Freja Installation Instructions

1 Prerequisite Actions for LITP Installation

Context Description:

- This section describes the prerequisites that must be in place before installing LITP.

Expected Result:

- You should be able to proceed to Boot LMS step from here.

Steps

1. Get LITP ISO+TORISO
   You have fetched the LITP ISO (CXP 902 1359) software media from GASK.

2. Obtain Deployment package from Solution Architect
   You should have a completed Site Engineering spreadsheet that lists all of the customer-specific data such as IP addresses, iLOs, MAC addresses and so on for your deployment. In addition you should receive the deployment description definition file and the deployment description inventory file which will need to be ftp'd to the LMS.

3. Check that iLO Licenses are activated
   HP blades are controlled through the iLO (Integrated Lights Out) interface, therefore HP advanced iLO License must be active on the LMS before installing LITP in order to mount the ISO by remote access. To install license follow 1.1 Add iLO 2 Licence

4. Ensure that HBA is set up on each managed node.
   If the blades have never been used in a LITP deployment follow 1.2 Setting up HBA on each blade .

5. Ensure that Disk Mirroring on the LITP management Server is configured.
   If this has not been done follow 1.3 Disk Mirroring on the Management Server.

6. OSS on Blade Cabling Instructions have been followed with TOR specific updates.

7. Ensure a redundant heartbeat has been been set up between the managed node blades.
   If this has not been specifically outlined in the OSS Blade Instructions above follow 1.4 VCS Network Configurations.

8. Ensure that you have the following browsers installed and that there is an up to date Java plug-in installed in the browser.
   - Windows Internet Explorer or Firefox for remote console
   - Windows Internet Explorer or Chrome for EMC-related tasks

9. OSS RC-has been upgraded to 13B

10. That the TOR hardware BOM is installed.
    This needs to have at minimum, 2 half-height G-8 blades with connections to the Storage, OSS Services, and OCS Services VLans, 1 Rack mounted G6 DL380 server with access to the Storage VLan and the Blade Enclosure housing the blades,a VNX SAN with free disks attached to the Storage VLan, an SFS installed and attached to Storage VLan

Next Step:

- Proceed to Boot LMS

1.1 Add iLO 2 Licence

Context Description

If a version of iLO2 does not have a licence key entered then it may not be possible to open a console. If you are prompted to enter an iLO 2 License key carry out the steps below:

Prerequisites

1. The iLO IP address and root password is known.
2. You have the site specific iLO licence number.

Result

ILO2 will be licensced.

Steps

1. Log in to the iLO web interface.
2. Go to the Administration tab and select licensing from the side menu.
3. Input the activation key in the space provided.

Post Requirement

This should enable the user to mount an ISO as well as use the ILO2 consoles.

1.2 Setting up HBA on each blade

Context Description

You only need to carry out the instructions in this section if the blades have never been used in a deployment. This step needs to be done on each blade in the deployment

Prerequisites

1. The iLO IP address of each blade and root password is known.

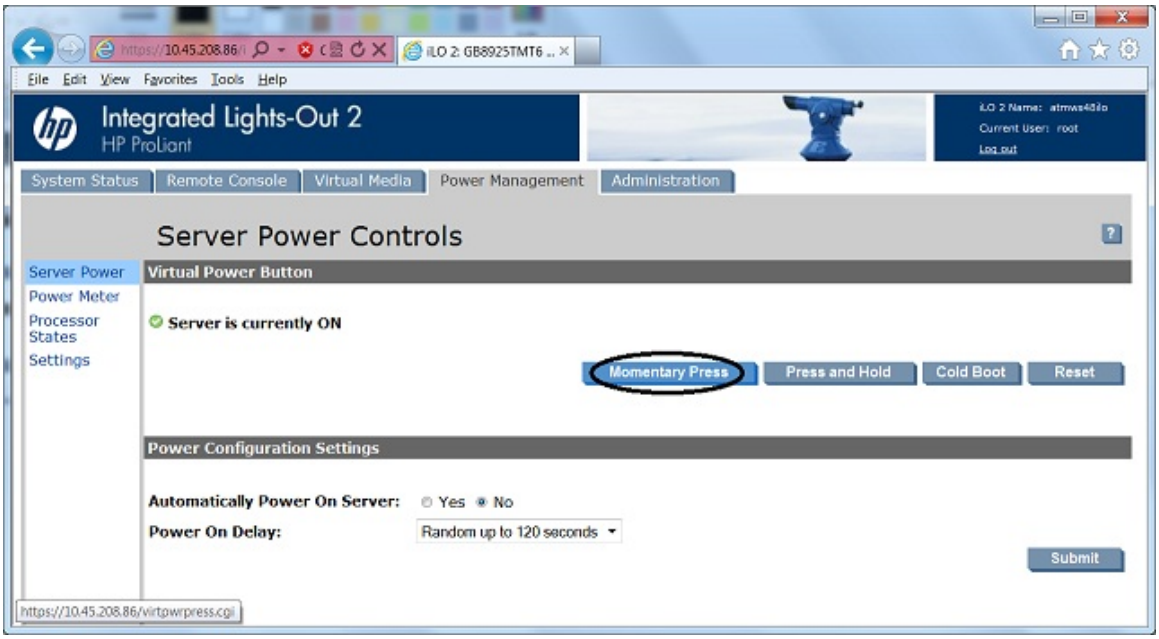
Result

Each blade will have Host Adapter BIOS enabled in both adapters through the QLogic FastUtil interface.

Steps

1. Log in to blade web iLO and open a console connection.

2. Reboot the blade.



3. During the boot up sequence you will be prompted to press Ctrl-Q. Press **CTRL-Q** to enter QLogic FastUtil.

4. Identify each of the two host adapters connected to the SAN.

5. For each of these, select **Host Adapter** (press ENTER) -> **Configuration Settings** -> **Adapter Settings** -> **Host Adapter BIOS**

6. Change the Host Adapter BIOS setting to **Enabled**

7. Exit and save settings

8. Choose Reboot option to apply and repeat steps 4 to 8 if necessary.
9. Repeat steps 1 to 8 for the second blade.

Post Requirement

Continue with normal workflow.

1.3 Disk Mirroring on the Management Server

Context Description

If the MS hardware has two disks, you are recommended to configure disk mirroring on the hardware level. This implements disk redundancy.
To check how many disks are present and configure disk mirroring, take the following steps:

Prerequisites

A rack mounted server with multiple disks must be installed.

Result

Disk Mirroring on Rack Mounted server should be configured by the end of this process.

Steps

1. Check the internal drive configuration, connect to the MS iLO using IE browser, then chose Remote Console -> Integrated Remote Console and reboot the server.

	The number of drives is listed under "Available Phyiscal Drives" in figure 2. The internal drive configuration is listed under "RAID Configurations" in figure 2 A reboot is required to Display the Smart Array Controller and RAID Configuration Screens (figures 1 and 2)

2. When the system is booting Press the escape key when prompted to view the Option ROM messages.

3. When you reach the HP Smart Array Controller POST message there should be either one or two logical drives configured based on your server type. See figure 1.

Figure 1: HP Smart Array Configuration

4. If there are two logical drives configured in the HP Smart Array, press F8 to run the Option ROM Configuration for Array Utility and delete any exiting logical drives.

	If you press F8 too quickly, you may go to "iLO 4 utility" instead of "Option ROM Configuration for Arrays". If this happens, exit "iLO 4 utility" and then press F8 again.

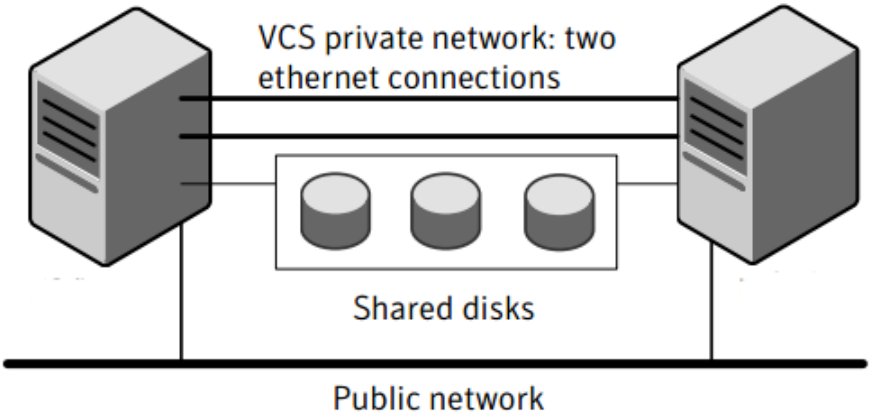
5. Create a mirrored logical drive as RAID1+0 as shown in figure 2.

Figure 2: HP Smart Array Mirror Logical Drive Creation

1.4 VCS Network Configurations

Context Description

For the VCS private network, two network channels must be available to carry heartbeat information. These network connections also transmit other VCS-related information. Each cluster configuration requires at least two network channels between the systems. The requirement for two channels protects your cluster against network partitioning. VCS heartbeat links make use of proprietary protocol LLT. The Ethernet networks for blades are defined in virtual connect and then assigned to server profile or blade profile.



Prerequisites

1. The TOR blades must be commissioned as described in document **OSS on Blades Cabling Installation Instruction** listed in the Release Note.
2. The HP Blade Enclosure Onboard Administrator IP addess and root password must be known.
3. The Server Name, Serial Number or Device Bay of the TOR blades must be known.

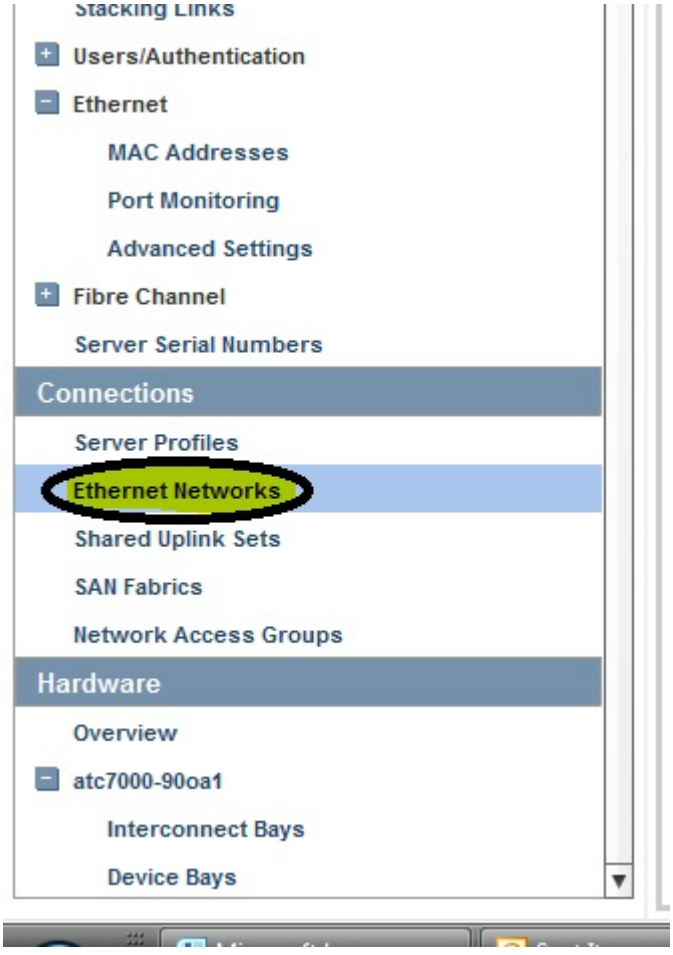
Result

The outcome of this task is a private VCS Ethernet network that carries heartbeat signaling between SC-1 and SC-2.

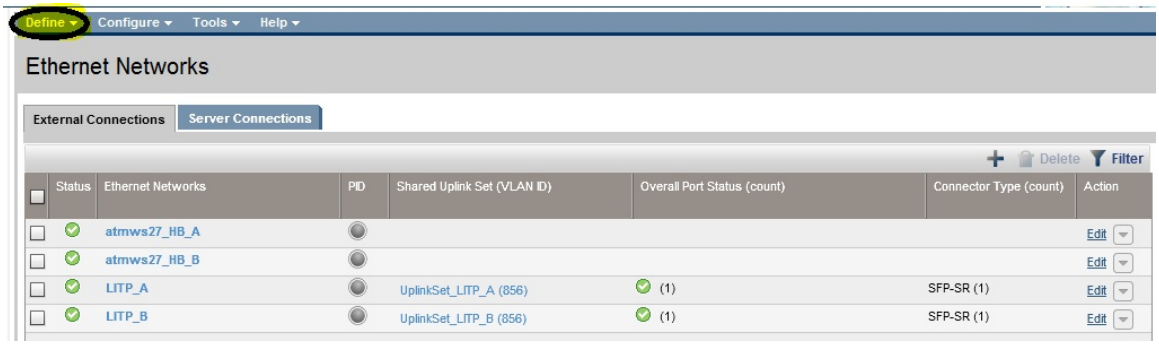
Steps

1. Log in to the HP Blade Enclosure Onboard Administrator.
2. Locate the TOR blades and power them off.
3. Click on virtual connect manager.

4. On virtual connect manager page, click on "Ethernet Networks" on the left side of the page

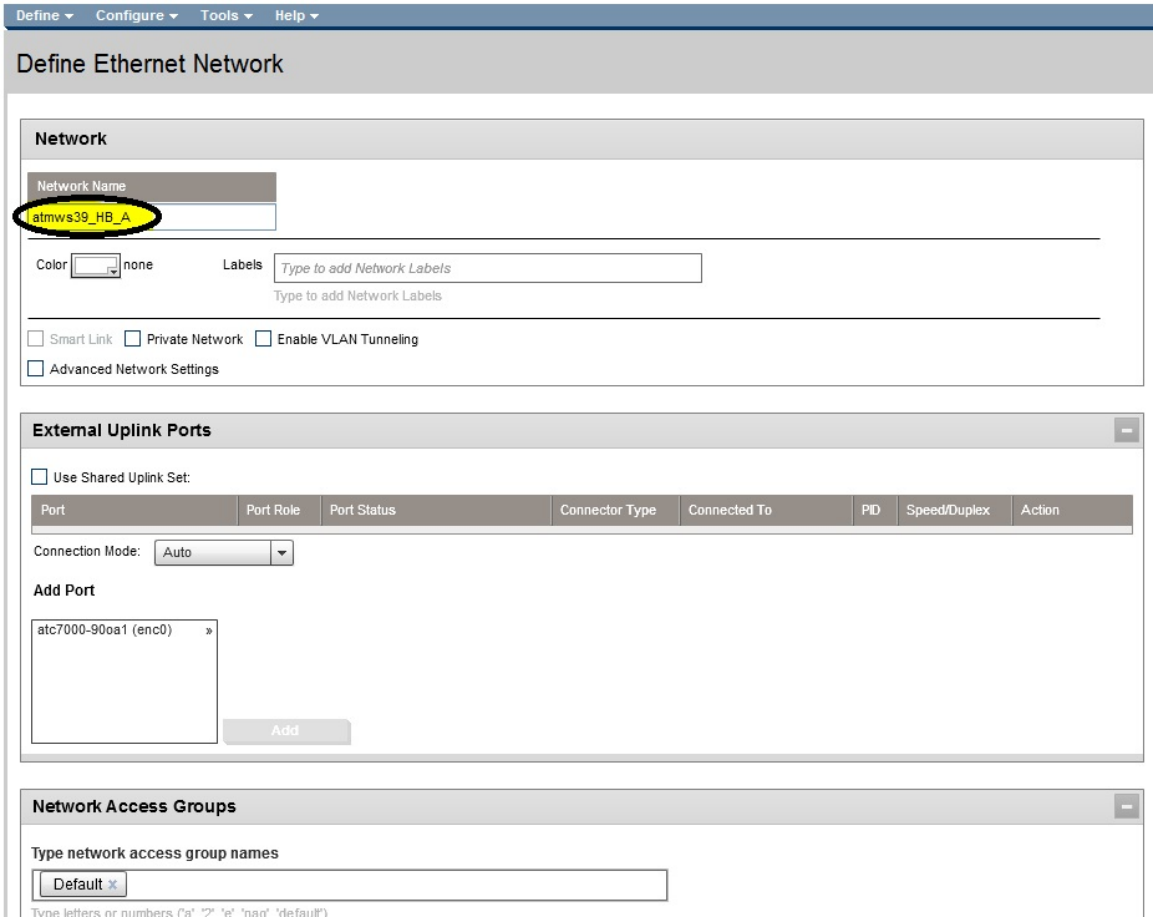


5. Select Define, and Select Ethernet Network (first option)

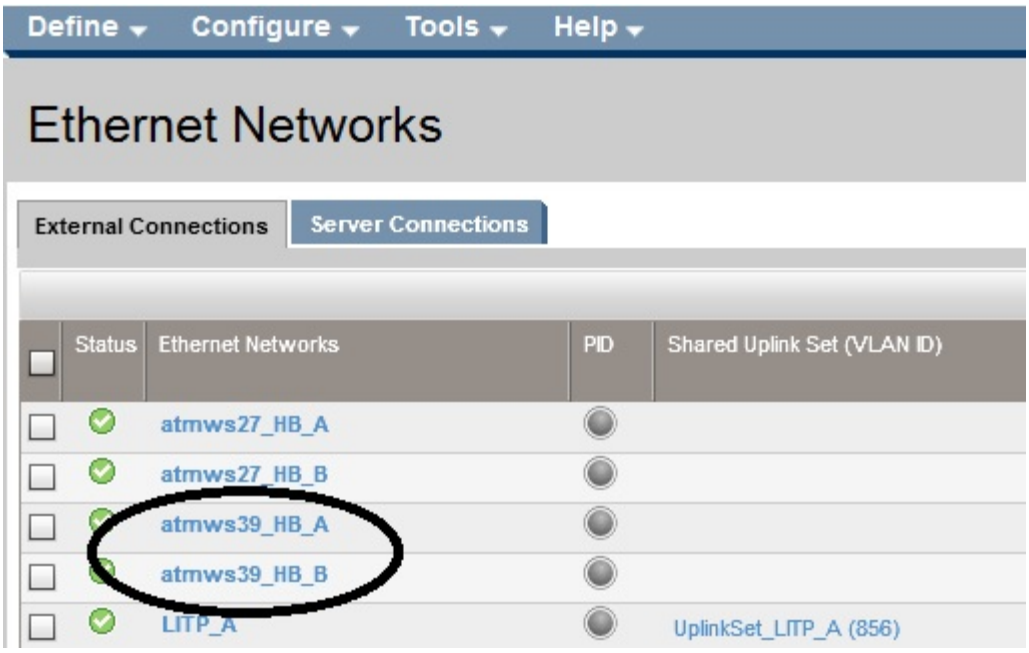


6. Create two new Ethernet networks, for eg: atmws39_HB_A and atmws39_HB_B, and apply the changes on the bottom of the page.

