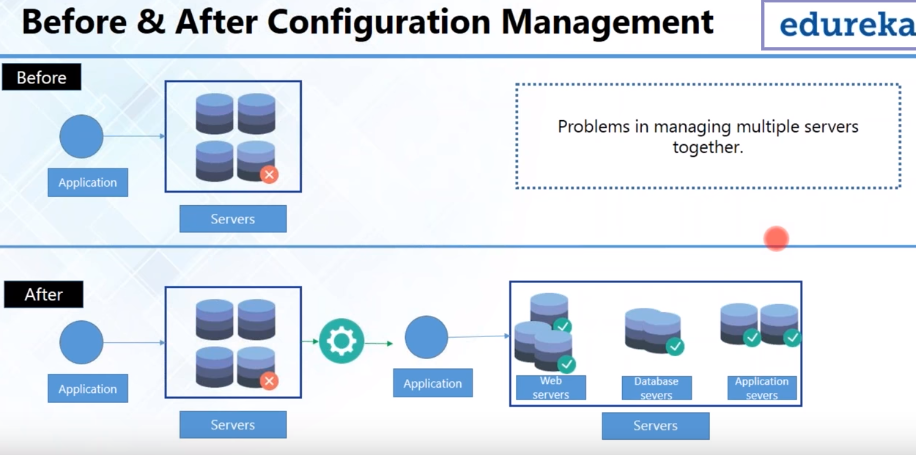
* If there is something going wrong with the system, we can get to know with configuration management and we can fix the issue. We can reconfigure the system with CM
* It documents all the versions as snapshots of system
* It stores the snapshot details in CMDB (configuration management database)

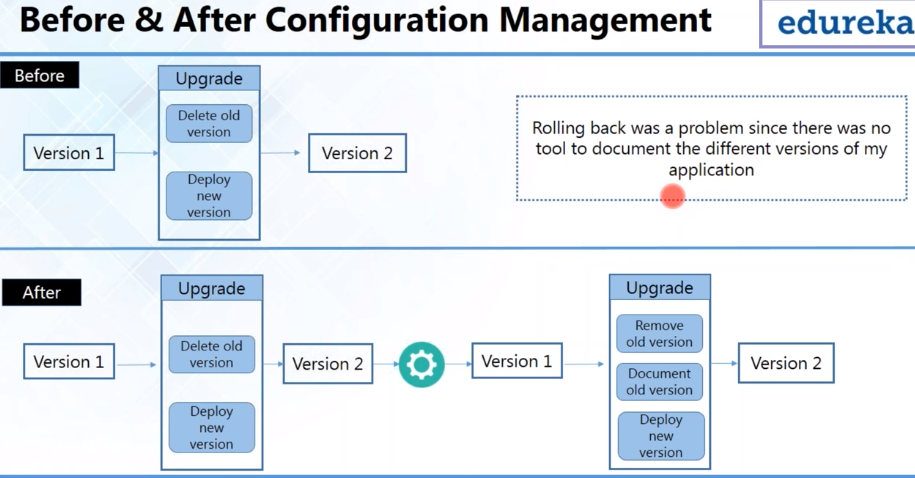
**Why CM:**

* Earlier we were managing servers by hand. We used to login the server, make changes and logoff
* It was difficult to make changes on multiple servers



* But, with the CM, it can be divided based on their tasks as above. This process is known as base lining
* Suppose we want to install lamp stack on servers
* Lamp stack is a software bundle. L for linux, a for apache, m for mysql and p for php
* Configuration for dedicated tasks can be done with scripts. We can use the same scripts for new machines as well
* With CM, it is easy to scale up and scale down the servers
* Earlier configuration was being done by admins by hand and was taking too much time for that

