Ansible

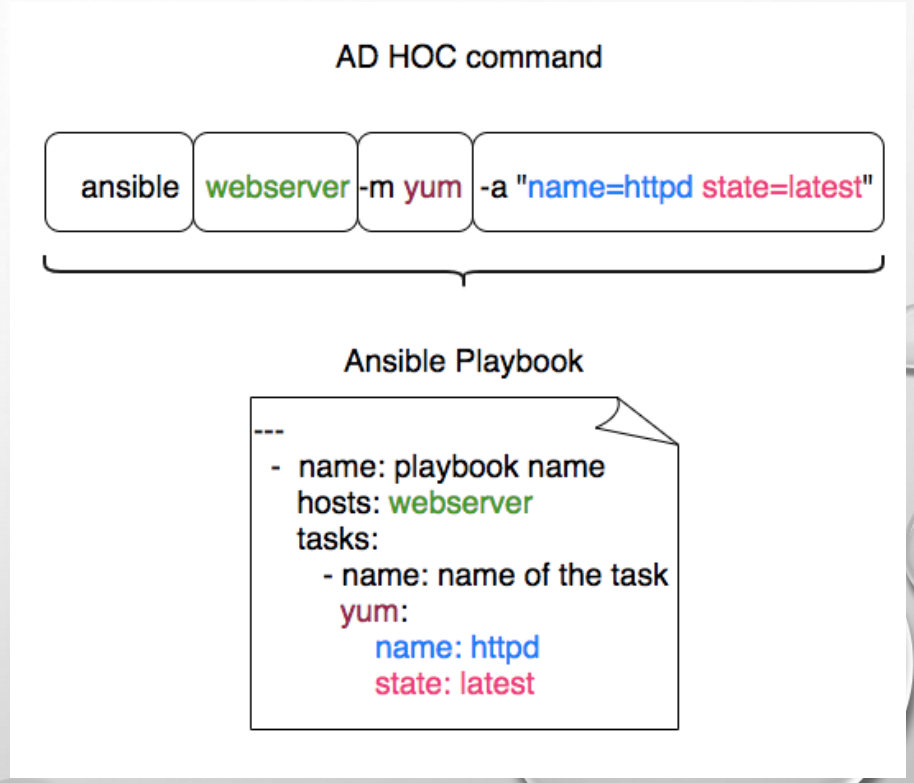
➢ Introduction To YAML: Ansible Automation Methods

We can perform any automation job by pushing the tasks in Ansible by two methods.

1. Command Method (Ansible Ad-hoc commands).
2. File Method (Ansible Playbooks).

Ansible ad-hoc commands are useful for quick tasks or tasks that don't require a lot of complexity or automation.

Playbooks, on the other hand, are best for complex tasks & workflows that require more organization, maintenance, and scaling.

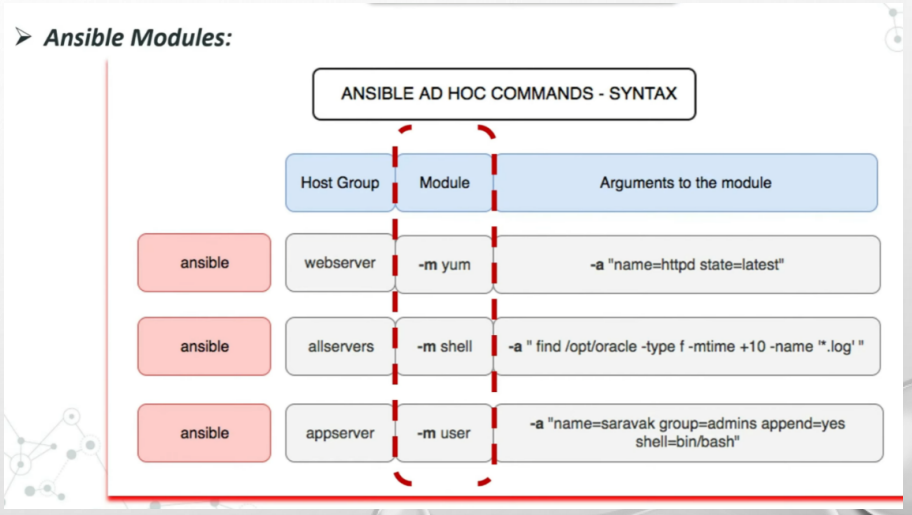


Ansible

➢ Introduction To YAML: Limitations of Ad-hoc Commands

1. You have to type the commands on the terminal of ansible control node for the task every time whenever you want to perform any automation job.
2. For large number of automations jobs/tasks you have to execute separate commands for each task/job.
3. Ad-hoc commands are not suitable for passing conditional statement.
4. Ad-hoc commands aren't suitable for such operations where multiple stdin are required. Like adding multiple users in the system. (Every time separate ad-hoc command needs to be executed for each user account to be created.)
5. Ad-hoc commands aren't suitable for calling the variables as well.

