Pathways Technologies

Data • Technology • Innovation

# 2025 | AWS CLOUD PRACTITIONER COURSE

## Your Path to Success: Discover the Cloud's Full Potential.

Pathways Technologies
Data • Technology • Innovation

# ABOUT THIS COURSE

## COURSE DESCRIPTION

This course is designed for individuals seeking a general understanding of AWS cloud - services, architecture, security, pricing, and deployment strategies to kickstart their cloud careers or enhance their existing skills. This course also helps you prepare for the AWS Certified Cloud Practitioner exam

## AUDIENCE

Salespeople, Legal, Marketing, Business Analysts, project managers, IT-related professionals, students, or simply someone looking to enhance your cloud computing skills

## PRE-REQUISITE

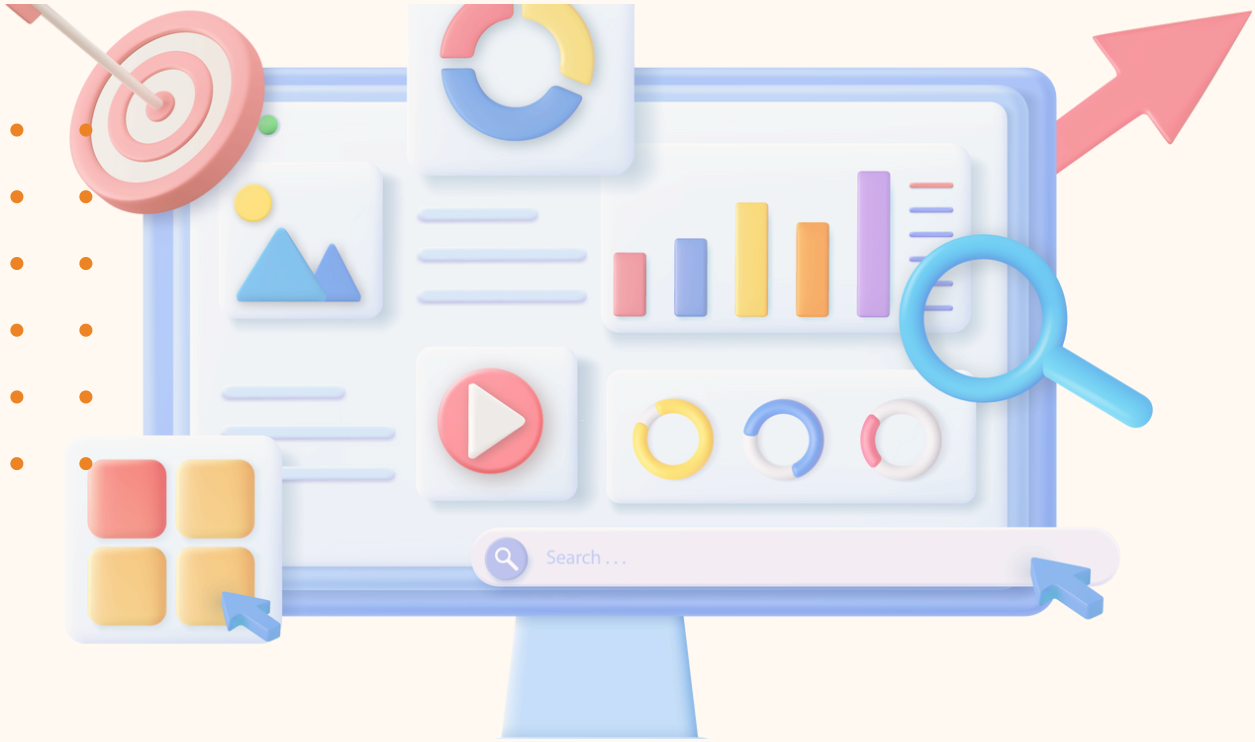
We recommend that attendees of this course have:

- General IT business knowledge
- General IT technical knowledge

aws PARTNER

Pathways Technologies
Data • Technology • Innovation

# COURSE OBJECTIVES




In this course you will learn to;

- Summarize the working definition of AWS
- Differentiate between on-premises, hybrid-cloud, and all-in cloud
- Describe the basic global infrastructure of the AWS Cloud
- Explain the six benefits of the AWS Cloud
- Describe and provide an example of the core AWS services, including compute, network, databases, and storage
- Identify an appropriate solution using AWS Cloud services with various use cases
- Describe the AWS Well-Architected Framework
- Explain the shared responsibility model
- Describe the core security services within the AWS Cloud
- Describe the basics of AWS Cloud migration
- Articulate the financial benefits of the AWS Cloud for an organization's cost management
- Define the core billing, account management, and pricing models
- Explain how to use pricing tools to make cost-effective choices for AWS services

aws PARTNER

Pathways Technologies
Data • Technology • Innovation

# COURSE OUTLINE

## Module 1: Introduction to Amazon Web Services (AWS)

- Introduction to AWS Cloud
- Overview of Services and Categories
- Differentiate between on-premise servers and cloud computing
- Identify the top benefits of cloud computing

## Module 2: Compute in the cloud

- Describe the benefits of Amazon Elastic Compute Cloud (Amazon EC2) at a basic level
- Identify the different Amazon EC2 instance types
- Differentiate between the various billing options for Amazon EC2
- Describe the benefits of Amazon EC2 Auto Scaling
- Summarize the benefits of Elastic Load Balancing
- Give an example of the uses for Elastic Load Balancing
- Summarize the differences between Amazon Simple Notification Service (Amazon SNS) and Amazon Simple Queue Services (Amazon SQS)
- Summarize additional AWS compute options

## Module 3: Global Infrastructure and Reliability

- Summarize the benefits of the AWS Global Infrastructure
- Describe the basic concept of Availability Zones
- Describe the benefits of Amazon CloudFront and Edge locations
- Compare different methods for provisioning AWS services

## Module 4: Networking

- Describe the basic concepts of networking
- Describe the difference between public and private networking resources
- Explain a virtual private gateway using a real-life scenario
- Explain a virtual private network (VPN) using a real life scenario
- Describe the benefit of AWS Direct Connect
- Describe the benefit of hybrid deployments
- Describe the layers of security used in an IT strategy
- Describe which services are used to interact with the AWS global network

aws PARTNER

Pathways Technologies
Data • Technology • Innovation

# COURSE OUTLINE

## Module 5: Storage and Databases

- Summarize the basic concept of storage and databases
- Describe the benefits of Amazon Elastic Block Store (Amazon EBS)
- Describe the benefits of Amazon Simple Storage Service (Amazon S3)
- Describe the benefits of Amazon Elastic File System (Amazon EFS)
- Summarize various storage solutions
- Describe the benefits of Amazon Relational Database Service (Amazon RDS)
- Describe the benefits of Amazon DynamoDB
- Summarize various database services

## Module 6: Security

- Explain the benefits of the shared responsibility model
- Describe multi-factor authentication (MFA)
- Differentiate between the AWS Identity and Access Management (IAM) security levels
- Describe security policies at a basic level
- Explain the benefits of AWS Organizations
- Summarize the benefits of compliance with AWS
- Explain primary AWS security services at a basic level

aws PARTNER

Pathways Technologies
Data • Technology • Innovation

# COURSE OUTLINE

## Module 7: Monitoring and Analytics

- Summarize approaches to monitoring your AWS environment
- Describe the benefits of Amazon CloudWatch
- Describe the benefits of AWS CloudTrail
- Describe the benefits of AWS Trusted Advisor

## Module 8: Pricing and support

- Understand AWS pricing and support models
- Describe the AWS Free Tier
- Describe key benefits of AWS Organizations and consolidated billing
- Explain the benefits of AWS Budgets
- Explain the benefits of AWS Cost Explorer
- Explain the primary benefits of the AWS Pricing Calculator
- Distinguish between the various AWS Support Plans

## Module 9: Migration and Innovation

- Understand migration and innovation in the AWS Cloud
- Summarize the AWS Cloud Adoption Framework (AWS CAF)
- Summarize six key factors of a cloud migration strategy
- Describe the benefits of various AWS data migration solutions, such as AWS Snowcone, AWS Snowball, and AWS Snowmobile
- Summarize the broad scope of innovative solutions that AWS offers
- Summarize the five pillars of the AWS Well-Architected Framework

aws PARTNER

Pathways Technologies
Data • Technology • Innovation

aws PARTNER

# CONTACT US

MOBILE:0791 183 444 | 0722540328
TEL:0771 616 839

www.pathwaystechnologies.com

info@pathwaystechnologies.com

Pathways Technologies Limited
236 Owashika, Lavington
Nairobi-Kenya

# AWS Introduction



Introduction

Pathways Technologies

# What is AWS?

AWS is a cloud based platform that enables you to build sophisticated, scalable applications.

It is applicable to diverse set of industries.

Use cases include:

- Enterprise IT, Backup and Storage, Big Data analytics
- Website hosting, mobile and Social apps
- Gaming

# AWS Global Infrastructure

The AWS global ecosystem consists of:

- AWS Regions
- AWS Availability Zones
- AWS Data Centers
- AWS Edge Locations

Global Infrastructure Regions & AZs

# AWS Regions

AWS has regions all around the world. Currently it has 26 launched Regions.

