Cloud Computing

Computing Services And More Over The Internet

Agenda

Cloud Computing

- What? Why?

AWS

- What? Why?
- AWS Management Console
- AWS Services
  - IAM
  - EC2 (AMI, EBS, SG)
  - S3
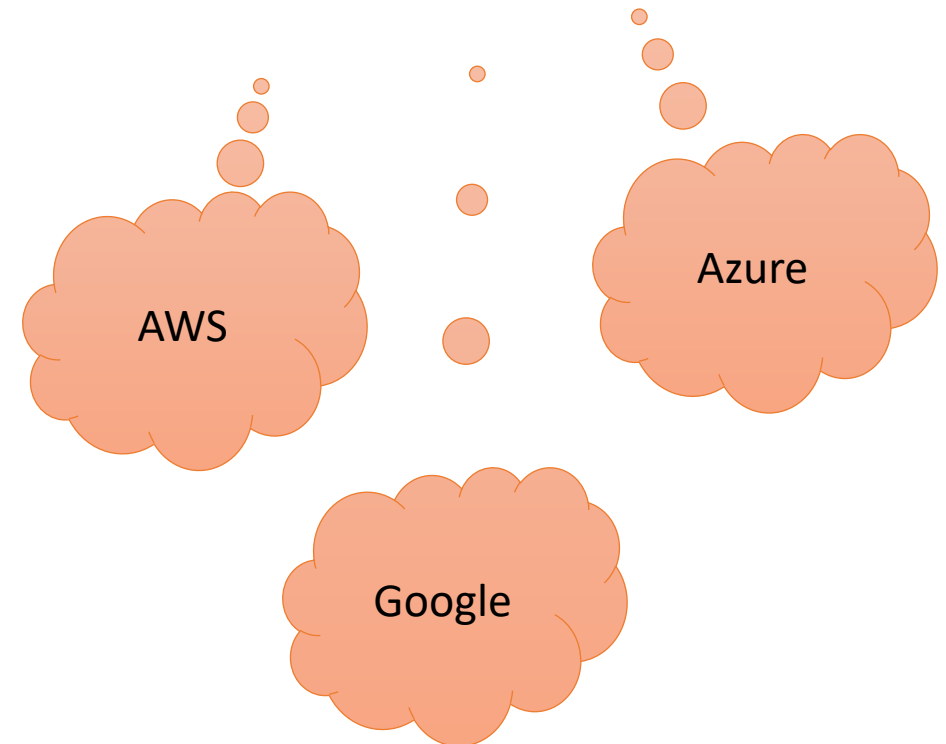 - Database Service (RDS & Dynamo)
  - Route 53
  - Lambda

Cloud Computing

Cloud Computing is defined as storing and accessing of data and computing services over the Internet/Cloud.

Examples

on-demand availability of computer services like servers, data storage, networking, databases, etc.

AWS

Azure

Google

Give access to data centers to many users

Benefits of Cloud Computing

Cheaper Solution	Large and small scale businesses today thrive on their data & they spent a huge amount of money to maintain this data.
Decreases the hardware and software demand	The only thing that user must be able to run is the cloud computing systems interface software, which can be as simple as Web browser, and the Cloud network takes care of the rest.
Unlimited Resources	It gives small businesses access to the technologies that previously were out of their reach.
Responsibility	With in-house IT Server, in case of any technical glitch you are completely responsible; it will seek a lot of attention, time and money for repair.

Types of Cloud

Private	Computing resources are deployed for one particular organization which governed, owned and operated by the same organization.
Public	Such type of cloud is used usually for B2C (Business to Consumer) type interactions. Computing resource is owned, governed and operated by government, an academic or business organization.
Hybrid	Such type of cloud can be used for both type of interactions – B2B (Business to Business) or B2C (Business to Consumer).

Benefits of Cloud Computing

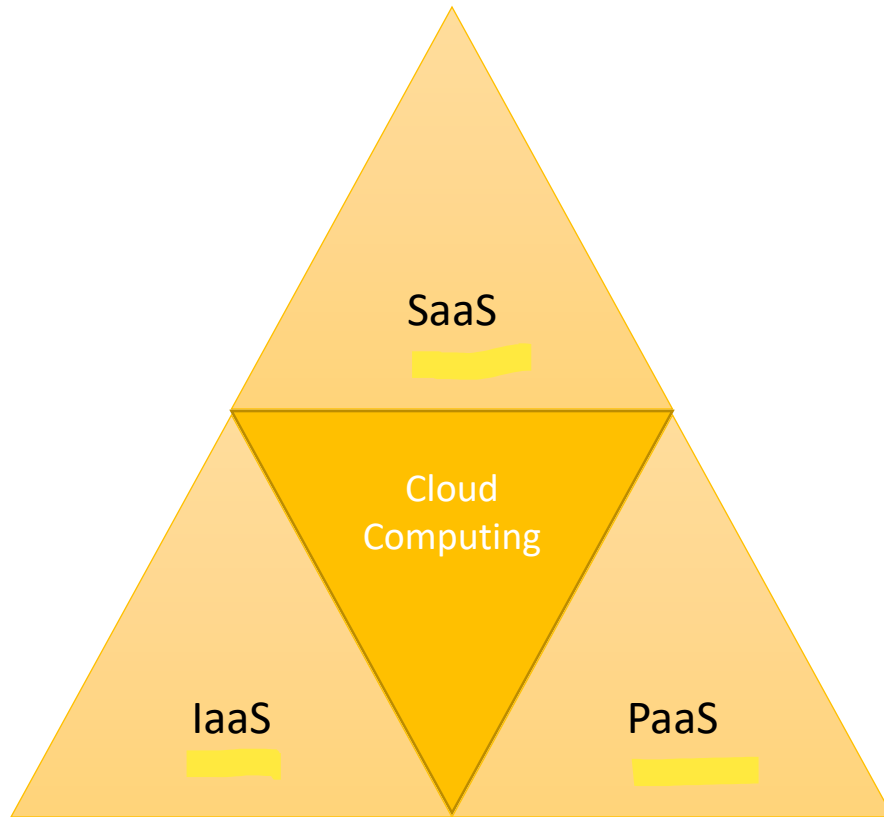|			
Lower IT Infrastructure And Computer Costs	Improved Performance	Fewer Maintenance Issues	Instant Software Updates
Improved Compatibility Between Operating Systems	Backup And Recovery	Performance And Scalability	Increased Storage Capacity And Data Safety

Cloud Computing Services

SaaS or software as a service is a software distribution model in which applications are hosted by a vendor or service provider and made available to customers over a network (internet)

PaaS provides a platform and environment to allow developers to build applications and services.

IaaS provides computing infrastructure like virtual server space, network connections, bandwidth, load balancers and IP addresses.

Utility computing is the process of providing service through an on-demand, pay-per-use billing method.

Grid Computing is a middleware to coordinate disparate IT resources across a network, allowing them to function as a whole.

Cloud Computing Architecture

A Cloud computing architecture consists of several components like a frontend platform, a backend platform or servers, a network or Internet service, and a cloud-based delivery service.

Web Browser

Web Browser

Internet

Database server

Web & Application Server

Front end

Back end
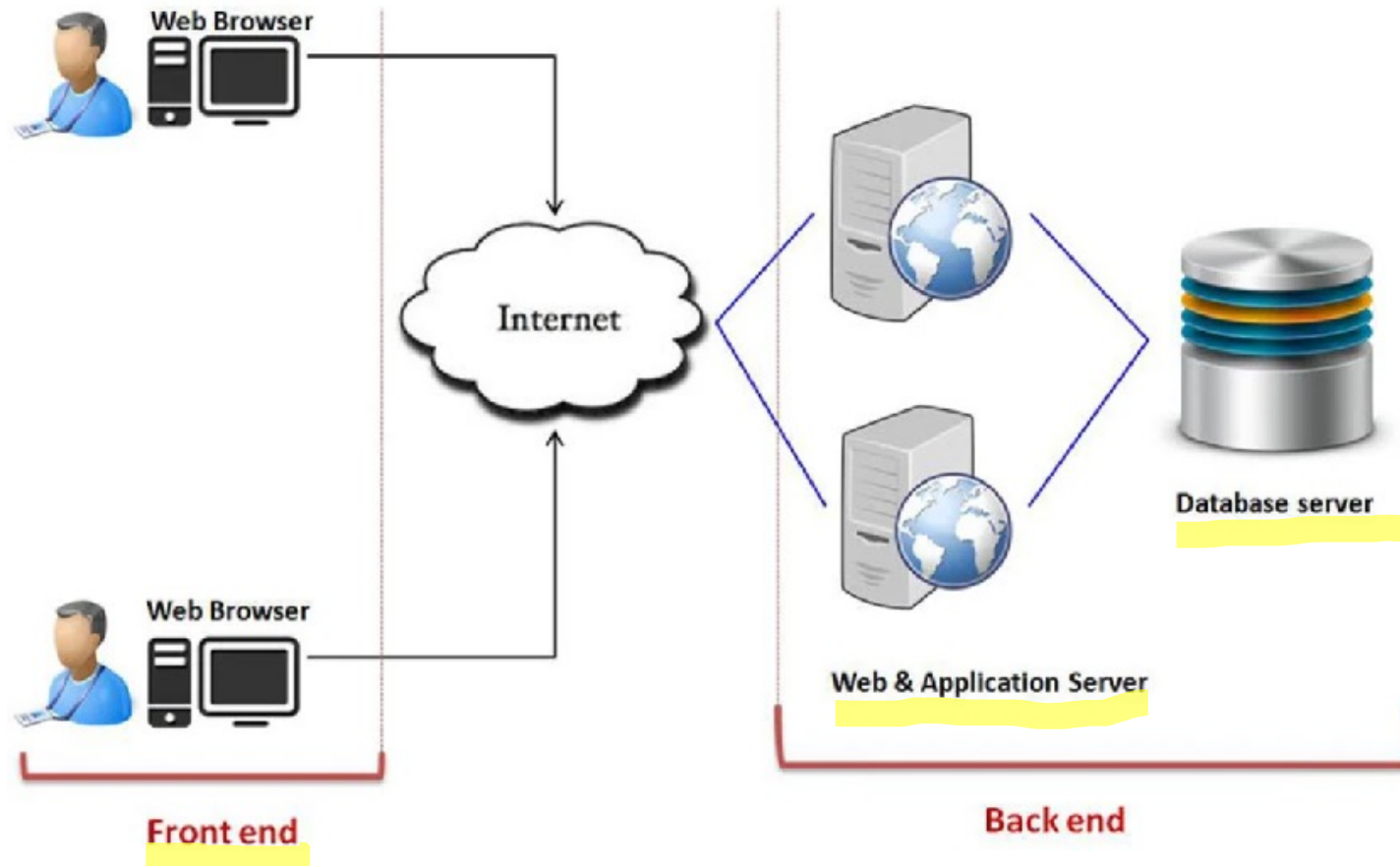
Components of Cloud Computing Architecture
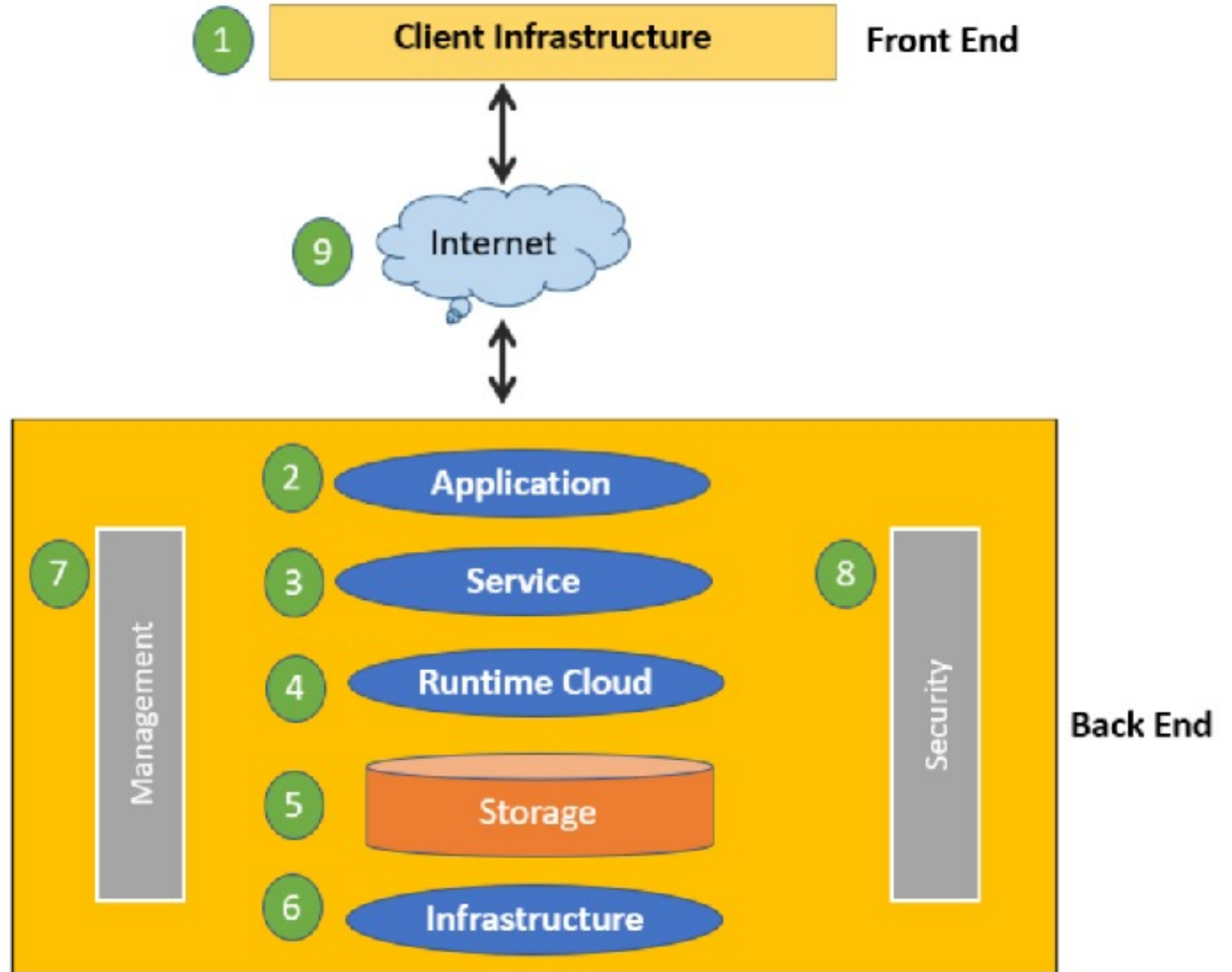
Components of Cloud Computing Architecture

