HPCC Systems® Administrator's Guide

Boca Raton Documentation Team

HPCC Systems® Administrator's Guide

Boca Raton Documentation Team

We welcome your comments and feedback about this document via email to <docfeedback@hpccsystems.com>

Please include **Documentation Feedback** in the subject line and reference the document name, page numbers, and current Version Number in the text of the message.

2025 Version 9.12.12-1

Introducing HPCC Systems® Administraton 4
Introduction 4
Architectural Overview 5
Hardware and Software Requirements 12
Hardware and Components 13
Thor Hardware 14
Roxie Hardware Configurations 15
Dali and Sasha Hardware Configurations 16
Other HPCC Systems Components 17
Routine Maintenance 18
Data Handling 19
Back Up Data 19
Log Files 24
Preflight 27
Preflight System Servers 28
Preflight Target Clusters 32
Preflight Thor 36
Preflight the Roxie Cluster 39
System Configuration and Management 42
Running the Configuration Manager 45
Environment.conf 52
Configuring HPCC Systems® for Authentication 57
Configuring ESP Server to use HTTPS (SSL) 72
User Security Maintenance 79
Dali and Security 123
Initialization under Systemd 128
Workunits and Active Directory 129
System Tools and Controls 130
Redefining nodes in a Thor Cluster 134
Best Practices 135
Cluster Redundancy 135
High Availability 137
Best Practice Considerations 139
Capacity Planning 144
Sample Sizings 146
System Resources 148
HPCC Systems Resources 148
Additional Resources 149

Introducing HPCC Systems® Administraton

Introduction

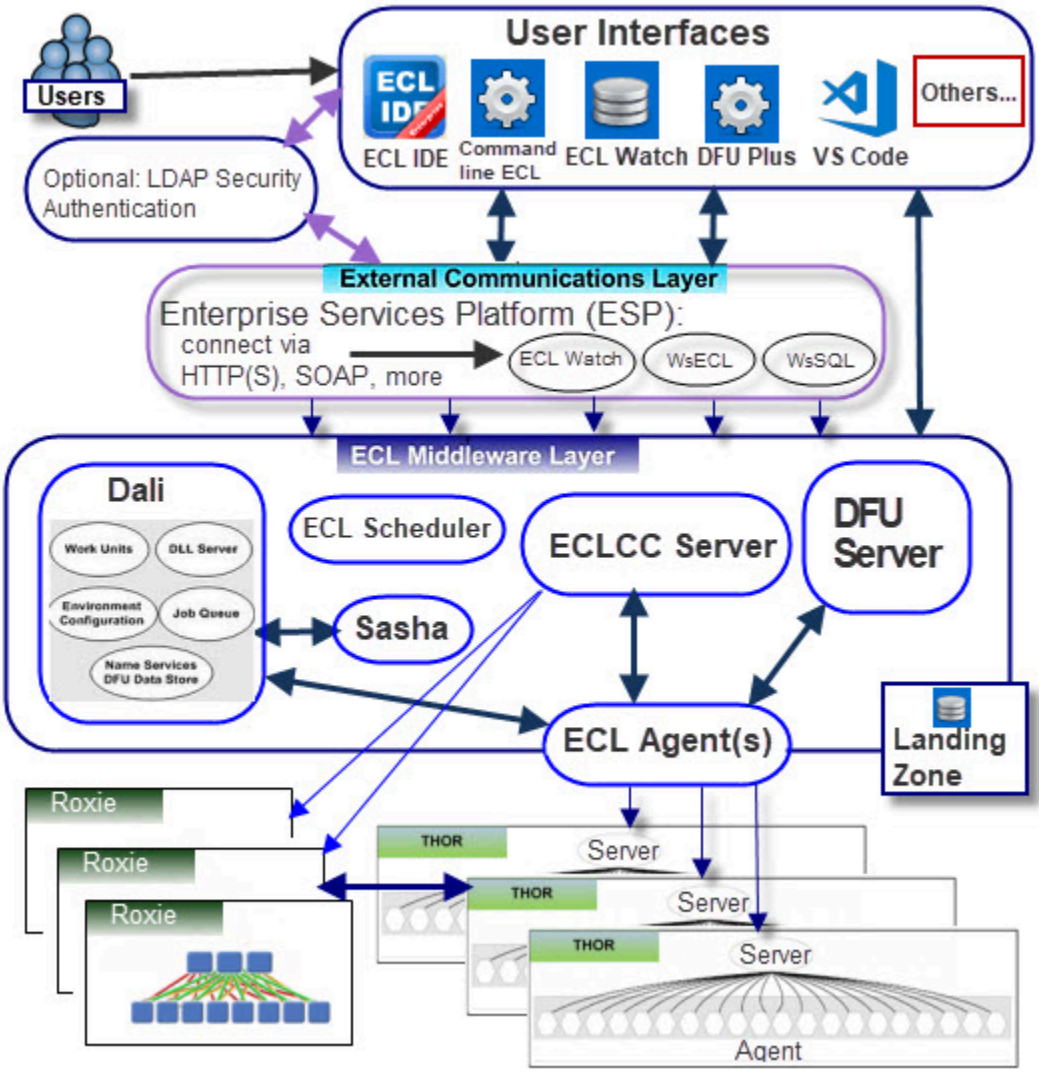
The HPCC (High Performance Computing Cluster) Systems platform is a massive parallel-processing computing platform that solves Big Data problems.

The HPCC Systems platform stores and processes large quantities of data, processing billions of records per second using massive parallel processing technology. Large amounts of data across disparate data sources can be accessed, analyzed, and manipulated in fractions of seconds. The HPCC Systems platform functions as both a processing and a distributed data storage environment, capable of analyzing terabytes of information.

Architectural Overview

An HPCC Systems® Platform consists of the following components: Thor, Roxie, ESP Server, Dali, Sasha, DFU Server, and ECLCC Server. LDAP security is optionally available.

Figure 1. HPCC Systems Architectural Diagram

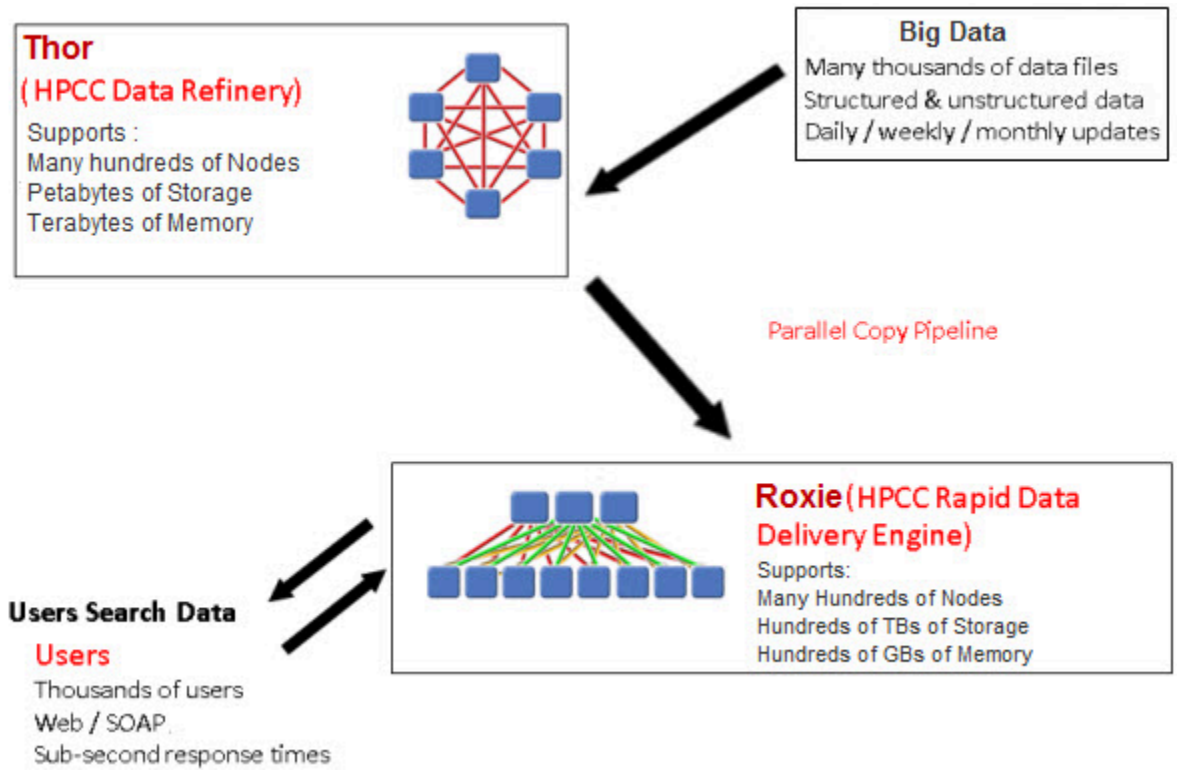


Data loading is controlled through the Distributed File Utility (DFU) server.

Data typically arrives on the landing zone (for example, by FTP). File movement (across components) is initiated by DFU. Data is copied from the landing zone and is distributed (sprayed) to the Data Refinery (Thor) by the ECL code. Data can be further processed via ETL (Extract, Transform, and Load process) in the refinery.

A single physical file is distributed into multiple physical files across the nodes of a cluster. The aggregate of the physical files creates one logical file that is addressed by the ECL code.

Figure 2. Data Processing



The data retrieval process (despraying) places the file back on the landing zone.

Clusters

An HPCC Systems environment contains clusters which you define and use according to your needs. The types of clusters used by HPCC Systems:

Thor

Data Refinery (Thor) -- Used to process every one of billions of records in order to create billions of "improved" records. ECL Agent (hThor) is also used to process simple jobs that would be an inefficient use of the Thor cluster.

Roxie

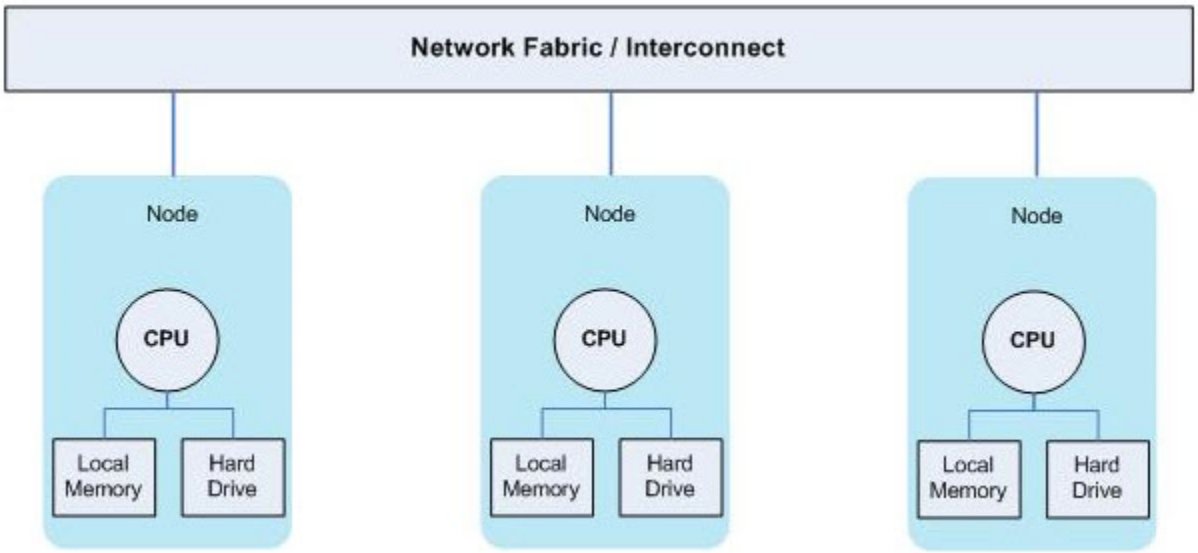
Rapid Data Delivery Engine (Roxie) -- Used to search quickly for a particular record or set of records.

Queries are compiled and published, usually in ECL Watch. Data moves in parallel from Thor nodes to the receiving Roxie nodes. Parallel bandwidth utilization improves the speed of putting new data into play.

ECL Agent

The ECL Agent's primary function is to send the job to execute on the appropriate cluster. The ECL Agent can act as a single-node cluster. That is called spawning an hThor cluster. hThor is used to process simple jobs that would otherwise be an inefficient use of Thor. For simple tasks, the ECL Agent will make a determination and perform the execution itself by acting as an hThor cluster.

Figure 3. Clusters



System Servers

The System Servers are integral middleware components of an HPCC Systems platform. They are used to control workflow and inter-component communication.

Dali

Dali is also known as the system data store. It manages workunit records, logical file directory, and shared object services. It maintains the message queues that drive job execution and scheduling.

Dali also performs session management. It tracks all active Dali client sessions registered in the environment, such that you can list all clients and their roles. (see *dalidiag -clients*)

Another task Dali performs is to act as the locking manager. HPCC Systems uses Dali's locking manager to control shared and exclusive locks to metadata.

Sasha

The Sasha server is a companion "housekeeping" server to the Dali server. Sasha works independently of, yet in conjunction with Dali. Sasha's main function is to reduce the stress on the Dali server. Wherever possible, Sasha reduces the resource utilization on Dali. A very important aspect of Sasha is coalescing, by saving the in-memory store to a new store edition.

Sasha archives workunits (including DFU Workunits) that are then stored in folders on a disk.

Sasha also performs routine housekeeping such as removing cached workunits and DFU recovery files.

Sasha can also run XREF, to cross reference physical files with logical metadata, to determine if there are lost/found/orphaned files. It then presents options (via EclWatch) for their recovery or deletion.

Sasha is the component responsible for removing expired files when the criteria has been met. The EXPIRE option on ECL's OUTPUT or PERSIST sets that condition.

DFU Server

DFU server controls the spraying and despraying operations used to move data in and out of Thor.

DFU services are available from:

- Standard libraries in ECL code.
- Client interfaces: Eclipse, ECL Playground, ECL IDE, and the ECL command line interface.
- DFU Plus command line interface.

ECLCC Server

ECLCC Server is the compiler that translates ECL code. When you submit ECL code, the ECLCC Server generates optimized C++ which is then compiled and executed. ECLCC Server controls the whole compilation process.

When you submit workunits for execution on Thor, they are first converted to executable code by the ECLCC Server.

When you submit a workunit to Roxie, code is compiled and later published to the Roxie cluster, where it is available to execute multiple times.

ECLCC Server is also used when the ECL IDE requests a syntax check.

ECLCC Server uses a queue to convert workunits one at a time, however you can have ECLCC Servers deployed in the system to increase throughput and they will automatically load balance as required.

ECL Agent

ECL Agent (hThor) is a single node process for executing simple ECL Queries.

ECL Agent is an execution engine that processes workunits by sending them to the appropriate cluster. ECL Agent processes are spawned on-demand when you submit a workunit.

ECL Scheduler

The ECL Scheduler provides a means of automating processes within ECL code or to chain processes together to work in sequence. ECL Scheduling is event-based. The ECL Scheduler monitors a Schedule list containing registered Workunits and Events and executes any Workunits associated with an Event when that Event is triggered.

ESP Server

ESP (Enterprise Service Platform) Server is the inter-component communication server. ESP Server is a framework that allows multiple services to be "plugged in" to provide various types of functionality to client applications via multiple protocols.

Examples of services that are plugged into ESP include:

- **WsECL:** Interface to published queries on a Roxie, Thor, or hThor cluster.
- **ECL Watch:** A web-based query execution, monitoring, and file management interface. It can be accessed via the ECL IDE or a web browser. See *Using ECL Watch*.

The ESP Server supports both XML and JSON Formats.

LDAP

You can incorporate a Lightweight Directory Access Protocol (LDAP) server to work with Dali to enforce the security restrictions for file scopes, workunit scopes, and feature access.

When LDAP is configured, you need to authenticate when accessing ECL Watch, WsECL, ECL IDE, or any other client tools. Those credentials are then used to authenticate any requests from those tools.

Topology Server

The topology server is an internal component used by ROXIE to keep track of the health of the different ROXIE processes in a cluster.

Client Interfaces

The following Client Interfaces are available to interact with the HPCC Systems platform.

ECL IDE

ECL IDE is a full-featured GUI providing access to your ECL code for ECL development. ECL IDE uses various ESP services via SOAP.

The ECL IDE provides access to ECL Definitions to build your queries. These definitions are created by coding an expression that defines how some calculation or record set derivation is to be done. Once defined, they can be used in succeeding ECL definitions.

ECL Watch

ECL Watch is a web-based query execution, monitoring, and file management interface. It can be accessed via ECL IDE, Eclipse, or a web browser. ECL Watch allows you to see information about and manipulate workunits. It also allows you monitor cluster activity and perform other administrative tasks.

Using ECL Watch you can:

- Browse through previously submitted workunits (WU). You can see a visual representation (graphs) of the data flow within the WU, complete with statistics which are updated as the job progresses.
- Search through files and see information including record counts and layouts or sample records.
- See the status of all system servers.
- View log files.
- Add users or groups and modify permissions.

See the *Using ECL Watch* Manual for more details.

ECL for Visual Studio

You can find and add the extension, ECL for Visual Studio Code in the Visual Studio Marketplace. This extension adds rich language support for the ECL language to VS Code.

Command Line Tools

Command line tools: **ECL, DFU Plus**, and **ECL Plus** provide command line access to functionality provided by the ECL Watch web pages. They work by communicating with the corresponding ESP service via SOAP.

See the *Client Tools* Manual for more details.

Support Utilities

There are a few additional components which are neither system servers nor client interfaces but nonetheless important in supporting HPCC Systems® tasks.

dafilesrv

The dafilesrv is a daemon that runs on system servers or physical locations on the HPCC Systems platform. This daemon enables the HPCC Systems components to have file access to physical file locations. File access could be spraying, despraying, copying, accessing logs, etc. Anything that requires a file from another node, must have dafilesrv running on that node. The dafilesrv daemon can be configured to allow SSL connections only, which secures remote access to physical files.

The dafilesrv daemon runs as a process on every HPCC Systems node. The daemon will typically keep running even if your system is stopped. That is an important fact to keep in mind if you are stopping your system for maintenance, installations, or upgrades. Even if you issue the HPCC Systems stop command, you would still need to stop the dafilesrv daemon.

One way to check and see if dafilesrv is running, is to issue a command such as the following:

```
ps -eaf | grep dafilesrv
```

Even if your HPCC System is stopped, you should still see that the dafilesrv is running.

Issue the following command to stop the dafilesrv daemon on a System V based system.

```
sudo /opt/HPCCSystems/sbin/hpcc-run.sh -a dafilesrv stop
```

Issue the following command to stop the dafiilesrv daemon on a Systemd based system.

```
hpcc-run.sh -c dafilesrv@mydafilesrv.service stop
```

You must have sudo permission level access in order to start or stop any of the HPCC Systems components, including dafilesrv.

After you verify that dafilesrv is not running you can then proceed with the installation.

ftslave

The ftslave is a process used when spraying data by dfuserver. The ftslave process is launched on demand as part of the spray process. There may be mutliple ftslave processes running on any given node at any given time, depending on the number of active sprays.

Hardware and Software Requirements

This chapter provides an overview of the hardware and software requirements for running the HPCC Systems platform optimally. While these requirements were significant when the HPCC Systems platform was first deployed many years ago, there have been substantial improvements in hardware since then. The platform now supports virtual containers and cloud deployments, making the requirements less significant even for large-scale (petabytes) bare-metal deployments. In fact, the HPCC Systems platform should perform satisfactorily on most modern hardware configurations.

Hardware and Components

This section provides some insight as to what sort of hardware and infrastructure optimally the HPCC Systems platform works well on. This is not an exclusive comprehensive set of instructions, nor a mandate on what hardware you must have. Consider this as a guide to use when looking to implement or scale your HPCC Systems platform. These suggestions should be taken into consideration for your specific enterprise needs.

The HPCC Systems platform is designed to run on commodity hardware, which makes building and maintaining large scale (petabytes) clusters economically feasible. When planning your cluster hardware, you will need to balance a number of considerations, including fail-over domains and potential performance issues. Hardware planning should include distributing HPCC Systems across multiple physical hosts, such as a cluster. Generally, one type of best practice is to run the HPCC Systems platform processes of a particular type, for example Thor, Roxie, or Dali, on a host configured specifically for that type of process.

Thor Hardware

Thor slave nodes require a proper balance of CPU, RAM, network, and disk I/O in order to operate most efficiently. A single Thor slave node works optimally when allocated 4 CPU cores, 8GB RAM, 1Gb/sec network and 200MB/sec sequential read/write disk I/O.

Hardware architecture can provide higher value within a single physical server. In such cases you can use multi-slave to configure your larger physical servers to run multiple Thor slave nodes per physical server.

It is important to note that the HPCC Systems platform by nature is a parallel processing system and all Thor slave nodes will be exercising at precisely the same time. So when allocating more than one HPCC Systems Thor slave per physical machine assure that each slave meets the recommended requirements.

For instance, 1 physical server with 48 cores, 96GB RAM, 10Gb/sec network and 2GB/sec sequential I/O would be capable of running ten (10) HPCC Systems Thor slaves at optimal efficiency. The order of optimization for resource usage in a Thor slave node is disk I/O 60%, network 30%, and CPU 10%. Any increase in sequential I/O will have the most impact on speed, followed by improvements in network, followed by improvements in CPU.

Network architecture is also an important consideration. HPCC Systems Thor nodes work optimally in a streamlined network architecture between all Thor slave processes.

RAID is recommended and all RAID levels suitable for sequential read/write operations and high availability are acceptable. For example, RAID1, RAID10, RAID5 (preferred), and RAID6.

Roxie Hardware Configurations

HPCC Systems Roxie processes require require a proper, yet different (from Thor) balance of CPU, RAM, network, and disk I/O in order to ensure efficient operations. A single HPCC Systems Roxie node works optimally when allocated 6 or more CPU cores, 24GB RAM, 1Gb/sec network backbone, and 400/sec 4k random read IOPS.

Each HPCC Systems Roxie node is presented two hard drives, each capable of 200/sec 4k random seek IOPS. Hard drive recommendations for Roxie efficiency are 15K SAS, or SSD. A good rule of thumb is the more random read IOPS the better and faster your Roxie will perform.

Running multiple HPCC Systems Roxie nodes on a single physical server is not recommended, except in the cases of virtualization or containers.

Configure your system to balance the size of your Thor and Roxie clusters. The number of Roxie nodes should never exceed the number of Thor nodes. In addition, the number of Thor nodes should be evenly divisible by the number of Roxie nodes. This ensures an efficient distribution of file parts from Thor to Roxie.

