How to connect to the cluster

You can only connect to the cluster from within the LUH network. If you want to connect from the “outside” (e.g. from Home), you'll have to establish a VPN connection to the university’s network first. See VPN Service for details.

Please note: Working from outside the LUH may feel much slower than from your office, depending on the quality (bandwidth, latency, symmetry of up- and downstream) of your internet connection. If graphical applications feel sluggish, it is usually due to the connection and/or the network components in between, not a technical problem with the cluster system. You should also keep in mind the necessary bandwidth for a graphical connection on a high-resolution screen. Quite frequently, you can drastically improve performance by experimenting with the compression settings of X2Go, reducing window sizes, or the use of Open OnDemand as described below in this guide. If you have a good internet connection, you should be able to work quite comfortably from Home, except perhaps if you are in countries that are distant to Germany.

We support the following methods for connecting to the compute cluster:

- SSH - secure shell client
- X2Go - remote desktop sessions
- Web browser - Open OnDemand web portal

The following addresses should be used to connect to the cluster system:

- **login.cluster.uni-hannover.de** in order to develop, prepare and submit jobs, editing text files.
- **transfer.cluster.uni-hannover.de** whenever you want to transfer data to/from the cluster.

☛IMPORTANT☚: The login machines are *not* to be used for computations. In order to keep these nodes accessible for everyone, processes that use too much cpu time will get killed automatically after 30 minutes of elapsed cpu-time (that translates to only a few minutes if you try to start a parallel job there. Don't do that). We will also kill anything that uses more “elaborate” mechanisms trying to circumvent the limits, and we will disable accounts that we determine to be acting in bad faith. Please use interactive jobs for tasks like pre- or post-processing and even some larger compilations in order to avoid the frustrating experience of sudden shut down of your application. You can, of course, also use our Open OnDemand web portal which is reachable under https://login.cluster.uni-hannover.de if you get overwhelmed with all the options of the batch system at first.

Please note: File transfers on the login nodes will also get aborted if the execution time limit is reached. On the transfer node, execution time is unlimited, but you can not submit batch jobs from there to avoid clogging the node. We recommend to use ssh/scp or rsync to transfer files to/from the cluster, in particular for large files. The rsync command is particularly useful if your connection is unstable, because it can continue transfers that have been aborted.

If you already know what you are doing: current hashes of the host key fingerprints of login and transfer

nodes

Connecting via ssh for the first time, you will probably get asked about the authenticity of the host key/hash presented to you by the login node you are trying to connect. This is done in order to ensure that you really connect to our servers and not a machine that someone could in theory have inserted as a so-called “man-in-the-middle-attack” in your connection or with another of various methods. So we suggest to check what the dialogue presents to you to ensure you do not enter your password at the wrong site. You will probably see only one of the following lines, since there are various algorithms used to compute the fingerprints and we do not know which one your tool will use - check for the algorithm used at the end of the line, e.g. ECDSA, ED25519.

The correct host key fingerprints are

```
MD5: bf:aa:71:bd:d1:19:d4:4f:b5:60:e7:cb:c4:26:85:a7 (ECDSA)
SHA256: cYIZZQC96J5bhannZnH2cGQLIBPLQVy29HNc6/vnyFg (ECDSA)
SHA1: fb:a1:47:ad:97:99:b3:66:86:06:8d:9d:56:46:61:3a:44:9c:a6:cb (ECDSA)

MD5: 3b:0a:17:0f:53:85:40:87:c2:be:ed:65:fb:40:59:8a (ED25519)
SHA256: sG8ZYabQctyGjxPD7X8K2IBxJIE5xHHZ9mQqjlVcjxo (ED25519)
SHA1: 51:f7:b9:0a:e3:84:8e:d8:2c:a4:e7:7d:42:14:d3:8b:62:2e:a6:5d (ED25519)

MD5: bd:b2:04:f9:4e:36:c1:b6:36:8e:d9:03:cc:5d:75:c4 (RSA)
SHA256: zkVF8Xyxmm7bOEKTNOlvRKe+nG1oHZDeOtY3Un60grg (RSA)
SHA1: ed:e3:ed:e6:bf:4f:29:9a:2c:72:92:c1:b4:ff:a6:b9:81:f6:6b:45 RSA)

MD5: e3:c7:80:67:68:5b:d9:78:df:50:d7:e1:c5:ae:bf:e7 (DSA)
SHA256: DuWre1exwDsyTzD4yIMy6Oc2CFBCmx5o+l0LmWjMvUc (DSA)
SHA1: 85:95:f5:c9:9b:2b:29:82:5f:13:70:b9:2b:43:44:84:22:8f:87:40 (DSA)
```

