## ****Tower****

前言:待

1. **Tower的安装**

* 1. **下载安装包 ansible-tower-setup**

https://www.ansible.com/products/tower/trial

* 1. **安装前配置及检查**
* 内存4G以上
* 创建目录文件：mkdir -p  /var/log/tower
* 配置修改

vim  inventory #标绿需要修改

|  |
| --- |
| [tower] localhost ansible\_connection=local [database]  [all:vars] admin\_password= 'eccom' pg\_host='localhost' pg\_port='5432'  pg\_database='awx' pg\_username='awx' pg\_password='admin' |

  关于集群版参考：

https://docs.ansible.com/ansible-tower/latest/html/quickinstall/install\_script.html

* 1. **实例postgreSQL数据库**
* mkdir -p /data/pgdata
* docker pull postgres:9.6 （需要安装docker）
* docker run --name postgres\_ansible -d -p 5432:5432 -e POSTGRES\_PASSWORD=admin -e PGDATA=/data/pgdata -e POSTGRES\_USER=awx -e POSTGRES\_DB=awx postgres:9.6

**验证操作**

|  |
| --- |
| docker exec -it postgres\_ansible bash  psql -U awx -d awx -h localhost -p 5432 ctrl+p+q #退出容器 |

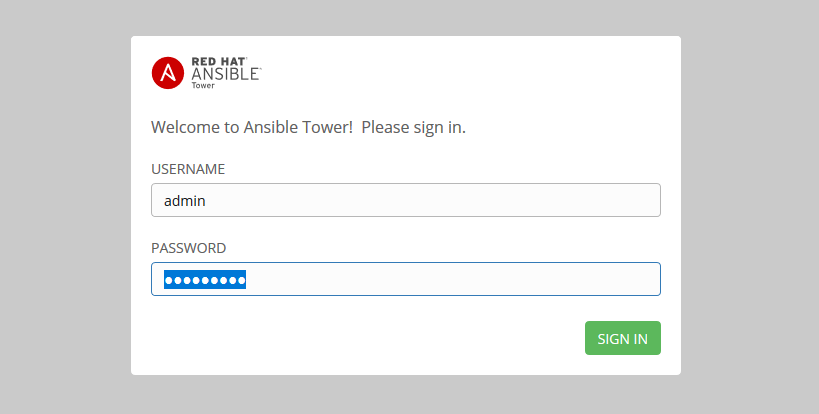
* 1. **执行安装脚本**

sh setup.sh

安装成功输出日志

|  |
| --- |
| PLAY [Install Tower isolated node(s)] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  skipping: no hosts matched  PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  localhost : ok=123 changed=50 unreachable=0 failed=0  The setup process completed successfully.  Setup log saved to /var/log/tower/setup-2019-04-12-17:55:10.log |

**可以通过80端口进行ansible tower 访问**

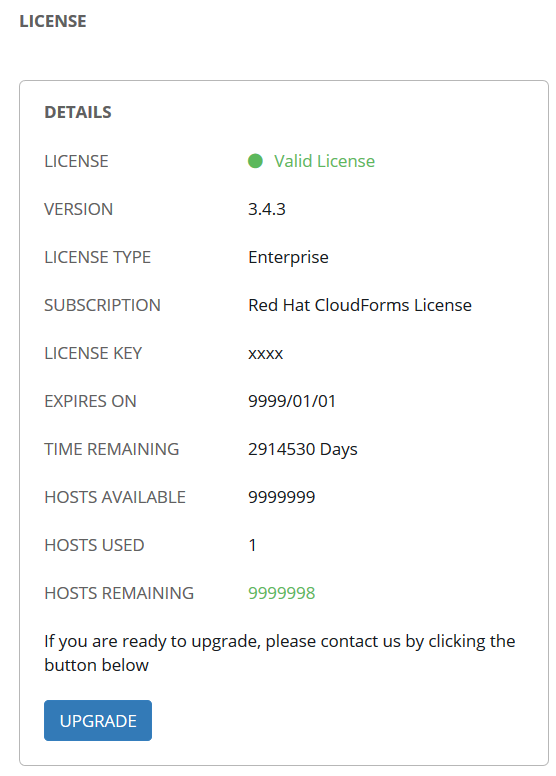
****

 license 官方免费申请 <https://www.ansible.com/license>

* 1. **企业版无限node破解**

先申请企业版使用30天license ,激活，再执行如下：

|  |
| --- |
| echo codyguo > /var/lib/awx/i18n.db  ansible-tower-service restart |



1. **使用教程**
   1. **面板介绍**

|  |  |
| --- | --- |
| # viewes |  |
| Dashboard | 仪表盘展示信息的 |
| Jobs | 跑过的任务记录 |
| Schedules | 计划任务 |
| My View | 查看用户的工作模版，和任务记录 |
| # resources |  |
| Templates | 任务模版，配置调用playbook执行时的各种参数，  从此处添加计划任务 |
| Credentials | 配置连接 机器/云主机api Key/自定义的凭证类型  的账号密码等信息 |
| Projects | 这里配置项目对应的playbook，可以从Git上拉取  或从本地文件夹读取playbook |
| Inventories | 资产清单 |
| Inventory Scripts | 自定义获取资产清单的脚本 |
| # access |  |
| Organizations | 组织管理 |
| Users | 用户管理 |
| Teams | 用户组管理 |
| # Administration |  |
| Credential Types | 自定义凭证类型，添加后可在Credentials中使用 |
| Notifications | 配置任务通知，支持电子邮件，Twillio电话等 |
| Management Jobs | 计划任务管理 |
| Instance Groups | 资产组管理 |
| Applications | 自定义应用 |
| Settings | 设置 |

