

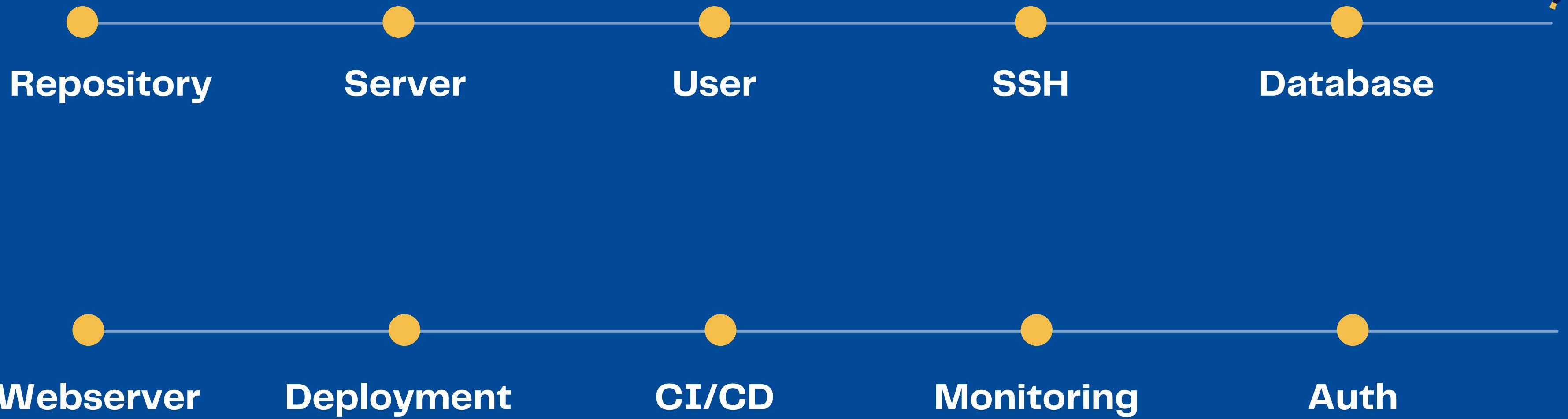
REPORT FINAL-TASK

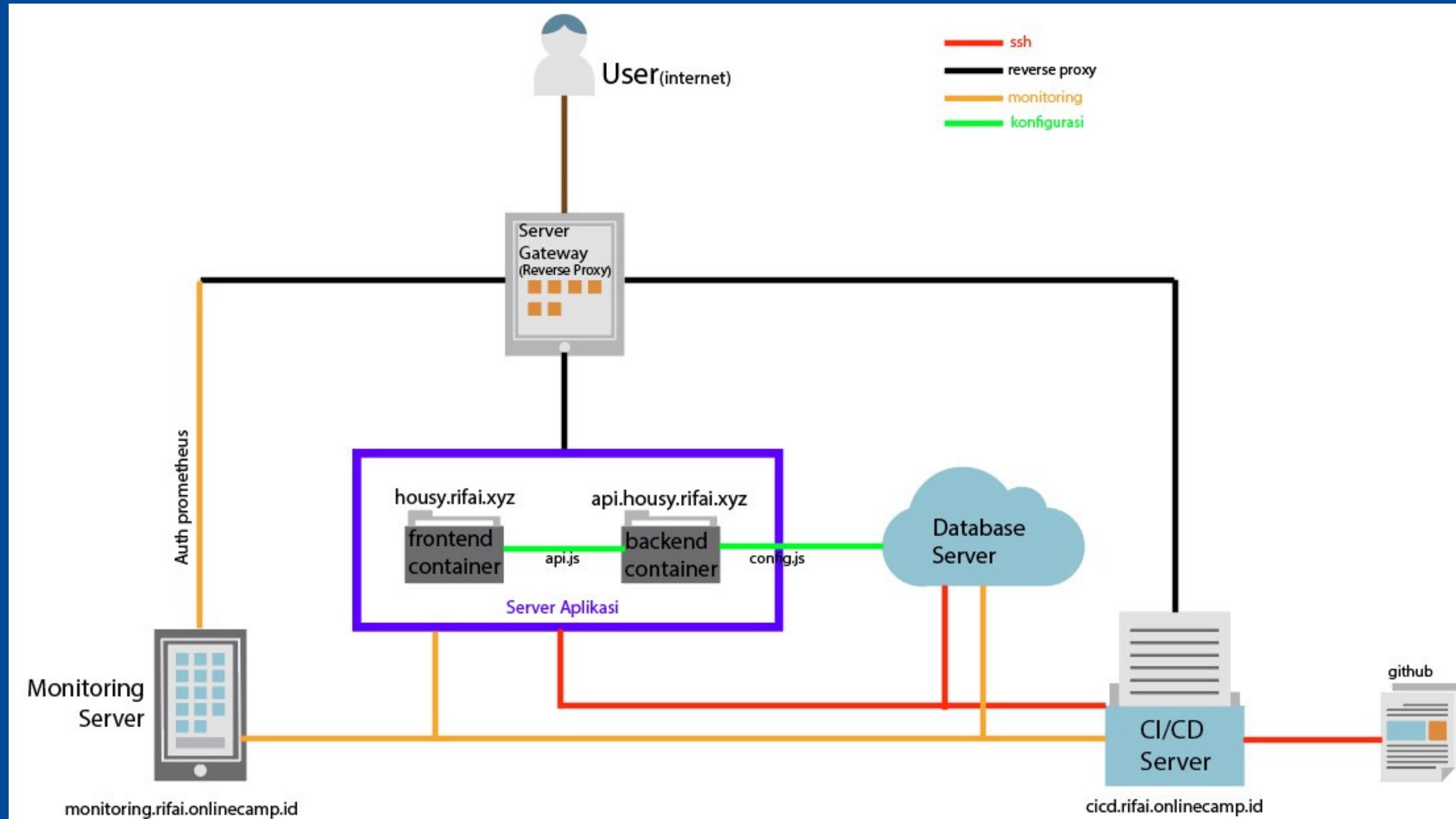
DevOps Engineer Batch 7
dumbways.id

Mochammad Rifai



TASK





REPOSITORY

- Clone Housy frontend
("https://github.com/sgnd/dumbflix-frontend")
- Clone Housy backend
("https://github.com/sgnd/dumbflix-backend")
- Change git remote to your own repository
- Create branch development & production
- Push the code to your repository



```
ubuntu@primary:~/aplikasi$ git clone https://github.com/sgnd/housy-frontend
Cloning into 'housy-frontend'...
remote: Enumerating objects: 563, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 563 (delta 2), reused 0 (delta 0), pack-reused 554
Receiving objects: 100% (563/563), 1.15 MiB | 1.52 MiB/s, done.
Resolving deltas: 100% (339/339), done.
ubuntu@primary:~/aplikasi$ git clone https://github.com/sgnd/housy-backend
Cloning into 'housy-backend'...
remote: Enumerating objects: 172, done.
remote: Counting objects: 100% (172/172), done.
remote: Compressing objects: 100% (60/60), done.
remote: Total 172 (delta 93), reused 170 (delta 92), pack-reused 0
Receiving objects: 100% (172/172), 58.71 KiB | 1.43 MiB/s, done.
Resolving deltas: 100% (93/93), done.
ubuntu@primary:~/aplikasi$ ls
housy-backend  housy-frontend
```

Clone Housy frontend dan Housy backend

Menggunakan perintah
git clone

Change git remote to your own repository

Menggunakan perintah
git remote add

```
ubuntu@primary:~/aplikasi/housy-frontend$ git remote remove origin
ubuntu@primary:~/aplikasi/housy-frontend$ git remote remove origin
fatal: No such remote: 'origin'
ubuntu@primary:~/aplikasi/housy-frontend$ git remote add origin git@github.com:rifaicham/housy-frontend.git
ubuntu@primary:~/aplikasi/housy-frontend$ git push -u origin main
Enumerating objects: 563, done.
Counting objects: 100% (563/563), done.
Compressing objects: 100% (262/262), done.
Writing objects: 100% (563/563), 1.15 MiB | 4.01 MiB/s, done.
Total 563 (delta 339), reused 563 (delta 339)
remote: Resolving deltas: 100% (339/339), done.
To github.com:rifaicham/housy-frontend.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

Create branch development & production

```
ubuntu@primary:~/aplikasi/housy-backend$ git branch development
ubuntu@primary:~/aplikasi/housy-backend$ git branch production
ubuntu@primary:~/aplikasi/housy-backend$ git branch -v
  development 2b397c3 Update Readme.md
* main          2b397c3 Update Readme.md
  production   2b397c3 Update Readme.md
```

Menggunakan perintah
git branch

```
ubuntu@primary:~/aplikasi/housy-backend$ git add .
ubuntu@primary:~/aplikasi/housy-backend$ git status
On branch production
nothing to commit, working tree clean
ubuntu@primary:~/aplikasi/housy-backend$ git commit -m "up production"
On branch production
nothing to commit, working tree clean
ubuntu@primary:~/aplikasi/housy-backend$ git push origin production
Enumerating objects: 172, done.
Counting objects: 100% (172/172), done.
Compressing objects: 100% (59/59), done.
Writing objects: 100% (172/172), 58.71 KiB | 14.68 MiB/s, done.
Total 172 (delta 93), reused 172 (delta 93)
remote: Resolving deltas: 100% (93/93), done.
remote:
remote: Create a pull request for 'production' on GitHub by visiting:
remote:     https://github.com/rifaicham/housy-backend/pull/new/production
remote:
To github.com:rifaicham/housy-backend.git
 * [new branch]      production -> production
```

Push the code to your repository

Menggunakan perintah
git push

Server

Requirements

- Server Nginx
- Server Frontend, Backend, Database, CI/CD
- Server Monitoring

Instructions

- Create multiple server with load balancing (Frontend and Backend)
- Create security group for all server using ufw



Make Server

dengan multipass
menggunakan perintah

**multipass launch -c 1 -m 1G -d
10G --name <nama server>**

