TEAM 1

I

DOCKER

08 Januari 2024

2

Pengertian Docker

Docker adalah layanan manajemen kontainer. Kata kunci Docker adalah mengembangkan, mengirimkan, dan menjalankan di mana saja.

Ide keseluruhan Docker adalah agar develop dapat mengembangkan aplikasi dengan mudah, mengirimkannya ke dalam container yang kemudian dapat dideploy di mana saja

3

Komponen Utama Docker

Docker Registry

Docker Registry merupakan tempat penyimpanan (publik atau privat) dimana kita bisa mengupload dan mendownload images. Registry public docker disebut dengan Docker Hub yang didalamnya terdapat banyak image, baik image yang dibuat sendiri ataupun image yang lain.

4

Komponen Utama Docker

Docker Images

Docker images merupakan sebuah template yang bersifat read-only. Image ini digunakan untuk menjalankan container.

Docker menyediakan cara yang sederhana untuk membangun image baru atau merubah image yang sudah ada. Dibangun oleh Docker user. Disimpan di Docker Hub atau registri lokal.

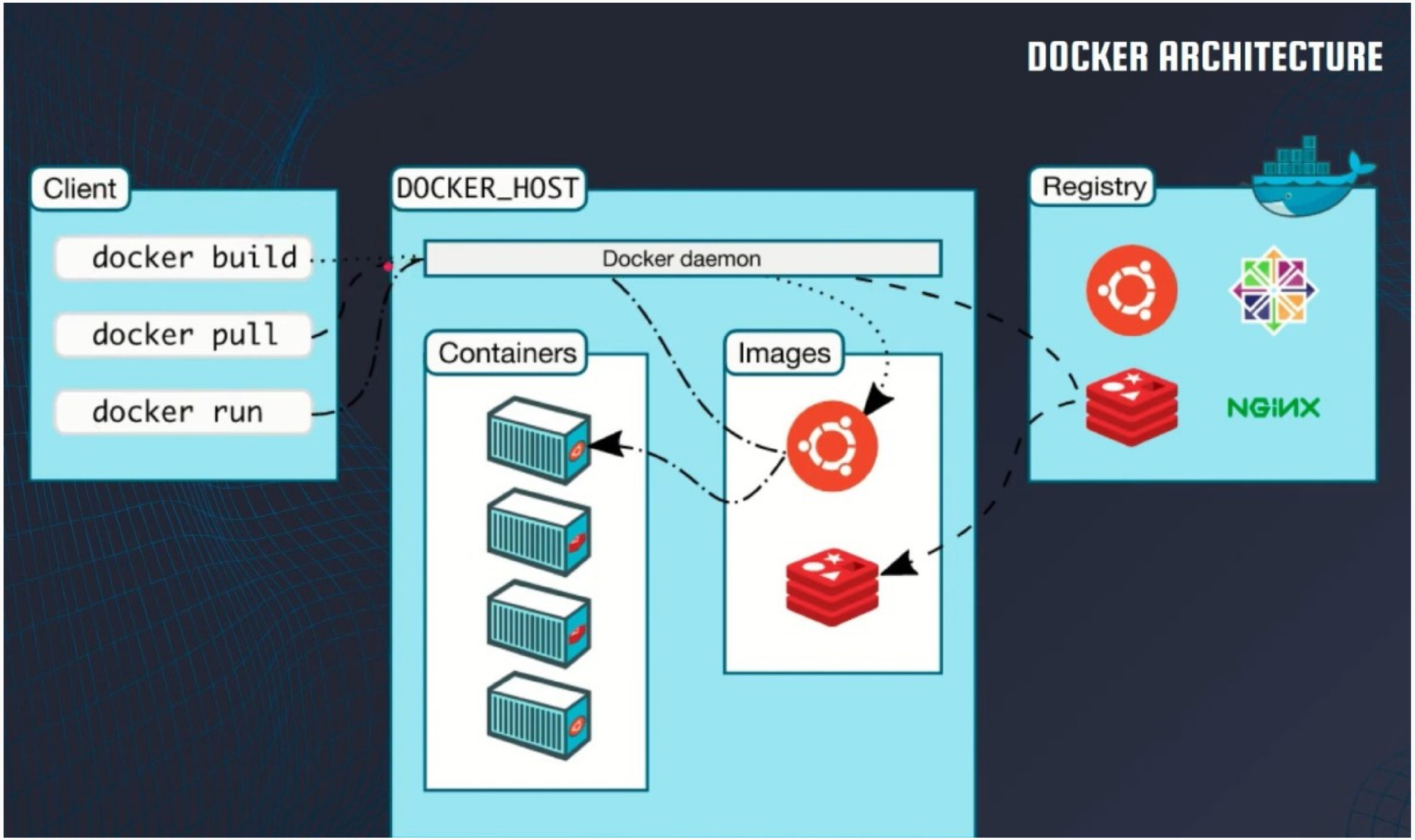
5

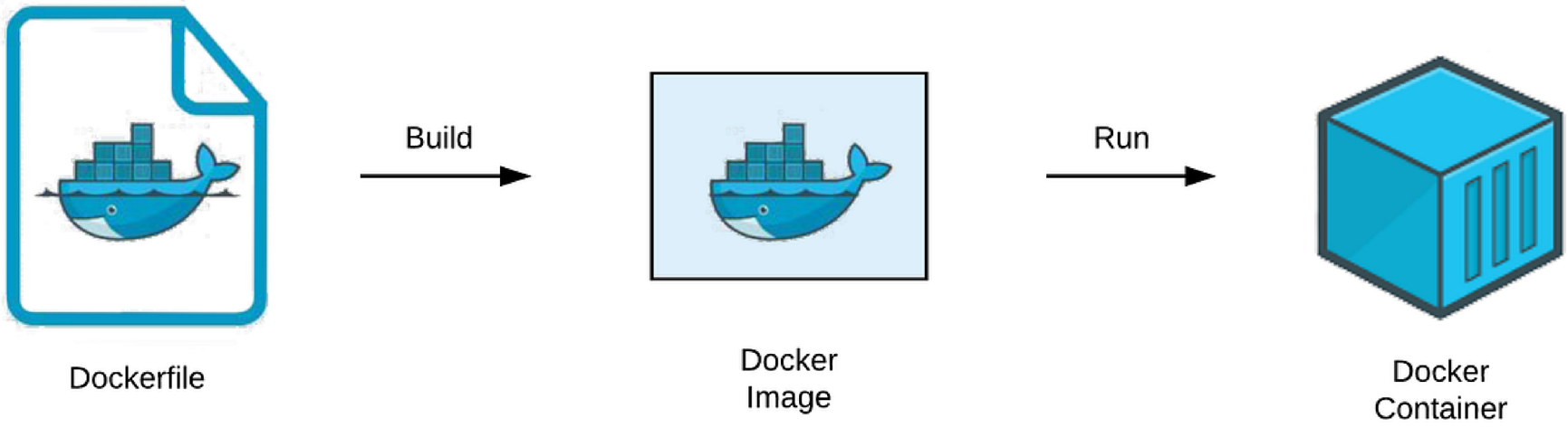
Komponen Utama Docker

Docker Container

Docker container merupakan sebuah image yang bersifat read-write yang berjalan di atas image. Docker menggunakan union-file sistem sebagai back-end file sistem containernya, dimana setiap perubahan yang disimpan pada container akan menyebabkan terbentuknya layer baru di atas base image. Jadi container merupakan layer dimana kita bisa melakukan instalasi aplikasi di dalamnya.

