

# Ansible Cheat Sheet

Last update: June 30, 2020

## # ansible CLI

- i** Inventory, w/o "all -i hostname,"
- m** Module name
- a** Module arguments
- e** Extra variables
- b** Become, add **-K** to prompt for password
- k** Connection password, **-u** for remote user
- v(vvv)** Verbosity
- check, -C** Predict what would change
- one-line, -o** Condense output on single line

## # ansible-playbook CLI (in addition to above)

- syntax-check** YAML syntax validation
- step** Asks before completing task
- start-at-task** Starts playbook at specific task

## Inventory

- all** Every host explicitly listed in the inventory
- ungrouped** Every host explicitly listed in the inventory not a member of any group

Nested Groups [new\_group:children]  
child\_host\_one  
child\_host\_two

Testing Inventory # ansible [host or group] --list-hosts  
Accepts wildcard = \*  
Multiple entries = host1,host2  
Logical AND = group1,&group2  
Logical NOT = group1,!group2

## Facts

- Get facts from a node (ad hoc) # ansible targetName -m setup
- In a playbook "{{ ansible\_fact }}"
- Include fact in inventory [group]  
host1 admin\_user=jane
- Local facts place .fact file (JSON or INI) into /etc/ansible/facts.d on target node
- Get local facts # ansible targetName -m setup -a "filter=ansible\_local"

## Delegated Facts

Facts gathered by delegated task are assigned by default to **delegate\_to** host, instead of host which produced the facts

The **delegate\_facts** directive set to True to assign gathered facts from the task to delegated host instead of current host

## Variables

- As a file vars\_files:  
- vars.yml
- Direct reference vars:  
project\_root: /etc/xyz
- In a playbook file:  
state: directory  
path: "{{ project\_root }}"
- Store output of a command as a variable shell: cat /etc/network | grep eth0  
register: address  
debug:  
msg: "addr= {{ address.stdout }}"

## Magic Variables

- hostvars** Contains the variables for managed hosts, and can be used to get values for another managed host's variables
- group\_names** Lists all groups for current managed host
- groups** Lists all groups and hosts in the inventory
- inventory\_hostname** Contains the hostname for the current managed host as listed in the inventory

## Delegation

- Run task on different target - name: run on remote server  
delegate\_to: remote\_server
- To local machine - name: run locally  
delegate\_to: localhost

## Simple Loops

```
yum:  
  name: "{{ item }}"  
  state: present  
  loop:  
    - http  
    - myapp
```

## Includes

```
tasks:  
  - include: db.yml
```

## Debugging

- Print always debug:  
msg: "running stuff"
- Print only at verbosity level debug:  
var: result  
verbosity: 2

## # ansible-galaxy CLI

- search** Searches Galaxy for the string specified
- info** Displays more detailed info about a role
- install** Downloads a role, then installs it locally
- list** Lists the roles that are found locally
- remove** Remove local role
- init** Creates a directory structure for a new role that will be developed
- init --offline** Used when Internet access is unavailable

# ansible-vault CLI	
<b>create edit view</b>	Create Edit View encrypted file
<b>encrypt</b>	Encrypt existing file
<b>rekey</b>	Change encrypted password
Roles	
In a playbook	<pre>roles:   - role1   - role2</pre>
Conditional Execution	
<pre>- shell: echo "only on Red Hat 7"   when:     - ansible_distribution == "RedHat"     - ansible_distribution_major_version == "7"</pre>	
<pre>- shell: echo "I've got '{{ foo }}'!"   when: foo is defined</pre>	
Handlers	
Declare/Notify in Task	<pre>- name: install apache   package:     name: apache2     state: latest   notify:     - restart apache</pre>
Execute	<pre>handlers:   - name: restart apache     service:       name: httpd       state: restarted</pre>
Common Modules with Examples	
Module: Command	
Change directory to “chdir” will only run when “creates” doesn't exist	<pre>- name: command example   command: /usr/bin/make_db.sh arg1   args:     chdir: somedir/     creates: /path/to/database   register: myoutput</pre>
Module: Get_URL	
Pull a file from a remote source	<pre>- name: get_url example   get_url:     url: http://example.com/file.conf     dest: /etc/foo.conf     mode: 0440     backup: yes no</pre>
Module: Shell	
Redirect	<pre>shell: somescript.sh &gt;&gt; somelog.txt</pre>
Env vars	<pre>shell: echo "\$HOME"</pre>
Module: Service	
Manipulate a service/daemon	<pre>- name: service example   service:     name: httpd     state: started stopped restarted     enabled: yes</pre>
Module: Reboot	
Reboot remote target	<pre>- name: Reboot target   reboot:     msg: "Rebooting now!"     reboot_timeout: 3600</pre>

Module: Copy	
Copy new ntp.conf file into place, backing up the original if it differs from the copied version	<pre>- name: copy example   copy:     src: /mine/ntp.conf     dest: /etc/ntp.conf     owner: root     group: root     mode: 0644     backup: yes</pre>
Module: File	
Create a File	<pre>- name: file example   file:     path: /etc/foo.conf     owner: foo     group: foo     mode: 0644</pre>
Dir (create if not exist)	<pre>- name: file example for dirs   file:     path: /etc/some_directory     state: directory     mode: 0755</pre>
Module: Package	
Manipulate a package	<pre>- name: package example   package:     name: ntpdate     state: present absent latest</pre>
Module: LineInFile	
Insert using regex	<pre>- name: example insert   lineinfile:     path: /etc/selinux/config     regexp: '^SELINUX='     line: 'SELINUX=enforcing'</pre>
Remove using regex	<pre>- name: example remove   lineinfile:     path: /etc/sudoers     state: absent     regexp: '^%wheel'</pre>
Module: Authorized_key	
Grab key from remote URL	<pre>- name: Set via remote url   authorized_key:     user: root     state: present     key: https://example.com/root.key</pre>
Module: Replace	
Replace each instance via regex	<pre>- name: New hostname   replace:     path: /etc/hosts     regexp:       '(\s+)old\.host\.name(\s+\.*)?\$',     replace: '\1new.host.name\2'</pre>