

Automation Ansible

Detail Materi

Indikator :
Dapat melakukan automasi dan bekerja
menggunakan Ansible

Memahami Konsep Dasar Ansible

Memahami Cara Kerja Ansible

Memahami Komponen Pada Ansible

Setting dan Konfigurasi Ansible

Modul Sekolah DevOps Cilsy

Hak Cipta © 2019 **PT. Cilsy Fiolution Indonesia**

Hak Cipta dilindungi Undang-Undang. Dilarang memperbanyak atau memindahkan sebagian atau seluruh isi buku ini dalam bentuk apapun, baik secara elektronik maupun mekanis, termasuk mecopy, merekam atau dengan sistem penyimpanan lainnya, tanpa izin tertulis dari Penulis dan Penerbit.

Penulis : Adi Saputra, Irfan Herfiandana & Tresna Widiyaman

Editor: Taufik Maulana, Rizal Rahman & Tresna Widiyaman

Revisi Batch 9

Penerbit : **PT. Cilsy Fiolution Indonesia**

Web Site : <https://cilsyfiolution.com> , <https://devops.cilsy.id>

Sanksi Pelanggaran Pasal 113 Undang-undang Nomor 28 Tahun 2014 tentang Hak Cipta

1. Setiap orang yang dengan tanpa hak melakukan pelanggaran hak ekonomi sebagaimana dimaksud dalam pasal 9 ayat (1) huruf i untuk penggunaan secara komersial dipidana dengan pidana penjara paling lama 1 (satu) tahun dan atau pidana denda paling banyak Rp100.000.000,00 (seratus juta rupiah).
2. Setiap orang yang dengan tanpa hak dan atau tanpa izin pencipta atau pemegang hak cipta melakukan pelanggaran hak ekonomi pencipta sebagaimana dimaksud dalam pasal 9 ayat (1) huruf c, huruf d, huruf f, dan atau huruf h, untuk penggunaan secara komersial dipidana dengan pidana penjara paling lama 3 (tiga) tahun dan atau pidana denda paling banyak Rp500.000.000,00 (lima ratus juta rupiah)
3. Setiap orang yang dengan tanpa hak dan atau tanpa izin pencipta atau pemegang hak melakukan pelanggaran hak ekonomi pencipta sebagaimana dimaksud dalam pasal 9 ayat (1) huruf a, huruf b, huruf e, dan atau huruf g, untuk penggunaan secara komersial dipidana dengan pidana penjara paling lama 4 (empat) tahun dan atau pidana denda paling banyak Rp1.000.000.000,00 (satu miliar rupiah)
4. Setiap orang yang memenuhi unsur sebagaimana dimaksud pada ayat (3) yang dilakukan dalam bentuk pembajakan, dipidana dengan pidana penjara paling lama 10 (sepuluh) tahun dan atau pidana denda paling banyak Rp4.000.000.000,00 (empat miliar rupiah)



Daftar Isi

Modul Sekolah DevOps Cilsy.....	2
Daftar Isi.....	3
7. Automation Ansible.....	5
Learning Outcomes.....	5
Outline Materi.....	5
7.1. Ansible.....	6
7.1.1. Pengenalan Ansible.....	6
7.1.2. Kelebihan Ansible.....	7
7.1.3. Cara menjalankan Ansible.....	8
7.1.3.1. Ansible Ad-Hoc.....	8
7.1.3.2. Ansible Playbooks.....	8
7.1.3.3. Ansible Tower.....	10
7.1.4. Roles.....	10
7.1.4.1. Role Directory Structure.....	10
7.1.5. Inventory.....	12
7.1.6. Fungsi dan Kegunaan Ansible.....	12
7.1.7. Exercise.....	13
7.2. Instalasi dan Setup Ansible.....	14
7.2.1. Instalasi Ansible.....	14
7.2.2. SSH Setup.....	14
7.2.3. Playbook Structure.....	16
7.3. Konfigurasi Ansible.....	25
7.3.1. Install nginx+php dengan Ansible 1 server EC2.....	25
7.3.1.1. Konfigurasi EC2.....	25
7.3.1.2. Install nginx dan php.....	31
7.3.2. Install nginx+php dengan Ansible pada 2 server EC2.....	38
7.3.3. Exercise.....	43

7.4. Ansible Tower.....	44
7.4.1. Mengenal Ansible Tower.....	44
7.4.2. Fitur-Fitur Ansible Tower.....	44
7.4.2.1. Dasbor Tower.....	45
7.4.2.2. Streaming Real Time.....	45
7.4.2.3. Multi-Playbook/Multi Job.....	46
7.4.2.4. Secure.....	46
7.4.3. Exercise.....	47
7.5. Integrated Notification.....	47
7.6. Instalasi Ansible Tower.....	48
7.6.1. Exercise.....	53
7.7. Summary.....	53

7.

Automation Ansible

Learning Outcomes

Setelah selesai mempelajari bab ini, peserta mampu :

1. Memahami konsep dasar Ansible
2. Memahami cara kerja Ansible
3. Memahami komponen pada Ansible
4. Setting dan Konfigurasi Ansible

Outline Materi

1. Pengenalan Ansible
2. Instalasi Ansible
3. Deploying menggunakan Ansible

7.1. Ansible

7.1.1. Pengenalan Ansible

Ansible adalah sebuah *tool Automation*. Ansible dapat membantu melakukan instalasi, *deployment*, bahkan mengupdate *server*. ansible dapat membantu seorang devops atau sistem administrator untuk melakukan otomasi di servernya. Ansible berguna untuk mempermudah pekerjaan mengatur dan mengelola server yang memiliki jumlah yang sangat banyak, dengan beragam aplikasi atau software di dalamnya. Ansible juga cukup aman digunakan karena *script* ansible dapat di encrypt sehingga hanya orang tertentu yang dapat membuka dan menjalankan *script* ansible tersebut. Ansible dapat berjalan hanya dengan koneksi SSH dan tidak membutuhkan *agent* apapun untuk menjalankannya, namun untuk menjalankan ansible di *server deployment* minimal memiliki *library* python di dalam server *deployment* tersebut.

Ansible dikembangkan oleh RedHat. Kamu dapat mencatat setiap proses deployment ataupun konfigurasi yang biasa dilakukan berulang - ulang terhadap beberapa server. Misal saat pertama kali kita memasang Ubuntu Server di 10 mesin, maka kita akan melakukan apt-get update serta memasang beberapa komponen seperti PHP5 dan Apache2. Sebenarnya tidak akan menjadi masalah, bila kita hanya melakukan sedikit hal. Tapi bayangkan bila harus melakukan konfigurasi yang cukup kompleks dan dilakukan secara berulang - ulang ke 10 mesin tersebut.

Mungkin kita dapat membuat BASH script atau dengan menggunakan bahasa pemrograman lain untuk mempermudah masalah kita. Namun dengan Ansible, kita dapat melakukan pengelolaan server yang lebih mudah dan terstruktur. Selain itu dapat menghemat waktu kita untuk mengeksekusi proses konfigurasi atau deployment yang sama terhadap beberapa server yang kita miliki.

7.1.2. Kelebihan Ansible

Berikut adalah kelebihan ansible adalah sebagai berikut

- mudah di pahami, karena penulisan kode ansble harus mengikuti format yaml, dan format yaml sangat dekat dengan sysadmin. jadi bukan hal sulit untuk memahami tool ini. menurut pengalaman saya, saya lebih cepat memahami ansible dibanding tool automation lainnya.
- agentless, seperti yang kita ketahui kalau automation yang lain seperti puppet ataupun chef di haruskan menginstall agent pada target, dan coba bayangkan jika server yang kalian manage adalah 1000 server, ini menjadi kerjaan tambahan buat kalian.
- terdokumentasi, seperti yang kita sadari. kalau kapasitas manusia sangat terbatas. sebagai contoh saat ini kita diminta untuk setup 1 secara kompleks. lalu di bulan berikutnya kita diminta lagi untuk setup server lain dengan konfigurasi yang identik dengan server sebelumnya. hasilnya mungkin bisa beda. nah dengan tool ini kita cukup menulis satu kali configure, dan kita bisa eksekusi puluhan kali dengan hasil yang sama.
- skala yang besar, mungkin belum pernah saya alami, andai saja kita seorang sysadmin dan diberi tanggung jawab untuk memanage server berjumlah 1000 pcs. (apa mungkin kita setup satu-satu)
- support command bash, nah ini favorit saya. ketika perintah yang kita butuhkan tidak ada pada module kita bisa menggunakan module command, dan kita bisa mengetikkan perintah bash.

7.1.3. Cara menjalankan Ansible

Ansible sendiri memiliki beberapa cara yang dapat kita gunakan untuk menjalankannya. Berikut ada tiga cara untuk menjalankan ansible, yaitu:

7.1.3.1. *Ansible Ad-Hoc*

Kita bisa menggunakan Ansible untuk kebutuhan 1 task saja. Yap, dengan menggunakan ad-hoc ini, kita dengan mudah mengotomasi perintah pada host. Namun ad-hoc ini memiliki kekurangan, yakni hanya 1 task dan tidak reusable. Pada cara ini, kita dapat menjalankan ansible dengan menggunakan beberapa perintah intruksi command line melalui terminal. Berikut beberapa contoh perintah Ansible Ad-Hoc.

```
ansible localhost -m ping
ansible localhost -a "free -m"
ansible localhost -a "hostname && uname -a"
ansible localhost -a "ps ax | grep python"
```

7.1.3.2. *Ansible Playbooks*

Ansible Playbooks merupakan cara menjalankan ansible yang terdiri dari **playbook** atau **file YML**. File YML sendiri terdiri dari **task**, dimana task akan memanggil **modul-modul** ansible. Ansible playbook akan di jalankan secara berurutan. Ansible playbook sendiri membuat otomatisasi instalasi dan konfigurasi server, dengan banyak server. Sebagai contoh misalnya kita memiliki 100 server dan dari 100 server itu ingin di install webserver, database server, dll. Dengan kasus seperti diatas kita dapat menggunakan ansible-playbook. Jadi dengan sekali perintah maka otomatis 100 server yang kita punya akan melakukan instalasi secara otomatis.

Ansible-playbook itu pada dasarnya memiliki 3 bagian, yaitu:

- *Target*
- *Variable*

- *Task*

Nah berikut ini adalah contoh kode fullnya yang secara sederhana, dalam kasus ini kita akan menginstall git dengan menggunakan sistem operasi ubuntu. kita akan membuat file dengan nama **playbook.yml** lalu paste kode dibawah ini:

```
---
- name: Update repository ansible-playbook
  hosts: localhost
  remote_user: taufik
  become: yes
  tasks:
    - name : update repo
      apt:
        update_cache: yes
```

Coba perhatikan kode diatas. Ada beberapa bagian yang harus diperhatikan:

- **name** digunakan untuk menampilkan deskripsi di terminal saat task tersebut sedang dijalankan.
- **host** digunakan untuk menginisialisasi **target** host mana yang akan kita install menggunakan ansible-playbook.
- **remote_user** user untuk sistem operasi ubuntu, disini saya menggunakan user **taufik** jadi secara otomatis usernya adalah **taufik**
- **become** sebagai option bahwa user vagrant tersebut memiliki kemampuan untuk menjadi user lain (secara default, bervalue **root**).
- **task** adalah perintah yang akan dilakukan (**Task**), dalam kali ini adalah untuk mengupdate repo. Task memiliki nested apt. Jika hal tersebut di convert ke bash maka akan menjadi 'sudo apt-get update'

7.1.3.3. Ansible Tower

Ansible Tower merupakan automation untuk kalangan Enterprise dan juga berbayar. Ansible Tower memiliki banyak kemudahan di bandingkan ansible tipe lain. Ansible Tower lebih mudah untuk mengontrol server di karenakan memiliki halaman UI, selain itu juga memiliki modul-modul lain. Ansible tower juga menyediakan sistem keamanan secara langsung pada layanannya.

7.1.4. Roles

Roles adalah cara memuat secara otomatis vars_files, tasks, dan handler tertentu berdasarkan pada struktur file yang dikenal. Mengelompokkan konten berdasarkan roles juga memungkinkan berbagi roles dengan pengguna lain dengan mudah.

Dengan cara ini, kita dapat melakukan task sesuai kebutuhan dan reusable bila mana ingin menggunakannya untuk kebutuhan lain. Roles didefinisikan menggunakan yaml file didalam struktur direktori yang telah ditentukan. Untuk lebih jelasnya, mari lihat contoh Role Directory Structure berikut.

7.1.4.1. Role Directory Structure

Berikut merupakan contoh stuktur dari sebuah project

```
site.yml
webservers.yml
fooservers.yml
roles/
  common/
    tasks/
    handlers/
    files/
    templates/
    vars/
    defaults/
    meta/
```



```
webservers/
  tasks/
  defaults/
  meta/
```

Dapat dilihat bahwa pada contoh terdapat 2 roles yakni **common** dan **webservers**. Di dalam masing masing roles terdapat folder yang masing masing berbeda sesuai kebutuhan.

Roles mengharapkan file berada dalam nama direktori tertentu. role harus menyertakan setidaknya satu dari direktori ini, namun tidak masalah untuk mengecualikan direktori yang tidak digunakan. Saat digunakan, setiap direktori harus berisi file **main.yml** , yang berisi konten yang relevan:

- tasks - berisi daftar tugas utama yang akan dijalankan oleh role.
- handlers - berisi handler, yang dapat digunakan oleh role.
- defaults - variabel default untuk role.
- vars - variabel lain untuk role.
- files - berisi file yang dapat digunakan untuk deploy melalui role
- templates - berisi template yang dapat digunakan melalui role
- meta - mendefinisikan beberapa data meta untuk role

Dalam eksekusi nya, ketika kita ingin menggunakan role tertentu, kita tinggal mendefinisikannya pada file yml (yang berada diluar folder **roles**) seperti berikut.

```
---
- name: Install web server
  hosts: all
  roles:
    - webservers
```

Ansible secara otomatis akan menjalankan isi folder dari **Roles webservers** seperti **tasks**, **default** dan **meta**.

7.1.5. Inventory

Inventory adalah file yang berisi list serta grup server kamu. Secara default filenya berada di `/etc/ansible/hosts`, tapi kamu juga bisa meletakkan file inventory ini di path lain dan menggunakan option `-i` atau `-inventory` untuk membaca file tersebut saat menjalankan Ansible.

Contoh inventory seperti ini

```
test.cilsy.com

[webserver]
server1.cilsy.com
server2.cilsy.com

[dbserver]
server3.cilsy.com

[myserver:children]
webserver
dbserver
```

Diatas kita mempunyai server `test.cilsy.com` yang tidak berada didalam grup manapun, `server1.cilsy.com` dan `server2.cilsy.com` yang berada didalam grup `webserver` serta `server3.cilsy.com` yang berada didalam grup `dbserver`.

Lalu untuk grup `webserver` dan `dbserver` sendiri merupakan anggota/children dari grup `myserver`. Dengan begitu `server1.cilsy.com` – `server3.cilsy.com` juga merupakan anggota grup `myserver`.

7.1.6. Fungsi dan Kegunaan Ansible

- Ansible dapat berjalan pada banyak server secara sekaligus.
- Ansible hanya memerlukan koneksi SSH dan tidak memerlukan agent pada server yang akan di deploy.

- Ansible dapat menginstall aplikasi, deploy software hingga mengupdate server.

7.1.7. Exercise

1. Jelaskan menurut kalian bagaimana cara kerja dari Ansible !
2. Ada berapa banyak dan apa saja model dari Ansible ?

7.2. Instalasi dan Setup Ansible

7.2.1. Instalasi Ansible

Pada tahap ini kita akan menginstallkan ansible, ansible sendiri membutuhkan python untuk memaksimalkan kinerjanya. Maka dari itu pertama kita harus mengupdate repository linux kita, lalu installkan python dan juga ansible seperti dibawah ini.

```
apt-get update
apt-get install python ansible
```

7.2.2. SSH Setup

Ansible menggunakan koneksi SSH untuk berkomunikasi dengan target host, dan SFTP untuk transfer modul. Maka dari itu, kita harus mensetup terlebih dahulu SSH antara **Control Node** (yang menjalankan **Ansible**) dan **Managed Node** (target Ansible). Tujuan dari setup ini, agar kita tidak lagi memasukan username password apabila melakukan koneksi SSH.

Pastikan **Control Node** dan **Managed Node** sudah terinstall SSH. Pada **Control Node**, kita buat **RSA Key Pair** dengan mengeksekusi perintah berikut.

```
ssh-keygen
```

Lalu akan muncul prompt untuk menyimpan **RSA** nya dan password dari **RSA**. Untuk saat ini, kita biarkan default saja, jadi kita tinggal tekan tombol Enter saja.

```
Generating public/private rsa key pair.
Enter file in which to save the key (/home/taufik/.ssh/id_rsa):
Created directory '/home/taufik/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/taufik/.ssh/id_rsa
Your public key has been saved in /home/taufik/.ssh/id_rsa.pub
```

The key fingerprint is:

SHA256:uPYfD6fYR7V0xe1XslmVW3N8r1kXDbEaNvaelZmyv0 taufik@hewlettpackard

Kurang lebih, setup nya akan seperti ini :

```
taufik@hewlettpackard:~$ ssh-keygen
Generating public/private rsa key pair.
Enter file in which to save the key (/home/taufik/.ssh/id_rsa):
Created directory '/home/taufik/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/taufik/.ssh/id_rsa
Your public key has been saved in /home/taufik/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:uPYfD6fYR7V0xe1XslmVW3N8r1kXDbEaNvaelZmyv0 taufik@hewlettpackard
The key's randomart image is:
+---[RSA 3072]---+
|                 *=B|
|                + %X|
|               . +=&|
|              o*.*@|
|             . S  .. &E|
|            .   . B o|
|           o   o..o .|
|          . . o *. . |
|         o.+..      |
+-----[SHA256]-----+
```

Langkah selanjutnya adalah untuk menyalin **Public key** ke **Managed Node**. Kita bisa melakukannya dengan mengeksekusi perintah berikut.

`ssh-copy-id username@remote_host`

Isi **username** dan **remote_host** sesuai dengan yang akan dijadikan **Managed Node**. Setelah itu, isikan **password** dari **username** tersebut. Kurang lebih setupnya menjadi seperti ini. **Anda harus mengganti username dan/atau IP target sesuai dengan target anda masing masing.**

```
taufik@hewlettpackard:~$ ssh-copy-id root@172.17.0.3
/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter
out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompt
ed now it is to install the new keys
root@172.17.0.3's password:

Number of key(s) added: 1

Now try logging into the machine, with:  "ssh 'root@172.17.0.3'"
and check to make sure that only the key(s) you wanted were added.
```

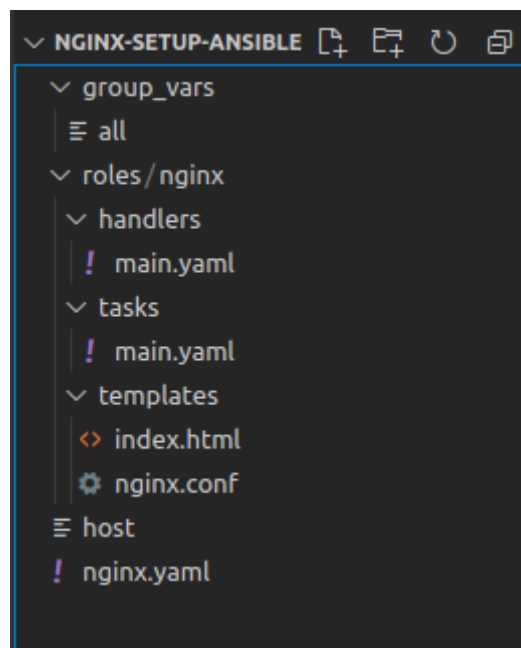


7.2.3. Playbook Structure

Setelah kita menginstall ansible, langkah selanjutnya adalah memahami struktur file dan folder ansible. File ansible dapat kita clone dengan menggunakan perintah di bawah ini.

```
git clone https://github.com/sdcilsy/nginx-setup-ansible.git
```

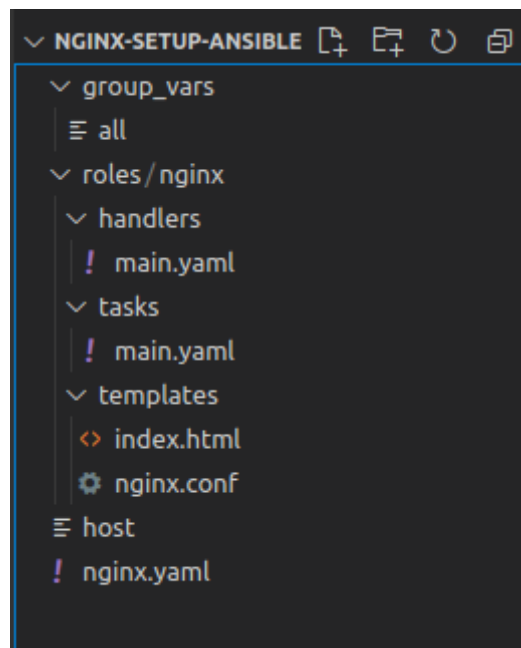
Setelah kita clone file ansible tersebut, kita bisa lihat isi file seperti berikut ini.



Pada ansible terdapat 2 folder yang bernama **group_vars** dan juga **roles** serta 1 file bernama **hosts**. Folder **group_var** berguna untuk mendefinisikan variable-variable untuk ansible, folder **roles** berguna untuk membuat fungsi-fungsi yang ada di ansible. Sedangkan file **hosts** berguna untuk mendeklarasikan ip server yang ingin kita provisioning atau install dengan ansible.

Pada bagian selanjutnya terdapat sebuah file dengan nama **all** pada folder **group_vars**. Nantinya file all akan berisi variable yang akan di gunakan pada ansible.

Kita akan mencoba install nginx dengan menggunakan ansible pada folder roles. Selanjutnya buat folder **nginx** di dalam folder **roles**, di dalam folder nginx akan di buat folder **tasks**, **templates** dan **handlers**. Folder **tasks** berisikan file perintah yang di gunakan pada ansible, file yang berada pada folder **tasks** adalah **main.yml**, folder **templates** berisikan file yang akan di kirim ke server misal **nginx.conf**. Sedangkan folder **handlers** berguna untuk **restart** atau **start** service **nginx**.



Kita akan coba jelaskan isi file dari **main.yml** dari folder **task** sebagai berikut ini.

```
---
- name: Update repository nginx
  apt_repository: repo='ppa:nginx/stable' state=present update_cache=yes
  become: yes
```

Perintah di atas merupakan perintah untuk mengupdate repository dari nginx

```
- name: Install Nginx
  apt: name={{ variable }} state=present update_cache=true
```

Selanjutnya perintah di atas berguna untuk menginstall nginx. Kita menggunakan variable yang di ambil dari folder group_vars dan file all,

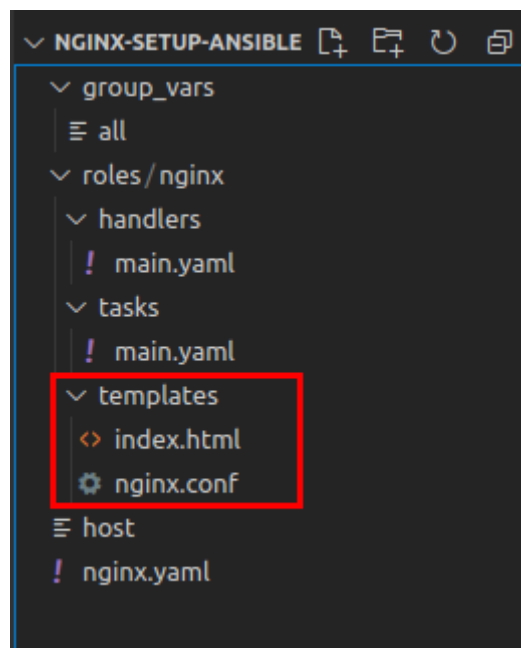


sehingga kita dapat mengkonfigurasi ansible secara dynamic. Kelebihan kita menggunakan variable secara dynamic adalah saat ada perubahan value, kita cukup mengubah pada file all yang berada di group_vars.

