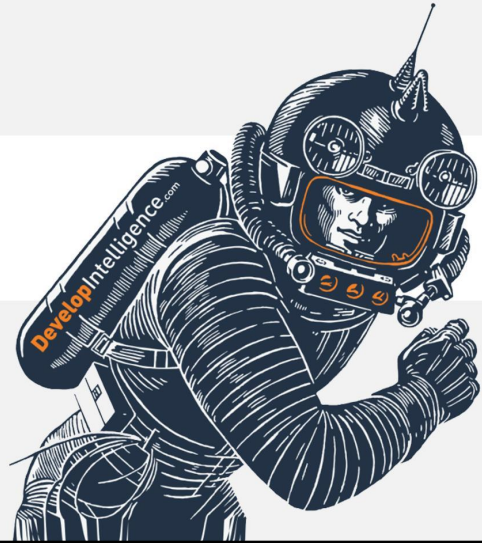


Playbooks

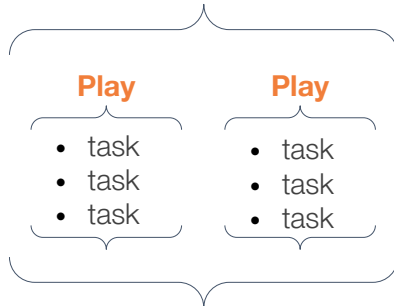




- More on Variables
- Handlers
- Loops
- Conditionals
- Review



Playbook



```
$ ansible-playbook -i hosts playbook.yml
```

```
---
- hosts: webservers
  remote_user: root

  tasks:
  - name: ensure apache is at the latest version
    yum:
      name: httpd
      state: latest
  - name: write the apache config file
    template:
      src: /srv/httpd.j2
      dest: /etc/httpd.conf

- hosts: databases
  remote_user: root

  tasks:
  - name: ensure postgresql is at the latest version
    yum:
      name: postgresql
      state: latest
  - name: ensure that postgresql is started
    service:
      name: postgresql
      state: started
```

This example playbook has 2 plays, each with 2 tasks. The first play runs against any hosts in the *webservers* group while the second runs against hosts in the *databases* group.



1. extra vars (always wins precedence)
2. include params
3. role (and include_role) params
4. set_facts / registered vars
5. include_vars
6. task vars (only for the task)
7. block vars (only for tasks in block)
8. role vars (defined in role/vars/main.yml)
9. play vars_files
10. play vars_prompt
11. play vars
12. host facts / cached set_facts
13. playbook host_vars/*
14. inventory host_vars/*
15. inventory file or script host vars
16. playbook group_vars/*
17. inventory group_vars/*
18. playbook group_vars/all
19. inventory group_vars/all
20. inventory file or script group vars
21. role defaults
22. command line values (eg "-u user")

https://docs.ansible.com/ansible/latest/user_guide/playbooks_variables.html#variable-precedence-where-should-i-put-a-variable



1. extra vars (--extra-vars in CLI)

In general, the more specific the higher the precedence.

6. task vars (only for the task)

15. inventory file or script host vars

8. role vars (defined in role/vars/main.yml)

19. inventory group_vars/all

11. play vars

22. command line values (eg "-u user")



```
lamp/  
  playbook.yml  
  hosts  
  group_vars/  
  host_vars/  
  
  roles/  
  library/
```

https://docs.ansible.com/ansible/latest/user_guide/playbooks_best_practices.html#directory-layout



- Run at the end of the play, if notified
- Run exactly once, even if notified

multiple times

```
tasks:
- name: template configuration file
  template:
    src: template.j2
    dest: /etc/foo.conf
  notify:
    - restart memcached
    - restart apache
handlers:
- name: restart memcached
  service:
    name: memcached
    state: restarted
- name: restart apache
  service:
    name: apache
    state: restarted
```



```
- name: return motd to registered var
  command: cat /etc/motd
  register: mymotd

- name: print return code of previous command
  debug:
    var: mymotd.rc

- name: print contents of /etc/motd
  debug:
    var: mymotd.stdout
```




- A task can be performed multiple times with differing values of a special "item" variable
- Use the **loop** (new way) or **with_items** (old way) directives

```
---  
- hosts: webservers  
  remote_user: root  
  
tasks:  
- name: install packages  
  apt:  
    name: "{{ item }}"  
    state: present  
  loop:  
    - python  
    - git  
    - apache  
  
- name: install packages  
  apt:  
    name: "{{ item }}"  
    state: present  
  with_items: # The old way  
    - python  
    - git  
    - apache
```



```
---
- hosts: webservers
  remote_user: root

  tasks:
    - name: install packages
      apt:
        name: "{{ item }}"
        state: present
      loop:
        - python
        - git
        - apache
      when: ansible_facts['distribution'] == "Ubuntu"

    - name: install packages
      yum:
        name: "{{ item }}"
        state: present
      loop:
        - python
        - git
        - apache
      when: ansible_facts['distribution'] == "CentOS"
```

```
---
- hosts: webservers
  remote_user: root

  tasks:
    - command: /bin/false
      register: result
      ignore_errors: True

    - command: /bin/something
      when: result is failed

    - command: /bin/something_else
      when: result is succeeded

    - command: /bin/still/something_else
      when: result is skipped
```

https://docs.ansible.com/ansible/latest/user_guide/playbooks_conditionals.html



Additional Resources



- Playbook Best Practices
https://docs.ansible.com/ansible/latest/user_guide/playbooks_best_practices.html