Set up an SSH key

confluence.atlassian.com/bitbucket/set-up-an-ssh-key-728138079.html

When you set up SSH, you create a key pair that contains a private key (saved to your local computer) and a public key (uploaded to Bitbucket). Bitbucket uses the key pair to authenticate anything the associated account can access. This two-way mechanism prevents man-in-the-middle .

This first key pair is your default SSH identity. If you need more than a default identity, you can set up additional keys.

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.

You can't use the same key between accounts. You must create for each individual Bitbucket account.

your options when setting up SSH:

Set up SSH for Git on Windows

Use this section to create a default identity and SSH key when you're using Git on Windows. By default, the system adds keys for all identities to the `/Users/<username>/.ssh` directory.

› Expand for details

Step 1. Set up your default identity

1. From the , enter `ssh-keygen` .

   For Windows 7 or earlier

   You can only enter `ssh-keygen` into the Git Bash window. It won't work in the Command prompt.

   The command prompts you for a file to save the key in:

   ```
   $ ssh-keygen
   Generating public/private rsa key pair.
   Enter file in which to save the key (/c/Users/emmap1/.ssh/id_rsa):
   ```

2. Press enter to accept the default key and path, `/c/Users/<username>/.ssh/id_rsa` .

   ℹ

   We recommend you keep the default key name unless you have a reason to change it.

   To create a key with a name or path other than the default, specify the full path to the key. For example, to create a key called `my-new-ssh-key` , you would enter the Windows path, shown here:

   ```
   $ ssh-keygen
   Generating public/private rsa key pair.
   Enter file in which to save the key (/c/Users/emmap1/.ssh/id_rsa):
   c:\Users\emmap1\.ssh\my-new-ssh-key
   ```

3. Enter and re-enter a passphrase when prompted.

   The command creates your default identity with its public and private keys. The whole interaction looks similar to :

   ```
   $ ssh-keygen
   Generating public/private rsa key pair.
   Enter file in which to save the key (/c/Users/emmap1/.ssh/id_rsa):
   Created directory '/c/Users/emmap1/.ssh'.
   Enter passphrase (empty for no passphrase):
   Enter same passphrase again:
   Your identification has been saved in /c/Users/emmap1/.ssh/id_rsa.
   Your public key has been saved in /c/Users/emmap1/.ssh/id_rsa.pub.
   The key fingerprint is: e7:94:d1:a3:02:ee:38:6e:a4:5e:26:a3:a9:f4:95:d4
   emmap1@EMMA-PC
   ```

4. List the contents of `.ssh` to view the key files.

   You should see something like the following:

   ```
   $ dir .ssh
   id_rsa id_rsa.pub
   ```

   The command displays two files, one for the public key (for example `id_rsa.pub`) and one for the private key (for example, `id_rsa`).

Step 2. Add the key to the ssh-agent

If you don't want to type your password each time you use the key, you'll need to add it to the ssh-agent.

1. To start the agent, run the following:

   ```
   $ eval $(ssh-agent)
   Agent pid 9700
   ```

2. Enter `ssh-add` followed by the path to the private key file:

   ```
   $ ssh-add ~/.ssh/<private_key_file>
   ```

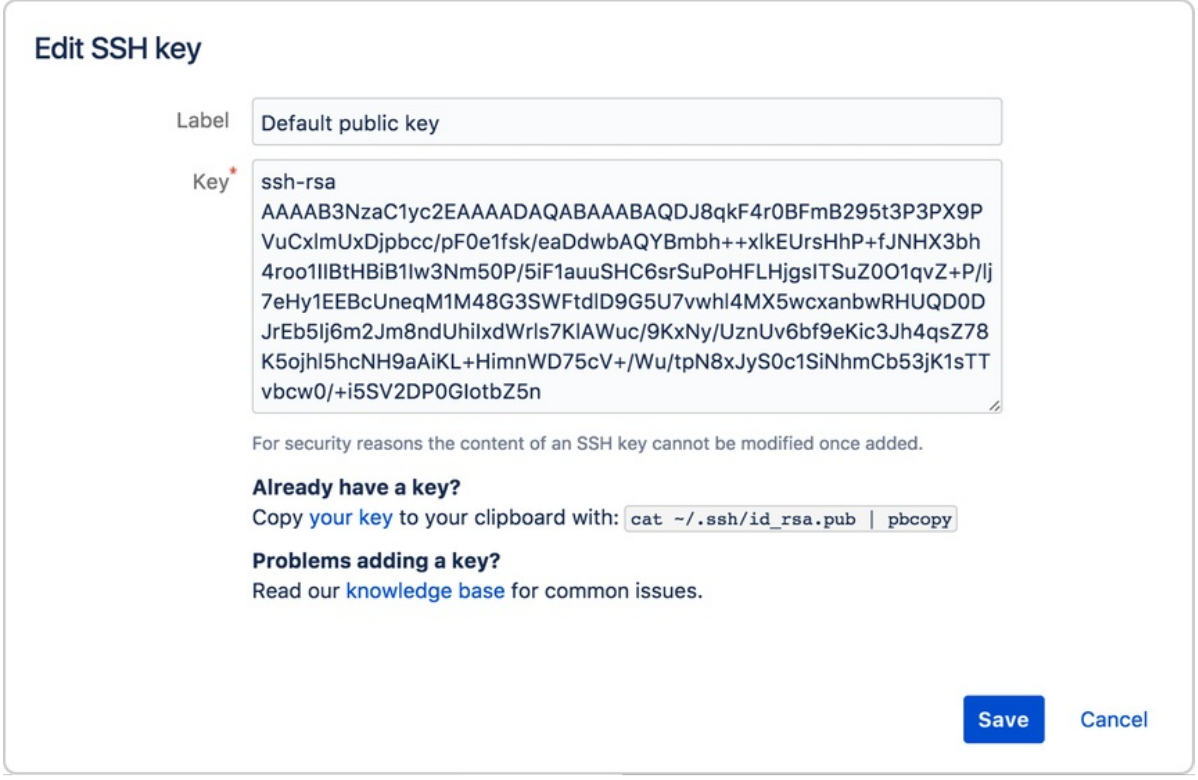
Step 3. the public key to your Bitbucket settings

