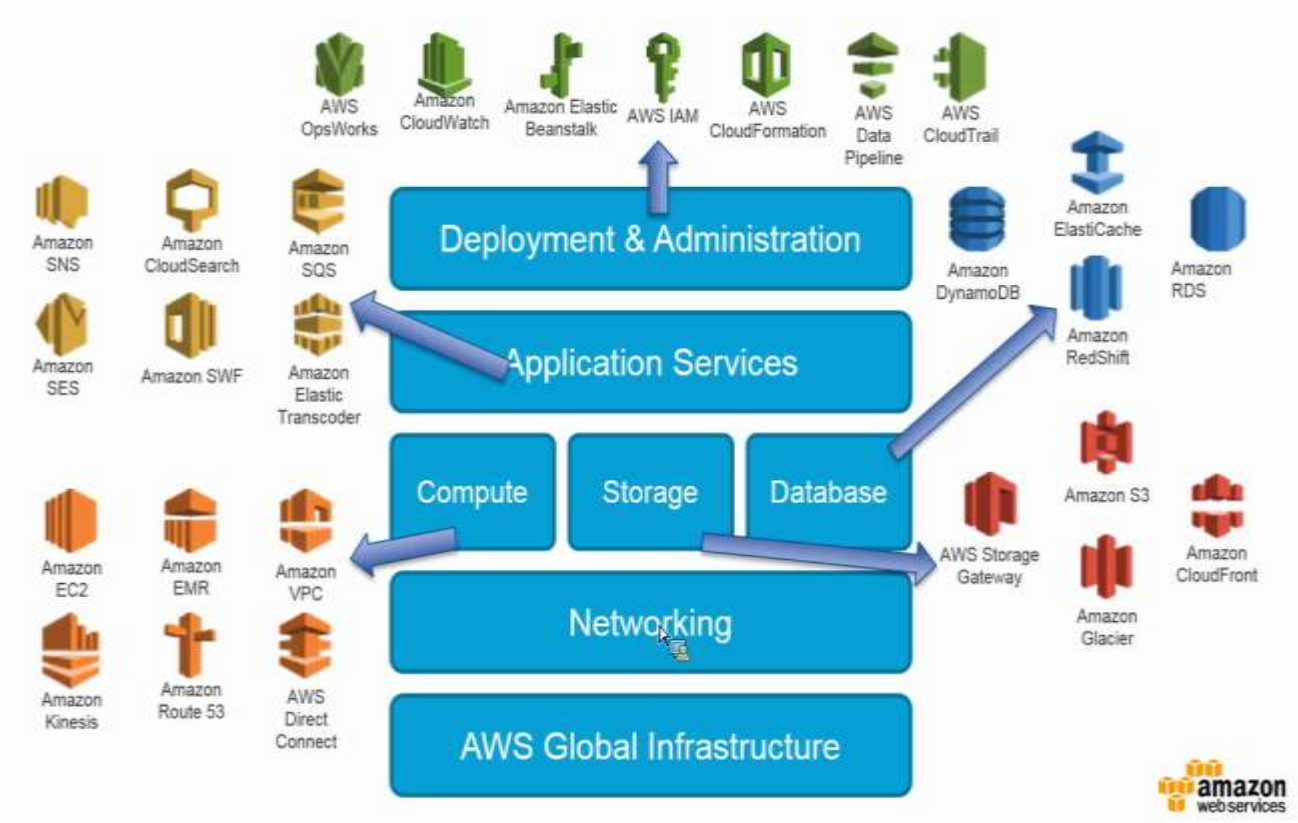
Amazon Web Services Interview Tips



SRIRAM

Cloud Gurus

Jeff Bezos

Andy Jassy

Bill Gates

Sriram

"Invention requires two things:

- ✓ The ability to try a lot of experiments
- ✓ Not having to live with the collateral damage of failed experiments"

– Andy Jassy CEO, Amazon Web Services

Never be a prisoner of your past, Be a Cloud Architect of your future

If you are not loving Cloud you might be doing wrong in spending patterns – Sriram, Cloud Chief Architect

Awards & Honors

Best Cloud Solution Architect – "On The Spot Award", Tata Consultancy Services, 2012

TCS Gems

To

Sriram Balasubramanian

In appreciation of your outstanding contribution to the organisation, you are awarded the

On The Spot Award(12-Apr-2013)

You are an inspiring role model to your colleagues.

Thank you for your dedication and commitment.

TATA

TATA CONSULTANCY SERVICES

Experience certainty.

Ajoy Mukherjee

EVP & Head, Global HR

Best Cloud Chief Architect Award, Syntel Consulting Inc., 2017

SYNTEL
Consider IT Done®

Best Architect Award

Dear Mr. Sriram Balasubramanian I would like to appreciate you on behalf of Banking & Financial Services for being a Chief Architect, in building and providing solutions in the area of Cloud Computing – Amazon Web Services (AWS), Microsoft Azure & DevOps project, which is more valuable towards client partner solutions with the alignment of our organization I look forward to your continued support.

2017

V. S. Raj
BU Head - BNFS

Preface

I have been involved in IT Software development since 1997. I have a unique combination of process, technical and industrial skills. As an Enterprise Architect, I have expert level of knowledge in agile and technology practices such as AWS, Azure, DevOps, java, Hadoop, SharePoint & .Net with this combination I can help process and technology people, understand the world. Worked in India, USA, and UK which creates a global experience and awarded as a Best Enterprise Architect. Dedicated "Amazon Web Services Interview Tips" book to my family members, friends. This Guide made handy and recollect everything at one shot.

Organization of this Book

Amazing Amazon Web Services is designed to make you to success in the interview by providing valuable questions on various topics along with the tips to achieve AWS Architect Certification. The progressive elaboration of AWS knowledge towards an AWS Architect is awesome. Enjoy Reading!

AWS Certified Professional

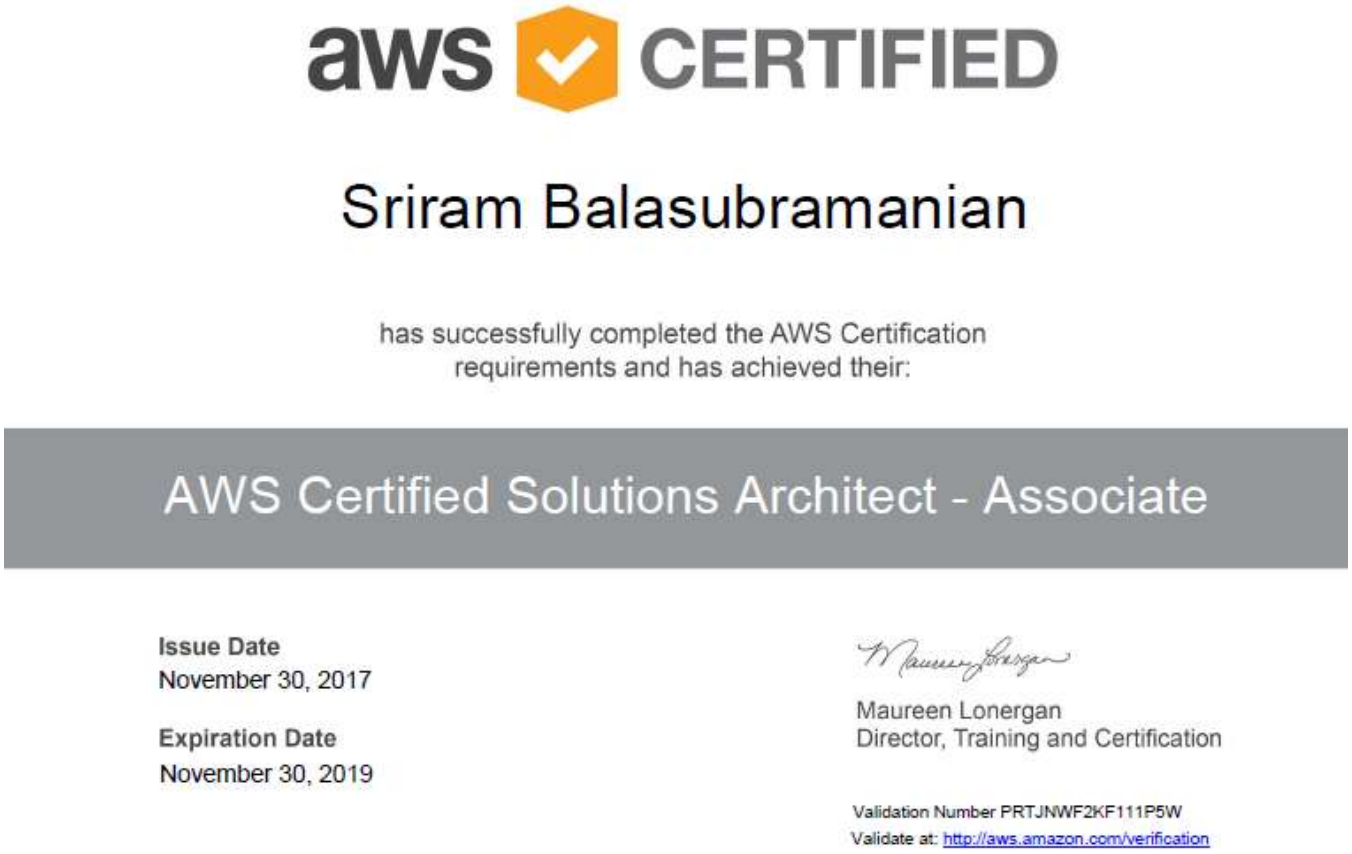


Table of Contents

1.Cloud Computing

What is Cloud Computing?

Practice of using a network of remote servers hosted on the Internet to store, manage, and process data, rather than a local server or a personal computer is called Cloud Computing.

Companies offering these computing services are called "cloud providers" and typically charge for cloud computing services based on usage, similar to how you are billed for water or electricity at home.
E.g.: AWS, AZURE, IBM Bluemix, GOOGLE CLOUD

This cloud model is composed of five essential characteristics, three service models, and four deployment models.

The primary reasons for moving to the cloud are: -

- **You don't need to maintain or administer any infrastructure**
- **It will never run out of capacity, since it is a virtually infinite**
- **You can access your cloud-based applications from anywhere, you just need a device which can connect to the internet**

What are the benefits of Cloud Computing?

- **Totally free from Maintenance i.e., You do not have to maintain or administer any infrastructure for the same.**
- **Lower Computing Cost**
- **Improved Performance**
- **Reduced Software Cost**
- **Instant Software Updates**
- **Unlimited Storage Capacity i.e., It will never run out of capacity, since it is virtually infinite.**
- **Increased Data Reliability**
- **Device Independence and the "always on! anywhere and any place" i.e., You can access your cloud-based applications from anywhere, you just need a device which can connect to the internet.**

Cloud Computing is the fastest growing part of network-based computing. It provides tremendous benefits to customers of all sizes: simple users, developers, enterprises and all types of organizations.

Why Cloud Computing?

- Lower TCO
- Reliability, Scalability & Sustainability
- Secure Store Management
- Low Capital Expenditure
- Frees from Internal Resources
- Utility Based
- Easy & Agile Deployment
- Device & Location Independent
- 24 * 7 Support
- Pay As You Use

What are the top 10 advantages of Cloud Computing?

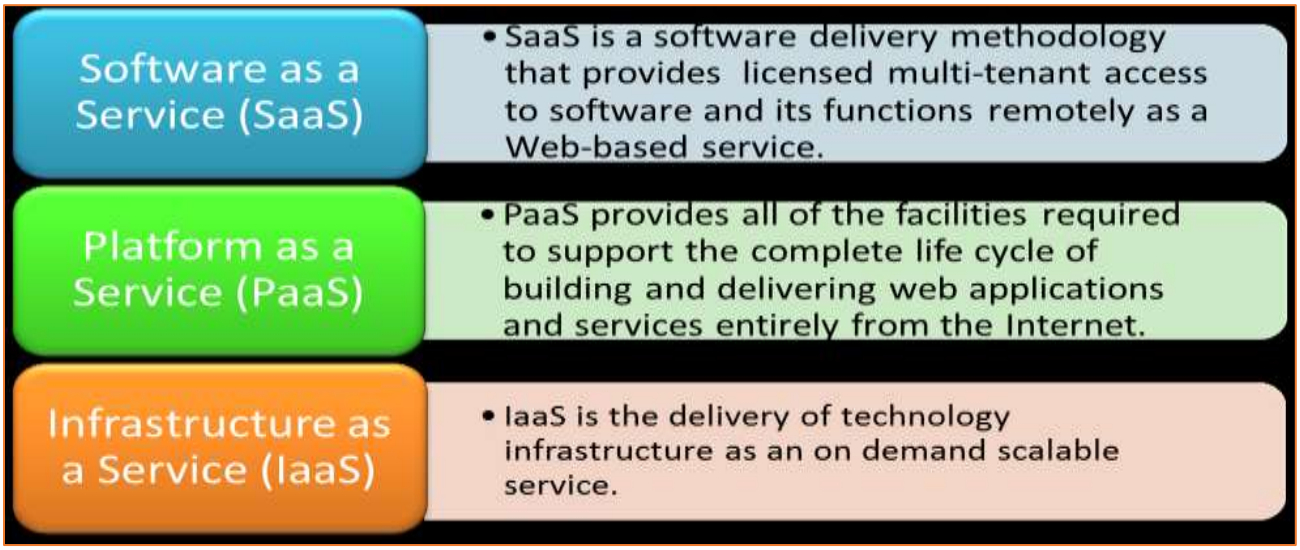
- Pay as you Go Model
- Increased Mobility
- Less or No CAPEX
- High Availability
- Easy to Manage
- High Productivity
- Environment Friendly
- Less Deployment Time
- Dynamic Scaling
- Shared Resources

What are the different layers (Service Models) of cloud computing?

Cloud computing consists of 3 layers in the hierarchy and these are as follows:

1. Infrastructure as a Service (IaaS) provides cloud infrastructure in terms of hardware like memory, processor speed etc.
2. Platform as a Service (PaaS) provides cloud application platform for the developers.
3. Software as a Service (SaaS) provides cloud applications which are used by the user directly without installing anything on the system. The application remains on the cloud and it can be saved and edited in there only.

What are the 3 service models of Cloud Computing?



IaaS

Infrastructure as a Service (IaaS) is the most basic category of cloud computing services. With IaaS, you rent IT infrastructure—servers and virtual machines (VMs), storage, networks, operating systems—from a cloud provider on a pay-as-you-go basis.

The Key features are: -

- Instead of purchasing hardware outright, users pay for IaaS on demand.
- Infrastructure is scalable depending on processing and storage needs.
- Saves enterprises the costs of buying and maintaining their own hardware.
- Because data is on the cloud, there is no single point of failure.
- Enables the virtualization of administrative tasks, freeing up time for other work.

PaaS

Platform-as-a-service (PaaS) refers to cloud computing services that supply an on-demand environment for developing, testing, delivering and managing software applications. PaaS is designed to make it easier for developers to quickly create web or mobile apps, without worrying about setting up or managing the underlying infrastructure of servers, storage, network and databases needed for development.

The Key features are: -

- PaaS provides a platform with tools to test, develop, and host applications in the same environment.
- Enables organizations to focus on development without having to worry about underlying infrastructure.
- Providers manage security, operating systems, server software, and backups.
- Facilitates collaborative work even if teams work remotely.

PaaS Examples: AWS Elastic Beanstalk, Microsoft Azure, Google Apps, Salesforce Force.com & IBM Bluemix

PaaS Billing: PaaS will typically be billed based on memory usage i.e., IBM Bluemix

SaaS

Software-as-a-service (SaaS) is a method for delivering software applications over the Internet, on demand and typically on a subscription basis. With SaaS, cloud providers host and manage the software application and underlying infrastructure and handle any maintenance, like software upgrades and security patching. Users connect to the application over the Internet, usually with a web browser on their phone, tablet or PC.

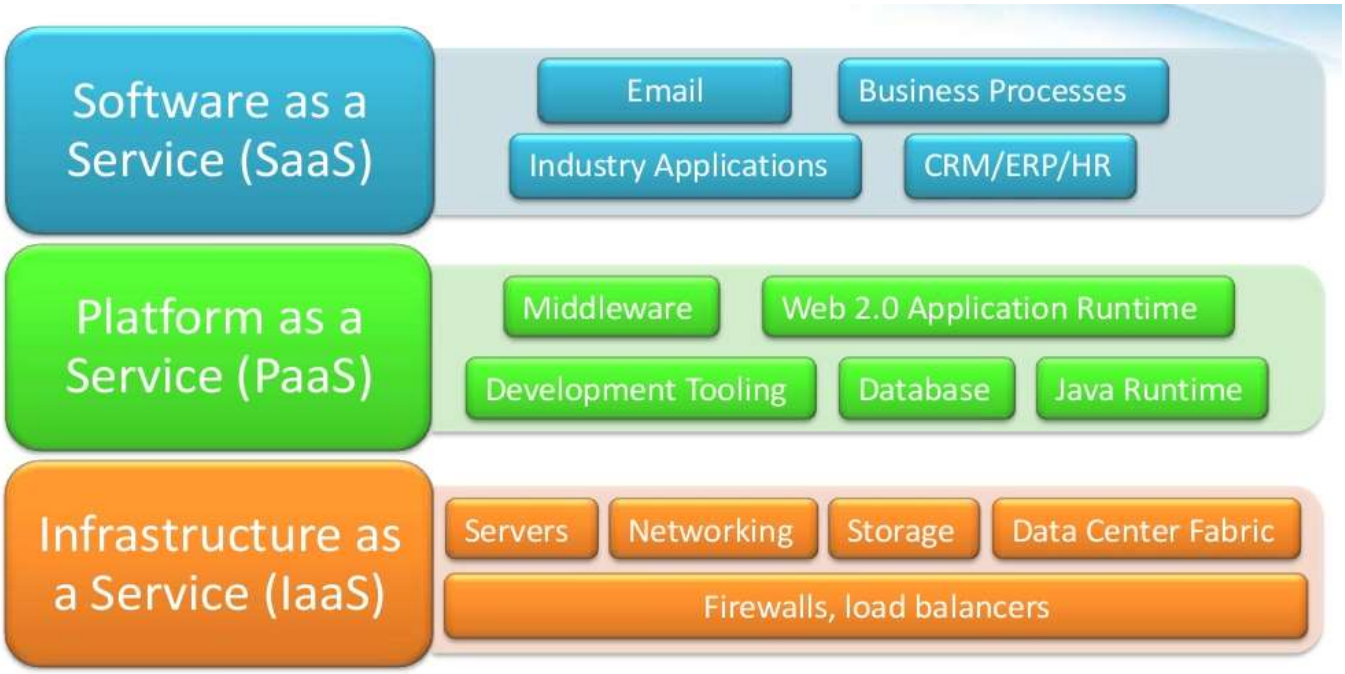
The Key features are: -

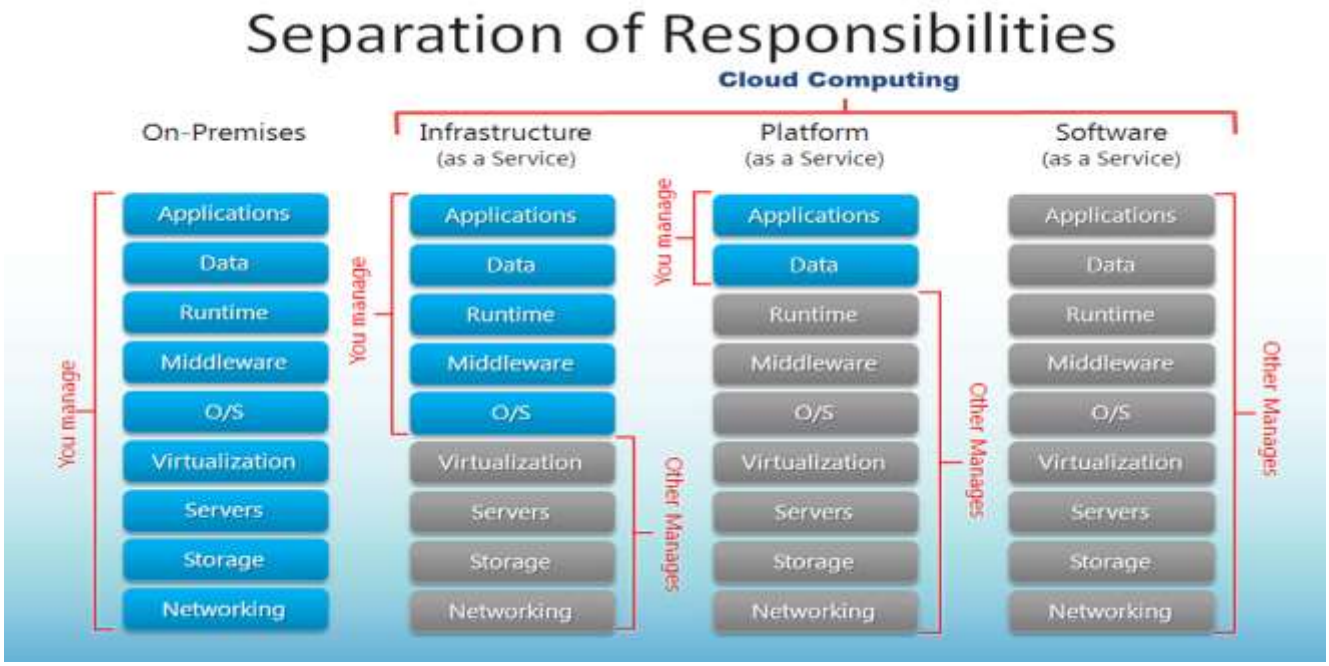
- **SaaS vendors provide users with software and applications on a subscription model.**
- **Users do not have to manage, install, or upgrade software; SaaS providers manage this.**
- **Data is secure in the cloud; equipment failure does not result in loss of data.**
- **Use of resources can be scaled depending on service needs.**
- **Applications are accessible from almost any Internet-connected device, from virtually anywhere in the world.**

SaaS Examples: **Microsoft Office 365, Salesforce.com, Intuit, Adobe Creative Cloud & Gmail**

SaaS Billing:

- **SaaS will typically have a monthly fee per user**
- **Multiple pricing tiers may be offered based on usage E.g. Microsoft Office 365**

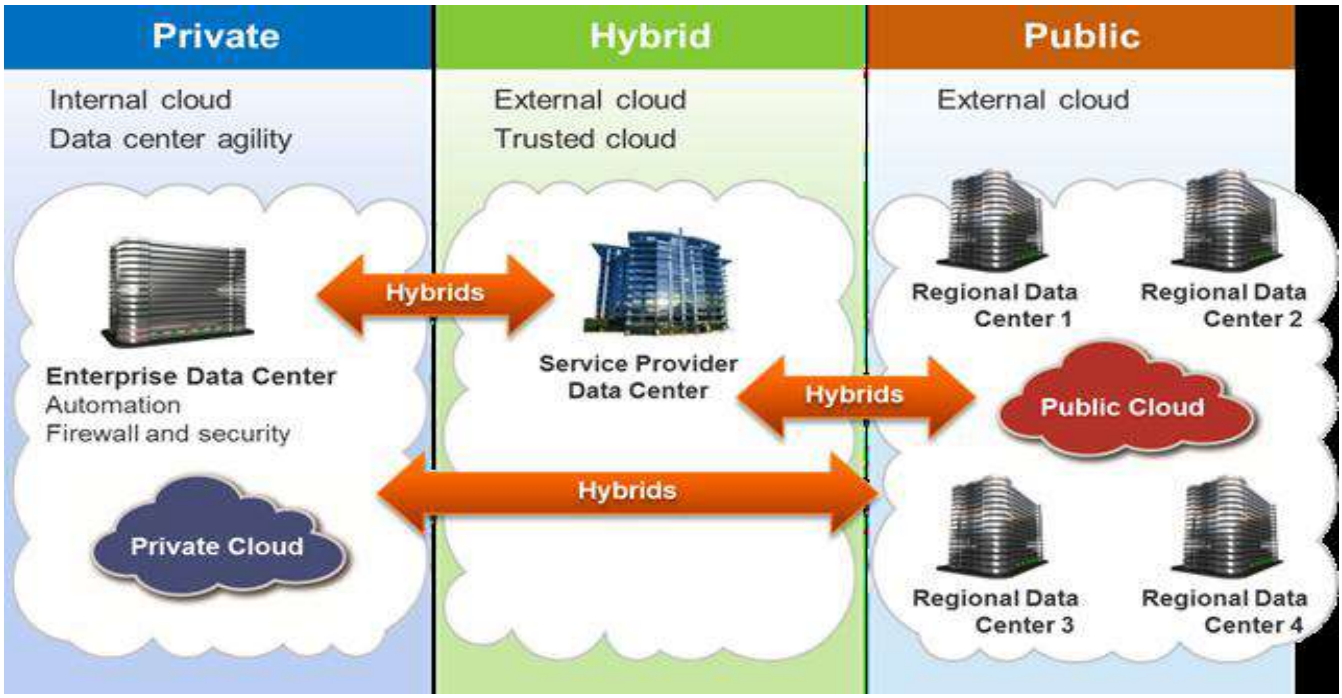


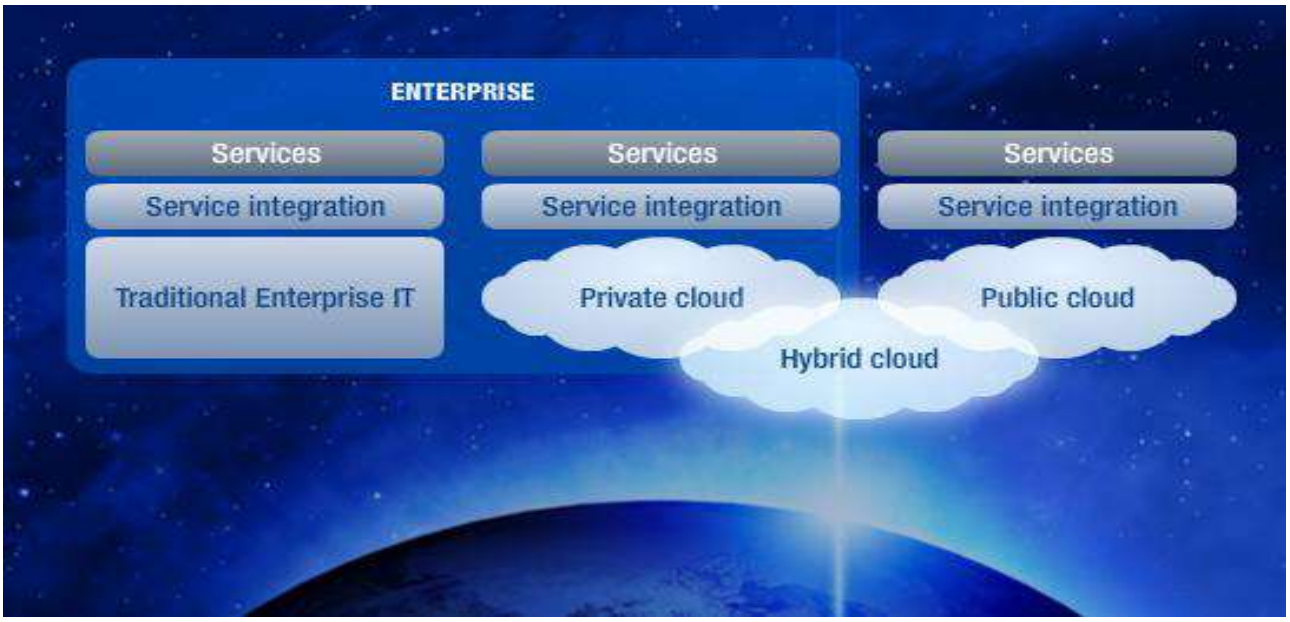


What are the 4 deployment models of Cloud Computing?

The NIST define three Deployment Models of Cloud Computing:

- **Public Cloud**
- **Private Cloud**
- **Hybrid Cloud**
- **Community Cloud**





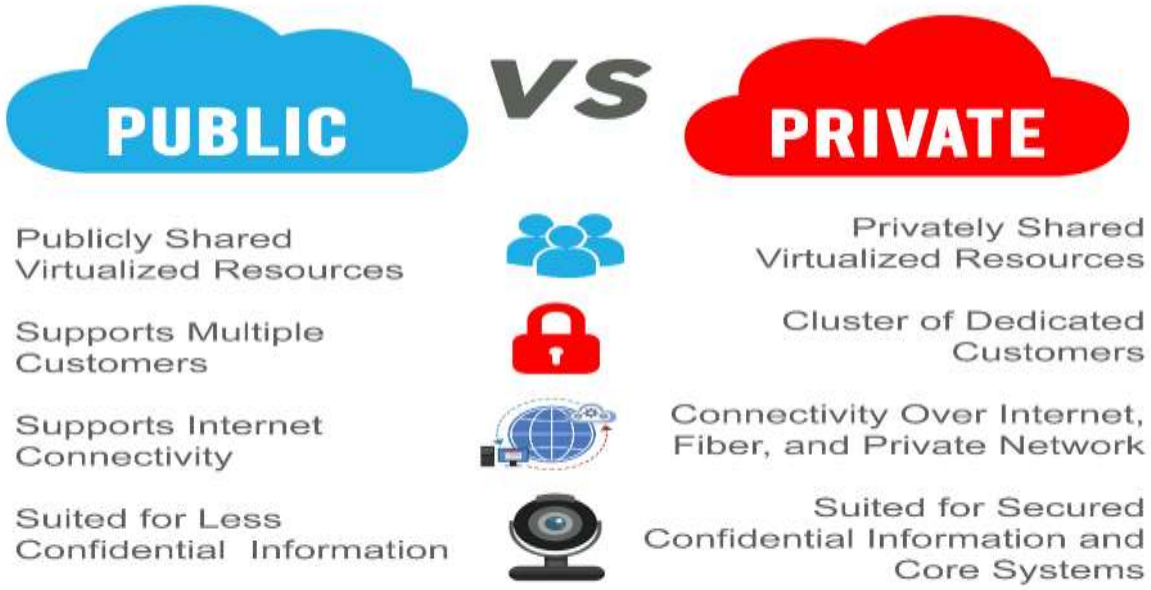
Public clouds are owned and operated by a third-party cloud service provider, which deliver their computing resources like servers and storage over the Internet. Made available to the general public or a large industry group.

Microsoft Azure is an example of a public cloud. With a public cloud, all hardware, software and other supporting infrastructure is owned and managed by the cloud provider. You access these services and manage your account using a web browser. The most common deployment model. Examples: All the cloud providers such as AWS, Microsoft Azure, IBM Bluemix, Salesforce, etc.,

Private Cloud works the same way as Public Cloud, but these services are provided to internal business units instead of to external public enterprises.

Operated solely for an organization, may be managed by the organization or a third party and Limits access to enterprise and partner network

A private cloud refers to cloud computing resources used exclusively by a single business or organization. A private cloud can be physically located on the company's on-site datacenter. Some companies also pay third-party service providers to host their private cloud. A private cloud is one in which the services and infrastructure are maintained on a private network. Examples: US DOD, Indian Military, Most of govt bodies and High revenue business.



Hybrid clouds combine public and private clouds, bound together by technology that allows data and applications to be shared between them. By allowing data and applications to move between private and public clouds, hybrid cloud gives businesses greater flexibility and more deployment options. Companies with limited Private Cloud infrastructure may 'cloud burst' into Public Cloud for additional capacity when required. A company Cloud also have Private Cloud at their main site and use Public Cloud for their Disaster Recovery location

Composition of two or more clouds (private, community, or public) bound together by standardized or proprietary technology that enables data and application portability

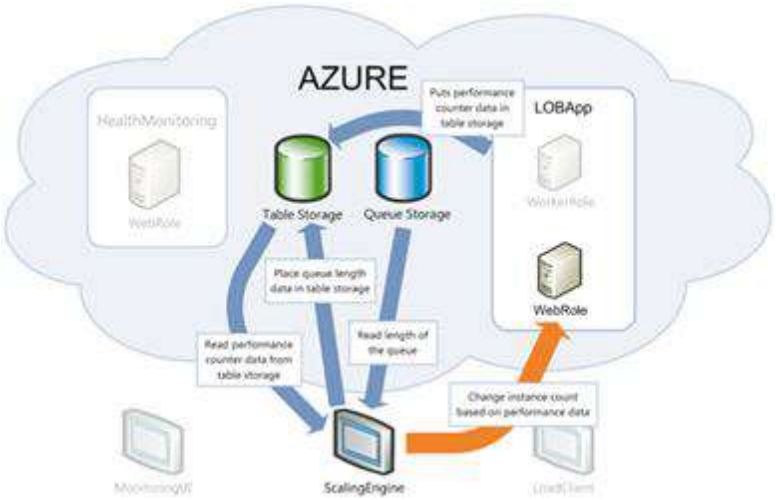
Community Cloud is similar to a traditional extranet, but with full shared data center services instead of just network connectivity between On-Premise Offices.

What are the 5 characteristics of Cloud Computing? How Private Cloud differ than On-Premise?

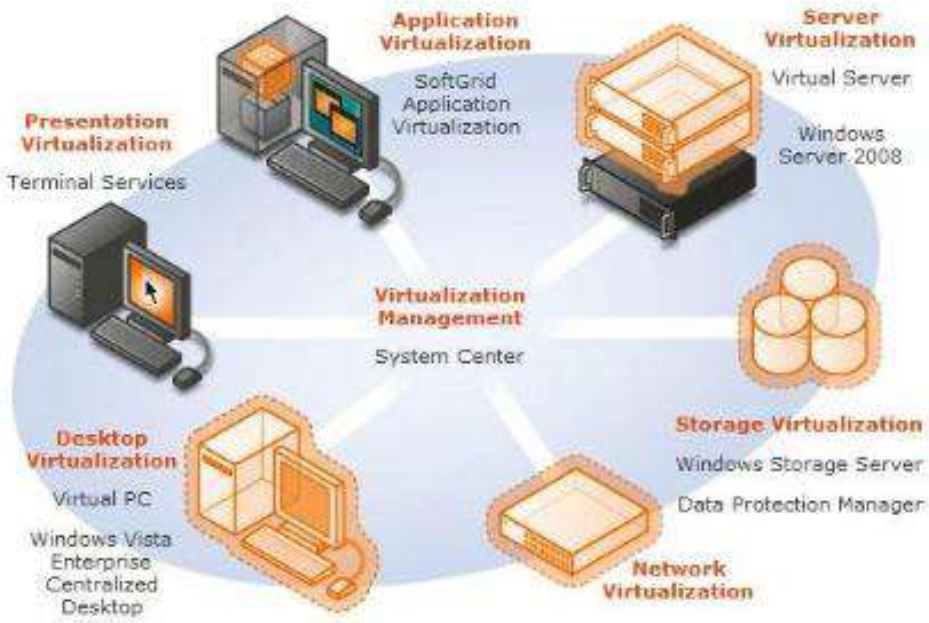
On-Demand Self Service: **On-demand self-service: A consumer can unilaterally provision computing capabilities, such as server time and network storage, as needed automatically without requiring human interaction with each service provider. – NIST**

Broad (Ubiquitous) network access: **Capabilities are available over the network and accessed through standard mechanisms that promote use by heterogeneous thin or thick client platforms (e.g., mobile phones, tablets, laptops, and workstations). – NIST**

Rapid elasticity: **Capabilities can be elastically provisioned and released, in some cases automatically, to scale rapidly outward and inward commensurate with demand. To the consumer, the capabilities available for provisioning often appear to be unlimited and can be appropriated in any quantity at any time. – NIST**



Resource pooling: **The provider's computing resources are pooled to serve multiple consumers using a multi-tenant model, with different physical and virtual resources dynamically assigned and reassigned according to consumer demand. There is a sense of location independence in that the customer generally has no control or knowledge over the exact location of the provided resources but may be able to specify location at a higher level of abstraction (e.g., country, state, or datacenter). Examples of resources include storage, processing, memory, and network bandwidth. – NIST**



Measured service: Cloud systems automatically control and optimize resource use by leveraging a metering capability1 at some level of abstraction appropriate to the type of service (e.g., storage, processing, bandwidth, and active user accounts). Resource usage can be monitored, controlled, and reported, providing transparency for both the provider and consumer of the utilized service. - NIST

Name the various layers of the cloud architecture?

There are 5 layers and are listed below

- CC- Cluster Controller
- SC- Storage Controller
- CLC- Cloud Controller
- Walrus
- NC- Node Controller

What is the way to secure data for carrying in the cloud?

One thing must be ensured that no one should seize the information in the cloud while data is moving from point one to another and also there should not be any leakage with the security key from several storerooms in the cloud. Segregation of information from additional companies' information and then encrypting it by means of approved methods is one of the options. Amazon Web Services offers you a secure way of carrying data in the cloud.

How to secure your data for transport in cloud?

Cloud computing provides very good and easy to use feature to an organization, but at the same time it brings lots of question that how secure is the data, which has to be transported from one place to another in cloud. So, to make sure it remains secure when it moves from point A to point B in cloud, check that there is no data leak with the encryption key implemented with the data you sending.

How does cloud computing provides on-demand functionality?

Cloud computing is a metaphor used for internet. It provides on-demand access to virtualized IT resources that can be shared by others or subscribed by you. It provides an easy way to provide configurable resources by taking it from a shared pool. The pool consists of networks, servers, storage, applications and services.

2.AWS Essentials

What is Amazon Web Services? What are its benefits?

