**Elroy Full Course**

1. **Chuẩn bị server**
   1. **những server cần chuẩn bị**

192.168.204.100 gitlab.mixcredevops.vip

192.168.204.204 admin.mixcredevops.vip

192.168.204.112 onlineshop1.mixcredevops.vip

192.168.204.113 onlineshop2.mixcredevops.vip

192.168.204.111 onlineshop.mixcredevops.vip

192.168.204.101 jenkins.mixcredevops.vip

192.168.204.102 jfrog.mixcredevops.vip

192.168.204.103 harbor.mixcredevops.vip

192.168.204.104 cloudbear.mixcredevops.vip

192.168.204.105 sqlserver1.mixcredevops.vip

192.168.204.106 sqlserver2.mixcredevops.vip

192.168.204.107 sqlserver3.mixcredevops.vip

192.168.204.108 build.mixcredevops.vip

192.168.204.109 devenv.mixcredevops.vip

192.168.204.110 staenv.mixcredevops.vip

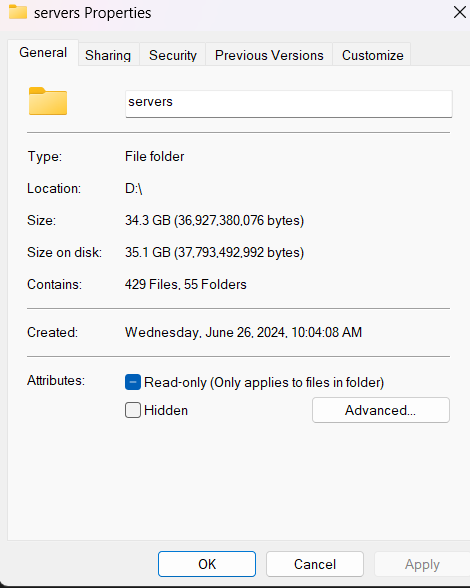
192.168.204.114 elk.mixcredevops.vip

192.168.204.115 grafana.mixcredevops.vip

192.168.204.116 prometheus.mixcredevops.vip

192.168.204.117 zabbix.mixcredevops.vip

* **Detail**: <https://docs.google.com/spreadsheets/d/1yGKbOFTTIduTAnGumeg1D0sEVBTh6V0CfbECurMAR4E/edit?gid=1901840679#gid=1901840679>



* 1. **Allow ssh by root**
  2. **Set up server ansible**
* **Chuẩn bị server ansible**
  + Install ansible

sudo apt update

sudo apt install -y software-properties-common

sudo apt-add-repository --yes --update ppa:ansible/ansible

sudo apt install -y ansible

* + ssh key đến các server còn lại
  + inventory.ini
* **install docker by ansible for** 
  + **servers**
    - jfog-server
    - harbor-server
    - cloudbear-server
    - sqlserver-1
    - sqlserver-2
    - sqlserver-3
    - build-server
    - dev-server
    - sta-server
  + roles
    - tasks

# tasks file for docker-installer

- name: Install required packages

  apt:

    name: "{{ item }}"

    state: present

  loop:

    - apt-transport-https

    - ca-certificates

    - curl

    - gnupg-agent

    - software-properties-common

- name: Add Docker GPG key

  apt\_key:

    url: https://download.docker.com/linux/ubuntu/gpg

    state: present

- name: Add Docker repository

  apt\_repository:

    repo: deb [arch=amd64] https://download.docker.com/linux/ubuntu bionic stable

    state: present

- name: Install Docker

  apt:

    name: docker-ce

    state: present

- name: Add user to docker group

  user:

    name: "root"

    groups: docker

    append: yes

- name: Download Docker Compose

  get\_url:

    url: https://github.com/docker/compose/releases/download/1.29.2/docker-compose-Linux-x86\_64

    dest: /usr/local/bin/docker-compose

    mode: 'a+x'

