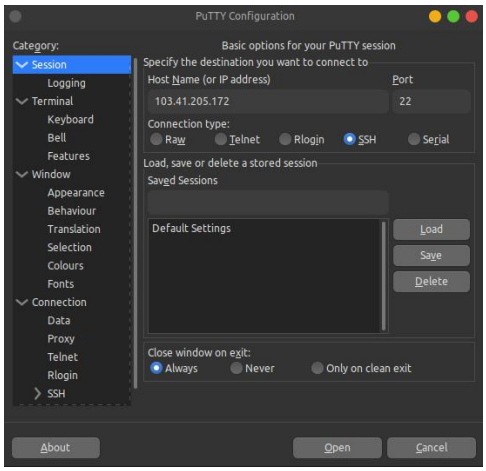
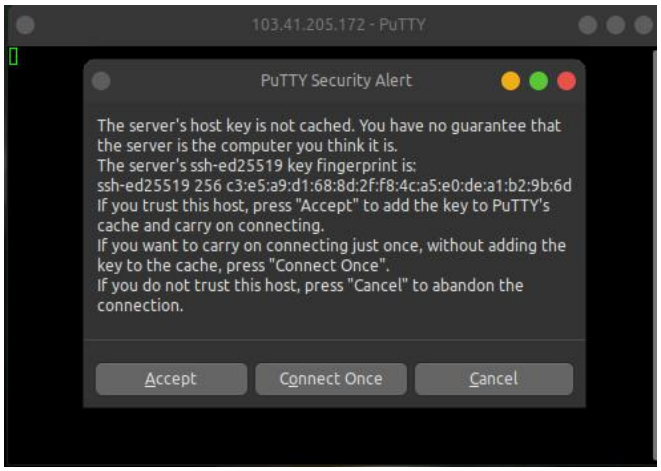
Remote Server