- Amazon Web Services(AWS) are a collection of remote services (Also called as web service) offered by the amazon.com over the internet to build and run an application.
- Amazon Web Services (AWS) is a secure cloud services platform, offering compute power, database storage, content delivery and other functionality to help businesses scale and grow.
- Millions of customers are currently leveraging AWS cloud products and solutions to build sophisticated applications with increased flexibility, scalability and reliability.
- Amazon Web Services (AWS) is robust, scalable and affordable infrastructure for cloud computing
- AWS is Elasticity: scale up or scale down as needed,
- We can get recourses instantly & AWS is fully on demand

The benefits are: -

Pay-per use model

You are only charged for disk space, CPU time and bandwidth that you use.

Instant scalability

Your Service automatically scales on AWS stack.

Reliable/Redundant

Data is redundant in the cloud. All services have built-in security

Security

AWS delivers a scalable cloud-computing platform that provides customers with end to-end security and end-to-end privacy.

Most services accessed via simple REST/SOAP API Libraries are available in all major languages.

Service Level Agreement

SLA between 99.99 and 100% availability

Amazon S3 maintains a durability of 99.99999%

Availability

Availability Zones exist on isolated fault lines, flood plains, and electrical grids to

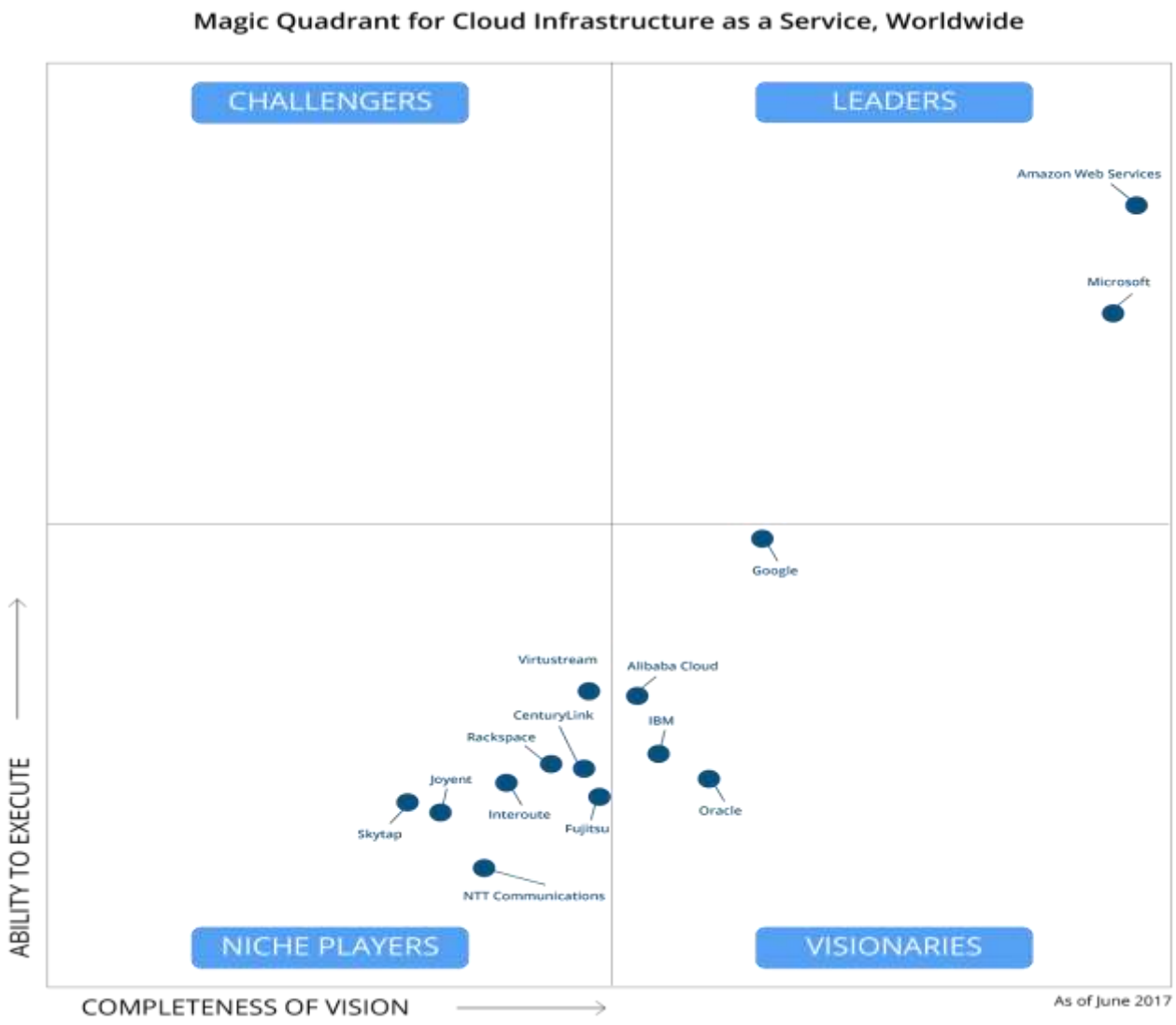
Substantially reduce the chance of simultaneous failure.

Support

AWS provides 24/7 support in the real-time operational status of all services around the globe

Why Amazon Web Services?

- **Fastest growing cloud computing platform on the planet**
- **Largest public cloud computing platform on the planet**
- **More and More organizations are outsourcing their IT to AWS**
- **The AWS certifications are the most popular IT certifications right now**
- **Top Paid certification according to Forbes**
- **AWS was named as a leader in the "Iaas Magic Quadrant" for the 7th Consecutive Year – Gartner**



What are the Amazon Web Services Global Infrastrcture?

The AWS Cloud spans 54 Availability Zones within 18 geographic Regions around the world.

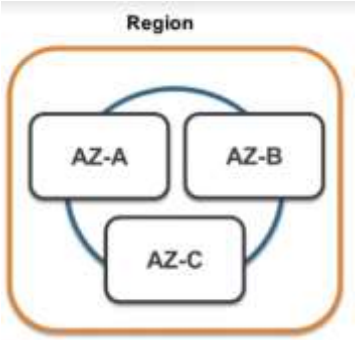
The following table list the AWS region name and code: -

Region Name	Region Code	Region Name	Region Code
US East (N. Virginia)	us-east-1	Asia Pacific (Mumbai)	ap-south-1
US East (Ohio)	us-east-2	Asia Pacific (Seoul)	ap-northeast-2
US West (N. California)	us-west-1	Asia Pacific (Singapore)	ap-southeast-1
US West (Oregon)	us-west-2	Asia Pacific (Sydney)	ap-southeast-2
Canada (Central)	ca-central-1	Asia Pacific (Tokyo)	ap-northeast-1
South America (São Paulo)	sa-east-1	EU (Frankfurt)	eu-central-1
China (Beijing)	cn-north-1	EU (Ireland)	eu-west-1
AWS GoVCloud	us-gov-west-1	EU (London)	eu-west-2

To know the latest information about AWS global infrastrcture

Please Visit Website: https://aws.amazon.com/about-aws/global-infrastructure/

The AWS Cloud infrastructure is built around Regions and Availability Zones ("AZs").



Regions

- **A Region is a physical location in the world where we have multiple Availability Zones. A grouping of AWS data centers within a specific region. Designed to be independent of other regions.**

- **AWS Regions are completely isolated from each other and are in different parts of the world and AWS Regions is**
  – A collection of data centers (Availability Zones or "AZ")
  – Each region has a set number of AZs

– All AZs in a region connected by high-bandwidth
– Cost vary from Region to Region
– Default Region in US East

- An AWS Region is a completely independent entity in a geographical area. There are two more Availability Zones in an AWS Region. Within a region, Availability Zones are connected through low-latency links. Since each AWS Region is isolated from another Region, it provides very high fault tolerance and stability. For launching an EC2 instance, we have to select an AMI within the same region.

Availability Zones

- An Availability Zone is
  – Subset of a Region
  – Physically isolated & independent infrastructure
  – High speed connectivity
  –Low latency
  – Every Region has a minimum of 2 AZs

- An Availability Zone consist of one or more discrete data centers, each with redundant power, networking and connectivity, housed in separate facilities. These Availability Zones offer you the ability to operate production applications and databases which are more highly available, fault tolerant and scalable than would be possible from a single data center
- Minimum two availability zones in a single region for high availability of AWS
- Individual datacenter within an AWS region. A region is made up of multiple datacenters and the fundamental property of AWS is building across the different availability zones and regions.
- Hosting across the regions based on the business promotion

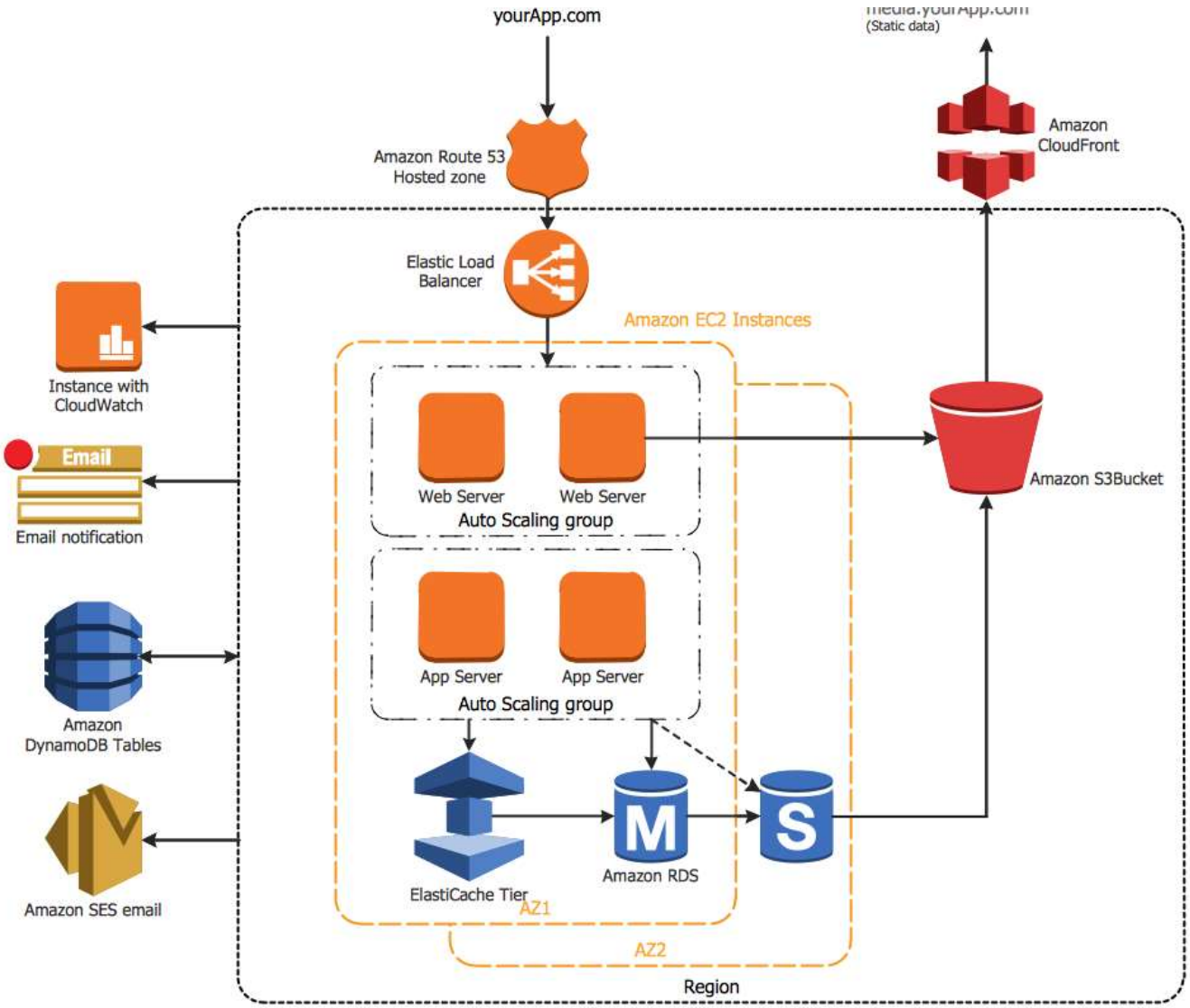
Edge Locations

- Edge Locations are endpoints for AWS which are used for caching content. Typically, this consists of CloudFront, Content Delivery Network (CDN). There are many more edge locations than regions. Currently there are over 100 edge locations. Based on the nearest edge location, customers can communicate and get data
- Locations built to deliver cached data across the world. CloudFront CDN utilizes this service for faster delivery to countries without AWS regions.

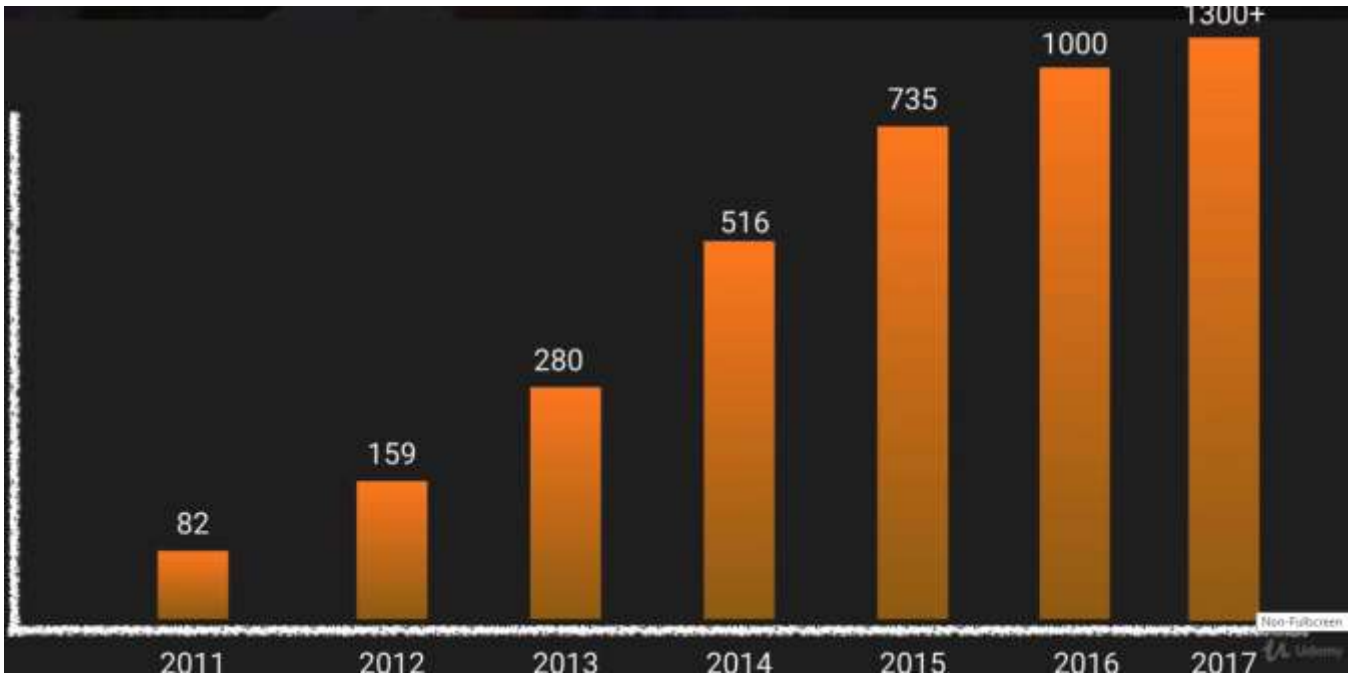
Edge Cache

- Edge Cache is used to store my frequently accessed data in the server

Explain Amazon Web Services Architecture Diagram?

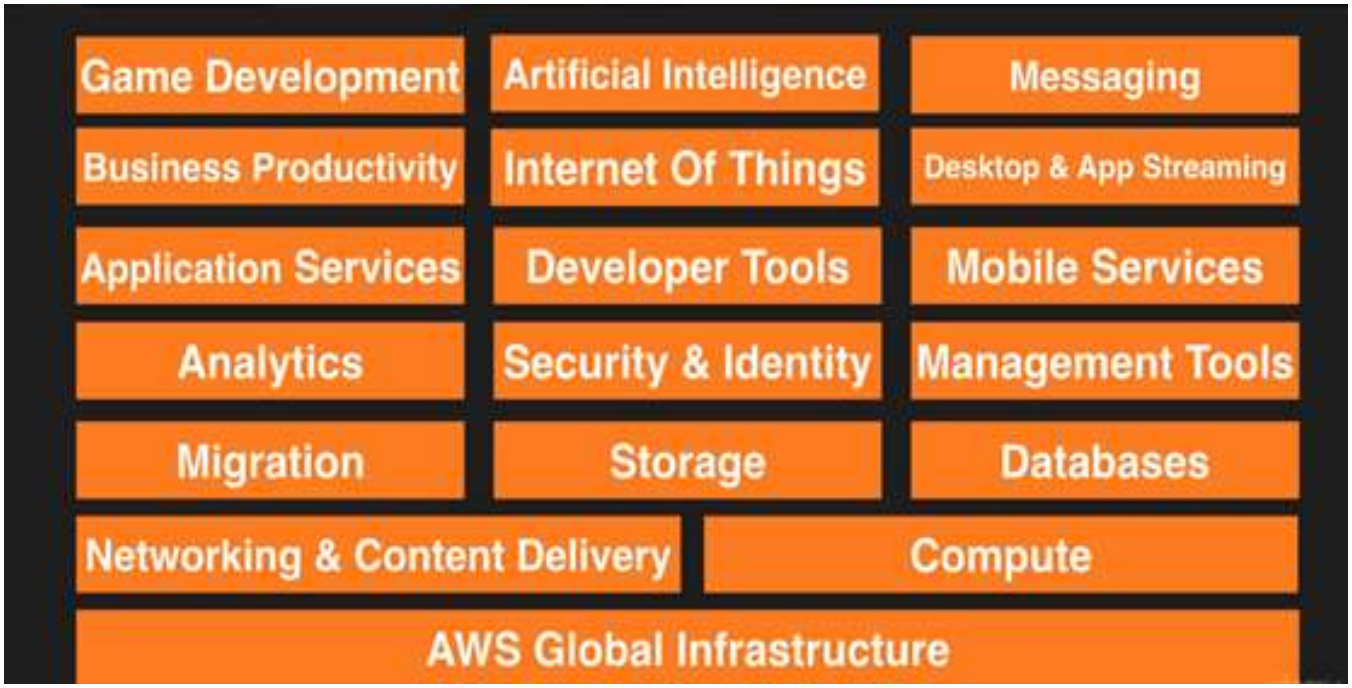


Share the Service Announcements in Amazon Web Services?



Explain Amazon Web Services Product Categories?

Amazon Web Services offers broad set of product categories in AWS platform are –



Compute
It is used to process data on the cloud by making use of powerful processors which serve multiple instances at a time.

Storage
The storage as the name suggests, is used to store data in the cloud, this data can be stored anywhere but content delivery on the other hand is used to cache data nearer to the user so as to provide low latency.

Database
The database domain is used to provide reliable relational and non-relational database instances managed by AWS.

Networking and Content Delivery
It includes services which provide a variety of networking features such as security, faster access etc.

Management Tools
It includes services which can be used to manage and monitor your AWS instances.

Security and Identity
It includes services for user authentication or limiting access to a certain set of audience on your AWS resources.

Application Services
It includes simple services like notifications, emailing and queuing.

Compute

EC2

Amazon Elastic Compute Cloud (EC2) provides resizable compute capacity in the cloud. Provides the virtual server in the AWS Cloud.

EC2 Container Service

Amazon ECS allows you to easily run and manage Docker containers across a cluster of Amazon EC2 instances.

Elastic Beanstalk

AWS Elastic Beanstalk (EBS) is an application container for deploying and managing applications.

Lambda

AWS Lambda is a compute service that runs your code in response to events and automatically manages the compute resources for you.

Elastic Load Balancing

Distributes network traffic across your set of Virtual Servers.

LightSail

Amazon LightSail is the easiest way to get started with AWS for developers who just need virtual private servers. LightSail includes everything you need to launch your project quickly – a virtual machine, SSD-based storage, data transfer, DNS management, and a static IP – for a low, predictable price.

Storage

S3

Amazon Simple Storage Service (S3) can be used to store and retrieve any amount of data.

Glacier

Amazon Glacier is a low-cost storage service that provides secure and durable storage for data archiving and backup.

EFS

Amazon Elastic File System (Amazon EFS) is a file storage service for Amazon Elastic Compute Cloud (Amazon EC2) instances.

Storage Gateway

AWS Storage Gateway securely integrates on-premises IT environments with cloud storage for backup and disaster recovery.

Databases

RDS

Amazon Relational Database Service (RDS) makes it easy to set up, operate, and scale familiar relational databases in the cloud.

Redshift

Amazon Redshift is a fast, fully managed, peta by te- scale data warehouse that makes it cost-effective to analyze all your data using your existing business intelligence tools.

DynamoDB

Amazon DynamoDB is a scalable NoSQL data store that manages distributed replicas of your data for high availability.

Elaticcache

Amazon Elasticache improves application performance by allowing you to retrieve inform ation from an in-memory caching system.

Networking & Content Delivery

VPC

Amazon Virtual Private Cloud (VPC) lets you launch AWS resources in a private, isolated cloud.

Route 53

Amazon Route 53 is a scalable and highly available Domain Name System (DNS) and Domain Name Registration service.

Direct Connect

AWS Direct Connect lets you establish a dedicated network connection from your network to AWS.

CloudFront

Amazon CloudFront provides a way to distribute content to end users with low latency and high data transfer speeds.

Migration

Snowball

Snowball is a petabyte-scale data transport solution that uses secure appliances to transfer large amounts of data into and out of the AWS cloud.

Database Migration Service

AWS Database Migration Service helps you migrate databases to AWS quickly and securely. The source database remains fully operational during the migration, minimizing downtime to applications that rely on the database.

Server Migration Service (SMS)

AWS Server Migration Service (SMS) is an agentless service which makes it easier and faster for you to migrate thousands of on-premises workloads to AWS. AWS SMS allows you to automate, schedule, and track incremental replications of live server volumes, making it easier for you to coordinate large-scale server migrations.

Developer Tools

CodeCommit

AWS CodeCommit is a highly scalable, managed source control service that hosts private Git repositories.

CodeDeploy

AWS CodeDeploy lets you fully automate code deployments.

CodePipeline

AWS CodePipeline is a continuous delivery service that enables you to model, visualize, and automate the steps required to release your software.

Code Build

AWS CodeBuild tool is a fully managed build service that compiles source code, runs tests, and produces software packages that are ready to deploy. With CodeBuild, you don't need to provision, manage, and scale your own build servers.

Management Tools

CloudWatch

Amazon CloudWatch provides monitoring for resources and applications.

CloudFormation

AWS CloudFormation lets you create and update a collection of related AWS resources in a predic table fashion.

CloudTrail

AWS CloudTrail provides increased visibility into user activity by recording API calls made on your account.

OpsWorks

AWS OpsWorks is a DevOps platform for managing applications of any scale or complexity on the AWS cloud.

Service Catalog

AWS Service Catalog allows organizations to manage approved catalogs of IT resources and make them available to employees via a personalized portal.

Config

AWS Config gives you inventory of your AWS resources, lets you audit resource configuration history, and notifies you when
resource configurations change.

Trusted Advisor

AWS Trusted Advisor inspects your AWS environment and finds opportunities to save money, improve system performance and reliability, or help close security gaps.

Security & Identity

IAM

AWS Identity and Access Management (IAM) lets you securely control access to AWS services and resources.

Inspector

Amazon Inspector enables you to analyze the behavior of the applications you run in AWS and helps you to identify potential security issues.

Certificate Manager

AWS Certificate Manager lets you easily provision, manage, and deploy Secure Sockets Layer/ Transport Layer Security (SSL/TLS) certificates for use with AWS services.

Directory Service

AWS Directory Service provides managed directories in the cloud.

WAF

AWS WAF (Web Application Firewall) protects web applications from attack by providing web traffic filtering against common web exploits like SQL injection.

Artifacts

AWS Artifact provides on-demand access to AWS' security and compliance reports and select online agreements.

Analytics

Athena

Amazon Athena is an ETL-like service launched in November 2016. It allows server-less querying of S3 content using standard SQL.

Kinesis

Amazon Kinesis is a cloud-based service for real-time data processing over large, distributed data streams. It streams data in real time with the ability to process thousands of data streams on a per-second basis. The service, designed for real-time apps, allows developers to pull any amount of data, from any number of sources, scaling up or down as needed. It has some similarities in functionality to Apache Kafka.

EMR

Amazon Elastic MapReduce (EMR) Provides a PaaS service delivering Hadoop for running MapReduce queries framework running on the web-scale infrastructure of EC2 and Amazon S3.

Cloud Search

Amazon CloudSearch is a managed service in the AWS Cloud that makes it simple and cost-effective to set up, manage, and scale a search solution for your website or application.

Data Pipeline

AWS Data Pipeline provides reliable service for data transfer between different AWS compute and storage services (e.g., Amazon S3, Amazon RDS, Amazon DynamoDB, Amazon EMR). In other words, this service is simply a data-driven workload management system, which provides a management API for managing and monitoring of data-driven workloads in cloud applications

Elastic Search

Amazon Elasticsearch Service provides fully managed Elasticsearch and Kibana services.

Quick Sight

Amazon QuickSight is a business intelligence, analytics, and visualization tool launched in November 2016. It provides ad-hoc services by connecting to AWS or non-AWS data sources.

Artificial Intelligence

Machine Learning

Amazon Machine Learning is a service that enables you to easily build smart applications.

Lex

Amazon Lex is an AWS service for building conversational interfaces for any applications using voice and text. With Amazon Lex, the same conversational engine that powers Amazon Alexa is now available to any developer, enabling you to build sophisticated, natural language chatbots into your new and existing applications.

Polly

Amazon Polly is a service that turns text into lifelike speech. Amazon Polly enables existing applications to speak as a first-class feature and creates the opportunity for entirely new categories of speech-enabled products, from mobile apps and cars, to devices and appliances.

Rekognition

Amazon Rekognition is a service that makes it easy to add image analysis to your applications. With Rekognition, you can detect objects, scenes, faces; recognize celebrities; and identify inappropriate content in images. You can also search and compare faces. Rekognition's API enables you to quickly add sophisticated deep learning-based visual search and image classification to your applications.

Internet of Things

iOT

AWS IoT is a managed cloud service that lets connected devices easily and securely interact with cloud applications and other devices.

Mobile Services

Mobile Hub

AWS Mobile Hub lets you quickly build, test, and monitor usage of your mobile apps.

Cognito

Amazon Cognito is a simple user identity and data synchronization service that helps you securely manage and synchronize app data for your users across their mobile devices.

Device Farm

AWS Device Farm helps you improve the quality of your Android, Fire OS, and iOS apps by testing them against real phones and tablets in the AWS Cloud.

Mobile Analytics

Amazon Mobile Analytics is a service that lets you easily collect, visualize, and understand app usage data at scale

Pinpoint

Amazon Pinpoint makes it easy to engage your customers by tracking the ways in which they interact with your applications. You can then use this information to create segments based on customer attributes and behaviors, and to communicate with those customers using the channels they prefer, including email, SMS and mobile push.

Application Services

AppStream

Amazon AppStream lets you stream resource intensive applications and games from the cloud to multiple end-user devices

SWF

Amazon Simple Workflow (SWF) coordinates all of the processing steps within an application.

API Gateway

Amazon API Gateway makes it easy to create, maintain, monitor, and secure APIs at any scale.

Elastic Transcoder

Amazon Elastic Transcoder lets you convert your media files in the cloud easily, at low cost, and at scale.

Step Functions

AWS Step Functions makes it easy to coordinate the components of distributed applications and microservices using visual workflows.

Messaging

SNS

Amazon Simple Notification Service (SNS) lets you publish messages to subscribers or other applications.

SQS

Amazon Simple Queue Service (SQS) offers a reliable, highly scalable, hosted queue for storing messages.

Customer Engagement

SES

Amazon Simple Email Service (SES) enables you to send and receive email.

Business Productivity

WorkDocs

Amazon WorkDocs is a fully managed, secure enterprise storage and sharing service with strong administrative controls and feedback capabilities that improve user productivity.

WorkMail

Amazon WorkMail is a managed email and calendaring service that offers strong security controls and support for existing desktop and mobile clients.

Desktop & App Streaming

WorkSpaces

Amazon WorkSpaces is a fully managed desktop computing service in the cloud.

AppStream

Amazon AppStream lets you stream resource intensive applications and games from the cloud to multiple end-user devices.

What are the key components of AWS (Amazon Web Service)?

The key components of AWS are: -

Route 53: A DNS web service

Simple E-mail Service: It allows sending e-mail using RESTFUL API call or via regular SMTP

Identity and Access Management: It provides enhanced security and identity management for your AWS account

Simple Storage Device or (S3): It is a storage device and the most widely used AWS service

Elastic Compute Cloud (EC2): It provides on-demand computing resources for hosting applications. It is very useful in case of unpredictable workloads

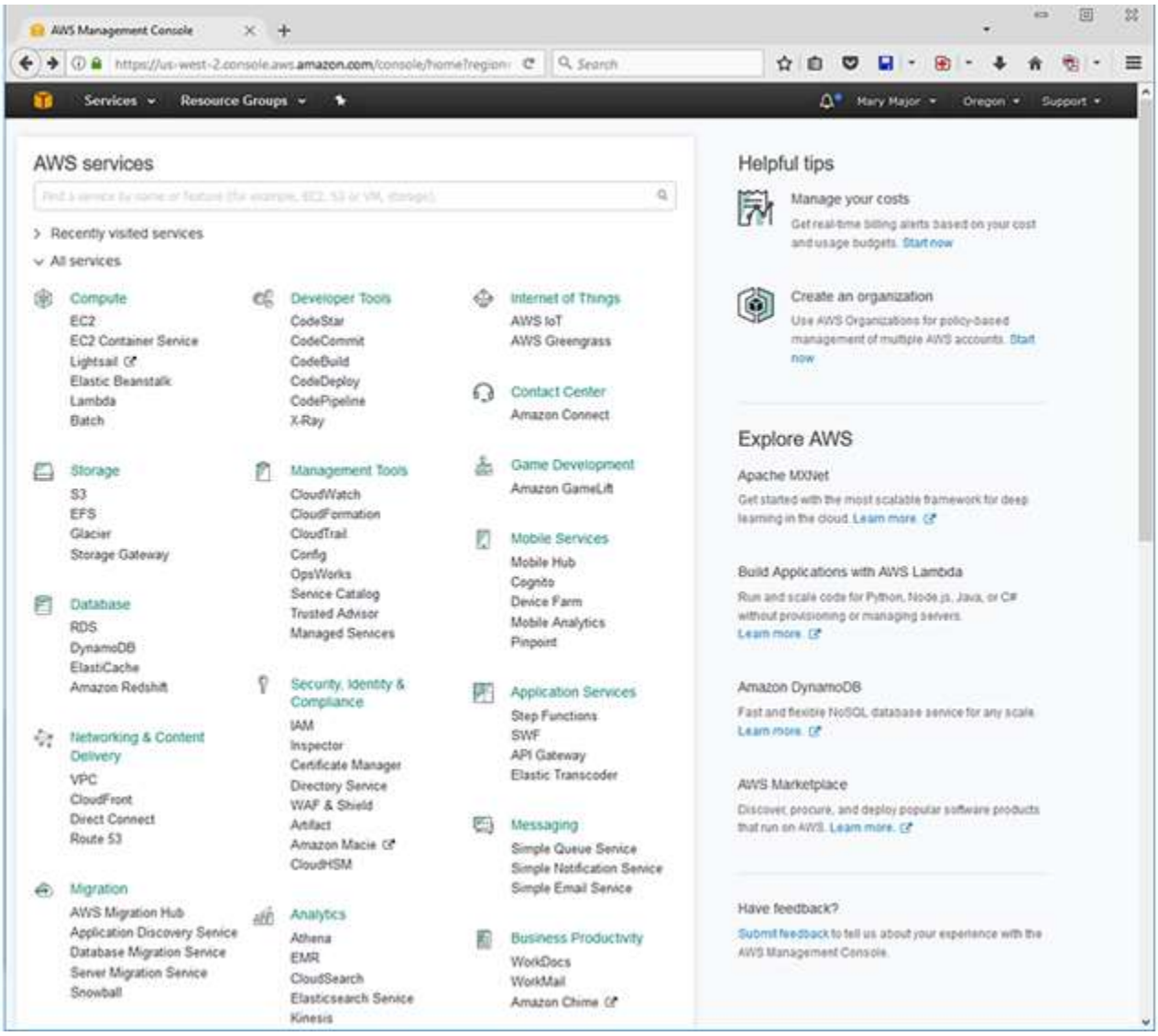
Elastic Block Store (EBS): It provides persistent storage volumes that attach to EC2 to allow you to persist data past the lifespan of a single EC2

CloudWatch: To monitor AWS resources, It allows administrators to view and collect key Also, one can set a notification alarm in case of trouble.

What Is the AWS Management Console?

The AWS Management Console is a web application that comprises and refers to a broad collection of service consoles for managing Amazon Web Services. When you first sign in, you see the console home page.

The home page provides access to each service console as well as an intuitive user interface for exploring AWS and getting helpful tips. Among other things, the individual service consoles offer tools for working with Amazon S3 buckets, launching and connecting to Amazon EC2 instances, setting Amazon CloudWatch alarms, and getting information about your account and about billing.



How Security is implemented in Amazon Web Services?

AWS provides a secure global infrastructure, plus a range of features that you can use to secure your data in the cloud. The following are highlights:

- **Physical access to AWS data centers is strictly controlled, monitored, and audited.**
- **Access to the AWS network is strictly controlled, monitored, and audited.**
- **You can manage the security credentials that enable users to access your AWS account using AWS Identity and Access Management (IAM).**
- **You can create fine-grained permissions to AWS resources and apply them to users or groups of users. You can apply ACL-type permissions on your data and can also use encryption of data at rest.**
- **You can set up a virtual private cloud (VPC), which is a virtual network that is logically isolated from other virtual networks in the AWS cloud.**
- **You can control whether the network is directly routable to the Internet. You control and configure the operating system on your virtual server.**
- **You can set up a security group, which acts as a virtual firewall to control the inbound and outbound traffic for your virtual servers.**
- **You can specify a key pair when you launch your virtual server, which is used to encrypt your login information. When you log in to your virtual server, you must present the private key of the key pair to decrypt the login information.**

What are the top 10 reasons to go with Amazon Web Services?

This might easily be the most popular of the top 10 reasons to go with AWS.

1. **Pricing**
2. **Flexibility & Scalability**
3. **Global Architecture**
4. **PaaS Offerings**
5. **Consistency & Reliability**
6. **Scheduling**
7. **Customization**
8. **Recovery**
9. **Security**
10. **API**

Compare AWS and OpenStack

Criteria	AWS	OpenStack
License	Amazon proprietary	Open Source
Operating System	Whatever cloud administrator provides	Whatever AMIs provided by AWS
Performing repeatable operations	Through Templates	Through text files

What is the difference between Region, Availability Zone and Endpoint in AWS?

In AWS, every region is an independent environment. Within a Region there can be multiple Availability Zones. Every Availability Zone is an isolated area. But there are low-latency links that connect one Availability Zone to another within a region.

An endpoint is just an entry point for a web service. It is written in a URL form. E.g. https://dynamodb.us-east-2.amazonaws.com is an endpoint for Amazon DynamoDB service. Most of the AWS services offer an option to select a regional endpoint for incoming requests. But many services in AWS do not support regions. E.g. IAM. So their endpoints do not have a region.

3.Compute

Amazon EC2 Virtual Servers in the Cloud	**Amazon EC2 Auto Scaling** Scale Compute Capacity to Meet Demand	**AWS Elastic Container Service** Run and Manage Docker Containers
Amazon Elastic Container Service for Kubernetes Run Managed Kubernetes on AWS	**Amazon Elastic Container Registry** Store and Retrieve Docker Images	**Amazon LightSail** Launch and Manage Virtual Private Servers
AWS Batch Run Batch Jobs at Any Scale	**AWS Beanstalk** Run and Manage Web Apps	**AWS Fargate** Run Containers without managing servers on clusters
AWS Lambda Run your code in Response to Events	**AWS Serverless Application Repository** Discover, Deploy, and Publish Serverless Applications Auto Scaling	**VMware Cloud on AWS** Build a Hybrid Cloud without Custom Hardware

EC2

What is EC2?

Amazon Elastic Compute Cloud (EC2) is a web service that provides resizable compute capacity in the cloud. It reduces the time required to obtain and boot new server instances to minutes, allowing you to quickly scale capacity, both up and down, as your computing requirements change.

The Amazon EC2 simple web service interface allows you to obtain and configure capacity with minimal friction. It provides you with complete control of your computing resources and lets you run on Amazon's proven computing environment.

Amazon EC2 reduces the time required to obtain and boot new server instances (called Amazon EC2 instances) to minutes, allowing you to quickly scale capacity, both up and down, as your computing requirements change.

Amazon EC2 changes the economics of computing by allowing you to pay only for capacity that you actually use. It provides developers the tools to build failure resilient applications and isolate themselves from common failure scenarios.

What are the benefits of EC2?

Easier and Faster – Amazon EC2 reduces the time required to obtain and boot new server instances to minutes, allowing you to quickly scale capacity, both up and down, as your computing requirements change.

Elastic and Scalable – Quickly add and subtract resources to applications to meet customer demand and manage costs. Avoid provisioning resources upfront for projects with variable consumption rates or short lifetimes.

High Availability – Amazon EC2 provides developers the tools to build failure resilient applications and isolate themselves from common failure scenarios.

Cost-Effective – Consume only the amount of compute, storage and other IT resources needed. No long-term commitment, minimum spend or up-front investment is required

What are the EC2 options?

