NAME

git-shell − Restricted login shell for Git−only SSH access

SYNOPSIS

chsh −s $(command −v git−shell) <user>
git clone <user>**@localhost:/path/to/repo.git**
ssh <user>**@localhost**

DESCRIPTION

This is a login shell for SSH accounts to provide restricted Git access. It permits execution only of server−side Git commands implementing the pull/push functionality, plus custom commands present in a subdirectory named **git−shell−commands** in the user's home directory.

COMMANDS

git shell accepts the following commands after the **−c** option:

git receive−pack <argument>, *git upload−pack <argument>*, *git upload−archive <argument>*
: Call the corresponding server−side command to support the client's *git push*, *git fetch*, or *git archive −−remote* request.

cvs server
: Imitate a CVS server. See **git-cvsserver**(1).

If a **~/git−shell−commands** directory is present, *git shell* will also handle other, custom commands by running "**git−shell−commands/<command> <arguments>**" from the user's home directory.

INTERACTIVE USE

By default, the commands above can be executed only with the **−c** option; the shell is not interactive.

If a **~/git−shell−commands** directory is present, *git shell* can also be run interactively (with no arguments). If a **help** command is present in the **git−shell−commands** directory, it is run to provide the user with an overview of allowed actions. Then a "git> " prompt is presented at which one can enter any of the commands from the **git−shell−commands** directory, or **exit** to close the connection.

Generally this mode is used as an administrative interface to allow users to list repositories they have access to, create, delete, or rename repositories, or change repository descriptions and permissions.

If a **no−interactive−login** command exists, then it is run and the interactive shell is aborted.

EXAMPLES

To disable interactive logins, displaying a greeting instead:

```
$ chsh −s /usr/bin/git−shell
$ mkdir $HOME/git−shell−commands
$ cat >$HOME/git−shell−commands/no−interactive−login <<\EOF
#!/bin/sh
printf '%s\n' "Hi $USER! You've successfully authenticated, but I do not"
printf '%s\n' "provide interactive shell access."
exit 128
EOF
$ chmod +x $HOME/git−shell−commands/no−interactive−login
```

To enable git−cvsserver access (which should generally have the **no−interactive−login** example above as a prerequisite, as creating the git−shell−commands directory allows interactive logins):

```
$ cat >$HOME/git−shell−commands/cvs <<\EOF
if ! test $# = 1 && test "$1" = "server"
```

```
then
        echo >&2 "git−cvsserver only handles \"server\""
        exit 1
fi
exec git cvsserver server
EOF
$ chmod +x $HOME/git−shell−commands/cvs
```

SEE ALSO

ssh(1), **git-daemon**(1), contrib/git−shell−commands/README

GIT

Part of the **git**(1) suite