1. **What is an Ansible?**

Ansible is an open-source software, powerful and agentless automation platform. It is used while deploying an application using ssh with zero downtime. It is also used in Configuration management, Cloud Provisioning, Application Deployment, Intra-Service Orchestration easily. It runs on many Unix-like systems, and can manage configurations of both Unix-like systems as well as Windows systems. And YAML is its declarative script. It is developed in Ruby, Python and PowerShell languages by Michael DeHaan and acquired by Red Hat in 2015.

1. What is configuration management?

Managing configurations of the project like in IT-Infrastructure identifying the configuration, controlling configuration and configuration audit. So this is like a practice of managing and automating all the configurations required for the applications, so that the application can run seamlessly on readily tuned infrastructure. And this reduces the product release time as well.

1. What is an Ansible Task?

Ansible Tasks are small blocks of code in the playbook that can be used to execute any job. For example, if you want to install a package or update a software, you can follow the below code notation, it is a single task to install git software:

Tasks:

- name: Installing Git Application

   yum: git

   state: present

And these tasks are reusable, and named as ansible roles and shared in Ansible Galaxy.

1. What are the advantages of using Ansible?

The main three advantages of Ansible are:

* **Agentless**: Ansible is fast and performs all functions over SSH and doesn't require agent installation. As long as the machine has ssh and python installed.
* **Very Simple**: Has very simple architecture, simple installation and easily manageable.
* **Idempotent**: Architecture of Ansible is structured around the concept of idempotency. Means the things which you do on a daily basis regularly, can be automated.
* **Declarative**: No procedural approach, Ansible is declarative, needs to be defined at a high level in Yaml and Ansible makes the things done.
* **Ansible Modules**: Ansible has so many built in modules, which gives Ansible more power to execute various variety of tasks.
* **Ansible Galaxy**: A website where Ansible users share their customized roles to make it reusable.

1. What is ansible-playbook?

The Ansible Playbooks contains details of remote hosts, user variables, tasks, handlers in it. Playbook may have one or more tasks and these tasks are executed by Ansible. Usually playbooks are .yaml files. Here is a sample playbook:

---

- name: Install Git

   hosts: linux

   gather\_facts: false

   become: true

   vars:

version: 2.19.1

   tasks:

- name: Install yum package

       yum: git

       state: present

1. Ansible Playbooks vs Roles?

|  |  |
| --- | --- |
| Roles | Playbooks |
| Roles are reusable subsets of a Play. | Playbooks may contain one or more Plays. |
| A Role is a set of tasks to be done. | Defines hosts, where roles should be executed. As it knows inventory. |
| Example: general, git. | Example: site.yml, myplay.yml. |

1. What is inventory or host-file in ansible?

Inventories are the host files where information of target servers are written. Inventory also known as host-file. By default this file is located in “/etc/ansible/hosts”. It contains the group of the servers, IP’s, and connection type etc.

1. What is an Ansible Galaxy?

Ansible Galaxy is a GUI service that lets Ansible users share their roles and modules. The Ansible Galaxy command line tool comes with normal Ansible. It is used to install roles from Galaxy or from a SCM system like GIT. Use this command to get roles from the Galaxy:

$ansible-galaxy install username.role\_name

1. Compare Ansible vs Chef vs Puppet?

|  |  |  |
| --- | --- | --- |
| Ansible | Chef | Puppet |
| Ansible Installation is very easy | Time consuming & complicated because of Chef Workstation | Time consuming due to certificate signing between master and agent |
| Simplest Technology | Complex than Ansible | Complex than Ansible |
| Written in YAML Script | Written in Ruby DSL language | Written in Ruby language |
| Agentless setup | Has an Agent | Has an agent |
| Configures any machine which has SSH and Python installed. | Configures only those machines which has Chef Agent installed | Configures only those machines which has Puppet Client installed |

1. What is the Use of Ansible?

From the day one Ansible focused on multi-tier deployments. So it is used to manage and deploy applications to remote nodes. It manages how the entire IT-Infrastructure inter relates. You were using commands or scripts to manage infrastructure or some automation scripts which would take a lot of time and effort. Now Using Ansible there are reusable roles and inventories, you just need to write a YAML script and you are done. By this you can automate repetitive tasks.

