



Ansible Ad hoc Commands – Ansible Tutorial for Beginners

📅 November 13, 2017

This article explains you about the Ansible Ad-Hoc Commands. At the end of this article, you will be able to understand What is Ansible ad-hoc commands and how to use ansible tool with some examples.

In the previous posts, we have explained the below topics. Refer those links to understand this topic from basics.

1. [What is Ansible, How Ansible works, Ansible Introduction, Ansible Basic Tutorials](#)
2. [Ansible Inventory Introduction – Ansible Beginner Tutorials](#)
3. [Ansible Ad hoc Commands – Ansible Tutorial for Beginners](#)
4. [Managing Ansible Configuration Files Explained with Examples](#)
5. [Understanding Ansible Playbook – Write your First Playbook](#)
6. [Ansible Roles Explained with Examples – Ansible Tutorials](#)
7. [How to use Ansible Vault to Protect Ansible Playbooks](#)

Lets get started.

ALSO WATCH THIS "ANSIBLE AD-HOC COMMAND" TUTORIAL VIDEO FREE ON OUR YOUTUBE CHANNEL

Ansible Ad-Hoc Commands with Examples Explained - Ansi...



What is Ansible Ad-hoc commands?



An Ansible ad-hoc commands can be used to do some quick task to perform some simple operations through the command line.

But real automation works using playbooks only. you may ask me if playbook is the real automation then Why do we use ad-hoc commands.

Answer is so simple, Everytime you may not needed to perform a big operations, Some time we are required to perform some simple tasks. So we dont need to write a big playbooks. Also ad-hoc command is a easiest way to explore Ansible for beginners.

Assume that, you wanted to restart or power off all of your infrastructure servers or if you wanted to get an uptime of all servers or to get an kernel versions or os information something like a small and quick tasks. On that cases, you can use a ad-hoc commands instead of writing a playbooks.

We will explain you Ansible Playbooks in detailed in the next article. Playbooks can be used to do a many tasks.

For example, if you want to install http web server, you are required to perform some multiple tasks.

1. Install a httpd package
2. Configuring the webserver
3. Starting and Enabling the httpd service.

For this kind of multiple task, you can write a playbook and execute it.

Even you can use the same task using the ad-hoc command, but for each task you need to run a command manually.

Hope you have got an idea? Why do we Ansible Ad-hoc commands and Playbooks?

so syntax of ad-hoc command is,

```
command hostgroup module/options[arguments]
```

where,

command is ansible

host group - It specifies on what machines, task to be performed

module or arguments - What actions to be performed.

Lets take an example, Assume that, you wanted to get a uptime of a list of hostgroup named "servers" as per specified in Ansible Inventory host file. Refer this link to know [what is Ansible Inventory](#). Also make sure you have [configured passwordless SSH login authentication](#). Because Ansible works over SSH Connection by default.

```
[root@localhost ~]# ansible servers -a "uptime"
node1 | SUCCESS | rc=0 >> 21:53:10 up 7 min, 2 users, load average: 0.49, 0.35, 0.20
node2 | SUCCESS | rc=0 >> 21:53:10 up 7 min, 1 user, load average: 0.01, 0.04, 0.05
```

Here we have used '-a' option rather than using a direct modules. This would execute a command 'uptime' on hostgroup "servers" and will give you the output of uptime in the screen.

Also, you can use a module rather than using a direct command. Some commonly used modules are apt/yum, copy, ec2, file, service, template, and user.

Let me use a module ping to find the server status instead of using direct command "ping".

```
[root@localhost ~]# ansible servers -m ping
node1 | SUCCESS => {  "changed": false,  "failed": false,  "ping": "pong"}node2 | SUCCESS => {
"changed": false,  "failed": false,  "ping": "pong"}
```

Here we have used a module "ping" with option "-m".

To know the list of available modules in ansible, execute the command.

```
[root@localhost ~]# ansible-doc -l | more
a10_server                Manage A10 Networks AX/SoftAX/Thunder/vThunder devices' server
object.a10_server_axapi3  Manage A10 Networks AX/SoftAX/Thunder/vThunder
devicesa10_service_group  Manage A10 Networks AX/SoftAX/Thunder/vThunder
devices' service groups.a10_virtual_server  Manage A10 Networks
AX/SoftAX/Thunder/vThunder devices' virtual servers.accelerate  Enable
accelerated mode on remote nodeaci_aep      Manage attachable Access Entity
Profile (AEP) on Cisco ACI fabrics (infra:AttEntityP).....
```

Documentation for each module can be accessed from the command line with the ansible-doc tool. To know the required arguments and also to get a help of a particular module. Execute the below command.

```
[root@localhost ~]# ansible-doc modulename
```

For example, to find the detailed information about the module "user".

```
[root@localhost ~]# ansible-doc user
```

where "user" is a module used to manage the user accounts.