- name: Access Permission

  command: chmod +x /usr/local/bin/docker-compose

  notify: Restart Docker

* + - handlers

- name: Restart Docker

  service:

    name: docker

    state: restarted

* + playboo

- name: Install Docker on all servers

  hosts:

    - docker\_installer\_servers

  become: true

  ignore\_unreachable: yes

  roles:

    - docker-installer

* **install nginx by ansible for**
  + loadbalancing-server
  + jfog-server
  + harbor-server
  + Jenkins-server
  + Cloudbear-server
  + Roles
    - Tasks

- name: Install Nginx

  apt:

    name: nginx

    state: present

    update\_cache: yes

- name: Enable UFW

  ufw:

    state: enabled

    policy: deny

- name: Allow HTTP Traffic

  ufw:

    rule: allow

    port: 80

    proto: tcp

- name: Allow HTTPS Traffic

  ufw:

    rule: allow

    port: 443

    proto: tcp

- name: Reload UFW

  command: ufw reload

  when: ansible\_ufw.status == 'active'

* + - Handlers

- name: Reload Nginx

  service:

    name: docker

    state: reloaded

* **Install gitlab-runner**
  + Dev-server
  + Sta-server
  + Build-server
  + Master-1
  + Master-2
  + Roles

- name: update package list

  apt:

    update\_cache: yes

- name: Add GitLab Runner GPG key

  apt\_key:

    url: https://packages.gitlab.com/gpg.key

    state: present

- name: Add GitLab Runner repository

  apt\_repository:

    repo: "deb https://packages.gitlab.com/runner/gitlab-runner/ubuntu/ focal main"  # Thay 'focal' bằng phiên bản Ubuntu của bạn

    state: present

    update\_cache: yes

- name: Install Gitlab Runner

  apt:

    name: gitlab-runner

    state: present

- name: Allow gitlab-runner to run commands without password

  lineinfile:

    path: /etc/sudoers

    line: "gitlab-runner ALL=(ALL) NOPASSWD: ALL"

    state: present

    validate: 'visudo -cf %s'

* + - Handlers

- name: Restart GitLab Runner

  service:

    name: gitlab-runner

    state: restarted

* **Install gitlab**
  + Handlers

- name: Reload Nginx

  service:

    name: docker

    state: reloaded

* + Tasks

- name: Install Nginx

  apt:

    name: nginx

    state: present

    update\_cache: yes

- name: Enable UFW

  ufw:

    state: enabled

    policy: deny

- name: Allow HTTP Traffic

  ufw:

    rule: allow

    port: 80

    proto: tcp

- name: Allow HTTPS Traffic

  ufw:

    rule: allow

    port: 443

    proto: tcp

- name: Reload UFW

  command: ufw reload

  when: ansible\_ufw.status == 'active'

* + Playbook

- name: Install Gitlab Runner on all servers

  hosts:

    - gitlab-server

  become: true

  ignore\_unreachable: yes

  roles:

    - gitlab-installer

* Install Jenkins
  + Set up install Jenkins
    - Tasks

- name: Install required packages

  apt:

    name: "{{ item }}"

    state: present

  loop:

    - openjdk-11-jdk

    - apt-transport-https

    - gnupg-agent

- name: Add Jenkins GPG key

  apt\_key:

    url: https://pkg.jenkins.io/debian/jenkins.io.key

    state: present

- name: Add Jenkins APT repository

  apt\_repository:

    repo: "deb http://pkg.jenkins.io/debian-stable binary/"

    state: present

    update\_cache: yes

- name: Install Jenkins

  apt:

    name: jenkins

    state: present

- name: Start Jenkins service

  systemd:

    name: jenkins

    state: started

    enabled: true

  notify: Restart Jenkins

- name: Enable UFW

  ufw:

    state: enabled

    policy: deny

- name: Allow 8080/tcp

  ufw:

    rule: allow

    port: 8080

    proto: tcp

* + - Handlers

- name: Restart Jenkins

  systemd:

    name: jenkins

    state: restarted

* + Install nginx

- name: Install Nginx

  apt:

