

Top Ansible Interview Questions

Ansible Interview Questions - Basic Level

Q1) What is Ansible?

Ans: Ansible is developed in [Python language](#). It is a software tool. It is useful while deploying any application using ssh without any downtime. Using this tool one can manage and configure software applications very easily.

Q2) Ansible Playbooks vs Roles

Ans:

Roles	Playbooks
Roles are reusable subsets of a play.	Playbooks contain Plays.
A set of tasks for accomplishing certain role.	Maps among hosts and roles.
Example: common, webserver.	Example: site.yml, fooservers.yml, webserver.yml.

Q3) What are the advantages of using Ansible?

Ans: The main three advantages of using this tool is,i.e. Ansible

- Agentless
- Very low overhead
- Good performance

Q4) Compare Ansible VS Puppet

Ans:

Ansible	Puppet
Simplest Technology	Complex Technology
Written in YAML language	Written in Ruby language
Automated workflow for Continuous	Visualization and reporting

Delivery	
Agent-less install and deploy	Easy install
No support for Windows	Support for all major OS's
GUI -work under progress	Good GUI
CLI accepts commands in almost any language	Must learn the Puppet DSL

Q5) How Ansible Works?

Ans: There are many similar automation tools available like [Puppet](#), Capistrano, Chef, Salt, Space Walk etc, but Ansible categorize into two types of server: controlling machines and nodes.

The controlling machine, where Ansible is installed and Nodes are managed by this controlling machine over SSH. The location of nodes are specified by controlling machine through its inventory.

The controlling machine (Ansible) deploys modules to nodes using SSH protocol and these modules are stored temporarily on remote nodes and communicate with the Ansible machine through a JSON connection over the standard output.

Ansible is agent-less, that means no need of any agent installation on remote nodes, so it means there are no any background daemons or programs are executing for Ansible, when it's not managing any nodes.

Ansible can handle 100's of nodes from a single system over SSH connection and the entire operation can be handled and executed by one single command 'ansible'. But, in some cases, where you required to execute multiple commands for a deployment, here we can build playbooks.

Playbooks are bunch of commands which can perform multiple tasks and each playbooks are in YAML file format.

Q6) What's the Use of Ansible.

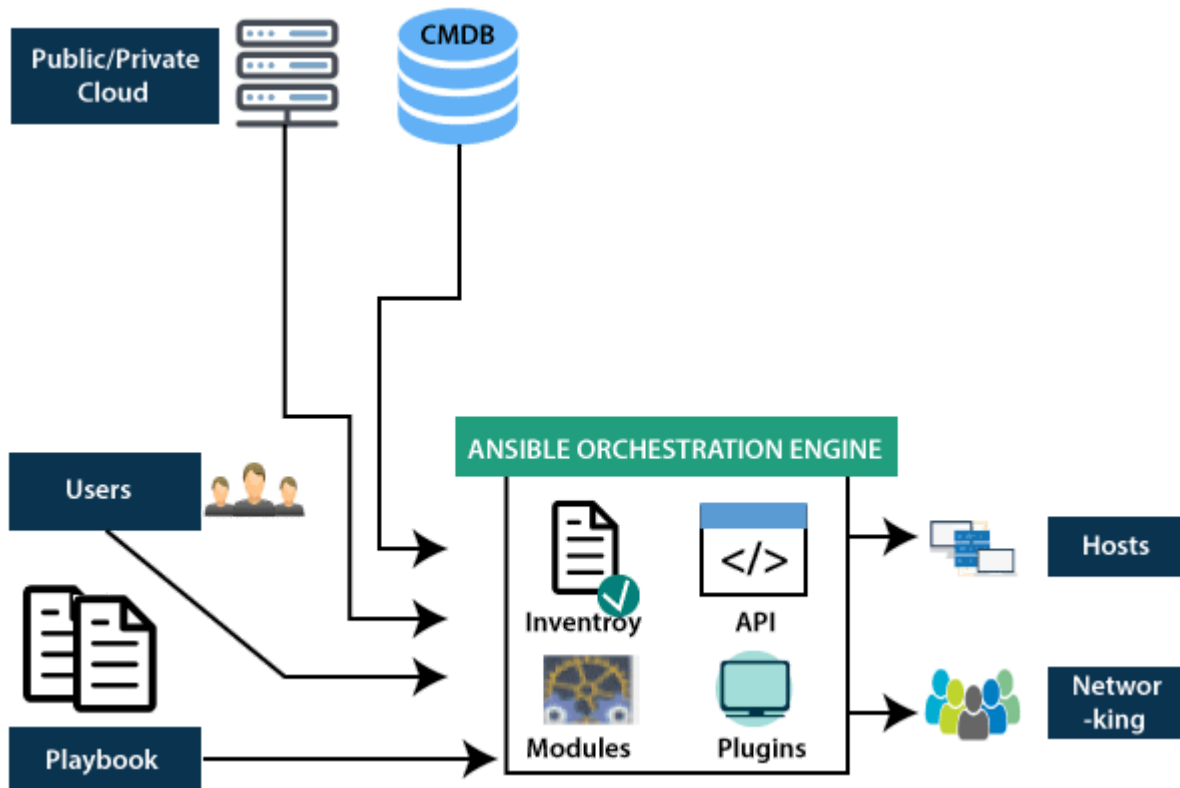
Ans: Ansible can be used in IT Infrastructure to manage and deploy [software applications](#) to remote nodes. For example, let's say you need to deploy a single software or multiple software to 100's of nodes by a single command, here ansible comes into picture, with the help of Ansible you can deploy as many as applications to many nodes with one single command, but you must have a little programming knowledge for understanding the ansible scripts.

