IDN.ID
INDONESIAN EXPERT FACTORY

Online Course

Ansible Basic Course

Trainer | Devops Consultant | Cloud Consultant
Moch Rafi Riadi

visit now lms.idn.id

Outline

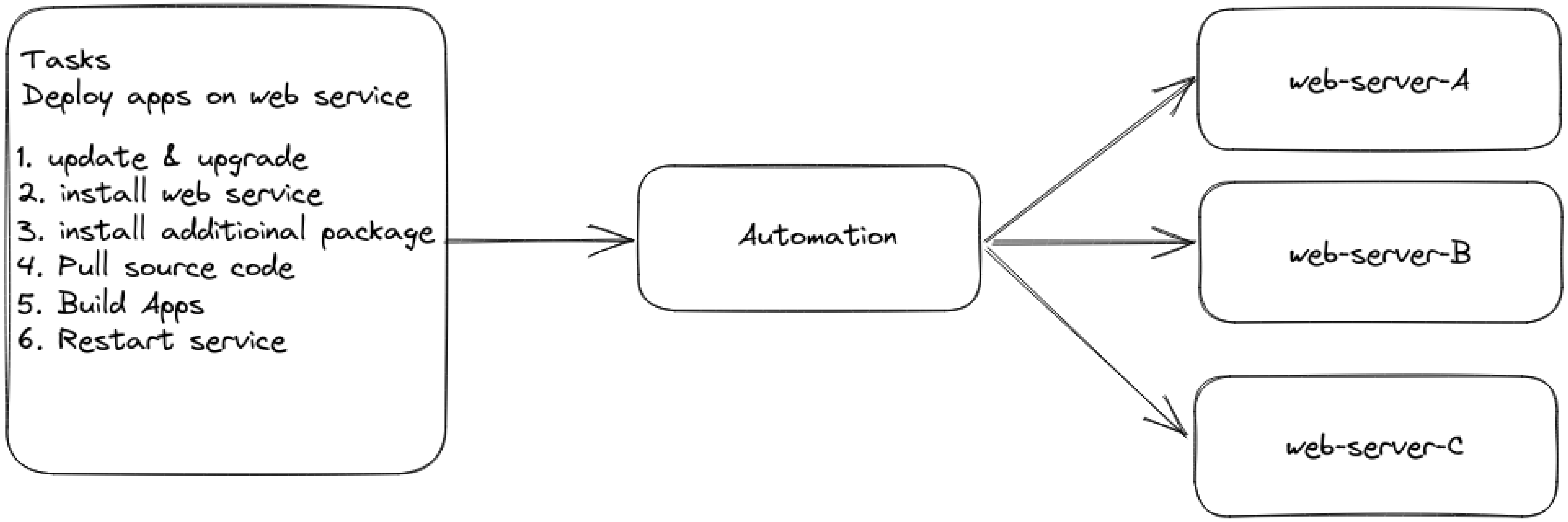
- Introduction Ansible
- Install Ansible & Setup Lab
- Introduction to YAML
- Inventory Files
- Ansible ad-hoc command
- Ansible Playbook
- Ansible Variables
- Ansible Facts
- Ansible Loops
- Ansible Conditionals
- Ansible Vaults
- Ansible Roles
- Ansible Galaxy

Introduction Ansible

IDN.ID
INDONESIAN EXPERT FACTORY

Why automation

Introduction ansible



What is Ansible ?

Introduction ansible

The ansible project is an open source community sponsored by Red Hat. It’s also a simple automation language that perfectly describes IT application environments in Ansible Playbooks.

Ansible Engine is a supported product built from the Ansible community project.

Ansible automates the management of remote systems and controls their desired state.

Why Ansible ?

