Ansible IT automation

START HERE

`ansible all -m ping`
Can I reach every host?

`ansible <group> -m command -a "<cmd>"`
Run a command

`ansible-inventory --graph`
Show groups and hosts

`ansible-playbook <file>`
Run a playbook

`ansible-playbook <file> --check --diff`
Dry run with changes

`ansible-playbook <file> --limit <host>`
Only some hosts

`ansible-vault encrypt <file>`
Protect secrets

`ansible-galaxy collection install <name>`
Get a collection

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

CONCEPTS

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

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

Make Ansible output reproducible: pin the inventory with -i and keep ansible.cfg in the project directory.

AD-HOC

`ansible all -m ping`
Test connection and Python on hosts
Uses the ping module, not ICMP.

`ansible <pattern> -m command -a "<cmd>"`
Run one command on hosts
The default module is command; shell features like pipes need -m shell.

`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
Add -a "filter=ansible_distribution" to narrow the output.*

`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

`ansible <pattern> -f <n>`
Parallel forks

PLAYBOOKS

`ansible-playbook <file>`
Run a playbook

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

`ansible-playbook <file> --check`
Dry run, change nothing
Some modules cannot predict their result in check mode.

PLAYBOOKS (cont.)

`ansible-playbook <file> --diff`
Show file changes
Combine with --check for a preview.

`ansible-playbook <file> --syntax-check`
Parse only

`ansible-playbook <file> --list-tasks`
List tasks that would run

`ansible-playbook <file> --list-hosts`
List targeted hosts

`ansible-playbook <file> --list-tags`
List available tags

`ansible-playbook <file> --step`
Confirm each task

`ansible-playbook <file> --start-at-task "<name>"`
Resume at a task

TARGETING & VARIABLES

`-l <pattern>` / `--limit <pattern>`
Restrict to hosts or groups
Patterns: web:db is a union, web:&prod an intersection, web:!old an exclusion.

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

`--skip-tags <tag>`
Skip tagged tasks

`-e <key>=<value>`
Set an extra variable
Extra vars have the highest precedence.

`-e @<file>`
Load variables from a file

`-v` / `-vvv`
More output detail
Each extra v adds detail; -vvvv shows connection debugging.

`--ask-vault-pass`
Prompt for the vault password

PRIVILEGES & CONNECTION

`-b` / `--become`
Escalate privileges

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

`--become-user <user>`
Become a user other than root

`-u <user>`
Remote login user

PRIVILEGES & CONNECTION (cont.)

`-k` / `--ask-pass`
Prompt for the SSH password

`--private-key <file>`
SSH key to use

`-c local`
Run on the control machine

INVENTORY

`ansible-inventory --graph`
Tree of groups and hosts

`ansible-inventory --list`
Full inventory as JSON

`ansible-inventory --host <host>`
Variables of one host

`ansible-inventory -y --list`
Inventory as YAML

Variable precedence (low to high)

Source	Wins over
role defaults	nothing
inventory vars	defaults
play vars	inventory
extra vars (-e)	everything

VAULT

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

`ansible-vault encrypt <file>`
Encrypt an existing file
Encrypts in place.

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

`ansible-vault view <file>`
Show decrypted content

`ansible-vault decrypt <file>`
Decrypt in place

`ansible-vault rekey <file>`
Change the vault password

`ansible-vault encrypt_string '<s>' --name <var>`
Encrypt a single value

`--vault-password-file <file>`
Read the password from a file
The file must not be committed to the repository.

`--vault-id <label>@prompt`
Labelled vault password

GALAXY

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

GALAXY (cont.)

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

`ansible-galaxy collection list`
Installed collections

`ansible-galaxy collection install <ns.name> -U`
Upgrade a collection

`ansible-galaxy role install <name>`
Install a role

`ansible-galaxy role install -r requirements.yml`
Install roles from a file

`ansible-galaxy role init <name>`
Scaffold a new role

`ansible-galaxy role list`
Installed roles

DOCS & CONFIG

`ansible-doc <module>`
Module documentation

`ansible-doc -s <module>`
Short snippet of a module

`ansible-config dump --only-changed`
Settings differing from defaults

`ansible --version`
Version, config file, paths

RECIPES

Safe rollout

```
ansible-playbook site.yml --syntax-check
ansible-playbook site.yml --check --diff -l web
ansible-playbook site.yml -l web
```

Minimal inventory.ini

```
# inventory.ini
[web]
web1.example.com
web2.example.com
[web:vars]
ansible_user=deploy
```

Minimal playbook

```
# site.yml
- hosts: web
  become: true
  tasks:
    - name: nginx installed
      ansible.builtin.package:
        name: nginx
        state: present
```

Encrypt a vars file

```
ansible-vault encrypt group_vars/all/vault.yml
ansible-playbook site.yml --ask-vault-pass
```