We've compiled a series on Ansible, title 'Preparation for the Deployment of your IT Infrastructure with Ansible IT Automation Tool', through parts 1-4 and covers the following topics.

Q7) Explain Ansible architecture?

Ans: Ansible automation engine is the main component of Ansible, that interacts directly with the configuration management database, cloud services and various users who write playbooks to execute it.

The below figure depicts the Ansible architecture:



The following are the components of the Ansible Automation engine:

- **Modules:** Ansible works effectively by connecting nodes and pushing out scripts called "Ansible modules". It helps to manage packages, system resources, files, libraries, etc.
- **Inventories:** These are the lists of nodes or hosts containing their databases, servers, IP addresses, etc.
- **APIs:** These used for commuting public or private cloud services.
- **Plugins:** Plugins augment Ansible's core functionality. Also offers extensions and options for the core features of Ansible - transforming data, connecting to inventory, logging output, and more.
- **Playbooks:** Describes the tasks that need to be executed. They are simple code files written in YAML format and can be used to declare configurations, automating tasks, etc.
- **Hosts:** Hosts are node systems which are automated by Ansible on any machines like Linux, RedHat, Windows, etc.
- **Networking:** Ansible can be used to automate multiple networks and services. It uses a secure and simple automation framework for IT operations and development.
- **Cloud:** A system of remote servers that allows you to store, manage, and process data, rather than a local server.
- **CMDB:** It is a type of repository which acts as a data warehouse for the IT installations.

Q8) What is CI/CD? And how Ansible is related to it?

Ans:

- CI/CD is one of the best software development practices to implement and develop code effectively. CI stands for Continuous Integration, and CD stands for continuous delivery.
- Continuous Integration is a collection of practices that drive developers to implement and check-in code to version control repositories.
- Continuous delivery picks up where continuous Integration ends. This process builds software in such a way that software will be released into production at any given time.
- Ansible is an excellent tool for CI/CD processes, which provide a stable infrastructure to a provision target environment and then deploy the application to it.

Q9) Can you create reusable content with Ansible?

Ans: Yes, Ansible has the concept of roles that helps to create reusable content. To create a role, you need to follow Ansible's conventions of structuring directories and naming files.

Q10) Is Ansible a Configuration management tool?

Ans: Configuration management is the practice to handle updates and manage consistency of a product's performance over a particular period of time. Ansible is an open-source IT Configuration Management tool, which automates a wide variety of challenges in complex multi-tier IT application environments.

Q11) What are the differences between the variable name and environment variables?

Ans:

Variable Names	Environment Variables
By adding strings, we can build variable names	By accessing existing variables, we can access environment variables
Supports adding more strings	Advanced playbooks section sets the environment variables.
Use the IPV4 address for variable names.	Use <code>{{ansible_env.SOME_VARIABLE}}</code> for remote environment variables

Q12) How to create an empty file with Ansible?