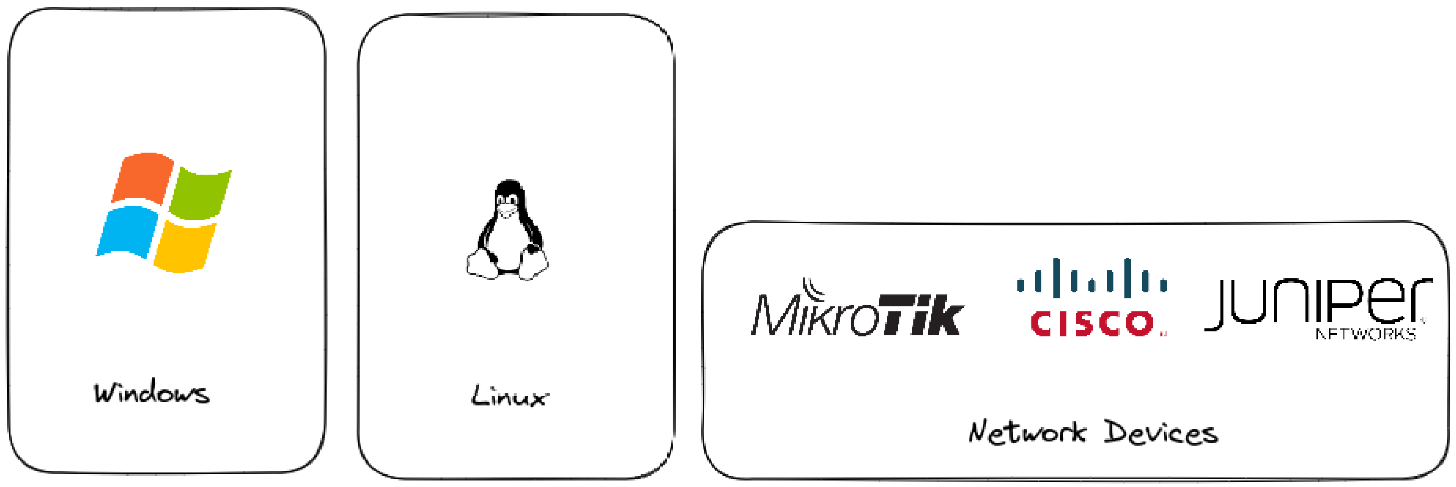
Introduction ansible

Simple	Powerfull	Agentless
Human readable automation	App deployment	Agentless architecture
No special coding skills needed	Configuration management	Uses OpenSSH & WinRM
Tasks executed in order	Workflow orchestration	No agents to exploit or update
Usable by every team	Network automation	Get started immediately
Get productive quickly	**Orchestrate the app lifecycle**	**More efficient & more secure**

