## Ansible Interview Questions

* + **1: What is Ansible, and what problem does it solve?**Ansible is an open-source configuration management tool that helps automate repetitive tasks. It can be used to manage configurations of various machines, deploy applications, and orchestrate systems. Ansible helps solve the problem of manually managing complex IT infrastructure by automating tasks and configurations.

**2: What is an Ansible Playbook?**An Ansible Playbook is a YAML file that defines a set of instructions for Ansible to execute on managed nodes. Playbooks allow users to automate complex tasks in a declarative manner. They specify the desired state of the system, and Ansible figures out the steps to achieve that state.

**3: How do you define inventory in Ansible?**Inventory is a list of managed nodes or servers that Ansible can manage. This information can be organized into groups for better manageability. Inventory can be defined in a static file (usually inventory.ini) or dynamically generated using scripts or cloud provider APIs.

**4: Explain the concept of idempotence in Ansible.**Idempotence in Ansible ensures that a task produces the same result when run multiple times. Ansible checks the current state of the system against the desired state specified in the playbook and only executes the necessary actions to achieve the desired state. This prevents unintended changes from repeated executions.

**5: What are Ansible Roles, and how do you use them?**Ansible Roles are reusable units that organize playbooks, tasks, variables, and files into a structured directory. Roles help reduce complexity, improve maintainability, and promote reusability of playbooks across different projects.

**6: Explain the difference between Ansible ad-hoc commands and playbooks.**Ansible ad-hoc commands are one-time commands used for executing simple tasks on managed nodes directly from the command line. Playbooks, on the other hand, are YAML files containing a set of tasks that can be run repeatedly in a structured and automated manner. Use ad-hoc commands for one-time tasks and playbooks for repetitive tasks.

**7: What is Ansible Tower's role-based access control (RBAC) feature, and why is it important?**Ansible Tower's RBAC feature allows you to define user roles and permissions. This is important for controlling access to sensitive infrastructure and automation tasks. You can grant users specific permissions to view, edit, or execute playbooks based on their roles. RBAC helps maintain security and prevent unauthorized access to critical configurations.

**8: How do you handle sensitive data like passwords in Ansible?**Ansible Vault is a recommended method for storing sensitive data like passwords. It encrypts the data, and you can access it within playbooks using a password. Alternatively, you can use environment variables or external vaults like HashiCorp Vault to store sensitive data.   
Encrypt: ansible-vault encrypt foo.yml bar.yml baz.yml Decrypt: ansible-vault decrypt foo.yml bar.yml baz.yml

**9: Explain the purpose of Ansible Galaxy.**Ansible Galaxy is a free website and command line tool that allows users to find, download, and share Ansible automation content like playbook, roles and collections.

**10: What is Ansible Tower, and how does it enhance Ansible's capabilities?**Ansible Tower is a web-based graphical interface for Ansible. It provides features like role-based access control (RBAC), job scheduling, centralized logging, and graphical dashboards. Ansible Tower enhances Ansible's capabilities by simplifying automation workflows, improving manageability, and providing a user-friendly interface for teams.

**11: How do you debug Ansible playbooks and tasks?**Ansible provides several debugging techniques. The -vvv flag displays a detailed output about the playbook or task being executed. The --check flag performs a dry run without making any changes on the managed nodes. You can also use the debug module to print variable values during playbook execution.

**12: Explain the concept of Ansible Facts.**Ansible Facts are automatically collected information about managed nodes when they connect to the Ansible machine. This information includes details like operating system, IP address, hardware information, and custom facts defined by users. Ansible Facts can be used within playbooks to make decisions based on the state of the managed nodes.

**13: Explain how you can integrate Ansible with other tools and technologies.**Ansible can be integrated with various tools and technologies to expand its capabilities. Here are some examples:

**Cloud platforms**: Ansible integrates with cloud platforms like AWS, Azure, and GCP to automate provisioning and management of cloud resources.  
**CI/CD pipelines**: Ansible can be integrated with CI/CD pipelines to automate infrastructure and application deployment as part of the development lifecycle.  
**Monitoring tools**: Ansible can be integrated with monitoring tools to trigger automation tasks based on alerts or events.  
**Configuration management databases**: Ansible can interact with configuration management databases like CMDBs to keep track of infrastructure configurations.

**14: What is Ansible Tower's role-based access control (RBAC) feature, and why is it important?**Ansible Tower's RBAC feature allows you to define user roles and permissions. This is important for controlling access to sensitive infrastructure and automation tasks. You can grant users specific permissions to view, edit, or execute playbooks based on their roles. RBAC helps maintain security and prevent unauthorized access to critical configurations.

15. How do you manage the RBAC of users for Ansible Tower?  
**RBAC Management in Ansible Tower:** Ansible Tower allows defining user roles with specific permissions (view, edit, execute) for playbooks and inventories. This ensures secure access control.

16. What is handlers in Ansible and why are they used?  
In Ansible, handlers are a way to trigger service actions (like restarting or stopping services) only when certain tasks notify them. Handlers are defined separately from tasks, and they are only executed if a task triggers a notification. They are useful for:

* **Restarting Services:** Automatically restarting a service if its configuration file changes.
* **Notifying Users:** Sending notifications when tasks fail or complete successfully.

17. I would like to run a specific set of tasks only on windows vms and not Linux vms is it possible?  
- name: Install package for Windows when: ansible\_os\_family == "Windows" # Install package specific to Windows  
- name: Install package for Linux when: ansible\_os\_family == "Linux" # Install package specific to Linux

18. Does Ansible support parallel execution of tasks?  
**Parallel Execution:** Yes, Ansible supports parallel execution of tasks across managed nodes using forks or threads, which can significantly speed up playbook execution.

19. What is the protocol that Ansible use to connect to windows vms ?  
**Protocol for Windows VMs:** Ansible can connect to Windows VMs using WinRM (Windows Remote Management) by default or PowerShell remoting.

20. Can you explain me the variable precedence in Ansible?  
**Extra Vars:** Variables passed on the command line with the -e flag. These have the highest precedence and can be used to override any other variable definition.

* **Inventory Vars:** Variables defined within host/group definitions in the inventory file. These are specific to individual hosts or groups.
* **Group Vars:** Variables defined for specific groups in the inventory file. These apply to all hosts within the group.
* **Playbook Vars:** Variables defined within the playbook itself. These are specific to the playbook and can be overridden by the previous three types.

21. What is CI/CD?  
Continuous Integration (CI) involves integrating changes to software continuously so that it's always ready to be built, tested, and deployed.  
Continuous Deployment automatically deploys integrated changes into the target environment after successful quality checks.

22. What is Configuration Management?  
Configuration Management tracks system updates over time. It allows for easy rollback of changes in case of bugs, minimizing downtime.

23. 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.

24. 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.

25. Explain Infrastructure as Code?  
Infrastructure as Code (IaC) organizes infrastructure management through code.  
Changes are made through code repositories, improving speed, consistency, and accountability.

26. 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.

27. What are Ansible tasks?  
Ansible tasks are unit actions that automate processes.  
Tasks can be used to install packages, update software, etc.

28. 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.

29. Explain Decision making in Ansible?  
In Ansible, decision making is often implemented using conditional statements. The most common way to express conditions is by using the when keyword. The when keyword allows you to execute a task based on whether a condition is true or false.

30. Explain Group Variables in Ansible?  
Group variables in Ansible allow you to define variables that are shared among multiple hosts within a specific group. This can be useful for setting common configurations, defining constants, or specifying group-specific details. Group variables are typically stored in a file named group\_vars/group\_name.yml, where group\_name is the name of the group for which you want to define variables.