

## Ansible Tower 自动化实践（5） - 构建高级作业 workflow

### 第一节：通过事实缓存提高性能

#### 事实缓存：fact caching

- **Ansible**事实是**Ansible**在受管主机上自动检测到的变量。
- 通常，每个**play**在执行第一个任务之前会先自动运行 **setup** 模块，以便从匹配 **play** 的主机模式的每个受管主机收集事实。
- 但运行 **setup** 模块来收集每个**play**的事实具有明显的性能后果，特别是对于大型的受管主机清单。
- 可加速执行**play**的一种方式是在关闭自动事实收集。
- 可在**play**中设置 **gather\_facts: no** 来实现。
- 但若后续在**play**中使用事实，将无法执行此操作。
- **playbook**还可通过使用 “魔法” 变量（**magic variable**）**hostvars** 来引用其他主机的事实。
- 如，在受管主机 **servera** 上运行的任务可通过引用变量

**hostvars['serverb']['ansible\_facts']['default\_ipv4']['address']**  
来访问 **serverb**  
的事实 **ansible\_facts['default\_ipv4']['address']** 的值。

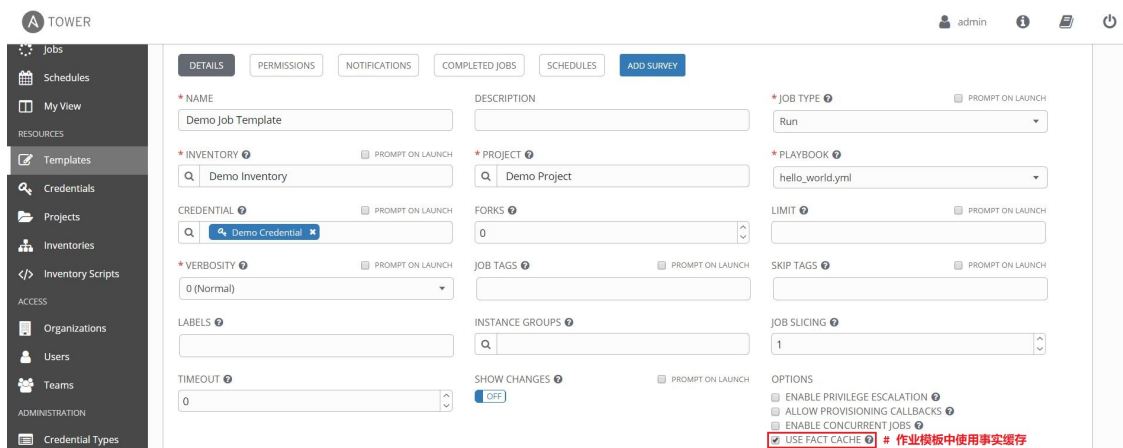
- 但这仅适用于此 **play** 或同一 **playbook** 中其更早的 **play** 已从 **serverb** 收集的事实。
- 可使用事实缓存解决这两个问题。
- 一个**playbook**可为清单中的所有主机收集事实并缓存这些事实，以便后续**playbook**可以在

不收集事实或手动运行 **setup** 模块的情况下使用它们。

#### 在Ansible Tower中启用事实缓存：

- **Ansible Tower 3.2**和更高版本包括对事实缓存的集成支持和用于事实缓存的数据库。
- 需在全球级别管理事实缓存的超时。
- 作业模板控制事实缓存（**fact caching**），即启用 **Use Fact Cache**。
- **Ansible Tower**中的每个主机都有一个控制事实何时过期的全局设置。
- 在 **Web UI** 的左侧导航栏中，选择 **Settings** 以显示 **Configure Tower** 窗格，然后单击 **JOBS**。
- **Per-Host Ansible Fact Cache Timeout** 设置控制缓存中的**Ansible**事实在收集后被视为  
有效的时间（单位为秒）。

- 默认值设为 0，表示缓存中存储的信息始终有效。
- 但若不定期收集用于更新缓存的事实，那么由于受管主机上的更改，将导致事实过时。
- 若要优化事实缓存，可将 `gather_facts: no` 设置为禁用play中的自动事实收集。
- 还需为使用包含这些play的playbook的任何Ansible Tower作业模板启用 Use Fact Cache。
- 然后，play依赖于事实缓存中的信息使用事实。
- 还需定期运行填充事实缓存的play，以使缓存的事实保持最新。
- 此 playbook 的 Ansible Tower 作业模板也需要启用 Use Fact Cache。



