**MY BLOG AWS/DEVOPS:**

Delivering IT resources over the internet.

AWS:-

Secure

Fault tolerant.

Region>>availability zone>>data center

Why we should take the help of cloud while we are able to create the VM by using our own laptop by using the hypervisor software?

We can create a VM but what if we get a request to create 1000 vm’s let say we have written some shell script and also created 1000 vm’s and delivered to the end user but what about the maintenance? Like patching, updates, scaling..etc one system admin or onne server admin will not be sufficicient and what if we want to shut down all our servers for a week? In case of cloud we need not pay

**Use case of AWS lambda**

Lets say we are creating an app . for that we need some spot/place to put the function or code which is known to be a server and lets say EC2 in AWS perspective. So in this case we need an EC2 instance to be dedicatedly assigned for this task irrespective of using the app or not. Our computing power is wasting and also we need to manage this Ec2 like by scaling and updating the patches.here so to manage this AWS provided a service called LAMBDA:  
  
**USE CASE:**

Imagine you're developing a serverless image processing application where users can upload images, and the application automatically generates thumbnail versions of those images. Here's how AWS Lambda comes into play:

User uploads an image: When a user uploads an image through your application's frontend or API, the image is sent to an AWS S3 bucket for storage.

Trigger configuration: You configure the S3 bucket to trigger an event whenever a new image is uploaded.

Lambda function setup: You create an AWS Lambda function that will be responsible for generating the thumbnail. This function is written in a supported programming language like Python, Node.js, or Java.

Event-driven execution: Whenever a new image is uploaded to the S3 bucket, the trigger fires and invokes the Lambda function.

Thumbnail generation: The Lambda function receives information about the uploaded image and its location in the S3 bucket. It then retrieves the image, processes it, and generates a thumbnail version using the code you've written.

Scalability: AWS Lambda automatically scales your function to handle the incoming requests. It manages the infrastructure needed to run your code, ensuring high availability and fault tolerance.

Result and output: Once the thumbnail is generated, the Lambda function can store it back into another S3 bucket or perform other operations, such as sending a notification or updating a database.

Billing: AWS Lambda bills you based on the number of invocations and the duration of each function execution. You pay only for the actual compute time used during the processing of each image.

By leveraging AWS Lambda, you don't need to worry about provisioning or managing servers to handle the image processing workload. Lambda abstracts away the underlying infrastructure, allowing you to focus on writing the code and letting AWS handle the scaling and operational aspects. This way, you can build highly scalable and cost-effective applications without the burden of traditional server management.

If we host our app/code inn the EC2 instance we will be getting an ip where If we host our app/code in the lambda we don’t have any ip or we don’t even know where the instance or server is created

Boto

Who will decide whether we should use lambda or EC2?

Developer or solution architechs will decide . based on the application they will decide

The main purpose is **cost optimization,security,compliance.**

When we need to run few python scripts every day morning 10am if we use the ec2 for this use case daily we need to create daily and tear it down after our task.

But in the case of lambda we can configure cloudwatch to trigger the lambda everyday at 10am so that lambda will run the scripts and tear down automatically when the task is done.

Like main() function in the java and go we have lamda handler function in the serverless architecture we have lambda handler() function from where the execution starts.

By using the function url we can check the output of the lambda function that we have written

**COST OPTIMIZATION PROJECT:**

We moving to the cloud because we can reduce cost and remove over head

But without using the cloud efficiently we cant reduce the cost.

Example: creating resources and keeping them unused also known as the stale resources.

When ever we create an ec2 instance and attached with the EBS volumes and we take snapshots for backing up the data in EBS. After work is done we deleted EC2 and volumes but forgot to delete the snap shots which may incur unnecessary cost

So here we write a python code in the lambda function which fetch all the snapshots and delete them.

Check the guthub account of abhishek veeramalla.

Delete –lambda,volumes,snapshots.policy-costoptebs

Boto3 is a python module which allows you to talk with the aws resources.

**HTTP protocol:**

Developed in 1990 by European scientist just to transfer text which contains hyper links over the internet.

Uses URL ,REQUEST RESPONSE MODEL,Methods,status codes.

HTTP stands for Hypertext Transfer Protocol. It is an application layer protocol used for communication between clients (such as web browsers) and servers (such as web servers) over a network, typically the internet. HTTP is the foundation of data communication on the World Wide Web and enables the retrieval and display of web pages, images, videos, and other resources hosted on web servers.

The concept of HTTP is based on the client-server model, where clients (usually web browsers) make requests to servers, and servers respond to those requests by providing the requested data or resources.

Key characteristics of HTTP:

1. **Statelessness**: HTTP is stateless, which means that each request from the client to the server is treated independently, without any memory of previous requests. This allows for simple and scalable interactions between clients and servers.
2. **Request-Response Model**: HTTP operates using a request-response model. Clients send HTTP requests to servers, and servers respond with HTTP responses containing the requested data or resource.
3. **Uniform Resource Locators (URLs)**: HTTP uses URLs to identify resources on the web. A URL is a standardized address that specifies the location of a resource, such as a web page or an image.
4. **Methods**: HTTP defines several methods (also known as verbs) that indicate the action to be performed on the resource. Common methods include GET (retrieve data), POST (submit data to be processed), PUT (update a resource), DELETE (remove a resource), and more.
5. **Status Codes**: HTTP responses include status codes that indicate the result of the request. For example, a status code of 200 indicates a successful request, while 404 indicates that the requested resource was not found.

**New learnings:**

1.Increemental backup

2. IOPS – input output operations in the per second

3. EFS can share storage to 100’s of Ec2 instances.by giving high throughput and IOPS.across multiple aws regions

4.Fsx can support both windows and linux OS by using SMB(server message block) and NFS(Network file system) protocols.

Projects:-

1. Deploy word press website in the aws ec2
2. Serverless webapp

Database :

1. Data is highly secure as it resides in VPC. Encryption at rest is done through AWS KMS and encryption in transit is done by SSL
2. Backup, fault tolerance, read replica,patching,scaling.

EC2 USER SCRIPT:

List of commands which will automatically run right after switching on the EC2 instance by root user. These commands perform few tasks like installing updates and installing softwares, downloading the required files from the internet..etc

Bootstrapping means running the commands when the machine starts.

