

# Syllabus Ansible

Tuesday, October 1, 2024 3:38 PM

## Day 1: Introduction to Ansible and Getting Started

- Overview of ansible: What is Ansible, its Advantages, and why use it?
- comparison with Shell and Python scripting for automation
- Installing Ansible on different platforms
- IDE(VS code) and plugin configuration

## Day 2: Ansible Adhoc Commands

- Passwordless Authentication
- Ansible Inventory
- Understanding Adhoc commands and their usage
- Examples of common Adhoc commands for system management tasks
- Exploring the power of Adhoc commands for quick tasks

## Day 3: Writing Your first Ansible Playbook

- Setting up Ansible configuration (including passwordless SSH Setup).
- Writing a simple playbook to automate a basic task.
- Introduction to Ansible structure: Inventory, Playbook, and Tasks.
- Hands on: Writing a playbook to install Nginx on target servers.

## Day 4: Understanding Ansible Roles

- What are Ansible roles and why use them?
- Exploring the folder structure of ansible roles.
- Comparing roles with playbooks and understanding their advantages.
- Hands on: Creating simple role and using it in a playbook.

## Day 5 : Deep dive into Ansible Roles

- Advanced usage of Ansible roles with a practical example project
- Ansible galaxy - Exploring pre-built Ansible roles
- Incorporating all the concepts learned so far into a complex project
- Best practice for organizing roles and playbook structure

## Day6: Ansible Variables and Precedence

- Understanding Ansible variables and their scope.
- Jinja2 Templating - Utilizing advanced templating features
- variable precedence: How Ansible resolves conflicts between different variable sources
- Hands- on: using variables in playbooks and roles

## Day 7: Ansible Conditionals and Loops

- Using conditionals in Ansible to control task execution
- Implementing loops for repetitive tasks.
- Practical examples of conditionals and loops in playbooks.

## Day 8: Error Handling in Ansible

- Dealing with errors and failures in Ansible Playbooks.
- Error handling techniques and best practices.
- Demonstrating error handling in practical scenarios.

## Day 9: Ansible Vault for Security

- Understanding Ansible Vault and its role in securing sensitive data.

- Encrypting and decrypting files using Ansible Vault
- Best Practices for managing secrets and sensitive data in Ansible.

Day 10: Policy as Code

Day 11: Network Automation using Ansible

Day 12: Ansible Tower Deep Dive

- Understanding Ansible Tower
- Comparison with Ansible command line and adhoc commands
- RBAC and security with Ansible Tower

Day 13: Advanced Ansible Project

- Terraform + Ansible Project

Day 14: Ansible interview Questions

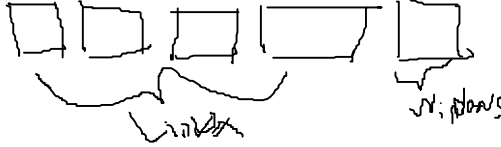
# Introduction to Ansible and Getting Started (Day 1)

Tuesday, October 1, 2024 3:36 PM

Day

Why?

10 Years -> System Admins -> salah satu activity system admin configuration management .



Manual Script

Misalkan mau upgrade versi java,, cukup menulis skrip shell.

Disinisialkan tugas dari sysadmin adalah untuk menconfigure vm dengan pengaturan sesuai perusahaan.

Misalkan sysadmin ingin configure 1 vm baru, harus memastikan

1. Up to date (supported)
2. openssh/wget/curl/shard (modul atau Library modul yang up to date)
3. ketergantungan apps pada vm, misalnya Java, javanya yang versi latest, yang di support, web server dan harus secure serta memiliki versi yang tepat
4. maintenance pada vm, pastikan ada cpu, memory

Ada banyak tools untuk automation, contohnya:

1. Puppet
2. Chef
3. Salt
4. Ansible

Puppet dan Chef itu menggunakan ruby dan agent, maka itu ribet sekali dan tidak ada waktu untuk banyak mempelajari hal tersebut.

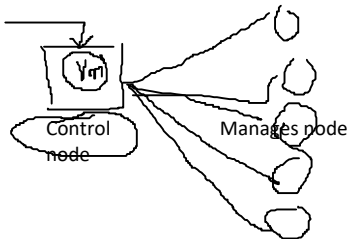
Oleh karena itu, ansible muncul.

Ansible

Yaml

You can write playbook in yaml.  
Ansible juga agentless.

Control Node  
Manage Node

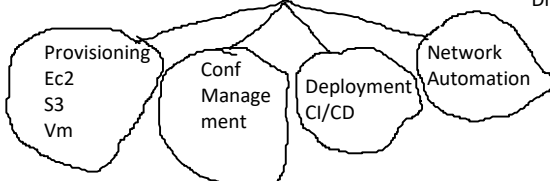


Salah satu prasyarat nya adalah install python maka ansible dapat mengontrol tiap vm.

