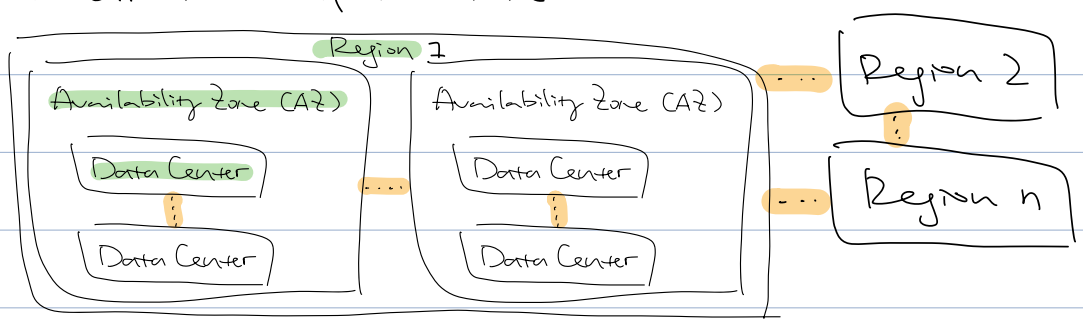
Week 1. Getting started with AWS

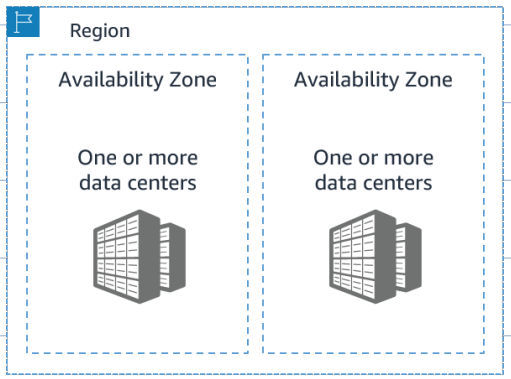
- 6个 Cloud computing 好处：
  - Pay as you go. 不需要先建好 data center, hardware 这些. 需要什么时候就计什么时候计费.
  - Benefit from massive economies of scale. 因为大量 computing 都 aggregate 到 cloud, 有 high economy of scale, 因此 avg cost, pay-as-you-go price 也更低.
  - Stop guessing capacity. 省了 capacity decision 时, 通常 w/ setting on expensive idle resources 或 dealing with limited capacity 状态. 和 cloud 可以 scale up or down as required.
  - Increase speed and agility.
  - No money running / maintaining data center.
  - Go globable in minutes.

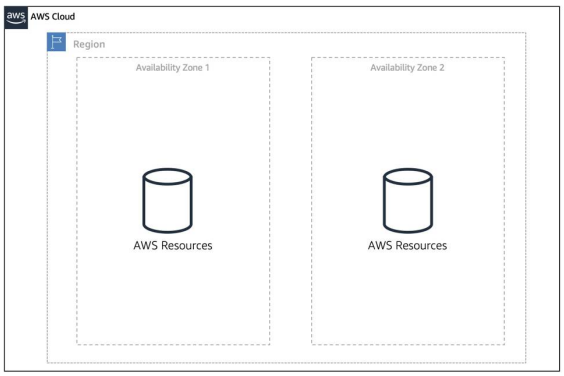
- AWS Globable Infrastructure.



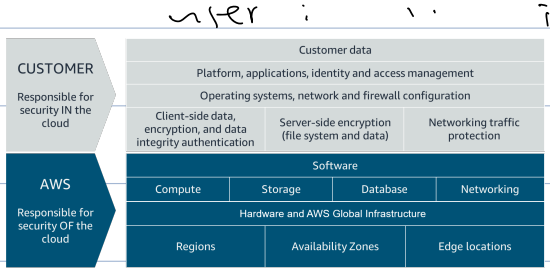
选 Region 考虑：① compliance ② latency ③ pricing ④ service availability

技术上 latency 最大问题. 用 edge cache 来解决.





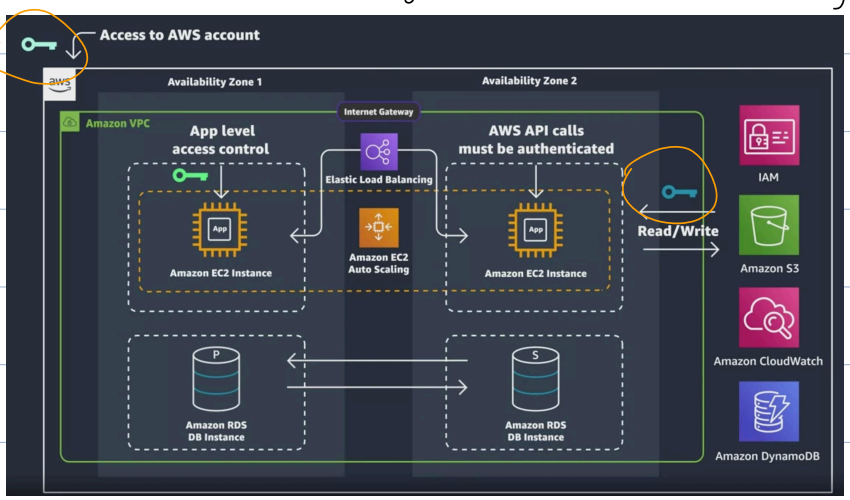
- Interacting with AWS
  - 云上3. 就不能physically manage 3, 所以通过 API (Application Programing Interface).
  - 3 main ways to interact with AWS API.
    1. AWS Mangent Console 我们的. web base
    2. AWS Command Line Interface (CLI)
    3. AWS SW Development Kits (SDKs)
- Security in AWS Clouds
  - user 和 AWS 同时对 Security 负责.
  - 'Shared Responsibility Model'
    - AWS : security of the Cloud
    - user : ... in ...