Client Infrastructure	Front-end component that provides a GUI. It helps users to interact with the Cloud
Application	Application can be any software or platform which a client wants to access.
Service	Manages which type of service you can access according to the client's requirements.
Runtime Cloud	Runtime cloud offers the execution and runtime environment to the virtual machines.
Storage	It provides a large amount of storage capacity in the Cloud to store and manage data.

Components of Cloud Computing Architecture

Infrastructure	Cloud infrastructure includes hardware and software components like servers, storage, network devices, virtualization software, and various other storage resources that are needed to support the cloud computing model.
Management	This component manages components like application, service, runtime cloud, storage, infrastructure, and other security matters in the backend.
Security	Security in the backend refers to implementing different security mechanisms for secure Cloud systems, resources, files, and infrastructure to the end-user.
Internet	Internet connection acts as the bridge or medium between frontend and backend.

Characteristics of Cloud Computing
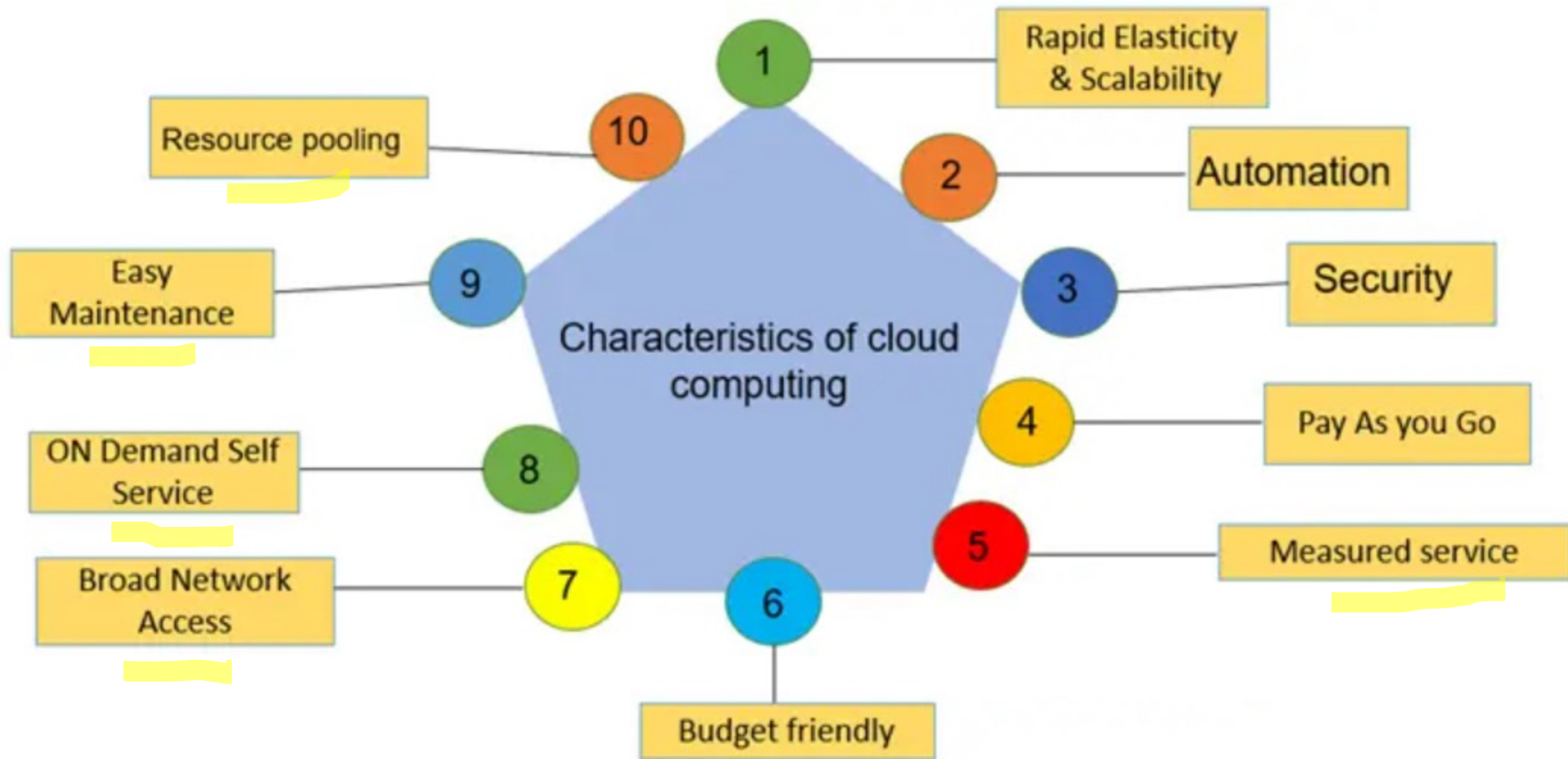
AWS

Entering In Cloud World

AWS : Introduction

IaaS

AWS = Amazon Web Services

AWS is the subsidiary of amazon.com

Pay only for what you use.

Infrastructure as a Service

AWS manages the infrastructure for you

Rapidly scale-up and scale-down

AWS : Introduction

AWS launched it's first Service: Amazon S3 (Simple Storage Service) in 2006 followed by SQS (Simple Queue Service) and EC2 (Elastic Compute Cloud)

As of now, AWS has 90+ Services and it's ever growing list of Services!

Form the basic storage and compute solutions to the most advanced AI/ML or VR/AR Services.

Most Popular Services :

- S3 – Simple Storage Service
- EC2 – Elastic Compute Cloud
- ELB – Elastic Load Balancer
- RDS – Relation Database Service
- CloudFront – A Global CDN

AWS Management Console

Its an interface which assist you in assessing and managing AWS Services

You can access this console via simple & intuitive Web based UI

You can login via AWS username and password (if created already)

Sign-in to AWS Console – https://aws.amazon.com

AWS Documentation

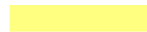With 90+ Services and 100s of features, AWS documentation is an ocean

Practically impossible to know/understand every specific detail of every service

Highly dynamic – keep changing / updated

Publicly accessed via https://aws.amazon.com/documentation

AWS - IAM

Identity and Access Management

IAM – Who? What? Which?

Who?	Users, Groups and Roles
What?	Permissions (e.g ReadOnly)
Which?	Resources : S3, EC2, RDS etc

What is IAM?

IAM enables you to manage users to access AWS Services and resources securely via Users, Group, Roles and Policies

IAM is a global service. You can't restrict it based on region.

IAM is absolutely FREE, but the resources used by the users are chargeable.

Why IAM?

Use fine-grained access control via AWS Console, or integrate it with your corporate directory

Permissions let you specify to AWS Services

Permissions are granted to IAM entities(users, groups, roles) and by default, these entities start with no permissions

To give entities permissions, you can attach a policy that specifies the type of access, the action that can be performed & the resources on which the action can be performed

IAM : Key Terminologies

Users	Individual User. A User can be added to a group or without group
Group	A Collection of IAM Users (with similar set of requirements). Easy to manage permissions with group.
Permissions/ Policies	Used to grant privileges
Roles	An IAM entity that defines a set of permissions for making AWS Service request.
Root User	The email Id that used to sign-up AWS account, it has unrestricted access to AWS account.
MFA	Multi-Factor Authentication. For Account security.

