* Playbooks are Ansible’s configuration, deployment, and orchestration language. They are a completely different way to use Ansible than in ad-hoc task execution mode.
* Playbooks are written in YAML format and have a minimum of syntax, which intentionally tries to not be a programming language or script, but rather a model of a configuration or a process.
* Writing in YAML format allow you to describe your automation jobs in a way that approaches plain English.
* It is easy to learn and easy to understand for new Ansible users, but it is also powerful for expert users.
* **Each playbook is composed of one or more ‘plays’** in a list. **A play maps a group of hosts to some well-defined roles, represented by ansible tasks**. **A task is a call to an Ansible module**.
* **YAML Syntax**
* For Ansible playbook, nearly every YAML file starts with a list. Each item in the list is a

list of **key/value pairs**, commonly called a “hash” or a “dictionary”.

* All YAML files should begin with "---", which indicates the start of a document.

All members of a list are lines beginning at the same indentation level starting with a “-“

(dash) character:

---

2 - hosts: webservers

3 tasks:

4 - name: ensure apache is installed

5 yum: name=httpd state=present

6 - hosts: databases

7 tasks:

8 - name: ensure mysql server is installed

yum: name=mysql-server state=present

* A play in playbook consists of three sections: the hosts section, the variables section, and the tasks section. You can include as many plays as you like in a single playbook.

**The hosts section**

* The hosts section defines hosts on which the play will be run, and how it will be run. In this section
* you can set the SSH username or other SSH-related settings needed to connect to the targeted hosts.

1 - hosts: webservers

2 remote\_user: root

* The hosts that a play will be run on must be set in the value of hosts. This value uses the same
* host-pattern-matching syntax as the one used in Ansible command line:
* • The following patterns target all hosts in inventory:

all

\*

• To address a specific host or set of hosts by name or IP addresses and wildcards:

one.example.com

one.example.com:two.example.com

192.168.1.1

192.168.1.\*

\*.example.com

• The following patterns address one or more groups. Groups separated by a colon indicate an

“OR” configuration (the host may be in either one group or the other):

webservers

webservers:dbservers

• To exclude groups:

webservers:!sydney

all hosts must be in the webservers group but not in the sydney group

• Group intersection:

webservers:&staging

all hosts must be in the webservers group and also in the staging group

• You can also use regular expressions. Start the pattern with a "\_":

*\_*(web|db).\*\.example\.com