

Ansible Ad-Hoc Commands



What is Ansible?

- ▶ **Ansible** is a free and open source automation tool that allows system administrators to configure and control hundreds of nodes from a central server without the need of installing any agents on the nodes.
- ▶ It relies on the **SSH** protocol to communicate with the remote nodes. Compared to other management tools such as **Puppet** and **Chef**, **Ansible** comes out as the favorite due to its ease of use, and installation.
- ▶ If you take other configuration management tools like puppet, chef, and CFEngine, server software is installed on one machine, and client machines are managed through the agent. Wherein Ansible, the nodes are managed by controlling machine (Ansible server) over SSH, so there won't be any agent running on node machines.



Ansible Inventory File

The Ansible inventory file defines the hosts and groups of hosts upon which commands, modules, and tasks in a playbook operate.

The file can be in one of many formats depending on your Ansible environment and plugins.

The default location for the inventory file is `/etc/ansible/hosts`.

```
#vim /etc/ansible/hosts
```

```
[web]
```

```
192.168.1.3
```

```
192.168.1.4
```

```
192.168.1.5
```

```
:wq
```



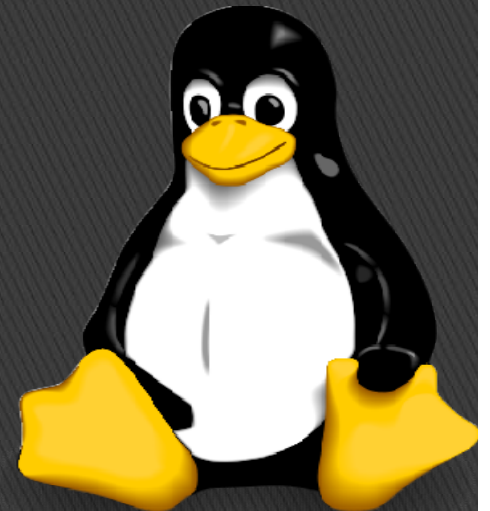
Important terms used in Ansible

- ▶ **Ansible server:**
 - ▶ The machine where Ansible is installed and from which all tasks and playbooks will be run
- ▶ **Module:**
 - ▶ Basically, a module is a command or set of similar Ansible commands means to be executed on the client-side
- ▶ **Task:**
 - ▶ A task is a section that consists of a single procedure to be completed
- ▶ **Inventory:**
 - ▶ File containing data about the ansible client servers. Defined in examples as hosts file
- ▶ **Play:**
 - ▶ Execution of a playbook



Ansible ad-hoc commands

- ▶ One of the simplest ways Ansible can be used is by using ad-hoc commands. These can be used when you want to issue some commands on a server or a bunch of servers. Ad-hoc commands are not stored for future uses but represent a fast way to interact with the desired servers.



Ansible ad-hoc command options

Option & Description

-b

To instruct ansible to become root user before executing the task

-m

It represents the module

-a, --args

Insert module arguments.

-u

Connect as different user (remote SSH user)

-h, --help

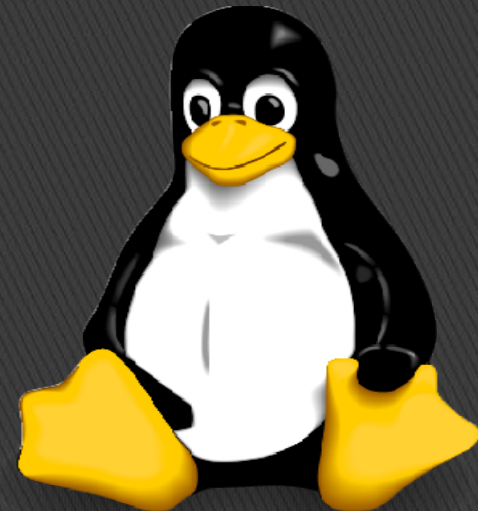
Display help content.

-v, --verbose

Run commands in verbose mode.

