Chapter 3.

classmate
Date 04/04/2023.
Page

EC2:

Amazon Elastic Compute Cloud (EC2) is a web service that provides a scalable computing capacity in the AWS cloud.

With Amazon EC2 developers can quickly launch instances with a variety of Operating Systems & pre-installed applications as well as customize the computing environment to suite their specific needs. They can also scale their infrastructure up or down as needed, paying only for what they used.

Amazon EC2 is a fundamental building block of AWS, & is widely used by companies of all sizes, from startups to enterpriceses to power their web applications, backend services & big data analytics workloads.

1. List out the Configuration options available for EC2:

Amazon EC2 provides a wide range of configuration options for instances including:

1 Instance Type:

EC2 offers variety of instance types optimized for diff workloads, such as compute-intensive, memory-intensive, storage optimized, GPU instances for ML & video rendering & many more.

2 OS:

EC2 supports a range of Operating Systems, including Amazon Linux, Ubuntu, Windows Server & more.

3 Storage:

EC2 provides diff storage options such as Elastic Block Store (EBS), Instance store & Amazon Elastic File System (EFS) each optimized for diff use-cases.

4 Network & Network Card:

EC2 allows users to configure Virtual Private Clouds (VPCs) & subnets, set up security groups & assign public or private IP addresses.

5 Availability:

EC2 enables users to configure instances to run in multiple Availability Zones for increased availability & Fault Tolerance.

6 Auto Scaling:

EC2 allows users to automate the scalling of instances Up or Down based on specific condition, such as traffic & CPU usage.

8. Security.

7 Monitoring: EC2 provides a monitoring tools such as Amazon CloudWatch that enable users to collect metrics on their instances

2. Explain EC2 users data & Give one sample script:

EC2 users data is a feature of Amazon EC2 that allows users to pass scripts or data to an instance when it launches.

It is possible to bootstrap our instances using an EC2 user data script. Bootstrapping means launching commands when a machine starts.

this script is only run once at the the time the instance first start or boots.

EC2 user data is used to automate boot task such as

1) Installing Updates
2) Installing Software
3) Downloading Common files from the internet.
4) Anything we can think of.

the EC2 users Data Script runs with the root user.

EC2 → instances → Launch instance → Addition details → User data (optional) Type this

Sample Script:

```
#!/bin/bash
yum update -y
yum install -y httpd
systemctl start httpd.service
systemctl enable httpd.service
echo "Hello World from $(hostname -f)" > /var/www/html/index.html
```

this script first updates the instances packages using yum package manager, then installs the Apache HTTP Server using 'yum install'. It then starts the apache server & configures it to start automatically on boot. Finally it creates an 'index.html' file in the Apache document root directory that displays a 'Hello World' message.

3. Explain Diff Types of EC2 Instance :

Amazon EC2 provides a wide range of instances types, optimized for diff workloads, such as compute, memory, storage & networking.

Diff Types of EC2 instance :

1. General Purpose Instances
2. Compute - Optimized Instances
3. Memory - Optimized Instances
4. Storage - optimized Instances
5. GPU Instances.

1. General Purpose Instances :

These instances are well suited for a variety of workloads, such as web severs, small to medium databases, code repositories, development & test environments.

Examples: t2, t3 & m5 instance families

this instances provide a balance blw compute, memory & Networking resources & can be used for a variety of diverse workloads.

2. Compute - Optimized

These instances are designed for the compute-intensive tasks or workloads that require high performance processors, such as High-performance computing, scientific modeling & ML, High-Performance web severs, batch processing, and dedicated gaming severs

Example: c5, c5n, & z1d instance families

3. Memory Optimized:

These instances are designed & optimized for memory-intensive workloads such as In-memory databases, Real-Time big data processing & High performance computing.

These instances are designed to deliver fast-performance for workloads that processes large data set in memory.

Example: r5, x1, & z1d instance families.

4. Storage Optimized:

Storage Optimized instances are designed for workloads that require high, sequential Read & Write access to very large data sets on local storage.

These instances are optimized for storage-intensive workloads, such as NoSQL databases, Data Warehousing and Hadoop clusters.

Example: i3, d2 and h1 instance families.

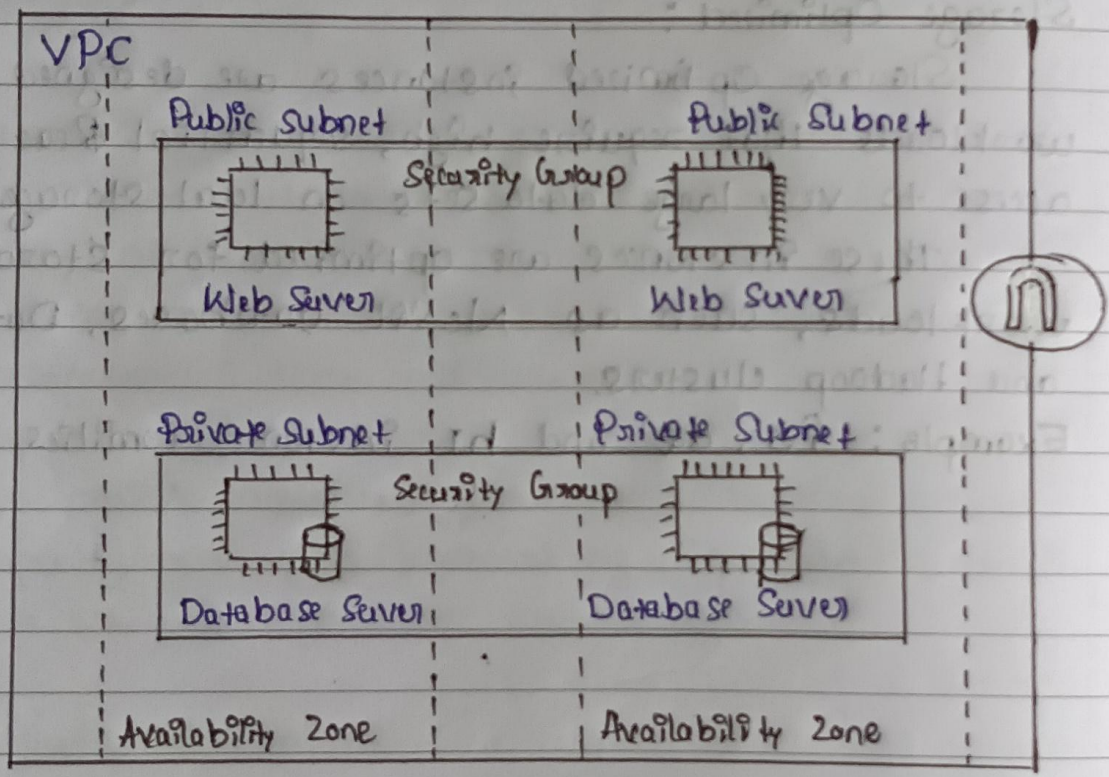
4. What are Security Groups? Explain How Security Groups Works? with a Diagram.

