

4.5 Git on the Server - Git Daemon

Git Daemon

Next we'll set up a daemon serving repositories using the "Git" protocol. This is a common choice for fast, unauthenticated access to your Git data. Remember that since this is not an authenticated service, anything you serve over this protocol is public within its network.

If you're running this on a server outside your firewall, it should be used only for projects that are publicly visible to the world. If the server you're running it on is inside your firewall, you might use it for projects that a large number of people or computers (continuous integration or build servers) have read-only access to, when you don't want to have to add an SSH key for each.

In any case, the Git protocol is relatively easy to set up. Basically, you need to run this command in a daemonized manner:

```
$ git daemon --reuseaddr --base-path=/srv/git/ /srv/git/
```

The `--reuseaddr` option allows the server to restart without waiting for old connections to time out, while the `--base-path` option allows people to clone projects without specifying the entire path, and the path at the end tells the Git daemon where to look for repositories to export. If you're running a firewall, you'll also need to punch a hole in it at port 9418 on the box you're setting this up on.

You can daemonize this process a number of ways, depending on the operating system you're running.

Since `systemd` is the most common init system among modern Linux distributions, you can use it for that purpose. Simply place a file in `/etc/systemd/system/git-daemon.service` with these contents:

```
[Unit]
Description=Start Git Daemon

[Service]
ExecStart=/usr/bin/git daemon --reuseaddr --base-path=/srv/git/ /srv/git/

Restart=always
RestartSec=500ms

StandardOutput=syslog
StandardError=syslog
SyslogIdentifier=git-daemon

User=git
Group=git

[Install]
WantedBy=multi-user.target
```

You might have noticed that Git daemon is started here with `git` as both group and user. Modify it to fit your needs and make sure the provided user exists on the system. Also, check that the Git binary is indeed located at `/usr/bin/git` and change the path if necessary.

Finally, you'll run `systemctl enable git-daemon` to automatically start the service on boot, and can start and stop the service with, respectively, `systemctl start git-daemon` and `systemctl stop git-daemon`.

On other systems, you may want to use `xinetd`, a script in your `sysvinit` system, or something else — as long as you get that command daemonized and watched somehow.

Next, you have to tell Git which repositories to allow unauthenticated Git server-based access to. You can do this in each repository by creating a file named `git-daemon-export-ok`.

```
$ cd /path/to/project.git  
$ touch git-daemon-export-ok
```

The presence of that file tells Git that it's OK to serve this project without authentication.

[prev](#) | [next](#)