# COMP8062 - Cloud Automation & Orchestration

Dr. David Stynes

# Commercial Cloud Architectures: AWS

Dr. David Stynes

# Regions vs Availability Zones

Highlight the difference between physical and virtual cloud connection points:

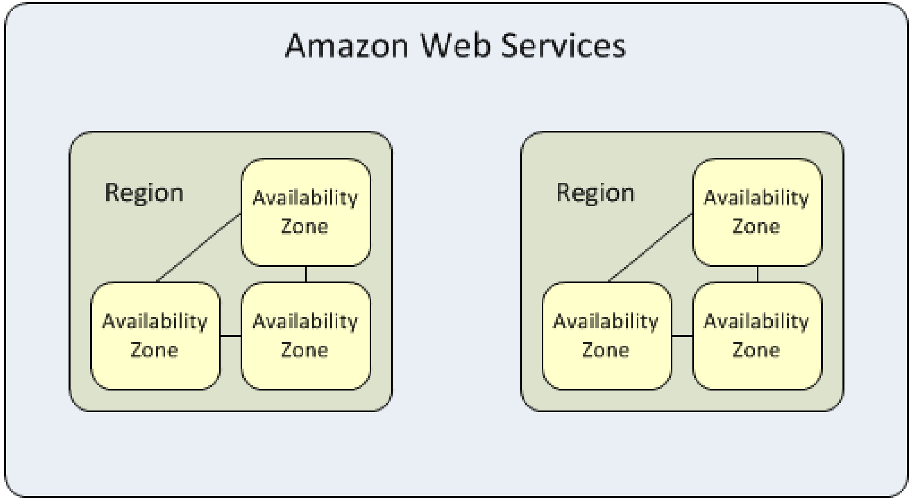
The AWS Cloud operates 27 geographic Regions containing 87 Availability Zones around the world.

8 new Regions and  24 Availability Zones are coming soon!

A region is a physical location in a particular geographic region. You typically host your data in the one closest to you to reduce latency.

An Availability Zone (AZ) is one or more data centres within a region

- Minimum 2 availability zones per region
- Dublin has 3 availability zones.



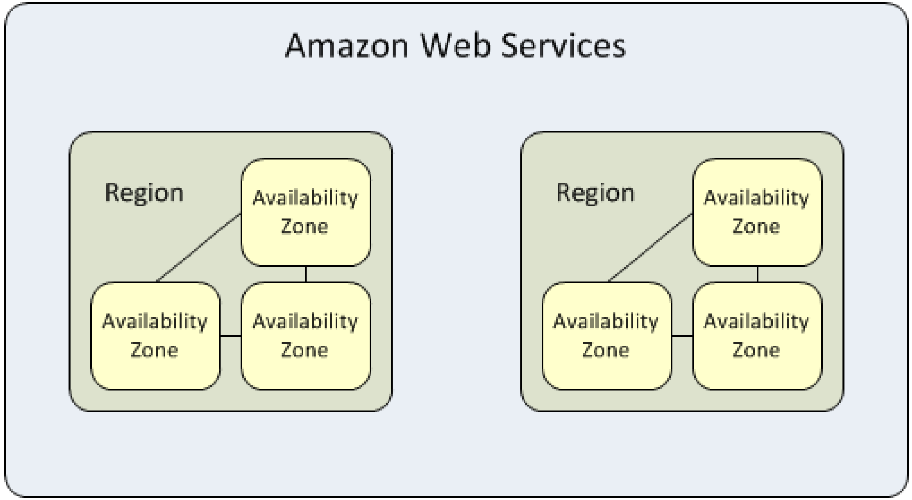
# Regions vs Availability Zones

Each Amazon EC2 region is designed to be completely isolated from the other Amazon EC2 regions. This achieves the greatest possible fault tolerance and stability.

When you view your resources, you'll only see the resources tied to the region you've specified. This is because regions are isolated from each other, and resources are not replicated across regions automatically.

When you launch an instance, you must select an AMI that's in the same region (possible to copy the AMI to the region you're using)

All communications between regions is across the public Internet (proper encryption methods should be used to protect data).



# Regions vs Availability Zones

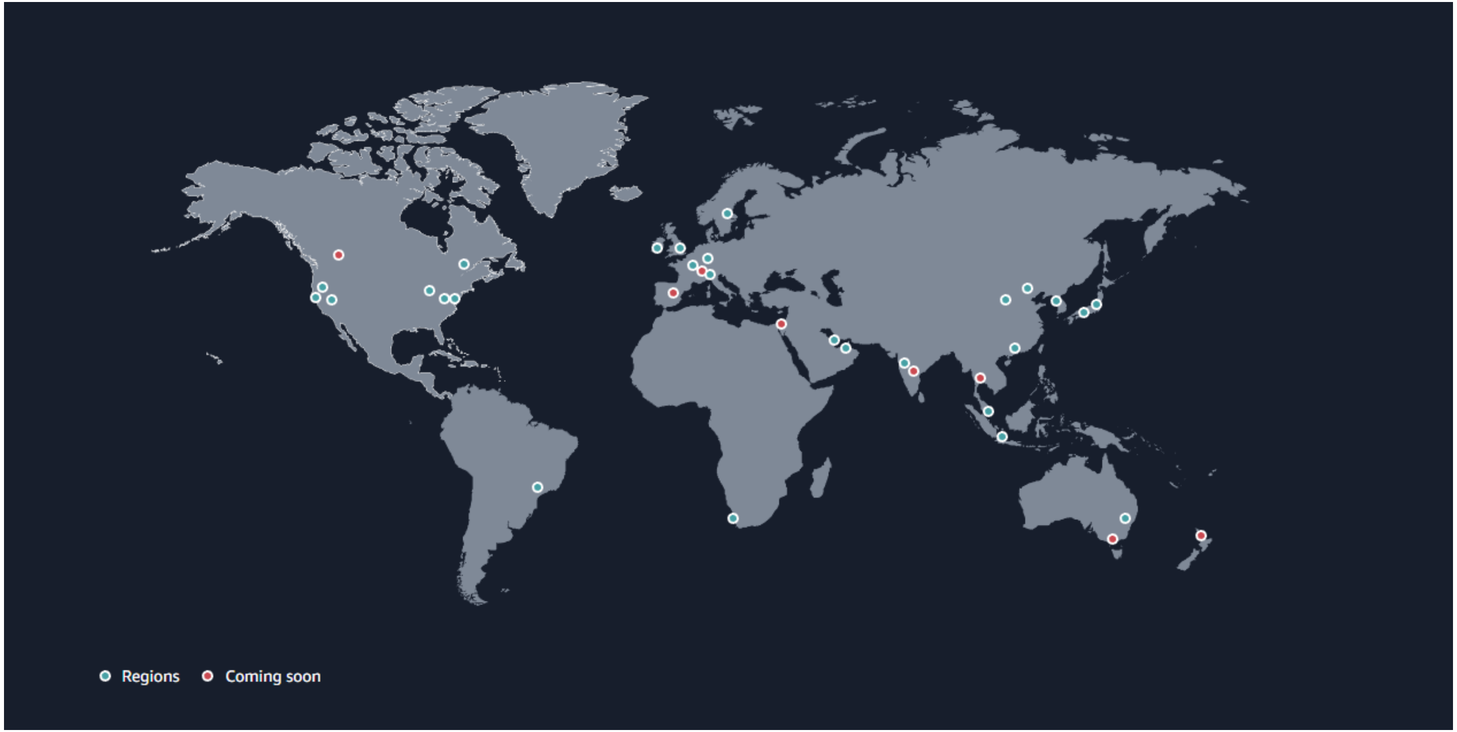
Availability zones need not be separated by multiple kilometres physically but by hundreds of meters within a physical compound which are completed isolated from each others failures with respect to power, networking, flood zones etc.

- This was clarified by AWS as initially little was said about the physical layout of AZs. They have since clarified that each AZ resides in a different building.
- Isolated from each other but close enough for low latency network connections
- These AZs are usually connected with direct fibre optic (latency 1-2ms) vs ~70ms to transfer traffic from NY to LA.

Region for Ireland is eu-west-1 with 3 availability zones (eu-west-1a, eu-west-1b, eu-west-1c).

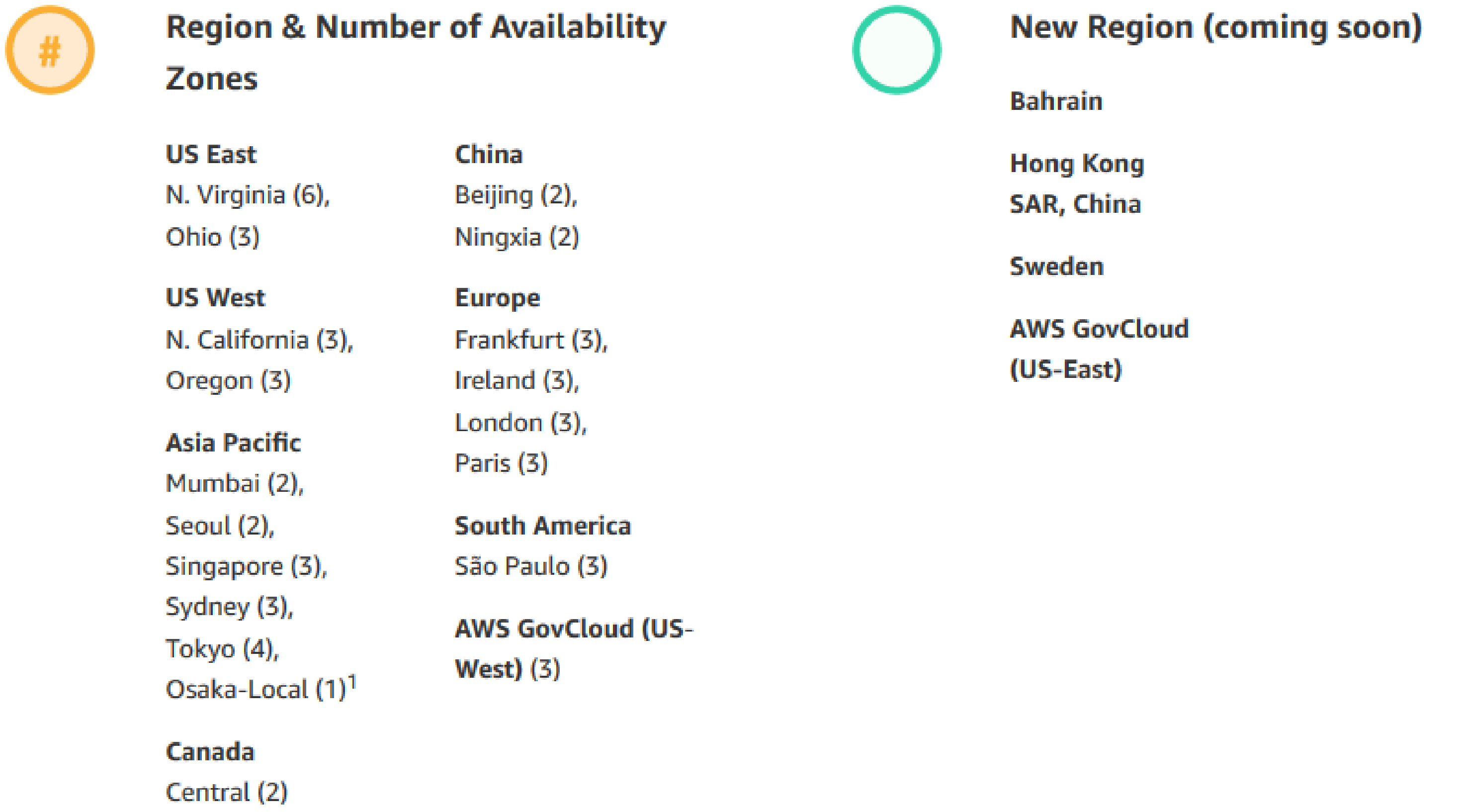
- Some gotcha's in the labs: Be careful of this when invoking the connect_to_region method in Boto3 e.g. eu-west-1 vs eu-west-1a.

