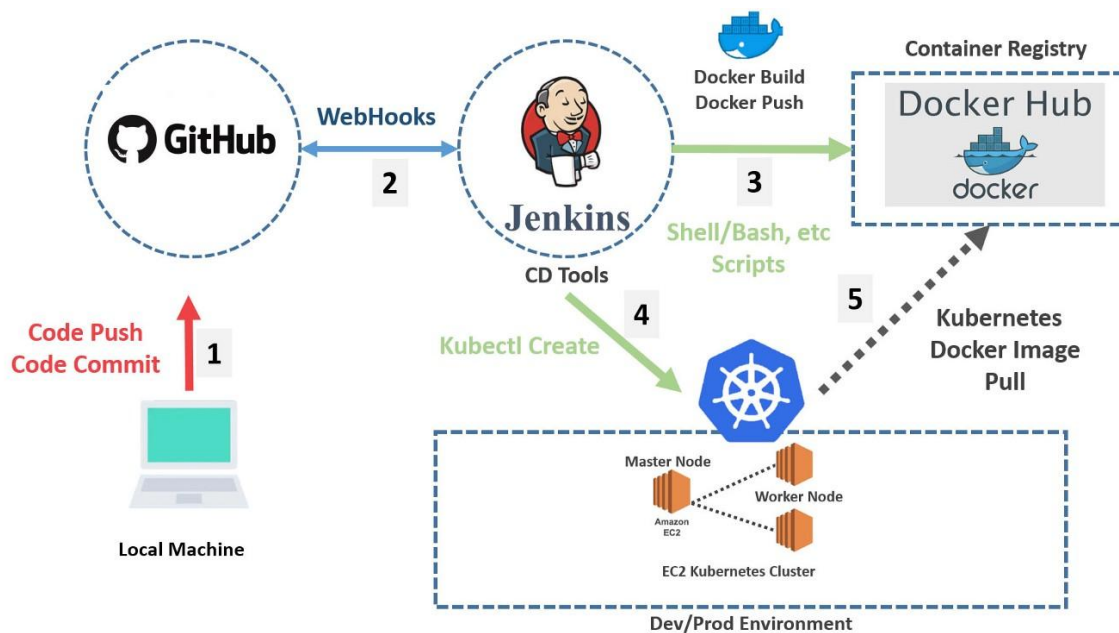


Membuat Project Website dengan Gitlab, Jenkins, DockerHub dan Kubernetes



Environment List:

1. Gitlab
2. Dockerhub
3. Jenkins
4. Kubernetes

Part - 1 Gitlab

1. Login gitlab
2. **Project -> New Project -> Create blank project**
3. Jika semua field dibawah sudah diisi, klik **Create project**

Project Name (sesuai kebutuhan anda)	first-bootcamp
Project URL (automatic)	automatic by gitlab
Project Slug (automatic)	automatic by gitlab
Project description (optional)	Membuat repository pertama untuk pelatihan bootcamp
Visibility Checklist	Private
Initialize repository with a README	checkboxlist



Create blank project

Create a blank project to house your files, plan your work, and collaborate on code, among other things.

New project · Create blank project

Project name

first-bootcamp

Project URL

https://gitlab.com/ wahyusutejo1986

Project slug

first-bootcamp

Want to house several dependent projects under the same namespace? [Create a group](#).

Project description (optional)

Membuat repository pertama untuk pelatihan bootcamp

Visibility Level

☒ Private

Project access must be granted explicitly to each user. If this project is part of a group, access will be granted to members of the group.

☐ Public

The project can be accessed without any authentication.

☒ Initialize repository with a README

Allows you to immediately clone this project's repository. Skip this if you plan to push up an existing repository.

Create project

Cancel

F first-bootcamp

- Project information
- Repository
- Issues 0
- Merge requests 0
- CI/CD
- Security & Compliance
- Deployments
- Monitor
- Infrastructure
- Packages & Registries
- Analytics
- Wiki
- Snippets
- Settings

« Collapse sidebar

wahyu > first-bootcamp

Project 'first-bootcamp' was successfully created.

F first-bootcamp

Project ID: 29708200

0

0

1 Commit 1 Branch 0 Tags 143 KB Files 143 KB Storage

Membuat repository pertama untuk pelatihan bootcamp

Catch your security vulnerabilities ahead of time!

GitLab can scan your code for security vulnerabilities. Static Application Security Testing (SAST) helps you worry less and build more.

[Learn more](#)



Auto DevOps

It will automatically build, test, and deploy your application based on a predefined CI/CD configuration.

Learn more in the [Auto DevOps documentation](#)

[Enable in settings](#)

4. Clone project first-bootcamp ke laptop atau local environment, copy url **Clone with HTTPS**

History Find file Web IDE Clone

Clone with SSH

git@gitlab.com:wahyusutejo1986/

Clone with HTTPS

https://gitlab.com/wahyusutejo1

Open in your IDE

Visual Studio Code (SSH)
Visual Studio Code (HTTPS)

```

gamuza@gamuza:~$
gamuza@gamuza:~$
gamuza@gamuza:~$ git clone https://gitlab.com/wahyusutejo1986/first-bootcamp.git
Cloning into 'first-bootcamp'...
Username for 'https://gitlab.com': wahyusutejo1986@gmail.com
Password for 'https://wahyusutejo1986@gmail.com@gitlab.com':
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 2.83 KiB | 2.83 MiB/s, done.
gamuza@gamuza:~$

```

Masukan username dan password gitlab

5. Masuk ke directory project **first-bootcamp**

Download contoh template berikut ini :

<https://drive.google.com/file/d/14x1goPSqPcDWyQmHSTN8Vmm1evKetPTL/view?usp=sharing>

atau download melalui repository

<https://gitlab.com/wahyusutejo1986/firstbootcamp>

6. Extract, kemudian copy kan hasil extract (folder assets dan index.html) tersebut kedalam folder first-bootcamp

```

gamuza@gamuza:~$ cd first-bootcamp/
gamuza@gamuza:~/first-bootcamp$
gamuza@gamuza:~/first-bootcamp$
gamuza@gamuza:~/first-bootcamp$
gamuza@gamuza:~/first-bootcamp$ ls
assets  index.html  README.md
gamuza@gamuza:~/first-bootcamp$

```

7. kemudian push atau upload file-file tersebut ke gitlab dengan cara seperti berikut :

