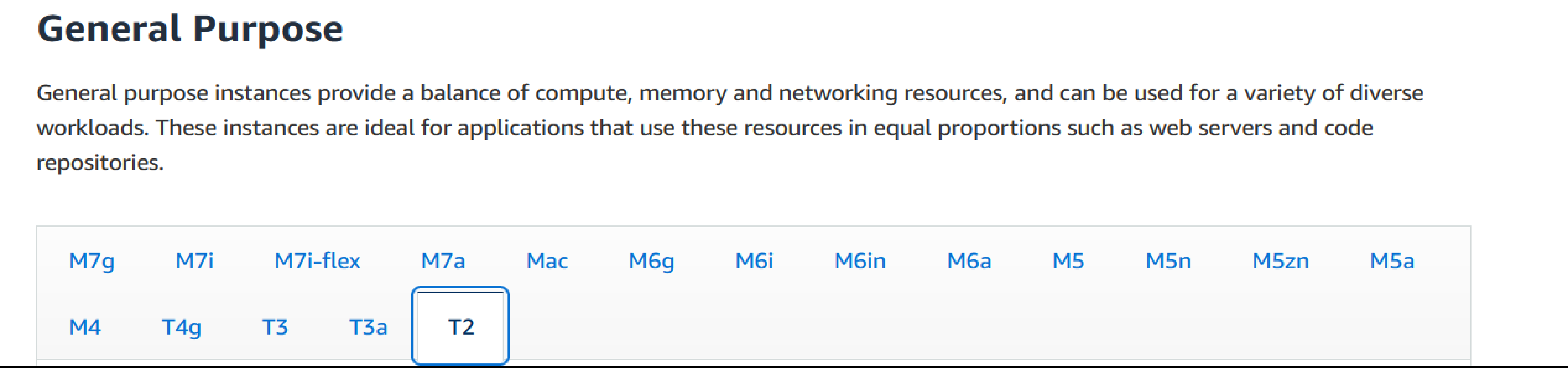
m5.2xlarge

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



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

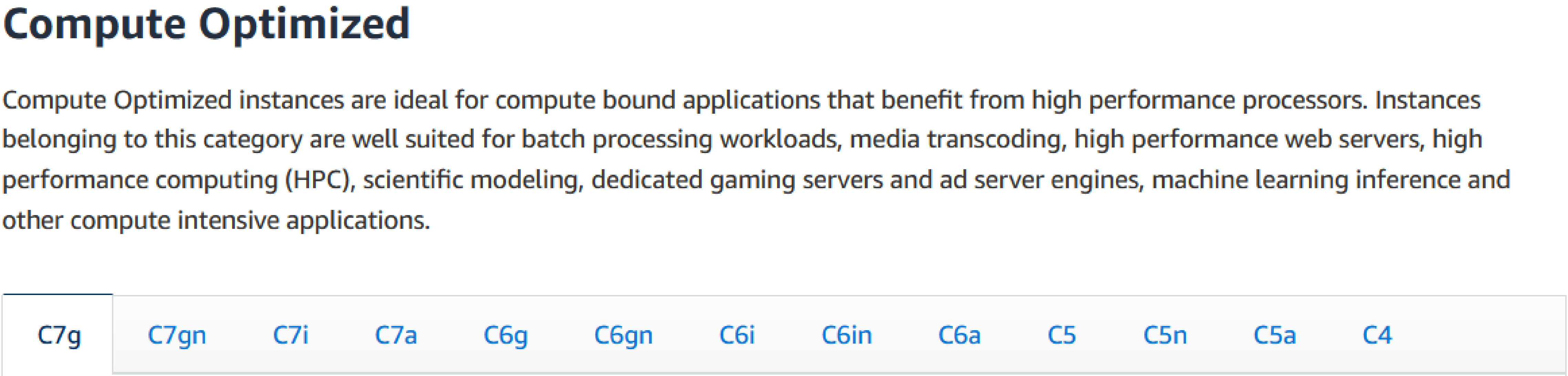


General Purpose

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.

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 modeling & machine learning
  - Dedicated gaming servers

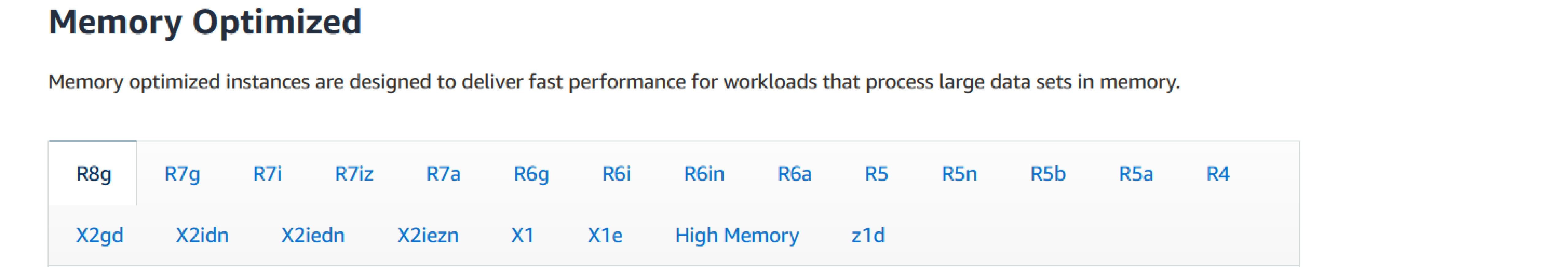


Compute Optimized

Compute Optimized instances are ideal for compute bound applications that benefit from high performance processors. Instances belonging to this category 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.

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

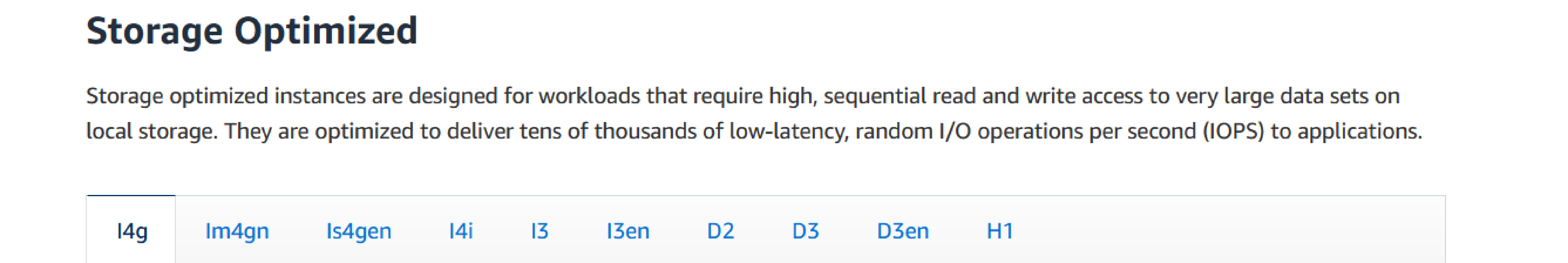


Memory Optimized

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

EC2 Instance Types – Storage Optimized

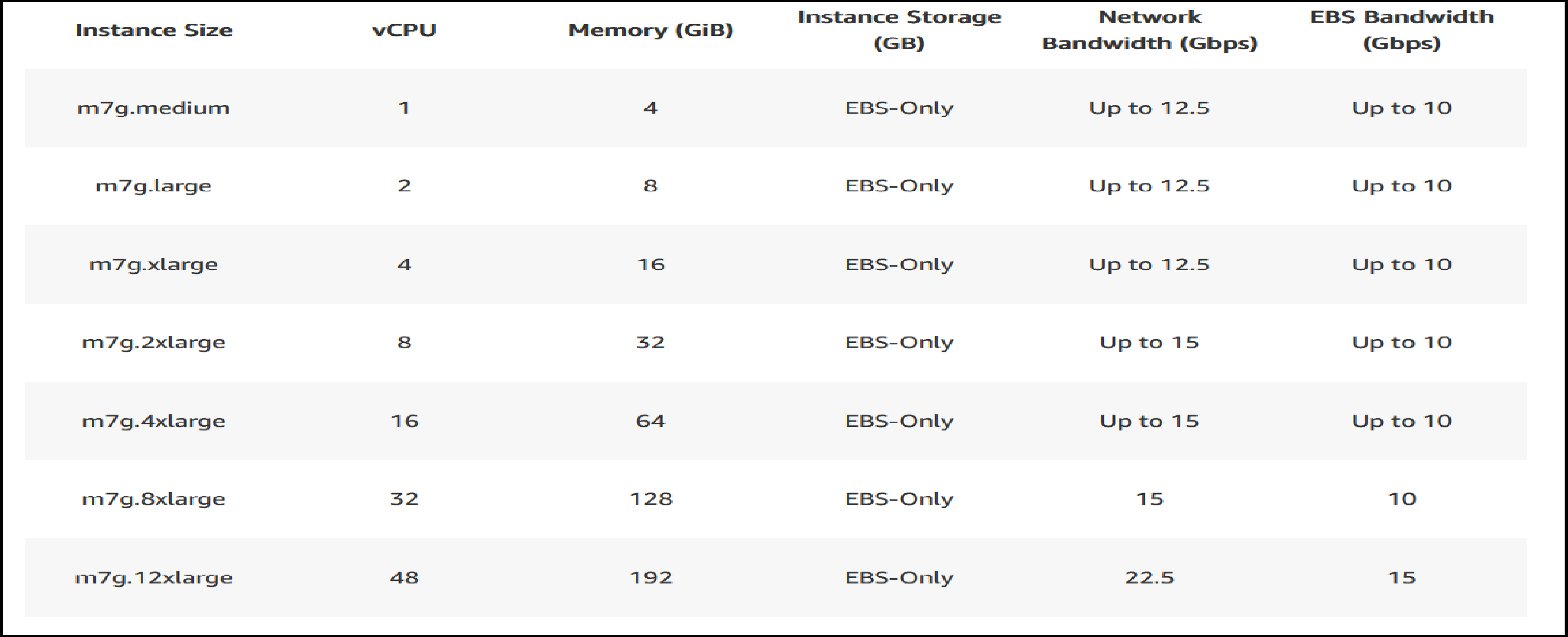
- Great for storage-intensive tasks that require high, sequential read and write access to large data sets on local storage
- Use cases:
  - High frequency online transaction processing (OLTP) systems
  - Relational & NoSQL databases
  - Cache for in-memory databases (for example, Redis)
  - Data warehousing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

EC2 Instance Types: Example

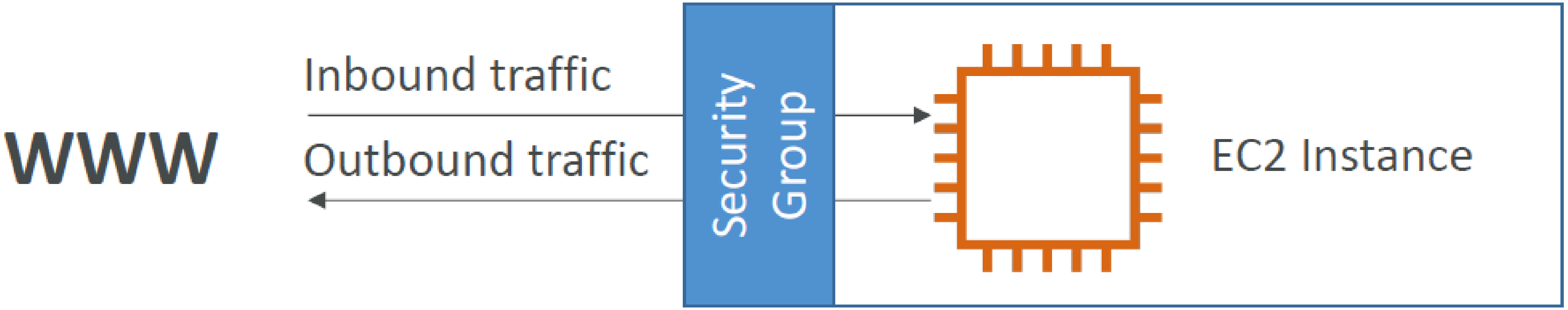


Instance Size	vCPU	Memory (GiB)	Instance Storage (GB)	Network Bandwidth (Gbps)	EBS Bandwidth (Gbps)
m7g.medium	1	4	EBS-Only	Up to 12.5	Up to 10
m7g.large	2	8	EBS-Only	Up to 12.5	Up to 10
m7g.xlarge	4	16	EBS-Only	Up to 12.5	Up to 10
m7g.2xlarge	8	32	EBS-Only	Up to 15	Up to 10
m7g.4xlarge	16	64	EBS-Only	Up to 15	Up to 10
m7g.8xlarge	32	128	EBS-Only	15	10
m7g.12xlarge	48	192	EBS-Only	22.5	15

Great website: ***https://instances.vantage.sh***

Introduction to Security Groups

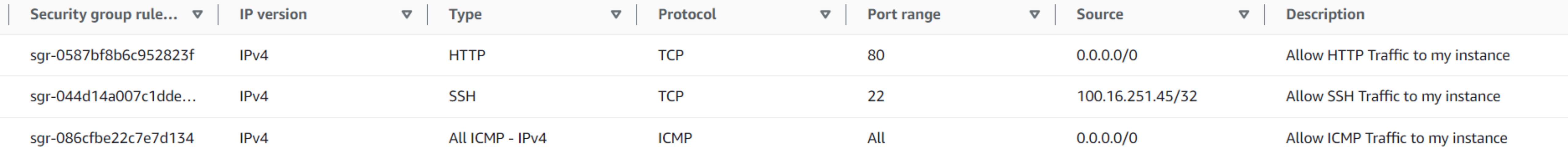
- Security Groups are the fundamental of network security in AWS
- They control how traffic is allowed into or out of our EC2 Instances.



