High Performance
Computing &
Big Data Services

hpc.uni.lu

hpc@uni.lu

@ULHPC

LUXEMBOURG
LET'S MAKE IT HAPPEN

Uni.lu HPC School 2020

PS1: Preliminaries (SSH - OpenOnDemand)

Uni.lu High Performance Computing (HPC) Team

T. Valette and A. Olloh

University of Luxembourg (UL), Luxembourg

http://hpc.uni.lu

Latest versions available on Github:

UL HPC tutorials: https://github.com/ULHPC/tutorials

UL HPC School: http://hpc.uni.lu/hpc-school/

PS1 tutorial sources: ulhpc-tutorials.rtfd.io/en/latest/preliminaries

Summary

Main Objectives of this Session

- Review **recommended setup for your personnal Laptop**
  - ↪ Windows, Mac OS or Linux
- Understand **SSH** (Secure SHell)
- **Review SSH configuration** allowing to connect to the UL HPC Platform
  - ↪ SSH configuration
  - ↪ Generate your SSH key pair
  - ↪ Upload your SSH key on ULHPC Identity Management Portal
- **Discover ULHPC Open OnDemand**
  - ↪ HPC Web portal
  - ↪ Job composer

OPEN OnDemand

Vocabulary related to HPC (1/2)

Compute node physical server on which we run the computation (your code)

Cluster group of compute nodes interconnected to each others

Processor/CPU Central Processing Unit usually refers to a processor, chip of the server that process the instructions of the program

Core 1 processor chip usually contains several CPUs named cores

GPU Graphics Processing Unit, chip designed for image processing and computer graphics

Vocabulary related to HPC (2/2)

Resources	Every component of the cluster that you have access. Can refer to CPU, core, memory, network switch...
Job	Allocation resources for a specific user and a specific amount of time
Reservation	Allocate a job in the future, in advance in respect with rules (priority, job type...)
Walltime	Maximum time allocated for a specific job
Job Scheduler	Software that schedule all the jobs according to their priority.
Job queue	Before being scheduled, jobs are waiting in a queue for being processed by the scheduler
Partition	Set of resources (nodes) with the same policies applied to it

Summary

Preliminaries

ULHPC Tutorials: Setup Preliminaries

ulhpc-tutorials.rtfd.io/en/latest/setup/preliminaries

- **Pre-configuration expected on your laptop**
- **Online Accounts**, at least:
  - ↪ Uni.lu HPC - read and accept the Acceptable Use Policy (AUP) 2.0
  - ↪ Github - recommended professional login: `<letter><name>` (Ex: `svarrette`)
  - ↪ Vagrant Cloud
  - ↪ Docker Hub

- All instructions/practical sessions guided against a Unix/Linux system
  - ↪ Native OS within **all** HPC facilities
  - ↪ interaction from any OS (Windows, Mac OS or Linux) is possible for **all** operations
    - ✓ ... unlike WebEx Training/Event

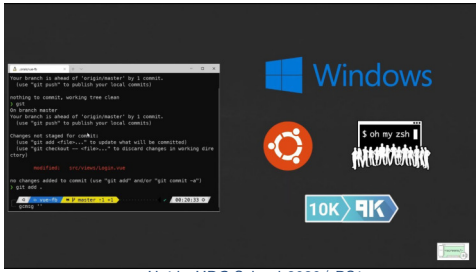
Recommended Software List

Platform	Software	Description	Usage
Mac OS	Homebrew	The missing package manager for macOS	`brew install ...`
Mac OS	iTerm2	enhanced Terminal	
Windows	Chocolatey	Package Manager for Windows	`choco install ...`
Windows	Windows Subsystem for Linux (WSL)	Emulation-like translation of Linux kernel system calls	
Windows	Ubuntu over WSL	Linux Ubuntu on Windows	
Windows	Windows Terminal		
Windows	MobaXTERM	Terminal with tabbed SSH client	
Windows	SourceTree	enhanced git GUI	
Windows/Linux	Virtual Box	Free hypervisor provider for Vagrant	
Windows/Linux	Vagrant	Reproducible environments made easy.	
Windows	Docker for Windows	Lightweight Reproducible Containers	
Linux	Docker for Ubuntu	Lightweight Reproducible Containers	

Preliminaries Setup for Windows

≥ Windows 10 release 1607

- New "**Windows Subsystem for Linux**" Ubuntu Linux binaries
  - ↪ can run directly on Windows without the need for the Linux Kernel running in a VM.
    - ✓ provides emulation-like translation of Linux kernel system calls to the Windows kernel
  - ↪ Windows *feature*, **requires to turn on Developer mode**
- Install Ubuntu within the Microsoft Store
  - ↪ You now have an Ubuntu terminal by running `bash` command
    - ✓ **(optional)** consider installing of-my-zsh and the Powerlevel10k prompt - guide
- Install Windows Terminal
  - ↪ Setup prompt in Windows Terminal & PowerShell
  - ↪ **Now open Ubuntu profile**
    - ✓ enjoy an *fully-featured Linux terminal*
      ... incl. native SSH
- *(also for OLD systems)* install MobaXterm



Preliminaries Setup for Linux *(incl. WSL Ubuntu)* & Mac OS

- Native SSH client available, accessible with your terminal
  - ↪ Make sure you can open and interact with a terminal on your computer
  - ↪ There exists **better** terminal applications than the default ones proposed on your system
    - ✓ Mac OS: iTerm2
    - ✓ Windows Terminal
    - ✓ Linux: Guake Terminal, Terminator
- If you already have an SSH Client, you should have this following output:

```
$(laptop)> ssh -V
OpenSSH_8.4p1, OpenSSL 1.1.1h  22 Sep 2020
```

- **(optional)** If you want a fancy et nice terminal layout, consider using `zsh` and `oh-my-zsh`
  - ↪ `guide` for Linux
  - ↪ `guide` for Mac OS