- 1) git init
- 2) git add .
- 3) git commit -m "inital commit"

```

gamuza@gamuza:~/first-bootcamp$ git init
reinitialized existing Git repository in /home/gamuza/first-bootcamp/.git/
gamuza@gamuza:~/first-bootcamp$ git add .
gamuza@gamuza:~/first-bootcamp$ git commit -m "inital commit"
[main 6ce9a35] inital commit
40 files changed, 11490 insertions(+)
create mode 100644 assets/css/bootstrap.min.css
create mode 100644 assets/css/flex-slider.css
create mode 100644 assets/css/font-awesome.css
create mode 100644 assets/css/owl-carousel.css
create mode 100644 assets/css/templatemo-art-factory.css
create mode 100644 assets/fonts/FlatIcon.woff
create mode 100644 assets/fonts/FontAwesome.otf
create mode 100644 assets/fonts/Flexslider-icon.eot
create mode 100644 assets/fonts/Flexslider-icon.svg
create mode 100644 assets/fonts/Flexslider-icon.ttf
create mode 100644 assets/fonts/Flexslider-icon.woff
create mode 100644 assets/fonts/Fontawesome-webfont.eot
create mode 100644 assets/fonts/Fontawesome-webfont.svg
create mode 100644 assets/fonts/Fontawesome-webfont.ttf
create mode 100644 assets/fonts/Fontawesome-webfont.woff
create mode 100644 assets/fonts/Fontawesome-webfont.woff2
create mode 100644 assets/fonts/slick.eot
create mode 100644 assets/fonts/slick.svg
create mode 100644 assets/fonts/slick.ttf
create mode 100644 assets/fonts/slick.woff
create mode 100644 assets/images/about-icon-01.png
create mode 100644 assets/images/about-icon-02.png

```

- 4) git push

```

gamuza@gamuza:~/first-bootcamp$ git push
Username for 'https://gitlab.com': wahyusutejo1986@gmail.com
Password for 'https://wahyusutejo1986@gmail.com@gitlab.com':
Enumerating objects: 48, done.
Counting objects: 100% (48/48), done.
Delta compression using up to 4 threads
Compressing objects: 100% (47/47), done.
Writing objects: 100% (47/47), 981.40 KiB | 10.55 MiB/s, done.
Total 47 (delta 0), reused 0 (delta 0)
To https://gitlab.com/wahyusutejo1986/first-bootcamp.git
   d115fbe..6ce9a35  main -> main
gamuza@gamuza:~/first-bootcamp$

```

Masukan username dan password gitlab
(anda dapat melakukan konfigurasi username dan password agar tidak perlu lagi memasukkan username dan password untuk setiap aksi)

5) files berhasil di upload ke repository gitlab untuk project **first-bootcamp**

main

first-bootcamp /

History

Find file

Web IDE

Clone

Initial commit

wahyu authored 5 minutes ago

6ce9a354

Upload File

README

Add LICENSE

Add CHANGELOG

Add CONTRIBUTING

Add Kubernetes cluster

Set up CI/CD

Configure Integrations

Add Security Testing

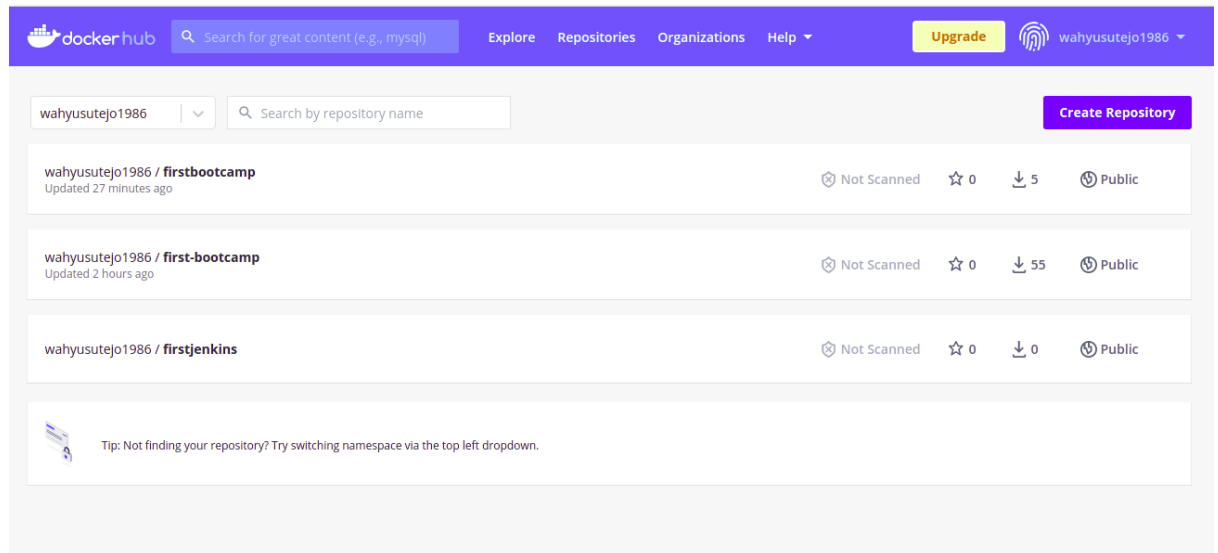
Name	Last commit	Last update
assets	initial commit	5 minutes ago
README.md	Initial commit	17 minutes ago
index.html	initial commit	5 minutes ago

README.md

Part - 2 Dockerhub

=====

1. Register akun gratis di <https://hub.docker.com/>
2. Jika proses image docker sudah terintegrasi dengan jenkins, maka image akan otomatis tercreate seperti gambar di bawah ini




3. Jika image container sudah tersedia di dockerhub maka kita akan lebih mudah untuk melakukan deployment di kubernetes ataupun platform lainnya yang support docker machine.

Part - 3 Jenkins

=====

1. Login ke Jenkins

`http://jenkins-{namateam}.bootcampdevops.id:8080/`



Welcome to Jenkins!

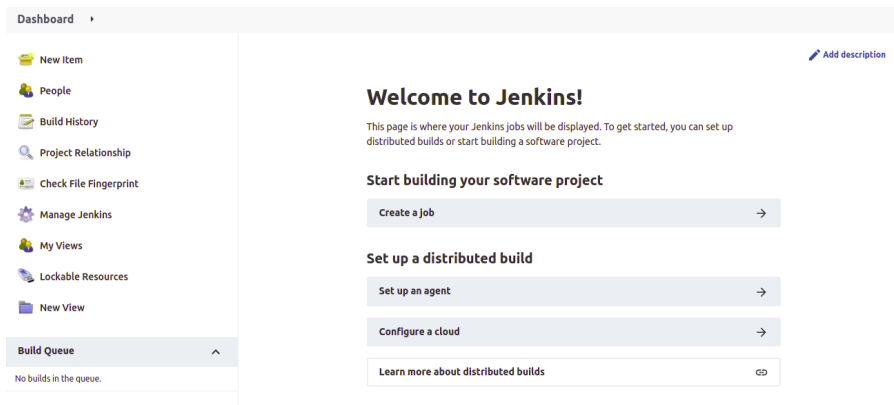
Username

Password

☐ Keep me signed in

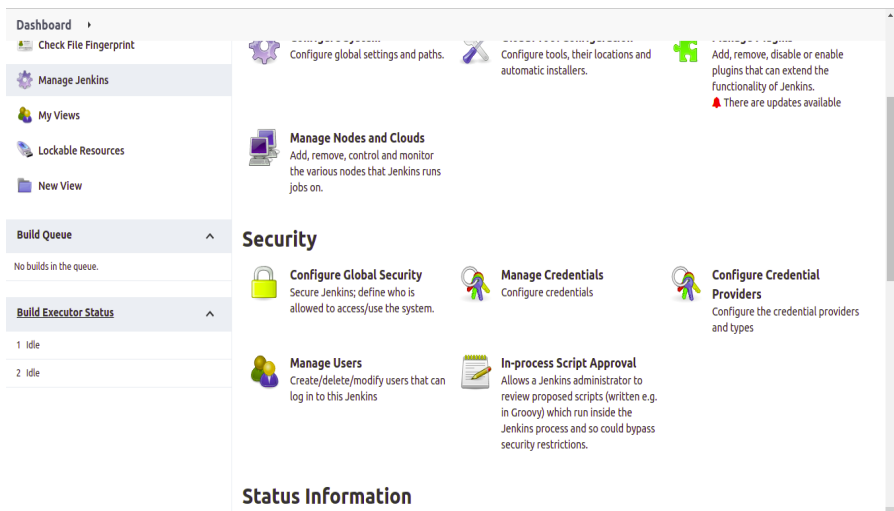
masukan username dan password

2. Tampilan dashboard jenkins



The screenshot shows the Jenkins Dashboard. On the left is a sidebar with navigation links: New Item, People, Build History, Project Relationship, Check File Fingerprint, Manage Jenkins, My Views, Lockable Resources, and New View. Below these is the 'Build Queue' section, which is currently empty. The main content area has a 'Welcome to Jenkins!' message and instructions. It includes a 'Start building your software project' section with a 'Create a job' button, and a 'Set up a distributed build' section with buttons for 'Set up an agent', 'Configure a cloud', and 'Learn more about distributed builds'.

3. Create credential untuk autentikasi dengan gitlab klik **Manage Credentials**



The screenshot shows the 'Manage Credentials' page in Jenkins. The left sidebar is the same as the dashboard. The main content area is titled 'Manage Credentials' and contains a list of credential types: 'Configure Global Security', 'Manage Nodes and Clouds', 'Security', 'Configure Global Security', 'Manage Credentials', 'Configure Credential Providers', 'Manage Users', 'In-process Script Approval', and 'Status Information'. The 'Manage Credentials' section is highlighted, showing a list of credentials. The 'Status Information' section at the bottom shows the Jenkins version and other system details.

4. Scroll dan Klik **global**



5. Klik **Add Credentials** (masukan gitlab username dan password)

Back to credential domains

Add Credentials

Kind

Username with password

Scope

Global (Jenkins, nodes, items, all child items, etc)

Username

gitlabu-username

☐ Treat username as secret

Password

ID

Description

gitlab connection

OK

Dashboard > Credentials

Credentials

T	P	Store	Domain	ID	Name
		Jenkins	(global)	70a43e68-babc-4e7d-aff4-32ec99770b5f	GitLab API token (Jenkins-gitlab-token)
		Jenkins	(global)	9a4da7c6-1e49-4eb1-9f03-9f354a3c1dfd	git (Gitlab-Auth-SSH)
		Jenkins	(global)	4e7ce8d6-08df-46fe-8dd5-daa89f65085f	wahyusutejo1986@gmail.com/***** (gitlab auth with username and password)

Icon: S M L

Stores scoped to Jenkins

P	Store	Domains
	Jenkins	(global)

6. ssh remote ke jenkins untuk membuat private keys dan public keys dengan perintah

ssh root@jenkins-{namateam}.bootcampdevops.id

```
root@jenkins-server:~# ls
id_rsa_gitlab  id_rsa_gitlab.pub  local-certs  snap
root@jenkins-server:~#
```

[illegible]

```
root@jenkins-server:~# cat id_rsa_gttlab.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQGDQZlGz5aR1Nhopa83gfEocZmdaVMP/rwUQ7E3EdIAfyB5m2W7DtKlC0ZnkTbz/tq6hoVBCbtl69M8Lgvtm8/F8AlkompCYXF0i+kr6
R5ux6d4f+cl+20BB6GRETfRtS10+am+QXBAWd8DQfVg+6NcJpTHto7cVbs4Z0krDDxtLxdquvhuw0u+X6ZEPKXSKCROA7QD9bWmGKBZd4h5LduowoyfscV9hLVOP9ydp82dGVx8vT81r
7818JBPwPwJCEc7S5a2ln70K30WmXt1Agch9iOVET/rDf0AJX+fQULXgom+ec+IOPCNWR5pTgwrotXQcVlcV3xjktg7Jl3LTqTdhjsongmPT14XVmdPr1PF/TVtpkPmbhFIPL5NMD
WagdnktVYLKA08bL1C53z5zQ787/2682WnwUuWkXRLaEVAB55y2xkLGduMSuvGT4vhp/7KESqJlGUCPHTXWUXZrD1Ahot+hXkIA08zMK2iM7/+lOmU= root@jenkins-server
root@jenkins-server:~#
```

SSH Keys

SSH keys allow you to establish a secure connection between your computer and GitLab.

Add a SSH key

To add a SSH key you need to [generate one](#) or use an [existing key](#).

Key

Paste your public SSH key, which is usually contained in the file `~/.ssh/id_ed25519.pub` or `~/.ssh/id_rsa.pub` and begins with `'ssh-ed25519'` or `'ssh-rsa'`. Do not paste your private SSH key, as that can compromise your identity.

```
ssh-ed25519 AAAAB3NzaC1yc2EAAAADAQABAAQGDQZ0gZr5a/1Nhp0A83gcF0cZmdaVMP/rwUQ7E3EsdiaFBS2mW7Dk3cdQZmkTbzj/QdhoV/Bcbls9M8Jdytm8/F8aLompC1YF1+Hr8R5uxfa/d/c+2J08BBRETRH/L5i0+mmxHPZwe8DFQV+6NcPHTCo7CvbsXzOkRkDdXtLxdqUJwhu0U+X6ZePxKv5SCKRQAQ7D09wBGKBZd4h5dUowofy5cVh0VPy9dp82dGVx8vT81r/8718JpPwJaJcETCSA52im70K3OWNIXt1IagQh9J0vETJ/DF8QJw+FeuXlQgm+ec+I0P2NNR5pTqwwr2XK0KcVxmMqk437LTtGhphdgmPDKVAMd1rPp1T1YkVakMbbHf8P8N5Wv8ag2nVtLXaQg8lC53z5qT88/67J2tHwUuWkXRlAEVAB55zKzLdGmUdu5vC7dwhp/7KE5qJGUCXPHTXWUXZrD1Ahot+H+kxiAO8Zmk2IMZ7+lomU+root@jenkins-server
```

Title

root@jenkins-server

Expires at

mm/dd/yyyy

Give your individual key a title. This will be publicly visible.

Key will be deleted on this date.

Add key

User Settings > SSH Keys > root@jenkins-server

Search settings

SSH Key

Title: root@jenkins-server

Created on: Sep 18, 2021 9:08am

Expires: Never

Last used on: Never

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgDD0Zigzar5a/1NhopA83gfEocZmdaVMP/rwUQ7E3E

Fingerprints

MDS: de:47:36:e3:d3:48:8c:53:7f:84:6a:7f:97:d2:be:61

SHA256: SE306r5XF1HmwCynqp1N03bmK71ARc/Su/VHLMEEOEU

Delete

8. Kembali ke Jenkins, untuk membuat credential baru menggunakan private keys yang telah dibuat sebelumnya, pilih **SSH Username with private key** (username sesuaikan dengan private key yang anda buat)

Kind

SSH Username with private key

Scope

Global (Jenkins, nodes, items, all child items, etc)

ID

Description

jenkins private keys

Username

root

☐ Treat username as secret

Private Key

Username

root

☐ Treat username as secret

Private Key

Enter directly

Key

Enter New Secret Below

NRRAAAWECARQAPATCAZU1UNZQY+RW31TANVRK4XANNBZIMLTU/00FCUAXALN3ANQY6ZLLUW75pXNGZpE282/7U0oaFQXHIm+vTPC4L7ZvPxfgi5KJqQL2CBdNtpk+kürsen2nf3PottA0RRRE0Ys66+SNPpvsRz8InnFEBVYPuJXiZ7R7RqQ3FW7F+HzpEZAwbS8XarLYbsDlPl+mXj85sUqikTgE0w9PcARigWxe1Ui3bjsK4Mn7HfDlYrzj8vXaFNNrLcFL70/Na//04vCT6T1q

Passphrase

9. Membuat project jenkins dan integrasikan dengan gitlab project


Klik **New Item**

Enter an item name “ first-bootcamp ”


Klik **"Fresstyle project"**

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
Creates a container that stores nested items in it. Useful for grouping things together. Unlike view, which is just a filter, a folder has its own namespace, so you can have multiple things of the same name as long as they are in different folders.

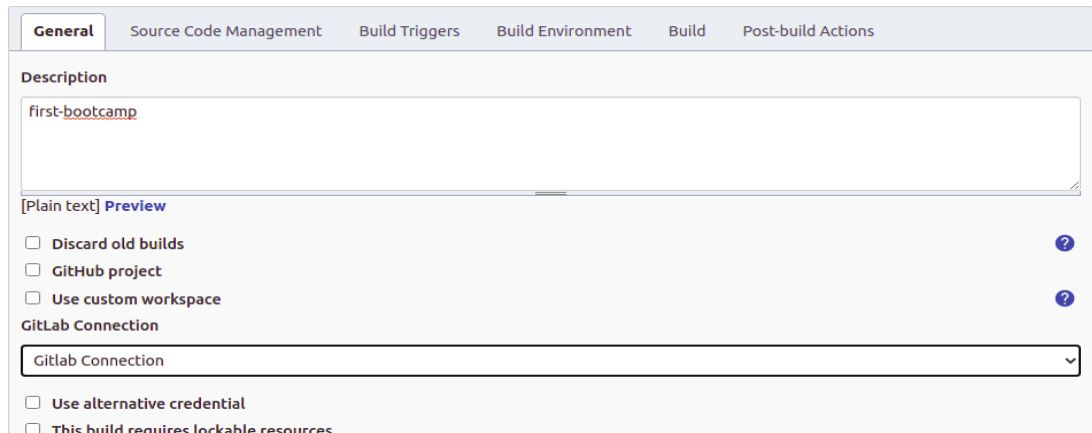
OK

Cancel

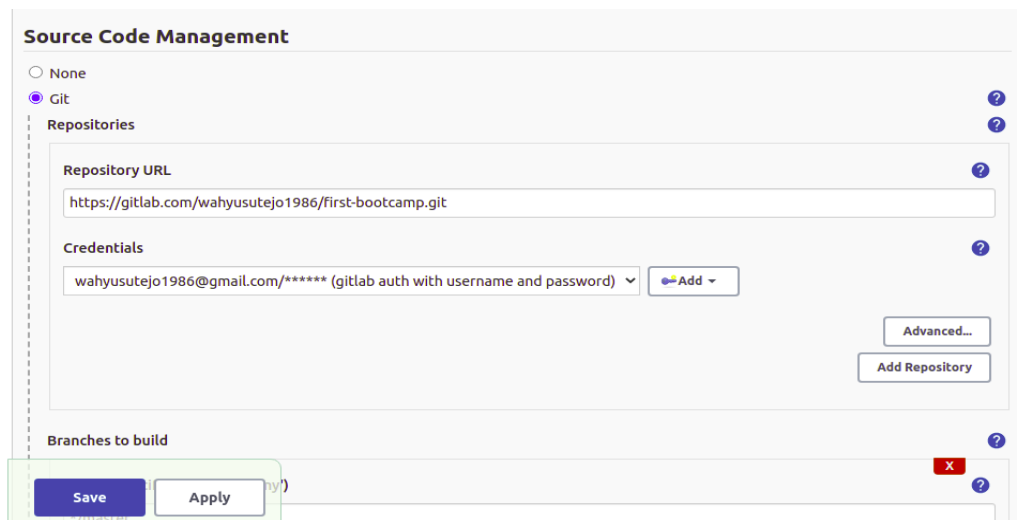
Klik OK

10. Masukan gitlab project yang akan kita integrasikan dengan jenkins

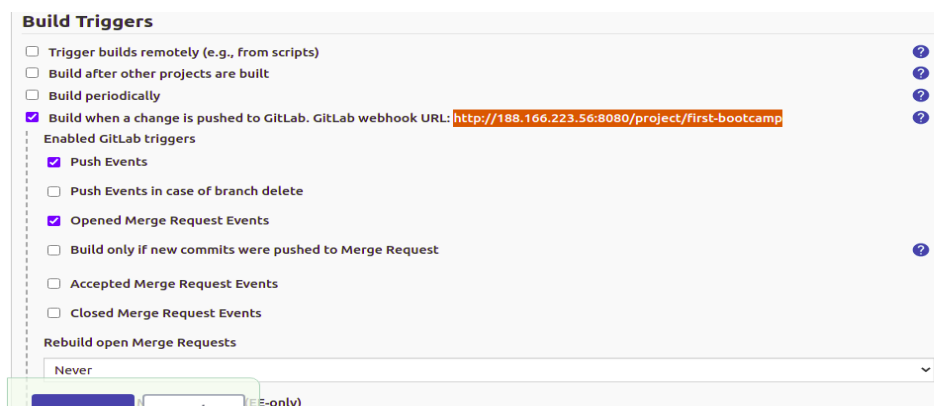
Isi **Description** sesuaikan dengan kebutuhan anda



Klik tab **Source Code Management**



Pada Bagian Build Trigger klik **Build when a change is pushed to gitlab** kemudian copy webhook url untuk kemudian akan kita gunakan di gitlab



Klik **Apply** kemudian **Save**

11. Integrasikan webhook jenkins ke gitlab project, pilih project **first-bootcamp**
Pilih **Settings->Webhook**

Webhooks

Webhooks enable you to send notifications to web applications in response to events in a group or project. We recommend using an [integration](#) in preference to a webhook.

URL

URL must be percent-encoded if necessary.

Secret token

Use this token to validate received payloads. It is sent with the request in the X-Gitlab-Token HTTP header.

Trigger

☒ **Push events**

URL is triggered by a push to the repository

☐ **Tag push events**

URL is triggered when a new tag is pushed to the repository

Karena pada praktikum ini tidak menggunakan SSL, jadi kita akan unchecked **Enable SSL Verification**

SSL verification
☐ **Enable SSL verification**

Add webhook

Project Hooks (0)
No webhooks found, add one in the form above.

Klik **Add Webhook**

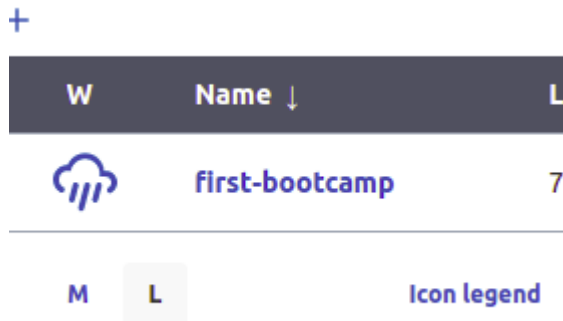
12. Test lakukan perubahan pada project **first-bootcamp**

Klik **README.md** - > **Edit** -> **Hapus** beberapa bagian

```
1 # first-bootcamp
2
3 Membuat repository pertama untuk pelatihan bootcamp
4 |
5 If you have run out of energy or time for your project, put a note at the top of the README saying that
6 development has slowed down or stopped completely. Someone may choose to fork your project or volunteer to step
7 in as a maintainer or owner, allowing your project to keep going. You can also make an explicit request for
8 maintainers.
```

Klik **Commit changes**

13. Test Build repo dengan jenkins, kembali ke dashboard jenkins
Klik **first-bootcamp**



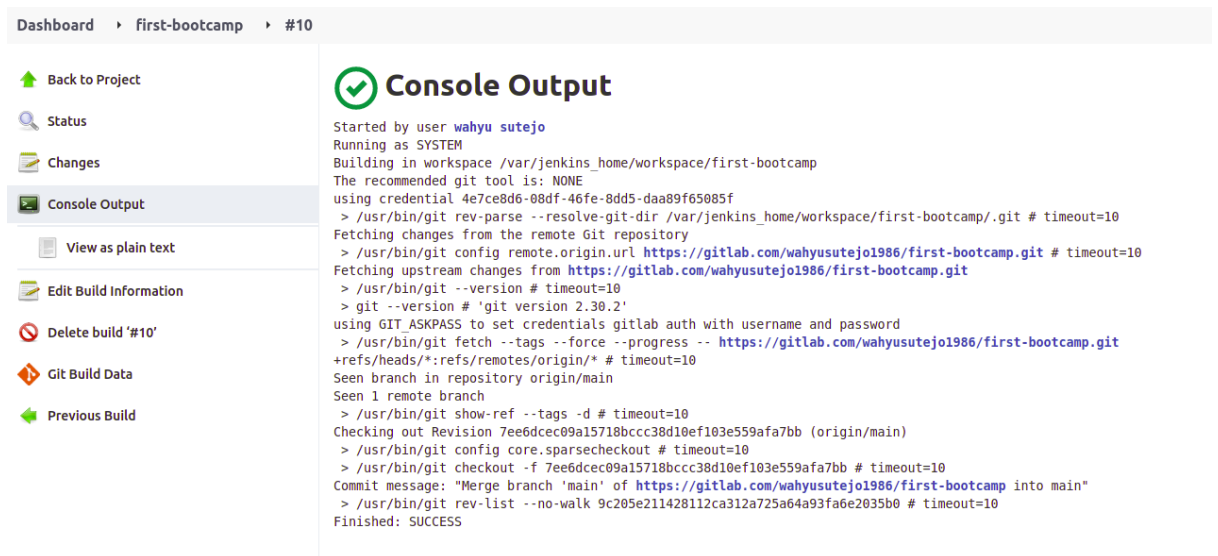
14. Klik **Build Now**, maka akan muncul build history pada bagian paling kiri bawah



Jika build success akan muncul simbol cheklist warna hijau, kemudian klik untuk melihat seluruh status report.