Security Groups:

Security Groups are the fundamental of Network Security in AWS.

Security Groups are Virtual Firewalls that control inbound & outbound traffic to & from an Amazon EC2 instance.

They act as a barrier blw the instance & the internet by allowing only authorized traffic to pass through.

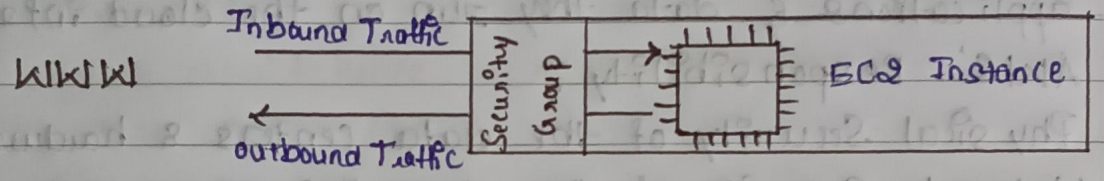


As shown in the diagram a Security Group is associated with one or more instances running in an Amazon VPC. Each Security Group consist of set of rules that specify the allowed traffic to & from the associated instances.

When a request is made to an instance, then the Security Group checks the rules to determine whether the request is allowed. If the request matches one of the allowed rules, then it is permitted to pass through the Security Group to the instance.

If the request does not match any of the allowed rules, then it is rejected, & the instance does not receive the traffic.

Security Group provide a simple & effective way to control network traffic to & from the instances in an Amazon VPC, & they can be used to implement fine-grained access control & enforce compliance requirements.



Security Group only contain allow rules & They Regulate

* Access to Ports
* Authorize IP ranges - IPv4 & IPv6
* Control of Inbound network i.e from other to the instance
* Control of Outbound network i.e from the instance to other.

7. Shared Responsibility Model for EC2:

The shared responsibility model for Amazon EC2 is a security model that outlines the division of responsibilities b/w AWS & its customers.

Under this model AWS is responsible for the security of the cloud infrastructure while customers are responsible for the security of the applications & data they run on the cloud infrastructure.

1. AWS' Responsibility:
   1 Physical Security of the data centers & hardware infrastruc
   2 Network Security of the cloud infrastructure
   3 Availability of EC2 instances & underlying infrastructure
   4 Obeying or Confirm with industry standards & regulation
   5 Isolation on physical hosts
   6 Replacing Faulty Hardware.

2. Customer Responsibilities:
   1 Security Groups Rules
   2 Security Configurations of EC2 instances including OS patches & updates.
   3 Data Security: Including encryption of data
   4 Network Security: including Firewall Configuration & Access control
   5 Application Security: including secure coding practice & vulnerability management.
   6 Managing IAM Roles assigned to EC2 & IAM user access management.
   7 Managing software & utilities installed on the EC2 instan
   8 Compliance or obeying with applicable laws & regulations

benefits of AWS Lambda.

8. Explain the tums i) Serverless ii) AWS Lambda & Give the

i) Serverless:

In AWS Serverless computing refers to a model of computing where the cloud provider manages the infrastructure and automatically provides & scales the resources as needed, without the need for the user to manage or maintain servers.

Serverless is a new paradigm in which the developers don't have to manage servers anymore. They just deploy code & functions

Serverless does not mean that there are no servers instead it means we just don't manage or see them.

ii) AWS Lambda:

AWS Lambda is a serverless compute service provided by AWS that allows users to run code without provisioning or managing servers.

With AWS Lambda, users only pay for the actual compute time they consume, rather than paying for a fixed amount of server capacity.

Lambda runs our code on highly-availability compute infrastructure & performs all of the administration of the compute resources including server & OS maintainance, capacity provisioning, automatic scalling & logging.

Benefits of AWS Lambda:

1 No Server Managment
2 Pay for Usage
3 Highly Scalable
4 Multiple Language Support
5 Integrated with other AWS Services
6 Easy Pricing
7 Easy to Deploy & Test.

1 No Server Management:

With AWS Lambda users don't need to worry about server management, scaling or maintainance. AWS automatically handles all the infrastructure & scaling for the user.

2. Pay for Usage:

AWS Lambda users only pay for the compute time that their code actually uses.

3 Highly Scalable:

AWS Lambda automatically scales up or down to handle the workload. So users don't need to worry about capacity planning or infrastructure management.

4 Multiple Language Support:

AWS Lambda supports variety of programming languages including Node.js, Java, Python, C#, GroLang, Ruby, etc.

5 Integrated with other AWS Services:

AWS Lambda integrates seamlessly with other AWS Services such as Amazon S3, Amazon DynamoDB, & Amazon API gateway.

6 Easy Pricing:

Pay per request & Compute time. Free tier of 1,00,00,00 AWS Lambda request & 4,00,000 of Compute time.

7 Easy to Deploy & Test:

AWS Lambda make it easy to deploy & test code changes, & provides tools for versioning & rollbacks.

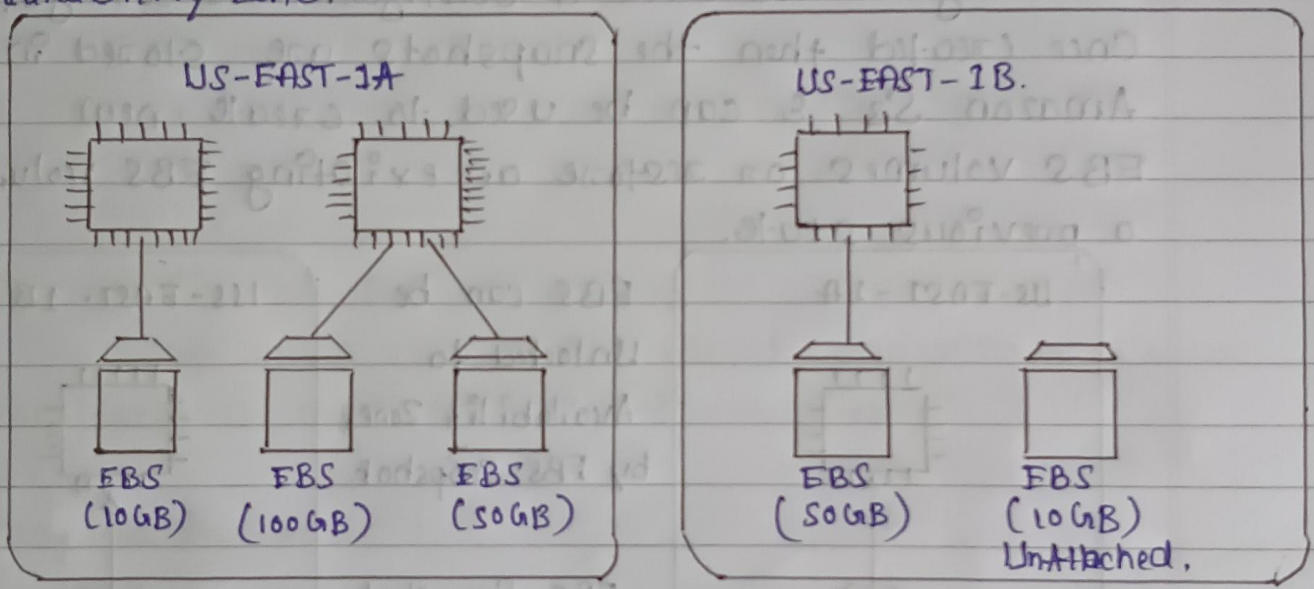
9. Explain EBS Volume in AWS with Diagram?

EBS Volume : Network Drive not a Physical Drive

EBS is a block storage service provided by AWS that allows users to create and attach persistent block storage volumes to their EC2 instances.

Elastic Block Store (EBS) volumes are highly available, durable & can easily be backed up and restored.

We can attach multiple EBS volumes to a single instance. But the Volume & instance must be in the same Availability Zone.



Features :

1 Analogy : Think of them as a "Network USB Stick".
2 It is a Network Drive not a physical Drive.
3 It can be detached from an EC2 instance & attached to another one quickly.
4 It is locked to an Availability zone. So an EBS in US-EAST-1A cannot be attached to US-EAST-1B. To move a volume across we first need to snapshot it.

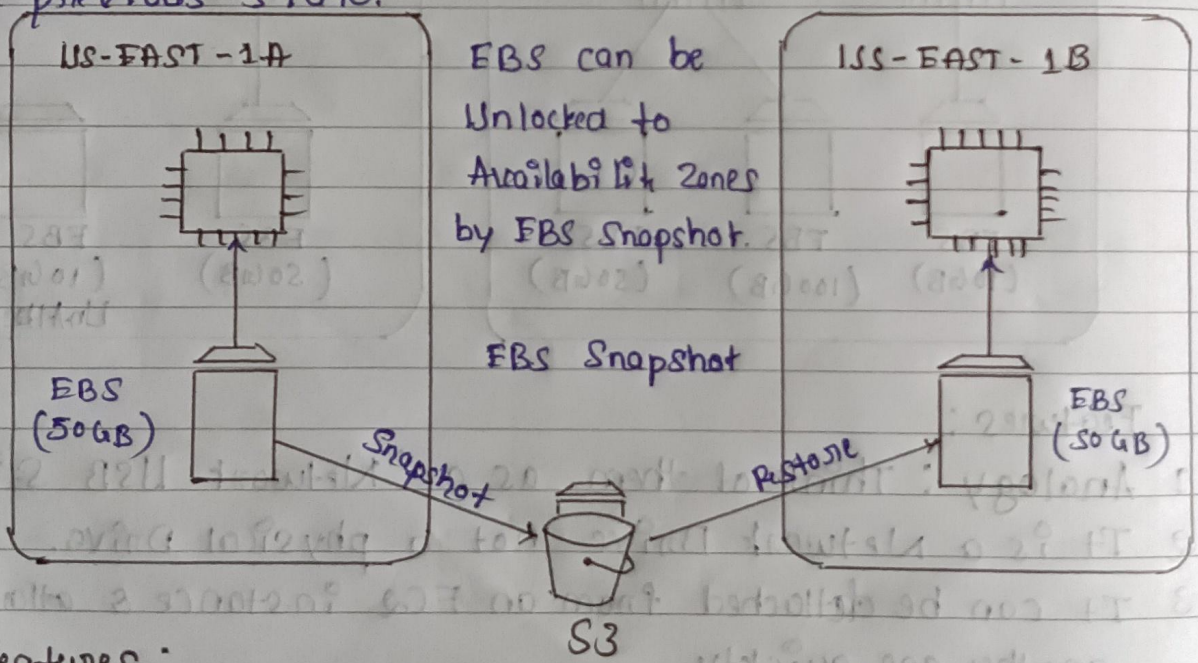
10 Explain EBS Snapshots with a diagram & list its feature?