Ans: To create an empty file, Ansible uses a file module. For this, we need to set up two parameters.

- **Path** - This place represents the location where the file gets created, either the relative or an absolute path. Also, the name of the file includes here.
- **State** - For creating a new file, this parameter should be set to touch.

Q13) How will you set the environment variable or any path for a task or entire playbook?

Ans: To set the environment variables, we use the environment keyword. We'll use it at the task or other levels in the play:

```
1 environment:  
2   PATH: "{{ ansible_env.PATH }}:/thingy/bin"  
3   SOME: value
```

Q14) How would you describe yourself in terms of what you do and how you'd like to be remembered?

Ans: Obviously I'd like to be remembered as a master of prose who forever changed the face of literature as we know it, but I'm going to have to settle for being remembered as a science fiction writer (and, more and more, critic) who wrote the occasional funny line and picked up a few awards.

Q15) Why are you attracted to science and science fiction?

Ans: Early imprinting, maybe, for the science fiction. When I was quite small a family friend let me read his 1950s run of 'Galaxy' magazine. My favourite aunt pressed John Wyndham's 'The Day of the Triffids' on me; a more terrifying great-aunt gave me G.K. Chesterton's fantastic novels; and so on.

The incurable addiction had begun. Meanwhile, science classes just seemed to be the part of school that made most sense, and I fell in love with Pelican pop-maths titles – especially Kasner's and Newman's 'Mathematics and the Imagination' and all those books of Martin Gardner's 'Scientific American' columns.

Q16) Tell us about your software company and what sort of software it produced(s).

Ans: This goes back to the 1980s and the Apricot home computers, the early, pretty and non-PC-compatible ones. My pal Chris Priest and I both used them for word processing, and he persuaded me to put together a disk of utilities to improve the bundled 'SuperWriter' w/p, mostly written in Borland Turbo Pascal 3 and later 4: two-column printing, automated book index preparation, cleaning the crap out of the spellcheck dictionary, patching SuperWriter to produce dates in UK format, and so on.

Then I redid the index software ('AnsibleIndex') in CP/M for the Amstrad PCW and its Locoscript word processors. When the Apricot market collapsed, I wrote an Apricot emulator in assembler so that people could keep using their horrible but familiar old software on a PC. Eventually, in a fit of nostalgia, I collected all my columns for 'Apricot File' and various Amstrad PCW magazines as books unoriginally titled 'The Apricot Files' and 'The Limbo Files'. (That's probably enough self-promotion, but there's lots more at <https://ansible.uk/>.)

Q17) Describe your newsletter Ansible and who it's aimed at.

Ans: It appears monthly and has been called the 'Private Eye' of science fiction, but isn't as cruel and doesn't (I hope) recycle old jokes quite as relentlessly. Though I feel a certain duty to list some bread-and-butter material like conventions, award winners and deaths in the field, 'Ansible' skips the most boring SF news – the long lists of books acquired, books published, book sales figures, major new remainders – in favour of quirkier items and poking fun at SF notables. The most popular departments quote terrible lines from published SF/fantasy and bizarre things said about SF by outsiders ('As Others See Us'). All the back issues of 'Ansible' since it started in 1979 can be read online.

Q18) So how does Ansible work? Please explain in detail?

Ans: Within the market, there are many automation tools like Puppet, Capistrano, Chef, Salt, Space Walk etc.

- When it comes to Ansible, this tool is categorized into two types of servers:
 1. Controlling machines
 2. Nodes
- Ansible is an agentless tool so it doesn't require any mandatory installations on remote nodes. So there is no background programs that are executed while it is managing any nodes.
- Ansible is able to handle a lot of nodes from a single system over SSH connection.
- Playbooks are defined as a bunch of commands where they are capable of performing multiple tasks and they are in YAML file format.

Q19) What does Ansible offer?

Ans:

- Security and Compliance policy integration
- Automated workflow for Continuous Delivery
- Simplified orchestration
- App deployment
- Configuration management
- Streamlined provisioning

Q20) Can we manage Windows Nano Server using Ansible?

Subscribe to our youtube channel to get new updates..!

Ans: No, it is not possible to manage Windows Nano Server using Ansible as it doesn't have the full access to the .Net framework, which is primarily used by internal components and modules.

Q21) Do we have any Web Interface/ Rest API etc for this?

Ans: Yes, Ansible Inc makes a great efficient tool. It is easy to use.

Q22) What is Ansible Tower?

Ans: Ansible is classified as a web-based solution which makes Ansible very easy to use. It is considered to be or acts like a hub for all of your automation tasks. The tower is free for usage till 10 nodes.

Q23) What are the features of Ansible Tower?

Ans:

- Ansible Dashboard
- Real-time job status updates
- Multi-playbook workflows
- Who Ran What Job When
- Scale capacity with tower clusters
- Integrated notifications
- Schedule ansible jobs
- Manage and track inventory
- Remote command execution
- REST API & Tower CLI Tool

Q24) How do change the documentation and submit it?

Ans: Usually, the documentation is kept in the main project folder in the git repository. For complete instructions on this can be available in docs.

Ansible Interview Questions for Experienced

Here are Advanced Ansible Interview Questions for Experienced

Q25) How do you access Shell Environment Variables?

Ans: If you are just looking to access the existing variables then you can use “env” lookup plugin.

For example:

Accessing the value of Home environment variable on management machine:

```
1 | local_home:"{{lookup('env','HOME')}}"
```

Q26) How can you speed up management inside in EC2?

Ans: It is not advised to manage a group of EC2 machines from your laptop.

The best way is to connect to a management node inside Ec2 first and then execute Ansible from there.

Q27) Is it possible to increase the Ansible reboot module to more than 600 seconds?

Ans: Yes, it is possible to increase the Ansible reboot module to specific values using the below syntax:

```
1 - name: Reboot a Linux system
2
3   reboot:
4
5     reboot_timeout: 1000
```

Q28) How can you use docker modules in Ansible?

Ans: Docker modules require docker-py installed on the host running Ansible.

```
1 $ pip install 'docker-py>=1.7.0'
```

The docker_service module also requires docker-compose

```
1 $ pip install 'docker-compose>=1.7.0'
```

Q29) Explain how you will copy files recursively onto a target host?

Ans: The copy file in Ansible has a recursive parameter. If you have to copy file for a large number of files, then the synchronizing module is the best choice for it.

```
1 - synchronize:
2   src: /first/absolute/path
3   dest: /second/absolute/path
4   delegate_to: "{{ inventory_hostname }}"
```

Q30) How can you disable Cowsay?

Ans: If Cowsay is installed then executing your playbooks within Ansible is very smooth.

Even if you think that you want to work in a professional cow free environment, then you will have two options:

1. Uninstall cowsay
2. Setting up value for the environment variable, like below


```
1 | Export ANSIBLE_NOCOWS=1
```

Q31) How can you access a list of Ansible_Variables?

Ans: By default, Ansible gathers facts under machines under management. Further, these facts are accessed in Playbooks and in templates. One of the best ways to view a list of all the facts that are available in a machine, then you need to run the setup module in the ad-hoc way:

```
1 | Ansible- m setup hostname
```

Once this statement is executed, it will print out a dictionary of all the facts that are available for that particular host. This is the best way to access the list of Ansible_variables.

Q32) How can you see all the variables specific to my host?

Ans: To see all the host-specific variables, that include all facts and other resources are:

```
1 | Ansible - m debug- a “var=hostvars[‘hostname’]” localhost
```

Q33) How do you access a variable name programmatically?

By adding strings together, the variables names are built programmatically like below format:

```
1 | {{ hostvars[inventory_hostname]['ansible_' + which_interface]['ipv4']['addre
```

'inventory_hostname' is a variable that represents the present host you are looping over.

Q34) How to configure a jump host for accessing servers that have no direct access?

Ans: We should set a ProxyCommand in the ansible_ssh_common_args inventory variable. For connecting to the relevant host, arguments defined in this variable are added to scp/ssh/sftp command line.

For example,

```
1 | [gatewayed]
2 | foo ansible_host=192.0.2.1
3 | bar ansible_host=192.0.2.2
```

With the following contents, create the group_vars/gatewayed.yml

```
1 | ansible_ssh_common_args: '-o ProxyCommand="ssh -W %h:%p -q user@gateway.exam'
```

Ansible Certification Training!

Explore Curriculum

When connecting to any hosts in the group gatewayed, Ansible will append these arguments to the command line.

