

# Secure the server

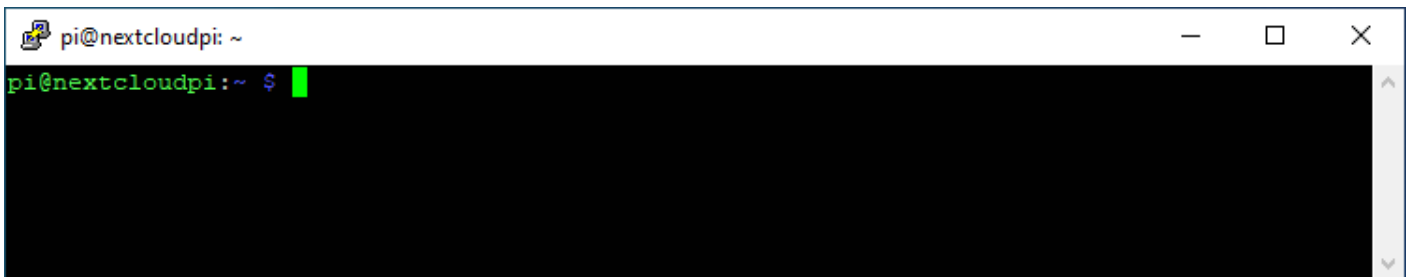
To make the server more secure we will

1. change the default ssh port and
2. we change from password authentication to key files.

1.

The default ssh port is 22. We change this to a random port: here we take the port 4200.

- connect to your server using PuTTY



- type in:

```
sudo nano /etc/ssh/sshd_config
```

- search for

```
#Port 22  
#AddressFamily any
```

- change it to

```
Port 4200  
#AddressFamily any
```

here you can choose any port you like and is not yet used by another program

- save and exit (STRG+O ; STRG+X)

- retart the sshd deamon

-type in

```
sudo systemctl restart sshd
```

2.

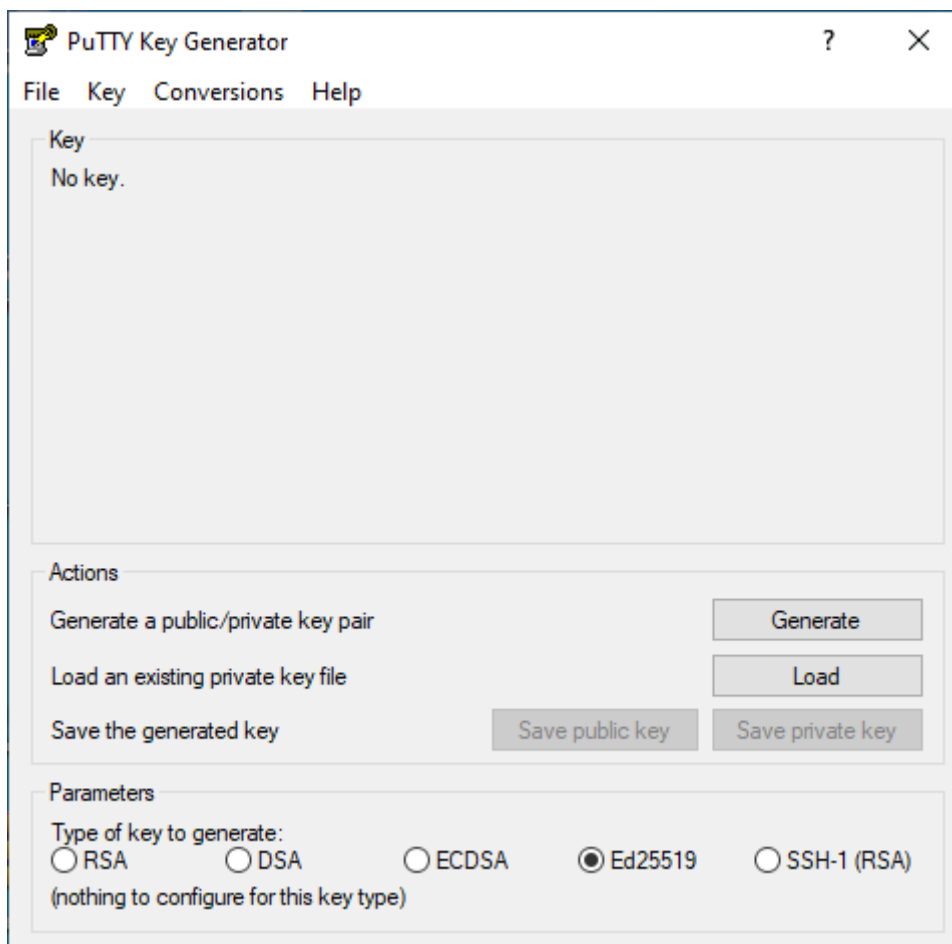
In the next step we will create a keyfile that include a long an encrypted password that we will use for authentication in stead of a normal passphrase.