# Regions vs Availability Zones



https://aws.amazon.com/about-aws/global-infrastructure/

# Regions vs Availability Zones



## Region & Number of Availability Zones

**US East**
N. Virginia (6),
Ohio (3)

**US West**
N. California (3),
Oregon (3)

**Asia Pacific**
Mumbai (2),
Seoul (2),
Singapore (3),
Sydney (3),
Tokyo (4),
Osaka-Local (1)[1]

**Canada**
Central (2)

**China**
Beijing (2),
Ningxia (2)

**Europe**
Frankfurt (3),
Ireland (3),
London (3),
Paris (3)

**South America**
São Paulo (3)

**AWS GovCloud (US-West)** (3)

## New Region (coming soon)

**Bahrain**

**Hong Kong SAR, China**

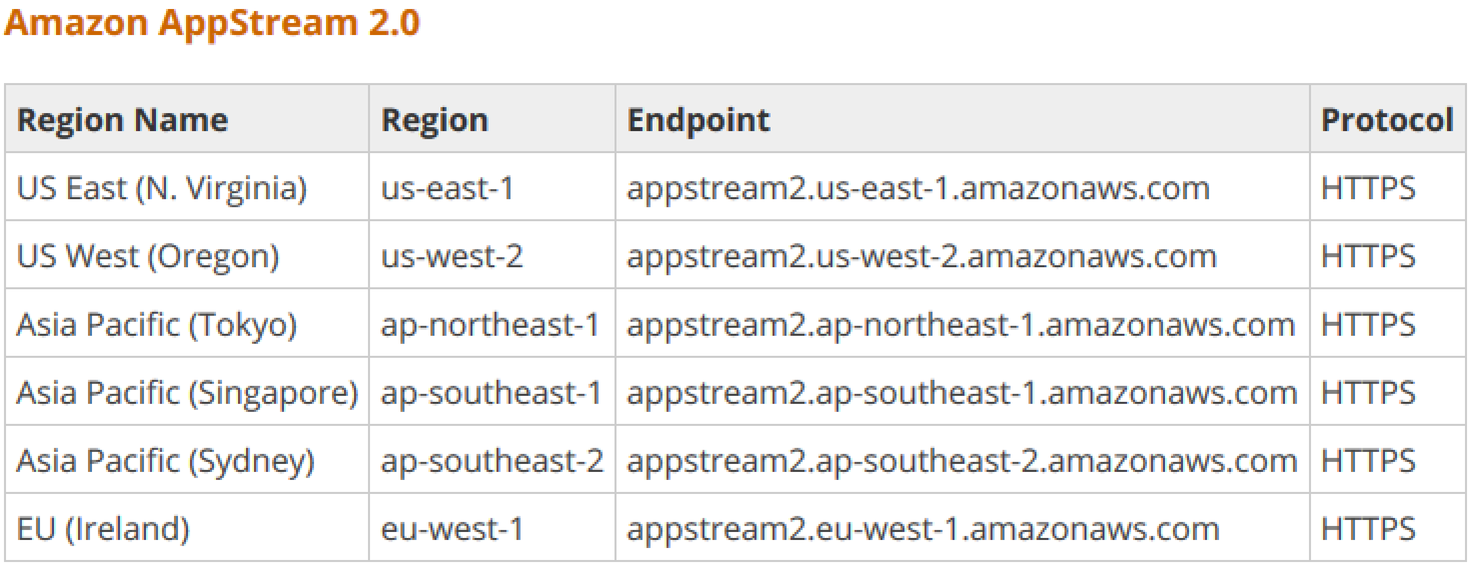
**Sweden**

**AWS GovCloud (US-East)**

# Endpoints

- There is one more concept around, and one that many get confused by: endpoints.
- There are several ways to access an AWS service, region and/or an availability zone and they are called endpoints.
- In other words, they are URLs acting as entry point for a web service. Again, they aim to reduce even further the latency of your applications. Not all the AWS services support endpoints however.
- So endpoints are just the door through which global, regional and AZ-related resources in the AWS world can be accessed.
- For example we will see later how to directly access an S3 resource (s3-eu-west-1.amazonaws.com for the eu-west-1 region) .
- https://docs.aws.amazon.com/general/latest/gr/rande.html

# Endpoints



## Amazon AppStream 2.0

| Region Name | Region | Endpoint | Protocol |
|---|---|---|---|
| US East (N. Virginia) | us-east-1 | appstream2.us-east-1.amazonaws.com | HTTPS |
| US West (Oregon) | us-west-2 | appstream2.us-west-2.amazonaws.com | HTTPS |
| Asia Pacific (Tokyo) | ap-northeast-1 | appstream2.ap-northeast-1.amazonaws.com | HTTPS |
| Asia Pacific (Singapore) | ap-southeast-1 | appstream2.ap-southeast-1.amazonaws.com | HTTPS |
| Asia Pacific (Sydney) | ap-southeast-2 | appstream2.ap-southeast-2.amazonaws.com | HTTPS |
| EU (Ireland) | eu-west-1 | appstream2.eu-west-1.amazonaws.com | HTTPS |

# Why choose a particular region?

1. Cost:
   - Different for each region (labour, taxes, infrastructure etc)
   - E.g. m3.Large EEC2 instance is ~$119 in EU Frankfurt but ~$100 in US Virginia.
   - Use the AWS cost calculator.

2. Latency:
   - Location of users vs data/applications (opt for locality)
   - CloudPing and CloudWatch (AWS service) provide latency of various AWS services in various regions from your browser.
   - Proof of Concept and load testing

3. Security & Compliance:
   - Healthcare, banking, finance – security and compliance rules (customer data moving from one location to another)
   - Many cloud services need to receive certs of compliance per country before they can be used.

# Amazon Machine Images (AMIs) vs Instances

An AMI is a template that contains a software configuration (e.g. OS, application server & apps)

Public AMIs:

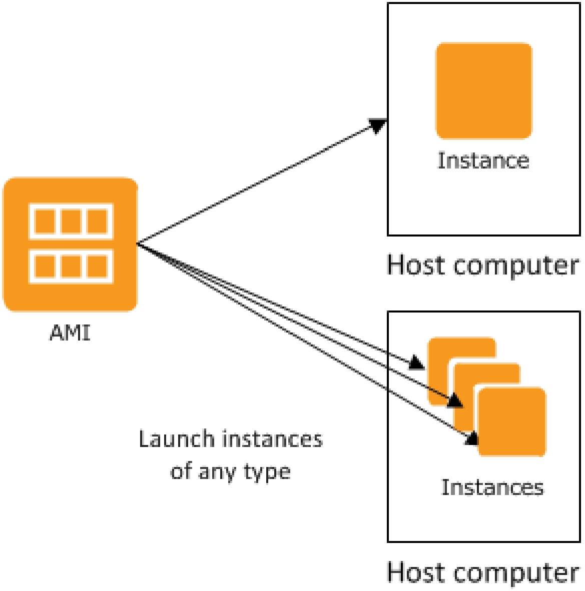
- Use pre-configured template.
- AMIs are up and running immediately
- Choose from Fedora, Ubuntu configurations and more

Private AMIs:

- Create an AMI containing your applications, libraries, data and associated configuration settings

Paid AMIs:

- Set a price for your AMI and let others purchase and use it (single payment or per hour).
- E.g. AMIs with a commercial DBMS



# Amazon Machine Images (AMIs) vs Instances

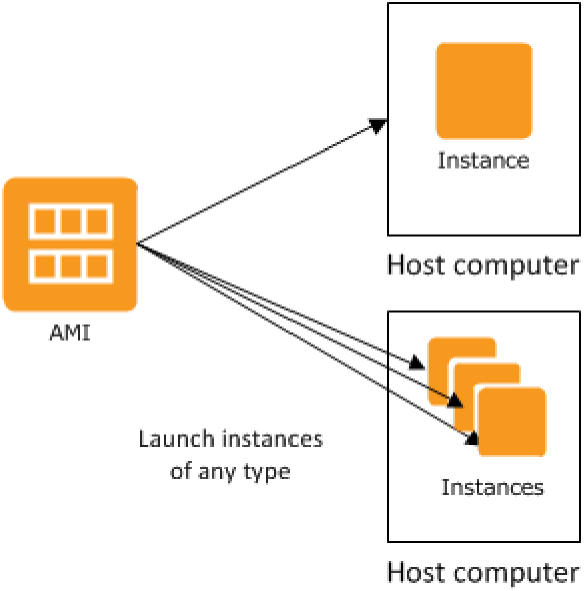
- From an AMI, you can launch 1 or more instances which is a copy of the AMI running in the virtual cloud

- An instance type specifies the hardware of the host computer used for your instance.

