



## Infraestrutura II

# Passo a passo: Criar nossa conta no GitLab

Vamos criar nossa conta no GitLab, que será a ferramenta utilizada para implementar nossas Pipelines.

## Objetivo final da prática

- Conclua o cadastro na instância do GitLab e deixar nosso usuário pronto para poder fazer upload de código para os repositórios.
- Carregue um código de teste.

## Formas de consumir o GitLab

Primeiramente devemos saber que o GitLab pode ser consumido de várias maneiras:

- **Usando o SAAS público:** acessando <https://gitlab.com/>, registrando-se lá podemos começar a usar todos os serviços em poucos segundos, com o único inconveniente que para poder executar os Pipelines, devemos validar nossa conta com um Cartão de Crédito, apesar de não ter nenhum custo.
- **Instalando-o em nossos servidores:** Seguindo uma série de etapas dedicadas às diferentes distribuições Linux, poderíamos ter nosso próprio GitLab instalado, tendo a possibilidade de executá-lo em contêineres Docker.

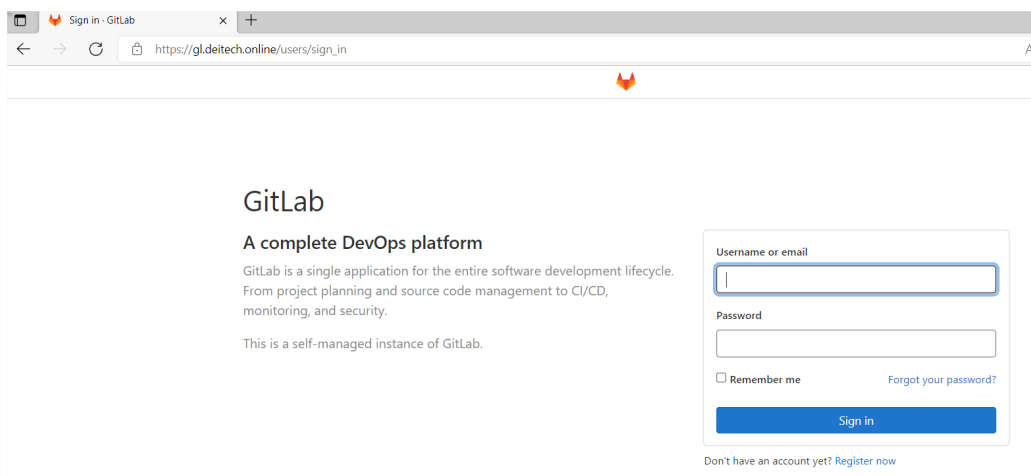
Esta última opção é a mais recomendada e utilizada no local de trabalho, agora, como exceção, exige hardware potente para fazê-lo, pois o hardware mínimo para que funcione de maneira ideal é de 4 CPUs e 8Gb de RAM.

Nesta situação, vamos fornecer a você uma instância do GitLab para as práticas de Infraestrutura II, que está hospedada em:

