

Konfigurasi Basic Ansible

Menurut Wikipedia , *Ansible* is an open-source software provisioning, configuration management, and application-deployment tool.[2] It runs on many Unix-like systems, and can configure both Unix-like systems as well as Microsoft Windows. It includes its own declarative language to describe system configuration.

[https://en.wikipedia.org/wiki/Ansible_\(software\)](https://en.wikipedia.org/wiki/Ansible_(software))

1. Instalasi

- Requirement

For	OS	IP Addr
Ansible	Ubuntu 18	192.168.11.216
Host1	CentOS7	192.168.217.130
Host2	CentOS7	192.168.217.131

Install Ansible pada Ubuntu 18, berikut adalah langkah-langkahnya

Untuk mendapatkan versi terbaru dari Ansible untuk Ubuntu, Anda dapat menambahkan PPA (arsip paket pribadi) proyek ke sistem Anda. Sebelum melakukan ini, Anda harus terlebih dahulu memastikan bahwa Anda memiliki paket perangkat lunak-properti-umum yang diinstal. Perangkat lunak ini akan membuatnya lebih mudah untuk mengelola ini dan repositori perangkat lunak independen lainnya:

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

Kemudian tambahkan Ansible PPA dengan mengetik perintah berikut:

```
$ sudo apt-add-repository ppa:ansible/ansible
```

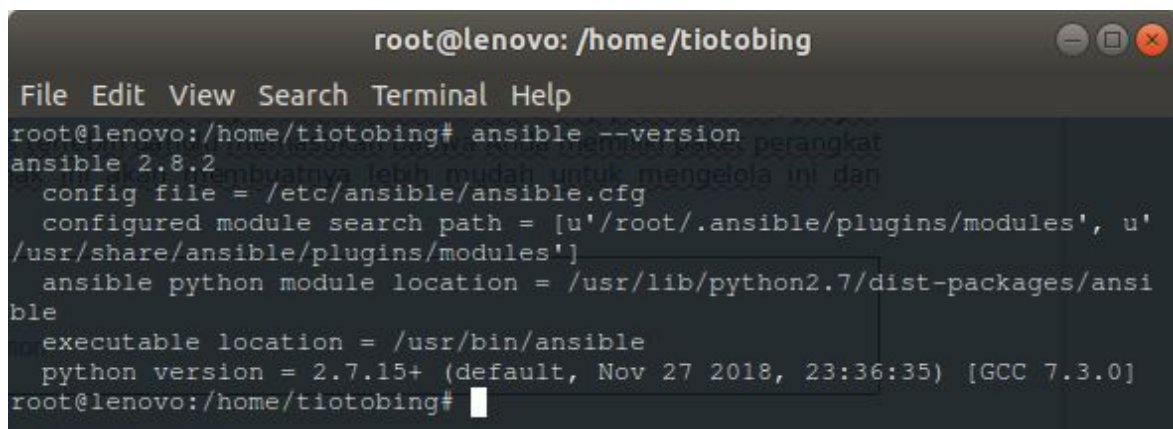
Selanjutnya, update sistem Anda sekali lagi sehingga mengetahui paket yang tersedia

```
$ sudo apt update
```

Setelah pembaruan ini, Anda dapat menginstal perangkat lunak Ansible:

```
$ sudo apt install ansible
```

Tunggu proses instalasi selesai.



```
root@lenovo: /home/tiotobing  
File Edit View Search Terminal Help  
root@lenovo:/home/tiotobing# ansible --version  
ansible 2.8.2  
  config file = /etc/ansible/ansible.cfg  
  configured module search path = [u'/root/.ansible/plugins/modules', u'/usr/share/ansible/plugins/modules']  
  ansible python module location = /usr/lib/python2.7/dist-packages/ansible  
  executable location = /usr/bin/ansible  
  python version = 2.7.15+ (default, Nov 27 2018, 23:36:35) [GCC 7.3.0]  
root@lenovo:/home/tiotobing#
```