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

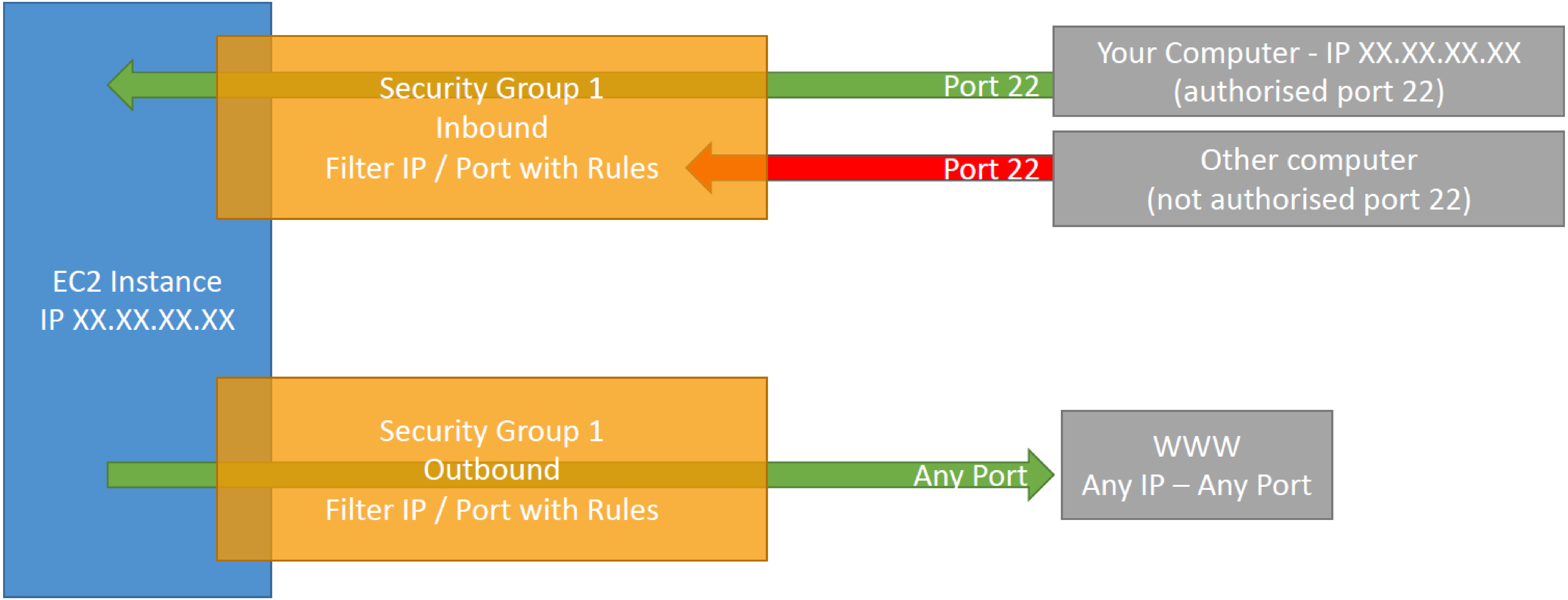
Security Groups Deeper Dive

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



Security group rule...	IP version	Type	Protocol	Port range	Source	Description
sgr-0587bf8b6c952823f	IPv4	HTTP	TCP	80	0.0.0.0/0	Allow HTTP Traffic to my instance
sgr-044d14a007c1dde...	IPv4	SSH	TCP	22	100.16.251.45/32	Allow SSH Traffic to my instance
sgr-086cfbe22c7e7d134	IPv4	All ICMP - IPv4	ICMP	All	0.0.0.0/0	Allow ICMP Traffic to my instance

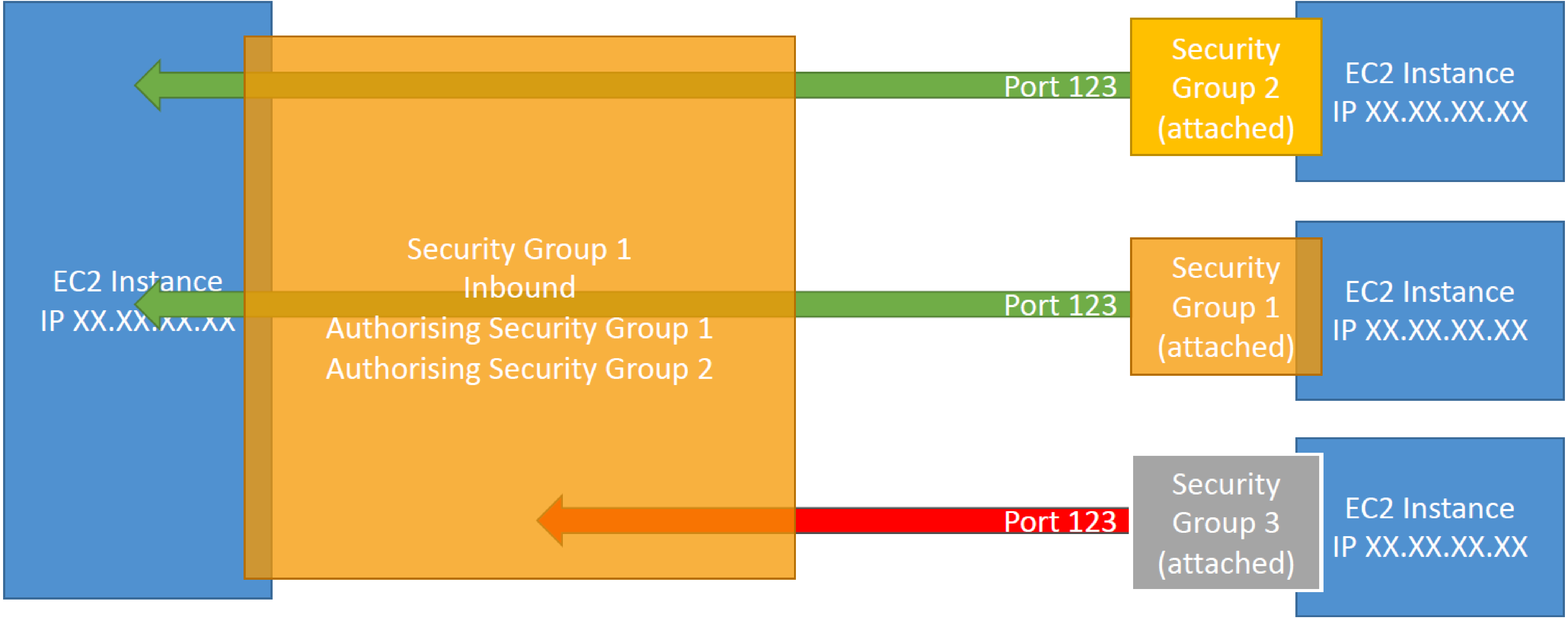
Security Groups Diagram



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 authorized 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	✓	✓	✓

SSH Troubleshooting

- **Students have the most problems with SSH**
- If things don’t work…
  - 1. Re-watch the lecture. You may have missed something
  - 2. Read the troubleshooting guide
  - 3. Try EC2 Instance Connect
- **If one method works (SSH, Putty or EC2 Instance Connect) you’re good**
- If no method works, that’s okay, the course won’t use SSH much

EC2 Instance Connect

- Connect to your EC2 instance within your browser
- No need to use your key file that was downloaded
- The “magic” is that a temporary key is uploaded onto EC2 by AWS
- **Works only out-of-the-box with Amazon Linux 2**
- Need to make sure the port 22 is still opened

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 Sc**ope – **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

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, pe—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

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)

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

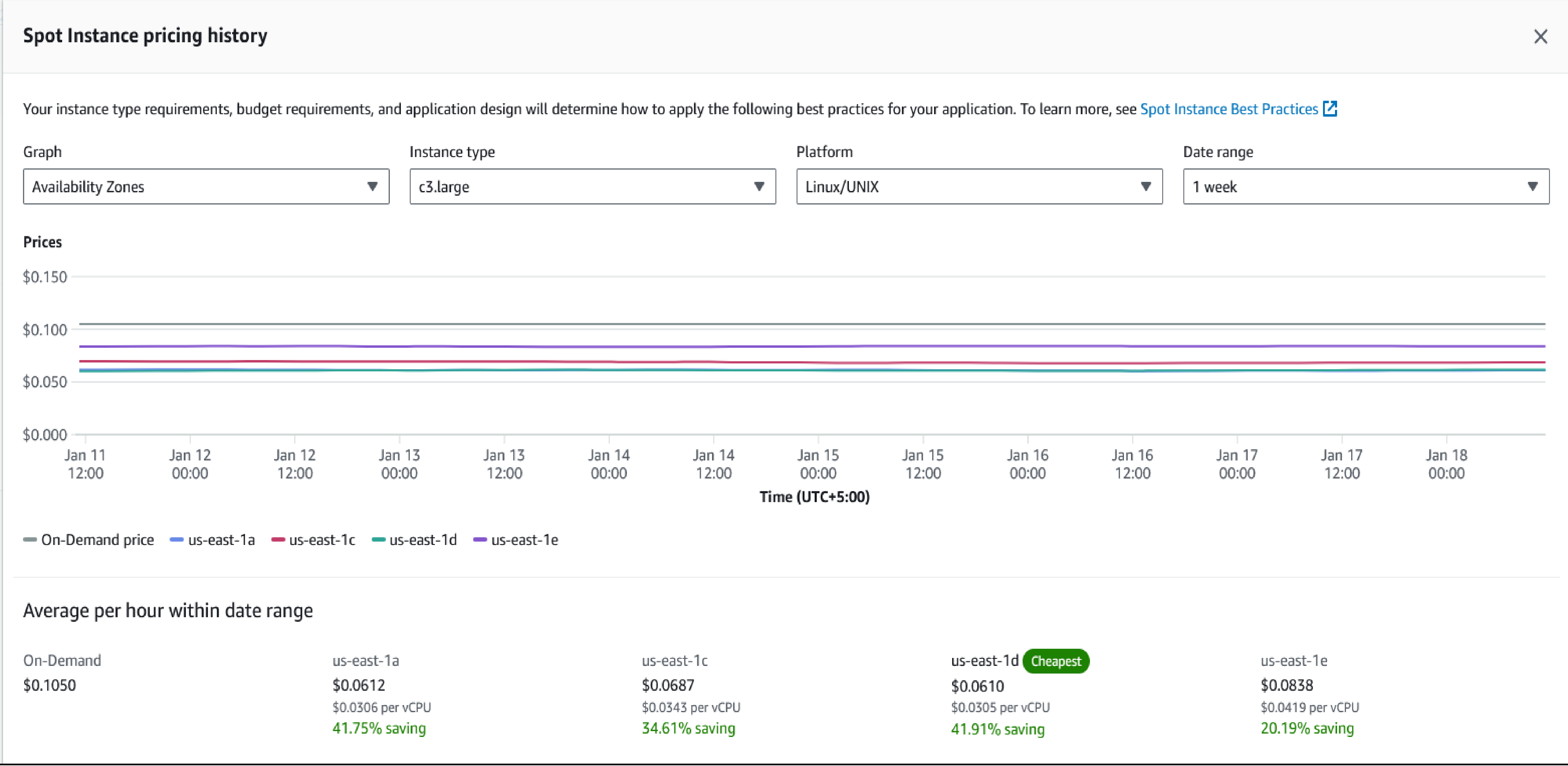
Which Purchasing Option is Right For Me?

