

【SOP】关于Jenkins2 Pipeline的使用以及Jenkins集成Jira

作者

王凯凯

概述

适用范围：适用于使用jenkins进行集成构建的项目

目的：项目集成构建

Jenkins的基本概念：

1. master,

也就是jenkins的server，是jenkins的核心，主要负责job的定时运行，将job分发到agent运行，和对job运行状态的监控。

2. agent/slave/node, agent是相对于master的概念，主要作用是监听master的指令，然后运行job。

3. executor, executor是虚拟的概念，每一个agent都可以设置executor的数量，表示可以同时运行的job的数量。

jenkins1的基本用法：

一般使用free

style的job类型，在job的里面调用一些脚本和插件来实现整个持续集成的过程，一般是将整个job发布到某台机器上运行。缺点是job的配置分布在脚本和插件中，配置不够集中，导致监控的时候log不容易查找。一般要实现整个持续集成发布流程需要多个

jobs来实现。

相对于jenkins1，jenkins2的核心概念是pipeline。在jenkins2中引入groovy脚本和虚拟的概念stage使得jenkins的配置和监控更集中，更清楚。

jenkins2的一些新概念：

1. step, 其实跟jenkins1中的概念一样，是jenkins里job中的最小单位，可以认为是一个脚本的调用和一个插件的调用。

2. node, 是pipeline里groovy的一个概念，node可以给定参数用来选择agent，node里的steps将会运行在node选择的agent上。这里与jenkins1的区别是，job里可以有多个node，将job的steps按照需求运行在不同的机器上。例如一个job里有好几个测试集合需要同

时运行在不同的机器上。

3. stage, 是pipeline里groovy里引入的一个虚拟的概念，是一些step的集合，通过stage我们可以将job的所有steps划分为不同的stage，使得整个job像管道一样更容易维护。pipeline还有针对stage改进过的view，使得监控更清楚。

pipeline的设计理念是实现基于groovy脚本，灵活，可扩展的持续发布（CD）工作流。

pipeline的功能和优点：

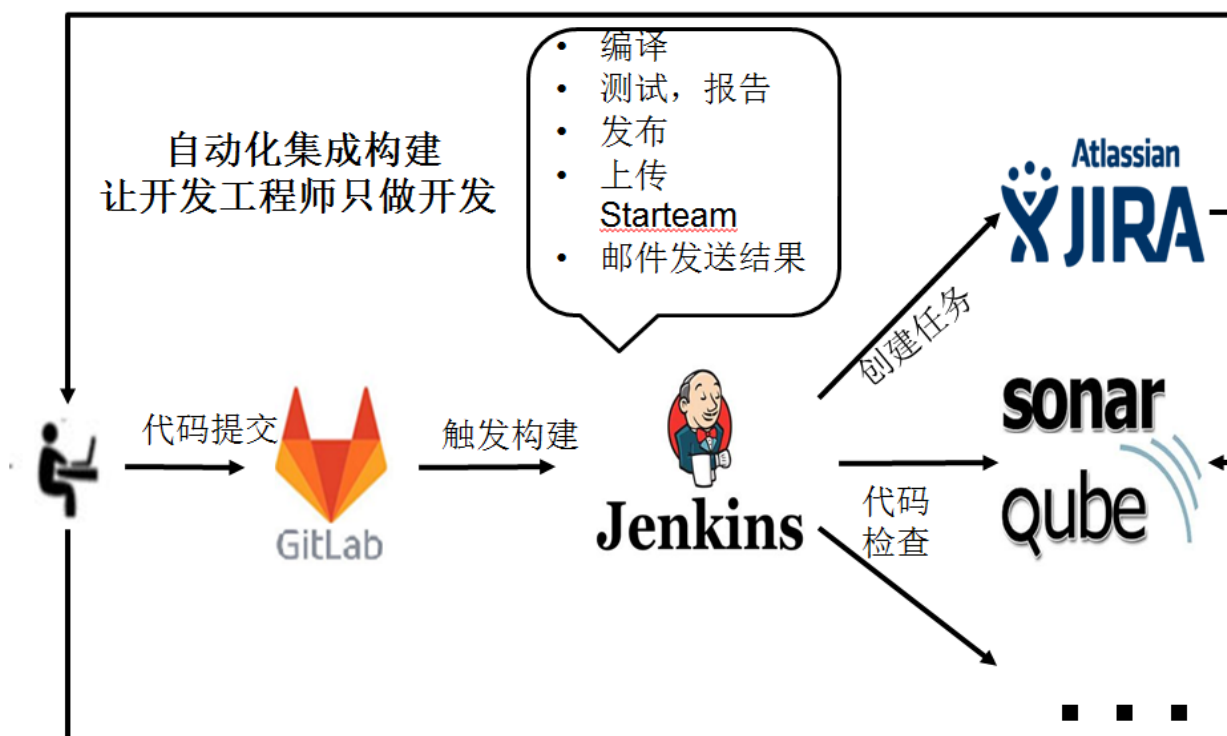
1. durable持久性：在jenkins的master按计划和非计划的重启后，pipeline的job仍然能够工作，不受影响。其实理解起来也很简单，jenkins的master和agent通过ssh连接，如果你知道nohup或disown的话，

就可以理解为啥master的重启不会影响agent上的job继续运行。

2. 可暂停性：pipeline基于groovy可以实现job的暂停和等待用户的输入或批准然后继续执行。

3. 更灵活的并行执行，更强的依赖控制，通过groovy脚本可以实现step，stage间的并行执行，和更复杂的相互依赖关系。

4. 可扩展性：通过groovy的编程更容易的扩展插件



前提

了解jenkins2的基本概念和使用场景

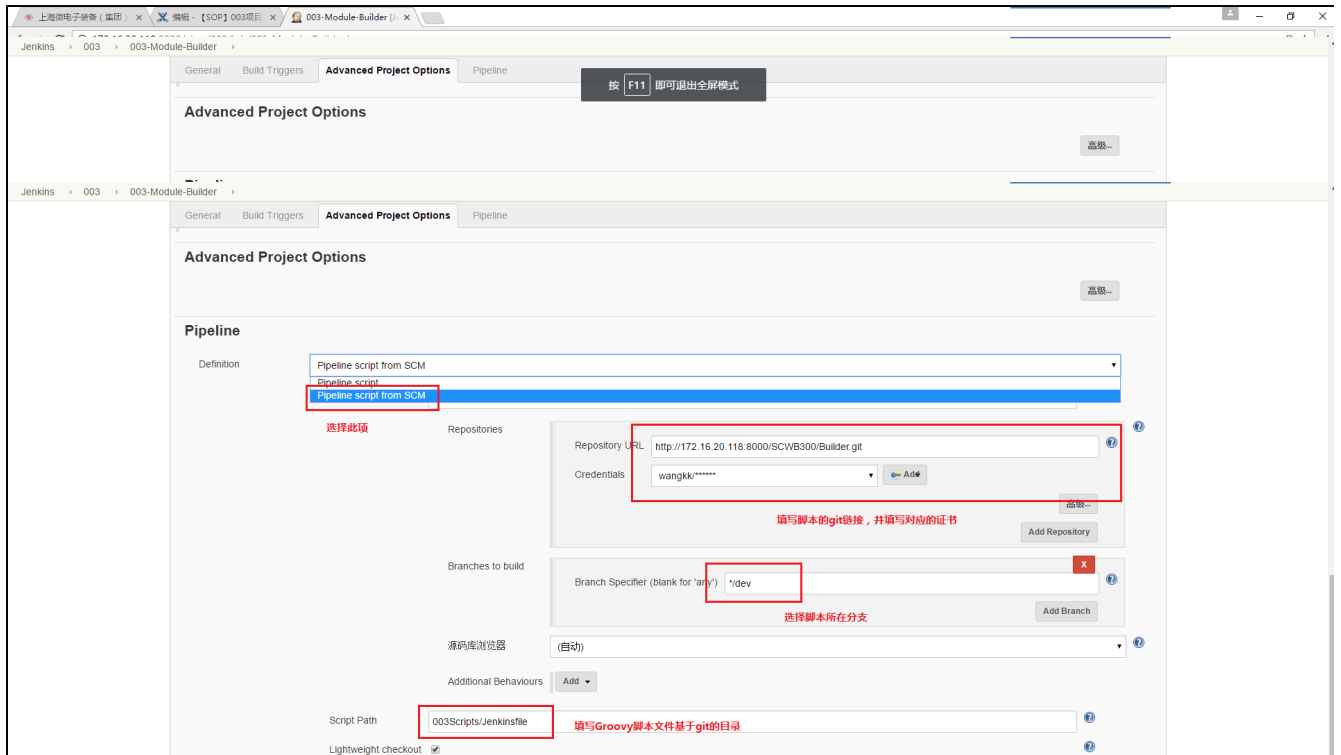
学习使用多种脚本语言：Python，Groovy，shell，bat等等

步骤

【步骤一】

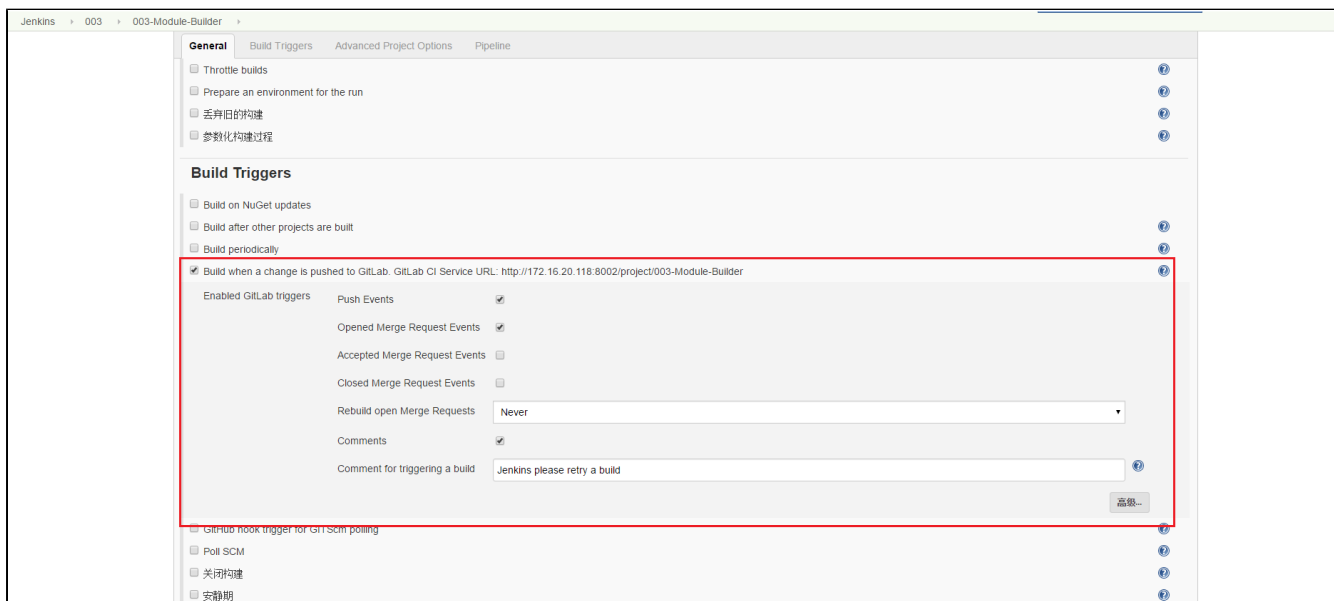
- 打开Jenkins对应的Job，选择配置
- 找到pipeline设置

【步骤二】配置Jenkins Job



【步骤三】

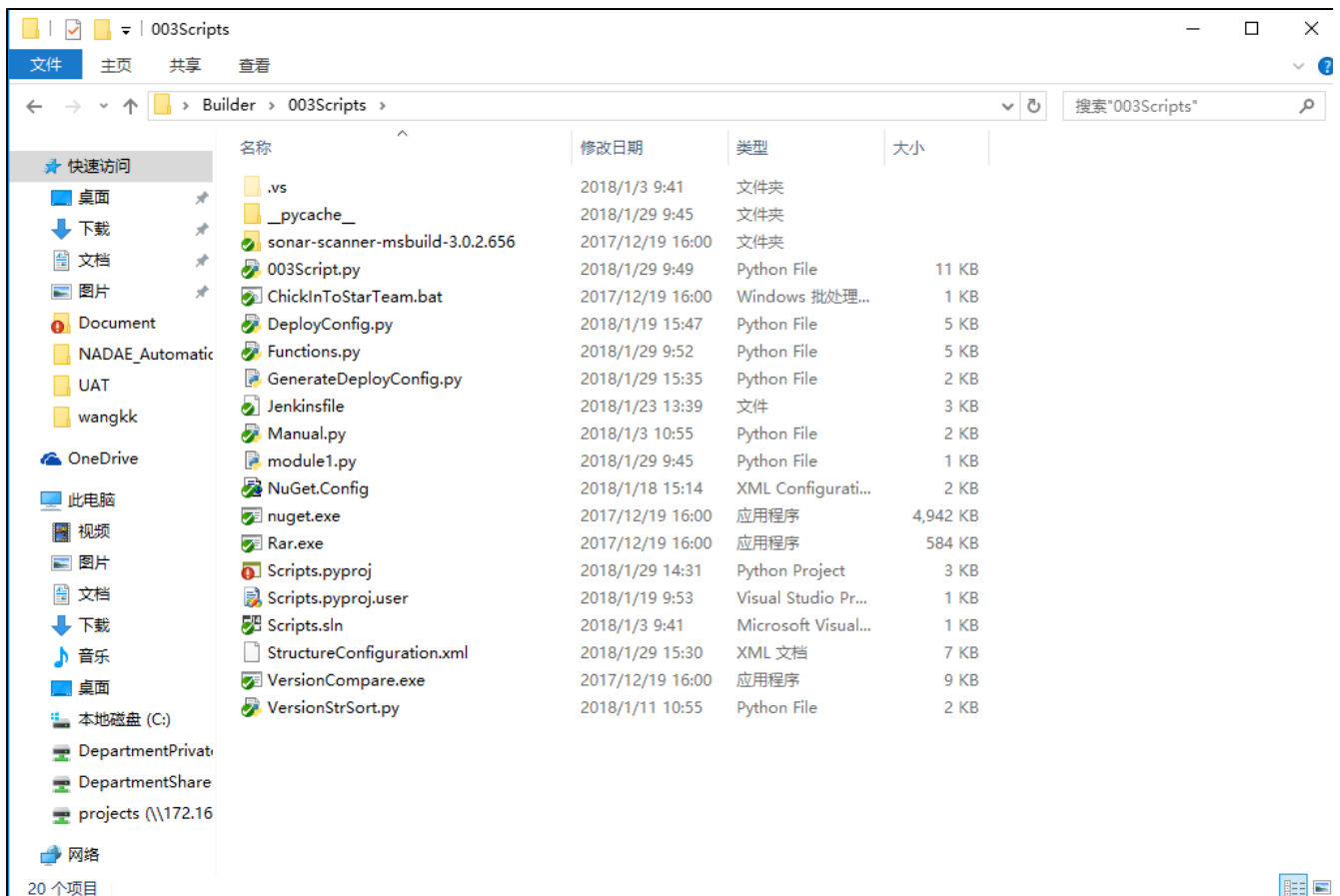
- 配置gitlab hook触发和Jenkins触发选项，提交脚本至Git即可



- 基于Jenkins2
Pipeline我们可以将脚本代码使用Git来进行版本控制，当有新功能时只需提交至Git即可。

【步骤四】

将脚本代码克隆到本地



【步骤五】在学习Groovy脚本和jenkins使用格式的基础上编辑Jenkinsfile文件里面的脚本

- 打开jenkins转到job页面，打开pipeline syntax语法
