

Day-1

1)**Generative AI:** Generative AI is a type of AI that can create new content such as text, images, code, audio, or videos based on learned patterns from data.

Example: ChatGPT generating text, DALL-E creating images, for AWS Bedrock.

2)**Agentic AI:** Agentic AI refers to AI systems that can **make decisions, plan steps, and take actions independently** to achieve a goal with minimal human input.

3)**AI Agent:** An AI agent is a software entity that **perceives its environment, makes decisions, and performs actions** to complete a task. It can use tools, APIs, or other systems.

Latency of cloud provider:

1)<https://www.cloudping.info/> 2)<https://cloudpingtest.com/azure>

Day 2

What is Cloud?

Cloud is a network of remote servers on the internet used to store, manage, and process data instead of local computers.

How many types of cloud Computing Models?

Cloud computing models are classified into Service Models and Deployment Models.

1. Service Models (What is provided)

These models define **what type of cloud service is offered:**

1. **IaaS (Infrastructure as a Service)**
 - Provides virtual machines, storage, and networking
 - Example: AWS EC2
2. **PaaS (Platform as a Service)**
 - Provides platform and runtime environment for applications
 - Example: Google App Engine
3. **SaaS (Software as a Service)**
 - Provides ready-to-use software over the internet
 - Example: Gmail, Salesforce

2. Deployment Models (How it is deployed)

These models define **how and where cloud services are deployed:**

1. **Public Cloud:** Shared infrastructure available to all users
2. **Private Cloud:** Dedicated infrastructure for a single organization
3. **Hybrid Cloud:** Combination of public and private clouds
4. **Community Cloud:** Shared by organizations with common requirements

What is Data Center / Cloud Provider? (AWS, GCP, Oracle, Salesforce)

Answer:

A data center or cloud provider is a company that owns and manages large server facilities and provides cloud services over the internet.

Examples: AWS, Google Cloud (GCP), Oracle Cloud, Salesforce.

What is Cloud Computing?

Answer:

Cloud computing is the practice of **using cloud services like servers, storage, databases, and software over the internet** on a pay-as-you-use basis.

How does Cloud Computing work?

Answer:

Cloud computing works by hosting applications and data on remote servers. Users access these resources through the internet without managing physical hardware.

What is Virtualization?

Answer:

Virtualization is a technology that allows **multiple virtual systems to run on a single physical machine** by sharing hardware resources.

What is a Virtual Machine (VM)?

Answer:

A virtual machine is a **software-based computer** that runs an operating system and applications like a real computer.

What are Deployment Models?

Answer:

Deployment models define how cloud services are provided:

- **Public Cloud:** Shared resources, open to all users
 - **Private Cloud:** Dedicated to one organization
 - **Hybrid Cloud:** Combination of public and private clouds
 - **Community Cloud:** Shared by organizations with common needs
-

In GCP everything starts with Project.

How to create a project in GCP. ----- interview Question

In project in GCP in azure it called as resource group.

Processor: A processor, also called **CPU (Central Processing Unit)**, is the brain of the computer that **executes instructions, performs calculations, and controls system operations**.

Types of Processors:

- **x86 Processors** – Used in desktops and servers (Intel, AMD)
 - **ARM Processors** – Used in mobiles and low-power devices (Apple M-series)
-

Whenever we access any terminal it open in below state
riteshdevops:~\$

Note: “~” Here it indicate “**user home**” location.

Day 3

The top 6 XaaS (Anything as a Service Model):

1. SaaS – Software as a Service

Meaning: Software delivered over the internet

Explanation: Users access software via a browser without installing or maintaining it.

Example: Gmail, Google Docs, Salesforce

2. PaaS – Platform as a Service

Meaning: Platform for application development

Explanation: Provides runtime, OS, and development tools so developers can build and deploy apps without managing infrastructure.

Example: Google App Engine, AWS Elastic Beanstalk

3. IaaS – Infrastructure as a Service

Meaning: Virtualized computing infrastructure

Explanation: Offers servers, storage, and networking on demand. Users manage OS and applications.

Example: AWS EC2, Microsoft Azure VM

4. FaaS – Function as a Service

Meaning: Serverless computing model

Explanation: Developers run code as functions triggered by events, without managing servers.

Example: AWS Lambda, Azure Functions

5. DaaS – Data as a Service

Meaning: Data delivered on demand

Explanation: Provides data through APIs or cloud platforms for analysis and decision-making.