The EC2 options are: -

OnDemand – Allows you to pay a fixed rate by hour (or by the second) with no commitment

Reserved – Provide you with a capacity reservation and offer a significant discount on the hourly charge for an instance 1 Year to 3 Year terms.

Spot – Enable you to bid whatever price you want for instance capacity, providing for even greater savings if your applications have flexible start and end times

Dedicated Hosts – Physical EC2 server is dedicated for your use. Dedicated Hosts can help you reduce costs by allowing you to use your existing server-bound software licenses.

In detail

OnDemand

- **Users that want the low cost and flexibility of Amazon EC2 without any up-front payment or long-term commitment**
- **Applications with Short-term, spiky or unpredictable workloads that cannot be interrupted**
- **Applications being developed or tested on Amazon EC2 for the first time**

Reserved

- **Applications with steady state or predictable usage**
- **Applications that require reserved capacity**
- **Users able to make upfront payments to reduce their total computing costs even further: -**
    - **Standard RI's (up to 75% off on demand)**
    - **Convertible RI's (up to 54 % off on demand) capability to change the attributes of the RI as long as the exchange results in the creation of Reserved Instances of equal or greater value**
    - **Scheduled RI's available to launch within the time windows you reserve. This option allows you to match your capacity reservation to a predictable recurring schedule that only fraction of a day, a week, or a month**

Spot

- **Applications that have flexible start and end times**
- **Applications that are only feasible at very low compute prices**
- **Users with urgent computing needs for large amounts of additional capacity**
- **Remember with spot instances of cost involved: -**

  - If you terminate the instance, you pay for the hour

  - If AWS terminates the spot instance, you get the hour it was terminated in for free

Dedicated Hosts

- **Useful for regulatory requirements that may not support multi-tenant virtualization**
- **Great for licensing which does not support multi-tenancy or cloud deployments**
- **Can be purchased On-Demand (hourly)**
- **Can be purchased as a Reservation for up to 70 % off the On-Demand price.**

What are the EC2 Instance Types? How do I remember?

Family	Speciality	Use case
D2	Dense Storage	Fileservers/Data Warehousing/Hadoop
R4	Memory Optimized	Memory Intensive Apps/DBs
M4	General Purpose	Application Servers
C4	Compute Optimized	CPU Intensive Apps/DBs
G2	Graphics Intensive	Video Encoding/ 3D Application Streaming
I2	High Speed Storage	NoSQL DBs, Data Warehousing etc
F1	Field Programmable Gate Array	Hardware acceleration for your code.
T2	Lowest Cost, General Purpose	Web Servers/Small DBs
P2	Graphics/General Purpose GPU	Machine Learning, Bit Coin Mining etc
X1	Memory Optimized	SAP HANA/Apache Spark etc

For easy remember, DR MC GIFT PX

D- Density R-Ram

M- Main choice for general purpose apps C-Compute

G-Graphics I-IOPS F- FPGA T-Cheap general purpose (Think T2 Micro)

P-Graphics (think Pics) X-Extreme Memory

What Is Amazon EC2 instance?

An EC2 instance is a virtual server in Amazon's Elastic Compute Cloud (EC2) for running applications on the Amazon Web Services (AWS) infrastructure.

What is T2 instances?

T2 instances are designed to provide moderate baseline performance and the capability to burst to significantly higher performance as required by your workload.

What is C4 instances?

C4 instances are ideal for compute-bound applications that benefit from high performance processors.

What is Instance Meta-data?

- Used to get information about an instance (Such as Public IP)
- Curl http://168.254.168.254/latest/meta-data/

What are the main features of Amazon Elastic Compute Cloud (EC2)?

There are following main features of Amazon EC2: -

Instance: EC2 has instances in place of real hardware. An instance is a virtual computing environment with memory and compute capability.

Amazon Machine Images (AMI): In EC2 there are preconfigured templates for our instances. These are known as Amazon Machine Images (AMIs). We can create an AMI with the software that we need to start and run the server. It contains Operating System as well as other software.

Configuration: Amazon EC2 supports multiple configurations of CPU, memory, networking and storage capacity for instances.

Security: Amazon EC2 provides security in AWS by using public private key value pairs. We store public key in AWS and keep the private key in a secure place.

Elastic Block Store (EBS): With EC2 we can use EBS to persist the large amount of data.
Availability Zones: We can deploy our applications in multiple geographic locations called Availability Zones in AWS EC2.

Elastic IP Address: We can use static IP addresses for dynamic cloud computing in EC2. These IP addresses help in scaling and maintain high availability of the application in AWS.

Firewall: AWS EC2 also supports the firewall that can be used to specify the protocol, port and source IP range allowed to access instances in EC2.

What are the steps to Create EC2?

We can create EC2 instance using Windows | Linux

STEPS TO CREATE EC2: WINDOWS

1. Choose an Amazon Machine Image (AMI)
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
8. Launch the Instance to check the server is running
9. Login in to the remote machine, using the public IP (e.g., Public DNS ec2-34-238-82-205.compute-1.amazonaws.com)

user: Administrator

Pwd: decrypted PEM FILE. (e.g. J4N))4vakEie(qHN$aVqBFmvUHXW35sE)

STEPS TO CREATE EC2: Linux - Amazon Linux

1. Choose an Amazon Machine Image (AMI)
2. Choose Instance Type
3. Configure Instance
4. Add Storage
5. Add Tags
6. Configure Security Group
7. Review
8. Launch the Instance
9. PEM file generation
10. Upload the Pem file and generate a public key using PUTTYGEN tool.
11. Now you will have. ppk file.
12. Login in the putty with the public ip address and execute the below commands

Upload the ppk file in the auth.

user: ec2-user

sudo su - root

Install the apache: yum install httpd -y

start the apache: service httpd start

13.Launch the IP address in the browser to check the apache running.

Steps to display the output

1. Next, download the static files from any free templates from google (Photovirant template)

2. Go to WinSCP tool
Type Host: 54.85.175.112 Username: ec2-user
Advanced: Load ppk file and click login Now WinSCP gets loaded

3. Login to putty as ec2-user

4. Enter into the root - sudo su - root, Provide the permissions chmod -Rf 777 /var/www/html

5. Put the file into the path - /var/www/html with full permission for the folder and the files.

6.Hit the ip address of the EC2, our application's page will be available.

Now we have successfully launched our application through EC2 Instances. Great Job.

What is the difference between Amazon S3 and Amazon EC2?

Amazon S3 is storage service in cloud. It is used to store large amount of data files. These files can be image files, pdf etc. like static data or these can be dynamic data that is created during runtime.

Amazon EC2 is a remote computing environment running in cloud. We can install our software and operating system on an EC2 instance. We can use it to run our servers like-Web server, Application server etc. So S3 is a storage system where as EC2 is a computing system in AWS.

How does Amazon EC2 works?

Amazon Elastic Compute Cloud (Amazon EC2) is a computing environment provided by AWS. It supports highly scalable computing capacity in AWS.

Instead of buying hardware for servers we can use Amazon EC2 to deploy our applications. So, there is no need to buy and maintain the hardware within our own datacenter. We can just rent the Amazon EC2 servers. Based on our varying needs we can use as few and as many Amazon EC2 instances.

It even provides auto-scaling options in which the instances scale up or down based on the load and traffic spikes. It is easier to deploy applications on EC2. Even we can configure security and networking in Amazon EC2 much easily than our own custom data center.

Explain can you vertically scale an Amazon instance? How?

Yes. This is an incredible characteristic of cloud virtualization and AWS. Spinup is a huge case when compared to the one which you are running with. Let up the instance and separate the root EBS volume from this server and remove. Next, stop your live instance, remove its root volume. Note down the distinctive device ID and attach root volume to your new server and start it again. This is the way to scaling vertically in place.

Explain storage for Amazon EC2 instance?

Amazon EC2 provides many data storage options for your instances. Each option has a unique combination of performance and durability. These storages can be used independently or in combination to suit your requirements.

There are mainly four types of storage provided by AWS: -

Amazon EBS: Its durable, block-level storage volumes that you can attach to a running Amazon EC2 instance. The Amazon EBS volume persists independently from the running life of an Amazon EC2 instance. After an EBS volume is attached to an instance, you can use it like any other physical hard drive. Amazon EBS encryption feature supports encryption feature.

Amazon EC2 Instance Store: Storage disk that is attached to the host computer is referred to as instance store. Instance storage provides temporary block-level storage for Amazon EC2 instances. The data on an instance store volume persists only during the life of the associated Amazon EC2 instance; if you stop or terminate an instance, any data on instance store volumes is lost.

Amazon S3: Amazon S3 provides access to reliable and inexpensive data storage infrastructure. It is designed to make web-scale computing easier by enabling you to store and retrieve any amount of data, at any time, from within Amazon EC2 or anywhere on the web.

Adding Storage: Every time you launch an instance from an AMI, a root storage device is created for that instance. The root storage device contains all the information necessary to boot the instance. You can specify storage volumes in addition to the root device volume when you create an AMI or launch an instance using block device mapping.

What are the Security Best Practices for Amazon EC2?

There are several best practices for secure Amazon EC2. Following are few of them.

- Use AWS Identity and Access Management (IAM) to control access to your AWS resources.
- Restrict access by only allowing trusted hosts or networks to access ports on your instance.
- Review the rules in your security groups regularly, and ensure that you apply the principle of least Privilege — only open up permissions that you require.
- Disable password-based logins for instances launched from your AMI. Passwords can be found or cracked and are a security risk.

Explain Stopping, Starting, and Terminating an Amazon EC2 instance?

Stopping and Starting an instance: When an instance is stopped, the instance performs a normal shutdown and then transitions to a stopped state. All of its Amazon EBS volumes remain attached, and you can start the instance again at a later time. You are not charged for additional instance hours while the instance is in a stopped state.

Terminating an instance: When an instance is terminated, the instance performs a normal shutdown, then the attached Amazon EBS volumes are deleted unless the volume's deleteOnTermination attribute is set to false. The instance itself is also deleted, and you can't start the instance again at a later time.

What are regions and availability zones in Amazon EC2? Explain in brief?

Amazon EC2 is hosted in multiple locations world-wide. These locations are composed of regions and Availability Zones. Each region is a separate geographic area. Each region has multiple, isolated locations known as Availability Zones.

Each region is completely independent. Each Availability Zone is isolated, but the Availability Zones in a region are connected through low-latency links. The following diagram illustrates the relationship between regions and Availability Zones.

What is Regions and Endpoints in AWS?

To reduce data latency in your applications, most Amazon Web Services products allow you to select a regional endpoint to make your requests. An endpoint is a URL that is the entry point for a web service. For example, https://dynamodb.us-west-2.amazonaws.com is an entry point for the Amazon DynamoDB service.

Some services, such as IAM, do not support regions; their endpoints therefore do not include a region.

A few services, such as Amazon EC2, let you specify an endpoint that does not include a specific region, for example, https://ec2.amazonaws.com. In that case, AWS routes the endpoint to us-east-1.

How to find your regions and Availability Zones using the Amazon EC2 CLI?

Use the ec2-describe-regions command as follows to describe your regions.

PROMPT> ec2-describe-regions

REGION us-east-1 ec2.us-east-1.amazonaws.com

REGION ap-northeast-1 ec2.ap-northeast-1.amazonaws.com

REGION ap-southeast-1 ec2.ap-southeast-1.amazonaws.com

What is a Placement Group in EC2?

AWS provides an option of creating a Placement Group in EC2 to logically group the instances within as single Availability Zone.

We get the benefits of low network latency and high network throughput by using a Placement Group. Placement Group is a free option as of now. When we stop an instance, it will run in same Placement Group in restart at a later point of time. The biggest limitation of Placement Group is that we cannot add Instances from multiple availability zones to one Placement Group.

What are the best practices for Amazon EC2?

To get the maximum benefit from and satisfaction with Amazon EC2. There are mainly four best practices.

- **Security and Network Best Practices**
- **Storage**
- **Resource Management**
- **Backup and Recovery**

What is the underlying Hypervisor for EC2?

Xen

How you're charged in Amazon EC2? Explain in detail?

- Charges varies upon AMIs backed and storage volumes.
- AMIs backed by instance storage charged for: AMI storage + Instance usage
- AMIs backed by Amazon EBS storage charged for: Volume storage + Usage in addition to the AMI + instance usage
- When an Amazon EBS-backed instance is stopped, you are not charged for instance usage, but you are still charged for volume storage.
- AWS charges a full instance hour for every transition from a stopped state to a running state, even if we transition the instance multiple times within a single hour.
- For example: if hourly instance charge for your instance is $0.10 and if you were to run that instance for one hour without stopping it, you would be charged $0.10. If you stopped and restarted that instance twice during that hour, then you would be charged $0.30 for that hour of usage (the initial $0.10, plus 2 x $0.10 for each restart).

What is the difference between scalability and elasticity?

Scalability is a characteristic of cloud computing through which increasing workload can be handled by increasing in proportion the amount of resource capacity. It allows the architecture to provide on demand resources if the requirement is being raised by the traffic.

Whereas, elasticity is being one of the characteristic provide the concept of commissioning and decommissioning of large amount of resource capacity dynamically. It is measured by the speed by which the resources are coming on demand and the usage of the resources.

What is an Elastic IP Address?

Amazon provides an Elastic IP Address with an AWS account. An Elastic IP address is a public and static IP address based on IPv4 protocol. It is designed for dynamic cloud computing. This IP address is reachable from the Internet.

If we do not have a specific IP address for our EC2 instance, then we can associate our instance to the Elastic IP address of our AWS account. Now our instance can communicate on the Internet with this Elastic IP Address.

What is ElastiCache?

ElastiCache is a web service that makes it easy to set up, manage, and scale distributed in-memory cache environments in the cloud.

What is the use of Amazon ElastiCache?

Amazon ElastiCache is mainly used for improving the performance of web applications by caching the information that is frequently accessed. ElastiCache webservice provides very fast access to the information by using in-memory caching.

Behind the scenes, ElastiCache supports open source caching platforms like-Memcached and Redis. We do not have to manage separate caching servers with ElastiCache. We can just add critical pieces of data in ElastiCache to provide very low latency access to applications that need this data very frequently.

What is Elastic File System?

- Supports the Network File System Version 4 (NFSv4) protocol
- You only pay for the storage you use (No Pre-Provisioning is required)
- Can Scale up to the Petabytes
- Can support thousands of concurrent NFS connections
- Data is stored across Multiple AZ's within a region
- Read After Write Consistency

What will happen when we reboot an Amazon EC2 instance?

When we reboot an Amazon EC2 instance, it reboots like computer. There is no effect on hard disk. It reboots with the latest configurations settings that were present just before the reboot. Once you terminate the instance, then only billing stops. You can reboot an instance multiple times but the billing will continue.

What are the steps to change the root EBS device of an Amazon EC2 instance?

We can follow these steps to change the root EBS device of an EC2 instance: Stop Amazon EC2 instance, detach root EBS volume, attach another EBS volume, Start Amazon EC2 instance

What is Public Key Credentials and how to install it?

Amazon EC2 uses public–key cryptography to encrypt and decrypt login information. Public–key cryptography uses a public key to encrypt a piece of data, such as a password, then the recipient uses the private key to decrypt the data. The public and private keys are known as a key pair.

After configuring the AMI to prevent logging in using a password, you must make sure users can log in using another mechanism.

How to disable Password-Based Logins for Root in Amazon EC2 Instance?

Using a fixed root password for a public AMI is a security risk that can quickly become known. Even relying on users to change the password after the first login opens a small window of opportunity for potential abuse.

Following are the steps to disable password-based remote logins for the root user.

1.Open the /etc/ssh/sshd_config file with a text editor and locate the following line:
#PermitRootLogin yes

2. Change the line to:
PermitRootLogin without-password

The location of this configuration file might differ for your distribution.

While connecting to your instance what are the possible connection issues one might face?

The possible connection errors one might encounter while connecting instances are

- **Connection timed out**
- **User key not recognized by the server**
- **Host key not found, permission denied**
- **Unprotected private key file**
- **Server refused our key or No supported authentication method available**
- **Error using MindTerm on Safari Browser**
- **Error using Mac OS X RDP Client**

What are the main uses of Amazon Elastic Compute Cloud (EC2)?

Amazon EC2 provides scalable computing resources for creating a software infrastructure. It is very easy to deploy application in Amazon E2.

Main uses of EC2 are: -
Easy Configuration: We can easily configure our servers in EC2 and manage the capacity. Control: EC2 provides complete control of computing resources even to developers. A user can run the EC2 environment according to his/her system's needs.

Fast Reboot: It is very fast to reboot the instances in EC2. It reduces the overall deployment and development time.

Scalability: In EC2 we can create a highly scalable environment based on the load that is expected on our application. Resilient: It is very easy to create and terminate servers in EC2. Due to this we can develop resilient applications in EC2.

How you will find out the instance id from within an ec2 machine?

wget -q -O – http://instance-data/latest/meta-data/instance-id

If you need programmatic access to the instance ID from within a script

die() { status=$1; shift; echo "FATAL: $*"; exit $status; }

EC2_INSTANCE_id="`wget -q -O – http://instance-data/latest/meta-data/instance-id || die \"wget instance-id has failed: $?\"`"

Is it possible to use AWS as a web host? What are the way of using AWS as a web host?

Yes, it is completely possible to host websites on AWS in 2 ways:

- **Easy –** S3 (Simple Storage Solution) is a bucket storage solution that lets you serve static content e.g. images but has recently been upgraded so you can use it to host flat .html files and your site will get served by a default Apache installation with very little configuration on your part (but also little control).
- **Trickier –** You can use EC2 (Elastic Compute Cloud) and create a virtual Linux instance then install Apache/NGinx (or whatever) on that to give you complete control over serving whatever/however you want. You use SecurityGroups to enable/disable ports for individual machines or groups of them.

How to access/ping a server located on AWS?

Using UI:

In your security group:

- Click the inbound tab
  Create a custom ICMP rule
  Select echo request
  Use range 0.0.0.0/0 for everyone or lock it down to specific IPs
  Apply the changes
  and you'll be able to ping.
- Using cmd: To do this on the command line you can run:
- ec2-authorize -P icmp -t -1:-1 -s 0.0.0.0/0

What are the 4 level of AWS premium support?

Basic, Developer, Business & Enterprise

What does the following command do with respect to the Amazon EC2 security groups?

ec2-create-group CreateSecurityGroup

A. Groups the user created security groups into a new group for easy access.

B. Creates a new security group for use with your account.

C. Creates a new group inside the security group.

D. Creates a new rule inside the security group.

Answer B

Explanation: A Security group is just like a firewall, it controls the traffic in and out of your instance. In AWS terms, the inbound and outbound traffic. The command mentioned is pretty straight forward, it says create security group, and does the same. Moving along, once your security group is created, you can add different rules in it. For example, you have an RDS instance, to access it, you have to add the public IP address of the machine from which you want access the instance in its security group.

You have a video trans-coding application. The videos are processed according to a queue. If the processing of a video is interrupted in one instance, it is resumed in another instance. Currently there is a huge back-log of videos which needs to be processed, for this you need to add more instances, but you need these instances only until your backlog is reduced. Which of these would be an efficient way to do it?

You should be using an On-Demand instance for the same. Why? First of all, the workload has to be processed now, meaning it is urgent, secondly you don't need them once your backlog is cleared, therefore Reserved Instance is out of the picture, and since the work is urgent, you cannot stop the work on your instance just because the spot price spiked, therefore Spot Instances shall also not be used. Hence On-Demand instances shall be the right choice in this case.

You have a distributed application that periodically processes large volumes of data across multiple Amazon EC2 Instances. The application is designed to recover gracefully from Amazon EC2 instance failures. You are required to accomplish this task in the most cost-effective way. Which of the following will meet your requirements?

A. Spot Instances

B. Reserved instances

C. Dedicated instances

D. On-Demand instances

Answer: A

Explanation: Since the work we are addressing here is not continuous, a reserved instance shall be idle at times, same goes with On Demand instances. Also it does not make sense to launch an On Demand

instance whenever work comes up, since it is expensive. Hence Spot Instances will be the right fit because of their low rates and no long-term commitments.

How is stopping and terminating an instance different from each other?

Starting, stopping and terminating are the three states in an EC2 instance, let's discuss them in detail:
Stopping and Starting an instance: When an instance is stopped, the instance performs a normal shutdown and then transitions to a stopped state. All of its Amazon EBS volumes remain attached, and you can start the instance again at a later time. You are not charged for additional instance hours while the instance is in a stopped state.

Terminating an instance: When an instance is terminated, the instance performs a normal shutdown, then the attached Amazon EBS volumes are deleted unless the volume's deleteOnTermination attribute is set to false. The instance itself is also deleted, and you can't start the instance again at a later time.

If I want my instance to run on a single-tenant hardware, which value do I have to set the instance's tenancy attribute to?

A. Dedicated
B. Isolated
C. One
D. Reserved
Answer A
Explanation: The Instance tenancy attribute should be set to Dedicated Instance. The rest of the values are invalid.

When will you incur costs with an Elastic IP address (EIP)?

A. When an EIP is allocated.
B. When it is allocated and associated with a running instance.
C. When it is allocated and associated with a stopped instance.
D. Costs are incurred regardless of whether the EIP is associated with a running instance.
Answer C
Explanation: You are not charged, if only one Elastic IP address is attached with your running instance. But you do get charged in the following conditions:
When you use more than one Elastic IPs with your instance.
When your Elastic IP is attached to a stopped instance.
When your Elastic IP is not attached to any instance.

How is a Spot instance different from an On-Demand instance or Reserved Instance?

First of all, let's understand that Spot Instance, On-Demand instance and Reserved Instances are all models for pricing. Moving along, spot instances provide the ability for customers to purchase compute capacity with no upfront commitment, at hourly rates usually lower than the On-Demand rate in each region.

Spot instances are just like bidding, the bidding price is called Spot Price. The Spot Price fluctuates based on supply and demand for instances, but customers will never pay more than the maximum price they have specified. If the Spot Price moves higher than a customer's maximum price, the customer's EC2 instance will be shut down automatically.

But the reverse is not true, if the Spot prices come down again, your EC2 instance will not be launched automatically, one has to do that manually. In Spot and On demand instance, there is no commitment for the duration from the user side, however in reserved instances one has to stick to the time period that he has chosen.

Are the Reserved Instances available for Multi-AZ Deployments?

A. Multi-AZ Deployments are only available for Cluster Compute instances types

B. Available for all instance types

C. Only available for M3 instance types

D. Not Available for Reserved Instances

Answer B

Explanation: Reserved Instances is a pricing model, which is available for all instance types in EC2.

How to use the processor state control feature available on the c4.8xlarge instance?

The processor state control consists of 2 states:

The C state – Sleep state varying from c0 to c6. C6 being the deepest sleep state for a processor

The P state – Performance state p0 being the highest and p15 being the lowest possible frequency.

Now, why the C state and P state. Processors have cores, these cores need thermal headroom to boost their performance. Now since all the cores are on the processor the temperature should be kept at an optimal state so that all the cores can perform at the highest performance.

Now how will these states help in that? If a core is put into sleep state it will reduce the overall temperature of the processor and hence other cores can perform better. Now the same can be synchronized with

other cores, so that the processor can boost as many cores it can by timely putting other cores to sleep, and thus get an overall performance boost.

Concluding, the C and P state can be customized in some EC2 instances like the c4.8xlarge instance and thus you can customize the processor according to your workload.
How to do it? You can refer this tutorial for the same.

What kind of network performance parameters can you expect when you launch instances in cluster placement group?

The network performance depends on the instance type and network performance specification, if launched in a placement group you can expect up to
10 Gbps in a single-flow,
20 Gbps in multiflow i.e., full duplex
Network traffic outside the placement group will be limited to 5 Gbps(full duplex).

To deploy a 4-node cluster of Hadoop in AWS which instance type can be used?

First let's understand what actually happens in a Hadoop cluster, the Hadoop cluster follows a master slave concept. The master machine processes all the data, slave machines store the data and act as data nodes.

Since all the storage happens at the slave, a higher capacity hard disk would be recommended and since master does all the processing, a higher RAM and a much better CPU is required. Therefore, you can select the configuration of your machine depending on your workload.

For e.g. – In this case c4.8xlarge will be preferred for master machine whereas for slave machine we can select i2.large instance. If you don't want to deal with configuring your instance and installing hadoop cluster manually, you can straight away launch an Amazon EMR (Elastic Map Reduce) instance which automatically configures the servers for you. You dump your data to be processed in S3, EMR picks it from there, processes it, and dumps it back into S3.

Where do you think an AMI fits, when you are designing an architecture for a solution?

AMIs (Amazon Machine Images) are like templates of virtual machines and an instance is derived from an AMI. AWS offers pre-baked AMIs which you can choose while you are launching an instance, some AMIs are not free, therefore can be bought from the AWS Marketplace. You can also choose to create your own custom AMI which would help you save space on AWS. For example, if you don't need a set of software on your installation, you can customize your AMI to do that. This makes it cost efficient, since you are removing the unwanted things.

How do you choose an Availability Zone?

Let's understand this through an example, consider there's a company which has user base in India as well as in the US.

Let us see how we will choose the region for this use case:

Regions	• Mumbai/N Virginia
Instance Type (Reserved Instance)	• e.g. amazon ec2- m4.4xlarge 16(vCPU), 64 GB RAM
Pricing(1 Year)	• Mumbai - $691/monthly - $0.9 hourly • N Virginia - $480/monthly - $0.6 hourly
Latency	• From USA to India - Low • From India to USA - High

So, with reference to the above figure the regions to choose between are, Mumbai and North Virginia. Now let us first compare the pricing, you have hourly prices, which can be converted to your per month figure. Here North Virginia emerges as a winner. But, pricing cannot be the only parameter to consider. Performance should also be kept in mind hence, let's look at latency as well. Latency basically is the time that a server takes to respond to your requests i.e., the response time. North Virginia wins again!

So, concluding, North Virginia should be chosen for this use case.

Is one Elastic IP address enough for every instance that I have running?

Depends! Every instance comes with its own private and public address. The private address is associated exclusively with the instance and is returned to Amazon EC2 only when it is stopped or terminated. Similarly, the public address is associated exclusively with the instance until it is stopped or terminated. However, this can be replaced by the Elastic IP address, which stays with the instance as long as the user doesn't manually detach it. But what if you are hosting multiple websites on your EC2 server, in that case you may require more than one Elastic IP address.

EBS

What is EBS? What are the EBS Volume Types?

Amazon EBS allows you to create storage volumes and attach them to Amazon EC2 instances. Once attached, you can create a file system on top of these volumes, run a database, or use them in any other way you would use a block device. Amazon EBS volumes are replaced in a specific availability zone, where they are automatically replicated to protect you from the failure of a single component.

- EBS consists of five volume types: -
    - SSD, General Purpose - GP2 (Up to 10,000 IOPS)
    - SSD, Provisioned IOPS -IO1 (More than 10,000 IOPS)
    - HDD, Throughput Optimized - ST1 - Frequently accessed workloads
    - HDD, Cold - SC1 - Less frequently accessed data
    - HDD, Magnetic - Standard - cheap, infrequently accessed storage
- You cannot mount 1 EBS volume to multiple EC2 instances instead use EFS
- Termination Protection is turned off by default, you must turn it on
- On an EBS-backed instance, the default action is for the root EBS volume to be deleted when the instance is terminated
- Root Volumes cannot be encrypted by default, you need a third-party tool (such as bit locker etc.,) to encrypt the root volume

Explain types of storage for the Root Device and difference between them?

There are 2 types of storage for the Root Device, as either backed by Amazon EBS or backed by Instance store. he former means that the root device for an instance launched from the AMI is an Amazon EBS volume created from an Amazon EBS snapshot.

This section summarizes the important differences between the two types of AMIs. The following table provides a quick summary of these differences.

Characteristic	Amazon EBS-Backed	Amazon Instance Store-Backed
Boot time	Usually less than 1 minute	Usually less than 5 minutes
Size limit	16 TiB	10 GiB
Root device volume	Amazon EBS volume	Instance store volume

Data persistence:

By default, the root volume is deleted when the instance terminates. * Data on any other Amazon EBS volumes persists after instance termination by default. Data on any instance store volumes persists only during the life of the instance.Data on any instance store volumes persists only during the life of the instance. Data on any Amazon EBS volumes persists after instance termination by default.

Upgrading:

The instance type, kernel, RAM disk, and user data can be changed while the instance is stopped. Instance attributes are fixed for the life of an instance.

Charges:

You're charged for instance usage, Amazon EBS volume usage, and storing your AMI as an Amazon EBS snapshot. You're charged for instance usage and storing your AMI in Amazon S3.

AMI creation/bundling Uses a single command/call

Requires installation and use of AMI tools Stopped State:

Can be placed in stopped state where instance is not running, but the root volume is persisted in Amazon EBS

Cannot be in stopped state:

instances are running or terminated

How do you pass custom environment variable on Amazon Elastic Beanstalk (AWS EBS)?

As a heads up to anyone who uses the .ebextensions/*.config way: nowadays you can add, edit and remove environment variables in the Elastic Beanstalk web interface.
The variables are under Configuration

Software Configuration:

What are the benefits of EBS vs. instance-store?

- EBS backed instances can be set so that they cannot be (accidentally) terminated through the API.

- EBS backed instances can be stopped when you're not using them and resumed when you need them again (like pausing a Virtual PC), at least with my usage patterns saving much more money than I spend on a few dozen GB of EBS storage.

- EBS backed instances don't lose their instance storage when they crash (not a requirement for all users, but makes recovery much faster)

- You can dynamically resize EBS instance storage.
- You can transfer the EBS instance storage to a brand-new instance (useful if the hardware at Amazon you were running on gets flaky or dies, which does happen from time to time)

- It is faster to launch an EBS backed instance because the image does not have to be fetched from S3.

Some of the Amazon EC instances types provide the option of using a directly attached block-device storage. This kind of storage is known as Instance Store. In other Amazon EC2 instances, we have to attach an Elastic Block Store (EBS).

Persistence: The main difference between Instance Store and EBS is that in Instance Store data is not persisted for long-term use. If the Instance terminates or fails, we can lose Instance Store data. Any data stored in EBS is persisted for longer duration. Even if an instance fails, we can use the data stored in EBS to connect it to another EC2 instance.

Encryption: EBS provides a full-volume encryption of data stored in it. Whereas Instance Store is not considered good for encrypting data.

Differences

- Instance Store Volumes are sometimes called Ephemeral Storage
- Instance store volumes cannot be stopped. If the underlying host fails, you will lose your data
- EBS backed instances can be stopped. You will not lose the data on this instance if it is stopped.
- You can reboot both, you will not lose your data
- By default, both ROOT volumes will be deleted on termination, however with EBS volumes, you can tell AWS to keep the root device volume

What is Snapshots?

A Snapshot is created by copying the data of a volume to another location at a specific time. We can even replicate same Snapshot to multiple availability zones.

How to create Snapshots of Root Device Volumes?

To create a snapshot for Amazon EBS volumes that serve as root devices, you should stop the instance before talking the snapshot.

How can I take a Snapshot of a RAID Array?

Problem - Take a snapshot excludes data held in the cache by applications and the OS. This tends not to matter on a single volume, however using multiple volumes in a RAID Array, this can be a problem due to inter dependencies of the array.

Solution:

- Take an application with consistent snapshot
- Stop the application from writing to disk
- Flush all caches to the disk

We can do this by

- Freeze the file system
- Unmount the RAID Array
- Shutting down the associated EC2 instance

What is the difference between Volume and Snapshot in Amazon Web Services?

In Amazon Web Services, a Volume is a durable, block level storage device that can be attached to a single EC2 instance. In plain words it is like a hard disk on which we can write or read from.

A Snapshot is created by copying the data of a volume to another location at a specific time. We can even replicate same Snapshot to multiple availability zones. So, Snapshot is a single point in time view of a volume. We can create a Snapshot only when we have a Volume. Also, from a Snapshot we can create a Volume. In AWS, we have to pay for storage that is used by a Volume as well as the one used by Snapshots.

Differences

- Volumes exist on EBS - Virtual Hard Disk
- Snapshot exist on S3
- You can take a snapshot of a volume, this will store that volume on S3
- Snapshots are point in time copies of volumes
- Snapshots are incremental, this means that only blocks that have changed since your last snapshots are moved to S3
- If this is your first snapshot, it may take some time to create
- Snapshots of encrypted volumes are encrypted automatically
- Volumes restored from encrypted snapshots are encrypted automatically
- You can share snapshots, but only if they are unencrypted
- These snapshots can be shared with other AWS accounts or made public

What is the difference between Elastic Beanstalk and CloudFormation?

Elastic Beanstalk automatically handles the deployment, from capacity provisioning, load balancing, auto-scaling to application health monitoring based on the code you upload it.
Cloud Formation is an automated provisioning engine to deploy entire cloud environments via JSON.

How you will change the root EBS device of my amazon EC2 instance?

- Stop the instance.
- Detach the root EBS volume.
- Attach the alternate EBS volume (as the root e.g. /dev/sda1)
- Start the instance.
- This presupposes that your alternate EBS volume is bootable, of course – it has to contain the bootable OS image.

Which of the following services you would not use to deploy an app?

A. Elastic Beanstalk
B. Lambda
C. Opsworks
D.CloudFormation
Answer B

Explanation: Lambda is used for running server-less applications. It can be used to deploy functions triggered by events. When we say serverless, we mean without you worrying about the computing resources running in the background. It is not designed for creating applications which are publicly accessed.

How does Elastic Beanstalk apply updates?

A. By having a duplicate ready with updates before swapping.

B. By updating on the instance while it is running

C. By taking the instance down in the maintenance window

D. Updates should be installed manually

Answer A

Explanation: Elastic Beanstalk prepares a duplicate copy of the instance, before updating the original instance, and routes your traffic to the duplicate instance, so that, incase your updated application fails, it will switch back to the original instance, and there will be no downtime experienced by the users who are using your application.

How is AWS Elastic Beanstalk different than AWS OpsWorks?

AWS Elastic Beanstalk is an application management platform while OpsWorks is a configuration management platform.

BeanStalk is an easy to use service which is used for deploying and scaling web applications developed with Java, .Net, PHP, Node.js, Python, Ruby, Go and Docker. Customers upload their code and Elastic Beanstalk automatically handles the deployment. The application will be ready to use without any infrastructure or resource configuration.

In contrast, AWS Opsworks is an integrated configuration management platform for IT administrators or DevOps engineers who want a high degree of customization and control over operations.

What happens if my application stops responding to requests in beanstalk?

AWS Beanstalk applications have a system in place for avoiding failures in the underlying infrastructure. If an Amazon EC2 instance fails for any reason, Beanstalk will use Auto Scaling to automatically launch a new instance. Beanstalk can also detect if your application is not responding on the custom link, even though the infrastructure appears healthy, it will be logged as an environmental event (e.g a bad version was deployed) so you can take an appropriate action.

Amazon Machine Image

What is Amazon Machine Image?

An Amazon Machine Image (AMI) is a template that contains a software configuration (for example, an operating system, an application server, and applications). From an AMI, we launch an instance, which is a copy of the AMI running as a virtual server in the cloud. We can launch multiple instances of an AMI.

AMI's are regional. You can only launch an AMI from the region in which it is stored. However, you can copy AMI's to other regions using the console, command line or the Amazon EC2 API

What does an AMI include?

An AMI includes the following things

- A template for the root volume for the instance
- Launch permissions decide which AWS accounts can avail the AMI to launch instances
- A block device mapping that determines the volumes to attach to the instance when it is launched

What is the relation between Instance and AMI?

AMI can be elaborated as Amazon Machine Image, basically, a template consisting software configuration part. For example, an OS, applications, application server. If you start an instance, a duplicate of the AMI in a row as an unspoken attendant in the cloud.

We can launch different types of instances from a single AMI. An instance type essentially determines the hardware of the host computer used for your instance. Each instance type offers different compute and memory capabilities.

After we launch an instance, it looks like a traditional host, and we can interact with it as we would any computer. We have complete control of our instances; we can use sudo to run commands that require root privileges.

Amazon Web Services provides several ways to access Amazon EC2, like web-based interface, AWS Command Line Interface (CLI) and Amazon Tools for Windows Powershell. First you need to signed up for an AWS account and you can access Amazon EC2.

Amazon EC2 provides a Query API. These requests are HTTP or HTTPS requests that use the HTTP verbs GET or POST and a Query parameter named Action.

How to determine the Root Device type of your AMI?

We can determine the Root Device type of AMI using following 2 methods.

Method 1: Following are the steps to determine the Root Device type of an AMI using the console

1.1 Open the Amazon EC2 console

1.2 In the navigation pane, click AMIs, and select the AMI

1.3 Check the value of Root Device Type in the Details tab as follows

1.3.1 If the value is ebs, this is an Amazon EBS-backed AMI

1.3.2 If the value is instance store, this is an instance store-backed AMI

Method 2: Following are the steps to determine the root device type of an AMI using the command line

We can use one of the following commands.

2.1 describe-images (AWS CLI)

2.2 Get-EC2Image (AWS Tools for Windows PowerShell)

What is the size limit for Amazon EC2 instance store-backed AMIs and Amazon EBS-backed AMIs?

- All AMIs are categorized as either backed by Amazon EBS or backed by instance store.
- Backed by Amazon EBS – means that the root device for an instance launched from the AMI is an Amazon EBS volume created from an Amazon EBS snapshot.
- Backed by instance store – means that the root device for an instance launched from the AMI is an instance store volume created from a template stored in Amazon S3.
- Root device size limit for –
    - Amazon EBS – Backed is 16 TiB
    - Amazon Instance Store-Backed is 10 GiB

What is shared AMI?