**Rollback:**

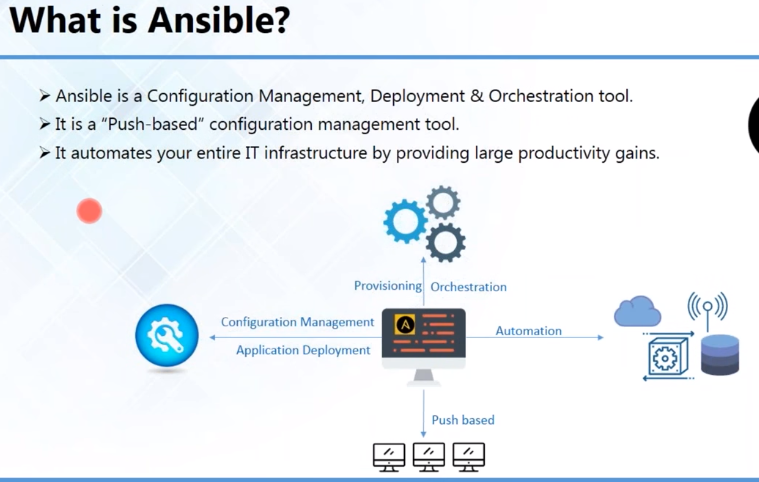


* Earlier, if there was any upgrade on the server, they were used to delete old deployment and place a new one
* And if the new version is not working then it was taking lot time to redeploy old package by restarting the server. We need downtime for that
* With CM, it documents every version of infrastructure. It removes the old configuration, but it will be well documented and then the new version will be deployed
* If new version has problem, we can rollback to old one with zero downtime

**CM tools:**

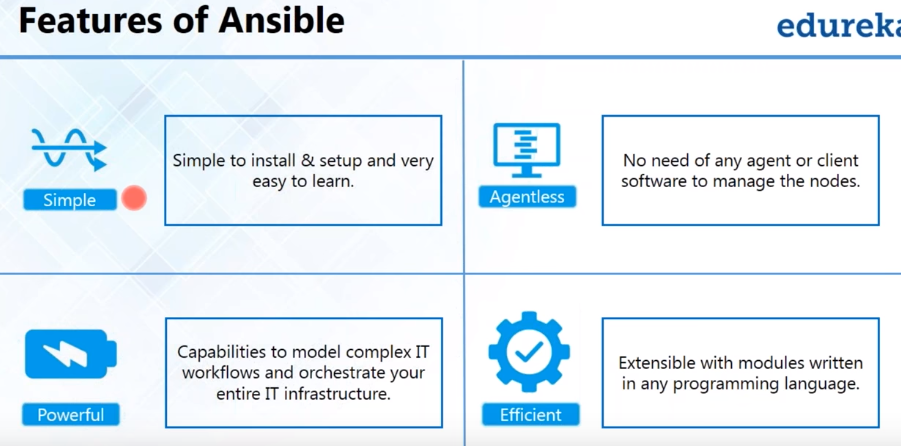
* Ansible, saltstack are the push based
* Puppet and chef are pull based
* There 4 are the most popular