Names can take the format us-east-1, eu-west-3

A region is a cluster of data centers or a geographical location with Availability zones.

Most AWS services are region-scoped. If we use a service in one region and try to use it another region, it will look like the first time of using the service.

**How to choose an AWS Region**

1. **Compliance with data governance and legal requirements:** data never leaves a region without your explicit permission.
2. **Proximity to customers**: reduce latency.
3. **Available services within a region:** new services and new features aren't available in every region.
4. **Pricing:** pricing varies region to region and is transparent in the service pricing page.

# AWS Availability Zones

Each region has many availability zones (usually 3, min is 2,max is 6).

Examples of AZs in an AWS Region(Sydney - ap-southeast-2):

- ap-southeast-2a
- ap-southeast-2b
- ap-southeast-2c

Each Availability zone (AZ) is one or more discrete data centers with redundant power, networking and connectivity.

They are **separate from each other**, so that they are isolated from disasters.

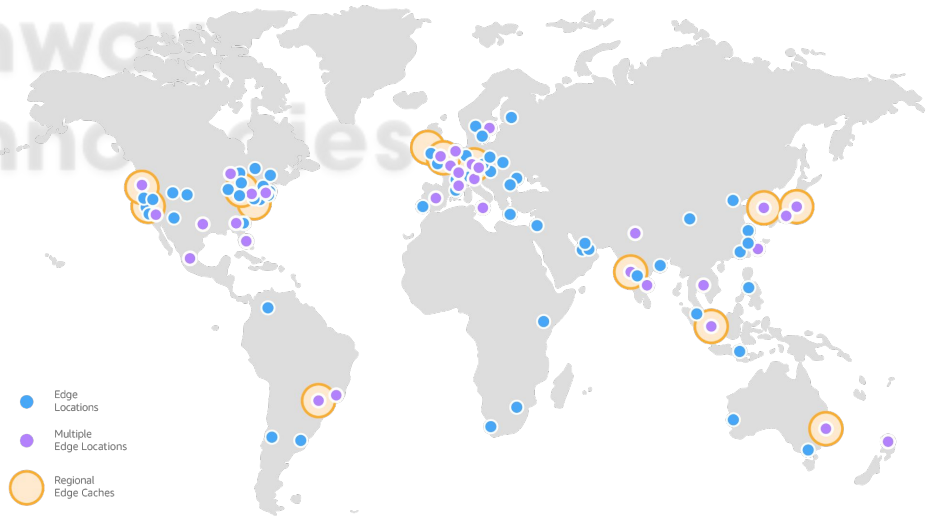
They are connected with high bandwidth, ultra-low latency networking.

# AWS Points of Presence (Edge locations)

Amazon has 410+ Points of presence (400+ Edge Locations and 13 Regional Cache) in 90+ cities across 48 countries.

Content is delivered to end users with lower latency.

Used by Amazon  Cloudfront to store cached copies of your content closer to your customers.



# Ways to access AWS services

# AWS Management Console

Is a web-based interface for accessing and managing AWS services.

AWS has Global services:

- Identity and Access Management (IAM).
- Route 53 (DNS service).
- CloudFront (Content Delivery Network)
- WAF (Web Application Firewall)

Most AWS services are Region-scoped:

- Amazon EC2 (Infrastructure as a Service).
- Elastic Beanstalk (Platform as a Service)
- Lambda (Function as a Service)

Region table: AWS Regional Services

# AWS Command Line Interface

Enables control of multiple AWS services directly from the command line within one tool.

Available for users on Windows, macOS and Linux.

Using this tool one can automate the actions that your services and applications perform through scripts. Example: use commands to launch an Amazon EC2 instance, connect to a specific Auto Scaling group and more.

Its open source and an alternative to AWS Management console.

# Software Development Kit

A set of libraries, language specific APIs, enables you to access and manage AWS services programmatically.

Make it easier for you to use AWS services through an API designed for your programming language or platform.

Enable you to use AWS services with your existing applications or create entirely new applications that will run on AWS.

Supports

- SDKs: Javascript, Python, PHP, .NET, Ruby, java, GO, Node.js, C++
- Mobile SDks (Android, iOS,..)
- IoT Device SDKs(Embedded C, ...)

Example of something built with SDK: AWS CLI is built on AWS SDK for Python.

# AWS CloudShell

An alternative to using CLI within the AWS management console.

Not available on all regions.

Is a terminal within the AWS environment that is free to use.

Credentials used are those of the logged in account.

# Identity and Access Management



IAM policies and roles

# Introduction: Users, Groups, Policies

IAM = Identity and Access Management

It is a **Global** service

**Root account**: created by default when you create an account and should not be used or shared (Best practice in security).

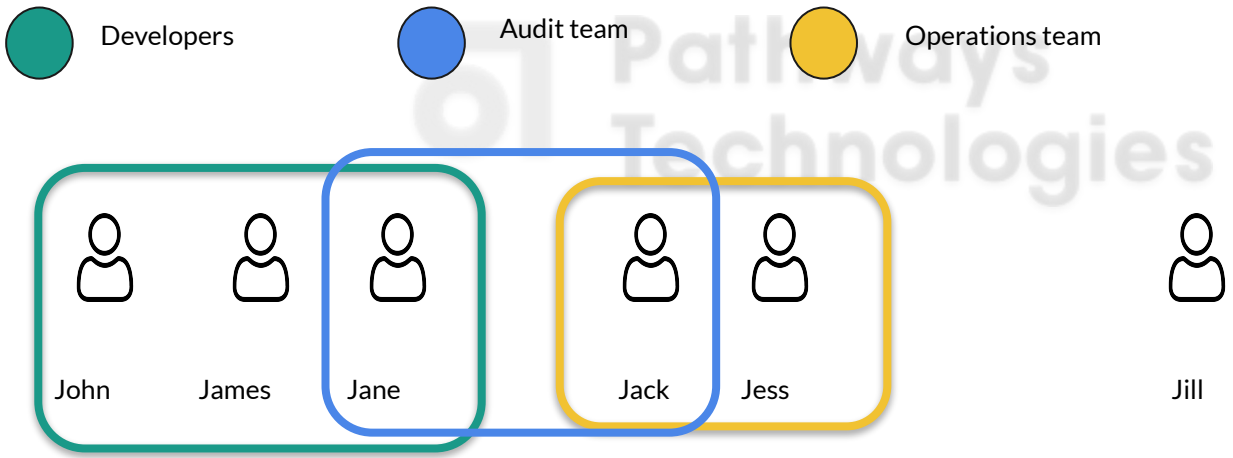
Instead create **Users**, these are people with your organization, and can be grouped.

**Groups** can only contain users, not other groups.

Users do not have to belong to a group, and user can belong to multiple groups

It is important to use the **least privilege principle**; dont give more permissions than a user requires.

# Visual representation of Users and Groups.



# IAM Policies: Permissions

Users and Groups are useful in assigning permissions to AWS services.

These are JSON documents called **Policies**.

When creating Policies it is important to apply the least privilege principle.

When a policy is attached to a Group the users in the group tend to inherit the said policy.

**Inline policy** is a policy directly attached to a User.

# IAM Policies Structure

Consists of:

- Version; policy language version
- Id: an identifier for the policy (optional).
- Statement: one or more individual statements (required).

Statement consists of :

- Sid: an identifier for the statement (optional)
- Effect: where the statement allows or denies access (Allow, Deny)
- Principal: account/user/role to which this policy is applied to
- Action: list of actions this policy allows or denies.
- Resource: list of resource that the IAM policy is applied to.
- Condition: conditions for when this policy is in effect (optional).

# IAM MFA and password

a) Password policy:

- Aid in higher security for your account

- Variables you can set include, minimum length, required specific character type, allow all IAM users to change their own password, password expiration and prevent password reuse

b) MFA - Multi Factor Authentication:

- MFA = password *you know* + security device *you own*.

- Protect your Root Accounts and IAM Users

Device options: Virtual MFA device, Universal 2nd Factor(U2F) Security key etc.

# AWS Access Keys, CLI and SDK

There are three option to access AWS:

a) AWS Management Console, protected by a password and MFA.
b) AWS Command Line Interface (CLI), protected by access keys.
c) AWS Software Developer Kit(SDK), mainly for code and protected by access keys.

Access keys are generated through the AWS Console.

Users manage their own access keys

**Access Keys are secret, just like a password. Don't share them**

Access Key ID ~= username

Secret Access Key ~= password

# IAM Roles

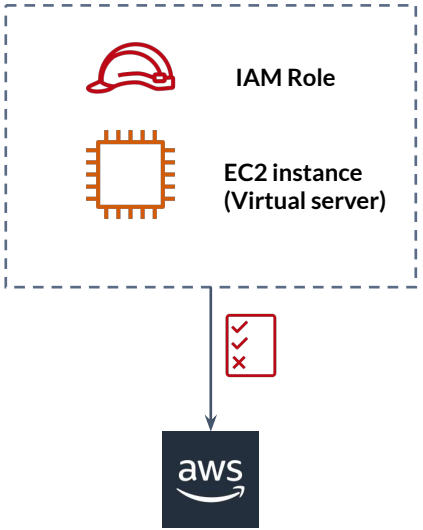
Some services will need to perform actions on your behalf

To assign permissions to AWS services we use IAM Roles.

