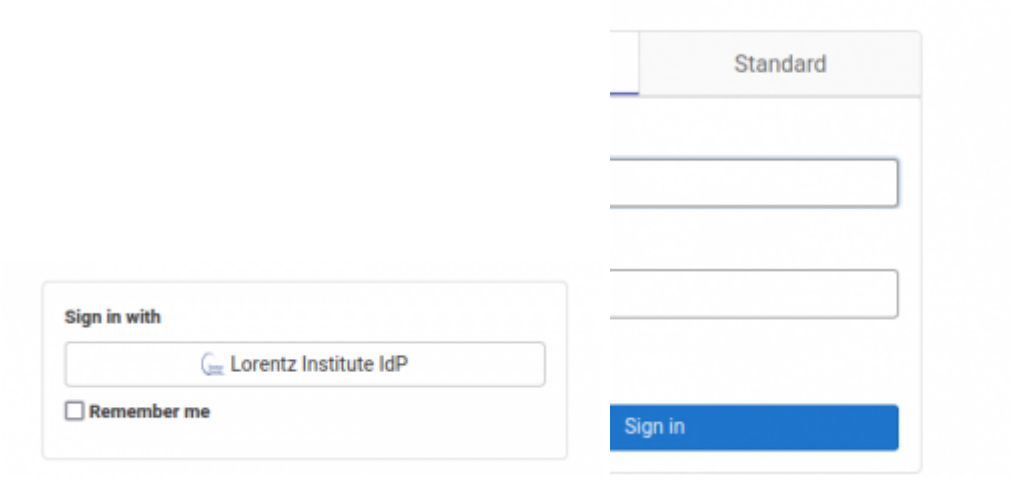
Gitlab Notes

IMPORTANT	Gitlab logins via the Lorentz Institute IdP might assign you a gitlab username different than expected. Please check https://gitlab.lorentz.leidenuniv.nl/-/profile/account and do **not** change it ssh -T git@gitlab.lorentz.leidenuniv.nl will also return your gitlab username

Lorentz Institute runs a private instance of Gitlab accessible to all institute members at https://gitlab.lorentz.leidenuniv.nl regardless of their role within the Institute.

You are strongly encouraged to use it to collaborate with your colleagues and fellow scientists anywhere in the world. Accounts for external users can be requested via support@lorentz.institute.nl.

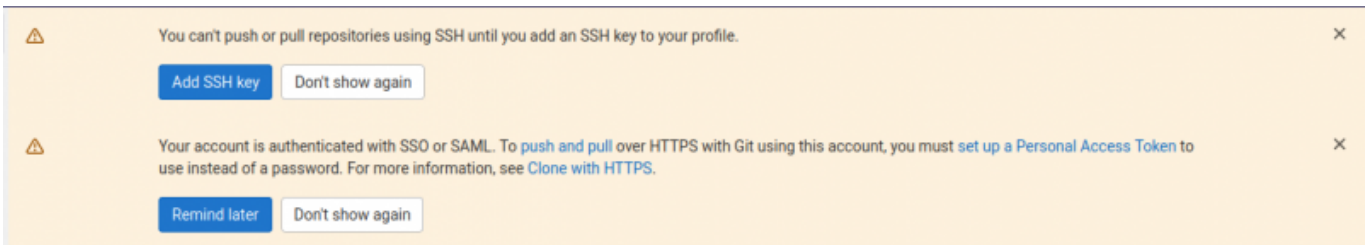
Lorentz Institute members **must use** the button *Sign in with Lorentz Institute IdP* to login via the institute Identity Provider Server, where as external users must use the *Standard* tab in the authentication form.



SSH Keys and Personal Access Tokens

Because access to the Institute Gitlab instance occurs via Single-Sign On[1] for all IL members, git *push* and *pull* operations must be performed either using SSH key-pairs (SSH protocol) or using Personal Access Tokens (HTTPS protocol). Failing to set up SSH key-pairs or Access Tokens will prevent IL members to interact via git with our Gitlab server.