Q35) Explain how you can generate encrypted passwords for the user module?

Ans: Ansible ad-hoc command is the easiest option:

```
1 | ansible all -i localhost, -m debug -a "msg={{ 'mypassword' | password_hash('"
```

The mkpasswd utility available on the Linux systems is also a best option:

```
1 | mkpasswd --method=sha-512
```

Q36) Can you keep data secret in the playbook?

Ans: Yes. If any task that you want to keep secret in the playbook when using -v (verbose) mode, the following playbook attribute will be helpful:

```
1 | - name: secret task
2 |   shell: /usr/bin/do_something --value={{ secret_value }}
3 |   no_log: True
```

It hides sensitive information from others and provides the verbose output.

Q37) What is idempotency?

Ans: Idempotence is an essential feature of Ansible, which helps you to execute one or more tasks on a server as many times needed, but without changing the result beyond the initial application.

Q38) Can you create encrypted files with Ansible?

Ans: Yes, using 'ansible-vault create' command, we can create encrypted files

```
1 | $ ansible-vault create filename.yaml
```

Q39) What is the difference between a playbook and a play?

Ans: A playbook is a list of plays. A play is a set of tasks and roles that run on one or more managed hosts. Play includes one or more tasks.

Q40) How will you get access to the ansible host when I delegate a task?

Ans: We can access it through host variables and even works for all the overridden variables like `ansible_port`, `ansible_user`, etc.

```
1 | original_host: "{{ hostvars[inventory_hostname]['ansible_host'] }}"
```

Q41) Explain the Ansible Tag's usage?

Ans: A tag is an attribute that sets the ansible structure(plays, tasks, roles). When there's an extensive playbook needed, it's more useful to run just a part of it as opposed to the entire thing. That's where tags usage is required.

Q42) How can you filter out tasks in tags?

Ans:

- Use `--tags` or `--skip-tags` options on the command line
- Use the `TAGS_RUN` and `TAGS_SKIP` options, If you're in Ansible configuration settings.

Q43) What are handlers?

Ans: In Ansible, handlers are just like normal tasks in a playbook but run when tasks include the `notify` directive and also indicate that it changed something. It runs only once after all the tasks executed in a particular play. It automatically loads through `roles/handlers/main.yaml`.

They are used to trigger the status of a service, such as restarting or stopping a service.

Q44) How will you upgrade Ansible?

Ans: Using the command `"sudo pip install ansible=="`, we can easily upgrade Ansible.

Q45) Ansible vs Chef?

Ans:

Ansible	Chef
---------	------

Ansible is easier to set up and provides faster performance	Compared to Ansible, Chef is not very easy to set up
Ansible uses YAML (Python) for managing configurations	Chef uses DSL (Ruby) for managing configurations
Highly scalable	Highly scalable
Ansible charges annually \$10,000	Chef Automate charges an annual fee of \$13700

Q46) Why don't you ship in X format?

Ans: They are several reasons for not shipping in X format. In general, it caters towards maintainability. Within the market, they are tons of different ways to ship software and it is very tedious to support all of them.

[Check Out Ansible Tutorials](#)

Q47) What is that Ansible can do?

Ans: Ansible can do the following for us:

1. Configuration management
2. Application deployment
3. Task automation
4. IT orchestration

Q48) Please define what is Ansible Galaxy?

Ans: Ansible Galaxy refers to the website Galaxy where the users will be able to share all the roles to a CLI (Command Line interface) where the installation, creation, and managing of roles happen

Q49) Can you explain how to handle various machines requiring different user accounts or ports to log in?

Ans: Just by setting inventories in the inventory file, we can handle various machines requiring different user accounts or ports to log in.

For example, the following hosts have different ports and usernames:

```
1 [webservers]
2 asdf.example.com ansible_port=5000 ansible_user=alice
3 jkl.example.com ansible_port=5001 ansible_user=bob
```

You can specify connection type to be used by:

```
1 [testcluster]
2 localhost ansible_connection=local
3 /path/to/chroot1 ansible_connection=chroot
4 foo.example.com ansible_connection=paramiko
```

File them in a group_vars/ file.

Q50) Do you know what language Ansible is written in?

Ans: Ansible is written in Python and PowerShell

Q51) Please explain what is Red Hat Ansible?

