# Ansible

## Ansible Commands

inventory files - /etc/ansible/hosts  
configuration file - /etc/ansible/ansible.cfg (manually create this file)

ansile all -i slaves.txt -m ping

ansible all -i slaves.txt -a “uname -a” (-a action)

ansible all -i slaves.txt -m ping (Grouping [ ])

ansible all -i slaves.txt -m yum -a “name=httpd state=present” -b  
ansible all -i slaves.txt -m service -a “name=httpd state=started” -b

ansible all -i slaves.txt -m copy -a “src=./slaves.txt dest=/tmp/slaves.txt” -b

ansible-playbook -i slaves.txt first.yaml –syntax-check  
ansible-playbook -i slaves.txt first.yaml

ansible-vault encrypt vault-pass.yaml

ansible-vault view vault-pass.yaml

ansible-vault view vault-pass.yaml --ask-vault-pass

ansible-galaxy init apache

## What is Configuration Management ?

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

## What are the other tools in market other than Ansible ?

Ansible, Bcfg2, CFEngine, Chef, Otter, Puppet, Quattor, SaltStack, Terraform, Pulumi and Vagrant.

## What is Ansible ?

* Ansible is a configuration management tool
* Created by Redhat
* Work on Master Slave Control Node Managed Node
* Slave ip in Inventory file
* ADHOC commands or playbooks to install, deploy or make configurations on slave or Managed Nodes

## How Ansible works ?

* Ansible works by connecting to your nodes via SSH and pushing out small programs, called modules to them.
* Modules are used to accomplish automation tasks in Ansible.
* These programs are written to be resource models of the desired state of the system.
* Ansible then executes these modules and removes them when finished.

## the different ways other than SSH by which Ansible can connect to remote hosts ?

default ssh, chroot, lxc, and jail containers

## Ansible is idempotency ?

Ansible tasks will only change the system if there is something to do.

repeteadly running the task will not change the result

example yum module wont install again and again

## How Ansible is different from Chef & Puppet ?

Ansible platform is written on python but It supports YAML command Scripts

Puppet platform is built with Ruby and it only supports Domain Specific Language (DSL) and Embedded Ruby (ERB).

Chef supports Ruby DSL with crucial prototype programming.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Metrics** | **Chef** | **Puppet** | **Ansible** | **Saltstack** |
| Availability | ✔ | ✔ | ✔ | ✔ |
| Ease of Setup | Not very easy | Not very easy | Easy | Not very easy |
| Management | Not very easy | Not very easy | Easy | Easy |
| Scalability | Highly Scalable | Highly Scalable | Highly Scalable | Highly Scalable |
| Configuration language | DSL(Ruby) | DSL(PuppetDSL) | YAML(Python) | YAML(Python) |
| Interoperability | High | High | High | High |
| Pricing (upto 100 nodes) | $13700 | $11200-$19900 | $10,000 | $15,000(approx.) |

## Where is the Ansible Configuration file located ?

/etc/ansible/ansible.cfg -> config file

/ect/ansible/hosts -> inventory file

## Where ansible logs are stored ?

By default they are disable by giving the path in ansible.cfg file they will start working

/var/lib/ansible.log --> enable in ansible.cfg file

## what is inventory file & config file ?

inventory file contains the maganed node/ slave machine Ip address

The inventory file can list individual hosts or user-defined groups of hosts.

config file contains ansible related configurations

## What are the types of Inventories ?

Static & Dynamic

## What is Dynamic Inventory & when we use it & for what ?

Python code to fetch all IP’s of running instances and update it to the inventory file

## Difference between hosts & groups ?

Each IP address is a host

we can combime multi ip into a single group

<https://docs.ansible.com/ansible/latest/user_guide/intro_inventory.html>

## Group vars vs Host vars ?

group\_vars/

group1 # here we assign variables to particular groups

group2 # ""

host\_vars/

hostname1 # if systems need specific variables, put them here

hostname2 # ""

## How to give multiple groups in hosts

hosts: web:test:group

hosts: <ip>,<ip>,<ip>

## Playbook order of execution:

1. Variable loading
2. Fact gathering
3. The pre\_tasks execution
4. Handlers notified from the pre\_tasks execution
5. Roles execution
6. Tasks execution
7. Handlers notified from roles or tasks execution
8. The post\_tasks execution
9. Handlers notified from post\_tasks execution

## What is AD Hoc commands ?

Single task command

to perform quick functions.

using the modules we can run

## What is play & playbook ?

A Playbook is a list of plays. It can contain a single play, or many.

An Ansible playbook is an organized unit of scripts that defines work for a server configuration

A Play is a list of tasks and roles that should be run. A Play can also define vars that should be used for that play.

