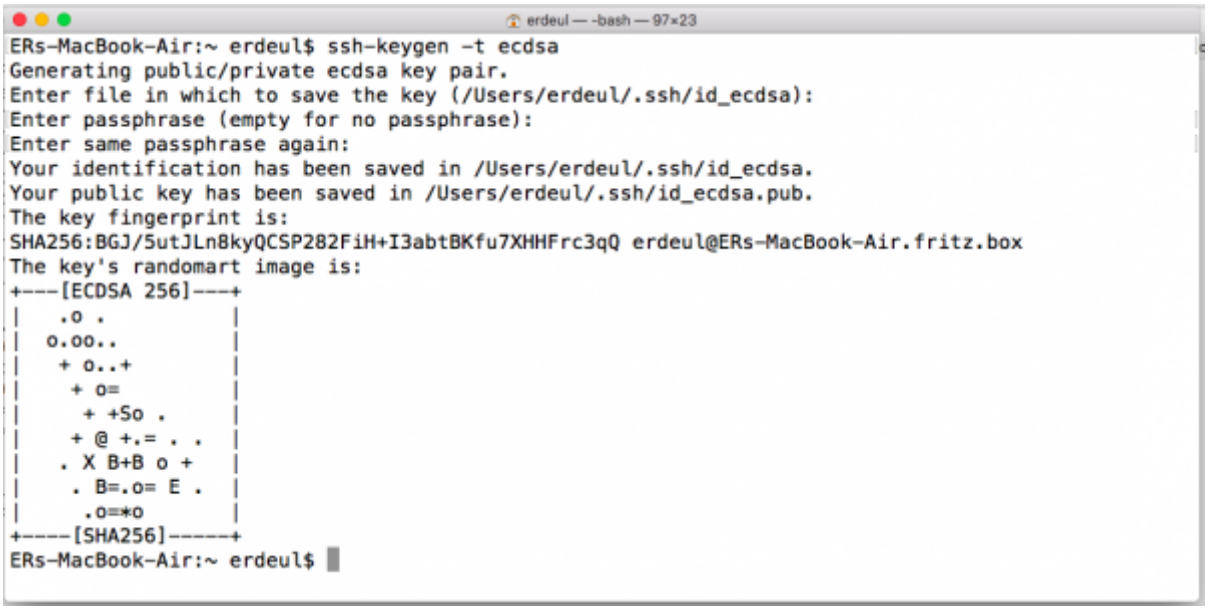
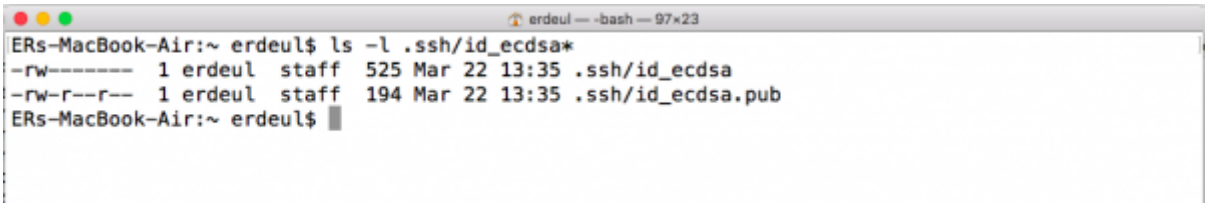
Setup key based login from MacOS

The procedure is very similar to the Linux procedure. So we first need to build a public/private keypair using the ssh-keygen utility:



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:



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

```
$ ssh-copy-id -i ~/.ssh/id_ecdsa.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_rsa.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:
One-time password (OATH) for `username`:
```

Type your password (and the 2FA passcode) 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.
```

The passwordless/2fa codeless ssh login is now in place.