6


DOCKER ARCHITECTURE
Client
docker build
docker pull
docker run
DOCKER_HOST
Docker daemon
Containers
Images
Registry
NGINX


Build
Run
Dockerfile
Docker
Image
Docker
Container

8

Dokumentasi

team1@appserver: ~

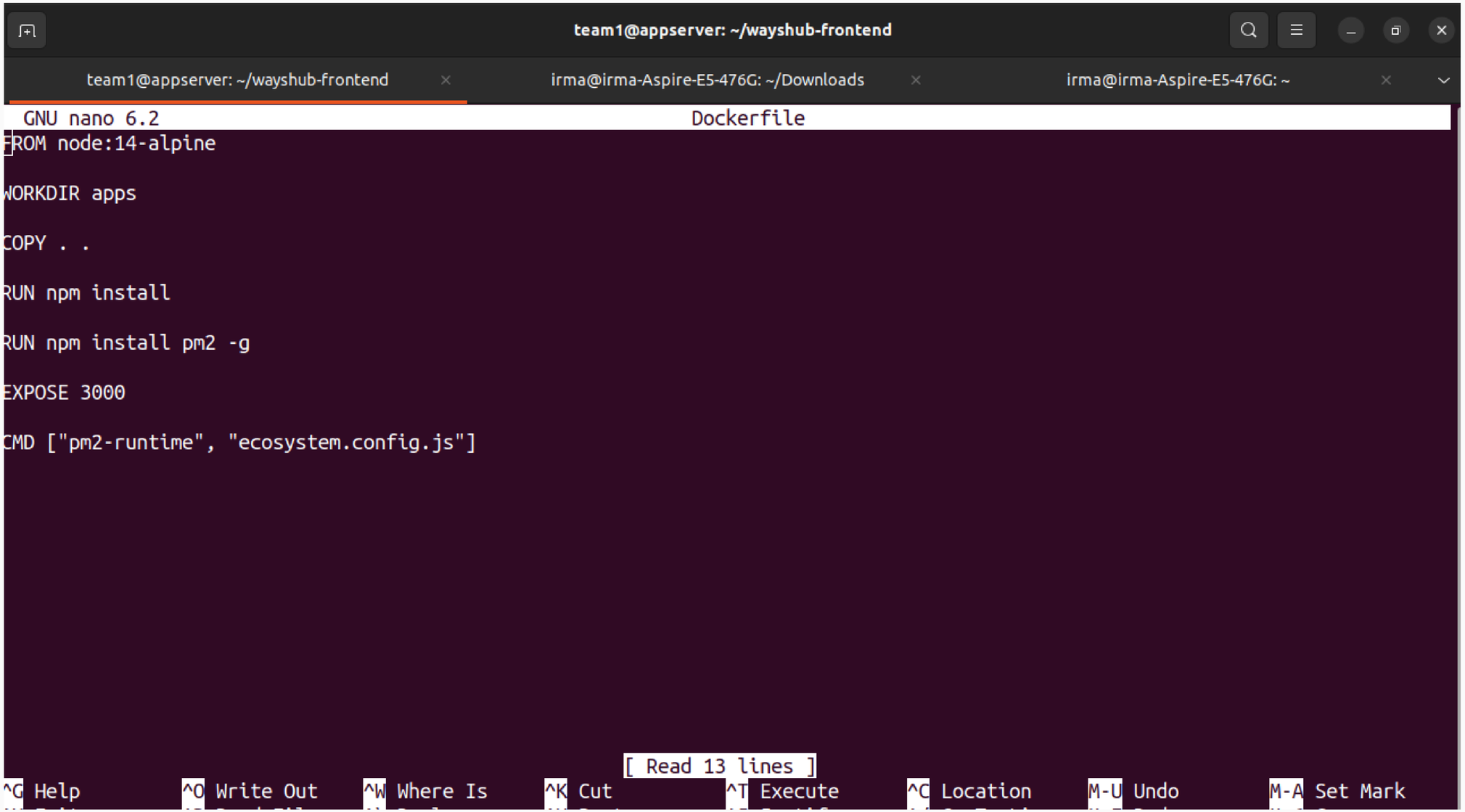
team1@appserver: ~ | irma@irma-Aspire-E5-476G: ~/Downloads | irma@irma-Aspire-E5-476G: ~

GNU nano 6.2 docker-compose.yaml *

```
version: "3.8"
services:

 database:
   container_name: mysql
   image: mysql:latest
   restart: always
   ports:
     - "3306:3306"
   environment:
     - MYSQL_ROOT_PASSWORD=tim1
     - MYSQL_USER=tim1
     - MYSQL_PASSWORD=tim1
     - MYSQL_DATABASE=wayshub
   volumes:
     - ~/myql:/var/lib/mysql
   networks:
     local:

 backend:
   container_name: be
   image: tim1/backend:1.0
   depends_on:
     - database
   ports:
     - "5000:5000"
   networks:
     local:

 frontend:
   container_name: fe
   image: tim1/frontend:1.0
   depends_on:
     - database
     - backend
   ports:
     - "3000:3000"
   networks:
     local:

 webserver:
   container_name: nginx
   restart: always
   image: nginx:latest
   depends_on:
     - database
```

