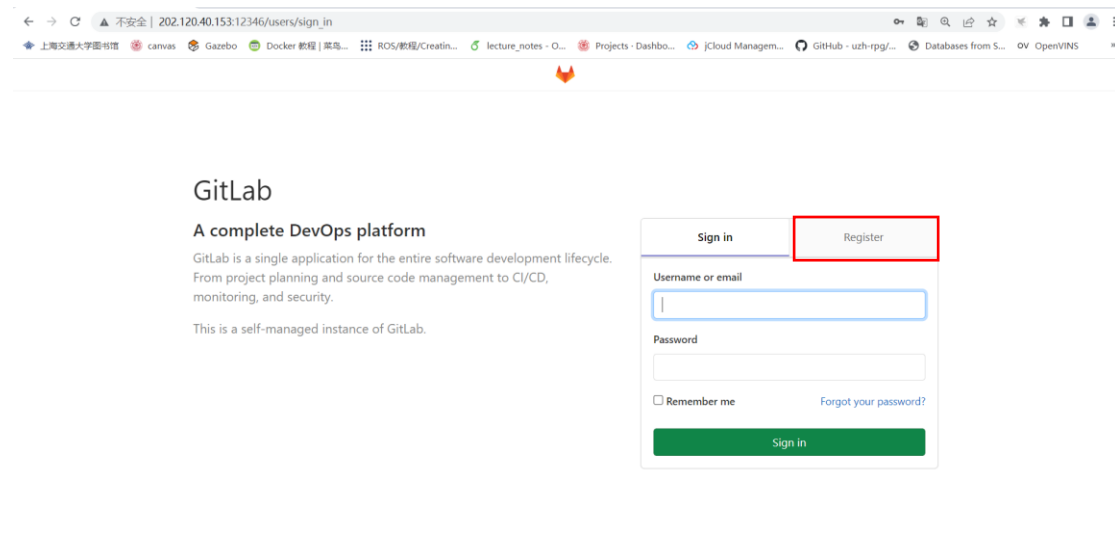


0. 注册:

在保证连接校内网络时, 访问 <http://202.120.40.153:12346/>, 点击 register

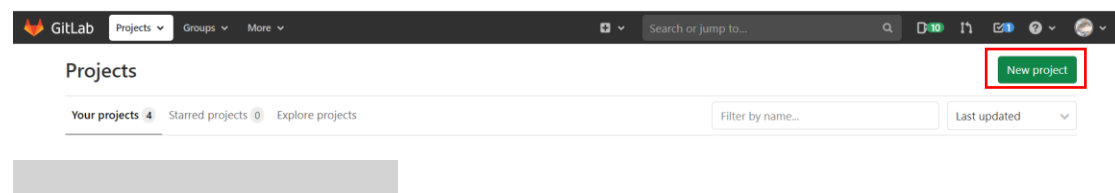


其中 First name 和 Last name 是 gitlab 中显示的名称, username 是登陆时用的用户名, 然后设置好邮箱, 密码, 点击 Register, 等待管理员同意后就可以正常登录 gitlab。

Sign in	Register
First name	Last name
<input type="text"/>	<input type="text"/>
Username	
<input type="text"/>	
Email	
<input type="text"/>	
Password	
<input type="text"/>	
Minimum length is 8 characters	
<input type="button" value="Register"/>	

1. 创建项目

1.1 登录之后如果是首次进入 gitlab, 可以点击提示页中的 New blank project, 直接进入 1.2, 否则正常情况下点击下图中的 new project 创建新项目。



1.2 输入项目名称，如果想要创建后就可以直接使用项目的代码仓库，请勾选最下方的 Initialize repository with a README 选项，否则，想要从已有的仓库直接导入到本项目的话就先不要勾选。

New project

A project is where you house your files (repository), plan your work (issues), and publish your documentation (wiki), among other things.

All features are enabled for blank projects, from templates, or when importing, but you can disable them afterward in the project settings.

Information about additional Pages templates and how to install them can be found in our [Pages getting started guide](#).