- **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

EC2 Spot Instance Requests

- Can get a discount of up to 90% compared to On-demand
- Define **max spot price** and get the instance while current spot price < max
  - The hourly spot price varies based on offer and capacity
  - If the current spot price > your max price you can choose to stop or terminate your instance with a 2 minutes grace period.
- Other strategy: **Spot Block**
  - “block” spot instance during a specified time frame (1 to 6 hours) without interruptions
  - In rare situations, the instance may be reclaimed
- **Used for batch jobs, data analysis, or workloads that are resilient to failures.**
- **Not great for critical jobs or databases**

EC2 Spot Instances Pricing



Spot Instance pricing history

Your instance type requirements, budget requirements, and application design will determine how to apply the following best practices for your application. To learn more, see Spot Instance Best Practices

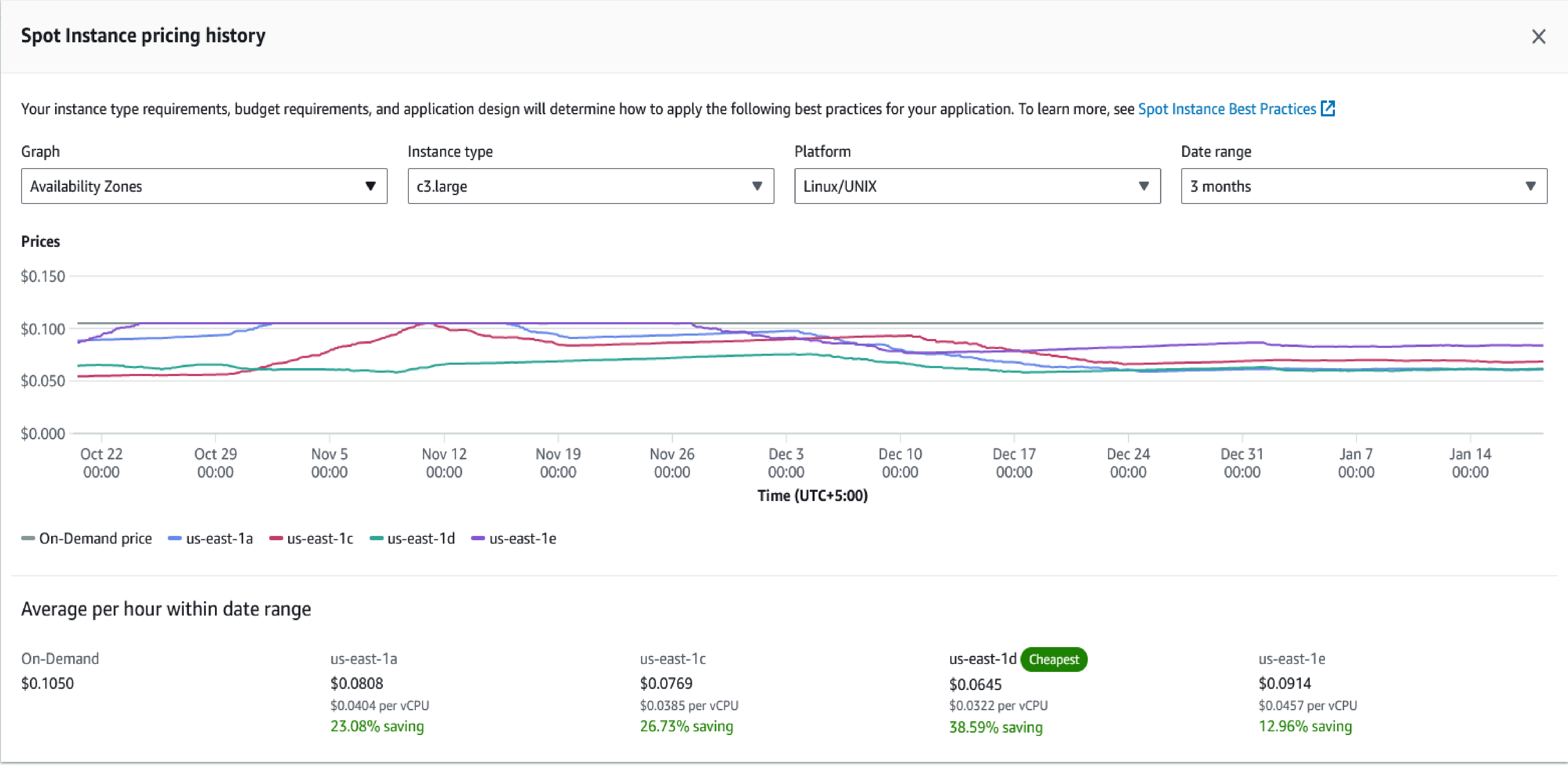
Graph	Instance type	Platform	Date range
Availability Zones	c3.large	Linux/UNIX	1 week

Prices

Average per hour within date range

On-Demand	us-east-1a	us-east-1c	us-east-1d Cheapest	us-east-1e
$0.1050	$0.0612	$0.0687	$0.0610	$0.0838
	$0.0306 per vCPU	$0.0343 per vCPU	$0.0305 per vCPU	$0.0419 per vCPU
	41.75% saving	34.61% saving	41.91% saving	20.19% saving

EC2 Spot Instances Pricing



Spot Instance pricing history

Your instance type requirements, budget requirements, and application design will determine how to apply the following best practices for your application. To learn more, see Spot Instance Best Practices

Graph: Availability Zones ▼

Instance type: c3.large ▼

Platform: Linux/UNIX ▼

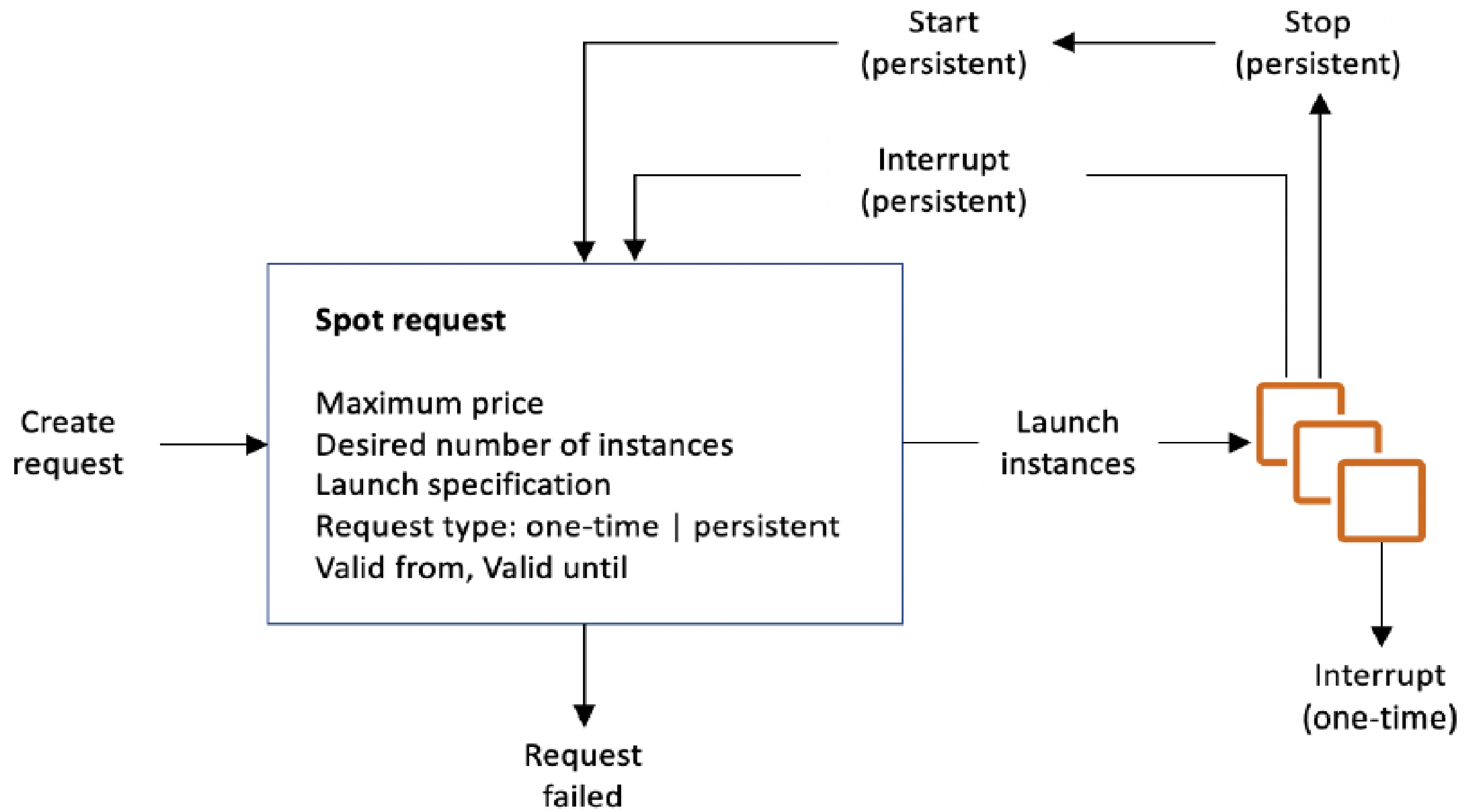
Date range: 3 months ▼

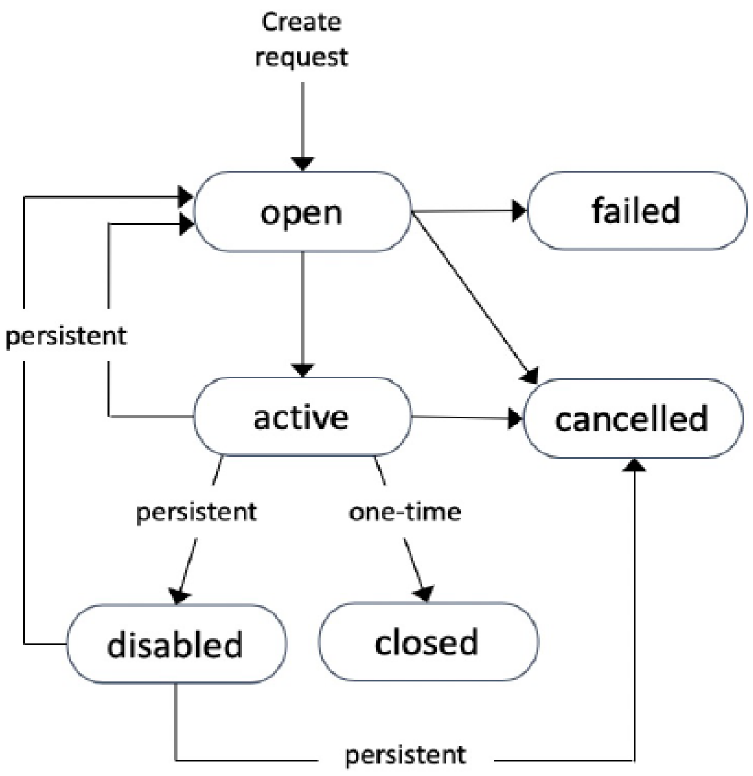
Prices

Average per hour within date range

On-Demand	us-east-1a	us-east-1c	us-east-1d Cheapest	us-east-1e
$0.1050	$0.0808	$0.0769	$0.0645	$0.0914
	$0.0404 per vCPU	$0.0385 per vCPU	$0.0322 per vCPU	$0.0457 per vCPU
	23.08% saving	26.73% saving	38.59% saving	12.96% saving

How to Terminate Spot Instances?





