Python install

Ansible only working in a python base.

Step 1: Update and Refresh Repository Lists

```
sudo apt update
```

Step 2: Install Supporting Software

The software-properties-common package gives you better control over your package manager by letting you add PPA (Personal Package Archive) repositories. Install the supporting software with the command:

```
sudo apt install software-properties-common
```

Step 3: Add Deadsnakes PPA

Deadsnakes is a PPA with newer releases than the default Ubuntu repositories. Add the PPA by entering the following:

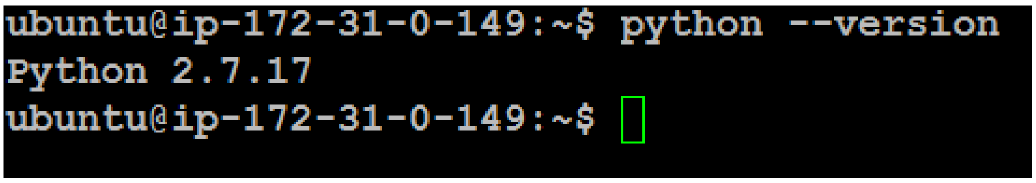
```
sudo add-apt-repository ppa:deadsnakes/ppa
```

The system will prompt you to press enter to continue. Do so, and allow it to finish. Refresh the package lists again:

```
sudo apt update
```

Step 4: Install Python 3

```
sudo apt install python3.8
```



```
ubuntu@ip-172-31-0-149:~$ python --version
Python 2.7.17
ubuntu@ip-172-31-0-149:~$
```

Ansible install

Ubuntu builds are available in a PPA.

You can update your system with unsupported packages from this untrusted PPA by adding ppa:ansible/ansible to your system's Software Sources.

```
sudo add-apt-repository --yes --update ppa:ansible/ansible
sudo apt install ansible
```

Creating key pair to access mange nodes

There are multiple ways to access the manage node from control node.

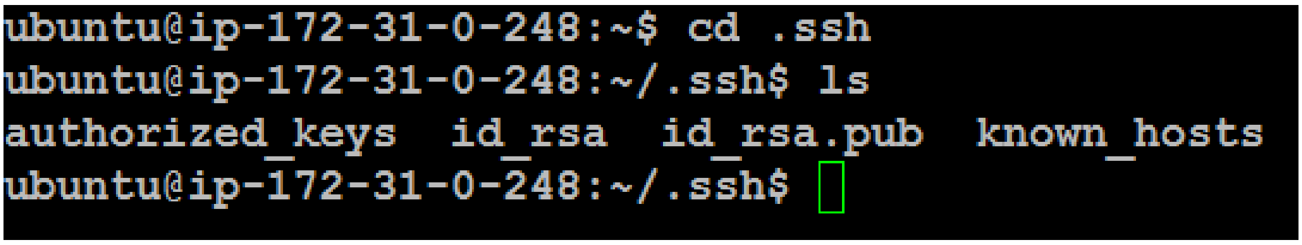
Here I used ssh protocol to access the manage node.

In a control node I had create two file,

id_rsa	→ private key pair
id_rsa.pub	→ public key pair

Copy a content of authorized_keys to id_rsa.pub

Copy a content of my .pem file to id_rsa

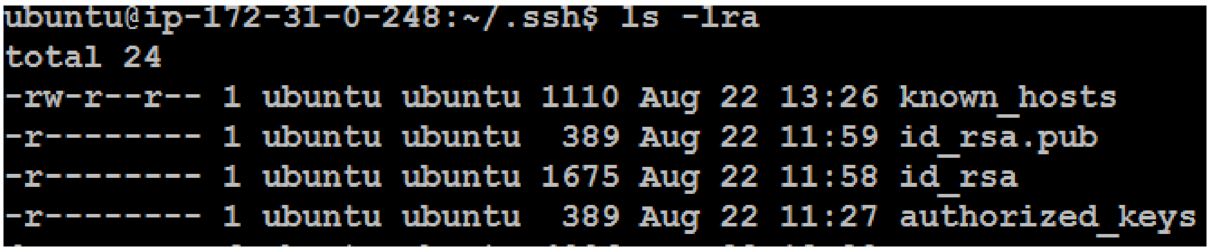


```
ubuntu@ip-172-31-0-248:~$ cd .ssh
ubuntu@ip-172-31-0-248:~/.ssh$ ls
authorized_keys  id_rsa  id_rsa.pub  known_hosts
ubuntu@ip-172-31-0-248:~/.ssh$
```

And I have changed a permission to all 3 files using chmod command.