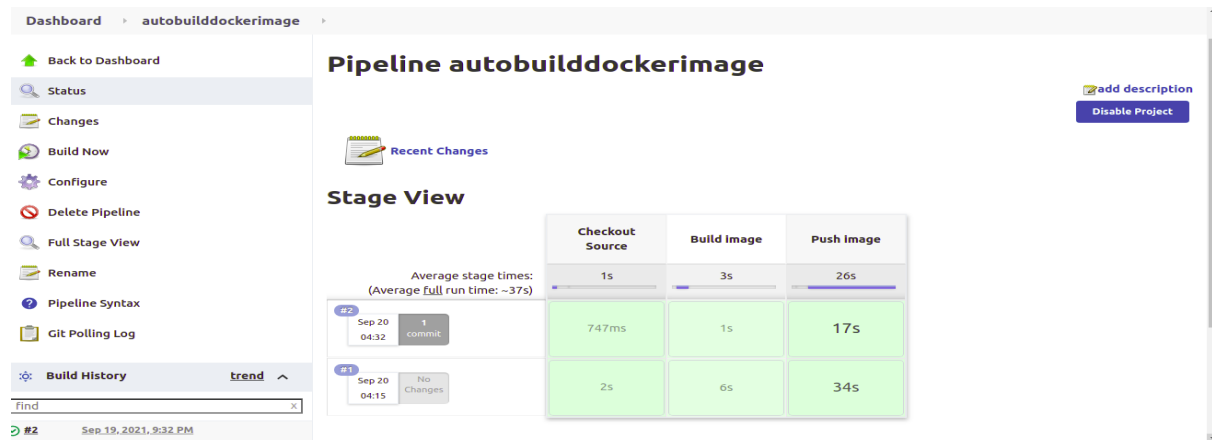


Klik **Console Output** untuk melihat status processnya



Build Image Docker dan push ke dockerhub melalui jenkins pipeline

Tampilan pipeline jenkins untuk build docker image dan push image ke dockerhub



Langkah-langkah membuat jenkins pipeline

- 1) Klik New Item
- 2) Masukan nama item, kemudian pilih pipeline seperti gambar dibawah

The screenshot shows the Jenkins 'Enter an item name' dialog. The input field contains 'autopushimagetodockerhub'. Below the input field, there are three options: Freestyle project, Pipeline, and Multi-configuration project. The 'Pipeline' option is selected. The 'OK' button is highlighted.

- 3) Klik OK
- 4) Klik tab **Build Triggers**, kemudian masukan H/02 * * * * seperti gambar dibawah

The screenshot shows the Jenkins 'Build Triggers' tab. The 'Schedule' field is set to 'H/02 * * * *'. The 'Poll SCM' checkbox is checked. The 'Save' button is highlighted.

- 5) Klik Tab **Pipeline**
- 6) Masukkan script berikut

```
pipeline {
  agent any

  stages {
    stage('Checkout Source') {
      steps {
        checkout([$class: 'GitSCM', branches: [[name: '**']], extensions: [],
userRemoteConfigs: [[credentialsId: 'ceaa6a55-d820-4d4f-a68c-d067300b55a1',
url: 'https://gitlab.com/wahyusutejo1986/firstbootcamp.git']]])
      }
    }

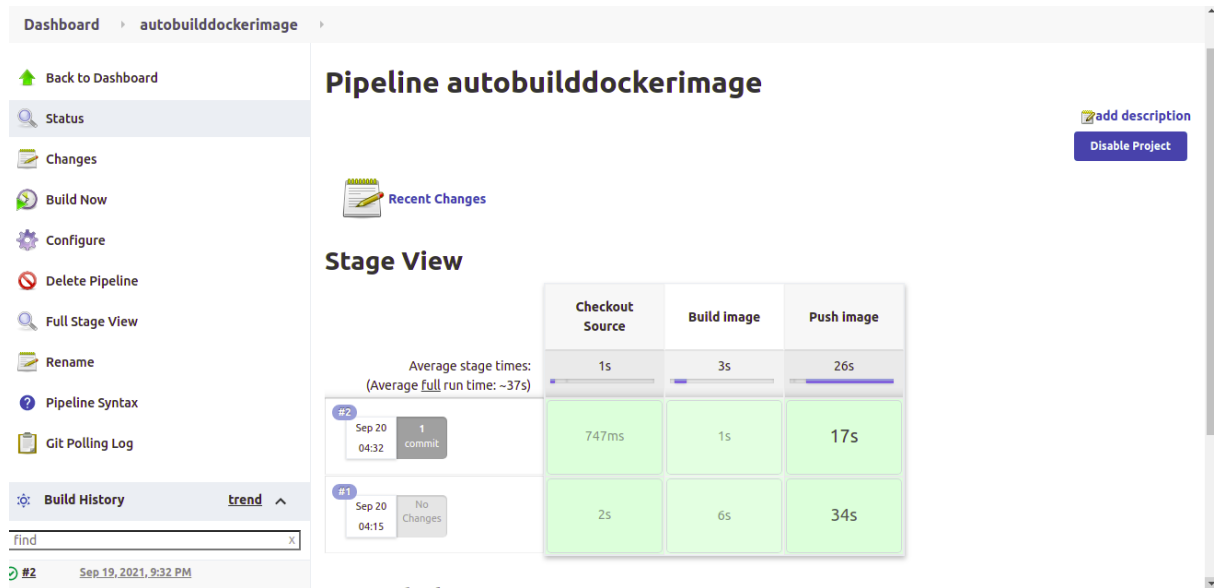
    stage("Build image") {
      steps {
        script {
          myapp =
docker.build("wahyusutejo1986/firstbootcamp:${env.BUILD_ID}")
        }
      }
    }

    stage("Push image") {
      steps {
        script {
          docker.withRegistry('https://registry.hub.docker.com',
'dockerhub_connection') {
            myapp.push("latest")
            myapp.push("${env.BUILD_ID}")
          }
        }
      }
    }
  }
}
```