➢ ***Ansible Modules:***

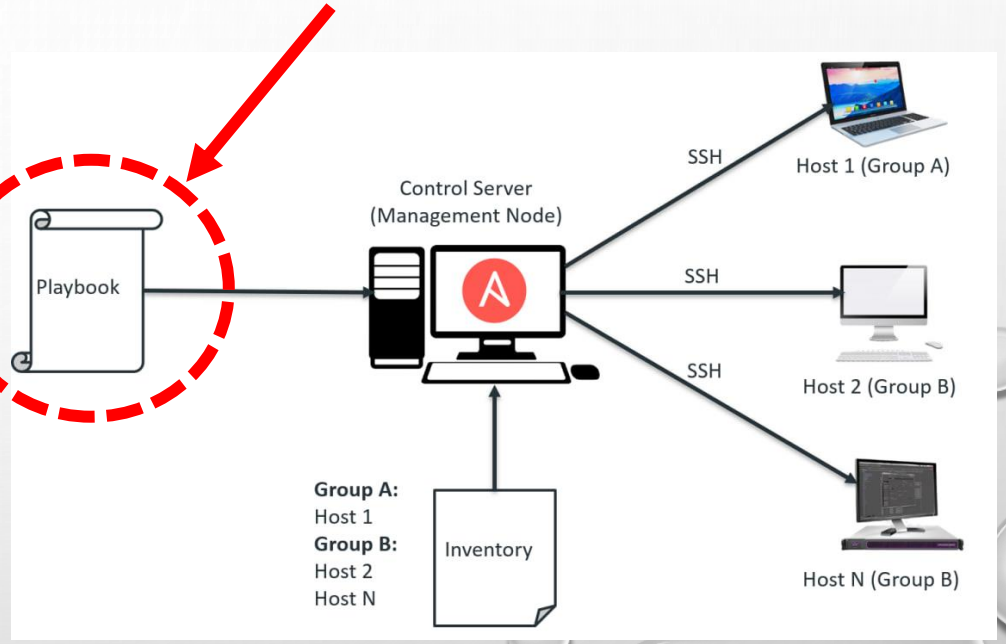


Ansible

➢ Introduction To YAML:

Benefits of Using Ansible Playbooks:

1. You don't need to manually type the commands on the terminal of ansible control node for the each and every task.
2. You can create a single playbook for large number of automations jobs/tasks.
3. You can easily pass conditional statements.
4. Single playbook is enough for multiple tasks which further supports multiple stdin.
5. You can easily use variables in Ansible playbooks.

