****This study guide is based on the video lesson available on TrainerTests.com****

Continuous Delivery and Deployment Study Guide

This chapter explores the concepts of continuous delivery and continuous deployment, two DevOps practices that extend continuous integration (CI).

Continuous Delivery

- **Definition:** Continuous delivery is an automated software development practice that builds, tests, and deploys code changes to a testing or staging environment after every code commit.
- **Relationship to CI:** Continuous delivery builds upon continuous integration. After code is committed to a central repository and goes through the CI process (build, test, report), continuous delivery automates the deployment process to a testing or staging environment.
- **Benefits:**
    - **Faster Feedback:** Developers receive quicker feedback on the impact of their changes through deployment to a testing or staging environment.
    - **Reduced Risk:** Deploying smaller batches of code to a testing environment allows for easier identification and resolution of issues before they reach production.
    - **Improved Collaboration:** Continuous delivery encourages a collaborative development environment where developers can integrate their changes frequently and observe the impact in a testing environment.
- **Process:**
    1. Code is committed to a central repository.
    2. The CI process builds, tests, and reports on the code.
    3. The continuous delivery pipeline automatically deploys the code to a testing or staging environment.
    4. Additional manual testing or approvals may occur before deploying the code to production.

Continuous Deployment

- **Definition:** Continuous deployment is an automated software development practice that builds, tests, and deploys code changes directly to production after every code commit, provided all automated tests pass.
- **Relationship to CI:** Similar to continuous delivery, continuous deployment builds upon continuous integration. However, with continuous deployment, successful completion of the CI pipeline automatically triggers deployment to production.

- **Benefits:**
  - **Faster Releases:** New features and bug fixes can be deployed to production as soon as they are developed and tested, leading to faster time to market.
  - **Improved Reliability:** Frequent deployments with automated testing can help to improve the overall reliability of the software.
- **Process:**
  1. Code is committed to a central repository.
  2. The CI process builds, tests, and reports on the code.
  3. If all automated tests pass, the continuous deployment pipeline automatically deploys the code directly to production.
  4. There is no manual approval step required before deployment to production.

Key Differences Between Continuous Delivery and Deployment

- **Deployment Target:** Continuous delivery deploys code to a testing or staging environment, while continuous deployment deploys code directly to production.
- **Approval Process:** Continuous delivery may involve manual approval before deploying to production, whereas continuous deployment is fully automated.
- **Risk Tolerance:** Continuous delivery offers a more cautious approach with a lower risk of deploying buggy code to production. Continuous deployment requires a high degree of confidence in automated testing and a tolerance for potential issues that may arise in production.

In conclusion, continuous delivery and continuous deployment are DevOps practices that extend continuous integration by automating the deployment process. They offer different benefits and risks, and the choice of approach depends on the specific needs and risk tolerance of the development team and project.

*See slides below:

Continuous Delivery

Continuous delivery is an extension of continuous integration since it automatically deploys all code changes to a testing and/or production environment after the build stage.

Code Repository

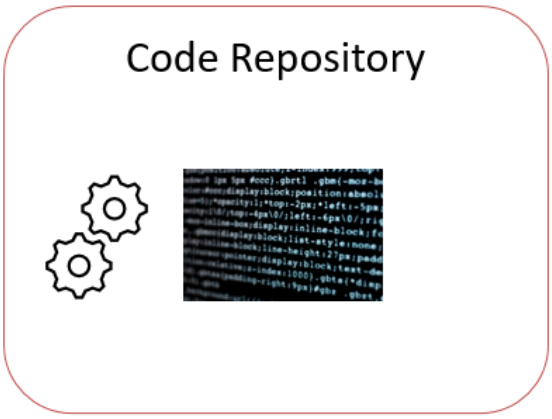
Continuous Delivery

Continuous delivery is an extension of continuous integration since it automatically deploys all code changes to a testing and/or production environment after the build stage.

Code Repository

Continuous Delivery

Continuous delivery is an extension of continuous integration since it automatically deploys all code changes to a testing and/or production environment after the build stage.

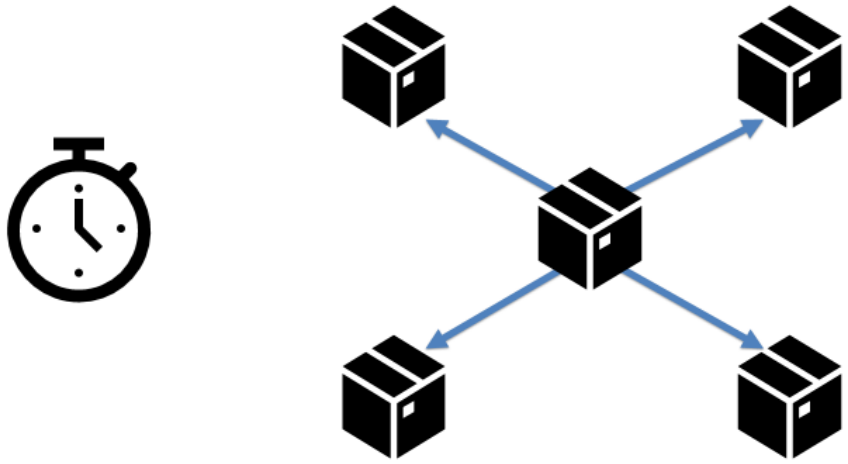


Continuous Delivery

Continuous delivery is an extension of continuous integration since it automatically deploys all code changes to a testing and/or production environment after the build stage.

Continuous Delivery

Continuous delivery is an extension of continuous integration since it automatically deploys all code changes to a testing and/or production environment after the build stage.



Continuous Deployment

Continuous deployment goes one step further than continuous delivery. With this practice, every change that passes all stages of your production pipeline is released to your customers.