

DRAGNET System Software

All DRAGNET nodes were installed by Mike Sipior (ASTRON) with CentOS 7 using cobbler and ansible. The cobbler and ansible settings are available in a git repo on the dragnet headnode at `/var/lib/git/dragnet.git/`

Most changes have been tracked here and should ideally go into the ansible/cobbler settings git repo. However, it is unlikely going to happen (time is better spent on other tasks), so the rough notes are tracked here in case we ever have to reinstall. (Obviously, up-to-date and completeness guarantees of this list are low, but it goes a long way.)

Many system software packages have been installed, settings changed, CentOS updated to 7.2, `/opt` (+ some `/usr/local`) installed (by Alexander), while Vlad and Cees installed all pulsar user tools under `/usr/local` (NFS).

LOFAR Builds

LOFAR software builds on DRAGNET can be build+deployed and selected/activated using the scripts in that repo, viewable under <https://svn.astron.nl/viewvc/LOFAR/trunk/SubSystems/Dragnet/scripts/>

- LOFAR-Dragnet-deploy.sh (takes ~15 mins)
- LOFAR-Dragnet-activate.sh (takes 10 s)

Normally, these scripts are kicked off via [Jenkins](#). (See my slides DRAGNET - Observatory operations by Alexander (3 Jul 2017) available from the [DRAGNET wiki start page](#) for what Jenkins buttons to press. If you don't have access to Jenkins, ask Arno (LOFAR software release manager).)

As described in the scripts, these scripts can also be run from the command-line *as user lofarbuild*. You then have to manually look up the release name to be used.

Regardless of which branch or tag you build via Jenkins, the Jenkins jobs *always* svn export from the trunk!

The LOFAR package built on DRAGNET is named Dragnet, as can be seen from the `cmake` command in the `LOFAR-Dragnet-deploy.sh`. This is simply a meta-package described in the package's [CMakeLists.txt](#).

Any LOFAR build on DRAGNET depends heavily on many dependencies, the paths of which are listed in hostname matching files under <https://svn.astron.nl/viewvc/LOFAR/trunk/CMake/variants/>

We only have `variants.dragnet` (auto-selected on our headnode) and a `variants.dragproc` symlink. *This means that `cmake` runs on other nodes will fail, unless you manually add another symlink locally!* (The reason is that such builds are slow anyway, unless done from/to local disks. Prefer building on the head node (or dragproc).)

Fixing LOFAR builds is thus often a matter of small commits to the config files and/or dependent software upgrades on DRAGNET, instead of fixing the deploy script. One deploy script caveat is that it assumes all DRAGNET nodes are working...

Other Packages installed by Alexander

Many packages installed by Alexander on DRAGNET have a `/home/alexander/pkg/PKGNAME-install.txt` with commands close to a shell script used to config/build/install the package on DRAGNET. If you need to upgrade/reinstall, just copy-paste each command line by line with your brain engaged.

QPID Message Broker Config for Operations

To keep this rather complex config beast as low profile as possible on DRAGNET, this is only set up on DRAGNET to facilitate observation feedback flowing back to Observatory systems (MoM). This is inevitable (COBALT expects the local qpid queues), although failure impact is low: status in MoM.

To use [resourcetool](#), qpid is also needed, but by always specifying a broker host on the command line, we can avoid tracking RO qpid config just for that. It also makes operations vs test systems explicit (ccu001 vs ccu199).

QPID is going to be used more and more, e.g. also for user ingest.

Reinoud (and Jan David) are the people to debug qpid trouble with.

QPID Config for Feedback

On DRAGNET, I created 3 queues on each node (twice, once for operations and once for the test system), and routes from all nodes to the head node, and from the head node to ccu001 (operations) and ccu199 (test).

See `/home/amesfoort/build_qpid_queues-dragnet.sh` although typically I use it as notes instead of running it nilly-willy... RO software also has scripts where I added our queues and routes in case everything would need to be reset.

Overview on a node (1st queue with pseudo-random name is from the viewing operation itself):
[amesfoort@dragnet ~]\$ qpid-stat -q

Queues						dur	autoDel	excl	msg	msgIn
queue	msgOut	bytes	bytesIn	bytesOut	cons					
					bind					
=====										
=====										
a1fe3b70-1595-4e4d-9313-8d1706861ba0:0.0							Y	Y	0	0
0	0	0	0		1	2				
lofar.task.feedback.dataproducts							Y		0	11.4k
11.4k	0	39.1m	39.1m		1	1				
lofar.task.feedback.processing							Y		0	0
0	0	0	0		1	1				
lofar.task.feedback.state							Y		0	0
0	0	0	0		1	1				
test.lofar.task.feedback.dataproducts							Y		0	61

61	0	185k	185k	1	1			
	test.lofar.task.feedback.processing					Y	0	0
0	0	0	0	1	1			
	test.lofar.task.feedback.state					Y	0	0
0	0	0	0	1	1			

Overview of all routes to dragnet (6 per node):

```
[amesfoort@dragnet ~]$ qpid-route route list
dragnet:5672 dragproc.control.lofar:5672
dragnet:5672 dragproc.control.lofar:5672
dragnet:5672 dragproc.control.lofar:5672
dragnet:5672 dragproc.control.lofar:5672
dragnet:5672 dragproc.control.lofar:5672
dragnet:5672 dragproc.control.lofar:5672
dragnet:5672 drg01.control.lofar:5672
dragnet:5672 drg01.control.lofar:5672
dragnet:5672 drg01.control.lofar:5672
dragnet:5672 drg01.control.lofar:5672
dragnet:5672 drg01.control.lofar:5672
dragnet:5672 drg01.control.lofar:5672
dragnet:5672 drg02.control.lofar:5672
[...]
dragnet:5672 drg22.control.lofar:5672
dragnet:5672 drg23.control.lofar:5672
dragnet:5672 drg23.control.lofar:5672
dragnet:5672 drg23.control.lofar:5672
dragnet:5672 drg23.control.lofar:5672
dragnet:5672 drg23.control.lofar:5672
dragnet:5672 drg23.control.lofar:5672
```

System Config Changes

On top of git repo with ansible/cobbler settings:

Crontab

casacore measures tables

On host dragnet (the script downloads once, then applies the update on all nodes), run the command every Mon 04:00 AM.

This auto-updates the casacore measures tables with info on observatories, solar bodies, leap seconds, int'l earth rotation (IERS) coefficients, etc.

```
[amesfoort@dragnet ~]$ sudo crontab -u lofarsys -l
0 4 * * 1 /opt/IERS/cron-update-IERS-DRAGNET.sh 2>
/home/lofarsys/lofar/var/log/IERS/cron-update-IERS-DRAGNET.log
```

resourcetool

On any host but dragnet (it has no RADB resources), run the [resourcetool](#) command with the -E and possibly -U option(s) every 20 mins starting 1 min past the hour.

This auto-updates storage claim end times in the Observatory's RADB. Else, Observatory systems will eventually think our disks are full and scheduling observations becomes impossible. But we manage disk space ourselves. (The tool also has some other useful capabilities.)

```
[amesfoort@any_but_dragnet ~]$ sudo crontab -u lofarsys -l
1,21,41 * * * * source /opt/lofar/lofarinit.sh; LOFARENV=PRODUCTION
/opt/lofar/bin/resourcetool --broker=scu001.control.lofar --end-past-tasks-
storage-claims > /home/lofarsys/lofar/var/log/resourcetool/cron-update-
resourcetool-$(hostname).log 2>&1
```

/etc

Apply /home/amesfoort/etc/* to /etc/

Other

```
newgrp dragnet
umask 0002
chmod 775 /opt
sudo chgrp dragnet /opt # and set setgid bit
/opt/lofar_versions owned by lofarbuild:dragnet mode 775 (or rely on setgid
bit)
sudo chgrp -R dragnet /usr/local/share/aclocal /usr/local/share/applications
/usr/local/share/info /usr/local/share/man # applications/ subdir needed to
install aoflagger as dragnet under /usr/local/
exit
```

as well as for the /data1 and /data2 dirs (/data on dragproc seemed fine when I looked, but maybe others did that manually) (and find out what the '+' is in ls -al)

