**Ansible documentation**

Ansible is an IT automation tool and configuration management tool.

Let’s say, you want to install ngnix in 100 different computers with same configuration. Its difficult to do manually. In such cases, we create configuration files and provision infra easily with Ansible.

Advantages of Ansible:

IT Automation

Configuration Management

Provision of infrastructure on cloud

What is Ansible?

Ansible delivers simple IT automation that ends repetitive tasks and frees up DevOps team for more strategic work.

It automates configuration management, cloud provisioning, application deployment, infra service orchestration, and many other IT needs.

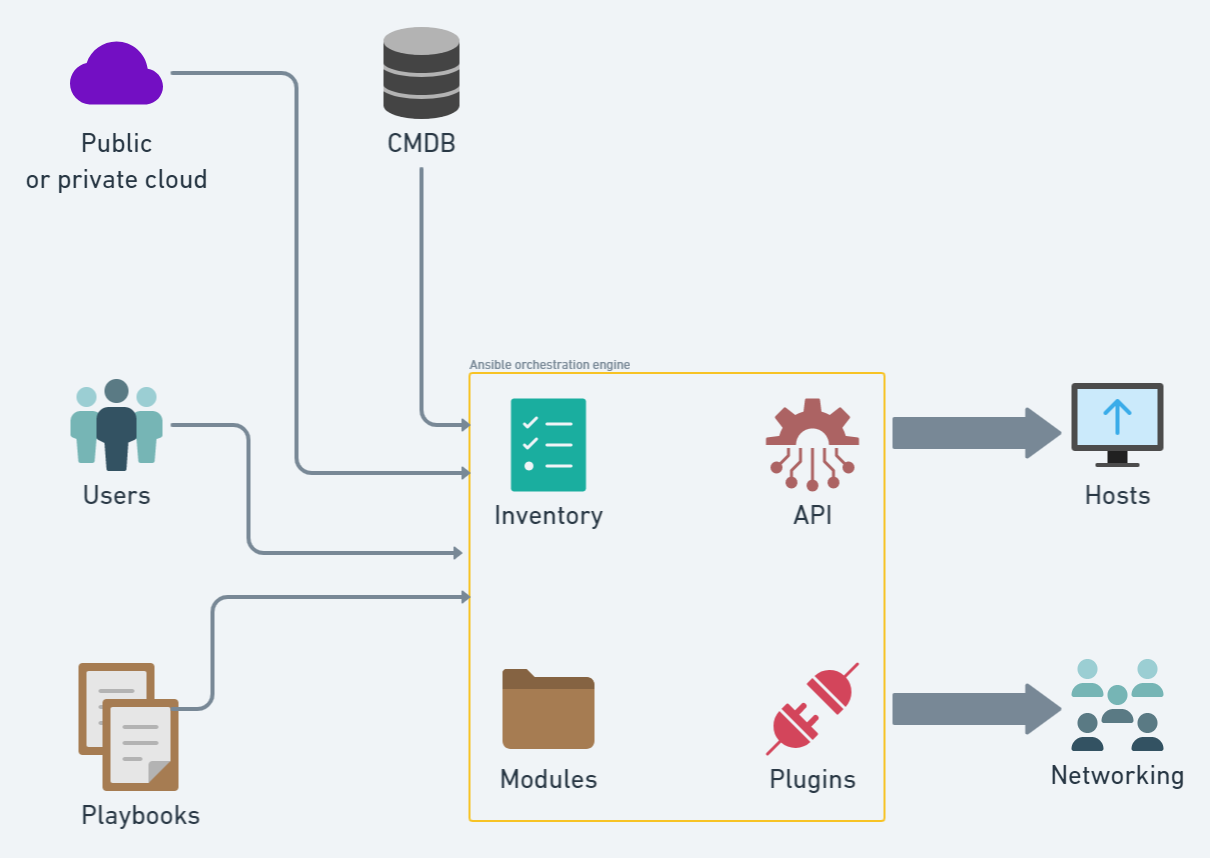
When Ansible is used as a configuration management tool, it is used to store the current state of our systems and helps us to maintain that state, it makes changes and deployment faster, removing the potential for human error while making system management predictable and scalable.

What is configuration management?

Configuration management is a process for maintaining computer systems, servers and software in a desired consistent state.

Its way to make sure that a system performs as its expected to as changes made over time.

