About AWS, Why AWS

AWS Management Console Introduction

Launching First EC2 Instance (Virtual Machine) (Elastic Compute Cloud)

How to take remote of that Instance using various methods?

Benefits of Cloud Computing

Linux Operating System (Amazon Linux similar to RHEL)

4 Sessions to complete Linux OS

Virtualization
Virtual Machine
Containerization
Container Vs Virtual Machine

AWS Journey
AWS Global Infrastructure (Regions, AZs, Edge Locations, Local & Wavelength Zone)
AWS List of Services

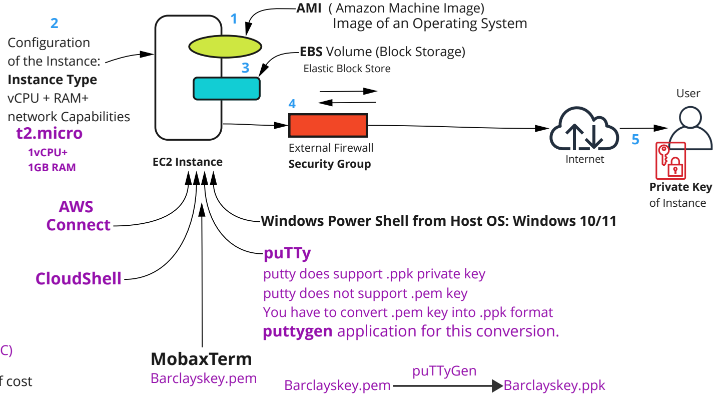
for Introduction

Launching First EC2 Instance (Virtual Machine) (Elastic Compute Cloud)

Launching First Virtual Machine in AWS

Methods to Connect EC2 Instance

1 Core = 2 vCPU



1 **AMI** (Amazon Machine Image)
Image of an Operating System

2 Configuration of the Instance:
Instance Type
vCPU + RAM+ network Capabilities
t2.micro
1vCPU+
1GB RAM

3 **EBS** Volume (Block Storage)
Elastic Block Store

4 External Firewall
Security Group

5 Internet → User, **Private Key** of Instance

EC2 Instance

AWS Connect

CloudShell

Windows Power Shell from Host OS: Windows 10/11

puTTy
putty does support .ppk private key
putty does not support .pem key
You have to convert .pem key into .ppk format
puttygen application for this conversion.

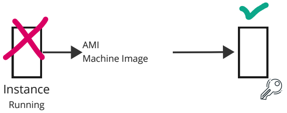
MobaxTerm
Barclayskey.pem

Barclayskey.pem → puTTyGen → Barclayskey.ppk

EC2 (In Free Tier A/C)

750 Hrs/mo Free of cost
using **t2.micro** instance
EBS Volume Size<=30GB

If the Key is lost

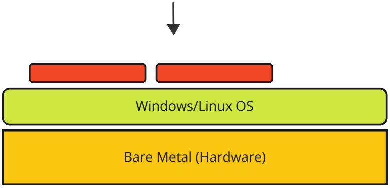


Why AWS?

- more than 200 fully featured services
- total major or minor services > 1500
- AWS is agile.
- AWS Global Cloud Infrastructure
  - Region (geolocation) =27 Live Regions
    - > 86 Availability zones => group of data centers
    - > 400 Edge locations
    - Direct connect
- AWS Free Tier Account
- Performance
- Deployment Speed
- Security

Virtualization

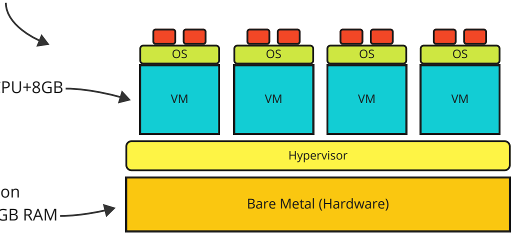
Physical Machine/Server



Performance of Server
70-90%   CPU Wasted
60-90% RAM Wasted

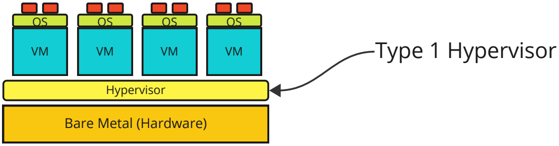
Configuration
16 vCPU+128 GB RAM

Virtualization



2vCPU+8GB

Configuration
16 vCPU+128 GB RAM



Type of Hypervisors

Type 1 Hypervisor

Directly install on Hardware layer

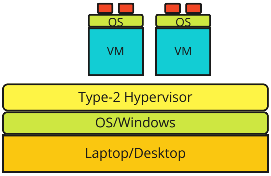
We use Type 1 hypervisors in Data Center

Examples: VMWare ESxi, Citrix XEN, AWS Nitro Hypervisor

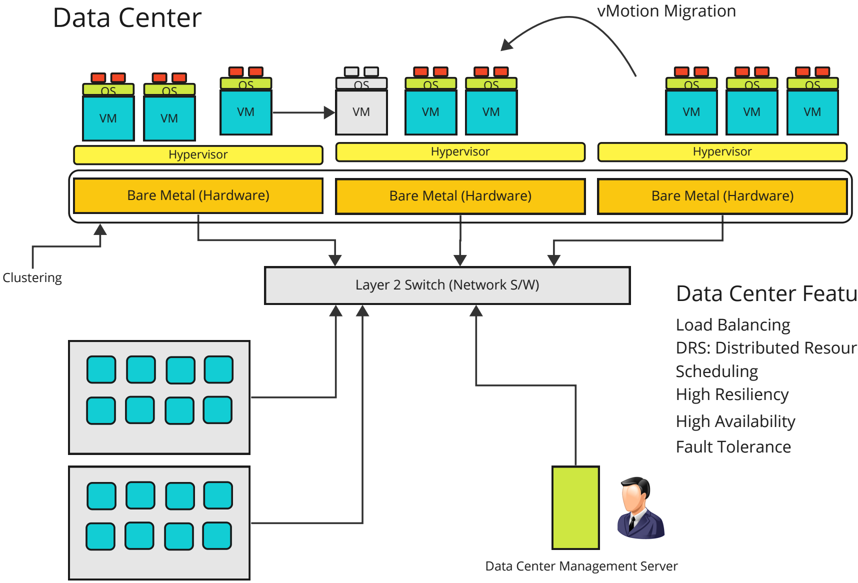
Type-2 Hypervisors

Type2 Hypervisor installed on OS

Examples: Virtual Box, VMWare Workstation etc.



Data Center



Data Center Features

Load Balancing

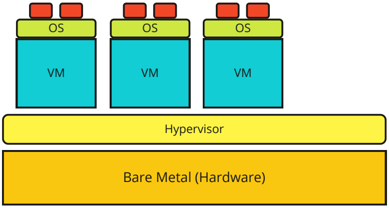
DRS: Distributed Resource Scheduling

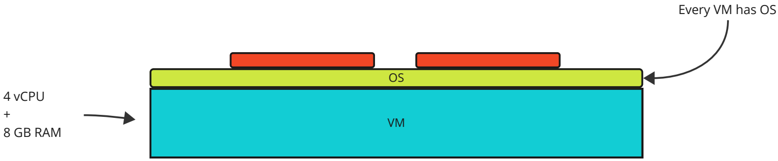
High Resiliency

High Availability

Fault Tolerance

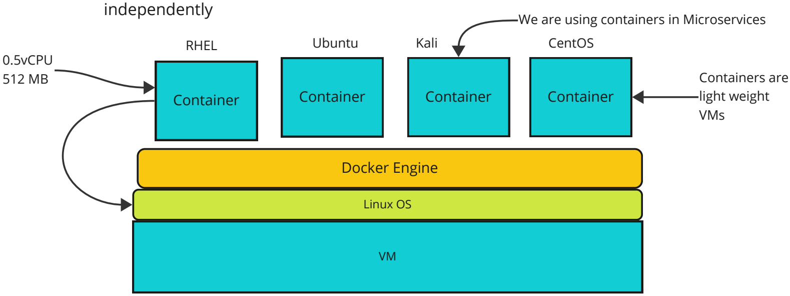
Containerization




Every VM has OS

4 vCPU
+
8 GB RAM

Container shares underlying OSs and has applications with all dependencies to run it independently

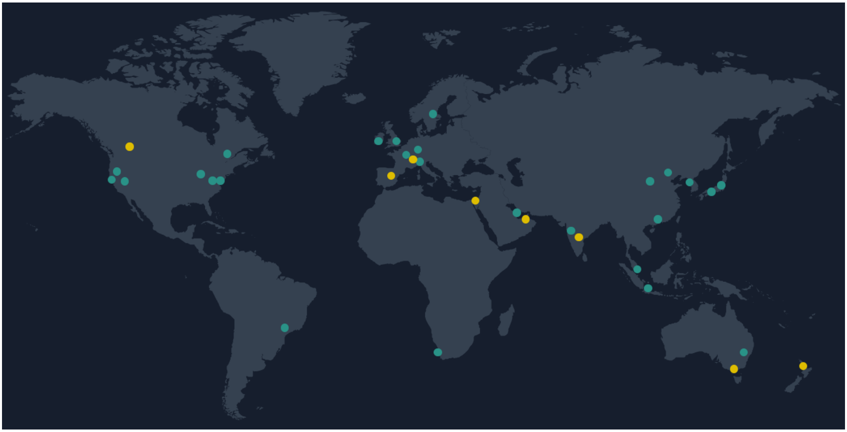

We are using containers in Microservices

0.5vCPU
512 MB

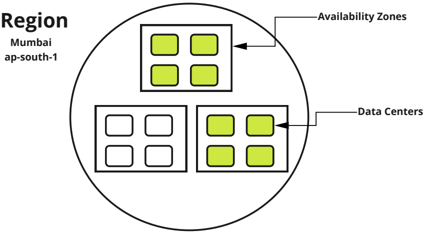
Containers are light weight VMs

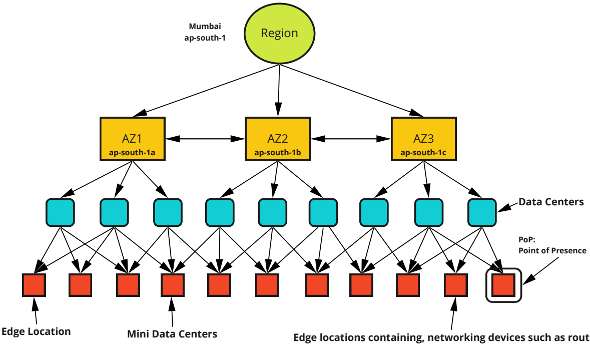
AWS Global Cloud Infrastructure

Region & Availability Zones
Region is an independent and separate geographic location.
Within a Region we have at least 2 or 3 isolated locations called Availability Zones



Region
AZ





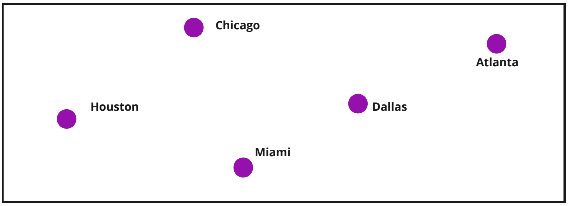
Edge locations containing, networking devices such as routers, switches and also containing number of Servers to cache memory for CDN (Content Delivery Network). You can find these edge location in all major cities, around the world for content distribution. More than 400 Edge Locations are there in AWS Global Infrastructure.

