UMass CS Homebrew HPC Guide

From CIIR System Administration

Contents

1 Function
2 Components
   2.1 CentOS 7
   2.2 TFTP
   2.3 DHCP
   2.4 DNS
   2.5 Apache
   2.6 Kickstart
   2.7 NFS
   2.8 Firewalld
   2.9 NIS
   2.10 NTP
   2.11 Ansible
   2.12 Slurm
   2.13 Ganglia
3 Common Tasks
   3.1 Running a Command on All Nodes
   3.2 Adding a New User
   3.3 Deleting a User
   3.4 Adding a New Node
   3.5 Add Node Key on Head Node
4 Old Documentation
   4.1 LDAP

Function

Components

CentOS 7

Installed from the CentOS 7 minimal ISO (build 1611).

```bash
# yum -y update
# yum group install "Development Tools" "System Administration Tools"
# yum install epel-release nfs-utils vim wget
```

Disable SELinux:

```bash
# vi /etc/sysconfig/selinux
Change SELINUX=Enforcing to SELINUX=disabled
```

TFTP

```bash
# yum install syslinux tftp-server
# mkdir /var/lib/tftpboot/pxelinux.cfg
# cp /usr/share/syslinux/pxelinux.0 /var/lib/tftpboot
# cp /usr/share/syslinux/chain.c32 /var/lib/tftpboot
# cp /usr/share/syslinux/menu.c32 /var/lib/tftpboot
# systemctl enable tftp
# systemctl start tftp
```

/var/lib/tftpboot/pxelinux.cfg/default:

```
Default menu.c32
prompt 0
timeout 100

Menu title PXE Boot Menu
Label 1
   menu default
   menu label "2" Install CentOS 7
   kernel initrd=initrd.img load_ramdisk=1 network ks=http://10.96.1.1/kickstart/compute.ks
Label 2
   menu label "3" Boot from local drive
   COM32 chain.c32
   APP00 0
```
Download a CentOS ISO, mount it and grab the kernel and initrd boot files:

```
# wget http://mirrors.mit.edu/centos/7/isos/x86_64/CentOS-7-x86_64-Minimal-1611.iso -O /tmp/CentOS-7-x86_64-Minimal-1611.iso
# mount -o loop,ro /tmp/CentOS-7-x86_64-Minimal-1611.iso /mnt
# cp /mnt/images/pxeboot/vmlinuz /var/lib/tftpboot/
# cp /mnt/images/pxeboot/initrd.img /var/lib/tftpboot/
# umount /mnt
```

**DHCP**

```
# yum install dhcp
# systemctl enable dhcpd.service

Edit /etc/dhcp/dhcpd.conf. This is a sample configuration with one node:

```
# dhcpd.conf
#
local-address 10.96.1.1;
option domain-name "dali";
option domain-name-servers 10.96.1.1;
default-lease-time 600;
max-lease-time 7200;
log-facility local7;
subnet 10.96.0.0 netmask 255.255.0.0 {  
  authoritative;
  allow booting;
  option routers 10.96.1.1;
  option subnet-mask 255.255.0.0;
  option ntp-servers 10.96.1.1;
  host compute-0-0 {  
    hardware ethernet D8:D3:85:EB:F7:D2;
    fixed-address 10.96.0.254;
    option host-name "compute-0-0";
    next-server 10.96.1.1;
    filename "pxelinux.0";
  }
}
```

```
# systemctl start dhcpd.service
```

**DNS**

```
# yum install bind bind-utils
# systemctl enable named.service

Edit /etc/named.conf to listen on the internal network interface and allow queries from the cluster nodes:

```
# named.conf
#
listen-on port 53 { 127.0.0.1; 10.96.1.1; }
allow-query { 10.96/16; 127.0.0.1; }
dnssec-validation no;
include "/etc/named/named.conf.local";
```

```
/etc/named/named.conf.local:

# named.conf.local
#
zone "dali" {  
  type master;
  file "/etc/named/zones/db.dali";
};
zone "96.10.in-addr.arpa" {  
  type master;
  file "/etc/named/zones/db.10.96";
};
```