**Connecting the EC2 with mobaxterm**

**https://www.jenkins.io/doc/tutorials/tutorial-for-installing-jenkins-on-AWS/**

commands that has to be used:

**important linux commands after Ec2 is installed.**

sudo su-

yum update

yum install.

mkdir

cd

ls lrt

wget(copy link).

**VPC:**

A diagram of a network

Description automatically generated

Internet gateway>>NACL>>Route table>>NAT gateway(Only for PSUBNET)>>SG>>Resource.

Peering connection is only possible In the same region

Monitoring the vpc traffic—network flow logs.

NACL (SUBNET LEVEL Protection)  
SG(Resource/instance)

NAT gateway(Network address translation)Is used when we want our resources in the private subnet to be connected to the internet.(When resource in the private subnet starts requesting data from the external world the ip od the resource should communicate with outside internet so to restrict this the NAT gateway will change the ip to its ip and then communicate with the internet).

Pinging the instance means sending the ICMP echo request to the instance for checking if the instance is able to connect in that specific network or not. And also checking how much time it is taking to respond back with the packets.

**CLOUDFRONT CDN:**

In summary, Amazon CloudFront's edge locations act as proxy servers to deliver cached content and can also be configured to function as reverse proxy servers when necessary.

**Load Balancers:**

**Scenario: l**ets a I have a deployed a game app in my EC2 initaily 30 users started accessing and playing it and our EC2 server is working fine and no downtime or slowness of the app. Lateron our game became popular and 1000’s of users started using then we face slownless of the app. For this case we forecasting the huge traffic we will deploy our app in multiple EC2 servers. The incoming load will be distributed by the Load balancer to the EC2 servers. If we have 100 requests the LB will distribute 35,35,30 to 3 servers. This is known as **ROUND ROBIN** technique

Networking 7 layers traffic flow:

When ever we(client) go and search for something lets say **linkedin.com/mohanmuppalla** in the google(or your fav browser) the request packet will be traversed through 7 layers to reach the server.

7.Application layer(this layer will decide which protocol the request should follow or must be initiated with{http,https,smtp})

6. presentation layer(based on the input protocol It will provide security in this layer. If it is https it will incorporate SSL/TLS encryption.now everyone using TLS).

5.session layer(a session will be created because the client should understand few things about the request)

4.Transport layer(Here the packet will be splitted in to small small chunks .since it is very difficult for server to decrypt and read a huge packet to make server work easy the packet is splitted)

3.Network layer(the packet will be traveller to multiple routers while going to the server which will be taken care by network layer)

2. Datalink layer(now the packet will be moving to the datacenter which is managed by the datalink layer)

1.physical layer(this layer contains switches that are connected to servers via cables)

Which load balancer we should choose?  
if you want to perform the traffic load balancing at the **application layer we use ALB**

The ALB will intercept the request. Lets say we have a http request the ALB will check the request information that is header , path that request wants to go and server it is trying to hit. Based on these details the ALB will send the packet to the appropriate service in the server or it also distribute the requests to multiple servers.

If we want to distribute traffic at transport layer at L4 we use NLB. It is very fast compared to the ALB because ALB will intercept the HTTP packet and read it where the NLB doesn’t perform any operations on the packet it just distribute the load. The packets will be of tcp/udp kind.

NLB creates a Sticky session that means if the first request is send to server1 then rest of the request by the same ip will also be send to the same server where as the ALB distribute it to different servers.

For gaming apps,youtube apps which need very less latency we use NLB.

When we want use the VPN/Firewall kind of apps which need a high level security we front face the **Gateway load balancer**.

If we front face the NLB/ALB in place of GWLB they can’t provide that level of security what GWLB is providing.

A whiteboard with blue writing

Description automatically generated

A whiteboard with blue writing

Description automatically generated

## How Do Ports Work?

An [IP address](https://www.howtogeek.com/341307/how-do-ip-addresses-work/) specifies a computer — or other network device — on a network. When one device sends traffic to another, the IP address is used to [route](https://www.howtogeek.com/22/adding-a-tcpip-route-to-the-windows-routing-table/) that traffic to the appropriate place. Once the traffic reaches the right place, the device needs to know which app or service to send the traffic on to. That's where [ports](https://www.howtogeek.com/394735/how-do-i-open-a-port-on-windows-firewall/) come in.

If the IP address is akin to a street address on a piece of mail, the port is something like the name of the person at that residence who gets the mail. For the most part, you don't need to worry about ports. But once in a while, you might encounter an app that's set to listen for traffic on the same port that another app already has in use. In that case, you'll need to identify the app that already has that port in use.

There are a number of ways to tell what application has a port locked, but we're going to walk you through a couple of built-in ways that use the [Command Prompt](https://www.howtogeek.com/194041/how-to-open-the-command-prompt-as-administrator-in-windows-10/), [PowerShell](https://www.howtogeek.com/742916/how-to-open-windows-powershell-as-an-admin-in-windows-10/), or the [Terminal](https://www.howtogeek.com/831728/7-ways-to-open-windows-terminal-on-windows-11/), and then show you a great freeware application that makes it even easier. All these methods should work no matter which version of Windows you use.

It ensures that one resource won't get overloaded, and that the traffic is spread out.

ALB—handles the HTTPS traffic that is layer7  
 two way communication

Used in the microservices architechture

NLB—handles the TCP/UDP/TLS(layer 4 )  
 One way communication

Used in the traditional architecture  
 to fastrack our streaming like youtube,Netflix, amazon prime.  
classic load balancer – not using now

ELB- Elastic load balancer

ELB doesn’t support the FARGATE.

Ip white listing is a list of ip addresses which will authorized to move.

When we stop and restart the ec2 instance a new ip will be generated every time so in order to keep it static we use a elastic ip concept.

Client server model:

N number of clients using a centralized server.

**How auto scaling is done?**

Based uopon the input traffic the ASG will create a new instance or remove the instance   
how this is done?

By the using the launch config option we are doing this  
There is an option called launch config

We create a auto scaling groups and configure min and max number of instances.

MONOLITHIC AND MICROSERVICE?Tightly couples arch vs loosely coupled arch.

