# Spring - Mar 11

## Agenda

1, introduction to AWS

2, Compute in Cloud

3, Global infrastructure and Security

4, Networking

5, Storage and database

6, Security

7, Monitoring and Analytics

8, Migration

## 0, AWS Certification

foundational

associate

Profesional

- solutions architect
- devops

Specialty

## 1, introduction to AWS

### client - server

sever

- CPU
- RAM

- Data
- Routers, Switch

## Traditional Approach

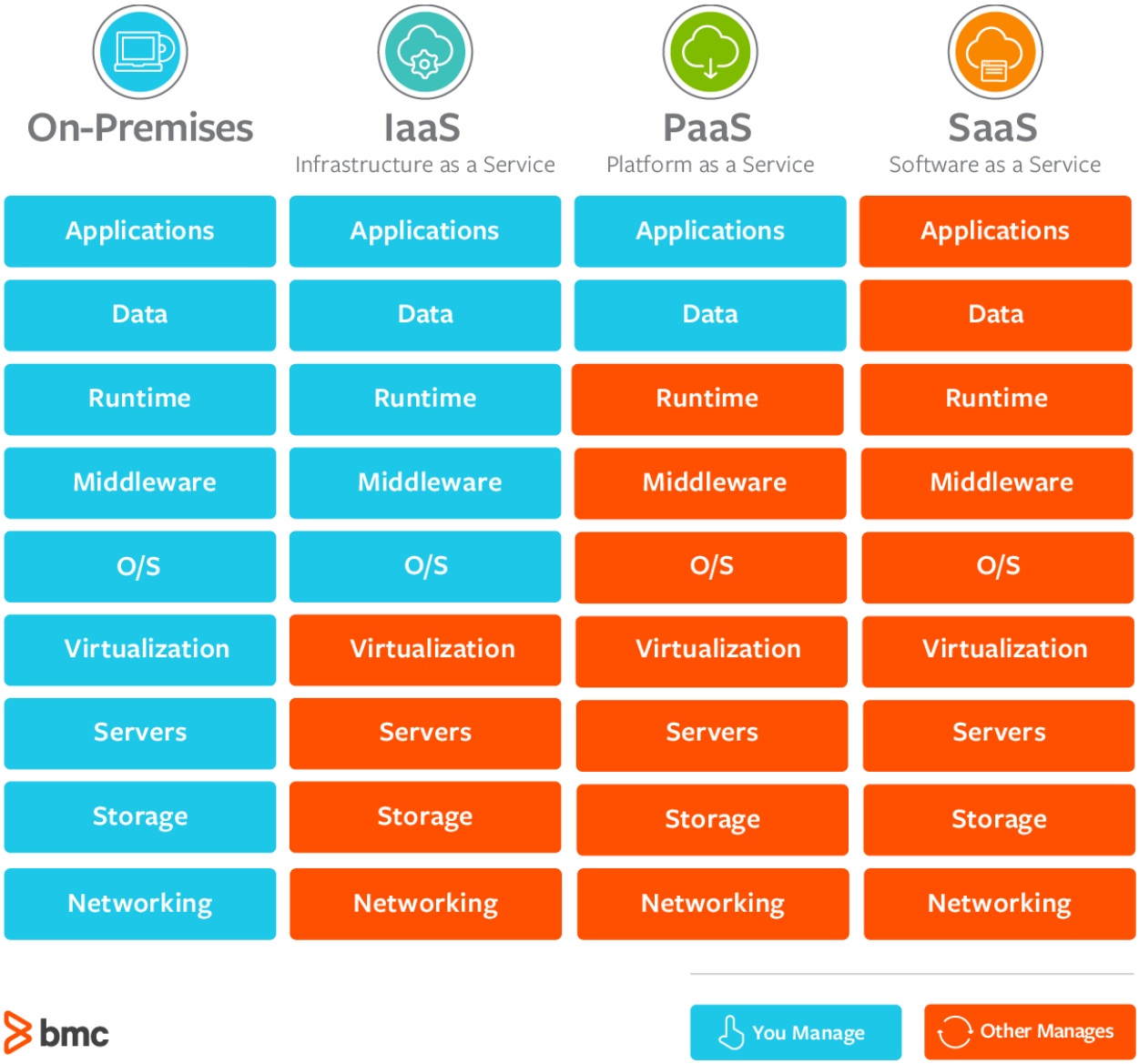
- pay for the rent for data center
- pay for power supply, cooling, maintenance
- adding and replacing hardware
- scaling is limited
- monitor 24/7