- 安装所需要的jira插件后，点击生成脚本代码

Overview

This **Snippet Generator** will help you learn the Pipeline Script code which can be used to define various steps. Pick a step you are interested in from the list, configure it, click **Generate Pipeline Script**, and you will see a Pipeline Script statement that would call the step with that configuration. You may copy and paste the whole statement into your script, or pick up just the options you care about. (Most parameters are optional and can be omitted in your script, leaving at default values.)

Steps

Sample Step:

Issue key:

Comment body:

Generate Pipeline Script

`JiraComment body: '代码构建失败。', issueKey: 'XXC'`

Global Variables

There are many features of the Pipeline that are not steps. These are often exposed via global variables, which are not supported by the snippet generator. See the [Global Variables Reference](#) for details.

【步骤六】添加生成的代码至脚本文件中

```

54     }
55     }
56     }
57     }
58     }
59 }
60 post{
61     failure{
62         emailx body: "<hr/>(本邮件是Jenkins自动下发的, 请勿回复!)<br/><hr/>项目名称: $gitlabSourceNames
63         subject: "Jenkins构建通知:$gitlabSourceNamespace - $gitlabSourceRepoName($gitlabBranch) Build #
64         to: '$gitlabUserEmail,wangkk@smee.com.cn'
65
66         jiraComment body: '代码构建失败.', issueKey: 'XXC'
67     }
68     always{
69         cleanWs()
70     }
71 }
72 }

```