★ 注意：

1. 首次运行 `playbook` 时，需设置 `gather_facts: yes` 以开启事实收集，并启用作业模板中的 `Use Fact Cache` 实现事实缓存，当再次运行该 `playbook` 时可设置 `gather_facts: no`，以利用事实缓存。
2. 启动作业时，Ansible Tower 会将每个受管主机的所有 `ansible_facts` 从正在运行的作业注入到 `memcache` 中。

第二节：创建作业模板调查以设置作业的管理变量：

- 建议Ansible用户编写可在不同情况下重复使用的playbook，或者当部署到系统时应具有略微不同的行为、配置，或在不同环境中运行。
- 处理此问题的一种简单方法是使用变量。
- 变量可以由Ansible通过多种方式设置值，根据它们的设置方式，可以覆盖值。
- 如，某个角色可以为变量提供默认值，该变量又可能会被清单或playbook为该变量设置的值覆盖。
- 通常情况下，最好只在一个位置设置变量的值，以帮助避免出现变量优先级的问题。

- 使用`ansible-playbook`运行`playbook`时，用户可以通过两种方式来交互式地设置变量的值。
- 首先，可通过将 `-e` 或 `--extra-vars` 选项用于命令来传递额外的变量。
- 额外的变量始终优先。
- 或者，`playbook`中可能有 `vars_prompt` 部分，可以在运行`playbook`时以交互方式提示用户

进行输入。

- `vars_prompt` 变量设置的值比额外变量的优先级更低，可以被各种变量覆盖。

- 在Ansible Tower中，该方法的工作方式略有不同。
- 可通过作业模板来设置额外的变量，可在启动作业模板时提示用户设置，也可以通过重新运行

启动时已定义额外变量的作业来自动设置。

- Ansible Tower不支持包含 `vars_prompt` 问题的`playbook`。
- 最接近 `vars_prompt` 的替代方式是Ansible Tower的调查（survey）功能。

### 定义额外变量：extra variables

- 在Ansible Tower中，可通过两种方式使用作业模板来直接设置额外的变量：

1. 通过在作业模板的 EXTRA VARIABLES 字段中以 YAML 或 JSON 格式输入额外的变量

来设置。

2. 若为 EXTRA VARIABLES 字段选择了 PROMPT ON LAUNCH，则系统将提示Ansible Tower

用户在使用作业模板来启动作业时以交互方式修改所使用的额外变量的列表。

- 这些额外的变量与 `-e` 或 `--extra-vars` 选项为`ansible-playbook`指定的变量完全相似。
- 通过额外变量设置的值始终优先。
- 如下所示，作业模板中设置额外变量：

Labels ?

Instance Groups ?

Job Slicing ?

Timeout ?

Show Changes ?

Options

Extra Variables ?

YAML JSON

```

1 ---
2 msg_color: green
3 welcome_msg: '{{ ansible_fqdn }} is up'

```

LAUNCH CANCEL SAVE

Figure 10.3: Adding extra variables to a Job Template

DEMO JOB TEMPLATE

OTHER PROMPTS PREVIEW

EXTRA VARIABLES ?

YAML JSON

```

1 ---
2 msg_color: green
3 welcome_msg: '{{ ansible_fqdn }} is up'
4