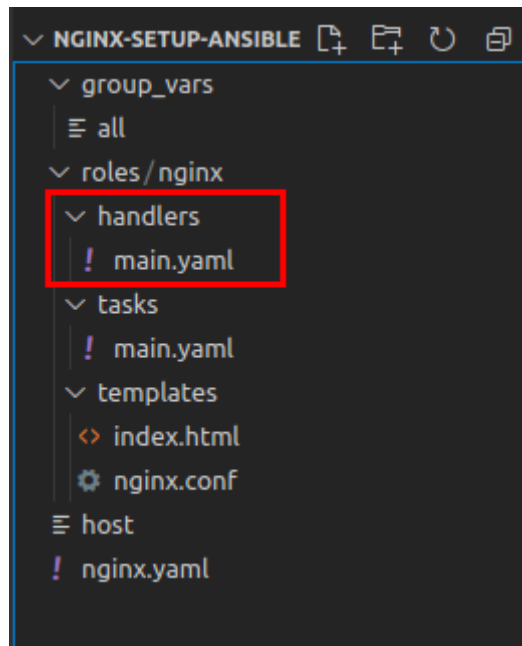
```
- name: Copy nginx configuration
  template: src=nginx.conf dest=/etc/nginx/conf.d/default.conf

- name: Copy index.html
  template: src=index.html dest=/var/www/html/index.html
```

Selanjutnya perintah di atas berguna untuk menyalin atau transfer file ke server, file yang kita transfer kita masukkan pada folder templates seperti gambar di bawah ini.



Selanjutnya adalah folder **handlers** yang berguna untuk melakukan start maupun restart service nginx. Pada folder handlers berisikan file **main.yml**, adapun struktur file direktori dapat kita lihat seperti di bawah ini.



File **main.yml** yang berada pada folder handler berisikan beberapa script perintah. Kita dapat melihat isinya seperti dibawah ini.

```
---
- name: start nginx
  service: name=nginx state=started

- name: reload nginx
  service: name=nginx state=reloaded

- name: restart nginx
  service: name=nginx state=restarted
```

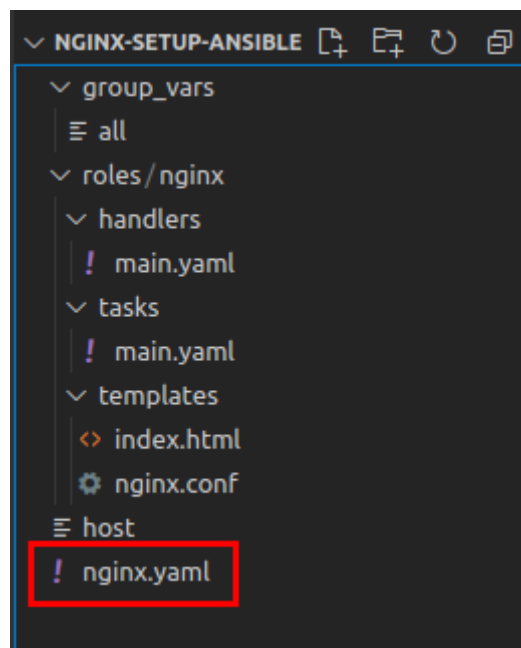
File yang terakhir yang harus kita pahami adalah file **hosts**, file ini berguna untuk mengatur server yang akan kita provisioning atau install melalui ansible. Berikut merupakan isi dari file host tersebut. **Anda harus mengganti ip pada Group Client dibawah, sesuai dengan target anda masing masing.**

```
[local]
127.0.0.1 ansible_ssh_private_key_file=/vagrant/awskeyprivate.pem
[client]
172.17.0.3
```



Perintah di atas merupakan contoh dari file hosts, file host berisikan ip server yang ingin kita provisioning atau install melalui ansible. Karena kita sudah setup SSH, kita tidak perlu lagi memasukkan username serta password.

Setelah folder dan file dari ansible sudah kita siapkan, selanjutnya untuk menjalankan ansible kita memerlukan file yml. Misalkan kita akan menggunakan nama **nginx.yml**, struktur file dapat kita lihat seperti gambar dibawah ini.



Disini kita akan coba jelaskan beberapa fungsi dari komponen script yang ada pada file nginx.yml sebagai berikut.

- **-name** berguna untuk menampilkan nama proses saat di jalankan
- **hosts : client** merupakan server yang akan di install menggunakan ansible file harus sesuai dengan file hosts
- **become : yes** merupakan perintah untuk menjalankan perintah sudo pada server yang akan kita install dengan ansible
- **roles : - nginx** merupakan kumpulan perintah yang berada di folder roles, sedangkan nginx adalah folder yang berada di roles

Setelah semua konfigurasi atau settingan telah di lakukan, sekarang kita dapat menjalankan ansible dengan menggunakan perintah seperti di bawah ini pada folder devops ansible.

```
ansible-playbook -i hosts nginx.yml
```

Maka hasilnya bisa kita lihat jika berhasil maka akan muncul seperti di bawah ini.


```
PLAY [Install Nginx with ansible] *****
TASK [Gathering Facts] *****
ok: [172.17.0.3]
TASK [nginx : Update repository] *****
ok: [172.17.0.3]
TASK [nginx : Install Nginx] *****
changed: [172.17.0.3]
TASK [nginx : Create web directory] *****
changed: [172.17.0.3]
TASK [nginx : Copy nginx configuration] *****
changed: [172.17.0.3]
TASK [nginx : Copy index.html] *****
ok: [172.17.0.3]
PLAY RECAP *****
172.17.0.3 : ok=6  changed=3  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
```

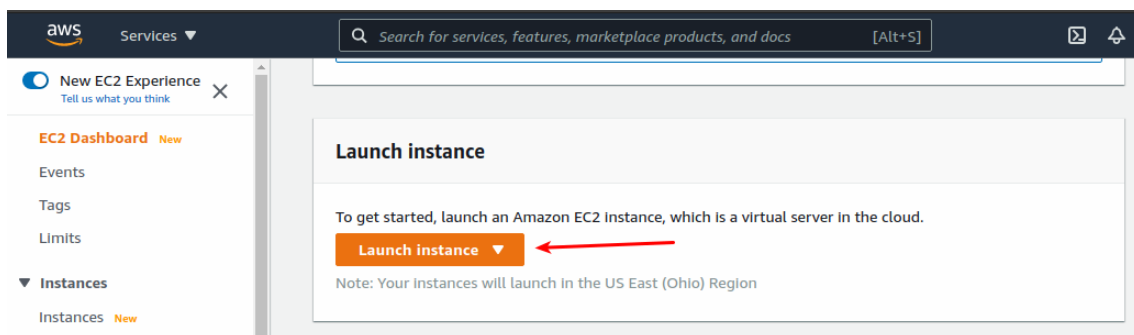


7.3. Konfigurasi Ansible

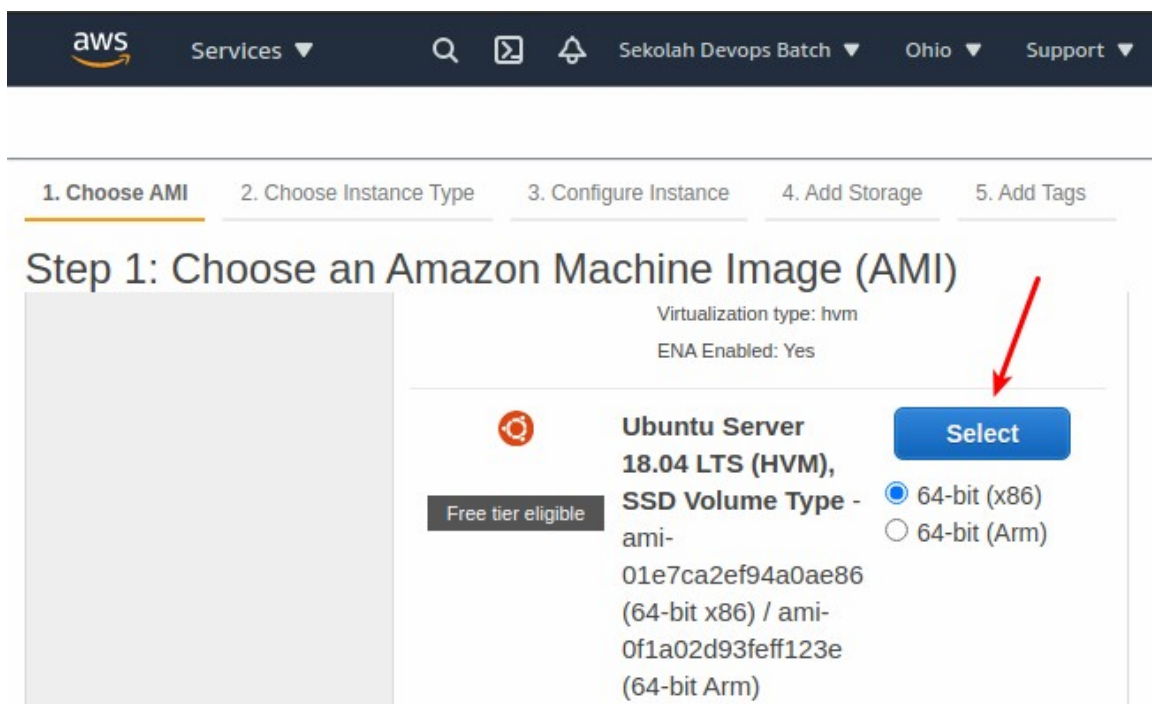
7.3.1. Install nginx+php dengan Ansible 1 server EC2

7.3.1.1. Konfigurasi EC2

Pada bagian ini kita akan coba menginstallkan nginx dan php dengan menggunakan ansible pada 1 buah server EC2. Tahap pertama yang akan kita lakukan adalah membuat EC2, kita bisa membuatnya dengan mengikuti beberapa tahapan berikut.



Setelah kita melakukan klik launch pada EC2 dashboar, sekarang kita pilih AMI Linux **Ubuntu Server 18.04** yang akan kita gunakan untuk ansible.



Pilih type instance yang akan kita gunakan, menggunakan t2.micro saya rasa sudah cukup.

Step 2: Choose an Instance Type

Instance Type	Instance Type	Instance Type	Number of Instances	On-Demand Price	Storage
<input type="checkbox"/>	t2	t2.nano	1	0.5	EBS only
<input checked="" type="checkbox"/>	t2	t2.micro Free tier eligible	1	1	EBS only
<input type="checkbox"/>	t2	t2.small	1	2	EBS only
<input type="checkbox"/>	t2	t2.medium	2	4	EBS only
<input type="checkbox"/>	t2	t2.large	2	8	EBS only
<input type="checkbox"/>	t2	t2.xlarge	4	16	EBS only
<input type="checkbox"/>	t2	t2.2xlarge	8	32	EBS only
<input type="checkbox"/>	t3	t3.nano	2	0.5	EBS only

Buttons: Cancel, Previous, Review and Launch, Next: Configure Instance Details

Pada langkah selanjutnya, kita biarkan default saja, kecuali anda ingin mengubah VPC, atau jumlah dari instancenya.

Step 3: Configure Instance Details

Configure the instance to suit your requirements. You can launch multiple instances from the same AMI, request Spot instances to take advantage of the lower pricing, assign an access management role to the instance, and more.

Number of instances: 1 [Launch into Auto Scaling Group](#)

Purchasing option: ☐ Request Spot instances

Network: vpc-6c71a007 (default) [Create new VPC](#)

Subnet: No preference (default subnet in any Availability Zone) [Create new subnet](#)

Auto-assign Public IP: Use subnet setting (Enable)

Placement group: ☐ Add instance to placement group

Buttons: Cancel, Previous, Review and Launch, Next: Add Storage

Kita akan masuk kedalam menu storage, disini kita biarkan saja default lalu klik Next.