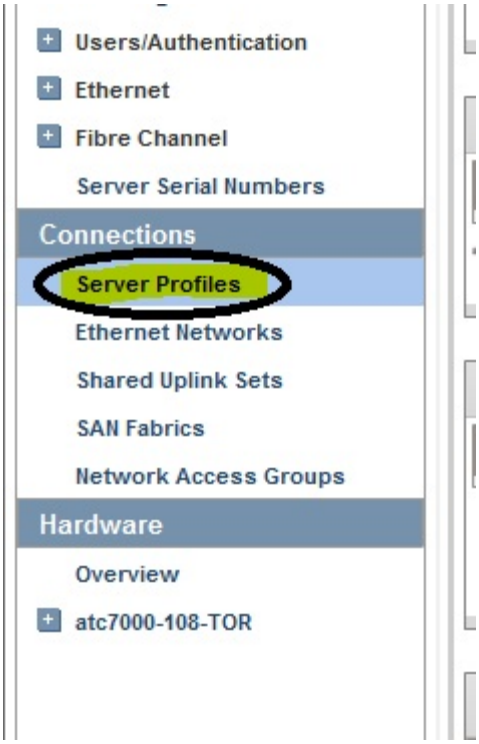
   NOTE: Leave all the options blank, since it is not using any uplinks



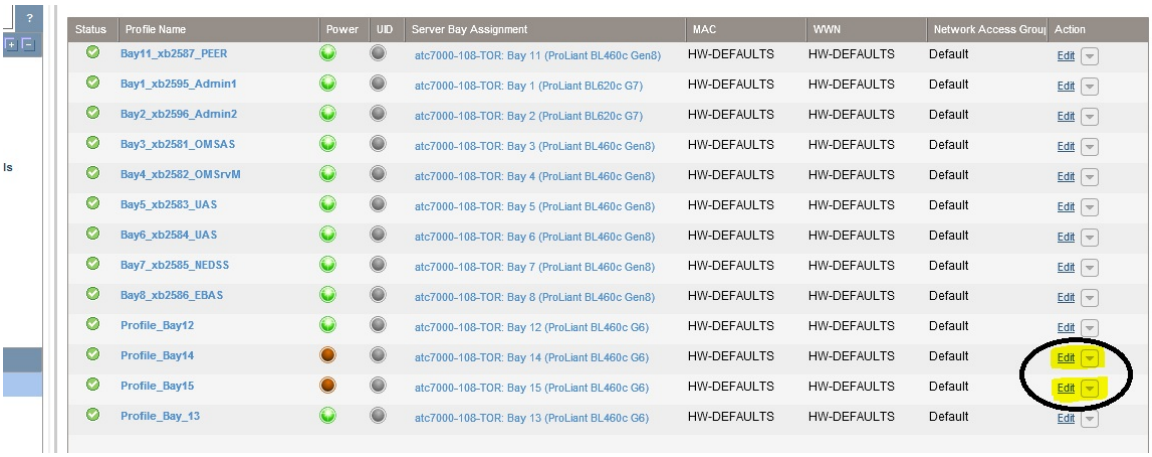
After having created the two networks, they appear as shown below.



7. Assign the newly created two Ethernet networks to your blade profiles.

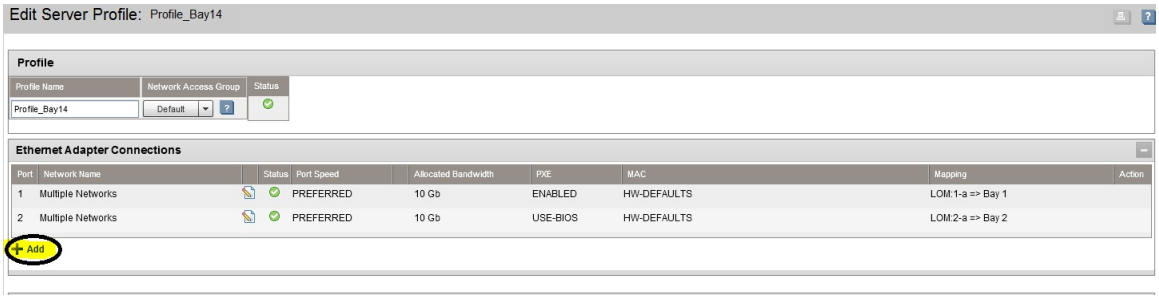


8. Click on edit (repeat the same for both blades SC-1 and SC-2)

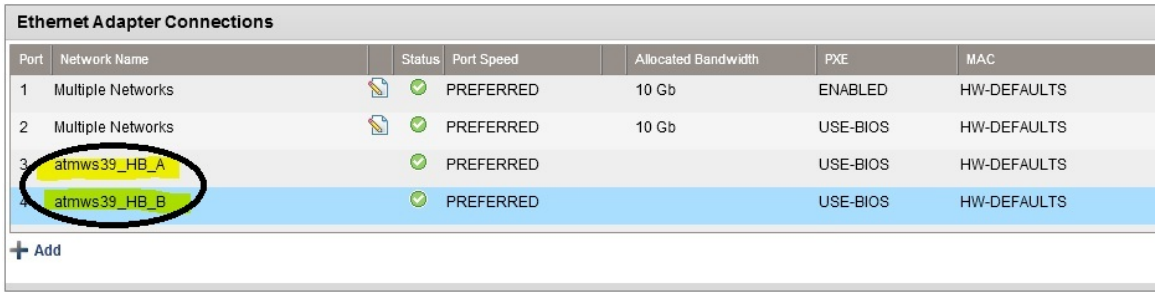


9. Click on "Add"

   Note: If "Add" option is not visible, right click on Adapter Connections and select "Add connection".



10. Select Networks, select newly created HB network. Add two networks.



11. Click "Apply"

Post Requirement

Proceed to 2 Boot LMS.

2 Boot LMS

Context Description:

- This section describes how to boot the LITP Management Server (LMS).

Prerequisites

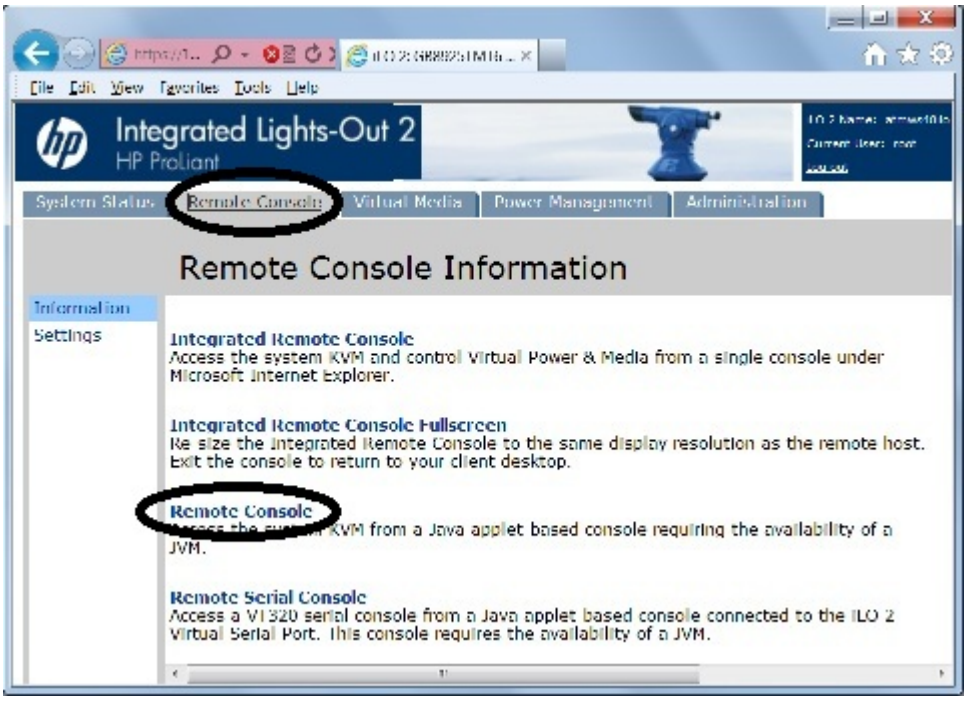
1. It is assumed that the LMS hardware has already been installed and connected to the customer switch and firewall infrastructure.
2. Before booting the LMS, ensure you have access to the LITP ISO file or the physical DVD. You find the ID and version of the LITP ISO to use in the **Release Binder** and the LITP ISO itself in GASK.
3. LMS iLO IP address and root password must be known.

Expected Result:

- At the ned of the following steps the LMS boots up and prompts for starting initial installation.

Steps

1. Log in to LMS iLO and open a Java applet based remote console connection.

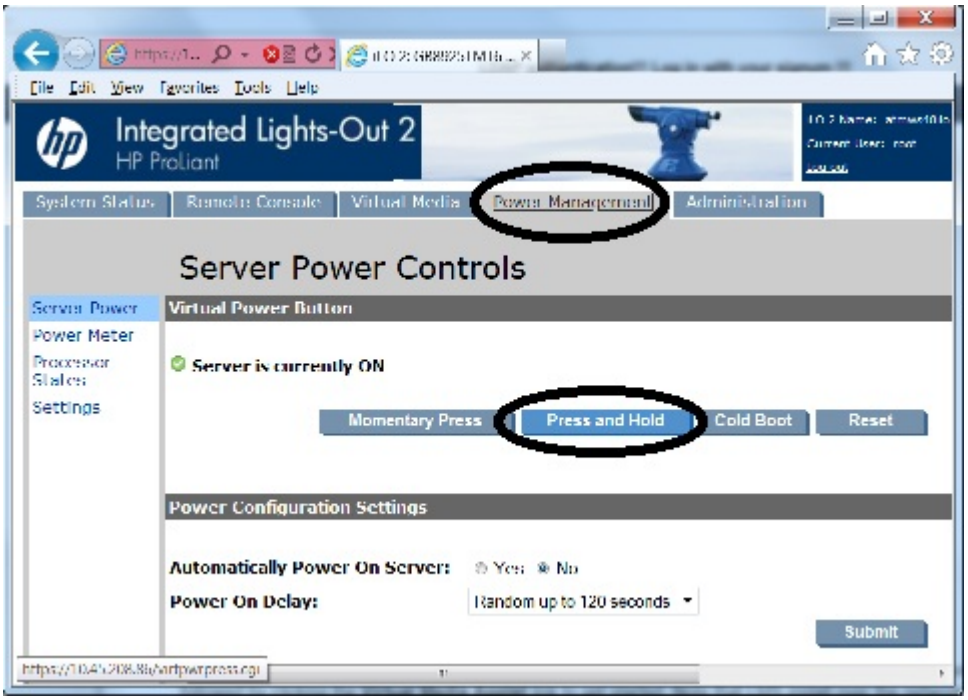


2. Insert the DVD into the DVD drive of the LMS.

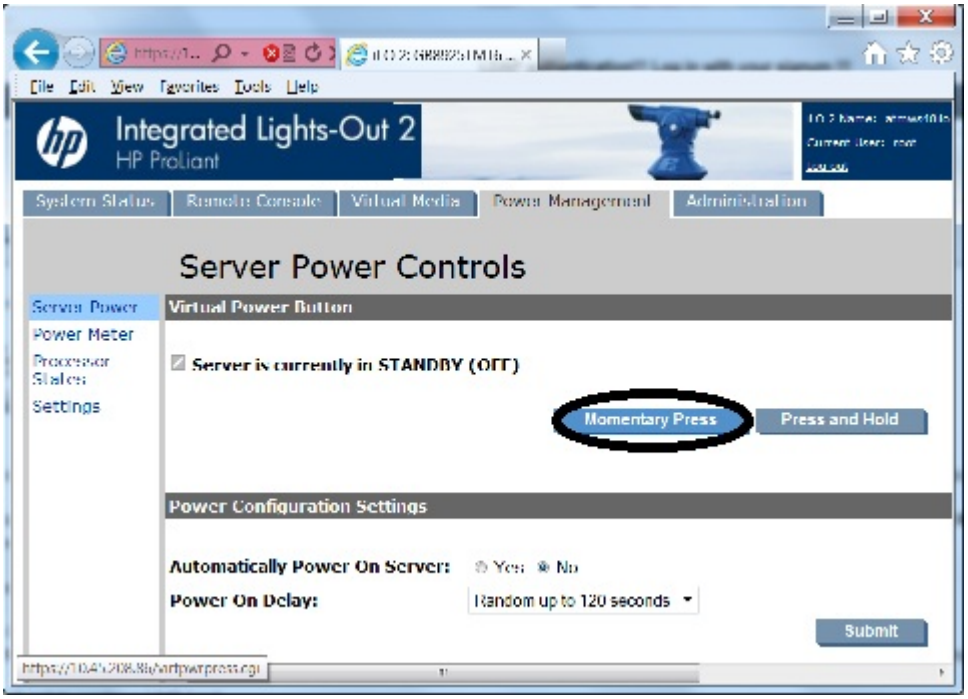
   Note: There is an option to install LMS over the network instead of using a DVD inserted into LMS. If you want to install over the network, select the **Virtual Media** tab in the iLO followed by clicking the **Virtual Media**

Applet link to get started. Note that LMS install over the network takes considerably longer than installing from a DVD inserted into its DVD drive.

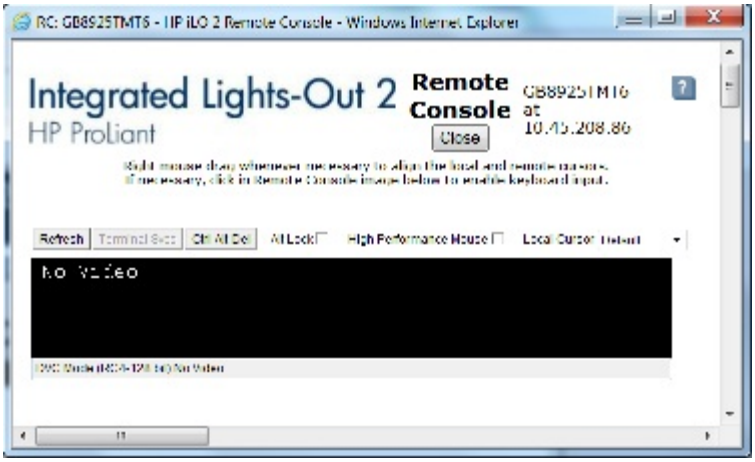
3. Power cycle the server. In the **Power Management** tab click **Press and Hold**.