Chmod 400 authorized_keys id_rsa id_rsa.pub

This will make those file is protected one.



```
ubuntu@ip-172-31-0-248:~/.ssh$ ls -lra
total 24
-rw-r--r-- 1 ubuntu ubuntu 1110 Aug 22 13:26 known_hosts
-r-------- 1 ubuntu ubuntu  389 Aug 22 11:59 id_rsa.pub
-r-------- 1 ubuntu ubuntu 1675 Aug 22 11:58 id_rsa
-r-------- 1 ubuntu ubuntu  389 Aug 22 11:27 authorized_keys
```

Add Manage node in control node

There are two file were create while install ansible.

One is for configure an ansible → ansible.conf

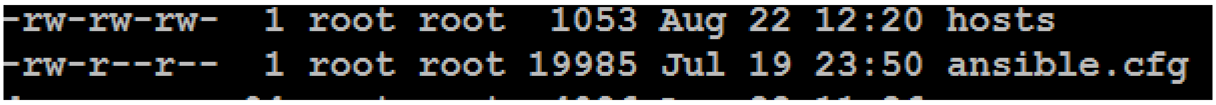
Second is for control a controls node. → hosts

Both a file in a directory /etc/ansible.

We have to add an instances IP address in a hosts file, ansible take those instances as his control nodes.

In a hosts file we have to permit a user to write the file then only those user can add the control nodes IP instances.

Chmod 666 hosts

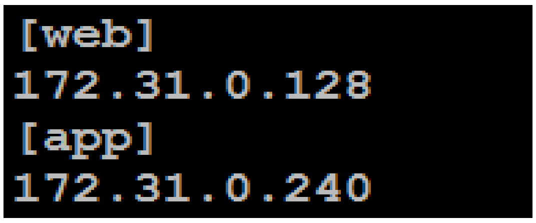


```
-rw-rw-rw-  1 root root  1053 Aug 22 12:20 hosts
-rw-r--r--  1 root root 19985 Jul 19 23:50 ansible.cfg
```

Grouping

A name provide inside a [] is take a group name by ansible.

If we want to call a particular instances we have to add inside the group to call that particular instances.



```
[web]
172.31.0.128
[app]
172.31.0.240
```

Install apache in web node

Here I install apache web server in a web group without using –i parameter.

Below command are used to install apache in Ubuntu

ansible web -m apt -a "name=apache2 state=present" –b

-m → is a module

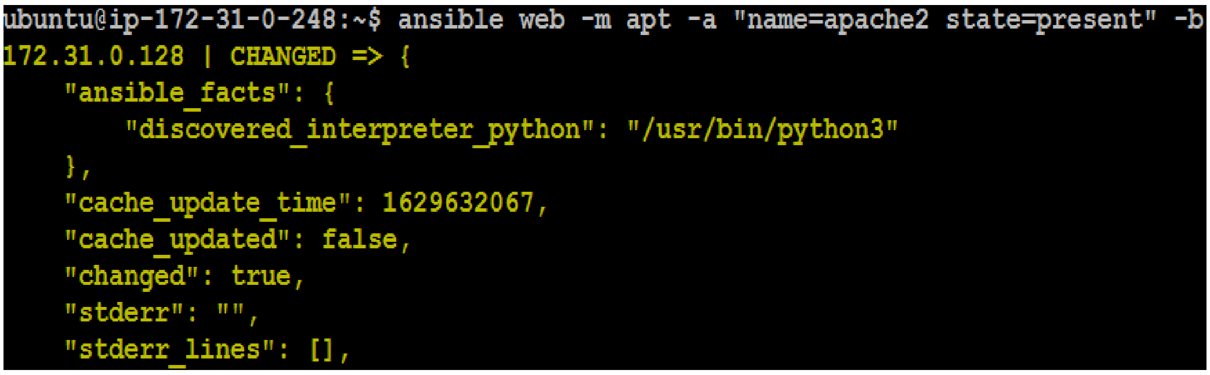
Apt→ is a command

-a → is an argument

Name → is a name of middleware

State → means install the middleware

-b → become a sudo user.

