AWS Architecting and SysOps

Computing on AWS, Part 1

June-July 2019

Contents

Amazon EC2
Auto Scaling

Amazon EC2

Virtualization concepts on AWS

Compute

Amazon EC2

Amazon EC2 Auto Scaling

AWS Lambda

Amazon Elastic Container Service

AWS Elastic Beanstalk

Elastic Load Balancing (ELB)

Benefits of Cloud Computing

- Speed
  - Vast amount of computing resources can be provisioned in minutes
- Cost
  - It eliminates the expense of buying computer hardware and software
  - You only pay for what you use
- Scalability
  - Easy to scale up and down your cloud capacity
- Better Security
  - With cloud, your data is stored in a secure location
- Accessibility
  - Easy to access data anywhere ➔ You can work from home!!
- And you still have complete control over your product

Elastic Compute Service (EC2)

- EC2 is a web service that provides secure, resizable compute capacity in the cloud
  - Its simple web interface allows you to obtain and configure capacity with minimal ease
- ➢Designed to make web-scale cloud computing easier for developers
- It provides you with complete control of your computing resources and lets you run on Amazon's computing environment
- 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
- EC2 changes the economics of computing by allowing you to pay only capacity that you actually use
- EC2 provide developers tools to build failure resilient applications and isolate them from common failure scenarios

Steps to use EC2

1. Choosing an AMI (Amazon Machine Image)
2. Choosing an instance type
3. Configure the instance
4. Adding storage
5. Adding tags
6. Configure security group
7. Review

1. Choosing an AMI (Amazon Machine Image)

- An AMI is a template that is used to create a new instance / machine based on user requirement
- The AMI contains compacted information about:
  - Software
  - Operating system
  - Volume
  - Access permissions
- Types of AMIs:
  - Predefined AMIs: created by Amazon and can be modified by the user
  - Custom AMIs: created by the user so that they can be reused
  - You could also get one from AMI Marketplace

2. Choosing an instance type

- An instance type specifies the hardware specifications that are required in he machine from the previous step
- Instances are divided into 5 main families:
    1) Compute optimized: used for heavy processing power
    2) Memory optimized: used for optimized in-memory cache
    3) GPU optimized: for large graphic requirements like gaming systems
    4) Storage optimized: setting up a storage server
    5) General purpose: when everything is equally balanced
- If you do not have specific requirements, select general purpose
- Instance types are fixed and their configurations cannot be altered
    - We do not have control over the hardware

EC2 instance types

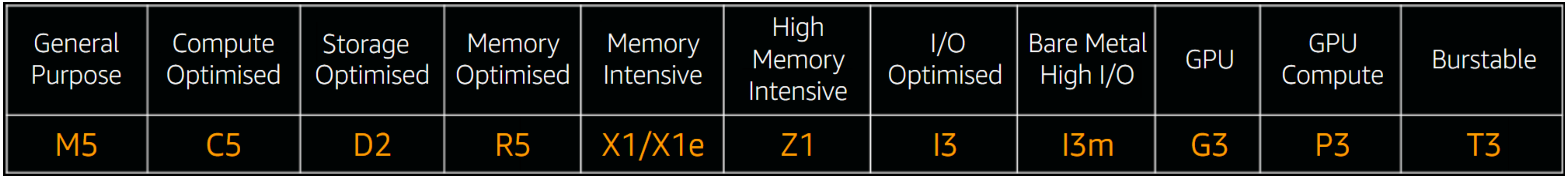
- Consider the following when choosing your instances:

✓ Core count
✓ Memory size
✓ Storage size & type
✓ Network performance
✓ CPU technologies

Depending on those parameters, EC2 is divided in different types of instances

https://aws.amazon.com/ec2/instance-types/

- General Purpose
  - A1, T3, T2, M5, M5a, M4, T3a
- Compute Optimized
  - C5, C5n, 4
- Memory Optimized
  - R5, R5a, R4X1e, X1, High Memory, z1d
- Accelerated Computing (GPU optimized)
  - P3, P2, G3, F1
- Storage Optimized
  - H1, I3, D2



