



Vlaanderen
is supercomputing

VSC HPC Introduction

ICTS KU Leuven

VSC staff:

Ehsan Moravveji

Mag Selwa

Wouter van Assche

Geert Jan Bex (Uhasselt)

Jan Ooghe

Material

- Everything is on Github:
<https://hpcleuven.github.io/HPC-intro/>
- Video Recordings
 - Scan the QR code
 - Recommended videos: ~ 2 hrs
 - Optional videos: ~1 hr



What is High Performance Computing?

- using supercomputers to solve advanced computation problems
- Reduce the computation time from days, years, decades, or centuries to minutes, hours, days, or weeks
- The key is parallelism



In practice, it is more like ...



The concept is simple: **Parallelism** = employing multiple processors for a single problem

Outline

- What is the VSC?
- What is a cluster?
- wICE/Genius Cluster
- Storage
- Login nodes & MFA
- Connection Setup
- Software environment
- How to submit jobs?
- Dedicated hardware
- Optional material
 - Conda for Python and R
 - Worker Framework





VSC

(Vlaams Supercomputer Centrum)

VSC PARTNERSHIP



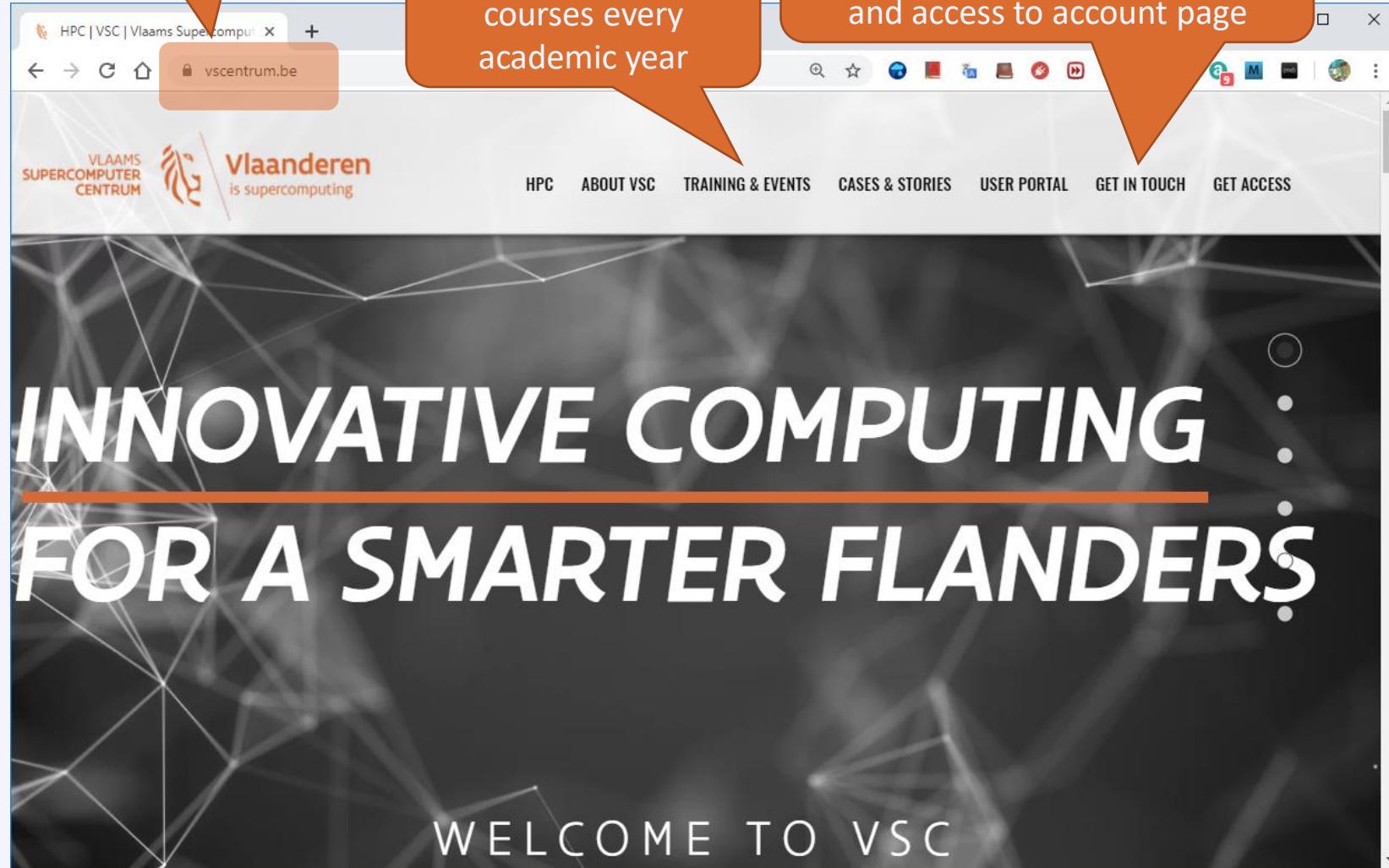
Supported by



VSC HPC Environments



www.vscentrum.be





Welcome to the User Portal

Here you can find the gateway to the User documentation of the Vlaams Supercomputer Centrum

[User documentation page](#)

Manage your VSC-account
partner institute account required

[VSC account page](#)

Search your keywords here;
e.g. qsub

Welcome to VSC documentation

VSC documentation 1.0 documentation »

next | index

Next topic
Getting access

This Page
Show Source

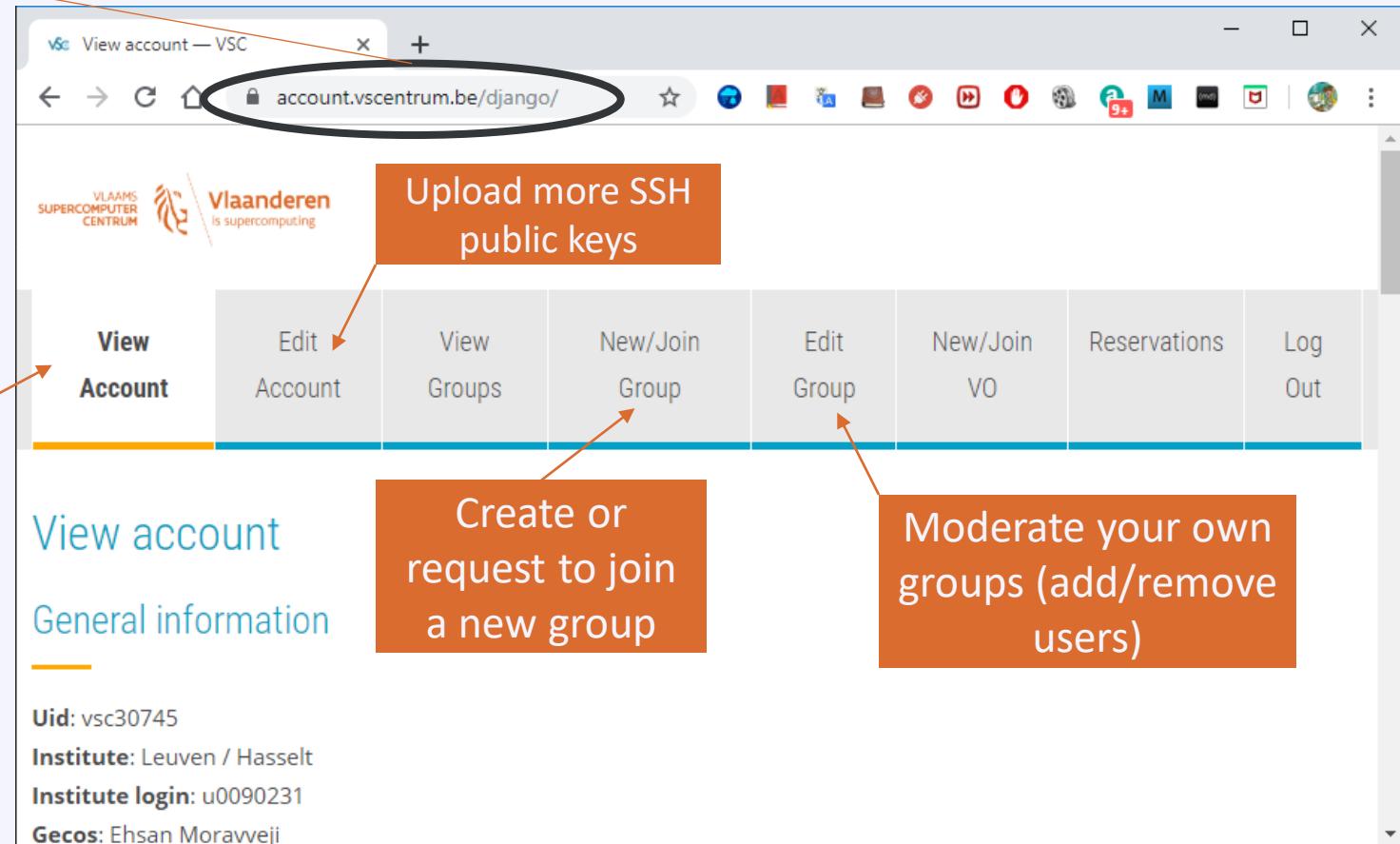
Quick search

Go

- Getting access
 - VSC accounts
 - How to request an account?
 - Next steps
 - Additional information
- Access and data transfer
 - Logging in to a cluster
 - Data storage
 - Transferring data
 - GUI applications on the clusters
 - VPN
- Software stack
 - Using the module system
 - Specialized software stacks
- Running jobs
 - Job script
 - Submitting and monitoring a job
 - Job output
 - Troubleshooting
 - Advanced topics
- Software development

v: latest

To manage your
VSC account:
account.vscentrum.be



Support and Services

Basic support

- Helpdesk (hpcinfo@kuleuven.be)
- Monitoring and reporting

Application support

- Installation and porting
- Optimisation and debugging
- Benchmarking
- Workflows and best practices

Training

- Documentation and tutorials
- Scheduled trainings / workshops
- On request workshops
- One-to-one sessions

Become a VSC user

- Create a secure (4096 bit) [SSH key pairs](#)
Upload it on the account page: www.account.vscentrum.be
- You need to [request a VSC account](#)
Normally processed swiftly
- Request [introductory credits](#) (2M free credits for 6 months)
- Request [project credits](#) (for supervisors and project leaders)
 - You need to create a VSC group
 - Add users to the group to give them access to use credits
 - Fill out the request form
- Extra storage requests
 - Scratch extension: free of charge
 - Staging fileset: 20 € per TB per year
- All service costs (compute and storage) are all explained
Go to ICTS service catalogus: <https://icts.kuleuven.be/sc>
Click on [High Performance Computing](#) (NL/EN)

The screenshot shows a web browser window titled 'View account — VSC'. The URL is 'account.vscentrum.be/django...'. The page header includes the Vlaams Supercomputer Centrum logo and the text 'Vlaanderen is supercomputing'. A navigation menu at the top has tabs for 'View Account' (which is selected), 'Edit Account', 'View Groups', 'New/Join Group', 'Edit Group', 'New/Join VO', 'Reservations', and 'Log Out'. Below the menu, there are sections for 'View account' and 'General information'.

The screenshot shows a web browser window titled 'ICTS Servicecatalogus'. The URL is 'icts.kuleuven.be/sc'. The page header includes the KU Leuven logo and the text 'ICTS SERVICECATALOGUS'. It features a 'MENU' button, language options ('NL EN'), and a user profile ('Ehsan Moravveji'). The main content area includes sections for 'Home', 'ICTS SERVICECATALOGUS', 'WAT IS DE ICTS SERVICECATALOGUS?', 'OPGELET' (with a note about netto prijzen), and 'ZOEKEN' (Search). There is also a graphic of a computer mouse on a blue book.

VSC training

- Introductory

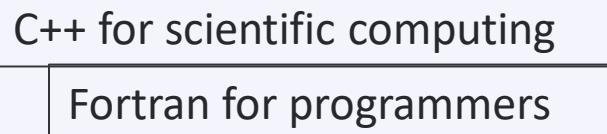


Linux

HPC intro



- Intermediate



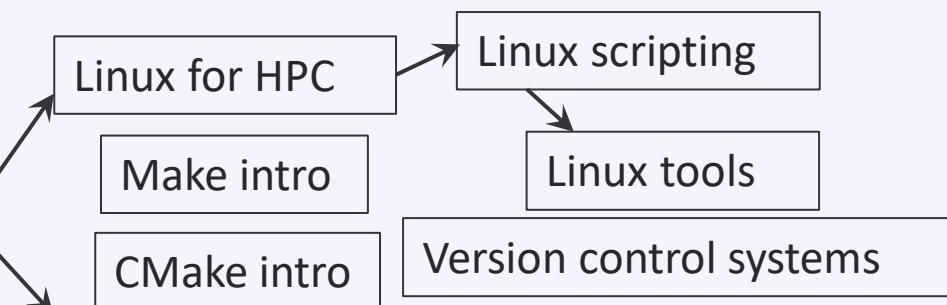
C

- Advanced

High Performance Python

- Specialized track

?

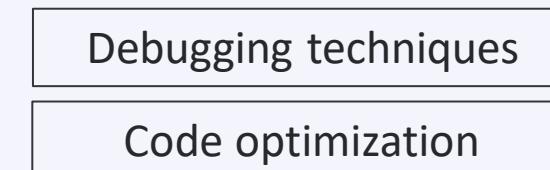


- Python as a second language
- Python: System programming
- Scientific Python
- Python for Software engineering
- Python for data science
- Python for machine learning

worker/atools

Julia good bad ugly

MPI OpenMP



To Acknowledge VSC in publications

Why?

- a contractual obligation for the VSC
- helps VSC secure funding
- you will benefit from it in the long run

At KU Leuven

- add the relevant papers to the virtual collection "High Performance Computing" in Lirias

In het nederlands

De rekeninfrastructuur en dienstverlening gebruikt in dit werk, werd voorzien door het VSC (Vlaams Supercomputer Centrum), gefinancierd door het FWO en de Vlaamse regering – departement EWI.

In English

The computational resources and services used in this work were provided by the VSC (Flemish Supercomputer Center), funded by the Research Foundation - Flanders (FWO) and the Flemish Government – department EWI.



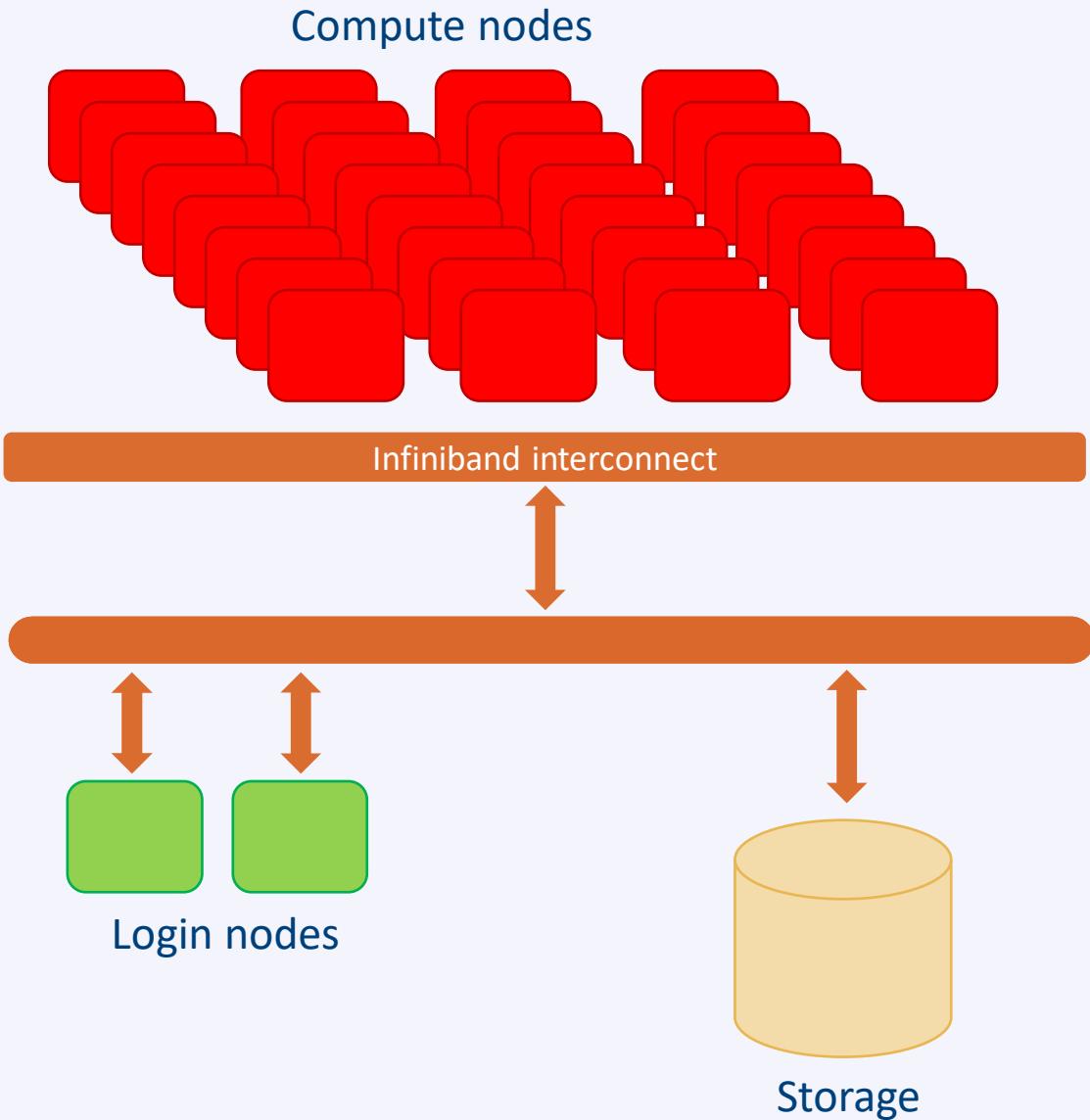
Tier-2 Clusters

What is a Cluster?

A cluster is comprised of **many powerful interconnected compute nodes**

Some Terminology:

	What	Components
Cluster	The primary part of a cluster are its many powerful interconnected compute nodes.	Compute nodes Login nodes Storage space Interconnect Network ...
Node	A single computer connected with other nodes via an interconnect network	CPU, Storage, Memory, ...
CPU	Processing Component of a computer, also called the processor .	Controllers, Cache Memory, Processing Cores
Core	Processing element of the CPU .	Control Unit, Arithmetic-Logic Unit, Memory.