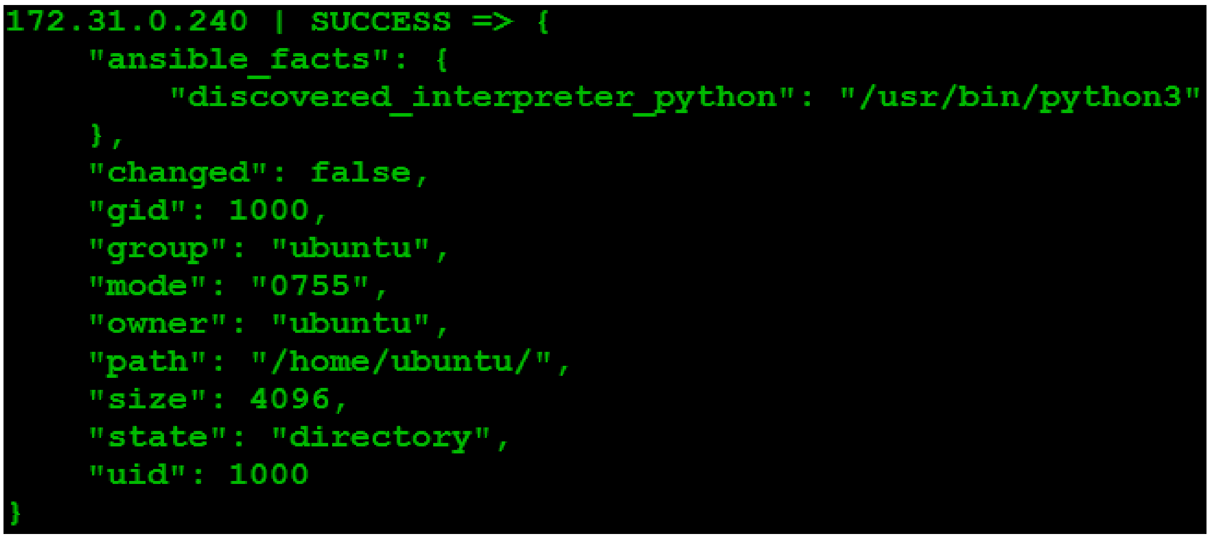


```
ubuntu@ip-172-31-0-248:~$ ansible web -m apt -a "name=apache2 state=present" -b
172.31.0.128 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "cache_update_time": 1629632067,
    "cache_updated": false,
    "changed": true,
    "stderr": "",
    "stderr_lines": [],
```

Creating a directory

In an ansible we can create a directory in control nodes through manage node. Below command are used to create a directory in control nodes.

ansible all -m ansible.builtin.file -a "dest=/home/ubuntu/test mode=755 owner=ubuntu group=ubuntu state=directory"

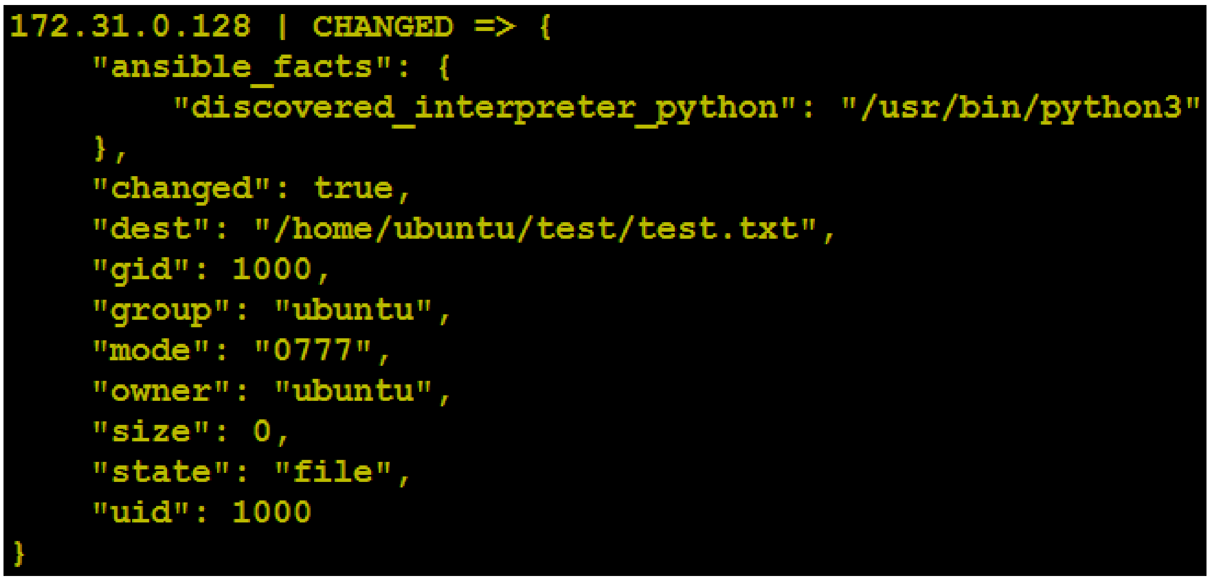


```
172.31.0.240 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "gid": 1000,
    "group": "ubuntu",
    "mode": "0755",
    "owner": "ubuntu",
    "path": "/home/ubuntu/",
    "size": 4096,
    "state": "directory",
    "uid": 1000
}
```

