Ansible Master Class

Jabir Ali V.
CI/CD & Automation Team
Mettle Networks

20 May 2021, Presented as part of Faculty Training, MCA Program, APJ Abdul Kalam Technological University

The Need

A simple use case

- Setting up PCs for Lab examinations

Difficult Issues

- Changes we are making on every PC is repetitive
- Configurations demand to be uniform across all PCs
- There should not be missing of any packages
- If we intent some replacement later, we should do it all over
- When scaling is needed, we have to do it manually

The Solution: Ansible

Simple Open Source IT engine, which helps to leverage productivity by

- IT automation
- Application deployment
- Intra service orchestration
- Provisioning
- Configuration Management
- Continuous delivery
- Security compliance

Features

- Simplicity
  - Learn
  - Setup
  - Extend modules in any language
- Batteries included
  - In the form of modules
  - Ansible manages infrastructure 2K+ modules available
  - Provision cloud components using modules
  - Monitoring tools, source control, Network, collaboration tools

Features (Cntd.)

- Infrastructure as code
  - It represents system’s final state as YAML
- Code vs. data
  - It separates code from data
  - Use generic code from templates, add data using variables
- Idempotence
  - Property of an operation that can be done multiple time without changing the result beyond the initial application

Features (Cntd.)

- Idempotence
  - Different ways to achieve desired state based on situations

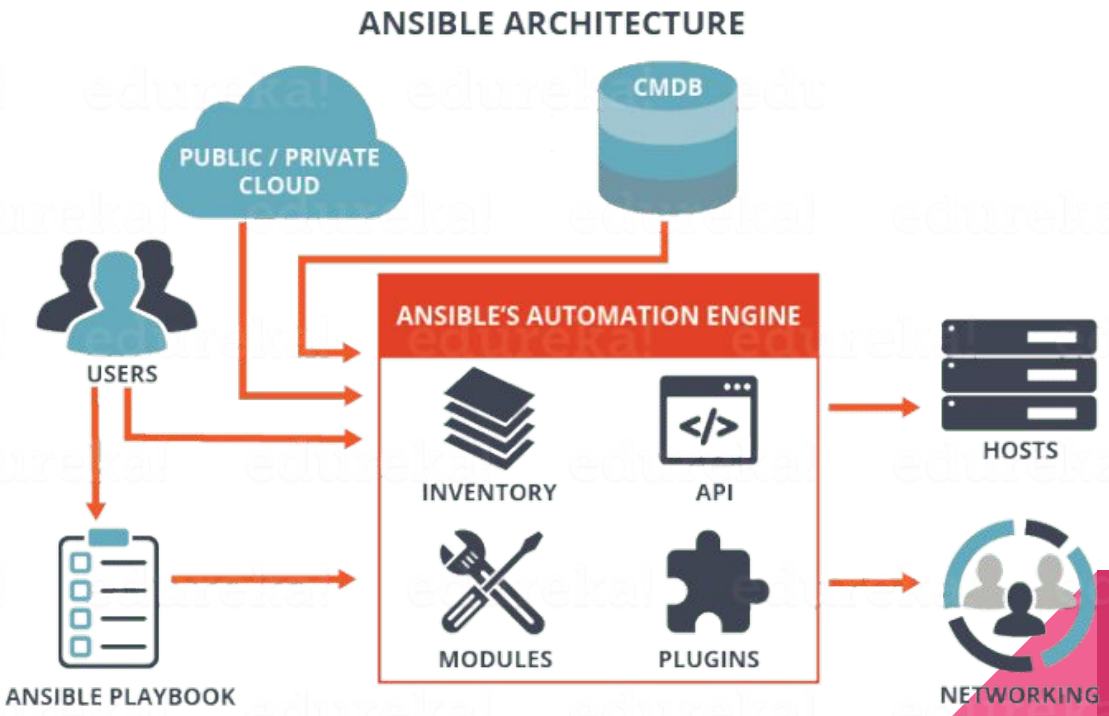
```
User
name=xyz
uid=5001
pass=pq
```

  - useradd, or usermod or passwd or nothing to achieve the above state

Why it is Different

- Agentless architecture (not like Puppetlabs or Chef)
- No custom PKI. SSH-based
- Configuration as data not Code
- Full Configuration management, orchestration and deployment

Under the Hood



Under the Hood (Cntd.)

- **Inventories**
  - List of hosts (nodes) along with their IP addresses, servers, databases etc. which needs to be managed.
- **APIs**
  - Used as transport for Cloud services, public or private
- **Modules**
  - Executed directly on remote hosts through playbooks.
  - Can control system resources, like services, packages, or files or execute system commands.eg: ping, shell, service, file etc .

Under the Hood (Cntd.)