## What are the components of Ansible ?

* Inventory file(s)
* Group vars
* Host vars
* inclue
* import
* become
* Roles
* Tasks
* remote\_user
* handler
* notify
* listen
* block
* rescue
* always

## Pre-task & Post-task ?

Tasks that can be runned before a role execution is started

- hosts: www

remote\_user: vagrant

sudo: yes

pre\_tasks:

- shell: echo 'I":" Beginning to configure web server..'

roles:

- nginx

tasks:

post\_tasks:

- shell: echo 'I":" Done configuring nginx web server...'

In the preceding example, we only printed some messages using the echo command. However, we can create tasks using any of the modules available with Ansible, which can run before, or after, applying roles.

## with item vs loop

The with\_ keywords rely on Lookup Plugins - even items is a lookup.

The loop keyword is equivalent to with\_list, and is the best choice for simple loops.

The loop keyword will not accept a string as input, see Ensuring list input for loop: query vs. lookup.

Generally speaking, any use of with\_\* covered in Migrating from with\_X to loop can be updated to use loop.

Be careful when changing with\_items to loop, as with\_items performed implicit single-level flattening. You may need to use flatten(1) with loop to match the exact outcome.

## What is handlers and Notify and listen?

when task is returns status changed it will notify handler and will execute

task not changed it returns status ok and notify will not work

we can use listen to listen handlers from different role and execute another handler

<https://medium.com/@george.shuklin/listen-feature-for-handlers-in-ansible-29183524c7e1>

## when condition ?

a condition when it is satisfied the play will run, example ansible\_os\_family == ‘Debain’

## What are ansible modules ?

pre defined libraries written in python

use modules to write tasks in playbook

750 + modules

* delegate to -> run only on a particular IP
* local action -> run on control node
* line in file -> edit single line in file
* block in file -> edit chunks of line
* template
* copy
* get\_url
* apt
* yum
* yum\_repo
* shell
* service
* unarchive
* command

## template module ?

The template module also copies a file to a remote server, but it allows you to use Jinja2 to render a template to a file dynamically. This enables you to use variables, such as Ansible facts, to customize a particular file for a specific server. ... The file uses the . j2 suffix so that you know it is a Jinja2 template.

## What is Jinja 2 template ?

Jinja is a web template engine for the Python programming language.

save file in .j2 format and using template module we can edit particular data

## How to include custom modules in Ansible ?

~/.ansible/plugins/modules/

## Difference between COPY & FILE & template modules ?

**copy** takes a file from host,"as-is",and copies it to the remote destination. backup=yes

**template** takes a file (template) from host,changes variables based on Jinja2 filtering,and copies it to the remote destination.

**file** used to touch, create dir, remove file, set permissions state=directory, state=absent

## Difference between SHELL & COMMAND modules ?

**shell** – Execute shell commands on targets

It is almost exactly like the command module but runs the command through a shell (/bin/sh) on the remote node.

**command** – Execute commands on targets

The command(s) will not be processed through the shell, so variables like $HOME and operations like "<", ">", "|", ";" and "&" will not work.

## What is Setup module ? what it does ?

task is setup and

This module is automatically called by playbooks to gather useful variables about remote hosts that can be used in playbooks. It can also be executed directly by /usr/bin/ansible to check what variables are available to a host. Ansible provides many facts about the system, automatically.

## What is variable in Ansible ?

create vars: and specify ports:2000

can include in playbook task

also while running playbook can pass var as -e

## What are different types of variables scopes?

Ansible has 3 main scopes:

* Global: this is set by config, environment variables and the command line
* Play: each play and contained structures, vars entries, include\_vars, role defaults and vars.
* Host: variables directly associated to a host, like inventory, facts or registered task outputs

## How variable precedence takes place ?

Configuration settings (lower precedence)

Command-line options

Playbook keywords

Variables (Highest Precedence)

## what is include\_var vs vars\_files ?

**vars\_files:**

variables from those files are included in the playbook

Since it is used in the start of the play

*vars\_files are read when the play starts. include\_vars are read when the play reaches the task.*

## What is gather\_facts ?

This module is automatically called by playbooks to gather useful variables about remote hosts that can be used in playbooks.