```

CANCEL NEXT

Figure 10.4: Adjusting extra variables on job launch

- 若生成的作业之后重新启动，则会再次使用相同的额外变量，且该变量不能更改。
- 相反，应从原始作业模板启动作业，并设置不同的额外变量。

作业模板调查: survey

- 额外的变量可能难以使用，因为启动该作业的用户需要了解哪些变量可用，以及它们应当如何

与作业模板的playbook搭配使用。

- 作业模板调查允许作业模板在用于启动作业时显示简写形式，提示用户输入用于为额外变量

设置值的信息。

- 提示用户输入与设置额外变量相比具有多个优势：

1. 用户无需详细了解额外变量的运行方式或使用情况。
2. 无需了解playbook所使用的额外变量的名称。

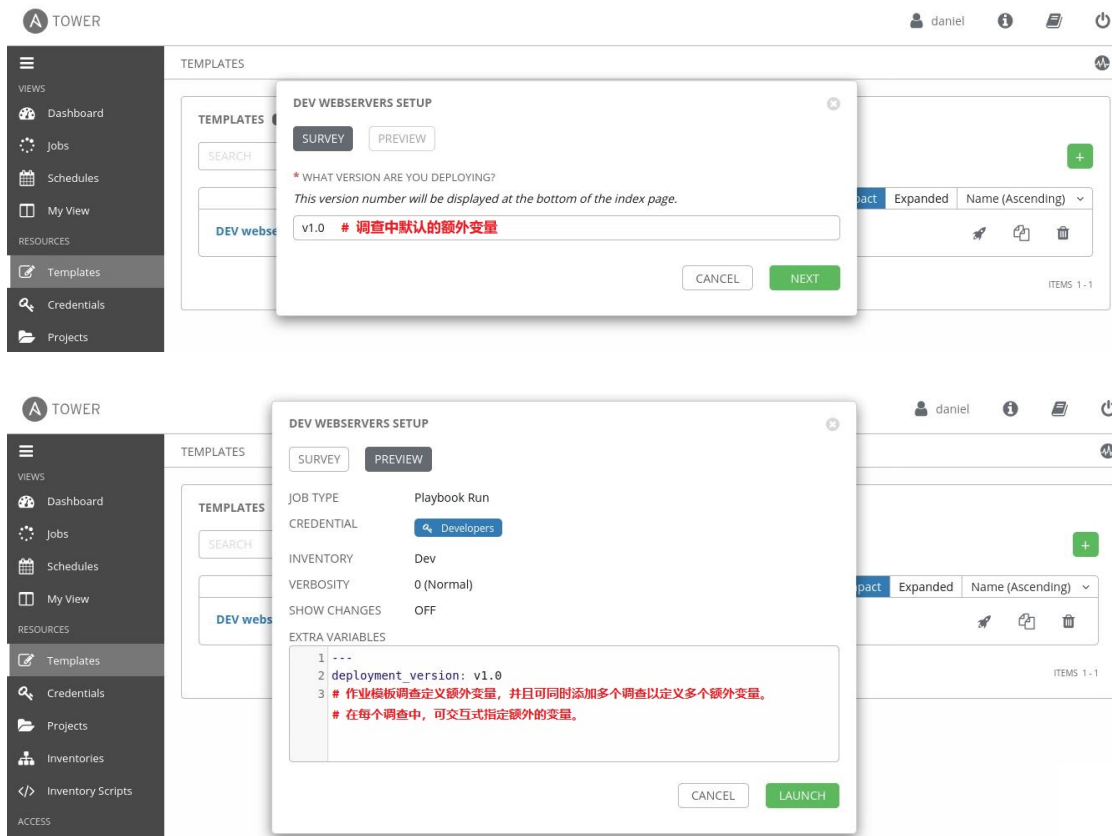
★ 注意：

1. 调查设置额外的变量，其优先级最高，会以任何其他方式覆盖同名的变量设置的值。
2. 包括作业模板的 EXTRA VARIABLES 字段或其 PROMPT ON LAUNCH 设置。
3. 调查和playbook中的 vars\_prompt 不能直接相互替代。
4. 通过 vars\_prompt 设置的变量比额外变量的优先级更低，会以多种方式被覆盖。

- 作业模板调查包含：用户友好的问题、回答类型、答案长度、默认回答

创建作业模板调查：

- 创建作业模板调查的过程，参看P368引导练习（略）。



### 第三节：创建工作流作业模板与启动工作流作业

#### 工作流作业模板：workflow job template

- 可按顺序运行多个**playbook**，而不是编写较大的**playbook**来自动化复杂的操作。
- 如，若要调配服务器，可能需要网络团队的**playbook**来分配IP地址到服务器并设置DNS记录，

再使用运维团队独立的**playbook**来安装和配置服务器的操作系统。

- 最后，将使用开发团队的**playbook**部署应用。
- 换句话说，需要遵循特定的工作流才能成功完成此过程。

- **Ansible Tower**中可让用户按顺序手动启动多个作业：

网络作业 -> 运维作业 -> 应用开发作业

- 最后，若其中某个**playbook**失败，可能需要运行其他**playbook**进行恢复。

- **Ansible Tower**支持工作流作业模板，它可将多个作业模板连接到工作流。
- 启动后，工作流作业模板将使用第一个作业模板启动作业，并根据它是否成功执行来确定是否要在下一步中启动作业模板。