In monolithic service architecture all the components are tightly coupled. If any of the service fails working the entire architecture fails to work. So to overcome this microservices came in to picture.in microservices architecture if one service fails entire architecture will not fail. The components will communicate to each other with the SQS and SNS(Simple notification Service)

SNS is a notification system, which pushes messages to its subscribers.

SQS is a queuing system, and the receivers have to pull the messages to be processed and deleted from the queue.

SNS and SQS can works well together.

**How we configure SNS?**

Create a SNS topic and add a DL or individual or service which needs to receive the msg. while configuring add the triggering service

How we configure SQS?

Create a queue and the msgs will be sending to the queue and end user/team/service will pull the msges from the queue

Used for decoupling the services in an architeccture

**AWS COMPUTE SERVICES:**

If you want access to your underlying architecture like OS you go with the EC2

If you don’t bother about Underlying Arch or OS you just want to host your App in the AWS go with Serverless Lambda

If you want to run your application in the containerized format in the AWS then you choose any of the two available containerized services(ECS,EKS)(Containers run on top of the EC2 instances)

If you want to choose containerized services but don’t want to bother about the underlying OS go with the serverless service for orchestratration of Containers i.e AWS fargate.(no need to EC2 here)

**How to recover the pem file with creds that is lost?**

We can stop the current Ec2 intance and then we detach the EBS volume and we create a new EC2 intance and attach the EBS volume . Here we can play the EBS volume which is the root volume and extract the PEM file and again start our original EC2 instance.

Remove defaut route 0.0.0.0/0(this connects with the internet gateway) from the route table associated to that specific subnet in order to restrict the outbound access from the EC2 instance

Two tier application:- manages both front end and backend  
Three tier application:- manges front,back,and DB

**How VPC peering is done:**

We simply attach the VPC end point link of one VPC to the route table of another VPC and vice-versa

We can create a isolated environment in the VPC to store sensitive work loads or data it means we create a private subnet with no internet gateway attached.

If instances in the isolated env want to connect with the internet then we can use NAT instance of another subnet by configuring the route table in the isolated subnet.

We can create VPC end ponts for specific AWS services such as S3, dynamo DB and can associate them with the VPC. For efficient and secure communication

Bastion Host?

Jump server that is hosted in the public subnet which helps public traffic to connect with the instance in the private subnet.

**How to tell one EC2 in one region to talk to EC2 in another region**?

1. VPC peering must be done
2. IP of one EC2 should be added to the SG of the other EC2S

**AWS container services:**

With aws container services we need not set up a separate containerization software in our machine  
 >> we can easily deploy manage and scale our application.

Amazon ECR is a centralized registry for developers to Store manage and deploy docker container images

Lets say we have a application we want it to be broken on to several different core functions so that we can manage,deploy and scale them independently.

**AWS ELASTICE BEANSTALK:**

It is used for creating EC2 based environments. We need not explicitly build networking and create code and ec2. Once we have all our requirements elastic beanstalk will take care of the rest part of creating the architecture.

It is PAAS service where we directly upload our application code which we have build and them just run it. Environment will be taken care by the AWS

A screenshot of a computer

Description automatically generated

A diagram of a computer program

Description automatically generated with medium confidence

AWS-CLOUD FORMATION:

It is a fully managed service ,Create delete and update resources in stacks

A screenshot of a computer

Description automatically generated

**VPC: virtual private cloud**

Virtual private cloud: a private network designed by us to keep everything private

To make the resource which are in VPC public facing we have to use various gateway

**What is the size of VPC and how it is determined?**

We do it by using ip address range. Whenever a devops engineer creates a VPC AWS asks what is the size of the VPC.. If you mention for example the ip address range is 172.16.0.0/16 then AWS will allocate you 255\*255 i.e 65536 ip addresses for our VPC it means we can deploy or use around 65536 instances inside the VPC.

Now the devops engineer will divide the ip addresses in to some sub parts i.e subnetworks

The NAT gateway will mask the ip of the resource in the private subnet either with the ip of load balancer(SNAT) or with the ip of router(NAT)

Public subnet is the one which any user will connect inside the VPC

How a resource inside a VPC will connect to the internet?  
Through Internet gateway

What is NACL?

Restrict the access at subnet level/Automation of SG

Virtual private gateway(only approved traffic will be allowed)

To make the connection from the public user to the VPC we suggest user to create a VPN(A tunnel which allows only approved traffic)

Although we have a VPN we also have a small doubt that our traffic is moving in the public internet world and also have a chance of data breach to prevent this we have launched a service called AWS DIRECT CONNECT(A dedicated Fiber channel connected from User to the AWS data center)

Path when a packet coming from the public internet to VPC

Internet gateway>>NAT Gateway>>Router>>NACL>>SG>>Resoure

NACL are stateless(They donot remember what traffic is gng through it)

SG(Stateful)

The traffic after passing the internet gateway and load balancer in the pubic subnet it doesn’t know the next steps to go so we create and configure the route tables and provide path for the traffic to reach its destination.

AWS tries to provide security in every possible manner:  
App security

Networking haedening

User identiy

Authentication and Authorization

Ddos Prevention.

Data integrity

Encryption.

**ROUTE 53:**

Why R53?

When ever you create any service like load balancer or ec3 then AWS will assign a ip for that specific load balancer

We cannot remember the ip address which is numeric  
whenever we restart our system the ip address will be changed

Domain name as a service

Mapping the ip address to the Domain name

Route 53 will resolve your DNS to the ip address

public subnet private subnet

**Services provided by the R53:**

Domain Registration  
Hosted Zones  
health checks

thr DNS resolves a

Registering the domain

Routing the input taffic to the nearest DT

Types of Routing

**STORAGE AND DATA BASE:**

Storage:

For EC2

Whenever we create an Ec2 instance we must attach some Storage for it and the app inside it must access the storage,cpu..etc

In AWS we have two Storage options to attach with the Ec2

1. Instance store(Not preferable)  
   Because whenever we restart the EC2 it go and finds a new instance
2. EBS(Preferable)  
   Data is stored as Blocks

First we create the EBS volume and then attach it to the EC2

Incremental backup is done using snapshots  
If we stop/terminate the instance the data in the EBS will remain safe.