**Ansible:**

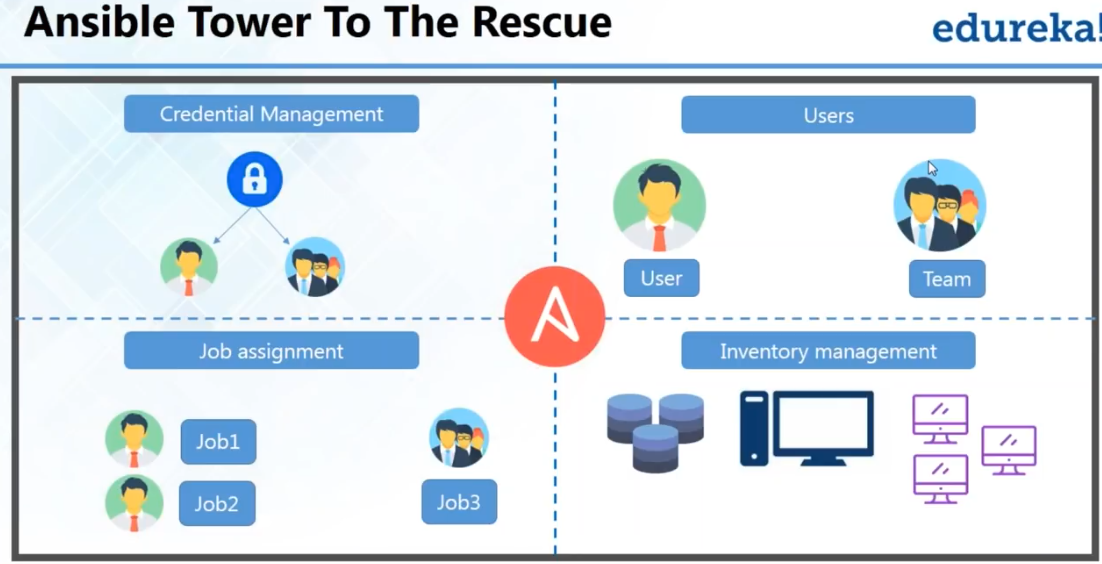


* It can automate network, cloud etc.

**Features of ansible:**

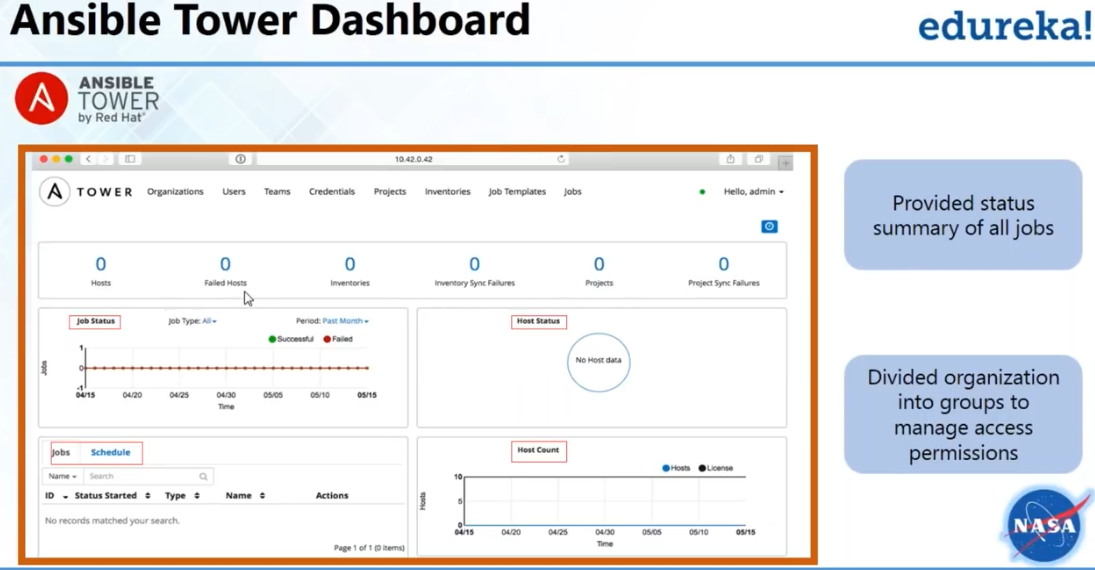


* It is very simple and easy to learn the yaml
* Its agentless, we do need any software to manage nodes, we can just make SSH connection to manage node machines
* Its very powerful, it got with more than 750 inbuilt modules, we can use them any purpose
* All those modules are efficient, which means we can customize them according to the needs



