AWS Certified Cloud Practitioner

Gi Wah Dávalos Loo

2023

1 Cloud Computing

1.1 What is

Cloud computing is the **on-demand delivery** of it resources (database storage, compute power, etc), which you can access and **provision the right type and size of computing almost instantly** and have a **pay-as-you-go** pricing.

1.2 Deployment Models of the Cloud

1. **Private:**
   - Complete control, meet specific business needs
   - Used by single organization
   - Cloud owned by third-party
2. **Public:**
   - Cloud resources owned and operated by third-party.
3. **Hybrid:**
   - Keep the control of some servers and extend some capabilities to the Cloud

1.3 Five Characteristics of Cloud Computing (Amazon POV)

- **On-demand self service:** Users can provide it resources without human interaction.
- **Broad network access:** Can access the AWS panel from the internet.
- **Multi-tenancy and resource pooling:** Multiple users share the same physical resources yet their it resources are isolated with security and privacy.
- **Rapid elasticity and scalability:** Automatically and quickly add or remove it resources.
- **Measured service:** Usage is measured and you pay what you use.

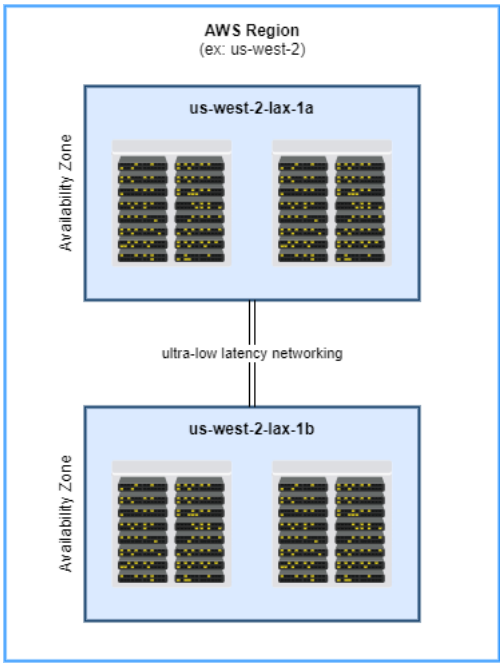
1.4 Advantages of Cloud Computing

- **Trade CAPEX for OPEX:** Pay on-demand don't own hardware.
- **Benefit from economies of scale:** If more people use AWS, AWS will acquire more hardware and become more efficient, then the pricing goes lower.
- **Use what you need:** Scale based on actual usage measure.
- **Increase speed and agility:** Instance resources almost immediately.
- **Globality:** Instance IT resources in many geographical locations.

1.5 Problems solved by the Cloud (Business with IT needs POV)

- **Flexibility:** Add, remove and change IT resources when needed.
- **Cost-effectiveness:** Pay as-you-go.
- **Scalability:** Increase IT resources when receiving larger loads.
- **Elasticity:** Scale-in and scale-out when needed.
- **High-availability and‘ Fault-tolerant:** You gotta trust AWS.
- **Agility:** Quick development process, test and launch software applications.

1.6 AWS Global Infrastructure



2 IAM

2.1 What is

IAM **(Identity and Access Management)** is a **global** AWS service that helps you manage the permissions of the access to AWS services and resources.

2.2 Users, Groups, Roles, Policies

2.2.1 Users

People within the organization. One AWS user account per physical user. You can assign policies to a user.

2.2.2 Groups

Can contain users only (not other groups). You can assign policies to a group.