1. From Bitbucket, choose **Bitbucket settings** from your avatar in the lower left.
   The **Account settings** page opens.
2. Click **SSH keys**.
   If you've already added keys, you'll see them on this page.
3. Open your `.ssh/id_rsa.pub` file (or whatever you named the public key file) and copy its contents.
   You may see an email address on the last line. It doesn't matter whether or not you include the email address.

4. From Bitbucket, click **Add key**.

5. Enter a **Label** for your new key, for example, `Default public key`.

6. Paste the copied public key into the SSH **Key** field.



Edit SSH key

Label: Default public key

Key*:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDJ8qkF4r0BFmB295t3P3PX9P
VuCxlmUxDjpbcc/pF0e1fsk/eaDdwbAQYBmbh++xlkEUrsHhP+fJNHX3bh
4roo1IlBtHBiB1lw3Nm50P/5iF1auuSHC6srSuPoHFLHjgsITSuZ0O1qvZ+P/lj
7eHy1EEBcUneqM1M48G3SWFtdlD9G5U7vwhl4MX5wcxanbwRHUQD0D
JrEb5lj6m2Jm8ndUhilxdWrls7KlAWuc/9KxNy/UznUv6bf9eKic3Jh4qsZ78
K5ojhl5hcNH9aAiKL+HimnWD75cV+/Wu/tpN8xJyS0c1SiNhmCb53jK1sTT
vbcw0/+i5SV2DP0GlotbZ5n
```

For security reasons the content of an SSH key cannot be modified once added.

Already have a key?
Copy your key to your clipboard with: `cat ~/.ssh/id_rsa.pub | pbcopy`

Problems adding a key?
Read our knowledge base for common issues.

7. Click **Save**.
   Bitbucket sends you an email to confirm the addition of the key.

   ⓘ
   Edit an SSH key

   After you add a key, you can edit the key's **Label** but not the key itself.

8. Return to the and verify your configuration and username by entering the following command:

   ```
   $ ssh -T git@bitbucket.org
   ```

   The command message tells you which of your Bitbucket accounts can log in with that

   ```
   conq: logged in as emmap1.
   You can use git or hg to connect to Bitbucket. Shell access is disabled.
   ```

   If you get an error message with `Permission denied (publickey)`, check the Troubleshoot SSH issues page for .

Now that you've got an SSH key set up, use the SSH URL the next time you clone a repository. If you already have a repository that you cloned over HTTPS, change the remote URL for your repository to its SSH URL.

Set up SSH on macOS/Linux

Whether you use Git or Mercurial, use this section to create a default identity and SSH key when on macOS or Linux. By default, the system adds keys to the `/Users/<yourname>/.ssh` directory on macOS and `/home//.ssh` on Linux.

› Collapse

Step 1. Set up your default identity

1. From the terminal, enter `ssh-keygen` at the command line.
   The command prompts you for a file to save the key in:

   ```
   $ ssh-keygen
   Generating public/private rsa key pair.
   Enter file in which to save the key (/Users/emmap1/.ssh/id_rsa):
   ```

2. Press the Enter or Return key to accept the default location.

   ⓘ

   We recommend you keep the default key name unless you have a reason to change it.

   To create a key with a name or path other than the default, specify the full path to the key. For example, to create a key called `my-new-ssh-key`, enter a path like the one shown at the prompt:

   ```
   $ ssh-keygen
   Generating public/private rsa key pair.
   Enter file in which to save the key (/Users/emmap1/.ssh/id_rsa):
   /Users/emmap1/.ssh/my-new-ssh-key
   ```

3. Enter and re-enter a passphrase when prompted.
   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//.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        |
|   . .           |
|     .           |
|                 |
|                 |
+-----------------+
```

4. List the contents of `~/.ssh` to view the key files.

```
$ ls ~/.ssh
id_rsa id_rsa.pub
```

   The command displays two files, one for the public key (for example `id_rsa.pub`) and one for the private key (for example, `id_rsa`).