It is a document with definition of who(app, AWS service etc), can use what(list of API call), under which conditions(list of services specific conditions) managed by IAM service.

Common roles include:

a) EC2 Instance Roles.
b) Lambda Function Roles.
c) Roles for CloudFormation.



# AWS Security Token Service (STS)

Used to grant limited and temporary access to AWS resources.

Token is valid upto one hour.

Some actions supported:

- AssumeRole:
  - Used for own account or cross account access.
- AssumeRoleWithSAML:
  - Return credentials for users logged with SAML
- AssumeRoleWithWebIdentity:
  - Return credentials for users logged within an IdP(Identity Provider).
  - AWS recommends against using this, unless you have no alternative.

# When to use IAM Role or STS

Use IAM Role if your app is running on AWS if it uses SDK for whatever language. It assumes the role automatically.

Use STS if you want to perform actions in completely different account or the app is running outside of AWS.

# Identity Federation

Is a system of trust between the IdP (Identity Provider) and the SP (Service Provider).

Lets users out of AWS to assume temp roles for accessing AWS resources.

The users assume the identity provided access role.

It is a common approach to building access control systems which manage users centrally within a central IdP and govern their access to multiple app and services acting as SPs.

# IAM Security Tools

- **IAM Credentials Report (account-level)**
  - A report that highlights all your account's users and status of their various credentials.

- **IAM Access Advisor (user-level)**
  - Access advisor shows the service permissions granted to users and when services were last accessed.
  - You can use this information to revise your policies.

# AWS Organizations

Is an account management service that enables you to consolidate multiple AWS accounts into an *organization* that can be managed centrally.

You can create new accounts and invite existing accounts to join the organization.

Features:

- Centralized management.
- Consolidated billing for all member accounts
- Hierarchical grouping of your accounts to meet budgetary, security or compliance needs.
- Policies to centralize control over the AWS services and API actions that each account can access.
- Global access
- Free

# IAM Guidelines& Best Practices

1. Use strong sign-in mechanisms:
    a. Create IAM policy to enforce MFA sign in.
    b. Configure strong password policy.
    c. Rotate credentials regularly.
2. Temporary Credentials:
    a. Implement least privileges policies.
    b. Consider permission boundaries. A permission boundary is an advanced feature for using a managed policy that sets the maximum permissions that an identity can grant to am IAM entity.
3. For workforce and machine identities that require secrets such as passwords to third party apps, store them with automatic rotations, AWS Secrets Manager.
4. Assign users to groups and assign permissions to groups, for easy permission management.



Never share IAM users & Access Keys

# IAM Practical

# Amazon Elastic Compute Cloud (EC2)

AWS Training: Amazon Elastic Compute Cloud

Pathways Technologies

# Amazon Elastic Compute Cloud (EC2) - Introduction



It is one of the most popular AWS offering.

It mainly consists in the capability of:

- Renting virtual machines(EC2)
- Storing data on virtual drives(EBS)
- Distributing load across machines (ELB)
- Scaling the services using an auto-scaling group (ASG)

Knowing EC2 is fundamental in understanding how the Cloud works.

# EC2 sizing and configuration options

Operating System (OS): Linux, Windows or Mac OS

How much compute power and cores (CPU)

How much random-access memory (RAM)

How much storage space:

- Network-attached (EBS and EFS)
- hardware

Network card: speed of the card, Public IP address

Firewall rules: security group

Bootstrap script(configure at first launch): EC2 User Data

# EC2 User Data

It is possible to bootstrap our instances using an EC2 USer Data script.

Bootstrapping means launching commands when a machine starts, the script is only run once at the first start.

EC2 user data is used to automate boot tasks such as:

- Installing updates
- Installing software
- Downloading common files from the internet
- Anything you can think of

The EC2 User Data Script runs with root user of the instance.

# EC2 instance types: Overview

You can use different types of EC2 instances that are optimised for different use cases. (Amazon EC2 Instance Types - Amazon Web Services)

Aws follow the following naming convention;

m5.2xlarge

m: instance class

5: generation (AWS improves over time)

2xlarge: size within the instance class

**General purpose**

**Compute optimized**

**Memory Optimized**

**Accelerated Computing**

**Storage optimized**

**Instance Features**

**Measuring Instance Performance**

# EC2 instance types: Cont.

General purpose :

- Great for diversity of workloads such as web server code repositories
- It is a balance between:
  - Compute
  - Memory
  - Networking

Compute Optimized:

- Great for compute-intensive tasks that require high performance processors
- Examples:
  - Batch processing workloads.
  - Media transcoding.
  - High performance web servers.
  - High performance computing.
  - Scientific modeling and machine learning.
  - Dedicated gaming servers.

# EC2 instance types: Cont.

Memory Optimized

- Fast performance for workloads that process large data sets in memory.
- Use cases:
  - High performance, relational/non-relational databases.
  - Distributed web scale cache stores.
  - In-memory databases optimized for BI (business intelligence)
  - Applications performing real-time processing of big unstructured data.

Storage optimized:

- Greate for storage-intensive tasks that require high, sequential read and write access to large data sets on local storage.
- Use cases:
  - High frequency online transaction processing (OLTP) systems.
  - Relational and NoSQL databases.
  - Cache for in-memory databases (for example, Redis)
  - Data warehousing applications.
  - Distributed file systems.

# Security Groups

They are a fundamental of network security in AWS. They act as a “firewall” on EC2 instances.

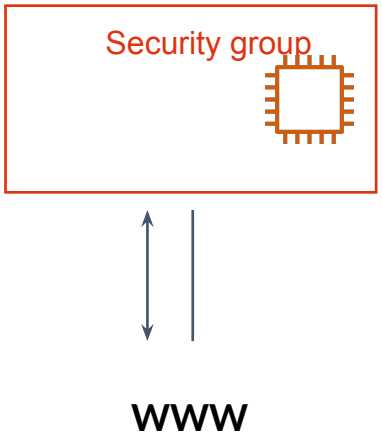
They control how traffic is allowed into or out of our EC2 instances.

Security groups only contain allow rules.

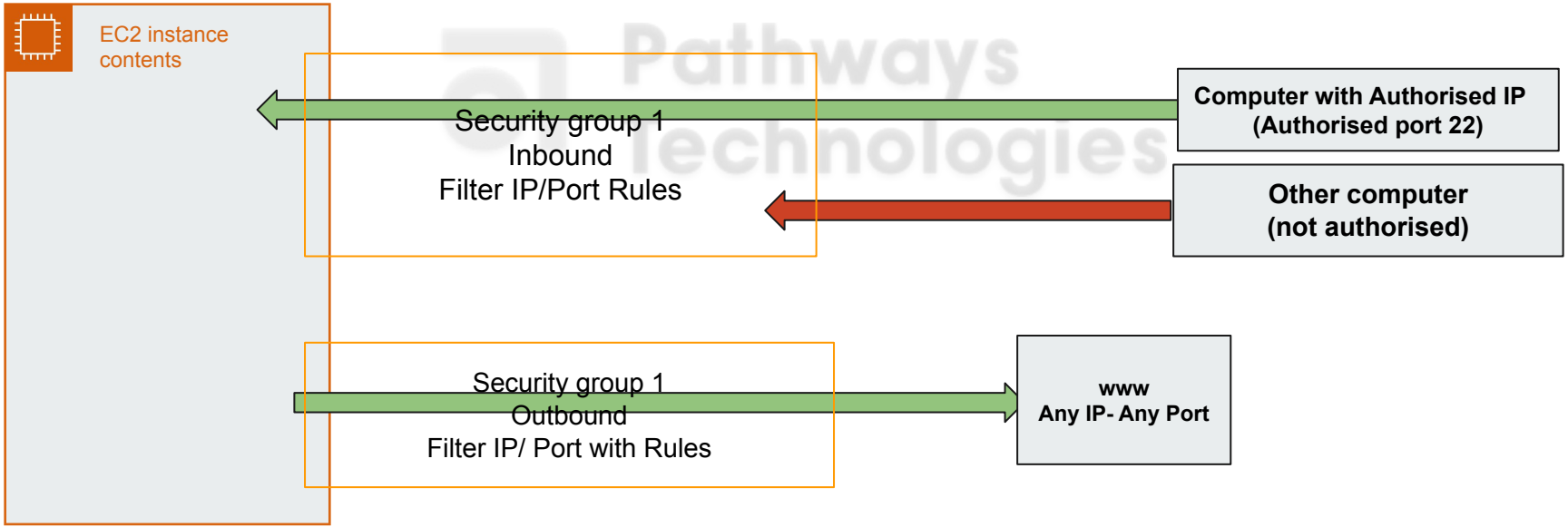
Security groups rules can reference by IP or by security group.

They regulate:

- Access to Ports.
- Authorised IP ranges - IPv4 and IPv6.
- Control of inbound network (other to the instance).
- Control of outbound network (from the instance to other).



# Security Group diagram



# Security Groups: Good to know

- Can be attached to multiple instances
- Locked down to a region/VPC combination
- Does live ‘outside ‘ of the EC2 -  if traffic is blocked the EC2 instance won’t see it.
- If your application is not accessible (timeout), then its a security group issue.
- If your application gives a “connection refused” error, then it’s an application error or it’s not launched.
- By default all inbound traffic is blocked
- All outbound traffic is authorised by default.