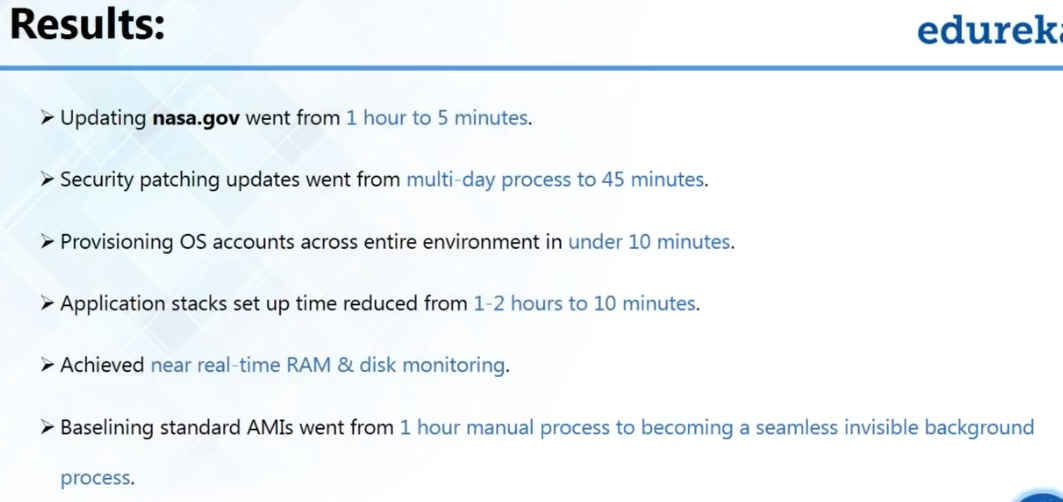
* Ansible tower is most like enterprise level. Web based interface to see the status of all. Easy to use
* We have credential management in this, we can give permissions to a single user or a team
* And, job assignment to a single user or to a team

**Ansible tower dashboard:**

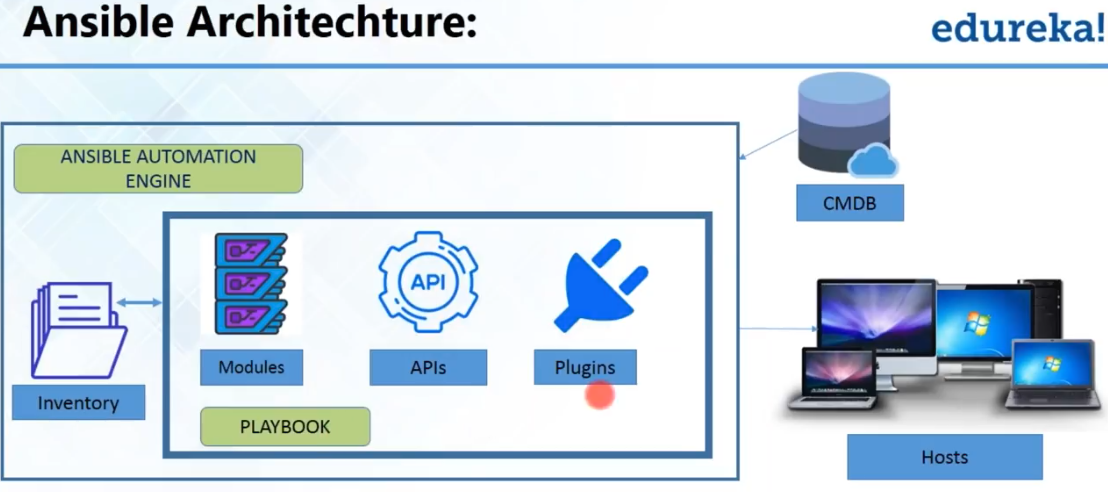


* We can check the job status as well in ansible dashboard

**NASA results:**



**Ansible architecture:**



* CMDB holds the data of IT assets
* There are plugins in ansible like action, connect plugins
* If we execute a yaml script, there must be something needs to be done in the server also, action plugin used to take care of that and connect is used to connect docker etc.
* We can use any kind of connection rather than SSH

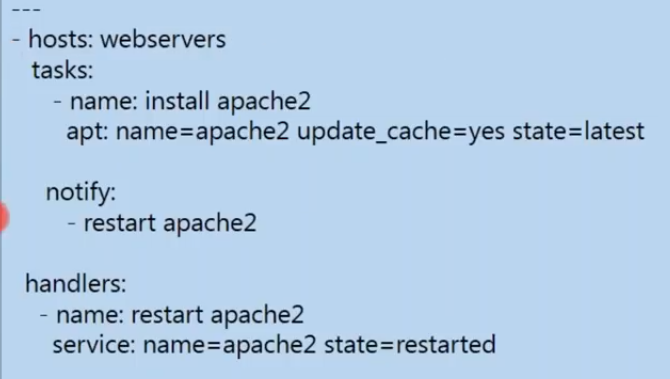
**How to write playbook:**

* Yaml is in reconstructed format, we can use it to reconstruct the date and use it in different environment also