Apakah Ansible hanya digunakan untuk configuration management?

Ansible

What?



Pada awalnya ansible hanya digunakan untuk alat automation. Akan tetapi, saat ini ansible menjadi alat yang sangat powerful.

Dimana configuration management adalah salah satu task yang dilakukan ansible.

Dalam Hal Provisioning, dapat dikatakan ansible bisa provision Ec2, Vm ware, dan infrastruktur lain

Dalam Hal Conf Manag, mengelola konfigurasi ratusan VM.

Dalam hal deployment, dapat membantu di fase CI/CD dan

Dalam hal network, bisa digunakan untuk otomasi network/ jaringan, banyak perangkat network seperti cisco, F5, Zscaler hal ini bisa di automasi dengan ansible, seperti automasi pada VLAN.

Ansible

Ansible vs shell scrip vs python

Why?

Misal kita punya shell scripting

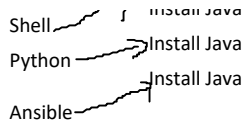
Shell -> Install Java

Python -> Install Java

... -> Install Java

Pertanyaannya adalah jika diberi tugas di organization, pendekatan mana yang akan diambil?  
-Shell sript inimanual, buang" waktu, konfigurasi seperti oracle db dll itu banyak yang perlu di configure.

-Python adalah pilihan yang tepat untuk automation, tpi ini memiliki kekurangan, diantaranya Harus mahir python programming, maintenance pythonnya, harus mengetahui modul" python yang sangat banyak dll



Kapan perlu menggunakan python over ansible?

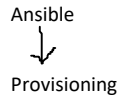
Ketika ada tugas seperti berbicara dengan API, katakanlah untuk create JIRA tiket, create Github melalui API. Karena ini tidak ada module ansible. Jika untuk provisioning di AWS, Azure, dll yang didukung module ansible, maka gunakan ansible.

configure.

-Python adalah pilihan yang tepat untuk automation, tpi ini memiliki kekurangan, diantaranya Harus mahir python programming, maintenance pythonnya, harus mengetahui modul" python yang sangat banyak, dll

- Ansible, Ansible menjawab kesulitan semua itu, yang mana hanya diperlukan instal ansible pada control node, dan hanya perlu mengetahui YAML.

Selain itu, ada juga ansible documentation untuk IaC di YAML, oleh karena itu, Ansible diperlukan.



Ketika diminta ansible untuk provisioning, kita harus melihat tools lain yang lebih powerfull untuk provisioning, yaitu terraform. Dan kesampingkan ansible sebagai provisioning, tpi sebagai konfigurasi management. Namun, ketika perusahaan mengharuskan untuk memakai ansible untuk everything, maka gunakan ansible. Katakanlah perusahaan telah subcription Ansible, maka gunakan ansible. Karena mendapat support secara penuh dari redhat.

How AnsibleWork Internally, secara internal persyaratannya adalah pada Node control, harus menginstall python dan juga pada manage node harus menginstall python. Why python? karena apapun file yaml yang ditulis -> diambil ansible, ansible mengambil file yaml, dikonversi ke python dan mengeksekusi modul python pada manage node.

Lets go practice

Install Ansible

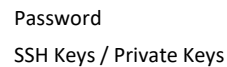
Using visual studio code  
Install extension

-Yaml  
-Ansible

-

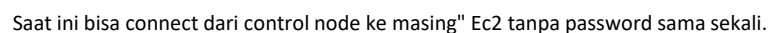
Wednesday, October 2, 2024 2:03 PM

## Passwordless Authentication



A diagram showing a direct connection between a Laptop and a VM. A box labeled 'Laptop' is connected by a straight arrow to a box labeled 'VM'.

Ada suatu case



# Membuat Keygen SSH

Thursday, October 3, 2024 10:48 AM

Yang harus dilakukan pertama kali adalah membuat ssh-keygen di server workstation.

Membuat keygen

```
ssh-keygen -t ed25519 -C "hilmy default"
```

Copy keygen tersebut ke server tujuan

```
ssh-copy-id -i ~/.ssh/id_ed25519.pub 10.120.110.28
```

Namun, bagaimana jika kita membuat keygen khusus untuk ansible saja?

```
ssh-keygen -t ed25519 -C "ansible"
```

Enter file masukkan dengan nama ansible seperti contoh agar tidak menimpa keygen yang sebelumnya.