If at least one of the fingerprints matches the one you get asked about, you can be reasonably sure that you are trying to connect to the cluster. In case you are using the command-line ssh client, it will remember an affirmative answer by adding a line with that fingerprint to your ~/.ssh/known_hosts file. So you'll probably get asked only once.

SSH Connection to the cluster

The most basic option to connect to the cluster system is via SSH. For this, an ssh client is required, which comes with most Linux distributions by default and also should be readily available under Mac OS (start a terminal, applications::utilities).

Under Windows, you'll need to install an SSH Client like e.g. PuTTY or much more feature-rich and user friendly MobaXTerm. You could also install a Unix-like environment like Cygwin to benefit from some Unix-like functionality under Windows, but be aware that for many use cases, there may be easier ways to get you started (see below).

When you've found a terminal, the following command will establish a connection to the cluster system.

```
ssh <username>@login.cluster.uni-hannover.de
```

Replace `<username>` with your cluster user name.

Please note: The settings dialogue of a tool like PuTTY itself is *not* a terminal, it's only a settings dialogue. So PuTTY only needs to know the hostname you want to connect to, it executes the ssh command automatically when connecting. It will first ask for your username, except if you entered username@ in front of the host name, which would also be ok. It is sufficient to enter “login.cluster.uni-hannover.de” into the “Host name” field, make sure to NOT add “ssh” in front of it.

If you want to use graphical programs on the cluster system, add the option `-X`, which enables X11 forwarding (see below). Depending on your system, you may need to use `-Y` instead of `-X`.

```
ssh -X <username>@login.cluster.uni-hannover.de
```

Please note: Again, tools like e.g. PuTTY have their own setup. To get X11 forwarding with these tools, use the corresponding button. In the PuTTY configuration settings, this would be Connection::SSH::X11::Enable X11 forwarding.

Passwordless Login via SSH Public Key

Using SSH keys instead of your account password provides a more secure and convenient way to log into the cluster. This method is highly recommended, especially if you frequently connect to the cluster or automate file transfers via `scp` or `rsync`. Important: passwordless does NOT mean *without a pass phrase* (which is the term used for passwords for ssh keys). Please ensure you are using a good passphrase to protect your private keys.

The setup requires generating a key pair on your local machine and appending the public key to the `authorized_keys` file in your cluster home directory.

Step 1: Generate an SSH Key Pair on Your Local Machine

Open a terminal on your **local machine** (Linux/macOS) or your Unix-like terminal environment (such as MobaXterm or WSL on Windows). Run the following command to generate a modern and secure Ed25519 key pair:

```
ssh-keygen -t ed25519 -f ~/.ssh/id_ed25519_cluster
```

When prompted, you **must set a strong passphrase** to encrypt your private key. Leaving the passphrase empty poses a significant security risk for you, your institute and for the cluster and all your colleagues using the cluster. Hopping from one server to the next via user accounts without a passphrase to obscure their whereabouts is standard “sport” even for unexperienced “hackers”.

This command creates two files in your “hidden” `~/.ssh/` directory (on Windows, this is typically located at `C:\Users\username\.ssh\`):

- `id_ed25519_cluster` (Your **private** key – keep this safe and confidential on your local machine, do NOT copy it to the cluster)

- `id_ed25519_cluster.pub` (Your **public** key – this is safe to share and will be copied to the cluster)



Your private SSH key is strictly confidential and must never leave your local device. Anyone who obtains this key can impersonate you and misuse your cluster account by gaining unauthorized access to your data and resources on the cluster.