* 1. **关于资产配置**
     1. **[[1]](#footnote-0)CLI Input Inventory**

|  |
| --- |
| tower-manage inventory\_import --source=/etc/ansible/hosts --inventory-name=dev\_env\_node  #先在界面端参考2.3.3建立个dev\_env\_node inventory |

可以修改资产清单文件路径

其他参数：

--source 指定inventory文件

--group-filter 从文件中通过组名过滤

--host-filter 通过host name过滤

--inventory-name 导入到指定名称资产清单

--inventory-id 导入到指定ID的资产清单

# name 和 id 选一个

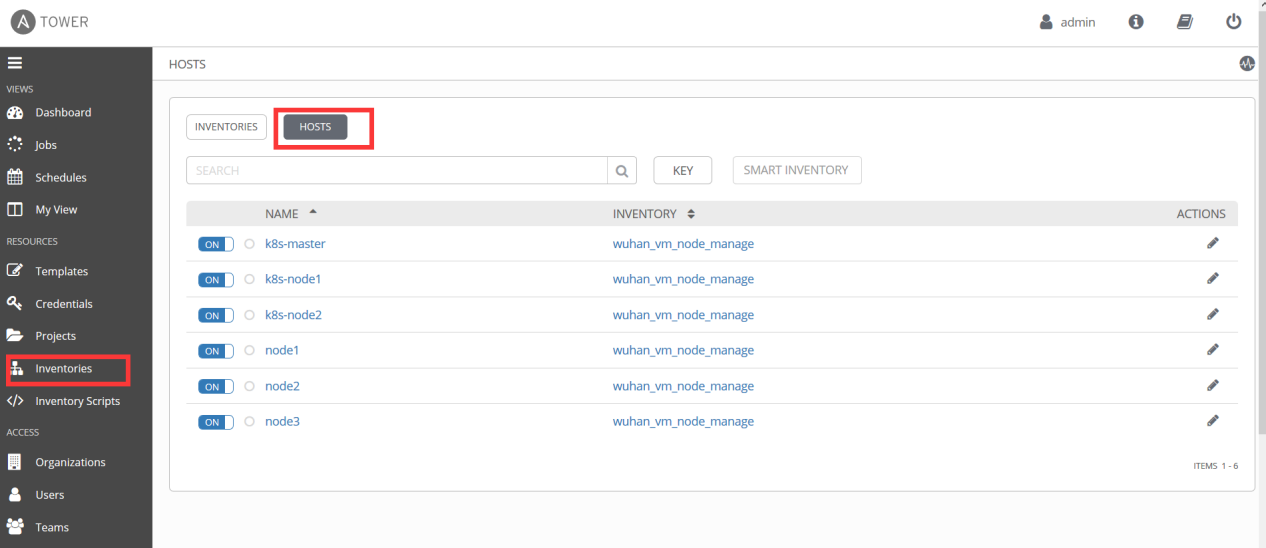