With ansible you can automate

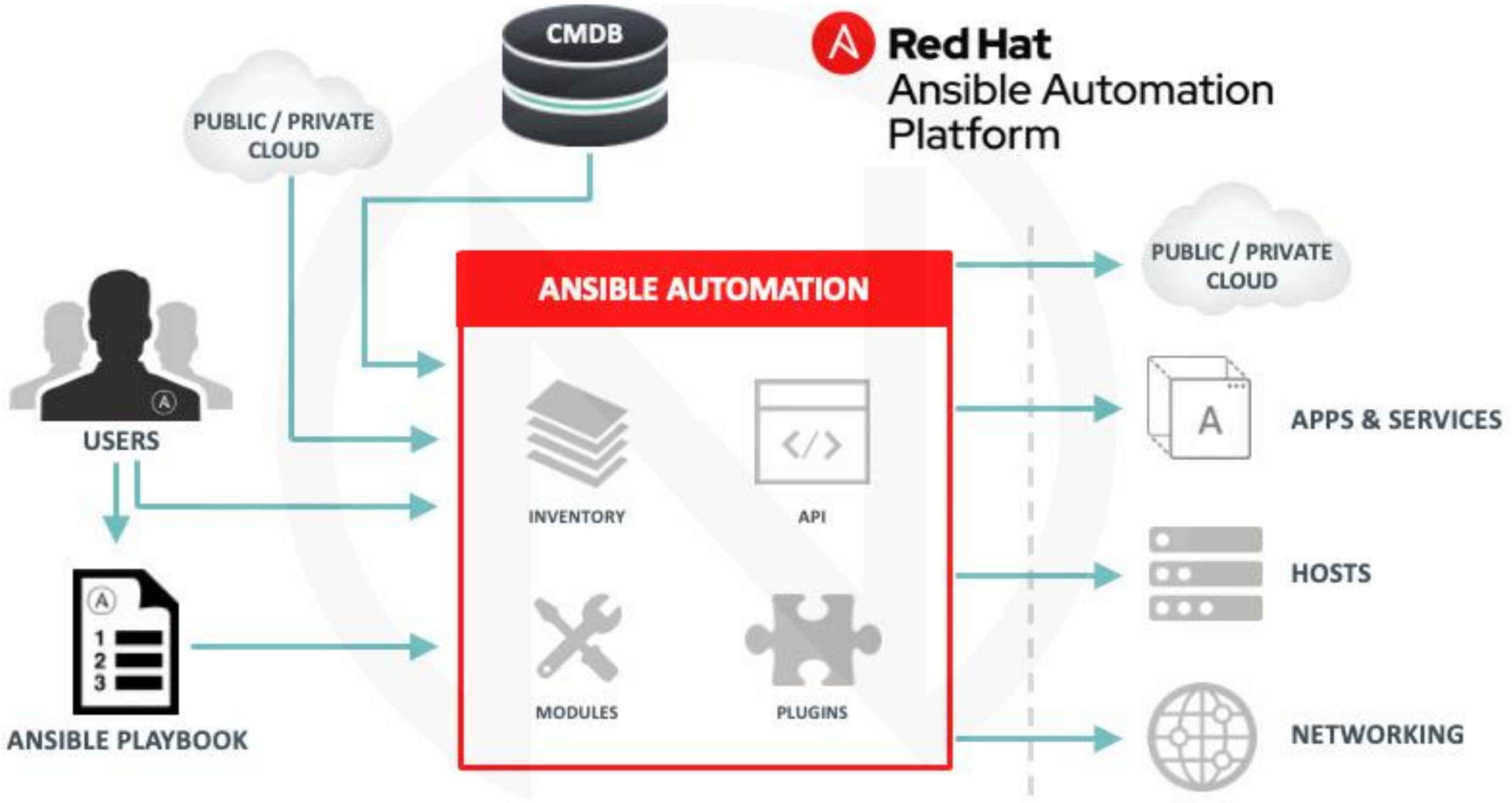
Introduction ansible

CROSS PLATFORM – Linux, Windows, UNIX



Ansible architecture

Introduction ansible

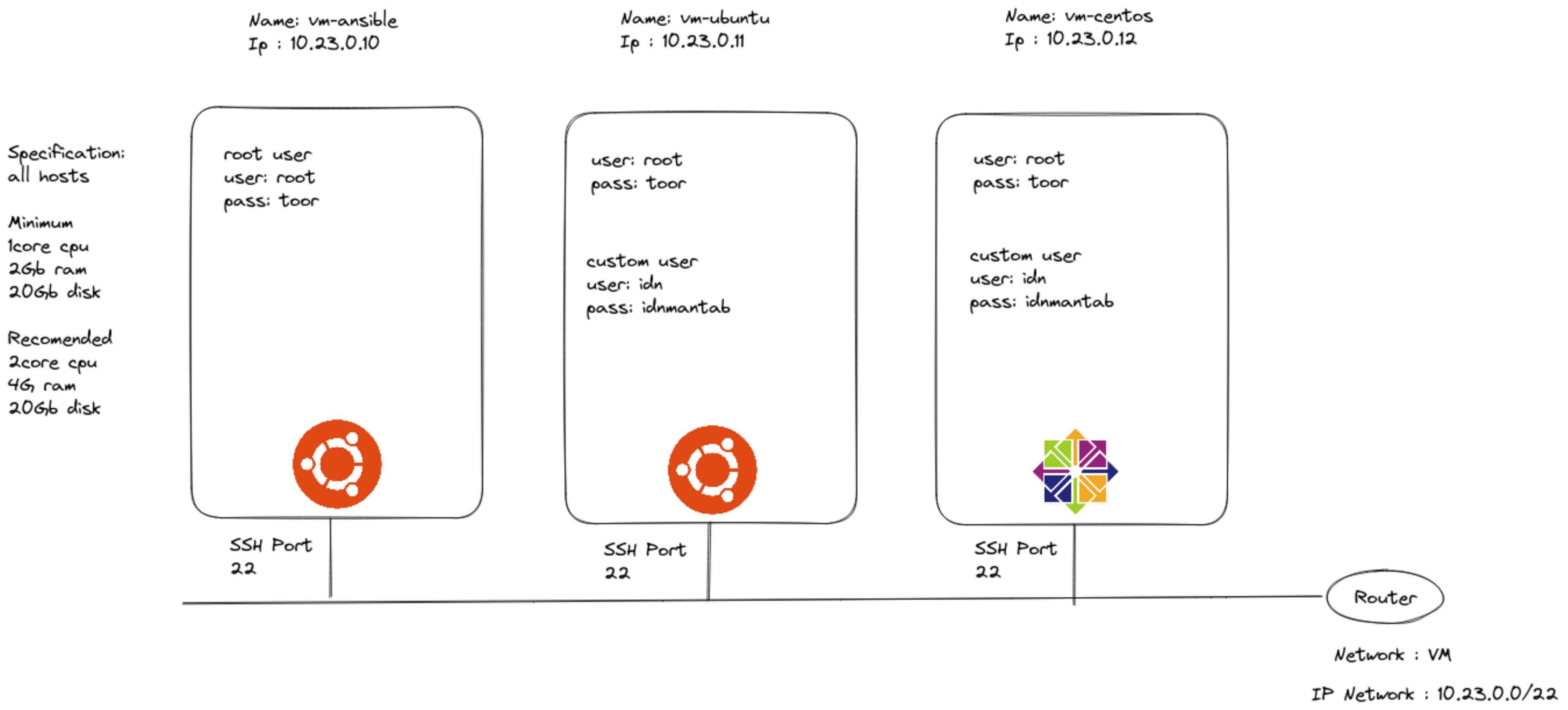


Setup Lab

IDN.ID
INDONESIAN EXPERT FACTORY

Lab Topology

Setup Lab



Introduction YAML

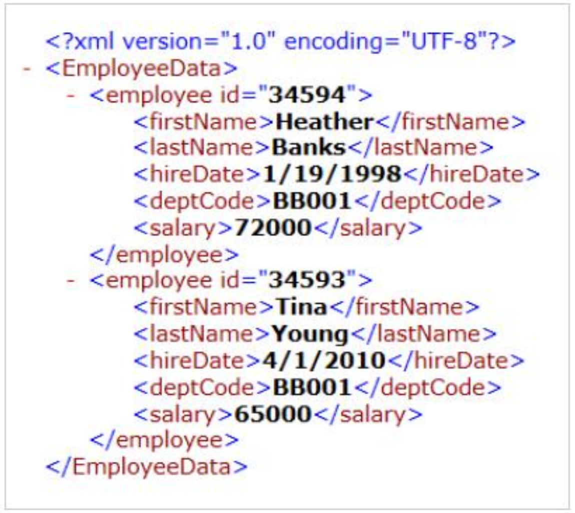
IDN.ID
INDONESIAN EXPERT FACTORY

What is YAML ?

Introduction YAML

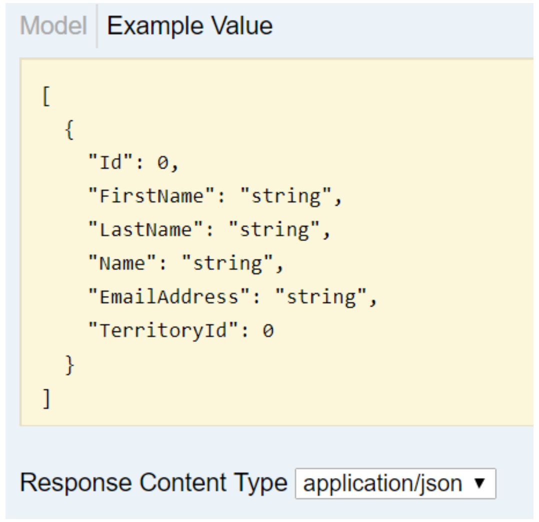
YAML YAML Ain't Markup Language

YAML is a human-friendly data serialization language for all programming languages.



```xml
  <?xml version="1.0" encoding="UTF-8"?>
- <EmployeeData>
  - <employee id="34594">
        <firstName>Heather</firstName>
        <lastName>Banks</lastName>
        <hireDate>1/19/1998</hireDate>
        <deptCode>BB001</deptCode>
        <salary>72000</salary>
    </employee>
  - <employee id="34593">
        <firstName>Tina</firstName>
        <lastName>Young</lastName>
        <hireDate>4/1/2010</hireDate>
        <deptCode>BB001</deptCode>
        <salary>65000</salary>
    </employee>
  </EmployeeData>
```