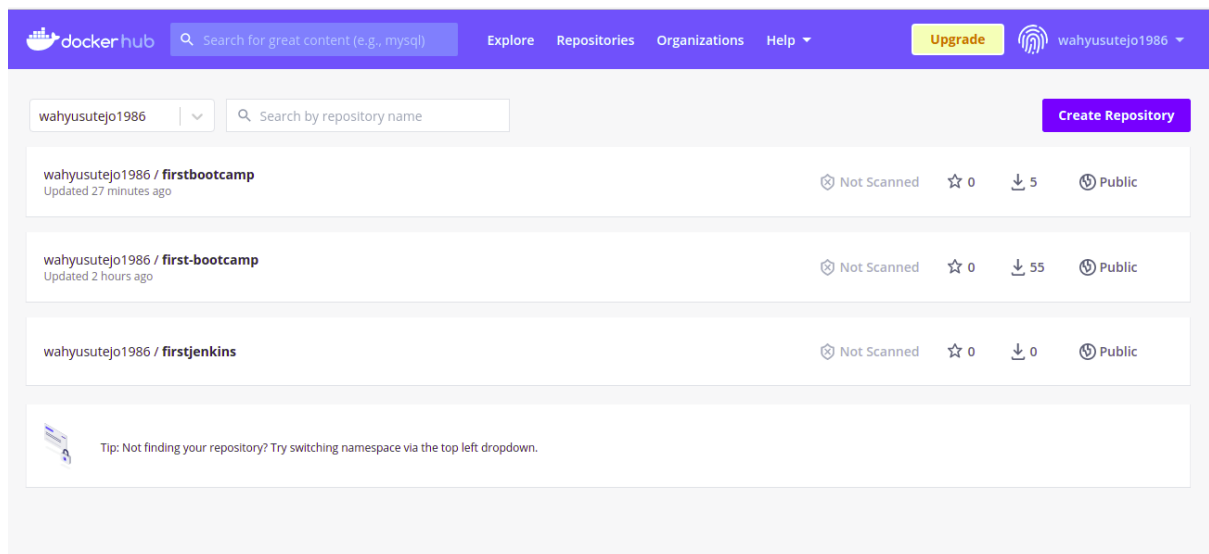
Note:

Docker Registry diisi dengan id credential dockerhub anda, kemudian untuk repository url diganti sesuai dengan url project di gitlab anda

- 7) Selanjutnya kita akan test pipeline yang sudah kita buat dengan cara klik **Build Now**
- 8) Tampilan proses pipeline jika berhasil akan seperti gambar dibawah ini



- 9) Periksa image di akun dockerhub anda, jika berhasil maka akan terdapat image baru sesuai dengan versi pipeline di jenkins, contoh image berhasil seperti gambar dibawah ini.