2.2.3 Roles

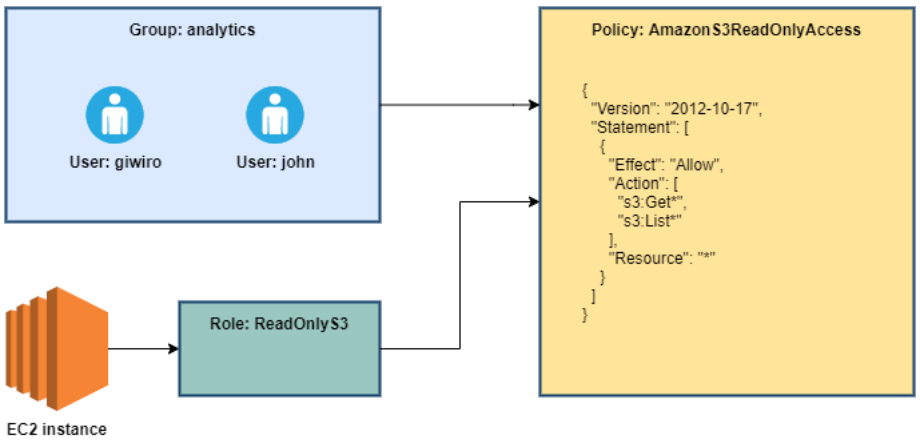
In order to bind a policy to an AWS service or resource (EC2, Lambda, etc), we need to it through a role.

2.2.4 Policies

JSON document that defines the access level of AWS services and resources. Example:

```
// AmazonEC2ReadOnlyAccess
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": "ec2:Describe*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "elasticloadbalancing:Describe*",
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": [
                "cloudwatch:ListMetrics",
                "cloudwatch:GetMetricStatistics",
                "cloudwatch:Describe*"
            ],
            "Resource": "*"
        },
        {
            "Effect": "Allow",
            "Action": "autoscaling:Describe*",
            "Resource": "*"
        }
    ]
}
```

2.3 IAM Elements Relationships



2.4 Access AWS

- **Management Console:** Web portal. Protected by password + MFA (Optional, but recommended)
- **CLI:** Command Line tool. Protected by access keys.
- **SDK:** Code library. Protected by access keys.

2.5 IAM Security Tools

- **IAM Credentials Report:** Account level tool that lists all users and the status of their credentials.
- **IAM Access Advisor:** User level tool. List of all permissions (policies) per service and when it was last accessed.

2.6 IAM Share responsibility model

- Management and monitoring of users, groups, roles and policies
- Enable MFA on all accounts
- Rotate all the keys often
- Analyze access patterns & review permissions

3 EC2

3.1 What is

EC2 (**Elastic Compute Cloud**) is a IaaS service wich spawns an instance of selected OS. It can be configured with a custom script at start in the **User Data** option in Step 3 of the creation of an EC2.

3.2 Security Groups

Firewall security rules that control how traffic is allowed in and out the EC2 instance. By default all ports are closed, except for those ports allowed in the security groups.

3.3 Connecting to EC2

- **SSH:** Using a ssh client, the public EC2 ip and the private key from the key pair creation.
- **EC2 Instance Connect:** Web platform that creates a disposable key-pair for access to the EC2. If you EC2 ssh ports are not allowed, then you won't be able to connect through Instance Connect.

3.4 EC2 Instance Purchase options

- **On Demand:** Billing per second, **highest cost** but no long commitment or upfront payment.
- **Reserved: 75% discount** but has reservation period commitment of 1 or 3 years (more years more discount).
  - Convertible: Every characteristic of the EC2 (family, instance type, platform, etc) can be changed.
  - Standard: Some attributes such as size, chan be changed but only on the existing instance (family, instance type or platform, for instance, cannot be changed).
- **Scheduled:** Reservation that recur on a daily, weekly or monthly basis with **fixed start time and duration** for **one year term**.
- **Spot Instances: 90% discount** but **it can be terminated at any time** (if your max price is bigger than the spotted price)
- **Dedicated Hosts:** Physical server with EC2 instance capacity dedicated to your use. Helps to address **compliance requirements** and allow to **use your existing server-bound software license**. Allocation is a **3 year commitment** and is more expensive.
- **Dedicated Instances:** Instances running on **hardware dedicated to you**.

