Set up SSH for Git

If you came to this page because you don't have SSH set up, then you have been using the secure hypertext transfer protocol (HTTPS) to communicate between your local system and Bitbucket Cloud. When you use HTTPS, you authenticate (supply a username and password) each time you take an action that requires a connection with Bitbucket. Who wants to do that? This page shows you how to use secure shell (SSH) to communicate with the Bitbucket server and avoid having to manually type a password all the time.

Set up SSH

Setting up an SSH identity can be prone to error. Allow yourself some time, perhaps as much as an hour depending on your experience, to complete this page. If you run into issues, check out Troubleshoot SSH Issues for extra information that may help you along. You can even skip this whole page and continue to use HTTPS if you want.

To use SSH with Bitbucket, you create an SSH identity containing a private key (on your local computer) and a public key (uploaded to Bitbucket) which create a key pair. After setting up SSH between your local system and Bitbucket, your system uses the key pair to authenticate you automatically to anything to which the associated account has access.

For security reasons, we recommend that you generate a new SSH key and replace the existing key on your account at least once a year.

There are a few important concepts you need when working with SSH identities and Bitbucket.

- You cannot reuse an identity's public key across accounts. You must create SSH identities for each individual Bitbucket account.
- You can associate multiple identities with a Bitbucket account.

Tell me why I would do that.

- RSA (R. Rivest, A. Shamir, L. Adleman are the originators) and digital signature algorithm (DSA) are key encryption algorithms. Bitbucket supports both types of algorithms. You

should create identities using whichever encryption method is most comfortable and available to you.

The following sections cover how to set up SSH for Git.

Set up SSH for Windows

Set up SSH for Mac OS/Linux

Step 1. Ensure you have an SSH client installed

SSH is most likely included with your version of Mac OS or Linux. To make sure, do the following to verify your installation:

1. From your terminal window, enter the following command, which identifies the version of SSH you have installed.
   If SSH is installed, you see something similar to the following:
   ```
   $ ssh -v
   OpenSSH_5.6p1, OpenSSL 0.9.8r 8 Feb 2011
   usage: ssh [-1246AaCfgKkMNnqsTtVvXxYy] [-b bind_address] [-c cipher_spec]
              [-D [bind_address:]port] [-e escape_char] [-F configfile]
              [-I pkcs11] [-i identity_file]
              [-L [bind_address:]port:host:hostport]
              [-l login_name] [-m mac_spec] [-O ctl_cmd] [-o option] [-p
   port]
   [-R [bind_address:]port:host:hostport] [-S ctl_path]
              [-W host:port] [-w local_tun[:remote_tun]]
              [user@]hostname [command]>
   ```
   If you have `ssh` installed, the terminal returns version information.
   If you don't have `ssh` installed, install it now.
2. List the contents of your `~/.ssh` directory.
   If you don't have an `.ssh` directory, don't worry, you'll create it the next section. If you have a `.ssh` directory or you may see something like this:
   ```
   $ ls -a ~/.ssh
   known_hosts
   ```
   If you have defined a default identity, you'll see the two `id_*` files:
   ```
   $ ls -a ~/.ssh
   .         ..        id_rsa        id_rsa.pub    known_hosts
   ```
   In this case, the default identity used RSA encryption (`id_rsa.pub`). If you want to use an existing default identity for your Bitbucket account, skip the next section and go to start the ssh-agent and load your keys.

Step 2. Set up your default identity

By default, the system adds keys for all identities to the `/Users/<yourname>/.ssh` directory on Mac OSX, or `/home/<yourname>/.ssh` on Linux. This procedure creates a default identity. If you have a default identity and you want to use it for Bitbucket, skip this step and go to start the ssh-agent and load your

keys. If you have an existing default identity but you forgot the passphrase, you can also use this procedure to overwrite your default identity and create a fresh one.

Want to Use Multiple Identities?

You can create multiple SSH identities. Doing this is an advanced topic and beyond the scope of this tutorial. For information on how to create multiple identities, see Configure multiple SSH identities for GitBash, Mac OSX, & Linux.

Use the following procedure to create a new default identity.