A shared AMI is an AMI that a developer created and made available for other developers to use.
One of the easiest ways to get started with Amazon EC2 is to use a shared AMI that has the components you need and then add custom content. You can also create your own AMIs and share them with others.

Note: Use a shared AMI at your own risk. Amazon can't vouch for the integrity or security of AMIs shared by other Amazon EC2 users. AWS recommends that you get an AMI from a trusted source.

How to update AMI tools at Boot Time?

AWS recommends that your AMIs download and upgrade the Amazon EC2 AMI creation tools during startup. This ensures that new AMIs based on your shared AMIs have the latest AMI tools.

For Amazon Linux, add the following to /etc/rc.local:

```
# Update the Amazon EC2 AMI tools
echo " + Updating EC2 AMI tools"
yum update -y aws-amitools-ec2
echo " + Updated EC2 AMI tools"
```

How to create your own Amazon Machine Image (AMI)?

You can customize an instance that is launched from a public AMI and then save that configuration as a custom AMI for your own use. Instances that you launch from your AMI use all the customizations that you've made.

AWS Lambda

What is AWS Lambda?

AWS Lambda is a service from Amazon to run a specific piece of code in Amazon cloud, without provisioning any server. So, there is no effort involved in administration of servers. In AWS Lambda, we are not charged until our code starts running. Therefore, it is very cost-effective solution to run code.

AWS Lambda can automatically scale our application when the number of requests to run the code increases. So, we do not have to worry about scalability of application to use AWS Lambda.

AWS Lambda is a compute service where you can upload code and create Lambda function. AWS Lambda takes care of provisioning and managing the server that you use to run the code. You don't have to worry about Operating System, Patching, Scaling, etc.,

You can use Lambda in the following ways: -

- As an event-driven compute service where AWS Lambda runs your code in response to events. These events could be changes to data in an Amazon S3 bucket or an Amazon DynamoDB table.
- As a compute service to run your code in response to HTTP requests using Amazon API Gateway or API calls mad using AWS SDKs.

What is a Serverless application in AWS?

In AWS, we can create applications based on AWS Lambda. These applications are composed of functions that are triggered by an event.

These functions are executed by AWS in cloud. But we do not have to specify/buy any instances or server for running these functions. An application created on AWS Lambda is called Serverless application in AWS.

How will you manage and run a serverless application in AWS?

We can use AWS Serverless Application Model (AWS SAM) to deploy and run a serverless application. AWS SAM is not a server or software.

It is just a specification that has to be followed for creating a serverless application. Once we create our serverless application, we can use CodePipeline to release and deploy it in AWS. CodePipeline is built on Continuous Integration Continuous Deployment (CI/CD) concept.

What are the main use cases for AWS Lambda?

Some of the main use cases in which AWS Lambda can be used are as follows: -

Web Application: We can integrate AWS Lambda with other AWS Services to create a web application that can scale up or down with zero administrative effort for server management, backup or scalability. Internet of Things (IoT): In the Internet of Things applications, we can use AWS Lambda to execute a piece of code on the basis of an event that is triggered by a device.

Mobile Backend: We can create Backend applications for Mobile apps by using AWS Lambda. Real-time Stream Processing: We can use AWS Lambda with Amazon Kinesis for processing real-time streaming data.

ETL: We can use AWS Lambda for Extract, Transform, and Load (ETL) operations in data warehousing applications. AWS Lambda can execute the code that can validate data, filter information, sort data or transform data from one form to another form.

Real-time File processing: AWS Lambda can also be used for handling any updates to a file in Amazon S3. When we upload a file to S3, AWS Lambda can create thumbnails, index files, new formats etc in real-time.

How does AWS Lambda handle failure during event processing?

In AWS Lambda we can run a function in synchronous or asynchronous mode. In synchronous mode, if AWS Lambda function fails, then it will just give an exception to the calling application. In asynchronous mode, if AWS Lambda function fails then it will retry the same function at least 3 times.

If AWS Lambda is running in response to an event in Amazon DynamoDB or Amazon Kinesis, then the event will be retried till the Lambda function succeeds or the data expires. In DynamoDB or Kinesis, AWS maintains data for at least 24 hours.

4.Storage

Amazon S3 Scalable Storage in the Cloud	**Amazon EBS** Block Storage for EC2	**AWS Elastic File System** Managed File Storage for EC2
Amazon Glacier Low-cost Achieve Storage in the cloud	**AWS Storage Gateway** Hybrid Storage Integration	**Amazon Snowball** Petabyte-Scale Data Transport
AWS Snowball Edge Petabyte-scale Data Transport with On-Demand Compute	**AWS Snowmobile** Exabyte-scale Data Transport	

S3

What is S3? What purpose S3 is designed for?

S3 stands for Simple Storage Service. You can use S3 interface to store and retrieve any amount of data, at any time and from anywhere on the web. Also, we can host a website in Amazon S3. most of the companies storing the documents, images and other files to S3. For S3, the payment model is "pay as you go". March 2006, Amazon launched Simple Storage Service (S3)

S3 is designed for:

- Remote data storage.
- Low cost, pay-as-you go.
- No up-front costs.
- High-availability.
- High bandwidth.

What is the significance in S3?

- S3 is object- based storage, it allows you to upload files
- Files can be from 0 bytes to 5 TB
- There is unlimited storage
- Files are stored in Buckets
- Write to S3 - HTTP 200 code for a successful write
- You can load files to S3 much faster by enabling multipart upload
- S3 is a universal namespace, that is name must be unique globally

  https://s3-eu-west-1.amazonaws.com/google

What are the S3 Storage Classes / Tiers?

- S3 (Durable, immediately available, frequently accessed)
- S3 - IA (Durable, immediately available, infrequently accessed)
- S3 - Reduced Redundancy Storage (data that is easily reproducible, such as thumb nails, etc)
- Glacier - archived data, where you can wait 3-5 hours before accessing

What are the core fundamentals of S3?

- Key (name)
- Value (data)
- Version ID
- Metadata

- Access Control Lists
- Object Based Storage File System
- Not Suitable to install an operating system on

What are the key Features of S3?

The key Features of S3 are: -

- 99.999999999% Durability.

- 99.99% Availability.

What are the key concepts of S3 storage?

The Key concepts are S3 storage are Buckets & Objects

Buckets

- A basic storage unit.

- Collection of Objects.

- It is a single level container, can contain multiple folders, or objects can be placed directly.

- Upload and download are easier.

- Name of the bucket should be globally unique.

- Allows maximum 100 buckets per user.

- No size restriction for Bucket.

- Data kept secured from unauthorized access through authentication mechanism.

Objects

- Equivalent to files.

- Allows max of 5TB for a single object.

What are the Object level properties?

The object level properties are:-

1. DETAILS:

STANDARD: 99.99999999%

STAND.INFREQUENT: LESSER THAN STANDARD.

Reduced Redundancy: 99.99% , LESSER THAN S.IA.

AES 256 - ADVANCED ENCRYPTION STANDARD.

2. PERMISSION:

- ME, ALL, RECOG.USER.

- OPEN, VIEW, EDIT.

3. META DATA: DATA'S DATA.
FORMAT OF THE FILE. EG: HTML / JPEG.

4. TAGS: INPROGRESS.

How to Access AWS S3 storage?

Accessible using simple HTTP URLs

- http://s3.amazonaws.com/bucket/key
- http://bucket.s3.amazonaws.com/key
- http://bucket/key

where bucket is a DNS CNAME record pointing to s3.amazonaws.com)

How Versioning is maintained in S3?

- Stores all versions of an object (including all writes and event if you delete an object)
- Great backup tool
- Once enabled, versioning cannot be disabled, only suspended.
- Integrates with Lifecycle rules
- Versioning's MFA Delete capability, which uses multi-factor authentication, can be used to provide an additional layer of security
- Cross Region Replication, requires versioning enabled on the source bucket

What is the Lifecycle Management in S3?

- Can be used in conjunction with versioning
- Can be applied current versions and previous versions
- Following actions can now be done: -
    - Transition to the standard - Infrequent Access Storage Class
      (128 Kb and 30 days after creation data)
    - Archive to the Glacier storage class (30 days after IA, if relevant)

How Security is done in S3?

- By default, all newly created buckets are PRIVATE
- You can setup access control to your buckets using: - Bucket Policies, ACL's
- S3 Buckets can be configured to create access logs which log all requests made to the s3 bucket. This can be done to another bucket

How Encryption is done in S3?

- In Transit: SSL/TLS
- At Rest
- Server-Side Encryption
    - S3 Managed Keys - SSE-S3
    - AWS Key Management Service, Managed Keys - SSE-KMS
    - Server-Side Encryption with Customer Provided Keys - SSE-C
- Client-Side Encryption

What are the different Storage Gateway on S3?

- File Gateway - For flat files, stored directly on S3
- Volume Gateway
    - Stored Volumes - Entire Dataset is stored on site and is asynchronously backed up to S3
    - Cached Volumes - Entire Dataset is stored on S3 and the most frequently accessed data is cached on site
- Gateway Virtual Tape Library (VTL)
    - Used for backup and uses popular backup applications like NetBackup, Backup Exec, Veeam etc.,

What are the important features of Amazon S3?

Some of the important features of Amazon S3 are as follows: Amazon S3 provides unlimited storage for files. File size in Amazon S3 can vary from 0 Bytes to 5 Terabytes. We have store files in Buckets in Amazon S3. In Amazon S3, names of buckets have to be unique globally. Amazon S3 is Object Based storage.

What is the maximum length of a file-name in S3?

Names are the object keys. The name for a key is a sequence of Unicode characters whose UTF-8 encoding is at most 1024 bytes long.

What is the scale of durability in Amazon S3?

Amazon S3 supports durability at the scale of 99.999999999% of time. This is 9 nines after decimal.

How can you check the disk space used by S3 bucket?

We can use s3cmd utility for this purpose. We can run s3cmd du command for this. We can also pass the bucket name as an argument to this command.

What are the Consistency levels supported by Amazon S3?

Amazon S3 supports Read after Write consistency when we create a new object by PUT. It means as soon as we Write a new object, we can access it. Amazon S3 supports Eventual Consistency when we overwrite an existing object by PUT. Eventual Consistency means that the effect of overwrite will not be immediate but will happen after some time. For deletion of an object, Amazon S3 supports Eventual Consistency after DELETE.

How can you send request to Amazon S3?

Amazon S3 is a REST service, you can send request by using the REST API or the AWS SDK wrapper libraries that wrap the underlying Amazon S3 REST API.

Mention what is the difference between Amazon S3 and EC2?

The difference between EC2 and Amazon S3 is that

EC2

It is a cloud web service used for hosting your application

It is like a huge computer machine which can run either Linux or Windows and can handle application like PHP, Python, Apache or any databases

S3

It is a data storage system where any amount of data can be stored

It has a REST interface and uses secure HMAC-SHA1 authentication keys

How many buckets can you create in AWS by default?

By default, you can create up to 100 buckets in each of your AWS accounts.

What is the command to copy all files from a S3 bucket to another bucket?

We can use s2cmd for this purpose. The command would be as follows: s3cmd sync s3://source/foo/bucket/ s3://destination/foo/bucket/

What are the different tiers in Amazon S3 storage?

Different Storage tiers in Amazon S3 are as follows:

S3 Standard: In this tier, S3 supports durable storage of files that become immediately available. This is used for frequently used files.

S3 Standard -Infrequent Access (IA): In this tier, S3 provides durable storage that is immediately available. But in this tier files are infrequently accessed.

S3 Reduced Redundancy Storage (RRS): In this tier, S3 provides the option to customers to store data at lower levels of redundancy. In this case data is copied to multiple locations but not on as many locations as standard S3.

How will you upload a file greater than 100 megabytes in Amazon S3?

Amazon S3 supports storing objects or files up to 5 terabytes. To upload a file greater than 100 megabytes, we have to use Multipart upload utility from AWS. By using Multipart upload we can upload a large file in multiple parts. Each part will be independently uploaded. It doesn't matter in what order each part is uploaded. It even supports uploading these parts in parallel to decrease overall time. Once all the parts are uploaded, this utility makes these as one single object or file from which the parts were created.

What happens to an Object when we delete it from Amazon S3?

Amazon S3 provides DELETE API to delete an object.

If the bucket in which the object exists is version controlled, then we can specify the version of the object that we want to delete. The other versions of the Object still exist within the bucket.

If we do not specify the version, and just pass the key name, Amazon S3 will delete the object and return the version id. And the object will not appear on the bucket. In case the bucket is Multi-factor authentication (MFA) enabled, then the DELETE request will fail if we do not specify a MFA token.

Mention what are the differences between Amazon S3 and EC2?

S3: Amazon S3 is just a storage service, typically used to store large binary files. Amazon also has other storage and database services, like RDS for relational databases and DynamoDB for NoSQL.

EC2: An EC2 instance is like a remote computer running Windows or Linux and on which you can install whatever software you want, including a Web server running PHP code and a database server.

How many buckets can you create in AWS by default?

By default, you can create up to 100 buckets in each of your AWS accounts.

How step you follow to make 10,000 files as public in S3?

I will generate a bucket policy which gives access to all the files in the bucket. The bucket policy can be added to a bucket through AWS console.

```
{
"Id": "…",
"Statement": [ {
```

```
"Sid": "...",
"Action": [
"s3:GetObject"
],
"Effect": "Allow",
"Resource": "arn:aws:s3:::bucket/*",
"Principal": {
"AWS": [ "*" ]
}
} ]
}
```

How do you see how much disk space is using by S3 bucket?

s3cmd can show you this by running s3cmd du, optionally passing the bucket name as an argument.

Write down the command you will use to copy all files from one S3 bucket to another with s3cmd?

s3cmd sync s3://from/this/bucket/ s3://to/this/bucket/

How many objects you can put in a S3 bucket? is there a limit to the number of objects I can put in an S3 bucket?

Write, read, and delete objects containing from 1 byte to 5 terabytes of data each. The number of objects you can store is unlimited.

How to delete files recursively from an S3 bucket?

aws s3 rm –recursive s3://your_bucket_name/foo/

Or delete everything under the bucket:

aws s3 rm –recursive s3://your_bucket_name

If what you want is to actually delete the bucket, there is one-step shortcut:

aws s3 rb –force s3://your_bucket_name

Can we disable versioning on a version-enabled bucket in Amazon S3?

No, we cannot disable versioning on a version-enabled bucket in Amazon S3. We can just suspend the versioning on a bucket in S3. Once we suspend versioning, Amazon S3 will stop creating new versions of the object. It just stores the object with null version ID. On overwriting an existing object, it just replaces the object with null version ID. So any existing versions of the object still remain in the bucket. But there will be no more new versions of the same object except for the null version ID object.

What are the use cases of Cross Region Replication Amazon S3?

We can use Cross Region Replication Amazon S3 to make copies of an object across buckets in different AWS Regions. This copying takes place automatically and in an asynchronous mode.

We have to add replication configuration on our source bucket in S3 to make use of Cross Region Replication. It will create exact replicas of the objects from source bucket to destination buckets in different regions.

Some of the main use cases of Cross Region Replication are as follows: Compliance: Sometimes there are laws/regulatory requirements that ask for storing data at farther geographic locations. This kind of compliance can be achieved by using AWS Regions that are spread across the world. Failover: At times, we want to minimize the probability of system failure due to complete blackout in a region.

We can use Cross-Region Replication in such a scenario. Latency: In case we are serving multiple geographies, it makes sense to replicate objects in the geographical Regions that are closer to end customer. This helps in reducing the latency.

Can we do Cross Region replication in Amazon S3 without enabling versioning on a bucket?

No, we have to enable versioning on a bucket to perform Cross Region Replication.

What are the different types of actions in Object Lifecycle Management in Amazon S3?

There are mainly two types of Object Lifecycle Management actions in Amazon S3. Transition Actions: These actions define the state when an Object transitions from one storage class to another storage class. E.g. a new object may transition to STANDARD_IA (infrequent access) class after 60 days of creation. And it can transition to GLACIER after 180 days of creation. Expiration Actions: These actions specify what happens when an Object expires. We can ask S3 to delete an object completely on expiration.

What are the security mechanisms available in Amazon S3?

Amazon S3 is a very secure storage service. Some of the main security mechanisms available in Amazon S3 are as follows: -

Access: When we create a bucket or an object, only the owner gets the access to the bucket and objects.

Authentication: Amazon S3 also support user authentication to control who has access to a specific object or bucket.

Access Control List: We can create Access Control Lists (ACL) to provide selective permissions to users and groups.

HTTPS: Amazon S3 also supports HTTPS protocol to securely upload and download data from cloud.

Encryption: We can also use Server-Side Encryption (SSE) in Amazon S3 to encrypt data.

You need to configure an Amazon S3 bucket to serve static assets for your public-facing web application. Which method will ensure that all objects uploaded to the bucket are set to public read?

A. Set permissions on the object to public read during upload.

B. Configure the bucket policy to set all objects to public read.

C. Use AWS Identity and Access Management roles to set the bucket to public read.

D. Amazon S3 objects default to public read, so no action is needed.

Answer B

Explanation: Rather than making changes to every object, its better to set the policy for the whole bucket. IAM is used to give more granular permissions, since this is a website, all objects would be public by default.

A customer wants to leverage Amazon Simple Storage Service (S3) and Amazon Glacier as part of their backup and archive infrastructure. The customer plans to use third-party software to support this integration. Which approach will limit the access of the third arty software to only the Amazon S3 bucket named "company-backup"?

A. A custom bucket policy limited to the Amazon S3 API in three Amazon Glacier archive "company-backup"

B. A custom bucket policy limited to the Amazon S3 API in "company-backup"

C. A custom IAM user policy limited to the Amazon S3 API for the Amazon Glacier archive "company-backup".

D. A custom IAM user policy limited to the Amazon S3 API in "company-backup".

Answer D

Explanation: Taking queue from the previous questions, this use case involves more granular permissions, hence IAM would be used here.

Can S3 be used with EC2 instances, if yes, how?

Yes, it can be used for instances with root devices backed by local instance storage. By using Amazon S3, developers have access to the same highly scalable, reliable, fast, inexpensive data storage infrastructure that Amazon uses to run its own global network of web sites. In order to execute systems in the Amazon EC2 environment, developers use the tools provided to load their Amazon Machine Images (AMIs) into Amazon S3 and to move them between Amazon S3 and Amazon EC2. Another use case could be for websites hosted on EC2 to load their static content from S3.

A customer implemented AWS Storage Gateway with a gateway-cached volume at their main office. An event takes the link between the main and branch office offline. Which methods will enable the branch office to access their data?

A. Restore by implementing a lifecycle policy on the Amazon S3 bucket.

B. Make an Amazon Glacier Restore API call to load the files into another Amazon S3 bucket within four to six hours.

C. Launch a new AWS Storage Gateway instance AMI in Amazon EC2 and restore from a gateway snapshot.

D. Create an Amazon EBS volume from a gateway snapshot and mount it to an Amazon EC2 instance.

Answer C

Explanation: The fastest way to do it would be launching a new storage gateway instance. Why? Since time is the key factor which drives every business, troubleshooting this problem will take more time. Rather than we can just restore the previous working state of the storage gateway on a new instance.

When you need to move data over long distances using the internet, for instance across countries or continents to your Amazon S3 bucket, which method or service will you use?

A. Amazon Glacier

B. Amazon CloudFront

C. Amazon Transfer Acceleration

D. Amazon Snowball

Answer C

Explanation: You would not use Snowball, because for now, the snowball service does not support cross region data transfer, and since, we are transferring across countries, Snowball cannot be used. Transfer Acceleration shall be the right choice here as it throttles your data transfer with the use of optimized network paths and Amazon's content delivery network upto 300% compared to normal data transfer speed.

How can you speed up data transfer in Snowball?

The data transfer can be increased in the following way:

By performing multiple copy operations at one time i.e. if the workstation is powerful enough, you can initiate multiple cp commands each from different terminals, on the same Snowball device.

Copying from multiple workstations to the same snowball.

Transferring large files or by creating a batch of small file, this will reduce the encryption overhead.

Eliminating unnecessary hops i.e. make a setup where the source machine(s) and the snowball are the only machines active on the switch being used, this can hugely improve performance.

Glacier

What is the use of Amazon Glacier?

Amazon Glacier is an extremely low-cost cloud-based storage service provided by Amazon. We mainly use Amazon Glacier for long-term backup purpose. Amazon Glacier can be used for storing data archives for months, years or even decades.

It can also be used for long term immutable storage based on regulatory and archiving requirements. It provides Vault Lock support for this purpose. In this option, we write once but can read many times same data. One use case is for storing certificates that can be issued only once and only the original person keeps the main copy.

Suppose that you are working with a customer who has 10 TB of archival data that they want to migrate to glacier. The customer has a 1-Mbps connection to the internet. Which service or feature provides the fastest method of getting data in to Amazon Glacier?

AWS Import | Export

I created a key in Oregon region to encrypt my data in North Virginia region for security purposes. I added two users to the key and an external AWS account. I wanted to encrypt an object in S3, so when I tried, the key that I just created was not listed. What could be the reason?

External aws accounts are not supported.

AWS S3 cannot be integrated KMS.

The Key should be in the same region.

New keys take some time to reflect in the list.

Answer C.

Explanation: The key created and the data to be encrypted should be in the same region. Hence the approach taken here to secure the data is incorrect.

Storage Gateway

What are the benefits of AWS Storage Gateway?

We can use AWS Storage Gateway (ASG) service to connect our local infrastructure for files etc with Amazon cloud services for storage. Some of the main benefits of AWS Storage Gateway are as follows:

Local Use: We can use ASG to integrate our data in multiple Amazon Storage Services like- S3, Glacier etc with our local systems. We can continue to use our local systems seamlessly.

Performance: ASG provides better performance by caching data in local disks. Though data stays in cloud, but the performance we get is similar to that of local storage.

Easy to use: ASG provides a virtual machine to use it by an easy to use interface. There is no need to install any client or provision rack space for using ASG. These virtual machines can work in local system as well as in AWS.

Scale: We get the storage at a very high scale with ASG. Since backend in ASG is Amazon cloud, it can handle large amounts of workloads and storage needs.

Optimized Transfer: ASG performs many optimizations, due to which only the changes to data are transferred. This helps in minimizing the use of bandwidth.

What are the main use cases for AWS Storage Gateway?

AWS Storage Gateway (ASG) is very versatile in its usage. It solves a variety of problems at an enterprise.

Some of the main use cases of ASG are as follows:

Backup systems: We can use ASG to create backup systems. From local storage data can be backed up into cloud services of AWS. On demand, we can also restore the data from this backup solution. It is a replacement for Tape based backup systems.

Variable Storage: With ASG, we can grow or shrink our Storage as per our needs. There is no need to add racks, disks etc to expand our storage systems. We can manage the fluctuations in our storage needs gracefully by using ASG.

Disaster Recovery: We can also use ASG for disaster recovery mechanism. We can create snapshots of our local volumes in Amazon EBS. In case of a local disaster we can use our applications in cloud and

recover from the snapshots created in EBS. Hybrid Cloud: At times we want to use our local applications with cloud services. ASG helps in implementing Hybrid cloud solutions in which we can utilize cloud storage services with our on premises local applications.

Snowball

What is AWS Snowball?

AWS provides a very useful service called Snowball for transporting very large amounts of data at the scale of petabytes. With Snowball, we can securely transfer data without any network cost. It is a physical data transfer solution to store data in AWS cloud.

Once we create a Snowball job in AWS console, Amazon ships a physical storage device to our location. We can copy our data to this storage device and ship it back. Amazon services will take the Snowball device and transfer the data to Amazon S3.

- Snowball can
    - Import to S3, Export from S3
- Snowball Edge
- Snowmobile

What is Transfer Acceleration in S3?

You can speed up transfers to S3 using S3 transfer acceleration. This costs extra and has the greatest impact on people who are in faraway location.

What is the purpose of Static Websites in S3?

You can use S3 to host static websites

- Serverless
- Very cheap, scales automatically
- STATIC only, cannot host dynamic sites

5.Database

Amazon Aurora High Performance Managed Relational Database	**Amazon RDS** Managed Relational Database Service for MySQL, PostgreSQL, Oracle, SQL Server and MariaDB	**Amazon DynamoDB** Managed NoSQL Database
Amazon Elastic Cache In-memory Caching System	**AWS Redshift** Fast, Simple, Cost-effective Data Warehousing	**Amazon Neptune** Fully Managed Graph Database Service
AWS Database Migration Service Migrate Database with Minimal Downtime		

RDS

What is Amazon RDS?

RDS stand for Relational Database Service is a web service that makes it easier to setup, operate, and scale a relational database in the cloud. It provides cost-efficient, resizable capacity for an industry-standard relational database and manages common database administration tasks.

In RDS, what is the maximum value you can set for my backup retention period?

35 days

In RDS, Automated backups are enabled by default for new DB instance, true or false?

True.

Summarize the Database Key Concepts

What are the AWS Database Types?

- RDS - OLTP
    - SQL, MySQL, PostgreSQL, Oracle, Aurora, MariaDB
- DynamoDB - NoSQL
- RedShift – OLAP
- Elastic Cache – In Memory Caching
    - Memcached, Redis

Aurora Scaling

Two copies of your data is contained in each availability zone, with minimum of 3 availability zones. Six copies of your data.

Aurora is designed to transparently handle the loss of up to two copies of data without affecting database write availability and up to three copies without affecting read availability.

Aurora storage is also self-healing. Data blocks and disks are continuously scanned for errors and repaired automatically

Aurora Replica

- 2 Types of Replicas are available
- Aurora Replicas (Currently 15)
- MySQL Read Replicas (Currently 15)

DynamoDB Vs RDS

- DynamoDB offers "Push Button" scaling, meaning that you can scale your database on the fly, without any down time.
- RDS is not so easy and you usually have to use a bigger instance size or to add a read replica

DynamoDB

- Stored on SSD storage
- Spread Across 3 geographically distinct data Centre's
- Eventual Consistent Reads (Default)
- Strongly Consistent Reads

Redshift Configuration

- Single Node (160 GB)
- Multi-Node
    - Leader Node (Manages Client Connections and receives queries)
    - Compute Node (Store Data and Perform queries and computations) up to 128 Compute Nodes

What is Elastic Cache?

Elastic Cache is a web service that makes it easy to deploy, operate, and scale an in-memory cache in the cloud. The service improves the performance of web applications by allowing you to retrieve information from fast, managed, in-memory caches, instead of relying entirely on slower disk-based databases. Elastic Cache supports two open source in-memory engines namely: - Memcached & Redis

If you want to run a database on an EC2 instance, which is the most recommended Amazon storage option, S3, RDS or EBS?

EBS

In S3, what does RRS stand for?

Reduced Redundancy Storage

If I launch a standby RDS instance, will it be in the same Availability Zone as my primary?

A. Only for Oracle RDS types

B. Yes

C. Only if it is configured at launch

D. No

Answer D

Explanation: No, since the purpose of having a standby instance is to avoid an infrastructure failure (if it happens), therefore the standby instance is stored in a different availability zone, which is a physically different independent infrastructure.

When would I prefer Provisioned IOPS over Standard RDS storage?

A. If you have batch-oriented workloads

B. If you use production online transaction processing (OLTP) workloads.

C. If you have workloads that are not sensitive to consistent performance

D. All of the above

Answer A

Explanation: Provisioned IOPS deliver high IO rates but on the other hand it is expensive as well. Batch processing workloads do not require manual intervention they enable full utilization of systems, therefore a provisioned IOPS will be preferred for batch-oriented workload.

How is Amazon RDS, DynamoDB and Redshift different?

Amazon RDS is a database management service for relational databases, it manages patching, upgrading, backing up of data etc. of databases for you without your intervention. RDS is a Db management service for structured data only.

DynamoDB, on the other hand, is a NoSQL database service, NoSQL deals with unstructured data. Redshift, is an entirely different service, it is a data warehouse product and is used in data analysis.

If I am running my DB Instance as a Multi-AZ deployment, can I use the standby DB Instance for read or write operations along with primary DB instance?

A. Yes

B. Only with MySQL based RDS

C. Only for Oracle RDS instances

D. No

Answer D

Explanation: No, Standby DB instance cannot be used with primary DB instance in parallel, as the former is solely used for standby purposes, it cannot be used unless the primary instance goes down.

Your company's branch offices are all over the world, they use a software with a multi-regional deployment on AWS, they use MySQL 5.6 for data persistence.

The task is to run an hourly batch process and read data from every region to compute cross-regional reports which will be distributed to all the branches. This should be done in the shortest time possible. How will you build the DB architecture in order to meet the requirements?

A. For each regional deployment, use RDS MySQL with a master in the region and a read replica in the HQ region

B. For each regional deployment, use MySQL on EC2 with a master in the region and send hourly EBS snapshots to the HQ region

C. For each regional deployment, use RDS MySQL with a master in the region and send hourly RDS snapshots to the HQ region

D. For each regional deployment, use MySQL on EC2 with a master in the region and use S3 to copy data files hourly to the HQ region

Answer A

Explanation: For this we will take an RDS instance as a master, because it will manage our database for us and since we have to read from every region, we'll put a read replica of this instance in every region where the data has to be read from. Option C is not correct since putting a read replica would be more efficient than putting a snapshot, a read replica can be promoted if needed to an independent DB instance, but with a Db snapshot it becomes mandatory to launch a separate DB Instance.

Can I run more than one DB instance for Amazon RDS for free?

Yes. You can run more than one Single-AZ Micro database instance, that too for free! However, any use exceeding 750 instance hours, across all Amazon RDS Single-AZ Micro DB instances, across all eligible database engines and regions, will be billed at standard Amazon RDS prices.

For example: if you run two Single-AZ Micro DB instances for 400 hours each in a single month, you will accumulate 800 instance hours of usage, of which 750 hours will be free. You will be billed for the remaining 50 hours at the standard Amazon RDS price.

Which AWS services will you use to collect and process e-commerce data for near real-time analysis?

A. Amazon ElastiCache

B. Amazon DynamoDB

C. Amazon Redshift

D. Amazon Elastic MapReduce

Answer B, C

Explanation: DynamoDB is a fully managed NoSQL database service. DynamoDB, therefore can be fed any type of unstructured data, which can be data from e-commerce websites as well, and later, an

analysis can be done on them using Amazon Redshift. We are not using Elastic MapReduce, since a near real time analyses is needed.

Can I retrieve only a specific element of the data, if I have a nested JSON data in DynamoDB?

Yes. When using the GetItem, BatchGetItem, Query or Scan APIs, you can define a Projection Expression to determine which attributes should be retrieved from the table. Those attributes can include scalars, sets, or elements of a JSON document.

A company is deploying a new two-tier web application in AWS. The company has limited staff and requires high availability, and the application requires complex queries and table joins. Which configuration provides the solution for the company's requirements?

A. MySQL Installed on two Amazon EC2 Instances in a single Availability Zone

B. Amazon RDS for MySQL with Multi-AZ

C. Amazon ElastiCache

D. Amazon DynamoDB

Answer D

Explanation: DynamoDB has the ability to scale more than RDS or any other relational database service, therefore DynamoDB would be the apt choice.

What happens to my backups and DB Snapshots if I delete my DB Instance?

When you delete a DB instance, you have an option of creating a final DB snapshot, if you do that you can restore your database from that snapshot. RDS retains this user-created DB snapshot along with all other manually created DB snapshots after the instance is deleted, also automated backups are deleted and only manually created DB Snapshots are retained.

Which of the following use cases are suitable for Amazon DynamoDB? (Choose 2 answers)

A. Managing web sessions.

B. Storing JSON documents.

C. Storing metadata for Amazon S3 objects.

D. Running relational joins and complex updates.

Answer C, D

Explanation: If all your JSON data have the same fields eg [id,name,age] then it would be better to store it in a relational database, the metadata on the other hand is unstructured, also running relational joins or complex updates would work on DynamoDB as well.

How can I load my data to Amazon Redshift from different data sources like Amazon RDS, Amazon DynamoDB and Amazon EC2?

You can load the data in the following two ways: -

You can use the COPY command to load data in parallel directly to Amazon Redshift from Amazon EMR, Amazon DynamoDB, or any SSH-enabled host.

AWS Data Pipeline provides a high performance, reliable, fault tolerant solution to load data from a variety of AWS data sources. You can use AWS Data Pipeline to specify the data source, desired data transformations, and then execute a pre-written import script to load your data into Amazon Redshift.

Your application has to retrieve data from your user's mobile every 5 minutes and the data is stored in DynamoDB, later every day at a particular time the data is extracted into S3 on a per user basis and then your application is later used to visualize the data to the user. You are asked to optimize the architecture of the backend system to lower cost, what would you recommend?

A. Create a new Amazon DynamoDB (able each day and drop the one for the previous day after its data is on Amazon S3.

B. Introduce an Amazon SQS queue to buffer writes to the Amazon DynamoDB table and reduce provisioned write throughput.

C. Introduce Amazon Elasticache to cache reads from the Amazon DynamoDB table and reduce provisioned read throughput.

D. Write data directly into an Amazon Redshift cluster replacing both Amazon DynamoDB and Amazon S3.

Answer C

Explanation: Since our work requires the data to be extracted and analyzed, to optimize this process a person would use provisioned IO, but since it is expensive, using a ElastiCache memoryinsread to cache the results in the memory can reduce the provisioned read throughput and hence reduce cost without affecting the performance.

You are running a website on EC2 instances deployed across multiple Availability Zones with a Multi-AZ RDS MySQL Extra Large DB Instance. The site performs a high number of small reads and writes per second and relies on an eventual consistency model. After comprehensive tests you discover that there is read contention on RDS MySQL. Which are the best approaches to meet these requirements? (Choose 2 answers)

A. Deploy ElastiCache in-memory cache running in each availability zone

B. Implement Sharding to distribute load to multiple RDS MySQL instances

C. Increase the RDS MySQL Instance size and Implement provisioned IOPS

D. Add an RDS MySQL read replica in each availability zone

Answer A, C

Explanation: Since it does a lot of read writes, provisioned IO may become expensive. But we need high performance as well, therefore the data can be cached using ElastiCache which can be used for frequently reading the data. As for RDS since read contention is happening, the instance size should be increased and provisioned IO should be introduced to increase the performance.

A startup is running a pilot deployment of around 100 sensors to measure street noise and air quality in urban areas for 3 months. It was noted that every month around 4GB of sensor data is generated. The company uses a load balanced auto scaled layer of EC2 instances and a RDS database with 500 GB standard storage. The pilot was a success and now they want to deploy at least 100K sensors which need to be supported by the backend. You need to store the data for at least 2 years to analyze it. Which setup of the following would you prefer?

A. Add an SQS queue to the ingestion layer to buffer writes to the RDS instance

B. Ingest data into a DynamoDB table and move old data to a Redshift cluster

C. Replace the RDS instance with a 6 node Redshift cluster with 96TB of storage

D. Keep the current architecture but upgrade RDS storage to 3TB and 10K provisioned IOPS

Answer C

Explanation: A Redshift cluster would be preferred because it easy to scale, also the work would be done in parallel through the nodes, therefore is perfect for a bigger workload like our use case. Since each month 4 GB of data is generated, therefore in 2 year, it should be around 96 GB. And since the servers will be increased to 100K in number, 96 GB will approximately become 96TB. Hence option C is the right answer.

DynamoDB

What is DynamoDB in AWS?

AWS provides a NoSQL database called Amazon DynamoDB. It can be used to store data in a NoSQL environment. DynamoDB gives very fast and predictable performance. It is highly scalable. We can use Amazon DynamoDB to create a database table to store and retrieve any amount of data. It is capable of serving very high volume of request traffic. Any level of request traffic. Amazon DynamoDB also provides support for automatically distributing the data and traffic of a table on multiple servers to handle the spikes in request traffic. Even after distributing the load it provides consistent performance.

What is DynamoDB?

Amazon DynamoDB is a fully managed NoSQL database service that provides fast and predictable performance with seamless scalability. You can use Amazon DynamoDB to create a database table that can store and retrieve any amount of data and serve any level of request traffic. Amazon DynamoDB automatically spreads the data and traffic for the table over a sufficient number of servers to handle the request capacity specified by the customer and the amount of data stored, while maintaining consistent and fast performance.

What are the main benefits of using Amazon DynamoDB?

Amazon DynamoDB is a highly scalable NoSQL database that has very fast performance. Some of the main benefits of using Amazon DynamoDB are as follows:-

Administration: In Amazon DynamoDB, we do not have to spend effort on administration of database. There are no servers to provision or manage. We just create our tables and start using them.

Scalability: DynamoDB provides the option to specify the capacity that we need for a table. Rest of the scalability is done under the hood by DynamoDB.

Fast Performance: Even at a very high scale, DynamoDB delivers very fast performance with low latency. It will use SSD and partitioning behind the scenes to achieve the throughput that a user specifies.

Access Control: We can integrate DynamoDB with IAM to create fine-grained access control. This can keep our data secure in DynamoDB.

Flexible: DynamoDB supports both document and key-value data structures. So it helps in providing flexibility of selecting the right architecture for our application.

Event Driven: We can also make use of AWS Lambda with DynamoDB to perform any event driven programming. This option is very useful for ETL tasks.

What is the basic Data Model in Amazon DynamoDB?

The basic Data Model in Amazon DynamoDB consists of following components: Table: In DynamoDB, a Table is collection of data items. It is similar to a table in a Relational Database. There can be infinite number of items in a Table. There has to be one Primary key in a Table. Item: An Item in DynamoDB is made up of a primary key or composite key and a variable number of attributes. The number of attributes in an Item is not bounded by a limit. But total size of an Item can be maximum 400 kilobytes. Attribute: In DynamoDB, we can associate an Attribute with an Item. We can set a name as well as one or more values in an Attribute. Total size of data in an Attribute is maximum 400 kilobytes.

