Ansible Cheatsheet

By Dejan Panovski • Updated on Jun 6, 2026 • Download PDF

Quick reference for Ansible commands, inventories, playbooks, modules, and vault

Ansible is an agentless automation tool that configures servers over SSH using YAML playbooks. This cheatsheet covers the commands, inventory syntax, ad hoc usage, common modules, and vault options you reach for most often.

Install & Verify

Set up Ansible on the control node. See the full Ansible install guide .

`sudo apt install ansible`	Install on Ubuntu, Debian, and derivatives
`sudo dnf install ansible`	Install on Fedora, RHEL, and derivatives
`pipx install --include-deps ansible`	Install latest in an isolated environment
`ansible --version`	Show version and config file path
`ansible-config dump`	Show all active configuration settings
`ansible-config dump --only-changed`	Show only non-default settings

Inventory

Define and inspect the hosts Ansible manages.

`ansible-inventory --list`	Show the full inventory as JSON
`ansible-inventory --graph`	Show groups and hosts as a tree
`ansible all --list-hosts`	List every host in the inventory
`ansible web --list-hosts`	List hosts in the web group
`ansible -i inventory.ini all -m ping`	Use a specific inventory file
`ansible -i 'host,' all -m ping`	Use an inline inventory (note the comma)

Inventory File (INI)

Basic structure of a static `inventory.ini` file.

`[web]`	Define a host group
`web1 ansible_host=192.168.1.21`	Host alias with connection address
`[all:vars]`	Variables applied to every host
`ansible_user=ubuntu`	SSH user for the connection
`ansible_python_interpreter=/usr/bin/python3`	Pin the remote Python path
`[prod:children]`	Group made up of other groups

Ad Hoc Commands

Run a single module against hosts without a playbook.

`ansible all -m ping`	Test SSH and Python on every host
`ansible web -a "uptime"`	Run a command (default `command` module)
`ansible web -m shell -a "ps aux | grep nginx"`	Use the shell module for pipes and redirection
`ansible web -b -m apt -a "name=nginx state=present"`	Install a package as root (`-b` = become)
`ansible web -m service -a "name=nginx state=restarted" -b`	Restart a service
`ansible web -m copy -a "src=a.conf dest=/etc/a.conf" -b`	Copy a file to the hosts
`ansible web -m setup`	Gather and print all host facts

Running Playbooks

Apply a playbook with ansible-playbook .

`ansible-playbook site.yml`	Run a playbook
`ansible-playbook --syntax-check site.yml`	Validate YAML and structure only
`ansible-playbook --check --diff site.yml`	Dry run and show would-be changes
`ansible-playbook site.yml --limit web1`	Run against a single host
`ansible-playbook site.yml --tags deploy`	Run only tasks with a tag
`ansible-playbook site.yml --skip-tags slow`	Skip tasks with a tag
`ansible-playbook site.yml --start-at-task "name"`	Begin at a named task
`ansible-playbook site.yml -e "var=value"`	Pass an extra variable
`ansible-playbook site.yml -K`	Prompt for the become (sudo) password

Common Modules

Frequently used built-in modules in tasks.

`ansible.builtin.apt`	Manage packages on Debian-based systems
`ansible.builtin.dnf`	Manage packages on RHEL-based systems
`ansible.builtin.service`	Start, stop, enable, and restart services
`ansible.builtin.copy`	Copy a file to managed hosts
`ansible.builtin.template`	Render a Jinja2 template to a file
`ansible.builtin.file`	Set path state, owner, group, and mode
`ansible.builtin.lineinfile`	Ensure a line is present in a file
`ansible.builtin.user`	Create and manage user accounts
`ansible.builtin.git`	Check out a Git repository
`ansible.builtin.systemd_service`	Manage systemd units directly

Playbook Keywords

Core directives used inside a play or task.

`hosts:`	Target group or host pattern for the play
`become: true`	Run tasks with privilege escalation (sudo)
`vars:`	Define variables for the play
`vars_files:`	Load variables from external files
`tasks:`	List of tasks to run in order
`handlers:`	Tasks triggered by `notify`
`notify:`	Trigger a handler when a task changes
`when:`	Run a task only if a condition is true
`loop:`	Repeat a task over a list
`register:`	Save a task result to a variable

Ansible Vault

Encrypt secrets so they are safe to commit.

`ansible-vault create secrets.yml`	Create a new encrypted file
`ansible-vault edit secrets.yml`	Edit an encrypted file
`ansible-vault view secrets.yml`	View without editing
`ansible-vault encrypt vars.yml`	Encrypt an existing plaintext file
`ansible-vault decrypt vars.yml`	Decrypt a file back to plaintext
`ansible-vault rekey secrets.yml`	Change the vault password
`ansible-playbook site.yml --ask-vault-pass`	Prompt for the vault password at run
`ansible-playbook site.yml --vault-password-file .pass`	Read the vault password from a file

Galaxy & Collections

Install and manage roles and collections.

`ansible-galaxy collection install community.general`	Install a collection
`ansible-galaxy collection list`	List installed collections
`ansible-galaxy role install geerlingguy.nginx`	Install a role from Galaxy
`ansible-galaxy role list`	List installed roles
`ansible-galaxy install -r requirements.yml`	Install from a requirements file
`ansible-galaxy init my_role`	Scaffold a new role directory