# Classic Ports to know

22 = SSH (Secure Shell) - log into a linux instance

21 = FTP (File Transfer Protocol) - upload files into a file share

22 - SFTP (Secure File Transfer Protocol) - upload files using SSH

80 = HTTP - access unsecure websites

443 = HTTPS - access secure websites

3389 = RPD  (Remote Desktop Protocol) - log into a windows instance.

# SSH

| | SSH | Putty |
|---|---|---|
| Linux | yes | |
| Mac | yes | |
| Windows < 10 | | yes |
| Windows >= 10 | yes | yes |

# EC2 Purchasing Options

On -Demand Instances - short workload, predictable pricing, billing by second. Has the highest cost but no upfront payment. Recommended for short-term and un-interrupted workloads.

Reserved (1 or  3 years).  Up to 72% discount compared to On-Demand.

- Reserved Instances - long workloads.
- Convertible Reserved Instances - long workloads with flexible instances

Savings Plan (1 or 3 years) - commitment to an amount of usage, longworkload.

Spot Instances - short workloads, cheap can lose instances (less reliable). Recommended for apps that have flexible start and end times. Can get a discount of up to 90% compared to On-demand.

Dedicated Hosts - book an entire physical server, control instance placement.  Recommended for server-bound s/w licenses.(On-demand or Reserved)

Dedicated Instances - no other customers will share your hardware.

Capacity Reservations - reserve capacity in a specific AZ for any duration

# Private vs Public vs Elastic IP

Networking has 2 sorts of IPs. IPv4 and IPv6:

- IPv4: 1.160.10.240
- IPv6: 1900:4545:3:200:f8ff:fe21:67cf

Public IP:

- Public IP means the machine can be identified on the internet.
- Must be unique across the web.
- Can be geolocated easily.

Private IP:

- Private IP means the machine can only be identified on a private network only.
- The IP must be unique across the private networks, can have the same IPs.
- Machines connect to WWW using a NAT + internet gateway (proxy).
- Only a specified range of IPs can be used as private IP

# Elastic IP

When you stop then start an EC2 instance, it can change its public IP. If you need to have a fixed public for your instance, you need an Elastic IP.

An Elastic IP is a public IP you own as long as you don't delete it.

You attach it to one instance at a time.

With an Elastic IP address, you can mask the failure of instance or software by rapidly remapping the address to another instance in your account.

AWS only allows 5 Elastic IPs in your account.(can be increased upon request)

Overall, try to avoid using Elastic IP:

- Instead opt for a random public IP and register a DNS name to it.

# Placement groups

It facilitates control over placement strategy of EC2 instances.

a) Cluster:

Clusters instances into a low-latency network performance group in a single AZ.

Drawback, If hardware fails all instances fails at the same time.

b) Spread:

All EC2 instance are on different hardware,can span across AZ and reduce risk of simultaneous failure.

Drawback, limited to 7 instances per AZ per placement group

Use case, for applications that need to maximize high availability and critical apps where each instance must be isolated from failure from each other.

c) Partition:

Spreads instances across many different partitions within an AZ, which require different set of racks.

Use cases, Big Data apps.

# Network Interface (Elastic Network interface)

A virtual network card.

Can be attached to an instance, detached and attached to another instance.

When you move a network interface from one instance to another, network traffic is redirected to the new instance.

Each Instance has a default network interface that can’t be detached.

You can create and attach additional network interfaces.

# EC2 Actions

Stop: EBS volume is present until next start.

Terminate: EBS volume is cleared.

Start: OS is boot.

Hibernate: The in-memory RAM state is preserved.

- The instance boot is much faster
- Under the hood the RAm state is written to a file in the root EBS volume.
- Use case, long running processes, saving the RAM state and services that take time to initialize.

# Practical: EC2

# Provision an EC2 instance

User data code:

```
#!/bin/bash

sudo yum update -y

sudo amazon-linux-extras install nginx1 -y

sudo systemctl enable nginx

sudo systemctl start nginx
```

# Extra - EC2 instance metadata

It allows EC2 instances to 'learn about themselves' without using an IAM role for that purpose.

http://169.254.169.254/latest/meta-data called within the EC2 instance.

You can retrieve the IAM role name from the metadata but you can not retrieve the IAM Policy.

Metadata = information about  the EC2 instance.

Userdata = launch script of the Ec2 instance.

# SSH in EC2 instance

Change permission level of the security key(.pem file)

Chmod 400 ‘myfile’.pem

Ssh into instance

Linux: ssh -i ‘myfile’.pem ec2-user@public-ip-of-instance

# EC2 Instance Storage



EBS, EFS and Instance Store

Pathways Technologies

# Amazon Elastic Block Storage (EBS)

Provide block level storage volumes, for use with EC2.

Consider it a network USB Drive.

Uses the network to communicate the instance which means their might be a bit of latency.

Can be detached and attached to instances.

It is locked to an AZ. To move a volume from one AZ to another you first need to snapshot it.

They have a provisiones capacity and recommended for data that must be quickly accessible and requires long term persistence.

**Delete on termination attribute:**

Controls the EBS behaviour when an instance is terminated. By default, the root EBS root EBS is terminated.

**EBS Snapshots:**

It is a backup of your EBS volume at a point in time.

It is not necessary to detach volume when taking a snapshot, but it is **recommended**.

# Amazon machine Image (AMI)

Is a template that contains a software configuration( example OS, application server and applications). From an AMI, you launch an instance, which is a copy of the AMI running as a virtual server in the cloud.

AWS publishes many AMIs that contain common software configurations for public use.

You can also create your own AMI/s; doing so enables you to quickly and easily start new instances that have everything you need.

**AMI Process, from an EC2 Instance:**

1. Start an EC2 instance and customize.
2. Stop (For data integrity).
3. Build an AMI, an EBS snapshot will also be created.
4. Launch an instance from other AMIs.

# EBS volume types

They come in 6 types:

- gp2/gp3 (SSD): General purpose SSD volume that balances price and performance for a wide variety of workloads. Ideal for boot volumes, medium-size single instance databases and development and testing environments.
- io1/io2 (SSD): High performance SSD(also known as Provisioned IOPS) for mission critical low latency or high throughput workloads. They provide constituent IOPS rate specified when creating the volume. Ideal for critical business apps with sustained IOPS performance, DB workloads or apps that need more than 16, 000 IOPS.
- st1 (HDD): (Throughput Optimized HDD) ) low cost HDD volume designed for frequently accessed, throughput intensive workloads. Ideal for large sequential workloads such as Amazon EMR, ETL data warehouses and log processing.
- sc1 (HDD): (Cold HDD) )lowest cost HDD volume designed for less frequently accessed workloads. These volume provide inexpensive block storage.

**Note:** only gp2, gp3, io1,io2 can be used as boot volumes.

# EBS Multi-Attach - io1/io2 Family

Attach the same EBS volume to multiple EC2 instances in the same AZ.

Each instance has full read and write permissions to the volume.

Volumes can not be created as boot volumes.

Use cases:

- To achieve higher application availability in clustered linux apps, example Teradata.
- Apps must manage concurrent write operations.

# Amazon EC2 Instance Store

Provides temporary block-level storage for your instance. Located on disks that are physically located to the host computer.

There is risk of data loss if hardware fails, storage is ephemeral.

High performance disk and Better I/O performance compared to EBS.

An EC2 instance loses their storage if they stop, hibernates or terminates

Good for buffer, cache, scratch data and temporary content.

If you create an AMI from the instance, the data on its instance store volumes isn't preserved.

If you change the instance type, an instance store will not be attached to the ner instance type.

# Amazon Elastic File System (EFS)

Provides a simple, serverless, set and forget elastic file system for use with AWS services and on-prem resources.

Managed NFS (Network File System) that can be mounted on many EC2 instances.

EFS works with EC2 instances in multi-AZ.

It is highly available, scalable and expensive(3 times gp2 cost).

Uses NFSv4.1 Protocol that is compatible with current  applications and tools.

Compatible with Linux based instances.

Encryption at rest can be enabled using AWS Key Management Service(KMS).

Use cases: content management, web serving, data sharing and Wordpress.

# EFS - performance

**EFS Scale:**

1000s of concurrent NFS clients and more than 10 GiBps of throughput.

Has the potential to grow to petabyte scale network file system automatically.

The following configurations impact the performance of an EFS; Performance mode, Throughput mode and Storage class.

**Performance mode:** Set at EFS creation time.

1. General Purpose mode (default): supports up to 35,000 IOPS and has lowest per-operation latency. Ideal for web servers,content management system.
2. Max I/O mode: supports 500,000+ IOPS and has higher per-operation latencies. Ideal for big data, media processing.

**Throughput mode:**

1. Bursting Throughput mode(default): Burst from 50 MiBps per TiB of storage of upto 100 MiBps.
2. Provisioned Throughput mode: Recommended for apps that have a relatively constant throughput. Set the throughput regardless of storage size.

# EFS - Storage Classes

There are four storage classes:

EFS Standard and Standard IA are regional storage classes that are designed to provide continuous availability to data, even when one or more AZ in an AWS region are unavailable.

1. Amazon EFS Standard class: Used for frequently accessed files. It's the storage class to which customer data is initially written for Standard Storage classes.
2. Amazon EFS Standard- IA class: reduces cost for files not frequently accessed files. Recommended for files that need to be readily available and save cost unless frequently accessed files. Example, keeping files accessible to satisfy audit requirements.

EFS One Zone storage classes are designed to provide continuous availability to data within a single AZ. Data might be lost in the event of a disaster in one AZ.

3. Amazon EFS One Zone-Standard class:used for frequently access files. It's the storage class to which customer data is initially written for One Zone Storage class.
4. Amazon EFS One Zone- IA class: reduces storage costs for files that are not accessed every day.

# Using EFS storage classes

To use EFS Standard and Standard - IA classes, create a file system -> Storage class select Standard.

To use Standard-IA -> enable EFS lifecycle management. Moves files from Standard storage to IA storage. Life cycle management is enabled by default with a setting of 30 days since last access.

To use EFS One Zone and One Zone - IA classes, create a file system -> choose Availability and Durability then choose One Zone.

To use Standard-IA -> enable EFS lifecycle management. Moves files from Standard storage to IA storage. Life cycle management is enabled by default with a setting of 30 days since last access.

# Summary

- EBS is a high-performance per-instance system designed to act as storage for a single EC2 instance, most of the time.
- EFS is a highly scalable file storage system designed to provide flexible storage for multiple EC2 instances.
- Instance Store is temporary block-level storage for your instance can be used as local cache and data is lost on instance termination.

# Practical: EFS

# Amazon Simple Storage Service



S3 Bucket

Pathways Technologies

# Introduction

Allows storage of objects(files) into buckets (“directories”).

Buckets must have a globally unique name.

Buckets are defined at the region level.

Naming convention:

- No uppercase
- No underscore
- 3-63 characters long
- Not an IP
- Must start with lowercase letter or number.

Objects(files) have a key.

The key is the Full path, and is composed of the prefix and the object name. Examples:

- s3://my-buckeT/my_file.txt
- s3://my-bucket/myfolder1/another_folder/my_file.txt

There are no concepts of “directories within buckets, just keys with very long names that contain slashes (“/”). Although the UI might trick you to think so.

# S3 Storage Classes

You can choose from a range of storage classes to select a fit for your business and cost needs.

There are two factors to consider when selecting an S3 storage class:

- How often you plan to retrieve your data
- How available you need your data to be

a) Amazon S3 Standard

Designed for frequently accessed data

Stores data in a minimum of three AZs

Good choice for a wide range of use cases, such as websites, content distribution and data analytics.

Has a higher cost than other storage classes intended for infrequently accessed data and archival storage.

b) Amazon S3 Standard-Infrequent access (S3 Standard-IA)

Ideal for infrequently accessed data but requires high availability when needed,

Similar to Amazon S3 Standard but has a lower storage price and higher retrieval price.

c) Amazon S3 One Zone-Infrequent Access (S3 One Zone-IA)

Stores data in single AZ

Has a lower storage price than Amazon S3 Standard-IA

Should be considered if you want to save cost on storage and you can easily reproduce your data in the event of an AZ failure

d) Amazon S3 Intelligent Tiering

Ideal for data with unknown or changing access patterns

Requires a small monthly monitoring and automation fee per object.

Amazon S3 monitors objects' access patterns, if you haven't accessed an object for 30 days S3 automatically moves it to the infrequent access tier, Amazon S3 Standard-AI

If you access an object in the infrequent access tier, Amazon S3 moves it to the frequent access tier, Amazon S3 standard

e) Amazon S3 Glacier Instant Retrieval

Works well for archived data that requires immediate access.

Can retrieve objects within a few milliseconds with the same performance as Amazon S3 Standard.

f) Amazon S3 Glacier Flexible Retrieval

Low-cost storage designed for data archiving

Able to retrieve objects within a few minutes to hours

Use case, used to store archived customer records or older photos and video files.

g) Amazon S3 Glacier Deep Archive

Lowest-cost object storage class for long term archiving of data that might be accessed once or twice a year.

Able to retrieve objects within 12 to 48 hours

# S3 versioning

You can version your files in Amazon S3 and is enabled at the bucket level.

Some key overwrite will increment the ‘version”:1,2…

It’s best practice to version your buckets:

- Protect against unintended deletes(ability to restore a version).
- Easy roll to previous version.

Note:

- Any file that is not versioned prior to enabling versioning will have version ‘null’.
- Suspending versioning does not delete the previous versions.

# S3 Encryption of objects

There are 4 methods of encrypting objects in S3:

- SSE-S3: encrypts S3 objects using keys handled & managed by AWS.(Server Side Encryption)
- SSE-KMS: leveraging AWS Key Management Service to manage encryption keys.(Server Side Encryption)
- SSE-C:  when you want to manage your own encryption keys. Amazon S3 doesn't store the encryption key.(Server Side Encryption)
- Client Side Encryption: customer manages the keys and encryption cycle.

Amazon S3 exposes both a HTTP endpoint and HTTPS endpoint. You are free to use the endpoint you wish, but  HTTPS is recommended.

Note HTTPS is mandatory for SSE-C.

# S3 Security

User based

- IAM policies - which API calls should be allowed for specific user from IAM console.

Resource based

- Bucket Policies - bucket wide rules from S3 console (allows cross account).
- Object Access Control List (ACL) - finer grain
- Bucket Access Control List (ACL) - less common

An IAM principal can access an S3 object if:

- The user IAM permissions allow it OR the resource policy ALLOWS it.
- AND there is NO explicit Deny

# S3 Bucket Policies

JSON based policies.

- Resources: buckets and objects
- Attached at the bucket level.
- Actions: Set of API to Allow or Deny
- Effect: Allow/ Deny
- Principal: The account or user to apply the policy to.

Use S3 bucket policy for:

- Grant public access to the bucket.
- Force objects to be encrypted at upload.
- Grant access to another account (Cross Account)

Networking:

- Supports VPC Endpoints (for instances in VPC without www internet)

Logging and Audit:

- S3 Access Logs can be stored in another S3 bucket
- API calls can be logged in AWS CloudTrail

User Security

- MFA Delete: MFA can be required in versioned buckets to delete objects.
- Pre-signed URLs: URLs that are valid only for a limited time (ex: premium video service for logged in users)

# S3 Websites

S3 can host static websites and have then accessible on the www

The website URL will be:

- <bucket–name>.s3-website.<AWS-region>.amozonaws.com

If you get a 403 (Forbidden) error, make sure the bucket policy allows public reads!

# S3 CORS (Cross Origin Resource Sharing)

Origin : Is a scheme(protocol), host( domain) and port.

Web Browser base mechanism to allow requests to other origins while visiting the main origin.

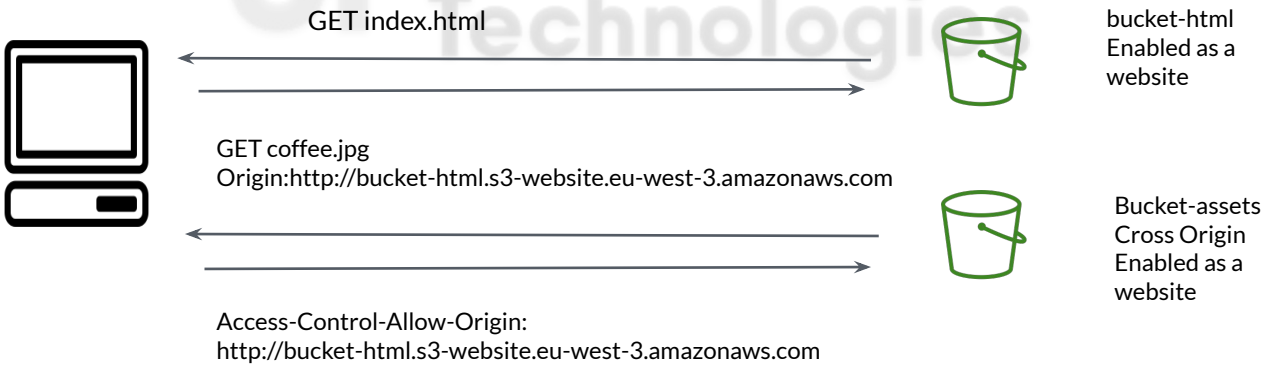
Example: http://www.example.com and http://other.example.com

The request won't be fulfilled unless the other origin allows for the request, using CORS Headers (ex: Access-Control-Allow-Origin)

If a client does a cross-origin request on our S3 bucket, we need to enable the correct CORS headers. Can be allowed for a specific region or for *(all origins).

# S3 CORS

Diagram explaining how S3 cors works:



# S3 ACLs

AWS recommends using bucket policies or IAM policies for access control.

It’s a legacy access control mechanism that predates IAM.

Attached to every bucket and object, defines which AWS accounts and groups are granted access and the typeof access.

The authorization decision depends on the union of all IAM policies, S3 bucket policies and S3 ACLs that apply. An explicit DENY always trumps an Allow.

# Practical: S3 Bucket

# High availability and Scalability

Load balancer, auto scaling group

# Elastic Load Balancer

It managed service that is used to forward traffic to multiple instances, containers and IP addresses.

It is integrated with other AWS offerings/services.

