# Ansible Tower 自动化实践（2）- 使用用户和团队管理访问权限（2）

用户类型：

- 默认情况下，Ansible Tower安装程序会创建对Ansible Tower安装具有完全控制权限的

  admin 用户。

- 使用特殊 admin 账户时，Ansible Tower管理员可以登录Web界面并创建其他用户。

- Ansible Tower中的三种用户类型：

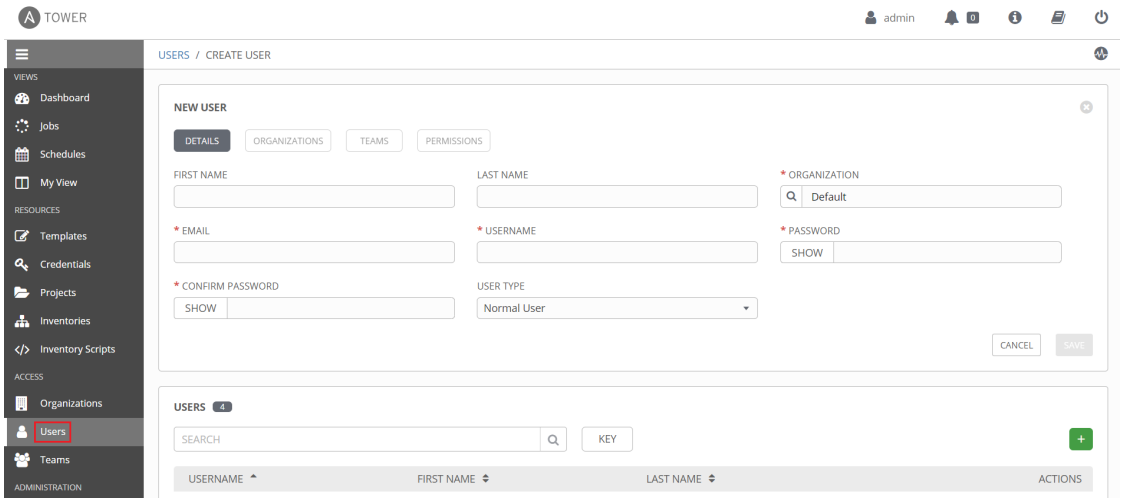
  1. System Administrator：
     a. 该用户类型（也称为超级用户）提供不受限制的访问权限，在整个Ansible Tower
        安装内执行任何操作。
     b. 该用户类型是特殊的独立角色（singleton role），对Ansible Tower上所有组织
        中的所有对象具有读写权限。
     c. 安装程序创建的 admin 用户具有系统管理员独立角色，因此仅应由Ansible Tower
        管理员使用。

  2. System Auditor：
     a. 该用户类型也具有特殊的独立角色，对整个Ansible Tower安装具有只读访问权限。

  3. Normal User：
     a. 该用户是标准用户类型，最初没有分配特殊角色，并以最少的访问权限开始。
     b. 该用户类型没有被分配任何独立角色，并且仅被分配与用户所属组织相关联的角色。
        * 注意：组织层级关系（organization -> team -> user）

