Agenda

- What is Ansible?

- Why Ansible?

- How Ansible works?

- SSH

- installation

- Inventory

- Playbook

- Modules

- Roles

- Let's write scripts

- Ansible Galaxy

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.

21 January 2020

Why Ansible?

- **No Agent-** As long as the box can be ssh'd into and it has python, it can be configured with Ansible.

- **Idempotent-** Ansible's whole architecture is structured around the concept of idempotency. The core idea here is that you only do things if they are needed and that things are repeatable without side effects.

- **Declarative Not Procedural-** Other configuration tools tend to be procedural do this and then do that and so on. Ansible works by you writing a description of the state of the machine that you want and then it takes steps to fulfill that description.

- **Tiny Learning Curve-** Ansible is quite easy to learn. It doesn't require any extra knowledge.

How Ansible Works ?
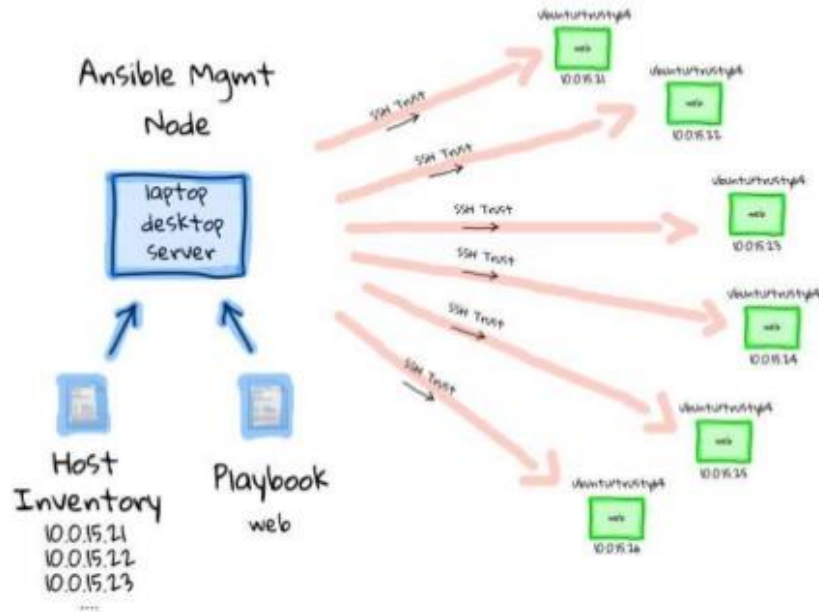
SSH configuration

Ansible installation

Inventory

- The Inventory is a description of the nodes that can be accessed by Ansible. By default, the Inventory is described by a configuration file, whose default location is in /etc/ansible/hosts

- The configuration file lists either the IP address or hostname of each node that is accessible by Ansible.

- Every host is assigned to a group such as web servers, db servers etc.

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

21 January 2020

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

- Playbook contains Plays.

- Plays contain tasks.

- tasks call modules.

21 January 2020

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
```

21 January 2020

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.

21 January 2020

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

21 January 2020

Roles

- Roles are a way to group tasks together into one container. We could have a role for setting up MySQL, another one for configuring iptables etc.

- Roles makes it easy to configure hosts. Any role can be performed on any host or group of hosts such as:

- hosts: all

  roles:

    - role_1

    - role_2

21 January 2020

Let's write ansible scripts

- simple yum install

- remote user

- variables

- debug

- Templates

- conditions

- loops

- handlers

21 January 2020

Ansible Galaxy

- *Ansible Galaxy* refers to the <u>Galaxy</u> website where users can share roles, and to a command line tool for installing, creating and managing roles.

- The ansible-galaxy command comes bundled with Ansible, and you can use it to install roles from Galaxy or directly from a git based SCM. You can also use it to create a new role, remove roles, or perform tasks on the Galaxy website.

```
$ ansible-galaxy install username.role_name
```

```
$ ansible-galaxy init role_name
```

```
hakase@ansible:~/project-lemp/roles$
hakase@ansible:~/project-lemp/roles$ tree .
.
├── common
│   ├── defaults
│   │   └── main.yml
│   ├── files
│   ├── handlers
│   │   └── main.yml
│   ├── meta
│   │   └── main.yml
│   ├── README.md
│   ├── tasks
│   │   └── main.yml
│   ├── templates
│   ├── tests
│   │   ├── inventory
│   │   └── test.yml
│   └── vars
│       └── main.yml
├── db
│   ├── defaults
│   │   └── main.yml
│   ├── files
│   ├── handlers
│   │   └── main.yml
│   ├── meta
│   │   └── main.yml
│   ├── README.md
│   ├── tasks
│   │   └── main.yml
│   ├── templates
│   ├── tests
│   │   ├── inventory
│   │   └── test.yml
│   └── vars
│       └── main.yml
└── web
    ├── defaults
    │   └── main.yml
    ├── files
    ├── handlers
    │   └── main.yml
    ├── meta
    │   └── main.yml
    ├── README.md
    ├── tasks
    │   └── main.yml
    ├── templates
    ├── tests
    │   ├── inventory
    │   └── test.yml
    └── vars
        └── main.yml

27 directories, 24 files
hakase@ansible:~/project-lemp/roles$
```