Access configuration

As an Abinit Developer, you might want to make it easier and more comfortable for you to access the relevant repositories and tune your access parameters whenever needed, without having to restart from scratch every time you change something. There are two software components you need to tweak in order to achieve this: SSH and Git.

Here, we will suppose that you have already configured your Gitlab account properly and added the relevant SSH public keys to your profile, and that you have opened a terminal. Note that with Gitlab account we mean an account on our **internal server** at gitlab.abinit.org and not an account on the gitlab.com portal!

### Configuring SSH

To be able to push your contributions to the Abinit Forge efficiently, you need to add the following section to your `~/.ssh/config`. Just create the file if it does not exist.

```plaintext
Host abinit-forge
   HostName gitlab.abinit.org
   User git
   ServerAliveInterval 52
   Compression yes
```

You will then be able to use the `abinit-forge` keyword with Git to clone, pull, and push, to your repository. Later, whenever you need to tune any SSH-related parameter, you will just have to edit `~/.ssh/config` and all your changes will be automatically and instantly available to Git, in all your working trees.

### Configuring Git for the first time

If this is the very first time you run Git and plan to contribute to Abinit, we urge you to set the following parameters before doing anything else, by typing the following:

```bash
mkdir -p $HOME/.config/git
git config --global user.name "Firstname Lastname"
git config --global user.email "someone@some@server.somedomain"
git config --global core.editor "my_preferred_editor"
git config --global core.excludesFile "$HOME/.config/git/ignore"
git config --global color.ui "auto"
git config --global merge.conflictstyle "diff3"
```

where you replace *Firstname, Lastname, someone@some@server.somedomain, and my_preferred_editor*, by your respective first name, last name, email address, and preferred editor.
Configuring Git for the Abinit Forge

Once you have configured SSH to access the Abinit Forge, if this is the only remote address you use, the only thing you have to do with Git is using the `abinit-forge` keyword when dealing with the Forge.

To clone your repository, just type the following:

```
git clone abinit-forge:DEVELOPER/abinit
```

where you replace `DEVELOPER` by your Abinit Forge login.

If you had already cloned your repository before setting SSH and want to benefit from this new configuration, just go to the top directory of your Abinit working tree and type:

```
git remote set-url origin abinit-forge:DEVELOPER/abinit
```

Additional info: Setup of the SSH environment

In order to avoid typing your password every time you issue a command that accesses gitlab, you have to introduce your public keys in your profile (https://gitlab.abinit.org/profile/keys).

On your local machine, generate a ssh key of RSA type only WITHOUT passphrase:

```
ssh-keygen -t rsa
```

and call it `id_rsa_gitlab`.

Then add a section in the `~/.ssh/config` file:

```
host gitlab
  Hostname gitlab.abinit.org
  User git
  KeepAlive yes
  IdentityFile ~/.ssh/id_rsa_gitlab
```

and then, copy the public key `id_rsa_gitlab.pub` on gitlab.

Now, you can use (on your local machine) the following syntax:

```
git clone gitlab:user_id/abinit.git
```

instead of the above-mentioned

```
git clone git@gitlab.abinit.org:user_id/abinit.git
```

To be sure the key is proposed each time git calls ssh, you can use ssh-agent:

```
ssh-agent # this starts the agent, and provides the process id
# execute the 3 lines of commands that ssh-agent proposes, e.g.
SSH_AUTH_SOCK=/tmp/ssh-ngsERHER3K1HS/agent.15589; export SSH_AUTH_SOCK;
```
Configuring Git for Launchpad

If you are using and/or developing Abinit-related packages located on Launchpad, we recommend you to add the following to your Git configuration:

```
ssh add ~/.ssh/id_rsa_gitlab # add the corresponding ssh key for gitlab
```

Once done, you can interact with Launchpad through the “lp:” prefix, e.g.:

```
git clone lp:abinit-fallbacks
```

or:

```
git clone lp:~abinit-fallbacks-developers/libpsml
```

From:
https://wiki.abinit.org/ - Tips for ABINIT users and developers

Permanent link:

Last update: 2019/11/05 01:05