High Performance
Computing &
Big Data Services

hpc.uni.lu

hpc@uni.lu

@ULHPC

LUXEMBOURG
LET'S MAKE IT HAPPEN

Getting Started on Grid'5000

Practical Tutorial Session

University of Luxembourg
08/09/2025

UNIVERSITY OF LUXEMBOURG

UNIVERSITY OF LUXEMBOURG

Table of Contents

Grid'5000 Getting Started Tutorial

Follow along with the tutorial at:

https://hpc-docs.uni.lu/g5k/getting-started-g5k/

A hands-on guide to using Grid'5000 for research

UNIVERSITY OF LUXEMBOURG

1. **Introduction**
2. First connection
3. First resource reservation
4. Advanced resource management
5. System deployment
6. Experiment automation
7. Summary and Wrap-up

What is Grid'5000?

Grid'5000 is a large-scale testbed for experiment-driven research in computer science

Key Features:

- 11 sites in France, Belgium and Luxembourg
- ~25,000 cores, 800 compute nodes, 550+ GPUs
- Highly configurable and controllable hardware (bare-metal)
- Advanced monitoring and measurement features
- Reproducible research support

Focus Areas: Parallel/distributed computing, cloud, HPC, Big Data, AI

Before we start

Make sure you have:

- A Grid'5000 account
- A Standard SSH client (OpenSSH on Mac OS, Linux or WSL)
- Your SSH key configured

⚠ Request your account here (select the group **luxembourg-misc**)

https://www.grid5000.fr/w/Special:G5KRequestAccountUMS

Key Concepts

Hierarchical Resource Organization:

Resource	Description
Cluster	Named group of homogeneous nodes
Node	Bare-metal server/machine
Core	Smallest reservable unit (CPU core)

Examples:

- Cluster: `larochette`, `clervaux`, `petitprince`
- Node: `larochette-1`, `clervaux-11`
- Luxembourg clusters: `clervaux` (48 nodes), `larochette` (7 nodes), `vianden` (1 node), `petitprince` (11 nodes)

Important resources

Grid'5000 Wiki: https://www.grid5000.fr

- Hardware specs: **https://www.grid5000.fr/w/Hardware**
- Status and monitoring: **https://www.grid5000.fr/w/Status**

Resource Availability:

- Check Drawgantt charts for real-time availability
- Plan reservations during off-peak hours
- Monitor maintenance schedules

UNIVERSITY OF LUXEMBOURG

1. Introduction
2. **First connection**
3. First resource reservation
4. Advanced resource management
5. System deployment
6. Experiment automation
7. Summary and Wrap-up

SSH keys setup

Before first login, generate your SSH keys:

```
user@pc: ssh-keygen -t ed25519
user@pc: cat .ssh/id_ed25519.pub
# Copy the output
```

Add your public key to Grid'5000:

1. Visit: https://api.grid5000.fr/stable/users/
2. Paste public key in "SSH Keys" tab
3. Save changes

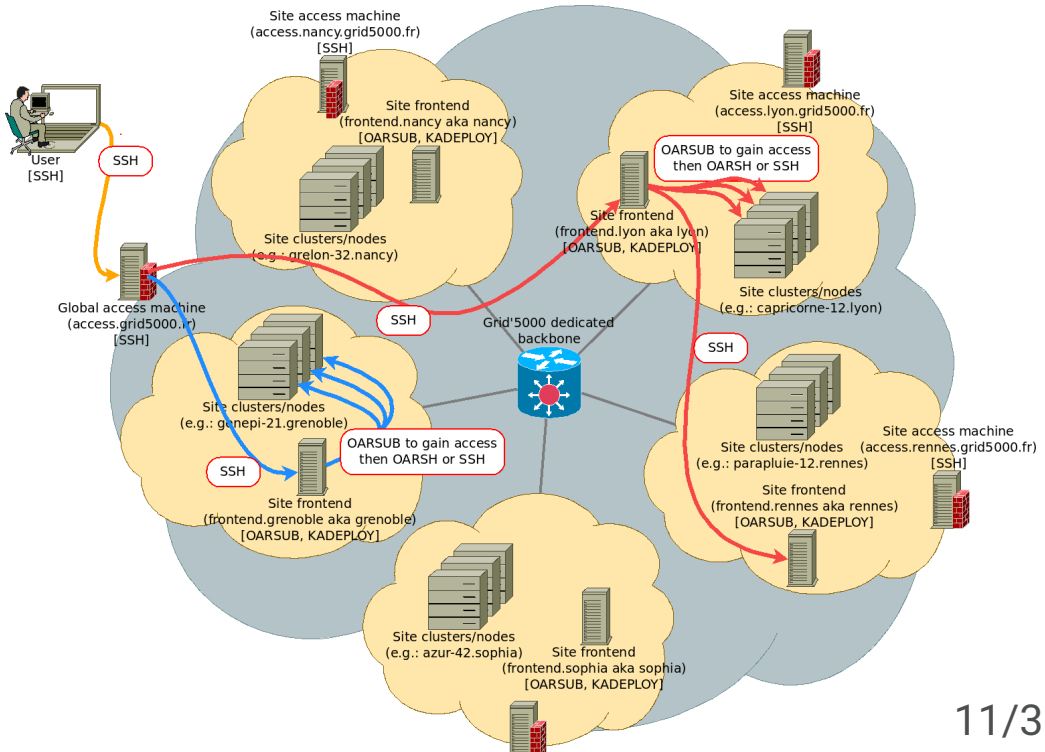
⚠ **Required before any Grid'5000 access**

First connection

Manual connection:

```
user@pc:~$ ssh access.grid5000.fr
user@access-north:~$ ssh luxembourg
user@fluxembourg:~$
```

You should see the welcome message.



