
ANSIBLE

Agenda

1.What is Ansible ?

2.Why Ansible ?

3.Ansible Use Cases

4.Architecture of Ansible

5.Configuration Management With Ansible

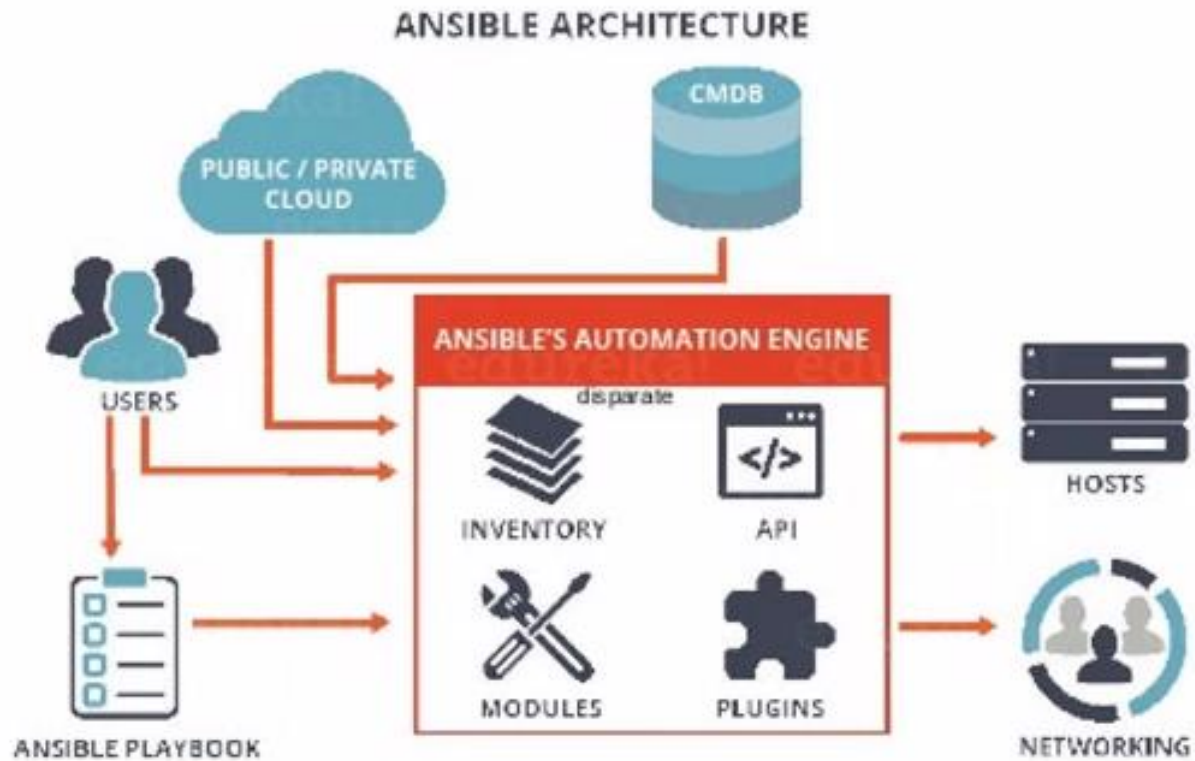
What is Ansible

- Ansible is an open-source configuration management and provisioning tool, similar to Chef, Puppet or Salt.
- It uses SSH to connect to servers and run the configured Tasks. Ansible lets you control and configure nodes from a single machine.
- What makes it different from other management software is that Ansible uses SSH infrastructure. The project was founded in 2013 and bought by Red Hat in 2015.

Ansible Use Cases

- Provisioning
- Configuration Management
- App Deployment
- Continuous Delivery
- Security & Compliance
- Orchestration

Architecture of Ansible

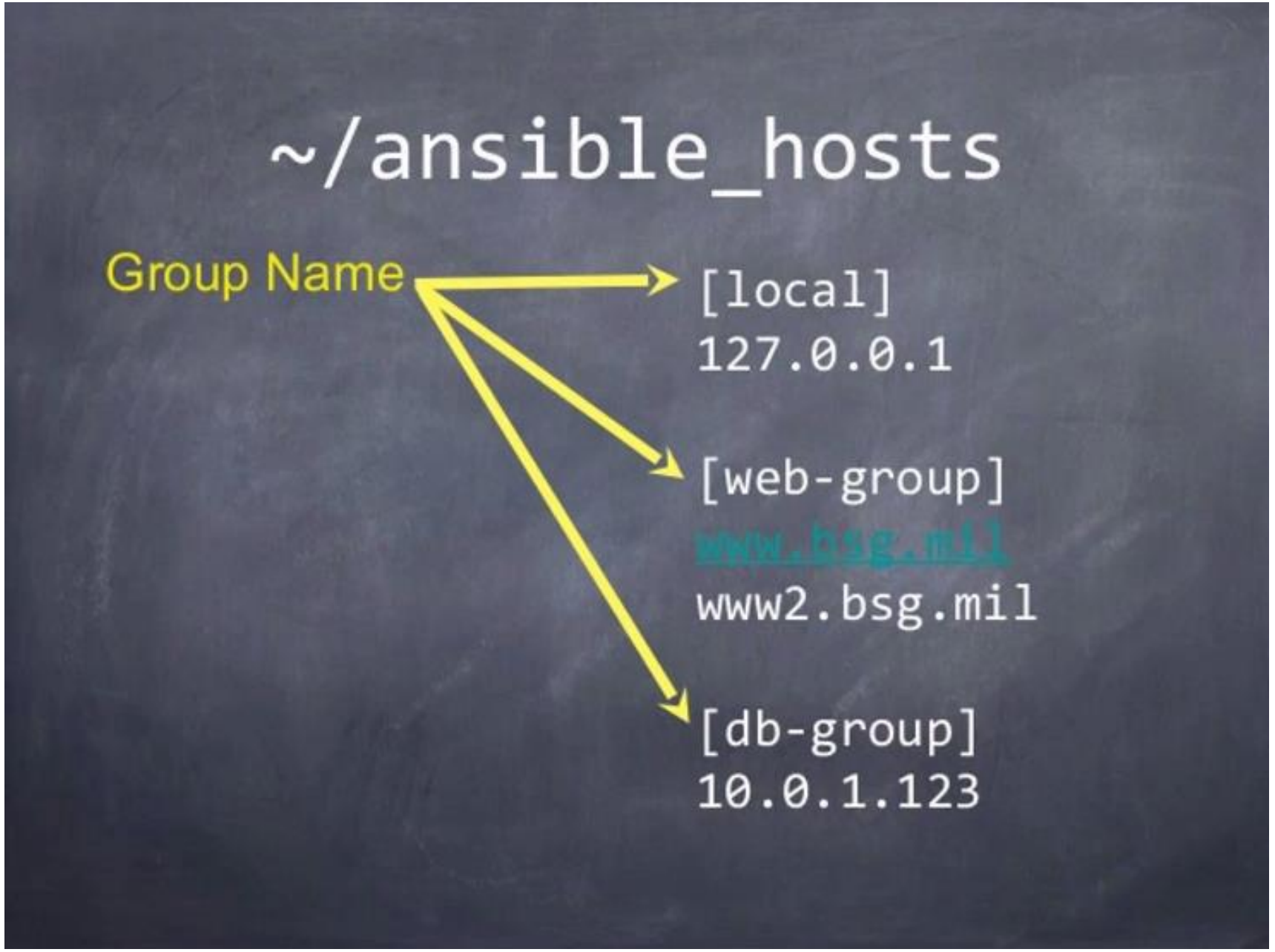


Example of an Inventory file

```
mail.example.com

[webservers]
foo.example.com
bar.example.com

[dbservers]
one.example.com
two.example.com
three.example.com
```


~/ansible_hosts
Group Name
[local]
127.0.0.1
[web-group]
www.bsg.mil
www2.bsg.mil
[db-group]
10.0.1.123

Playbook

Playbooks are simple YAML files. These files are descriptions of the desired state of your systems. Ansible then does the hard work of getting your systems to that state no matter what state they are currently in. Playbooks make your installations, upgrades and day-to-day management repeatable and reliable.

Playbooks are simple to write and maintain. Playbooks are written in a natural language so they are very easy to evolve and edit.

Playbook contains Plays.

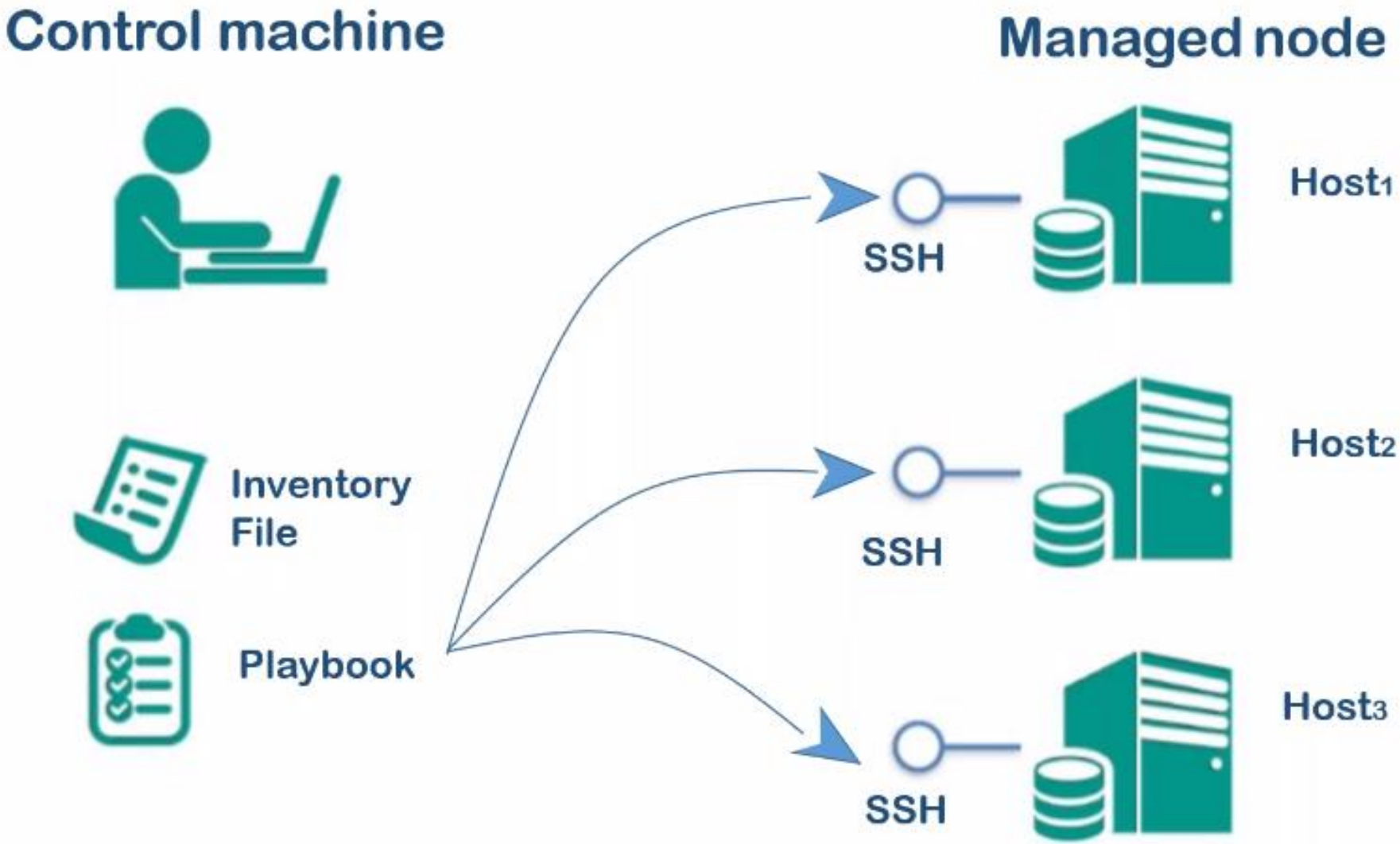
Plays contain tasks.

tasks call modules.

Example of an ansible playbook

```
---
- hosts: webservers
  remote_user: root
  tasks:
  - name: ensure apache is at the latest version
    yum: name=httpd state=latest
  - name: ensure apache is running
    service: name=httpd state=started enabled=yes
```

Playbooks – Push Mode



Ansible Playbook run

```
 09:55 PM $ ansible-playbook playbook.yml

PLAY [lamp] ********************************************************************

GATHERING FACTS ****************************************************************
ok: [192.168.33.3]

TASK: [apt update_cache=yes] ***************************************************
ok: [192.168.33.3]

TASK: [Ensure Apache/PHP is installed.] ****************************************
changed: [192.168.33.3] => (item=apache2,php5-common,libapache2-mod-php5,php5-cli)

TASK: [Ensure Apache is started.] **********************************************
ok: [192.168.33.3]

PLAY RECAP *********************************************************************
192.168.33.3               : ok=4    changed=1    unreachable=0    failed=0
```

Modules

There are over 1000 modules provided by Ansible to automate every part of the environment. Modules are like plugins that do the actual work in Ansible, they are what gets executed in each playbook task.

Each module is mostly standalone and can be written in a standard scripting language (such as Python, Perl, Ruby, Bash, etc.). One of the guiding properties of modules is idempotency, which means that even if an operation is repeated multiple times, it will always place the system into the same state.

Example of Modules

There are lots of modules such as :

Service, file, copy, iptables etc.

Any Module can be used as :

ansible 127.0.0.1 -m service -a "name=httpd state=started"

ansible localhost -m ping