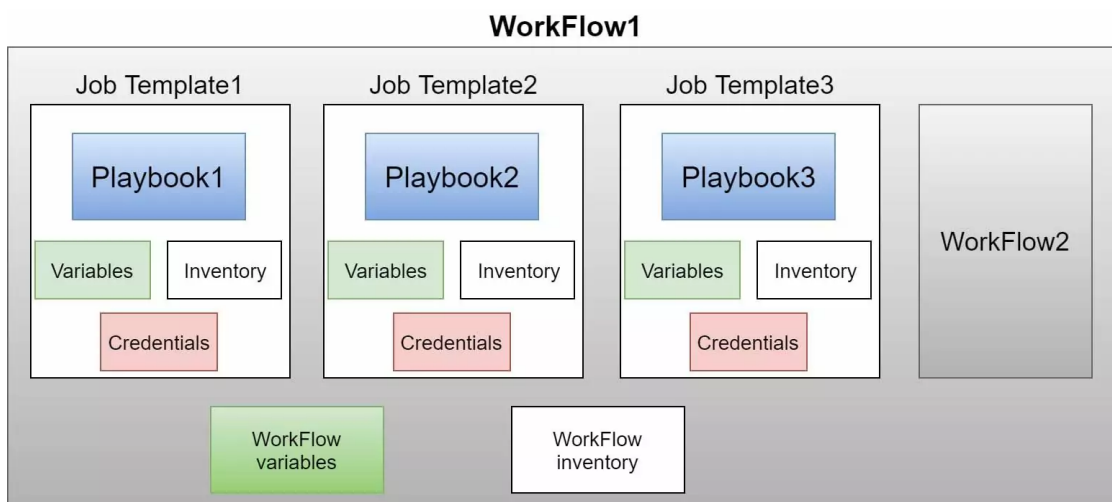
- 这允许启动一系列作业，并在作业失败时自动执行恢复步骤。

- 启动工作流作业模板的方式：

1. **Ansible Tower web UI** 手动启动
2. 作为计划的作业启动
3. 使用 **Ansible Tower API** 通过外部程序启动

- 工作流作业模板以串行方式运行作业模板，并使用图形工作流编辑器，将多个作业模板链接，

并根据上一个是否成功来运行不同的作业模板。



创建工作流作业模板：

- 需先创建工作流作业模板，然后才能定义工作流并与之关联。
- 可以通过或不通过组织来创建工作流作业模板。
- 在组织的上下文内创建工作流作业模板要求该用户拥有该组织的 **admin** 角色。

- 要创建不属于某个组织的工作流作业模板，需要单例 **System Administrator** 用户类型。
- 除作业模板外，还可以将同步项目（**P**）或同步清单（**I**）的作业合并到工作流中，分别在方框的左下角表示。
- 有助于确保在使用依赖于项目和清单资源的作业模板之前更新它们。

★ **注意：**

当调查添加到工作流作业模板时，所生成的额外变量可由工作流执行的每个作业访问。

**启动工作流作业：**

- 无论用户是否被分配单独作业模板的 **execute** 角色，只要用户对工作流作业模板具有 **execute** 角色就能执行该工作流作业模板。
- 创建与启动工作流作业模板的过程，参看P378引导练习（略）。

TEMPLATES / From Dev to Test

From Dev to Test

DETAILS PERMISSIONS NOTIFICATIONS COMPLETED JOBS SCHEDULES ADD SURVEY WORKFLOW VISUALIZER

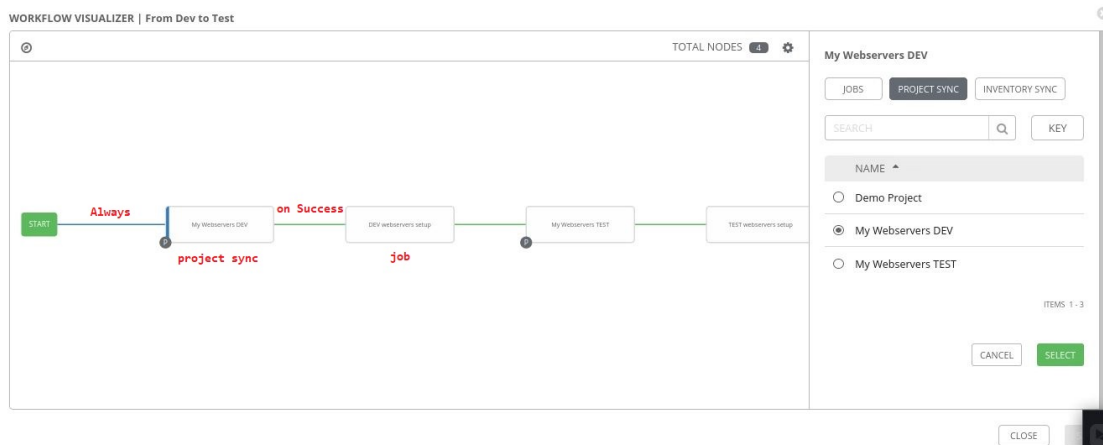
\* NAME From Dev to Test DESCRIPTION Deploy to Dev and on success deploy to test ORGANIZATION Default

