| **Track(s): Devops** |
| --- |
| **Project ID: 206** |
| **Project Title: Simple-Flask-Application** |
| **Group ID: Group 3** |
| **Project Short 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. |
| **Project Context (third party interfaces, APIs, or other third party tools that will interact with your proposed design):**  **Jenkins (CI/CD Tool):**   * **Jenkins will connect to a Git repository (e.g., GitHub) to automatically start the pipeline when new code is pushed.** * **It will also send notifications via Slack or email to update on the status of the pipeline (success or failure).**   **Docker (Containerization):**   * **Docker will package the application into containers for easy deployment.** * **Jenkins will interact with Docker Hub to store and retrieve container images.**   **Ansible (Configuration Management):**   * **Ansible will automate the deployment process to a cloud environment (AWS).** * **It will interact with AWS to set up the infrastructure (e.g., EC2 instances) and deploy the application.**   **AWS (Cloud Provider):**   * **The application will be deployed to AWS.** * **Ansible and Jenkins will work with AWS tools (e.g., EC2) to manage the deployment and infrastructure.**   **Automated Testing Tools:**   * **Automated tests will run after each build to ensure the application works as expected.** * **Tools like PyTest will be used for testing** |
| **Example Application of the Proposed Project:**  This project will automate the process of building, testing, and deploying a simple web application. Here’s how it works:   1. **Code Push**:    * A developer updates the application (e.g., a Python web app) and pushes the changes to a GitHub repository. 2. **Jenkins Pipeline Starts**:    * Jenkins automatically detects the new code and starts the pipeline.    * It pulls the latest code from GitHub. 3. **Build the Application with Docker**:    * Jenkins builds a Docker image of the application, packaging it with all its dependencies.    * The image is then pushed to Docker Hub for storage. 4. **Run Automated Tests**:    * Jenkins runs automated tests to make sure the new code works properly.    * If the tests fail, the team is notified via Slack or email. 5. **Deploy with Ansible**:    * If tests pass, Ansible automatically deploys the updated application to an AWS EC2 instance.    * The application is now live on the cloud. 6. **Notifications**:    * Throughout the process, Jenkins sends notifications to the team to update them on the status of the build, test, and deployment. |
| **Tools (Hardware and Software needed for the project / needed to build a prototype):**  **Docker, Jenkins,Git,GitHub,Ansible, Terraform,AWS, CICD** |
| **Deliverables:**  **· CI/CD pipeline plan documented.**  **· Working Jenkins pipeline triggered by Git commits.**  **· Jenkins job that builds Docker images.**  **· Docker images automatically pushed to Docker Hub or a registry.**  **· Ansible playbooks ready and tested.**  **· Successful deployment of the application to a cloud server.**  **· Kubernetes integration (if applicable).**  **· Fully refined CI/CD pipeline in Jenkins.**  **· Documented CI/CD pipeline process.**  **· Final deployment verified and tested on the cloud.** |
| **Project Notes (Background, Data, references…..etc):**  Simple-Flask-Application |