Why?

- Spread load across multiple downstream instances.
- Expose a single point of access (DNS) to your app.
- Seamlessly handle failures of downstream instances.
- Do regular health checks to your instances.
- Provide SSL termination HTTPS for your apps.
- Enforce stickiness with cookies.
- High availability across zones.
- Separate public traffic from private.

# ELB types

There are 4 kinds:

1. Classic Load balancer (old generation).
2. Application Load Balancer (HTTP, HTTPS, Websocket).
3. Network Load Balancer (TCP, TLS {secure TCP}, UDP).
4. Gateway Load Balancer (operates at layer 3, network, IP protocol).

# Application Load Balancer (ALB)

Designed to handle layer 7 (application layer).

Load balancing to multiple HTTP applications across machines (target groups).

Load balancing to multiple applications on the same machine (containers).

Support redirects (from HTTP to HTTPS).

Support routing to different routing tables:

- Routing based on path on URI
- Routing based on hostname in URL
- Routing based on Query String Headers.

Great for microservices and container-based applications eg ECS and Docker. Has a port mapping feature to redirect to dynamic port in ECS, in comparison  we'd need multiple CLB per app.

# Network Load Balancer

Allow to forward TCP and UDP traffic to instances:

Handle millions of request per second

Less latency ~ 100 ms (ALB 400 ms).

Has one static IP per AZ and supports assigning Elastic IP, helpful for whitelisting specific IP.

Are used for extreme performance TCP or UDP traffic

Not included in the AWS free tier.

# Gateway Load Balancer

Helps you easily deploy, scale and manage your third-party virtual appliances, such as firewalls, intrusion detection and prevention systems.

It combines a transparent network gateway, a single entry and exit for all traffic, and distributes traffic while scaling your appliance.

Operates in the network layer(3).

# Sticky Sessions - Session Affinity

The same client is always redirected to the same instance behind a load balancer. Compatible with CLB and ALB.

The cookie used for stickiness has an expiration date you control.

Use case: make sure the user doesn't lose his session data.

**Note:** Enabling stickiness may bring imbalance to the load over the backend EC2 instances.

Cookie names:

- Application-based cookies:

  It is a custom cookie generated by the target.

  Can include any custom attributes required by the custom attributes required by the application

  Cookie name must be specified for each target group.

- Application cookie:

  Generated by the load balancer.

  Cookie name is AWSALB, AWSALBAPP or AWSALBTG.

# Cross zone balancing

Each load balancer instance distributes evenly across all registered instances in all AZ.

ALB - Always on and no charges for inter AZ data.

NLB - Disabled by default and you pay for inter AZ data if enabled.

CLB - Disabled by default and no charges for inter AZ data if enabled.

# Auto Scaling Group (ASG)

The goal is to:

- Scale out, add EC2 to match an increase in load.
- Scale in, remove EC2 instances to  match a decrease load.
- Ensure we have a minimum and maximum number of machines running.
- Automatically register new instances to a load balancer.

Attributes:

Launch configuration(AMI + instance type, EC2 user data, EBS volume, Security groups, SSH key pair).

Min size / max size / initial capacity.

Network + subnet information.

Load balancer information.

Scaling policies.

# ASG - Alarms and Special Rules

It is possible to scale an ASG based on cloudwatch alarms.

An Alarm monitors a metric (eg Average CPU).

Metrics are computed for the overall ASG instances. Based on the alarm we can create scale out policies and scale in policies.

It is possible to define better auto scaling rules that are directly managed by EC2. Example target average CPU usage, number of requests on the ELB per instance and average network in/out.

# ASG Dynamic Scaling Policies

1. Target tracking scaling: Eg. i want the average ASG CPU to stay around 40%.
2. Simple step scaling: eg. when a cloudwatch alarm is triggered(CPU > 70 %), then add two units.
3. Scheduled Actions: Anticipate a scaling based of known events.
4. Predictive scaling: continuously forecast load and scheduling scaling ahead.

Good metrics to scale on include:

- CPU utilization.
- RequestCountPerTarget
- Average network in/out
- Any custom metric.

Note: After scaling activity happen, the cool down period starts. During this period, the ASG willl not launch or terminate additional instances. It is recommended to use a custom AMI to reduce cooldown period.

# Practical ELB and ASG

# AWS Databases



RDS, Aurora…

Pathways Technologies

# AWS Relational Database Service(RDS)

Its is a managed DB service for DB and uses SQL as a query language.

It allows you to create databases in the cloud that are managed by AWS.

- Postgres
- Mysql
- Aurora
- Mysql
- Oracle
- mariaDB

# Why RDS vs EC2?

Automated provisioning, OS patching

Continuous backups and restore to specific timestamp (point in time restore).

Monitoring dashboards.

Read replicas for improved read performances

Multi AZ setup for disaster recovery.

Maintenance windows for upgrades

Scaling capability (vertical and horizontal).

Storage backed by EBS (gp or io).

You can’t SSH into your instance.

# RDS Backups

Backups are automatically enabled in RDS. There are two types:

A) Automated backups:

- Daily full backups of the db.
- Transaction logs are backed up up by RDS every  minutes
- Ability to restore at any point in time.
- 7 days default retention period.

B) DB snapshots:

- Manually triggered by the user
- Retention of backups for as long as you want

# RDS - Cont

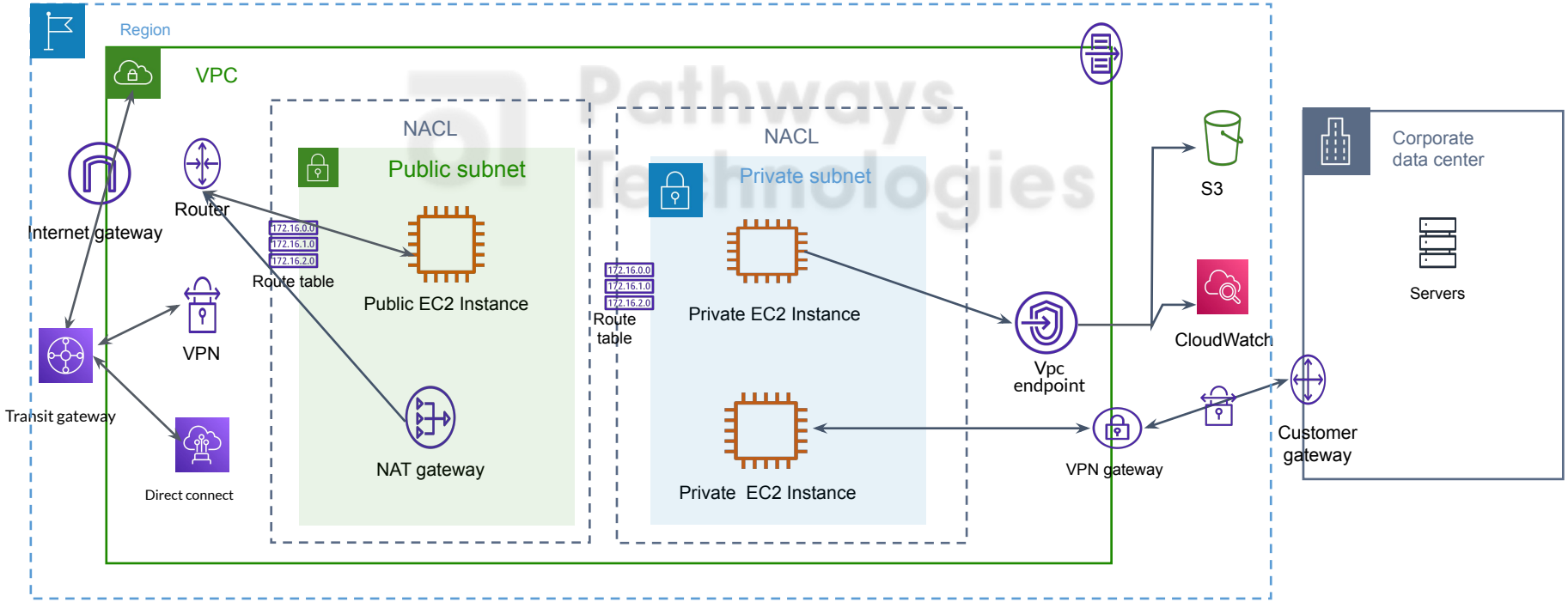
# Networking in AWS



Amazon Virtual Private Cloud (VPC)

Pathways Technologies

## VPC Components



# CIDR

Classes InterDomain Routing - a method for allocating IP addresses

Used  in Security Groups rules and AWS networking in general.

They help to define an IP address range:

- WW.XX.YY.ZZ/32 => one IP
- WW.XX.YY.ZZ/0=> all IPs
- But we can define: 192.168.0.0/26=> 192.168.0.0 - 192.168.0.63

Note:

- /32 - no octet can change
- /24 - last octet can change
- /16 - last 2 octets can change
- /8- last 3 octets can change
- /0 - all octets can change

# VPC in AWS

VPC = Virtual Private Cloud

A networking service that you can use to establish boundaries around your AWS resources.