--overwrite 覆盖主机和组，默认不覆盖

--overwrite-vars 覆盖主机变量

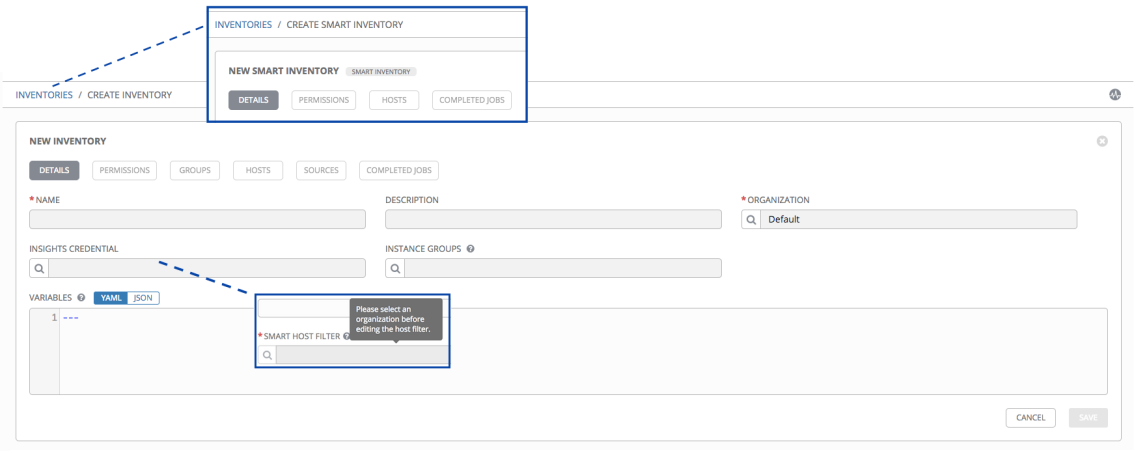
--keep-vars 保持主机变量

--enabled-value 导入的主机状态是否激活默认激活

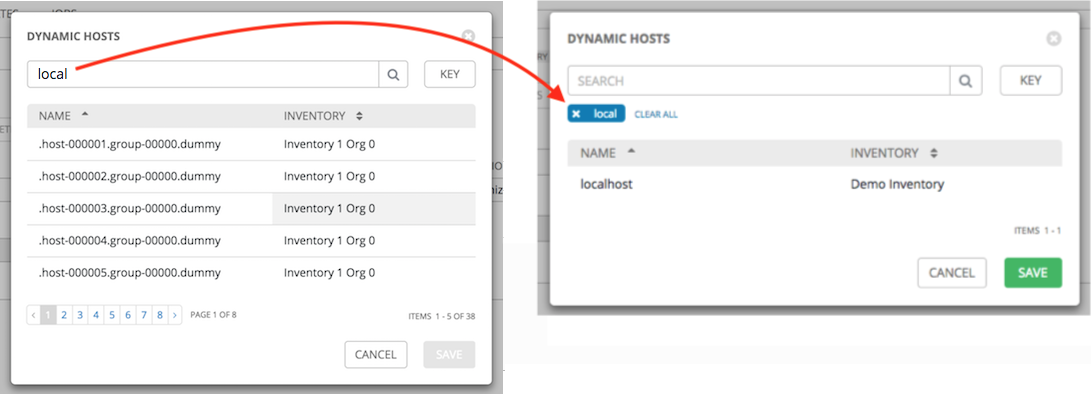
导入如下结果如下



