LiCO 6.1.0
Installation Guide (for EL8)

Eighth Edition (December 2020)

Reading instructions

- To ensure that you get correct command lines using the copy/paste function, open this Guide with Adobe Acrobat Reader, a free PDF viewer. You can download it from the official Web site https://get.adobe.com/reader/.
- Replace values in angle brackets with the actual values. For example, when you see <*_USERNAME> and <*_PASSWORD>, enter your actual username and password.
- Between the command lines and in the configuration files, ignore all annotations starting with “#”.

Contents

Chapter 1. Overview

Introduction to LiCO

Lenovo Intelligent Computing Orchestration (LiCO) is an infrastructure management software for high-performance computing (HPC) and artificial intelligence (AI). It provides features like cluster management and monitoring, job scheduling and management, cluster user management, account management, and file system management.

With LiCO, users can centralize resource allocation in one supercomputing cluster and carry out HPC and AI jobs simultaneously. Users can perform operations by logging in to the management system interface with a browser, or by using command lines after logging in to a cluster login node with another Linux shell.

Typical cluster deployment

This Guide is based on the typical cluster deployment that contains management, login, and compute nodes.

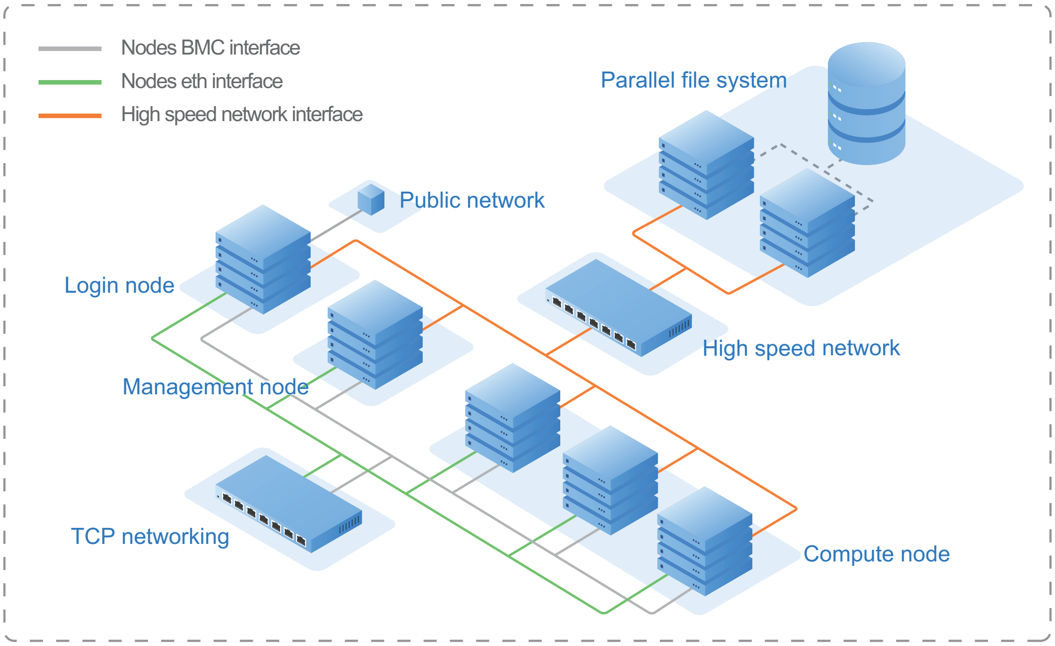


Figure 1. Typical cluster deployment

Elements in the cluster are described in the table below.

Table 1. Description of elements in the typical cluster

Element	Description
Management node	Core of the HPC/AI cluster, undertaking primary functions such as cluster management, monitoring, scheduling, strategy management, and user & account management.
Compute node	Completes computing tasks.
Login node	Connects the cluster to the external network or cluster. Users must use the login node to log in and upload application data, develop compilers, and submit scheduled tasks.
Parallel file system	Provides a shared storage function. It is connected to the cluster nodes through a high-speed network. Parallel file system setup is beyond the scope of this Guide. A simple NFS setup is used instead.
Nodes BMC interface	Used to access the node's BMC system.
Nodes eth interface	Used to manage nodes in cluster. It can also be used to transfer computing data.
High speed network interface	Optional. Used to support the parallel file system. It can also be used to transfer computing data.

Note: LiCO also supports the cluster deployment that only contains the management and compute nodes. In this case, all LiCO modules installed on the login node need to be installed on the management node.

Operating environment

Cluster server:

Lenovo ThinkSystem servers

Operating system:

CentOS / Red Hat Enterprise Linux (RHEL) 8.2

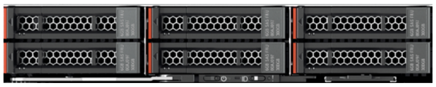
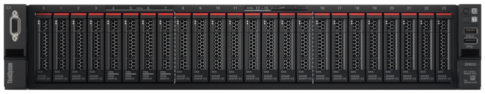
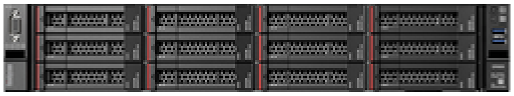
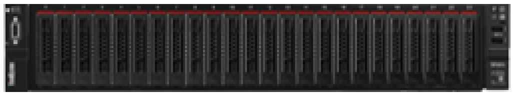
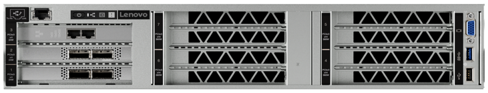
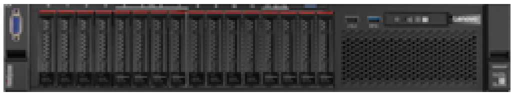
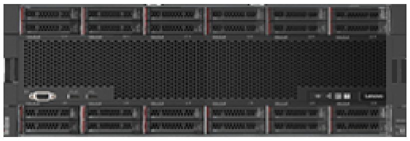
Client requirements:

- Hardware: CPU of 2.0 GHz or above, memory of 8 GB or above
- Browser: Chrome (V 62.0 or higher) or Firefox (V 56.0 or higher) recommended
- Display resolution: 1280 x 800 or above

Supported servers and chassis models

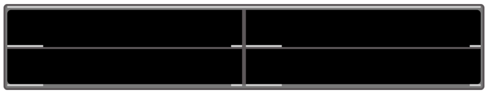
LiCO can be installed on certain servers, as listed in the table below.

Table 2. Supported servers

Product code	Machine type	Product name	Appearance
sd530	7X21	Lenovo ThinkSystem SD530 (0.5U)	
sd650	7X58	Lenovo ThinkSystem SD650 (2 nodes per 1U tray)	
sr630	7X01, 7X02	Lenovo ThinkSystem SR630 (1U)	
sr645	7D2X, 7D2Y	Lenovo ThinkSystem SR645 (1U)	
sr650	7X05, 7X06	Lenovo ThinkSystem SR650 (2U)	
sr655	7Y00, 7Z01	Lenovo ThinkSystem SR655 (2U)	
sr665	7D2V, 7D2W	Lenovo ThinkSystem SR665 (2U)	
sr670	7Y36, 7Y37, 7Y38	Lenovo ThinkSystem SR670 (2U)	
sr850	7X18, 7X19	Lenovo ThinkSystem SR850 (2U)	
sr850p	7D2F, 7D2G, 7D2H	Lenovo ThinkSystem SR850P (2U)	
sr950	7X11, 7X12, 7X13	Lenovo ThinkSystem SR950 (4U)	

LiCO can be installed on certain chassis models, as listed in the table below.

Table 3. Supported chassis models

Product code	Machine type	Model name	Appearance
d2	7X20	D2 Enclosure (2U)	
n1200	5456, 5468, 5469	NeXtScale n1200 (6U)	

Prerequisites

- Refer to LiCO best recipe to ensure that the cluster hardware uses proper firmware levels, drivers, and settings: https://support.lenovo.com/us/en/solutions/ht507011.
- Refer to the OS part of LeSI 20B_SI best recipe to install the OS security patch: https://support.lenovo.com/us/en/solutions/HT511104.
- Unless otherwise stated in this Guide, all commands are executed on the management node.
- To enable the firewall, modify the firewall rules according to “Firewall settings” on page 51.
- It is important to regularly patch and update components and the OS to prevent security vulnerabilities.
- Additionally it is recommended that known updates at the time of installation be applied during or immediately after the OS deployment to the managed nodes and prior to the rest of the LiCO setup steps.
- LiCO leverages OpenHPC packages which aggregate a number of common ingredients required to deploy and manage High Performance Computing (HPC) Linux clusters including provisioning tools, resource management, I/O clients, development tools, and a variety of scientific libraries. Lenovo provides a download of the most recent version of OpenHPC which is unmodified from what is distributed by OpenHPC. There are known open-source components within OpenHPC that have known, registered, vulnerabilities. None of these issues has been assessed as critical. However, it is recommended that the user update or remove such components using the native package management tools.

Chapter 2. Deploy the cluster environment

If the cluster environment already exists, you can skip this chapter.

Install an OS on the management node

Install an official version of CentOS 8.2 or RHEL 8.2 on the management node. You can select the minimum installation.

Run the following commands to configure the memory and restart OS:

```
echo '* soft memlock unlimited' >> /etc/security/limits.conf

echo '* hard memlock unlimited' >> /etc/security/limits.conf

reboot
```

Deploy the OS on other nodes in the cluster

Configure environment variables

Step 1. Log in to the management node.

Step 2. Run the following commands to configure environment variables for the entire installation process:

```
su root

cd ~

vi lico_env.local
```

Step 3. Run the following commands to edit the `lico_env.local` file:

```
# Management node hostname

sms_name="head"

# Set the domain name

domain_name="hpc.com"

# Set OpenLDAP domain name

lico_ldap_domain_name="dc=hpc,dc=com"

# set OpenLDAP domain component

lico_ldap_domain_component="hpc"

# IP address of management node in the cluster intranet

sms_ip="192.168.0.1"

# The network adapter name corresponding to the management node IP address

sms_eth_internal="eth0"

# Subnet mask in the cluster intranet. If all nodes in the cluster already have

# OS installed, retain the default configurations.
```

```
internal_netmask="255.255.0.0"

# BMC username and password

bmc_username="<BMC_USERNAME>"

bmc_password="<BMC_PASSWORD>"

# original OS repository directory

repo_backup_dir="/install/custom/backup"

# OS mirror pathway for xCAT

iso_path="/isos"

# Local repository directory for OS

os_repo_dir="/install/custom/server"

sdk_repo_dir="/install/custom/sdk"

# Local repository directory for xCAT

xcat_repo_dir="/install/custom/xcat"

# link name of repository directory for Lenovo OpenHPC

link_ohpc_repo_dir="/install/custom/ohpc"

# link name of repository directory for LiCO

link_lico_repo_dir="/install/custom/lico"

# link name of repository directory for LiCO-dep

link_lico_dep_repo_dir="/install/custom/lico-dep"

# Local repository directory for Lenovo OpenHPC, please change it

# according to this version.

ohpc_repo_dir="/install/custom/ohpc-2.0.0"

# LiCO repository directory for LiCO, please change it according to this version.

lico_repo_dir="/install/custom/lico-6.1.0"

# LiCO repository directory for LiCO-dep, please change it according to this version.

lico_dep_repo_dir="/install/custom/lico-dep-6.1.0"

# Total compute nodes

num_computes="2"

# Prefix of compute node hostname. If OS has already been installed on all nodes in the

# cluster, change the configuration according to actual conditions.

compute_prefix="c"

# Compute node hostname list. If OS has already been installed on all nodes

# in the cluster, change the configuration according to actual conditions.

c_name[0]=c1
```

```
c_name[1]=c2
# Compute node IP list. If OS has already been installed on all nodes in the cluster,
# change the configuration according to actual conditions.
c_ip[0]=192.168.0.6
c_ip[1]=192.168.0.16
# Network interface card MAC address corresponding to the compute node IP. If OS has
# already been installed on all nodes in the cluster, change the configuration according
# to actual conditions.
c_mac[0]=fa:16:3e:73:ec:50
c_mac[1]=fa:16:3e:27:32:c6
# Compute node BMC address list.
c_bmc[0]=192.168.1.6
c_bmc[1]=192.168.1.16
# Total login nodes. If there is no login node in the cluster, the number of logins
# must be "0". And the 'l_name', 'l_ip', 'l_mac', and 'l_bmc' lines need to be removed.
num_logins="1"
# Login node hostname list. If OS has already been installed on all nodes in the cluster,
# change the configuration according to actual conditions.
l_name[0]=l1
# Login node IP list. If OS has already been installed on all nodes in the cluster,
# change the configuration according to actual conditions.
l_ip[0]=192.168.0.15
# Network interface card MAC address corresponding to the login node IP.
# If OS has already been installed on all nodes in the cluster, change the configuration
# according to actual conditions.
l_mac[0]=fa:16:3e:2c:7a:47
# Login node BMC address list.
l_bmc[0]=192.168.1.15
# icinga api listener port
icinga_api_port=5665
```

Step 4. Save the changes to `lico_env.local`.
This Guide assumes that the node's BMC username and password are consistent. If they are inconsistent, they need to be modified during the installation.

Step 5. Run the following commands to make the configuration file take effect:

```
chmod 600 lico_env.local
source lico_env.local
```

After the cluster environment is set up, configure the IP address of the public network on the login or management node. In this way, you can log in to LiCO Web portal from external network.

Create a local repository

Different steps should be followed depending on the operating system.

For CentOS

Step 1. Run the following command to create a directory for ISO storage:

```
mkdir -p ${iso_path}
```

Step 2. Download the `CentOS-8.2.2004-x86_64-dvd1.iso` and `CHECKSUM` file from https://centos.org/download/.

Step 3. Copy the file to `${iso_path}`.

Step 4. Run the following commands to get verification code of the iso file, ensure this verification code is the same as the verification code in `CHECKSUM`.

```
cd ${iso_path}

sha256sum CentOS-8.2.2004-x86_64-dvd1.iso

cd ~
```

Step 5. Run the following commands to mount image:

```
mkdir -p ${os_repo_dir}

mount -o loop ${iso_path}/CentOS-8.2.2004-x86_64-dvd1.iso ${os_repo_dir}
```

Step 6. Run the following commands to configure local repository:

```
cat << eof > ${iso_path}/EL8-OS.repo

[AppStream]

name=appstream

baseurl=file://${os_repo_dir}/AppStream/

enabled=1

gpgcheck=1

gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-centosofficial

[BaseOS]

name=baseos

baseurl=file://${os_repo_dir}/BaseOS/

enabled=1

gpgcheck=1

gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-centosofficial

eof

cp -a ${iso_path}/EL8-OS.repo /etc/yum.repos.d/
```

Step 7. Run the following commands to close the original repo:

```
mkdir -p ${repo_backup_dir}
```

```
mv /etc/yum.repos.d/CentOS* ${repo_backup_dir}

dnf clean all

dnf makecache
```

Step 8. Run the following commands to enable nginx:

```
dnf module reset nginx

dnf module enable -y nginx:1.16
```

For RHEL

Step 1. Run the following command to create a directory for ISO storage:

```
mkdir -p ${iso_path}
```

Step 2. Copy the `RHEL-8.2.0-20200404.0-x86_64-dvd1.iso` and `RHEL-8.2.0-20200404.0-x86_64-dvd1.iso.MD5SUM` files to the `${iso_path}` directory.

Step 3. Run the following commands to check the validity of the iso file:

```
cd ${iso_path}

md5sum -c RHEL-8.2.0-20200404.0-x86_64-dvd1.iso.MD5SUM

cd ~
```

Step 4. Run the following commands to mount image:

```
mkdir -p ${os_repo_dir}

mount -o loop ${iso_path}/RHEL-8.2.0-20200404.0-x86_64-dvd1.iso ${os_repo_dir}
```

Step 5. Run the following commands to configure local repository:

```
cat << eof > ${iso_path}/EL8-OS.repo

[AppStream]

name=appstream

baseurl=file://${os_repo_dir}/AppStream/

enabled=1

gpgcheck=1

gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release

[BaseOS]

name=baseos

baseurl=file://${os_repo_dir}/BaseOS/

enabled=1

gpgcheck=1

gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-redhat-release

eof

cp -a ${iso_path}/EL8-OS.repo /etc/yum.repos.d/
```

Step 6. Run the following commands to enable nginx:

```
dnf module reset nginx
dnf module enable -y nginx:1.16
```

Install Lenovo xCAT

Step 1. Download the package from https://hpc.lenovo.com/downloads/20b/confluent-3.0.6-xcat-2.16.0.lenovo2-el8.tar.bz2.

Step 2. Upload the package to the `/root directory` on the management node.

Step 3. Run the following commands to create xcat local repository:

```
dnf install -y bzip2 tar

mkdir -p $xcat_repo_dir

cd /root

tar -xvf confluent-3.0.6-xcat-2.16.0.lenovo2-el8.tar.bz2 -C $xcat_repo_dir

cd $xcat_repo_dir/lenovo-hpc-el8

./mklocalrepo.sh

cd ~
```

Step 4. Run the following commands to install xcat:

```
dnf install -y xCAT

source /etc/profile.d/xcat.sh
```

Prepare OS mirrors for other nodes

For CentOS

Step 1. Run the following command to prepare the OS image for the other nodes:

```
copycds ${iso_path}/CentOS-8.2.2004-x86_64-dvd1.iso
```

Step 2. Run the following command to confirm that the OS image has been copied:

```
lsdef -t osimage
```

Note: The output should be as follows:

```
centos8.2-x86_64-install-compute (osimage)

centos8.2-x86_64-install-service (osimage)

centos8.2-x86_64-netboot-compute (osimage)

centos8.2-x86_64-statelite-compute (osimage)
```

Step 3. Run the following command to disable the Nouveau module:

```
chdef -t osimage centos8.2-x86_64-install-compute addkcmdline=\

"rdblacklist=nouveau nouveau.modeset=0 R::modprobe.blacklist=nouveau"
```

Note: The Nouveau module is an accelerated open-source driver for NVIDIA cards. This module should disabled before the installation of the CUDA driver.

For RHEL

Step 1. Run the following command to prepare the OS image for the other nodes:

```
copycds ${iso_path}/RHEL-8.2.0-20200404.0-x86_64-dvd1.iso
```

Step 2. Run the following commands to confirm that the OS image has been copied:

```
lsdef -t osimage
```

Note: The output should be as follows:

```
rhels8.2.0-x86_64-install-compute (osimage)

rhels8.2.0-x86_64-install-service (osimage)

rhels8.2.0-x86_64-netboot-compute (osimage)

rhels8.2.0-x86_64-statelite-compute (osimage)
```

Step 3. Run the following command to disable the Nouveau module:

```
chdef -t osimage rhels8.2.0-x86_64-install-compute addkcmdline=\

"rdblacklist=nouveau nouveau.modeset=0 R::modprobe.blacklist=nouveau"
```

Note: The Nouveau module is an accelerated open-source driver for NVIDIA cards. This module should be disabled before installation of the CUDA driver.

Set xCAT node information

Note: If the ThinkSystem SR635/SR655 server is used in other nodes, change "serialport=0" to "serialport=1" before running the following commands.

Step 1. Run the following commands to import the compute node configuration in the *lico_env.local* file to xCAT:

```
for ((i=0; i<$num_computes; i++)); do

mkdef -t node ${c_name[$i]} groups=compute,all arch=x86_64 netboot=xnba mgt=ipmi \

bmcusername=${bmc_username} bmcpassword=${bmc_password} ip=${c_ip[$i]} \

mac=${c_mac[$i]} bmc=${c_bmc[$i]} serialport=0 serialspeed=115200;

done
```

Step 2. Run the following commands to import the login node configuration to xCAT:

```
for ((i=0; i<$num_logins; i++)); do

mkdef -t node ${l_name[$i]} groups=login,all arch=x86_64 netboot=xnba mgt=ipmi \

bmcusername=${bmc_username} bmcpassword=${bmc_password} ip=${l_ip[$i]} \

mac=${l_mac[$i]} bmc=${l_bmc[$i]} serialport=0 serialspeed=115200;

done
```

Step 3. (Optional) If the BMC username and password of the node are inconsistent, run the following command to make them consistent:

```
tabedit ipmi
```

Step 4. Run the following command to configure the root account password for the node:

```
chtab key=system passwd.username=root passwd.password=<ROOT_PASSWORD>
```

Add host resolution

Note: If the cluster already has the OS installed and can resolve the IP address through the hostname, skip this section.

Run the following commands to add host resolution:

```
chdef -t site domain=${domain_name}
```

```
chdef -t site master=${sms_ip}
```

```
chdef -t site nameservers=${sms_ip}
```

```
sed -i "/^\s*${sms_ip}\s*.*$/d" /etc/hosts
```

```
sed -i "/\s*${sms_name}\s*/d" /etc/hosts
```

```
echo "${sms_ip} ${sms_name} ${sms_name}.${domain_name} " >> /etc/hosts
```

```
makehosts
```

Configure DHCP and DNS services

Note: If all nodes in the cluster have the OS installed, skip this step.

Run the following commands to configure DHCP and DNS services:

```
makenetworks
```

```
makedhcp -n
```

```
makedhcp -a
```

```
makedns -n
```

```
echo "search ${domain_name}" > /etc/resolv.conf
```

```
echo "nameserver ${sms_ip}" >> /etc/resolv.conf
```

Note: To ensure that the management node is pointing at the same DNS as other nodes, refer to the following two Web sites:
https://sourceforge.net/p/xcat/wiki/XCAT_iDataPlex_Cluster_Quick_Start/#install-xcat-on-the-management-node

https://sourceforge.net/p/xcat/wiki/Cluster_Name_Resolution/

Install a node OS through the network

Note: If all nodes in the cluster have the OS installed, skip this section.