- Choose based on the amount of memory or compute power you need.

- Remote access to your machine

× Windows (default RDP, port 3389 using RDC)
× Linux (default SSH, port 22 using Putty or built in)



# Amazon Machine Images (AMIs) vs Instances

- An instance type specifies the hardware of the host computer used for your instance.
- AWS EC has 5 categories of instance types:

1. General purpose

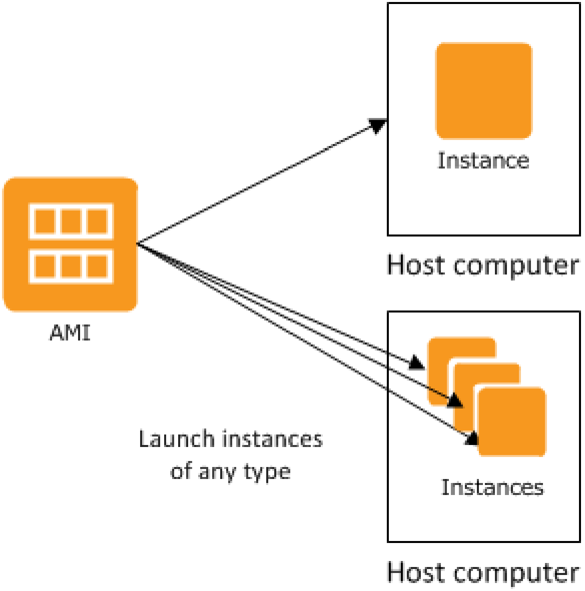
- Memory to CPU ratios for general purpose use
- Come with Fixed performance (M4 instances)
- Or burstable performance (T2) -provides a baseline level of CPU performance with the ability to burst above the baseline. T2 instances are for workloads that don’t use the full CPU consistently, but occasionally need to burst.

2. Compute optimised

- More CPU resources than RAM
- Suitable HPC workloads or scale out computer intensive apps

3. Memory Optimised

- Larger memory (database and in memory caching)



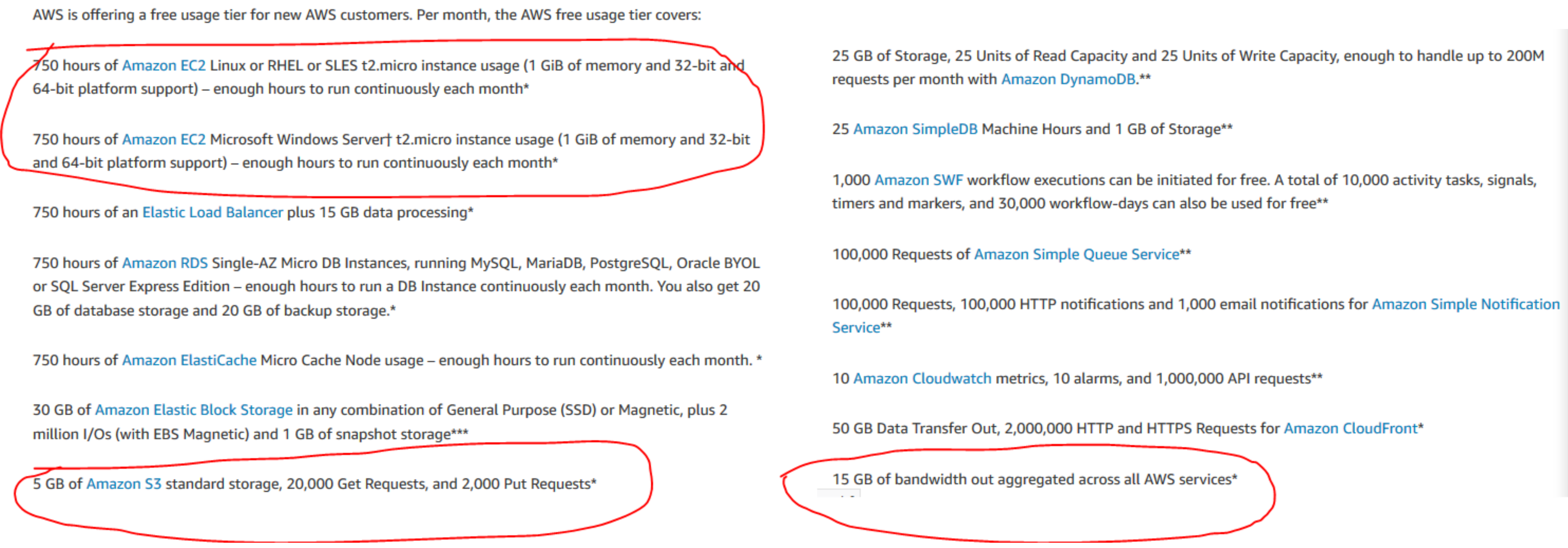
# Amazon Machine Images (AMIs) vs Instances

4. Accelerated Computing (GPUs)
5. Storage Optimised instances

Pricing Model:

- × Free Usage Tier (https://aws.amazon.com/free/)
- × On Demand Tier
  - Start and stop whenever you like (costs rounded to nearest hour). Most expensive. No long term commitment.
- × Reserved instances
  - Pay up front for 1 or 3 years in advance (lower cost, significant discount on hourly rate)
  - Unused can be sold on secondary market (AWS Reserved instance Marketplace)
- × Spot Instances
  - Specify what you’re willing to pay i.e. allows customers to bid on unused AWS EC2 capacity. Run them as long as their max bid exceeds the current spot price (instances may get started and stopped without any warning)

# AWS: Free Tier



AWS is offering a free usage tier for new AWS customers. Per month, the AWS free usage tier covers:

750 hours of Amazon EC2 Linux or RHEL or SLES t2.micro instance usage (1 GiB of memory and 32-bit and 64-bit platform support) – enough hours to run continuously each month*

