

Table of Contents

Neutrino Computing Cluster for CERN-ND.....	1
Near Detector Computing Cluster.....	1
Connect to Near Detector Cluster.....	1
SSH information.....	1
Connect to a specific server/node.....	1
Information about SSH.....	1
Issues with SSH to the Near Detector Computer Cluster.....	1
Using ssh-agent.....	2
Computers.....	2
Storage.....	3
Accessing the NAS (neutnas00.cern.ch).....	3
Ganglia Cluster Monitor.....	3
HTCondor.....	3
Rules to use Condor.....	3
For Condor beginners.....	4
Running Processes if Condor is Running:.....	4
HTCondor useful commands.....	4
Condor FAQs.....	5
Gitlab CERN-ND repositories.....	5
Configure git.....	5
Support.....	5

Neutrino Computing Cluster for CERN-ND

Near Detector Computing Cluster

The Near Detector Computing Cluster dedicated to ND effort is located in building 3179 [↗](#), IdeaSquare [↗](#).

Connect to Near Detector Cluster

Near Detector Computer Cluster is accessible from within CERN's General Public Network. Remote access has to be done through the **cenf-nd.cern.ch** OpenStack Virtual Machine service. Users can connect to the cluster with their lxplus username and password.

SSH information

Remote access with X11 enabled: Open a terminal and SSH (Secure Socket Shell) to the VM node:

```
ssh -i .ssh/<your key> <your CERN login-id >@cenf-nd.cern.ch -X -C -t ssh <your CERN login-id >@neutplatform.cern.ch X
```

Connect to a specific server/node

It is not allowed/denied to connect directly to a specific node (and this is for security reasons)

Anyone that has a valid cern account and he/she is accepted on our cluster can ssh on the server that he/she wants, only through our VM tunneling (see below)

The information about the server names and hardware specifications can be found on an attachment below with the name Computers.txt.

Information about SSH

The protocol for the secured connection is SSH (more information about the protocol at <https://security.web.cern.ch/security/recommendations/en/ssh.shtml>).

Windows users can use PuTTY for ssh connection:

<https://espace.cern.ch/winservices-help/NICEApplications/HelpForApplications/Pages/UsingPuttyCERN.aspx> [↗](#)

Linux users can connect via ssh:

<https://twiki.cern.ch/twiki/bin/view/LinuxSupport/SSHatCERNFAQ>

Issues with SSH to the Near Detector Computer Cluster

If any user is not able to connect via SSH to the cluster go to the home directory of your lxplus account and delete the `~/.k5login` file. You will then be able to connect to the Near Detector cluster.

Using ssh-agent

If you are using a Mac, Keychains will store your ssh key passphrase so you will not have to type it each time. This way, it is really painless to remote login to any machine which has your public key. However, take care not to leave your desktop unlocked when you are away.

For linux, if you use the GUI, you will probably have ssh-agent running (type `ps -ef | grep ssh-agent` to check). Otherwise, if you ssh to sessions, you can do something similar as follows:

- copy this file `sshAgent.sh` to `$HOME/bin`.
- each time you login, or whenever you open a new terminal window, type `source sshAgent.sh`.
- add your ssh key `ssh-add $HOME/.ssh/id_dsa` and type in your passphrase.

From this point onwards, you will not be asked for the passphrase (or password !) whenever you try to login to any remote machine which has your ssh public key. Here are some useful commands:

- `ssh-add -D` to delete all the identities.
- `ssh-add -l` to list your current identities.

Note that you can also chain to an agent on your desktop/laptop computer from your agent on another computer when you do `ssh -A`

For more info on ssh follow this [\[\[https://twiki.cern.ch/twiki/bin/view/LinuxSupport/SSHatCERNFAQ\]\]](https://twiki.cern.ch/twiki/bin/view/LinuxSupport/SSHatCERNFAQ)[\[link\]](#)

Computers

The computers of the cluster are [DELL PowerEdge 1950](#).

