## Ansible Interview Questions

1. What is CI/CD?
   * Continuous Integration (CI) helps developers build, test, and integrate code changes frequently.
   * Continuous Delivery (CD) ensures that code can be deployed to production at any time.
2. What is Configuration Management?
   * Configuration Management tracks system updates over time.
   * It allows for easy rollback of changes in case of bugs, minimizing downtime.
3. How does Ansible work?
   * Ansible uses modules, playbooks, and plugins.
   * Modules are small codes executed on host machines, playbooks define workflows, and plugins run on the control machine for logging.
4. What are the features of Ansible?
   * Agentless: No software on managed nodes.
   * Python: Built on easy-to-learn Python.
   * SSH: Uses passwordless network authentication.
   * Push architecture: Pushes small codes to client nodes.
   * Setup: Easy to set up with a low learning curve.
   * Manage Inventory: Stores machine addresses in a simple text format.
5. Explain Infrastructure as Code?
   * Infrastructure as Code (IaC) organizes infrastructure management through code.
   * Changes are made through code repositories, improving speed, consistency, and accountability.
6. What is Ansible Galaxy?
   * Ansible Galaxy is a repository of shareable Ansible roles.
   * Roles can be dropped into playbooks for execution.
   * Also used for distributing packages known as collections.
7. Explain Ansible modules in detail?
   * Ansible modules are like standalone scripts that perform tasks.
   * Core Modules are maintained by the Ansible team.
8. What is a YAML file and how do we use it in Ansible?
   * YAML is a readable text file format used in Ansible playbooks.
   * More readable than JSON or XML.
9. What are Ansible tasks?
   * Ansible tasks are unit actions that automate processes.
   * Tasks can be used to install packages, update software, etc.

## Intermediate Ansible Interview Questions

1. How to setup a jump host to access servers having no direct access?
   * Set a ProxyCommand in the ansible\_ssh\_common\_args inventory variable.
   * yaml

ansible\_ssh\_common\_args: '-o ProxyCommand="ssh -W %h:%p -q user@gateway.example.com"'

1. How to automate the password input in playbook using encrypted files?
   * Use a password file for encrypted files.
   * yaml

ansible\_ssh\_common\_args: '-o ProxyCommand="ssh -W %h:%p -q user@gateway.example.com"' ansible-playbook launch.yml --vault-password-file ~/ .vault\_pass.py

1. What are callback plugins in Ansible?
   * Callback plugins control output during cmd program runs.
   * Examples: log\_plays, mail (for email on playbook failures).
   * Custom callbacks can be added.
2. What is Ansible Inventory and its types?
   * Static inventory: List of managed hosts in a text file.
   * Dynamic inventory: Generated by a script or plugins.
   * Example:

yaml

plugin: aws\_ec2 regions: ap-south-1 filters: tag:tagtype: testing

1. What is Ansible Vault?
   * Ansible Vault encrypts sensitive data in playbooks.
   * Encrypt: ansible-vault encrypt foo.yml bar.yml baz.yml
   * Decrypt: ansible-vault decrypt foo.yml bar.yml baz.yml
2. What is the ad-hoc command in Ansible?
   * One-line playbooks for specific tasks.
   * Syntax: ansible [pattern] -m [module] -a "[module options]"
3. Install Nginx using Ansible playbook?
   * Example playbook to install Nginx:
   * yaml

- hosts: stagingwebservers

gather\_facts: False

tasks:

- name: install nginx

apt:

pkg: nginx

state: installed

update\_cache: true

1. What is the difference between Ansible and Puppet?
   * Ansible: Server pushes configuration, no agent polling, easier setup.
   * Puppet: Client pulls config, agent polls, harder setup.
2. What is Ansible Tower and what are its features?
   * Ansible Tower is an enterprise-level solution for managing Ansible.
   * Features: Workflow Editor, Real-Time Analysis, Audit Trail, Execute Commands Remotely.
3. Explain how you will copy files recursively onto a target host?
   * Use the synchronize module.
   * yaml

- name: Copy Files Recursively

hosts: target\_host

tasks:

- name: Synchronize Files

synchronize:

src: /path/to/source/

dest: /path/to/destination/

delegate\_to: localhost

1. What is the best way to make Content Reusable/Redistributable?
   * Use Ansible roles for better organization and reuse.
2. What are handlers?
   * Handlers are special tasks triggered by the "notify" directive.
   * yaml

tasks:

- name: Install Nginx

apt:

pkg: nginx

state: installed

update\_cache: true

notify:

- start nginx

handlers:

- name: start nginx

service:

name: nginx

state: started

1. How does dot notation and array notation of variables are different?
   * Dot notation may have issues with certain characters.
   * Array notation allows dynamic variable composition.

## Advanced Ansible Interview Questions

### 1. How does Ansible synchronize module work?

### Ansible synchronize is like rsync for playbooks.

### Copies files with features like archive, compress, delete.

### Requires rsync on both source and target.

### yaml

### - hosts: host-remote

tasks:

- name: Sync from sync\_folder

synchronize:

src: /var/tmp/sync\_folder

dest: /var/tmp/

### How does the Ansible firewalld module work?

### Manages firewall rules on hosts.

### Uses zones and services.

### yaml

### - name: Permit traffic for https service

ansible.posix.firewalld:

service: https

permanent: yes

state: enabled

- name: Block traffic on port 8081/tcp

ansible.posix.firewalld:

port: 8081/tcp

permanent: yes

state: disabled

### How is the Ansible set\_fact module different from vars, vars\_file, or include\_var?

### set\_fact: Set variables dynamically in tasks.

### Others (vars, vars\_file, include\_var): Set variables before playbook execution.

### When is it unsafe to bulk-set task arguments from a variable?

### Bulk-setting task arguments from a variable can introduce security risks.

### Precedence issues and collision with facts can occur.

### Explain Ansible register.

### register: Stores task execution output.

### Useful for capturing different outputs on each host.

### Variable is valid throughout playbook execution.

### Yaml

- hosts: all

tasks:

- name: Find all txt files in /home

shell: "find /home -name \*.txt"

register: find\_txt\_files

- debug:

var: find\_txt\_files

### How can we delegate tasks in Ansible?

### In Ansible, you can delegate tasks to run on a different machine than the one defined in the play using the delegate\_to keyword. This is useful when certain tasks should be executed on a specific host or a local machine rather than the target hosts.

### Yaml

- hosts: webservers

serial: 5

tasks:

- name: Take machine out of ELB pool

ansible.builtin.command: /usr/bin/take\_out\_of\_pool {{ inventory\_hostname }}

delegate\_to: 127.0.0.1

- name: Actual steps would go here

ansible.builtin.yum:

name: acme-web-stack

state: latest

- name: Add machine back to ELB pool

ansible.builtin.command: /usr/bin/add\_back\_to\_pool {{ inventory\_hostname }}

delegate\_to: 127.0.0.1