Databases – Categories

1. SQL -> tables 🡪 we have rows(also called records) & columns
2. Non-SQL -> JSON 🡪 each set of data is called document

Faster compared to SQL

Why SQL when JSON is faster? We can’t store relational data

Foreign key column cant have id’s that are not available in it’s referential primary key column of the referenced table

**Stored Procedure**

It is a group of SQL statements that are stored together in DB. Based on parameters in procedure & values passed during call it can perform one or more DML operations & return values if any.

**Stored procedures vs Functions**

Stored procedures perform larger activities & can modify data whereas functions does computations & data retrieval providing flexibility in SQL activities.

Procedures can insert, update, delete or modify data whereas functions do specific task

Functions return a single parameter using return statement, procedures may not usually return parameters however can return single or multiple parameters.

Procedures can include transaction controls like commit/rollback whereas functions don’t.

Procedures can include exception handling using begin & end whereas functions don’t support.

Procedures are called using **call** or **execute** statements whereas functions can be called similar to column or expression

**Python**

Programming languages –

1. high level languages – C, C++, java, Python
2. Assembly level language – shud know in & out of hardware
3. Binary language – 0 & 1’s

Python – easy to use

Large library supporting different applications like web development, games, big data, testing, AI/data science, IOT, embedded systems

Free to use

X = 1

here X is called variable & 1 is called value

**AWS**

**Regions –(n.virginia, Mumbai,…..) 21+**

**Availability regions – 3+**

These are data centers with in a region that are isolated from each other in terms of failure. They are interconnected but separate facilities with power, cooling, network & located in close proximity.

It’s purpose is to provide redundancy & fault tolerance. If 1 AZ goes down due to some disaster, other AZ’s within same region remain operational

**Services**

1. **IaaS**
2. **Paas**
3. **SaaS**

**EC2**

**It is a service used to get or create a server in AWS**

**Server -> to connect to it, we need key pair-2 keys**

**Security group -> there would be traffic/requests coming to the server, using security group we control what requests it can take**

**S3 (Simple storage service)**– used to store

**Buckets –** are container objects that store data

**Downloading files using url** – we should select files in bucket – go to actions, click make public using ACL & then click on the file in the bucket it opens up a page where url is available it can be shared to download the file

**Versioning**

**Server access logging**

If I have 2 buckets & logging info of one bucket would be captured in second 1, if it’s enabled & ur destination bucket is selected

Properities -> server access logging

**Replication**

I have 2 buckets, data added on 1 bucket can be used to create copy of 1 bucket to other

**Static web hosting**

We can host a static web page from s3, we need to upload the index.html of ur web page to s3 bucket

**Life cycle policy**

Based on life cycle policy, data is moved from one storage class to other because of cost efficiency we do this

**Snapshot** – image of single volume

**AMI(Amazon Machine Image)** – consolidated snapshot of an ec2 instance(all volumes)

We can also take snapshots of individual volumes as well

Snapshot is backup of volume can be used to create similar volume but can’t be used directly as volume

**Storage classes** – we have different classes based on **availability.**

**Load Balancer –** It would receive http requests from different users & distribute among all available server & collect response, send back

**Apache web serevr**

**#!/bin/bash**

**# Use this for your user data (script without newlines)**

**# install httpd (Linux 2 version)**

**sudo yum update -y**

**sudo yum install -y httpd.x86\_64**

**sudo systemctl start httpd.service**

**sudo systemctl enable httpd.service**

**sudo sh -c 'echo "Hello World from $(hostname -f)" > /var/www/html/index.html'**

;Sa5pf-PJrA=;s=\*hhP\*hC9K3JcX0EA$ - windows instance password rdp

**VPC(**Virtual private cloud) isolated virtual network that allows you to launch AWS resources in a defined virtual network environment. It gives you control over your network settings, including IP addressing, subnets, routing tables, network gateways, and security settings.

– isolated network in the region

- would be available for all AZ’s within that region

- need to specify to which region the VPC belongs while creation

**1. create VPC Network**

**2. create subnet**

**3. create internet gateways**

**4. create route table – are connected subnets**

**We need to edit route ways – make route entries**

**& add subnet associations**

**Nat gateways –** create it under public subnet & include it in route table of private VPC to connect to private subnet from public & make any updations

NAT gateway is a way to connect to internet for private subnet to make any updations

**VPC peering**

If there are 2 vpcs & wanted to connect to them, its called vpc peering, useful for secured communication

**Cloud Watch**

Can be used for monitoring of servers

**Cloud trail**

Used to monitor aws account

**RDS(Relational DB server):**

1. Create server & install required DB installations
2. AWS service – RDS directly creates db server