What are the different APIs available in Amazon DynamoDB?

Amazon DynamoDB supports both document as well as key based NoSQL databases. Due to this APIs in DynamoDB are generic enough to serve both the types. Some of the main APIs available in DynamoDB are as follows: -

CreateTable, UpdateTable, DeleteTable, DescribeTable, ListTables,

PutItem, GetItem, BatchWriteItem, BatchGetItem, UpdateItem, DeleteItem, Query & Scan

When should be use Amazon DynamoDB vs. Amazon S3?

Amazon DynamoDB is used for storing structured data. The data in DynamoDB is also indexed by a primary key for fast access. Reads and writes in DynamoDB have very low latency due to the use SSD. Amazon S3 is mainly used for storing unstructured binary large objects-based data. It does not have a fast index like DynamoDB.

So, we should use Amazon S3 for storing objects with infrequent access requirements. Another consideration is size of the data. In DynamoDB the size of an item can be maximum 400 kilobytes. Whereas Amazon S3 supports size as large as 5 terabytes for an object. Therefore, DynamoDB is more suitable for storing small objects with frequent access and S3 is ideal for storing very large objects with infrequent access.

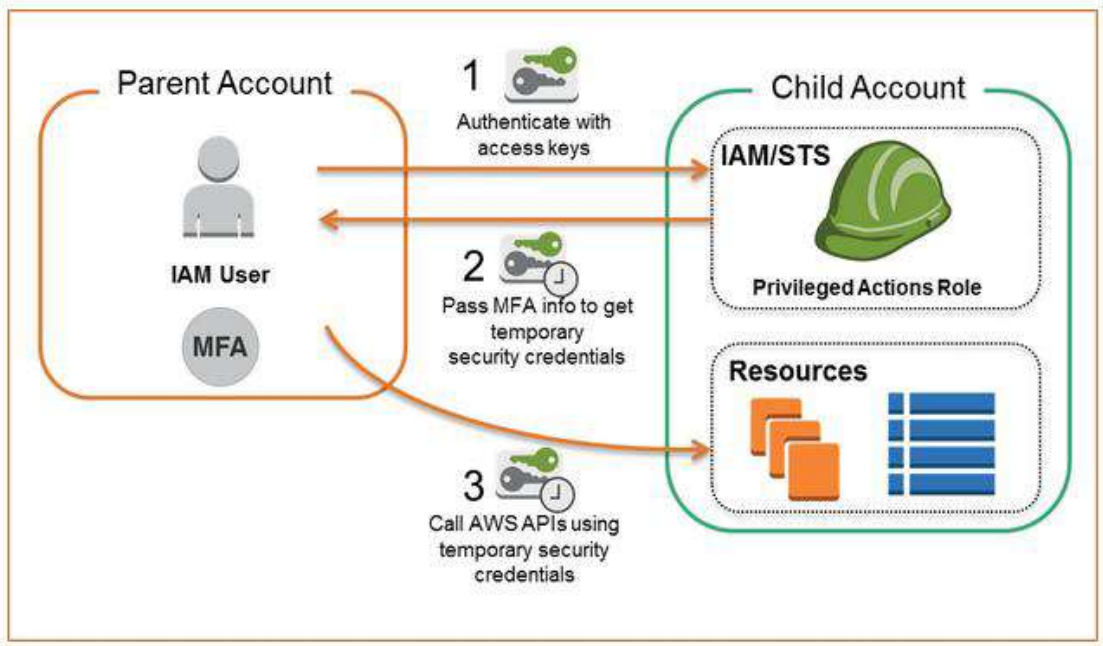
Redshift

What is Redshift?

Redshift is a fast, fully managed, petabyte-scale data warehouse service that makes it simple and cost-effective to efficiently analyze all your data using your existing business intelligence tools.

6.Security, Identity & Compliance

AWS Identity & Access Management Manage User Access and Encryption Keys	**Amazon Cloud Directory** Create Flexible Cloud Native Directories	**Amazon Cognito** Identity Management for your Apps
Amazon GaurdDuty Managed Threat Detection Service	**Amazon Inspector** Analyze Application Security	**Amazon Macie** Discover, Classify, and Protect your Data
AWS Certificate Manager Provision, Manage and Deploy SSL/TLS Certificates	**AWS CloudHSM** Hardware-based Key Storage for Regulatory Compliance	**Amazon Directory Service** Host and manage Active Directory
AWS Firewall Manager Central Management of Firewall Rules	**AWS Key Management Service** Managed Creation and Control of Encryption Keys	**AWS Organizations** Policy-based Management for Multiple AWS Accounts
AWS Secrets Manager Rotate, Manage and Retrieve Secrets	**AWS Single Sign-on** Cloud Single Sign-on (SSO) Service	**AWS Shield** DDoS Protection
AWS WAF Filter Malicious Web Traffic		



IAM

What is IAM? What is IAM service?

- IAM stands for Identity and Access Management
- IAM is a web services that enable you to manage users and group permissions in AWS
- It is targeted at organizations with multiple users or systems that use AWS products such as Amazon Elastic Compute Cloud, Amazon Relational Database Service, and the AWS Management Console
- AWS Identity and Access Management (IAM) is a web service that helps you securely control access to AWS resources for your users. You use IAM to control who can use your AWS resources (authentication) and what resources they can use and in what ways (authorization).

What does IAM gives you?

- Centralized control of your AWS account, Granular Permissions & Multifactor Authentication
- Identity Federation (Including Active Directory, Facebook, LinkedIn etc.,)
- Provide temporary access for users | devices and services where necessary
- Allow you to set up your own password rotation policy
- Integrates with many different AWS services
- Supports PCI DSS Compliance

What are the important components of IAM?

The important components of IAM are as follows:

- IAM User: An IAM User is a person or service that will interact with AWS. User can sign into AWS Management Console for performing tasks in AWS. (End Users)

- IAM Group: An IAM Group is a collection of IAM users under one set of permissions. We can specify permission to an IAM Group. This helps in managing large number of IAM users. We can simply add or remove an IAM User to an IAM Group to manage the permissions.

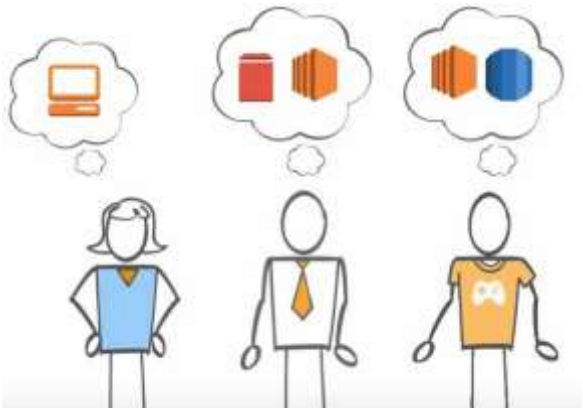
- IAM Role: An IAM Role is an identity to which we give permissions. A Role does not have any credentials (password or access keys). We can temporarily give an IAM Role to an IAM User to perform certain tasks in AWS.

- IAM Permission: In IAM we can create two types of Permissions. Identity based and Resource based. We can create a Permission to access or perform an action on an AWS Resource and assign it to a User, Role or Group. We can also create Permissions on resources like S3 bucket, Glacier vault etc and specify who has access to the resource.

- **IAM Policy:** An IAM Policy is a document in which we list permissions to specify Actions, Resources and Effects. This document is in JSON format. We can attach a Policy to an IAM User or Group.

Why we go for IAM?

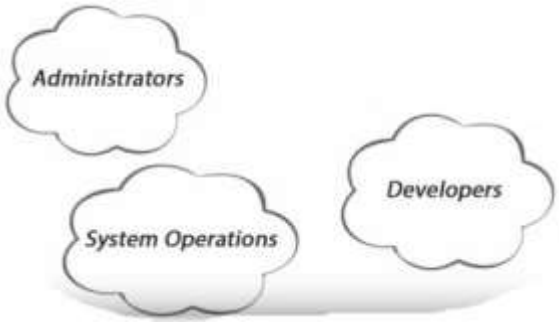
- To avoid a security and logistical headache
- When you create an AWS account, it has permissions to do anything and everything with all the resources
- IAM Allows you to limit access as needed and gives you the peace of mind that approved people are accessing the right resources in the desired manner
- IAM will allow us to create multiple users with individual security credentials and permissions, with this IAM, each user is allowed to do only what they need to do



- Each user in the AWS account must have a unique set of credentials to access the console

How IAM works?

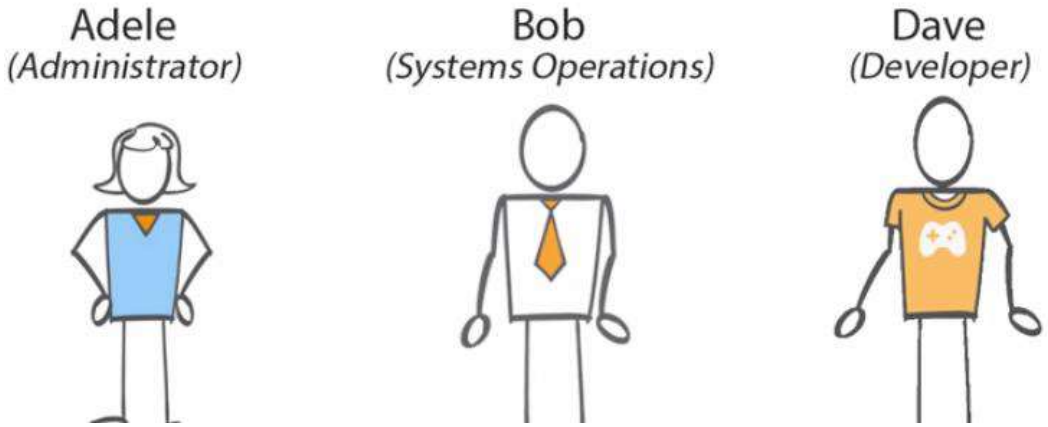
- **Different types of users have different set of permissions**



- **Administrators need to access all AWS resource**

- **Developers need only access on Amazon Elastic Compute Cloud (EC2)**

- **We can use IAM to create a unique user for each employee and define their permissions**



What is a Group?

- A group is a collection of IAM users
- After you set permissions on a group, those permissions are set to all users in the group
    - Even if we create user, we need to use groups to set permissions.
    - We need to manage access for number if groups instead of managing access for every individual user.

› Administrator Access
Provides full access to AWS services and resources.

- We can able to,
    - Create a Group
    - Review the Group
    - Attach policy
    - Change the Group name
    - Delete a Group
    - Adding User to the Group

What is Multi-Factor Authentication?

AWS Multi-Factor Authentication (MFA) is a simple best practice that adds an extra layer of protection on top of user name and password. With MFA enabled, when a user signs into an AWS website, the will be prompted for their user name and password, as well as for an authentication code from their AWS MFA device. Taken together, these multiple factors provide increased security for your AWS.

- MFA provides additional security by requiring users to use a password and an authentication code from an external device.

- MFA is especially recommended for the AWS root accounts and account with administrator permissions since they have access to all your AWS resources.

Two types:

1. Hardware MFA device [Like your RSA token]

2. Virtual MFA device

Notes:

- It's not region specific.

- The Created Roles, Users, policies, groups etc are Universal, thus can be used across the regions.

Summarize the IAM Key Points

- IAM consists of the following: -
    - Users
    - Groups (A way to group our users and apply policies to them collectively)
    - Roles
    - Policy Documents (using JSON)
- IAM is Universal. It does not apply to regions at this time.
- The "root account" is simply the account created when first setup your AWS account. It has complete Admin access.
- New Users have No Permissions when first created
- New Users are assigned "Access Key ID & Secret Access Keys" when first create
- These are not the same as a password, and you cannot use the Access key ID & Secret Access Key to Login in to the console. You can use this to access AWS via the APIs and Command Line however
- Always setup Multifactor Authentication on your root account
- You can create and customize your own password rotation policies
- Roles are more secure than storing your access key and secret access key on individual EC2 instances
- Roles are easier to manage
- Roles can be assigned to an EC2 instance AFTER it has been provisioned using both the command line and AWS Console
- Roles are universal, you can use them in any region

What is AWS Certificate Manager?

AWS Certificate Manager (ACM) handles the complexity of provisioning, deploying, and managing certificates provided by ACM (ACM Certificates) for your AWS-based websites and applications. You use ACM to request and manage the certificate and then use other AWS services to provision the ACM Certificate for your website or application. As shown by the following illustration, ACM Certificates are currently available for use with only Elastic Load Balancing and Amazon CloudFront. You cannot use ACM Certificates outside of AWS.

Like any other cloud computing environment, in AWS security is very important. We use AWS Certificate Manager (ACM) to handle the administration of security certificates (ACM Certificates) provided by AWS. ACM can be used to provision, deploy and manage the certificates in cloud environment. It can be used to provision as certificate on AWS based website. AWS certificates have a limitation that they can not be used outside AWS

How will you manage multiple users and their access rights with Amazon IAM?

AWS Identity and Access Management (IAM) is a web service in AWS cloud. It provides us APIs to create multiple Users and manage their permissions on AWS resources. A user in AWS is an identity with unique security credentials that can be used to access AWS Services and Resources. With IAM we do not need to share passwords or access keys. IAM makes it easy to enable or disable a User's access as per the configuration.

We can implement best practices of security like least privilege, granting unique credentials to every User within AWS account etc. by using IAM.

What is the AWS Key Management Service?

The AWS Key Management Service (AWS KMS) is a managed service that makes it easy for you to create and control the encryption keys used to encrypt your data.

What is AWS WAF? What are the potential benefits of using WAF?

AWS WAF is a web application firewall that lets you monitor the HTTP and HTTPS requests that are forwarded to Amazon CloudFront and lets you control access to your content. Based on conditions that you specify, such as the IP addresses that requests originate from or the values of query strings, CloudFront responds to requests either with the requested content or with an HTTP 403 status code (Forbidden. You can also configure CloudFront to return a custom error page when a request is blocked.

Benefits of using WAF:

- **Additional protection against web attacks using conditions that you specify. You can define conditions by using characteristics of web requests such as the IP address that the requests originate from, the values in headers, strings that appear in the requests, and the presence of malicious SQL code in the request, which is known as SQL injection.**
- **Rules that you can reuse for multiple web applications**
- **Real-time metrics and sampled web requests**
- **Automated administration using the AWS WAF API**

7.Networking & Content Delivery

Amazon VPC Isolated Cloud Resources	**Amazon CloudFront** Global Content Delivery Network	**Amazon Route 53** Scalable Domain Name System
Amazon API Gateway Build, Deploy, and Manage APIs	**AWS Direct Connect** Dedicated Network Connection to AWS	**AWS Load Balancing** High Scale Load Balancing

VPC

What is Amazon VPC?

It enables you to launch Amazon Web Services (AWS) resources into a virtual network that you have defined. This virtual network closely resembles a traditional network that you would operate in your own data center, with the benefits of using the scalable infrastructure of AWS.

What is DNS?

If you have used the internet, you have used DNS. DNS is used to convert human friendly domain names (e.g. http://amazon.com) into an Internet Protocol (IP) address (e.g. http://192.68.56.1)

IP addresses are used by computers to identify each other on the network. IP addresses commonly come in two different forms such as IPV4 and IPV6.

What is CIDR?

Classless inter-domain routing (CIDR) is a set of Internet protocol (IP) standards that is used to create unique identifiers for networks and individual devices. The IP addresses allow particular information packets to be sent to specific computers. ... That system is known as CIDR notation.

What is Subnet in AWS?

VPC and Subnet Basics. A virtual private cloud (VPC) is a virtual network dedicated to your AWS account. It is logically isolated from other virtual networks in the AWS Cloud. You can launch your AWS resources, such as Amazon EC2 instances, into your VPC.

A subnetwork or subnet is a logical subdivision of an IP network. The practice of dividing a network into two or more networks is called subnetting. Computers that belong to a subnet are addressed with a common, identical, most-significant bit-group in their IP address.

What is Rout Tables?

A route table contains a set of rules, called routes, that are used to determine where network traffic is directed.

Each subnet in your VPC must be associated with a route table; the table controls the routing for the subnet. A subnet can only be associated with one route table at a time, but you can associate multiple subnets with the same route table.

The following are the basic things that you need to know about route tables:

- Your VPC has an implicit router.
- Your VPC automatically comes with a main route table that you can modify.
- You can create additional custom route tables for your VPC.
- Each subnet must be associated with a route table, which controls the routing for the subnet. If you don't explicitly associate a subnet with a particular route table, the subnet is implicitly associated with the main route table.
- You cannot delete the main route table, but you can replace the main route table with a custom table that you've created (so that this table is the default table each new subnet is associated with).

What is Internet Gateways?

An Internet gateway is a horizontally scaled, redundant, and highly available VPC component that allows communication between instances in your VPC and the Internet. It therefore imposes no availability risks or bandwidth constraints on your network traffic.

An Internet gateway serves two purposes: to provide a target in your VPC route tables for Internet-routable traffic, and to perform network address translation (NAT) for instances that have been assigned public IPv4 addresses.

An Internet gateway supports IPv4 and IPv6 traffic.

To enable access to or from the Internet for instances in a VPC subnet, you must do the following:

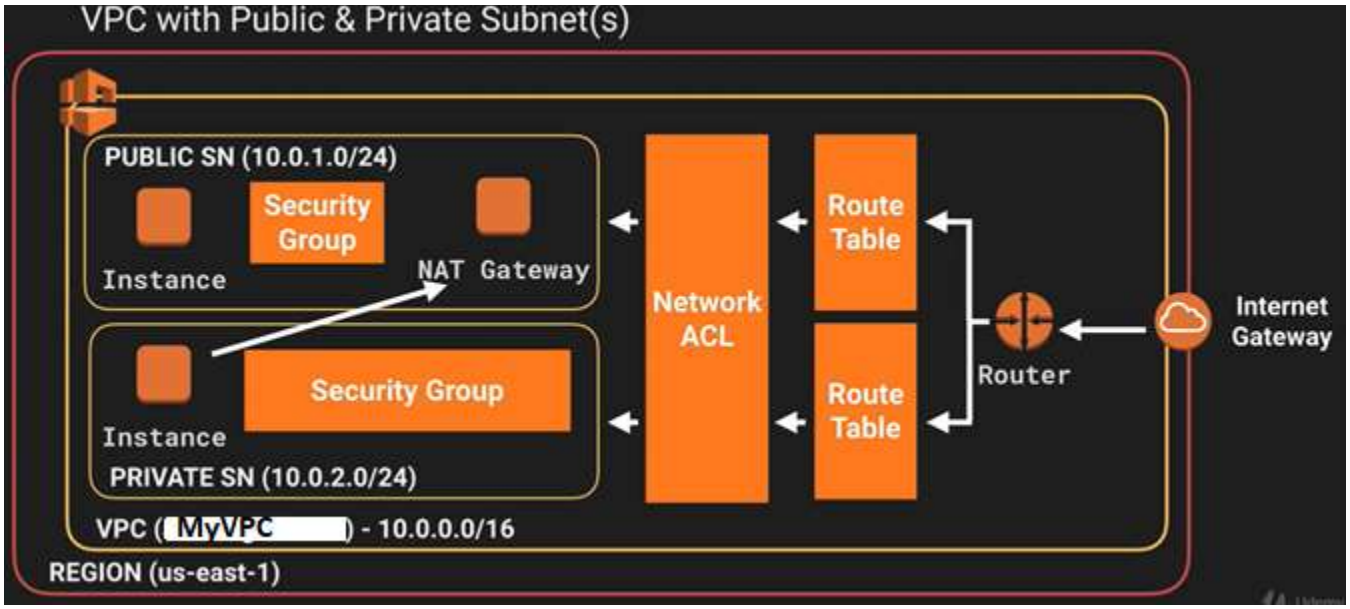
- Attach an Internet gateway to your VPC.
- Ensure that your subnet's route table points to the Internet gateway.
- Ensure that instances in your subnet have a globally unique IP address (public IPv4 address, Elastic IP address, or IPv6 address).
- Ensure that your network access control and security group rules allow the relevant traffic to flow to and from your instance.

What is IPV4 and IPV6?

IPV4 space is a 32-bit field and has over 4 billion different address (4,294,967296 to be precise)

IPV6 was created to solve the depletion issue and has an address space of 128 bits which is 340 undecillion addresses (340,282,366,920,938,463,374,607,431,768,211,456)

Sketch the VPC Flow?



What is NAT Instances?

- When creating a NAT instance, disable source | destination check on the Instance
- NAT instances must be in a public subnet
- There must be a route out of the private subnet to the NAT instance, in order for this to work
- The amount of traffic that NAT instances can support depends on the instance size. If you are bottlenecking, increase the instance size.
- You can create high availability using autoscaling groups, multiple subnets in different AZs, and a script to automate failover
- Behind a Security Group

What is NAT Gateway?

- Preferred by the enterprise
- Scale automatically up to 10Gbps
- No need to patch
- Not associated with security groups
- Automatically assigned a public IP address
- Remember to update your route tables
- No need to disable Source | Destination checks
- More secure than a NAT instance

What is Network ACLS?

- Your VPC automatically comes a default network ACL, and by default it allows all outbound and inbound traffic
- You can create custom network ACLS. By default, each custom network ACL denies all in-bound and outbound traffic until you add rules
- Each subnet in your VPC must be associated with a network ACL. If you don't explicitly associate a subnet with a network ACL, the subnet is automatically associated with the default ACL.
- You can associate a network ACL with multiple subnets; however, a subnet can be associated with only one network ACL at a time. When you associate a network ACL with a subnet, the previous association is removed
- Network ACLs contain a numbered list of rules that is evaluated in order, starting with the lowest numbered rule.
- Network ACLs have separate inbound and outbound rules, and each rule can either allow or deny traffic
- Network ACLs are stateless; responses to allowed inbound traffic are subject to the rules for outbound traffic and vice versa.
- Block IP Addresses using network ACLs not Security Groups

What is VPC Flow Logs?

- You cannot enable flow logs for VPCs that are peered with your VPC unless the peer VPC is in your account
- You cannot tag a flow log
- After you have created a flow log, you cannot change its configuration; for example, you can't associate a different IAM role with the flow log.
- Traffic generated by instances when they contact the Amazon DNS server. If you use your own DNS server, then all traffic to that DNS server is logged.
- Traffic generated by a Windows instance for Amazon Windows license activation
- Traffic to and from 169.254.169.254 for instance metadata
- DHCP traffic
- Traffic to the reserved IP address for the default VPC router

In VPC with private and public subnets, database servers should ideally be launched into which subnet?

- With private and public subnets in VPC, database servers should ideally launch into private subnets.

What action is required to establish an Amazon VPC?

We need to assign a static internet-routable IP address to an Amazon VPC customer gateway.

Network ACLs: A network access control List (ACL) is an optional layer of security for your VPC that acts as a firewall for controlling traffic in and out of one or more subnets. You might set up the network ACLs with rules similar to your security groups in order to add an additional layer of security to your VPC. For more information about the differences between security groups and network ACLs.

Security Groups: A security group acts as a virtual firewall for your instance to control inbound and outbound traffic. When you launch an instance in a VPC, you can assign the instance to up to five security groups. Security groups act at the instance level, not at the subnet level. Therefore, each instance in a subnet in your VPC could be assigned to a different set of security groups. If you don't specify a particular group at launch time, the instance is automatically assigned to the default security group for the VPC.

Why use VPC in AWS?

Normally, each EX2 instance you launch is randomly assigned a public IP address in the amazon EC2 address space. VPC allows you to create an isolated portion of the AWS cloud and launc EC2 instances that have private address in the range of your choice. (10.0.0.0 for instance)

Can you describe the steps of create default VPC in AWS?

We can create a default VPC, we do the following to set it up for you: -

1.Create a default subnet in each availability zone

2.Creat an Internet gateway and connect it to your default VPC

3.Create a main route table for your default VPC with a rule that sends all traffic destined for the Internet gateway.

4.Create a default security group and associate it with your default VPC.

5.Create a default network access control list (ACL) and associate it with your default VPC.

6. Associate the default DHCP options set for your AWS account with your default VPC.

What are the three features provided by Amazon that you can increase and monitor the security?

Amazon VPC provides three features that you can use to increase and monitor the security for your VPC.

Security groups: Acts as a firewall for associated Amazon EC2 instances, controlling both inbound and outbound traffic at the instance level

Network Access Control List (ACLs) Act as a firewall for associated subnets, controlling both inboud and outbound traffic at the subnet level

Flow logs Capture information about the IP traffic going to and from network interfaces in your VPC.

What is the difference between Network ACLS and Security groups in AWS?

Network ACLS: A network access control list is an optional layer of security for your VPC that acts as a firewall for controlling traffic in and out of one or more subnets. You might set up Network ACLS with rules similar to your security groups in order to add an additional layer of security to your VPC. For more information about the differences between security groups and network ACLs.

Security Groups: A security group acts as an virtual firewall for your instance to control inbound and outbound traffic. When you launch an instance in a VPC, you can assign the instance up to five security groups. Security groups act at the instance level, not the subnet level. Therefore, each instance in a subnet in your VPC could be assigned to a different set of security groups. If you don't specify a particular group at launch time, the instance is automatically assigned to the default security group for the VPC.

The basic differences between network ACLs and Security groups are: -

1.Security Group operates at subnet level operates at the instance level

2.Supports allow rules and deny rules. Supports allow rules only is stateless.

3.Return traffic must be explicitly allowed by rules is stateful. Return traffic is automatically allowed, regardless of any rules

4.We process rules in number order when deciding whether to allow traffic. We evaluate all rules before deciding whether to allow traffic

5.Automatically applies to all instances in the subnets it is associated with. (not rely on security group). Applies to an instance only if someone specifies the security group when launching the instance or associates the security group with the instance later on.

What benefits to VPC security groups give you that EC2 security groups don't?

1.Being able to change the security group after the instance is launched

2.Being able to specify any protocol with a standard number, rather than just TCP, UDP or ICMP

What are the benefits of using a Virtual Private Cloud in AWS?

We can get following benefits by using Virtual Private Cloud (VPC) in an AWS account: We can assign Static IPv4 addresses to our instances in VPC. These static IP addresses will persist even after restarting an instance. We can even use IPv6 addresses with our instances in VPC. VPC also allows us to run our instances on single tenant hardware. We can define Access Control List (ACL) to add another layer of security to our instances in VPC. VPC also allows for changing the security group membership of instances while they are running.

If you want to launch Amazon Elastic Compute Cloud (EC2) instances and assign each instance a predetermined private IP address you should:

A. Launch the instance from a private Amazon Machine Image (AMI).

B. Assign a group of sequential Elastic IP address to the instances.

C. Launch the instances in the Amazon Virtual Private Cloud (VPC).

D. Launch the instances in a Placement Group.

Answer C

Explanation: The best way of connecting to your cloud resources (for ex- ec2 instances) from your own data center (for eg- private cloud) is a VPC. Once you connect your datacenter to the VPC in which your instances are present, each instance is assigned a private IP address which can be accessed from your datacenter. Hence, you can access your public cloud resources, as if they were on your own network.

Can I connect my corporate datacenter to the Amazon Cloud?

Yes, you can do this by establishing a VPN (Virtual Private Network) connection between your company's network and your VPC (Virtual Private Cloud), this will allow you to interact with your EC2 instances as if they were within your existing network.

Is it possible to change the private IP addresses of an EC2 while it is running/stopped in a VPC?

Primary private IP address is attached with the instance throughout its lifetime and cannot be changed, however secondary private addresses can be unassigned, assigned or moved between interfaces or instances at any point.

Why do you make subnets?

A. Because there is a shortage of networks

B. To efficiently utilize networks that have a large no. of hosts.

C. Because there is a shortage of hosts.

D. To efficiently utilize networks that have a small no. of hosts.

Answer B

Explanation: If there is a network which has a large no. of hosts, managing all these hosts can be a tedious job. Therefore, we divide this network into subnets (sub-networks) so that managing these hosts becomes simpler.

Which of the following is true?

A. You can attach multiple route tables to a subnet

B. You can attach multiple subnets to a route table

C. Both A and B

D. None of these.

Answer B

Explanation: Route Tables are used to route network packets, therefore in a subnet having multiple route tables will lead to confusion as to where the packet has to go. Therefore, there is only one route table in a subnet, and since a route table can have any no. of records or information, hence attaching multiple subnets to a route table is possible.

In CloudFront what happens when content is NOT present at an Edge location and a request is made to it?

A. An Error "404 not found" is returned

B. CloudFront delivers the content directly from the origin server and stores it in the cache of the edge location

C. The request is kept on hold till content is delivered to the edge location

D. The request is routed to the next closest edge location

Answer B

Explanation: CloudFront is a content delivery system, which caches data to the nearest edge location from the user, to reduce latency. If data is not present at an edge location, the first time the data may get transferred from the original server, but from the next time, it will be served from the cached edge.

If I'm using Amazon CloudFront, can I use Direct Connect to transfer objects from my own data center?

Yes. Amazon CloudFront supports custom origins including origins from outside of AWS. With AWS Direct Connect, you will be charged with the respective data transfer rates.

If my AWS Direct Connect fails, will I lose my connectivity?

If a backup AWS Direct connect has been configured, in the event of a failure it will switch over to the second one. It is recommended to enable Bidirectional Forwarding Detection (BFD) when configuring your connections to ensure faster detection and failover. On the other hand, if you have configured a

backup IPsec VPN connection instead, all VPC traffic will failover to the backup VPN connection automatically. Traffic to/from public resources such as Amazon S3 will be routed over the Internet. If you do not have a backup AWS Direct Connect link or a IPsec VPN link, then Amazon VPC traffic will be dropped in the event of a failure.

Route 53

What is Route 53? What are its Features>

Amazon Route 53 is a highly available and scalable cloud Domain Name System (DNS) web service.

It is designed to give developers and businesses an extremely reliable and cost-effective way to route end users to Internet applications by translating names like www.example.com into the numeric IP addresses like 192.0.2.1 that computers use to connect to each other. Amazon Route 53 is fully compliant with IPv6 as well.

Amazon Route 53 effectively connects user requests to infrastructure running in AWS – such as Amazon EC2 instances, Elastic Load Balancing load balancers, or Amazon S3 buckets

Features:

Traffic Flow

Easy-to-use and cost-effective global traffic management: route end users to the best endpoint for your application based on latency, health, and other considerations.

Latency Based Routing

Route end users to the AWS region that provides the lowest possible latency.

Geo DNS

Route end users to a particular endpoint that you specify based on the end user's geographic location.

Private DNS for Amazon VPC

Manage custom domain names for your internal AWS resources without exposing DNS data to the public Internet.

DNS Failover

Automatically route your website visitors to an alternate location to avoid site outages.

Health Checks and Monitoring

Amazon Route 53 can monitor the health and performance of your application as well as your web servers and other resources.

Domain Registration

Amazon Route 53 offers domain name registration services, where you can search for and register available domain names or transfer in existing domain names to be managed by Route 53.

Weighted Round Robin

Amazon Route 53 offers Weighted Round Robin (WRR) functionality.

Amazon ELB Integration

Amazon Route 53 is integrated with Elastic Load Balancing (ELB).

Management Console

Amazon Route 53 works with the AWS Management Console. This web-based, point-and-click, graphical user interface lets you manage Amazon Route 53 without writing any code at all.

What are the different routing policies available in Route 53?

Route 53 provides multiple options for creating a Routing policy. Some of these options are as follows:

Simple Routing: In this option, Route 53 will respond to DNS queries based on the values in resource record set.

Weighted Routing: In this policy, we can specify the weightage according to which multiple resources will handle the load. E.g. If we have two webservers, we can divide load in 40/60 ration on these servers.

Latency Routing: In this option, Route 53 will respond to DNS queries with the resources that provide the best latency.

Failover Routing: We can configure active/passive failover by using this policy. One resource will get all the traffic when it is up. Once first resource is down, all the traffic will be routed to second resource that is active during failover.

Geolocation Routing: As the name suggests, this policy works on the basis of location of end users from where requests originate.

You have an EC2 Security Group with several running EC2 instances. You changed the Security Group rules to allow inbound traffic on a new port and protocol, and then launched several new instances in the same Security Group. The new rules apply:

A. Immediately to all instances in the security group.

B. Immediately to the new instances only.

C. Immediately to the new instances, but old instances must be stopped and restarted before the new rules apply.

D. To all instances, but it may take several minutes for old instances to see the changes.

Answer A.

Explanation: Any rule specified in an EC2 Security Group applies immediately to all the instances, irrespective of when they are launched before or after adding a rule.

To create a mirror image of your environment in another region for disaster recovery, which of the following AWS resources do not need to be recreated in the second region? (Choose 2 answers)

A. Route 53 Record Sets

B. Elastic IP Addresses (EIP)

C. EC2 Key Pairs

D. Launch configurations

E. Security Groups

Answer A,B.

Explanation: Elastic IPs and Route 53 record sets are common assets therefore there is no need to replicate them, since Elastic IPs and Route 53 are valid across regions

A customer wants to capture all client connection information from his load balancer at an interval of 5 minutes, which of the following options should he choose for his application?

A. Enable AWS CloudTrail for the loadbalancer.

B. Enable access logs on the load balancer.

C. Install the Amazon CloudWatch Logs agent on the load balancer.

D. Enable Amazon CloudWatch metrics on the load balancer.

Answer A

Explanation: AWS CloudTrail provides inexpensive logging information for load balancer and other AWS resources This logging information can be used for analyses and other administrative work, therefore is perfect for this use case.

A customer wants to track access to their Amazon Simple Storage Service (S3) buckets and also use this information for their internal security and access audits. Which of the following will meet the Customer requirement?

A. Enable AWS CloudTrail to audit all Amazon S3 bucket access.

B. Enable server access logging for all required Amazon S3 buckets.

C. Enable the Requester Pays option to track access via AWS Billing

D. Enable Amazon S3 event notifications for Put and Post.

Answer A

Explanation: AWS CloudTrail has been designed for logging and tracking API calls. Also this service is available for storage, therefore should be used in this use case.

Which of the following are true regarding AWS CloudTrail? (Choose 2 answers)

A. CloudTrail is enabled globally

B. CloudTrail is enabled on a per-region and service basis

C. Logs can be delivered to a single Amazon S3 bucket for aggregation.

D. CloudTrail is enabled for all available services within a region.

Answer B, C

Explanation: Cloudtrail is not enabled for all the services and is also not available for all the regions. Therefore, option B is correct, also the logs can be delivered to your S3 bucket, hence C is also correct.

What happens if CloudTrail is turned on for my account but my Amazon S3 bucket is not configured with the correct policy?

CloudTrail files are delivered according to S3 bucket policies. If the bucket is not configured or is misconfigured, CloudTrail might not be able to deliver the log files.

How do I transfer my existing domain name registration to Amazon Route 53 without disrupting my existing web traffic?

You will need to get a list of the DNS record data for your domain name first, it is generally available in the form of a "zone file" that you can get from your existing DNS provider.

Once you receive the DNS record data, you can use Route 53's Management Console or simple web-services interface to create a hosted zone that will store your DNS records for your domain name and follow its transfer process.

It also includes steps such as updating the nameservers for your domain name to the ones associated with your hosted zone. For completing the process, you have to contact the registrar with whom you registered your domain name and follow the transfer process. As soon as your registrar propagates the new name server delegations, your DNS queries will start to get answered.

AWS Load Balancing

When should we use a Classic Load Balancer vs. an Application load balancer?

A Classic Load Balancer is used for simple load balancing of traffic across multiple EC2 instances. An Application Load Balancer is more suited for Microservices based architecture or container-based architecture. Mainly in these architectures there is a need to do load balancing as well as there is need to route traffic to multiple services on same EC2 instance.

How many subnets are needed for the Application Load Balancers?

You will need at least 2 public subnets in order to deploy an application load balancer

Explain how the buffer is used in Amazon web services?

The buffer is used to make the system more robust to manage traffic or load by synchronizing different component. Usually, components receive and process the requests in an unbalanced way, With the help of buffer, the components will be balanced and will work at the same speed to provide faster services.

Suppose you have an application where you have to render images and also do some general computing. From the following services which service will best fit your need?

A. Classic Load Balancer

B. Application Load Balancer

C. Both of them

D. None of these

Answer B

Explanation: You will choose an application load balancer, since it supports path-based routing, which means it can take decisions based on the URL, therefore if your task needs image rendering it will route it to a different instance, and for general computing it will route it to a different instance.

What is the difference between Scalability and Elasticity?

Scalability is the ability of a system to increase its hardware resources to handle the increase in demand. It can be done by increasing the hardware specifications or increasing the processing nodes.

Elasticity is the ability of a system to handle increase in the workload by adding additional hardware resources when the demand increases (same as scaling) but also rolling back the scaled resources, when the resources are no longer needed. This is particularly helpful in Cloud environments, where a pay per use model is followed.

How will you change the instance type for instances which are running in your application tier and are using Auto Scaling. Where will you change it from the following areas?

A. Auto Scaling policy configuration

B. Auto Scaling group

C. Auto Scaling tags configuration

D. Auto Scaling launch configuration

Answer D

Explanation: Auto scaling tags configuration, is used to attach metadata to your instances, to change the instance type you have to use auto scaling launch configuration.

You have a content management system running on an Amazon EC2 instance that is approaching 100% CPU utilization. Which option will reduce load on the Amazon EC2 instance?

