SSH login without password

Your aim

You want to use Linux and OpenSSH to automate your tasks. Therefore you need an **automatic** login from host A / user a to Host B / user b. You don't want to enter any passwords, because you want to call `ssh` from a within a shell script.

How to do it

First log in on A as user a and generate a pair of authentication keys. Do not enter a passphrase:

```
a@A:~> ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/a/.ssh/id_rsa):
Created directory '/home/a/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/a/.ssh/id_rsa.
Your public key has been saved in /home/a/.ssh/id_rsa.pub.
The key fingerprint is:
3e:4f:05:79:3a:9f:96:7c:3b:ad:e9:58:37:bc:37:e4 a@A
```

Now use `ssh` to create a directory `~/.ssh` as user b on B. (The directory may already exist, which is fine):

```
a@A:~> ssh b@B mkdir -p .ssh
b@B's password:
```

Finally append a's new public key to `b@B:.ssh/authorized_keys` and enter b's password one last time:

```
a@A:~> cat .ssh/id_rsa.pub | ssh b@B 'cat >> .ssh/authorized_keys'
b@B's password:
```

From now on you can log into B as b from A as a without password:

```
a@A:~> ssh b@B
```

A note from one of our readers: Depending on your version of SSH you might also have to do the following changes:

- Put the public key in `.ssh/authorized_keys2`
- Change the permissions of `.ssh` to `700`
- Change the permissions of `.ssh/authorized_keys2` to `640`

Keywords: ssh openssh password **Author: Mathias Kettner**