You can have multiple VPCs in an AWS region (max. 5 per region - soft limit)

Max. CIDR per VPC is 5, for each CIDR:

- Min size /28 (16 IP addresses)
- Max. size is /16 (65536)

Because VPC is private, only the private IPv4 ranges are allowed:

- 10.0.0.0 - 10.255.255.255 (10.0.0.0/8)
- 172.16.0.0 - 172.31.255.255 (172.16.0.0/12)
- 192.168.0.0 - 192.168.255.255 (192.168.0.0/16)

**Your VPC CIDR should not overlap with other networks (e.g., corporate)**

## Default VPC Walkthrough

All new AWS accounts have a default VPC.

New EC2 instances are launched into the default VPC if no subnet is specified.

Default VPC has internet connectivity and all EC2 instances inside it have public IPv4 DNS names.

We also get a public and a private IPv4 DNS names.

# Subnets

Is a section of a VPC in which you can group resources based on security or operational needs. They can be Public or Private. Is tied to an AZ and we define a CIDR block for it.

Private subnets contain resources that should be accessible only through your private network, such as a database that contains customers' personal information.

Public subnets contain resources that need access to be accessible by the public internet, such as an online store's website.

AWS does reserve  IP addresses(first 4 and last 1) in each subnet

These 5 IP addresses are not available for use and can't be assigned to an EC2 instance

Example: If CIDR block 10.0.0.0/24, the reserved IP addresses are:

- 10.0.0.0 - Network Address
- 10.0.0.1 - reserved by AWS for the VPC router
- 10.0.0.2 -  reserved by AWS for mapping to amazon-provided DNS
- 10.0.0.3 -  reserved by AWS for future use
- 10.0.0.255 - Network Broadcast Address. AWS does not support broadcast in a VPC, therefore the address is reserved.

# Connectivity

# Internet Gateway (IGW)

Allows resources (EC2 instances) in VPC to connect to the internet.

It scales horizontally and is highly available and redundant

Must be created separately from a VPC

One VPC can only be attached to one IGW and vice versa

IGWs on their own do not allow Internet access…

Route tables must also be edited to add routes from subnets.

# Virtual Private Gateway

To access private resources in a VPC, you can use virtual private gateway.

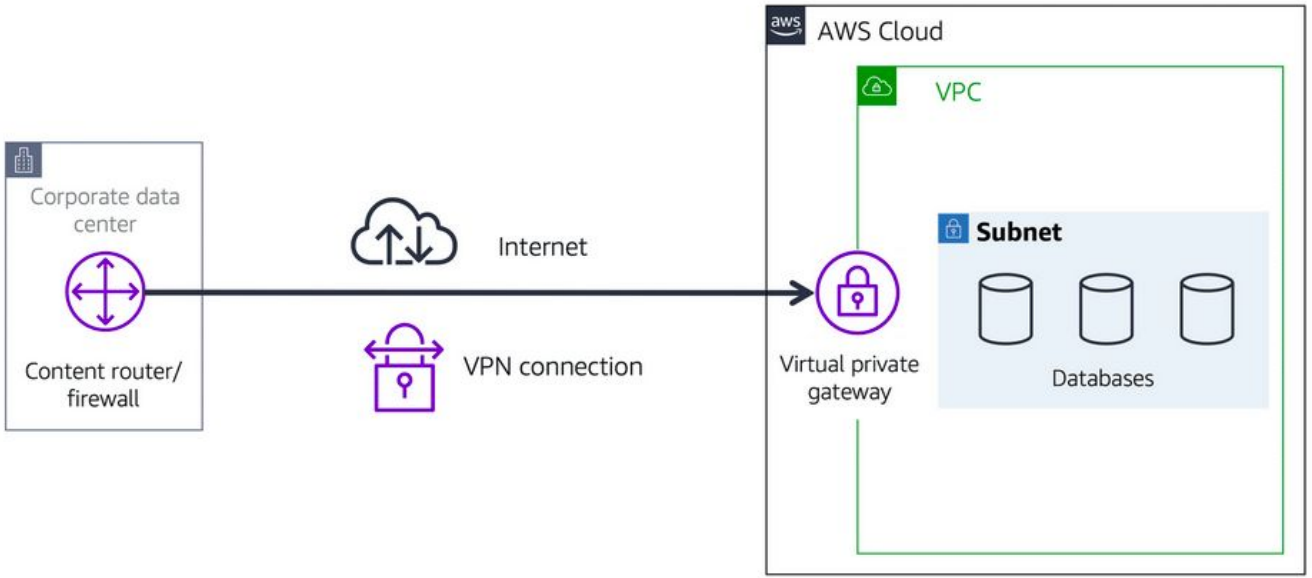
A Virtual Private Network(VPN) connection encrypts or protects your internet traffic from all other requests around it.

A Virtual Private Gateway is the component that allows protected internet traffic to enter into the VPC.

A virtual private gateway enables you to establish a virtual private network connection between your VPC and private network, such as on-premises data center or internal corporate network.

A Virtual Private Gateway allows traffic into VPC only if it is coming from an approved network.

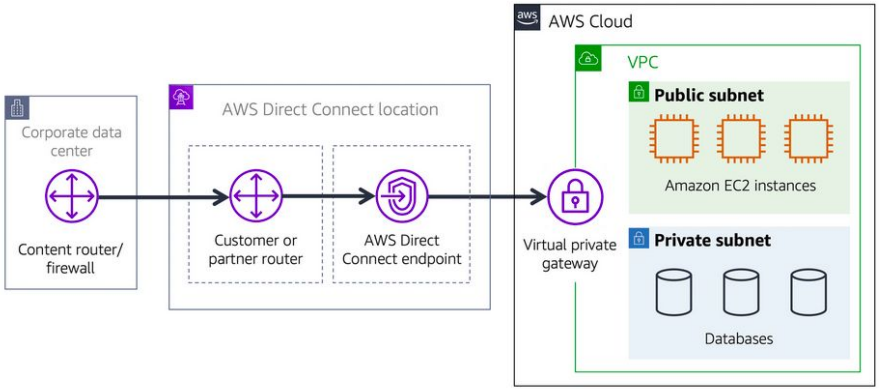
# Virtual Private Gateway



# AWS Direct Connect

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

The private connection that AWS Direct Connect provides helps you reduce network costs and increase the amount of bandwidth that can travel through your network.



# Bastion Host & Nat Gateway

**Bastion host**

We can use a Bastion host to SSH into our private EC2 instances

The bastion is in the public subnet which is then connected to all other private subnets

Bastion Host security group must be tightened.

**Nat Gateway**

It allows EC2 instances in private subnets to connect to the internet.

AWS-managed NAT, higher bandwidth, high availability no administration.

Pay per hour for usage and bandwidth.

NATGW is created in a created in specific Availability Zone, uses an Elastic IP.

Can’t be used by EC2 instance in the same subnet (only from other subnets).

# NACL and Security Groups

NACL (Network Access Control List) are like a firewall control traffic from and to subnets.

One NACL per subnet, new subnets are assigned the default NACL

You define NACL Rules:

- Rules have a number (1 - 32766), higher precedence with lower number.
- First rule match will drive the decision.
- Example: if you define# 100 ALLOW 10.0.0.1/32 and #200 DENY 10.0.0.1/32, the IP address will be allowed becaused 100 has a higher precedence over 200.
- The last rule is an asterisk (*) and denies a request in case of no case of rule match.
- It is recommended adding rules by increment of 100

Newly created NACLs will deny everything

NACL are a great way of blocking a specific IP address at the subnet level.

The Default NACL accepts everything inbound/outbound with subnets it's associated with.

Don't modify the Default NACL, instead create custom NACLs.

# VPC Peering

Privately connect two VPCs using AWS' network

Make them behave as if they were in the same network.

Must not have overlapping CIDRs

VPC Peering connection is not transitive(Must be established for each VPC that need to communicate with one another).

You must update route tables in each VPCs subnets to ensure EC2 instances can communicate with each other.

Note:

- You can create VPC Peering connection between VPCs in different AWS accounts/regions.
- You can reference a security group in a peered VPC (works across accounts - same region).

Route tables must also be edited to add routes from subnets.

# VPC Endpoints

Every AWS service is publicly exposed (public URL)

VPC Endpoints (powered by AWS PrivateLink) allows you to connect to AWS services using a private network instead of using the public Internet.

They are redundant and scale horizontally

They remove the need of IGW, NATGW, … to access AWS services.

Types of endpoint:

1. Interface Endpoint
    a. Provisions an ENI (private IP address) as an entry point (must attach a Security Group).
    b. Supports most aws services.
2. Gateway Endpoints
    a. Provisions a gateway and must be used as a target in route table.
    b. Supports both S3 and DynamoDB.

Route tables must also be edited to add routes from subnets.

# AWS Site to Site VPN

Requires two services:

Virtual Private Gateway (VGW):

1. VPN concentrator on the AWS side of the VPN.
2. VGW is created and attached to the VPC from which you want to create the Sit-to-Site connection.
3. Possibility to customize the ASN (Autonomous System Number)

Customer Gateway (CGW)

1. S/w application or physical device on customer side of the VPNconnection.

# Practical: VPC

# Containers on AWS



ECS, Fargate, ECR

Pathways Technologies

# Amazon Elastic Container Service (ECS)

Amazon's own container platform to launch Docker containers on AWS.

Is a highly scalable and fast container management service.