A. Create a load balancer, and register the Amazon EC2 instance with it

B. Create a CloudFront distribution, and configure the Amazon EC2 instance as the origin

C. Create an Auto Scaling group from the instance using the CreateAutoScalingGroup action

D. Create a launch configuration from the instance using the CreateLaunchConfigurationAction

Answer A

Explanation: Creating alone an autoscaling group will not solve the issue, until you attach a load balancer to it. Once you attach a load balancer to an autoscaling group, it will efficiently distribute the load among all the instances. Option B – CloudFront is a CDN, it is a data transfer tool therefore will not help reduce load on the EC2 instance. Similarly, the other option – Launch configuration is a template for configuration which has no connection with reducing loads.

When should I use a Classic Load Balancer and when should I use an Application load balancer?

A Classic Load Balancer is ideal for simple load balancing of traffic across multiple EC2 instances, while an Application Load Balancer is ideal for microservices or container-based architectures where there is a need to route traffic to multiple services or load balance across multiple ports on the same EC2 instance.

What does Connection draining do?

A. Terminates instances which are not in use.

B. Re-routes traffic from instances which are to be updated or failed a health check.

C. Re-routes traffic from instances which have more workload to instances which have less workload.

Drains all the connections from an instance, with one click.

Answer B

Explanation: Connection draining is a service under ELB which constantly monitors the health of the instances. If any instance fails a health check or if any instance has to be patched with a software update, it pulls all the traffic from that instance and re-routes them to other instances.

When an instance is unhealthy, it is terminated and replaced with a new one, which of the following services does that?

A. Sticky Sessions

B. Fault Tolerance

C. Connection Draining

D. Monitoring

Answer B

Explanation: When ELB detects that an instance is unhealthy, it starts routing incoming traffic to other healthy instances in the region. If all the instances in a region becomes unhealthy, and if you have instances in some other availability zone/region, your traffic is directed to them. Once your instances become healthy again, they are re-routed back to the original instances.

CloudFront

Summarize the CloudFront Important Point?

Edge Location: This is the location where content will be cached. This is separate to an AWS Region/AZ

Origin: This is the origin of all the files that the CDN will distribute. This can be either an S3 bucket, an EC2 instance, an Elastic Load Balancer or Route 53

Distribution: This is the name given the CDN which consists of a collection of Edge Locations.

Web Distribution: Typically used for Websites

Edge locations are not just READ only, you can write to them too (put an object on to them)

Objects are cached for the life of the TTL (Time To Live)

You can clear cached objects, but you will be charged

How do we get higher performance in our application by using Amazon CloudFront?

If our application is content rich and used across multiple locations, we can use Amazon CloudFront to increase its performance.
Some of the techniques used by Amazon CloudFront are as follows:

Caching: Amazon CloudFront caches the copies of our application's content at locations closer to our viewers. By this caching our users get our content very fast. Also, due to caching the load on our main server decreases.

Edge / Regional Locations: CloudFront uses a global network of Edge and Regional edge locations to cache our content. These locations cater to almost all of the geographical areas across the world. Persistent Connections: In certain cases, CloudFront keeps persistent connections with the main server to fetch the content quickly.

Other Optimization: Amazon CloudFront also uses other optimization
optimization techniques like TCP initial congestion window etc to deliver high performance experience.

What is the mechanism behind Regional Edge Cache in Amazon CloudFront?

A Regional Edge Cache location lies between the main webserver and the global edge location. When the popularity of an object/content decreases, the global edge location may take it out from the cache. But Regional Edge location maintains a larger cache. Due to this the object/content can stay for long time in Regional Edge location. Due to this CloudFront does not have to go back to main webserver. When it does not find any object in Global Edge location it just looks for in Regional Edge location. This improves the performance for serving content to our users in Amazon CloudFront.

What are the benefits of Streaming content?

We can get following benefits by Streaming content: Control: We can provide more control to our users for what they want to watch. In a video streaming, users can select the locations in video where they want to start watching from. Content: With streaming our entire content does not stay at a user's device. Users gets only the part they are watching. Once the session is over, content is removed from the user's device. Cost: With streaming there is no need to download all the content to a user's device. A user can start viewing content as soon as some part is available for viewing. This saves costs since we do not have to download a large media file before starting each viewing session.

What is Lambda@Edge in AWS?

In AWS, we can use Lambda@Edge utility to solve the problem of low network latency for end users. In Lambda@Edge there is no need to provision or manage servers. We can just upload our Node.js code to AWS Lambda and create functions that will be triggered on CloudFront requests. When a request for content is received by CloudFront edge location, the Lambda code is ready to execute. This is a very good option for scaling up the operations in CloudFront without managing servers.

What are the different types of events triggered by Amazon CloudFront?

Different types of events triggered by Amazon CloudFront are as follows: Viewer Request: When an end user or a client program makes an HTTP/HTTPS request to CloudFront, this event is triggered at the Edge Location closer to the end user. Viewer Response: When a CloudFront server is ready to respond to a request, this event is triggered. Origin Request: When CloudFront server does not have the requested object in its cache, the request is forwarded to Origin server. At this time this event is triggered. Origin Response: When CloudFront server at an Edge location receives the response from Origin server, this event is triggered.

What is Geo Targeting in Amazon CloudFront?

In Amazon CloudFront we can detect the country from where end users are requesting our content. This information can be passed to our Origin server by Amazon CloudFront. It is sent in a new HTTP header. Based on different countries we can generate different content for different versions of the same content.

These versions can be cached at different Edge Locations that are closer to the end users of that country. In this way we are able to target our end users based on their geographic locations.

What are the main features of Amazon CloudFront?

Some of the main features of Amazon CloudFront are as follows: Device Detection Protocol Detection Geo Targeting Cache Behavior Cross Origin Resource Sharing Multiple Origin Servers HTTP Cookies Query String Parameters Custom SSL.

8.Management Tools

AWS CloudWatch Monitor Resources and Applications	**AWS Auto Scaling** Scale Multiple Resources to Meet Demand	**Amazon CloudFormation** Create and Manage Resources with Templates
AWS CloudTrail Track User Activity and API Usage	**AWS Config** Track Resource Inventory & Changes	**AWS OpsWorks** Automate Operations with Chef & Puppet
AWS Service Catalog Create and Use Standardized Products	**AWS System Manager** Gain Operational Insights and Take Action	**AWS Trusted Advisor** Optimize Performance and Security
AWS Personal Health Dashboard Personalized View of AWS Service Health		

CloudWatch

What are the main options available in Amazon CloudWatch?

Amazon CloudWatch is a monitoring service by Amazon for cloud-based AWS resources. Some of the main options in Amazon CloudWatch are as follows:

Logs: We can monitor and store logs generated by EC2 instances and our application in CloudWatch. We can store the log data for time period convenient for our use.

Dashboard: We can create visual Dashboards in the form of graphs to monitor our AWS resource in CloudWatch.

Alarms: We can set alarms in CloudWatch. These alarms can notify us by email or text when a specific metric crosses a threshold. These alarms can also detect the event when an Instance starts of shuts down.

Events: In CloudWatch we can also set up events that are triggered by an Alarm. These events can take an automated action when a specific Alarm is triggered.

CloudWatch is for performance Monitoring, CloudTrail is for auditing
Standard Monitoring = 5 Minutes | Detailed Monitoring = 1 Minute

Amazon CloudFormation

What is Amazon CloudFormation?

AWS CloudFormation is a service that helps you model and set up your AWS resources. So that you can spend less time managing those resources and more time focusing on your applications that run in AWS.

Codifies creation of stack of resources stack could be:

- ELB

- AutoScaling group

- EC2

- RDS [DataBase]

- All Connections between them.

What are the benefits of CloudFormation?

The benefits of Cloud Formation are: -

- Your Infrastructure as CODE.

- Can be version Controlled

- No more guessing.. Who did.. what.. where..

- Modularization

- Enforce "One way to deploy"

- CF costs nothing but for the resources created using it.

What are key points in Amazon CloudFormation?

The key point in Cloud Formation are:-

- If you sign up with Cloud formation means, you are signing up with ALL AWS SERVICES, that CF can create,

- Setting up Alarms!

- 200-300 templates are available to choose.

- Templates are JSON based

- Templates can accept "RunTime" parameters [Instance type / Key Pair].

What are the various sections of CloudFormation?

The Sections in the Templates of Cloud Formation:

There are 7 sections:

1. Version [of CF] - Which Year it was build, in date format.

2. Description - It gives details about the template.

3. Parameters - Runtime variables, like subnet, VPCID, InstanceTypes, DB Name, etc.

4. Mappings - Available types and allocated.

5. Resources [AWS to create] - ELB, WebServers, AppSvrs, etc.

6. Properties - Specific to Resources

7. OutPuts - Final outputs upon creation.

Auto-scaling

What is Auto-scaling?

How does Auto-scaling work in AWS? Auto-scaling is the ability of a system to scale itself automatically based on the triggers like- crashing of a server or low performance. AWS extensively supports Auto-scaling. It provides tools to create, configure and automatically start new instances without any manual intervention. We can set the thresholds at which new instances will come up. Or we can monitor the metrics like API response time, number of requests per seconds and based on these metrics, let the AWS provision and start new servers.

Auto- scaling is one of the remarkable features of AWS where it permits you to arrange and robotically stipulation and spin up fresh examples without the requirement for your involvement. This can be achieved by setting brinks and metrics to watch. If those entrances are overcome, a fresh example of your selection will be configured, spun up and copied into the weight planner collection.

What are lifecycle hooks used for in Autoscaling?

A. They are used to do health checks on instances

B. They are used to put an additional wait time to a scale in or scale out event.

C. They are used to shorten the wait time to a scale in or scale out event

D. None of these

Answer B

Explanation: Lifecycle hooks are used for putting wait time before any lifecycle action i.e launching or terminating an instance happens. The purpose of this wait time, can be anything from extracting log files before terminating an instance or installing the necessary software's in an instance before launching it.

A user has setup an Auto Scaling group. Due to some issue the group has failed to launch a single instance for more than 24 hours. What will happen to Auto Scaling in this condition?

A. Auto Scaling will keep trying to launch the instance for 72 hours

B. Auto Scaling will suspend the scaling process

C. Auto Scaling will start an instance in a separate region

D. The Auto Scaling group will be terminated automatically

Answer B

Explanation: Auto Scaling allows you to suspend and then resume one or more of the Auto Scaling processes in your Auto Scaling group. This can be very useful when you want to investigate a configuration problem or other issue with your web application, and then make changes to your application, without triggering the Auto Scaling process.

OpsWorks

How is AWS OpsWorks different than AWS CloudFormation?

OpsWorks and CloudFormation both support application modelling, deployment, configuration, management and related activities. Both support a wide variety of architectural patterns, from simple web applications to highly complex applications. AWS OpsWorks and AWS CloudFormation differ in abstraction level and areas of focus.

AWS CloudFormation is a building block service which enables customer to manage almost any AWS resource via JSON-based domain specific language. It provides foundational capabilities for the full breadth of AWS, without prescribing a particular model for development and operations. Customers define templates and use them to provision and manage AWS resources, operating systems and application code.

In contrast, AWS OpsWorks is a higher-level service that focuses on providing highly productive and reliable DevOps experiences for IT administrators and ops-minded developers. To do this, AWS OpsWorks employs a configuration management model based on concepts such as stacks and layers, and provides integrated experiences for key activities like deployment, monitoring, auto-scaling, and automation. Compared to AWS CloudFormation, AWS OpsWorks supports a narrower range of application-oriented AWS resource types including Amazon EC2 instances, Amazon EBS volumes, Elastic IPs, and Amazon CloudWatch metrics.

A company needs to monitor the read and write IOPS for their AWS MySQL RDS instance and send real-time alerts to their operations team. Which AWS services can accomplish this?

A. Amazon Simple Email Service

B. Amazon CloudWatch

C. Amazon Simple Queue Service

D. Amazon Route 53

Answer B

Explanation: Amazon CloudWatch is a cloud monitoring tool and hence this is the right service for the mentioned use case. The other options listed here are used for other purposes for example route 53 is used for DNS services, therefore CloudWatch will be the apt choice.

What happens when one of the resources in a stack cannot be created successfully in AWS OpsWorks?

When an event like this occurs, the “automatic rollback on error” feature is enabled, which causes all the AWS resources which were created successfully till the point where the error occurred to be deleted.

This is helpful since it does not leave behind any erroneous data, it ensures the fact that stacks are either created fully or not created at all. It is useful in events where you may accidentally exceed your limit of the no. of Elastic IP addresses or maybe you may not have access to an EC2 AMI that you are trying to run etc.

What automation tools can you use to spin up servers?

The API tools can be used for spinup services and also for the written scripts.

- Those scripts could be coded in Perl, bash or other languages of your preference.
- There is one more option that is patterned administration and stipulating tools such as a dummy or improved descendant.
- A tool called Scalr can also be used and finally we can go with a controlled explanation like a Rightscale.

9.Application Integration & Customer Engagement

AWS Step Functions Coordinate Distributed Applications	**AWS Simple Queue Service (SQS)** Managed Message Queues	**Amazon CloudFormation** Pub/Sub, Mobile Push and SMs
Amazon MQ Managed Message Broker for ActiveMQ		

Amazon Connect Cloud based Contact Center	**Amazon Pinpoint** Push Notifications for Mobile Apps	**Amazon Simple Email Service (SES)** Email Sending and Receiving

What is SQS?

Amazon SQS is a web service that gives you access to a message queue that can be used to store messages while waiting for a computer to process them

Amazon SQS is a distributed queue system that enables web service applications to quickly and reliably queue messages that one component in the application generates to be consumed by another component. A queue is a temporary repository for messages that are awaiting processing.

Using Amazon SQS, you can decouple the components of an application so they run independently, with Amazon SQS easing message management between components. Any component of a distributed application can store messages in a fail-safe queue. Message can contain up to 256 KB text in any format. Any component can later retrieve the messages programmatically using the Amazon SQS API.

The queue act as a buffer between the component producing and saving data, and the component receiving the data for processing. This means the queue resolves issues that arise if the producer is producing work faster than the consumer can process it, or if the producer or consumer are only intermittently connected to the network.

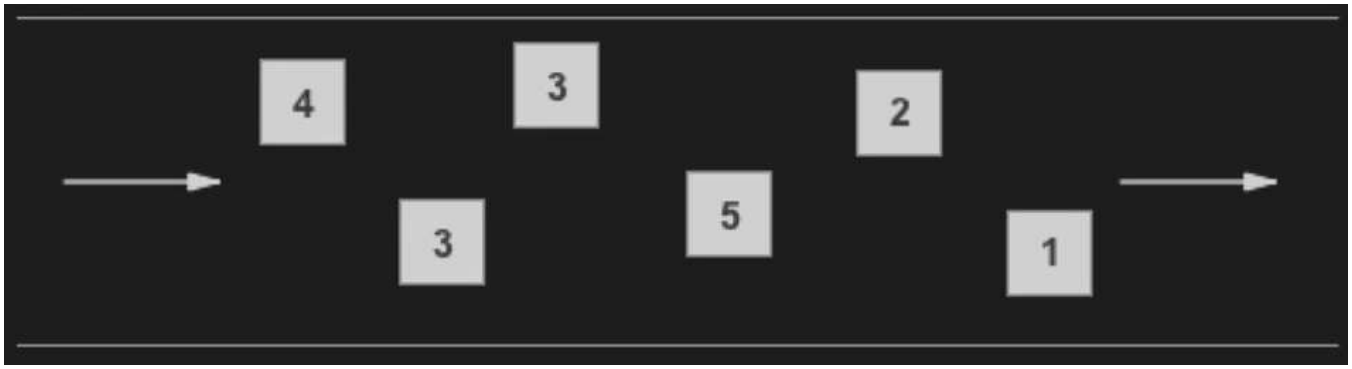
What are the Queue Types?

There are two types of Queue namely: -

- Standard Queue (default)
- FIFO Queues

Standard Queues

Amazon SQS offers standard as the default queue type. A standard queue lets you have a nearly-unlimited number of transactions per second.



Standard queues guarantee that a message is delivered at least once.

However, occasionally (because of the highly distributed architecture that allows high throughput), more than one copy of a message might be delivered out of order. Standard queues provide the best effort ordering which ensures that messages are generally delivered in the same order as they sent.

FIFO Queues

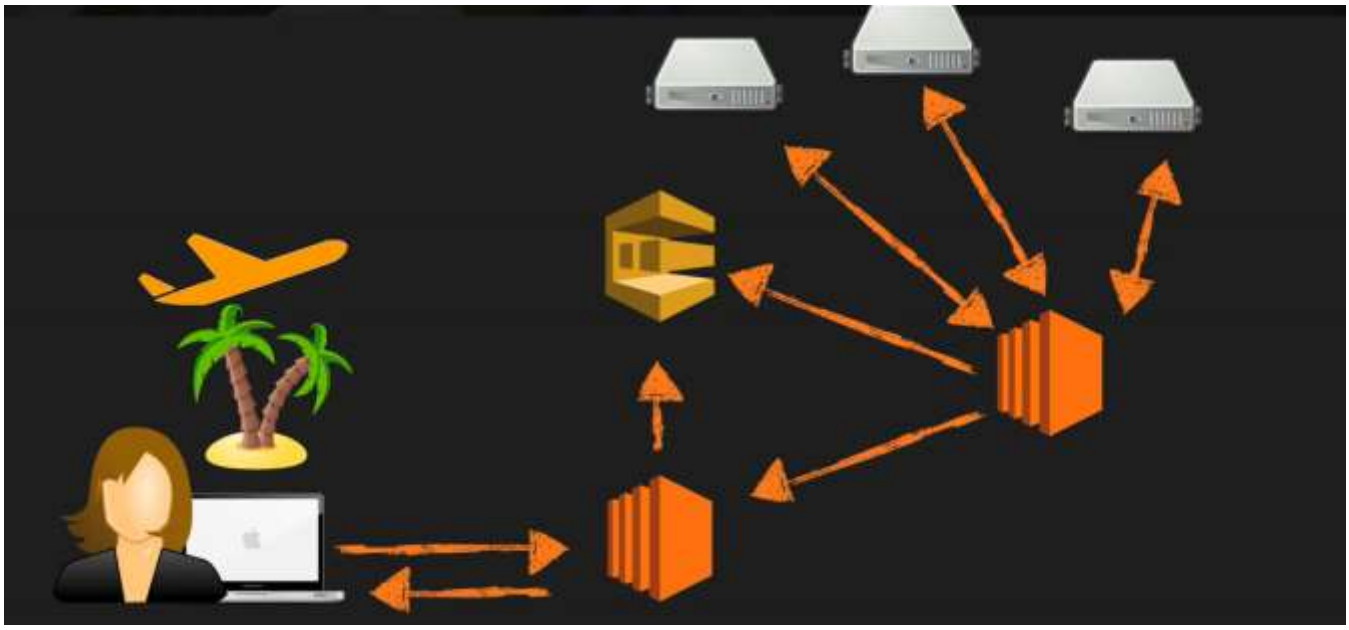
The FIFO queue complements the standard queue. The most important features of this queue type are FIFO (First-In-First-Out) delivery and exactly once processing: The order in which messages are sent and received is strictly preserved and a message and deletes it; duplicates are not introducing into the queue.

FIFO queues also support message groups that allow multiple ordered message groups within a single queue. FIFO queues are limited to 300 transactions per second (TPS) but have all the capabilities of standard queues.

Summarize the SQS Key Points

- SQS is pull based, not pushed base
- Messages are 256 KB in size
- Messages can be kept in the queue from 1 minute to 14 days. The default is 4 days
- Visibility Time Out is the amount of time that the message is invisible in the SQS queue after a reader picks up that message
- Provided the job is processed before the visibility time out expires, the message will then be deleted from the queue. If the job is not processed within that time, the message will become visible again and another reader will process it. This could result in the same message being delivered twice.
- Visibility time out maximum is 12 hours
- SQS guarantees that your messages will be processed at least once.
- Amazon SQS long polling is a way to retrieve messages from your Amazon SQS queues. While the regular short polling returns immediately, even if the message queue being polled is empty, long polling does not return a response until a message arrives in the message queue or the long poll time out
- Queues can either be standard or FIFO

How SQS will be more effective in Ecommerce Travel Website?



How to use Amazon SQS?

Amazon SQS is a message passing mechanism that is used for communication between different connectors that are connected with each other. It also acts as a communicator between various components of Amazon. It keeps all the different functional components together. This functionality helps different components to be loosely coupled and provide an architecture that is more failure resilient system.

What are the advantages of messaging queues to decouple components?

Messaging queues is a very good approach to build a decoupled system. In a messaging queue there is asynchronous communication. The components are connected by using a queue or a buffer.

It provides following advantages:

Concurrency: More than one component can concurrently access the messaging queue.

High Availability: Since messages are persisted in the queue, a component can re-read a message even in case of failure. This leads to higher availability of the whole system.

Load Spikes: In case of sudden increase in load, a messaging queue can gracefully handle the scenario. It will collect all the messages and process these asynchronously.

How can we implement Message Queue based system in AWS?

Following techniques can be used to build a Message Queue based system in AWS: Amazon SQS: We can use Amazon SQS as a queue/buffer between components. In this way different components can be isolated. Service Interface: We can design every component to expose a service interface and make it responsible for its own scalability.

The component will interact with other components asynchronously by using SQS. Machine Image: We can put the logical parts of software in the Amazon Machine mage for that component, so that it can be deployed in an automated manner. Stateless: Also, is it important the make the applications stateless for asynchronous communication.

What is SWF?

Amazon Simple Workflow Service (Amazon SWF) is a web service that makes it easy to coordinate work across distributed application components.

Amazon SWF enables applications for a range of use cases, including media processing, web application back-ends, business process workflows, and analytics pipelines, to be designed as a coordination of tasks. Tasks represent invocations of various processing steps in an application which can be performed by executable code, web service calls, human actions, and scripts.

Compare SWF Vs SQS?

- **SQS has a retention period of 14 days, SWF up to 1 year for workflow executions**
- **Amazon SWF presents a task-oriented API, whereas Amazon SQS offers a message-oriented API**
- **Amazon SWF ensures that a task is assigned only once and is never duplicated. With Amazon SQS, you need to handle the duplicated messages and may also need to ensure that a message is processed only once.**
- **Amazon SWF keeps track of all the tasks and events in an application. With Amazon SQS, you need to implement your own application-level tracking, especially if your application uses multiple queues.**

SWF Actors

Workflow Starters - An Application that can initiate (start) a workflow. Could be your e-commerce website when placing an order or a mobile app searching for bust times

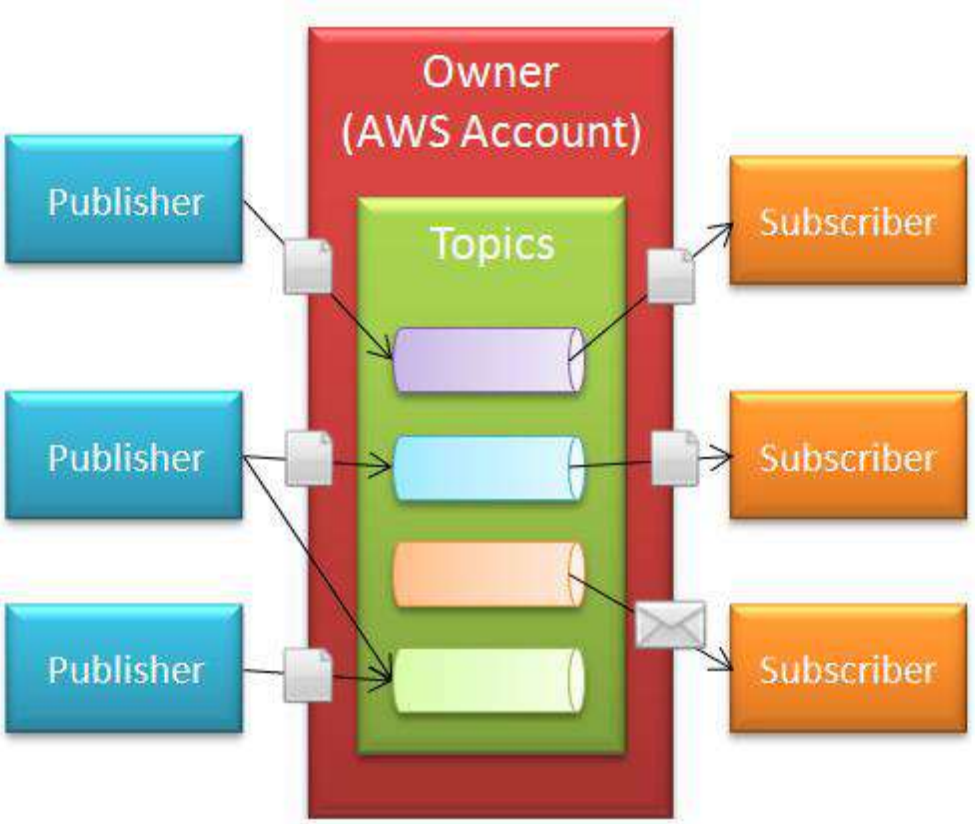
Deciders - Control the flow of activity tasks in a workflow execution. If something has finished in a workflow (or fails) a Decider decides what to do next

Activity Workers - Carry out the activity tasks

What is SNS?

Amazon Simple Notification Service (Amazon SNS) is a web service that makes it easy to set up, operate, and send notifications from the cloud. It provides developers with a highly scalable, flexible, and cost-effective capability to publish message from an application and immediately deliver them to subscribers or other applications.

Push notifications to Apple, Google, Fire OS, and Windows devices, as well as Android devices in China with Baidu Cloud Push.



Besides pushing cloud notifications directly to mobile devices, Amazon SNS can also deliver notifications by SMS text message or email, to Amazon Simple Queue Service (SQS) queues, or to any HTTP endpoint.

SNS notifications can also trigger Lambda functions. When a message is published to an SNS topic that has a Lambda function subscribed to it, the Lambda function is invoked with the payload of the published message. The Lambada function receives the message payload as an input parameter and can manipulate the information in the message, publish the message to other SNS topics, or send the message to other AWS services.

SNS allows you to group multiple recipients using topics. A topic is an "Access Point" for allowing recipients to dynamically subscribe for identical copies of the same notification. One topic can support

deliveries to multiple endpoint types - For example, you can group together iOS, Android and SMS recipients. When you publish once to a topic, SNS delivers appropriately formatted copies of your message to each subscriber.

To prevent messages from being lost, all messages published to Amazon SNS are stored redundantly across multiple availability zones.

What are the benefits of SNS?

- Instantaneous, push-based delivery (no polling)
- Simple APIs and easy integration with applications
- Flexible message delivery over multiple transport protocols
- Inexpensive, Pay-as-you-go model with no up-front costs
- Web-based AWS Management Console offers the simplicity of a point-and -click interface

Who are all the SNS Subscribers?

- HTTP
- HTTPS
- Email
- Email-JSON
- SQS
- Application
- Lambda

What is the difference between Amazon SNS and Amazon SQS?

- Amazon SNS allows applications to send time-critical messages to multiple subscribers through a "push" mechanism, eliminating the need to periodically check or "poll" for updates.

- Amazon SQS is a message queue service used by distributed applications to exchange messages through a polling model and can be used to decouple sending and receiving components—without requiring each component to be concurrently available.

- Amazon SQS stands for Simple Queue Service. Whereas, Amazon SNS stands for Simple Notification Service. SQS is used for implementing Messaging Queue solutions in an application. We can decouple the applications in cloud by using SQS.

- Since all the messages are stored redundantly in SQS, it minimizes the chance of losing any message. SNS is used for implementing Push notifications to a large number of users. With SNS we can deliver messages to Amazon SQS, AWS Lambda or any HTTP endpoint. Amazon SNS is widely used in sending messages to mobile devices as well. It can even send SMS messages to cell phones.

In Short

- Both messaging services in AWS
- SNS - Push
- SQS - Polls (Pulls)

How about SNS Pricing?

- Users pay $0.50 per 1 million Amazon SNS Requests
- $0.06 per 100,000 Notification deliveries over HTTP
- $0.75 per 100 Notification deliveries over SMS
- $2.00 per 100,000 Notification deliveries over Email

What is SES?

Amazon SES is an email platform that provides an easy, cost-effective way for you to send and receive email using your own email addresses and domains.

For example, you can send marketing emails such as special offers, transactional emails such as order confirmations, and other types of correspondence such as newsletters. When you use Amazon SES to receive mail, you can develop software solutions such as email autoresponders, email unsubscribe systems, and applications that generate customer support tickets from incoming emails. You only pay for what you use, so you can send and receive as much or as little email as you like.

Why use Amazon SES?

Building a large-scale email solution is often a complex and costly challenge for a business. You must deal with infrastructure challenges such as email server management, network configuration, and IP address reputation. Additionally, many third-party email solutions require contract and price negotiations, as well as significant up-front costs. Amazon SES eliminates these challenges and enables you to benefit from the years of experience and sophisticated email infrastructure Amazon.com has built to serve its own large-scale customer base.

10.Analytics

Amazon Athena Query Data in S3 using SQL	**Amazon EMR** Hosted Hadoop Framework	**Amazon CloudSearch** Managed Search Service
Amazon Elasticsearch Service Run and Scale Elasticsearch Clusters	**Amazon Kinesis** Work with Real-time Streaming Data	**Amazon Kinesis** Fast, Simple, Cost-Effective Data Warehousing
Amazon Kinesis Fast Business Analytics Service	**AWS Data Pipeline** Orchestration Service for Periodic, Data-Driven Workflows	**Amazon Glue** Prepare and Load Data

What is Amazon EMR?

Amazon Elastic MapReduce (Amazon EMR) is a managed cluster platform that simplifies running big data frameworks, such as Apache Hadoop and Apache Spark, on AWS to process and analyze vast amounts of data. By using these frameworks and related open-source projects, such as Apache Hive and Apache Pig, you can process data for analytics purposes and business intelligence workloads. Additionally, you can use Amazon EMR to transform and move large amounts of data into and out of other AWS data stores and databases, such as Amazon Simple Storage Service (Amazon S3) and Amazon DynamoDB.

What is Amazon Elastic Map Reduce (EMR)?

Amazon provides support for running MapReduce algorithm by Amazon Elastic MapReduce platform. Amazon EMR can be used to run the big data-based software like- Apache Hadoop, Apache Spark etc. in AWS cloud. Amazon EMR can be used for running on large datasets as well as for analyzing the big data. It supports business analytics and related functions. In addition, we can use Amazon EMR with Amazon S3 to transform very large data sets and databases. It also works very well with NoSQL DB like DynamoDB.

What are the main features of Amazon CloudSearch?

Amazon CloudSearch is mainly used for implementing a search solution of a website or application in AWS. It is highly scalable service and very easy to manage. It can index and search structured data as well as plain text.

Main feature of Amazon CloudSearch is: -

Prefix search, Full text search, Boolean search, Term boosting, Range search, Autocomplete Suggestions, Faceting, Highlighting

What is AWS Data Pipeline?

AWS Data Pipeline is a web service for automating the transformation of large scale data in AWS cloud.

We can use AWS Data Pipeline to define data-driven workflows. In such a workflow tasks follow a sequential pattern, where one task waits for completion of another task in the flow.

What is the difference between AWS Data Pipeline and Amazon Simple Workflow Service?

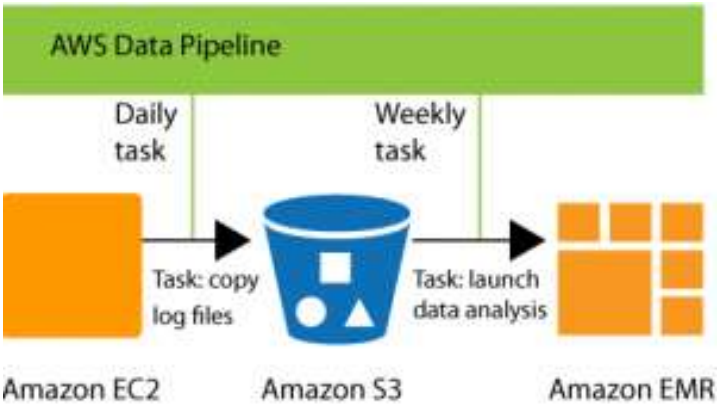
AWS Data pipeline is mainly used for data driven workflows that are popular in Big Data systems. AWS Data pipeline can easily copy data between different data stores and it can execute data transformations. To create such data flows, little programming knowledge is required. Amazon Simple Workflow Service

(SWS) is mainly used for process automation. It can easily coordinate work across distributed application components.

We can do media processing, backend flows, analytics pipelines etc. with SWS. So, it is not limited to just Data driven flows.

What is AWS Data Pipeline? and what are the components of AWS Data Pipeline?

AWS Data Pipeline is a web service that you can use to automate the movement and transformation of data. With AWS Data Pipeline, you can define data-driven workflows, so that tasks can be dependent on the successful completion of previous tasks.



The following components of AWS Data Pipeline work together to manage your data:

- A pipeline definition specifies the business logic of your data management. For more information, see Pipeline Definition File Syntax.
- A pipeline schedules and runs tasks. You upload your pipeline definition to the pipeline, and then activate the pipeline. You can edit the pipeline definition for a running pipeline and activate the pipeline again for it to take effect. You can deactivate the pipeline, modify a data source, and then activate the pipeline again. When you are finished with your pipeline, you can delete it.
- Task Runner polls for tasks and then performs those tasks. For example, Task Runner could copy log files to Amazon S3 and launch Amazon EMR clusters. Task Runner is installed and runs automatically on resources created by your pipeline definitions. You can write a custom task runner application, or you can use the Task Runner application that is provided by AWS Data Pipeline. For more information, see Task Runners.

What is Amazon Kinesis Firehose?

Amazon Kinesis Firehose is a fully managed service for delivering real-time streaming data to destinations such as Amazon Simple Storage Service (Amazon S3) and Amazon Redshift.

What Is Amazon CloudSearch and its features?

Amazon CloudSearch is a fully managed service in the cloud that makes it easy to set up, manage, and scale a search solution for your website or application.

You can use Amazon CloudSearch to index and search both structured data and plain text. Amazon CloudSearch features:

- **Full text search with language-specific text processing**
- **Boolean search**
- **Prefix searches**
- **Range searches**
- **Term boosting**
- **Faceting**
- **Highlighting**
- **Autocomplete Suggestions**

What is an activity in AWS Data Pipeline?

An activity AWS Data Pipeline is an Action that is initiated as a part of the pipeline. Some of the activities are: -

- **Elastic MapReduce (EMR)**
- **Hive jobs**
- **Data copies**
- **SQL queries**
- **Command-line scripts**

What is a schedule in AWS Data Pipeline?

In AWS Data Pipeline we can define a Schedule. The Schedule contains the information about when will pipeline activities run and with what frequency. All schedules have a start date and a frequency. E.g. One schedule can be run every day starting Mar 1, 2016, at 6am. Schedules may also have an end date, after which the AWS Data Pipeline service will not execute any activity.

What is the main framework behind Amazon Elastic MapReduce (EMR)?

Apache Hadoop is the main framework behind Amazon EMR. It is a distributed data processing engine. Hadoop is Open source Java based software framework. It supports data-intensive distributed

applications running on large clusters of commodity hardware. Hadoop is based on MapReduce algorithm in which data is divided into multiple small fragments of work. Each of these tasks can be executed on any node in the cluster. In AWS EMR, Hadoop is run on the hardware provides by AWS cloud.

What are different states in AWS EMR cluster?

AWS EMR has following cluster states: STARTING – In this state, cluster provisions, starts, and configures EC2 instances BOOTSTRAPPING – In this state cluster is executing the Bootstrap process RUNNING – State in which cluster is currently being run WAITING – In this state cluster is currently active, but there are no steps to run TERMINATING - Shut down of cluster has started TERMINATED - The cluster is shut down without any error TERMINATED_WITH_ERRORS - The cluster is shut down with errors.

What are the use cases for Amazon Kinesis Streams?

Amazon Kinesis Streams helps in creating applications that deal with streaming data. Kinesis streams can work with data streams up to terabytes per hour rate. Kinesis streams can handle data from thousands of sources. We can also use Kinesis to produce data for use by other Amazon services. Some of the main use cases for Amazon Kinesis Streams are as follows:

Real-time Analytics: At times for real-time events like-Big Friday sale or a major game event, we get a large amount of data in a short period of time. Amazon Kinesis Streams can be used to perform real time analysis on this data and make use of this analysis very quickly. Prior to Kinesis, this kind of analysis would take days. Whereas now within a few minutes we can start using the results of this analysis.

Gaming Data: In online applications, thousands of users play and generate a large amount of data. With Kinesis, we can use the streams of data generated by an online game and use it to implement dynamic features based on the actions and behavior of players.

Log and Event Data: We can use Amazon Kinesis to process the large amount of Log data that is generated by different devices. We can build live dashboards, alarms, triggers based on this streaming data by using Amazon Kinesis.

Mobile Applications: In Mobile applications, there is wide variety of data available due to the large number of parameters like- location of mobile, type of device, time of the day etc. We can use Amazon Kinesis Streams to process the data generated by a Mobile App. The output of such processing can be used by the same Mobile App to enhance user experience in real time.

11.Business Productivity

Alexa for Business **Empower your organization with Alexa**	**Amazon Chime** **Frustration-free Meetings, Video Calls, and Chat**	**Amazon WorkDocs** **Enterprise Storage and Sharing Service**
Amazon WorkMail **Secure and Managed Business Email and Calendaring**		

What is WorkDocs?

Amazon WorkDocs is a fully managed, secure enterprise storage and sharing service with strong administrative controls and feedback capabilities that improve user productivity.

What is WorkMail?

Amazon WorkMail is a managed email and calendaring service that offers strong security controls and support for existing desktop and mobile clients.

Which AWS responsible for managed email and calendaring?

WorkMail is a managed email and calendaring service with strong security controls and support for existing desktop and mobile email clients. You can access their email, contacts, and calendars wherever you use Microsoft Outlook, your browser, or your iOS and Android mobile devices. You can integrate Amazon WorkMail with your existing corporate directory and control both the keys that encrypt your data and the location where your data is stored.

