Ansible commands to communicate with AWS

---

1. Ansible Ad-Hoc Commands

Command	Use / Task
`ansible all -m ping`	Checks if all hosts are reachable via SSH
`ansible webservers -m ping`	Pings only hosts in the “webservers” group
`ansible all -m command -a "uptime"`	Gets uptime from all hosts
`ansible all -m command -a "df -h"`	Checks disk usage on all hosts
`ansible all -m command -a "free -m"`	Shows memory usage
`ansible all -m shell -a "echo Hello from $(hostname)"`	Runs shell command and prints hostname
`ansible all -m copy -a "src=/etc/hosts dest=/tmp/hosts"`	Copies file from master to all slaves
`ansible all -m file -a "path=/tmp/test_dir state=directory mode=0755"`	Creates a directory on all nodes
`ansible all -m yum -a "name=httpd state=present"`	Installs `httpd` on RedHat/CentOS
`ansible all -m apt -a "name=nginx state=present"`	Installs `nginx` on Debian/Ubuntu
`ansible all -m service -a "name=nginx state=started"`	Starts `nginx` service
`ansible all -m service -a "name=nginx state=stopped"`	Stops `nginx` service

`ansible all -m reboot`	Reboots all machines (be cautious)

🔹 2. Ansible Inventory Commands

Command	Use / Task
`ansible-inventory --list -i hosts.ini`	Displays parsed inventory as JSON
`ansible-inventory --graph -i hosts.ini`	Shows a tree view of inventory

🔹 3. Ansible Playbook Commands

Command	Use / Task
`ansible-playbook -i hosts.ini playbook.yml`	Runs a playbook on specified inventory
`ansible-playbook --syntax-check playbook.yml`	Checks playbook syntax
`ansible-playbook -i hosts.ini playbook.yml --check`	Simulates the playbook (dry-run)
`ansible-playbook -i hosts.ini playbook.yml --start-at-task="Install Packages"`	Starts execution from a specific task

🔹 4. Managing Facts (System Info)

Command	Use / Task
`ansible all -m setup`	Gathers system information (facts) from all hosts
`ansible all -m setup -a 'filter=ansible_os_family'`	Shows only OS family of the hosts

`ansible all -m setup -a 'filter=ansible_eth*'`	Shows Ethernet interface facts

🔹 5. User and Permissions

Command	Use / Task
`ansible all -m user -a "name=deploy state=present"`	Creates user `deploy`
`ansible all -m user -a "name=deploy state=absent"`	Deletes user `deploy`
`ansible all -m file -a "path=/tmp/file.txt state=touch"`	Creates an empty file
`ansible all -m file -a "path=/tmp/file.txt state=absent"`	Deletes a file
`ansible all -m authorized_key -a "user=ubuntu key='{{ lookup('file', '/home/user/.ssh/id_rsa.pub') }}'"`	Adds SSH key to user

🔹 6. Advanced: Tags, Limits, and Forks

Command	Use / Task
`ansible-playbook -i hosts.ini playbook.yml --tags install`	Runs only tasks with `tags: install`
`ansible-playbook -i hosts.ini playbook.yml --limit web1`	Runs playbook only on host `web1`
`ansible-playbook -i hosts.ini playbook.yml --forks 10`	Runs 10 parallel connections

🔹 7. Vault (Encrypting Secrets)

Command	Use / Task

`ansible-vault create secret.yml`	Creates an encrypted file
`ansible-vault edit secret.yml`	Opens encrypted file for editing
`ansible-vault view secret.yml`	Views encrypted file
`ansible-playbook site.yml --ask-vault-pass`	Runs playbook using a vault password
`ansible-vault encrypt vars.yml`	Encrypts an existing file
`ansible-vault decrypt vars.yml`	Decrypts the file

📌 Extra Tips

Use `-b` or `--become` for sudo/root privileges:

```
ansible all -m apt -a "name=nginx state=present" -b
```

-

Set default user in `ansible.cfg`:

```
[defaults]
remote_user = ubuntu
inventory = ./hosts.ini
```

-