3.5 EC2 Shared responsibility model

- Manage Security Groups rules
- OS patches & updates
- Software installed in the EC2
- IAM Roles assigned to the EC2
- Data security on your instance

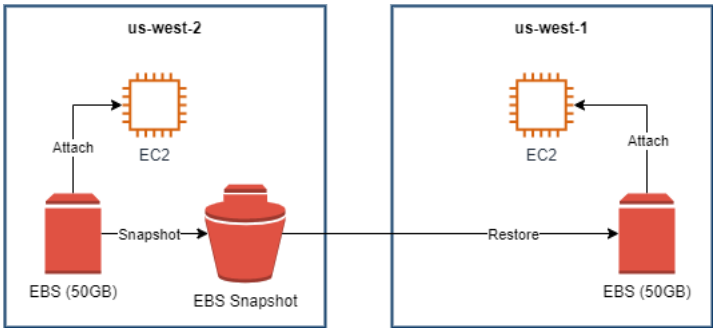
4 EBS

4.1 What is

EBS (Elastic Block Store) Volume is a network drive that can be attached to a EC2 while they run. EBS can only be **assigned to one EC2 at a time**, they it is bound to a **specific availability zone** and it has a provisioned capacity in GBs and IOPS.

4.2 EBS Snapshots

Snapshots are **backups a EBS Volume at a point in time**, and can be copied across regions/AZ. Since you cannot move an EBS across regions/AZ, you can create a Snapshot in the region, and then create a new EBS from the snapshot in the desired region/AZ.



4.3 AMI

It stands for Amazon Machine Image and are a **customization** of an EC2 instance (With additional software, config, OS, etc). they are built for a **specific region** but can be copied across regions. You can launch EC2 instances from:

- Public AMI (AWS provided)
- Your own AMI (maintained by you development team
- AWS Marketplace AMI (Someone else's AMI)

4.4 EC2 Instance Store

Ephemeral storage (Data is lost upon restart) that is located in the same physical location of the EC2 instance and provides a **better I/O performance**. It's good for buffering and caching but it's **more expensive**.

4.5 EFS

Managed NFS (network file system) that **can be bond to multiple EC2 across multiple AZ**. It is **HA, scalable and very expensive**.

4.6 EC2 Storage Shared responsibility model

- Setting backup / snapshot procedures
- Setting up data encryption
- Responsibility of any data on the volumes
- Understand the risk of using EC2 Instance Store

5 ELB & ASG

5.1 Scalability

Ability to accommodate a larger load by making the hardware stronger (vertical) or by adding nodes (horizontal).

5.1.1 Vertical scaling

By adding more hardware capacity, the instances can handle greater load of requests. For example, change from a *t2.micro* to a *t2.large*. It is very common for non-distributed systems such as databases. You **scale up** when the hardware capacity increases, and **scale down** when decreases.

5.1.2 Horizontal scaling

By adding the number of instances you can handle greater load of requests, and it implies using distributed systems, such as modern applications. You **scale out** when the number of instances increases, and **scale in** when decreases.

5.2 High Availability

Instances are across multiple Availability Zones (AZ). Using multi AZ ELB and ASG. (Usually related to horizontal scaling)

5.3 Elasticity

Once a system is already scalable, elasticity means that you can dynamic and automatically scale based on the load.

5.4 Agility

(NOT RELATED, MEANT TO DISTRACT) Cloud capacity to provide IT resources without human interaction (one click away).

5.5 ELB

Elastic Load Balancer (ELB) are servers that forward internet traffic to multiple EC2 instances downstream.