**AWS S3**why S3?

>S3 allows you to create buckets in which you can store anything(media,jpg,csv,excell)  
>S3 is a Global Service Anything stored in the S3 bucket can be Globally Accesible. Via HTTP protocol (So need not bother about the Availabillity)(while creating bucket it will ask to select region …? Buket is created in a specific region but data in the bucket is globally accessible [so as per our latency convenience we create buckets in the nearest regions])  
>Can store any type of file  
>Store data in the form of objects and can provide them set of permssions  
>different types of storage based on how frequently we access it  
>static website hosting  
>Can be integrated with the many services for trigger  
>server side encryption of data stored  
>4$ per year   
>Maximum size of file we can store in S3 in 5TB  
>versioning:- when we update the existing file and re upload it the new file will not override the existing one but it store the older version of it as well. We can also retrieve the deleted bucket.

What is 11 9’s and why it is called ?

It is highly reliable around 99.999999999%. let say 1 billion(100 crores) of objects stored for 100 years in an S3 bucket. There is a chance of just deleting 1 object.

**AWS CLI :**Directly from UI or the aws console it is very hectic to create n number of resources.

CLI is a python utility. We pass statements

AWS CLI is a layer between User and AWS API

It converts the aws commands that we are passing in to the api and makes a call to aws

**Why we are using AWS CLI when we have CFT and terraform?**

CLI is best In few use cases. Like listing all the S3 buckets in the specific aws account we can simply give a command **aws s3 ls .** but to create a bunch of resources we have to give complex commands so in this case terraform and CFT are better

**Cloud Formation Templates:**

It is a IAC tool

It takes the json/yaml script as inputs to create infra  
it supports versioning(can track changes on the stacks that were made in past  
it is declarative(what you see is what you have)

As a devops engineer we mostly use yaml over json

Yaml is easy and use indendaation to separate the code lines where the json uses curly braces and brackets.

We can either directly open the cft in the AWS UI or we can use CLI

Detect drift:- when we click this option in CFT it shows you all the changes made on the infra in the UI created through CFT

**IMPLEMENTING CI/CD ON AWS:**

For CI/CD solution we usually use open source tools like Github actions, gitlab,Jenkins,argocd ..etc

Github>>>Jenkins>>>docker/sonarcube/maven>>

To keep our projects secure we usually use private git repositories or download the GITLAB in our own servers.

So in this case we might face issues in managing and scaling this.

So AWS came up with new services for this task.

**AWS CODECOMMIT:** Alternative for github and used for storing the projects securely

**CODE PIPELINE:**

We write a webhook for implementing continuous integration and invoking continuous delivery

What steps involved in the CI?

Checkout the code>>Build and UT>>code scan>>image build>>image scan>>image push

These are not standard steps however steps might vary based on the application

After the implementation of CI is done the invoking of CD can be done by Ansible/shell or now a days all use Argo CD 3D

AWS codepipeline works as an orchestrator and trigger the aws code build

Steps:

All the steps will be done in the build spec file which is written in the yaml

A computer screen shot of a cartoon character

Description automatically generated

A screenshot of a computer

Description automatically generated

**CODE DEPLOY:**

Deploy the application in the EC2 instance.

Either in Jenkins or whatever ci/cd we need to install an agent in Jenkins agent is used for connecting master and the slave node

Similar in the code deploy also we must install agent on the EC2

Steps:

* Create application in the code deploy
* Create Ec2
* Create Agent on EC2
* Give permissions that are required for Code deploy to talk with EC2
* In the deployment group name we give EC2 name as deployment group

**AWS CLOUD WATCH:**

It is a gatekeeper which watches the activities happening on the aws services.

Tasks:

Monitoring

Real time metrics(you can ask hey can you give me the utilization of CPU)

Alarms(Hey CW send me alarm if my cpu utilization metric exceeds 60%)

Loginsights(which service is trying to access our service)

Custom metrics(our own metric which we create)

Cost Optimization(we can delete unnecessary resources that are created and not using)

Scaling(CW will say hey your CPU has reaches an utilization of 80% by knowing this we will increase the CPU/Instances through auto scaling groups)

Alerting

Reporting

Real time task: create a log group with your service

In the log insights we can write queries and get insights on our services

**AWS CLOUDFRONT:**

AWS managed service which provide solution for the CDN

**Lets say your are living in usa your friend in india upoads a photo/video in** Instagram. Usually the photo/video will be stored in the centralized location .this centralized location lets say it is in india so when you try to access that photo you need to cross n number of routers/hopes to access that photo. To prevent this AWS found a service called CDN. By using this service AWS will store the photo/video copies in the edge locations of all the regions so whenever we try to check the photo instead of going to the centralized server which is located in india . AWS will say hey I have a copy of it in the edge location and provide it to you in very less time

**Project:**

Create an S3 bucket enabling the static website hosting and disabling the public access on the bucket

Now create a CDN distribution by adding the bucket name and give the edge locations we can also give the all edge locations after creation is done we will be having a URL which can be directly accessible through the edge location .

When we access through the CDN endpoint link the end user just can see the content where if he connects through the S3 end point he will be able change it and knows it is an s3 object. So for security purpose also CDN is important

**ECR: ELASTIC CONTAINER REGISTRY:**

ECR is a hub which stores and manages the Docker containers

Docker hub is a public repository everybody in the world can use your repositories

Container is a package which contains application code and the software dependencies which are required to run this app code.

By default ECR repo is a private repo. However we can also make It private.

Why it starts with elastic in all the AWS services-highly available and scalable

**ECS VS EKS VS KUBERNETES:**

When we have docker platform why we are looking at other solutions like ECS?

Docker failed to provide two features like **AUTO HEALING** And **AUTO SCALING**

When a bad user checkedin and try to down the app which is running on docker platform the docker doesn’t have the capacity of Auto healing and making the app up automatically . even if we continoulsy monitor it and up the application there will be a change in the ip and ultimately user will fail to access the application. And when there is a unusual or unexpected traffic since the docker is using the compute power of the host VM it fail to serve unusual traffic . so is there is an evolution of concept called container orchestration environment which can solve the problem of auto healing and auto scaling.

There are many Kubernetes platforms came in to picture like Openshift(which has an underlying OS of Redhat and provide multiple security feautures)

AWS don’t want to use the concept of Kubernetes like pods,services,deployments so they started their own solution ECS which has the features like  
Tasks  
Tasks definitions.  
Services  
Clusters

Disadvantage of ECS:- It is only restricted to AWS we cannot migrate and work on the same way we did in AWS

Kubernetes has a huge community and ppl from many organizations and fields who keeps on raising issues and adding features to the platform

In Kubernetes we have an option called custom resource development.