Jangan masukkan passwordnya pada vm.

```
[root@ansible ~]# ssh-keygen -t ed25519 -C "ansible"
Generating public/private ed25519 key pair.
Enter file in which to save the key (/root/.ssh/id_ed25519): /root/.ssh/ansible
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/ansible.
Your public key has been saved in /root/.ssh/ansible.pub.
The key fingerprint is:
SHA256:7uRfR0lTwut7pgXZwJdwH99F34KUITFMn4Fbc6srcJc ansible
The key's randomart image is:
+--[ED25519 256]--+
|      o=o+=.++|
|      o+===oX|
|      o+o*+B|
|      . .o0|
|      S   += .|
|      .. . E...|
|      oo .....|
|      + .....o|
|      o...  .+|
+-----[SHA256]-----+
[root@ansible ~]#
```

# Cara pintas cepat membuat keygen

Thursday, October 3, 2024 11:34 AM

Pada server tujuan pastikan PasswordAuthentication yes

Membuat ssh-keygen

```
ssh-keygen -t ed25519 -C "hilmy default"
```

Tidak usah dikasih passphrase code

Langsung copy keygen tersebut ke server tujuan

```
ssh-copy-id root@10.120.110.28
```

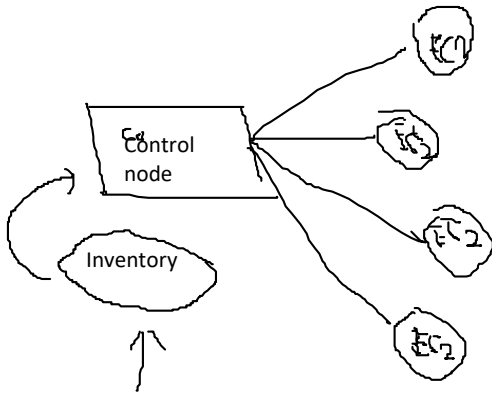
```
ssh-copy-id root@10.120.110.30
```

Atau bisa juga dengan membuat user terlebih dahulu

```
ssh-copy-id ansible@10.120.110.28
```

# Ansible Inventory

Thursday, October 3, 2024 11:43 AM



Ibarat kita mengetahui flow dan cara kerja antara control node dan manage. Tpi secara sistem, bagaimana control node mengetahui setiap node yang dia kontrol atau manage node?

Kuncinya ada pada inventory file,

Inventory bisa dalam 2 format

1. inventory.ini

2. inventory dalam yaml

Inventory.ini, .ini disitu hanya ekstension.

Ada 2 cara dalam mengeksekusi task di ansible, yang pertama adalah dengan

1.playbooks [YAML]

2.adhoc commands

Adhoc command diperuntukan untuk simple task seperti install apache, create file, reboot, dll

Namun jika memerlukan pengaturan yang kompleks seperti install database oracle, install konfigurasi dan itu reusable, Git, cdllab, modular/roles.

Contoh penggunaan ansible

```
[root@ansible ansible]# ansible -i inventory.ini -m shell -a "sudo mkdir /home/coba-ansible" app
root@10.120.110.28 | CHANGED | rc=0 >>

[root@ansible ansible]# ansible -i inventory.ini -m shell -a "sudo reboot" app
root@10.120.110.28 | UNREACHABLE! => {
  "changed": false,
  "msg": "Failed to connect to the host via ssh: Shared connection to 10.120.110.28 closed.",
  "unreachable": true
}
```

ansible -i inventory.ini -m shell -a "sudo mkdir /home/coba-ansible" app

ansible merupakan task untuk ansible

-i inventory.ini merupakan inventory untuk host nya

-m shell merupakan module shell di ansible untuk menjalankan perintah shell

-a "sudo reboot" merupakan atribut untuk menjalankan perintah

app merupakan nama group dalam inventory.ini



# Modul-modul adhoc command

Thursday, October 3, 2024 4:13 PM

[https://docs.ansible.com/ansible/latest/command\\_guide/intro\\_adhoc.html#rebooting-servers](https://docs.ansible.com/ansible/latest/command_guide/intro_adhoc.html#rebooting-servers)

# Day 3 Write Your Ansible playbook

Thursday, October 3, 2024 4:43 PM

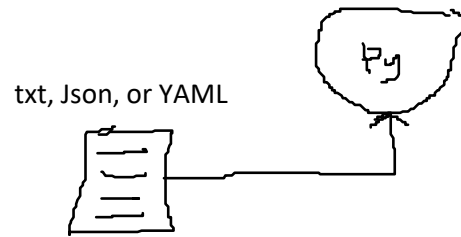
- Understanding YAML basics and ansible playbooks structure.
- Introduction to Ansible structure: playbook, play, modules, tasks, and collections
- Handson: Writing a playbook to install apache2 and deplou a static app on aws

# Yaml

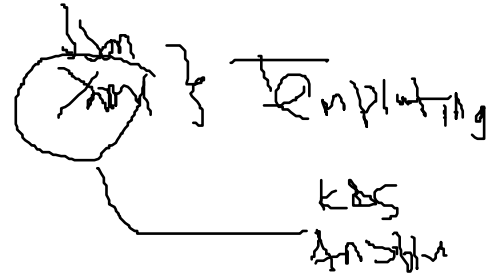
Friday, October 4, 2024 2:32 PM



Data serialization yang dapat dibaca manusia.

