**1. GitHub Repository Setup:**

* **Task:**
  + Create a new GitHub repository named "CloudDevOpsProject"
  + Initialize the repository with a README.
* **Deliverables:**
  + URL to the GitHub repository.

**2. Containerization with Docker:**

* **Task:**
  + Deliver Dockerfile for building the application image.
  + Find the source code from: https://github.com/IbrahimAdell/FinalProjectCode.git
* **Deliverables:**
  + Dockerfile is committed to the repository.

**3. Container Orchestration with kubernetes:**

* **Task:**
  + Setup Kubernetes cluster.
  + Create iVolve namespace.
  + Configure Deployment, service, ingress to access the application.
* **Deliverables:**
  + Required YAML files are committed to the repository.

**4. Infrastructure Provisioning with Terraform:**

* **Task:**
  + Deliver Terraform scripts for AWS resource provisioning:
    - VPC, Subnet, Security Groups.
    - 2 EC2 instances for application deployment.
    - Use S3 Terraform Backend state.
    - Integrate CloudWatch for monitoring.
    - Use Terraform Modules.
* **Deliverables:**
  + Terraform scripts are committed to the repository.

**5. Configuration Management with Ansible:**

* **Task:**
  + Deliver Ansible playbooks for EC2 instance configuration:
    - Install the required packages (e.g., Git, Docker, Java).
    - Install the required packages for Jenkins.
    - Install the required packages for SonarQube.
    - Set up necessary environment variables.
    - Use Ansible roles.
    - Use Dynamic Inventory.
* **Deliverables:**
  + Ansible modules and inventory are committed to the repository.

**6. Continuous Integration with Jenkins:**

* **Task:**
  + Deliver Jenkins pipeline configuration in Jenkinsfile:
    - Stages: Unit Test, SonarQube Test, Build JAR, Build Image, Push Image, Delete Image Locally, Update Manifests, Push Manifests.
    - Use Shared Library.
    - Use Jenkins slave.
* **Deliverables:**
  + Jenkins file is committed to the repository.
  + Shared library directory (vars) is committed to the repository.

**7. Continuous Deployment with ArgoCD:**

* **Task:**
  + Configure ArgoCD to sync and deploy your app into cluster.
* **Deliverables:**
  + ArgoCD Application is committed to the repository.

**8. Documentation:**

* **Task:**
  + Deliver comprehensive documentation:
    - Setup instructions.
    - Architecture overview.
    - Troubleshooting guidelines.
* **Deliverables:**
  + Documentation is available in the repository.