1. Open a terminal in your local system.
2. Enter `ssh-keygen` at the command line.
   The command prompts you for a file where you want to save the key. If the `.ssh` directory doesn't exist, the system creates one for you.
   ```
   $ ssh-keygen
   Generating public/private rsa key pair.
   Enter file in which to save the key (/Users/emmap1/.ssh/id_rsa):
   ```
3. Press the Enter or Return key to accept the default location.
4. Enter and re-enter a passphrase when prompted.
   Unless you need a key for a process such as script, you should always provide a passphrase. The command creates your default identity with its public and private keys. The whole interaction will look similar to the following:
   ```
   $ ssh-keygen
   Generating public/private rsa key pair.
   Enter file in which to save the key (/Users/emmap1/.ssh/id_rsa):
   Created directory '/Users/emmap1/.ssh'.
   Enter passphrase (empty for no passphrase):
   Enter same passphrase again:
   Your identification has been saved in /Users/emmap1/.ssh/id_rsa.
   Your public key has been saved in /Users/emmap1/.ssh/id_rsa.pub.
   The key fingerprint is:
   4c:80:61:2c:00:3f:9d:dc:08:41:2e:c0:cf:b9:17:69 emmap1@myhost.local
   The key's randomart image is:
   +--[ RSA 2048]----+
   |*o+ooo.          |
   |.+.=o+ .         |
   |. *.* o .        |
   | . = E o         |
   |    o . S        |
   |   . .           |
   |     .           |
   |                 |
   |                 |
   +-----------------+
   ```
5. List the contents of `~/.ssh` to view the key files.
   ```
   $ ls -a ~/.ssh
   ```

Step 3. Start the ssh-agent and load your keys

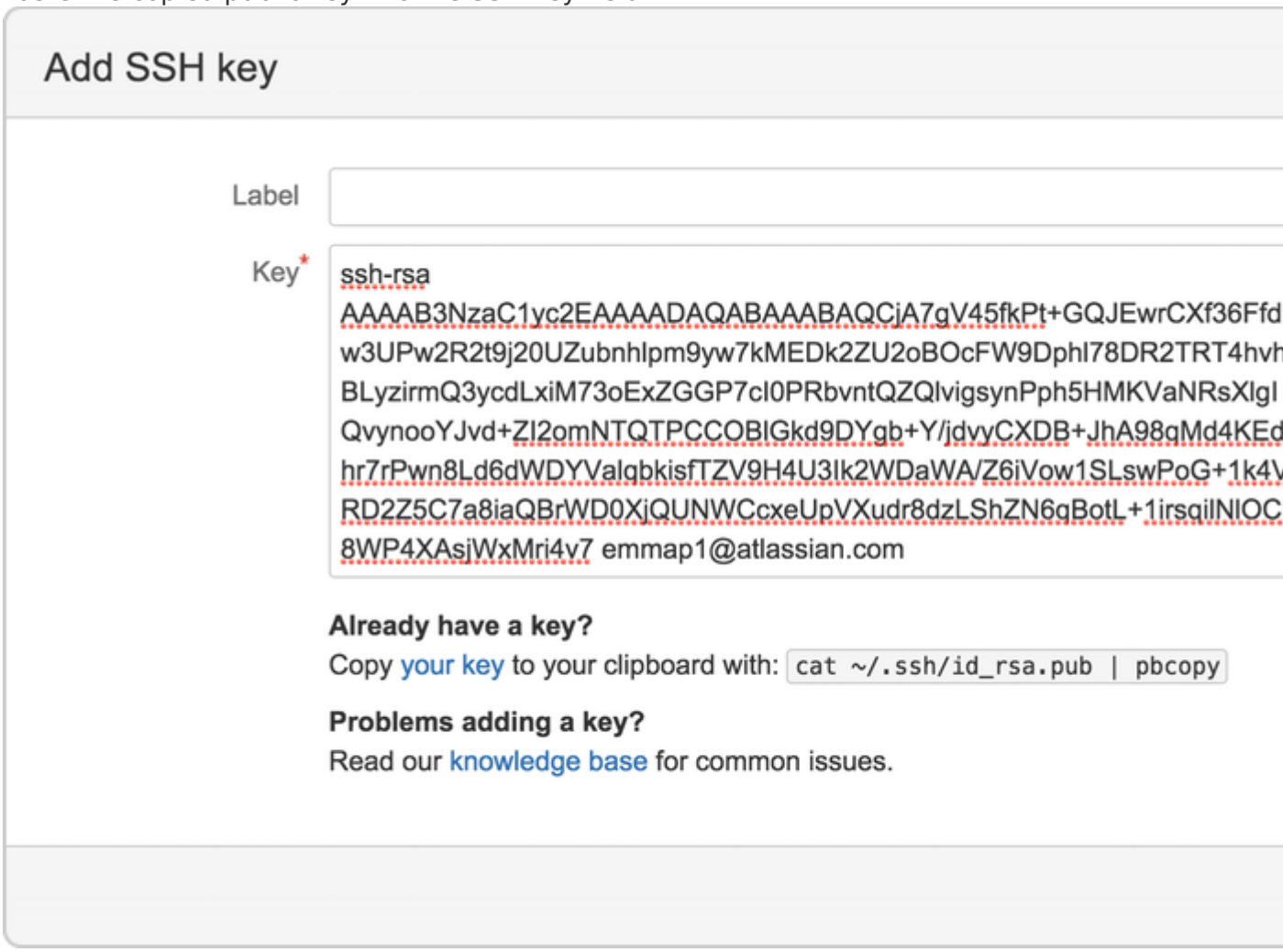
If you are running OSX 10.6.8 or later you can skip this step. The OSX 10.6.8 system asks for your connection parameters the first time you try to establish a SSH connection. Then, it automatically starts the `ssh-agent` for you. If you don't have OSX 10.6.8 or are running another Linux operating system, do the following:

1. Open a terminal window and enter the `ps -e | grep [s]sh-agent` command to see if the agent is running.
   ```
   $ ps -e | grep [s]sh-agent
    9060 ?? 0:00.28 /usr/bin/ssh-agent -l
   ```
2. If the agent isn't running, start it manually with the following command:
   ```
   $ ssh-agent /bin/bash
   ```
3. Load your new identity into the `ssh-agent` management program using the `ssh-add` command.
   ```
   $ ssh-add ~/.ssh/id_rsa
   Enter passphrase for /Users/emmap1/.ssh/id_rsa:
   Identity added: /Users/emmap1/.ssh/id_rsa (/Users/emmpa1/.ssh/id_rsa)
   ```
4. Use the `ssh-add` command to list the keys that the agent is managing.
   ```
   $ ssh-add -l
   2048 7a:9c:b2:9c:8e:4e:f4:af:de:70:77:b9:52:fd:44:97
   /Users/manthony/.ssh/id_rsa (RSA)
   ```

Step 4. Install the public key on your Bitbucket account

1. From Bitbucket Cloud, choose avatar > Bitbucket settings from the application menu. The system displays the Account settings page.
2. Click SSH keys.
   The SSH Keys page displays. If you have any existing keys, those appear on this page.
3. Back in your terminal window, copy the contents of your public key file.
   For example, in Linux you can `cat` the contents.
   ```
   $ cat ~/.ssh/id_rsa.pub
   ```
   In Mac OSX, the following command copies the output to the clipboard:
   ```
   $ pbcopy < ~/.ssh/id_rsa.pub
   ```
4. Back in your browser, enter a Label for your new key, for example, `Default public key`.

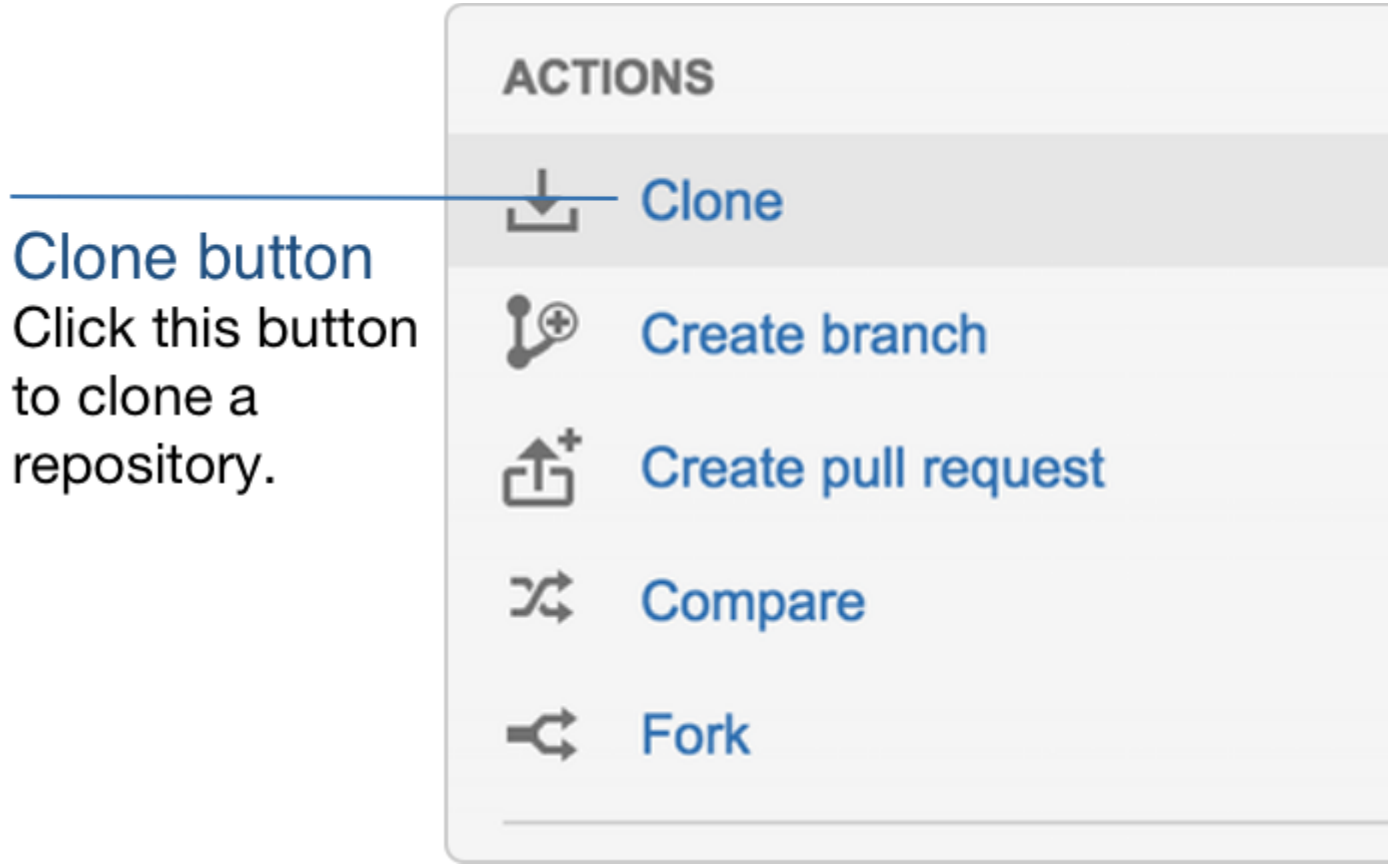
5. Paste the copied public key into the SSH Key field:



6. Press Add key.
   The system adds the key to your account. Bitbucket sends you an email to confirm addition of the key.

Step 5. Change an existing repo from HTTPS to the SSH protocol

The URL you use for a repo depends on which protocol you are using, HTTPS or SSH. The Clone button of your repository has a quick way for you to see these URLS.



Experiment for a moment, clicking back and forth between the SSH and the HTTPS protocol links to see how the URLs differ. The table below shows the format based on protocol.

SSH URL format	`git@bitbucket.org :< accountname>/<reponame>.git` ***or*** `ssh://git@bitbucket.org /< accountname>/<reponame>.git`
HTTPS URL format	`https://<accountname>@bitbucket.org/< accountname>/<reponame>.git`

To make the change, go to a terminal on your local system and navigate to your repository locally. Then, do the following:

1. View your current repo configuration.
   You should see something similar to the following:

```
$ cd ~/<path_to_repo>
$ cat .git/config
[core]
    repositoryformatversion = 0
    filemode = true
    bare = false
    logallrefupdates = true
    ignorecase = true
    precomposeunicode = true
[remote "origin"]
    fetch = +refs/heads/*:refs/remotes/origin/*
    url = https://emmap1@bitbucket.org/emmap1/bitbucketspacestation.git
```

```
[branch "master"]
    remote = origin
    merge = refs/heads/master
```

As you can see, the `url` is using the HTTPS protocol. There are a number of ways to change this value, the easiest way is just to edit the repo's configuration file.

2. Open the `~/<path_to_repo>/.git/config` file with your favorite editor.
3. Change the `url` value to use the SSH format for that repo.
   When you are done you should see something similar to the following:

```
    fetch = +refs/heads/*:refs/remotes/origin/*
    url = git@bitbucket.org:emmap1/bitbucketspacestation.git
```

Step 6. Make a change under the new protocol

1. From the local directory of your repository, create a new file called `README.txt`.
2. Add the following text to the `README.txt` file:

```
------------------------------
This repo is a practice repo I am using to learn bitbucket.
You can access this repo with SSH or with HTTPS.
```

3. Save and close the file.
4. Add and then commit your change to your local repo.

```
$ git add README.txt
$ git commit -m "making a change under the SSH protocol"
```

5. Push your changes.
   The system warns you that it is adding the Bitbucket host to the list of known hosts.

```
$ git push
Warning: Permanently added the RSA host key for IP address
'207.223.240.182' to the list of known hosts.
Counting objects: 5, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 314 bytes, done.
Total 3 (delta 1), reused 0 (delta 0)
remote: bb/acl: emmap1 is allowed. accepted payload.
To git@bitbucket.org:emmap1/bitbucketspacestation.git
   d3bb337..f0b152f master -> master
```

6. Open the repo Overview in Bitbucket to view your commit.