- You can only cancel Spot Instance requests that are **open, active, or disabled.**
- Cancelling a Spot Request does not terminate instances
- You must first cancel a Spot Request, and then terminate the associated Spot Instances

Amazon EC2 - Associate

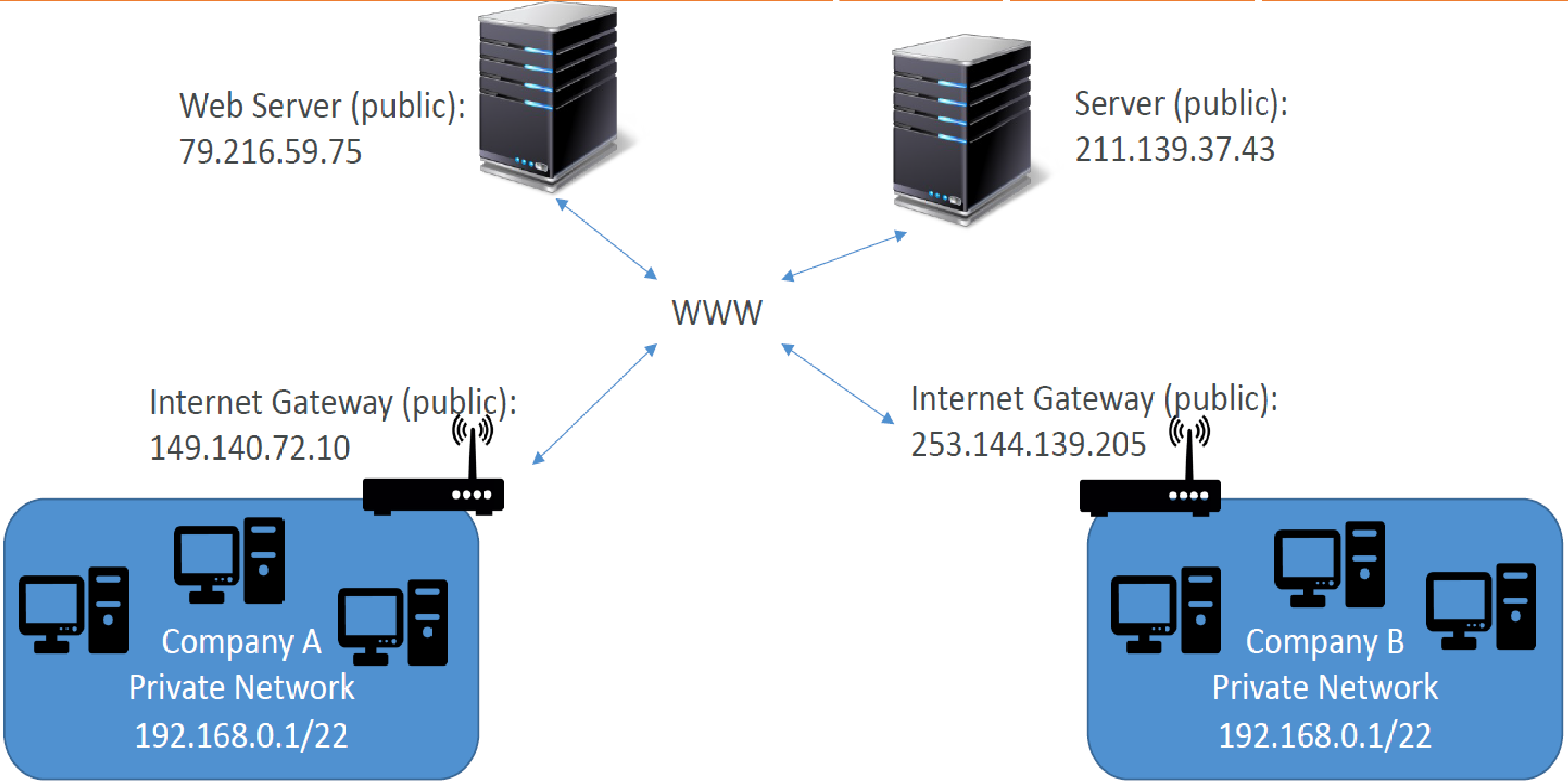
Private vs Public IP (IPv4)

- Networking has two sorts of IPs. IPv4 and IPv6:
  - IPv4: **1.160.10.240**
  - IPv6: **2006:2100:5451:8:200:f8ff:fe27:94da**

- In this course, we will only be using IPv4.
- IPv4 is still the most common format used online.
- IPv6 is newer and solves problems for the Internet of Things (IoT).

- IPv4 allows for 3.7 billion different addresses in the public space
- IPv4: [0-255].[0-255].[0-255].[0-255

Private vs Public (IPv4) Example



Web Server (public):
79.216.59.75

Server (public):
211.139.37.43

WWW

Internet Gateway (public):
149.140.72.10

Internet Gateway (public):
253.144.139.205

Company A
Private Network
192.168.0.1/22

Company B
Private Network
192.168.0.1/22

Private vs Public (IPv4) Fundamental Differences

- Public IP:
  - Public IP means the machine can be identified on the internet (WWW)
  - Must be unique across the whole web (not two machines can have the same public IP).
  - Can be geo-located easily
- Private IP:
  - Private IP means the machine can only be identified on a private network only
  - The IP must be unique across the private network
  - BUT two different private networks (two companies) can have the same IPs.
  - Machines connect to WWW using a NAT + internet gateway (a proxy)
  - Only a specified range of IPs can be used as private IP

Elastic IPs

- When you stop and then start an EC2 instance, it can change its public IP.
- If you need to have a fixed public IP for your instance, you need an Elastic IP
- An Elastic IP is a public IPv4 IP you own as long as you don’t delete it
- You can attach it to one instance at a time

Elastic IPs

- With an Elastic IP address, you can mask the failure of an instance or software by rapidly remapping the address to another instance in your account.
- You can only have 5 Elastic IP in your account (you can ask AWS to increase that).
- Overall, **try to avoid using Elastic IP:**
  - They often reflect poor architectural decisions
  - Instead, use a random public IP and register a DNS name to it
  - Or, as we'll see later, use a Load Balancer and don't use a public IP

Private vs Public IP(IPv4) In AWS EC2 – Hands On

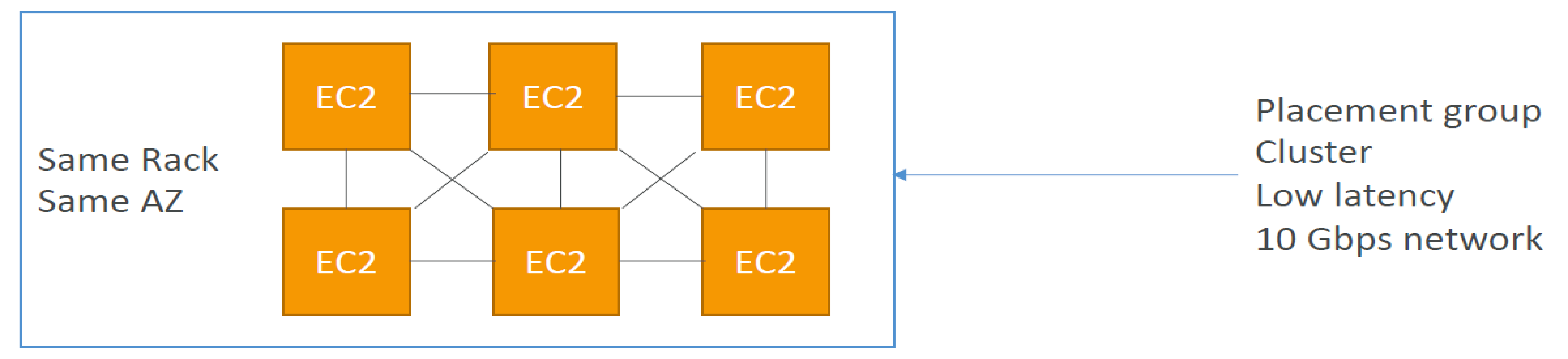
- By default, your EC2 machine comes with:
  - A private IP for the internal AWS Network
  - A public IP, for the WWW.
- When we are doing SSH into our EC2 machines:
  - We can't use a private IP, because we are not in the same network
  - We can only use the public IP.
- If your machine is stopped and then started, **the public IP can change**

Placement Groups

- Sometimes you want control over the EC2 Instance placement strategy
- That strategy can be defined using placement groups
- When you create a placement group, you specify one of the following strategies for the group:
    - ***Cluster***—clusters instances into a low-latency group in a single Availability Zone
    - ***Spread***—spreads instances across underlying hardware (max 7 instances per group per AZ)
    - ***Partition***—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)

Placement Groups

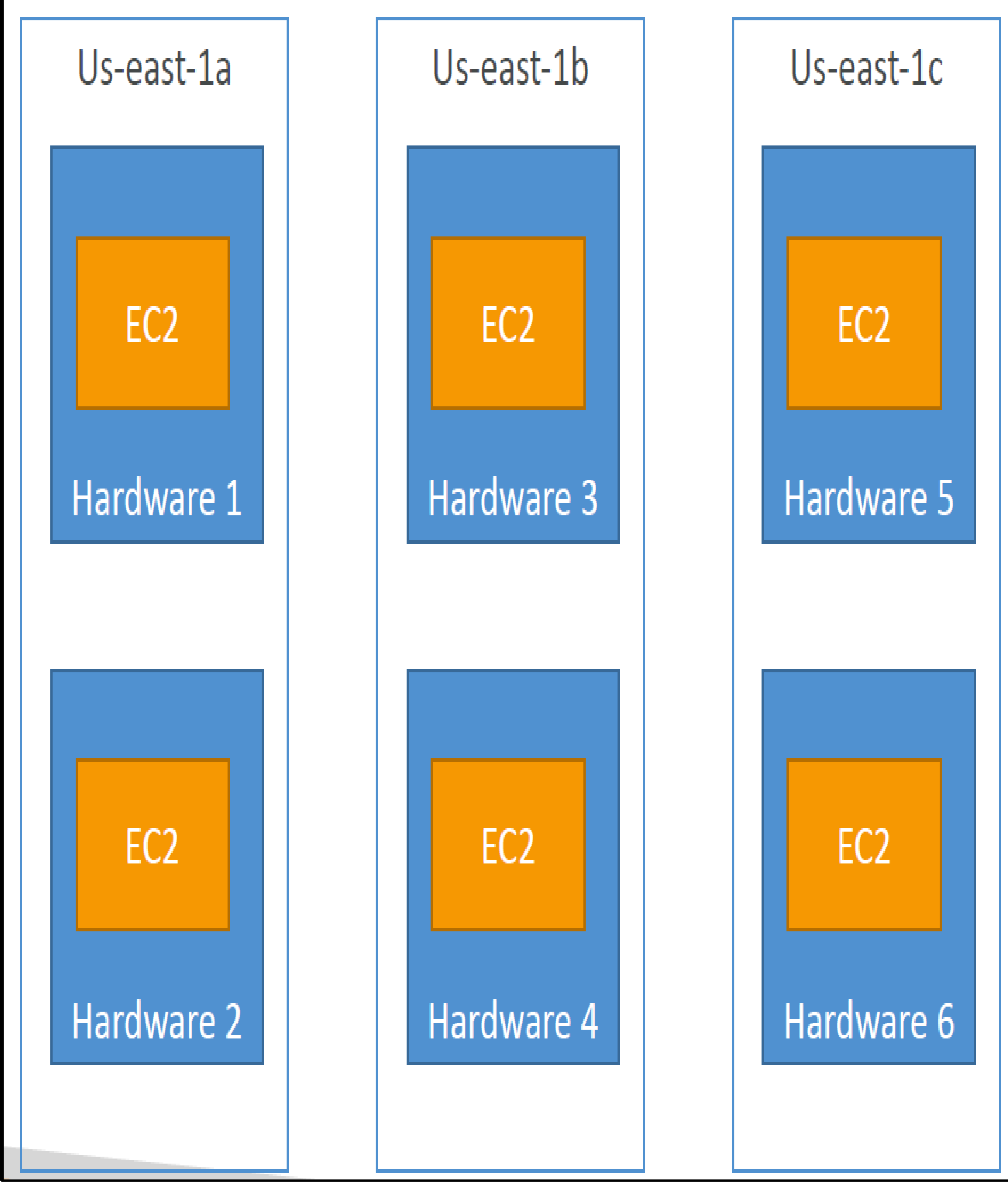
- Cluster



- Pros: Great network (10 Gbps bandwidth between instances with Enhanced Networking enabled - recommended)
- Cons: If the rack fails, all instances fails at the same time
- Use case:
  - Big Data job that needs to complete fast
  - Application that needs extremely low latency and high network throughput

