Jenkins Pipeline

1. 认识 Pipeline

1.1 Pipeline 是什么？

Pipeline 是 Jenkins 的核心功能，提供一组可扩展的工具。通过 Pipeline 的 DSL 语法可以完成从简单到复杂的交付流水线实现。jenkins 的 Pipeline 是通过 Jenkinsfile（文本文件）来实现的。这个文件可以定义 Jenkins 的执行步骤，例如检出代码。

1.2 Jenkinsfile

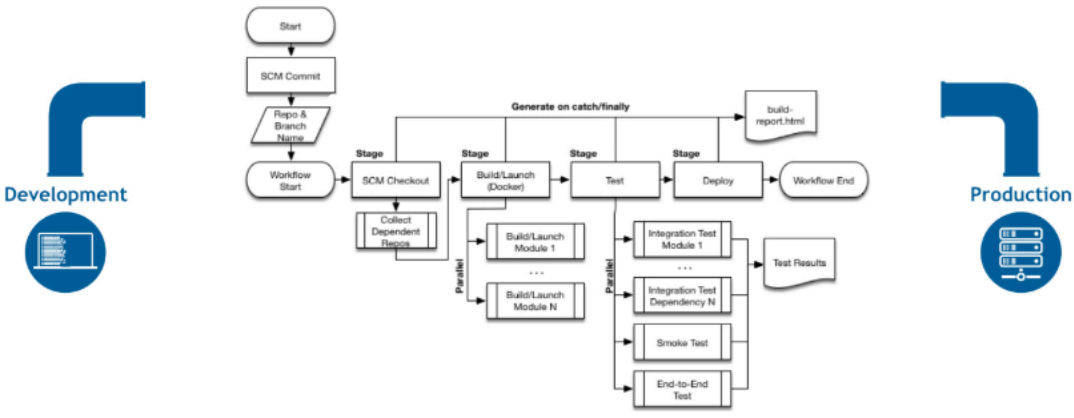
Jenkinsfile 使用两种语法进行编写，分别是声明式和脚本式。声明式和脚本式的流水线从根本上是不同的，声明式是 jenkins 流水线更友好的特性 :相比脚本式的流水线语法，提供更丰富的语法特性。声明式流水线使编写和读取流水线代码更容易设计。

1.3 为什么使用 Pipeline？

本质上，jenkins 是一个自动化引擎，它支持许多自动模式。流水线向 Jenkins 添加了一组强大的工具，支持用例、简单的持续集成到全面的持续交付流水线。通过对一系列的发布任务建立标准的模板，用户可以利用更多流水线的特性，比如：

代码化: 流水线是在代码中实现的，通常会存放到源代码控制，使团队具有编辑、审查和更新他们项目的交付流水线的能力。
耐用性：流水线可以从 Jenkins 的 master 节点重启后继续运行。
可暂停的：流水线可以由人功输入或批准继续执行流水线。
解决复杂发布： 支持复杂的交付流程。例如循环、并行执行。
可扩展性： 支持扩展 DSL 和其他插件集成。

构建一个可扩展是 Jenkins 的核心价值，流水线可以通过 ShareLibrary 的方式来扩展。下面是一个 CD 的场景实例



2. Pipeline 概念

参考文档：https://jenkins.io/zh/doc/book/pipeline/

2.1 node（节点）

节点是一个机器，可以是 Jenkins 的 master 节点也可以是 slave 节点。通过 node 指定当前 job 运行的机器（这个是脚本式语法）

2.2 Stage（阶段）

stage 定义了在整个流水线的执行任务的概念性的不同的阶段。

2.3 step（步骤）

step 是每个阶段中要执行的每个步骤。

3. 流水线语法

3.1 声明式流水线

在声明式流水线语法中，`pipeline` 块定义了整个流水线中完成的所有的工作。

Jenkinsfile (Declarative Pipeline)