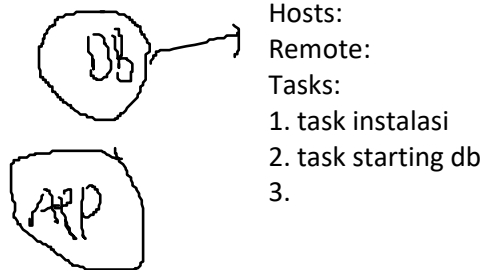


Kenapa Yaml tidak json? karena lebih mudh dibaca manusia



Ansible Playbook itu List of plays

contohnya



Playbook = list of plays

play 1 play 2 dalam play terdapat

1. Hosts:
2. remote-user:
3. Tasks:
  - module
  - module
  - module

contoh:

jika ingin install postgres dan start service postgresql

tasks:

-name: Ensure postgresql is at the latest version

ansible.builtin.yum:

name: postgresql

state: latest

-name: Ensure that postgresql is started

ansible.builtin.service

name: postgresql

state: started

Contoh pada modul 3 di github

---

```
- hosts: all (Hanya 1 host)
  become: true (eksekusi menggunakan root user)
  tasks:
    - name: Install apache httpd
      ansible.builtin.apt:
        name: apache2
        state: present
        update_cache: yes
    - name: Copy file with owner and permissions
      ansible.builtin.copy:
        src: index.html
        dest: /var/www/html
        owner: root
        group: root
        mode: '0644'
```

# Eksekusi yaml

Wednesday, October 9, 2024 10:44 AM

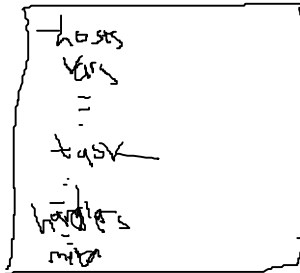
ansible-playbook -i inventory.ini first-playbook.yaml

# Day 4 Ansible Role

Wednesday, October 9, 2024 10:24 AM

- What are ansible roles and why use them?
- Exploring the folder structure of Ansible roles.
- Comparing roles with playbooks and understanding their advantages.
- Hands-on: Creating a simple role and using it in a playbook.

## Roles



Pada days 3 kemarin, kita menjalankan playbook dengan hanya menggunakan 2 tasks. bayangkan jika kita memiliki puluhan task dan ada puluhan vars.

Kita harus memerhatikan konsep readability dan modularity. untuk memenuhi konsep tersebut, maka diperlukan sebuah konsep yang dinamakan ansible roles.

Ansible Roles adalah cara untuk mengatur dan mengelompokkan tugas-tugas (tasks), variable-variable, file, template, dan handler dalam ansible untuk mempermudah pengelolaan playbooks yang kompleks. Dengan roles, konfigurasi server atau aplikasi dapat dipecah menjadi unit-unit yang lebih modular, terstruktur, dan dapat digunakan kembali di berbagai playbook.

Secara singkat, roles memungkinkan untuk:

- membagi playbook menjadi bagian-bagian yang lebih kecil dan mudah dipahami.
- mengelompokkan tugas-tugas tertentu berdasarkan fungsi atau komponen yang dikonfigurasi.
- memisahkan konfigurasi dari data (variabel).
- meningkatkan keterbacaan, keteraturan, dan kemudahan dalam pemeliharaan.

## Struktur ansible roles

Sebuah role memiliki struktur direktori yang khas dan terstandarisasi. Contoh struktur role sederhana adalah sebagai berikut:

```
css
Copy code

roles/
├── nama_role/
│   ├── tasks/
│   │   └── main.yml
│   ├── handlers/
│   │   └── main.yml
│   ├── templates/
│   ├── files/
│   ├── vars/
│   │   └── main.yml
│   ├── defaults/
│   │   └── main.yml
│   ├── meta/
│   │   └── main.yml
│   ├── tests/
│   └── README.md
```

Penjelasan dari setiap direktori:

- **tasks/**: Berisi daftar tugas yang akan dijalankan oleh role, biasanya dimulai dari main.yml.
- **handlers/**: Berisi handler, yang dipanggil oleh tasks ketika kondisi tertentu terpenuhi.
- **templates/**: Berisi file template Jinja2 yang bisa digunakan dalam tasks untuk menghasilkan konfigurasi dinamis.
- **files/**: Berisi file-file statis yang mungkin perlu disalin ke host yang dikelola.
- **vars/**: Berisi variabel-variabel yang digunakan dalam role.
- **defaults/**: Berisi variabel default yang memiliki prioritas lebih rendah (dapat diubah oleh variabel lain).
- **meta/**: Berisi metadata tentang role seperti dependencies (ketergantungan) role lain.
- **tests/**: Berisi pengujian atau playbook testing untuk role ini.