5.5.1 Why?

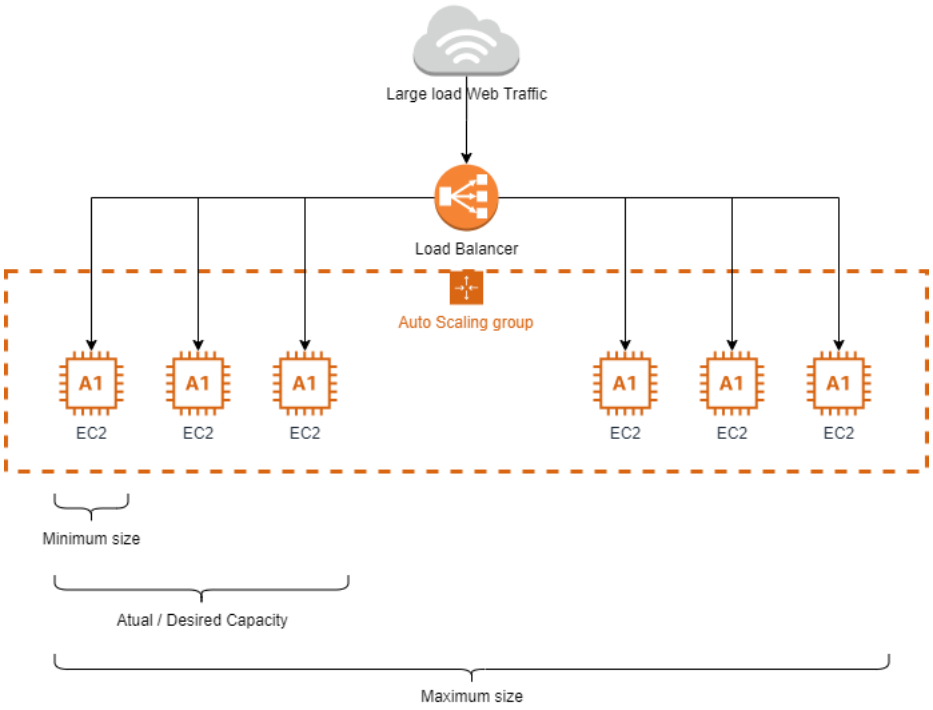
- Spread load across multiple downstream instances
- Expose single point of access to your application
- Seamlessly handle failures of downstream instances
- Do regular health check on instances
- SSL termination
- Is a manged load balancer (AWS oversees)

5.5.2 Kinds of load balancers

- Application Load Balancer - Layer 7 (HTTP Only)
- Network Load Balancer - Layer 4 (Raw TCP)

5.6 ASG

Auto Scaling Group (ASG) are a collection of EC2's that ensure to have a minimum and a maximum number of machines running. It automatically registers new instances to a load balancer. When it needs more instances (configured logic based on hardware statistics) it **scales out**. It also replaces unhealthy instances.



6 S3

6.1 What is

One of the building blocks of AWS that is advertised as **infinitely scaling** storage.

6.2 Anatomy

S3 allows people to store **objects** ("files") in **buckets** which must have a **global unique name** but they are **defined at region level**. All the objects have a **key** that is the full path to the object, e.g. *s3://my-bucket/my_file.txt* and have a **max size of 5TB**.

6.3 Security

6.3.1 User based:

IAM policies $attachedtotheuserorgroupthatwillhaveaccess$

6.3.2 Resource Based:

- **Bucket Policies:** JSON based rules $SameasIAMPolicy - allowcomplexrules$
- **Object Access Control List (ACL):** Simple rules (ALLOW/DENY) over single object, finer-grain control.
- **Bucket Access Control List (ACL):** Simple rules over the whole bucket.

6.3.3 Encryption

6.4 Websites

S3 can host static websites. The website URL will be: ¡bucket-name¿.s3-website-¡aws-region¿.amazonaws.com

6.5 Versioning

The S3 versioning is enabled at **bucket level** and it keeps versions of all the objects in S3 when they are changed. Why use versioning?

- Protect against unintended deletes
- Easy roll back

6.6 Access Log

Ability to **log all access** to S3 buckets (maybe for auditing purposes), e.g. any request made to S3, authorized or denied.

6.7 Replication: CRR &SRR

Allow to duplicate in real time an S3 bucket. In order to achieve that you **must enable versioning** in source and destination.

6.7.1 CRR use cases

- Compliance (high available buckets in case of disaster
- Replication across accounts
- Low latency access

6.7.2 SRR use cases

- Aggregate log
- Live replication from production to test account

6.8 Storage Classes

S3 has **High durability (99.999999999%)** for all storage classes.

- **Standard - General Purpose**: 99.99% available.
- **Standard - Infrequent Access (IA)**: 99.9% available, lower cost but has retrieval fee.
- **One Zone - Infrequent Access**: 99.5% available, lower cost than IA but it is located in a single AZ and has retrieval fee.
- **Intelligent Tiering**: 99.9% available, cost optimized by moving between FA and IA.
- **Glacier**: 99.9% available, data is retained for longer terms (year) and has retrieval fee. Retrieval time:
  - Expedited (1 to 5 minutes).
  - Standard (3 to 5 hours).
  - Bulk (5 to 12 hours).
- **Glacier Deep Archive**: 99.9% available, data is retained for longer terms (year) and has retrieval fee. Retrieval time:
  - Standard (12 hours).
  - Bulk (48 hours).

6.9 Shared Responsibility Model

- S3 Versioning.
- S3 Bucket Policies.
- S3 Replica setup.
- Loggin and Monitoring.
- S3 Storage classes.
- Data encryption at rest and in transit.

6.10 Snowball, Snowball Edge, Snowmobile

All three services use physical data transportation that helps to move great amount of data in or out of AWS. You pay per data transfer job.

6.10.1 Snowball:

Can be used to large data cloud migration, disaster recovery.

6.10.2 Snowball Edge:

100TB capacity and has computational capability (can process data on the go). It supports EC2 AMI and Lambda functions.

6.10.3 Snowmobile:

Transfer exabytes of data. Best solution if you're going to transfer more than 10PB.

6.11 Storage Gateway

S3 can be used as a *"hybrid cloud"* set up. This means part of you infrastructure is on-premise and part of it is on the cloud. This allows the on-premise storage to seamlessly use the AWS Cloud.

7 Databases

7.1 RDS

Relational Database Service (RDS) is a **managed DB service** that handles: PostgreSQL, MySQL, MariaDB, Oracle, Microsoft SQL Server, Aurora (AWS proprietary database). It's great for OLTP (On-Line Transaction Processing).

7.1.1 Aurora:

Cloud Optimized DB (5x better) and compatible with PostgreSQL and MySQL, but it's more expensive (20% more). It automatically grows in increments of 10GB (up to 64TB).

7.2 ElastiCache

Managed Redis or Memcached databases. **In-memory key-value databases** with high performance. Great for caching Real heavy queries.

7.3 DynamoDB

Fully Managed, serverless, key-value, high-available (replication 3 AZ) database with **single, digit millisecond latency**.

7.4 Redshift

Columnar database, based on PostgreSQL and has a SQL interface for performing queries. It's great for OLAP (On-Line Analytical Processing).

7.5 EMR

Elastic Map Reduce helps you create a **Hadoop cluster**. It supports Apache Spark, HBase, Presto, Flink ... It's great for data processing, machine learning, indexing websites, big data...

7.6 Athena

Fully **query engine** with **SQL capabilities**. It is used to query data in S3.

7.7 DMS

Database Migration Service (DMS), quickly and securely **migrates databases to AWS**.

7.8 Glue

Is a **fully serverless ETL (Extract Transform Load)** service.

8 Other Compute Services

8.1 EC2

(Previously discussed)

8.2 ECS

Elastic Container Service (ECS) enables the **launch of Docker containers** but **you must provision & maintain the infrastructure (EC2 instances)**. AWS takes care of starting and stopping containers and has integration with **ELB (Application)**.

8.3 Fargate

Fargate enables the **launch of Docker containers** and **you do not provision the infrastructure (serverless)** it is much simpler than ECS.

8.4 ECR

Elastic Container Registry (ECR) is a **private docker registry** on AWS.

8.5 AWS Lambda

Lambda is a **serverless** service which runs **virtual functions** that are limited by time (execution time), runs **on demand** (**event-driven**: gets invoked when needed) and can **scale automatically**. The pricing is based on: number of requests, RAM, execution time.

8.6 AWS Batch

Fully managed batch processing at any scale. A batch job will start and end. It will **dynamically launch EC2 instances (can be Spot Instances)** and the actual batch job will be defined as **Docker images** and **run on ECS**.

8.7 AWS LightSail

Standalone service which **provide fully managed virtual servers, storage, databases and networking**. It has low cost, **predictable pricing** and is an alternative instead of using common AWS services (EC2, RDS, ELB, etc). That's why it is great for people with **little cloud experience**. It has **high availability** but **NO AUTO-SCALING and limited AWS integration**.