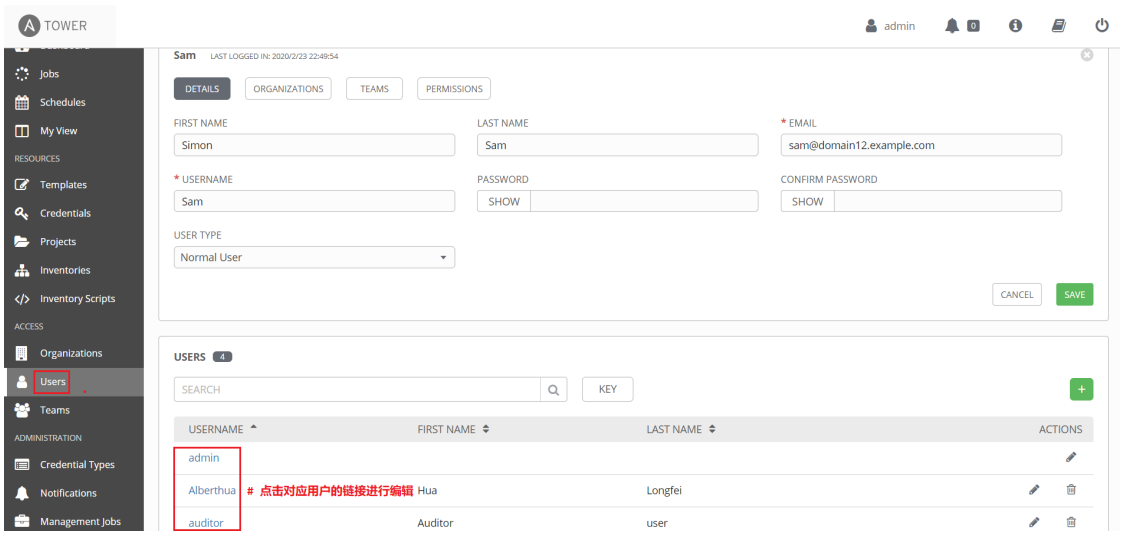
创建用户：

- 使用 admin 用户登录Ansible Tower Web界面创建用户，如下所示：




编辑用户：

- 使用 admin 用户登录Ansible Tower Web界面编辑用户，如下所示：




组织角色：organization roles

- 重要：

  新创建的用户根据其用户类型从其组织中继承特定角色，即用户类型与所在组织决定用户角色。

- 可以在创建后为用户分配其他角色，以授予查看、使用或更改其他Ansible Tower对象的权限。

- 一个组织本身就是其中一个对象。

- 可以为用户分配组织的四种角色：

1. Organizational Admin：
   a. 被分配了组织的Admin角色后，用户可以管理该组织的所有方面，包括读取和更改组织，
      以及添加和删除组织中的用户和团队。
   b. 存在多个相关的管理角色，限制Admin的访问权限：

* Project Admin：
可以创建、读取、更新和删除组织中的任何项目（CRUD权限）。
在与 *Inventory Admin* 权限结合使用时，允许用户创建作业模板。

* Inventory Admin：
可以创建、读取、更新和删除组织中的任何清单（CRUD权限）。
与 *Job Template Admin* 和 *Project Admin* 角色结合使用时，允许用户完全
控制组织内的作业模板。

* Credential Admin：
可以创建、读取、更新和删除共享凭据

* Notification Admin：
可用于分配通知

* Workflow Admin：
可以在组织内创建工作流

* Job Template Admin：
可以对作业模板上的非敏感字段进行更改。

2. Auditor：
   a. 被分配组织的Auditor角色时，用户会获得该组织的只读（ro）访问权限。

3. Member：
   a. 组织 Member 角色只能让用户查看属于组织成员的用户列表，以及所分配的组织
   角色。
   b. 与组织Admin和Auditor角色不同， Member 角色不为用户提供组织所包含的任何
   资源的权限，如团队、凭据、项目、清单、作业模板、工作模板和通知。

4. Read：
   a. 组织 Read 角色只能让用户查看属于组织成员的用户列表，以及所分配的组织角色。
   b. 因此，拥有组织的 Member 角色的用户等同于拥有组织的 Read 角色的用户。

5. Excute：
   a. 具有 Excute 角色的用户会获得在组织中执行作业模板（*job template*）和工作流
   作业模板（*workflow job template*）的权限。

* 注意：
  1. 拥有 System Administrator 独立角色的用户将继承Ansible Tower内的每个组织的
     Admin 角色。
  2. 拥有 System Auditor 独立角色的用户将继承Ansible Tower内的每个组织的 Auditor
     角色。
  3. Normal User 类型创建的用户会被分配组织的 Member 角色，这是在 Ansible Tower
     中创建用户时分配。
  4. 稍后可以添加其他角色，包括其他组织的其他 Member 角色。

