**Ansible**

Ansible is an IT automation tool. It can configure systems, deploy software, and orchestrate more advanced IT tasks such as continuous deployments or zero downtime rolling updates.

Ansible’s main goals are **simplicity and ease-of-use**. It also has a strong focus on security and reliability, featuring a minimum of moving parts, usage of OpenSSH for transport and a language that is designed around auditability by humans–even those not familiar with the program.

* It is a configuration and Management tool
* Configuration: RAM, Storage, OS, Software and IP address
* Management: Update, Delete, Add
* Ansible is simple open source IT engine which automates application deployment
* Orchestration(everything goes in a flow), Security and compliance.
* Uses YAML scripting language which works on KEY-VALUE pair
* Ansible GUI is called Ansible tower. It was just drag and drop
* Used python for back end

Ansible : PUSH CHEF: PULL

PUSH: if we have many servers then it will push the notification for updates in all devices.

PULL: it will go to client server and ask for the notifications for update

**Advantages**

* Very consistent and light weight and no constrains regarding the OS or underlying HW
* Secure less due to agent less capability and open SSH security features.
* Doesn’t require any special system skills to install and use it (YAML)
* Push mechanism

**Disadvantages**

* Ansible does not have any notion of state like other automation tools such as Puppet
* Ansible does not track any dependencies and simply executes sequential tasks and stops when tasks finish, fail, or any error comes.
* Ansible has external dependencies to python modules

**Creating Inventory**

To create an inventory

**create a directory in that create one file and open that file by using cat or vi command**

#<hostname> ansible\_host=<host IP> ansible\_ssh\_pass=<host password>

EX: #hnncentos9 ansible\_host=172.16.238.232 ansible\_ssh\_pass=hnnbatch3

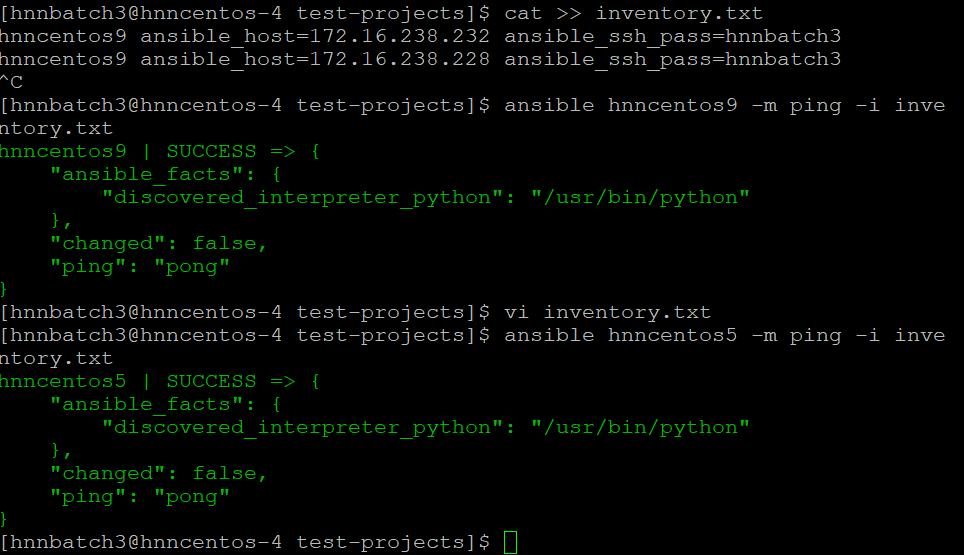
**We will use Ansible to make sure that all the hosts are reachable**

#ansible <hostname> -m ping -i <inventory file name>

#ansible hnncentos9 -m ping -i inventory.txt

**If you want to ping all the servers at a time**

#ansible all -m ping -i inventory.txt



**YAML**

We use YAML because it is easier for humans to read and write than other common data formats like XML or JSON. Further, there are libraries available in most programming languages for working with YAML.

