VSC HPC Introduction

ICTS KU Leuven
VSC staff:
Ehsan Moravveji
Mag Selwa
Geert Jan Bex (Uhasselt)
Jan Ooghe
Wouter van Assche

Questions and problems troubleshooting

- Blackboard Collaborate session: https://eu.bbcollab.com/guest/ce4229a2f9eb4cc5b90f490d660eb1c2

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?
- Genius Cluster
- Storage
- Login nodes & MFA
- Connection Setup
- Software environment
- How to submit jobs?
- Dedicated hardware
- How to choose resources?
- Optional material
  - Linux in brief
  - Conda for Python and R
  - Worker Framework

VSC

(Vlaams Supercomputer Centrum)

VSC PARTNERSHIP

Supported by

fwo

ASSOCIATIE UNIVERSITEIT GENT

universitaire associatie BRUSSEL

ASSOCIATIE KU LEUVEN

ASSOCIATIE UNIVERSITEIT-HOGESCHOLEN LIMBURG

auha antwerp university association

VSC HPC Environments

Tier-0

PRACE

Vlaanderen
is supercomputing

Tier-1

Hortense

Tier-2

GHENT UNIVERSITY — **STEVIN**

VUB VRIJE UNIVERSITEIT BRUSSEL — **HYDRA**

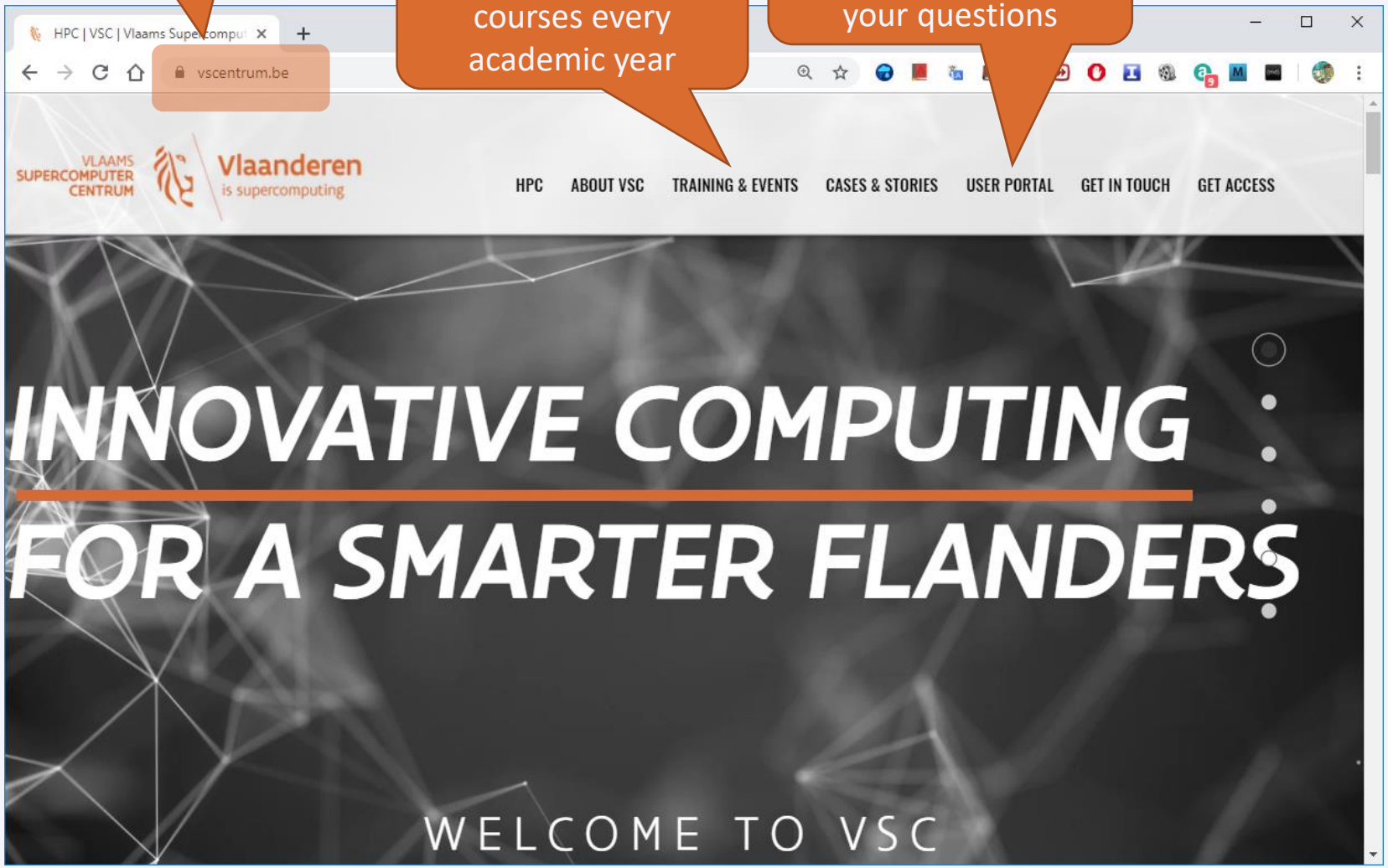
Universiteit Antwerpen — **HOPPER/LEIBNIZ**

KU LEUVEN ▸▸ UHASSELT — **GENIUS/WICE**

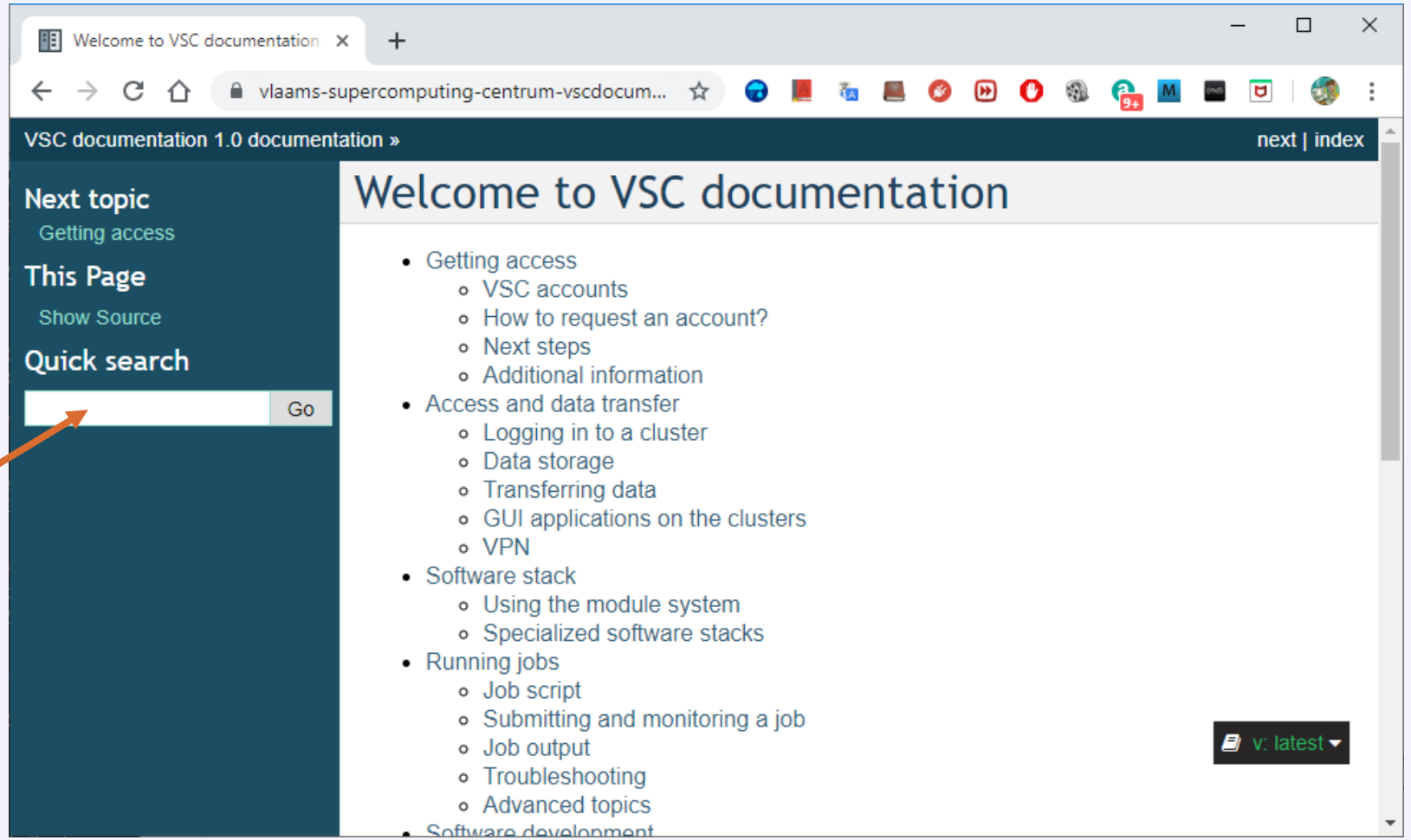
www.vscentrum.be

Very rich suite of courses every academic year

Documentation! Answers >80% of your questions



VLAAMS SUPERCOMPUTER CENTRUM

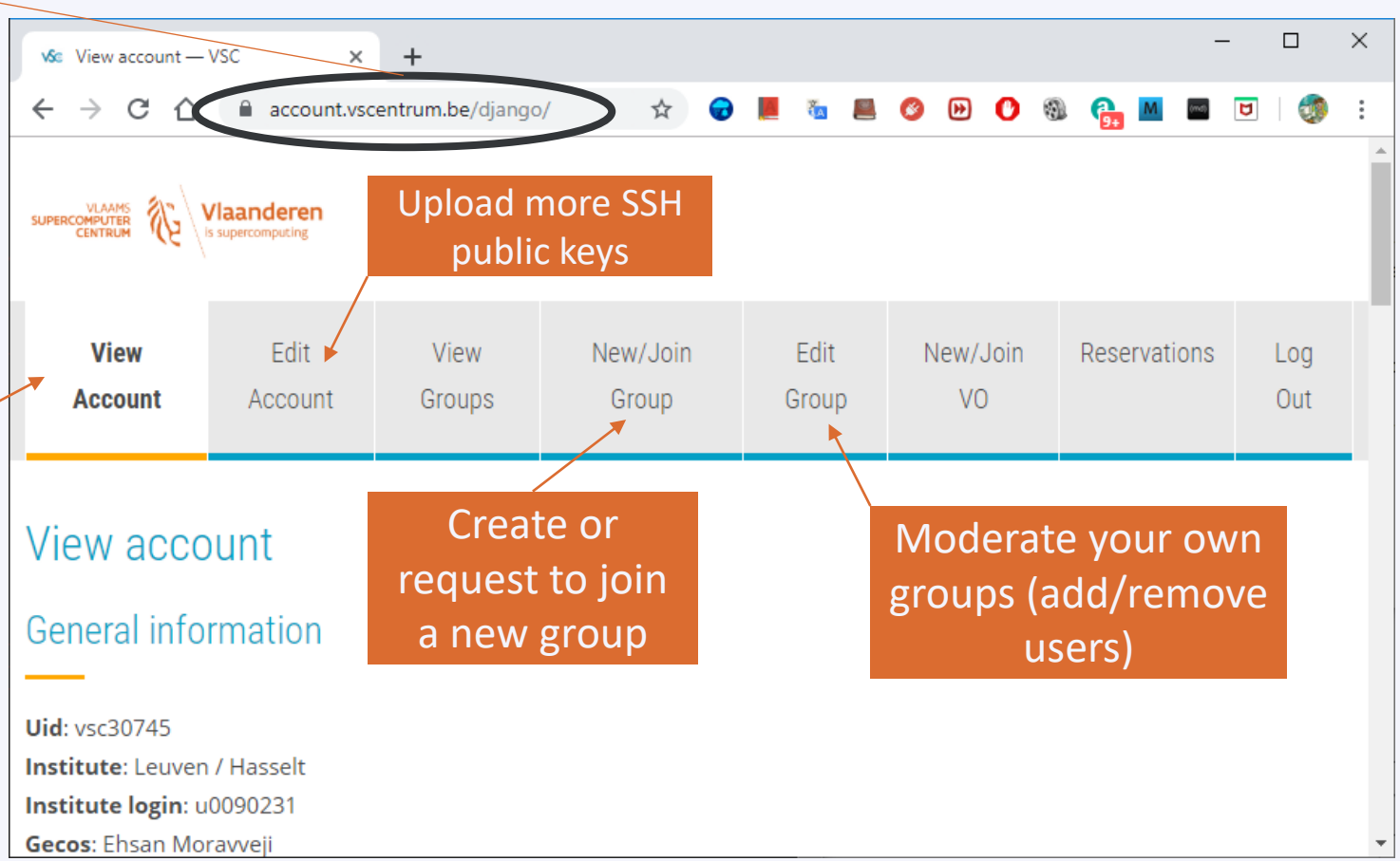


Search you keywords here; e.g. qsub

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

To manage your VSC account: account.vscentrum.be



Upload SSH Key
Request account

Upload more SSH public keys

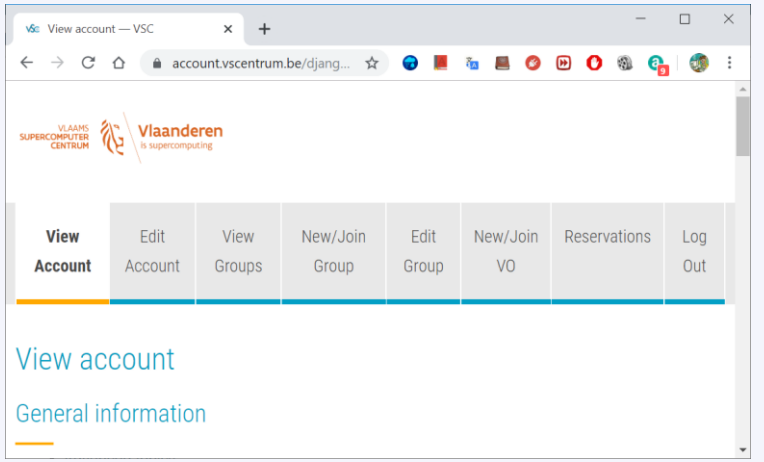
Create or request to join a new group

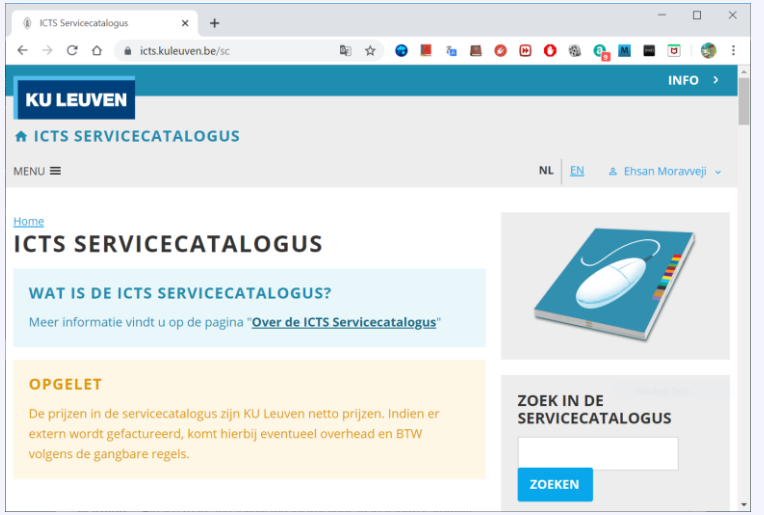
Moderate your own groups (add/remove users)