管理用户组织角色：

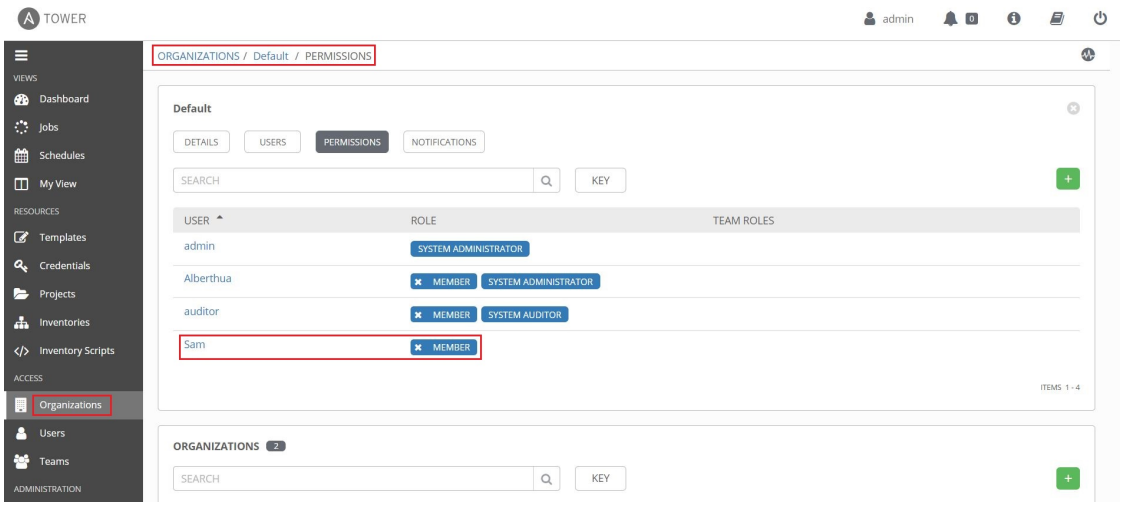
- 在组织中对用户角色的完全管理需要以下步骤：

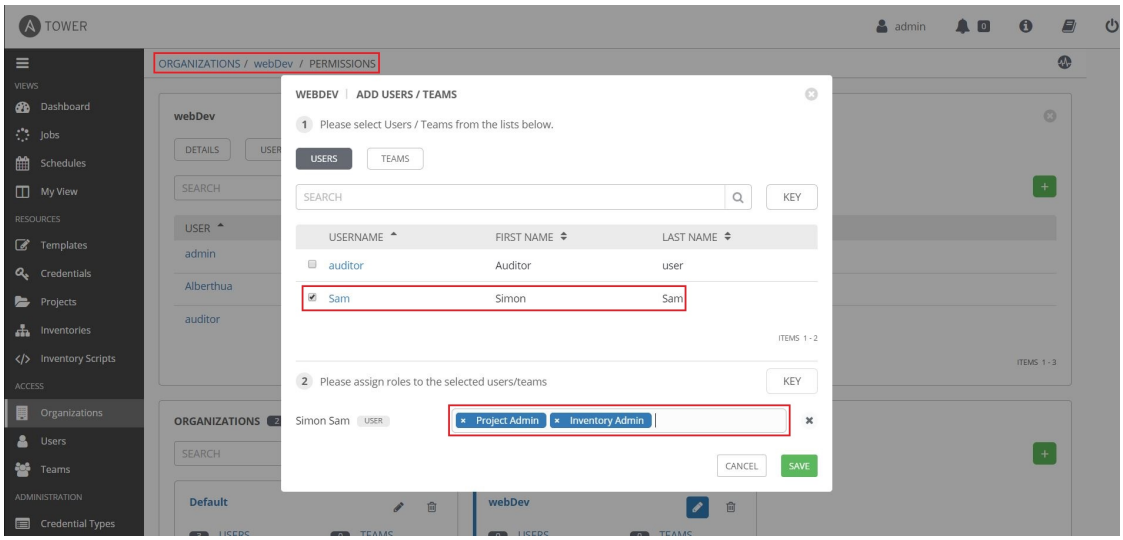
admin login Ansible Tower Web -> Organizations -> Permissions