1. What is Ansible tower?

Ansible Tower is a commercial product(Ansible with support and extra features) from RedHat . It is used in simplifying the job of ansible automation. Ansible is very easy to use in towers as it acts like a hub for all automation tasks. The Ansible tower is free for usage of upto 10 nodes.

1. What are Ansible vaults?

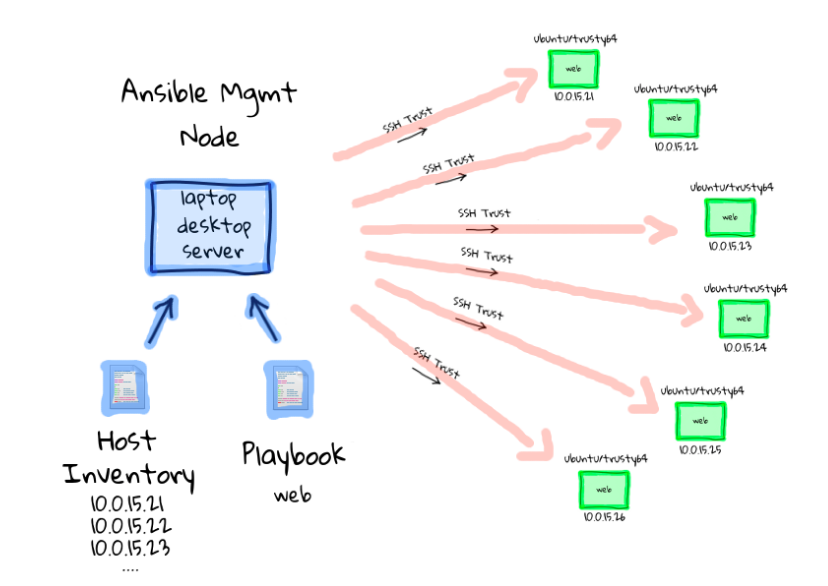
The Ansible vaults are used to keep your sensitive data like passwords or keys in encrypted files, rather than as plaintext in playbooks or roles so that the data can be protected. Not only about protecting data but also makes it access into the playbooks. The files can be encrypted and unencrypted as the Vault is implemented with file-level granularity. These are very user friendly.

1. **What about Ansible architecture?**

Ansible has a very simple architecture. It has control over the configurations of your IT-Infrastructure. Automates the cloud provisioning, configuration management, infrastructure as a code, application deployment, intra-service orchestration, and many other IT needs. It has no agent, just needs ssh and python installed in target servers. The SSH protocol and python interpreter enables Ansible to copy modules and executes them in target servers. At last it removes copied modules from target serves.

The main components of Ansible architecture are:

* Ansible Modules
* Plugins & API
* Inventory/hosts file
* Playbooks



1. How does Ansible work? Please explain in detail?

There are so many configuration management tools like Puppet, Chef, CEFengine, Salt, etc. And the most popular tool is Ansible, in this tool infrastructure is categorized into two type:

1. Ansible Server

2. Target Servers

As Ansible is an agentless tool so it doesn’t require any installations on target servers (remote nodes). So setup and managing nodes is very simple. Ansible can handle huge no.of nodes over SSH connection and entire operations can be executed by one single command “ansible”. Playbooks written in YAML contain one or more play, each play has one or more tasks.

$ansible-playbook playbook\_name.yml

1. Do we have any Web Interface/ Rest API etc for Ansible?

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

1. What is the use of –start-at-task in ansible?

start-at-task option, will start executing the play from the task you specify and subsequent tasks are executed. The prior tasks are skipped.

1. Explain Ansible facts?

Ansible Facts are unchangeable information about the remote hosts. Ansible collects almost all the information about the target hosts as it runs a playbook. The task of collecting this remote system information is called Gathering Facts. To generate facts, ansible runs the setup module. And the command is

$ ansible- m setup hostname

this will print out a dictionary of all the facts available for that host.

