

# Ansible – Configuration Management

<https://docs.ansible.com/>

## Topics

- Introduction to Ansible
- Setting up Ansible
- Introduction to YAML
- Inventory Files
- Playbooks
- Variables
- Conditionals
- Loops
- Roles

## Control Node

Redhat or CentOS – `$ sudo yum install ansible`

Fedora – `$ sudo dnf install ansible`

Ubuntu – `$ sudo apt-get install ansible`

PIP – `$ sudo pip install ansible`

Install pip if not present

`$ sudo yum install epel-release`

`$ sudo yum install python-pip`

Install Ansible using pip Upgrade Ansible using pip

`$ sudo pip install ansible` `$ sudo pip install --upgrade ansible`

Install Specific Version of Ansible using pip

`$ sudo pip install ansible==2.4`

# Ansible Inventory

Inventory contains the list of hosts to be managed/configured

## Default Inventory → /etc/ansible/hosts

server1.example.com

server2.example.com

[db]

server3.example.com

server4.example.com

[web]

server5.example.com

server6.example.com

## Connection

Linux – SSH

Windows – Powershell Remoting

e.g:

web1 ansible\_host=server1 ansible\_connection=ssh ansible\_user=root ansible\_ssh\_pass =xyz

web2 ansible\_host=server2 ansible\_connection=winrm ansible\_user=administrator  
ansible\_password=xyz

## Ansible Playbook

- Defines plays containing tasks to be performed on managed hosts.
- File format is YAML
  - Play – Defines a set of activities (tasks) 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

## Playbook Format

-

name: Play 1

hosts: localhost

tasks:

- name: Execute command 'date'

command: date

## Run

\$ ansible-playbook <playbook file name>

## Ansible Configuration Files

**/etc/ansible/ansible.cfg**

[defaults]

[inventory]

[privilege\_escalation]

[paramiko\_connection]

[ssh\_connection]

[persistent\_connection]

[colors]

**\$ ANSIBLE\_CONFIG=<path to custom cfg file>**

## Configuration file Precedence

0 /etc/ansible/ansible.cfg

1 ~/.ansible.cfg

2 ./ansible.cfg

3 ANSIBLE\_CONFIG

Single configuration can be set anywhere in the hierarchy of config files:

```
$ export ANSIBLE_GATHERING=explicit
```

## View Configuration

```
$ ansible-config list
```

```
$ ansible-config view
```

```
$ ansible-config dump
```

```
$ export ANSIBLE_GATHERING=explicit
```

```
$ ansible-config dump | grep GATHERING
```

```
DEFAULT_GATHERING(env: ANSIBLE_GATHERING) = explicit
```

## Facts

[https://docs.ansible.com/ansible/latest/user\\_guide/playbooks\\_vars\\_facts.html](https://docs.ansible.com/ansible/latest/user_guide/playbooks_vars_facts.html)

- name: Gather facts

```
gather_facts: yes no
```

```
gather_facts: true false
```

```
gather_facts: TRUE FALSE
```

```
gather_facts: True False
```

## Creating and Distributing SSH key

```
$ ssh-keygen
```

```
id_rsa id_rsa.pub
```

```
$ ssh-copy-id -i id_rsa <user>@<server>
```

## Privilege Escalation

- Become Super user (sudo) → **become: yes**
- Become Method – sudo (pfexec, doas, ksu, runas) → **become\_method: <method-name>**
- Become another user → **become\_user: <user-name>**

## Privilege Escalation in Inventory File

Server1 ansible\_ become=yes ansible\_ become\_user=<user-name>

## Privilege Escalation in Configuration File

/etc/ansible/ansible.cfg

become = True

become\_method = doas

become\_user = <user-name>

## Privilege Escalation using command Line

\$ ansible-playbook --become --become-method=doas --become-user=<user> --ask-become-pass

# Modules

[https://docs.ansible.com/ansible/2.9/modules/list\\_of\\_all\\_modules.html](https://docs.ansible.com/ansible/2.9/modules/list_of_all_modules.html)

\$ ansible -m <module-name> <hosts>

e.g:

\$ ansible -m ping all

\$ ansible -a 'cat /etc/hosts' all

## Check Mode or Dry Run

\$ ansible-playbook playbook.yml --check

## Start at

```
$ ansible-playbook playbook.yml --start-at-task <task-name>
```

## Tags

```
$ ansible-playbook playbook.yml --tags "install"
```

```
$ ansible-playbook playbook.yml --skip-tags "install"
```

# Variables

[https://docs.ansible.com/ansible/latest/user\\_guide/playbooks\\_variables.html](https://docs.ansible.com/ansible/latest/user_guide/playbooks_variables.html)

- Define variables in inventory/playbook file
- Use the variable in playbook in Jinja format I.e withing `{{ }}`

## inventory

```
web1 ansible_host=172.20.1.100
```

```
web2 ansible_host=172.20.1.101 dns_server=10.5.5.4
```

```
[web_servers]
```

```
web1
```

```
Web2
```

```
[web_servers:vars]
```

```
dns_server=10.5.5.3
```

## Playbook

```
vars:
```

```
    dns_server: 10.5.5.5
```

```
tasks:
```

```
  - nsupdate:
```

```
    server: '{{ dns_server }}
```

```
$ ansible-playbook playbook.yml --extra-vars "dns_server = 10.5.5.6"
```

## Variable Precedence

1. Role Defaults
2. Group vars
3. Host vars
4. Host Facts
5. Play vars
6. Role vars
7. Include vars
8. Set Facts
9. Extra vars

## Variable Scope

- Host
- Group
- Play
- Global/Playbook

## Register Variables

[https://docs.ansible.com/ansible/latest/user\\_guide/playbooks\\_variables.html#registering-variables](https://docs.ansible.com/ansible/latest/user_guide/playbooks_variables.html#registering-variables)

- Create variables from the output of an Ansible task with the task keyword **register**.
- Use registered variables in any later tasks in the play.

## Magic Variables

[https://docs.ansible.com/ansible/latest/user\\_guide/playbooks\\_vars\\_facts.html#information-about-ansible-magic-variables](https://docs.ansible.com/ansible/latest/user_guide/playbooks_vars_facts.html#information-about-ansible-magic-variables)

- hostvars
- groups

- group\_names
- inventory\_hostname

e.g:

```
msg: '{{ hostvars['<hostname>'].ansible_host }}'
```

```
msg: '{{ hostvars['<hostname>'].ansible_facts.architecture }}'
```

```
msg: '{{ hostvars['<hostname>'].ansible_facts.devices }}'
```

```
msg: '{{ hostvars['<hostname>'].ansible_facts.mounts }}'
```

```
msg: '{{ hostvars['<hostname>'].ansible_facts.processor }}'
```

```
msg: '{{ hostvars['<hostname>']['ansible_facts']['processor'] }}'
```

# Conditionals

- when

## Operators

- or
- and

## Loops

- Loop keyword to iterate over a list

## Blocks

- Groups tasks

## Error Handling

- Rescue block for action to be taken in case of failure
- always block to be executed at the end irrespective of task status



## Task failure

- `any_errors_fatal: true`
- `max_fail_percentage: 30`
- `ignore_errors: yes` # to be specified with task
- `failed_when: <check>` # at task level

# Ansible Filters

[https://docs.ansible.com/ansible/latest/user\\_guide/playbooks\\_filters.html](https://docs.ansible.com/ansible/latest/user_guide/playbooks_filters.html)

## Templates

- Use Jinja 2 format for creating templates
- Use templates instead of copy module to copy the files to servers with interpolated values

## Includes

- Use `include_vars` module to import variable from a file
- Create an inventory hierarchy as below:
  - **Inventory**
    - `inventory`
    - **host\_vars**
      - `<hostname alias>.yaml`
    - **group\_vars**
      - `<group name>.yaml`

```
$ ansible-inventory -i inventory/ -y
```

- **include\_tasks** to include tasks from other .yaml file

e.g:

```
tasks: - name: Install MySQL Packages
        - include_tasks: tasks/db.yml
        - include_tasks: tasks/web.yml
```

# Roles

- Code Reusablty
- Initialize a role using below command:  

```
$ ansible-galaxy init mysql
```
- Roles contains below folders
  - tasks
  - vars
  - defaults
  - handlers
  - templates
- Use roles in a playbook as below

```
- name: Install and Configure MySQL
  hosts: db-server 1.....db-server100
  roles:
    - mysql
```
- Look for ansible roles at <https://galaxy.ansible.com/>
- Use ansible-galaxy command to serach a role using CLI

```
$ ansible-galaxy search <keyword>
```
- Install a role using below command:

```
$ ansible-galaxy install <role-name>
$ ansible-galaxy install geerlingguy.mysql -p ./roles
```
- List roles

```
$ ansible-galaxy list
```

# Strategy

- Linear (Default)
- Free
  - `strategy: free`
- Batch
  - `serial: 3`
  - forks value in `.cfg` file