



Think^{nyx}[®]

Ansible Cheat Sheet

Ansible Setup

To Install Ansible on Ubuntu Linux

```
# To update the package information from configured sources  
$ sudo apt update  
# To install software properties pre-requisite package  
$ sudo apt install software-properties-common  
# To configure ansible repository (PPA)  
$ sudo add-apt-repository --yes --update ppa:ansible/ansible  
# To install Ansible package  
$ sudo apt-get install ansible  
# To check Ansible version  
$ ansible --version
```

To Install Ansible on RHEL

```
# To install EPEL packages for RHEL 8  
$ sudo yum install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm  
# To enable Ansible engine repository for RHEL 8  
$ sudo subscription-manager repos --enable ansible-2.9-for-rhel-8-x86_64-rpms  
# To install Ansible package  
$ sudo yum install ansible  
# To check Ansible version  
$ ansible --version
```

Ansible Inventory

The Ansible inventory file defines the hosts and groups of hosts on which ad-hoc commands, modules, and tasks in a playbook can run

```
# Default Inventory location  
$ /etc/ansible/hosts  
# Provide the hostgroup name  
[hostgroup]  
# Provide the host IP address  
ansible_host=<IP>  
# Provide the username of the host  
ansible_user=<user_name>  
# Provide the port number of the host to connect to  
ansible_port=<port_number>  
# Provide the password of the host to connect to  
ansible_ssh_pass=<password>  
# Provide the connection mechanism of the host to connect to  
ansible_connection=<ssh/winrm>  
ssh for unix  
winrm for windows
```

Example of an Inventory file

```
vi /etc/ansible/hosts  
[webserver]  
ansible_host=10.0.1.14 ansible_user=test_user  
ansible_ssh_pass=admin ansible_connection=ssh
```



Inventory Setup and Ping test

```
# To set up hosts by editing the hosts file in the default Ansible directory  
$ sudo vi /etc/ansible/hosts  
# To test the connectivity on servers individually  
$ ansible <server_name> -m ping  
# To test the connectivity on particular server group  
$ ansible <server_group_name> -m ping
```

Ansible Hosts Pattern

Patterns in Ansible are used to manage the hosts

```
# To use a specific module on all hosts in inventory  
$ ansible all -m <module_name> -a <arguments>  
# To use a specific module on all hosts in inventory not appearing within a group  
$ ansible ungrouped -m <module_name> -a <arguments>  
# To use a specific module on all hosts with an IP starting 10.0.0.*  
$ ansible 10.0.0.* -m <module_name> -a <arguments>  
# To use a specific module on the group defined as webservers  
$ ansible webservers -m <module_name> -a <arguments>
```

Ansible Ad-hoc Commands

Ad-hoc commands in Ansible allows you to execute simple tasks at the command line against one or set of hosts without writing a playbook

```
# To check hosts system's info (facts)  
$ ansible <host_inventory> -m setup | less  
# To copy files  
$ ansible <host_inventory> -m copy -a "src=/etc/thinknyxfile dest=/tmp"  
# To create new user  
$ ansible <host_inventory> -m user -a "name=thinknyx uid=9999 state=present"  
# To delete user  
$ ansible <host_inventory> -m user -a "name=thinknyx uid=9999 state=absent"  
# To install apache webserver package  
$ ansible <host_inventory> -m apt -a "name=apache2 state=present"  
# To updated existing apache webserver package to the latest version  
$ ansible <host_inventory> -m apt -a "name=apache2 state=latest"  
# To uninstall apache webserver package  
$ ansible <host_inventory> -m apt -a "name=apache2 state=absent"  
# To start apache webserver services  
$ ansible <host_inventory> -m service -a "name=apache2 state=started"  
# To stop apache webserver services  
$ ansible <host_inventory> -m service -a "name=apache2 state=stopped"  
# To restart apache webserver service  
$ ansible <host_inventory> -m service -a "name=apache2 state=restarted"
```

Filter in Ad-hoc

Example of using filter in ad-hoc

```
$ ansible <hostgroup> -m setup -a "filter=ansible_distribution*"
```

Ansible Playbooks

Ansible playbook is an organized unit of automation steps or a blueprint of automation tasks without human involvement

```
# To list affected hosts  
$ ansible-playbook <YAML file> --list-hosts  
  
# To list affected tasks  
$ ansible-playbook <YAML file> --list-tasks  
  
# To check the syntax  
$ ansible-playbook <YAML file> --syntax-check  
  
# To perform Dry run or Smoke test  
$ ansible-playbook <YAML file> --check  
  
# To execute the ansible playbook  
$ ansible-playbook <YAML file>
```

