Siapkan 3 unit server. Bisa dalam bentuk VM di laptop/PC atau VPS

- Karena saya menggunakan linux ubuntu, saya mempersiapkan 3 unit server denngan install vagrant terlebih dahulu

sudo apt-get install virtualbox virtualbox-dkms

wget -O- https://apt.releases.hashicorp.com/gpg | sudo gpg --dearmor -o /usr/share/keyrings/hashicorp-archive-keyring.gpg

echo "deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com $(lsb_release -cs) main" | sudo tee /etc/apt/sources.list.d/hashicorp.list

sudo apt update && sudo apt install vagrant

- create 3 sever pada vargant dengan konfugasri sebagai berikut

```
Vagrant.configure("2") do |config|
  (1..3).each do |i|
    config.vm.define "server#{i}" do |server|
      server.vm.box = "ubuntu/xenial64"
      server.vm.network "private_network", type: "dhcp"
      server.vm.provider "virtualbox" do |v|
        v.memory = 2024
        v.cpus = 2
      end
    end
  end
end
```

- install ansible pada vm utama (diluar vagrant)

sudo apt update

sudo apt install ansible

- generate keygen dan menaruh pada masing" host vagrant

key-gen

memberikan nama keygen dengan vagrant, tergenerate 2 file vagrant.pem dan vagrant.pem.pub

copy isi dari vagrant.pem dan memasukannya pada file ~/.ssh/authorized_keys pada masing - masing host vagrant

- setup ansible

karena file ansible pada dir /etc tidak ada maka dicreate dahulu dengan mkdir ansible

kemudian configurasi file nano /etc/ansible/hosts dengan value berikut

[web-servers]

server1 ansible_host=192.168.56.4

server2 ansible_host=192.168.56.5

server3 ansible_host=192.168.56.6

[all:vars]

ansible_user=vagrant

ansible_ssh_private_key_file=/home/user/Documents/Vagrant/vagrant.pem

kemudian test koneksi

ansible -m ping all

jika terhubung maka outputnya seperi berikut

```
server3 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
server1 | SUCCESS => {
    "ansible_facts": {
```

```
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
server2 | SUCCESS => {
    "ansible_facts": {
        "discovered_interpreter_python": "/usr/bin/python3"
    },
    "changed": false,
    "ping": "pong"
}
```

- install docker pada tiap host dengan menggunakan ansible

install_docker.yaml

```
---
- name: Install Docker and Prerequisites
  hosts: all
  become: yes
  tasks:
    - name: Update package cache
      apt:
        update_cache: yes
      when: ansible_os_family == "Debian"

    - name: Install required packages
      apt:
        name: "{{ item }}"
        state: present
      loop:
        - apt-transport-https
```

```
        - ca-certificates
        - curl
        - software-properties-common
        - docker.io  # Install Docker
        - python3-pip  # Optional for Kubernetes package installation
      when: ansible_os_family == "Debian"

    - name: Start and enable Docker
      systemd:
        name: docker
        enabled: yes
        state: started
```

install_kubernetes_tools.yml

```
---
- name: Install Kubernetes Tools
  hosts: all
  become: yes
  tasks:
    - name: Install kubeadm, kubelet, and kubectl
      apt:
        name: "{{ item }}"
        state: present
      loop:
        - kubeadm
        - kubelet
        - kubectl
```

initialize_master_node.yml

```
---
- name: Initialize Kubernetes Master Node
```

```
  hosts: server1
  become: yes
  tasks:
    - name: Initialize Kubernetes Master Node
      command: kubeadm init --pod-network-cidr=192.168.0.0/16
      register: kubeadm_output

    - name: Save kubeadm join command
      lineinfile:
        path: /home/vagrant/kubeadm_join.sh
        create: yes
        line: "{{ kubeadm_output.stdout_lines[0] }}"
```

join_cluster.yml

```
---
- name: Join Node to Kubernetes Cluster
  hosts: server2, server3
  become: yes
  tasks:
    - name: Run kubeadm join command
      shell: "bash /home/vagrant/kubeadm_join.sh"
```

install_cni.yml

```
---
- name: Install CNI (Flannel)
  hosts: server1
  become: yes
  tasks:
    - name: Install Flannel CNI
      command: kubectl apply -f
https://raw.githubusercontent.com/coreos/flannel/master/Documentation/kube-flannel.yml
```

- setup garafana loki

untuk mempercepat saya menggunakan ansible untul pull images

```
---
- name: Pull Loki
  hosts: server2, server3
  tasks:
    - name: Pull Loki Docker Image
      docker_image:
        name: grafana/loki:latest
        source: pull
        state: present
      register: loki_image_result


---
- name: Pull Loki and Grafana Docker Images
  hosts: server1
  tasks:
    - name: Pull Grafana Docker Image
      docker_image:
        name: grafana/grafana:latest
        source: pull
        state: present
      register: grafana_image_result
```