```
PS C:\Users\champz> multipass launch -c 1 -m 1G -d 5G --name gateway
Launched: gateway
PS C:\Users\champz> multipass ls
Name          State      IPv4           Image
primary       Running    172.19.152.5   Ubuntu 20.04 LTS
app           Running    172.19.156.209  Ubuntu 20.04 LTS
database      Running    172.19.159.25   Ubuntu 20.04 LTS
gateaway      Running    172.19.149.89   Ubuntu 20.04 LTS
monitoring    Running    172.19.148.146  Ubuntu 20.04 LTS
PS C:\Users\champz>
```

```
frontend.conf
---
upstream lb-frontend{
    least_conn;
    server 172.19.156.209:1001;
    server 172.19.156.209:1002;
}

server {
    server_name housy.rifai.xyz;

    location / {
        proxy_pass http://lb-frontend;
    }
}
```

Load Balancing dalam konfigurasi nginx

```
backend.conf
---
upstream lb-backend{
    least_conn;
    server 172.19.156.209:1003;
    server 172.19.156.209:1004;
}

server {
    server_name api.housy.rifai.xyz;

    location / {
        proxy_pass http://lb-backend;
    }
}
```

USER

- Create user with password
- Disable sign in without password



```
nano setup-user.yml
---
- name: Create User
  hosts: all
  become: true
  vars_files:
    - vars/setup-user-vars.yml
  tasks:

    - name: Create new user
      user:
        name: "{{username}}"
        password: "{{password}}"
        groups:
          - sudo
          - admin
        state: present

    - name: Add public key to authorized_keys
      ansible.posix.authorized_key:
        user: "{{ username }}"
        state: present
        key: "{{ lookup('file', '/home/ubuntu/.ssh/id_rsa.pub') }}"

    - name: Allow specific users to log in
      ansible.builtin.lineinfile:
        dest: /etc/ssh/sshd_config
        regexp: '^AllowUsers'
        line: 'AllowUsers {{ username }}"
        state: present

    - name: Add {{ username }} to sudoers file
      ansible.builtin.lineinfile:
        path: /etc/sudoers
        regexp: '^{{ username }}"
        line: '{{ username }} ALL=(ALL) NOPASSWD: ALL'
        validate: 'visudo -cf %s'

    - name: Enable Password Authentication
      lineinfile:
        path: /etc/ssh/sshd_config
        search_string: 'PasswordAuthentication no'
        line: PasswordAuthentication yes

    - name: Restart SSH Service
      service:
        name: ssh
        state: restarted
```

Create user with password

Membuatnya dalam ansible-playbook

```
- name: Create new user
  user:
    name: "{{username}}"
    password: "{{password}}"
    groups:
      - sudo
      - admin
    state: present
```

username dan password didapatkan dari vars

```
nano setup-user-vars.yml
---
username: fai
password: password
```

```
nano setup-user.yml
---
- name: Create User
  hosts: all
  become: true
  vars_files:
    - vars/setup-user-vars.yml
  tasks:

    - name: Create new user
      user:
        name: "{{username}}"
        password: "{{password}}"
        groups:
          - sudo
          - admin
        state: present

    - name: Add public key to authorized_keys
      ansible.posix.authorized_key:
        user: "{{ username }}"
        state: present
        key: "{{ lookup('file', '/home/ubuntu/.ssh/id_rsa.pub') }}"

    - name: Allow specific users to log in
      ansible.builtin.lineinfile:
        dest: /etc/ssh/sshd_config
        regexp: '^AllowUsers'
        line: 'AllowUsers {{ username }}"
        state: present

    - name: Add {{ username }} to sudoers file
      ansible.builtin.lineinfile:
        path: /etc/sudoers
        regexp: '^{{ username }}"
        line: '{{ username }} ALL=(ALL) NOPASSWD: ALL'
        validate: 'visudo -cf %s'

    - name: Enable Password Authentication
      lineinfile:
        path: /etc/ssh/sshd_config
        search_string: 'PasswordAuthentication no'
        line: PasswordAuthentication yes

    - name: Restart SSH Service
      service:
        name: ssh
        state: restarted
```

Disable sign in without password

Membuatnya dalam ansible-playbook

```
- name: Enable Password Authentication
  lineinfile:
    path: /etc/ssh/sshd_config
    search_string: 'PasswordAuthentication no'
    line: PasswordAuthentication yes
```

```
ubuntu@app:~$ sudo su fai
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

fai@app:/home/ubuntu$
```

Testing user

```
ubuntu@primary:~/ansible$ ssh fai@172.29.149.56
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-94-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Sun Jan 16 07:36:51 WIB 2022

System load:  0.01      Processes:          117
Usage of /:   9.3% of 14.37GB  Users logged in:     1
Memory usage: 69%
Swap usage:   0%          IPv4 address for eth0: 172.29.149.56

1 update can be applied immediately.
To see these additional updates run: apt list --upgradable

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

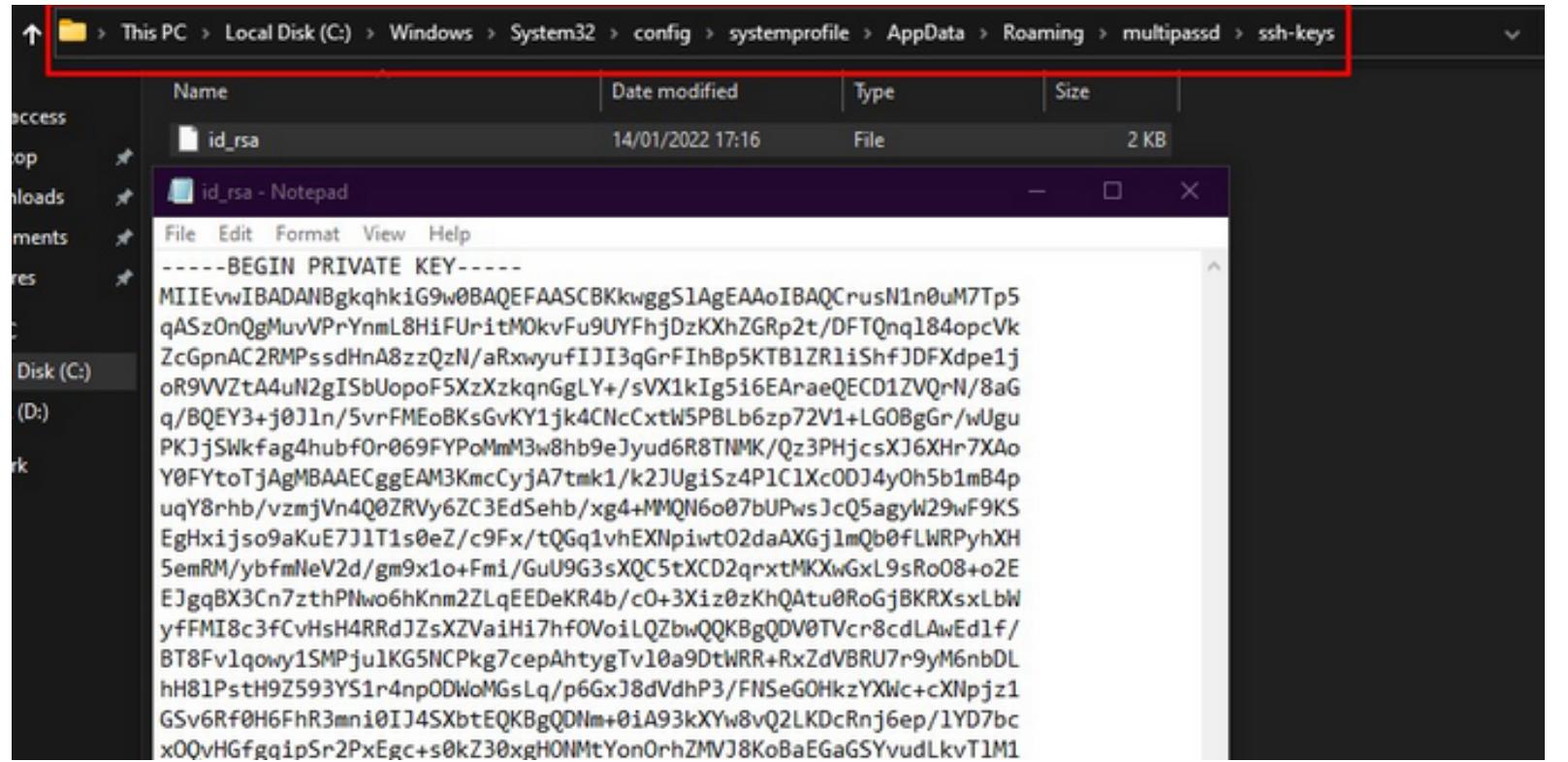
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.
```

- Akses via server yang dibuat usernya
- Akses dari server lain

SSH

- SSH Key for access the server without username & password
- SSH Key for access the git without username & password





SSH Key for access the server without username & password

Multipass SSH
dan simpan kedalam file
<namafilessh>.pem
kemudian ubah hak akses dengan

```
ubuntu@primary:~/ansible$ ssh -i ssh.pem ubuntu@172.23.136.201
Welcome to Ubuntu 20.04.3 LTS (GNU/Linux 5.4.0-94-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

System information as of Sun Jan 16 17:08:55 WIB 2022

System load:          0.02
Usage of /:           19.1% of 19.21GB
Memory usage:         32%
Swap usage:           0%
Processes:            140
Users logged in:      0
IPv4 address for br-e8ed67c64c07: 172.18.0.1
IPv4 address for docker0:   172.17.0.1
IPv4 address for eth0:    172.23.136.201

0 updates can be applied immediately.