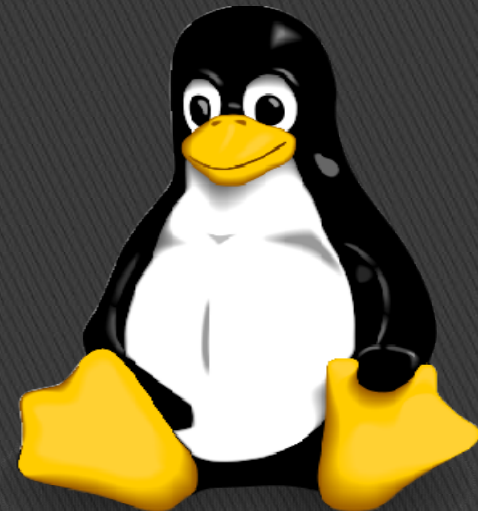
-i

Inventory file



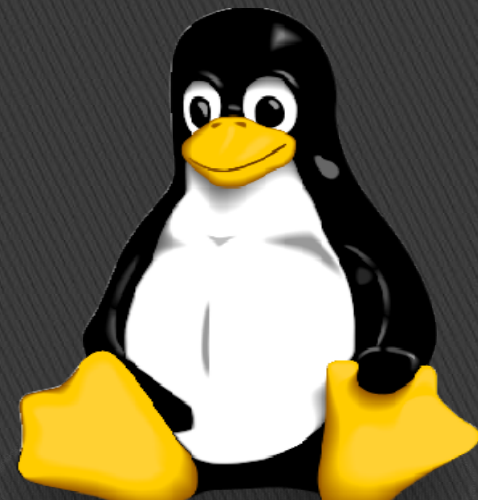
Checking connectivity with ad-hoc commands

- ▶ `$ansible -i /etc/ansible/hosts web -m ping`



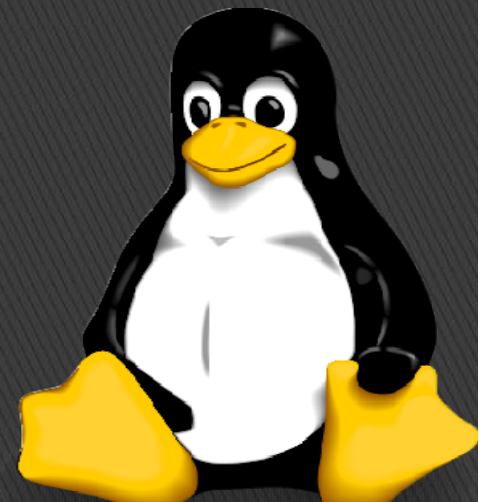
To check the partitions on all remote hosts

```
$ansible -i /etc/ansible/hosts web -m command -a "df -h"
```



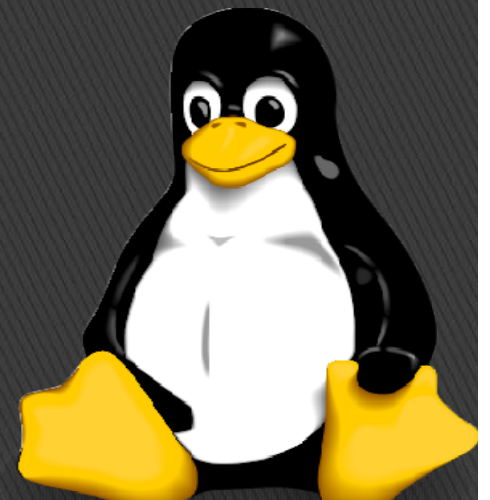
Check memory usage on all remote hosts.

```
$ansible -I /etc/ansible/hosts web -m command -a "free -m"
```



Checking Uptime for all servers.

```
$ansible -I /etc/ansible/hosts web -m command -a "uptime"
```



Checking connectivity with ad-hoc commands with limit

- ▶ You can issue the same command only on a specific host if needed.

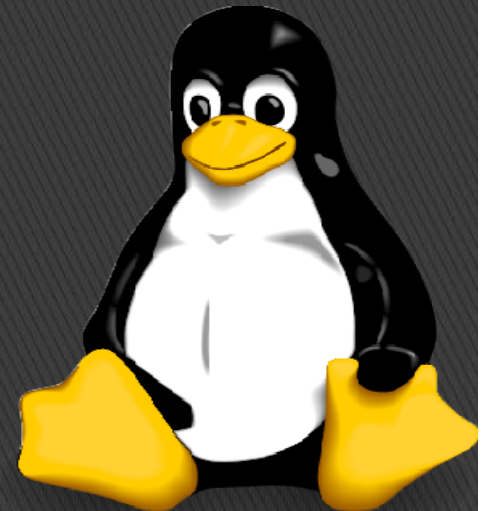
```
$ ansible -i /etc/ansible/hosts web -m ping --limit 192.168.1.2
```

- ▶ **Explanation:**

- ▶ --Limit

parameter can be used to issue commands only on specific hosts in the host's file

- ▶ Name of the host as defined in the inventory file



Copy file with ad-hoc commands

- ▶ Copy the file “qq.txt” on all hosts in the inventory file

```
$ansible -i /etc/ansible/hosts web -m copy -a  
"src=/etc/ansible/qq.txt dest=/home/" -b -u root
```