<https://gitlab.ctd.academy/>

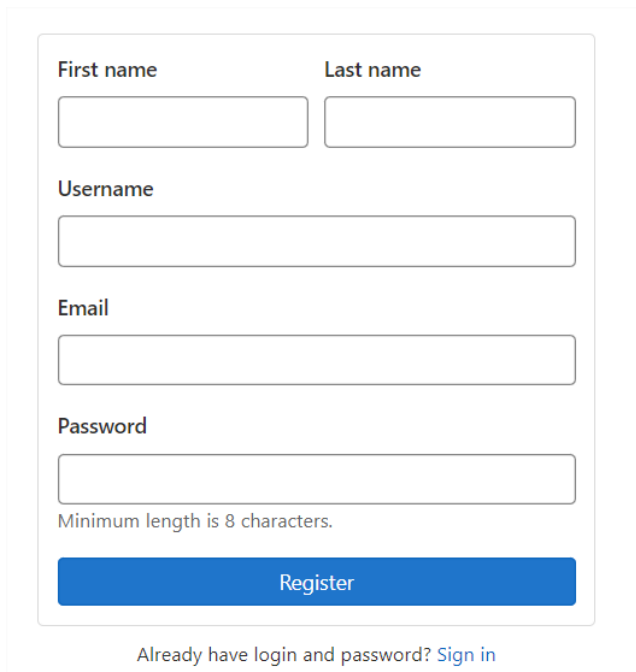
## Cadastro e preparação do ambiente

1. Na tela inicial, vamos em “ Register now ”



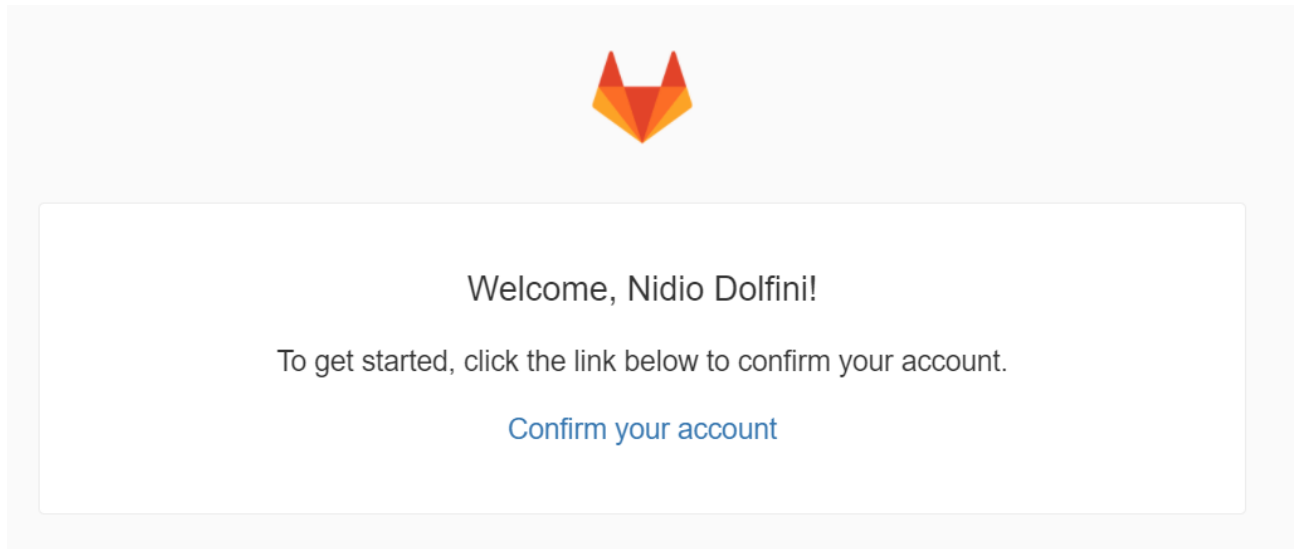
The screenshot shows the GitLab sign-in page in a web browser. The browser's address bar displays the URL [https://gl.deitech.online/users/sign\\_in](https://gl.deitech.online/users/sign_in). The page features the GitLab logo and the text "A complete DevOps platform". Below this, it states: "GitLab is a single application for the entire software development lifecycle. From project planning and source code management to CI/CD, monitoring, and security. This is a self-managed instance of GitLab." On the right side, there is a sign-in form with fields for "Username or email" and "Password". Below these fields are checkboxes for "Remember me" and a link for "Forgot your password?". A blue "Sign in" button is at the bottom of the form. At the very bottom of the page, there is a link that says "Don't have an account yet? Register now".

2. Os dados que devemos preencher são os seguintes, ele nos informará automaticamente se o nome de usuário ou e-mail selecionado está em uso.



The registration form consists of several input fields arranged vertically. At the top, there are two side-by-side fields labeled "First name" and "Last name". Below these is a single field labeled "Username". This is followed by a field labeled "Email". Then comes a field labeled "Password", with a note below it stating "Minimum length is 8 characters." At the bottom of the form is a large blue button labeled "Register". Below the button, there is a link that says "Already have login and password? Sign in".