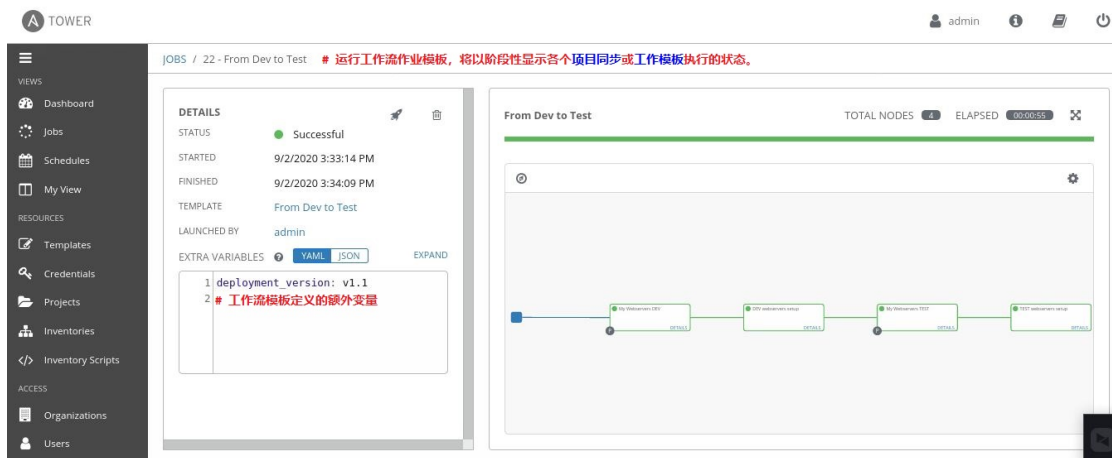
INVENTORY PROMPT ON LAUNCH LABELS OPTIONS

EXTRA VARIABLES YAML JSON PROMPT ON LAUNCH

1 ---  
2 deployment\_version: "v1.1" # 定义工作流作业模板的额外变量，将作用于其中的所有作业模板。

LAUNCH CANCEL SAVE





#### 第四节：调度作业与配置通知

调度作业执行：

- **Ansible Tower**允许用户配置调度的作业，按照可自定义的计划启动作业模板。

调度的管理作业：

- 默认情况下，**Ansible Tower**附带两个特殊的调度作业。
- 这两个计划适用于内置的管理作业，它们通过清理活动流和历史作业执行的旧日志信息，在**Ansible Tower**服务器上执行定期维护。

- **Cleanup Job Schedule:**

1. 删除历史作业的详细信息，以节省空间。
2. 默认情况下，在每周的星期天运行一次，以删除超过 120 天的作业信息，但可通过编辑计划来更改运行的时间和保留的数据量。

辑

- **Cleanup Activity Schedule:**

1. 默认情况下，在每周的星期二运行一次，以从活动流中删除超过 355 天的信息。
2. 可通过编辑日计划来更改运行的时间和保留的数据量。

报告作业执行结果：

- 使用**Ansible Tower**管理企业的**Ansible**基础架构的好处是集中式日志记录和审计。
- 执行作业时，有关作业执行的详细信息记录在**Ansible Tower**数据库中。
- 用户稍后可以应用此数据库来确定过去作业执行的历史结果。
- 历史作业执行详细信息有助于管理员确认调度和委派的作业执行是成功还是失败。

- 检索历史作业执行详细信息的功能很有帮助，但对于与关键功能相关的作业，管理员可能希望立即通知作业是成功还是失败。

- **Ansible Tower**可立即发送作业执行结果的警报。
- 若要使用此功能，管理员可以创建通知模板（**notification template**）来定义通知的发送方式。



- **Ansible Tower**支持多种发送通知的机制。
- 一些机制基于开放协议（如电子邮件和 **IRC**），另一些则基于专有解决方案（如 **HipChat** 和 **Slack**）。

#### 通知模板：

- 通知模板在组织的上下文中定义。
- 在创建后，通知模板可用于发送由**Ansible Tower**为该组织运行的作业结果的通知。
- 通知模板定义发送通知的机制。
- 支持的机制包括：
  - 电子邮件
  - **Slack**
  - **Twilio**
  - **PagerDuty**
  - **HipChat**
  - **Webhook**
  - **IRC**

#### 创建通知模板：

- 创建与启用通知模板的过程，参看**P386**引导练习（略）。