Pipeline as Code

Section 4

What you will learn in this section

Theory:

- Pipeline as Code
- Jenkinsfile
- Pipeline Plugin

Practice:

- Implement a CD pipeline

What is Pipeline as Code

Video 4.1

What you will learn in this video

- What is Pipeline as Code
- How pipelines are implemented in Jenkins 2
- Benefits of using pipelines

Pipeline as Code

- A set of features allowing to **define and combine job processes with code stored under version control**

Pipeline as Code

- A set of features allowing to **define and combine job processes with code stored under version control**

- Jenkins **jobs and configuration defined as code**

Pipeline as Code

- A set of features allowing to **define and combine job processes with code stored under version control**
- Jenkins **jobs and configuration defined as code**
- Discover, manage, and run jobs for **multiple repositories and branches**

Pipeline as Code

- A set of features allowing to **define and combine job processes with code stored under version control**
- Jenkins **jobs and configuration defined as code**
- Discover, manage, and run jobs for **multiple repositories and branches**
- **Reduce manual creation and management** of jobs

Infrastructure as Code

- **Manage and provision IT infrastructure through code** rather than manual configuration

Infrastructure as Code

- **Manage and provision IT infrastructure through code** rather than manual configuration
- **Automate** IT processes

Infrastructure as Code

- **Manage and provision IT infrastructure through code** rather than manual configuration
- **Automate** IT processes
- **Repeatable** configuration

Types of Pipelines

- CI using multiple branches and repositories

Types of Pipelines

- CI using multiple branches and repositories
- CD to multiple environments

Types of Pipelines

- CI using multiple branches and repositories
- CD to multiple environments
- Data processing (ETL)

Types of Pipelines

- CI using multiple branches and repositories
- CD to multiple environments
- Data processing (ETL)
- Event-driven workflows

Types of Pipelines

- CI using multiple branches and repositories
- CD to multiple environments
- Data processing (ETL)
- Event-driven workflows
- Build promotion

Types of Pipelines

- CI using multiple branches and repositories
- CD to multiple environments
- Data processing (ETL)
- Event-driven workflows
- Build promotion
- Conditional processing

