



LEIDEN UNIVERSITY MEDICAL CENTER

# Connecting to other machines

**Jeroen F. J. Laros**

**Leiden Genome Technology Center**

**Department of Human Genetics**

**Center for Human and Clinical Genetics**



*Servers*

Remote machines can be very convenient:

- One central machine for calculation.
- Cuts expenses.
- No one wants a cluster in their office.
- Specialised software only in one place.
- Multiple users can use it at the same time.
- ...

## *Logging in*

There are lots of ways to connect to a server.

- HTTP – When visiting websites.
- IMAP – When fetching mail.
- ...

## *Logging in*

There are lots of ways to connect to a server.

- HTTP – When visiting websites.
- IMAP – When fetching mail.
- ...

In order to execute commands, we need to *log in*.

We use a *secure* protocol to log in.

- Most plain text protocols are blocked by firewalls.
- When working with patient data, we don't want eavesdropping.
- The connection from your machine to the server is *encrypted*.

## *Secure Shell*

```
1 $ ssh user@host
```

Listing 1 : Using Secure Shell (ssh)..

Keyword	Description
<b>user</b>	Your <i>username</i> on the <i>server</i> .
<b>host</b>	Name of the <i>server</i> .

Table 1 : Parameters of ssh.

## *Secure Shell*

```
1 $ ssh user@host
```

Listing 1 : Using Secure Shell (ssh)..

Keyword	Description
<b>user</b>	Your <i>username</i> on the <i>server</i> .
<b>host</b>	Name of the <i>server</i> .

Table 1 : Parameters of ssh.

```
1 $ ssh course@shark
```

Listing 2 : Example.

## *Copying data*

We frequently need to transfer data before and after we do an analysis.

- The input needs to be on the server.
- The output needs to be copied back.

## *Copying data*

We frequently need to transfer data before and after we do an analysis.

- The input needs to be on the server.
- The output needs to be copied back.

We also use a secure protocol to copy.

- If Secure Shell works, then this will work too (same protocol).
- Two way traffic.
  - Copy data from your machine to the server (uploading).
  - Copy data from the server to your machine (downloading).



## Secure Copy

```
1 $ scp localfile user@host:/path/to/remotefile
```

Listing 3 : Copying something to the server.

Keyword	Description
<b>localfile</b>	Name of the file on <i>your</i> computer.
<b>user</b>	Your <i>username</i> on the <i>server</i> .
<b>host</b>	Name of the <i>server</i> .
<b>/path/to/</b>	Directory on the <i>server</i> to store the file.
<b>remotefile</b>	Name of the file on the <i>server</i> .

Table 2 : Description of the parameters.

## Secure Copy

```
1 $ scp localfile host:
2 $ scp host:remotefile .
```

Listing 4 : Example.

Keyword	Description
<b>user</b>	The username that you use on your <i>local</i> machine.
<b>/path/to/</b>	The home directory of the user on the server.
<b>remotefile</b>	The same as the name of the local file.
<b>localfile</b>	May be replaced by a “.” when copying something from the server.

Table 3 : Some defaults (when left empty).

### *Windows*

Windows does not have the `ssh` command, but there are programs that give the same functionality.

PuTTY – A Free Telnet/SSH Client.

A software package containing (amongst others):

- PuTTY: Secure Shell client.
- PSCP: Secure Copy client.
- More related tools available on the website.

<http://www.chiark.greenend.org.uk/~sgtatham/putty>

# Connecting from an other OS

## Windows

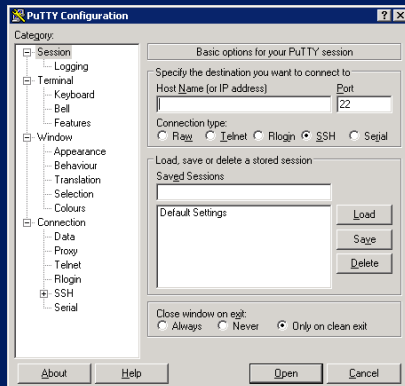
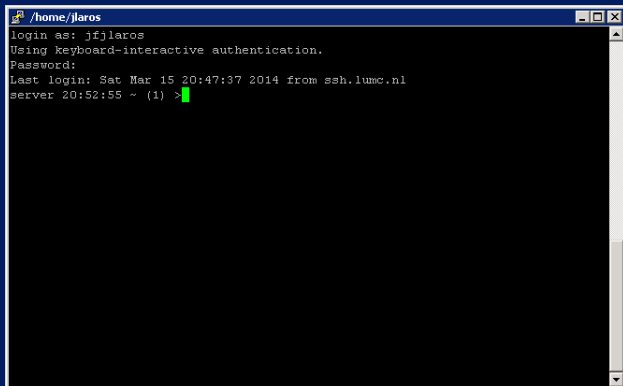


Figure 1 : Connecting to a server using PuTTY.