For example there is a feature which is highlighting in the ECS or any other Container orchestration toll but we don’t have that in Kubernetes then we can develop that feature and can deploy through the CRD. This is the advantage of the vast community.

Argocd is the Kubernetes controller tool

The main advantage of the ECS is we directly go ask AWS to get a containers cluster for your app where the containers are deployed on the EC2 or it should use the Fargate.

Pod –definition which define how your containers should—this is in traditional K8S(pod.yaml),It gives what all resources and volumes your container need., single container or multi container.

Where in the AWS we use Task definition to define how the containers should run.(task definition .yaml)

When containers are running we create services which help to create load balancers

Steps to work with ECS:

* Create the cluster
* Pick either of the options where your container should run(EC2/FARGATE)

**Secrets Management in AWS:**

In the AWS we have two solutions for storing the secrets  
1.systems manager(It has parameter store to store the secrets)

2.secrets Manager(It rotate the secrets every 180 days for very high sensitive info we use this secrets manager)

3.Hashicorp vault.(Used by most of the companies)

We have to write the role which help services to connect with the systems or secrets manager

**AWS CONFIG:**

We have a rule while creating a EC2 instance i.e we should enable the detailed monitoring. We created two instances with one enabled with monitpring and one disabled.

Every organization has their own set of compliances. Lets say some orgs will make it mandatory to put tags for resources. Some orgs may ask to enable monitoring for the resources. So now AWS config will allow us to create the compliance rules where all the resources in the organization should adhere to.

We write the compliance rules by using the lambda function.

**AWS CLOUD MIGRATION STRATEGIES.**

Rehost

Replatform

Repurchase

Refactor/Reacrchitechture(when we need change on-prem monolithic to microservice arch in cloud we try to use best possible services to improve the working of the app).

Relocate

Retire

Retain

The above all 7 strategies are on paper

But in real time we need to create a project which involves the following

* Preparation
* Planning
* Migration
* Monitoring
* Optimize(improve).

We don’t migrate a single day we do migrate it around 2-3 months based on the project. After monitoring the migrated application we will try to improve the service.

The 7R’s comes in the preparation and planning and migration phase

REHOST: it follows lift and shift strategy . lets say we have a Kubernetes applications in the on-premises with 8 microservices deployed on pods. Now to migrate we create a Kubernetes by EKS and create pods by using EC2. So when there is no dependency on OS,Hardware..etc we can go with the REHOST strategy.

**Elastic File System(EFS):**It is a shared file system for linux systemswhere multiple instances can access(Read and write) it at the same time

EFS is a region based storage service. Instance from different AZ’s can access it

**AWS RDS:**

Service that runs Relational data bases in the AWS cloud

It also automate DB tasks

AWS RDB engine offers data encryption while storing sending and receiving data

Types of DB Engine AWS supports:

1. My SQL
2. MS SQL server
3. Oracle DB
4. Maria DB
5. PostgreSQL
6. AWS AURORA(designed by AWS which provides highly available data and 5times faster than Mysql ,manage large amount of data)