Last login: Sun Jan 16 15:37:13 2022 from 172.23.128.1
ubuntu@server1:~$
```

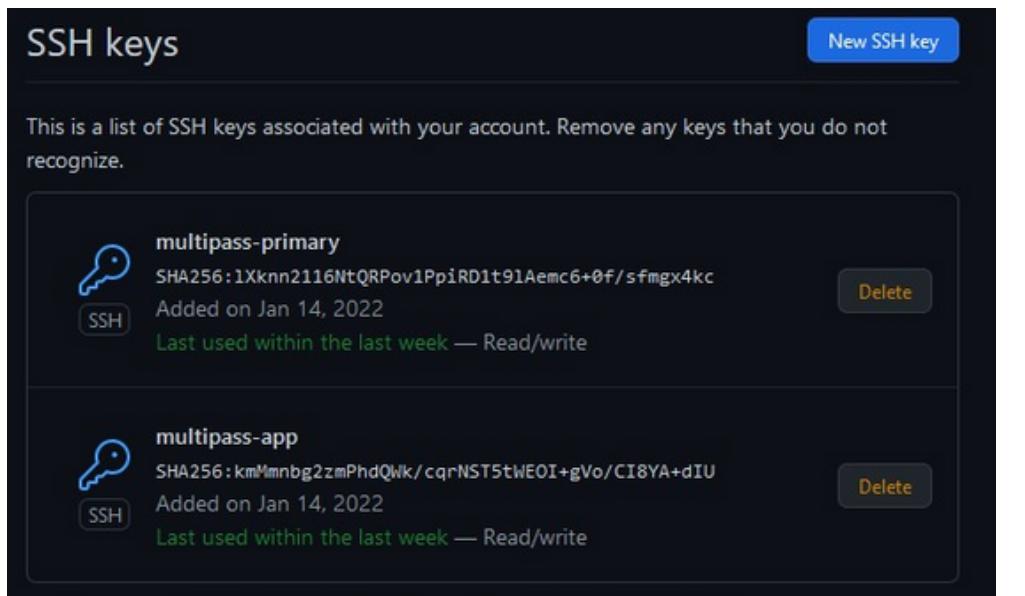
chown 400 namafilessh.pem

untuk read-only

dan akses server1 dengan komputer host dengan perintah

ssh -i <namafilessh>.pem user@ipsserver

```
ubuntu@primary:~$ cd .ssh
ubuntu@primary:~/ssh$ ls
authorized_keys  id_rsa  id_rsa.pub  known_hosts
ubuntu@primary:~/ssh$ cat id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQgQC/1r3nxMNZiagghGx7cmFLKwpS+aqvsfS1jjVIEfDjrq
7X+iR/VH+v0aB0mt9U1CQ0q15Ht0TozsNSw7e8/p0+RkPF+3qNQUuXurhYoj4aGaPonnZKZEaSSgIyy
pQedFFNwLUCYexokY68gVAvHqIC0lf+TNLyJlJwf3ggDcnyzuQknxsrybCkkfElnpVd0LMTTxd89cFu01Y
6rj11inwGGE86SYub6B8CI6wGgM+v9gbSil4zQh+2EhCiD8fQUQT0BFOMStQ+bQDYhsTYZiqXyVNnHEDY0
S1550URS1UyqSYRQkFZ2LqvQbwFoMPA+G4EL3FDYYItezAbZTFN7aqM1UscKKoFKh7jSTsG9+mvLTL/8I
yEb5Yo8Q65JT4zVrgL84sGq3ggY9m3iXYIw/6pDxTNpkA8nPihIwgpytseDqtF8pM/62C46QLQTmiR9j0w
QxyDkiNBE6v0pZyRArpph13vurVymKeaPwTKa2oGaV2wVXAimVA1MO/TbsU=  ubuntu@primary
ubuntu@primary:~/ssh$ []
```



```
ubuntu@primary:~/ssh$ ssh -T git@github.com
Hi rifaicham! You've successfully authenticated, but GitHub does not provide shell
access.
ubuntu@primary:~/ssh$ []
```

SSH Key for access the git without username & password

Generate ssh dengan **ssh-keygen** akses **.ssh/id_rsa.pub** dan copy isi dari file **id_rsa.pub** kedalam SSH keys github

Test dengan perintah **ssh git@github.com -t**

DATABASE

- Setup PostgreSQL with Docker
- Set the volume location in
/home/username/
- Allow database to remote from another
server
- Create database housy and create table
with sequelize



```
nano docker-compose-database.yml
---
version: '3'
services:
  postgresql:
    image: postgres:latest
    container_name: pgsql-database
    environment:
      POSTGRES_USER: root
      POSTGRES_PASSWORD: root
      POSTGRES_DB: housy
    ports:
      - 5432:5432
  volumes:
    - /home/ubuntu/pgsql/data:/var/lib/postgresql/data/
```

```
---
- name: Setup Database with postgre
  hosts: database
  become: true
  tasks:
    - name: Copying docker compose file
      copy:
        src: docker-compose-database.yml
        dest: /home/ubuntu/
    - name: Run docker compose
      shell:
        cmd: "docker-compose -f docker-compose-database.yml up -d"
```

Setup PostgreSQL with Docker

- Membuat docker-compose
- deploy menggunakan ansible-playbook

volume yang dibuat untuk postgresql

```
ubuntu@server1:~$ ls
docker-compose-app.yml      docker-compose-node_exporter.yml  housy-frontend
docker-compose-database.yml  housy-backend                  pgsql
ubuntu@server1:~$
```

Allow database to remote from another server

- Akses database dari localhost
Menggunakan perintah

**psql -U userdatabases -d -h
localhost -p 5432**

- Akses database dari server lain

**psql -U user -d databases -h
ipdatahost -p 5432**

```
ubuntu@database:~$ psql -U root -d housy -h localhost -p 5432
Password for user root:
psql (10.19 (Ubuntu 10.19-0ubuntu0.18.04.1), server 14.1 (Debian 14.1-1.pgdg110+1))
WARNING: psql major version 10, server major version 14.
          Some psql features might not work.
Type "help" for help.

housy=# \dt
Did not find any relations.
housy=#
```

```
ubuntu@app:~$ psql -U root -h 172.27.53.250 -p 5432
Password for user root:
psql (12.9 (Ubuntu 12.9-0ubuntu0.20.04.1), server 14.1 (Debian 14.1-1.pgdg110+1))
WARNING: psql major version 12, server major version 14.
          Some psql features might not work.
Type "help" for help.

root=#
```

```

nano docker-compose-database.yml
---
version: '3'
services:
  postgresql:
    image: postgres:latest
    container_name: pgsql-database
    environment:
      POSTGRES_USER: root
      POSTGRES_PASSWORD: root
      POSTGRES_DB: housy
    ports:
      - 5432:5432
    volumes:
      - /home/ubuntu/pgsql/data:/var/lib/postgresql/data/

```

Create database housy and create table with sequelize

- Digenerate melalui docker-compose
- cek dengan perintah **psql -U root -h localhost --list**

```

ubuntu@server1:~$ sudo docker exec -it pgsql-db bash
root@abc9884d29b2:/# psql -U root -h localhost --list
                                         List of databases
   Name   | Owner | Encoding | Collate | Ctype | Access privileges
---+-----+-----+-----+-----+-----+-----+
housy | root  | UTF8    | en_US.utf8 | en_US.utf8 | =c/root
postgres | root  | UTF8    | en_US.utf8 | en_US.utf8 | root=CTc/root
template0 | root  | UTF8    | en_US.utf8 | en_US.utf8 | =c/root
template1 | root  | UTF8    | en_US.utf8 | en_US.utf8 | =c/root
(4 rows)

```

```

ubuntu@app:~$ sudo docker exec -it backend1 /bin/sh
/app # sequelize db:migrate
Sequelize CLI [Node: 10.24.1, CLI: 6.3.0, ORM: 6.13.0]

Loaded configuration file "config/config.json".
Using environment "development".
== 20200417164621-create-user: migrating =====
== 20200417164621-create-user: migrated (0.048s)

== 20200417174512-create-house: migrating =====
== 20200417174512-create-house: migrated (0.032s)

== 20200417190918-create-transaction: migrating =====
== 20200417190918-create-transaction: migrated (0.029s)

/app #

```

```

ubuntu@database:~$ psql -U root -d housy -h localhost -p 5432
Password for user root:
psql (10.19 (Ubuntu 10.19-0ubuntu0.18.04.1), server 14.1 (Debian 14.1-1.
WARNING: psql major version 10, server major version 14.
          Some psql features might not work.
Type "help" for help.