## Cloud Computing

*on demand delivery of IT resource*

- pay as you go pricing
    - how much pay depends on how much you use
- provision exactly the right size and type of computing resources
- accessed instantly

## SaaS vs PaaS vs IaaS




# Deployment models for cloud computing

- public cloud: cloud based
  - put everything on the cloud
- private cloud: on-premises
- Hybrid cloud: hybrid
  - some parts of your services are in the local, and some parts are in the cloud

# 2, Compute in Cloud

## EC2

Amazon Elastic Compute Cloud

- general purpose instance
  - provide a very balanced compute memory and networking resources
  - application server/gaming server/backend server for enterprise application
- compute optimized instance
- memory optimized instance
  - process large data size in the memory
- accelerated computing instance
  - use a hardware accelerators and processors
    - double, treble cores of processors
- storage optimized instance
  - designed for workloads that require high sequential read and write access to large data sets on local storage.

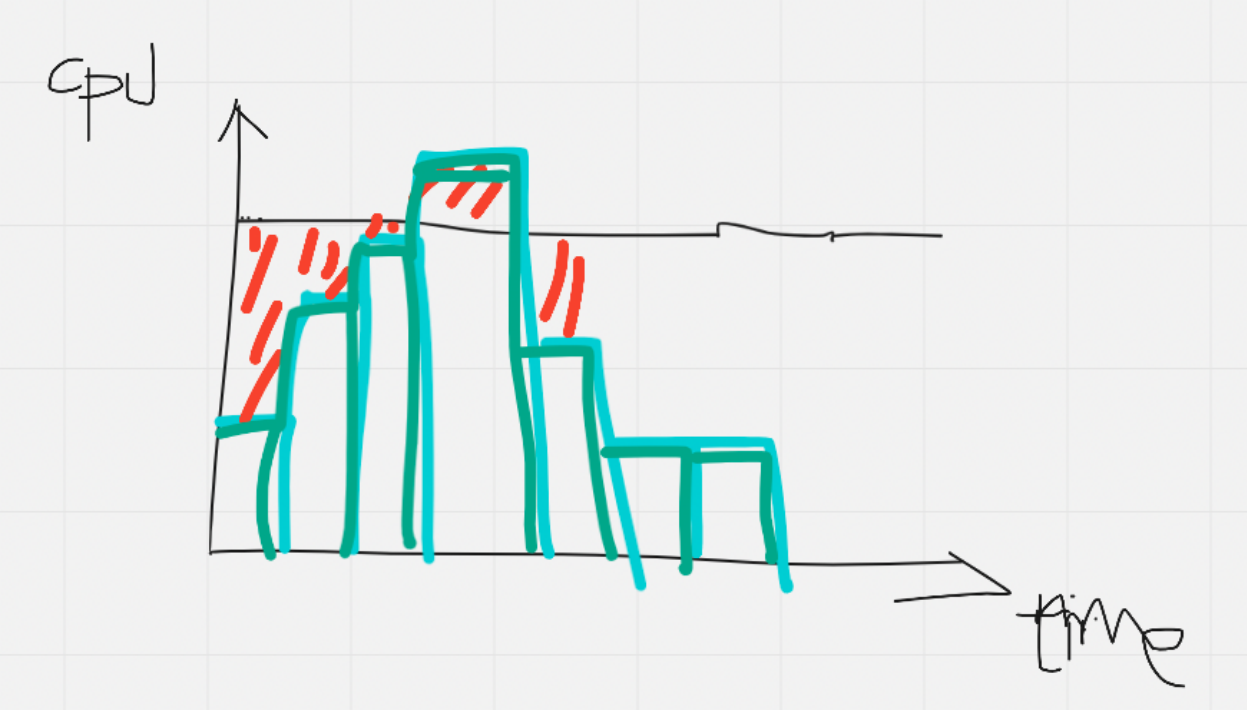
EC2 Pricing

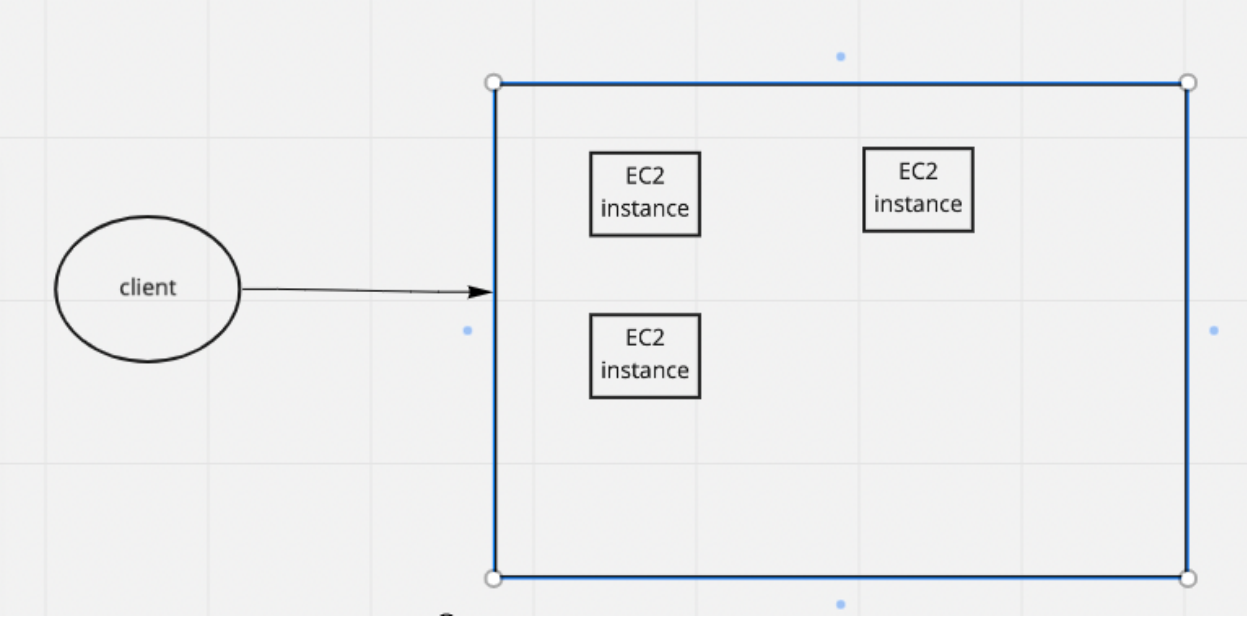
- spot instance
- reserved instance
- on demand instance

Scaling

vertical scaling/horizontal scaling

- **Auto Scaling - AWS(important concept)**
  - belong to a kind of horizontal scaling (automatically)
  - price depends on how much you really use



