NETWORK AUTOMATION WITH ANSIBLE

I PUTU HARIYADI

admin@iputuhariyadi.net

APA ITU ANSIBLE?

- Menurut situs Ansible, **Ansible** merupakan mesin otomatisasi Teknologi Informasi (TI) sederhana yang dapat mengotomatisasi *cloud provisioning*, manajemen konfigurasi, penerapan aplikasi, *intra-service orchestration* dan kebutuhan TI lainnya.
- Menurut situs Edureka, keuntungan menggunakan *Ansible* antara lain:

1. **Simple**, menggunakan sintak penulisan yang ditulis menggunakan **YAML** yang disebut dengan **playbook**.
2. **Agentless**, tidak diperlukan **agent** atau *software* khusus untuk diinstalasi pada host yang diautomasi.
3. **Powerful & Flexible**, memiliki banyak modul untuk manajemen infrastruktur, jaringan, sistem operasi dan layanan.
4. **Efficient**, tidak ada *software* yang diperlukan untuk instalasi pada server sehingga lebih banyak sumber daya yang dapat digunakan oleh aplikasi.

- *Ansible* dapat melakukan hal-hal berikut:

1. **Provisioning**
   Ansible memastikan paket-paket yang dibutuhkan akan diunduh dan diinstalasi sehingga aplikasi dapat diterapkan.

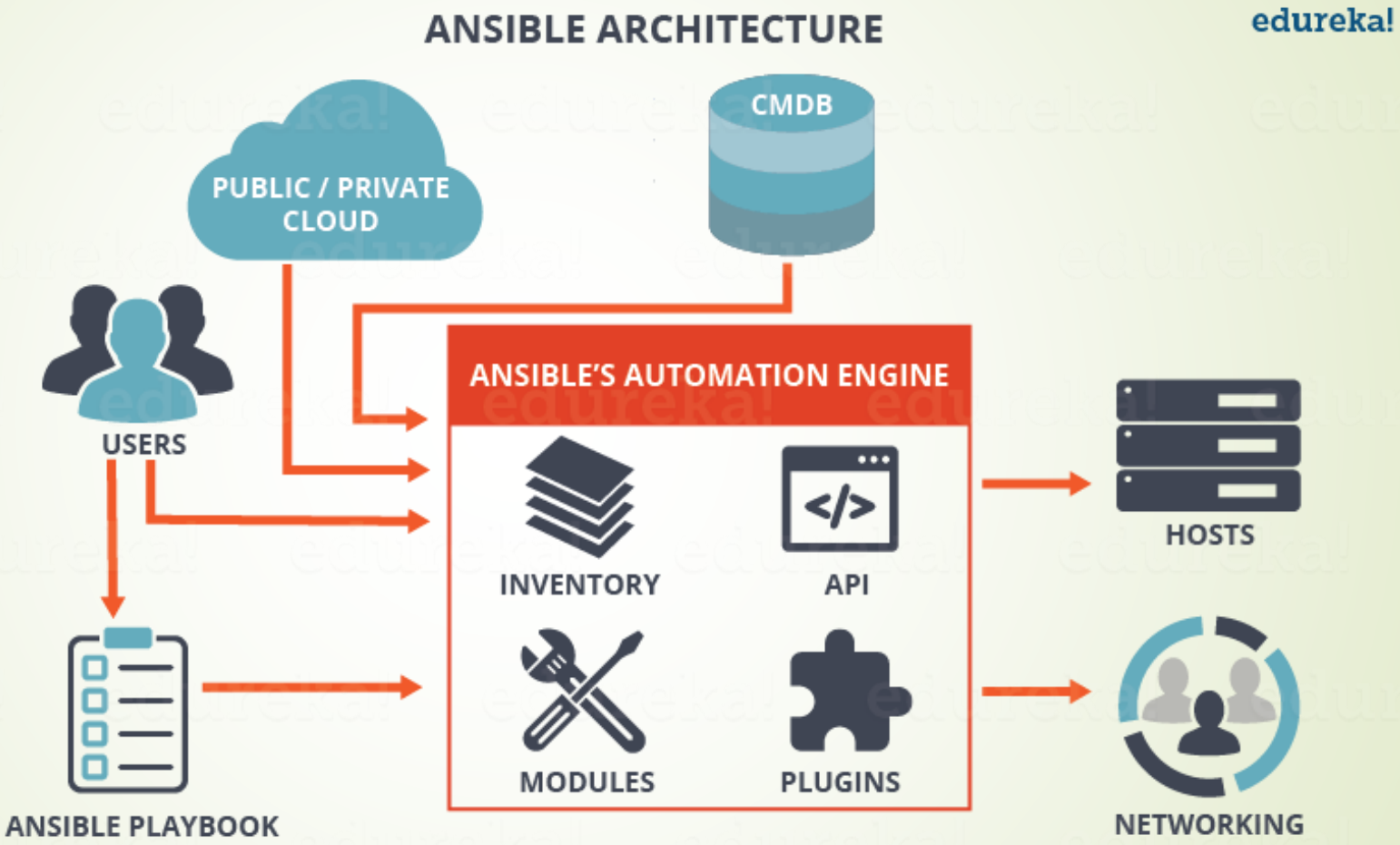
APA ITU ANSIBLE?

