Speed up your Ansible Playbooks

Eloy Coto Pereiro on May 08th, 2015

I'm always impressed with my friends of Streetlife. They built a new infrastructure based on immutable deploys. I'm surprised about how fast they are able to build this AMIs, they can start a new full configured server in only 6 min.

Three years ago I've started using Ansible. My deploys **were** always up to twenty minutes, so one day I decided to achieve an eight-min deploy. Eight min will be the time that I have to build a new server.

To get this eight minute build, I focused on the following tasks:

APT cache time:

When you have different roles in Ansible, usually you always set *update_cache* in the apt task. This is pretty useful, but painful with different roles.

Ansible provides the cache_valid_time option. If you enable it, Ansible will check the cache age. If cache age is less than your value, apt-get update will be ignored.

```
- name: Installing useful packages
  apt:
    name: "{{ item }}"
    update_cache: yes
    cache_valid_time: 3600
  with_items:
    - htop
    - ngrep
    - vim
```

Async Tasks:

Async tasks are in Ansible from version 1.7. It works like this: while one long task is running, another short task can be executed.

For example, if you want to install pip dependencies and bower dependencies: both are needed, both can run at the same time and both take few minutes.

With async tasks, tasks can be executed and forget, but this background task can be checked later. So Ansible provides the option to get the task status in any time.

The following example show how pip and bower dependencies will run in two new coroutines. While the dependencies are being installed in the system, another task will create the users (or any other task). Before the playbook is finished, it's going to be checked if the coroutines has finished properly.

```
- name: bower install requirements (Coroutine #2)
  bower:
    path: /my_app/frontend/
    state: latest
  async: 1000
  poll: 0
  register: bower_install

- name: Add the users in the platform
  user:
    name: "{{item.name}}"
    shell: /bin/bash
    groups: "{{item.groups}}"
    state: "{{item.status}}"
  with_items:
    - {name: eloy, state: present, groups: admin}
    - {name: www, state: present, groups: admin}
    - {name: ftp, state: present, groups: admin}

- name: PIP result check (Check coroutine #1)
  async_status:
    jid: "{{ pip_install.ansible_job_id }}"
  register: job_result
  until: job_result.finished
  retries: 30

- name: Bower result check (Check coroutine #2)
  async_status:
    jid: "{{ bower_install.ansible_job_id }}"
  register: job_result
  until: job_result.finished
  retries: 30
```

Ansible async task and loops:

When you start with Ansible, you use a lot of *with_items*. Loops are not supported in Async tasks, but the following workaround can be used:

```
# vars.yml
dependencies:
  - bison
  - gcc
  - git
  - make
  - mercurial
```

```
- name: Installing dependencies
  apt:
    name: "{{ ','.join(dependencies) }}"
    update_cache: yes
    cache_valid_time: 3600
  async: 1000
  poll: 0
```

Happy deploy!

Share this:

Tags

ansible async debian deploy devops

Comments:

4 Comments Acalustra 1 Login

Recommend 2 Share Sort by Best

Join the discussion…

LOG IN WITH OR SIGN UP WITH DISQUS ?

Name

Adam Chainz • 3 years ago
Thanks! I'm building AMIs too for deployment and I have some things that could be parallelized. Didn't realize async could do that when it came out - I thought it was only for long-running tasks that might time out the connection. Thanks for the relevant example!
1 • Reply • Share ›

Michael Kubler • a year ago
So you got to 8min builds? I'm looking at implimenting your suggetions. Have added the apt cache_valid_time which will be good.

I found your post because I was looking at async tasks, I've got a mercurial clone that takes AGES and is definitely up for being converted into an async task, probably even a fire and forget version.

Cheers!
• Reply • Share ›

eloycoto Mod Michael Kubler • a year ago
Hey, I'm using vsphere/packer that makes the process a bit slow. But mainly our deploys are fast. I use to work with async task on packages install (APT/YUM or pip) or git clones and some bower/npm. Highly recommend it.

Regards
1 • Reply • Share ›

goldyfruit • 2 years ago
Very nice, thanks for the explanations and the examples.
• Reply • Share ›

DISQUS

Related articles:

- Ansible: how to maintain multiple ssh-keys in different servers
- trick and tips en ansible