## 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.
   * Extras Modules are community-maintained and might get discontinued.
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.
2. How to automate the password input in playbook using encrypted files?
   * Use a password file for encrypted files.
   * Example:

yamlCopy code

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:

yamlCopy code

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. How can looping be done over a list of hosts in a group, inside of a template?
   * Use Jinja2 templating.
   * Example:

jinjaCopy code

{% for host in groups['db\_servers'] %} {{ host }} {% endfor %}

1. What is the ad-hoc command in Ansible?
   * One-line playbooks for specific tasks.
   * Syntax: ansible [pattern] -m [module] -a "[module options]"
2. Install Nginx using Ansible playbook?
   * Example playbook to install Nginx:

yamlCopy code

- hosts: stagingwebservers gather\_facts: False tasks: - name: install nginx apt: pkg=nginx state=installed update\_cache=true

1. How do I access a variable name programmatically?
   * Build variable names by adding strings together.
   * Example:

jinjaCopy code

{{ hostvars[inventory\_hostname]['ansible\_' + which\_interface]['ipv4']['address'] }}

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.
   * Example:

yamlCopy code

- synchronize: src: /first/absolute/path dest: /second/absolute/path delegate\_to: "{{ inventory\_hostname }}"

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.
   * Example:

yamlCopy code

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 to generate encrypted passwords for a user module?
   * Use Ansible ad-hoc command or Passlib library.
2. 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.

### Example:

### yamlCopy code

### - 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.

### Example:

### yamlCopy code

### - 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.

### set\_fact allows on-the-fly value preparation.

### 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.

### Caution needed to avoid overwriting by malicious values.

### Explain Ansible register.

### register: Stores task execution output.

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

### Variable is valid throughout playbook execution.

### Example:

### yamlCopy code

### - 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?

### Use delegate\_to keyword for task delegation.

### Example:

### yamlCopy code

### - 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

### These simplified explanations should make the concepts clearer. Let me know if you have any more questions or if there's anything else I can help you with!