Dali and Sasha Hardware Configurations

The HPCC Systems platform Dali processes store cluster metadata in RAM. For optimal efficiency, provide at least 48GB of RAM, 6 or more CPU cores, 1Gb/sec network interface and a high availability disk for a single HPCC Systems Dali. The HPCC Systems platform Dali processes are one of the few native active/passive components. Using standard "swinging disk" clustering is recommended for a high availability setup. For a single HPCC Systems platform Dali process, any suitable High Availability (HA) RAID level is fine.

Sasha only stores data to locally available disks, reading data from Dali then processing it by archiving workunits (WUs) to disk. It is beneficial to configure Sasha for a larger amount of archiving so that Dali does not keep too many workunits in memory. This requires a larger amount of disk space.

Allocating greater disk space for Sasha is sound practice as configuring Sasha for more archiving better benefits Dali. Since Sasha assists Dali by performing housekeeping, it works best when on its own node. Ideally, you should avoid putting Sasha and Dali on the same node, because the node that runs these components is extremely critical, particularly when it comes to recovering from losses. Therefore, it should be as robust as possible: RAID drives, fault tolerant, etc.

Sasha/Dali Interactions

A critical role of Sasha is in coalescing. When Dali shuts down, it saves its in-memory store to a new store edition by creating a new *dalisdsXXXX.xml*, where XXXX is incremented to the new edition. The current edition is recorded by the filename store.XXXX

An explicit request to save using *dalidiag*:

```
dalidiag . -save
```

The new editions, as per the above example are created the same way. During an explicit save, all changes to SDS are blocked. Therefore all clients will block if they try to make any alteration until the save is complete.

There are some options (though not commonly used) that can configure Dali to detect quiet/idle time and force a save in exactly the same way an explicit save request does, meaning that it will block any write transactions while saving.

All Dali SDS changes are recorded in a delta transaction log (in XML format) with a naming convention of *daliincXXXX.xml*, where XXXX is the current store edition. They are also optionally mirrored to a backup location. This transaction log grows indefinitely until the store is saved.

In the normal/recommended setup, Sasha is the primary creator of new SDS store editions. It does so on a schedule and according to other configuration options (for example, you could configure for a minimum delta transaction log size). Sasha reads the last saved store and the current transaction log and replays the transaction log over the last saved store to form a new in-memory version, and then saves it. Unlike the Dali saving process, this does not block or interfere with Dali. In the event of abrupt termination of the Dali process (such as being killed or a power loss) Dali uses the same delta transaction log at restart in order to replay the last save and changes to return to the last operational state.

Other HPCC Systems Components

ECL Agent, ECLCC Server, DFU Server, the Thor master, and ECL Watch are administrative processes which are used for supporting components of the main clusters.

For maximum efficiency you should provide 24GB RAM, 6+ CPU cores, 1Gb/sec network and high availability disk(s). These components can be made highly available in an active/active fashion.

Routine Maintenance

In order to ensure that your HPCC Systems platform keeps running optimally, some care and maintenance is required. The following sections address routine maintenance tasks for your HPCC Systems platform.

Data Handling

When you start working with your HPCC Systems platform, you will want to have some data on the system to process. Data gets transferred to the HPCC Systems platform by a process called a spray. Likewise to get data out from an HPCC Systems platform it must be desprayed.

As the HPCC Systems platform is a computer cluster the data gets deployed out over the nodes that make up the cluster. A *spray* or import is the relocation of a data file from one location (such as a Landing Zone) to a cluster. The term spray was adopted due to the nature of the file movement -- the file is partitioned across all nodes within a cluster.

A *despray* or export is the relocation of a data file from a Data Refinery cluster to a single machine location (such as a Landing Zone). The term despray was adopted due to the nature of the file movement -- the file is reassembled from its parts on all nodes in the cluster and placed in a single file on the destination.

A *Landing Zone* (or drop zone) is a physical storage location defined in your system's environment. There can be one or more of these locations defined. A daemon (dafilesrv) must be running on that server to enable file sprays and desprays. You can spray or despray some files to your landing zone through ECL Watch. To upload large files, you will need a tool that supports the secure copy protocol, something like a WinSCP.

For more information about the HPCC Systems platform data handling see the *HPCC Systems® Data Handling* and the *HPCC Systems® Data Tutorial* documents.

Back Up Data

An integral part of routine maintenance is the backup of essential data. Devise a backup strategy to meet the needs of your organization. This section is not meant to replace your current backup strategy, instead this section supplements it by outlining special considerations for HPCC Systems®.

Backup Considerations

You probably already have some sort of a backup strategy in place, by adding HPCC Systems® into your operating environment there are some additional considerations to be aware of. The following sections discuss backup considerations for the individual HPCC Systems components.

Dali

Dali can be configured to create its own backup. It is strongly recommended that the backup be kept on a different server or node for disaster recovery purposes. You can specify the Dali backup folder location using the Configuration Manager. You may want to keep multiple generations of backups, to be able to restore to a certain point in time. For example, you may want to do daily snapshots, or weekly.

You may want to keep backup copies at a system level using traditional methods. Regardless of method or scheme you would be well advised to backup your Dali.

You should try to avoid putting Dali, Sasha, and even your Thor Master on the same node. Ideally you want each of these components to be on separate nodes to not only reduce the stress on the system hardware (allowing the system to operate better) but also enabling you to recover your entire environment, files, and workunits in the event of a loss. In addition it would affect every other Thor/Roxie cluster in the same environment if you lose this node.

Restoring Dali from backup

If configured correctly, Dali creates a backup or mirror copy to a secondary location on another physical server. (Bare-metal only).

Systems can be configured with their own scheduled backup to create a snapshot of the primary store files to a custom location. The same steps apply when using a snapshot copy of a backup set as when using the mirror copy. In other words, this technique applies to either bare-metal or k8s deployments.

The Dali meta files are comprised of:

1. **store.<NNNN>** (e.g., store.36). This file is a reference to the current Dali meta file edition. There should never be more than one of these files. The NNNN is used to determine the current base and delta files in use.

2. **dalisds<NNNN>.xml** (e.g., dalisds36.xml). This is the main Dali meta info file, containing all logical file, workunit, and state information. Sasha (or Dali on save) periodically creates new versions (with incrementally rising NNNN's). It will keep the last T copies (default 10) based on the configuration option "keepStores".

3. **daliinc<NNNN>.xml** (e.g., daliinc36.xml). This is the delta transaction log. Dali continuously writes to this file, recording all changes that are made to any meta data. It is used to playback changes and apply them to the base meta info from the dalisds<NNNN>xml file.

   Specifically, when Sasha creates a new store version, it loads the base file (e.g., dalisds36.xml), then loads and applies the delta file (e.g., daliinc36.xml). Sasha then has its own independent representation of the current state and saves a new base file (e.g., dalisds(NNNN+1).xml).

4. **dalidet<NNNN>.xml** (e.g., dalidet36.xml). This file is created at the point that Sasha starts the process of creating a new base file. At which point it atomically renames the delta transaction file to a 'det' file (short for 'detached'). For example, it renames daliinc36.xml to dalidet36.xml. Dali then continues to write new transactions to daliinc36.xml.

5. **dalisds_<MMMM>.bv2** files. These files are in effect part of the main store (part of dalisdsNNNN.xml). They are single large values that were deemed too big to keep in Dali memory, and written to disk separately instead (and are loaded on demand).

If Dali is shutdown cleanly and saves its files as expected, the daliinc*.xml and dalidet*.xml files are not needed, since it saves the entire state of the store directly from internal memory, and on startup, there is no daliincNNNN.xml or dalidetNNNN.xml related to the new version.

These transaction delta files are only used by Sasha when creating new versions of the base store or if Dali has been stopped abruptly (e.g., machine rebooted). If Dali restarts after an unclean exit, there will be a daliincNNN.xml (and possibly a dalidetNNNN.xml file if Sasha was actively creating a new version at the time). In those cases, Dali will load these files in addition to the base file.

By default Dali's main data store directory is /var/lib/HPCCSystems/hpcc-data/dali/ . In other words, all meta data is written to and read from this location.

When restoring from a backup:

1. Make sure Dali is not running

2. Make sure the /var/lib/HPCCSystems/hpcc-data/dali folder is empty.

3. Copy all pertinent backup file into the /var/lib/HPCCSystems/hpcc-data/dali folder:

- One store.NNNN file
- One dalisdsNNNN.xml file
- <=1 daliincNNNN.xml file (only if present)
- <=1 dalidetNNNN.xml file (only if present)
- All dalisds_MMMM.bv2 files.

Other/older dalisds/daliinc/dalidet editions could be copied, but the above are the only ones that will be used. In other words, only the NNNN version based on the single store.NNNN file will be loaded.

The automatic back to a mirror location is bare-metal only. In a cloud deployment, it is assumed that the storage choices provided by the cloud provider are providing redundancy, such as multi-zone replication.

In either case, and/or if a manual strategy has been used to copy Dali's files on a schedule, the process of restoring from a backup should be the same.

Sasha

Sasha is the component that does the SDS coalescing. It is normally the sole component that creates new store editions. It's also the component that creates the XREF metadata that ECLWatch uses. Be aware that Sasha can create quite a bit of archive data. Once the workunits are archived they are no longer available in the Dali data store. The archives can still be accessed through ECL Watch by restoring them to Dali.

If you need high availability for archived workunits, you should back them up at a system level using traditional backup methods.

DFU Server

DFU Server has no data. DFU workunits are stored in Dali until they are archived by Sasha.

ECLCC Server

ECLCC Server stores no data. ECL workunits are stored in Dali and archived by Sasha.

ECL Agent

ECL Agent stores no data.

ECL Scheduler

ECL Scheduler stores no data. ECL Workunits are stored in Dali.

ESP Server

ESP Server stores no data. If you are using SSL certificates, public and private keys they should be backed up using traditional methods.

Thor

Thor, the data refinery, as one of the critical components of HPCC Systems® needs to be backed up. Backup Thor by configuring replication and setting up a nightly back up cron task. Backup Thor on demand before and/or after any node swap or drive swap if you do not have a RAID configured.

A very important part of administering Thor is to check the logs to ensure the previous backups completed successfully.

Backupnode

Backupnode is a tool that is packaged with the HPCC Systems platform. Backupnode allows you to backup Thor nodes on demand or in a script. You can also use backupnode regularly in a crontab or by adding a backupnode component with Configuration Manager to your environment. You would always want to run it on the Thor master of that cluster.

The following example is one suggested way for invoking backupnode manually.

```
  /bin/su - hpcc -c "/opt/HPCCSystems/bin/start_backupnode thor" &
```

The command line parameter must match the name of your Thor cluster. In your production environment, it is likely that you would provide descriptive names for your Thor clusters.

For example, if your Thor cluster is named thor400_7s, you would call start_backupnode thor400_7s.

```
  /bin/su - hpcc -c "/opt/HPCCSystems/bin/start_backupnode thor400_7s" &
```

Backupnode run regularly

To run backupnode regularly you could use cron. For example, you may want a crontab entry (to backup thor400_7s) set to run at 1am daily:

```
  0 1 * * * /bin/su - hpcc -c "/opt/HPCCSystems/bin/start_backupnode thor400_7s" &
```

Backupnode writes out its activity to a log file. That log can be found at:

/var/log/HPCCSystems/backupnode/MM_DD_YYYY_HH_MM_SS.log

The (MM) Month, (DD) Day, (YYYY) 4-digit Year, (HH) Hour, (MM) Minutes, and (SS) Seconds of the backup comprising the log file name.

The main log file exists on the Thor master node. It shows what nodes it is run on and if it finished. You can find other backupnode logs on each of the Thor nodes showing what files, if any, it needed to restore.

It is important to check the logs to ensure the previous backups completed successfully. The following entry is from the backupnode log showing that backup completed successfully:

```
00000028 2014-02-19 12:01:08 26457 26457 "Completed in 0m 0s with 0 errors"
00000029 2014-02-19 12:01:08 26457 26457 "backupnode finished"
```

Roxie

Roxie data is protected by three forms of redundancy:

- **Original Source Data File Retention:** When a query is published, the data is typically copied from a remote site, either a Thor or a Roxie. The Thor data can serve as backup, provided it is not removed or altered on Thor. Thor data is typically retained for a period of time sufficient to serve as a backup copy.
- **Peer-Node Redundancy:** Each Agent node typically has one or more peer nodes within its cluster. Each peer stores a copy of data files it will read.
- **Sibling Cluster Redundancy:** Although not required, Roxie may run multiple identically-configured Roxie clusters. When two clusters are deployed for Production each node has an identical twin in terms of queries and/or data stored on the node in the other cluster. This configuration provides multiple redundant

copies of data files. With three sibling Roxie clusters that have peer node redundancy, there are always six copies of each file part at any given time; eliminating the need to use traditional backup procedures for Roxie data files.

Landing Zone

The Landing Zone is used to host incoming and outgoing files. This should be treated similarly to an FTP server. Use traditional system level backups.

Misc

Backup of any additional component add-ons, your environment files (environment.xml), or other custom configurations should be done according to traditional backup methods.

Log Files

The HPCC Systems platform provides a wealth of information which can be used to debug, track transactions, application performance, and troubleshooting purposes. You can review the HPCC Systems platform messages as they are reported and captured in the log files. Log files can help you in understanding what is occurring on the system and useful in troubleshooting.

Component Logs

HPCC Systems component files are written to **/var/log/HPCCSystems** (default location). You can optionally configure your HPCC Systems platform to write the logs to a different directory. You should know where the log files are, and refer to the logs first when troubleshooting any issues.

You can find the log files in subdirectories named corresponding to the components that they track. For example, if you have a Thor cluster named 'mythor' its logs would be found in a subdirectory named 'mythor'.

In each of the component subdirectories, there are several log files. Most of the log files use a logical naming convention that includes the component name and timestamp in the name of the log file. There is also usually a link for the component with a simple name, such as esp.log which is a short cut to the latest current log file for that component.

Understanding the log files, and what is normally reported in the log files, helps in troubleshooting the HPCC Systems platform clusters.

As part of routine maintenance you may want to backup, archive, and remove the older log files. Some log files can grow quite large and you should be mindful of available disk space where the system writes out its log files. It could prove to be helpful to separate your log file directory from your OS or component file system.

The Log Fields

The log files of all major HPCC Systems components provide specific information relative to each component. The information that gets logged is configurable. HPCC Systems component logs follow a format defined in the environment.conf file **logfields** setting. Optionally you can configure to report additional information.

By default, the logs are configured to report the following columns: TIM, DAT, MLT, MID, PID, TID, COD, QUO, PFX

MID	Message ID
DAT	Date
TIM	Time
MLT	MilliTime
PID	Process ID
TID	Thread ID
PFX	Prefix (not output on all messages)
QUO	Quoted message. The actual message reported.
COD	Code

Below is an example ESP log entry from /var/log/HPCCSystems/myesp/esp.log (based on the stock default **logfields setting**):

```
000001EE 2018-08-29 15:00:46.653 17746 17775
```

```
 "TxSummary[activeReqs=2;contLen=-1;rcv=2ms;user=@127.0.0.1;req=GET wsdfu;total=3ms;]"
```

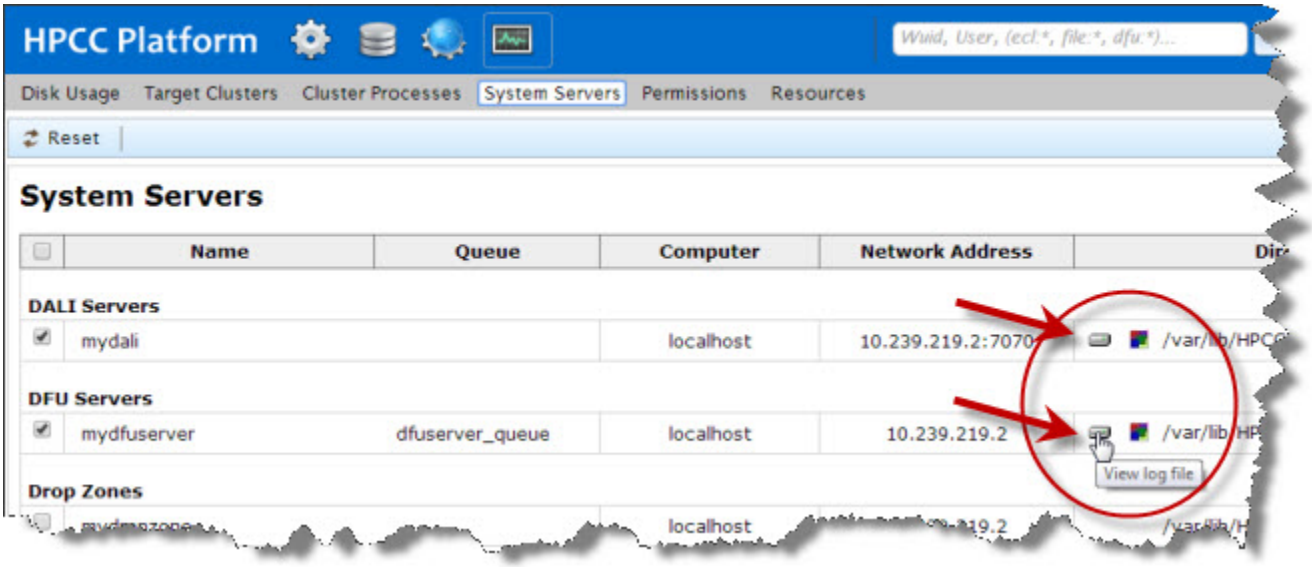
For more information about configuring the log file contents see the environment.conf section.

Accessing Log Files

You can access and view the log files directly by going to the component log directory from a command prompt or a terminal application. You can also view the component log files through ECL Watch.

To view logs on ECL Watch, click on the **Operations** icon, then click on the **System Servers** link. That opens the System Servers page in ECL Watch. There are several HPCC Systems components listed on that page. In the **Directory** column for each component there is a computer drive icon. Click the icon in the row for the component log you wish to view.

Figure 4. Logs in ECL Watch



You can also view log files from the other links under the Operations icon in ECL Watch.

1. Click on the **Target Clusters** link to open the tab with links to your system's clusters.
2. Click on the computer drive icon (circled in red in the above figure), in the row of the cluster and node of the component log you wish to view.

To view cluster process logs:

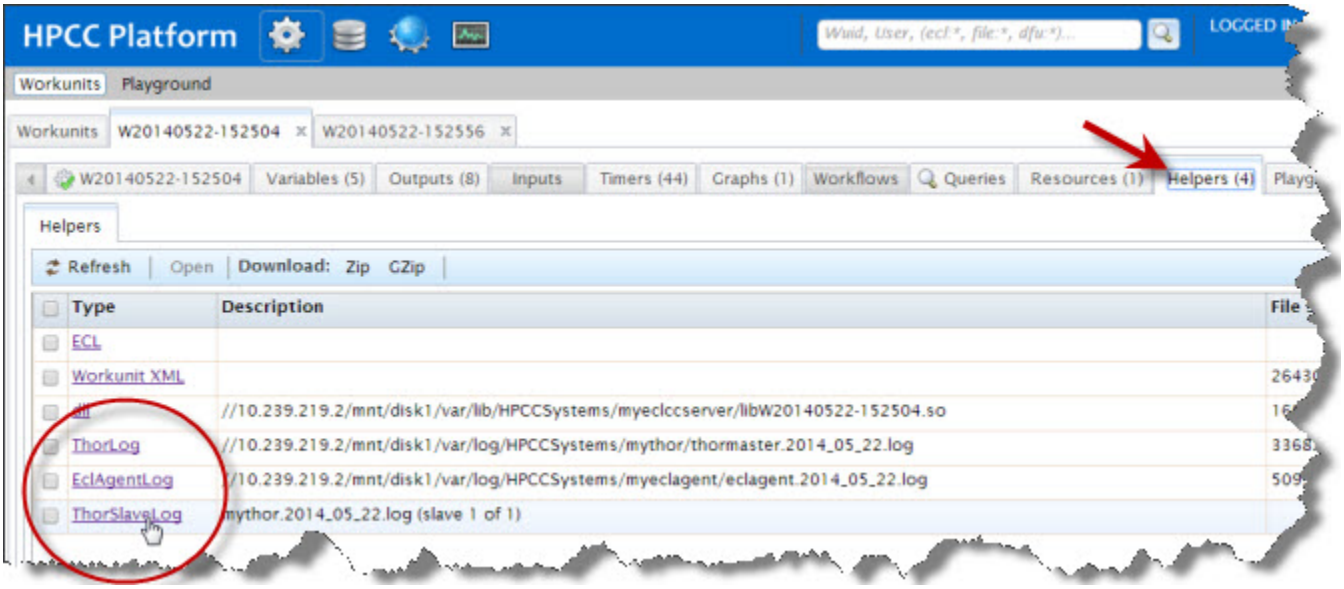
1. Click on the **Cluster Processes** link to open the tab with links to your system's clusters processes.
2. Click on the cluster process you wish to view more information about.

   For example, click on the **myroxie** link. You will then see a page of all that components nodes. You will see computer drive icon, in the row of each node. Click that icon to see the logs for the cluster process for that node.

Log files in ECL Workunits

You can also access the Thor or ECL Agent log files from the ECL Workunits. (not available for Roxie workunits) In ECL Watch when examining the Workunit details, you will see a **Helpers** tab. Click on the Helpers tab to display the relevant log files for that particular workunit.

Figure 5. Logs in ECL Watch Workunits



Preflight

The first step in certifying that the platform is installed and configured properly is to run a preflight check on the components. This ensures that all machines are operating and have the proper executables running. This also confirms there is adequate disk space, available memory, and acceptable available CPU % values.