17 Local Zones **28 Wavelength Zones** for ultralow latency applications	**245 Countries and Territories Served**	**115 Direct Connect Locations**

Local Zone

Local Zones are creating near to large populations, or IT & Industrial Hub.

Region: N.Virginia



Wavelength Zone

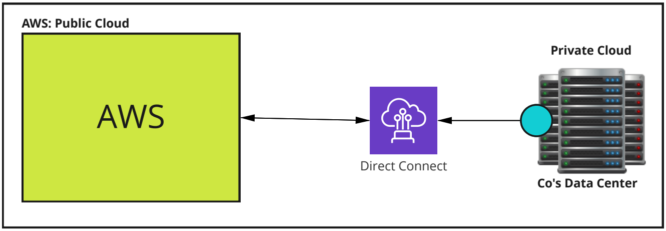
AWS Wavelength is an infrastructure offering optimized for mobile edge computing applications.

This is basically meant for 5G mobile technology.

Direct Connect

It provides connectivity between physical data center with AWS Cloud

Hybrid Cloud



AWS Direct Connect makes it easy to establish a dedicated network connection from your ***premises to AWS***. Using AWS Direct Connect, you can establish private connectivity between ***AWS and your datacenter,*** office, or colocation environment.

AWS EC2 (Elastic Compute Cloud)

EC2 is a compute service

You can Create and Manage Virtual Machines (EC2 Instances)

AMI (Amazon Machine Image)

Instance Types

EBS Volume (Elastic Block Store)

Security Group

EIP (Elastic IP)

Key Pairs

Snapshot

Bootstrapping

Bastion Host

IAM & AWS CLI

Identity and Access Management

Users

Roles

Policies

AWS CLI

EC2 (Elastic Compute Cloud)

AMI (Amazon Machine Image)

Template of OS, Image of OS

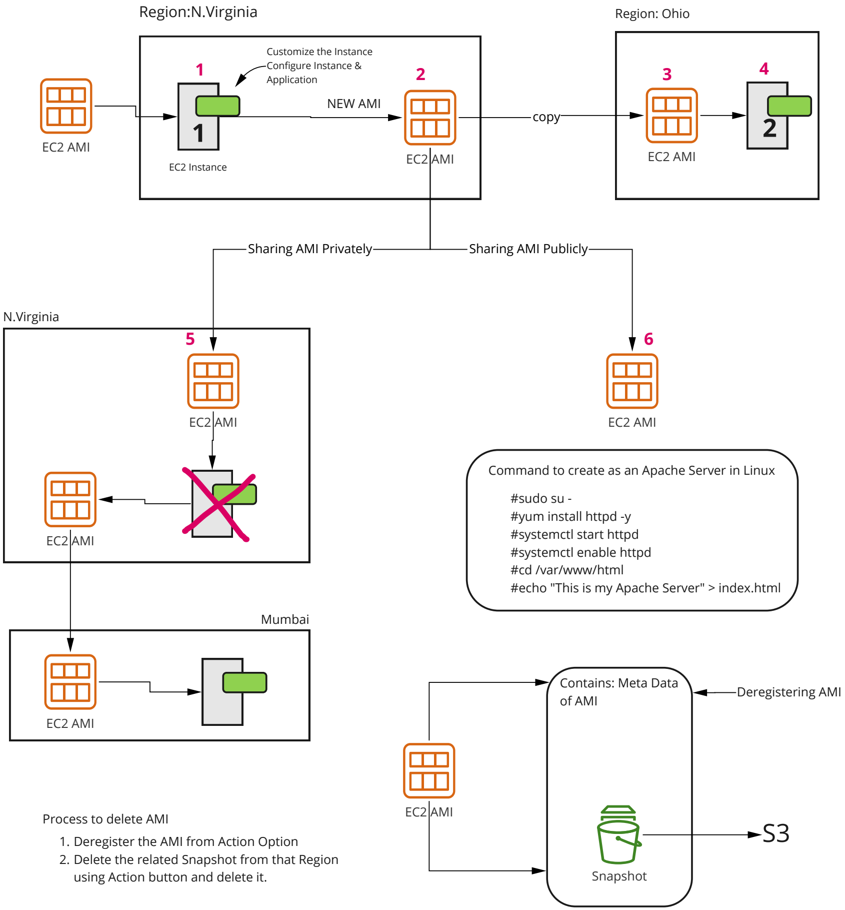
It contains OS + Configuration of OS + Applications + User Data

Quick Start

My AMI

AWS Marketplace

LAB



Command to create as an Apache Server in Linux

```
#sudo su -
#yum install httpd -y
#systemctl start httpd
#systemctl enable httpd
#cd /var/www/html
#echo "This is my Apache Server" > index.html
```

Process to delete AMI

1. Deregister the AMI from Action Option
2. Delete the related Snapshot from that Region using Action button and delete it.

Instance Type

configuration of Instance: contains vCPUs, RAM, Network Interface Capabilities, & EBS Volume throughput etc.

Number of Families within the Instance Type

1 ***General Purpose*** provides a balance of compute, memory and networking resources.

m5n.8xlarge

2 ***Compute Optimized***

3 ***Memory Optimized***

4 ***Storage Optimized***

5 ***Accelerated Computing***

Instance Plans: (Scheme)

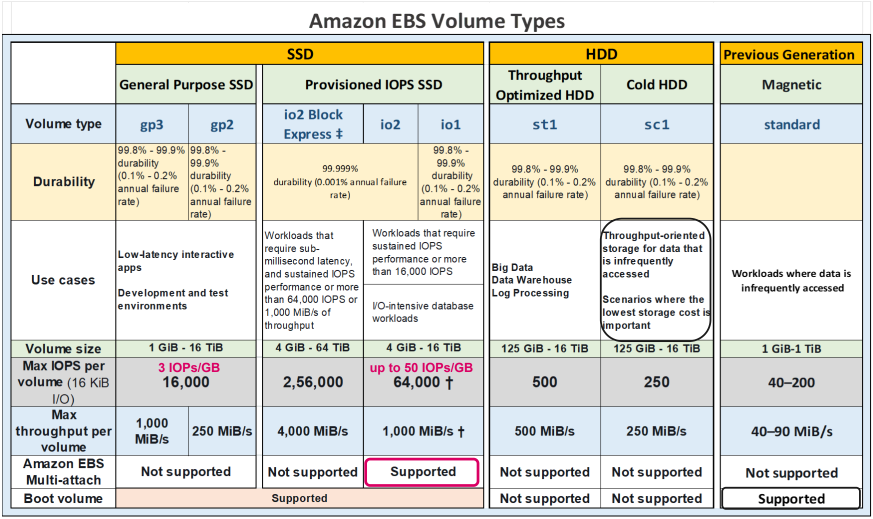
On Demand (you will pay for every hour)

Reserved Instance: (for either 1 entire year or 3 year time)

you will get good discount, it could be approx 30-40%

Spot Instances: created on free resources of AWS Data Centers

getting upto 80% discount



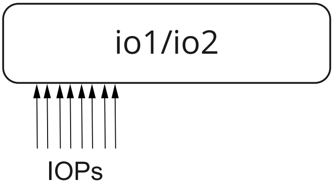
Amazon EBS Volume Types

	SSD					HDD		Previous Generation
	General Purpose SSD		Provisioned IOPS SSD			Throughput Optimized HDD	Cold HDD	Magnetic
Volume type	gp3	gp2	io2 Block Express ‡	io2	io1	st1	sc1	standard
Durability	99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)	99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)	99.999% durability (0.001% annual failure rate)		99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)	99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)	99.8% - 99.9% durability (0.1% - 0.2% annual failure rate)	
Use cases	Low-latency interactive apps Development and test environments		Workloads that require sub-millisecond latency, and sustained IOPS performance or more than 64,000 IOPS or 1,000 MiB/s of throughput	Workloads that require sustained IOPS performance or more than 16,000 IOPS I/O-intensive database workloads		Big Data Data Warehouse Log Processing	Throughput-oriented storage for data that is infrequently accessed Scenarios where the lowest storage cost is important	Workloads where data is infrequently accessed
Volume size	1 GiB - 16 TiB		4 GiB - 64 TiB	4 GiB - 16 TiB		125 GiB - 16 TiB	125 GiB - 16 TiB	1 GiB-1 TiB
Max IOPS per volume (16 KiB I/O)	3 IOPs/GB 16,000		2,56,000	up to 50 IOPs/GB 64,000 †		500	250	40–200
Max throughput per volume	1,000 MiB/s	250 MiB/s	4,000 MiB/s	1,000 MiB/s †		500 MiB/s	250 MiB/s	40–90 MiB/s
Amazon EBS Multi-attach	Not supported		Not supported	Supported		Not supported	Not supported	Not supported
Boot volume	Supported					Not supported	Not supported	Supported

Provisioned IOPs

IOPs= Input/Output Operations/sec (these are number of thread to read/write data with any volume)

Capacity of Volume
4Gb - 64 TB



io1/io2

IOPs

High Performance EBS Volumes

max.no.of IOPs 64000 (Express Version)

io1/io2 up to 50 IOPs per GB

Example: 100 GB (max. 100 x 50 = 5000 IOPs)

General Purpose

SSD Volume

Capacity of Volume (1 GB to 16 TB)

Example: 100 GB = 3 x 100 = 300 IOPs
1500 GB = 3 1500 = 4500 IOPs

min no. of IOPs = 100
max. no. of IOPs = 16000

Multi-Attach EBS Volume

condition to attach volume with multiple instances

Volume Type: io1/io2

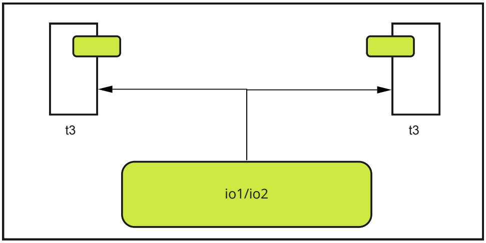
All instances & the volume should be in the same Availability Zone

Use only Nitro Based Instance

A1, C5, C5a, C5ad, C5d, C5n, C6a, C6g, C6gd, C6gn, C6i, C6id, D3, D3en, DL1, G4, G4ad, G5, G5g, Hpc6a, I3en, I4i, Im4gn, Inf1, Is4gen, M5, M5a, M5ad, M5d, M5dn, M5n, M5zn, M6a, M6g, M6gd, M6i, M6id, p3dn.24xlarge, P4, R5, R5a, R5ad, R5b, R5d, R5dn, R5n, R6g, R6gd, R6i, R6id, **T3**, T3a, T4g, u-3tb1.56xlarge, u-6tb1.56xlarge, u-6tb1.112xlarge, u-9tb1.112xlarge, u-12tb1.112xlarge, VT1, X2gd, X2idn, X2iedn, X2iezn, and z1d

LAB

Availability Zone



t3

t3

io1/io2

Conditions and Requirements to configure multi attach volumes.