1. Masukkan IP server, klik open

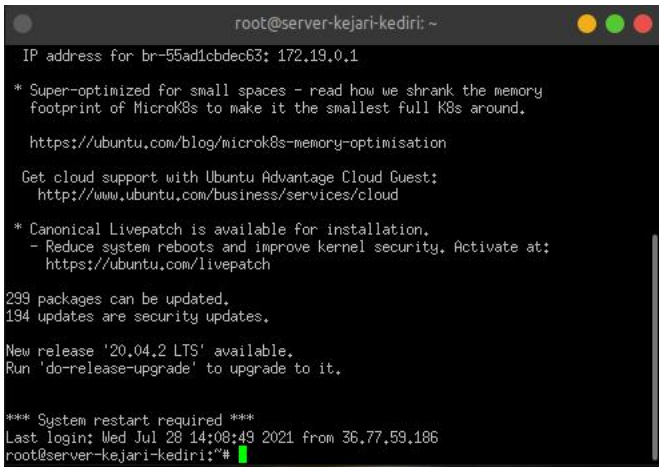


2. Pilih Accept



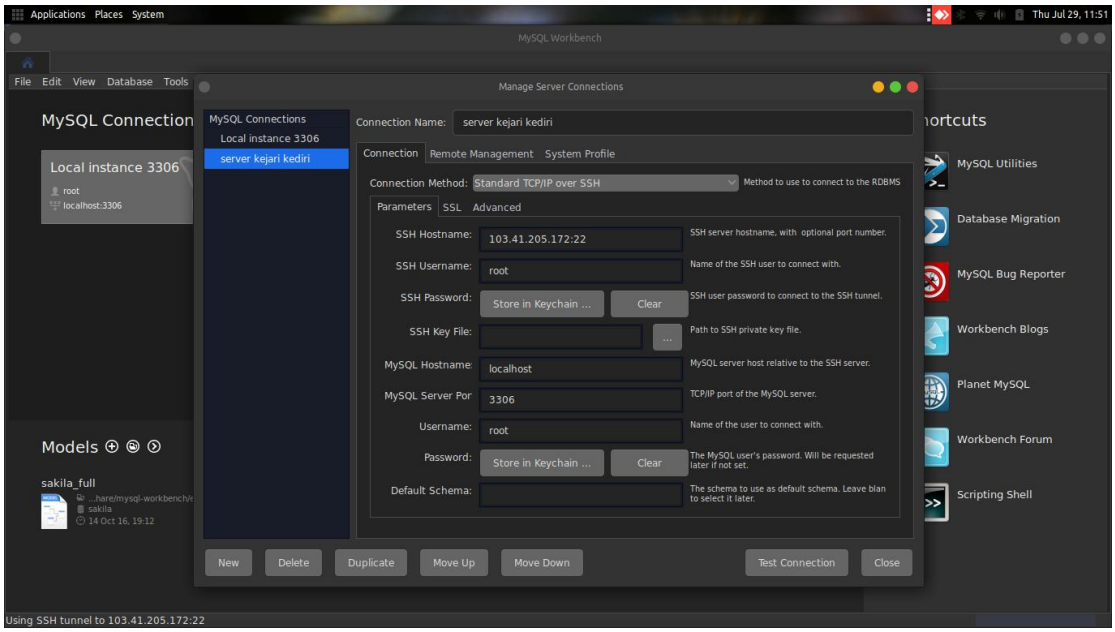
3. Masukkan Username dan Password

4. Jika login berhasil

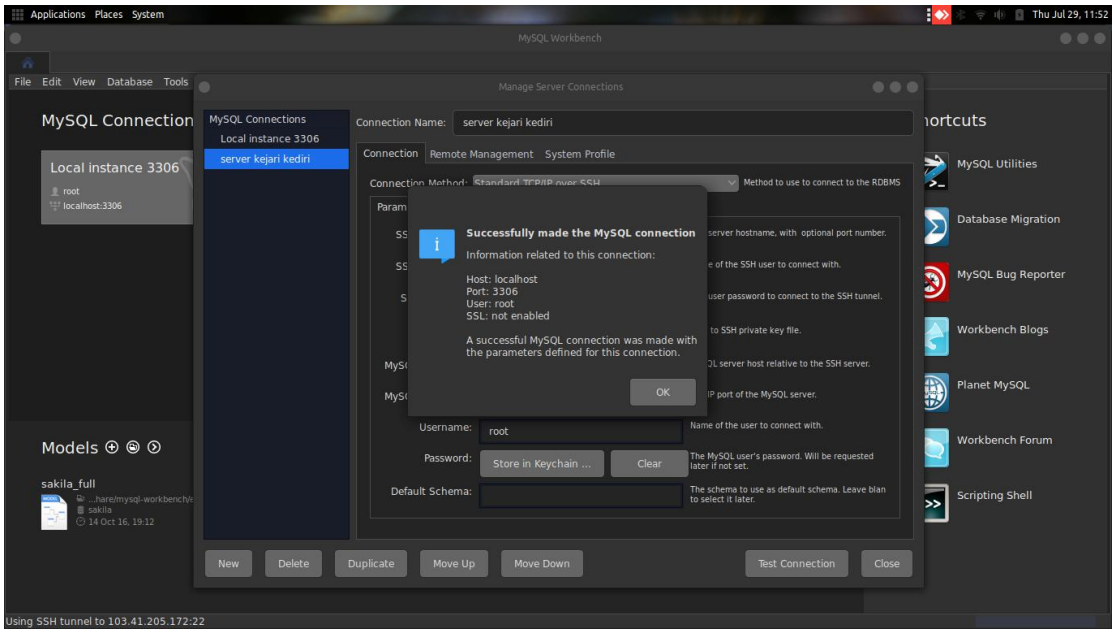


Remote Database

1. Gunakan DBMS tool untuk mengelola database (MySQL Workbanch)