750 hours of Amazon EC2 Microsoft Windows Server† t2.micro instance usage (1 GiB of memory and 32-bit and 64-bit platform support) – enough hours to run continuously each month*

750 hours of an Elastic Load Balancer plus 15 GB data processing*

750 hours of Amazon RDS Single-AZ Micro DB Instances, running MySQL, MariaDB, PostgreSQL, Oracle BYOL or SQL Server Express Edition – enough hours to run a DB Instance continuously each month. You also get 20 GB of database storage and 20 GB of backup storage.*

750 hours of Amazon ElastiCache Micro Cache Node usage – enough hours to run continuously each month. *

30 GB of Amazon Elastic Block Storage in any combination of General Purpose (SSD) or Magnetic, plus 2 million I/Os (with EBS Magnetic) and 1 GB of snapshot storage***

5 GB of Amazon S3 standard storage, 20,000 Get Requests, and 2,000 Put Requests*

25 GB of Storage, 25 Units of Read Capacity and 25 Units of Write Capacity, enough to handle up to 200M requests per month with Amazon DynamoDB.**

25 Amazon SimpleDB Machine Hours and 1 GB of Storage**

1,000 Amazon SWF workflow executions can be initiated for free. A total of 10,000 activity tasks, signals, timers and markers, and 30,000 workflow-days can also be used for free**

100,000 Requests of Amazon Simple Queue Service**

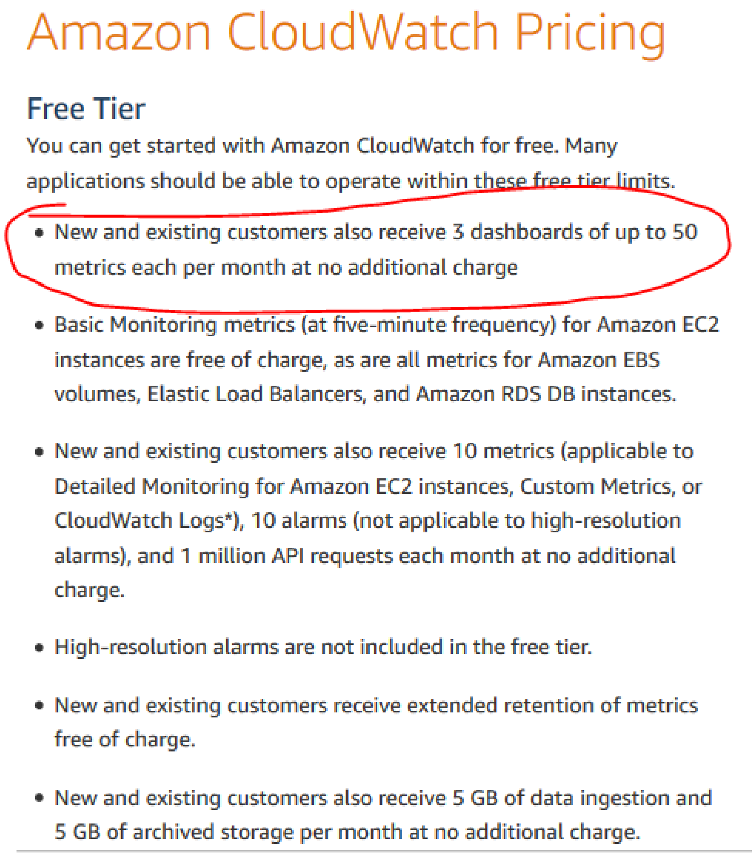
100,000 Requests, 100,000 HTTP notifications and 1,000 email notifications for Amazon Simple Notification Service**

10 Amazon Cloudwatch metrics, 10 alarms, and 1,000,000 API requests**

50 GB Data Transfer Out, 2,000,000 HTTP and HTTPS Requests for Amazon CloudFront*

15 GB of bandwidth out aggregated across all AWS services*

# AWS: Free Tier



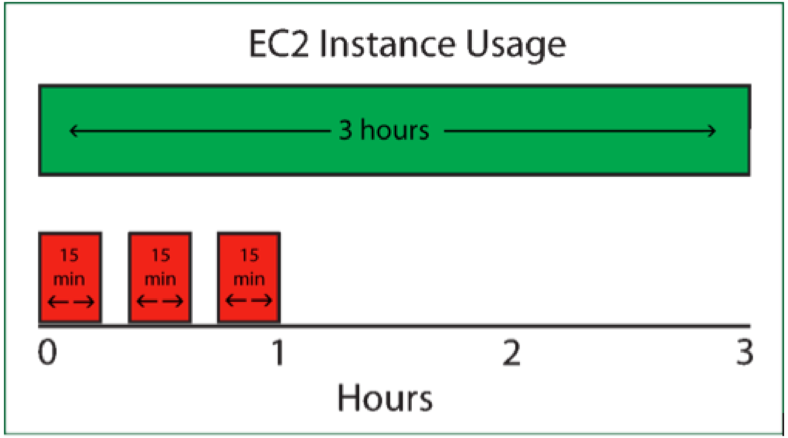
## Amazon CloudWatch Pricing

### Free Tier

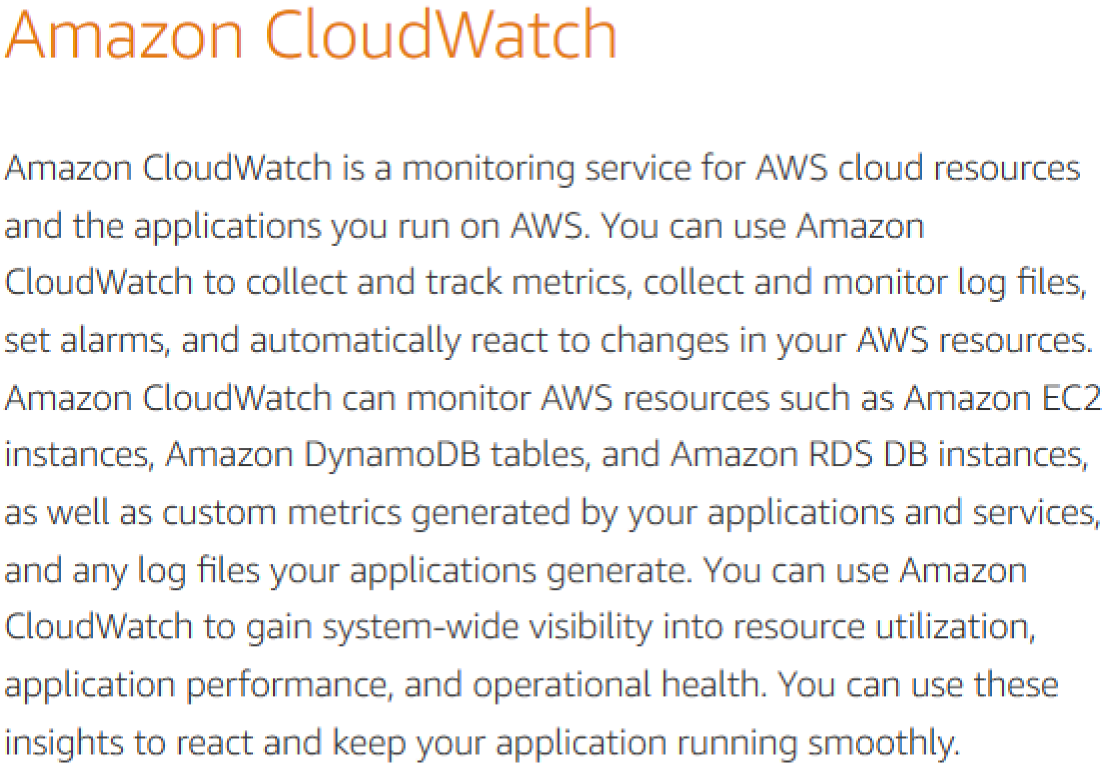
You can get started with Amazon CloudWatch for free. Many applications should be able to operate within these free tier limits.