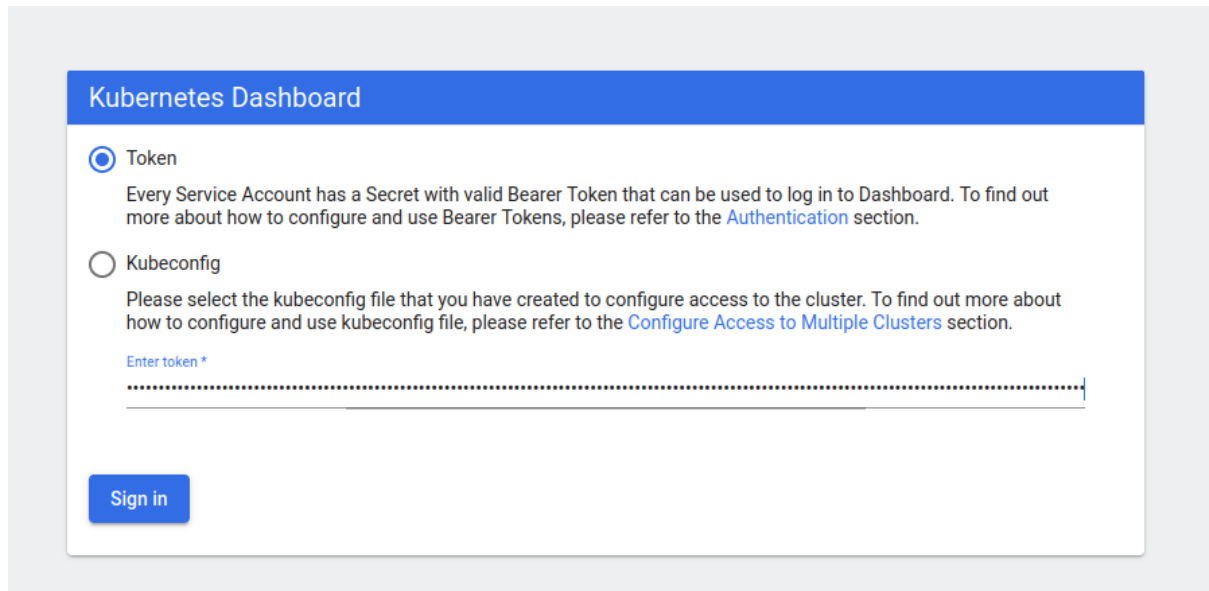


Part - 4 Kubernetes

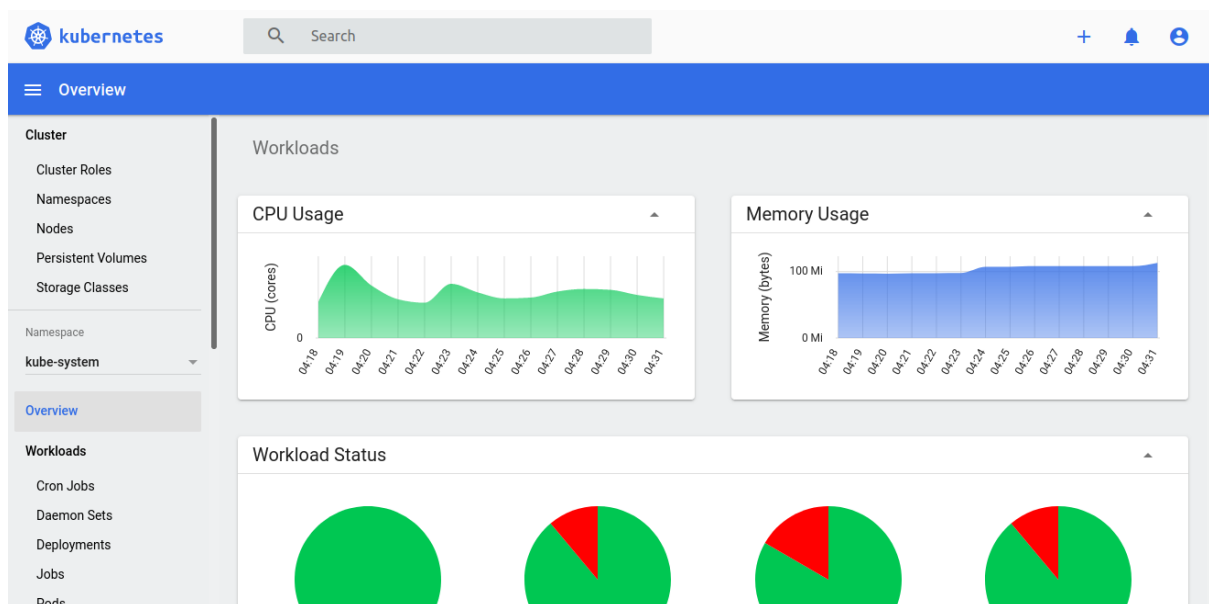
=====

Automatic Deploy aplikasi ke kubernetes jika hasil build dan test success

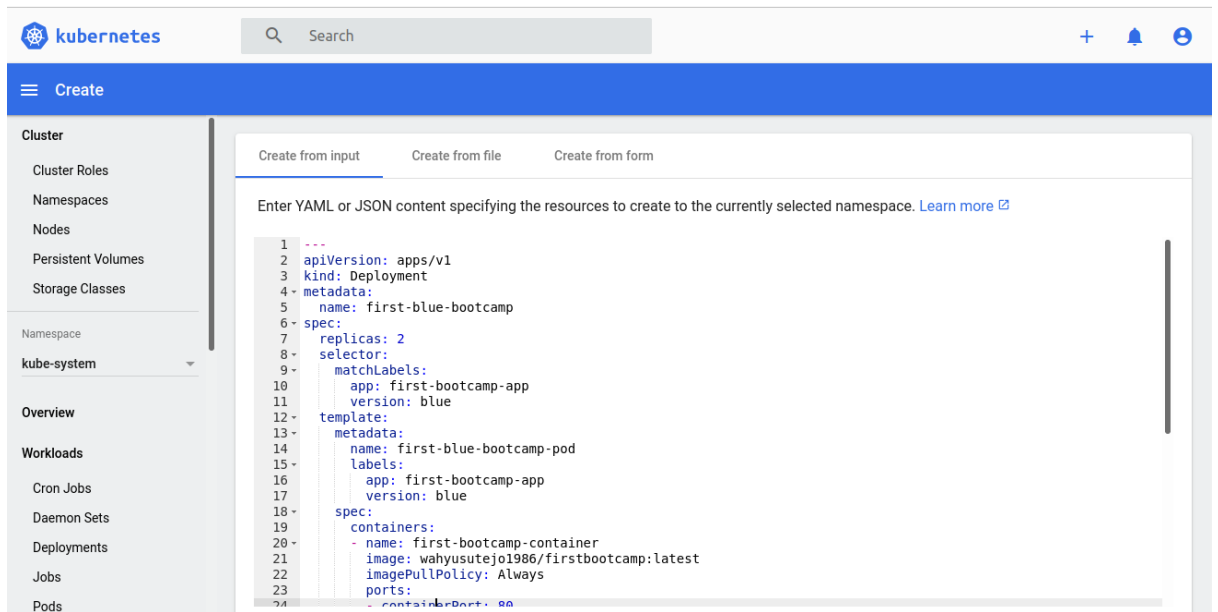
1. Login kubernetes



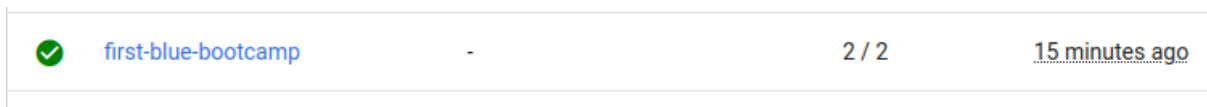
Tampilan jika login kubernetes berhasil



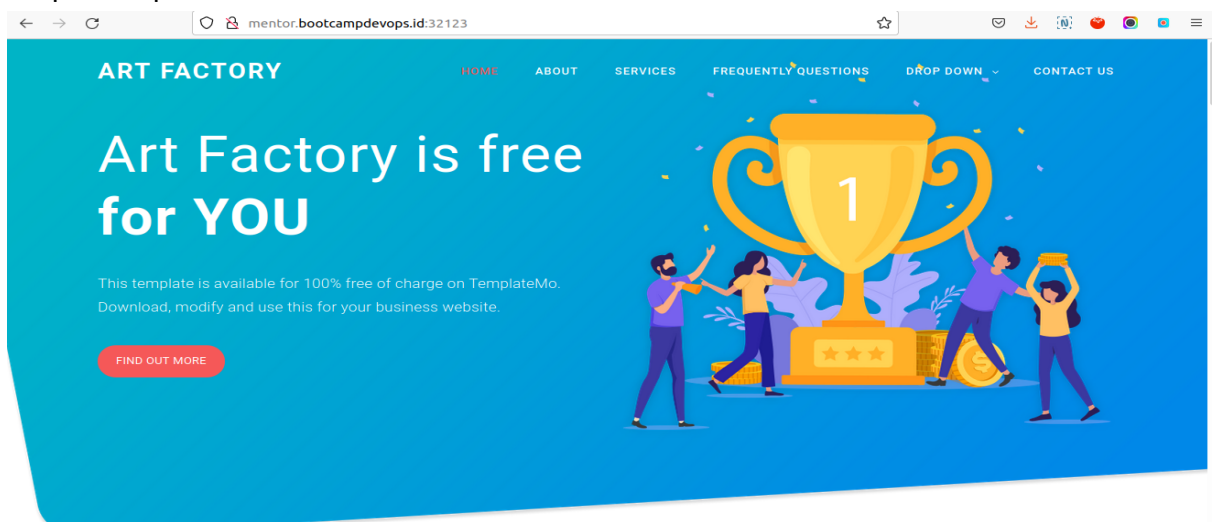
2. Deploy container menggunakan docker image di dockerhub
copy isi dari file **firstbootcamp.yml** lalu
3. Klik tanda **+** di kubernetes, maka akan muncul tampilan seperti ini, kemudian paste seluruh isi dari file **firstbootcamp.yml**



4. Klik **Upload**, tunggu hingga proses selesai
5. Jika deployment berhasil maka akan muncul keterangan seperti gambar dibawah ini



6. Akses aplikasi melalui browser <http://namadomain:32123/> maka akan muncul tampilan seperti dibawah ini



Selamat anda telah berhasil proses CI/CD aplikasi menggunakan kubernetes, pekerjaan rumah selanjutnya adalah bagaimana caranya agar keseluruhan proses menjadi automation.