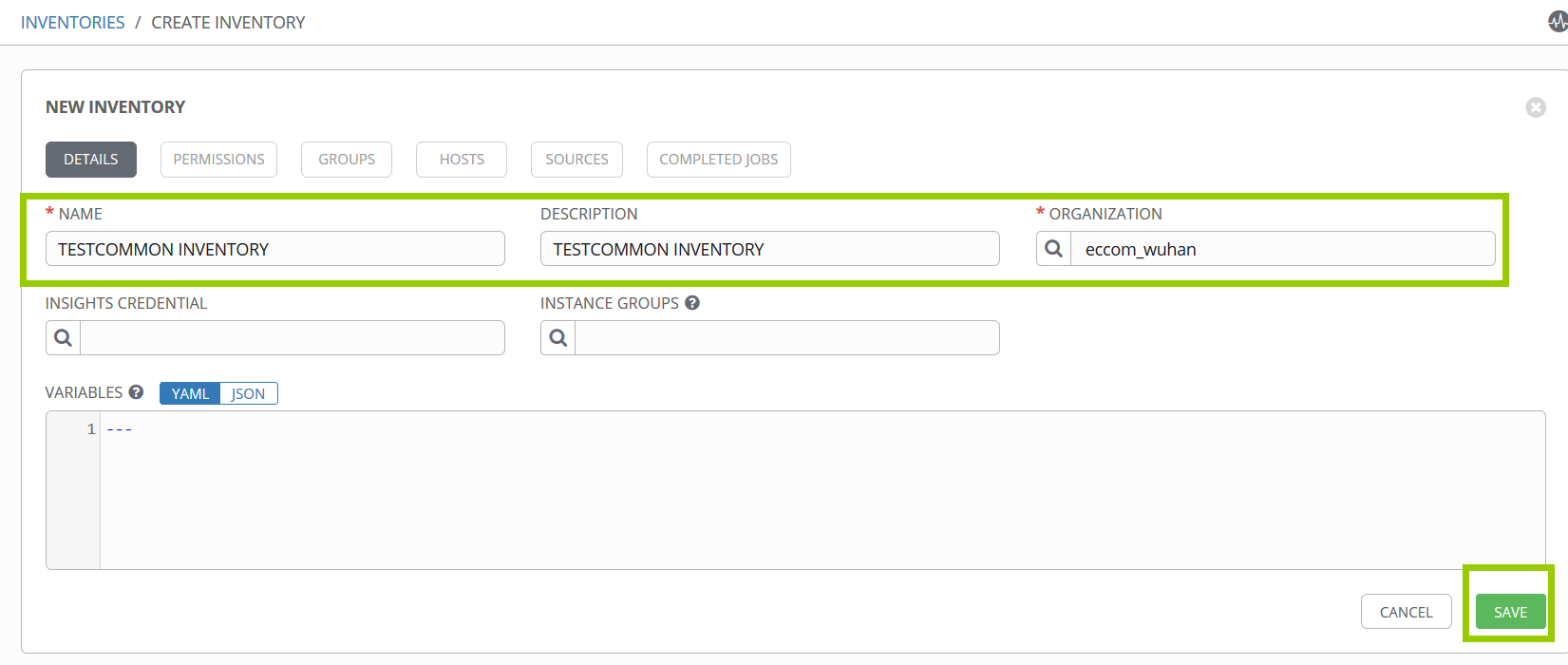
### **Smart** Inventory



现有资产再组资源组可采用此方法，填写必要必要信息后，通过筛选功能选取新组成员。



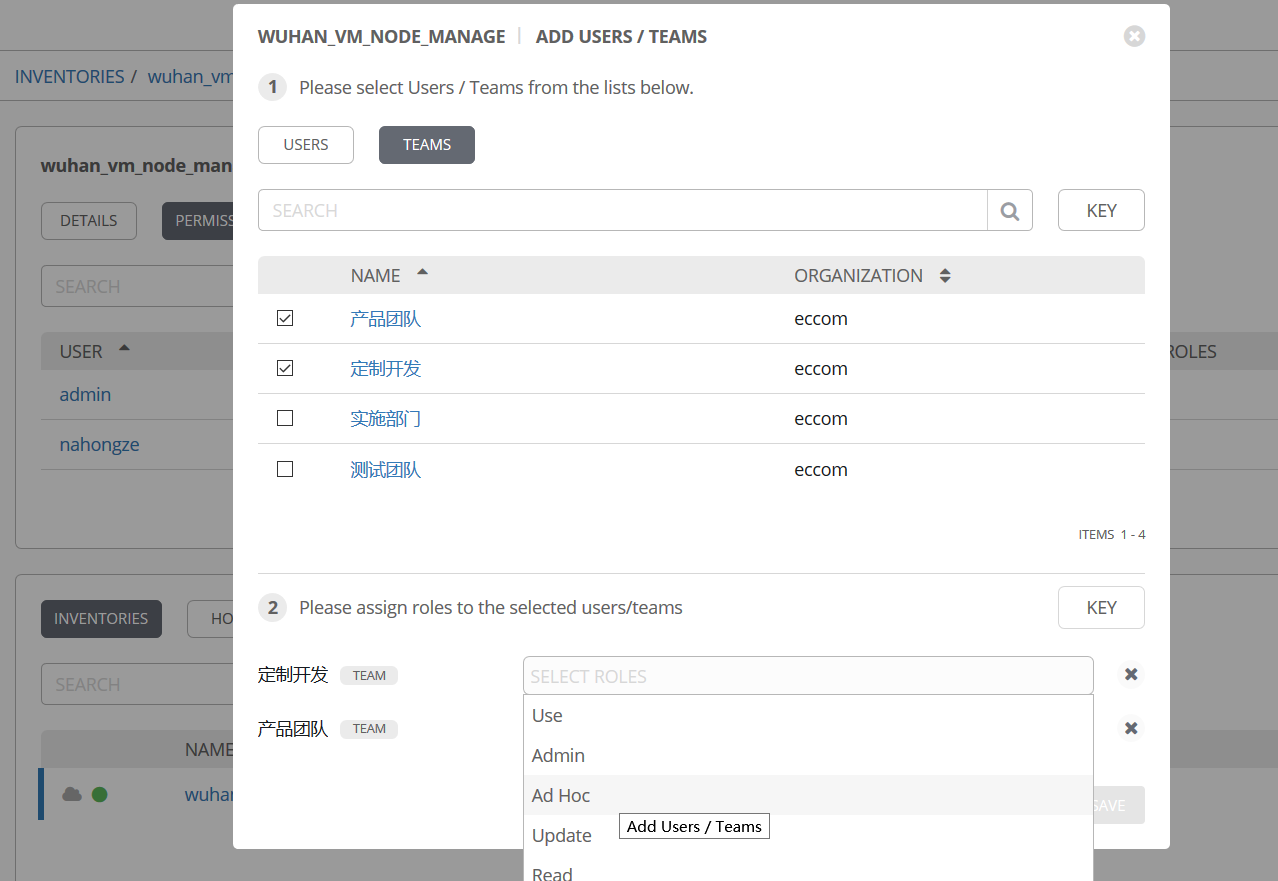
### **Common I**nventory



选择添加inventory后，填写必要信息，点击保存之后才可继续配置权限，组，主机，源以及查看已完成的任务。