LAB

How to create an additional volume to connect with EC2 Instance

How to connect it with instance?

How to configure the volume for the instance?

Attach the volume
Create a partition in the volume
Provide File System
Mount the partition

Elastic IP Address (EIP)

EIP is a Fixed or static public IPv4 address

Chargeable but in AWS Free Tier A/c One EIP is Free

Max 5 EIPs can be allocated to your AWS Account Region

EIPs are needed for DNS (Domain Name System) reverse entry or EIP is required for NAT(Network address translation) Gateway

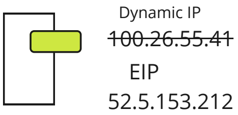
EIPs are also needed for Global Accelerators

if you need fixed or non changeable IP address in that case EIP will be used.

By default VMs (instances) will have dynamic public IP address

To provide public Static IP (EIP) is a two step process.

1. Allocate an EIP to your AWS Account
2. Associate the EIP with your EC2 Instance or Global Accelerator or NAT G/W



Procedure/steps to dissassociate the EIP from the Instance

1. Dissassociate the EIP from the Instance
2. Release the EIP from you AWS Account.

Security Group (External F/W for Instance)

Snapshot (method to take backup of EBS)

Configuration/concept of Bastion Host/Jump Server

Bootstrapping (method to configure VM at launch time using a script)

EFS (NAS)

Que: What is the need of EIP? When it is required?

Calculate EBS Volume Price:

type : io2
size : 1000 GB
IoPs : 5000

$ 450/mo
$ 471/mo

Security Group: External Firewall to be attached with EC2 Instance

Its a bunch of firewall rules

You would write rules in Security Group to allow or restrict traffic

You can connect multiple security groups with one instance.

Max 5 security groups can be connected with one instance

Max 2500 security groups can be created per region/VPC

You find rules written in Security Group are **permissive** in nature, it means you cannot create rules that deny access.

Security Group are **stateful in nature**. In stateful, when you send a request from your instance, acknowledgement traffic for that request is allowed.

Security Group Sections

Inbound: to filter incoming traffic

Max 60 rules can be written in Inbound

It does filter the traffic on the basis of **Protocol**, **Port Number** and **IP address/NID**

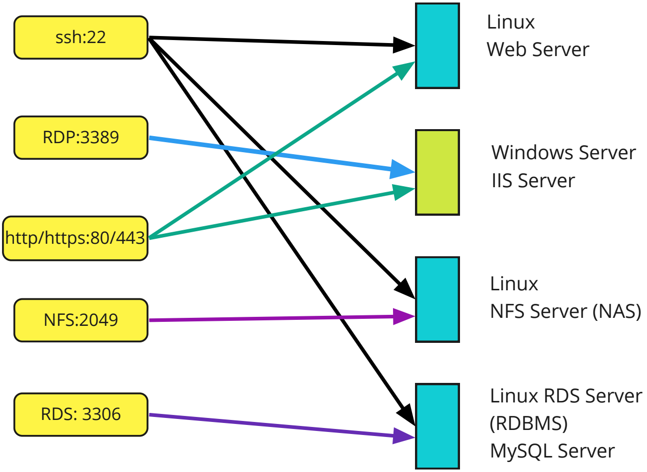
By default, in inbound, ***all traffic is denied, therefor you will write rules to allow traffic***

Outbound: to filter outgoing traffic

Max 60 rules can be written in Outbound

It does filter the traffic on the basis of Protocol, Port Number and IP address/NID

By Default, in outbound, **all traffic is allowed**



EC2 Key Pair:

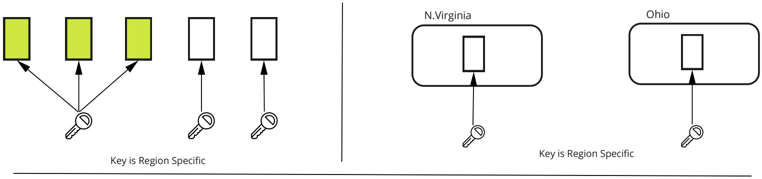
Public Key Cryptography

Public Key

It is used to encrypt the information, and it belongs to AWS

Private Key

It is used to decrypt the encrypted information, and we download it.



You can also use third party tools to create and use your own key pairs

AWS generated key used **2048** bit and ***SSH-2 RSA*** Algorithm

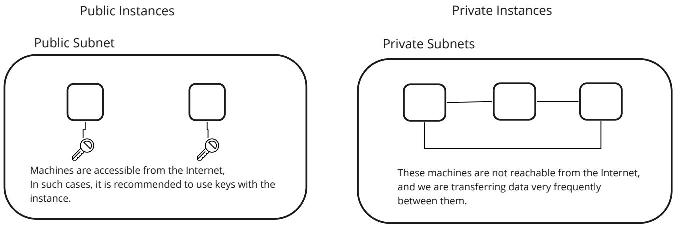
AWS account can have upto 5000 keys pairs per region

finds keys both in .pem or .ppk format

.pem is used for OpenSSH tools

.ppk is used to puTTy

Important: you can also launch an Instance without a key pair



Machines are accessible from the Internet,
In such cases, it is recommended to use keys with the instance.

These machines are not reachable from the Internet, and we are transferring data very frequently between them.

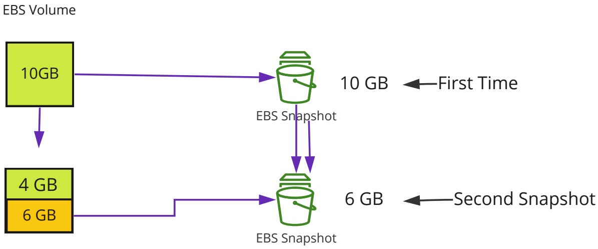
Snapshot

Snapshot is a backup and recovery method for EBS volume

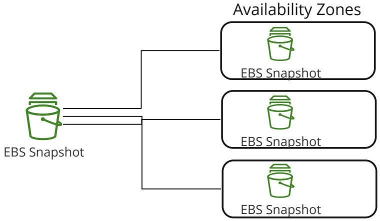
The snapshot is a ***point in time*** backup of an EBS Volume

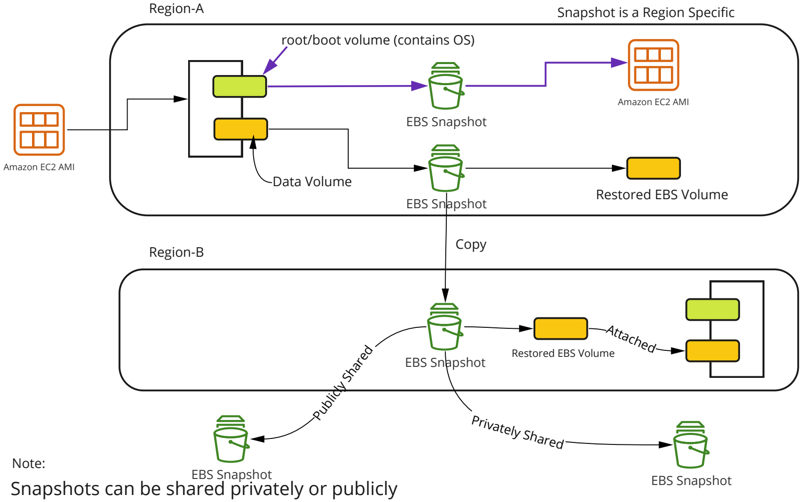
EBS Snapshots are incremental and cost effective solution

if multiple backups are taken of a volume, they are incremental



Snapshots are stored in ***S3 storage space***





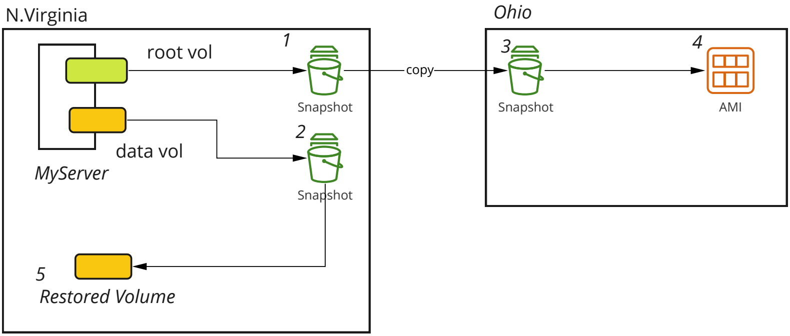
Note:

Snapshots can be shared privately or publicly

Task #1

Assignment

Time = 15 Minutes



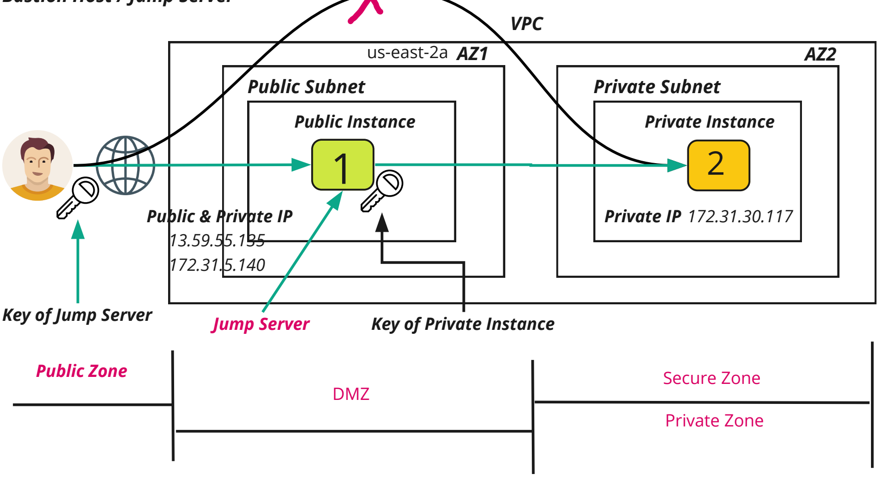
Bootstrapping: *It is a method to configure EC2 Instance at launch time using either Shell Script (Linux) or Power shell script for Windows*

Let's configure Apache Server using Bootstrapp

```
#!/bin/bash
sudo su -
yum install httpd -y
systemctl start httpd
systemctl enable httpd
cd /var/www/html
echo "This is my First Bootstrap Server" > index.html
```

Bastion Host / Jump Server

Windows Instance supports both RDP & SSH protocol



IAM (Identity and Access Management)

Users
Groups
Roles
Permissions

IAM can be controlled through:

AWS Management Console
AWS Command Line Interface (CLI)
AWS SDKs

Some Technical Concepts:

Principals

Three Principals

IAM Root Users
IAM User
Roles or Temporary Security Tokens

Users

Programmatic Access User (AWS CLI, AWS SDKs, or with other development Tools)

In this you will have **access key** and **secret access key** for the user

AWS Management Console User

You will use user interface for login, need username and password

Team (IT)

Production

VM
Service
Configuration

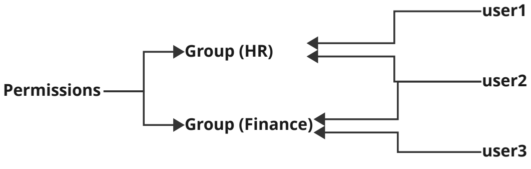
Training

User: John

EC2 Read Only Access

S3 Read Only Access

Cloudwatch Full Access



user1

Group (HR)

user2

Permissions

Group (Finance)

user3

AWS CLI (Command Line Interface)

How to install AWS CLI in your host machine (laptop)?

Launching Instance using cli:

```
aws ec2 run-instances --image-id ami-xxxxxxxx --count 1 --instance-type t2.micro --key-name MyKeyPair --security-group-ids sg-903004f8 --subnet-id subnet-6e7f829e
```

```
aws ec2 run-instances --image-id ami-0de5988cb74e68c72 --count 1 --instance-type t2.micro --key-name BarclaysKey --security-group-ids sg-0191124cc71ae8a6a --subnet-id subnet-0cd2381d2ad7a0321
```

Permissions: (Policies) >850

JSON to write policies

Visual Editor Tool

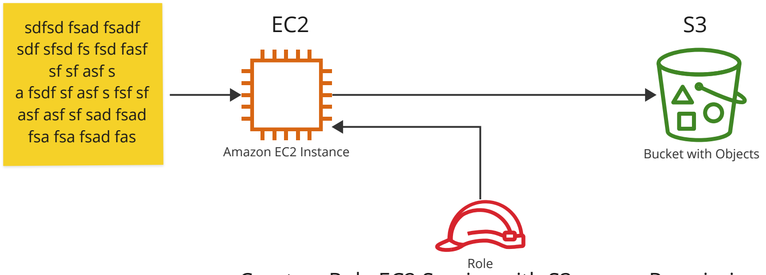
Scenario: need to create a user with very limited access over EC2 Service, and for that you need to write a policy in Visual Editor

how to login as a root account in AWS CLI?

Root credentials:

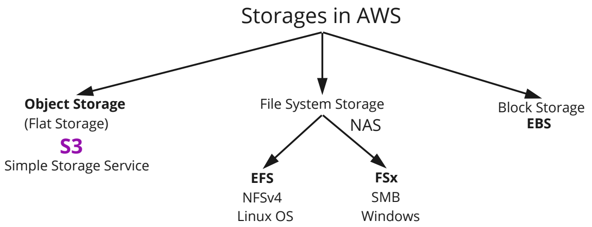
IAM Role:

it is an identity to provide access to one AWS Service to access another AWS Service



Create a Role EC2 Service with S3 access Permission

Storages in AWS



Object Storage
(Flat Storage)
S3
Simple Storage Service

File System Storage
NAS

EFS
NFSv4
Linux OS

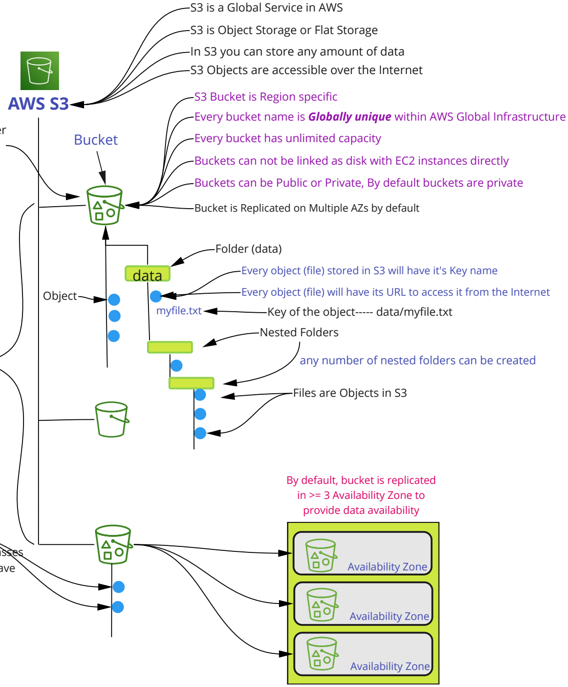
FSx
SMB
Windows

Block Storage
EBS

Understanding AWS S3 (Simple Storage Service)

Sanjay Sharma

Object Storage
Flat Storage
Storage Accessible over the Internet



AWS S3

S3 is a Global Service in AWS
S3 is Object Storage or Flat Storage
In S3 you can store any amount of data
S3 Objects are accessible over the Internet

Bucket

S3 Bucket is Region specific
Every bucket name is ***Globally unique*** within AWS Global Infrastructure
Every bucket has unlimited capacity
Buckets can not be linked as disk with EC2 instances directly
Buckets can be Public or Private, By default buckets are private
Bucket is Replicated on Multiple AZs by default

Folder (data)
data
Every object (file) stored in S3 will have it's Key name
Every object (file) will have its URL to access it from the Internet
Object
myfile.txt
Key of the object----- data/myfile.txt
Nested Folders
any number of nested folders can be created
Files are Objects in S3

By default, bucket is replicated in >= 3 Availability Zone to provide data availability

Bucket is a kind of container for objects (files)

Buckets can be used for:

- Static Web Site Hosting
- Versioning
- Replication (Same & Cross Region)
- Object Life Cycle Management
- Data Encryption
- Server Access Logging
- Transfer Acceleration
- Event Notification

S3 Quota or Limitation:

By default, you can create up to ***100 buckets*** in each of your AWS accounts. This limit can be increased up to ***1000*** Bucket on demand.

S3 Object Storage Class

Amazon S3 offers a range of storage classes designed for different use cases or to save cost.

- *Standard*
- *Intelligent-Tiering*
- *Standard -IA (Infrequent Access)*
- *One Zone - IA*
- *Glacier Instance Retrieval*
- *Glacier Flexible Retrieval*
- *Glacier Deep Archival*

LAB:

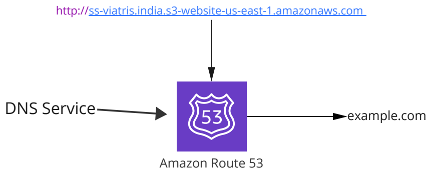
S3
1. *Creating Bucket*
2. *Uploading Files (Objects)*
3. *Sharing Objects*
4. *Access Objects over the Internet*
5. Deleting S3 Bucket

Storage Class
Web Site Hosting

In S3 5GB Space is Free of Cost in Free Tier

Static Web Site Hosting

Pre-Requisite: Template of static Website



http://ss-viatris.india.s3-website-us-east-1.amazonaws.com

DNS Service

Amazon Route 53

example.com

Versioning

To maintain multiple variants on an object is called versioning

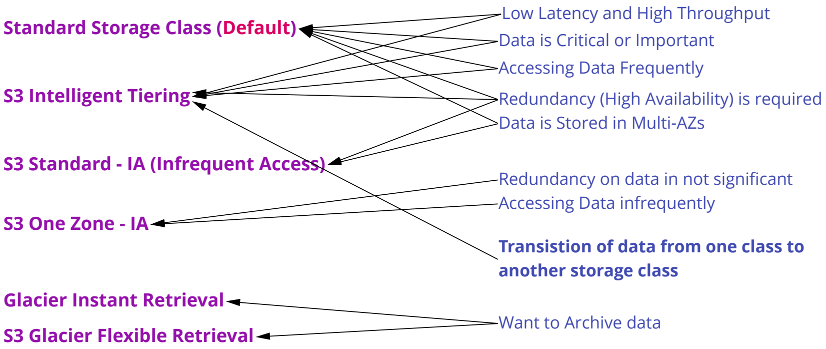
Versioning is disabled by default on buckets

Once you enable it, can't be disabled, it can only be suspended.

if the object is deleted accidently, can be restored because of versioning

Storage Class

Use case of data



Standard Storage Class (Default)

S3 Intelligent Tiering

S3 Standard - IA (Infrequent Access)

S3 One Zone - IA

Glacier Instant Retrieval

S3 Glacier Flexible Retrieval

Low Latency and High Throughput

Data is Critical or Important

Accessing Data Frequently

Redundancy (High Availability) is required

Data is Stored in Multi-AZs

Redundancy on data in not significant

Accessing Data infrequently

Transistion of data from one class to another storage class

Want to Archive data

In One zone, data is stored in single AZ while in Standard-IA data is stored in multiple AZs

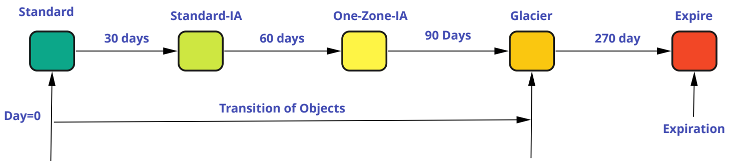
S3 Pricing

Standard Storage Class (Default)	$**0.023** per GB/mo
S3 Intelligent Tiering	$0.023 per GB
S3 Standard - IA (Infrequent Access)	$0.0125 per GB
S3 One Zone - IA	$0.01 per GB
Glacier Instant Retrieval	$0.004 per GB
S3 Glacier Flexible Retrieval	$0.0036 per GB

Life Cycle Management

Transition Action (From one storage class to another storage Class)

Expiration Action (or need to delete objects after a certain period of time)



Standard

30 days

Standard-IA

60 days

One-Zone-IA

90 Days

Glacier

270 day

Expire

Day=0

Transition of Objects

Expiration

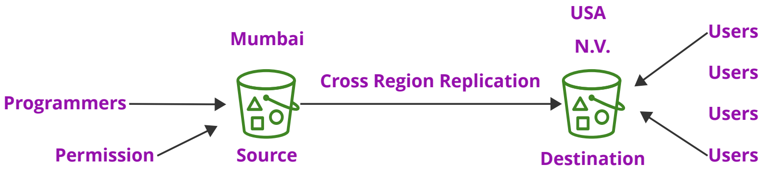
Current Data (new Data/uploading)

non-current data (existing objects)

Replication

Same Region Replication

Cross Region Replication



USA

N.V.

Mumbai

Users

Users

Users

Users

Programmers

Cross Region Replication

Permission

Source

Destination

Pre-Requisites for the replication

Buckets must be in different Regions

Enable Versioning on both buckets

CloudFormation

Stack

Elastic Beankstalk

it's good example of PaaS

It is used to deploy application in AWS

Lambda

AWS Lambda Functions

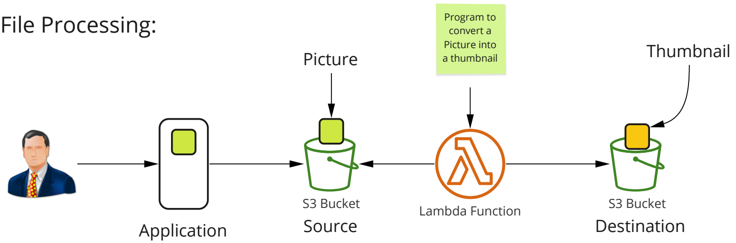
Lambda is a serverless service to run code

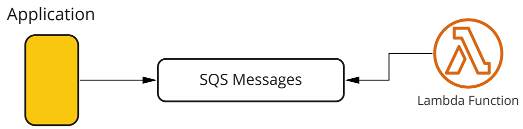
Where Lambda can be used?

Lambda functions and triggers are the core components of building applications on AWS Lambda. A Lambda function is the code and runtime that process events, while a trigger is the AWS service or application that invokes the function. To illustrate, consider the following scenarios:

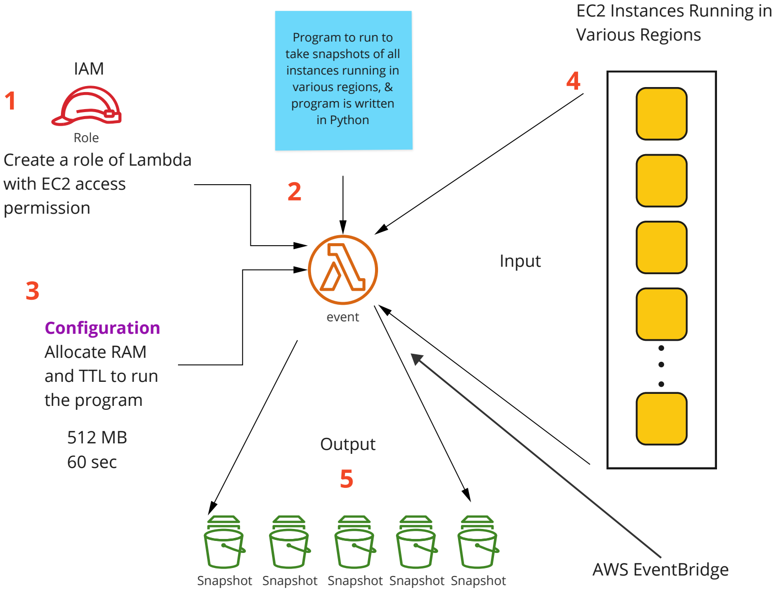
1. ***File processing*** – Suppose you have a photo sharing application. People use your application to upload photos, and the application stores these user photos in an Amazon S3 bucket. Then, your application creates a thumbnail version of each user's photos and displays them on the user's profile page. In this scenario, you may choose to create a Lambda function that creates a thumbnail automatically. Amazon S3 is one of the supported AWS event sources that can publish object-created events and invoke your Lambda function. Your Lambda function code can read the photo object from the S3 bucket, create a thumbnail version, and then save it in another S3 bucket.
2. ***Data and analytics*** – Suppose you are building an analytics application and storing raw data in a DynamoDB table. When you write, update, or delete items in a table, DynamoDB streams can publish item update events to a stream associated with the table. In this case, the event data provides the item key, event name (such as insert, update, and delete), and other relevant details. You can write a Lambda function to generate custom metrics by aggregating raw data.
3. ***Websites*** – Suppose you are creating a website and you want to host the backend logic on Lambda. You can invoke your Lambda function over HTTP using Amazon API Gateway as the HTTP endpoint. Now, your web client can invoke the API, and then API Gateway can route the request to Lambda.
4. ***Mobile applications*** – Suppose you have a custom mobile application that produces events. You can create a Lambda function to process events published by your custom application. For example, you can configure a Lambda function to process the clicks within your custom mobile application.
5. ***Orchestration*** (Task Administration)

File Processing:





LAB:



IAM

1

Role

Create a role of Lambda with EC2 access permission

Program to run to take snapshots of all instances running in various regions, & program is written in Python

2

3

Configuration
Allocate RAM and TTL to run the program

512 MB
60 sec

EC2 Instances Running in Various Regions

4

Input

Output

5

AWS EventBridge

Memory (MB)	Price per 1ms
128	$0.0000000021
512	$0.0000000083

Duration: 12017.01 ms

Billed Duration: 12018 ms

Total Price = 12018 x 0.0000000021 =

I need to run this lambda function daily on a specific time

Memory Size: 512 MB

Max Memory Used: 95 MB

AWS EventBridge

Lambda Function

Lambda Function

Serverless Service to run code
It runs code in a serverless manner without provisioning any Instance

Serverless service has some characteristics

1. No Server management
2. Flexible Scaling
3. No Idle Capacity
4. High Availability

Serverless Applications have several components and layers:

Compute
API
Storage (we have ephemeral storage (temp storage))
Interprocess Messaging
Orchestration

Let's say if TTL = 40 sec, and program takes 10 sec to complete, 10 sec is execution t
Lambda will charge for 10 sec, not for 40 sec.

Lambda provide compute resources to run program.

Use of memory(RAM) and CPU Power are proportionate in Lambda

→ ✓ RAM ↑ → CPU ↑

Min RAM =128 MB
Max RAM = 10 GB

Execution	TTL
30 Seconds	40 sec

→ Lambda can also provide TTL (Time to Live), time in which Lambda Function should be completed. ***TTL is the max amount of time in which program must be completed.***

Eg: if you allocate some TTL = 40 seconds, it means program must be completed within 40 seconds, otherwise lambda function will die.

AWS Lambda will not charge for the TTL, lambda will charge only to execution time.

Pricing in Lambda depends on two factors:

1. Amount of RAM is consumed to run the Program, not allocated RAM

   *eg: if you allocate 512 MB RAM, and program takes only **128 MB** to execute the program, you will pay only for 128 MB RAM, not for 512 MB.*

2 *Lambda charges in milliseconds not in seconds*

eg: you consume 10 sec to run the program, it means you will pay for 10000 ms

1 second = 1000 milliseconds

Example: for 128 MB RAM and 10000 ms Lambda charges $0.0000000021

and you consume 10 sec to run the program
= 10000 x 0.0000000021 =

VPC (Virtual Private Gateway)

An Isolated network within a Region.

Every VPC (Network) will have its CIDR Block (Network ID)

IP Addresses
Within VPC we have some components as below:
Internet Gateway
Subnets
RouteTables
NAT Gateways
EIP
NACL (Network Access Control List)

IP Addresses

IPv4 **IPv6**

Class A,B,C,D and E

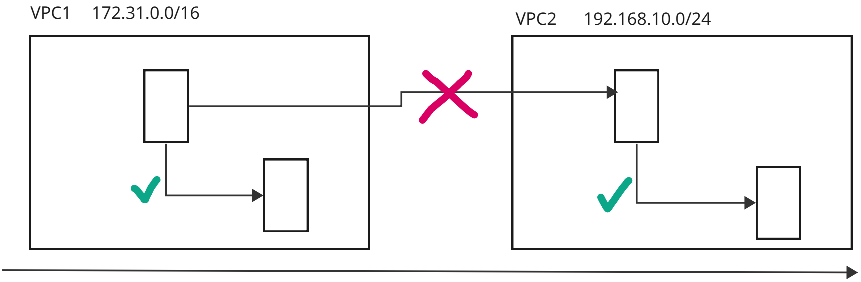
We are using Class A,B and C in AWS VPC

Public IPs

Private IPs

We you create a VPC Generally select Ranges from Private IP Address

Class A	10.0.0.0-10.255.255.255
CLASS B	172.16.0.0 - 172.31.255.255
Class C	192.168.0.0- 192.168.255.255



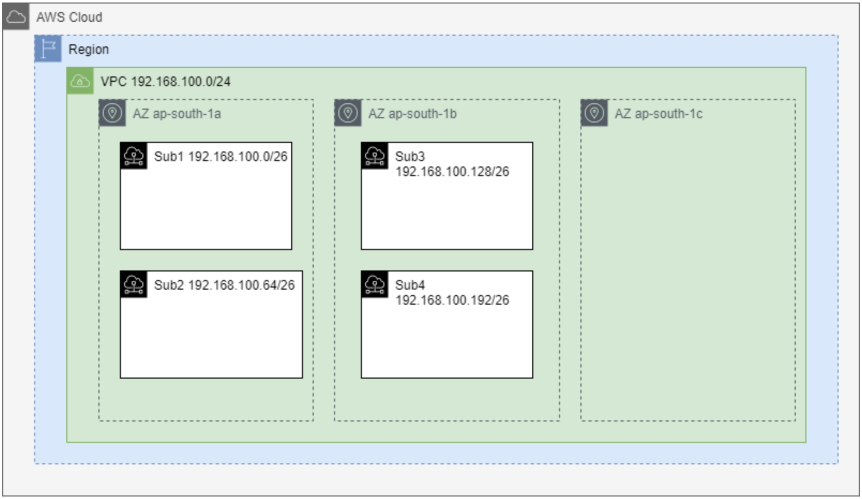
CIDR (Classless Inter Domain Routing)

ClassFull

Class-A 10.0.0.0/8

Class-B 172.31.0.0/16

10.0.0.0/20, 10.0.0.0/24
10.0.0.0/16, 172.16.10.0/24



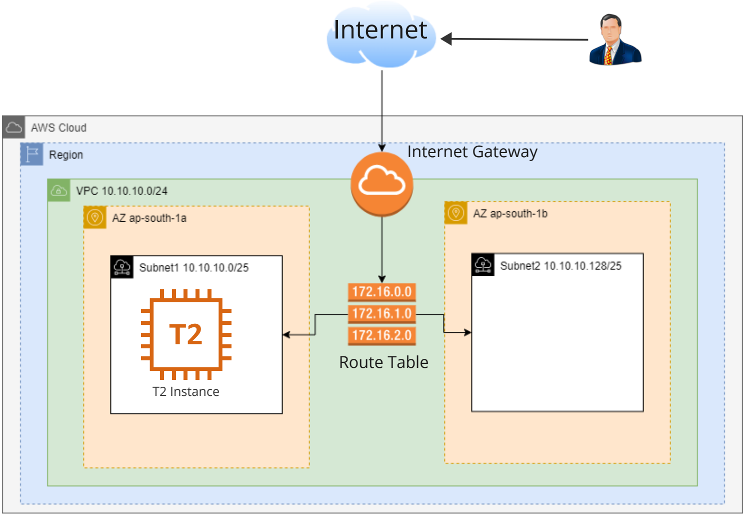
In every region you will find a default VPC with CIDR 172.31.0.0/16