housy=# \dt
Did not find any relations. sebelum sequelize
housy=# \dt
                                         List of relations
 Schema |        Name        | Type | Owner
-----+-----+-----+-----+
 public | SequelizeMeta | table | root
 public | houses       | table | root
 public | transactions | table | root
 public | users        | table | root
(4 rows)

housy#

```

setelah sequelize

WEB SERVER

- Setup web server nginx / apache2
- SSL support – Integrate Cloudflare with Let's Encrypt

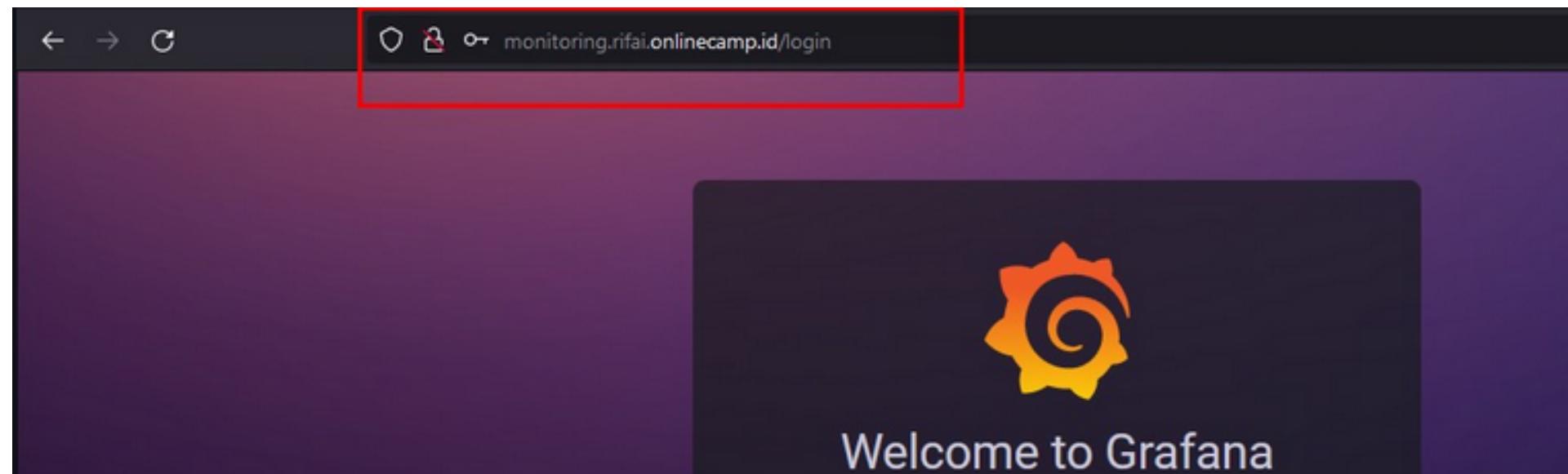


```
- name: Setup Nginx
hosts: gateway
become: true
tasks:
  - name: Install Nginx
    apt:
      name: nginx
      state: present
      update_cache: yes

  - name: Copying frontend.conf
    copy:
      src: frontend.conf
      dest: /etc/nginx/conf.d

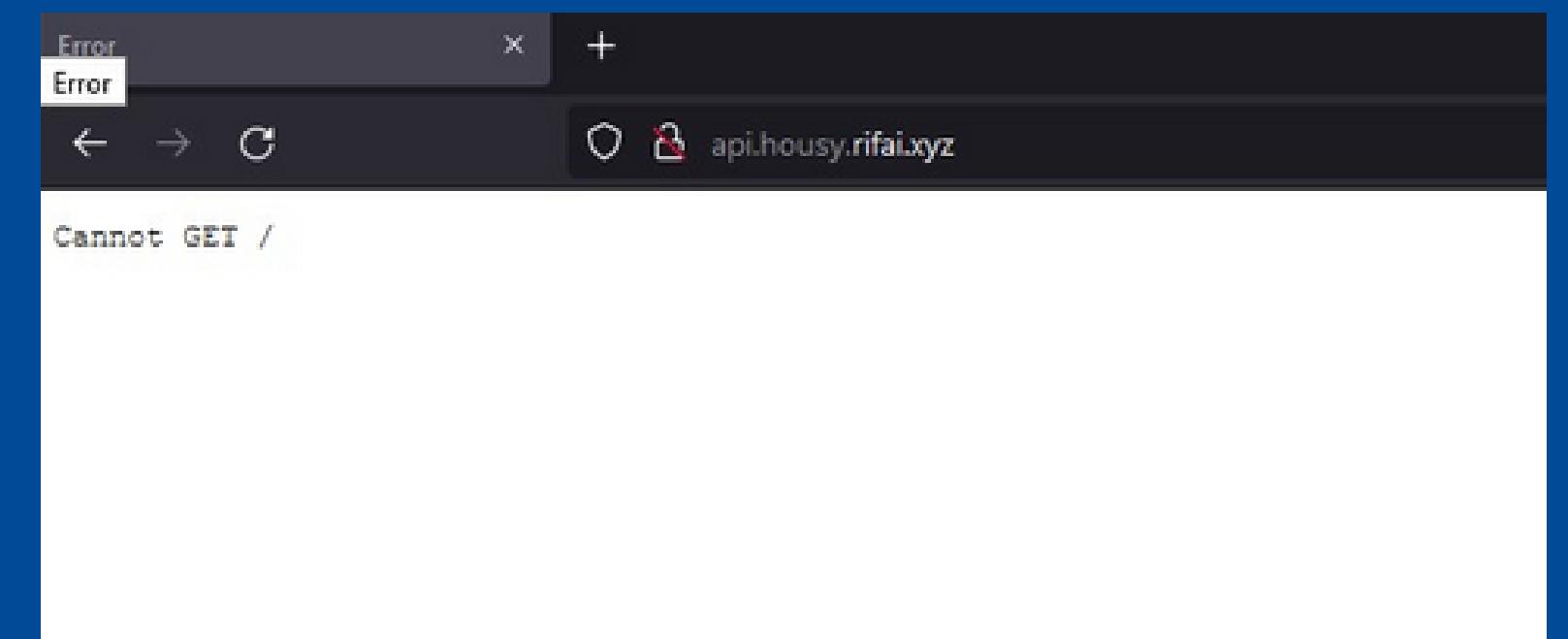
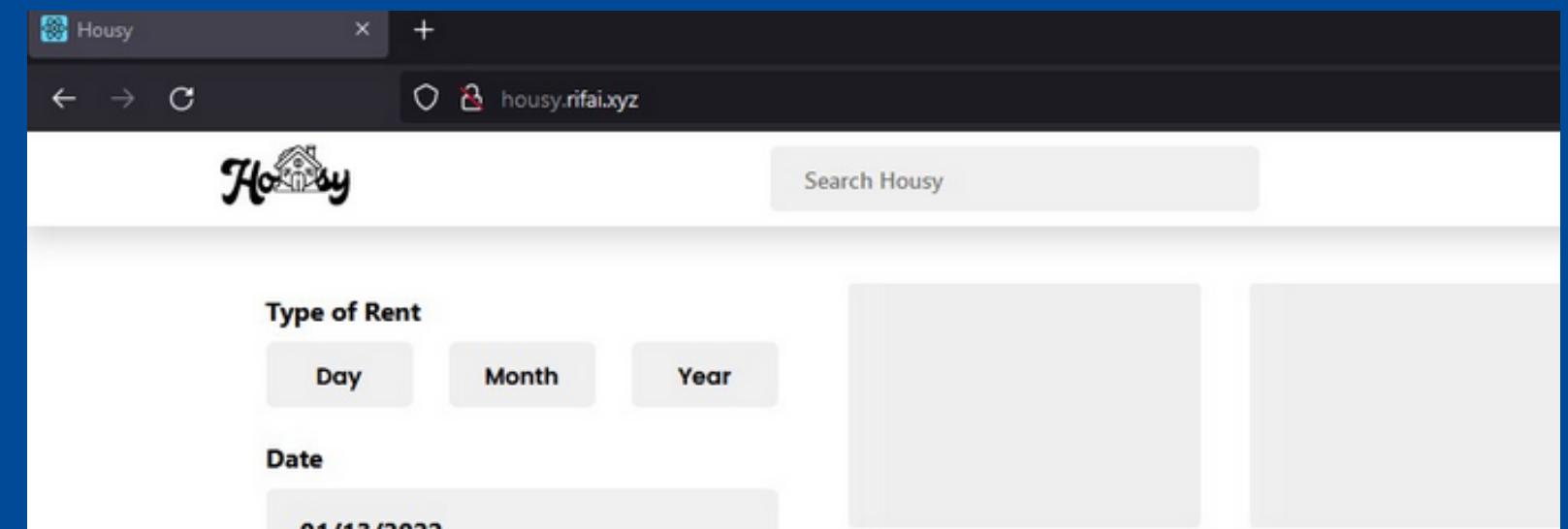
  - name: Copying backend.conf
    copy:
      src: backend.conf
      dest: /etc/nginx/conf.d

  - name: Copying monitoring.conf
    copy:
      src: monitoring.conf
      dest: /etc/nginx/conf.d
```



Webserver

- Install nginx via ansible-playbook dan copy file konfigurasi untuk frontend, backend, dan monitoring



DEPLOYMENT

- Use the branch production
- Deploy applications with docker
- Integration housy frontend, backend and database