Tier-2 Clusters @ KU Leuven

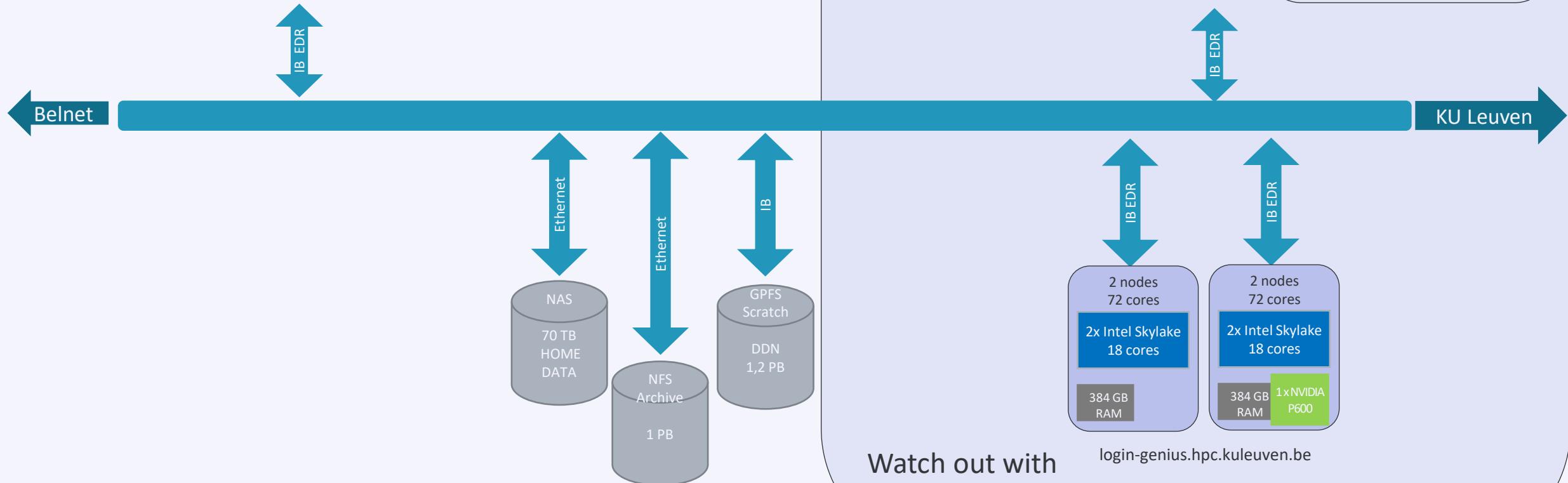
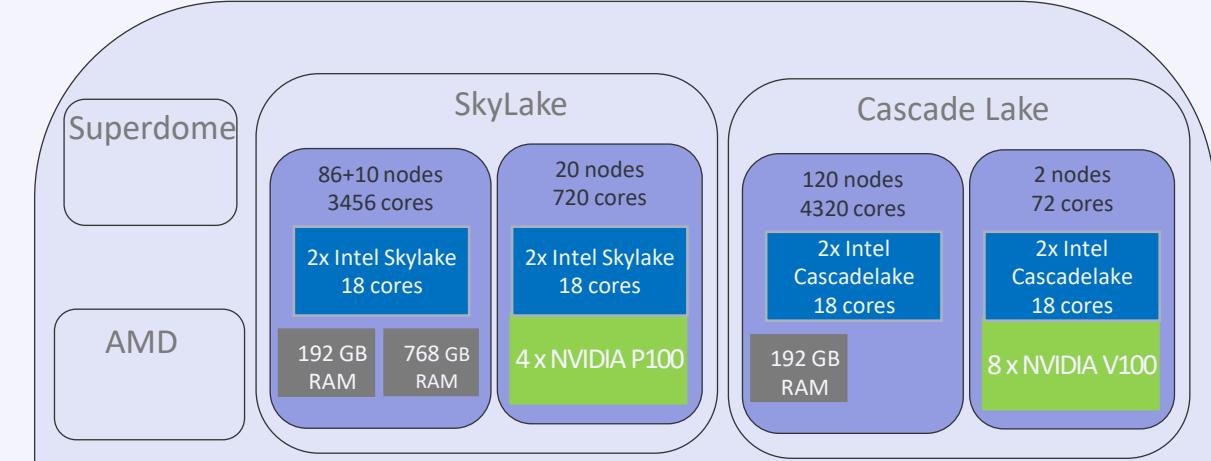
Genius (since 2018)
250 nodes; 8,936 cores



wICE (since 9/2022)
186 nodes; 13,392 cores



wICE (since 9/2022)
186 nodes; 13,392 cores

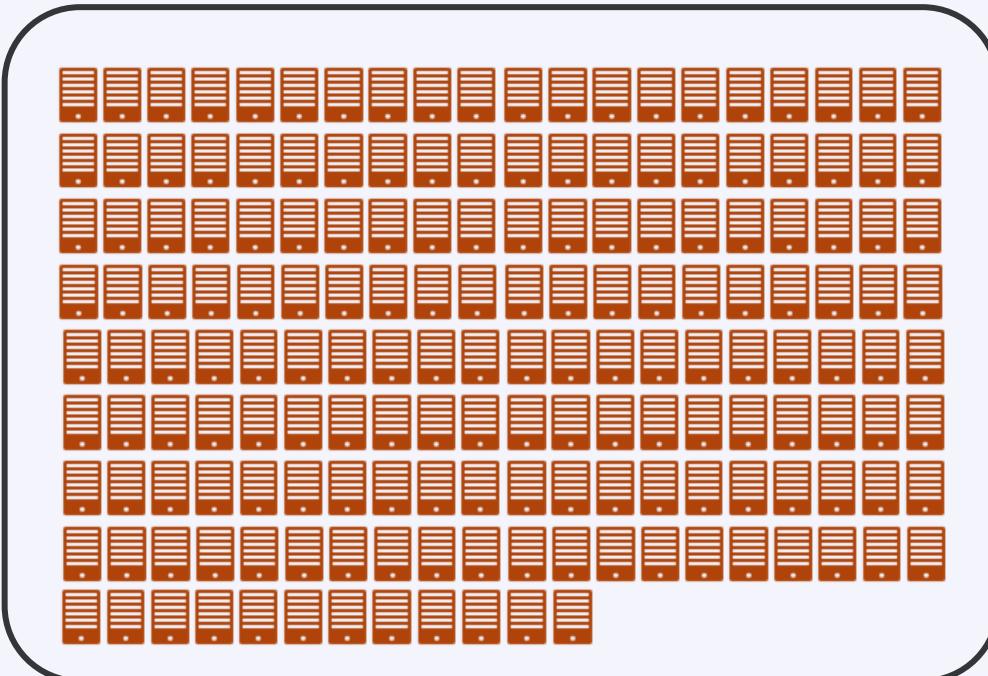


Genius

Tier-2 Overview

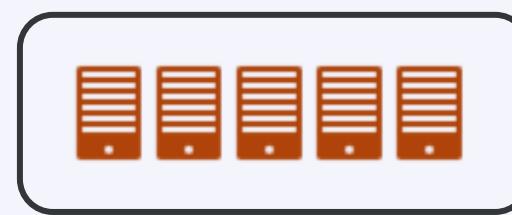
wICE

Compute nodes



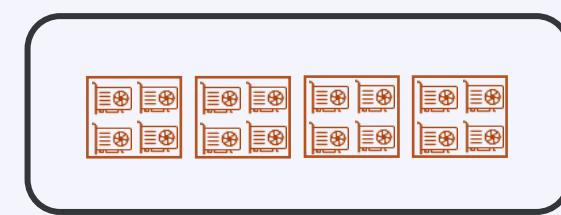
172x IceLake 72c 256 GB

Large memory nodes



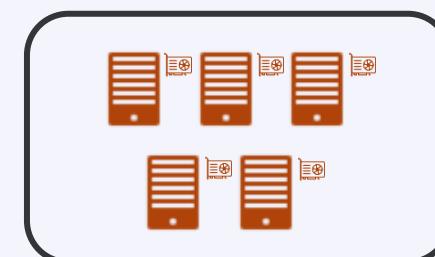
5x IceLake 72c 2048 GB

GPU nodes



4x IceLake 72c 512 GB
4 A100 SXM4 80GB

Interactive nodes



5x IceLake 64c 512 GB
1 A100 80GB

No Dedicated
Login Nodes

Tier-2 Cluster: wICE

Type of node	CPU type	Inter-connect	# cores	installed mem	local discs	# nodes
Icelake	Xeon 8358	IB HDR-100	72	256 GB	960 GB	172
Icelake large mem	Xeon 8358	IB HDR-100	72	2048 GB	960 GB	5
Icelake GPU	Xeon 8358 4xA100 SXM2 80GB	IB HDR-100	72	512 GB	960 GB	4
Icelake Interactive	Xeon 8358 1xA100 SXM2 80GB	IB HDR-100	64	512 GB	960 GB	5

Storage

Overview of the storage infrastructure

- Your files are owned only by you.
Other VSC users have no permission to read/write/execute your files (POSIX)
- A VSC account has 3 default storages (free of charge)
 - \$VSC_HOME
 - \$VSC_DATA
 - \$VSC_SCRATCH
- You can additionally request staging storage
- Different storage volumes have different:
 - mount point
 - size and performance
 - use case
 - backup and maintenance policy
- More info on [ICTS Service Catalog](#) (EN/NL)

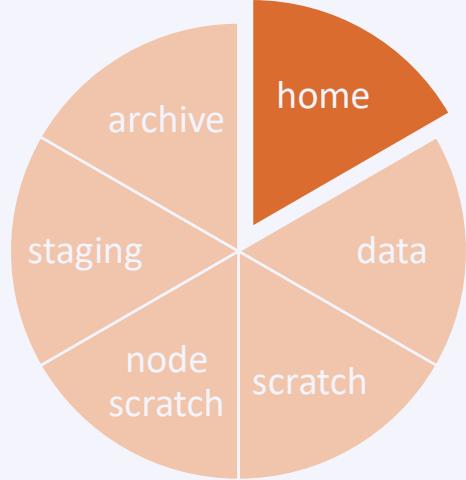
Storage

- [Request form for extra storage](#)
- [More information](#)
- Do not** use /tmp
It is only 10 GB and is reserved for the OS
and root processes
Your application can crash if using /tmp
- You are automatically logged into your home
folder upon login.
Make sure you immediately go to your other
storages, e.g.
`$ cd $VSC_DATA`
- Always check your storage balance using
myquota command

Example

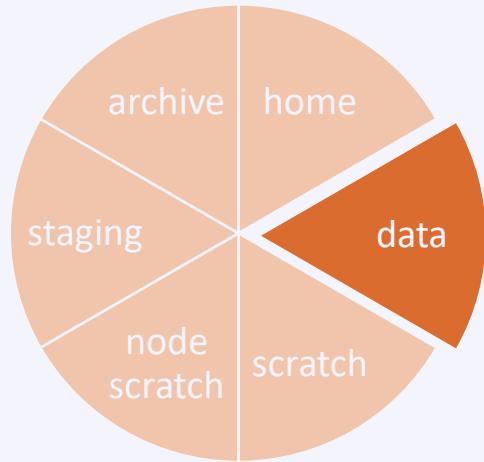
```
$ myquota
file system $VSC_HOME
    Blocks: 1479M of 3072M
    Files: 12934 of 100k
file system $VSC_DATA
    Blocks: 12G of 75G
    Files: 1043k of 10000k
file system $VSC_SCRATCH
    Blocks: 15M of 1.5T
```

Storage



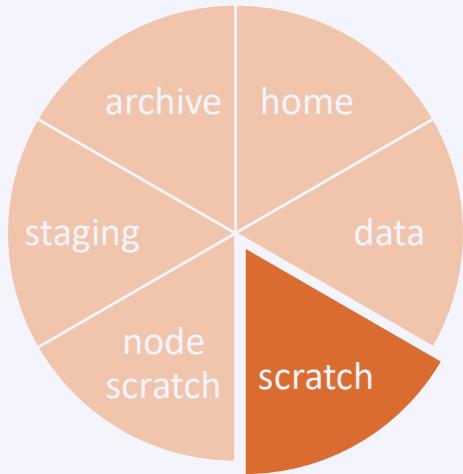
Storage	home folder
Env. Variable	\$VSC_HOME
Filesystem Type	NFS
Access	Global
Backup	Hourly, daily, weekly (until last month) inside the .snapshot folder.
Default Quota	3 GB
Extension	Not possible
Usage	Only storing SSH keys, config files
Remarks	<ul style="list-style-type: none">- Stay away from using it- Can easily overflow:<ul style="list-style-type: none">+ Your jobs may crash+ Login issues

Storage



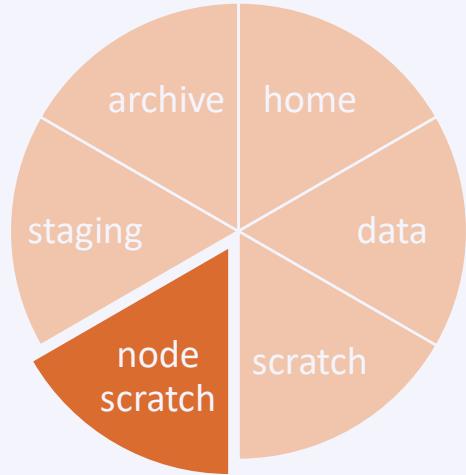
Storage	data folder
Env. Variable	\$VSC_DATA
Filesystem Type	NFS
Access	Global
Backup	Hourly, daily, weekly (until last month) inside the .snapshot folder.
Default Quota	75 GB
Extension	On purchase
Usage	Your data, code, software, results
Remarks	<ul style="list-style-type: none">- Permanent storage for initial/final results- Not optimal for intensive or parallel I/O

Storage



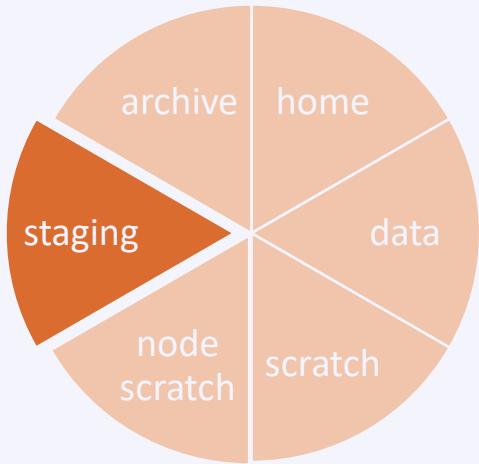
Storage	scratch folder
Env. Variable	\$VSC_SCRATCH
Filesystem Type	Lustre
Access	Global
Backup	delete after 30 days from last access
Default Quota	500 GB
Extension	Free
Usage	Intensive, parallel I/O, temporary files
Remarks	<ul style="list-style-type: none">- Recommended storage for all jobs- Copy scratch files to VSC_DATA or local storage after jobs are done- Deleted files cannot be recovered

Storage



Storage	Node scratch folder
Env. Variable	\$VSC_SCRATCH_NODE
Filesystem Type	Lustre
Access	On compute node, only at runtime
Backup	None
Default Quota	591 GB
Extension	Read about beeOND
Usage	Temporary storage at runtime
Remarks	<ul style="list-style-type: none">- Fastest I/O, attached to the node- Is cleaned after job terminates- Copy the data to your home, scratch, or staging before job ends

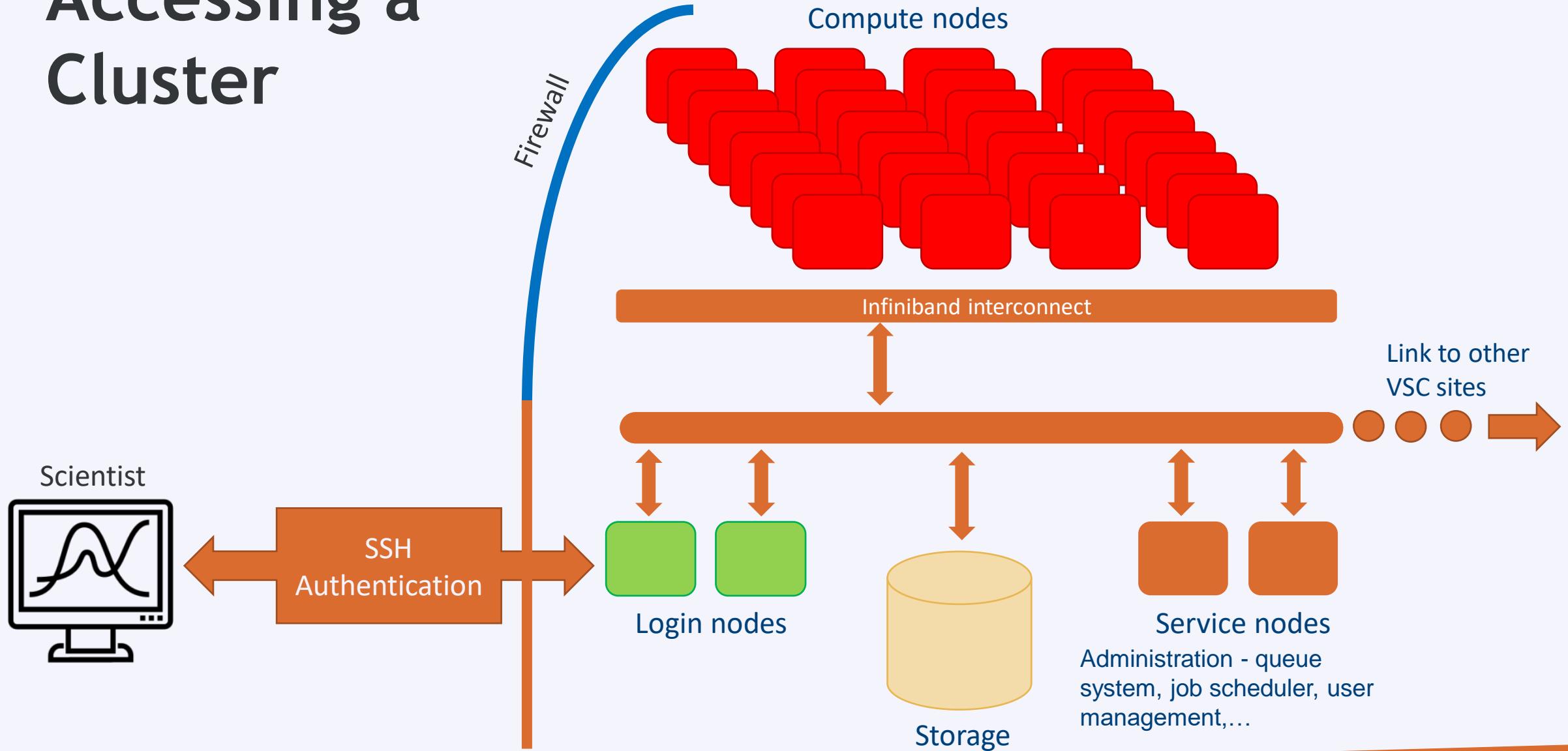
Storage



Storage	Staging folder
Path	/staging/leuven/stg_000XX
Filesystem Type	Lustre
Access	On demand, only Tier-2@KUL
Backup	None
Default Quota	None
Extension	On purchase, from 1 TB
Usage	Permanent; share with a group
Remarks	<ul style="list-style-type: none">- Accessible from login/compute nodes- Fast, parallel I/O

Login Nodes & MFA

Accessing a Cluster



Using Login Nodes

- To develop code
- To check your storage and credit balance
- To manage jobs (submit, check status, debug, resubmit, ...)
- To move data around
 - within VSC: use data, scratch, staging, archive
 - outside VSC: copy/sync from/to your local storage
- To pre-process or post-process your data/jobs
- To visualize your data
- To share files/folders

Tips

Command
\$ top
Show use of
the system

- Compile your software on compute nodes (via e.g. interactive job)
- Login nodes are shared resources
- Do not** execute heavy-lifting tasks (core, memory)
- Instead, submit jobs
- Check \$ slurmtop to see how busy the cluster is

Warning

Login Hosts on Different Machines/partitions

- Windows: [PuTTY](#) or [MobaXterm](#) or NX
Linux/Mac: [terminal](#) or NX
- To login, you need an active VSC number and a hostname
\$ ssh -X vscXXXXX@<hostname>

Cluster / Partition	<hostname>	Remark(s)
Genius	login.hpc.kuleuven.be	Recommended
	login{1,2}-tier2.hpc.kuleuven.be	No GPU
	login{3,4}-tier2.hpc.kuleuven.be	Nvidia Quadro P6000
wICE		Accessible only via Genius

This is not a computationa GPU – single precision only – visualization purpose



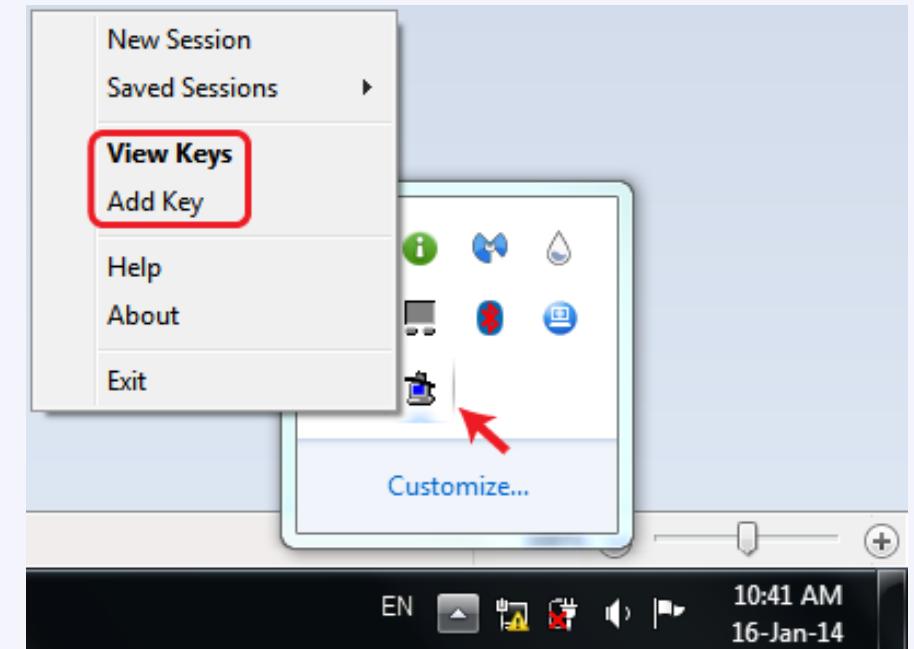
Login Setup

- PuTTY
- Terminal

Activate Your SSH Agent

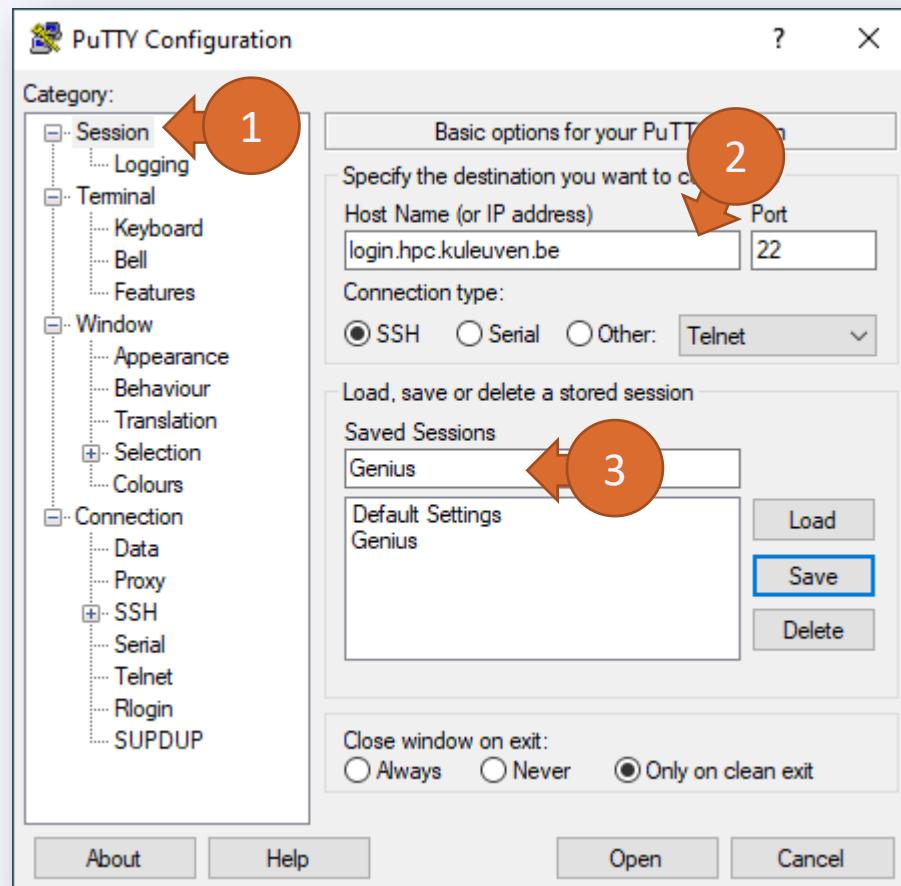
(Windows only)