- New and existing customers also receive 3 dashboards of up to 50 metrics each per month at no additional charge
- Basic Monitoring metrics (at five-minute frequency) for Amazon EC2 instances are free of charge, as are all metrics for Amazon EBS volumes, Elastic Load Balancers, and Amazon RDS DB instances.
- New and existing customers also receive 10 metrics (applicable to Detailed Monitoring for Amazon EC2 instances, Custom Metrics, or CloudWatch Logs*), 10 alarms (not applicable to high-resolution alarms), and 1 million API requests each month at no additional charge.
- High-resolution alarms are not included in the free tier.
- New and existing customers receive extended retention of metrics free of charge.
- New and existing customers also receive 5 GB of data ingestion and 5 GB of archived storage per month at no additional charge.

In some cases, leaving your resources running maximizes your free tier benefits. For example, if you run an Amazon EC2 instance for only a portion of an hour, AWS counts that as an entire hour. Therefore, if you stop and start an Amazon EC2 instance three times in a single hour, you use up three hours of your monthly allotment. The following diagram illustrates how this works. Both the red and green usage scenarios below use up three hours of your monthly allotment.



# AWS: CloudWatch



## Amazon CloudWatch

Amazon CloudWatch is a monitoring service for AWS cloud resources and the applications you run on AWS. You can use Amazon CloudWatch to collect and track metrics, collect and monitor log files, set alarms, and automatically react to changes in your AWS resources. Amazon CloudWatch can monitor AWS resources such as Amazon EC2 instances, Amazon DynamoDB tables, and Amazon RDS DB instances, as well as custom metrics generated by your applications and services, and any log files your applications generate. You can use Amazon CloudWatch to gain system-wide visibility into resource utilization, application performance, and operational health. You can use these insights to react and keep your application running smoothly.

# AWS: Non-Free Tier

Check pricing for whatever services you want in advance!!

For example, you can find prices for AWS EC2 instances here:

https://aws.amazon.com/ec2/pricing/

# Typical ways to use EC2

1. Computation

Or

2. Deploying web applications
   - Run your base system in a minimum # of VMs
   - Monitor the system load (user traffic)
   - Load is distributed to VMs
   - If over some threshold -> increase # of VMs
   - If lower than threshold -> decrease # of VMs

# What is load balancing?

Automatically distributes incoming traffic across multiple EC2 instances.

You create a load balancer and register instances with the load balancer in one or more Availability Zones. The load balancer serves as a single point of contact for clients. This enables you to increase the availability of your application.

If an EC2 instance fails, Elastic Load Balancing automatically reroutes the traffic to the remaining running EC2 instances.

If a failed EC2 instance is restored, Elastic Load Balancing restores the traffic to that instance.

Elastic Load Balancing can also serve as the first line of defence against attacks on your network. You can offload the work of encryption and decryption to your load balancer so that your EC2 instances can focus on their main work.

You can configure your load balancer to distribute requests to EC2 instances in multiple Availability Zones, minimizing the risk of overloading one single instance. If an entire Availability Zone goes offline, the load balancer routes traffic to instances in other Availability Zones.

# Elastic IP Addresses

IPv4:

- July 2015 (ARIN runs out of IPv4), RIPE ran out in 2012, APNIC in 2011.

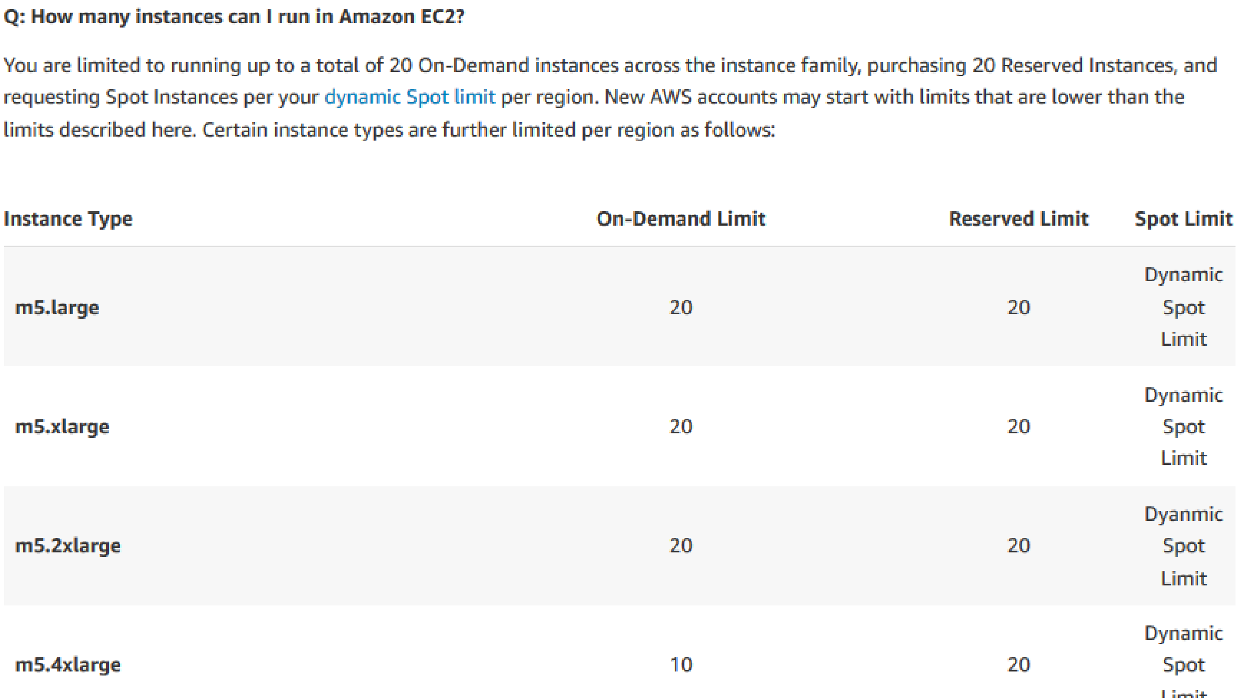
Classic EC2 (Instance IP Addressing): You are assigned two IPs: a privateIP and a publicIP (via NAT). The public IP is associated with an instance **not** your account so you can’t reuse this public IP after it’s been disassociated with your instance i.e. after you terminate the instance.

