

What is Jenkins Pipeline?

Jenkins Pipeline (or simply "Pipeline" with a capital "P") is a suite of plugins which supports implementing and integrating *continuous delivery pipelines* into Jenkins.

A *continuous delivery pipeline* is an automated expression of your process for getting software from version control right through to your users and customers. Every change to your software (committed in source control) goes through a complex process on its way to being released. This process involves building the software in a reliable and repeatable manner, as well as the progression of the built software (called a "build") through multiple stages of testing and deployment.

Pipeline provides an extensible set of tools for modeling simple-to-complex delivery pipelines "as code" via the [Pipeline Domain Specific Language \(DSL\) syntax](#). [1: [Domain-Specific Language](#)]

Typically, the definition of a Jenkins Pipeline is written into a text file (called a **Jenkinsfile**) which in turn is checked into a project's source control repository. [2: [Source Control Management](#)] This is the foundation of "Pipeline-as-Code"; treating the continuous delivery pipeline a part of the application to be versioned and reviewed like any other code. Creating a **Jenkinsfile** provides a number of immediate benefits:

- Automatically create Pipelines for all Branches and Pull Requests
- Code review/iteration on the Pipeline
- Audit trail for the Pipeline
- Single source of truth [3: en.wikipedia.org/wiki/Single_Source_of_Truth] for the Pipeline, which can be viewed and edited by multiple members of the project.

While the syntax for defining a Pipeline, either in the web UI or with a **Jenkinsfile**, is the same, it's generally considered best practice to define the Pipeline in a **Jenkinsfile** and check that in to source control.

Here's an example of a **Jenkinsfile**: