Two Git Repos: 'git_origin' and 'git_cloned'

Learn about the original, clone, and remote repository.

We'll cover the following

- Creating a repository and cloning it
- Similarities and differences
- Remote
  - Origin
- Fetch and push

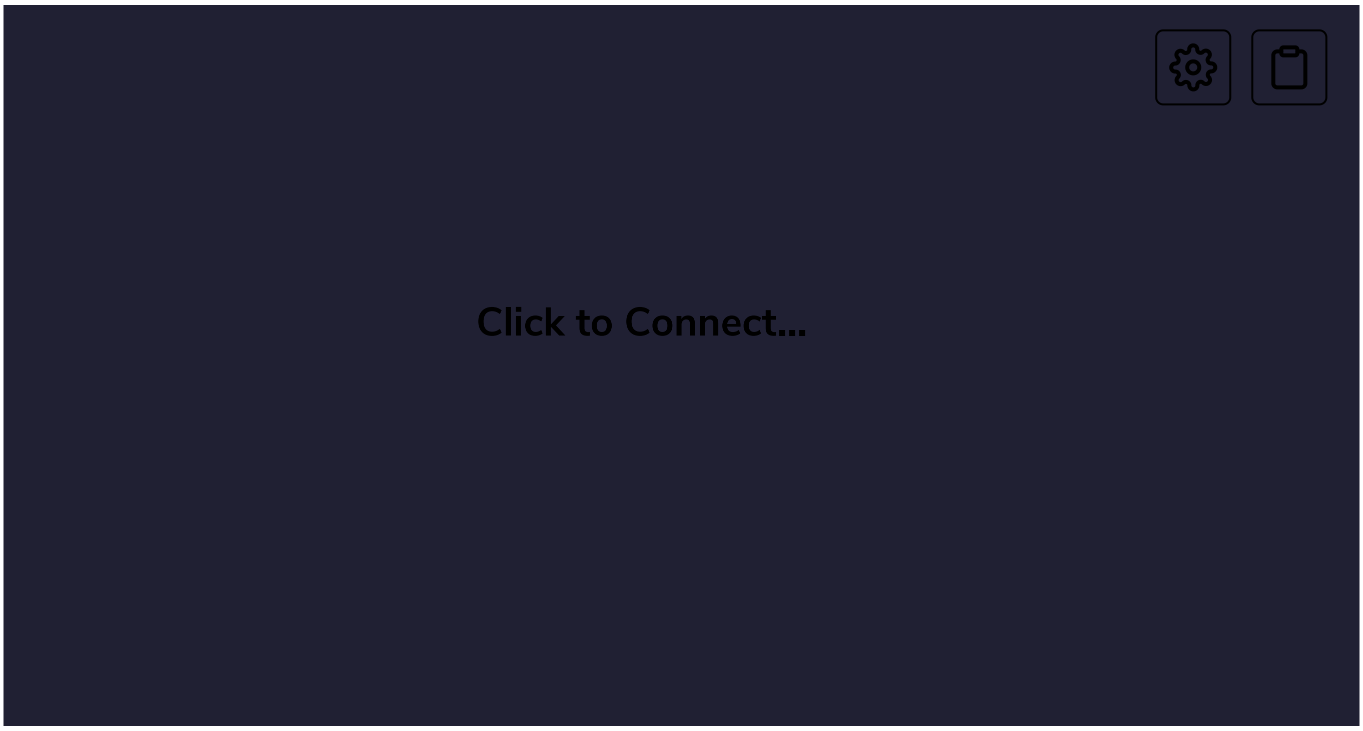
Creating a repository and cloning it

You're going to start by creating a simple Git repository and then clone it:

```
mkdir -p lgthw_pull
cd lgthw_pull
mkdir git_origin
cd git_origin
git init
echo 'first commit' > file1
git add file1
git commit -am file1
cd ..
git clone git_origin git_cloned
```

Terminal 1

Terminal



Similarities and differences

The two repositories (the folders: `git_origin` and `git_cloned`) contain identical content:

```
diff git_origin/file1 git_cloned/file1
```

However, their `.git/config` files differ in instructive ways.

The `git_origin` folder has this in its `.git/config` file:

```
cat git_origin/.git/config
```

While the `git_cloned` folder has this in its `.git/config` file:

```
cat git_cloned/.git/config
```

Terminal 1

Terminal

Remote

While the `git_origin` has no visibility of any *remotes*, the cloned one does. The `git_cloned` repository has a remote called “origin.”

Its URL is (in this case) pointed at the directory `git_origin` that is local on the host. URLs can also be an `http://` or `https://` one or even `ssh://` or `git://`. The last is a Git-specific protocol that is rarely seen these days. When you see a Git repository cloned with `git@` (i.e., `git@github.com:kubernetes/kubernetes`), then this is in fact using the `ssh://` protocol under the hood.

If you go to the `git_cloned` repository and ask it for information about its remotes:

```
cd git_cloned
git remote
```

You get the name `origin` back.

Terminal 1

Terminal

Origin

The name `origin` is the default name for a remote, but it does not have a special meaning. It could be renamed to `BitBucket` or `GitLab`, for example.

Fetch and push

The above remotes are divided into `(fetch)` and `(push)` actions. These relate to two different actions on remotes, i.e., getting changes from a remote or pushing changes to a remote.

These actions can work against different remotes. For example, the output of `git remote -v` might be:

```
git remote -v
```

Terminal 1

Terminal

But I've never seen an example of these entries being different from one another in the wild.

Back
Introduction: Fetching and Pulling Con...

Next
The 'git fetch' Command

✓ Mark as Completed

Report an Issue

Ask a Question
(https://discuss.educative.io/tag/two-git-repos-git_origin-and-git_cloned__fetching-and-pulling-content__learn-git-the-hard-way)