Install pkgs:

```
lsof
environment-modules
smartmontools
numactl-devel (deps on numactl-libs)
hwloc
hwloc-devel
binutils-devel
atop
htop
strace
ftp
```

```
tcpdump
#libnet-devel # for custom arping to ping by MAC
#libpcap-devel # idem
iperf3
nethogs
erfa-devel
armadillo-devel
python-astropy
python-jinja2 # for the FACTOR imaging pipeline module
python-daemon
python-matplotlib-qt4
python-psycopg2 mysql-connector-python PyGreSQL # LOFAR mysql, postgresql
DB python interface modules (used for self-tests only?)
python2-mock # for python LOFAR self-tests under SAS/ and elsewhere
qpid-cpp-server-linearstore (add to qpid pkgs)
patch
elfutils
deltarpm
NetworkManager-config-routing-rules # for policy based routing using
NetworkManager
libgtkmm-2.4-dev libsigc++-2.0-dev # optional; for AOFlagger's rficonsole
dbus-c++-devel # required for awimager2's near-copy of CASA libsynthesis
openblas-devel # required for sagecal
pyfits # required for rmsynthesis
ds9 # required by ds9
geom # required by Shapely python module, for the FACTOR pipeline
progressbar # for losoto (LOfar SOlutions TOol,
https://github.com/revoltek/losoto)
xorg-x11-server-Xvfb # for LOTAAS pipeline
mercurial
vim-X11
colordiff
ddd

# for slurm (NOTE: -devel pkgs only needed on head node to build RPMs; non-
devel needed on other nodes)
munge
munge-devel
readline-devel
pam-devel
lua-devel
mailx # also for robinhood
man2html
freeipmi-devel
json-c-devel
rrdtool-devel
libibmad-devel
libibumad-devel
perl-Switch # to install created slurm RPMs
perl-DBI # idem
```

```
# to create slurm and lustre client RPMs, head node only:
rpm-build
perl-ExtUtils-MakeMaker

# ensure not specifically installed:
libpng12-devel # since we use libpng-devel (implied)

On drg nodes (ib tools)
libmlx4
libibverbs-utils
libibverbs
perftest
qperf

libibverbs-devel # on head node is enough
librdmacm-devel # idem
mstflint # idem

# Python packages N/A in CentOS package manager; use pip install
python-monetdb # for LOFAR GSM (imaging); on the head node we did: sudo
pip install --target=/usr/local/lib/python2.7/site-packages python-monetdb
xmlrunner # for LOFAR Pipeline tests; on the head node we did: sudo
pip install --target=/usr/local/lib/python2.7/site-packages xmlrunner

/etc/yum/pluginconf.d/fastestmirror.conf
enabled=0
(into ansible)

dragnet node: time is no longer in UTC (fixed; but check ansible)

dragnet node:
sudo systemctl enable nfs-server
sudo systemctl start nfs-server

dragnet node:
dd if=/dev/urandom bs=1 count=1024 >/etc/munge/munge.key (set owner=munge
group=munge mode=400)
other nodes:
copy this file to local /etc/munge/
each host:
sudo systemctl enable munge
sudo systemctl start munge

add slurm system user and group
create slurm RPMs on hostname
install RPMs on all hosts
copy slurm.conf and gres.conf to all hosts # RPM install creates
/etc/slurm/
create /var/spool/slurmd/
```

```

dragnet node:
sudo systemctl enable slurmctld slurmd
sudo systemctl start slurmctld slurmd
dragproc:
sudo systemctl enable slurmctld slurmd slurmdbd
sudo systemctl start slurmctld slurmd slurmdbd
drg:
sudo systemctl enable slurmd
sudo systemctl start slurmd

set GPUs in persistence mode before slurmd starts (on drgXX nodes only): Use
my ~amesfoort/nvidia-smi-pm.service copied into /usr/lib/systemd/system/ and
then run:
sudo systemctl daemon-reload
sudo systemctl enable nvidia-smi-pm

check networking/interface settings in ansible, and in
/etc/sysconfig/network-scripts. Use system-config-network tool to edit.
ONBOOT=no too often. MTU 9000 for 10G i/f, ib netmask must be /16 (not /13
as it clashes with cep2 routes), etc...
add routes to CEP2 (and others). Use dragnet-node-routes-10g.sh
Set CONNECTED_MODE=Yes See /home/alexander/Downloads/linux-
kernel/linux-3.10.85/Documentation/infiniband/ipoib.txt
Why the heck is this route in drg* routing tables??? (Useful for virt 10G/ib
netw?): 169.254.0.0/16 dev ib0 scope link metric 1005
Has Mike fixed the (cobbler?) routing issue via portal? Must go via PD-0
(xxx.5 -> .6 or vice versa)
- Add ping test script: also useful to see what hostnames/domainnames should
work.
And fix the idiotic domainname crap!!!

systemctl enable NetworkManager-dispatcher.service
systemctl start NetworkManager-dispatcher.service

Correct table example drgl6 (except that CEP2 routes and sub-tables can now
be removed):
[amesfoort@drgl6 network-scripts]$ ip ru
0: from all lookup local
1000: from 10.168.145.1 lookup 1
32766: from all lookup main
32767: from all lookup default
[amesfoort@drgl6 ~]$ ip r l t 1
10.135.252.0/24 via 10.175.255.201 dev ens5 proto static
10.135.253.0/24 via 10.175.255.202 dev ens5 proto static
10.135.254.0/24 via 10.175.255.203 dev ens5 proto static
10.135.255.0/24 via 10.175.255.204 dev ens5 proto static
[amesfoort@drgl6 network-scripts]$ ip r
default via 10.151.255.254 dev em1 proto static metric 100
10.134.224.0/19 dev ib0 proto kernel scope link src 10.134.224.18 metric
150
10.144.0.0/13 dev em1 proto kernel scope link src 10.149.160.18 metric