Adding A New User
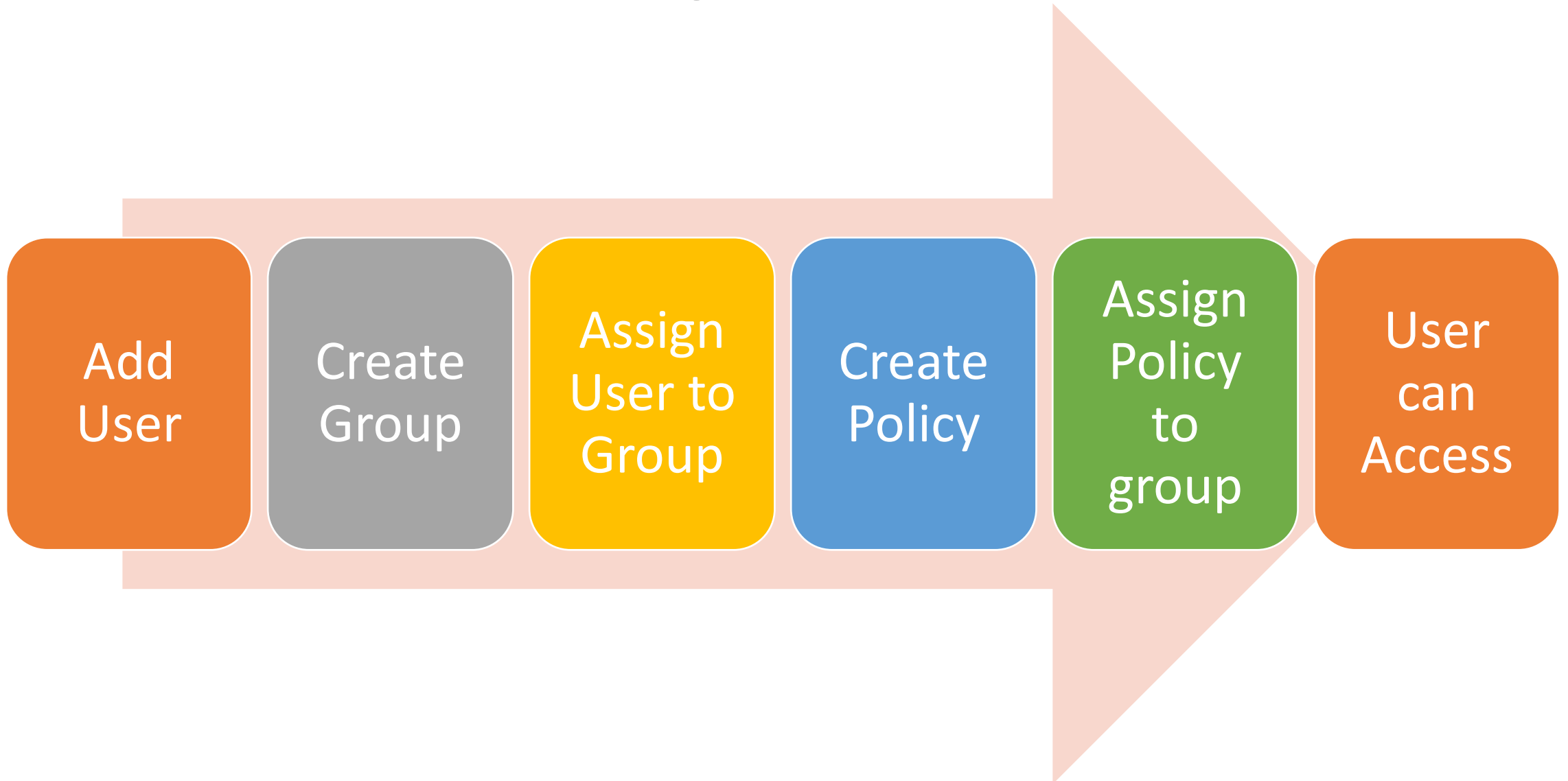
IAM : Best Practices

Reduce or remove root user for day-to-day activities

Don't share user credentials among team members. Create individual user.

Configure strong password policy

Enable MFA for privileged user

Rotate Security credentials regularly

Manage permissions with group

EC2

Elastic Compute Cloud

What is EC2?

EC2 in plain English can be defined as Virtual Server.

Launched in 2006. EC2 is one of the oldest service of AWS.

Allows user to rent VM where they can run their own applications

You can launch EC2 instance from predefined templates (AMI's) offered by Amazon or 3rd party vendor

You can launch ES2 instance for the region of your choice based on the geolocation of your target audience.

Why EC2?

Requirement

- You want to move your application to the server so that it's globally accessible over the Internet

Challenges

- Huge CapEx
- Procurement cycle can go for a week to few months
- Maintenance
- Sizing up or down
- Security
- Performance
- Server Latency

Why EC2?

Server	Pre-Cloud	Cloud
Expenditure	High one time CapEx	Timely monthly OpEx
Procurement Cycle	A few weeks to months	A few minutes
Maintenance	Nightmare	Infrastructure and network managed by AWS
Size up or down	A few weeks to months	A few minutes
Security	Nightmare	State-of-the-Art infrastructure Security
Performance	May be using outdated hardware	New gen, Fast compute hardware
Geographical spread	Mostly restricted to a single location	Global – across many countries

Benefits of EC2

On-demand instant server

Highly reliable (designed for 99.99% availability)

No long term commitments / upfront payments (pay as you go)

Scale up and scale down capacity based on your requirement

Per second billing – EC2 Linux instance are billed in one second increments.
Windows instances are billed on hourly basis (may change)

AMIs – Amazon Machine Images

AMIs are predefined templates or master images which are used to launch(create) your Virtual Servers or EC2 instances.

AMIs are pre-configured with OS, Web-servers, Databases and Programming languages etc. Hence you get server ready in minutes!

AMIs also contain information for volumes to be attached to the instance

Which OS and software would you like to have for your VM?

AMIs

Amazon Linux AMI, Windows Server 2016/2012 are commonly used by beginners as they are free tier eligible

You can select an AMI provided by AWS, the user community or through the market place

You can also create your own AMI and share them with other AWS accounts or with the community.

You can Buy, Share and Sell AMIs.

EC2 Instance Families / Types

How many CPUs you want? How much memory is sufficient to your workload? Is your workload CPU intensive or memory hungry?

Amazon EC2 provides a wide selection of instance optimized to fit for different use cases.

Instance type comprise various combination of CPU, memory, storage or network capacity and give you the flexibility to choose the appropriate mix of resources for your app.

One instance type can have multiple configuration

Instance families

- General purpose – T2, M5, M4
- Compute Optimized – C5, C4
- Memory Optimized – R4, X1, X1e
- Storage Optimized – H1, I3
- Accelerated Computing – P3, G3

What are Security Groups

Security group acts as virtual firewall to your server. It can control traffic going-in or coming-out from your server.

You can add security rules to SG that allow traffic **to** or **from** the instance

At the time of launching EC2 instance you can select the default SG or you can create your own.

You can also configure SG rules to control inbound and outbound traffic

IP & CIDR

IP

- Internet Protocol
- An IP address is an identifier for a computer/device over the network.
- Examples -
  - 192.168.100.101
  - 192.168.100.101
  - 192.168.100.101

CIDR

- Classless Inter-Domain Routing
- Short-cut notation for describing blocks or range of IP addresses.
- Examples -
  - 192.168.100.101/32 represents as 1 IP
  - 192.168.100.101/31 represents as 2 IP
  - 192.168.100.101/30 represents as 4 IP
  - 192.168.100.101/24 represents as 256 IP

0.0.0.0/0 – represents all the IPs

Security Group – Problem Statements

Problem Statement 01	Security of DB Servers are of utmost importance. To make your DB server highly secure , you need to ensure that traffic that originates only from you web server hosted at port 192.168.100.101 should be allowed to reach your DB Server. Rest every request should be **blocked**.
Problem Statement 02	You want your internal web application to open only from within your office network which has a range of 8 Ips viz 192.168.100.80, 192.168.100.81 ……. 192.168.100.87. You need to ensure that the http request originated from your company network should be rendered only.
Problem Statement 03	Your company website must be accessible from anywhere over HTTP and HTTPS. Whst should be the best configuration?

Security Groups – Points to remember

Every EC2 instance must have at least 1 SG. Max 5.

At the time of launch of EC2 instance, it is mandatory step to create your own SG or choose from the default one.

For each SG, you add rules that control the inbound traffic to instance, and a separate set of rules that control the outbound traffic.

SG are stateful. Responses to allowed inbound traffic are automatically allowed to flow out, regardless of outbound rules.

Lab : Launch an EC2 Instance

Expected completion time : approx. 5 minutes

- Form the moment we click launch instance button till we get 2/2 status check

Instance = Virtual Machine = VM = Virtual Server = Server = EC2

- All of the above mean the same, and can be used interchangeably.

You will have the option to choose and configure -

- Operation System (AMI)
- RAM size / CPU (Instance type)
- Storage Space
- Virtual Firewall (Security Group)

Steps to deploy a website on EC2 instance

Connect	Connect to your Windows Server instance
Install IIS	Once connected, install IIS on the instance
Test	Test your installation. Can you access IIS default website from instance web browser?
Add	Add your website to IIS
Develop	Develop a simple HTML Page
Deploy	Deploy the above page inside your website that you created in IIS
Access	Access your website from browser using public DNS

EC2 – Security Precautions

Never ever share your AWS root user password

Don't use root user for day to day operations

Do activate MFA on your root account

Never ever post your AWS user credential on Github / Forums

SG – start with minimum permission rules

AWS documentation is precise and well written. Google any topic by pre-fixing AWS keyword and you'll land there. Example- AWS Security Group.

AWS Free Tier Cost Optimization

AWS FREE tier has many limitations in terms of usage. Example – EC2 has limits on type of instance you can use & hours of usages per month.

The FREE tier of EC2 provides you with 750 hours usage of t2.micro instance.

If you are using any resource NOT covered under FREE tier or beyond the hourly limit, be careful as you have to pay for it.

STOP/TERMINATE the instance/ resources, if you are not actively using them.

Amazon S3

Simple Cloud Storage

Amazon S3 : Simple Storage Service

It's cloud based storage service

Can store images, videos, pdf, log files or any static file

Unlimited storage capacity i.e. almost infinite scalability

Chances of data lost in S3 is almost 0.

Used alone or in combination of other services

Can be accessed from Console, API, CLI.

You can NOT use Amazon S3 for …..

Installing

- Database
- Operating System
- Any other program viz web server, ftp etc

Server side execution

- Running program that require server side processing (PHP, Java, DotNet, Node Code)

Why?

- S3 is an Object storage, NOT a block storage service.

S3 – Buckets and Objects

Logical container to organize your files in Amazon S3. Similar to **Folder.**

Remember

Bucket name is globally unique. This allow your bucket to be used as URL that can be accessed publicly.

The bucket name can be between 3 and 63 character long and can contain lower-case, numbers, periods and dashes.

There is no maximum bucket size limit

Wisely choose AWS region for your bucket to optimize latency and minimize cost.

S3 – Buckets and Objects

S3 stores data as Objects. Objects are stored in Buckets.

Remember

You can store unlimited number of objects in a bucket
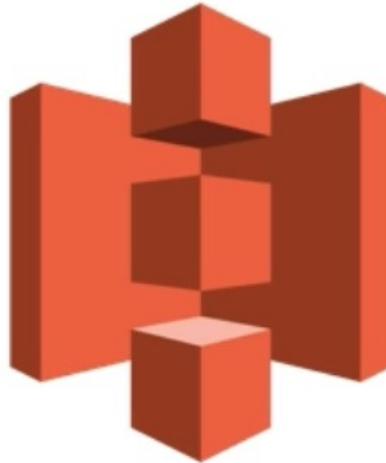
Maximum size of a single object – 5 TB

You can store as many objects as you want in bucket and write, read and delete objects in your bucket.

For objects larger than 100MiB, consider using the Multipart upload capability.

S3 – Image Representation

Bucket

Amazon S3

Bucket with Objects

S3 – Key Features

Designed for 99.999999999 durability (Eleven 9's)

99.99 Availability

Support versioning

Supports object life cycle management

S3 is secured and can use encryption

S3 Data Consistency Model

S3 is an example of distributed system. To achieve high availability it stores multiple copies of data across multiple servers within AWS data centres.

Upload New Object

- If you upload a new object to S3, it is immediately visible.

Update

- If you update an existing object it may take sometime to reflect the modified object.
- Changes must replicate across S3, which can take sometime.

S3 Storage Classes

S3 Standard	S3 Standard - IA	S3 One Zone - IA	Amazon Glacier
This is for data that require higher durability, availability and performance for frequently accessed data.	This is for data which is accessed less frequently, but require rapid access when needed.	Unlike other Amazon storage classes which stores data in minimum of 3 Availability Zones (AZs), S3 One Zone – IA stores data in a single AZ	Amazon Glacier is secure, durable and extremely low-cost storage service for data archiving.

Glacier objects are NOT available for real-time access. You must restore archived object before you can access them. |

S3 Storage Classes Comparison

Parameters for Comparison	S3 Standard	S3 Standard - IA	S3 One Zone - IA	Amazon Glacier
Design for Durability	99.999999999	99.999999999	99.999999999	99.999999999
Design for Availability	99.99	99.9	99.5	NA
Availability SLA	99.9	99	99	NA
Availability Zones	>= 3	>= 3	1	>= 3
Price				
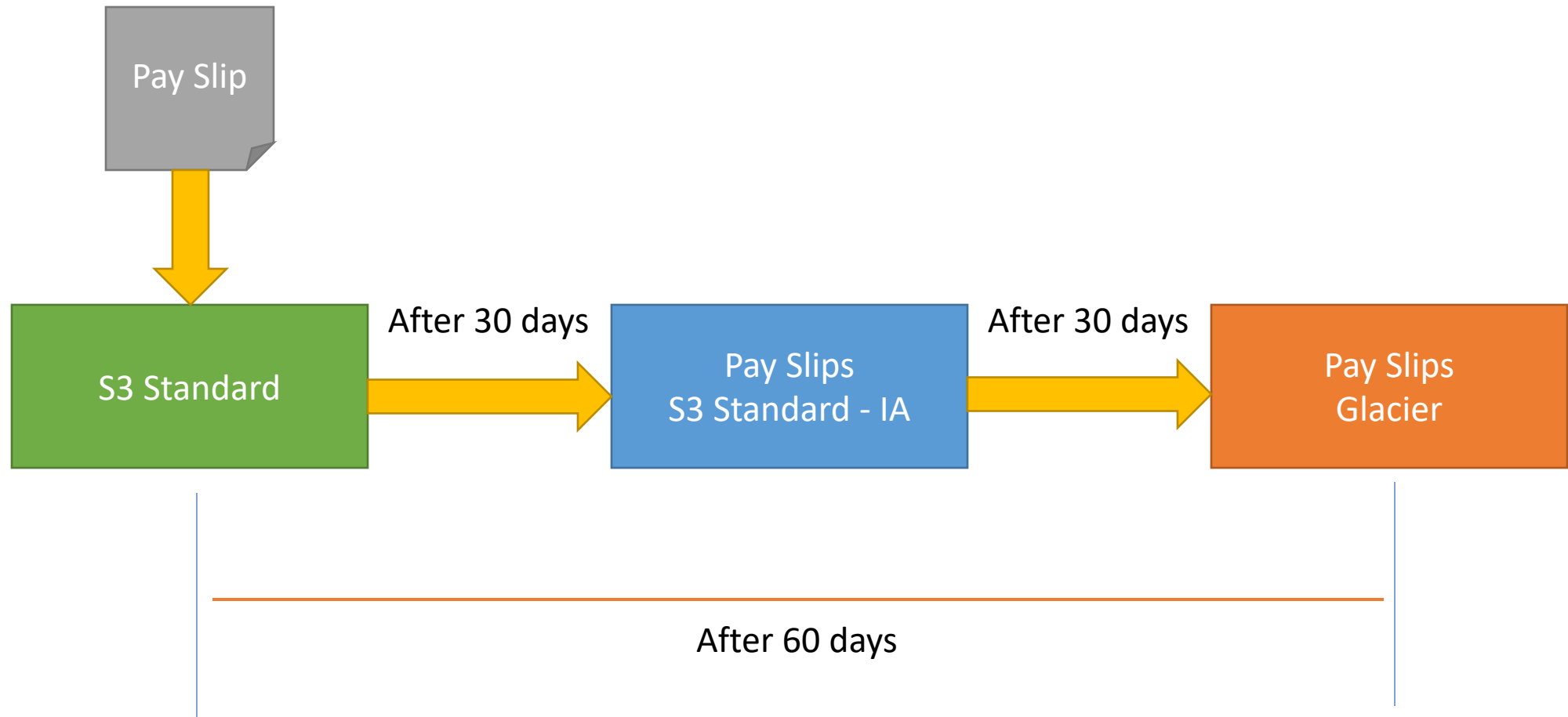
S3 Permissions

Bucket Policies	Attach only to the S3 Bucket
	Permissions defined in the policy is applied to all the object present in Bucket.
	Policy can define **Allow** or **Deny** actions
S3 Access Control List (ACL)	If you want to manage permissions on individual objects within a bucket, S3 ACL enable you to apply policies on the objects themselves, wherein bucket policy can then be applied to the bucket level only.

How Does Authorization Works With Multiple Access Control Mechanism?

What happens when all the three policies- i.e. IAM Policy, S3 Bucket Policy and S3 ACL Policy – all are present at the same time?

In accordance to the principle of least –privileges, decision default to DENY and an explicit DENY will always trumps an ALLOW

For example, if an IAM policy grants access to an object, S3 bucket policy DENY access to the object and there is no S3 ACL, then access will be denied.

Similarly, if no method specify ALLOW, then the request will be denied by default.

Only if no method specifies a DENY and one or more method specifies it ALLOW, will the request be allowed

AWS Database Service

Entering the Database World

AWS Database Service - Introduction

AWS provides following types of Database services -

RDBMS served through
Amazon RDS

NoSQL served through
Amazon Dynamo

In-memory caching

Data-warehouse

What is Amazon RDS?

RDS makes it easy to setup, operate and scale a relational database in the cloud.

It offers resizable capacity while automating time consuming administrative tasks such as database setup, patching and backups.

Amazon RDS is available for several database instance type: optimized for memory, CPU / IO etc.

It is cost efficient.

RDS Provides 6 familiar RDBMS engines:

Oracle	MS SQL Server
MySQL	MariaDB
PostgreSQL	Amazon Aurora

Benefits of RDS

RDS Instance Type

RDS provides a selection of instance types optimized to fit different relational database use cases.

Instance type comprise various combo of CPU, Memory, Storage and Network capacity, giving you more flexibility to choose appropriate mix of resources for your database

Each instance type includes one or more instance sizes, allowing you to scale your resources to the requirement of your target workload.

RDS Instance Class Types -

Standard – m4, m3, m1

Memory optimized – x1, x1e, r4, r3, m2

Burstable Performance – t2

Amazon DynamoDB

Amazon DynamoDB is non-relational database (NoSQL)

Fully managed, multi-region, multi-master database.

Consistent single digit millisecond latency.

Reliable performance at any scale.

Built-in security, backup and restore and in-memory caching.

More than 100,000 customers has chosen DynamoDB for mobile, web, gaming, IoT and many other application types.

Benefits of DynamoDB

Performance at Scale	DynamoDB delivers consistent single-digit millisecond responsiveness at any scale. Build apps with virtually unlimited throughput and storage.
Fully Managed	DynamoDB is serverless database that automatically scales up and down; continuously backup you data for protection. DynamoDB gives your globally distributed apps fast access to local data by replicating table across multiple AWS regions.
Enterprise Ready	Built for mission critical workloads. Your data is secured with encryption and guaranteed reliability with a Service Level Agreement.

DynamoDB Use Cases

Serverless Web Apps	Build powerful web apps with automatically scale up and down.
Microservices Data Store	Build flexible and reusable microservices data store for consistent and fast performance
Mobile Backend	Build personalized mobile apps with smooth experience for your users.
Gaming	Create responsive games for mobile, console and desktop with DynamoDB.
IoT	Connect you high volume, high velocity IoT data.