Step 4: Add Storage

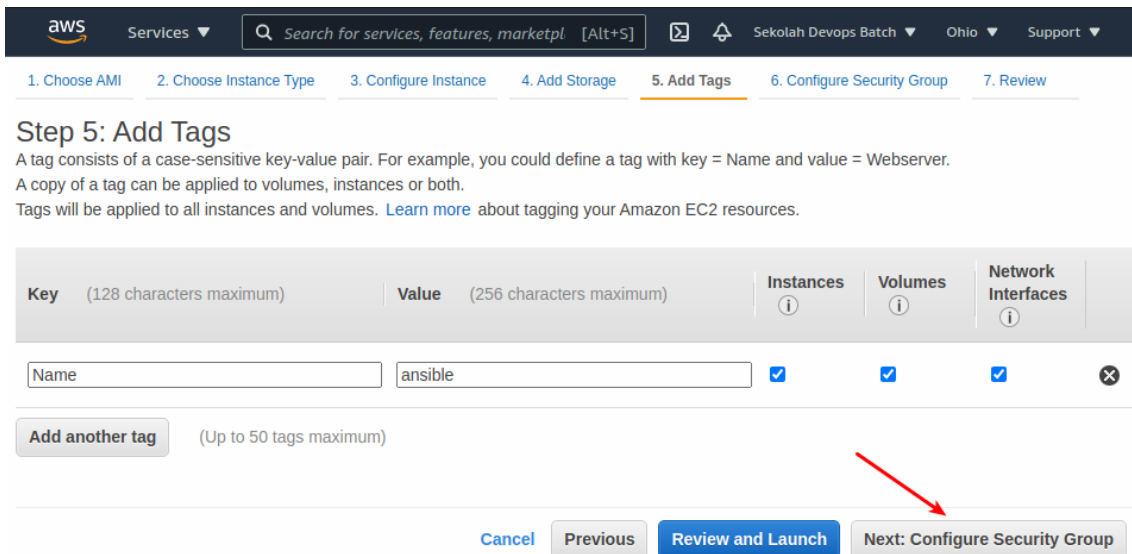
Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encryption
Root	/dev/sda1	snap-0f1492ba76f867da2	8	General Purpose S3	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

[Add New Volume](#)

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Add Tags](#)

Langkah selanjutnya adalah membuat nama tags. Isi **Name** dengan value **ansible**, lalu klik Next Configuration Security Group.



Step 5: Add Tags

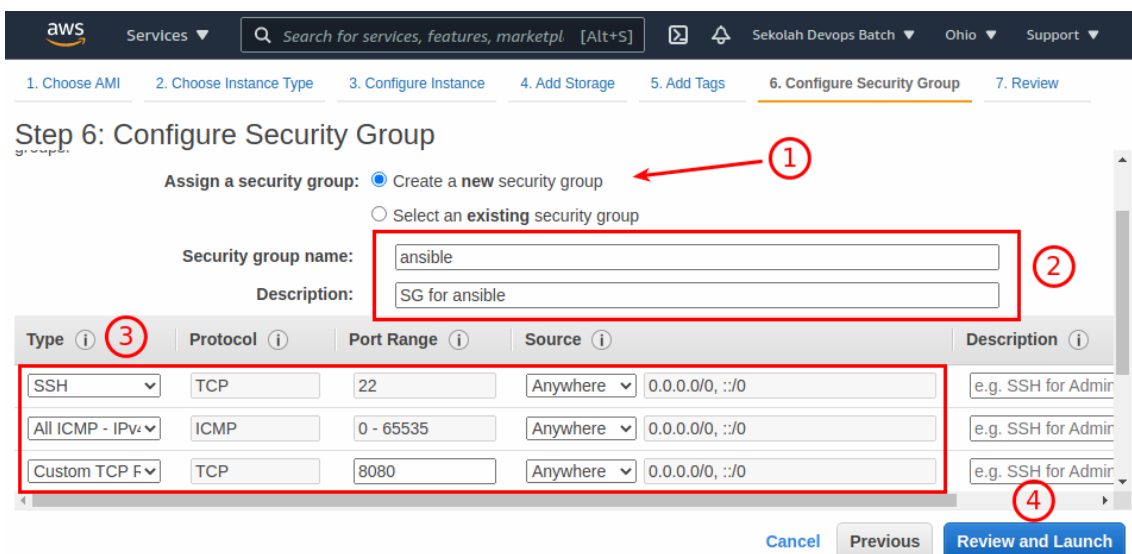
A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. A copy of a tag can be applied to volumes, instances or both. Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key	Value	Instances	Volumes	Network Interfaces
Name	ansible	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

[Add another tag](#) (Up to 50 tags maximum)

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Configure Security Group](#)

Selanjutnya adalah bagian security group, buat sebuah type protocol dengan **SSH, TCP Port 8080** dan **ICMP IPv4** dengan source **anywhere** seperti gambar dibawah ini. Lalu klik review untuk melanjutkan.



Step 6: Configure Security Group

Assign a security group: ☒ Create a new security group [1](#)
☐ Select an existing security group

Security group name: [2](#)
Description:

Type	Protocol	Port Range	Source	Description
SSH 3	TCP	22	Anywhere	0.0.0.0/0, ::/0
All ICMP - IPv4	ICMP	0 - 65535	Anywhere	0.0.0.0/0, ::/0
Custom TCP F	TCP	8080	Anywhere	0.0.0.0/0, ::/0

[Cancel](#) [Previous](#) [Review and Launch](#) [4](#)

Pada bagian review, kita tinggal mengeklik tombol **Launch**



aws Services Search for services, features, marketpl [Alt+S] Sekolah Devops Batch Ohio Support

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 7: Review Instance Launch

▼ AMI Details [Edit AMI](#)

Ubuntu Server 18.04 LTS (HVM), SSD Volume Type - ami-01e7ca2ef94a0ae86

Free tier eligible

Ubuntu Server 18.04 LTS (HVM), EBS General Purpose (SSD) Volume Type. Support available from Canonical (<http://www.ubuntu.com/cloud/services>).

Root Device Type: ebs Virtualization type: hvm

▼ Instance Type [Edit instance type](#)

Instance Type	ECUs	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance
t2.micro	-	1	1	EBS only	-	Low to Moderate

[Cancel](#) [Previous](#) [Launch](#)

Lalu, buat **key pair** baru untuk koneksi SSH. Isikan nama, lalu download. Kemudian klik **Launch**.

Select an existing key pair or create a new key pair

A key pair consists of a **public key** that AWS stores, and a **private key file** that you store. Together, they allow you to connect to your instance securely. For Windows AMIs, the private key file is required to obtain the password used to log into your instance. For Linux AMIs, the private key file allows you to securely SSH into your instance.

Note: The selected key pair will be added to the set of keys authorized for this instance. Learn more about [removing existing key pairs from a public AMI](#).

1

2 ansible

3 [Download Key Pair](#)

4

You have to download the **private key file** (*.pem file) before you can continue. **Store it in a secure and accessible location.** You will not be able to download the file again after it's created.

[Cancel](#) [Launch Instances](#)

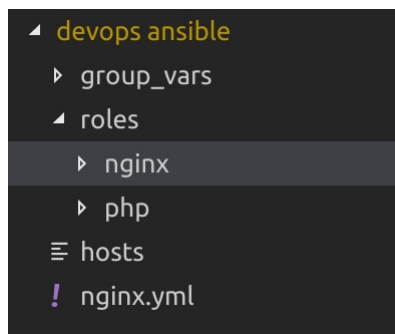
Tunggu sampai EC2 berjalan. Langkah selanjutnya adalah untuk menginstall nginx+php pada server EC2 yang telah di buat.

7.3.1.2. Install nginx dan php

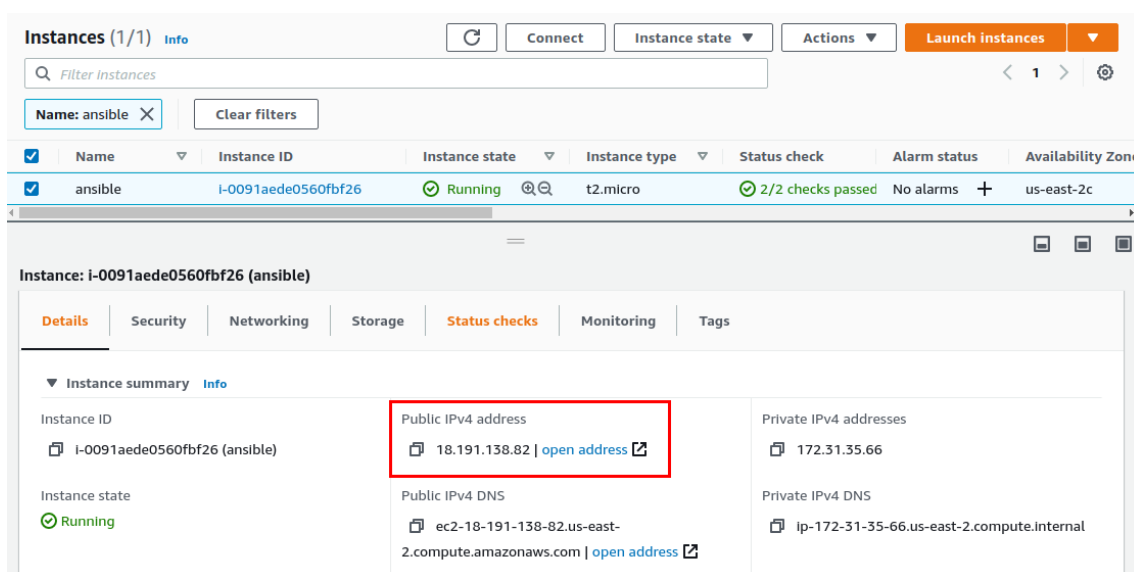
Untuk menginstal nginx+php dengan menggunakan ansible kita harus menyiapkan struktur file dan folder sebagai berikut. Kita bisa menggunakan repository yang sudah dibuat.

```
git clone https://github.com/sdcilsy/nginx-php-setup-ansible.git
```

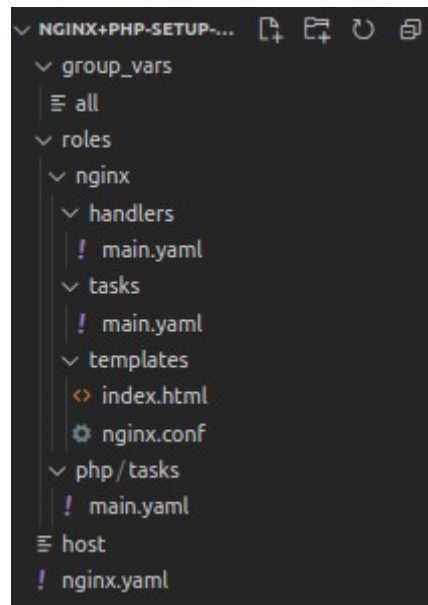
Namun, kita perlu untuk mengedit beberapa file. Kita akan membahasnya sebentar lagi.



Langkah selanjutnya, kita cek ip public dari instance EC2 yang telah kita buat sebelumnya, catat ip public tersebut untuk selanjutnya kita akan pergunakan pada ansible.



Folder yang perlu kita atur adalah **nginx** dan **php** yang berada di folder **roles**, selanjutnya kita akan menkonfigurasi file **main.yml** berada pada folder **nginx/task** seperti di bawah ini struktur folder dan filenya.



File **main.yml** tersebut kita edit beberapa perintah script seperti di bawah ini.

```
---
- name: Update Repository
  apt: update_cache=true
  become: yes

- name: Install Nginx
  apt: name=nginx state=present update_cache=true
  notify:
    - start nginx

- name: Copy nginx configuration
  template: src=default.conf dest=/etc/nginx/sites-enabled/default
  notify:
    - restart nginx

- name: Copy web html
```




```
template: src=index.html dest=/var/www/html/index.html
```

Selanjutnya kita akan mengecek file yang ada pada **templates**, file **nginx.conf** yang akan kita kirim pada server isinya kurang lebih seperti di bawah ini.

```
server {
    listen 80 default_server;
    listen [::]:80 default_server;

    root /var/www/html;
    # Add index.php to the list if you are using PHP
    index index.html index.php index.htm index.nginx-debian.html;
    server_name _;
    location / {
        # First attempt to serve request as file, then
        # as directory, then fall back to displaying a 404.
        try_files $uri $uri/ =404;
    }

    location ~ \.php$ {
        include snippets/fastcgi-php.conf;
        fastcgi_pass unix:/var/run/php/php7.0-fpm.sock;
    }
}
```

Jika kita ingin mengirim file web (**index.html**), kita dapat meletakkan file tersebut pada **templates** dengan isi sebagai berikut.

