Linux: Manually Installing SSH Keys

Here's some steps on how to manually install SSH public keys in a host.

1. SSH Key Folder

Navigate to the home folder for the user, with the following command:

```
cd /home/username
```

Check if the .ssh folder exists (it is hidden, requiring the -a switch):

```
ls -al
```

If the .ssh folder does not exist, create it with the following commands (from the user folder):

```
sudo mkdir ~/.ssh
```

```
sudo chmod 0700  ./.ssh
```

2. SSH Key File

Enter the ssh key folder with:

```
cd ./.ssh
```

And, check if any key files are there, with:

```
ls -l
```

```
glwhite@jumpvm:~$ cd ./.ssh
glwhite@jumpvm:~/.ssh$ ls -l
total 4
-rw-r--r-- 1 root root 381 Apr 27 23:57 authorized_keys
glwhite@jumpvm:~/.ssh$
```

Make sure the key file has the same name that was defined in the ssh config file, in previous steps.

If no key file, create one with (making sure to use the correct key file name):

```
sudo touch ./authorized_keys
```

Set permissions on the key file:

```
sudo chmod 600 ./authorized_keys
```

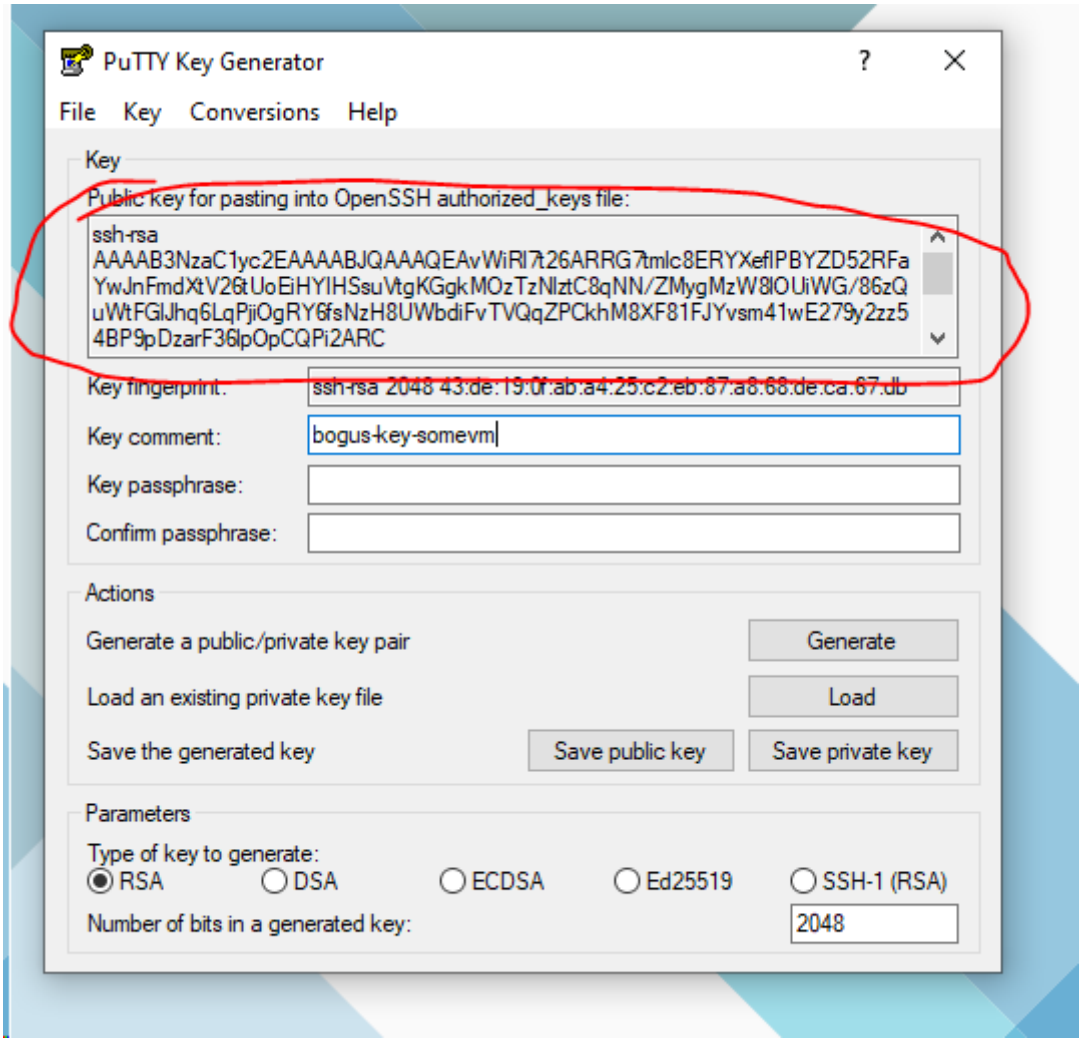
From Windows

This section is for copying the public key string from a Windows, host.

The tricks to successfully pasting in an SSH key to the ssh key file are:

- Always paste the key string as a single line
- The key string must being with, “ssh-rsa“
- Strip out any “Begin SSH2 PUBLIC KEY” and ending
- The key line should contain the key comment at the end of the line, for easy identification
- A key string should be of the form: ssh-rsa [really long base64 key string here] [key comment]
- Single whitespace is needed between each component of the key line
- The key comment must have no whitespace in it

The easiest way to get this string is to load a key in PuttyGen.
Then, paste the entire key string directly from the text window of the form, like this:



From Linux

If you do not have password-based SSH access to your server available, you will have to complete the above process manually.

We will manually append the content of your `id_rsa.pub` file to the `~/.ssh/authorized_keys` file on your remote machine.

To display the content of your `id_rsa.pub` key, type this into your local computer:

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

You will see the key's content, which should look something like this:

```
Outputssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAACAQCqqI6MzstZYh1TmWWv11q5O3pISj2ZFl9HgH1JLknLLx44+tXfJ7mIrKNxO
OwxIxvcBF8PXSYvobFYEZjGIVCEAjrUzLiIxbyCoxVyle7Q+bqgZ8SeeM8wzytsY+dVGcBxF6N4JS+zVk5eMcV385gG3
Y6ON3EG112n6d+SMXY0OEBIcO6x+PnUSGHrSgpBgX7Ks1r7xqFa7heJLLt2wWwkARptX7udSq05paBhcpB0pHtA1
Rfz3K2B+ZVIpSDfki9UVKzT8JUmwW6NNzSgxUfQHGwnW7kj4jp4AT0VZk3ADw497M2G/12N0PPB5CnhHf7ovgy6nL
1ikrygTKRFmNZISvAcywB9GVqNAVE+ZHDSCuURNsAInVzgYo9xgJDW8wUw2o8U77+xiFxgI5QSZX3Iq7YLMgeksaO
```

```
4rBJEa54k8m5wEiEE1nUhLuJ0X/vh2xPff6SQ1BL/zkOhvJCACK6Vb15mDOeCSq54Cr7kvS46itMosi/uS66+PujOO+xt/
2FWYepz6ZlN70bRly57Q06J+ZJoc9FfBCbCyYH7U/ASsmY095ywPsBo1XQ9PqhnN1/YOorJ068foQDNVpm146mUpIL
Vxmq41Cj55YKHEazXGsdBIbXWhcrRf4G2fJLRcGUr9q8/lERo9oxRm5JFX6TCmj6kmiFqv+Ow9gI0x8GvaQ==
demo@test
```

Access your remote host using whichever method you have available.

Once you have access to your account on the remote server, you should make sure the `~/.ssh` directory exists. This command will create the directory if necessary, or do nothing if it already exists:

```
mkdir -p ~/.ssh
```

Now, you can create or modify the `authorized_keys` file within this directory. You can add the contents of your `id_rsa.pub` file to the end of the `authorized_keys` file, creating it if necessary, using this command:

```
echo public_key_string >> ~/.ssh/authorized_keys
```

In the above command, substitute the `public_key_string` with the output from the `cat ~/.ssh/id_rsa.pub` command that you executed on your local system. It should start with `ssh-rsa AAAA...`.

Finally, we'll ensure that the `~/.ssh` directory and `authorized_keys` file have the appropriate permissions set:

```
chmod -R go= ~/.ssh
```

This recursively removes all "group" and "other" permissions for the `~/.ssh/` directory.

If you're using the **root** account to set up keys for a user account, it's also important that the `~/.ssh` directory belongs to the user and not to **root**:

```
chown -R sammy:sammy ~/.ssh
```

> NOTE: The above example uses sammy as the username. Change this to the appropriate username for the target account.

Connecting to SSH Server from Windows

See this page for steps on how to connect to a Linux host from Windows: Connecting to SSH Server from Windows

---

Revision #4
Created 1 September 2025 06:20:01 by glwhite
Updated 7 May 2026 05:29:17 by glwhite