Step 2. Add the key to the ssh-agent

If you don't want to type your password each time you use the key, you'll need to add it to the ssh-agent.

1. To start the agent, run the following:

```
$ eval `ssh-agent`
Agent pid 9700
```

2. Enter `ssh-add` followed by the path to the private key file:

```
$ ssh-add -K ~/.ssh/<private_key_file>
```

```
$ ssh-add ~/.ssh/<private_key_file>
```

3. (macOS only) So that your computer remembers your password each time it restarts, open (or create) the `~/.ssh/config` file and add these lines to the file:

```
Host *
  UseKeychain yes
```

Step 3. (Mercurial only) Enable SSH compression

Enabling SSH compression is recommended but not required.

By default, Git automatically performs compression when sending or retrieving data, but Mercurial doesn't. Enabling SSH compression can speed up sending and retrieving data, drastically in some cases.

To enable SSH compression:

1. Open the Mercurial global configuration file (`~/.hgrc`).
2. Add line to the UI section:

   ```
   ssh = ssh -C
   ```

   When you are done the file should look similar to this:

   ```
   [ui]
   # Name data to appear in commits
   username = Emma <emmap1@atlassian.com>
   ssh = ssh -C
   ```

3. Save and close the file.

Step 4. Add the public key to your Bitbucket settings

1. From Bitbucket, choose **Bitbucket settings** from your avatar in the lower left.
   The **Account settings** page opens.
2. Click **SSH keys**.
   If you've already added keys, you'll see them on this page.
3. In your terminal window, copy the contents of your public key file. If you renamed the key, replace `id_rsa.pub` with the public key file name.

   Linux, you can `cat` the contents:

   ```
   $ cat ~/.ssh/id_rsa.pub
   ```

   On macOS, the following command copies the output to the clipboard:

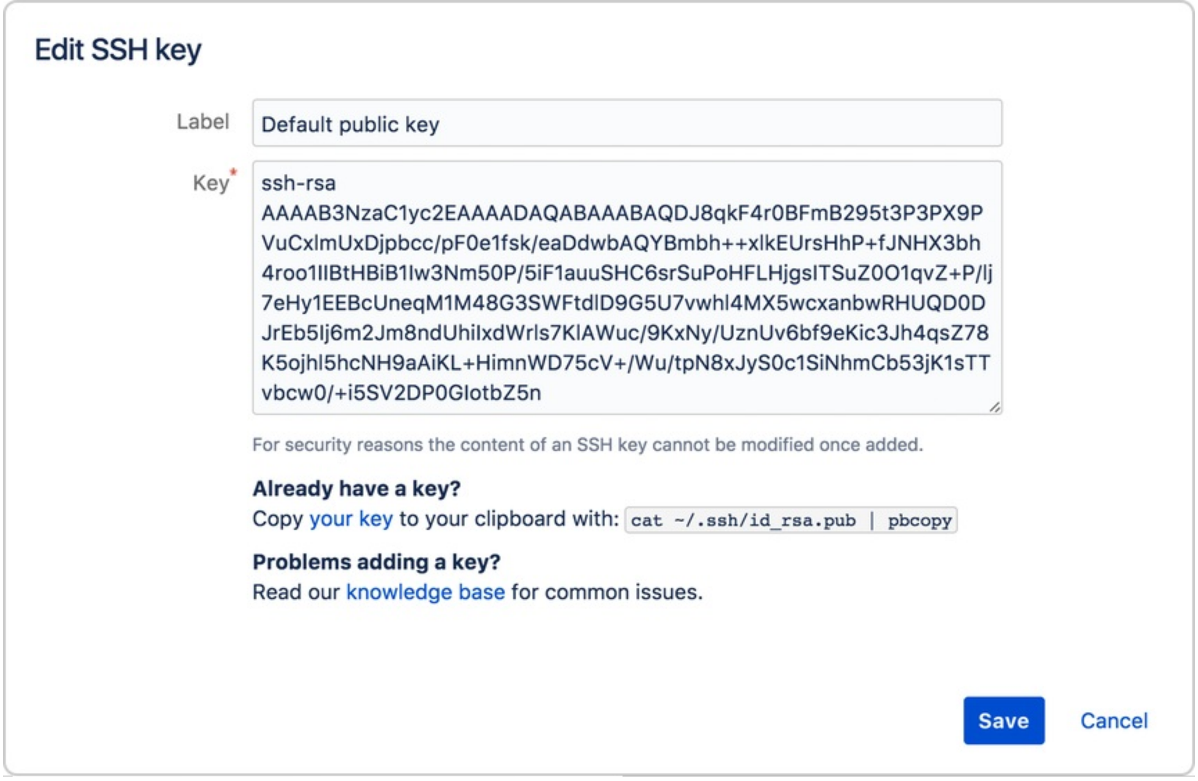
   ```
   $ pbcopy < ~/.ssh/id_rsa.pub
   ```

4. Select and copy the key output in the clipboard.
   If you have problems with copy and paste, you can open the file directly with Notepad. Select the contents of the file (just avoid selecting the end-of-file characters).

5. From Bitbucket, click **Add key**.

6. Enter a **Label** for your new key, for example, `Default public key`.

7. Paste the copied public key into the SSH **Key** field.
   You may see an email address on the last line when you paste. It doesn't matter whether or not you include the email address in the **Key**.



Edit SSH key

Label: Default public key

Key*:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDJ8qkF4r0BFmB295t3P3PX9P
VuCxlmUxDjpbcc/pF0e1fsk/eaDdwbAQYBmbh++xlkEUrsHhP+fJNHX3bh
4roo1IlBtHBiB1lw3Nm50P/5iF1auuSHC6srSuPoHFLHjgsITSuZ0O1qvZ+P/lj
7eHy1EEBcUneqM1M48G3SWFtdlD9G5U7vwhl4MX5wcxanbwRHUQD0D
JrEb5lj6m2Jm8ndUhilxdWrls7KlAWuc/9KxNy/UznUv6bf9eKic3Jh4qsZ78
K5ojhl5hcNH9aAiKL+HimnWD75cV+/Wu/tpN8xJyS0c1SiNhmCb53jK1sTT
vbcw0/+i5SV2DP0GlotbZ5n
```

For security reasons the content of an SSH key cannot be modified once added.

Already have a key?
Copy your key to your clipboard with: `cat ~/.ssh/id_rsa.pub | pbcopy`

Problems adding a key?
Read our knowledge base for common issues.

8. Click **Save**.
   Bitbucket sends you an email to confirm the addition of the key.

ℹ
Edit an SSH key

After you add a key, you can edit the key's **Label** but not the key itself. To change the key's contents, you need to delete and re-add the key.

9. Return to the terminal window and verify your configuration and username by entering the following command:

   ```
   $ ssh -T git@bitbucket.org
   ```

   The command message tells you which of your Bitbucket accounts can log in with that key.

   ```
   conq: logged in as emmap1.
   You can use git or hg to connect to Bitbucket. Shell access is disabled.
   ```

   If you get an error message with `Permission denied (publickey)`, check the Troubleshoot SSH issues page for help.

Now that you've got an SSH key set up, use the SSH URL the next time you clone a repository. If you already have a repository that you cloned over HTTPS, change the remote URL for your repository SSH URL.