Tip: You can also create a project from the command line. [Show command](#)

Blank project | Create from template | Import project

Project name
test6.3

Project URL
http://202.120.40.153:12346/chuan.cao/

Project slug
test6.3

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

Project description (optional)
Description format

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.
- ☐ Internal
The project can be accessed by any logged in user.
- ☐ 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](#)

1.3 点击 create project,如果上一步勾选了 Initialize repository with a README，左侧会直接出现 Repository 项，可以参考 2. CI 开发的步骤，否则，可以参考 Without_README_operation.pdf 文档创建 repository 或导入已有 repository。

GitLab | Projects | Groups | More

Search or jump to...

test6.3

Project overview

- Details
- Activity
- Releases
- Repository**
- Issues (0)
- Merge Requests (0)
- CI / CD
- Operations
- Packages & Registries
- Analytics
- Wiki
- Snippets

Repository

The Auto DevOps pipeline has been enabled and will be used if no alternative CI configuration file is found. Container registry is not enabled on this GitLab instance. Ask an administrator to enable it in order for Auto DevOps to work.

[Settings](#) [More information](#)

caio chuan > test6.3 > Details

Project 'test6.3' was successfully created.

test6.3 Project ID: 9

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

master | test6.3 | +

History | Find file | Web IDE | Clone

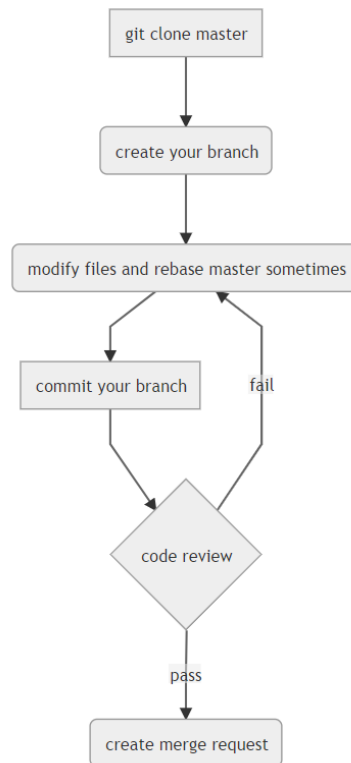
Initial commit
caio chuan authored just now

1fc97d34

[README](#) [Auto DevOps enabled](#) [Add LICENSE](#) [Add CHANGELOG](#) [Add CONTRIBUTING](#) [Add Kubernetes cluster](#)

Name	Last commit	Last update
README.md	Initial commit	just now

2. CI 开发 操作流程图



2.0 准备工作

注意保证本机上已经安装好了 git，如果没有安装，linux 下可以直接用

`sudo apt-get install git`

指令安装，windows 下可以参考 git 官网 git 选择对应操作系统的 git 安装。

安装 git 后，设置本机的信息，包括用户名与邮箱，使用以下语句（只需设置一次）：

```
git config --global user.name "用户名"
```

```
git config --global user.email "邮箱@example.com"
```

2.1 从主分支拉取当前 project

首先进入项目所需要存放的本机目录，如 `/home/nano/gitlab_projects` 下，运行以下指令：

```
cd /home/nano/gitlab_projects
```

```
git clone http://202.120.40.153:12346/chuan.cao/test6.3.git
```

```
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
nano@ubuntu:~$ git clone http://202.120.40.153:12346/chuan.cao/test6.3.git
正克隆到 'test6.3'...
Username for 'http://202.120.40.153:12346': chuan.cao
```