For CentOS, run the following commands to set and install the necessary OS mirror:

```
mkdir /opt/xcat/share/xcat/netboot/syslinux
```

```
touch /opt/xcat/share/xcat/netboot/syslinux/pxelinux.0
```

```
nodeset all osimage=centos8.2-x86_64-install-compute
```

```
rsetboot all net -u
```

```
rpower all reset
```

For RHEL, run the following commands to set and install the necessary OS mirror:

```
mkdir /opt/xcat/share/xcat/netboot/syslinux
```

```
touch /opt/xcat/share/xcat/netboot/syslinux/pxelinux.0
```

```
nodeset all osimage=rhels8.2.0-x86_64-install-compute
```

```
rsetboot all net -u
```

```
rpower all reset
```

Note: It takes several minutes to complete the OS installation. You can use the following command to check the progress:

```
nodestat all
```

Create local repository for other nodes

```
cp /etc/yum.repos.d/EL8-OS.repo /var/tmp

sed -i '/^baseurl=/d' /var/tmp/EL8-OS.repo

sed -i "/name=appstream/a\baseurl=http://${sms_name}${os_repo_dir}/AppStream/" \

/var/tmp/EL8-OS.repo

sed -i "/name=baseos/a\baseurl=http://${sms_name}${os_repo_dir}/BaseOS/" \

/var/tmp/EL8-OS.repo

xdcp all /var/tmp/EL8-OS.repo /etc/yum.repos.d/
```

Attention: If the OS of other nodes are CentOS, run the following commands to close the repo:

```
psh all "mkdir -p ${repo_backup_dir}"

psh all "mv /etc/yum.repos.d/CentOS* ${repo_backup_dir}"

psh all "dnf clean all"

psh all "dnf makecache"
```

Enable nginx for other nodes

Run the following commands to enable nginx for other nodes:

```
psh all dnf module reset nginx

psh all dnf module enable -y nginx:1.16
```

Configure the memory for other nodes

Run the following commands:

```
xdcp all /etc/security/limits.conf /etc/security/limits.conf

psh all reboot
```

Checkpoint A

Run the following commands to check and ensure that the installation is complete:

```
psh all uptime
```

Note: The output should be as follows:

```
c1: 05:03am up 0:02, 0 users, load average: 0.20, 0.13, 0.05

c2: 05:03am up 0:02, 0 users, load average: 0.20, 0.14, 0.06

l1: 05:03am up 0:02, 0 users, load average: 0.17, 0.13, 0.05

……
```

Note: If you cannot run these commands,check if the xCAT is successfully installed on the management node, and passwordless SSH is set between the management node and other nodes. You can copy the id_rsa file and the id_rsa.pub file from the management node to other nodes, and run these commands again.

Install infrastructure software for nodes

List of infrastructure software to be installed

Note: In the **Installation node** column, M stands for “Management node”, L stands for “Login node”, and C stands for “Compute node”.

Table 4. Infrastructure software to be installed

Software name	Component name	Version	Service name	Installation node	Notes
nfs	nfs-utils	2.3.3	nfs-server	M	/
chrony	chrony	3.5	chronyd	M, C, L	/
slurm	ohpc-slurm-server	2.0	munge, slurmctld	M	/
	ohpc-slurm-client	2.0	munge, slurmd	C, L	/
icinga2	icinga2	2.12.0	icinga2	M, C, L	/
singularity	singularity-ohpc	3.4.2	/	M	/
mpi	openmpi4-gnu9-ohpc	4.0.4	/	M	At least ont MPI type required
	mpich-ofi-gnu9-ohpc	3.3.2	/	M	
	mvapich2-gnu9-ohpc	2.3.2	/	M	

Configure a local repository for the management node

Step 1. Download the package from https://hpc.lenovo.com/lico/downloads/6.1/Lenovo-OpenHPC-2.0.CentOS_8.x86_64.tar.

Step 2. Upload the package to the `/root directory` on the management node.

Step 3. Run the following commands to configure the local Lenovo OpenHPC repository:

```
mkdir -p $ohpc_repo_dir

cd /root

tar xvf Lenovo-OpenHPC-2.0.CentOS_8.x86_64.tar -C $ohpc_repo_dir

rm -rf $link_ohpc_repo_dir

ln -s $ohpc_repo_dir $link_ohpc_repo_dir

$link_ohpc_repo_dir/make_repo.sh
```

Configure a local repository for login and compute nodes

Step 1. Run the following commands to add a local repository:

```
cp /etc/yum.repos.d/Lenovo.OpenHPC.local.repo /var/tmp

sed -i '/^baseurl=/d' /var/tmp/Lenovo.OpenHPC.local.repo

sed -i '/^gpgkey=/d' /var/tmp/Lenovo.OpenHPC.local.repo

echo "baseurl=http://${sms_name}${link_ohpc_repo_dir}/CentOS_8" \

>> /var/tmp/Lenovo.OpenHPC.local.repo

echo "gpgkey=http://${sms_name}${link_ohpc_repo_dir}/CentOS_8\

/repodata/repomd.xml.key" >> /var/tmp/Lenovo.OpenHPC.local.repo
```

Step 2. Run the following commands to distribute files for other nodes:

```
xdcp all /var/tmp/Lenovo.OpenHPC.local.repo /etc/yum.repos.d/

psh all echo -e %_excludedocs 1 \>\> ~/.rpmmacros
```

The operating system itself should have enough packages installed. Otherwise, the subsequent installation steps may fail.

Configure LiCO dependencies repositories

Step 1. Download the package from https://hpc.lenovo.com/lico/downloads/6.1/lico-dep-6.1.0.el8.x86_64.tgz.

Step 2. Upload the package to the `/root directory` on the management node.

Step 3. Run the following commands to configure the repository for the management node:

```
mkdir -p $lico_dep_repo_dir

cd /root

tar -xvf lico-dep-6.1.0.el8.x86_64.tgz -C $lico_dep_repo_dir

rm -rf $link_lico_dep_repo_dir

ln -s $lico_dep_repo_dir $link_lico_dep_repo_dir

$link_lico_dep_repo_dir/mklocalrepo.sh
```

Attention: Before running the commands, ensure that the management node has configured a local operating system repository for the above and the subsequent actions.

Step 4. (Optional) If the cluster already exists, check to ensure that your version is consistent with "List of LiCO dependencies to be installed" on page 29.

Step 5. Run the following commands to configure the repository for other nodes:

```
cp /etc/yum.repos.d/lico-dep.repo /var/tmp

sed -i '/^baseurl=/d' /var/tmp/lico-dep.repo

sed -i '/^gpgkey=/d' /var/tmp/lico-dep.repo

sed -i "/name=lico-dep-local-library/a\baseurl=http://${sms_name}\

${link_lico_dep_repo_dir}/library/" /var/tmp/lico-dep.repo

sed -i "/name=lico-dep-local-library/a\gpgkey=http://${sms_name}\

${link_lico_dep_repo_dir}/RPM-GPG-KEY-LICO-DEP-EL8" /var/tmp/lico-dep.repo

sed -i "/name=lico-dep-local-standalone/a\baseurl=http://${sms_name}\
${link_lico_dep_repo_dir}/standalone/" /var/tmp/lico-dep.repo
sed -i "/name=lico-dep-local-standalone/a\gpgkey=http://${sms_name}\
${link_lico_dep_repo_dir}/RPM-GPG-KEY-LICO-DEP-EL8" /var/tmp/lico-dep.repo
xdcp all /var/tmp/lico-dep.repo /etc/yum.repos.d
```

Obtain the LiCO installation package

Step 1. Obtain the LiCO 6.1.0 release package for EL8 `lico-release-6.1.0.el8.tar.gz` and LiCO license file from https://commercial.lenovo.com/cn/en/signin.

Step 2. Upload the release package to the management node.

Configure the local repository for LiCO

Step 1. Run the following commands to configure the local repository for the management node:

```
mkdir -p $lico_repo_dir
tar zxvf lico-release-6.1.0.el8.tar.gz -C $lico_repo_dir --strip-components 1
rm -rf $link_lico_repo_dir
ln -s $lico_repo_dir $link_lico_repo_dir
$link_lico_repo_dir/mklocalrepo.sh
```

Step 2. Run the following commands to configure the local yum repository for the other nodes:

```
cp /etc/yum.repos.d/lico-release.repo /var/tmp
sed -i '/baseurl=/d' /var/tmp/lico-release.repo
sed -i "/name=lico-release-host/a\baseurl=http://${sms_name}\
${link_lico_repo_dir}/host/" /var/tmp/lico-release.repo
sed -i "/name=lico-release-public/a\baseurl=http://${sms_name}\
${link_lico_repo_dir}/public/" /var/tmp/lico-release.repo
```

Step 3. Run the following command to distribute repo files:

```
xdcp all /var/tmp/lico-release.repo /etc/yum.repos.d/
```

Configure the xCAT local repository

Step 1. Run the following commands to configure the local repository for the other nodes:

```
cp /etc/yum.repos.d/lenovo-hpc.repo /var/tmp
sed -i '/^baseurl=/d' /var/tmp/lenovo-hpc.repo
sed -i '/^gpgkey=/d' /var/tmp/lenovo-hpc.repo
echo "baseurl=http://${sms_name}${xcat_repo_dir}/lenovo-hpc-el8" \
>>/var/tmp/lenovo-hpc.repo
echo "gpgkey=http://${sms_name}${xcat_repo_dir}/lenovo-hpc-el8\
/lenovohpckey.pub" >> /var/tmp/lenovo-hpc.repo
```

Step 2. Run the following command to distribute the repo files:

```
xdcp all /var/tmp/lenovo-hpc.repo /etc/yum.repos.d/
```

Install Slurm

Step 1. Run the following command to install the base package:

```
dnf install -y lenovo-ohpc-base
```

Step 2. Run the following command to install Slurm:

```
dnf install -y ohpc-slurm-server
```

Step 3. Run the following command to install the Slurm client:

```
psh all dnf install -y ohpc-base-compute ohpc-slurm-client lmod-ohpc
```

Step 4. (Optional) Run the following command to prevent non-root logins to the compute nodes:

```
psh compute echo "\""account required pam_slurm.so"\"" \>\> /etc/pam.d/sshd
```

Note: Do not perform this step if you want to allow non-root logins to the compute nodes regardless of whether a Slurm job is running on those nodes. If this step is performed, non-root logins to the compute nodes will only be allowed if a Slurm job is already running on those nodes under a particular username. In this case, non-root ssh logins will work for that particular username for the duration that Slurm job is running on those nodes.

Step 5. (Optional) To save the previous job information and use memory accounting function, install and configure slurm accounting function based on the information on https://slurm.schedmd.com/accounting.html.

Configure NFS

Step 1. Run the following commands to create the shared directory `/opt/ohpc/pub`:

a. Manage node sharing `/opt/ohpc/pub` for OpehHPC:

```
dnf install -y nfs-utils

echo "/opt/ohpc/pub *(ro,no_subtree_check,fsid=11)">> /etc/exports

exportfs -a
```

b. Install NFS for Cluster Nodes:

```
psh all dnf install -y nfs-utils
```

c. Configure shared directory for cluster nodes:

```
psh all mkdir -p /opt/ohpc/pub

psh all echo "\""${sms_ip}:/opt/ohpc/pub /opt/ohpc/pub nfs nfsvers=4.0,nodev,noatime \

0 0"\"" \>\> /etc/fstab
```

d. Mount shared directory:

```
psh all mount /opt/ohpc/pub
```

Attention: This directory is mandatory. If you have already shared this directory from the management node and mounted it on all other nodes, you can skip this step.

Step 2. Run the following commands to create the user shared directory (taking `/home` as an example):

a. Manage node sharing `/home`:

```
echo "/home *(rw,async,no_subtree_check,fsid=10,no_root_squash)" >> /etc/exports
```

```
exportfs -a
```

b. Unmount the mounted `/home`:

```
psh all "sed -i '/ \/home /d' /etc/fstab"

psh all umount /home
```

c. Configure the shared directory for cluster nodes:

```
psh all echo "\""${sms_ip}:/home /home nfs nfsvers=4.0,nodev,nosuid,noatime \

0 0"\"" \>\> /etc/fstab
```

d. Mount shared directory:

```
psh all mount /home
```

Configure Chrony

Note: If the chronyd service has already been configured for nodes in the cluster, skip this section.

Unsynchronized cluster time might cause unexpected problems.Configure chronyd service referring to https://chrony.tuxfamily.org/documentation.html.

GPU driver installation

You must install the GPU driver on all the GPU compute nodes. If only a subset of nodes have GPUs, replace the **compute** argument in **psh** commands with the node range corresponding to GPU nodes.

Install the GPU driver

Step 1. Run the following commands to configure the operating system to start on the text console and then restart the system:

```
psh compute systemctl set-default multi-user.target

psh compute reboot
```

Note: This step is required only when the operating system is configured to start on a graphical desktop.

Step 2. Download the NVIDIA driver from https://us.download.nvidia.cn/tesla/450.51.06/nvidia-driver-local-repo-rhel8-450.51.06-1.0-1.x86_64.rpm, and copy it to the shared directory `/home`.

Step 3. Run the following commands:

```
psh compute dnf install -y gcc gcc-c++ make

psh compute dnf install -y kernel-devel-$(uname -r) kernel-headers-$(uname -r)

psh compute rpm -ivh /home/nvidia-driver-local-repo-rhel8-450.51.06-1.0-1.x86_64.rpm

psh compute dnf install -y cuda-drivers
```

Step 4. Run the following command on the GPU nodes to determine if GPU can be identified:

```
psh compute nvidia-smi | xcoll
```

Note: If you cannot identify the GPU information by running the command, run the following command and reboot all GPU nodes. Then re-run the command.

```
psh compute reboot
```

Configure automatic start for the GPU driver

Run the following commands to configure automatic start for the GPU driver:

```
cat <<eof> /tmp/nvidia-modprobe-loader.service

[Unit]

Description=NVIDIA ModProbe Service

After=syslog.target

Before=slurmd.service

[Service]

Type=oneshot

ExecStart=/usr/bin/nvidia-modprobe -u -c=0

RemainAfterExit=yes

[Install]

WantedBy=multi-user.target

eof

xdcp compute /tmp/nvidia-modprobe-loader.service \

/usr/lib/systemd/system/nvidia-modprobe-loader.service

psh compute systemctl daemon-reload

psh compute systemctl enable nvidia-persistenced --now

psh compute systemctl enable nvidia-modprobe-loader.service --now
```

Configure Slurm

Step 1. Download `slurm.conf` from https://hpc.lenovo.com/lico/downloads/6.1/examples/conf/ to `/etc/slurm` on the management node, and modify this file according to the instructions in “slurm.conf” on page 20.

Step 2. Download `cgroup.conf` from https://hpc.lenovo.com/lico/downloads/6.1/examples/conf/ to `/etc/slurm` on the management node.

Step 3. Run the following commands to distribute the configuration:

```
xdcp all /etc/slurm/slurm.conf /etc/slurm/slurm.conf

xdcp all /etc/slurm/cgroup.conf /etc/slurm/cgroup.conf

xdcp all /etc/munge/munge.key /etc/munge/munge.key
```

Step 4. For GPU nodes only: Download `gres.conf` from https://hpc.lenovo.com/lico/downloads/6.1/examples/conf/ to `/etc/slurm` on the GPU node, and modify this file according to the instructions in “gres.conf” on page 20.

Step 5. Run the following commands to start other node services:

```
systemctl enable munge

systemctl enable slurmctld

systemctl restart munge

systemctl restart slurmctld
```

Step 6. Run the following commands to start other node service:

```
psh all systemctl enable munge
```

```
psh all systemctl restart munge
```

```
psh all systemctl enable slurmd
```

```
psh all systemctl restart slurmd
```

slurm.conf

The following typical fields need to be configured:

- Cluster name:

  ```
  ClusterName=mycluster
  ```

- Management node name:

  ```
  ControlMachine=c031
  ```

- GPU scheduling:

  ```
  GresTypes=gpu
  ```

  Note: In the cluster, this entry is used when a GPU node is included. If the cluster includes no GPU node, delete this entry.

- Cluster node definitions:

  ```
  NodeName=c031 Gres=gpu:4 CPUs=28 RealMemory=200000 State=UNKNOWN
  ```

  ```
  NodeName=c032 Gres=gpu:4 CPUs=28 RealMemory=200000 State=UNKNOWN
  ```

  – **Gres**: Number of GPUs

  – **CPUs**: Number of CPUs on a node.

  – **RealMemory**: Memory size of a node (Unit: M).

- Partition definitions:

  ```
  PartitionName=compute Nodes=c0[31-32] Default=YES MaxTime=INFINITE State=UP
  ```

  ```
  PartitionName=compute1 Nodes=c0[31-32] Default=NO MaxTime=INFINITE State=UP
  ```

  Notes:

  – **Default**: identifies whether this partition is the default partition. When submitting a job, you can select a partition. If you do not select a partition, the default partition is used.

  – **Nodes**: the NodeName list. If NodeName is irregular, Nodes=[nodename1,nodename2,...] is allowed.

- Enforced part limit definitions:

  ```
  EnforcePartLimits=ALL
  ```

  Attention: Use this configuration if you want to submit a direct error response when a job requests resources that exceed the cluster resource amount. Otherwise, the job remains in the queue.

For more details about how to configure `slurm.conf`, refer to the official Slurm site: https://slurm.schedmd.com/slurm.conf.html.

gres.conf

This configuration file describes the GPUs installed on the GPU nodes and the GPU memory. The content of this file may vary based on the GPU node.

Modify the following content:

```
Name=gpu File=/dev/nvidia[0-3]
```

Note: In `/dev/nvidia[0-3]`, `[0-3]` should be changed to your actual GPU configuration. For example, `/dev/nvidia0` means one GPU card, whereas `/dev/nvidia[0-1]` means two GPU cards.

Install Icinga2

If IB device is prepared and IB drive installation is required, install IB drive in the OS referring to https://support.lenovo.com/us/en/solutions/HT511104 before installing Icinga2. USB network card influences IB network card invoked by MPI. Therefore, it is recommended to add “rmmod cdc_ether” in power on procedure to remove USB network card.

Step 1. Run the following commands to install icinga2:

```
dnf install -y icinga2

psh all dnf install -y icinga2
```

Step 2. Run the following commands to install LiCO icinga2 plugin:

```
dnf install -y nagios-plugins-ping

dnf install -y lico-icinga-plugin

psh all dnf install -y lico-icinga-plugin
```

Step 3. Run the following command to open API function on the management node:

```
icinga2 api setup
```

Step 4. Run the following commands to configure the icinga2 on the management node:

```
icinga2 node setup --master --disable-confd

systemctl restart icinga2
```

Step 5. Run the following commands to configure icinga2 agent for other nodes:

```
psh all icinga2 pki save-cert --trustedcert \

/var/lib/icinga2/certs/trusted-parent.crt --host ${sms_name}


for ((i=0;i<$num_computes;i++));do

ticket=`icinga2 pki ticket --cn ${c_name[${i}]}`

psh ${c_name[${i}]} icinga2 node setup --ticket ${ticket} --cn ${c_name[${i}]} \

--endpoint ${sms_name} --zone ${c_name[${i}]} --parent_zone master --parent_host \

${sms_name} --trustedcert /var/lib/icinga2/certs/trusted-parent.crt \

--accept-commands --accept-config --disable-confd

done

for ((i=0;i<$num_logins;i++));do

ticket=`icinga2 pki ticket --cn ${l_name[${i}]}`

psh ${l_name[${i}]} icinga2 node setup --ticket ${ticket} --cn ${l_name[${i}]} \

--endpoint ${sms_name} --zone ${l_name[${i}]} --parent_zone master --parent_host \
${sms_name} --trustedcert /var/lib/icinga2/certs/trusted-parent.crt \

--accept-commands --accept-config --disable-confd

done

psh all systemctl restart icinga2
```

Step 6. Run the following commands to configure global template file on management node:

```
mkdir -p /etc/icinga2/zones.d/global-templates

echo -e "object CheckCommand \"lico_monitor\" {\n  command = [ PluginDir + \"/lico-\

icinga-plugin\" ]\n}" > /etc/icinga2/zones.d/global-templates/commands.conf

chown -R icinga:icinga /etc/icinga2/zones.d/global-templates
```

Step 7. Run the following commands to define zone file:

```
mkdir -p /etc/icinga2/zones.d/master

echo -e "object Host \"${sms_name}\" {\n check_command = \"hostalive\"\n \

address = \"${sms_ip}\"\n vars.agent_endpoint = name\n}\n" >> \

/etc/icinga2/zones.d/master/hosts.conf


for ((i=0;i<$num_computes;i++));do

echo -e "object Endpoint \"${c_name[${i}]}\" {\n host = \"${c_name[${i}]}\"\n \

port = \"${icinga_api_port}\"\n log_duration = 0\n}\nobject \

Zone \"${c_name[${i}]}\" {\n endpoints = [ \"${c_name[${i}]}\" ]\n \

parent = \"master\"\n}\n" >> /etc/icinga2/zones.d/master/agent.conf

echo -e "object Host \"${c_name[${i}]}\" {\n check_command = \"hostalive\"\n \

address = \"${c_ip[${i}]}\"\n vars.agent_endpoint = name\n}\n" >> \

/etc/icinga2/zones.d/master/hosts.conf

done


for ((i=0;i<$num_logins;i++));do

echo -e "object Endpoint \"${l_name[${i}]}\" {\n host = \"${l_name[${i}]}\"\n \

port = \"${icinga_api_port}\"\n log_duration = 0\n}\nobject \

Zone \"${l_name[${i}]}\" {\n endpoints = [ \"${l_name[${i}]}\" ]\n \

parent = \"master\"\n}\n" >> /etc/icinga2/zones.d/master/agent.conf

echo -e "object Host \"${l_name[${i}]}\" {\n check_command = \"hostalive\"\n \

address = \"${l_ip[${i}]}\"\n vars.agent_endpoint = name\n}\n" >> \

/etc/icinga2/zones.d/master/hosts.conf

done
```

```
echo -e "apply Service \"lico\" {\n check_command = \"lico_monitor\"\n \

max_check_attempts = 5\n check_interval = 1m\n retry_interval = 30s\n assign \

where host.name == \"${sms_name}\"\n assign where host.vars.agent_endpoint\n \

command_endpoint = host.vars.agent_endpoint\n}\n" > \

/etc/icinga2/zones.d/master/service.conf

chown -R icinga:icinga /etc/icinga2/zones.d/master

systemctl restart icinga2
```

Step 8. Run the following commands to enable service:

```
psh all systemctl enable icinga2

systemctl enable icinga2
```

Note: Run the following command to check the configuration:

```
icinga2 daemon -C
```

Install MPI

Step 1. Run the following commands to install three modules (OpenMPI, MPICH, and MVAPICH) to the system:

```
dnf install -y openmpi4-gnu9-ohpc mpich-ofi-gnu9-ohpc mvapich2-gnu9-ohpc ucx-ib-ohpc
```

Step 2. Set the default module.

Run the following command to set OpenMPI module as the default:

```
dnf install -y lmod-defaults-gnu9-openmpi4-ohpc
```

Run the following command to set the MPICH module as the default:

```
dnf install -y lmod-defaults-gnu9-mpich-ofi-ohpc
```

Run the following command to set the MVAPICH module as the default:

```
dnf install -y lmod-defaults-gnu9-mvapich2-ohpc
```

Note: MVAPICH requires that Infiniband or OPA is present and working correctly. The following packages should be installed to support Infiniband or OPA:

```
dnf list installed libibmad5 librdmacm1 rdma infinipath-psm dapl-devel \

dapl-utils libibverbs-utils
```

Interconnect support between MPI types

Table 5. Interconnect support between MPI types

MPI	Ethernet (TCP)	InfiniBand	Omni-Path
MPICH	○		
MVAPICH2		○	
MVAPICH2 (psm2)			○
OpenMPI	○	○	○
OpenMPI (PMIx)	○	○	○

Dependencies between MPI types

Observe the following dependencies for MPI installation:

- To use MVAPICH2 (psm2), install `mvapich2-psm2-gnu9-ohpc`.
- To use OpenMPI (PMIx), install `openmpi4-pmix-slurm-gnu9-ohpc`.
- `openmpi4-gnu9-ohpc` and `openmpi4-pmix-slurm-gnu9-ohpc` are incompatible.
- `mvapich2-psm2-gnu9-ohpc` and `mvapich2-gnu9-ohpc` are incompatible.

