[SSH] Generate SSH Key

`Ssh-keygen` is a tool for creating new authentication key pairs for SSH. Such key pairs are used for automating logins, single sign-on, and for authenticating hosts.

Creating an SSH key on Windows

1. Check for existing SSH keys

You should check for existing SSH keys on your local computer. *You can use an existing SSH key with remote Server if you want.*

Open a command prompt, and run:

```
cd %userprofile%/.ssh
```

- If you see "No such file or directory", then there aren't any existing keys: ***go to step 3***.
- Check to see if you have a key already:

  ```
  dir id_*
  ```

  If there are existing keys, you may want to use those.

2. Back up old SSH keys

If you have existing SSH keys, but you don't want to use them when connecting to remote Server, you should back those up.

In a command prompt on your local computer, run:

```
mkdir key_backup
copy id_rsa* key_backup
```

3. Generate a new SSH key

If you don't have an existing SSH key that you wish to use, generate one as follows:

1. Log in to your local computer as an administrator.
2. In a command prompt, run:

   ```
   ssh-keygen -t rsa -C "your_email@example.com"
   ```

   Associating the key with your email address helps you to identify the key later on.

   Note that the `ssh-keygen` command is only available if you have already installed Git (with Git Bash).
   You'll see a response similar to this:

   blocked URL
3. Just press <Enter> to accept the default location and file name. If the `.ssh` directory doesn't exist, the system creates one for you.
4. Enter, and re-enter, a passphrase when prompted. The whole interaction will look similar to this:
   blocked URL
5. You're done!

Creating an SSH key on Linux & macOS

1. Check for existing SSH keys

You should check for existing SSH keys on your local computer. *You can use an existing SSH key with remote Server if you want.*

Open a terminal and run the following:

```
cd ~/.ssh
```

- If you see "No such file or directory, then there aren't any existing keys: ***go to step 3***.
- Check to see if you have a key already:

  ```
  ls id_*
  ```

- If there are existing keys, you may want to use them.

2. Back up old SSH keys

If you have existing SSH keys, but you don't want to use them when connecting to Bitbucket Server, you should back those up.

Do this in a terminal on your local computer, by running:

```
mkdir key_backup
cp id_rsa* key_backup
```

3. Generate a new key

If you don't have an existing SSH key that you wish to use, generate one as follows:

1. Open a terminal on your local computer and enter the following:

   ```
   ssh-keygen -t rsa -C "your_email@example.com"
   ```

   Associating the key with your email address helps you to identify the key later on.

   You'll see a response similar to this:

   blocked URL
2. Just press <Enter> to accept the default location and file name. If the `.ssh` directory doesn't exist, the system creates one for you.
3. Enter, and re-enter, a passphrase when prompted.
   The whole interaction will look similar to this:

   blocked URL

4. You're done!

Related articles

- Creating Branches for Projects with Upcoming Releases
- [Gerrit - GitHub] Update repo committer rights
- [CI - Jenkins] Update Jenkins jobs
- [Artifactory] Sync Bintray to JCenter/Maven-Central
- [Artifactory] Sync Artifacts from Bintray to Maven Central