

Continuous Integration & Continuous Delivery

- Ayushi Arora

What will be Covered

- What is DevOps
- DevOps Practices
- What is Continuous Integration
- What is Continuous Delivery
- What is Continuous Deployment
- Importance of CI/CD
- CI/CD Tools
- Jenkins
- CI/CD Pipeline
- Pipeline Concepts
- CI/CD Implementation
- Exercises



DevOps -

DevOps is a software development practice that promotes collaboration between development and operations, resulting in faster and more reliable software delivery. Commonly referred to as a culture, DevOps connects people, process, and technology to deliver continuous value.

It involves Continuous Development , continuous testing , continuous integration , continuous deployment and continuous monitoring throughout its development life cycle.



DevOps Practices -

- Continuous Integration
- Continuous Delivery
- Microservices
- Infrastructure as Code
- Monitoring and Logging
- Communication and Collaboration



What is Continuous Integration - CI

Continuous integration (CI) is the practice of automating the integration of code changes from multiple contributors into a single software project.

This allows developers to frequently merge code changes into a central repository where builds and tests then run automatically.



What is Continuous Delivery - CD

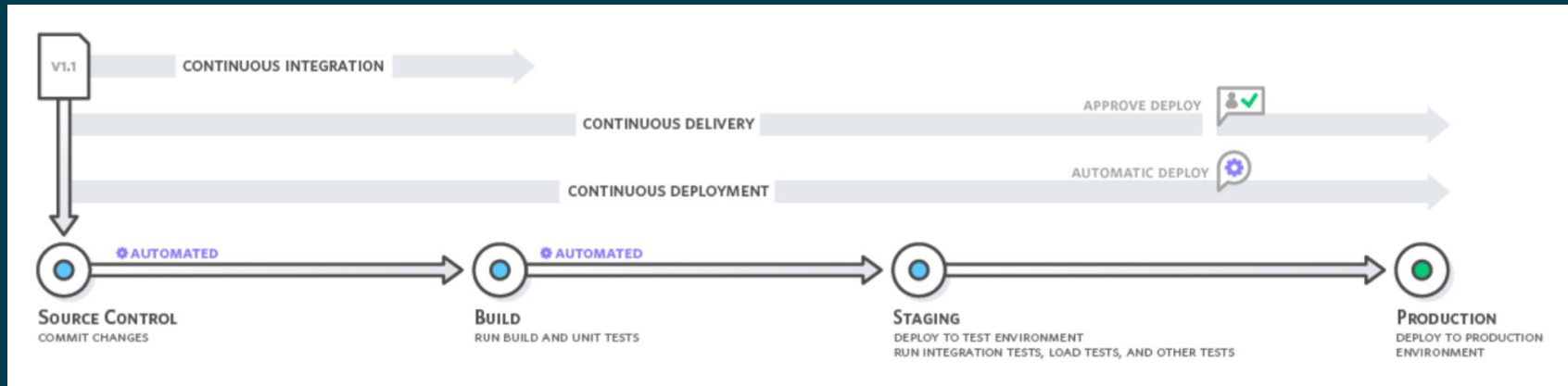
Continuous Delivery is an extension of continuous integration since it deploys all code changes to a testing and/or staging environment after the build stage. With continuous delivery, we can decide to release daily, weekly, fortnightly, or whatever suits our business requirements.

However, to get the benefits of continuous delivery, deploy as early as possible because small batches of release are easy to troubleshoot in case of a problem.



What is 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. There's no human intervention, and only a failed test will prevent a new change to be deployed to production.



Importance of CI/CD -

- It reduces the number of repeated processes across all project operations from the development to the deployment stage and receiving feedback on the software.
- Its core importance is to ensure that software being developed is deployable to end-users but with better quality and lower risks of failure.
- Successful implementation of CI practices can boost confidence in software development, such that the team will know that tests are passed, and the result is a functioning testable product with each build.