Set up SSH with Sourcetree on Windows

Whether you use Git or Mercurial, use this section to generate an SSH key using Sourcetree.

› Expand for details

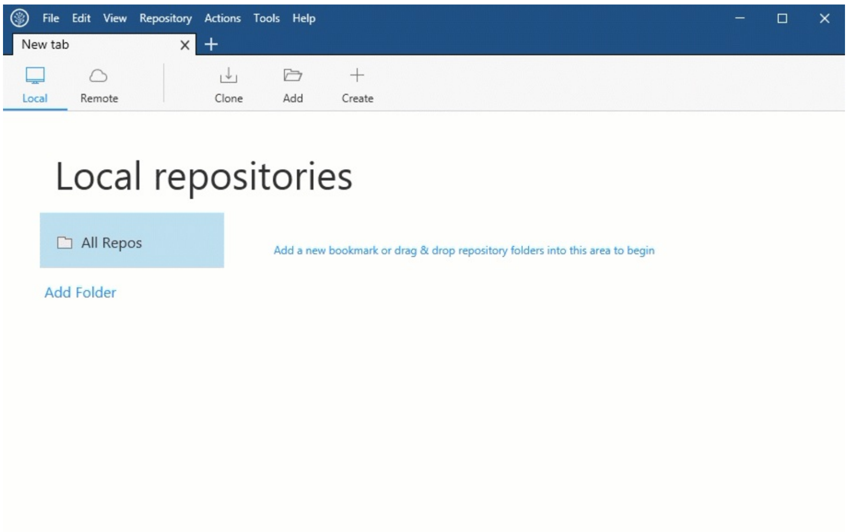
Step 1. Install Sourcetree and add your Bitbucket account

1. If you don't yet have Sourcetree, go to https://www.sourcetreeapp.com/ and click the **Download free** button.
2. Click the to install Sourcetree. Refer to the Install Sourcetree page for more details.

    1. You may see the **Load SSH Key?** dialog after installation. Click **No** if you don't have one and want to use Sourcetree to create one.

    2. You may see the **Sourcetree: Mercurial not found** dialog after installation. Choose an appropriate option or select **Download an embedded version of Mercurial for Sourcetree along to use**.

3. **SSHPreferred Protocol** If you don't connect your account during set up, click **Remote** to the **Remote repositories** page and click **Add an account**.



Local repositories

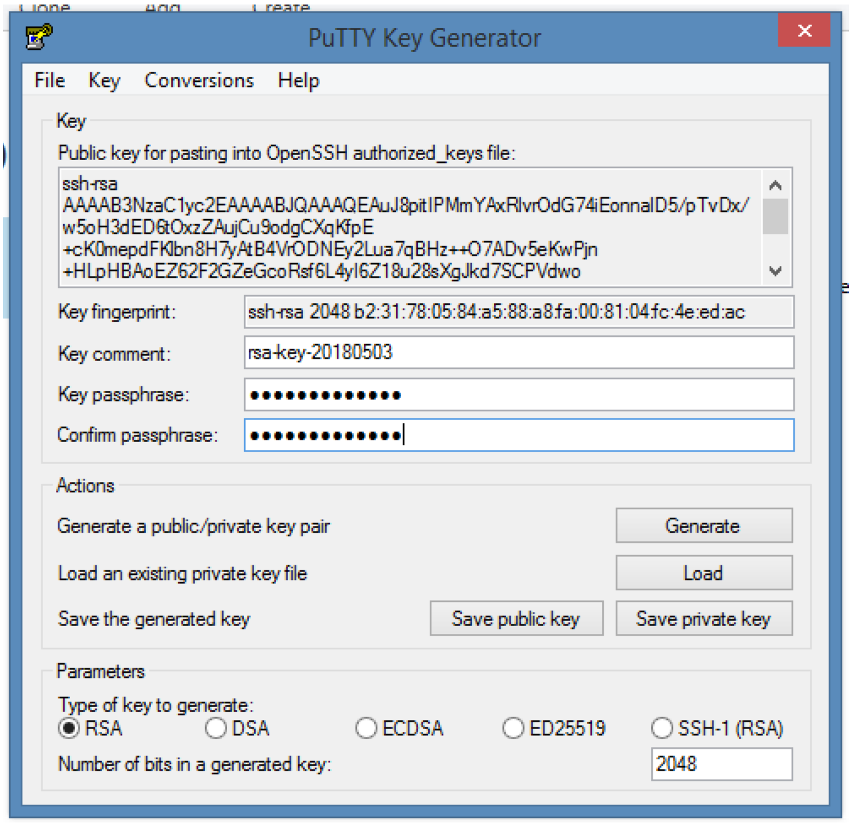
All Repos

Add a new bookmark or drag & drop repository folders into this area to begin

Add Folder

Step 2. Create an SSH key

1. From **Tools**, select **Create or Import SSH Keys**.
2. From the **PuTTY Key Generator** dialog, click the **Generate** button.
3. When SSH key generation is complete, you see the public key and a few other fields.
4. Enter a passphrase for your SSH key in the **Key passphrase** and **Confirm passphrase** fields.



Public key for pasting into OpenSSH authorized_keys file:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAABJQAAAQEAuJ8pitIPMmYAxRlvrOdG74iEonnalD5/pTvDx/
w5oH3dED6tOxzZAujCu9odgCXqKfpE
+cK0mepdFKlbn8H7yAtB4VrODNEy2Lua7qBHz++O7ADv5eKwPjn
+HLpHBAoEZ62F2GZeGcoRsf6L4yl6Z18u28sXgJkd7SCPVdwo
```

Key fingerprint: ssh-rsa 2048 b2:31:78:05:84:a5:88:a8:fa:00:81:04:fc:4e:ed:ac

Key comment: rsa-key-20180503

Key passphrase: ••••••••••••••

Confirm passphrase: ••••••••••••••

Generate a public/private key pair

Load an existing private key file

Save the generated key

Type of key to generate:
● RSA ○ DSA ○ ECDSA ○ ED25519 ○ SSH-1 (RSA)

Number of bits in a generated key: 2048

5. **Save public key**. From the save dialog, choose where to save your public key, name the file with the `.pub` file extension, and click **Save**.
6. Click **Save private key**. From the save dialog, choose where to save your private key, name the file, and click **Save**.
7. Close the **PuTTY Key Generator** dialog.

. (Mercurial only) Enable SSH compression

ℹ

Enabling SSH compression is recommended but not required. You'll need to clone and open a Mercurial repository to complete these steps.

By default, Git automatically performs compression when sending or retrieving data, but Mercurial doesn't. Enabling SSH compression can speed up sending and retrieving data, drastically in some cases.

To enable SSH compression:

1. Click **Settings** in the top right of the repository window.
   The repository settings may open to the **Remotes** tab. If not, click the **Remotes** tab.
2. Click **Edit Config File** to open the Mercurial global configuration file (`~/.hgrc`).
3. Add line to the UI section:

   ```
   ssh = ssh -C
   ```

   When you're done the file should look similar to this:

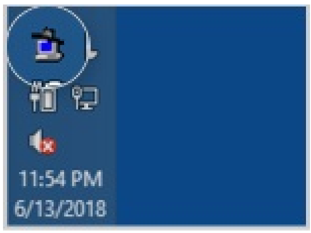
   ```
   [ui]
   # name and email (local to this repository, optional), e.g.
   username = Emma <emmap1@atlassian.com>
   ssh = ssh -C
   ```

4. Save and close the file.

Step 4. Install your private key on Pageant

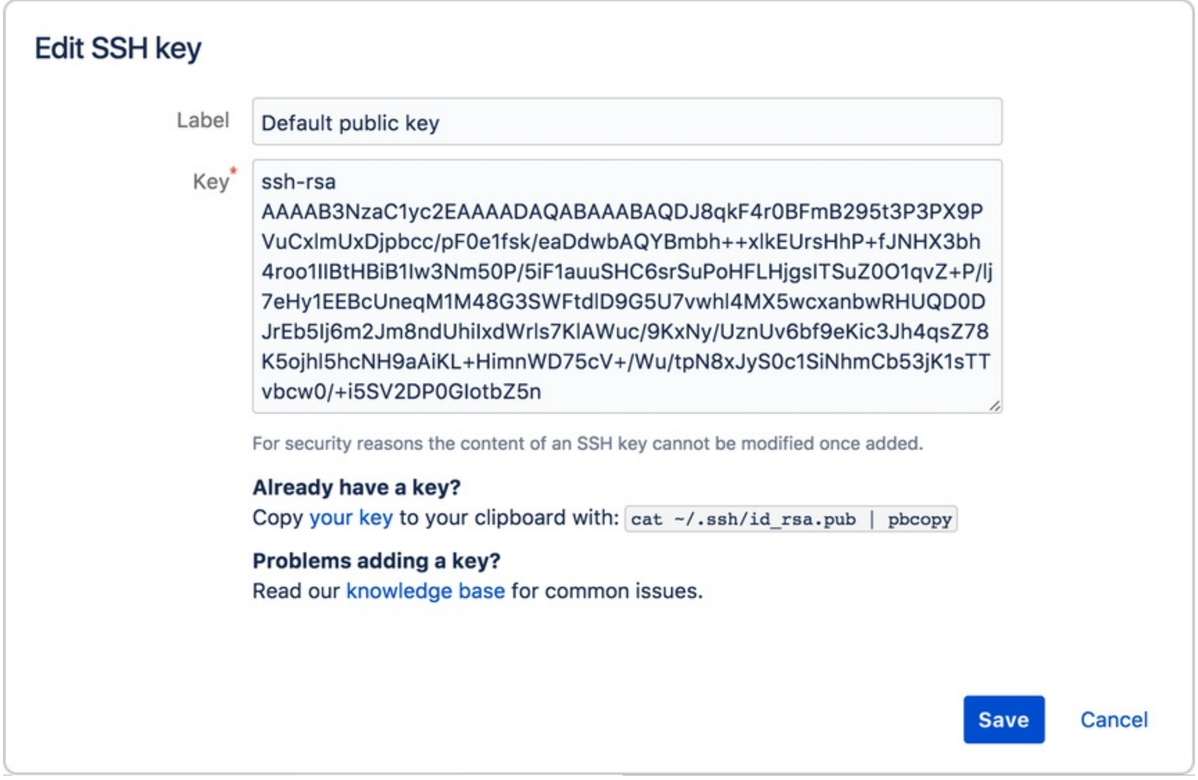
Sourcetree comes with an SSH authentication agent called Pageant. Load your private key into Pageant to automatically authenticate so that you don't need to enter your passphrase.

1. in your system tray to open the **Key List** dialog.
2. **Add Key** to open**Select Private Key File**
3. Navigate to the private key file you saved in Step 1 and click **Open**.
4. Enter the passphrase for your SSH key and click **OK**.
   Pageant shows your key in the running list.
5. **Close**



Step 5. Add the public key to your Bitbucket settings

1. From Sourcetree, open the **PuTTY Key Generator** dialog by going to **Tools > Create or Import SSH Keys**.
2. Click **Load**, navigate to your SSH folder, and click the private key. Make sure you're looking at All files if you don't see your private key.
3. Enter your passphrase for the SSH key and click **OK**.
4. Copy the public key in the first field.
5. From Bitbucket, choose **Bitbucket settings** from your avatar in the lower left.
   The **Account settings** page opens.
6. Click **SSH keys**.
   If you've already added keys, you'll see them on this page.
7. Click **Add key**.
8. Enter a **Label** for your new key, for example, `Default public key`.
9. Paste the copied public key into the SSH **Key** field.