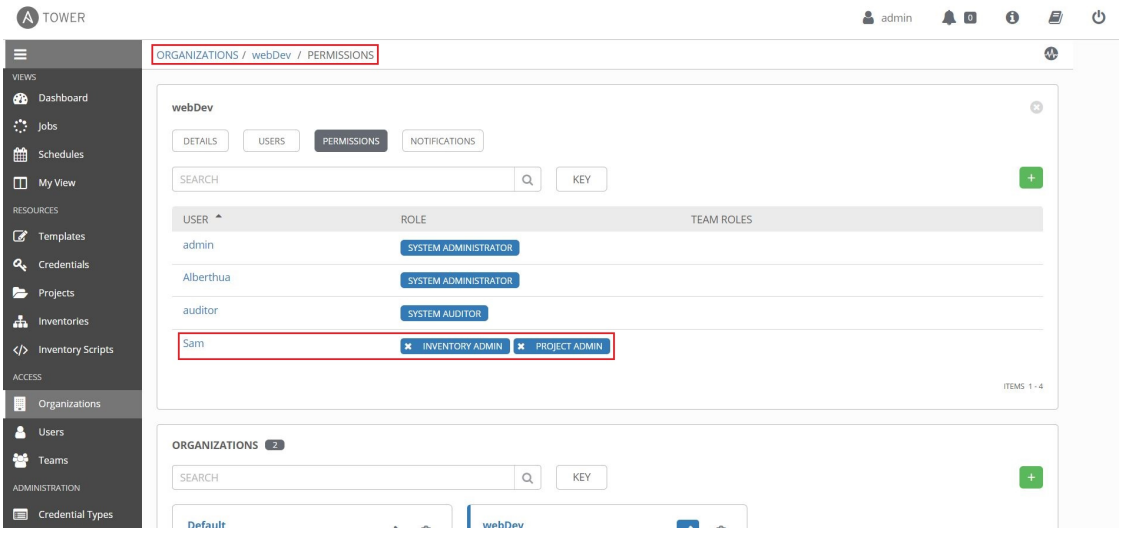
- 若用户未显示在列表中，并且需要组织中的角色，或者用户存在并需要其他组织角色，

  可使用以下步骤：
  1. 在 ADD USERS/TEAMS 屏幕中的 USERS 下，选中所需用户旁边的框。
  2. 单击 SELECT ROLES 下拉列表，再选择用户所需的组织角色。
     此步骤可以重复多次，为一个用户添加多个角色。
  3. 单击 SAVE，将角色分配给该组织的用户。
  4. 单击角色前面的 X ，以从用户中删除现有的角色。