SSH config (recommended)

Edit `~/.ssh/config` on your local machine:

```
Host g5k
  User <g5k_login>
  Hostname access.grid5000.fr
  ForwardAgent no

Host *.g5k
  User <g5k_login>
  ProxyCommand ssh g5k -W "$(basename %h .g5k):%p"
  ForwardAgent no
```

Now you can connect directly:

```
user@pc:~$ ssh luxembourg.g5k
```

UNIVERSITY OF LUXEMBOURG

1. Introduction
2. First connection
3. **First resource reservation**
4. Advanced resource management
5. System deployment
6. Experiment automation
7. Summary and Wrap-up

Reserving resources

Basic reservation with OAR:

```
[user@fluxembourg|~]$ oarsub -r now -q testing
# Filtering out exotic resources (vianden).
# Set walltime to default (3600 s).
OAR_JOB_ID=256908
# Advance reservation request: waiting for validation...
# Reservation valid --> OK
```

Check reservation status:

```
[user@fluxembourg|~]$ oarstat -u
```

Job States: W (waiting), L (launching), R (running), E (error), F (finished)

Connecting to the reservation

Get detailed job information:

```
user@fluxembourg:~$ oarstat -f -j <JOB_ID>
```

Connect to allocated node:

```
# Method 1: Direct SSH
[user@fluxembourg|~]$ ssh clervaux-11


# Method 2: OAR shortcut (single reservation only)
[user@fluxembourg|~]$ oarsub -C
```

Verify you're on the compute node - prompt should change!

Working on the node

Storage locations:

- `~/` - Home directory (25GB, NFS-mounted, persistent)
- `/tmp/` - Local disk (fast I/O, non-persistent)
- Storage groups - Large shared volumes

Best practice: Work in `/tmp/` for performance, copy results to `~/`

Example workflow:

```
user@clervaux-11:~$ cd /tmp/
# Download, compile, run experiments
# Copy important results back to ~/
```

Practical example

Download and run NAS Parallel Benchmarks:

```
user@clervaux-11:~$ cd /tmp/
user@clervaux-11:~$ wget 'https://www.nas.nasa.gov/assets/npb/NPB3.4.3.tar.gz'
user@clervaux-11:~$ tar -xzvf NPB3.4.3.tar.gz
user@clervaux-11:~$ cd NPB3.4.3/NPB3.4-OMP
user@clervaux-11:/tmp/NPB3.4.3/NPB3.4-OMP$ cp config/suite.def.template config/
suite.def
user@clervaux-11:/tmp/NPB3.4.3/NPB3.4-OMP$ cp config/make.def.template config/
make.def
user@clervaux-11:/tmp/NPB3.4.3/NPB3.4-OMP$ make -j$(nproc) suite
```

Practical Example

Run benchmarks and save results:

```
user@clervaux-11:/tmp/NPB3.4.3/NPB3.4-OMP$ mkdir /tmp/benchs
user@clervaux-11:/tmp/NPB3.4.3/NPB3.4-OMP$ for bench in $(ls bin); do ./bin/
$bench | tee /tmp/benchs/$bench.txt; done
user@clervaux-11:/tmp/NPB3.4.3/NPB3.4-OMP$ cp -R /tmp/benchs ~/benchs-$OAR_JOBID
```

Root Access with sudo-g5k

Gain administrator privileges:

```
user@clervaux-11:~$ sudo-g5k
user@clervaux-11:~$ sudo -iu root
root@clervaux-11:~#
```

Now you can:

- Install system packages: `apt update && apt install ...`
- Modify system configuration
- Install drivers, kernels, file systems, etc.