Computer	Status	Cores	Memory(GB)	OS
neut	condor master, ganglia server	8	16	CERN CentOS 7 (CC7) ↗
neut00	condor worker	8	16	CERN CentOS 7 (CC7) ↗
neut01	condor worker	8	16	CERN CentOS 7 (CC7) ↗
neut02	condor worker	8	16	CERN CentOS 7 (CC7) ↗
neut04	condor worker	8	16	CERN CentOS 7 (CC7) ↗
neut05	condor worker	8	16	CERN CentOS 7 (CC7) ↗
neut06	condor worker	8	16	CERN CentOS 7 (CC7) ↗
neut07	condor worker	8	16	CERN CentOS 7 (CC7) ↗
neut08	condor worker	8	16	CERN CentOS 7 (CC7) ↗
neut09	condor worker	8	16	CERN CentOS 7 (CC7) ↗
neut26	condor worker	8	16	CERN CentOS 7 (CC7) ↗
neut27	condor worker	8	16	CERN CentOS 7 (CC7) ↗
neut28	condor worker	8	16	CERN CentOS 7 (CC7) ↗
neut29	condor worker	8	16	CERN CentOS 7 (CC7) ↗
neut30	condor worker	8	16	CERN CentOS 7 (CC7) ↗
neut31	condor worker	8	16	CERN CentOS 7 (CC7) ↗
neutnas00	nas	4	4	Embedded Linux

Storage

For storage there is a QNAP Turbo NAS TS-1253U with 32 TB space (*neutnas00.cern.ch*).

Accessing the NAS (*neutnas00.cern.ch*)

The users of the Near Detector cluster are categorized either as special (**S**) or normal (**N**). **S** users have a 1.5 TB quota at *neutnas00.cern.ch* and **N** users have a 100GB quota at *neutnas00.cern.ch*. On each node the users have access to their personal *neutnas00* directory through the path:

/mnt/nas00/users/USERNAME

There is also a scratch folder, which is common for all users, with 4TB storage located at the path:

/mnt/nas00/scratch

A local scratch folder can also be found on all nodes except the master

<noautolink>(neut.cern.ch)</noautolink> :

/mnt/localscratch

Each user can check the used space and quota of his personal *neutnas00* folder with the following script:

/mnt/nas00/users/check-quota.sh

```
[clezos@neut ~]$ /mnt/nas00/users/check-quota.sh
=====
Username       = 'clezos'
Available space = '100MB'
Used space     = '89M /mnt/nas00/users/clezos'
=====
[clezos@neut ~]$
```

Note: The scratch folders (**both** at *neutnas00* and the local) will be **deleted every Sunday at 09:00**.

Ganglia Cluster Monitor

The cluster load can be monitored from:

<http://cenf-nd.cern.ch/ganglia/?c=NearDetectorCluster>

HTCondor

The batch system being used by Neutrino is HTCondor. There are plenty information about condor at the official documentation. A basic guide for simple job submissions is included in this documentation. However more information about HTCondor can be found at [link1](#), [link2](#) and [link3](#)

Note: To submit a HTCondor job you must login to the ***master node** : *neut.cern.ch*

```
ssh -i .ssh/<your key> <your CERN login-id >@cenf-nd.cern.ch -X -C -t ssh <your CERN login-id >@neut.cern.ch X
```

However every user is able to connect to every worker machine and run a job locally. Remote access with X11:

```
ssh -i .ssh/<your key> <your CERN login-id >@cenf-nd.cern.ch -X -C -t ssh <your CERN login-id >@neutxx.cern.ch X
```

Rules to use Condor

The reason for the following rules is to protect the servers from heavy load which can disturb other users. Condor is a powerfull tool. Therefore we ask you to be careful and to respect the Rules. If we notice an overcharge of a server caused by condor-jobs, we have the possibility to freeze and stop your jobs.

- Let your jobs finish and then store the results. If possible, don't store intermediate results during loops in your application. The reason is again the load on the homeserver.
- A good amount of jobs is less than 200. Try to build your application, that it uses less than 200 jobs.
- If you submit more than 200 jobs, use the command: notification=never in your submit-file otherwise your mailbox is filled and our mailserv gets heavily loaded.
- The maximum amount of jobs is 2000 Jobs. Don't submit more than this amount.

For Condor beginners

- At Near Detector computer cluster, users should submit condor jobs only on neut.
 - ◆ when you log on neut with your CERN AFS account, condor environment should already be setup, you can verify this by running 'condor_status', if it does not work, send a mail to neutplatform.support@cern.ch.

