Introduction to Ansible

Agenda

1. What is Ansible
2. Why Ansible
3. Ansible Use Cases
4. Architecture of Ansible
5. Configuration Management With Ansible
6. Quick Demo

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.

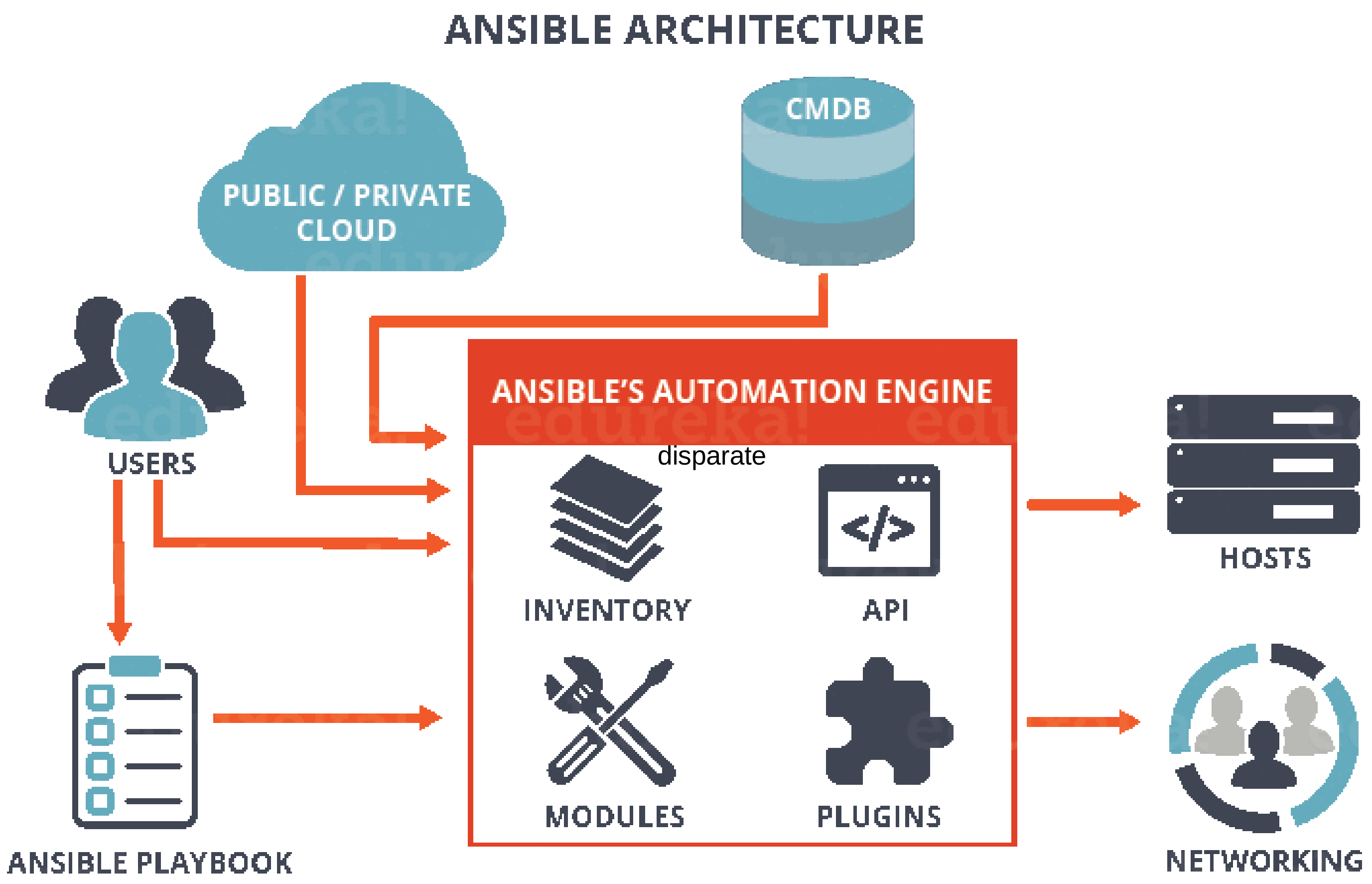
Why Ansible

- **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.

Ansible Use Cases

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

Architecture of Ansible



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. The inventory file can be in one of many formats such as yaml, INI 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
```

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
```

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
```

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
```

Companies that are using Ansible

twitter

CISCO

verizon

NASA

Atlassian

redhat

Configuration Management with Ansible

Ansible is the simplest solution for configuring the nodes. It’s designed to be minimal in nature, consistent, secure and highly reliable. Any developer, tester or IT manager can easily configure nodes. Any IT person can write playbooks easily.

Ansible configurations are simple data descriptions of your infrastructure (human readable) ensuring everyone on your team will be able to understand the meaning of each configuration task.

Ansible requires nothing more than a password or SSH key in order to start managing systems and can start managing them without installing any agent software.

DEMO