```
nano docker-compose-app.yml
---
version: '3.9'

services:

  frontend1:
    container_name: frontend1
    build: ./housy-frontend
    stdin_open: true
    ports:
      - 1001:3000

  backend1:
    container_name: backend1
    build: ./housy-backend
    stdin_open: false
    ports:
      - 1003:5000
    environment:
      SEQ_DB: housy
      SEQ_USER: root
      SEQ_PW: root
      PORT: 5000
    DATABASE_URL: postgres://root:root@localhost/housy
```

Deployment

- Use the branch production
- Deploy applications with docker

docker-compose digunakan untuk deploy aplikasi dalam container

```
nano setup-app.yml
---
- name: Setup-Aplikasi-Frontend-Backend
  hosts: app
  become: true
  tasks:
    - name: Copy docker compose
      copy:
        src: docker-compose-app.yml
        dest: /home/ubuntu/
    - name: clone housy-frontend production branch
      ansible.builtin.git:
        repo: https://github.com/rifaicham/housy-frontend.git
        dest: /home/ubuntu/housy-frontend
        single_branch: yes
        version: production
    - name: clone housy-backend production branch
      ansible.builtin.git:
        repo: https://github.com/rifaicham/housy-backend.git
        dest: /home/ubuntu/housy-backend
        single_branch: yes
        version: production
    - name: Run docker compose
      shell: "docker-compose -f docker-compose-app.yml up -d"
      args:
        executable: /bin/bash
```

```
- name: clone housy-frontend production branch
  ansible.builtin.git:
    repo: https://github.com/rifaicham/housy-frontend.git
    dest: /home/ubuntu/housy-frontend
    single_branch: yes
    version: production
- name: clone housy-backend production branch
  ansible.builtin.git:
    repo: https://github.com/rifaicham/housy-backend.git
    dest: /home/ubuntu/housy-backend
    single_branch: yes
    version: production
```

dalam file ansible-playbook untuk deploy aplikasi sudah dipull menggunakan branch production

```

import axios from "axios";

// Set config defaults when creating the instance
export const API = axios.create({
  baseURL: "http://api.housy.rifai.xyz/api/v1",
});

// Alter defaults after instance has been created
export const setAuthToken = (token) => {
  API.defaults.headers.common["Authorization"] = `Bearer ${token}`;
};

ubuntu@app:~/housy-frontend$
```

```

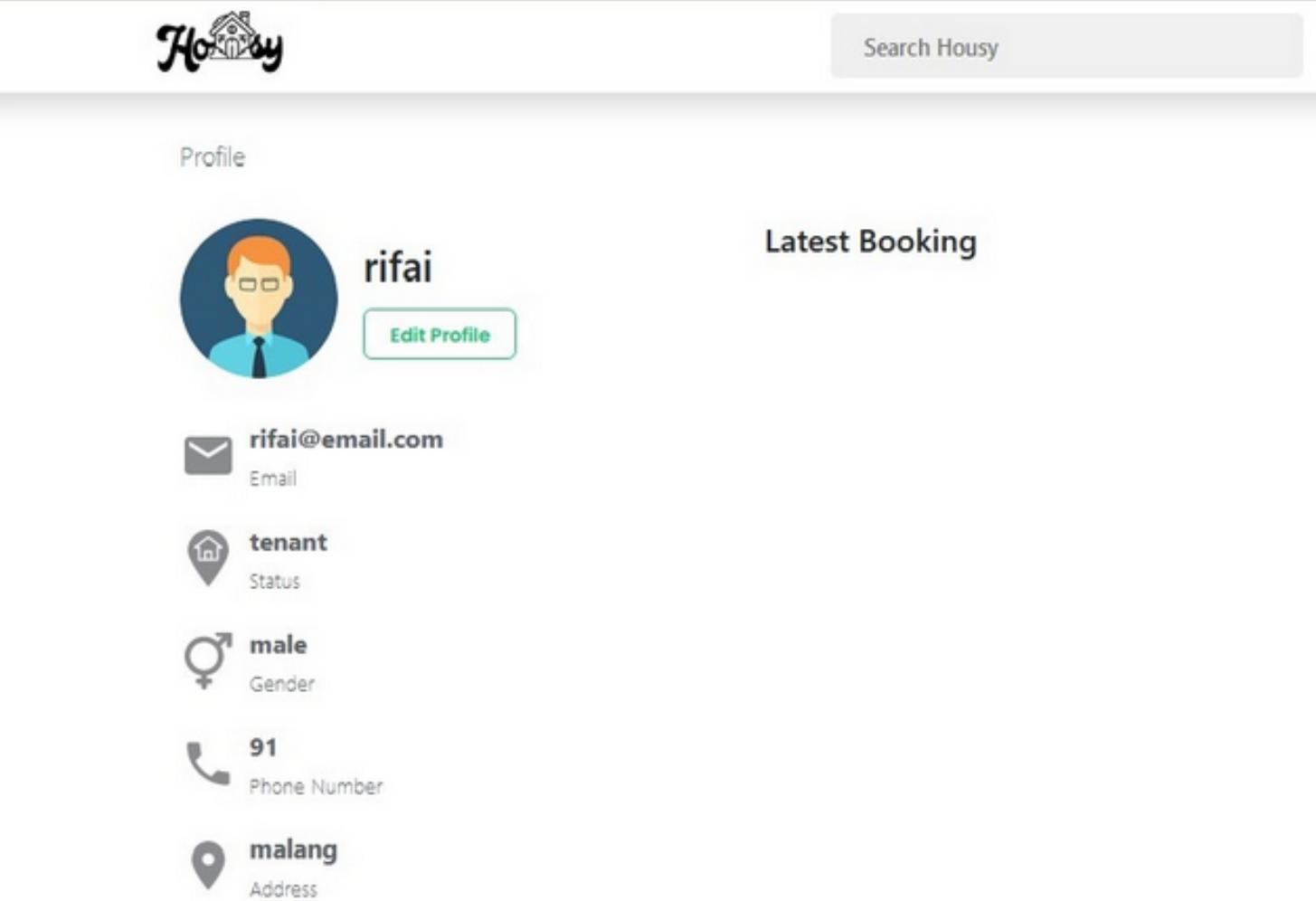
ubuntu@app:~/housy-backend$ cat config/config.json
{
  "development": {
    "username": "root",
    "password": "root",
    "database": "housy",
    "host": "database.housy.rifai.xyz",
    "dialect": "mysql"
  },
  "test": {
    "username": "root",
    "password": null,
    "database": "database_test",
    "host": "127.0.0.1",
    "dialect": "mysql"
  },
  "production": {
    "use_env_variable": "DATABASE_URL",
    "dialect": "postgres",
    "protocol": "postgres"
  }
}
```

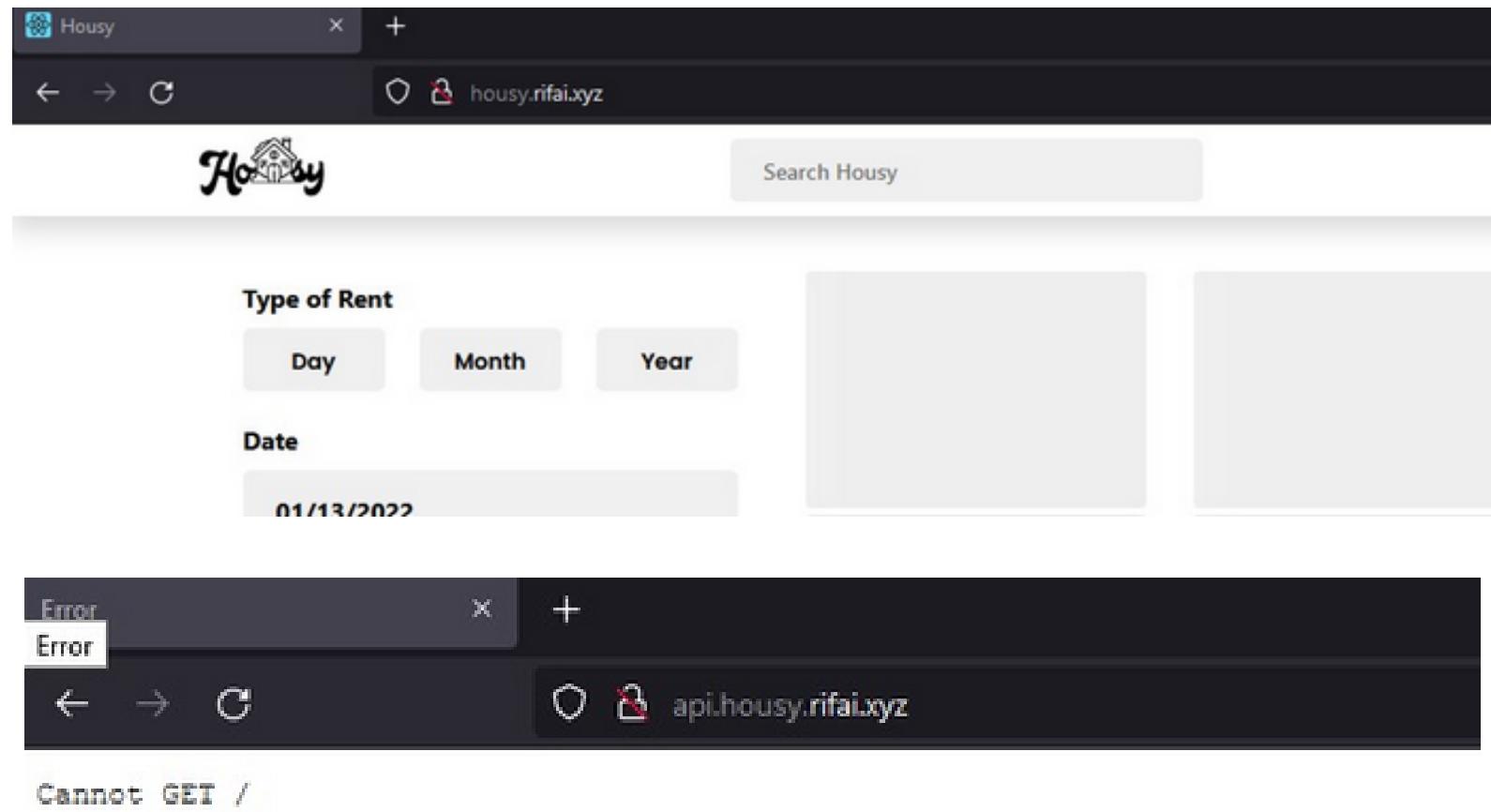
Deployment

- Integration housy frontend, backend and databaser

Untuk frontend, diubah baseURL dalam api.js kedalam alamat IP backend

Untuk backend host dalam config.js diubah ke alamat database





Deployment

- Reverse proxy

Frontend dapat diakses melalui
housy.rifai.xyz
backend
api.housy.rifai.xyz

CICD

- Use Jenkins docker
- setup job
- trigger by push