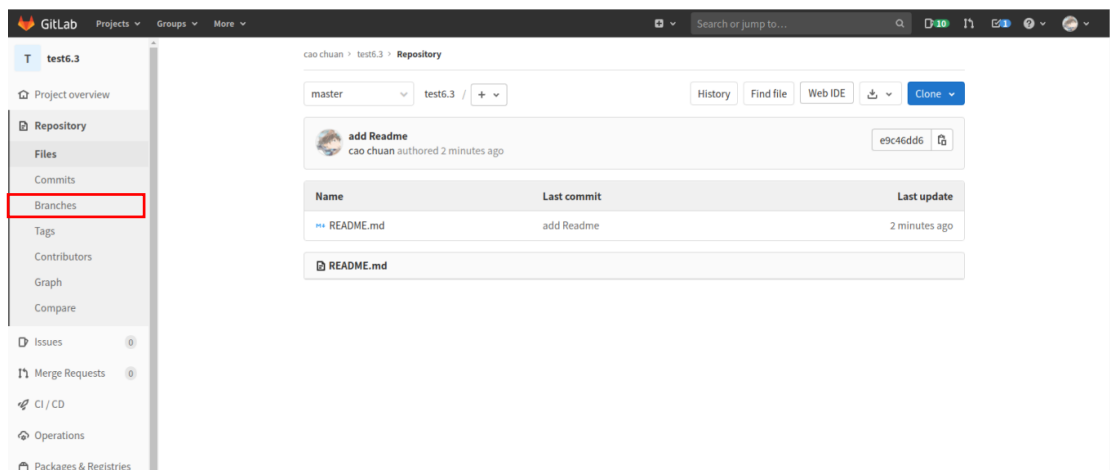
```
文件(F) 编辑(E) 查看(V) 搜索(S) 终端(T) 帮助(H)
nano@ubuntu:~$ git clone http://202.120.40.153:12346/chuan.cao/test6.3.git
正克隆到 'test6.3'...
Username for 'http://202.120.40.153:12346': chuan.cao
Password for 'http://chuan.cao@202.120.40.153:12346':
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
展开对象中: 100% (3/3), 完成。
nano@ubuntu:~$
```