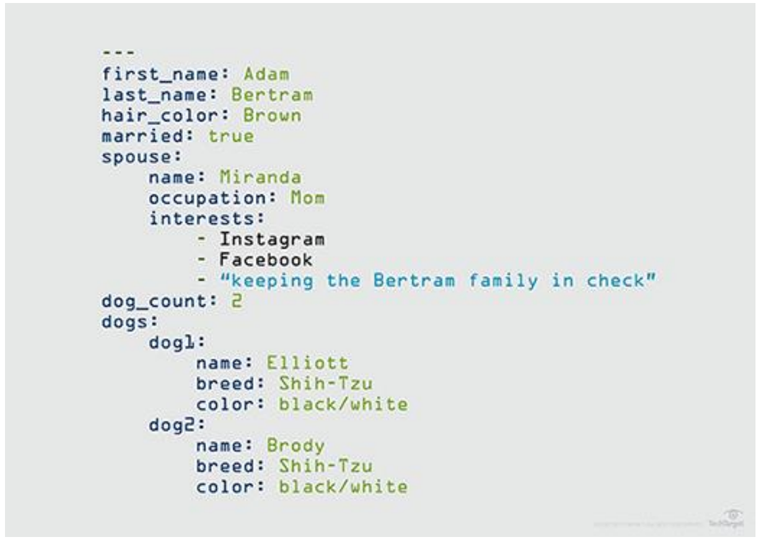
XML



```json
[
  {
    "Id": 0,
    "FirstName": "string",
    "LastName": "string",
    "Name": "string",
    "EmailAddress": "string",
    "TerritoryId": 0
  }
]
```

Response Content Type: application/json

JSON



```yaml
---
first_name: Adam
last_name: Bertram
hair_color: Brown
married: true
spouse:
    name: Miranda
    occupation: Mom
    interests:
        - Instagram
        - Facebook
        - "keeping the Bertram family in check"
dog_count: 2
dogs:
    dog1:
        name: Elliott
        breed: Shih-Tzu
        color: black/white
    dog2:
        name: Brody
        breed: Shih-Tzu
        color: black/white
```

YAML

YAML

Introduction YAML

```
Key: Value

Example
Nama: Rafi
Job: Engineer
```

Key Value Pair

```
Key:
- value1
- value2

Example
Nama:
-   Rafi
-   Riadi

Job:
-   Engineer
-   Developer
```

Array / Lists

```
Data:
   key: value

Example
Trainer:
   Rafi: Engineer
   Riadi: Developer
```

Dictionary / Map

Spaces

Introduction YAML

Dictionary / Map

```
Data:
   key: value

Example
Trainer:
   Rafi: Engineer
   Riadi: Developer
```

YAML - Advanced

Introduction YAML

Key Value/Dictionary/Lists

```
Karyawan:
- Engineer:
    Rafi: Sysadmin
- Developer
    Riadi: Backend
```

✓

Key Value/Dictionary/Lists

```
Karyawan:
- Engineer:
  Rafi: Sysadmin
- Developer
Riadi: Backend
```

✗

Inventory Files

Inventory

Inventory files

An inventory is a file containing:
- Hosts
- Groups
- Inventory-specific data (variables)
- Static or dynamic sources

Inventory default location files
/etc/ansible/hosts

Inventory format with INI or YAML

inventory

```
10.23.0.10
10.23.0.11

[groups]
10.23.0.10
10.23.0.11
```

Inventory with variable

Inventory files

Inventory Parameters:
- ansible_connection = ssh/winrm/cmd
- ansible_port = 22/5986
- ansible_user = root/idn
- ansible_ssh_password = toor/idnmantab
- ansible_host = 10.23.0.x