3. Receberemos uma confirmação e somos informados de que devemos verificar nosso e-mail; Verifique corretamente as pastas Caixa de entrada e Spam, o endereço de onde o e-mail é enviado é [gitlab@ctd.academy](mailto:gitlab@ctd.academy)



4. Confirmamos nossa conta, com a qual agora podemos fazer login, fazendo isso nos perguntará qual papel desempenhamos , selecionamos “Devps Enginner” e como motivo “Quero aprender o básico do Git”. Isso personaliza nossa experiência com a plataforma.
5. Na tela inicial de boas vindas selecionamos “Create Project”, depois “Create blank Project” e por fim, completamos os dados do “Project Name”; deixamos a visibilidade disso em “Privado”.



3)

New project > Create blank project

Project name

infra2

Project URL

https://gl.deitech.on... nidio

Project slug

infra2

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

Project description (optional)

Description format

Project deployment target (optional)

Select the deployment target

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 except external users.

☐ Public  
The project can be accessed without any authentication.

Project Configuration

☒ 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

6. Por padrão é criado um repositório Git no qual podemos fazer upload de nossos arquivos, incluindo um README.md com instruções básicas para esse fim. Embora como em qualquer repositório Git possamos fazer upload de nossos arquivos através da interface Web, não é seu uso ideal, então agora vamos preparar nosso ambiente para poder fazer upload de código que estão no nosso computador; Serão necessários alguns passos nos quais devemos usar nossa linha de comando / console

6.1. Abrimos um Shell em nosso computador, como o GitBash ou o Terminal Linux, com o objetivo de gerar uma chave SSH, para isso executamos:

```
ssh-keygen -t rsa
```

Ele nos pedirá um nome e uma senha (opcional), deixamos tudo por padrão.

Lemos seu conteúdo com o comando:

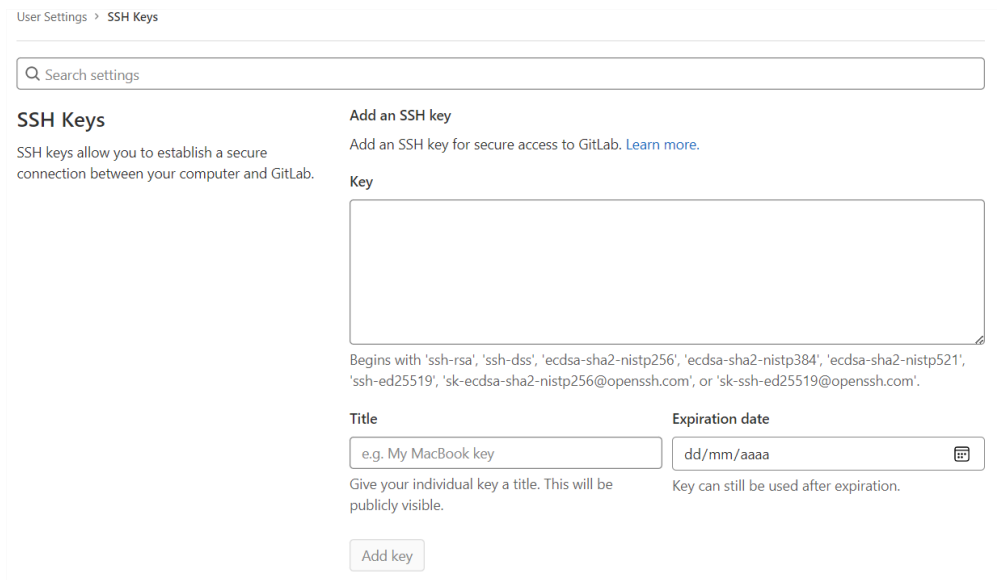
```
cat ~/.ssh/id_rsa.pub
```

nos será mostrado um conteúdo como

```
nidio@jarvis:~$ cat ~/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQgQC4gCFiUGLZs7n2mB7gIMCmFsTLykgUOGAuLjsP6CDCHNF695MyDr6YKl4
a7u0WMOiZnm3XFHNTdIK+8/Mh3RYaj6z4YGAERG2vP+ol10tFks5DNM5mCJgBuiknGy9Muz+80jW3q2s9PW0HIjc0wHwQHP
cWljBOBIV6QhKjaldFTXfg8NpPS5hjwCIB2uZfcHwd6735wVz8GzbMzf6W/cUmmIZbQxUyXn1E4KGyospw/kfnmFWxOIsHL
8W5QZ38nqIhmTFTU6JF384h6bT7kEamtYzXkz06hhQScBd+QrKaEybx0LSjEtn/FrxwMeHmzSwbHfdHxQkp2xCpchiY/bqu
lIh0jy4QrHxJ5+Px1ekY8/k/po28tugAChTC6b3yWvukkkUaIWIALwenGavTBA/dm+3Jiv69kfmEPeJdvn/ti7kmLt4cn6b
Qp6QQA9tLsQvYQz9/iXQGJ9IJEH7YCFXeyI7oIfmnJwEqfHwL7C5fFq3brma3u69wtC0dxD408TU= nidio@jarvis
```

Não fechamos nosso Shell.

- 6.2. Vamos ao Gitlab, na seção "SSH Keys" ([Chaves SSH • Configurações do usuário • GitLab](#))
- 6.3. todo o conteúdo que foi mostrado no ponto 6.1 com o comando cat (desde o início com ssh-rsa incluído) e colamos em na seção "Key", damos um nome em "Title" e, opcionalmente, uma data de expiração, depois em "Add key".



User Settings > SSH Keys

Search settings

### SSH Keys

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

**Add an SSH key**  
Add an SSH key for secure access to GitLab. [Learn more.](#)

**Key**

Begins with 'ssh-rsa', 'ssh-dss', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', 'ecdsa-sha2-nistp521', 'ssh-ed25519', 'sk-ecdsa-sha2-nistp256@openssh.com', or 'sk-ssh-ed25519@openssh.com'.

**Title**  
e.g. My MacBook key  
Give your individual key a title. This will be publicly visible.

**Expiration date**  
dd/mm/aaaa  
Key can still be used after expiration.

**Add key**

- 6.4. Uma vez salvo no GitLab, devemos verificar se podemos nos conectar ao nosso ambiente, para isso retorne ao nosso Shell e execute

```
ssh -i .ssh/chave-ssh -T git@gitlab.ctd.academy
```

Se for bem sucedido, devemos receber a mensagem "Welcome to GitLab, @MY USERNAME!"

Com isso podemos enviar nosso código para o repositório do GitLab

## . first code

Há um exemplo, vamos fazer o upload de um código fonte feito em Java, com toda a estrutura padrão que o Maven constrói para o projeto, pode ser baixado aqui [dhinfra](#)

1. Uma vez baixado e descompactado, nós vá para a pasta "dhinfra" em nosso Shell. Uma vez lá, inicializamos o repositório com o

```
git init
```

2. Adicionamos a fonte remota com o comando, lembrando que ao invés de **"seuuser"** deverá colocar seu nome de usuário e no lugar de "infra2.git" irá o nome do seu projeto.

```
git remote add origin git@gitlab.ctd.academy:seuuser/infra2.git
```

3. Vamos adicionar os arquivos do projeto no stage para commit com o comando:

```
git add .
```

4. agora vamos fazer o commit:

```
git commit -m "Primeiro Commit"
```

5. Mudamos para o branch de teste (IMPORTANTE: por padrão o GitLab protege o branch principal, por isso usamos um branch de teste)

```
git branch -M test
```

Finalmente enviamos nosso código

```
git push -uf origin test
```

6. Observamos no GitLab, no Branch correspondente, que nosso código já está carregado.



test ▾

infra2 / + ▾

History

Find file

Web IDE

↓ ▾

Clone ▾



**Primeiro Commit**

Nidio Dolfini authored 1 minute ago

44a8b284



Name	Last commit	Last update
src	Primeiro Commit	1 minute ago
pom.xml	Primeiro Commit	1 minute ago