Beginner Questions

1. What is Ansible?
- Ansible is an open-source automation tool for configuration management, application deployment, orchestration, and provisioning.

2. What is idempotency in Ansible?
- It means running a playbook multiple times will not cause unwanted changes. Example: `state: present` installs a package only if missing.

3. What are Ansible Playbooks?
- YAML files that define automation tasks.

4. Difference between Ansible Ad-Hoc commands and Playbooks?
- Ad-Hoc: single command for quick tasks.
- Playbook: reusable automation in YAML.

5. Difference between `copy` and `template` modules?
- Copy: static file transfer.
- Template: uses Jinja2 templating with variables.

6. How do you run a playbook with extra variables?
- `ansible-playbook site.yaml -e "var=value"`

7. How to skip fact gathering?
- Use `gather_facts: no` in playbook.

8. How do you create a user with password in Ansible?
```yaml
- user:
name: devops
password: "{{ 'mypassword' | password_hash('sha512') }}"
```

Intermediate Questions

1. Difference between Ansible Roles and Playbooks?
- Playbooks define tasks. Roles organize playbooks into reusable units.

2. What are Handlers?
- Special tasks triggered only when notified (e.g., restart a service after config change).

3. Example of Register and When?
```yaml
- command: which git
register: git_check
ignore_errors: yes

- apt:
name: git
state: present
when: git_check.rc != 0
```

4. How do you manage secrets in Ansible?
- Using Ansible Vault (`ansible-vault create secrets.yml`).

5. How to install Apache on Ubuntu and Nginx on CentOS?

```yaml
- apt:
name: apache2
state: present
when: ansible_os_family == "Debian"

- yum:
name: nginx
state: present
when: ansible_os_family == "RedHat"
```

6. What is Ansible Galaxy?
- Repository for sharing and reusing roles.

7. What is Ansible Tower / AWX?
- Web-based UI and API for managing Ansible automation.

Advanced Questions

1. How do you structure a large Ansible project?

```
project/
■■■ site.yaml
■■■ roles/
■ ■■■ web/
■ ■ ■■■ tasks/main.yml
■ ■ ■■■ templates/
■ ■ ■■■ handlers/main.yml
■ ■ ■■■ vars/main.yml
```

2. Write playbook: Create dir, clone Git repo, restart service if changed.

```yaml
- file:
path: /opt/myapp
state: directory

- git:
repo: https://github.com/example/app.git
dest: /opt/myapp
register: git_status

- service:
name: myapp
state: restarted
when: git_status.changed
```

3. Difference between `with_items` and `loop`?

- `loop` is newer, more flexible. Use `loop`.

4. What happens if a playbook fails on one host?
- Other hosts continue, failed host stops unless `ignore_errors: yes` is used.

5. How do you use Ansible with AWS EC2 (Dynamic Inventory)?
- Use `amazon.aws.aws_ec2` plugin with inventory file. Example:

```yaml
plugin: amazon.aws.aws_ec2
regions:
- us-east-1
filters:
instance-state-name: running
```

6. How do you scale Ansible performance?
- Use forks, delegate_to, async tasks, and dynamic inventory.

7. Explain Ansible Pull vs Push?
- Push: control node pushes tasks to hosts.
- Pull: hosts pull playbooks from a central repo.