Category	Examples of AWS Services in the Category	AWS Responsibility
Infrastructure services	Compute services, such as Amazon Elastic Compute Cloud (Amazon EC2)	AWS manages the underlying infrastructure and foundation services.
Container services	Services that require less management from the customer, such as Amazon Relational Database Service (Amazon RDS)	AWS manages the underlying infrastructure and foundation services, operating system, and application platform.
Abstracted services	Services that require very little management from the customer, such as Amazon Simple Storage Service (Amazon S3)	AWS operates the infrastructure layer, operating system, and platforms, as well as server-side encryption and data protection.

Category	AWS Responsibility	Customer Responsibility
Infrastructure services	AWS manages the infrastructure and foundation services.	You control the operating system and application platform, as well as encrypting, protecting, and managing customer data.
Container services	AWS manages the infrastructure and foundation services, operating system, and application platform.	You are responsible for customer data, encrypting that data, and protecting it through network firewalls and backups.
Abstracted services	AWS operates the infrastructure layer, operating system, and platforms, as well as server-side encryption and data protection.	You are responsible for managing customer data and protecting it through client-side encryption.

- Protecting AWS Root Users
  - 当 sign up 一个新的 account 时会创建 root user.
    或 log in
    ↳ great power = great responsibility
  - 用 MFA (multi-factor authentication) 来加强安全
- AWS Identity and Access management (IAM)



• 这个 IAM 管

管理在 AWS 中 action, 和使用 API call.

基本都是 JSON language.

IAM

- Defi:
  - IAM is a web service for managing access to AWS account and resources.
  - It provides centralized control with authentication and authorization. ↳ access 许可证是干什么 ↳ 对资源access
  - It allows to share access without sharing credentials and offers granular access control to resources. 颗粒状的
- Features:
  - gloable, not specific to any region.
  - integrated with many AWS service.
  - supports MFA, identity federation. (e.g. 用手势 x).
- IAM user credentials.
  - access to AWS Management Console.
  - programmatic access to AWS Command Line Interface (AWS CLI) and AWS Application Pyramy Interface (AWS API)
- IAM group. 为了方便管理.
- IAM Policy Example

```
{
"Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": "*",
        "Resource": "*"
    }]
}
```

Vision: policy language 版本

Effect: 这时 statement 会 allow 还是 deny

Action: * 代表全部. 要不然就写读取

Resource: 读时需要 cover 的 obj

Eg.

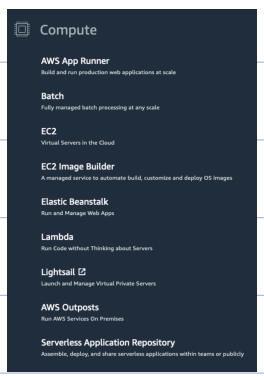
```
{"Version": "2012-10-17",
    "Statement": [{
        "Effect": "Allow",
        "Action": [
            "iam: ChangePassword",
            "iam: GetUser"
            ]
        "Resource":
"arn:aws:iam::123456789012:user/${aws:username}"
    }]
}
```

允许用这个resource干这两件事.

- Role based auess in AWS
  就像给个临时许可证

Week 2: AWS compute & networking

- **AWS Compute**
  - 要host application 要先建 server.
  - server 用来handle HTTP request & send response to user
    HTTP service 常见有：
    - Windows option (Eg. Internet Information Service IIS)
    - Linux option (Eg. Apache HTTP web server, Nginx...)



为了在AWS上run HTTP，
要在AWS Mangnet Console中找
对应 computer power in service (服务)

- EC2. Amazon Elastic Compute Cloud.
  - EC2, a web service provides secure, resizable compute capacity in cloud. Allows to provision virtual service called EC2 instance.
  - To create EC2 instance, need to define:
    ① hardware specifications (Eg. CPU, memory, network, storage ...)
    ② logical configuration (Eg. Networking location, firewall rules, authentication, OS).

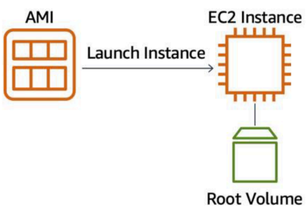
    当launching EC2 instance时，1st 要选择在 Amazon Machine Image (AMI) 里选要用的OS.

• AMI

- 在AWS cloud中 OS installation 不是user的任务了, 而是OS已经在建好的AMI中了. 为什么要instal OS, 在traditional infustructure中, 为了spinning up server.

