**Deployment Runbooks & Rollback Procedures**

**Title:** *Deployment & Rollback Execution Guide for Smart 5G Optimization*

**A. Purpose**

Ensure a consistent and repeatable deployment process across staging and production environments, with clearly defined rollback mechanisms to mitigate risk.

**B. Deployment Runbook Structure**

| **Step** | **Task** | **Responsible** | **Tools/Commands** | **Checkpoint** |
| --- | --- | --- | --- | --- |
| 1 | Pre-deployment validation (system health) | DevOps | Ansible, custom health-check scripts | All nodes online, disk/mem thresholds met |
| 2 | Pull latest code from Git (tagged release) | DevOps | git pull, GitHub Actions | Version X.X.X checked out |
| 3 | Infrastructure provisioning (if needed) | Cloud Admin | Terraform | Resources created successfully |
| 4 | Configuration deployment (CI/CD) | DevOps | Jenkins, Ansible playbooks | Configs deployed, no errors |
| 5 | Service restart and sanity checks | NOC Engineer | systemctl, curl, Grafana dashboards | Services reachable, no alarms triggered |

**C. Rollback Plan Outline**

| **Trigger** | **Rollback Action** | **Recovery Time Objective** |
| --- | --- | --- |
| Deployment failure (error logs) | Revert to previous release via tagged Git commit | < 15 minutes |
| Major performance degradation | Redeploy autosaved config snapshot + restart pods | < 30 minutes |
| Critical system outage | Initiate full restore from backup | < 1 hour |

Version tagging, snapshots, and automated rollback testing should be part of your CI pipeline.