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

1) What is infrastructure

2) Challenges with Infrastructure Mgmt

3) What is Cloud Computing

4) Why Cloud Computing

5) Cloud Providers

6) AWS (Amazon Web Services)

- Cloud Platform
- 2006 onwards
- 190+ Countries
- Global Infrastructure
  - Regions
  - Availability Zones

================
AWS - Services
================

=> Offering 200+ services

1) EC2 : Elastic Compute Cloud (Virtual machines)

2) EBS : Elastic Block Store (storage for machine - hd/sdd)

3) S3 : Simple Storage Service (Unlimited storage)

4) RDS : Relational Database Service (Oracle / MySQL/ PostGres)

5) Route 53 : DNS Mapping (application url mapping to domain name)

6) IAM : Identity & Access Mgmt (user & user-permissions mgmt)

7) VPC : Virtual Private Cloud (Network for cloud resources)

8) Lambdas : Serverless computing (run your app without thinking about servers)

==
What service required for Java Developers
==

1) EC2
2) Load Balancer
3) S3
4) RDS
5) IAM
6) Lambdas

======================
Pay As You Go Model
======================

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

=> AWS will charge amount for services we have used (monthly bill will be generated)

=> To encourage new learners, AWS providing Free Tier account for 1 year

Note: As part of free tier account few services are free of cost.

=> If we use any paid service then monthly bill will be generated based on usage.

*** AWS will not deduct amount from our card directley.

*** AWS will send email reminders for bill payment.

*** If we don't pay bill amount, AWS will suspend our account (we can't login)

Note: If we get bill, we can request AWS support team to waive off bill (for first time)

========
AWS EC2
========

=> EC2 stands for Elastic Compute Cloud

=> It is used to create Virtual machines in cloud

Ex: Windows, Linux, Mac

=> EC2 is a paid service (hourly billing)

Note: Under free tier account we can use t2.micro instances for free of cost

(monthly 750 hours upto 1 year)

=================
EC2 Terminology
=================

=> AMI : Amazon Machine Image (It represents OS for our machine)

windows ami
linux ami
mac ami

=> Instance Type : It represents configuration

t2.micro (1 GB) ---> free tier eligble
t2.medium (4 GB)
t2.large (8 GB)

=> Key pair : For secured connection (pem file / ppk file)

=> Network : VPC provides required network for our machine

=> Storage : EBS will provide default storage

windows : 30 GB
linux : 8 GB

Task - 1 : Create Windows VM and connect to that VM using RDP Client.

Windows will run on RDP Protocaol (Port Number: 3389)

Task - 2 : Create Linux VM and connect to that VM using MobaXterm / Putty / Winscp

Linux will run on SSH protocol (Port Number: 22)

=========
Linux OS
=========

=> It is free OS
=> Multi User OS
=> Secured
=> CLI Based OS

===============
Linux Commands
==============

whoami : logged in username

pwd : present working directory path

date : current date

cal : display calendar

mkdir : To create directory

touch : To create empty files

cat : create file with content + append content + view file content

ls : list content

cd : Change directory

cp : copy content from one file to another file

mv : rename file/directory name

head : get first 10 lines of file

tail : get last 10 lines of file

grep : File search

vi : Visual Editor for file editing

rm : To remove files/ directory

ifconfig : To get ip address of computer

ping : To check connectivity

wget : To download a file from internet based on URL

curl : To send http request to given URL

==============================
Installing Softwares in Linux
==============================