Edit SSH key

Label: Default public key

Key*:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDJ8qkF4r0BFmB295t3P3PX9P
VuCxlmUxDjpbcc/pF0e1fsk/eaDdwbAQYBmbh++xlkEUrsHhP+fJNHX3bh
4roo1IIBtHBiB1Iw3Nm50P/5iF1auuSHC6srSuPoHFLHjgsITSuZ0O1qvZ+P/lj
7eHy1EEBcUneqM1M48G3SWFtdlD9G5U7vwhl4MX5wcxanbwRHUQD0D
JrEb5Ij6m2Jm8ndUhiIxdWrls7KIAWuc/9KxNy/UznUv6bf9eKic3Jh4qsZ78
K5ojhl5hcNH9aAiKL+HimnWD75cV+/Wu/tpN8xJyS0c1SiNhmCb53jK1sTT
vbcw0/+i5SV2DP0GlotbZ5n
```

For security reasons the content of an SSH key cannot be modified once added.

Already have a key?
Copy your key to your clipboard with: `cat ~/.ssh/id_rsa.pub | pbcopy`

Problems adding a key?
Read our knowledge base for common issues.

10. Click **Save**.
    Bitbucket sends you an email to confirm the addition of the key.

    ℹ
    Edit an SSH key

    After you add a key, you can edit the key's **Label** but not the key itself. T

Now that you've got an SSH key set up, use the SSH URL the next time you clone a repository. If you already have a repository that you cloned over HTTPS, change the remote URL for your repository to its SSH URL.

Set up SSH with Sourcetree on macOS

Whether you use Git or Mercurial, use this section to generate an SSH key using Sourcetree.

› Expand for details

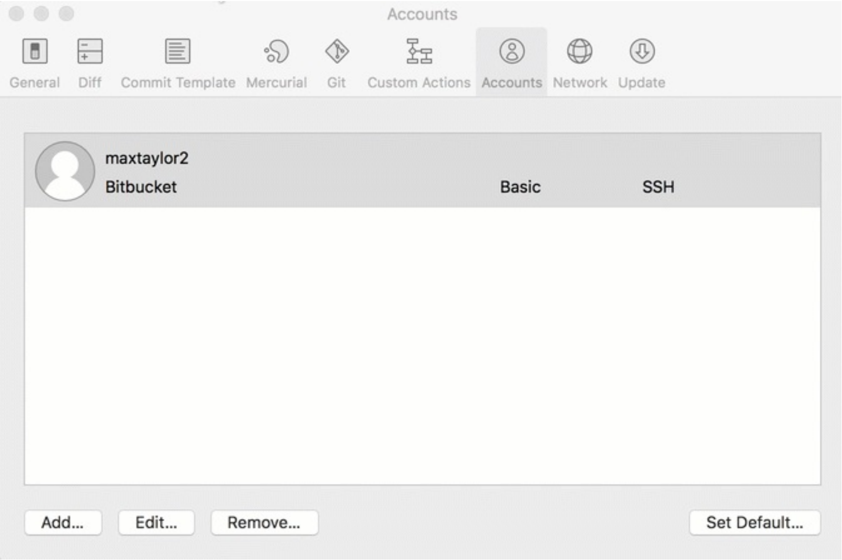
Step 1. Install Sourcetree and add your Bitbucket account

1. If you don't yet have Sourcetree, go to https://www.sourcetreeapp.com/ and click the **Download free** button.
2. Open the ZIP file to install Sourcetree. Refer to the Install Sourcetree page for more details.
3. If you don't connect your account during set up, you can add it from the **Accounts** tab by selecting **Preferences** from the **Sourcetree** menu.

Step 2. Create an SSH key

Follow these steps if you don't already have an SSH key for an account. If you do have an SSH key and you want to generate another key, you'll have to use the terminal because you can't use Sourcetree to create a second key.

Creating an SSH key looks something like this:



maxtaylor2 Bitbucket	Basic	SSH

1. From the **Sourcetree** menu, select **Preferences**.
2. Click the **Accounts** tab, select the account where you want to add the SSH key and click **Edit**.
3. Change the **Protocol** to **SSH** if it's not already selected.

4. Hold down the key on your keyboard to see the **Generate Key** button.

   If you've already generated an SSH key for this account from Sourcetree, the OPTION key won't do anything. Use your existing key or generate another key from the terminal.

5. Click**Generate Key**.
6. **PassphraseConfirm Passphrase**
7. Click **Create**.

Step 3. (Mercurial only) Enable SSH compression

