1. Describe the Ansible register.

Ansible register is used to store the output from task execution in a variable. This is useful when we have different outputs from each remote host. The register value is valid throughout the playbook execution so we can make use of set\_fact to manipulate the data and provide input to other tasks accordingly.

- hosts: all tasks:

name: find all txt files in /home shell: "find /home -name \*.txt" register: find\_txt\_files

debug:

var: find\_txt\_files

In the above example, we are searching for all .txt files in the remote host’s home folder and then capturing it in find\_txt\_files and displaying that variable.

1. In Ansible, how can we delegate tasks?

Task delegation is an important feature of Ansible since there might be use cases where we would want to perform a task on one host with reference to other hosts. We can do this using the delegate\_to keyword.

For example, if we want to manage nodes in a load balancer pool we can do:

- hosts: webservers

serial: 5

tasks:

- name: Take machine out of ELB pool

ansible.builtin.command: /usr/bin/take\_out\_of\_pool {{ inventory\_hostname }}

delegate\_to: 127.0.0.1

- name: Actual steps would go here

ansible.builtin.yum:

name: acme-web-stack

state: latest

- name: Add machine back to ELB pool

ansible.builtin.command: /usr/bin/add\_back\_to\_pool {{ inventory\_hostname }}

delegate\_to: 127.0.0.1

We are also defining serial to control the number of hosts executing at one time. There is another shorthand syntax called local\_action which can be used instead of delegate\_to.

...

tasks:

- name: Take machine out of ELB pool

local\_action: ansible.builtin.command /usr/bin/take\_out\_of\_pool {{ inventory\_hostname }}

...

But there are few exceptions also such as include, add\_host, and debug tasks that cannot be delegated.

1. What is the best way to install Ansible on a CentOS system?

Step 1: Update your Control Node

yum update

Step 2: Install the EPEL Repository

yum install epel-release

Step 3: Install Ansible

yum install Ansible

4. What is Ansible and how does it differ from other Configuration Management software?

Ansible is **an open source IT automation engine that automates provisioning, configuration management, application deployment, orchestration, and many other IT processes**.

**Ansible issues all commands from a central location to perform these tasks.** **No other client software is installed on the node machines**. It uses SSH to connect to the nodes. Ansible only needs to be installed on the control machine (the machine from which you will be running commands) which can even be your laptop.

**It has only master running on the server machine, but no agents running on the client machine**. It uses ssh connection to login to client systems or the nodes you want to configure. Client machine VM requires no special setup, hence it is faster to setup!

1. What are the various parts of ansible? Describe the architecture of Ansible.

The Ansible orchestration engine interacts with a user who is writing the Ansible playbook to execute the Ansible orchestration and interact along with the services of private or public cloud and configuration management database.

Inventory

Inventory is lists of nodes or hosts having their IP addresses, databases, servers, etc. which are need to be managed.

API's

The Ansible API's works as the transport for the public or private cloud services.

Modules

Ansible connected the nodes and spread out the Ansible modules programs. Ansible executes the modules and removed after finished. These modules can reside on any machine; no database or servers are required here. You can work with the chose text editor or a terminal or version control system to keep track of the changes in the content.

Plugins

Plugins is a piece of code that expends the core functionality of Ansible. There are many useful plugins, and you also can write your own.

Playbooks

Playbooks consist of your written code, and they are written in YAML format, which describes the tasks and executes through the Ansible. Also, you can launch the tasks synchronously and asynchronously with playbooks.

Hosts

In the Ansible architecture, hosts are the node systems, which are automated by Ansible, and any machine such as RedHat, Linux, Windows, etc.

Networking

Ansible is used to automate different networks, and it uses the simple, secure, and powerful agentless automation framework for IT operations and development. It uses a type of data model which separated from the Ansible automation engine that spans the different hardware quite easily.

Cloud

A cloud is a network of remote servers on which you can store, manage, and process the data. These servers are hosted on the internet and storing the data remotely rather than the local server. It just launches the resources and instances on the cloud, connect them to the servers, and you have good knowledge of operating your tasks remotely.

CMDB

CMDB is a type of repository which acts as a data warehouse for the IT installations.

1. What are the requirements for Ansible Server?

**System Requirements**

|  | **Required** |
| --- | --- |
| **Subscription** | Valid Red Hat Ansible Automation Platform |
| **OS** | Red Hat Enterprise Linux 7.7 or later 64-bit (x86) or 8.2 or later 64-bit (x86) |
| **Ansible** | version 2.9 required |
| **RAM** | 4Gb minimum |
| **CPUs** | 2 minimum |
| **Disk** | 20Gb dedicated hard disk space  Dependent on size of collections stored |
| **Browser** | A currently supported version of Mozilla FireFox or Google Chrome |
| **Database** | PostgreSQL version 10 |

We cannot install Ansible Tower and a Private Automation Hub instance on the same node. The inventory file can handle the installation of both Ansible Tower and Automation Hub. These instructions are only for the installation of Automation Hub.

1. Describe a handful of Ansible's basic terms and concepts.

Before we get into the important Ansible commands, first, let’s understand its basic terminology.

* **Server**: An entity that provides service for our Ansible
* **Machine**: A physical machine, a VM, or a container
* **Target machine**: An end machine to be configured by Ansible
* **Task**: An action
* **Playbook**: A location where YAML files are written and executed

1. Describe the Infrastructure as Code idea (IaC).

Infrastructure as Code or IaC is a process that DevOps teams should follow to have a more organized way of managing the infra. Instead of some throwaway scripts or manually configuring any cloud component, there should be a code repo where all of these will lie and any change in configuration should be done through it. It is wise to put it under source control also. This improves speed, consistency, and accountability.

1. What do you mean by ad-hoc commands? Give a specific example.

Well, ad-hoc commands are nothing but a command which is used to do something quickly and it is more sort of one-time use. Unlike, the playbook is used for a repeated action which is something that is very useful in the 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.

1. In Ansible, what are the variables?

Ansible uses variables **to manage differences between systems**. With Ansible, you can execute tasks and playbooks on multiple different systems with a single command. To represent the variations among those different systems, you can create variables with standard YAML syntax, including lists and dictionaries.

11. What is the difference between a variable name and a variable that is part of the environment?

Variable Name                                                      Environment Variable

You need to add strings to create variable names.       You need existing variables to access                                                                                                    environment variables.

You can easily create multiple variable names by       To create environment variables, you must

adding strings.                                                              refer to the advanced Ansible playbook.

Use the IPV4 address for variable names.             Use {{ansible\_env.SOME\_VARIABLE}} for remote                                                                     environment variables.

12. What are the Ansible Modules, and what do they do? Describe its various types.

Ansible modules are like functions or standalone scripts which run specific tasks idempotently. The return value of these are JSON string in stdout and input depends on the type of module. These are used by Ansible playbooks.

There are 2 types of modules in Ansible:

* **Core Modules**

The core Ansible team is responsible for maintaining these modules thus these come with Ansible itself. The issues reported are fixed on priority than those in the “extras” repo.

* **Extras Modules**

The Ansible community maintains these modules so, for now, these are being shipped with Ansible but they might get discontinued in the future. These can be used but if there are any feature requests or issues they will be updated on low priority.

Now popular extra modules might enter into the core modules anytime. You may find these separate repos for these modules as ansible-modules-core and ansible-modules-extra respectively.

13. What is an Ansible Task, exactly?

The task is a unit action of Ansible. It helps by breaking a configuration policy into smaller files or blocks of code. These blocks can be used in automating a process. For example, to install a package or update a software

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