```

```
100
10.168.0.0/13 dev ens5 proto kernel scope link src 10.168.145.1 metric
100
10.176.0.0/13 via 10.175.255.254 dev ens5 proto static

cexec (C3 Cluster Command & Control Suite) into ansible, incl /etc/c3.conf
symlink to /usr/local/etc/c3.conf
casacore + casacore-python + measures_tables
LOFAR build
lofardal build
/mnt (or /net (/net N/A on debian)) automounts

lofarbuild jenkins pub key (head node) into ansible (separate task, since it
may change)
lofarsys ssh authorized ssh key (?)
LOFAR settings: lofarsys sudo RT, shmem, ptrace, max CPU(+GPU?) clock when
observing(?), RLIMIT_MEMLOCK (also for ibverbs), ... (see
c3t009:/etc/rc.local.d/)

add LOFAR/trunk/RTCP/Cobalt/OutputProc/etc/sudoers.d/setcap_cobalt to
/etc/sudoers.d/ and ensure it's included via /etc/sudoers (it seems Mike
changed sudoers on dragnet, but not elsewhere, now equal, but not yet in
ansible)

lofarsys is NFS account; fix this (note: fix up ssh login failures, since
keys are then no longer accessible)
lofarsys: ensure dirs exist on all nodes (local account):
lofar/var/{run,log}

lofarsys: ~/.ssh/config:
-----
NoHostAuthenticationForLocalhost yes

Host dragnet dragnet.control.lofar dragproc dragproc-10g
dragproc.control.lofar dragproc-10g.online.lofar drg?? drg??.control.lofar
drg??-10g drg??-10g.online.lofar drg??-ib drg??-ib.dragnet.infiniband.lofar
StrictHostKeyChecking no
-----

qpid script to create local queues once

CUDA pkgs on *all* nodes after adding NVIDIA 'cuda' repo: (we use the
'elrepo' driver and the 'cuda' cuda pkgs)
cuda-repo-rhel7
Note: the following 2 pkgs + deps will go into /usr/local, while we want
them into /opt, so ask Mike which rpm he used instead
cuda-toolkit-7-0 # if still needed
cuda-toolkit-7-5

install pkgs from ~/pkg such as log4cplus, ...
```



```
add changed /etc/modulefiles/* to ansible

/etc/security/limits.conf:
set 'nofile' soft limit to 4k (hard was requested to 10k, but meaningless?)

----- Remove on all nodes:
add '--auth no' to allnodes:/usr/lib/systemd/system/qpidd.service
(only needed on the src side of the g, but since we need to fwd, it's needed
on all our nodes)
ExecStart=/usr/sbin/qpidd --config /etc/qpidd/qpidd.conf --auth no
----- Replace by on all nodes (qpidd.service is replaced on pkg update):
Add auth=no in /etc/qpidd/qpidd.conf
-----
Then run: (maybe there's a systemctl command to do both of this in one go?)
sudo systemctl daemon-reload
sudo systemctl enable qpidd