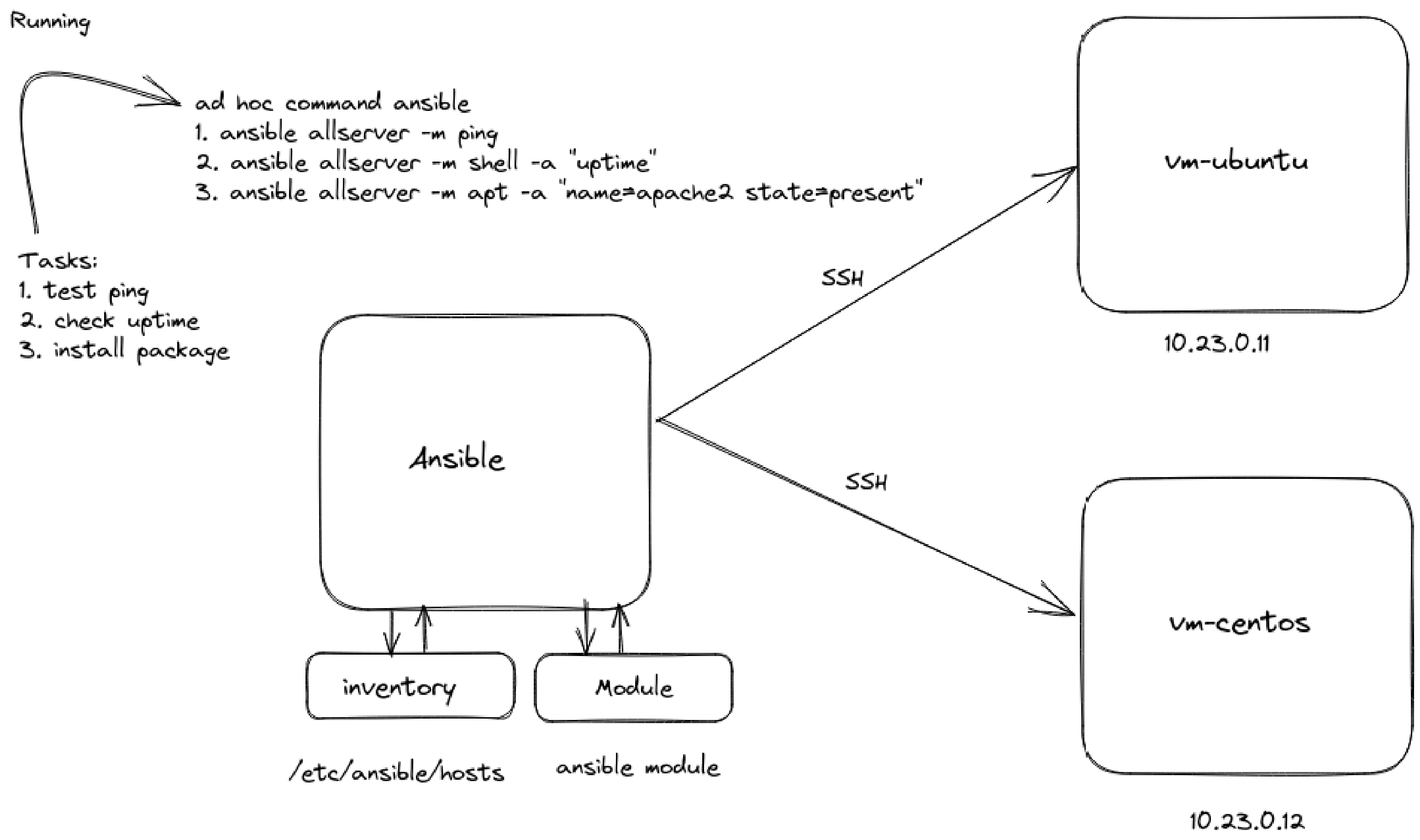
```
inventory

10.23.0.11 ansible_connection=winrm
10.23.0.12 ansible_connection=ssh
ansible_port=2023

[groups]
ubuntu ansible_host=10.23.0.11  ansible_user=idn
10.23.0.12 ansible_user=root
ansible_ssh_password=idnmantab
```

Ansible ad-hoc

IDN.ID
INDONESIAN EXPERT FACTORY

Ad-hoc commands

Ansible ad-hoc



why use ad-hoc command?

Ansible ad-hoc

ad hoc commands are great for tasks you repeat rarely.

An ad hoc command looks like this:

```
$ ansible [pattern] -m [module] -a "[module options]"
```

Use case for ad hoc tasks

Ansible ad-hoc

Rebooting Server

Managing files

Managing packages

Managing users and groups

Managing services

Ansible Modules

Ansible ad-hoc

Modules are discrete units of code that can be used from the command line or in a playbook task.

Module index :