Example of --limit flag to execute the playbook on limited server group

```
$ ansible-playbook <YAML file> --limit <group_name or server_name>
```

Using Collections in a playbook

```
---  
- hosts: web  
  tasks:  
    - import_role:  
      name: namespace.collection.role
```

Ansible Collections

Collections are a distribution format for Ansible content that can include playbooks, roles, modules, and plugins. This feature has been newly added in Ansible

```
# To install ansible collection  
$ ansible-galaxy collection install <namespace.collection>  
  
# To initialize new collection with base structure of a collection  
$ ansible-galaxy collection init <namespace.collection>  
  
# To build a collection from inside the root directory of the collection.  
$ ansible-galaxy collection build
```

Directory Structure

```
collection/  
  docs/  
  galaxy.yml  
  meta/  
    runtime.yml  
  plugins/  
    modules/  
      module1.py  
    inventory/  
    .../  
  README.md  
  roles/  
    role1/  
    role2/  
    .../  
  playbooks/  
    files/  
    vars/  
    templates/  
    tasks/  
    tests/
```

Ansible Playbook

```
---  
- name: Package Installation  
hosts: web  
tasks:  
  - name: Install git  
    apt:  
      name: git  
      state: present  
  
  - name: Install apache2 webserver  
    apt:  
      name: apache2  
      state: present  
  
  - name: create a user  
    user:  
      name: webuser  
      state: present
```

Ansible Playbook with Tags

```
---  
- hosts: web  
tasks:  
  - name: install git  
    apt:  
      name: git  
      state: present  
    tags: git  
  
  - name: install apache2 webserver  
    apt:  
      name: apache2  
      state: present  
    tags: apache  
  
  - name: Create a user  
    user:  
      name: webuser  
      state: present  
    tags: webuser
```

If you run this playbook with --tags git, Ansible will run the tasks tagged git and will skip the other two tasks that does not have a tag named "git"

```
$ ansible-playbook <YAML file> --tags git
```

Ansible Tags

Tags are useful to run only specific part of a playbook instead of entire playbook

```
---  
- name: Install ntp  
  apt:  
    name: ntp  
    state: present  
  tags: ntp
```

Ansible Vault

Ansible vault allows you to keep your sensitive data encrypted

```
# To create a new encrypted file  
$ ansible-vault create <><file_name>>  
# To edit the encrypted file  
$ ansible-vault edit <><file_name>>  
# To view the encrypted file  
$ ansible-vault view <><file_name>>  
# To encrypt a file  
$ ansible-vault encrypt <><file_name>>  
# To decrypt a file  
$ ansible-vault decrypt <><file_name>>  
# To rekey your vault password  
$ ansible-vault rekey <><file_name>>
```



Role Directory Structure

```
roles/
  tasks/
    -> main.yml
  handlers/
    -> main.yml
  files/
    -> web/index.html
  templates/
  vars/
    -> main.yml
  defaults/
    -> main.yml
  meta/
    -> main.yml
  tests/
    -> inventory
    -> test.yml
  README.md
```

meta/main.yml

```
galaxy_info:
  author: Roopam Gaikar
  description: Apache webserver
  Role
  company: Thinknyx Technologies
```

Ansible Role

Ansible Roles provide a well defined and reusable directory structure for writing playbooks

```
# To initialize a new role
$ ansible-galaxy init apache
```

tasks/install.yml

```
---
- name: Install apache2 package
  apt:
    name: apache2
    state: present
```

tasks/copy.yml

```
---
- name: Copy index.html file
  copy:
    src: web/index.html
    dest: /var/www/html
  notify:
    - restart apache
```

tasks/service.yml

```
---
- name: Start and enable apache2 service
  service:
    name: apache2
    state: restarted
    enabled: yes
```

Think^{nyx}[®]

files

```
web/index.html
```

tasks/main.yml

```
---
- import_tasks: install.yml
- import_tasks: copy.yml
- import_tasks: service.yml
```

handlers/main.yml

```
---
- name: restart apache
  service:
    name: apache2
    state: restarted
```

Using Roles Inside a Playbook

```
---
- hosts: web
  tasks:
    - include_role:
        name: apache
```



Think^{nyx}[®]

YOU TRUST, WE DELIVER



www.thinknyx.com



support@thinknyx.com



+91 9810344919/9717917973

Reach out to us at: