


Automation Deploy - Github Action Laravel

 Source : [Automate your Laravel app deployment with Github Actions](#)

1. Buat folder bernama `.scripts` didalam folder project kita kemudian di dalam nya kita buat file bernama `deploy.sh` dan isikan script dibawah

```
1  #!/bin/bash
2  set -e
3
4  echo "Deployment started ..."
5
6  # Enter maintenance mode or return true
7  # if already is in maintenance mode
8  # (php artisan down) || true
9
10 # Pull the latest version of the app
11 git pull origin master
12
13 # Install composer dependencies
14 # composer install --no-dev --no-interaction --prefer-dist --optimize-autoloader
15
16 # Clear the old cache
17 # php artisan clear-compiled
18
19 # Recreate cache
20 # php artisan optimize
21
22 # Compile npm assets
23 # npm run prod
24
25 # Run database migrations
26 # php artisan migrate --force
27
28 # Exit maintenance mode
29 # php artisan up
30
31 echo "Deployment finished!"
```

2. Kemudian buat folder `.github/workflows` tambahkan `deploy.yaml` dan isikan script dibawah

```
1  name: Deploy
2
3  # Trigger the workflow on push and
4  # pull request events on the production branch
5  on:
6    push:
7      branches:
8        - master
9    pull_request:
```

```

10   branches:
11     - master
12
13   # Authenticate to the the server via ssh
14   # and run our deployment script
15   jobs:
16     deploy:
17       runs-on: ubuntu-20.04
18       steps:
19         - uses: actions/checkout@v2
20         - name: Deploy to server
21           uses: appleboy/ssh-action@master
22           with:
23             host: ${ secrets.HOST }
24             username: ${ secrets.USERNAME }
25             port: ${ secrets.PORT }
26             key: ${ secrets.SSHKEY }
27             script: "cd /var/www/html/lightclass-dev && ./scripts/deploy.sh"

```

3. Kemudian setup ssh key di server menggunakan perintah dibawah (execute di server)

```

1  ssh-keygen -t rsa -b 4096 -C "temmy.alex@gmail.com"

```

4. Kemudian masukkan ssh key kedalam ssh-agent menggunakan perintah dibawah (execute di server)

```

1  $ eval "$(ssh-agent -s)"
2  $ ssh-add ~/.ssh/id_rsa

```

5. Tambahkan public key ssh ke dalam folder `authorized_keys` menggunakan script dibawah (execute di server)

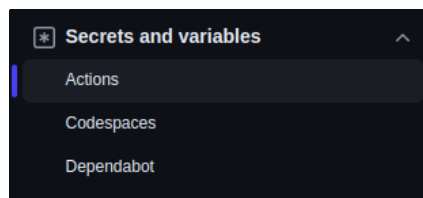
```

1  $ cat ~/.ssh/id_rsa.pub >> ~/.ssh/authorized_keys

```

Copy SSH Key ke Github

6. Click `Settings > Secrets and variables > Actions`



7. Kemudian buat secret dengan click Button `New repository secret` dan Buatlah key dan isikan value nya (disesuaikan dengan kebutuhan) seperti pada gambar dibawah



8. Untuk value dengan key `SSHKEY` value nya berasal dari perintah dibawah yang di execute di server

```
1 $ cat ~/.ssh/id_rsa
```

9. Untuk value dengan key `USERNAME` berasal dari username dari ssh atau bisa juga di check menggunakan perintah `whoami`

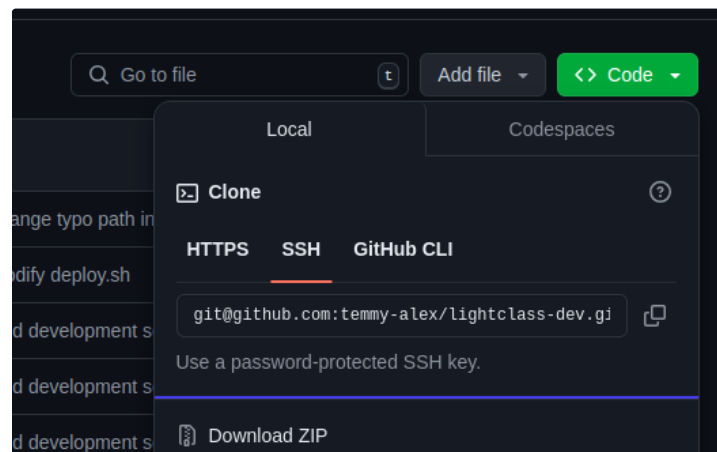
10. Kemudian masuk ke Settings > Deploy keys dan tambahkan key dalam hal ini contohnya `PROD_SERVER` dan isikan value yang berasal dari perintah

```
1 $ cat ~/.ssh/id_rsa.pub
```

11. Kemudian masuk ke dalam folder project yang berada di server dan tambahkan perintah

```
1 git remote set-url origin git@github.com:USERNAME/REPOSITORY.git
2 git fetch
```

Untuk mengambil format diatas dapat dilihat pada gambar dibawah ini (pilih tab ssh)



12. Kemudian jalankan script dibawah agar script deploy nya memiliki permission

```
1 sudo chmod -R 777 ./var/www/html/lightclass-dev/.scripts
```

13. Dan check melalui menu Actions pada github