    name: nginx

    state: present

    update\_cache: yes

- name: Enable UFW

  ufw:

    state: enabled

    policy: deny

- name: Allow SSH Connection

  ufw:

    rule: allow

    port: 22

    proto: tcp

- name: Allow HTTP Traffic

  ufw:

    rule: allow

    port: 80

    proto: tcp

- name: Allow HTTPS Traffic

  ufw:

    rule: allow

    port: 443

    proto: tcp

- name: Reload UFW

  command: ufw reload

* + - Handlers

- name: Reload Nginx

  service:

    name: docker

    state: reloaded

* + Setup Jenkins listen 80
    - Default

---

nginx\_conf\_path: /etc/nginx/conf.d/jenkins.conf

listen\_port: 80

server:

  name: jenkins-server

  address: 192.168.204.101

  port: 8080

* + - Templates

upstream server {

    server {{ server.address }}:{{ server.port }};

}

server {

    listen {{ listen\_port }};

    location / {

        proxy\_pass http://server;

        proxy\_set\_header Host $host;

        proxy\_set\_header X-Real-IP $remote\_addr;

        proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

        proxy\_set\_header X-Forwarded-Proto $scheme;

    }

}

* + - Tasks

- name: Configure install jenkins

  template:

    src: nginx.conf.j2

    dest: "{{ nginx\_conf\_path  }}"

    owner: root

    group: root

    mode: '0644'

- name: Update Nginx to listen on port 9999

  lineinfile:

    path: /etc/nginx/sites-available/default

    regexp: '^\s\*listen\s+(80|[::]:80)\s+default\_server;'

    line: '    listen 9999 default\_server;'

    state: present

    backup: yes

  notify: Reload nginx

* + - Handlers

- name: Reload nginx

  service:

    name: nginx

    state: reloaded

* + Playbook

- name: Install jenkins on server

  hosts:

    - jenkins-server

  become: true

  ignore\_unreachable: yes

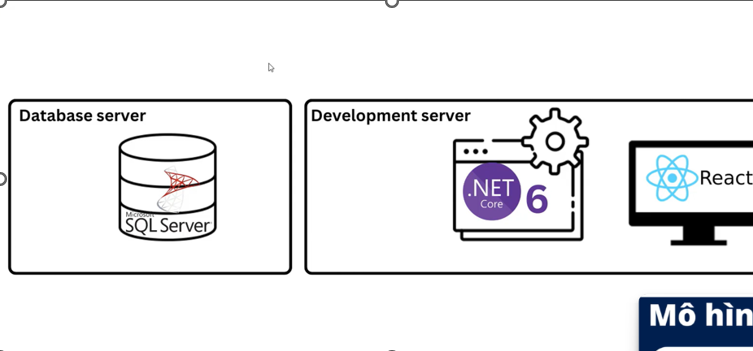
  roles:

    - nginx-installer

    - jenkins-installer

    - jenkins-nginx-installer

1. **Trien khai thu cong**
   1. **Setup database**

****

* + 1. **Cloudbeaver**
* Default

cloudbear\_data\_path: /opt/cloudbeaver/docker-compose.yml

* Handl ers

- name: Restart CloudBeaver

  command: docker-compose restart

  args:

    chdir: /opt/cloudbeaver

* Tasks

- name: Create Cloudbear directory

  file:

    path: /opt/cloudbeaver

    state: directory

    mode: '0755'

- name: Deploy docker-compose.yml

  template:

    src: docker-compose.yml.j2

    dest: /opt/cloudbeaver/docker-compose.yml

- name: Start CloudBeaver using Docker Compose

  command: docker-compose up -d

  args:

    chdir: /opt/cloudbeaver

- name: Enable UFW

  ufw:

    state: enabled

    policy: deny

- name: Allow 8978/tcp

  ufw:

    rule: allow

    port: 8978

    proto: tcp

* Template

version: '3.8'

services:

  cloudbeaver:

    image: dbeaver/cloudbeaver

    container\_name: cloudbeaver

    restart: always

    ports:

      - "8978:8978"

    volumes:

      - cloudbeaver-data:/opt/cloudbeaver/workspace

volumes:

  cloudbeaver-data:

    driver: local

* Playbook

- name: Install jenkins on server

  hosts:

    - cloudbear-server

  become: true

  ignore\_unreachable: yes

  roles:

    - docker-installer

    - cloudbear-installer

    - nginx-installer

    - cloudbear-nginx-installer

* + 1. **Set up database**
* Default

file\_installer: docker-compose.yml

wkdr\_path: /opt/sqlserver

* Handlers

- name: Restart Sqlserver

  command: docker-compose restart

  args:

    chdir: "{{ wkdr\_path }}"

* Tasks

- name: Create Cloudbear directory

  file:

    path: "{{ wkdr\_path }}"

    state: directory

    mode: '0755'

- name: Deploy docker-compose.yml

  template:

    src: docker-compose.yml.j2

    dest: "{{ wkdr\_path/file\_installer }}"

- name: Start CloudBeaver using Docker Compose

  command: docker-compose up -d

  args:

    chdir: "{{ wkdr\_path }}"

- name: Enable UFW

  ufw:

    state: enabled

    policy: deny

- name: Allow 8978/tcp

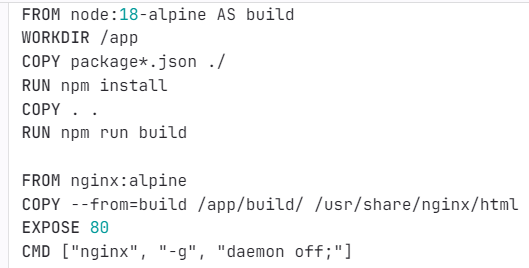
  ufw:

    rule: allow

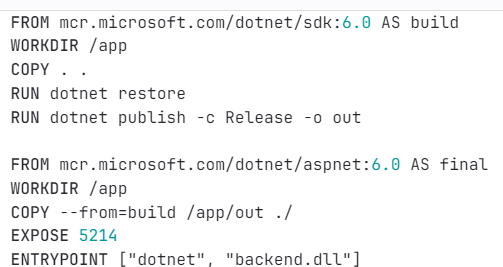
    port: 1433

    proto: tcp

* 1. **Write docker file**
     1. **Frontend**

****

* + 1. **Backend**

****

1. **Trien khai du an tren moi truong env**
   1. **Thiet lap source code len git**
   2. **Dang ky gitlab-runner**
   3. **Setup jfrog**
      1. **Install jfrog**
      2. **Change workdir for 1030**
   4. **Setup harbor**
      1. **Install harbor**
      2. **Thiet lap chung chi ca for harbor**

* Tao chung chi rootca
* Tao file mo rong choc ho chung chi may chu ma muon su dung tu rootca => su dung cho nhieu ten mien hoac dia chi khac nhau
* Tao chung chi tu ky va khoa rieng tu cho CA
* Tao ra CSR va khoa rieng tu cho may chu
* CA => ký chứng chỉ máy chủ vừa tạo
* Sử dụng chứng chỉ
  + 1. **Copy CA for server need to authentication**
  1. **Setup basic pipeline**

1. **Devsecops**
   1. **Sec for dev**

* **Scan with codeclimate**
* **Scan with trivi**
* **Scan with synk**
  1. **Sec for ops**
* **Scan image with trivi**
  1. **DAST**
* **Scan with arachini**
  1. **Test perfomancing**
* **Test performancing with k6**

1. **Chatops**
   1. **Report\_notify\_bot**
   2. **Pipeline\_bot**
2. **Monitoring**
   1. **Monitoring by Grafana**

* **Setup node exporter**
* **Setup Grafana**
* **Setup Prometheus**
  + **Install**
  + **Edit exporters in Prometheus.yml**