General Purpose	Compute Optimised	Storage Optimised	Memory Optimised	Memory Intensive	High Memory Intensive	I/O Optimised	Bare Metal High I/O	GPU	GPU Compute	Burstable
M5	C5	D2	R5	X1/X1e	Z1	I3	I3m	G3	P3	T3

3. Configure instance

- Several options to choose from:

✓Number of instances

✓Purchasing options

✓Network configuration: subnet and IP

✓IAM role

✓Shutdown behavior

  - Stopping: temporary shutting down the system
  - Terminating: returning control back to Amazon

✓Enable termination protection

  - You cannot terminate your instance unless this is not disabled

✓Monitoring

  - Using CloudWatch, basic monitoring is free of charge

➢ In the IP section, if a public DNS is necessary to access the EC2 instance, you need to set:
Auto-assign Public IP to Enable

➢ Under ‘Advanced details’->’User data’, you can add bootstrap scripts that are executed when the virtual machine starts up

EC2 purchasing options

- On-Demand Instances
  - Pay, by the second, for the instances that you launch
- Reserved Instances
  - Purchase, at a significant discount, instances that are always available, for a term from one to three years
- Scheduled Instances
  - Purchase instances that are always available on the specified recurring schedule, for a one-year term
- Spot Instances
  - Request unused EC2 instances, which can lower your EC2 costs significantly

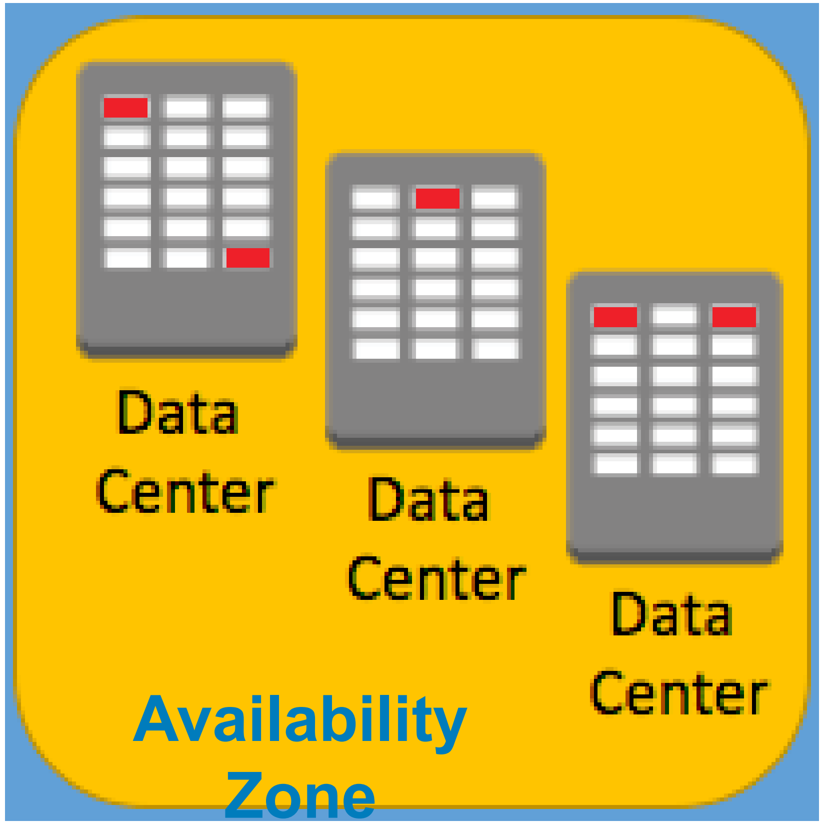
EC2 purchasing options

- Dedicated Hosts
  - Pay for a physical host that is fully dedicated to running your instances
- Dedicated Instances
  - Pay, by the hour, for instances that run on single-tenant hardware.
- Capacity Reservations
  - Reserve capacity for your instances in a specific AZ for any duration

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

Placement groups

- When you launch a new EC2 instance, the EC2 service attempts to place the instance, within the availability zone, so that all of your instances are spread out across underlying hardware to minimize correlated failures