Here i will show the way with a Windows PC and the program Putty. You can get it here:

<https://www.ssh.com/ssh/putty/download>

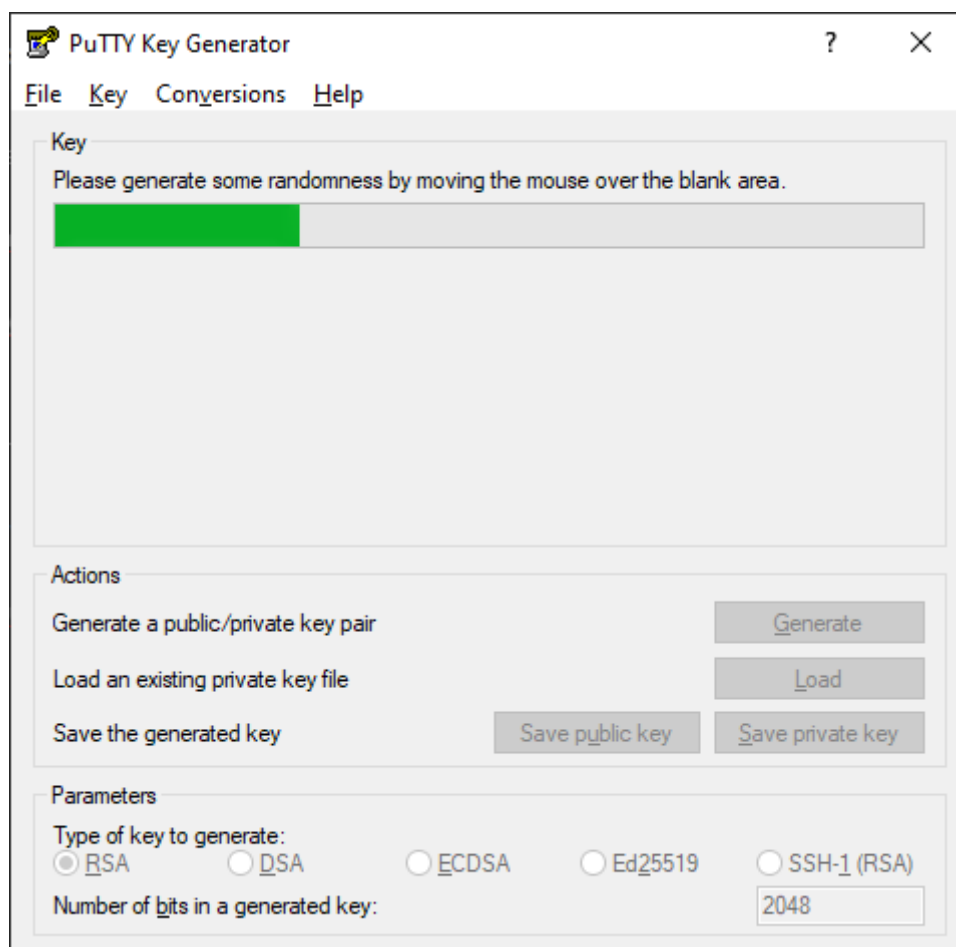
After downloading and installing Putty

- start PuTTYgen



- change Type of key to generate: from "RSA" to "ED25519"

- then klick Generate
- move the mouse over the blank area under the green bar



- replace the comment with something more specific for you connection; here it is the nextcloud server so i choose "nextcloud"

PuTTY Key Generator

File Key Conversions Help

Key

Public key for pasting into OpenSSH authorized\_keys file:

```
ssh-ed25519
AAAAC3NzaC1lZDI1NTE5AAAAIOT5OOY6KQpQ2nnSzINQQonp1yKw4iJRTZMCIYR
WgAjC nextcloud
```