* 1. **资产分权与组织管理功能**

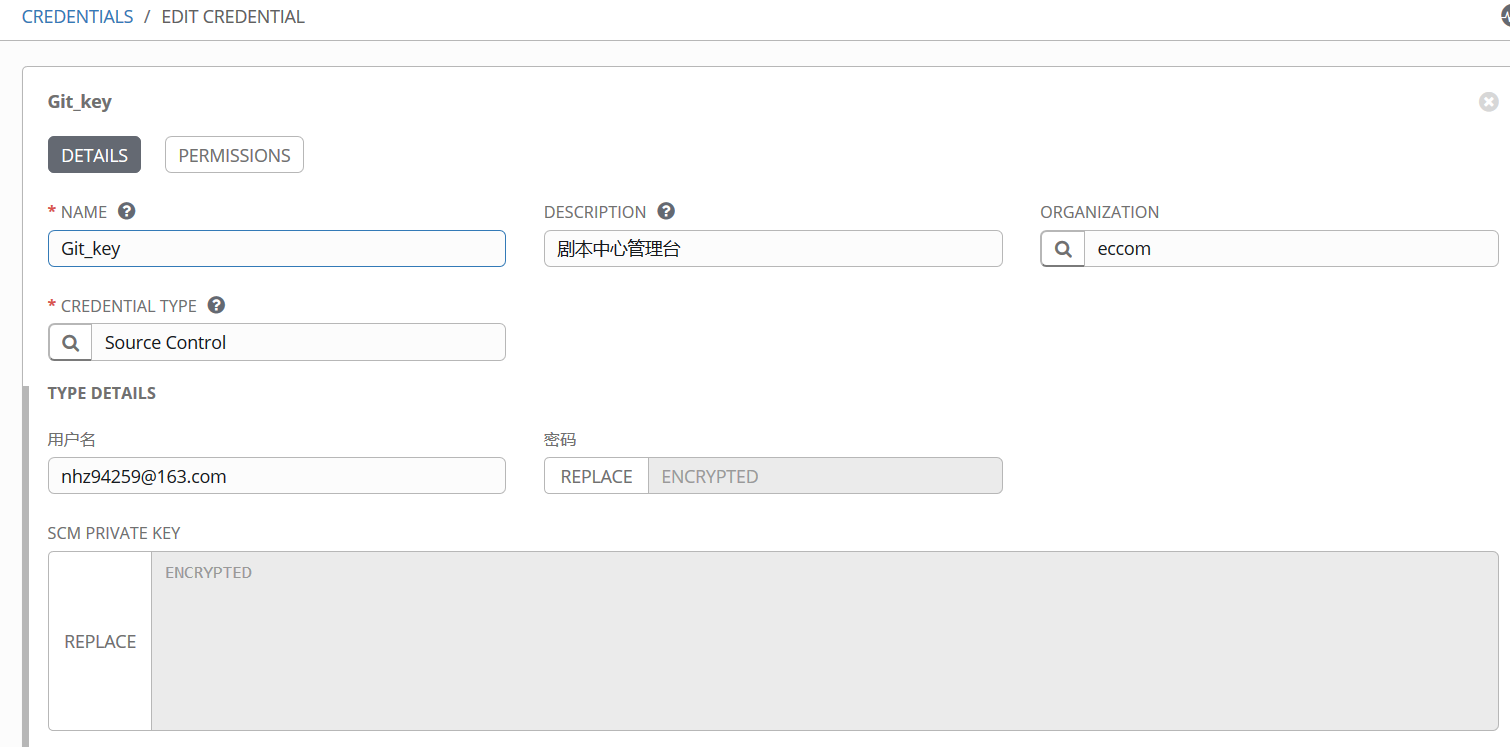
功能简单只副一张资产分配权限的图



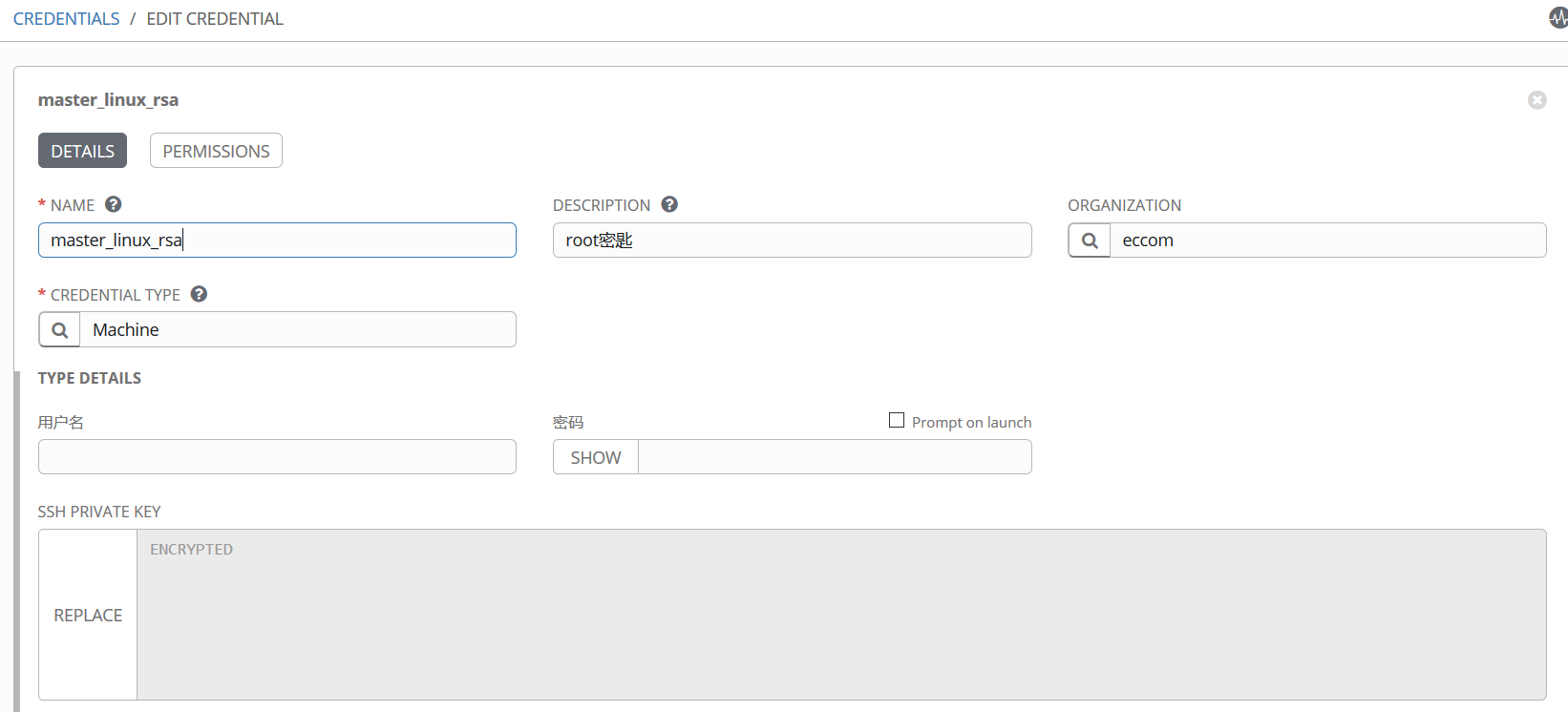
* 1. **配置git**

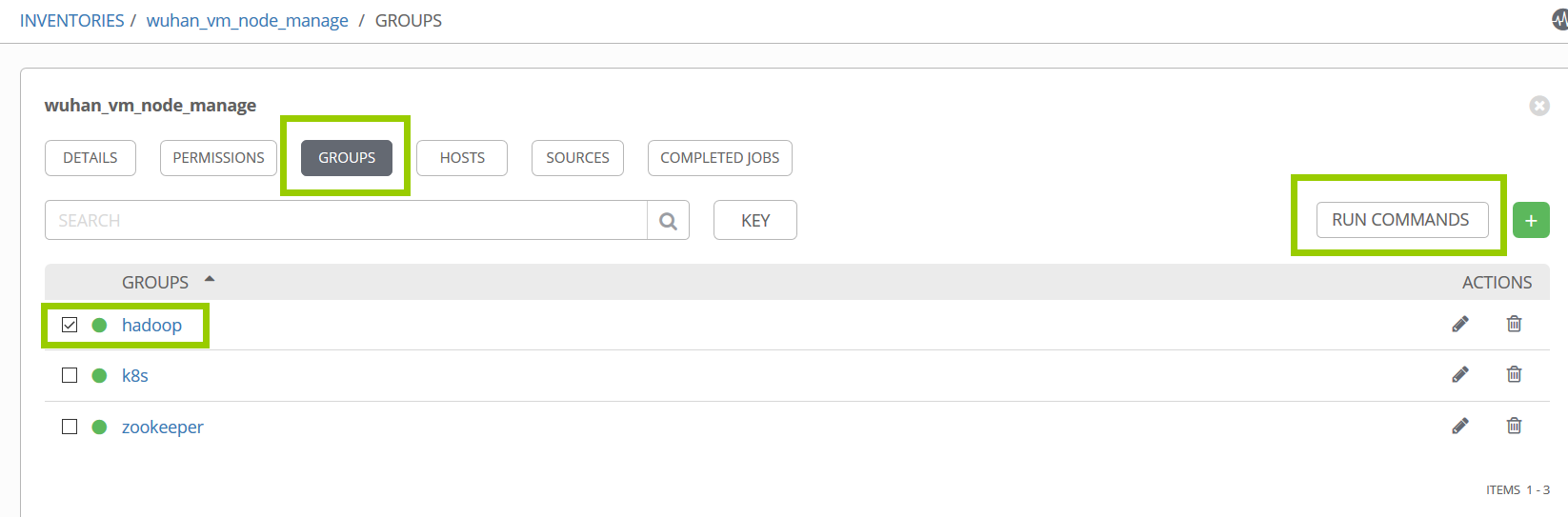
### 创建版本控制凭证

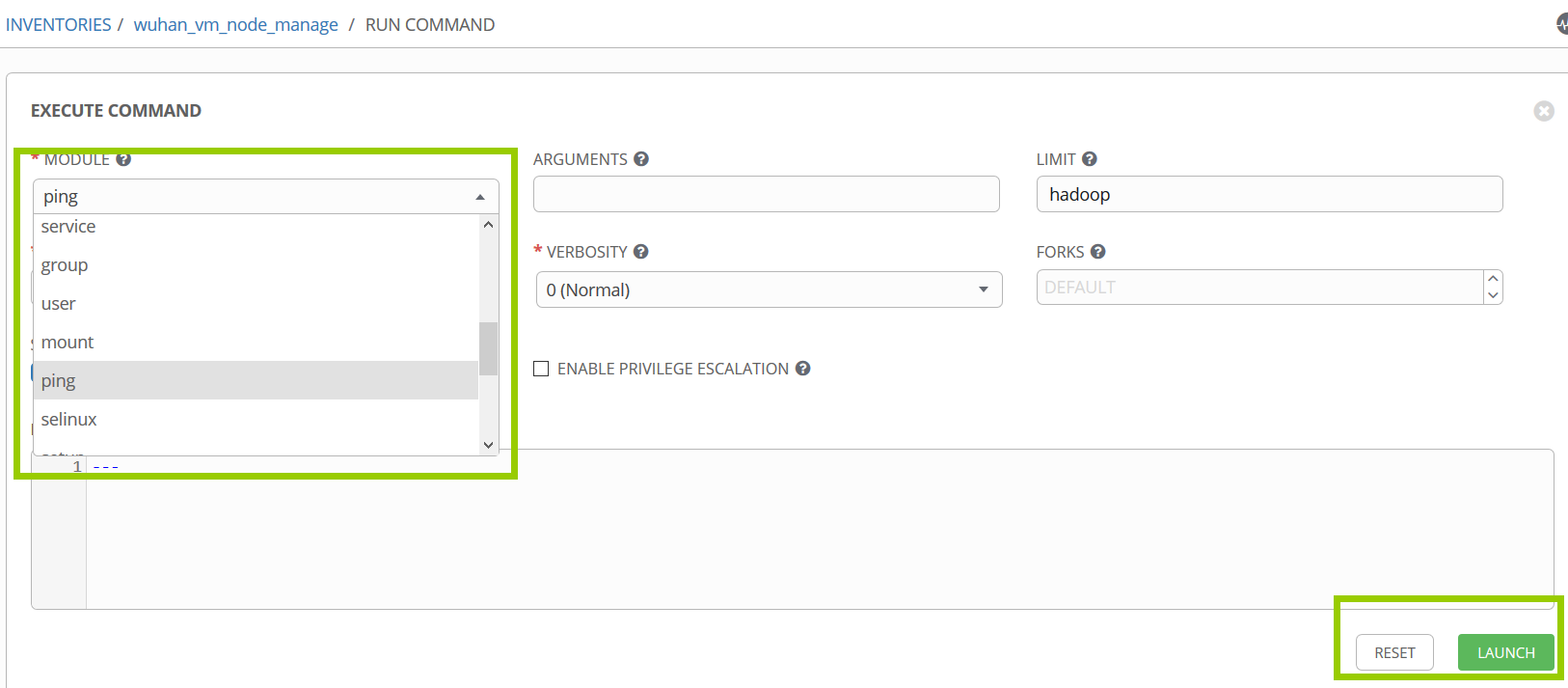
* 填写必要信息，选择版本控制，输入git账号，选择私钥文件，点击保存



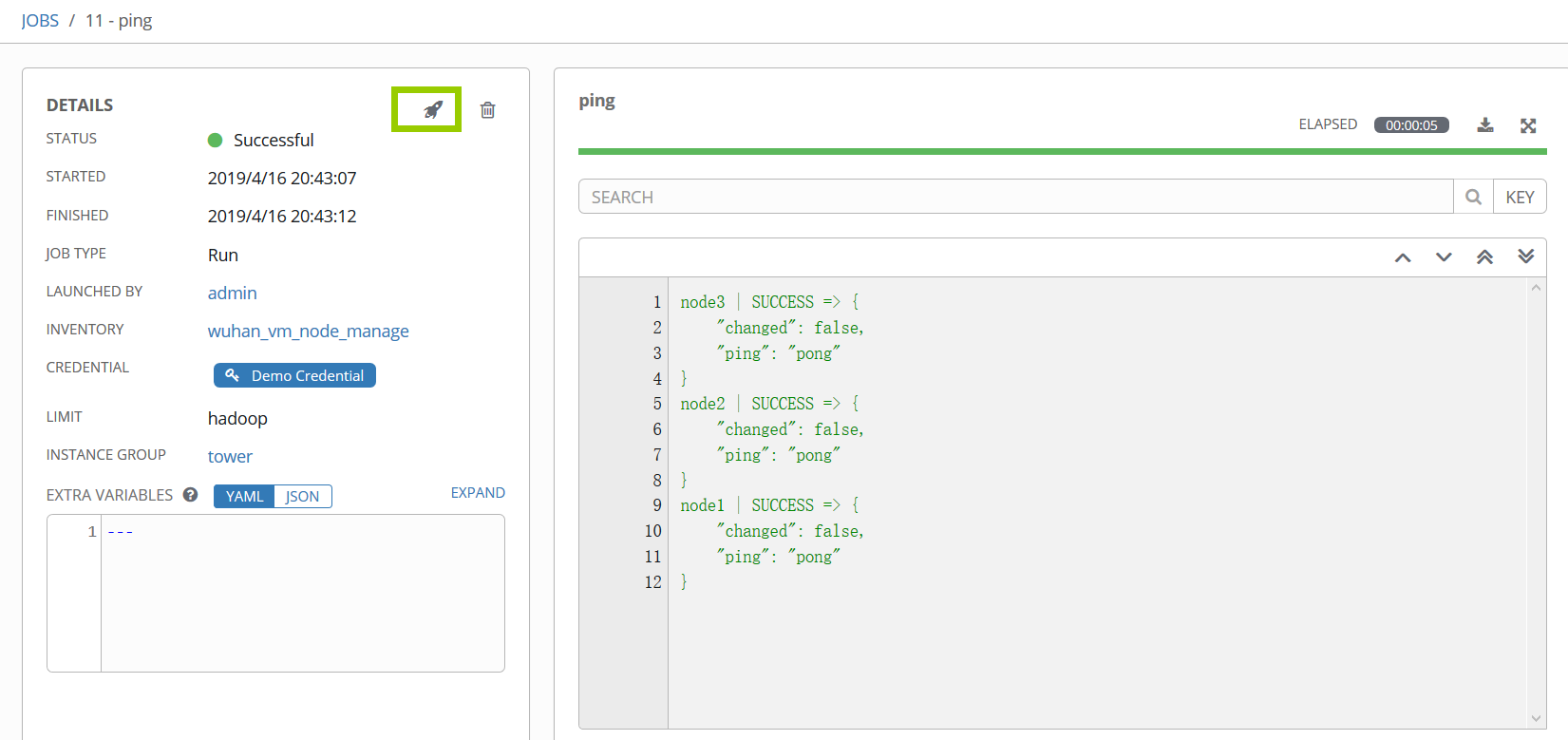
### 创建登陆机器凭证



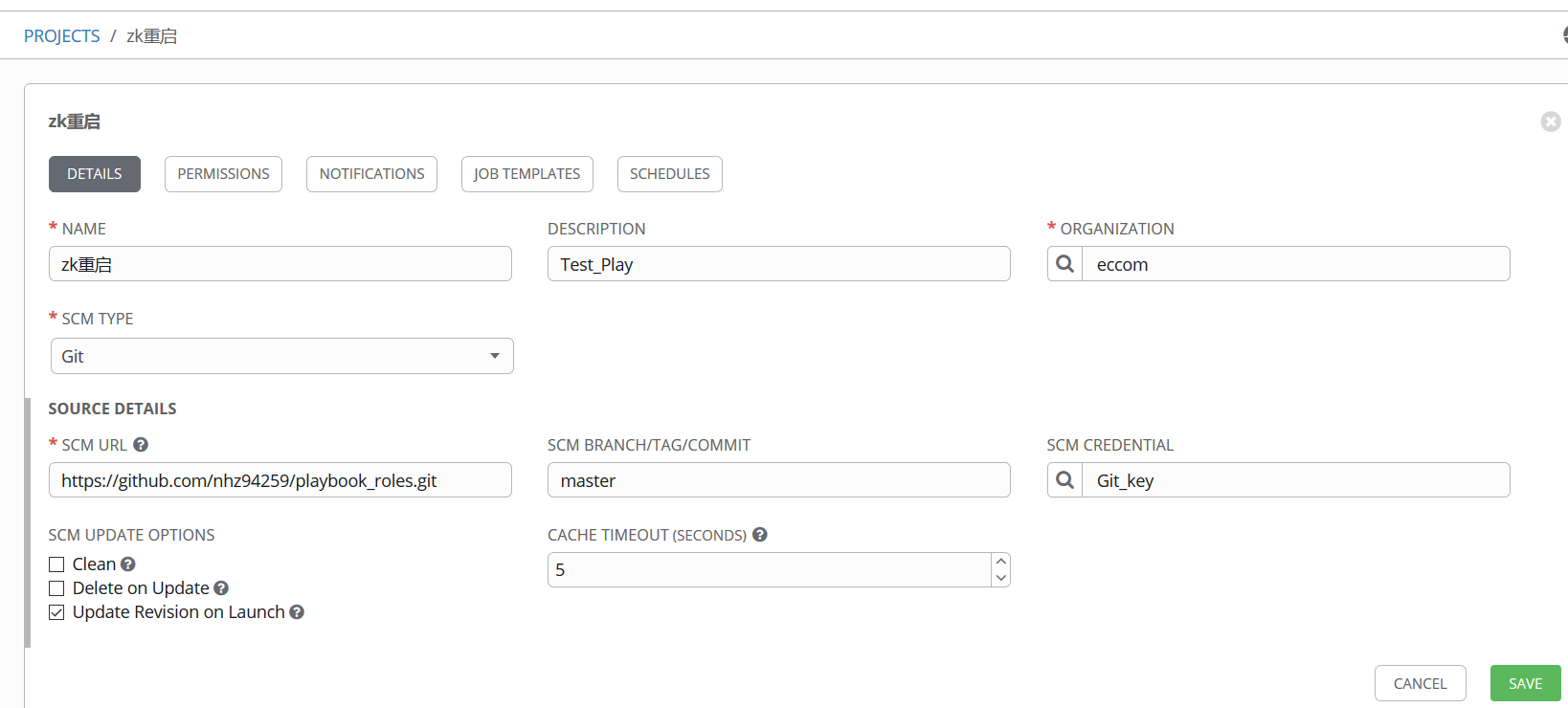
* 1. **Ad-Hoc**
* 选择执行组
* 选择执行命令类型输入命令