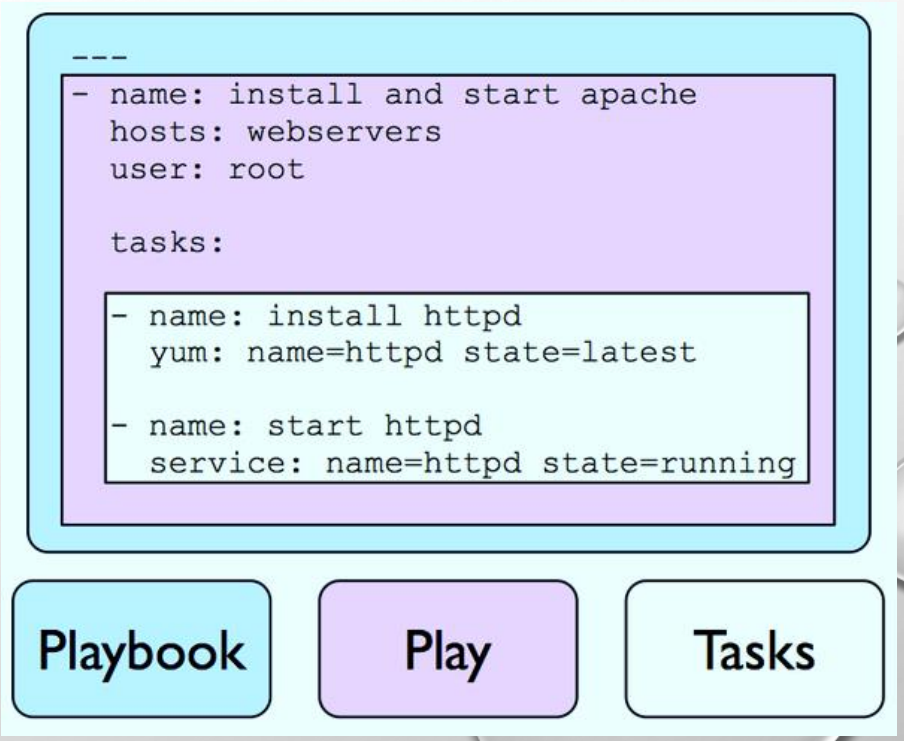


Ansible

➢ Introduction To YAML:

Ansible Playbook:

1. Ansible Playbook is a file which contains Ansible tasks which need to be performed on ansible managed nodes.
2. It is written in YAML. We must know and follow the rules of YAML in order to create any playbook.
3. Ansible Playbook Path: This file can be created any where in the system.
4. Name: Name can be anything but the extension should be YAML or YML only.
5. Ansible playbooks are human readable & are developed in basic text language called YAML.

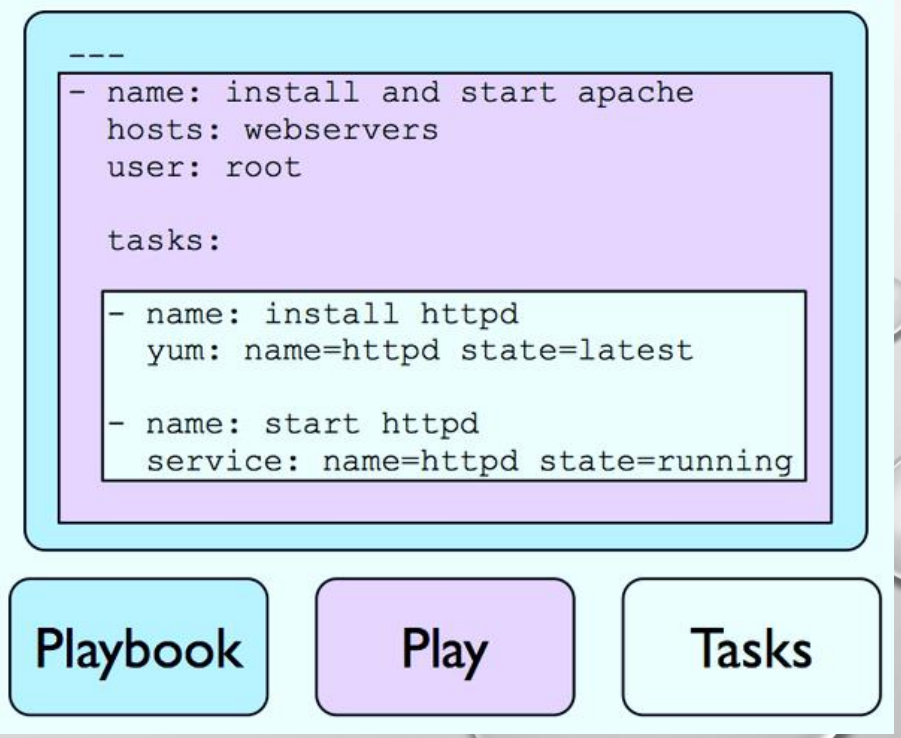


Ansible

➢Introduction To YAML:

Ansible Playbook Writing:

1. Ansible Playbooks are written in YAML which stands for ‘Yet Another Markup Language’.
2. YAML is a human-readable data serialization language that is often used for writing configuration files.
3. Depending on whom you ask, YAML stands for yet another markup language or YAML ain't markup language (recursive acronym), which emphasizes that YAML is for data, not documents.
4. YAML is generally used for orchestration or configuration.
5. YAML is also used in AWS CloudFormation templates and Kubernetes configurations.



