==========
What is Infrastructure ?
==========

=> Infrastructure means The resources which are required to run our business

- Servers (Machines)
- Storage
- Security
- Network
- Power

=> To purchase this infrastructure we need to spend lot of money and time

Note: If we purchase and manage our infrastructure then it is called as On-Premise infrastructure.

=> Challenges with on-prem infrastructure

1) Lot of investment (time & money)
2) Man power
3) Security
4) Scalability (increase/decrease)
5) Availability
6) Backup

=> To overcome these challenges companies are using cloud infrastructure.

=> There are several companies in market to provide IT infrastructure for rent. Those companies are called as Cloud Providers.

1) Amazon (AWS)
2) Microsoft (Azure)
3) Google (GCP)
4) Salesforce

==========
What is Cloud Computing ?
==========

=> The process of delivering IT resources on demand basis through internet based on pay as you go model.

=> Pay as you go model means pay the money for usage

Ex: credit bill, post paid bill

=> We have several advantages with cloud computing

1) No investement
2) Less Cost
3) Scalability
4) Availability
5) Security
6) Unlimited Storage
7) Backup

==========
AWS Cloud
==========

=> AWS stands for Amazon Webservices.

=> Amazon company started AWS cloud in 2006.

=> Today 190+ countries using AWS cloud services

=> Instead of we are puchasing, managing and maintaining infrastructure simply we can use AWS cloud infrastructure.

=> AWS providing services based on pay as you go model.

=> AWS having global infrastructure using Regions & Availability Zones.

=> We can create free tier account (1 year validity)

=> In AWS cloud we have 200+ services (some are free and some are paid)

=> If we use paid service in AWS, bill be generated.

Note: in AWS, bill amount auto-deduction will not happen. If we don't pay bill amount aws will suspend our account.

Note: As a beginner we can request AWS support team to get free credits.

1) What is IT infrastructure
2) On-Premise infrastructure
3) Challenges with on-prem infrastructure
4) Cloud Computing
5) Advantages with Cloud computing
6) AWS cloud introduction
7) Regions and AZ's

==========
AWS Services
==========

EC2 : Elastic Compute Cloud => Virtual Servers

S3 : Simple Storage Service => Unlimited Storage

RDS : Relational Database Service => RDBMS

IAM : Identity and Access Management => Users, Groups & Roles

VPC : Virtual Private Cloud => Network

Route 53 : DNS => Domain Mapping

EKS : Elastic Kubernetes Service => K8s Cluster

Beanstack : Platform as a service

Lambdas : Serverless computing => Run application without thinking abt server

Cloud Watch : To monitor resources

==========
EC2 Service
==========

=> EC2 stands for Elastic Compute Cloud

=> EC2 is used to create virtual servers (machines)

=> EC2 is one of the most famous service in AWS cloud

=> EC2 is a paid service (hourly billing)

started @ 9:00 AM & stopped @ 9:15 AM => 1 hour billing

started @ 9:30 AM & stopped @ 9:45 AM => 1 hour billing

=> To encourage learners, AWS provided "t2.micro" as free of cost for 1 year. Monthly 750 hours.

==========
EC2 Components
==========

1) AMI
2) Instance Type
3) Keypair
4) VPC
5) Security Groups
6) EBS (storage)

1) AMI : Represents softwares configuration required for vm (os, server...)

Ex : Windows AMI, Amazon Linux AMI, Ubuntu AMI, Red HAT AMI....

2) Instance Type : Machine configuration

Ex: t2.micro(1GB), t2.medium(4GB), t2.large(8GB)....

Note: t2.micro instance is free tier eligible (1 year & monthly 750 hours)

3) Key Pair : It is used to estalish secured connection with EC2 VM.

- AWS will store public key
- AWS will provide private key for us

Note: one key pair we can use for multiple ec2 instances.

4) VPC : It provides Network for aws resources

Note: To encourage beginners, aws providing one default vpc.

5) Security Group : Firewall (inbound & outbound)

=> Inbound rules are used to allow incoming traffic
=> Outbound rules will allow outgoing traffic

Note: One machine can have multiple security groups also

Note: One security group we can use for multiple machines also.

Windows => RDP => 3389

Linux => SSH => 22

Webserver => HTTP => 80

MySQL => 3306

HTTPS : 443

6) Storage : EBS Volumes (hard disc/ssd)

Windows: 30 GB (default)
Linux : 8 GB (default)

Note: EBS vol max capacity : 16TB

==========
Types of IPs in AWS
==========

=> We have 3 types of IP's

1) Private IP : fixed ip address for internal communication (with in vpc).

2) Public IP : For outside acess (it is dynamic ip). When we restart vm then public ip gets changed.

3) Elastic IP : Fixed public ip (It is commercial)

private ip : 172.31.0.185

public ip : 3.109.211.21

afer restart : 3.110.183.46

==========
Lab practice on Elastic IP
==========

Step-1 : Allocate Elastic IP (getting from AWS)

Step-2 : Associate elastic ip to EC2 instance

Step-3 : Re-start EC2 instance(ip will not change)

Step-4 : De-Associate elastic ip from ec2

Step-5 : Release elastic ip to aws.

##########
16-July-2024
##########

==========
What is user data ?
==========

=> It is used to execute the script while launching EC2 VM.

Note: User data will execute only once when the machine is started.

```
#! /bin/bash

sudo su
yum install httpd -y
cd /var/www/html
echo > "<h1>Welcome to Ashok IT</h1>"
service httpd start
```

==========
What is Load Balancer
==========

=> When we run our application in single server then we have to face below challenges

1) One server should handle all reqs
2) Burden will increase on server
3) Response will be delayed for clients

```
        4) Server can crash
        5) Single Point Of Failure
        6) Business Loss

=> To avoid above problems, we will run our application using Load Balancer.

=> Load Balancer is used to distribute load to multiple servers in round robbin fashion.

=> We have below advantages with Load Balancer

        1) App will run in multiple severs

        2) Load will be distributed

        3) Burden will be reduces on servers

        4) Fast Performance

        5) High Availability


=> We have 4 types of load balancers in aws

        1) Application Load Balancer

        2) Network Load balancer

        3) Gateway Load Balancer

        4) Classic Load balancer (previous generation)

=======================
Load Balancer Lab Task
=======================

#### Step-1) Create EC2 VM-1 with below user data

#! /bin/bash
sudo su
yum install httpd -y
cd /var/www/html
echo "<html><h1>Life Insurance Server - 1</h1></html>" > index.html
service httpd start

#### Step-2) Create EC2 VM-2 with below user data

#! /bin/bash
sudo su
yum install httpd -y
cd /var/www/html
echo "<html><h1>Life Insurance Server - 2</h1></html>" > index.html
service httpd start

### Step-3) Add these 2 instances to one "Target Group"

### Step-4) Create Load Balancer with Target Group (ALB)

                        Schme : Internet Facing


### Step-5) Access our aplication using LBR DNS URL

============================================================
Assignment : https://www.youtube.com/watch?v=QvEJ8--zneU
============================================================
```

```
#############
17-July-2024
##############

=============
Auto Scaling
=============

=> Auto scaling is used to increase or decrease no.of servers based on incoming traffic.

                scale up -> increase the servers when required

                scale down -> decrease the servers when not required

=> We have below advantages with auto scaling

                1) high availability

                2) fault tolerance

                3) cost management


======================
EC2 service Summary
======================

1) What is EC2 & Why
2) What is AMI
3) What is instance type
4) What is key-pair
5) What is Security Group
6) How to launch Linux VM
7) How to connect with Linux VM using SSH client
8) Types IP's in AWS
9) What is User Data
10) What is Load Balancer
11) What is Auto Scaling


=============================================================================
Spring boot with AWS RDS DB : https://www.youtube.com/watch?v=GSu1g9jvFhY
=============================================================================

================================
Why to go for cloud database ?
================================

=> For every application database is required to store data permanently

=> We can use database in 2 ways

                1) On-Prem database

                2) Cloud Database

================================
Challenges with On Prem Database
================================

1) We need to take care of our DB server

2) We need to download & install DB server
```

3) We need to purchase DB licenses

4) We need to secure database server

5) We need to maintain DB backup

Note: To overcome all theses challenges we will use Cloud Databases.