Install Singularity

Singularity is an HPC-facing lightweight container framework.

Step 1. Run the following command to install Singularity:

```
dnf install -y singularity-ohpc
```

Step 2. Edit the file `/opt/ohpc/pub/modulefiles/ohpc` by adding the following content to the end of the `module try-add` block:

```
module try-add singularity
```

Step 3. In the `module del` block, add the following content as the first line:

```
module del singularity
```

Step 4. Run the following command:

```
source /etc/profile.d/lmod.sh
```

Changes to `/opt/ohpc/pub/modulefiles/ohpc` may be lost when the default modules are changed with the installation of the `lmod-defaults*` package. In that case, either modify `/opt/ohpc/pub/modulefiles/ohpc` again, or add `module try-add singularity` to the end of `/etc/profile.d/lmod.sh`.

Checkpoint B

Step 1. Run the following command to test if Slurm is properly installed:

```
sinfo
```

Notes:

- The output should be as follows:

  ```
  PARTITION AVAIL TIMELIMIT NODES STATE NODELIST

  normal* up 1-00:00:00 2 idle c[1-2]

  ……
  ```

- The status of all nodes should be **idle**; **idle*** is unacceptable. If the status is not **idle**, identify the causes by checking the logs in `/var/log/slurmctld.log` of managenent node and the logs in `/var/log/slurmd.log` of status error nodes.

Step 2. Run the following command to add a **test** account:

```
useradd test -m --uid 65530

psh all useradd test --uid 65530
```

Step 3. Log in a compute node by using the test account and the test program distrubuted by Slurm:

```
su - test

mpicc -O3 /opt/ohpc/pub/examples/mpi/hello.c
```

```
srun -n 8 -N 1 -w <NODENAME> -p <PARTITIONNAME> --pty /bin/bash

prun ./a.out
```

Note: The output should be as follows:

```
Master compute host = c1

Resource manager = slurm

Launch cmd = mpiexec.hydra -bootstrap slurm ./a.out

Hello, world (8 procs total)

--> Process # 0 of 8 is alive. -> c1

--> Process # 4 of 8 is alive. -> c2

--> Process # 1 of 8 is alive. -> c1

--> Process # 5 of 8 is alive. -> c2

--> Process # 2 of 8 is alive. -> c1

--> Process # 6 of 8 is alive. -> c2

--> Process # 3 of 8 is alive. -> c1

--> Process # 7 of 8 is alive. -> c2
```

Step 4. Run the following command to end the test:

```
exit
```

Note: If you need to leave from “test” user session, type “exit” again.

Step 5. Run the following command to remove the test user:

```
psh all userdel test

userdel test -r
```

After the command is completed, you are switched to the root user of the management node.

Chapter 3. Install LiCO dependencies

Cluster check

If you skipped Chapter 2 "Deploy the cluster environment" on page 5, follow this section to make sure that the cluster environment is ready. Otherwise, proceed to "List of LiCO dependencies to be installed" on page 29.

Check environment variables

Run the following commands to check the environment variables *${sms_name}*, *${lico_ldap_domain_name}*, and *${lico_repo_dir}*:

```
echo $sms_name;echo $lico_repo_dir;echo $lico_ldap_domain_name
```

Notes:

- The output should be as follows:

  ```
  head
  /install/custom/lico-6.1.0
  dc=hpc,dc=com
  ```

- If there is no output, see "Configure environment variables" on page 5.

Check the LiCO dependencies repository

Run the following commands to check the LiCO dependencies repository:

```
dnf repolist | grep lico-dep-local
```

Notes:

- The output should be as follows:

  ```
  lico-dep-local-library lico-dep-local-library 166
  lico-dep-local-standalone lico-dep-local-standalone 81
  ```

- If there is no output, see "Configure LiCO dependencies repositories" on page 15.

Check the LiCO repository

Run the following command to check the LiCO repository:

```
dnf repolist | grep lico-release
```

Notes:

- Following is the correct output:

  ```
  lico-release-host lico-release-host 72
  lico-release-public lico-release-public 38
  ```

- If there is no output, refer to "Configure the local repository for LiCO" on page 16.

Check the OS installation

Go to “Checkpoint A” on page 13 to check the OS installation for the cluster. If the OS installation check fails, reconfigure the cluster OS referring to “Deploy the OS on other nodes in the cluster” on page 5.

Check NFS

Note: If the cluster does not use NFS as the distributed file system, skip this section.

Run the following commands to check the NFS service on the management node:

```
systemctl status nfs-server | grep Active && exportfs -v | grep -E '/home|/opt/ohpc/pub'
```

Notes:

- The output should be as follows：

  ```
  Active: active (exited) since Sat 2019-10-12 16:04:21 CST; 2 days ago
  ```

  ```
  /opt/ohpc/pub <world>(sync,wdelay,hide,no_subtree_check,sec=sys,ro,secure,root_squash,no_all_
  squash)
  ```

  ```
  /home <world>(async,wdelay,hide,no_subtree_check,sec=sys,rw,secure,no_root_squash,no_all_squash)
  ```

- If the status is not 'active (exited)' or there is no output for **exportfs**, go to “Configure NFS” on page 17.

Run the following commands to check the mounting points on all other nodes:

```
psh all df | grep -E '/home | /opt/ohpc/pub'
```

Notes:

- The output should be as follows:

  ```
  c032: 10.1.1.31:/home 485642240 111060992 374581248 23% /home
  ```

  ```
  c032: 10.1.1.31:/opt/ohpc/pub 485642240 111060992 374581248 23% /opt/ohpc/pub
  ```

- If the status is no output, go to “Configure NFS” on page 17.

Check Slurm

Run the following commands to check slurmctld on the management node:

```
systemctl status slurmctld | grep Active
```

Notes:

- The output should be as follows:

  ```
  Active: active (running) since Tue 2018-07-24 19:02:49 CST; 1 months 20 days ago
  ```

- If the status is not 'active (running)', go to “Install Slurm” on page 17 and “Configure Slurm” on page 19.

Run the following commands to check slurmd on the compute nodes:

```
psh compute systemctl status slurmd | grep Active
```

Notes:

- The output should be as follows:

  ```
  c031: Active: active (running) since Tue 2018-07-24 19:02:49 CST; 1 months 20 days ago
  ```

  ```
  c032: Active: active (running) since Sat 2018-07-21 17:16:59 CST; 1 months 23 days ago
  ```

- If the output does not contain all compute nodes, go to “Install Slurm” on page 17 and “Configure Slurm” on page 19.

Check MPI and Singularity

Run the following command to check MPI and Singularity on the management node:

```
module list
```

Notes:

- The output should be as follows:

  ```
  Currently Loaded Modules:
  ```

  ```
  1) prun/2.0 2) gnu9/9.3.0 3) openmpi4/4.0.4 4) singularity/3.4.2 5) ohpc
  ```

- If the outputs does not contain one of the following: openmpi3, mpich, or mvapich2, go to "Install MPI" on page 23.
- If the output does not contain "singularity", go to "Install Singularity" on page 24.

Check OpenHPC installation

Go to "Checkpoint B" on page 24 to check the OpenHPC installation for the cluster. If the OpenHPC installation check fails, reconfigure OpenHPC referring to "Install infrastructure software for nodes" on page 14.

List of LiCO dependencies to be installed

Note: In the **Installation node** column, M stands for "Management node", L stands for "Login node", and C stands for "Compute node".

Table 6. LiCO dependencies to be installed

Software	Component	Version	Service	Installation node	Notes
rabbitmq	rabbitmq-server	3.8.2	rabbitmq-server	M	
mariadb	mariadb-server	10.3.17	mariadb		
influxdb	influxdb	1.8.0	influxdb	M	
confluent	confluent	3.0.4	confluent	M	
openldap	slapd-ssl-config	1.0.0	slapd	M	
	nss-pam-ldapd	0.9.9	nslcd	M, C, L	
	libuser	0.62		M	
	python3-libuser	0.62		M	

Install RabbitMQ

LiCO uses RabbitMQ as a message broker.

Step 1. Run the following command to install RabbitMQ on the management node:

```
dnf install -y rabbitmq-server
```

Step 2. Run the following commands to start RabbitMQ service:

```
systemctl enable rabbitmq-server --now
```

Install MariaDB

LiCO uses MariaDB as an object-related database for data storage.

Step 1. Run the following command to install MariaDB on the management node:

```
dnf install -y mariadb-server mariadb-devel
```

Step 2. Run the following command to start MariaDB:

```
systemctl enable mariadb --now
```

Step 3. Run the following commands to configure MariaDB for LiCO:

Note: The username and password will be used in installing lico-passwd-tool. Therefore, keep a record of them when installing MariaDB.

```
mysql

create database lico character set utf8 collate utf8_bin;

create user '<USERNAME>'@'localhost' identified by '<PASSWORD>';

grant ALL on lico.* to '<USERNAME>'@'localhost';

exit
```

Step 4. Run the following commands to configure MariaDB limits:

```
sed -i "/\[mysqld\]/a\max-connections=1024" /etc/my.cnf.d/mariadb-server.cnf

mkdir /usr/lib/systemd/system/mariadb.service.d

cat << eof > /usr/lib/systemd/system/mariadb.service.d/limits.conf

[Service]

LimitNOFILE=10000

eof

systemctl daemon-reload

systemctl restart mariadb
```

Install InfluxDB

LiCO uses InfluxDB as a time series database for storage monitoring.

Step 1. Run the following commands to install InfluxDB:

a. Install InfluxDB:

```
dnf install -y influxdb
```

b. Start InfluxDB:

```
systemctl enable influxdb --now
```

Step 2. Run the following commands to create InfluxDB users:

a. Enter the InfluxDB shell:

```
influx
```

b. Create database:

```
create database lico
```

c. Use database:

```
use lico
```

d. To create an administrator user, ensure that the password is a string:

```
create user <INFLUX_USERNAME> with password '<INFLUX_PASSWORD>' with all privileges
```

e. Exit the influxDB shell:

```
exit
```

f. Do configuration:

```
sed -i '/# auth-enabled = false/a\ auth-enabled = true' /etc/influxdb/config.toml
```

g. Restart InfluxDB:

```
systemctl restart influxdb
```

Install Confluent

Note: If you need to use the Web console, see "Configure the Confluent Web console" on page 50.

Step 1. Run the following command to install Confluent:

```
dnf install -y confluent_server
```

Step 2. Run the following commands to start confluent:

```
systemctl enable confluent --now
```

Step 3. Run the following commands to create confluent count:

```
source /etc/profile.d/confluent_env.sh

confetty create /users/<CONFLUENT_USERNAME> password=<CONFLUENT_PASSWORD> role=admin
```

Configure user authentication

Install OpenLDAP-server

Note: If OpenLDAP is configured or other authentication services are used in the cluster, skip this section.

OpenLDAP is the open-source version of the lightweight directory access protocol. It is recommended to use OpenLDAP to manage users. However, LiCO also supports other authentication services that compatible with Linux-PAM.