```
docker-compose-jenkins.yml
---
version: '3.9'
services:
  jenkins:
    image: jenkins/jenkins:lts
    ports:
      - 8080:8080
      - 50000:50000
    privileged: true
    user: root
    container_name: jenkins
    volumes:
      - ~jenkins:/var/jenkins_home
      - /var/run/docker.sock:/var/run/docker.sock
      - /usr/local/bin/docker:/usr/local/bin/docker
```

CI/CD

- Deploy jenkins use docker

```
setup-jenkins.yml
---
- name: Setup Jenkins Docker
  hosts: app
  become: true
  tasks:
    - name: Copy docker compose ci/cd
      copy:
        src: docker-compose-cicd.yml
        dest: /home/ubuntu/

    - name: Run docker compose
      shell: "docker-compose -f docker-compose-cicd.yml up -d"
      args:
        executable: /bin/bash
```

CI/CD

- Setup job

Enter an item name

» Required field

Freestyle project
This is the central feature of Jenkins. Jenkins will build your project, combining any SCM with any build system, and this can be even used for something other than software build.

Pipeline
Orchestrates long-running activities that can span multiple build agents. Suitable for building pipelines (formerly known as workflows) and/or organizing complex activities that do not easily fit in free-style job type.

Multi-configuration project
Suitable for projects that need a large number of different configurations, such as testing on multiple environments, platform-specific builds, etc.

Folder OK
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder creates a

frontend-development

General Source Code Management Build Triggers Build Environment Build Post-build Actions

Description

[Plain text] Preview

Discard old builds
 GitHub project
Project url

This build requires lockable resources

General Source Code Management Build Triggers Build Environment Build Post-build Actions

Execute concurrent builds if necessary

Source Code Management

None
 Git

Repositories

Repository URL

Credentials

Add Repository

Branches-to-build

Save Apply

Build Triggers

Trigger builds remotely (e.g., from scripts)
 Build after other projects are built
 Build periodically
 GitHub Branches
 GitHub Pull Requests
 GitHub hook trigger for GITScm polling
 Poll SCM

CI/CD

- Setup job

A screenshot of the GitHub settings interface for the repository 'rifaicham/housy-frontend'. The 'Webhooks' tab is selected in the sidebar. A red box highlights the 'Add webhook' button in the top right corner of the main content area.

A screenshot of the 'Webhooks / Add webhook' configuration page. The 'Payload URL' field contains 'http://172.19.159.26:8080/github-webhook/'. The 'Content type' dropdown is set to 'application/json'. Under 'Which events would you like to trigger this webhook?', the 'Just the push event.' option is selected. The 'Active' checkbox is checked. A red box highlights the 'Payload URL' field and the 'Content type' dropdown.

A screenshot of a CI build status card. It shows a green checkmark icon and the text 'Build #3 (Jan 14, 2022, 3:58:51 PM)'. Below it, there's a 'Changes' section with a list item: '1. update red.md by rifai (commit: a540f36) (details / githubweb)'. Other sections include 'Started by user rifai', 'Revision: a540f368c81d20ce9fb7475b8177489871d5cbcd', 'Repository: https://github.com/rifaicham/housy-frontend.git', and a list of branches. A red box highlights the first item in the 'Changes' list.

- Setup job backend

The screenshot shows the Jenkins job configuration page for a job named 'housy-backend'. The configuration tabs at the top are General, Source Code Management, Build Triggers, Build Environment, Build, and Post-build Actions. The General tab is selected.

General

- Discard old builds
- GitHub project
 - Project url: `https://github.com/rifaicham/housy-backend`
- This build requires lockable resources
- This project is parameterized
- Throttle builds
- Disable this project
- Execute concurrent builds if necessary

Source Code Management

- None
- Git
 - Repository: `https://github.com/rifaicham/housy-backend.git`
 - Credentials: `- none -` ()
 -
-
- Branches to build**

 - Branch Specifier (blank for any): `*/development`
 -

- Repository browser**: `(Auto)`
- Additional Behaviours**

 -

Build Triggers

- Trigger builds remotely (e.g., from a script)
- Build after other projects are built
- Build periodically
- GitHub Branches
- GitHub Pull Requests
- GitHub hook trigger for GITScm polling
- Poll SCM

Build Environment

- Delete workspace before build starts
- Use secret text(s) or file(s)

Console Output

Started by user rifaai
Running as SYSTEM
Building in workspace /var/lib/jenkins/workspace/backend-development
The recommended git tool is: NONE
No credentials specified

```
> git rev-parse --resolve-git-dir /var/lib/jenkins/workspace/backend-development/.git # timeout=10
Fetching changes from the remote Git repository
> git config remote.origin.url https://github.com/rifaicham/housy-backend.git # timeout=10
Fetching upstream changes from https://github.com/rifaicham/housy-backend.git
> git --version # timeout=10
> git --version # 'git version 2.25.1'
> git fetch --tags --force --progress -- https://github.com/rifaicham/housy-backend.git +refs/heads/*:refs/remotes/origin/* # timeout=10
> git rev-parse refs/remotes/origin/development^{commit} # timeout=10
Checking out Revision fe9838859a2ffff40b0d6637a80719af1afe8b0e (refs/remotes/origin/development)
> git config core.sparsecheckout # timeout=10
> git checkout ^fe9838859a2ffff40b0d6637a80719af1afe8b0e # timeout=10
Commit message: "update readme by rifaai"
> git rev-list --no-walk fe9838859a2ffff40b0d6637a80719af1afe8b0e # timeout=10
Finished: SUCCESS
```



Limitasi dalam CI/CD menggunakan jenkins dalam multipass windows10

- Publish over SSH disuspend oleh jenkins, jadi untuk instalasi pluggin tidak akan berhasil

A screenshot of a web browser showing the Jenkins plugin distribution page for 'publish-over-ssh'. The URL is <https://plugins.jenkins.io/publish-over-ssh/>. The page shows a warning message: 'Plugin distribution has been suspended, see <https://www.jenkins.io/security/plugins/#suspensions> for details.' A yellow warning bar at the bottom also displays this information.

- Default di jenkins tidak mendukung perintah sudo, maka ketika digunakan bersamaan dengan multipass akan menimbulkan proses build gagal

A screenshot of a GitHub issue page for <https://github.comcanonical/multipass/issues/1437>. The issue is closed and titled 'Cannot run multipass as a non-admin user on macOS or Linux #1437'. A comment from 'Saviq' on April 26, 2020, states: '@pixlwave Multipass service / daemon has to run as root to be able to access certain system facilities (at the very least create virtual network devices etc). There's just no way to run the service as non-root. Running `multipass`, the command line client, without being a member of `sudo`, `admin` or some other administrative group is something we'll work on, soon.' This comment is highlighted with a red box.

```
ubuntu@gabungan:~$ sudo usermod -aG docker $USER
ubuntu@gabungan:~$ docker ps
Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Get "http://<2Fvar%2Frund%2Fdocker.sock/v1.24/containers/json": dial unix /var/run/docker.sock: connect: permission denied
ubuntu@gabungan:~$ sudo docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS NAMES
```

```
15:28:12 Cloning into 'housy-frontend'...
15:28:13 + cd housy-frontend
15:28:13 + docker build -t housy-frontend .
15:28:14 Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post "http://<2Fvar%2Frund%2Fdocker.sock/v1.24/build?buildargs=%7B%7D&cachedfrom=%5B%5D&cgroupparent=&cpuperiod=0&cpuquota=0&cpusetcpus=&cpusetmems=&cpushares=0&dockerfile=dockerfile&labels=%7B%7D&memory=0&memswap=0&networkmode=default&rm=1&shmsize=0&t=housy-frontend&target=&ulimits=null&version=1": dial unix /var/run/docker.sock: connect: permission denied
15:28:14 Build step 'Execute shell' marked build as failure
```

MONITORING

- Build with docker
- monitoring dashboard



MONITORING

- Build with Docker

```
version: '3'

services:
  node_exporter:
    image: prom/node-exporter:latest
    container_name: node_exporter
    ports:
      - 9100:9100
    command:
      - '--path.procfs=/host/proc'
      - '--path.rootfs=/rootfs'
      - '--path.sysfs=/host/sys'
      - '--collector.filesystem.mount-points-exclude=^(sys|proc|dev|host|etc)(\$\$|/)'
    restart: unless-stopped
    volumes:
      - /proc:/host/proc:ro
      - /sys:/host/sys:ro
      - /:/rootfs:ro
```