For Ansible, nearly every YAML file starts with a list. Each item in the list is a list of key/value pairs, commonly called a “hash” or a “dictionary”. So, we need to know how to write lists and dictionaries in YAML.

**All members of a list are lines starting with “-”**

-----------------------------

**#A list of fruits**

* **Apple**
* **Orange**
* **Mango**
* **Banana**

**------------------------------------**

A dictionary is represented in a simple **key: value** form

------------------------------------

**#an employee record**

**Venugopal:**

**name: Venugopal**

**job: DevOps**

**Ansible Playbooks**

**It is a set of instructions you provide ansible to do it’s work.**

* **Deploy n number of VMs into public and private cloud**
* **Provision storage to all VMs**
* **Setup network configuration on private VMs**
* **Setup cluster configuration**
* **Configure Web servers on public VMs**
* **Configure DB servers on private VMs**
* **Setup loadbalancing between web server VMs**
* **Setup monitoring components**
* **Install and configure backup on VMs**

**All playbook are written in YAML format, A playbook contain set of play and task to perform.**

**Playbook --- A single YAML file**

**Play--- Define a set of activities to be run on hosts**

**Task--- An action to be performed on the host**

* **Execute a Command**
* **Run a Script**
* **Install a package**
* **Shutdown/restart**

**What is Modules?**

The different actions run by task is called Modules.

How to run Ansible playbook

**#ansible-playbook <playbook name>**

**Ex: #ansible-playbook playbook.yml**

If you want to see more information on Playbook

**#ansible-playbook –help**

1. **First you create a ansible playbook**

#vi playbook-pingtest.yml

- Start of the first play

name: Test connectivity to all servers

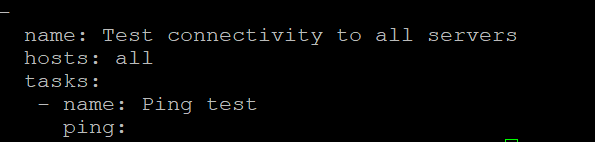
hosts: all

tasks:

- name: Ping test

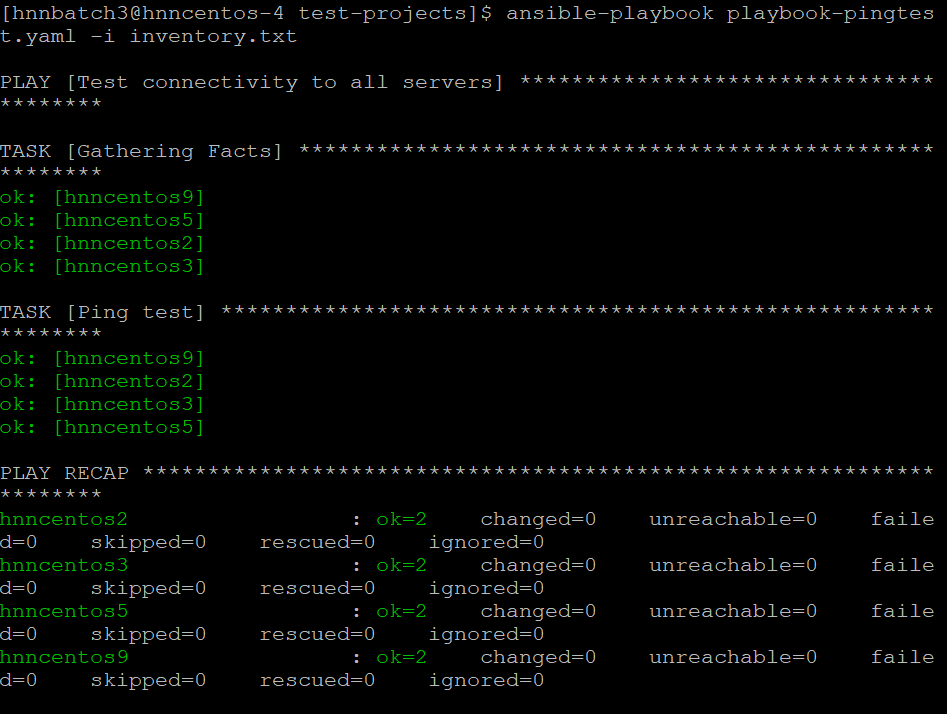
ping:

**save the file**

****

**To check the ping activity with the help of playbook**

**#ansible-playbook playbook-pingtest.yaml -i inventory.txt**

****

**To copy a file by using Playbook**

**First you create a directory and copy an inventory file to that directory.**

**Then you create a playbook**

**-**

**name: Copy file to all servers**

**hosts: all**

**tasks:**

**- name: Copy file**

**Copy:**

**src:**

**dest:**

**Modules**

**Command**

**Parameter comments**

**chdir cd into this directory before running the command**

**creates a filename, when it already exists, this step will not be run**

**Scripts**

* **Runs a local scripts on a remote after transferring it**
* **Copy script to remote systems**
* **Execute script on remote systems**

**-**

**name: Play1**

**hosts: localhost**

**tasks:**

**- name: Run a script on remote server**

**script: /some/local/script.sh -arg1 -arg2**

**Service**

* **Manages services – Start, Stop, Restart**

**-**

**name: Start Services in Order**

**hosts: localhost**

**tasks:**

**- name: start the database service**

**service: name=postgresql state:started**

**tasks:**

**- name: start the database service**

**service:**

**name: postgresql**

**State: started**

**Why “started” and not “start”?**

* **If service is not started => start it**
* **If service is already started=> do nothing**

**This is called an idempotency—an operation is idempotent if the result of performing it once is exactly same as the result of performing it repeatedly without any intervening actions.**

**Lineinfile**

* **Search for a line in a file and replace it or add it if it doesn’t exist.**

**-**

**name: Add DNS server to resolv.conf**

**hosts: localhost**

**tasks:**

**- lineinfile:**

**path: /etc/resolv.conf**

**line: ‘nameserver 10.1.250.10’**

**Variables**

* **Stores information that varies with each host**

**name: Add DNS server to resolv.conf**

**hosts: localhost**

**vars:**

**dns\_server: 10.1.250.10**

**tasks:**

**- lineinfile:**

**path: /etc/resolv.conf**

**line: nameserver ‘{{dns\_server}}’**

**==================================================================================**

**Ansible server(host server)**

**We need to install ansible in host server**

**# sudo yum install ansible -y**

**To add new group in host file**

**#vi /etc/ansible/hosts**

**To untag the Inventory and sudo user in ansible.cfg file**

**#vi /etc/ansible/ansible.cfg**

**To create the user in host server**

**#useradd <username>**

**#passwd<password>**

**To add this user into sudo file(to give full permissions)**

**#visudo ------add the user under root as NOPASSWD=ALL**

**#vi /etc/ssh/sshd\_config--- To give password authentication --- Yes**

**Slave Server**

**Useradd**

**Passwd**

**Visudo**

**Vi /etc/ssh/sshd\_config**

**And then do passwordless authentication by using SSH key**

**To list out file in servers from host system**

**#ansible dev(groupname) -a “ls”**

**To create a file in servers from host system**

**Syntax: ansible dev[indexvalue] – a “create a file”**

**#ansible dev[0] -a “touch file1” ----- here dev is group name as we created already**

**#ansible dev[0] -a “mkdir venu”**

**To install packages**

**#ansible dev[0] -a “sudo yum install git -y”**

**Ansible Modules**

* **Ansible ships with number of modules(called library modules) that can be executed directly to remote hosts or playbooks.**
* **Your library of modules can reside on any machine, and there are no servers, daemons or database required**
* **The default location for the inventory file is /etc/ansible/hosts**