CI/CD Tools -

- Jenkins
- Terraform
- Azure devOps
- GitLab CI
- GOCD
- CircleCI
- CodeShip
- TeamCity
- Travis CI
- CodeFresh

.....many more



Jenkins -

Jenkins is an open-source Continuous Integration tool written in Java for orchestrating a chain of actions to achieve the Continuous Integration process in an automated fashion. It supports the complete development life cycle of software from building, testing, documenting the software, deploying, and other stages of the software development life cycle.



Installation Steps Link -

<https://coralogix.com/blog/how-to-install-and-configure-jenkins-on-the-mac-os/>



CI/CD Pipeline -

A pipeline is a set of steps that team implement to deliver software faster and easier. One of the primary purposes of a pipeline is to keep the software development process organized and focused.

```
pipeline {  
  agent any ❶  
  stages {  
    stage('Build') { ❷  
      steps {  
        // ❸  
      }  
    }  
    stage('Test') { ❹  
      steps {  
        // ❺  
      }  
    }  
    stage('Deploy') { ❻  
      steps {  
        // ❼  
      }  
    }  
  }  
}
```

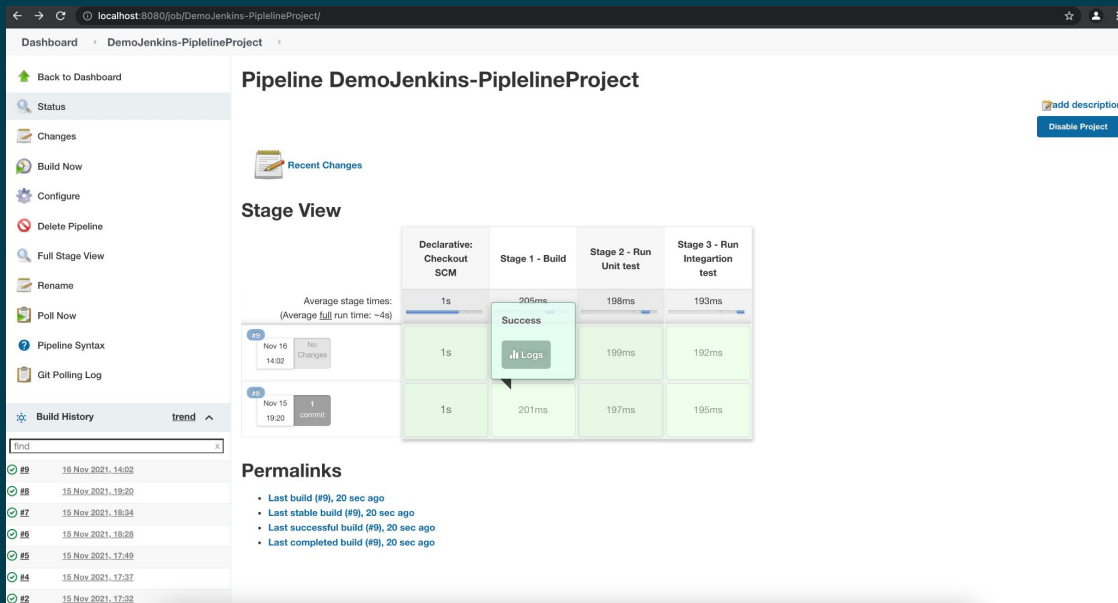
Pipeline concepts -

- Pipeline - A Pipeline's code defines your entire process, which typically includes stages for building an application, testing it and then delivering it. The pipeline block defines the entire process.
- Agent - Execute this Pipeline or any of its stages, on any available agent.
- Stage - A stage block defines a distinct subset of tasks performed through the entire Pipeline (e.g. "Build", "Test" and "Deploy" stages)
- Step - A single task, it tells Jenkins what to do at a particular point in time (or "step" in the process).



Implementation CI/CD -

- 1- Create a CI/CD pipeline which triggers build periodically (Cron Job)
- 2- Create a CI/CD pipeline which trigger builds with Poll SCM in Jenkins



Exercises -

- 1- Add a stage to the existing Pipeline script.
- 2- Create a Pipeline in Jenkins using Pipeline script which triggers in every 2 minutes.

Thank You