2. **Configuration Management**
   Menetapkan dan mempertahankan konsistensi dari kinerja produk dengan mencatat dan memperbaharui informasi lengkap yang menjelaskan perangkat keras dan lunak perusahaan. Seperti versi dan pembaharuan yang telah diterapkan pada paket *software* yang terinstal dan lokasi serta alamat jaringan dari perangkat keras.
3. **Application Deployment**
   Ansible dapat memanajemen secara efektif keseluruhan *life cycle* dari aplikasi, mulai dari *development* sampai produksi.
4. **Security and Compliance**
   Kebijakan keamanan dapat didefinisikan pada *Ansible* sehingga proses pemindaian dan pemulihan kebijakan ke lokasi dapat diintegrasikan ke dalam proses secara otomatis.
5. **Orchestration**
   Ansible menyediakan *orchestration* dalam arti menyelaraskan permintaan bisnis dengan aplikasi, data, dan infrastruktur. Ini mendefinisikan kebijakan dan tingkat layanan melalui alur kerja otomatis, penyediaan, dan manajemen perubahan sehingga menciptakan aplikasi selaras dengan infrastruktur yang dapat ditingkatkan dari atas ke bawah berdasarkan kebutuhan setiap aplikasi.

TERMINOLOGI PADA ANSIBLE (Sumber: Edureka)

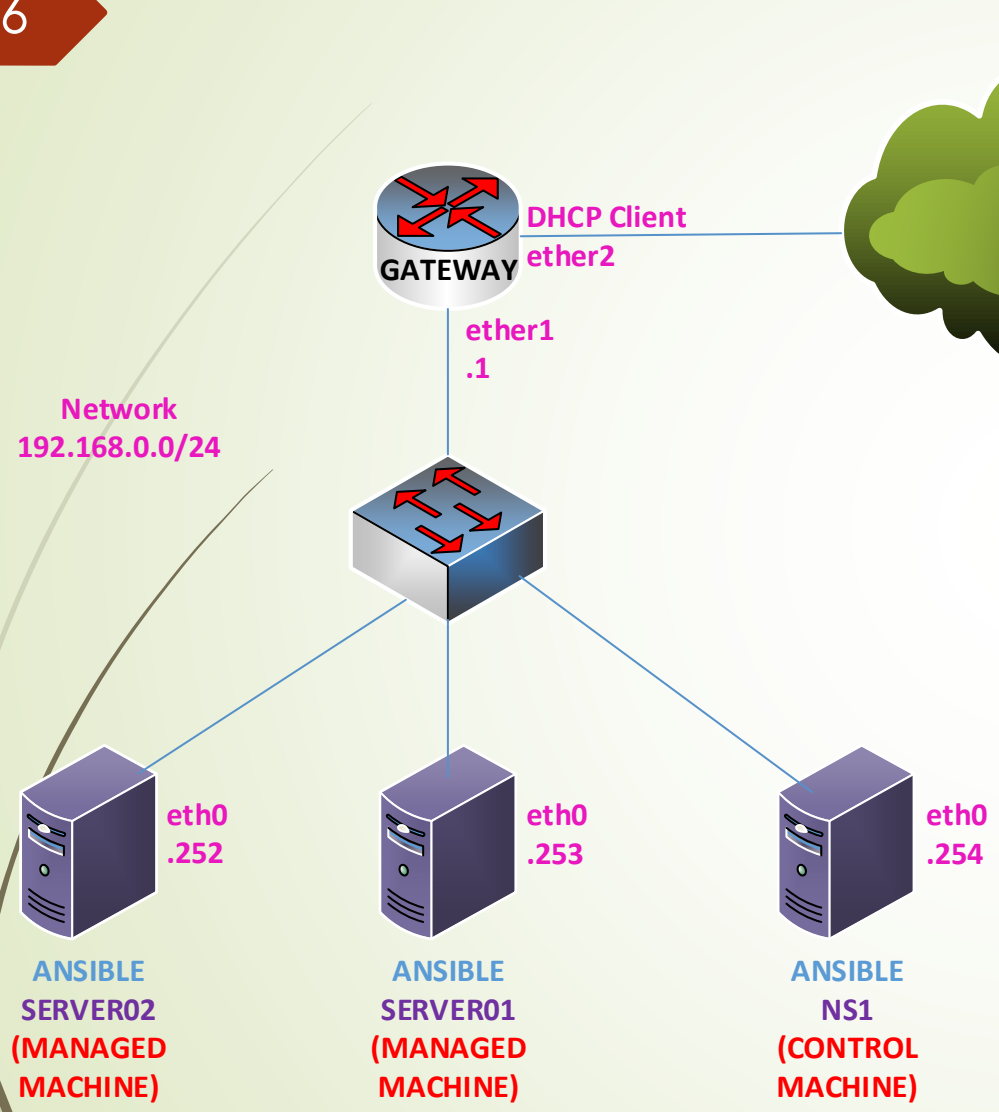
- **Controller Machine:** mesin dimana *ansible* diinstalasi dan bertanggungjawab untuk menjalankan *provisioning* pada server yang dikelola.
- **Inventory:** file inisialisasi yang memuat informasi tentang server yang dikelola.
- **Playbook**: Titik masuk untuk *Ansible provisioning*, dimana automasi didefinisikan melalui tugas (tasks) menggunakan format YAML..
- **Task:** Blok yang mendefinisikan satu prosedur untuk dieksekusi, sebagai contoh instalasi *package* tertentu.
- **Module:** Modul merupakan abstraksi dari tugas sistem, seperti berkaitan dengan package atau membuat dan mengubah file. *Ansible* memiliki banyak modul built-in, namun dapat juga dibuat modul custom.
- **Role:** Cara yang telah ditentukan sebelumnya untuk mengatur playbook dan file lainnya untuk memfasilitasi berbagi pakai dan menggunakan kembali bagian dari *provisioning*.
- **Play:** *provisioning* yang dieksekusi mulai dari awal sampai akhir disebut dengan *play*. Dengan kata lain, eksekusi dari *playbook* disebut dengan *play*.
- **Facts:** variable global yang memuat informasi tentang sistem, seperti interface jaringan atau sistem operasi.
- **Handlers:** Digunakan untuk memicu perubahan status dari service, seperti me-*restart* atau menghentikan *service*.