12.Desktop & App Streaming

Amazon WorkSpaces **Desktop Computing Service**	**Amazon AppStream 2.0** **Stream Desktop Applications Securely to a Browser**	

What is WorkSpaces?

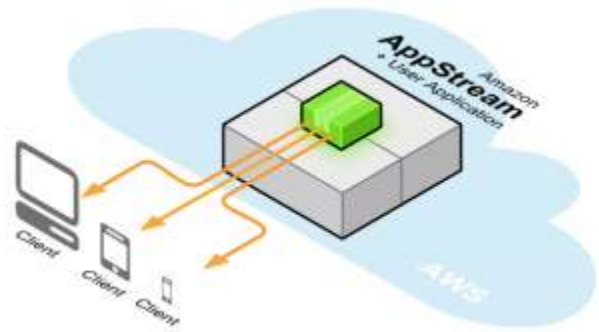
Amazon WorkSpaces is a fully managed desktop computing service in the cloud.

What is AppStream?

Amazon AppStream lets you stream resource intensive applications and games from the cloud to multiple end-user devices.

What is Amazon AppStream and advantage of using AppStreaming?

Amazon AppStream is an application streaming service that lets you stream your existing resource-intensive applications from the cloud without code modifications.



Advantages of Streaming Your Application:

Interactively streaming your application from the cloud provides several benefits:

- **Remove Device Constraints – You can leverage the compute power of AWS to deliver experiences that wouldn't normally be possible due to the GPU, CPU, memory or physical storage constraints of local devices.**
- **Support Multiple Platforms – You can write your application once and stream it to multiple device platforms. To support a new device, just write a small client to connect to your streaming application.**
- **Fast and Easy Updates – Because your streaming application is centrally managed by Amazon AppStream, updating your application is as simple as providing a new version of your streaming application to Amazon AppStream. You can immediately upgrade all of your customers without any action on their part.**
- **Instant On – Streaming your application with Amazon AppStream lets your customers start using your application or game immediately, without the delays associated with large file downloads and time-consuming installations.**
- **Improve Security – Unlike traditional boxed software and digital downloads, where your application is available for theft or reverse engineering, Amazon AppStream stores your streaming application binary securely in AWS datacenters.**

- **Automatic Scaling – You can use Amazon AppStream to specify capacity needs, and then the service automatically scales your streamed application and connects customers' devices to it.**

What are the advantages of using AppStream in AWS?

We can use Amazon AppStream to stream our resource-intensive applications from AWS cloud.

The Main advantages of AppStream are: -

Device Constraints: Since WS provides computing environment, there are no device constraints like CPU, memory etc. on the application.

Fast and Easy Upgrade: since AppStream manages the application, it is very easy to upgrade and send updates to application. New version of the application can be immediately released.

Multiple Platforms: Since AppStream supports multiple platforms, we can write the application code one time and stream it to multiple device platforms. We just have to write a small client to support a new device.

Instant On: Amazon AppStream can be used for starting the application immediately. There is no delay related to large file download or installation in AppStream. Security: Application in AppStream is safe from theft or any other form of plagiarism. Applications are stored very securely in AppStream.

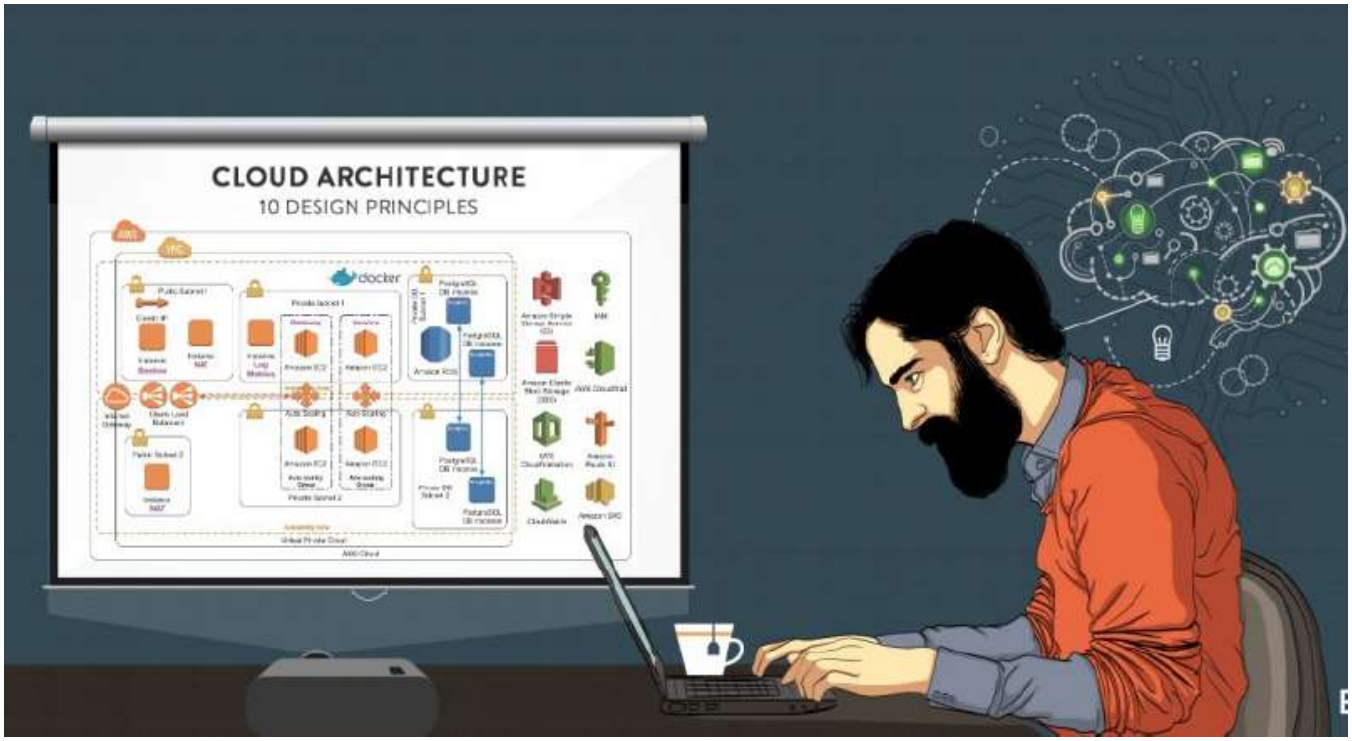
Auto-Scaling: Amazon AppStream supports Autoscaling based on the need as well as spikes in traffic requests.

13.AWS Architecture

Design Principles for AWS Cloud Architecture

Cloud computing is one of the boons of technology, making storage and access of documents easier and efficient. For it to be reliable, the cloud architecture need to be impeccable. It needs to be secure, high performing and cost efficient. A good cloud architecture design should take advantage of some of the inherent strengths of cloud computing – elasticity, ability to automate infrastructure management etc. Cloud architecture design needs to be well thought out because it forms the backbone of a vast network. It cannot be arbitrarily designed.

There are certain principles that one needs to follow to make the most of the tremendous capabilities of the Cloud. Here are ten design principles that you must consider while architecting for AWS cloud.



1. Automate Everything

Unlike traditional IT infrastructure, Cloud enables automation of a number of events, improving both your system's stability and the efficiency of your organization. Some of the AWS resources you can use to get automated are: -

AWS Elastic Beanstalk: This resource is the fastest and simplest way to get an application up and running on AWS. You can simply upload their application code and the service automatically handles all the details, such as resource provisioning, load balancing, auto scaling, and monitoring.

Amazon EC2 Auto recovery: You can create an Amazon CloudWatch alarm that monitors an Amazon EC2 instance and automatically recovers it if it becomes impaired. A word of caution though – During instance recovery, the instance is migrated through an instance reboot, and any data that is in-memory is lost.

Auto Scaling: With Auto Scaling, you can maintain application availability and scale your Amazon EC2 capacity up or down automatically according to conditions you define.

Amazon CloudWatch Alarms: You can create a CloudWatch alarm that sends an Amazon Simple Notification Service (Amazon SNS message when a particular metric goes beyond a specified threshold for a specified number of periods.

Amazon CloudWatch Events: The CloudWatch service delivers a near real-time stream of system events that describe changes in AWS resources. Using simple rules that you can set up in a couple of minutes, you can easily route each type of event to one or more targets: AWS Lambda functions, Amazon Kinesis streams, Amazon SNS topics, etc.

AWS OpsWorks Lifecycle events: AWS OpsWorks supports continuous con□guration through lifecycle events that automatically update your instances configuration to adapt to environment changes. These events can be used to trigger Chef recipes on each instance to perform specific configuration tasks.

AWS Lambda Scheduled events: These events allow you to create a Lambda function and direct AWS Lambda to execute it on a regular schedule.

As an architect for the AWS Cloud, these automation resources are a great advantage to work with.

2. Implement loose coupling

IT systems should ideally be designed in a way that reduces interdependencies. Your components need to be loosely coupled to avoid changes or failure in one of the components from affecting others.

Your infrastructure also needs to have well defined interfaces that allow the various components to interact with each other only through specific, technology-agnostic interfaces. Modifying any underlying operations without affecting other components should be made possible.

In addition, by implementing service discovery, smaller services can be consumed without prior knowledge of their network topology details through loose coupling. This way, new resources can be launched or terminated at any point of time.

Loose coupling between services can also be done through asynchronous integration. It involves one component that generates events and another that consumes them. The two components do not integrate through direct point-to-point interaction, but usually through an intermediate durable storage layer. This approach decouples the two components and introduces additional resiliency. So, for example, if a process that is reading messages from the queue fails, messages can still be added to the queue to be processed when the system recovers.

Lastly, building applications in such a way that they handle component failure in a graceful manner helps you reduce impact on the end users and increase your ability to make progress on your offline procedures.

3.Focus on services, not servers

A wide variety of underlying technology components are required to develop manage and operate applications. Your architecture should leverage a broad set of compute, storage, database, analytics, application, and deployment services. On AWS, there are two ways to do that. The first is through managed services that include databases, machine learning, analytics, queuing, search, email, notifications, and more. For example, with the Amazon Simple Queue Service (Amazon SQS) you can offload the administrative burden of operating and scaling a highly available messaging cluster, while paying a low price for only what you use. Not only that, Amazon SQS is inherently scalable.

The second way is to reduce the operational complexity of running applications through server-less architectures. It is possible to build both event-driven and synchronous services for mobile, web, analytics, and the Internet of Things (IoT) without managing any server infrastructure.

4. Database is the base of it all

On AWS, managed database services help remove constraints that come with licensing costs and the ability to support diverse database engines that were a problem with the traditional IT infrastructure? You need to keep in mind that access to the information stored on these databases is the main purpose of cloud computing.

There are three different categories of databases to keep in mind while architecting: -

Relational databases: Data here is normalized into tables and also provided with powerful query language, flexible indexing capabilities, strong integrity controls, and the ability to combine data from multiple tables in a fast and efficient manner. They can be scaled vertically and are highly available during failovers (designed for graceful failures).

NoSQL databases: These databases trade some of the query and transaction capabilities of relational databases for a more flexible data model that seamlessly scales horizontally. NoSQL databases utilize a variety of data models, including graphs, key-value pairs, and JSON documents. NoSQL databases are widely recognized for ease of development, scalable performance, high availability, and resilience.

Introduce redundancy to remove single points of failure, by having multiple resources for the same task. Redundancy can be implemented in either standby mode (functionality is recovered through failover while the resource remains unavailable) or active mode (requests are distributed to multiple redundant compute resources, and when one of them fails, the rest can simply absorb a larger share of the workload).

Data warehouse: A specialized type of relational database, optimized for analysis and reporting of large amounts of data. It can be used to combine transactional data from disparate sources making them available for analysis and decision-making.

5. Be sure to remove single points of failure

A system is highly available when it can withstand the failure of an individual or multiple component (e.g., hard disks, servers, network links etc.). You can think about ways to automate recovery and reduce disruption at every layer of your architecture. This can be done with the following processes:

It is crucial to have a durable data storage that protects both data availability and integrity. Redundant copies of data can be introduced either through synchronous, asynchronous or Quorum based replication.New item

Detection and reaction to failure should both be automated as much as possible.

Automated Multi –Data Center resilience is practiced through Availability Zones across data centers that reduce the impact of failures. Fault isolation improvement can be made to traditional horizontal scaling by Sharding (a method of grouping instances into groups called shards, instead of sending the traffic from all users to every node like in the traditional IT structure.)

Introduce redundancy to remove single points of failure, by having multiple resources for the same task. Redundancy can be implemented in either standby mode (functionality is recovered through failover while the resource remains unavailable) or active mode (requests are distributed to multiple redundant compute resources, and when one of them fails, the rest can simply absorb a larger share of the workload).

6. Optimize for cost

At the end of the day, it often boils down to cost. Your cloud architecture should be designed for cost optimization by keeping in mind the following principles:

You can reduce cost by selecting the right types, configurations and storage solutions to suit your needs. Implementing Auto Scaling so that you can scale horizontally when required or scale down when necessary can be done without any extra cost.List item #1

You can reduce cost by selecting the right types, configurations and storage solutions to suit your needs. Implementing Auto Scaling so that you can scale horizontally when required or scale down when necessary can be done without any extra cost.

7. Caching

Applying data caching to multiple layers of an IT architecture can improve application performance and cost efficiency of application.

There are two types of caching: -

Application data caching: Information can be stored and retrieved from fast, managed, in-memory caches in the application, which decreases load for the database and increases latency for end users.

Edge caching: Content is served by infrastructure that is closer to the viewers lowering latency and giving you the high, sustained data transfer rates needed to deliver large popular objects to end users at scale.

Amazon CloudFront, the content delivery network consisting of multiple edge locations around the world is the edge caching service whereas Amazon ElastiCache makes it easy to deploy, operate and scale in-memory cache in the cloud.

8. Security

Security is everything! Most of the security tools and techniques used in the traditional IT infrastructure can be used in the cloud as well. AWS is a platform that allows you to formalize the design of security controls in the platform itself. It simplifies system use for administrators and those running IT and makes your environment much easier to audit in a continuous manner.

Some ways to improve security in AWS are:

Utilize AWS features for Defense in depth – Starting at the network level, you can build a VPC topology that isolates parts of the infrastructure through the use of subnets, security groups, and routing controls.

AWS operates under a shared security responsibility model, where AWS is responsible for the security of the underlying cloud infrastructure and you are responsible for securing the workloads you deploy in AWS.

Reduce privileged access to the programmable resources and servers to avoid breach of security. The overuse of guest operating systems and service accounts can breach security.

Create an AWS CloudFormation script that captures your security policy and reliably deploys it, allowing you to perform security testing as part of your release cycle, and automatically discover application gaps and drift from your security policy.

Testing and auditing your environment is key to moving fast while staying safe. On AWS, it is possible to implement continuous monitoring and automation of controls to minimize exposure to security risks.

Services like AWS Con□g, Amazon Inspector, and AWS Trusted Advisor continually monitor for compliance or vulnerabilities giving you a clear overview of which IT resources are in compliance, and which are not.

9. Think Adaptive and Elastic

The architecture of the cloud should be such that it supports growth of users, traffic, or data size with no drop-in performance. It should also allow for linear scalability when and where an additional resource is added. The system needs to be able to adapt and proportionally serve additional load.

Whether the architecture includes vertical scaling, horizontal scaling or both; it is up to the designer, depending on the type of application or data to be stored. But your design should be equipped to take maximum advantage of the virtually unlimited on-demand capacity of cloud computing.

Consider whether your architecture is being built for a short-term purpose, wherein you can implement vertical scaling. Else, you will need to distribute your workload to multiple resources to build internet-scale applications by scaling horizontally. Either way, your architecture should be flexible enough to adapt to the demands of cloud computing.

Also, knowing when to engage stateless applications, stateful applications, stateless components and distributed processing, makes your cloud very effective in its storage.

10. Treat servers as disposable resources

One of the biggest advantages of cloud computing is that you can treat your servers as disposable resources instead of fixed components. However, resources should always be consistent and tested. One way to enable this is to implement the immutable infrastructure pattern, which enables you to replace the server with one that has the latest configuration instead of updating the old server.

It is important to keep the configuration and coding as an automated and repeatable process, either when deploying resources to new environments or increasing the capacity of the existing system to cope with extra load. Bootstrapping, Golden Images or a Hybrid of the two will help you keep the process automated and repeatable without any human errors.

Bootstrapping can be executed after launching an AWS resource with default configuration. This will let you reuse the same scripts without modifications.

But in comparison, the Golden Image approach results in faster start times and removes dependencies to configuration services or third-party repositories. Certain AWS resource types like Amazon EC2 instances, Amazon RDS DB instances, Amazon Elastic Block Store (Amazon EBS) volumes, etc., can be launched from a golden image.

When suitable, use a combination of the two approaches, where some parts of the configuration get captured in a golden image, while others are configured dynamically through a bootstrapping action. Not to be limited to the individual resource level, you can apply techniques, practices, and tools from software development to make your whole infrastructure reusable, maintainable, extensible, and testable.

Architecture Scenarios

Architecture Scenario 1: Web Application Hosting

Highly available and scalable web hosting can be complex and, expensive, Dense peak periods and wild swings in traffic patterns result in low utilization of expensive hardware. Amazon Web Services provides the reliable scalable, secure and high performance, infrastructure required for web applications while enabling an elastic scale out and scale down infrastructure to match IT costs in real time as customer traffic fluctuates.

Architecture Scenario 2: Disaster Recovery for Local Applications

Disaster recovery is about preparing for and recovering from any event that has a negative impact on your IT systems. A typical approach involves duplicating infrastructure to ensure the availability of spare capacity in the event of a disaster.

Amazon Web Services allows you to scale up your infrastructure on an as-needed basis. For a disaster recovery solution, this results in significant cost savings. The following diagram shows an example of a disaster recovery setup for a local application.

Architecture Scenario 3: File Synchronization Service

Given the straightforward, stateless client-server architecture in which web services are viewed as resources and can be identified by their URLs, development teams are free to create file sharing and syncing applications for their departments, for enterprises, or for consumers directly.

This diagram represents the core architecture of a scalable and cost-effective file sharing and synchronization platform, using Amazon Web Services.

Architecture Scenario 4: Online Games

Online games back-end infrastructures can be challenging to maintain and operate. Peak usage periods, multiple players, and high volumes of write operations are some of the most common problems that operations teams face.

But the most difficult challenge is ensuring flexibility in the scale of that system. A popular game might suddenly receive millions of users in a matter of hours, yet it must continue to provide a satisfactory player experience. Amazon Web Services provides different tools and services that can be used for building online games that scale under high usage traffic patterns.

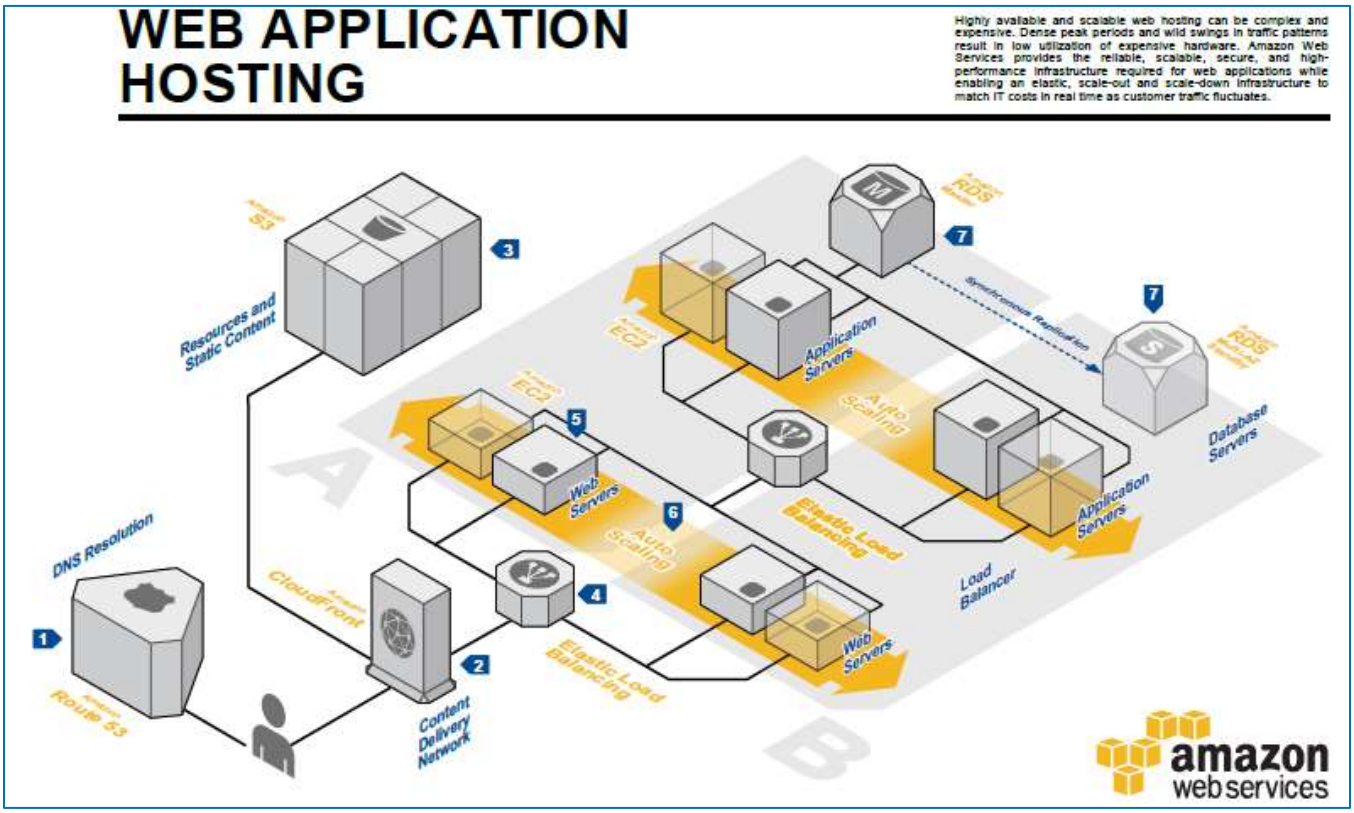
This document presents a cost-effective online game architecture featuring automatic capacity adjustment, a highly available and high-speed database, and a data processing cluster for player behavior analysis.

Architecture Scenario 5: Financial Services Grid Computing

Financial services grid computing on the cloud provides dynamic scalability and elasticity for operation when compute jobs are required and utilizing services for aggregation that simplify the development of grid software.

On demand provisioning of hardware, and template driven deployment, combined with low latency access to existing on-premise data sources make AWS a powerful platform for high performance grid computing systems.

Architecture: Web Application Hosting



System Overview

1 The user's DNS requests are served by **Amazon Route 53**, a highly available Domain Name System (DNS) service. Network traffic is routed to infrastructure running in Amazon Web Services.

2 Static, streaming, and dynamic content is delivered by **Amazon CloudFront**, a global network of edge locations. Requests are automatically routed to the nearest edge location, so content is delivered with the best possible performance.

3 Resources and static content used by the web application are stored on **Amazon Simple Storage Service (S3)**, a highly durable storage infrastructure designed for mission-critical and primary data storage.

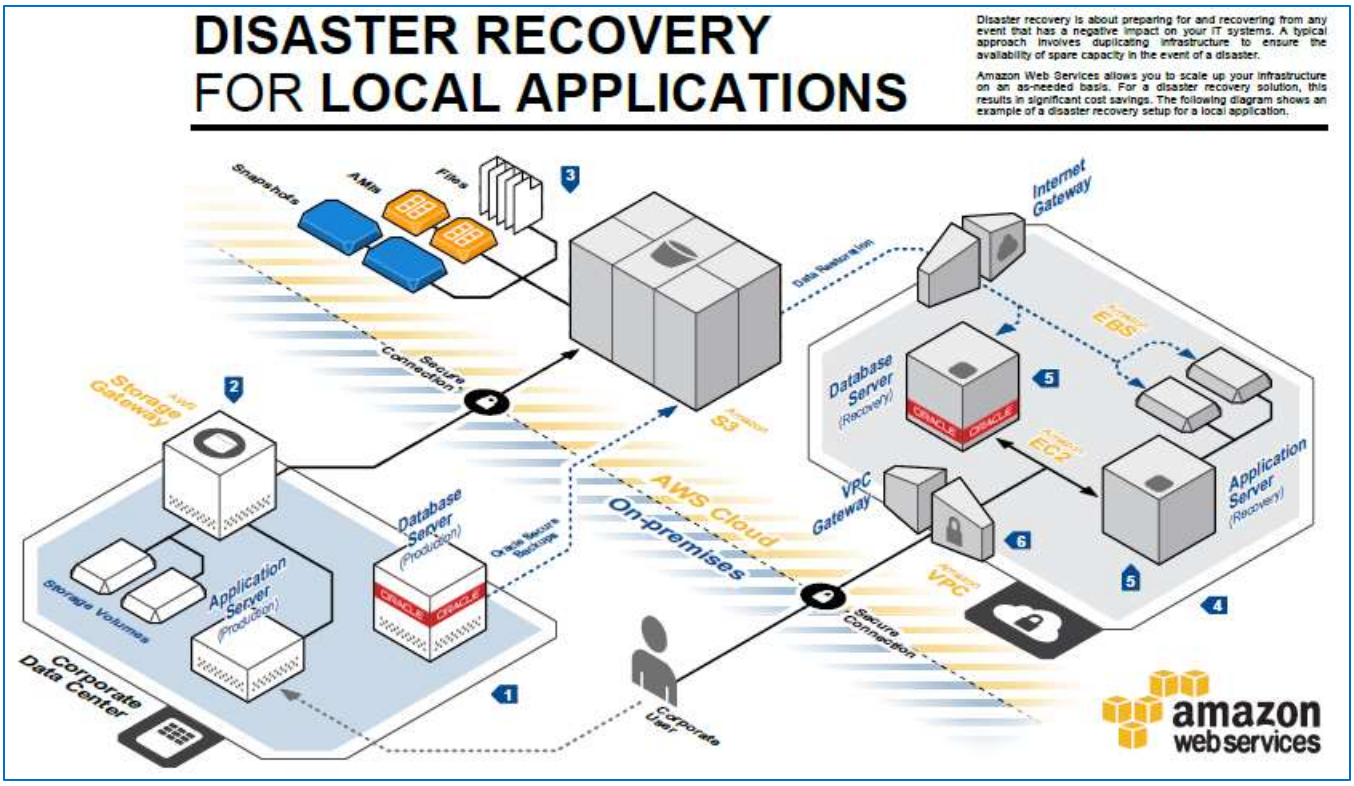
4 HTTP requests are first handled by **Elastic Load Balancing**, which automatically distributes incoming application traffic among multiple **Amazon Elastic Compute Cloud (EC2)** instances across Availability Zones (AZs). It enables even greater fault tolerance in your applications, seamlessly providing the amount of load balancing capacity needed in response to incoming application traffic.

5 Web servers and application servers are deployed on Amazon EC2 instances. Most organizations will select an **Amazon Machine Image (AMI)** and then customize it to their needs. This custom AMI will then become the starting point for future web development.

6 Web servers and application servers are deployed in an **Auto Scaling** group. Auto Scaling automatically adjusts your capacity up or down according to conditions you define. With Auto Scaling, you can ensure that the number of **Amazon EC2** instances you're using increases seamlessly during demand spikes to maintain performance and decreases automatically during demand to minimize costs.

7 To provide high availability, the relational database that contains application's data is hosted redundantly on a multi-AZ (multiple Availability Zones–zones A and B here) deployment of **Amazon Relational Database Service** (Amazon RDS).

Architecture: Disaster Recovery for Local Applications



System Overview

1 A corporate data center hosts an application consisting of a database server and an application server with local storage for a content management system.

2 **AWS Storage Gateway** is a service connecting an on-premises software appliance with cloud-based storage. **AWS Storage Gateway** securely uploads data to the AWS cloud for cost effective backup and rapid disaster recovery.

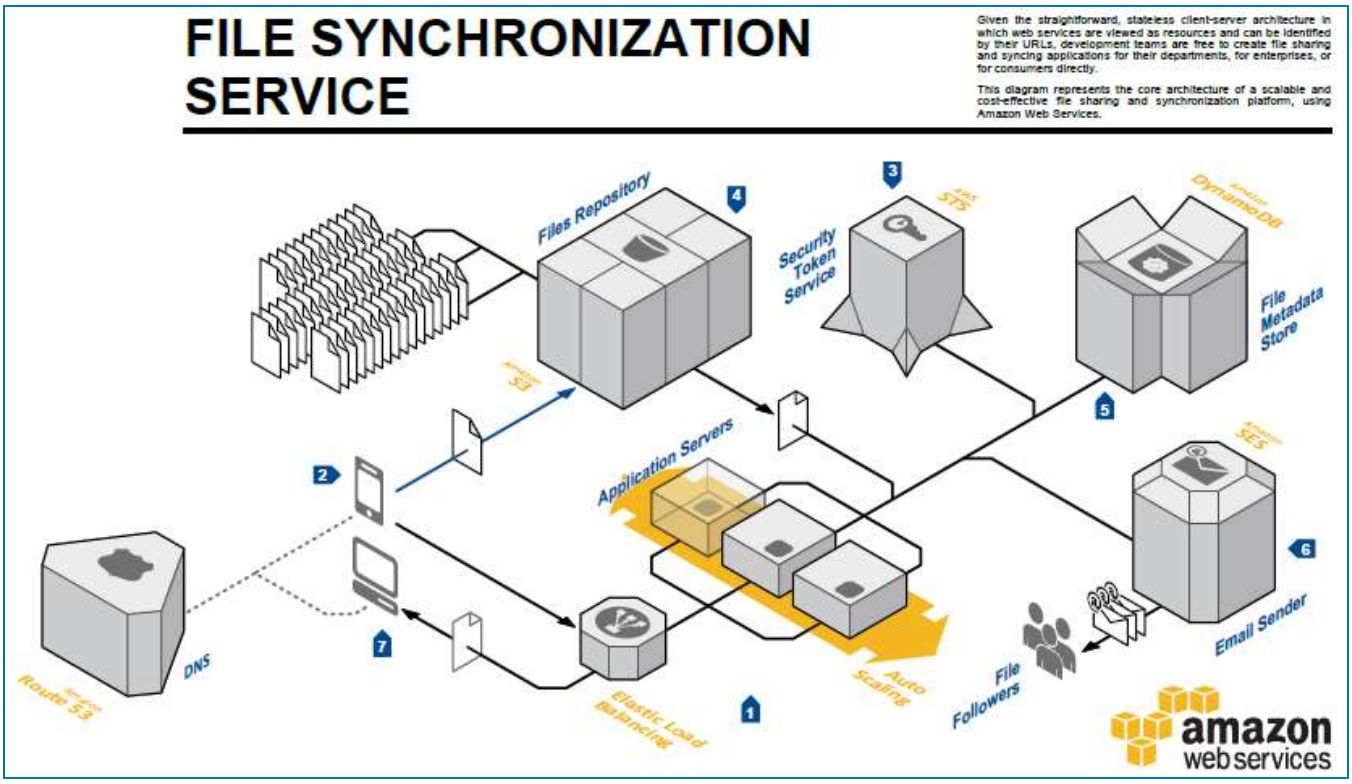
3 Database server backups, application server volume snapshots, and **Amazon Machine Images** (AMI) of the recovery servers are stored on **Amazon Simple Storage Service** (Amazon S3), a highly durable and cost-effective data store. AMIs are pre-configured operating system and application software that are used to create a virtual machine **Amazon Elastic Compute Cloud** (Amazon EC2). Oracle databases can directly back up to Amazon S3 using the **Oracle Secure Backup** (OSB) Cloud Module.

4 In case of disaster in the corporate data center, you can recreate the complete infrastructure from the backups on **Amazon Virtual Private Cloud** (Amazon VPC). **Amazon VPC** lets you provision a private, isolated section of the AWS cloud where you can recreate your application.

5 The application and database servers are recreated using **Amazon EC2**. To restore volume snapshots, you can use **Amazon Elastic Block Store** (EBS) volumes, which are then attached to the recovered application server.

6 To remotely access the recovered application, you use a VPN connection created by using the VPC Gateway.

Architecture: File Synchronization Service



System Overview

1 The file synchronization service endpoint consists of an **Elastic Load Balancer** distributing incoming requests to a group of application servers hosted on **Amazon Elastic Compute Cloud** (Amazon EC2) instances. An **Auto Scaling** group automatically adjusts the number of **Amazon EC2** instances depending on the application needs.

2 To upload a file, a client first needs to request the permission to the service and get a security token.

3 After checking the user's identity, application servers get a temporary credential from **AWS Security Token Service** (STS). This credential allows users to upload files.

4 Users upload files into **Amazon Simple Storage Service** (Amazon S3), a highly durable storage infrastructure designed for mission-critical and primary data storage. **Amazon S3** makes it easy to store and retrieve any amount of data, at any time. Large files can be uploaded by the same client using multiple concurrent threads to maximize bandwidth usage.

5 File metadata, version information, and unique identifiers are stored by the application servers on an **Amazon DynamoDB** table. As the number of files to maintain in the application grows, **Amazon DynamoDB** tables can store and retrieve any amount of data, and serve any level of traffic.

6 File change notifications can be sent via email to users following the resource with **Amazon Simple Email Service** (Amazon SES), an easy-to-use, cost-effective email solution.

7 Other clients sharing the same files will query the service endpoint to check if newer versions are available. This query compares the list of local files checksums with the checksums listed in an **Amazon DynamoDB** table. If the query finds newer files, they can be retrieved from **Amazon S3** and sent to the client application.

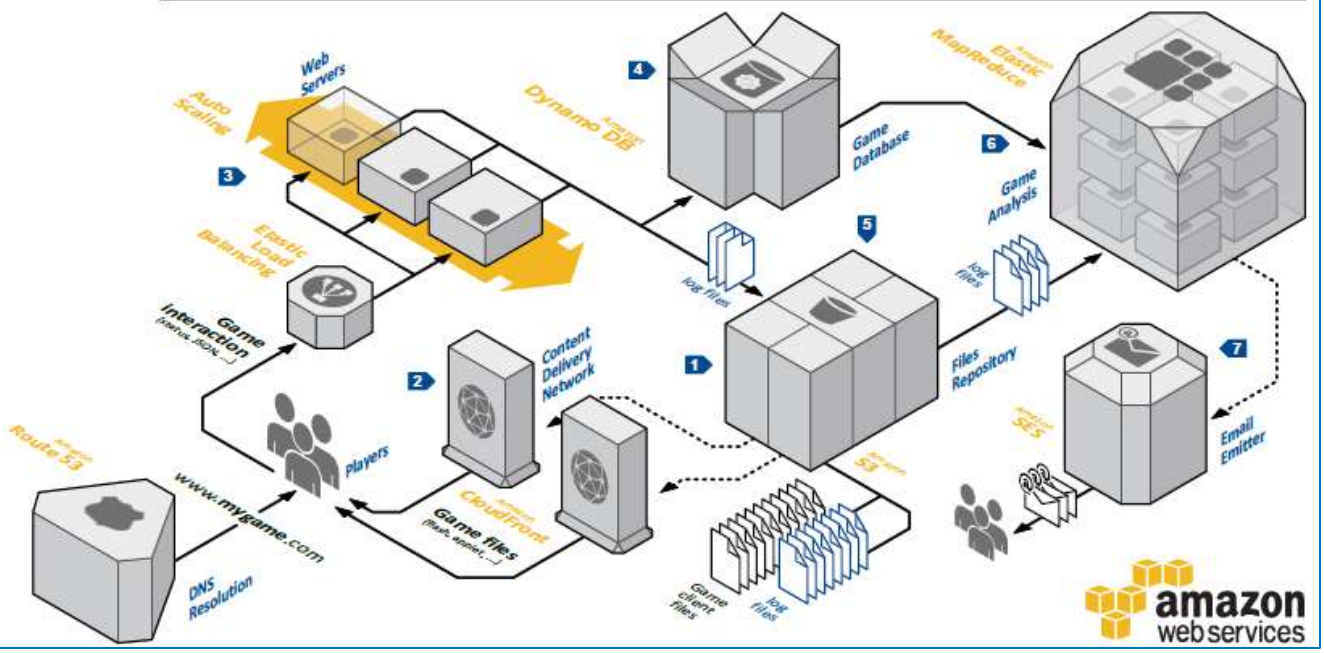
Architecture: Online Games

ONLINE GAMES

Online games back-end infrastructures can be challenging to maintain and operate. Peak usage periods, multiple players, and high volumes of write operations are some of the most common problems that operations teams face.

But the most difficult challenge is ensuring flexibility in the scale of that system. A popular game might suddenly receive millions of users in a matter of hours, yet it must continue to provide a satisfactory player experience. Amazon Web Services provides different tools and services that can be used for building online games that scale under high usage traffic patterns.

This document presents a cost-effective online game architecture featuring automatic capacity adjustment, a highly available and high-speed database, and a data processing cluster for player behavior analysis.



System Overview

1 Browser games can be represented as client-server applications. The client generally consists of static files, such as images, sounds, flash applications, or Java applets. Those files are hosted on **Amazon Simple Storage Service** (Amazon S3), a highly available and reliable data store.

2 As the user base grows and becomes more geographically distributed, a high-performance cache like **Amazon CloudFront** can provide substantial improvements in latency, fault tolerance, and cost. By using **Amazon S3** as the origin server for the **Amazon CloudFront** distribution, the game infrastructure benefits from fast network data transfer rates and a simple publishing/caching workflow.