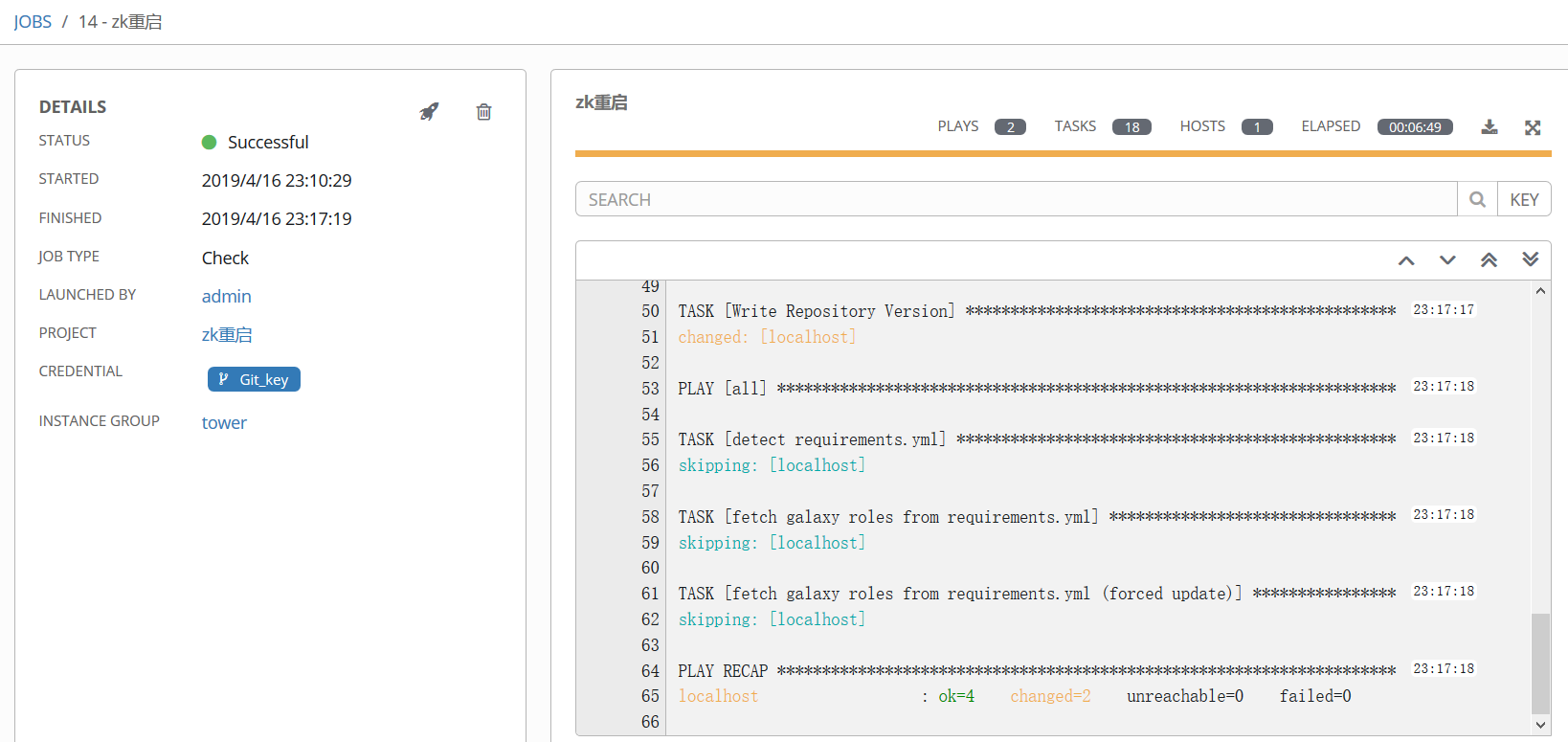
* 执行结果及再执行



* 1. **Playbook**
* 项目中建立，如下建立



* jobs中查看执行状态



官方文档https://docs.ansible.com/ansible-tower/latest/html/userguide/overview.html

1. [↑](#footnote-ref-0)