- When you install PuTTY, the agent called **Pageant** is automatically installed
- Open Pageant; it hides inside your bottom-right tray
- Click on “Add Key” and brows to your private key(s) folder
- After choosing a key, you are asked for a passphrase
- If successful, agent always remembers your SSH keys and Certificates (Multi-Factor Authentication)

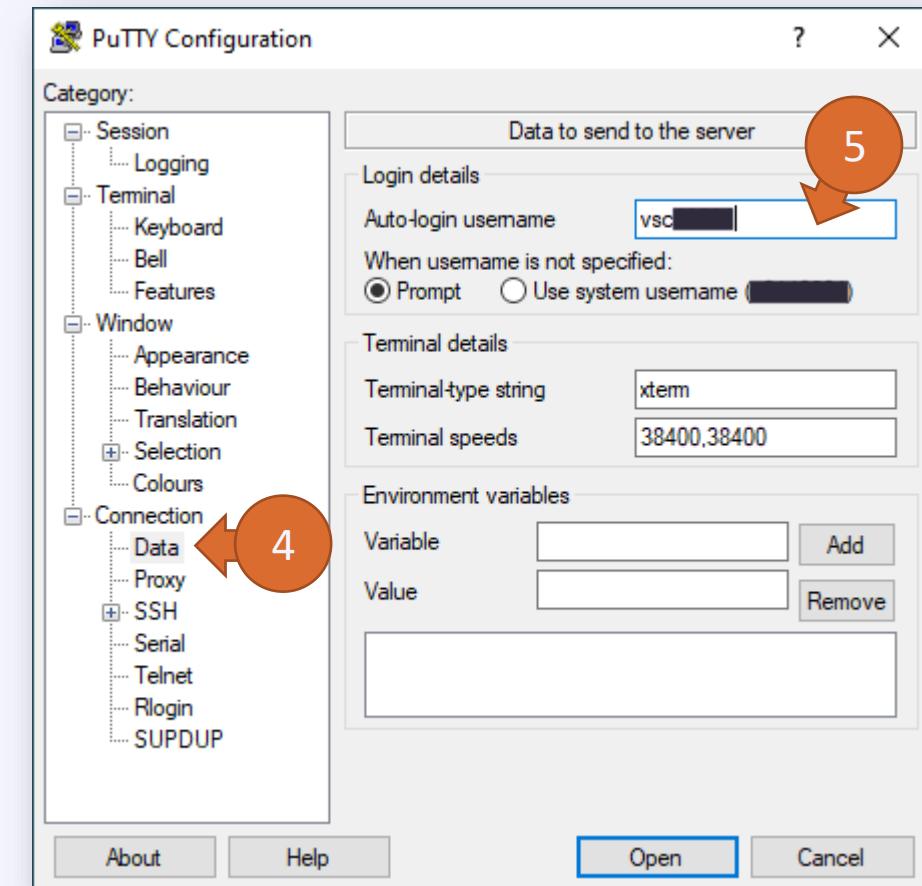


Setup PuTTY

(Windows only)

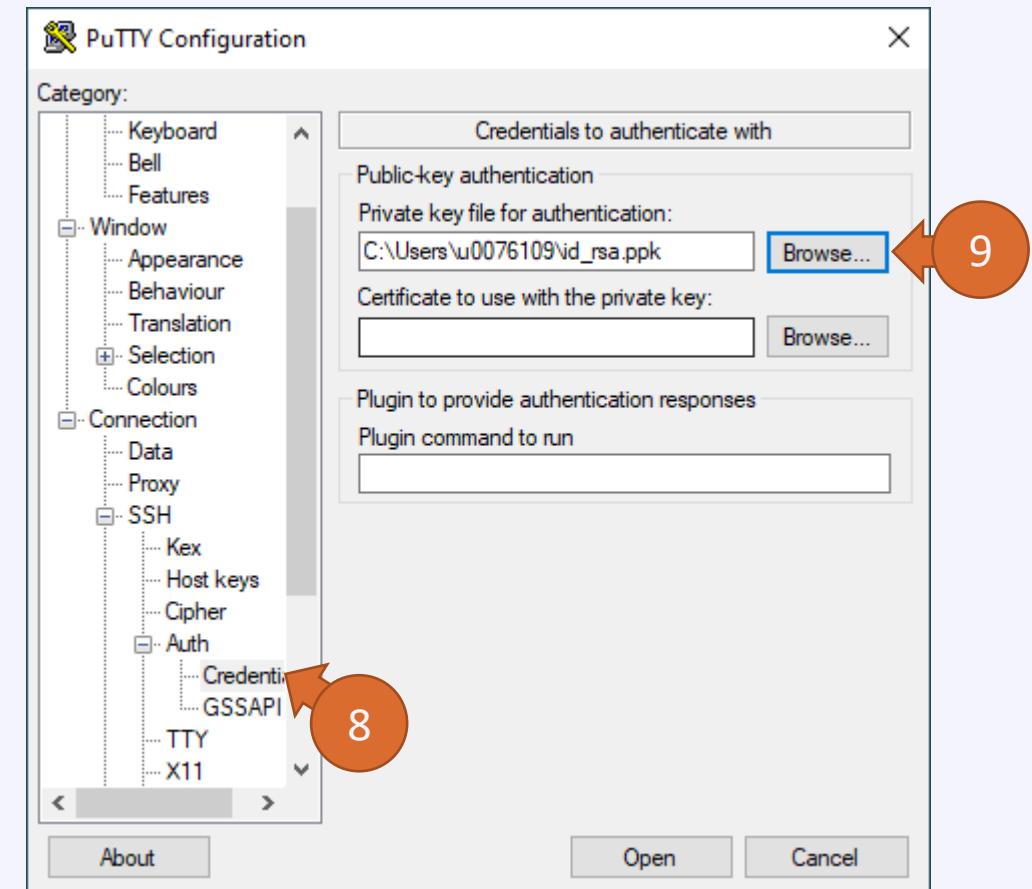
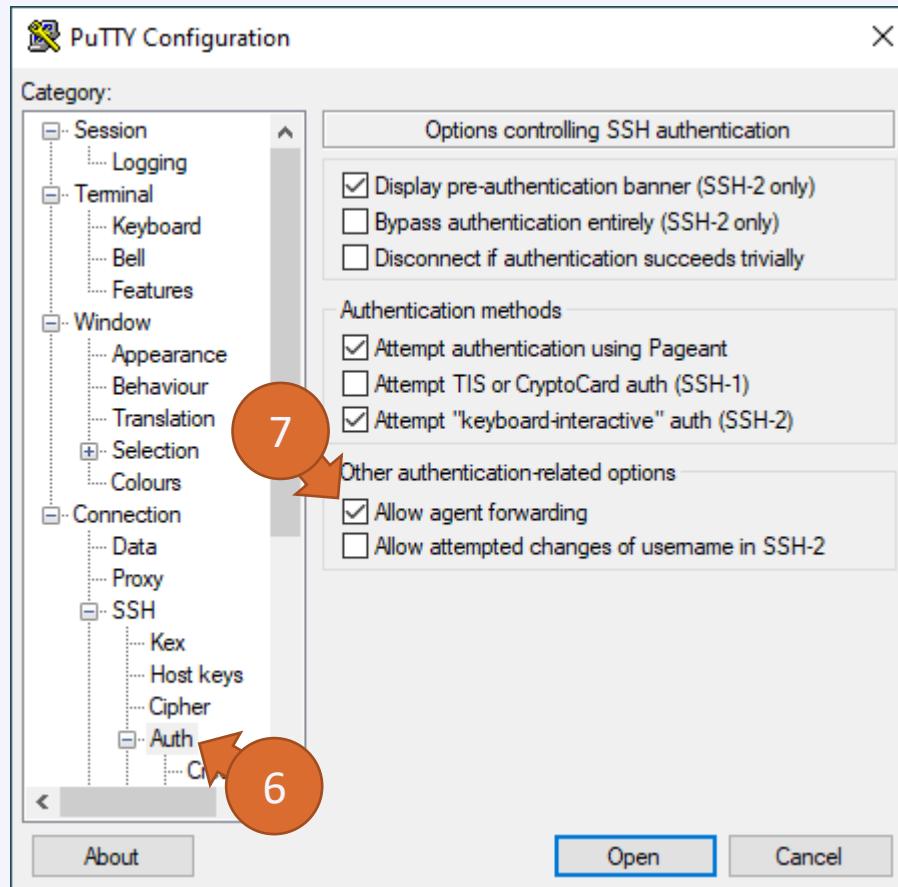


Host Name:
login.hpc.kuleuven.be



Setup PuTTY

(Windows only)



9

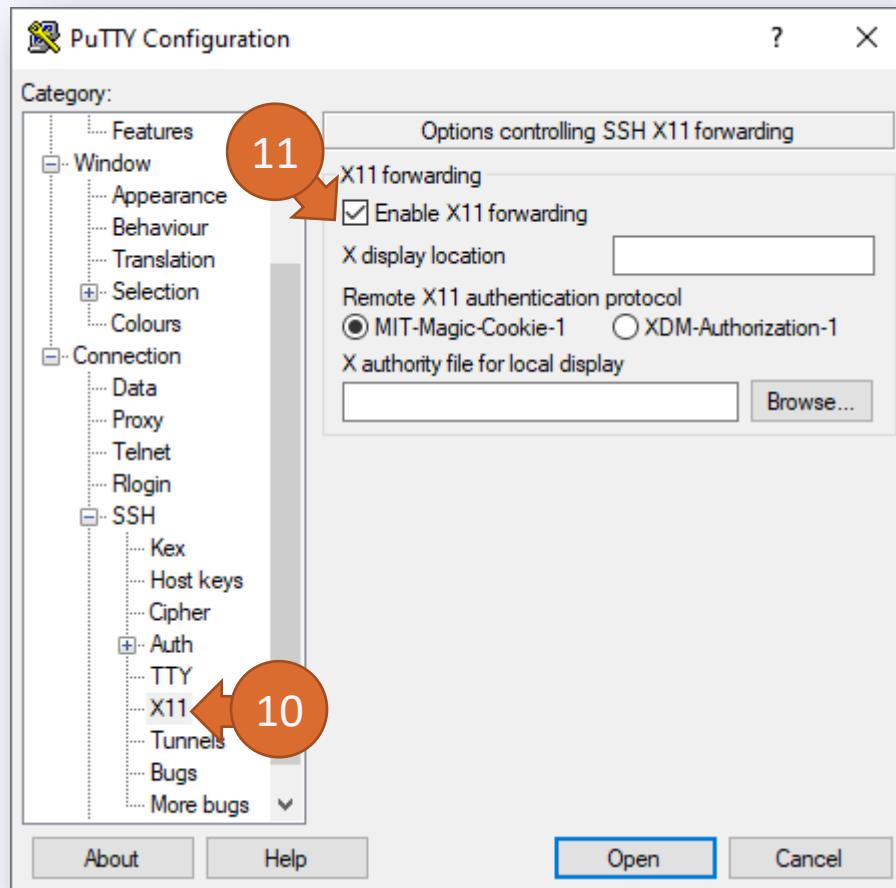
8

7

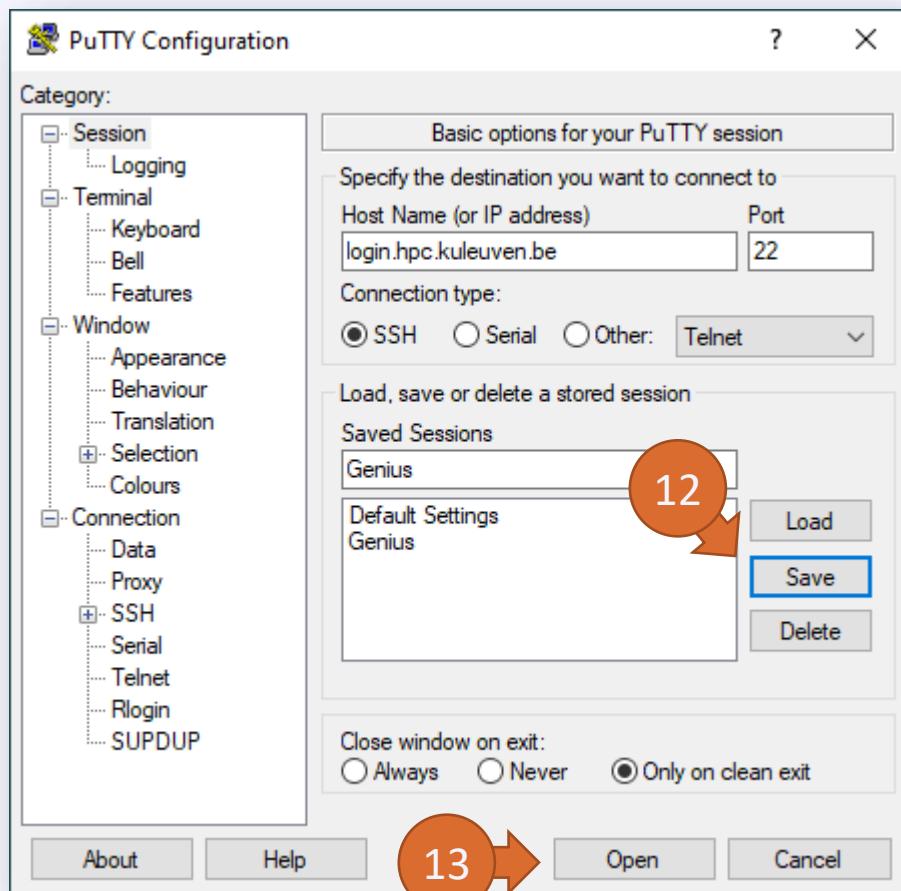
6

Setup PuTTY

(Windows only)



Setup PuTTY (Windows only)



If PuTTY asks for **password**, exit immediately, and check the path to your private key. Else you will be blocked for 24h.

The image contains three stacked screenshots of a PuTTY session window titled 'login-genius.hpc.kuleuven.be - PuTTY'.
The top screenshot shows the initial connection message: 'Using username "vsc[REDACTED"]'. It then prompts for a 'Passphrase for key "the key pair that is used for my VSC account"'. A red arrow points from the text 'If PuTTY asks for password, exit immediately, and check the path to your private key. Else you will be blocked for 24h.' to this passphrase prompt.
The middle screenshot shows the user entering the passphrase 'vsc[REDACTED]' followed by a command to open a URL: '| Please open https://firewall.vscentrum.be/PDindFdqlcpIgl9qXPo to complete log > on.'. A red arrow points from the previous screenshot's warning text to this step.
The bottom screenshot shows the final output after the passphrase is entered. It includes a small logo of a person running, the server information 'Informatie over deze server (FQDN): tier2-p-login-2.genius.hpc.kuleuven.be', and the command history 'tier2-p-login-2 [Oct/07 13:15] vsc[REDACTED]@tier2-p-login-2 ~ \$'. A large red curved arrow starts from the warning text and points to the bottom screenshot, indicating the progression of the setup process.

Connecting via Terminal

(Linux and Mac)

- Use ssh to connect:

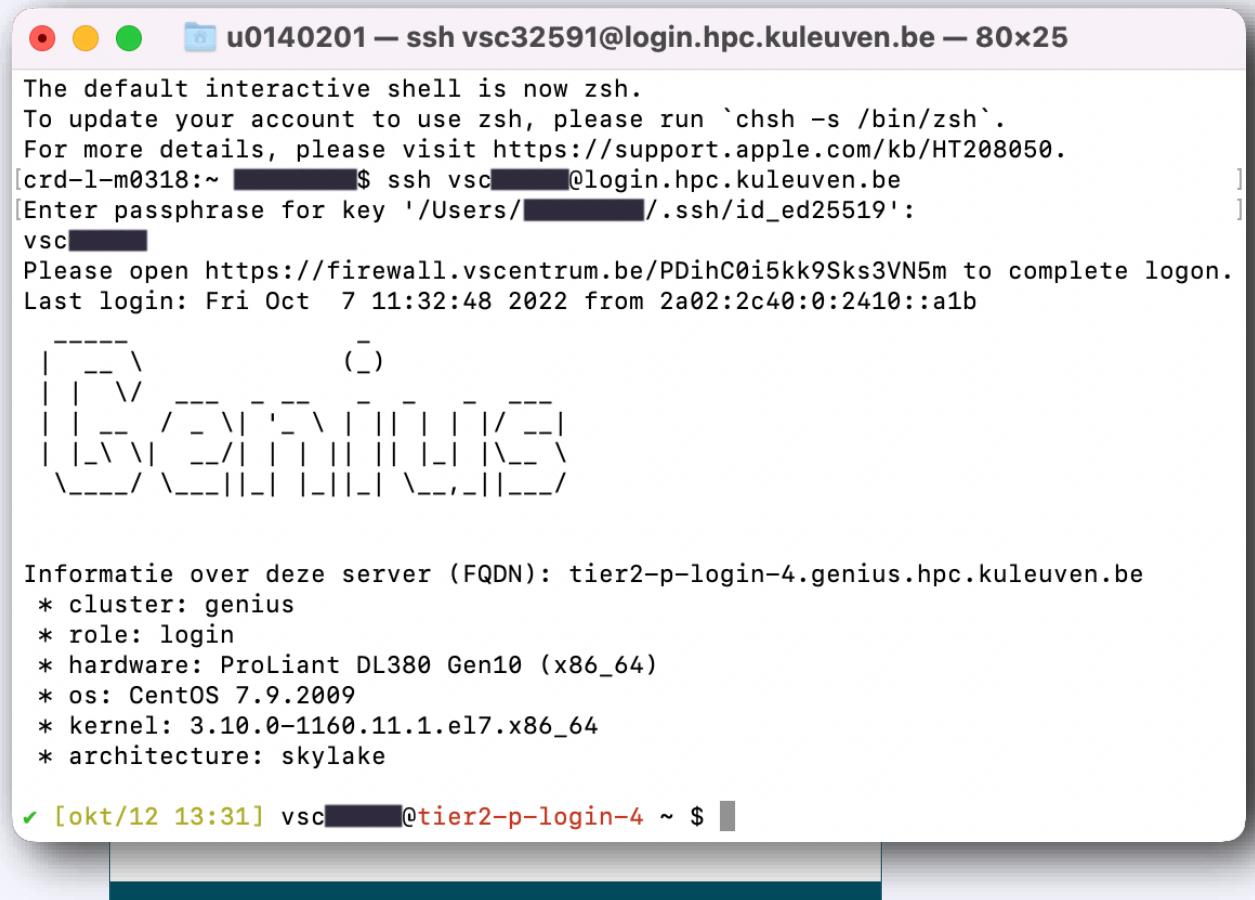
```
$ ssh vscXXXXX@<host_name>
```

- If key not found:

```
$ ssh -i </path/to/keyfile> ...
```

If asked for **password**,
please stop connecting
and contact support,
otherwise after a few
attempts you will be
blocked for 24h.