Always encrypt your key with a passphrase. Never share your private key, do not upload it anywhere or store it on unencrypted external drives. Sharing your private key or account violates our Terms of Use. If you suspect your keys have been compromised, remove them immediately from your cluster account and contact support.

Step 2: Copy the Public Key to the Cluster

You need to append the contents of your local **public** key (`.pub`) to the `~/.ssh/authorized_keys` file on the cluster.

The easiest and safest way to automate this from a Linux, macOS, or modern Windows terminal is using the `ssh-copy-id` utility:

```
ssh-copy-id -i ~/.ssh/id_ed25519_cluster.pub username@login.cluster.uni-hannover.de
```

Replace `username` with your actual cluster user name. You will be prompted to enter your cluster password one last time.

Alternative (Manual Copy): If `ssh-copy-id` is not available on your local system, manually log into the cluster, ensure the `.ssh` directory exists, and append your public key string to `~/.ssh/authorized_keys`.

Step 3: Configure Strict Permissions

The SSH daemon on the cluster will silently reject your key pair if the file permissions are too open. Ensure that only your user account has access to these directories and files.

Log into the cluster and execute the following commands to enforce strict permissions:

```
chmod 700 ~/.ssh
chmod 600 ~/.ssh/authorized_keys
```

Step 4: Connect Using Your SSH Key

Once configured, you can log in using your private key by passing the `-i` flag:

```
ssh -i ~/.ssh/id_ed25519_cluster username@login.cluster.uni-hannover.de
```

Tip 1: Simplify Connection via SSH Config File

To avoid typing the full username, hostname, and identity file every time, you can create or edit the `config` file on your **local machine** (`~/.ssh/config`):

```
Host luh-cluster
    HostName login.cluster.uni-hannover.de
    User <username>
    IdentityFile ~/.ssh/id_ed25519_cluster
    X11Forwarding yes
```

With this configuration in place, you can connect to the cluster and automatically enable X11 forwarding by simply typing:

```
ssh luh-cluster
```

Tip 2: Simplify Connections Using ssh-agent

On most Linux systems, the system will automatically start a tool called ssh-agent for you that is able to securely cache your passphrase so you do not need to always re-enter it for every individual command. Add your cluster key to its cache using

```
ssh-add -t 1d ~/.ssh/id_ed25519_cluster
```

In the example, the passphrase for the key will be cached for one day, after which it will automatically be forgotten.

Remote Desktop Sessions via X2Go

One way to enable the usage of graphical programs on the cluster is to run an X-Server on your local machine, which means nothing more than running a piece of software that other software (called X-Clients in this context, but really just the graphic programs that you start) can make graphic requests to. Conveniently, an X-Server is usually the foundation for graphic environments under Linux so it is usually already installed here, and graphical connections tunneled through ssh (via `ssh -X` or `ssh -Y`, see above) should work right out of the box, but Mac OS users will typically need to install an external package such as XQuartz. Under Windows, MobaXterm comes with an X-Server, while for PuTTY there is VcXsrv.

For improved performance, however, it may still be useful to install and configure the X2Go client, as connections made via X2Go are compressed and optimized. The standard “X” protocol is not designed for use over wide-area networks, and newer tools can improve performance quite a bit. Th X2Go client is part of most Linux distributions' package repositories.

On Windows, X2Go-client version 4.1.2.0 has been tested to work with the cluster.

X2Go can be obtained here. **During installation, please make sure to check the "full installation" box to install all fonts. Otherwise, you may experience strange error messages later on.**

Please configure the X2Go client as described in X2Go client configuration.

IMPORTANT: Never delete the `~/.x2go` directory inside your cluster home folder, as it contains configuration data. Removing this directory on the cluster will corrupt your X2Go environment and completely block all your future X2Go connections to the cluster.

X2Go client configuration

Please note: We recommend using X2Go in broker mode to avoid having multiple desktop sessions. See section about X2Go Broker.

After starting the X2Go client, either using a desktop short-cut or using the start menu, a configuration dialogue is displayed. In this dialogue you should specify a session name and make the following four entries.

1. **Host**: login.cluster.uni-hannover.de
2. **Login**: Your user name
3. **SSH-Port**: 22
4. **Session type**: XFCE

If you have already set up an SSH public key (as described in the previous section), you can use it for your X2Go connection as well. In the same configuration dialogue:

- In the field **Use RSA/DSA key for ssh connection**, provide the path to your **private** key file (e.g., `~/.ssh/id_ed25519_cluster` or `C:\Users\username\.ssh\id_ed25519_cluster`).
- Select the checkbox **Try auto login (via SSH Agent or default SSH key)** to enable a seamless, passwordless desktop session.

The completed configuration dialogue is depicted in figure 1. Entries in the red boxes have to be set accordingly. Afterwards leave the configuration assistant by clicking the **OK** button.

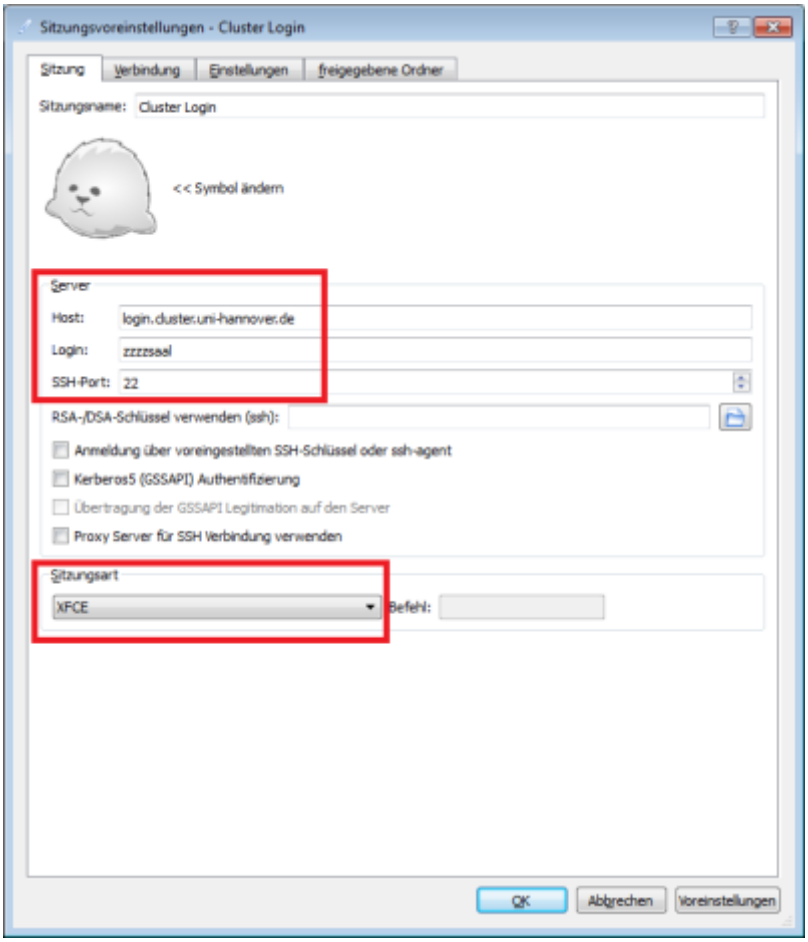


Fig. 1: X2Go configuration dialogue, entries in red boxes have to be set

On the right side of the main window the newly created session name is displayed, see figure figure 2. You can start this session by clicking on the session name (in the upper right corner in figure figure 2) or by entering the session name in the dialogue box named **session**.

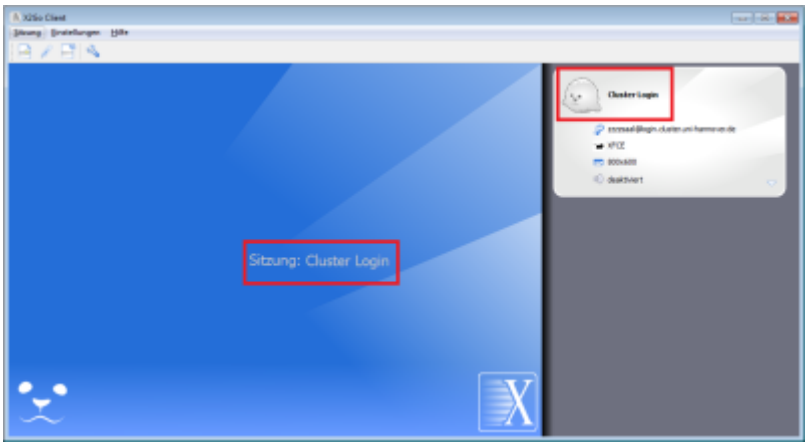


Fig. 2: Start a new session by clicking or entering the session name

The first time a connection is established, the login nodes' host-key is unknown. A notification will pop up and you need to accept the host-key (see figure figure 3) by pressing **yes**.

