Contents

[Ansible ad-hoc command cheat sheet: 1](#_Toc514324759)

[1. Finding the syntax of a module: 1](#_Toc514324760)

[2. PlayBooks: 2](#_Toc514324761)

[3. Adding conditions in task: 2](#_Toc514324762)

[4. Ansible Loops 2](#_Toc514324763)

[5. Become a different user: 3](#_Toc514324764)

[6. Include other palybooks or tasks: 3](#_Toc514324765)

[7. Check mode and diff mode ( -D --diff how the file will change before it changed) 3](#_Toc514324766)

[8. Command and shell readonly 3](#_Toc514324767)

[Templates—jinja language 3](#_Toc514324768)

[Handlers: tasks that are fired when a task makes a change 3](#_Toc514324769)

[Handlers run before the change happened: 4](#_Toc514324770)

[Debug messages to appear on certain verbose: 4](#_Toc514324771)

[Configuration files: 4](#_Toc514324772)

[9 . ansible-inventory-grapher : to visualize inventory hierarchies 5](#_Toc514324773)

[pip install ansible-inventory-grapher 5](#_Toc514324774)

[Variables: 5](#_Toc514324775)

[Host variables: variables for single host 5](#_Toc514324776)

[Vars\_prompt: provide value at rutime 5](#_Toc514324777)

[Encyptingvariables: anisble\_vault 5](#_Toc514324778)

[Set facts..to gather facts from exsisting facts 6](#_Toc514324779)

[Roles: https://galaxy.ansible.com/ for predefined roles: 6](#_Toc514324780)

[Tools: 7](#_Toc514324781)

[Assignments : 7](#_Toc514324782)

# Ansible ad-hoc command cheat sheet:

## Finding the syntax of a module:

ansible-doc <module name>

## PlayBooks:

Structure:

 Connection specification: hosts, forks, serial

 User specification: remote\_user, become, become\_user etc.

 Variable inclusion: vars, vars\_files, vars\_promptetc.

 Logic before tasks: pre\_tasks, roles

 Tasks and handlers: tasks, handlers

## Adding conditions in task:

-name: <task name>

<moule>

when: <condition>

**Example:**

|  |
| --- |
| - name: install httpd (Red Hat)  yum:  name: httpd  when: 'ansible\_distribution == "RedHat"' |

## Ansible Loops

<http://docs.ansible.com/ansible/latest/user_guide/playbooks_loops.html>

1. **With\_items**

|  |
| --- |
| - name: add database users  mysql\_user:  - name: "{{ item.name }}"  password: "{{ item.password }}"  with\_items:  - name: bob  password: abc123  - name: sys  password: superSekrit! |

## Become a different user:

Become: <yes no > to escalate privilages

Become\_user: <username>

Become\_method: <sudo>

## Include other palybooks or tasks:

Include: <playbook name>

Tasks:

* Include: <task.yml>

### Check mode and diff mode ( -D --diff how the file will change before it changed)

### 8. Command and shell readonly

- name: get list of files in a directory

command: ls /path/to/directory

register: directory\_contents

changed\_when: false

check\_mode: true

## Templates—jinja language

$db\_host = '{{ database\_host }}';

$db\_name = '{{ database\_name }}';

$db\_user = '{{ database\_username }}';

$db\_pass = '{{ database\_password }}';

$db\_port = {{ database\_port}};

Populated:

- template:

src: path/to/template.j2

dest: /path/to/result

## Handlers: tasks that are fired when a task makes a change

tasks:

- name: install httpd configuration file

template:

src: etc/httpd/conf/httpd.conf.j2

dest: /etc/httpd/conf/httpd.conf

notify: reload apache

handlers:

- name: reload apache

service:

name: apache

state: reloaded

## Handlers run before the change happened:

- meta: flush\_handlers

10. verbose –v , -vv, -vvv ,-vvvv ,-vvvvv

## Debug messages to appear on certain verbose:

debug:

msg: "This will appear with -vv but not before"

verbosity: 2

├── ansible.cfg

├── inventory

├── library

├── playbooks

└── plugins

   ├── filter

   └── lookup

## Configuration files:

[defaults]