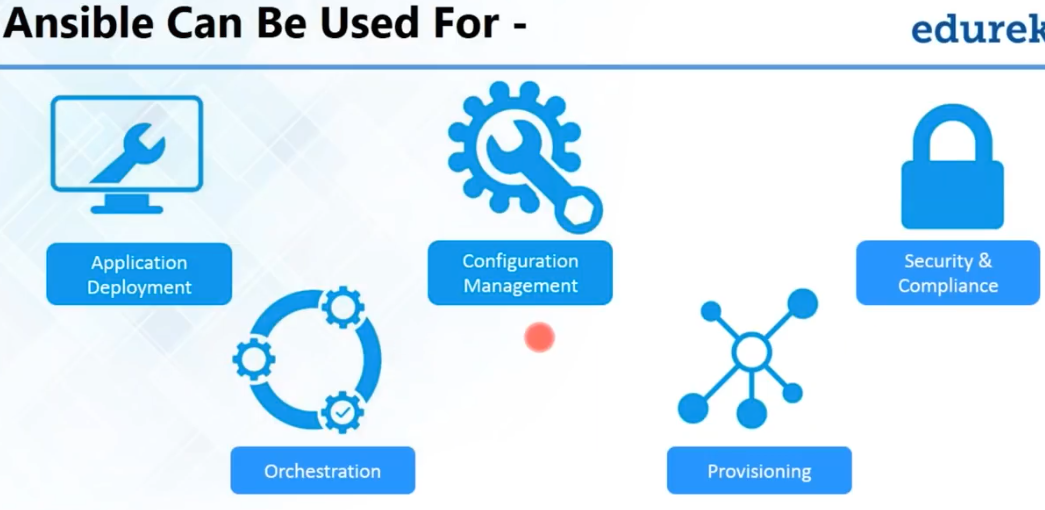
* First, we have to write the list of host machines where we want to run
* Then the variables gathering the facts
* Then the tasks, the tasks get executes in order
* Handlers are also task. To execute this, we need triggers in tasks