## Keuntungan Menggunakan Ansible Roles

1. **Modular dan Dapat Digunakan Kembali**: Role dapat digunakan kembali di banyak playbook tanpa perlu menulis ulang tugas-tugas.
2. **Struktur yang Terorganisir**: Dengan mengikuti struktur standar, memelihara konfigurasi dan playbook menjadi lebih mudah.
3. **Kolaborasi Lebih Mudah**: Pembagian tugas ke dalam roles membuat tim lebih mudah untuk bekerja pada bagian-bagian yang terpisah dari playbook secara paralel.
4. **Pemeliharaan Lebih Mudah**: Dengan mengisolasi tugas ke dalam roles, perubahan atau pembaruan lebih mudah dikelola tanpa memengaruhi seluruh playbook.

## Contoh Penggunaan Role dalam Playbook

```
yaml Copy code  
  
---  
- hosts: webservers  
  roles:  
    - nginx  
    - php
```

Dalam contoh ini, playbook akan menggunakan role `nginx` dan `php`, yang berarti semua tugas, handler, file, atau template dari role tersebut akan dijalankan pada host `webservers`.

## Kesimpulan

Ansible roles merupakan pendekatan modular untuk membangun playbook yang terstruktur dan mudah dipelihara, terutama ketika proyek atau konfigurasi menjadi semakin kompleks.

# Sekilas Ansible Galaxy

Wednesday, October 9, 2024 11:15 AM

**Ansible Galaxy** adalah ekosistem dan command-line tool yang digunakan dalam Ansible untuk menemukan, mengunduh, mengelola, serta berbagi *roles* dan *collections* yang dibuat oleh komunitas atau tim Ansible. Tujuan dari Ansible Galaxy adalah mempermudah pengguna dalam menggunakan role dan collection yang sudah ada, tanpa harus menulis dari awal. Dengan demikian, Anda bisa mempercepat proses otomatisasi, meningkatkan produktivitas, dan memastikan konsistensi dalam infrastruktur yang Anda kelola.

## Komponen Ansible Galaxy

### 1. Ansible Galaxy Website ([galaxy.ansible.com](https://galaxy.ansible.com)):

- Ini adalah portal web tempat Anda dapat menemukan dan berbagi role atau collection yang sudah dibuat oleh pengguna Ansible lainnya. Di sini, Anda dapat mencari role yang sesuai untuk tugas tertentu seperti menginstal Nginx, mengelola pengguna, mengkonfigurasi firewall, dan lain-lain.
- Anda juga dapat mengunggah role atau collection yang Anda buat sendiri agar orang lain bisa menggunakannya.

### 2. Ansible Galaxy Command Line Tool:

- Ansible Galaxy juga merupakan *tool* berbasis command-line yang memungkinkan pengguna untuk melakukan beberapa operasi terkait *roles* dan *collections* langsung dari terminal. Misalnya, Anda dapat mengunduh role dari Galaxy, membuat role baru, atau memeriksa dependensi role.
- Beberapa perintah utama yang sering digunakan adalah `ansible-galaxy install`, `ansible-galaxy init`, `ansible-galaxy list`, dan `ansible-galaxy remove`.

## Fungsi dan Perintah Penting dalam Ansible Galaxy

### 1. Install Role atau Collection

- Untuk mengunduh dan menginstal role atau collection yang ada di Galaxy, Anda dapat menggunakan perintah berikut:

```
bash
Copy code
ansible-galaxy install [nama_role]
```

Contoh:

```
bash
Copy code
ansible-galaxy install geerlingguy.nginx
```

Perintah ini akan mengunduh role `nginx` dari pengguna `geerlingguy` ke dalam direktori default `roles/`. Role tersebut dapat langsung digunakan di `playbook`.

### 2. Install dari Requirements File

- Anda juga dapat menginstal beberapa role sekaligus dengan membuat file `requirements.yml` yang berisi daftar role atau collection yang ingin diinstal.
- Contoh isi `requirements.yml`:

```
yml
Copy code
- src: geerlingguy.nginx
- src: git+https://github.com/someuser/somerole.git
```

- Kemudian, Anda menginstalnya dengan perintah:

```
bash
Copy code
ansible-galaxy install -r requirements.yml
```

### 3. Membuat Role Baru

- Untuk membuat struktur role baru yang siap digunakan, Anda bisa menggunakan perintah:

```
bash
Copy code
ansible-galaxy init [nama_role]
```

Contoh:

```
bash
Copy code
ansible-galaxy init my_new_role
```

Ini akan membuat struktur direktori standar seperti `tasks/`, `handlers/`, `vars/`, dll., yang sudah siap untuk dikonfigurasi lebih lanjut.