- 示例：为用户配置不同组织中的角色（Sam用户在Default组织与webDev组织中的角色）






第二节：使用团队高效地管理用户
团队：team

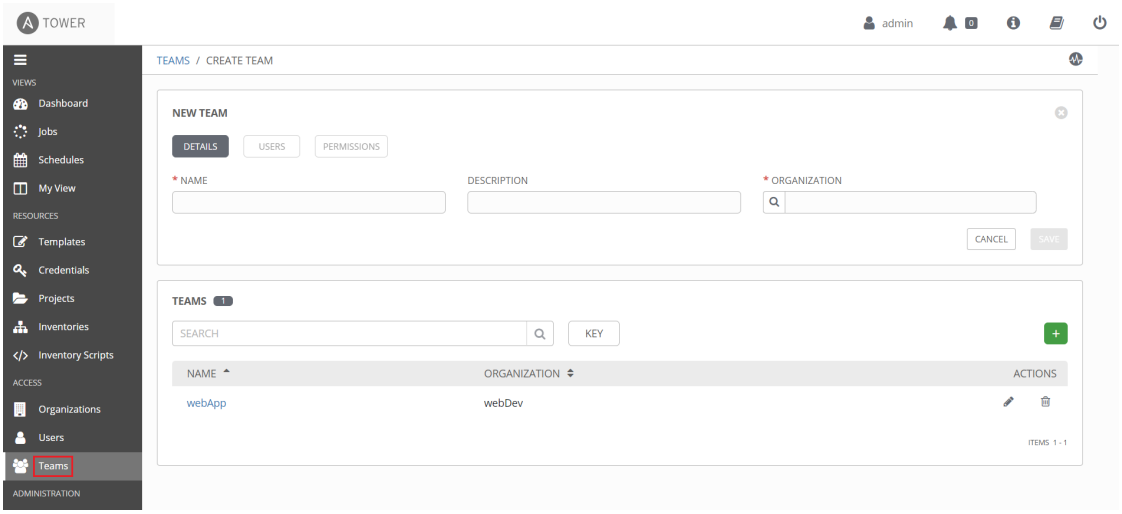
- 团队是用户组。
- 属于团队成员的用户将继承分配给该团队的角色。
- Ansible Tower管理员可以将角色分配给代表一组用户的团队，而不是将相同的角色分配给

  多个用户。

- 在Ansible Tower中，用户以对象的形式存在于整个Tower范围的级别。
- 用户可以在多个组织中拥有角色。
- 一个团队属于一个组织，但 System Administrator 可以分配属于其他组织的团队角色。
- 可以使用团队更轻松地将Ansible Tower资源的特定角色分配给一组用户。

创建团队：

- 使用 admin 用户登录Ansible Tower Web界面创建团队，如下所示：




团队角色：

- 可以为用户分配特定的团队角色。

- 这些角色控制用户是否被视为该团队的一部分，能否对其进行管理，或者可以查看其成员资格。

- 可以为用户分配以下一个或多个团队角色：

1. member：
   a. 团队 member 角色可让用户继承授予给该团队的Ansible Tower资源的角色。
   b. 它还授予用户查看团队用户和相关团队角色的能力。

2. admin：
   a. 团队 admin 角色授予用户对团队的完全控制权。
   b. 拥有此团队角色的用户可以管理团队的用户及其关联的团队角色。
   c. 拥有团队 admin 角色的用户也可以管理团队被分配了 admin 角色的资源的团队角色。
   d. 拥有团队 admin 角色的用户只能在资源本身也授予了团队 admin 角色时管理资源的
      团队角色。
      如，用户若要为某一团队授予项目的 use 角色，该用户必须同时拥有该团队和项目的
      admin 角色。

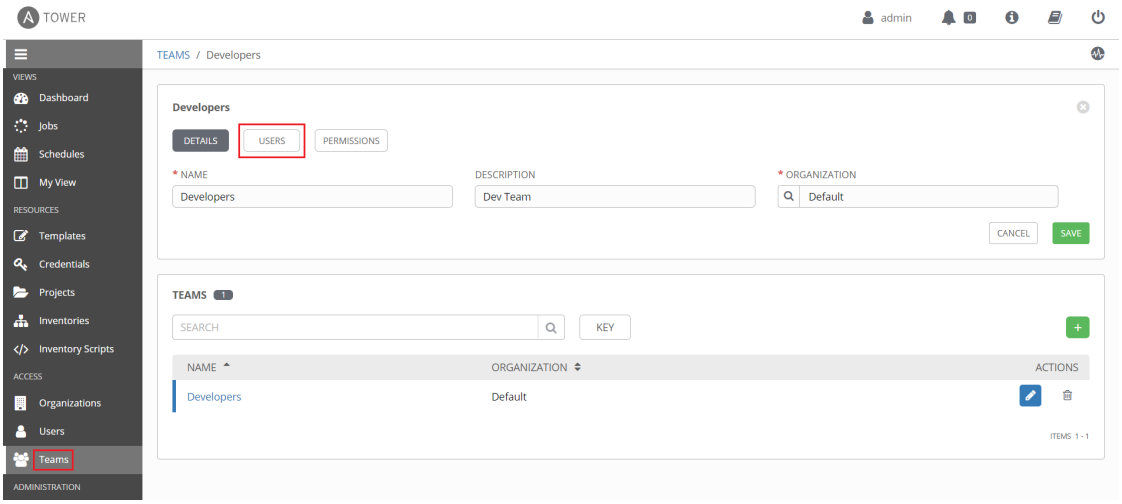
3. read：
   a. 团队 read 角色授予用户查看团队用户及其相关团队角色的能力。
   b. 但是，被分配了团队 read 角色的用户不会继承已针对Ansible Tower资源授予该团队
      的角色。