- **CMDB**
  - It is a repository that acts as a data warehouse for IT installations
  - It holds data relating to a collection of IT assets (configuration items (CI)), as well as to describe relationships between such assets
- **Cloud**
  - It is a network of remote servers hosted on the Internet to store, manage, and process data, rather than a local server
  - We can launch your resources and instances on cloud and connect to your servers

Under the Hood (Cntd.)

- **Plugins**
  - Allows to execute Ansible tasks as a job build step.
    Plugins are pieces of code that augment Ansible's core functionality. Ansible ships with a number of handy plugins, and you can easily write your own. For example,
    - *Action* plugins are front ends to modules and can execute tasks on the controller before calling the modules themselves
    - *Cache* plugins are used to keep a cache of 'facts' to avoid costly fact-gathering operations
    - *Callback* plugins enable you to hook into Ansible events for display

      or logging purposes.

Under the Hood (Cntd.)

- **Networking**
  - Ansible can also be used to automate different networks.
  - It uses a data model (a playbook or role) that is separate from the Ansible automation engine that easily spans different network hardware
- **Hosts**
  - Node systems which are getting automated by Ansible
  - It can be any kind of machine – Windows, Linux, RedHat, etc.

Under the Hood (Cntd.)

For the basic task

```
- name: install nginx
  apt: name=nginx
```

Ansible will:

1. Generate a python script that installs the nginx package
2. Copy the script to pc1 pc2 and pc3
3. Execute the script on pc1 pc2 and pc3
4. Wait for the script to complete execution

Under the Hood (Cntd.)

- **Playbooks**
  - Simple files written in YAML format which describes the tasks to be executed by Ansible
  - Playbooks can declare configurations, but they can also orchestrate the steps of any manual ordered process
  - They can launch tasks synchronously or asynchronously

Prerequisites

- SSH : basic operation and commands and concept of public-key cryptography
  - ssh-keygen
  - ssh-copy-id
  - ssh
- YAML
- INI
- Any VM hypervisor (for hands on)
  - virtualbox

Prerequisites (Cntd.)

- **INI file**
  - Contains configuration information in a simple, predefined format.
  - It is used by Windows OSs and Windows-based apps to store information about the user's preferences and operating environment.
  - These files are plain text files with a basic structure comprised of properties and sections. Eg:

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

Prerequisites (Cntd.)

- **YAML**
  - Yet Another Markup language
  - Human Readable & Machine Readable
  - Can be called as a Superset of json (every json is parsable as yaml)
  - Closer to natural language
  - Self Documented Code

    Eg:

```yaml
some_key:
  some_other_key: some_val
  some_list:
    - item1
    - item2
```

Prerequisites (Cntd.)

- **YAML Syntax**
  - Start of file ---
  - Comments #
  - List - item or [item]
  - Maps key:value or {key:value}
  - line folding >

```
some_key:
    some_other_key: some_val
    some_list:
      - item1
      - item2
```

Python dictionary Equivalent

```
{'some_key': {'some_other_key': 'some_val'},'somelist':['item1','item2']}
```

Json Equivalent

```
{"some_key": {"some_other_key": "some_val"},"somelist":["item1","item2"]}
```

Ansible concepts

- **Control Node**
  - Any machine with Ansible installed
- **Managed Nodes**
  - The network devices (and/or servers) you manage with Ansible.
- **Inventory**
  - A list of managed nodes.
  - Also sometimes called a “hostfile”
- **Modules**
  - The units of code Ansible executes
  - Each module has a particular use
  - Eg: ping, apt, yum, user, copy, file, services

Ansible concepts (Cntd.)

- **Tasks**
  - The units of action in Ansible
  - You can execute a single task once with an ad hoc command
- **Playbooks**
  - Ordered list of tasks
  - Written in YAML
  - It preserves the order
- **Collections**
  - A distribution format for Ansible content that can include playbooks, roles, modules, and plugins
  - We can install and use collections through Ansible Galaxy.

First Ansible Playbook

```
---
- hosts: all
 tasks:
   - name: Make sure that we can connect to the machine
     ping:
```

- To run:

```
ansible-playbook -i inventory all first_play.yaml
```

- Equivalent Ad-hoc command:

```
ansible -i inventory all -m ping
```

Advanced Topics

- **Roles**
  - Collection of tasks to achieve a certain goal
  - Mechanism of breaking playbooks into multiple files
- **Templates**
  - We use jinja2 for this
- **Ansible Galaxy**
  - Repository for different ansible roles
- **Ansible Tower**
  - Web based solutions that makes ansible even more easy to use

Resources

- Installation Guide
  - Installation Guide — Ansible Documentation
- Hands-on Environment Setup
  - https://github.com/jabir366/MasterClass-Ansible
- Getting started
  - https://docs.ansible.com/ansible/latest/user_guide/intro_getting_started.html

Thanks.