### 4. Mengelola Collections

- Selain role, Ansible juga memiliki *collections*, yaitu kumpulan dari beberapa role, modul, plugin, dan `playbook` yang dikemas bersama.
- Untuk menginstal sebuah collection, gunakan perintah:

```
bash
Copy code
ansible-galaxy collection install [nama_collection]
```

Contoh:



```
bash
Copy code
ansible-galaxy collection install ansible.posix
```

#### 5. Mengecek Role atau Collection yang Terinstal

- o Untuk melihat role atau collection yang sudah terinstal, Anda bisa menjalankan:

```
bash
Copy code
ansible-galaxy list
```

Ini akan menampilkan daftar semua role atau collection yang tersedia di sistem lokal Anda.

#### 6. Menghapus Role atau Collection

- o Jika ingin menghapus role atau collection, gunakan:

```
bash
Copy code
ansible-galaxy remove [nama_role]
```

atau

```
bash
Copy code
ansible-galaxy collection remove [nama_collection]
```

### Perbedaan antara Role dan Collection

- **Role:** Sebuah unit modular yang biasanya mencakup *tasks*, *handlers*, *files*, *templates*, *variables*, dll. Roles digunakan untuk mengotomatisasi tugas-tugas spesifik di dalam playbook Ansible.
- **Collection:** Sebuah paket yang lebih besar dan komprehensif yang bisa mencakup banyak role, serta tambahan komponen lain seperti modul, plugin, dan playbook. Collection lebih fleksibel dan bisa memuat lebih banyak fitur daripada role tunggal.

### Manfaat Ansible Galaxy

1. **Menghemat Waktu dan Usaha:** Anda tidak perlu menulis dari nol untuk konfigurasi yang umum. Misalnya, Anda bisa langsung menginstal role untuk mengkonfigurasi Nginx atau Apache tanpa harus membuat semuanya sendiri.
2. **Kolaborasi Komunitas:** Galaxy adalah tempat berkumpulnya komunitas Ansible yang besar. Pengguna dari seluruh dunia dapat berbagi role dan collection yang mereka buat, sehingga Anda bisa memanfaatkan karya orang lain yang sudah diuji.
3. **Pembaruan Mudah:** Role atau collection yang Anda instal dari Galaxy bisa diperbarui dengan mudah jika ada versi baru yang dirilis.
4. **Konsistensi dan Standarisasi:** Dengan menggunakan role atau collection dari Galaxy, Anda dapat menerapkan standarisasi dalam konfigurasi infrastruktur Anda, yang akan membuat pengelolaan lebih konsisten dan teratur.

### Contoh Kasus Penggunaan

Misalnya, Anda ingin mengotomatisasi instalasi dan konfigurasi Nginx pada banyak server. Alih-alih menulis semua *tasks* dari awal, Anda bisa mencari di Ansible Galaxy dan menemukan role seperti `geerlingguy.nginx`, yang sudah digunakan oleh banyak orang dan dioptimalkan untuk kebutuhan umum.

Anda tinggal menginstalnya dengan:

```
bash
Copy code
ansible-galaxy install geerlingguy.nginx
Kemudian gunakan role ini di playbook Anda:
```

```
yaml
Copy code
---
- hosts: webservers
  roles:
    - geerlingguy.nginx
```

Ini akan langsung mengonfigurasi Nginx pada semua server yang tercakup di grup `webservers`.

### Kesimpulan

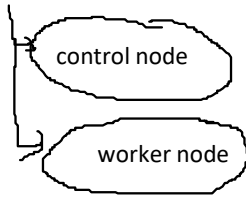
**Ansible Galaxy** adalah ekosistem dan tool yang sangat bermanfaat dalam mengelola *roles* dan *collections* di Ansible. Ini memberikan akses mudah ke modul-modul yang sudah dibuat oleh komunitas, sehingga memudahkan pengguna untuk mengotomatisasi tugas-tugas di infrastruktur mereka tanpa perlu memulai dari awal.