Example: Google BigQuery, Snowflake

6. STaaS – Storage as a Service

Meaning: Cloud-based storage service

Explanation: Offers scalable storage without managing physical disks or hardware.

Example: Amazon S3, Google Cloud Storage

Model	Provides
SaaS	Software
PaaS	Development Platform
IaaS	Infrastructure
FaaS	Event-driven Functions
DaaS	Data Access
STaaS	Storage

Practical performed:

Step 1 :- Go to any cloud provider(GCP / AWS / Azure / etc)

Step 2 :- Create Virtual machine

Step 3 :- install docker

```
sudo -i
```

```
apt-get update && apt install docker.io -y
```

```
docker -version
```

Step 4 :- Download website from <https://www.tooplate.com/free-templates>

wget https://www.tooplate.com/zip-templates/2153_fireworks_composer

Step 5 :- Run the nginx container with downloaded website

```
docker run -ti -d -p 80:80 -v /root/2153_fireworks_composer:/usr/share/nginx/html nginx
```

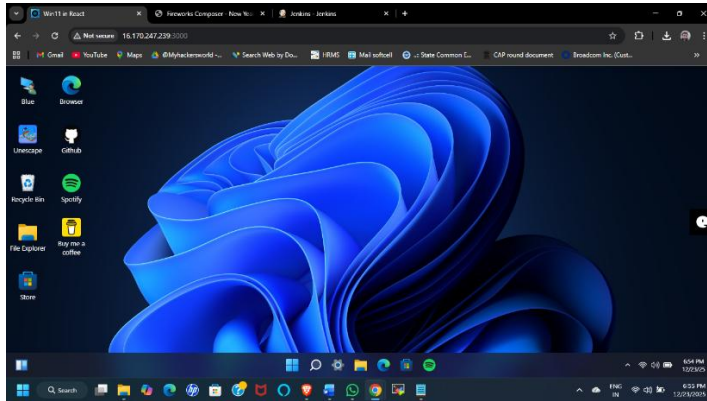
Step 6 :- Go to IP address:80 and verify

Run App on container with different port

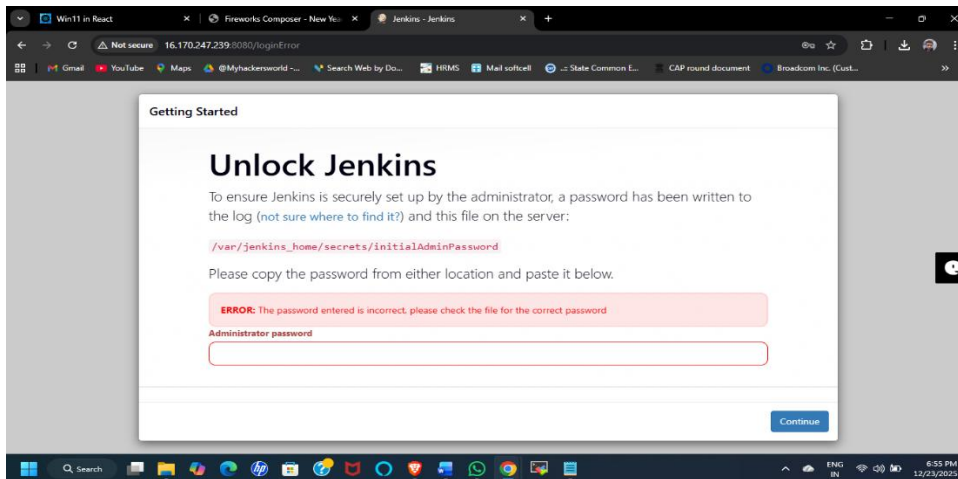
1)docker run -ti -d -p 80:80 -v /root/2153_fireworks_composer:/usr/share/nginx/html nginx



2) `docker run -d --restart unless-stopped --name win11react -p 3000:3000 blueedge/win11react:latest`