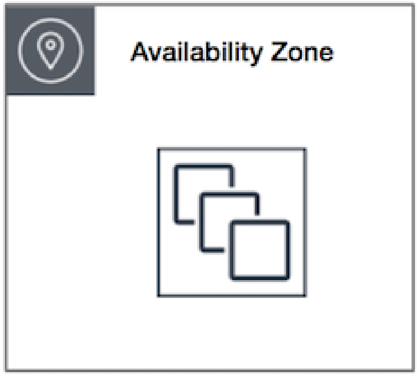
- You can influence the placement of your instances to meet the needs of your workload

- Strategies:
  - Cluster
  - Partition
  - Spread

- There is no charge for creating a placement group

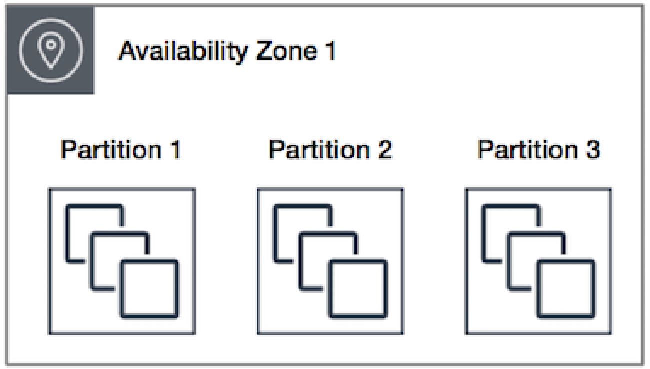
Placement groups strategies

Cluster



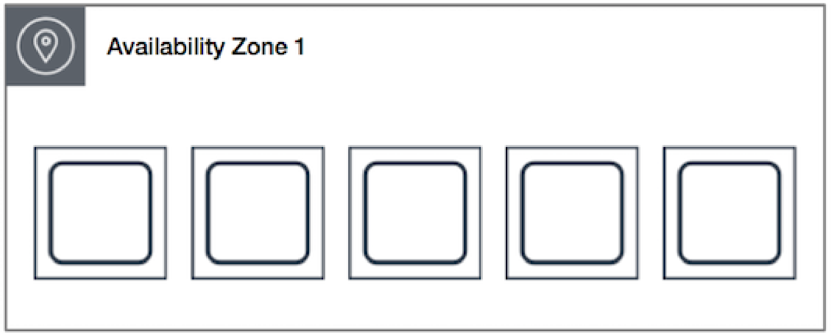
- Packs instances close together inside the AZ
- It enables workloads to achieve the low-latency network performance necessary for tightly-coupled node-to-node communications

Partition



- Spreads your instances across logical partitions
- Groups of instances in one partition do not share the underlying hardware with groups of instances in different partitions

Spread

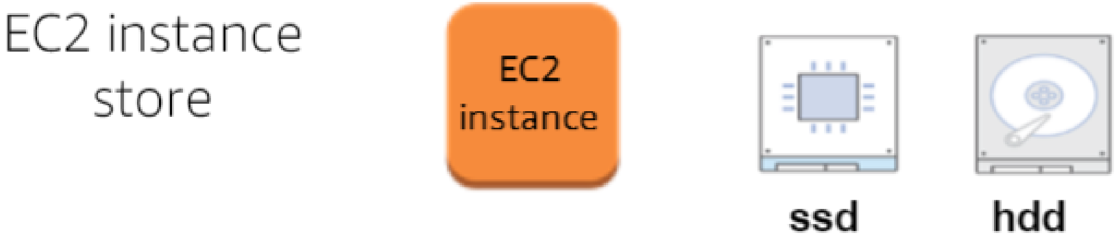


- Strictly places a small group of instances across distinct underlying hardware to reduce correlated failures

4. Adding storage

- It is necessary to add storage to the instance I’m about to launch

- Deciding the type of storage:
  - Ephemeral storage
    - Temporary and free
  - Amazon EBS
    - EBS SSD or HDD volumes
  - Amazon S3
    - Buckets



- The size (in GBs), volume type, the location where the disk is mounted, and whether the volume needs to be encrypted
  - Free users: up to 30GB of SSD or Magnetic storage

5. Adding tags

- To help you manage your instances, images, and other EC2 resources, you can optionally assign your own metadata to each resource in the form of tags

- A tag is a label that you assign to an AWS resource
  - Each tag consists of a key and an optional value, both of which you define.

- Tags enable you to categorize your AWS resources in different ways
  - Ex/ By purpose, owner, or environment

- This is useful when you have many resources of the same type as you can quickly identify a specific resource based on the tags you've assigned to it.
  - Ex/ You could define a set of tags for your account's EC2 instances that helps you track each instance's owner

6. Configuring security groups

- Security groups are a set of firewall rules that sit in front of an instance and protects it from unintended traffic
- When you launch an instance, you can specify one or more security groups
  - Otherwise, we use the default security group
- You can add rules to each security group that allow traffic to or from its associated instances.
- You can modify the rules for a security group at any time
  - New rules are automatically applied to all instances associated with the security group
- When AWS decides whether to allow traffic to reach an instance, all the rules from all the security groups that are associated with the instance are evaluated

➢Here is where you can fine tune the access to your instance based on port numbers and based on IP @ from which it can be accessed

7. Review

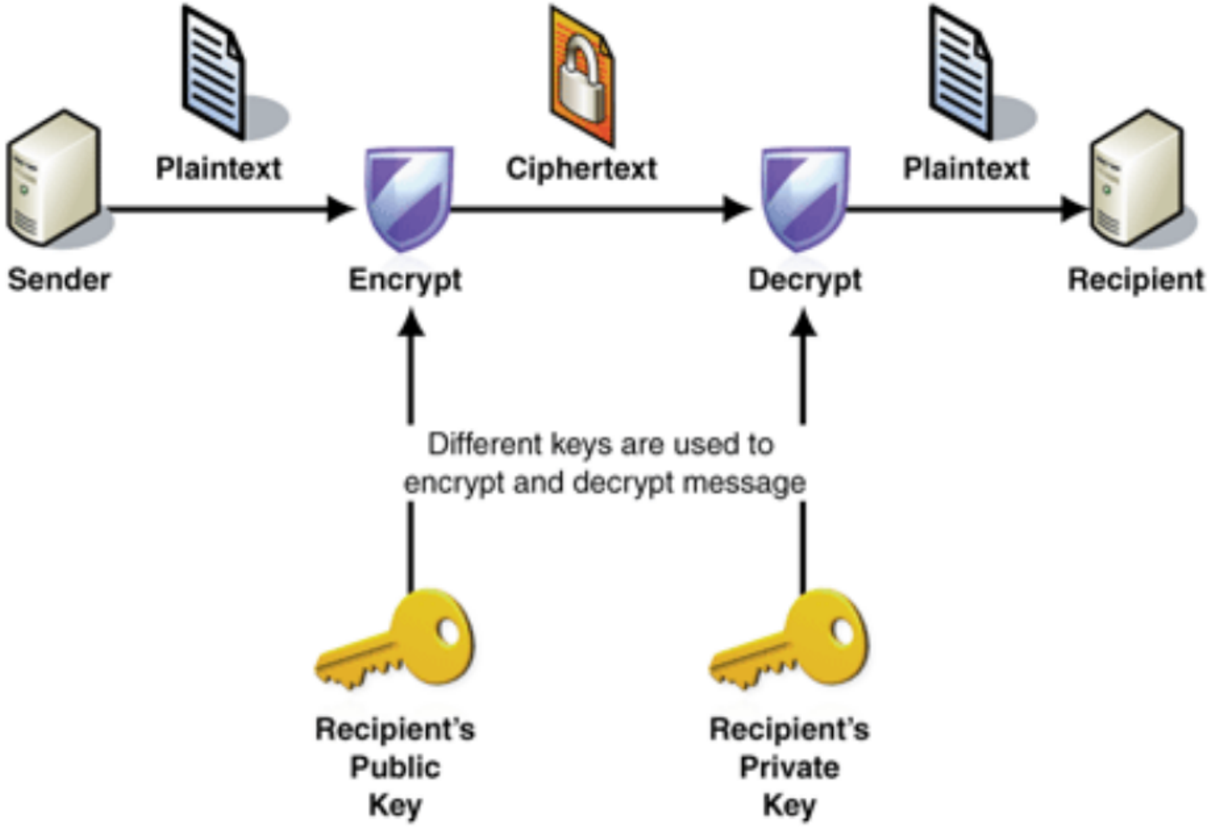
- Finally you get to review the whole changes or configurations you have made to find whether they still meet your requirements, and then click to launch your EC2 instance

- Wait!!

