



DEVOPS-204

Continuous Integration & Continuous Delivery with Gitlab

Continuous Integration & Continuous Delivery with Gitlab

OVERVIEW

Skill Level : Professional

Suitable for : Members of Engineering and DevOps teams who want to use GitLab to produce software in short cycles through Continuous Integration and Continuous Delivery practices.

Duration : 2 Days

Continuous Integration (CI) is a vital technique in software development that streamlines the integration of code modifications from multiple team members into a unified project. A primary DevOps practice, it empowers developers to regularly combine code changes into a central repository, where automated builds and tests are executed. Automated tools are used to ensure the integrity of new code is thoroughly tested before it can enter a main production system.

Git — a source code version control system — is the heart of the CI process. Additional checks such as automated code quality tests, syntax style review tools, and various other measures ensure the robustness and efficiency of the codebase.

Continuous Delivery (CD) is a popular software development practice in DevOps which uses automation to speed the release of new code through a series of processes for software development, deployment, and feedback loops.

In this course, the trainee will learn how to use GitLab to develop software through both Continuous Integration and Continuous Delivery (CI/CD) practices. The course will help the trainee develop a working knowledge of proper git workflow, as well as the different stages that need to happen in the pipeline of each branch.

PREREQUISITES

- DEVOPS-102: Linux Fundamentals (or equivalent)
- DEVOPS-103: Git Fundamentals (or equivalent)
- DEVOPS-104: Continuous Integration Basics (or equivalent)

COURSE OUTLINE

What is CI/CD?

- Concept
- Benefits
- Overall Process

Git Workflow

- Branches
 - Master/main
 - Develop
 - Feature
 - UAT
 - Release
 - Hotfixes
- Merge Requests & Code Review
- Linking branches / merge requests to issues via Git commit messages

Deployment Pipelines

- How to set up a pipeline
- Code Quality Check
- Unit Testing
- Integration Testing
- Functional Testing
- Static Application Security Testing (SAST)
- Dynamic Application Security Testing (DAST)
- Load Testing
- Fuzz Testing
- Build (w/ Buildpacks)
- Push
- Deploy

Best Practices

- 12 Factor App pattern
- Deployment Environments
 - Dev
 - UAT
 - Production
- Secrets Management
- Github Package Registry
- Release Versioning

Release Deployment Strategies & Rollback

- Shipping to End-Users
- Recreate
- Ramped
- Blue/Green
- Canary
- A/B Testing
- Shadow
- Rollback



Enquiries



+63 2 5322 2307



training-sales@orangeandbronze.com