- Open ECL Watch in your browser using the following URL:

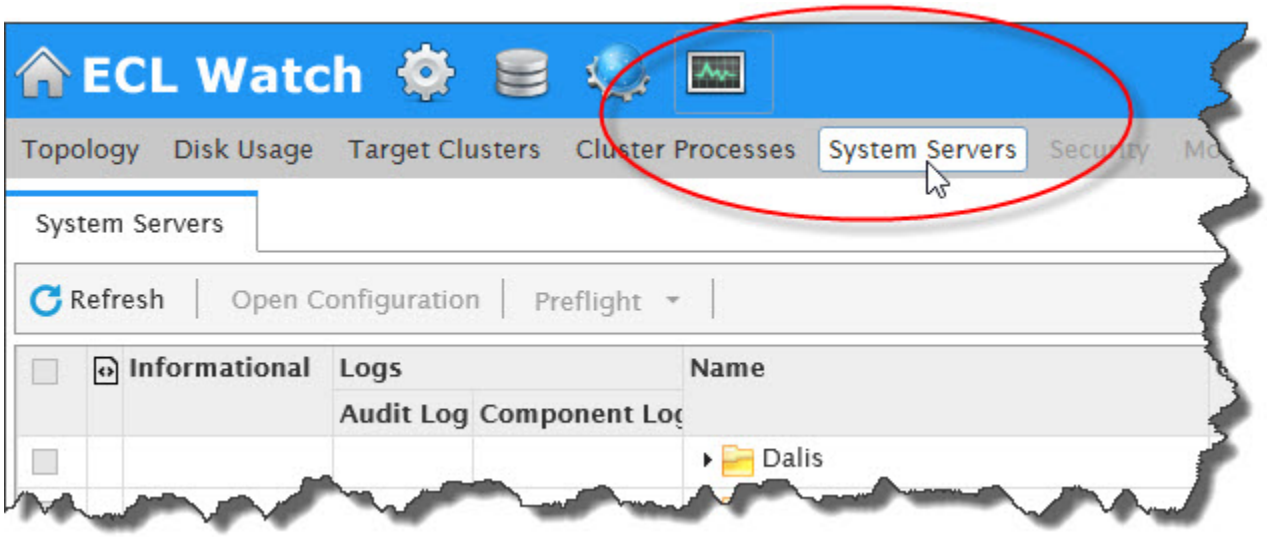
  http://nnn.nnn.nnn.nnn:pppp (where nnn.nnn.nnn.nnn is your ESP Server's IP Address and pppp is the port. The default port is 8010)

	**Note:** That your IP address could be different from the ones provided in these figures. Please use the IP address provided by your installation.

Preflight System Servers

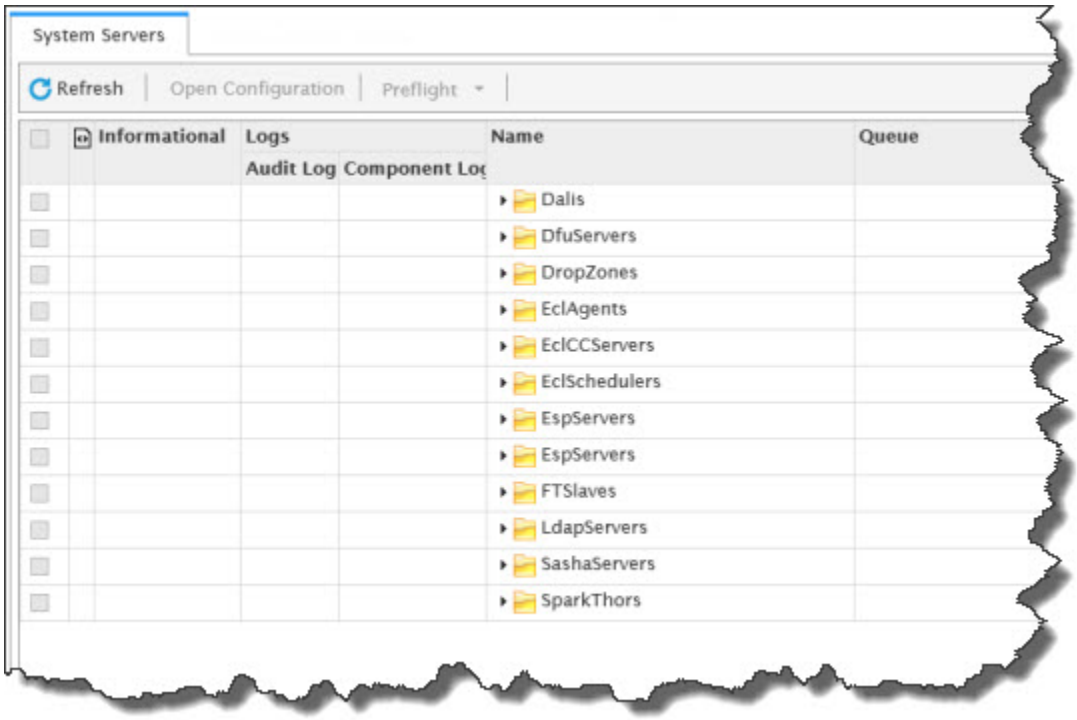
1. Click on the **Operations** icon then click on the **System Servers** link.

Figure 6. System Servers link



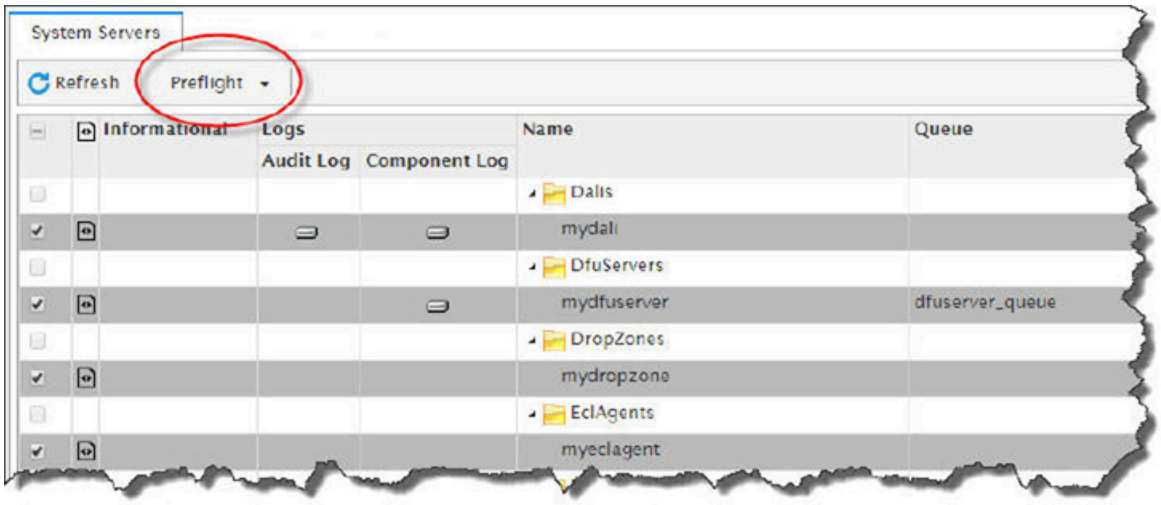
A screen similar to the following displays.

Figure 7. System Servers page



2. Expand the folder for the System Server then check the box next to the desired component(s).

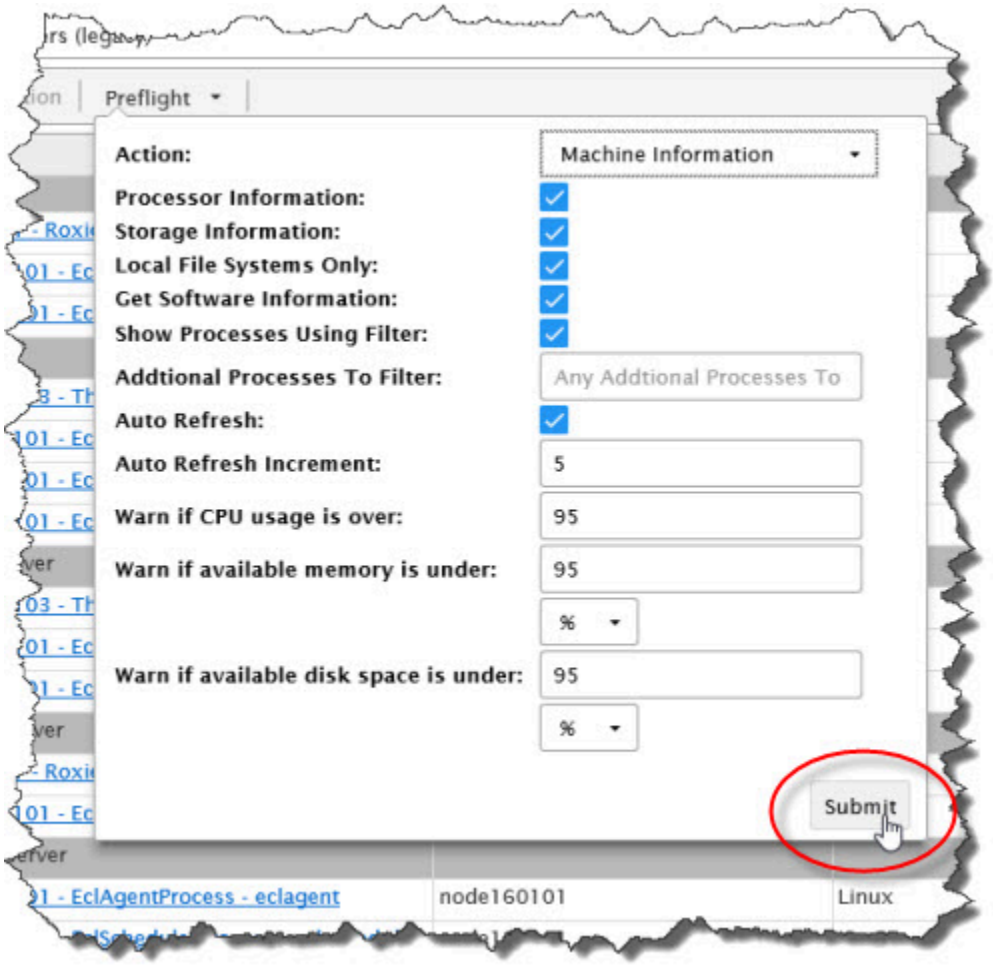
Figure 8. Select System Servers



With the servers selected, the preflight action button activates and you can press it to display the preflight options.

3. Check or uncheck any desired options then Press the **Submit** button to start preflight.

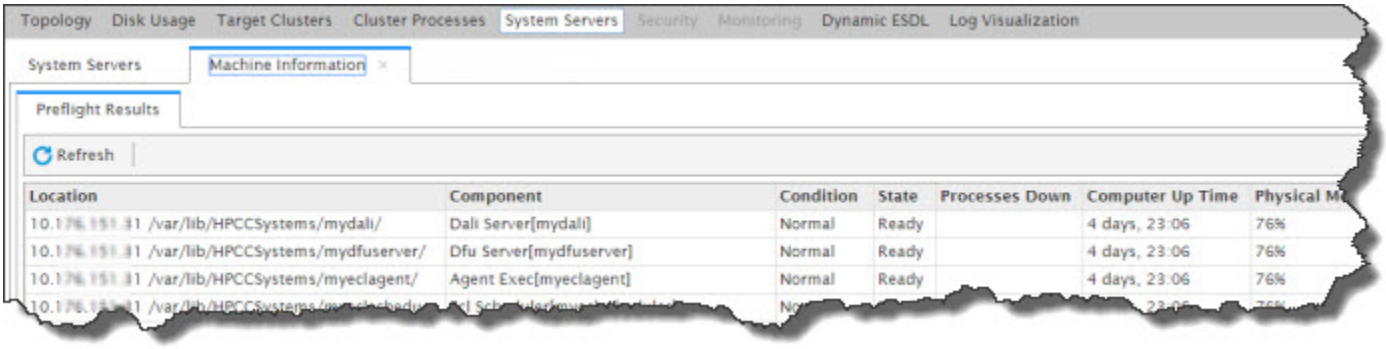
Figure 9. Submit



EXPECTED RESULTS:

After pressing Submit, a screen similar to the following displays.

Figure 10. System Component Information

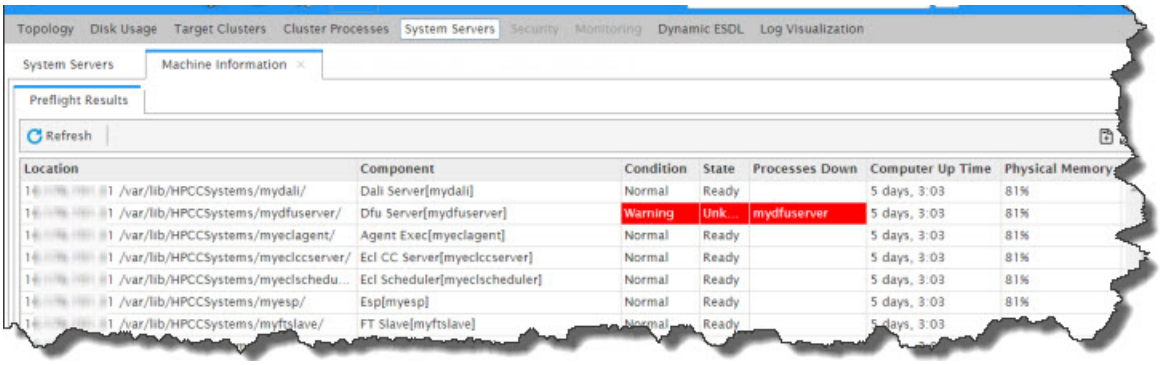


This screen displays information about the selected system components. This information indicates whether the components are actually running appropriately. The resulting page shows useful information about each component. The component name, location, condition, the component state, how long the component has been up and running, the amount of disk usage, memory usage and other information is available at a glance.

If there are any alerts, the component(s) are highlighted, indicating they require further attention.

For example, the following image indicates there is an issue with the DFU Server.

Figure 11. System Server Alert

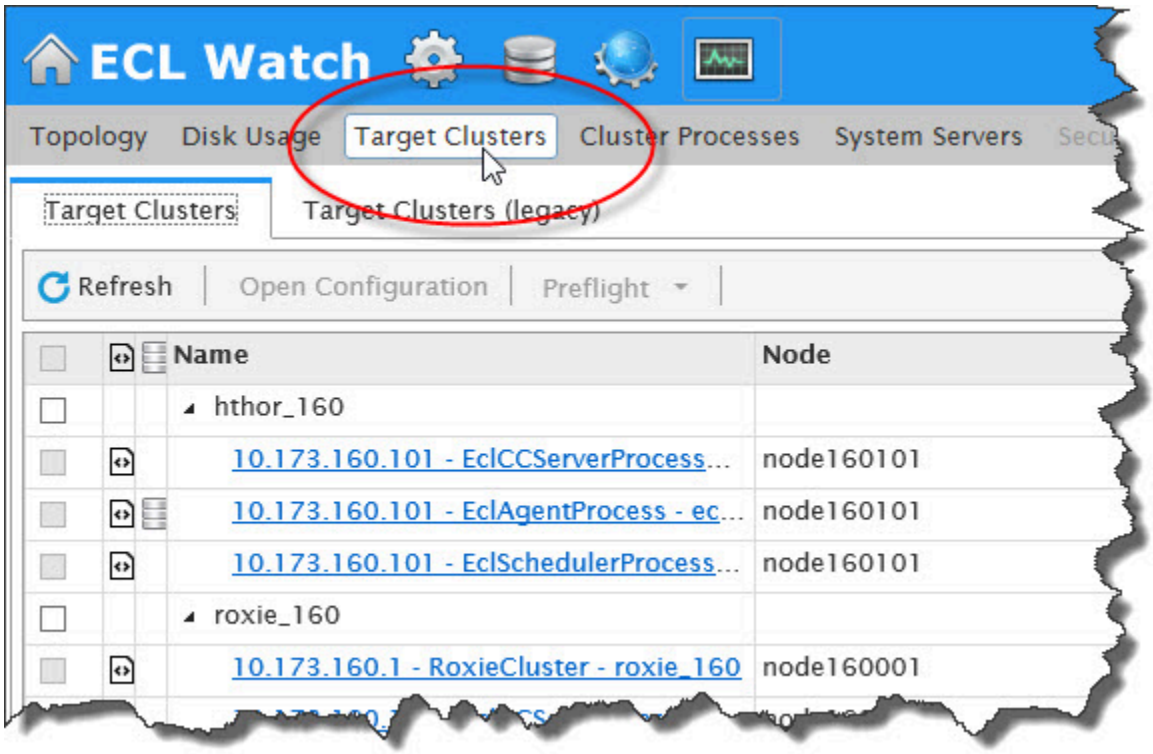


Preflight Target Clusters

Use the Target Clusters link to preflight all your clusters.

1. Click on the **Operations** icon then click on the **Target Clusters** link.

Figure 12. Target Clusters Link



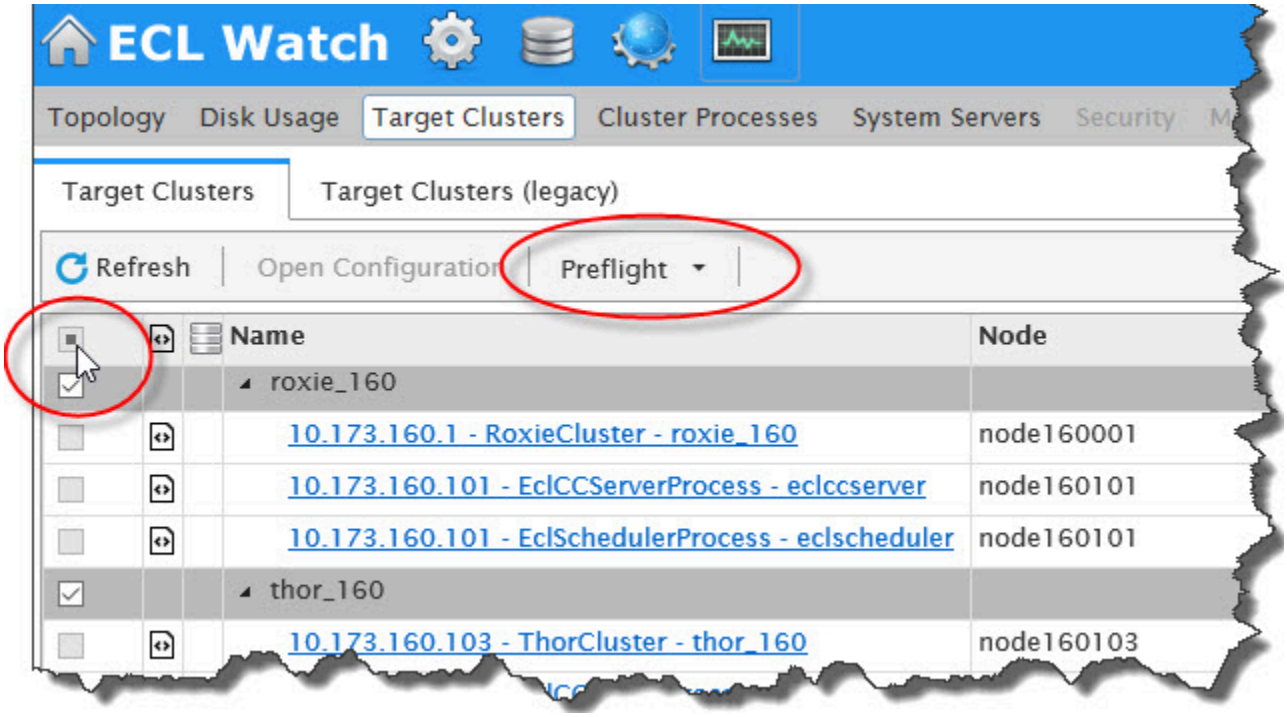
This displays a detailed listing of all your systems' Clusters.

2. Click on the select all check box, in the top row on the left side, to select all of the target clusters.

   Optionally, you can just check the box(es) next to only the cluster(s) you want to preflight. If you choose to preflight all Target Clusters, you do not need to preflight Thor and Roxie separately as detailed below.

   With the clusters selected, the preflight action button activates and you can press it to display the preflight options.

Figure 13. Select Target Clusters



3. Select or de-select any desired options, then press the **Submit** button at the bottom to start preflight.

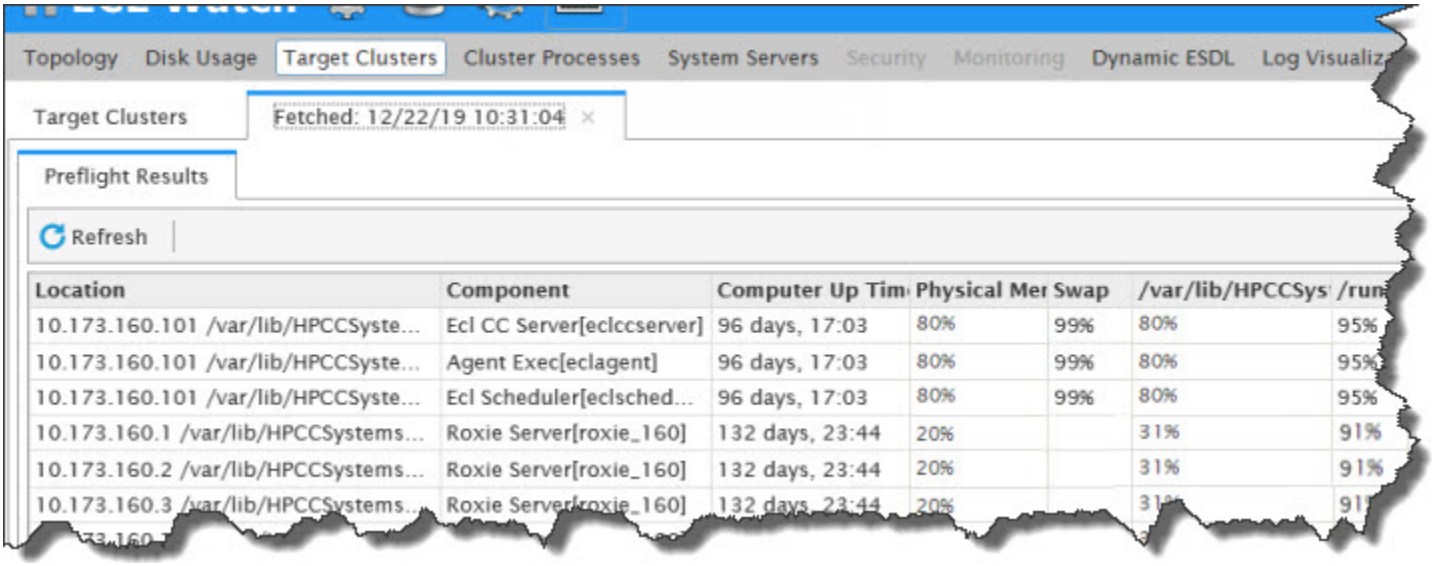
Figure 14. Submit

NOTE: Depending on the size of your system, there could be a slight delay in displaying the results.

EXPECTED RESULTS:

After pressing **Submit**, a screen similar to the following should display.

Figure 15. Target Cluster Information



This screen displays information on your system's component nodes. This information can help to indicate if everything is operating normally or can help to point out any potential concerns.