- IP addresses are assigned at random from an IP address pool
- Direct public->private NAT mapping.

Enter Elastic IP: Static IP address (associated with your account) where you can mask failure of an instance by programmatically remapping the public IP to another instance in your account.

- Rather than waiting on a technician to reconfigure your host or waiting for DNS re-propagation, EC2 enables you to programmatically remap your Elastic IP address to a replacement instance.
- Charged ~$0.01 per hour if not using them (pricing varies per region). Limited to 5 elastic IPs per region

# AWS EC2: Limited Scalability



**Q: How many instances can I run in Amazon EC2?**

You are limited to running up to a total of 20 On-Demand instances across the instance family, purchasing 20 Reserved Instances, and requesting Spot Instances per your dynamic Spot limit per region. New AWS accounts may start with limits that are lower than the limits described here. Certain instance types are further limited per region as follows:

| Instance Type | On-Demand Limit | Reserved Limit | Spot Limit |
|---|---|---|---|
| **m5.large** | 20 | 20 | Dynamic Spot Limit |
| **m5.xlarge** | 20 | 20 | Dynamic Spot Limit |
| **m5.2xlarge** | 20 | 20 | Dyanmic Spot Limit |
| **m5.4xlarge** | 10 | 20 | Dynamic Spot |

*Note that cc2.8xlarge, hs1.8xlarge, cr1.8xlarge, G2, D2, and I2 instances are not available in all regions.*

If you need more instances, complete the Amazon EC2 instance request form with your use case and your instance increase will be considered. Limit increases are tied to the region they were requested for.

# AWS Educate vs Free Tier

- Educational resource/curricula – hands on experience with AWS services

  https://aws.amazon.com/education/awseducate/

- Free Tier –Service available to all new consumers

- Can possibly create a new account, using a virtual credit card such as Revolut/N26, to access Free Tier.*

# AWS Auto Scaling

- Auto Scaling enables you to follow the demand of your applications closely, reducing the need to manually provision Amazon EC2 capacity in advance e.g. you can set a condition to add new Amazon EC2 instances in increments to the Auto Scaling group when the average utilization of your Amazon EC2 fleet is high; and similarly, you can set a condition to remove instances in the same increments when CPU utilization is low.

- If you have predictable load changes, you can set a schedule through Auto Scaling to plan your scaling activities. You can use Amazon CloudWatch to send alarms to trigger scaling activities and Elastic Load Balancing to help distribute traffic to your instances within Auto Scaling groups. Auto Scaling enables you to run your Amazon EC2 fleet at optimal utilization.

- For example, you can define a scale up condition to increase your Amazon EC2 capacity by 10% and a scale down condition to decrease it by 5%

- Notice the useful cloud management & networking services mentioned to date: CloudWatch, LoadBalancing, Auto Scaling

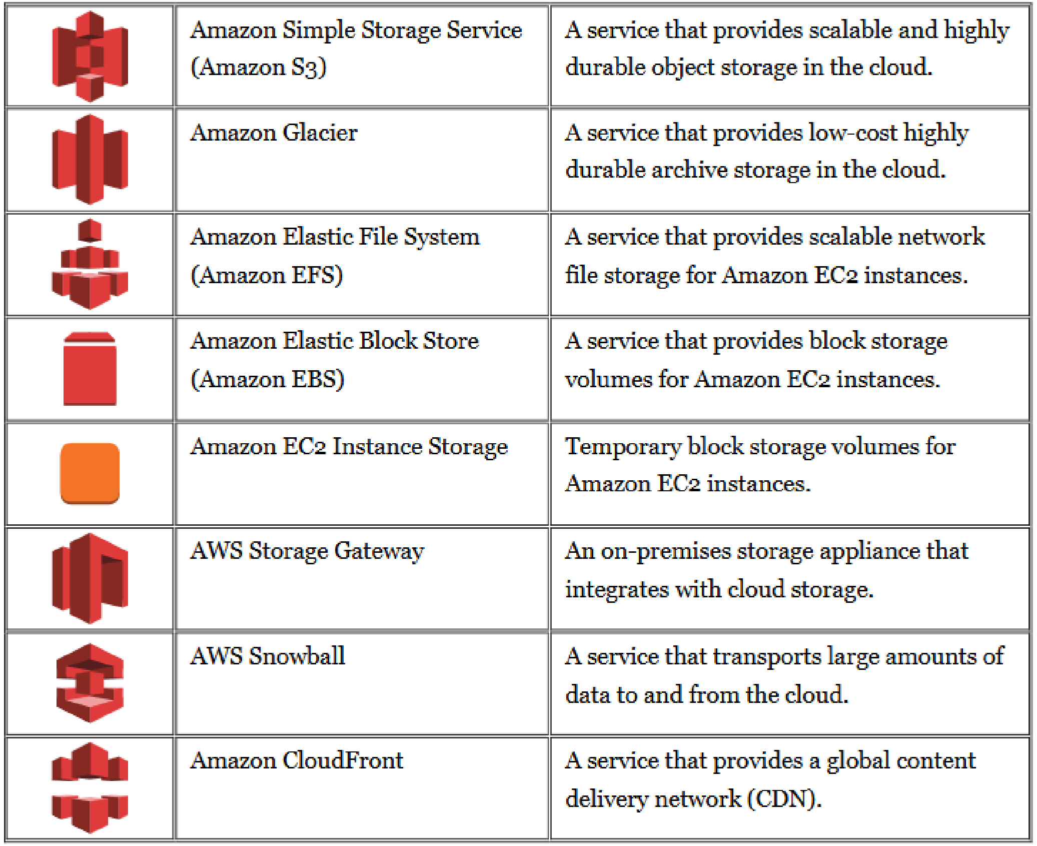
# AWS EC2 Terminology Summary

- Image i.e. an AMI: Stored images or “flavours” of machines (software) that can be turned into instances. The AMI specified OS, servers, applications etc
- Instance: One running virtual machine
- Instance Type: Hardware Configuration e.g. cores, memory, disk
- Volume: Temporary hard disk (EBS) associated with an instance
- Key Pair: Credentials used to access the VM from the command line
- Region: Geographic Location, price, laws, network locality
- Availability Zone(s): Sub division of the region that is fault-dependant.