**Playbook example:**

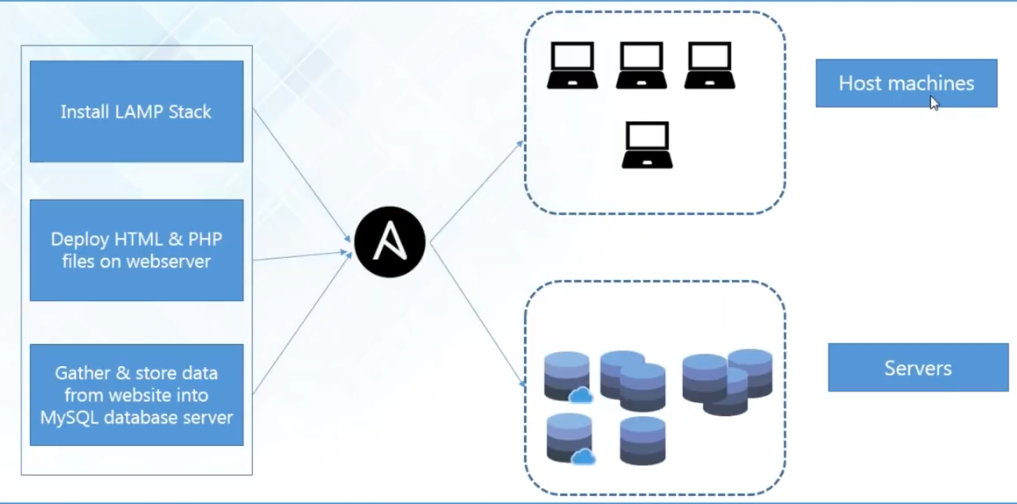


* Notify is the trigger to handler whose job is to restart apache2

**ansible can be used for:**

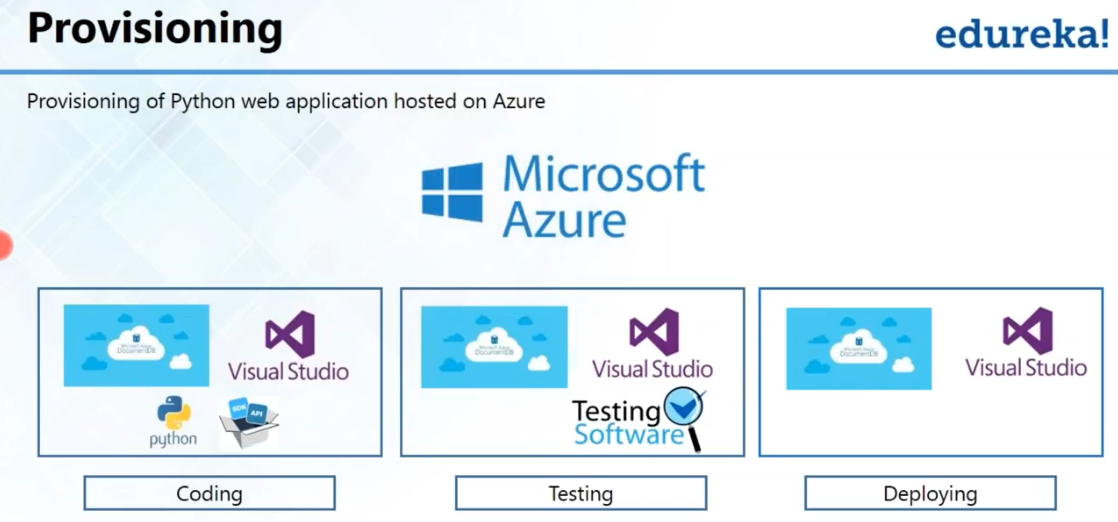


**Orchestration:**

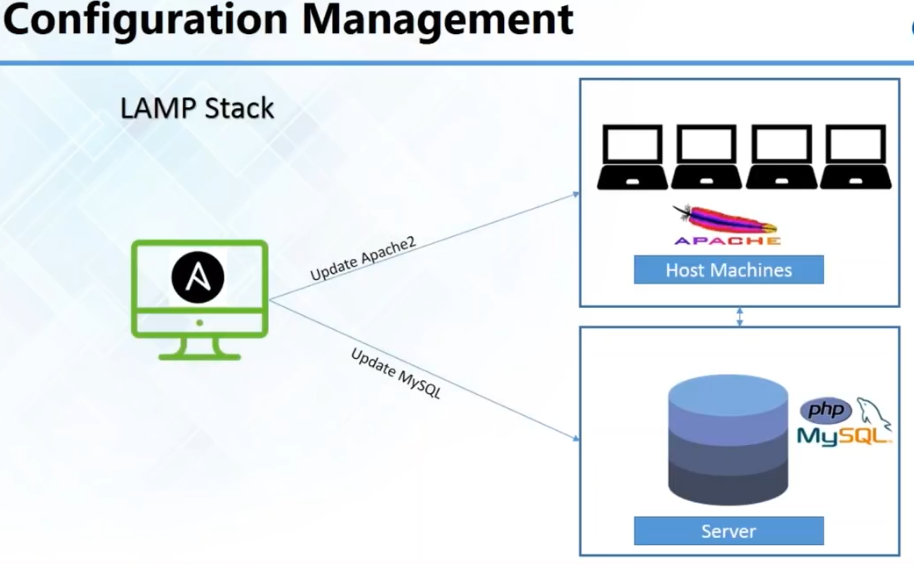


* If we have configuration on machines. We can know the interaction between the machines
* We need orchestration tool to manage

