ANSIBLE Q&A

**www.zippyops.com**

172-172, 5th floor Old Mahabalipuram Road

(Above Axis Bank-PTC Bus Stop)  
Thuraipakkam  
Chennai 600097

**🖂**[**zippyops@gmail.com**](mailto:zippyops@gmail.com)

**✆+91 7010585768**

**1.What Is Ansible?**

Ansible is an open source automation platform. It can help us with configuration management, application deployment or task automation. It can deploy an application using SSH without any downtime. Ansible is developed or written in [Python](https://www.educba.com/course/web-application-development-pyramid-micro-framework-python/) language.

**2. What Are The Advantages and use Of Ansible?**

Ansible has a huge number of benefits:  
•**No Agent:** Agent is not required for setting up Ansible. If Box can support ssh and it has python, then no issue to set up Ansible.  
•**Idempotent:** Architecture of Ansible is totally structured around the concept of idempotency. The main or core idea is that only those things need to be added which are needed, and those things will be repeatable without side effects.  
•**Declarative not procedural:** a Normal attitude of other configuration tools of following a procedural process, means do this then do that and so on. But Ansible normally writes the description of the state of machine what we want and it takes proper steps toward fulfilling that description.  
•**Very Easy to learn and low overhead.**

**3. How Ansible Works?**

Ansible mainly categorized into two types of server: controlling machine and Nodes.  
Ansible will install on the controlling machine and nodes are managed inside this controlling machine by SSH. Nodes location are specified by controlling machine through its inventory.  
Ansible deploys modules to nodes using SSH protocol, these modules are mainly stored temporarily in remote nodes and communicate with the Ansible machine through a [JSON](https://www.educba.com/course/json-training/) connection over the standard output.  
Agent installation in remote nodes is not required for Ansible as it is agentless. So any background process or demon process not running for Ansible when it’s not managing any nodes.  
Ansible can able to handle more than 100 of nodes in a single system over SSH connection and only one single command ‘ansible’ can be handled entire operation. But some scenarios we can build ‘playbooks’ where we require to execute multiple commands for a deployment.  
Playbooks are actually holding all the sequential command needs to be executed for performing multiple tasks. Playbooks are in YAML file format.

**4. What’s the use of Ansible?**

Ansible mainly used in IT infrastructure to manage or deploy applications to remote nodes. Suppose we want to deploy one application in 100 nodes by just executing one command, then Ansible is actually coming into the picture, but should need some knowledge on Ansible script to understand or execute the same.

**5. When Should I Use {{ }}? Also, How To Interpolate Variables Or Dynamic Variable Names?**

One of the standard rules is ‘always use {{}} except when:’. Conditionals are always run through Jinja2 as to resolve the expression. So when:failed\_when: and changed\_when: are always templated and we should avoid adding {{}}.  
In other cases except when clause we have to use brackets, otherwise distinguish between an undefined variable and String will be very difficult.

**6. How Do I Generate Crypted Passwords For The User Module?**

A normal mkpasswd utility which is available in normal [Linux system](https://www.educba.com/course/linux-system-administration-with-python/) is a great option for generating crypted passwords.  
mkpasswd –method=sha-512  
Is somehow this utility not available in our system, suppose we are using OS X, then also we can easily generate this password by using Python. But for the same, need to install Passlib password hashing library in our system.  
Once it installed, execute command will generate SHA512 password values.  
python -c “from passlib.hash import sha512\_crypt; import getpass; print sha512\_crypt.encrypt(getpass.getpass())”.

**7. What Is The Best Way To Make Content Reusable/ redistributable?**

There have 3 ways to reuse files in playbooks of Ansible. 3 ways include, imports and roles.  
Include and Imports are actually helping to create multiple small files of a large playbook or break up a large playbook of multiple small files. Those small files can be used across multiple parent playbooks or even for multiple times within the same playbook.  
Roles are mainly used to manage multiple tasks in a package together. It can include variables, handlers, or even modules and other plugins. Roles also can be uploaded and shared by Ansible Galaxy.

**8. How Do I Copy Files Recursively Onto A Target Host?**

Ansible Playbook is a really simple configuration management and multi-machine deployment system. If we compare with another, it is very popular and well suited for deploying a complex application.  
And Roles in Ansible are automatically loading certain var\_files, task, and handlers based on a known file structure. Roles mainly grouping all the contents and packaged together. It also allows easy sharing of roles with other users.multi-machine deployment system. If we compare with another, it is very popular and well suited for deploying a complex application.  
And Roles in Ansible are automatically loading certain var\_files, task, and handlers based on a known file structure. Roles mainly grouping all the contents and packaged together. It also allows easy sharing of roles with other users.

|  |  |
| --- | --- |
| ****Ansible**** | ****Puppet**** |
| Excellent Performance, Agent less install and deploy. | Performance is little questionable compare to Ansible and it’s an agent based install. |
| Based on Python language | Based on [Ruby](https://www.educba.com/course/ruby-programming-part-1-ruby-basics/) language |
| CLI accepts commands in almost all language. | All customer must learn the Puppet DSL. |
| As it is very and not tested well, it can’t give a proper and mature solution for every situation always. | It always provides mature solution compare to Ansible. |
| GUI is in work in progress. | Good GUI |
| No support for [Windows](https://www.educba.com/course/windows-server-2012/) | Support for all major OS’s |

**9. Difference between a variable name and Environment Variables?**

|  |  |
| --- | --- |
| **Variable Name** | **Environment Variable** |
| Variable name can be built by adding String. | To access the environment variable need to access existing variable. |
| {{ hostvars[inventory\_hostname][‘ansible\_’ + which\_interface][‘ipv4’][‘address’] }} | # … vars: local\_home: “{{ lookup(‘env’,’HOME’) }}” |
| We can add Strings | If we want to add the variable we need to open advance playbooks section. |
| For variable name, we use IPV4 address. | For Remote environment variables, use {{ ansible\_env.SOME\_VARIABLE }} |

10. **How would you describe yourself in terms of what you do and how you’d like to be remembered?**

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.