Running Processes if Condor is Running:

ps auwx | grep condor :

```
condor 1303 0.0 0.0 39536 3812 ? Ss Nov10 0:06 /usr/sbin/condor_master -f
root 2612 0.0 0.0 22708 3836 ? S Nov10 3:37 condor_procd -A /var/run/condor/procd_pipe -L /var/log/condor/ProcLog -R 1000000 -S 60 -C 986
condor 2622 0.2 0.0 70436 14032 ? Ss Nov10 42:55 condor_collector -f
condor 2628 0.3 0.0 49364 13736 ? Ss Nov10 59:43 condor_negotiator -f
condor 2629 0.0 0.0 62180 4808 ? Ss Nov10 0:22 condor_schedd -f
condor 2630 0.0 0.0 40008 3800 ? Ss Nov10 0:05 condor_startd -f
```

condor_master: This program runs constantly and ensures that all other parts of Condor are running. If they hang or crash, it restarts them.

condor_collector: This program is part of the Condor central manager. It collects information about all computers in the pool as well as which users want to run jobs. It is what normally responds to the condor_status command. It's not running on your computer, but on condor.cs.wisc.edu.

condor_negotiator: This program is part of the Condor central manager. It decides what jobs should be run where. It's not running on your computer, but on condor.cs.wisc.edu.

condor_startd: If this program is running, it allows jobs to be started up on this computer--that is, hal is an "execute machine". This advertises hal to the central manager (more on that later) so that it knows about this computer. It will start up the jobs that run.

condor_schedd: If this program is running, it allows jobs to be submitted from this computer--that is, hal is a "submit machine". This will advertise jobs to the central manager so that it knows about them. It will contact a condor_startd on other execute machines for each job that needs to be started.

condor_shadow (Not shown above): For each job that has been submitted from this computer, there is one condor_shadow running. It will watch over the job as it runs remotely. In some cases it will provide some assistance (see the standard universe later.) You may or may not see any condor_shadow processes running, depending on what is happening on the computer when you try it out.

HTCondor useful commands

condor_status: List slots in HTCondor pool and their status: Owner (used by owner), Claimed (used by HTCondor), Unclaimed (available to be used by HTCondor), etc.

condor_status	explanation
Arch	Is the architecture of the processor (INTEL is Intel 32-bit, X86_64 is Intel 64-bit)
State/Activity: Owner/Idle	The machine is being used by a user, and is not available to run Condor jobs
State/Activity: Claimed/Idle	A job is assigned to this machine, but not running yet
State/Activity: Claimed/Busy	A job is assigned to this machine and is running
State/Activity: Unclaimed/Idle	The machine is ready to run jobs, but no jobs are assigned to this machine
All other	see manual on page 240

LoadAV	measure of the amount of work that a computer performs. See also: Load on Unix Systems ↗
Mem	The amount of RAM in megabytes available for a Condor Job
ActvtyTime	How long the machine is running the current state

For more Condor commands follow this link.

Condor FAQs

[CondorFAQs](#) [↗](#)

Gitlab CENF-ND repositories

- [cenf-nd/WG1](#) [↗](#)
- [cenf-nd/WG2](#) [↗](#)
- [cenf-nd/WG3](#) [↗](#)
- [cenf-nd/WG4](#) [↗](#)
- [cenf-nd/WG5](#) [↗](#)

Configure git

If you are using git for the first time, it is important to set your git name and email address which will be used by git as author for your commits (if you do not specify someone else explicitly). Failure to do this properly may cause problems when you try to push to an internal or private git repositories hosted on the CERN GitLab server. You can check that your name and email address are properly set by running

```
git config --list
```

Look for user.name and user.email . If it is not yet set, please run the following commands (which set the information for all your git repositories on lxplus):

```
git config --global user.name "Your Name"
git config --global user.email "your.name@cern.ch"
```

There are two other setting that we recommend as well:

```
git config --global push.default simple
git config --global http.postBuffer 1048576000
git config --global http.emptyAuth true # Required on CC7
```

The push setting makes some operations more straightforward. The second addresses an issue [↗](#) with large pushes via plain http or krb5. The third addresses an issue with libcurl and krb5.

For more information see the [GitLab Help at CERN](#) [↗](#)

Support

NeutrinoCluster CVMFS Deployment

For any additional information / problems / requests / suggestions please contact :
neutplatform.support@cern.ch

[↗](#) Back to the CERN Neutrino Platform-Computing Main Page

Major updates:

-- NectarB - 21-Sept-2016

This topic: CENF > NeutrinoClusterCERN

Topic revision: r42 - 2018-12-18 - NectarB



Copyright &© 2008-2020 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback