**Topic: DevOps**

**Team Members**

Muhammad Raees 19F-0963

Mummad Ahmad 19F-0101

Sohaib Hassan 19F-0930

**Meeting Agenda:**

Discussion on first application deployment on Cloud server.

**Discussed:**

* **M. Ahmad:** I successfully deployed a Node.js project on AWS by following a manual deployment process. First, I set up an Amazon EC2 instance in my preferred AWS region, ensuring it had the necessary security group configurations. After connecting to the instance via SSH, I installed Git and Node.js. I then cloned my project repository from GitHub onto the EC2 instance, installed project dependencies, and set up any required environment variables. Starting the Node.js application with PM2 kept it running smoothly in the background. Optionally, I configured domain and DNS settings, set up Nginx as a reverse proxy, and ensured proper security group and firewall configurations. Monitoring, regular updates, and backups were crucial for maintenance. This deployment process allowed me to host my Node.js application securely on AWS.
* **M. Raees:** The seamless integration of AWS instance, Jenkins, and GitHub for deploying a Node.js application represents a modern and efficient approach to software development and deployment. Leveraging the power of cloud infrastructure with AWS, the automation capabilities of Jenkins, and the collaborative nature of GitHub, this setup streamlines the entire development lifecycle. It allows developers to easily build, test, and deploy their Node.js applications while ensuring consistency and reliability in the process. By embracing this integrated approach, teams can achieve faster development cycles, rapid code delivery, and a more efficient, automated workflow that ultimately enhances productivity and maintains the quality of their applications.
* **Sohaib Hassan:**I executed a manual deployment of my Node.js project on AWS, establishing an EC2 instance with the necessary security group configurations. After installing Git and Node.js, I cloned the project from GitHub and managed dependencies, environment variables. Continuous maintenance tasks, such as monitoring, updates, and backups, were implemented to ensure secure and stable hosting on AWS.