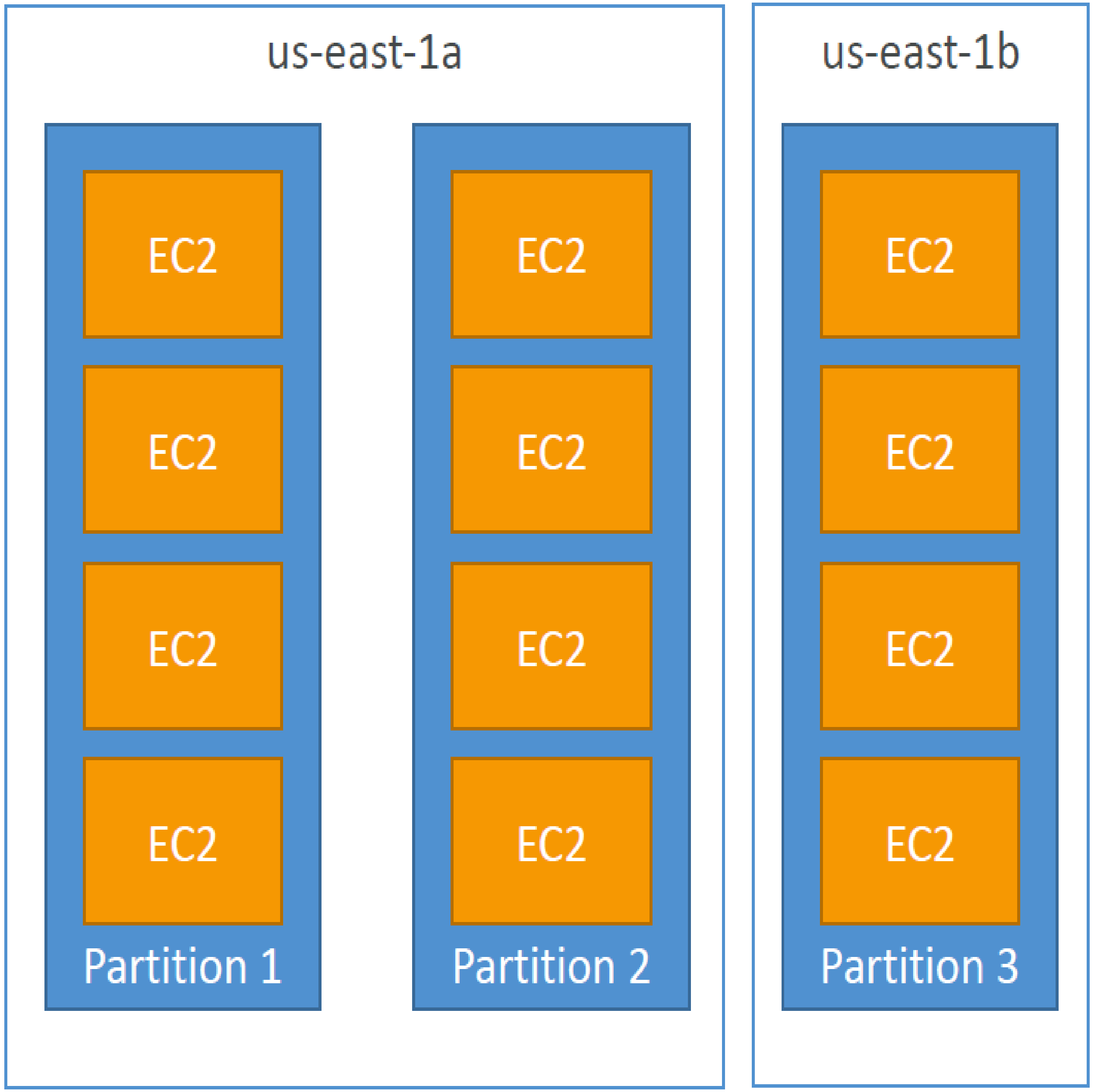
Placement Groups

- Spread
- **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



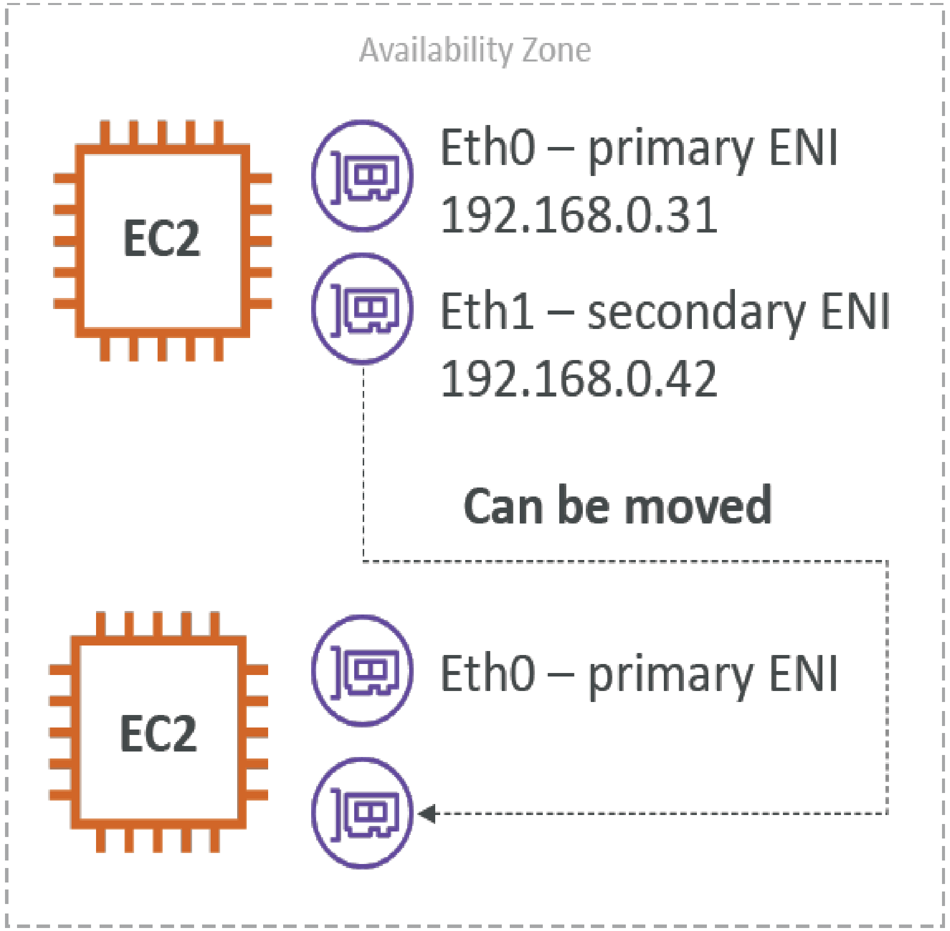
Placement Groups [Partition]

- Up to 7 partitions per AZ
- Can span across multiple AZs in the same region
- 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
- EC2 instances get access to the partition information as metadata
- Use cases: HDFS, HBase, Cassandra, Kafka



Elastic Network Interfaces (ENI)

- Logical component in a VPC that represents a **virtual network card**
- The ENI can have the following attributes:
  - Primary private IPv4, one or more secondary IPv4
  - One Elastic IP (IPv4) per private IPv4
  - One Public IPv4
  - One or more security groups
  - A MAC address
- You can create ENI independently and attach them on the fly (move them) on EC2 instances for failover
- Bound to a specific availability zone (AZ)

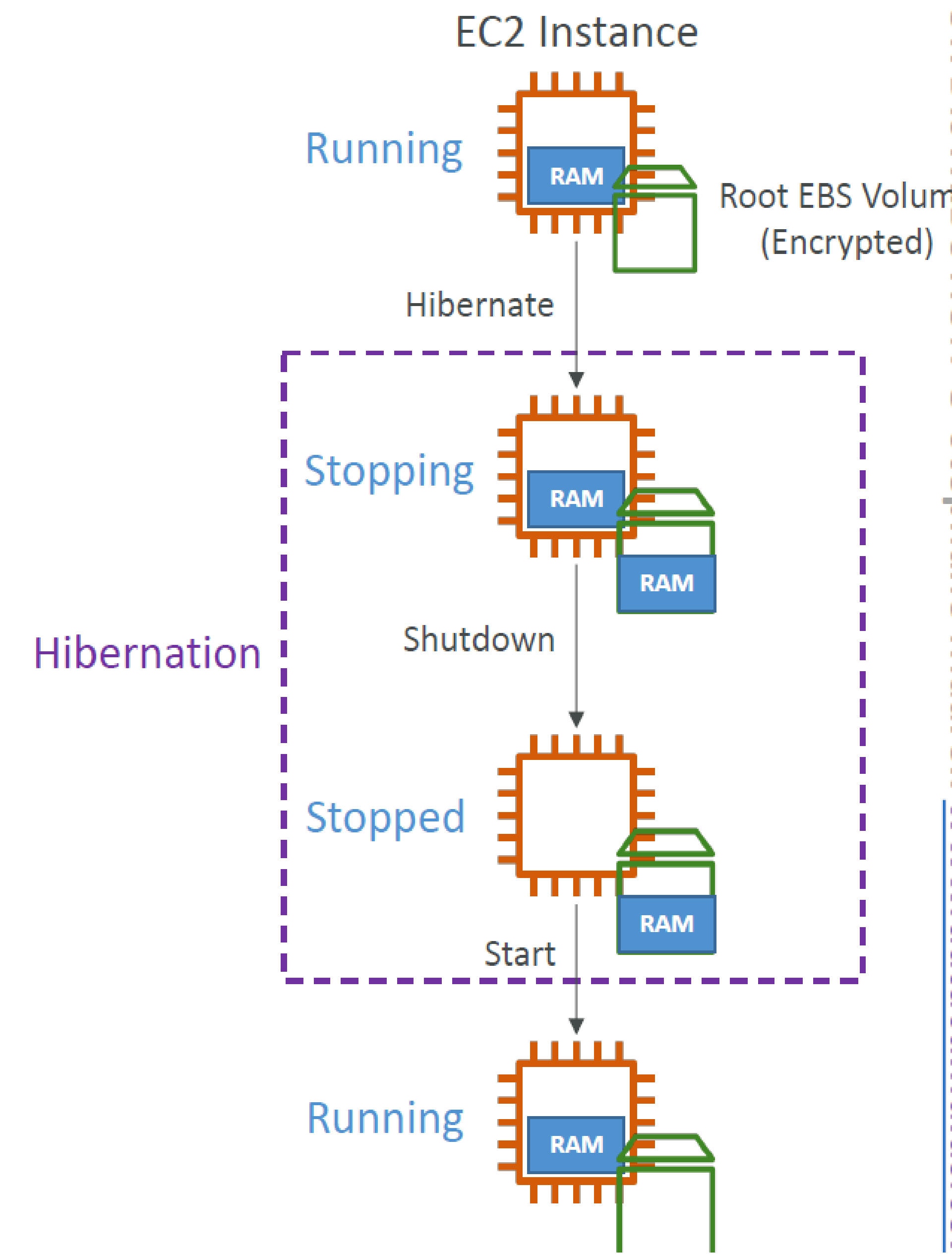


EC2 Hibernate

- We know we can stop, terminate instances
  - **Stop** – the data on disk (EBS) is kept intact in the next start
  - **Terminate** – any EBS volumes (root) also set-up to be destroyed is lost
- On start, the following happens:
  - First start: the OS boots & the EC2 User Data script is run
  - Following starts: the OS boots up
  - Then your application starts, caches get warmed up, and that can take time!