VLAAMS SUPERCOMPUTER CENTRUM

Become a VSC user

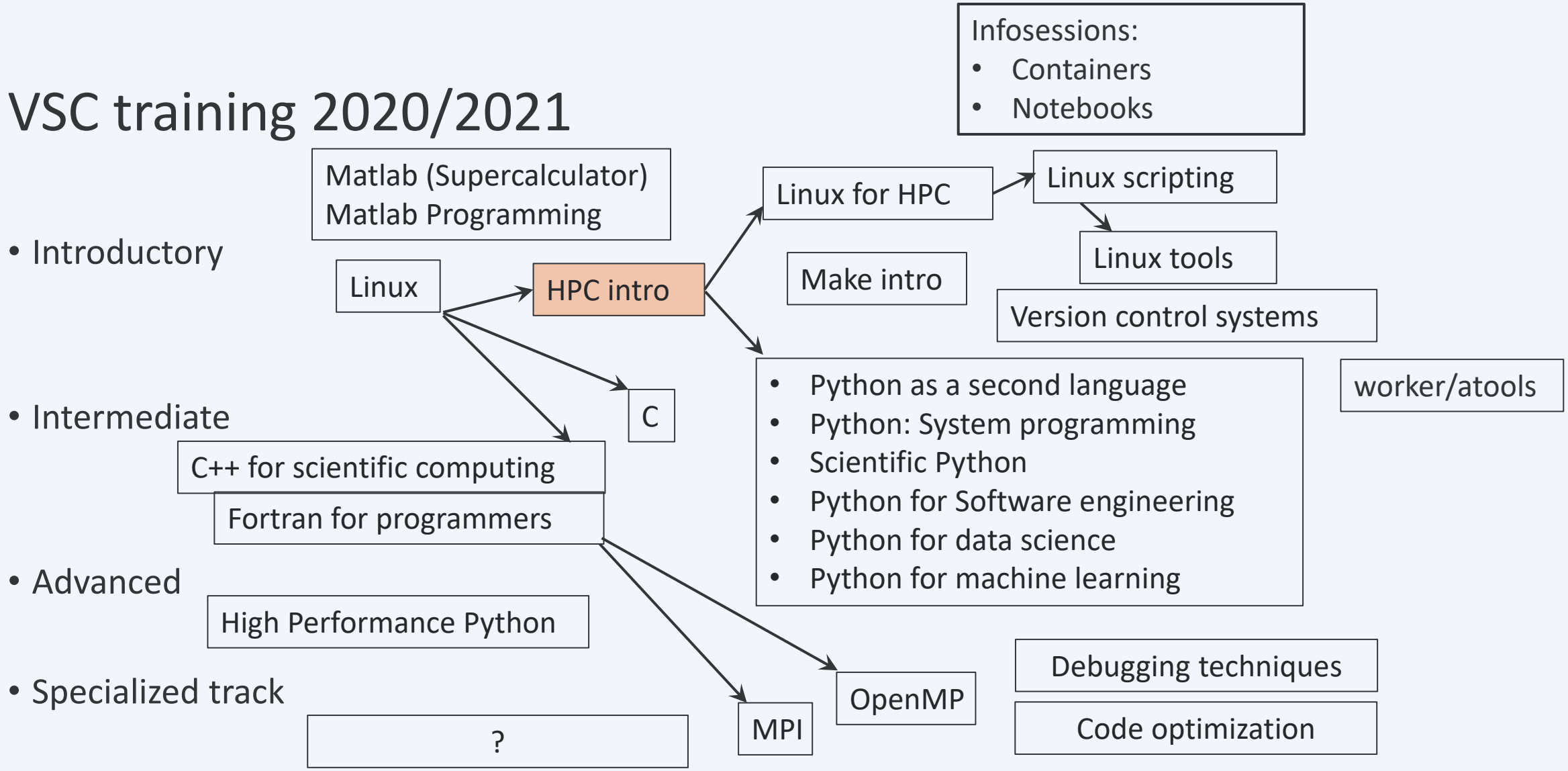
- ❑ Create a secure (4096 bit) SSH key pairs
  Upload it on the account page: www.account.vscentrum.be
- ❑ You need to request a VSC account
  Normally processed swiftly
- ❑ Request introductory credits (2M free credits for 6 months)
- ❑ Request project credits (for supervisors and project leaders)
  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)





VLAAMS SUPERCOMPUTER CENTRUM

VSC training 2020/2021



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

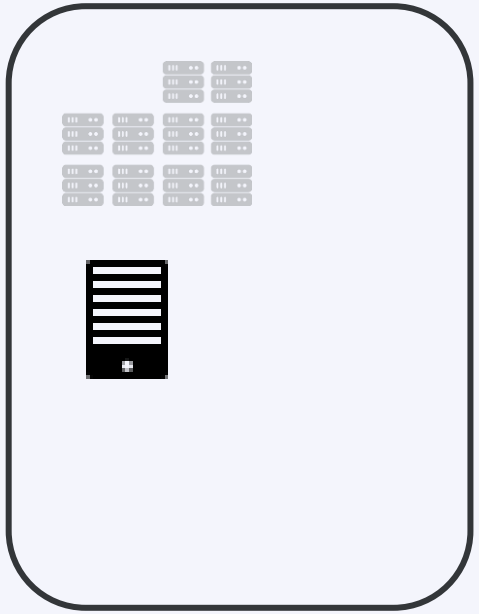
wICE (since 9/2022)
186 nodes; 13,392 cores

Genius Overview

Compute nodes

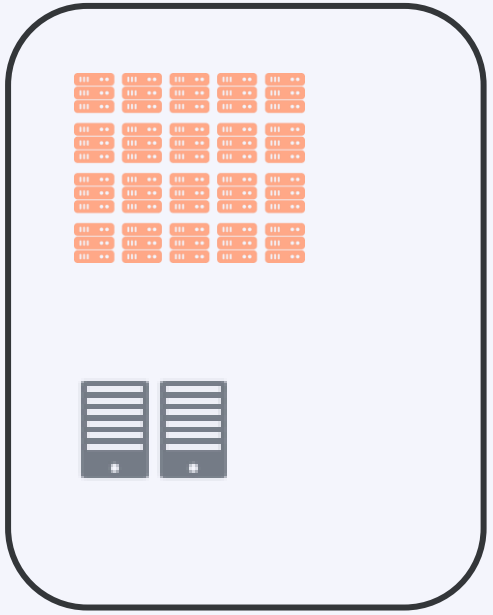
Thin nodes
+ 96x Skylake 36c 192 GB
+ 144x CascadeLake 36c 192 GB

Large memory nodes



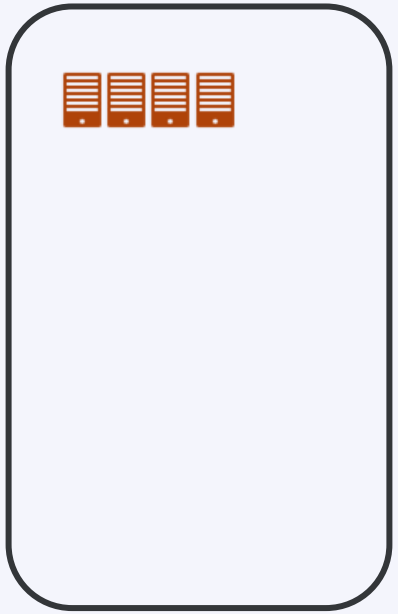
BigMem
+ 10x Skylake 36c 768GB
+ 1x Superdome 112c 6 TB

GPU nodes



GPUs
+ 20x Skylake 36c 192 GB
  4x P100 16GB
+ 2x CascadeLake 36c 768GB
  8x V100 32GB

Exp nodes



AMD
4x AMD Naples 64c 256 GB

Tier-2 Cluster: Genius

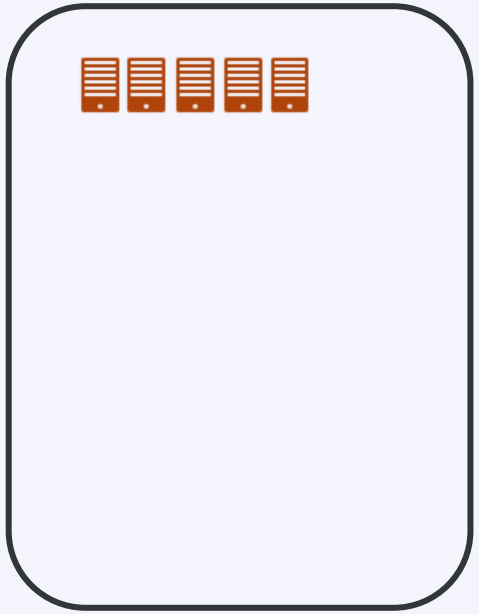
Type of node	CPU type	Inter-connect	# cores	installed mem	local discs	# nodes
SkyLake	Xeon 6140	IB-EDR	**36**	**192 GB**	800 GB	**86**
SkyLake **large mem**	Xeon 6140	IB-EDR	**36**	**768 GB**	800 GB	**10**
SkyLake **GPU**	Xeon 6140 **4xP100** SXM2	IB-EDR	**36**	**192 GB**	800 GB	**20**
CascadeLake	Gold 6240	IB-EDR	**36**	**192 GB**	800 GB	**144**
CascadeLake **GPU**	Gold 6240 **8xV100** SMX2	IB-EDR	**36**	**768 GB**	800 GB	**2**
SkyLake **Superdome**	Gold 6132	Flex Grid	**14**	**6 TB**	6 TB	**8**

wICE Overview

Compute nodes

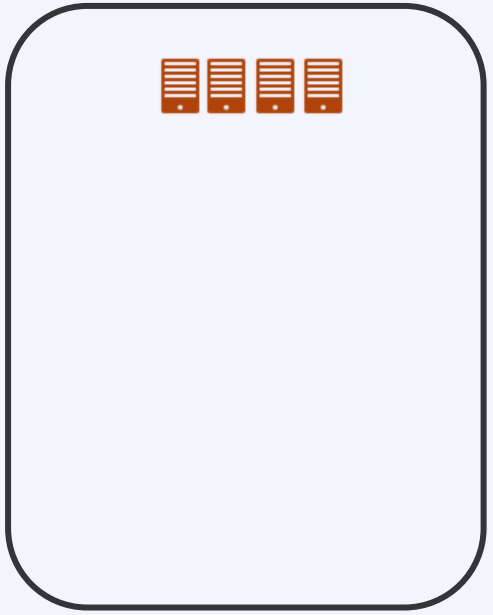
Thin nodes
+ 172x Icelake 72c 256 GB

Large memory nodes



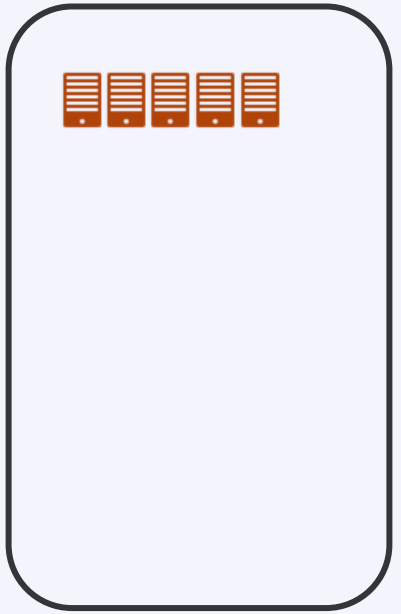
BigMem
+ 5x Icelake 72c 2 TB

GPU nodes



GPUs
+ 4x Icelake 72c 512 GB
  4x A100 80 GB

Interactive nodes



+ 5x Icelake 64c 512 GB
  1x A100 80 GB

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**

Technical Hardware Specifications

	Tier 2					
Cluster name	Genius			wICE		
Processor type	SkyLake		Cascade Lake	Ice Lake		
Cores per node	36		36	72		
Base Clock Speed	2.3 GHz		2.6 GHz	2.6 GHz		
Total nodes	86	10	144	172	5	5 + 4
Node memory (GB)	192	768	192	256	2048	512
Memory per core (GB)	5.2	21.2	5.3	3.4	25	6.8
Total cores	3,456		4,320	13,392		
Peak performance (Flops/cycle)	16 DP FLOPs/cycle: 8-wide FMA (fused multiply-add) instructions AVX-512		16 DP FLOPs/cycle: 8-wide FMA (fused multiply-add) instructions AVX-512			
Network	Infiniband EDR		Infiniband EDR	Infiniband HDR-100		
Cache (L1 KB/L2 KB/L3 MB)	18x(32i+32d) / 18x1024 / 25 MB		18x(32i+32d) / 18x1024 / 25 MB	72x(48i+32d) / 72x 1280K / 48 MB		

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)

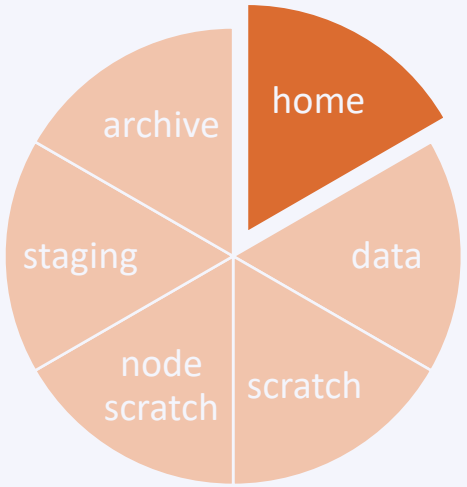
Storage

- 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

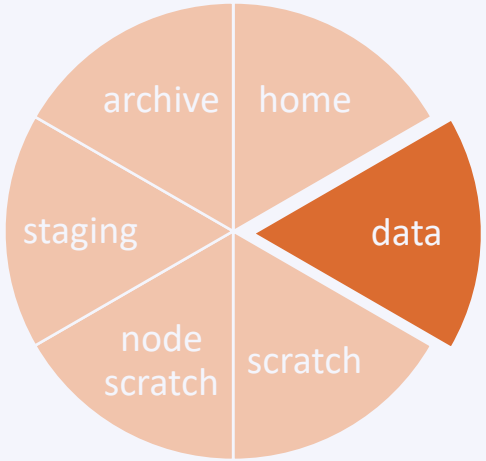
```
$ myquota
file system $VSC_HOME
    Blocks: 1479M of 3072M
    Files: 12934 of 100k
file system $VSC_DATA
    Blocks: 12G of 75G
    Files: 1043k of 10000k
file system $VSC_SCRATCH
    Blocks: 15M of 1.5T
```

Storage



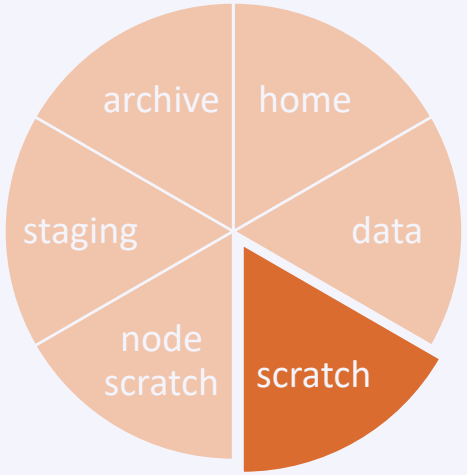
Storage	home folder
Env. Variable	`$VSC_HOME`
Filesystem Type	NFS
Access	Global
Backup	Hourly, daily, weekly (until last month) inside the `.snapshot` folder.
Default Quota	3 GB
Extension	Not possible
Usage	Only storing SSH keys, config files
Remarks	- Stay away from using it - Can easily overflow: + Your jobs may crash + Login issues

Storage



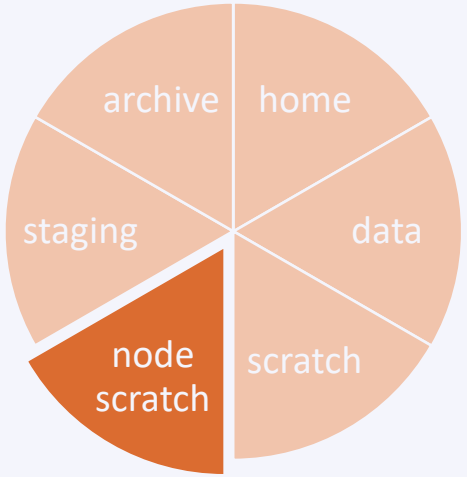
Storage	data folder
Env. Variable	`$VSC_DATA`
Filesystem Type	NFS
Access	Global
Backup	Hourly, daily, weekly (until last month) inside the `.snapshot` folder.
Default Quota	75 GB
Extension	On purchase
Usage	Your data, code, software, results
Remarks	- Permanent storage for initial/final results - Not optimal for intensive or parallel I/O

Storage



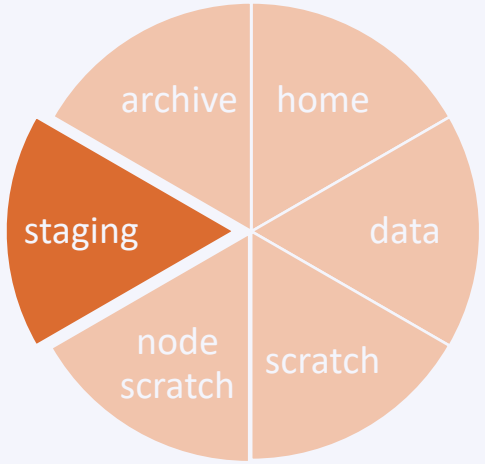
Storage	scratch folder
Env. Variable	`$VSC_SCRATCH`
Filesystem Type	Lustre
Access	Global
Backup	delete after 30 days from last access
Default Quota	500 GB
Extension	Free
Usage	Intensive, parallel I/O, temporary files
Remarks	- Recommended storage for all jobs - Copy scratch files to VSC_DATA or local 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	- Fastest I/O, attached to the node - Is cleaned after job terminates - Copy the data to your home, scratch, or staging before job ends

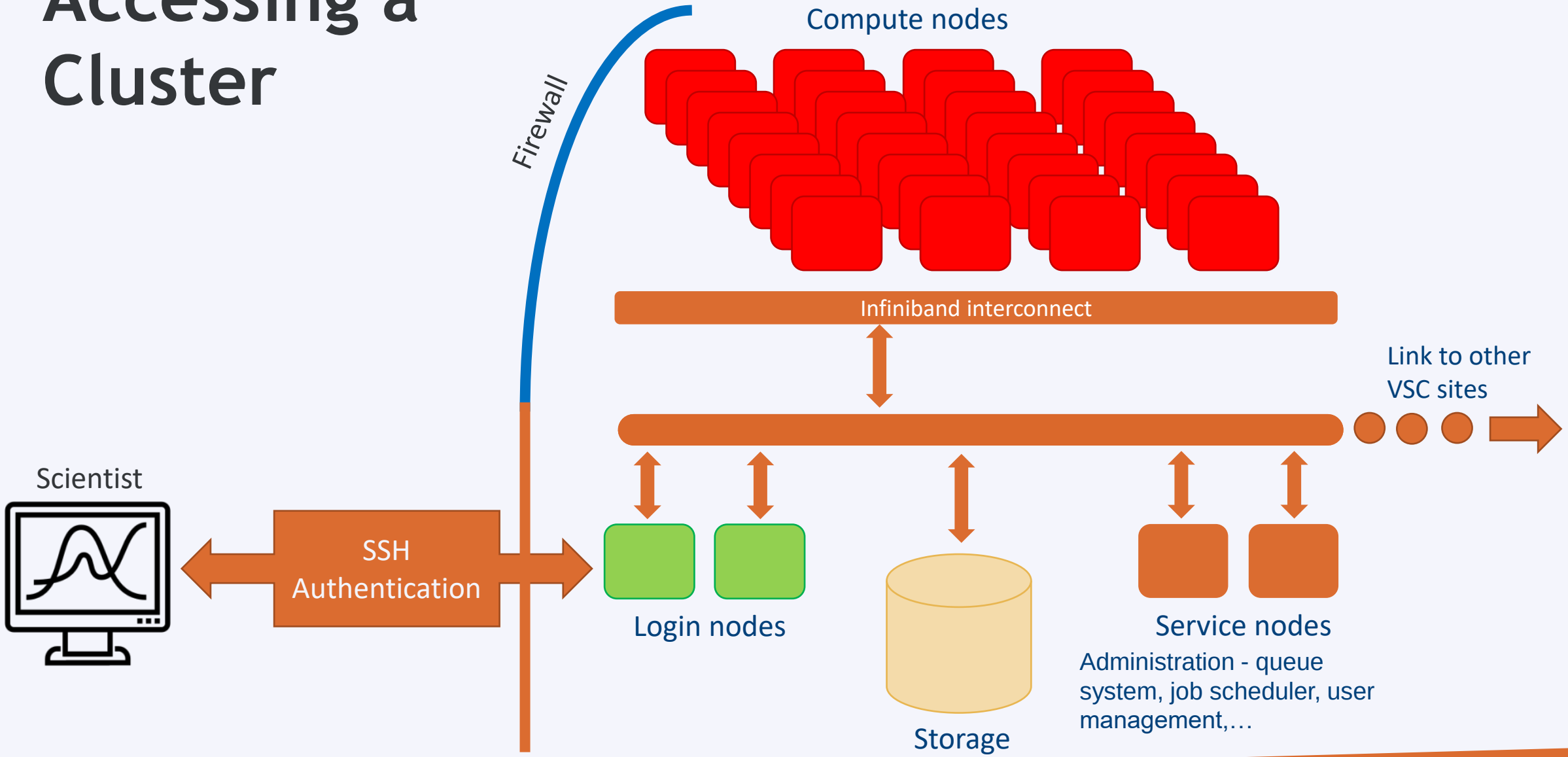
Storage



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	- Accessible from login/compute nodes - Fast, parallel I/O

Login Nodes & MFA

Accessing a Cluster



VLAAMS SUPERCOMPUTER CENTRUM

Using Login Nodes

Tips

- ❑ To develop and/or compile code and/or software
- ❑ 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
  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

Warning

- ❑ Login nodes are shared resources
- ❑ **Do not** execute heavy-lifting tasks (core, memory)
- ❑ Instead, submit jobs

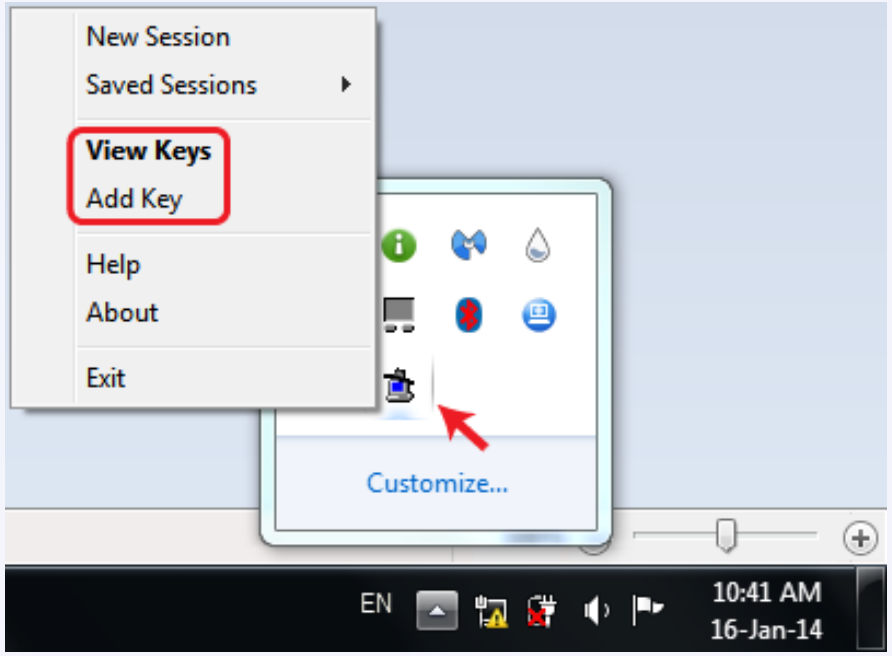
Login Setup

- ❑ PuTTY
- ❑ Terminal

Activate Your SSH Agent

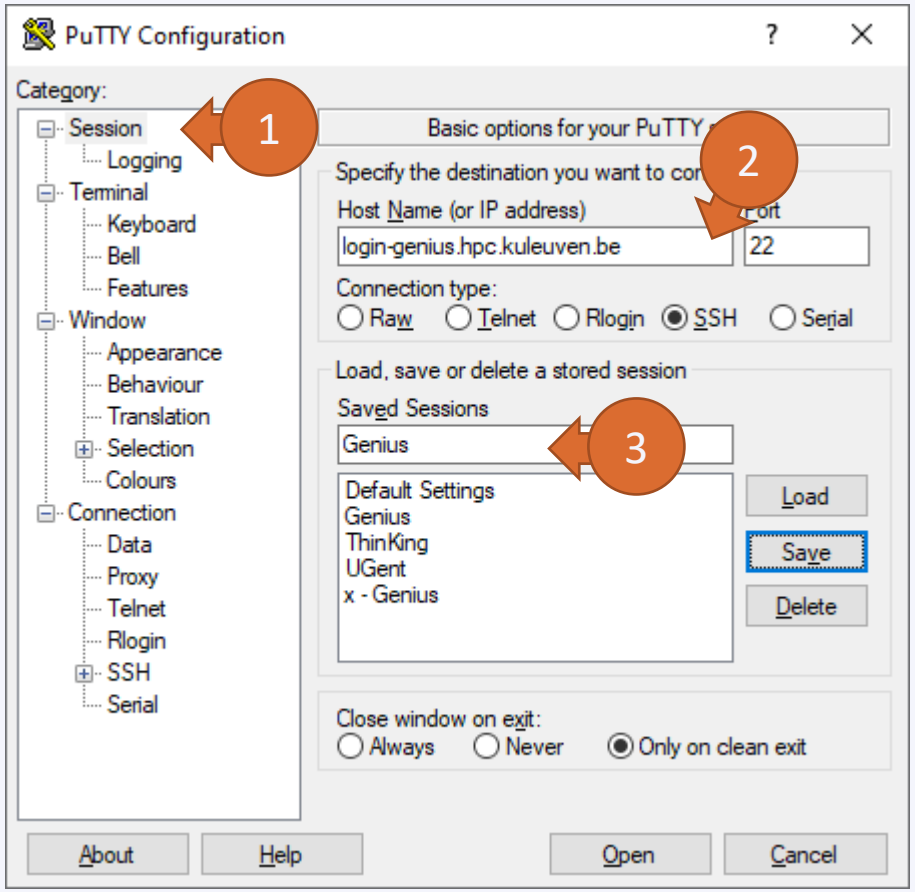
(Windows only)

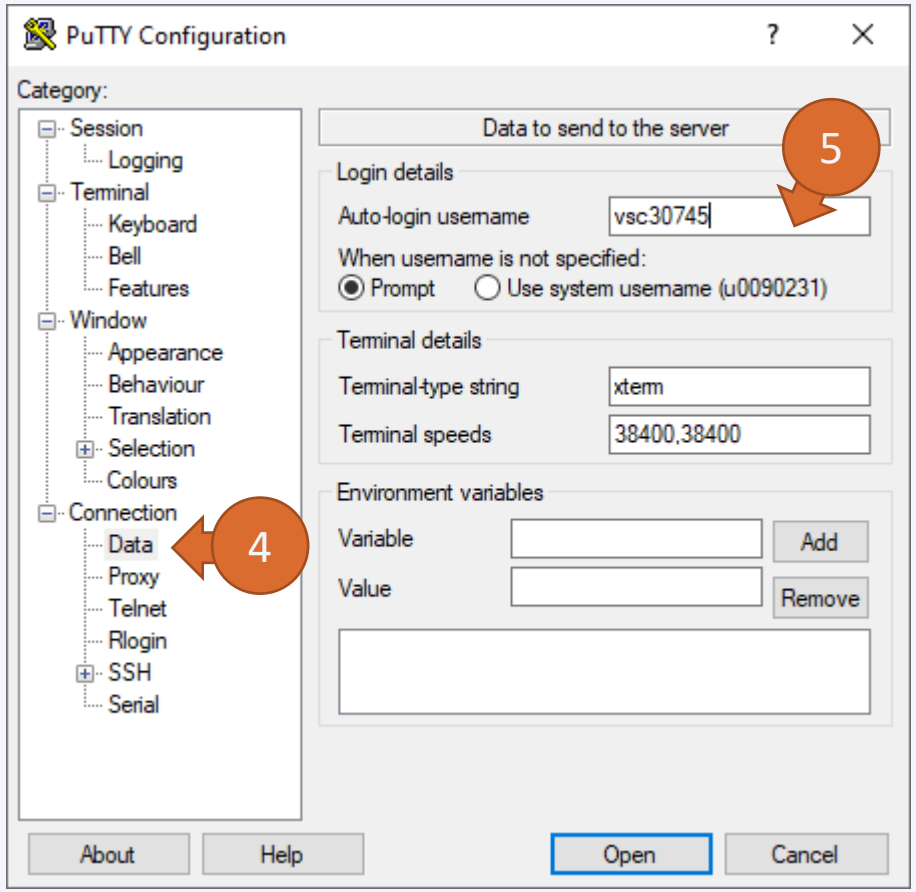
- When you install PuTTY, the agent called **Pageant** is automatically installed
- Open Pageant; it hides inside your bottom-right tray
- Click on “Add Key” and brows to your private key(s) folder
- After choosing a key, you are asked for a passphrase
- If successful, agent always remembers your SSH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



Setup PuTTY in 12 Clicks

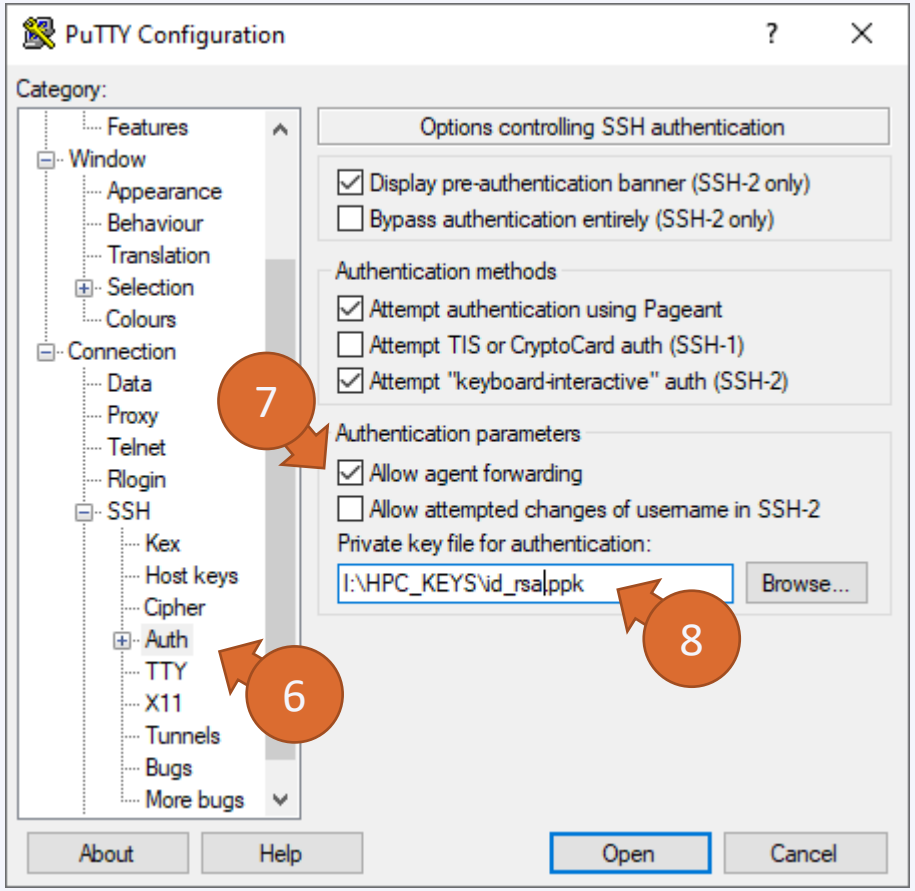
(Windows only)

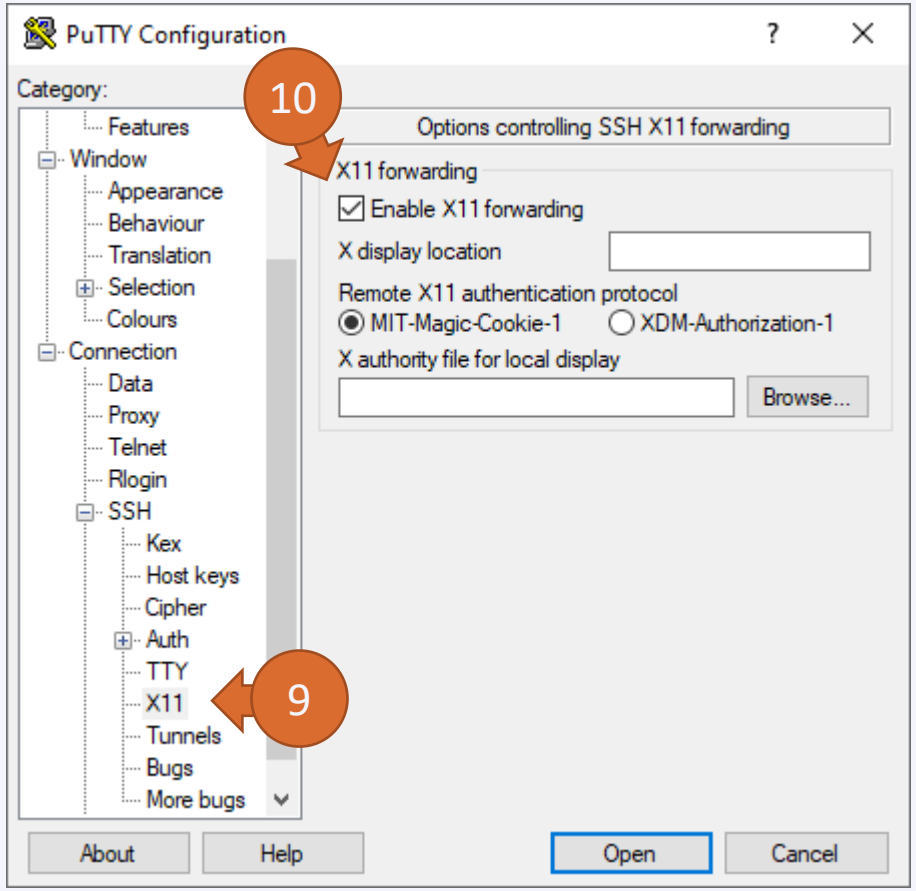




Setup PuTTY in 12 Clicks

(Windows only)

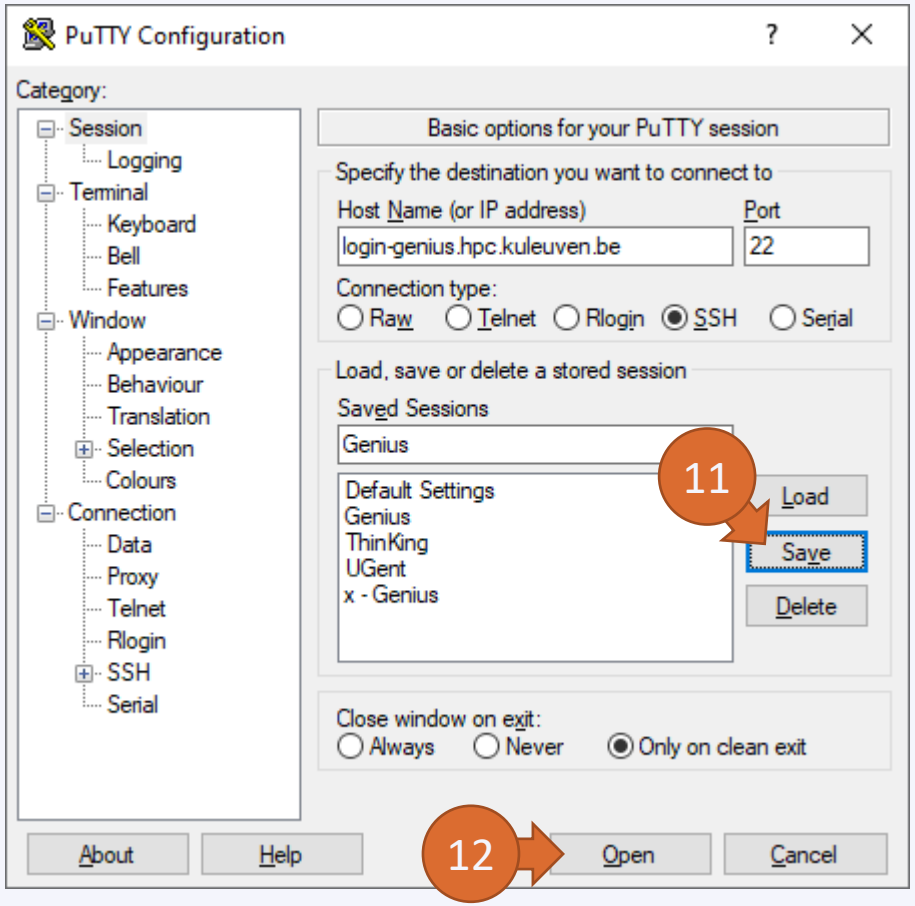


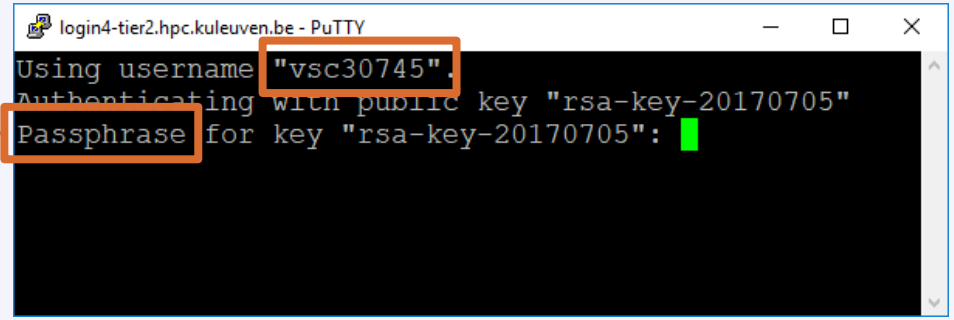


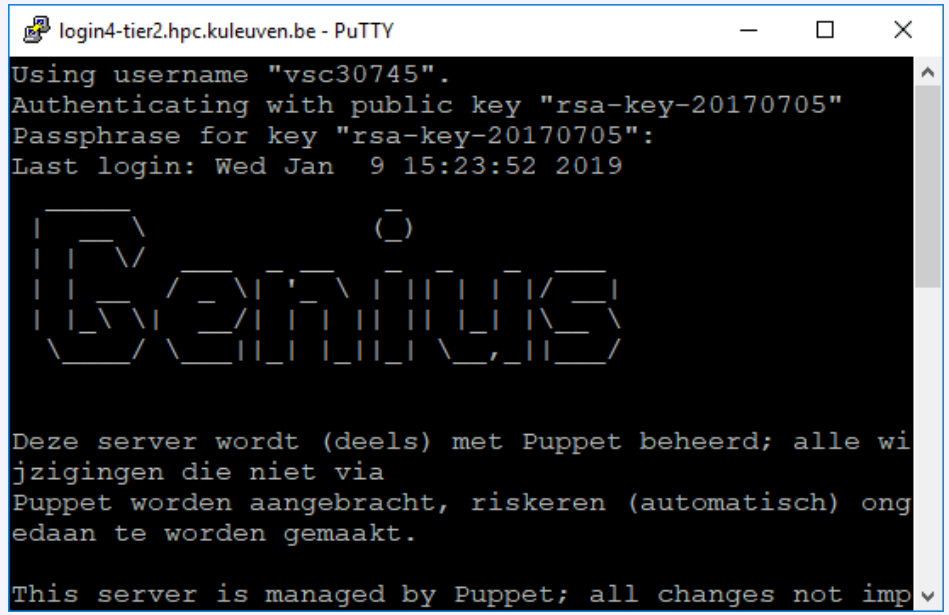
Setup PuTTY in 12 Clicks

(Windows only)

