# CSE251 Recitation 0

3/4/2019

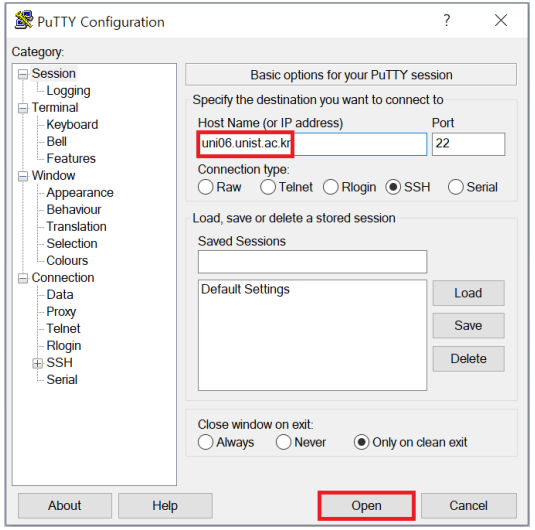
This slide is from CMU, and modified.

# Connecting

## SSH

Windows users: PuTTY

Mac/Linux users: Use ‘ssh’ command at terminal




Make sure you are working on UNIST network(like UNIST_AAA)

uni06.unist.ac.kr - PuTTY

```
login as: cs20xxxxxx
cs20xxxxxx@uni06.unist.ac.kr's password:
```

Type your account

Nothing is seen here even if you are typing a password

# ls <dir> - LiSt

- Lists the files in the present working directory, or, if specified, `dir`.
- `pwd` tells you your Present Working Directory.

```
jbiggs@blueshark ~ $ ls
cover_letter.pdf  factorial.py   Movies     resume.pdf     test.wav
demo.py           foo2.py        Music      school         timer.py
Desktop           foo.txt        Pictures   solutions.py   www
display.py        Fravic.pdf     private    src
Documents         Library        public     Templates
Downloads         Minecraft.jar  Public     test.py
jbiggs@blueshark ~ $ pwd
/afs/andrew.cmu.edu/usr10/jbiggs
jbiggs@blueshark ~ $
```

# mkdir <dirname> - MaKe DIRectory

- Makes a directory `dirname` in your present working directory.
- Directories and folders are the **same thing!**

```
jbiggs@blueshark ~ $ ls
cover_letter.pdf  factorial.py   Movies     resume.pdf     test.wav
demo.py           foo2.py        Music      school         timer.py
Desktop           foo.txt        Pictures   solutions.py   www
display.py        Fravic.pdf     private    src
Documents         Library        public     Templates
Downloads         Minecraft.jar  Public     test.py
jbiggs@blueshark ~ $ cd private/
jbiggs@blueshark ~/private $ mkdir 15-213
jbiggs@blueshark ~/private $ cd 15-213
jbiggs@blueshark ~/private/15-213 $
```

# cd <directory> - Change Directory

- Changes your present working directory to `directory`
- Your main tool for navigating a unix file system

```
jbiggs@blueshark ~ $ ls
cover_letter.pdf  factorial.py   Movies     resume.pdf    test.wav
demo.py           foo2.py        Music      school        timer.py
Desktop           foo.txt        Pictures   solutions.py  www
display.py        Fravic.pdf     private    src
Documents         Library        public     Templates
Downloads         Minecraft.jar  Public     test.py
jbiggs@blueshark ~ $ cd private/
jbiggs@blueshark ~/private $
```

# mv <src> <dest> - MoVe

- cp works in exactly the same way, but copies instead
  - for copying folders, use cp -r
- dest can be into an existing folder (preserves name), or a file/folder of a different name
- Also used to re-name files without moving them
- src can be either a file or a folder

```
jbiggs@blueshark ~ $ cd private/
jbiggs@blueshark ~/private $ mkdir 15-213
jbiggs@blueshark ~/private $ cd 15-213
jbiggs@blueshark ~/private/15-213 $ mv ~/Downloads/datalab-handout.
tar .
```

# `rm <file1> <file2> … <filen>` - ReMove

- Essentially the delete utility
- To remove an (empty) directory, use `rmdir`
  - To remove a folder and its contents, use `rm -rf`
    - **Please be careful, don't delete your project.**
    - **There is no "Trash" here. It's gone.**
    - **If someone asks you to use `rm -rf /` ignore them**

# What's in a file? (using `cat`)

- `cat <file1> <file2> … <filen>` lets you display the contents of a file in the terminal window.
  - Use `cat -n` to add line numbers!
- You can *combine* multiple files into one!
  - `cat <file1> … <filen> > file.txt`
- Good for seeing what's in small files.
- Try `cat -n bits.c`. Too big, right?

# Linux file pathing

- ~ is your **HOME DIRECTORY**
  - This is where you start from after you SSH in
  - On bash, you can also use $HOME
- **.** is an alias for your **PRESENT WORKING DIRECTORY**!
- .. is the file path for the **PARENT DIRECTORY** of your present working directory!
- / is the file path for the **TOP-LEVEL DIRECTORY**
  - You probably won’t use this too much in this class

# Generating a ssh-key

You can find it on our gitlab repo
https://class.unicss.org/cse251-2019-spring/cse251-2019-spring

### Creating and adding your key to Gitlab

Firstly, follow the instructions here to create a ssh key pair on your work machine (e.g. uni server).

- https://docs.gitlab.com/ee/ssh/#generating-a-new-ssh-key-pair

This step creates a pair of files, whose default names are `id_rsa` and `id_rsa.pub`.

While generating the keys, check where the key goes. There are several locations that these would land:

- `/home/<your-user-name>/.ssh/id_rsa.pub`
  - This is the default in most cases.
- `/student_home/<your-user-name>/.ssh/id_rsa.pub`
  - This might be the case on the uni server.

You can add the *public* key to your account following the instruction here.

- http://fab.academany.org/2018/labs/fablabat3flo/students/mat-bgn/wiki/sshkey.html

To obtain your key in plain text, use `cat`.

```
cat <your-pub-key>
```

Then copy-paste the key. Note that in many cases, your machine adds newlines in your key string which you should remove.

Bad example:

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDUOer0D41OwP4QRTj/PKnZnyzWvaAO8DPYEJPJyayqwDuA0Z23JLVhLjvGwGT8
kp5H3Bsl3I84fEgDt+pNSQQwf/gtrZFSffmIqy0BAwPoJSBNPqyb0ZPUEwkn4K9fkwwEsncnzNnEYk4aVR0blkmLfpCN4GJpaug7a2IwaMqA86p2yHsMHg1Jqad
peoF6yxT8/PY7nVl0UUPBOY+WTQX2r+k+U+UZK/TZLe/Vsp hyungon@xxx
```

Good example:

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDUOer0D41OwP4QRTj/PKnZnyzWvaAO8DPYEJPJyayqwDuA0Z23JLVhLjvGwGT8kp5H3Bsl3I84fEgDt+pNSQQ
```

# Generating a ssh-key

## RSA SSH keys

RSA keys are the most common ones and therefore the most compatible with servers that may have an old OpenSSH version. Use them if the GitLab server doesn't work with ED25519 keys.

The minimum key size is 1024 bits, defaulting to 2048. If you wish to generate a stronger RSA key pair, specify the `-b` flag with a higher bit value than the default.

The old, default password encoding for SSH private keys is insecure; it's only a single round of an MD5 hash. Since OpenSSH version 6.5, you should use the `-o` option to `ssh-keygen` to encode your private key in a new, more secure format.

If you already have an RSA SSH key pair to use with GitLab, consider upgrading it to use the more secure password encryption format by using the following command on the private key:

```
ssh-keygen -o -f ~/.ssh/id_rsa
```

## Generating a new SSH key pair

Before creating an SSH key pair, make sure to read about the different types of keys to understand their differences.

To create a new SSH key pair:

1. Open a terminal on Linux or macOS, or Git Bash / WSL on Windows.
2. Generate a new ED25519 SSH key pair:

```
ssh-keygen -t ed25519 -C "email@example.com"
```

Or, if you want to use RSA:

```
ssh-keygen -o -t rsa -b 4096 -C "email@example.com"
```

The `-C` flag adds a comment in the key in case you have multiple of them and want to tell which is which. It is optional.

3. Next, you will be prompted to input a file path to save your SSH key pair to. If you don't already have an SSH key pair and aren't generating a deploy key, use the suggested path by pressing `Enter`. Using the suggested path will normally allow your SSH client to automatically use the SSH key pair with no additional configuration.

   If you already have an SSH key pair with the suggested file path, you will need to input a new file path and declare what host this SSH key pair will be used for in your `~/.ssh/config` file.
4. Once the path is decided, you will be prompted to input a password to secure your new SSH key pair. It's a best practice to use a password, but it's not required and you can skip creating it by pressing `Enter` twice.

   If, in any case, you want to add or change the password of your SSH key pair, you can use the `-p` flag:

```
ssh-keygen -p -o -f <keyname>
```

Now, it's time to add the newly created public key to your GitLab account.

Both ed25519 and rsa are possible.
-C option is optional, not required.

You can set a path and password, or use default setting by pressing Enter key triple times.

# Generating a ssh-key

```
[cs20111100@uni06 ~]$ ssh-keygen -o -t rsa -b 4096
Generating public/private rsa key pair.
Enter file in which to save the key (/students_home/cs20111100/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /students_home/cs20111100/.ssh/id_rsa.
Your public key has been saved in /students_home/cs20111100/.ssh/id_rsa.pub.
The key fingerprint is:
e5:79:6d:5c:3e:3c:7a:eb:78:04:e3:07:ae:48:ac:83 cs20111100@uni06
The key's randomart image is:
+--[ RSA 4096]----+
|                 |
|                 |
|          .     .|
|         o . *.o |
|        S o + B+.|
|         o . +.oo|
|       . o . ..o. |
|      E o . .  o..|
|         .    .oo |
+-----------------+
[cs20111100@uni06 ~]$
```

Then you will be able to see a screen like above.
The red box indicates the location of ssh-key just generated.

# Generating a ssh-key

```
[cs20111100@uni06 ~]$ cat /students_home/cs20111100/.ssh/id_rsa.pub
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQDah7hbf6ThDD3Qm5FVoUPwYuueGxvqlBtHOBummH98n8ULpWB
GCxAVSNu/9ZJDJrjvpTqlR2SP6+/BX4lBtRzCLubP//hMpJ3DObDCegXA595EOOPAfiFLZeryPwOZSNh3tmEksS
FVV8Fk5BYI1mYknpLwMf03DsbK0b6mRvAUA7fM59i0SD7gIlC9x3p/TM+lyZpU4gxANRaSCIa3QxK+kLTB3dlnx
EOset3A2Wnc4iesCgs9rSRkvkSEhv4jPguqhTKTKLU4wzevz5ffbFUxZNtu0JRCvcRfBfwwASMXEOEW+KcmFyTT
N/MT2Myc/Zu/19MgPOvHD2RzwArSQSY+BCAkPTQcXuxWk/FBe4B4IW9nqVD6QYh08zuMKtYLqlKUbXloU0eWxVn
AJOZZ25hSuKbV8wfakB0hpxdn6wst21L866659G6f5S1+b4CYW5buHi86VmAK6sCuus6OUQnBTzSh2S4BkyDBYC
UhdEScDpNsBhYi70W1BcBgLRFPp6bNSBAZRhk5vDiKSTqFHVFINmL8rRFDs7dtalJ4zPBK+h5mmHivZB7Idw4Pt
ed3BzPECRCtM0YHViV3syy2r+FaX/KccW5nqo0HpZO9utqGusfnYay6P1kwXe/PibiltMB1UREpRuiv75yWT8wX
Nrlo6z31ObR/bC18ewkmvpbRhQ== cs20111100@uni06
[cs20111100@uni06 ~]$
```

You can find your ssh-key by cat command like above.
Entire words surrounded by red box is the ssh-key.
In putty, just dragging a text automatically copies the contents(not Ctrl+C).

# Adding your key to gitlab

## Creating and adding your key to Gitlab

Firstly, follow the instructions here to create a ssh key pair on your work machine (e.g. uni server).

- https://docs.gitlab.com/ee/ssh/#generating-a-new-ssh-key-pair

This step creates a pair of files, whose default names are `id_rsa` and `id_rsa.pub`.

While generating the keys, check where the key goes. There are several locations that these would land:

- `/home/<your-user-name>/.ssh/id_rsa.pub`
  - This is the default in most cases.
- `/student_home/<your-user-name>/.ssh/id_rsa.pub`
  - This might be the case on the uni server.

You can add the *public* key to your account following the instruction here.

- http://fab.academany.org/2018/labs/fablabat3flo/students/mat-bqn/wiki/sshkey.html

To obtain your key in plain text, use `cat`.

```
cat <your-pub-key>
```

Then copy-paste the key. Note that in many cases, your machine adds newlines in your key string which you should remove.

Bad example:

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDUOer0D41OwP4QRTj/PKnZnyzWvaAO8DPYEJPJyayqwDuA0Z23JLVhLjvGwGT8
kp5H3Bsl3I84fEgDt+pNSQQwf/gtrZFSffmIqy0BAwPoJSBNPqyb0ZPUEwkn4K9fkwwEsncnzNnEYk4aVR0blkmLfpCN4GJpaug7a2IwaMqA86p2yHsMHg1Jqad
peoF6yxT8/PY7nVl0UUPBOY+WTQX2r+k+U+UZK/TZLe/Vsp hyungon@xxx
```

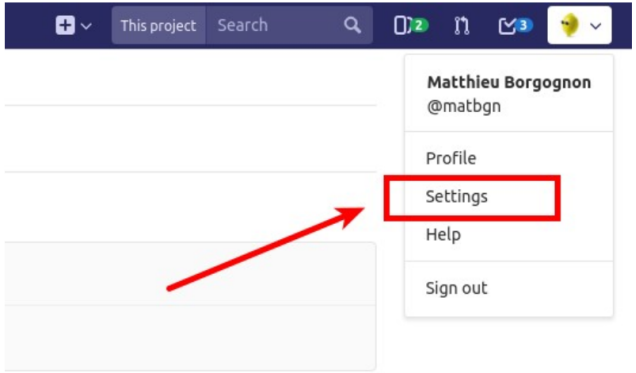
Good example:

```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDUOer0D41OwP4QRTj/PKnZnyzWvaAO8DPYEJPJyayqwDuA0Z23JLVhLjvGwGT8kp5H3Bsl3I84fEgDt+pNSQQ
```

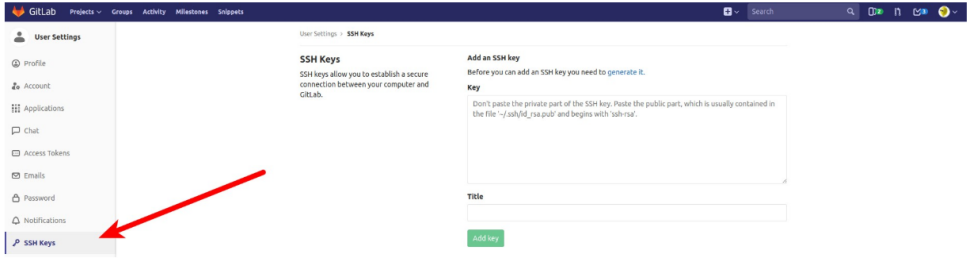
# Adding your key to gitlab

## Profile settings in GitLab

First of all go to your peronal settings in the upper right corner:




Then select SSH Key subcategory:



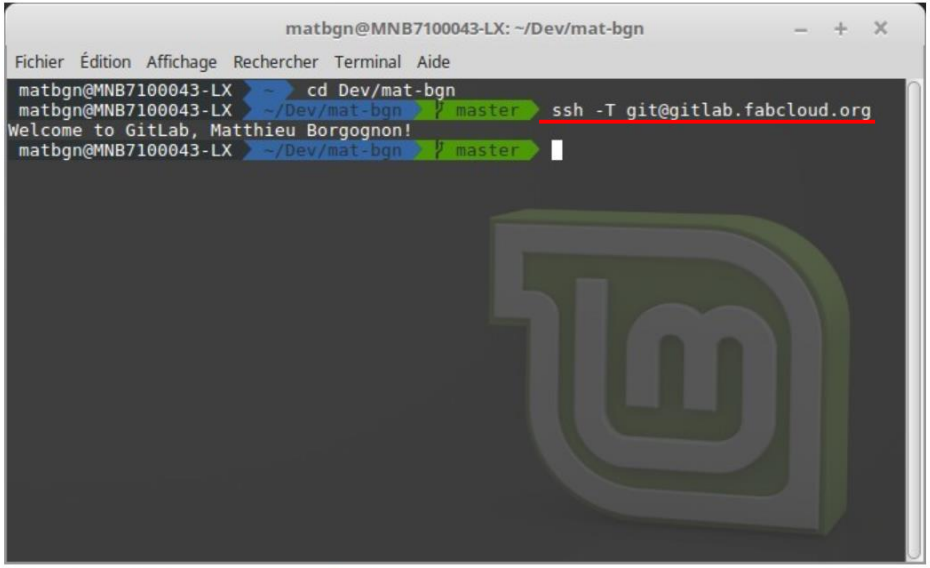

You are now readdy to generate your keys.

# Test if your ssh-key is enrolled properly

## Test your SSH connection

Before doing anything else try your connection to Gitlab with this command:

```
ssh -T git@gitlab.fabcloud.org
```




You should recieve a Welcome message like that.

In our case, try
ssh –T git@class.unicss.org –p 4002
-p option is needed because
our ssh port number is 4002.

If you see “Welcome ~”, the key is enrolled properly.

Now, you can follow the direction of Lab0.

# Recommendation

- Completes whole codes on my computer and then tests on uni06 server?
  - No! Your codes may not work on uni06 server.
  - If you are not working on uni06 server, please test your code on uni06 server periodically.
- Ask Google and TAs.
  - Make an issue on gitlab if your question is somewhat general(so that it can be shared).
  - TA office hour will be announced within this week, or you can make an appointment by e-mail.