• Relationship btw AMI & EC2 instance.

EC2 instance are live instantiations of what is defined in an AMI, like the relationship btw class and object.



class is something you model & define.

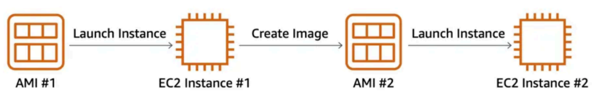
object 〃 〃 interact with.

现在这个情况, class = AMI.

object = EC2 instance (EC2 instance future can install web server & serve user)

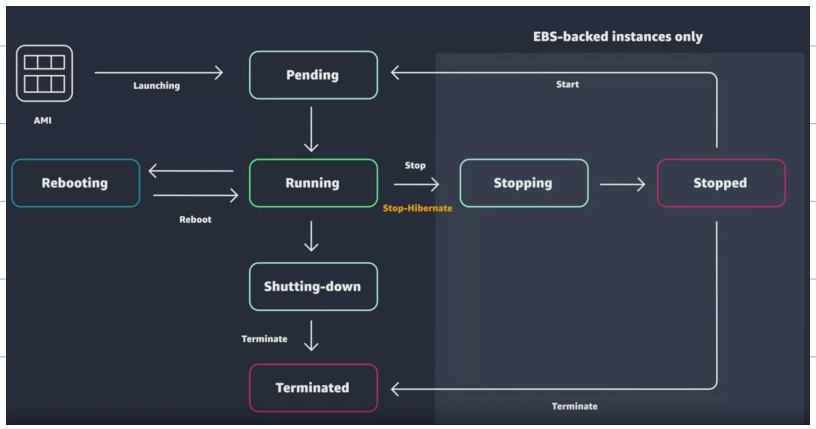
- 当launch new instance时, AWS allocates a virtual machine that runs on a hypervisor. 然后 the AMI you selected is copied to the root device volume, which contains the image used to boot the volume. 最后 you get a server you can connect to and install packages and any additional SW.

- AMI is reusable.



这是之前就有的　　　　这是想要copy的东西.

- Amazon EC2 instance lifecycle.



- terminate 不一定是坏事，例如对 SW update，先 copy 一个到有 update 的 EC2 instance，在这个上做对 update，检查好了后再 terminate 原来那个 instance.
- 只 charge 在 running state 的 EC2 instance.
- **EC2 instance components:**

  virtual processors (vCPU) + memory + network

  (+ instance storage + graphic processing units (GPUs))

- **instance type = prefix identifying + size**

Instance Family	Description	Use Cases
General purpose	Provides a balance of compute, memory, and networking resources, and can be used for a variety of workloads.	Scale-out workloads such as web servers, containerized microservices, caching fleets, distributed data stores, and development environments.
Compute optimized	Ideal for compute-bound applications that benefit from high-performance processors.	High-performance web servers, scientific modeling, batch processing, distributed analytics, high-performance computing (HPC), machine/deep learning, ad serving, highly scalable multiplayer gaming.
Memory optimized	Designed to deliver fast performance for workloads that process large data sets in memory.	Memory-intensive applications such as high-performance databases, distributed web-scale in-memory caches, mid-size in-memory databases, real-time big-data analytics, and other enterprise applications.
Accelerated computing	Use hardware accelerators or co-processors to perform functions such as floating-point number calculations, graphics processing, or data pattern matching more efficiently than is possible with conventional CPUs.	3D visualizations, graphics-intensive remote workstations, 3D rendering, application streaming, video encoding, and other server-side graphics workloads.
Storage optimized	Designed for workloads that require high, sequential read and write access to 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 that replicate their data across different instances.	NoSQL databases, such as Cassandra, MongoDB, and Redis, in-memory databases, scale-out transactional databases, data warehousing, Elasticsearch, and analytics.

(VPCs)

- By default, EC2 instance 存在于 Amazon Virtual Private Cloud 有效减少各种 transmission cost，当熟悉 AWS 之后，可以设地址到) customer VPC，这样能 restrict access with additional routing and connectivity mechanism.
- Architect for High Availability:

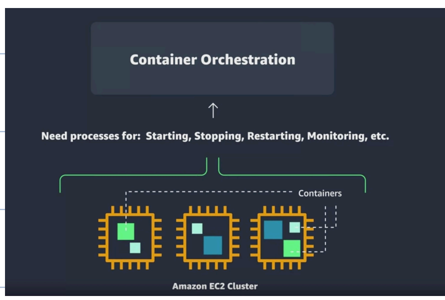
  在 network 中，instance 存在于自己选的 availability zone 中.

  在 AZ 中的 AWS service 必须 be architect of high availability.

  所以在上面 EC2 instance component 中有 size.
- An EC2 instance transmissions btw diff states 从生到死.
- Reverse Capacity with Reversed Instance (RI)

  RI 比 on-Demand instance 更省钱.