Sample zone files for .dali and 10.96:

```
/etc/named/zones/db.dali:

IN NS nsl.dali.  
2817042700 ; Serial  
684800 ; Refresh  
86400 ; Retry  
2419200 ; Expire  
684800 ; Negative Cache TTL
IN SOA nsl.dali. admin.dali. (  
2817042700 ; Serial  
684800 ; Refresh  
86400 ; Retry  
2419200 ; Expire  
684800 ; Negative Cache TTL
)

dali.dali. IN A 10.96.1.1  
nsl.dali. IN A 10.96.1.1  
compute-0-0 IN A 10.96.0.254
```
/etc/named/zones/db.10.96:

```
$IN   SOA  ns1.dali.  admin.dali.  ( 2017042700 ; Serial
                           604800 ; Refresh
                           86400 ; Retry
                           2419200 ; Expire
                           604800 ) ; Negative Cache TTL

IN   NS  ns1.dali.

.1     IN   PTR  dali.dali.

254.0  IN   PTR  compute-0-0.dali.
```

# systemctl start named.service

Turn off NetworkManager on the network interfaces, typically a line in /etc/sysconfig/network-scripts/ifcfg-* like:

Update: Turning off NetworkManager in the ifcfg-* scripts prevents the firewall zone from being loaded during boot. I have turned on NetworkManager on blake2 and everything seems to be okay. DNS should still be disabled in NetworkManager.conf as described below.

```
# NM_CONTROLLED=no

Edit /etc/NetworkManager/NetworkManager.conf:

```
[main]
dns=none
```

Restart NetworkManager. Manually configure /etc/resolv.conf:

```
# resolv.conf
# search dali.cs.umass.edu
nameserver 127.0.0.1
nameserver 128.119.240.1
nameserver 128.119.40.12
```

Apache

```
# yum install httpd
# mount -o loop,ro /tmp/CentOS-7-x86_64-Minimal-1611.iso /mnt
# cp -r /mnt /var/www/html/centos-7-install
# umount /mnt
# systemctl enable httpd.service
# systemctl start httpd.service
```

Kickstart

```
# mkdir /var/www/html/kickstart

Edit /var/www/html/kickstart/compute.ks:

```
# compute.ks
#
# text
install
url --url http://10.96.1.1/centos-7-install/
lang en_US.UTF-8
keyboard --vckeymap=us --xlayouts='us'
#network --device eno1 --onboot yes --bootproto dhcp --hostname "$HOSTNAME"
%include /tmp/network.ks
rootpw --iscrypted $6$rj5K1Zb0bTCSCFr6$XHPrRVV7oMNZcrP6jCulNeqtlzNkjq7hsOJAw8QArXMzIyo2xT8LCNlu7HlqjQnaJH8EsdPPy2kcVz90MhjJS.
firewall --disabled
authconfig --enableshadow --passalgo=sha512
selinux --disabled
timezone America/New_York --isUtc
eula --agreed
ignoredisk --only-use=sd
bootloader --location=mbr --boot-drive=sda
zerombr
clearpart --all --initlabel --drives=sda
part / --fstype="ext4" --ondisk=sda --size=30517
part swap --fstype="swap" --ondisk=sda --size=15258
reboot
%packages --ignoremissing
@Core
%end
%pre
echo "network --device enal --onboot yes --bootproto dhcp --hostname "SHOSTNAME"" > /tmp/network.ksc
%post
#cat /root/.ssh
#cat /etc/NetworkManager/NetworkManager.conf
```

3 of 9 06/20/2018 04:34 PM
NFS

# yum install nfs-utils autofs
# systemctl enable nfs.service
# systemctl enable autofs.service
# mkdir /share

Edit /etc/exports:

# exports
#
/home  10.96.0.0/255.255.0.0(rw,async,no_root_squash)
/share  10.96.0.0/255.255.0.0(rw,async,no_root_squash)

# systemctl start nfs.service
# systemctl start autofs.service

Firewalld

Firewalld has a decent GUI for configuration. Make sure X connections are forwarded and then run:

# yum install firewall-config xorg-x11-xauth urw-fonts
# firewall-config

Firewalld comes with some predefined zones. The interface connecting to the Internet should be added to the public zone while the private cluster interface should be in the trusted zone. In the public zone, go to the Masquerading tab and check the Masquerade zone box.

For the trusted zone I basically checked off all services – we completely trust connections to the head node from the compute nodes.

NIS

User and group info were originally stored in LDAP, but I couldn’t figure out how to do that and still authenticate to an external LDAP server. By switching to NIS it is easy to authenticate to CSCF’s LDAP server.

# yum install ypserv tokyocabinet nss-pam-ldapd

Create the file /var/yp/securenets with the appropriate internal cluster network:

# securenets
#
255.0.0.0        127.0.0.0
255.255.0.0      10.32.0.0

Establish the NIS domain, start the NIS servers and initialize the database:

# echo "NISDOMAIN=dali" >> /etc/sysconfig/network
# systemctl start ypserv.service ypxfrd.service yppasswrd.service
# /usr/sbin/ypinit -m

Edit /etc/nsswitch.conf to use NIS for at least these lines:

passwd:     files sss nis
shadow:     files sss nis
group:      files sss nis
netgroup:   files sss nis

After adding accounts or groups make sure to update the maps:

# make -C /var/yp

Note that by default NIS will not push out users or groups with UIDs/GIDs less than 1000. This can be changed in /etc/login.defs.

Edit /etc/nslcd.conf to use CSCF’s LDAP server:

uid nslcd
gid ldap
uri ldap://squall.cs.umass.edu/
base dc=cs,dc=umass,dc=edu
bindn cn=cg_ciir,cn=computer_groups,dc=cs,dc=umass,dc=edu
bindp cscfauth
Make sure you have a copy of CSCF's SSL certificates in /etc/openldap/cacerts.

```bash
# chmod 600 /etc/nslcd.conf
# systemctl enable nslcd.service
# systemctl start nslcd.service
# authconfig --enableldap --enableldapauth --enablelocalauth --ldapserver=squall.cs.umass.edu --ldapbasedn=dc=cs,dc=umass,dc=edu --updateall
```

**NTP**

```bash
# systemctl enable chronyd.service
# systemctl start chronyd.service
```

**Ansible**

This is where the bulk of the configuration of the cluster nodes goes. Kickstart partitions the disk and provides a barebones OS install. Everything else should be done from here.

```bash
# yum install ansible
```

There are a lot of different ways Ansible can be configured to build the cluster nodes. As a rough guideline, I'm defining groups in /etc/ansible/hosts based on the node type:

```bash
# hosts
#
[compute]
compute-0-0
compute-0-1
compute-0-2
[gp]u
gpu-0-0
gpu-0-1
[nas]
nas-0-0
nas-0-1
nas-0-2
```

Playbooks are created in /etc/ansible/playbooks and use a node type (for identification) and number (for execution order) as part of the name:

```bash
# ls -l /etc/ansible/playbooks
compute-00-packages.yml
compute-01-autofs.yml
compute-02-sssd.yml
compute-30-slurm.yml
compute-40-ganglia.yml
compute-50-services.yml
yum-update.yml
```

Files that are part of playbooks are stored in /etc/ansible/files and use the format TYPE-filename for easy identification. For example, compute-auto.home will be copied to all compute nodes as auto.home.

```bash
# ls -l /etc/ansible/files
compute-auto.home
compute-auto.master
compute-auto.share
compute-gmond.conf
compute-munge.key
compute-slurm
compute-slurm.conf
```

**Slurm**

Most of this is cribbed from [http://sysadm.mielnet.pl/building-and-installing-rpm-slurm-on-centos-7/](http://sysadm.mielnet.pl/building-and-installing-rpm-slurm-on-centos-7/)

```bash
# yum install munge munge-libs munge-devel readline-devel openssl-devel perl-ExtUtils-MakeMaker perl-DBI perl-Switch
# wget https://www.schedmd.com/downloads/archive/slurm-16.05.10-2.tar.bz2
# rpmbuild -ta slurm-16.05.10-2.tar.bz2
# cd /root/rpmbuild/RPMS/x86_64
# rpm -ivh *
# mv * /home/RPMS
```

06/20/2018 04:34 PM
Add the slurm user and group, create a couple of directories:

```bash
# groupadd -g 450 slurm
# useradd -u 450 -g 450 slurm
# mkdir /var/log/slurm
# chown slurm:slurm /var/log/slurm
# mkdir /var/lib/slurm
# chown slurm:slurm /var/lib/slurm
```

Create the database:

```bash
# systemctl enable mariadb
# systemctl start mariadb
# mysql_secure_installation
# mysql
    mysql> GRANT ALL ON slurm_acct_db.* TO 'slurm'@'localhost' IDENTIFIED BY 'some_pass' WITH GRANT OPTION;
    mysql> CREATE DATABASE slurm_acct_db;
```

Create `/etc/slurm/slurmdbd.conf`:

```bash
 AuthType=auth/munge
 DbdAddr=localhost
 DbdHost=localhost
 SlurmUser=slurm
 DebugLevel=4
 LogFile=/var/log/slurm/slurmdbd.log
 PidFile=/var/run/slurmdbd.pid
 StorageType=accounting_storage/mysql
 StorageHost=localhost
 StoragePass=some_pass
 StorageUser=slurm
 StorageLoc=slurm_acct_db
```

Create the Munge key:

```bash
# create-munge-key
# systemctl enable munge
# systemctl start munge
# systemctl status munge
```

Create the Slurm configuration file:

```bash
# Need sample here...
# systemctl enable slurm
# systemctl start slurm
# systemctl status slurm
```

Add the cluster to Slurm:

```bash
# sacctmgr add cluster <name>
```

**Ganglia**

```bash
# yum install ganglia ganglia-web ganglia-gmond ganglia-gmetad
```

Edit `/etc/ganglia/gmond.conf` with info about the cluster:

```bash
name = "Dali"
host = dali.dali
```

Edit `/etc/ganglia/gmetad.conf` with info about the cluster:

```bash
gridname "Dali"
# These settings were brought over from the Rocks configuration of Ganglia
RRAs "RRA:AVERAGE:0.5:1:2440"
    "RRA:AVERAGE:0.5:4:1440"
    "RRA:AVERAGE:0.5:17:2440"
    "RRA:AVERAGE:0.5:67:2440"
    "RRA:AVERAGE:0.5:576:3740"
```

# systemctl enable gmond.service
Common Tasks

Running a Command on All Nodes

Use the parallel shell (pdsh). Examples:

```
# pdsh -w compute-0-[0-3] uptime
```

If the `pdsh-mod-dshgroup` package is installed, you can define groups in `/etc/dsh/group` and refer to them:

```
# pdsh -g all uptime
# pdsh -g compute uptime
# pdsh -g gpu uptime
```

Adding a New User

Deleting a User

Adding a New Node

Add Node Key on Head Node

When logging into a new (or rebuilt) node for the first time the node’s host key can be added directly to `/etc/ssh/ssh_known_hosts` with:

```
# ssh -o UserKnownHostsFile=/etc/ssh/ssh_known_hosts <node name>
```

Old Documentation

LDAP

Account and group information is now stored in NIS. This is the documentation for storing that information in LDAP.

These example are for a cluster with the hostname of ‘dalihn’ and the domain name of ‘dali’. Most of this was cribbed from http://www.itzgeek.com/how-tos/linux/centos-how-tos/step-step-openldap-server-configuration-centos-7-rhel-7.html

```
# yum install openldap openldap-servers openldap-clients
# systemctl enable slapd.service
# systemctl start slapd.service
```

Create a password hash for the LDAP administrator. Do not use this example:

```
# slappasswd
New password: ChangeMe2017
Re-enter new password: ChangeMe2017
{SSHA}Jrkb6JB7uA1HAlVF3ludrL4z+bbiSFGK
```

Create `db.ldif`:

```
dn: olcDatabase={2}hdb,cn=config
changetype: modify
replace: olcSuffix
olcSuffix: dc=dali

dn: olcDatabase={2}hdb,cn=config
changetype: modify
replace: olcRootDN
olcRootDN: cn=ldapadm,dc=dali

dn: olcDatabase={2}hdb,cn=config
changetype: modify
replace: olcRootPW
olcRootPW: {SSHA}Jrkb6JB7uA1HAlVF3ludrL4z+bbiSFGK
```

Load the configuration into the LDAP server:

```
# ldapmodify -Y EXTERNAL -H ldapi:/// -f db.ldif
```

Create a `monitor.ldif` file to limit monitor access to the LDAP administrator account:

```
dn: olcDatabase={1}monitor,cn=config
changetype: modify
replace: olcRootPW
olcRootPW: {SSHA}Jrkb6JB7uA1HAlVF3ludrL4z+bbiSFGK
```

Load the configuration into the LDAP server:

```
# ldapmodify -Y EXTERNAL -H ldapi:/// -f monitor.ldif
```

Create a monitor.ldif file to limit monitor access to the LDAP administrator account:
Create an SSL certificate:

```sh
# openssl req -new -x509 -nodes -out /etc/openldap/certs/dalihn-crt.pem -keyout /etc/openldap/certs/dalihn-key.pem -days 3650
# chown ldap:ldap /etc/openldap/certs/dalihn-crt.pem /etc/openldap/certs/dalihn-key.pem
# chmod 640 /etc/openldap/certs/dalihn-key.pem
```

Copy the sample database configuration file to `/var/lib/ldap` and update the file permissions:

```sh
# cp /usr/share/openldap-servers/DB_CONFIG.example /var/lib/ldap/DB_CONFIG
# chown ldap:ldap /var/lib/ldap/DB_CONFIG
```

Add some LDAP schemas:

```sh
# ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/openldap/schema/cosine.ldif
# ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/openldap/schema/nis.ldif
# ldapadd -Y EXTERNAL -H ldapi:/// -f /etc/openldap/schema/inetorgperson.ldif
```

Create the base.ldif for the domain:

```sh
# ldapadd -x -W -D "cn=ldapadm,dc=dali" -f base.ldif
```

Here is a sample .ldif file for adding a new user:

```sh
# drp.ldif

# cn=drp,ou=People,dc=dali
objectClass: top
objectClass: account
objectClass: posixAccount
objectClass: shadowAccount

uid: drp
uidNumber: 25480
gidNumber: 100
homeDirectory: /home/drp
loginShell: /bin/bash
gecos: Daniel R. Parker
userPassword: {crypt}x
shadowLastChange: 17058
shadowMin: 0
shadowMax: 99999
shadowWarning: 7
```
# ldapadd -x -W -D "cn=ldapadm,dc=dali" -f drp.ldif

Edit `/etc/openldap/ldap.conf` to point to itself as the LDAP server:

```
TLS_CACERTDIR /etc/openldap/cacerts
SASL_NOCANON on
URI ldap://127.0.0.1/
BASE dc=dali
```

Edit `/etc/nslcd.conf` to use the LDAP server:

```
uri ldap://127.0.0.1/
base dc=dali
rootpwmoddn cn=ldapadm,dc=dali
```

Edit `/etc/openldap/slapd.d/cn=config/olcDatabase={2}hdb.ldif` to allow users to change their password:

```
1. Append these lines to the bottom...

olcAccess: {0}to attrs=userPassword by self write by dn.base="cn=ldapadm,dc=dali" write by * none olcAccess: {1}to * by
   dn.base="cn=ldapadm,dc=dali" write by self write by * read
```

Configure rsyslog to log LDAP events to a separate file. Edit `<code>/etc/rsyslog.conf` and add the line:

```
local4.* /var/log/ldap.log
```

Restart the rsyslog service.


- This page was last modified on 23 May 2018, at 15:16.
- This page has been accessed 63 times.