

Ansible Cheat Sheet

Last update: Oct 18, 2018

# ansible CLI	
-i	Inventory path
-m	Module name
-a	Module arguments
-b	Become, add -K to prompt for password
-v(vvv)	Verbosity
--check, -C	Shows what would change
--one-line, -o	Try to 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 -a "filter='facter_*'"
In a playbook	{{ ansible_fact }}
Include fact in inventory	[group] host1 admin_user=jane
Local facts	place .fact file into /etc/ansible/facts.d on target node
Get Local facts	# ansible targetName -m setup -a "filter=ansible_local"
Delegated Facts	
Any facts gathered by a delegated task are assigned by default to the delegate_to host, instead of the host which actually produced the facts	
The delegate_facts directive can be set to True to assign the gathered facts from the task to the delegated host instead of the 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 the 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 someserver delegate_to: someserver
Delegate task to a host group	- name: restart web servers service: name: memcached state: restarted delegate_to: "{{ item }}" with_items: "{{ groups['webservers'] }}"
To local machine	delegate_to: localhost
Simple Loops	
	yum: name: "{{ item }}" state: present with_items: - 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
Roles	
In a playbook	roles: <ul style="list-style-type: none"> - role1 - role2
Conditional Execution	
	- shell: echo "only on Red Hat 7" when: (ansible_distribution == "RedHat" ansible_distribution_major_version == "7")
	- shell: echo "I've got '{{ foo }}'!" when: foo is defined
Handlers	
Declare/Notify in Task	- name: install apache action: package name: apache2 state: latest notify: - start apache2
Execute	handlers: - name: start apache2 action: service name=apache2 state=started
Common Modules with Examples	
Module: Command	
Change directory to "chdir" and will only run when "creates" doesn't exist	- name: command example command: /usr/bin/make_db.sh arg1 args: chdir: somedir/ creates: /path/to/database register: myoutput
Module: Get_URL	
Pull a file from a remote source	- name: get_url example get_url: url: http://example.com/path/file.conf dest: /etc/foo.conf mode: 0440 backup: yes no

Module: Shell	
Redirect	shell: somescript.sh >> somelog.txt
Env vars	shell: echo "\$HOME"
Module: Service	
Manipulate a service/daemon	- name: service example service: name: httpd state: started stopped restarted enabled: yes
Module: Copy	
Copy new ntp.conf file into place, backing up the original if it differs from the copied version	- name: copy example copy: src: /mine/ntp.conf dest: /etc/ntp.conf owner: root group: root mode: 0644 backup: yes
Module: File	
Create a File	- name: file example file: path: /etc/foo.conf owner: foo group: foo mode: 0644
Dir (create if not exist)	- name: file example for dirs file: path: /etc/some_directory state: directory mode: 0755
Module: Package	
Manipulate a package	- name: package example package: name: ntpdate state: present absent latest
Module: LineInFile	
Insert using regex	- name: example insert lineinfile: path: /etc/selinux/config regexp: '^SELINUX=' line: 'SELINUX=enforcing'
Remove using regex	- name: example remove lineinfile: path: /etc/sudoers state: absent regexp: '^%wheel'