9 Deployments and managing infrastructure at scale

9.1 Cloudformation

Infrastructure as code (json or yaml) that directly represents the resources you want to add, supports almost all AWS services.

9.2 CDK

Cloud Development Kit (CDK) is an **Infrastructure as Code** (js/ts, python, java, .net) that is generates cloudformation json/yaml, and deploys it.

9.3 Elastic BeanStalk

Platform as a Service (PaaS) [Similar to Heroku] managed service, it handles: Instance & OS configuration. It supports many platforms (go, java, java + tomcat, .net, nodejs, php, ruby, docker, etc).

9.4 CodeDeploy

Deploys and upgrades applications onto servers. It uses a codedeploy-agent.

9.5 CodeCommit

Source-control git-based repository.

9.6 CodeBuild

Compiles source code, run tests and product artifacts to be deployed.

9.7 CodePipeline

Orchestrate the different steps to publish code to production (CI/CD).
Code $\rightarrow$ *Build* $\rightarrow$ *Test* $\rightarrow$ *Provision* $\rightarrow$ *Deploy*

9.8 CodeArtifact

Storage for artifacts $software dependencies$. Works with common dependencies manager $npm, yarn, maven, twine, p$

9.9 CodeStar

Unified view for developers to do CI/CD and code in one place. It automatically creates and manages CodeCommit + CodeBuild + CodeDeploy + CodePipeline + EC2 + Beanstalk.

9.10 Cloud9

Cloud IDE in the web browser. It allows pair programming (real-time).

9.11 SSM

AWS System Manager (SSM) **patch, configure and run commands at scale** in EC2 & On-Premise. It also **gets operational insight** of the infrastructure. It uses ssm-agent.

9.11.1 SSM Session Manager

Allows to start a secure shell on your EC2 and on-premise servers.

9.12 OpWorks

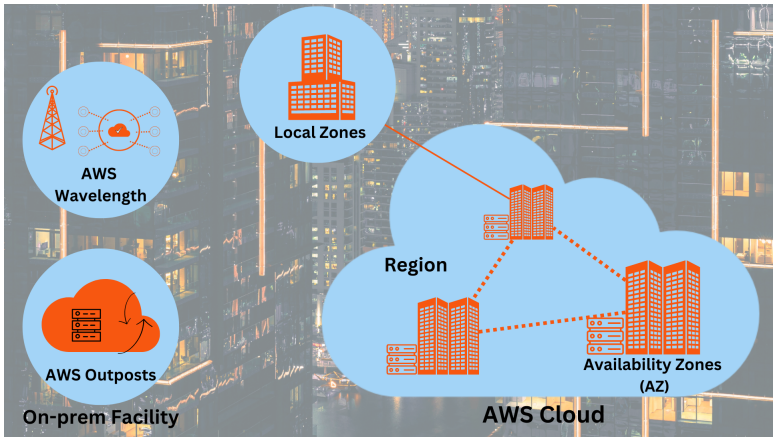
Managed Chef and Puppet. It allows you performs resource provisioning and ongoing future server configurations.

10 Global Infrastructure

10.1 Why make a global application?

- **Standard - General Purpose**: 99.99% available.
- **Deployed in multiple geographies:** Regions and Edge Locations.
- **Decreases latency:** Deploy applications closer to users.
- **Disaster recovery:** Fail-over to another region.
- **Attack protection:** Harder to attack.

10.2 Components



10.3 Global AWS applications

10.3.1 Route 53

Managed DNS.

10.3.2 Cloudfront

Content Delivery Network (CDN). Improves read performance because content is cached at an edge location. DDoS protection.

10.3.3 S3 Transfer Acceleration

Improve transfer speed by transferring to an edge location.

10.3.4 Global Accelerator

Improve global application availability and performance by using the AWS internal network through edge location.

10.3.5 Outpost

On-premise local private AWS infrastructure.

10.3.6 WaveLength

Infrastructure embedded within telcom providers datacenter at the edge of the 5G networks.

10.3.7 Local Zones

Places AWS resources closer to end users (reduce latency).