Step 1. Run the following command to install OpenLDAP:

```
dnf install -y slapd-ssl-config openldap-servers
```

Step 2. Run the following commands to modify the configuration file:

```
sed -i "s/dc=hpc,dc=com/${lico_ldap_domain_name}/" /usr/share/openldap-servers/lico.ldif

sed -i "/dc:/s/hpc/${lico_ldap_domain_component}/" /usr/share/openldap-servers/lico.ldif

sed -i "s/dc=hpc,dc=com/${lico_ldap_domain_name}/" /etc/openldap/slapd.conf
```

```
slapadd -v -l /usr/share/openldap-servers/lico.ldif -f /etc/openldap/slapd.conf -b \

${lico_ldap_domain_name}
```

Step 3. Run the following command to get the key:

```
slappasswd
```

Step 4. Edit the /etc/openldap/slapd.conf file to cover the contents of the rootpw with the key obtained.

```
rootpw <ENCRYPT_LDAP_PASSWORD>
```

Step 5. Run the following commands to change the role for configuration file:

```
chown -R ldap:ldap /var/lib/ldap

chown ldap:ldap /etc/openldap/slapd.conf
```

Step 6. Edit configuration files /etc/sysconfig/slapd, and ensure that the following commands are uncommented.

```
SLAPD_URLS="ldapi:/// ldap:/// ldaps:///"

SLAPD_OPTIONS="-f /etc/openldap/slapd.conf"
```

Step 7. Run the following commands to start OpenLDAP service:

```
systemctl enable slapd --now
```

Step 8. Run the following command to check service:

```
systemctl status slapd
```

Install libuser

The libuser module is a recommended toolkit for OpenLDAP. The installation of this module is optional.

Step 1. Run the following command to install libuser:

```
dnf install -y libuser python3-libuser
```

Step 2. Download libuser.conf from https://hpc.lenovo.com/lico/downloads/6.1/examples/conf/ to /etc on the management node, and modify this file referring to the instructions in the file.

Install OpenLDAP-client

Run the following commands to install OpenLDAP-client:

```
echo "TLS_REQCERT never" >> /etc/openldap/ldap.conf

xdcp all /etc/openldap/ldap.conf /etc/openldap/ldap.conf
```

Install nss-pam-ldapd

nss-pam-ldapd is a name service switching module and pluggable authentication module. LiCO uses this module for user authentication.

Step 1. Run the following command to install nss-pam-ldapd on the management node:

```
dnf install -y nss-pam-ldapd
```

Step 2. Run the following command to install nss-pam-ldapd on the other node:

```
psh all dnf install -y nss-pam-ldapd
```

Step 3. Download nslcd.conf from https://hpc.lenovo.com/lico/downloads/6.1/examples/conf/ to /etc on the management node, and modify this file referring to the instructions in the file.

Step 4. Run the following command to modify the privilege of the configuration file:

```
chmod 600 /etc/nslcd.conf
```

Step 5. Run the following command to distribute the configuration:

```
xdcp all /etc/nslcd.conf /etc/nslcd.conf
```

```
psh all chmod 600 /etc/nslcd.conf
```

Step 6. Run the following commands to start service:

```
systemctl enable nslcd --now
```

```
psh all systemctl enable nslcd --now
```

Install authselect-nslcd-config

Step 1. Run the following commands to install authselect-nslcd-config on the management node:

```
dnf install -y authselect-nslcd-config
```

```
authselect select nslcd with-mkhomedir --force
```

Step 2. Run the following commands to install authselect-nslcd-config on the other node:

```
psh all dnf install -y authselect-nslcd-config
```

```
psh all authselect select nslcd with-mkhomedir --force
```

Chapter 4. Install LiCO

List of LiCO components to be installed

Note: In the **Installation node** column, M stands for “Management node”, L stands for “Login node”, and C stands for “Compute node”.

Table 7. List of LiCO components to be installed

Software	Component	Version	Service	Installation node	Notes
lico-core	lico-core	6.1.0	lico	M	
lico-portal	lico-portal	6.1.0		L	
lico-core-extend	lico-confluent-proxy	1.1.4		M	
	lico-vnc-proxy	1.1.4	lico-vnc-proxy	M	
lico-env	lico-ai-scripts	1.0.1		C	Only for AI functions
lico monitor	lico-icinga-mond	1.1.0	lico-icinga-mond	M	
	lico-icinga-plugin	1.1.0		M, C, L	Required if you need to install lico-icinga-mond
	lico-vnc-mond	1.1.2	lico-vnc-mond	C	Required if you need to run VNC
	lico-monitor-tools	1.1.0		M, C, L	Required if you need to install icinga
lico alarm notification	lico-sms-agent	1.2.4	lico-sms-agent	M	Required if you need to send alerts via SMS
	lico-wechat-agent	1.2.4	lico-wechat-agent	M	Required if you need to send alerts via WeChat

Table 7. List of LiCO components to be installed (continued)

Software	Component	Version	Service	Installation node	Notes
	lico-mail-agent	1.3.5	lico-mail-agent	M	Required if you need to send alerts via e-mails
lico manager	lico-file-manager	1.1.2	lico-file-manager	M	Essential components
lico-builder	lico-container-builder	1.0.0	lico-container-builder lico-continer-builder-proxy	M, C, L	Required if you need to build images

Install LiCO on the management node

Step 1. Run the following command to install the LiCO module on the management node:

```
dnf install -y lico-core lico-file-manager lico-confluent-proxy lico-icinga-mond
```

Step 2. Perform the following optional steps as required:

If you need to...	Run the following commands:
Use the VNC component	`dnf install -y lico-vnc-proxy`

Provide e-mail, SMS, and WeChat services	`dnf install -y lico-mail-agent`
	`dnf install -y lico-sms-agent`
	`dnf install -y lico-wechat-agent`

Use a single node to build images	`dnf install -y lico-container-builder`

Step 3. Run the following command to restart services:

```
systemctl restart confluent
```

Install LiCO on the login node

Run the following command to install the LiCO module on the login node:

```
psh login dnf install -y lico-workspace-skeleton lico-portal
```

Install LiCO on the compute nodes

Run the following command to install the LiCO module on the compute nodes:

```
psh compute dnf install -y lico-ai-scripts lico-workspace-skeleton
```

Note: If you need to use the VNC component, see “Configure VNC” on page 49.

Configure the LiCO internal key

Run the following command to configure the LiCO internal key:

```
xdcp all /etc/lico/lico-auth-internal.key /etc/lico/lico-auth-internal.key
```

Chapter 5. Configure LiCO

Configure the service account

Notes:

- The username or password of MariaDB, InfluxDB, Confluent, LDAP are configured in this Guide.
- Obtain the username and password of icinga2 through the `/etc/icinga2/conf.d/api-users.conf` file.

On the management node, use the tool `lico-password-tool`.

Follow the following prompt to enter the username or password for MariaDB, InfluxDB, Confluent, Icinga2, and LDAP:

```
lico-password-tool
```

Configure cluster nodes

Step 1. Run the following command to import the cluster information to the system:

```
cp /etc/lico/nodes.csv.example /etc/lico/nodes.csv
```

Step 2. Run the following command to edit the cluster information file:

```
vi /etc/lico/nodes.csv
```

Notes:

- It is recommended that you download this file to your local PC and edit it using Excel or other table editing software. After that you can upload it to the management node and overwrite the original file.
- If you are facing a large cluster that is already been configured by xCAT, see "Edit nodes.csv from xCAT dumping data" on page 56.

Room information

Below is an example of the room information table.

Table 8. Room information table

room	name	location_description
	Shanghai Solution Room	Shanghai Zhangjiang

Enter one entry of information for the fields **name** and **location_description**.

Logic group information

Managers can use logic groups to divide nodes in the cluster into groups. The logic groups do not impact the use of computer resources or permissions configurations.

Below is an example of the logic group information table.

Table 9. Logic group information table

group	name
	login

You need to enter at least one logic group name in the **name** field.

Room row information

Room row refers to the rack order in the room. Enter the information about the rack row where the cluster node is located.

Below is an example of the room row information table.

Table 10. Room row information table

row	name	index	belonging_room
	row1	1	Shanghai Solution Room

Enter at least one entry of row information in the fields below:

- **name**: row name (must be unique in the same room)
- **index**: row order (must be a positive integer and be unique in the same room)
- **belonging_room**: name of the room where the row belongs

  Note: Add this information to the room information table.

Rack information

Below is an example of the rack information table.

Table 11. Rack information table

rack	name	column	belonging_row
	rack1	1	row1

Enter at least one entry of rack information in the fields below:

- **name**: rack name (must be unique in the same room)
- **column**: rack location column, also known as rack number (must be a positive integer and be unique in the same row)
- **belonging_row**: name of the row where the rack belongs

  Note: Add this information to the row information table.

Chassis information

If there is a chassis in the cluster, enter the chassis information.

Below is an example of the chassis information table.

Table 12. Chassis information table

chassis	name	belonging_rack	location_u_in_rack	machine_type
	chassis1	rack1	7	7X20

The fields in this table are described as follows:

- **name**: chassis name (must be unique in the same room)
- **belonging_rack**: rack location name (must use the name configured in the rack information table)
- **location_u_in_rack**: location of the chassis base in the rack (Unit: U). In a standard cabinet, the value should be between 1 and 42. For example, a chassis base is located at 5U.
- **machine_type**: chassis type (see “Supported servers and chassis models” on page 3)

Node information

Enter the information about all nodes in the cluster into the node information table. Due to its width, the example node information table is displayed in two split parts.

Table 13. Node information table (Part 1)

node	name	nodetype	immip	hostip	machine_ type	ipmi_user
	head	head	10.240.212.13	127.0.0.1	7X58	<BMC_ USERNAME>

Table 14. Node information table (Part 2)

ipmi_pwd	belonging_rack	belonging_chassis	location_u	groups
<BMC_PASSWORD>	rack1		2	login

The fields are described as follows:

- **name**: node hostname (domain name not needed)
- **nodetype**: **head** means management node; **login** means login node; **compute** means compute node.
- **immip**: IP address of the node’s BMC system
- **hostip**: IP address of the node on the host network
- **machine_type**: product name for the node (see “Supported servers and chassis models” on page 3)
- **ipmi_user**: XCC (BMC) account for the node
- **ipmi_pwd**: XCC (BMC) password for the node
- **belonging_rack**: name of the rack where the node is located (need to add the configured name to the rack information table). If the node belongs to a chassis, leave this field blank.
- **belonging_chassis**: name of the chassis where the node is located (need to add the configured name to the chassis information table). If the node belongs to a rack, leave this field blank.
- **location_u**: node location. If the node is located in the chassis, enter the slot in the chassis in which the node is located. If the node is located in a rack, enter the location of the node base in the rack (Unit: U).
- **groups**: name of the node location logic group. One node can belong to multiple logic groups. Group names should be separated by “;”. Configure the logic group name in the logic group information table.

Configure generic resources

This module only executes when the scheduler is slurm. Do one of the following to configure generic resource:

- If no generic resources are configured by default, GPU resource is in cluster and accounting is required, run the following command：

  ```
  cp /etc/lico/gres.csv.example /etc/lico/gres.csv
  ```

- If Slurm is configured with other generic resource, and accounting is required for these resources, run the following command:

  ```
  vi /etc/lico/gres.csv
  ```

  Note: To ensure the historical billing information accuracy, the generic resource removed from `gres.csv` will still remain in the system database.

Gres information

Following is an example of the gres information table：

code	display_name	unit
gpu	GPU	card

Enter at least one entry of generic resource information in the fields below:

- **code**: Code should align with the generic resource type defined in your scheduler. If you have installed LiCO following this document, you can fill it according to your configuration of GresTypes in `slurm.conf`.
- **display_name**: Name of generic resource displayed in the LiCO system. A meaningful display name is recommended.
- **unit**: Unit of resource.

List of cluster services

Note: In the **Installation node** column, M stands for “Management node”, L stands for “Login node”, and C stands for “Compute node”.

Table 15. List of cluster services

Software	Component	Service	Default port	Installation node
lico	lico-core	lico	18080/tcp	M
	lico-confluent-proxy		18081/tcp	M
	lico-vnc-proxy	lico-vnc-proxy	18082/tcp, 18083/tcp	M
	lico-vnc-mond	lico-vnc-mond		C
	lico-sms-agent	lico-sms-agent	18092/tcp	M
	lico-wechat-agent	lico-wechat-agent	18090/tcp	M
	lico-mail-agent	lico-mail-agent	18091/tcp	M
	lico-file-manager	lico-file-manager	18085/tcp	M

Table 15. List of cluster services (continued)

Software	Component	Service	Default port	Installation node
	lico-container-builder	lico-container-builder, lico-container-builder-proxy	18084/tcp, 18086/tcp	M, C, L
lico dependencies	ngnix	ngnix	80/tcp, 443/tcp	L, M
	rabbitmq	rabbitmq-server	5672/tcp	M
	mariadb	mariadb	3306/tcp	
	confluent	confluent	4005/tcp, 13001/tcp	M
	influxdb	influxdb	8086/tcp, 8088/tcp	M
	ldap	slapd	389/tcp,636/tcp	M
		nslcd		M, C, L
cluster	nfs	nfs	111/tcp, 111/udp, 2049/tcp, 2049/udp	M
	chrony	chronyd		M
	slurm	munge		M, C
		slurmctld	6817/tcp	M
		slurmd	6818/tcp	C
	Icinga2	icinga2	5665/tcp, 5665/udp	M, C, L
	dns	named	53/udp	M
	dhcp	dhcpd	67/udp	M

Configure LiCO components

For more inforamtion about configuring LiCO, refer to https://hpc.lenovo.com/lico/downloads/6.1/configuration/host/configuration.html.

lico-vnc-mond

Step 1. Create a file named `/var/tmp/vnc-mond.ini` and run the following commands:

```
[vnc]

url=http://127.0.0.1:18083/api/vnc/session/

timeout=30
```

Note: Replace 127.0.0.1 with the actual IP address of the management node.

Step 2. Run the following command to distribute the configuration file:

```
xdcp compute /var/tmp/vnc-mond.ini /etc/lico/vnc-mond.ini
```

lico-portal

To prevent conflictions, you may need to modify some pathway files for nodes installed with the `lico-portal` module, which provides external Web services with different ports.

/etc/nginx/nginx.conf

You can edit `/etc/nginx/nginx.conf` by changing the port to **8080**:

```
listen 8080 default_server;

listen [::]:8080 default_server;
```

If you want to hide the server version information, modify `/etc/nginx/nginx.conf` by turning off **server_tokens**:

```
http{

......

sendfile on;

server_tokens off;

……

}
```

/etc/nginx/conf.d/https.conf

You can edit `/etc/nginx/conf.d/https.conf` by changing the https default port 443 to other ports:

```
listen <port> ssl http2;
```

Note: Ensure that the port is not used by other applications and is not blocked by the firewall.

/etc/nginx/conf.d/sites-available/lico.conf

You can edit `/etc/nginx/conf.d/sites-available/lico.conf` by replacing the first line to the following content:

```
set $lico_host 127.0.0.1;
```

Note: If `lico-portal` does not run on the management node, you can change 127.0.0.1 to the IP address of the management node.

/etc/lico/portal.conf

You can edit `/etc/lico/portal.conf` by adding custom shortcut links. Refer to `/etc/lico/portal.conf.example` for the configuration format.

Initialize the system

Run the following command to initialize LiCO:

```
lico init
```

Initialize users

Step 1. (Optional) To use LDAP to manage user, find the following configuration in the LiCO configuration file `/etc/lico/lico.ini.d/user.ini` and change the value to “true”:

```
use_libuser = false
```

Step 2. (Optional) Run the following commands to add an LDAP user with username and password:

```
luseradd <HPC_ADMIN_USERNAME> -P <HPC_ADMIN_PASSWORD>

psh all "su - <HPC_ADMIN_USERNAME> -c whoami" | xcoll
```

Note: Use LDAP_PASSWORD configured in "Install OpenLDAP-server" on page 31.

Step 3. Run the following command to import the user to LiCO:

```
lico import_user -u <HPC_ADMIN_USERNAME> -r admin
```

Import system images

Obtain image_bootstrap.zip from https://hpc.lenovo.com/lico/downloads/6.1/images/host/image_bootstrap.zip, and generate and upload LiCO specified images based on the instructions on https://hpc.lenovo.com/lico/downloads/6.1/images/host/readme.html.

Chapter 6. Start and log in to LiCO

Start LiCO

Step 1. Run the following command to start nginx:

```
psh login systemctl enable nginx --now
```

Step 2. Run the following command to start LiCO-related service:

```
systemctl enable lico-file-manager --now
```

Step 3. Run the following commands to start LiCO-Monitor service:

```
systemctl enable lico-icinga-mond --now
```

```
systemctl enable lico-confluent-proxy --now
```

Step 4. (Optional) If use the VNC component, run the following command to start LiCO-vnc service:

```
systemctl enable lico-vnc-proxy --now
```

Step 5. (Optional) To use e-mail, SMS, and WeChat components, run the following commands to start LiCO-notice service：

```
systemctl enable lico-mail-agent --now
```

```
systemctl enable lico-sms-agent --now
```

```
systemctl enable lico-wechat-agent --now
```

Step 6. (Optional) To use a single node to build images, run the following commands to start container-builder service:

```
systemctl enable lico-container-builder --now
```

```
systemctl enable lico-container-builder-proxy --now
```

Step 7. Run the following command to start LiCO:

```
systemctl enable lico --now
```

Step 8. Run the following command to delete the `lico_env.local` file:

```
rm –r /root/lico_env.local
```

Log in to LiCO

After the LiCO service is started, you can access LiCO by opening `https://<ip of login node>:<port>/` in a Web browser.

Note: Replace `port` with the port number you set in `/etc/nginx/conf.d/https.conf` in “lico-portal” on page 43.

If the installation is correct, the LiCO login page opens. You can log in using the LDAP account set in “Initialize users” on page 44.

Configure LiCO services

The LiCO service configuration file is located in `/etc/lico/lico.ini` and `/etc/lico/nodes.csv`. This configuration file controls the operating parameters for various LiCO background service components. You can modify this configuration file as needed.

If the configuration or the operating status of components mentioned in this document is changed when LiCO is running, run the following command to restart LiCO:

```
systemctl restart lico
```

Note: Configurations not mentioned in the instructions in this section can be modified after consulting with service staff. Modifications made without a service consultation could result in a system failure.

Chapter 7. Appendix: Important information

Configure VNC

Install the VNC module only on compute nodes that require the VNC functionality and GUI.

Standalone VNC installation

To install VNC on a compute node, follow the steps in this section.

Step 1. Run the following commands on a compute node where you want to install the VNC function:

```
dnf install -y gdm tigervnc tigervnc-server
```

```
dnf install -y lico-vnc-mond
```

Step 2. Edit `/etc/gdm/custom.conf` on the compute node, and make the following changes:

```
[xdmcp]
```

```
Enable=true
```

Step 3. Run the following commands on the compute node to start VNC:

```
systemctl start lico-vnc-mond
```

```
vncserver -securitytypes=none
```

VNC batch installation

To install VNC on all compute nodes, do the following:

Step 1. Install VNC:

```
psh compute dnf install -y lico-vnc-mond
```

```
psh compute dnf install -y gdm tigervnc tigervnc-server
```

Step 2. Edit `/etc/gdm/custom.conf`, and make the following changes:

```
[xdmcp]
```

```
Enable=true
```

Step 3. Distribute profile:

```
xdcp compute /etc/gdm/custom.conf /etc/gdm/custom.conf
```

Step 4. Start VNC:

```
psh compute systemctl start lico-vnc-mond
```

```
psh compute vncserver -securitytypes=none
```

Configure the Confluent Web console

To open the management node console from LiCO web portal, configure and restart the management node before the configurations take effect.

For CentOS

Step 1. Edit the `/etc/default/grub` file by adding the following text to the end of `GRUB_CMDLINE_LINUX`:

For the ThinkSystem SR635/SR655 server, add:

```
console=ttyS1,115200
```

For other server models, add:

```
console=ttyS0,115200
```

Step 2. Start the UEFI mode or legacy mode.

To start the legacy mode, run the following command:

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```

To start the UEFI mode, run the following command:

```
grub2-mkconfig -o /boot/efi/EFI/centos/grub.cfg
```

For RHEL

Step 1. Edit the `/etc/default/grub` file by adding the following text to the end of `GRUB_CMDLINE_LINUX`:

For the ThinkSystem SR635/SR655 server, add:

```
console=ttyS1,115200
```

For other server models, add:

```
console=ttyS0,115200
```

Step 2. Start the UEFI mode or legacy mode.

To start the legacy mode, run the following command:

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```

To start the UEFI mode, run the following command:

```
grub2-mkconfig -o /boot/efi/EFI/redhat/grub.cfg
```

LiCO commands

Change a user’s role

Run the following commands to change a user’s role:

```
lico change_user_role -u <ROLE_USERNAME> -r admin

Parameter interpretation:
```

```
-u Specify the username to modify
```

```
-r Specify the role to be set (admin/operator/user)
```

Resume a user

Run the following command to resume a user:

```
lico resume_user -u <SUSPENDED_USERNAME>
```

Parameter interpretation:

Directly specify the user who need to be resumed.

Delete a user

Run the following command to delete a user:

```
lico delete_user -u <DELETED_USERNAME>
```

Parameter interpretation:

Directly specify the user who need to be deleted.

Import a user

For more information, refer to: “Initialize users” on page 44.

Import AI images

For more information, refer to: “Import system images” on page 45.

Generate nodes.csv in confluent

If the confluent is in the cluster with the cluster information configured, you can generate the cluster configuration file of LiCO by using the following command:

```
lico export_nodes_from_confluent
```

Notes:

- Before running this command, ensure that the confluent management node information is configured in LiCO.
- After running this command, `export_nodes.csv` will be generated in the current directory by default. You should rename the file to `nodes.csv` and reedit it according to “Configure cluster nodes” on page 39.
- For more information about parameters of the command, refer to the help file of the command.

Firewall settings

Considering the security of the system, we recommend that you enable the firewall on the management node and the login node.

Run the following commands to install and enable the firewall:

```
dnf install -y firewalld
```

```
systemctl enable firewalld
systemctl start firewalld
```

Note: If you have set up the cluster and installed LiCO following this document, you can set up your firewall referring to the official firewall setup document: https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/7/html/security_guide/sec-using_firewalls.

