



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



- ✓ VSC
- ✓ Tier-2 Clusters
- ✓ Storage
- ✓ Login Nodes & MFA
- ✓ Connection Setup
  - PuTTY
  - Terminal
- ✓ NX
  - About
  - Setup
- ✓ File Transfer with Filezilla
- ✓ Using Open OnDemand
- ✓ Starting to Compute
- ✓ Conda Package Management
  - Python
  - R
- ✓ Worker Framework
- ✓ Demo: Test for yourself



# VSC

(Vlaams Supercomputer Centrum)

# VSC PARTNERSHIP



Supported by



# VSC HPC Environments



[www.vscentrum.be](http://www.vscentrum.be)

Very rich suite of  
courses every  
academic year

Documentation!  
Answers >80% of your questions  
and access to account page

The screenshot shows the homepage of the Vlaams Supercomputer Center (VSC) website. At the top, there is a navigation bar with links for Home, About VSC, Systems & Services, Showcase, News & Events, VSC Training, User Portal, and Access. Below the navigation bar, there is a large banner for "Users Day 2023". The banner features the text "Vlaams Supercomputer Center Presents" at the top left, followed by "USERS DAY" in large, stylized letters, and "2023" below it. To the right of the banner, there is a date "24 OCTOBER 2023" and a large "SAVE THE DATE" message. The banner has a dark blue background on the left and an orange background on the right. At the bottom left of the banner, there is a button labeled "Learn More". The overall design is modern and professional.



## 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

<https://docs.vscentrum.be>

Search your keywords here;  
e.g. SSH key

VLAAMS  
SUPERCOMPUTER  
CENTRUM documentation

About Contact Accounts Data **Compute** Cloud FAQs

Search in VSC Docs. Ctrl + K

Section Navigation

- Infrastructure
- Jobs
- Software

Read The Docs v: latest

Compute

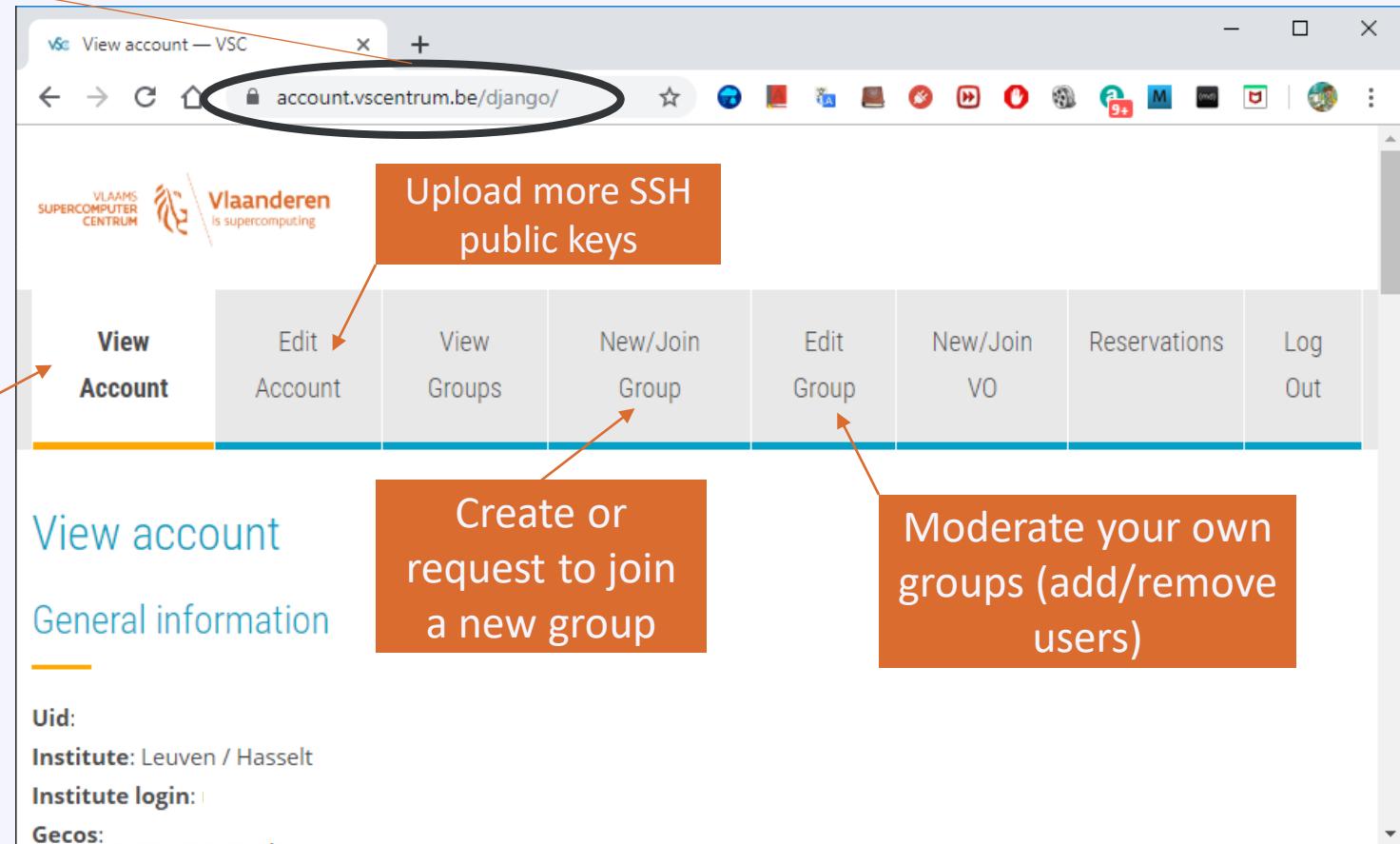
Edit on GitHub

Compute

## Infrastructure

- 1. Tier-1 Infrastructure
  - 1.1. Hortense @ HPC-UGent
- 2. Tier-2 Infrastructure
  - 2.1. UAntwerp Tier-2 Infrastructure
  - 2.2. VUB Tier-2 Infrastructure
  - 2.3. HPC-UGent Tier-2 Infrastructure
  - 2.4. KU Leuven/UHasselt Tier-2 Infrastructure
- 3. Storage Infrastructure
- 4. Archive of Old Clusters
  - 4.1. Tier-1 Breniac
  - 4.2. Thinking Cluster

To manage your  
VSC account:  
[account.vscentrum.be](http://account.vscentrum.be)



# Support and Services

## Basic support

- Helpdesk ([hpcinfo@kuleuven.be](mailto: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

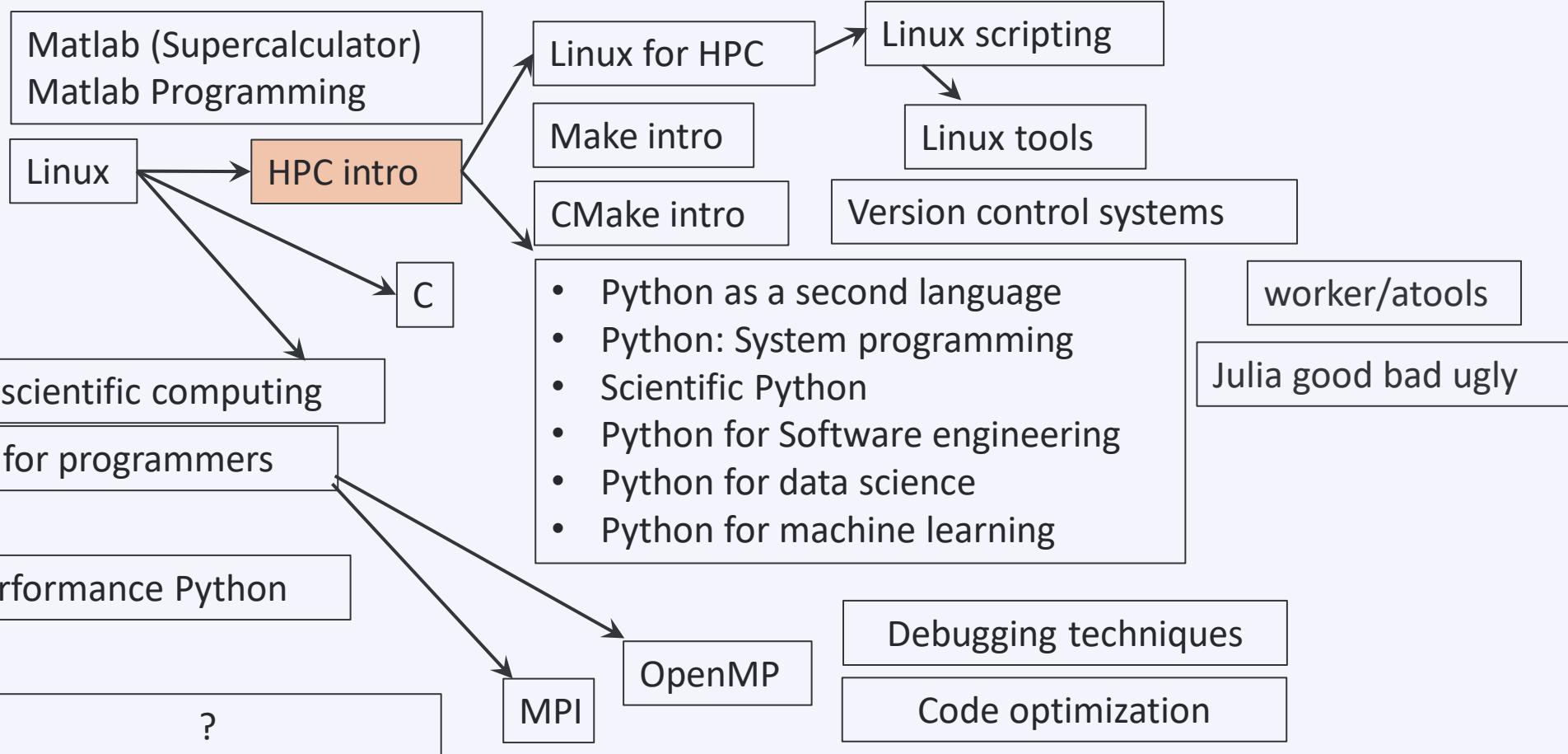
- Create a secure (4096 bit) [SSH key pairs](#)  
Upload it on the account page: [www.account.vscentrum.be](http://www.account.vscentrum.be)
- You need to [request a VSC account](#)  
Normally processed swiftly
- Request [introductory credits](#) (2M free credits for 6 months)
- Request [project credits](#) (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/djang...'. 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, the main content area is titled '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'. A navigation menu has options for 'Home', 'ICTS SERVICECATALOGUS', 'WAT IS DE ICTS SERVICECATALOGUS?', 'OPGELET', and 'ZOEK IN DE SERVICECATALOGUS'. The 'OPGELET' section contains text about netto prijzen and overhead. On the right side, there is an image of a computer mouse.

# VSC training

- Introductory



- Intermediate

- Advanced

- Specialized track

Infosessions:

- Containers
- Notebooks

worker/atools

Julia good bad ugly

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

Debugging techniques

Code optimization

Stay up-to-date <https://www.vscentrum.be/en/education-and-trainings>

# 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

# Tier-2 Clusters @ KU Leuven

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

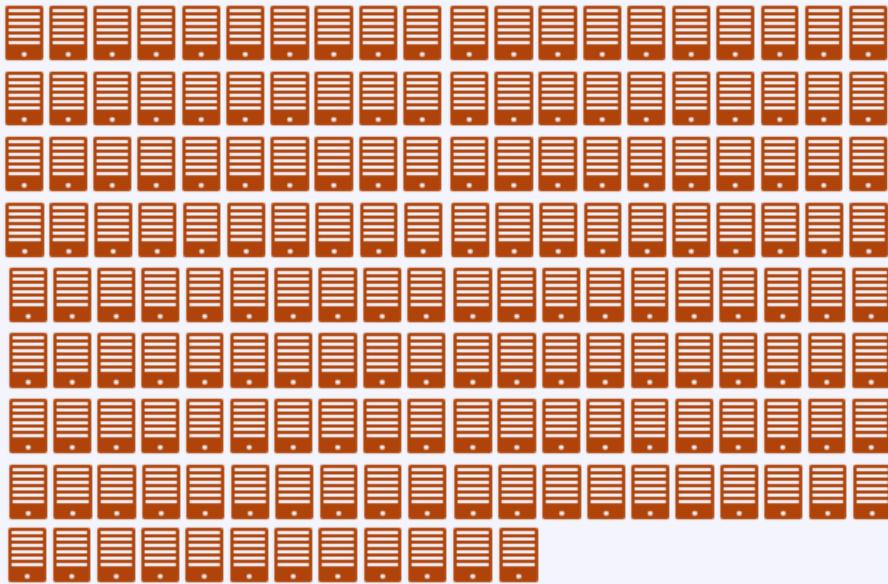


**wICE** (since 9/2022)  
186 nodes; 13,392 cores



# Tier-2 Cluster - wICE

Compute nodes



172x IceLake 72c 256 GB

Large memory nodes



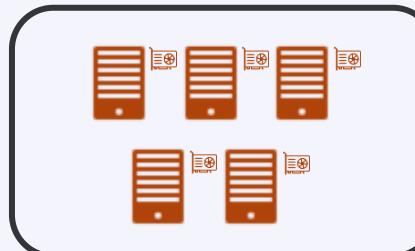
5x IceLake 72c 2048 GB

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 <b>4xA100 SXM2</b> 80GB	IB HDR-100	72	512 GB	960 GB	4
Icelake Interactive	Xeon 8358 <b>1xA100 SXM2</b> 80GB	IB HDR-100	64	512 GB	960 GB	5

# Storage

# Overview of the storage infrastructure

- ✓ Only you own your files (POSIX)  
Users can share folders via [VSC groups](#)
- ✓ 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

Example

## 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.

Immediately go to your other storages, e.g.

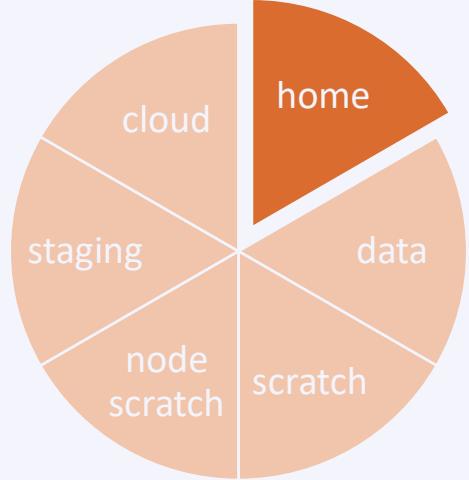
```
$ cd $VSC_DATA
```

## Always check your storage balance using myquota command

```
$ 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
```

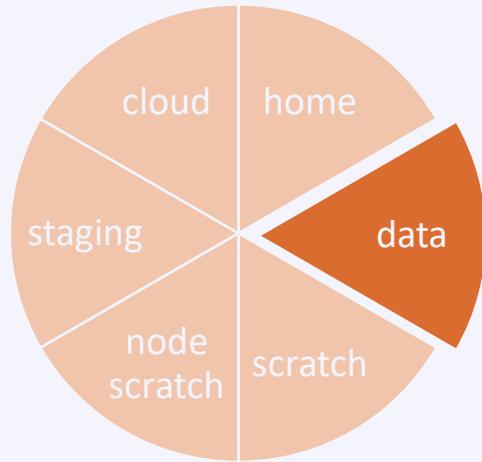
- ✓ [Request form for extra storage](#)
- ✓ [More information](#)

# Storage



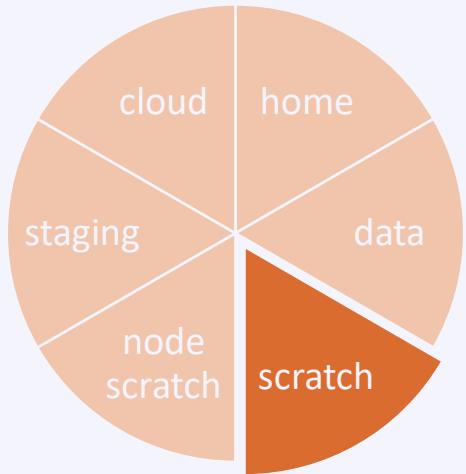
Storage	home folder
Env. Variable	\$VSC_HOME
Filesystem Type	NFS
Access	Global
Backup	Hourly, daily, weekly (until last month) inside the .snapshot folder.
Default Quota	3 GB
Extension	Not possible
Usage	Only storing SSH keys, config files
Remarks	<ul style="list-style-type: none"><li>- Stay away from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