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>YAY, ANSIBLE SUCCESSFULLY SETUP NGINX</title>
```



```
</head>
<body>
  <h1>Welcome to Sekolah Devops Cilsy</h1>
</body>
</html>
```

Kemudian pada folder **handlers** terdapat file **main.yml** yang isinya sebagai berikut.

```
---
- name: start nginx
  service: name=nginx state=started

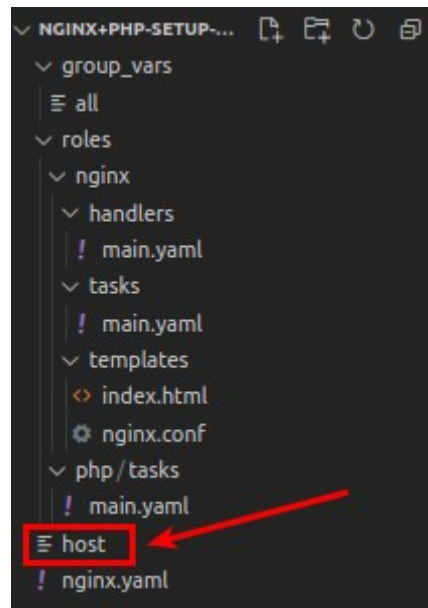
- name: reload nginx
  service: name=nginx state=reloaded

- name: restart nginx
  service: name=nginx state=restarted
```

Setelah semua file pada folder **nginx** sudah kita buat, maka selanjutnya kita akan buat file yang ada di folder **php** dibawah folder **roles**. Setelah folder **php** sudah kita buat, kita akan membuat file **main.yml** pada folder **tasks** dengan isi sebagai berikut ini.

```
---
- name: Install PHP Package
  apt: name={{ php_ver }} state=present update_cache=true
  become: yes
```

Setelah selesai dengan file di atas, selanjutnya kita konfigurasi file **host** pada ansible. Adapun struktur file dan folder bisa kita lihat seperti berikut.



Isikan IP Public yang tadi sudah kita simpan dari EC2 instance, lalu isi keypair yang sebelumnya sudah didownload. Kita akan melakukan login menggunakan user **Ubuntu**. **Anda harus mengganti IP dan juga Key pair pada file host**

```
[client]
18.191.138.82 ansible_ssh_private_key_file=/home/taufik/ansible-do_13-1.pem

[all:vars]
ansible_user=ubuntu
```

Selanjutnya kita akan membuat file nginx.yml seperti di bawah ini. File **nginx.yml** sendiri berada di dalam folder devops_ansible atau folder ansible.

```
---
- name: Install Nginx+PHP with ansible
  hosts: client
  become: yes
  roles:
    - nginx
    - php
```

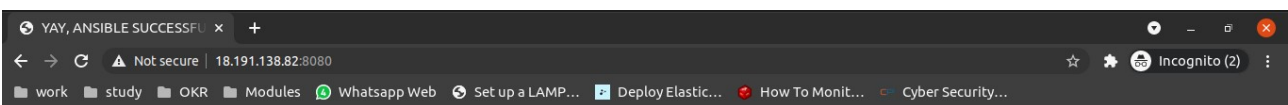
Setelah semua sudah di buat maka langkah selanjutnya adalah dengan menjalankan ansible dengan menggunakan perintah berikut.

```
ansible-playbook -i hosts nginx.yml
```

Jika berhasil, maka akan muncul tampilan seperti gambar dibawah ini.

```
PLAY [Install Nginx+PHP with ansible] *****
TASK [Gathering Facts] *****
ok: [18.191.138.82]
TASK [nginx : Update repository] *****
ok: [18.191.138.82]
TASK [nginx : Install Nginx] *****
changed: [18.191.138.82]
TASK [nginx : Create web directory] *****
changed: [18.191.138.82]
TASK [nginx : Copy nginx configuration] *****
changed: [18.191.138.82]
TASK [nginx : Copy index.html] *****
changed: [18.191.138.82]
TASK [php : Install PHP Package] *****
changed: [18.191.138.82]
RUNNING HANDLER [nginx : start nginx] *****
ok: [18.191.138.82]
RUNNING HANDLER [nginx : restart nginx] *****
changed: [18.191.138.82]
PLAY RECAP *****
18.191.138.82 : ok=9  changed=6  unreachable=0  failed=0  skipped=0  rescued=0  ignored=0
```

Setelah itu kita check aplikasi yang sudah kita setting tadi dengan mengakses **IP Public** ke browser denga port **8080** dan hasilnya akan muncul seperti gambar dibawah ini.



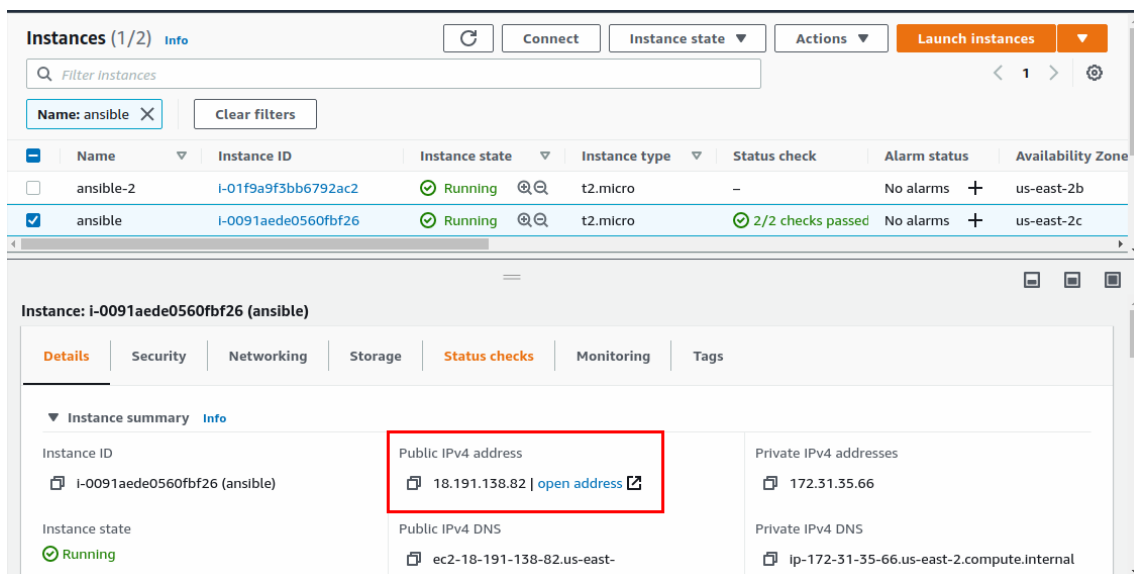
Welcome to Sekolah Devops Cilsy



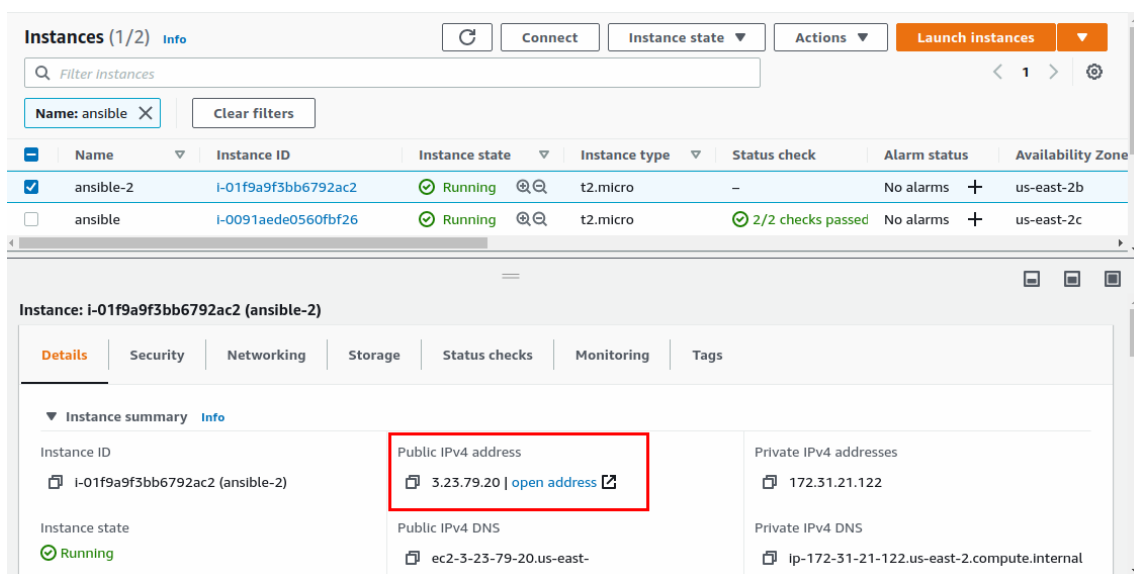
7.3.2. Install nginx+php dengan Ansible pada 2 server EC2

Pada bagian ini kita akan meginstall nginx dan php pada 2 server EC2, berbeda dengan yang sebelumnya yang hanya menggunakan 1 server EC2. Kita hanya perlu menambahkan hosts atau server yang ingin kita install, langkah konfigutasinya tidak jauh berbeda dengan yang sebelumnya.

Langkah pertama adalah megecek IP Public pada EC2 Instance yang sudah kita buat sebelumnya, seperti gambar dibawah ini.



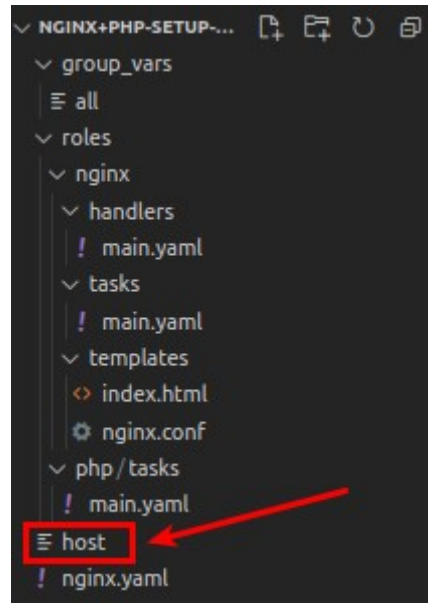
The screenshot shows the AWS Management Console 'Instances' page. A table lists two instances: 'ansible-2' and 'ansible'. The 'ansible' instance is selected. Below the table, the 'Instance: i-0091aede0560fbf26 (ansible)' details are shown. In the 'Instance summary' section, the 'Public IPv4 address' is listed as '18.191.138.82 | open address', which is highlighted with a red rectangular box.



The screenshot shows the AWS Management Console 'Instances' page. A table lists two instances: 'ansible-2' and 'ansible'. The 'ansible-2' instance is selected. Below the table, the 'Instance: i-01f9a9f3bb6792ac2 (ansible-2)' details are shown. In the 'Instance summary' section, the 'Public IPv4 address' is listed as '3.23.79.20 | open address', which is highlighted with a red rectangular box.



Bisa kita lihat disana terdapat dua buah instance yang masing-masing memiliki IP Public yaitu **18.191.138.82** dan **3.23.79.20**. Selanjutnya ip tersebut kita masukkan ke dalam file hosts , adapun struktur file atau folder bisa kita lihat sebagai berikut.



Ubah file host menjadi seperti dibawah ini.

```
[client]
18.191.138.82 ansible_ssh_private_key_file=/home/taufik/ansible-do_13-1.pem
3.23.79.20 ansible_ssh_private_key_file=/home/taufik/ansible-do_13-1.pem

[all:vars]
ansible_user=ubuntu
```

Jika sudah kita konfigurasi, selanjutnya kita simpan file host tersebut. Langkah selanjutnya adalah menjalankan kembali ansible dengan menggunakan perintah berikut.

```
ansible-playbook -i hosts nginx.yml
```

Jika berhasil maka akan muncul seperti gambar dibawah ini. Pada awal eksekusi, anda akan diminta untuk mengkonfirmasi koneksi menuju **instace ke-2**. Anda bisa mengetik **yes** untuk melanjutkan.