=> We will use package managers to install softwares in linux

```
        Amazon linux / Red Hat Linux : yum

        Ubuntu Linux / Debian Linux : apt

------------------------
Install git client s/w
------------------------

git --version

sudo yum install git

git --version

--------------------------
Install Maven software
--------------------------

mvn -version

sudo yum install maven

mvn -version

java -version

# To install java latest LTS version
sudo yum install java

# Install java 11 v
sudo amazon-linux-extras install java-openjdk11

# install java 1.8v
sudo yum install java-1.8.0-openjdk


----------------
Assignment
----------------

1) Login into aws account and setup linux vm using ec2
2) Connect to linux vm using mobaxterm
3) Clone git repo in linux vm (springboot-api)
4) Change embedded server port in application.propertierties
5) package our application using maven
6) Run boot app in linux vm and share URL of our application to access

============================
ðŸ”¥ Today's Assignment ðŸ”¥
============================

1) Create EC2 VM using Amazon Linux AMI

2) Connect to VM using MobaXterm / Putty

3) Install MYSQL DB Server

4) Test MySQL DB connectivity with Workbench

5) Configure MySQL DB details in Spring Boot application

6) Test SpringBoot app functionality

*Note: App should be able to perform DB operations*
```

```
============
DB Setup
============

=> Take EC2 vm
=> Install DB server in EC2 VM


Note: If we setup db on our own we have to deal with below challenges

1) Setup DB server
2) Handle security
3) Handle backup
4) Handle Administration


Note: If someone delete our ec2 vm then we will loose our db

=> To overcome these problems, AWS provided RDS

================================
RDS - Relational Db service
================================

=> RDS is a fully managed service in AWS

=> It is a paid service.

=> We can use RDS based on pay as you go model.


==========
AWS - IAM
==========

=> To use AWS cloud services we need AWS account

=> We have 2 types of accounts in AWS

                        1) Root Account (super account)

                        2) IAM Account (limited permissions)

Note: When we signup in AWS by default it will become root account.

=> Root account is very powerful account, we can access everything in aws using root account.

#### Note: For every root account one unique account number will be available.


                        AWS Account NO : ################

Note: We shouldn't share our AWS root account credentials with anyone.


=> In company, we will get IAM account to use AWS cloud services in our project.

Note: In project, for every team member IAM account will be provided with limited permissions.


### IAM - Identity & Access Management
```

=> Using IAM service, we can manage users, groups, permissions and roles.

1) Web Console Access (Login access through UI)

2) Programmatic Access (Access Key & Secret Key)

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

S3 => Simple Storage Service in AWS Cloud

=> It is used to store unlimited data in AWS cloud

=> S3 is object based storage

Object = file (txt / pdf / audio / video)

Ex: Amazon Prime

- Movies
- Web-series
- Standup comedies

=> To seggregate our objects we will use Buckets in S3

Note: Bucket means collection of objects

=> Once we upload object in bucket, it will generate URL for our object.

Note: By default objects are private (we can make them as public also)

Note: S3 is paid service

==========
Static Website Hosting Using S3
==========

-> Create S3 Bucket

-> Upload website content in s3 bucket as objects

=> Enable Static website hosting

(Bucket -> Properties -> Static Website hosting)

Configure index.html & error.html

Note: It will generate URL to access our website.

http://mywebsite0011.s3-website.ap-south-1.amazonaws.com/

==========
Application URL mapping to Domain Name
==========

=> When we host our website we got lengthy URL

=> When can't share lengthy urls to customers

Note: We need to map application lengthy url / ip address to domain name

```
                Ex:  www.gmail.com
                     www.flipkart.com
                     www.irctc.com


=> We can use Route 53 service for domain mapping


1) Purchase domain in Route 53
2) Pay domain bill amount
3) Map domain to app url


================================
What is Server less Computing ?
================================

=> Run your application without thinking about servers

=> To achieve serverless computing we will use AWS Lambdas

=> AWS Lambdas will charge based on "Pay As You Use" approach


Note: Our code will be executed using Lambda fucntions in AWS

=> If our code is executed then only bill will be generated.



===========
Summary
==========

Linux

EC2 : To create Virtual Machines

RDS : To create Cloud Database

IAM : Identity & User Access Mgmt

S3 : Unlimited Storage

Route 53 : Domain Mapping

Lambdas : Serverless Computing
```