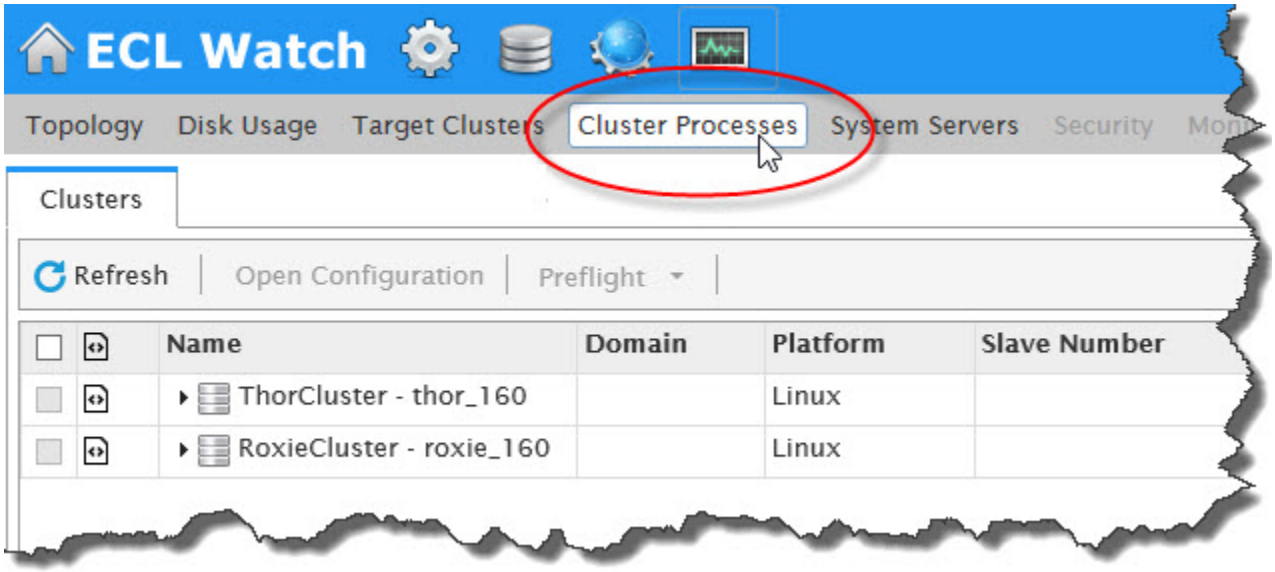
If there are any notable alerts, they are highlighted. These alerts usually require some attention.

If you have any alerts you should examine the specified component further. It is indicative of some kind of problem or abnormality.

Preflight Thor

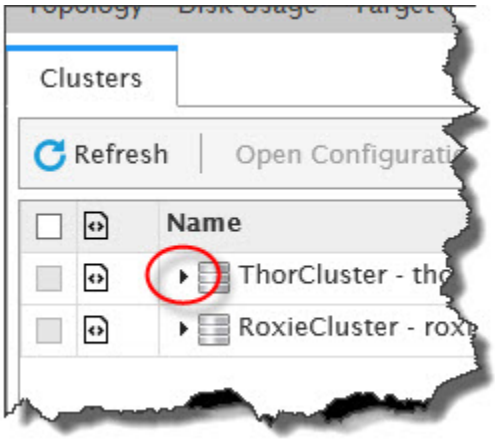
1. Click on the **Operations** icon then click on the **Cluster Processes** link.

Figure 16. Cluster Processes Link



2. Expand the Thor cluster by clicking on the arrow next to the **ThorCluster** link.

Figure 17. Thor Cluster link

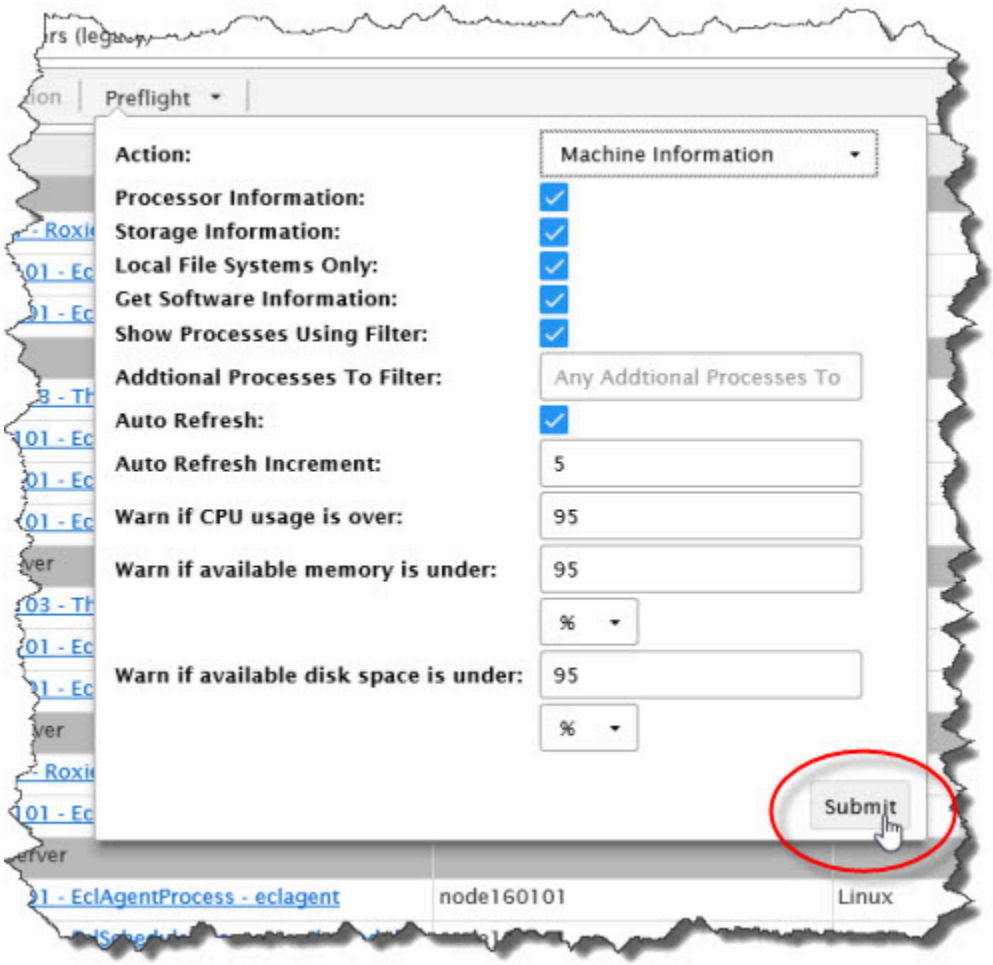


3. Check the box next to any individual nodes to examine or check the **Select All** checkbox in the first row.

4. With the systems selected, the preflight action button activates and you can press it to display the preflight options.

5. Select or de-select any desired options, then press the **Submit** button at the bottom to start preflight.

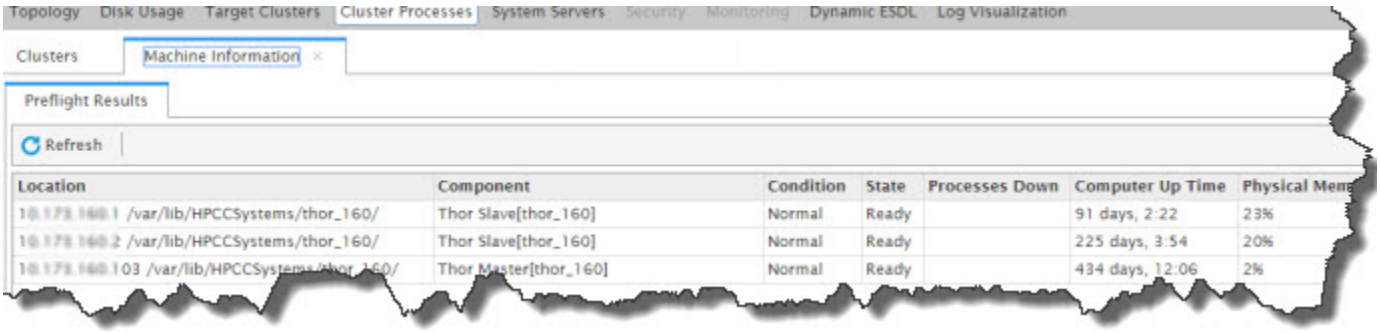
Figure 18. Submit



EXPECTED RESULTS:

After pressing Submit, a screen similar to the following displays.

Figure 19. Cluster Process results



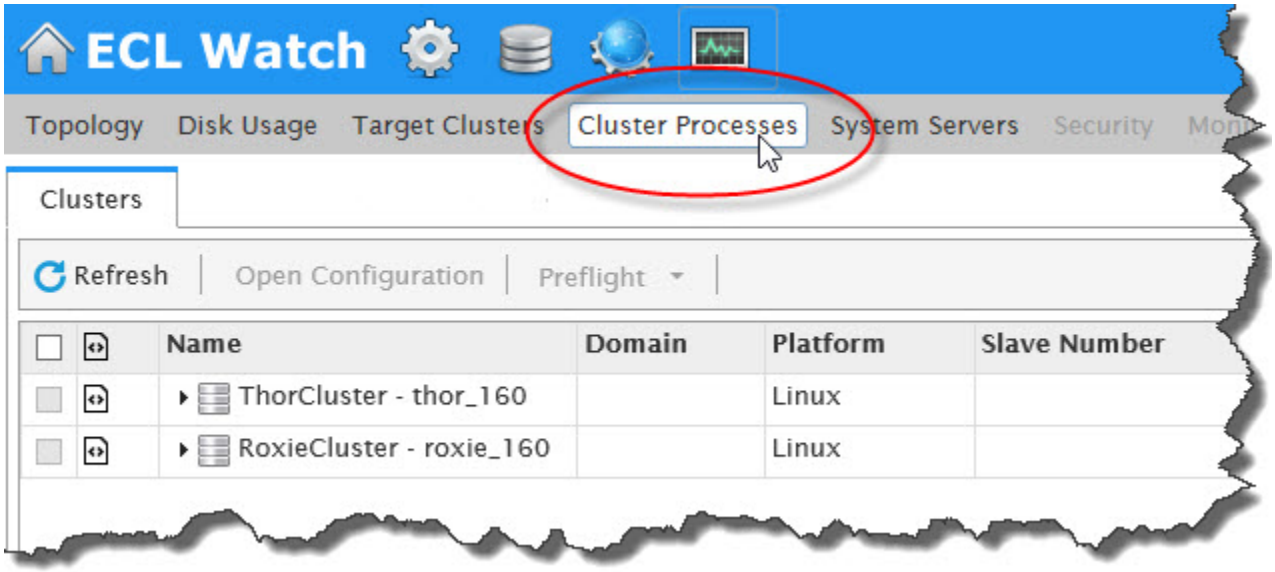
This displays information on your selected cluster(s). This information can help to indicate if everything is operating normally or can help to point out any potential concerns.

If there are any notable alerts, they are highlighted. The alerts usually require some additional attention.

Preflight the Roxie Cluster

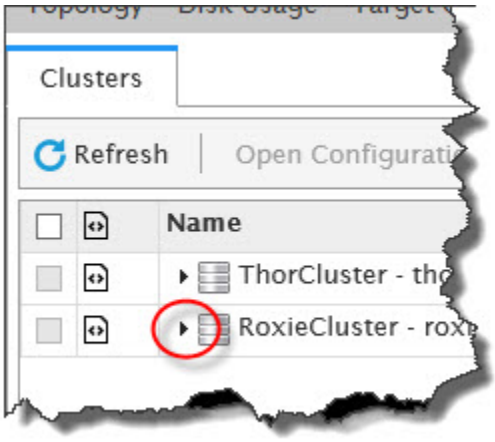
1. Click on the **Operations** icon then click on the **Cluster Processes** link.

Figure 20. Cluster Processes Link



2. Expand the Roxie cluster by clicking on the arrow next to the **RoxieCluster** link.

Figure 21. RoxieCluster link

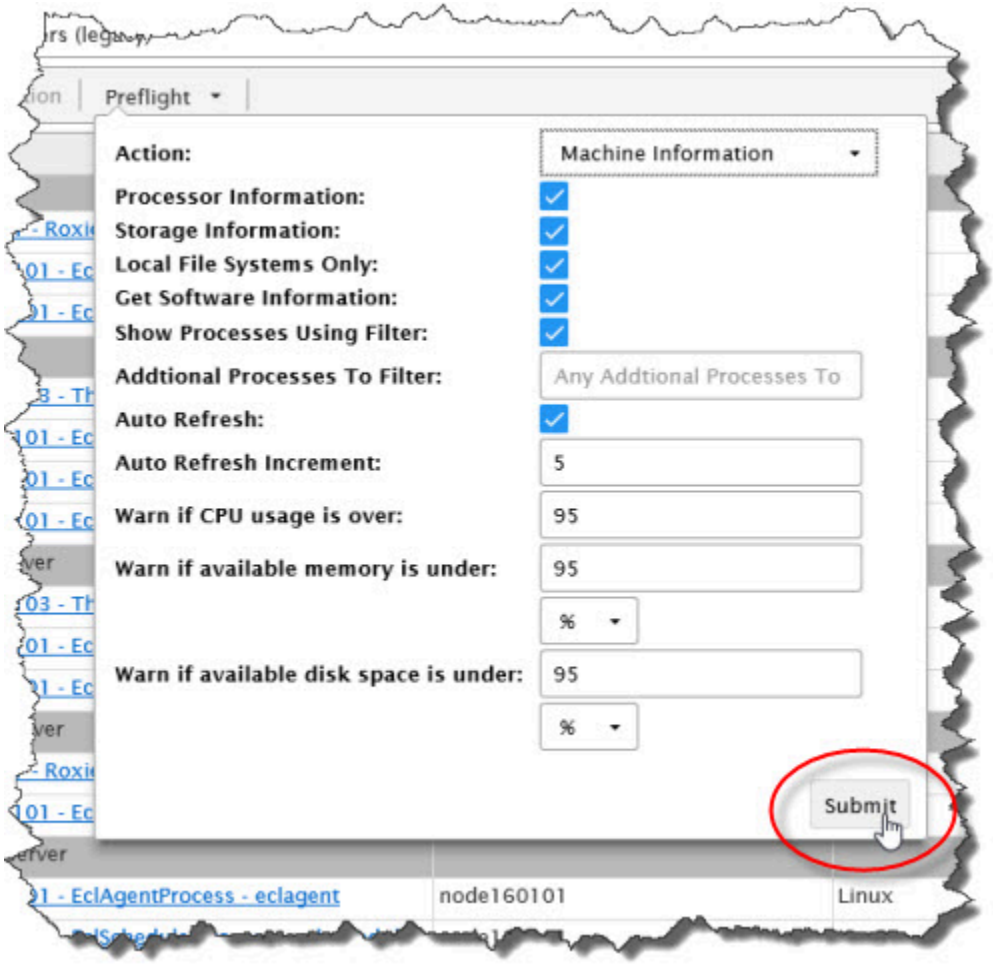


3. Check the box next to any individual nodes to examine or check the **Select All** checkbox in the first row.

4. With the systems selected, the preflight action button activates and you can press it to display the preflight options.

5. Select or de-select any desired options, then press the **Submit** button at the bottom to start preflight.

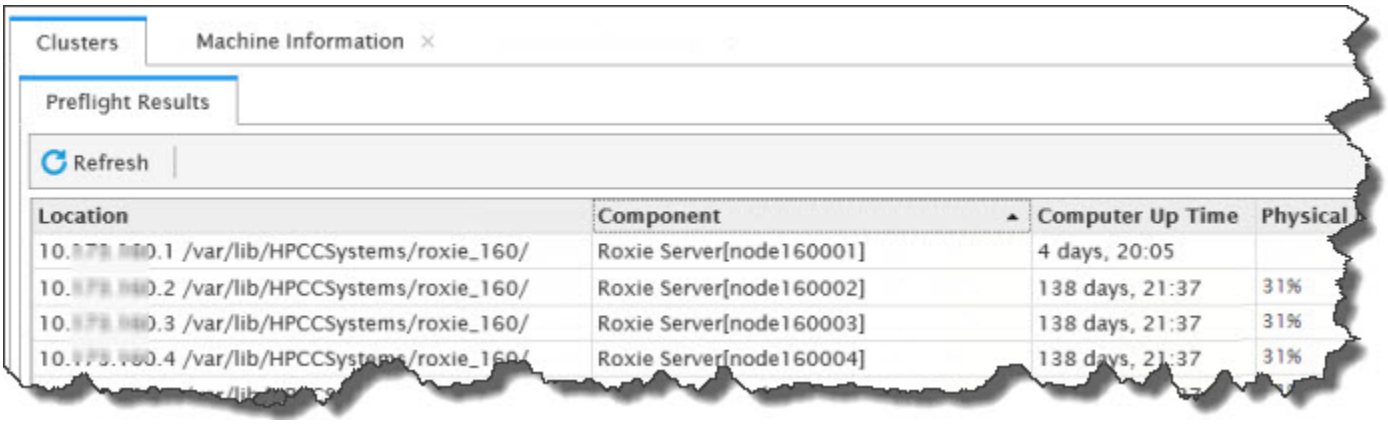
Figure 22. Submit



EXPECTED RESULTS

After pressing Submit, a screen similar to the following should display.

Figure 23. Roxie system information



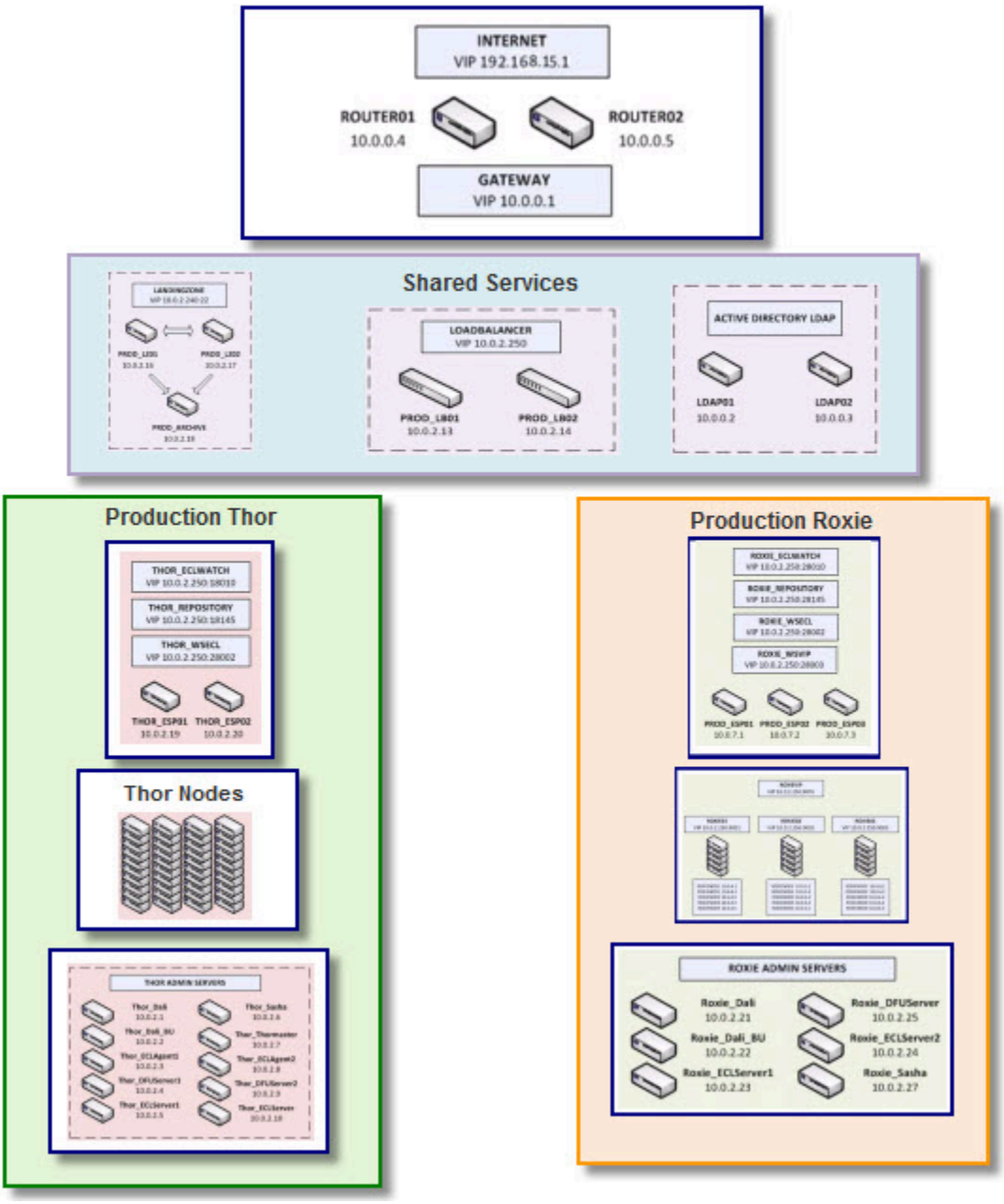
This indicates whether the Roxie nodes are running, and some additional information about them.

If there are any notable alerts, they are highlighted. The alerts usually require some additional attention.

System Configuration and Management

The HPCC Systems platform require configuration. The Configuration Manager tool (configmgr) included with the system software is a valuable piece of setting up your HPCC Systems platform. The Configuration Manager is a graphical tool provided that can be used to configure your system. Configuration Manager has a wizard that you can run which will easily generate an environment file to get you configured, up and running quickly. There is an advanced option available through Configuration Manager which allows for a more specific configuration, while still using the graphical interface. If desired you can edit the environment files using any xml or text editor however the file structure must remain valid.

Figure 24. Sample Production Configuration



Configuration Manager is the utility with which we configure the HPCC Systems® platform. The HPCC Systems platform's configuration is stored in an XML file named **environment.xml**. Once you generate an environment (xml) file, it gets saved into a source directory (default is **/etc/HPCCSystems/source**). You then need to stop the system to copy it into the active HPCC Systems directory, then distribute it into place on to each node and restart the HPCC Systems platform. At no time during configuration do you work on the live environment file.

When you install the HPCC Systems package, a default single-node environment.xml file is generated. After that, you can use the Configuration Manager to modify it and/or create a different environment file to configure components, or add nodes. There is a Configuration Manager wizard to help create an environment file.

Give any environment file you create a descriptive name that would indicate what it is for in the source. For example, you might create an environment without a Roxie, you could call that file *environmentNoRoxie.xml*.

You would then copy the new configuration file you generate from the source directory to the **/etc/HPCCSystems** directory. Rename the file to environment.xml, and restart the system in order to reconfigure your system.

Configuration Manager also offers an **Advanced View** which allows more granularity for you to add instances of components or change the default settings of components for more advanced users. Even if you plan to use the Advanced View, it is a good idea to start with a wizard generated configuration file and use Advanced View to edit it.

More information and specific details for each Configuration Manager component and attributes of those components is detailed in *Using Configuration Manager*.

Running the Configuration Manager

This section will guide you through configuring an HPCC Systems environment using the Configuration Manager.

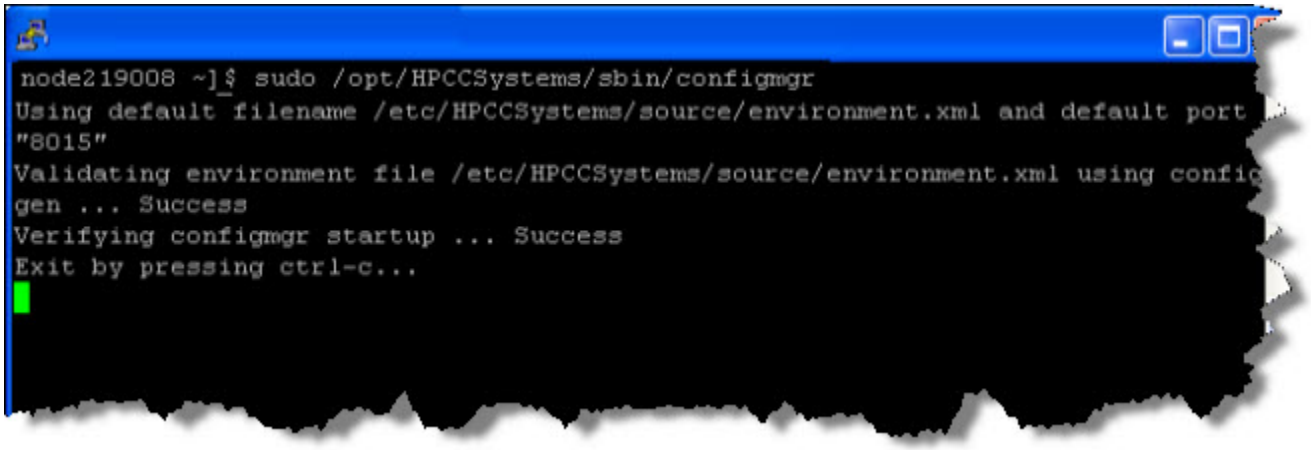
The HPCC Systems package should already be installed on ALL nodes.

You can use any tool or shell script you choose.

1. SSH to a node in your environment and login as a user with sudo privileges. We would suggest that it would be the first node, and that it is a support node, however that is up to your discretion.

2. Start the Configuration Manager service on the node (again we would suggest that it should be on a support node, and further that you use the same node to start the Configuration Manager every time, but this is also entirely up to you).

```
sudo /opt/HPCCSystems/sbin/configmgr
```



3. Using a Web browser, go to the Configuration Manager's interface:

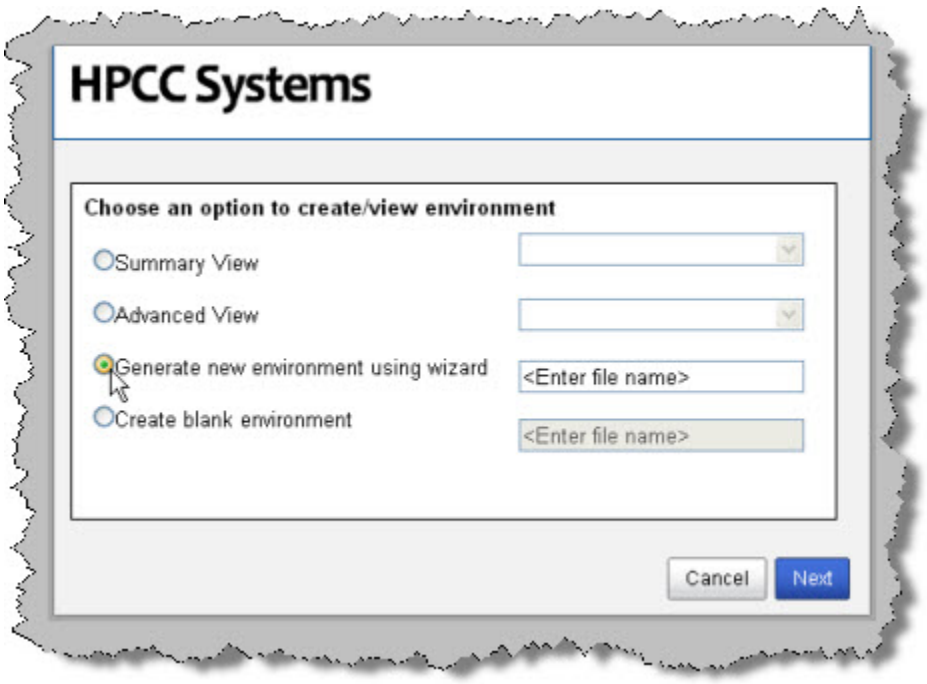
```
http://<ip of installed system>:8015
```

   The Configuration Manager startup wizard displays.

There are different ways to configure your HPCC Systems platform. You can use the **Generate environment wizard** and use that environment or experienced users can then use the **Advanced View** for more specific customization. There is also the option of using **Create blank environment** to generate an empty environment that you could then go in and add only the components you would want.

Environment Wizard

1. To use the wizard select the **Generate new environment using wizard** button.



2. Provide a name for the environment file.

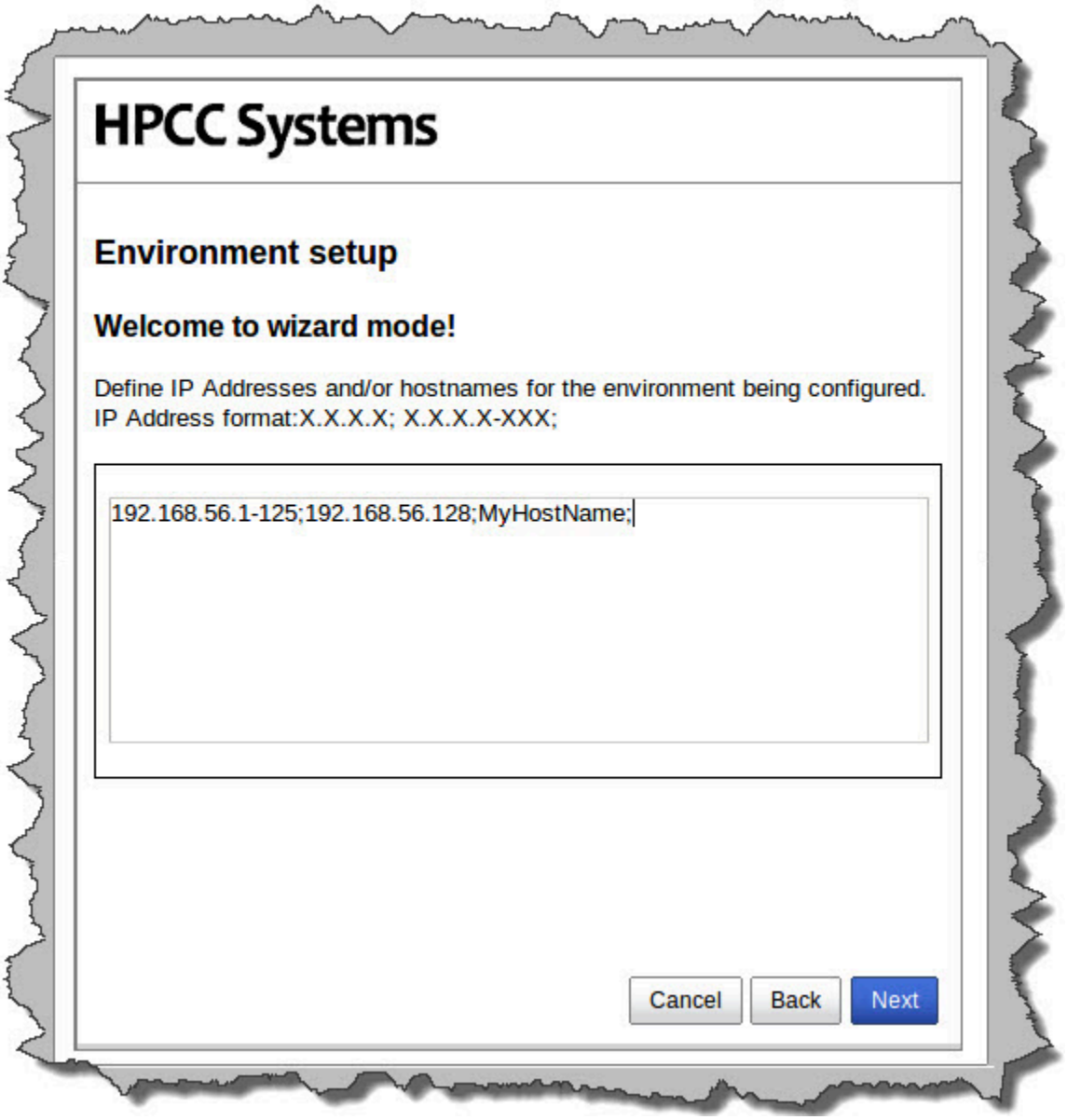
   This will then be the name of the configuration XML file. For example, we will name our environment *NewEnvironment* and this will produce a configuration XML file named *NewEnvironment.xml* that we will use.

3. Press the Next button.

   Next you will need to define the IP addresses that your HPCC Systems platform will be using.

4. Enter the IP addresses or hostname(s).

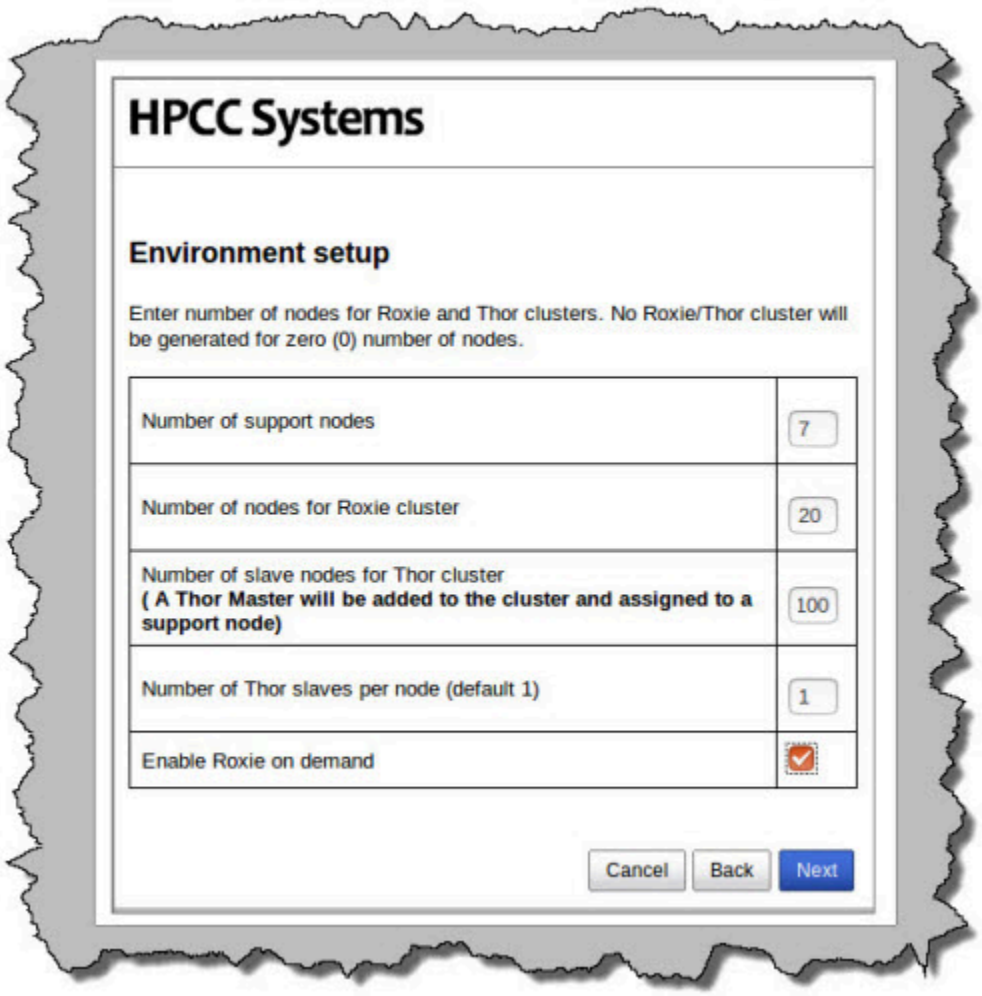
   IP Addresses can be specified individually using semi-colon delimiters. You can also specify a range of IPs using a hyphen (for example, nnn.nnn.nnn.x-y). In the image below, we specified the IP addresses 10.239.219.1 through 10.239.219.100 using the range syntax, and also a single IP 10.239.219.111.



5. Press the Next button.

   Now you will define how many nodes to use for the Roxie and Thor clusters.

6. Enter the appropriate values as indicated.



Number of support nodes: Specify the number of nodes to use for support components. The default is 1.

Number of nodes for Roxie cluster: Specify the number of nodes to use for your Roxie cluster. Enter zero (0) if you do not want a Roxie cluster.

Number of slave nodes for Thor cluster Specify the number of slave nodes to use in your Thor cluster. A Thor master node will be added automatically. Enter zero (0) if you do not want any Thor slaves.

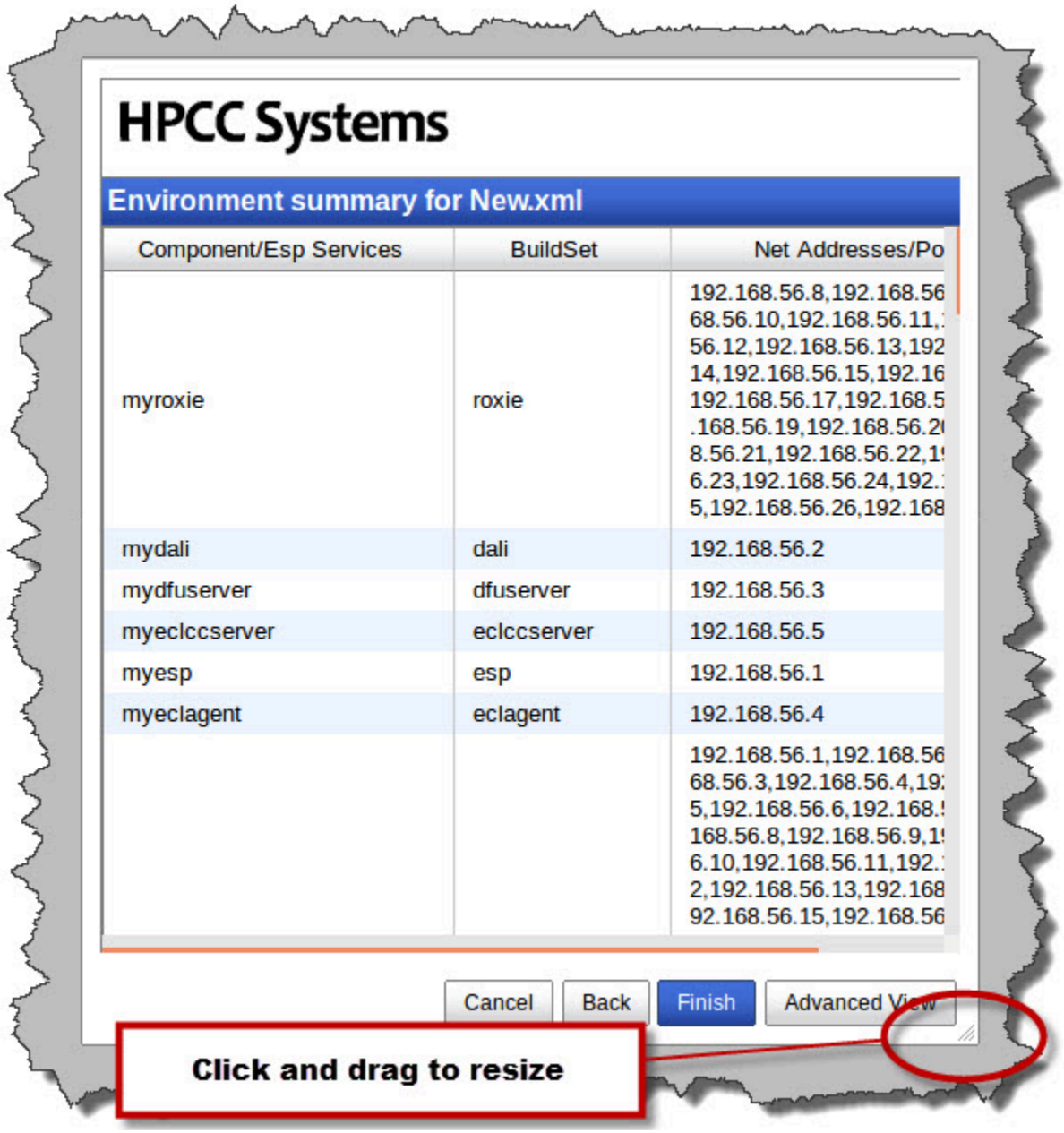
Number of Thor slaves per node (default 1) Specify the number of Thor slave processes to instantiate on each slave node. Enter zero (0) if you do not want a Thor cluster.

Enable Roxie on demand Specify whether or not to allow queries to be run immediately on Roxie. (Default is true)

7. Press the **Next** button

The wizard displays the configuration parameters.

8. Press the **Finish** button to accept these values or press the **Advanced View** button to edit in advanced mode.



You will now be notified that you have completed the wizard.

At this point, you have created a file named NewEnvironment.xml in the **/etc/HPCCSystems/source** directory

Keep in mind, that your HPCC Systems configuration may be different depending on your needs. For example, you may not need a Roxie or you may need several smaller Roxie clusters. In addition, in a production [Thor] system, you would ensure that Thor and Roxie nodes are dedicated and have no other processes running on them. This document is intended to show you how to use the configuration tools. Capacity planning and system design is covered in a training module.

Distribute the Configuration

1. Stop the HPCC Systems platform.

   If it is running stop the HPCC Systems platform (on every node), using a command such as this:

   ```
   sudo systemctl stop hpccsystems-platform.target
   ```

   Note: You may have a multi-node system and a custom script such as the one illustrated in Appendix of the **Installing and Running the HPCC Systems Platform** document to start and stop your system. If that is the case please use the appropriate command for stopping your system on every node.

   Be sure the HPCC Systems platform is stopped before attempting to copy the environment.xml file.

2. Back up the original environment.xml file.

   ```
   # For example
   sudo -u hpcc cp /etc/HPCCSystems/environment.xml /etc/HPCCSystems/source/environment-date.xml
   ```

   Note: The live environment.xml file is located in your **/etc/HPCCSystems/** directory. Configuration Manager works on files in **/etc/HPCCSystems/source** directory. You must copy from this location to make an environment.xml file active.

   You can also choose to give the environment file a more descriptive name, to help differentiate any differences.

   Having environment files under source control is a good way to archive your environment settings.

3. Copy the new .xml file from the source directory to the /etc/HPCCSystems and rename the file to *environment.xml*

   ```
   # for example
   sudo -u hpcc cp /etc/HPCCSystems/source/NewEnvironment.xml /etc/HPCCSystems/environment.xml
   ```

4. Copy the **/etc/HPCCSystems/environment.xml** to the **/etc/HPCCSystems/** on to *every* node.

   You may want to use a script to push out the XML file to all nodes. See the *Example Scripts* section in the Appendix of the **Installing and Running the HPCC Systems platform** document. You can use the scripts as a model to create your own script to copy the environment.xml file out to all your nodes.

5. Restart the HPCC Systems platform on all nodes.

Environment.conf

A component of the HPCC Systems platform on bare-metal configuration is the environment.conf file. Environment.conf contains some global definitions that the configuration manager uses to configure the HPCC Systems platform. In most cases, the defaults are sufficient.

The environment.conf file only works for bare-metal deployments. For container or cloud deployments the environment.conf is not valid, instead there are environment settings which can be configured by setting values in the Helm charts. See the Containerized HPCC Systems documentation for containerized or cloud deployments.

WARNING: These settings are essential to proper system operation. Only expert level HPCC Systems administrators should attempt to change any aspects of this file.

By default the environment.conf file is located:

```
/etc/HPCCSystems
```

Environment.conf is required upon startup of the HPCC Systems platform. The environment.conf is where the HPCC Systems environment file is defined.

```
/opt/HPCCSystems/environment.xml
```

This is also where the working path is defined.

```
path=/opt/HPCCSystems
```

The working path is used by several aspects of the application, changing this could cause needless complications. By default the application installs there, and sets many resources to that as well.

The default envrionment.conf:

```
## Default environment configuration file for OpenHPCC

[DEFAULT]
configs=${CONFIG_DIR}
path=${INSTALL_DIR}
classpath=${INSTALL_DIR}/classes
runtime=${RUNTIME_PATH}
lock=${LOCK_PATH}
# Supported logging fields: AUD,CLS,DET,MID,TIM,DAT,PID,TID,NOD,JOB,USE,SES,COD,MLT,MCT,NNT,COM,QUO,PFX,ALL,S
logfields=TIM+DAT+MLT+MID+PID+TID+COD+QUO+PFX+AUD
pid=${PID_PATH}
log=${LOG_PATH}
user=${RUNTIME_USER}
group=${RUNTIME_GROUP}
#umask=022
#nice=0
home=${HOME_DIR}
environment=${ENV_XML_FILE}
sourcedir=${CONFIG_SOURCE_PATH}
blockname=${DIR_NAME}
interface=*
# enable epoll method for notification events (true/false)
use_epoll=true
#epoll_hdlperthrd=10
```

```
# allow kernel pagecache flushing where enabled (true/false)
allow_pgcache_flush=true
# report UDP network stats
udp_stats=true
mpStart=7101
mpEnd=7500
mpSoMaxConn=128
mpTraceLevel=0
# enable SSL for dafilesrv remote file access (SSLNone/false | SSLOnly/true | SSLFirst | UnsecureFirst | Unse
# Enabling requires setting the HPCCPassPhrase, HPCCCertFile, and HPCCPrivateKeyFile values
#dfsUseSSL=SSLNone

#Specify location of HPCC PKI public/private key files
# note: if HPCCPassPhrase specified it must be encrypted
#HPCCPassPhrase=
#HPCCCertificateFile=${HOME_DIR}/${RUNTIME_USER}/certificate/certificate.pem
#HPCCPublicKeyFile=${HOME_DIR}/${RUNTIME_USER}/certificate/public.key.pem
#HPCCPrivateKeyFile=${HOME_DIR}/${RUNTIME_USER}/certificate/key.pem

jvmoptions=-XX:-UsePerfData
#Options to enable remote debugging of Java service or application
#jvmoptions=-XX:-UsePerfData -agentlib:jdwp=transport=dt_socket,server=y,suspend=n,address=2000
#JNI_PATH=/absolute/path/to/alternative/libjvm.so

# Python plugins can call python cleanup code on exit, but this seems to cause lockups in some Tensorflow exa
# In most cases, skipping the cleanup is harmless and avoids these lockups
skipPythonCleanup=true

# Multiple paths can be specified (separate with :, or ; on Windows).
# Relative paths are assumed to be relative to ${INSTALL_DIR}/versioned
additionalPlugins=python3

# To en-/disable Drop Zone restriction.
# Default is enabled (true).
useDropZoneRestriction=true
# If set, will force matching local file paths to become remote reads, e.g:
#forceRemotePattern=/var/lib/HPCCSystems/hpcc-data/eclagent/*

# Dafilesrv: default client side connection settings (NB: 0 = disable/use system defaults)
#dafsConnectTimeoutSeconds=100
#dafsConnectRetries=2
#dafsMaxReceiveTimeSeconds=0

# Dafilesrv: set to change number of seconds before retrying an unresponsive dafilesrv connection (default 10
# NB: for now this only applies to the last cached server
#dafsConnectFailRetrySeconds=10
```

The default environment.conf file includes several comments and explanations for many of the values defined in it.

Path considerations

Most of the directories are defined as absolute paths:

```
configs=/etc/HPCCSystems
path=/opt/HPCCSystems
classpath=/opt/HPCCSystems/classes
runtime=/var/lib/HPCCSystems
lock=/var/lock/HPCCSystems
```

The HPCC Systems platform will not run properly without the proper paths, and in some cases needs the absolute path. If a process or component can't find a path you will get an error message such as the following:

```
"There are no components configured to run on the node..."
```

If the path changes from HPCCSystems, it does NOT change in the environment.xml file. Any changes would require manually modifying the environment.xml file.

The log file, *hpcc-init.log* is written to the HPCCSystems path.

The logfields Setting

The **logfields** setting declares the fields to include in the component logs. You can customize which fields appear in your logs based on your business needs.

The syntax to use for logfields is to include the desired columns with a plus (+) sign, and use the minus (-) to specify any columns to exclude. For example, if you wanted to use the STD columns and exclude PFX, you could enter:

```
logfields=TIM+DAT+MLT+MID+PID+TID+COD+QUO
```

or

```
logfields=STD-PFX
```

The following table reflects all the available logging fields in the order in which they are written to the log file.

AUD	Audience: (Operator | User | Monitor | Performance | Internal | Programmer | Legacy | Audit)
CLS	Class: (Disaster | Error | Warning | Information | Progress | Legacy | Event | Unknown | All)
DET	Detail (unsigned int)
MID	Message ID (unsigned int)
TIM	Time: POSIX.2-1992 and by ISO C99 (%H:%M:%S)
DAT	Date: ISO 8601 format (%Y-%m-%d)
MCT	MicroTime: %02d:%02d:%02d.%06d
MLT	MilliTime: %02d:%02d:%02d.%03d
PID	Process ID (unsigned int)
TID	Thread ID (unsigned int)
SES	Session ID (unsigned int64)
NOD	Node (local endpoint url)
JOB	Job ID (unsigned int64)
USE	User ID (unsigned int64)
COM	Component (unsigned int)
QUO	Quote (message)
COD	Code (int)
PFX	Prefix: Error or Warning

The following are logfield macros which provide a bundled group of columns:

ALL	Include All available logfields

STD	Only Include standard logfields: TIM, DAT, MLT, MID, PID, TID, COD, QUO, PFX

Using nice

The HPCC Systems platform supports *nice*-based priorities using the nice Linux utility which invokes scripts and programs with specified priorities. The priority assigned to a process indicates to the CPU to provide more or less time than to other processes. A nice value of -20 is the highest priority, and a value of 19 is the lowest.

The default environment.conf file is delivered with the nice value disabled. If you wish to use nice to prioritize HPCC Systems platform processes, you need to modify the environment.conf file to enable nice. You can also adjust the nice value in environment.conf.

Other Environment.conf items

Some other items used by or referred to in environment.conf.

deploymentName Creates an environment variable in a bare-metal deployment that can be retrieved using the ECL built-in function--GETENV().

```
deploymentName: myenv1
```

Use_epoll It is an event mechanism to achieve better performance in more demanding applications where number of watched file descriptors is large.

Logfields Categories available to be logged. These consist of Time(TIM), Date(DAT), Process ID (PID), Thread ID (TID), etc.

Interface In the default environment.conf there is a value for interface. The default value for that is:

```
interface=*
```

The default value of * assigns the interface to an open ip address, in any order. Specifying an interface, such as Eth0, will assign the specified node as the primary.

Environment.conf and Java Settings

Specify the JNI_PATH setting if HPCC cannot find the adequate JNI library (libjvm) or if you want an alternate libjvm. For example:

```
JNI_PATH=/absolute/path/to/alternative/libjvm.so
```

When targeting custom classes, all dependencies must be accessible by declaring their location on the classpath, or including in the host jar file.

Custom java classes should be built using compatible Java versions. To check the Java version used to build a given class use:

```
javap: javap -verbose MyClass | grep "major"
```

When targeting a given method, ensure the signature declared in the embedded Java section matches the actual method signature. To get the signature, use:

```
javap: javap –s MyClass
```

Remote Access over TLS

Configuring your system for remote file access over Transport Layer Security (TLS) requires modifying the **dafilesrv** setting in the *environment.conf* file.

To do this either uncomment (if they are already there), or add the following lines to the *environment.conf* file. Then set the values as appropriate for your system.

```
#enable SSL for dafilesrv remote file access
#HPCCPassPhrase=true
HPCCCertFile=/certfilepath/certfile
HPCCPrivateKeyFile=/keyfilepath/keyfile
```

Set the dfsUseSSL=true and set the value for the paths to point to the certificate and key file paths on your system. Then deploy the *environment.conf* file (and cert/key files) to all nodes as appropriate.

Note: HPCCPassPhrase should be left commented out unless a passphrase was used to create the keys.

When dafilesrv is enabled for TLS (port 7600), it can still connect over a non-TLS connection (port 7100) to allow legacy clients to work.

Key file Additional Information

The Private and public keys need to be generated in PEM format. The same key file pairs should be installed across the cluster. These keys **must** be the same on every node in the cluster.

The HPCCCertFile and HPCCPublicKeyFile values must exist and be uncommented in environment.conf file as indicated above. The HPCCPassPhrase is only used when a passphrase was used in creation of the keys.

A good way to ensure the appropriate implementation of the secure key files is, as documented in the *Installing & Running the HPCC Systems Platform* book, to use the *install-cluster.sh* script.

Configuring HPCC Systems® for Authentication

This section details the steps to configure your HPCC Systems platform to use authentication. There are currently a few ways to use authentication with your HPCC Systems platform: simple htpasswd authentication, LDAP, or another plugin security method.

The htpasswd authentication method is basic password authentication. It only grants or denies access to a user, based upon MD5 encrypted password authentication.

LDAP authentication offers more features and options. LDAP can not only authenticate users, but adds granularity to the authentication. LDAP allows you to control grouped access to features, functions, and files.

You should consider your system needs and decide which of these methods is appropriate for your environment.

When implementing any form of authentication, we strongly recommend that you enable your ESP server to use HTTPS (SSL) and set ALL service bindings to only use HTTPS. This ensures that credentials are passed over the network using SSL encryption. See *Configuring ESP Server to use HTTPS (SSL)* **for details.**

You should not attempt this until you have already deployed, configured, and certified the environment you will use.

Using htpasswd authentication

htpasswd provides basic password authentication to the entire system. This section contains the information to install and implement htpasswd authentication.

Connect to Configuration Manager

In order to change the configuration for HPCC Systems components, connect to the Configuration Manager.

1. Stop all HPCC Systems components, if they are running.
2. Verify that they are stopped. You can use a single command, such as :

   ```
   sudo /opt/HPCCSystems/sbin/hpcc-run.sh status
   ```

3. Start Configuration Manager.

   ```
   sudo /opt/HPCCSystems/sbin/configmgr
   ```

4. Connect your web browser to the Configuration Manager web interface.

   (using the url of http://<*configmgr_IP_Address*>:8015, where <*configmgr_IP_Address*> is the IP address of the node running Configuration Manager)

5. Select the **Advanced View** radio button.
6. Use the drop list to select the XML configuration file.

   Note: Configuration Manager **never** works on the active configuration file. After you finish editing you will have to copy the environment.xml to the active location and push it out to all nodes.

7. Check the **Write Access** box.

Default access is read-only. Many options are only available when write-access is enabled.

Enabling htpasswd authentication in HPCC Systems

8. Create an instance of the **Security Manager** Plugin:

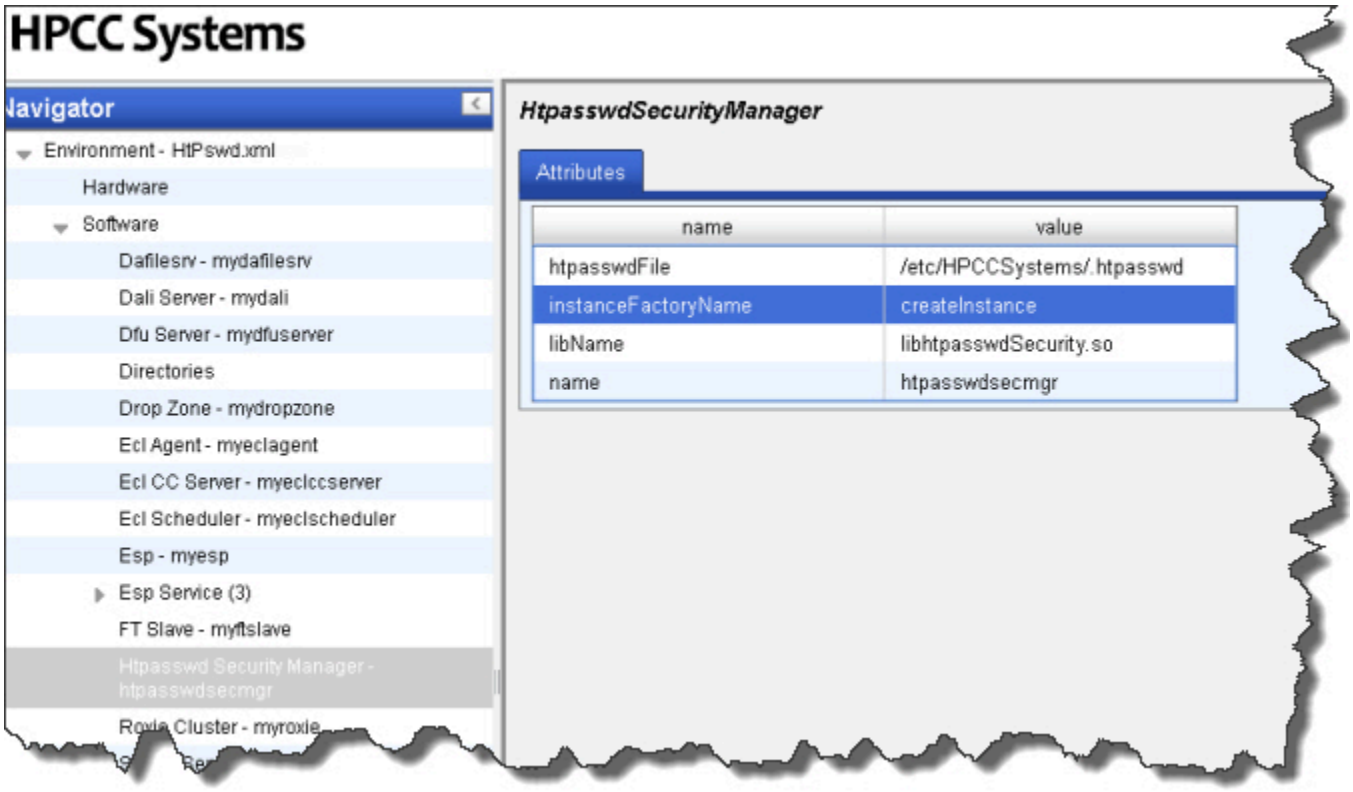
   a. Right-click on Navigator Pane on the left side.

   b. Select **New Components**

   c. Select the **htpasswdsecmgr** component

9. Configure the htpasswd plugin

Figure 25. Security Mgr Configuration page



   a. Enter the location of the Htpasswd file containing the username and password on the Linux file system for the value of **htpasswdFile**

   b. **InstanceFactoryName** is the name of the security manager factory function, implemented in the security library. The default is "createInstance". For implementing Htpasswd, leave the default.

   c. Provide a library name value for **libName**. For Htpasswd, use libhtpasswdSecurity.so

   d. Provide an instance **name** for the name value. For example, htpasswdsecmgr.

10. Select **Esp - myesp** in the Navigator panel on the left hand side.

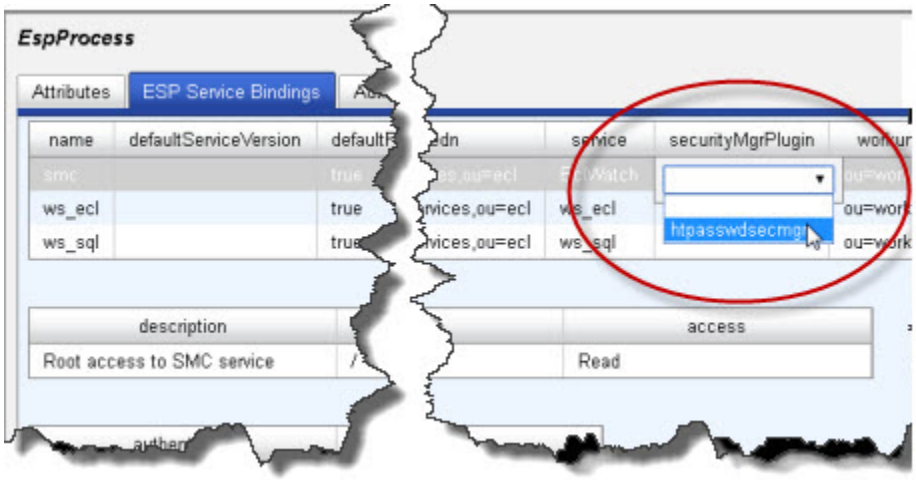
Note: If you have more than one ESP Server, each one should have its own authentication set up.

11 Associate the Security Manager Plugin with the ESP binding(s)

a. Click on the target **Esp** in the Navigator Pane on the left side.

b. Select the **ESP Service bindings** tab

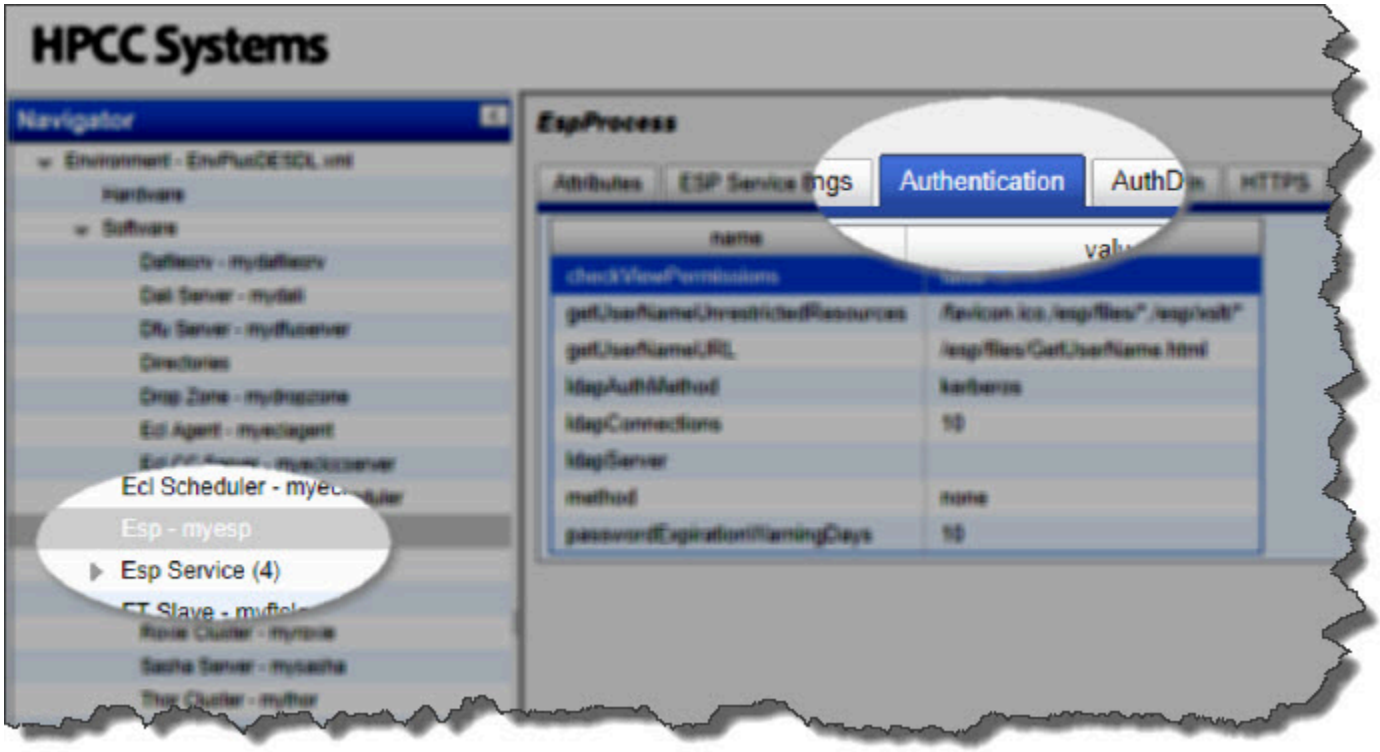
c. On the target binding(s) select the appropriate securityMgrPlugin instance from the drop list.



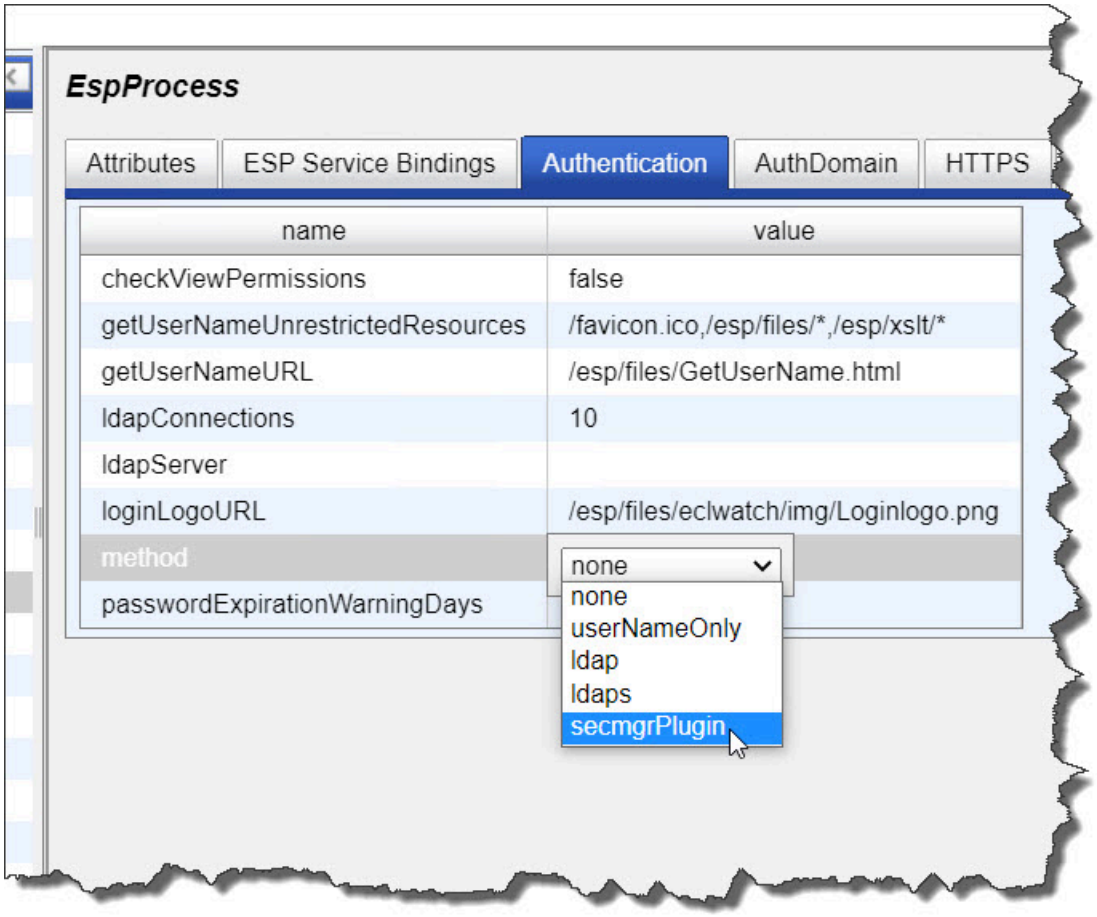
12 Select the security Plugin for each service that requires a security manager.

For example, in the above image, select htpasswdsecmgr for the smc service. Then, select it for ws_ecl and every other service that you want to use htpassword security.

13 Select the **Authentication** tab



14. Click on the value column drop list to display the choices for **method**.



15. Choose **secmgrPlugin** from the drop list.

16. Click on the disk icon to save.

User administration with htpasswd

Users and passwords are kept in the htpasswd file. The htpasswd file must exist on the ESP Node where you have enabled authentication. HPCC Systems only recognizes MD5 encrypted passwords.

The default location is: **/etc/HPCCSystems/.htpasswd** on the ESP node that has been configured to authenticate, but it is configurable from the Htpasswd Security Manager as outlined above (step 9).

You can use the htpasswd utility to create the .htpasswd file to administer users.

You may already have the htpasswd utility on your system, as it is a part of some Linux distributions. Check your Linux distribution to see if you already have it. If you do not have it you should download the utility for your distribution from The Apache Software Foundation.

For more information about using htpasswd see: http://httpd.apache.org/docs/2.2/programs/htpasswd.html.

Single User Security Manager

The Single User security manager is a specialized security manager that allows a username/password combination to be specified on the ESP startup command line. At runtime, when you attempt to access any authenticating ESP feature, such as ECL Watch, you must specify a username/password combination.

A single user security manager could be useful for a custom deployment where you do not want to configure an entire LDAP server or create a Linux HTPASSWD file, such as a classroom environment or a custom HPCC Systems Virtual Machine.

See the *Security Manager Plugin Framework* document for more information on configuring and deploying Security Manager plugins.

Using LDAP Authentication

This section contains the information to install and implement LDAP based authentication. LDAP Authentication provides the most options for securing your system, or parts of your system. In addition to these configuration settings you should run the **initldap** utility to create the required default HPCC Systems Admin user on your LDAP server.

If you choose to use LDAP authentication you must enable LDAP security in your HPCC Systems configuration. With LDAP security enabled on your system you can then choose to enable file scope security. You can choose to use LDAP authentication without enabling file scope security. The following sections describe how to enable LDAP authentication and file scope security for your HPCC Systems platform.

Connect to Configuration Manager

In order to change the configuration for HPCC Systems components, connect to the Configuration Manager.

1. Stop all HPCC Systems components, if they are running.

2. Verify that they are stopped. You can use a single command, such as :

   ```
   sudo /opt/HPCCSystems/sbin/hpcc-run.sh -a hpcc-init status
   ```

3. Start Configuration Manager.

   ```
   sudo /opt/HPCCSystems/sbin/configmgr
   ```

4. Connect to the Configuration Manager web interface.

   (using the url of http://<*configmgr_IP_Address*>:8015, where <*configmgr_IP_Address*> is the IP address of the node running Configuration Manager)

5. Select the **Advanced View** radio button.

6. Use the drop list to select the XML configuration file.

Note: Configuration Manager **never** works on the active configuration file. After you finish editing you will have to copy the environment.xml to the active location and push it out to all nodes.

Modifying the configuration

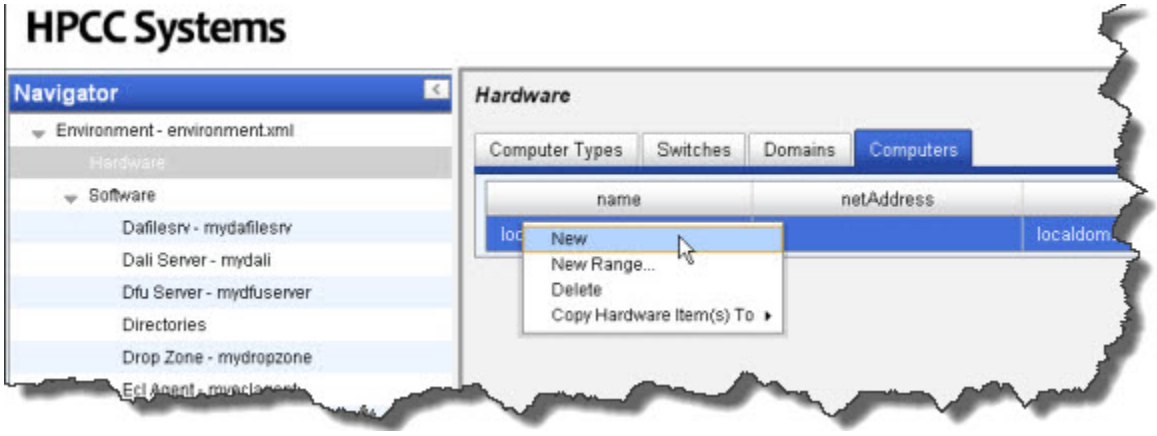
Follow the steps below to modify your configuration.

1. Check the box for **Write Access.**

2. From the **Navigator** pane, select **Hardware**.

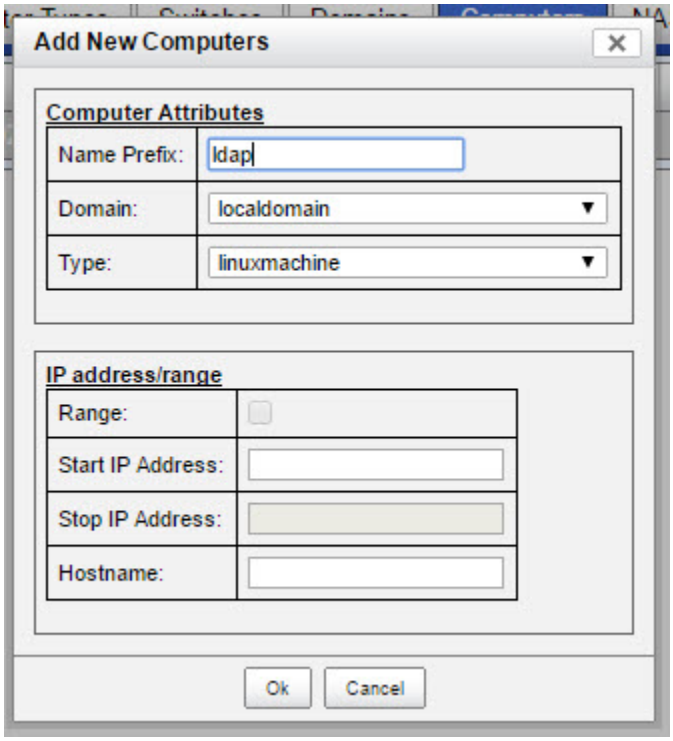
3. Select the **Computers** tab from the panel on the right.

4. Right-click on the table below computers and select **New** from the pop up menu.



The **Add New Computers** dialog displays.

5. Fill in the values for the **Computer Attributes**



a. Provide a **Name Prefix**, for example: ldap.

This helps you to identify it in the list of computers.

b. Fill in **Domain** and **Type** with the values of your domain name, as well as the types of machines you are using.

In the example above, **Domain** is localdomain, and the **Type** is linuxmachine. These should correspond to your domain and type.

If you need to add a new domain or machine type to your system to be able to define an existing LDAP server, you should set these up first in the other two tabs in the hardware section.

c. Add the IP address as appropriate for the LDAP server.

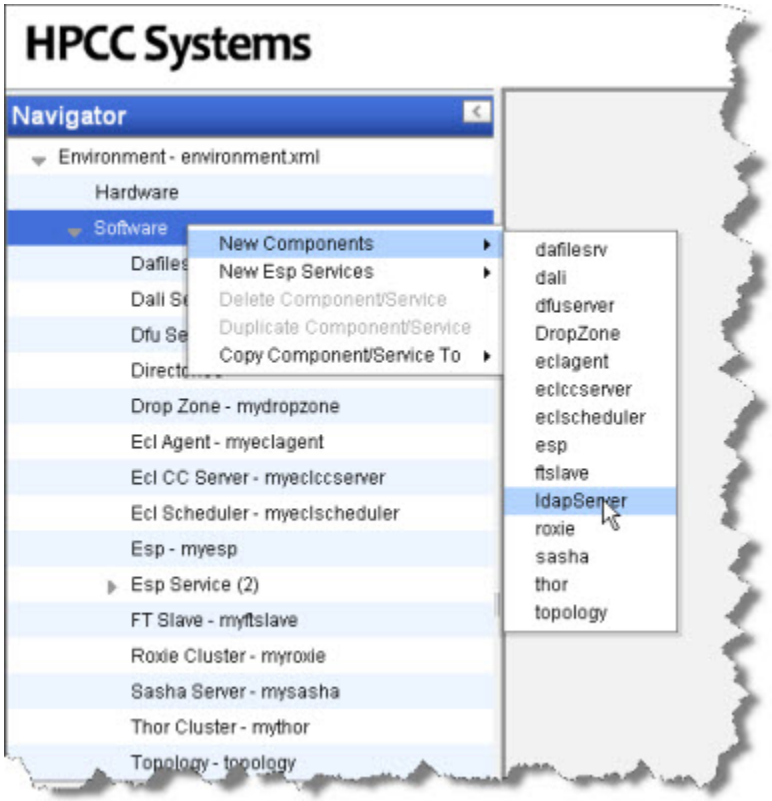
d. Press the **Ok** button.

e. Click on the disk icon to save.

Adding the ldapServer component

After the LDAP Server node has been added to the Hardware configuration, configure the Software LDAP server definition.

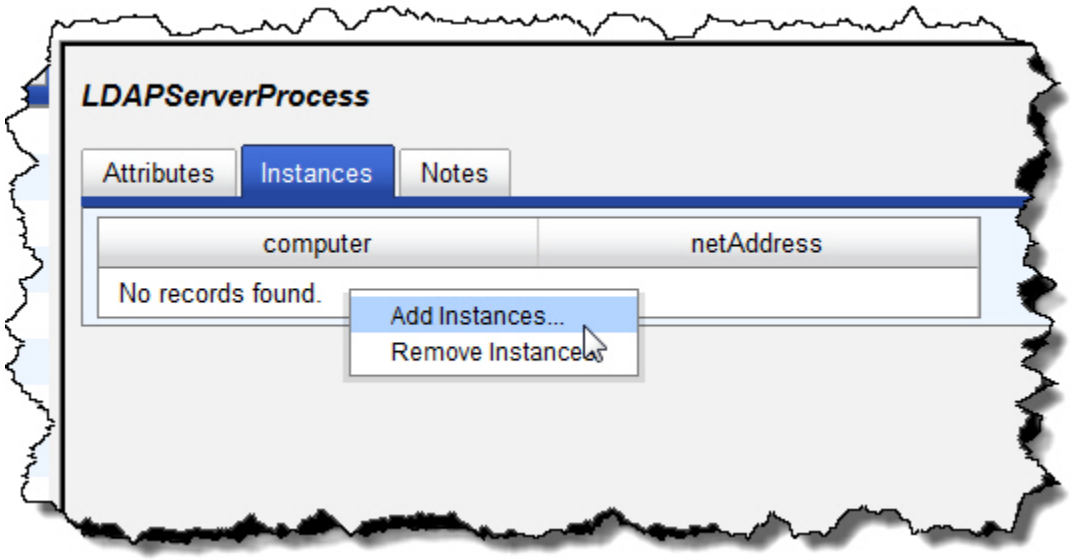
1. Right-click on **Navigator** Pane and choose **New Components** from the pop-up menu, then choose **ldapServer** from the pop-up menu.



	Note: The ldapServer component is merely a definition that specifies an existing LDAP server. It does not install one.

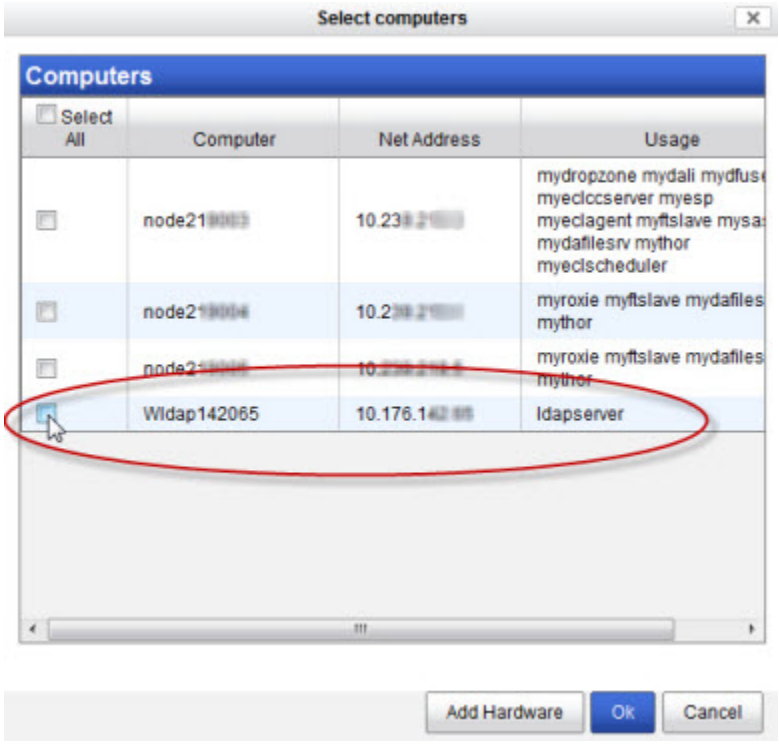
2. Fill in the **LDAP Server Process** properties:

   a. On the **Instances** tab, Right-click on the table on the right hand side, choose **Add Instances...**



   The **Select computers** dialog appears.

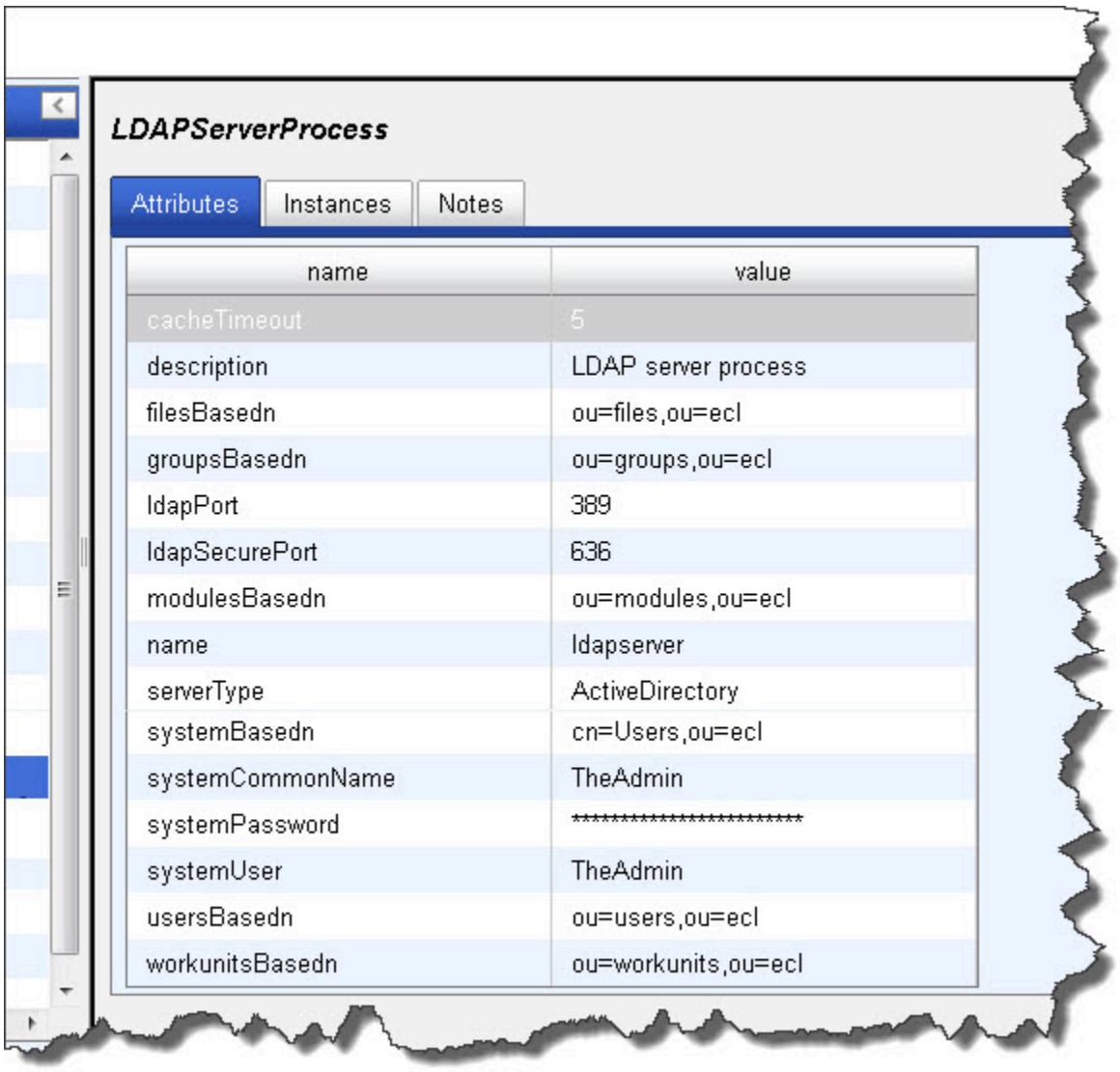
   b. Select the computer to use by checking the box next to it.



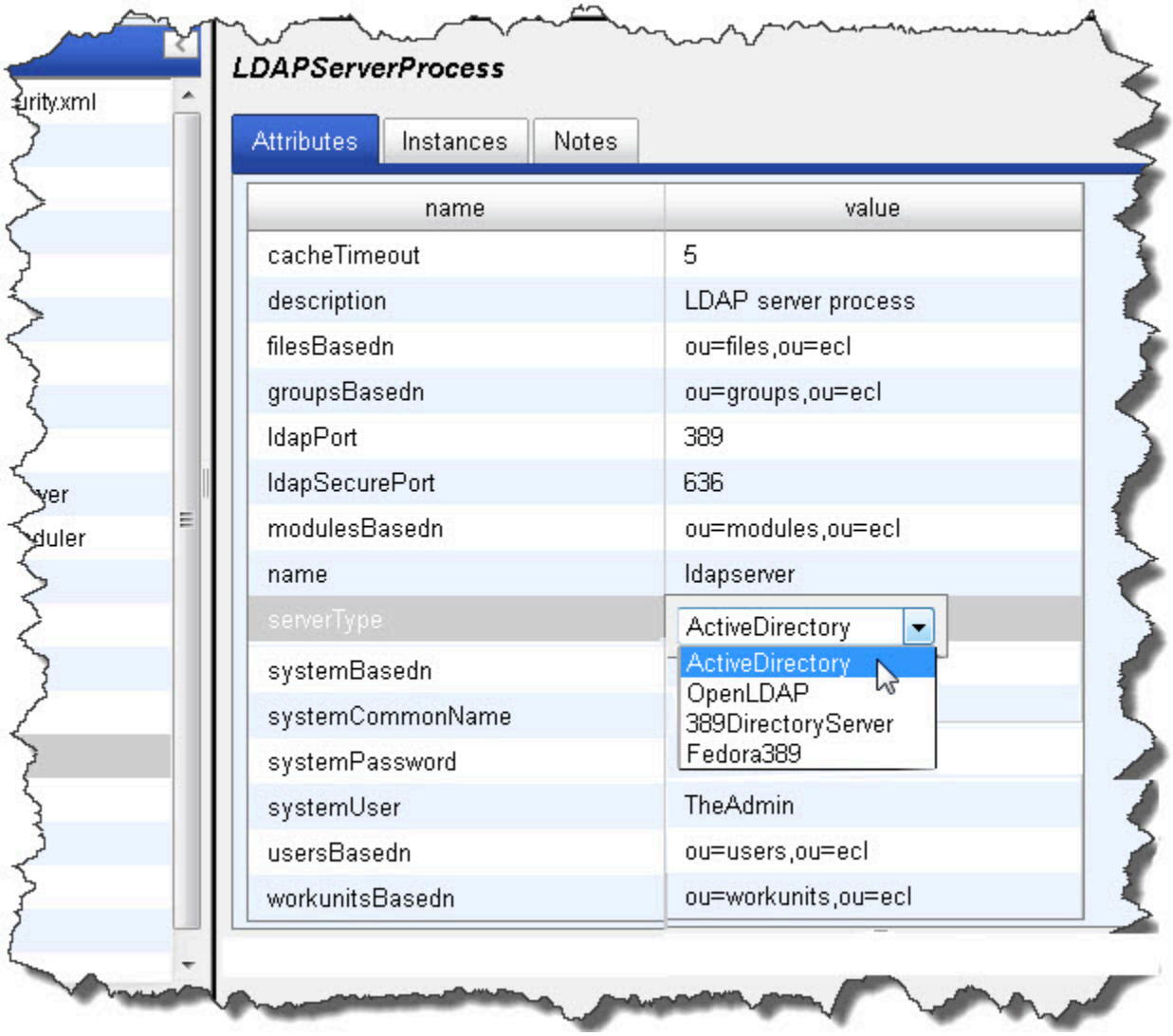
   This is the computer you added in the **Hardware** / **Add New Computers** portion earlier.

   c. Press the **Ok** button.

d. Fill in the **Attributes** tab with the appropriate settings from your existing LDAP Server.



e. Choose the LDAP server type from the serverType attribute drop box.



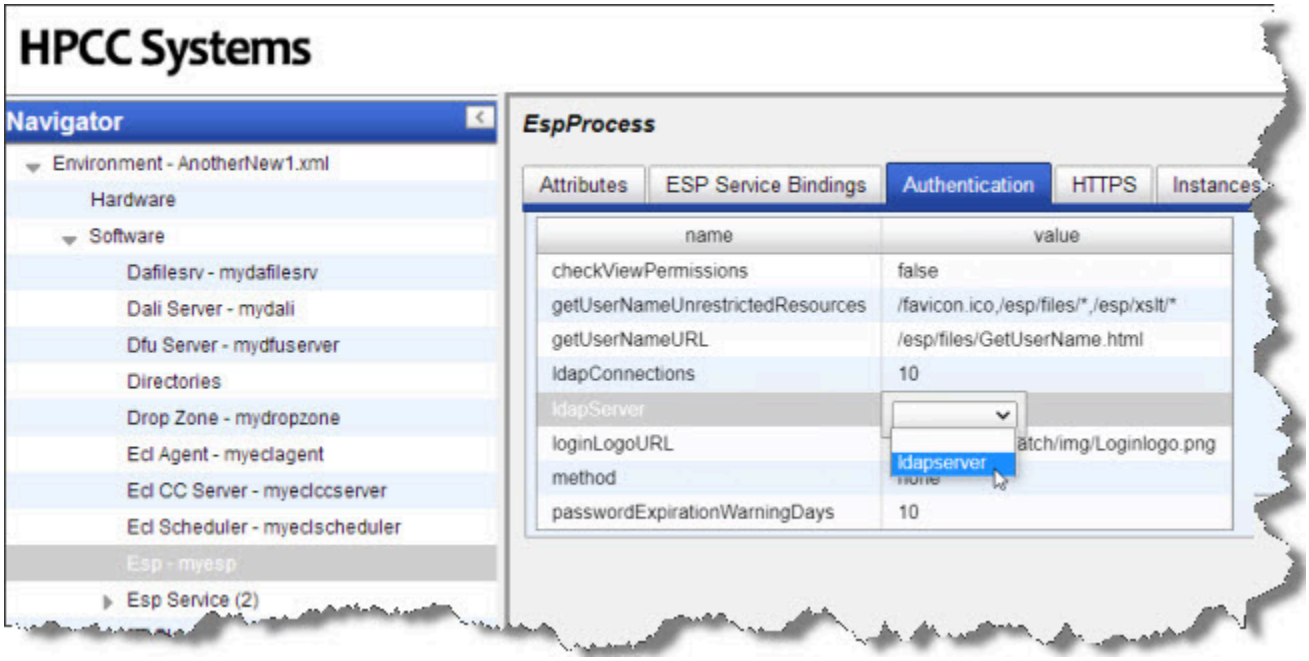
NOTE: Support for OpenLDAP has been deprecated. The option is included only for legacy purposes.

f. Click on the disk icon to save.

Note: The **cacheTimeout** value is the number of minutes that permissions are cached in ESP. If you change any permissions in LDAP, the new settings will not take effect until ESP and Dali refresh the permissions. This could take as long as the cacheTimeout. Setting this to 0 means no cache, but this has performance overhead so it should not be used in production.

3. In the Navigator pane, click on **ESP -- myesp**

4. On the **EspProcess** page on the right hand side, select the **Authentication** tab.



Fill in the appropriate values:

a. Change the **ldapConnections** to the number appropriate for your system (10 is for example only, may not be necessary in your environment).

b. Select the **ldapServer** component that you added earlier from the drop list, for example: ldapserver.