4. When the server power is off, click **Momentary Press**.

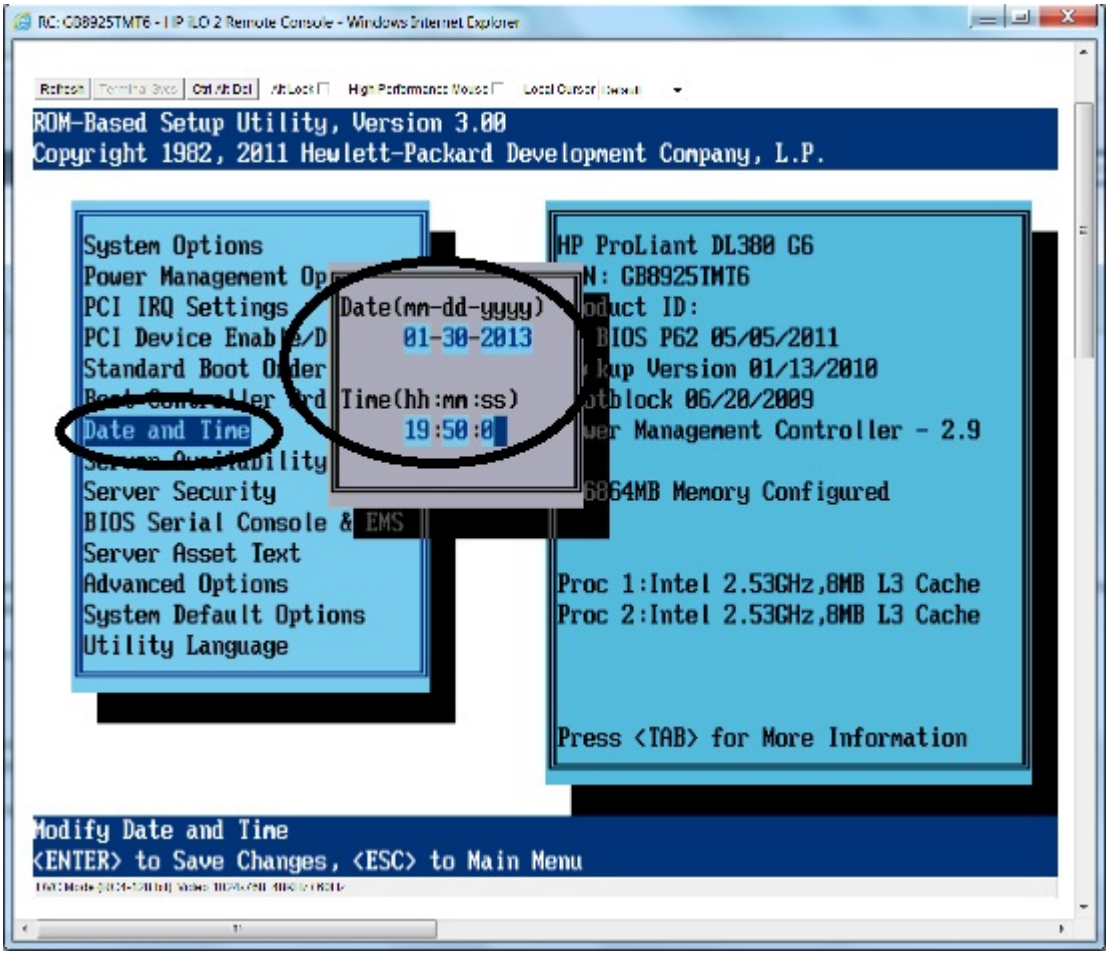


5. In the console you will see message **No Video** being displayed a minute or two.

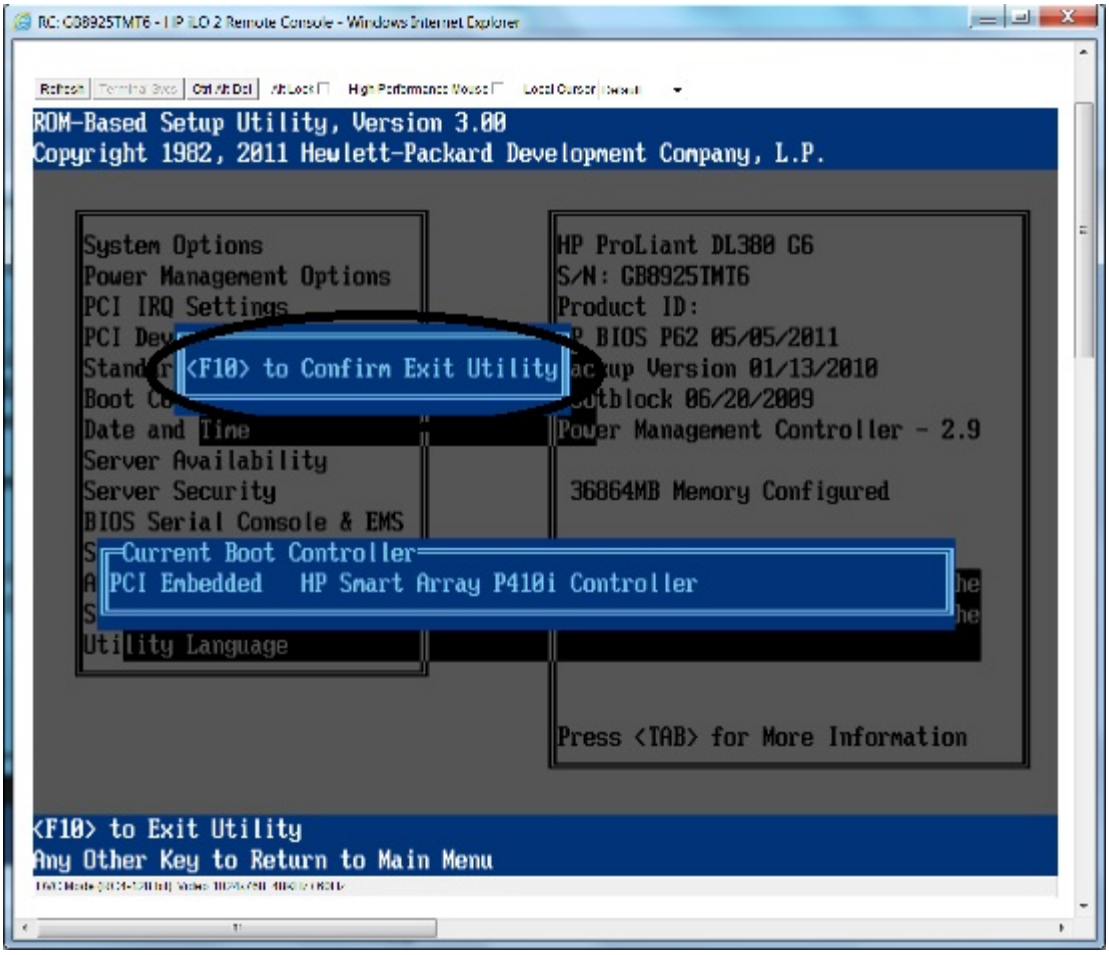


6. While the LMS is rebooting press F9 when asked to, to view BIOS settings. Step down to **Date and Time** usi ng the arrow keys and press **Enter** to edit. If BIOS date and/or time is incorrect, enter current date and time and press **Enter** to save. Press **Esc** to exit BIOS.

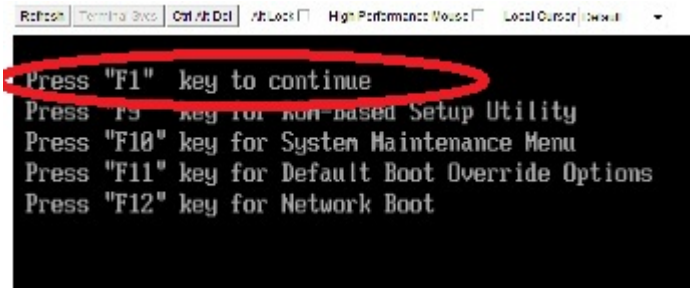
   Note: Discrepancies between the LMS BIOS clock and the time on the NTP server (set later in the install process) could cause problems for password ageing.



7. Press **F10** to continue LMS boot.



8. Press **F1** to continue.



Next Step:

- Proceed to 3 Install LMS

3 Install LMS

Context Description

- This section describes how to install the LITP Management Server (LMS).

Prerequisites

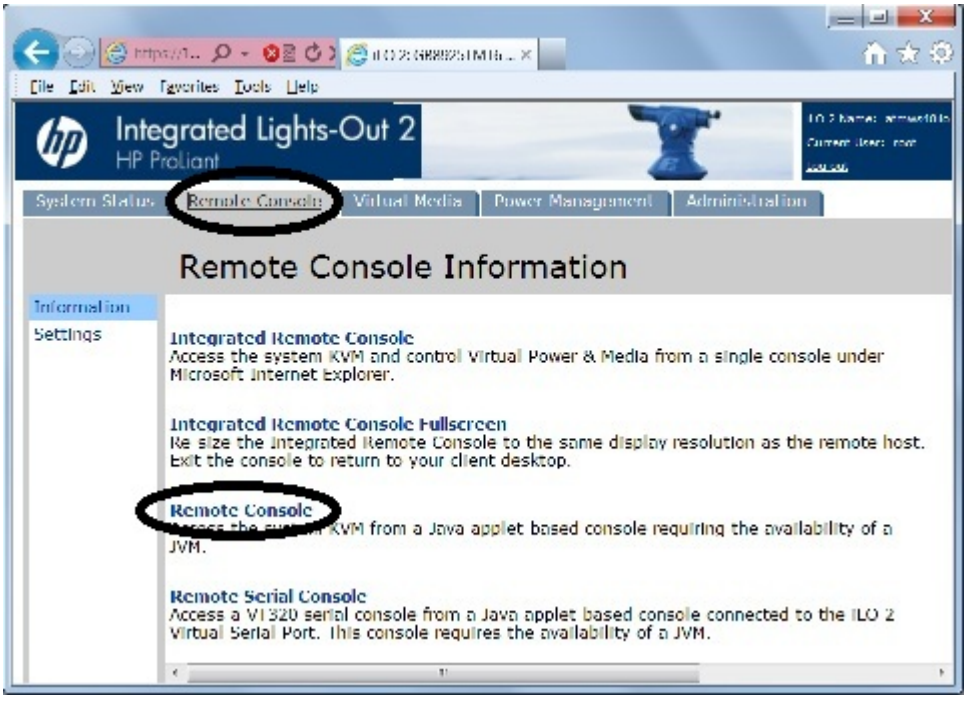
1. Section 2 Boot LMS must be completed.
2. Ensure you have **Site Engineering Data** available. For this task you will need the LMS IP, network mask, default gateway and the IP of at least one name DNS and NTP server.
3. LMS iLO IP address and root password must be known.

Expected Result

- When you have completed these instructions the LMS should be installed and ready for the installer to load the Deployment Description.

Steps

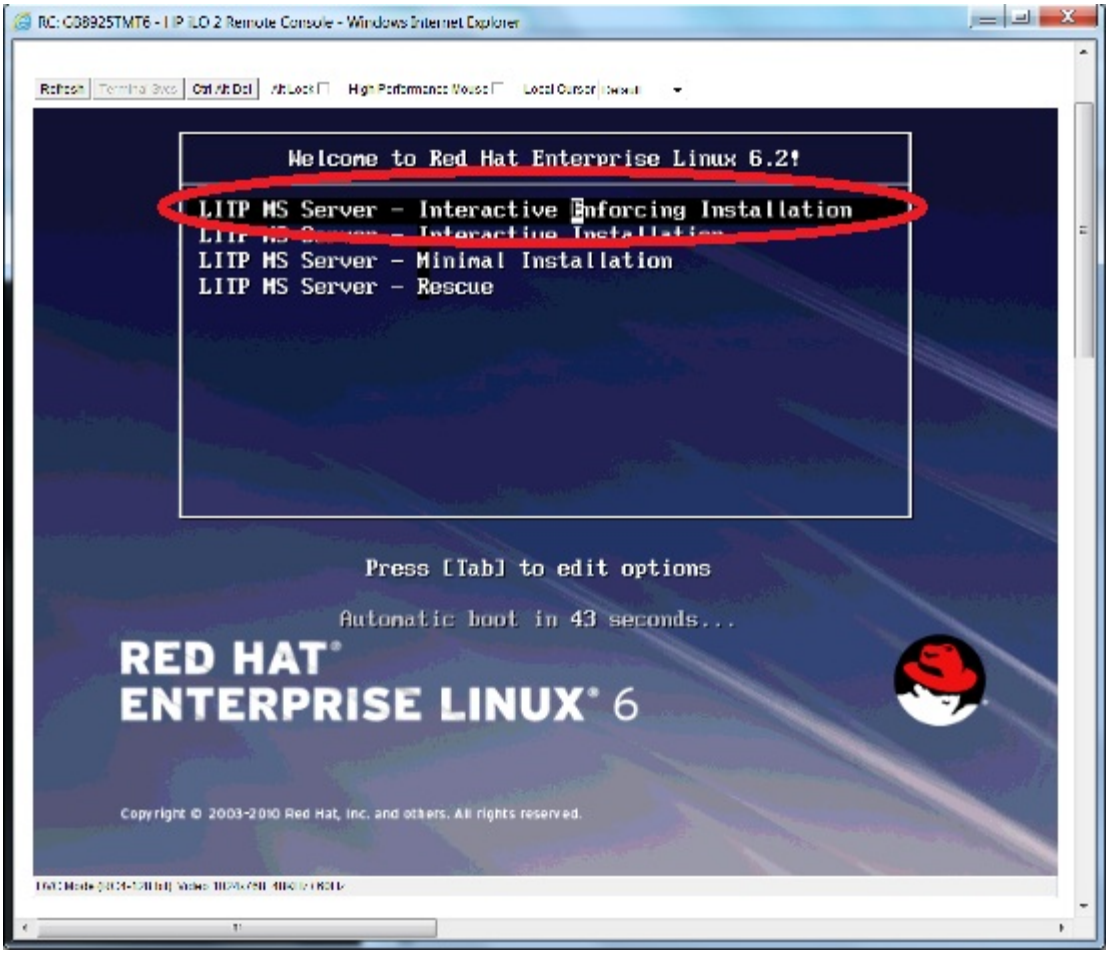
1. Log in to LMS iLO and open a Java applet based remote console connection.



2. While the LMS is booting you will be presented with a number of installation options. Choose **Interactive Enforcing Installation**.

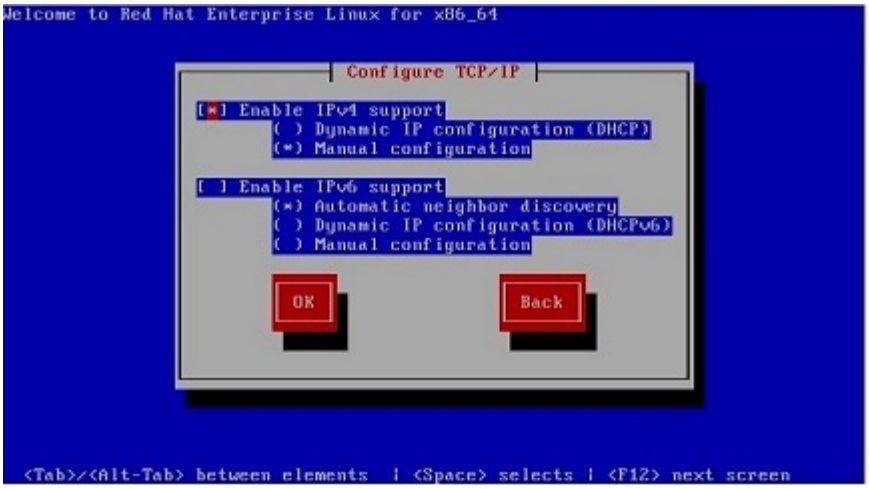
   Interactive Installation allows you to control installation parameters such as disk partitioning type, and install the LMS with SELinux enforcing mode set to enabled. In enforcing mode, policies are enforced for services and users - provided users are mapped to restricted SELinux users.

   Note: If you do not select an installation option, the Interactive Enforcing option is selected by default after approximately one minute.



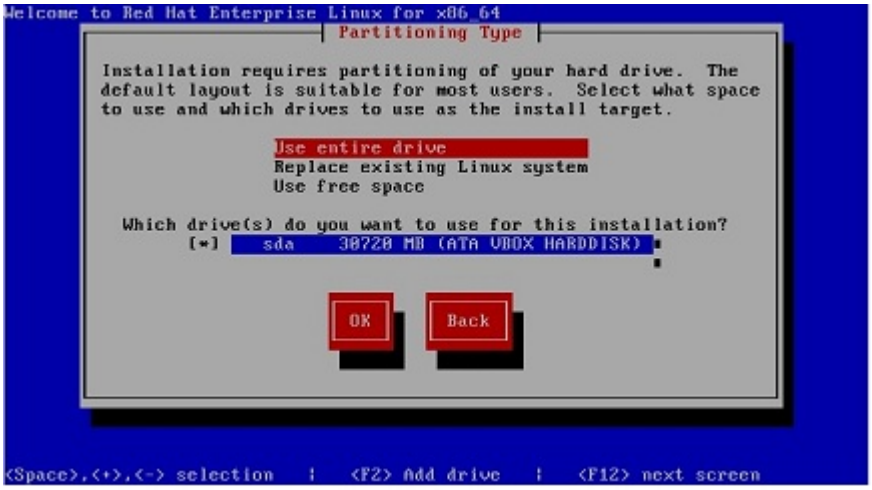
3. Select networking device **eth0** and then **OK**.

4. Select your TCP/IP options by enabling IPv4 Support with Manual configuration and make sure IPv6 is disabled. You must tab to the Enable IPV6 support and press the space bar to make sure that it is unchecked. The screen should appear as below when finished. Select **OK** when done.



5. Enter an IP address, Gateway and Name Server (DNS) for your configuration and select **OK**. Any working DNS server will do, e.g. the local DNS or the OSS-RC infra servers.

6. Select the "Use entire drive" option and then **OK**. The installation will now proceed automatically and will be complete when you can log in to the LMS. This takes about 20 minutes.



7.

8. When the installation is done, log out from the iLO and make a command line login to the LMS main interface with a Tool such as Putty using the following username and password:

   Username: root
   password: passw0rd
   note: 0 above is the number zero

9. Reset the root password.

   You will be prompted to reset the root password. It will first ask you to re-enter the temporary password: passw0rd. Then it will ask you to choose a new password. You will have to reset the password using a "strong" password (use a mix of numbers and characters ad avoid common words). You must then re-enter the new password. When you have logged in, the hostname for the LMS will show as "ms1".

   Note: Password reset times out after 30 seconds.
   If you have not reset the password within 30 seconds the system will log you out. It will then be necessary to log in again using the default username and password as outlined in Step 8

10. Reset the Date and Time

    Check the date and time on the LMS, enter the following command:

    ```
    [root@ms1 ~]# date
    ```

    If the date and time are incorrect, you need to reset the time, and enter the following commands otherwise go to Step 11:

    ```
    [root@ms1 ~]# service puppet stop
    [root@ms1 ~]# service puppetmaster stop
    [root@ms1 ~]# puppetca --clean ms1
    [root@ms1 ~]# rm -fr /var/lib/puppet/ssl
    [root@ms1 ~]# vi /etc/ntp.conf
    ```

    This allows you to update file /etc/ntp.conf. Change the following line specifying the IP address of a working and reachable NTP server, e.g. the customer's own NTP server or the OSS-RC infra servers.

    ```
    server <NTP server IP address>
    ```

    Save the file, then enter the following in order to synchronize the time with the server supplied in /etc/ntp.conf.

```
[root@ms1 ~]# service ntpd stop
[root@ms1 ~]# ntpdate <NTP server IP address>
[root@ms1 ~]# service ntpd start
[root@ms1 ~]# service puppetmaster start
[root@ms1 ~]# service puppet start
```

11. Validate the Deployment Description tree is empty

    After installing the LMS, you will have an empty Deployment Description tree on the LMS. Enter the following command to confirm that the Deployment Description tree is empty:

    ```
    [root@ms1 ~]# litp / validate
    ```

    When the Deployment Description tree is empty, the command will return errors as follows:

    ```
    /bootmgr              ERROR      "Validate failed for /bootmgr"
    /bootmgr       ERROR      "username:  property username has no value for a
    mandatory field"
    /bootmgr       ERROR      "password:  property password has no value for a
    mandatory field"
    /bootmgr       ERROR      "server_url:  property server_url has no value for a
    mandatory field"
    /cfgmgr               ERROR      "Validate failed for /cfgmgr"
    /cfgmgr        ERROR      "sitepp:  LitpItem._validate: property sitepp is not
    supported for this class"
    ```

12. Check the Puppet Cycle

    If you have reset the time on the LMS, enter the following command to check that Puppet cycle is not using the cached catalog with the following command:

    ```
    [root@ms1 ~]# egrep "Finished catalog run|Using cached catalog"
    /var/log/messages
    ```

    If Puppet is not using cached catalogue, you will see the following response:

    ```
    MS1 puppet-master[24774]: (//ms1/Puppet) Finished catalog run in xx.xxx
    seconds
    ```

Note: Puppet will look for additional resources that will not be available at this time and the message above may be accompanied by some unrelated errors. Ignore these.