ANSIBLE ARCHITECTURE



Sumber: edureka (https://cdn.edureka.co/blog/wp-content/uploads/2016/11/Ansible-Architechture-What-Is-Ansible-Edureka.png)

RANCANGAN JARINGAN UJICOBA



Rancangan jaringan ujicoba disimulasikan menggunakan **VMWare Workstation 14 Pro**. Terdapat 4 (empat) **Virtual Machine (VM)** yang dibuat yaitu:

- **1 VM Router Gateway** menggunakan **MikroTik Cloud Hosted Router (CHR).** *Interface* **ether1** di Mikrotik Gateway menggunakan **mode network connection host-only (VMnet1)**. Sedangkan untuk *interface* **ether2** menggunakan mode **NAT**.
- **1 VM Ansible Control Machine** dan **2 VM Managed Machine** menggunakan **Linux CentOS 6.8.** Keseluruhan interface jaringan **eth0** baik pada **Control Machine** maupun **Managed Machine** menggunakan **mode network connection host-only (VMnet1)**.

INSTALASI & KONFIGURASI MIKROTIK CHR SEBAGAI ROUTER GATEWAY UNTUK KONEKSI INTERNET

- Unduh **Mikrotik CHR** dari situs Mikrotik pada alamat https://mikrotik.com/download.
- Lakukan instalasi **Mikrotik CHR** sebagai **Virtual Machine (VM)** pada **VMWare Workstation 14 Pro**.
- Akses *VM Mikrotik CHR* pada *VMWare Workstation* maka akan tampil inputan **MikroTik Login**, seperti berikut:

```
MikroTik 6.39.1 (stable)
MikroTik Login: admin
Password:
```

Masukkan "**admin**" dan tekan **Enter**. Selanjutnya tampil inputan **Password:** Tekan **Enter** karena secara default user "*admin*" tidak memiliki *password* (**tanpa sandi)**.

- Menampilkan informasi **interface** yang dimiliki oleh *MikroTik*.

```
[admin@MikroTik] > interface print
Flags: D - dynamic, X - disabled, R - running, S - slave
 #     NAME                                TYPE       ACTUAL-MTU L2MTU
 0  R  ether1                              ether            1500
 1  R  ether2                              ether            1500
```

- Mengatur **DHCP Client** pada interface **ether2**.

```
[admin@MikroTik] > ip dhcp-client add interface=ether2 disabled=no
```

KONFIGURASI PADA MIKROTIK CHR SEBAGAI ROUTER GATEWAY UNTUK KONEKSI INTERNET

- Memverifikasi pengaturan **DHCP Client** pada **interface ether2**.

```
[admin@MikroTik] > ip dhcp-client print
Flags: X - disabled, I - invalid
 #    INTERFACE          USE ADD-DEFAULT-ROUTE STATUS       ADDRESS
 0    ether2             yes yes               bound        192.168.126.132/24
```

- Mengatur pengalamatan **IP** pada **interface ether1**

```
[admin@MikroTik] > ip address add address=192.168.0.1/24 interface=ether1
```

- Menampilkan informasi pengalamatan **IP** pada **interface**.

```
[admin@MikroTik] > ip address print
Flags: X - disabled, I - invalid, D - dynamic
 #    ADDRESS            NETWORK         INTERFACE
 0 D  192.168.126.132/24 192.168.126.0   ether2
 1    192.168.0.1/24     192.168.0.0     ether1
```

- Mengatur agar **router gateway** sebagai **Server DNS** untuk jaringan lokal.

```
[admin@MikroTik] > ip dns set allow-remote-requests=yes
```

KONFIGURASI PADA MIKROTIK CHR SEBAGAI ROUTER GATEWAY UNTUK KONEKSI INTERNET

- Mengatur **Internet Connection Sharing (ICS)**

```
[admin@MikroTik] > ip firewall nat add chain=srcnat out-interface=ether2 action=
masquerade
```

- Menampilkan informasi pengaturan **ICS**.

```
[admin@MikroTik] > ip firewall nat print
Flags: X - disabled, I - invalid, D - dynamic
 0    chain=srcnat action=masquerade out-interface=ether2
```

- Memverifikasi koneksi ke Internet

```
[admin@MikroTik] > ping google.com
  SEQ HOST                                     SIZE TTL TIME  STATUS
    0 118.98.109.217                             56 128 3ms
    1 118.98.109.217                             56 128 2ms
    sent=2 received=2 packet-loss=0% min-rtt=2ms avg-rtt=2ms max-rtt=3ms
```

TAHAPAN INSTALASI DAN KONFIGURASI CENTOS 6.8

- Lakukan instalasi **Linux CentOS 6.8** sebagai **Virtual Machine (VM)** pada **VMWare Workstation** yang difungsikan sebagai **Ansible Control Machine**. Langkah-langkahnya dapat mengikuti tutorial "Instalasi Linux CentOS 6.4 Pada VMWare Workstation 10". Link situs terdapat pada bagian **Referensi** di halaman terakhir.
- Setelah selesai instalasi, lakukan penonaktifan **SELinux** dan **service IPTables**.
- Penonaktifan fitur **SELinux** dilakukan dengan mengubah nilai dari parameter **SELINUX=enforcing** menjadi **SELINUX=disabled** pada file **/etc/selinux/config**.

```
[root@ns1 ~]# nano /etc/selinux/config
```

```
  GNU nano 2.0.9                                        File: /etc

# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
#     disabled - No SELinux policy is loaded.
SELINUX=disabled
```

Simpan perubahan dengan menekan **CTRL+O** dan **Enter**. Tekan **CTRL+X** untuk keluar dari *editor nano*.

- Menonaktifkan fitur **IPTables**.

```
[root@ns1 ~]# chkconfig iptables off
```

TAHAPAN INSTALASI DAN KONFIGURASI CENTOS 6.8

- Memverifikasi hasil penonaktifan fitur **IPTables**.

```
[root@ns1 ~]# chkconfig --list iptables
iptables        0:off   1:off   2:off   3:off   4:off   5:off   6:off
```

- Men-**shutdown VM Ansible Control Machine** untuk proses **Clone**.

```
[root@ns1 ~]# poweroff
```

- **Clone VM Ansible Control Machine** dengan jenis **linked clone** sebanyak **2 (dua) VM** yang difungsikan sebagai **Ansible Managed Machine Server01 & Server02**.
- Lakukan pengaturan pengalamatan IP pada **interface eth0** pada masing-masing **Machine** dengan ketentuan sebagai berikut:

1. **Control Machine: 192.168.0.254/24**
2. **Managed Machine Server01: 192.168.0.253/24**
3. **Managed Machine Server02: 192.168.0.252/24**

Alamat IP **Default gateway** dan **Server DNS** untuk seluruh machine adalah **192.168.0.1**. **Perhatian: khusus untuk Managed Machine lakukan perbaikan MAC-Address yang tidak konsisten pada interface eth0 sebagai dampak proses Clone VM**. Perbaikan dapat dilakukan dengan mengikuti tutorial "Troubleshoot Networking Pada Cloned Linux CentOS 6.8 Virtual Machine di VMWare Workstation 12".

- Verifikasi menggunakan perintah **ping** ke salah satu situs di Internet untuk memastikan seluruh **Machine** dapat terkoneksi ke Internet.

MENGATUR PENGALAMATAN IP PADA CONTROL MACHINE

- Mengatur pengalamatan IP pada *interface* **eth0** menggunakan **editor nano**.

```
[root@ns1 ~]# nano /etc/sysconfig/network-scripts/ifcfg-eth0
```

```
  GNU nano 2.0.9

DEVICE=eth0
HWADDR=00:0C:29:44:15:88
TYPE=Ethernet
#UUID=3186d39e-4332-41ec-a6c4-da82bdf98834
ONBOOT=yes
NM_CONTROLLED=no
BOOTPROTO=static
IPADDR=192.168.0.254
PREFIX=24
GATEWAY=192.168.0.1
DNS1=192.168.0.1
```

Simpan perubahan dengan menekan **CTRL+O** dan **Enter**. Tekan **CTRL+X** untuk keluar dari *editor nano*.

