**Ansible Ad-Hoc Commands**

An Ansible ad-hoc command uses the /usr/bin/ansible command-line tool to automate a single task on one or more managed nodes. Ad-hoc commands are quick and easy, but they are not reusable. So why learn about ad-hoc commands first? Ad-hoc commands demonstrate the simplicity and power of Ansible

[**Why use ad-hoc commands?**](https://docs.ansible.com/ansible/latest/user_guide/intro_adhoc.html#why-use-ad-hoc-commands)

* [Use cases for ad-hoc tasks](https://docs.ansible.com/ansible/latest/user_guide/intro_adhoc.html#use-cases-for-ad-hoc-tasks)
  + [Rebooting servers](https://docs.ansible.com/ansible/latest/user_guide/intro_adhoc.html#rebooting-servers)
  + [Managing files](https://docs.ansible.com/ansible/latest/user_guide/intro_adhoc.html#managing-files)
  + [Managing packages](https://docs.ansible.com/ansible/latest/user_guide/intro_adhoc.html#managing-packages)
  + [Managing users and groups](https://docs.ansible.com/ansible/latest/user_guide/intro_adhoc.html#managing-users-and-groups)
  + [Managing services](https://docs.ansible.com/ansible/latest/user_guide/intro_adhoc.html#managing-services)
  + [Gathering facts](https://docs.ansible.com/ansible/latest/user_guide/intro_adhoc.html#gathering-facts)

**Syntax**

**$ ansible [host-pattern] -m [module] -a "[module options]" –i inventory**

**a ==> passes the arguments**

**1] $ ansible all -m ping Ping remote server**

**2] $ ansible-doc-l List all modules installed on a system**

**3] $ ansible-doc [module\_name] Info about the module**

**4] $ ansible all --list-hosts List all servers inside of inventory**

**5] $ ansible all --list-hosts -v v tells which inventory is being used**

**Ansible Command-line options**

|  |  |
| --- | --- |
| **CONFIGURATION FILE DIRECTIVES** | **COMMAND-LINE OPTION** |
| **Inventory** | **-i** |
| **remote\_user** | **-u** |
| **Become** | **--become, -b** |
| **Become-method** | **--become-method** |
| **Become\_user** | **--become-user** |
| **Become\_ask\_pass** | **--ask=become-pass, -k** |

**USER module**

**$ansible all –m user –a “name=ec2-user state=present”**

name: venkat ----🡪mandatory

password: password: "{{ password | password\_hash('sha512') }} ----🡪 password is a variable

uid: 10001

state: present or absent

remove: yes/no ------🡪This only affects state=absent

group:

force: yes/no --🡪 This only affects state=absent

append: yes/no -🡪

**COMMAND module is the default**

Run arbitory commands

**$anisble all –m command –a “hostname”**

**$ansible all –m command –a “hostanme” –o --🡪displays o/p in sinfle line**

**SHELL module**

**$anisble all –m shell –a “hostname”**

**$ansible all –m shell –a “hostanme” –o --🡪displays o/p in sinfle line**

**SHELL VS COMMAND**

* With the Command module the command will be executed without being proceeded through a shell. As a consequence some variables like $HOME are not available. And also stream operations like <, >, | and & will not work. Pipeline, redirection.
* The Shell module runs a command through a shell, by default /bin/sh. This can be changed with the option executable. Piping and redirection are here therefor available.
* The command module is more secure, because it will not be affected by the user’s environment

**COPY MODULE**

**$ansible localhost -m copy -a "content="welcome" dest="/home/devops/playbook/file" owner=devops mode=777 group=devops force=yes"**