IL users that do not set up SSH key-pairs or Access Tokens will likely receive upon login a warning message like the one in the figure below

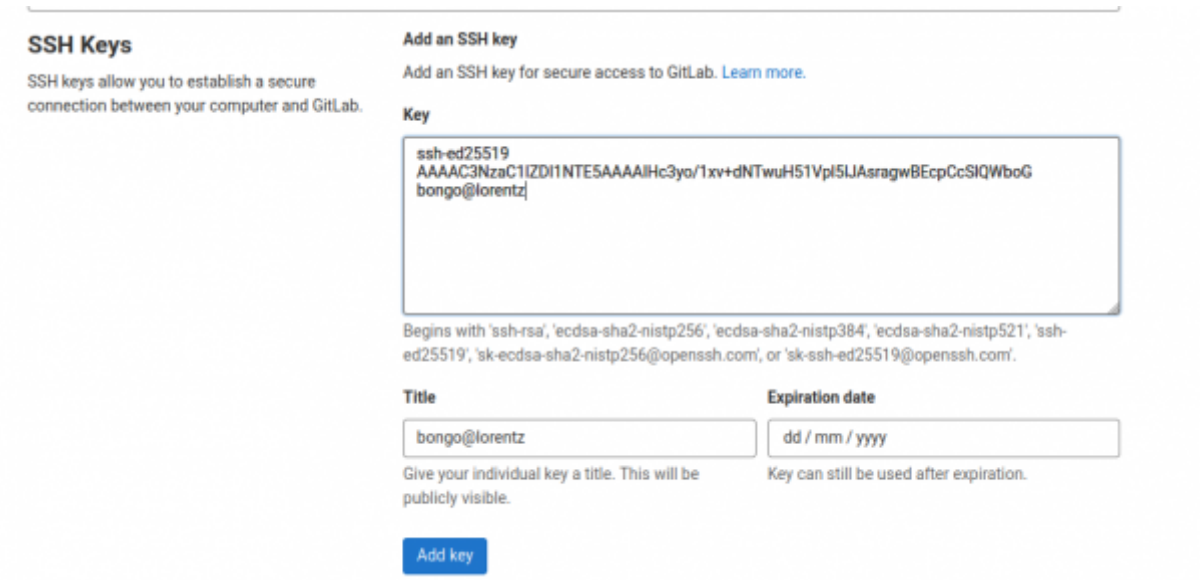


Generate an SSH key-pair and add it to Gitlab

On the workstation/laptop from which you would like to interact with the IL giltab server generate a private/public key pair `$HOME/.ssh/id_ed25519`[2] and `$HOME/.ssh/id_ed25519.pub` using the command

```
ssh-keygen -t ed25519 -C "bongo@gitlab"
```

Visit https://gitlab.lorentz.leidenuniv.nl/-/profile/keys, copy the contents of the newly created `$HOME/.ssh/id_ed25519.pub` and paste it in the *Key* area like in the figure below. Click on *Add key*.



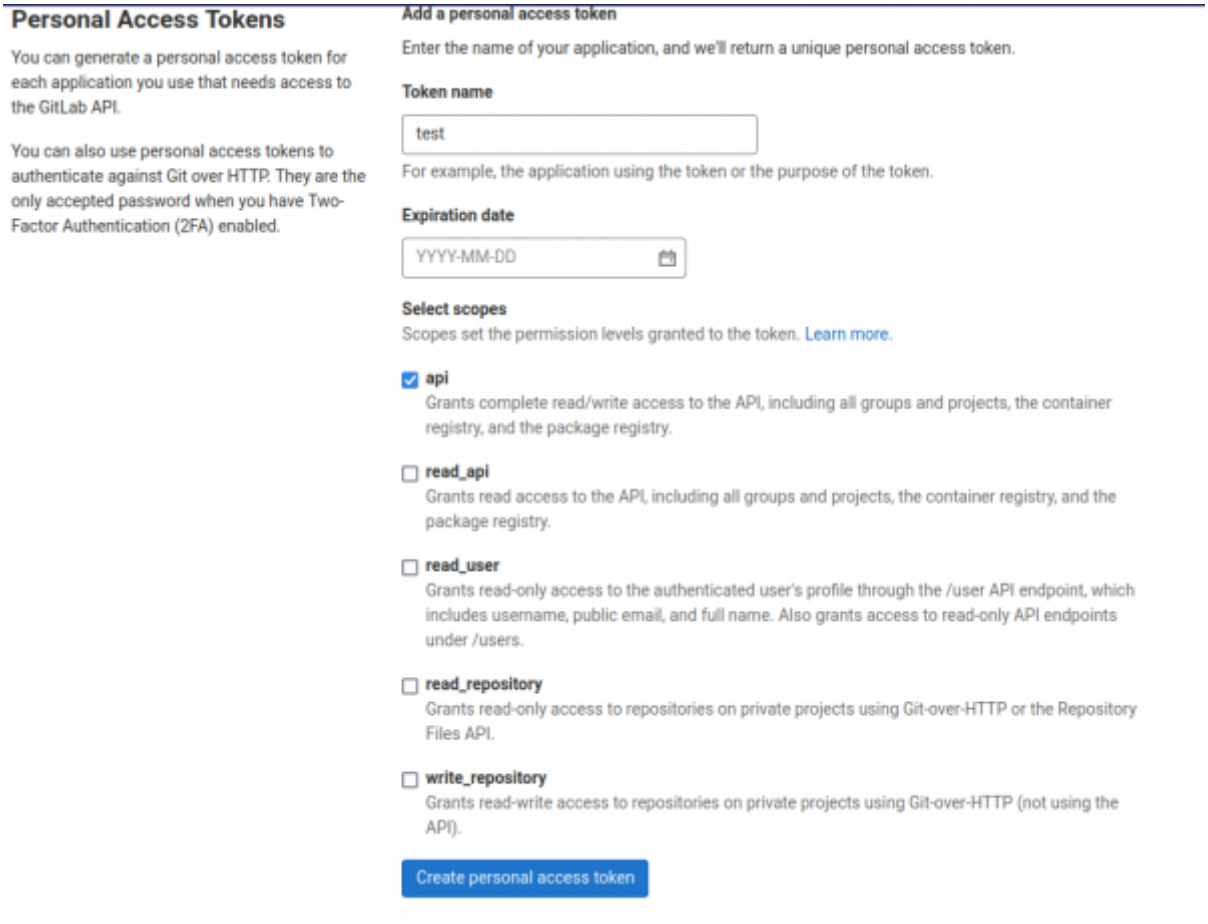
You can non perform git operations over the *SSH* protocol. Test that your set up is correct

```
ssh  -T git@gitlab.lorentz.leidenuniv.nl
Welcome to GitLab, @bongo1!
```

