

# Study Guide: CI/CD, Git, YAML, and Advanced Deployment Strategies

## Section 1: Git Basics

1. `git pull` and `git pull --rebase`:
  - Understand the purpose of these commands.
  - Know when to use `--rebase` and its implications.
2. **Unstaging Changes:**
  - Learn how to unstage changes using the appropriate Git command.
  - Be able to describe common scenarios where this is useful.
3. `git commit --amend`:
  - Explain the function of `git commit --amend` and when to use it.
  - Understand its impact on commit history.

## Section 2: Git Merge Strategies

1. **Rebase Operation:**
  - Describe what happens to commit history during a rebase operation.
  - Differentiate between rebase and merge.
2. **Interactive Rebase:**
  - Explain the purpose of interactive rebasing.
  - Describe scenarios where interactive rebase is valuable.

## Section 3: YAML

1. **YAML Anchors:**
  - Define YAML anchors and how they are used in YAML documents.
  - Recognize examples of YAML anchors in configuration files.
2. **Multi-line Strings:**
  - Understand how multi-line strings are represented in YAML.
  - Learn the common indicators for multi-line strings.

## Section 4: CI/CD Basics

1. **Continuous Integration (CI):**
  - Define CI and its importance in the software development process.
  - Explain how CI contributes to code quality.
2. **Continuous Delivery (CD):**
  - Clarify the role of CD in software development.
  - Discuss the difference between CI and CD.
3. **Immutable Infrastructure:**
  - Define “Immutable Infrastructure” in the context of CI/CD.
  - Explain why it’s relevant in modern software deployment.

## Section 5: Git Commits

1. **Clean Commits:**
  - Understand why clean commits are important.
  - Describe the characteristics of a clean commit message.

## Section 6: CircleCI Basics

1. **CircleCI Introduction:**

- Define CircleCI and its role in CI/CD.
- Enumerate the benefits of using CircleCI.

## 2. CircleCI Job and Workflow:

- Differentiate between a job and a workflow in CircleCI.
- Explain their significance in defining CI/CD pipelines.

## 3. Context and Environment:

- Distinguish between “context” and “environment” in CircleCI.
- Understand how they are used to manage secrets and configuration.

## Section 7: Canary Deploys

### 1. Canary Deployments:

- Define what Canary deployments are and why they exist.
- Understand their primary objectives in the software release process.

### 2. Control Group:

- Define the concept of a control group in Canary deployments.
- Clarify the role of the control group in monitoring and testing.

## Test Preparation Tips:

- Review Git commands and operations, including pull, merge, and rebase.
- Practice using Git commands to manipulate commit history.
- Study the basics of YAML syntax, including anchors and multi-line strings.
- Understand the core concepts of CI/CD and why they are essential in modern software development.
- Explore Git commit best practices and clean commit message formats.
- Learn about CircleCI, its key components, and how to configure CI/CD pipelines.
- Familiarize yourself with Canary deployments, feature flags, and risk management strategies.

Use this study guide to review the key concepts and topics covered in the test, and don't forget to practice with real-world examples and scenarios to solidify your understanding. Good luck with your test preparation!