Cloud module

Clustering module

Command module

Database module

File module

System module

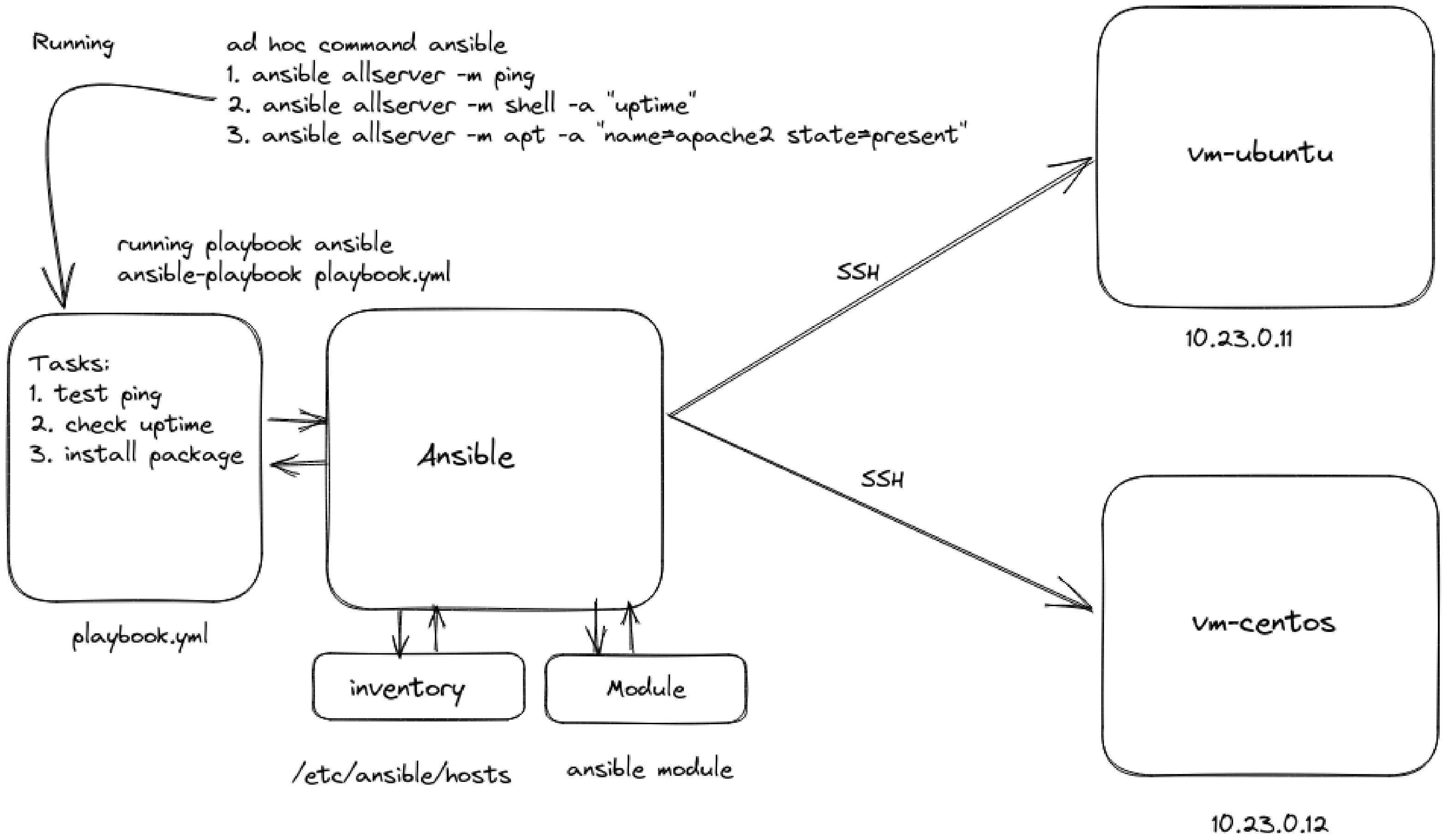
etc

Ansible Playbook

IDN.ID
INDONESIAN EXPERT FACTORY

Ansible playbooks

Ansible playbooks



Playbook

Ansible playbooks

Ansible Playbooks offer a repeatable, re-usable, simple configuration management and multi-machine deployment system, one that is well suited to deploying complex applications.

Playbook can:
- Declare configurations
- orchestrate steps of any manual ordered process, on multiple sets of machines, in a defined order
- Launch tasks synchronously or asynchronously

Playbook format

Ansible playbooks

Playbook format with yaml

```
Playbook.yml

---
- name: Update web servers
  hosts: webservers
  remote_user: root

  tasks:
  - name: Ensure apache is at the latest version
    ansible.builtin.yum:
      name: httpd
      state: latest

- name: Update db servers
  hosts: databases
  remote_user: root

  tasks:
  - name: Ensure postgresql is at the latest version
    ansible.builtin.yum:
      name: postgresql
      state: latest
```

Hosts

Ansible playbooks

Playbook.yml

```
---
- name: Update web servers
  hosts: ubuntu
  remote_user: root

  tasks:
  - name: Ensure apache is at the latest version
    apt:
      name: apache2
      state: latest
```

$\longrightarrow$

Inventory

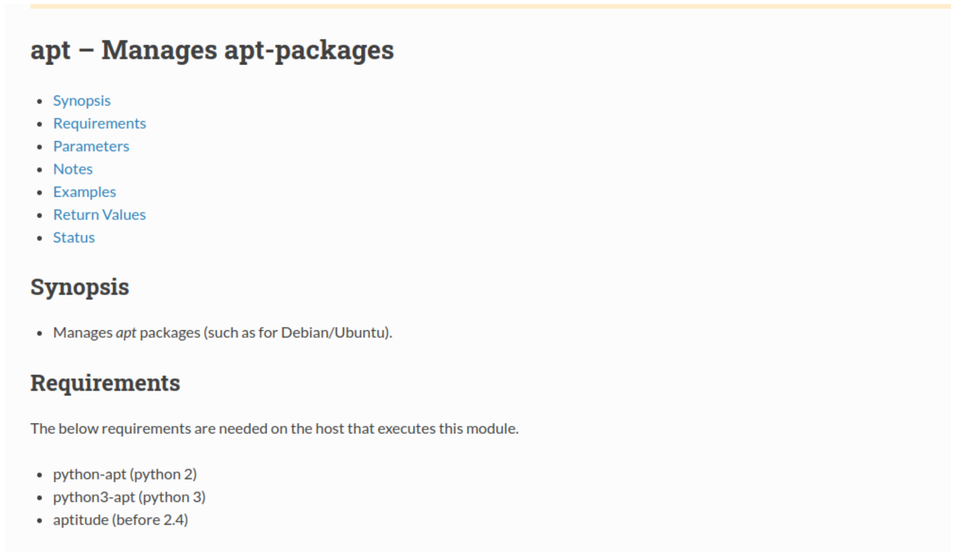
```
[ubuntu]
10.23.0.11

[centos]
10.23.0.121
```

Module

Ansible playbooks

Playbook.yml

```
---
- name: Update web servers
  hosts: ubuntu
  remote_user: root

  tasks:
  - name: Ensure apache is at the latest version
    apt:
      name: apache2
      state: latest

  - name: Ensure apache is at the latest version
     systemd:
       name: apache2
       state: restarted
```



apt – Manages apt-packages

- Synopsis
- Requirements
- Parameters
- Notes
- Examples
- Return Values
- Status

Synopsis

- Manages *apt* packages (such as for Debian/Ubuntu).

