Ansible IT automation

CONCEPTS

Ansible connects over SSH and needs no agent. Playbooks describe a desired state, so running one twice should change nothing the second time.

Inventory hosts and groups
↓ pattern
Playbook plays and tasks
↓ SSH
Modules run on targets
↓ result
Report ok / changed / failed

Prefer modules over raw shell: they report changed correctly and are idempotent.

AD-HOC

ansible all -m ping
Test connection and Python on hosts

ansible *<pattern>* **-m command -a "***<cmd>***"**
Run one command on hosts

ansible *<pattern>* **-m shell -a "***<cmd>***"**
Run through a shell

ansible *<pattern>* **-b -m apt -a "name=***<pkg>* **state=present"**
Install a package as root

ansible *<pattern>* **-m setup**
Gather facts

ansible *<pattern>* **-m copy -a "src=***<f>* **dest=***<path>***"**
Copy a file to hosts

ansible *<pattern>* **-m service -a "name=***<svc>* **state=restarted"**
Restart a service

ansible *<pattern>* **--list-hosts**
Show which hosts match

PLAYBOOKS

ansible-playbook *<file>*
Run a playbook

ansible-playbook *<file>* **-i** *<inventory>*
Use a specific inventory

PLAYBOOKS (cont.)

ansible-playbook *<file>* **--check**
Dry run, change nothing

ansible-playbook *<file>* **--diff**
Show file changes

TARGETING & VARIABLES

-l *<pattern>* **/ --limit** *<pattern>*
Restrict to hosts or groups

-t *<tag>* **/ --tags** *<tag>*
Run only tagged tasks

-e *<key>*=*<value>*
Set an extra variable

PRIVILEGES & CONNECTION

-b / --become
Escalate privileges

-K / --ask-become-pass
Prompt for the sudo password

INVENTORY

ansible-inventory --graph
Tree of groups and hosts

VAULT

ansible-vault create *<file>*
New encrypted file

ansible-vault encrypt *<file>*
Encrypt an existing file

ansible-vault edit *<file>*
Edit an encrypted file

GALAXY

ansible-galaxy collection install *<ns.name>*
Install a collection

ansible-galaxy collection install -r requirements.yml
Install from a requirements file

DOCS & CONFIG

ansible-doc *<module>*
Module documentation