Ans: Ansible and Ansible Tower by Red Hat, both are an end to end complete automation platforms which are capable of providing the following features or functionalities:

1. Provisioning
2. Deploying applications
3. Orchestrating workflows
4. Manage IT systems
5. Configuration of IT systems
6. Networks
7. Applications

All of these activities are dealt by Ansible where it can help the business to solve the real time business problems.

Q52) Is Ansible is an open source tool?

Ans: Yes, Ansible is an open source tool which is a powerful automation software tool that one can use.

Q53) Why you have to learn Ansible?

Ans: Ansible is more a tool for servers but does it have anything for networking. If you closely look into it, there is some support available in the market for networking devices. Using this tool, it will give you an overall view of your environment and also the knowledge how it works when it comes to network automation.

It is one of those tools where it is considered to be good to explore a new tool.

Q54) What are Ansible server requirements?

Ans: You need to have a virtual machine with Linux installed, which has Python 2.6 version or higher.

Q55) How to install Ansible on CentOS?

Ans: Step 1: Update your Control Node

```
1 | yum update
```

Step 2: Install the EPEL Repository

```
1 | yum install epel-release
```

Step 3: Install Ansible

```
1 | yum install Ansible
```

Q56) How can you connect to other devices within Ansible?

Ans: Once, Ansible is installed and the basic setup has been completed, an inventory is created. This would be the base and one can start testing ansible. To connect to a different device then you have to use “Ping module”. This can be used as a simple connection test.

```
1 | Ansible - m ping all
```

Q57) Can you build your own modules with Ansible?

Ans: Yes, we can create or own modules within Ansible.

It is an open source tool which primarily works on Python. If you are good at programming in Python you can start creating your own modules in few hours from scratch and you don't need to have any prior knowledge of the same.

Q58) How can you find information in Ansible?

Ans: After completing the basic setup, one has to make sure to find out the module called “setup” module. Using this setup module, you will be able to find out a lot of information.

Q59) What does Fact mean in Ansible?

Ans: The term “Facts” is commonly used in Ansible environment. They are described in the playbooks areas where it displays known and discovered variables about the system. Facts are used to implement conditionals executions and also used for getting ad-hoc information of the information.

You can see all the facts via:

```
1 | $ ansible all- m setup
```

So if you want to extract only certain part of the information then you can use “setup” module where you will have an option to filter out the output and just get hold of the fact that you are in need of.

Q60) What is ask_pass in ansible?

Ans: The ask_pass is a control in Ansible Playbook.

This controls whether ansible playbook to prompt a password by default. Usually, the default behavior is no:

```
1 | It is always set to ask_pass=True
```

If you are using SSH keys for authentication purposes then you really don't have to change this setting at all.

Q61) Explain What is ask_sudo_pass

Ans: This control is very similar to ask_pass

The ask_sudo_pass controls the Ansible Playbook to prompt a sudo password. Usually, the default behavior is no:

```
1 | ask_sudo_pass= True
```

One has to make sure and change this setting where the sudo passwords are enabled most of the time.

Q62) Explain what is ask_vault_pass?

Ans: Using this control we can determine whether Ansible Playbook should prompt a password for the vault password by default. As usual, the default behavior is no

```
1 | ask_vault_pass= True
```

Q63) Explain Callback_plugin in Ansible?

Ans: Callbacks are explained as a piece of code in ansible environments where get is used call a specific event and permit the notifications.

This is more sort of a developer related feature and allows low-level extensions around ansible so that they can be loaded from different locations without any problem.

Q64) Explain Module utilities in Ansible?

Ans: Ansible provides a wide variety of module utilities which help the developers while developing their own modules. The basic.py is a module which provides the main entry point for accessing the Ansible library and using those as basics one can start off working.

Q65) Where is the unit testing is available in Ansible?

Ans: Unit tests for all the modules are available in `.test/units/modules`. Firstly you have to setup your testing environment

Q66) Explain in detail about ad-hoc command?

Ans: Well, ad-hoc commands is nothing but a command which is used to do something quickly and it is more sort of a one-time use. Unlike, the playbook is used for a repeated actions which is something that is very useful in Ansible environment. But there might be scenarios where we want to use ad-hoc commands which can simply do the required activity and it is a nonrepetitive activity.