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







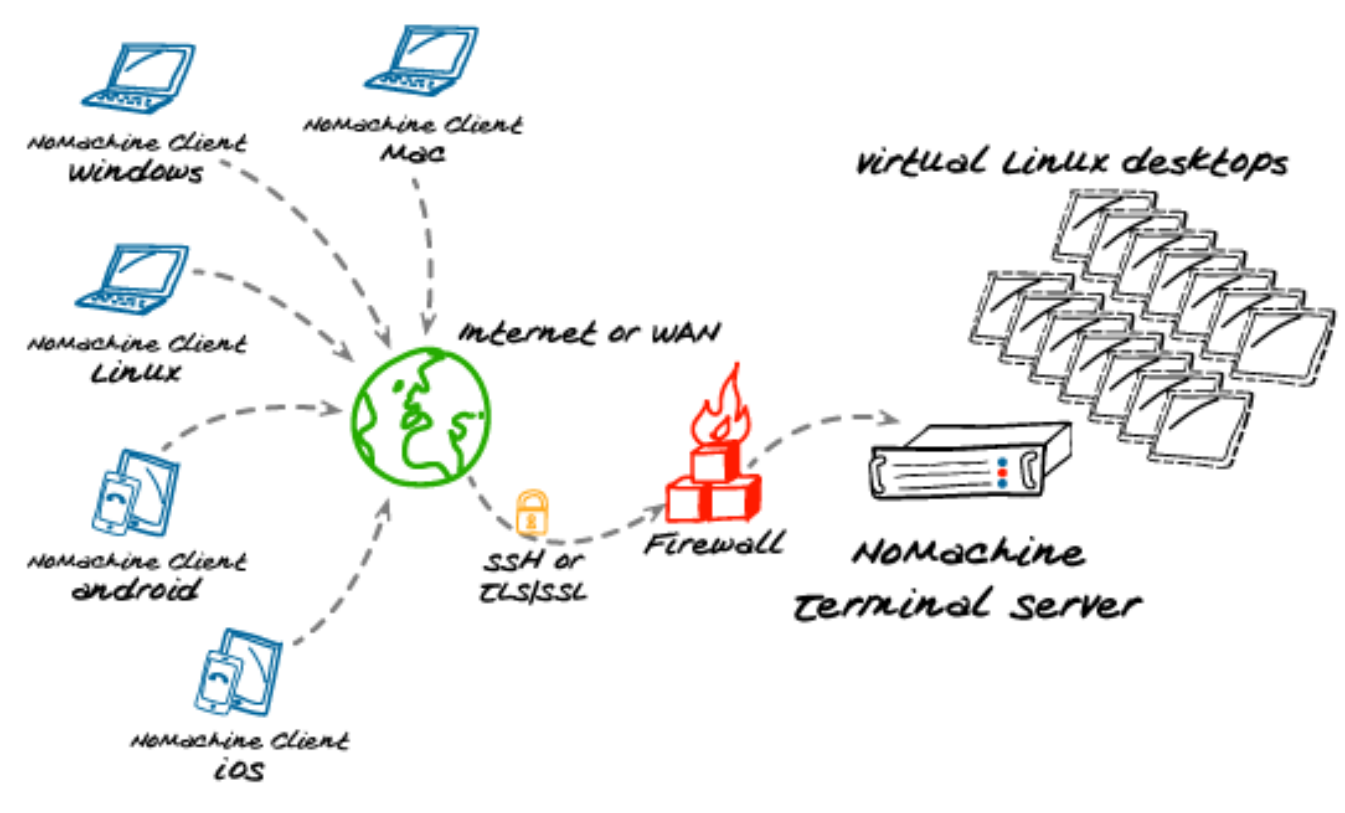
NX

- ❑ 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

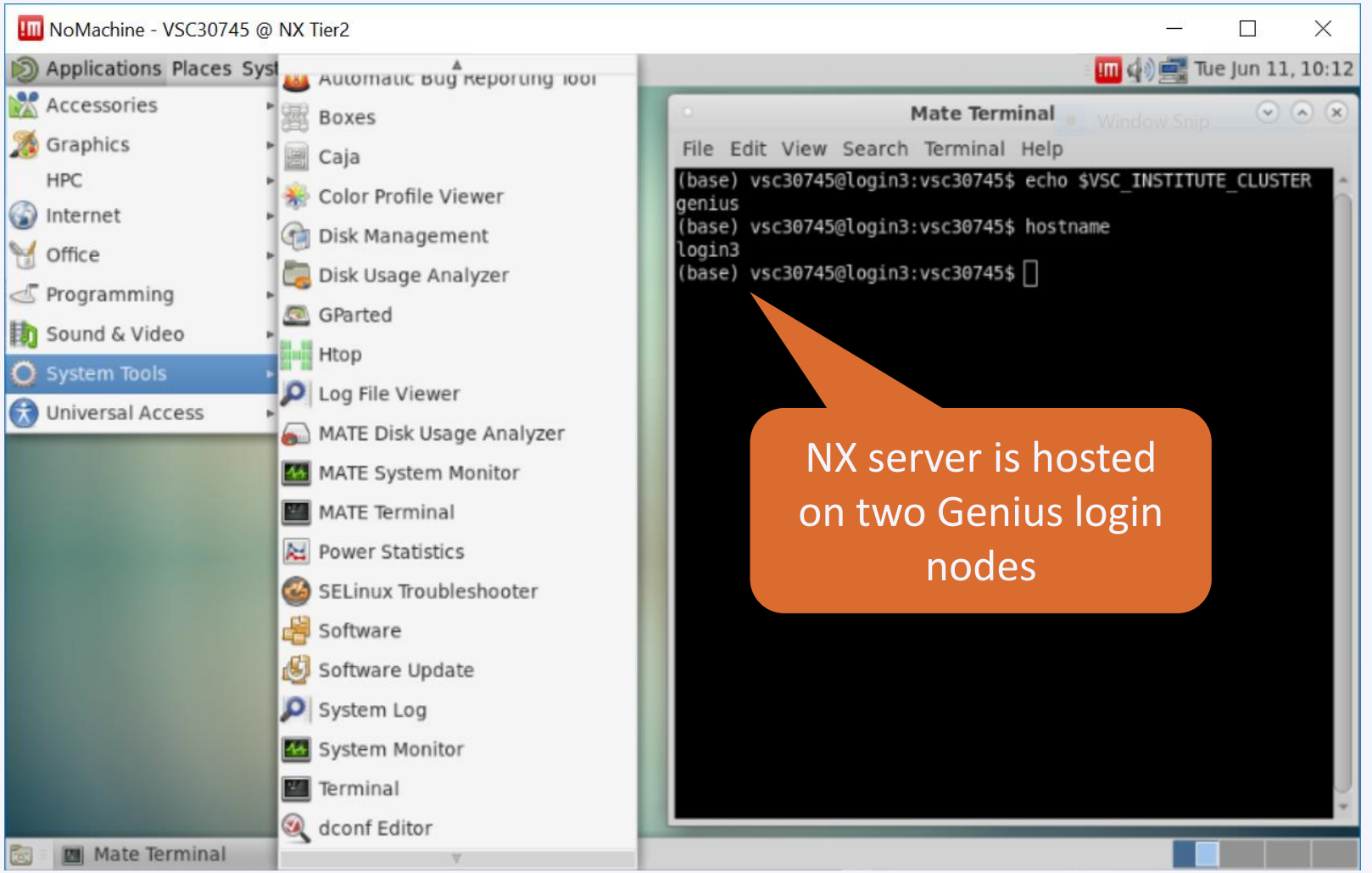


VLAAMS SUPERCOMPUTER CENTRUM

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

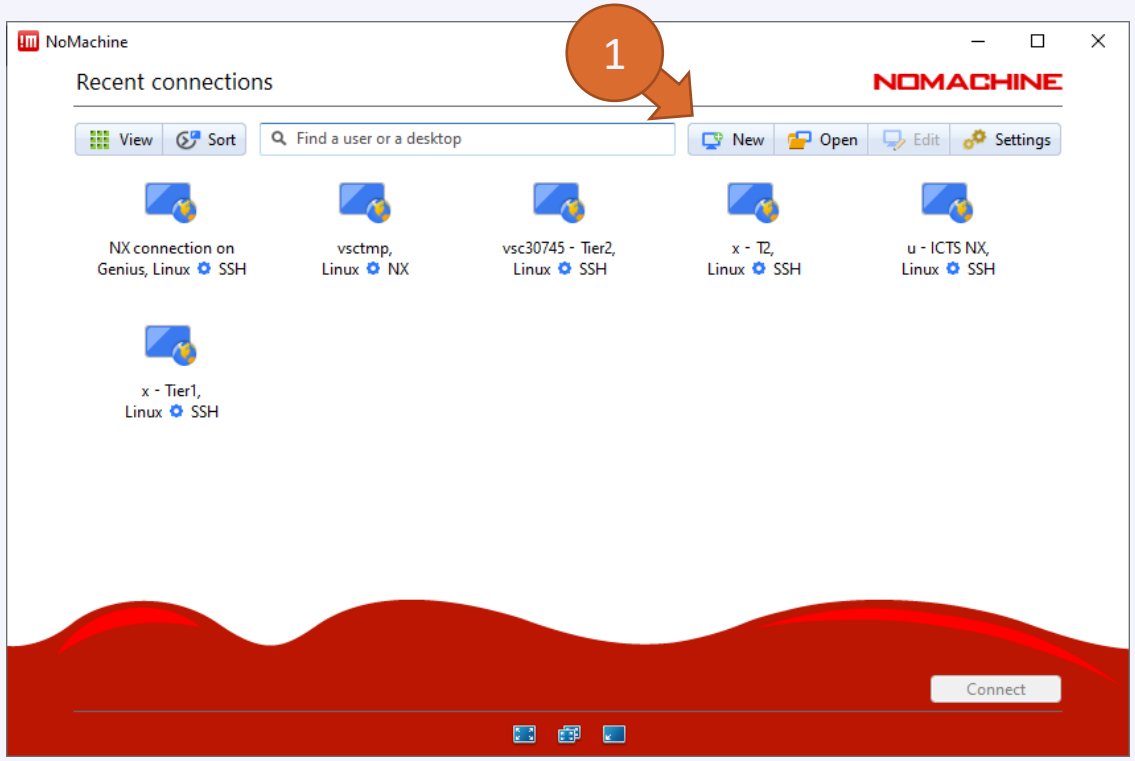
NX virtual desktop

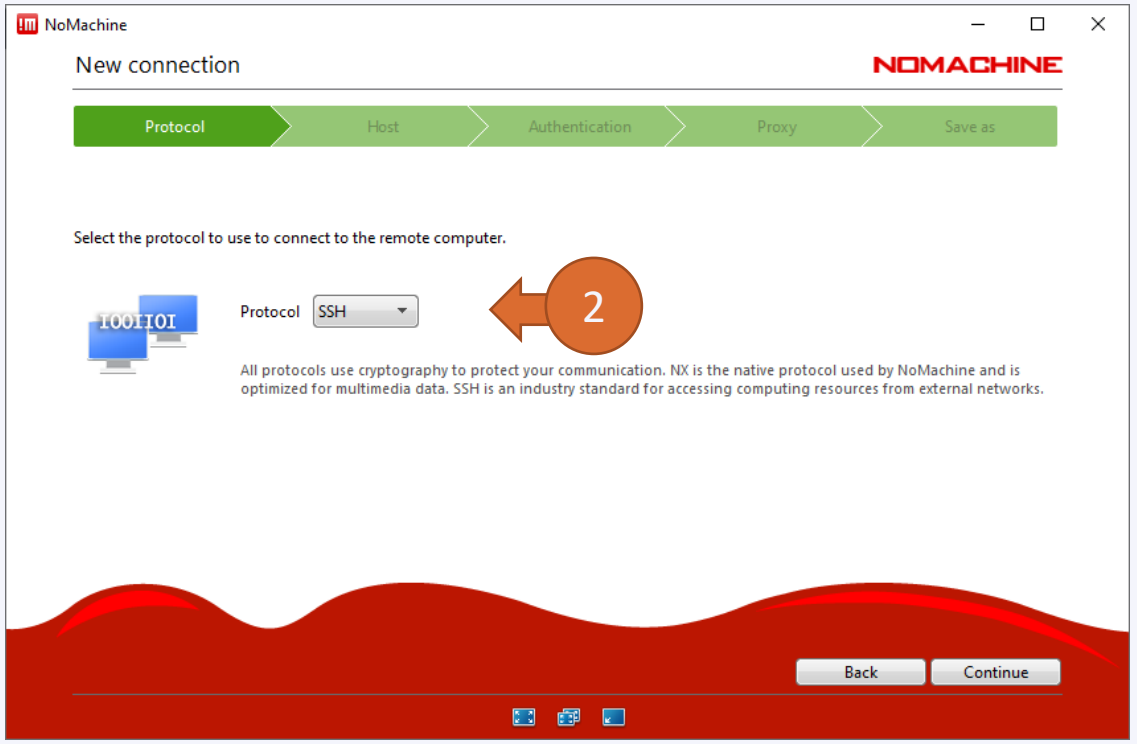


NX: available software

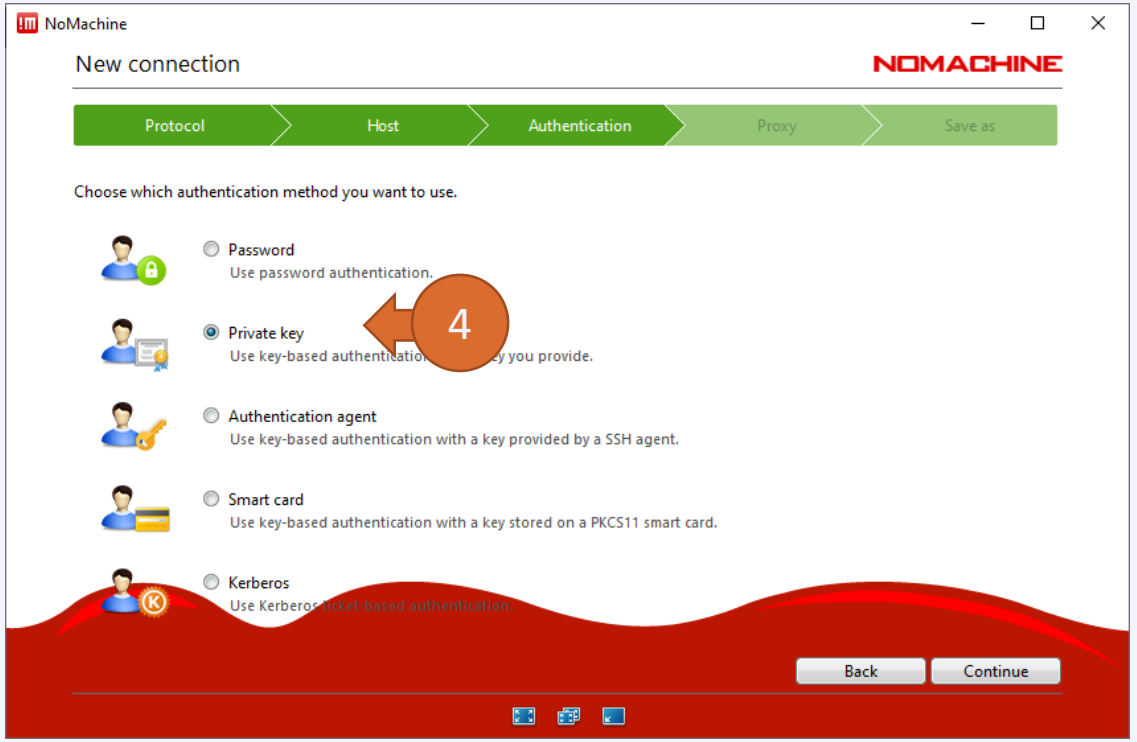
- **Accesories**: Gedit, Vi IMproved, Emacs (dummy version), Calculator
- **Graphics**: gThumb (picture viewer), Xpdf Viewer
- **Internet**: Firefox
- **HPC**: **Computation**: Matlab (2018a), RStudio, SAS
- **Visualisation**: Paraview, VisIt, VMD
- **Programming**: Meld Diff Viewer (visual diff and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