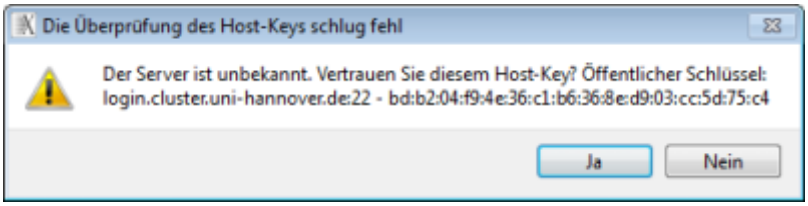


Fig. 3: Host key verification dialogue

After a connection was successfully established an XFCE Desktop is displayed as depicted in figure figure 4.

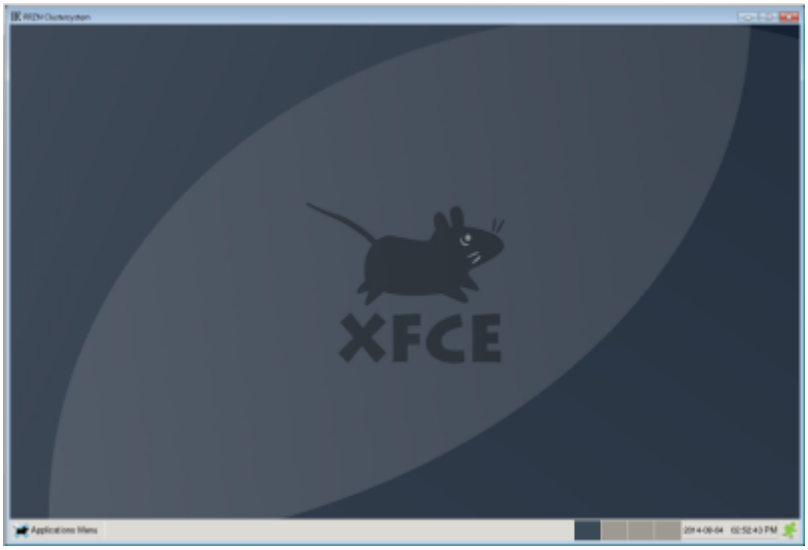


Fig. 4: XFCE desktop with applications menu in the lower left corner

The *Applications Menu* in the bottom left corner can be used to start a console window and then load modules or submit jobs into the queue. You can open editors e.g. to write or edit batch scripts. Particularly interactive jobs which open graphical program windows can be run. To end your session either go to the *Applications Menu* or press the little green icon in the bottom right corner of your desktop.

When connecting to the cluster using `login.cluster.uni-hannover.de` you will be assigned to one of the login nodes used for the cluster to balance the connection load. Therefore, it may happen that by next connection you will be taken to a different login node rather than to a node you already have a running X2Go session. In order to avoid maintaining multiple simultaneously running sessions, which easily may lead to confusion, we recommend using X2Go client in broker mode, see the next section.

Note: Suspended X2Go sessions will be terminated after four weeks without prior notice.

Note: We had an issue that made it impossible to enter special characters using AltGr in a terminal within an X2Go session. We hope that this is now solved.

X2Go Broker

If you would like to reconnect to a graphical session, use X2Go broker. For example you could start a session at the university and reconnect to it at home. In order to do this, you have to establish a connection to the cluster via X2Go broker. After installing X2Go client, proceed as described below.

X2Go broker on Linux

Use the following command to establish a connection with X2Go broker (the following command should be on one line. Replace <username> with your cluster username).

```
x2goclient --broker-url=ssh://<username>@x2gobroker.cluster.uni-hannover.de/usr/bin/x2gobroker --broker-autologin
```

X2Go broker on Windows

Either edit the existing shortcut to X2Go or create a new one (chose “Eigenschaften” in Picture figure 5).

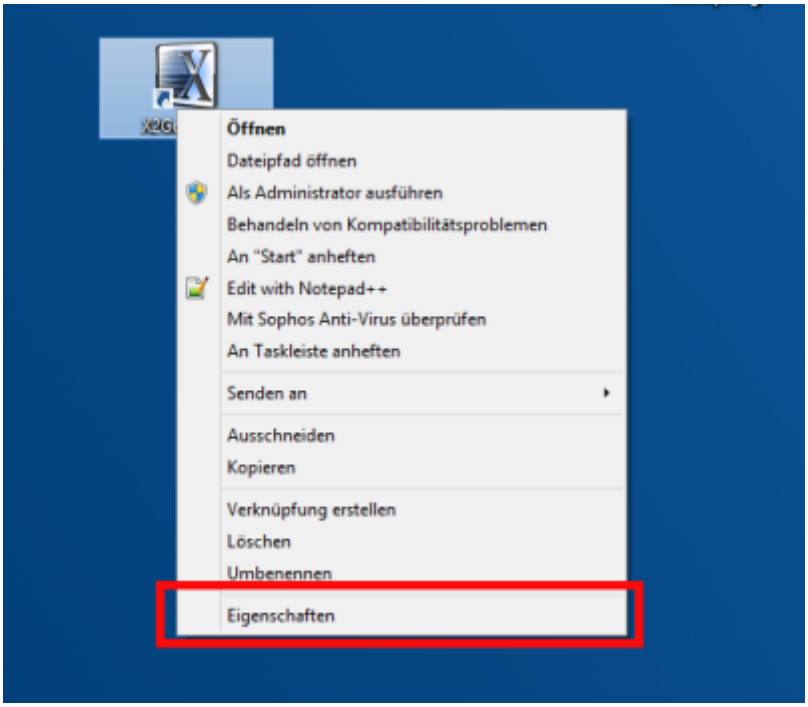


Fig. 5: Edit X2Go shortcut

Extend the command given as “Ziel”, see Picture figure 6, with the following parameters (the following command should be on one line. Replace <username> with your cluster username).

```
"[...]x2goclient.exe" --broker-url=ssh://<username>@x2gobroker.cluster.uni-hannover.de/usr/bin/x2gobroker
```