Git in a Nutshell

git-scm.com

- **Distributed Version Control System**
  - ↪ Manage changes in documents and Keep the history
- **Synchronization with a server** (optional)
  - ↪ Collaborative work
  - ↪ Backup of your documents/code
- Well suited for **text** files (source code, LaTeX article etc.)
  - ↪ can also be used for binary files (images, HDF5, Office docs. . .)
    - ✓ if the size is reasonable
  - ↪ typically avoid to version temporary (scratch) files
  - ↪ extension exists to support large files (Git-LFS)

Installation Notes

git-scm.com

```
### Mac OS X, using Homebrew - https://brew.sh
brew install git git-gui git-flow gitk tig kdiff3
### Ubuntu / WSL / RedHat
{ sudo apt-get | yum } install git-core git-flow tig gitk kdiff3
### Windows, using Chocolatey - https://chocolatey.org/
$> choco.exe install git gitflow-avh
```

- **Windows**: Git for Windows: incl. Git Bash/GUI & Shell Integration
  - ↪ (**better**) Windows Subsystem for Linux (WSL)
    - ✓ Ubuntu for Windows within the Microsoft Store

Installation Notes

git-scm.com

```
### Mac OS X, using Homebrew - https://brew.sh
brew install git git-gui git-flow gitk tig kdiff3
### Ubuntu / WSL / RedHat
{ sudo apt-get | yum } install git-core git-flow tig gitk kdiff3
### Windows, using Chocolatey - https://chocolatey.org/
$> choco.exe install git gitflow-avh
```

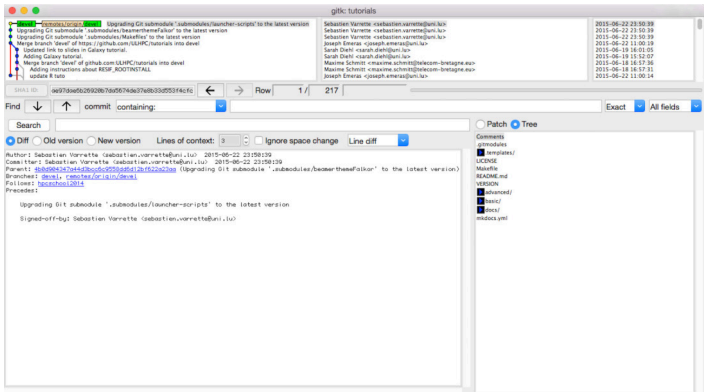
- **Windows**: Git for Windows: incl. Git Bash/GUI & Shell Integration
  - ↪ (**better**) Windows Subsystem for Linux (WSL)
    - ✓ Ubuntu for Windows within the Microsoft Store

Your Turn!

- Ensure you have git installed on your system `git --version`

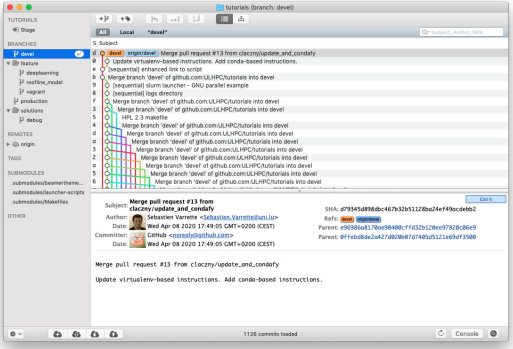
Other related tools

- Git GUI (default): **GitK**



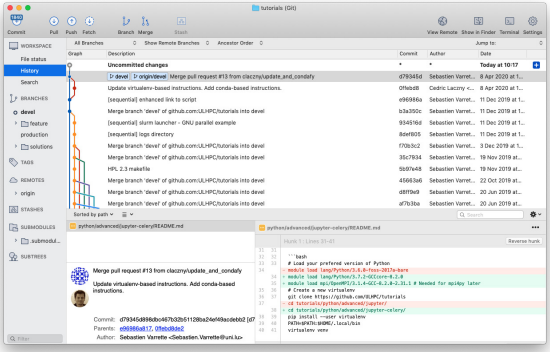
Other related tools

- Git GUI (Mac OS): **GitX-dev** http://rowanj.github.io/gitx/



Other related tools

- Git GUI (Windows/Mac OS): **SourceTree**
  - ↪ Alternatives: `https://git-scm.com/downloads/guis/`



Preliminary Configurations

- You **SHOULD at least** configure your name and email to commit
  ↪ **(optional but useful)** inform Git of your GPG signing key

```
### Basic Git defaults - open a Terminal
git config --global user.name  "Firstname LastName"
git config --global user.email "<email>@<domain>"
git config --global user.signingkey <gpg-keyID>
git config --global color.ui true
# Check that the changes are effective (see also ~/.gitconfig or ~/.config/git/config)
git config -l | grep user
```

Preliminary Configurations

- You **SHOULD at least** configure your name and email to commit
  ↪ **(optional but useful)** inform Git of your GPG signing key

```
### Basic Git defaults - open a Terminal
git config --global user.name  "Firstname LastName"
git config --global user.email "<email>@<domain>"
git config --global user.signingkey <gpg-keyID>
git config --global color.ui true
# Check that the changes are effective (see also ~/.gitconfig or ~/.config/git/config)
git config -l | grep user
```

Hands-on Git Config

▶ url ◀ | github | src

- Git Installation and useful derived tools — `git-flow`, `tig`, `kdiff3`, `meld`...
- Git Initial Setup — `git config --global [...]`

Shell Integration: Bash prompt

For experienced users only!