Creating a file in control nodes

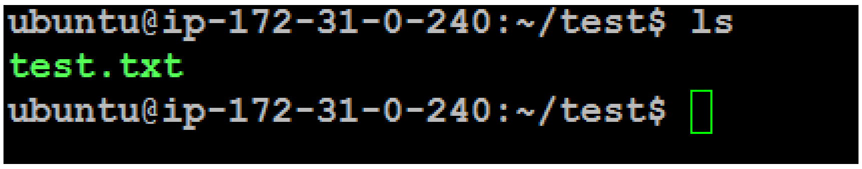
In an ansible we can create a file in control nodes through manage node. Below command are used to create a directory in control nodes.

ansible all -m ansible.builtin.file -a "path=/home/ubuntu/test/test.txt mode=777 owner=ubuntu group=ubuntu state=touch"



```
172.31.0.128 | CHANGED => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": true,
    "dest": "/home/ubuntu/test/test.txt",
    "gid": 1000,
    "group": "ubuntu",
    "mode": "0777",
    "owner": "ubuntu",
    "size": 0,
    "state": "file",
    "uid": 1000
}
```

In a state arguments that value of state must be one of: absent, directory, file, hard, link, touch, got.



```
ubuntu@ip-172-31-0-240:~/test$ ls
test.txt
ubuntu@ip-172-31-0-240:~/test$
```

Using windows environment

To access a control node we have to install python in windows server.

To configure a windows server as control node winRm would help us to connect the server.

Ansible requires PowerShell version 3.0 and .NET Framework 4.0 or newer to function on older operating systems like Server 2008 and Windows 7. The base image does not meet this requirement.

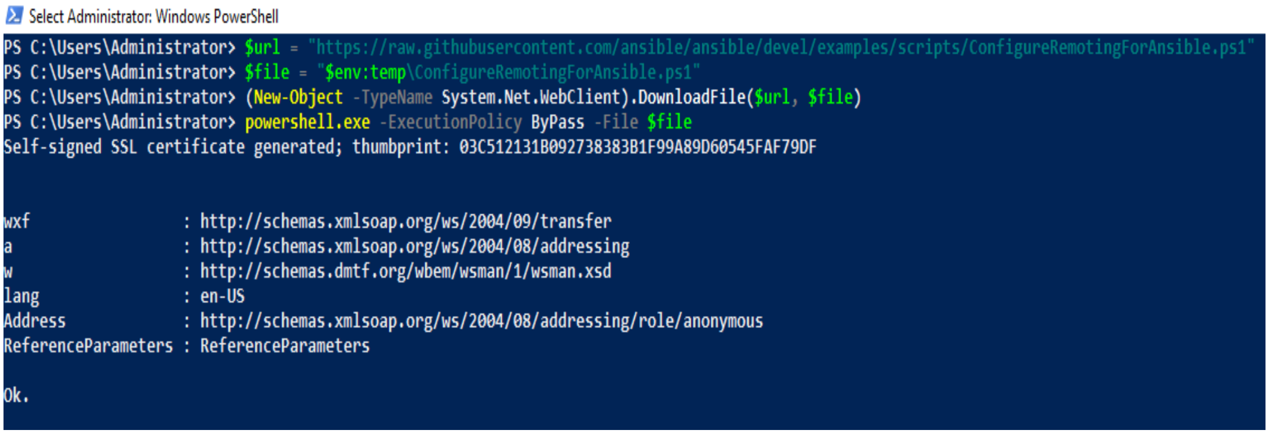
$psversiontable→ helps to identify the version

Once PowerShell has been upgraded to at least version 3.0, the final step is for the WinRM service to be configured so that Ansible can connect to it. There are two main components of the WinRM service that governs how Ansible can interface with the Windows host: the listener and the service configuration settings.

ConfigureRemotingForaansible.ps1 script is use to sets up both HTTP and HTTPS listeners self-signed certificate and enables the Basic authentication option on the service.