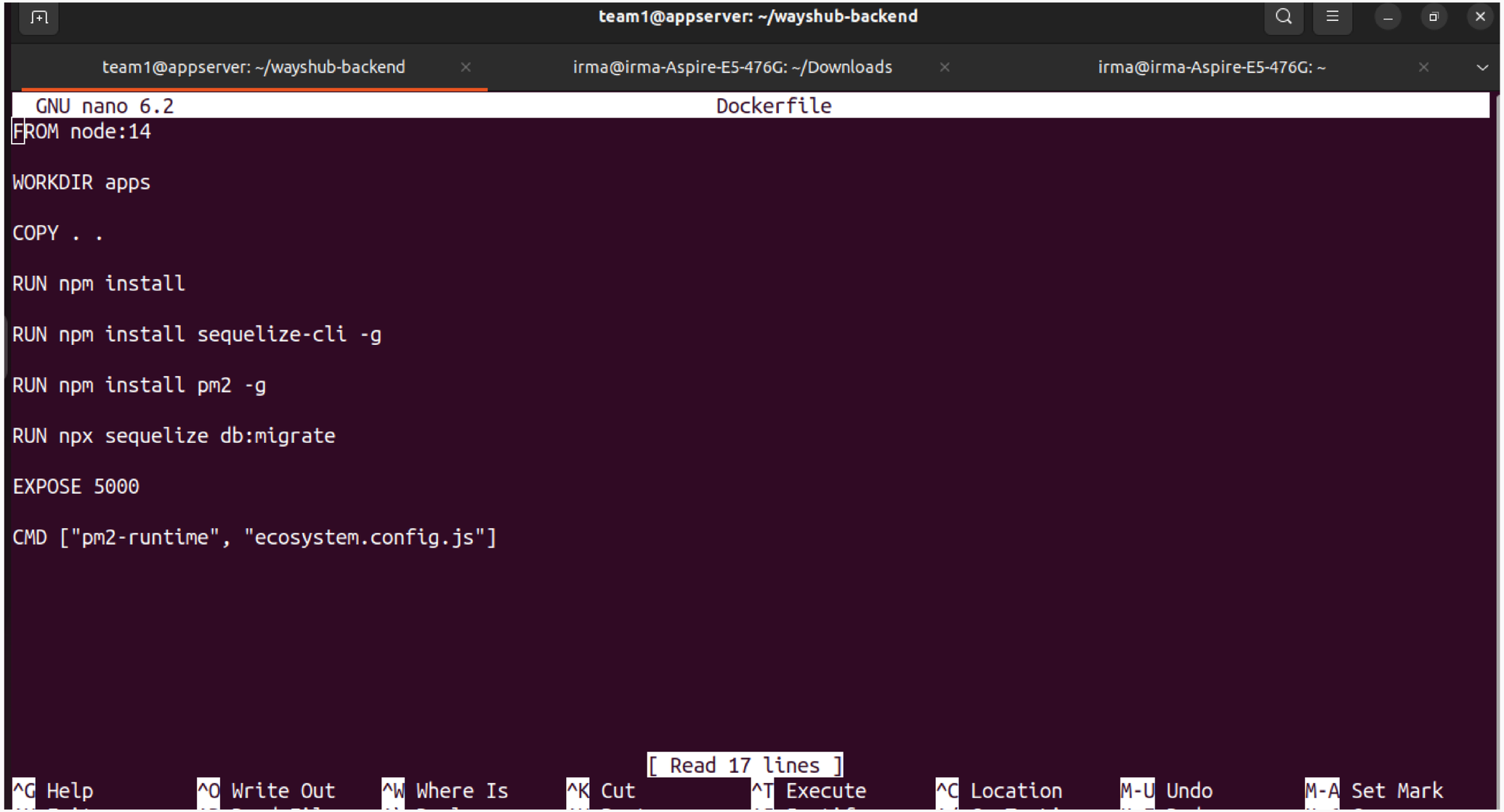
9

Dokumentasi Docker File



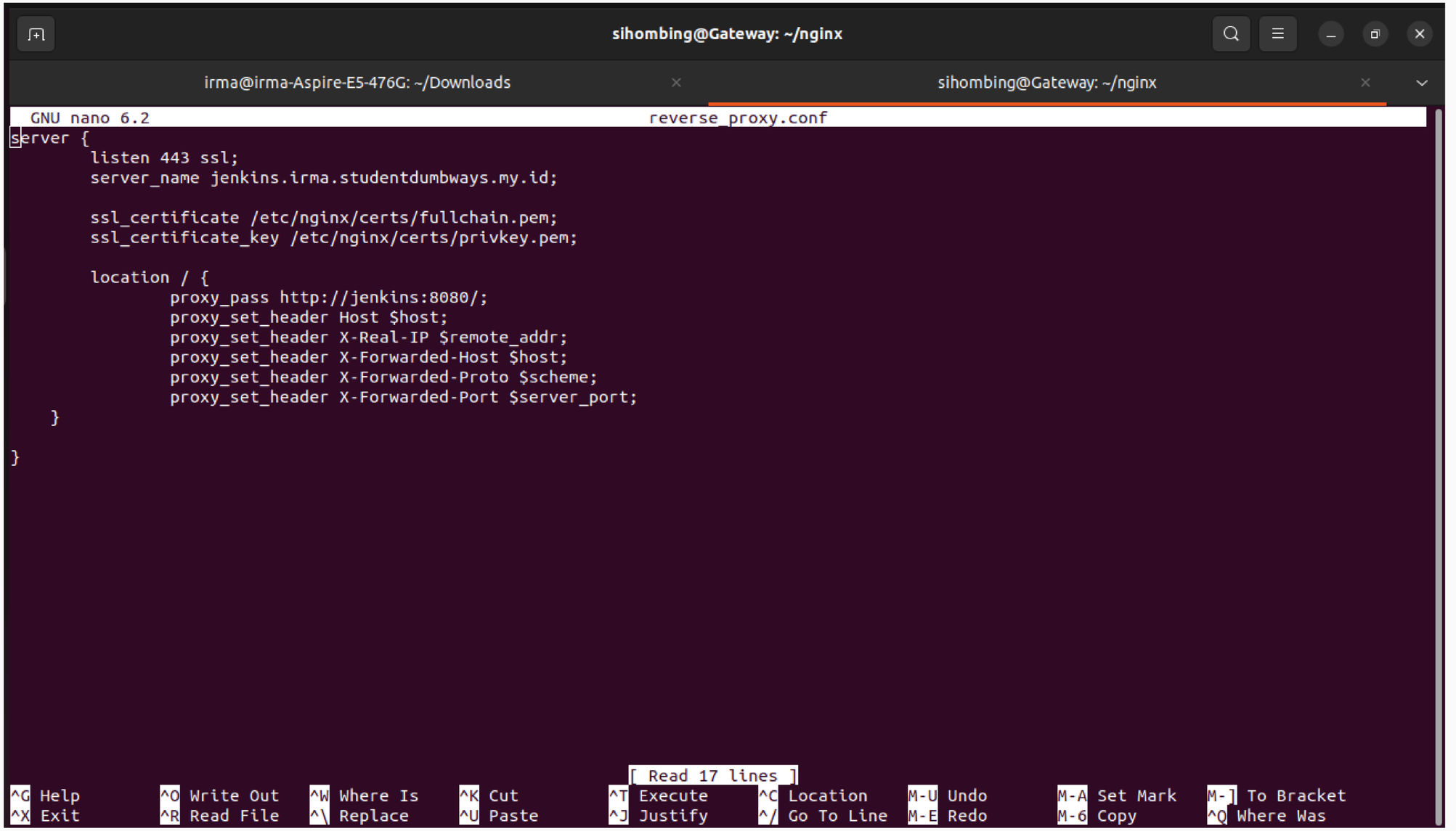
10

Dokumentasi Docker File



11

Dokumentasi reverse_proxy



12

Dokumentasi docker images

```
team1@appserver:~$ docker images
REPOSITORY            TAG       IMAGE ID       CREATED          SIZE
nurliza/backend       latest    13b20bdbf80e   2 days ago       1GB
nurliza/frontend      latest    b4d0cf5c6d68   3 days ago       340MB
tim1/frontend         1.0       fefe0c2c898a   5 days ago       340MB
tim1/backend          1.0       8cf61deb4723   5 days ago       1GB
mysql                 latest    73246731c4b0   2 weeks ago      619MB
nginx                 latest    d453dd892d93   2 months ago     187MB
team1@appserver:~$
```

13

Dokumentasi built aplikasi frontend

```
kerfile  README.md  debug.log  ecosystem.config.js  package-lock.json  package.json  public  src  yarn.lock
n1@appserver:~/wayshub-frontend$ docker build -t tim1/frontend:1.0 .
Building 231.7s (10/10) FINISHED                                                                                docker:defaul
[internal] load .dockerignore                                                                                          0.3
=> transferring context: 2B                                                                                            0.0
[internal] load build definition from Dockerfile                                                                       0.2
=> transferring dockerfile: 181B                                                                                       0.0
[internal] load metadata for docker.io/library/node:14-alpine                                                          3.2
[1/5] FROM docker.io/library/node:14-alpine@sha256:434215b487a329c9e867202ff89e704d3a75e554822e07f3e0c0f9e606121b33   11.2
=> resolve docker.io/library/node:14-alpine@sha256:434215b487a329c9e867202ff89e704d3a75e554822e07f3e0c0f9e606121b33    0.1
=> sha256:0dac3dc27b1ad570e6c3a7f7cd29e88e7130ff0cad31b2ec5a0f222fbe971bdb 6.44kB / 6.44kB                             0.0
=> sha256:434215b487a329c9e867202ff89e704d3a75e554822e07f3e0c0f9e606121b33 1.43kB / 1.43kB                             0.0
=> sha256:4e84c956cd276af9ed14a8b2939a734364c2b0042485e90e1b97175e73dfd548 1.16kB / 1.16kB                             0.0
=> sha256:f56be85fc22e46face30e2c3de3f7fe7c15f8fd7c4e5add29d7f64b87abdaa09 3.37MB / 3.37MB                             0.5
=> sha256:8f665685b215c7daf9164545f1bbdd74d800af77d0d267db31fe0345c0c8fb8b 37.17MB / 37.17MB                           1.8
=> sha256:e5fca6c395a62ec277102af9e5283f6edb43b3e4f20f798e3ce7e425be226ba6 2.37MB / 2.37MB                             1.1
=> extracting sha256:f56be85fc22e46face30e2c3de3f7fe7c15f8fd7c4e5add29d7f64b87abdaa09                                   0.6
=> sha256:561cb69653d56a9725be56e02128e4e96fb434a8b4b4decf2bdeb479a225feaf 448B / 448B                                 1.1
=> extracting sha256:8f665685b215c7daf9164545f1bbdd74d800af77d0d267db31fe0345c0c8fb8b                                   7.6
=> extracting sha256:e5fca6c395a62ec277102af9e5283f6edb43b3e4f20f798e3ce7e425be226ba6                                   0.3
=> extracting sha256:561cb69653d56a9725be56e02128e4e96fb434a8b4b4decf2bdeb479a225feaf                                   0.0
[internal] load build context                                                                                          0.7
=> transferring context: 5.51MB                                                                                        0.5
[2/5] WORKDIR apps                                                                                                     0.7
[3/5] COPY . .                                                                                                         0.4
[4/5] RUN npm install                                                                                                131.4
```

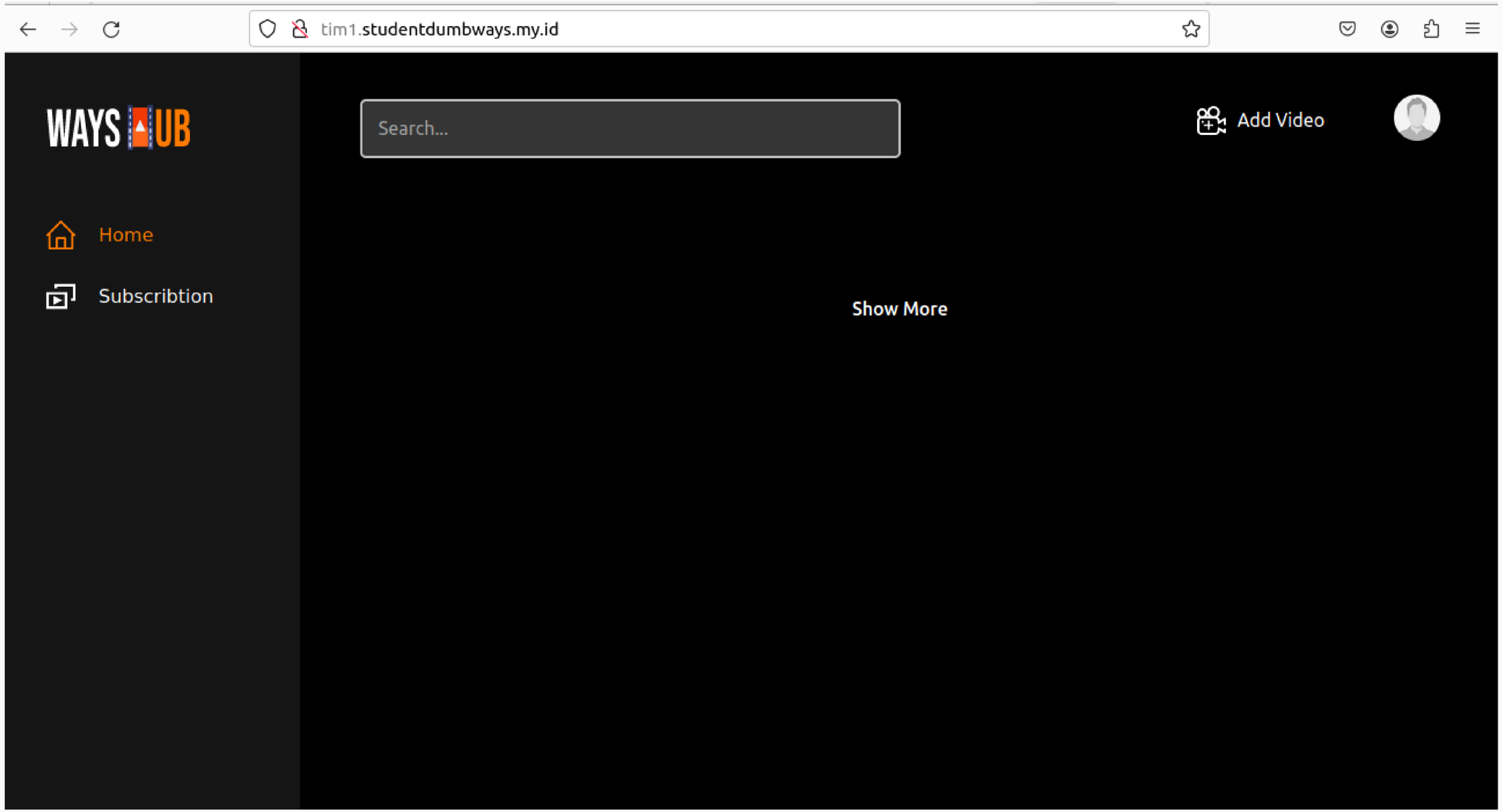
TEAM 1

14 Dokumentasi built aplikasi backend

```
3306/tcp, 33060/tcp
team1@appserver:~/wayshub-backend$ docker build -t tim1/backend:1.0 .
[+] Building 47.8s (12/12) FINISHED                                                          docker:default
 => [internal] load build definition from Dockerfile                                                   0.1s
 => => transferring dockerfile: 238B                                                                   0.0s
 => [internal] load .dockerignore                                                                      0.1s
 => => transferring context: 2B                                                                        0.0s
 => [internal] load metadata for docker.io/library/node:14                                             1.0s
 => [1/7] FROM docker.io/library/node:14@sha256:a158d3b9b4e3fa813fa6c8c590b8f0a860e015ad4e59bbce5744d2f6fd8461aa 0.0s
 => [internal] load build context                                                                      0.2s
 => => transferring context: 3.63kB                                                                    0.1s
 => CACHED [2/7] WORKDIR apps                                                                          0.0s
 => CACHED [3/7] COPY . .                                                                              0.0s
 => CACHED [4/7] RUN npm install                                                                       0.0s
 => CACHED [5/7] RUN npm install sequelize-cli -g                                                      0.0s
 => CACHED [6/7] RUN npm install pm2 -g                                                                0.0s
 => [7/7] RUN npx sequelize db:migrate                                                                 7.1s
 => exporting to image                                                                                38.8s
 => => exporting layers                                                                               38.7s
 => => writing image sha256:f2afff90665271882dddcfa79d9ab8b9575f2a1b5df2cc9e1f11387eea367f8c          0.0s
 => => naming to docker.io/tim1/backend:1.0                                                            0.0s
team1@appserver:~/wayshub-backend$ cd
team1@appserver:~$ cd wayshub-frontend/
team1@appserver:~/wayshub-frontend$ ls
Dockerfile  README.md  debug.log  ecosystem.config.js  package-lock.json  package.json  public  src  yarn.lock
team1@appserver:~/wayshub-frontend$ docker build -t tim1/frontend:1.0 .
[+] Building 177.3s (9/9) FINISHED                                                           docker:default
 => [internal] load build definition from Dockerfile                                                   0.5s
```

15

Dokumentasi aplikasi berjalan



TEAM 1

I

Jenkins

08 Januari 2024

2

Pengertian Jenkins

1. Jenkins adalah alat otomatisasi open source yang digunakan untuk membangun dan menguji proyek perangkat lunak
2. Alat ini memudahkan developer untuk mengintegrasikan perubahan pada suatu proyek
3. Jenkins mencapai integrasi berkelanjutan dengan bantuan plungin
4. Jenkins cocok untuk membangun pipeline CI/CD karena fleksibilitasnya, keterbukaannya, kemampuan pluginnya, dan sifatnya yang mudah digunakan.

3

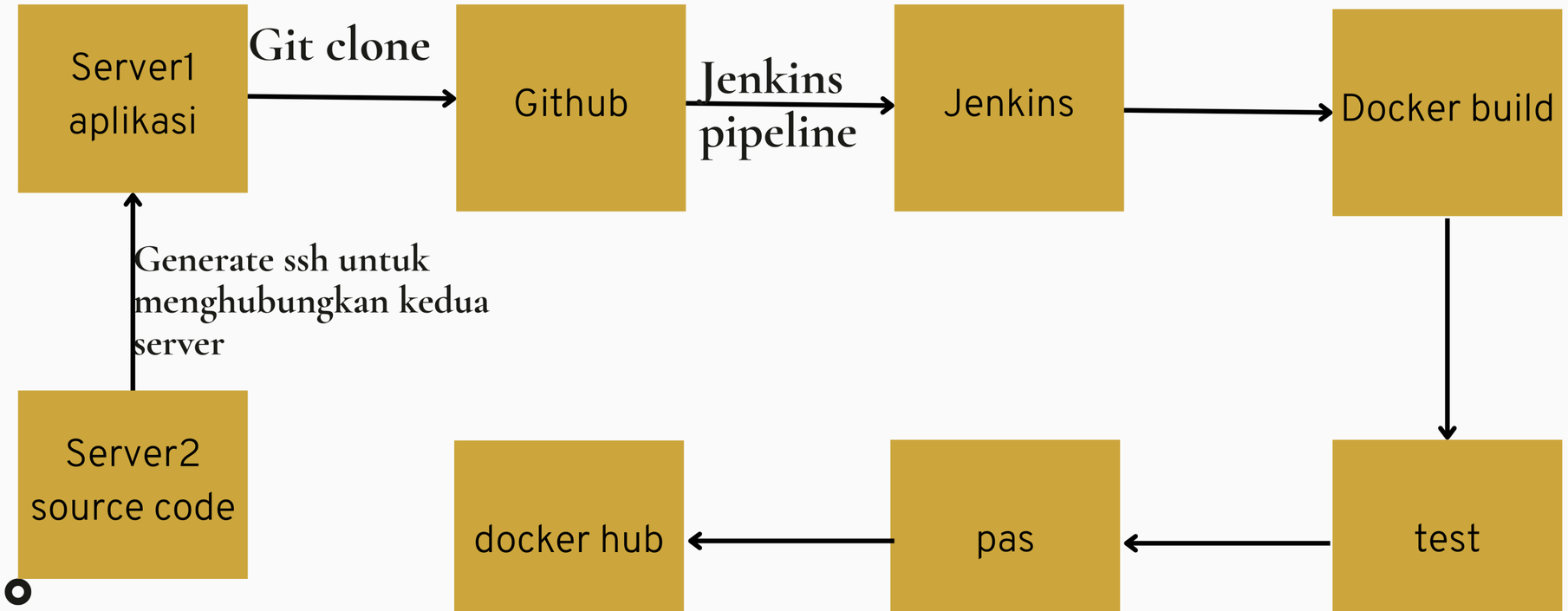
Building CI/CD with Jenkins

Membangun aplikasi dengan menggunakan jenins

Dalam pengerjaan projek ini menggunakan dua server, server pertama digunakan untuk menjalankan aplikasi dan server ke dua digunakan untuk menjalankan jenkins

4

Building CI/CD with Jenkins



5 Dokumentasi

irma@irma-Aspire-E5-476G: ~/Downloads × irma@irma-Aspire-E5-476G: ~/Downloads × sihombing@Gateway: ~ ×

GNU nano 6.2 docker-compose.yaml

```
version: "3.8"
services:

  jenkins:
    image: jenkins/jenkins:lts
    container_name: jenkins
    ports:
      - "8080:8080" # Port Jenkins
    volumes:
      - ./jenkins_home:/var/jenkins_home
    networks:
      local:


  webserver:
    container_name: nginx
    restart: always
    image: nginx:latest
    ports:
      - "80:80"
      - "443:443"
    volumes:
      - "./nginx:/etc/nginx/conf.d"
      - "./nginx/certs:/etc/nginx/certs"
    networks:
      local:

networks:
  local:
```

6

Dokumentasi reverse_proxy jenkins

irma@irma-Aspire-E5-476G: ~/Downloads × irma@irma-Aspire-E5-476G: ~/Downloads × sihombing@Gateway: ~/nginx ×

```
  GNU nano 6.2                                  reverse_proxy.conf
server {
        listen 443 ssl;
        server_name jenkins.irma.studentdumbways.my.id;

        ssl_certificate /etc/nginx/certs/fullchain.pem;
        ssl_certificate_key /etc/nginx/certs/privkey.pem;

        location / {
                proxy_pass http://jenkins:8080/;
                proxy_set_header Host $host;
                proxy_set_header X-Real-IP $remote_addr;
                proxy_set_header X-Forwarded-Host $host;
                proxy_set_header X-Forwarded-Proto $scheme;
                proxy_set_header X-Forwarded-Port $server_port;
    }

}
```

7 Dokumentasi pipeline frontend

eline script

Script ?

```
pipeline {
    agent any
    environment{
        credential = 'cicd'
        server = 'team1@103.150.93.34'
        directory = '/home/team1/wayshub-frontend'
        branch = 'main'
        service = 'frontend'
        image = 'nurliza/frontend'
    }
    stages {
        stage('Pull code dari repository'){
            steps {
                sshagent([credential]) {
                    sh '''ssh -o StrictHostKeyChecking=no ${server} << EOF
                    docker compose down
                    cd ${directory}
                    git pull origin ${branch}
                    exit
```

8 Dokumentasi pipeline frontend

Pipeline script

Script ?

```
pipeline {
    agent any
    environment{
        credential = 'cicd'
        server = 'team1@103.150.93.34'
        directory = '/home/team1/wayshub-backend'
        branch = 'main'
        service = 'backend'
        image = 'nurliza/backend'
    }
    stages {
        stage('Pull code dari repository'){
            steps {
                sshagent([credential]) {
                    sh '''ssh -o StrictHostKeyChecking=no ${server} << EOF
                    docker compose down
                    cd ${directory}
                    git pull origin ${branch}
                    exit
```

9

Dokumentasi building, testing, dan deploy frontend

	Pull code dari repository	Building application	Testing application	aplikasi on top docker	Push image to docker hub	send notification to discord
Average stage times: (Average full run time: ~2min 10s)	13s	47s	10s	7s	46s	815ms
#5 Jan 04 23:17 No Changes	14s	4s	15s	9s	15s	713ms
#4 Jan 04 23:13 No Changes	19s	9s	14s	7s	7s	1s
#3 Jan 04 19:59 No Changes	8s	9s	14s	21s	3min 29s	1s

10

Dokumentasi building, testing, dan deploy backend

	Pull code dari repository	Building application	Testing application	Deploy aplikasi on top docker	Push image to docker hub	send notification to discord
Average stage times: (Average full run time: ~2min 49s)	11s	1min 26s	8s	3s	47s	746ms
#2 Jan 04 23:31 No Changes	7s	41s	16s	7s	1min 34s	1s
#1 Jan 04 23:27 No Changes	15s	2min 10s failed	121ms failed	100ms failed	117ms failed	118ms failed

TEAM 1

I

GitLab

08 Januari 2024

2

Pengertian GitLab

GitLab adalah repositori Git berbasis web yang menyediakan repositori terbuka dan pribadi secara gratis. GitLab adalah platform DevOps lengkap yang memungkinkan para profesional melakukan semua tugas dalam sebuah proyek —mulai dari perencanaan proyek dan manajemen kode sumber hingga pemantauan dan keamanan. Selain itu, GitLab memungkinkan tim untuk berkolaborasi dan membangun perangkat lunak yang lebih baik.

3

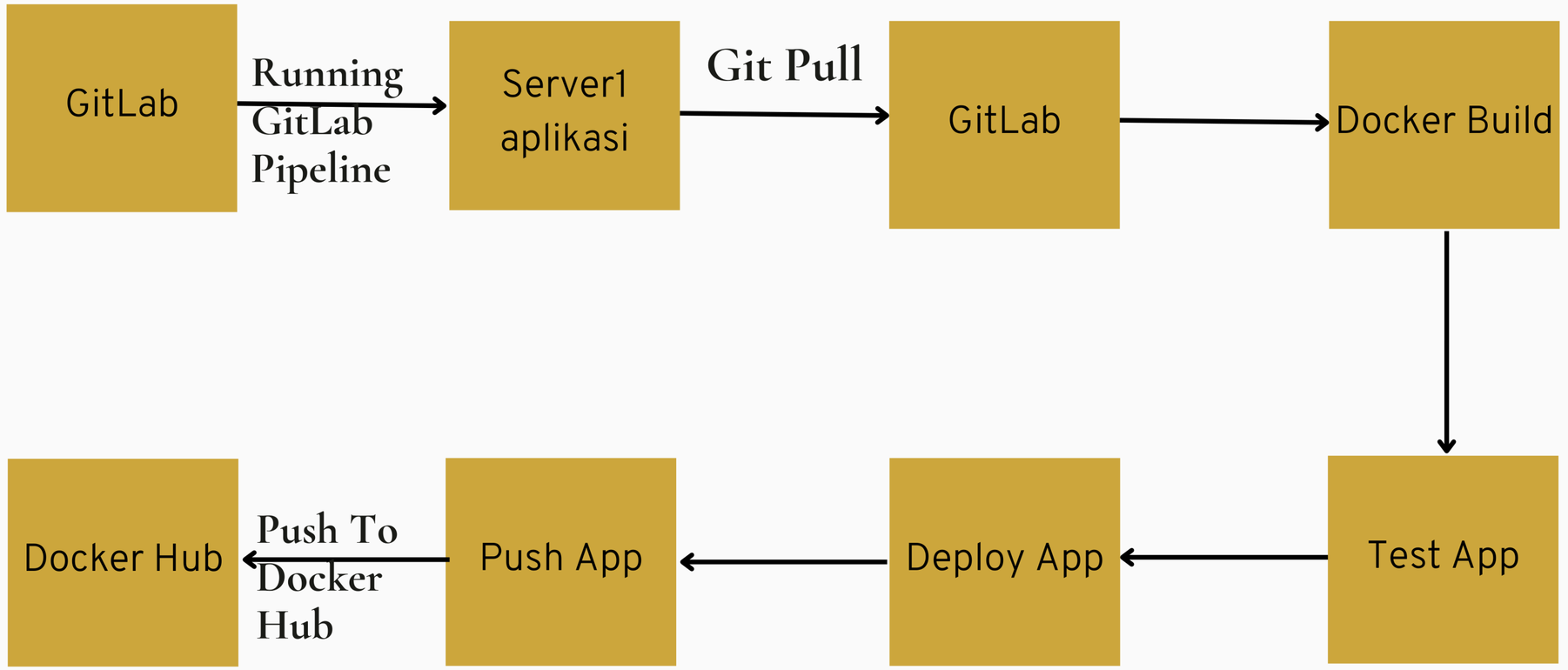
Building CI/CD GitLab Pipeline

Membangun aplikasi menggunakan GitLab

Pada bagian ini aplikasi akan berjalan menggunakan GitLab pipeline yang akan berjalan pada docker di server.

