

# Using Mercurial on Shared Web Hosting

August 25, 2009

*Mercurial* is a distributed version control system.

Combine it with *ssh*, and it's easy to make backups to any server, local or remote.

In theory, the 'hg + ssh' combination will work on any remote server, even shared (i.e., cheap) web hosting accounts.

But some web hosts don't support Mercurial, and their admins might be less than willing to install it for you.

So here's how to get around that problem (for this example, I'm assuming you have ssh access to your web host account, that the web hosting OS is some flavor of Linux):

(1) Download the Mercurial source code:

```
$ wget http://mercurial.selenic.com/release/mercurial-1.3.1.tar.gz
```

(2) Unpack, and install using the "**Per-user installation**" instructions from the [unix install guide](#), i.e.:

```
$ make install-home # add PYTHON=/path/to/python2.3-or-newer if necessary
$ export PYTHONPATH=${HOME}/lib/python # bash syntax, ymmv
$ export PATH=${HOME}/bin:$PATH # add these to your .bashrc
```

Now, with *.bashrc* updated to include *~/lib/python* in *\$PYTHONPATH* and *~/bin* in *\$PATH*, running Mercurial at the prompt should work:

```
$ hg version
Mercurial Distributed SCM (version 1.3.1)

Copyright (C) 2005-2009 Matt Mackall and others
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
```

Next, the hard part: getting ssh in non-interactive mode (which is how hg uses it to push/pull/clone/etc.) to recognize the newly updated *.bashrc*.

[This post](#) suggested a way around the problem:

- \* Create (or edit it if it already exists) a file named *~/.ssh/environment*
- \* and put this line inside:

```
ENV=~/.kshrc
```

This will make ssh export the ENV variable before calling ksh. In non interactive mode, ksh will source the file pointed to by the ENV variable. (see *man ksh* for more details).

Using a *~/.ssh/environment* file, though, requires that [PermitUserEnvironment](#) in *sshd\_config* is set to "yes".

Unfortunately, the admins running my particular web host account did not have "PermitUserEnvironment" set to yes, and they wouldn't change it, for "security" reasons.

The answer turned out to be: use the `--remotecmd` option in hg to explicitly source *.bashrc* before attempting to run hg itself.

So, for example, a pull from the remote server would look like this:

```
$ hg pull --remotecmd='source ~/.bashrc; hg' ssh://username@server/path/to/repository
```

Using `--remotecmd` will avoid the "remote: bash: hg: not found" error.

---

Archived from the original at <http://denis.papathanasiou.org/>

 Bitcoin Donate: [14TM4ADKJbaGEi8Qr8dh4KfPBQmjTshkZ2](https://blockchain.info/address/14TM4ADKJbaGEi8Qr8dh4KfPBQmjTshkZ2)