

Basic tools : Last tips

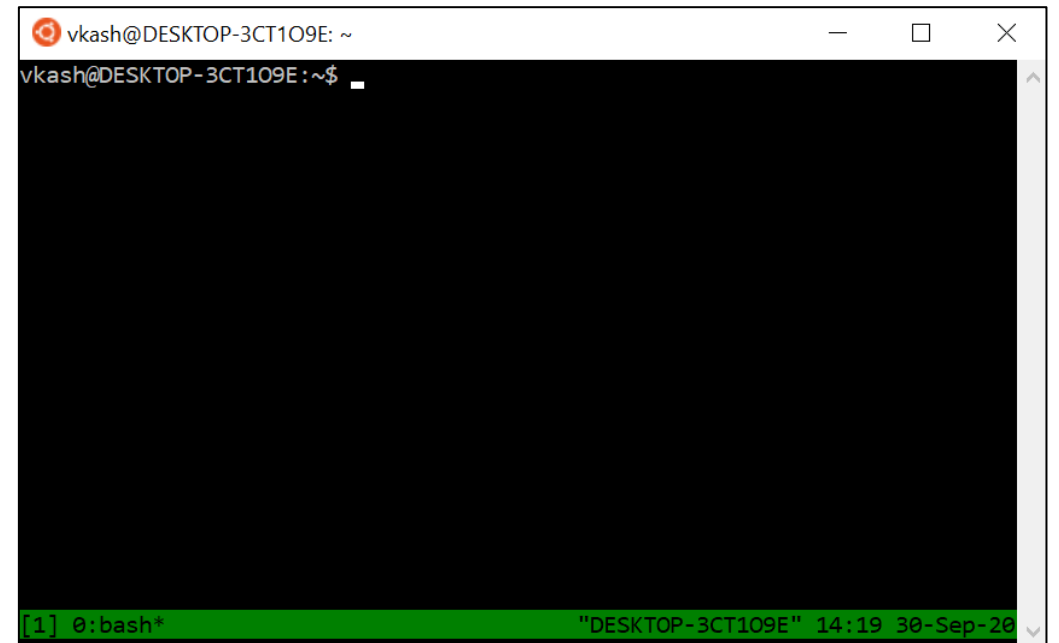
KASHTANOVA Victoriya

Inria, Epione

Tmux

tmux is a terminal multiplexer for Unix-like operating systems.

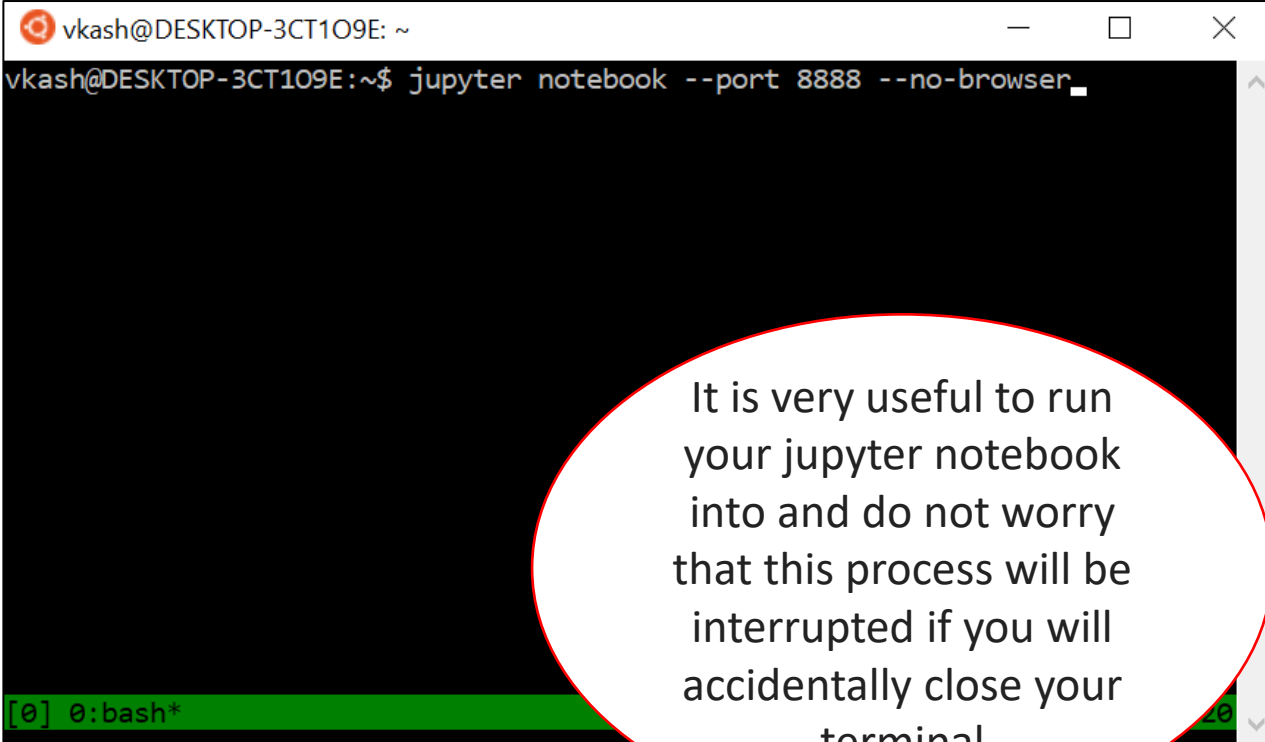
- It allows **multiple terminal sessions** to be accessed simultaneously in a single window.
- It is useful **for running more than one command-line program at the same time**.
- It can also be used **to detach processes from their controlling terminals, allowing remote sessions to remain active** without being visible



Tmux shortcuts

- start new session :
tmux
- start new with session name:
tmux new -s myname
- list sessions:
tmux ls
- attach session by name or number:
tmux attach -t myname (number)
- kill session by name or number :
tmux kill-session -t myname (number)

Go back to the terminal : **Ctrl+B**, then **D**

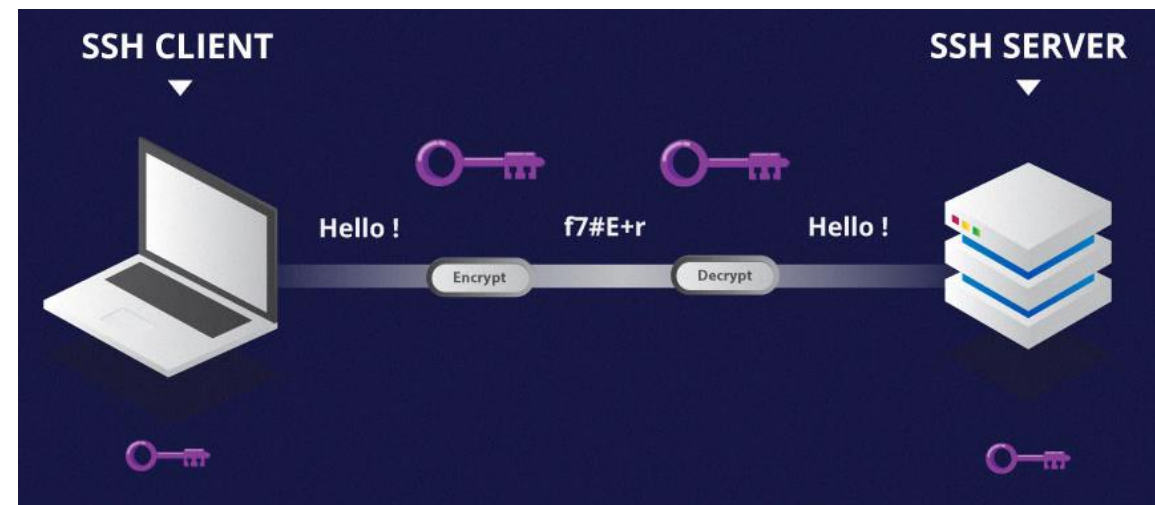
A terminal window titled 'vkash@DESKTOP-3CT109E: ~' with standard window controls. The prompt is 'vkash@DESKTOP-3CT109E:~\$' and the command 'jupyter notebook --port 8888 --no-browser' has been entered. The terminal background is black with white text. At the bottom, a green status bar shows '[0] 0:bash*'.

```
vkash@DESKTOP-3CT109E: ~  
vkash@DESKTOP-3CT109E:~$ jupyter notebook --port 8888 --no-browser  
[0] 0:bash*
```

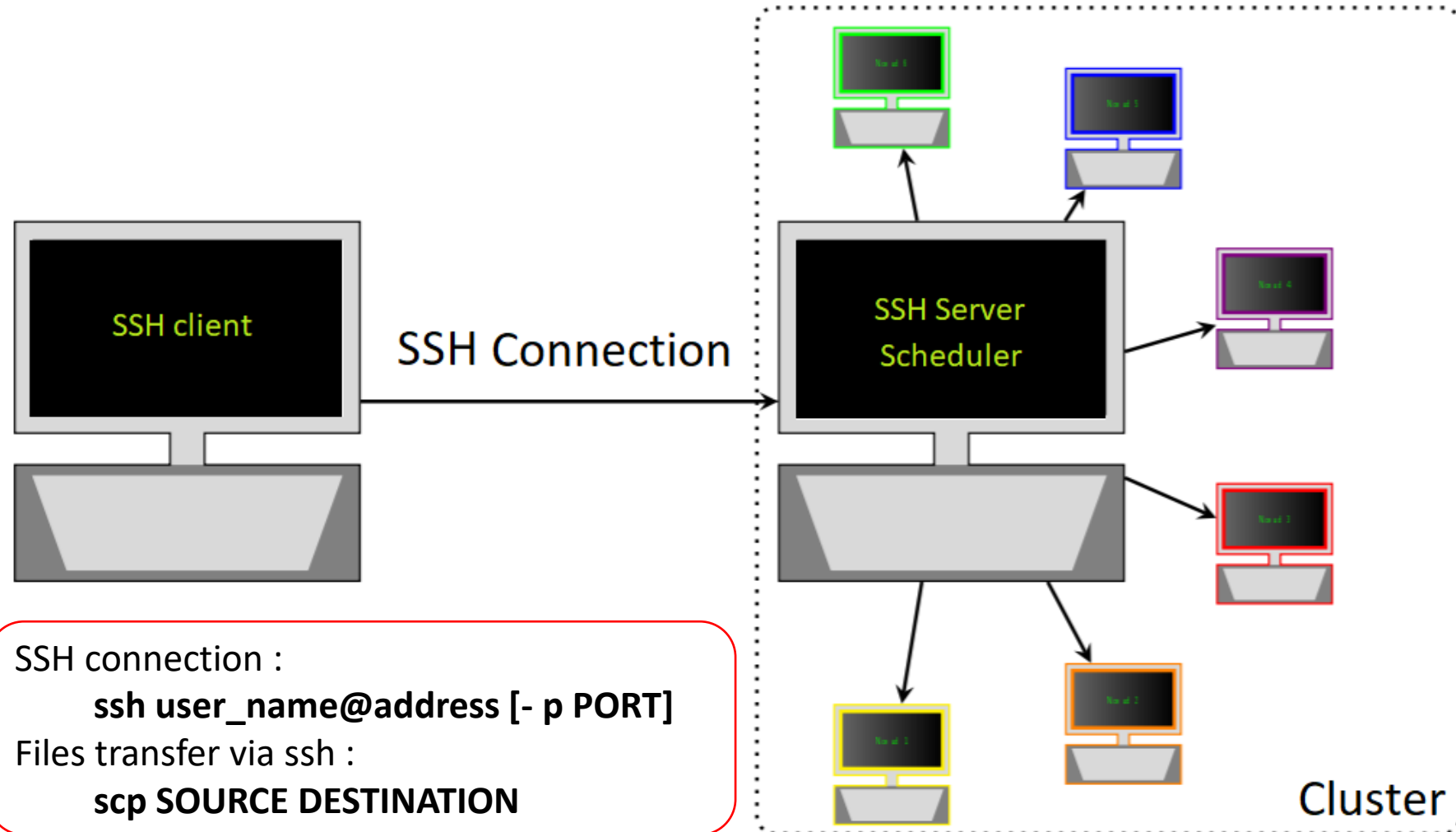
It is very useful to run your jupyter notebook into and do not worry that this process will be interrupted if you will accidentally close your terminal

SSH

- **Secure Shell (SSH)** is a cryptographic network protocol for operating network services securely over an unsecured network. Typical applications include remote command-line, login, and remote command execution.
- **SSH provides a secure channel** over an unsecured network by using a client–server architecture, connecting an SSH client application with an SSH server **via a secure key**.

