Adding SSH Keys with ssh-copy-id

If you are attempting to add an SSH key (to a Linux host), from another Linux host, you can use a built-in utility called, `ssh-copy-id` .

The `ssh-copy-id` tool is included by default in many operating systems, so you may have it available on your local system. For this method to work, you must already have password-based SSH access to your server.

Due to its simplicity, this method is highly recommended if available. If you do not have `ssh-copy-id` available to you on your client machine, you may use one of the two alternate methods provided in this section (copying via password-based SSH, or manually copying the key).

> NOTE: Using ssh-copy-id requires password authentication enabled.
> See the bottom of this page for how to temporarily enable password auth on the remote host.

Requires Password Auth

This utility is easy to use, before disabling password authentication.
But, it does encounter issues, trying to copy a public key to a remote host, while simultaneously logging into it with a different key.
That quite often fails.
So, first, we will ensure that password authentication is enabled on the remote host.

Log into the remote host, and open the sshd_config file, located at: /etc/ssh/sshd_config.

Locate the line with 'PasswordAuthentication', and set it to yes, like this:

```
...
PasswordAuthentication yes
...
```

Save and close the config file.

Restart the ssh service, with this:

```
sudo systemctl restart ssh
```

Now, you can use the utility from the local linux VM.

SSH-Copy-Id Usage

To use the utility, you specify the remote host that you would like to connect to, and the user account that you have password-based SSH access to.

> NOTE: The account you log in to the local Linux host with, will also be the account, whose public SSH key will be pushed to the remove Linux host. So, this will be the account to which your public SSH key will be copied.

The abbreviated syntax is:

```
ssh-copy-id username@remote_host
```

You can also specify the public key file and port, like this:

```
ssh-copy-id -i ~/.ssh/id_rsa.pub USER@HOST -p PORT
```

You may see the following message:

```
OutputThe authenticity of host '203.0.113.1 (203.0.113.1)' can't be established.
ECDSA key fingerprint is fd:fd:d4:f9:77:fe:73:84:e1:55:00:ad:d6:6d:22:fe.
Are you sure you want to continue connecting (yes/no)? yes
```

This means that your local computer does not recognize the remote host. This will happen the first time you connect to a new host. Type “yes” and press `ENTER` to continue.

Next, the utility will scan your local account for the `id_rsa.pub` key that we created earlier. When it finds the key, it will prompt you for the password of the remote user’s account:

```
Output/usr/bin/ssh-copy-id: INFO: attempting to log in with the new key(s), to filter out any that are already installed
/usr/bin/ssh-copy-id: INFO: 1 key(s) remain to be installed -- if you are prompted now it is to install the new keys
username@203.0.113.1's password:
```

Type in the password (your typing will not be displayed, for security purposes) and press `ENTER`.

The utility will connect to the account on the remote host using the password you provided.

It will then copy the contents of your `~/.ssh/id_rsa.pub` key into a file in the remote account's home `~/.ssh` directory called `authorized_keys`.

You should see the following output:

```
OutputNumber of key(s) added: 1


Now try logging into the machine, with:   "ssh 'username@203.0.113.1'"
and check to make sure that only the key(s) you wanted were added.
```

Disable SSH Password Auth

Once you have pushed the SSH key to the remote host, you need to disable password authentication.

From a terminal session with the remote host, open the sshd_config file (same as earlier), located at: /etc/ssh/sshd_config.

Locate the line with 'PasswordAuthentication', and set it to no, like this:

```
...
PasswordAuthentication no
...
```

Save and close the config file.

Restart the ssh service, with this:

```
sudo systemctl restart ssh
```

Now, you have added the public SSH key of your local linux user account, to the remote Linux VM. And, you should have SSH key authenticated access to the remote Linux host.

Revision #4
Created 1 September 2025 07:19:43 by glwhite
Updated 1 September 2025 07:42:11 by glwhite