运行后会提示输入 gitlab 的用户名和密码，按照登陆 gitlab 的用户名和密码输入后会自动下载 test6.3 文件夹，运行

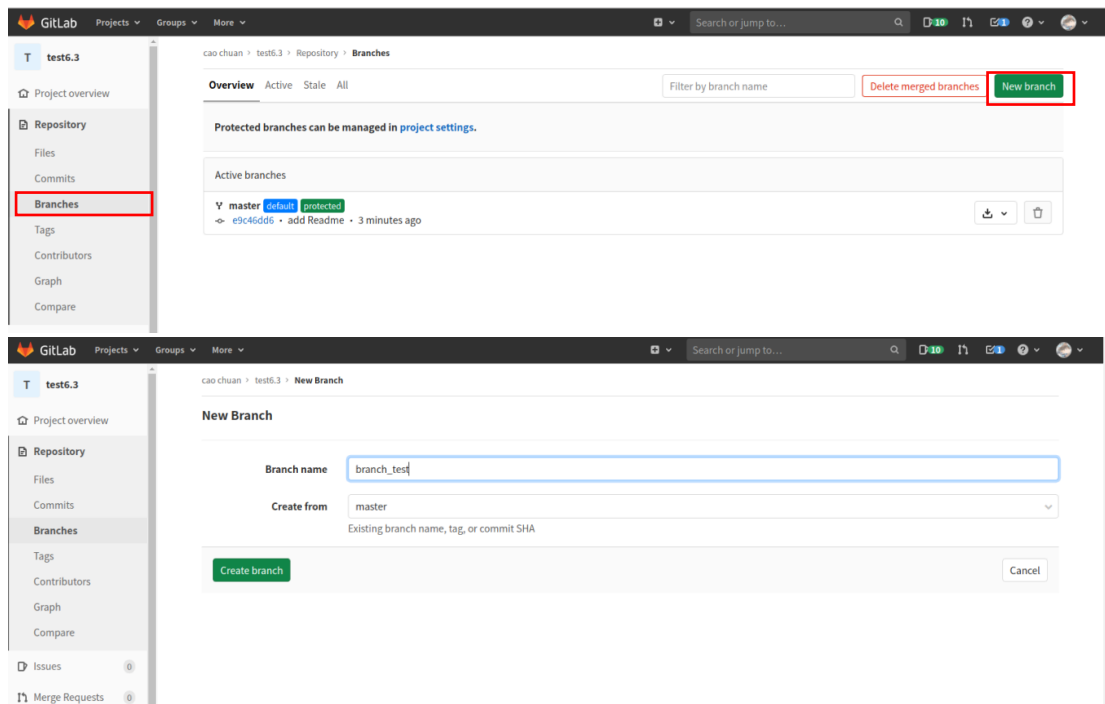
cd test6.3
后进入项目。

2.2 在远程仓库新建 branch

可以在项目左侧 Repository->branches->New branch 新建分支，输入新的分支名称

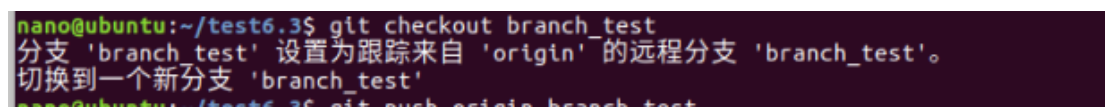


YOUR_BRANCH_NAME, 选择从 master 分支创建。



2.3 在本地切换分支并上传

git checkout YOUR_BRANCH_NAME,



仓库修改文件，如添加文件夹后



然后运行

git add .

git commit -m "add gongchuang"

```

nano@ubuntu:~/test6.3$ git add .
nano@ubuntu:~/test6.3$ git commit -m "add gongchuang"
[branch_test eaa051a] add gongchuang
12 files changed, 2347 insertions(+)
create mode 100644 ekf_unicycle/CMakeLists.txt
create mode 100644 ekf_unicycle/models/unicycle/model.config
create mode 100644 ekf_unicycle/models/unicycle/model.sdf
create mode 100644 ekf_unicycle/package.xml
create mode 100644 ekf_unicycle/scripts/.scan.py.swp
create mode 100755 ekf_unicycle/scripts/control_signal_transfer.py
create mode 100755 ekf_unicycle/scripts/controller_lyh_zyx.py
create mode 100755 ekf_unicycle/scripts/ekf_slam_lyh_zyx.py
create mode 100755 ekf_unicycle/scripts/meas_publisher.py
create mode 100644 ekf_unicycle/scripts/meas_publisher.pyc
create mode 100644 ekf_unicycle/world/4cylinder.world
create mode 100644 ekf_unicycle/world/4cylinder_new.world

```

提示信息中会有更改的内容，然后将其上传到新建分支中

git push origin YOUR_BRANCH_NAME

输入用户名和密码，上传完成后可以在远端仓库看到修改的内容。

```

nano@ubuntu:~/test6.3$ git push origin branch_test
Username for 'http://202.120.40.153:12346': chuan.cao
Password for 'http://chuan.cao@202.120.40.153:12346':
对象计数中: 19, 完成.
Delta compression using up to 8 threads.
压缩对象中: 100% (17/17), 完成.
写入对象中: 100% (19/19), 17.94 KiB | 5.98 MiB/s, 完成.
Total 19 (delta 1), reused 0 (delta 0)
remote:
remote: To create a merge request for branch_test, visit:
remote:   http://202.120.40.153:12346/chuan.cao/test6.3/-/merge_requests/new?merge_request%5Bsource_branch%5D=branch_test
remote:
To http://202.120.40.153:12346/chuan.cao/test6.3.git
d8dd056..eaa051a  branch_test -> branch_test
nano@ubuntu:~/test6.3$

