如果你还不理解 RBAC，看看 Jenkins 如何做到的

你好，我是悟空。

本文目录如下：

前言

上次我们已经聊过关于自动化部署的四个话题：

* 丝滑的打包部署，一套带走
* 喝杯咖啡，一键部署完成！（建议收藏）
* 喝杯咖啡，一键部署前端项目
* 用代码实现流水线部署，像诗一般优雅

这次我们要接着上面的话题聊下如何来管理 Jenkins 用户的部署权限。

通过本篇你可以学习到如下内容：

* RBAC 的基础知识。
* Jenkins 的角色权限插件的使用。
* 实战：通过角色来管理用户的部署权限。

一、RBAC

Jenkins 对权限的支持是比较弱的，存在以下不足：

* 有多个流水线任务，期望不同用户能看到的任务不一样。
* 一个项目有多套环境，期望用户只能部署某些环境。
* 有的项目只让用户有查看权限，期望不给部署权限。

针对上面的不足，Jenkins 可以通过 **RBAC 插件**的方式来实现对权限的管控。RBAC 是常见的权限控制方案。

> `RBAC（Role-Based Access Control）`：基于角色的权限控制。通过角色关联用户，角色关联权限的方式间接赋予用户权限。

如下图所示，三个用户对应了三种角色，每个角色关联了不同的部署任务，通过这种关联方式间接赋予了用户权限。

二、角色权限插件

目前发现这个角色权限插件是比较好用的，推荐给大家使用。

插件名：Role-based Authorization Strategy。可以到插件管理那里进行安装，如下图所示：

角色权限策略插件

三、选择授权策略

Jenkins 自带了多种授权策略，如下图所示，在全局安全设置中可以选择授权策略。

对应的访问路径如下：

```
Dashboard->Manage Jenkins->Configure Global Security
```

当我们安装好 `Role-based Authorization Strategy` 插件后，就会出现一个 `Role-Base Strategy` 授权策略。

Jenkins 授权策略

下面开始演示如何基于这个授权策略来分配多个流水线的部署权限。

四、创建演示用户

为了更好的演示角色权限管理功能，我创建了 3 个用户以及 4 个流水线任务。

创建用户的路径为：

```
http://<你的jenkins地址>:8080/securityRealm/
```

我在 jenkins 后台创建了几个用户：

* 开发组长1：用户名=passjava-master1
* 测试组长1：用户名=passjava-tester1
* 项目经理1：用户名=passjava-pm1

如下图所示：

五、创建演示任务

我创建了 4 个部署流水线任务，分别对应项目一和项目二的测试和生产环境。如下图所示：

项目一和项目二的测试和生产环境

然后还创建了两个分组：正式环境分组和测试环境分组。

正式环境分组：demo-prod-env

* 项目一正式环境：对应 passjava-prod-project1 任务
* 项目二正式环境：对应 passjava-prod-project2 任务

测试环境分组：demo-test-env

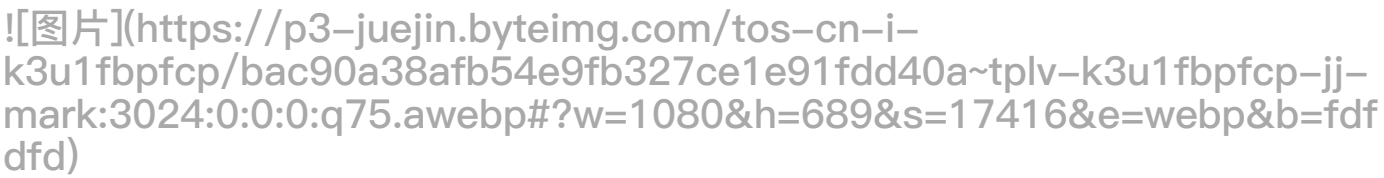
* 项目一测试环境：对应 passjava-test-project1 任务
* 项目二测试环境：对应 passjava-test-project2 任务

分组的好处是可以归类以及可以按组分配权限。

六、创建演示角色

6.1 创建角色的页面路径

创建角色的路径如下图所示：



创建角色的页面路径

```
页面地址：
http://<你的jenkins服务ip>:8080/manage/role-strategy/
访问路径：
Dashboard->Manage Jenkins->Manage and Assign Roles->Manage Roles
```

