ANSIBLE

Ansible is an open-source tool used for configuration management, application deployment, along with infrastructure orchestration.

Ansible is a push-based configuration management tool

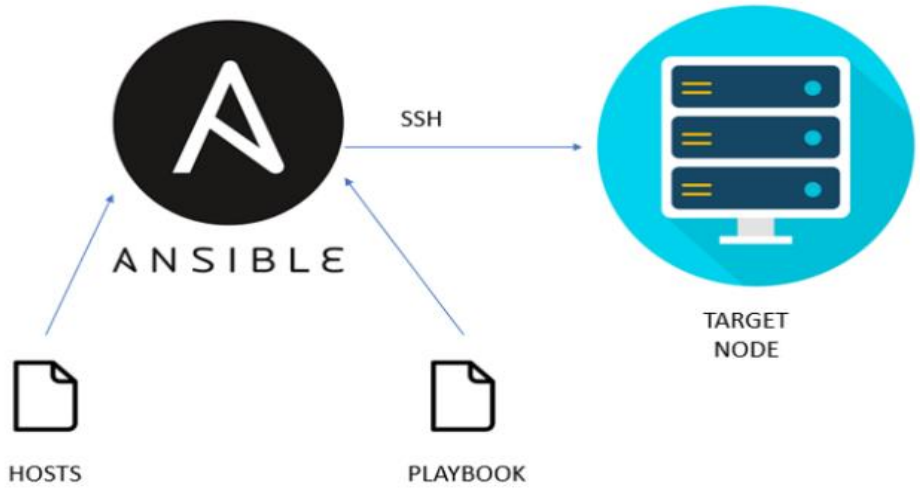
1. Ansible does not require any dedicated agent running on the target host machines.
2. The requirement for host machine is python installed on them and a proper ssh connection between the controller and the host machines.

Advantages:

- free/open source
- Very simple to setup
- Agent less configuration tool

Architecture:

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



Plugins: While modules execute on the target system, plugins execute on the control node. Plugins offer options and extensions for the core features of Ansible

Controller Machine: Controller is a Machine where Ansible is installed

Inventory: Inventory is the place where we put information about all our managed machine host machines

Default Location: /etc/ansible

Installation:

Ubuntu:

sudo apt-add-repository ppa:ansible/ansible
sudo apt update
sudo apt install ansible

Redhat:

sudo yum update -y
sudo yum -y install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
sudo yum install ansible

Ansible INVENTORY:

Static Inventory

- It is a file which contains the ip and configuration of target host machines on which we want execute our playbooks
- Default location of host file is/etc/ansible/hosts

AD-HOC Commands

ansible <hosts> -m <module> -a <arguments of the module>

- ansible all -m ping
- ansible frontend -m ping
- ansible backend -m ping
- ansible all -m shell "touch f1"
- ansible all -m shell -a ls
- ansible all -m shell -a uptime

Ansible Playbook

Playbook: Playbooks are list of tasks written in yaml format that gets executed on host machines

- Can contain n number of plays
- Each play is designated to run n number tasks
- Each task is designated to execute a single module (i.e., only one module per task)

Executing Playbook:

Ansible-playbook <file_name>.yaml

YUM and APT Module:

state:

- present will simply ensure that a desired package is installed.
- latest will update the specified package if it's not of the latest available version.
- absent will remove the specified package.

update_cache: yes tells Ansible's apt module to refresh the caches before applying whatever change is necessary (i.e., apt-get update)

Ansible facts:

Ansible facts are data gathered about host machines and returned back to the controller. Ansible facts are stored in JSON format and setup module is used to gather all the facts

ansible all -m setup

ansible all -m setup | grep "distribution"

ansible all -m setup | grep "pkg"

ansible all -m setup | grep "architecture"

ansible all -m setup | grep "cpu"

Ansible Register and Debugging:

Ansible register is a way to capture the output from task execution and store it in a variable.

Ansible run a task only if previous task is successfully changed using register module

- Register module will save the output of a task to a variable in JSON format.
- We can register the output of one task and on another task, we can check whether information about state in variable is changed

Ansible Run specific tasks / plays

Using tags, we can run specific task or play in a playbook

To run only task with the tag

ansible-playbook <playbook_name>.yaml --tags="unzip "

ansible-playbook <playbook_name>.yaml --tags="git,unzip"

To skip a tagged play / task

ansible-playbook <playbook_name>.yaml --skip-tags="git"

ansible-playbook <playbook_name>.yaml --skip-tags="git,unzip"

Ansible Roles:

Ansible Roles are a way of breaking down a playbook into multiple files. Therefore, simplifying writing complex playbooks and making them easier to reuse.

Default location of roles is/etc/ansible/roles (Create the directory if it is not present)

Ansible provides a feature called Ansible Galaxy that helps you create roles.

ansible-galaxy init → To create a Role

```
docker
├── README.md
├── defaults
│   └── main.yml
├── files
│   └── file
├── handlers
│   └── main.yml
├── meta
│   └── main.yml
├── tasks
│   └── main.yml
├── templates
│   └── template.j2
├── tests
│   ├── inventory
│   └── test.yml
└── vars
    └── main.yml
```

Tasks — Contains the main list of tasks that are to be executed by the role.

Handlers — The handlers directory is used for storing Ansible handlers. Handlers are the tasks which are executed only if they are invoked by another task

Defaults — Contains the default variables that are going to be used by this role. It has the least precedence in Ansible Roles

Vars — This directory consists of other variables that are going to be used by the role.

Files — Contains static files that can be copied to the host machines by this role.

Meta — Defines metadata for this role. Basically, it contains files that establish role dependencies.

templates – Similar to files templates are used to copy files to host machines but with dynamic values. This usesjinja2 format templating.

Tomcat Installation Role: https://github.com/opqtech/ansible-tomcat-role

Docker Installation Role: https://github.com/opqtech/ansible-docker-role

Dynamic Inventory:

With scalable cloud infrastructure it is difficult to maintain a static inventory file as we might miss those instances that are newly created or created by auto-scaling. Ansible provides a way to capture these new host machines with Dynamic Inventory concept using plugins Ansible dynamic inventory plugin issupported from Ansible 2.7 version.

Pre-requisites:

1. Install boto3

   sudo yum install -y python3-pip

   pip install boto3

2. Install AWS CLI and configure with access_Key and secret_Key

aws_ec2 Plugin:

```
plugin: aws_ec2

regions:
- ap-south-1

keyed_groups:
- key: tags
  prefix: ins_tags
- key: architecture
  prefix: arch
- key: instance_type
  prefix: type
- key: key_name
  prefix: key
```

Configuration:

```
[defaults]
inventory      = /etc/ansible/aws_ec2.yaml

[inventory]
enable_plugins = aws_ec2
```

Commands:

ansible-inventory --graph

ansible-inventory --list

ansible <tag> -m ping