# Lab Ansible Role

Wednesday, October 9, 2024

11:48 AM

ansible-galaxy role init test  
Kubernetes



Pada control node mungkin diperlukan untuk setup dan configure seperti api server, etcd, container runtime, container network,

Pada worker node mungkin diperlukan untuk setup and configure kubelet, how worker node bisa join ke control node,

jadi secara keseluruhan playbook ini seperti yang telah di jelaskan, bisa setup 50 sampai 60 task, itu baru hanya kubernetes, belum lagi jika install dan configure database oracle, diperlukan setup and configuration yang cukup rumit juga.

dengan menggunakan ansible role, dapat membuat playbook menjadi lebih mudah dibaca, lebih modular, serta lebih mudah dan simple lagi menggunakan ansible galaxy

Bagaimana jika ingin menjadikan role dari playbook yang sudah di jalankan kemarin?  
on github ansible-role

## Day 5 Ansible-Galaxy/ Ansible role Deep dive

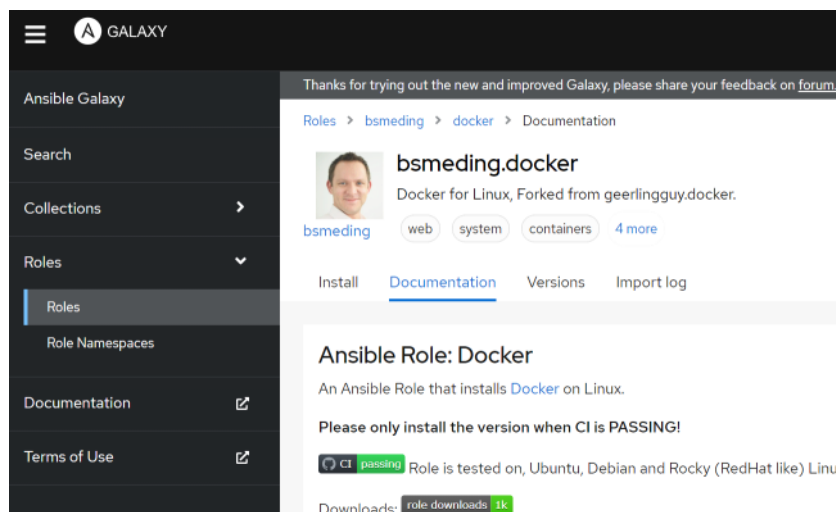
Wednesday, October 9, 2024 5:27 PM

Misalkan diberikan task untuk install docker dan konfigurasi docker dengan ansible.

```
Register this system with Red Hat Insights: insights-client --register
Create an account or view all your systems at https://red.ht/insights-dashboard
Last login: Thu Oct 10 11:15:27 2024 from 10.151.45.23
[root@ansible ~]# ansible-galaxy role -h
usage: ansible-galaxy role [-h] ROLE_ACTION ...

positional arguments:
  ROLE_ACTION
    init      Initialize new role with the base structure of a role.
    remove    Delete roles from roles_path.
    delete   Removes the role from Galaxy. It does not remove or alter the actual GitHub repository.
    list      Show the name and version of each role installed in the roles_path.
    search    Search the Galaxy database by tags, platforms, author and multiple keywords.
    import    Import a role into a galaxy server
    setup     Manage the integration between Galaxy and the given source.
    info      View more details about a specific role.
    install   Install role(s) from file(s), URL(s) or Ansible Galaxy

options:
  -h, --help  show this help message and exit
[root@ansible ~]#
```

The screenshot shows the Ansible Galaxy web interface. On the left is a dark sidebar with navigation links: 'Ansible Galaxy', 'Search', 'Collections', 'Roles' (selected), 'Role Namespaces', 'Documentation', and 'Terms of Use'. The main content area has a header with a notification about feedback. Below it is a breadcrumb trail: 'Roles > bsmeding > docker > Documentation'. The role 'bsmeding.docker' is displayed, with a description 'Docker for Linux, Forked from geerlingguy.docker.' and tags 'web', 'system', 'containers', and '4 more'. There are tabs for 'Install', 'Documentation' (active), 'Versions', and 'Import log'. The 'Documentation' tab shows the role's details: 'Ansible Role: Docker', 'An Ansible Role that installs Docker on Linux.', and a warning 'Please only install the version when CI is PASSING!'. A status bar indicates 'CI [passing]' and 'Role is tested on, Ubuntu, Debian and Rocky (RedHat like) Linux'. At the bottom, it shows 'Downloads: role downloads 1k'.

Seperti kemarin, kemarin kita membuat sebuah role dengan httpd, yang mana itu adalah role yang dibuat dari ansible galaxy, yang menciptakan struktur folder didalamnya yang lengkap dengan semua file dasar yang diperlukan. (ansible init)

ansible-galaxy role install bsmeding.docker --ignore-certs

Hasil copy nya tersedia di

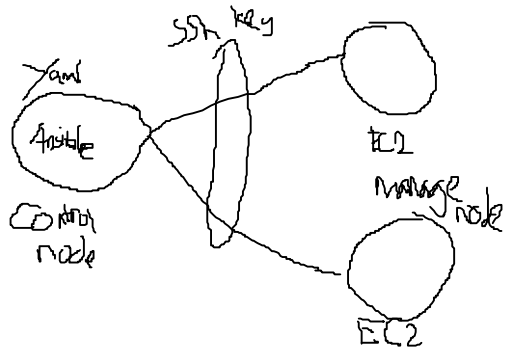
/root/.ansible/roles/bsmeding.docker

cara push role yang sudah dibuat ke ansible galaxy

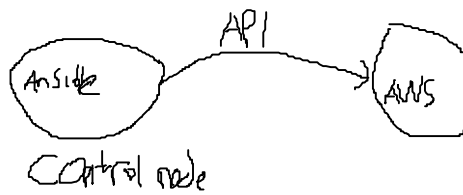
ansible-galaxy import muhammadhilly9 (repository github)

