Setup SSH Keys

With the implemenation of two-factor authentication on the ssh protocol at the Observatory, you need to setup two ssh keys to make life easy. These two key are:

1. To login from your laptop to an Observatory machine
2. To login between computers at the Observatory

Below we deal with these two cases.

Login from outside the Observatory

Login from the internet is usually done from your own personal computer. Of course that is a MacBook, but for all those 'other system' users we describe belog how to setup a private/public key pair to allow seemless lgon to the Observatory computers.

From Windows

For Windows, you can use `putty`, `MobaXterm` or `Bitvise Tunnelier` to open a terminal session to a Linux desktop or server computer. Below we describe the seutp for each program separately:

- Setup putty
- Setup MobaXterm
- Setup Bitvise Tunnelier

From MacOS

From Linux

setup ssh for key based login

We need to create a private/public key set to allow passwordless login via ssh. To do this run the sshkey-ge command:

```
$ ssh-keygen -t ecdsa
Generating public/private ecdsa key pair.
Enter file in which to save the key (/home/testuser1/.ssh/id_ecdsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/testuser1/.ssh/id_ecdsa
Your public key has been saved in /home/testuser1/.ssh/id_ecdsa.pub
The key fingerprint is:
SHA256:xb4Rs37UbXt3Wn5cHkdKWy2ZDBbor9F83IYNLhjsfIU
testuser1@bree.strw.leidenuniv.nl
The key's randomart image is:
```

```
+---[ECDSA 256]---+
|           ...   |
|         .. o    |
|         o=. + o.|
|         o++E.O.+|
|        So+*.=.@o|
|         .=+* BoB|
|          o+.o =O|
|          ..   +B|
|               . o|
+----[SHA256]-----+
```

For both question about passphrase, just hit enter (we will not be using passphrases). This will also have generated two files in your personal .ssh directory:

```
$ ls -ltr id_ecdsa*
-rw------- 1 testuser1 users 537 Mar 22 12:13 id_ecdsa
-rw-r--r-- 1 testuser1 users 195 Mar 22 12:13 id_ecdsa.pub
```

The file id_rsa.pub must be transferred to the remote host. For this we can use ssh-copy-id:

```
$ ssh-copy-id -i ~/.ssh/id_edcsa.pub username@remote-host
```

This may produce the following message:

```
/usr/bin/ssh-copy-id: INFO: Source of key(s) to be installed:
"/home/username/.ssh/id_edcsa.pub"
The authenticity of host 'remote-host (123.123.123.123)' can't be
established.
ECDSA key fingerprint is SHA256:tygMarTe3SOjTcY9HzldKThxQzsTeiYHg5JmjB2bxeg.
Are you sure you want to continue connecting (yes/no)? yes
```

Having confirmed the access key to remote-host, the copy operation will commence:

```
/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@remote-host's password:
```

Type your password to actually start the file copy.

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

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

Login between computers at the Observatory