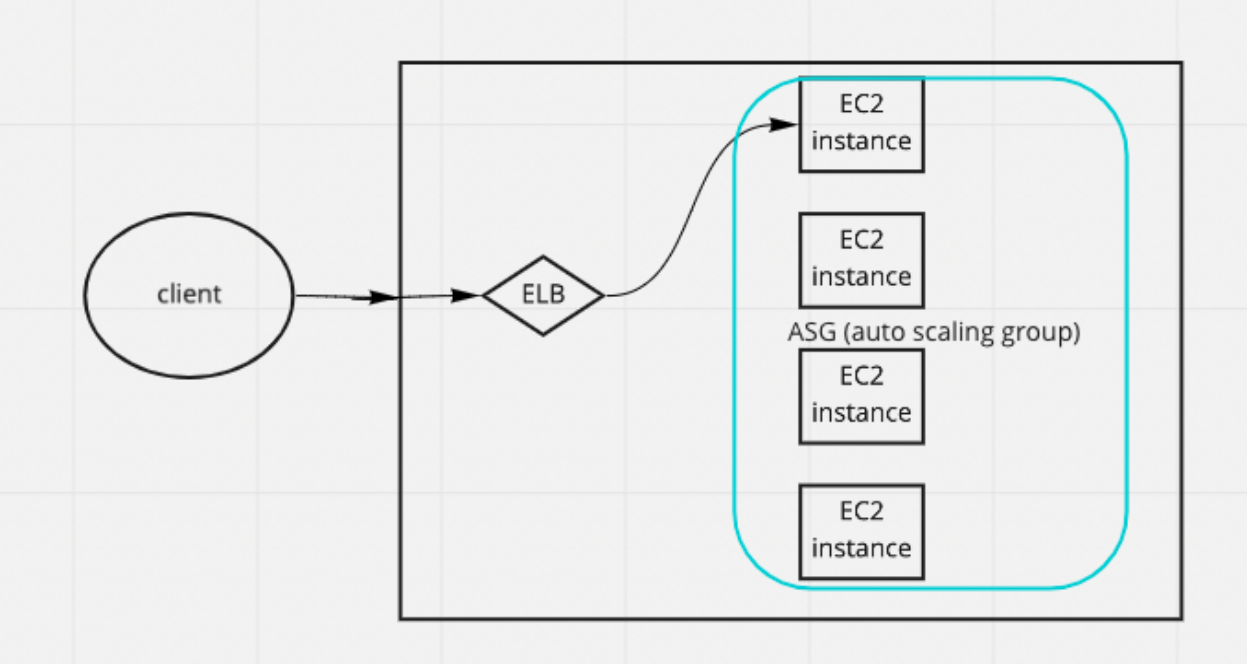




in this case

- originally, provide two instances for client
- when more requests come in, the Auto Scaling will add one more instance to fit the request requirements.
- if one is not enough, add more

- you just pay how much you use

## Elastic Load Balancing




In this case

- When request is sent by client, ELB will decide which EC2 instance to handle the request
- sticky session
  - route the specific request to specific session

ASG - auto scaling group

## Monolithic application vs Microservice Application

- Monolithic application
  - closely coupled: if one component is down, may the whole server will be down
  - easy to establish(e.g less than 100 users)
- MicroServices (very suitable for AWS)
  - loosely coupled

Attention:

One instance can service for several MicroService

One MicroService also may demand multiple instances to support

Instance and MicroService are not direct relationship

## SNS (Amazon Simple Notification Service)

## SQS (Amazon Simple Queue Service)

## AWS lambda (serverless)

## Virtualization

*Virtualization is a software based or just a representation of application server or storage and network.*

- virtual machine
- hypervisor
  - hypervisor is a software which will create a virtual machine for you.

## Containers

- docker

EC2 is virtualization based

## ECS (Amazon Elastic Container Service)

- container base
- support docker containers
  - just deploy docker in the ECS

## ECR (Amazon Elastic Container Registry/Repository)

- use to manage dockers

- inventory

## EKS (Amazon Elastic Kubernetes Service)

k8s

***Kubernetes***, also known as K8s, is an open-source system for automating deployment, scaling, and management of containerized applications.

- use to manage all (e.g 100)dockers?