If Puppet is using a cached catalogue, you will see the following response:

```
MS1 puppet-master[29370]: (//ms1/Puppet) Using cached catalog
```

Hit **Ctrl+Z** to escape the grep program.

If Puppet fails without finishing the catalog run and/or you find Puppet error messages in `/var/log/messages`, enter the following commands:

```
[root@ms1 ~]# service puppet stop
[root@ms1 ~]# service puppetmaster stop
[root@ms1 ~]# puppetca --clean ms1
[root@ms1 ~]# rm -rf /var/lib/puppet/ssl/*
[root@ms1 ~]# service puppetmaster start
[root@ms1 ~]# service puppet start
```

Next Step:

- Proceed to 4 Transfer Deployment Description to LMS

4 Transfer Deployment Description to LMS

Context Description

- The Deployment Description Inventor models specific provisioning details related to - for instance - network interfaces, software components, software applications and their configurations. For each deployment the specific characteristics of these various components will vary according to the real underlying hardware: IP addresses, user names, passwords etc. This instruction describes how to take the Deployment Description definition and inventory provided by the solution architect and place it in a location on the LITP Management Server where it can be loaded in to the deployment model.

Prerequisites

1. Section 3 Install LMS must be finished.
2. The LMS IP address and root password is known.
3. You have been provided with a deployment_description_definition.xml and a deployment_description_inventory.sh by the solution architect

Expected Result:

- The files are ready to be loaded in to the deployment model.

Steps

1. Log in to your Windows computer.

2. Upload the following files to LMS using an SFTP client, e.g. PSFTP or FileZilla. The XML files may be put in /opt/ericsson/nms/litp/bin/samples/DL380/.

   * File **deployment_description_definition.xml**
   * File **deployment_description_inventory.sh**

3. Log in to LMS.

4. Make sure that the files you uploaded exist in /opt/ericsson/nms/litp/bin/samples/DL380/

Next Step:

- Proceed to 5 Load Deployment Description - Definition.

5 Load Deployment Description - Definition

Context Description

- This section describes how to load the Deployment Description Definition.

Prerequisites

1. Section 3 Install LMS must be completed.
2. Section 4 Transfer Deployment Description to LMS must be completed
3. The LMS IP address and root password must be known.

Expected Result

- The deployment description definition has been loaded and the deployment model has been created.

Steps

1. Log on to the LMS.

2. Execute the following in the LMS to make the Deployment Description definition XML file executable.

```
[root@MS1 ~]# chmod +x
/opt/ericsson/nms/litp/bin/samples/DL380/deployment_description_definition.xml
```

3. Load the Definition in to the deployment description model.
   This step loads the components into the deployment model and creates the model inventory which can be updated with site specific information.
   Load the deployment_description_definition.xml you uploaded in the Preparing Deployment Description step by entering the following commands:

```
[root@MS1 ~]# cd /root
[root@MS1 ~]# litp / load
/opt/ericsson/nms/litp/bin/samples/DL380/deployment_description_definition.xml
[root@MS1 ~]# litp /definition materialise
```

   Note: the full path of the file must be loaded in.

4. Check the status of the deployment Description by entering the following command:

```
[root@ms1 ~]# litp / show -rp
```

This command displays a list of the deployment description model components and their status. At this stage, most definition objects should be in status [Initial], with ntp and yum repositories in status [applied].

Next Step

- Proceed to 6 Load Deployment Description - Inventory.

6 Load Deployment Description - Inventory

Context Description

- This section describes how to run the Deployment Description inventory script.

Prerequisites

1. Section 5 Load Deployment Description - Definition must be completed.
2. Section 4 Transfer Deployment Description to LMS must be completed.

Expected Result

- A Deployment Description model is successfully created and verified, storage and sfs share are created and configured.

Steps

1. Run the Deployment Description inventory script as shown below to create a Deployment Description tree under /opt/ericsson/nms/litp/etc/puppet/manifests/inventory/. Use the script you uploaded in the Transfer Deployment Description to LMS step.

   Note: There are over a 120 Automated Steps in the validation and storage configuration of the inventory. This can take up to twenty minutes to verify and configure.

   The Discover and LUN allocation sections of the script takes some time to complete. If you have a second (putty)shell console open on the LMS, you can periodically check /var/log/messages to see when LUNs are created.The Deployment Description inventory script includes a step to validate the entire inventory. If you see validate errors at this stage, run the clean up procedures present in the Troubleshooting Guide.
   Then make necessary corrections in your inventory and re-run this step.

   ```
   [root@MS1 ~]# chmod +x /opt/ericsson/nms/litp/bin/samples/<deployment
   type>/<name of inventory script>.sh
   [root@MS1 ~]# cd /root
   [root@MS1 ~]# sh /opt/ericsson/nms/litp/bin/samples/<deployment type>/<name of
   inventory script>.sh
   ```

2. Check log messages.
   Installation logs are created relative to where you run the inventory and bootmanager scripts from. If you run the scripts from root they will be in the following locations:
   /root/logs/landscape_bootmgr.log
   /root/logs/landscape_inventory.log
   If you see error messages in the log files, run clean_all.sh as outlined in the Troubleshooting Guide and restart the deployment process.

3. Open a new terminal window and enter the following command to view background tasks:

```
[root@MS1 ~]# litp / show_jobs
  "command": "materialise",
  "result": null,
  "vpath": "/definition",
  "jobid": "043d5686-c5e3-11e1-8876-d067e503190a"
```

4. Check that puppet catalog run has finished successfully and that the compiled catalog is in use by entering the following command. Check for messages indicating that puppet is finished its catalog.

```
[root@MS1 ~]# tail -f -n 0 /var/log/messages | grep "Finished catalog run"

Jun  4 18:06:58 ms1 puppet-agent[2597]: Finished catalog run in 122.11 seconds
Jun  4 18:06:58 ms1 puppet-master[2420]: (//ms1/Puppet) Finished catalog run
in 122.11 seconds
```

5. Check that cobbler authentication is set up.

   Note: The text "module=authn_testing" in the response indicates that authentication is set up.

```
[root@MS1 ~]# grep "authn_testing" /etc/cobbler/modules.conf
#    authn_testing    -- username/password is always testing/testing (debug)
module = authn_testing
```

6. Check the status of the Deployment Description by entering the following command:

```
[root@MS1 ~]# litp / show -rp
```

   Note: The Deployment Description component status is displayed using colour codes. At this stage:
   - Deployment Description Definition items remain in status [Initial].
   - When puppet generates config files for objects they are set to status [Applied].
   - Resource pools are in status [Available].
   - Objects assigned to pools are in status [Allocated].

Next Step

- Proceed to 7 Load Boot Manager and Verify Installation

7 Load Boot Manager and Verify Installation

Context Description

- This section outlines how to run the boot manager and verify the installation.

Prerequisites

1. Section 5 Load Deployment Description - Definition must be completed.
2. Section 6 Load Deployment Description - Inventory must be completed.

Expected Result

- LMS and peer blades are installed and verified.

Steps

1. Using the provided boot manager script run the kickstart of the nodes**.**
   The file is located in the /opt/ericsson/nms/litp/bin/samples/DL380/ directory. To run the boot manager enter the following command:

   ```
   [root@MS1 ~]# cd /root
   [root@MS1 ~]# sh
   /opt/ericsson/nms/litp/bin/samples/DL380/dl380_multi_blade_boot_mgr.sh
   ```

   When you enter this command, the cobbler installation and update server causes the peer blades to PXE boot, install and configure both RHEL and LITP. Private and shared storage are mounted to each peer blade and then restart, finally CMW is deployed.
   Following a successful kickstart and reboot of a managed node, the Puppet agent connects to the Puppet master on the MS and applies the Puppet catalogue.

2. Your deployment includes an SFS server. Check if the shares exist. ssh to SC-1 and enter the following command to check that shares are created:

   ```
   [root@MS1 ~]# ssh sc-1
   ...
   [root @SC-1 ~]# showmount -e <SFS Server IP Address>
   ```

   If this is successful you will see a long list of directories created after the /cluster mount point.

3. Run the command "cobbler sync". If no errors are output, then you can proceed to the next step.

4. Check that Cobbler has imported the distribution and profile, enter the following command.

   ```
   [root@ms1 ~]# cd /opt/ericsson/nms/litp/bin/samples/DL380/
   [root@ms1 ~]# cobbler list
   ```

   This should return a single MS with 2 blades for this configuration.

5. After running boot manager, check the status of your deployment at five minute intervals over a half-hour period by entering the following command:

```
[root@ms1 ~]# litp /inventory show -rp
```

Each component's status is displayed using colour codes;

- Deployment Description Definition items remain in status [Initial].
- Over a period of approximately 30 minutes, objects such as nodes, tools, users and CMW campaigns move to from status [Applying] to status [Applied].
- There is currently a bug on the sfs components where they succeed but their state never gets changed from status [Applying] to status [Applied].

6. Observe deployment on managed nodes.
Deployment time varies depending on the size of your configuration. During this time, you can open the iLO f or each node to observe progressof the deployment.

7. Increase the Number of Processes on Peer Nodes.
Currently, the number of processes on peer nodes is set at 1024. You need to increase this to 4096 per node, take the following steps on each peer node:

```
vi /users/litp_jboss/.bash_profile

paste in this content and save:
ulimit -u 4096
ulimit -n 65535
```

```
vi /etc/security/limits.conf

paste in this content and save:
litp_jboss          soft    nproc           2047
litp_jboss          hard    nproc           16384
litp_jboss          soft    nofile          1024
litp_jboss          hard    nofile          65536
```

8. Check that deployment is complete.
Deployment is complete when all the Core MW Campaigns are in status COMMITTED on each managed node. You can check this by entering the following commands on each managed node:

```
[root@MS1 ~]# ssh sc-1

Last login: Fri Apr 13 16:01:28 2012 from ms1

[root@SC-1 ~]# cmw-repository-list --campaign | xargs cmw-campaign-status

ERIC-instcmpg_cmwbackport4-CXP123456_1-R1A01=COMMITTED
```

Note: It will take 10 to 15 minutes for CMW to be deployed before the preceding command can run successfully.

9. Execute Campaigns
   You must now execute campaigns,by entering the following command:

```
litp /inventory/deployment1/cluster1/cmw_cluster_config/campaign_generator
execute
```

10. Re - check the Deployment
    Enter the following commands again on each peer server:

```
[root @MS1 ~]# ssh sc- 1

Last login: Fri Apr  13 16 : 01 : 28 2012 from ms1

[root @SC - 1 ~]# cmw-repository-list --campaign | xargs cmw-campaign-status

ERIC-instcmpg_cmwbackport4-CXP123456_1-R1A01=COMMITTED
```

11. Change admin password
    After deployment, log onto the MS using the following default login details:
    username:litp_admin
    password:passw0rd
    Once you have logged on as user litp_admin, change the password to a 'strong' UNIX password. That is a mixed-case password with non-alphanumeric characters using the passwd command as follows:

```
[litp_admin@ms1 ~]# passwd
```

    You will then be prompted to enter and confirm a new password.

12. Verify Deployed Components
    Use the Verify command to verify the deployed system against the Deployment Description by entering the following command:

```
[root@ms1 ~]# litp /inventory verify
```

13. Check Puppet Configuration
    Check that Puppet is running by entering the following command:

```
[root@MS1 ~]# service puppet status

[root@MS1 ~]# puppetd (pid  29106) is running...
```

14. Congratulations you have successfully completed the installation.

8 Troubleshooting

8.1 Apply clean up scripts

Context Description

This section applies If you encounter problems or errors and you need to restart the deployment process. Do **NOT** r un the procedures in this section if your deployment boots successfully.

The LITP ISO contains the following cleanup files:

- clean_landscape.sh
- clean_all.sh

They are located in /opt/ericsson/nms/litp/bin/samples/clean_up

Prerequisites

You have entered a definition and/or inventory that you would like to remove from the deployment tree.

Result

The deployment tree is cleaned up.

Steps

Run either of the following commands depending on your needs. You don't have to run both.

- clean_landscape.sh
  Run `clean_landscape.sh` if your definition or inventory contains erroneous CLI commands. `clean_landscape.sh` cleans up the inventory and definition and restarts the service. Enter the following command:

  ```
  [root@MS1 ~]# /opt/ericsson/nms/litp/bin/samples/clean_up/clean_landscape.sh
  ```

- clean_all.sh
  Run `clean_all.sh` if you wish to remove the entire LITP configuration including Deployment Description, Cobbler, puppet etc. Enter the following command:

  ```
  [root@MS1 ~]# sh /opt/ericsson/nms/litp/bin/samples/clean_up/clean_all.sh
  ```

Post Requirement

Proceed to 5 Load Deployment Description - Definition.

8.2 Boot Manager script error

Context Description

If your inventory includes an SFS pool, but the SFS pool does not exist in the SFS server, you will not see any error messages when the inventory is run. Instead you will see an error when you run the boot manager script in section 7 Load Boot Manager and Verify Installation

Prerequisites

1. The Boot Manager failed displaying an error.
2. No SFS pool for TOR exists in the SFS server. To verify, log in to the SFS console as user master and run command `storage disk list detail`. No SFS pool for TOR can be seen in the output.

Result

After having completed this task the Boot Manager script error is cleared.

Steps

1. Create the SFS pools as outlined in chapter **7 VNX Configuration for TOR Deployment** of document **EMC VNX Configuration for OSS-RC ENIQ and TOR**.

Post Requirement

8.1 Apply clean up scripts and proceed to 5 Load Deployment Description - Definition.

8.3 Logging and monitoring

Abstract

LITP has a number of methods of logging. Details on each method are outlined here.

Logging and monitoring

Logging summary

Logging type	Server	Log files location	Log rotation	Centralised
LITP	LMS and blades	`/var/log/litp.log` (Error and Info messages) `/var/log/litp/litp-debug.log`	Yes	Yes
Deployment Description	LMS	`/var/log/litp/litp_service.log`	Yes	No
Core MiddleWare	Blades	`/cluster/storage/no-backup/coremw/var/log/saflog`	Yes	No
Puppet	LMS and blades	`/var/log/messages`	Yes	Yes
System	LMS and blades	`/var/log/messages`	Yes	Yes
SELinux	LMS (and blades ?)	`/var/log/audit/audit.log`	Yes	?

LITP logging

The log files that are used are the following:

- `/var/log/litp.log`
- `/var/log/litp/litp-debug.log`

Hardware and storage logging is written to these files. They are under the control of the syslog daemon. They are centralised and also rotated.

Deployment Description logging

Deployment Description log is where the rest of the LITP components write to. The log file is stored here:

- `/var/log/litp/litp_service.log`

Core MiddleWare logging

Applications using Core MiddleWare write to specific logs. These log files are available on the blades only at path:

- `/cluster/storage/no-backup/coremw/var/log/saflog`

Puppet logging

As mentioned above, Puppet writes to the `/var/log/messages` log. Puppet also writes to some specific Puppet reports that can be viewed from the Puppet Dashboard. See section **LITP Admin Portal** below on how to access Puppet Dashboard. These report files are stored in `/var/lib/puppet/reports/.` Note that report files are deleted after four days by root crontab to preserve disk space.

System logging

The messages logfile is a key source of logging information on the LITP platform and should be used as one the first places you look for errors. Processes such as Puppet, Cobbler, OS write to the messages log file. As Puppet is a key process on the LITP Platform it is important you ensure it is behaving correctly. It is recommended you check /var/log/messages for any Puppet related errors. The messages log file is stored here:

- `/var/log/messages`

SELinux logging

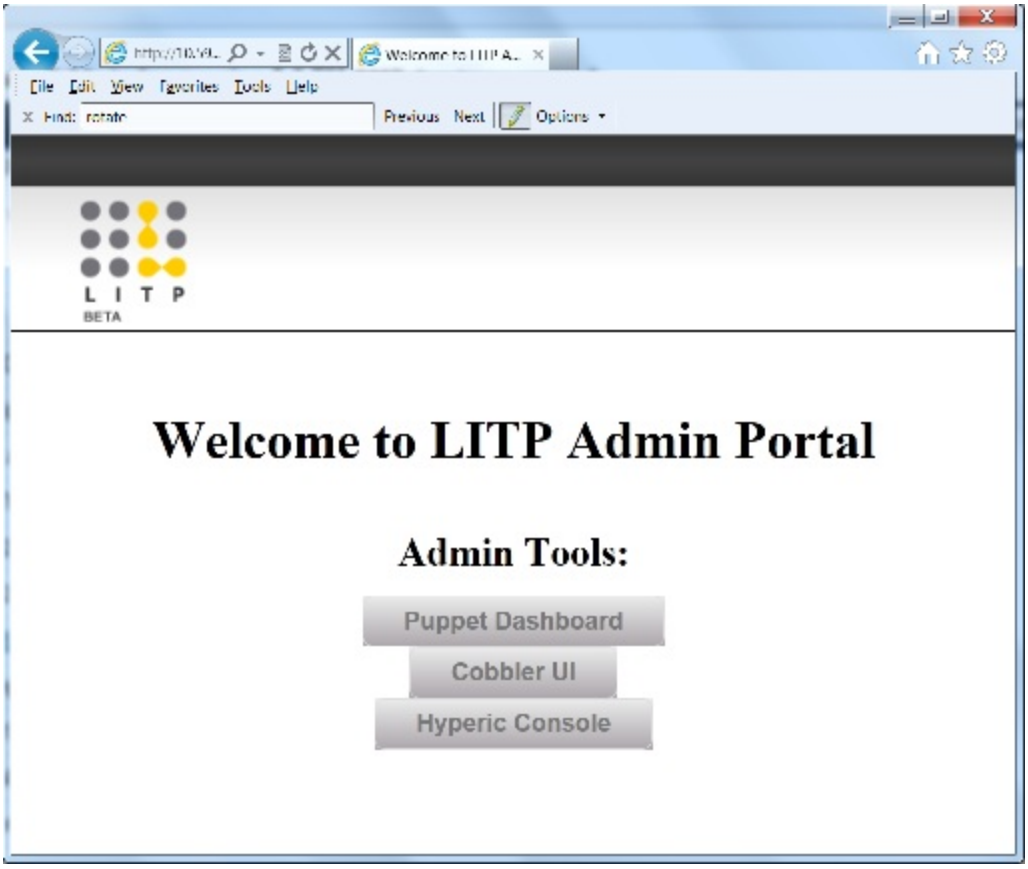
The audit log file is a log file used to store SELinux success and denied messages. The audit log file is stored here:

- `/var/log/audit/audit.log`

It is recommended you monitor this file for SELinux violations and report on them accordingly.

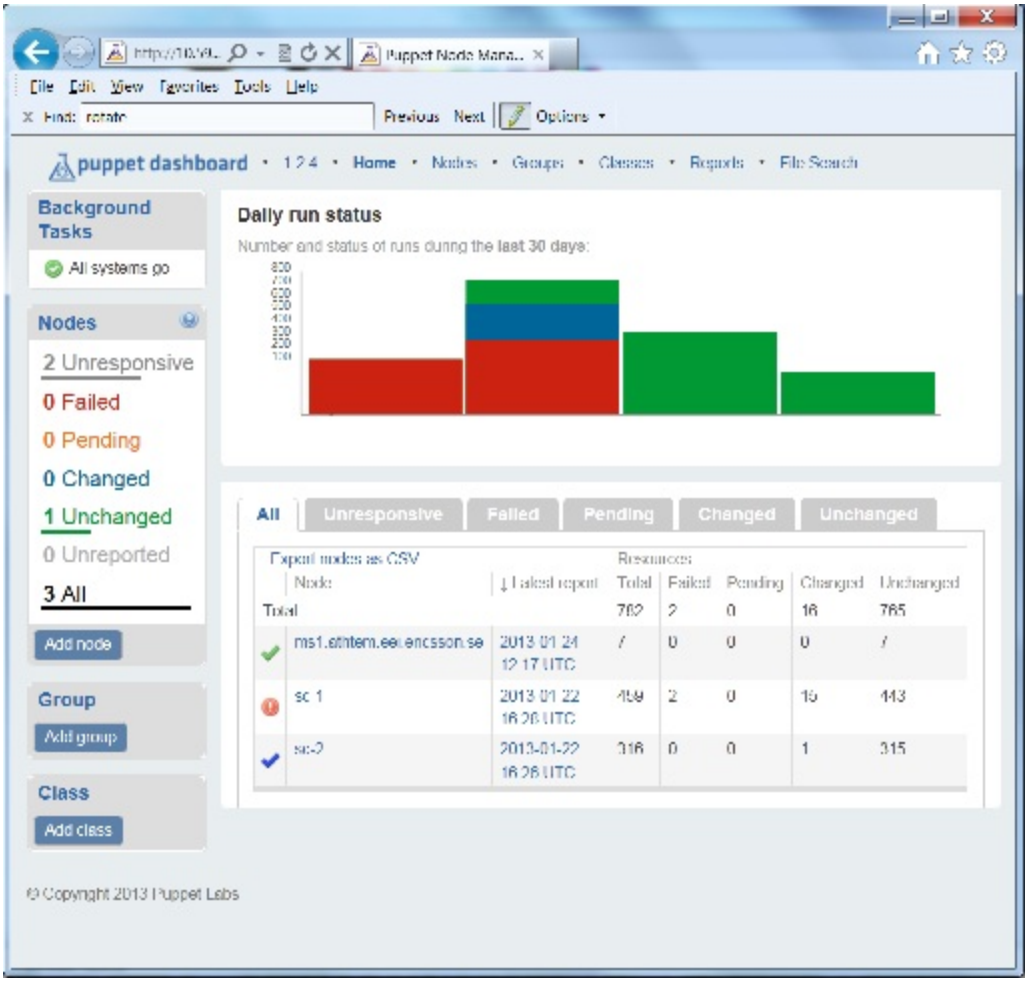
LITP Admin Portal

The Admin Portal on LITP provides access to Puppet, Cobbler and Hyperic. Enter the LMS IP address on your browser to launch it the portal.



Puppet Dashboard

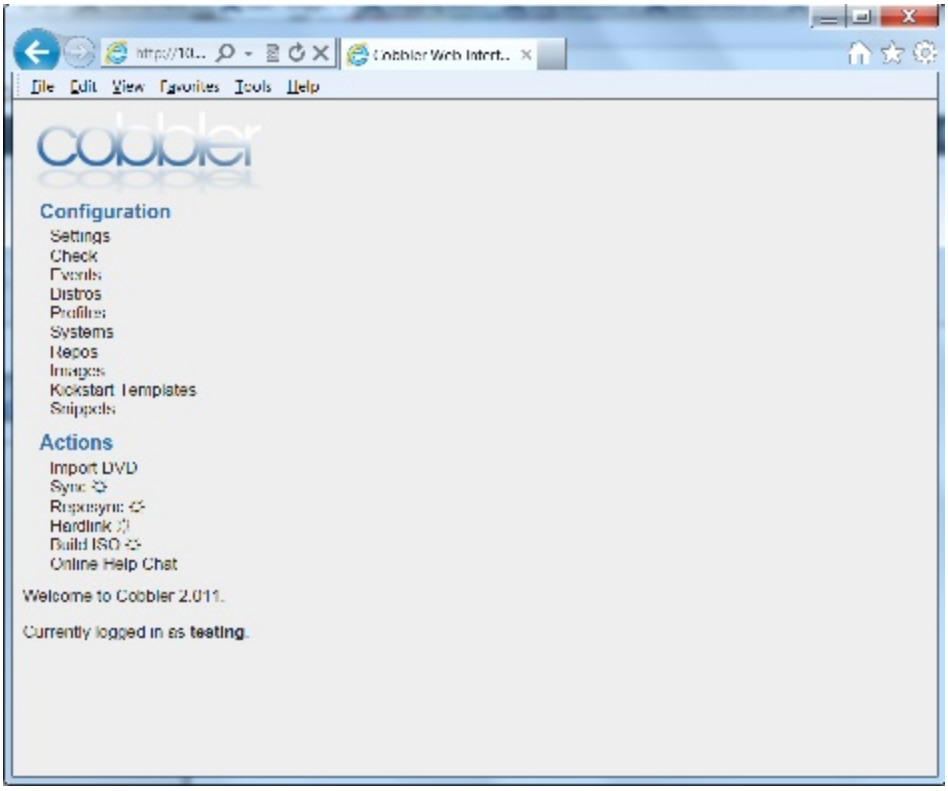
Click button **Puppet Dashboard** to launch the Puppet Dashboard shown below. Here you can view the daily logs and reports of LMS and each blade.



Cobbler UI

Click button **Cobbler UI** on the LITP Admin Portal page to log in to Cobbler. You will be prompted for a user name and password. The default user name as well as password is "testing".

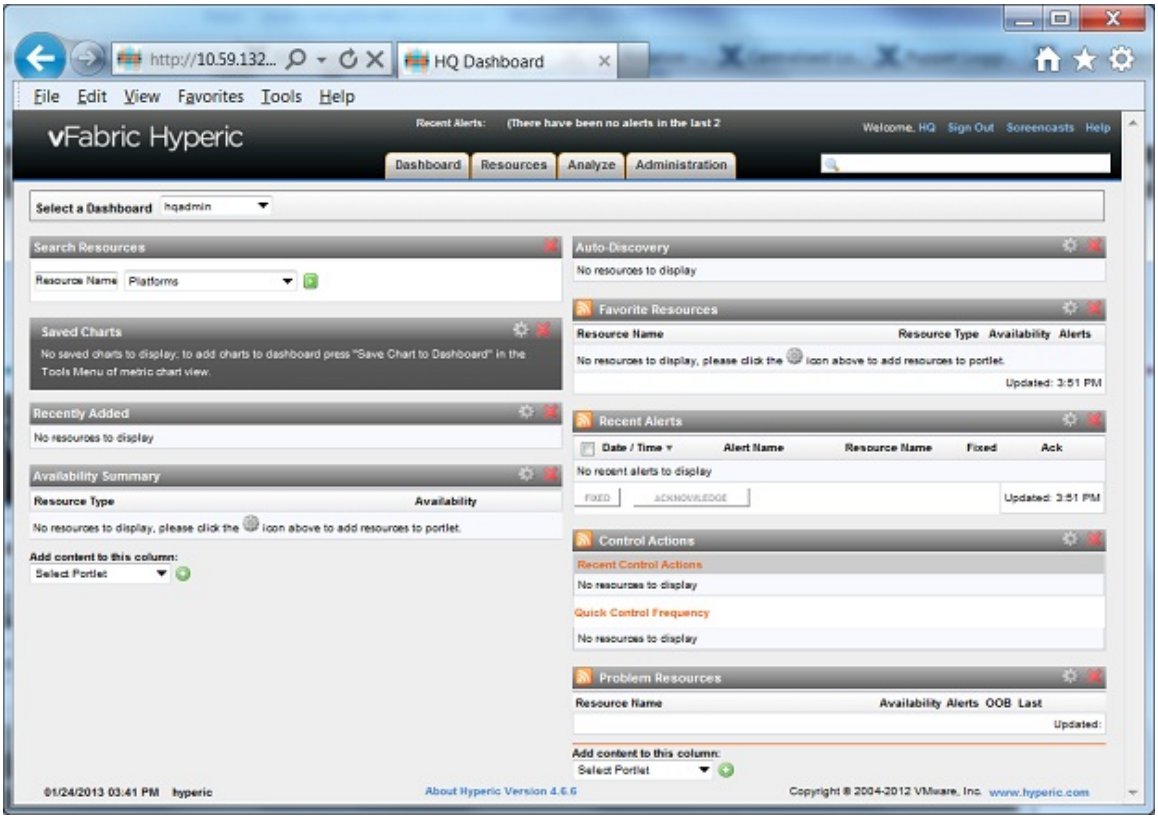
The Cobbler User Interface (UI) provides information on the installation process, such as the kick-starts on each blade. If any events are logged you can see them here. You can check Cobbler settings and also check the Cobbler configuration.



Hyperic Console

Click button **Hyperic Console** on the LITP Admin Portal page to log in to Hyperic. You will be prompted for a user name and password. The default user name as well as password is "hqadmin". After login you see the HypericDashboard, as displayed below.

Hyperic Central Monitoring Service (aka Hyperic Server) is deployed on the LMS and Hyperic Agents are deployed on blades. Each Hyperic Agent communicates with the Hyperic Central Monitoring Service on LMS. Hyperic monitors the platform server enclosures, servers, JBOSS OSS applications/components and JVM's as well as 3PP products such as CMW and VCS.



Related Links

-

8.4 Tear down storage - TOR Boot and TOR App

Context Description

TOR storage may need to be re-created for various reasons. This task tells how to remove the TOR Boot and TOR App LUNs and corresponding RAID groups.

Prerequisites

1. The TOR Boot and TOR App LUNs are configured either in full or partly.
2. TOR blade iLO IPs and root passwords are known.
3. TOR blade nic IPs and root passwords are known.
4. VNX SP IP address and root password is known.

Result

After having completed this task the entire TOR Boot and TOR App LUNs and their corresponding RAID groups are gone.

Steps

1. Log in to the CLI of SFS as user master.

2. Run the following command and identify the name of the export file system in the TOR SFS storage pool. For the name of the TOR SFS storage pool, see label **export_storadm_storage_pool** in **Site Engineering Data** . Here you find also the name of the export file system. Look for label **export1_path** and strip off the leading /vx/ to get the name of the file system.

   > storage fs list

3. Run the following command to see if this file system is shared or not. If it is, it should appear twice in the output (once for each blade), otherwise not.

   > nfs share show

4. If the export file system is shared, run the following two commands to delete the shares.

   > nfs share delete <export1_path> <blade1 IP>
   > nfs share delete <export1_path> <blade2 IP>

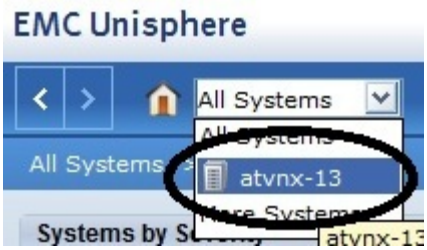
5. Now delete the export file system itself by running the following command.

   > storage fs destroy <export1_path without the leading /vx/>

6. Log out from SFS.

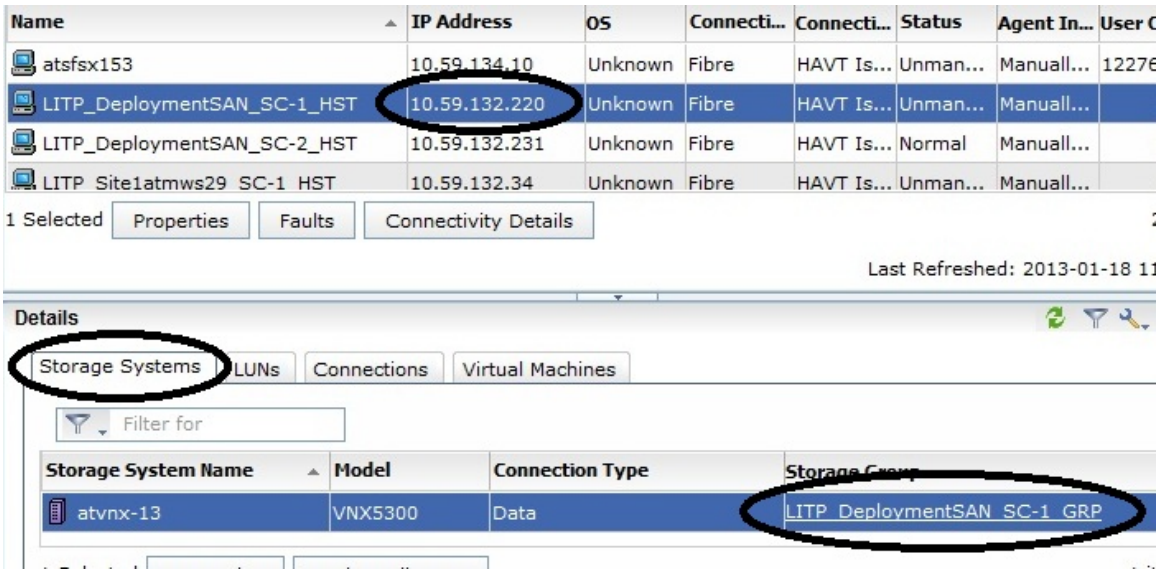
7. Log in to the EMC Unisphere GUI of VNX.

8. Select VNX.

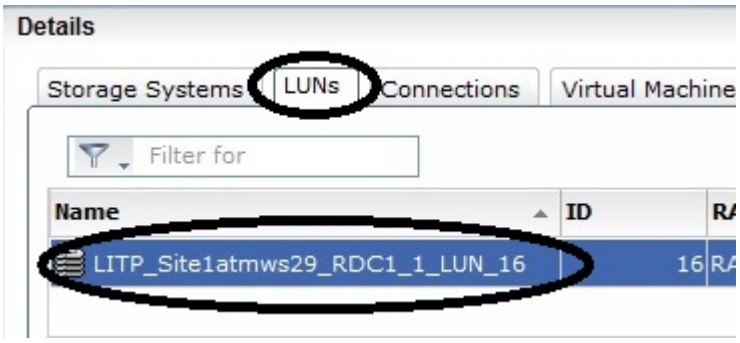


9. Select **Hosts -> Host List**.

10. Identify the name of the first blade based on its IP address. Select the **Storage Systems** tab and make a note about its storage group name.



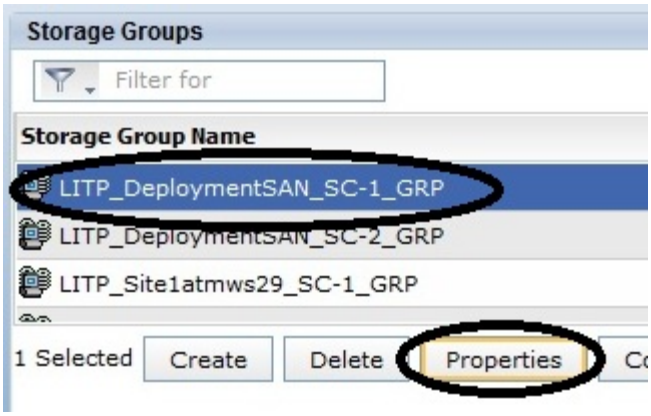
11. Select the LUNs tab and make a note about the LUN names. There should be two LUNs and not one as shown in the screenshot below. One LUN is for TOR Boot and the second is for TOR App.



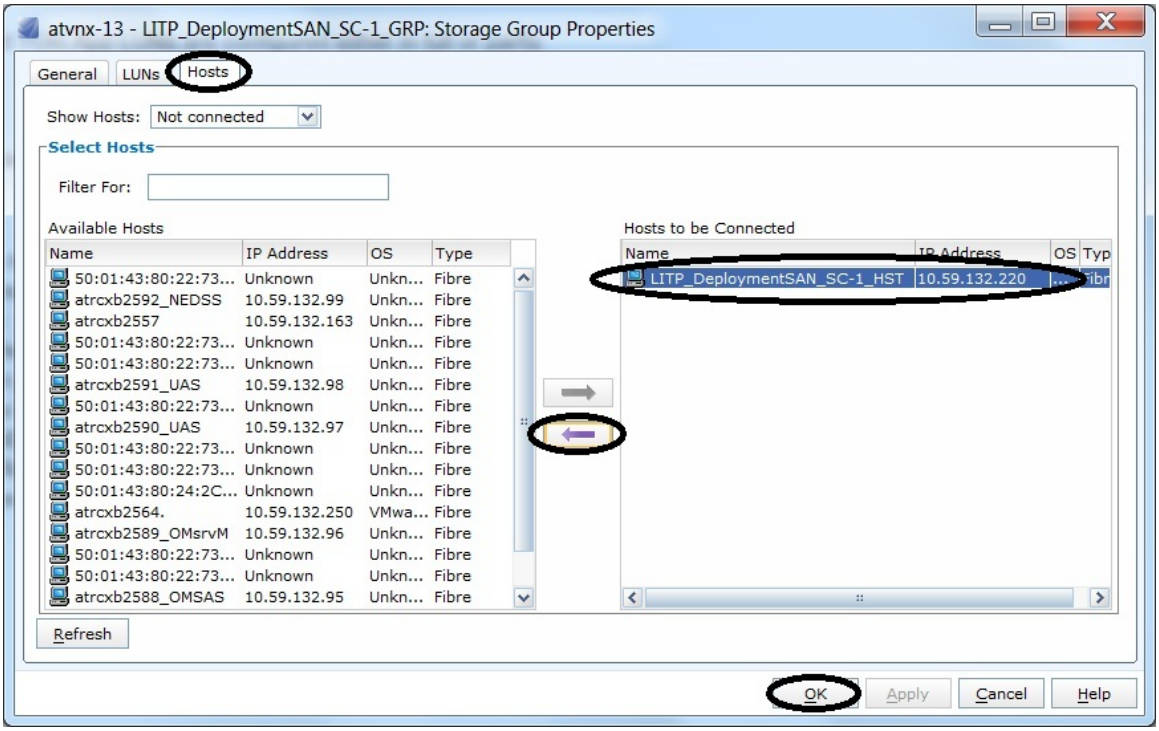
12. Repeat steps 10 and 11 for the second blade.

13. Select **Hosts -> Storage Groups**.

14. Select the storage group of the first blade and click **Properties**.

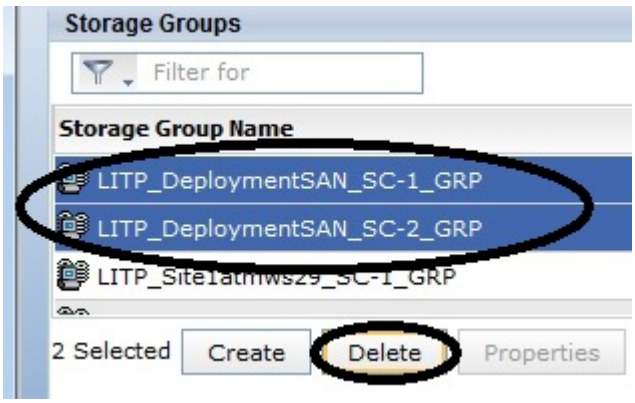


15. Select the **Hosts** tab.and select the blade in the right list. Click the left arrow button and **OK**.



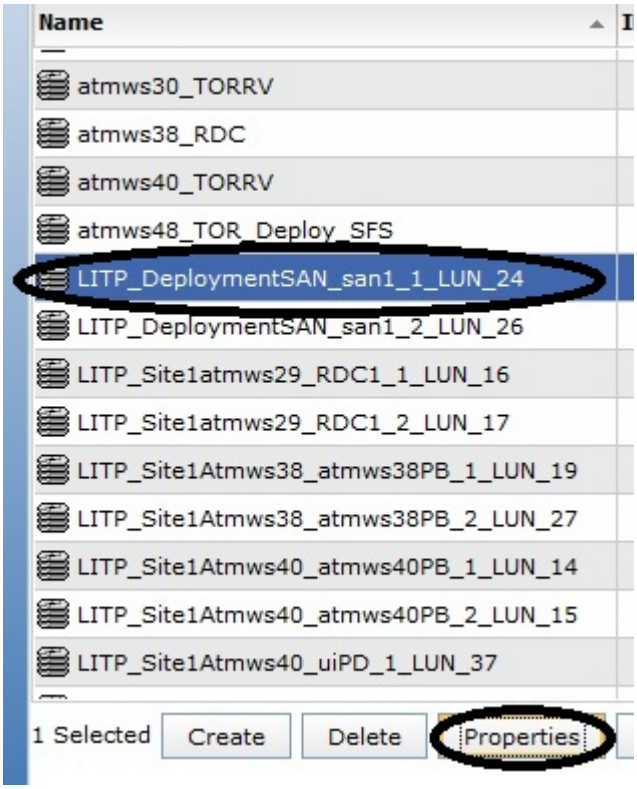
16. Repeat steps 14 and 15 for the second blade.

17. Select the blade storage groups and click **Delete**.

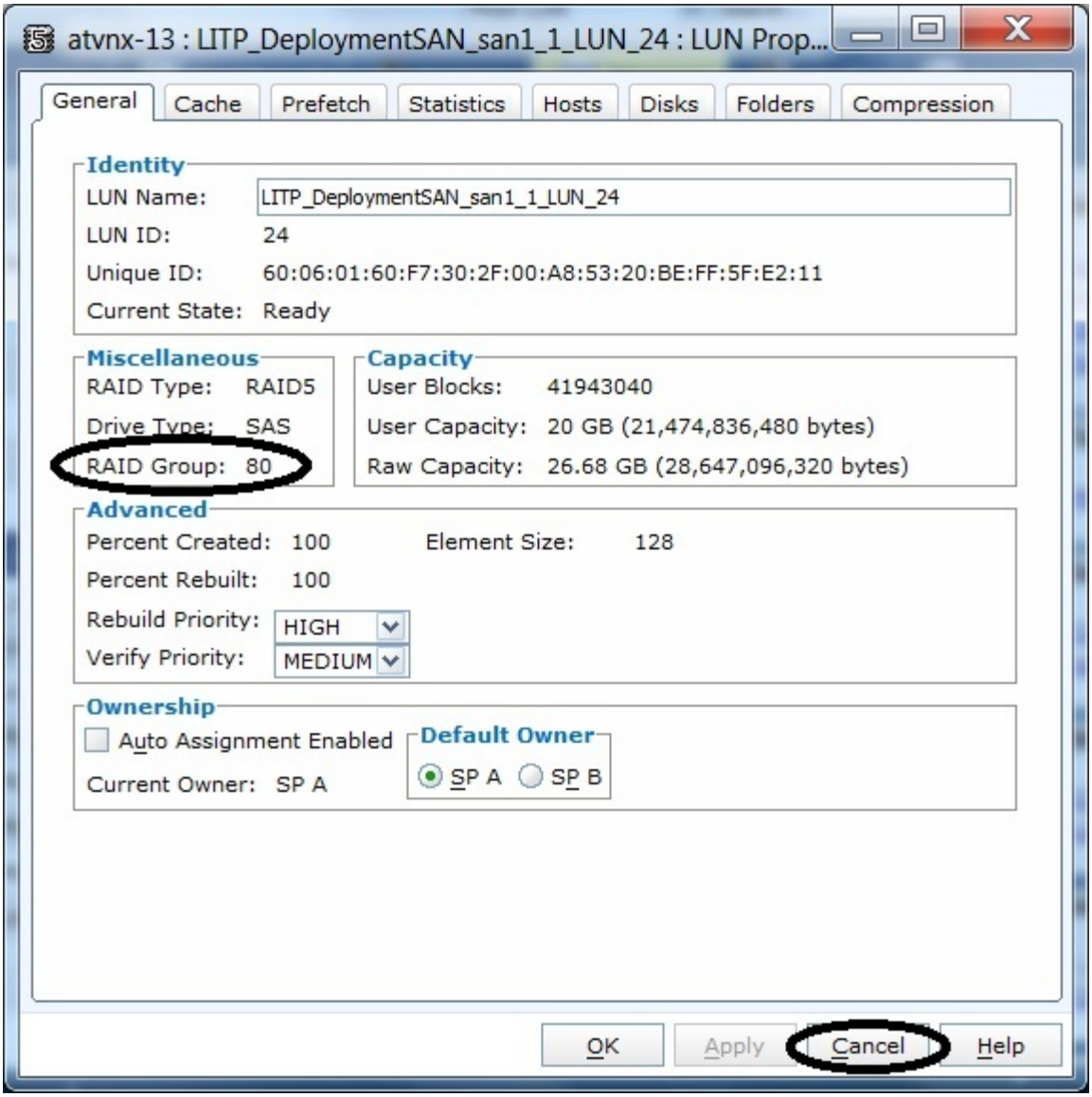


18. Select **Storage -> LUNs**.

19. Select the TOR Boot LUN of the first blade and click **Properties**.



20. Make a note about the RAID group ID and click **Cancel**.

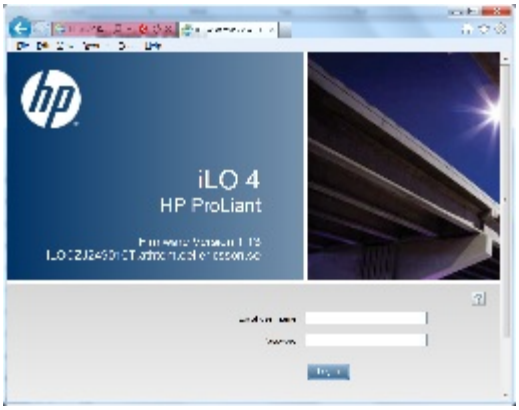


21. Repeat steps 19 and 20 for the TOR App LUN of the first blade.

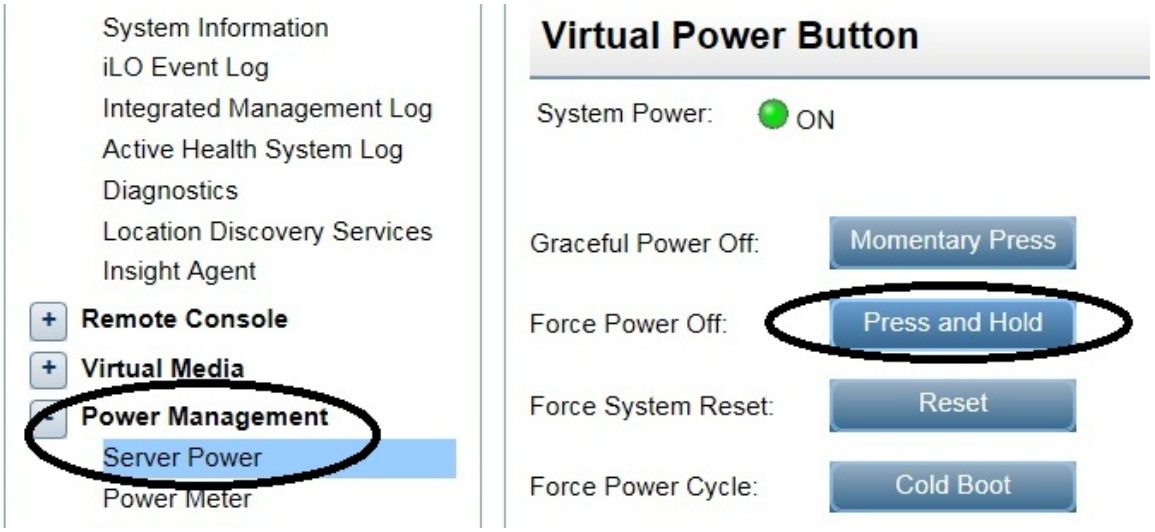
22. Select the TOR Boot LUN of the first blade and click **Delete**. Repeat this for the TOR App LUN of the first blade and then also for the TOR Boot LUN and TOR App LUN of the second blade.

    Note: All four LUNs can actually be co-selected and deleted in one go instead of deleting the LUNs one by one.

23. Log in to the iLO of the first blade.

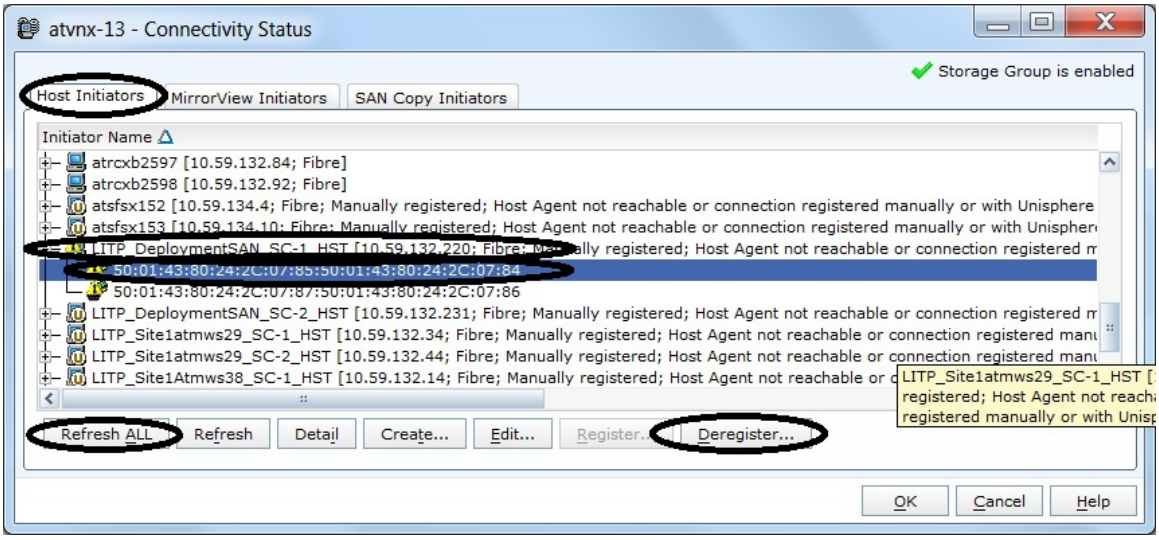


24. Select **Power Management -> Server Power**. Click **Force Power Off** to shut down the blade. Wait until the system power is off.



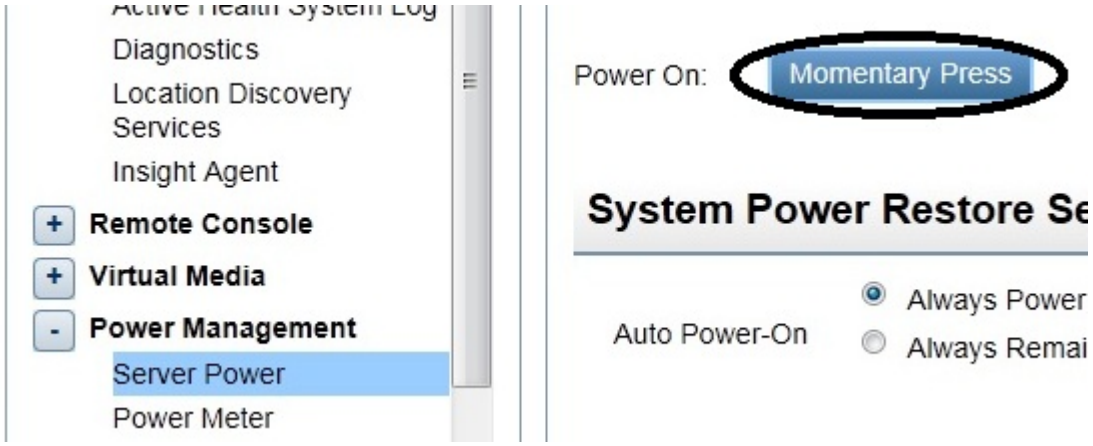
25. Switch back to the EMC Unisphere GUI and select **Hosts -> Connectivity Status**.

26. Select the **Host Initiators** tab. Locate the first blade and expand its entry. Select its first HBA UID and click **D eregister**. Click **Refresh All** to update the window.



27. Repeat step 26 for the second HBA UID of the first blade and then click **OK**.

28. Switch to the iLO of the first blade. Power on the blade by clicking **Power On**.



29. Repeat steps 23 to 28 for the second blade.

30. Switch to the EMC Unisphere GUI and select **Storage -> Storage Pools**.

31. Select the **RAID Groups** tab and select the TOR Boot and TOR App RAID group IDs identified earlier and click **Delete**.

Post Requirement

If your intention is to rebuild the LUNs and RAID groups for TOR Boot and TOR App only, i.e. **not** reinstall LMS and rebuild the TOR SFS and Hot Spare LUNs and RAID groups, then perform the following steps:

1. Create the TOR Boot and TOR App RAID groups as outlined in section **7.3.1 Create TOR RAID Groups** of **EMC VNX Configuration for OSS-RC, ENIQ and TOR**.
2. Run clean_all.sh in LMS as described in section 8.1 Apply clean up scripts of 8 Troubleshooting.
3. Restart blade installation as described in section 5 Load Deployment Description - Definition after having taken care of the possible problems that made you reinstall the blades.

If you intend to reinstall TOR from scratch, i.e. all TOR storage, LMS and blades, proceed to 8.5 Tear down storage - TOR Hot Spare.

8.5 Tear down storage - TOR Hot Spare

Context Description

TOR storage may need to be re-created for various reasons. This task tells how to remove the TOR Hot Spare LUN and RAID group.

Prerequisites

1. The TOR Hot Spare LUN is configured either completely or partly.
2. VNX SP IP address and root password is known.

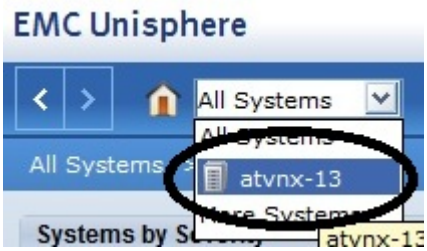
Result

After having completed this task the entire TOR Hot Spare LUN and its RAID group is gone.

Steps

1. Log in to the EMC Unisphere GUI of VNX.

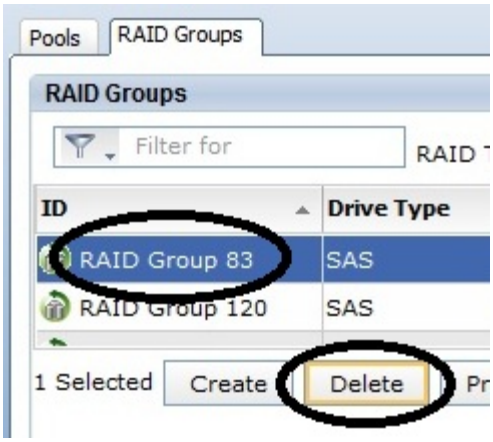
2. Select the VNX for TOR. The VNX for your TOR is specified in **Site Engineering Data**.



3. Select **Storage -> Storage Pools** on the EMC Unisphere main menu.

4. Select the **RAID Groups** tab and locate the TOR Hot Spare RAID group. The TOR Hot Spare RAID group ID is specified in **Site Engineering Data**. Select the TOR Hot Spare RAID group and locate the TOR Hot Spare LUN. Select the TOR Hot Spare LUN and click **Delete**.

5. Finally delete the TOR Hot Spare RAID group by making sure the Hot Spare RAID group is still selected and then click **Delete**.



Post Requirement

Proceed to 8.6 Tear down storage - TOR SFS.

8.6 Tear down storage - TOR SFS

Context Description

TOR storage may need to be re-created for various reasons. This task tells how to remove the TOR SFS LUN and RAID group.

Prerequisites

1. Section 8.4 Tear down storage - TOR Boot and TOR App must be completed.
2. The TOR SFS LUN is configured either completely or partly.
3. SFS console IP and master password is known.
4. VNX SP IP address and root password is known.

Result

After having completed this task the entire TOR SFS LUN and its RAID group is gone.

Steps

1. Log in to the CLI of SFS as user master.

2. Run the following command and identify the name of the two SFS file systems in the TOR SFS storage pool. For the name of the TOR SFS storage pool, see label **export_storadm_storage_pool** in **Site Engineering Data**. Here you find also the name of the file systems. Look for labels **export_storadm_path** a nd **export_storobs_path** and strip off the leading /vx/ to get the name of the file systems.

   > storage fs list

3. Run the following command to see if these file systems are shared or not. If they are they appear in the output, otherwise not.

   > nfs share show

4. If these file systems are shared, run the following two commands to delete the shares.

   > nfs share delete <export_storadm_path> <LMS IP>
   > nfs share delete <export_storobs_path> <LMS IP>

5. Now delete the file systems themselves by running the following two commands.

   > storage fs destroy <export_storadm_path without the leading /vx/>
   > storage fs destroy <export_storobs_path without the leading /vx/>

6. Run the following command to list all storage pools and identify the TOR SFS storage pool. In this list you should find the name of the TOR SFS storage pool referred to in step 2.

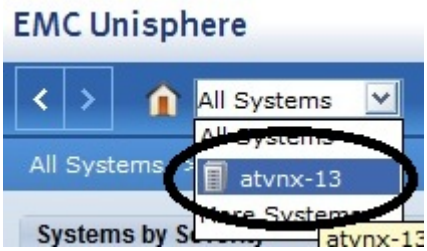
   > storage disk list detail

7. Destroy the TOR SFS storage pool by running the following command.

   > storage pool destroy <name of the TOR SFS storage pool>

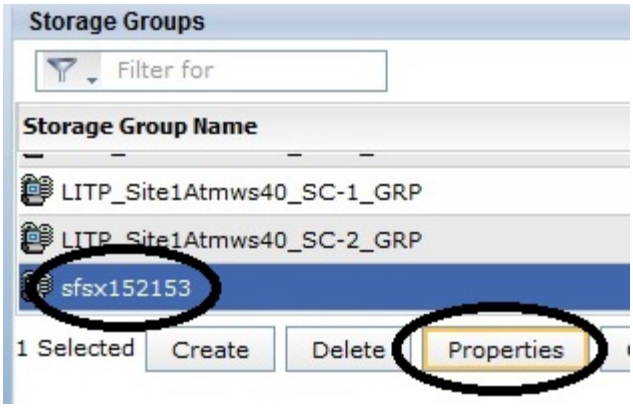
8. Log out from SFS.
9. Log in to the EMC Unisphere GUI of VNX.

10. Select the TOR VNX system. The VNX system name for your TOR is specified in **Site Engineering Data**.

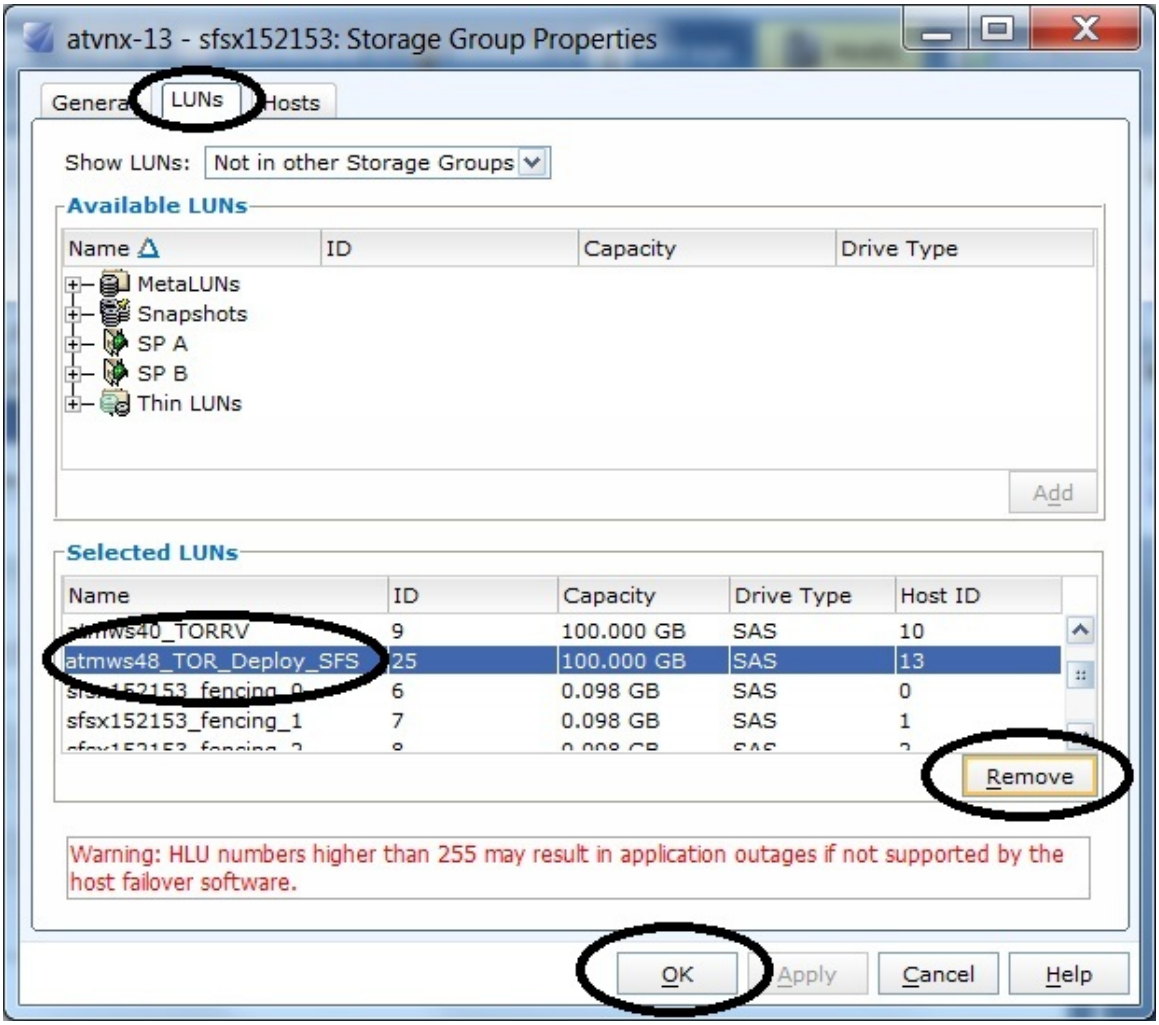


11. Select **Hosts -> Storage Groups**.

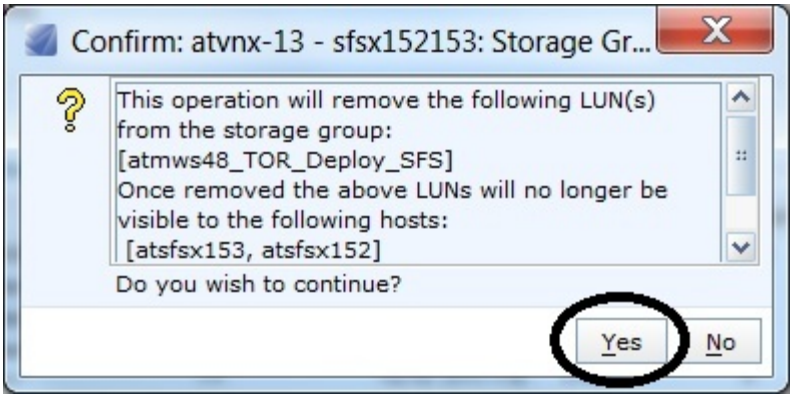
12. Locate and select the OSS-RC storage group. Click on the **Properties** button.



13. In the **Storage Group Properties** window that appears, select the **LUNs** tab and locate the TOR SFS LUN. Select the LUN and click on the **Remove** button followed by **OK**.

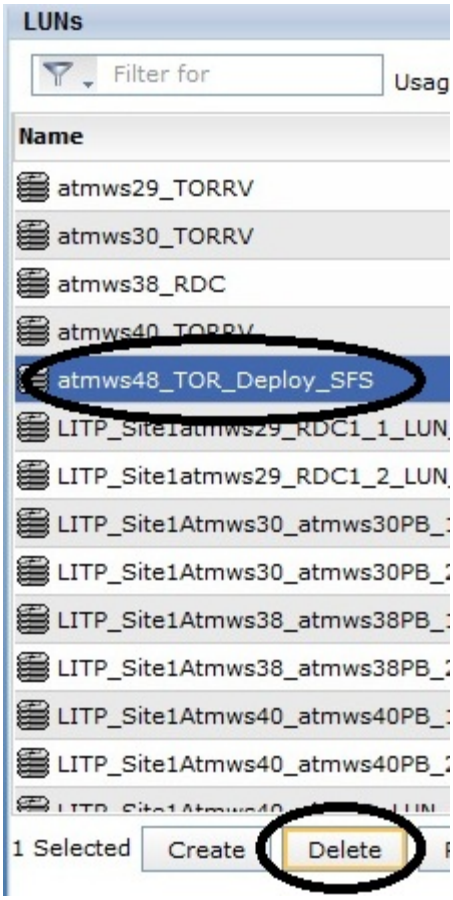


14. Confirm LUN removal by clicking **Yes**.



15. Select **Storage -> LUNs** in the EMC Navisphere main menu.

16. Locate and select the TOR SFS LUN. Click **Delete**.



17. Select **Storage -> Storage Pools** in the EMC Navisphere main menu.

18. Select the **RAID Groups** tab and locate the TOR SFS RAID group ID. The RAID group ID is specified in **Site Engineering Data**. Select the TOR SFS RAID group and click **Delete**.

Post Requirement

9 Glossary

Note: All abbreviations and expressions in bold are either described on this page or in 10 Terminology.

3PP	3rd Party Product. A product developed and/or maintained outside Ericsson organisation boundary.
AIS	Application Interface Specification
AIT	Automatic Install Tool, **CBA** component for software package (**Campaign**) generation
AMF	Availablility Management Framework (Implementation of **OpenSAF** in **CMW**
API	Application Programming Interface
BIOS	Basic Input/Output System
BS	Backup Server, executes Veritas Netbackupp Enterprise server
CBA	Component Based Architecture
CCB	Configuration Change Bundle
CKPT	Checkpoint Service
CLI	Command Line Interface
CLM	Cluster Membership Service
CM	Configuration Management
CMW	**Core MiddleWare**
CMW AMF	**Core MiddleWare** Availability Management Framework – Component of **CMW** which provides interfaces and functionality for Service Availability
CMWEA	**CMW** Environment Adapter(/Abstraction). Provides an interface layer to allow alternate platforms to be used with **CMW**.
COM	Common Operation Management
COTS	Commercial Off the Shelf
CRUD	Create, Read, Update, Delete

CSI	Component Service Instance . Represents the workload that the **AMF** assigns to a component.
DAC	Discretionary Access Control
DAE	Clariion Disk Array Enclosure
DHCP	Dynamic Host Configuration Protocol and tools
Distro	Short for distribution, distro is a specific distribution of Linux that is built from the common Linux operating system and includes additional applications. Red Hat, Debian and SuSe are all examples of a distro.
DM	Device Mapper, Linux Kernel
DNS	Domain Naming Services Protocol and tools
DTS	Distributed Tracing Service
DX	Developer Experience. Suite of Tools with **CMW**/**CBA** t o aid modeling etc.
EA	Environment Adapter. (A **CBA** component providing a common interface to **OS** services for **CBA** components, **CMW**)
ECC	Error-Correcting Code
EXT3/EXT4	Extended Filesystem v3, v4. Default Linux Journalling Filesystems
FC	Fibre Channel
FM	Fault Management
FOSS	Free Open Source Software- a Product developed by or with substantial involvement of general public community, usually available under the conditions of one of the commonly practiced open source licenses
HBA	Host Bus Adapter
iLO	Integrated Lights-Out
IMM	Information Model Management, **CMW** component
J2EE	Java Enterprise Edition, 2 defines the standard for developing multitier enterprise applications

JBoss	JavaBeans Open Source Software Application Server i s an application server that implements the Java Platform, Enterprise Edition (Java EE).
JavaOAM	JavaOAM is a Java library that gives the application developer access to cluster friendly services, **CBA** com ponent
JMS	The Java Module System specifies a distribution format for collections of Java code and associated resources.
KVM	Kernel Virtual Machine, default VM in **RHEL** 6
LBS	Linux Blade System, evolution of **LOTC**
LSB	Linux Standard Base
LDAP	LightWeight Directory Access Protocol
LDE	Linux Distribution Extension, **OS** independent version of **LOTC**
LDE-BASE	Linux Distribution Extension, Base/**LSB** Portion
LDE-RHEL	Linux Distribution Extension – **RHEL** specific portion
LDE-SLES	Linux Distribution Extension – **SLES** specific portion
Linux HA	Open Source High Availability Cluster Suite for Linux. Also known as **OpenAIS**.
LITP	Linux IT Platform. A collection of a Linux distribution and a variety of third party and in house developed applications that will be used by at least Ericsson **OSS PDU** to roll out their own applications to customers around the world. The LITP product is designed to be modular in its structure to allow for relatively easy replacement of any single component. LITP includes **C MW** as one of the tightly integrated **3PP** Products. LITP does not aim to provide for abstraction from **CMW**, instead it aims to compliment **CMW** own interfaces with these introducing required level of infrastructure awareness.
LITP Site	An instance of a deployment of the **LITP** Product in combination with a given selection of **OSS** applications suite in a specific customers environment. Characterized by potentially unique combination of products, computing hardware, storage, backup, networks and other infrastructure resources

LM	Licensing Manager, **CBA** component, wrapper for Sentinel
LOG	Logging Service
LOTC	Linux Open Telecom Cluster, **OS** services & configuration support for **CMW**, based on **SLES**
LSB	Linux Standards Base
LUN	Logical Unit Number - a number used to identify a logical unit, which is a device addressed by the **SCSI** pr otocol or similar protocols such as Fibre Channel or iSCSI
LV	Logical Volume. A logical representation of storage space, from a conceptual point of view similar to a partition. A Logical Volume could also be an aggregation of multiple Physical Volumes (PV).
LVM/LVM2	Logical Volume Manager
MBCS	Message Based Checkpoint Service
MDS	Message Distribution Service
MMAS	Multi Media Application Server. **CBA** compliant **J2EE** s erver.Version 3 is based on **JBoss**. Previous versions were based on ORCAS
MPIO	Multipath Input/Output. A means of providing redundant communication between systems. Commonly used between hosts and storage.
MS	Management Station, jumpstart server in OSS-RC & ENIQ on x86 blade systems. Controls system install and upgrade. Monitoring software, server side executes on this server. Previously known as Management Work Station (MWS).
MVC	Model, View, Controller framework pattern for web applications
NAS	Network Attached Storage
NBI	Northbound Interface
NFS	Network File System

NIC	Network Interface Controller
NTF	Notification Service, **CMW** Component
NTP	Network Time Protocol
O&M	Operations and Management/Operations and Maintenance
OCRI	OSS Common Runtime Infrastructure, old name for the **LITP**
OMBS	Operations and Maintenance Backup Server. Ericsson backup solution built upon Veritas Netbackup Enterprise Server
OpenSAF	Open-source middleware, consistent with **SAF** specific ations (**AIS**), and developed by **OpenSAF** Project
OS	Operating System
OSS	Operations Support Systems is a Programme Delivery Unit in Ericsson global organisation responsible for development and delivery of a variety of Applications
OVA	**OSS** Virtual Application. A test application which will be used to do ongoing feature-tests of the **LITP** platform. The application features will grow to test features developed within a sprint.
PDU	Product Development Unit
PL	Payload Node, **CMW** node. Executes application s/w
PV	Physical Volume. A hard disk, hard disk partition or Logical Unit Number (**LUN**) of an external storage device.
PXE	Preboot Execution Environment
RAID	Redundant Array of Independent Disks. Storage technology that combines multiple disk drive components into a logical unit. Data is distributed across the drives in one of several ways called RAID levels
RAM	Random Access Memory
RBAC	Role Based Access control

RDA	Reference Deployment Architecture
RHEL	Red Hat Enterprise Linux
RPM	Red Hat Package Manager
SAF	Service Availability Forum
SAN	Storage Area Network
SC	System Controller, **CMW** node for control of cluster. Can execute application s/w.
SCSI	Small Computer System Interface, standard for physically connecting and transferring data between computers and peripheral devices
SDK	Software Development Kit
SDP	Software Delivery Package. **Core Middleware** software packaging format - for example a bundle or **campaign**.
SELinux	Security Enhanced Linux
SFS	Symantec File Store. **NAS** application. Exports **SAN L UN**s over a network as an **NFS** share.
SG	Service Group, Contains one or more **SU**s that participates in a redundancy model
SI	Service Instance. Aggregates the **CSI**s within a **SU**
SLES	SuSE Linux Enterprise Server
SMF	Software Management Framework, a **CMW** component .
SP	Clariion Service Processor
SPI	Service Provider Instance
SU	Service Unit. Aggregates a set of components to provide a higher level of service
TBAC	Target Based Access Control
TFTP	Trivial File Transfer Protocol and associated tools
TIPC	Transparent Inter-process Communication protocol

VCS	Veritas Cluster Server, Commercial cluster product from Symantec
VG	Volume Group. The highest level of abstraction used within the Logical Volume Manager (**LVM**). It gathers together a collection of Logical Volumes (**LV**) and Physical Volumes (**PV**) into one administrative unit.
VLAN	Virtual Local Area Network
WWPN	World Wide Port Number
WWNN	World Wide Node Number
XML	eXtensible Markup Language
YMER	Name for New **OSS** Architecture, previously **OSS** ONE
YUM	Yellowdog Updater, Modified

10 Terminology

Note: All abbreviations and expressions in bold are either described on this page or in 9 Glossary.

