**Ansible**

* Ansible is a radically simple IT automation platform that makes your applications and systems easier to deploy. Avoid writing scripts or custom code to deploy and update your applications— automate in a language that approaches plain English, using SSH, with no agents to install on remote systems.
* Ansible is a tool which can automate the configuration
* Administrator can write simple codes in YAML and these codes are deployed on to no.of servers, which configures them to the desired state.
* It automates:
* Configuration management – configuring systems
* Orchestration – which means it brings together a no.of applications and decides an order in which these are executed .
* Deployment of the applications
* Unlick Puppet or chef ansible is a type of push config management tool
* Master/Slave/Control node where we write our configuration file, it is responsible for pushing this code to Client/Slave nodes when required.

**Installation:**

***Installing Ansible on AWS:***

It involves following steps

1. Install Ansible on Master/Control node
2. Configure SSH access to Ansible Client/Slave
3. Setting up ansible Host and Test connection
4. Install Ansible on Master/Control node

Set Up in AWS Account:

* Lunch 2 instances of **Ubuntu** t2.micro type and allow all traffic.
* Rename instances one as Master and One as Client

Using Powershell

Connect to EC2 instances using following command

ssh -i "<path of pem key>" <user\_name>@<Private\_dns>

**On Master Node:**

Install ansible using following commands

Ref: <https://docs.ansible.com/ansible/latest/installation_guide/installation_distros.html>

$ sudo apt update

$ sudo apt install software-properties-common

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

$ sudo apt install ansible

**On Client Node**

Install Python using following commands

$ sudo apt update

$ sudo install python

1. Configure SSH access to Ansible Client/Slave

Keyless Access from Master to Slave/ Control node to Host nodes:

Generate ssh key on Master node:

$ Ssh-keygen

$ cat id\_rsa.pub

Copy Publick key(the output of above command) of Master node into authorized\_keys file on client node

* Authorized\_keys file is available in .ssh folder

To Test: on master node run the following command

Sudo ssh <username>@<public ip of client node>

1. Setting up ansible Host and Test connection

On Master Node:

Sudo vi /etc/ansible/hosts

Add the following lines at the end

[<groupname>]

<client\_name> ansible\_ssh\_host=<private\_ip> ansible\_ssh\_user=<username> ansible\_ssh\_pass=<password>

Ex:

[httpd\_servers]

Client1 ansible\_ssh\_host=10.5.34.29 ansible\_ssh\_user=ubuntu

Client2 ansible\_ssh\_host=10.5.34.178 ansible\_ssh\_user=ubuntu

Check Master is communicating with Slave or not with the following command

Ansible -m ping all/groupname/hostname

**Ansible Playbook to Install Apache Tomcat on Client Nodes:**

1.yml

++++++++++++++++++++++++++++++++++++++++++++++++

Copy below code to run your first playbook

---

- name: sample playbook

hosts: ansible\_clients

remote\_user: root

become: true

tasks:

- name: install httpd

apt: name=apache2 update\_cache=yes state=latest

2.yml

+++++++++++++++++++++++++++++++++++++

---

- name: sample playbook

hosts: ansible\_clients

remote\_user: root

become: true

tasks:

- name: install httpd

apt: name=apache2 update\_cache=yes state=latest

- name: content

copy: content="Web Server Installed Successfully…" dest=/var/www/html/index.html