EBS Snapshot:

EBS Snapshots are point-in-time copies of an EBS volume which can be used for backup, Disaster recovery, & to create a new EBS volumes.

So we can make a backup or take Snapshot of our EBS volume at a point in time.

EBS snapshots capture the entire state of an EBS volume at a particular point in time, including all its data & configurations settings. Once created then the snapshots are stored in Amazon S3 & can be used to create new EBS volumes or restore an existing EBS volume to a previous state.



Features:

1 Point-in-time Backup:

EBS snapshots capture a point-in-time copy of an EBS volume, including all its data & config settings.

2 Incremental Backups:

EBS Snapshots are incremental backups, which means that only the blocks that have changed since the last snapshots are saved. This make the backup process faster & more efficient.

3 Automated backups:

EBS snapshots can be automated using Amazon CloudWatch Events or AWS Lambda which means that we can schedule backups to occur at regular intervals.

4. Low Cost:

EBS snapshots are low-cost backup solution, as we pay only for the data that is stored in the snapshot.

5 EBS Snapshot Archive:

Move the snapshot to Archieve tier that is 75% cheaper.

6 Recycle Bin for EBS Snapshots:

Setup rules to retain deleted snapshots. So we can recover them after accidental deletion,

7 Easy to Restore:

EBS snapshots can be easily restored to a new EBS volume or an existing one.

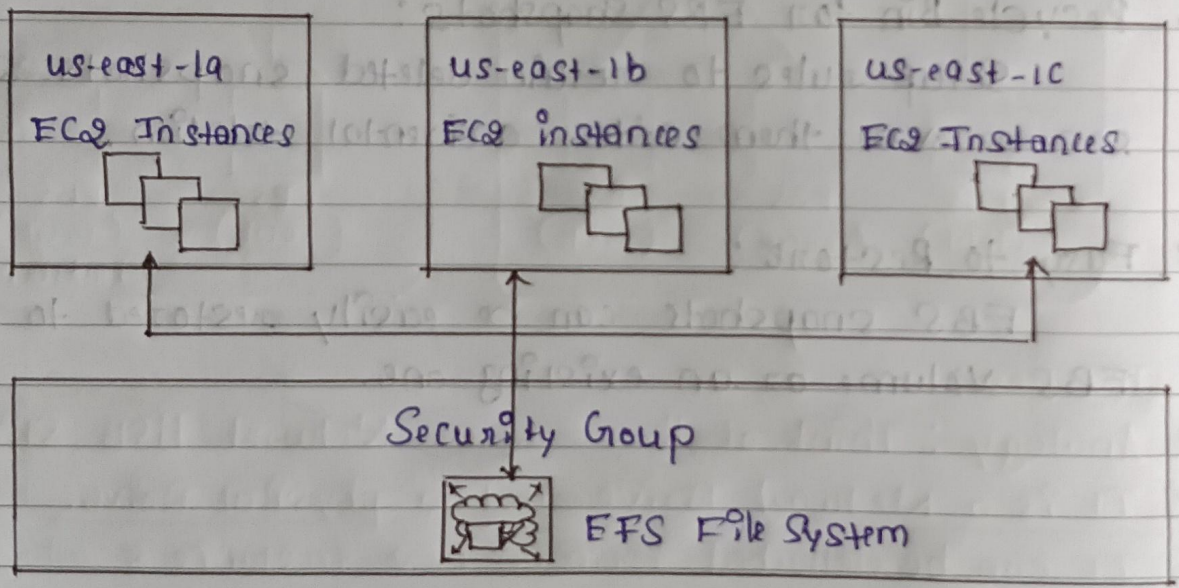
11 Explain EFS in AWS with Diagram. Give the diff b/w EBS & EFS

EFS:

Elastic File System provides serverless, scalable, & fully elastic file storage service so that we can share file data without managing storage capacity & performance.

It is designed to provide scalable, highly available & durable file storage for use with Amazon EC2 instances, Amazon ECS & EKS

With EFS we can create file systems that can be accessed by multiple EC2 instances simultaneously. this makes it ideal for use cases where we need to share files b/w multiple instances. or when we need a file system that can scale as our application grows



Amazon EFS has simple web service interface, so we can create & configure file system quickly & easily. Amazon EFS supports the Network File System (NFS) version 4 protocol

With Amazon EFS we pay only for the storage used by our file system.

12. Shared Responsibility model for EC2 storage:

Similar to EC2 instances, EBS & EFS also follow a shared responsibility model where AWS & Customers have diff responsibilities for securing the data stored in these storage services.

1 AWS Responsibilities:

1 Physical Security of the data centers where EBS & EFS are hosted.
2 Network Security of the infrastructure that supports EBS & EFS
3 Security & maintainance of the storage system used by EBS & EFS
4 Patching & maintainance of OS used by EBS & EFS.
5 Replacing faulty Hardware
6 Ensuring their employees cannot access our data.

2. Customer Responsibilities:

1 Encryption of data stored in EBS & EFS
2 Managing the access to the data stored in EBS & EFS, including setting up security groups & managing user accounts.
3 Regularly patching & maintaining the OS & applications that use the data stored in EBS & EFS.
4 Monitoring the data stored in EBS & EFS for the security breaches
5 Backing up the data stored in EBS & EFS. (Snapshot)
6 Understanding the risk of using EC2 instance store.

Scalability:

Scalability refers to the ability of a system, application or infrastructure to handle an increasing workload or user demand without compromising on performance or availability.

AWS provides various scalable services that can be used to build a scalable applications or systems.

Types of Scalability in AWS:

1 Vertical Scalability:

Vertical Scalability in AWS, also known as **Scaling up**, refers to the ability of a system to handle increased workload or demand by adding more resources to a single instance

This can include increasing the amount of memory, CPU or storage on an instance.