Enabling SSH compression is recommended but not required. You'll need to clone and open a Mercurial repository to complete these steps.

By default, Git automatically performs compression when sending or retrieving data, but Mercurial doesn't. Enabling SSH compression can speed up sending and retrieving data, drastically in some cases.

To enable SSH compression:

1. Click **Settings** in the top right of the repository window.
   The repository settings may open to the **Remotes** tab. If not, click the **Remotes** tab.
2. Click **Edit Config File** to open the Mercurial global configuration file (`~/.hgrc`).
3. Add line to the UI section:

   ```
   ssh = ssh -C
   ```

   When you're done the file should look similar to this:

   ```
   [ui]
   # name and email (local to this repository, optional), e.g.
   username = Emma <emmap1@atlassian.com>
   ssh = ssh -C
   ```

4. Save and close the file.

Step 4. Add the public key to your Bitbucket settings

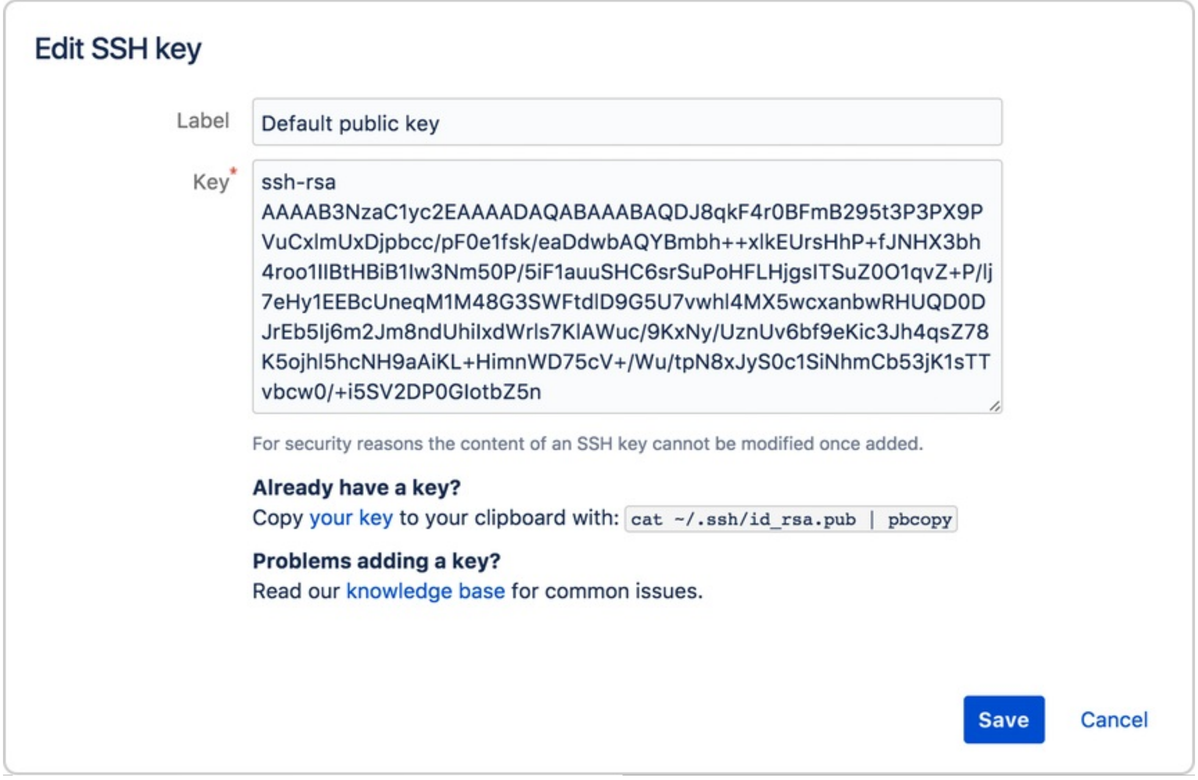
1. From Bitbucket, choose **Bitbucket settings** from your avatar in the lower left.
   The **Account settings** page opens.
2. Click **SSH keys**.
   If you've already added keys, you'll see them on this page.
3. Select your account from your **Accounts** tab in Sourcetree.

4. Click the **Copy to Clipboard** button to copy your public SSH key.

5. From Bitbucket, click **Add key**.

6. Enter a **Label** for your new key, for example, `Default public key`.

7. Paste the copied public key into the SSH **Key** field.



Edit SSH key

Label: Default public key

Key*:

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAABAQDJ8qkF4r0BFmB295t3P3PX9P
VuCxlmUxDjpbcc/pF0e1fsk/eaDdwbAQYBmbh++xlkEUrsHhP+fJNHX3bh
4roo1IlBtHBiB1lw3Nm50P/5iF1auuSHC6srSuPoHFLHjgsITSuZ0O1qvZ+P/lj
7eHy1EEBcUneqM1M48G3SWFtdlD9G5U7vwhl4MX5wcxanbwRHUQD0D
JrEb5lj6m2Jm8ndUhilxdWrls7KlAWuc/9KxNy/UznUv6bf9eKic3Jh4qsZ78
K5ojhl5hcNH9aAiKL+HimnWD75cV+/Wu/tpN8xJyS0c1SiNhmCb53jK1sTT
vbcw0/+i5SV2DP0GlotbZ5n
```

For security reasons the content of an SSH key cannot be modified once added.

Already have a key?
Copy your key to your clipboard with: `cat ~/.ssh/id_rsa.pub | pbcopy`

Problems adding a key?
Read our knowledge base for common issues.

8. Click **Save**.
   Bitbucket sends you an email to confirm the addition of the key.

   ⓘ
   Edit an SSH key

   After you add a key, you can edit the key's **Label** but not the key itself.

Now that you've got an SSH key set up, use the SSH URL the next time you clone a repository. If you already have a repository that you cloned over HTTPS, change the remote URL for your repository to its SSH URL.