Requirements

The below requirements are needed on the host that executes this module.

- python-apt (python 2)
- python3-apt (python 3)
- aptitude (before 2.4)

Run

Ansible playbooks

- Execute Ansible Playbook
- Syntax: ansible-playbook <playbook file name>

Ansible Variables

Variable

Ansible Variables

Stores informations that varies with each host

Valid variable names

Not all strings are valid Ansible variable names. A variable name can only include letters, numbers, and underscores. Python keywords or playbook keywords are not valid variable names. A variable name cannot begin with a number.

This table gives examples of valid and invalid variable names:

Valid variable names	Not valid
foo	*foo,python keywords such as async and lambda
foo_env	Playbook keywords such as environtment
foo_port	foo-port, foo port, foo.port
foo5, _foo	5foo, 12

Using Variable

Ansible Variables

```
---
-  name : Set firewall configurations
    hosts: web
    tasks:
    - firewalld:
          service: https
          permanent: true
          state: enabled

    - firewalld:
          port: '{{ http_port }}'/tcp
          permanent: true
          state: disabled

    - firewalld:
          port: '{{ snmp_port }}'/udp
          permanent: true
          state: disabled
```

```
# Sample Inventory File

web http_port=8081 snmp_port=161-162
```

```
# Sample variable File – web.yml

http_port: 8081
snmp_port: 161-162
```

Ansible Facts

IDN.ID
INDONESIAN EXPERT FACTORY

Facts

Ansible Facts

Ansible facts are data related to your remote systems, including operating systems, IP addresses, attached filesystems, and more. You can access this data in the ansible_facts variable

```
{
    "ansible_all_ipv4_addresses": [
        "REDACTED IP ADDRESS"
    ],
    "ansible_all_ipv6_addresses": [
        "REDACTED IPV6 ADDRESS"
    ],
    "ansible_apparmor": {
        "status": "disabled"
    },
"ansible_distribution": "CentOS",
    "ansible_distribution_file_variety": "RedHat",
    "ansible_distribution_major_version": "7",
    "ansible_dns": {
        "nameservers": [
            "127.0.0.1"
        ]
}
```

Ansible Loops

IDN.ID
INDONESIAN EXPERT FACTORY

Loops

Ansible loops

Ansible offers the loop until keywords to execute a task multiple times.

Examples of commonly-used loops include changing ownership on several files and/or directories with the file module, creating multiple users with the user module, and repeating a polling step until a certain result is reached.

Using Loops

Ansible loops

```
-   name: install package
    hosts: ubuntu
    tasks:
       - apt: name=apache2  state=present
       - apt: name=squid  state=present
       - apt: name=bind9  state=present
       - apt: name=samba  state=present
       - apt: name=mysql  state=present
```

```
-   name: install package
    hosts: ubuntu
    tasks:
       - apt: name='{{ item }}' state=present
         loop:
           - apache2
           - squid
           - bind9
           - samba
           - mysql
```

Ansible Conditionals

Conditionals – when

Ansible Conditionals

In a playbook, you may want to execute different tasks, or have different goals, depending on the value of a fact (data about the remote system), a variable, or the result of a previous task.

You may want the value of some variables to depend on the value of other variables. Or you may want to create additional groups of hosts based on whether the hosts match other criteria. You can do all of these things with conditionals.

Using Conditionals – when

Ansible Conditionals

```
-   name: install nginx
    hosts: ubuntu
    tasks:
       - name: Install nginx on ubuntu
         apt:
             name: nginx
             state: present
```

```
-   name: install nginx
    hosts: centos
    tasks:
       - name: Install nginx on centos
         yum:
             name: httpd
             state: present
```

```
-   name: install nginx
    hosts: ubuntu
    tasks:
       - name: Install nginx on ubuntu
         apt:
             name: nginx
             state: present
         when: ansible_os_family == "Debian"

       - name: Install nginx on centos
         yum:
             name: httpd
             state: present
         when: ansible_os_family == "RedHat"
```

Ansible Vaults

Vaults

Ansible Vaults

Ansible Vault encrypts variables and files so you can protect sensitive content such as passwords or keys rather than leaving it visible as plaintext in playbooks or roles

To use Ansible Vault you need one or more passwords to encrypt and decrypt content.

Using Vaults

Ansible Vaults

```
-   name: install nginx
    hosts: ubuntu
    tasks:
       - name: Install nginx on ubuntu
         apt:
            name: nginx
            state: present
         when: ansible_os_family == "Debian"

       - name: Install nginx on centos
         yum:
            name: httpd
            state: present
         when: ansible_os_family == "RedHat"
```

```
3061323363346134383765383366633364306163656130333837366131383833
3565653635353162306132336334613438376538336663336430616365613O3
3383736613138383335656536353531623061323363346134383765383366633
3643061636561303338373661313838333565653635353162306132336334613
4383765383366633364306163656130333837366131383833356565363535531
6230613233633461343837653833666333643061636561303338373661313838
3335656536353531623061323363346134383765383366633364306163656613
0333837366131383833356565363535316230613233633461343837653833666
3336430616365613033383736613138383335656536353531623061323363346I
3438376538336663336430616365613033383736613138383335656536353533
1623061323363346134383765383366633364306163656130333837366131383838
3335656536353531623061323363346134383765383366633364306163656613
03338373661313838333565653635353162
```