```
PLAY [Install Nginx+PHP with ansible] *****
TASK [Gathering Facts] *****
The authenticity of host '3.23.79.20 (3.23.79.20)' can't be established.
ECDSA key fingerprint is SHA256:S+GKtAekH6fgP50KkGpfD5y7ZMcH2M9KU8CX7VcF4q0.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
ok: [18.191.138.82]
ok: [3.23.79.20]

TASK [nginx : Update repository] *****
ok: [18.191.138.82]
changed: [3.23.79.20]

TASK [nginx : Install Nginx] *****
ok: [18.191.138.82]
changed: [3.23.79.20]

TASK [nginx : Create web directory] *****
changed: [3.23.79.20]
ok: [18.191.138.82]

TASK [nginx : Copy nginx configuration] *****
ok: [18.191.138.82]
changed: [3.23.79.20]

TASK [nginx : Copy index.html] *****
ok: [18.191.138.82]
changed: [3.23.79.20]

TASK [php : Install PHP Package] *****
ok: [18.191.138.82]
changed: [3.23.79.20]

RUNNING HANDLER [nginx : start nginx] *****
ok: [3.23.79.20]

RUNNING HANDLER [nginx : restart nginx] *****
changed: [3.23.79.20]

PLAY RECAP *****
18.191.138.82      : ok=7    changed=0    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
3.23.79.20        : ok=9    changed=7    unreachable=0    failed=0    skipped=0    rescued=0    ignored=0
```

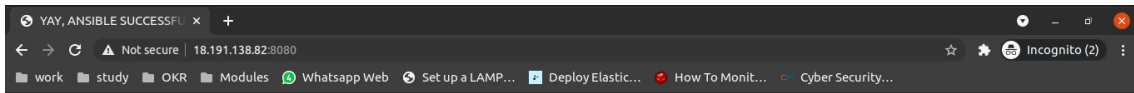
Ada beberapa perbedaan yakni ada yang statusnya **OK** dan **Changed**. Mengapa instance pertama **OK** semua, sedangkan instance kedua ada yang **Changed** ?

Ini terjadi karena pada **instance pertama**, kita sudah mengkonfigurasi sebelumnya, sedangkan pada **instance kedua**, kita baru mengkonfigurasi pertama kali.

Kita hanya perlu mengecek hasil yang sudah kita buat tadi dengan mengakses IP Public dengan menggunakan browser seperti dibawah ini.

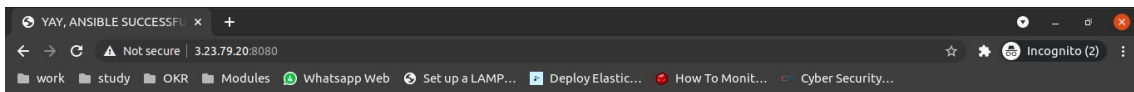
Instance pertama.





Welcome to Sekolah Devops Cilsy

Instance kedua



Welcome to Sekolah Devops Cilsy

Untuk mencoba instalasi paket-paket linux yang baru, kita dapat menggunakan ansible-playbook. Kita dapat mengambilnya dengan menclone repository seperti berikut.

```
git clone https://github.com/adisaputra10/ansible-examples.git
```

Berikut merupakan file atau data mengenai ansible-example yang dapat kita lihat seperti dibawah ini.



407 commits	6 branches	0 releases	71 contributors
Branch: master	New pull request	Create new file	Upload files
		Find file	Clone or download
webknjaz Merge pull request #238 from jcsmith/cleanup_lamp_haproxy Latest commit 725816b 9 days ago			
jboss-standalone	Replace old = syntax with newer valid yaml syntax.	10 days ago	
lamp_haproxy	Fix missing single quotes/syntax error.	9 days ago	
lamp_simple	Removed for common role unnecessary	2 years ago	
lamp_simple_rhel7	Update all	2 years ago	
language_features	minor typo	4 months ago	
mongodb	Fix header formatting	2 months ago	
rust-module-hello-world	Add a binary rust example module	3 months ago	
tomcat-memcached-failover	remove iptables line for https and use ansible_default_ipv4 for memca...	3 years ago	
tomcat-standalone	become instead of sudo	2 years ago	
windows	The win_msi module has been deprecated	5 months ago	
wordpress-nginx	Merge pull request #173 from clam/fixing-issue-172	2 years ago	
wordpress-nginx_rhel7	Merge pull request #167 from sadsfae/master	2 years ago	
.gitignore	Remove .DS_Store files from openshift playbook example. Added .DS_sto...	4 years ago	
README.md	changes	6 years ago	
README.md			

Untuk dapat menjalankan beberapa konfigurasi tersebut, kita hanya perlu masuk ke folder, lalu setup host sesuai ip tujuan. Kemudian di jalankan.

Misalkan kita akan coba menjalankan konfigurasi lamp simple, pertama kita pindah kedalam folder berikut.

```
cd lamp_simple
```

Kemudian seting host atau alamat ip yang akan kita gunakan pada saat saling tukar-menukar informasi. Script dapat kita lihat sebagai berikut.

```
[web]
127.0.0.1

[client]
52.221.216.43 ansible_ssh_private_key_file=/vagrant/awskeyprivate.pem
```

Setelah itu jalankan dengan ansible dengan perintah.

```
ansible-playbook -i hosts site.yml
```

Untuk file konfigurasi yang lain tidak jauh berbeda cara mengaksesnya dengan yang barusan. Kita hanya perlu melakuakn beberapa step seperti berikut.



