**Ansible:** (Ansible which is an open source automation platform whereas [Terraform for infrastructure provisioning](https://medium.com/@mitesh_shamra/infrastructure-as-a-code-with-terraform-e7021bf28d7d)).

Ansible does ***configuration management****,* ***application deployment***, along with ***infrastructure orchestration***. Ansible is **procedural** rather than **declarative**.

**In ansible**, we **define** **what we want to do** and **ansible go through each and every step for that**.

**In terraform**, we **specify what state we want to achieve,** and **it makes sure we are at that state by creating, modifying** or **destroying needed resources**.

Ansible doesn’t manage any state so we need to define how we want to keep track of created resources using tags or other properties while terraform keeps the state of infrastructure, so we don’t need to worry about duplicate resource creation.

Generally, it is recommended **terraform for provisioning the infrastructure**, and **Ansible for configuring the software**.

Once upon a time, managing servers reliably and efficiently was a challenge. System administrators managed server by hand, installing software manually, changing configuration and managing services on servers. As managed servers grew and managed services become more complex, scaling manual process was time-consuming and hard.

Then came **Ansible** which is **helpful in creating** the **group of machines**, define how to **configure them**, **what action** to be **taken on them**. **All these configurations** and **actions** can be **triggered** from a **central location** which can be your **local system** (named ***controller machine***).

Ansible **uses** **SSH** to **connect** to **remote hosts** and **do** the **setup**, **no software needed** to be **installed beforehand on a remote host**. **It’s simple, agentless, powerful and flexible**. It **uses** **YAML** in **form** of **ansible playbook**. **Playbook** is **a file** where **automation** is **defined through tasks**. A **task** is a **single step** to be **performed like installing a package**.

Ansible works by **connecting** to **remote hosts** (using SSH) defined in **inventory file**, which contains information about servers to be managed. Ansible then executes defined modules or tasks inside a playbook. Execution of playbook which is called the play. We can use predefined organised playbook called roles, which are used for sharing and reusing a provisioning.

**The terminology used in ansible:**

* **Controller Machine**: **Machine** where **Ansible is installed.**
* **Inventory**: **Information** regarding **servers** to be **managed.**
* **Playbook**: **Automation** is **defined** using **tasks defined** in **YAML format**.
* **Task**: **Procedure** to be **executed**.
* **Module**: **Predefined** commands **executed directly** on **remote hosts.**
* **Play**: **Execution** of a **playbook.**
* **Role**: a **Pre-defined** way for **organizing playbooks.**
* **Handlers**: **Tasks** with **unique names** that will only be **executed** if **notified by another task.**

Installing pip first using easy\_install and then ansible using pip on Mac OS. Please [look here](https://docs.ansible.com/ansible/latest/installation_guide/intro_installation.html) to install for other platforms.

sudo easy\_install pip

sudo pip install ansible

Once above command executed, run command below to make sure that ansible is installed properly.

ansible --version

The output should be something like below.

ansible 2.5.3  
config file = None  
configured module search path = [u'/Users/mitesh/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']  
ansible python module location = /Library/Python/2.7/site-packages/ansible  
executable location = /usr/local/bin/ansible  
python version = 2.7.10 (default, Oct 6 2017, 22:29:07) [GCC 4.2.1 Compatible Apple LLVM 9.0.0 (clang-900.0.31)]

Ansible reads the ssh keys form ~/.ssh/id\_rsa. We need to make sure we have public key setup on all remote hosts as we already done using terraform while creation of a [remote EC2 instance](https://medium.com/@mitesh_shamra/manage-aws-vpc-with-terraform-d477d0b5c9c5).

For running ansible command, we need inventory file which is expected to be at a specified path: “/etc/ansible/hosts”. We can change its path using ansible config file (**ansible.cfg** file) in ansible workspace and define inventory file path there. We need to define username which we are going to use during ssh in ansible config file.

File: ansible.cfg  
[defaults]  
inventory = ./inventory  
remote\_user = ec2-user

Create an **inventory** file and add the IP address (dummy)of a remote host.

File: inventory  
[all]  
18.191.176.209[server]  
18.191.176.209

Once this is done, let’s execute below command to ping all given remote host.

ansible all -m ping

Ansible executes ping command to a remote host and gives below output:

18.191.176.209 | SUCCESS => {  
"changed": false,  
"ping": "pong"  
}

We can even create groups in the inventory file and execute ansible commands by replacing all with a group name. In below example, the server is our group name specified in the inventory file.

ansible server -m ping

Let’s look at playbooks to execute a series of actions. We need to make sure we define playbooks as idempotent so that they can run more than once without having any side effects. Ansible executes playbook in a sequential manner from top to bottom.

Sample playbook is like:

---  
- hosts: [hosts]  
 tasks:  
 - [first task]  
 - [second task]

We are going to create a directory on our remote node using playbook for all hosts. Below mentioned playbook will create test directory in /home/ec2-user path.

---  
- hosts: all  
 tasks:  
 — name: Creates directory  
 file: path=/home/ec2-user/test state=directory

When we execute above playbook using command “**ansible-playbook playbook.yml**” we get below result. In this, the first result is gathering facts. This happens as ansible executes a special module named **“setup”** before executing any task. This module connects to a remote host and gathers all kinds of information like IP address, disk space, CPU etc. Once this is done, our create directory task is executed to create the test directory.

PLAY [all] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*TASK [Gathering Facts] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
ok: [18.191.176.209]TASK [Creates directory] \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
changed: [18.191.176.209]PLAY RECAP \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
18.191.176.209 : ok=2 changed=1 unreachable=0 failed=0

There are many modules and commands available to be executed on remote hosts. With ansible, we can do a server setup, software installation and lot more tasks.