Note : If any options starts with "=" in the above command, then its a mandatory option for the module.
Example :

```
- append
  If `yes`, will only add groups, not set them to just the list in `groups`.
  (Choices: yes, no)[Default: no]
= name
  Name of the user to create, remove or modify.
  (Aliases: user)
```

"name" is mandatory option to be used in user module.

Example 1: Using module, Create a user called "john" using a module "user".

```
[root@localhost ~]# ansible servers -m user -a "name=john password=redhat"
node1 | SUCCESS => {  "changed": true,  "comment": "",  "createhome": true,  "failed": false,
"group": 1000,  "home": "/home/john",  "name": "john",  "password": "NOT_LOGGING_PASSWORD",
"shell": "/bin/bash",  "state": "present",  "system": false,  "uid": 1000}node2 | SUCCESS => {  "changed":
```

```
true, "comment": "", "createhome": true, "failed": false, "group": 1000, "home": "/home/john",
"name": "john", "password": "NOT_LOGGING_PASSWORD", "shell": "/bin/bash", "state": "present",
"system": false, "uid": 1000}
```

Where,

-m is a option to use a module

user is a module name to manage the user accounts

-a is options to pass the arguments "name=john password=redhat"

Above command will create a user call "john" in all the servers mentioned in Ansible Inventory file.

Example 2: Without using a module "yum", install a httpd package.

```
[root@localhost ~]# ansible all -a "yum -y install httpd"
```

Where, -a is a option to pass the arguement. If no module is specified, then it consider that as a comand and execute at the client servers.

Example 3: Using a module "yum", install a httpd package.

```
[root@localhost ~]# ansible all -m yum -a "name=httpd state=present"
```

That's all about the basics of Ansible Ad-hoc command how to use. This would help you when we write a playbooks. [ansible](#), [Ansible Ad hoc commands](#), [Ansible Ad hoc commands Tutorials](#), [Ad hoc commands](#), [Ansible Ad hoc commands management](#), [ansible Ad hoc commands with examples](#), [ansible Ad hoc commands realtime examples](#), [Ansible Ad hoc commands configuration](#),
Going forward, we will play more with ansible tool with some test cases.

Support Us: Share with your friends and groups.

Stay connected with us on social networking sites, Thank you.

[YouTube](#) | [Facebook](#) | [Twitter](#) | [Pinterest](#) | [Rss](#)

[ansible Ad hoc commands examples](#), [ansible tutorial](#), [ansible tutorial for beginners](#), [ansible beginners tutorial](#), [ansible complete tutorial](#), [ansible dynamic Ad hoc commands](#), [ansible tower Ad hoc commands](#), [ansible ad hoc multiple commands](#), [ansible ad hoc ping](#), [ansible ad hoc shell command](#)

Next

How to Create a LDAP Users and Groups using
LDIF file
(<https://www.learnitguide.net/2017/11/how-to-create-ldap-users-and-groups.html>)

Previous

Ansible Inventory Introduction - Ansible
Beginner Tutorials
(<https://www.learnitguide.net/2017/10/ansible-inventory-introduction-tutorial.html>)

POST A COMMENT

[Disqus](#) [Facebook](#) [Blogger](#)

0 Comments

learnitguide

 Login ▾ Recommend Tweet Share

Sort by Newest ▾



Start the discussion...

LOG IN WITH

OR SIGN UP WITH DISQUS 

Be the first to comment.

[!\[\]\(6befd466863f06afb75445d91429f055_img.jpg\) Subscribe](#) [!\[\]\(688aa1e033730bef6d86099638866a42_img.jpg\) Add Disqus to your site](#) [Add DisqusAdd](#) [!\[\]\(a1c00788e46bf4d408b69524d2b554d8_img.jpg\) Disqus' Privacy Policy](#) [Privacy Policy](#) [Privacy Policy](#)