Hosts\_file = ./inventory—can containe hosts\_vars or group\_vars

roles\_path = ./roles

library = ./library

filter\_plugins = ./plugins/filter

lookup\_plugins = ./plugins/lookup

callback\_whitelist = profile\_tasks,timer

[ssh\_connection]

pipelining = True

control\_path = %(directory)s/ssh-%%h-%%p-%%r

## 9 . ansible-inventory-grapher : to visualize inventory hierarchies

## pip install ansible-inventory-grapher

ansible-inventory-grapher -q target | \

dot -Tpng | display png:-

ansible-playbook playbooks/simple/add-inventory-graph.yml

## Variables:

## Host variables: variables for single host

## Vars\_prompt: provide value at rutime

- hosts: certificate\_authority

vars\_prompt:

- name: ca\_password

prompt: "Please enter your CA password"

tasks:

- name: sign certificate

command: openssl ca -in req.pem \

-out newcert.pem -passin env:CA\_PASSWD

environment:

CA\_PASSWD: "{{ ca\_password }}"

* Registered variables: store the results of a task
* Include\_vars: toinclude variables as task on playbook
* Var\_files : for more tha one file, but cannot include it in roles
* In roles use vars/main.yml : for variables used in roles

Command line extra vars are useful for setting configuration at run-time. –extra-vars

 Set lots of variables at once by including a variables file using **-e @filename.yml** — can be useful for overriding defaults during an outage

## Encyptingvariables: anisble\_vault

 create: ansible-vault create secrets.yml

 edit: ansible-vault edit secrets.yml

 view: ansible-vault view secrets.yml

 encrypt existing file: ansible-vault encrypt secrets.yml

 decrypt existing file: ansible-vault encrypt secrets.yml

 change password: ansible-vault rekey secrets.yml

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

## Set facts..to gather facts from exsisting facts

**This module aloows to set new variables**

## Roles: <https://galaxy.ansible.com/> for predefined roles:

**Example:**   
$ ansible-galaxy install 10forge.docker

ansible-galaxy init rolename

ansible-galaxy install -p playbooks/application/env/roles \

-r playbooks/application/env/requirements.yml –f

**Role versions**:

git pull upstream master

git tag v2.3

git push upstream v2.3

testrole/

├── defaults ├── tasks

│   └── main.yml │   └── main.yml

├── files ├── templates

├── handlers ├── tests

│   └── main.yml │   ├── inventory

├── meta │   └── test.yml

│   └── main.yml └── vars

└── README.md └── main.yml

## Tools:

1. Syntax Check: --syntax-check

$ ansible-playbook --syntax-check playbooks/simple/broken-syntax.yml

1. **Ansible lint**
2. Code reviews: git diff master | ansible-review
3. git ls-files | xargs ansible-review

## Inventory groups:

**Group of groups :**

[all:children]]

loadbalancer

web

database

## Assignments :

* uses the debug task to show amount of memory free on the target host
* runs a debug task if a variable run\_me is set.
* Stores the contents of the results of listing the directory of httpd\_directory
* Using the [resmo.mysql](https://galaxy.ansible.com/resmo/mysql/) from galaxy.ansible.com, install and configure a database called catalogue:
* database\_name: shop
* database\_user: shop
* database\_password: sh0p
* A starting playbook exists in playbooks/with\_roles/install\_db.yml, you'll need to install the role, and set up the inventory.

**GDW Ec2 creation and oracle client**

Provision\_EC2\_Instance\_Pipeline :

1. Check DNS
2. Providionec2 instance:

Runtime: ..extra vras.. <key=value>..buildnumber,ami,keypair,platform,vpc,subnet,roleprofile,volumesize,instancetype…..-----🡪 yml file

|  |
| --- |
| Hosts, roles-------------jdw-jumpboxrole |

Fodlerrole

|  |
| --- |
| Files—cloudformation json script,ip.txt,stackname.txt  Tasks: Set \_fact : stackname, create a stack..using cloud formationmodule of ansible –give template paratmeters to template , add ip to host file using lineinfile,stackname to stackname.txt file  Templates:  Vars: main.yml file has vars: template ( gdwjson) and default amuid and other variables |

Provsion dns

Gdwoneclick:

Install oracleclient: ….

Create group using group tasks create gid, create user using user task create uid , gid , create base directory , copy zip file to base directory,install yum packages..with item loop, run shell to inatall oracle client, update bash profile using line in file module ,create tns ora ,blockinfile command ,

Cloud formation stack: automating resource provising example of infrastructure as code

Templates..can be text file,json can pass parameters for different stacks

Stack

|  |  |
| --- | --- |
| |  | | --- | | Resources: like..ec2,s3,vpc,a unit of deployemnt | |