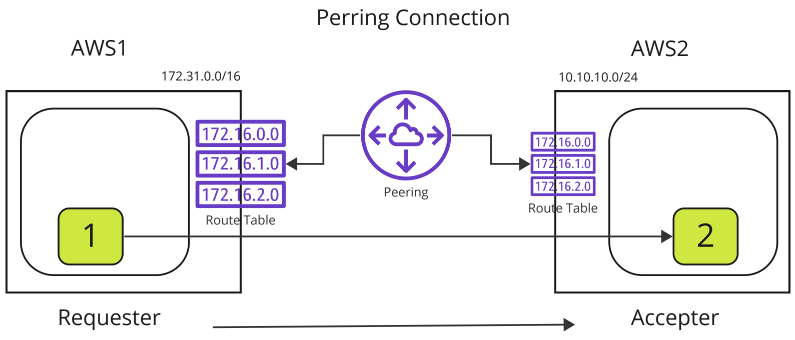
Creating a simple VPC

VPC (NID:CIDR) 10.10.10.0/24

Public Subnets → Subnet1 10.10.10.0/25

Public Subnets → Subnet2 10.10.10.128/25



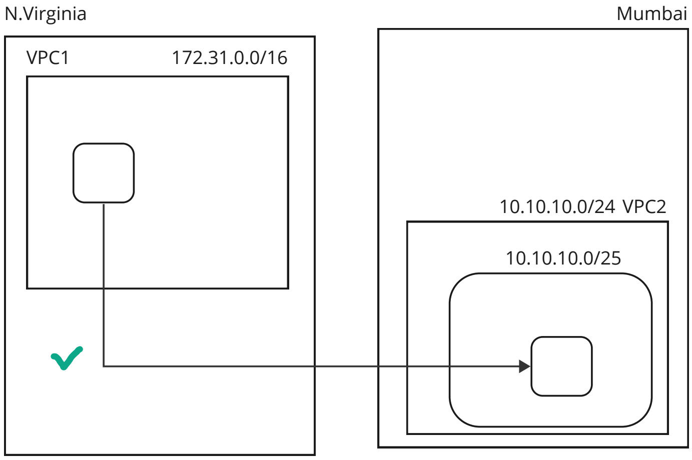


Peering connection provides connectivity between two VPCs These VPCs can be in same Region same AWS A/C or VPCs can be in different accounts.

How to delete Peering connection?

How to delete VPCs?

Assignment



Task:

Assignment

Elastic Beanstalk Work:

https://docs.aws.amazon.com/elasticbeanstalk/latest/dg/tutorials.html

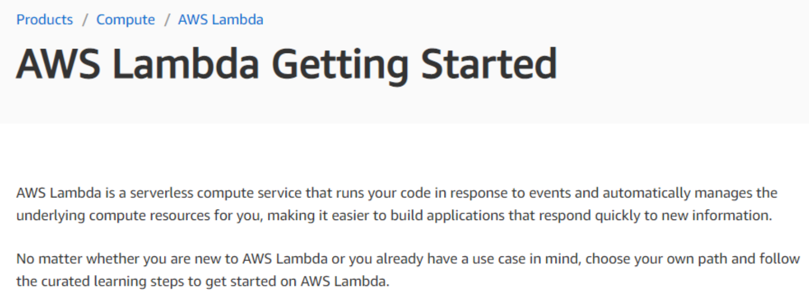
Download Application → nodejs.zip

Deploy it in Elastic Beanstalk, with with load balancing enabled

and also allows Private key enabled on ec2 launched by ElasticBeanstalk

Assignment

https://aws.amazon.com/lambda/getting-started/



Products / Compute / AWS Lambda

AWS Lambda Getting Started

AWS Lambda is a serverless compute service that runs your code in response to events and automatically manages the underlying compute resources for you, making it easier to build applications that respond quickly to new information.

No matter whether you are new to AWS Lambda or you already have a use case in mind, choose your own path and follow the curated learning steps to get started on AWS Lambda.

Assignment

https://serverlessland.com/getting-started/lambda/

Getting Started with Lambda HTTP

CDK (Cloud Development Kit)
RDS (Relational Database Service): SQL Database
Aurora (Serverless)
DynamoDB: NoSQL Database Server; DynamoDB is Serverless
Elasticache
API

What is CDK?

Cloud Development Kit

Infrastructure as Code

TypeScript, JavaScript, Python, Java, C#/.net and Go

About CDK

Lab/Excercise:

Install and configure the CDK CLI

Create you first CDK Project

Deploy your firs infrastructure

Install and configuration of AWS CLI

Some important files

bin/**cdk-demo.ts**: This is the entry point to your CDK application

lib/**cdk-demo-stack.ts**: This is where your main CDK application stack is defined

package.json This is where you define your project dependencies etc

cdk.json: This file tells the toolkit how to run your application

lib/cdk-demo-stack.ts

```
import * as cdk from 'aws-cdk-lib';
import { Construct } from 'constructs';
import * as ec2 from 'aws-cdk-lib/aws-ec2';
// import * as sqs from 'aws-cdk-lib/aws-sqs';

export class CdkDemoStack extends cdk.Stack {
  constructor(scope: Construct, id: string, props?: cdk.StackProps) {
    super(scope, id, props);

    new ec2.Vpc(this, 'mainVPC', {
       maxAzs: 2,
       subnetConfiguration: [
          {
             cidrMask: 24,
             name: 'public-subnet',
             subnetType: ec2.SubnetType.PUBLIC,
          }
       ]
    // The code that defines your stack goes here

    // example resource
    // const queue = new sqs.Queue(this, 'CdkDemoQueue', {
    //   visibilityTimeout: cdk.Duration.seconds(300)
    });
  }
}
```

Assignment

Creating an AWS Fargate service using the AWS CDK

https://docs.aws.amazon.com/cdk/v2/guide/ecs_example.html

AWS RDS (Relational Database Service)

Amazon Aurora

- → **Provision Server**
- → **Serverless**

Oracle, MySQL, Postgesql, MS-SQL,

RDBMS

RDS

Redshift

Datawarehouse

DynamoDB

NoSQL Database

AWS Database Services

RDBMS
NoSQL
In-memory Caching Service

About RDS:

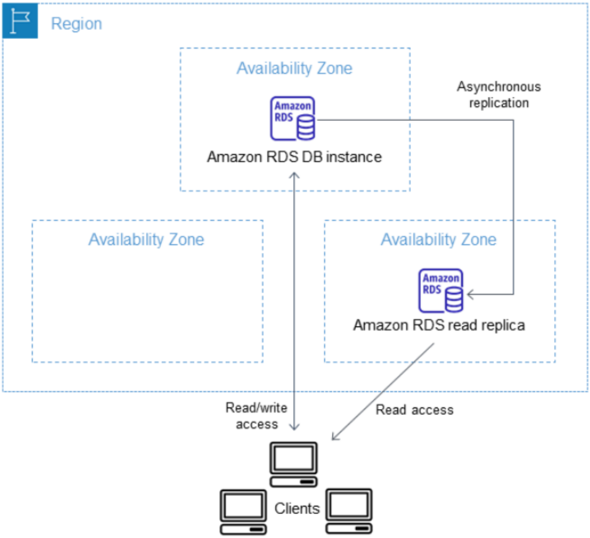
1 **Assignment**

Create a web server and an Amazon RDS DB instance

https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/TUT_WebAppWithRDS.html

This document helps you install an Apache web server with PHP and create a MySQL database. The web server runs on an Amazon EC2 instance using Amazon Linux, and the MySQL database is a MySQL DB instance. Both the Amazon EC2 instance and the DB instance run in a virtual private cloud (VPC) based on the Amazon VPC service

RDS Read Replica



DynamoDB

Unstructured and semi structured Database

No SQL Database use alternate models for data management such as key-value pairs or document storage

Examples of Unstructured database:

Weather Data
Employees Emails
Surveillance Data   IOT
Stock Market Data
Gaming Data

DynamoDB is fully Managed, multiregion, mulimaster, durable database with built in security

It can handle 10 trillion requests per day
20 million requests per second

DynamoDB is a Serverless Service

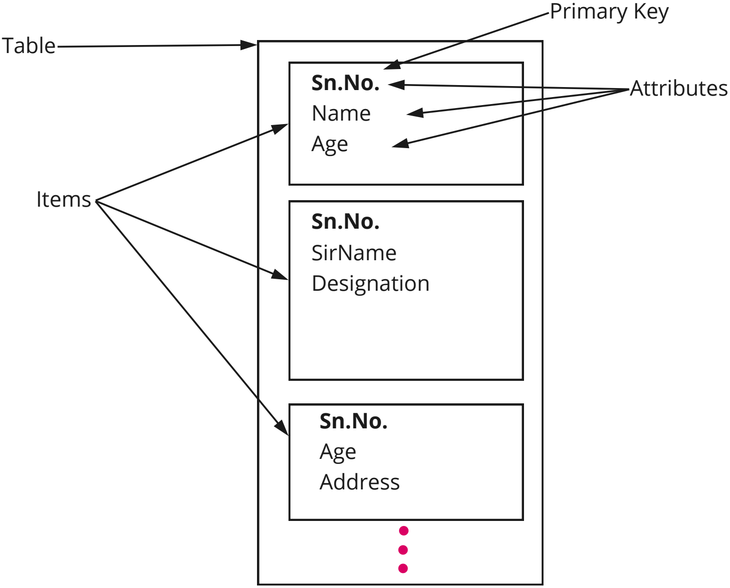
You can store any amount of data in DynamoDB Tables

You can scale up or scale down your **tables throughput** capacity with any downtime.

DynamoDB provides on-demand backup capabilities
Backup of tables is possible
Max retention period is 35 days
it delete expire items from tables automatically .
TTL allows you to define a per-item timestamp.

High Availability and Durability

Tables are replicated in muli-AZs in the Region
Where to use DynamoDB
DynamoDB Table Structure



DynamoDB supports two different kind of Primary Keys

Partition Key

Partition Key and Sort Key (Composite Primary Key)

S.No.
Country

DynamoDB Features:

It allows rapid replication of your data among multiple AZs in a Region.

Read/Write Capacity Mode

1. **On-Demand**
2. **Provisioned (default, free tier eligible)**

On-Demand

it is good if you create new tables with unknown workloads.
it is good if you have unpredictable application traffic
it is good if you prefer the ease of paying for only what you use.

Provisioned

it is good if you have predictable traffic
it is good if you run applications whose traffic is consistent and ramps gradually
it is good if you can forecast capacity requirement to control costs.

DynamoDB Read Capacity Unit (RCU)

Eventually Consistent (Fast and cheaper) (1 RCU = 8 KB/sec)

Strongly consistent (little slow but expensive)(1 RCU = 4 KB/sec)

Transactional (ACID: Atomicity, Consistency, isolation and durability)

DynamoDB Write Capacity Unit (WCU) (1 WCU= 1 KB/Sec)

Pricing: it charges for reading, writing and for storing data.

DynamoDB Limits

Quota of 256 tables per AWS Region

Size: no practical limit

AWS DynamoDB
25GB
Free of Cost

Assignment

Amplify, Lambda & DynamoDB

Build a Basic Web Application

https://aws.amazon.com/getting-started/hands-on/build-web-app-s3-lambda-api-gateway-dynamodb/

In this tutorial, you will create a simple web application. You will first build a static web app that renders "Hello World." Then you will learn how to add functionality to the web app so the text that displays is based on a custom input you provide.