**To install package by using module**

**#ansible dev[0] -b -m yum -a “pkg=git state=present” ---------- here b acts as a root and m acts as a module**

**#ansible dev[0] -b -m yum -a “pkg=git state=latest”-------------To install latest version**

**#ansible dev[0] -b -m yum -a “pkg=git state=absent” -----------To remove package**

**#ansible dev[0] -b -m user -a “name=developer” -K -------------To add new user**

**#ansible dev[0] -b -m copy -a “src=file1 dest=/tmp”-------------To copy a file**

**Ansible Playbook**

* **Playbooks in ansible are written in YAML language.**
* **It is a human readable & serialization language commonly used for configuration files.**
* **You can write code consisting of vars, tasks, handlers, files, templates and roles.**
* **Each playbook is composed of one or more modules in a list**
* **Module is a collection of configuration files.**
* **Playbooks are mainly divided into sections like**

**Target section: defines host against which playbook tasks has to be executed.**

**Variable section: Defines variables.**

**Task section: List of all modules that we need to run in an order**

**--- #TASK**

**- hosts: dev (group name)**

**user: ansible (username)**

**become: yes**

**connection: ssh**

**tasks:**

**- name: install git dev servers**

**action: yum name=git state=installed**

**To run playbook**

**ansible-playbook <file name>**

**YAML**

**For ansible, every YAML file starts with a list**

* **Each item in the list is a list one key-value pairs commonly called dictionary.**
* **All YAML files have to begin with “---”and ends with “…”**

**For example:**

**--- # A list of fruits**

**Fruits:**

**-mango**

**-apple**

**-papaya**

**-guava**

* **A dictionary is required in a simple key:value**

**--- # customer details**

**Customer:**

**Name: Venu**

**Gender: Male**

**Go to ansible server and login and create a playbook**

**--- # Target playbook**

**hosts: <group name>**

**user: <username>**

**become: yes**

**connection: ssh**

**gather\_facts: yes**

**Ansible Roles**

**Ansible role is a concept that deals with ideas rather than events, its basically another level of abstraction used to organize playbooks.**

**They provide a skeleton for an independent and reusable collection of variables , tasks, templates, files, and modules which can be automatically loaded into the playbook.**

**Create Ansible role**

**To create a role, a tool called Ansible Galaxy comes bundled with ansible.**

**Syntax:**

**ansible-galaxy init <rolename>**

**Example: ansible-galaxy init samplerole**

**Change the directory to /etc/ansible/roles and create a file by using above syntax**

**Then, /etc/ansible/roles/samplerole ----- list the contents**

Default—Contains the default variables that are going to be used by this role.

Files—Contains files that can be deployed by this role. It contains files that need to be sent to the hosts while configuring the role.

Handlers--- Contains handlers which may be used by this role or even anywhere outside this role.

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

Tasks--- Contains the main list of tasks that are to be executed by the role. It contains the main.yml file for that particular role.

Templates—Contains files which can be modified and added to the host being provisioned jinja2(template language) is used to achieve the modifications.

Tests--- This directory is used to integrate testing with ansible playbooks

Vars--- This directory consists of other variables that are going to be used by the role. These variables can be defined in your playbook, but it’s a good habit to define them in this section.

**Let’s write a playbook by using role**

**We will install prerequisites, MongoDB and node.js**

**Once role is created, the next step is to configure the roles.**

**#/etc/ansible/roles/mongodb/tasks ---- here you write a yaml file**

**Ansible Ad-Hoc Commands**

We will use Ansible to make sure all the hosts are reachable

#ansible all -m ping

Let us print the hostname of all the hosts

**#ansible all -a hostname**

How long the hosts are up?

#ansible all -a uptime

Does my dev servers have any disk space free?

#ansible dev -a free

Let us install Docker on app servers

#ansible dev -b -a "yum install -y docker-engine" -kK

Do you want a command to run on one machine at a time ?

#ansible all -f 1 -a "free"

To create a group

#ansible dev -b -m group -a "name=admin state=present" -K

To create a user

# ansible dev -b -m user -a "name=devops group=admin createhome=yes" -K

We will copy file from control node to app servers.

#ansible dev -m copy -a "src=/vagrant/test.txt dest=/tmp/test.txt"