3) `docker run -it -d -p 8080:8080 jenkins/jenkins:latest`



Day 4

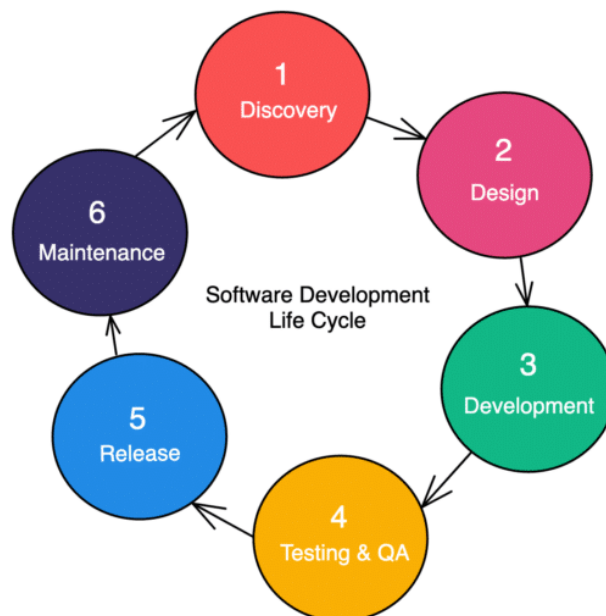
DevOps

DevOps is the solution for development and operation issue/problems.

DevOps helps teams for communication, collaboration and integration.

DevOps aims to shorten the Software development life cycle and provide continuous delivery with high quality software.

SDLC: - Software Development Life Cycle (SDLC) is a systematic process used to plan, design, develop, test, deploy, and maintain high-quality software.



Waterfall Model: –

The Waterfall model is a **linear and sequential SDLC model** where each phase must be completed before moving to the next phase. Once a phase is finished, changes are difficult to implement.

Agile Model: –

Agile is an **iterative and incremental SDLC model** where software is developed in small cycles called **sprints**, allowing continuous customer feedback and frequent releases.

DevOps: -

DevOps is a culture, methodology, and set of practices that integrates software development (Dev) and IT operations (Ops) to enable faster, reliable, and **continuous delivery** of high-quality software through collaboration, automation, and monitoring.

Note: -

Waterfall is suitable for projects with **fixed requirements**, **Agile** works best for projects needing **flexibility and continuous feedback**, and **DevOps** is ideal for fast, **automated, and continuous delivery** in dynamic environments.

Code Management: –

Code management is the **practice of tracking, controlling, and collaborating** on software code using version control systems **like Git and platforms like GitHub**.

Note: **Git** is a tool used to track and manage code changes, while **GitHub** is a platform that hosts Git repositories online, allowing teams to collaborate, review code, and manage projects.

Continuous Delivery (CD): manual approval

Code is automatically built and tested, but **production deployment needs manual approval**.

Continuous Deployment: Fully automatic

Code is automatically built, tested, and **deployed to production without any manual step**.

DevOps Life Cycle:

DevOps lifecycle is a continuous loop of planning, development, deployment, and monitoring using automation tools to deliver faster and reliable software.

Plan → Code → Build → Test → Release → Deploy → Operate → Monitor



1. **Plan** – Requirement planning and tracking
Tools: Jira, Azure Boards
2. **Code** – Application development and version control
Tools: Visual Studio, Git, GitHub, GitLab
3. **Build** – Convert source code into executable
Tools: Maven, Gradle

4. **Test** – Automated testing for quality assurance
Tools: Selenium (Web/Application testing), JMeter(performance)
5. **Release** – Package and prepare the application
Tools: Jenkins (in virtual or 3tire environment) for Helm (Kubernetes environment)
6. **Deploy** – Deploy application to servers/cloud
Tools:
Docker, Kubernetes -> Containerized deployment
Ansible -> Application deployment and operation activities
Terraform -> Infra deployment
7. **Operate** – Run and manage applications
Tools: Ansible(Best), AWS, Azure
8. **Monitor** – Monitor performance and logs
Tools: CloudWatch (for AWS Cloud), Prometheus & Grafana (for k8s), Splunk (for log)

Day5 – Linux

1) Docker Playground lab:-

<https://labs.play-with-docker.com>

2) Tool: -

mobaxterm lab Application

1. What is an Operating System (OS)?

An **Operating System** is system software that acts as an **interface between the user and computer hardware**.

It manages **CPU, memory, storage, files, devices, and processes** and provides services for applications to run.

Examples: Windows, Linux, macOS, Unix

2. Linux is a Kernel – what does that mean?

Linux itself is only a kernel, not a full OS.

When the Linux kernel is combined with tools, libraries, and a shell, it becomes a **Linux Operating System (distribution)**.

