**1. What is Ansible?**

It is a software tool that is developed in Python language. It is useful while deploying any application using ssh without any downtime. Using this tool one can manage and configure software applications very easily.

**2. What’s the Use of Ansible?**

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.

**3. What are the major advantages of using Ansible?**

Using Ansible can be beneficial in numerous ways. Then three of its advantages are mentioned below:

o Agentless

o Very low overhead and easy to learn

o Great performance

o Consistent with security

o Reliable

**4. How does Ansible work?**

There are two main categories of server type in Ansible: the nodes and controlling machine. It simply uses the SSH protocol to deploy modules to nodes. These nodes stored in remote nodes interact with Ansible Machine. The Ansible has the capability to manage more than 100 nodes in one single system.

**5. What is continuous delivery in terms of Ansible?**

It is the practice that involves delivering the software as soon as it is developed. For this, there is a need to use the versioning control system. Even in the live production system the software consistently updates.

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

CD stands for continuous delivery, and CI stands for continuous integration; both are software development practices.

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, providing a stable infrastructure for provisioning the target environment and then deploying the application to it.

**7. 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 a Python 2.6 version or higher. you fulfill these requirements and you’re good to go!

Name different modules in Ansible.

In Ansible there are two major types of modules: core modules and extra modules.

Core Modules: These modules are the first preference of the Ansible team. The core modules come with Ansible software.

Extra Modules: The extra modules are reusable but for some reason, they always get a lower rate of response to issues. These are also maintained and managed by the Ansible Community. Although the extra modules are merged with Ansible but one can use it separately in the future.

**8. What is the difference between Ansible and Puppet?**

Ansible: The Ansible has the simplest technology written in the YAML language. It can be quickly installed and deployed because of agentless architecture. The Ansible supports automated workflow for continuous delivery.

Puppet: The puppet has complex technology in comparison to Ansible. This is written in Ruby language. To access this, it is important to learn Puppet DSL.

**9. What are the things Ansible can do?**

With the Ansible these are the following things one can do:

Deployment of application

Configuration management

Task automation

IT orchestration

**10. 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.

**11. What is Ansible Tower?**

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.

**12. Give a brief about Ansible architecture.**

The Ansible is highly based on the agentless architecture. This structure enables you to connect your nodes. The pool of modules can deon any system without any daemons, server or the database. The SSH protocol enables it to execute these modules. It removes them as soon as work is done.

**13. What’s the difference between the environment variable and variable name?**

The variable name can be created by adding stings. On the other hand, for the access of environment variables, there is a need to access the existing variables. The variable name uses ipv4 for the available name. For remote environment variables {{ ansible\_env.SOME\_VARIABLE }} is used.

**14. What language Ansible is written in?**

Ansible is written in PowerShell and Python programming language.

**15. Name the different components of Ansible.**

Ansible automation is consist of the following elements:

İnventories

Modules

APIs

Host

Playbooks

Cloud

Networking

**16. 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

**17. Explain ad-hoc commands.**

The ad-hocs are used to take action on the hosts without writing the playbooks. So, if you have to reboot the hosts in a specific group then there are two ways to do that. You can either create a new playbook or you can simply use the one-off ad-hoc command.

**18. 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.

**19. 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’)}}”

**20. 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

**21. How do you upgrade Ansible?**

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

**22. 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

**23. Does Ansible support AWS?**

There are hundreds of modules present in Ansible that support AWS. It includes:

Auto Scaling groups

CloudFormation

Virtual Private Cloud (VPC)

Security Groups

Relational Database Service (RDS)

CloudTrail

Elastic Cloud Compute (EC2)

**24. What are the variables in Ansible?**

Variables in Ansible are very similar to variables in any programming language. Just like any other variable, an Ansible variable is assigned a value which is used in computing playbooks. You can also use conditions around the variables. Here’s an example:

- hosts: your hosts

vars:

port Tomcat : 8080

Here, we’ve defined a variable called port\_Tomcat and assigned the port number 8080 to it. Such a variable can be used in the Ansible Playbook.

**25. What is an Ansible Task?**

Ansible Tasks allow you to break up bits of configuration policy into smaller files. These are blocks of code that can be used to automate any process. For example, if you wish to install a package or update a software, you can follow the below code snippet:

Install <package\_name>, update <software\_name>

**26. What is Ansible's role and how are they different from the playbook?**

Ansible Roles is basically another level of abstraction used to organize playbooks. They provide a skeleton for an independent and reusable collection of variables, tasks, templates, files, and modules which can be automatically loaded into the playbook. Playbooks are a collection of roles. Every role has specific functionality.

Let’s understand the difference between Ansible roles and playbook with an example.

Suppose you want your playbook to perform 10 different tasks on 5 different systems, would you use a single playbook for this? No, using a single playbook can make it confusing and prone to blunders. Instead, you can create 10 different roles, where each role will perform one task. Then, all you need to do is, mention the name of the role inside the playbook to call them.

**27. 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.

**28. 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.

**29. 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.

**30. 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

**31. 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.

**32. 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.

**33. What is inventory in ansible?**

/etc/ansible/hosts file called as inventory. It contains the group of the server name or IP’s.

**34. What are possible jinja2 templates?**

It is a file that contains all dynamic configurations parameters which will be having .j2 extension.