## What is ansible Vault ?

Ansible Vault can encrypt the playbook if it contains username and password

AES256 encryption

ansible-vault create vault.yml

ansible-vault encrypt encrypt\_me.txt

ansible-vault view vault.yml

ansible-vault edit vault.yml

ansible-vault decrypt vault.yml

ansible-vault rekey encrypt\_me.txt

ansible-playbook --vault-id dev@prompt site.yml

ansible-playbook --vault-id dev@dev-password --vault-id prod@prompt site.yml

ansible-vault encrypt\_string <password\_source> '<string\_to\_encrypt>' --name '<string\_name\_of\_variable>'

## If a string is encrypted in a file with a password then how to pass the password using parameter while decrypting ?

ansible-playbook --ask-vault-pass site.yml

## If a file is encrypted using password & password is stored in a file how to pass the file to decrypt the file ?

ansible-playbook --vault-password-file /path/to/my/vault-password-file site.yml

## Can we recover ansible-vault password if lost ?

Unless you brute force the password there is no way to recover it. I suggest you use a password manager to store your password and share the password database within your organization.

## tags vs skip tags

to run a specific playbook we can use tags

skip tags will skip that particular task and executes everything else

## What is Roles ?

In Ansible, the role is the primary mechanism for breaking a playbook into multiple files. ... This simplifies writing complex playbooks, and it makes them easier to reuse.

## Differentiate Playbook vs role ?

Role is primarly used for Code Reusability

Roles contain folder like vars, files, handlers, tasks, meta

ansible-galaxy command is used to create a role

in playbook we will attach role

## How to create roles ?

ansible-galaxy init <role\_name>

## How to install a Role ?

ansible-galaxy install <role\_name>

## How to install multiple roles ?

ansible-galaxy install -r requirements.yml

requirements.yaml file

## what is import role vs include role/ Difference between include & import ?

import tasks will be parsed at the beginning when you run your playbook

include tasks will be parsed at the moment Ansible hits them

## Template folder vs file folder in roles ?

**Template folder**

This folder contains the template files used by the role to create the actual configuration files.

These are then deployed by the role to the remote hosts.

They are Jinja2 template engine scripts that enable loops and other features.

**File Folder**

This folder holds all extra files that are required to achieve the role task.

These files usually get dispatched to remote hosts as part of certain tasks.

They are usually static, and they do not contain any variables to change, be copied, extracted, or compressed to the remote host.

## Role dir Structure ?

**tasks**/main.yml - the main list of tasks that the role executes.

**handlers**/main.yml - handlers, which may be used within or outside this role.

**library**/my\_module.py - modules, which may be used within this role (see Embedding modules and plugins in roles for more information).

**defaults**/main.yml - default variables for the role (see Using Variables for more information). These variables have the lowest priority of any variables available, and can be easily overridden by any other variable, including inventory variables.

**vars**/main.yml - other variables for the role (see Using Variables for more information).

**files**/ - files that the role deploys.

**templates**/main.yml - templates that the role deploys.

**meta**/main.yml - metadata for the role, including role dependencies.

## Defaults folder in roles

defaults/main.yml is a configuration file that you can use to define default values for variables used in your role.

It allows for a centralized management of the default values of the variable of the role.

Default values are always vulnerable because they change a lot depending on the needs and policies of the user.

Having this solution allows one file to change all the values

## Difference between default & vars directory in Roles ?

If roles/x/vars/main.yml exists, Ansible adds the variables in that file to the play.

If roles/x/defaults/main.yml exists, Ansible adds the variables in that file to the play.

The defaults directory is for defining the variable defaults. The variables in default have the lowest priority thus becoming easy to override. If definition of a variable is nowhere else, the variable in defaults/main.yml will be used.

## Can we disable automatic facts gathering in Ansible ?

gather\_facts = no, use explictly setup

## How error handling can be done in Ansible ?

using block rescue always

## What is changed\_when & failed\_when in Ansible ?

**changed\_when** = mark it as changed when a condition is met

**failed\_when** = mark it as failed when a condition is met, Ram and CPU checking

## How to ignore failed commands in Ansible ?

ignore\_errors: yes

## How to control the command failure in Ansible ?

changed\_when & failed\_when

## How to debug your playbook ?

using register and debug

## What is block/ rescue/ always in Ansible ?

we can group multiple tasks togethor in blocks

**block** -> try for errors

**rescue** -> work when block has error

**always** -> always gets executed

## What is register in Ansible?

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

cmd – The command that ran to generate the output.

**stdout** – The output of the command.

stderr – The error output of the command.

start – The date and time when the command began executing.

end – The date and time when the command finished executing.

delta – The time taken to run the command. This is the difference between the end and the start properties.

stdout\_lines – An array containing each output line of the command. Same as stdout, but stdout separates the lines using a newline (\n) characters instead of arrays.

stderr\_lines – An array containing each error output line of the command. Same as stderr, but stderr separates the lines using newlines (\n) characters instead of arrays.

## What is debug in Ansible

This module prints statements during execution and can be useful for debugging variables or expressions without necessarily halting the playbook. msg, var

## what is –check & --diff in ansible ? / What is Dry Run in Ansible & how to do that ?

**--check** Check mode is just a simulation test run

**--diff** shows the before and after changes

**--check --diff** shows what would have been the changes

## What is Privilege Escalation in Ansible ?

Ansible uses existing privilege escalation systems to execute tasks with root privileges or with another user’s permissions.

Because this feature allows you to ‘become’ another user, different from the user that logged into the machine (remote user), we call it become.

The become keyword leverages existing privilege escalation tools like sudo, su, pfexec, doas, pbrun, dzdo, ksu, runas, machinectl and others.

**become** set to yes to activate privilege escalation.

**become\_user** set to user with desired privileges — the user you become, NOT the user you login as. Does NOT imply become: yes, to allow it to be set at host level. Default value is root.

## What is lookup in Ansible playbook ?

You can use lookup plugins to access data from outside sources (files, databases, key/value stores, APIs, and other services) within your playbooks.

## How you can run your all tasks at once ?

async : 45 maximum time i give to the task to complete

poll : 0 doesnot wait for the task to get complete simultaneously runs all tasks

setting poll to 0 will create erros when using yum module as it has dependency

## loop -> syncronize module

A wrapper around rsync to make common tasks in your playbooks quick and easy.

## Installing Tomcat ?

- hosts: all

  remote\_user : ec2-user

  become: yes

  tasks:

  - name: install the latest version of apache using ansible

    yum:

      name: httpd

      state: present

  - name: starting apache

    service:

      name: httpd

      state: started

  - name: copying files

    copy:

      src: /index.html

      dest: /var/www/html/index.html

      mode: '0777'

### Creating EC2

- hosts: localhost

  remote\_user: ec2-user

  become: yes

  tasks:

    - name: Creating EC2

      ec2:

        key\_name: LaptopKey

        instance\_type: t2.micro

        image: ami-0bcf5425cdc1d8a85

        region: "ap-south-1"

        count: 1

        vpc\_subnet\_id: subnet-ea6a06a6

        assign\_public\_ip: yes

## Installing Jenkins

- hosts: jenkins

  remote\_user : ec2-user

  become: yes

  vars:

    ports: 9000

  tasks:

    - name: Installing jenkins

      yum\_repository:

        name: jenkins

        description: jenkins YUM repo

        baseurl: http://pkg.jenkins.io/redhat-stable

        gpgkey: http://pkg.jenkins.io/redhat-stable/jenkins.io.key

    - name: Installing java and Jenkins

      yum:

        name: "{{ item }}"

        state: present

      loop:

        - java

        - jenkins

    - name: going to change port no

      lineinfile:

        path: /etc/sysconfig/jenkins

        regexp: '^JENKINS\_PORT='

        line: JENKINS\_PORT={{ports}}

      notify: restart jenkins

    - name:

      service:

        name: jenkins

        state: started

  handlers:

  - name: restart jenkins

    service:

      name: jenkins

      state: restarted

## Using Ansible Loop

- hosts: all

  remote\_user : ec2-user

  become: yes

  tasks:

    - name: install the latest version of apache using ansible

      yum:

        name: "{{ item }}"

        state: present

      loop:

        - php

        - mysql

        - unzip

        - http\_present

## Installing Tomcat

- hosts: tomcat

  remote\_user : ec2-user

  become: yes

  vars:

    tomcat\_port: 9090

  tasks:

  - name: yum update

    yum:

      name: "\*"

      state: latest

  - name: installing java

    yum:

      name: java-1.8.0-openjdk

      state: present

    when: ansible\_os\_family == "centos"

  - name: installing java

    apt:

      name: java-1.8.0-openjdk

      state: present

    when: ansible\_os\_family == "debian"

  - name: Download tomcat

    get\_url:

      url: https://mirrors.estointernet.in/apache/tomcat/tomcat-9/v9.0.46/bin/apache-tomcat-9.0.46.tar.gz

      dest: /opt

      mode: '777'

  - name: Extract apache.tar

    unarchive:

      src: /opt/apache-tomcat-9.0.46.tar.gz

      dest: /opt

      remote\_src: yes

      mode: '777'

  - name: Template a file to /etc/file.conf

    template:

      src: /home/ec2-user/server.xml.j2

      dest: /opt/apache-tomcat-9.0.46/conf/server.xml

  - name: stop tomcat

    shell: nohup /opt/apache-tomcat-9.0.46/bin/shutdown.sh &

  - name: start tomcat

    shell: nohup /opt/apache-tomcat-9.0.46/bin/startup.sh &

  - name: copying files

    copy:

      src: /home/ec2-user/sample.war

      dest: /opt/apache-tomcat-9.0.46/webapps/sample.war

      mode: '0777'