AWS Elasticache

Memcached or Redis Protocol-compliant server nodes

nodes, shards and clusters.

Glue/Kinesis/Cloudwatch/ECS/CodePipeline+/APIGateways/

SAM
(Serverless Application Model)

Components

AWS SAM Template Specification

AWS SAM CLI

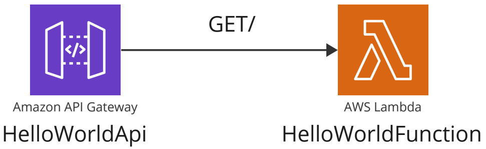
Benefits using SAM

Single-Deployment Configuration

Extensions of AWS CloudFormation

Built in Best Practices

Local Debug and Testing



GET/

Amazon API Gateway

HelloWorldApi

AWS Lambda

HelloWorldFunction

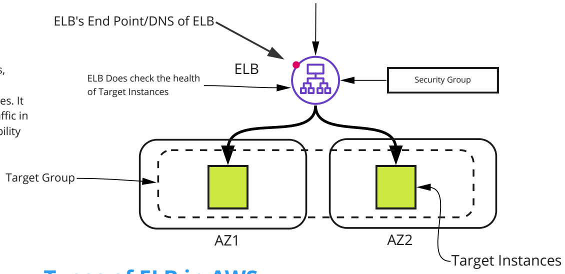
Output → hello world

Sanjay Sharma

Introduction with AWS ELB

Elastic Load Balancer

Elastic Load Balancing automatically distributes incoming application traffic across multiple targets, such as Amazon EC2 instances, containers, IP addresses, Lambda functions, and virtual appliances. It can handle the varying load of your application traffic in a single Availability Zone or across multiple Availability Zones.



ELB's End Point/DNS of ELB

ELB

ELB Does check the health of Target Instances

Security Group

Target Group

AZ1

AZ2

Target Instances

Types of ELB in AWS

ALB
Application Load Balancer

80/443 with http/https
it works on Layer 7
Application Layer

NLB
Network Load Balancer

it does support all logical ports (1-65535), it works on layer 4, Transport Layer

Classic Load Balancer

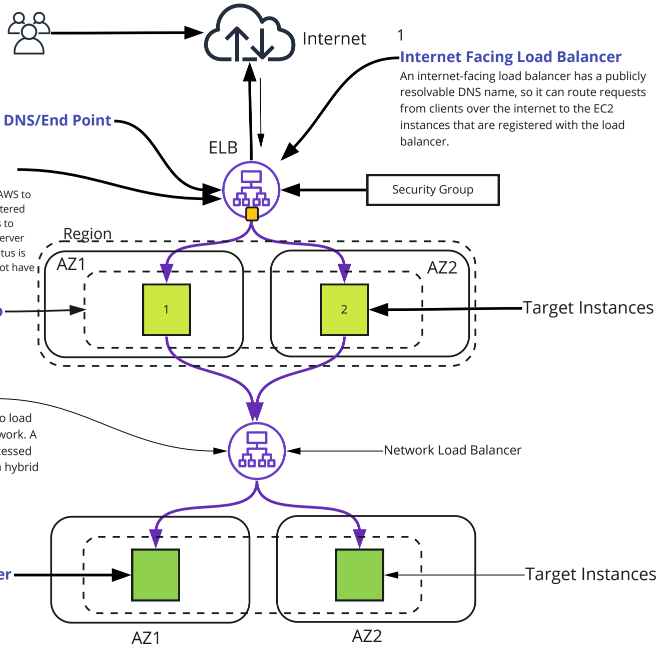
Pervious Generation Load Balancer
It does support both Layer4 and Layer 7

GWLB
Gateway Load Balancer

Gateway Load Balancer makes it easy to deploy, scale, and manage your third-party virtual appliances. It gives you one gateway for distributing traffic across multiple virtual appliances, while scaling them up, or down, based on demand.

Elastic Load Balancer

Type of ELB (Architectural Point of View)



Internet

1

Internet Facing Load Balancer

An internet-facing load balancer has a publicly resolvable DNS name, so it can route requests from clients over the internet to the EC2 instances that are registered with the load balancer.

ELB's DNS/End Point

ELB

ELB does check the health of Target Instances

The ELB Health Check is used by AWS to determine the availability of registered EC2 instances and their readiness to receive traffic. Any downstream server that does not return a healthy status is considered unavailable and will not have any traffic routed to it.

Security Group

Region

AZ1

AZ2

1

2

Target Group

Target Instances

2

Internal Load Balancer

Internal load balancers are used to load balance traffic inside a virtual network. A load balancer frontend can be accessed from an on-premises network in a hybrid scenario.

Network Load Balancer

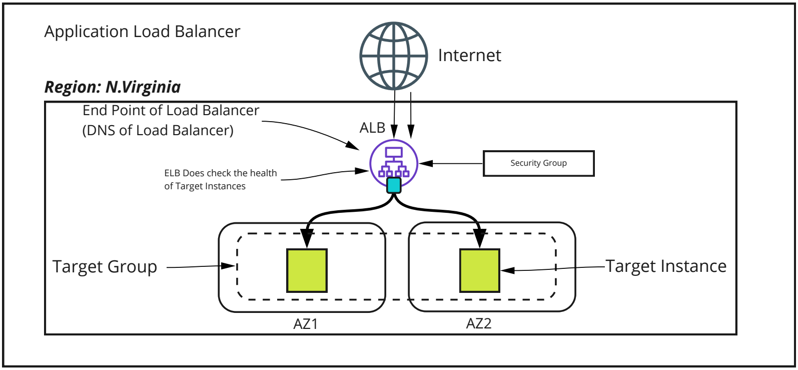
Backend Server

Target Instances

AZ1

AZ2

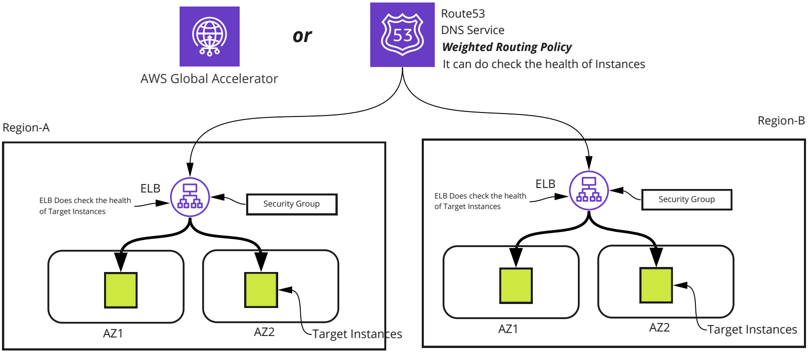
Assignment



Task

Just Open Lightweight Service and create an Instance, verify & delete.

Cross Region Load Balancing



AWS Global Accelerator

or

Route53
DNS Service
Weighted Routing Policy
It can do check the health of Instances

Region-A

ELB

ELB Does check the health
of Target Instances

Security Group

AZ1

AZ2

Target Instances

Region-B

ELB

ELB Does check the health
of Target Instances

Security Group

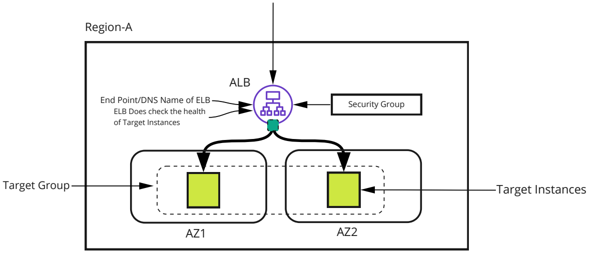
AZ1

AZ2

Target Instances

LAB:

Application Load Balancer



Region-A

ALB

End Point/DNS Name of ELB

ELB Does check the health
of Target Instances

Security Group

Target Group

Target Instances

AZ1

AZ2

Important Facts about ELB:

Traffic Distribution

Continuous Health Check

ELB can be Public or Private

Target Instances must be in diff AZs

Algorithm by default: ***Round Robin***

Every ELB can be accessed using its end-point (DNS name of ELB)

But if ELB is used with ***Global Accelerator***, GA will/can use EIPs

End Point of ELB and also be mapped with a Domain Name using Route53 to open website using a Domain Name.

ELB can also be integrated with Auto Scaling to manage traffic load at backend.

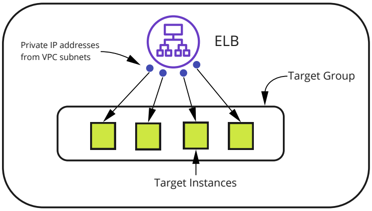
ELB is highly available and Scalable

ELB will connect with Security Group in order to filter traffic

ELB can also be connected with WAF (***Web Application Firewall***)

Internal ELB uses Private IP address to distribute the load within the VPC

We can set ***SSL Certificate*** to allow/configure https traffic



ELB

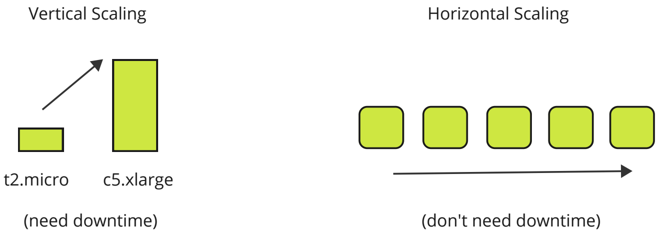
Private IP addresses
from VPC subnets

Target Group

Target Instances

Instances
IP addresses >> ENI
Lambda Functions
Application Load Balancer

Auto Scaling



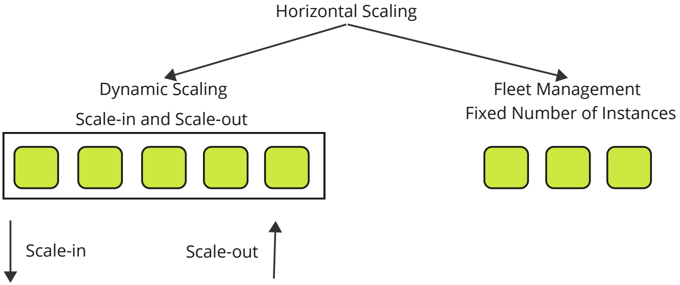
Vertical Scaling

t2.micro c5.xlarge

(need downtime)

Horizontal Scaling

(don't need downtime)



Horizontal Scaling

Dynamic Scaling

Scale-in and Scale-out

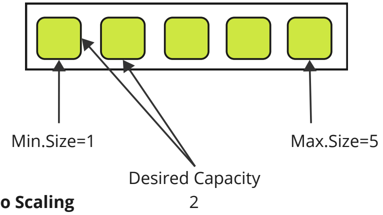
Scale-in

Scale-out

Fleet Management

Fixed Number of Instances

Auto Scaling Group: Its a logical group of EC2 Instances participating in Auto Scaling



Min.Size=1

Max.Size=5

Desired Capacity
2

Benefits of Auto Scaling

Improved Fault Tolerance

Important: Auto scaling, it either terminates the instance or it launches the instance it never stops or starts the instance.