11. **So how does Ansible work? Please explain in detail?**

Within the market, they 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.

**12**. **Do we have any Web Interface/ Rest API etc for this?**

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

**13**. **What is Ansible Tower?**

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.

14. **How do change the documentation and submit it?**  
Usually, the documentation is kept in the main project folder in the git repository.  
For complete instructions on this can be available in docs.

15. **How do you access Shell Environment Variables?**  
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:

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

15. **How can you speed up management inside in EC2?**  
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.

16. **How can you disable Cowsay?**  
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

Export ANSIBLE\_NOCOWS=1

**17**. **How can you access a list of Ansible\_Variables?**  
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:

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.

**18.** **How can you see all the inventory variables that are defined in the host?**  
The best way to see all the inventory variables is by executing this command below:

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

**19.** **Why don’t you ship in X format?**  
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.

**20.** **What is that Ansible can do?**  
Ansible can do the following for us:  
1. Configuration management  
2. Application deployment  
3. Task automation  
4. IT orchestration.

**21**. **Please define what is Ansible Galaxy?**  
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.

**22**. **Do you know what language Ansible is written in?**  
Ansible is written in Python and PowerShell.

23. **Please explain what is Red Hat Ansible?**  
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

24. **Is Ansible is an open source tool?**  
Yes, Ansible is an open source tool which is a powerful automation software tool that one can use.

25. **Why you have to learn Ansible?**  
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.

26. **What are Ansible server requirements?**  
If you are a windows user then you need to have a virtual machine in which Linux should be installed.  
It requires Python 2.6 version and higher.

27. **How can you connect to other devices within Ansible?**  
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.

Ansible - m ping all

28. **Can you build your own modules with Ansible?**  
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.

29. **How can you find information in Ansible?**  
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.

30. **What does Fact mean in Ansible?**  
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:

$ 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.

31. **What is ask\_pass in ansible?**  
 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:

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.

32. **Explain What is ask\_sudo\_pass**  
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:

ask\_sudo\_pass= True

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

33. **Explain what is ask\_vault\_pass?**  
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

ask\_vault\_pass= True

34. **Explain Callback\_plugin in Ansible?**  
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.

35. **Explain Module utilities in Ansible?**   
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.

36. **Where is the unit testing is available in Ansible?**  
Unit tests for all the modules are available in .test/units/modules.  
Firstly you have to setup your testing environment

37. **Explain in detail about ad-hoc command?**  
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.

38. **Is there a web interface / REST API / etc?**

Yes, Ansible, Inc makes a great product that makes Ansible even more powerful and easy to use.

39. **How do I submit a change to the documentation?**

Documentation for Ansible is kept in the main project git repository, and complete instructions for contributing can be found in the docs.

40. **How to install Ansible?**

Installation of Ansible Ubuntu 14.04

The best way to get Ansible for Ubuntu is to add the project’s PPA (personal package archive) to your system.

To do this effectively, we need to install the software-properties-common package, which will give us the ability to work with PPAs easily. (This package was called python-software-properties on older versions of Ubuntu.)

sudo apt-get update

sudo apt-get install software-properties-common

Once the package is installed, we can add the Ansible PPA by typing the following command:

sudo apt-add-repository ppa:ansible/ansible

For the PPA addition, press enter.

After that we can refresh our system package, we can see available PPA packages and can install the software.

sudo apt-get install ansible  
sudo apt-get update  
Through Ansible, we have the software required to administer our servers.

**41. What is the best way to make content reusable/redistributable?**

If you have not done so already, read all about “Roles” in the playbooks documentation. This helps you make playbook content self-contained and works well with things like git sub modules for sharing content with others.

If some of these plugin types look strange to you, see the API documentation for more details about ways Ansible can be extended.

42. **How do I see all the inventory vars defined for my host?**

You can see the resulting vars you define in inventory running the following command:

ansible -m debug -a "var=hostvars['hostname']" localhost

43. **What is Ansible Role?**

Ansible can interact with configured clients from the command line with the ansible command, and how you can automate configuration with playbooks run through the ansible-playbook command.

The first step in creating a role is creating its directory structure. To create the base directory structure, we’re going to use a tool bundled with Ansible called ansible-galaxy:

$ ansible-galaxy init azavea.packer

azavea.packer was created successfully

That command will create an azavea.packer directory with the following structure:

├── README.md

├── defaults

│ └── main.yml

├── files

├── handlers

│ └── main.yml

├── meta

│ └── main.yml

├── tasks

│ └── main.yml

├── templates

└── vars

└── main.yml

**44. Difference between Variable name and Environment Variables.**

|  |  |
| --- | --- |
| **Variable Name** | **Environment Variables** |
| Variable Name can be built by adding strings. | To access the environment variable need to access existing variables. |
| {{ hostvars[inventory\_hostname][‘ansible\_’ + which\_interface][‘ipv4’][‘address’] }} | # … vars: local\_home: “{{ lookup(‘env’,’HOME’) }}” |
| We can add strings | if we want to set variables, see the advanced playbooks section. |
| For Variable names we use ipv4 address. | For Remote environment variables, use {{ ansible\_env.SOME\_VARIABLE }} |

45. **What are the Advantages of Ansible?**

* Agentless
* Very low overhead
* Good performance

46.