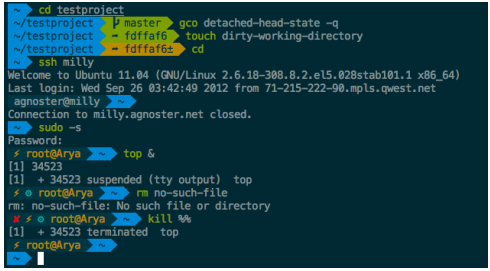
- Bash Prompt integration: `__git_ps1()` in your `PS1` variable
  ↪ See `contrib/completion/git-prompt.sh` (Git Sources)

Shell Integration: Bash prompt

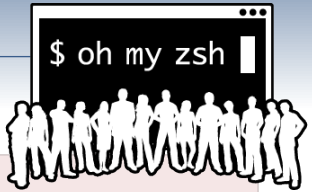
For experienced users only!

- Bash Prompt integration: `__git_ps1()` in your `PS1` variable
  - ↪ See `contrib/completion/git-prompt.sh` (Git Sources)

- **OR Agnoster Theme for Bash**
  - ↪ requires a Powerline-patched font

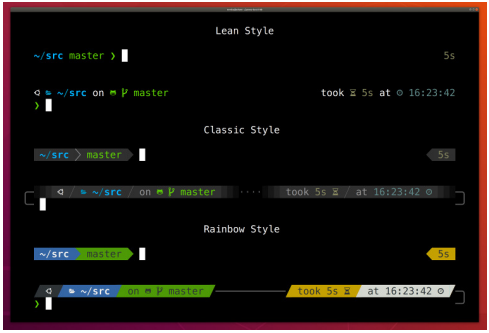


Shell Integration: ZSH

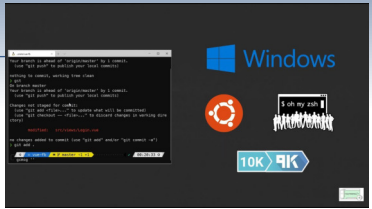


For experienced users only!

- Oh my ZSH!:
  - ↪ community-driven framework
  - ↪ manages your Zsh configuration
  - ↪ multiple plugins, (super nice) themes
    - ✓ Ex: agnoster theme / powerline
  - ↪ Install: `https://ohmyz.sh/#install`
- Consider Powerlevel10k theme for ZSH

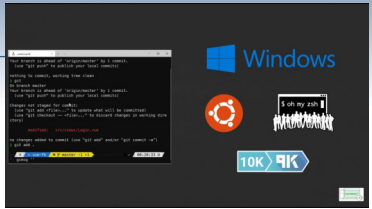


Shell Integration: Powershell



- You probably want to install Windows Terminal
  - ↪ Install the Windows Terminal from the Microsoft Store.
  - ↪ Install the Cascadia Code PL font

Shell Integration: Powershell



- You probably want to install Windows Terminal
  - ↪ Install the Windows Terminal from the Microsoft Store.
  - ↪ Install the Cascadia Code PL font

For experienced users only!

- Install posh-git and oh-my-posh:

```
# Install posh-git and oh-my-posh in (administrator) powershell
# If Error: "Running scripts is disabled on this system"
Get-ExecutionPolicy -List
Set-ExecutionPolicy RemoteSigned -Scope CurrentUser
# Install
Install-Module posh-git -Scope CurrentUser
Install-Module oh-my-posh -Scope CurrentUser
Set-Theme Paradox
```

Summary

SSH: Secure Shell

- Ensure **secure** connection to remote (UL) server
  - ↪ establish **encrypted** tunnel using **asymmetric keys**
    - ✓ **Public** `id_rsa.pub` **vs. Private** `id_rsa` (**without** `.pub`)
    - ✓ typically on a non-standard port (**Ex**: 8022) *limits kiddie script*
    - ✓ Basic rule: 1 machine = 1 key pair
  - ↪ the private key is **SECRET**: **never** send it to anybody
    - ✓ Can be protected with a passphrase

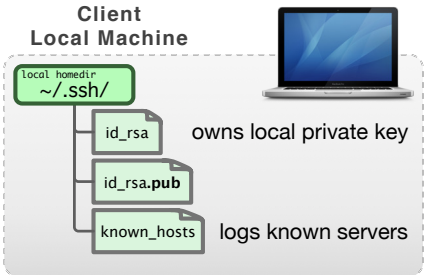
SSH: Secure Shell

- Ensure **secure** connection to remote (UL) server
  - ↪ establish **encrypted** tunnel using **asymmetric keys**
    - ✓ **Public** `id_rsa.pub` **vs. Private** `id_rsa` (**without** `.pub`)
    - ✓ typically on a non-standard port (**Ex**: 8022) *limits kiddie script*
    - ✓ Basic rule: 1 machine = 1 key pair
  - ↪ the private key is **SECRET**: **never** send it to anybody
    - ✓ Can be protected with a passphrase
- SSH is used as a secure backbone channel for **many** tools
  - ↪ Remote shell **i.e** remote command line
  - ↪ File transfer: `rsync`, `scp`, `sftp`
  - ↪ versionning synchronization (`svn`, `git`), github, gitlab etc.