- Mengaktifkan perubahan pengaturan pengalamatan IP pada *interface* **eth0**.

 # service network restart

VERIFIKASI PADA CONTROL MACHINE

- Memverifikasi pengalamatan IP pada *interface* **eth0**.

```
[root@ns1 ~]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0C:29:44:15:88
          inet addr:192.168.0.254  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fe44:1588/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:1811 errors:0 dropped:0 overruns:0 frame:0
          TX packets:14 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:113490 (110.8 KiB)  TX bytes:900 (900.0 b)
          Interrupt:19 Base address:0x2024
```

- Memverifikasi koneksi ke salah satu situs di Internet sebagai contoh **google.com**.

```
[root@ns1 ~]# ping google.com
PING google.com (74.125.24.138) 56(84) bytes of data.
64 bytes from 74.125.24.138: icmp_seq=1 ttl=128 time=75.4 ms
64 bytes from 74.125.24.138: icmp_seq=2 ttl=128 time=80.2 ms
^C
--- google.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1601ms
rtt min/avg/max/mdev = 75.490/77.851/80.212/2.361 ms
```

Terlihat koneksi ke Internet berhasil dilakukan. Tekan **CTRL+C** untuk menghentikan **ping**.

MENGATUR PENGALAMATAN IP PADA MANAGED MACHINE SERVER01

- Mengatur pengalamatan IP pada *interface* **eth0** menggunakan **editor nano**.

```
[root@server01 ~]# nano /etc/sysconfig/network-scripts/ifcfg-eth0
```

```
  GNU nano 2.0.9

DEVICE=eth0
HWADDR=00:0C:29:A5:DC:7C
TYPE=Ethernet
#UUID=3186d39e-4332-41ec-a6c4-da82bdf98834
ONBOOT=yes
NM_CONTROLLED=no
BOOTPROTO=static
IPADDR=192.168.0.253
PREFIX=24
GATEWAY=192.168.0.1
DNS1=192.168.0.1
```

Simpan perubahan dengan menekan **CTRL+O** dan **Enter**. Tekan **CTRL+X** untuk keluar dari *editor nano*.

- Mengaktifkan perubahan pengaturan pengalamatan IP pada *interface* **eth0**.

 # service network restart

VERIFIKASI PADA MANAGED MACHINE SERVER01

- Memverifikasi pengalamatan IP pada *interface* **eth0**.

```
[root@server01 ~]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0C:29:A5:DC:7C
          inet addr:192.168.0.253  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fea5:dc7c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2485 errors:0 dropped:0 overruns:0 frame:0
          TX packets:14 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:155601 (151.9 KiB)  TX bytes:900 (900.0 b)
          Interrupt:19 Base address:0x2024
```

- Memverifikasi koneksi ke salah satu situs di Internet sebagai contoh **ansible.com**.

```
[root@server01 ~]# ping ansible.com
PING ansible.com (104.24.17.59) 56(84) bytes of data.
64 bytes from 104.24.17.59: icmp_seq=1 ttl=128 time=80.0 ms
64 bytes from 104.24.17.59: icmp_seq=2 ttl=128 time=60.2 ms
^C
--- ansible.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1915ms
rtt min/avg/max/mdev = 60.294/70.182/80.070/9.888 ms
```

Terlihat koneksi ke Internet berhasil dilakukan. Tekan **CTRL+C** untuk menghentikan **ping**.

MENGATUR PENGALAMATAN IP PADA MANAGED MACHINE SERVER02

- Mengatur pengalamatan IP pada *interface* **eth0** menggunakan **editor nano**.

```
[root@server02 ~]# nano /etc/sysconfig/network-scripts/ifcfg-eth0
```

```
  GNU nano 2.0.9

DEVICE=eth0
HWADDR=00:0C:29:F3:AA:0E
TYPE=Ethernet
#UUID=3186d39e-4332-41ec-a6c4-da82bdf98834
ONBOOT=yes
NM_CONTROLLED=no
BOOTPROTO=static
IPADDR=192.168.0.252
PREFIX=24
GATEWAY=192.168.0.1
DNS1=192.168.0.1
```

Simpan perubahan dengan menekan **CTRL+O** dan **Enter**. Tekan **CTRL+X** untuk keluar dari *editor nano*.

- Mengaktifkan perubahan pengaturan pengalamatan IP pada *interface* **eth0**.

 # service network restart

VERIFIKASI PADA MANAGED MACHINE SERVER02

- Memverifikasi pengalamatan IP pada *interface* **eth0**.

```
[root@server02 ~]# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:0C:29:F3:AA:0E
          inet addr:192.168.0.252  Bcast:192.168.0.255  Mask:255.255.255.0
          inet6 addr: fe80::20c:29ff:fef3:aa0e/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:2676 errors:0 dropped:0 overruns:0 frame:0
          TX packets:14 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:167705 (163.7 KiB)  TX bytes:900 (900.0 b)
          Interrupt:19 Base address:0x2024
```

- Memverifikasi koneksi ke salah satu situs di *Internet* sebagai contoh **yahoo.com**.

```
[root@server02 ~]# ping yahoo.com
PING yahoo.com (98.138.252.38) 56(84) bytes of data.
64 bytes from media-router-fp1.prod.media.vip.ne1.yahoo.com (98.138.252.38): icmp_seq=1 ttl=128 time=344 ms
64 bytes from media-router-fp1.prod.media.vip.ne1.yahoo.com (98.138.252.38): icmp_seq=2 ttl=128 time=319 ms
^C
--- yahoo.com ping statistics ---
3 packets transmitted, 2 received, 33% packet loss, time 2028ms
rtt min/avg/max/mdev = 319.584/332.211/344.838/12.627 ms
```

Terlihat koneksi ke Internet berhasil dilakukan. Tekan **CTRL+C** untuk menghentikan **ping**.

INSTALASI ANSIBLE PADA CONTROL MACHINE

1. Menambahkan repository **Extra Packages for Enterprise Linux (EPEL)** karena *Ansible* tidak tersedia secara *default* pada **Yum Repositories**.

   # rpm -ivh http://dl.fedoraproject.org/pub/epel/6/i386/epel-release-6-8.noarch.rpm

2. Menginstalasi *Ansible*.

   # yum install ansible -y

3. Memverifikasi hasil instalasi *Ansible*.

   # ansible --version

```
ansible 2.4.1.0
  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/python2.6/site-packages/ansible
  executable location = /usr/bin/ansible
  python version = 2.6.6 (r266:84292, Jul 23 2015, 14:41:34) [GCC 4.4.7 20120313 (Red Hat 4.4.7-11)]
```

MENGATUR PUBLIC KEY AUTHENTICATION UNTUK SSH

1. Membuat **key pair** menggunakan utilitas **ssh-keygen** pada **control machine**.
   # ssh-keygen

```
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa):
```

   Tekan **Enter** pada inputan **Enter file in which to save the key (/root/.ssh/id_rsa):** untuk menggunakan lokasi dan nama file *default* penyimpanan *key* yaitu **/root/.ssh/id_rsa**.

```
Created directory '/root/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
```

   Tekan **Enter** pada inputan "**Enter passphrase (empty for no passphrase):**" dan "**Enter same passphrase again:**" yang tampil untuk apabila ingin mengosongkan **passphrase**. Tampil proses pembuatan *key*. Tunggu hingga proses selesai dilakukan.

2. Menyalinkan **public key** dari **control machine** ke **managed machines** menggunakan utilitas **ssh-copy-id**.
   # ssh-copy-id -i root@192.168.0.253

MENGATUR PUBLIC KEY AUTHENTICATION UNTUK SSH

- Ketik **yes** dan tekan **Enter** pada pesan konfirmasi **Are you sure you want to continue connection (yes/no)?.**

```
The authenticity of host '192.168.0.253 (192.168.0.253)' can't be established.
RSA key fingerprint is d1:ab:a4:e0:8b:64:42:19:cc:e4:4c:06:76:43:06:64.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '192.168.0.253' (RSA) to the list of known hosts.
root@192.168.0.253's password:
Now try logging into the machine, with "ssh 'root@192.168.0.253'", and check in:

  .ssh/authorized_keys

to make sure we haven't added extra keys that you weren't expecting.
```

Selanjutnya tampil inputan **password** untuk *user* **root** untuk **managed machine server01** dengan alamat IP **192.168.0.253** yaitu "**12345678**".

Dengan cara yang sama, lakukan untuk **managed machine server02** dengan alamat IP **192.168.0.252**

 # ssh-copy-id -i root@192.168.0.252

MENGATUR PUBLIC KEY AUTHENTICATION UNTUK SSH

3. Memverifikasi hasil pengaturan *public key authentication* untuk *SSH* dengan cara mengakses **managed machines** menggunakan **SSH Client**. Sebagai contoh SSH ke **server01:**
   # ssh root@192.168.0.253
   Anda dapat mengakses ke server01 tanpa password. Untuk keluar dari **server01**, ketik "**exit**" dan tekan **Enter**.

   Dengan cara yang sama, lakukan ujicoba SSH ke **server02**:
   # ssh root@192.168.0.252

MENGATUR ANSIBLE INVENTORY

- **Inventory** merupakan file inisialisasi yang digunakan oleh *Ansible* untuk mendaftarkan dan mengelompokkan mesin atau host yang akan dikelola.
- Lokasi dari file **inventory** secara *default* adalah "**/etc/ansible/hosts**".
- Menambahkan **managed machines** ke file *inventory* menggunakan editor **nano**.
  # nano /etc/ansible/hosts
  [servers]
  server01 ansible_host=192.168.0.253
  server02 ansible_host=192.168.0.252

  Simpan perubahan dengan menekan **CTRL+O** dan tekan **Enter**. Keluar dari editor *nano* dengan menekan **CTRL+X**.
- Memverifikasi hasil penambahan *inventory*.
  # cat /etc/ansible/hosts

CARA MENJALANKAN ANSIBLE

Terdapat 3 (tiga) cara untuk menjalankan *Ansible* yaitu:

1. **Ad-Hoc**
   ansible <inventory> -m
   Sebagai contoh eksekusi *ansible* untuk *group inventory* dengan nama "**servers**" dan modul **ping**.
   # ansible servers –m ping

2. **Playbooks**
   ansible-playbook filename.yml
   Sebagai contoh eksekusi *ansible playbook* dengan nama file **vsftpd.yml**.
   # ansible-playbook vsftpd.yml

3. **Automation Framework**
   Menurut situs Ansible, **Ansible Tower** merupakan solusi berbasis web yang membuat *Ansible* lebih mudah digunakan untuk tim TI dan dirancang untuk menjadi pusat semua tugas otomasi serta memungkinkan untuk mengontrol akses pengguna. **Inventory** dapat dikelola secara grafis atau disinkronisasikan dengan beragam sumber **Cloud**. **Tower** juga mencatat (*log*) semua pekerjaan, terintegrasi dengan **LDAP** dan memiliki **API REST**. Tersedia pula tool **Command Line** untuk memudahkan integrasi dengan **Jenkins** juga.

ANSIBLE AD-HOC (1)

- Memverifikasi koneksi ke seluruh **managed machines** untuk *group* "**servers**" menggunakan modul **ping**.

```
[root@ns1 ~]# ansible servers -m ping
server02 | SUCCESS => {
    "changed": false,
    "failed": false,
    "ping": "pong"
}
server01 | SUCCESS => {
    "changed": false,
    "failed": false,
    "ping": "pong"
}
```

ANSIBLE AD-HOC (2)

- Memverifikasi koneksi ke **managed machine** tertentu, sebagai contoh ke **server01** menggunakan modul **ping**.

```
[root@ns1 ~]# ansible server01 -m ping
server01 | SUCCESS => {
    "changed": false,
    "failed": false,
    "ping": "pong"
}
```

- Memverifikasi koneksi ke **managed machine** tertentu, sebagai contoh ke **server02** menggunakan modul **ping**.

```
[root@ns1 ~]# ansible server02 -m ping
server02 | SUCCESS => {
    "changed": false,
    "failed": false,
    "ping": "pong"
}
```

ANSIBLE AD-HOC (3)

- Menampilkan informasi **system uptime** dari *managed machine* dengan nama group "**servers**" menggunakan modul **command**.

```
[root@ns1 ~]# ansible servers -m command -a "uptime"
server02 | SUCCESS | rc=0 >>
 10:06:45 up  3:46,  2 users,  load average: 0.04, 0.03, 0.09

server01 | SUCCESS | rc=0 >>
 10:06:52 up  3:46,  2 users,  load average: 0.02, 0.05, 0.09
```

- Menampilkan informasi ukuran memori baik yang **free** maupun **used** dalam satuan **megabytes** dari *managed machine* group "**servers**" menggunakan modul **command**.

```
[root@ns1 ~]# ansible servers -m command -a "free -mt"
server02 | SUCCESS | rc=0 >>
             total       used       free     shared    buffers     cached
Mem:          1006        676        329          2         69        424
-/+ buffers/cache:        182        823
Swap:         1023          0       1023
Total:        2030        676       1353

server01 | SUCCESS | rc=0 >>
             total       used       free     shared    buffers     cached
Mem:          1006        677        328          2         69        424
-/+ buffers/cache:        183        822
Swap:         1023          0       1023
Total:        2030        677       1352
```

ANSIBLE AD-HOC (4)

- Melakukan **reboot** pada *managed machine* pada group "**servers**" dengan menggunakan modul **shell**.

```
[root@ns1 ~]# ansible servers -m shell -a "reboot"
server02 | SUCCESS | rc=0 >>


server01 | SUCCESS | rc=0 >>
```

- Memverifikasi apakah paket **ftp** telah terinstalasi pada *managed machine* pada group "**servers**" dengan menggunakan modul **yum.**

```
[root@ns1 ~]# ansible servers -m yum -a "list=ftp"
server01 | SUCCESS => {
    "changed": false,
    "failed": false,
    "results": [
        {
            "arch": "i686",
            "envra": "0:ftp-0.17-54.el6.i686",
            "epoch": "0",
            "name": "ftp",
            "release": "54.el6",
            "repo": "base",
            "version": "0.17",
            "yumstate": "available"
        }
    ]
}
```

Output **yumstate: available** berarti paket **ftp** belum terinstalasi pada **server01**.