- Container Services on AWS
  - container orchestration services:
    - Amazon Elastic Container Service (Amazon ECS)
    - 〃 〃 Kubernetes 〃 (Amazon EKS)



当 container 数量太大时就很难 manage → EKS

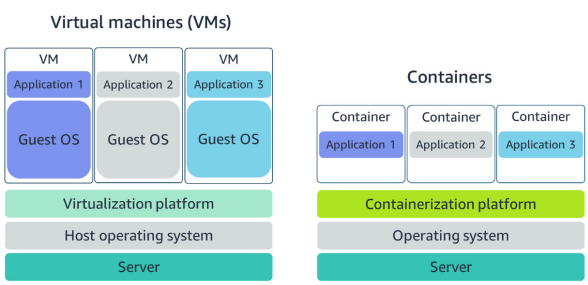
Container

· def: containers are used as a solution to problems of traditional compute, including the issue of getting SW to run reliably when it moves from one compute environment to another. It's a standardized unit that packages up your code and all of its dependents.

Docker

Docker is a popular container runtime that simplifies the management of the entire OS stack needed for container isolation, including networking and storage. Docker makes it easy to create, package, deploy and run.

WHAT IS THE DIFFERENCE BETWEEN CONTAINERS AND VMS?



Orchestrate Containers

在AWS中, containers run on EC2 instance. 常常有在 instance 中run好几个 containers. 都在一个 instance 上run 是好管理, 但 lack of availability and scalability. 一般会部署多个 container 在多个 instance 上run, across 多个 AZ.

当manage compute at a large scale时，need to know：

① How to place containers on instance.

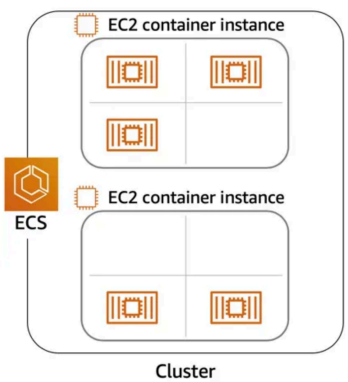
② What happens if container fails.

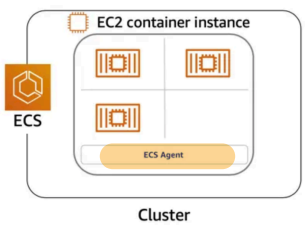
③ ` ` instance ` `.

④ How to monitor deployments of containers.

MANAGE CONTAINERS WITH AMAZON ELASTIC CONTAINER SERVICE (AMAZON ECS)

Amazon ECS is an end-to-end container orchestration service that allows you to quickly spin up new containers and manage them across a cluster of EC2 instances.





To run & manage, need a ECS Agent.

To prepare your application to run on ECS, need to create task definition in JSON.

Eg. simple task definition runs on Nginx web server.

```
{
    "family": "webserver",
    "containerDefinitions": [ {
        "name": "web",
        "image": "nginx",
        "memory": "100",
        "cpu": "99"
    } ],
    "requiresCompatibilities": [ "FARGATE" ],
    "networkMode": "awsvpc",
    "memory": "512",
    "cpu": "256"
}
```

(>kubernetes

跟ECS对应的EKS. EKS与ECS是conceptually similar，对应关于：

① EC2 instance with ECS Agent installed and configured 的 container instance. With EKS 的 worker node.

② ECS container 的 task. 在EKS ecosystem，的 pod.

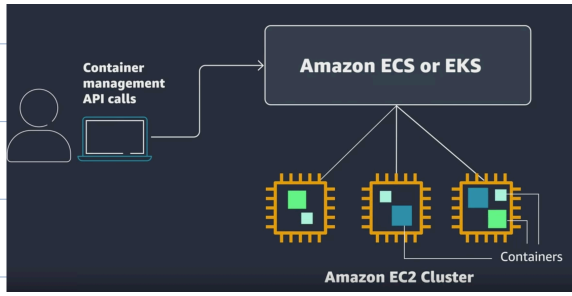
③ ECS runs on AWS native technology. EKS runs on top of kubernetes.

- Serverless.

只需要关注你在意的，余下的交给 AWS.

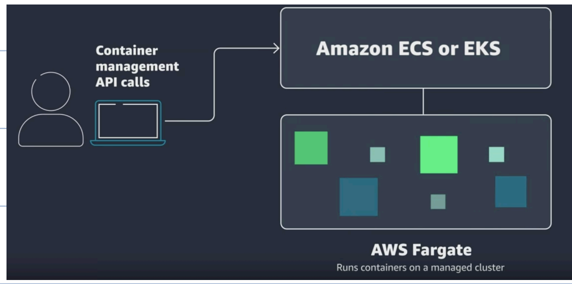
- Serverless with AWS Fargate.

之前：



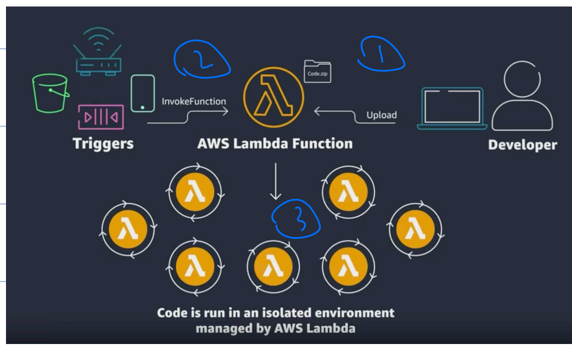
EC2 肯定不是 serverless

现在：



Fargate 是 serverless

- AWS Lambda



①：自己写 lambda function.

