Self-Study Guide: CI/CD Concepts

Continuous Integration (CI)

Definition: A Development practice where developers integrate code into a shared repository frequently, verified by automated builds and tests.

Key Benefits:

- Early bug detection
- Reduced integration problems
- Faster release cycles

Continuous Delivery (CD)

Definition: Automated deployment of code changes to testing/staging environments after the build stage.

Key Benefits:

- Always-ready-to-deploy code
- Reduced deployment risk
- Faster time to market

Continuous Deployment

Definition: Automatic deployment of every change that passes automated tests to production.

Key Benefits:

- Rapid feature delivery
- Reduced manual intervention
- Consistent deployment process

Jenkins Best Practices

1. Pipeline as Code: Define pipelines in Jenkinsfile

2. Agent Management: Use dedicated build agents
3. Security: Implement proper authentication and authorization
4. Backup: Regular backup of Jenkins configuration
5. Monitoring: Set up health checks and alerts

Troubleshooting Common Issues

Webhook Not Triggering

```bash
# Check Jenkins logs
sudo tail -f /var/log/jenkins/jenkins.log

# Verify GitHub webhook delivery
# In GitHub: Settings → Webhooks → Recent Deliveries
```

SSH Connection Failed

```bash
# Test SSH connection manually from Jenkins server
ssh -i /path/to/key.pem ec2-user@172.31.19.59

# Check security groups allow port 22
# Verify SSH key permissions
chmod 400 your-key.pem
```

File Permission Issues

```bash
# On NFS server, ensure proper permissions
sudo chown -R ec2-user:ec2-user /mnt/apps
sudo chmod -R 755 /mnt/apps
```