Amazon EC2 instance can be vertically scalled by choosing a large instance size. For example: if our application runs on **t2.micro**, scalling this application vertically means running it on **t2.large**.

2 Horizontal Scalability:

Horizontal Scalability in AWS, also known as **Scaling out**, refers to the ability of a system to handle increased workload or demand by adding more instances to to the system.

This can include adding more web servers, application servers or database servers to a system.

Amazon ELB can be used to distribute traffic across multiple instances, ensuring that the workload is evenly distributed & that there is no single point failure.

It is easy to horizontally scale thanks the cloud offerings or providers such as AWS.

15. What is Load Balancing? Why use a Load Balancer?

Load Balancing:

Load Balancing refers to the process of distributing incomming traffic to multiple targets, such as EC2 instances, containers & IP addresses, in order to optimize resource utilization & improve application availability & scalability.

Load Balancers are servers that forward internet traffic to multiple EC2 instances or servers downstream.

Why use a Load Balancer:

Load Balancers are used to distribute incomming network traffic or workloads across multiple resources such as servers to achieve several benefits, including:

1 Improved Performance:

Load Balancing helps to improve the performance of a system by distributing traffic across multiple resources, which enables each resource to handle a smaller portion of the overall workload. This helps to prevent any single resource from becoming overloaded.

2. Improved Availability:

AWS Load balancers help improve the availability of our application by distributing traffic across multiple resources, such as Amazon EC2 instances, containers or IP addresses. this helps to ensure that our application remains available even if one or more resources become unavailable.

3 Scalability:

AWS Load Balancer allow us to easily scale our application by distributing traffic across multiple resources. this enables our app to handle a larger volume of traffic as need

Down Stream instances → Less load instances.

4 Automatic Health Checks:

AWS Load Balancer automatically perform health checks on our resources to ensure that they are available & responding properly. If a resource fails a health check, then the load balancer automatically stops sending traffic to that resource & redirects traffic to other healthy resources.

5 Simplified Architecture:

AWS Load balancer simplify our application-Architecture by providing a single point of entry for incoming traffic.

6 Security:

AWS Load balancers provide SSL / TLS offloading, which enables them to handle SSL / TLS encryption & decryption.

What is Elastic Load Balancer? Explain any 2 kinds of Load Balan with Diagram?

Elastic Load Balancer (ELB)

ELB is a managed service provided by AWS that automatically distributes incoming network traffic or workloads across multiple resources such as EC2 instances, containers or IP addresses to achieve better performance, scalability & availability of our applications.

ELB is a fully managed service, which means that AWS take care of the underlying infrastructure, scaling & maintainance of the load balancer, allowing us to focus on building & running our application.

Types of Load Balancers:

1. Application Load Balancer
2. Network Load Balancer
3. Gateway Load Balancer.

1 Application Load Balancer (ALB)

Application Load Balancer makes routing decisions at the application layer i.e HTTP/HTTPS. So ALBs are used to route HTTP/HTTPS i.e Layer 7 Traffic.

ALB is a Layer 7 Load balancer that can route traffic based on application-level content such as URL or cookie information.

ALB supports several advanced features such as content-based routing, path-based routing, host-based routing & redirection based on rules.

ALB also supports integrated certificate management for SSL/TLS encryption & decryption.

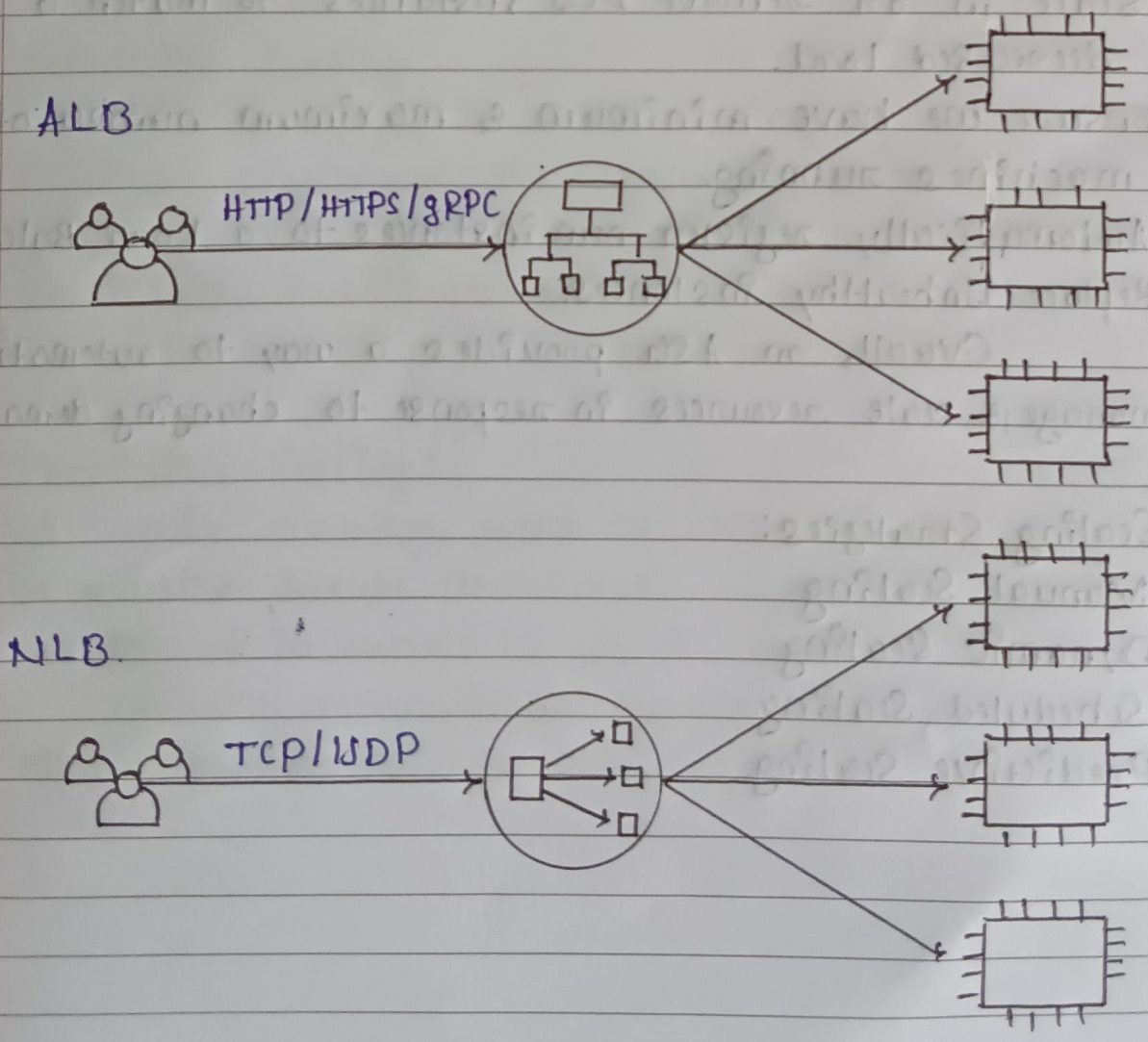
2 Network Load Balancer:

A Network Load Balancer makes routing decisions at the transport layer (TCP/UDP/SSL).

Network Load Balancer can handle millions of requests per second. & it is used to route TCP or Layer-4 traffic.

NLB is a Layer-4 load balancer that is designed to handle High-throughput traffic with low latency & high scalability.

NLB can be used to route traffic to targets across multiple VPCs or regions. NLB is particularly well-suited for use cases such as gaming, streaming & other latency-sensitive applications.



Auto Scaling Group :(ASG)

ASG is a service provided by AWS that enables automatic scaling of resources such as Amazon EC2 instances based on predefined rules.

An ASG contains collection of EC2 instances that are treated as a logical grouping for the purposes of automatic scaling & management.

Both maintaining numbu of instances in as ASG & automatic scaling are the core functionality of Amazon EC2 Auto Scaling service.

The Goal of an ASG is to:

1. Scale out or add EC2 instances to match an increased load.
2. Scale in or remove EC2 instances to match a decreased load.
3. Ensure we have minimum & maximum numbu of machines running.
4. Automatically register new instances to a Load Balancer.
5. Replace Unhealthy instances.

Overall, an ASG provides a way to automatically manage & scale resources in response to changing demand.

Scaling Strategies:

1. Manual Scaling
2. Dynamic Scaling
3. Scheduled Scaling
4. Predictive Scaling

1 Manual Scaling:

This is the most basic scaling strategy, which involves manually adjusting the number of instances in an ASG. This can be done using the AWS Management Console, CLI or API.

2 Dynamic Scaling:

This strategy involves scaling the number of instances in an ASG based on specific metrics such as CPU usage, network traffic or custom metrics.

Dynamic scaling is basically responding to changing demand

Ex: When CloudWatch Alarm triggered if CPU > 70%. then add 2 units if CPU < 30% then remove 1 unit

3 Scheduled Scaling:

This strategy involves setting up a schedule for scaling up or down the num of instances in ASG. This can be useful for applications that have predictable traffic pattern, such as those that experience increased traffic during bussiness hours.

4 Predictive Scaling:

This strategy uses Machine Learning algorithms to predict future demand & proactively scale the number of instances in an ASG.

It will automatically provide right number of EC2 instances in advance.

Chapter 5.

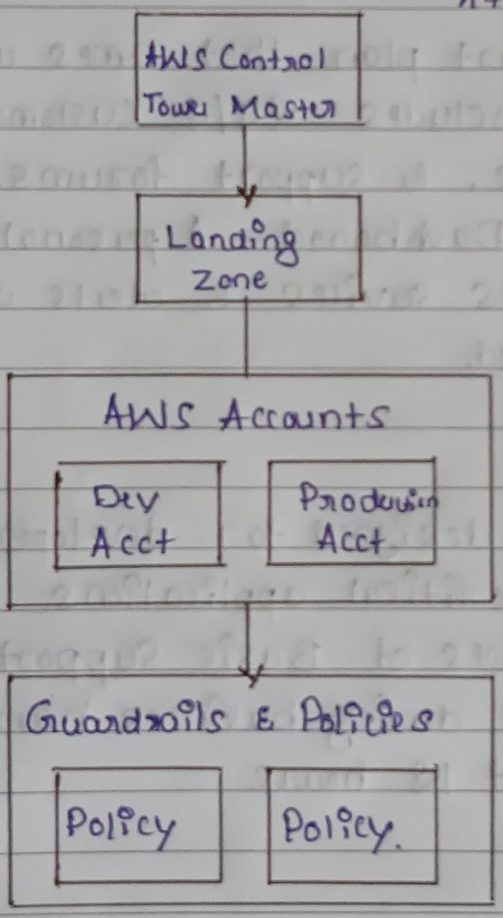
classmate Date 05/04/2023 Page

AWS Control Tower:

AWS Control Tower is a managed service that helps organizations to set up and govern a secure and compliant multi-account AWS environment, based on best practices.

It provides a central location for defining & enforcing AWS best practices across an organization's AWS accounts & automates the process of creating new accounts.

If we are an enterprice with a large number of applications & distributed teams, then cloud setup & governance can be complex & time consuming. That's where AWS Control Tower comes in.



At the top of the diagram we can see the AWS control Tower Master, which is the central Hub for AWS environment.

With Control Tower we create something called Landing zone which is a pre configured environment with foundational security & compliance controls such as IAM, logging & auditing.

Basically an organization being created by AWS Control Tower if we don't have an already.

AWS Accounts include multiple accounts which are created & managed automatically by AWS Control Tower.