```
prometheus.yml
---
global:
  scrape_interval: 5m

scrape_configs:
  - job_name: "prometheus-metrics"
    scrape_interval: 5m
    static_configs:
      - targets: ['172.19.146.231:9100']
  - job_name: "node_exporter_metrics"
    scrape_interval: 5m
    scrape_timeout: 1m
    tls_config:
      insecure_skip_verify: true
    static_configs:
      - targets: ['172.19.147.152:9100','172.19.158.16:9100']
```

```
docker-compose-monitoring.yml
---
version: '3'
services:
  prometheus:
    image: prom/prometheus:latest
    container_name: prometheus
    restart: unless-stopped
    volumes:
      - /home/ubuntu/prometheus.yml:/etc/prometheus/prometheus.yml
      - /home/ubuntu/web.yml:/etc/prometheus/web.yml
    command:
      - '--config.file=/etc/prometheus/prometheus.yml'
      - '--web.config.file=web.yml'
      - '--storage.tsdb.path=prometheus'
      - '--web.console.libraries=/etc/prometheus/console_libraries'
      - '--web.console.templates=/etc/prometheus/consoles'
      - '--web.enable-lifecycle'
    ports:
      - 9090:9090

  grafana:
    image: grafana/grafana:latest-ubuntu
    container_name: grafana
    volumes:
      - /home/ubuntu/grafana/data:/var/lib/grafana
    ports:
      - 3000:3000
    user: "1000"
    restart: unless-stopped
```

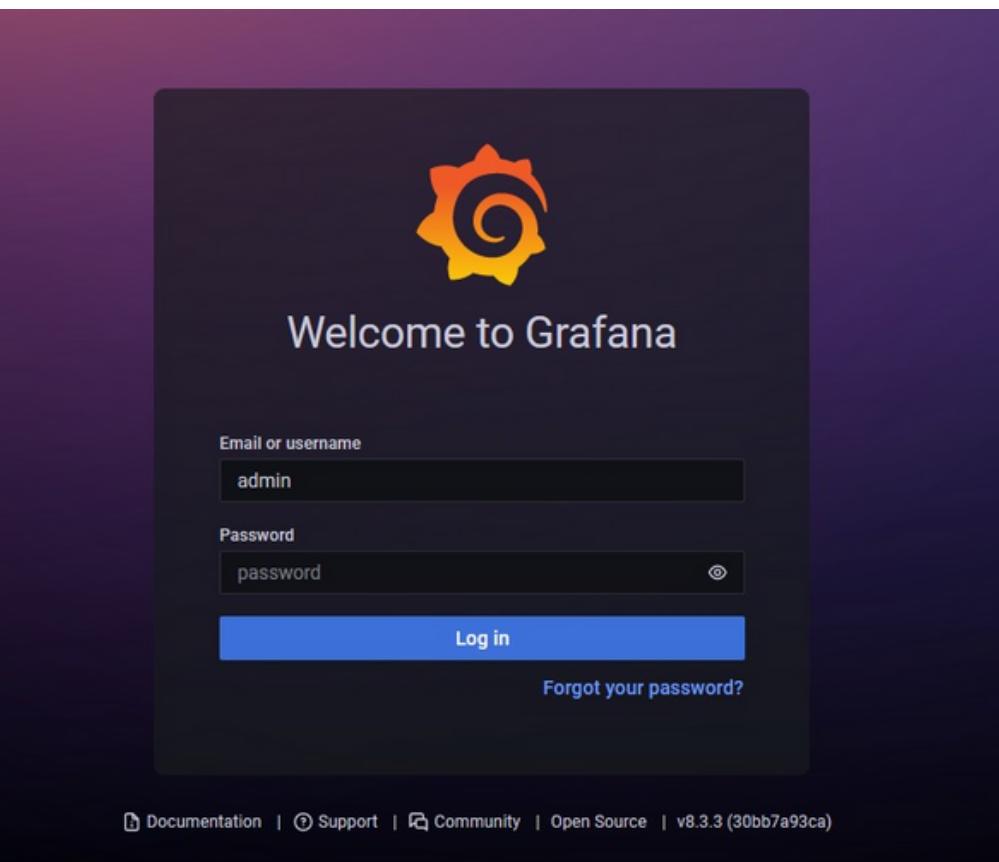
```
setup-monitoring.yml
---
- name: Setup monitoring
hosts: monitoring
become: true
tasks:
- name: Copying docker compose monitoring file
copy:
  src: setup-monitoring.yml
  dest: /home/ubuntu/

- name: Copying prometheus.yml file
copy:
  src: files/monitoring/prometheus.yml
  dest: /home/ubuntu/

- name: Copying web.yml file
copy:
  src: files/monitoring/web.yml
  dest: /home/ubuntu/

- name: Run compose up
shell:
  cmd: "docker-compose -f setup-monitoring.yml up -d"

- name: Change grafana folder permission
shell: "sudo chown 1000:1000 grafana/data/"
args:
  executable: /bin/bash
```



MONITORING

- Build with Docker

Setelah disiapkan semua filenya

- prometheus.yml
- node_exporter
- docker-compose untuk deploy monitoring dalam container
- ansible-playbook untuk monitoring

maka bisa dijalankan dengan perintah

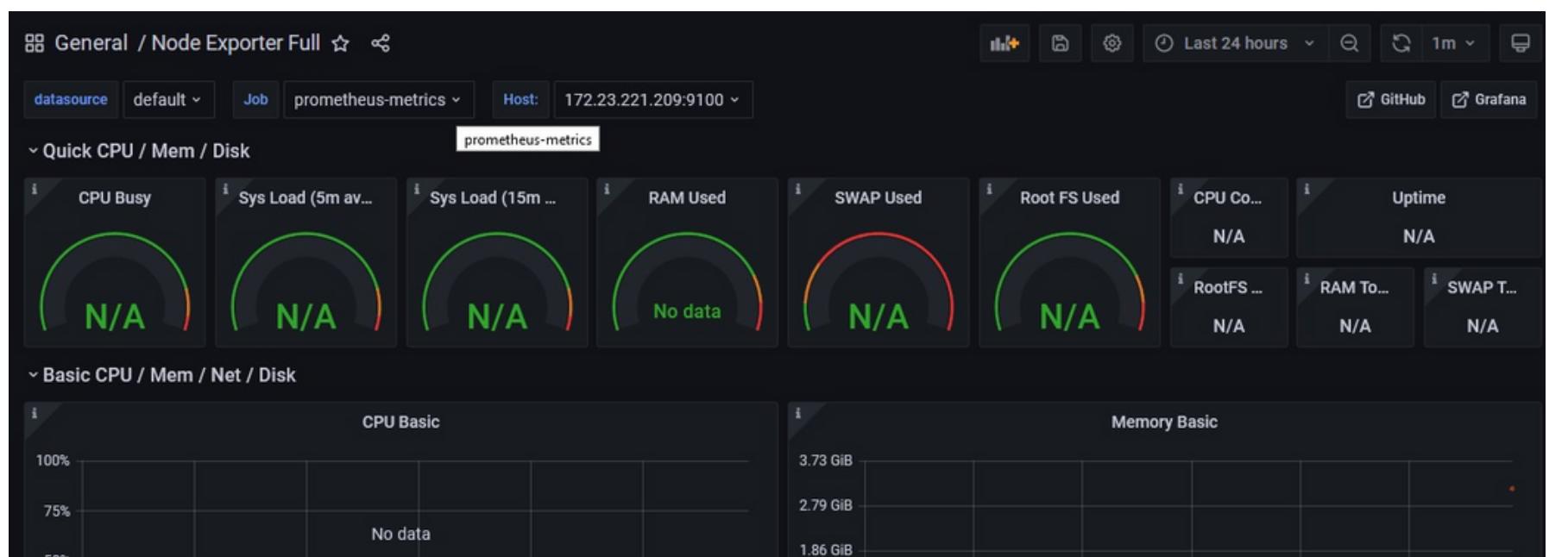
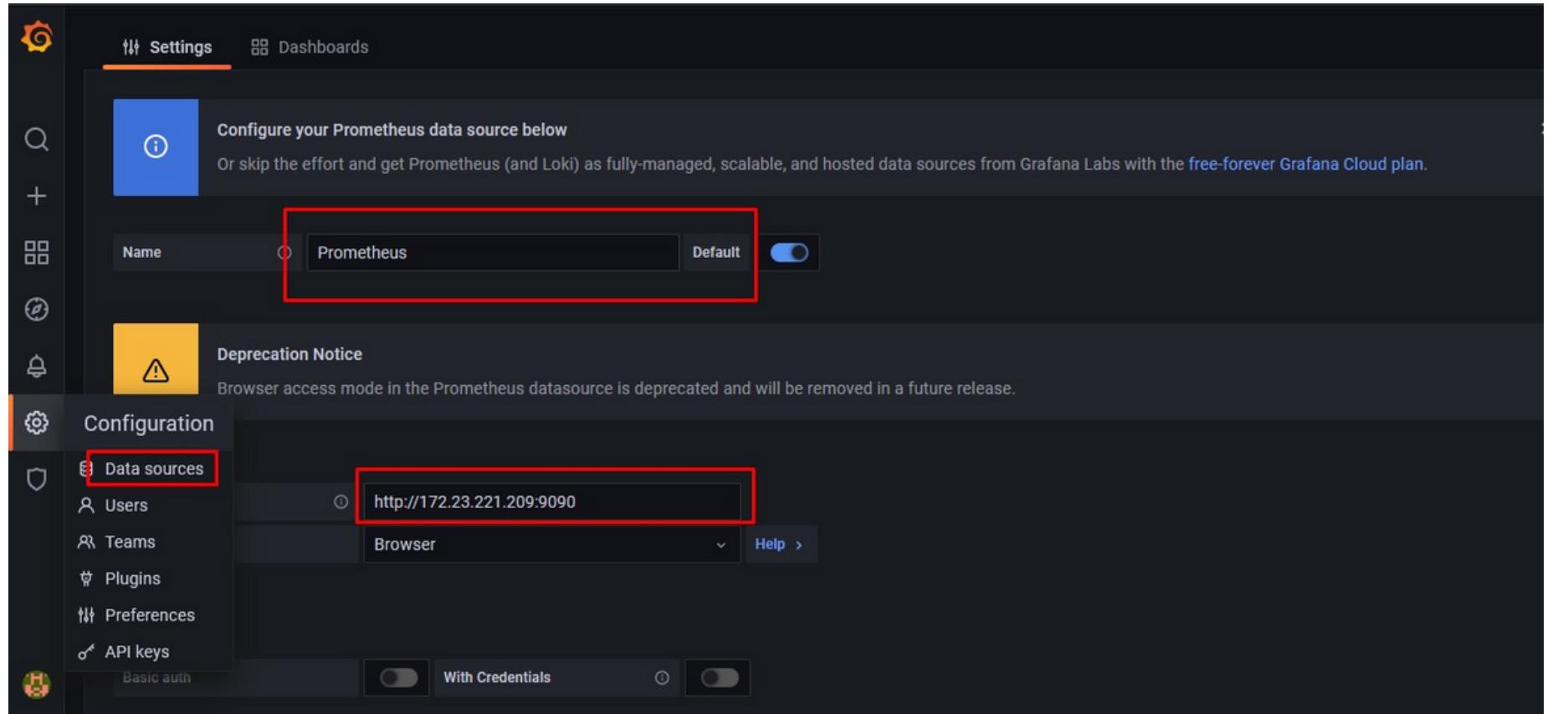
ansible-playbook -t setup-monitoring main.yml

Setelah proses selesai maka dapat diakses melalui ipmonitoring:3000 atau via reverse proxy yang sudah disetel sebelumnya

MONITORING

- Monitoring Dashboard

Buat Data source terlebih dahulu dan pilih prometheus



Dashboard atau tampilan untuk monitoring server bisa build sendiri atau meng-import dari grafana.com/dashboard

AUTH

- Amankan prometheus web akses



AUTH

- Amankan prometheus

Install apache2-utils
sudo apt install apache2-utils
generate password dengan htpasswd
sudo htpasswd -c /etc/nginx/.htpasswd <username>
didalam konfigurasi nginx
prometheus.conf dapat ditambah
dengan

