Sshwifty Web SSH & Telnet Client

Sshwifty is a SSH and Telnet connector made for the Web. It can be deployed on your computer or server to provide SSH and Telnet access interface for any compatible (standard) web browser.

Web Interface

Build Status

Install

Binary

Compiled binaries can be found at the release section of the page.

Please be advised that those binaries are generated by an automatic proccess, the author of this project will NOT verify that they work. You will have to try it at your own risk.

Docker Image

If Docker is installed on your machine, you may use our prebuilt Docker Image by executing following command:

```
$ docker run --detach \
  --restart always \
  --publish 8182:8182 \
  --name sshwifty \
  niruix/sshwifty:latest
```

When TLS is desired and you don't want to setup Docker Volumes, you can use `SSHWIFTY_DOCKER_TLSCERT` and `SSHWIFTY_DOCKER_TLSCERTKEY` environment variables to import credential files to the container and automatically apply them:

```
$ openssl req \
  -newkey rsa:4096 -nodes -keyout domain.key -x509 -days 90 -out domain.crt
```

```
$ docker run --detach \
  --restart always \
  --publish 8182:8182 \
  --env SSHWIFTY_DOCKER_TLSCERT="$(cat domain.crt)" \
  --env SSHWIFTY_DOCKER_TLSCERTKEY="$(cat domain.key)" \
  --name sshwifty \
  niruix/sshwifty:latest
```

The `domain.crt` and `domain.key` must be a valid TLS certificate and key file located on the same machine which the `docker run` command will be executed upon.

Compile from source code (Recommanded if you're a developer)

The following tools are required in order to build the software from source code:

- `git` to download the source code
- `node` and `npm` to build front-end application
- `go` to build back-end application

To start the build process, execute:

```
$ git clone https://github.com/nirui/sshwifty
$ cd sshwifty
$ npm install
$ npm run build
```

When done, you can found the newly generated `sshwifty` binary inside the current working directory.

Notice: `Dockerfile` contains the entire build procedure of this software. Please refer to it when you encounter any compile/build related issue.

Deploy on the cloud

To deploy this project onto the cloud, Google App Engine or Heroku for example, you need to first download the source code, then generate it locally before deploying it.

`npm run generate` command will generate all static files and automatically call `go generate ./...` to bind those static files directly into program source code. And you need those generated source code to get the software to function.

Trying to deploy ungenerated code directly to cloud will lead to failure, as required source code is missing.

Also keep in mind, if the cloud deployment process is `git` based, you may have to modify `.gitignore` file in order to allow all required files to be uploaded.

Révision #1
Créé 2022-04-04 16:04:23 UTC par garfieldtux
Mis à jour 2022-04-04 16:04:40 UTC par garfieldtux