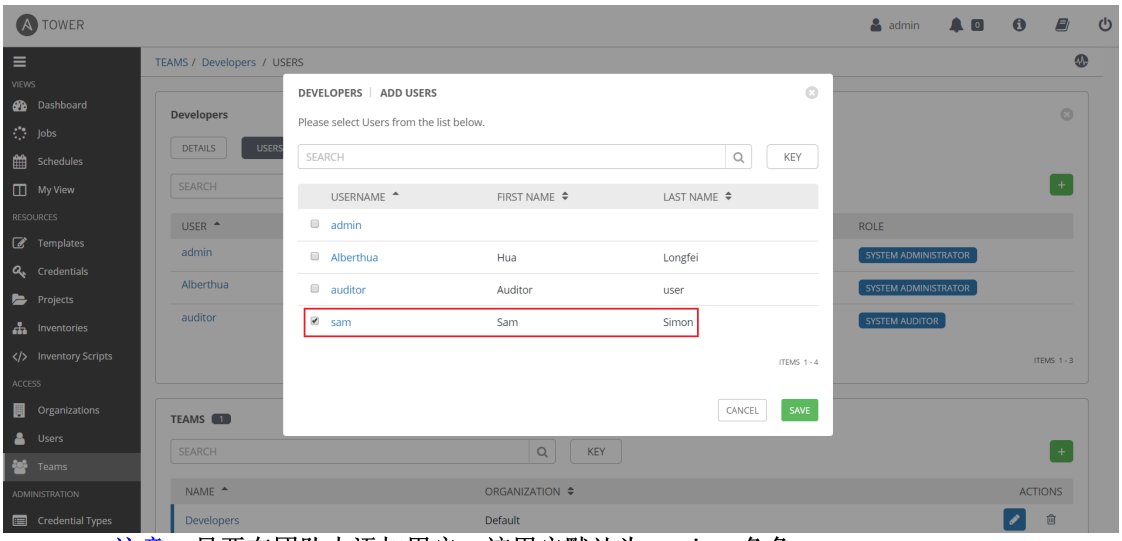
   * 注意：
     1. 在实践中，大多数组织不使用除 member 之外的团队角色。
     2. 团队成员资格通过外部身份验证进行管理，或者 Organization Administrator
        和 System Administrator 角色用于管理目的，System Auditor 用于审计要求，
        而不是对各个团队的read角色。
     3. 可赋予用户的3种角色类型：
        *organization role、team role、Ansible Tower resource role*

向团队添加用户：

- 完成团队创建后，可以向团队添加用户。
- 若要将拥有 member 角色的用户添加到组织中的团队，如下所示：

以 admin 用户身份，或者以团队所属组织的 admin 角色的用户，登录Ansible Tower Web界面。




* 注意：只要在团队中添加用户，该用户默认为member角色。

设置团队角色：

- 自红帽Ansible Tower 3.4起，向组织中的团队添加具有 admin 或 read 角色的用户需要

  tower-cli工具，这是红帽Ansible Tower REST API的命令行工具，或者直接使用红帽

  Ansible Tower API来添加。

- tower-cli role grant命令为用户授予团队 admin 或 read 角色。

```
[student@workstation ~]$ tower-cli role grant --user 'joe' \
> --target-team 'Operators' --type 'admin'
```

```
[student@workstation ~]$ tower-cli role grant --user 'jennifer' \
> --target-team 'Architects' --type 'read'
```