```
cd <nama_direktori>
#cek file host ,lalu jalankan perintah seperti di bawah ini
ansible-playbook -i hosts site.yml
```

Untuk mendapatkan contoh lebih banyak mengenai ansible, kita dapat mengakses [alamat](https://docs.ansible.com/ansible/2.5/user_guide/playbooks.html) berikut https://docs.ansible.com/ansible/2.5/user_guide/playbooks.html

7.3.3. Exercise

Soal Kasus : Bayangkan kalian sedang bekerja di perusahaan Sangkuriang Corp, disana kalian memiliki 5 instance yang masih kosong dan harus di installkn nginx dan php untuk kebutuhan web server. Maka dari itu

1. Konfigurasi kelima server tersebut menggunakan layanan Ansible.

7.4. Ansible Tower

7.4.1. Mengenal Ansible Tower

Ansible Tower adalah sebuah provisioning tool yang dikembangkan oleh RedHat. Ansible memiliki tampilan visualisasi yang berguna untuk membantu dan memudahkan dalam menggunakan ansible. Ansible tower merupakan aplikasi yang berbayar, terdapat 2 paket ansible tower seperti gambar dibawah ini.

STANDARD	PREMIUM
FOR ENTERPRISE IT OPERATIONS	FOR MISSION-CRITICAL DEVOPS
8x5 support Maintenance & upgrades Enterprise features	24x7 support Maintenance & upgrades Enterprise features
Up to 100 nodes \$10,000/year	Up to 100 nodes \$14,000/year
Over 100 nodes Contact us for pricing	Over 100 nodes Contact us for pricing
View production support service level	View production support service level

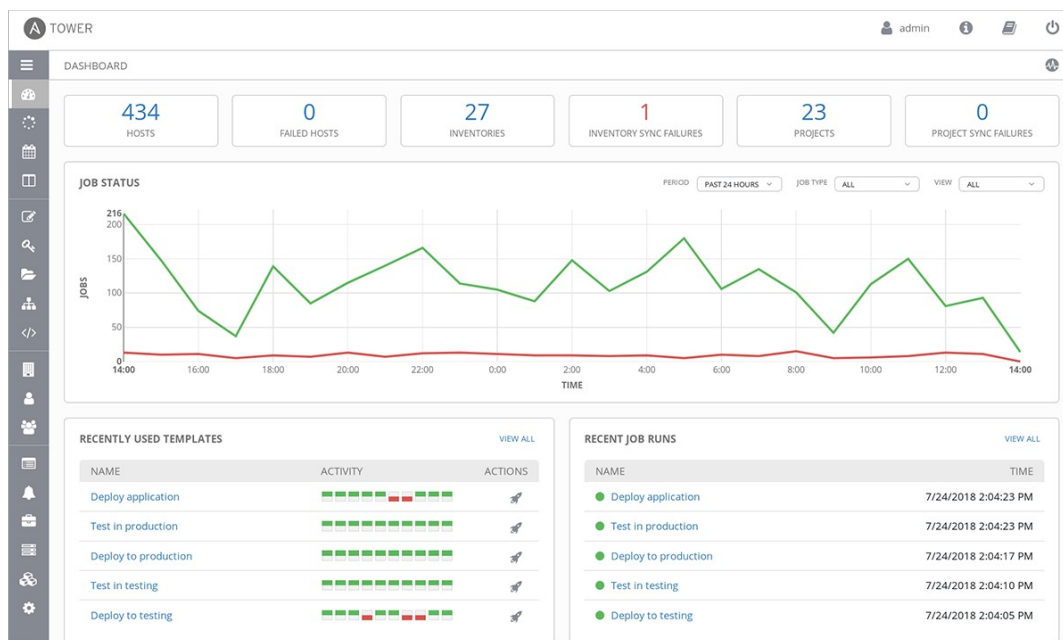
Disana terdapat kedua paket yang memiliki kelebihanannya tersendiri seperti paket standart yang memberikan beberapa support selama 6x5. Sedangkan yang premium memiliki jadwal support sampai 24x7.

7.4.2. Fitur-Fitur Ansible Tower

Ansible tower memiliki fitur yang lebih komplit dari pada ansible yang lain, hal ini di karenakan ansible tower memang berbayar, sehingga dapat dukungan penuh dari Redhat yang dapat memudahkan penggunaanya, adapun fitur-fitur ansible tower sebagai berikut.

7.4.2.1. Dasbor Tower

Ansible menyediakan tampilan yang dapat membantu pekerjaan menjadi lebih mudah, pada dasbor ansible tower kita dapat melihat host dan status inventaris kita. Kita dapat mengatur pekerjaan dan status pekerjaan dengan sebuah grafik yang dapat di monitoring secara langsung. Berikut adalah gambarannya.



7.4.2.2. Streaming Real Time

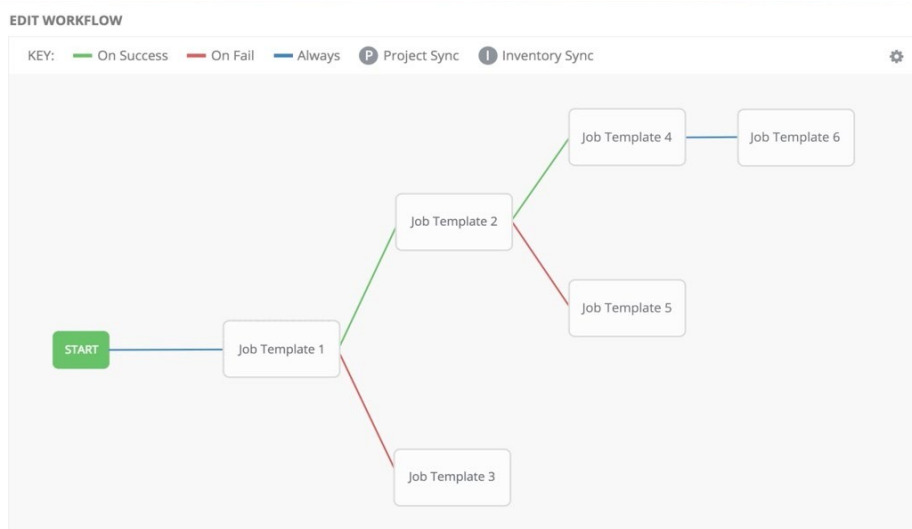
Di dalam Ansible Tower, playbook menjalankan streaming secara real time. Saat Ansible mengotomatiskan seluruh infrastruktur, kita akan melihat pekerjaan dan tugas selesai pada setiap mesin dan setiap keberhasilan atau kegagalan deploy.



The screenshot displays the Ansible Tower interface for a job titled 'Deploy software'. On the left, the 'DETAILS' panel shows the job status as 'Successful', started at 7/20/2018 6:17:31 PM, and finished at 7/20/2018 6:17:44 PM. The job type is 'Run', launched by 'admin', with inventory 'Cloud dev environment', project 'Deployment', and playbook 'deploy.yml'. Credentials include 'AWS devel keys' and 'Dev environment ssh key'. The instance group is 'tower'. On the right, the 'Deploy software' panel shows a list of tasks and their results. The tasks include 'PLAY [all]', 'TASK [Check required permissions]', 'TASK [Fix permissions if needed]', 'TASK [Deploy updates]', and 'TASK [Restart service]'. The results show that the tasks were successful, with some changes made to the system. A 'PLAY RECAP' at the bottom summarizes the results: 3 OK, 2 changed, 0 unreachable, and 0 failed.

7.4.2.3. Multi-Playbook/Multi Job

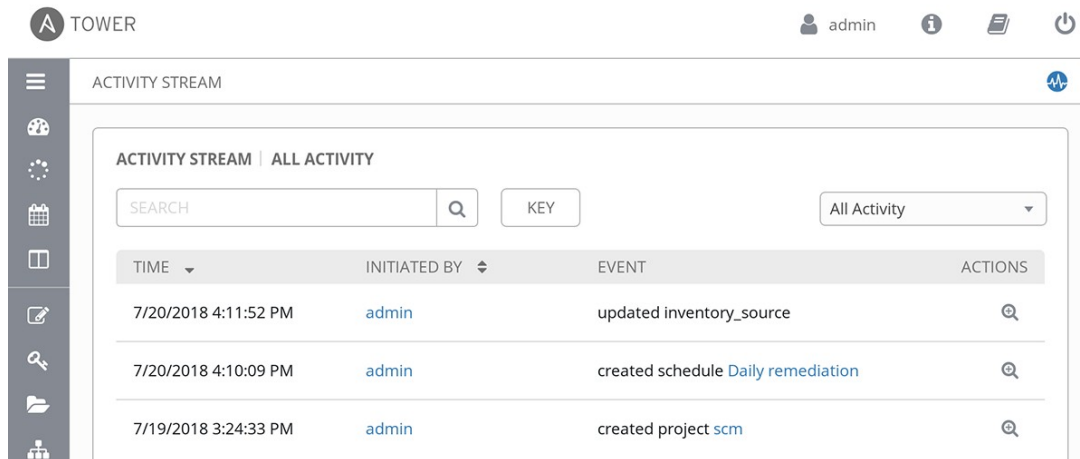
Ansible Tower dapat menjalankan multi-playbook atau multi job secara sekaligus. Alur kerja Ansible tower memungkinkan banyak proses. Kita dapat membangun kerja, menerapkan konfigurasi sistem, dan mendeploy aplikasi.



7.4.2.4. Secure

Dengan Ansible Tower, semua aktivitas otomasi disimpan dengan aman. Siapa yang menjalankannya, bagaimana mereka menyesuaikannya, apa yang dilakukannya, di mana itu terjadi semua disimpan dengan aman dan dapat dilihat kemudian, atau diekspor melalui API Ansible Tower.



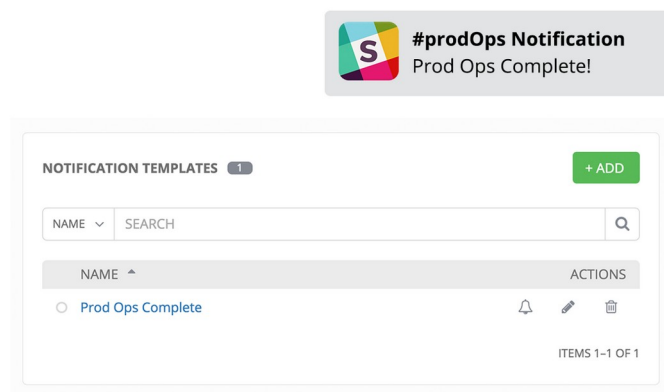


7.4.3. Exercise

1. Jelaskan menurut kamu benefit apa saja yang bisa kita dapatkan apabila menggunakan ansible tower ?

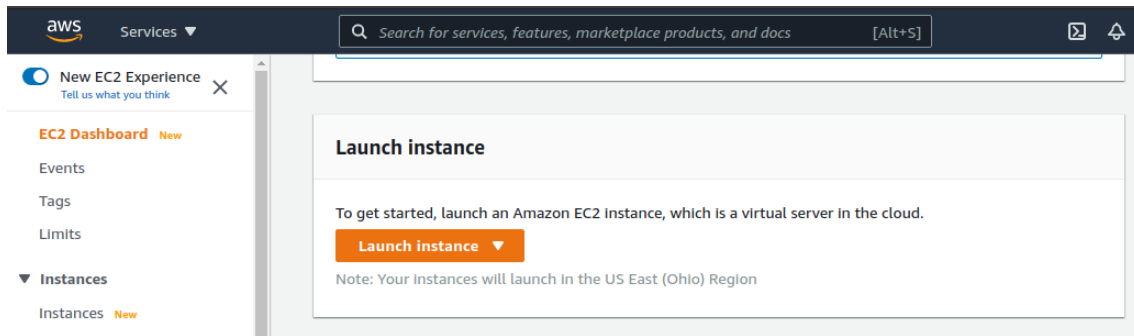
7.5. Integrated Notification

Ansible memiliki beberapa kelebihan lain yang diantaranya adalah integrasi dengan aplikasi slack. Dengan integrasinya ansible dan slack memungkinkan semua tim mendapatkan notifikasi ketika ansible dijalankan maupun berjalan.

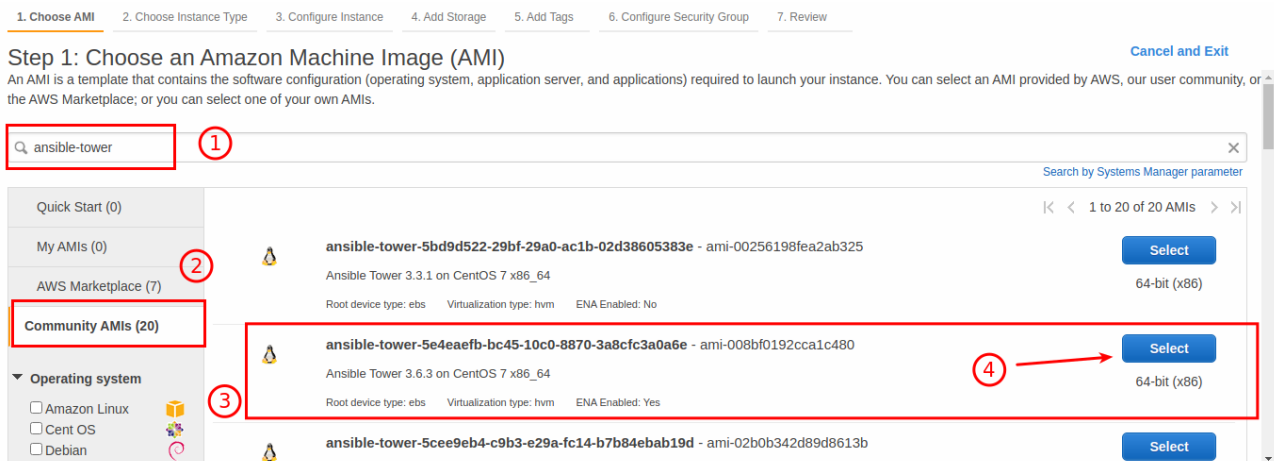


7.6. Instalasi Ansible Tower

Pada bagian ini kita akan mempelajari bagaimana cara instalasi ansible tower. Untuk tahap installasinya kita dapat mengklik launch instance di dashboard EC2 seperti di bawah ini.



Setelah itu pada bagian **search bar**, kita isikan **ansible-tower**, lalu kita klik **Community AMIs**. Kemudian pilih ansible tower dengan versi paling baru (disini v3.6.3, mungkin nanti ada pembaharuan) seperti gambar di bawah.



Selanjutnya pilih type instance yang akan kita gunakan, disini kita gunakan t2.micro.

Step 2: Choose an Instance Type

Amazon EC2 provides a wide selection of instance types optimized to fit different use cases. Instances are virtual servers that can run applications. They have varying combinations of CPU, memory, storage, and networking capacity, and give you the flexibility to choose the appropriate mix of resources for your applications. [Learn more](#) about instance types and how they can meet your computing needs.

Filter by: All instance types Current generation Show/Hide Columns

Currently selected: t2.micro (Variable ECUs, 1 vCPUs, 2.5 GHz, Intel Xeon Family, 1 GiB memory, EBS only)

	Family	Type	vCPUs	Memory (GiB)	Instance Storage (GB)	EBS-Optimized Available	Network Performance	IPv6 Support
<input type="checkbox"/>	General purpose	t2.nano	1	0.5	EBS only	-	Low to Moderate	Yes
<input checked="" type="checkbox"/>	General purpose	t2.micro	1	1	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.small	1	2	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.medium	2	4	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.large	2	8	EBS only	-	Low to Moderate	Yes
<input type="checkbox"/>	General purpose	t2.xlarge	4	16	EBS only	-	Moderate	Yes
<input type="checkbox"/>	General purpose	t2.2xlarge	8	32	EBS only	-	Moderate	Yes

Cancel Previous **Review and Launch** Next: Configure Instance Details

Kemudian lakukan konfigurasi pada storage, biarkan saja secara default. Setelah itu klik next untuk melanjutkan.

aws Services Resource Groups

1. Choose AMI 2. Choose Instance Type 3. Configure Instance 4. Add Storage 5. Add Tags 6. Configure Security Group 7. Review

Step 4: Add Storage

Your instance will be launched with the following storage device settings. You can attach additional EBS volumes and instance store volumes to your instance, or edit the settings of the root volume. You can also attach additional EBS volumes after launching an instance, but not instance store volumes. [Learn more](#) about storage options in Amazon EC2.

Volume Type	Device	Snapshot	Size (GiB)	Volume Type	IOPS	Throughput (MB/s)	Delete on Termination	Encrypted
Root	/dev/sda1	snap-0119301371ea434fe	8	General Purpose SSD (gp2)	100 / 3000	N/A	<input checked="" type="checkbox"/>	Not Encrypted

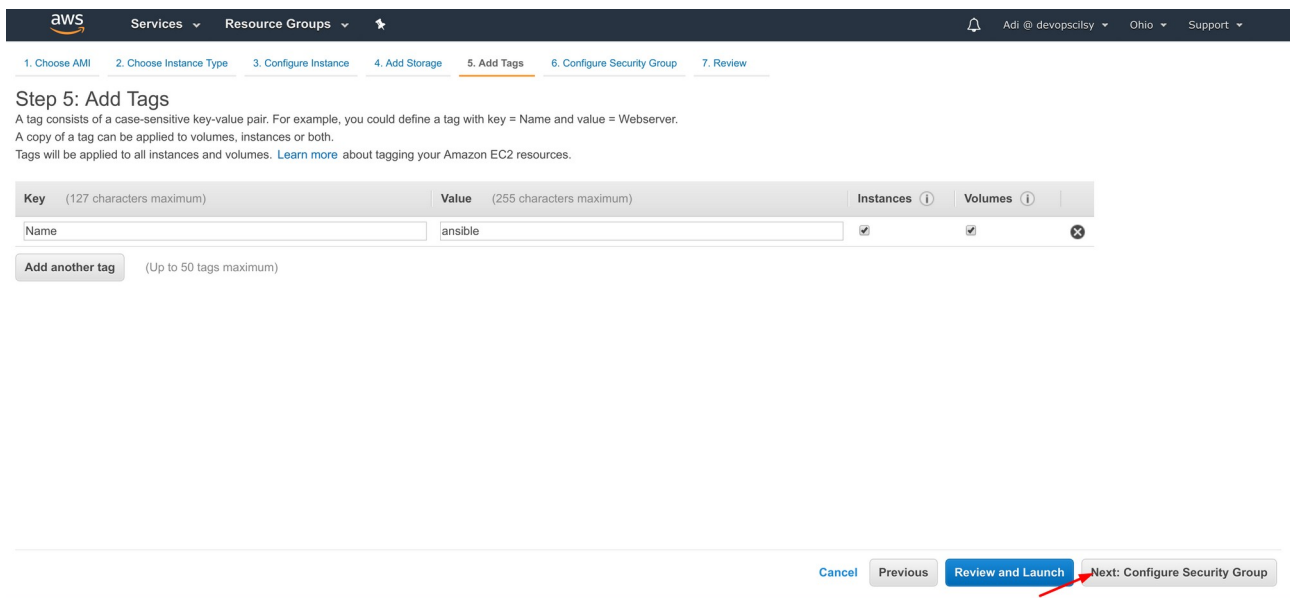
Add New Volume

Free tier eligible customers can get up to 30 GB of EBS General Purpose (SSD) or Magnetic storage. [Learn more](#) about free usage tier eligibility and usage restrictions.

Cancel Previous **Review and Launch** Next: Add Tags

Selanjutnya adalah membuat nama tags, isikan nama tag seperti gambar di bawah , dan klik Next Configuration Security Group untuk melanjutkan.





Step 5: Add Tags

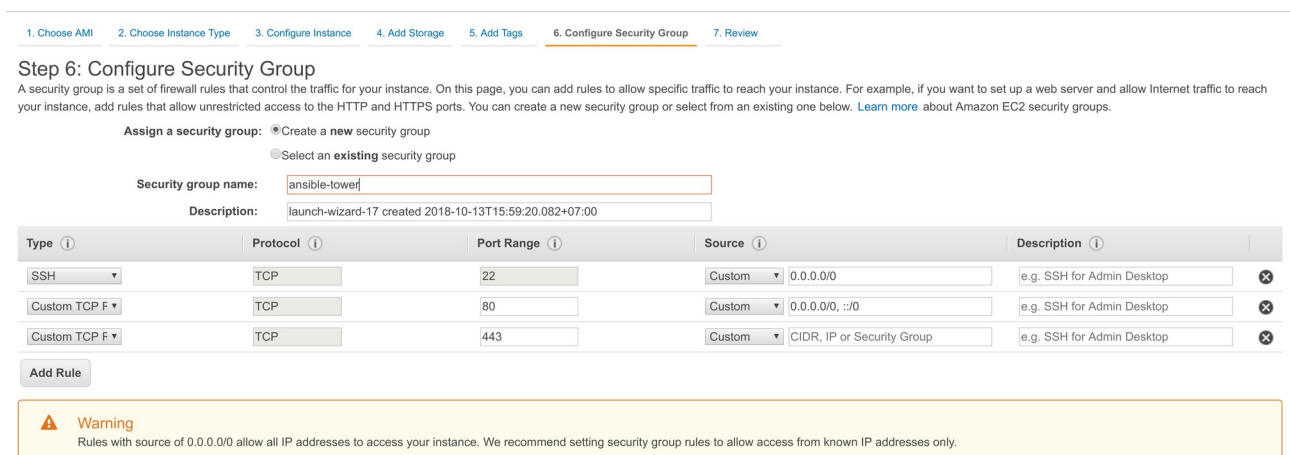
A tag consists of a case-sensitive key-value pair. For example, you could define a tag with key = Name and value = Webserver. A copy of a tag can be applied to volumes, instances or both. Tags will be applied to all instances and volumes. [Learn more](#) about tagging your Amazon EC2 resources.

Key (127 characters maximum)	Value (255 characters maximum)	Instances	Volumes
Name	ansible	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

[Add another tag](#) (Up to 50 tags maximum)

[Cancel](#) [Previous](#) [Review and Launch](#) [Next: Configure Security Group](#)

Untuk bagian security group kita isikan beberapa protocol menggunakan port 80, 443, dan 22 seperti gambar dibawah ini.



Step 6: Configure Security Group

A security group is a set of firewall rules that control the traffic for your instance. On this page, you can add rules to allow specific traffic to reach your instance. For example, if you want to set up a web server and allow Internet traffic to reach your instance, add rules that allow unrestricted access to the HTTP and HTTPS ports. You can create a new security group or select from an existing one below. [Learn more](#) about Amazon EC2 security groups.

Assign a security group: ☒ Create a new security group ☐ Select an existing security group

Security group name:

Description:

Type	Protocol	Port Range	Source	Description
SSH	TCP	22	Custom 0.0.0.0/0	e.g. SSH for Admin Desktop
Custom TCP F	TCP	80	Custom 0.0.0.0/0 ::0	e.g. SSH for Admin Desktop
Custom TCP F	TCP	443	Custom CIDR, IP or Security Group	e.g. SSH for Admin Desktop

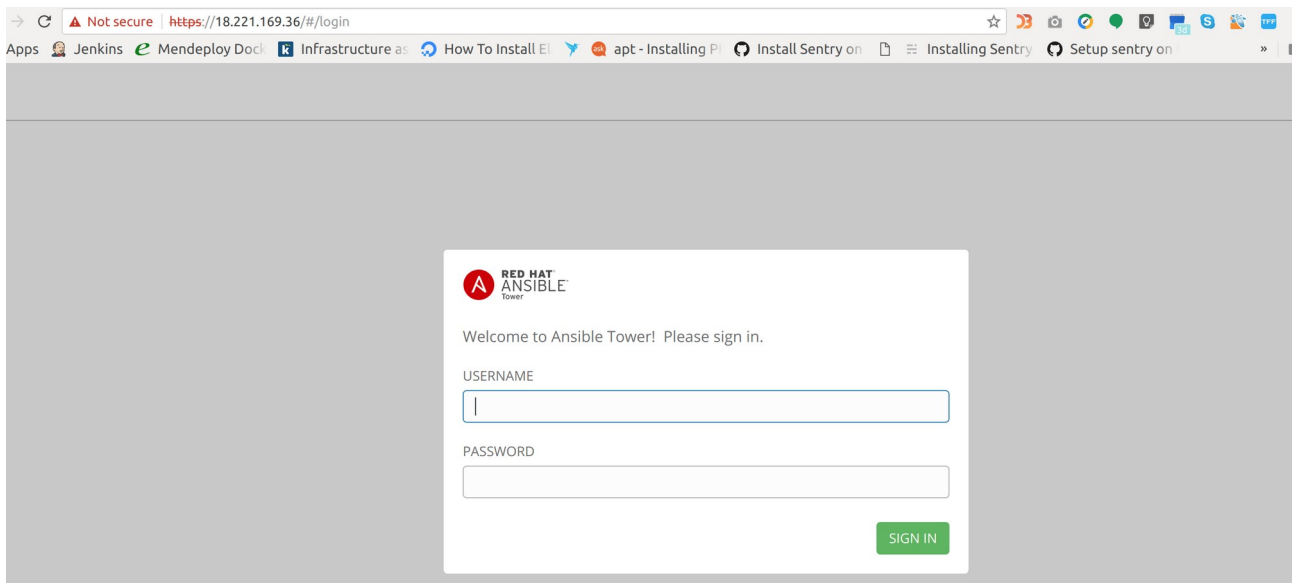
[Add Rule](#)

Warning
Rules with source of 0.0.0.0/0 allow all IP addresses to access your instance. We recommend setting security group rules to allow access from known IP addresses only.

Selanjutnya klik next hingga kita selesai membuat sebuah instance baru dengan menggunakan Ansible Tower yang berbayar.

Setelah selesai dikonfigurasi, kita dapat mengakses ansible tower dengan membuka browser dan mengetikan alamat server kita seperti di bawah ini.





Untuk mendapatkan username dan password untuk login pada web GUI Ansible, kita hanya perlu melakukan remote SSH terlebih dahulu pada server ansible tersebut. Lakukan perintah seperti gambar berikut, sehingga kita bisa mendapatkan user dan password login ansible.