You can also see all the facts using the below command

$ ansible all- m setup

1. **What is Role in Ansible?**

Roles are collections of certain tasks variables and handlers. These Roles are shared over Ansible Galaxy to be reusable/redistributable for other Ansible users as well.

1. What are the different components of Ansible?

Ansible consist of the following components:

* Inventories
* Modules
* Variables
* Plugins & APIs
* Hosts
* Playbooks
* Facts
* Roles
* Vault and
* Handlers

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

Let say, suppose these hosts have different ports, username and connection type as one is linux and other is windows machine.

Setting variables in the inventory file is the easiest way for this as shown:

[web\_servers]

abc.example.com   ansible\_port=8000   ansible\_user=user1   ansible\_connection=ssh

xyz.example.com   ansible\_port=8001   ansible\_user=user2   ansible\_connection=winrm

1. **How do you disable Cowsay?**

If Cowsay is installed then executing your playbooks within Ansible is very smooth. But your screen will be mess up with ansible output like this,

   \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

< Ansible output when enabled COWSAY >

  --------------------------------

     \     ^\_\_^

        \    (oo)\\_\_\_\_\_\_

               (\_\_)\                 )\/\

                 ||-----w|

                 ||           ||

To get rid of these message disable cowsay by setting the ANSIBLE\_NOCOWS environment variable to 1

$ export ANSIBLE\_NOCOWS=1

Or you can also disable it from the “ansible.cfg” file.

[defaults]

nocows=1

1. **How to see all the inventory variables that are defined in the host?**

To see all the inventory variables execute the following command:

$ansible -m debug -a var=hostvars[web\_servers] localhost

1. What is that Ansible can do?

Ansible can do the following:

* Configuration management
* Application deployment
* Task automation
* IT orchestration

1. **In which language Ansible is written in?**

Ansible is written in Python, PowerShell and in Ruby by Michael DeHaan.

1. **What is Red Hat Ansible?**

Both Ansible and Ansible Tower are Red Hat products, both are complete automation platforms with the following features:

* Cloud Provisioning
* Deploying applications
* Orchestration
* Manage IT Infrastructure
* Configuration Management
* Networking

But Ansible is an open source and Ansible Tower is a commercial enterprise product that comes with RedHat Support and with some extra features like GUI, etc.

1. Is Ansible an open source tool?

Yes, Ansible is an open source tool which is a powerful automation software tool for configuration management from RedHat.

1. What are Ansible server requirements?

Ansible can configure windows systems, but Windows systems can’t be an ansible server without virtualization. Ansible Server should be a Linux machine with ssh and python 2.6+ version installed.

1. **What is the best way to make content redistributable?**

Create Roles for a set of tasks and share it over Ansible Galaxy, so that it can be reusable/redistributable.

1. **How to generate encrypted passwords for the user module?**

Using Ansible ad-hoc command you can generate encrypted passwords for modules:

ansible all -i localhost -m debug -a msg={{ my\_password | password\_hash(sha512, my\_secretsalt) }}

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

Whatever might be the other device you get the IP of that machine and save it to the inventory file and of course you have a ping module.

$ ansible - m ping hostname

1. **Explain Module in Ansible?**

Ansible has a huge number of modules (called the ‘module library’) as it can be written in any language. Actually Ansible works by connecting target servers and copying and executing a small program in those target servers those small programs are called Ansible modules. Playbook is a high level declarative script, you just have to invoke modules and modules does the entire thing happen.

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

Yes, we can create our own Ansible modules. As it can be written in any language it becomes easy for everyone to write their own custom modules. And these are capable of controlling system resources, like services, directories, files, and handling executing system commands also.

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

Ansible is very simple and powerful, to get all the information of all target server just need to execute the following command:

ansible all -m setup

1. What is ask\_pass in ansible?

This controls whether an ansible playbook should prompt for a password by default or not. Usually, it won’t ask. You can set True or False for it in /etc/ansible/ansible.cfg file. ask\_pass = True/False

1. What is ask\_sudo\_pass?

This controls whether an ansible playbook should prompt for a sudo password by default or not. Usually, it won’t ask. You can set True or False for it in /etc/ansible/ansible.cfg file. ask\_sudo\_pass = True/False

1. What is ask\_vault\_pass?

This controls whether an ansible playbook should prompt for a vault password by default or not. Usually, it won’t ask. You can set True or False for it.