Key fingerprint: ssh-ed25519 255 bc:86:c4:af:a4:7e:98:79:40:74:11:4e:ae:23:97:10

Key comment: nextcloud

Key passphrase:

Confirm passphrase:

Actions

Generate a public/private key pair Generate

Load an existing private key file Load

Save the generated key Save public key Save private key

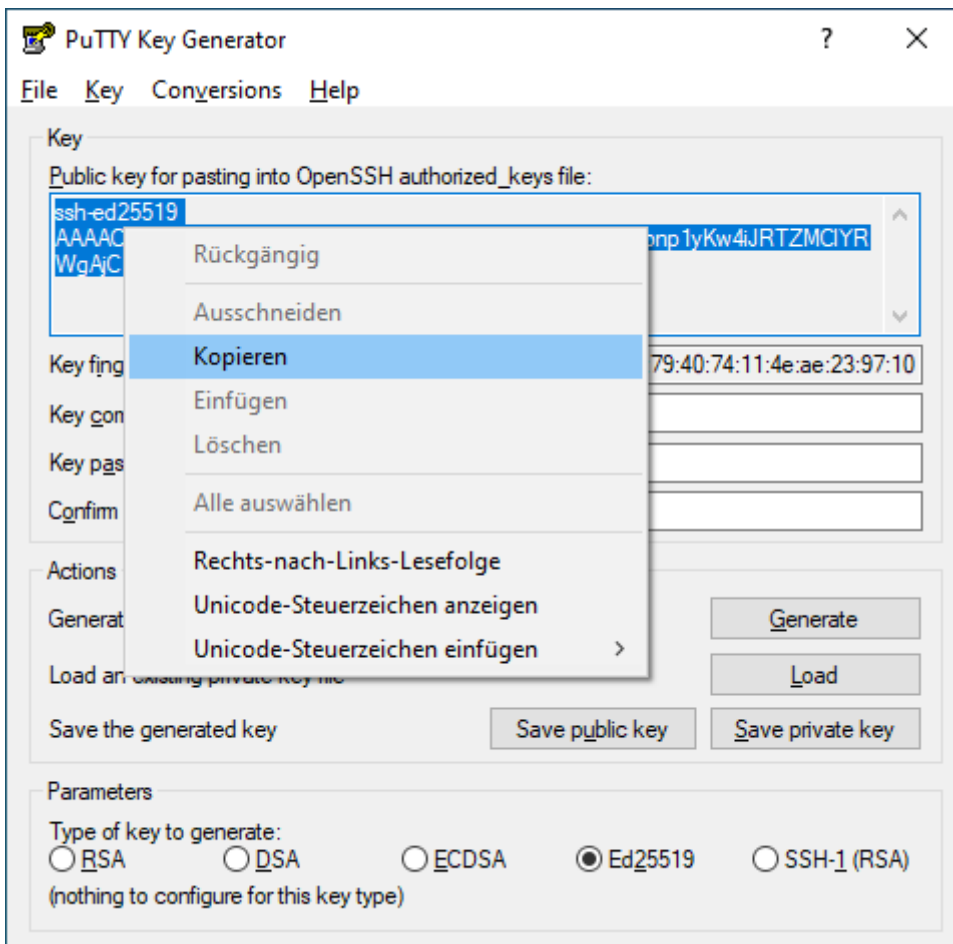
Parameters

Type of key to generate:

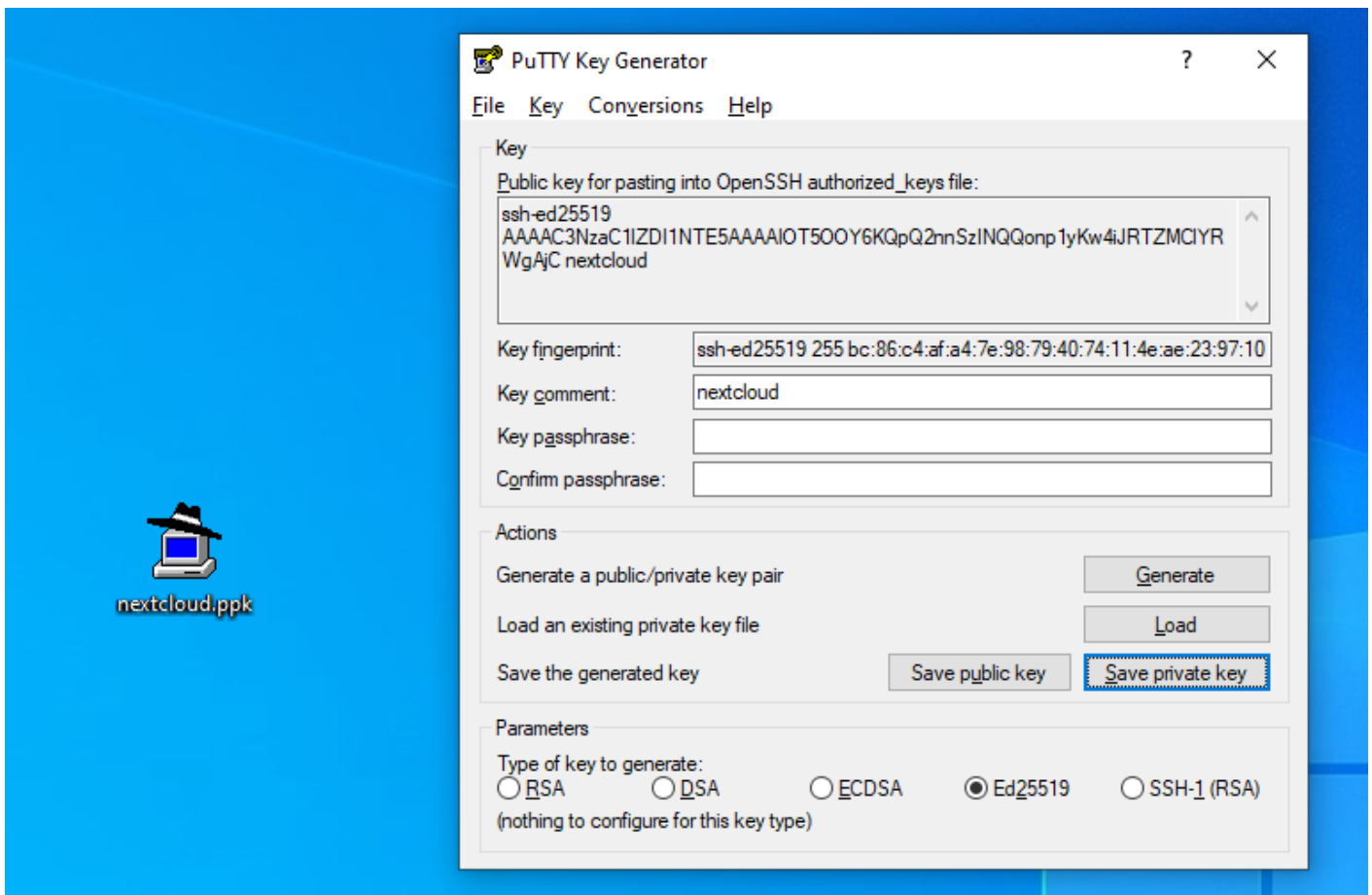
☐ RSA ☐ DSA ☐ ECDSA ☒ Ed25519 ☐ SSH-1 (RSA)

(nothing to configure for this key type)

- now copy the shown key

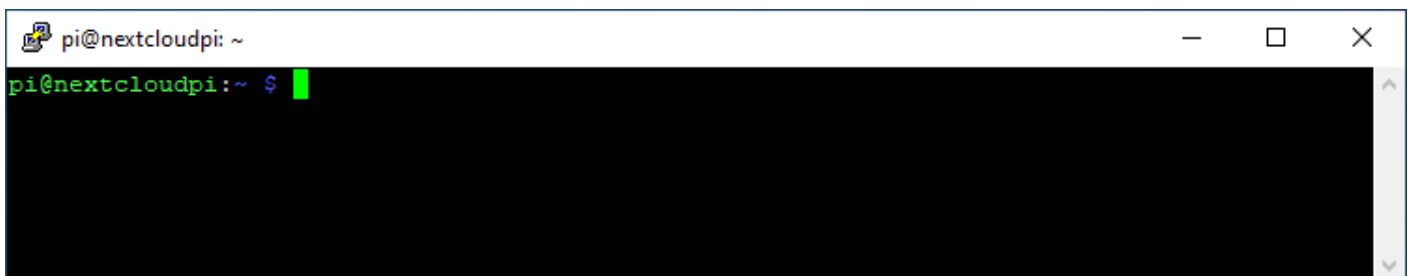


- save this in a simple text file on your computer for later
- click on "Save private key" and save it on your computer
- you will be asked for a passphrase, this is to protect your private key file. It's up to you if you want to.



- this .ppk file (here it is the nextcloud.ppk) is your secure private key and nobody else should have this. Please keep it safe!

- now log into your server you want to secure with this key using PuTTY



- look if there is already a hidden .ssh folder in your home directory type in:

```
ls -la
```

- if not, create one type in:

```
mkdir .ssh
```

- change into this directory, type in:

```
cd .ssh
```

- type in:

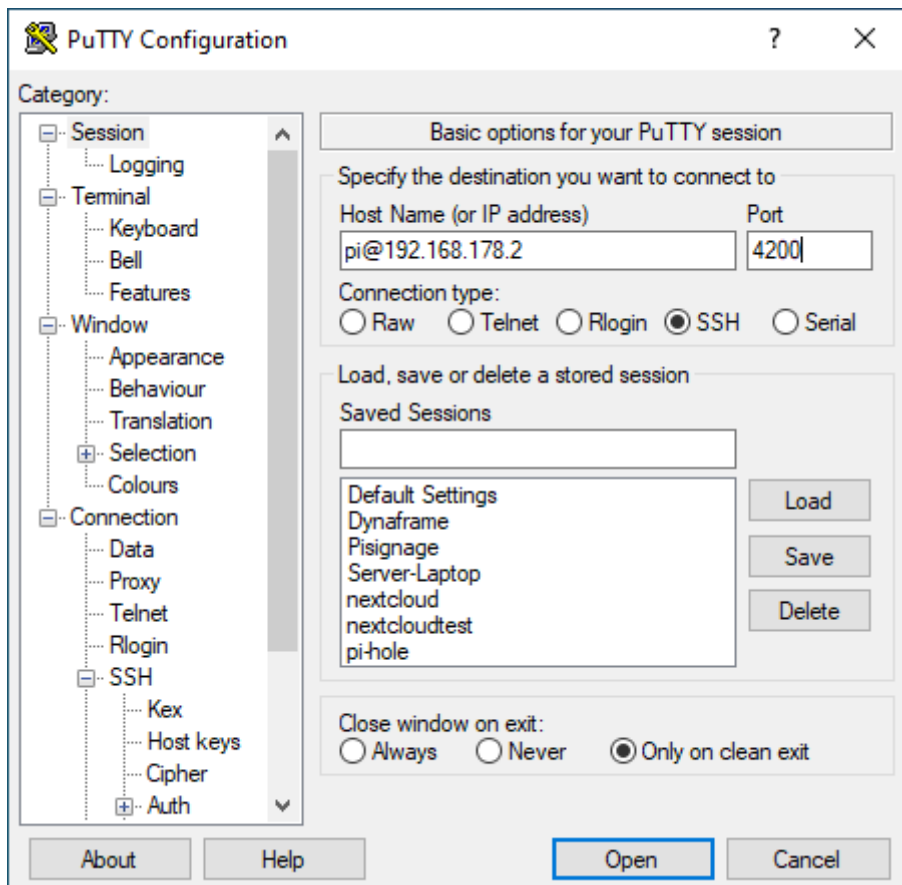
```
nano authorized_keys
```

- copy the key you saved in a text file earlier

- save and exit (STRG+O ; STRG+X)

- now you can connect via putty to your server using your .ppk key file

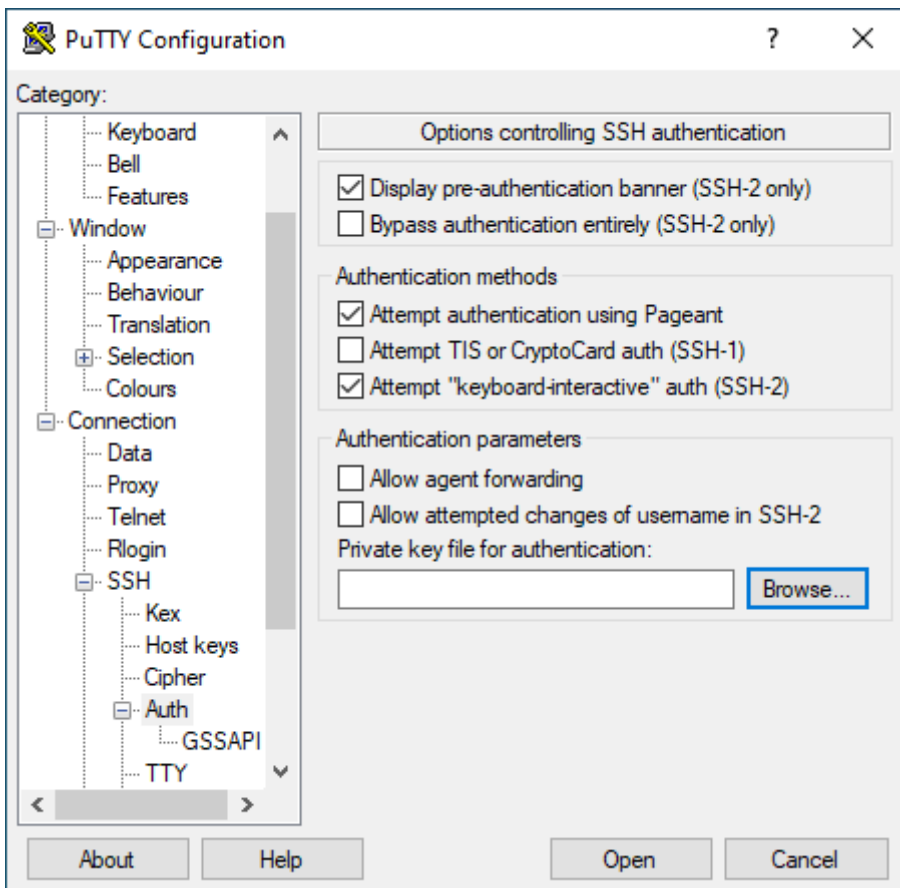
- open PuTTY



- type in your hostname and port

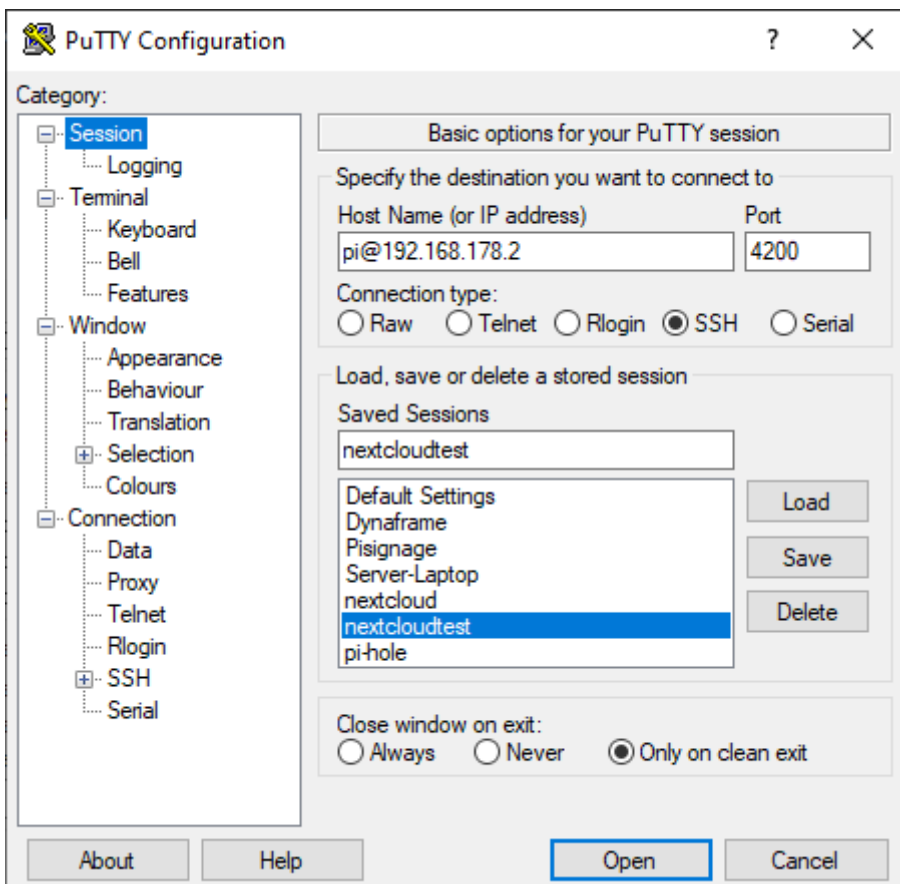
- on the left got to >"Connetion" >"SSH" >"Auth"

