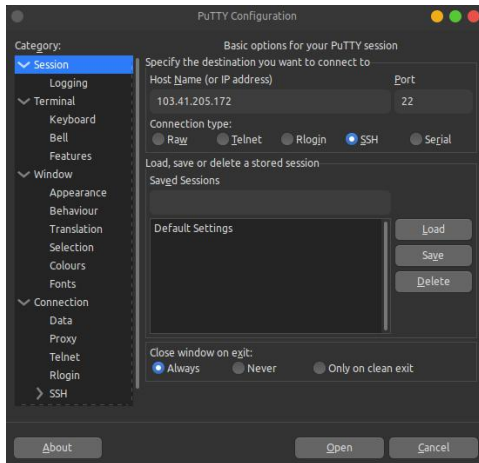


Remote Server

1. Masukkan IP server, klik open



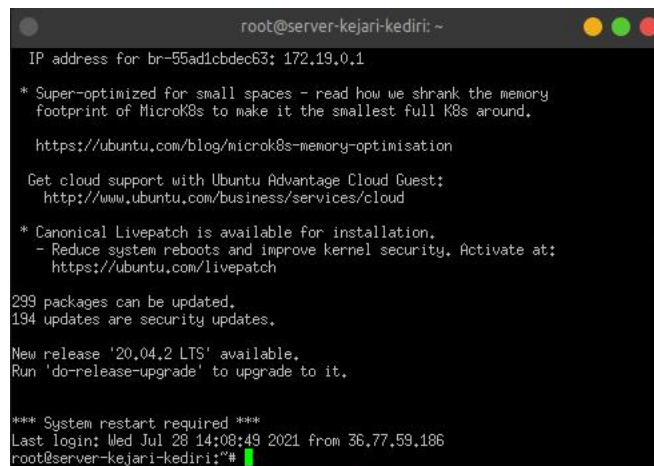
2. Pilih Accept



3. Masukkan Username dan Password



4. Jika login berhasil

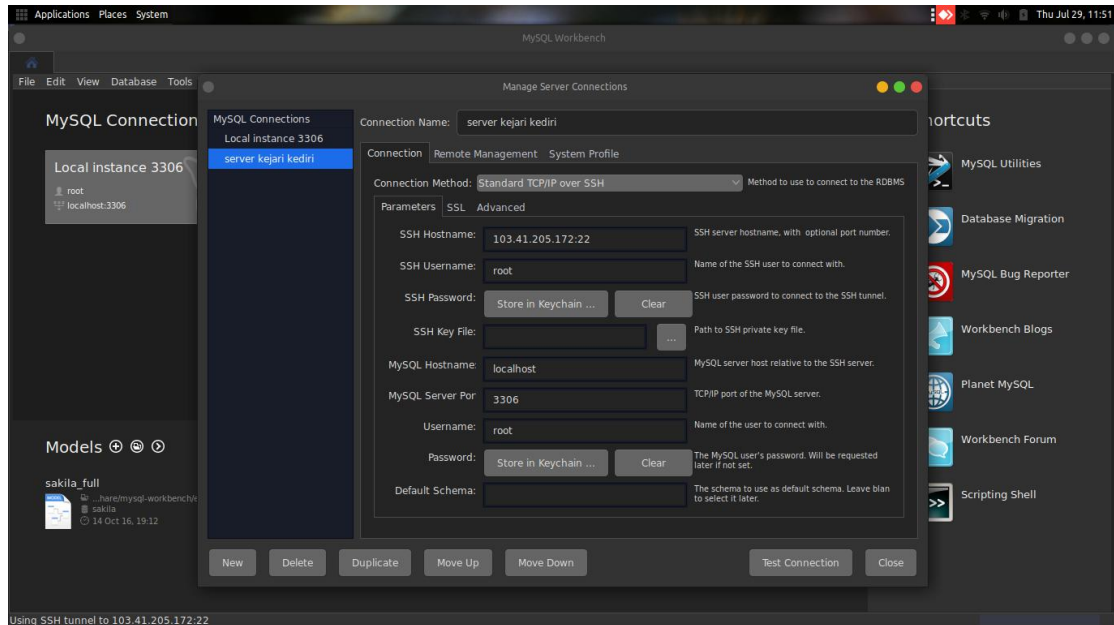


```
root@server-kejari-kediri: ~
IP address for br-55ad1cbdec63: 172.19.0.1
* Super-optimized for small spaces - read how we shrank the memory
  footprint of MicroK8s to make it the smallest full K8s around.
  https://ubuntu.com/blog/microk8s-memory-optimisation
Get cloud support with Ubuntu Advantage Cloud Guest:
  http://www.ubuntu.com/business/services/cloud
* Canonical Livepatch is available for installation.
  - Reduce system reboots and improve kernel security. Activate at:
    https://ubuntu.com/livepatch
299 packages can be updated.
194 updates are security updates.
New release '20.04.2 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

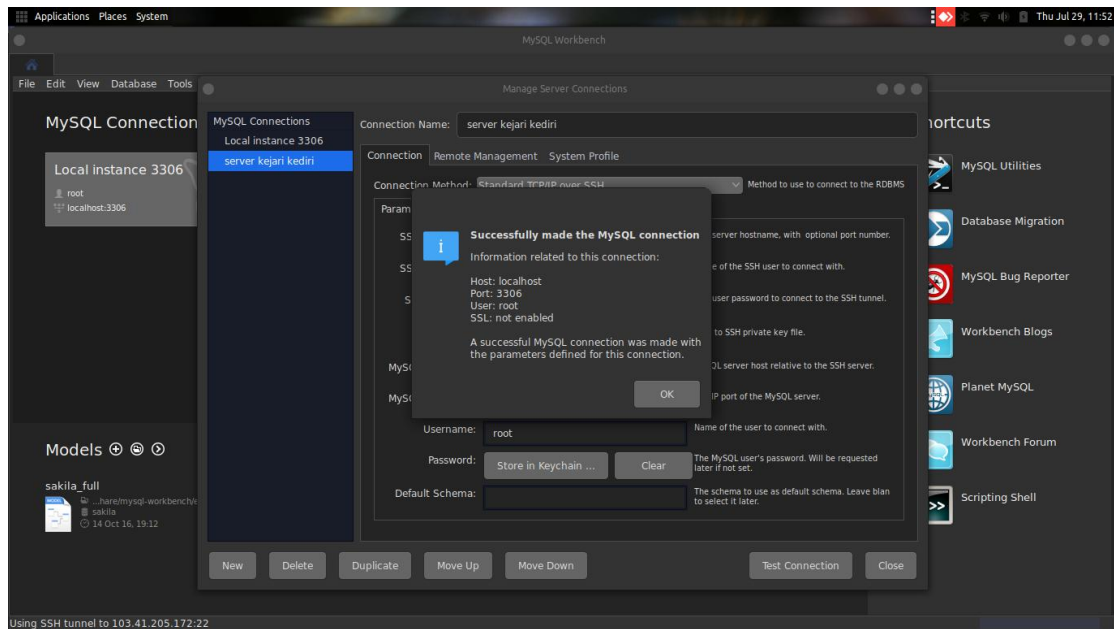
*** System restart required ***
Last login: Wed Jul 28 14:08:49 2021 from 36.77.59.186
root@server-kejari-kediri:~#
```

Remote Database

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

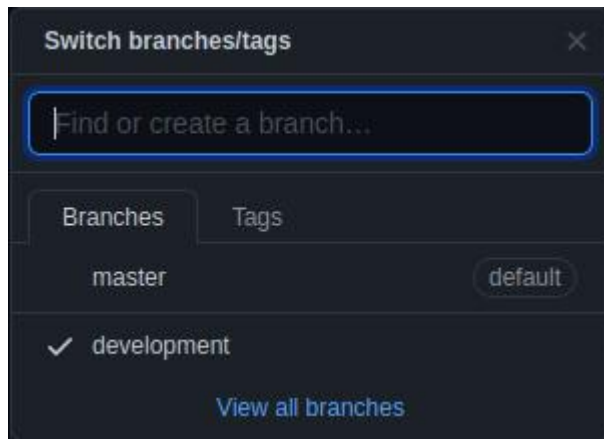


3. Tes koneksi

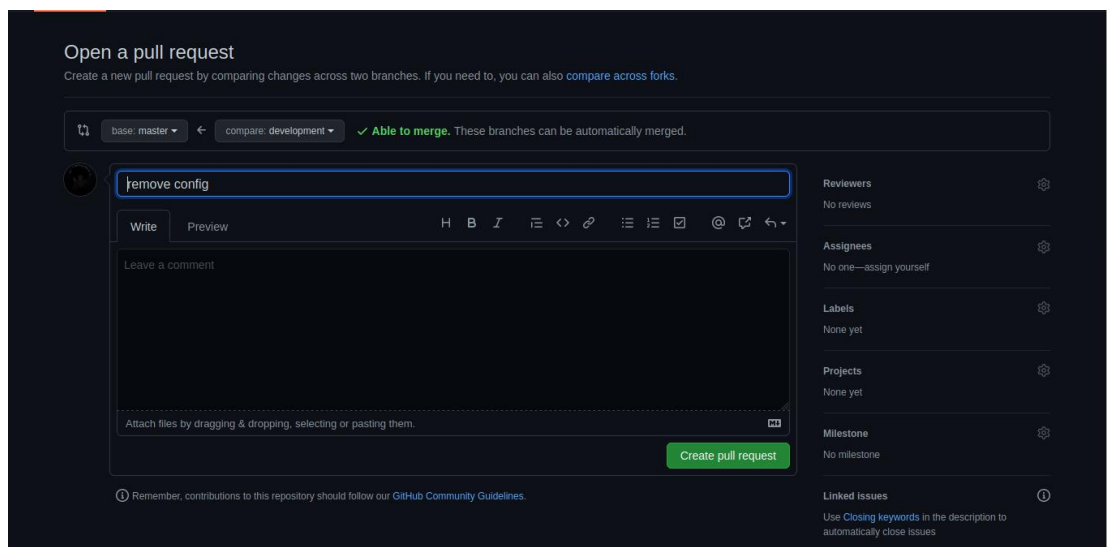


Development

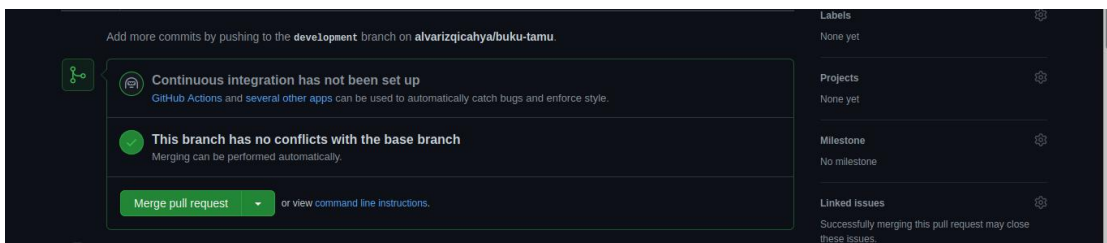
1. Diusahakan menggunakan git (github)
2. Gunakan branch yang berbeda untuk development dan production



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

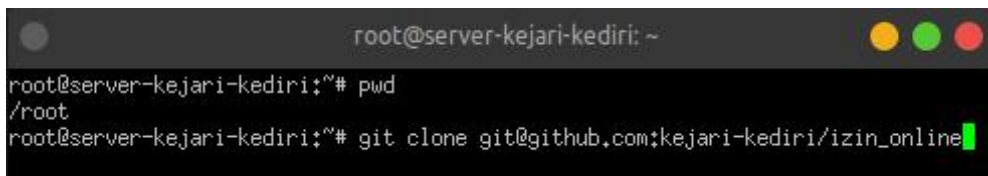
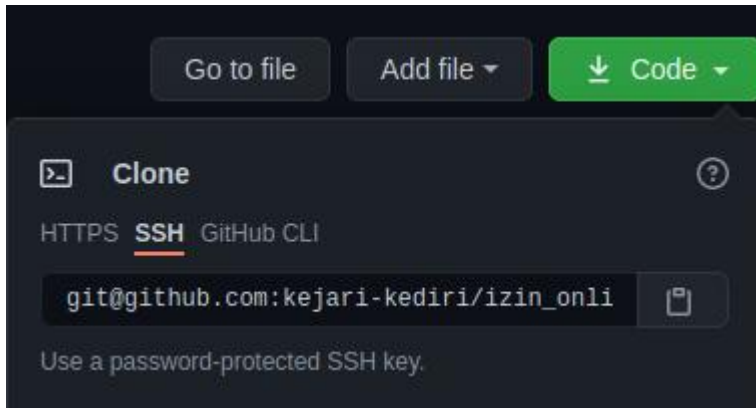


4. Setelah memastikan tidak ada error pada program lakukan Merge pull request



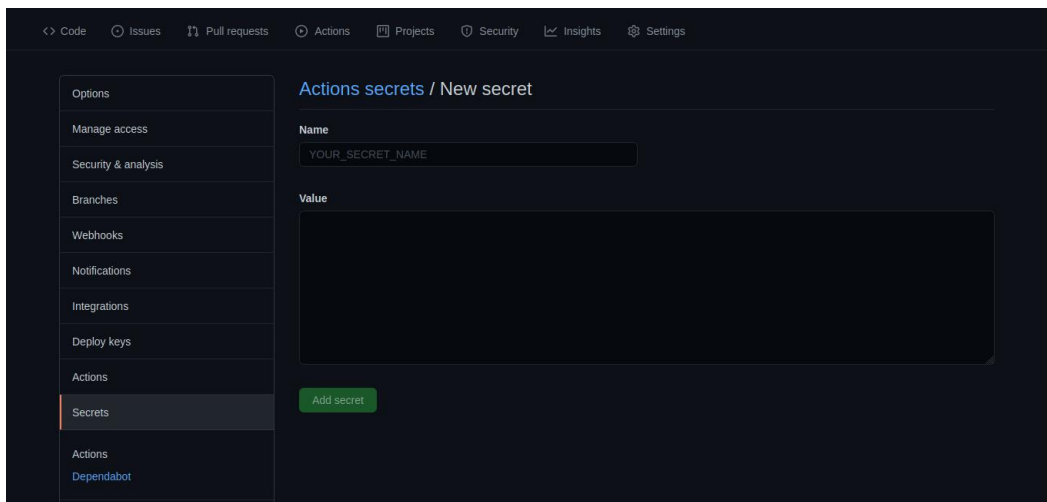
Deployment

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:

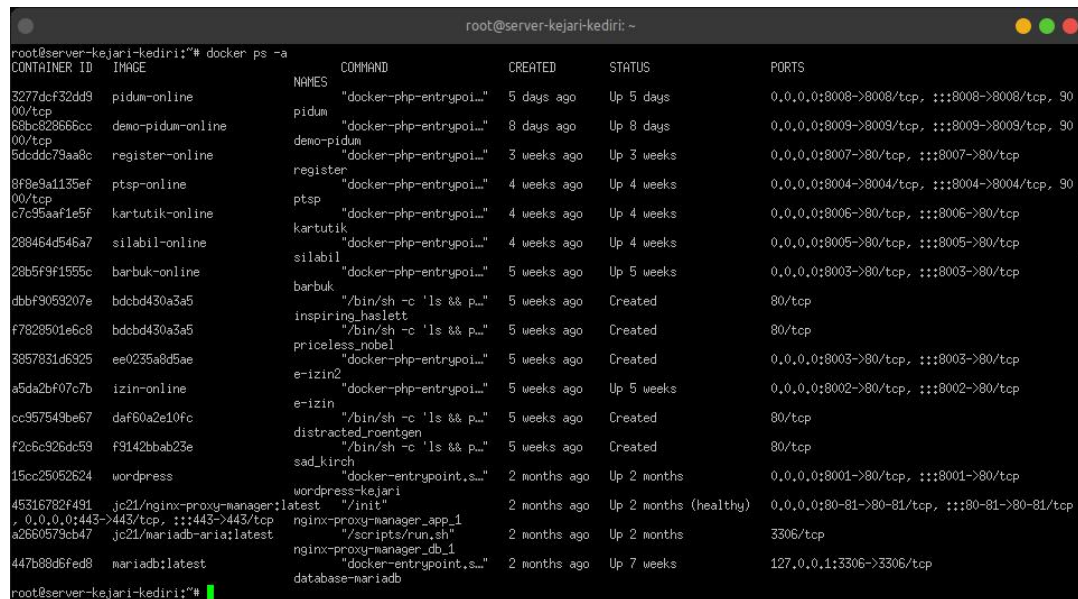
```
FROM php:7.3.5-apache-stretch
```

```

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

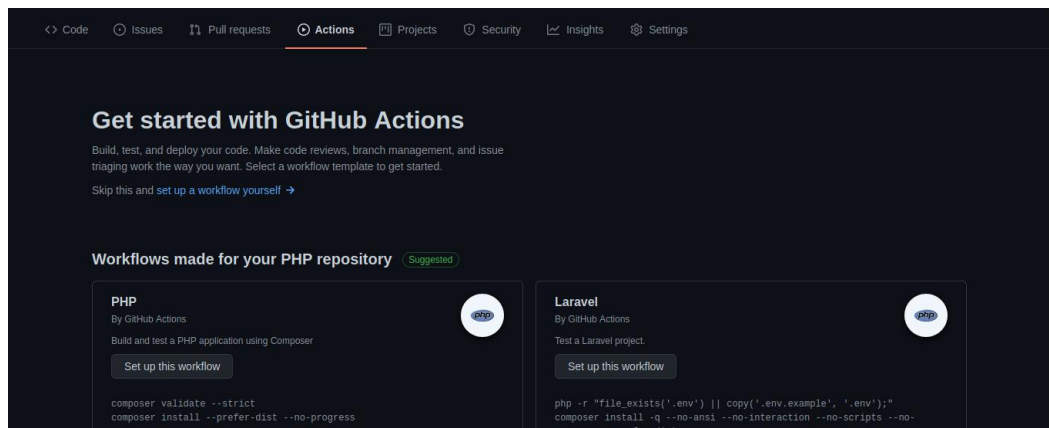
```

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

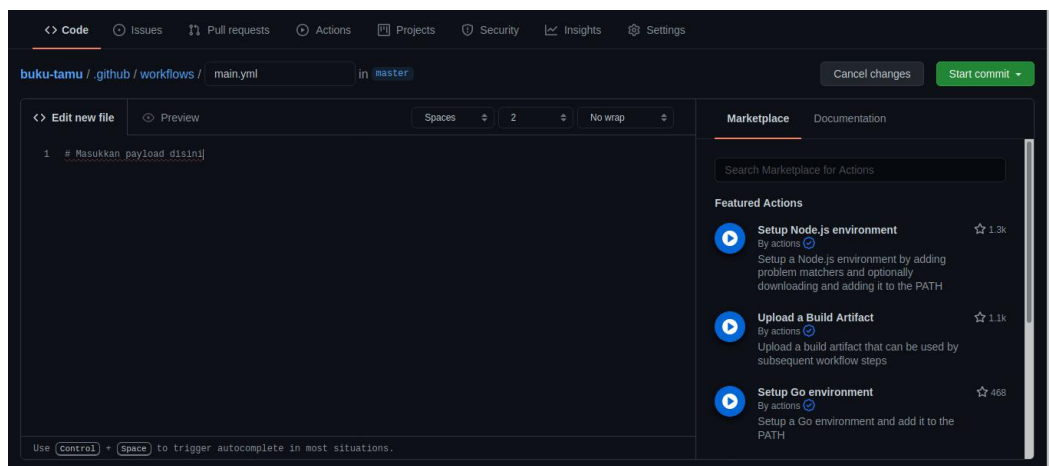


CONTAINER ID	IMAGE	NAMES	COMMAND	CREATED	STATUS	PORTS
3277df32dd9	pidum-online	pidum	"docker-php-entrypoi..."	5 days ago	Up 5 days	0.0.0.0:8008->8008/tcp, :::8008->8008/tcp, 90
00/tcp	demo-pidum-online	demo-pidum	"docker-php-entrypoi..."	8 days ago	Up 8 days	0.0.0.0:8009->8009/tcp, :::8009->8009/tcp, 90
68bc68866cc	register-online	register	"docker-php-entrypoi..."	3 weeks ago	Up 3 weeks	0.0.0.0:8007->80/tcp, :::8007->80/tcp
5dcd5c79aa8c	ptsp-online	ptsp	"docker-php-entrypoi..."	4 weeks ago	Up 4 weeks	0.0.0.0:8004->8004/tcp, :::8004->8004/tcp, 90
00/tcp	kartutik-online	kartutik	"docker-php-entrypoi..."	4 weeks ago	Up 4 weeks	0.0.0.0:8006->80/tcp, :::8006->80/tcp
c7c95aaf1e5f	silabil-online	silabil	"docker-php-entrypoi..."	4 weeks ago	Up 4 weeks	0.0.0.0:8005->80/tcp, :::8005->80/tcp
288464d546a7	barbuk-online	barbuk	"docker-php-entrypoi..."	5 weeks ago	Up 5 weeks	0.0.0.0:8003->80/tcp, :::8003->80/tcp
28b5f9f1555c	bdcdb430a3a5	inspiring_haslett	"/bin/sh -c 'ls && p..."	5 weeks ago	Created	80/tcp
dbbf9059207e	bdcdb430a3a5	priceless_nobel	"/bin/sh -c 'ls && p..."	5 weeks ago	Created	80/tcp
f7828501e6c8	ee0235a8d5ae	e-izin2	"docker-php-entrypoi..."	5 weeks ago	Created	0.0.0.0:8003->80/tcp, :::8003->80/tcp
3857831d6925	a5da2b07c7b	e-izin	"docker-php-entrypoi..."	5 weeks ago	Up 5 weeks	0.0.0.0:8002->80/tcp, :::8002->80/tcp
a5da2b07c7b	daf60a2e10fc	distracted_roentgen	"/bin/sh -c 'ls && p..."	5 weeks ago	Created	80/tcp
cc957549be67	f9142bbab23e	sad_kirch	"/bin/sh -c 'ls && p..."	5 weeks ago	Created	80/tcp
f2c6c926dc59	wordpress	wordpress-kejari	"docker-entrypoint.s..."	2 months ago	Up 2 months	0.0.0.0:8001->80/tcp, :::8001->80/tcp
15cc25052624	45316782f491	nginx-proxy-manager	"/init"	2 months ago	Up 2 months (healthy)	0.0.0.0:80-81->80-81/tcp, :::80-81->80-81/tcp
45316782f491	, 0.0.0.0:443->443/tcp, :::443->443/tcp	nginx-proxy-manager_app_1	"/scripts/run.sh"	2 months ago	Up 2 months	3306/tcp
, 0.0.0.0:443->443/tcp, :::443->443/tcp	a2660579cb47	nginx-proxy-manager_db_1	"docker-entrypoint.s..."	2 months ago	Up 7 weeks	127.0.0.1:3306->3306/tcp
a2660579cb47	447b88d8fed8	database-mariadb				

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: ${ secrets.PORT }
        script: |
          cd izin_online
          git pull origin master
          git status
          docker build -t izin-online .
          docker rm e-izin -f
          docker run -e DB_HOST=database-mariadb -e DB_PORT=3306 -e
DB_USER=root -e DB_PASS=segopecell2 -e DB_NAME=izin -e
BASE_URL=https://izin.kejari-kediri.go.id/ -e
API_KEY_WA=C1tJ52rlxPGQq57gRmHSikvK5UbfVRgZrR9pHtVC5u3OK6a5ud9KnL
zyDtNVmGN5 --name e-izin -p 8002:80 -v "$PWD":/var/www/html --network
database-network -d izin-online

```

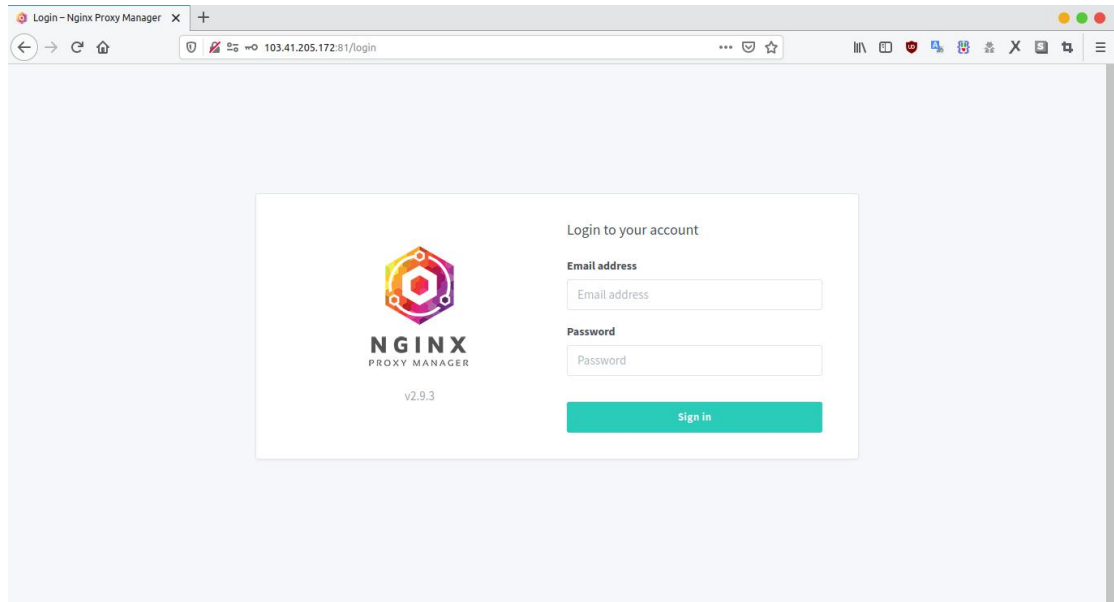
Ketentuan: branches sesuaikan dengan branch utama, script sesuaikan dengan project masing - masing.

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

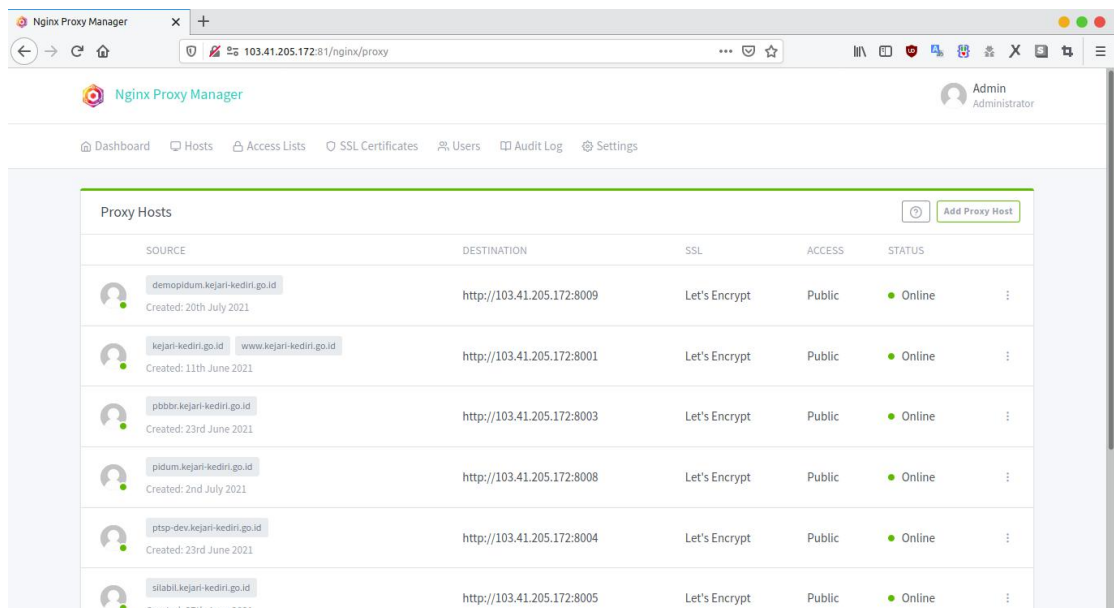
NB: Tahap tersebut hanya untuk project yang belum pernah terbangun integrasi antara github ke server. Jika sudah terbangun integrasi hanya perlu Push atau Marge pull request pada branch utama.

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 Request a new SSL dan switch force SSL

The screenshot shows the 'New Proxy Host' dialog box with the 'Details' tab selected. The 'Domain Names' field is empty. The 'Scheme' is set to 'http'. The 'Forward Hostname / IP' field is empty. The 'Forward Port' is set to '80'. There are three toggle switches: 'Cache Assets' (off), 'Block Common Exploits' (off), and 'Websockets Support' (off). The 'Access List' dropdown is set to 'Publicly Accessible'. At the bottom right are 'Cancel' and 'Save' buttons.

The screenshot shows the 'New Proxy Host' dialog box with the 'SSL' tab selected. The 'SSL Certificate' dropdown is set to 'None'. There are four toggle switches: 'Force SSL' (off), 'HTTP/2 Support' (off), 'HSTS Enabled' (off), and 'HSTS Subdomains' (off). At the bottom right are 'Cancel' and 'Save' buttons.

4. Cek dengan mengakses domain yang sudah dibuat