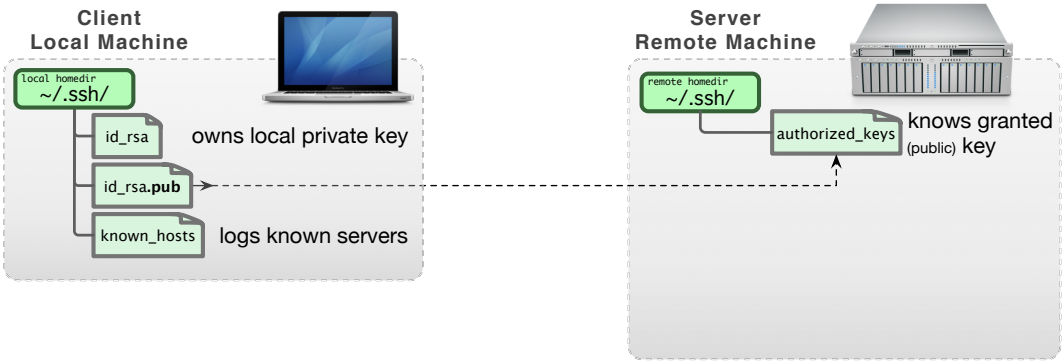
SSH: Secure Shell

- Ensure **secure** connection to remote (UL) server
  - ↪ establish **encrypted** tunnel using **asymmetric keys**
    - ✓ **Public** `id_rsa.pub` **vs. Private** `id_rsa` (**without** `.pub`)
    - ✓ typically on a non-standard port (**Ex**: 8022) *limits kiddie script*
    - ✓ Basic rule: 1 machine = 1 key pair
  - ↪ the private key is **SECRET**: **never** send it to anybody
    - ✓ Can be protected with a passphrase
- SSH is used as a secure backbone channel for **many** tools
  - ↪ Remote shell **i.e** remote command line
  - ↪ File transfer: `rsync`, `scp`, `sftp`
  - ↪ versionning synchronization (`svn`, `git`), github, gitlab etc.
- Authentication:
  - ↪ ~~password~~ (disable if possible)
  - ↪ (*better*) **public key authentication**

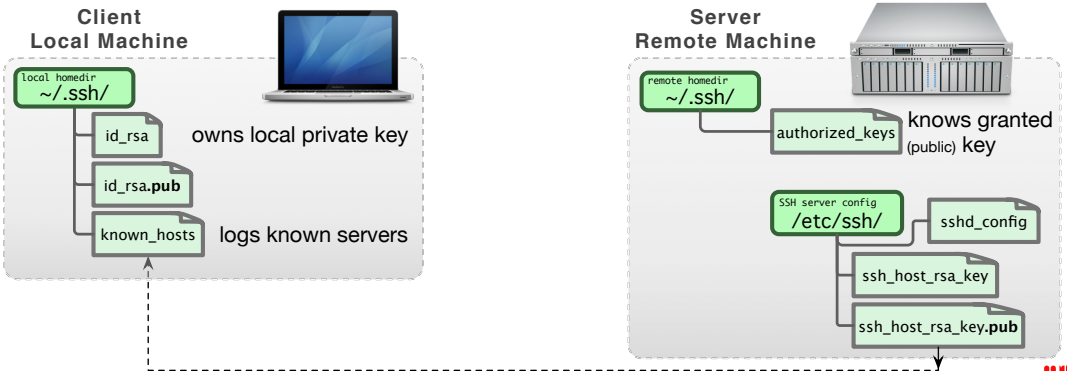
SSH: Public Key Authentication



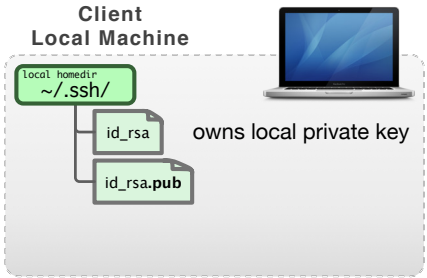
SSH: Public Key Authentication

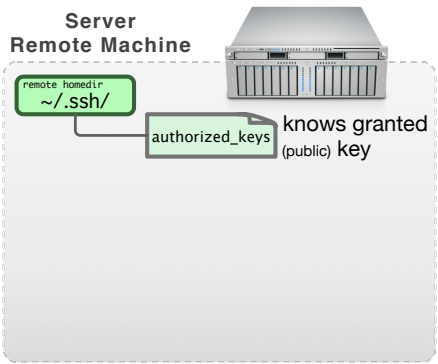


SSH: Public Key Authentication

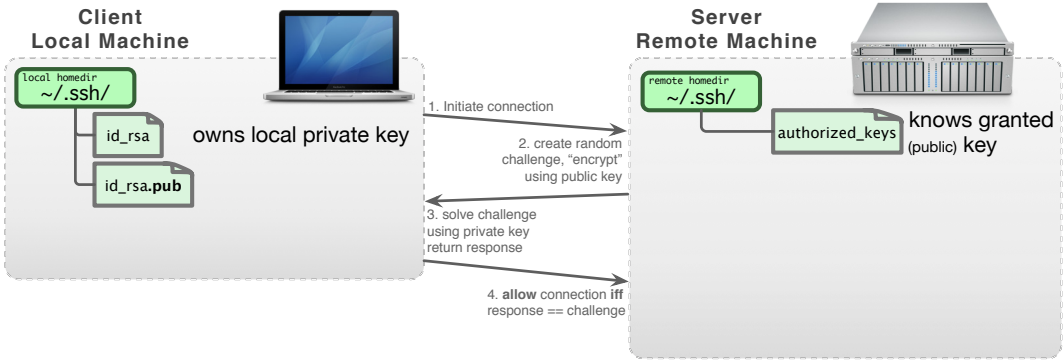


SSH: Public Key Authentication





SSH: Public Key Authentication



SSH: Public Key Authentication

- **Restrict** to public key authentication: `/etc/ssh/sshd_config`:

```
PermitRootLogin no
# Disable Passwords
PasswordAuthentication no
ChallengeResponseAuthentication no
```

```
# Enable Public key auth.
RSAAuthentication yes
PubkeyAuthentication yes
```

SSH Setup on Linux / Mac OS

- OpenSSH natively supported; configuration directory : `~/.ssh/`
  - ↪ package `openssh-client` (Debian-like) or `ssh` (Redhat-like)
- SSH Key Pairs (public vs private) generation: `ssh-keygen`
  - ↪ specify a **strong** passphrase
    - ✓ protect your **private** key from being stolen **i.e.** impersonation
    - ✓ **drawback:** passphrase must be typed to use your key

