Ansible

## **Beginner Level Ansible Interview Questions For Freshers**

1. What is Ansible?

* Ansible is an open-source platform that facilitates configuration management, task automation, or application deployment.

1. List Ansible’s advantages

* - It’s agentless and only requires SSH service running on the target machines
* Python is the only required dependency and, fortunately, most systems come with the language pre-installed
* It requires minimal resources, so there’s low overhead
* It’s easy to learn and understand since Ansible tasks are written in YAML.
* Unlike other tools, most of which are Procedural, ansible is declarative; define the desired state, and Ansible fulfills the requirements needed to achieve it

1. What are CD and CI, and what is Ansible’s relationship with them?

* In CD, developers build software that can be released into production at any given time. CI, on the other hand, consists of each developer uploading regularly scheduled integrations (usually daily), resulting in multiple integrations every day. Ansible is an ideal tool for [CI/CD processes](https://www.simplilearn.com/tutorials/devops-tutorial/continuous-delivery-and-continuous-deployment), providing a stable infrastructure for provisioning the target environment and then deploying the application to it.

1. Describe how Ansible works.

* This is one of the most frequently asked ansible interview questions where the interviewer wants to know whether you actually know the tool in and out or not. You can start this way - ansible is broken down into two types of servers: controlling machines and nodes. Ansible is installed on the controlling computer, and the controlling machines manage the nodes via SSH.
* The controlling machine contains an inventory file that holds the node system’s location. Ansible runs the playbook on the controlling machine to deploy the modules on the node systems. Since Ansible is agentless, there’s no need for a third-party tool to connect the nodes.

1. State the requirements for the Ansible server.

* You need a virtual machine with Linux installed on it, running with Python version 2.6 or higher.

1. Explain what a “playbook” is

* A playbook has a series of YAML-based files that send commands to remote computers via scripts. Developers can configure entire complex environments by passing a script to the required systems rather than using individual commands to configure computers from the command line remotely. Playbooks are one of Ansible’s strongest selling points and often referred to as the tool’s building blocks.

1. How do you set up Ansible?

* You can use either the Python installer or a Linux-based installation process, such as apt or yum.

1. You can use either the Python installer or a Linux-based installation process, such as apt or yum.

* It’s an enterprise-level web-based solution that increases Ansible’s accessibility to other IT teams by including an easy-to-use UI (user interface). Tower’s primary function is to serve as the hub for all of an organization’s automation tasks, allowing users to monitor configurations and conduct rapid deployments.

## **Intermediate Ansible Interview Questions**

1. What is “idempotency”?

* idempotency is an important Ansible feature. It prevents unnecessary changes in the managed hosts. With idempotency, you can execute one or more tasks on a server as many times as you need to, but it won’t change anything that’s already been modified and is working correctly. To put it in basic terms, the only changes added are the ones needed and not already in place.

1. What is Ansible Galaxy?

* This is a tool bundled with Ansible to create a base directory structure. Galaxy is a website that lets users find and share Ansible content. You can use this command to download roles from the website:
* $ ansible-galaxy install username.role\_name

1. How do you use Ansible to create encrypted files?

* To create an encrypted file, use the ‘ansible-vault create’ command.

$ ansible-vault create filename.yaml

You will get a prompt to create a password, and then to type it again for confirmation. You will now have access to a new file, where you can add and edit data.

1. What are “facts” in the context of Ansible?

* Facts are newly discovered and known system variables, found in the playbooks, used mostly for implementing conditionals executions. Additionally, they gather ad-hoc system information.

You can get all the facts by using this command:

$ ansible all- m setup

1. Explain what an ask\_pass module is.

* It’s a playbook control module used to control a password prompt. It’s set to True by default.

1. What’s an ad hoc command?

* Users initiate ad hoc commands to initiate actions on a host without using a playbook. Consider it a one-shot command.

1. Explain the difference between a playbook and a play.

* A play is a set of tasks that run on one or more managed hosts. Plays consist of one or more tasks. A playbook consists of one or more plays.

1. What exactly is a configuration management tool?

* Configuration management tools help keep a system running within the desired parameters. They help reduce deployment time and substantially reduce the effort required to perform repetitive tasks. Popular configuration management tools on the market today include Chef, Puppet, Salt, and of course, Ansible.

Advanced Ansible Interview Questions For Experienced Professionals

1. What are tags?

* When there’s an extensive playbook involved, sometimes it’s more expedient to run just a part of it as opposed to the entire thing. That’s what tags are for.

1. Speaking of tags, how do you filter out tasks?

* You can filter out tasks in one of two ways:
* Use –tags or –skip-tags options on the command line
* If you’re in Ansible configuration settings, use the TAGS\_RUN and TAGS\_SKIP options.

1. What’s a handler?

* In Ansible, a handler is similar to a regular task in a playbook, but it will only run if a task alerts the handler. Handlers are automatically loaded by roles/<role\_name>/handlers/main.yaml. Handlers will run once, after all of the tasks are completed in a particular play.

1. How do you test Ansible projects?

* This is another frequently asked ansible interview question. Try elaborating the answer to this question rather than just answering the testing methods in one word. There are three testing methods available:

Asserts

Asserts duplicates how the test runs in other languages like Python. It verifies that your system has reached the actual intended state, not just as a simulation that you’d find in check mode. Asserts shows that the task did the job it was supposed to do and changed the appropriate resources.

Check Mode

Check mode shows you how everything would run if no simulation was done. Therefore, you can easily see if the project behaves the way you want it to. On the downside, check mode doesn’t run scripts and commands used in roles and playbooks. To get around this, you have to disable check mode for specific tasks by running “check\_mode: no.”

Manual Run

Just run the play and verify that the system is in its desired state. This testing choice is the easiest method, but it carries an increased risk because the results in a test environment may not be the same in a production environment.

1. How do you upgrade Ansible?

* Upgrading Ansible is easy. Just use this command: sudo pip install ansible==<version-number>

1. When do you use {{ }}?

* One of Ansible’s most basic rules is: “Always use {{ }} except when:”

1. Explain how to access shell environment variables.

* You can access the controlling machine’s existing variables by using the “env” lookup plugin. For instance, to access the value of the management machine’s home environment variable, you’d enter:

local\_home:”{{lookup(‘env’,’HOME’)}}”

1. How do you keep data secret in a playbook?

* If you want to keep secret data but still be able to share it publicly, then use Vault in playbooks. But if you’re using –v (verbose) mode and don’t want anyone to see the results, then use:

name: secret task

shell: /usr/bin/do\_something --value={{ secret\_value }}

no\_log: True