AWS ELB

Balancing your App

Amazon ELB – Elastic Load Balancing

In simple English, an ELB divides the amount of work that a singe computer has to do among multiple computers.

ELB automatically distributes incoming app traffic across multiple EC2 instances (servers).

It can distribute your app traffic in a single AZ or across multiple AZs

Benefits of ELB

High availability	Secure	Elastic
Flexible	**Robust monitoring and Auditing**	**Hybrid Load Balancing**

ELB Types

Application Load Balancer

- Most widely used
- Load balancing of HTTP / HTTPS traffic.
- Best suited for modern app architecture viz microservices and containers

Network Load Balancer

- It's the latest.
- Load balancing of TCP traffic where extreme performance is required.
- Can handle millions of request per second while maintaining low-latency.
- Optimized to handle sudden and volatile traffics

Classic Load Balancer

- Provides basic load balancing

AWS Lambda

Go Serverless

Amazon Lambda

Run code without provisioning or managing infrastructure.

Save costs by paying only for the compute time you use—by per-millisecond—instead of provisioning infrastructure upfront for peak capacity.

Optimize code execution time and performance with the right function memory size

With Lambda, you can run code virtually for any type of application or backend service – all with Zero administration.

Benefits of Lambda

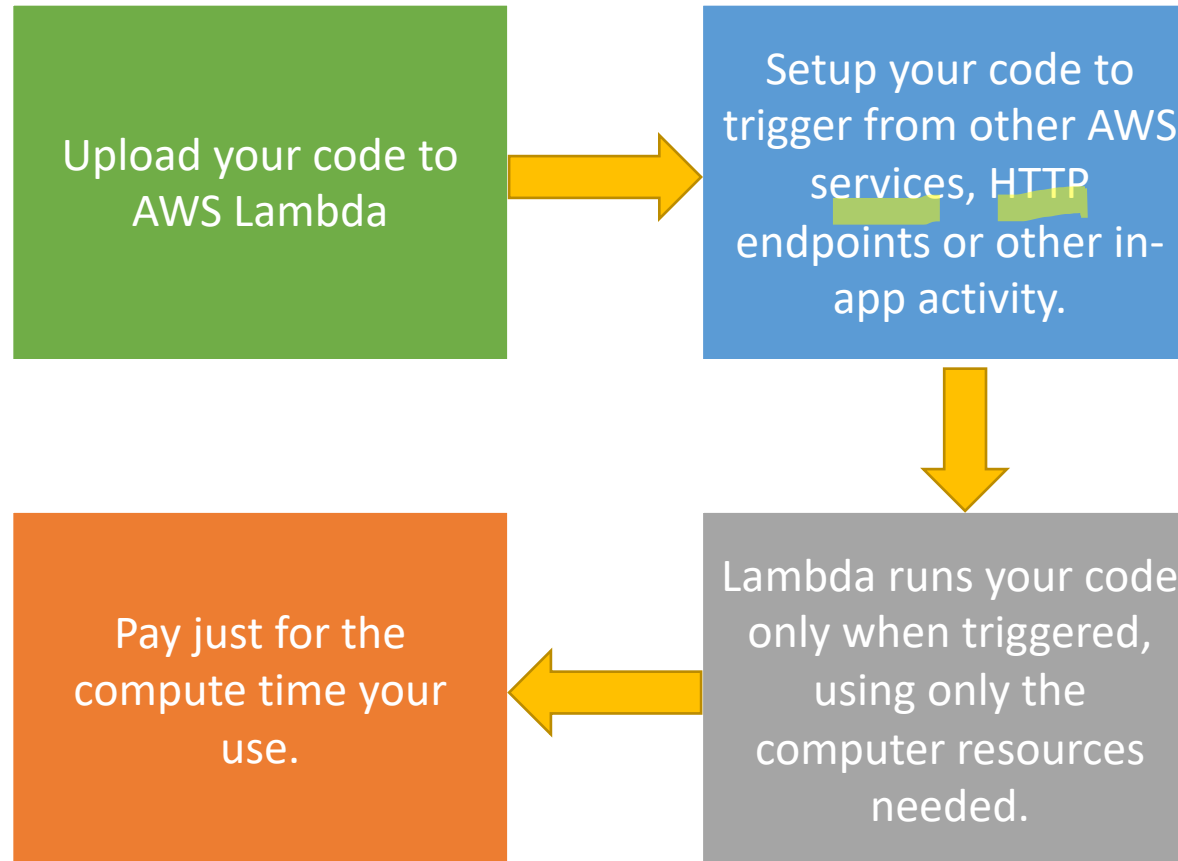| No Servers to Manage	Continuous Scaling	Sub-second Metering
• Lambda automatically runs your code without provisioning of server • Just write the code and upload it to Lambda	• Lambda auto scale your app by running code in response to each trigger. • Your code runs in parallel and processes each trigger individually, scaling precisely with the size of the workload.	• With Lambda, you are charged for every 100ms you code executes and number of times you code triggers. • You don't have to pay anything when your code is not running.

How Lambda works ?

Upload your code to AWS Lambda

Setup your code to trigger from other AWS services, HTTP endpoints or other in-app activity.

Lambda runs your code only when triggered, using only the computer resources needed.

Pay just for the compute time your use.

Lambda Use Cases

Web Apps	By combining Lambda with other AWS Services, developers can build powerful web apps that auto scale up and down and highly available configured data centers – with Zero Cost.
Real time File Processing	You can use Lambda and Kenisis to process real time streaming data for app activity tracking, transaction order processing, metrics generation, log filtering, social media analysis etc
Extract, Transform, Load	You can use Lambda for data validation, filtering, sorting and other trasnformations.
IoT Backend	You can build Serverless backend with Lambda to handle web, mobile, IoT and other 3rd party API requests.
Mobile Backend	Lambda makes it easy to create rich and personalized app experiences.

Language Runtime Supported by Lambda

Node.JS

Java

Go / GoLang

C#

Python

Thank You

Now you have required foundation to go ahead on you exciting and lucrative journey.

ALL THE BEST