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

❶ 在任何可用的代理上，执行流水线或它的任何阶段。
❷ 定义 "Build" 阶段。
❸ 执行与 "Build" 阶段相关的步骤。
❹ 定义"Test" 阶。
❺ 执行与"Test" 阶段相关的步骤。
❻ 定义 "Deploy" 阶段。
❼ 执行与 "Deploy" 阶段相关的步骤。

3.2 脚本式流水线

在脚本化流水线语法中，一个或多个 `node` 块在整个流水线中执行核心工作。 虽然这不是脚本化流水线语法的强制性要求，但它限制了你的流水线的在`node`块内的工作做两件事：

1. 通过在Jenkins队列中添加一个项来调度块中包含的步骤。 节点上的执行器一空闲，该步骤就会运行。
2. 创建一个工作区(特定为特定流水间建立的目录)，其中工作可以在从源代码控制检出的文件上完成。
   Caution: 根据你的 Jenkins 配置,在一系列的空闲后，一些工作区可能不会自动清理 。参考 JENKINS-2111 了解更多信息。

Jenkinsfile (Scripted Pipeline)

```
node {                  (1)
    stage('Build') {    (2)
        //              (3)
    }
    stage('Test') {     (4)
        //              (5)
    }
    stage('Deploy') {   (6)
        //              (7)
    }
}
```

(1) 在任何可用的代理上，执行流水线或它的任何阶段。
(2) 定义 "Build" 阶段。 `stage` blocks 在脚本化流水线语法中是可选的。 然而，在脚本化流水线中实现`stage` 块 ， 可以清楚的显示Jenkins UI中的每个`stage`的任务子集。
(3) 执行与 "Build" 阶段相关的步骤。
(4) 定义 "Test" 阶段。
(5) 执行与 "Test" 阶段相关的步骤。
(6) 定义 "Deploy" 阶段。
(7) 执行与 "Deploy" 阶段相关的步骤。

4. 定义 Jenkinsfile

4.1 BlueOcean

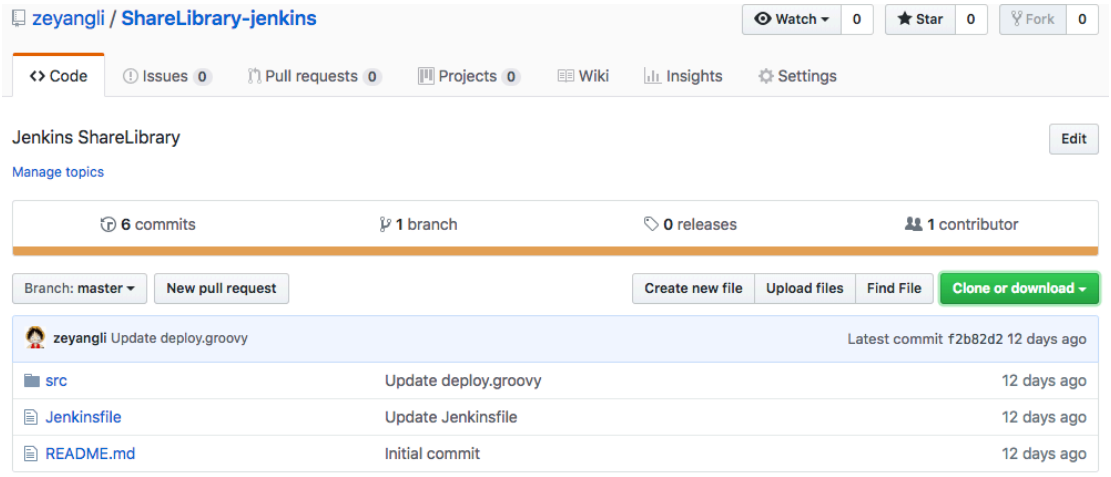
如果你是新手，Blue Ocean 可以帮助你设置流水线，通过图形化流水线编辑器自动创建和编写 Jenkinsfile。（需要安装 blueocean 插件）

4.2 WebUI

项目->配置

4.3 SCM

Jenkinsfile 编写完成后上传到 gitlab 进行版本控制。



再通过项目的配置 gitlab 仓库地址和分支信息以及文件名称信息。