SSH Setup on Linux / Mac OS

- OpenSSH natively supported; configuration directory : `~/.ssh/`
  - ↪ package `openssh-client` (Debian-like) or `ssh` (Redhat-like)
- SSH Key Pairs (public vs private) generation: `ssh-keygen`
  - ↪ specify a **strong** passphrase
    - ✓ protect your **private** key from being stolen **i.e.** impersonation
    - ✓ ~~**drawback:** passphrase must be typed to use your key~~ `ssh-agent`

SSH Setup on Linux / Mac OS

- OpenSSH natively supported; configuration directory : `~/.ssh/`
  - ↪ package `openssh-client` (Debian-like) or `ssh` (Redhat-like)
- SSH Key Pairs (public vs private) generation: `ssh-keygen`
  - ↪ specify a **strong** passphrase
    - ✓ protect your **private** key from being stolen **i.e.** impersonation
    - ✓ ~~**drawback:** passphrase must be typed to use your key~~ `ssh-agent`

DSA Keys + RSA 1024/2048 bits Dare deprecated now!

SSH Setup on Linux / Mac OS

- OpenSSH natively supported; configuration directory : `~/.ssh/`
  - ↪ package `openssh-client` (Debian-like) or `ssh` (Redhat-like)
- SSH Key Pairs (public vs private) generation: `ssh-keygen`
  - ↪ specify a **strong** passphrase
    - ✓ protect your **private** key from being stolen **i.e.** impersonation
    - ✓ ~~**drawback:** passphrase must be typed to use your key~~ `ssh-agent`

DSA Keys + RSA 1024/2048 bits Dare deprecated now!

```
$> ssh-keygen -t rsa -b 4096 -o -a 100                  # 4096 bits RSA
(better) $> ssh-keygen -t ed25519 -o -a 100             # new sexy Ed25519
```

Private (identity) key

`~/.ssh/id_{rsa,ed25519}`

Public Key

`~/.ssh/id_{rsa,ed25519}.pub`

SSH Setup on Windows: the OLD way

- Putty Suite, includes: http://www.chiark.greenend.org.uk/~sgtatham/putty/
  - ↪ PuTTY, the free SSH client
  - ↪ Pageant, an SSH authentication agent for PuTTY tools
  - ↪ PLink, th PuTTy CLI and PuTTYgen, an RSA and DSA key generation utility

SSH Setup on Windows: the OLD way

- Putty Suite, includes: http://www.chiark.greenend.org.uk/~sgtatham/putty/
  - ↪ PuTTY, the free SSH client
  - ↪ Pageant, an SSH authentication agent for PuTTY tools
  - ↪ PLink, th PuTTy CLI and PuTTYgen, an RSA and DSA key generation utility

PuTTY $\neq$ OpenSSH

SSH Setup on Windows: the OLD way

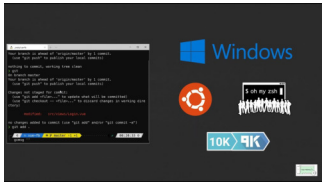
- Putty Suite, includes: http://www.chiark.greenend.org.uk/~sgtatham/putty/
  - ↪ PuTTY, the free SSH client
  - ↪ Pageant, an SSH authentication agent for PuTTY tools
  - ↪ PLink, th PuTTy CLI and PuTTYgen, an RSA and DSA key generation utility

PuTTY ≠ OpenSSH

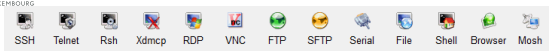
- Putty keys are **NOT** supported by OpenSSH (yet can be exported)
- Binding Pageant with OpenSSH agent is **NOT** natively supported
  - ↪ Third-party tools like ssh-pageant are made for that
  - ↪ Combine nicely with Git bash https://git-for-windows.github.io/
- with PLink, hostnames eventually refer to **PuTTY Sessions**
  - ↪ **NEVER** to SSH entries in `~/.ssh/config`
  - ↪ This usage might be hidden... Ex: `$GIT_SSH` etc.

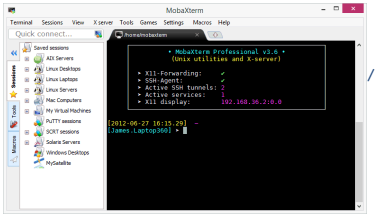
SSH Setup on Windows

- New "**Windows Subsystem for Linux**" Ubuntu Linux binaries
  - ↪ Windows feature, requires to turn on Developer mode
  - ↪ Install Ubuntu within the Microsoft Store
  - ↪ You now have an Ubuntu terminal by running `bash` command
    - ✓ (**optional**) consider installing of-my-zsh and the Powerlevel10k prompt - guide
- Install Windows Terminal
  - ↪ Setup prompt in Windows Terminal and PowerShell - guide
  - ↪ Now open Ubuntu profile to enjoy an fully-featured Linux terminal, incl. native SSH

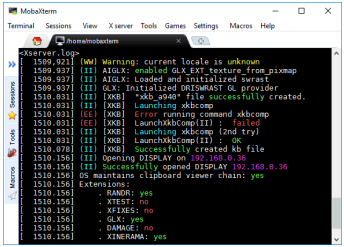


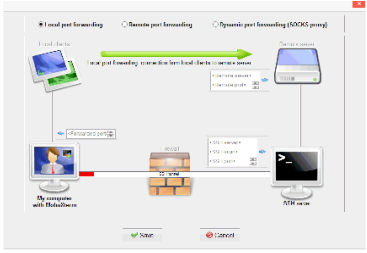
SSH Setup on Windows with X11 support



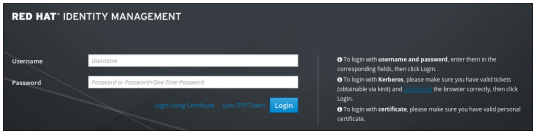


- Use MobaXterm to Run Graphical Linux App
  - ↪ [tabbed] Sessions management
  - ↪ X11 server w. enhanced X extensions
  - ↪ SSH gateway / tunnels wizards
  - ↪ **Run your WSL Ubuntu profile by default**
    - ✓ share Ubuntu home (and SSH setup)