To use the ConfigureRemotingForaansible.ps1script run the below command

```
$url =
"https://raw.githubusercontent.com/ansible/ansible/devel/examples/scripts/Configure
RemotingForAnsible.ps1"
$file = "$env:temp\ConfigureRemotingForAnsible.ps1"
(New-Object -TypeName System.Net.WebClient).DownloadFile($url, $file)
powershell.exe -ExecutionPolicy ByPass -File $file
```



```
PS C:\Users\Administrator> $url = "https://raw.githubusercontent.com/ansible/ansible/devel/examples/scripts/ConfigureRemotingForAnsible.ps1"
PS C:\Users\Administrator> $file = "$env:temp\ConfigureRemotingForAnsible.ps1"
PS C:\Users\Administrator> (New-Object -TypeName System.Net.WebClient).DownloadFile($url, $file)
PS C:\Users\Administrator> powershell.exe -ExecutionPolicy ByPass -File $file
Self-signed SSL certificate generated; thumbprint: 03C512131B092738383B1F99A89D60545FAF79DF


wxf                 : http://schemas.xmlsoap.org/ws/2004/09/transfer
a                   : http://schemas.xmlsoap.org/ws/2004/08/addressing
w                   : http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd
lang                : en-US
Address             : http://schemas.xmlsoap.org/ws/2004/08/addressing/role/anonymous
ReferenceParameters : ReferenceParameters

Ok.
```

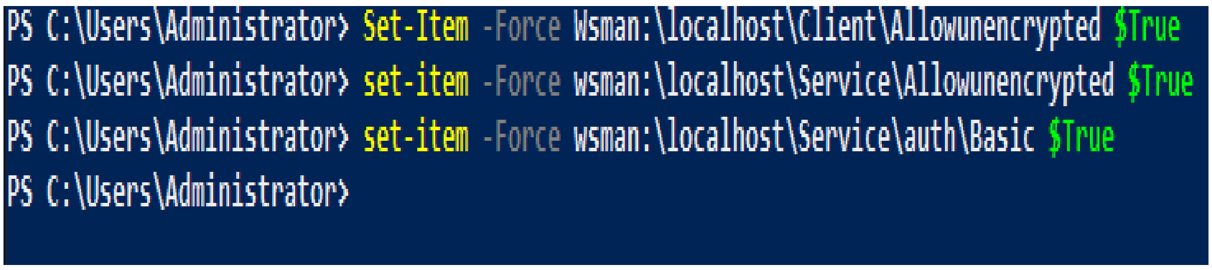
The WinRM services listens for requests on one or more ports. Each of these ports must have a listener created and configured.

To view the current listeners that are running on the WinRM service, run the following command:

```
winrm enumerate winrm/config/Listener
```

To enable remote scripting a winrm we have to run following command to unencrypt.

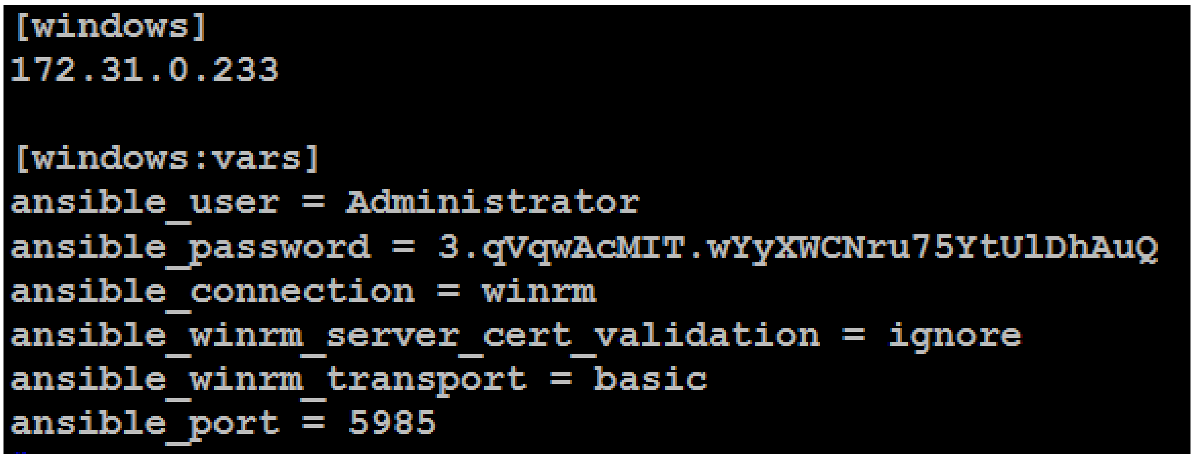
```
Set-Item -Force Wsman:\localhost\Client\Allowunencrypted $True
Set-Item -Force wsman:\localhost\Service\Allowunencrypted $True
Set-Item -Force wsman:\localhost\Service\auth\Basic $True
```


```
PS C:\Users\Administrator> Set-Item -Force Wsman:\localhost\Client\Allowunencrypted $True
PS C:\Users\Administrator> set-item -Force wsman:\localhost\Service\Allowunencrypted $True
PS C:\Users\Administrator> set-item -Force wsman:\localhost\Service\auth\Basic $True
PS C:\Users\Administrator>
```

