Publishing Your Project

coursera.org/learn/git-distributed-development/supplement/BVjUO/publishing-your-project

sSuppose you want to make your project available to others, either on your local machine or on the network, for cloning, pushing, pulling, etc. First, you should create a bare version of your project, as in:

```
1
$ git clone --bare git-test /tmp/git-test
```

where you see that **/tmp/git-test** now contains a copy of your project's repository, but none of the working files themselves.

A local user can easily make a new cloned copy with the same command without the **--bare** option, as in:

```
1
$ git clone /tmp/git-test my-git
```

But what about network users?

To make a copy available using the native git protocol, you will need to have the git daemon service installed. To make your bare repository accessible, you have to create an empty file in its main directory or in the **.git** subdirectory:

```
1
$ touch /tmp/git-test/git-daemon-export-ok
```

You can configure **git daemon** to run automatically by configuring either **xinetd** or **inetd** on your system, and in doing so, you can control behavior rather precisely, but to keep things simple you can just simply run the daemon in background:

```
1

$ git daemon &
```

Note that you do not have to run this as superuser. Then, a remote user can simply clone your repository with:

```
1

$ git clone 192.168.1.100:/tmp/git-test my-git
```

This gives them the ability only to clone and fetch, not to push changes back. To give everyone the ability to push changes back, you can do:

```
1

$ git daemon --enable=receive-pack
```

but this should only be done in a very friendly environment. Additionally, this will be done for all git repositories on the system. To enable write access for just one repository, you have to instead add to the **config** file in the repository the lines:

```
1

2
```

```
[daemon]

      receivepack = true
```

Note there is no **-** in **receivepack**.

To enable access through the http protocol, you will have to have a proper web server installed and running (probably apache) and know a little about how to configure it. You can place your project directory either under **/var/www/html**, or in the unprivileged place **/home/username/public_html** (in which case, your server has be configured to allow such access).

Before access is available, you have to go the project directory and run the command:

```
1

$ git --bare update-server-info
```

Accessing through the web server can then be done through these commands:

```
1

2

$ git clone https://192.168.1.100/git-test my-git

$ git clone https://192.168.1.100/~username/git-test
```

where you have to substitute the right IP address or domain, and the right username.

Suppose you want to make your project available to someone not using git, or you want to store archived material of your current working tree without including the **.git** repository information directories.

This is easy to do with **git archive** as in:

```
1
```

```
$ git archive --verbose HEAD | bzip2 > myproject.tar.bz2
```

If you want to create an archive corresponding to a particular point in time rather than the latest state, say to tag v1.7.1, you could do:

```
1

$ git archive --verbose v1.7.1 | bzip2 > myproject_v1.7.1.tar.bz2
```