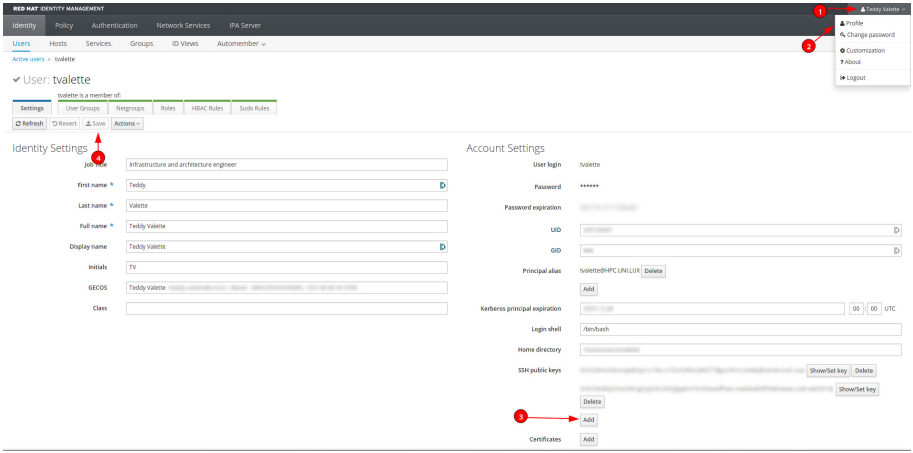


Send your public key to ULHPC



- Connect to **ULHPC IPA portal** (the url is in your registration email)
  - ↪ Go to **your profile** page (step 1 & 2)
  - ↪ Add your **PUBLIC** SSH key (step 3)
  - ↪ **Save** your profile (step 4)

Send your public key to ULHPC



SSH in Practice

~/.ssh/config

```
$> ssh [-X] [-p <port>] <login>@<hostname>
# Example:  ssh -p 8022 svarrette@access-iris.uni.lu
```

```
Host <shortname>
    Port <port>
    User <login>
    Hostname <hostname>
```

- ~/.ssh/config:
  - ↪ Simpler commands
  - ↪ Bash completion `$> ssh iri<TAB>`

SSH in Practice ~/.ssh/config

```
$> ssh [-X] [-p <port>] <login>@<hostname>
# Example:  ssh -p 8022 svarrette@access-iris.uni.lu
```

```
Host *.ext_ul
    ProxyCommand ssh -q iris-cluster \
         -W `basename %h .ext_ul`:%p
# UL HPC Platform -- http://hpc.uni.lu
Host iris-cluster
    Hostname     access-iris.uni.lu
Host aion-cluster
    Hostname     access-aion.uni.lu
Host *-cluster
    User         login #ADAPT accordingly
    Port         8022
    ForwardAgent no
```

```
Host <shortname>
    Port <port>
    User <login>
    Hostname <hostname>
```

- ~/.ssh/config:
  - ↪ Simpler commands
  - ↪ Bash completion `$> ssh iri<TAB>`

SSH in Practice ~/.ssh/config

```
$> ssh [-X] [-p <port>] <login>@<hostname>
# Example:  ssh -p 8022 svarrette@access-iris.uni.lu
```

```
Host *.ext_ul
    ProxyCommand ssh -q iris-cluster \
         -W `basename %h .ext_ul`:%p
# UL HPC Platform -- http://hpc.uni.lu
Host iris-cluster
    Hostname     access-iris.uni.lu
Host aion-cluster
    Hostname     access-aion.uni.lu
Host *-cluster
    User         login #ADAPT accordingly
    Port         8022
    ForwardAgent no
```

```
Host <shortname>
    Port <port>
    User <login>
    Hostname <hostname>
```

- ~/.ssh/config:
  - ↪ Simpler commands
  - ↪ Bash completion `$> ssh iri<TAB>`

```
$> ssh iris-cluster
$> ssh work
$> ssh work.ext_ul
```

SSH Training - Application to Uni.lu HPC Access

Your Turn!

Hands-on SSH

▶ url ◀ | github | src

- **SSH Key Management**
  - ↪ Generate strong RSA (4096 bits) & *[reference]* ED25519 Key pairs
- **Upload the keys on the ULHPC Identity Management Portal**
- Custom **SSH Configuration** `~/.ssh/config`
  - ↪ *(advanced users only)* SSH Agent, SOCKS Proxy

SSH Training - Application to Uni.lu HPC Access

Your Turn!

Hands-on SSH

▶ url ◀ | github | src

- **SSH Key Management**
  - ↪ Generate strong RSA (4096 bits) & *[reference]* ED25519 Key pairs
- **Upload the keys on the ULHPC Identity Management Portal**
- Custom **SSH Configuration** `~/.ssh/config`
  - ↪ *(advanced users only)* SSH Agent, SOCKS Proxy

Hands-on Access to Uni.lu HPC Facility

▶ url ◀ | github | src

- **Connect** to UL HPC (Linux / Mac OS / Unix) (Step 1)

Data transfer using SSH - SCP

- SCP (Secure CoPy):
  - ↪ from your local terminal
  - ↪ plain linear copy

```
$> scp [-P <port>] [remote:]src/ [remote:]dst/
```

```
# If you have configured your SSH client as explained before
scp /tmp/test_file iris-cluster:~/
# If you haven't
scp -P 8022 /tmp/test_file tvalette@access-iris.uni.lu:~/
```

Data transfer using SSH - RSync

- RSync (Remote SYNChronization):
  - ↪ from your local terminal
  - ↪ algorithm to optimize the transfer of files
    - ✓ based on differences between local files and remote ones