==========
AWS RDS
==========

=> RDS stands for Relational Database service.

=> RDS is fully managed service in AWS cloud.

=> Using RDS service we can setup relational databases in aws cloud.

Ex: Oracle, MySQL, Postgres, SQL Server.....

=> If we use RDS then AWS will take care of our Database management.

Ex: License, security, backup

=> RDS works based on Pay as you go model.

##
Steps to create MYSQL DB using AWS RDS
##

1) Login into AWS management console

2) Goto RDS Service

3) Click on 'Create Database'

Choose a database creation method : Standard create
Engine Option : MySQL
Template : Free Tier
Username : admin
Password : Choose a passord
Public Access : yes
Initial Database name : jrtp

4) Click on 'Create Database' (It will take few minutes of time to create)

Note: Notedown username and password of the database

5) Once Database created, it will provide database Endpoint URL to access

6) Enable MySQL - 3306 port number in Security Group inbound rules.

======================
Database Credentials
======================

username : admin
password : ashokit2024
Endpoint URL : database-2.cnys8a2a2umm.ap-south-1.rds.amazonaws.com

Note: We can use MySQL workbench to check database connection. We can configure above credentials in

our springboot application.

=========
AWS IAM
=========

=> Identity & Access Management is a free service in aws cloud

=> IAM is used to manage below functionalities in AWS cloud

a) Users
b) Groups
c) Policies / Permissions
d) Roles

Note: Using this IAM service we can decide who can into login into our aws cloud account and which services they can access in our aws cloud account.

=> We can access AWS cloud platform in 2 ways

a) Root Account (super account)

b) IAM account (limited permissions)

Note: When we create account in AWs cloud by default we will get ROOT user account. Root user will have access for everything in AWS cloud.

Note: It is not recommended to use Root account for daily activities.

=> For daily activities we will use IAM account.

=> In one root account, we can create multiple IAM accounts.

Note: In Real-Time, for every team member one IAM account will be provided with limited permissions.

===============
IAM Lab Task
===============

1) Login into AWS account as root user

2) Go to IAM service and create IAM user with below policies

1) EC2FullAccess

Note: Provide web console access & generate security credentials

Note: Web Console access is used to login into aws account from browser.

=> Security Credentials are used to connect with AWS cloud with below options

a) Terraform
b) aws cli
c) java/dot net/python

========
AWS S3
=======

=> S3 stands for simple storage service

=> S3 is used for unlimited storage

=> S3 works based on Object storage (object = file)

```
=> S3 is a paid service (for storing & for retriving)

=> S3 maintains data using buckets concept

=> Buckets are used to seperate objects logically.
  Bucket means group of objects


        ex :     netflix_tollywood_movies_bucket
                        netflix_bollywood_movies_bucket
                        netflix_hollywood_movies_bucket

Note: When we create bucket, aws will generate one URL for that.

=> When we upload object in the bucket then object url will be generated.


================================
Static website hosting using s3
================================

-> Website means collection of web pages

-> Websites are divided into 2 types

                1) static website

                2) dynamic website

-> static website will give same response for all  users

-> dynamic website will give response based on user


Step-1 : Create S3 bucket

                  - Enable ACL
                  - Allow Public Access

Step-2 : Upload website files (index.html & error.html) as objects in bucket

                  - Grant public read access

Step-3 : Enable static website hosting

                  - bucket -> properties -> static website hosting

Step-4 : Access website URL

        URL : http://ashokitwebsite001.s3-website.ap-south-1.amazonaws.com/


===========
Assignment
===========

1) Develop spring boot web application to store course details

                        - course name
                        - course duration
                        - course price
                        - course image

Note: Course image should be stored into AWS s3 bucket and course info should be stored into db
```

table.

Note: In database we will store image url which is uploaded in aws s3 bucket.

Note: To implement this task we need IAM user with S3FullAccess & Security Credentials of IAM user.

Note: We need to use AWS SDK to perform this operation.

==========
Route 53
==========

=> It is a DNS service in AWS cloud.

=> DNS stands for Domain Name System

Note: DNS runs on port number 53.

=> It is used to map our application with domain name.

ex: www.gmail.com
www.facebook.com
www.naukri.com

======================
Route 53 Lab Task
======================

@@@@ Domain Mapping :: https://youtu.be/f7bfbUPSONI?si=tqr4jlHeF6pQXW4Z

1) Check domain availability & place the order

Note: the domain price changes based domain extensiom

.link -> 5 USD (least price)

2) Pay the amount for domain purchase

Note: Once the payment is completed then our domain gets activated.

3) Map application URL to domain name

4) Access our application using domain name.

==================
Elastic Beanstalk
==================

-> To deploy one java web application manually we need to perform below operations

1) Create Network

2) Create Security group

3) Create Virtual Machine (s)

4) Install Java software in Virtual machine

5) Install Webserver to run java web application

6) Deploy application to server

7) Re-start the server

8) Create LBR

9) Create AutoScaling etc..

Instead of we are preparing platform to run our application, we can use Elastic Beanstalk service to run our web applications

-> End-to-end web application management service in AWS cloud.

-> In AWS, Elastic Beanstalk provides Platform As a Service (PaaS).

-> Easily we can run our web aplications on AWS cloud using Elastic Beanstalk service.

-> We just need to upload our project code to Elastic Beanstalk then it will take care of deployment and infrastructure(resources) creation.

-> Elastic Beanstack will take care of softwares and servers which are required to run our application.

-> Elastic Beanstalk will take care of deployment, capacity provisioning, load balancer and auto scaling etc..

Note: Elastic Beanstack will charge for the resources which got created to run application.

==========
Serverless Computing
==========

-> Serverless computing means run the application without thinking about servers.

-> AWS will take care of servers required to run our application.

-> AWS lambdas are used to implement serverless computing

-> AWS Lambdas works based on pay as you use model.

-> Pay as you use means you pay the bill if your code is executed.

-> If we are not getting any requests to our application then we no need to pay the bill.

==========
Running Java Code with AWS Lambda
==========

1) Create Lambda Function with 'java 11' runtime

- Author from scratch
- Select 'Enable Functional URL' option
- Select Auth Type as None

2) Upload jar file in 'Code Source' section

3) Configure Handler in Runtime

Class Name : in.ashokit.LambdaHandler

Method Name : handleRequest

Handler Syntax : className :: methodName

Ex: in.ashokit.LambdaHandler::handleRequest

```
======================
AWS Cloud Summary
======================

1) What is IT infrastructure

2) On-Premise infrastructure

3) Challenges with On-Prem

4) Cloud Computing

5) Advantages with Cloud Computing

6) Cloud Providers

7) AWS Cloud Introduction

8) AWS Services Overview

9) Regions & Availability Zones

10) EC2

     - AMI
     - instance types
     - keypair
     - security groups
     - LBR
     - ASG

11) RDS

12) IAM
                        - Users
                        - Groups
                        - Policies
                        - Roles
13) S3
                        - Buckets
                        - Objects
                        - Static Website Hosting

14) Elastic Beanstack

15) Serverless Computing (Pay as you use)

16) AWS Lambda

17) Billing Overview
```