```

You pushed to **branch_test** 15 hours ago

Create merge request

branch_test

test6.3 / +

History

Find file

Web IDE

Clone



add gongchuang

cao chuan authored 3 minutes ago

eaa051ae



Name	Last commit	Last update
ekf_unicycle	add gongchuang	3 minutes ago

2.4 创建 merge request

You pushed to **branch_test** 15 hours ago

Create merge request

branch_test test6.3 / +

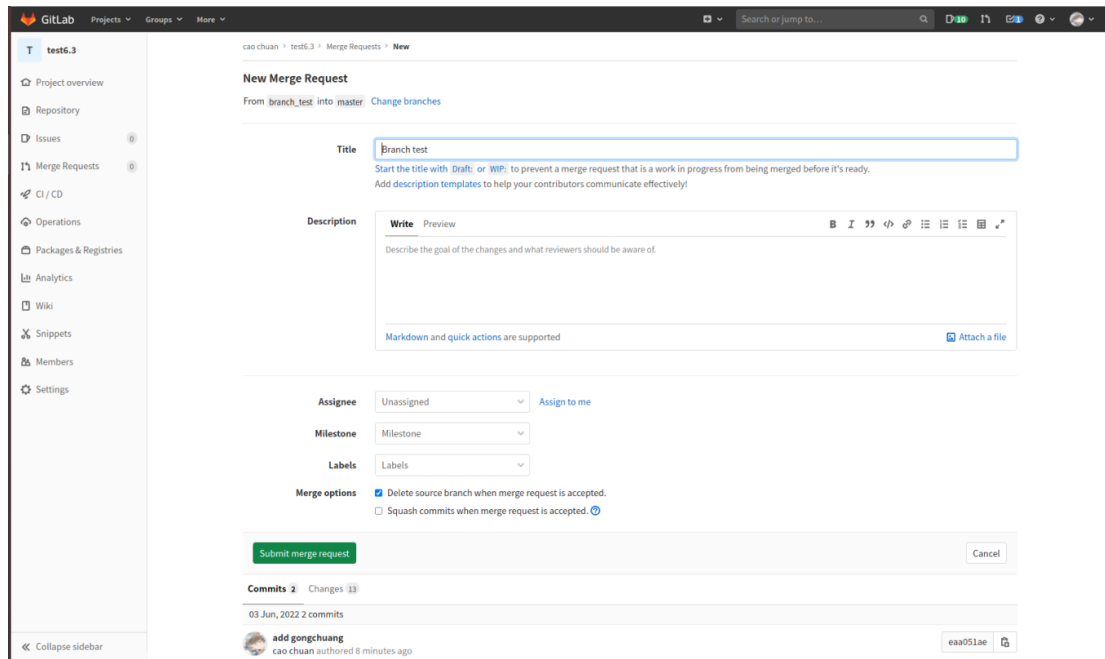
History Find file Web IDE Clone

add gongchuang
cao chuan authored 3 minutes ago

eaa051ae

Name	Last commit	Last update
ekf_unicycle	add gongchuang	3 minutes ago

点击上方按钮或是左侧 merge request -> new merge request 按钮，



分别填入 merge request 的名称，描述，标签，处理人等信息，如果想要 merge 后暂时保留分支，可以取消勾选 merge options 第一行

master 中的 yml 文件会自动检测分支变化并给出代码检测报告，包括 bug，smell 和 unittest 覆盖率等，可以在 SonarQube:202.120.40.153:9001 中查看项目的报告。检测通过后可以在左侧 Merge Requests 中点击 New merge request，选择将 YOUR_BRANCH_NAME 合并到 master 中，等待大家的审阅，投票通过后 merge 到 master 中。

3. CI-yml 文件配置

CI 配置文件.gitlab-ci.yml 只有在根目录下才能生效，请不要修改根目录下的 yml 文件和 property 文件以免 CI 无法正常工作。

yml 文件中规定了不同的工作阶段，本项目中分为代码分析，环境构建以及测试三个阶段，可以给创建的任务指定阶段，如代码风格审查任务，可以指定为代码分析阶段，任务中需要指定运行的镜像作为基础环境，在 script 中是具体的执行代码，如代码风格审查任务需要使用 sonarqube 服务，就在 script 中添加该服务。另外，在 only/except 等选项中可以设置该任务的触发条件。也可以使用 rules 来代替 only/except 选项触发检查。

代码质量规则

python

C++