一、Pipeline流水线

1、Pipeline&Jenkinsfile定义

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。

一条流水线通过Jenkinsfile来描述，其构成包括：

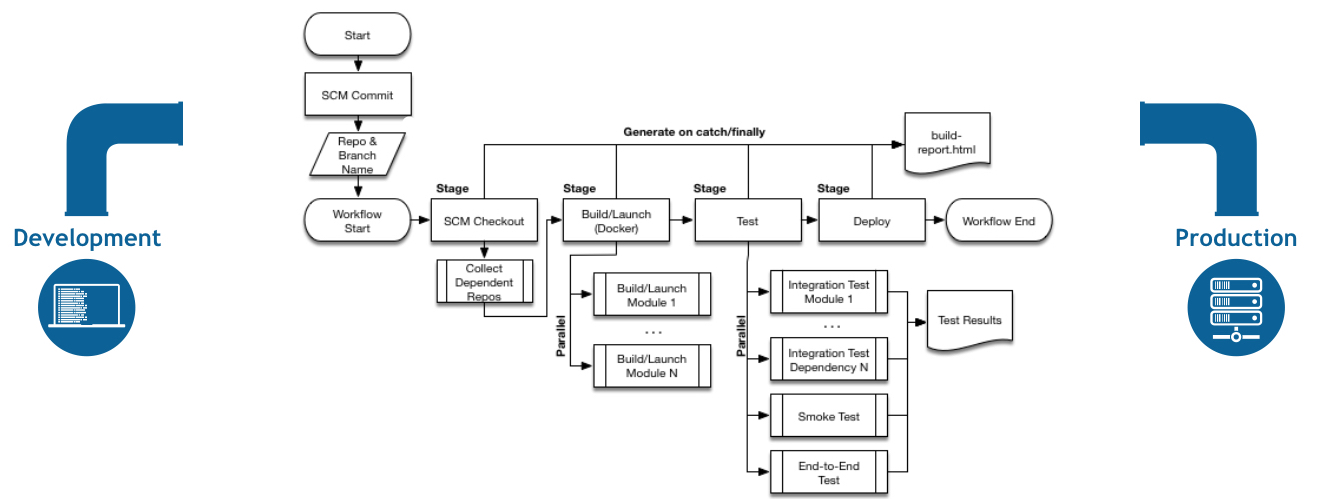
1、指定node节点/workspace

2、指定运行选项

3、指定stages阶段

4、指定构建后操作

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，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



2、Pipeline语法

2.1、node节点/workspace

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。

例如：

agent { node { label "master" //指定运行节点的标签

customWorkspace "${workspace}" //指定运行工作目录(可选)

}

}

node参数包括：

- any：在任何可用的节点上执行pipeline

- none：没有执行agent的时候默认

- label：在指定标签上的节点运行pipeline

2.2、options运行选项

options 指令允许从流水线内部配置特定于流水线的选项。 流水线提供了许多这样的选项, 比如buildDiscarder,但也可以由插件提供, 比如 timestamps。

例如：

options {

timestamps() //日志时间戳

skipDefaultCheckout() //删除隐式checkout scm语句

disableConcurrentBuilds() //禁止并行

timeout(time: 1, unit:'HOURS') //流水线超时1h

}

- buildDiscarder: 为最近的流水线运行的特定数量保存组件和控制台输出。

- disableConcurrentBuilds: 不允许同时执行流水线。 可被用来防止同时访问共享资源等。

- overrideIndexTriggers: 允许覆盖分支索引触发器的默认处理。

- skipDefaultCheckout: 在agent 指令中，跳过从源代码控制中检出代码的默认情况。

- skipStagesAfterUnstable: 一旦构建状态变得UNSTABLE，跳过该阶段。

- checkoutToSubdirectory: 在工作空间的子目录中自动地执行源代码控制检出。

- timeout: 设置流水线运行的超时时间, 在此之后，Jenkins将中止流水线。

- retry: 在失败时, 重新尝试整个流水线的指定次数。

- timestamps 预测所有由流水线生成的控制台输出，与该流水线发出的时间一致。

2.3、stages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。例如： GetCode、Build、Test、Deploy、CodeScan每个阶段。

例如：

stages {

//下载代码

stage("GetCode"){ //阶段名称

steps{ //步骤

timeout(time: 5, unit:'MINUTES'){ //步骤超时时间

script{ //运行代码

println('获取代码')

}

}

}

}

//构建

stage("Build"){

steps{ //步骤

timeout(time: 20, unit:'MINUTES'){ //步骤超时时间

script{ //运行代码

println('应用打包')

}

}

}

}

//代码扫描

stage("CodeScan"){

steps{ //步骤

timeout(time: 30, unit:'MINUTES'){ //步骤超时时间

script{ //运行代码

println('代码扫描')

}

}

}

}

}

2.4、post构建后操作

定义一个或多个steps ，这些阶段根据流水线或阶段的完成情况而 运行(取决于流水线中post部分的位置). post 支持以下 post-condition 块中的其中之一：always、changed、failure、 success、unstable和aborted。这些条件块允许在 post 部分的步骤的执行取决于流水线或阶段的完成状态。

- always：无论流水线或者阶段的完成状态。

- changed：只有当流水线或者阶段完成状态与之前不同时。

- failure：只有当流水线或者阶段状态为”failure”运行。

- success：只有当流水线或者阶段状态为”success”运行。

- unstable：只有当流水线或者阶段状态为”unstable”运行。例如：测试失败。

- aborted：只有当流水线或者阶段状态为”aborted “运行。例如：手动取消。

例如：

post {

always { //总是执行

script{

println("always")

}

}

success { //成功后执行

script{

currentBuild.description = "\n 构建成功"

}

}

failure { //失败后执行

script{

currentBuild.description = "\n 构建失败"

}

}

aborted { //取消后执行

script{

currentBuild.description = "\n 构建取消"

}

}

}