

# Preparation for the git tutorial

## 1/7

- Since November 13<sup>th</sup>, 2020 Github no longer accepts usernames and passwords to upload your code from a cluster or your personal computer.  
<https://github.blog/2020-07-30-token-authentication-requirements-for-api-and-git-operations/>
- Among the suggested ways of logging in, there is **SSH keys**. I will teach you how to setup SSH keys for github. There are other methods but I do not think they suit this course.
  - For a detailed discussion of the SSH PKI technology and the commands read the MNXB01-manual.pdf . Here I will just show the practical commands to run for the tutorial.
  - For a detailed discussion why we suggest this method for this course read the dedicated slide about github command line access at the end of these slides.

# Preparation for the git tutorial

## 2/7

- Create an account on <https://github.com>
- To create an account click on the “Sign Up” button in the upper right corner.



# Preparation for the git tutorial

## 3/7

1) Login to Aurora (if possible...)

2) Generate a new SSH key pair with the following command:

```
ssh-keygen -b 4096 -f ~/.ssh/id_rsa_github
```

2.1) Choose a password for your private key.

**!!! NEVER GENERATE KEYS WITHOUT PASSWORDS.  
Read MNXB01-manual to understand why that is bad.**

3) Test that your password is working with this command:

```
ssh-keygen -y -f ~/.ssh/id_rsa_github
```

3.1) If the password didn't work, the program will say that you provided an "incorrect passphrase". Start again from step 2

4) Copy (ctrl +c ) and paste (ctrl+shift+v) this entire text below in your terminal to add to your ssh configuration information on how to login to github. You may need to press enter:

```
cat << EOF >> ~/.ssh/config

# Access to github
Host github.com
    HostName github.com
    IdentityFile ~/.ssh/id_rsa_github

EOF
```

5) check the ssh config file contains the lines added above.

```
cat ~/.ssh/config
```

# Preparation for the git tutorial

## 4/7

6) make sure the files have the proper permissions

```
chmod 600 ~/.ssh/id_rsa_github
chmod 644 ~/.ssh/id_rsa_github.pub
```

7) Check the contents of **your public key**:

```
cat ~/.ssh/id_rsa_github.pub
```

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQCAQCjVKDNRkMUMdEsY25jfXGCMhXL/57L
XsX5Re11cJ7mMq91tTpzhV+miedOwq30+KM5iPlPoN3QpJlZ26BBcrUJ/+pury7rN/W/
YfYMb+KOez74j8eT1gNNYfArZZmKfe9XMFB73XYyChmDZZkEz7UuGPYA2TdDKGBA4cg
9MqrvXsnM8FbLfnKHBsu2rrRH51tJM7VlMkWrGwHv9UAsndoDEtaj0qaF0SaQ8qz+CK55
o7HSBSIr1/0uQwgH+yOPbaJvKORfXTp7ewIw3xDpYDtGpP744ZI+q4Bzg67c4DixHfMN
2PDbLSM1AdrftTiaLMVePAHTdptVtfl1AWHmtikqLhPLzK3H342kMauXj9ne27wh2lMf
XFIWg8vzOo+fmidjSQ9hFvczMeaKikvkpL16BF3CCS8st5TmkpyOtRohYvAehY/dpsMVV
9exbpnEt8yU6XVx25qJiuUls0p1iXtJdqESrHgS9VqFGMq9Ew9W21mPT7JX92vXpUZ0T
6yvFDfvOOd1Yy8/23ECzdyqpQyk43LrSpX38ELA3K0+8ZN0mpB+c8mxwTA0I/dCnCeS
6iiCrOhP87CA8Wb5MScS7Q94z+T3jn3wAXbR/uUbTtXJE/klykknbINfB8xo9
3cII9GIv9UxRQSMKeBWRZdH9bIAi1xoRhpAgENpAgylKr6DSQ== pflorido@aurora1
```

On your terminal:

7.1) select **all the text** from `ssh-rsa` to `aurora1` (or whatever is in the end of your public key, it depends on the frontend node where you logged in)

7.2) copy the selection with `ctrl-shift-c` (or right-click and then select *copy*)

We will use it later.

# Preparation for the git tutorial

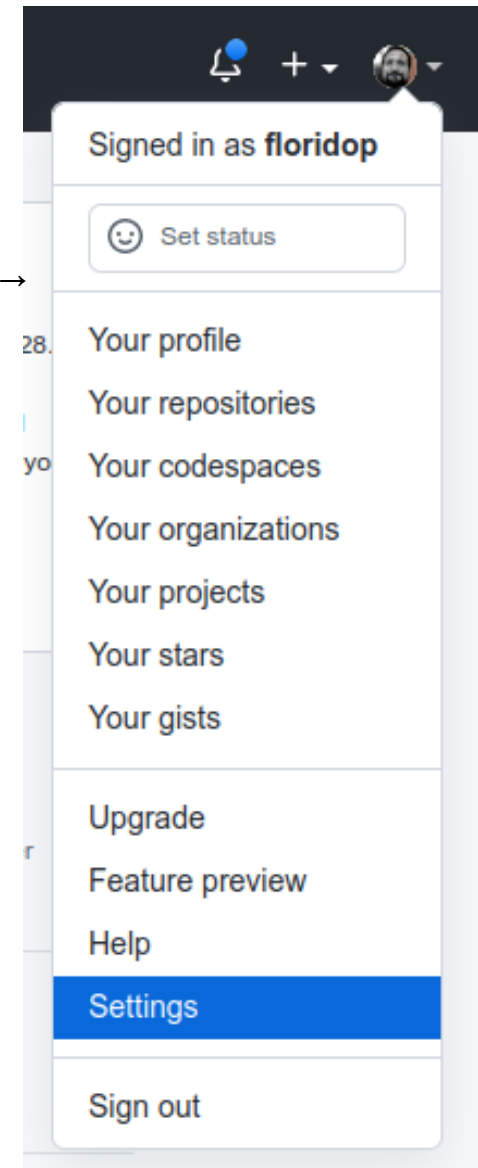
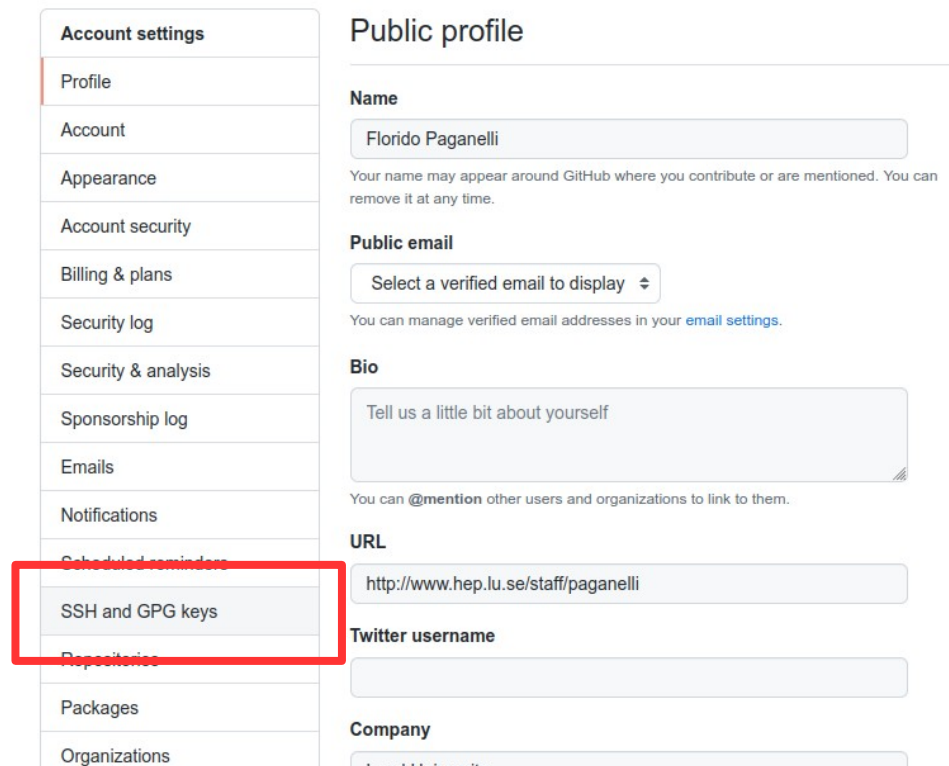
## 5/7

8) Sign in (login) on [www.github.com](https://www.github.com)



9) Click on your account avatar in the upper right corner and select “Settings” →

10) Click on “SSH and GPG Keys” in the lower left corner:



# Preparation for the git tutorial

## 6/7

11) Click on New SSH key

SSH keys

New SSH key

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.

12) Paste the key copied at step (7.2) in slide 4 in the **Key** field and add the name **Aurora** to the **Title** field as in the picture below.

- PASTE YOUR KEY NOT MINE OR I WILL BE ABLE TO ACCESS AND HACK YOUR ACCOUNT! :D

- DO NOT PASTE YOUR **PRIVATE KEY EVER!**

SSH keys / Add new

Title

Aurora

Key

```
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQCAQCjvKDNrkMUMdEsY25jfXGCMhXL/57LXsX5Re11cJ7mMq91tTpsz
V+miedOwq30+KM5iPIPoN3QpJIZ26BBcrUJ/+pury7rN
/W/YfYMB+K0ez74j8eT1gNNYfArZZmKfe9XMFB73XYyChmDZZkEz7UuGPYA2TdDKGBA4cg9MqrvXsnM8FbL
fnKHBsu2rrRH51tJM7VIMkWrGwHv9UAsndoDEtaj0qaF0SaQ8qz+CK55o7HSBSlr1
/C0uQwgH+yOPbaJvKORfXTp7ewlw3xDpYDtGpP744ZI+q4Bzg67c4DixHfMN2PDbLSM1AdrTlaLMVePAHTdpt
Vtf1AWHmtikqLhPLzK3H342kMauXj9ne27wh2IMfXFIWg8vzOo+fmidjSQ9hFvczMeaKikvkpL16BF3CCS8st5Tm
kpyOtRohYvAehY
/dpsMVV9exbpnEt8yU6XVx25qJiuUls0p1iXtJdqESrHgS9VqFGMq9Ew9W21mPT7JX92vXpUZ0T6yvFDfvOOD1
Yy8/23ECzdyqpQyk43LrSpX38ELA3K0+8ZN0mpB+c8mxwTA0I
/dCnCeS6iiCrOhP87CA8Wb5MScS7Q94z+T3jn3wAXbR/uUbTtXJE
/klykknblNfB8xo93cll9Glv9UxRQSMKeBWRZdH9bIAi1xoRhpAgENpAgylKr6DSQ== pflorido@aurora1
```

Add SSH key

# Preparation for the git tutorial

## 7/7

13) You should see something like this below.

### SSH keys

[New SSH key](#)

This is a list of SSH keys associated with your account. Remove any keys that you do not recognize.



SSH

#### Aurora

SHA256:1B48Tf65MTBJwNv3wsk74GGN1qoWYT1ooRp4AVx21hA

Added on Sep 10, 2021

Last used within the last week — Read/write

[Delete](#)

Check out our guide to [generating SSH keys](#) or troubleshoot [common SSH problems](#).

# Validating the github server SSH identity

- As you know from the MNXB01-manual.pdf, there is one missing piece of information:

## **What is the server SSH Key hash?**

- This can be found at the following pages:  
<https://docs.github.com/en/github/authenticating-to-github/keeping-your-account-and-data-secure/githubs-ssh-key-fingerprints>
- Be sure to check that page before connecting!



# Github command line authentication methods

- The official github documentation regarding these methods is here:  
<https://docs.github.com/en/github/authenticating-to-github/keeping-your-account-and-data-secure/about-authentication-to-github#authenticating-with-the-command-line>
- There are three suggested ways:
  - **Token Authentication**, which is the recommended way, same as username and password but you get a special password from the github website for each repository. The token is very hard to remember. It's something like  
`==mjd//#&vkhnwkb237y61398hiw--`  
So the problem here is since no one remembers this, github suggests you to use their own client called `gh` that will securely store your tokens on whatever machine you use it. I personally think it is not appealing for this course. We do not want to teach you how to use one specific vendor-locked tool that only works with github. If you want to use this, read their documentation here:  
<https://docs.github.com/en/get-started/getting-started-with-git/caching-your-github-credentials-in-git>  
most likely, **this tool will disappear with github**. I do not consider it valuable knowledge.
  - **SSH Keys**, which is more or less how we login to Aurora already. I will teach you this because **you can use it in many other ways**, such as for logging in to Aurora, university servers, login to any other possible revision control server out there (gitlab, cvs, svn, mercurial...)
    - **It's a good piece of knowledge that will not disappear in a year or so - it has been on for more than 20!**
  - **Authorizing for SAML single sign-on**: this is only relevant for companies and it involves anyway one of the two solutions above already set up.