3 Requests from the game application are distributed by **Elastic Load Balancing** to a group of web servers running on **Amazon Elastic Compute Cloud** (Amazon EC2) instances. **Auto Scaling** automatically adjusts the size of this group, depending on rules like network load, CPU usage, and so on.

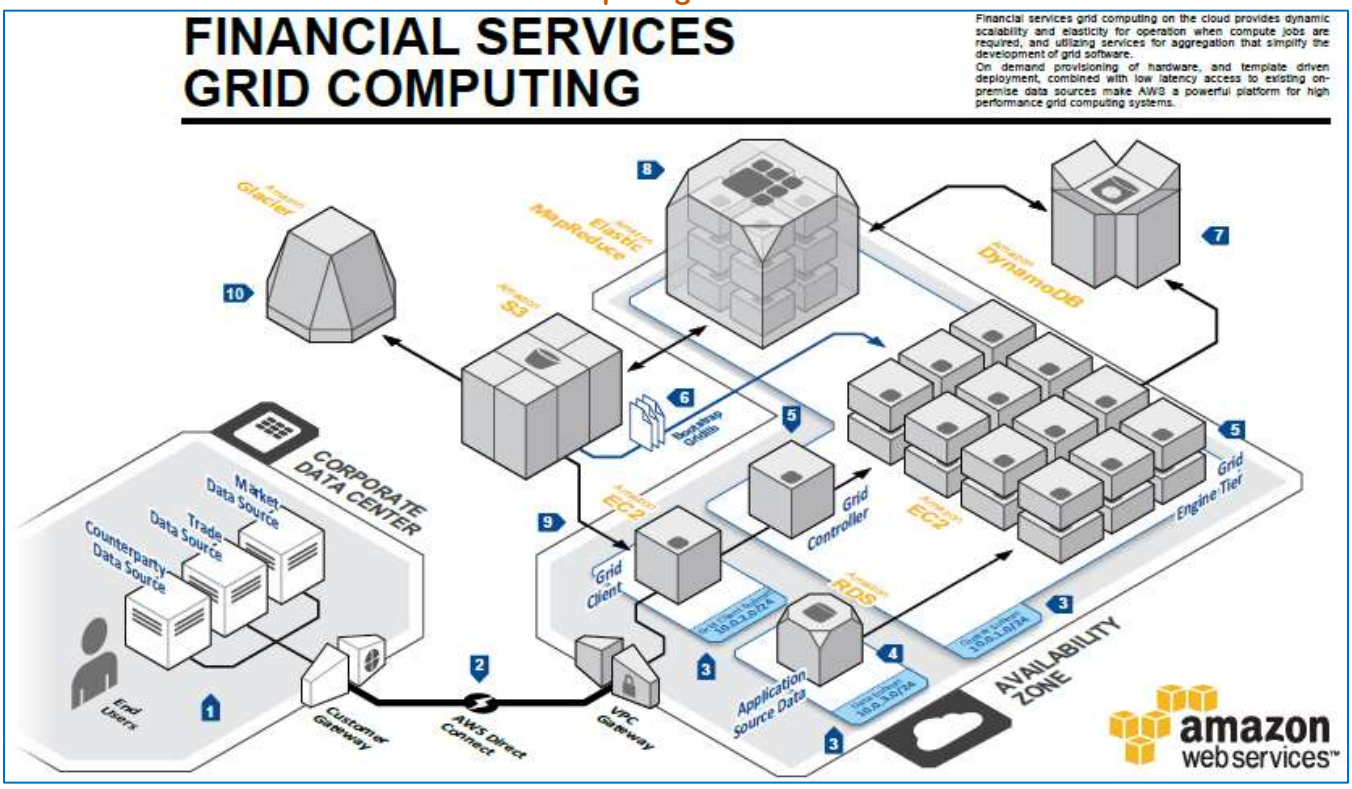
4 Player data is persisted on **Amazon DynamoDB**, a fully managed NoSQL database service. As the player population grows, **Amazon DynamoDB** provides predictable performance with seamless scalability.

5 Log files generated by each web server are pushed back into **Amazon S3** for long-term storage.

6 Managing and analyzing high data volumes produced by online games platforms can be challenging. **Amazon Elastic MapReduce** (Amazon EMR) is a service that processes vast amounts of data easily. Input data can be retrieved from web server logs stored on **Amazon S3** or from player data stored in **Amazon DynamoDB** tables to run analytics on player behavior, usage patterns, etc. Those results can be stored again on **Amazon S3**, or inserted in a relational database for further analysis with classic business intelligence tools.

7 Based on the needs of the game, **Amazon Simple Email Service** (Amazon SES) can be used to send email to players in a cost-effective and scalable way.

Architecture: Financial Services Grid Computing



System Overview

1 Date sources for market, trade, and counterparties are installed on startup from on premise data sources, or from **Amazon Simple Storage Service** (Amazon S3).

2 **AWS DirectConnect** can be used to establish a low latency and reliable connection between the corporate data center site and AWS, in 1 to 10Gbit increments. For situations with lower bandwidth requirements, a VPN connection to the **VPC Gateway** can be established.

3 Private subnetworks are specifically created for customer source data, compute grid clients, and the grid controller and engines.

4 Application and corporate data can be securely stored in the cloud using the **Amazon Relational Database Service** (Amazon RDS).

5 Grid controllers and grid engines are running **Amazon Elastic Compute Cloud** (Amazon EC2) instances started on demand from **Amazon Machine Images** (AMIs) that contain the operating system and grid software.

6 Static data such as holiday calendars and QA libraries and additional gridlib bootstrapping data can be downloaded on startup by grid engines from **Amazon S3**.

7 Grid engine results can be stored in **Amazon DynamoDB**, a fully managed database providing configurable read and write throughput, allowing scalability on demand.

8 Results in **Amazon DynamoDB** are aggregated using a map/reduce job in **Amazon Elastic MapReduce** (Amazon EMR) and final output is stored in **Amazon S3**.

9 The compute grid client collects aggregate results from **Amazon S3**.

10 Aggregate results can be archived using **Amazon Glacier**, a low-cost, secure, and durable storage service.

What key components of Amazon Web Service (AWS) do you use in your project?

We use following key components of AWS in our project:

Amazon Simple Storage Service or (S3): We use AWS S3 to store our data in cloud. Mostly the data is encrypted before storing it. Also, data is replicated in multiple availability zones.

Amazon Elastic Compute Cloud (EC2): We use Amazon EC2 for running our programs in cloud. It gives us scalable computing resources on-demand for hosting applications. We can use Autoscaling to handle high traffic demands.

Elastic Load Balancing (ELB): ELB is used for distributing the traffic in multiple nodes. This is also an important part of scalability solution. Amazon

CloudWatch: We use Amazon CloudWatch to monitor resources in AWS cloud. It helps in not only viewing but also in setting alerts based on key metrics of the AWS components.

Route 53: For DNS management we use Route 53 service of AWS.

Identity and Access Management (IAM): We use IAM for implementing security, identity management and authentication in AWS cloud.

How can your failover gracefully in AWS?

We can use Elastic IPs to implement failover in AWS. Elastic IP is a static IP and it is dynamically re-mappable. In case there is a failure at one node, we can quickly remap and failover to another set of servers. It will lead to routing of traffic to the new servers. It is also useful when we upgrade from old to new versions or when some piece of hardware fails.

What is the use of Availability Zones in AWS?

In AWS, Availability Zone is similar to a logical datacenter. We can deploy our application in multiple availability zones to ensure high availability of the application. Amazon provides RDS Multi-Availability Zone deployment functionality to automatically replicate database updates across multiple Availability Zones. This makes it easier to create and maintain highly available enterprise software systems.

Why AWS systems are built on "Design to Fail" approach?

At the core of an AWS system is, "Design for Fail" principle. It means if we design the software for failure nothing will fail. If we follow a pessimist approach while designing architecture in the cloud, we will assume that things will fail. To handle such failure, we will always create a system that can have automated recovery from failure. An AWS system is designed to automatically recover from design, execution and deploy stage failures.

What are the best practices to build a resilient system in AWS?

We can follow these best practices to build a resilient system in AWS:

Backup: We need a useful and fast, backup and restore strategy for our data. The backup and restore process should be automated. Reboot: Since nodes crash and new nodes restart in AWS, it is good to build threads that automatically resume on reboot of the node.

Re-sync: The system in AWS cloud should be able to re-sync itself by reloading messages from queues. Images: We need to maintain pre-configured and pre-optimized virtual images to restore the system. Also these images should be pre-configured to restart processes on reboot automatically. In-memory sessions: Wherever possible we should minimize the use of in-memory sessions and stateful user context in AWS.

What are the tools in AWS that can be used for creating a system based on "Design to Fail" principle?

AWS provides many tools for creating a strong system based on "Design to Fail" principle. Some of these are:

Elastic IPs: We can failover gracefully by using Elastic IPs in AWS. An Elastic IP is a static IP that is dynamically re-mappable. We can quickly remap and failover to another set of servers so that application traffic is routed to the new set of servers. It is also very useful when we want to upgrade from old to new version of software.

Availability Zones: We can use multiple Availability Zones to introduce resiliency in AWS system. An Availability Zone is like a logical datacenter. By deploying application in multiple availability zones, we can ensure highly availability. Amazon RDS: In AWS, Amazon RDS provides deployment functionality to automatically replicate database updates across multiple Availability Zones. Machine Image: We can maintain an Amazon Machine Image to restore and clone environments easily in a different Availability Zone. We can use multiple Database slaves across Availability Zones and setup hot replication with these Machine images.

Amazon CloudWatch: This is a real-time open source monitoring tool in AWS that provides visibility

visibility on AWS cloud. We can take appropriate actions in case of hardware failure or performance degradation by setting alerts on CloudWatch. Auto scaling: We can maintain an auto-scaling group to maintain a fixed number of servers. In case of failure or performance degradation unhealthy Amazon EC2 instances are replaced by new ones. Amazon EBS: We can set up cron jobs to take incremental snapshots of Database and upload it automatically to Amazon S3. In this way, data is persisted independent of the instances.

Amazon RDS: We can set the retention period for backups by using Amazon RDS. It can also perform automated backups.

How can we build a Scalable system in AWS?

To build a scalable system, we have to follow the principle of Service Oriented Architecture (SOA). The modern word for this is Microservices architecture. Behind a scalable system there are loosely coupled components. Once we build components that are loosely coupled i.e. there is less dependency between them. If one component fails or performs slow, still the other components keep working as if there is no failure. In such a system, it is very easy to build horizontal scaling. We can add multiple servers for components that are heavily used based on the load. We can also add asynchronous communication between components to make the system scalable. This reduces the probability of single point of failure. With loosely coupled components, it is easier to use scalability options present in AWS cloud.

What are the different ways to implement Elasticity in AWS?

Elasticity can be built in AWS in following ways: Periodic Cyclic Scaling: In this case we scale the system at a fixed interval of time like-daily, monthly, quarterly. This is Period based scaling. Proactive Event-based Scaling: When we are expecting a big spike in traffic due to seasonal nature or a special business event (new product launch, holiday weekend), we go for proactive event-based scaling. It is done on one-time basis for a limited time. Auto-scaling: Based on increase in demand that is not known in advance, we can setup a monitoring service. Once demand reaches a threshold, we can scale up the system automatically. It can be based on metrics like- CPU load, memory usage, number of client requests.

What are the benefits of bootstrapping instances in AWS?

Following are the main benefits of bootstrapping instances in AWS: We can recreate the different environments for Dev, QA, and Production etc. with minimal effort by using bootstrapping. Bootstrapping instances gives more control over cloud-based resources in AWS. It also minimizes the occurrence of human related deployment errors.

One main benefit for Bootstrapping is that it can create a Self-Healing and Self-discoverable environment. Such a system is more resilient to hardware failure in Production.

What are the best practices to Automate deployment in AWS?

Some of the best practices to automate deployment in AWS are: Library: We can create a library of scripts that are frequently used for installation and configuration. AMI: We can manage the configuration and deployment process using agents bundled inside an Amazon Machine Image. Bootstrap: We can Bootstrap the instances of components in AWS.

How will you automate your software infrastructure in AWS?

We can use following tactics to automate the software infrastructure in AWS: Auto-scaling: Amazon EC2 can be used for defining Auto-scaling groups for different clusters of servers. It helps in automated handling of traffic spikes and server failure.

CloudWatch: We can monitor vital metrics like- CPU, Memory, Disk I/O, and Network I/O of our servers by using Amazon CloudWatch. It can also help us in taking appropriate actions like launching new servers etc.

Simple DB: We can store and retrieve machine configuration information in machine images in AWS. This helps in automated deployment process. We can store these images in Simple DB in AWS. SimpleDB can also be used to store information about an instance such as its IP address, machine name and role.

Amazon S3: We can create an automated build process that dumps the latest builds into a bucket in Amazon S3. During startup an application read the latest version from Amazon S3 bucket. Failover: While creating AWS architecture, an application component should not assume that it would be up all the time. SO, we can dynamically attach the IP address of a new node to the cluster. Also, we can build automatic failover for servers and start a new clone in case of a hardware failure.

What are the AWS specific techniques for parallelization of software work?

We can use following techniques to parallelize the work in AWS:

Multi-threading: Amazon S3 can handle requests in multi-threading mode. We can create application that can serve concurrent requests from Amazon S3.

DB Requests: Amazon SimpleDB also supports multiple threads. It can be used for concurrent GET requests to get data from SimpleDB. For writing to DB we can use BATCHPUT requests.

MapReduce: Another parallelization technique is to create a JobFlow by using Amazon Elastic MapReduce Service batch processes. It can make the long running tasks finish faster in MapReduce execution mode.

Elastic Load Balancing: Also, we can use Elastic Load Balancing service to distribute the load across multiple web app servers dynamically.

Why it is recommended to keep dynamic data closer to the compute and static data closer to the end user in Cloud computing?

Data proximity is an important principle of Cloud Computing. If we keep the right kind of data at right place, it can help build an excellent enterprise software system. The purpose of keeping dynamic data closer to compute resources is that it can reduce the latency while processing. There is no need for servers to fetch data from remote locations. Even MapReduce algorithm recommends keeping dynamic data nodes closer to compute servers.

Since there is always inherent network latency in a cloud computing environment, this practice can improve the overall performance of computation by saving time from data transfer between servers for processing. Another benefit is that in Cloud we pay for the in and out bandwidth by the GBs of data transfer. So, the cost of data transfer can increase overall costs. In case there is a big chunk of external data that has to be processed in the cloud, we first transfer the data to nodes near the execution environment. And then process the data in parallel mode. It is a common practice in Data warehouse operations to first move the entire database in cloud and then process it in parallel threads. For multi-tier web applications data is stored into and retrieved from relational databases. In such a scenario the recommended architecture is to create app server and db nodes in same cloud environment. Generally there is free data transfer within cloud nodes. Keeping app and db nodes in same cloud can save time as well as money for internal data transfer. For static data like images, pdf, video etc., the recommended approach is to keep it closer to the end user. This kind of data can be cached in nodes that are closer to the user consuming it. This can drastically reduce the access latency for consumer, and provide better user experience.

What are the features in AWS for keeping static data closer to end user?

AWS provides following features to support the static data proximity to end user:
CloudFront: Amazon CloudFront can cache the content in an Amazon S3 bucket for multiple edge locations that are closer to the end user location. Availability Zones: We can use the same Availability Zone to create a cluster of servers. This makes sure that data is in proximity to the processing servers. Physically Ship Data: Yes, we can ship data drives to Amazon by using Import/Export service27. Many a times it is cheaper and faster to move large amounts of data using the sneakernet28 than to upload it over the Internet.

What are the best practices to ensure the security of an application in cloud?

Following are the best practices to ensure the security of a cloud-based application:

Latest Patches: We should regularly download patches from a third-party vendor's web site and update our Amazon Machine Images (AMI).

Amazon Machine Image (AMI): It is advisable to redeploy server instances from the new Amazon Machine Images (AMI) and test the applications for any regression failure. The new patches should not break the existing functionality. Uniform Deployment: We have to ensure that all the instances are deployed with the latest AMI.

Test Automation: Automated test scripts have to be developed to run periodic security checks on applications in cloud.

Third Party: All the third-party software in cloud should be configured with the most secure settings.

Admin User: Whenever possible, the running of any process as a root or Administrator login should be avoided.

Why encryption should be used in Amazon S3?

Amazon S3 is simple storage service. We can create a highly-scalable, reliable, and low-latency data store in Amazon S3. We can use a simple web service interface to store and retrieve data in S3 buckets. These APIs are available at all the time from anywhere in the world. Since these APIs are widely accessible data stored needs security. To keep the data secure, we can encrypt it. Since S3 is Amazon proprietary technology, it is recommended to use our own Encryption strategy on the data stored in Amazon S3.

What are the best practices of Software Security in Cloud?

Some of the best practices of Software Security in cloud are:

Protect data in transit: During transmission of data from one place to another place, we should use secure socket layer (SSL). This is usually done by HTTPS protocol. To do this we need a certificate from a reputed certification authority like VeriSign. Based on the certificate the server can be authenticated by a client browser.

Virtual Private Cloud: We can create virtual private cloud by using Amazon VPC. This can help us in isolating the servers logically within AWS cloud. This can ensure that data transfer is secure within our virtual private cloud.

Protect data at rest: In case we have sensitive information like- Date of Birth, SSN, Passwords etc., we can encrypt this data. So that even if someone gets a copy of the data they cannot decrypt it easily. In Amazon S3, we should always encrypt the sensitive data.

Protect AWS credentials: In AWS there are different types of credentials. We AWS access keys that are used for accessing REST API. Since these keys are sent over web, we should use HTTPS protocol so that these cannot be compromised or tampered during transit.

Embedding Credentials in AMI: Some people make the mistake of embedding AWS credentials in Amazon Machine Image (AMI). We should pass these credentials as an argument during the launch of an AMI.

Key Rotation: We should keep rotating the secret access key on a regular basis. So that even if it is compromised, it cannot be used.

What automation tools can be used to create new servers in AWS?

We can use following ways to create new servers in AWS:

- Puppet: We can use tool like Puppet to write scripts that can create new servers in AWS.
- Custom solution: We can also write our own Perl/bash scripts to spin up new servers in AWS.
- Opscode Chef: We can use third party tools like Opscode Chef for creating new servers in AWS.

14.AWS Migration

What are the pre-requisites for AWS Cloud Migration?

Evaluate your Actual Environment

# of Servers running =

Check for your current Storage:

How much data will you need to migrate?

Review your actual infrastructure and determine service version number. (Ex. php 5.6, Apache 2.4.x, etc)

--

1. Migration Status

ARE YOU ALREADY IN THE CLOUD?

If you already own an account and have something in the cloud, review:

Are you using the Correct type of servers? What type? ----------

Have you migrated your application data?

- **Yes**
- **No**

Using Dedicated IP.

2. Databases

Are you running an optimized database server or on an RDS ?

Correct Database Engine used? (MariaDB, Percona DB, Mongo DB, Mysql)

Is the database separated from the Web Server?

- **Server**
- **RDS**

Is it highly available?

- **Yes**
- **No**

Is there any replication/Failover Solution?

- **Yes**
- **No**

3. Backups & Disaster Recovery

The perfect solution, recover from any disaster or hard situation

- **Automated Snapshots.**
- **Automated AMI's.**
- **Content Being sent to Amazon S3.**
- **RDS Automated Snapshots.**
- **RDS Backup Policy in Days.**

4. Security

- **Assigned just the essential Security Groups to Instances**
- **Assigned just the essential Security Groups to Databases**
- **Is the environment secured in a VPC?**

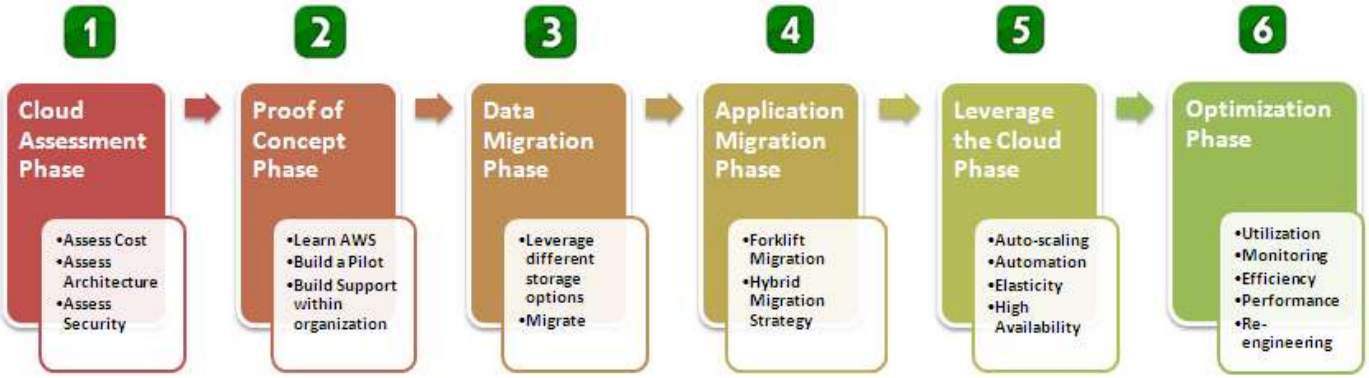
5. Monitoring - CloudWatch Metrics & Custom Metrics

Do not forget to monitor these metrics on your servers, so you can notice any trouble at any

- **CPU Utilization**
- **Memory Available**
- **Disk IO**
- **NetworkIn**
- **NetworkOut**
- **Free Storage Space Available**

What are the steps involved in for AWS Cloud Migration?

Migration Strategy



Phases	Benefits
Cloud Assessment • Financial Assessment (TCO calculation) • Security and Compliance Assessment • Technical Assessment (Classify application types) • Identify the tools that can be reused and the tools that need to be built • Migrate licensed products • Create a plan and measure success	**Business case for migration (Lower TCO, faster time to market, higher flexibility & agility, scalability + elasticity)** **Identify gaps between your current traditional legacy architecture and next -generation cloud architecture**
Proof of Concept • Get your feet wet with AWS • Build a pilot and validate the technology • Test existing software in the cloud	**Build confidence with various AWS services** **Mitigate risk by validating critical pieces of your proposed architecture**
Moving your Data • Understand different storage options in the AWS cloud • Migrate fileservers to Amazon S3 • Migrate commercial RDBMS to EC2 + EBS • Migrate MySQL to Amazon RDS	**Redundancy, Durable Storage, Elastic Scalable Storage** **Automated Management Backup**
Moving your Apps • Forklift migration strategy • Hybrid migration strategy • Build "cloud-aware" layers of code as needed • Create AMIs for each component	**Future-proof scaled-out service-oriented elastic architecture**
Leveraging the Cloud • Leverage other AWS services • Automate elasticity and SDLC • Harden security • Create dashboard to manage AWS resources • Leverage multiple availability zones	**Reduction in CapEx in IT** **Flexibility and agility** **Automation and improved productivity** **Higher Availability (HA)**
Optimization • Optimize usage based on demand • Improve efficiency • Implement advanced monitoring and telemetry • Re-engineer your application • Decompose your relational databases	**Increased utilization and transformational impact in OpEx** **Better visibility through advanced monitoring and telemetry**

15.AWS Lab

AMAZON EC2 - LAB

- Create an AWS account.
- Login to AWS account and navigate to EC2 service.
- Launch on 64bit Linux instance based on Amazon Linux AMI.
- Generate key pair and define security group.
- Access the new instance using Putty or any other SSH client.
- Install PHP and Apache.
- Build new AMI of running instance.
- Transfer AMI in any other region.

AMAZON ELB- LAB

- Navigate to EC2 service.
- Make sure you have two EC2 instance running.
- Install PHP and Apache and create index.html as default page.
- Navigate to Load Balancers under “Network and Security”.
- Create a new load balancer and add both instances.
- Once active, it will generate a new ELB URL.
- Try to access the URL. You should be able to see output of index.html
- Now shutdown one instance and try to access the same URL again.
- You should be able to access index.html again.

AMAZON AUTOSCALING - LAB

- Download Autoscaling and CloudWatch tools.
- Setup the tools by exporting environment variables.
- For autoscaling:
    - Create launch configuration
    - Create auto scaling group
    - Define the scale up and scale down policies.
- Using CloudWatch tools:
    - Create an alarm to call scale up policy
    - Create an alarm to call scale down policy

AMAZON STORAGE - LAB

- Navigate to S3 service.
- Create an S3 bucket.
- Upload data to S3 bucket using AWS console.
- Install s3cmd utility on EC2 instance.
- Upload data to S3 bucket using s3cmd.
- Enable static website monitoring option.
- Try to access bucket content using S3 URL.
- Edit bucket life cycle so that data older than 60 days gets archived to Amazon Glacier.

AMAZON RDS- LAB

- Navigate to RDS Service.
- Create a MySQL Database.
- Specify instance size and credentials.
- Note down the DB end point.
- Try to access database using any MySQL client.

AMAZON CLOUDFRONT- LAB

- Navigate to CloudFront service.
- Create a new distribution of type 'download'.
- Select Amazon S3 bucket as an origin.
- Keep all values default.
- Create the distribution.
- Note down the dynamically generated CloudFront URL.
- Try to access S3 bucket objects using CloudFront URL.

AMAZON CLOUDWATCH- LAB

- Navigate to CloudWatch service.
- Create a new alarm with following parameters.
    - Select statistics for 5 minutes average.
    - Select CPU Utilization metric for an EC2 instance.
    - If alarm triggers, an email notification should be sent out.
- Try to generate some load on server so that alarm triggers.
- Check the status of alarm. If it turns RED, then you should get an email notification.

AMAZON IAM- LAB

- Navigate to IAM Service.
- Create a new group called 'developers'.
- Create a new user 'developer1' in the group of developers.
- Assign Read Only policy to 'developer1'
- Note down the IAM URL to login via console.
- Login with user 'developer1' .
- Try to create S3 bucket.
- You should get an error while trying to create bucket.

AMAZON VPC - LAB

- Navigate to VPC Service.
- Create a VPC with public subnet option only.
- Launch a new EC2 instance in VPC.
- Review following parameters:
    - Internet gateway
    - ACL
    - Routing Table

AMAZON ROUTE53- LAB

- Navigate to Route53 Service.
- Create a hosted zone file for your domain.
- Create one A record and for domain 'www1.yourdomain.com' and point it to the IP address of EC2 instance.
- Configure ELB in US East and Singapore region.
- Create a Failover record for both ELBs created in above step.

AMAZON SES- LAB

- Navigate to SES service.
- Create SMTP credentials.
- Verify sender email address.
- Try to send out a test email using SMTP details provided by Amazon

AMAZON SNS - LAB

- **Navigate to SNS service.**
- **Create a new topic called 'mytopic'.**
- **Create a new subscriber which will use Email protocol.**
- **Confirm the subscription.**
- **Try to send a test message.**

16.AWS Sample Resume

Technical Manager – Cloud

Operations

- Direct daily operations of department, analyzing workflow, establishing priorities, developing standards and setting deadlines
- Facilitate the evaluation and selection of software product standards, as well as the design of standard software configurations
- Consult with application or infrastructure development projects to fit systems or infrastructure to architecture, and identify when it is necessary to modify the solution architecture to accommodate project needs
- Consult with users, management, vendors, and technicians to assess computing needs and system requirements
- Work with proposals to assess project feasibility and requirements
- Control operational budget and expenditures
- Work closely with the program management office (PMO) to ensure alignment of plans with what is being delivered.

Project Team

- Work with department heads, managers, supervisors, vendors, and team members, to solicit cooperation and resolve problems
- Stay abreast of advances in technology
- Understand the entire business process and gather business requirement from the business users for AWS, IoT implementation
- Analyze the requirement | change requests received and set up meeting with the customers to discuss implementation details
- Provide the impact analysis and design documents for business requirements to business users
- Estimate the efforts and develop plan for each phase for customer approvals
- Review the approved project plans to plan and coordinate project activity
- Review Architecture Assessment at project initiation time to ensure proper alignment
- Coordinate solution architecture implementation and modification activities time to time
- Review all solution architecture design and analysis work from various business users
- Assign and review the work of systems analysts, programmers, and architects and guide them to improve the quality of work
- Review and approve all systems charts and programs prior to their successful implementation
- Prepare and review operational reports or project progress reports

- Manage backup, security and user help systems
- Purchase necessary IoT Devices for On Premises to communicate with cloud
- Provide users with technical support for computer problems
- Problem Management – resolve recurring incidents, perform break fixes and implement preventive action items
- Incident Management – log, prioritize and resolve incidents and track them against various SLAs
- Provide 24*7 technical support for production related issues and get resolved as per SLAs
- Provide value add to customers by tuning applications to reduce the run time and improve the performance of the applications and create necessary user supporting documents which allow faster meads to resolve any production issue

Cloud Architect: Sample Resume 1

Project Title: AWSIoTConnect

Environment: AWS (EC2, VPC, ELB, S3, EBS, RDS, Route53, ELB, Cloud Watch, CloudFormation, AWS Auto Scaling, Lambda, Elastic Bean Stalk), **IOT**, MySQL, SQL, AWS CLI, Unix/Linux, Shell scripting, Jenkins, Chef, Tomcat.

Description: The goal of AWSIoTConnect Project is to create a distributed IOT Platform for the Digital IOT World for different IOT Verticals. With AWSIoTConnect we can derive the Value provided by IOT Architecture, once we have the important information extracted we can create an IOT Data Brokerage Model to sell the important data to a Third-Party analytics Vendor or a Public Cloud Provider who provides IaaS. Providing IoT support to the different market segment such as Manufacturing -including infrastructure, awareness & safety, Energy/Utility including Oil & Gas, Transportation - including transportation systems, vehicles, and Non-vehicular, Smart City Applications, Retail IOT, HealthCare IoT, Finance-UBI & BFSI- Blockchain based IoT.

Responsibilities:

- Configured Windows & Linux environments in both public and private domains.
- Integrated Amazon Cloud Watch with Amazon EC2 instances for monitoring the log files and track metrics.
- Proficient in AWS services like VPC, EC2, S3, ELB, Auto Scaling Groups(ASG), EBS, RDS, IAM, CloudFormation, Route 53, CloudWatch, CloudFront, CloudTrail, Snowball, SES.
- Used security groups, network ACL's, internet gateways and route tables to ensure a secure zone for organization in AWS public cloud.

- Created S3 buckets in the AWS environment to store files, sometimes which are required to serve static content for a web application.
- Used IAM for creating roles, users, groups and also implemented MFA to provide additional security to AWS account and its resources.
- Written cloud formation templates in json to create custom VPC, subnets, NAT to ensure successful deployment of web applications.
- Maintained the monitoring and alerting of production and corporate servers using Cloud Watch service.
- Configured AWS Identity Access Management (IAM) Group and users for improved login authentication.
- Created AWS S3 buckets, performed folder management in each bucket, managed cloud trail logs and objects within each bucket.
- Created EBS volumes for storing application files for use with EC2 instances whenever they are mounted to them.
- Experienced in creating RDS instances to serve data through servers for responding to requests.
- Created snapshots to take backups of the volumes and also images to store launch configurations of the EC2 instances.
- Managed automated backups and created own backup snapshots when needed.
- Work with IOT and streaming protocols such as MQTT, LWM2M, SQS, AMQP, Kafka
- Develop AWS IoT Web based Application and Web Services that enables devices to connect to AWS services and other devices
- Work with Device Gateway/Device Registry to enable secure device connections and streaming of data
- Work with Message Broker / Rules Engine to filter, transform and act upon device data with business rules
- Deliver messages to other AWS services

Cloud Architect: Sample Resume 2

Project Title: AWS Support & Maintenance

Environment: AWS (EC2, VPC, ELB, S3, EBS, RDS, Route53, ELB, Cloud Watch, CloudFormation, AWS Auto Scaling, Lambda, Elastic Bean Stalk), **IOT**, MySQL, SQL, AWS CLI, Unix/Linux, Shell scripting, Jenkins, Chef, Tomcat.

Responsibilities:

- Responsible for architecting, designing, implementing and supporting of cloud-based infrastructure and its solutions.
- Created Highly Available Environments using Auto-Scaling, Load Balancers to spin up/down the servers and was responsible to send notifications through SNS for every activity occurred in the cloud environment and automated all configurations using Ansible.
- Developed Cloud Formation scripts to build on demand EC2 instance formation.
- Possess good knowledge in creating and launching EC2 instances using AMI's of Linux, Ubuntu, RHEL, and Windows and wrote shell scripts to bootstrap instance.
- Implemented Amazon RDS multi-AZ for automatic failover and high availability at the database tier.
- Configured and scheduled the scripts to automate the module installation in the environment.
- Created AWS S3 buckets, performed folder management in each bucket, managed cloud trail logs and objects within each bucket.
- Configured S3 to host Static Web content.
- Managing Amazon Web Services (AWS) infrastructure with automation and orchestration tools such as Chef, Ansible.
- Experienced in creating multiple VPC's and public, private subnets as per requirement and distributed them as groups into various availability zones of the VPC.
- Created NAT gateways and instances to allow communication from the private instances to the internet through bastion hosts.
- Created and configured elastic load balancers and auto scaling groups to distribute the traffic and to have a cost efficient, fault tolerant and highly available environment.
- Implemented domain name service (DNS) through Route 53 to have highly available and scalable applications.
- Written Templates for AWS infrastructure as a code using Ansible to build staging and production environments.
- Maintained edge location to cache data with CDN using Cloud Front to deliver data with less latency. Scaled distributed in-memory cache environment in the cloud using Elastic cache.

Cloud Architect: Sample Resume 3

Project Title: AWS Support & Maintenance

Environment: AWS (EC2, VPC, ELB, S3, EBS, RDS, Route53, ELB, Cloud Watch, CloudFormation, AWS Auto Scaling, Lambda, Elastic Bean Stalk), IOT, MySQL, SQL, AWS CLI, Unix/Linux, Shell scripting, Jenkins, Chef, Tomcat.

Responsibilities:

- Designing and deploying scalable, highly available, and fault tolerant systems on AWS
- Selecting the appropriate AWS service based on data, compute, database, or security requirements
- Lift and shift of an existing on-premises application to AWS Ingress and egress of data to and from AWS
- Identifying appropriate use of AWS architectural best practices
- Estimating AWS costs and identifying cost control mechanisms.
- Selecting appropriate AWS services to design and deploy an application based on given requirements
- Migrating complex, multi-tier applications on AWS
- Designing and deploying enterprise-wide scalable operations on AWS
- Implementing cost control strategies
- Picking the right AWS services for the application.
- Leveraging AWS SDKs to interact with AWS services from the application.
- Optimizing performance of AWS services used by the application.
- Code-level application security (IAM roles, credentials, encryption, etc.)
- Actively monitor, research and analyze ways in which the services in AWS can be improved.
- Manage and configure AWS services as per the business needs
- Creating and managing AMI and snapshots
- Upgrade and downgrade of AWS resources (CPU, Memory, EBS)
- Creating AWS instances
- Monitoring servers thorough Amazon Cloud Watch, SNS
- Creating and managing the S3 buckets
- Configuring IAM roles and security

About Sriram

Having 18 years of successful experience in architectural design, development, delivery and manage complex projects with high-performance in real time solutions. I have Worked in US, UK & India, expert in managing continuous business operations and support.

Four pillars of operations make success in the customer engagement such as business operations, team management, project management, customer management.

Business Operations

- **Handled both technology & business teams get more focussed on business objectives via agile methodologies with ROI over $20 billion on each account**
- **Work closely with client partner to generate more business from the client**
- **Work with sales team to enhance the business and provide the incredible support to the customer**
- **100% success in deals winning across multiple vendors**
- **Successful Demand creation and placed the resources on niche skills**
- **Generated more billability to team to enhance the business of an organization**
- **Build a customer rapport geographically to support their needs**
- **Generated a good profit margin for my organization**

Team Management

- **Delivery on time much less overhead**
- **Team collaboration & guidance to grow as per the client expectations**
- **Ensure adherence to defined development life cycle, good software design practices, and architecture strategy and intent.**
- **Prompt remediation of any issues & blockers for the team**
- **Being a certified professional & technical expertise by providing solutions to multiple technologies such as Java, Bigdata, Amazon Web Service, Microsoft Azure, SharePoint,.Net & DevOps**
- **Developed full stack developer to handle the project effectively and efficiently to handle the complex situations**
- **Being an enterprise coach i taken care of team, meeting monthly to shape them in agile career**
- **Produced zero defect bug free product**

Project Management

- Responsible for the performance and success of azure projects to include planning and managing scope, schedule, cost, quality, risk, resources, procurement, communication and tracking of project results

- Work with project charter | plan, roles, tasks, milestones, budgets and measures of success.
- Ensure client requirements are captured accurately and completely. Create and maintain project documentation. Facilitate day-to-day coordination while adhering to standards and sponsor expectations in cloud.

- Monitor projects on an ongoing basis, evaluate progress, quality and manage issue resolution.

- Monitor financial delivery and issue management processes and escalate issues.

- Develop project risk management plans to ensure timely delivery, testing and commissioning of all projects with no impact to business continuity.

- Serve as primary contact to project team members, customers, senior department managers and key stakeholders for status updates and critical change initiatives.

Customer Management

- Business improvement across the geographical location & Cloud promotion for the existing and new projects
- Product Demonstration and customization delivery to the client
- Cost Reduction in maintaining the essential services and efficient storage
- Solution Architecture for cloud-based project
- Cloud Assessment & Migration
- Customer Satisfaction Index above 95% consistently

Achievements

- Generated revenue from 10 to 20 billion for the past 2 years for my company
- Cost reduction benefit to the client over 30% in effectively managing cloud platform
- Build the high-performance team in an agile fashion