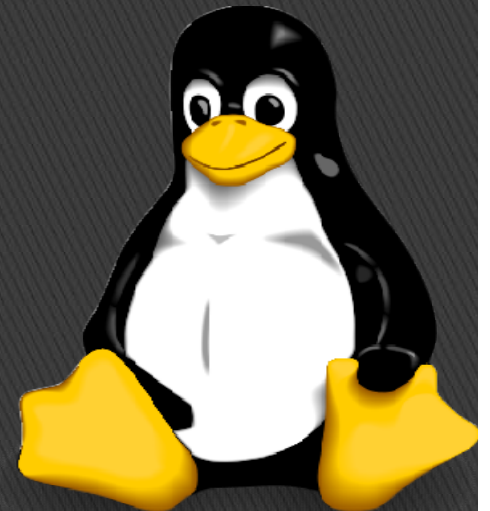
Note: (complete following steps before run above command)

Create qq.txt file in server in

```
$sudo touch /etc/ansible/qq.txt
```

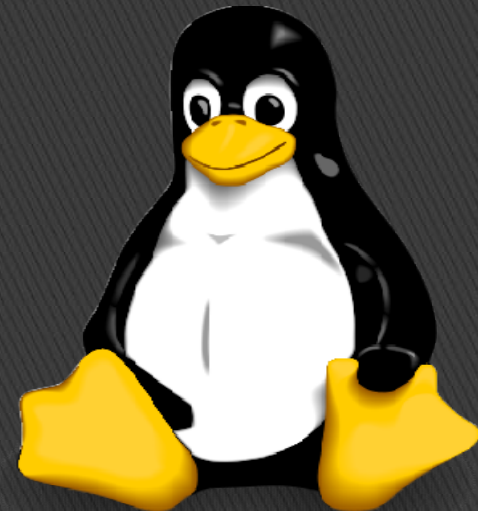
Copy ssh-copy-id for root account

```
$ssh-copy-id root@192.168.43.49
```



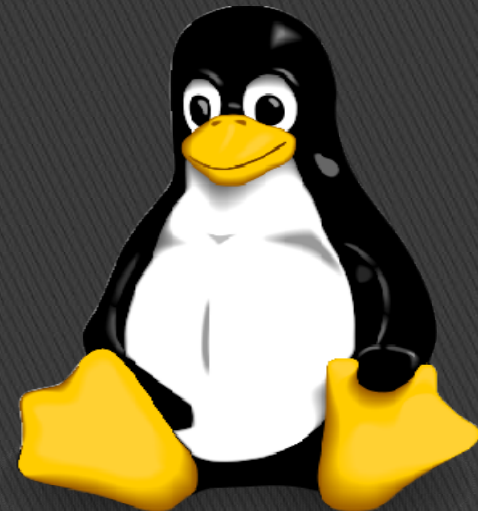
Create a Directory with 755 permission using ansible ad hoc command

```
$ ansible -i /etc/ansible/hosts web -m file -a  
"path=/mnt/data state=directory mode=0755" -b -u  
root
```



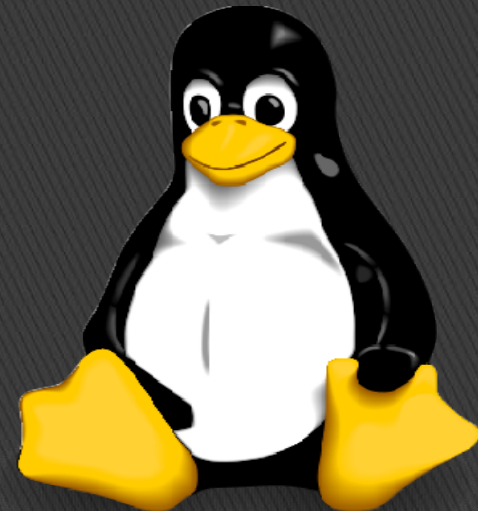
Create a file with 755 permission using ansible ad hoc commands

```
$ ansible -i /etc/ansible/hosts web -m file -a "path=/mnt/imp.txt  
state=touch mode=0755" -b -u root
```



Change ownership of a file using ansible ad hoc command

- ▶ `$ ansible -I /etc/ansible/hosts web -m file -a "path=/mnt/a1 group=sachin owner=sachin" -b -u root`



Managing packages with ad-hoc commands

- ▶ Using Ansible's ad-hoc commands, you can also install software packages to your managed hosts. All you need to issue is a one-liner to install a single package to a group of managed hosts.

Install the Apache httpd server on your

- ▶ `$ansible -i /etc/ansible/hosts web -m yum -a "name=httpd state=present" -b -u root`

Note:

Yum server must be working on client system (setup proper Ip address in `/etc/yum.repos.d/server.repo` file)



Managing services with ad-hoc commands

- ▶ Ansible's ad-hoc commands to start and enable the httpd service, so your Apache httpd web server is up and running.

```
$ansible -i /etc/ansible/hosts web -m service -a "name=httpd  
enabled=yes" -b -u root
```

- ▶ To start, restart, and stop a service, change the value of the state argument to started to start the service, restarted to restart a service, and stopped to stop a service:

```
$ansible -i /etc/ansible/hosts web -m service -a "name=httpd  
state=started" -b -u root
```