6.2 三种角色

这个插件可以创建三种角色：

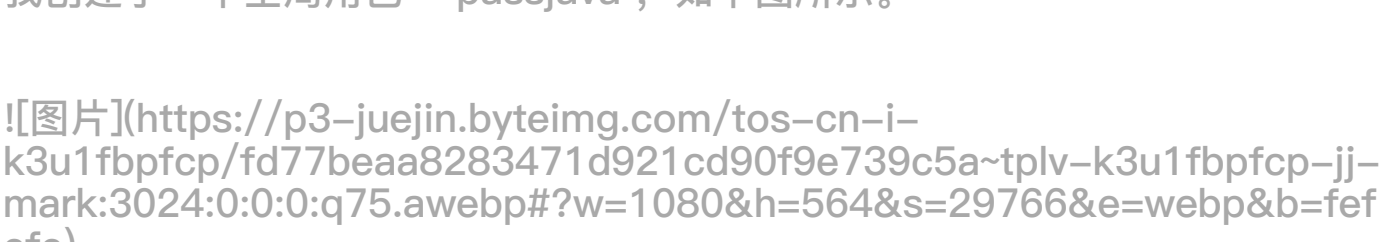
* Global roles：全局角色，例如管理员、作业创建者、匿名等，允许在全局基础上设置总体、代理、任务、运行、查看和 SCM 权限。
* Item roles：任务角色，允许在任务、分组上设置特定权限。
* Agent roles：Agent 角色，本篇用不上。

6.3 全局角色

全局角色适用于 Jenkins 中的任何任务，**并覆盖你在任务角色中指定的任何内容**。也就是说，当你在全局角色中授予角色权限 `Job/Read`，无论你在任务角色中指定什么，都允许该角色读取所有任务。

所以为了分配不同任务权限给不同角色，这里对于全局角色勾选一个 `Overall Read` 权限即可。如下图所示：

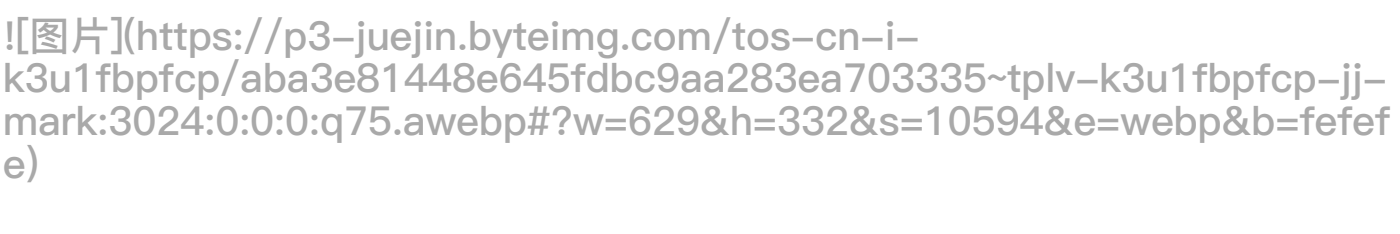
我创建了一个全局角色：`passjava`，如下图所示。



创建全局角色 passjava

后续将上面创建的三个用户加到这个 Global 角色中即可。

注意： 如果这三个用户不加入到 passjava 角色中的话，后续这三个用户登录会提示报错，如下图所示：



6.4 任务角色

我创建了三种任务角色：

* passjava-master：具有部署项目一和项目二的正式环境的权限。
* passjava-tester：具有部署项目一的和项目二的测试环境的权限。
* passjava-pm：具有查看项目一和项目二的正式环境和测试环境的权限，没有部署权限。

如下图所示：

项目角色

下面解释下上面的权限配置：

* Role：代表角色名称
* Pattern：代表正则表达式。例如，如果将该字段设置为 `passjava-prod.*`，则该角色将匹配名称以 `passjava-prod` 开头的所有任务，更多匹配规则可到该插件的官网上查看。点击蓝色的 `passjava-prod.*`，则能看到匹配成功的任务：`passjava-prod-project1`任务 和 `passjava-prod-project2`任务，如下图所示：