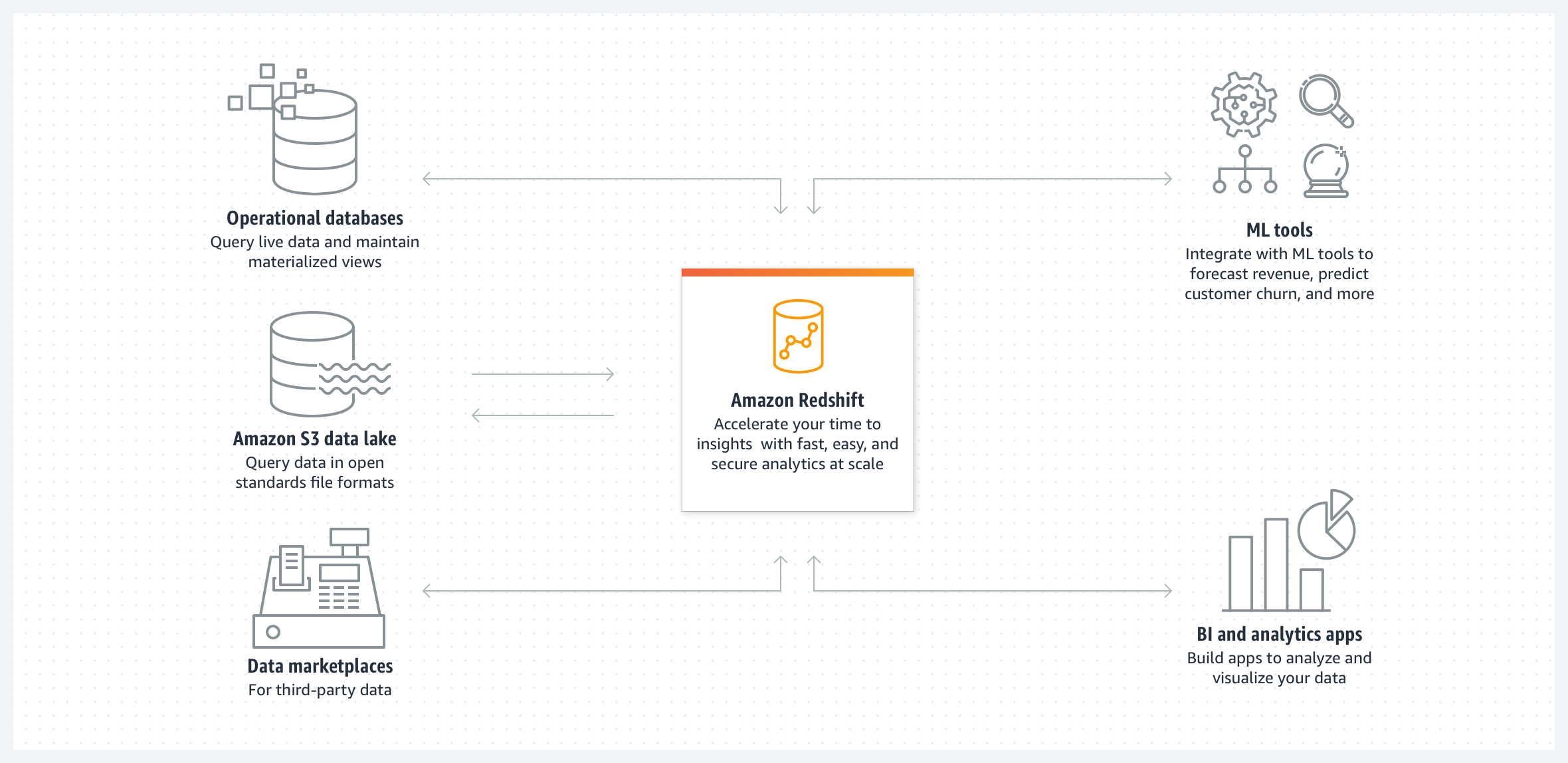
**AWS Dynamo DB:**

Service that run non-relational databases.(Key value pairs)

As AWS DynamoDB is a serverless database, you do not have to manage servers or an operational system to use it.

AMAZON REDSHIFT:

Data ware house where data from all the sources are stored and we can impend perform analytics on it by using BI tools,ML algorithms



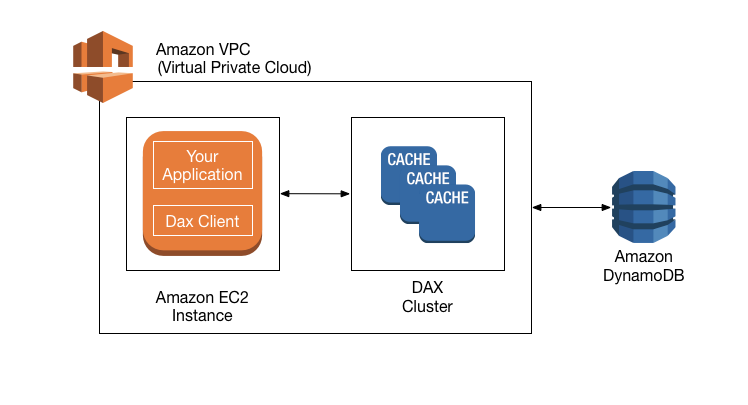
**AWS DMS:**

Data migration service is used for transferring data from one db to another DB

We have source DB and destination DB. Source DB will be operational while migrating the data.

DMS is low cost,reliable, and very least down time.

Additional DB services.

1. AWS Document DB(NON RDB that supports mongoDB)
2. Neptune(Graph DB used for creating Graphs for various purposes)
3. Quantum Ledger DB(provides historical data of all your app changes)
4. AWS managed Block chain
5. AWS Elastic cache(adding caching layers on top of DB for quick reading of data in DB)
6. AWS DynamoDB accelerator(DAX)(Deploying DAX client on ec2 for cache purpose and placing it infront of dynamo 

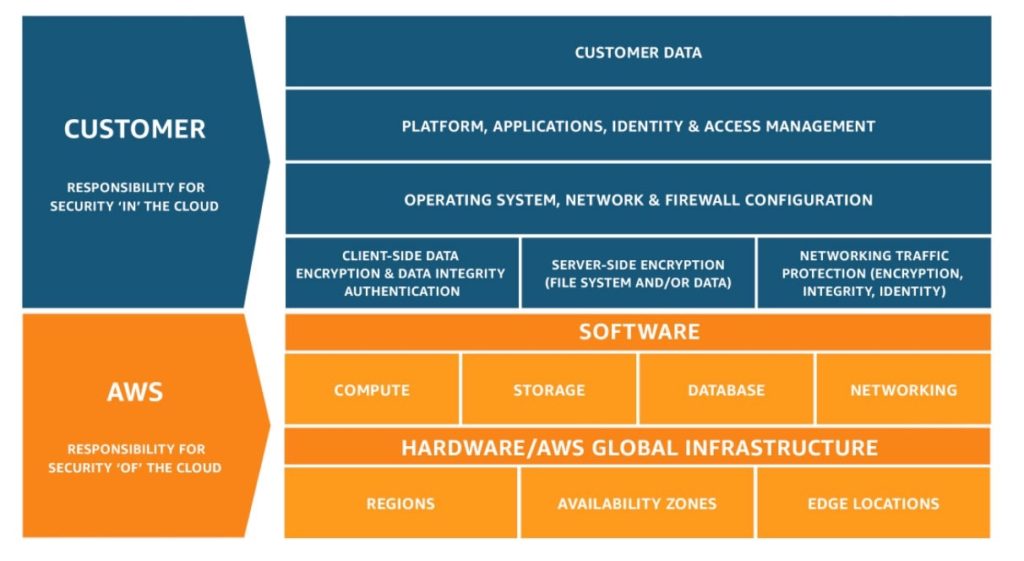
AWS SMS(Server migration service): used for migrating server without using any agent

Aws data migration hub:

Used for tracking the data migration across the AWS accounts

**AWS Shared Responsibility :**

## **Responsibility of AWS**



AWS's responsibility is the security of the cloud.

AWS manages all infrastructure layers.

Some of the infrastructure layers are:

* Data centers
* Hardware and software
* Virtualization
* Networking

Customers:

1. Encryption
2. Protection
3. Network/firewall configuration

## **Responsibility of a Customer**

Customers' responsibility is the security of everything they make in AWS Cloud.

Customers (you) have complete control over your content.

Customer manages AWS services, software, and access to the data.

**User Access:**

IAM features are:

* AWS account root user
* IAM Users
* IAM policy(default policies and custom policies [json scripts])
* IAM groups
* IAM roles(same as users will get temp creds but this role will allow you to ork on a specific service in the AWS console)
* Multi-factor authentication

**IAM**

Authenticate and Authorize any user

Provide MFA to the account(console>>Security creds option>>)

Authentication(creating user and providing him credentials)

Authorization(providing permissions to the user to access the services by adding the policies to him,adding user to specific group which is already created by configuring specific access policies)

IAM ROLE:- User associated with the resource

**AWS Organizations(**central location to manage multiple AWS accounts**)**

>>Container of AWS Accounts

>>it Allows you to manage permissions of your organizational accounts

**Organizational units (OUs)** are groups of accounts in the AWS organization.

OUs are used to manage accounts with the same or similar permissions more easily.

The permission policy of OU is applied to all OU accounts.

SCP service control policy manage permission for the OU

20 accounts can be linked

Volume discounts can be obtained

Root account pass must be strong

The root account should only be used for billing donot deploy any resources inside it.

Cloud trail is on per account basis but billing is associated on a single biucket

## **AWS Artifact**

AWS Artifact is a service.

It provides access to compliance reports and AWS security on demand.

It consists of AWS Artifact Reports and AWS Artifact Agreements.

There are many regulations and reports you can find in AWS artifacts.

**AWS Artifact Agreements** are agreements about the use of certain types of information.

**AWS Artifact Reports** give you compliance reports.

## **AWS Shield**

AWS Shield gives protection against DoS and DDoS attacks.

Denial of services/Distributed denial of services >> these are the attacks from a person or a bot which push excessive flow of data in to the website or the app which overwhelm the website with execisive traffic.

It provides standard and advanced protection.

AWS Shield Standard protection protects all AWS users at no expense.

**Extra level Security services:**

1. **KMS (key management service)[secure app data with cryptographic keyS]**
2. **AWS WAF(Web app firewall>>controls/or monitors application network requests) prohibit sql injection attacks**
3. **Amazon Inspector**

It checks the application for software versions and other vulnerabilities.

It offers you a report of all security issues and solutions recommendations for your application.

1. **Amazon GuardDuty**

Amazon GuardDuty is a threat detection service.

It detects threats for AWS resources and infrastructure.by contionously monitoring the app

**Monitoring and Analytics:**

**Cloud Watch:**

Used for monitoring AWS services

Creating alarms and sending alerts by using SNS topic

**Cloud Trial:**

Account level monitoring,detects suspicious API calls to the AWS account  
it tracks the identity,IP,Location of the person who logged in to the AWS console

**AWS Trust advisor:**

It is a webbased ,real time recommendation service. It helps in choosing the AWS service based on our request to optimize the cost

**AWS Pricing and Support**

* [AWS Pricing and Support](https://www.w3schools.com/aws/aws_cloudessentials_ps_intro.php)
* [AWS Free Tier](https://www.w3schools.com/aws/aws_cloudessentials_ps_freetier.php)
* [AWS Pricing Models](https://www.w3schools.com/aws/aws_cloudessentials_ps_pricemodels.php)
* [AWS Billing Dashboard](https://www.w3schools.com/aws/aws_cloudessentials_ps_billingdashboard.php)
* [AWS Consolidated Billing](https://www.w3schools.com/aws/aws_cloudessentials_ps_consolidatedbilling.php)
* [AWS Budgets](https://www.w3schools.com/aws/aws_cloudessentials_ps_budgets.php)
* [AWS Cost Explorer](https://www.w3schools.com/aws/aws_cloudessentials_ps_costexplorer.php)
* [AWS Support Plans](https://www.w3schools.com/aws/aws_cloudessentials_ps_supportplans.php)
* [AWS Marketplace](https://www.w3schools.com/aws/aws_cloudessentials_ps_marketplace.php)

**AWS migration and Innovation:**

## **What is AWS CAF?**

AWS CAF is a framework that walks you through migration of applications to the cloud.

It provides suggestions assisting you in the migration process.

CAF has six focus areas:

1. Business(investments,profits,loses)
2. People(HR)
3. Governance
4. Platform(Solution Architech)
5. Security
6. Operations(operation analyst)

Above are also know as perspectives

## **What Are Migration Strategies?**

Migration Strategies are plans that help you move your applications into the cloud.

There are six most common strategies you can implement for your application migration:

1. Rehosting(Lift and shift your app)
2. Replatforming(Lift ,Tinker and shift)
3. Refactoring(change the architect itself like app core,environment..etc)
4. Repurchasing(changing the business type—paas to saas)
5. Retaining(keeping business crucial apps that has to be tinkered)
6. Retiring(Removing unnecessary apps

**AWS SNOW FAMIMLY:**

Devices that transport data in and out of the AWS

Snow cone  
Snow ball  
Snow mobile.

**AWS WELL ARCHITECTED FRAMEWORK:**

## **What is AWS Well-Architected Framework?**

AWS Well-Architected Framework is a tool that uses best practices to find improvements for your applications in the cloud.

It helps you in five areas:

1. Operational excellence(It is a capacity to monitor and manage the apps)
2. Security(Record audit and evaluate changes)
3. Reliability
4. Performance efficiency
5. Cost optimization

Those areas are also called the **five pillars** of AWS Well-Architected Framework.

Best Practices for Reliablilty in the cloud:- Foundation,Change management,Failure Management.

**AWS Global Infra structure:**

---every AWS region is completely isolated with each

---Each AWS region have min of 3 AZ’s and Each AZ consists minimum of 1 or more Dc’S

---Az’S In the same region are connected with the low latency links

**AWS outpost** is a servicewhich allow user to run aws services in the on-prem.

**AWS wavelength** is used for reducing the latency for the specific apps that supports 5G network doubt how it works??

**Edge locations** are CDN end points for cloud front   
**Regional Edge Caches** sit between your CloudFront Origin servers and the Edge Locations.

AWS step function:

It is a AWS service which is used for maintain or creating a proper workflow of an app which is created by using the microservices or orchestration tools …etc by using a series of steps

By using AWS Step Functions, you can abstract the complexity of managing the flow of your serverless application and focus on the business logic of each component. This results in more maintainable and scalable applications, with built-in error handling and monitoring capabilities.

**Amazon MQ** is a managed message broker service provided by Amazon Web Services (AWS). It is designed to facilitate the communication and exchange of messages between different software components, systems, and applications in a distributed and decoupled manner. Amazon MQ is based on the popular messaging protocols, such as Advanced Message Queuing Protocol (AMQP) and Message Queuing Telemetry Transport (MQTT).

**EXAM Questions:**

Free Exam practice:- <https://www.examtopics.com/exams/amazon/aws-certified-cloud-practitioner/view/>

Exam questions:How do you provide high availability for Web app hosted in AWS  
ALB infront by spreading instances in multi AZ

How consolidated billing is reduced across the multiple accounts?