- Before launching the EC2 instance, AWS is going to give us the opportunity to create a key pair:

✓The private key is downloaded by the user

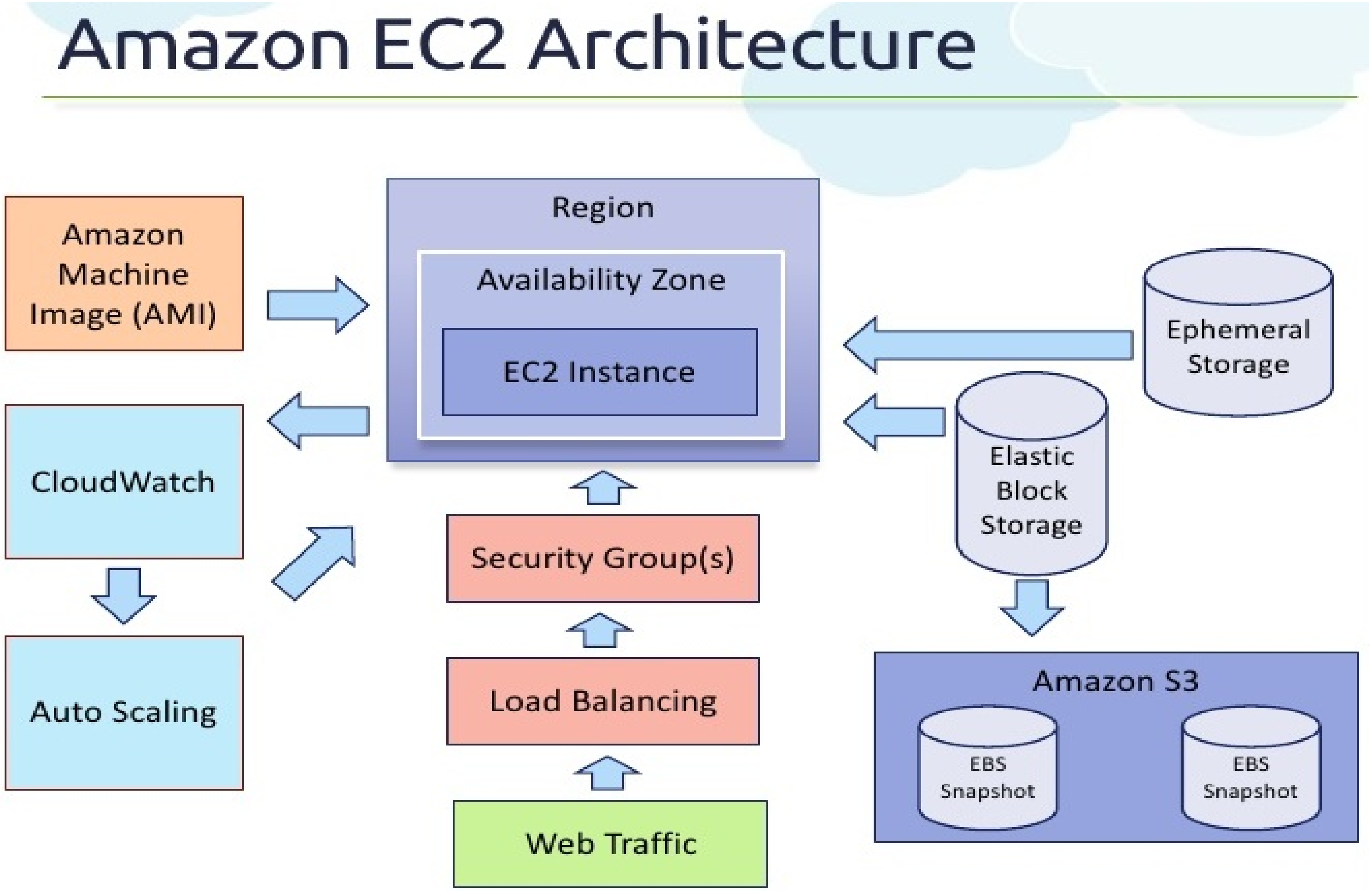
✓The public key, is used by AWS to confirm the identity of the user