正则表达式匹配成功 项目

* Job：任务的权限，我们勾选 Build（部署），Cancel（取消部署），Read（查看任务）即可。

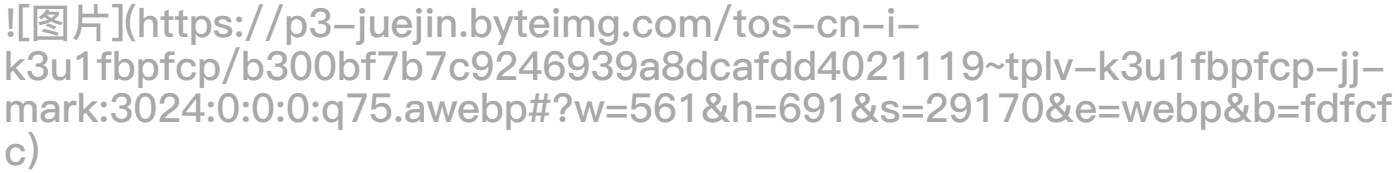
6.5 分配角色

创建好全局角色和任务角色，我们就可以将用户加入到对应的任务角色中，用户和角色是多对多的关系。比如用户张三可以具有角色 A 和角色 B 的权限，角色 A 的权限也可以给用户张三和李四。

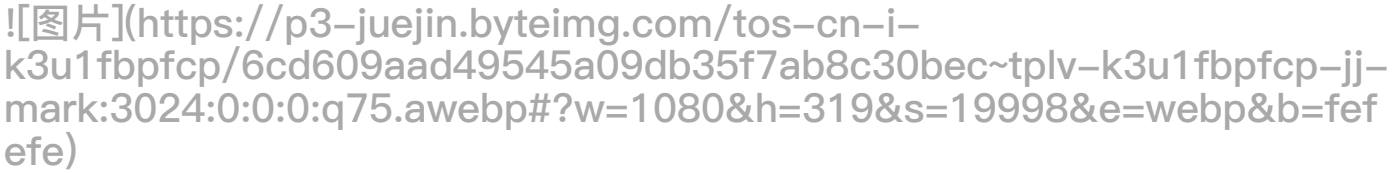
6.5.1 分配全局角色

通过 `Assign Roles` 功能将三个用户都加入到 passjava 角色中。

6.5.2 分配项目角色



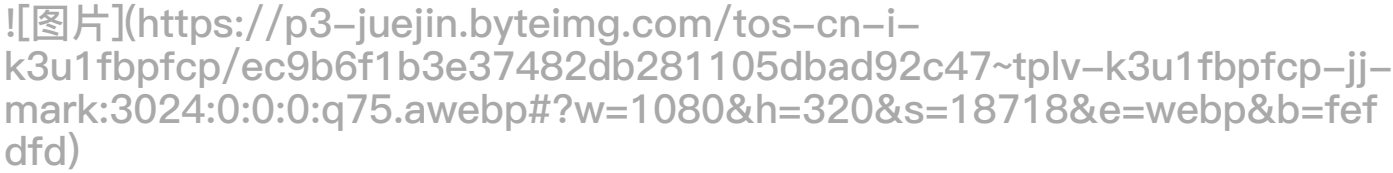
* 开发组长具有 passjava-master 权限，可以部署项目一和项目二的正式环境。开发组长登录系统后看到的任务列表如下图所示：



开发组长查看项目列表

* 项目经理具有 passjava-pm 权限，可以查看项目一和项目二的测试和正式环境的部署情况。项目经理登录系统后看到的任务列表如下图所示：

* 测试组长具有 passjava-tester 权限，可以部署项目一和项目二的测试环境。测试组长登录系统后看到的任务列表如下图所示：



而是组长查看项目列表

可以从上面的结果看出用户、角色、权限分配完成，不同的用户可以部署不同的环境。

七、总结

通过本篇的学习，我们了解了 RBAC 以及角色权限插件的使用。通过实战掌握了如何配置不同用户具有不同角色，不同角色具有不同任务的权限，从而可以更安全地管理用户的部署权限。

回到最开始提到的 Jenkins 的不足之处我们来看看解决方案是怎么样的：

* 有多个流水线任务，期望不同用户能看到的任务不一样，解决方案是可以给不同角色分配不同的任务，不同用户赋予不同角色。如实战中的开发组长和测试组长看到的任务不一样。
* 一个项目有多套环境，期望用户只能部署某些环境，解决方案是对多套环境创建对应的多个任务，多个角色拥有对应环境的任务，并对用户赋予对应环境的角色。如实战中开发组长只能看到正式环境的任务。
* 有的项目只让用户有查看权限，期望不给部署权限，解决方案是添加一个只能查看对应项目的任务的角色，并给用户赋予这个角色，如实战中项目经理1只有查看权限，没有部署权限。