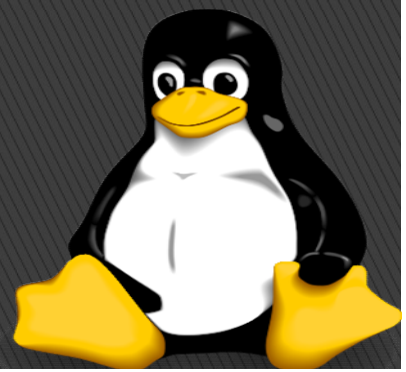
Install & Configure of Ansible

What is Ansible?

Ansible is the leading free and opensource configuration management, automation tool. With Ansible, Linux users can control and manage hundreds of servers from a single server known as the Ansible Control Node.

Ansible makes it possible for System and Network administrators to provision software and settings on various network devices. Ansible can be used to manage both Unix and Windows host systems. Ansible was initially released 7 years ago.

The latest stable version at the point of writing this article is Ansible 2.10.5.Unlike many other automation tools such as Chef and Puppet, Ansible rides on SSH protocol to communicate with remote host systems. Without the need of installing agents on remote systems, Ansible comes highly recommended since very little CPU and RAM resources are used up by remote systems.



Install Ansible

#wget https://dl.fedoraproject.org/pub/epel/epel-release-latest-7.noarch.rpm

#yum install epel-release-latest-7.noarch.rpm

#yum update -y
#yum install git python-level python-pip openssl ansible -y

#ansible -version

Create user & set sudo privilege

```
#useradd    sachin
#passwd     sachin

#vim    /etc/sudoers
Sachin    ALL=(ALL)   NOPASSWD: ALL      (line  101)

:wq
```

Setup ssh on all system

#vim /etc/ssh/sshd_config

PermitRootLogin yes (remove #)

#PasswordAuthentication yes (remove #)
#PermitEmptyPasswords no
#PasswordAuthentication no (add #)

:wq
#systemctl restart sshd

start ssh service (must be running)

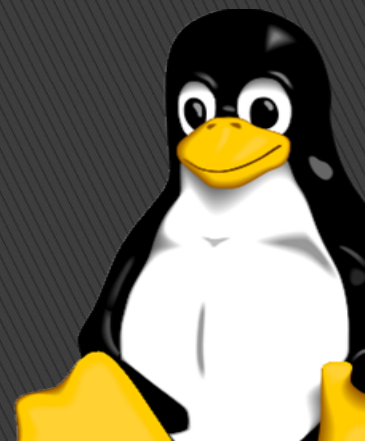
```
$sudo  systemctl  status  sshd
$sudo  systemctl   start   sshd
$sudo  systemctl   restart  sshd
$sudo  systemctl   enable sshd
```

we neet to create the host file in the /etc/ansible directory

```
$sudo  mkdir   /etc/ansible
$cd   /etc/ansible
$sudo   vim   hosts
[web-server]
192.168.1.3
192.168.1.4
```

Generate SSH key and copy it to remote servers

- $ssh-keygen
- $ssh-copy-id sachin@192.168.1.3



Use ping module to test ansible:

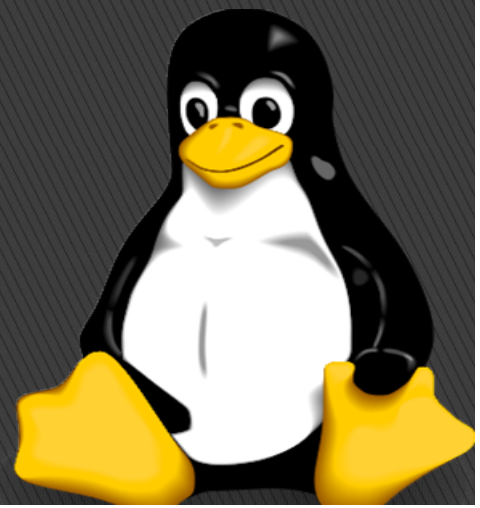
```
$ansible -i /etc/ansible/hosts web-server -m ping
```

or

```
$ansible -i hosts 192.168.1.3 -m ping
```

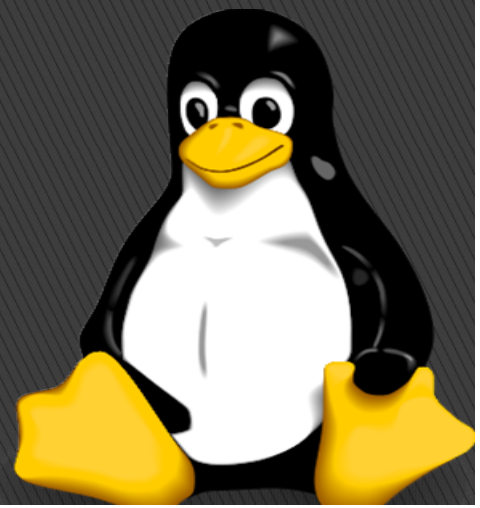
The -i option is used to provide path to inventory file. You should get the same output for hosts group name.

```
$ansible -i hosts web -m ping
```



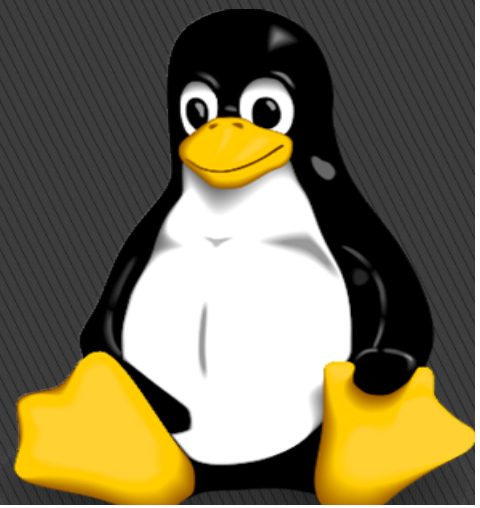
To check the partitions on all remote hosts

$ansible -m command -a "df -h" web-servers



Check memory usage on all remote hosts.

```
# ansible -m command -a "free -m" web-servers
```



Checking Uptime for all 3 servers.

```
# ansible -m command -a "uptime" web-servers
```