Set the firewall on the management node

Step 1. Do the following to add all ports:

a. Add RPC application port by running the following commands:

Note: All the following ports are default settings. You can check your settings by running the **rpcinfo -p** command.

```
firewall-cmd --zone=public --add-port=111/tcp --permanent
```

```
firewall-cmd --zone=public --add-port=111/udp --permanent
```

```
firewall-cmd --zone=public --add-port=2049/tcp --permanent
```

```
firewall-cmd --zone=public --add-port=2049/udp --permanent
```

```
firewall-cmd --zone=public --add-port=20048/tcp --permanent
```

```
firewall-cmd --zone=public --add-port=20048/udp --permanent
```

```
firewall-cmd --zone=public --add-port=52891/udp --permanent
```

```
firewall-cmd --zone=public --add-port=33504/tcp --permanent
```

```
firewall-cmd --zone=public --add-port=39123/tcp --permanent
```

```
firewall-cmd --zone=public --add-port=52656/udp --permanent
```

b. Add SSH service port by running the following command:

```
firewall-cmd --zone=public --add-port=22/tcp --permanent
```

c. Add nginx service port by running the following command:

Note: You can configure Port 443 based on your actual conditions.

```
firewall-cmd --zone=public --add-port=443/tcp --permanent
```

d. Add httpd service port by running the following command:

```
firewall-cmd --zone=public --add-port=80/tcp --permanent
```

e. Add Icinga2 gmond port by running the following command:

```
firewall-cmd --zone=public --add-port=5665/udp --permanent
```

f. Add Slurm slurmctld port by running the following command:

```
firewall-cmd --zone=public --add-port=6817/tcp --permanent
```

g. Add OpenLDAP slapd port by running the following commands:

```
firewall-cmd --zone=public --add-port=636/tcp --permanent
```

```
firewall-cmd --zone=public --add-port=389/tcp --permanent
```

h. Add lico-confluent-proxy port by running the following command:

```
firewall-cmd --zone=public --add-port=18081/tcp --permanent
```

i. Add lico-core port by running the following command:

```
firewall-cmd --zone=public --add-port=18080/tcp --permanent
```

j. Add ports managed by LiCO:

```
firewall-cmd --zone=public --add-port=25000-27500/tcp --permanent
```

k. Add DNS service port by running the following command:

```
firewall-cmd --zone=public --add-port=53/udp --permanent
```

l. Add DHCP service port by running the following command:

```
firewall-cmd --zone=public --add-port=67/udp --permanent
```

Step 2. Run the following commands to add the internal network interface into the public zone:

```
firewall-cmd --zone=public --add-interface=eth0 --permanent

firewall-cmd --zone=public --add-interface=eth1 --permanent
```

Note: For eth0 and eth1, refer to your internal and external network interface.

Step 3. Run the following command to enable roles:

```
firewall-cmd --complete-reload
```

Set the firewall on the login node

Step 1. Run the following commands to add roles to public zone:

```
firewall-cmd --zone=public --add-port=22/tcp --permanent
```

- Add nginx service port, you can adjust 443 to your setting:

```
firewall-cmd --zone=public --add-port=443/tcp --permanent
```

- Add ports managed by LiCO:

```
firewall-cmd --zone=public --add-port=25000-27500/tcp --permanent
```

Step 2. Run the following commands to add the internal and external network interface into the public zone:

```
firewall-cmd --zone=public --add-interface=eth0 --permanent

firewall-cmd --zone=public --add-interface=eth1 --permanent
```

Note: eth0 and eth1 refer to your internal and external network interface.

Step 3. Run the following command to enable roles:

```
firewall-cmd --complete-reload
```

SSHD settings

If you want to prevent the security vulnerabilities of system, or if there are some security issues with SSHD, we recommend you to change the default settings.

Improve SSHD security

Step 1. Run the following commands to configure SSHD:

```
echo "Ciphers aes128-ctr,aes192-ctr,aes256-ctr" >> /etc/ssh/sshd_config

echo "MACs hmac-sha2-512-etm@openssh.com,hmac-sha2-256-etm@openssh.com,umac-128-etm@\

openssh.com,hmac-sha2-512,hmac-sha2-256,hmac-ripemd160,hmac-sha1" >> /etc/ssh/sshd_config
```

Step 2. Run the following command to restart SSHD:

```
systemctl restart sshd
```

Slurm issues troubleshooting

This section lists the solutions to some Slurm issues.

Node status check

You can use the Slurm command **sinfo** to check the node status.

- If the status is **drain**, you can use the following command to change the node status to normal:

  ```
  scontrol update NodeName=host1 State=RESUME
  ```

- If the node status is **down**, you are recommended to do the following:
  - Use the Slurm command **scontrol show nodes** to view the detailed node information and view the **reason** in the output of this command.
  - Check to ensure that all nodes have the same `slurm.conf` file under `/etc/slurm`.
  - Check to ensure that the **slurmd** and **munge** services are active on all the nodes, and that the **slurmctld** service is active on the management node.
  - Check to ensure that all nodes have the same date and that the **ntpd** service is active on all nodes.

Memory allocation error

When you use **srun** or **prun** to run an MPI program, you may encounter the following error message:

```
Failed to create a completion queue (CQ):

……

Error: Cannot allocate memory
```

In this case, check whether soft memlock and hard memlock are unlimited in the file `/etc/security/limits.conf` on the management node and the compute nodes.

If the memlocks are limited, run the following commands to set them as unlimited and then restart the nodes before the changes take effect:

```
echo '* soft memlock unlimited' >> /etc/security/limits.conf

echo '* hard memlock unlimited' >> /etc/security/limits.conf

reboot
```

Status setting error

If you set the slurm queue node status to “DOWN”, but the status is automatically changed to “IDLE”, you can edit slurm.conf file in `/etc/slurm`, and set the ReturnToService to 0: ReturnToService=0.

InfiniBand issues troubleshooting

To install the InfiniBand driver, run the following command on the management node to change file `/root/.rpmmacros`. If necessary, back up this file.

```
psh all "sed -i 's/%_excludedocs 1/%_excludedocs 0/' /root/.rpmmacros"
```

Installation issues troubleshooting

When you are installing the NVIDIA driver and CUDA, the following error might be reported:

```
nvidia-modeset: Loading NVIDIA Kernel Mode Setting Driver for UNIX platforms 396.26

Spectre V2 : System may be vulnerable to spectre v2

nvidia_drm: loading module not compiled with retpoline compiler
```

In this case, upgrade the kernel, kernel-devel, gcc and gcc=c++ packages. Then repeat the steps of “Install the GPU driver” on page 18.

Note: The RPMs and non-RPM-controller files from the previous GPU driver and CUDA installs should be removed first.

When you are running makenetworks, the following error might be reported:

```
Warning: The network entry '10_16_0_0-255_255_0_0' already exists in xCAT networks table.

Cannot create a definition for '10_16_0_0-255_255_0_0'

.......
```

This will not cause any functional issue. You can ignore the error.

When you are running makedhcp -n, the following error might be reported:

```
Warning: No dynamic range specified for 10.16.0.0. If hardware discovery is being used, a dynamic range

is required

……
```

This will not cause any functional issue. You can ignore the error.

XCAT issues troubleshooting

If timeout message is displayed when pushing operating system to other nodes, log in to BMC of the nodes, and click **BMC Configuration → Network** to check whether **IPMI over LAN** is enabled.

When you have problems in using XCAT, go to: https://hpc.lenovo.com/users/documentation/

Running job issue troubleshooting

When you run a GPU job, the following error message might be displayed:

```
failed call to cuInit: CUDA_ERROR_UNKNOWN: unknown error

retrieving CUDA diagnostic information for host: c1
```

In this case, run the following commands on the management node:

```
psh compute modprobe nvidia-uvm

psh compute nvidia-modprobe -u -c=0
```

MPI issues troubleshooting

When running an Open MPI program, the following error might be displayed:

```
WARNING: Open MPI accepted a TCP connection from what appears to be a another Open MPI process

but cannot find a corresponding process entry for that peer.
```

If the TCP connection is ignored, the Open MPI program might not be executed properly.

When the Open MPI program uses the unroutable USB NICs, whose name might be “enp0s20f0u1u6” or similar under RedHat/CentOS 8 ， this warning might be displayed. Select one of the following workarounds to resolve this issue:

Note: In the following workarounds, change <USB_NIC_NAME> to the absolute name of your unroutable USB NICs based on your actual conditions.

- Disable the USB NICs on all nodes by running the following command:

  ```
  psh all ifconfig <USB_NIC_NAME> down
  ```

  Note: This step might interrupt the running Lenovo management tools, such as OneCLI. When you want to use OneCLI, re-enable the NICs for a while.
- Instruct Open MPI to ignore the NICs by running the following command:

  ```
  mpirun --mca btl_tcp_if_exclude <USB_NIC_NAME>
  ```

  Note: It is recommended to create the custom system-wide MPI templates.
- Permanently disable USB NICs by running the following command:

  ```
  rmmod cdc_ether
  ```

  Note: This step might permanently disable OneCLI and other Lenovo managenet tools.

Edit nodes.csv from xCAT dumping data

When installing LiCO on a large cluster that has been configured in xCAT, you can run the following commands to edit the `nodes.csv` file:

Step 1. Dump nodes information from xCAT:

```
tabdump -f <save_filename> nodelist
```

Step 2. Dump BMC information from xCAT:

```
tabdump -f <save_filename> ipmi
```

Step 3. Dump MAC information from xCAT:

```
tabdump -f <save_filename> mac
```

Note: The dumping files are in CSV format. You can easily edit these files and merge them into your `nodes.csv` file using Excel or any other table editor.

Notices and trademarks

Notices

Lenovo may not offer the products, services, or features discussed in this document in all countries. Consult your local Lenovo representative for information on the products and services currently available in your area. Any reference to a Lenovo product, program, or service is not intended to state or imply that only that Lenovo product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any Lenovo intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any other product, program, or service.

Lenovo may have patents or pending patent programs covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

Lenovo (United States), Inc.
8001 Development Drive
Morrisville, NC 27560
U.S.A.
Attention: Lenovo Director of Licensing

LENOVO PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

Changes are made periodically to the information herein; these changes will be incorporated in new editions of the publication. To provide better service, Lenovo reserves the right to improve and/or modify the products and software programs described in the manuals included with your computer, and the content of the manual, at any time without additional notice.

The software interface and function and hardware configuration described in the manuals included with your computer might not match exactly the actual configuration of the computer that you purchase. For the configuration of the product, refer to the related contract (if any) or product packing list, or consult the distributor for the product sales. Lenovo may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

The products described in this document are not intended for use in implantation or other life support applications where malfunction may result in injury or death to persons. The information contained in this document does not affect or change Lenovo product specifications or warranties. Nothing in this document shall operate as an express or implied license or indemnity under the intellectual property rights of Lenovo or third parties. All information contained in this document was obtained in specific environments and is presented as an illustration. The result obtained in other operating environments may vary.

Lenovo may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Any references in this publication to non-Lenovo Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this Lenovo product, and use of those Web sites is at your own risk.

Any performance data contained herein was determined in a controlled environment. Therefore, the result obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

This document is copyrighted by Lenovo and is not covered by any open source license, including any Linux agreement(s) which may accompany software included with this product. Lenovo may update this document at any time without notice.

For the latest information or any questions or comments, contact or visit the Lenovo Web site:

https://support.lenovo.com

Trademarks

LENOVO, LENOVO logo, THINKPAD, THINKPAD logo, TRACKPOINT, ULTRACONNECT, and Yoga are trademarks of Lenovo. Microsoft, Windows, Direct3D, BitLocker, and Cortana are trademarks of the Microsoft group of companies. Ubuntu is a registered trademark of Canonical Ltd. The terms HDMI and HDMI High-Definition Multimedia Interface are trademarks or registered trademarks of HDMI Licensing LLC in the United States and other countries. Wi-Fi, Wi-Fi Alliance, and Miracast are registered trademarks of Wi-Fi Alliance. USB-C is a trademark of USB Implementers Forum. All other trademarks are the property of their respective owners. © 2020 Lenovo.