
Ansible Assignment 1

1. What is the difference between CI and CD?

Continuous Integration is something that is used for streamlining the development and deployment process. These lead to the more rapid development of cohesive software. Continuous Delivery is on the other hand is a process where your code after being pushed to a remote repository can be taken to production at any time.

2. What is Configuration Management, and how does it work?

It's a practice that we should follow in order to keep track of all updates that are going into the system over a period of time. This also helps in a situation where a major bug has been introduced to the system due to some new changes and we need to fix it with minimum downtime. Instead of fixing the bug, we can roll back the new changes(which caused this bug) as we have been tracking those.

3. What is Ansible, and describe it's working?

Ansible is a combination of multiple pieces working together to become an automation tool. Mainly these are modules, playbooks, and plugins.

- Modules are small codes that will get executed. There are multiple inbuilt modules that serve as a starting point for building tasks.
- Playbooks contain plays which further is a group of tasks. This is the place to define the workflow or the steps needed to complete a process
- Plugins are special kinds of modules that run on the main control machine for logging purposes. There are other types of plugins also.

The playbooks ran via an Ansible automation engine. These playbooks contain modules that are basically actions that run in host machines. The mechanism is followed here is the push mechanism, so ansible pushes small programs to these host machines which are written to be resource models of the desired state of the system.

4. What distinguishes Ansible from other similar tools?

- Ansible calls its configuration files "playbooks".
 - Ansible only uses a master running on the server machine, but no agents running on the client machine. It uses an SSH connection to log in to the client systems or the
-

nodes you want to configure, and the client machine VM has no need for special setup.

- Ansible is faster and easier to set up.
- Ansible uses YAML (Yet Another Markup Language) to manage configurations, a language that's similar to English, and the server pushes the configurations to the individual nodes.

5. What is the purpose of the Ansible Galaxy?

Galaxy is a repository of Ansible roles that can be shared among users and can be directly dropped into playbooks for execution. It is also used for the distribution of packages containing roles, plugins, and modules also known as collection. The `ansible-galaxy-collection` command implements similar to `init`, `build`, `install`, etc like an `ansible-galaxy` command.

6. Can you go over the Ansible modules in detail?

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.

7. What is a YAML file, and how does Ansible use it?

YAML or files are like any formatted text file with few sets of rules just like JSON or XML. Ansible uses this syntax for playbooks as it is more readable than other formats.

8. What are the different types of Ansible tasks?

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>

9. What are the best ways to use YAML files in high-level programming languages like Java, Python, and others?

In java:

```
// We need to declare Topic class with necessary attributes such as name, total_score, user_score, sub_topics
```

```
List<Topic> topics = new ArrayList<Topic>();
```

```
topics.add(new Topic("General String", 4, 3));
```

```
topics.add(new Topic("string1", 5, 3));
```

```
topics.add(new Topic("string 2", 25, 33));
```

```
// We want to save this Topic in a YAML file
```

```
Topic topic = new Topic("new topic", 24, 22, topics);
```

```
// ObjectMapper is instantiated just like before
```

```
ObjectMapper om = new ObjectMapper(new YAMLFactory());
```

```
// We write the `topic` into `topic.yaml`
```

```
om.writeValue(new File("/src/main/resources/topics.yaml"), topic);
```

```
name: "new topic"
```

```
total_score: 24
```

```
user_score: 22
```

```
sub_topics:
```

```
- name: "General string"
```

```
total_score: 4
```

```
user_score: 3
```

```
- name: "string1"
```

```
total_score: 5
```

```
user_score: 3
```

```
- name: "string2"
```

```
total_score: 25
```

```
user_score: 33
```

Similarly, we can read from YAML also:

```
// Loading the YAML file from the /resources folder
ClassLoader classLoader = Thread.currentThread().getContextClassLoader();
File file = new File(classLoader.getResource("topic.yaml").getFile());
// Instantiating a new ObjectMapper as a YAMLFactory
ObjectMapper om = new ObjectMapper(new YAMLFactory());
// Mapping the employee from the YAML file to the Employee class
Topic topic = om.readValue(file, Topic.class);
```

We can use the pyyaml library and read and write easily in YAML format in python.

10. How to set up a jump host to access servers having no direct access?

First, we need to set a ProxyCommand in `ansible_ssh_common_args` inventory variable, since any arguments specified in this variable are added to the `sftp/scp/ssh` command line when connecting to the relevant host(s). For example

```
[gatewayed]
```

```
staging1 ansible_host=10.0.2.1
```

```
staging2 ansible_host=10.0.2.2
```

To create a jump host for these we need to add a command in `ansible_ssh_common_args`

```
ansible_ssh_common_args: '-o ProxyCommand="ssh -W %h:%p -q
user@gateway.example.com"'
```

In this way whenever we will try to connect to any host in the `gatewayed` group ansible will append these arguments to the command line.

11. How can I use encrypted files to automate password entry in a playbook?

To automate password input we can have a password file for all the passwords of encrypted files will be saved and ansible can make a call to fetch those when required.

```
ansible_ssh_common_args: '-o ProxyCommand="ssh -W %h:%p -q
user@gateway.example.com"'
```

This can also be achieved by having a separate script that specifies the passwords. But in this case, we need to print a password to stdout to work without annoying errors.

```
ansible-playbook launch.yml --vault-password-file ~/ .vault_pass.py
```

12. What are Ansible callback plugins?

Callback plugins basically control most of the output we see while running cmd programs. But it can also be used to add additional output. For example `log_plays` callback is used to record playbook events to a log file, and `mail` callback is used to send email on playbook failures. We can also add custom callback plugins by dropping them into a `callback_plugins` directory adjacent to `play`, inside a role, or by putting it in one of the callback directory sources configured in `ansible.cfg`.

13. What is Ansible Inventory and what are the many types of it?

In Ansible, there are two types of inventory files: Static and Dynamic.

- **Static inventory** file is a list of managed hosts declared under a host group using either hostnames or IP addresses in a plain text file. The managed host entries are listed below the group name in each line.
- **Dynamic inventory** is generated by a script written in Python or any other programming language or by using plugins(preferable). In a cloud setup, static inventory file configuration will fail since IP addresses change once a virtual server is stopped and started again.

14. What is an Ansible Vault, exactly?

Ansible vault is used to keep sensitive data such as passwords instead of placing it as plaintext in playbooks or roles. Any structured data file or any single value inside the YAML file can be encrypted by Ansible.

To encrypt a file

```
ansible-vault encrypt sample.yml bar.yml baz.yml
```

To decrypt

```
ansible-vault decrypt sample.yml bar.yml baz.yml
```