## AWS Fargate

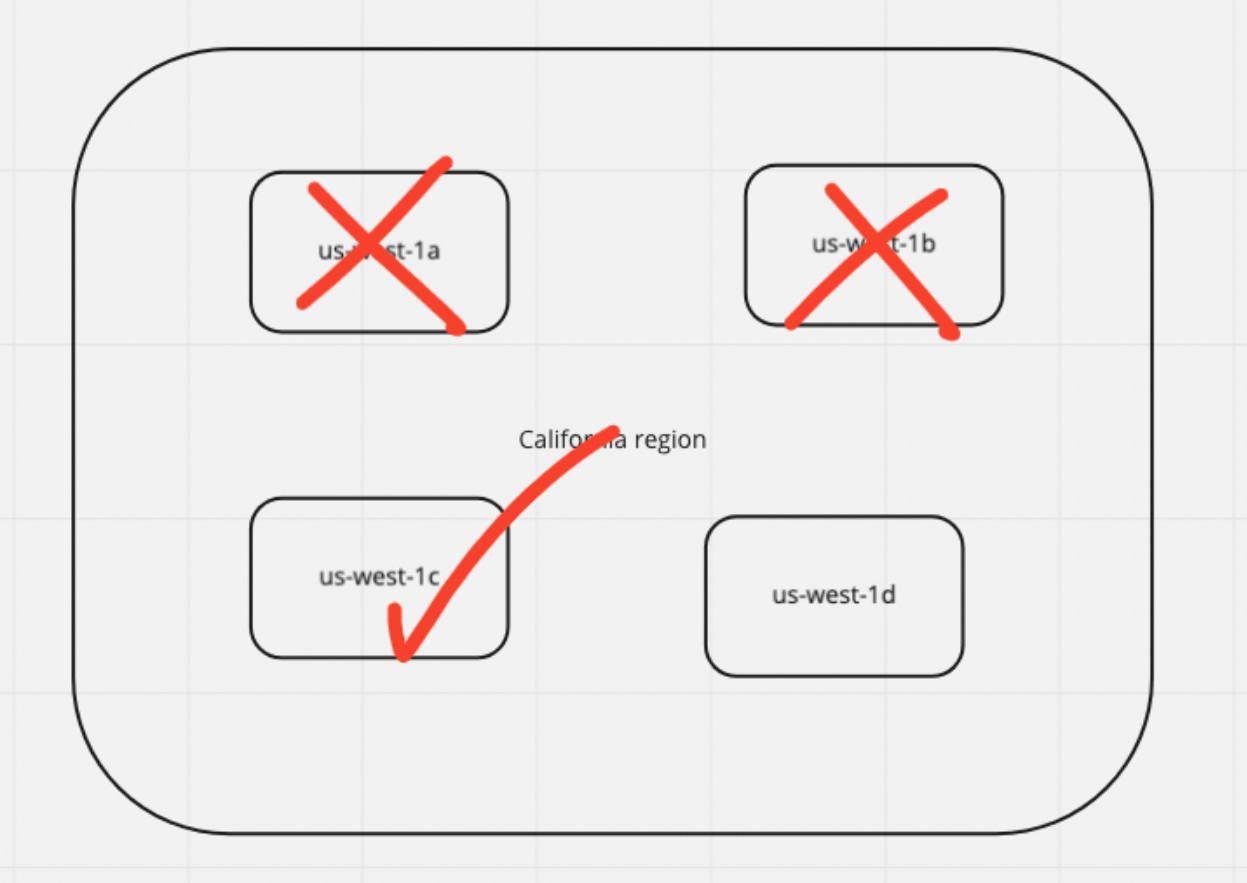
- manage infrastructure

# 3, Global infrastructure and Security

## AWS Region & Availability Zone

Region: physical place

Availability Zone: physical data center in that region




## Edge Location

- cache your data at edge location

e.g people in Asia want to access data that's in US(long distance)

- they don't go to US to access data, just go to edge location to get the cached data

## AWS Elastic Beanstalk

- adjust capacity
- load balancing
- automatic scaling
- application health monitoring

## AWS CloudFormation

*AWS CloudFormation is* ***a service that gives developers and businesses an easy way to create a collection of related AWS and third-party resources, and provision and manage them in an orderly and predictable fashion****.*

- we just need  write codes, not need to care about infrastructure and so on. The cloud will set them for us.