EC2 use cases

- EC2 is a core compute unit that can be used for almost anything:

✓Application server
✓Web server
✓Database server
✓Game server
✓Mail server
✓Media server
✓Catalog server
✓File server
✓Computing server
✓Proxy server



EC2: understanding better pricing

- Running EC2 instances are charged by the second, with a minimum of 1 minute
  - If you start and stop and instance for 20 seconds, you are charged 1 minute
  - If you start and stop and instance for 1 min 20 secs, you are charged 1 min and 20 secs

- If you stop an EC2 instance, you are not charged
  - But you are charged for any EBS volume associated per-second
  - And for any elastic IP address assigned to the stopped instance

- If you terminate an EC2 instance, you are not charged anymore for the instance
  - By default, root EBS volume is automatically deleted but if you choose not to delete it or if you had other EBS volumes associated to the instance, you will incur in charges
  - If you don’t release the elastic IP address, you will incur in charges

Auto Scaling

Horizontal Scaling of virtual servers

EC2 Auto Scaling

- Allows to automatically add or remove EC2 instances according to conditions you define to help you maintain application availability

- With fleet management feature you can maintain the health and availability of your fleet
  - Status checks are performed every minute and each returns a pass or a fail status.
  - ✓ Overall status OK: if all checks pass
  - ✓ Overall status impaired: if one or more checks fail

- And with dynamic and predictive scaling features, you can add or remove EC2 instances
  - Dynamic scaling responds to changing demand
  - Predictive scaling automatically schedules the right number of EC2 instances based on predicted demand
  - Dynamic scaling and predictive scaling can be used together to scale faster

Scaling

- Dynamic Scaling
  - You can follow the demand curve for your applications closely, reducing the need to manually provision EC2 capacity in advance.
  - Ex/ you can use target tracking scaling policies to select a load metric for your application, such as CPU utilization. Or you could set other metrics to set targets that EC2 Auto Scaling will use to automatically adjust the number of EC2 instances as needed to maintain your target
- Predictive Scaling
  - It uses machine learning to schedule the right number of EC2 instances in anticipation of approaching traffic changes
  - It predicts future traffic, including regularly-occurring spikes, and provisions the right number of EC2 instances in advance
  - Its machine learning algorithms detect changes in daily and weekly patterns, automatically adjusting their forecasts, removing the need for manual adjustment of EC2 Auto Scaling parameters as cyclicality changes over time

Fleet Management

- You can use EC2 Auto Scaling to detect impaired EC2 instances and unhealthy applications, and replace the instances without your intervention
  - This ensures that your application is getting the compute capacity that you expect
- EC2 Auto Scaling will perform three main functions to automate fleet management for EC2 instances:
- Monitor the health of running instances

✓Your application is able to receive traffic and EC2 instances are working properly. EC2 Auto Scaling periodically performs health checks to identify any instances that are unhealthy

✓Replace impaired instances automatically

✓EC2 Auto Scaling automatically terminates any unhealthy EC2 instance and replaces it with a new one. You don’t need to respond manually when an instance needs replacing

✓Balance capacity across Availability Zones

✓EC2 Auto Scaling can automatically balance instances across AZ, and it launches new instances so that they are balanced between AZ as evenly as possible across your entire fleet

Compute documentation

- Amazon EC2 documentation
  - https://docs.aws.amazon.com/ec2/
- Amazon EC2 Auto Scaling
  - https://aws.amazon.com/ec2/autoscaling/
- AWS Lambda
  - https://docs.aws.amazon.com/lambda/
- Amazon Elastic Container Service (ECS)
  - https://docs.aws.amazon.com/ecs/
- AWS Elastic Beanstalk
  - https://docs.aws.amazon.com/elastic-beanstalk/