**CI/CD Pipeline Setup Documentation**

**Objective:** Establish an automated Continuous Integration/Continuous Deployment (CI/CD) pipeline to support rapid deployment, testing, and rollback of telco cloud applications and services across hybrid environments.

**Platform:** Azure DevOps, GitHub Actions, Jenkins, and Kubernetes integrated with cloud-native deployment strategies.

**Components:**

1. **Source Code Repositories:**
   * Hosted in GitHub with protected branches and pull request policies.
   * Webhooks integrated with Azure DevOps and Jenkins.
2. **Build Automation:**
   * Jenkins and Azure DevOps pipelines triggered on code commit.
   * Docker images built from microservices and stored in Azure Container Registry (ACR).
3. **Testing Stages:**
   * Unit tests using JUnit and PyTest.
   * Integration tests simulated using Postman/Newman.
   * Load testing with Apache JMeter.
4. **Artifact Management:**
   * Binaries and containers stored in Azure Artifacts and ACR.
5. **Deployment Automation:**
   * Kubernetes manifests and Helm charts used to deploy to AKS and GKE clusters.
   * Terraform used for IaC provisioning of underlying environments.
   * Canary releases and Blue-Green deployments configured.
6. **Monitoring & Rollbacks:**
   * Integrated with Prometheus + Grafana for monitoring.
   * Rollbacks triggered automatically on failed health checks or regression tests.
7. **Security:**
   * Secrets management using Azure Key Vault and GitHub Secrets.
   * Pipeline access secured with RBAC and OAuth2.

**Workflow:**

Git Commit → Jenkins/Azure DevOps Trigger → Build Docker Image → Run Tests → Push to Registry → Deploy to AKS/GKE → Monitor → Auto-Rollback (if required)

**Documentation Artifacts:**

* CI/CD pipeline YAML files.
* Helm chart templates.
* Deployment and rollback policy document.
* Secure credential rotation procedure.