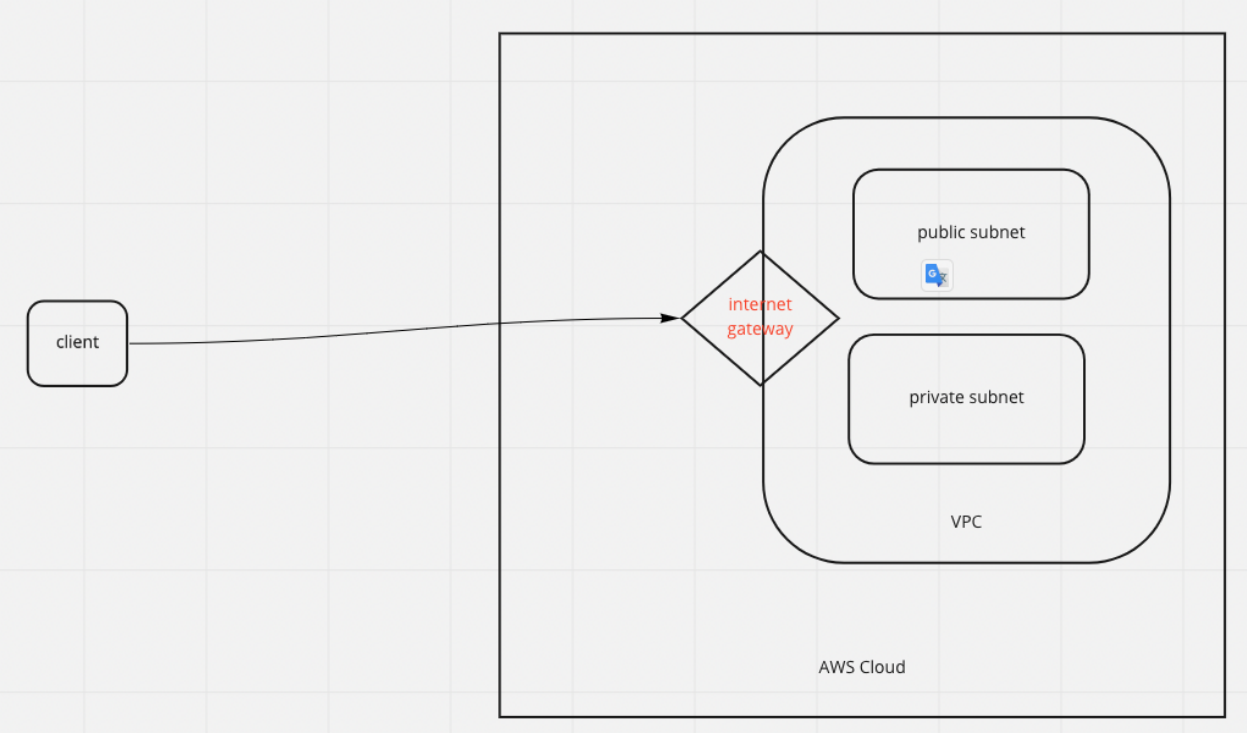
# 4, Networking

## VPC (Amazon Virtual Private Cloud)

*subnet is a section of VPC - contains resources like EC2 instances*

- public subnets
  - can be accessed by public client must be with a Internet gateway
- private subnets
  - only can be accessed by private service and only can access public subnet, and sensitive data can be stored in it

## Internet gateway




in this case

client → VPN → Internet gateway

## Virtual private gateway

## AWS Direct Connect

A service that enables you to establish a dedicated private connection between your data center and VPC.

## ACLs (Networking access control lists ) - security

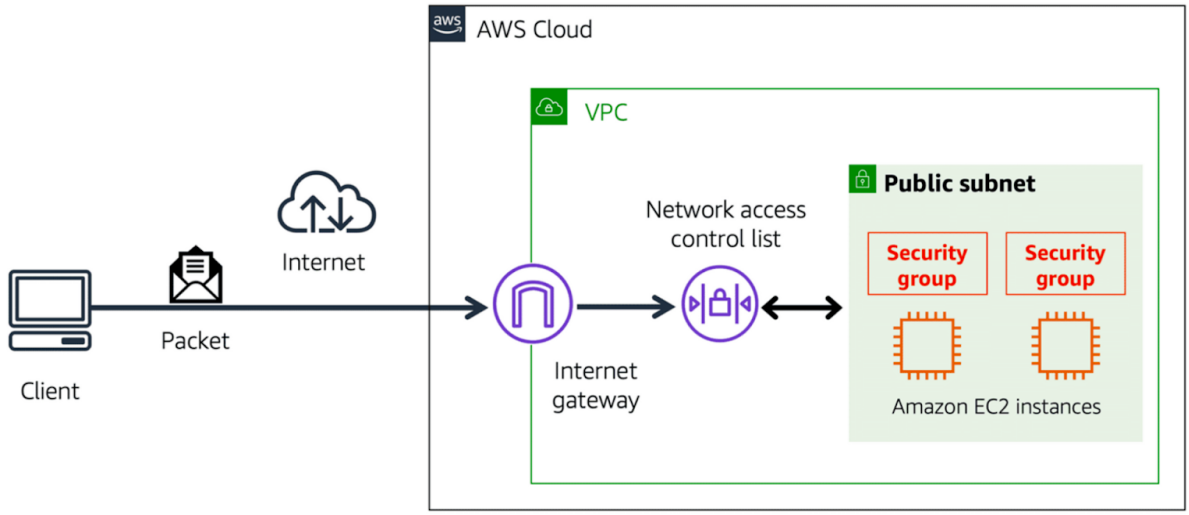
a virtual firewall

- control inbound and outbound traffic at the **subnet level**

## Security group

also work as the virtual firewall

- control inbound and outbound traffic for an Amazon EC2 instance at **instance level**




## Route 53

- is a DNS web service.

# 5, Storage and database

## EBS (Elastic Block Store)(Hard Disk)

- incremental backup  - just update the different part of data
- full backup - each time you do the backup, put all the data in the storage

## S3 (Simple Storage Service) - object storage

- object level storage

s3 storage classes

- s3 standard
- s3 standard- infrequent access
- s3 one zone - infrequent access
- s3 intelligent tiering

- s3 glacier
- s3 glacier deep archive

## Amazon Elastic File storage

- store files - we can mount file system to instance

## RDS

- relational database
- database engines
  - Aurora (serverless)
  - PostgreSQL(not serverless)
  - MySQL
  - MariaDB(similar as MySQL)
  - Oracle Database
  - Microsoft SQL Server

## DynamoDB

key-value pair data store

## Redshift

- used for data warehouse
- big data
- big data analytics

## ElasticCache

- redis
- memocache

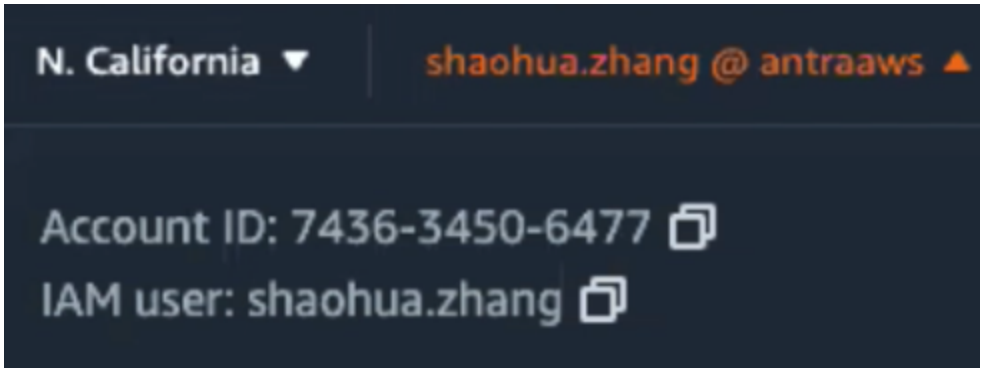
# 6, Security - AWS

shared responsibility model

- customer responsibility (security in the cloud)
- AWS responsibility (security of the cloud)

## IAM - Identity Access Management

Root user → create different IAM users

- IAM users, group, roles
  - 

- IAM policies - set different permissions in IAM

## AWS Organizations

manage your AWS account

# 7, Monitoring and Analytics

## Amazon CloudWatch - Web service

monitor - threshold

- metrics - EC2 instance will send some metrics to CloudWatch → alert to user
- dashboard

## AWS CloudTrail

- record API calls for your account

# 8, Migration

When you do migration, you need to think about 6 perspectives blow:

## perspectives

- business usage
- people
- governance
- platform - other option → google cloud and so on
- Security
- Operations

## physical devices

- AWS Snowcone
- AWS Snowball
- AWS Snowmobile