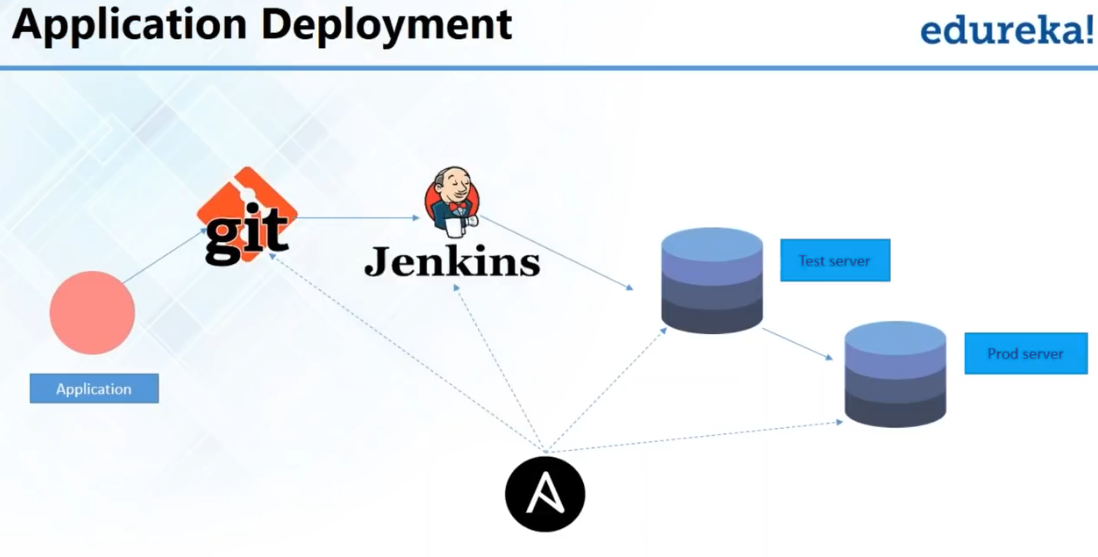
**Provisioning:**



* It means, ansible will make sure the necessity softwares properly installed on the machines



**Application deployment:**

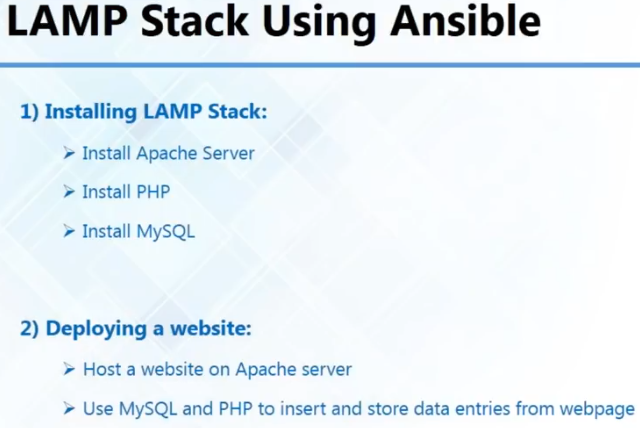


* As the changes are being done, we need to configure in test and prod servers
* Ansible keep monitoring the git and if changes occur it automatically configures in test and prod servers

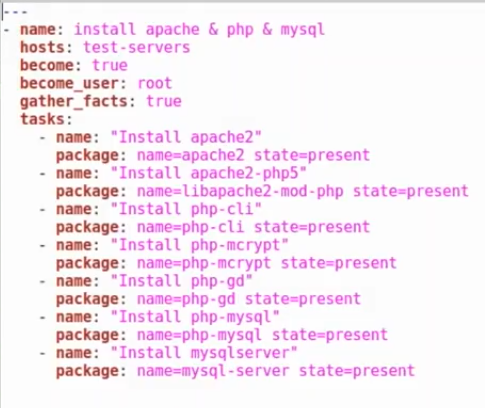
**Security:**



* We simply set up the security in playbooks
* We can set firewalls, lock the users etc. and policy as well



* Lamp stack yml file



* As above, hosts are the servers on which we want to install
* Become: true activates the privileges escalation
* It gathers the facts