NOTE	In some exceptional cases[3] your gitlab username might differ from your Institute username. This is normal and no action is required on your side. Nonetheless, you are encouraged to take note of it because you will need it for all git operations.

Generate a Gitlab Personal Access Token

Visit https://gitlab.lorentz.leidenuniv.nl/-/profile/personal_access_tokens, choose a token name, assign the necessary scopes[4], and click on *Create Personal Access Token*.



Your personal access token will look something like `xyzop-ybVxByblaUBLA7RUAGAs` (this is a fake one). Please take note of it and store it in a very safe place. Disclosing this token is equivalent to disclosing your credentials! Should a token be compromised at any time, just visit https://gitlab.lorentz.leidenuniv.nl/-/profile/personal_access_tokens to revoke any compromised tokens.

Your IL account is now enabled to perform git operations over HTTPS.

Examples

In the examples that follow it is shown how you can synchronise for the first time your local git repo to Gitlab and viceversa. The examples below assume you are typing all commands in a BASH GNU/Linux terminal and that you are familiar with the basics of git by reading carefully the following documents

- https://gitlab.lorentz.leidenuniv.nl/help/gitlab-basics/start-using-git
- https://git-scm.com/docs/user-manual

Local repo to gitlab

```
# create test repo
mkdir test
cd test/
git init .
echo "Hello World" >> README.md
```

```
git add *
git commit . -m "Test"
git push --set-upstream https://gitlab.lorentz.leidenuniv.nl/bongo1/test.git
master
Username for 'https://gitlab.lorentz.leidenuniv.nl': bongo1
Password for 'https://bongo1@gitlab.lorentz.leidenuniv.nl': <Personal Access
Token>

Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 231 bytes | 115.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
remote:
remote:
remote: The private project bongo1/test was successfully created.
remote:
remote: To configure the remote, run:
remote:   git remote add origin
https://gitlab.strw.leidenuniv.nl/bongo1/test.git
remote:
remote: To view the project, visit:
remote:   https://gitlab.strw.leidenuniv.nl/bongo1/test
remote:
remote:
remote:
To https://gitlab.lorentz.leidenuniv.nl/bongo1/test.git
 * [new branch]      master -> master
branch 'master' set up to track
'https://gitlab.lorentz.leidenuniv.nl/bongo1/test.git/master'.
```

Remote repo to local file system

```
git clone https://gitlab.strw.leidenuniv.nl/bongo1/test.git
Cloning into 'test'...
Username for 'https://gitlab.strw.leidenuniv.nl': bongo1
Password for 'https://bongo1@gitlab.strw.leidenuniv.nl': <Personal Access
Token>
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (3/3), done.
```

1)

In other words it uses access and refresh tokens rather than usernames and passwords

2)

This is your private key which must remain secret otherwise your account is compromised.

3)

Think of users with the same private email address or same username but different domain.

4)

Read also as permissions