②：当有 trigger 激活时会开始 call lambda function.

③：可以有很多个 λ func，都是 isolated.

Lambda func 不适用于须需长期在线的 action，而是等到条件被激活后使用的

coursera 视频后面有 sample 教程.

tiny summary：AWS compute service ∈ { EC2, ECS/EKS, Lambda }

AWS Networking

Networking on AWS

- **Networking**: how you connect computers around the world and allow them to communicate with one another.

Amazon VPC (Virtual Private Cloud)

- VPC create walls/boundaries where application and resources is isolated from outside.
- To 创建 VPC 时，需要 ① region ② IP Range in CIDR notation
  To 创建 subnet，need：① VPC ② AZ ③ IP Range

- **CIDR** (Classless Inter-Domain Routing) notation.

  E.g. 192.168.1.0/24

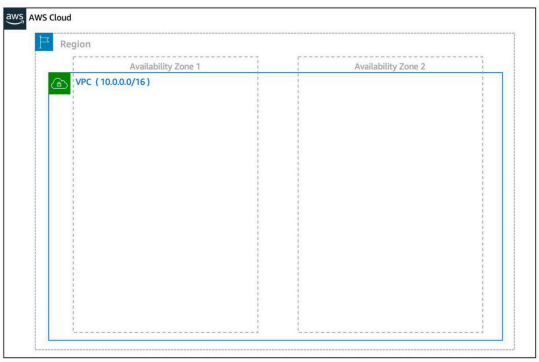
  fixed flexible 从 00000000 开始 24 个 bits 都 fixed.

  总共 32-24 = 8 个 bits flexible，有 2^8 = 256 个 IP 可以用.

∴ .0/24 的 range 比 .0/16 range 小. (2^{32-16} = 65,536)

- Eg. 实操 Amazon VPC

Step 1: create VPC



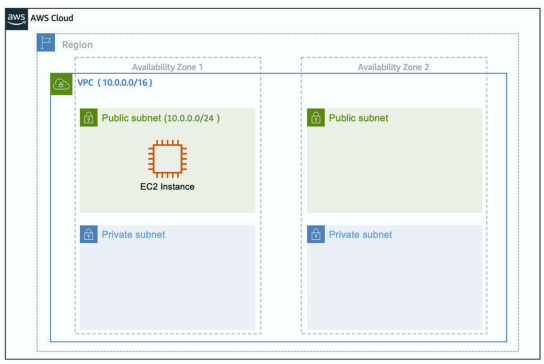
要：

① name of VPC

② Region of VPC to live on

③ IP Range for VPC in CIDR

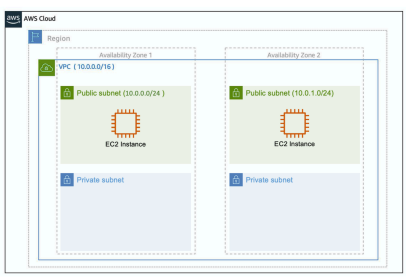
Step 2: create Subnet



- Subnets = smaller networks inside base network or a virtual area networks (VLANs)
- 在AWS中，subnet帮助 high availability and privacy of connectivity options.

要：① VPC want to live in
② AZ want to live in
③ CIDR block of subnet

- High Availability with a VPC

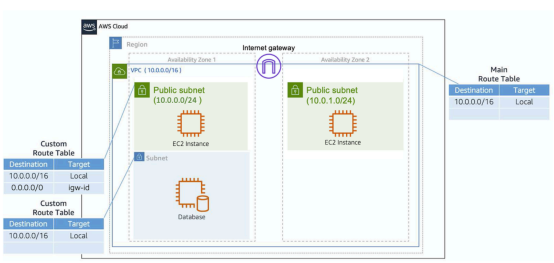


copy paste 相同的.

- Reserved IP.

这些 reserved 用于 ① routing ② DNS (Domain Name System) ③ networking management.

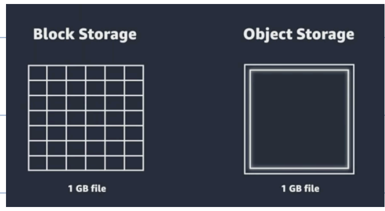
- Gateway
  - Internet Gateway - (IGW)
    To ensure internet connectivity for VPC.
  - Virtual Private Gateway (VGW)
    Allows to connect AWS VPC to another private network.
- AWS VPC Routing and Security.
  - Main Route Table.
    当创建 VPC 时. AWS 自动生成 main route table, 用来 determine where network traffic is directed. AWS 会默认当创建 VPC with subnets 时, you want traffic to follow between them.
  - Custom Route Table
    want to be more granular abt how you route your traffic for specific subnets. 当有 custome route table 时, 就不会用 main route table. 默认 by default, custom 会创建 local 如上.

