Contents

Ansible TL;DR

Ansible Core

- command line tool
- connects over ssh or winrm to target machine(s)
- python 2.4+ required on target systems
- adhoc command
  ```
  ansible localhost -m ping
  ```
- playbook command
  ```
  ansible-playbook playbook.yml
  ```
- ansible-doc is a **key command**
  ```
  ansible-doc -l | more
  ansible-doc user
  ```

Ansible Flow

- adhoc or playbook executed
- ssh (or winrm) connection to target
- gather facts (playbook)
- playbook sent to target
- playbook executes

Ansible Terminology

- Ansible Controller/Server: Server where Ansible is installed
- Target Server(s): The systems to be managed
- Playbook: Combination of Target Servers, Vars and Plays/Tasks
- Task: procedure to be completed on client
- Module: Command or set of commands executed on client
  - yum, apt, dnf, command etc...
- Role: way to organise tasks/files for reuse
  ```
  mkdir roles
  ansible-galaxy role init ./roles/package
  ```
- Fact: information retrieved from client
- Inventory: file containing data about clients
  ```
  ansible-1 ansible_host=192.100.5 ansible_user=devops
  ```
- Play: execution of tasks against clients
- Handler: task called when notified

---

ansible.cfg

- determines how ansible will behave
- specify location of inventory file
- privilege escalation
- private key location
- ssh fine tuning
- check configuration
  ```
  ansible-config list
  ansible-config dump
  ```
  - output in orange default overriden

Ansible Adhoc

- examples
  ```
  ansible all -m ansible.builtin.setup
  ansible all -m ansible.builtin.setup -a 'filter=ansible_*_mb'
  ansible web -m command -a "ls -l /etc/ansible/facts.d/"
  ansible web -m ansible.builtin.copy -a "src=/etc/hosts dest=/tmp/hosts"
  ansible web -m ansible.builtin.file -a "dest=/srv/foo/a.txt mode=600"
  ansible ansible-1 -m ansible.builtin.yum -a "name=acme state=present"
  ansible all -m ansible.builtin.user -a "name=foo password=<crypted password here>"
  ansible webservers -m ansible.builtin.service -a "name=httpd state=started"
  ```

Ansible playbook

- Simple example playbook
  ```
  - name: **This is a play in a playbook**
    hosts: web
    tasks:
    - name: **This is a Task in a play**
      # this is a module
      ansible.builtin.file:
        path: /tmp/testfile.txt
        state: touch
  ```

Playbook parts

- name: A description file for play or task
- hosts: Where the task will execute
- tasks: section used to order commands and states
  ```
  - name: This is a Task in a play
    file:
      path: /tmp/testfile.txt
      state: touch
  ```
  - file: Ansible module used to create file
  - state: what the end result should be for task

Playbook move to Roles

```
mkdir roles
ansible-galaxy role init ./roles/package
```

- copy tasks from playbook to roles/role_name/tasks folder
- copy vars from playbook to roles/role_name/vars or defaults folder

- copy handler from playbook to roles/role_name/handles folder
- copy files to roles/role_name/files folder
- copy templatess to roles/role_name/templates folder
- update roles/role_name/meta/main.yml
- create site.yml to call the role(s)

Playbook using Roles

- Using Roles

  ```
  - name: Install and start Apache HTTPD
    hosts: web
    roles:
      - package
      - index
  ```

  better to use include_role or import_role than role

Benefits

- Agentless
- flexible - orchestrate end to end
- Efficient - no additional software
- Powerful - model complex flows
- Fast - built on top of python
- free

Ansible Tower / Automation Controller

- Grapical User Interface
- RBAC Access
- Model complex Business processes
- License Required (AWX is upstream)
- Scale automation beyond single engineer
- Moving to Container based execution

This is what we will see