2. Konfigurasi akses SSH untuk Ansible Host

Pertama buat sebuah public key yang akan digunakan untuk bisa berkoneksi dengan host nantinya.

```
$ sudo ssh-keygen
```

```

root@localhost:~
File Edit View Search Terminal Help

[root@localhost ~]# ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Passphrases do not match. Try again.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/.ssh/id_rsa.
Your public key has been saved in /root/.ssh/id_rsa.pub.
The key's fingerprint is:
SHA256:LTMUHgnvQ7ounKf5A8EnyLZHakVj1N3fEWKClzp9ZKg root@localhost.localdomain
The key's randomart image is:
+--[RSA 2048]-----+
| .+. .++o+ ..      |
|... .. oo+=+..    |
| o o   .++ + o     |
|= = .++ + o       |
|. +. S=           |
|. o .E=          |
| o o.            |
| +oo             |
| o.o.            |
| +----+         |
+---[SHA256]-----+
[root@localhost ~]#

```

Setelah terbentuk public key, selanjutnya adalah buka key tersebut

```
$ cat ~/.ssh/id_rsa.pub
```

Copy hasilnya kemudian buka terminal baru dan ketikkan command berikut

```
$ ssh user@ansible_server_IP
$ su -
$ nano ~/.ssh/authorized_keys
# lalu copy hasi id_rsa.pub tadi ke file authorized_keys
$ exit
```

3. Setting Up Ansible Hosts

Langkah ketiga adalah mendaftarkan hosts / client yang akan kita provisioning nantinya dengan automasi ansible.

Seperti diawal saya menggunakan 2 VM yang nantinya akan saya berinama host1 dan host2 sedangkan untuk server ansible saya beri nama main-Server

Berikut langkah-langkah menambahkan hosts pada Ansible

```
$ sudo nano /etc/ansible/hosts
```

Tambahkan hosts dan IP address.

```
[servers]
host1 ansible_host=192.168.217.130
host2 ansible_host=192.168.217.131
```

Saya menambahkan sebuah group dengan nama *servers* dan setiap host untuk IP yang telah ditentukan.

Save file

Coba test konfigurasi dengan

```
$ ansible -m ping all
```

maka hasilnya

```
Output
host1 | UNREACHABLE! => {
  "changed": false,
  "msg": "Failed to connect to the host via ssh.",
  "unreachable": true
}
...
```

Why ?

Ya karena pada basic nya ansible berkomunikasi base on ssh, jadi kalau ssh masih belum solve, kita tidak akan bisa ketahap selanjutnya, termasuk untuk Pink koneksi saja.

Setelah mendapatkan public key yang kita create tadi, kita dapat memberikan public key tersebut sebagai agent untuk client agar dapat berkomunikasi

Pada ansible server, lakukan command berikut

```
$ ssh-copy-id -i root@IP_add_Host1
$ ssh-copy-id -i root@IP_add_Host2

dst..(jika ada server lain yang akan ditambahkan)
```

Dan sebelumnya pada konfigurasi di `/etc/ansible/hosts` kita menyatukan ketiga server tersebut kedalam sebuah group bernama *servers*. Kita akan memberi permission dan akses untuk root. Untuk melakukan ini, kita akan membuat direktori dalam struktur konfigurasi Ansible yang disebut `group_vars`. Dalam folder ini, kita dapat membuat file berformat YAML untuk setiap grup yang ingin kita konfigurasi:

```
$ sudo mkdir /etc/ansible/group_vars
$ sudo nano /etc/ansible/group_vars/servers
```

file `/etc/ansible/group_vars/servers` kita isi dengan script berikut

```
---
```

```
ansible_user: root
```

save & close

4. Test

Ketikkan command dibawah ini untuk test ping untuk ketiga server

```
$ ansible -m ping all
```

```
root@lenovo:/home/tiotobing# ansible -m ping all
host1 | SUCCESS => {
  "ansible_facts": {
    "/etc/ansible/vars": {
      "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
  },
  "ansible_user": "root"
}
host2 | SUCCESS => {
  "ansible_facts": {
    "/etc/ansible/vars": {
      "discovered_interpreter_python": "/usr/bin/python"
    },
    "changed": false,
    "ping": "pong"
  },
  "ansible_user": "root"
}
```

Voila !!

Sekarang test menjalankan command uptime untuk host 1

```
root@lenovo: /home/tiotobing
File Edit View Search Terminal Help
root@lenovo:/home/tiotobing# ansible -m command -a "uptime" 'host1'
host1 | CHANGED | rc=0 >>
 03:59:00 up  3:01,  3 users,  load average: 0,00, 0,01, 0,05
```

-end-