Examples of Linux OS: Ubuntu, Red Hat, CentOS, Debian

3. What is a Kernel?

The **kernel** is the core of the operating system that **connects hardware with software**, managing CPU, memory, storage, and devices so applications can run properly.

Note:

- The kernel runs in **privileged mode** and directly interacts with hardware.
-

4. What is a Shell?

A **shell** is a **command-line interpreter** that allows users to communicate with the kernel.

Role:

- Takes user commands
- Interprets them
- Passes them to the kernel for execution

Common shells:

- bash (Bourne Again Shell)
- sh

- zsh
- csh

5. What is DOS?

DOS (Disk Operating System) is a **basic operating system** that lets users interact with a computer using **text-based commands**.

It supports **single user and single task at a time** and was mainly used in early computers.

Key points:

- Command-line based
- No multitasking
- Limited memory management
- Mostly used in early PCs

Example: MS-DOS

6. What is Unix and Linux? (Difference)

Unix is a **commercial, multi-user, multitasking operating system** developed at **AT&T Bell Labs**.

It is **not free**, very **stable and secure**, and mainly used in **enterprise servers**.

- **Examples:** AIX, Solaris, HP-UX

Linux is a **free, open-source, Unix-like operating system** developed by **Linus Torvalds**.

It supports **multi-user and multitasking** and is widely used in **servers, cloud, and DevOps**.

- **Examples:** Ubuntu, Red Hat, CentOS
-

7. What is APT?

APT (Advanced Package Tool) is a **package management system** used in Debian-based Linux distributions.

Purpose:

- Install software
- Update packages
- Resolve dependencies automatically

Example commands:

apt update

apt upgrade

apt install nginx

8. Command Explanations (Interview Style)

1)hostname

2)hostname -i [Displays the **internal/private IP address** of the system Used to identify the system inside a network]

curl ifconfig.me [Shows the **public IP address** of the system Used to check internet-facing IP]

Note (Interview): RUN v/s CMD

Run is OS level command.

CMD is application-level command.

2)[.] means present directory

UAM (User Access Management) Team:

adduser ritesh =>Create User

passwd ritesh =>change password

Userdel ritesh =>delete user

addgroup devops =>creat group

usermod -a -G devops(grp_name) ritesh(user_name) =>adding user in group

getent group – show all entry group/user and information =>**VIMP Interview Question**

delgroup devops(grp_name) =>remove a group from system.

Note: [Users which are part of devops grp will be retain in system] =>**VIMP Interview Question**

chage ritesh =>change user password expiry information

ifplugstatus =>tell whether a cable is plugged or not.

Generate public key and Private key

Command:

ssh-keygen

Notes:

cd /root/.ssh/ ---→location of key

id_ed25519 ---→Private key

id_ed25519.pub ---→Public key

```

root@ip-172-31-2-61:~# ssh-keygen
Generating public/private ed25519 key pair.
Enter file in which to save the key (/root/.ssh/id_ed25519):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_ed25519
Your public key has been saved in /root/.ssh/id_ed25519.pub
The key fingerprint is:
SHA256:NxGM3Yyp6YIc8nha7QaXbYYUvLawuK5jj92LTzbgILQ root@ip-172-31-2-61
The key's randomart image is:
+--[ED25519 256]--+
|      .    +.=    |
|      o    =.o    |
|      o o.    |
|. o o + o .    |
|.E B O *S o    |
|. = X B =. .    |
|      * B +    |
|..= = +    |
|o=o+=.=.    |
+-----[SHA256]-----+
root@ip-172-31-2-61:~#

```

=====Docker=====

Install Docker by script:

```
curl -fsSL https://get.docker.com -o install-docker.sh
```

```
sh install-docker.sh
```

```
docker run -it -d -p 8080:8080 jenkins/jenkins:latest
```

```
docker --version
```

```
docker ps -a
```

```
docker pull hello-world
```

```
docker images
```

```
docker run hello-world
```

```
docker exec -it <container_name_or_id> /bin/bash → command connect to a running container
```

Steps to build docker file:

```
FROM ubuntu
```

```
MAINTAINER riteshv5678@gmail.com
```

```
RUN apt-get update
```

```
RUN apt-get install nginx -y
```

```
CMD ["echo","Image created"]
```

Above docker file will create docker image

Below step will upload image to docker hub

docker login

docker tag 7cb4718674b6 ritesh9967/mydocker42:latest

docker push ritesh9967/mydocker42:latest