Allocation	A temporary or permanent assignment of an available **Resource** in one of the **Deployment Description Inventory** Pools to an instance of a **Resource** in a specific **Role** inside specific **Node** and/or **Cluster** in a specific site.
Application Configurable Service	Provides functionality for storing application configuration information, built upon **IMM**.
Application	Software that provides functions which are required by an IT service. Each application may be part of more than one IT service. An application runs on one or more servers or clients.
Application Developer	**LITP** User who defines generic **LITP** solution architecture.
Bare Metal Restore	Technique in the field of data recovery and restoration where the backed up data is available in a form which allows one to restore a computer system from "bare metal", i.e. without any requirements as to previously installed software or operating system.
Blade	A blade is a self-contained server, which collectively fits into an enclosure with other blades. The blade servers themselves contain only the core processing elements, making them hot-swappable. A single blade typically holds hot-plug hard-drives, multiple I/O cards, memory, multi-function network interconnects, and Integrated Lights Out remote management. For additional storage, blades can connect to another storage blade or to a network attached **SAN**. **LITP** can be deployed either as a single blade or multiblade deployment e.g. DL380.
Blade Enclosure	Enclosure provides all the power, cooling, and I/O infrastructure needed to support modular server, interconnect and storage components e.g. HP c7000 Blade Enclosure. Sometimes known as a chassis.
BootMgr	Component of **LITP** that provides operating system kickstart functions for configured managed nodes, using **DHCP**, **PXE** and **TFTP**.

Bundle	A **CMW SDP** which contains the functionality to be installed. Can be comprised of **RPM**s, binaries, install scripts and ETT.xml file which describes the contents in terms of **SAF**.
Campaign	A Core **MiddleWare SDP**. Includes software, installation instructions and availability directives.The main types of **campaign** are Installation **campaign**s and Upgrade **campaign**s. **Campaign**s are also used to change the size of a **cluster**, and deploying an application on additional (or less) **nodes**. A **campaign** contains a campaign.xml which defines the **campaign**.
Campus Cluster	A single **Deployment** whic is geographically dispersed across two Sites, Site A and Site B.
Category	Representation of functionality of a **Server**. A **Server** m ay have one or many categories.
CfgMgr	Component of **LITP** that provides an interface to configuration enforcement system. The interface creates and deploys **Puppet** configuration files to the **M anagement Server**.
Client	The part of a client server application that the user directly interfaces with – for example, an email client.
Cloud	Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction.
Cluster	A group of **Node**s that can be used for resilient redundant execution of services.
Component	Represents a **resource** realized by zero or more processes. Components in **LITP** are most frequently comprised of chosen **3PP** product(s) and matching **LIT P API** software which implements an interface between generalized **LITP API** framework and vendor specific implementation. Chosen **3PP** products can be swapped out of the **LITP** at any point and the only impact should be limited to writing a new implementation of this "wrapper" software to facilitate new **3PP** to function with the use of same set of **LITP** generalized **API**s.

Component Definition	Specifies application component requirements in terms of: • **Resources** • **Service**s to be provided • Constraints
Configuration Manager	Provides functionality to manage the centralized configuration of a group of **Server**s e.g. **OS**, **3PP**, **Infra structure** and **Core MiddleWare**.
Core Middleware	Core MiddleWare (**CMW**) is a base middleware component that provides services needed for developing highly available applications in a clustered system. Core Middleware provides Availability, Logging, Software Management and Snapshotting features. **CMW** is an Ericsson software Product providing implementations for some of the key **OpenS AF** specifications tailored to provide platform **infrastruc ture** for development of platform aware applicaitons. **C MW** is central to the **OSS** Applications development strategy
CMW Cluster	Group of servers forming a single **CMW Cluster**
Definition	Documents the systems/environment specification for an application installation.
Deployment	A component representing a collection of **clusters** and/ or individual **nodes** within a logical deployment on a customer site. A group of **LITP** Components providing Interface implementations for bootstrapping **Node** quality Operating Systems on top of available **System**s. This component also provides Interfaces for deployment of Application software. Synonymous with the term Site. A Deployment comprises of **solution set**s with specified size parameters.
Deployment Description Definition	The highest level, **infrastructure**-independent description of the desired **deployment**. The old name landscape definition may still be seen.
Deployment Description Inventory	A full account of all **resource**s available and their allocation per definition requirements. The old name landscape inventory may still be seen.
Deployment Entity	Synonymous with Application.

Deployment Manager	Deployment Manager is responsible for the generation and execution of **Core MiddleWare** installation and upgrade **campaign**s (application software installion and upgrade). Also provides a **CLI** for human interaction. The old name Landscape Service may still be seen.
Deployment Model	A static representation of the Deployment Description. The Deployment Model is the actual object model of the system that is held in memory and can be exported to JSON
Execution	A group of **LITP** Components providing Interface implementations for runtime controls of the **LITP** Site, e.g. resilient execution, security, etc.
Execution Manager	Provides functionality that manages the execution and availability of software e.g. **VCS** and **CMW** managing **S olution Set**s and **3PP**s.
Geographical Redundancy	A **deployment** consisting of a Primary and Secondary Site where the Secondary Site provides redundancy against environmental damage of the Primary Site
Hypervisor	A hypervisor is a virtualisation platform that allows multiple operating systems to run on a host computer at the same time. Hypervisors are currently classified in two types: A Type 1 (or native or bare-metal) hypervisor is software that runs directly on a given hardware platform (as an operating system control program). A guest operating system thus runs at the second level above the hardware. Examples: include Xen, Oracle VM, VMware's ESX Server. A Type 2 (or hosted) hypervisor is software that runs within an operating system environment. A "guest" operating system thus runs at the third level above the hardware. Examples: VMware Server, VMware Workstation, VMware Fusion, the open source QEMU, Microsoft's Virtual PC and Virtual Server products, SWsoft's Parallels Workstation and Parallels Desktop.
Infrastructure	A group of **LITP** Components providing management Interface implementations for various kinds of **System**s and Storage, managed through the use of Inventory owned collections

Infrastructure Pools	Defines the list of **resource**s available to a site and the allocation of these **resource**s in a **system**/environment specification. Examples of these **resource**s are: • **Servers** • IP Addresses
Installation Engineer	**LITP** User who performs the install/upgrade activities on **LITP**.
Interfaces	A well defined way for a human or program to interact with the **LITP** Product or its Components. Content of these interfaces is determined by the **LITP** Architecture Defines the required interfaces to be implemented in a site/environment specification. Examples are: • EMC Storage **CLI** Interfaces • Veritas storage equipment **CLI** Interfaces
LogMgr	Component of **Deployment Manager** that provides interface to log management functions (enable, disable of logs).
LITP	Linux IT Platform. **LITP** is collection of a Linux distribution and a variety of third party and in-house developed applications that will be used by Ericsson to roll out their own applications to customers around the world. The **LITP** product is designed to be modular in its structure to allow for relatively easy replacement of any single component. **LITP** includes **CMW** as one of the tightly integrated **3PP** Products.
Management Server	The **LITP server**. Single point of management of one or more **Deployment**s. Includes such functions as Install & Upgrade, User Management, Monitoring, Configuration Management and Deployment Management
Management Work Station	Solaris Management Server.
Multiblade	Multiblade means an environment consisting of more than one blade. These blades can be either contained in one blade enclosure, or multiple blade enclosures. See **Blade** and **Blade Enclosure**.

Network	A Component of **LITP** providing for configuration data and generic control methods for Networks, for example - IPv4/IPv6 networks, associated 802.1Q **VLAN** information, information on available pools of IP address resources and factory methods for such resource allocation. In general terms, a network is a collection of hardware components and computers interconnected by communication channels that allow sharing of resources and information
Node	A Node results from installation of the selected Operating System on physical, virtual or cloud hardware. Node is a minimal possible block for assignment of **Roles**.
Northbound Interface	An Interface which is provided outside the boundary of the **LITP** Interface Product, for example to be used by the **OSS** applications to interact with the platform. Content of these interfaces is determined by the requirements of the **OSS** Architecture.
OSS	Operation Support Systems - an Ericsson Product Development Unit responsible for application development.
OSS Supporting Services	Group of services consisting of **DNS**, **DHCP**, **LDAP** and **NTP**
Peer Server	**LITP Client**. Previously Managed **Node**. Member of **CMW Cluster**, executes **OSS** application software or **OSS** Supporting Services
Plugin	A plugin extends the functionality of an application. A plugin adds to or extends the functionality of the Python system - by adding functionality to represent **NFS resources**, **server resources**, ip addresses etc.
Pool	The inventory of hardware and resources available to the **Deployment Manager** Pool provides access to more than 1 **resource** and tracks the **resource** allocation, such as the allocation of IP addresses to **servers**/components.
Product	Either software or hardware product that can be shipped to the customer either in its own right or as part of an aggregated product delivery.
Puppet	Puppet is a configuration management system supporting automated system administration tasks such as adding users, installing packages, and updating **server** configurations based on a centralized **server**.

Puppet Manifest	Describes the end state of a deployed **system**, and the **Puppet** software takes care of making sure the **system** meets that end state.
Resource	Any inventory object that can be used to deliver a desired state of the runtime environment. Resources include computing, storage and other kinds of resources such as IP addresses. Any kind of IT infrastructure resource that may be required and consumed by an Application. Resources are provided by **Service**s as part of the running Applications
Role	Role is used to describe the purpose of something or what it is used for. In **LITP** context, a role is a combination of **Service**s that can be associated with **Node**s and **Cluster**s. **Resource**s may be both provided and consumed by the Roles. Where **Resource**s are provided by a Role they will be included into **Deployment Description Inventory** to allow for subsequent discovery and configuration of the consuming Roles. Each Role defines any **resource**s it may need.
Sequence	A sequence allows the **LITP** designer to order the sequence in which resources are executed - regardless of other elements (Site, **Node**, **Role**).
Server	A physical (rack or blade) or virtual computer unit that is connected to a network and provides software functions that are used by other computers.
Service	An Interface providing way of describing a specific Application's Service in terms of **Resource**s required as well as general method for provisioning and execution control of these services, e.g. status/stop/start/restart.
Site/Site Definition	Site Definition - describes requirement for each of the sites, i.e. **cluster**s, **node**s and their mapping to **role**s.
Snapshot	The current state of a configuration item, process or any other set of data recorded at a specific point in time.
Service Availability Forum	Consortium of industry-leading communications and computing companies working together to develop and publish high availability and management software specifications.

Solution Architect	**LITP** User who defines customer **deployment** configurations based on an Application Designer generic solution architecture.
Solution	Collection of applications which provides a defined set of features for an operator.
Solution Definition	Contains **cluster** definitions and **server** type definitions.
Solution Set	Similar to Site Definition but more flexible, as it defines **server** types rather than **server**s.
Storage	Hardware providing persistence of data, applications, **system** state and Operating System.
System	A Component of the **LITP** providing for configuration data and generic control methods for a computing entity capable of executing a managed instance of a Linux Operating System within an **LITP** Site
System Administrator	**LITP** User who administers and monitors a **LITP deployment**.
System Integrator	**LITP** User who configures site installations.
SystemFactory	A Component of **LITP** providing factory and collection Interfaces for creation and manipulation of System instances of various kinds, e.g. standalone **server**s, **blade**s and virtual machines.
Upgrade Manager	A specific use case of the **Deployment Manager** where differences in Infrastructure and **Deployment** can be managed.
UserMgr	A human interface to classic user access management facilities, such as **LDAP**.
VCS Cluster	Connects multiple, independent **system**s into a management framework for increased availability. Each **system**, or **node**, runs its own operating system and cooperates at the software level to form a **cluster**.
Virtualisation	The creation of a virtual (rather than actual) version of something, such as a hardware platform, operating system, a storage device or network **resource**s.
Virtual Guest	Virtualised server (VM)

Virtual Host	Server hosting the **Hypervisor** upon which the VMs are deployed
Volume	A Component of **LITP** providing for configuration data and generic control methods for a Storage Network Area volume.
VolumeFactory	A **LITP** Component providing factory and collection Interfaces for creation and manipulation of Volume instances of various kinds. Implementations of this component interfaces may interact directly with specific **3PP SAN** Storage appliances, for example EMC Clariion.

11 Concepts

Abstract

This section discusses a few important concepts in LITP platform.

Concepts

LITP Deployment (Cobbler, Puppet, Deployment Description)

LITP handles installation and configuration of Cobbler and Puppet on the LITP Management Server (LMS). Cobbler is a universal boot server supporting software installation via network boot, reinstalls, and virtualization. Cobbler glues different technologies in order to make easy to build up the components of an efficient provisioning server, perfect for mass or frequent deployment of RedHat Linux systems.

Cobbler provides support for:

- PXE Server Support
- DHCP Server Integration
- Kickstart server with Templates
- YUM Repository Management

During LITP installation the Deployment Description and Puppet services are key to system definition and deployment. The Deployment Manager determines the allocation of resources from hardware to computing, storage resources, and configuration of monitoring resources. The Administrator defines resources available to the LITP platform and assigns them accordingly.

Puppet is a configuration management system supporting automated system administration tasks such as adding users, installing packages, and updating server configurations based on a centralized server. When using Puppet you need to define a policy (called a manifest) that describes the end state of your system, and the Puppet software takes care of making sure the system meets that end state. THe Puppet manifest is created once your Deployment Description is complete. For more information refer to the **LITP CLI User Guide**. <Will be added when available.>

The Puppet system is split into two parts: (i) a central server and (ii) the clients. The server runs a dæmon called **puppetmaster**. The clients run **puppetd**, which both connect to, and receives connections from the puppetmaster. The manifest is written on the puppetmaster. If Puppet is used to manage the LITP Management Server, it also runs the **puppetd** client.

A solution may be defined as:

- SAN Storage - block storage requirements for Nodes (for booting) and for storing data accessed via NFS
- NFS Storage - NFS mount requirements needed by Nodes/Applications
- Firewall Ports - Firewall TCP/IP port requirements needed by Nodes/Applications, provided by the NFS server
- Users/Groups - requirements for Users and/or Groups to be created
- Application Type - Application can be specified as J2EE (distinct from J2SE or anything else)
- Application Availability - an Application can be specified with an availability model (e.g. active/standby, active/active)
- Application Security Policy requirement - the SELinux policy required by an application or service

Core Middleware

Core MW is a base middleware component that provides services needed for developing highly available applications in a clustered system. Its functional scope includes availability and manageability support. On LITP it forms a core part of each node. From an administration perspective it is useful to understand the Core MW processes for troubleshooting, monitoring and for third party application integration. The key Core MW processes are **CMWPD** (PayLoad processes) and **CMWSC** (System Controller processes). The System Controller (SC) node executes administrative functions for the entire cluster and manages any services to be executed on the PayLoad nodes. The PayLoad (PL) nodes execute the bulk load of traffic on the cluster.

JBoss

The **JBoss (JavaBeans Open Source Software Application Server)** is an open-source Java EE-based application server runtime platform used for building, deploying, and hosting highly-transactional Java applications and services. As it is Java- based, the JBoss application server operates cross-platform: usable on any operating system that supports Java. A Java EE application developed according to Java EE standard can be deployed in any Java EE application server making it vendor independent.

Application servers are system software upon which web applications run. Application Servers consist of web server connectors , runtime libraries, database connectors, and the administration code needed to deploy, configure, manage, and connect these components on a web host. An application server runs behind a web server (e.g. Apache). See http://www.jboss.org/ for more information on JBoss.

SELinux Security Concepts

LITP security features are based on SELinux concepts such as identities, domains and roles. These features are summarized in this section.

SELinux identities

SE Linux identities determine which domains can be entered, that is they determine what actions the user can carry out. A SELinux identity and a UNIX login name are similar, however they behave differently. Running the su command does not change the user identity under SELinux, for example.

SELinux Domains

A domain is a list of tasks a process can execute, or what actions a process can perform on different types. A domain is similar to a UNIX uid. Every process runs in a domain, this determines the access a process has. Some examples of domains are sysadm_t - the system administration domain, and user_t- the general unprivileged user domain.

SELinux Types

A type is assigned to an object and determines the users who get access to that object. Objects are entities such as directories, files and sockets. Types are similar to domains, except a domain applies to process and a type applies to objects.

Type Enforcement

Within Linux, implementing Type Enforcement (TE), gives priority to Mandatory Access Control (MAC) over Discretionary Access Control (DAC). Access clearance is first given to a subject (for example, process) accessing objects (for example files, records, messages) based on rules defined in an attached security context.

SELinux Roles

Roles define which SELinux user identities can have access to which domains. The domains that a user role can access are predefined in policy configuration files. If a role is not authorised to enter a domain, then access is denied. Roles are created by the existence of one or more declarations in a TE rules file in $SELINUX_SRC/domains/*

SELinux Security Context

A security context contains the attributes that are associated with objects such as files, directories, processes and TCP sockets. A security context is made up of the identity, role and domain or type.

SELinux Policies

Policies are a set of rules governing things such as the roles a user has access to; which roles can enter which domains and which domains can access which types. You can edit your policy files according to how you want your system set up.