EC2 Hibernate

- Introducing **EC2 Hibernate:**
  - The in-memory (RAM) state is preserved
  - The instance boot is much faster! (the OS is not stopped / restarted)
  - Under the hood: the RAM state is written to a file in the root EBS volume
  - The root EBS volume must be encrypted
- **Use cases:**
  - Long-running processing
  - Saving the RAM state
  - Services that take time to initialize



EC2 Hibernate – Good to Know

- **Supported Instance Families** – C3, C4, C5, I3, M3, M4, R3, R4, T2, T3, …
- **Instance RAM Size** – must be less than 150 GB.
- **Instance Size** – not supported for bare metal instances.
- **AMI** – Amazon Linux 2, Linux AMI, Ubuntu, RHEL, CentOS & Windows…
- **Root Volume** – must be EBS, encrypted, not instance store, and large
- Available for **On-Demand, Reserved** and **Spot** Instances
- An instance can **NOT** be hibernated more than 60 days

Amazon EC2 – Instance Storage

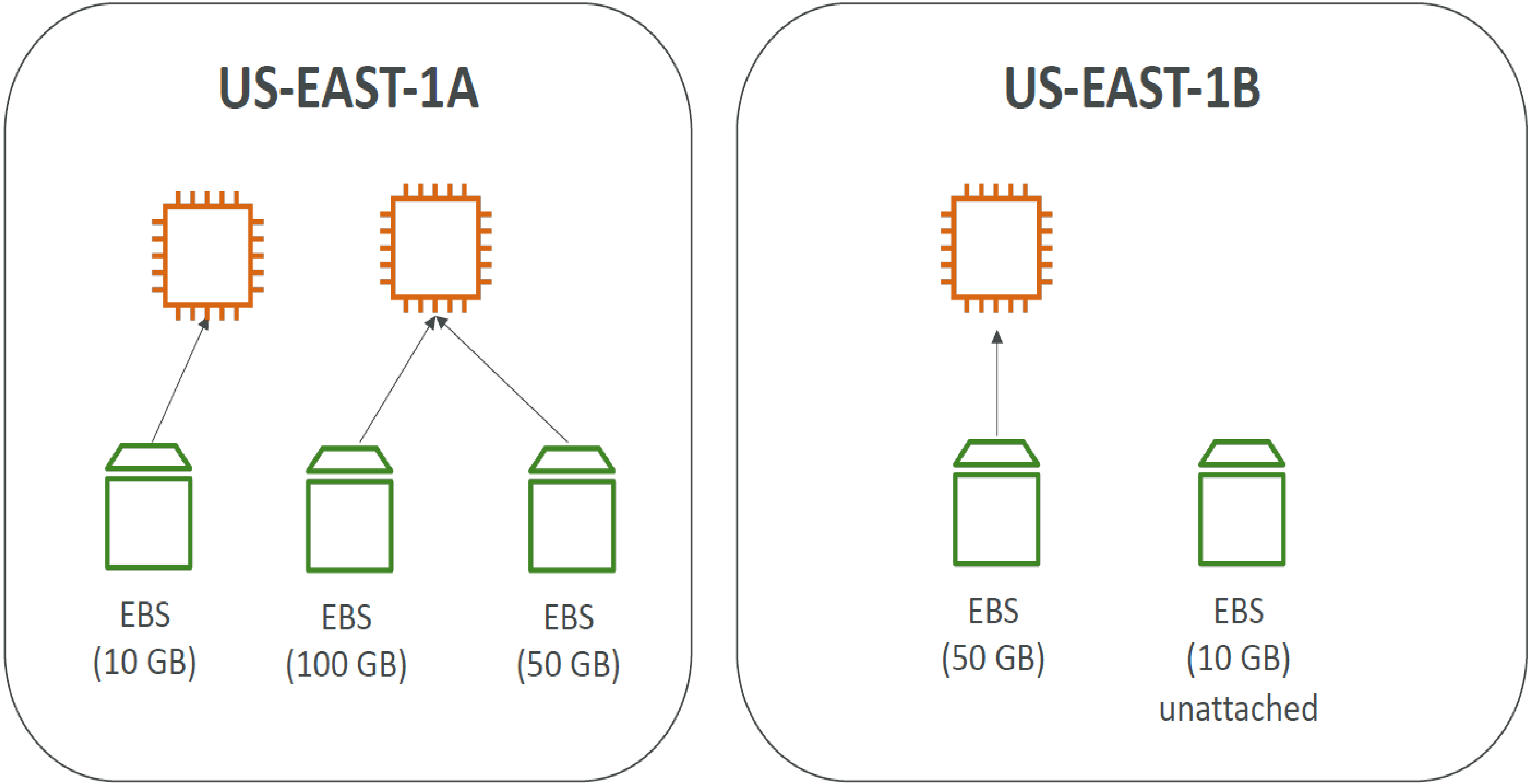
What is an EBS Volume?

- 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** •
- 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

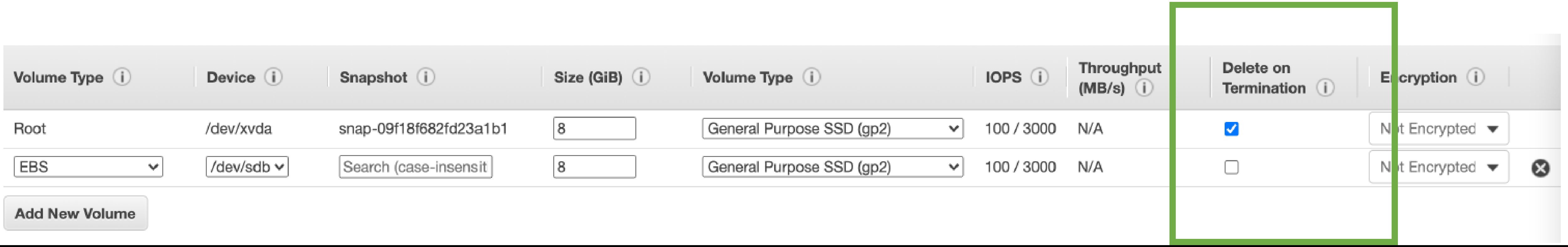
EBS Volume

- **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 - Example



EBS – Delete on Termination Attribute

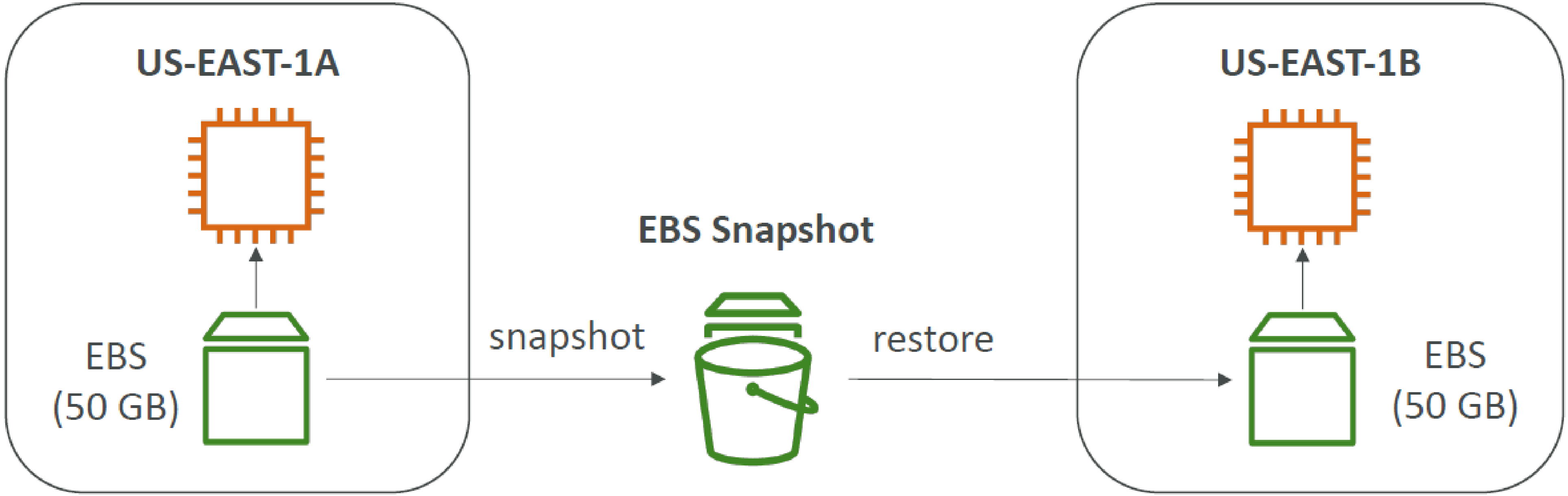


Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/xvda	snap-09f18f682fd23a1b1	8	General Purpose SSD (gp2)	100 / 3000	N/A	☑	Not Encrypted
EBS	/dev/sdb	Search (case-insensit	8	General Purpose SSD (gp2)	100 / 3000	N/A	☐	Not Encrypted

- Controls the EBS behavior 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**

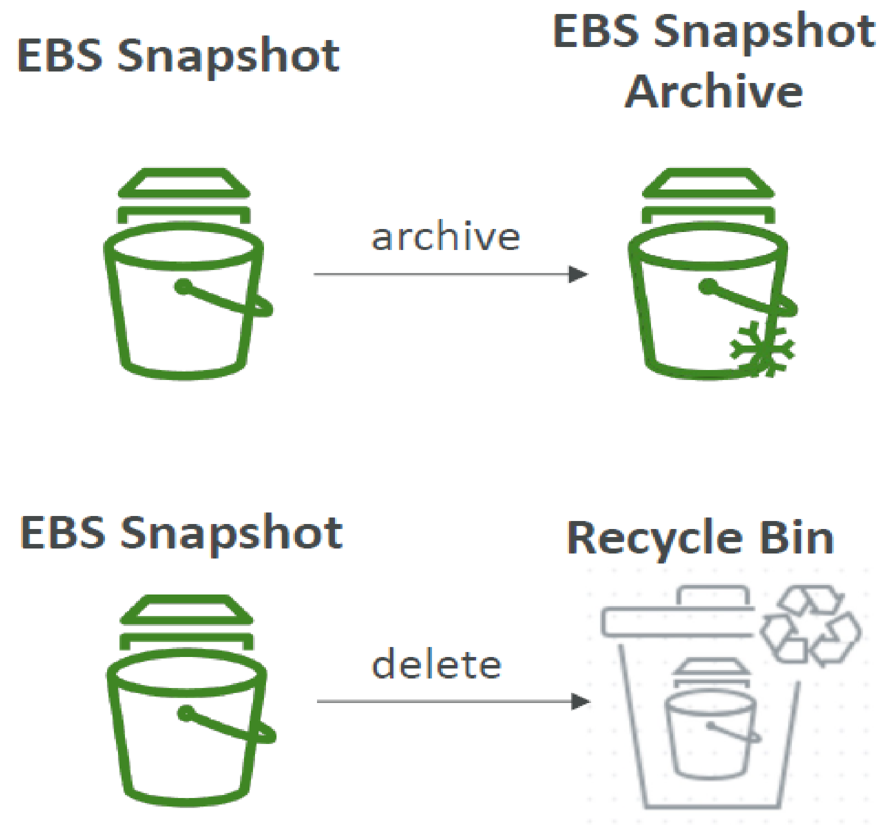
EBS Snapshots

- 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)
- **Fast Snapshot Restore (FSR)**
  - Force full initialization of snapshot to have no latency on the first use ($$$)