The Guardrails & Policies are used to enforce rules & standards across the AWS environment. Guardrails are predefined rules, while policies are custom rules, that organization can define to enforce their own specific requirements.

2. Explain the diff types of AWS Support Plans?

AWS Support plan :

An AWS Support Plan is a service provided by AWS that offers technical support & guidance to AWS customers.

Types of Support Plans :

1 Basic Support Plan
2 Developer Support Plan
3 Business Support Plan
4 Enterprice Support Plan
5 Enteprice On-Ramp Support Plan

1 Basic Support Plan :

This is the free support plan that comes with every AWS account. It includes 24/7 customer servile, documentation, whitepapers, & support forums.

AWS Personal Health Dashboard - A personalized view of the health of AWS suvices & alerts when our resources are impaired.

2 Developer Support Plan :

this support plan is designed for developers & businesses running non-critical applications.

It includes all the features of Basic Support as well as email support during business hours & guaranteed response time of 12 hours.

3 Business Support Plan :

This support plan is designed for businesses running production workloads on AWS.

It includes all the features of developer supports, as well as 24/7 phone, chat & email support, a guaranteed response time of 1 hour & access to AWS trusted Advisor.

4. AWS Enterprice On-Ramp Support Plan:

This support plan is designed for businesses running production or business critical workloads.

* All of Business Support Plan +
* Access to a Pool of Technical Account Managers (TAM).

5 Enterprice Support Plan:

This is the most comprehensive support plan offered by AWS. It is designed for large-scale or mission critical deployments, & it includes all the features of Business support as well as dedicated TAM.

4. Explain the 4 Types of Pricing model in AWS

Pricing Model:

A Pricing model refers to the diff ways the customers can pay for AWS services.

Types:

1. On-Demand
2. Reserved Instances
3. Saving Plans
4. Spot Instance
5. Dedicated Hosts:

1 On-Demand:

This is a pay-as-you-go pricing model where customers pay for the resources they use on an hourly or per-second basis without any upfront-cost or long term commitments.

2 Reserved Instances:

This pricing model provides customers with significant discount for committing to use a specific instance type in a particular AWS region for a specified amount of time, usually 1 or 3 years.

The longer the commitment, the higher the discount. This pricing model is ideal for applications with steady-state or predictable workloads.

3 Saving Plans:

This pricing model provides customer with a discount on their AWS usage in exchange for committing to a certain amount of usage, either hourly or annually, for a one- or 3 year term.

Customer can choose b/w 2 Types of Saving Plans

1) Compute Saving Plans
2) EC2 instance saving plans.

4 Spot Instances:

This pricing model allows customers to bid on unused EC2 instances, allowing them to take advantage of unused capacity & save up to 90% as compared On-Demand pricing.

5 Dedicated Hosts:

This pricing model allows customers to Rent an entire physical Server that is dedicated to their use.

This model provides customers with complete control over the underlying hardware & can be good fit for applications with specific compliance or regulatory requirements.

4 Pricing Models:

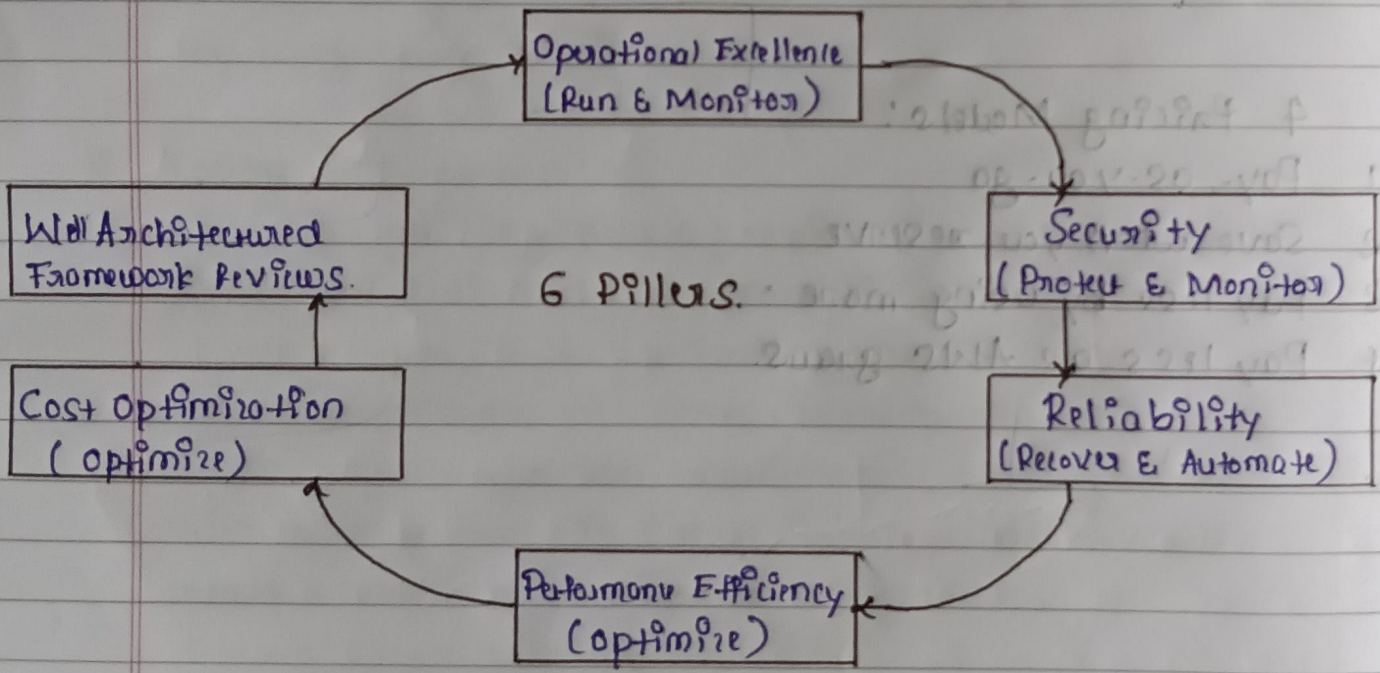
1 Pay-as-you-go
2 Save when you reserve
3 Pay less by using more
4 Pay less as AWS grows.

