Creating SSH Keys in Linux

Creating an SSH key is straightforward on a linux client, using this command:

```
ssh-keygen
```

By default recent versions of `ssh-keygen` will create a 3072-bit RSA key pair, which is secure enough for most use cases (you may optionally pass in the `-b 4096` flag to create a larger 4096-bit key).

> NOTE: See this page for naming conventions, before actually creating a new key: SSH Key Naming Convention

Also. The command allows you to specify the key type (-t), key size (-b), key comment (-C), and output filename (-f) like this:

```
ssh-keygen -t rsa -b 4096 -C "rsa4096-glwhite-hadron-20230913" -f ./rsa4096-glwhite-hadron-20230913.key
```

Here are some different type and size combinations:

```
# Create an RSA key with key size of 4096 bits...
ssh-keygen -t rsa -b 4096

# Create a DSA key...
# This is an old government key type (Digital Signature Algorithm), defaulting to a 1024 bit size.
ssh-keygen -t dsa

# Create an ECDSA521 key...
ssh-keygen -t ecdsa -b 521

# Create an ED25519 key...
# NOTE: This type is not widely accepted, yet.
ssh-keygen -t ed25519
```

If you ran the command without specifying a filename (-f), it will prompt you for the output location:

```
OutputGenerating public/private rsa key pair.
Enter file in which to save the key (/your_home/.ssh/id_rsa):
```

You can accept the default path (pressing enter), to save the key pair into the `.ssh/` subdirectory in your home directory, or specify an alternate path.

If a key already exists at the path, you may see the following prompt:

```
Output/home/your_home/.ssh/id_rsa already exists.
Overwrite (y/n)?
```

> WARNING: If you choose to overwrite the key on disk, you will **not** be able to authenticate using the previous key anymore. Be very careful when selecting yes, as this is a destructive process that cannot be reversed.

You should then see the following prompt:

```
OutputEnter passphrase (empty for no passphrase):
```

Here you optionally may enter a secure passphrase, which is highly recommended. A passphrase adds an additional layer of security to prevent unauthorized users from logging in. To learn more about security, consult our tutorial on How To Configure SSH Key-Based Authentication on a Linux Server.

You should then see the output similar to the following:

```
OutputYour identification has been saved in /your_home/.ssh/id_rsa
Your public key has been saved in /your_home/.ssh/id_rsa.pub
The key fingerprint is:
SHA256:/hk7MJ5n5aiqdfTVUZr+2Qt+qCiS7BIm5Iv0dxrc3ks user@host
The key's randomart image is:
+---[RSA 3072]----+
|              .|
|             + |
|            +  |
| .          o . |
|o       S   . o  |
| + o. .oo. ..  .o|
|o = oooooEo+ ...o|
|.. o *o+=.*+o....|
|    =+=ooB=o.... |
```

```
+----[SHA256]-----+
```

Revision #4
Created 10 April 2025 05:08:43 by glwhite
Updated 1 August 2025 08:54:41 by glwhite