You must provision and maintain infrastructure (the EC2 instances). AWS takes care of starting and stopping the instances.

Has integrations with the application load balancer (ALB).

There are two launch types:

- ECS launch type: Configure and deploy EC2 instances in your cluster.
- Fargate launch type: Serverless

# Features

A serverless option, AWS Fargate. No need to n=manage servers, handle capacity planning or isolate container workloads for security.

Integration with IAM, assign granular permissions for each of your containers.

As a fully managed service, ECS comes with AWS configuration and operational best practices built-in.

Can create CI/CD pipelines that take the following actions:

- Monitor changes to code
- Build new Docker images from source
- Pushes the image to an image repo such as Amazon ECR or Docker hub
- Updates your amazon ECS services to use the new image in your application.

Support for sending your container instance log information to CloudWatch logs instead of storing them within your container instances.

# When to use Fargate or EC2 launch type

Fargate:

- Large workloads that need to be optimised for low overhead
- Small workloads that have occasional burst
- Tiny workloads
- Batch workloads

EC2 :

- Workloads that require consistently high CPU core and memory usage.
- Large workloads that need to be optimised for price
- Your applications need to access persistent storage
- You must directly manage your infrastructure

# ECS components

**Cluster:** is a logical grouping of tasks or services. You can use clusters to isolate your applications

**Containers and Images:** your app component must be cong=figured to run in containers. Containers are created from a read-only template called image.

**Task definitions;** is a text file that describes one or more containers that form your application. It's in JSON format.

**Task:** is that instantiation of a task definition within a cluster.

**Service:** used to run and maintain your desired number of tasks simultaneously in an Amazon ECS.

**Container agent:** Runs on each container instance within an ECS cluster. Sends info about the current running tasks and resources utilization of your containers to ECS, it starts and stops tasks whenever it receives a request from Amazon ECS.

# IAM Roles for ECS Tasks



EC2 Instance profiles:

- Used by the ECS agent.
- Makes API calls to ECS services.
- Send container logs to cloudwatch logs
- Pull Docker image from ECR
- Reference sensitive data in Secret Manager or SSM Parameter Store

ECS Task Role:

- Allow each task to have a specific role
- Use different roles for the different ECS services you can run
- Task role is defined in the task definition.

# Amazon Elastic Container Registry (ECR)

It’s a fully managed container registry offering high-performance hosting, so you can reliably deploy application images and artifacts anywhere.

Store manage and deploy containers on AWS pay for what you use.

Fully integrated with ECS and IAM for security, backed by Amazon S3.

Support image vulnerability scanning version, tag, image lifecycle

# AWS Lambda



Serverless

Pathways Technologies

# Introduction

A compute service that lets you code without provisioning or managing servers.

Runs your code on a high-available compute infrastructure and performs all of the administration of the compute resources, including server and OS maintenance, capacity provisioning and automatically scaling and logging.

Supports a number of languages: Java, Go, Powershell, Node.js, C#, Python and Ruby

Organize your code into Lambda functions.

You pay for the compute that you consume.

AWS Lambda functions can be configured to run up to 15 per execution. You can set the timeout to any value between 1 second and 15 minutes.

# When to use Lambda?

If you don’t need to manage your own compute resources. Then you should use EC2 or Elastic Beanstalk.

Ec2 your are responsible for provisioning capacity, monitoring fleet, health and performance and using AZs for fault tolerance.

Elastic Beanstalk enables you to deploy and scale apps onto Amazon EC2. You retain ownership and full control over underlying EC2 instances.

# Features

Concurrency and scaling controls: gives you fine-grain control over the scaling and responsiveness of your prod apps

Functions defined as container images: Use your predefined container image tooling, workflows and dependencies to build, test and deploy your Lambda functions.

Code Signing: lambda provides trust and integrity controls that let you verify that only unaltered code that approved developers have published is deployed in your Lambda functions.

Lambda extensions: used to augment your Lambda functions. Easy integration with your favourite tools for monitoring, observability, security and governance.

Function Blueprints: Sample code snippets.

Database access: A DB proxy manages a pool of DB connections and relays queries from the function.

File systems access: Configure function to mount an EFS

# Use cases

1. Web applications: serve the front-end code via S3 and Amazon Cloudfront.
2. Web and mobile backends: the front-ends interact with the backend via API Gateway. Integrated authorization and authentication are provided by Amazon Cognito or APN Partners like Auth0.
3. Data Processing: event-based processing tasks triggered by data changes in data stores or streaming data ETL tasks with Amazon Kinesis and lambda.
4. Parallelized computing tasks: splitting highly complex, long lived computations to individual tasks across many lambda functions instances to process data more quickly in parallel
5. Internet of Things workloads; Processing data generated by physical IoT devices.

# Decoupling applications



SQS, SNS

Pathways Technologies

# Amazon Simple Queue Service (SQS)

Its a fully managed distributed message queueing service.

Enables decoupling and scaling of microservices, distributed systems and serverless applications.

Messages are not pushed to receivers, they have to poll SQS to receive messages.

Max storage 14 days with default of 4 days.

The first 1 million monthly request are free, after you are charged on the number of requests made.

You can delay the visibility on a new message if consumers need time before processing it using ‘delivery delay’

Dead letter queue, a queue targeted by other queues for messages that can’t be processed/consumed.

# Message lifecycle

A producer sends a message to a queue and the message is distributed across th SQS servers redundantly.

A consumer processes a message from the queue when it’s ready. When a message is being processed, it remains in the queue,(to prevent other consumers from processing the message again set visibility timeout).

The consumer deletes message from the queue to prevent from being received and processed again when the visibility timeout expires.

# SQS uses

Used to decouple heavy weight processing

Buffer or batch work

Smooth spiky workloads

Process out of order or ordered messages

Improve resiliency and perceived performance

Design pull-based service-to-service messaging

# Types

**SQS Standard**:

- Is the default queue type.
- A message if delivered at least once, but occasionally more than once.
- Messages may be delivered in an order different from which they were sent.
- Support nearly unlimited number of API calls per second.

Use case:

Decouple live user request from intensive background work

Allocate tasks to multiple worker nodes

Batch messages for future processing

# Types - part 2

**FIFO queues**:

- Have all the capabilities of the standard but designed to enhance ordering of messages.
- A message is delivered once and remains available until a consumer processes and deletes it. No duplicates.
- The order in which messages are sent and received  is strictly preserved.

Use case:

Processing user entered inputs in order entered.

E-commerce order management system where order is critical.

Online ticketing system where tickets are distributed on a first come first served basis.

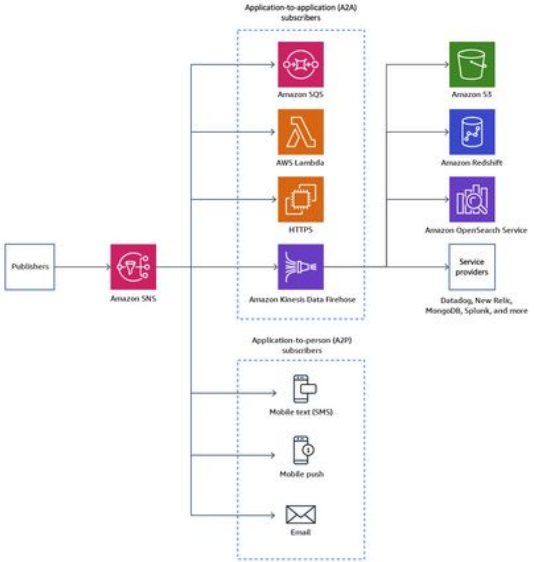
# Amazon Simple Notification Service (SNS)

Is a managed service that provides message delivery from publishers to subscribers.

Publishers communicate asynchronously with subscribers by sending messages to a topic. A topic is a logical access point and communication channel.

Consumers can sub to the SNS topic and receive published messages.

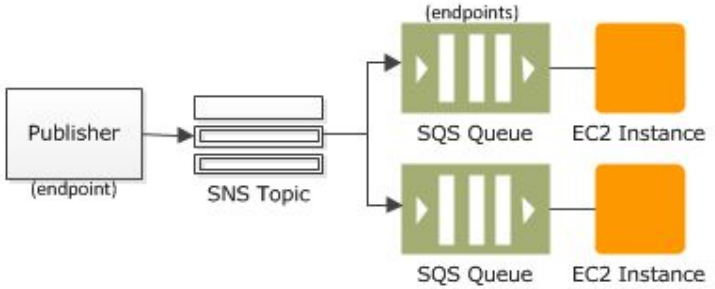
Cost is per number of messages that you publish.



# Fan out pattern

When a message published to an SNS topic is replicated and pushed to multiple endpoints.

This facilitates parallel asynchronous processing.



# Practical: SQS

# Cost optimisation



The well architected framework

Pathways Technologies

# Design principles

The well architected framework provides recommended design principles for cost optimization:

1. Implement cloud financial management. Monitor cost proactively, establish budgets and forecasts, keep up-to-date with new services.
2. Expenditure and usage awareness.
3. Cost effective resources
4. Manage demand and supply resources. Avoid overprovisioning, perform analysis on the workload demand.
5. Optimize over time.

# Useful resources

- AWS pricing calculator
- AWS budgets
- AWS cost explorer
- AWS well architected framework