Ansible

➢ Introduction To YAML:

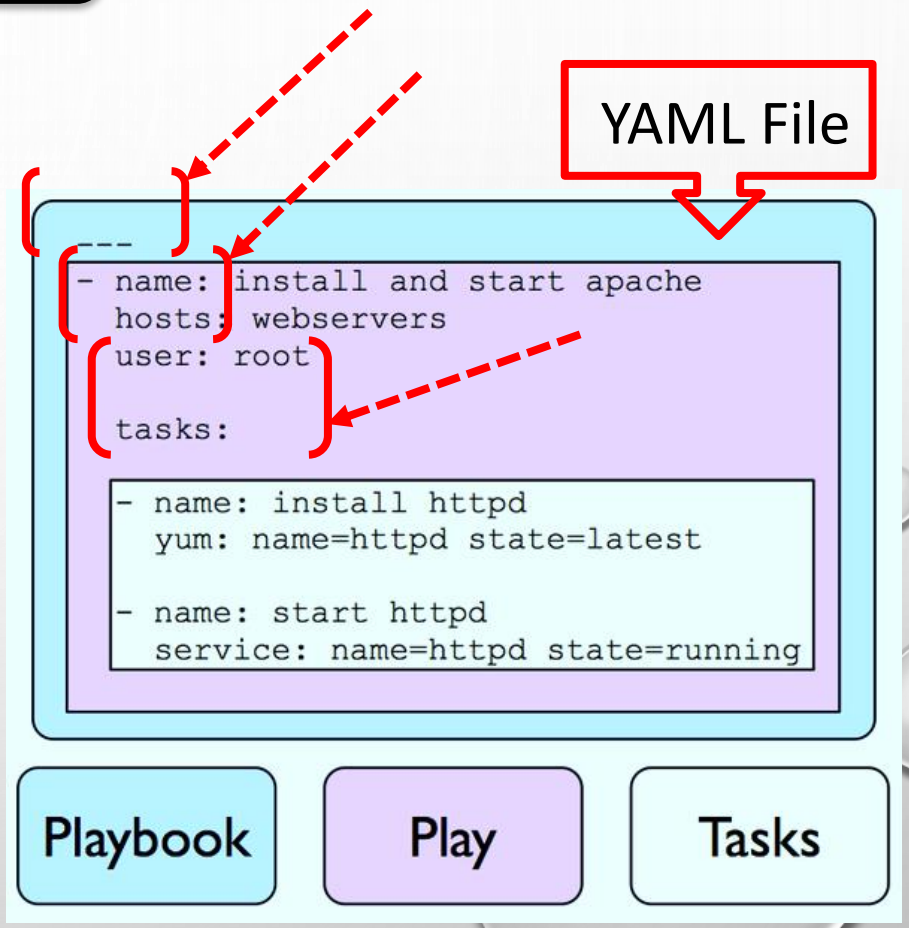
Ansible Playbook Writing:

1. Each YAML file start with three dashes “---” and ends with three dots “…” (which is optional).
2. Every element (member) of the list should be written in a new line with the same indentation starting with hyphen “-” and space.
3. YAML uses simple key: value pair to represent the data separated by space. (kay: value).
4. Data rules for creating YAML files supports:

✓ String Format

✓ Array Format / Listing Format

✓ Mapping Format



Ansible

➢ Introduction To YAML:

Ansible Playbook Writing:

1. String Format:

Single Key value pair is called the string format. (Pair of single key and its single value).

Key: Value

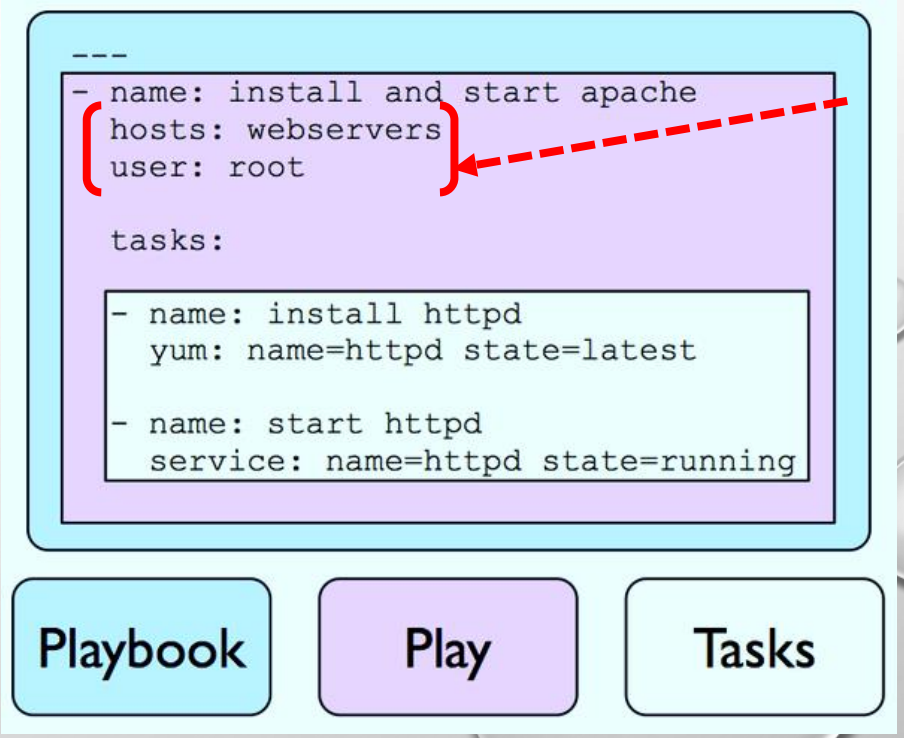
e.g.:

RAM: Ayodhya

hosts: webservers

user: root

package: httpd



Ansible

➢ Introduction To YAML:

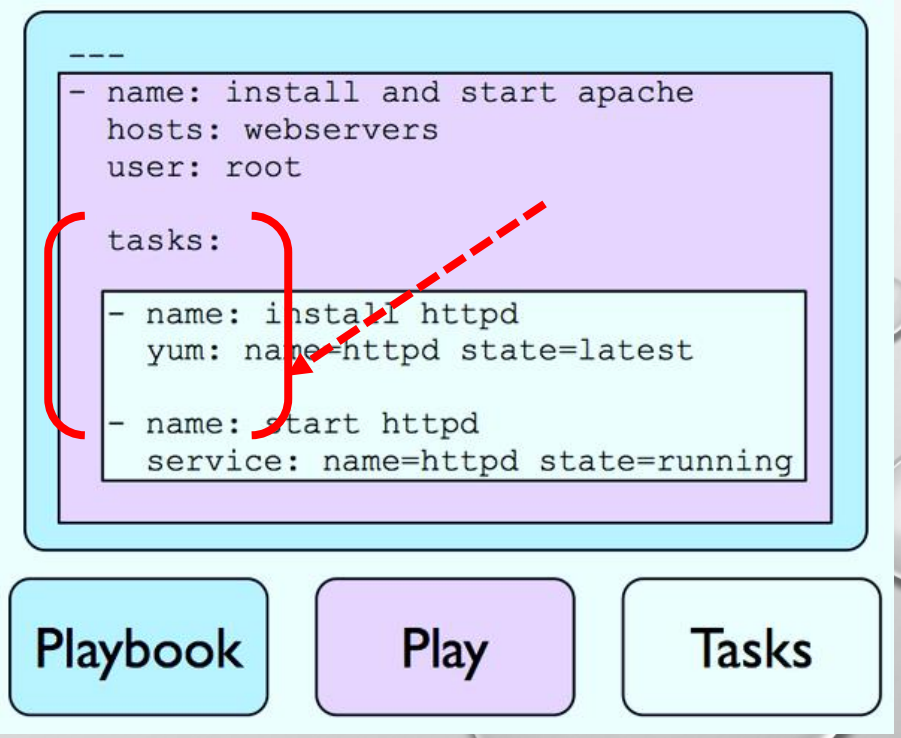
Ansible Playbook Writing:

2. Array/Listing Format:

Key value pair having different values of the same key is called the array/listing format. (Multiple values of any key exists here.)
Key:
- value1
- value2
- value3

e.g.:
package:
- httpd
- vsftpd
- firewalld

Space before - is not mandatory in this format to represent item in the list. Item can start from the same indentation (Margin) as of key.



- **Introduction To YAML:**

Ansible Playbook Writing:

3. Mapping Format:
Key having multiple sub-keys in it as value pairs which further may have different values.

```
Key:
 Key1:
  - value1
  - value2
 Key2:
  - value3
```

e.g.:

```
Shyam
 Phone: 9410219232
 Address: Goa
 Skill:
  - Linux
  - Java
  - Flutter
```

Space (single or many) before - is mandatory in this format to represent item in the list. Item can't start from the same indentation (Margin) as of key. But all list items should have the same indentation.