# Storage



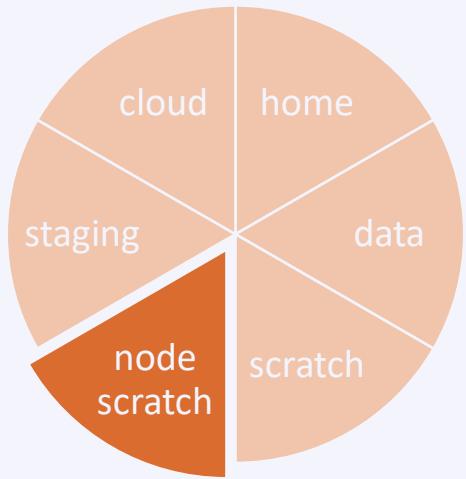
Storage	data folder
Env. Variable	\$VSC_DATA
Filesystem Type	NFS
Access	Global
Backup	Hourly, daily, weekly (until last month) inside the .snapshot folder.
Default Quota	75 GB
Extension	On purchase
Usage	Your data, code, software, results
Remarks	<ul style="list-style-type: none"><li>- Permanent storage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

# Storage



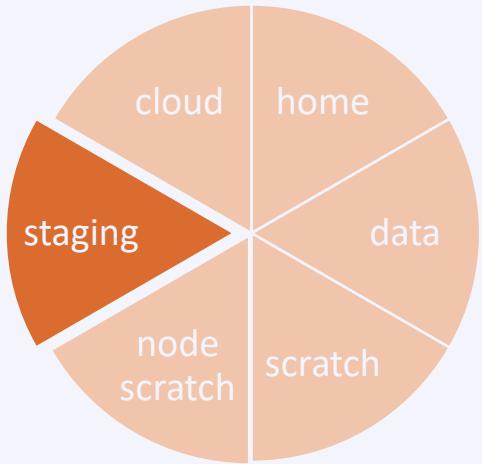
Storage	scratch folder
Env. Variable	\$VSC_SCRATCH
Filesystem Type	Lustre
Access	Global
Backup	delete after 30 days from last access
Default Quota	500 GB
Extension	Free
Usage	Intensive, parallel I/O, temporary files
Remarks	<ul style="list-style-type: none"><li>- Recommended storage for all jobs</li>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

# 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	<a href="#">Read about beeOND</a>
Usage	Temporary storage at runtime
Remarks	<ul style="list-style-type: none"><li>- Fastest I/O, attached to the node</li><li>- Is cleaned after job terminates</li><li>- Copy the data to your home, scratch, or staging before job ends</li></ul>

# Storage

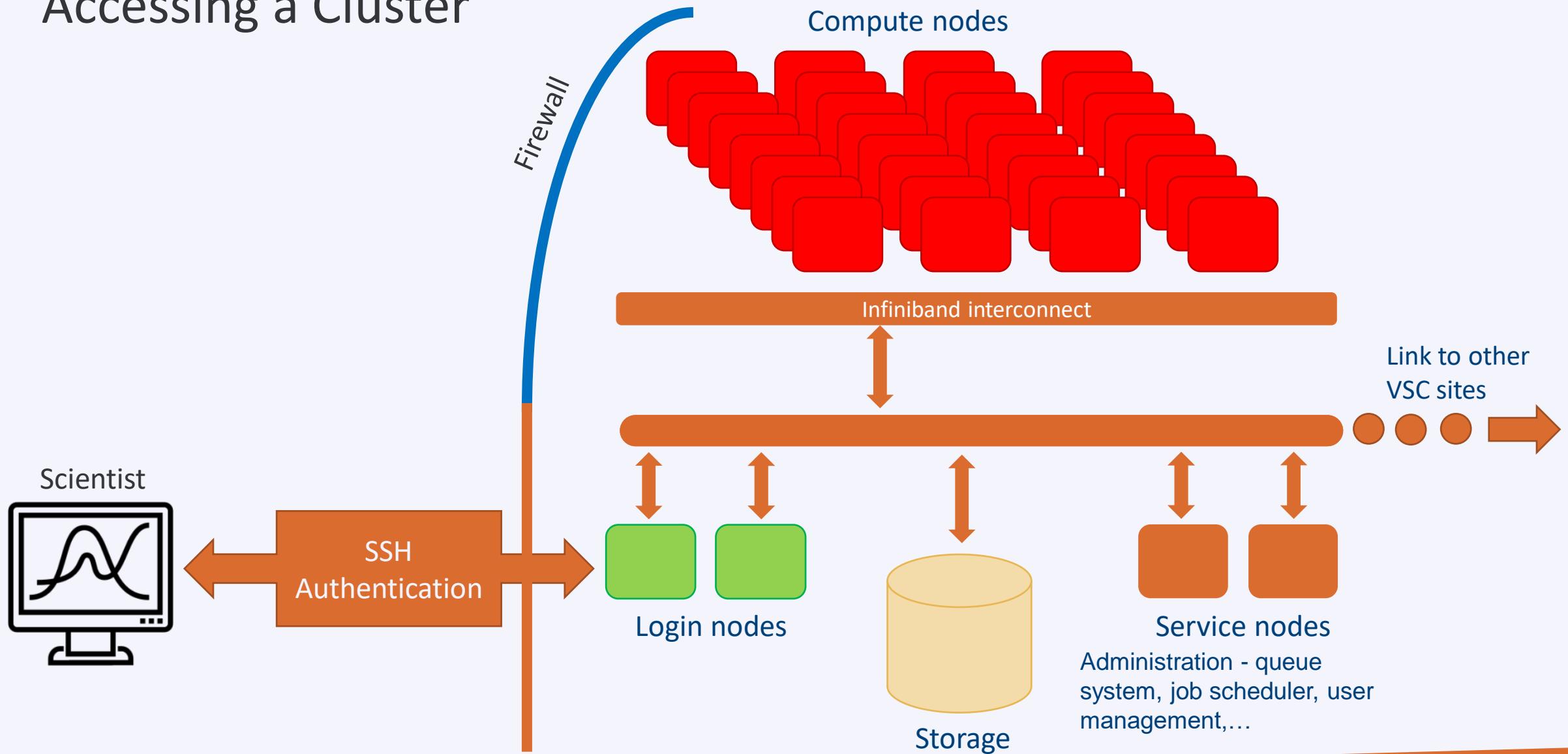


Storage	Staging folder
<b>Path</b>	/staging/leuven/stg_000XX
<b>Filesystem Type</b>	Lustre
<b>Access</b>	On demand, only Tier-2@KUL
<b>Backup</b>	None
<b>Default Quota</b>	None
<b>Extension</b>	On purchase, from 1 TB
<b>Usage</b>	Permanent; share with a group
<b>Remarks</b>	<ul style="list-style-type: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



## 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-/post-process your data/jobs
- To visualize your data
- To share files/folders

Tips

## Do NOT execute heavy-lifting tasks (core, memory)

Warning

Login nodes are shared resources

Instead, submit jobs: e.g. Compile your software on compute nodes.

Check `$ slurmtop` to see how busy the cluster is

# Login Hosts on Different Machines/partitions

**Windows**

- ✓ PuTTY
- ✓ Terminal
- ✓ NX

**Linux/Mac:**

- ✓ Terminal
- ✓ NX

Single-precision GPU,  
only for visualizations

To login, you need an active VSC number and a hostname

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

Cluster / Partition	<hostname>	Remark(s)
Genius	<b>login.hpc.kuleuven.be</b>	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

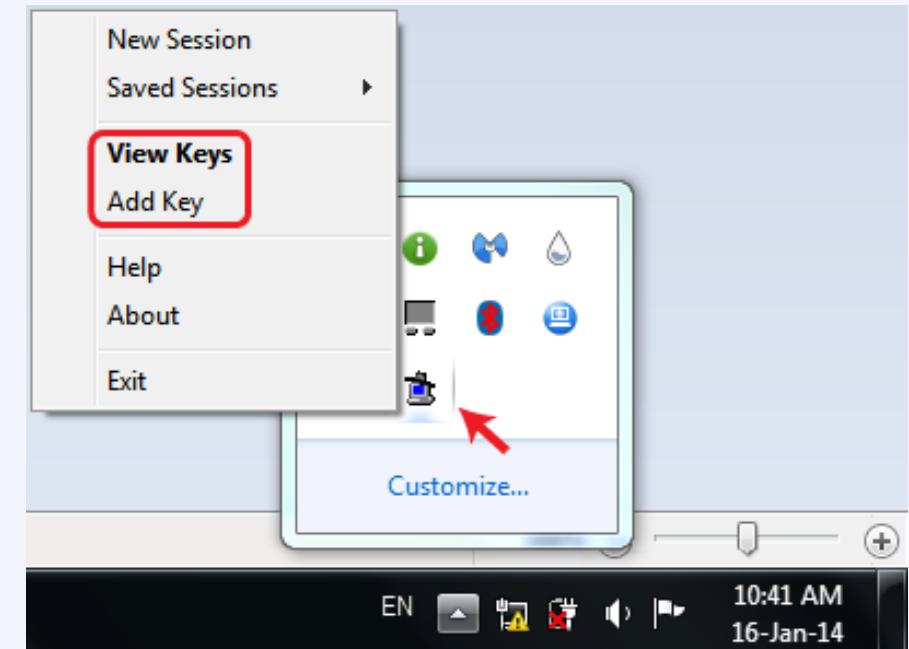
## Connection Setup

- PuTTY
- Terminal

# Activate Your SSH Agent

(Windows only)

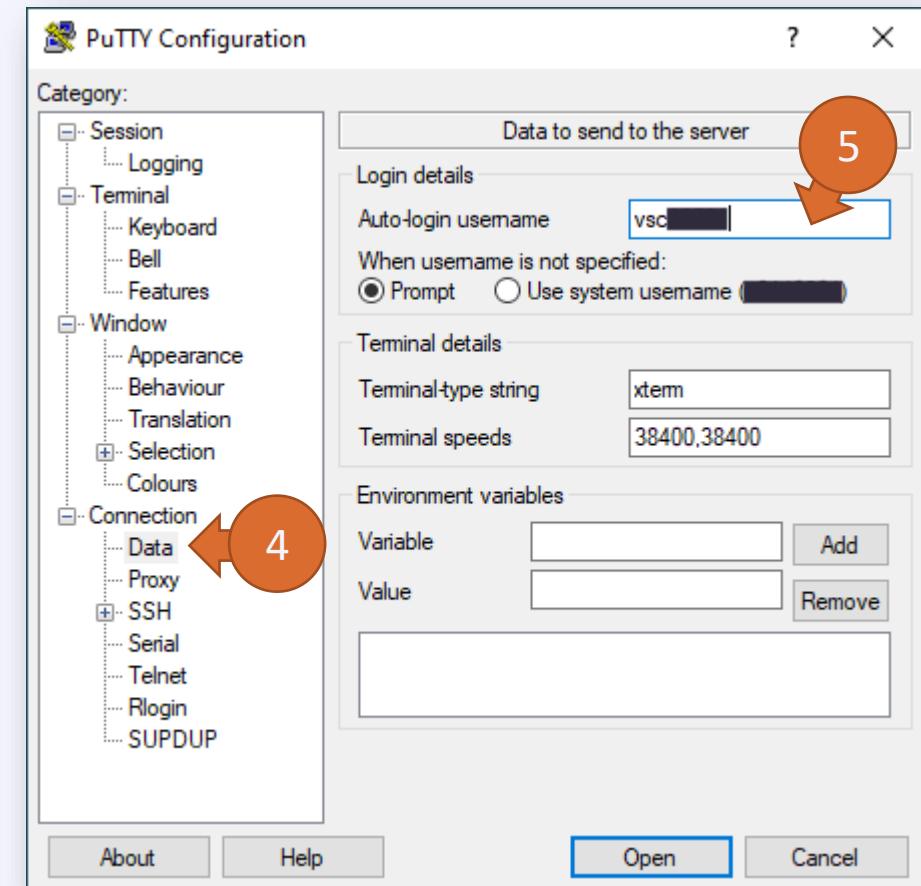
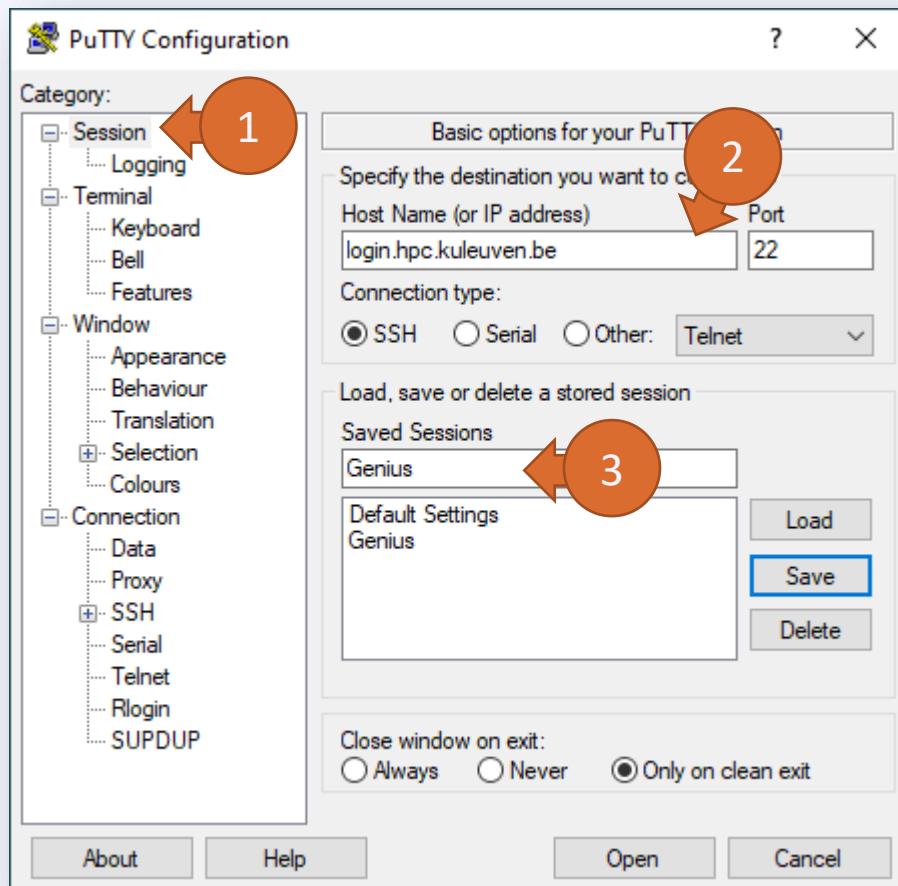
- ✓ When you install PuTTY, the agent called **Pageant** is automatically installed
- ✓ Open Pageant; it hides inside your bottom-right tray
- ✓ Click on “Add Key” and brows to your private key(s) folder
- ✓ After choosing a key, you are asked for a passphrase
- ✓ If successful, agent always remembers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



# Setup PuTTY

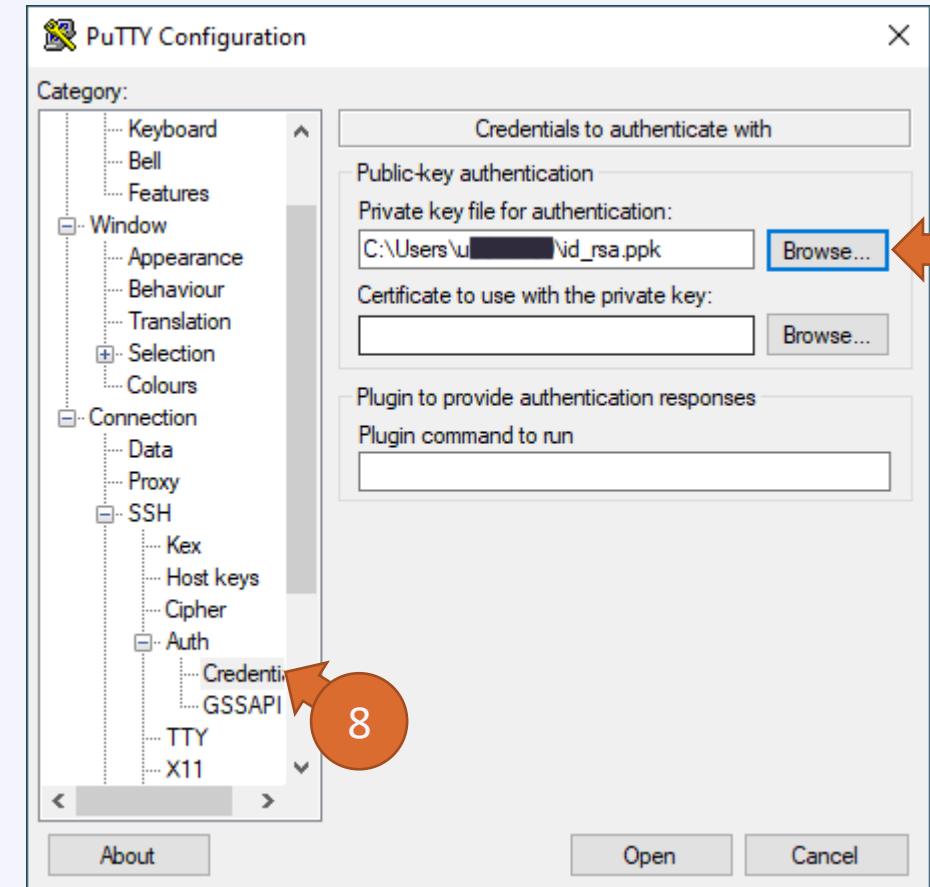
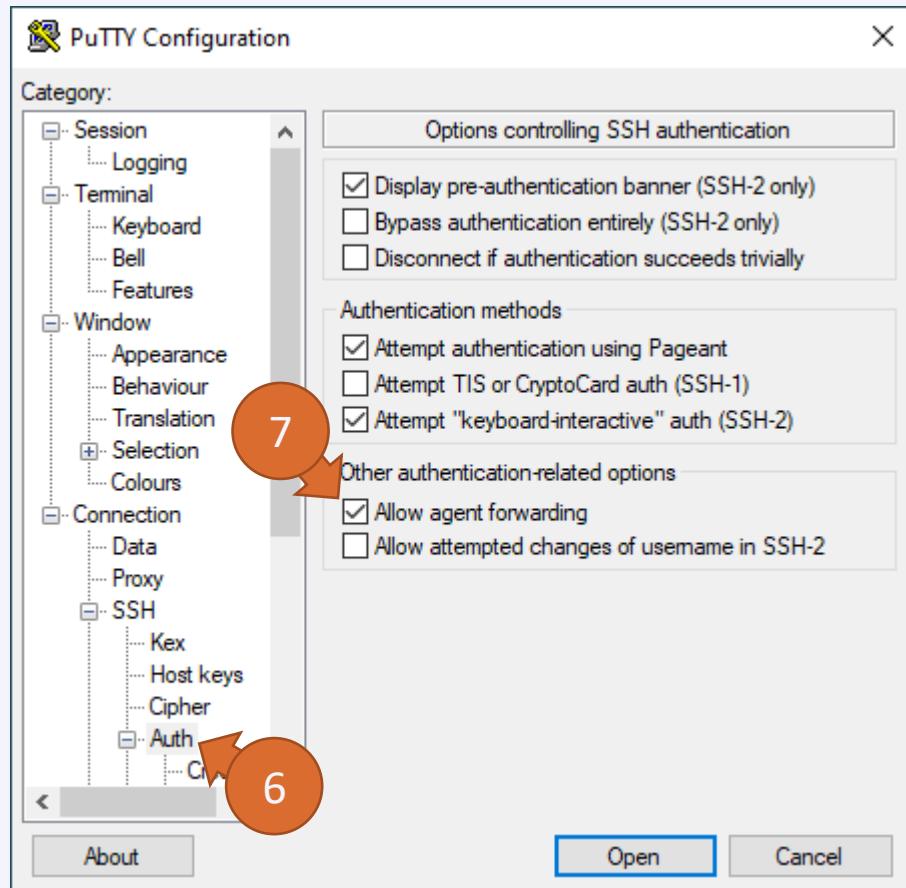
(Windows only)

Host Name:  
**login.hpc.kuleuven.be**



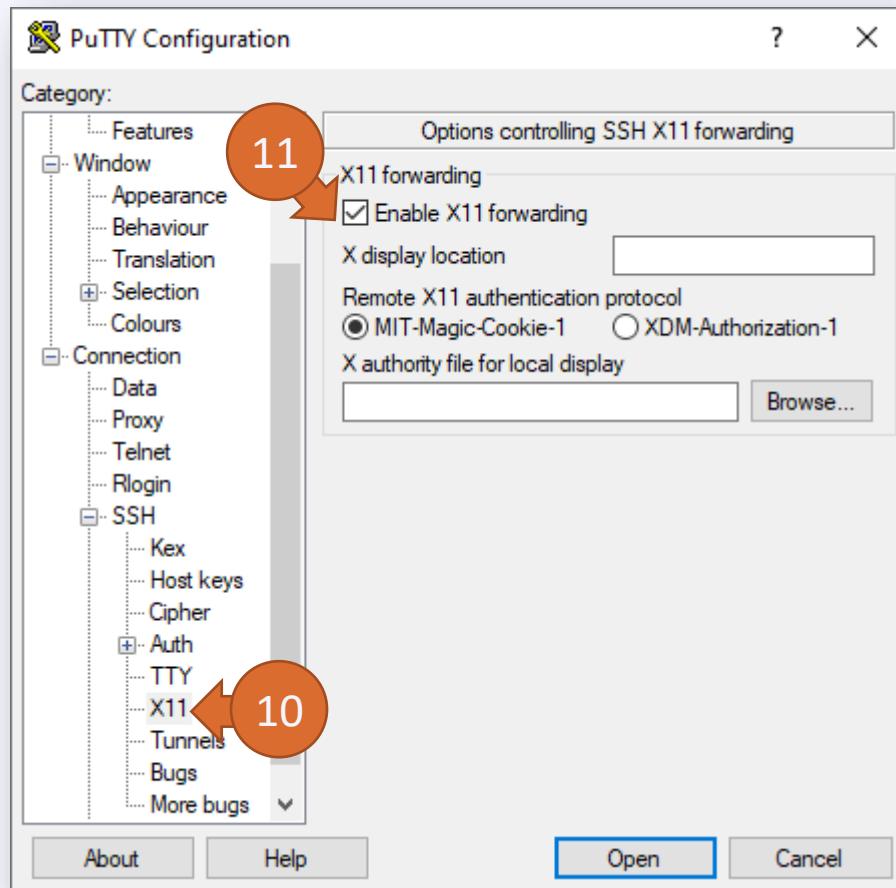
# Setup PuTTY

(Windows only)



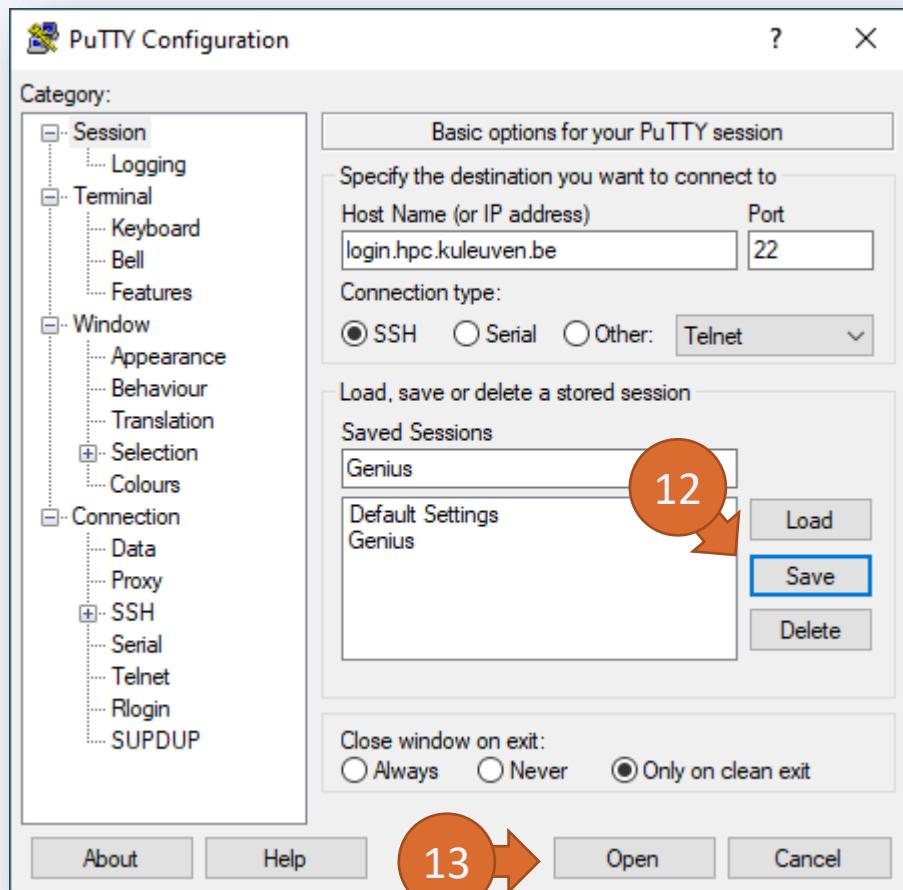
# Setup PuTTY

(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 the PuTTY terminal window. The top screenshot shows the initial connection message: 'Using username "vsc[REDACTED]". Authenticating with public key "the key pair that is used for my VSC account"'. The middle screenshot shows the message 'Passphrase for key "the key pair that is used for my VSC account":' followed by a password entry field. The bottom screenshot shows the final successful login message: 'Informatie over deze server (FQDN): tier2-p-login-2.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 tier2-p-login-2 [Oct/07 13:15] vsc[REDACTED]@tier2-p-login-2 ~ \$'. Orange arrows point from the text in the callout box to the 'Passphrase' prompt in the middle screenshot and the final login message in the bottom screenshot.

# Connecting via Terminal

(Linux and Mac)

- ✓ Use ssh to connect:

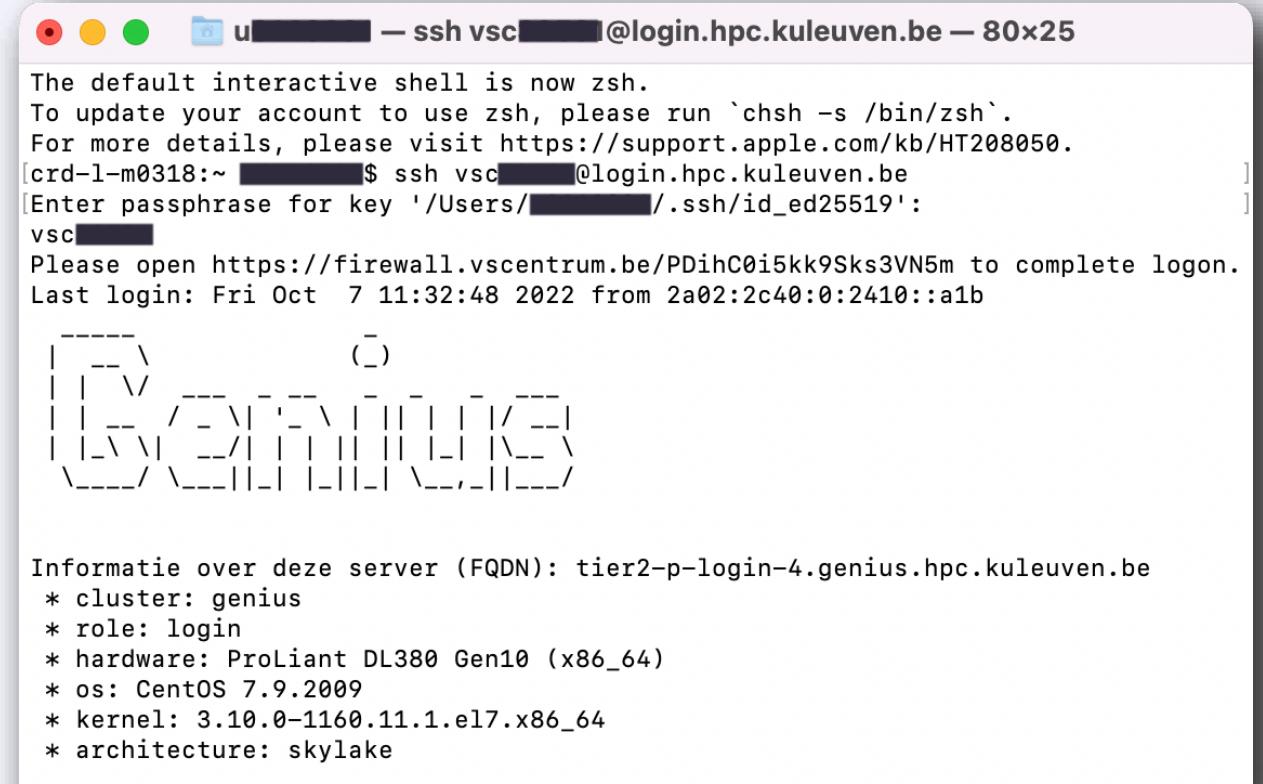
```
$ ssh vscXXXXX@<host_name>
```

- ✓ 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:~ vscXXXXX\$ 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  
[REDACTED]  
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

## (Linux and Mac)

# With SSH Agent

- ✓ 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 vscXXXXXX@<hostname>

**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:~ [REDACTED]$ ssh vsc[REDACTED]@login.hpc.kuleuven.be  
[Enter passphrase for key '/Users/[REDACTED]/.ssh/id_ed25519':  
vsc[REDACTED]  
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  
  
[REDACTED]  
|  _-- \  |  ( )  
| | \ \ / | | | |
| | - - / \ | - - \ | - - / |  
| | \ \ / | - - / | - - \ | - - / |  
| | \ \ / | - - / | - - \ | - - / |  
| | \ \ / | - - / | - - \ | - - / |  
  
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] vsc[REDACTED]@tier2-p-login-4 ~ $ [REDACTED]
```



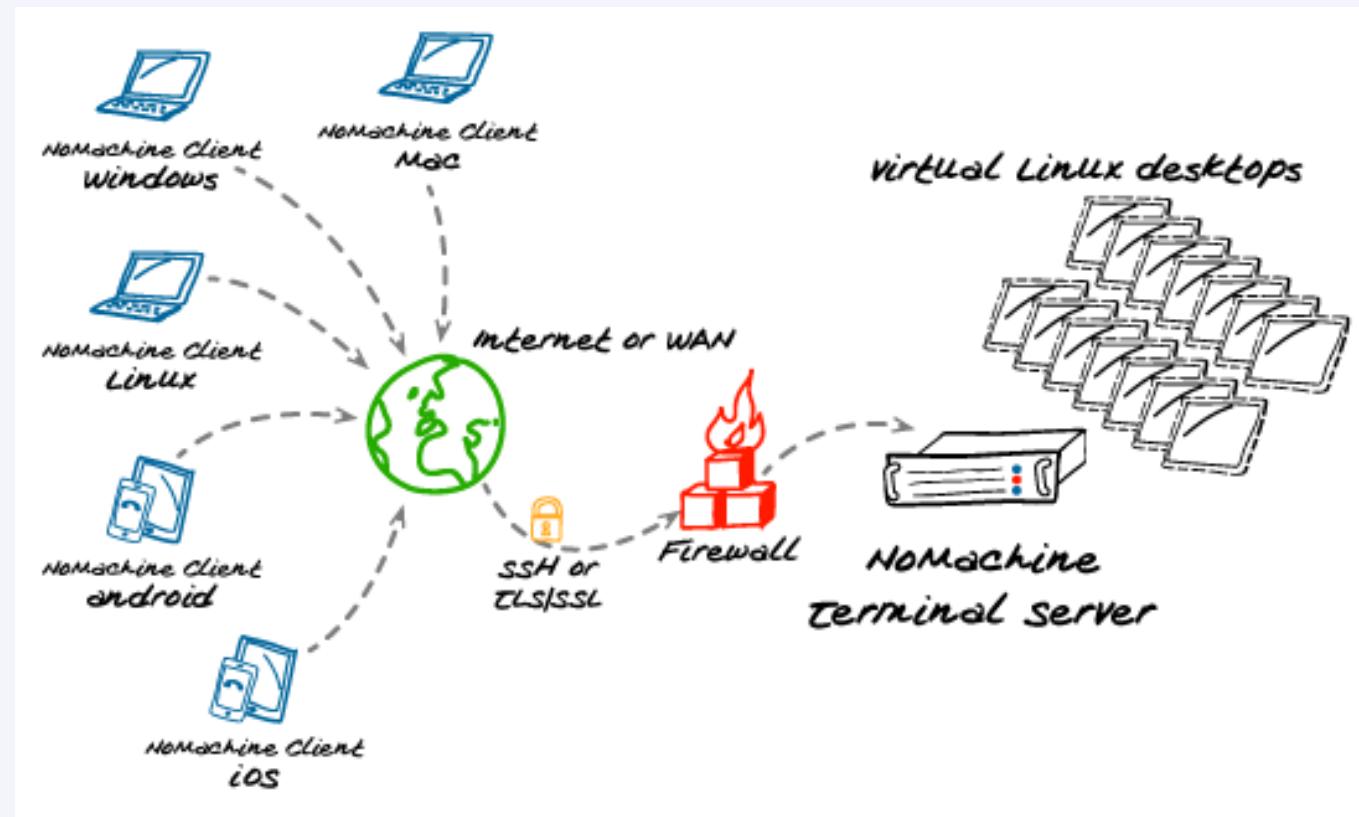
NX

- 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

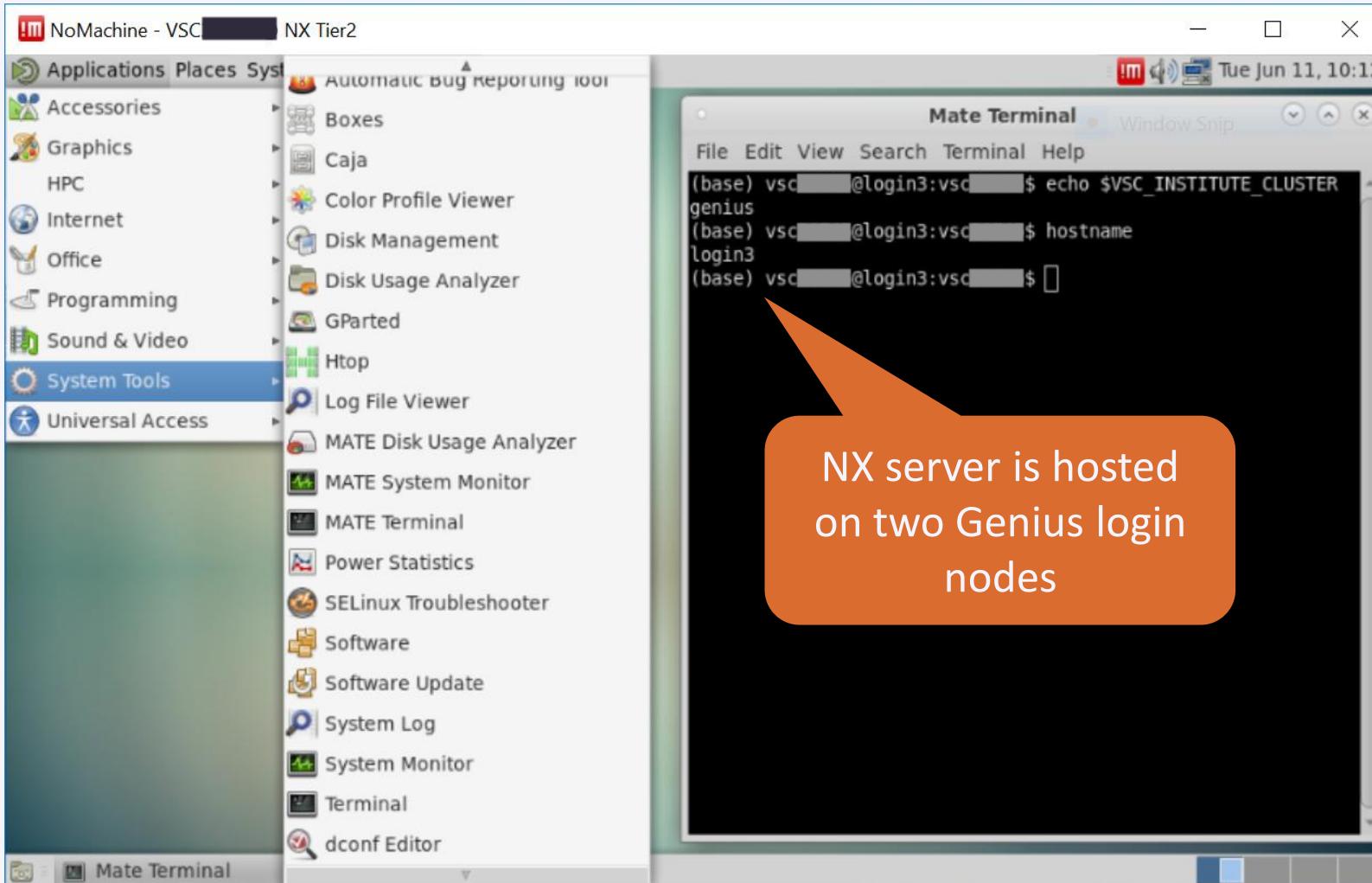
## Remark:

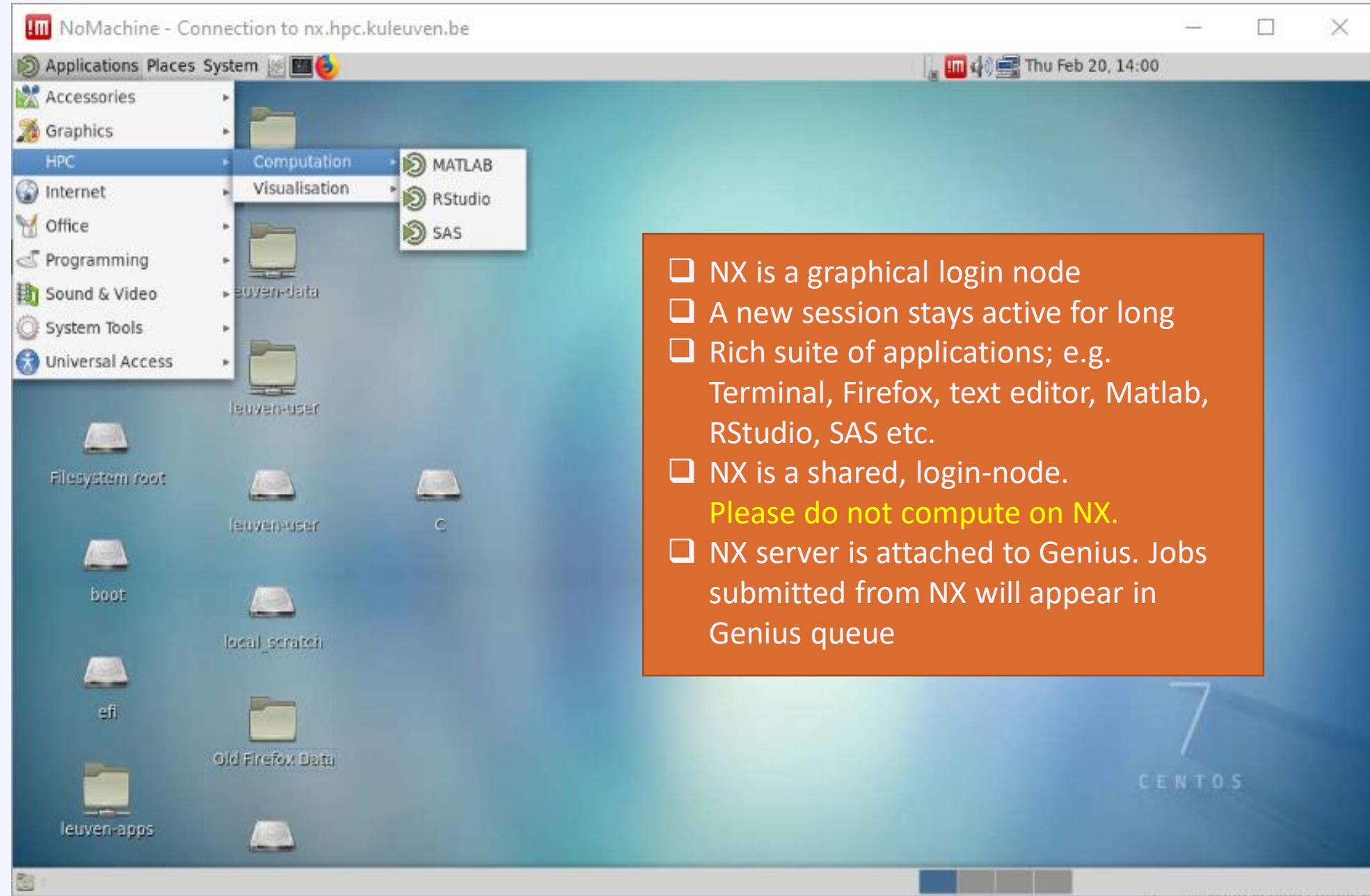
NX is shared among tens of users

**Do not compute/test your code there**

Instead, submit jobs from NX to compute nodes

# NX virtual desktop





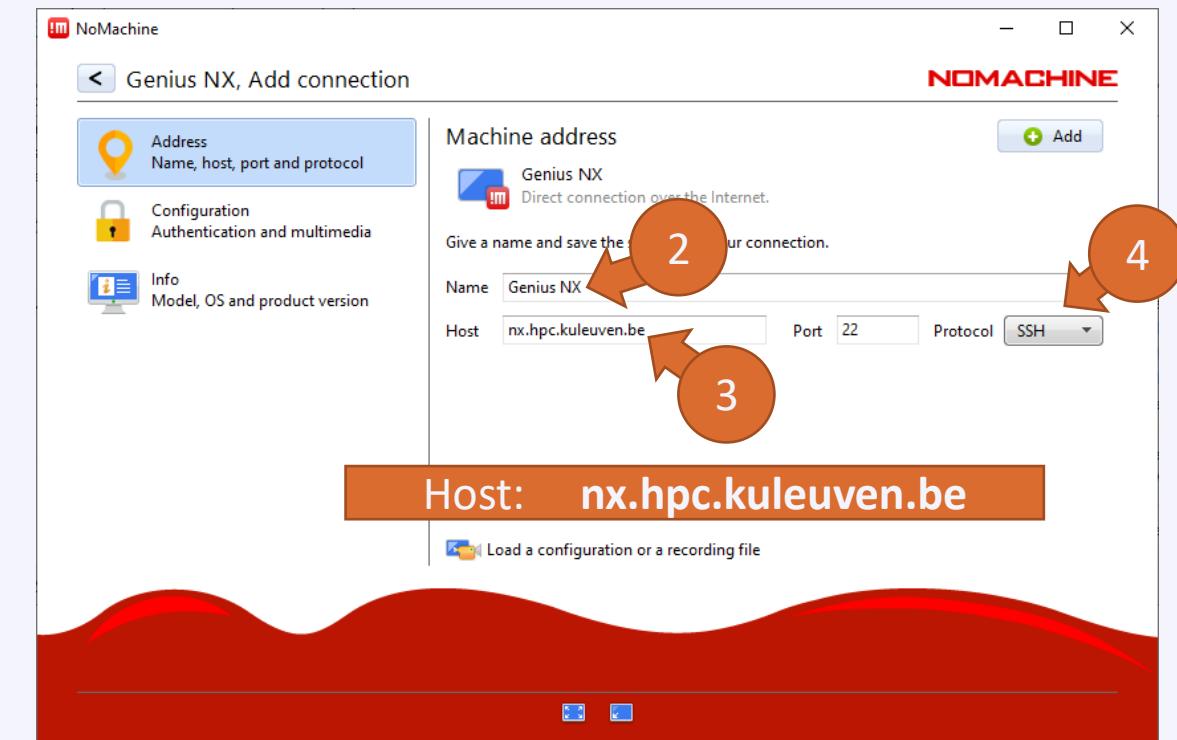
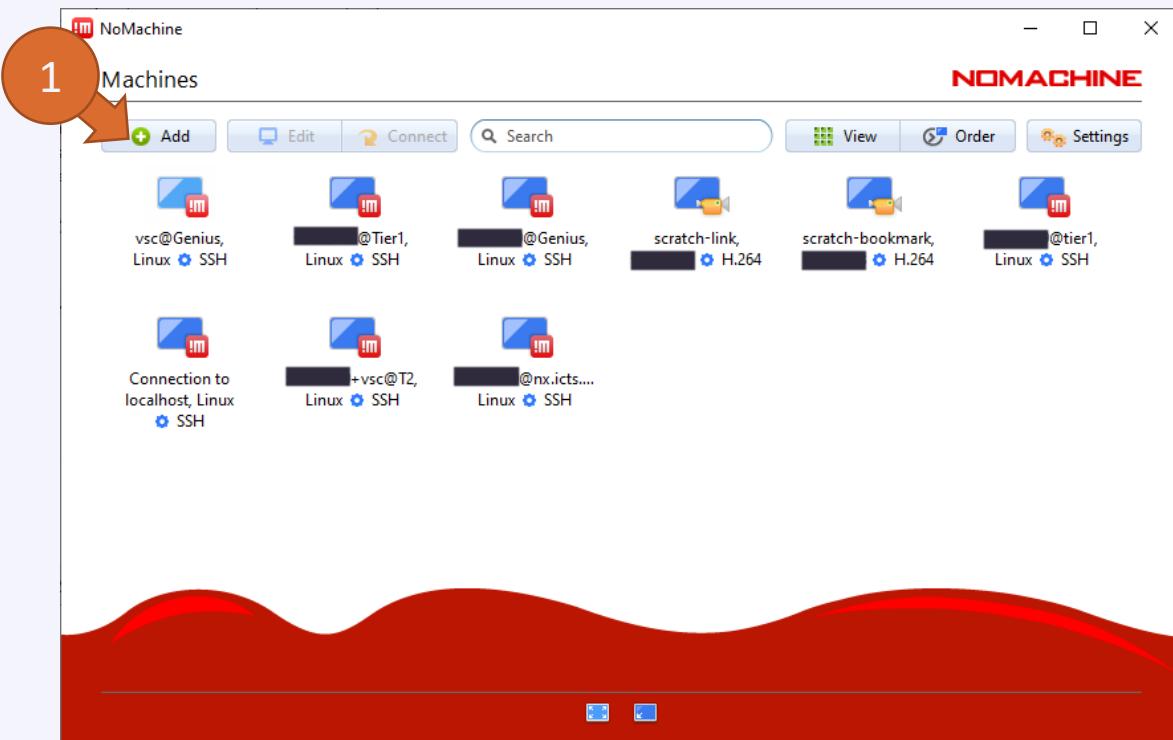
- NX is a graphical login node
- A new session stays active for long
- Rich suite of applications; e.g.  
Terminal, Firefox, text editor, Matlab,  
RStudio, SAS etc.
- NX is a shared, login-node.  
**Please do not compute on NX.**
- NX server is attached to Genius. Jobs  
submitted from NX will appear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

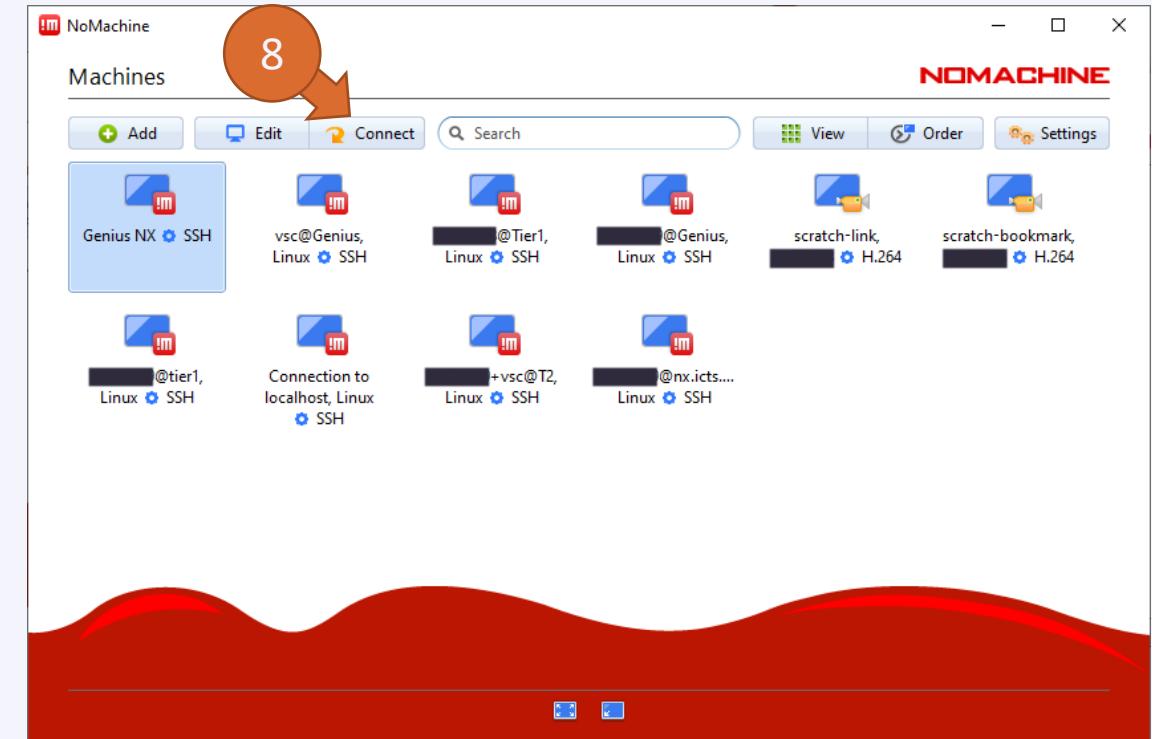
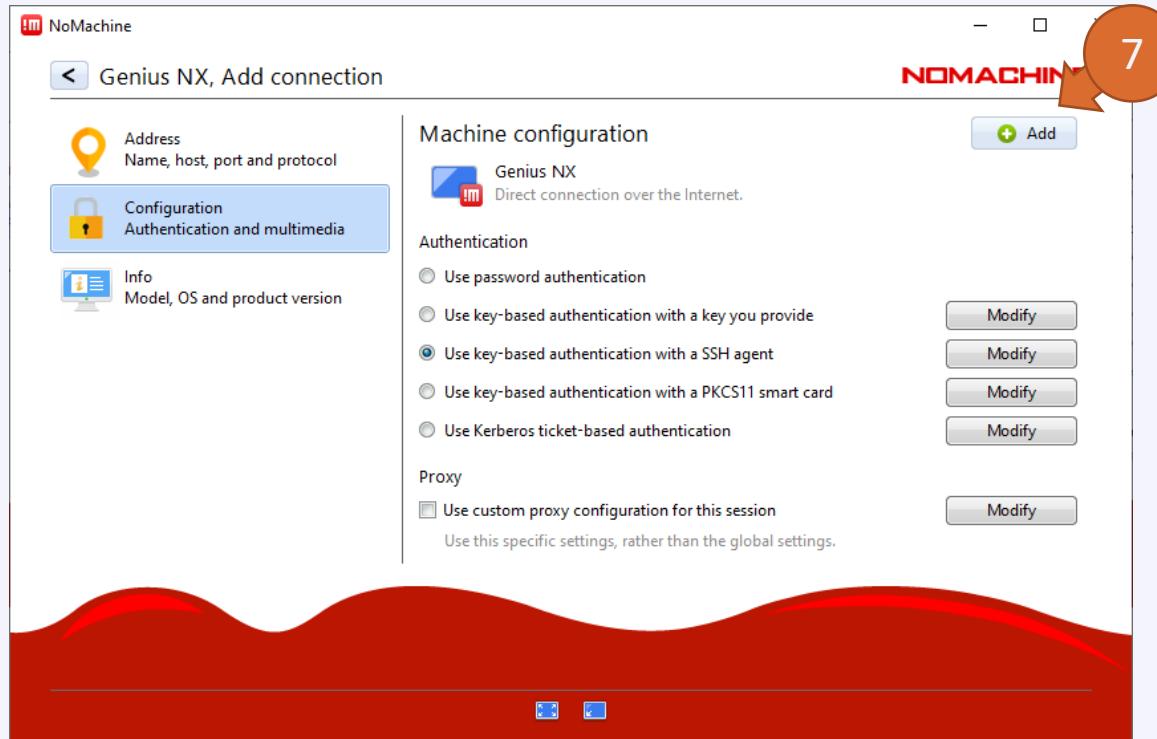
# Setup NX



# Setup NX

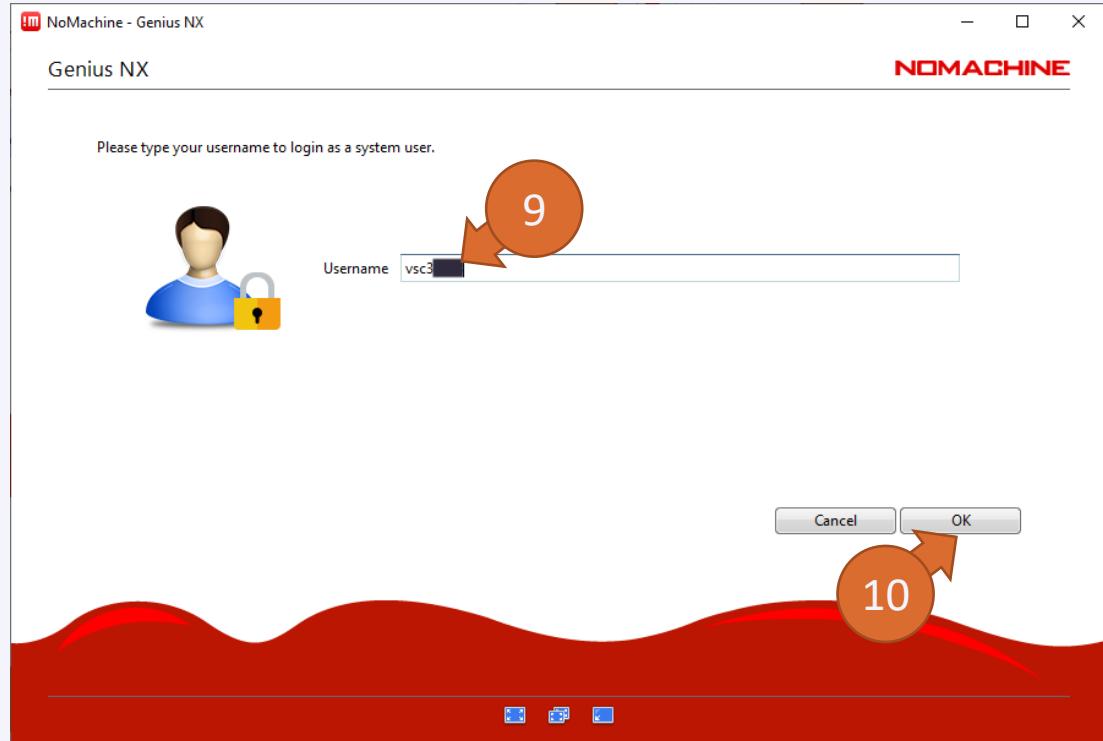


# Setup NX

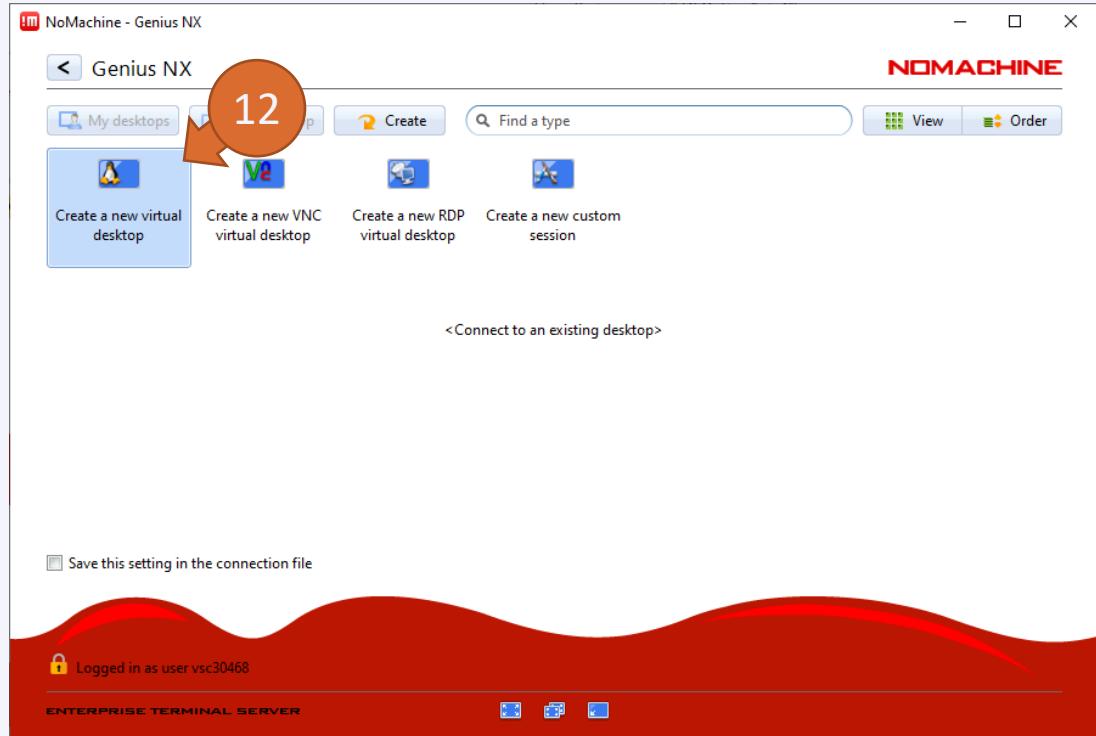


# Setup NX

First open putty connection to open firewall or go to [firewall.vscentrum.be](http://firewall.vscentrum.be) for further authentication

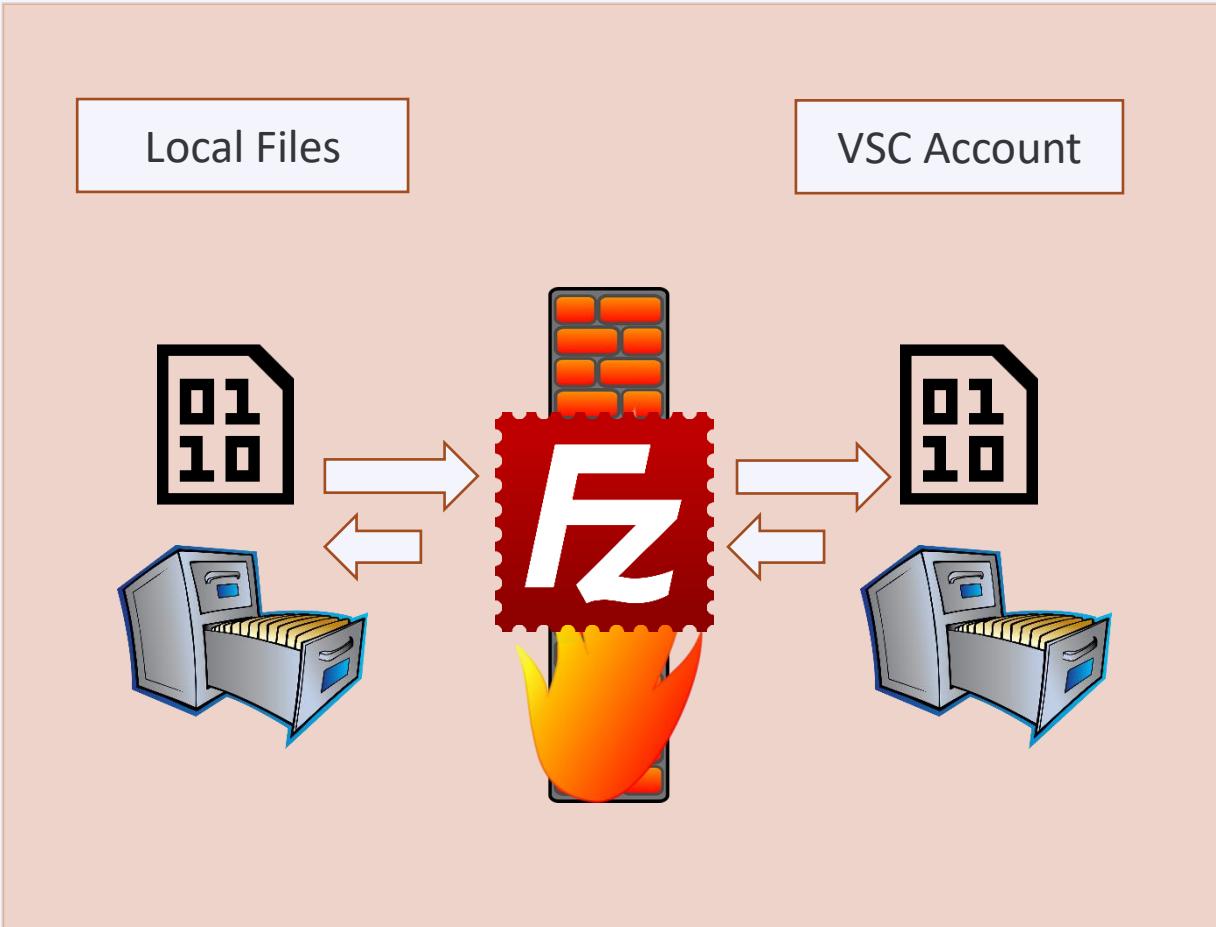


# Setup NX



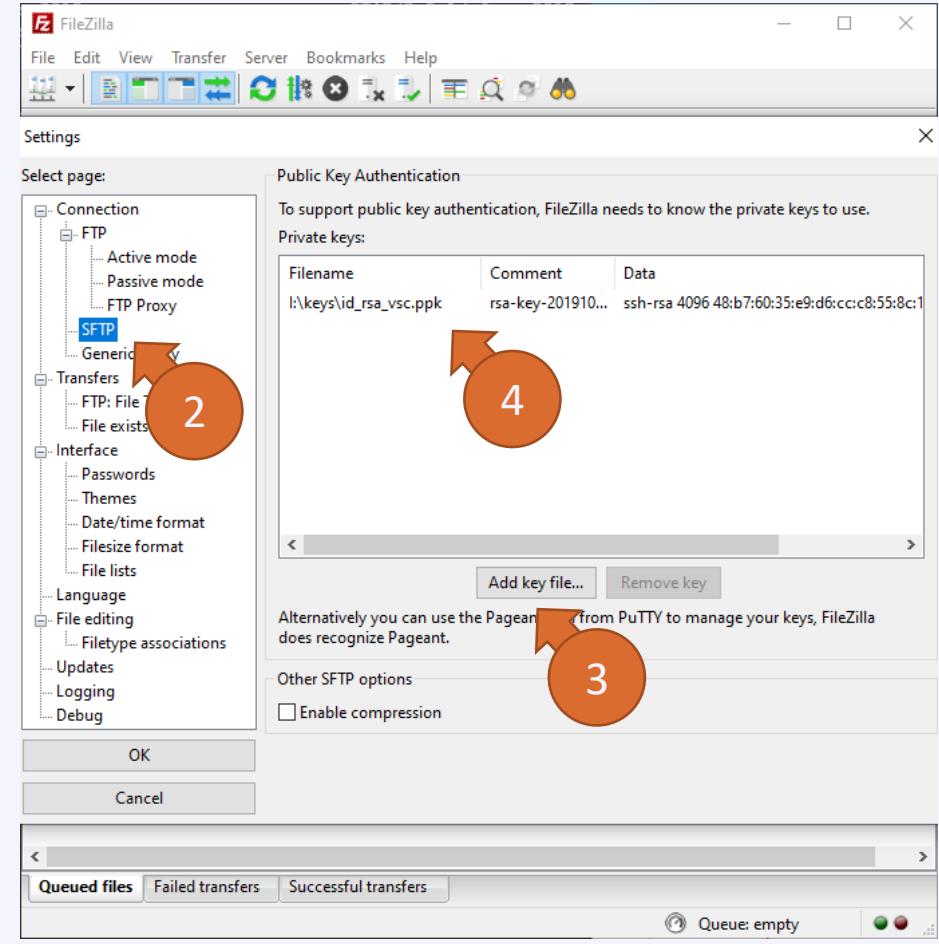
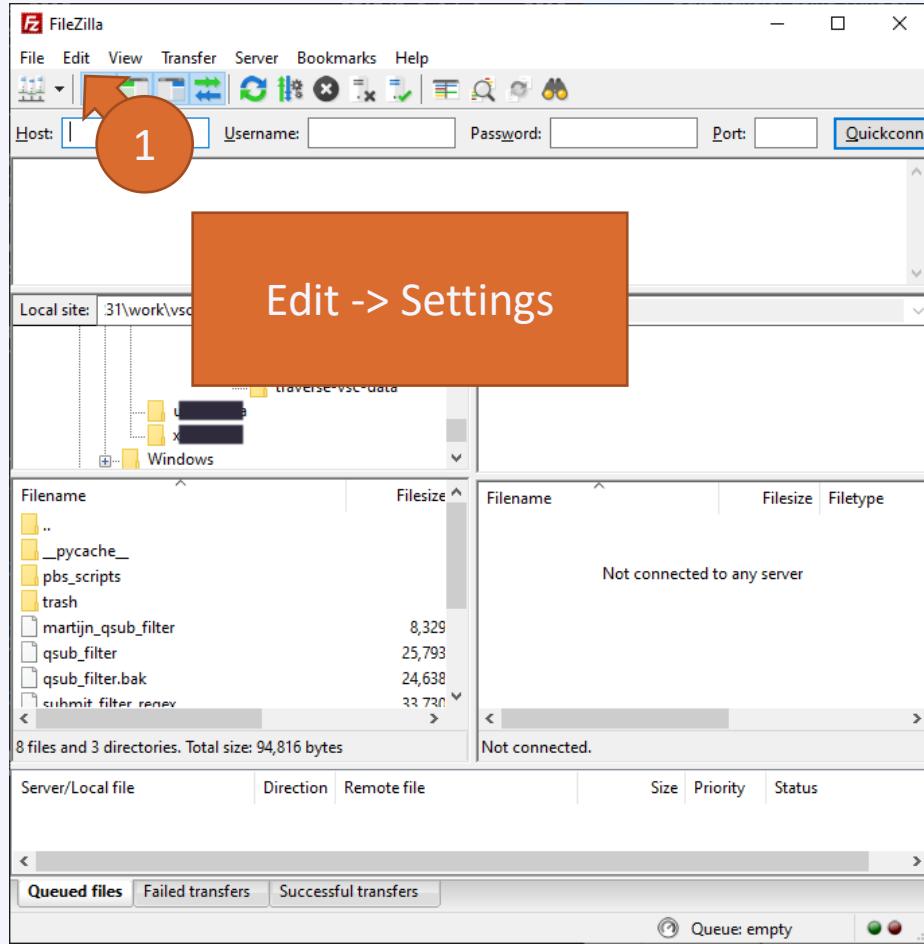
## File Transfer with FileZilla

# File transfer



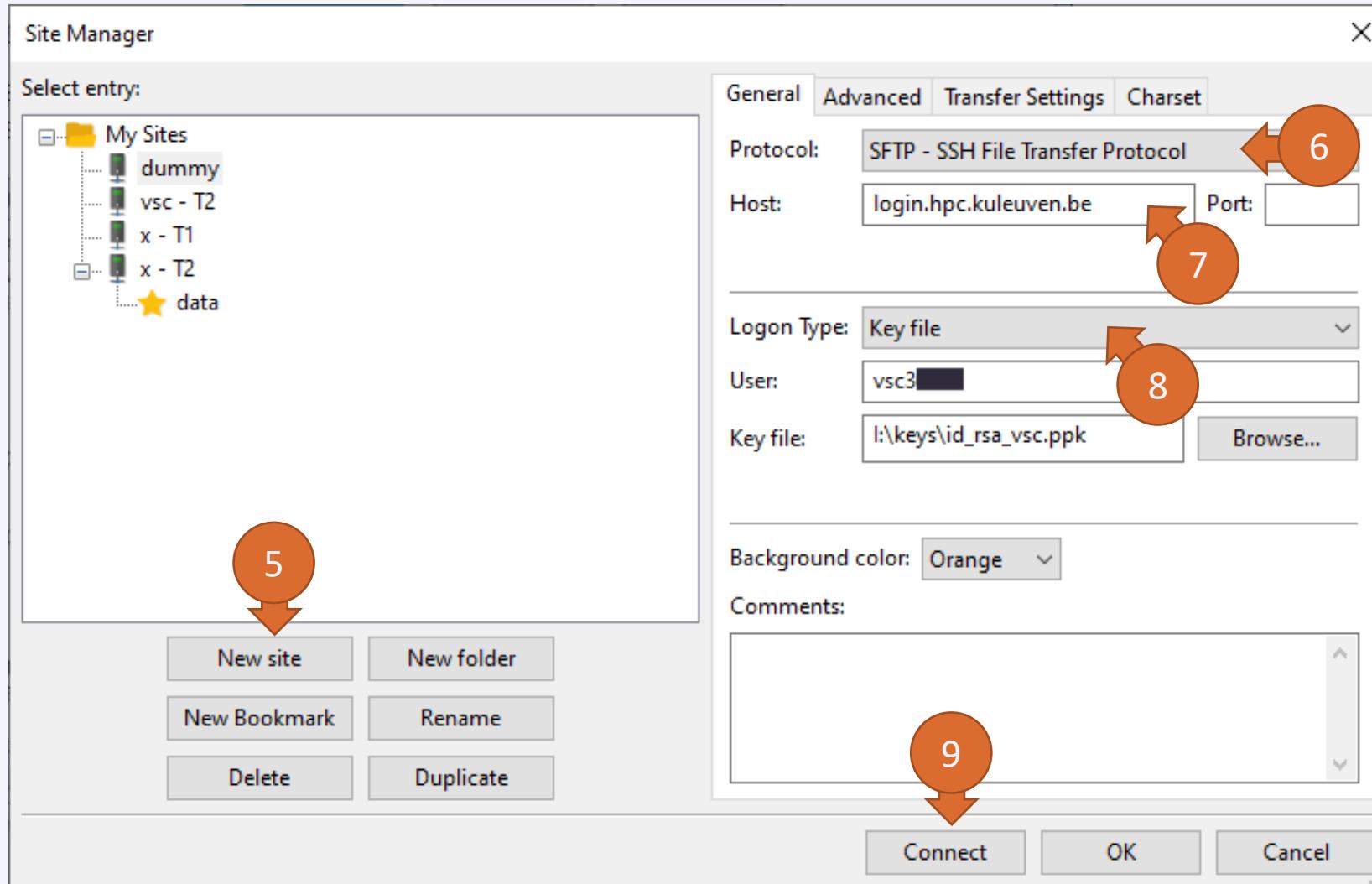
Application	OS
<a href="#">FileZilla</a>	Windows, Linux, Mac
<a href="#">WinSCP</a>	Windows
rsync, scp	Linux, Mac
<a href="#">Globus</a>	Window, Linux, Mac

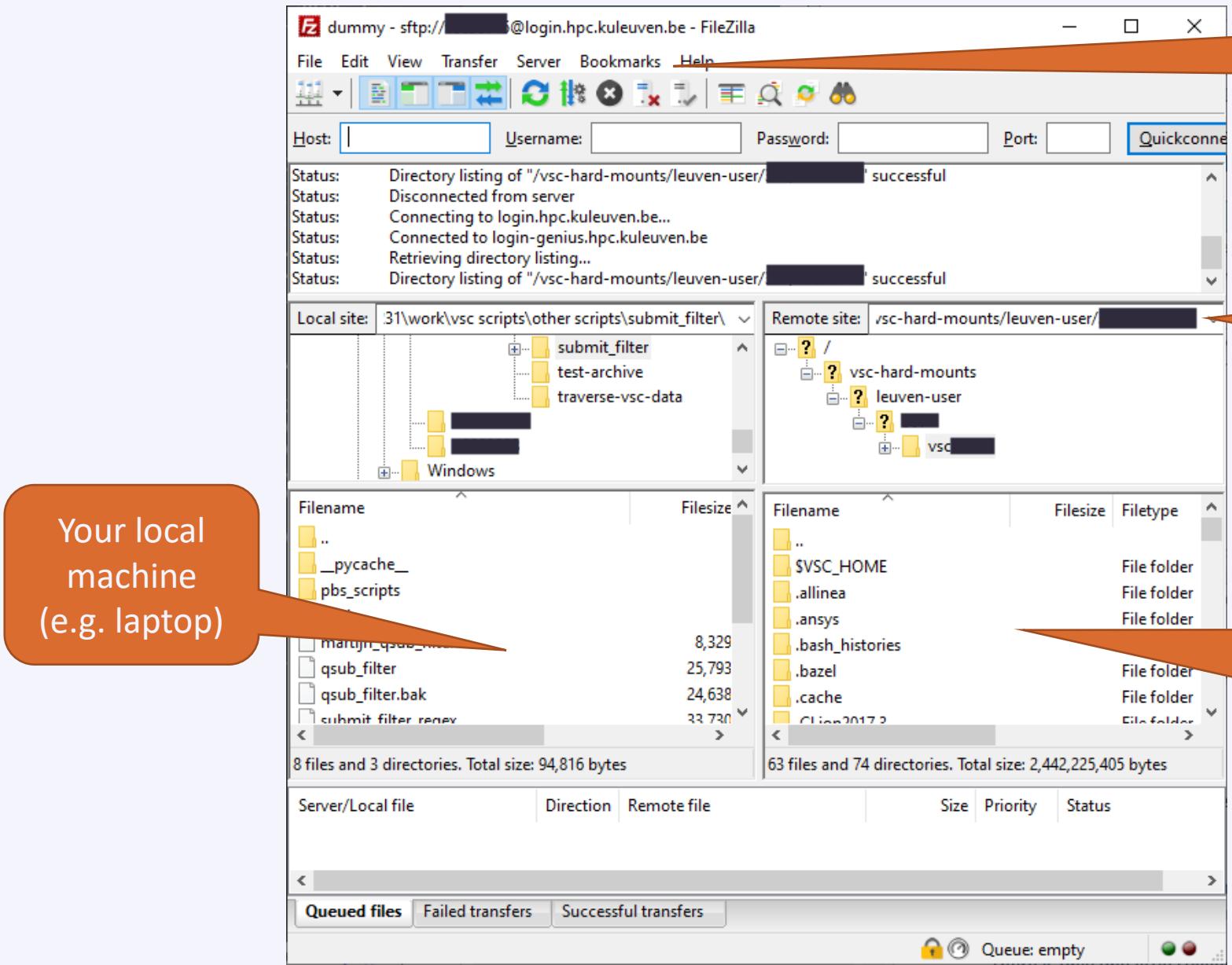
# File transfer



# File transfer

First open putty connection to open firewall or go to [firewall.vscentrum.be](http://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!



An orange rectangular overlay covers the bottom left portion of the image, containing the text "Open Ondemand".

**Open Ondemand**



# Open OnDemand

- ✓ Access clusters 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





E.g. to start a Jupyter Notebook

- ✓ Pick a valid Slurm credit account
- ✓ Default partition: batch  
interactive: Max 8 cores, 1 GPU (shard), 16 hr
- ✓ Default resources:  
1 core, 1 hour, 3400MB RAM, no GPU

(At the moment) you cannot use scratch and staging

**Jupyter Lab**

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

---

**Interactive Apps**

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

**Account**

**Partition**

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

**Number of hours**

**Number of cores**

**Required memory per core in megabytes**

**Number of nodes**

**Number of gpu's**

[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: Rocky Linux 8.x
- ✓ Toolchains:
  - Intel (icc, icpc, ifort; Intel MPI; Intel MKL)
  - FOSS (gcc, g++, gfortran; OpenMPI; ScaLAPACK, OpenBLAS, FFTW)
  - Toolchain year on Genius from 2018a; on wlCE from 2021a
- ✓ Note: **Never mix FOSS and Intel compilers (gives dependency conflict)**

Command	Remark
module av	List all available modules
module av Python	List all Python-related modules
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
- ✓ Specific Python/R packages is managed by users via conda
- ✓ 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

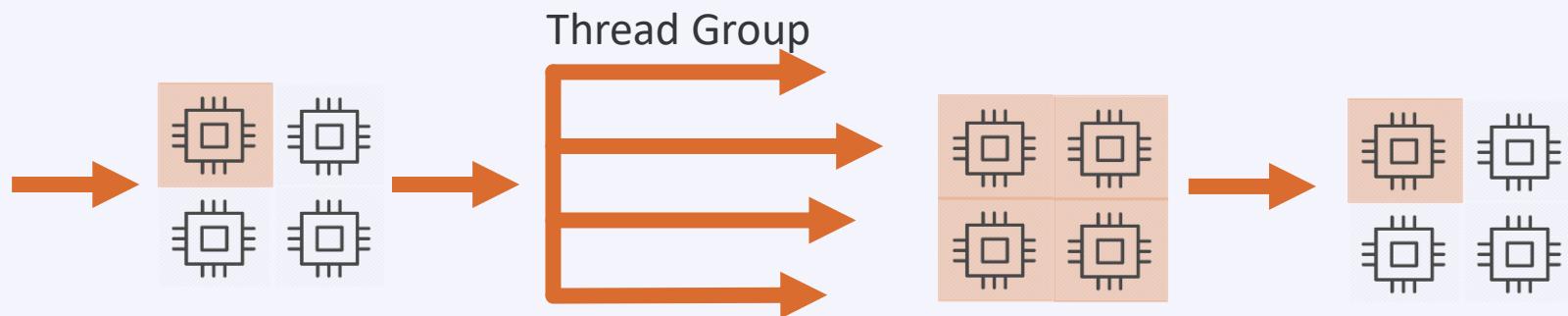
- ✓ **Nodes:** how many compute servers to request?
- ✓ **Cores:** how many cores per node to use?
- ✓ **Memory requirement:** how much memory each core needs?
- ✓ **Partition:** gpu, bigmem, superdome, amd
- ✓ **Walltime:** how long to use resources?
- ✓ **Storage:** how much storage (data, scratch,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



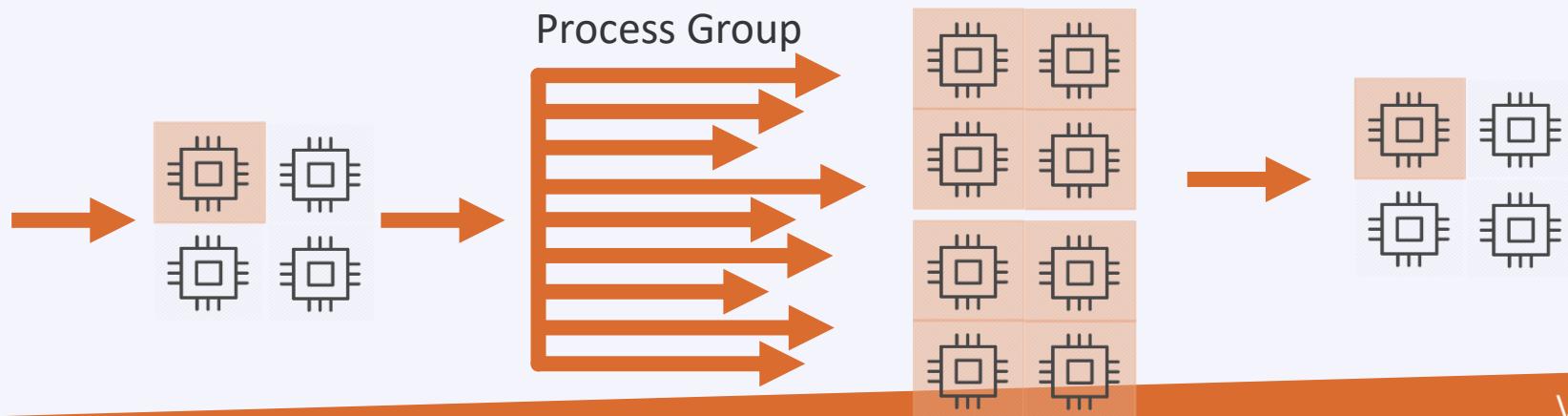
## Serial Application (1 process on 1 core)



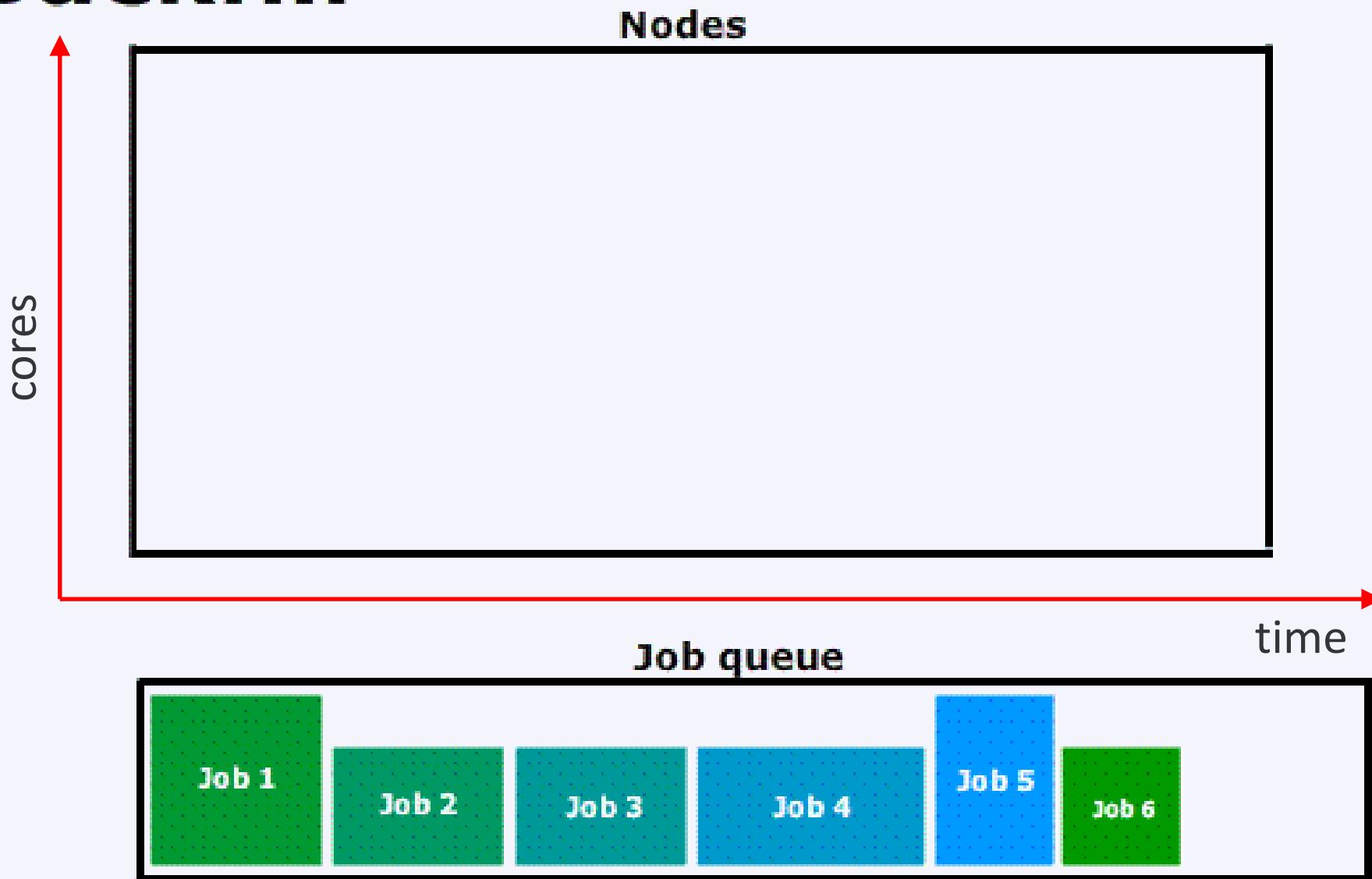
## Multi-Core Appl. (N threads on N cores from **1 node**)



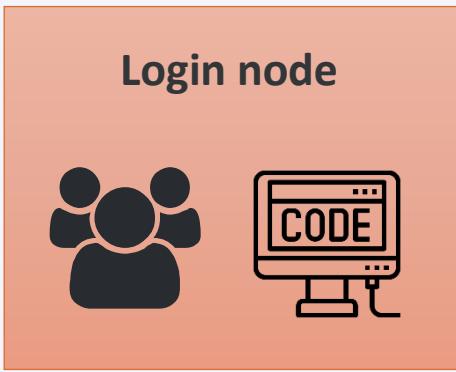
## Distributed Appl. (many processes on multiple cores/nodes)



# Backfill



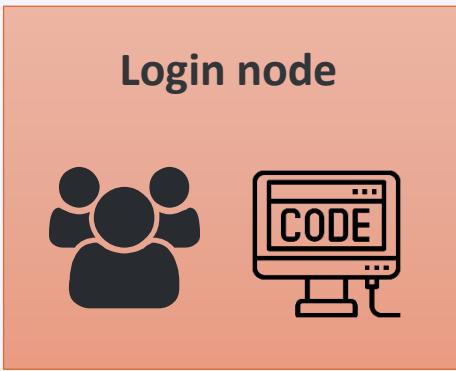
## Interactive Job



`srun ...`

Compute nodes

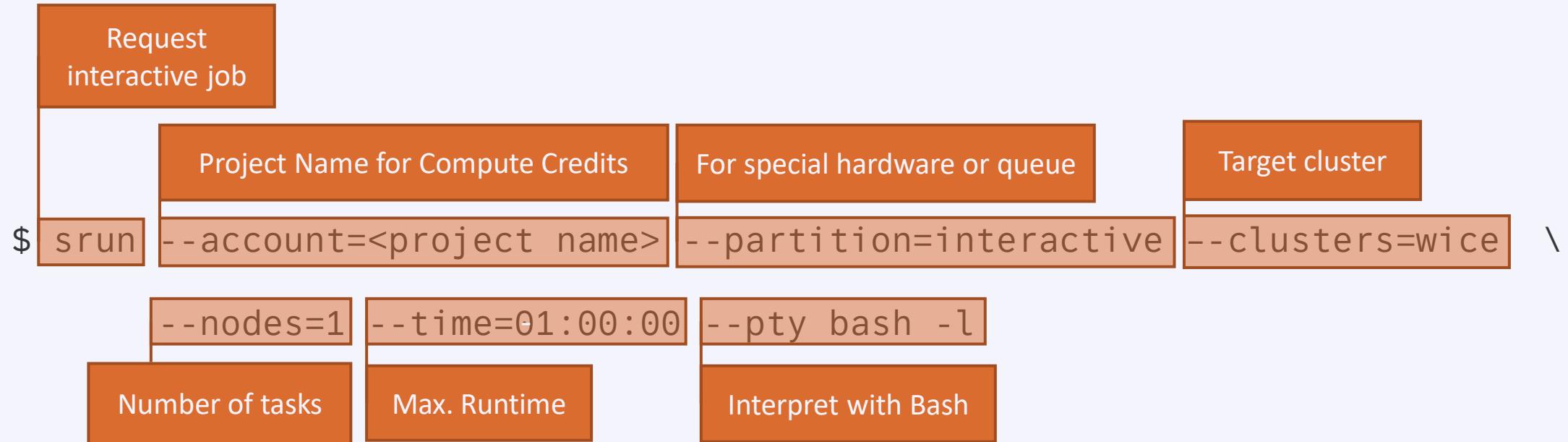
## Batch Job



`sbatch script.slurm`

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 (e.g. Hybrid MPI + OpenMP)
--partition=<partition_name>	Default: "batch" partition
--mem-per-cpu=<XG>	Minimum memory per core
--time=<dd-hh:mm:ss>	e.g. 1-12:30:00
--job-name=<job_name>	
--output=<file_template>	STDOUT; default="slurm-%j.out"
--error=<file_template>	Default: redirect to STDOUT
--mail-type=FAIL,BEGIN,END	
--mail-user=<email-address>	
--export=<ALL, key=value>	Additional variables to pass to job
--account=<account_name>	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	--clusters	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
-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)

# Interactive Jobs

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

```
$ srun --clusters=wice --account=lp_hpcinfo --pty /bin/bash -l
```

- ✓ Interactive job with X-forwarding

```
$ srun --clusters=wice --account=lp_hpcinfo -x11 --pty /bin/bash -l
```

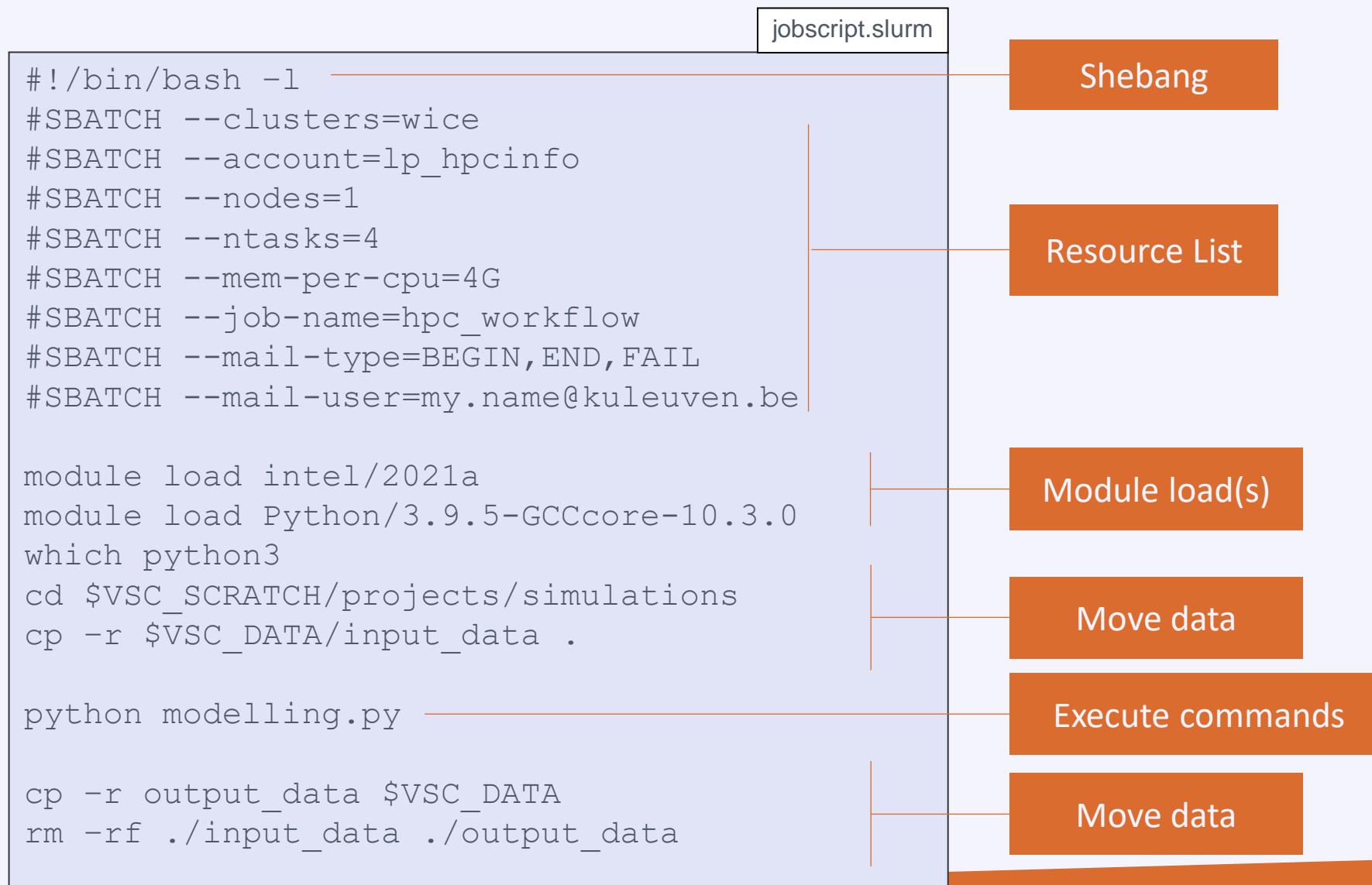
- ✓ Request fraction of a node from interactive partition

```
$ srun --clusters=wice --account=lp_hpcinfo --nodes=1 --ntasks=4 \
--partition=interactive --pty /bin/bash -l
```

- ✓ Request a GPU accelerator

```
$ srun --clusters=wice --account=lp_hpcinfo --nodes=1 --ntasks=18 \
--partition=gpu --gpus-per-node=1 --pty /bin/bash -l
```

# Example Slurm Job Script



# Batch Workload

Job Script

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

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

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-<JobID>.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/rocky8/icelake/2021a/
softwar
e/intel-compilers/2021.2.0/compile
r/2021.2.0/linux/bin/intel64/icc
cp: cannot stat '/apps/leuven/trai
ning/test': No such file or directo
ry
Hello World
```

Resource Summary

Stdout

# Output File

Example

```
$ scontrol show job --clusters=wice 60049330
JobId=60049330 JobName=f70.slurm
  UserId=vsc3... (253...) GroupId=vsc3... (253...)
Account=lp_my_project QOS=lp_my_project
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](https://slurm.schedmd.com/resource_limits.html)

# Other Partitions on wlCE

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
```

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 --clusters=wice <JobID> \$ slurm_jobinfo <JobID>	Detailed job info (very useful to diagnose issues)
\$ squeue --clusters=wice --long	Status of all recent jobs
\$ squeue --clusters=wice --start	Give a rough estimate of start time
\$ sinfo --clusters=wice	Info about the state of available partitions and nodes
\$ sacct --clusters=wice --batch --job <JobID>	Show minimal info about a queue or partition (-p)
\$ slurmtop \$ scontrol --clusters=wice show node <hostname>	Overview of the cluster 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

# 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

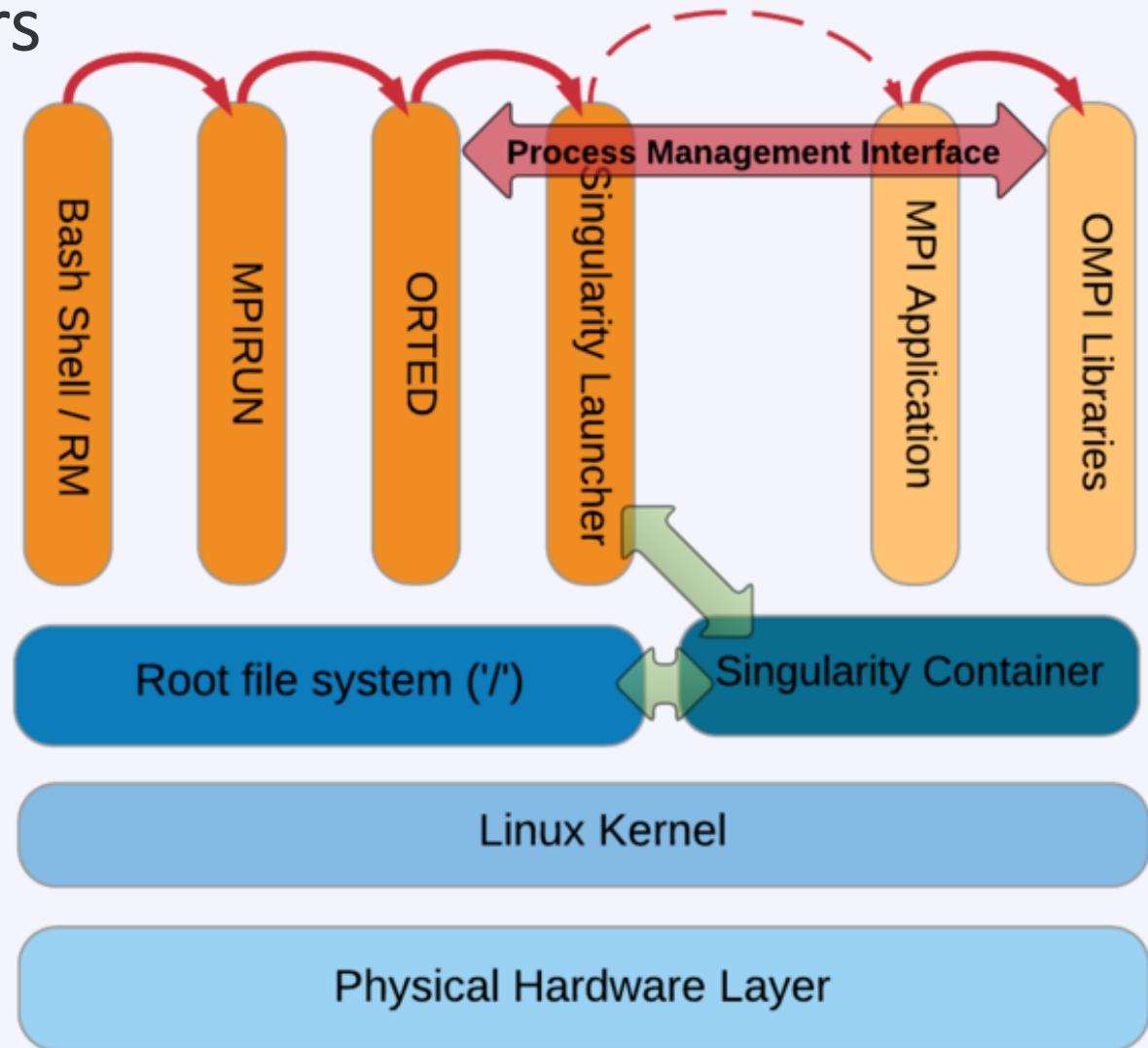
Interactive Job

```
$ srun --account=lp_myproject --clusters=wice \
    --partition=interactive --nodes=1 --time=16:00:00 \
    --gpus-per-node=1 --pty bash -l
```

# Apptainer (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
script.slurm
#SBATCH --time=30:00
#SBATCH --nodes=2
#SBATCH --ntasks=72
#SBATCH --clusters=wice
#SBATCH --account=lp_hpcinfo
apptainer run Project.simg
./model.exe
```



# 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?         | \$ sam-quote            |
| ✓ How to check available credits? | \$ sam-balance          |
| ✓ Did the new credits arrive?     | \$ sam-list-allocations |

# Credits 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 --ntasks-per-node 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 --account=<account-name>	Job statements from an account



# Conda Package Management: Python and R

# Conda on wICE

[https://docs.vsc.eu/en/latest/software/python\\_package\\_management.html?highlight=miniconda#install-python-packages-using-conda](https://docs.vsc.eu/en/latest/software/python_package_management.html?highlight=miniconda#install-python-packages-using-conda)

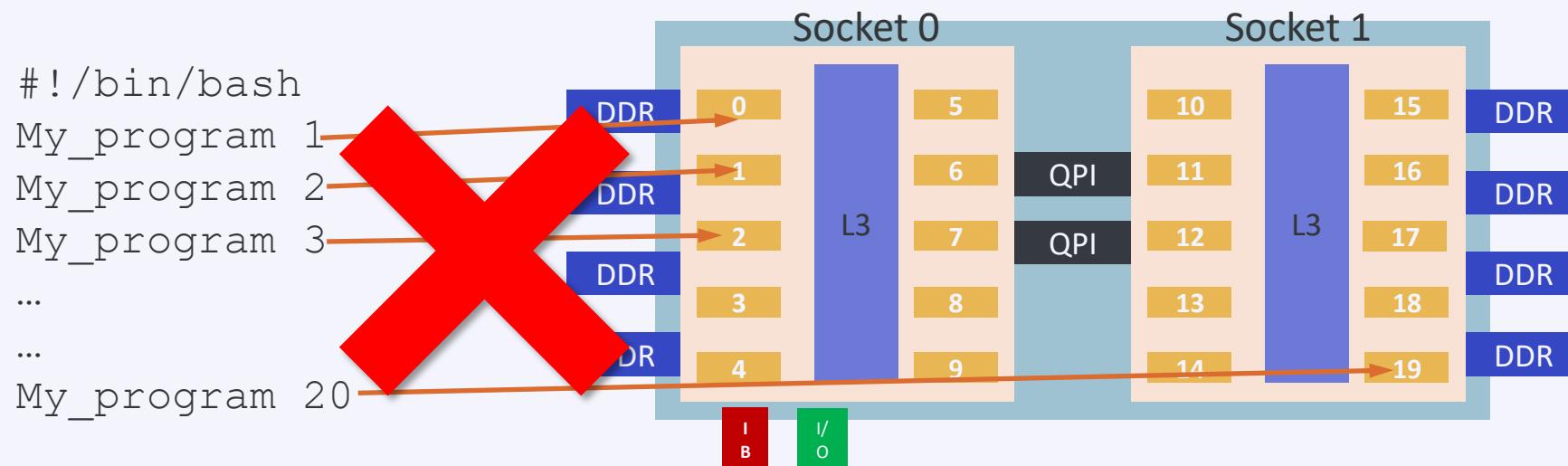
- ✓ Good practice would be 2 separate paths of conda installations for each cluster.
- ✓ Beginning of the slurm script matters. Use `#!/bin/bash -l` not `#!/bin/bash` to load the env properly.
- ✓ Universal `.bashrc`

```
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

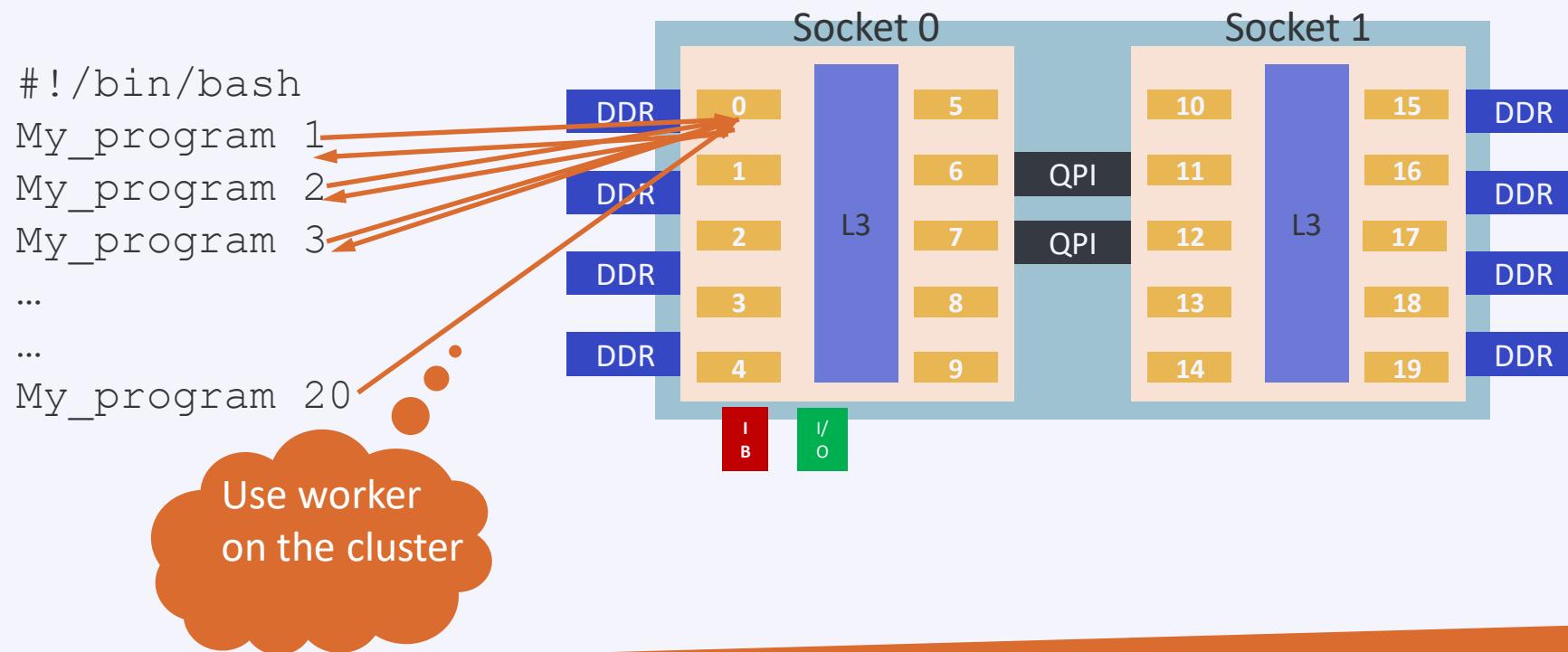
# Sequential...

- ✓ 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

Talk to us about  
worker framework

- ✓ 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

Lucky  
you!

# Use case: parameter exploration

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

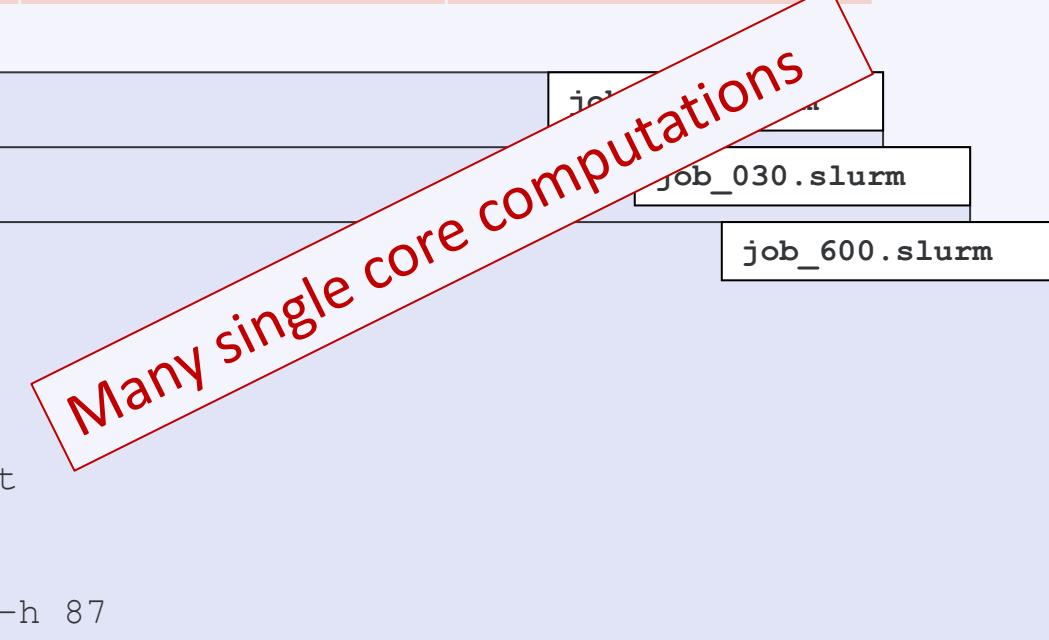
```
#!/bin/bash -l
#SBATCH --account=lp_myproject
#SBATCH --cluster=wice
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=1
#SBATCH --time=10:00
cd $SLURM_SUBMIT_DIR
weather -p 1.0e05 -t 293.0 -h 87
```

Many single core computations

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

job\_030.slurm

job\_600.slurm



# 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 --account=lp_hpcinfo

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

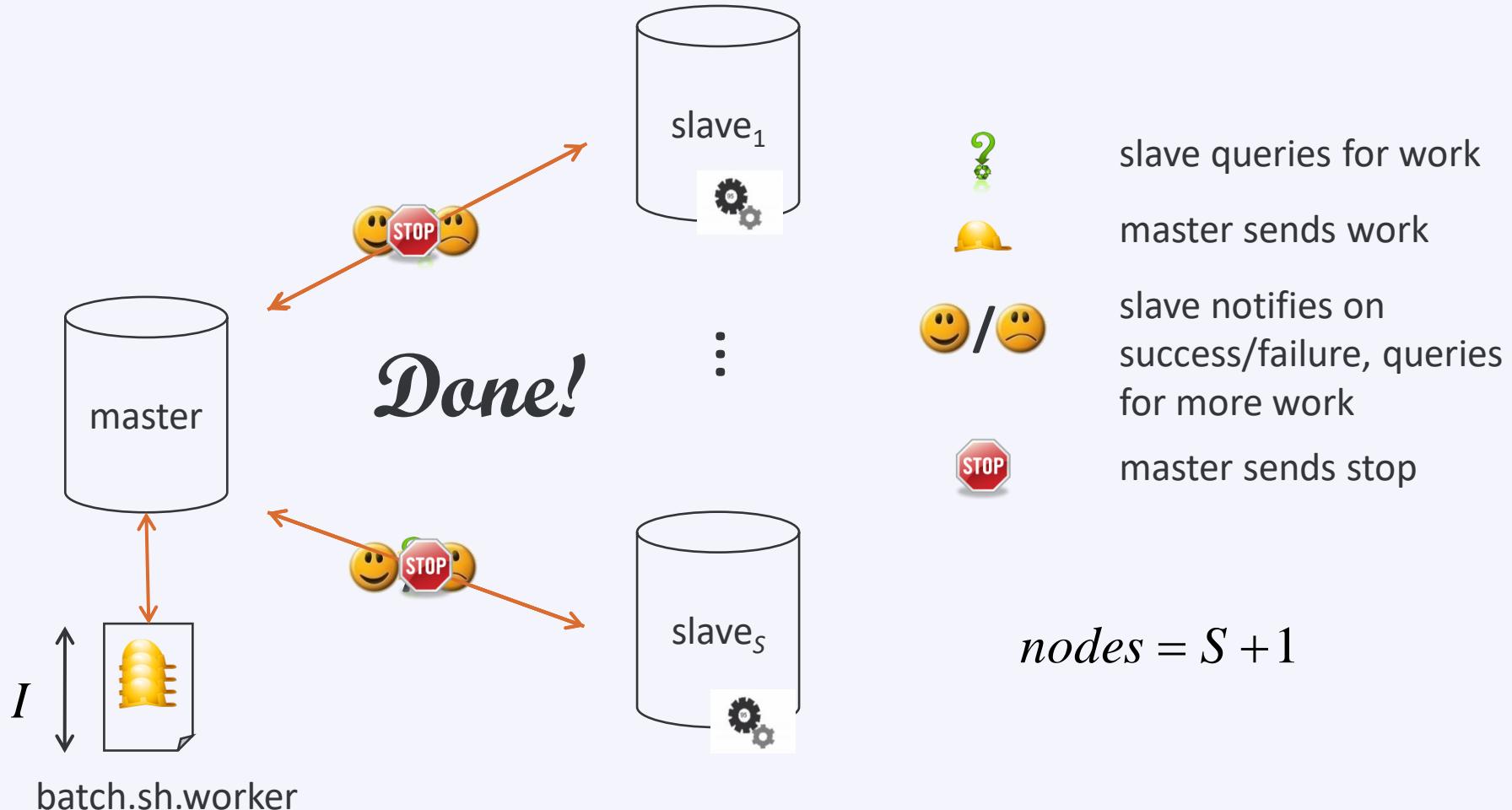
```
#From wICE interactive session:
$ module --force purge
$ module use /apps/leuven/rocky8/icelake/2021a/modules/all
$ module load worker/1.6.12-foss-2021a
$ wsub -data data.csv -batch job.slurm
```

# 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

$$walltime_{job} \geq \frac{N \cdot walltime_{work\ item}}{nodes \cdot ntasks-per-node}$$

# Worker processing: informally



# How to use worker well?

- ✓ Many work items, i.e.,  $\#work\ items/\#proc \gg 1$
- ✓ 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
- ✓ Here 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
- }  **$72 \times 72$  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
#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/rocky8/icelake/2021a/modules/all -----  
worker/1.6.12-foss-2021a
```

## From Genius Login Node

```
$ module av worker  
----- /apps/leuven/rocky8/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](https://github.com/gjbex/worker/tree/development_slurm/examples/bash_example)

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

letter	counter	alpha.csv
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/rocky8/skylake/2021a/modules/all
module load worker/1.6.12-foss-2021a-wice
wsub -batch alp.slurm -data alpha.csv
```

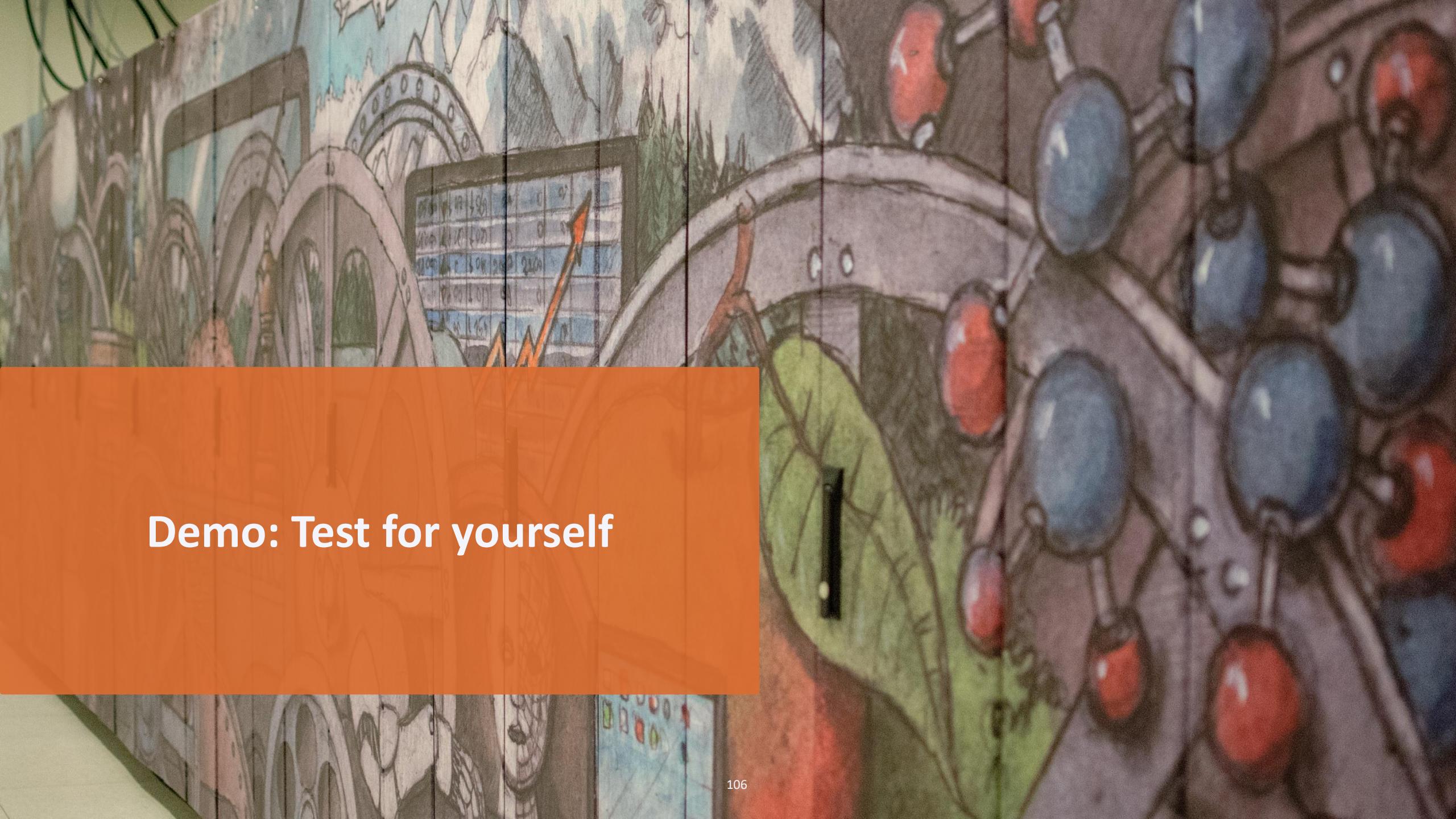
```
total number of work items: 26
Submitted batch job 60008899 on cluster wice
[Sat Jul 14 13:07:11 2023] --> [Mon Jul 17 10:44:00 2023]
```

# Submitting worker jobs

- ✓ From interactive job on wICE

```
#!/bin/bash -l
module --force purge
module use /apps/leuven/rocky8/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
```

The background of the slide features a vibrant, colorful illustration of a train station. On the left, a train with several green and brown cylindrical cargo containers is visible. In the center, there's a large digital screen displaying a grid of numbers and some orange arrows pointing upwards. The right side of the image shows a close-up of several circular objects, possibly train wheels or parts of a mechanical system, in shades of blue, red, and grey.

**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

# Questions

Helpdesk:

[hpcinfo@kuleuven.be](mailto:hpcinfo@kuleuven.be) or [https://admin.kuleuven.be/icts/HPCinfo\\_form/HPC-info-formulier](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*