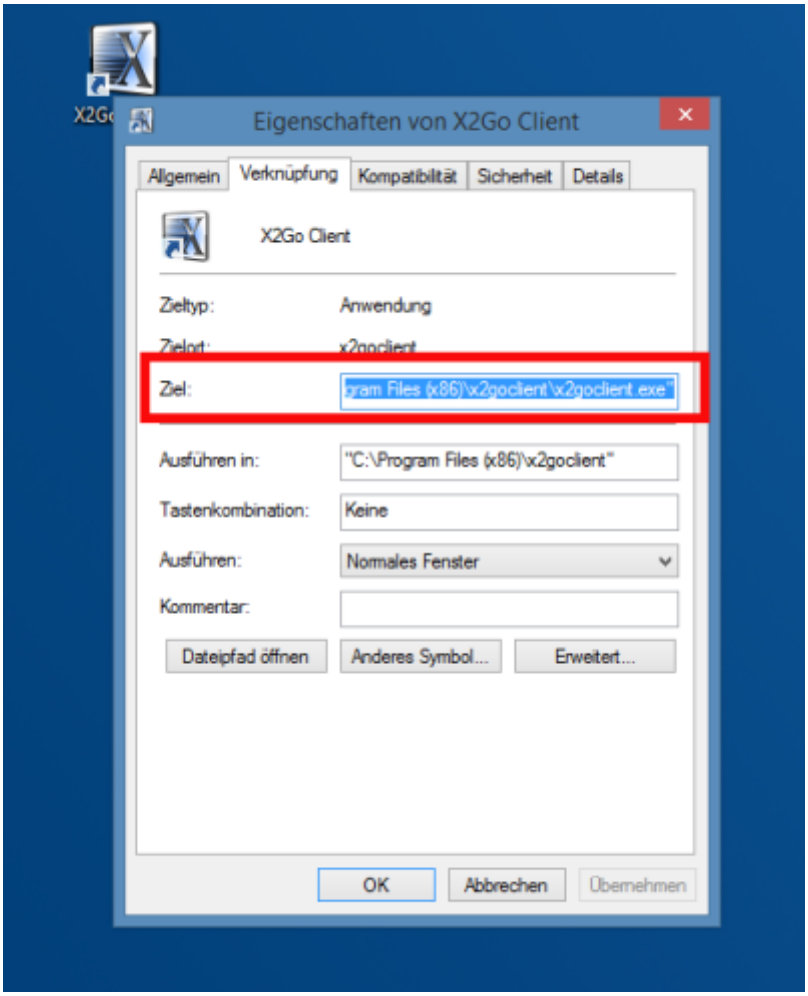


Fig. 6: X2Go broker command

After providing your password, a session is listed in the X2Go window, see picture . Choose this session. You will get a desktop on the cluster system. You can reconnect to this session and continue working graphically.

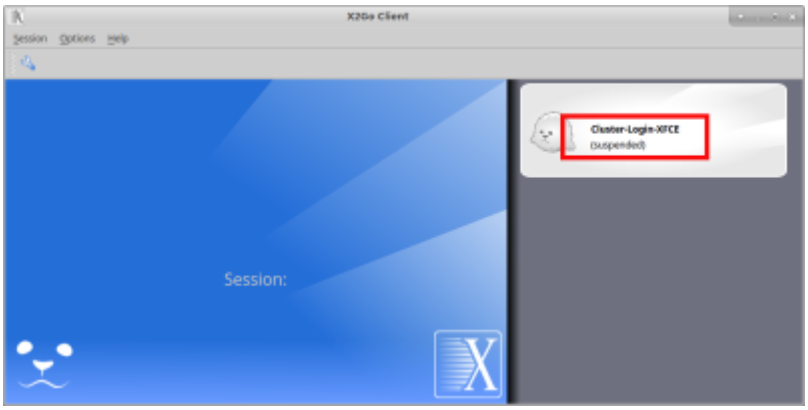


Fig. 7: X2Go broker session

Please note: Suspended X2Go sessions will be terminated after four weeks without prior notice.

Cluster Web Portal

The web interface powered by the software package Open OnDemand allows you to access the LUIS

cluster resources using a web browser without installing additional software on your personal workstation. The portal is currently confirmed to work with newer versions of Google Chrome (70+), Firefox (85+) and Microsoft Edge (90+). Any device connected to the university network and compatible with these browsers can be used. Safari is partially supported.

From within the Open OnDemand environment, you can:

- Create, submit, cancel and monitor batch jobs
- Open a terminal connection to cluster login servers and compute nodes
- Browse, edit, download and upload files in/to your HOME, BIGWORK and PROJECT directories
- Run noVNC Remote Desktop sessions on compute nodes for heavy GUI applications
- Run other preconfigured interactive applications like Jupyter/JupyterLab, MATLAB, COMSOL, etc

The Open OnDemand website provides additional information about the current and future directions of the project.

How to connect to the web portal

Please note: To access the portal, make sure you are connected to the University network, e.g. via the LUIS VPN Service.

Log on to the cluster web portal using your normal cluster login credentials by opening a new page in your web browser pointing to https://login.cluster.uni-hannover.de (see figure figure 8).



Fig. 8: Cluster web portal login page

Once you have been connected to the portal, you will be presented with the main dashboard page, see figure figure 9. There, you will find several menus to enable access to the different Applications for File Browsing, Job Management and Interactive Computing. Tutorial videos in the section “`Getting started with OnDemand`” in the dashboard page (klick on “LUIS OnDemand Dashboard” on the top left of the web page) explain further details about using the web portal.

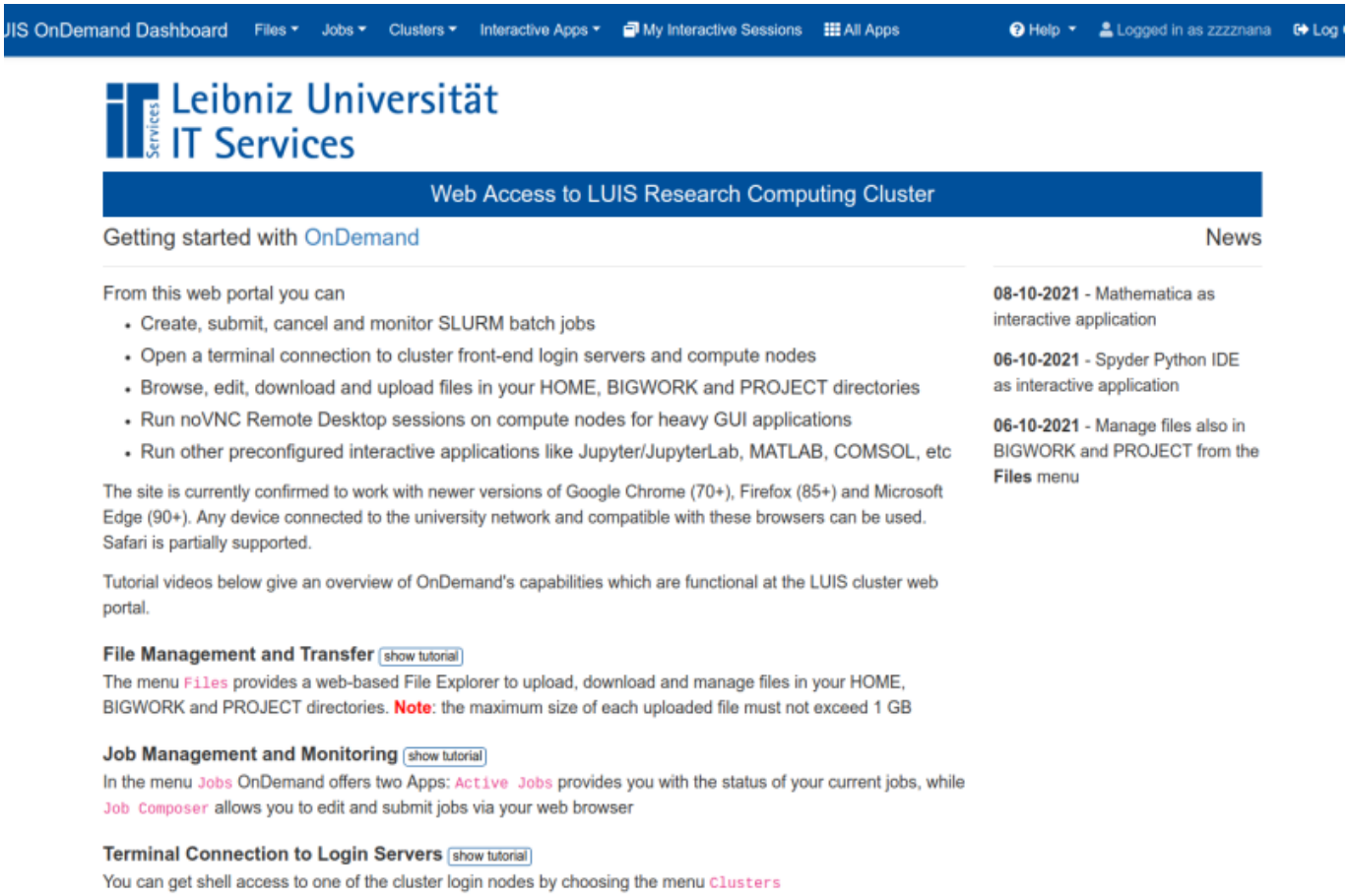


Fig. 9: Cluster web portal dashboard page

Services available via Open OnDemand

- Remote visualization for interactive 3D work
- Jupyter on the cluster
- Code-Server on the cluster

From:
https://docs.cluster.uni-hannover.de/ - **Cluster Docs**

Permanent link:
https://docs.cluster.uni-hannover.de/doku.php/guide/connecting_to_cluster

Last update: **2026/11/05 07:42**