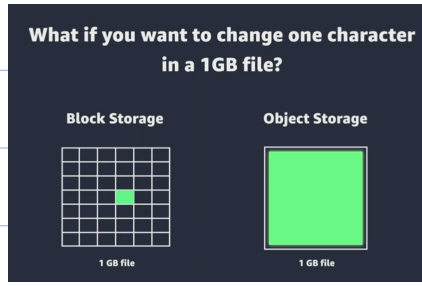


Week 3: Storage & Databases on AWS

- **Storage on AWS**
  - Storage Types on AWS

    ① block ② object ③ file





只用改一个 block　　　需要改整个档

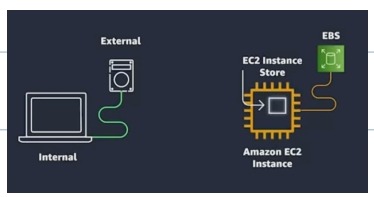
→ object storage often follow a WORM model ↩ write once, read many

- File Storage
  - ideal when require a centralized access to files that need to be easily shared and managed by multiple host computers.
  - typically, this storage is mounted onto multiple hosts and requires file locking and integration with existing file system communication protocols.
  - common use : ① large content repositories
    ② development environment
    ③ user home directories
- Block Storage
  - Split files into file-size chunks of data cell (block) that have their own address, so each block can be retrieved effectively
  - is optimized for low latency operations, it's a typical

storage for high-performance enterprise workloads, such as databases or enterprise resource planning (ERP) systems.

- Object Storage.
  - Stored in a flat structure instead of hierarchy. Each obj is a file with unique identifier. Indentifier, along with any additional metadata, is bundled with the data and stored.
  - Can store almost every type of data, and no limit to the number of obj stored, making it easy to scale.
  - 常用于 storing large data sets, unstructed files.

- Amazon EC2 Instance Storage and Amazon Elastic Block Storage.



EC2 Instance Storage:

∪: Fast

∩: life cycle bundled with instance, so it's often refered as ephemeral storage.

EBS:

∪: 像个硬盘. ① life cycle 不受限 ② 就在同个AZ中的某个instance用.
③ 就多个instance同时用, EBS Multi-Attach.

- EBS.
  - Block-level storage can attach to EC2 instance.
  - Storage devices are called Amazon EBS volumes. (存多种卷儿)
  - Scale EBS 有两个方法:
    ① Increase volume size, 但是有size limit (16 TB).
    ② Attach multiple volumes to single Amazon EC2 instance.
  - Use case:
    ① OS ② DB ③ enterprise app ④ throughput-intensive application.

Amazon EBS Volume Types

	EBS Provisioned IOPS SSD	EBS General Purpose SSD	Throughput Optimized HDD	Cold HDD
Description	Highest performance SSD designed for latency-sensitive transactional workloads	General purpose SSD that balances price and performance for a wide variety of transactional workloads	Low-cost HDD designed for frequently accessed, throughput intensive workloads	Lowest cost HDD designed for less frequently accessed workloads
Use Cases	I/O-intensive NoSQL and relational databases	Boot volumes, low-latency interactive apps, development, and test	Big data, data warehouses, log processing	Colder data requiring fewer scans per day
Volume Size	4 GB-16 TB	1 GB-16 TB	500 GB-16 TB	500 GB-16 TB
Max IOPS/Volume	64,000	16,000	500	250
Max Throughput/Volume	1,000 MB/s	250 MB/s	500 MB/s	250 MB/s

- Obj Storage with Amazon S3
  - Amazon S3 (an object storage service)
    - it's a standalone storage that isn't tied to compute. It enables retrieving data from anywhere else on the web.
    - In S3中, obj 都存在 container called buckets.
      Steps: ① 选 region ② bucket name. must be unique across all AWS accounts.

http://doc.s3.amazonaws.com/2006-03-01/AmazonS3.html
Bucket
Object/Key

- Use case :

① backup & storage ② media sharing ③ SW delivery

④ data lakes ⑤ static website ⑥ static content.

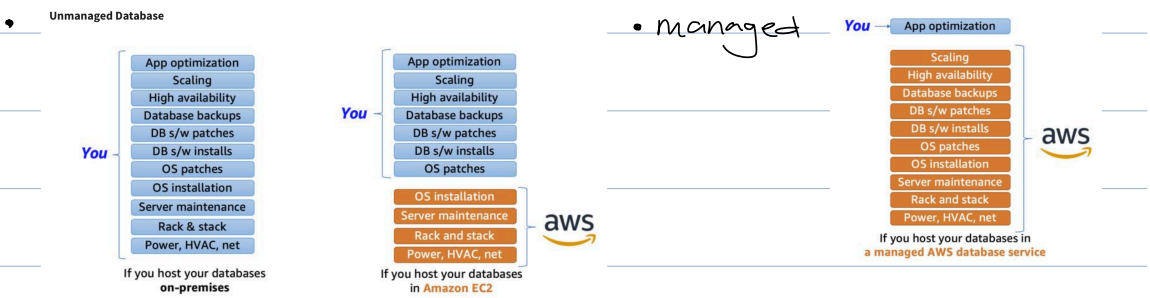
- S3 bucket policy

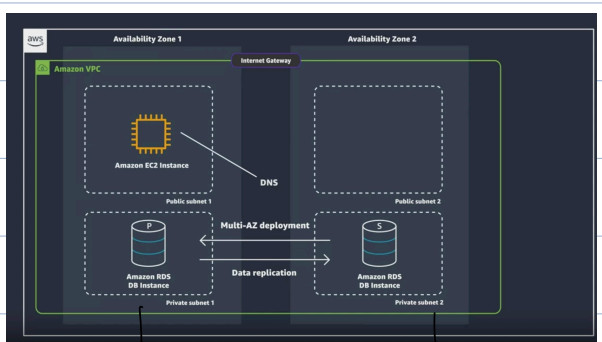
```
{
  "Version":"2012-10-17",
  "Statement":[
    {
      "Sid":"PublicRead",
      "Effect":"Allow",
      "Principal": "*",
      "Action":["s3:GetObject"],
      "Resource":["arn:aws:s3:::employeebucket/*"]
    }
  ]
}
```

- Encrypt S3 :
  - Server side encryption : encrypt when saving / decrypt when downloading.
  - Client-side encryption : 自己搞.
- Use Versioning to Preserve Objects.
- Amazon S3 Storage Classes.
  1. S3 Standard
  2. S3 Intelligent-Tiering
  3. S3 Standard-Infrequent Access
  :

- Choose the right storage service.
  - EC2 Instance Storage
  - EBS
  - S3
  - EFS (Elastic File System) & FSx

- Database on AWS
  - Database on AWS
    - Relational DB (MySQL. Amazon Aurora) → super fast & large volume
      - ☺ : ① joins ② reduced redundancy ③ familiarity ④ accuracy.

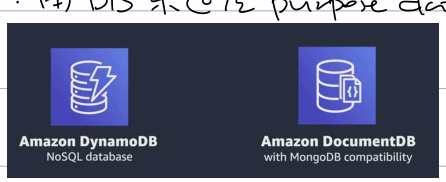


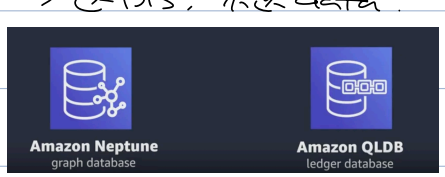


↓ Primary DB ↓ Secondary DB

当前个AZ中的DB连不上了就连 Secondary DB.

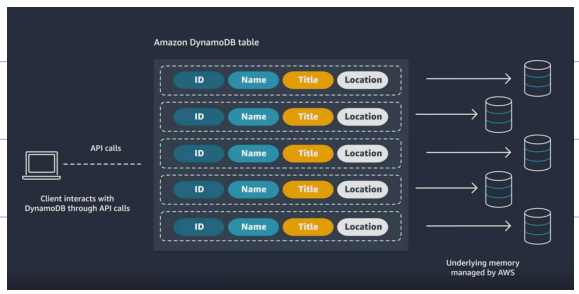
- Amazon Relational DB Service (RDS).
  - Types:
    1. commercial: Oracle. SQL Server
    2. open source: MySQL. MariaDB. PostgreSQL
    3. cloud native: Amazon Aurora.
  - DB instance:
    1. Standard 2. memory optimized 3. burstable performance.
  - EBS (Elastic Block Storage) volume storage:
    1. General Purpose (SSD)
    2. Provisioned IOPS (SSD)
    3. Magnetic storage (not recommend)
  - Work with Amazon RDS in Amazon Virtual Private Cloud.
  - To create DB subnet group:
    1. AZ include the subnet want to added
    2. subnet in that AZ where DB instances are placed.
  - 用 AWS Identity and Access Management (IAM) 来控制访问
  - Backup data: 1. automatic backup
    2. manual snapshots
- Purpose Built DB on AWS
  - 用 DB 来适应 purpose data. → 选 DB, 不选 data.





- Amazon DaynoDB

不用自己建 table 或 index，只需要放 data 就行



- 不用自己 maintance
- Amazon DaynoDB 每天处理 10 trillion+ request

Database Type	Use Cases	AWS Service
Relational	Traditional applications, ERP, CRM, e-commerce	Amazon RDS, Amazon Aurora, Amazon Redshift
Key-value	High-traffic web apps, e-commerce systems, gaming applications	Amazon DynamoDB
In-memory	Caching, session management, gaming leaderboards, geospatial applications	Amazon ElastiCache for Memcached, Amazon ElastiCache for Redis
Document	Content management, catalogs, user profiles	Amazon DocumentDB (with MongoDB compatibility)
Wide column	High-scale industrial apps for equipment maintenance, fleet management, and route optimization	Amazon Keyspaces (for Apache Cassandra)
Graph	Fraud detection, social networking, recommendation engines	Amazon Neptune
Time series	IoT applications, DevOps, industrial telemetry	Amazon Timestream
Ledger	Systems of record, supply chain, registrations, banking transactions	Amazon QLDB

Week 4: Monitoing, Optimizing and Going Servie on AWS

- **Monitoing on AWS**
  - Monitor before user notice
  - U of Monitoing:
    1. respond to operational issue proactsivity before you end user are aware of them.
    2. improve the performance and relicbility of your resources.
    3. relognize securisy threats and events
    4. create mole cost-effective solutsons.
  - Eable visibilty.
    - Detect anomalous behavior in your environments.
    - Set alarms to alert you when something's not right.
    - Visualize logs and metrics with the AWS Management Console.
    - Take automated actions like scaling.
    - Troubleshoot issues.
    - Discover insights to keep your applications healthy.