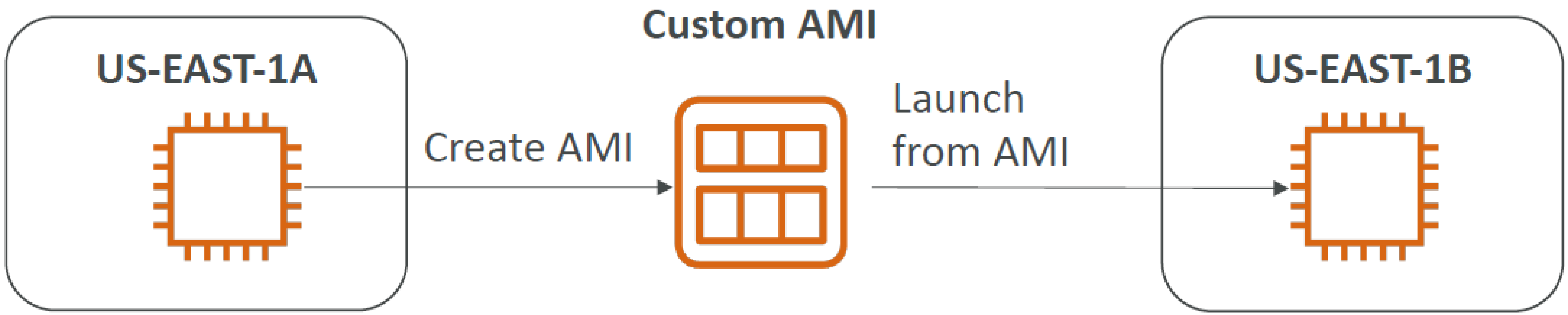


AMI Overview

- 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
- `



EC2 Instance Store

- EBS volumes are **network drives** with good but “limited” performance
- **If you need a high-performance hardware disk, use EC2 Instance Store**

- Better I/O performance
- EC2 Instance Store lose their storage if they’re stopped (ephemeral)
- Good for buffer / cache / scratch data / temporary content
- Risk of data loss if hardware fails
- Backups and Replication are your responsibility

EBS Volume Types

- EBS Volumes come in 6 types
  - **gp2 / gp3 (SSD):** General purpose SSD volume that balances price and performance for a wide variety of workloads
  - **io1 / io2 (SSD):** Highest-performance SSD volume for mission-critical low-latency or high-throughput workloads
  - **st1 (HDD):** Low-cost HDD volume designed for frequently accessed, throughput intensive workloads
  - **sc1 (HDD):** Lowest cost HDD volume designed for less frequently accessed workloads

- EBS Volumes are characterized in Size | Throughput | IOPS (I/O Ops Per Sec)
- When in doubt always consult the AWS documentation – it's good!
- **Only gp2/gp3 and io1/io2 can be used as boot volumes**

EBS Volume Types Use cases

General Purpose SSD

- Cost effective storage, low-latency
- System boot volumes, Virtual desktops, Development and test environments
- 1 GiB - 16 TiB
- **gp3:**
  - Baseline of 3,000 IOPS and throughput of 125 MiB/s
  - Can increase IOPS up to 16,000 and throughput up to 1000 MiB/s independently
- **gp2:**
  - Small gp2 volumes can burst IOPS to 3,000
  - Size of the volume and IOPS are linked, max IOPS is 16,000
  - 3 IOPS per GB, means at 5,334 GB we are at the max IOPS

EBS Volume Types Use cases

Provisioned IOPS (PIOPS) SSD

- Critical business applications with sustained IOPS performance
- Or applications that need more than 16,000 IOPS
- Great for databases workloads (sensitive to storage perf and consistency)
- io1/io2 (4 GiB - 16 TiB):
  - Max PIOPS: 64,000 for Nitro EC2 instances & 32,000 for other
  - Can increase PIOPS independently from storage size
  - io2 have more durability and more IOPS per GiB (at the same price as io1)
- io2 Block Express (4 GiB – 64 TiB):
  - Sub-millisecond latency
  - Max PIOPS: 256,000 with an IOPS: GiB ratio of 1,000:1
- Supports EBS Multi-attach

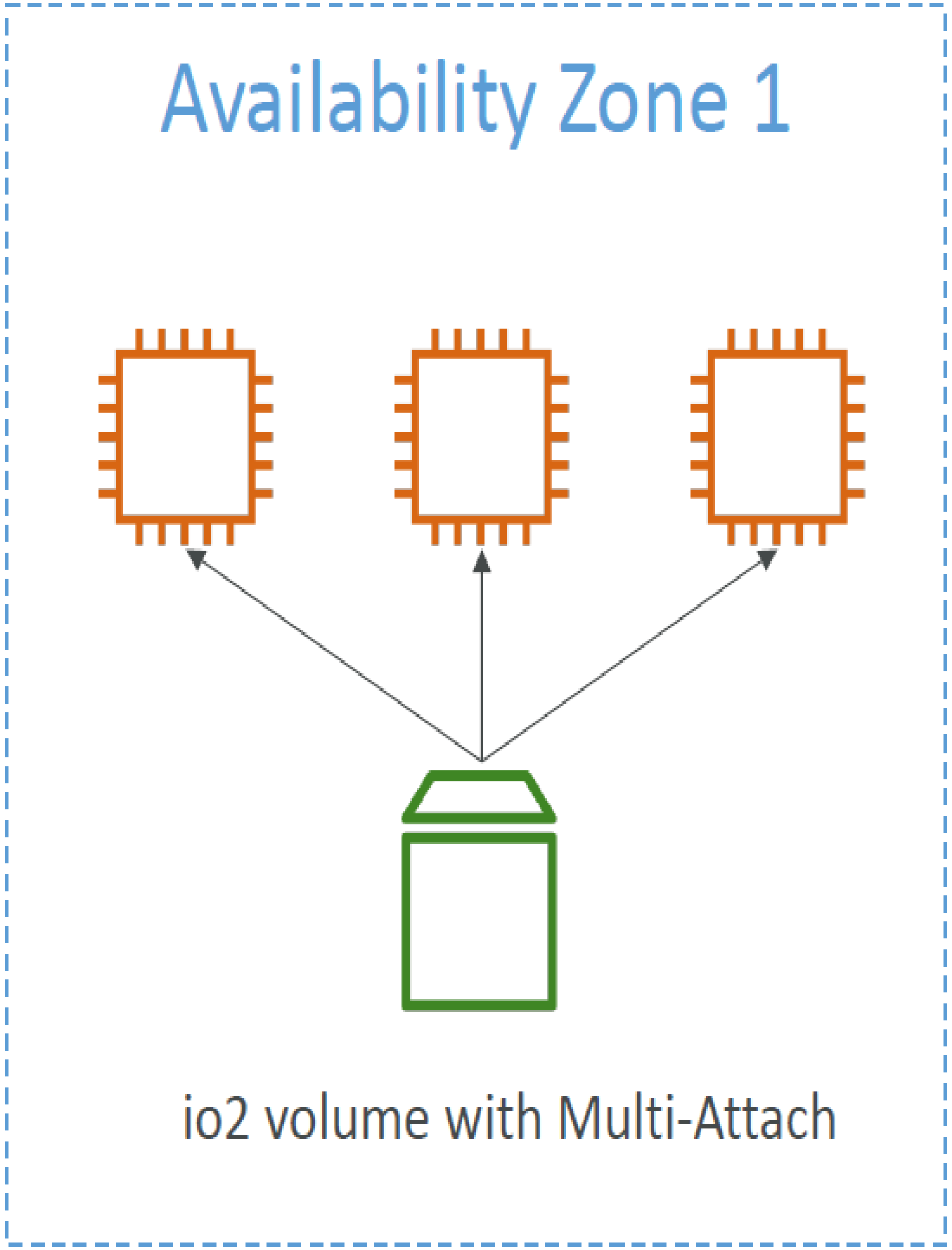
EBS Volume Types Use cases

Hard Disk Drives (HDD)

- Cannot be a boot volume
- 125 GiB to 16 TiB
- Throughput Optimized HDD (st1)
  - Big Data, Data Warehouses, Log Processing
  - **Max throughput** 500 MiB/s – max IOPS 500
- **Cold HDD (sc1):**
  - For data that is infrequently accessed
  - Scenarios where lowest cost is important
  - Max throughput 250 MiB/s – max IOPS 250

EBS Multi-Attach – io1/io2 Family

- Attach the same EBS volume to multiple EC2 instances in the same AZ
- Each instance has full read & write permissions to the high-performance volume
- Use case:
  - Achieve **higher application availability** in clustered Linux applications (ex: Teradata)
  - Applications must manage concurrent write operations
- **Up to 16 EC2 Instances at a time**
- Must use a file system that's cluster-aware (not XFS, EXT4, etc.…)



EBS Encryption

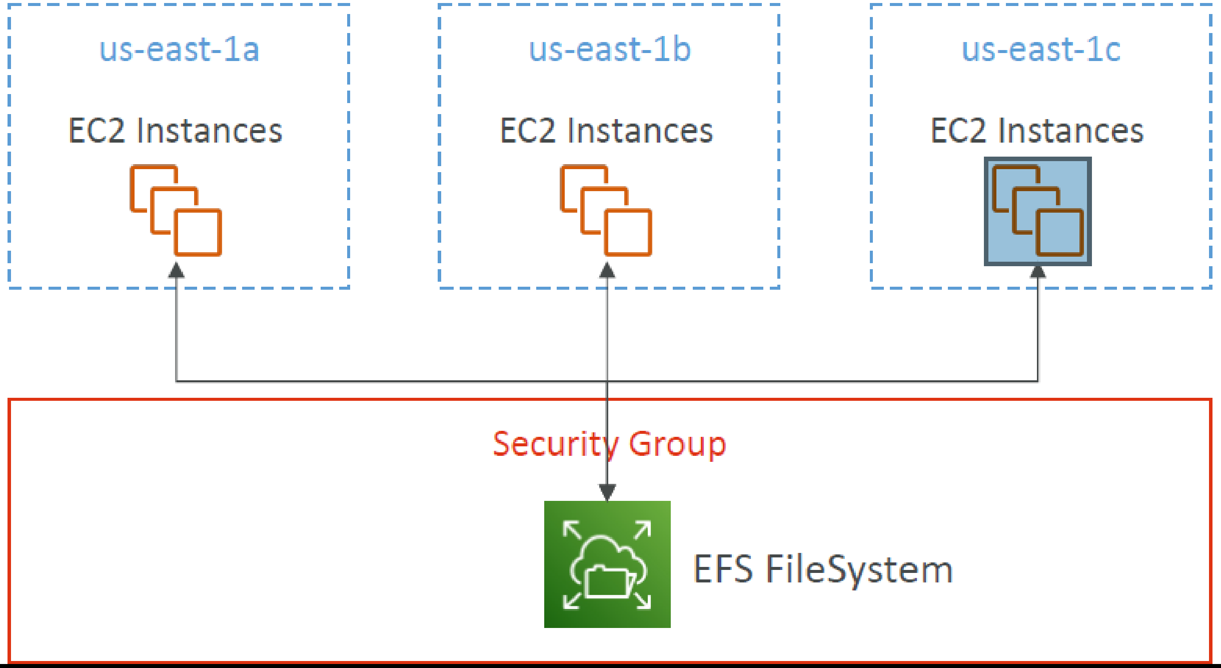
- When you create an encrypted EBS volume, you get the following:
  - Data at rest is encrypted inside the volume
  - All the data in flight moving between the instance and the volume is encrypted
  - All snapshots are encrypted
  - All volumes created from the snapshot
- Encryption and decryption are handled transparently (you have nothing to do)
- Encryption has a minimal impact on latency
- EBS Encryption leverages keys from KMS (AES-256)
- Copying an unencrypted snapshot allows encryption
- Snapshots of encrypted volumes are encrypted

Encryption: Encrypt an unencrypted EBS Volume

- Create an EBS snapshot of the volume
- Encrypt the EBS snapshot (using copy)
- Create new ebs volume from the snapshot (the volume will also be encrypted)
- Now you can attach the encrypted volume to the original instance

Amazon EFS – Elastic File System

- Managed NFS (network file system) that can be mounted on many EC2
- EFS works with EC2 instances in multi-AZ
- Highly available, scalable, expensive (3x gp2), pay per use



