FAQ

2023-11-19T17:10:58.026Z

# Frequently Asked Questions

## How do I use NoMachine?

### NoMachine Client

The first step to using NoMachine is to get and install the NoMachine Enterprise Desktop Client, which can be downloaded at [NoMachine Enterprise Client](https://www.nomachine.com/product&p=NoMachine%20Enterprise%20Client).

!!! warning “Network restrictions” The NoMachine servers are only visible from on-site, with a VPN connection or via some advanced SSH tunneling.

The long term goal is to have our NoMachine nodes integrated into OnDemand, but at the current time that is a work in progress.

### Reserving a Node

Login nodes are open to any connection, but the gpu-sm[01-02]-[01-20] nodes will require that you have a running job on the node before you are allowed to log in to the node desktop. To claim the node as your own, submit a job like:

[john.hanks@login01 ~]$ srun --exclusive --partition=gpu --gpus=a40:1 --time=8:00:00 --pty bash -l  
[john.hanks@gpu-sm01-10 ~]$

You can then attach to the assigned node. For longer sessions, using sbatch to wrap the sleep command will use a batch job which is not subject to the job being lost if, for instance, the session with the srun were dropped or network connection to your laptop lost. That sbatch would look like:

[john.hanks@gpu-sm01-10 ~]$ sbatch --job-name=NOMACHINE --partition=gpu --gpus=a40:1 --time=8:00:00 --exclusive --wrap "sleep 8h"  
Submitted batch job 40547  
[john.hanks@gpu-sm01-10 ~]$ squeue -j 40547  
 JOBID PARTITION NAME USER ST TIME NODES NODELIST(REASON)  
 40547 gpu NOMACHIN john.han R 0:09 1 gpu-sm01-01

And the assigned node is now yours for the duration of the job.

!!! warning “Use --exclusive” Be sure to include the --exclusive flag, or other people will be able to run jobs on the node if any CPUs are available.

### Connecting to the Node

Begin by setting up an SSH tunnel to the node. Following the job submission examples above, our node is gpu-sm01-10 so we need to tunnel to that node via a cluster login node, to the NX service on port 4000. In a terminal connect with

# Off-site and not on VPN, jump through login-01 or login-02  
 $ ssh -J login-01 -L 24000:localhost:4000 YOUR.LOGIN@gpu-sm01-10  
  
 # On-site or on VPN, jump through login01 or login02  
 $ ssh -J login01 -L 24000:localhost:4000 YOUR.LOGIN@gpu-sm01-10

The ssh commands show here will prompt as needed to log you in and tunnel port 24000 on your local laptop/workstation to the compute node’s port 4000 where the NoMachine service is listening.

Once the tunnel is connected, create a connection in the NoMachine client with the settings:

* Host: localhost
* Port: 24000
* Protocol: NX

Note: 24000 is an arbitrary value, to connect to multiple nodes just change that to any value between 4096 and 65535. If there is an error due to a collision just pick another port.

### Future

Watch the #hpc-community Slack channel for updates as we make progress on putting a better interface around NoMachine desktop access.

## ${THING} used to work, broken now.

When something that was working, stops working, here are some of the common things to look for:

* Has anything been added to your ${HOME}/.bashrc or ${HOME}/.bash\_profile?
* Have you,possibly inadvertently, installed something into ${HOME}/.local which causes a conflict in binaries or libraries?
* Are you hitting a quota or full disk issue?
* Has your input data or parameters changed? Now using a larger data set, for instance.
* Have you tested from an incognito browser window and/or flushed the browser cache?

If none of those things seem to be the issues, reach out to us for additional help troubleshooting.

## Why can’t I use sudo?

The cluster environment is a shared resource, with many people and groups accessing it. Because of that, data privacy is 100% dependent on unix file permissions and ACLs. Anyone with sudo access can bypass those permissions, which means as a rule we cannot give out sudo to individual users who are not approved admins for the cluster. Additionally, the cluster environment is complex and administrative activities need to be carried out in a restricted, planned and well communicated fashion, something that becomes less realistic to accomplish as the number of people with sudo access increases. In order to keep the environment as stable and secure as possible, access to sudo is very restricted.

The flip side of that requirement is that it means we (sysadmins) have to be ***very*** responsive to address user requirements. If you have a need that you believe requires sudo, contact us and we’ll either take care of the parts requiring sudo access or help you achieve the same goal from userspace without needing sudo. As an example of why sudo is rarely a hard requirement, the cluster application stack (module environment) is almost entirely configured without the use or need of sudo access.

## Why is my ${HOME} so small?

Cluster operation has a hard dependency on a fast, responsive ${HOME} and software application stack filesystem. Without a working ${HOME} and ability to start applications, the cluster cannot function. The small quota on ${HOME} is to discourage people from running jobs against ${HOME} and negatively impacting performance. The ${HOME} filesystem is specifically tuned to work well with user owned software installs and scripts, configuration files, etc., but is not a good place to store intermediate files from jobs or large data sets. Project and scratch spaces are provided and optimized for these use cases, see the Storage section for more details about what is available and the best location for different uses.

It is, however, convenient to be able to access locations through the ${HOME} path. By using symlinks, you can create your own personal namespace in ${HOME}. For example:

[john.hanks@cluster ~]$ mkdir mydata  
[john.hanks@cluster ~]$ cd mydata  
[john.hanks@cluster mydata]$ ln -s /path/to/something ./something  
[john.hanks@cluster mydata]$ ln -s /path/to/somethingelse ./somethingelse  
[john.hanks@cluster mydata]$ ls  
something somethingelse  
[john.hanks@cluster mydata]$

By using symlinks there a location was created which can be referred to in scripts as ${HOME}/mydata and has links to the actual storage locations I use, making them all easy to navigate to from ${HOME}.

!!! warning “Warning: Use a subdirectory for symlinks in ${HOME}”

Only put symlinks to other filesystems in a subdirectory of  
`${HOME}` not directly into `${HOME}`. Symlinks directly in `${HOME}` will need  
to resolve during login and if a given storage system is down, will block login.