# Connecting from an other OS

## *Windows*



```

/home/jlaros
login as: jfjlaros
Using keyboard-interactive authentication.
Password:
Last login: Sat Mar 15 20:47:37 2014 from ssh.lumc.nl
server 20:52:55 ~ (1) >
  
```

Figure 2 : A terminal when connected to a server.

### *Typical workflow*

When doing an analysis, the general workflow looks like this:

- First copy the input data to the server.
- Log on to the server.
- Run the analysis remotely.
- Copy the results from the server.
- Clean up the input data and the results on the server.
- Log out.

## *Typical workflow: an example*

Step one: preparing the input.

On your machine, copy the raw data to the server, then log in on the server.

```
1  $ scp reads.fq course@shark:
2  $ ssh course@shark
```

Listing 5 : Copy data to the server and log in.

Now the file **reads.fq** is available on the server.

*Typical workflow: an example*

Step two: The analysis.

On the server, you can do an analysis.

```
1 $ bwa aln ./indexes/chr17.fa reads.fq > reads.sai
2 $ bwa samse ./indexes/chr17.fa reads.sai \
3   reads.fq > reads.sam
4 $ samtools view -bt ./indexes/chr17.fa \
5   -o reads.bam reads.sam
6 $ samtools sort reads.bam reads.bam.sorted
7 $ samtools pileup -vcf ./indexes/chr17.fa \
8   reads.bam.sorted.bam > reads.pileup
```

Listing 6 : Example pipeline.



*Typical workflow: an example*

Step three: Retrieving the output.

Copy the output from the server back to your own machine.

```
1 $ scp course@shark:reads.pileup .
```

Listing 7 : Copy data from the server.

*Typical workflow: an example*

Step three: Retrieving the output.

Copy the output from the server back to your own machine.

```
1 $ scp course@shark:reads.pileup .
```

Listing 7 : Copy data from the server.

Step four: Cleaning up.

Clean up on the server and leave.

```
1 $ rm reads.*  
2 $ logout
```

Listing 8 : Delete temporary files and log out.

# Why remote servers?

## *Clusters*



Figure 3 : Roadrunner supercomputer.

## *Clusters*

Massive parallel computing.

- A large number of computers working together.
- Analyse lots of samples at the same time.
- Sometimes a way to reduce memory requirements (if the problem permits it).
- Very suitable for NGS, especially alignment.

## *Clusters*

Massive parallel computing.

- A large number of computers working together.
- Analyse lots of samples at the same time.
- Sometimes a way to reduce memory requirements (if the problem permits it).
- Very suitable for NGS, especially alignment.

Cons:

- Not all problems are suitable for parallel computation.
- Programs must be adjusted to make use of a cluster.
  - Chop the problem op in parts / combine the results.

# Why remote servers?

## *Clusters*

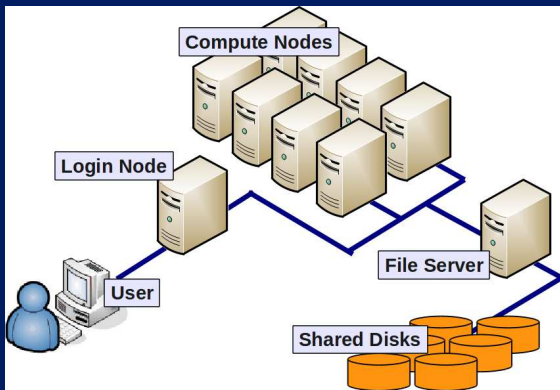


Figure 4 : Schematic overview of a cluster.

# Why remote servers?

## *Clusters*

General characteristics of a cluster.

- Jobs are submitted to a *control node*.
- The control node dispatches a job to a free *worker node*.
- Jobs are monitored.
  - If a worker node doesn't finish for some reason, the job gets dispatched to an other worker node.
  - If all worker nodes are finished, the control node can alert the user that his jobs are finished.
- Jobs can be prioritised.
- ...



Michiel van Galen  
Jeroen Laros

<https://humgenprojects.lumc.nl/trac/humgenprojects/wiki/NGS-intro>