sudo systemctl restart qpidd
(& check if systemctl enable qpidd (and start qpidd) are indeed in ansible)

added routing table entries for drg*, dragproc in ansible

-----
for lustre mount cep4 (drg nodes only (need ib atm), further install by hand
atm (need rpm rebuild from src rpm)). On all drgXX nodes:
# create /etc/modprobe.d/lnet.conf with:
options lnet networks=o2ib(ib0)

# create/adjust /etc/modprobe.d/ko2iblnd.conf with:
#comment out any 'alias' and 'options' lines other than the next (which MUST
match the settings on the Lustre MGS (and thus all other clients as well)):
options ko2iblnd peer_credits=128 peer_credits_hiw=64 credits=2048
concurrent_sends=256 ntx=2048 map_on_demand=32 fmr_pool_size=2048
fmr_flush_trigger=512 fmr_cache=1
# optional:
install ko2iblnd /usr/sbin/ko2iblnd-probe

# create mount point as root:
mkdir -p /cep4data

# append to /etc/fstab
meta01.cep4.infiniband.lofar@o2ib:meta02.cep4.infiniband.lofar@o2ib:/cep4-fs
/cep4data lustre defaults,ro,flock,noauto 0 0

-----

LTA stuff (may turn out unnecessary when we can finally ingest from DRAGNET,
since it'll have to go via lexar nodes)

globus-gass-copy-progs
voms-clients-cpp
```

```
#voms-clients-java # not needed

wget
http://repository.egi.eu/sw/production/cas/1/current/repo-files/EGI-trustanc
hors.repo
sudo mv EGI-trustanchors.repo /etc/yum.repos.d/
(currently on dragnet, it's called egi.repo)
-----
[EGI-trustanchors]
name=EGI-trustanchors
baseurl=http://repository.egi.eu/sw/production/cas/1/current/
gpgkey=http://repository.egi.eu/sw/production/cas/1/GPG-KEY-EUGridPMA-RPM-3
gpgcheck=1
enabled=1
-----

ca-policy-egi-core # from EGI-trustanchors repo

wget
http://www.lofar.org/operations/lib/exe/fetch.php?media=public:srmclient-2.6
.28.tar.gz
wget
http://www.lofar.org/wiki/lib/exe/fetch.php?media=public:lta-url-copy.sh.gz

(already set up SRM module file)

/etc/vomses, $HOME/.voms/vomses, $HOME/.glite/vomses any filename, e.g.
lofar-voms
-----
"lofar" "voms.grid.sara.nl" "30019"
"/O=dutchgrid/O=hosts/OU=sara.nl/CN=voms.grid.sara.nl" "lofar"
-----

/etc/grid-security/vomsdir/lofar/voms.grid.sara.nl.lsc
-----
/O=dutchgrid/O=hosts/OU=sara.nl/CN=voms.grid.sara.nl
/C=NL/O=NIKHEF/CN=NIKHEF medium-security certification auth
-----
```

From:
<https://www.astron.nl/lofarwiki/> - **LOFAR Wiki**

Permanent link:
https://www.astron.nl/lofarwiki/doku.php?id=dragnet:system_software&rev=1503018249

Last update: **2017-08-18 01:04**

