Raymii.org ⎼⋀⋀⋀⎼
(https://raymii.org/s/)

Quis custodiet ipsos custodes?

RSS
Feed(https://raymii.org/s/feed.xml)
|
Gopher(gopher://raymii.org:70)

> Search

Home(/s/index.html)

All
Pages(/s/tags/all.html)

refcode=7435ae6b8212)

Ansible - Only do action if on specific distribution (Debian, Ubuntu, CentOS or RHEL) or distribution version (ubuntu precise, ubuntu trusty)

Home(../index.html) / Tutorials(../tags/tutorials.html)
 / Ansible - Only do action if on specific distribution (Debian, Ubuntu, CentOS or RHEL) or distribution
version (ubuntu precise, ubuntu trusty)(Ansible_-
_Only_if_on_specific_distribution_or_distribution_version.html)

16-12-2018 | Remy van Elst | Text only version of this article(Ansible_-
_Only_if_on_specific_distribution_or_distribution_version.txt)

Table of Contents

This Ansible playbook example helps you execute actions only if you are on a certain distribution. You might
have a mixed environment with CentOS and Debian and when using Ansible to execute actions on nodes you
don't need to run Yum on Debian, or Apt on CentOS. Some package names are different and such, so this helps
you with an only if statement to select a specific distribution. As a bonus, you also get an `only_if` for specific
distribution versions, like Ubuntu precise (12.04 LTS) or Ubuntu Trusty (14.04 LTS).

If you like this article, consider sponsoring me by trying out a Digital Ocean VPS. With this link you'll get a $5 VPS
for 2 months free (as in, you get $10 credit). (referral link)(https://www.digitalocean.com/?
refcode=7435ae6b8212)

* 2018-12-16: update ansible syntax to version 2.5, use become

- 2015-09-24: Added `package` module, changed only_if to when
- 2014-09-11: Initial release

Specific Distribution

On a specific action, add the following `when` statement:

```
when: ansible_distribution == 'CentOS' or ansible_distribution == 'Red Hat Enterprise Linux'
```

This is for RHEL and Centos, the following is for Debian/Ubuntu:

```
when: ansible_distribution == 'Debian' or ansible_distribution == 'Ubuntu'
```

This example playbook installs `Apache2` on both Debian/Ubuntu and CentOS. This example used apache because the name package name is different on the two distributions.

```
---
- hosts: example
  become: true
  user: remy
  connection: ssh

  tasks:
  - name: Install apache
    apt:
      name: {{ item }}
      state: latest
    with_items:
     - apache2
    when: ansible_distribution == 'Debian' or ansible_distribution == 'Ubuntu'

  - name: Install httpd
    yum:
      name: {{ item }}
      state: latest
    with_items:
     - httpd
    when: ansible_distribution == 'CentOS' or ansible_distribution == 'Red Hat Enterprise Linux'

  - name: restart apache
    service:
      name: apache2
      state: started
      enabled: yes
    when: ansible_distribution == 'Debian' or ansible_distribution == 'Ubuntu'

  - name: restart httpd
    service:
      name: httpd
      state: started
      enabled: yes
    when: ansible_distribution == 'CentOS' or ansible_distribution == 'Red Hat Enterprise Linux'
```

Specific Distribution Version

You might also need to do different actions based on distribution version, because some things are available on CentOS 6 but not on 5, or on Ubuntu Lucid you need to install some backported packages and not on Ubuntu Precise.

For those situations, you can use either the `{{ ansible_distribution_version }` or `{{ ansible_distribution_release }}` variable. See some example output from `ansible all -m setup -a "filter=ansible_distribution*"` :

```
    "ansible_distribution": "CentOS",
    "ansible_distribution_release": "Final",
    "ansible_distribution_version": "5.9"

    "ansible_distribution": "CentOS",
    "ansible_distribution_release": "Final",
    "ansible_distribution_version": "6.4"

    "ansible_distribution": "Ubuntu",
    "ansible_distribution_release": "lucid",
    "ansible_distribution_version": "10.04"

    "ansible_distribution": "Ubuntu",
    "ansible_distribution_release": "precise",
    "ansible_distribution_version": "12.04"

    "ansible_distribution": "Debian",
    "ansible_distribution_release": "wheezy",
    "ansible_distribution_version": "7"
```

Using these, you can filter the output by changing the `when` statement in your ansible playbook:

```
 when: ansible_distribution == 'CentOS' and ansible_distribution_version == '6.4'

 when: ansible_distribution == 'Ubuntu' and ansible_distribution_release == 'precise'

 when: ansible_distribution == 'Debian' and ansible_distribution_version == '7'

 when: ansible_os_family == "RedHat" and ansible_lsb.major_release|int >= 5
```

Package module (2015 short update)

As my former colleague Stein pointed me to, Ansible 2.0 has been released and it features the `package` module. This is a generic module that installs, upgrade and removes packages using the underlying OS package manager. This module actually calls the pertinent package modules for each system (apt, yum, etc).

This means that if you use this article because you want a package install on Debian and CentOS, you can now just do the following:

```
 - name: install (or upgrade to) the latest version of htop
   package:
     name: htop
     state: latest
```

If a package has different names on different distributions, like Apache (apache2 on ubuntu, httpd on CentOS) you still need to use a when statement.

Read more about the package module on the ansible Docs website(https://docs.ansible.com/ansible/package_module.html)

Tags: ansible (../tags/ansible.html)apt (../tags/apt.html)configuration-management (../tags/configuration-management.html)deb (../tags/deb.html)deployment (../tags/deployment.html)devops (../tags/devops.html)ntp (../tags/ntp.html)packages (../tags/packages.html)python (../tags/python.html)tutorials (../tags/tutorials.html)yum (../tags/yum.html)

Generated by ingsoc(/s/software/ingsoc.html) | Cluster Status(/s/software/Sparkling_Network.html) | About(/s/static/About.html)