```
bmduser@bmduser-UX360UAK:~/Downloads$ ssh -t keybaru1.pem centos@18.221.169.36
Last login: Mon Aug 13 16:06:55 2018 from nat-pool-rdu-t.redhat.com

Welcome to Ansible Tower!

Log into the web interface here:

https://ec2-18-221-169-36.us-east-2.compute.amazonaws.com/

Username: admin
Password: avFm66Jq8pkd

The documentation for Ansible Tower is available here:

http://www.ansible.com/tower/

For help, visit http://support.ansible.com/
[centos@ip-172-31-16-112 ~]$
[centos@ip-172-31-16-112 ~]$ sudo su

Welcome to Ansible Tower!

Log into the web interface here:

https://ec2-18-221-169-36.us-east-2.compute.amazonaws.com/

Username: admin
Password: avFm66Jq8pkd

The documentation for Ansible Tower is available here:

http://www.ansible.com/tower/

For help, visit http://support.ansible.com/
[root@ip-172-31-16-112 centos]# ^C
```



Selanjutnya kita coba lakukan login dengan menggunakan username dan password yang sudah kita dapatkan tadi. Kemudian kita akan mendapat halaman untuk request free trial seperti di bawah ini. Klik tombol Request Lisence.

TOWER LICENSE

Welcome to Ansible Tower! Please complete the steps below to acquire a license.

- 1 Please click the button below to visit Ansible's website to get a Tower license key.

REQUEST LICENSE

- 2 Choose your license file, agree to the End User License Agreement, and click submit.

* LICENSE FILE

BROWSE

No file selected.

* END USER LICENSE AGREEMENT

7. General. If any provision of this EULA is held to be unenforceable, that shall not affect the enforceability of the remaining provisions. This agreement shall be governed by the laws of the State of New York and of the United States, without regard to any conflict of laws provisions. The rights and obligations of the parties to this EULA shall not be governed by the United Nations Convention on the International Sale of Goods.

Copyright © 2015 Red Hat, Inc. All rights reserved. "Red Hat" and "Ansible Tower" are registered trademarks of Red Hat, Inc. All other trademarks are the property of their respective owners.

- ☒ I agree to the End User License Agreement

SUBMIT

Setelah request di klik, maka akan muncul tampilan dashboard seperti dibawah ini.

SELECT A FREE ANSIBLE TOWER TRIAL

✓

FREE ANSIBLE TOWER TRIAL - ENTERPRISE FEATURES

For enterprise IT operations that require more than 10 nodes

Standard and Premium Ansible Tower editions include powerful, enterprise features not available in the self-support edition:

- LDAP and Active Directory support
- System tracking
- Audit trails
- Push-button self-service with surveys
- 8x5 or 24x7 support

First Name*

Last Name*

Email*

Company Name*

Role*

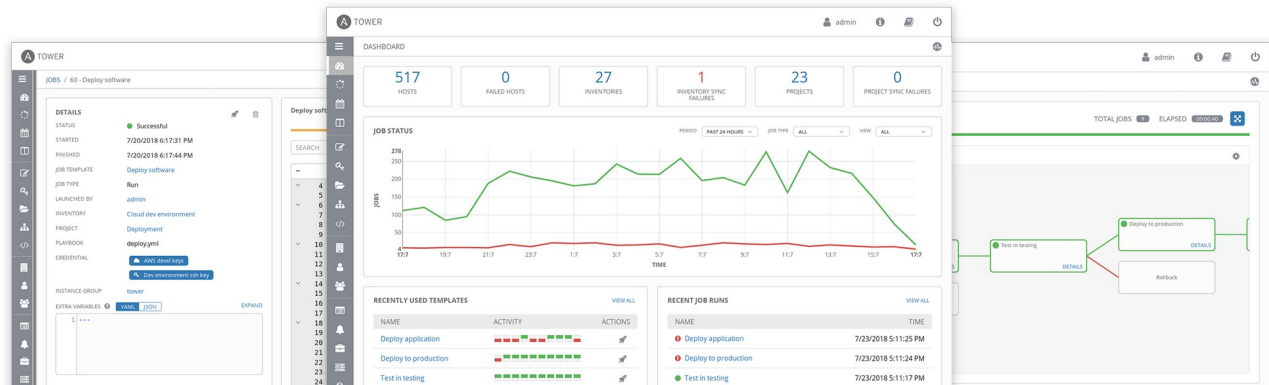
Phone Number*

of machines to manage*

SUBMIT



Jika kita sudah mendapatkan email konfirmasi, maka kita bisa ikuti link dan diarahkan ke halaman dashboard Ansible



7.6.1. Exercise

1. Coba lakukan konfigurasi pada Ansible Tower !

7.7. Summary

Pada bab ini kita sudah mempelajari mengenai ansible, bagaimana menjalankan layanan ini dengan menggunakan berbagai macam model seperti Ad-Hoc, Playbook dan Ansible Tower. Kita sudah melakukan konfigurasi ansible untuk menginstallkan nginx dan php pada beberapa server yang sudah berjalan secara sekaligus. Ansible juga dapat kita integrasikan dengan berbagai macam aplikasi yang salah satunya adalah slack.