By leveraging SCP(Service controll policy)

IAM ROLE:- User associated with the resource

Doubts—32,78,87

RDS deployed on EC2 need to be patched every week how it can be done?

Connect to AWS sys manager and to automate the weekly patches according to the schedule

AWS Enterprise support:- 24/7 support with less cost with TAM to help

Aws billing support available to all support levels-- AWS Customer service

Global deployment of compute and storage – resource groups

Cheap cost of ec2--- reserved and spot.

Run or terminate at any point of the day,90% discount, – spot

Trusted adivosr: -- cost and security advices after monitoring the app

Decrease network latency :- cloud front, global accelerator.

Bi tool and reporint dash boards:- quick sights

Data closely to its primary used:- Global foor print

Cloud and on-prem which is easy and managed by AWS—PHYSICAL STORAGE,NETWORK INFRA

Server side encryption:- customer

Which service supports Hadoop framework:- aws EMR

ETL Tool- AWS Glue

Redshift:- data ware house

**EXAM Q’s on CAF:**

Common stake holders of CAF:- CTO, Tech Leaders, Engineers, Archcitechs.

**EXAM Q’s on Technical:**

On-demand instance:

Dedicated instance;

Reserved instance: not an instance we can take a longterm contract.

AWS partner network:- used for infra migration.

Encryption done to data before sending data to client is called **Client side encryption** and can be done by using SDK

AWS Shield: Ddos Protection, for apps running on R53 and Global Accelerator

Scale out/Horizantal :- increasing number of computers

Scale in/vertical:- increasing cpu

When we want to check if a user followed all the governance rules..etc;:- we use cloud trail for their security

AWS WAF:- Layer 7

AWS debugging:- amazon pin point

Ec2 instance can access files in EFS(Regional service) at any AZ,REGION,VPC.

High availability b y default: EFS,Dynamo DB

Auto encryption available for EFS,S3.

Finding under untilized resources; Trusted advisor,aws cost explorer.

Best way to protect the data that is accidentally deleted in S3 bucket is S3 versioning.

**AWS Transit Gateway** - AWS Transit Gateway connects VPCs and on-premises networks through a central hub. This simplifies your network and puts an end to complex peering relationships. It acts as a cloud router –

In ASG scale in reduce the number of AWS EC2 instance based on input load  
In ASG scale out increase the number of AWS EC2 instance based on input load

To get separate invoice for prod and dev we need to create separate accounts for prod and dev.

Best storage option for AWS EC2 to improve caching capabilities is Instance store.

S3 glacier deep archive – retention for around 7 to 10 years.

**AWS Storage Gateway** - AWS Storage Gateway is a hybrid cloud storage service that gives you on-premises access to virtually unlimited cloud storage. All data transferred between the gateway and AWS storage is encrypted using SSL (for all three types of gateways - File, Volume and Tape Gateways). AWS Storage Gateway cannot be used for data archival.

AWS kendra:- intelligent search service foe old docs,files..etc

AWS Personalize ;- building app with ML technology

AWS Comprehend:NLP Service

AWS LEX: it builds conversational interface where it use ASR to convert speech to text and NLU to understand the intent of the text

Cloud foundations: used to help deploy configure and secure its new work loads and ensuring ready for on going operations

Security tests or penetration testings:- penetration testing

**EXAM Q’s on cost pricing and billing and support:**

Three fundemantal drivers of cost with AWS are: compute,storage and outbound data transfer

AWS publish billing reports to the AMAZON S3 bucket

AWS budget: send alerts to the users when threshold is crossed and also set a custom budgets and sent alerts when budget exceeds

AWS cost and pricing :- can estimate pricing for your resources

AWS abuse team:- when aws ip is misused

AWS cloud trail:- encryption on by default

One region min three AZ’s

One AZ one or more discrete data centers

Aws cloud trail : default encryption by AWS

Minimum time any instance can be billed: 60sec

AWS Business Support:- Architectiural guidance

AWS ramp-enterprise :- 24/7 critical support

AWS development support:- testing and deploying any application on AWS

Data Transfer in a same region and inbound data transfer will not be charged.

AWS shield standard:- enabled for all customers by default without cost.

**On-demand instance:-** no upfront payments, no long term commitments and pay for compute capacity per hour or per second.

Dedicated host:- ec2 server dedicated to a single customer that runs In a vpc.used when you want to leverage a existing software

Dedicated instance:- Ec2 server run on a vpc on a hardware which is dedicated to a single customer.

Reserved instance:- up to 75% compared to on-demand  
good for resources with predictbale usage.

**Spot-instance**

* Purchase spare computing capacity with no upfront commitment at discounted hourly rates.
* Provides up to 90% off the On-Demand price.
* Used for apps with flexible start and end time

**There are four AWS support plans available:**

* Basic – billing and account support only (access to forums only).
* Developer – business hours support via email.
* Business – 24×7 email, chat, and phone support.
* Enterprise – 24×7 email, chat, and phone support.

Enterprise support comes with a Technical Account Manager (TAM).

Developer allows one person to open unlimited cases.

Business and Enterprise allow unlimited contacts to open unlimited cases.

by using the aws **cost explorer** we can check past 13 month cost and can forecast next 3 month costs

**aws pricing calc** can estimate pricing for the resources you are planning to deploy

**Advante of using AWS OVER TRADITIONAL IT**

AWS Trade capital expense for variable expense

Need not estimate the cost

Highest possible discount offered for spot instance:- 90  
Gaurd duty:-- monitoring and finding malicious activity

AWS inspector :- automated security assessment and un intended entrance to the network, vulnerabilllitites

If you donot have phone for MFA connect to the virtual MFA

AWS TRANSIT GATEWAY(centralized hub for connecting VPC)  
AWS STORAGE GATEWAY(Hybrid storage unit)

Data encryption is enabled for S3 and AWS storage gateway.

Site to site VPN components:virtual private gateway and customer gateway

Mandatory elements of IAM policy while writing with the JSON:- Effect and Action

SG is Stateful and NACL is Stateless

Gaurd duty:-- monitoring and finding malicious activity

Hourly report :- AWS cost and usage report

Spot instance can be interrupted

On-demand –can pay for hour or second

Cloud watch billing metric is stored in the US-EAST-1 N.virginia