Amazon EFS – Elastic File System

- Use cases: content management, web serving, data sharing, WordPress
- Uses NFSv4.1 protocol
- Uses security group to control access to EFS
- **Compatible with Linux based AMI (not Windows)**
- Encryption at rest using KMS

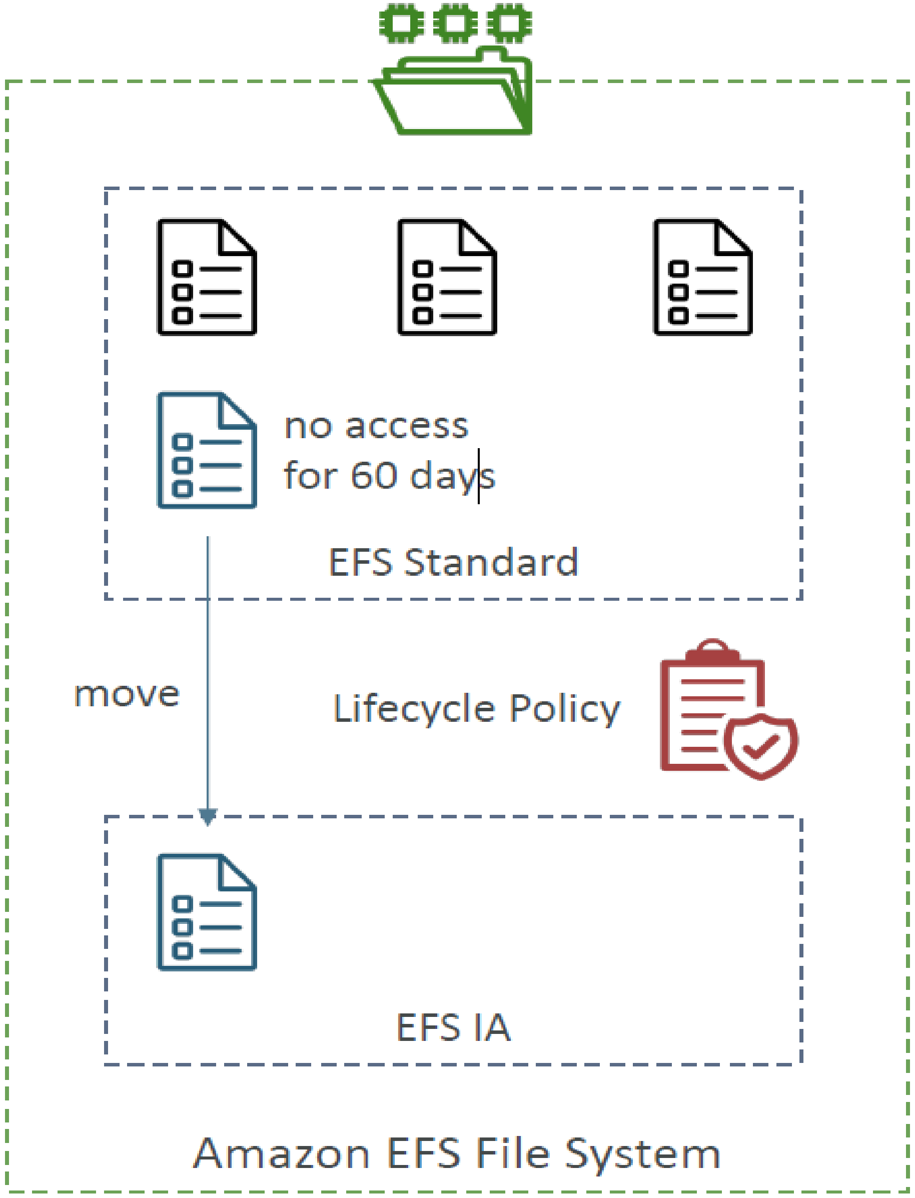
- POSIX file system (~Linux) that has a standard file API
- File system scales automatically, pay-per-use, no capacity planning!

EFS –Storage Classes

- **Storage Tiers (lifecycle management feature move file after N days)**
- **Standard:** for frequently accessed files
- **Infrequent access (EFS-IA):** cost to retrieve files, lower price to store. Enable EFS-IA with a Lifecycle Policy

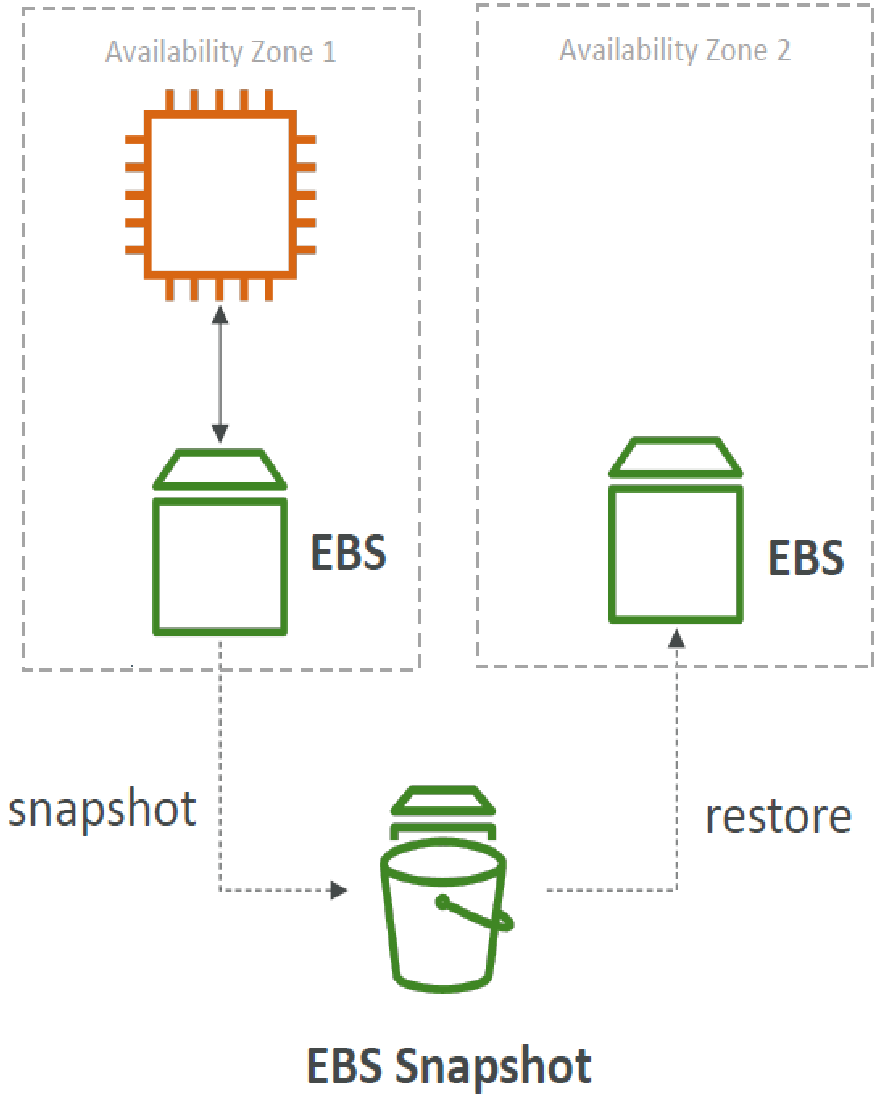
- **Availability and durability**
  - **Standard**: Multi-AZ, great for prod
  - **One Zone:** One AZ, great for dev, backup enabled by default, compatible with IA (EFS One Zone-IA)

- Over 90% in cost savings



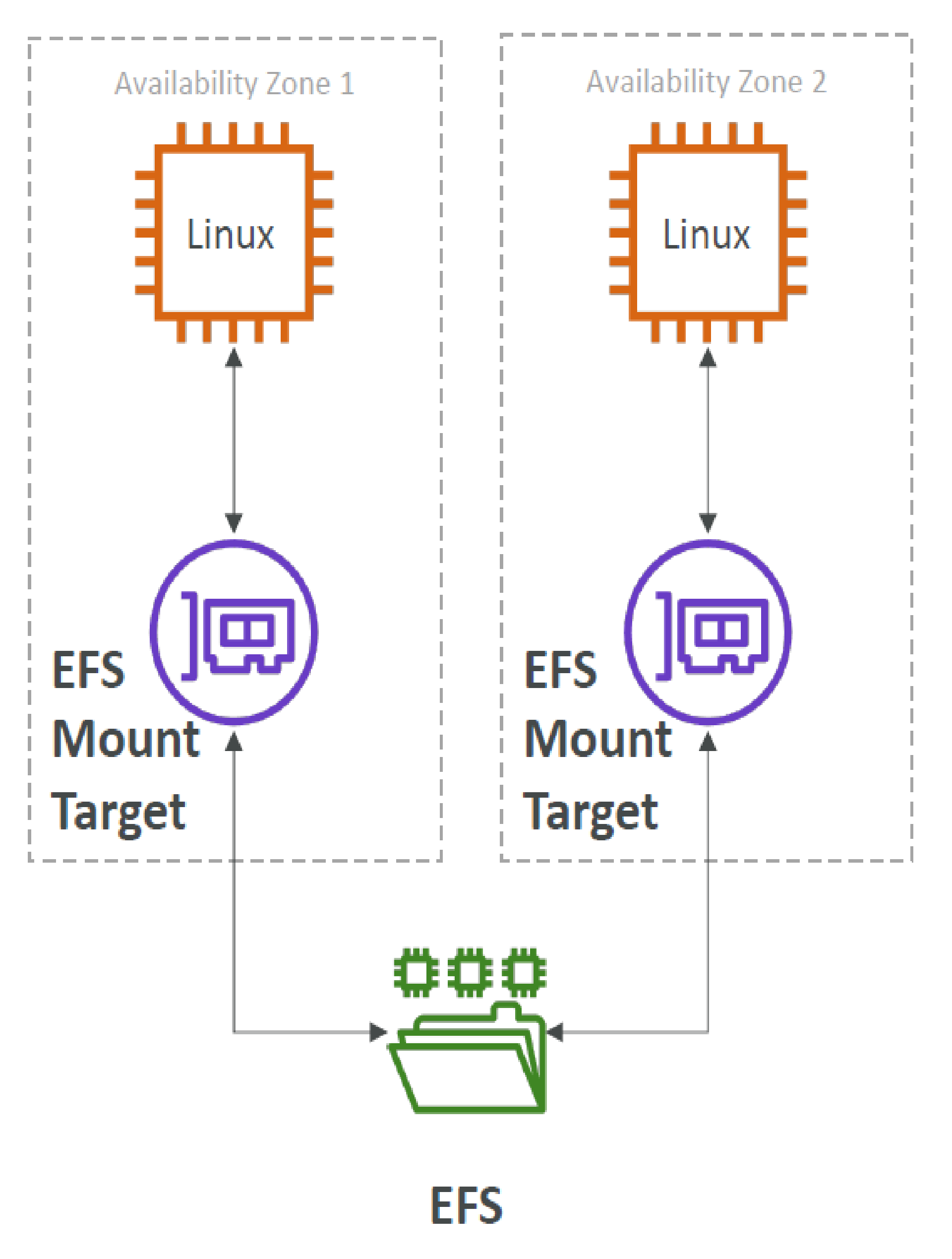
EBS vs EFS – Elastic Block Storage

- **EBS volumes…**
  - one instance (except multi-attach io1/io2)
  - are locked at the Availability Zone (AZ) level
  - gp2: IO increases if the disk size increases
  - gp3 & io1: can increase IO independently
- **To migrate an EBS volume across AZ**
  - Take a snapshot
  - Restore the snapshot to another AZ
  - EBS backups use IO and you shouldn't run them while your application is handling a lot of traffic
- Root EBS Volumes of instances get terminated by default if the EC2 instance gets terminated. (you can disable that)



EBS vs EFS – Elastic File System

- Mounting 100s of instances across AZ
- EFS share website files (WordPress)
- Only for Linux Instances (POSIX)
- EFS has a higher price point than EBS
- Can leverage EFS-IA for cost savings
- Remember: EFS vs EBS vs Instance Store


Availability Zone 1
Availability Zone 2
Linux
Linux
EFS Mount Target
EFS Mount Target
EFS