Setup NX in 10 Steps

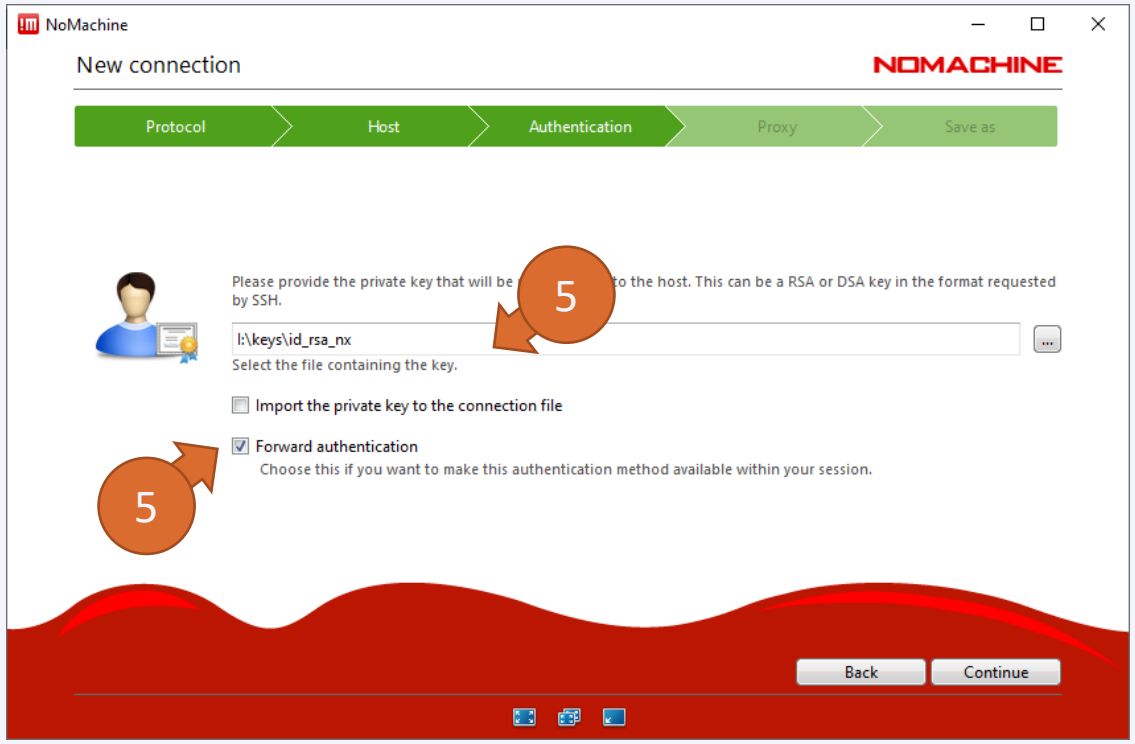


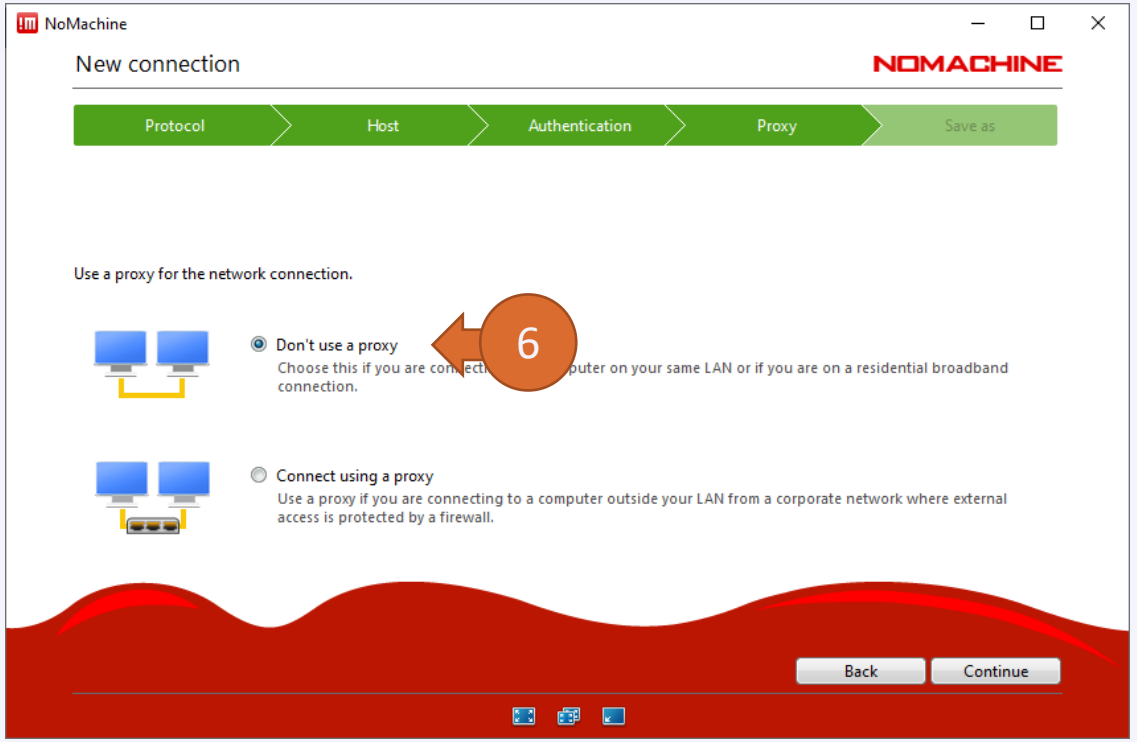


Setup NX in 10 Steps

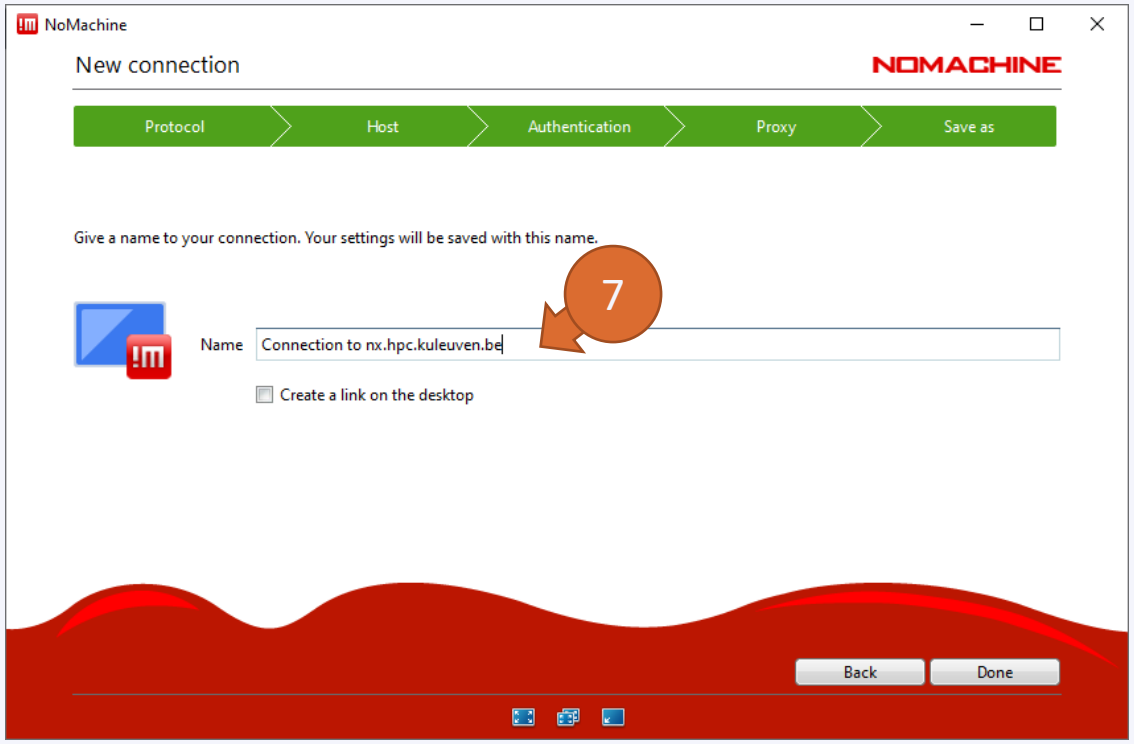


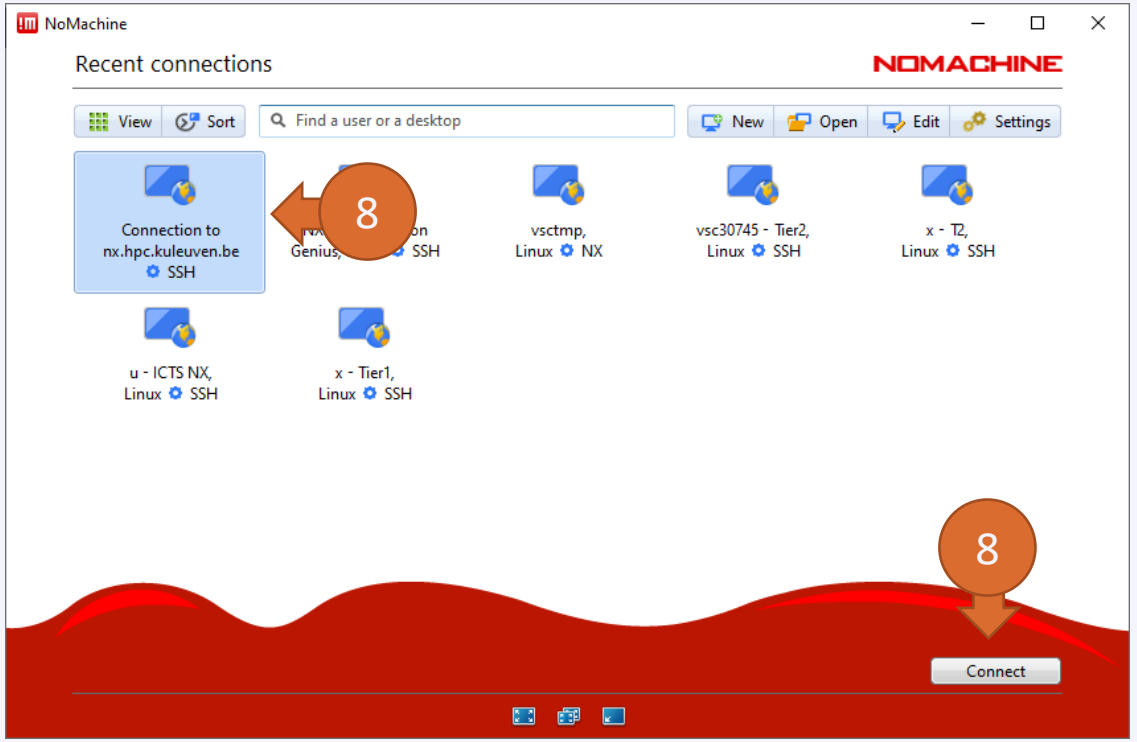
Setup NX in 10 Steps



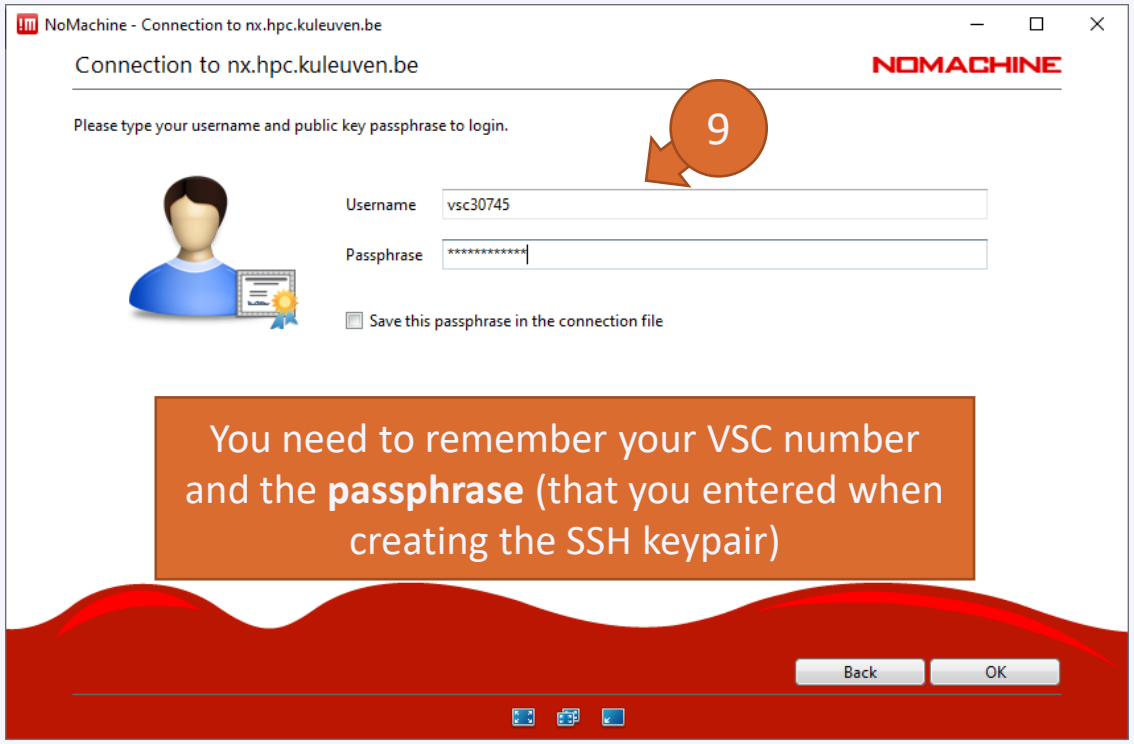


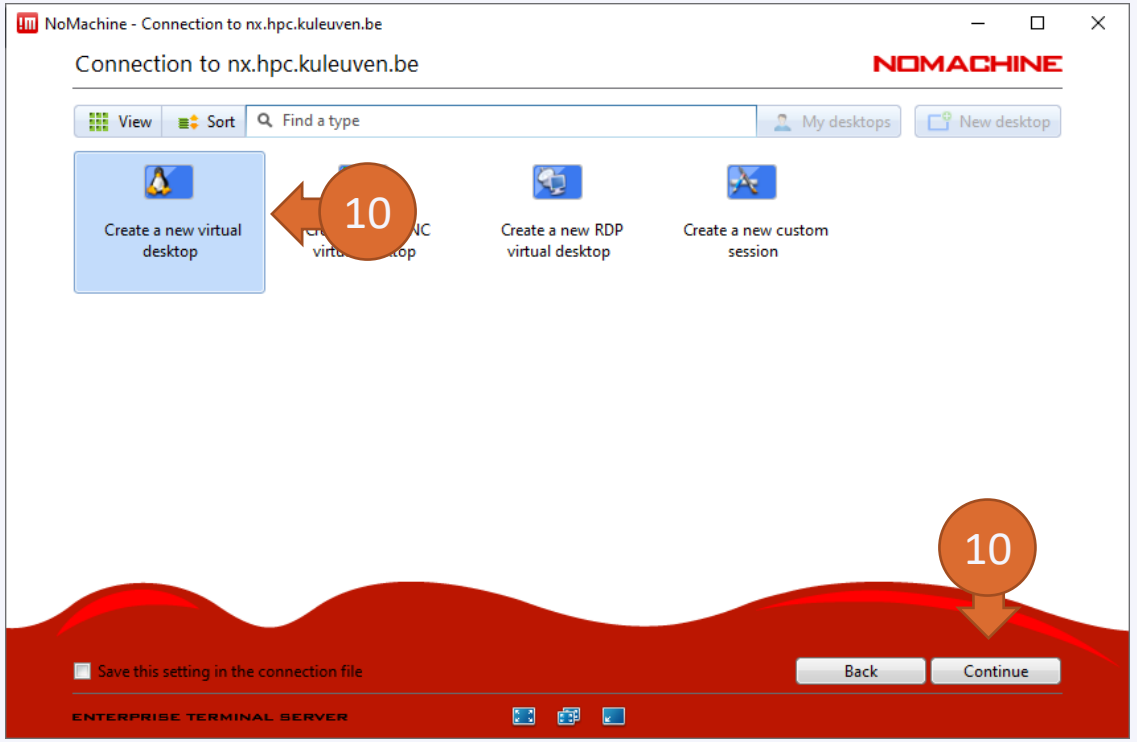
Setup NX in 10 Steps

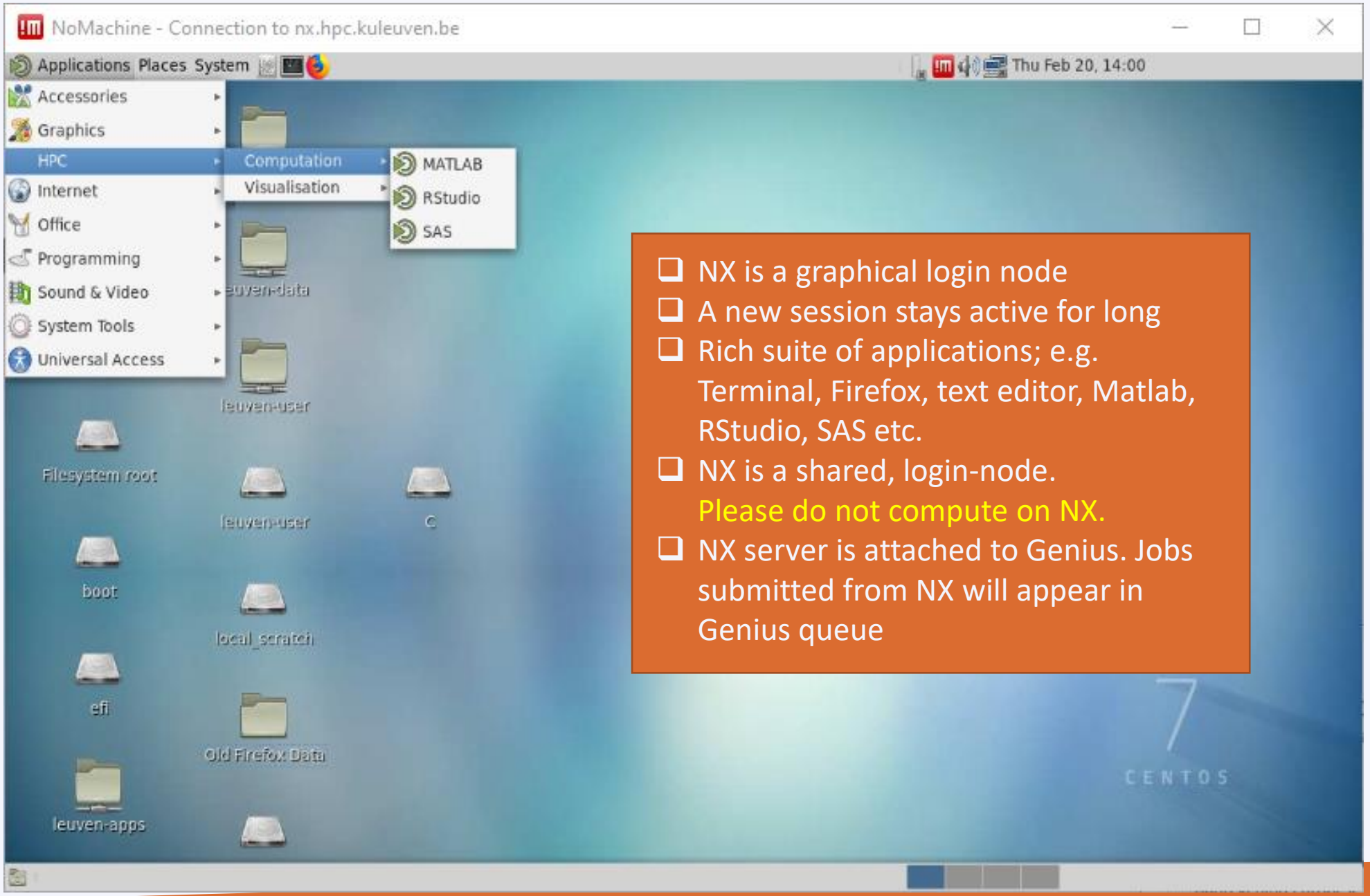




Setup NX in 10 Steps



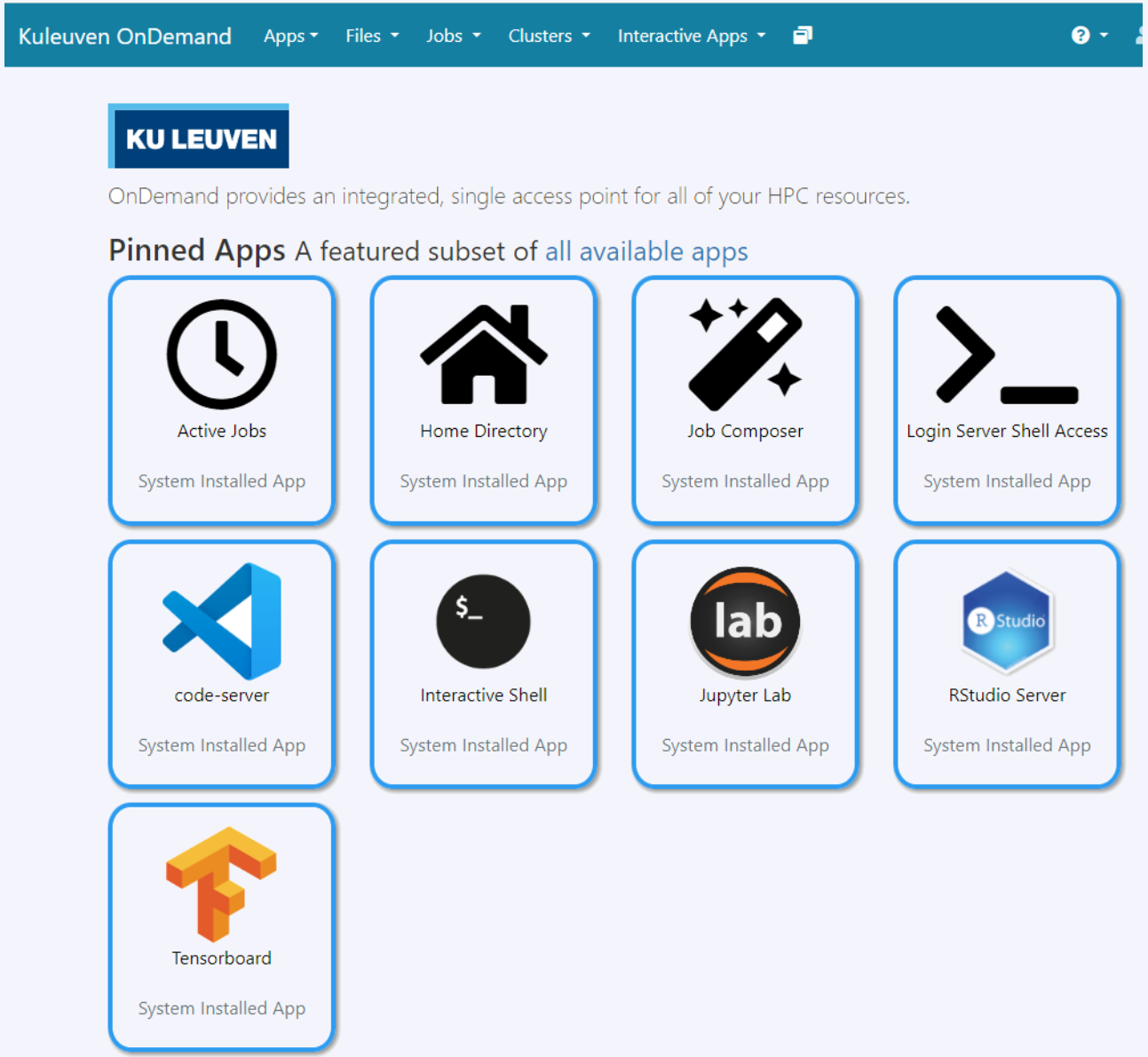




- NX is a graphical login node
- A new session stays active for long
- Rich suite of applications; e.g. Terminal, Firefox, text editor, Matlab, RStudio, SAS etc.
- NX is a shared, login-node. Please do not compute on NX.
- NX server is attached to Genius. Jobs submitted from NX will appear in Genius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

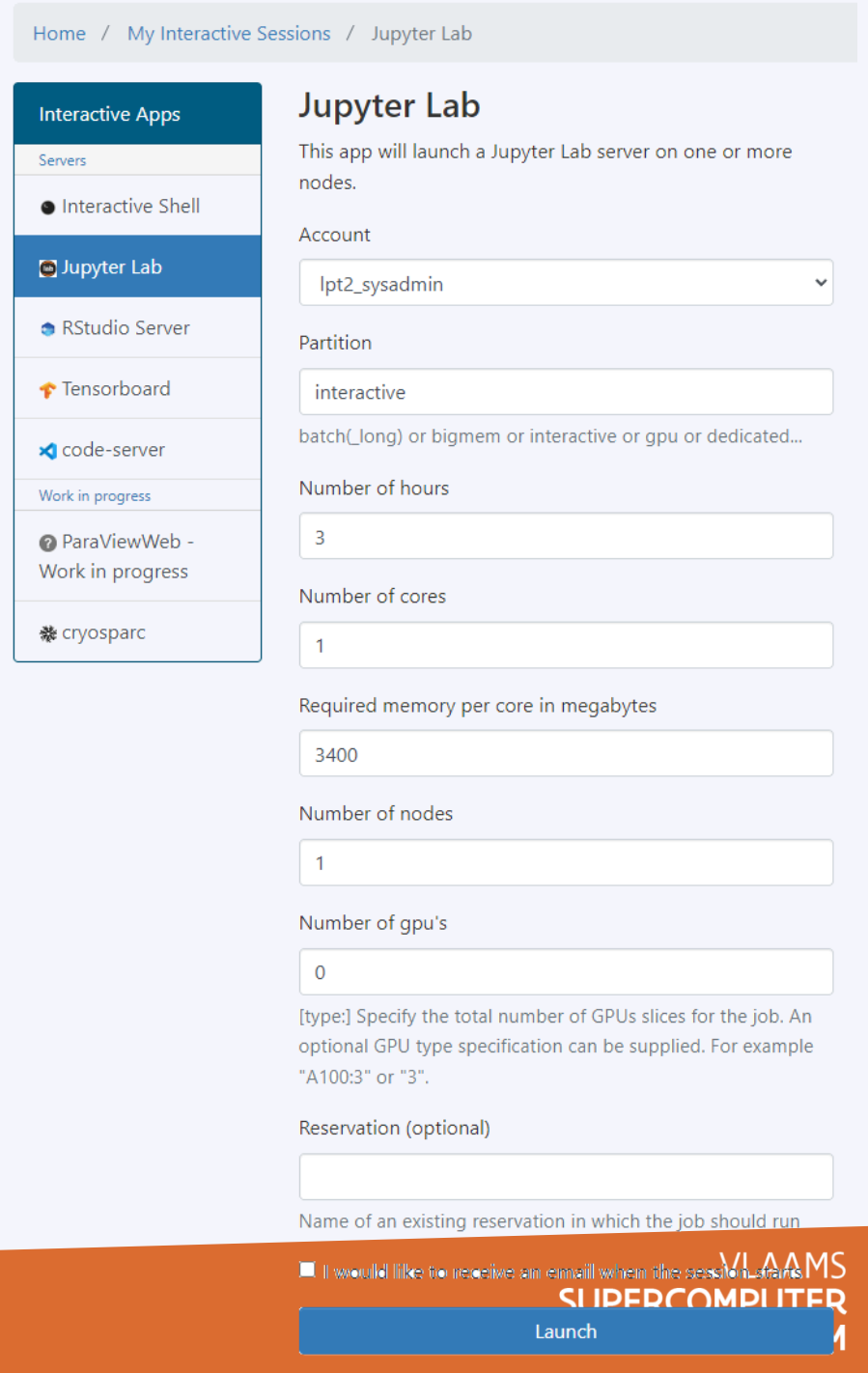
Open OnDemand

- 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



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



Software Stack

Software: Available Modules