## Day 6 Ansible Variables and Precedence

Friday, October 11, 2024 10:42 AM



Pada bab sebelumnya, kita menggunakan ansible untuk talk dengan virtual machine.



Pada bab ini karena kita ingin provisioning, maka kita akan talk dengan api aws. kemudian, akan membuat beberapa resources di aws. bisa jadi ec2, s3 bucket, dan bukan hanya aws, bisa di azure, gcp atau network seperti cisco.

Jadi yang harus dilakukan adalah install modul ansible pada control node. Jadi ketika modul ini diinstall, maka dia akan berbicara dengan API karena modul ini pada dasarnya adalah python code.

Jadi ansible akan execute code python di control node, kemudian python ini berbicara dengan api dari aws atau berbicara dengan api cisco atau azure, gcp atau apapun.

sehingga ada konsep yang dinamakan ansible collection.

Lengkap detail ada di dokumentasi ansible

# Demo create resource on aws

Friday, October 11, 2024 2:47 PM

Pertama-tama, kita harus install ansible collection awsnya:

ansible collection aws dapat didapatkan dari ansible galaxy -> collection -> cari amazon

- ansible-galaxy collection install amazon.aws

sebagaimana dijelaskan sebelumnya, untuk akses API aws diperlukan python, python tersebut membutuhkan sebuah module python yang bernama boto3.

Maka didalam control node, harus terinstall boto3

- pip install boto3

# Ansible with GCP

Wednesday, October 16, 2024 4:26 PM

<https://cloud.google.com/compute/docs/connect/standard-ssh#openssh-client>

# Note Command Ansible

Thursday, October 10, 2024 2:56 PM

Pada server tujuan pastikan PasswordAuthentication yes

Membuat ssh-keygen

```
ssh-keygen -t ed25519 -C "hilmy default"
```

Tidak usah dikasih passphrase code

Langsung copy keygen tersebut ke server tujuan

```
ssh-copy-id root@10.120.110.28
```

```
ssh-copy-id root@10.120.110.30
```

Atau bisa juga dengan membuat user terlebih dahulu

```
ssh-copy-id ansible@10.120.110.28
```

ad hoc command ansible dengan module shell:

```
- ansible -i inventory.ini -m shell -a "sudo mkdir /home/coba-coba-ansible" web (web disini adalah grouping pada inventory.ini)
```

```
- ansible -i inventory.ini -m shell -a "sudo reboot" app
```

running playbook:

```
- ansible-playbook -i inventory.ini first-playbook.yaml
```

Install ansible galaxy dari galaxy server:

```
- ansible-galaxy role install bsmeding.docker --ignore-certs
```

membuat role apache dengan ansible galaxy:

```
- ansible-galaxy role init httpd
```

push role ke ansible galaxy server milik kita (wajib di push terlebih dahulu ke github)

```
- ansible-galaxy import muhammadhilmy9 (nama-repository) --token 21f73155c2a63444270ec8fc2478105c59787447
```

cara dapatin tokennya: masuk ke menu collection -> API Token -> load token copy

```
curl -vvvk http://10.120.110.28:80
```

# Etc konfigurasi

Thursday, October 10, 2024 3:10 PM

10.120.110.20

Root

P@ssw0rdhil

Cara cek repository

yum list | grep -i ansible

Github token

ghp\_fOPL5HI0kYM2MpAV3uJh5W34IN94702CHa0f

Clone github via https

git config --global http.sslVerify false

Ansible-galaxy ssl

sudo update-ca-certificates --fresh

export SSL\_CERT\_DIR=/etc/ssl/certs

Fixed myself bt adding --ignore-certs

export https\_proxy=proxy.danamon.co.id:8080

export http\_proxy=proxy.danamon.co.id:8080

export no\_proxy=10.197.30.240

install boto3

pip3 install --trusted-host pypi.org --trusted-host pypi.python.org --trusted-host files.pythonhosted.org  
boto3

Access Key & Secret Key Aws user ansible

Access Key

AKIA6PRGD6DR3RZ5H5W5

Secret Key

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



# Arsitektur & Others

Tuesday, October 15, 2024 10:35 AM

