DevOps Engineer

Project 1: Automated Deployment Pipeline with Jenkins and Docker

- **Objective**: Implement an automated CI/CD pipeline using Jenkins, Docker, and Ansible.
- **Description**: Develop a pipeline to automate the build, testing, and deployment of a sample application. Utilize Jenkins for continuous integration, Docker for containerization, and Ansible for configuration management. Implement automated testing and deployment to a cloud environment.
- **Technologies to use**: Docker, Kubernetes, Jenkins, Ansible, CI/CD, Cloud Deployment.

Week 1: Initial Setup & Planning
Tasks:

- **Install Jenkins and Docker:** Set up the Jenkins server and Docker on local or cloud environments.
- **Create a basic Dockerized application:** Develop a simple application (e.g., a web app) and Dockerize it.
- **Set up Ansible:** Install Ansible for configuration management.
- **Pipeline Plan:** Plan the structure of the CI/CD pipeline (build, test, deploy) and document tasks.

Deliverables:

- Jenkins and Docker environments installed and configured.
- Dockerized application running locally.
- Basic Ansible setup.
- CI/CD pipeline plan documented.

Week 2: Jenkins & CI Integration
Tasks:

- **Create Jenkins Jobs:** Configure Jenkins to build the Dockerized application.
- **Integrate Git:** Set up GitHub or GitLab repository and integrate with Jenkins for continuous integration.
- **Automated Testing:** Add basic testing (e.g., unit tests) into the Jenkins pipeline for automated builds.
- **Set Up Notifications:** Configure email/Slack notifications for pipeline success or failure.

Deliverables:

- Working Jenkins pipeline triggered by Git commits.
- Jenkins job that builds Docker images.

- Automated testing included in the pipeline.
- Notifications configured for pipeline updates.

Week 3: Docker & Deployment
Tasks:

- **Integrate Docker Hub or private registry:** Push Docker images to Docker Hub or a private registry.
- **Configure Ansible for Deployment:** Write Ansible playbooks to automate application deployment to a cloud environment (e.g., AWS, GCP).
- **Deployment Testing:** Test deployment process with Docker and Ansible to ensure smooth automation.

Deliverables:

- Docker images automatically pushed to Docker Hub or a registry.
- Ansible playbooks ready and tested.
- Successful deployment of the application to a cloud server.

Week 4: CI/CD Refinement & Final Testing
Tasks:

- **Kubernetes Integration:** Optional task — configure Kubernetes to orchestrate the deployment of Docker containers if desired.
- **Refine the Pipeline:** Refine Jenkins jobs for efficiency (e.g., parallel stages, caching).
- **Full Testing and Deployment:** Conduct full tests of the CI/CD pipeline, from code commit to deployment.
- **Documentation:** Complete documentation of the project, including pipeline design, deployment process, and any issues faced.

Deliverables:

- Kubernetes integration (if applicable).
- Fully refined CI/CD pipeline in Jenkins.
- Documented CI/CD pipeline process.
- Final deployment verified and tested on the cloud.