```
ubuntu@server1:~$ sudo apt install apache2-utils
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libapr1 libaprutil1
The following NEW packages will be installed:
  apache2-utils libapr1 libaprutil1
0 upgraded, 3 newly installed, 0 to remove and 0 not upgraded.
Need to get 260 kB of archives.
After this operation, 970 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://archive.ubuntu.com/ubuntu focal/main amd64 libapr1 amd64 1.6.5-1ubuntu1 [91.4 kB]
Get:2 http://archive.ubuntu.com/ubuntu focal/main amd64 libaprutil1 amd64 1.6.1-4ubuntu2 [84.7 kB]
Get:3 http://archive.ubuntu.com/ubuntu focal-updates/main amd64 apache2-utils amd64 2.4.41-4ubuntu3.9 [84.3 kB]
Fetched 260 kB in 2s (120 kB/s)
Selecting previously unselected package libapr1:amd64.
(Reading database ... 64050 files and directories currently installed.)
Preparing to unpack .../libapr1_1.6.5-1ubuntu1_amd64.deb ...
Unpacking libapr1:amd64 (1.6.5-1ubuntu1) ...
Selecting previously unselected package libaprutil1:amd64.
Preparing to unpack .../libaprutil1_1.6.1-4ubuntu2_amd64.deb ...
Unpacking libaprutil1:amd64 (1.6.1-4ubuntu2) ...
Selecting previously unselected package apache2-utils.
Preparing to unpack .../apache2-utils_2.4.41-4ubuntu3.9_amd64.deb ...
Unpacking apache2-utils (2.4.41-4ubuntu3.9) ...
Setting up libapr1:amd64 (1.6.5-1ubuntu1) ...
Setting up libaprutil1:amd64 (1.6.1-4ubuntu2) ...
Setting up apache2-utils (2.4.41-4ubuntu3.9) ...
Processing triggers for man-db (2.9.1-1) ...
Processing triggers for libc-bin (2.31-0ubuntu9.2) ...
ubuntu@server1:~$ []
```

```
ubuntu@server1:~$ sudo htpasswd -c /etc/nginx/.htpasswd rifai
New password:
Re-type new password:
Adding password for user rifai
ubuntu@server1:~$ []
```

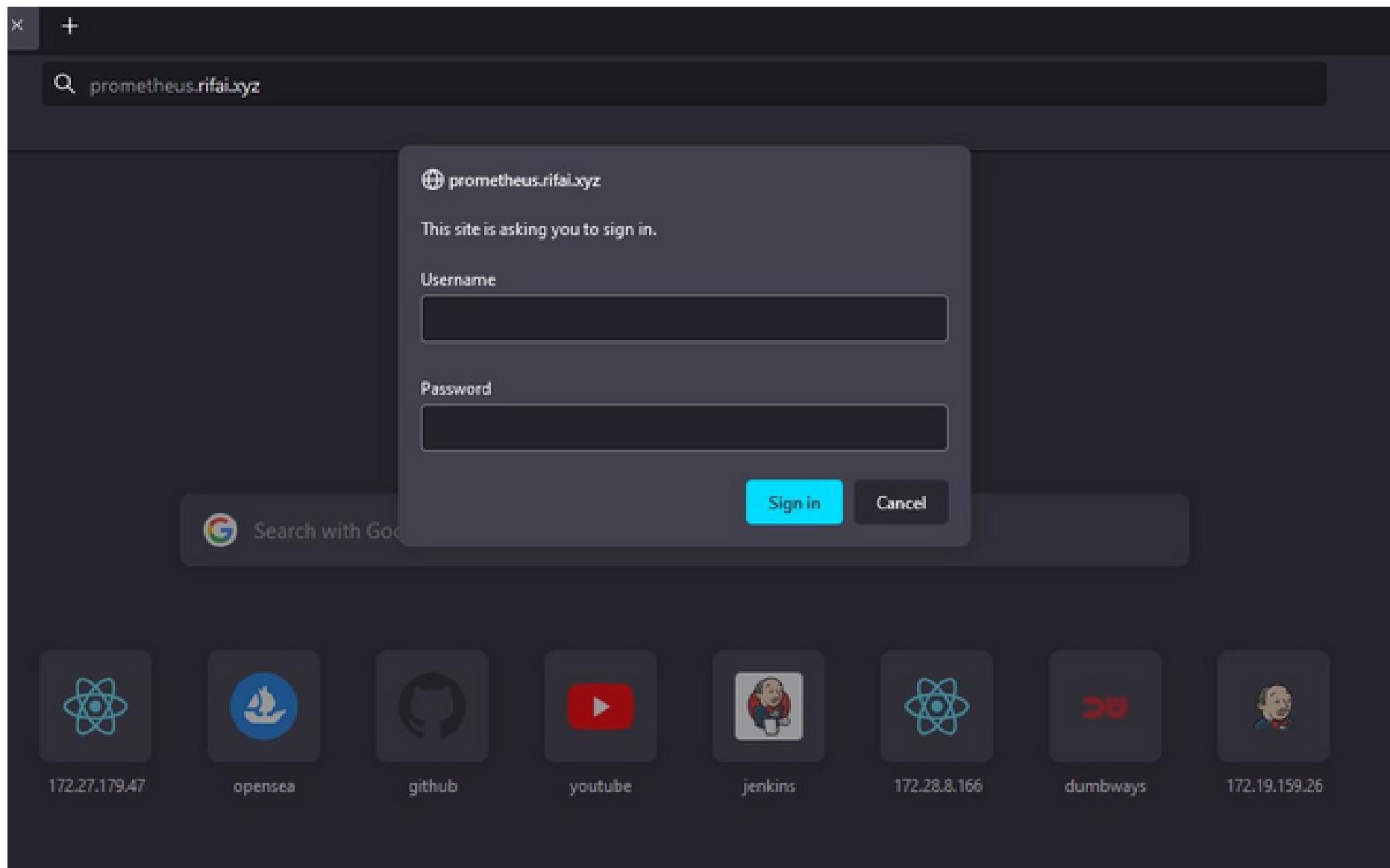
```
GNU nano 4.8          /etc/nginx/conf.d/prometheus.conf
server {
  server_name prometheus.rifai.xyz;

  location / {
    auth_basic "prometheus";
    auth_basic_user_file /etc/nginx/.htpasswd;

    proxy_pass http://172.27.181.114:9090;
  }
}
```

AUTH

- Amankan prometheus



Akses webserver prometheus, maka akan muncul untuk memasukkan username dan password yang sudah diset sebelumnya