02

Simplificando o playbook with_items

Transcrição

Nós criamos o primeiro playbook de Ansible e aprendemos como é feito. Ele foi usado para instalar o PHP e o Apache na máquina virtual. A seguir, veremos como melhorar o playbook, como criar o primeiro loop e tornar nosso código mais eficiente com menos linhas. Também veremos como organizar melhor o inventário e com isso, usaremos comandos menores.

Atualmente, `provisioning.yml` está assim:

```
- hosts: all
  tasks:
    - name: 'Instala o PHP5'
      apt:
        name: php5
        state: latest
      become: yes
    - name: 'Instala o Apache2'
      apt:
        name: apache2
        state: latest
      become: yes
    - name: 'Instala o modphp'
      apt:
        name: libapache2-mod-php5
        state: latest
      become: yes
```

Observe que temos três tasks que instalam pacotes usando `apt` , a maior diferença entre eles é o nome e comentário. Para fazermos melhorias, adicionaremos uma nova task que será responsável por instalar os `pacotes de dependência do sistema operacional` . Preencheremos `item` com a palavra reservada `item` , depois, vamos incluir o loop usando o parâmetro `with_items` que ficará no nível da task (ou seja, não faz parte da mesma).

```
- hosts: all
  tasks:
    - name: 'Instala pacotes de dependencia do sistema operacional'
      apt:
        name: "{{ item }}"
        state: latest
      become: yes
      with_items:
        - php5
        - apache2
        - libapache2-mod-php5

    - name: 'Instala o PHP5'
      apt:
        name: php5
        state: latest
      become: yes
```

```
    with_items:
      - php5
      - apache2
      - libapache2-mod-php5

  - name: 'Instala o Apache2'
    apt:
      name: apache2
      state: latest
    become: yes
  - name: 'Instala o modphp'
    apt:
      name: libapache2-mod-php5
      state: latest
    become: yes
```

Usamos a estrutura das tasks anteriores, colocamos o nome dos pacotes nessa lista, integrante do `with_items` . E cada entrada será um item substituído onde citamos a palavra reservada `item` , uma variável especial que o Ansible cria na execução dos loops. Agora podemos remover o restante do código, após as alterações teremos o seguinte:

```
- hosts: all
  tasks:
    - name: 'Instala pacotes de dependencia do sistema operacional'
      apt:
        name: "{{ item }}"
        state: latest
      become: yes
      with_items:
        - php5
        - apache2
        - libapache2-mod-php5
```

Rodaremos novamente o comando de execução no terminal, para verificar se o código continua a funcionar.

```
$ ansible-playbook provisioning.yml -u vagrant -i hosts --private-key .vagrant/machines/wordpres

PLAY [all] *********************************************************************

TASK [Gathering Facts] *********************************************************
ok: [172.17.177.40]

TASK [Instala pacotes de dependencia do sistema operacional] *******************
ok: [172.17.177.40] => (item=[u'php5', u'apache2', u'libapache2-mod-php5'])

PLAY RECAP *********************************************************************
172.17.177.40                  : ok=2    changed=0    unreachable=0    failed=0
```

Veremos que o Ansible vai conseguir encontrar todos os pacotes instalados, não sendo necessário fazer uma nova instalação. A seguir, experimentaremos adicionar um novo pacote e conferir se o que afirmamos é real. Incluiremos o `php5-gd` logo abaixo de `libapache2-mod-php5` .

```
- hosts: all
  tasks:
    - name: 'Instala pacotes de dependencia do sistema operacional'
      apt:
        name: "{{ item }}"
        state: latest
      become: yes
      with_items:
        - php5
        - apache2
        - libapache2-mod-php5
        - php5-gd
```

Ao executarmos o comando no terminal, seremos informados sobre a instalação do `php5-gd` .

```
$ ansible-playbook provisioningg.yml -u vagrant -i hosts --private-key .vagrant/machines/wordpre

PLAY [all] *********************************************************************

TASK [Gathering Facts] *********************************************************
ok: [172.17.177.40]

TASK [Instala pacotes de dependencia do sistema operacional] *******************
ok: [172.17.177.40] => (item=[u'php5', u'apache2', u'libapache2-mod-php5', u'php5-gd'])

PLAY RECAP *********************************************************************
172.17.177.40                  : ok=2    changed=0    unreachable=0    failed=0
```

Conseguimos fazer com que os comandos para instalação e administração de pacote ficassem **mais curtos**. Após a alteração, é possível administrar os pacotes com apenas uma task. Simplificamos nosso trabalho e facilitamos também o processo de incluir o restante das nossas dependências.

O próximo passo será escrever no código, quais dependências serão necessárias para o funcionamento do PHP - estas são solicitadas pelo Wordpress, que usa diversas extensões da linguagem. Precisaremos ainda instalar o servidor mySQL(`mysql-server-5.6`) e a integração de Python e MySQL (`python-mysqldb`), a última é pedida pelo Ansible e utilizada na execução de alguns módulos na máquina que será configurada.

```
- hosts: all
  tasks:
    - name: 'Instala pacotes de dependencia do sistema operacional'
      apt:
        name: "{{ item }}"
        state: latest
      become: yes
      with_items:
        - php5
        - apache2
        - libapache2-mod-php5
        - php5-gd
        - libssh2-php
        - php5-mcrypt
        - mysql-server-5.6
```

```
    - python-mysqldb
    - php5-mysql
```

Com essas dependências, nós conseguimos fazer o Ansible controlar o servidor que faremos a seguir, e teremos um ambiente completo de PHP, MySQL e Apache. Tudo será administrado por um único comando, evitando escrever diversas linhas por dependência.

Em seguida, executaremos o comando novamente, que deve demorar para fazer o download dos pacotes na sua máquina. No meu caso, já tinha feito a instalação para adiantar o processo. O retorno no meu terminal será:

```
$ ansible-playbook provisioning.yml -u vagrant -i hosts --private-key .vagrant/machines/wordpres

PLAY [all] ********************************************************************************

TASK [Gathering Facts] ********************************************************************
ok: [172.17.177.40]

TASK [Instala pacotes de dependencia do sistema operacional] ******************************
ok: [172.17.177.40] => (item=[u'php5', u'apache2', u'libapache2-mod-php5', u'php5-gd', u'libssh2

PLAY RECAP ********************************************************************************
172.17.177.40                  : ok=2    changed=0    unreachable=0    failed=0
```

Temos um ambiente completo, no qual é possível administrá-lo de forma mais organizada. Deixamos o playbook com seu conteúdo bem disponibilizado, sem repetições de tasks com a mesma utilidade, e se precisarmos incluir ou remover uma dependência, resolveremos com uma simples alteração na listagem.

O que aprendemos como fazer substituição de variável, um tópico que nos aprofundaremos mais adiante. A seguir, veremos como usar o arquivo de inventário para simplificar a maneira como nos comunicamos com o Ansible e caso seja preciso trabalhar com um ambiente heterogêneo, com diferentes usuários e formas de autenticações em diversas máquinas.