AWS Well Architectured Framework 6 Pillers:

The AWS Well Architectured Framework helps cloud Architetes to build the most secure, high performing strong & efficient infrastructure possible for their applications & workloads.

The AWS Well Architectured Framework based on 6 key Pillers:

1. Operational Excellence
2. Security
3. Reliability
4. Performance Efficiency
5. Cost optimization
6. Sustainability.



The diagram shows the 6 pillers of AWS Well-Architectured Framework in a circular arrangment. Each piller represents a key area that the architet & engineers should consider when designing & operating infrastructure on AWS.

The Arrows b/w the piller indicates that these pillers are interdependent & must be considered together to build a well architectured system or framework.

1. Operational Excellence:

This pillar focuses on running & monitoring systems, to deliver bussiness value & continuously improving processes & procedures.

It includes the ability to automate processes, perform routine maintainance & respond to events.

Design Principles:

1 Perform operations as code - Infrastructure as code
2 Annotate Documentation - Automate the creation of annotated document after every build.
3 Make frequent, small, reversible changes → So that in case of any failure we can reverse it.
4 Refine operation procedures frequently
5 Anticipate or expect failure
6 Learn from all opeational failure.

2. Security:

This pillar is concerned with protecting information, systems & assets while delivering business value through risk assesments, data protection & monitoring.

Security on the cloud is a big concern for everyone on the cloud. Infrastructure should be designed such that it serves complete data protection.

Design Principles:

1 Enable Traceability: Monitor, Alert & Audit actions.
2 Apply Security at all layers.: Utilize multiple security controls.
3 Protect data in transit & at rest
4 Keep People away from data: Eliminate the need for direct access or manual processing of data
5 Prepare for Security events: Create incident management & investigation policy.

3 Reliability:

This pillar focuses on the ability to recover from failures & disruptions, testing & validating the recovery proledures.

This pillar encompases the ability of a workload to perform its intended function correctly & consistently when it's expelled to.

Design Principles: 1) Automatically recover from failure
2) Test Recovery Procedures
3) Scale Horizontally.

4. Performance ~~Efficiency~~:

This Pillar concerns the efficient use of computing resources to meet business requirements & to maintain that efficiency as demond changes as well as minimizing waste.

Design Principles:

1) Go Global in minutes: Deploy you work load in multiple AWS Regions
2) Use Suveless Architecture: Remove the need for you to run & maintain physical seveus.

5. Cost Optimization:

This Pillar focuses on maximizing business value while minimizing costs through the use of appropriate resources as well as automation of cost management.

This Pillar includes the ability to run the systems to deliver bussiness value at the lowest price point.

Design Principles:

1) Implement Cloud Financial Management
2) Measure overall efficiency.
3) Analyze & attribute expenditure.

6. Sustainability:

this pillar focuses on minimizing the environmental impacts of running cloud workloads.

The discipline of sustainability addresses the long-term environmental, economic, & social impact of our business activities.

Design Principles:

1) Maximize Utilization
2) Use managed services
3) Understand your impact.

Purchasing options available for EC2 instances:

EC2 is a cloud computing service provided by AWS that allows customers to rent Virtual servers in the cloud.

EC2 offers several purchasing options for its instances, which refers to diff ways in which customers can pay for & use the virtual servers provided by the EC2.

Purchasing options:

1. On-Demand Instances
2. Reserved instances
3. Savings plans
4. Spot Instances
5. Dedicated Hosts
6. Dedicated instances
7. Capacity Reservations

1 On-Demand Instances:

This is the most flexible option & allows customers to pay for compute capacity by hour or second, with no longterm commitments.

* Payment for what you use
  1) Linux or Windows - billing per second after 1st minute
  2) All other os - billing per hour.
* Recommended for short-term & un-interrupted work loads.

2 Reserved instances:

This option provides customers with a significant discount upto 75% compared to on-demand instances, for committing to use the instances for one or 3 year term.

We can reserve a specific instance attributes like instance type, Region & OS.

We can buy & sell in the Reserved instance Marketplace.

3. Savings plans:

This is a newer plan by AWS & it helps us to reduce our compute cost.

So it provides customers with a way to save money on their compute usage by committing to a specific amount of usage over 1 or 3 year term.

Saving plan is far more flexible & provides significant cost savings as compared to On-Demand. AWS recommends using savings plans over Reserved Instance.

4 Spot Instances:

This option allows customers to bid on an unused EC2 instances & potentially save up to 90% compared to On-Demand instances. & it is the most cost-efficient one.

The price of spot instances fluctuates based on supply & demand. & customers can set max price they are willing to pay.

Instances that we can 'lose' at any point of time if our max price is less than the current spot price.

This is recommended for batch jobs, Big Data analysis, & workloads that are recoverable to failure. and this is not recommended for critical jobs or databases.

5. Dedicated Hosts:

This option allows customers to have an entire physical server dedicated to their use, providing them with more control over the underlying hardware & OS.

Customers can purchase dedicated hosts on-demand or as a reservation for up to 3 years.

This is the most expensive option & useful for companies that have strong compliance needs.

6. Dedicated Instances:

There is no purchasing option called, "Dedicated Instances" in AWS EC2 anymore. This option was deprecated in 2017 & replaced by "Dedicated Hosts"

This option is for the instances running on hardware that's dedicated to you.

It may share hardware with other instances in same account.

7 Capacity Reservations:

This option allows customers to reserve capacity for their instances in a Specific Availability Zone for any duration, upto 3 years.

With this plan customers can ensure that they have the compute capacity they need when they need it. without having to worry about availability of resources.

This option is available for both On-Demand Instances & reserved instances.