ANSIBLE AD-HOC (4)

```
server02 | SUCCESS => {
    "changed": false,
    "failed": false,
    "results": [
        {
            "arch": "i686",
            "envra": "0:ftp-0.17-54.el6.i686",
            "epoch": "0",
            "name": "ftp",
            "release": "54.el6",
            "repo": "base",
            "version": "0.17",
            "yumstate": "available"
        }
    ]
}
```

Output **yumstate: available** berarti paket ftp belum terinstalasi pada **server02**.

- Menginstalasi paket **ftp** pada *managed machine* pada group "**servers**" dengan menggunakan modul **yum.**

```
[root@ns1 ~]# ansible servers -m yum -a "name=ftp state=installed"
```

Setelah proses instalasi selesai dilakukan, lakukan verifikasi kembali menggunakan:

```
[root@ns1 ~]# ansible servers -m yum -a "list=ftp"
```

Output **yumstate: installed** berarti paket **ftp** telah terinstalasi baik pada **server01** maupun **server02**.

ANSIBLE PLAYBOOK (1)

- Membuat file *playbook* menggunakan editor **nano** dengan ekstensi "**yml**" untuk menginstalasi paket vsftpd pada *managed machine* pada group "**servers**" dan menjalankan serta mengaktifkan *service vsftpd* ketika booting awal *Linux CentOS*.

 # nano vsftpd.yml

```
---
- hosts: servers
  remote_user: root
  gather_facts: false
  tasks:
   - name: install vsftpd package
     yum: name=vsftpd state=present

   - name: action to start and enable vsftpd
     service: name=vsftpd state=started enabled=yes
```

- Simpan perubahan dengan menekan **CTRL+O** dan **Enter**. Tekan **CTRL+X** untuk keluar dari editor *nano*.

ANSIBLE PLAYBOOK (2)

- *Ansible* **Check Mode (Dry Run)** digunakan sebagai simulasi.

```
[root@ns1 ~]# ansible-playbook vsftpd.yml --check

PLAY [servers] *****************************************************************

TASK [install vsftpd package] **************************************************
changed: [server02]
changed: [server01]

TASK [action to start and enable vsftpd] ***************************************
changed: [server02]
changed: [server01]

PLAY RECAP *********************************************************************
server01                   : ok=2    changed=2    unreachable=0    failed=0
server02                   : ok=2    changed=2    unreachable=0    failed=0
```

ANSIBLE PLAYBOOK (3)

- Mengeksekusi *playbook* file "**vsftpd.yml**" menggunakan **ansible-playbook**.

```
[root@ns1 ~]# ansible-playbook vsftpd.yml

PLAY [servers] *****************************************************************

TASK [install vsftpd package] **************************************************
changed: [server02]
changed: [server01]

TASK [action to start and enable vsftpd] ***************************************
changed: [server01]
changed: [server02]

PLAY RECAP *********************************************************************
server01                   : ok=2    changed=2    unreachable=0    failed=0
server02                   : ok=2    changed=2    unreachable=0    failed=0
```

ANSIBLE PLAYBOOK (4)

- Memverifikasi **paket vsftpd** telah terinstal.

```
[root@ns1 ~]# ansible servers -m yum -a "list=vsftpd"
```

Output **yumstate: installed** berarti paket **vsftpd** telah terinstalasi baik pada **server01** maupun **server02**.

- Memverifikasi status **service vsftpd** apakah telah berjalan.

```
[root@ns1 ~]# ansible servers -m command -a "service vsftpd status"
 [WARNING]: Consider using service module rather than running service

server01 | SUCCESS | rc=0 >>
vsftpd (pid 2336) is running...

server02 | SUCCESS | rc=0 >>
vsftpd (pid 2336) is running...
```

- Memverifikasi **service vsftpd** telah di **enable**.

```
[root@ns1 ~]# ansible servers -m command -a "chkconfig --list vsftpd"
server02 | SUCCESS | rc=0 >>
vsftpd          0:off   1:off   2:on    3:on    4:on    5:on    6:off

server01 | SUCCESS | rc=0 >>
vsftpd          0:off   1:off   2:on    3:on    4:on    5:on    6:off
```

ADA PERTANYAAN?

TERIMAKASIH

REFERENSI

- Ansible Documentation, http://docs.ansible.com/
- Edureka Website, https://www.edureka.co/
- Instalasi Linux CentOS 6.4 Pada VMWare Workstation 10, https://www.slideshare.net/putuhariyadi/instalasi-centos-64-pada-vmware-workstation-10
- I Putu Hariyadi Website, Troubleshoot Networking Pada Cloned Linux CentOS 6.8 Virtual Machine di VMWare Workstation 12, https://iputuhariyadi.net/2016/10/16/troubleshoot-networking-pada-cloned-linux-centos-6-8-virtual-machine-di-vmware-workstation-12/