SSH Deploy Key Documentation

Release 0.1.1

Travis Bear

February 03, 2014

Contents

CHAPTER 1

Overview

SSH deploy-key is a high-performance tool that pushes out a user's ssh key file to one or more remote hosts that have a common password. ssh-deploy-key makes it possible to quickly deploy to hundreds or thousands of servers.

CHAPTER 2

Source Code

ssh-deploy-key code is hosted on Bitbucket. See

https://bitbucket.org/travis_bear/ssh-deploy-key

CHAPTER 3

Contents

3.1 Alternatives

Why use SSH Deploy Key?

3.1.1 The Case

There are lots of ways to copy out an ssh key to a remote host, such as by hand, with ssh-copy-id, or with configuration management tools.

Although ssh-deploy-key is not ideal for every situation, its speed and ease of use make it a good choice in many cases.

3.1.2 Alternatives

Clearly there are other good options for deploying ssh keys.

Deploying by Hand

ssh-deploy-key cannot deploy an ssh key to a host is on a different network, behind a jump box. In that case, deployment by hand is the way to go. But in other cases, ssh-copy-id is a better option. Even when just copying a key out to a single host, it's a faster, easier, and more reliable option. These advantages only increase when copying keys out to multiple hosts.

ssh-copy-id

ssh-copy-id is a great tool, but it's not the ideal solution for every scenario.

- ssh-copy-id is not installed by default on all systems, notably on Mac OS.
- ssh-copy-id has no concept of 'smart append'. It will append a key to the authorized keys file regardless of whether that key is already present.
- Scripting the use of ssh-copy-id for deploying to multiple remote hosts can be challenging:
  - Password is entered interactively for each host.
  - In the case where there are numerous remote hosts that have not seen before, you'd need to interactively allow each host to be added to your known_hosts file.

Configuration Management Tools

Configuration management tools like Puppet, Chef, Ansible, etc. can do a fine job of deploying your ssh key(s) to numerous remote hosts. But if you are not already set up to use them for key distribution, this solution can be overkill.

3.2 Compatibility

3.2.1 Python Versions Supported

ssh-deploy-key has been tested on these versions of Python:

- python 2.7

3.2.2 Unsupportable Python Versions

SSH Deploy Key uses the Paramiko ssh library. Paramiko is not a "pure python" solution. It has binary dependencies that make it incompatible with non-cpython implementations, including:

- jython
- pypy

3.2.3 Python 3

Python 3 support is a priority and will be tested soon.

3.3 Installation Notes

ssh-deploy-key is normally installed via pip. However, on some systems, there are source files that must be installed first.

3.3.1 Install the Prerequisites

ssh-deploy-key depends on the paramiko ssh library, which requires the Python sources. You can install these using the normal package managers for your OS.

Debian/Ubuntu (apt-get)

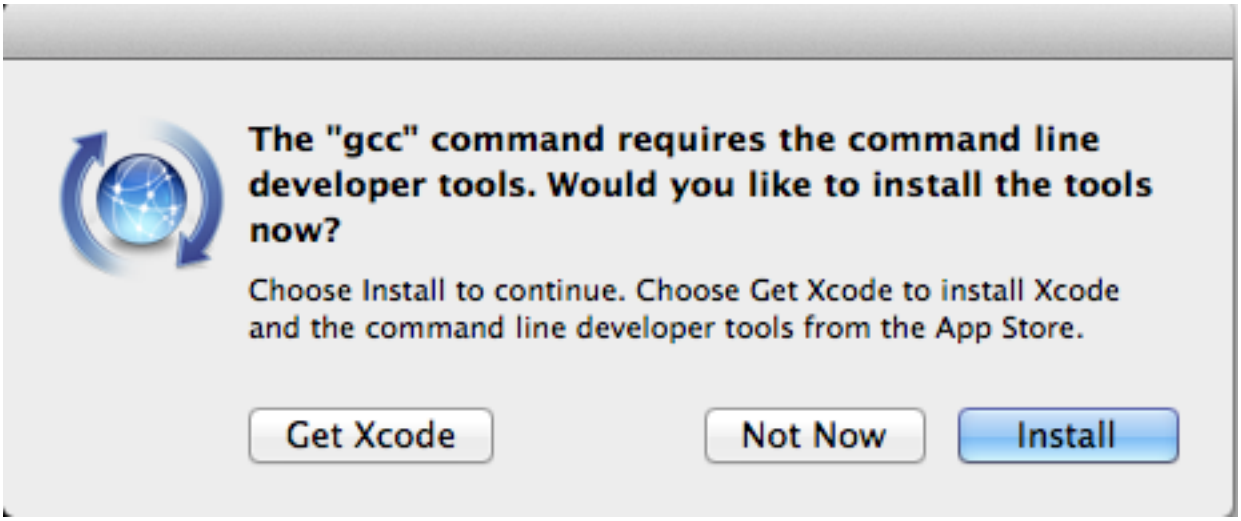
```
sudo apt-get install python-dev
```

Red Hat/Centos (yum)

```
sudo yum install python-devel
```

OS X

You will need a compiler installed – either XCode or gcc. Normally, you can just run the command to install ssh-deploy-key (see below), and if no compiler is available on your system, you will be prompted to install one:



If this happens, click the 'install' button, then run the pip command again.

3.3.2 Install ssh-deploy-key via Pip

Once the development libraries are in place, the best way to install ssh-deploy-key is via pip. To get pip, see http://www.pip-installer.org/en/latest/installing.html

Then,

```
sudo pip install ssh-deploy-key
```

3.4 Usage

ssh-deploy-key [options] [remote host[s]]

3.4.1 Options

```
usage: ssh-deploy-key [-h] [-a AUTHORIZED_KEYS] [-d] [-k KEY_FILE]
                      [-m TIMEOUT_SECONDS] [-o PORT] [-p PASSWORD]
                      [-s SSH_DIR] [-t THREADS] [-u USERNAME]
                      [hosts [hosts ...]]

Distribute an ssh key to remote hosts.

positional arguments:
  hosts                 Zero or more remote hosts to receive the ssh key. If
                        this value is unspecified, remote hosts will be read
                        from standard in.

optional arguments:
-h, --help              show this help message and exit
```

```
-a AUTHORIZED_KEYS, --authorized_keys AUTHORIZED_KEYS
                        Name of the remote authorized keys file. (Changing
                        this setting is uncommon.)
-d, --append            Add the ssh key to the end of the remote authorized
                        keys file instead of overwriting it. (SMART APPEND
                        NOT YET IMPLEMENTED).  Default is false.
-k KEY_FILE, --key_file KEY_FILE
                        Path to the local public ssh key file. Default is
                        ~/.ssh/id_rsa.pub
-m TIMEOUT_SECONDS, --timeout_seconds TIMEOUT_SECONDS
                        Timeout value (in seconds) for connecting to each
                        host. Default is 3
-o PORT, --port PORT    The ssh port to connect to the remote hosts on.
                        Default is 22
-p PASSWORD, --password PASSWORD
                        Password to use on remote hosts. If not specified
                        here, you will be prompted for this interactively.
-s SSH_DIR, --ssh_dir SSH_DIR
                        Directory to copy the key into on the remote host.
                        Default is ~/.ssh
-t THREADS, --threads THREADS
                        Number of threads to use for simultaneous key
                        distribution. Default is 100.
-u USERNAME, --username USERNAME
                        Username to use on remote hosts. Default is <current user>
```

3.4.2 Examples

These are some of the common ways to use ssh-deploy-id

Specifying remote hosts interactively

ssh-deploy-key can run interactively. The user will be prompted for additional hosts until typing 'exit' or ^D.

```
[~/git/ssh-deploy-key/bin]$ ./ssh-deploy-key
Enter common password for remote hosts:
Distributing key '/Users/travis/.ssh/id_rsa.pub' to remote hosts in overwrite mode.
Enter one hostname per line.  Terminate with 'exit' or ^D.
192.168.1.112
  copying key to travis@192.168.1.112:~/.ssh/authorized_keys... [SUCCESS!]
192.168.1.113
  copying key to travis@192.168.1.113:~/.ssh/authorized_keys... [SUCCESS!]
exit
```

Note that if you do not specify a password for the remote host on the command line, you will prompted for it interactively.

Specifying remote hosts on the command line

```
[~/git/ssh-deploy-key/bin]$ ./ssh-deploy-key 192.168.1.112 192.168.1.101
Enter common password for remote hosts:
Distributing key '/Users/travis/.ssh/id_rsa.pub' to remote hosts in overwrite mode.
  copying key to travis@192.168.1.112:~/.ssh/authorized_keys... [SUCCESS!]
  copying key to travis@192.168.1.101:~/.ssh/authorized_keys... [SUCCESS!]
```

With Shell Redirection

ssh-deploy-key accepts piped input. For example, if you had a script to generate a list of hosts, you could run it this way

```
get_host_list.sh | ssh-deploy-key
```

From a data File

If you have a data file listing your hosts already, you can redirect standard in from the file

```
ssh-deploy-key < host_list
```

Specifying the username and password on the command line

```
ssh-deploy-key -u root -p p@ssw0rd host1
```

3.5 License

3.5.1 License Version

SSH deploy key is licensed under version 2.1 of the GNU Lesser GPL.

3.5.2 License Text

The full text of the license is available here:

http://www.gnu.org/licenses/lgpl-2.1.txt