# Problems/Risks with EC2

- On power off, all hard disk data is lost (unless you explicitly use S3 or save your EBS volume)
- IP addresses are assigned at random
- Can’t turn off public IP address
- Do not forget to terminate instances.

# AWS: Storage Services



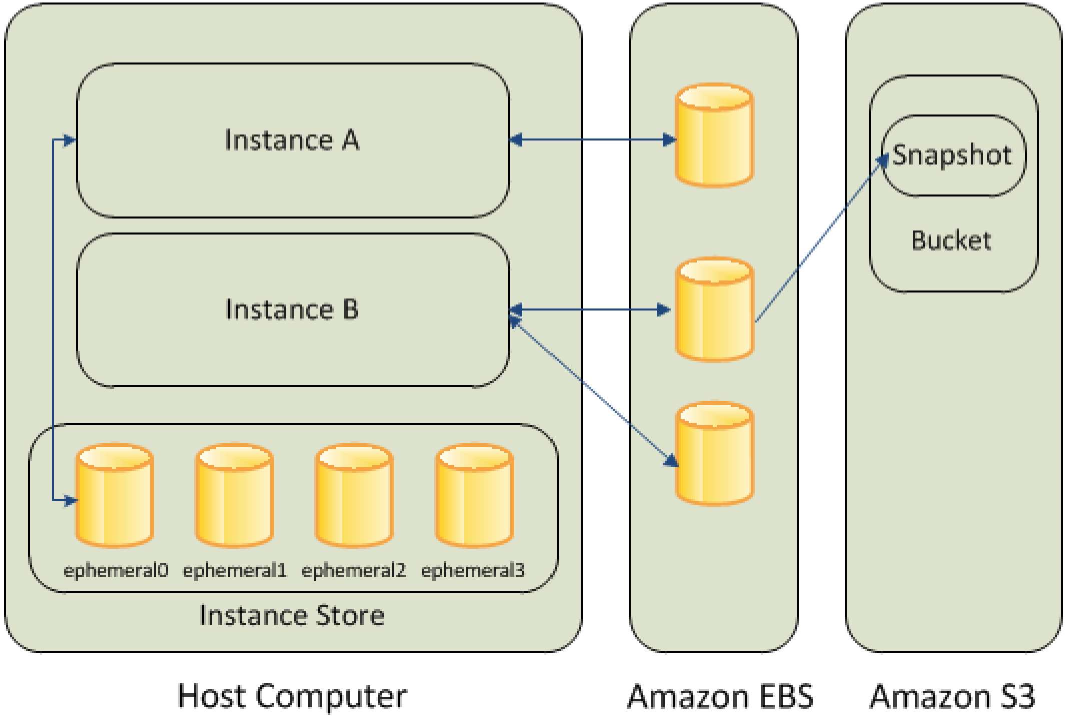
| | | |
|---|---|---|
| | Amazon Simple Storage Service (Amazon S3) | A service that provides scalable and highly durable object storage in the cloud. |
| | Amazon Glacier | A service that provides low-cost highly durable archive storage in the cloud. |
| | Amazon Elastic File System (Amazon EFS) | A service that provides scalable network file storage for Amazon EC2 instances. |
| | Amazon Elastic Block Store (Amazon EBS) | A service that provides block storage volumes for Amazon EC2 instances. |
| | Amazon EC2 Instance Storage | Temporary block storage volumes for Amazon EC2 instances. |
| | AWS Storage Gateway | An on-premises storage appliance that integrates with cloud storage. |
| | AWS Snowball | A service that transports large amounts of data to and from the cloud. |
| | Amazon CloudFront | A service that provides a global content delivery network (CDN). |

# Storage associated with an AWS Instance (VM)

3 options:

1. AWS Elastic Block Storage (EBS)
2. Amazon EC2 Instance Storage
3. AWS Simple Storage Service (S3

The following figure shows the relationship between these types of storage.



# Elastic Block Storage (EBS)

- An EBS volume is a virtual disk of a fixed size and can be mounted as a filesystem on a running EC2 instance.
- After an EBS volume is attached to an instance, you can use it like any other physical hard drive.
- The volume persists independently from the running life of an instance.
- Multiple volumes can be attached to an instance
- You can also detach an EBS volume from one instance and attach it to another instance.
- To keep a backup copy of your data, you can create a *snapshot* of an EBS volume, which is stored in Amazon S3. You can create an EBS volume from a snapshot, and attach it to another instance
- Data is replicated across multiple servers in an Availability Zone to prevent loss of data from any single component

# EC2 Instance Store

- Many instances can access storage from disks that are **physically** attached to the host computer.
- This disk storage is referred to as instance store.
- Instance store provides **temporary** block-level storage for instances.
- The data on an instance store volume persists only during the life of the associated instance;
- If you stop or terminate an instance, any data on instance store volumes is lost

# S3

- Allows you to store and retrieve any amount of data, at any time, from within Amazon EC2 or anywhere on the web.

- Amazon S3 stores data objects redundantly on multiple devices across multiple facilities.

    - These objects could be data files (e.g. word docs, photos etc) or for example Amazon EC2 uses Amazon S3 for storing AMIs or EC2 also uses S3 to store snapshots (backup copies) of the data volumes.

- Objects are the fundamental entities stored in Amazon S3. Every object stored in Amazon S3 is contained in a bucket. Amazon S3 buckets are similar to Internet domain names. Objects stored in the buckets have a unique key value and are retrieved using a HTTP URL address.

- Designed to sustain concurrent loss of data in two facilities e.g. 3+ copies across multiple available domains.

# S3

- Buckets store data (analogous to a folder). They are the fundamental container in Amazon S3 for data storage.

- A bucket contains objects (files between 1B-5TB)

- A bucket has a name that must be **globally** unique!!!

- A bucket has a flat directory structure (despite the appearance given by the web console)

- Access a bucket via HTTP:
  - http://bucket.s3.amazonaws.com/object

- Buckets are limited at 100 buckets per account
  - No limit on the # of objects that can be stored in a bucket
  - Object stores data and metadata. Objects stored in a region and never leave a region
  - You cannot modify or append data to an existing object.

# Methods for accessing AWS

1. Web console

2. Command line tools

3. Cloud API: e.g. Boto3 python library