Pipeline Flow

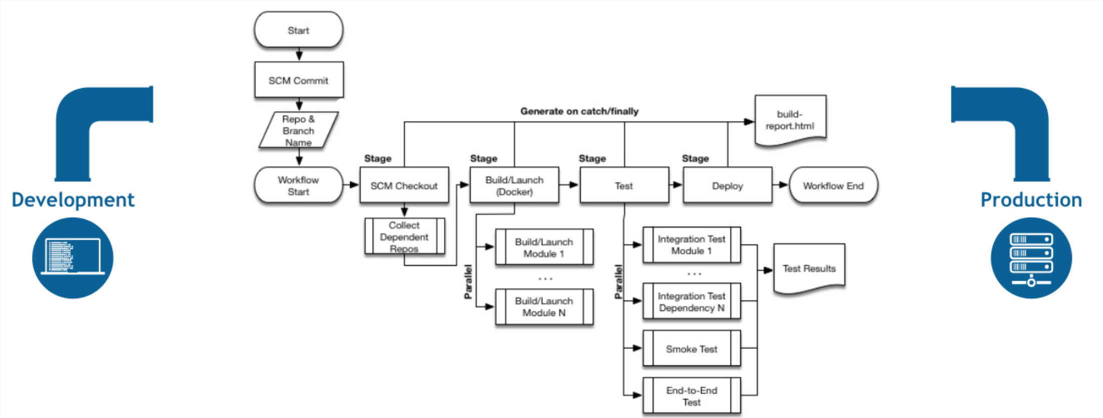


Image: https://jenkins.io/doc/book/pipeline ©

How pipelines are implemented in Jenkins 2

- Pipeline Plugin

How pipelines are implemented in Jenkins 2

- Pipeline Plugin

- Jenkinsfile

How pipelines are implemented in Jenkins 2

- Pipeline Plugin
- Jenkinsfile
- Domain-specific language (DSL) using Groovy

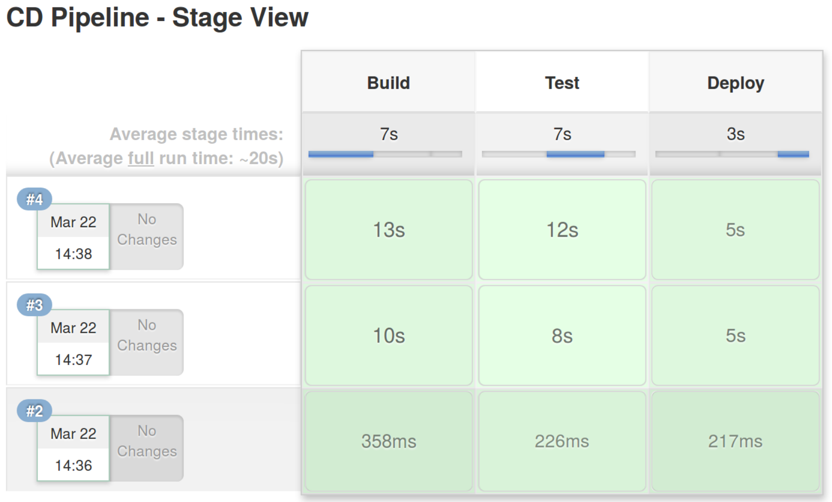
How pipelines are implemented in Jenkins 2

- Pipeline Plugin
- Jenkinsfile
- Domain-specific language (DSL) using Groovy
- Pipeline shared libraries

How pipelines are implemented in Jenkins 2

- Pipeline Plugin
- Jenkinsfile
- Domain-specific language (DSL) using Groovy
- Pipeline shared libraries
- Stage views

Pipeline Visualization



CD Pipeline - Stage View

	Build	Test	Deploy
Average stage times: (Average full run time: ~20s)	7s	7s	3s
#4 Mar 22 14:38 No Changes	13s	12s	5s
#3 Mar 22 14:37 No Changes	10s	8s	5s
#2 Mar 22 14:36 No Changes	358ms	226ms	217ms

Benefits of Pipelines

- **Code**: Pipelines are implemented in code and checked into source control

Benefits of Pipelines

- **Code**: Pipelines are implemented in code and checked into source control
- **Resilient**: Pipelines can survive Jenkins restarts

Benefits of Pipelines

- **Code**: Pipelines are implemented in code and checked into source control
- **Resilient**: Pipelines can survive Jenkins restarts
- **Pausable**: Pipelines can stop and wait for manual approval

Benefits of Pipelines

- **Code**: Pipelines are implemented in code and checked into source control
- **Resilient**: Pipelines can survive Jenkins restarts
- **Pausable**: Pipelines can stop and wait for manual approval
- **Versatile**: Pipelines support complex requirements (loops, conditionals, matrix, parallelisation)

Benefits of Pipelines

- **Code**: Pipelines are implemented in code and checked into source control
- **Resilient**: Pipelines can survive Jenkins restarts
- **Pausable**: Pipelines can stop and wait for manual approval
- **Versatile**: Pipelines support complex requirements (loops, conditionals, matrix, parallelisation)
- **Extensible**: DSL extensions and shared library can be integrated

Benefits of Pipelines

- **Code**: Pipelines are implemented in code and checked into source control
- **Resilient**: Pipelines can survive Jenkins restarts
- **Pausable**: Pipelines can stop and wait for manual approval
- **Versatile**: Pipelines support complex requirements (loops, conditionals, matrix, parallelisation)
- **Extensible**: DSL extensions and shared library can be integrated
- **Visual**: Pipeline can be visualized thus improving visibility to non-technical people

Next Video

Video 4.2 Jenkinsfile