- tower-cli命令示例：

  1. 使用tower-cli时，需要运行 tower-cli config 命令来指定红帽Ansible Tower
     的主机，以及用于访问的用户名和密码。
  2. $ tower-cli config verify_ssl false：允许未验证的SSL连接

```
[root@ansible-tower ~]# tower-cli config verify_ssl False
Configuration updated successfully.
[root@ansible-tower ~]# tower-cli config username admin
Configuration updated successfully.
[root@ansible-tower ~]# tower-cli config password redhat
Configuration updated successfully.
[root@ansible-tower ~]# tower-cli config host 192.168.5.175
Configuration updated successfully.
[root@ansible-tower ~]# tower-cli login --password redhat admin
{
 "id": 2,
 "type": "o_auth2_access_token",
 "url": "/api/v2/tokens/2/",
 "created": "2020-02-24T14:31:04.292965Z",
 "modified": "2020-02-24T14:31:04.304887Z",
 "description": "Tower CLI",
 "user": 1,
 "refresh_token": null,
 "application": null,
 "expires": "3019-06-27T14:31:04.289771Z",
 "scope": "write"
}
Configuration updated successfully.
[root@ansible-tower ~]#
```

```
[root@ansible-tower ~]# tower-cli config

# User options (set with `tower-cli config`; stored in ~/.tower_cli.cfg).
verify_ssl: False
username: admin
password: redhat
oauth_token: 55Xl9xhqWteEiTqrsi8wLibQUsToJD
host: 192.168.5.175

# Defaults.
use_token: False
verbose: False
certificate:
format: human
color: True
description_on: False

[root@ansible-tower ~]# cat ~/.tower cli.cfg
[general]
verify_ssl = False
username = admin
password = redhat
oauth_token = 55Xl9xhqWteEiTqrsi8wLibQUsToJD
host = 192.168.5.175
```

2. 若在不同组织中存在相同名称的团队名，使用**tower-cli**命令时，需使用团队名**id**来

   唯一指定。

3. 否则即使指定了组织名，将依然报错！

```
[root@ansible-tower ~]# tower-cli role grant --user 'sam' --organization 'Datacenter.sh' --target-team 'Developers' --type 'read'
Error: Cannot look up team exclusively by name, because multiple team objects exist with that name.
Please send an ID. You can get the ID for the team you want with:
  tower-cli team list --name "Developers"
[root@ansible-tower ~]# tower-cli team list --name "Developers"
== ========== ============
id    name     organization
== ========== ============
 3 Developers            3
 2 Developers            1
== ========== ============
[root@ansible-tower ~]# tower-cli organization list
== =============
id     name
== =============
 3 Datacenter.sh
 1 Default
== =============
[root@ansible-tower ~]# tower-cli role grant --user 'sam' --organization 'Datacenter.sh' --target-team 3 --type 'read'
Error: You can only give a role to one type of resource at a time.
[root@ansible-tower ~]# tower-cli role grant --user 'sam' --target-team 3 --type 'read'
Resource changed.
== ==== ==== ===========
id user type target_team
== ==== ==== ===========
62    5 read N/A
== ==== ==== ===========
```

# 若在不同组织中存在同名的团队名称，需使用id进行指定，否则报错。

4. `$ tower-cli role list --user '<user_name>'`：查看用户的角色

```
[root@ansible-tower ~]# tower-cli role list --user 'daniel'
== ====== ============= ==============
id  type   resource_name resource_type
== ====== ============= ==============
11 Member Default       organization
45 Admin  Developers    team
== ====== ============= ==============
[root@ansible-tower ~]#
```