- OS: Linux CentOS 7.x (Genius) and Rocky Linux 8.x (wICE)
- Toolchains (Genius):
  Intel 2018a (icc, icpc, ifort; Intel MPI; Intel MKL)
  FOSS 2018a (gcc, g++, gfortran; OpenMPI; ScaLAPACK, OpenBLAS, FFTW)
- wICE: 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/skylake/2021a/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

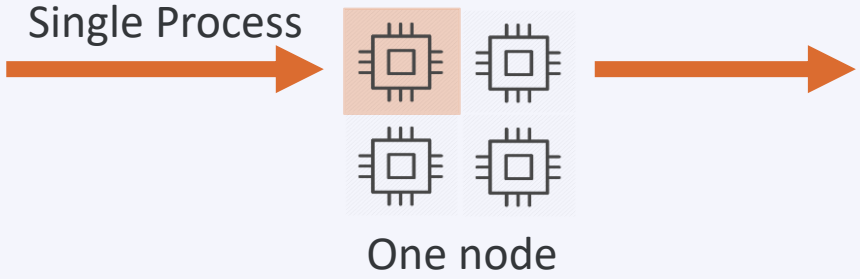
- You can always install your desired software in your $VSC_DATA
  Use Intel or FOSS toolchains
- Compile your code on a compute node (with interactive job)
- If you cannot, ask us for help
- Python/R packages for AI and ML must be installed by the users themselves
- Read more about Python Package Management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

Starting to Compute

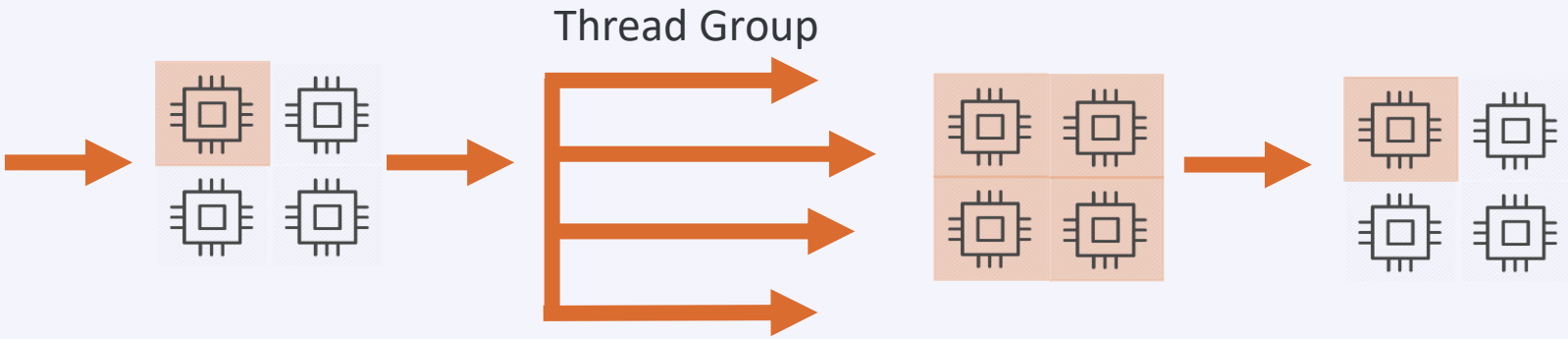
Resource Glossary

- ❑ **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?

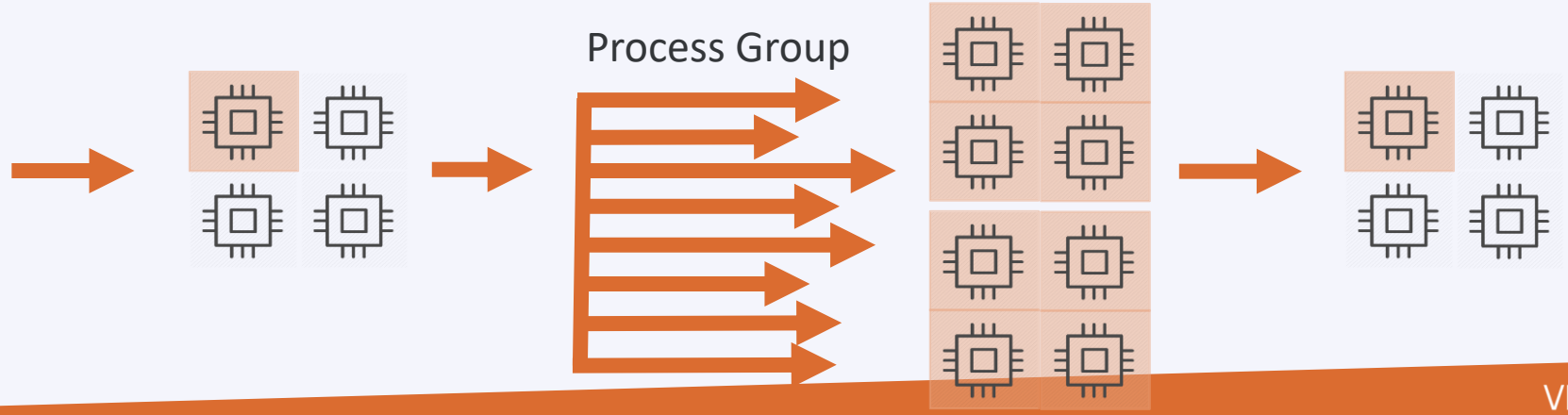
Serial Application
(1 process on 1 core)



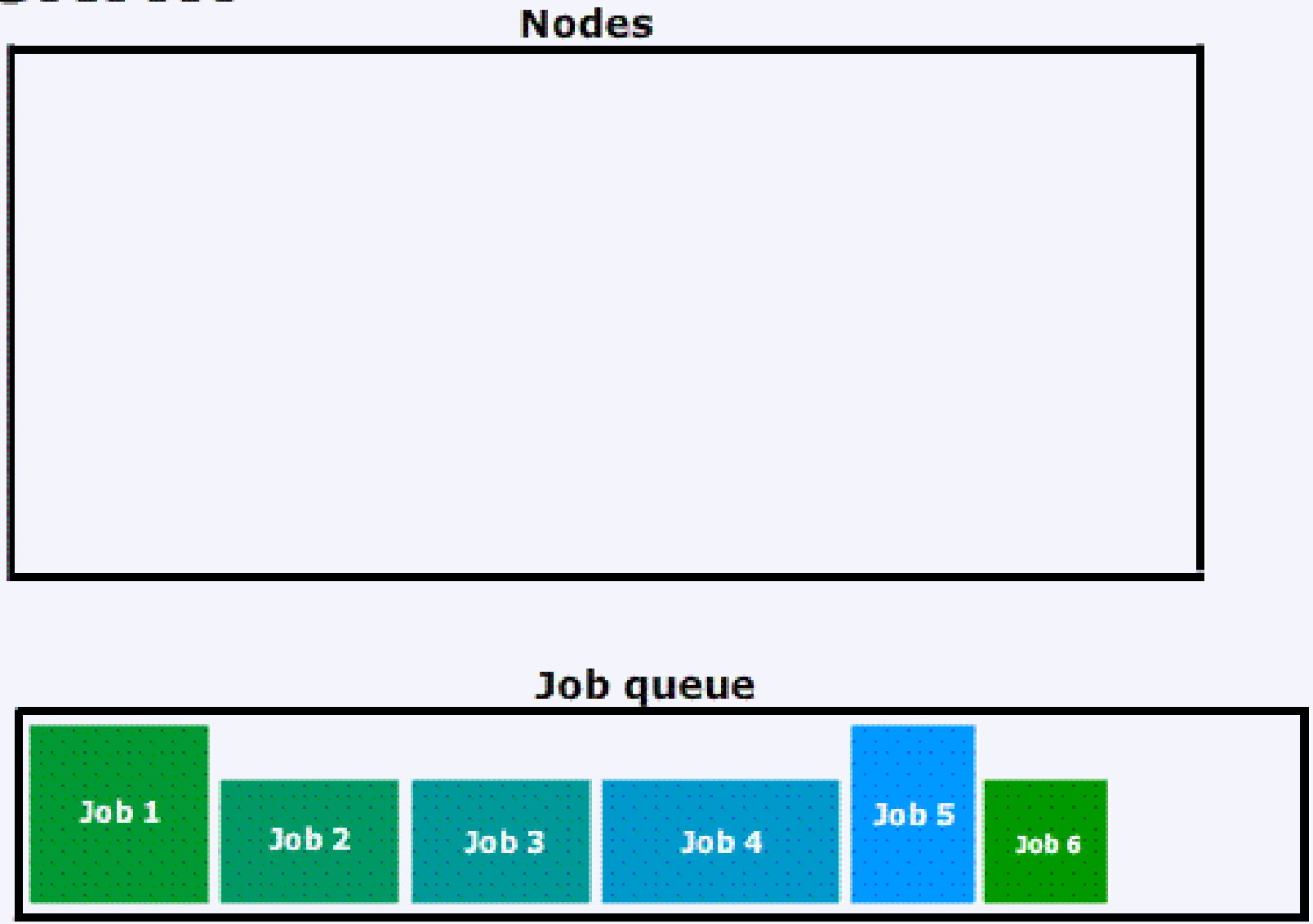
Multi-Core Appl.
(N threads on N cores from **1 node**)