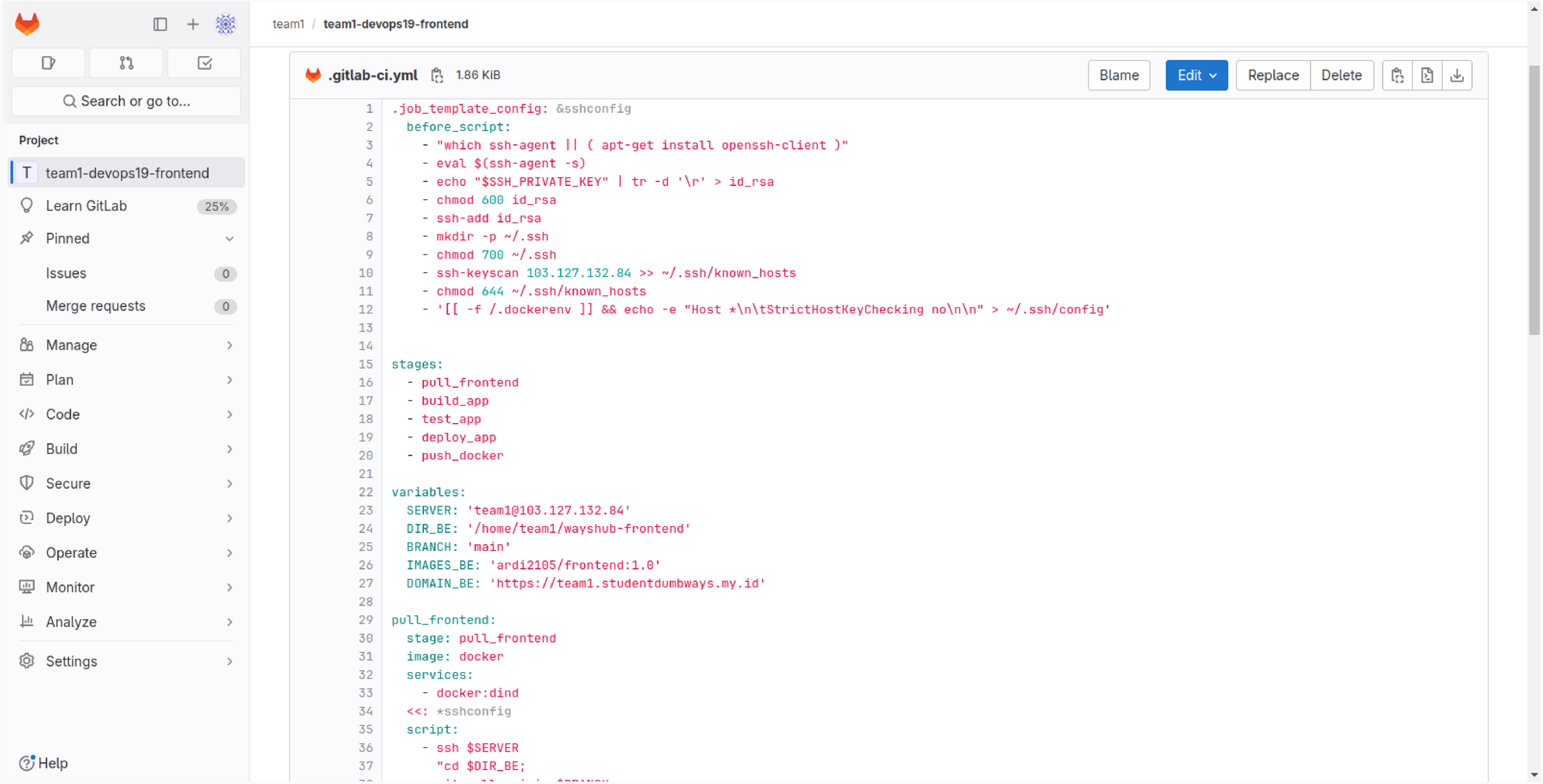
4

Building CI/CD GitLab Pipeline



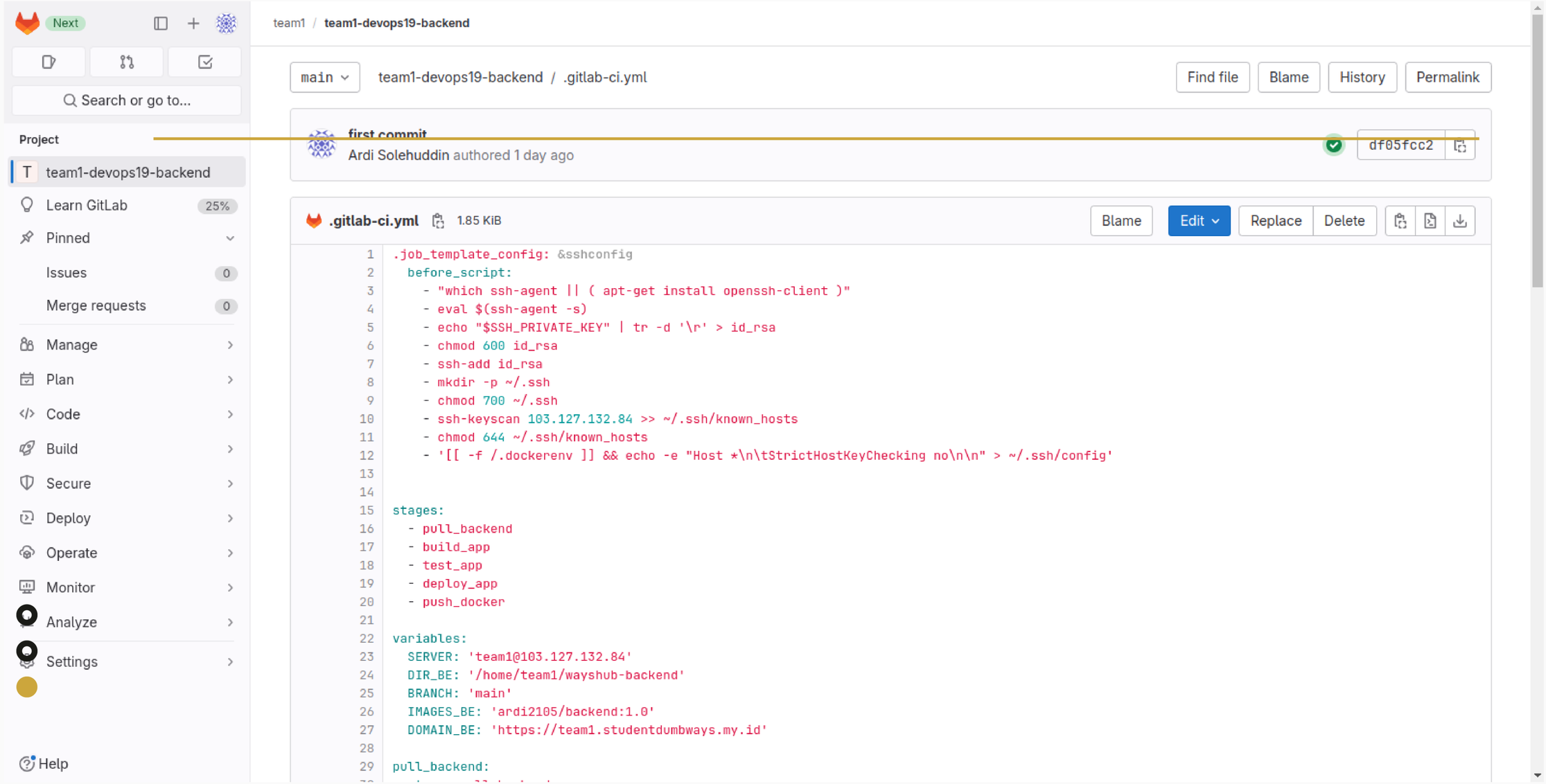
TEAM 1

5 Dokumentasi GitLab Script Frontend



TEAM 1

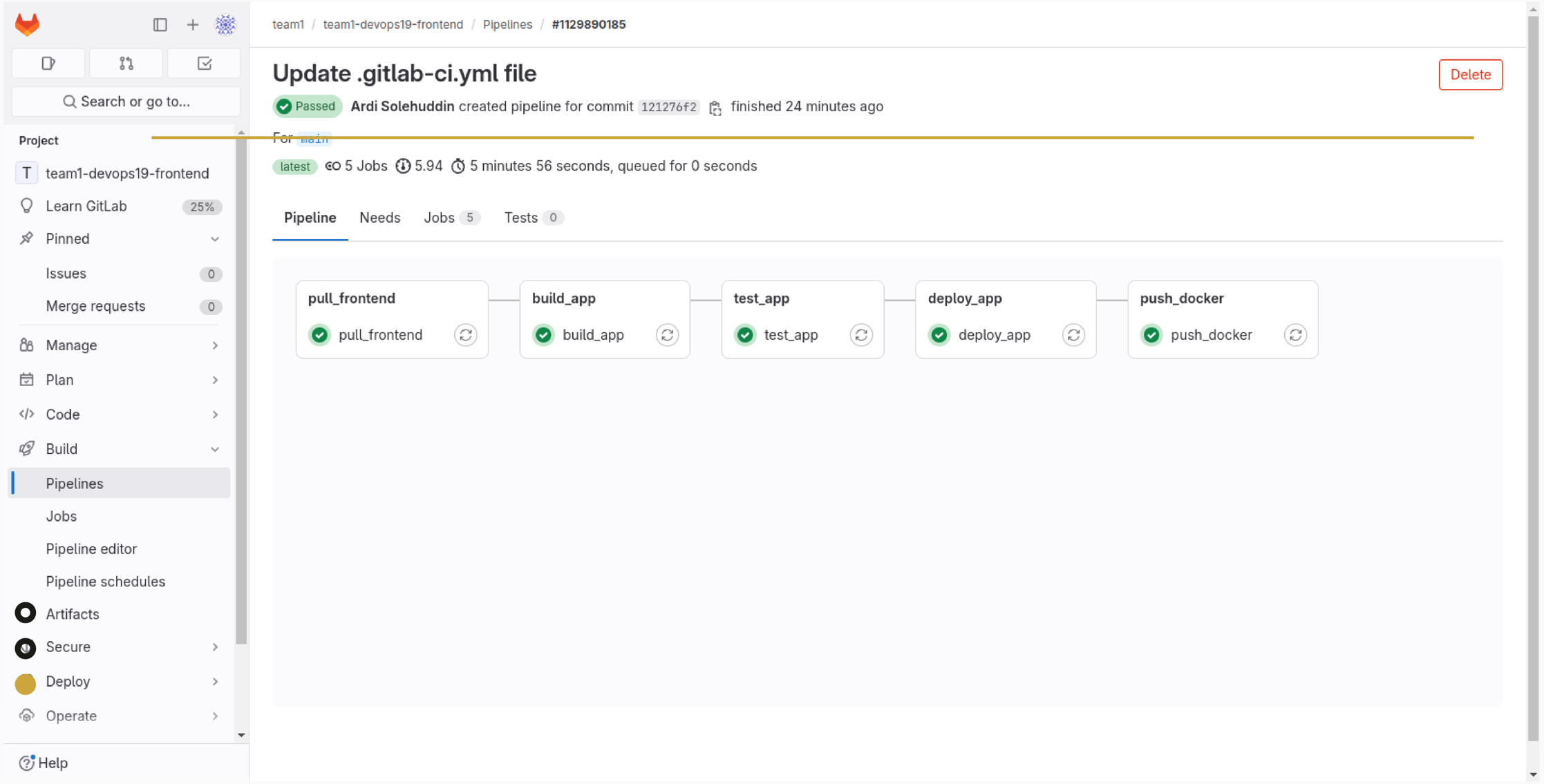
6 Dokumentasi GitLab Script Backend



7

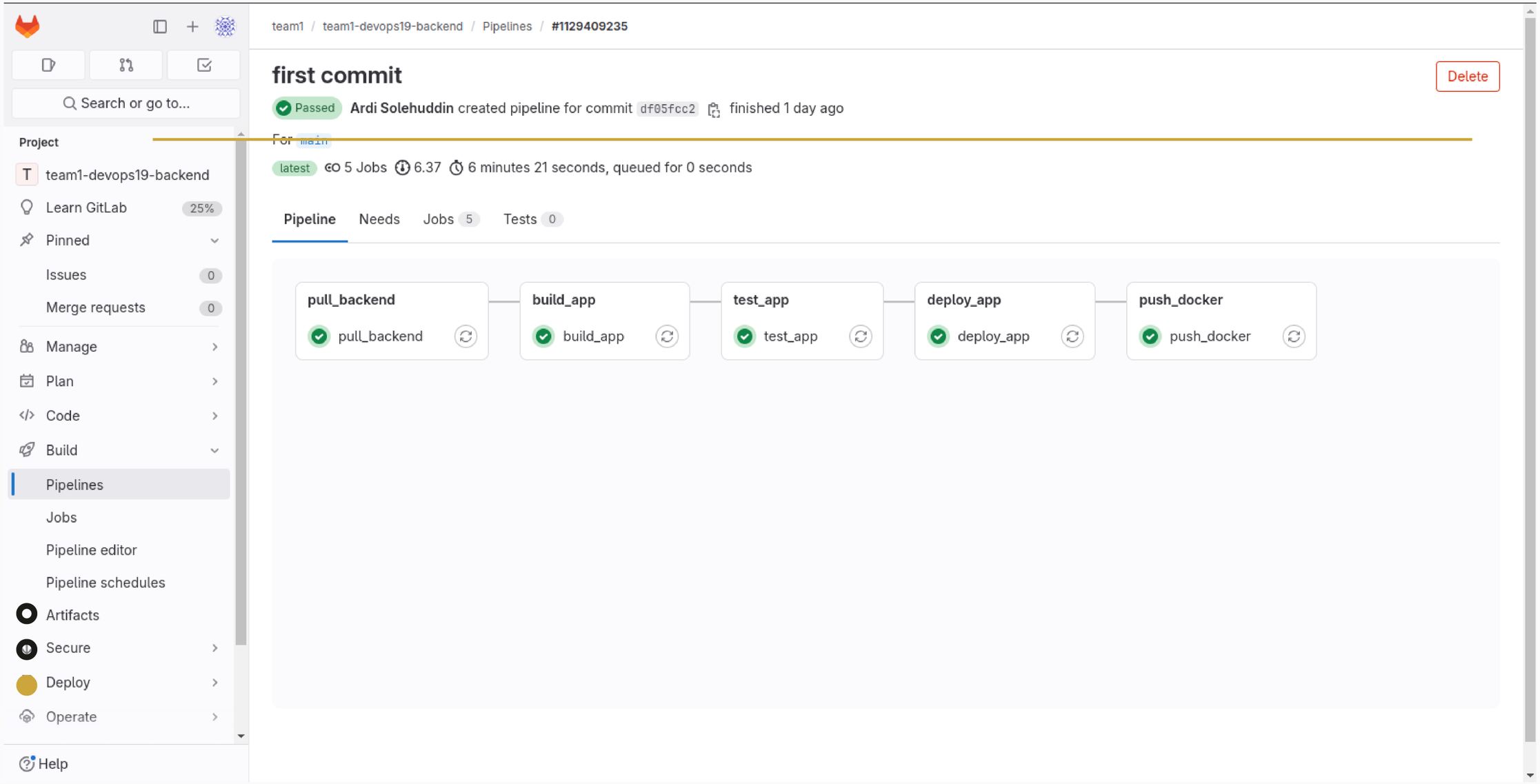
Dokumentasi Pulling, Building, Testing, Deploy dan Push Image Frontend

TEAM 1

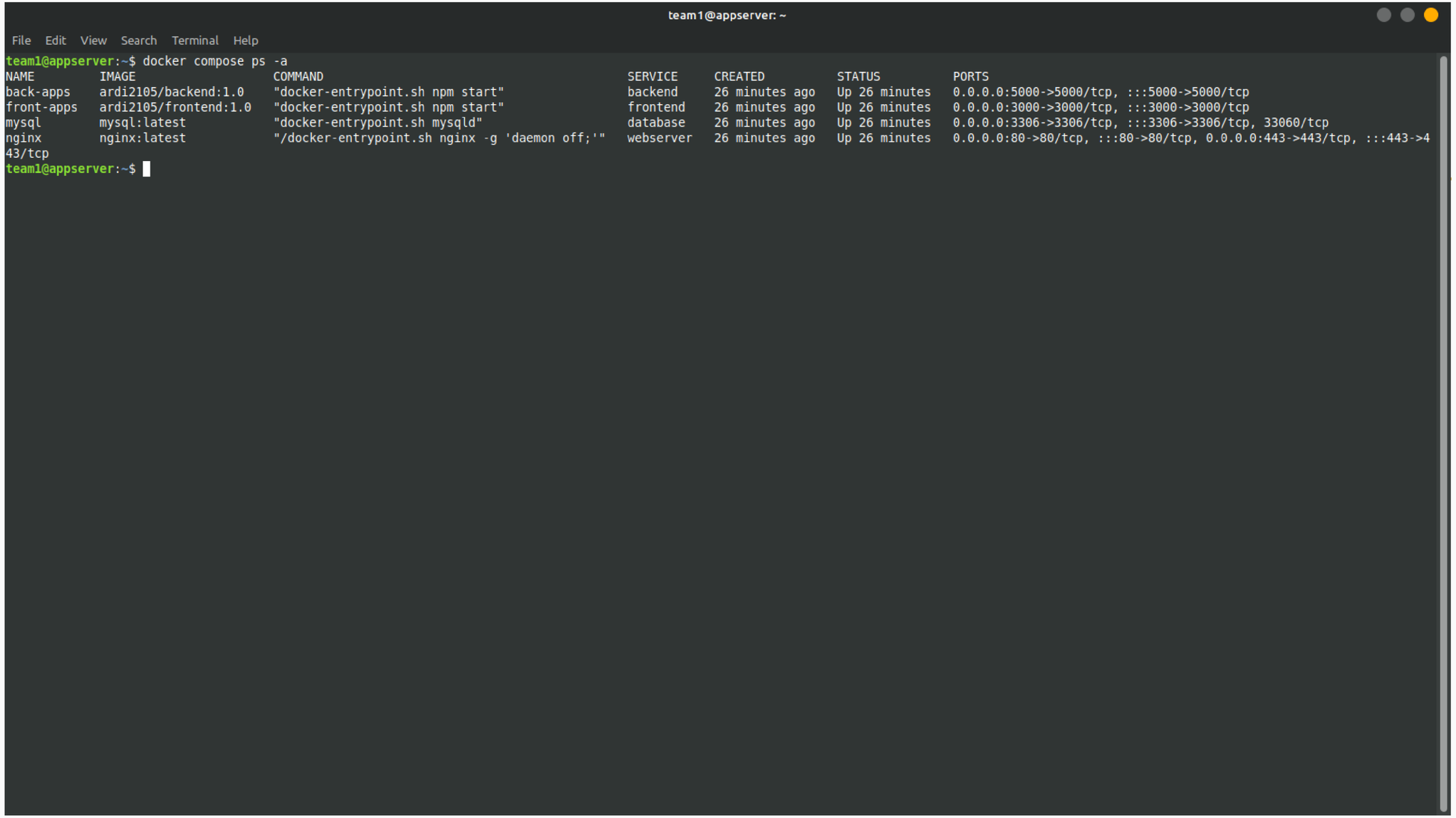


8 Dokumentasi Pulling, Building, Testing, Deploy dan Push Image Backend

TEAM 1



Dokumentasi Docker Compose



Terima Kasih