**Ansible architecture and How works?**

****

**How Ansible works?**

1. Ansible does not use any agent. Yes, you heard it right! Ansible also does not use any additional custom security infrastructure, which makes it very flexible, and it can run on anything.
2. It manages entities/servers via SSH (secure shell)
3. Ansible works by connecting pour nodes/servers and pushing out small program’s vis SSH. Called “ansible modules” to them. These programs are written to be resource models of the desired state of the system. Ansible then executes these modules (over SSH by default) and removes them when finished.
4. Ansible modules can be written any language that can return JSON (Ruby, python, Golang, bash etc.,)
5. There’s also various python APIs for extending Ansible connection types (SSH is not the only transport possible)

**Components of ansible**

**Inventory**: Inventory is lists of nodes or hosts having their IP addresses, databases, servers, etc. which need to be managed.

**APIs**: The ansible APIs works as the transport for the public or private cloud services.

**Modules**: Ansible connected the nodes and spread out the ansible module’s programs. Ansible executes the modules and removed after finished. These modules can reside on any machine; no database or servers are required here. You can work with the chose text editor or terminal or version control system to keep track of the changes in the content.

**Hosts**: In the ansible architecture, hosts are the node systems, which are automated by ansible and any machine such as RedHat, Linux, Windows etc.,

**Networking**: ansible is used to automate different networks, and it uses the simple, secure, and powerful agentless automation framework for IT operations and development. It uses a type of data model which separated from the ansible automation engine that spans the different hardware quite easily.

**Cloud**: A clous is a network of remote servers on which you can store, manage, and process the data. These servers are hosted on the internet and storing the data remotely rather than the local server. It just launches the resources and instances on the cloud, connect them to the servers, and you have good knowledge of operating your tasks remotely.

**CMDB**: CMDB is a type of repository which act as data warehouse for the IT installations,

**Plugins**: Plugins is a piece of code that expends the core functionality of ansible. There are many useful plugins, and you also can write your own.

**Playbooks**: Playbooks consists of your written code, and they are written in YAML format, which describes the tasks and executes through the ansible. Also, you can launch the tasks synchronously and asynchronously with playbooks.

**Ansible concepts:**

**Control Node**:

Any machine with ansible installed. We can run ansible commands and playbooks and playbooks by invoking the ansible or ansible playbook command from any control node. We can use any computer that has a python installation as control node laptops, shared desktops, and servers can all run ansible. However, we cannot use a windows machine as a control node. We can have multiple control nodes as well.

**Managed Nodes:**

The network devices (and/or severs) we manage with ansible. Managed nodes are also sometimes called “hosts”. Ansible is not installed on managed nodes.

**Collections**:

Collections are a distribution format for ansible content that can include playbooks, roles, modules, and plugins. We can install and use collections through ansible Galaxy.

**Inventory**:

A list of managed nodes. An inventory file is also sometimes called a “host file”, Our inventory can specify information like IP address for each managed node. An inventory can also organize managed nodes, creating and nesting groups for easier scaling, typically at /etc/ansible/hosts, provide a custom inventory path using -I parameter when running commands & playbooks.

**Tasks**:

The units of action in ansible. We can execute a single task once with an ad hoc command.

**Modules**:

The units of code ansible executes. Each module has a particular use, from administrating users on a specific type of database to managing VLAN interface on a specific type of network device. We can invoke a single module with a task or invoke several different modules in a playbook.

**Playbooks**:

Ordered lists of tasks, saved so we run those tasks in that order repeatedly. Playbooks can include variable as well as tasks. Playbooks are written in YAML and are easy to read, write, share, and understand.

**Controller node setup:**

A system where the ansible is installed and configured to connect and execute commands on nodes.

**Generating Custom SSH keys**

**Setting up ssh:**

sudo apt-get install openssh-server

**Generating new ssh keys:**

ssh-keygen

**ssh-copy-id hostname (it itf password- based)**

ssh-copy-id -I ~/.ssh/my\_cutoms\_key user@host

**Time to check SSH connection**

ssh -I ~/.ssh/my\_custom\_key user@host

**how to install Ansible?**

There are multiple ways to install ansible, here showing ubuntu

Example:

$ sudo apt update

$ sudo apt install software-properties-common

$sudo add-apt-repository –yes –update ppa:ansible/ansible

$ sudo apt install ansible

**Check ansible version:**

Ansible -version

**Testing Connectivity with Managed Nodes**

**Using a custom SSH key, checking remote connection**

Ansible all -m ping –private-key= ~/.ssh/my\_custom\_key

**For playbook:**

Ansible-playbook myplaybook.yml –private-key= ~/.ssh/my\_custom\_key

**Using password**

Ansible all -m ping –ask-pass

Ansible-playbook myplaybook.yml –ask-pass

**Why we need Ansible**

Inventory in ansible