And we have to install winrm in Ubuntu machine also. We have to install winrm through python following command is used install the winrm.

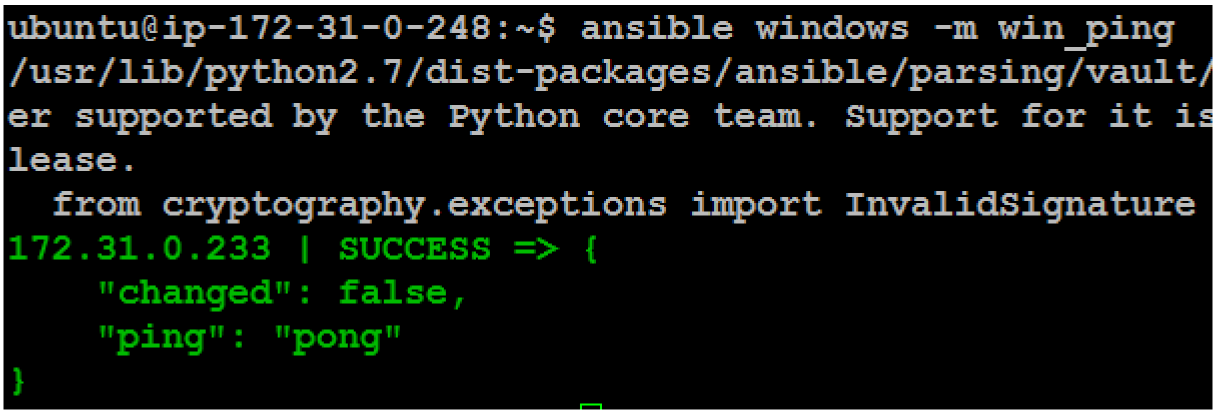
sudo apt-get install -y python3-winrm

After install winrm we have to configure the host file. In hosts file we have add the IP address. With IP address we have to configure few more command to access windows node.


```
[windows]
172.31.0.233

[windows:vars]
ansible_user = Administrator
ansible_password = 3.qVqwAcMIT.wYyXWCNru75YtUlDhAuQ
ansible_connection = winrm
ansible_winrm_server_cert_validation = ignore
ansible_winrm_transport = basic
ansible_port = 5985
```

Finally both control and mange node were ready to communicate.


```
ubuntu@ip-172-31-0-248:~$ ansible windows -m win_ping
/usr/lib/python2.7/dist-packages/ansible/parsing/vault/
er supported by the Python core team. Support for it is
lease.
  from cryptography.exceptions import InvalidSignature
172.31.0.233 | SUCCESS => {
    "changed": false,
    "ping": "pong"
}
```

Not logging Password

The 'NOT LOGGING PASSWORD' part is a built-in Ansible security feature. It means that the password can't be extracted from logs.

By default Ansible sends output about plays, tasks, and module arguments to your screen (STDOUT) on the control node. If you want to capture Ansible output in a log, you have three options:

- To save Ansible output in a single log on the control node, set the log_path configuration file setting. You may also want to set display_args_to_stdout, which helps to differentiate similar tasks by including variable values in the Ansible output.
- To save Ansible output in separate logs, one on each managed node, set the no_target_syslog and syslog_facility configuration file settings.
- To save Ansible output to a secure database, use Ansible Tower. Tower allows you to review history based on hosts, projects, and particular inventories over time, using graphs and/or a REST API.

Protecting sensitive data with no_log

If you save Ansible output to a log, you expose any secret data in your Ansible output, such as passwords and user names. To keep sensitive values out of your logs, mark tasks that expose them with the no_log: True attribute. However, the no_log attribute does not affect debugging output, so be careful not to debug playbooks in a production environment.