PpSessions 9 & 10 – Lecture Topics

1. Introduction to DevOps

DevOps is a set of practices that combines software development and IT operations. Its main goal is to shorten the development lifecycle and deliver high-quality software continuously. DevOps emphasizes collaboration, automation, integration, and continuous feedback between development and operations teams.

2. DevOps Ecosystem

The DevOps ecosystem includes a wide range of tools and practices that support the entire software delivery lifecycle. Common categories include:

- Version control (e.g., Git)
- Continuous Integration/Deployment (CI/CD) tools (e.g., Jenkins, GitLab CI)
- Configuration management (e.g., Ansible, Chef)
- Monitoring and logging (e.g., Prometheus, ELK stack)
- Containerization and orchestration (e.g., Docker, Kubernetes)

3. DevOps Phases

The DevOps lifecycle typically includes the following phases:

- **Plan** – Define and prioritize work.
- **Develop** – Code and build.
- **Build** – Compile and create binaries or artifacts.
- **Test** – Run automated and manual tests.
- **Release** – Prepare and approve deployment.

- **Deploy** – Deploy to production or staging.
- **Operate** – Maintain and monitor application.
- **Monitor** – Gather metrics and insights for feedback.

4. Introduction to Containerisation

Containerisation is the process of bundling an application with all its dependencies, libraries, and configuration files into a single package called a container. This ensures consistent performance across different environments and simplifies deployment.

5. Introduction to Docker

Docker is a platform that enables developers to create, deploy, and run applications in containers. It offers portability, scalability, and efficiency by isolating applications from the underlying infrastructure.

6. Creating Docker Images Using Dockerfile

A Dockerfile is a text file with a set of instructions used to create a Docker image. Key Dockerfile commands include:

- `FROM` – Base image
- `COPY` – Copy files into the image
- `RUN` – Execute commands inside the image
- `CMD` – Default command to run

7. Container Lifecycle

The lifecycle of a container includes:

- **Create** – Define the container

- **Start** – Run the container
- **Run** – Execute processes in the container
- **Pause/Stop** – Temporarily or completely stop processes
- **Remove** – Delete container and resources

Session 11 – Lecture Topics

1. Introduction to YAML

YAML (YAML Ain't Markup Language) is a human-readable data format often used in configuration files. It is indentation-sensitive and is used extensively in tools like Kubernetes, Ansible, and Docker Compose.

Example:

yaml
CopyEdit

```
apiVersion: v1
kind: Pod
metadata:
  name: mypod
spec:
  containers:
  - name: mycontainer
    image: nginx
```

2. Introduction to Docker Swarm and Docker Stack

- **Docker Swarm**: A native clustering and orchestration tool for managing a group of Docker engines. It supports scaling, load balancing, and fault tolerance.
- **Docker Stack**: Allows you to deploy a full application stack using a `docker-compose.yml` file in Swarm mode.

3. Introduction to Kubernetes

Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications. It helps manage workloads using objects like:

- **Pods**
- **Deployments**
- **Services**
- **ConfigMaps** and **Secrets**

4. Creating a Kubernetes Cluster

Creating a cluster involves:

- Master node (control plane)
- Worker nodes
  Tools for setup include:
- **Minikube** for local clusters
- **kubeadm** for production clusters
- Cloud providers (GKE, EKS, AKS)

5. Creating Services in Kubernetes

Services expose your application to the network and other pods. Common types:

- **ClusterIP** – Internal access only
- **NodePort** – Exposes app on each node’s IP at a static port
- **LoadBalancer** – Provisions an external IP

6. Deploying an Application Using Dashboard

Kubernetes Dashboard is a web-based UI to manage and monitor the cluster. Features include:

- Deploying and scaling applications
- Viewing logs and metrics
- Managing secrets, config maps, and volumes