- click on "Browse"



- select your private key file (.ppk)

- go back to "Session" and under "Save Sessions" give it a name and click "Save"

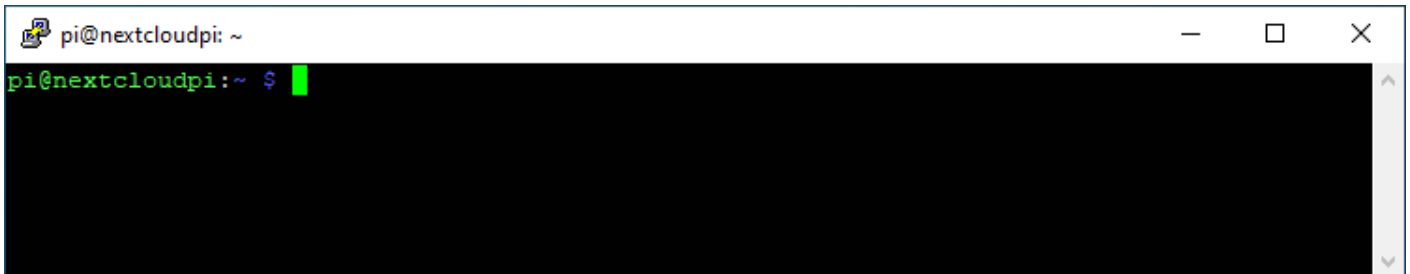




Now you have successfull saved your connection to your server with a key file.

When everything works fine and you can connect with your new keyfile it's time to disable the password authentication for ssh.

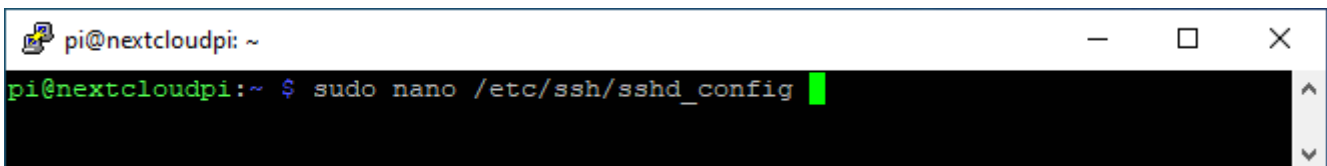
- connect to your server

A terminal window titled 'pi@nextcloudpi: ~' with standard window controls. The prompt 'pi@nextcloudpi:~ \$' is shown with a green cursor, indicating a successful connection to the server.

```
pi@nextcloudpi: ~  
pi@nextcloudpi:~ $
```

- type in:

```
sudo nano /etc/ssh/sshd_config
```

A terminal window titled 'pi@nextcloudpi: ~' showing the command 'sudo nano /etc/ssh/sshd\_config' entered at the prompt. A green cursor is at the end of the command.

```
pi@nextcloudpi: ~  
pi@nextcloudpi:~ $ sudo nano /etc/ssh/sshd_config
```

- search for:

```
#LoginGraceTime 2m  
#PermitRootLogin prohibit-password
```

- change it to

```
#LoginGraceTime 2m  
PermitRootLogin no
```

- this disables the possible login via the user: root

- next search for:

```
# To disable tunneled clear text passwords, change to no here!  
#PasswordAuthentication yes
```

- change it to

```
# To disable tunneled clear text passwords, change to no here!  
PasswordAuthentication no
```

- save and exit (STRG+O ; STRG+X)

- restart the ssh daemon;

- type in:

```
sudo systemctl restart sshd
```

Now your server allows no longer connections without a key file.

---

Revision #7

Created 11 March 2021 06:34:07 by Jan

Updated 26 November 2021 08:17:07 by Jan