1. **Where is the unit testing available in Ansible?**

Unit tests are available in test/units. To test any file in ansible follow the command:

$ ansible-test units --tox <filename>

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

Instead of writing playbooks you can just execute an ad-hoc command. For example, if we want to copy a file to all hosts in a particular group (stageservers). In simple terms, if you have only one task to do, then why to write playbooks. As in mentiond example. For repeated tasks list of actions can be stored in playbooks and can run whenever needed.

1. **How Can you submit a change to the Documentation in Ansible?**

Ansible documentation is shared over GitHub, there is an option “Edit on GitHub” on the right top corner of docs.ansible.com website. So through GitHub we can submit a change in Ansible Documentation.

1. How do you access Shell Environment Variables?

You can access the existing variables using “env” lookup plugin. For example: Accessing the value of Home environment variable on management machine:

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

1. How can you speed up management inside EC2?

It is not advised to manage a group of EC2 machines from your PC. The best way is to connect to a management node inside Ec2 first and then execute Ansible from there.

1. When to use {{}} ?

A steadfast rule is ‘always use {{ }} except when when:‘. Conditionals are always run through Jinja2 to resolve the expression, so when: failed\_when: and changed\_when: are always templated and you should avoid adding {{}}.

In most other cases you should always use the brackets, even if previously you could use variables without specifying (like with\_ clauses), as this made it hard to distinguish between an undefined variable and a string.

Another rule is ‘moustaches don’t stack’. We often see this:

{{ somevar\_{{other\_var}} }}

The above DOES NOT WORK, if you need to use a dynamic variable use the hostvars or vars dictionary as appropriate:

{{ hostvars[inventory\_hostname]['somevar\_' + other\_var] }}

1. How to generate crypto passwords for the user module?

A normal mkpasswd utility which is available in a normal Linux based system is a great option for generating encrypted 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, we need to install the Passlib password hashing library in our system. Then executing the following command will generate SHA512 password values.

python -c “from passlib.hash import sha512\_crypt; import getpass; print sha512\_crypt.encrypt(getpass.getpass())”.

1. Explain Callback\_plugin in Ansible?

Callback plugins enable adding new behaviors to Ansible when responding to events. Custom Callback plugins can be created by either dropping it into a *callback\_plugins* directory adjacent to your play, inside a role, or by putting it in one of the callback directory sources configured in *ansible.cfg*. Plugins are loaded in alphanumeric order. Examples of predefined plugins are as follows:

‘mail’ callback sends email on playbook failures.

‘log\_plays’ callback records logs of ansible events.

1. What is the way to access shell environment variables in Ansible?

Users need to use the ‘env’ lookup plugin in order to access the existing variables. If you want to access the value of the office/home environment on the machine, you need to write the following code in your playbook.

vars :

“{{ lookup(‘env’,’variable’) }} For local variables

“{{ ansible\_env . SOME\_VARIABLE }}” for remote variables.

1. When should you test playbooks and roles?

In ansible, Tests can be added either in new Playbooks or to existing Playbooks. Therefore, most of the testing jobs offer a clean hosting each time. By using this testing methodology, you need to make very little to no code changes.

1. How to keep secret data in a playbook?

Want to keep secret data in your ansible content and still share it publicly, then you can use Vault in playbooks. If you’re using –v (verbose) mode, and don’t want to show the results, then following command can be used:

  name: secret task

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

  no\_log: True

 This can be used to hide sensitive data from others. The no\_log attribute can also apply to an entire play.

- hosts: all

    no\_log: True

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

This can be done by the below command:

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

In the above command, we’re basically accessing the hostname of the first machine in the webservers group. If you’re using a template to do this, use the Jinja2 ‘#set’ or you can also use set\_fact, like shown below:

- set\_fact: headnode={{ groups[['webservers'][0]] }}

- debug: msg={{ hostvars[headnode].ansible\_eth0.ipv4.address }}

1. What is the method to check the inventory vars defined for the host?

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