

Ansible

Why, What and How ;)

Brice Ekane - (brice.ekane@univ-rennes.fr)

ISTIC Rennes - France

2024-2025

git clone <https://github.com/bekane/ansible.git>

Full copy from Pr. Olivier Barais courses

with some materials from Philippe merle, Théo Giraudet, Jamie Duncan, and

Avinash Pawar

Talk outline

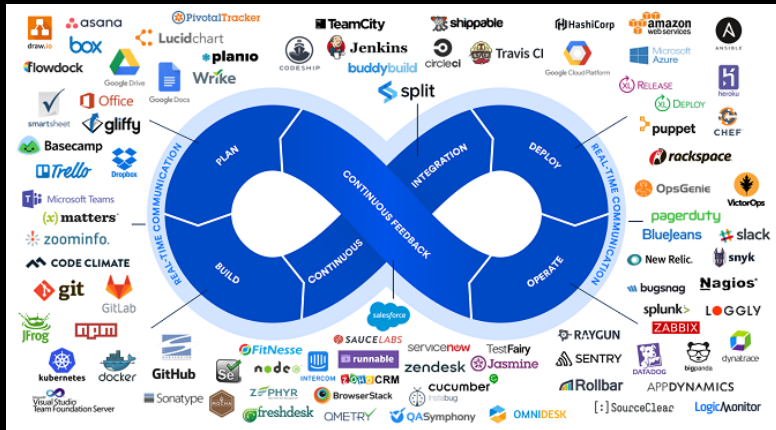
- ▶ Why ?
 - ▶ Automation ?
 - ▶ Why Infrastructure as code ?
- ▶ What ?
 - ▶ Ansible main concepts
- ▶ How ?
 - ▶ Best practices
- ▶ Related work

Why (IAC) ?

AUTOMATION == DOCUMENTATION

If done properly, the process of automating a process can become the documentation for the process. Everything in Ansible (and most IAC tool) revolves around this core concept.

The DevOps Jungle



The Jungle of Infrastructure as Code



Agenda

1. Ansible Introduction
2. Why Ansible?
3. Ansible Architecture / How Ansible Works?
4. Ansible Components/Concepts
5. Ansible – Playbooks
6. Ansible – Role
7. Demo

Introduction - What is Ansible?

- “ - Simple, agent-less and powerful open source IT automation Tool ” -
- Provisioning - Configuration Management - Application Deployment -
- Continuous Delivery - Security & Compliance - Orchestration - Products:
- Ansible CLI - Ansible Tower

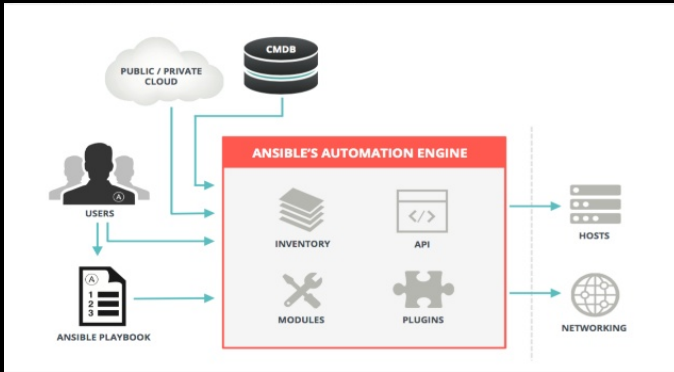


Why Ansible?

- ▶ It is very simple and powerful
- ▶ Agent-less – No need for agent installation and management
- ▶ Secure – SSH based connection
- ▶ Python / yaml based
- ▶ Highly flexible and configuration management of systems.
- ▶ Large number of ready to use modules available for system management
- ▶ Custom modules can be added if needed
- ▶ Human readable & Self documenting
- ▶ Idempotent

How Ansible Works?

- ▶ Ansible works by connecting to your nodes and pushing out small programs, called “Ansible Modules” to them. Ansible then executes these modules (over SSH by default) and removes them when finished.



How Ansible Works?

- ▶ Send commands via SSH
- ▶ Requires installation on the master side only
- ▶ Encrypted exchange
- ▶ May require Python on slave side (which can be installed by Ansible)
- ▶ Supported by RedHat
- ▶ Requires a key-secured SSH connection

Ansible - Installation

```
sudo apt-get install ansible
```

Ansible Main Concepts

- ▶ Inventory
 - ▶ File where you declare the list of hosts and groups
- ▶ Modules
- ▶ Playbooks : A playbook is a YAML file where you run a series of tasks on the hosts
- ▶ Variables : Ansible provides a mechanism for overriding variables
- ▶ Templates
- ▶ Roles
 - ▶ are ways of automatically loading certain vars_files, tasks, templates, and handlers based on a known file structure. Grouping content by roles also allows easy sharing of roles with other users

Ansible - Hosts

Example Hosts File

```
[servers]  # Name of the group
server1
server2

[bdd]
10.0.0.1  # Domain name or host IP
```

- ▶ The hosts file contains the IPs or domain names of the remote machines on which to run the playbooks.
- ▶ The file can be in either YAML or INI format.
- ▶ It is possible to restrict the execution of a state to a specific group.

Simple NGINX Install Playbook

Ansible Playbook Example

```
- name: Install NGINX
  hosts: webservers
  become: true
  tasks:
    - name: Update apt repository
      apt:
        update_cache: yes
    - name: Install NGINX
      apt:
        name: nginx
        state: present
    - name: Ensure NGINX is running
      service:
        name: nginx
        state: started
        enabled: yes
```

Simple 'site.yml' Example for Apache Role

Ansible Playbook

```
---
- name: Configure Apache Web Servers
  hosts: webserver
  become: true

  roles:
    - role: apache
      vars:
        apache_listen_port: 80
        apache_document_root: /var/www/html
        apache_server_admin: admin@example.com
```


Best Practices

- ▶ https://docs.ansible.com/ansible/2.8/user_guide/playbooks_best_practices.html
- ▶ <https://github.com/redhat-cop/automation-good-practices>
- ▶ <https://people.redhat.com/bdumont/Central-Region-Lunch-n-Learns/Ansible>