- Amazon CloudWatch.
  - For Basic Monitoing & Detailed Monitoing.
    - Break down metrics namespace. dimensions.

Learn the CloudWatch Logs Terminology

Log data sent to CloudWatch Logs can come from different sources, so it's important you understand how they're organized and the terminology used to describe your logs.

Log event: A log event is a record of activity recorded by the application or resource being monitored, and it has a timestamp and an event message.

Log stream: Log events are then grouped into log streams, which are sequences of log events that all belong to the same resource being monitored. For example, logs for an EC2 instance are grouped together into a log stream that you can then filter or query for insights.

Log groups: Log streams are then organized into log groups. A log group is composed of log streams that all share the same retention and permissions settings. For example, if you have multiple EC2 instances hosting your application and you are sending application log data to CloudWatch Logs, you can group the log streams from each instance into one log group. This helps keep your logs organized.

- Optimization
  - Optimizing Solutions on AWS
    - Availability.

Availability (%)	Downtime (per year)
90% ("one nine")	36.53 days
99% ("two nines")	3.65 days
99.9% ("three nines")	8.77 hours
99.95% ("three and a half nines")	4.38 hours
99.99% ("four nines")	52.60 minutes
99.995% ("four and a half nines")	26.30 minutes
99.999% ("five nines")	5.26 minutes

    - Improve Application Availability.
    - Use 2nd AZ
    - Manage Replication, Redirection, and High Availability.
  - Amazon EC2 Auto Scaling.
  - Route Traffic with Amazon Elastic Load Balancing.
    - Load Balancer

- **FEATURES OF ELB**

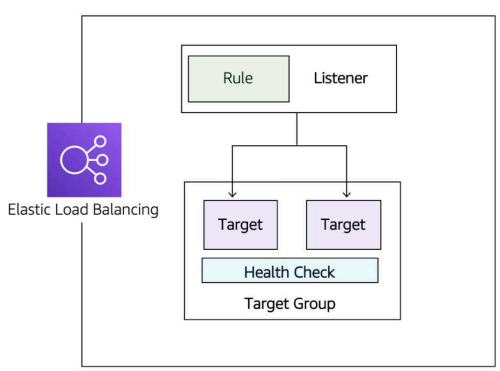
The ELB service provides a major advantage over using your own solution to do load balancing, in that you don't need to manage or operate it. It can distribute incoming application traffic across EC2 instances as well as containers, IP addresses, and AWS Lambda functions.

- The fact that ELB can load balance to IP addresses means that it can work in a hybrid mode as well, where it also load balances to on-premises servers.
- ELB is highly available. The only option you have to ensure is that the load balancer is deployed across multiple Availability Zones.
- In terms of scalability, ELB automatically scales to meet the demand of the incoming traffic. It handles the incoming traffic and sends it to your backend application.

    - Health check

- **ELB COMPONENTS**

The ELB service is made up of three main components.



- **Listeners:** The client connects to the listener. This is often referred to as client-side. To define a listener, a port must be provided as well as the protocol, depending on the load balancer type. There can be many listeners for a single load balancer.
- **Target groups:** The backend servers, or server-side, is defined in one or more target groups. This is where you define the type of backend you want to direct traffic to, such as EC2 Instances, AWS Lambda functions, or IP addresses. Also, a health check needs to be defined for each target group.
- **Rules:** To associate a target group to a listener, a rule must be used. Rules are made up of a condition that can be the source IP address of the client and a condition to decide which target group to send the traffic to.

Network Load Balancer (NLB)

NLB uses a flow hash routing algorithm. The algorithm is based on:

- The protocol
- The source IP address and source port
- The destination IP address and destination port
- The TCP sequence number

If all of these parameters are the same, then the packets are sent to the exact same target. If any of them are different in the next packets, then the request may be sent to a different target.

NLB has sticky sessions. Different from ALB, these sessions are based on the source IP address of the client instead of a cookie.

NLB supports TLS offloading. NLB understands the TLS protocol. It can also offload TLS from the backend servers similar to how ALB works.

NLB handles millions of requests per second. While ALB can also support this number of requests, it needs to scale to reach that number. This takes time. NLB can instantly handle this amount of requests.

NLB supports static and elastic IP addresses. There are some situations where the application client needs to send requests directly to the load balancer IP address instead of using DNS. For example, this is useful if your application can't use DNS or if the connecting clients require firewall rules based on IP addresses. In this case, NLB is the right type of load balancer to use.

NLP preserves source IP address. NLB preserves the source IP address of the client when sending the traffic to the backend. With ALB, if you look at the source IP address of the requests, you will find the IP address of the load balancer. While with NLB, you would see the real IP address of the client, which is required by the backend application in some cases.