Improved Availability

Improved Cost Management

Component of Auto Scaling Group

1 **Auto Scaling Group**: Min & Max Size and Desired Capacity, and condition to increase or decrease number of instances on the basis of Metrics.

2 **Launch Configuration:**

We specify **AMI ID**, Instance Type, Key Pair, Security Group, Block Storage (EBS), User Data (to configure instance)

AMI ID

Target Tracking Scaling Policy

Avg. CPU Utilization

if Avg CPU Utilization>=35%, an addition instance will join the group.
and it can go up to max size of auto scaling group.

Avg CPU Utilization of multiple instance

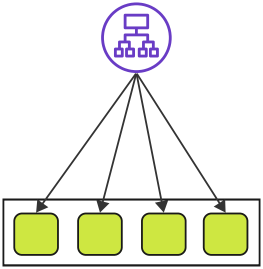
50% Instance1 40% Instance2

Avg CPU Utilization = 45%

50% Instance1 Avg CPU Utilization = 50%

99%/35%= 2.8 Approx
3 More instance will
join the group

Integration of Auto Scaling with ELB



Create ELB

condition to trigger action is CPU Util>=35%

Current usage is 95%

Number of instances expected = 95/35

=Approx 3 More instances

CICD/APIGW/Kinesis/SystemManager/Security&Encryption/StepFuntion/Appsyn

AWS SDK

AWS SNS -- **Simple Notification Server**

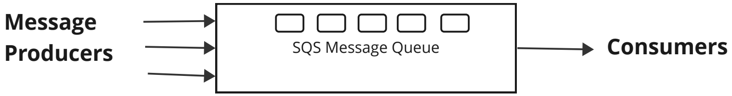
SNS is push based system

and it pushes notifications to Google, FireOS, Windows, Apple and Anroid devices

You can push messages to SMS Text message system, emails, SQS queues.

SNS can trigger Lambda Functions also.

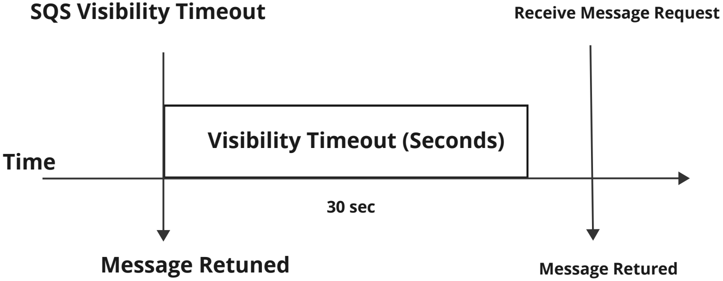
AWS SQS -- Simple Queue Service



it can contain upto 256 KB in any format json, xml etc.

Standard Queues (Default)
FIFO Queues

SQS Visibility Timeout



Default Visibility Timeout=30sec
Visibility Timeout can be changed
Max Visibility Timeout can be 12 Hours
But messages can be kept in queues from 1 min to 14 days, and default retention period is 4 days.

Assignment #8 **15 Min**

Send Messages Between Distributed Applications with Amazon Simple Queue Service (SQS)

https://aws.amazon.com/getting-started/hands-on/send-messages-distributed-applications/?ref=gsrchandson

AWS CloudWatch

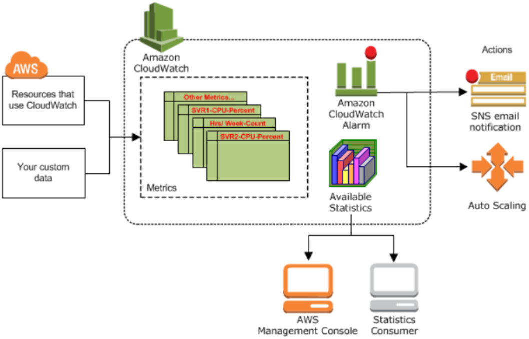
Resource/Service Monitoring Service

Basic Monitoring

Its Free, Polls data in every 5 min, works with limited metrics, Ingest only 5 GB data

Detailed Monitoring

Chargeable, data poll time to monitor service can be reduced < 5min



AWS CloudWatch measures the performance on the basis of Metrics.

AWS EC2 Metrics

CPUUtilization
DiskReadOps
NetworkIn
NetworkOut
NetworkPacketIn etc.

AWS EBS Metrics

EBSReadOps
EBSWriteOps
EBSReadBytes
EBSWriteBytes etc.

AWS Billing and Cost Metrics

Estimated Charges
ServiceName
LinkedAccount etc.

Benefits of CloudWatch Monitoring

- **AWS Resource Monitoring**
- **EC2 Instance Monitoring**
- **Setting Alarm and Actions**
- **Viewing Graphical Representation**
- **Viewing Statistics**
- **Monitoring and Storing Logs**

Task

Create 2 EC2 Instances, and generate some traffic on these instances by sending packets continuously.

```
c:/> ping <ip_address_Instance> -l 1024 -t
```

Stay for 5 minutes to have some output in CloudWatch and Create CloudWatch DashBoard and create multiple widgets.

Task

Alarms:

Create an alarm for an EC2 instance's CPU Utilization, if CPU Utilization >= 70% one time in 5 minutes, in that case the Instance must be stopped automatically.

Task

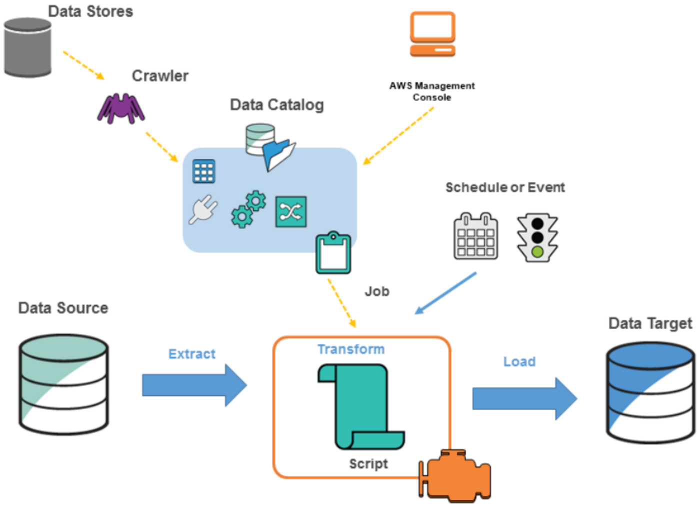
Using CloudWatch to create a Bill Alarm by enabling estimated charges for AWS Account.

Create a Billing Alarm to Monitor Your Estimated AWS Account Charges.

AWS Glue

(Glue is ETL(Extract, Transform and Load) Service), and it is Serverless

AWS Glue is a serverless data integration service that makes it easy for analytics users to discover, prepare, move, and integrate data from multiple sources. You can use it for analytics, machine learning, and application development. It also includes additional productivity and data ops tooling for authoring, running jobs, and implementing business workflows.
With AWS Glue, you can discover and connect to more than 70 diverse data sources and manage your data in a centralized data catalog. You can visually create, run, and monitor extract, transform, and load (ETL) pipelines to load data into your data lakes. Also, you can immediately search and query cataloged data using Amazon Athena, Amazon EMR, and Amazon Redshift Spectrum.
AWS Glue consolidates major data integration capabilities into a single service. These include data discovery, modern ETL, cleansing, transforming, and centralized cataloging. It's also serverless, which means there's no infrastructure to manage. With flexible support for all workloads like ETL, ELT, and streaming in one service, AWS Glue supports users across various workloads and types of users.
Also, AWS Glue makes it easy to integrate data across your architecture. It integrates with AWS analytics services and Amazon S3 data lakes. AWS Glue has integration interfaces and job-authoring tools that are easy to use for all users, from developers to business users, with tailored solutions for varied technical skill sets.
With the ability to scale on demand, AWS Glue helps you focus on high-value activities that maximize the value of your data. It scales for any data size, and supports all data types and schema variances. To increase agility and optimize costs, AWS Glue provides built-in high availability and pay-as-you-go billing.



Glue Data Catalog

Each AWS account has one AWS Glue Data Catalog per AWS Region. Each Data Catalog is a highly scalable collection of tables organized into databases. A table is metadata representation of a collection of structured or semi-structured data stored in sources such as Amazon RDS, Apache Hadoop Distributed File System, Amazon OpenSearch Service, and others. The AWS Glue Data Catalog provides a uniform repository where disparate systems can store and find metadata to keep track of data in data silos. You can then use the metadata to query and transform that data in a consistent manner across a wide variety of applications.

Glue Crawler

An AWS Glue crawler connects to a data store, progresses through a prioritized list of classifiers to extract the schema of your data and other statistics, and then populates the Glue Data Catalog with this metadata. Crawlers can run periodically to detect the availability of new data as well as changes to existing data, including table definition changes. Crawlers automatically add new tables, new partitions to existing table, and new versions of table definitions. You can customize Glue crawlers to classify your own file types.

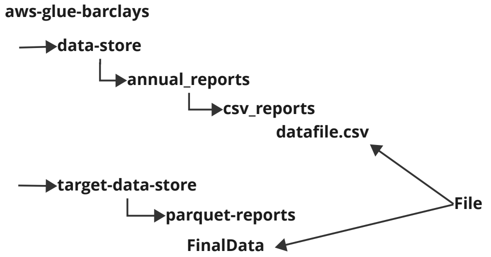
2

Assignment: Submit

Steps to work in Glue

1 Create IAM Role for Glue with AWSGlueServiceRole & S3FullAccess Permission

2 Create Directory Structure ins S3



aws-glue-barclays

data-store

annual_reports

csv_reports

datafile.csv

target-data-store

parquet-reports

FinalData

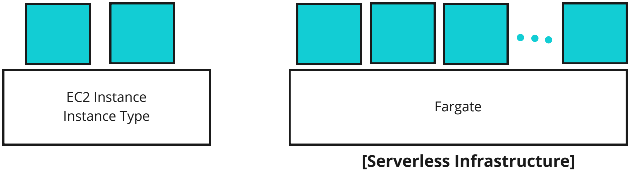
File

3 Glue: Configure DataBase information

Table Information

Manage Glue Studio to declare Jobs

ECS (Elastic Container Service)



3

Assignment: Submit

Creating a container image for use on Amazon ECS

https://docs.aws.amazon.com/AmazonECS/latest/userguide/create-container-image.html

CI/CD Pipeline
CodePipeline

Continuous Delivery and Continuous Integration

Code Pipeline does:

It establishes a consistent release process

Use to automate application release process

It speeds up delivery while improving quality

You can view Pipeline history, progress at a glance, and can use your favorite tools.

Security and Encryption

how to secure infrastructure or applications

Security Group, NACL, VPN

WAF Service, Guard Duty,
Firewall Manager, Inspector, Shield
GWLB with third party appliances such as IDS/IPS

Encryption : KMS (Key Management Service)