

# How to access your git repositories remotely

## 1/ A bit of theory.

The webrunner have a specific directory containing your projects. It is located in the working directory "~/.srv/runner", and is named "project" (see [directory documentation](#)). Whenever you need to create a new git repository, you have to create it here (it can be a bare or non-bare repository). This directory should be the central location for all your projects. By default, this directory contains the main "slapos" git repository.

In your working directory (~/.srv/runner), you may find 2 hidden directories, named ".git-public" and ".git-private", which will respectively host weblinks for your public and private projects. Git repositories inside ".git-public" can be cloned anonymously, but authentication is needed to push. Git repositories inside ".git-private" are just for you and for your development team. They can only be cloned by authenticated users. Finally, pushing to any repository can only be done by authenticated users.

## 2/ Prepare the environment.

To make one of your repositories public or private, first create it in the project directory. Then, create a symbolic link to the chosen git folder (".git-public", or ".git-private"). Here is an example, which create a public repository called "myrepo".:

```
# Go to your project folder
$ cd ~/.srv/runner/project
# Prepare your repository
$ git init myrepo
$ cd ~/.srv/runner
# Create a link to your directory in order to make it public
$ ln -s `pwd`/project/myrepo/.git/ `pwd`/.git-public/myrepo.git
```

## 3/ Clone It !

You can now clone your new git repository. First, take note of the URL of the git repositories on the SlapOS instance webpage (number 6 or 7) :

□

To clone the repository, you now need to append its name to this URL. Following the previous example, you need to take the git\_public\_url, and then append "myrepo.git". You can now "git clone" it !! Here:

```
$ git clone https://[2001:67c:1254:9c::875c]:9684/git-public/myrepo.git
```

## 4/ More.

- You can cancel the remote access at any moment by just deleting the symbolic link.
- You can create bare repositories, which are made to be accessed remotely. The advantage of the technique below is that you can deploy a Software Release from the git repository, and develop at the same time.
- Be sure not to push on the current branch of the remote repository. You will get an warning from git.
- In the case of a public repository, you will have to manually enable the option http.receivepack (or pushing will be impossible). To do so, please go to your git repository, and then run

```
$ git config http.receivepack true
```