Host Name:
login.hpc.kuleuven.be



The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit <https://support.apple.com/kb/HT208050>.
[crd-l-m0318:~] \$ ssh vscXXXXX@login.hpc.kuleuven.be
[Enter passphrase for key '/Users/vscXXXXX/.ssh/id_ed25519':
vscXXXXX
Please open <https://firewall.vscentrum.be/PDihC0i5kk9Sks3VN5m> to complete logon.
Last login: Fri Oct 7 11:32:48 2022 from 2a02:2c40:0:2410::a1b
[----] [----] [----] [----] [----] [----] [----]
[----] [----] [----] [----] [----] [----] [----]
[----] [----] [----] [----] [----] [----] [----]
[----] [----] [----] [----] [----] [----] [----]
[----] [----] [----] [----] [----] [----] [----]
[----] [----] [----] [----] [----] [----] [----]
[----] [----] [----] [----] [----] [----] [----]
[----] [----] [----] [----] [----] [----] [----]
Informatie over deze server (FQDN): tier2-p-login-4.genius.hpc.kuleuven.be
* cluster: genius
* role: login
* hardware: ProLiant DL380 Gen10 (x86_64)
* os: CentOS 7.9.2009
* kernel: 3.10.0-1160.11.1.el7.x86_64
* architecture: skylake
✓ [okt/12 13:31] vscXXXXX@tier2-p-login-4 ~ \$

Connecting via Terminal

With SSH Agent (Linux and Mac)

- Check your SSH Agent. Is your SSH key found?

```
$ ssh-add -l
```

- If your SSH Agent is not running:

```
eval $(ssh-agent)
```

- If your key is not found, add it to the Agent:

```
$ ssh-add </path/to/keyfile>
```

- Use ssh to connect:

```
$ ssh vscXXXXX@<hostname>
```

Host Name:
login.hpc.kuleuven.be



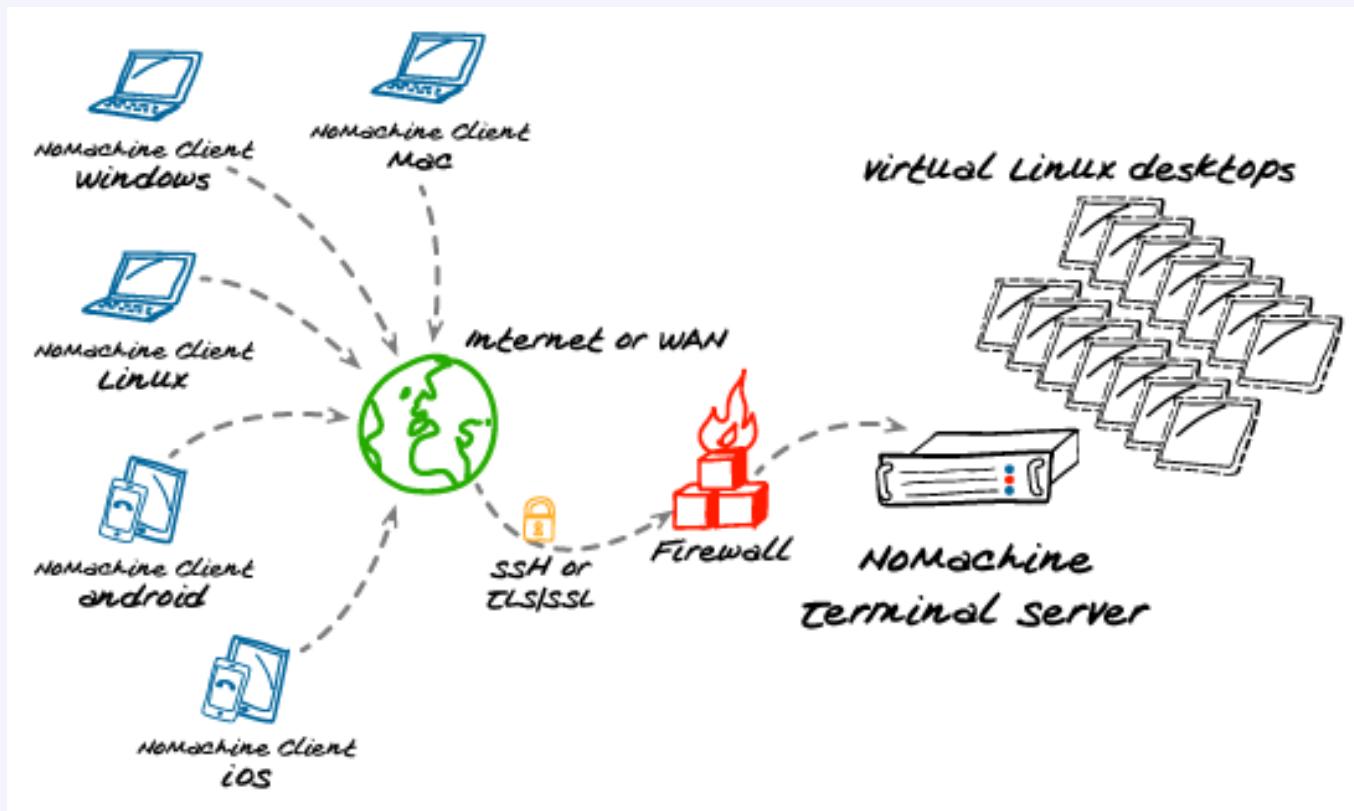
NX

- About
- Setup

NX – The Graphical Login

NX is:

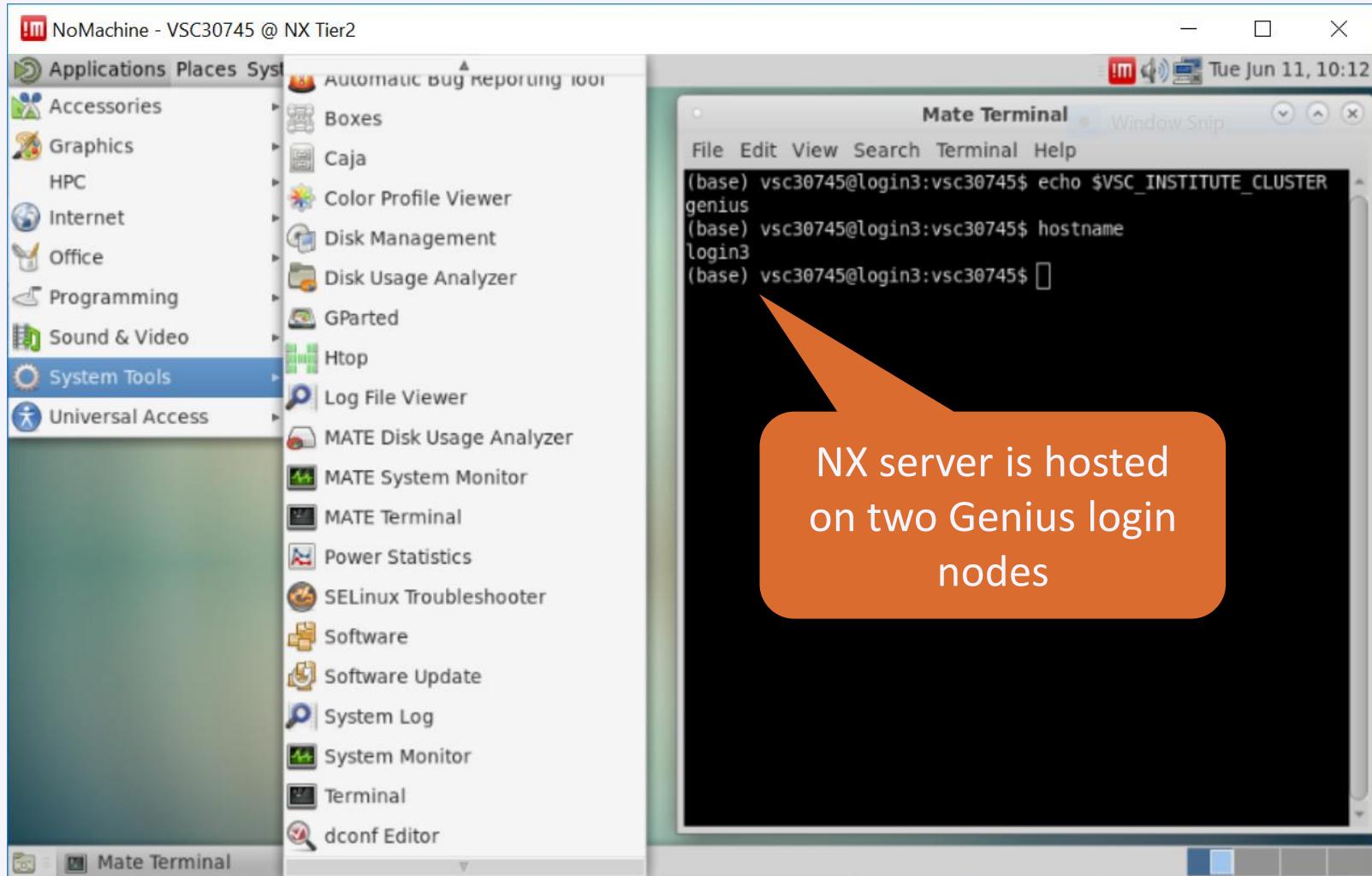
- a graphical remote desktop server
- login to VSC
- start a virtual Linux instance(s)
- Client: [NoMachine](#)
- [Configuration Guide](#)
- Using OpenSSH key
[Convert your key](#)

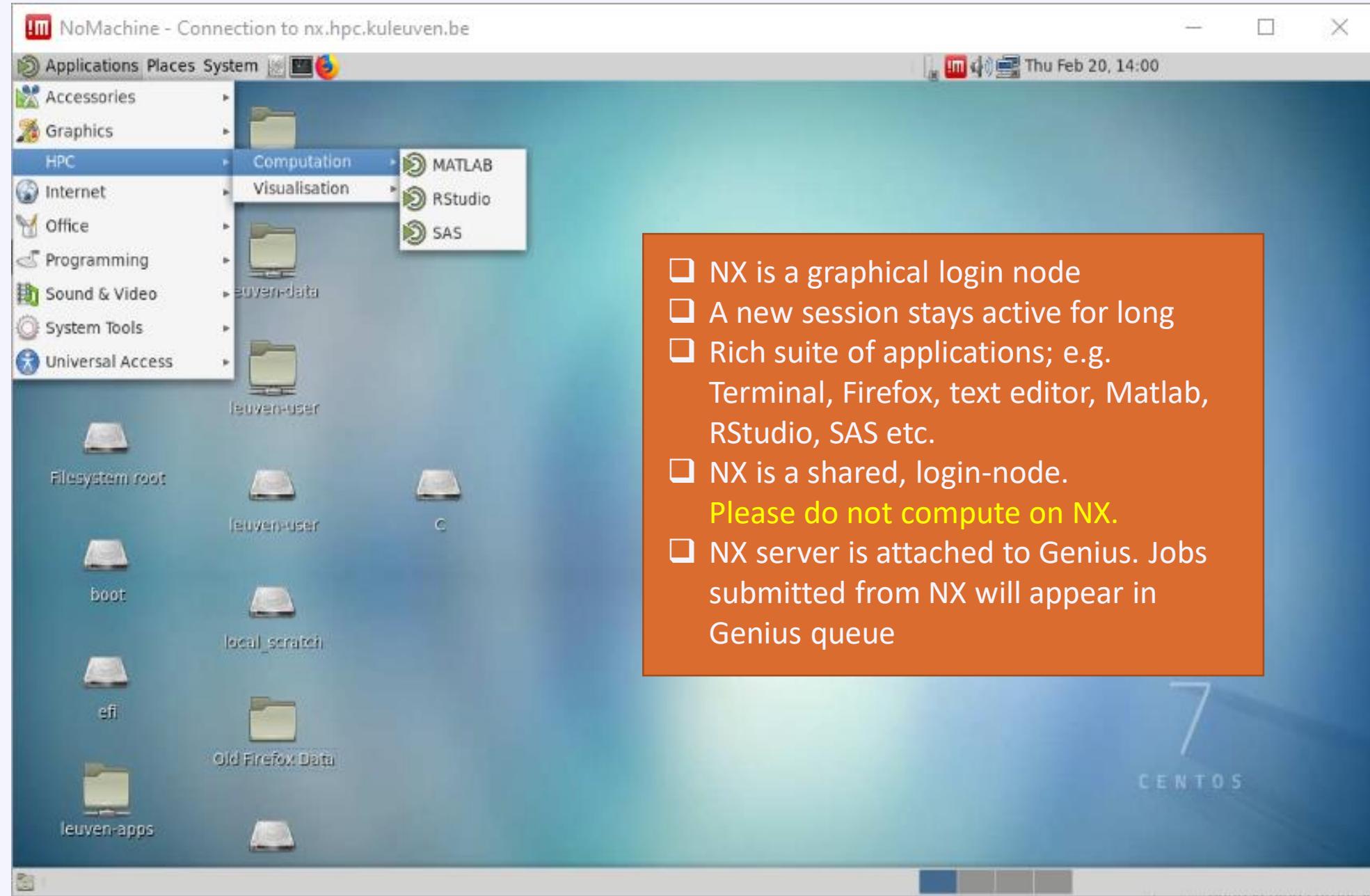


Advantages of NX

- ❑ Graphical login
- ❑ A session stays alive/active even if you close your client (e.g. laptop)
- ❑ Resume a session, and immediately keep working where you left off
- ❑ More interactive jobs
- ❑ Available apps:
 - Internet browser, terminal, ...
 - Text editor, PDF reader, Image viewer, ...
 - Matlab, RStudio, SAS
- ❑ **Remark:**
 - NX is shared among tens of users
 - Do not compute/test your code there**
 - Instead, submit jobs from NX to compute nodes

NX virtual desktop



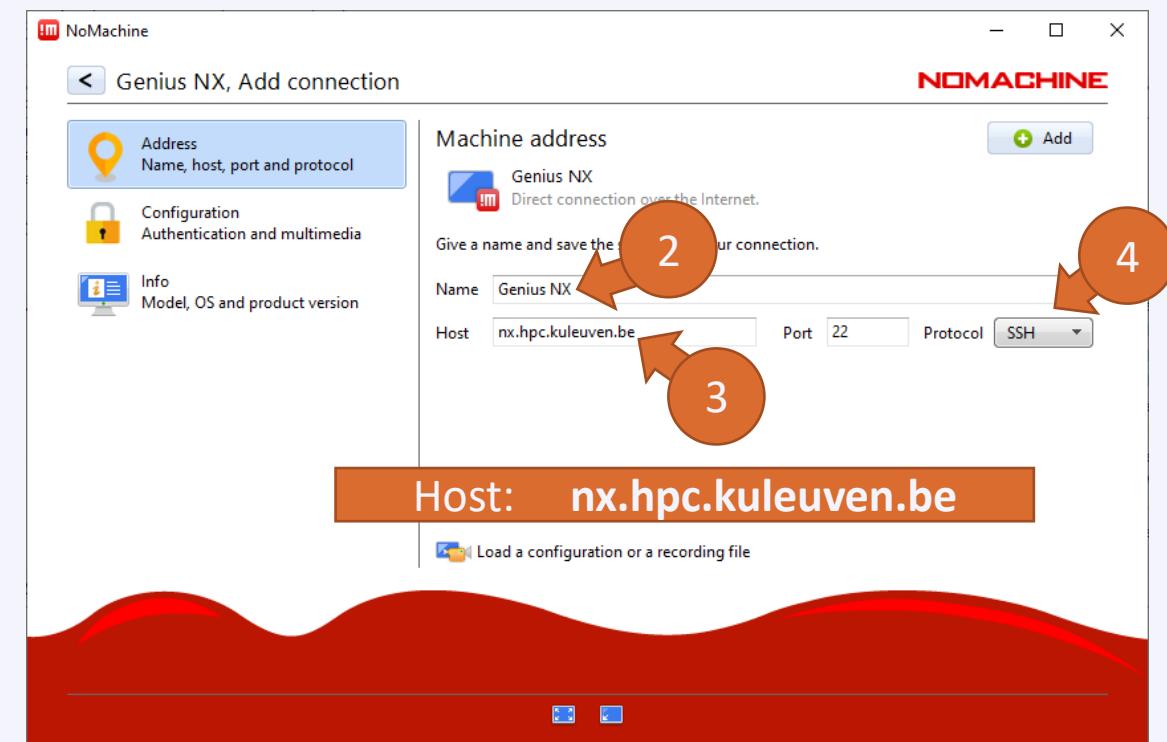
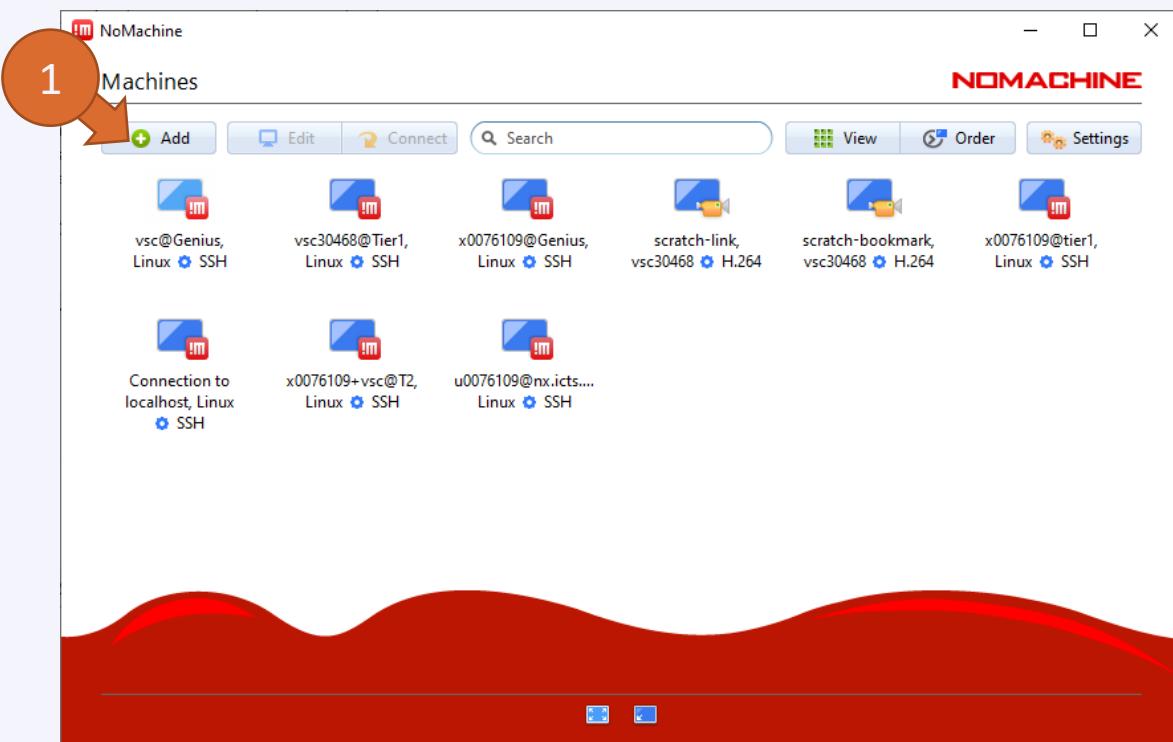


- NX is a graphical login node
- A new session stays active for long
- Rich suite of applications; e.g.
Terminal, Firefox, text editor, Matlab,
RStudio, SAS etc.
- NX is a shared, login-node.
Please do not compute on NX.
- NX server is attached to Genius. Jobs
submitted from NX will appear in
Genius queue

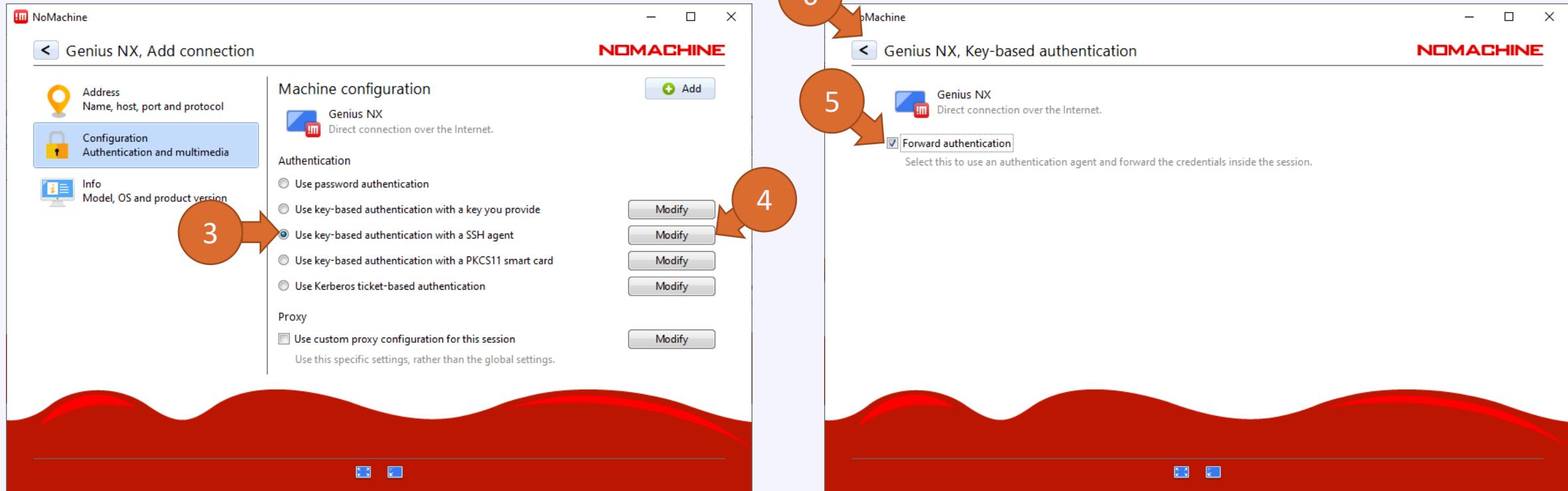
NX: available software

- Accessories:** Gedit, Vi IMproved, Emacs (dummy version), Calculator
- Graphics:** gThumb (picture viewer), Xpdf Viewer
- Internet:** Firefox
- HPC: Computation:** Matlab (2018a), RStudio, SAS
- Visualisation:** Paraview, VisIt, VMD
- Programming:** Meld Diff Viewer (visual diff and merge tool)
- System tools:** File Browser, Terminal
- Additionally:** Gnuplot (graphing utility), Filezilla (file transfer tool), Evince (PDF, PostScript, TIFF, XPS, DVI Viewer)
- Software launched through modules from Terminal.

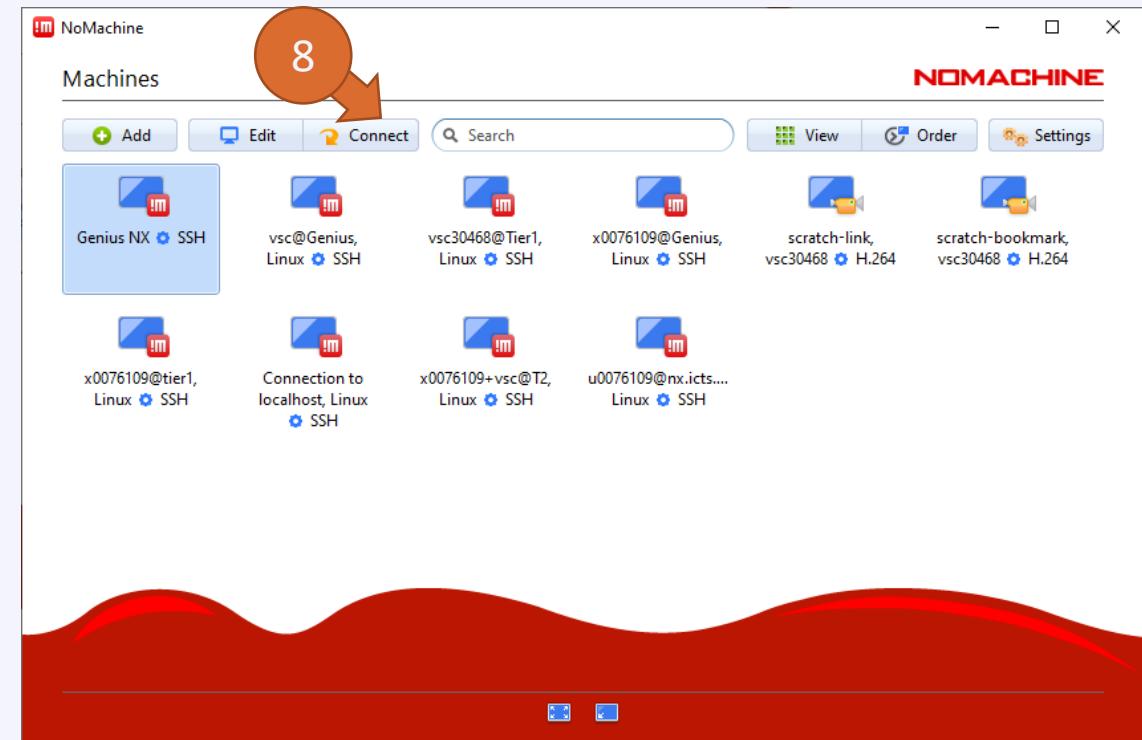
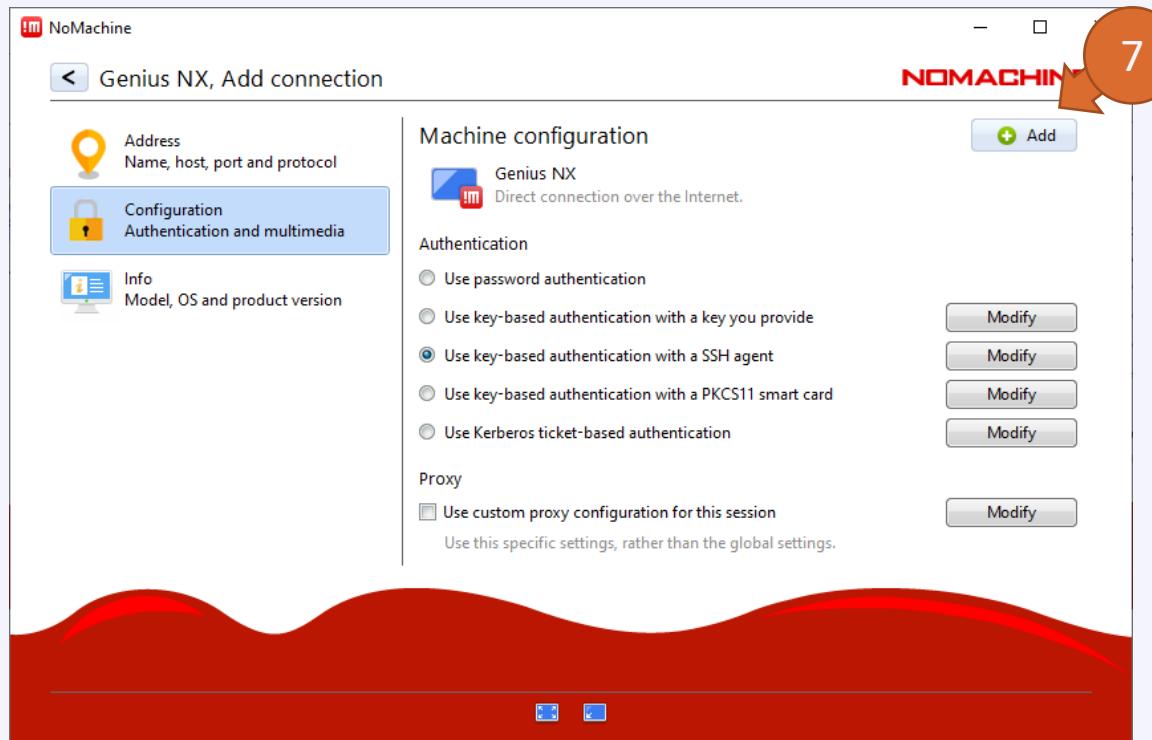
Setup NX



Setup NX

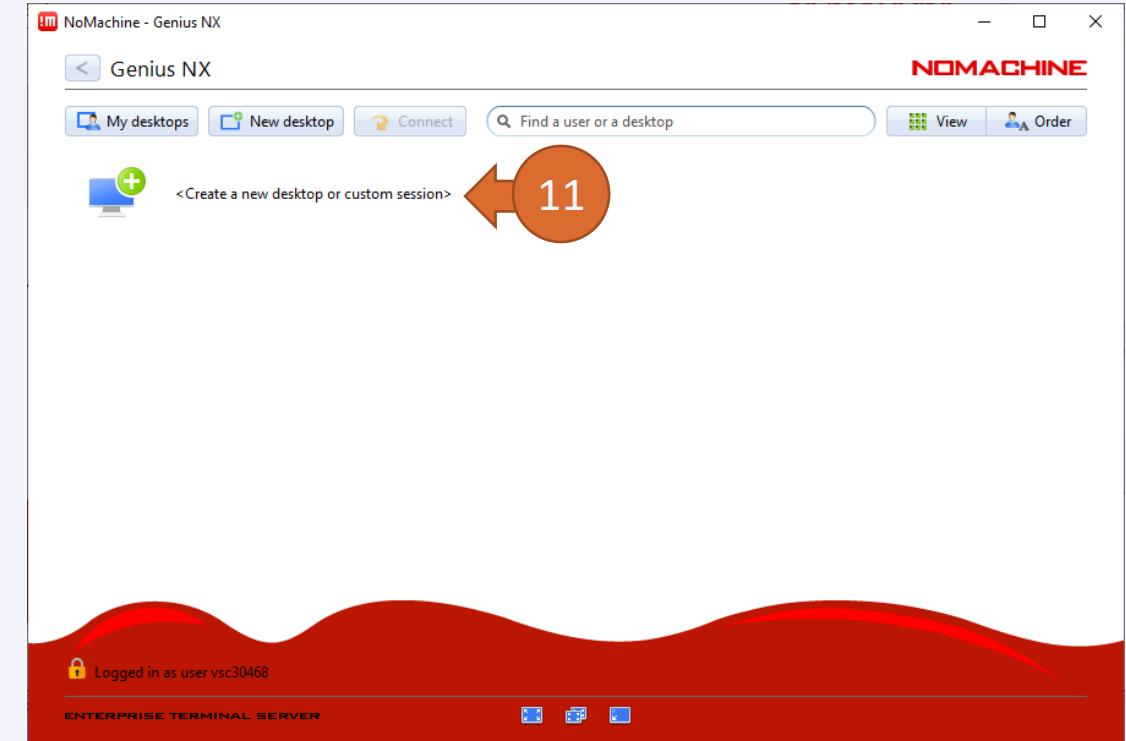
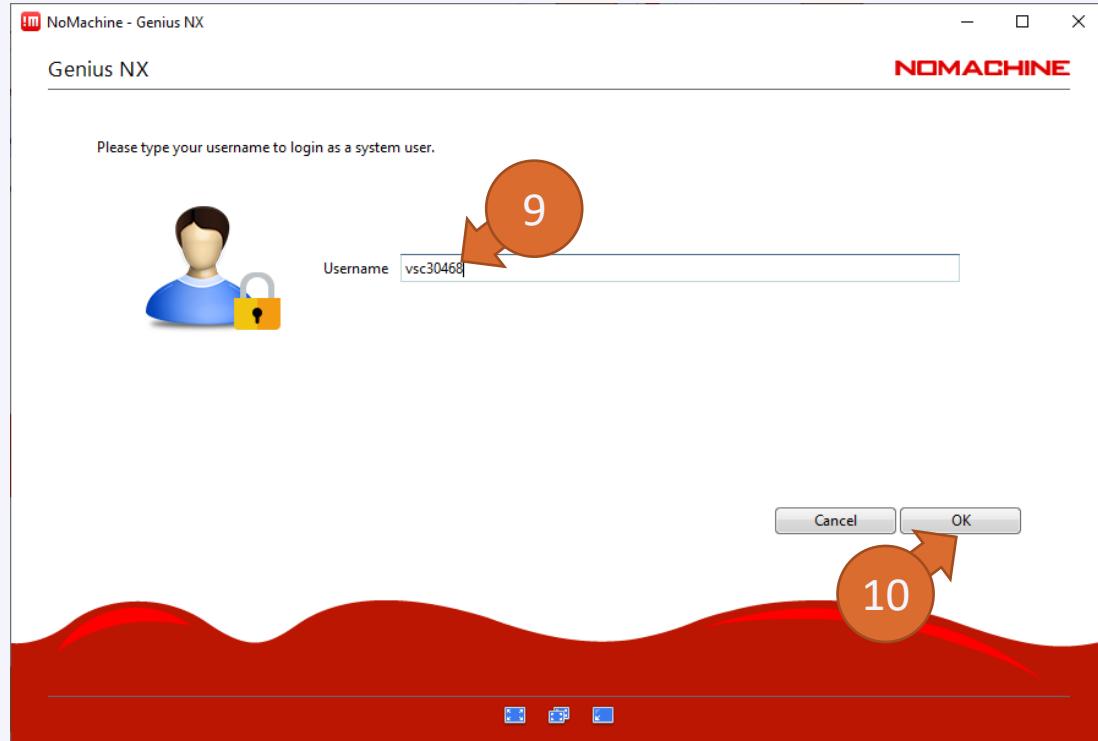


Setup NX

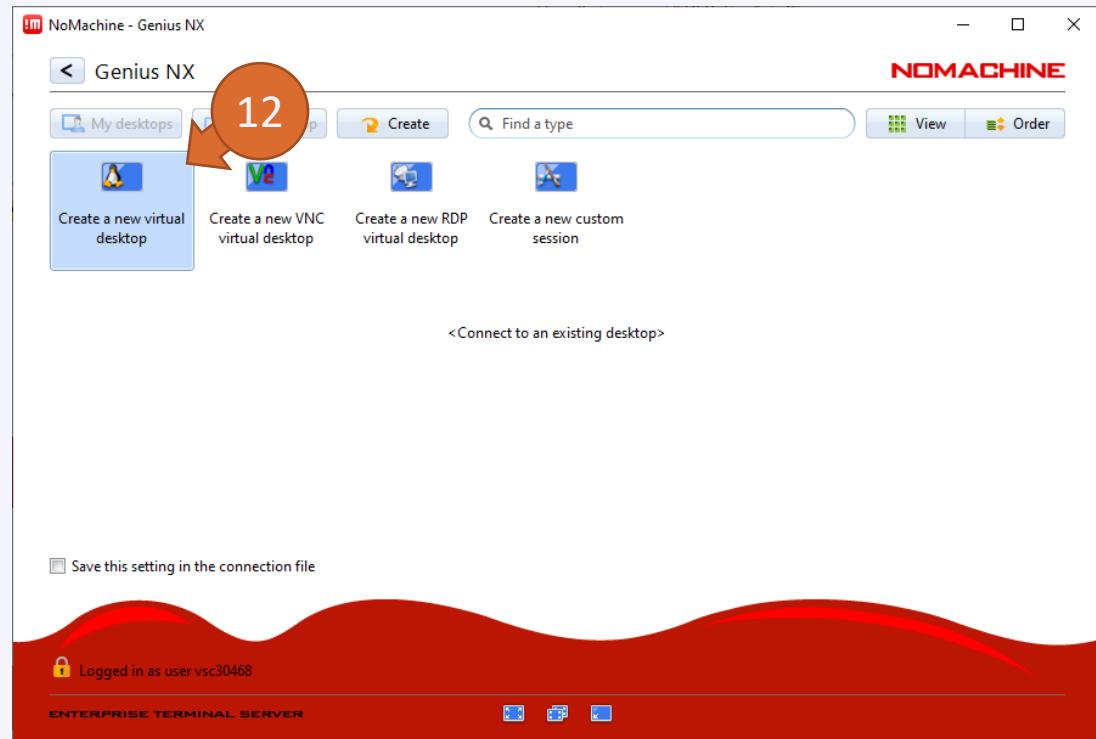


Setup NX

First open putty connection to open firewall or go to firewall.vscentrum.be for further authentication

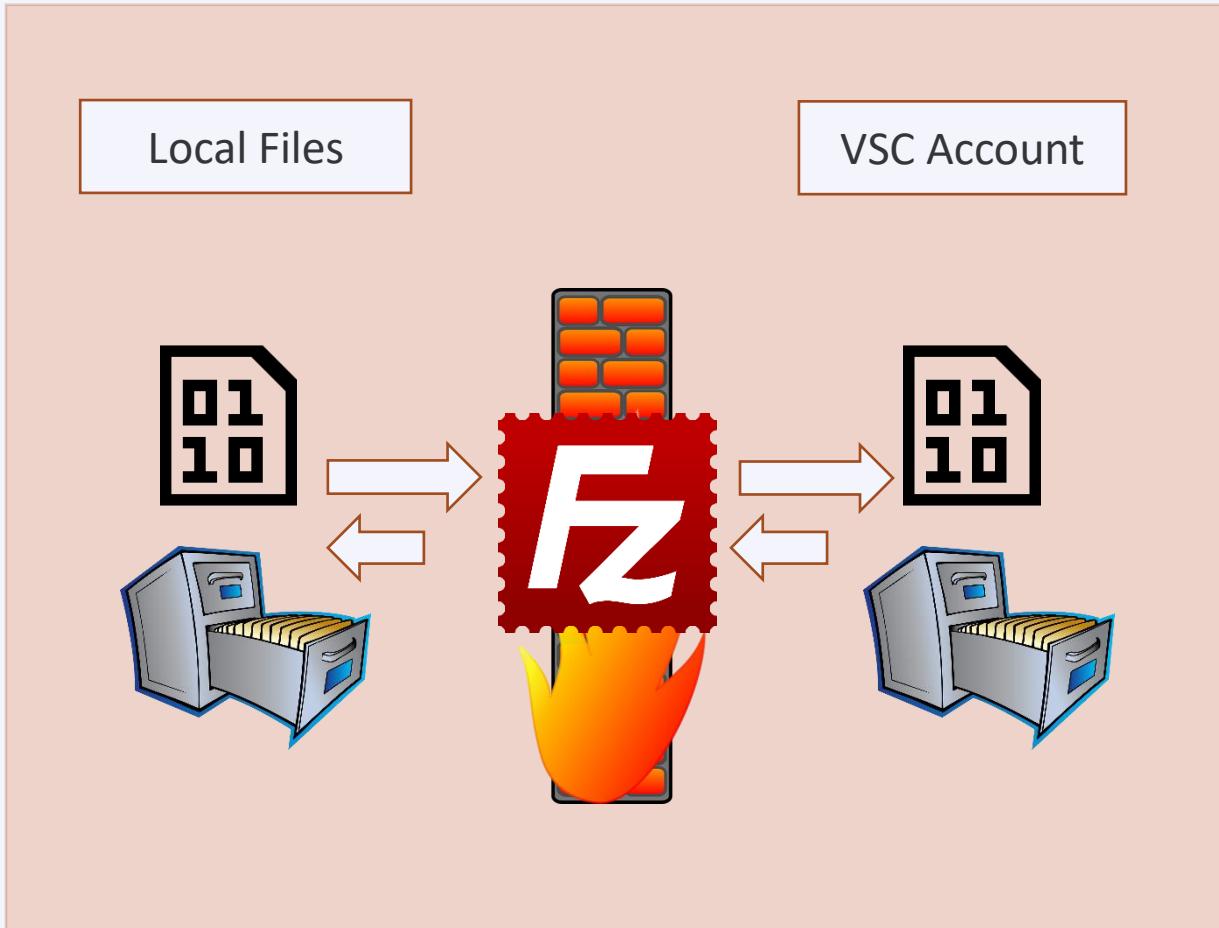


Setup NX in 10 Steps



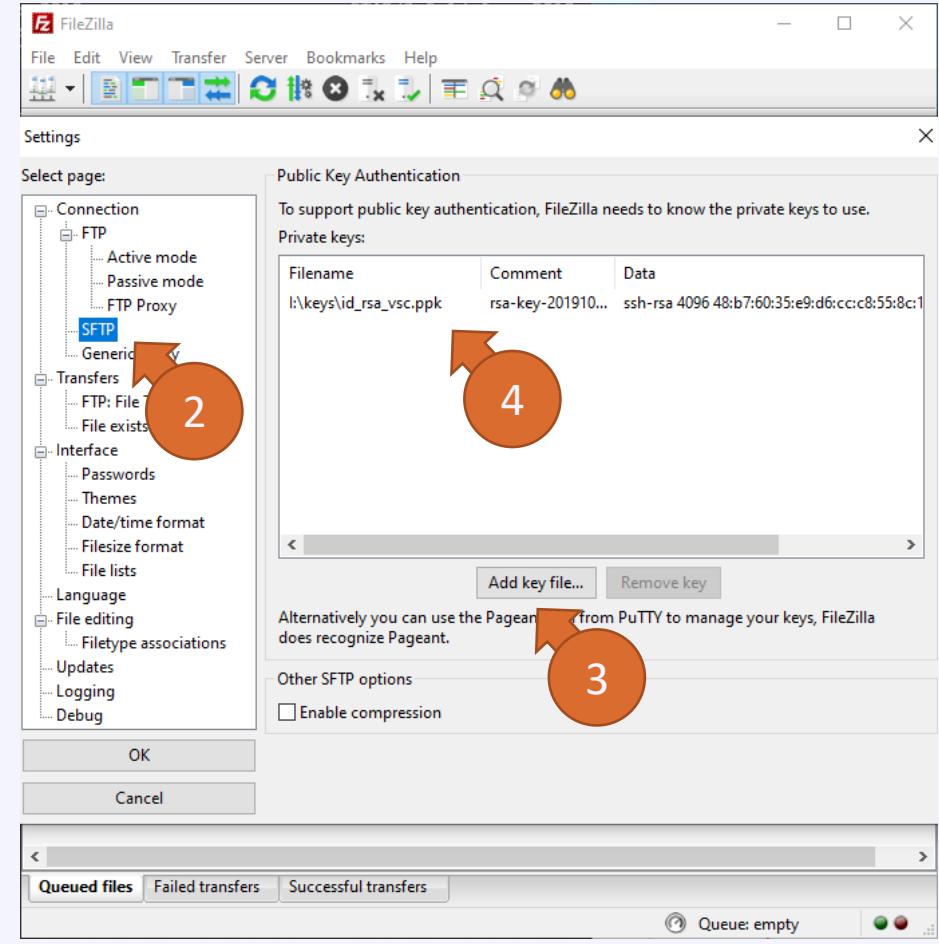
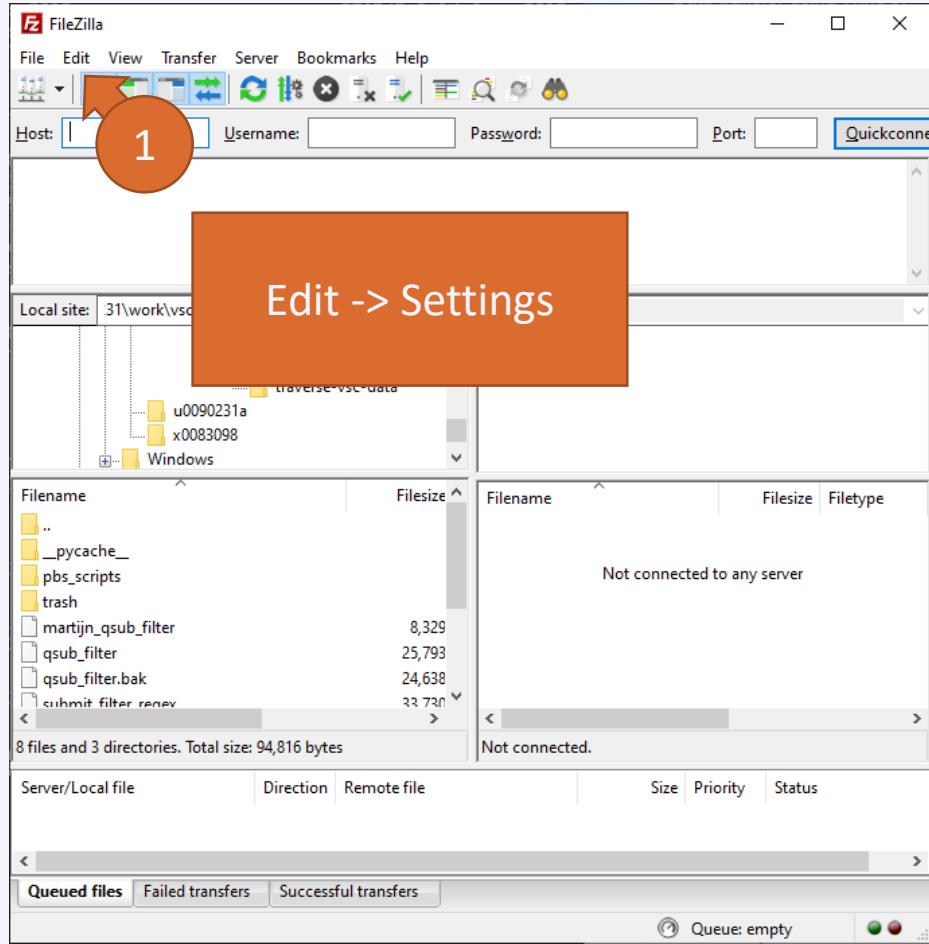
File Transfer with FileZilla

File transfer



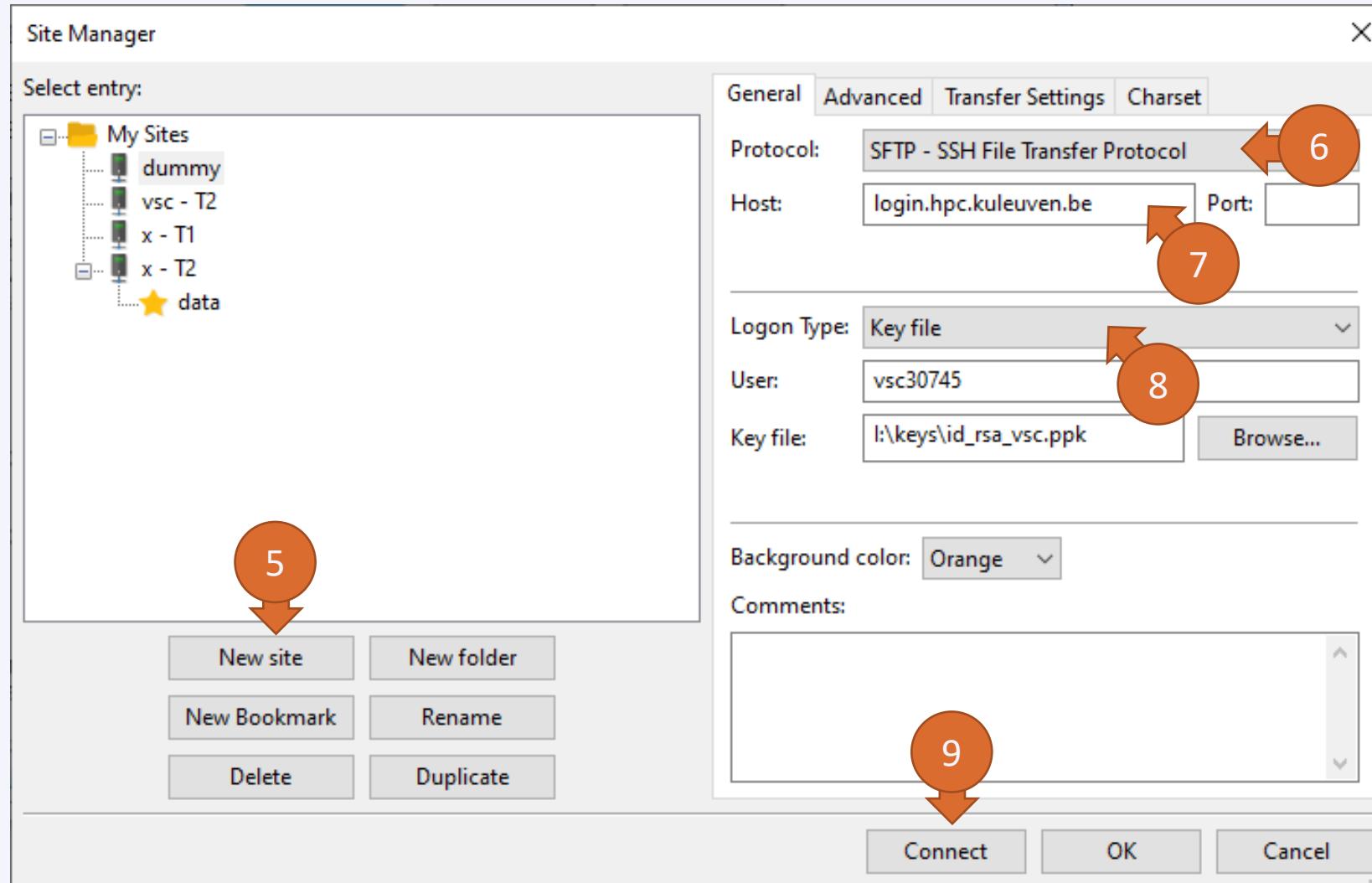
Application	OS
FileZilla	Windows, Linux, Mac
WinSCP	Windows
rsync, scp	Linux, Mac
Globus	Window, Linux, Mac

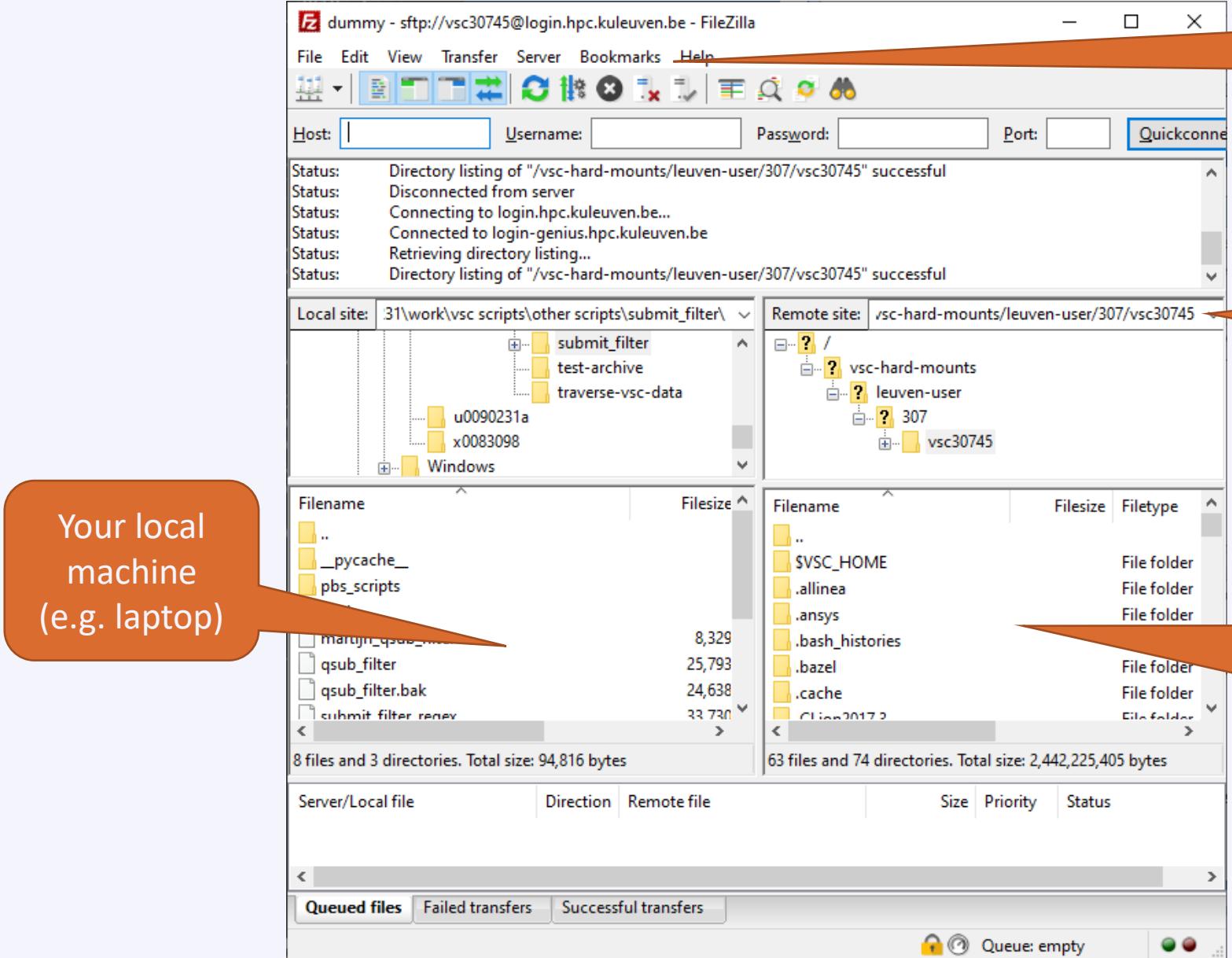
File transfer



File transfer

First open putty connection to open firewall or go to firewall.vscentrum.be for further authentication





Your local
machine
(e.g. laptop)

For convenience, you'd
better bookmark your data
and scratch folders!

Instead, go to your
\$VSC_DATA folder, e.g.
/data/leuven/399/vsc39934

Your VSC storage (e.g.
\$VSC_DATA,
\$VSC_SCRATCH).
Note: by default you
arrive at your own
\$VSC_HOME. Copy
nothing there!



Open OnDemand

- Access wICE via web browser
- <https://ondemand.hpc.kuleuven.be>
- Login via KULeuven MFA
- File browser
- Interactive apps + integrated shell
- Create, start and monitor jobs
- Develop, compile and test your code
- VSCode editor
- Jupyter Lab, RStudio and Tensorboard

KU LEUVEN

OnDemand provides an integrated, single access point for all of your HPC resources.

Pinned Apps A featured subset of all available apps

 Active Jobs System Installed App	 Home Directory System Installed App	 Job Composer System Installed App	 Login Server Shell Access System Installed App
 code-server System Installed App	 Interactive Shell System Installed App	 Jupyter Lab System Installed App	 RStudio Server System Installed App
 Tensorboard System Installed App			



Open OnDemand

- E.g. to start a Jupyter Notebook
- Pick a valid Slurm credit account
- Default partition: batch
interactive: Max 8 cores, 1 GPU, 16 hr
- Default resources:
1 core, 1 hour, 3400MB RAM, no GPU
- (At the moment) you cannot use scratch and staging

Interactive Apps

Servers
● Interactive Shell
Jupyter Lab
● RStudio Server
● Tensorboard
● code-server
Work in progress
● ParaViewWeb - Work in progress
● cryosparc

Jupyter Lab

This app will launch a Jupyter Lab server on one or more nodes.

Account

lpt2_sysadmin

Partition

interactive

batch(_long) or bigmem or interactive or gpu or dedicated...

Number of hours

3

Number of cores

1

Required memory per core in megabytes

3400

Number of nodes

1

Number of gpu's

0

[type:] Specify the total number of GPUs slices for the job. An optional GPU type specification can be supplied. For example "A100:3" or "3".

Reservation (optional)

Name of an existing reservation in which the job should run

I would like to receive an email when the session starts

Software Stack

Software: Available Modules

- OS: Linux CentOS 7.x (Genius) and Rocky Linux 8.x (wlCE)
- Toolchains (Genius):
Intel 2018a (icc, icpc, ifort; Intel MPI; Intel MKL)
FOSS 2018a (gcc, g++, gfortran; OpenMPI; ScaLAPACK, OpenBLAS, FFTW)
- wlCE: default is 2021a
- Note: **Never mix FOSS and Intel compilers (gives dependency conflict)**

Command	Remark
module av	List all installed modules
module av Python	List all Python-related modules
module use /apps/leuven/icelake/2022b/modules/all	Expose other modules from non-default toolchains, by including them in the search
module spider Python	Get more info
module load Python/3.6.4-intel-2018a	Load a specific module
module list	List all loaded modules and their dependencies
module unload Python/3.6.4-intel-2018a	Unload a module (but dependencies still stay)
module purge	Remove all modules from your work session

Software: Your Specific Needs

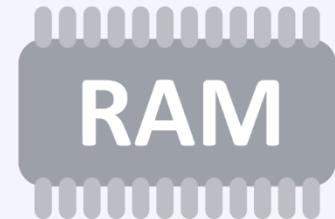
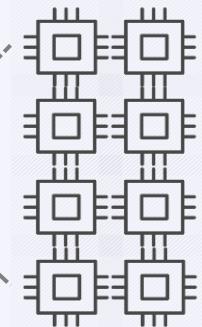
- You can always install your desired software in your \$VSC_DATA
Use Intel or FOSS toolchains
- Compile your code on a compute node (with interactive job)
- If you cannot, ask us for help
- Python/R packages for AI and ML must be installed by the users themselves
- Read more about [Python Package Management](#)
- Read more about [R Package Management](#)

A colorful illustration of a steam train engine. The engine is brown with a large stack of colorful balloons attached to its side. The balloons are in various colors including red, blue, green, and yellow. The background is a light blue sky with some clouds.

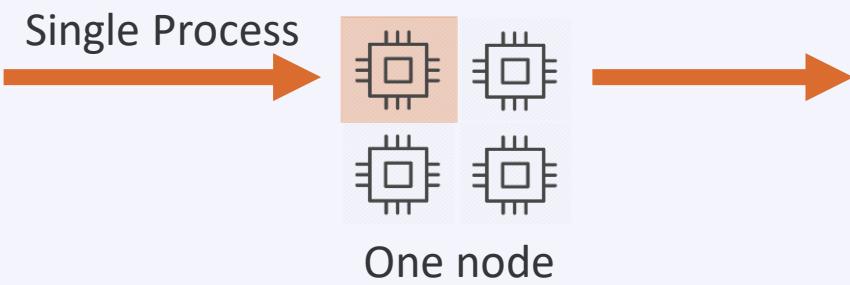
Starting to Compute

Resource Glossary

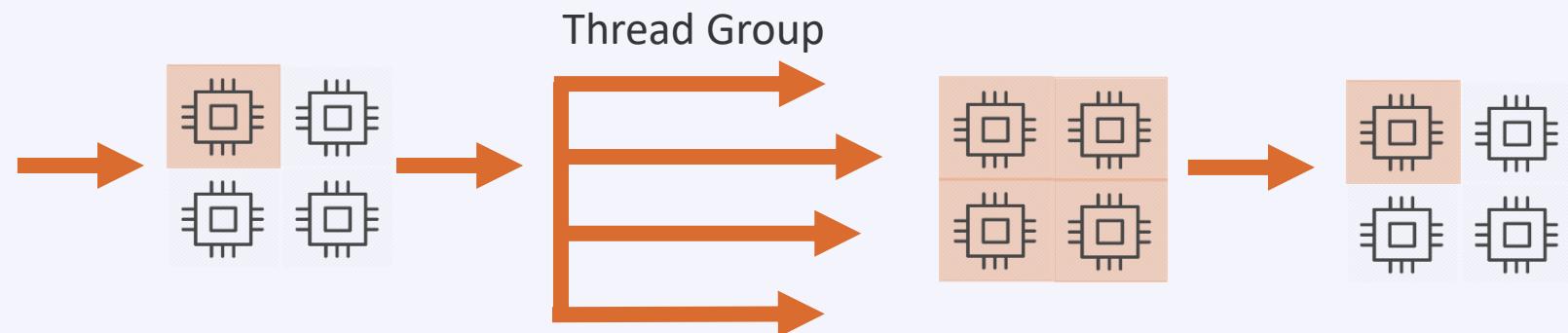
- Nodes:** how many compute servers to request?
- Cores:** how many cores per node to use?
- Memory requirement:** how much memory each core needs?
- Partition:** gpu, bigmem, superdome, amd
- Walltime:** how long to use resources?
- Storage:** how much storage (data, scratch, etc) the job needs?
- Credits:** how many compute credits will be consumed?



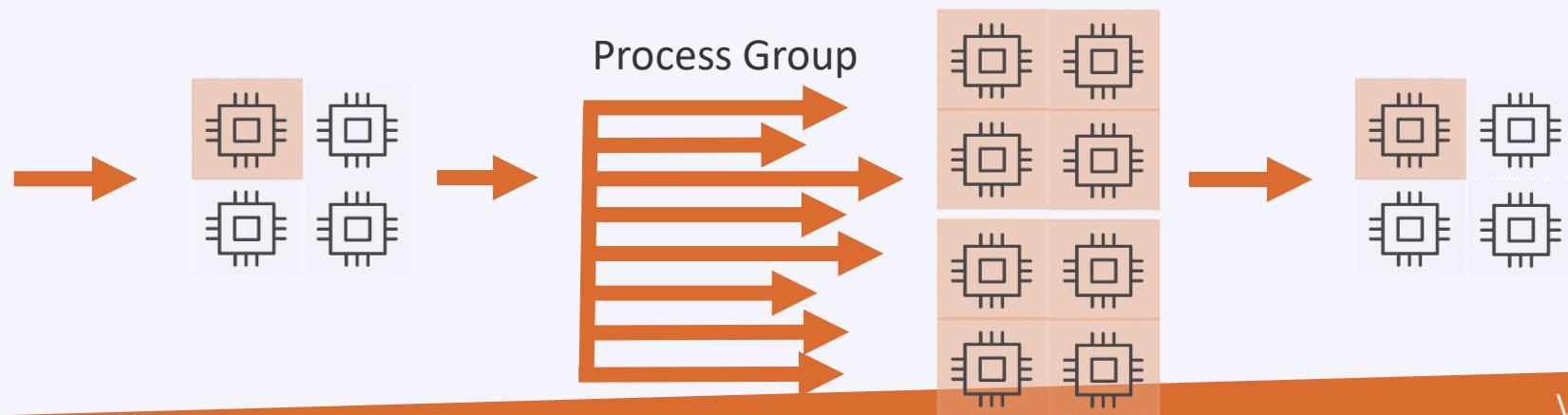
Serial Application (1 process on 1 core)



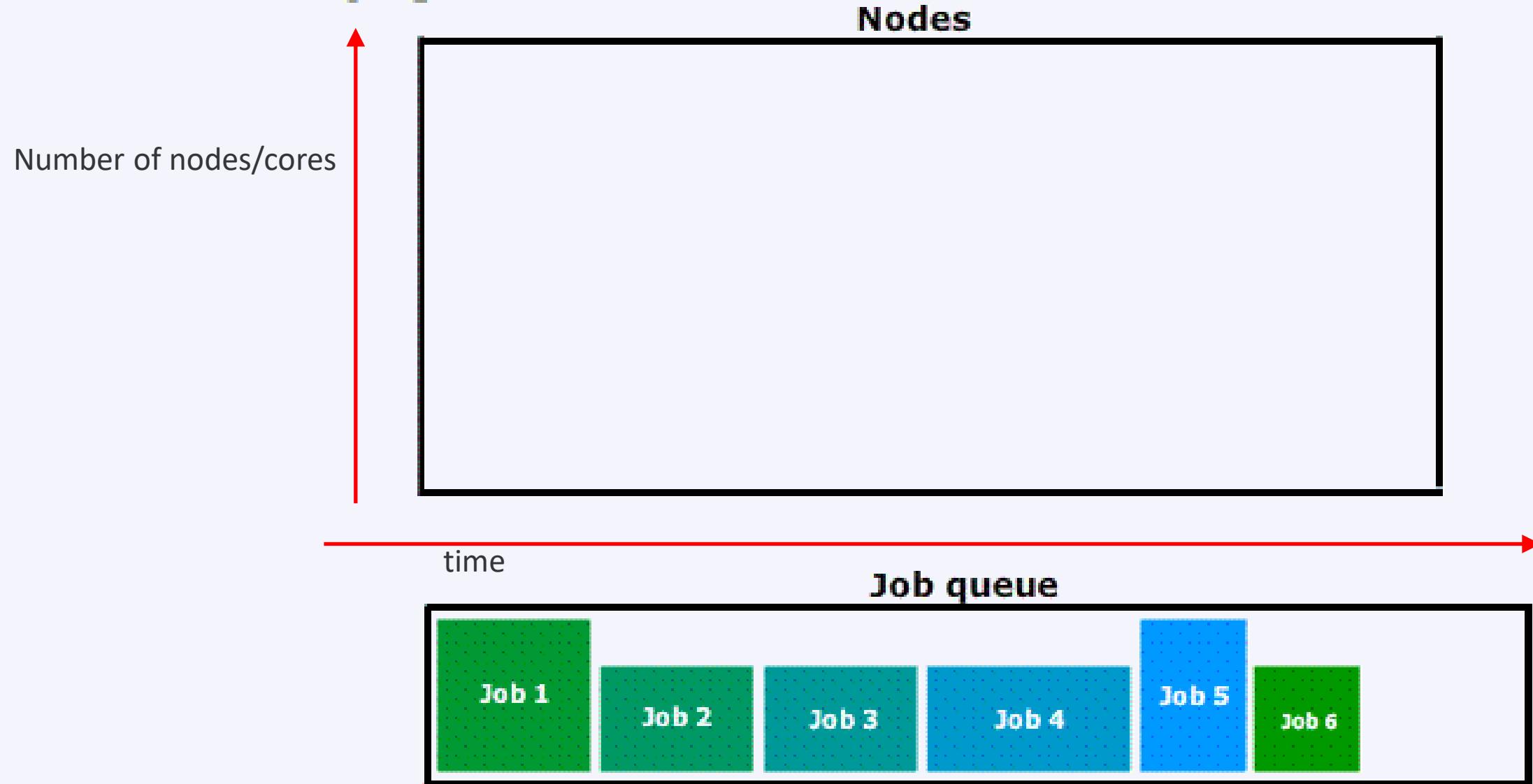
Multi-Core Appl. (N threads on N cores from **1 node**)



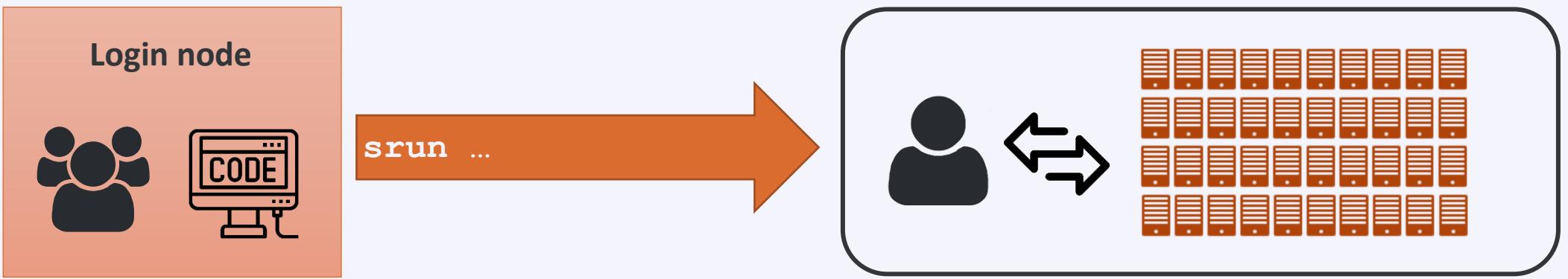
Distributed Appl. (many processes on multiple cores/nodes)



Backfill

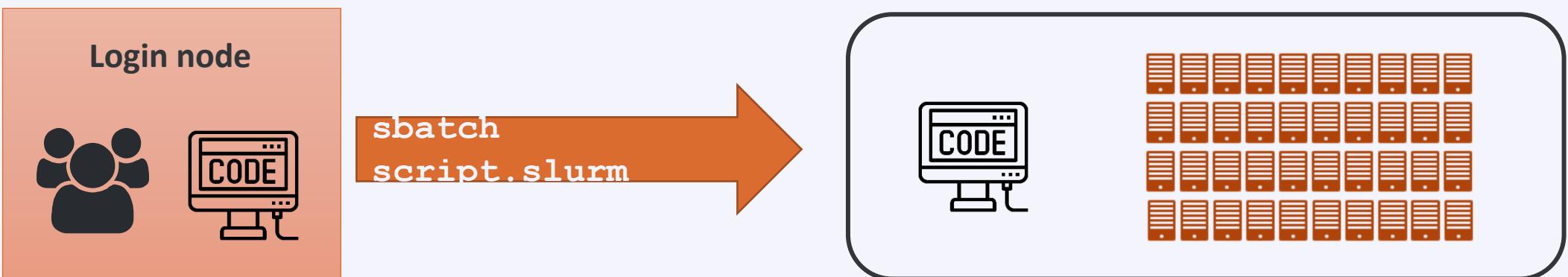


Interactive Job



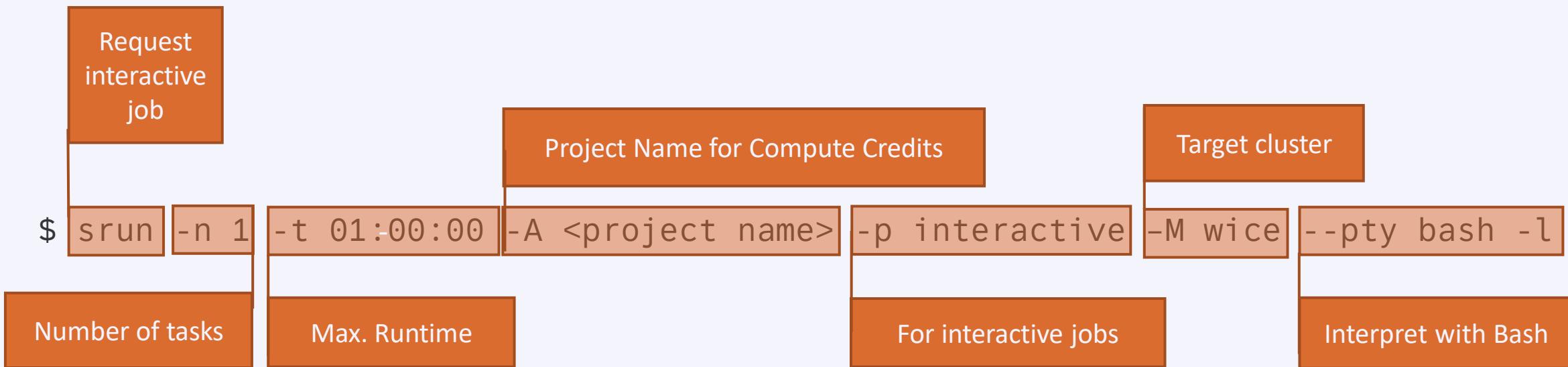
Compute nodes

Batch Job



Compute nodes

Interactive Job on wICE



Remark

- Specifying <project name> for credits is mandatory, e.g.: **-A lp_hpcinfo**
- Implicit defaults for an interactive job on wICE are: **-n 1**, **-t 01:00:00**, **--mem-per-cpu=2000M**

Jobs: Slurm submit options

SLURM	Remarks
--nodes=X --ntasks-per-node=Y --ntasks=X*Y --ntasks=X --cpus-per-task=Y	or or (for Hybrid MPI/OpenMP only)
--partition=gpu	Default: "batch" partition
--mem-per-cpu=XG	Default: average memory/core
--time=dd-hh:mm:ss	
-J jobname	
-o <file_template>	output
-e <file_template>	Default: same as stdout
--mail-type=FAIL,BEGIN,END	
--mail-user=<mailaddress>	
--export=ALL, key=value	Additional variables to pass to job
-A account	Mandatory (credits)

Argument Shorthands

Some (not all) of the
sbatch, srun and
salloc command line
arguments have
shorthands

Shorthand	Full Argument	Meaning
-A	--account	Slurm account name
-a	--array	Job array range
-M	--cluster	Machine or cluster name
-c	--cpus-per-task	Num cores per task (default=1)
-d	--dependency	Job dependency (after, afterok, afterany, ...)
-I	--input	STDIN filename
-e	--error	STDERR filename
-o	--output	STDOUT filename
-G	--gpus	Num GPUs per job
-t	--time	Maximum walltime
-N	--nodes	Minimum num nodes
-n	--ntasks	Maximum num tasks
-p	--partition	Partition name

Jobs: Defaults

If no explicit resources are given your job will start

- ✓ with 1 node 1 core,
- ✓ with walltime of 1hr
- ✓ in batch partition
- ✓ jobs with no cluster given start on Genius cluster (and get job-id starting with number 5)
- ✓ soon the Slurm accounts are available also on Genius (with the same account name)

Interactive Jobs

- Interactive job: 1 core for 1 hour (default)

```
$ srun -M wice -A lp_hpcinfo --pty /bin/bash -l
```

- Interactive job with X-forwarding

```
$ srun -M wice -A lp_hpcinfo -x11 --pty /bin/bash -l
```

- Request fraction of a node from interactive partition

```
$ srun -M wice -A lp_hpcinfo -N 1 -n 4 -p interactive --pty /bin/bash -l
```

- Request a GPU accelerator

```
$ srun -M wice -A lp_hpcinfo -N1 -n 18 -p gpu -G 1 --pty /bin/bash -l
```

Example: Slurm Job Script

```
#!/bin/bash -l  
  
#SBATCH --cluster wice  
#SBATCH --account lp_hpcinfo  
#SBATCH --nodes 1  
#SBATCH --ntasks 4  
#SBATCH --mem-per-cpu 4G  
#SBATCH --job-name hpc_workflow  
#SBATCH --mail-type BEGIN,END,FAIL  
#SBATCH --mail-user my.name@kuleuven.be  
  
module --force purge  
module load intel/2021a  
module load Python/3.9.5-GCCcore-10.3.0  
  
which python3  
  
cd $VSC_SCRATCH/projects/simulations  
cp -r $VSC_DATA/input_data .  
  
python modelling.py  
  
cp -r output_data $VSC_DATA  
rm -rf ./input_data ./output_data
```

Shebang

Resource List

Module load(s)

Move data

Execute commands

Move data

Batch Workload

- Submit the job to the batch server
- Receive a unique JobID
- Error and output files

Job Script

```
#!/bin/bash -l  
...  
...
```

Command Line

```
$ sbatch simulation.slurm
```

JobID

Submitted batch job **60042478** on cluster
wice

stderr, stdout

```
$ ls *.out  
slurm-60042478.out
```

Output File

- STDERR and STDOUT are by default redirected to a single file:
- slurm-<Job ID>.out
- Contains job info, all errors and warnings, and printouts
- Always study it
- Address all warnings and errors (if you can)
- Typical error examples ...

STDOUT + STDERR

```
$ ls slurm-* .out  
slurm-60042478.out
```

Out of Memory

```
slurmstepd: error: Detected 1 oom-kill event(s).  
Some of your processes may have been killed by the  
cgroup out-of-memory handler.
```

Short Walltime

```
slurmstepd: error: *** JOB 60042478 ON s28c11n2  
CANCELLED AT 2023-02-08T10:03:43 DUE TO TIME LIMIT  
***
```

Low Disk Space

```
IOError: [Errno 122] Disk quota exceeded
```

Output File

- Always created
- slurm-<job_id>.out
- Contains all standard output and error (instead of screen)
- Always study it
- Standard Output and Error channels can be redirected to other files:

```
#SBATCH --output ...
#SBATCH --error ...
```

stdout

```
$ ls slurm-* .out
slurm-60041238.out
```

Output File

```
SLURM_JOB_ID: 60033947
SLURM_JOB_USER: vscXXXXXX
SLURM_JOB_ACCOUNT: lp_wice_pilot
SLURM_JOB_NAME: testjob
SLURM_CLUSTER_NAME: wice
SLURM_JOB_PARTITION: batch
SLURM_NNODES: 1
SLURM_NODELIST: m28c30n4
SLURM_JOB_CPUS_PER_NODE: 72
Date: Tue Jan 10 17:02:04 CET 2023
Walltime: 00-01:00:00
=====
/apps/leuven/icelake/2021a/software/intel-compilers/2021.2.0/compiler/2021.2.0/linux/bin/intel64/icc
cp: cannot stat '/apps/leuven/training/test': No such file or directory
Hello World
```

Resource Summary

Stdout

Inspecting Jobs

Example

```
$ scontrol show job -M wice 60049330
JobId=60049330 JobName=f70.slurm
  UserId=vsc3....(253...) GroupId=vsc3....(253...)
Account=lp_metacommunity_models QOS=lp_metacommunity_models
JobState=PENDING Reason=QOSGrpBillingMinutes
  SubmitTime=2023-02-16T00:24:58 EligibleTime=2023-02-16T00:24:58
  Partition=batch NodeList= NumNodes=4-4 NumCPUs=288 NumTasks=288 CPUs/Task=1
  TRES(cpu=288,mem=979200M,node=4,billing=733
  MinCPUsNode=72 MinMemoryCPU=3400M
  WorkDir=/vsc-hard-mounts/leuven-data/3.../vsc3.../Odonate_SOM
  StdErr=/vsc-hard-mounts/leuven-data/3.../vsc3.../Odonate_SOM/slurm-60049330.out
  StdIn=/dev/null
  StdOut=/vsc-hard-mounts/leuven-data/3.../vsc3.../Odonate_SOM/slurm-60049330.out
```

Diagnosis

Why is my job in pending/hold state?

Check out the “Reason” on Slurm docs: https://slurm.schedmd.com/resource_limits.html

Other Partitions on wICE

GPU

```
#SBATCH --partition=gpu
#SBATCH --nodes=1
#SBATCH --ntasks=18
#SBATCH --gpus-per-node=1
```

Big Memory

```
#SBATCH --partition=bigmem
#SBATCH --nodes=1
#SBATCH --tasks-per-node=72
#SBATCH --mem-per-cpu=28000M
```

Managing & Monitoring Jobs

Command	Purpose
<code>sbatch ...</code>	Submit a batch job
<code>srun ...</code>	Submit an interactive job
<code>scancel --cluster=wice <JobID></code>	Cancel a specific pending or running job
<code>scontrol show job --cluster=wice <JobID></code> <code>slurm_jobinfo <JobID></code>	Detailed job info (very useful to diagnose issues)
<code>squeue --cluster=wice -long</code>	Status of all recent jobs
<code>squeue --cluster=wice -start</code>	Give a <i>rough</i> estimate of start time
<code>sinfo --cluster=wice</code>	Info about the state of available partitions and nodes
<code>sacct --cluster=wice --batch --job <JobID></code>	Show minimal info about a queue or partition
<code>slurmtop --cluster=wice</code>	Overview of the cluster

Interactive Partition

- Accessible via command line and Open OnDemand
- To quickly compile, test, debug your (parallel) application
- To pre-/post-process your data and make visualizations
- Short queue time
- 5 dedicated nodes on wICE
- 64 cores per node, 1 GPU (=7 GPU instances), 512 GB memory
- Max resource per user: 8 cores, 1 GPU instance, 16 hour walltime

Interactive Job

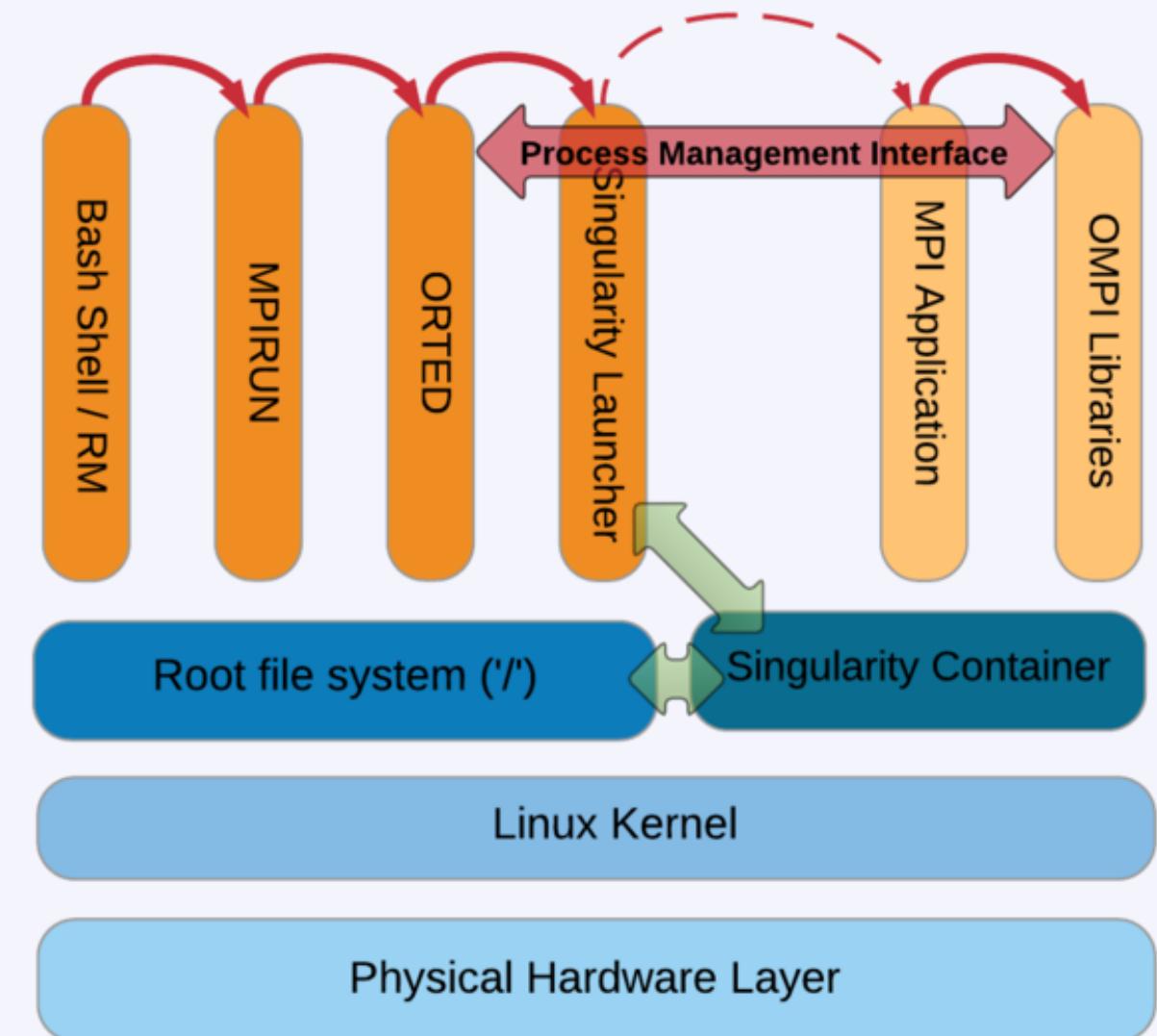
```
$ srun -n 1 -p interactive -t 16:00:00 -G 1 --pty bash -l
```

Singularity Containers

- What?
 - + Self-contained OS & software & data
- Why?
 - + fully resolved dependency chains
 - + portable workflow
- How?
 - + You create the image
 - + Run it on Genius/wICE
 - + MPI/OpenMP is supported

```
#!/bin/bash -l
#SBATCH -t 30:00
#SBATCH -N 2
#SBATCH -n 72
#SBATCH -M wice
#SBATCH -A lp_hpcinfo
singularity run Project.simg ./model.exe
```

Slurm Script



Credits

Credits card concept:

- **Pre-authorization**: holding the balance as unavailable until the merchant clears the transaction
- Balance to be held as unavailable: based on requested resourced (walltime, nodes)
- **Actual charge** based on what was really used: used walltime (you pay only what you use, e.g. when job crashes)
- See output file

How to check the rates?
Check service catalogue

How to check available credits?

```
$ sam-balance
```

Did the new credits arrive?

```
$ sam-list-allocations
```

Credit Pricing

- You pay as you go!
- For academic projects:
1000k credits = 3.5 EUR
- Credits needed for a job:

#credits (k) = Walltime (hr) × #nodes × Factor

- For shared nodes, you pay a fraction of the costs,
based on the ppn specified

Cluster / Partition	Credits/hr
Genius Cascadelake	11.3
Genius Cascadelake 8 GPUs	40
Genius Skylake 4 GPUs	20
Genius Skylake BigMem	12
Genius Skylake	10
Genius / Superdome	10
Genius / AMD nodes	10
wIICE / Thinnode	10,99
wIICE Bigmem	19
wIICE / 1 GPU	11,25
wIICE 4GPUs	45
wIICE Interactive	-

How to Manage Credits?

Command	Purpose
sam-balance	List active projects and available credits
sam-list-allocations	List the validity dates of different projects/allocations
sam-list-usagerecords	List current charge rates for different nodes
sam-statement -A <account-name>	Job statements from an account



Conda Package Management: Python and R

Conda on wice

https://docs.vscentrum.be/en/latest/software/python_package_management.html?highlight=iniconda#install-python-packages-using-conda

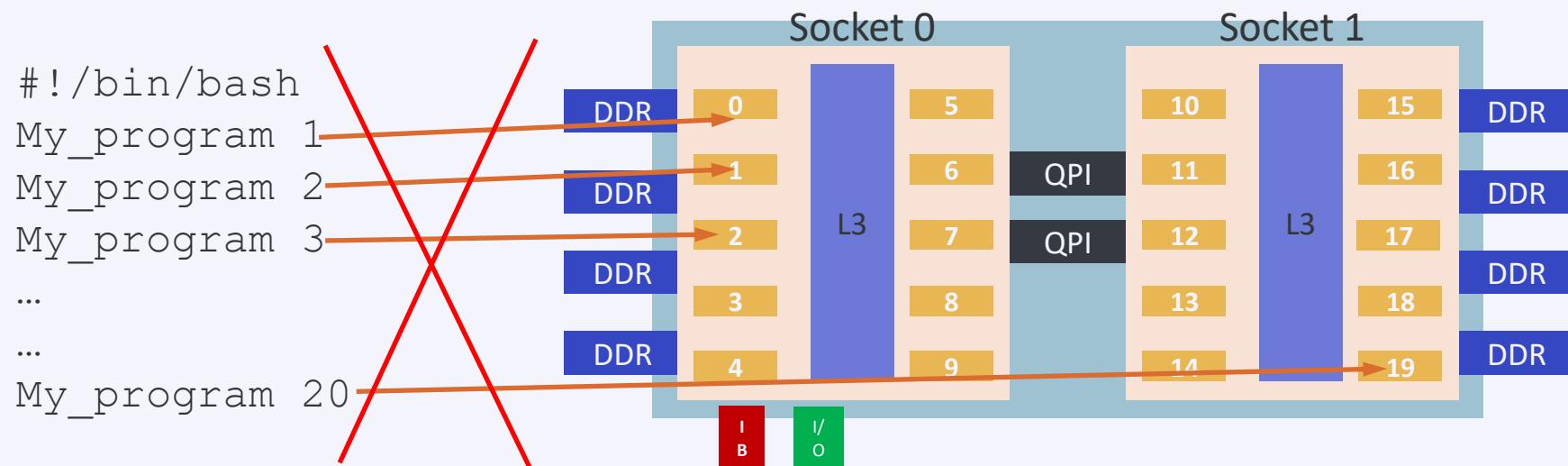
- ✓ Good practice would be 2 separate paths of conda installations for each cluster.
- ✓ Beginning of the slurm script matters. Use `#!/bin/bash -l` not `#!/bin/bash` to load the env properly.
- ✓ Universal .bashrc

```
case ${VSC_INSTITUTE_CLUSTER} in
genius)
    export PATH="${VSC_DATA}/miniconda3/bin:${PATH}"
    ;;
wice)
    export PATH="${VSC_DATA}/miniconda3-wice/bin:${PATH}"
    ;;
esac
```

Worker Framework

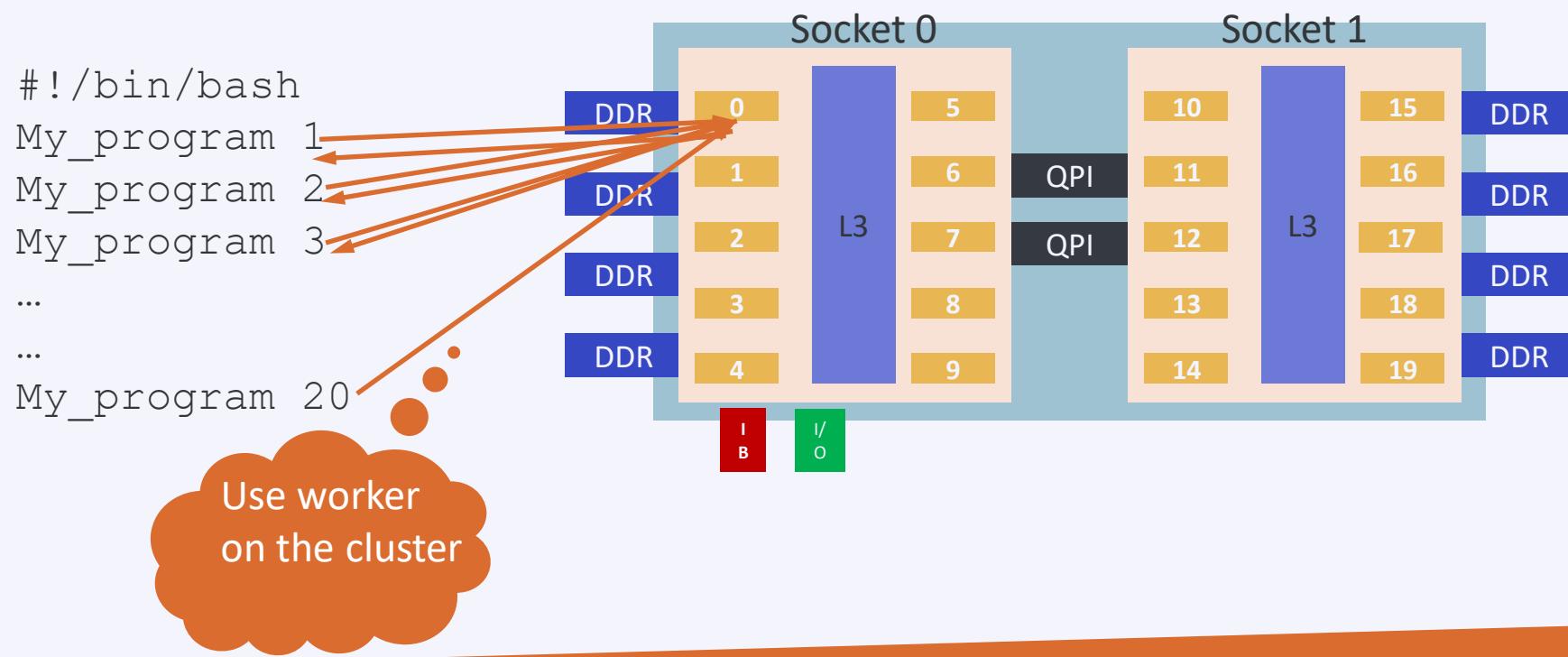
Sequential...

- A shell program consists of a sequential list of commands
- -> bash script on HPC cluster will run sequentially



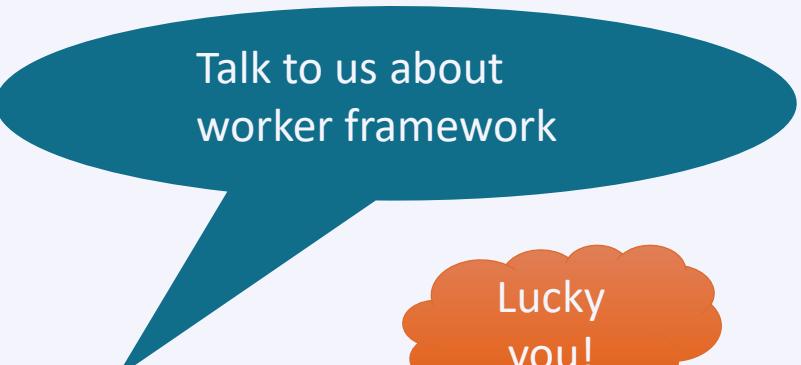
Sequential...

- A shell program consists of a sequential list of commands
 - -> bash script on HPC cluster will run sequentially



Parallel Computing

- Serial:
 - one program, on one core
- 'Embarrassingly parallel' problems:
 - lots of runs of one program, with different parameters
- Problems that require 'real' parallel algorithms
 - OpenMP
 - MPI : Message Passing Interface



Talk to us about
worker framework



Lucky
you!

Use case: parameter exploration

temperature	pressure	humidity
293.0	1.0e05	87
...
313.0	1.3e05	75

```
#!/bin/bash -l
#SBATCH --ntasks=1
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=1
#SBATCH --time=10:00
#SBATCH -A lp_myproject
cd $SLURM_SUBMIT_DIR
weather -p 1.0e05 -t 293.0 -h 87
```

Many single core computations

The diagram illustrates the execution of multiple parallel tasks. It shows two separate boxes, each containing a command-line script. The first box is labeled 'job_030.slurm' and the second is labeled 'job_600.slurm'. Both boxes contain identical code, except for the final command which specifies different parameters: 'weather -p 1.0e05 -t 293.0 -h 87' for job_030 and 'weather -p 1.3e05 -t 313.0 -h 75' for job_600. A large red diagonal banner across the middle of the boxes contains the text 'Many single core computations'.

Solution: worker with -data

temperature	pressure	humidity	data.csv
293.0	1.0e05	87	
...	

```
#!/bin/bash -l
#SBATCH --cluster=wice
#SBATCH --nodes=2
#SBATCH --ntasks-per-node=72
#SBATCH --time=01:00:00
#SBATCH -A lp_hpcinfo

cd ${SLURM_SUBMIT_DIR}
weather -p $pressure -t $temperature -h $humidity
```

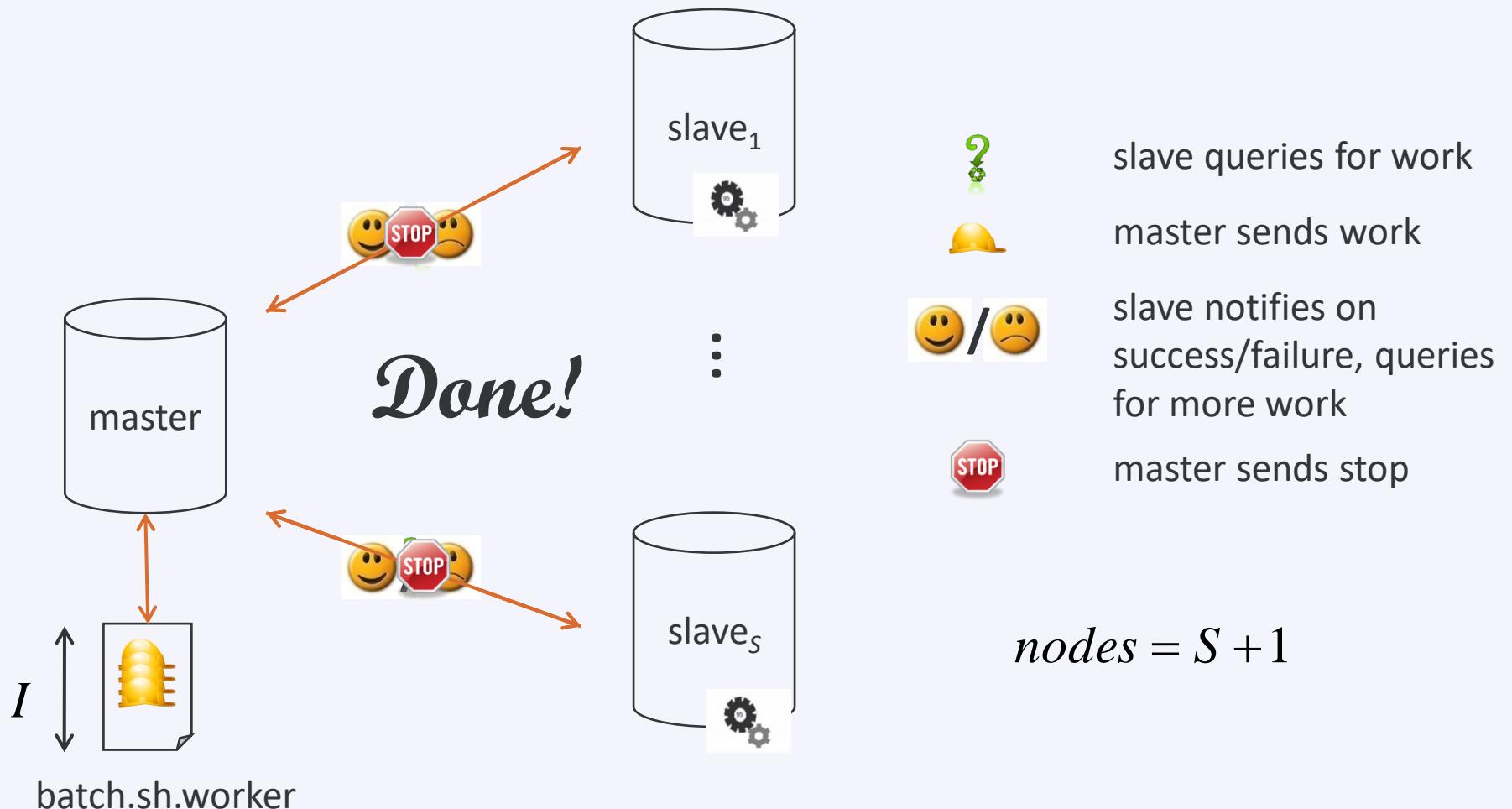
```
#From wICE interactive session:
$ module --force purge
$ module use /apps/leuven/icelake/2021a/modules/all
$ module load worker/1.6.12-foss-2021a-wice
$ wsub -data data.csv -batch job.slurm
```

Data exploration: steps

- Write slurm script with parameters
- Create Excel sheet with data
 - Convert to CSV format
- Submit with `wsub`
 - walltime is time to complete all work items

$$\text{walltime}_{\text{job}} \geq \frac{N \cdot \text{walltime}_{\text{work item}}}{\text{nodes} \cdot \text{ppn}}$$

worker processing: informally



How to use worker well?

- Many work items, i.e., `#work items/#proc >> 1`
- `time(work item) > 1 minute`
- Work item is not multithreaded
- Work item is multithreaded
 - will work, but user must be careful to request the right resources
 - Use `-threaded <n>` flag with `wsub`
- There be dragons: licensing!

worker & multithreading

- Some software uses multithreading automatically, e.g.,
 - R
 - Matlab
 - Will use as many threads as there are cores, regardless of system load
 - 36 cores/node
 - 36 work items/node
- } **72×72 threads/node**

Oversubscription: ***very inefficient!!!***

Controlling number of threads

- R, most of the time: OMP_NUM_THREADS=1

- Matlab
 - Use maxNumCompThreads(1) function call
 - Use compiler flag: mcc -singleCompThread ...

```
#!/bin/bash -l           program_pe.slurm
#SBATCH -account=lp_myproject
#SBATCH --cluster=wice
#SBATCH --time=1:00:00
#SBATCH --nodes=2
#SBATCH --ntasks-per-node=72

module load R
cd $SLURM_SUBMIT_DIR

export OMP_NUM_THREADS=1
Rscript program $a $b
```

Help on worker

- See documentation
<http://worker.readthedocs.io/>
 - Each command has help, use -h
-
- ✓ All resources have to be specified inside slurm script
 - ✓ wresume does not work yet. Only wsub is so far supported
 - ✓ -master flag does not work yet

Submitting worker jobs

From wICE (Interactive) Node

```
$ module av worker  
----- /apps/leuven/icelake/2021a/modules/all -----  
worker/1.6.12-foss-2021a
```

From Genius Login Node

```
$ module use /apps/leuven/skylake/2021a/modules/all  
$ module av worker  
----- /apps/leuven/skylake/2021a/modules/all -----  
  
worker/1.6.12-foss-2021a-wice      worker/1.6.12-intel-2021a  (D)  
worker/1.6.12-foss-2021a
```

Submitting worker jobs

- Let us use one of the worker examples:

https://github.com/gjbex/worker/tree/development_slurm/examples/bash_example

- alp.slurm

```
#!/bin/bash -l
#SBATCH --nodes=1
#SBATCH --ntasks-per-node 27
#SBATCH --time=00:30:00
#SBATCH -A lpt2_sysadmin
#SBATCH --cluster=wice
cd $SLURM_SUBMIT_DIR
alphabet.sh $letter $counter
```

alpha.csv

letter	counter
a	1
b	2
c	3
d	4
e	5
f	6
...	

Submitting worker jobs

From Genius

```
#!/bin/bash -l
module purge
module use /apps/leuven/skylake/2021a/modules/all
module load worker/1.6.12-foss-2021a-wice
wsub -batch alp.slurm -data alpha.csv
```

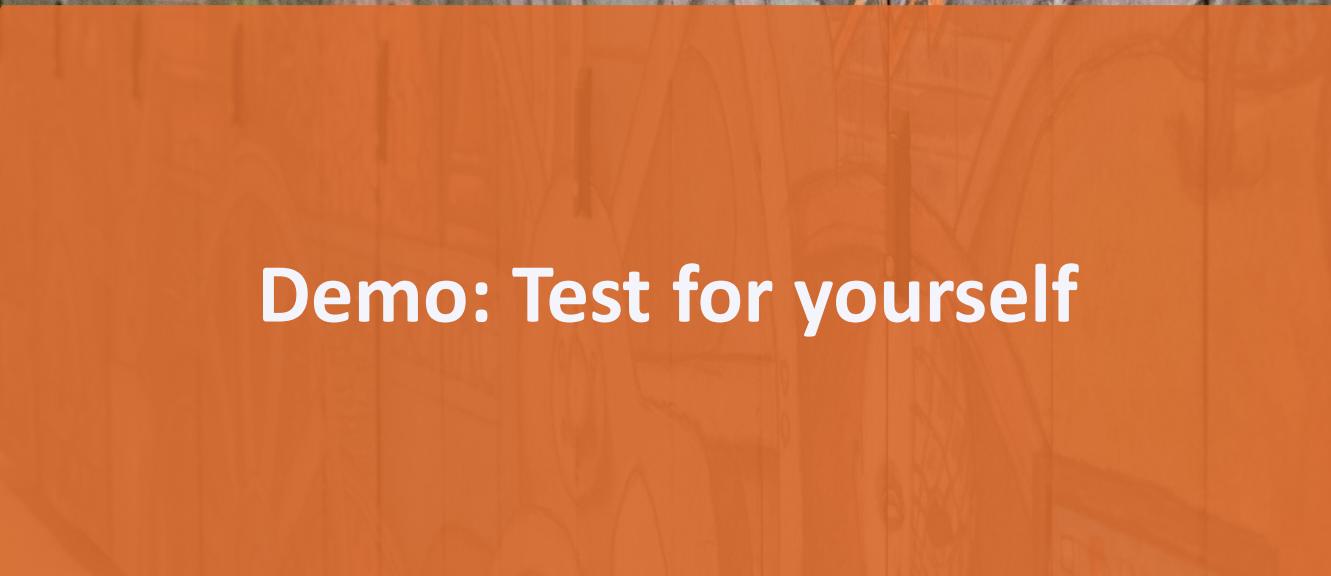
```
total number of work items: 26
Submitted batch job 60008899 on cluster wice
[Sat Jun 12 07:00:00 2021]
```

Submitting worker jobs

- ✓ From interactive job on wICE

```
#!/bin/bash -l
module --force purge
module use /apps/leuven/icelake/2021a/modules/all
module load worker/1.6.12-foss-2021a
wsub -batch alp.slurm -data alpha.csv
```

```
total number of work items: 26
Submitted batch job 60008889 on cluster wice
```



Demo: Test for yourself

demo/test yourself

- ✓ Request membership to lp_hpcinfo group (account.vscentrum.be)
- ✓ Login with putty
- ✓ Filetransfer with Filezilla
- ✓ Login with NX
- ✓ Check disk quota
- ✓ Check the credits
- ✓ Check/load/list/unload/purge module

demo/test yourself

- ✓ Copy intro training files /apps/leuven/training/HPC_intro/ to your \$VSC_DATA
- ✓ Submit cpujob to the cluster
- ✓ List all your jobs squeue -M wice
- ✓ Check the information about the cpujob slurm_jobinfo -M wice <job_ID>
- ✓ Modify the mpi.slurm script to request 1 node, 72 cores for 30 minutes and get the notification about job start/end by e-mail
- ✓ Check the status of all the jobs

demo - monitoring

- ✓ Submit an interactive job

Run your program on a compute node

Open a new terminal and ssh to a compute node

Check the resources usage (`top`)

- ✓ Submit a batch GPU job

While the job is running get the information about the node `slurm_jobinfo...`

and check usage of resources on the node `ssh, top, nvidia-smi`

demo – conda installation

✓ Install miniconda in your \$VSC_DATA directory

- \$ wget https://repo.continuum.io/miniconda/Miniconda3-latest-Linux-x86_64.sh
- \$ bash Miniconda3-latest-Linux-x86_64.sh -b -p \$VSC_DATA/miniconda3

✓ Add a PATH to conda:

- \$ export PATH="\$VSC_DATA/miniconda3/bin:\$PATH"

✓ Check if conda is added to your \$PATH (\$ which conda)

Add it to \$PATH in your .bashrc

- \$ echo 'export PATH="\$VSC_DATA/miniconda3/bin:\$PATH"' >> .bashrc

demo – conda usage

- ✓ Create a conda environment including Jupyter

```
$ conda create -n science jupyter numpy scipy
```

- ✓ Activate this environment

```
$ source activate science
```

- ✓ Add matplotlib package to this environment

```
$ conda install matplotlib
```

- ✓ Return to original environment

```
$ conda deactivate
```

demo – worker

- ✓ Copy intro training files (`/apps/leuven/training/worker/`) to your `$VSC_DATA`
- ✓ Go to `exercise1` directory
- ✓ Submit worker job
- ✓ Check the output file

Command	Purpose
\$ sbatch ...	Submit a batch job
\$ srun ...	Submit an interactive job
\$ scancel --cluster=wice <JobID>	Cancel a specific pending or running job
\$ scontrol show job --cluster=wice <JobID>	Detailed job info (very useful to diagnose issues)
\$ slurm_jobinfo <JobID>	
\$ squeue --cluster=wice --long	Status of all recent jobs
\$ squeue --cluster=wice --start	Give a rough estimate of start time
\$ sinfo --cluster=wice	Info about the state of available partitions and nodes
\$ sacct --cluster=wice --batch --job <JobID>	Show minimal info about a queue or partition (-p)
\$ slurmtop	Overview of the cluster
\$ scontrol --cluster=wice show node <hostname>	Get detailed information about the status of a node
\$ sam-balance	Overview of all your available credit projects
\$ sam-list-allocations	Detailed overview of your credit allocation history

Questions

Helpdesk:

hpcinfo@kuleuven.be or https://admin.kuleuven.be/icts/HPCinfo_form/HPC-info-formulier

VSC web site:

<http://www.vscentrum.be/>

VSC documentation: <https://docs.vscentrum.be/en/latest/>

VSC agenda: training sessions, events

Systems status page:

<http://status.vscentrum.be>

*Stay Connected
to vsc*