Ansible Roles

IDN.ID
INDONESIAN EXPERT FACTORY

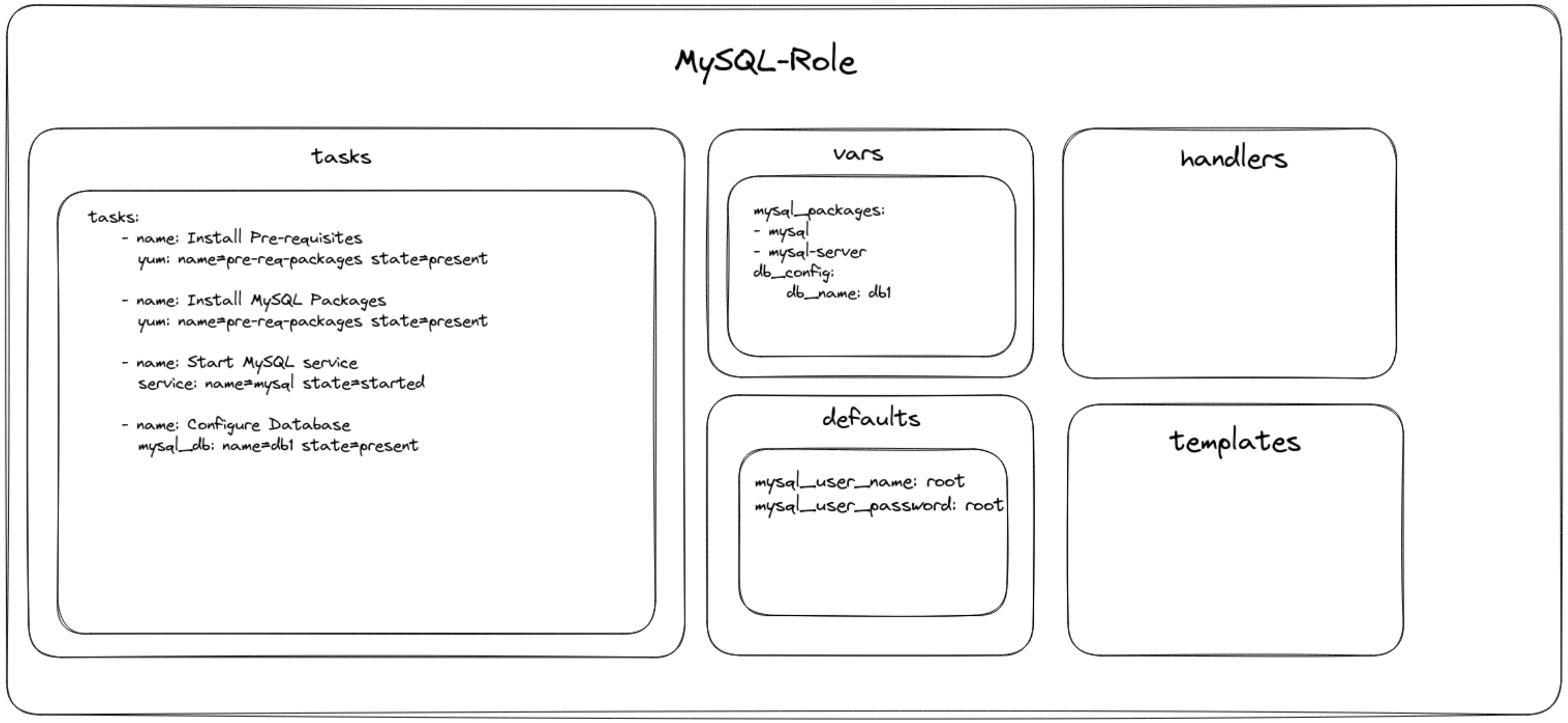
Roles

Ansible Roles

Roles let you automatically load related vars, files, tasks, handlers, and other Ansible artifacts based on a known file structure. After you group your content in roles, you can easily reuse them and share them with other users.

Roles directory structure

Ansible Roles



```
tasks:
    - name: Install Pre-requisites
      yum: name=pre-req-packages state=present

    - name: Install MySQL Packages
      yum: name=pre-req-packages state=present

    - name: Start MySQL service
      service: name=mysql state=started

    - name: Configure Database
      mysql_db: name=db1 state=present
```

```
mysql_packages:
- mysql
- mysql-server
db_config:
    db_name: db1
```

```
mysql_user_name: root
mysql_user_password: root
```

Storing and finding roles

Ansible Roles

By default, Ansible looks for roles in the following locations:

- In collections, if you are using them
- In a directory called roles/, relative to the playbook file
- In the configured roles_path. The default search path is "~/.ansible/roles:/usr/share/ansible/roles:/etc/ansible/roles".
- In the directory where the playbook file is located

Ansible Galaxy

IDN.ID
INDONESIAN EXPERT FACTORY

Galaxy

Ansible Galaxy

Galaxy is a hub for finding and sharing Ansible content.

Use Galaxy to jump-start your automation project with great content from the Ansible community. Galaxy provides pre-packaged units of work known to Ansible as Roles, and new in Galaxy 3.2, Collections.

Terimakasih

IDN.ID
INDONESIAN EXPERT FACTORY

Let’s Talk

If you have any other questions or would like us to clarify anything else, please, let me know.
We are always glad to help in any way I can.

Address
Jl. Anggrek Rosliana No.12A
Slipi, Jakarta Barat, Indonesia

Contact
0813-2120-6598
rafi@idn.id
www.idn.id