**Ansible Interview Question**

**What is Ansible?**

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.

**Ansible Playbooks vs Roles ?**

|  |  |
| --- | --- |
| **Roles** | **Playbooks** |
| Roles are reusable subsets of a play. | Playbooks contain Plays. |
| A set of tasks for accomplishing certain role. | Mapps among hosts and roles. |
| Example: common, webservers. | Example: site.yml, fooservers.ym |

**Comapre Ansible VS Puppet ?**

|  |  |
| --- | --- |
| **Ansible** | **Puppet** |
| Simplest Technology | Complex Technology |
| Written in YAML language | Written in Ruby language |
| Automated workflow for Continuous Delivery | Visualization and reporting |
| Agent-less install and deploy | Easy install |
| No support for Windows | Support for all major OS’s |
| Good GUI | GUI - work under progress |
| CLI accepts commands in almost any language | Must learn the Puppet DSL |

**How Ansible Works?**

There are many similar automation tools available like [Puppet](https://mindmajix.com/puppet-training), Capistrano, Chef, Salt, Space Walk etc, but Ansible categorize into two types of server: controlling machines and nodes and not required nay agent.  
Puppet and cheff are pull based(explain), ansible and salt stack are push based.

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.

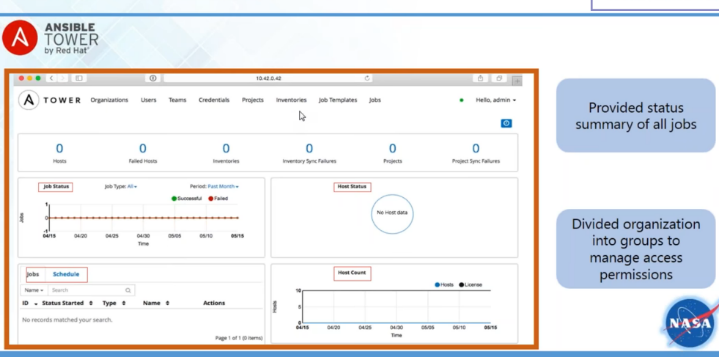
Playbooks are bunch of commands which can perform multiple tasks and each playbooks are in YAML file format.

**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. **What’s the Use of Ansible ?**

Ansible can be used in IT Infrastructure to manage and deploy [software applications](https://mindmajix.com/software-application-development-courses) 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.

This will useful during rollback of version because ansible will take old version version backup before installing new version.

**Do we have any Web Interface/ Rest API etc for this?**  
Yes, Ansible Inc makes a great efficient tool. It is easy to use.

**Ansible Tower ?**

**Ansible tower provide a dashboard. With this we can check all the user,teams, projects, inventories.   
We can also schedule the job this provide every job with unqiue ID helps for tracking purpose.**

**We can also check job status .**

**We can check host count and how many are up and running.**

**How do access Shell Environment Variables?**  
 If you are just looking to access the existing variables then you can use “env” lookup plugin.

example:  
Accessing the value of Home environment variable on management machine:

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

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

Gather\_facts will gather useful variables like files, keys to remote node.

To check the list of variables in machine are :

Ansible -m setup hostname

**How can you see all the inventory variables that are defined in the host?**

**Adhoc command :** ansible -m debug -a "var=hostvars[' inventoru\_name ']" localhost

Task:

-name: list variables

debug: var=hostvars[inventoru\_name] localhost

**What is the code you need to write for accessing a variable name?**

{{ hostvars[inventory\_hostname][‘ansible\_’ which\_interface][‘ipv4’][‘address’] }}

**How to access a variable of the first host in a group?**

{{ hostvars[groups['webservers'][0]]['ansible\_eth0']['ipv4']['address'] }}

**What is that Ansible can do ?**

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

**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

**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.

**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.

**What is ask\_pass in ansible?**

Is a control which tells playbook to prompt password r not , by delault 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.

**Explain What is ask\_sudo\_pass ?**

Similar to ask\_pass but here it control the sudo pass.

Ask\_sudo\_pass= true

**ask\_vault\_pass :** whether Ansible Playbook should prompt a password for the vault password by default

Ask\_vault\_pass= true

**Explain Callback\_plugin in Ansible?**

Which is used to call specific event or permit the notifications.

**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

**How can I set the PATH or any other environment variable for a task or entire playbook?**

Setting environment variables can be done with the environment keyword. It can be used at the task or the play level:

environment:

PATH: "{{ ansible\_env.PATH }}:/thingy/bin"

SOME: value

**How do I access shell environment variables?**

vars:

local\_home: "{{ lookup('env','HOME') }}"

**Ansible supports dot notation and array notation for variables. Which notation should I use?**

If your variable contains dots (.), colons (:), or dashes (-) it is safer to use the array notation for variables.

item[0]['checksum:md5']

item['section']['2.1']

item['region']['Mid-Atlantic']

It is {{ temperature['Celsius']['-3'] }} outside.

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

Great question! Documentation for Ansible is kept in the main project git repository, and complete instructions for contributing can be found in the docs README [viewable on GitHub](https://github.com/ansible/ansible/blob/devel/docs/docsite/README.md).

**How do I keep secret data in my playbook?**

We can achieve this using ansible\_vault

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

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.

**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 }} |

**What is Continuous Delivery?**

Continuous delivery is a practice of delivering the software as soon as it developed. In this method, we need to use versioning control system. The software is constantly updated in live production systems.

**Briefly Explain Pipelining in Ansible.**

It provide single connection instead of multiple connection or stages.

**Why would you want to disable ansible facts.**  
 You can disable facts if it its not being used to save on memory used for storing the variables created during facts/

**How do I handle different machines needing different user accounts or ports to log in with?**

Ansible\_port: port number

Ansible\_user: user name

## Include Import and roles ?

## How do I copy files recursively onto a target host?

**SSH connection in ansible ?**

## Models in ansible ?

## How do I access a variable of the first host in a group?

## **Ansible Tutorial – Modules**

**Hiding data in remote machine :**

Consider here we want to hide file(spoiler.txt) in remote machine.

**PlayBook :**

---

- include\_vars: spoilers.yml

- name: Put the spoiler text in the tmp directory on the remote server.

copy:

content="{{spoiler\_text}}"

dest=/tmp/spoiler\_text.txt

## [Running a Playbook With Vault](https://docs.ansible.com/ansible/2.5/user_guide/playbooks_vault.html#id4)

ansible-playbook site.yml --ask-vault-pass

here we specify password interactively.

We can also specify password using file or script(This will support ansible 1.7 later version).

ansible-playbook site.yml --vault-password-file ~/.vault\_pass.txt

ansible-playbook site.yml --vault-password-file ~/.vault\_pass.py

We can also set password as below in env variable.

ANSIBLE\_VAULT\_PASSWORD\_FILE=~/.vault\_pass.txt

## [Using encrypt\_string](https://docs.ansible.com/ansible/2.5/user_guide/playbooks_vault.html#id6)

ansible-vault encrypt vars/spoilers.yml --vault-password-file ~/.vault\_pass.txt