⚠ **Warning:** Using `sudo-g5k` triggers full node redeployment after job ends

UNIVERSITY OF LUXEMBOURG

1. Introduction
2. First connection
3. First resource reservation
4. **Advanced resource management**
5. System deployment
6. Experiment automation
7. Summary and Wrap-up

Advanced resource selection

Reserve specific cluster:

```
user@fluxembourg:~$ oarsub -r now -p clervaux
```

Reserve by hardware properties:

```
user@fluxembourg:~$ oarsub -r now -p "core_count > 8"
user@fluxembourg:~$ oarsub -r now -p "cputype LIKE 'Intel Xeon%'"
```

Reserve exotic resources:

```
user@fluxembourg:~$ oarsub -r now -p vianden -t exotic
```

Reserve multiple nodes for longer time:

```
user@fluxembourg:~$ oarsub -r now -l nodes=2,walltime=4:00
```

Advanced reservations

Schedule future reservations:

```
user@fluxembourg:~$ oarsub -r "2025-06-10 10:00, 2025-06-10 12:00"
```

Extend running reservation:

```
user@fluxembourg:~$ oarwalltime <oar_job_id> +1:00
```

Always clean up when done:

```
user@clervaux-11:~$ logout
[user@fluxembourg|~]$ oardel <JOB_ID>
```

💡 **Good practice:** Delete unused reservations to free resources for others

UNIVERSITY OF LUXEMBOURG

1. Introduction
2. First connection
3. First resource reservation
4. Advanced resource management
5. **System deployment**
6. Experiment automation
7. Summary and Wrap-up

System deployment

Reserve node for deployment:

```
user@fluxembourg:~$ oarsub -r now -t deploy -q testing
```

Deploy custom environment:

```
user@fluxembourg:~$ kadeploy3 -e debian12-min -m
clervaux-3.luxembourg.grid5000.fr
# or with job ID:
user@fluxembourg:~$ oarsub -C <job_id>
user@fluxembourg:~$ kadeploy3 -e debian12-min
```

System deployment

Available environments:

- `debian11-min`, `debian12-min` - Minimal Debian
- `debian11-nfs`, `debian12-nfs` - With NFS home mounting
- `ubuntu2404-min`, `ubuntu2404-nfs` - Ubuntu variants

List all environments:

```
user@fluxembourg:~$ kaenv3
```

Serial console access

Access node console (useful for debugging):

```
user@fluxembourg:~$ kaconsole3 -m clervaux-3
```

Reboot deployed node:

```
user@fluxembourg:~$ kareboot3 simple -m clervaux-3
```

Exit console: Type `&.` or press `ESC` 4 times

Use case: Recover from network misconfigurations, debug boot issues

UNIVERSITY OF LUXEMBOURG

1. Introduction
2. First connection
3. First resource reservation
4. Advanced resource management
5. System deployment
6. **Experiment automation**
7. Summary and Wrap-up

Commands

Submit command with reservation:

```
user@fluxembourg:~$ oarsub -r now "lscpu > lscpu.txt"
```

Script-based experiments:

```
user@fluxembourg:~$ chmod +x experiment.sh
user@fluxembourg:~$ oarsub -t deploy -r now "./experiment.sh" -q testing
```

Monitor job output:

```
user@fluxembourg:~$ multitail -i OAR.<jobid>.stdout -i OAR.<jobid>.stderr
```

Day vs Night usage:

- Day: Interactive work, shorter reservations
- Night/Weekends: Automated, longer experiments

REST API Usage

Programmatic resource discovery:

- Sites: **https://api.grid5000.fr/stable/sites/**
- Luxembourg clusters: **https://api.grid5000.fr/stable/sites/luxembourg/clusters/**
- Node details: **https://api.grid5000.fr/stable/sites/luxembourg/clusters/vianden/nodes/vianden-1**

REST API Usage

Automated job submission:

```
import requests

payload = {
    'resources': 'nodes=1',
    'types': ['deploy'],
    'command': './experiment.sh',
    'properties': f"cluster='{cluster}'"
}
job = requests.post(api_job_url, data=payload).json()
```

UNIVERSITY OF LUXEMBOURG

1. Introduction
2. First connection
3. First resource reservation
4. Advanced resource management
5. System deployment
6. Experiment automation
7. **Summary and Wrap-up**

Summary: Key Commands

Command	Purpose
`oarsub -r now`	Reserve resource immediately
`oarstat -u`	Check your reservations
`oarsub -C`	Connect to reserved node
`oardel <job_id>`	Delete reservation
`sudo-g5k`	Gain root access
`kadeploy3 -e <env>`	Deploy OS environment
`kaconsole3 -m <node>`	Access serial console
`oarwalltime <job> +1:00`	Extend reservation

Best Practices

Resource Management:

- Always delete unused reservations
- Use `/tmp/` for intensive I/O operations
- Copy important results to home directory
- Plan reservations during off-peak hours

Experimentation:

- Test interactively first, then automate
- Use version control for experiment scripts
- Document your environment and dependencies
- Monitor resource usage and platform status

Best Practices

Troubleshooting:

- Check Grid'5000 status page for issues
- Use `kaconsole3` for network problems
- Monitor job output files for errors

Next Steps & Resources

Advanced Topics:

- Network reconfiguration (KaVLAN)
- Multi-site experiments
- Custom environment creation

Key Documentation:

- Tutorial: **https://hpc-docs.uni.lu/g5k/getting-started-g5k/**
- Main wiki: **https://www.grid5000.fr/w/Home**

Support:

- Support page: **https://www.grid5000.fr/w/Support**
- User mailing list and forums

UNIVERSITY OF LUXEMBOURG

Questions?