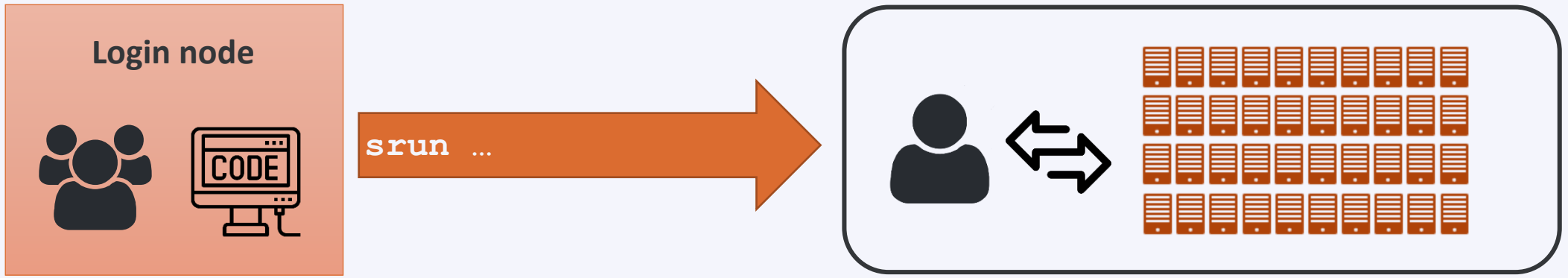
Distributed Appl.
(many processes on multiple cores/nodes)



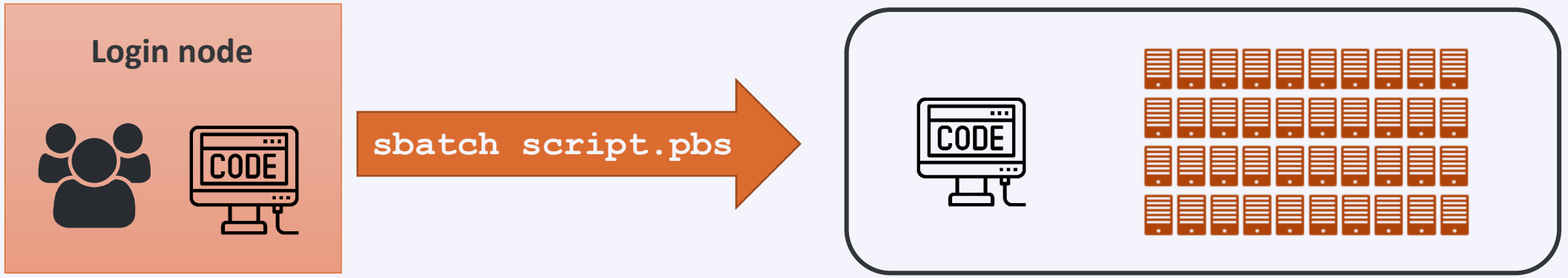
Backfill



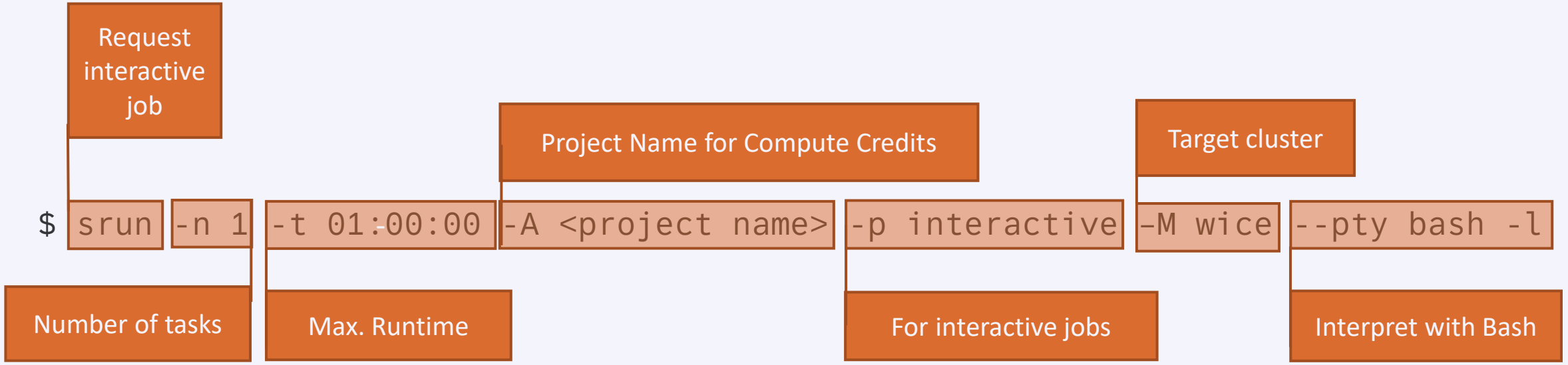
Interactive Job



Batch Job



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`

Interactive Jobs

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

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

- Interactive job with X-forwarding

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

- Request fraction of a node from interactive partition

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

- Request a GPU accelerator

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

Example: Slurm Job Script

```
#!/bin/bash

#SBATCH --cluster wice
#SBATCH --account lp_hpcinfo_training
#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

Execure commands

Move data

Batch Workload

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

Job Script

```
#!/bin/bash
…
```

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

STDOUT + STDERR

```
$ cat slurm-60042490.out
SLURM_JOB_ID: 60042490
SLURM_JOB_USER: vsc30745
SLURM_JOB_ACCOUNT: lp_hpcinfo
SLURM_JOB_NAME: hpc_workflow
SLURM_CLUSTER_NAME: wice
SLURM_JOB_PARTITION: batch
SLURM_NNODES: 1
SLURM_NODELIST: s28c11n2
SLURM_JOB_CPUS_PER_NODE: 8
Date: Wed Feb  8 10:41:21 CET 2023
Walltime: 00-01:00:00
==========================================================================
/apps/leuven/skylake/2018a/software/Python/3.6.4-intel-2018a/bin/python3
cp: cannot stat '/data/leuven/307/vsc30745/input_data': No such file or
directory
python: can't open file 'modelling.py': [Errno 2] No such file or directory
cp: cannot stat 'output_data': No such file or directory
```

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
`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>` `slurm_jobinfo <JobID>`	Detailed job info (very useful to diagnose issues)
`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
`slurmtop --cluster=wice`	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

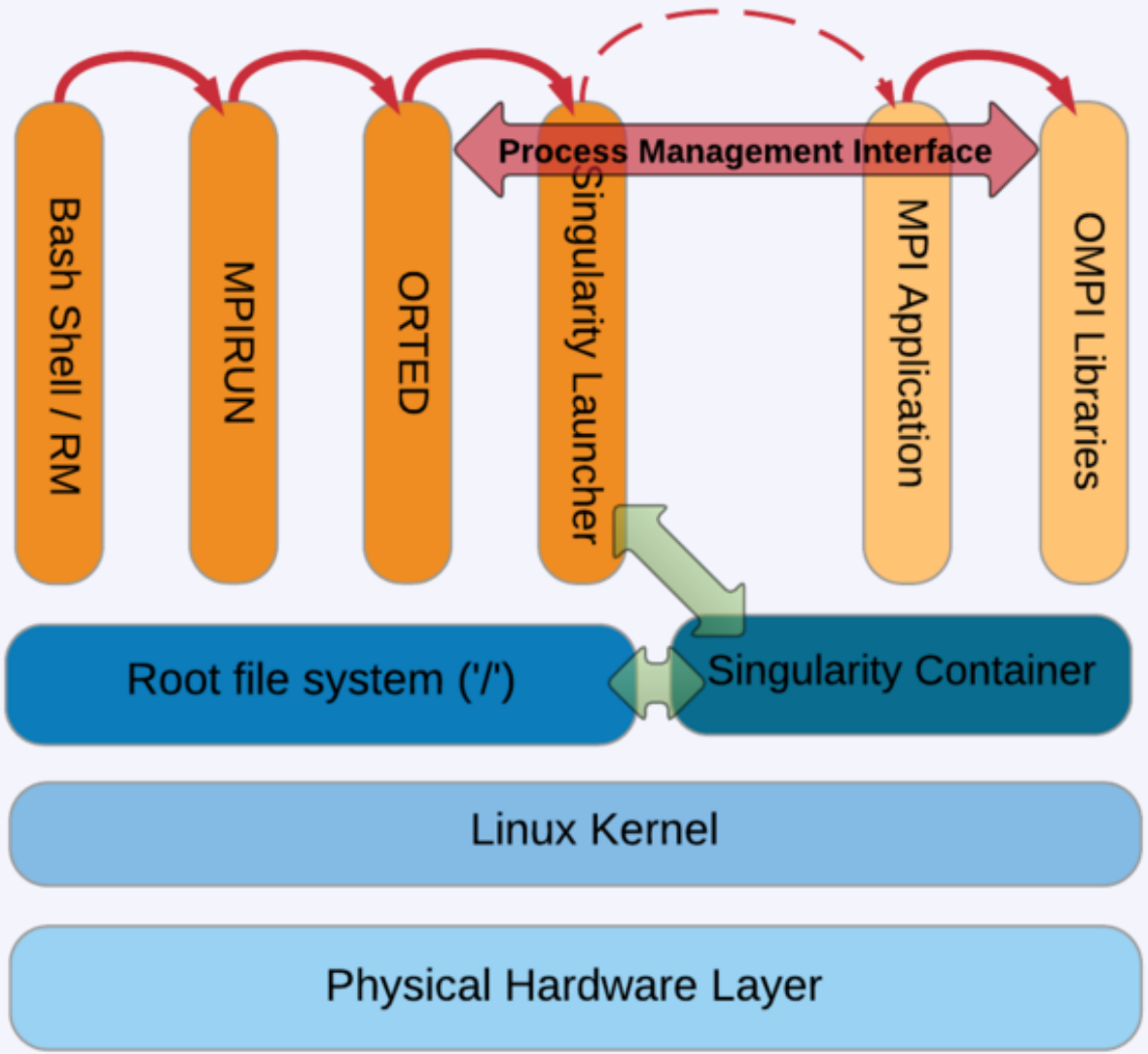
- What?
  + Self-contained OS & software & data
- Why?
  + fully resolved dependency chains
  + portable workflow
- How?
  + You create the image
  + Run it on Genius
  + MPI/OpenMP is supported

Slurm Script

```
#!/bin/bash -l

#SBATCH -t 30:00
#SBATCH -N 2
#SBATCH -n 72
#SBATCH -M wice
#SBATCH -A lp_hpcinfo_training


singularity run Project.simg ./model.exe
```



Credit Pricing

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

  #credits = 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
ThinKing / Haswell	6.68
ThinKing / IvyBridge	4.76

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

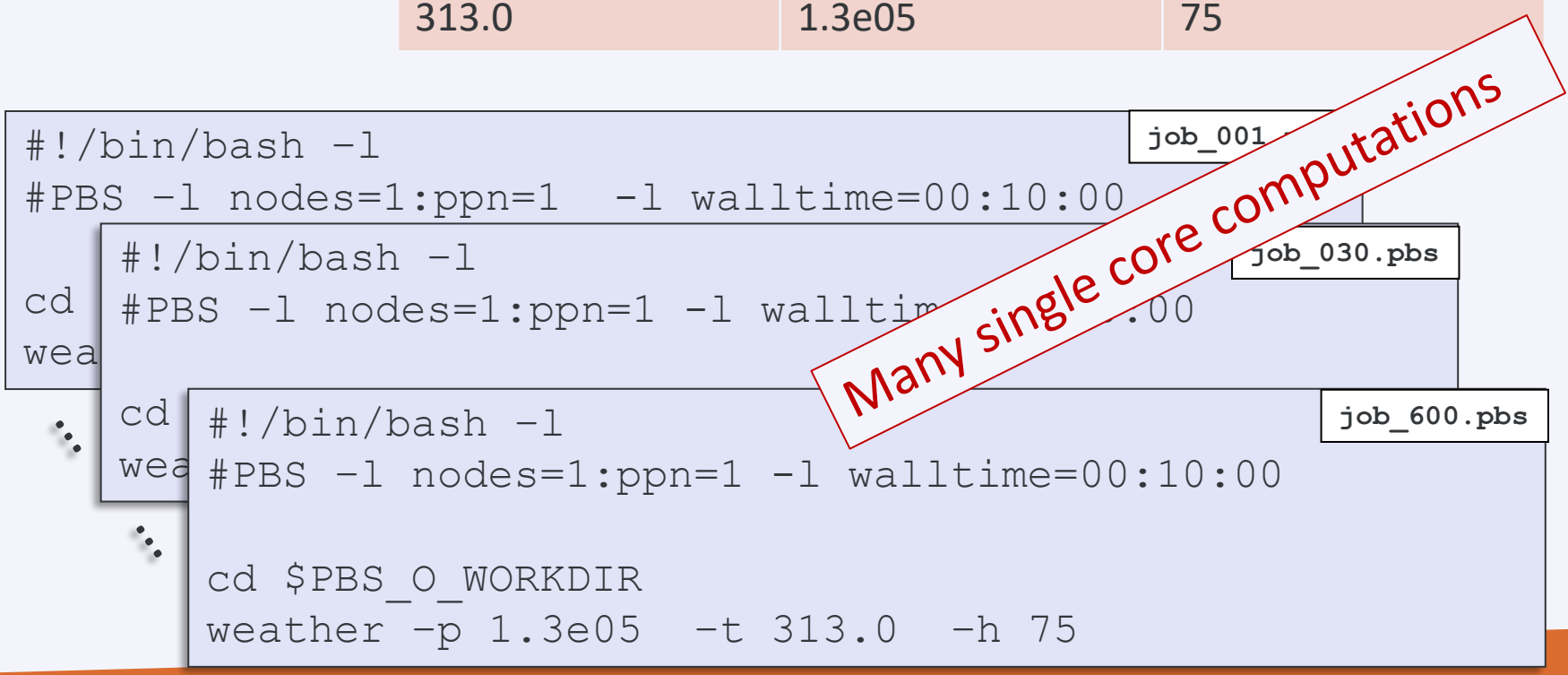
Worker Framework

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

Use case: parameter exploration

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



Solution: worker with -data

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

data.csv

job.pbs

```
#!/bin/bash -l
#PBS -l nodes=5:ppn=20  -l walltime=01:20:00

cd $PBS_O_WORKDIR
weather -p $pressure  -t $temperature  -h $humidity
```

```
$ module load worker
$ wsub  -data data.csv  -batch job.pbs
```

Data exploration: steps

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

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

Demo: Test for yourself

demo/test yourself

- Request membership to lp_hpcintro_training 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_HOME`
- Submit `cpujob` to the cluster
- List all your jobs (`qstat`)
- Check the information about the cpujob (`checkjob`)
- Modify the mat.pbs script to request 1 node, 36 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, htop`)

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 – notebooks

- Start an interactive GPU job

```
$ qsub -I -l walltime=30:00 -l nodes=1:ppn=9:gpus=1 -l partition=gpu -A default_project
```

- Activate conda environment

```
$ source activate science
```

- Go to your working directory (you can use `$PBS_O_WORKDIR` if you qsub from there)
  Start notebook

```
$ jupyter notebook --port ${USER:3} --ip $(hostname)
```

```
$ jupyter notebook --port 30468 --ip $(hostname)
```

- Open the link in the browser in NX and test your notebook

demo – worker

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

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.kuleuven.be/hpc

Stay Connected to VSC

LinkedIn®