# Check for existing SSH key pair

Before generating a new SSH key pair first check if you already have an SSH key on your client machine because you don’t want to overwrite your existing keys.

**# ls -al ~/.ssh/id\_\*.pub**

Generate a new SSH key pair

The following command will generate a new 4096 bits SSH key pair with your email address as a comment:

**# ssh-keygen -t rsa -b 4096 -C** [**your\_email@domain.com**](mailto:your_email@domain.com)

Press Enter to accept the default file location and file name:

Enter file in which to save the key (/home/yourusername/.ssh/id\_rsa):

Next, the ssh-keygen tool will ask you to type a secure passphrase.

Whether you want to use passphrase it’s up to you, if you choose to use passphrase you will get an extra layer of security. In most cases, developers and system administrators use SSH without a passphrase because they are useful for fully automated processes. If you don’t want to use a passphrase just press Enter.

Enter passphrase (empty for no passphrase):

To be sure that the SSH keys are generated you can list your new private and public keys with:

**# ls ~/.ssh/id\_\***

output

/home/yourusername/.ssh/id\_rsa /home/yourusername/.ssh/id\_rsa.pub

# Copy the public key

Now that you have generated an SSH key pair, in order to be able to login to your server without a password you need to copy the public key to the server you want to manage.

The easiest way to copy your public key to your server is to use a command called ssh-copy-id. On your local machine terminal type:

**#ssh-copy-id remote\_username@server\_ip\_address**

In order to copy some specific file

**#ssh-copy-id -f -i /path-to-pub-file remote\_username@server\_ip\_address**

You will be prompted to enter the remote\_username password:

**# remote\_username@server\_ip\_address's password:**

Once the user is authenticated, the public key will be appended to the remote user authorized\_keys file and connection will be closed.

If by some reason the ssh-copy-id utility is not available on your local computer you can use the following command to copy the public key:

**# cat ~/.ssh/id\_rsa.pub | ssh remote\_username@server\_ip\_address "mkdir -p ~/.ssh && chmod 700 ~/.ssh && cat >> ~/.ssh/authorized\_keys && chmod 600 ~/.ssh/authorized\_keys"**

# Login to your server using SSH keys

After completing the steps above you should be able log in to the remote server without being prompted for a password.

To test it just try to login to your server via SSH:

**# ssh remote\_username@server\_ip\_address**

If everything went well, you will be logged in immediately

# No need to pass username for using ssh command

It is possible to set username for a certain hostname when we use ssh command. So it is no longer needed to write username and int enough to write:

**# ssh remote\_host**

We have to set username for that specific hostname in .ssh/config file like below file:

Host remote\_host

Hostname <remote\_host>

User <username>

IdentityFile ~/.ssh/id\_rsa

ForwardAgent yes