2. Buat koneksi database

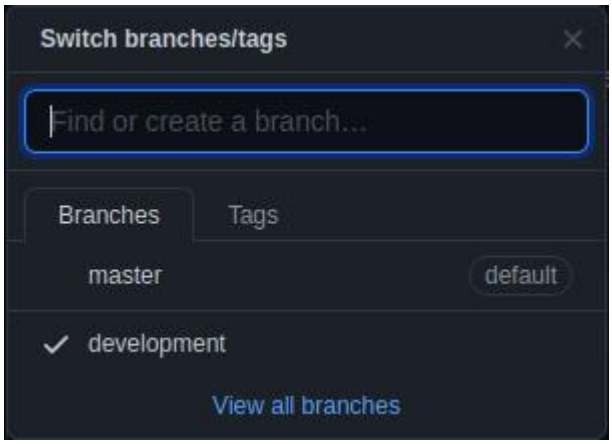


3. Tes koneksi

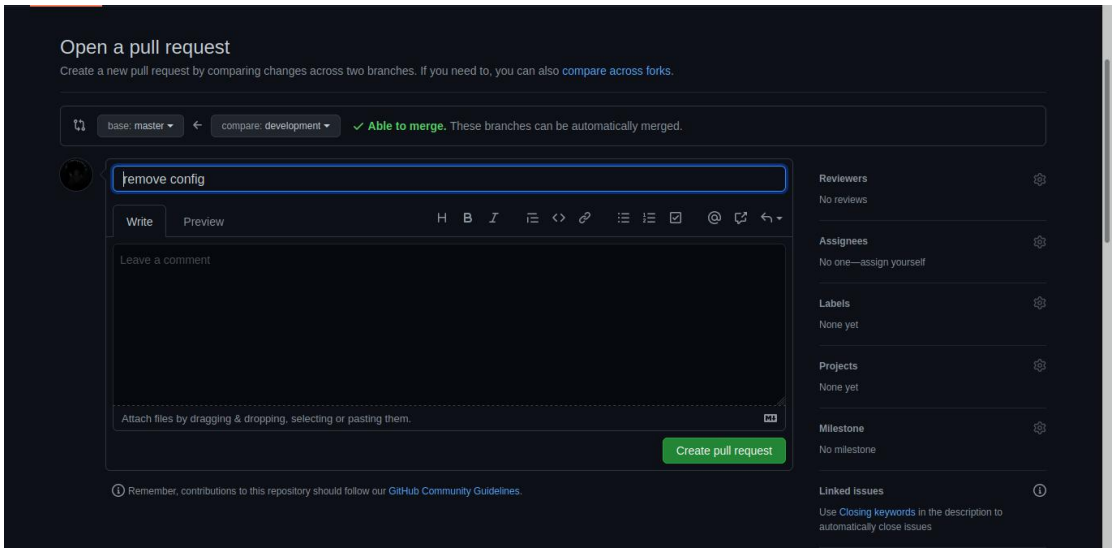


Development

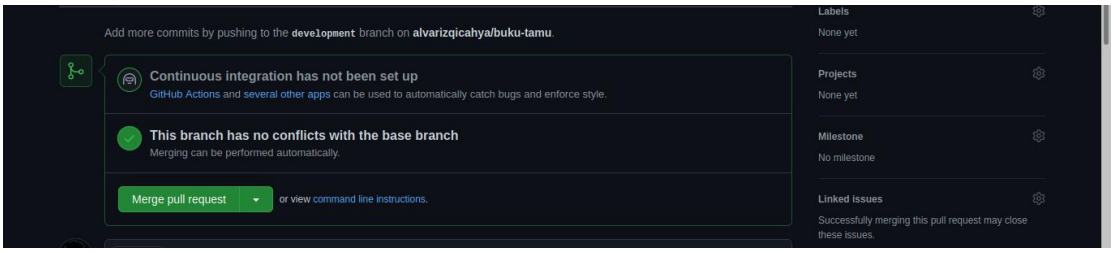
1. Diusahakan menggunakan git (github)
2. Gunakan branch yang berbeda untuk development dan production



3. Buat pull request ke branch utama ketika program sudah melewati tahap testing

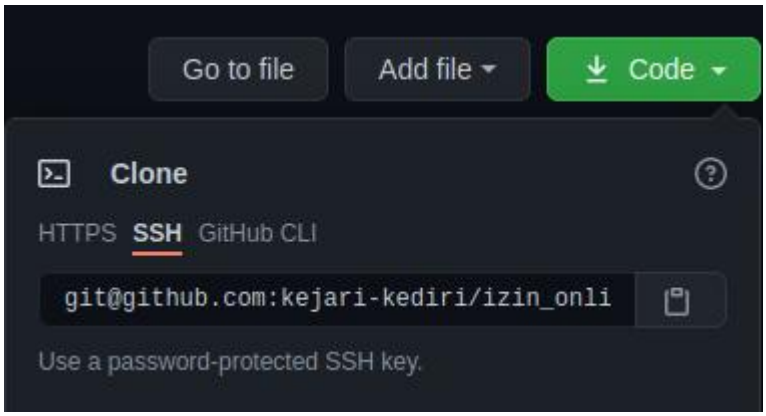


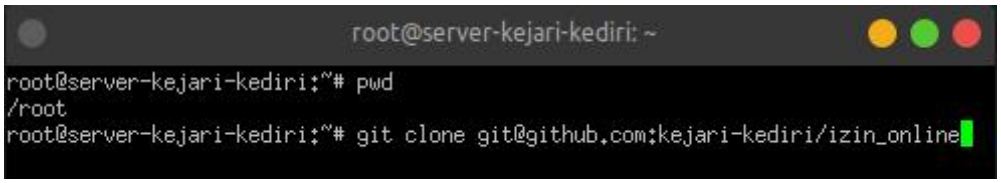
4. Setelah memastikan tidak ada error pada program lakukan Merge pull request



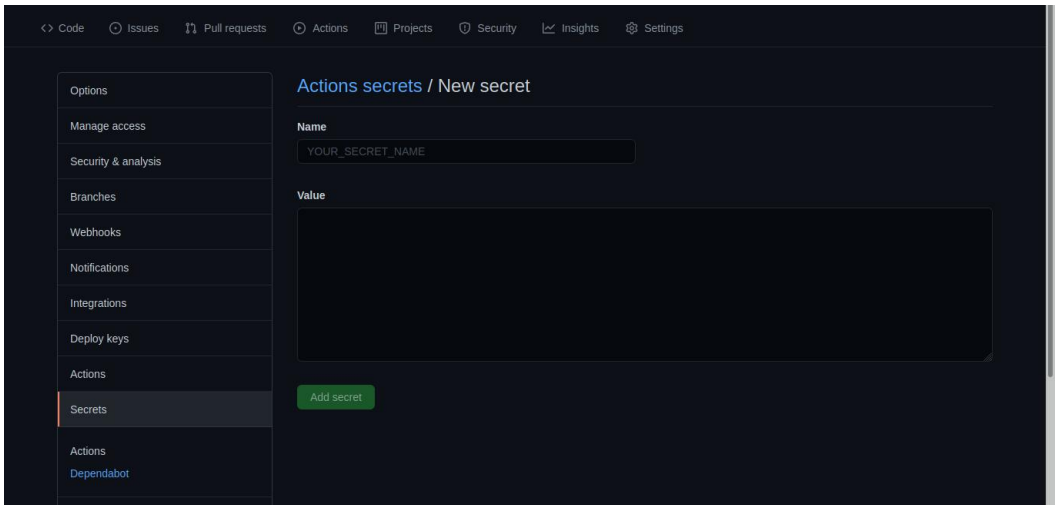
Deployment

1. Masuk ke server menggunakan remote ssh
2. Cloning repository via ssh, pastikan berada pada directory /root





3. Integrasikan github dengan ssh server, masukkan sesuai konfigurasi ssh server seperti hostname, username, password dan port. Contoh; Name: HOSTNAME, Value: 103.41.205.172
   Action secret digunakan untuk payload workflows pada github.



4. Karena menggunakan Docker pastikan pada project terdapat Dockerfile, payload Dockerfile sesuaikan dengan kebutuhan framework yang digunakan.

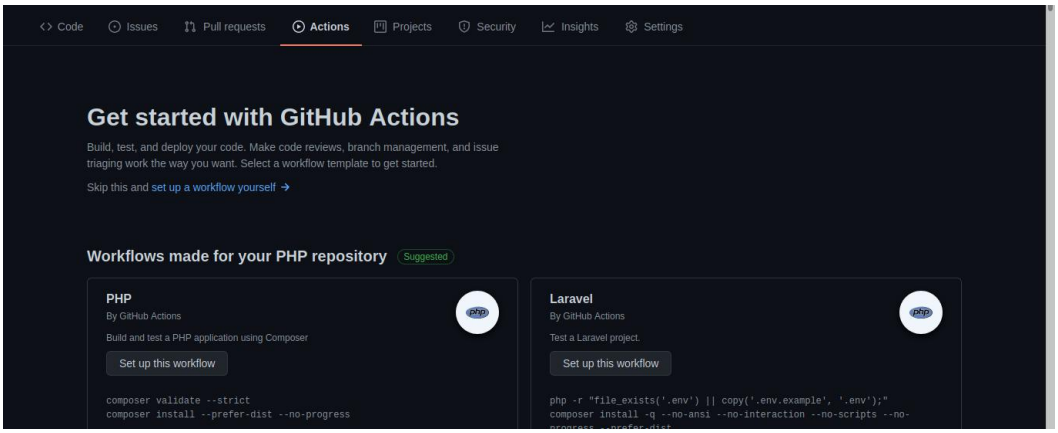
   Contoh payload Dockerfile:

```
FROM php:7.3.5-apache-stretch
```

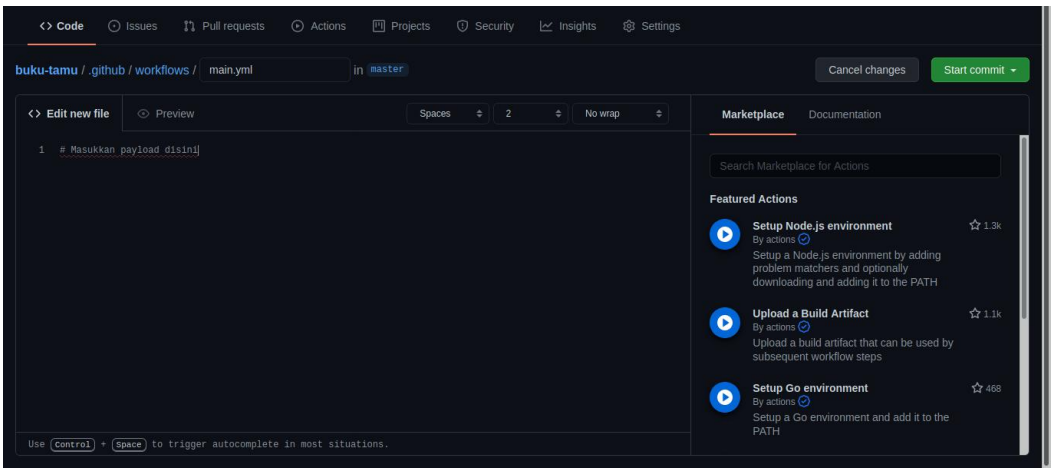
```
RUN apt-get update && \
    apt-get install -y nano && \
    docker-php-ext-install \
        mysqli \
        sockets \
        bcmath && \
    a2enmod rewrite
ENV WORKDIR /var/www/html
WORKDIR $WORKDIR
ADD . $WORKDIR
COPY . $WORKDIR
RUN ls && pwd
EXPOSE 80
```

5. Sebelum membuat payload workflows, cek terlebih dahulu port yang sudah digunakan pada docker, port yang sudah digunakan tidak boleh digunakan lagi

6. Buat workflow pada github



7. Buat payload workflows sesuai dengan kebutuhan



contoh payload workflows:

```
# This is a basic workflow to help you get started with Actions

name: CD

# Controls when the action will run.

on:

  # Triggers the workflow on push or pull request events but only for the master branch

  push:

    branches: [ master ]

  # Allows you to run this workflow manually from the Actions tab

  workflow_dispatch:


# A workflow run is made up of one or more jobs that can run sequentially or in parallel

jobs:
```

```
  # This workflow contains a single job called "build"
  build:
    # The type of runner that the job will run on
    runs-on: ubuntu-latest
    # Steps represent a sequence of tasks that will be executed as part
of the job
    steps:
      - name: Deploy Using SSH
        uses: appleboy/ssh-action@master
        with:
          host: ${{ secrets.HOST }}
          username: ${{ secrets.USERNAME }}
          password: ${{ secrets.PASSWORD }}
          port: ${{ secrets.PORT }}
          script: |
              cd izin_online
              git pull origin master
              git status
              docker build -t izin-online .
              docker rm e-izin -f
              docker run -e DB_HOST=database-mariadb -e DB_PORT=3306 -e
DB_USER=root -e DB_PASS=segopecel12 -e DB_NAME=izin -e
BASE_URL=https://izin.kejari-kediri.go.id/ -e
API_KEY_WA=C1TJ52r1xPGQq57gRmHSikvK5UbfVRgZrR9pHtVC5u3OK6a5ud9KnL
zyDtNVmGN5 --name e-izin -p 8002:80 -v "$PWD":/var/www/html --network
database-network -d izin-online
```

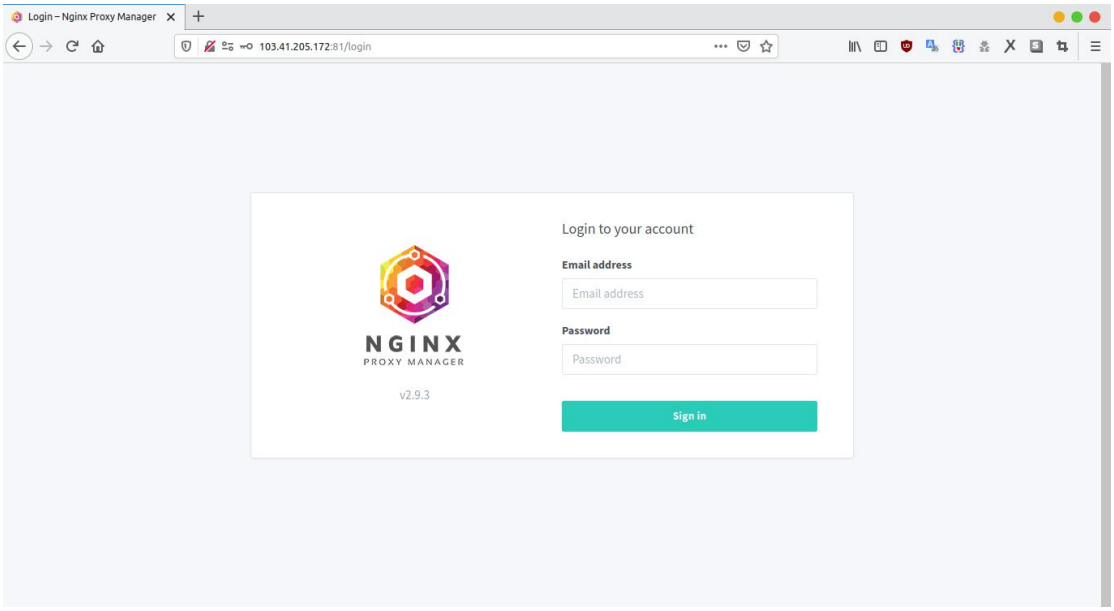
Ketentuan: branches sesauikan dengan branch utama, script sesuaikan dengan project masing - masing.

8. Lakukan Push atau Marge pull request pada github untuk deploy project ke server

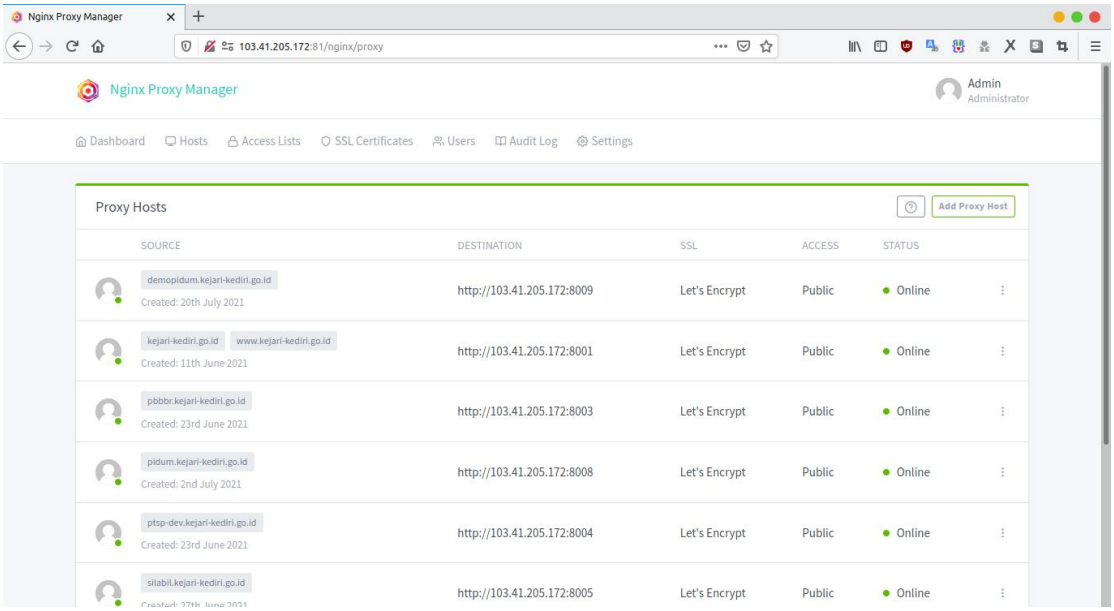
NB: Tahap tersebut hanya untuk project yang belum pernah terbangun integrasi antara github ke server. Jika sudah terbangun integrasi hanya perlu Push atau Marge pull request pada branch utama.

Mounting Domain

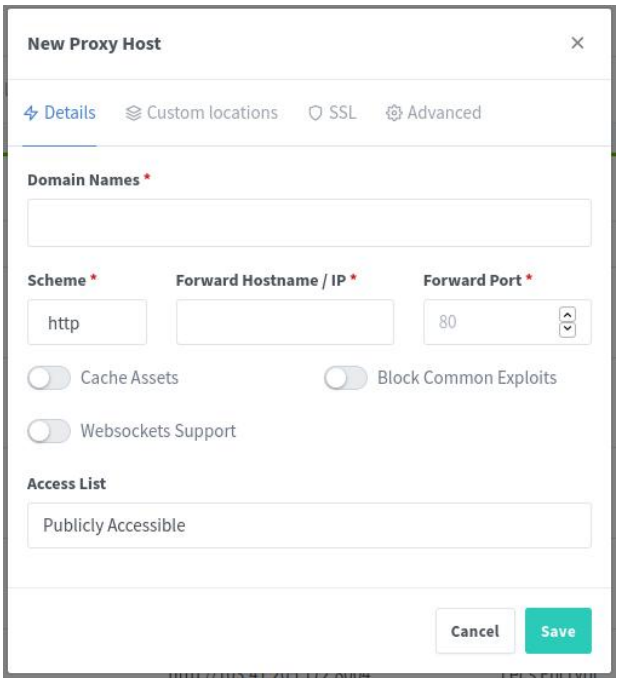
1. Buka nginx proxy manager

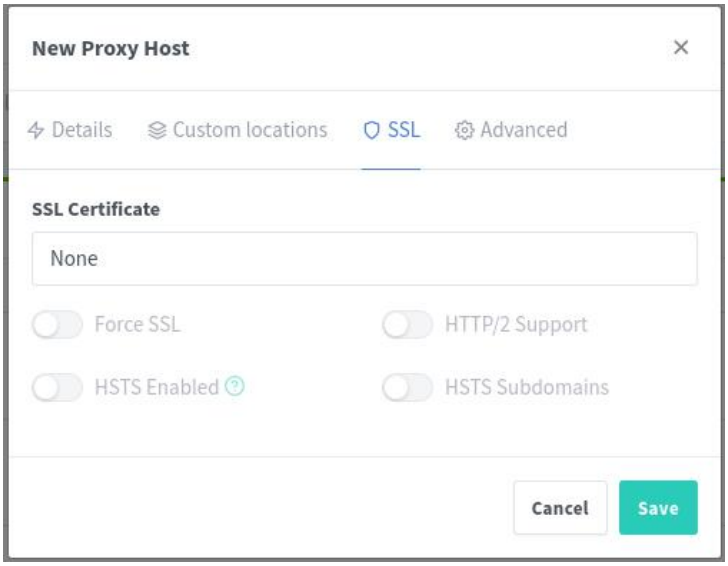


2. Masuk ke tab Proxy Hosts



3. Tambah Proxy Host. Pada tab Details isikan nama domain, ip server dan forward port (port docker). Pada tab SSL Requst a new SSL dan switch force SSL





4. Cek dengan mengakses domain yangg sudah dibuat