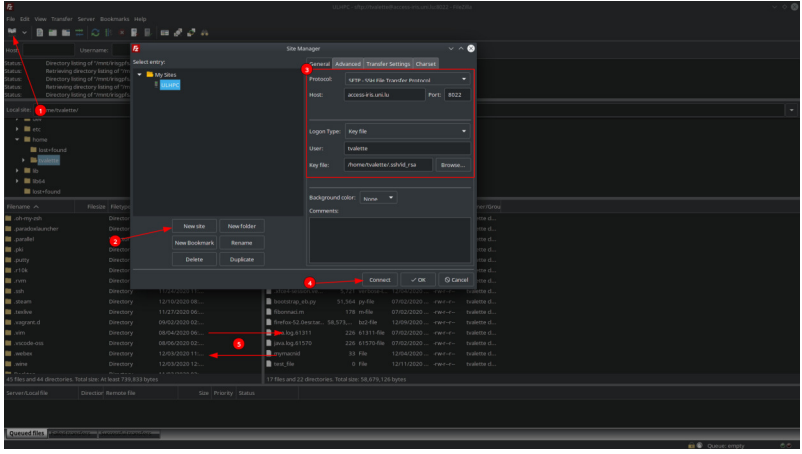
```
$> rsync [-e '<ssh options>'] [remote:]src/ [remote:]dst/
```

```
# If you have configured your SSH client as explained before
rsync /tmp/test_file iris-cluster:~/
# If you haven't
rsync -e 'ssh -p 8022' /tmp/test_file tvalette@access-iris.uni.lu:~/
```

Data transfer using SSH - FileZilla

- FileZilla:
  - ↪ Well known GUI (with a big community) compatible with Windows, Linux and MacOS
  - ↪ Open "Site Manager" (step 1)
  - ↪ Create a new site (step 2)
  - ↪ Fill the following fields (step 3)
    - ✓ Protocol: SFTP - SSH File Transfer Protocol
    - ✓ Host: access-iris.uni.lu
    - ✓ Port: 8022
    - ✓ Logon Type: Key file
    - ✓ User:
    - ✓ Key file:
  - ↪ Connect and drag and drop files from your local to yout remote site and vice versa

Data transfer using SSH - FileZilla



Data transfer Training

Your Turn!

Hands-on Data transfer using SSH

▶ url ◀ | github | src

- **Install your favorite tool**
  - ↪ Mac OS/Linux : scp, rsync, Filezilla
  - ↪ Windows : WSL (scp, rsync), Filezilla
    - ✓ You can also use the MobaXTerm's data transfer feature
- Start transfering your files `wget`, `scp`, `rsync`, MobaXterm(SFTP)

ULHPC School Tutorials

- You **need to use git to get training materials** required required for Practical Sessions
  - ↪ Training materials are stored on Github: `ULHPC/tutorials` repository
  - ↪ You should **clone** this repository:
    - ✓ **locally** on your laptop to have quick access to the material offline
    - ✓ **remotely** within your ULHPC homedir to be able to practice and follow the PS
  - ↪ At any point of time, you can **pull** the latest updates `git pull` (or *(better)* `make up`)

- **Pre-requisite**: make sure you have a working SSH and `git` environment!

ULHPC School Tutorials

- You **need to use git to get training materials** required required for Practical Sessions
  - ↪ Training materials are stored on Github: `ULHPC/tutorials` repository
  - ↪ You should **clone** this repository:
    - ✓ **locally** on your laptop to have quick access to the material offline
    - ✓ **remotely** within your ULHPC homedir to be able to practice and follow the PS
  - ↪ At any point of time, you can **pull** the latest updates `git pull` (or *(better)* `make up`)

- **Pre-requisite**: make sure you have a working SSH and `git` environment!

Your Turn!

- Open your terminal (iTerm2, Terminator, MobaXTerm, ...) & follow next instructions

Keep Sync with Uni.lu HPC Training Material

```
### First time: clone the repo under a meaningfull path
$> mkdir -p ~/git/github.com/ULHPC
$> cd ~/git/github.com/ULHPC
$> git clone https://github.com/ULHPC/tutorials.git
$> cd tutorials
$> make setup
### Next times: pull latest changes
$> cd ~/git/github.com/ULHPC/tutorials
$> make up     # update both branches (production and devel)

# Prepare a dedicated (separated) working directory
$> mkdir -p ~/tutorials/ULHPC-School-2020
$> cd ~/tutorials/ULHPC-School-2020
# create a symbolic link pointing to the tutorial reference material
$> ln -s ~/git/github.com/ULHPC/tutorials  ref.d
# Now  $HOME/tutorials/ULHPC-School-2020/ref.d/ points to reference training material
```

Summary

ULHPC Open OnDemand (OOD) Portal

OPEN OnDemand
openondemand.org

- **Open OnDemand (OOD)**
  - ↪ Web portal
    - ✓ compatible with Windows, Linux and MacOS (thanks to browsers)
    - ✓ use your ULHPC credential to login
  - ↪ Provides a web access to the HPC resources and integrates
    - ✓ a file management system
    - ✓ a job management system (job composer, monitoring your submitted jobs, . . .)
    - ✓ an interactive command-line shell access
    - ✓ interactive apps with graphical desktop environments

ULHPC Open OnDemand (OOD) Portal

OPEN OnDemand
openondemand.org

- **Open OnDemand (OOD)**
  - ↪ Web portal
    - ✓ compatible with Windows, Linux and MacOS (thanks to browsers)
    - ✓ use your ULHPC credential to login
  - ↪ Provides a web access to the HPC resources and integrates
    - ✓ a file management system
    - ✓ a job management system (job composer, monitoring your submitted jobs, . . .)
    - ✓ an interactive command-line shell access
    - ✓ interactive apps with graphical desktop environments

Warning!

- The **ULHPC OOD portal** is **NOT** accessible outside the UniLu network
  - ↪ If you want to use it, you will need to setup a VPN to access the UniLu network *vpn.uni.lu*

ULHPC Open OnDemand (OOD) Portal

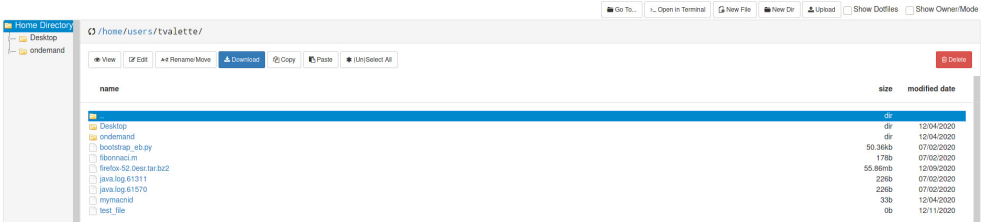
OPEN OnDemand
openondemand.org

- **Open OnDemand (OOD)**
  - ↪ Web portal
    - ✓ compatible with Windows, Linux and MacOS (thanks to browsers)
    - ✓ use your ULHPC credential to login
  - ↪ Provides a web access to the HPC resources and integrates
    - ✓ a file management system
    - ✓ a job management system (job composer, monitoring your submitted jobs, . . .)
    - ✓ an interactive command-line shell access
    - ✓ interactive apps with graphical desktop environments

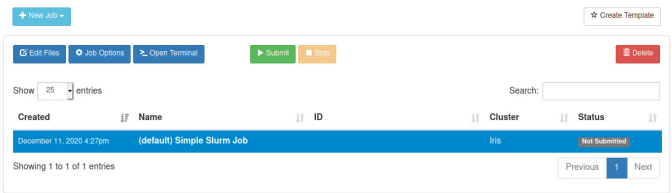
Warning!

- The **ULHPC OOD portal** is **NOT** accessible outside the UniLu network
  - ↪ If you want to use it, you will need to setup a VPN to access the UniLu network *vpn.uni.lu*
  - ↪ The portal is in *(still)* under active development state.
    - ✓ Missing features and bugs can be reported to the ULHPC team via the support portal

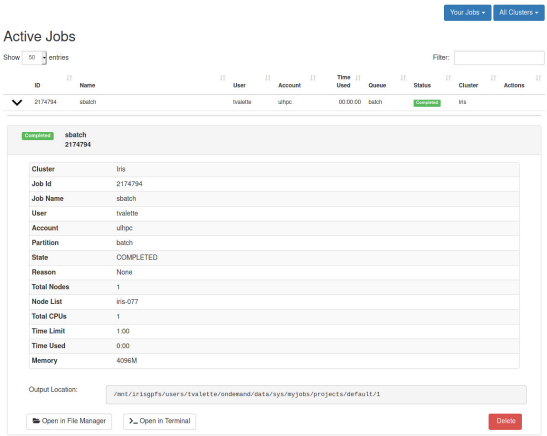
Open OnDemand - File management



Open OnDemand - Job composer

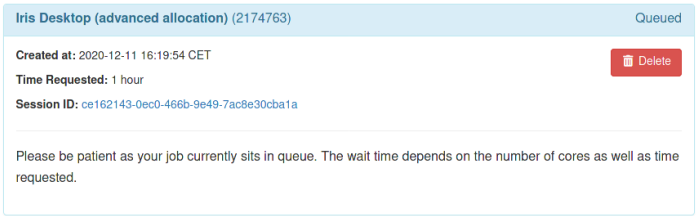


Open OnDemand - Jobs list

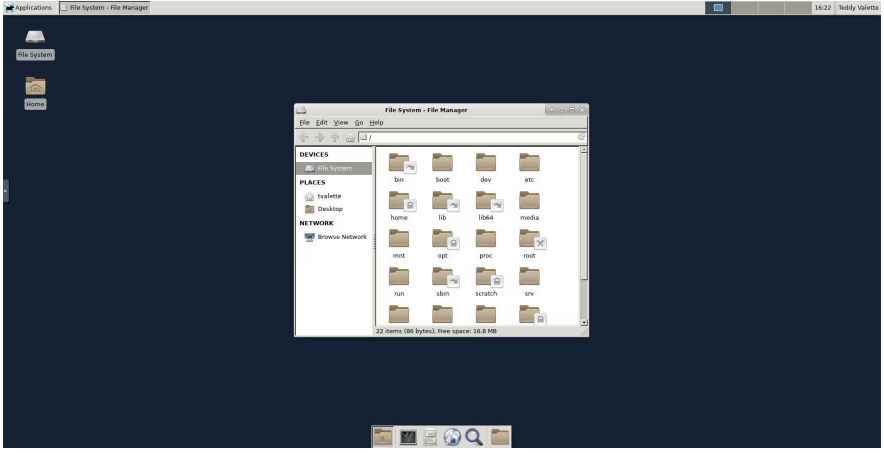


Open OnDemand - Shell access

Open OnDemand - Interactive sessions



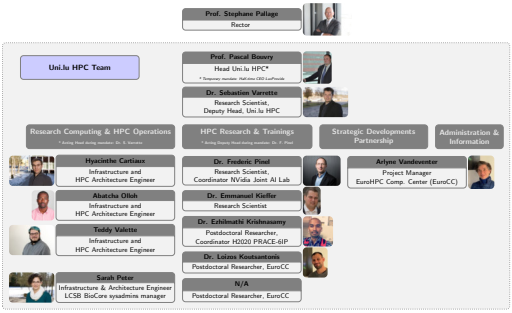
Open OnDemand - Graphical Desktop Environment



Thank you for your attention...

Questions?

High Performance Computing @ Uni.lu



University of Luxembourg, Belval Campus
Maison du Nombre, 4th floor
2, avenue de l'Université
L-4365 Esch-sur-Alzette
mail: hpc@uni.lu

https://hpc.uni.lu