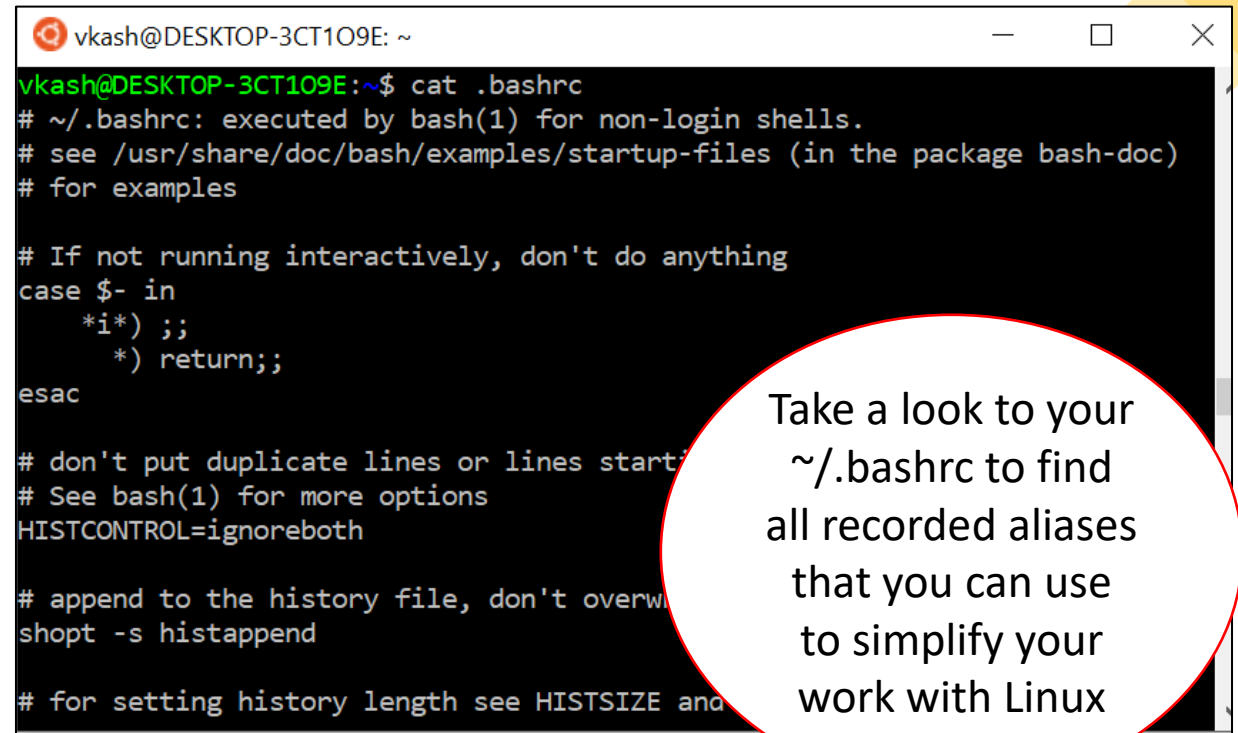


Exemple : SSH connection to server



.bashrc

- The `~/.bashrc` file determines the behavior of interactive shells.
- A good look at this file can lead to a better understanding of your **Bash**.
- You can find there useful aliases and add yours.
- All information (variables, aliases, function etc) recorded in the `~/.bashrc` file will be kept and can be used by all running terminals.

A terminal window titled 'vkash@DESKTOP-3CT1O9E: ~' showing the output of the command 'cat ~/.bashrc'. The output displays the standard bashrc configuration, including comments about non-login shells, interactive shell handling, and history settings. A red circle highlights a portion of the text on the right side of the terminal window.

```
vkash@DESKTOP-3CT1O9E: ~$ cat ~/.bashrc
# ~/.bashrc: executed by bash(1) for non-login shells.
# see /usr/share/doc/bash/examples/startup-files (in the package bash-doc)
# for examples

# If not running interactively, don't do anything
case $- in
    *i*) ;;
    *) return;;
esac

# don't put duplicate lines or lines starting with space
# See bash(1) for more options
HISTCONTROL=ignoreboth

# append to the history file, don't overwrite
shopt -s histappend

# for setting history length see HISTSIZE and
```

Take a look to your
`~/.bashrc` to find
all recorded aliases
that you can use
to simplify your
work with Linux