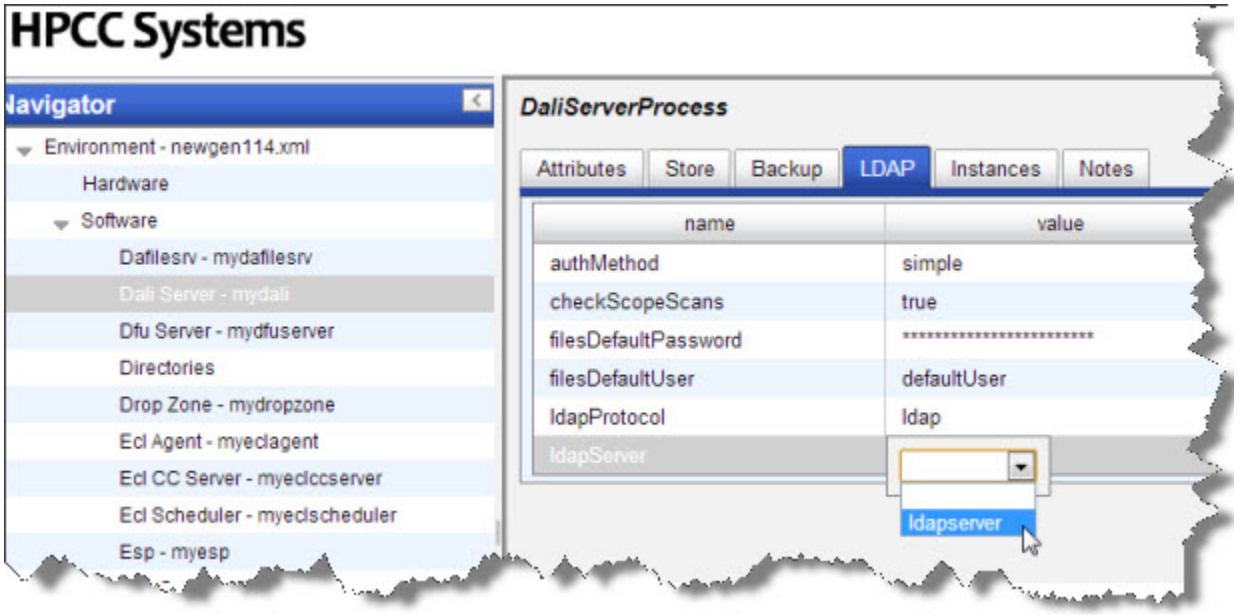
c. Change the **method** value to ldap.

d. Select the ESP Service bindings tab. Verify that your LDAP settings appear in the **resourcesBasedn** and **workunitsBasedn**

e. Click on the disk icon to save.

5. To enable the file scope permissions, configure the file scope security for the Dali Server.

   In the Navigator pane, click on the **Dali Server -- mydali**



   Fill in the values as appropriate:

   a. Select the **LDAP** tab.

   b. Change the **authMethod** to simple

   c. Set the **checkScopeScans** value to *true*.

      Only set this value to true when you want file scope security enabled. Security settings can have three states.

      - None, no authentication and no file scope security.
      - LDAP security for authentication only, without enabling file scope security.
      - LDAP authentication and file scope security enabled.

   d. Change the LDAP values as appropriate to match the settings in your LDAP server component in configuration manager.

      For example, change the **ldapServer** to the value you gave your LDAP Server, in our example it is: *ldapserver.*

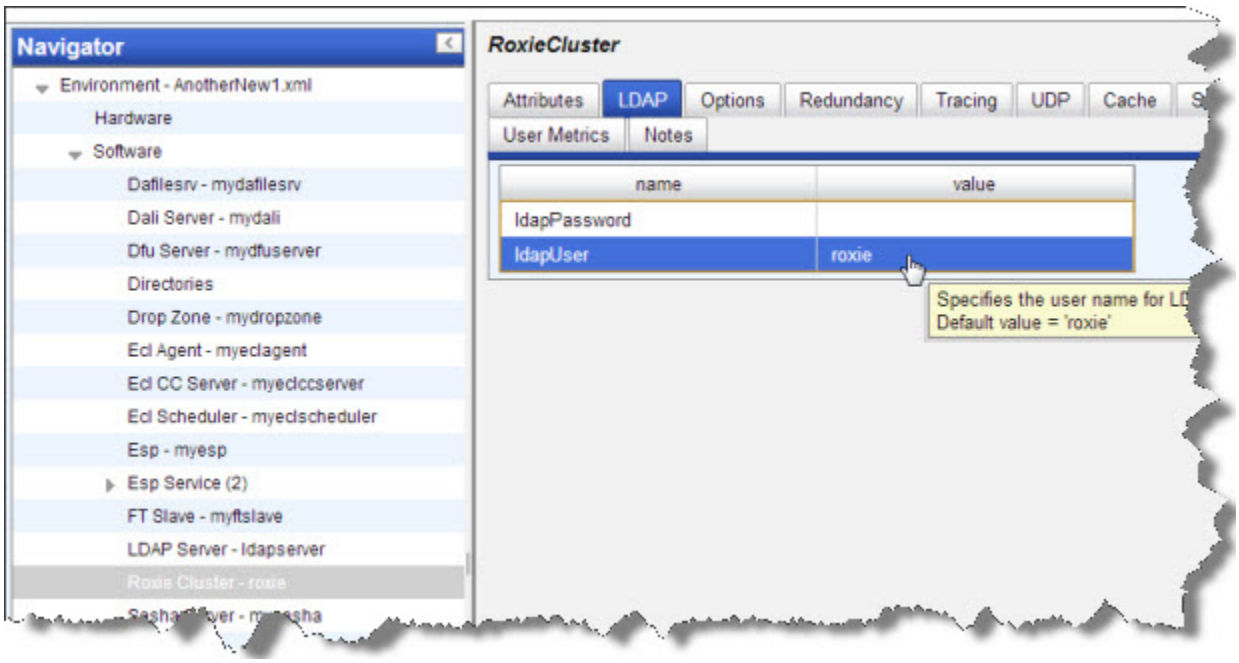
      Confirm the change when prompted.

      The **filesDefaultUser** is an LDAP account used to access files when no user credentials are supplied. This is similar to a guest account, so it should be an account with **very** limited access, if used at all. To disable access without credentials, leave **filesDefaultUser** blank.

      The **filesDefaultPassword** is the password for that account.

   e. Click on the disk icon to save.

6. In the Navigator pane, click on the **Roxie Cluster -- myroxie**



a. On the **RoxieCluster** page on the right hand side, select the **LDAP** tab.

b. Locate the **ldapUser** field and verify that there is a valid HPCC Systems user who is a member of the Authenticated Users group on your LDAP server. For example, the "*roxie*" user assumes that the "*roxie*" user is a valid HPCC Systems authenticated user.

c. Add the password security for Roxie by adding it to the **ldapPassword** field on the same tab.

	In order to run Roxie queries with File Scope security, ensure that a Roxie user is created in the list of authenticated users. In the following section, *Adding and editing users*, add the *roxie* user and make sure that password is the same as the one entered in Configuration Manager.

Installing the Default Admin user

After enabling your configuration for LDAP security, you must copy your environment file to the /etc/HPCCSystems directory. See the section *Configuring a Multi-Node System* for more info about configuring your system. With the correct environment.xml file in place, you must then run the **initldap** utility that initializes the security components and the default users.

The initldap Utility

The initldap utility creates the HPCC Systems Administrator's user account and the HPCC Systems OUs for a newly defined LDAP server. The initldap utility extracts these settings from the LDAPServer component(s) in the environment.xml bound to the configured ESPs.

You run the **initldap** utility once you complete your configuration with LDAP components enabled and have distributed your environment.xml file to all nodes.

```
sudo /opt/HPCCSystems/bin/initldap
```

The **initldap** utility prompts you for LDAP Administrator credentials. Enter the appropriate values when prompted.

The following example of initldap for a 389DirectoryServer deployment.

```
Enter the '389DirectoryServer' LDAP Admin User name on '10.123.456.78'...Directory Manager
Enter the LDAP Admin user 'Directory Manager' password...********

Ready to initialize HPCC Systems LDAP Environment, using the following settings
       LDAP Server     : 10.123.456.78
       LDAP Type       : 389DirectoryServer
       HPCC Admin User : HPCCAdmin389
Proceed?  y/n
```

Configuring ESP Server to use HTTPS (SSL)

The HPCC Systems Enterprise Services Platform server (ESP) supports Secure Sockets Layer (SSL), a protocol used to send and receive private data or documents.

SSL works by using a private key to encrypt and decrypt data transferred over the SSL connection. By convention, URLs using an SSL connection start with HTTPS instead of HTTP.

The SSL option in the ESP Server allows secure and encrypted communication between a browser or SOAP client application and the HPCC Systems platform.

SSL capabilities are configured in the Configuration Manager, but require a certificate be installed on the ESP server. The OpenSSL libraries provide a means to create the necessary certificate files in one of two ways.

- You can use the OpenSSL libraries to create a private key and a Certificate Signing Request (CSR) to purchase a certificate from a Certificate Issuing Authority (such as, VeriSign).
- You can use that CSR to generate your own self-signed certificate and then install the certificate and private key to your ESP Server.

In either case, once installed and configured, the network traffic is encrypted and secure. The Public and Private Keys use 2048-bit RSA encryption.

These server keys are read at runtime by the ESP process. It is important the installed keys have correct ownership and permissions. Typically, it is the HPCC user and their public key (certificate.cer) with read permissions such as 0444 (or 0644), along with the private key (privatekey.cer) with more restrictive permissions of 0400 (or 0600).

Generate an RSA Private Key

Use the OpenSSL toolkit to generate an RSA Private Key and a Certificate Signing Request (CSR). This can also be the basis for a self-signed certificate. Self-signed certificates are useful for internal use or testing.

The following example, creates a 2048-bit RSA Private Key which is encrypted using Triple-DES encryption and stored in Privacy Enhanced Mail (PEM) format.

```
openssl genrsa -des3 -out server.key 2048
```

When prompted, provide a passphrase. This is used as the basis for the encryption.

Remember this passphrase as you will need to enter it into the Configuration Manager later.

Generate a CSR (Certificate Signing Request)

After you have a private key, you can use it to create a Certificate Signing Request (CSR). You can use your CSR to request a signed certificate from a Certificate Authority (such as Verisign or Network Solutions). You can also use the CSR to create a self-signed certificate.

```
openssl req -new -key server.key -out server.csr
```

Answer the questions when prompted:

Country Name (2 letter code):	
State or Province Name (full name):	
Locality Name (eg, city) :	
Organization Name (eg, company) :	
Organizational Unit Name (eg, section) :	
Common Name (e.g., server's hostname):	
Email Address :	
A challenge password (optional):	
An optional company name (optional):	

Generate a Self-Signed Certificate

To generate a temporary certificate, which is good for up to 365 days, issue the following command:

```
openssl x509 -req -days 365 -in server.csr -signkey server.key -out server.crt
```

When prompted, enter the passphrase you used earlier when creating your CSR.

Installing the Private Key and Certificate to your ESP Server

You must install the certificate and private key on **all** ESP server node(s) that will host a service binding using SSL. Copy the keys and certificates to the correct locations and set the appropriate ownership and permissions

Your Private Key and certificate must be copied to /var/lib/HPCCSystems/myesp/ as illustrated in the following example.

1. Copy the certificate (crt) file to the required location on the ESP server(s) :

   ```
   sudo cp server.crt /var/lib/HPCCSystems/myesp/server.crt
   ```

2. Change the owner of the file to be HPCC:

   ```
   sudo chown hpcc:hpcc /var/lib/HPCCSystems/myesp/server.crt
   ```

3. Set the file permissions:

   ```
   sudo chmod 644 /var/lib/HPCCSystems/myesp/server.crt
   ```

4. Copy the private key to the ESP server(s):

   ```
   sudo cp server.key /var/lib/HPCCSystems/myesp/private.key
   ```

5. Change the owner of the file to be HPCC:

```
sudo chown hpcc:hpcc /var/lib/HPCCSystems/myesp/private.key
```

6. Set the file permissions:

```
sudo chmod 600 /var/lib/HPCCSystems/myesp/private.key
```

Configure HTTPS on your ESP Server

Start Configuration Manager in Advanced Mode

1. Start the Configuration Manager Service on one node (usually the first node is considered the head node and is used for this task, but this is up to you).

```
sudo /opt/HPCCSystems/sbin/configmgr
```

2. Using a Web browser, go to the Configuration Manager's interface.

   Use the url of http://nnn.nnn.nnn.nnn:pppp, where nnn.nnn.nnn.nnn is the IP address of the node running Configuration Manager and pppp is the port (default is 8015).

   The Configuration Manager startup wizard displays.

3. Select **Advanced View**.

4. Select an XML file from the drop list.

   This list is populated from versions of an environment XML file in your server's **/etc/HPCCSystems/source/** directory.

   Tip: The XML file that matches the active environment.xml is highlighted.

5. Press the **Next** button.

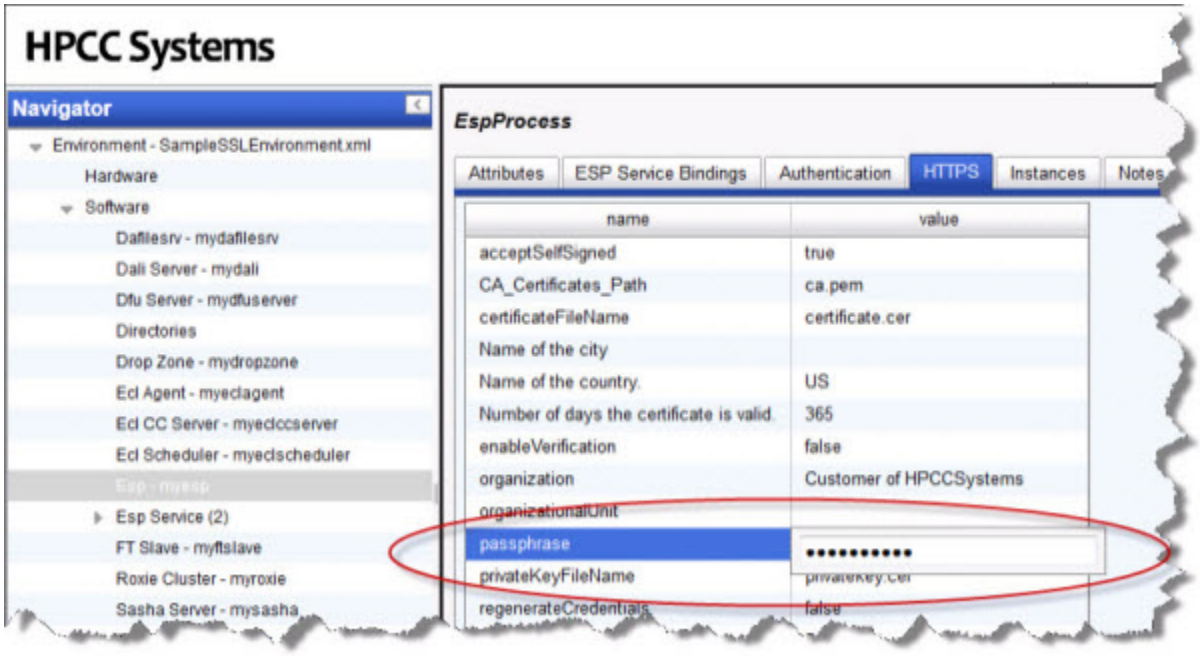
   The Configuration Manager Advanced View interface displays.

6. Check the **Write Access** box at the top of the page.

Configure ESP

1. Select ESP - MyEsp in the Navigator panel on the left side.
2. Select the **HTTPS** tab.

Figure 26. Select HTTPS Tab

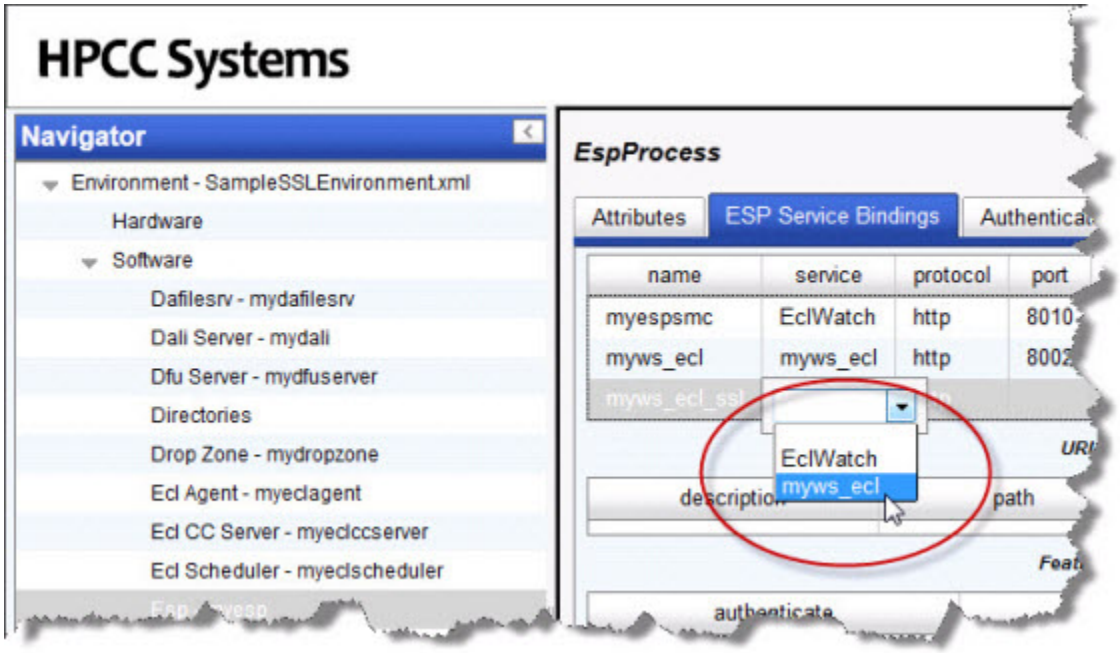


3. In the **passphrase** entry control, enter the passphrase you used earlier when you created the private key.
4. When prompted, provide the passphrase again.
5. Click the disk icon to save.

Configure one or more SSL-Enabled Service Bindings

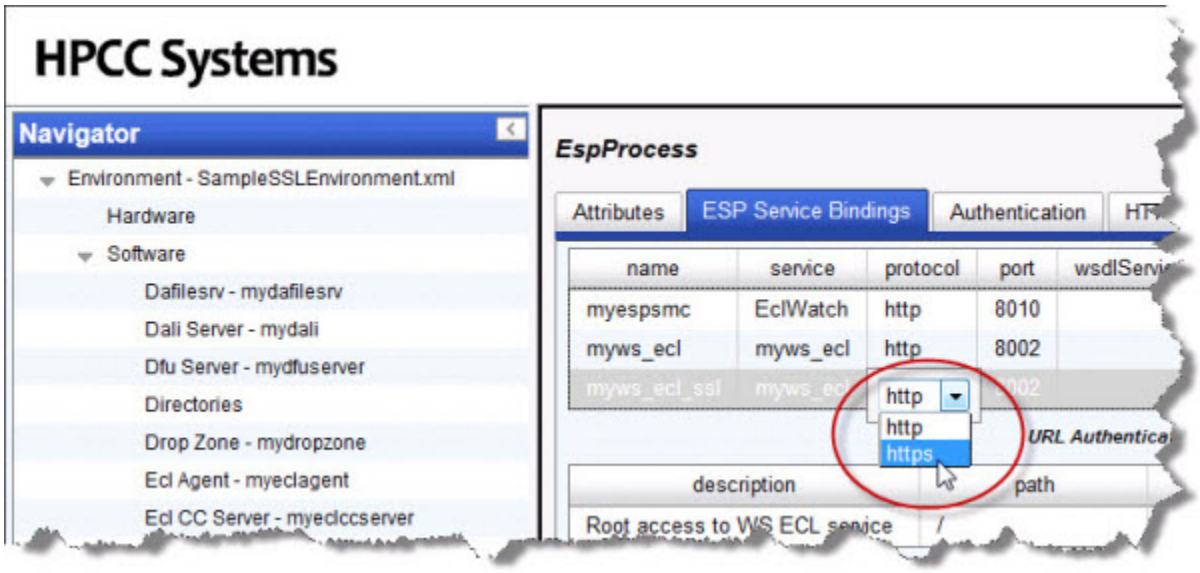
1. Select the ESP Service Bindings tab.
2. Right-click on the list of services, then select **Add**.
3. Provide a name for the binding (e.g., myws_ecl_ssl)
4. Select myws_ecl from the service drop-list.

Figure 27. myws_ecl



5. Select https from the protocol drop-list.

Figure 28. Select HTTPS



Note: If you have not previously edited the port, the change from http to https triggers Configuration Manager to automatically change the port to the default port for https (18002). It only updates automatically if the port has not been edited.

6. Click the disk icon to save

To ensure security, once you have confirmed access to your secure service via https, you delete the service binding which uses http. You should then repeat the process for **all** other service bindings.

Distribute the environment configuration file to all nodes, Restart, and Certify

Once your environment is set up as desired, you must copy the configuration file out to the other nodes.

1. If it is running, stop the system.

   Make sure system is stopped before attempting to move the environment.xml file.

2. Back up the original environment.xml file

   ```
   # for example
   sudo cp /etc/HPCCSystems/environment.xml /etc/HPCCSystems/environment.bak
   ```

   Note: the "live" environment.xml file is located in your **/etc/HPCCSystems/** directory. ConfigManager works on files in **/etc/HPCCSystems/source** directory. You must copy the XML file from this location to make an environment.xml file active.

3. Copy the NewEnvironment.xml file from the source directory to the /etc/HPCCSystems and rename the file to environment.xml

   ```
   # for example
   sudo cp /etc/HPCCSystems/source/NewEnvironment.xml /etc/HPCCSystems/environment.xml
   ```

4. Copy the **/etc/HPCCSystems/environment.xml** to the **/etc/HPCCSystems/** on every node.

   You might prefer to use a script to automate this step, especially if you have many nodes. See the Example Scripts section in the Appendix of the Installing and Running the HPCCPlatform manual.

5. Restart the HPCC system and certify the components as usual.

User Security Maintenance

Configuring an HPCC Systems® platform to use Active Directory or LDAP-based security allows you to set permissions to control access to Features, File Scopes, and Workunit Scopes.

Introduction

HPCC Systems® maintains security in a number of ways. HPCC Systems® can be configured to manage users' security rights by pointing either at Microsoft's Active Directory on a Windows system, or a 389Directory Server on Linux systems.

Using the Permissions interface in ECL Watch, administrators can control access to features in ECL IDE, ECL Watch, ECL Plus, DFU Plus, and the ECL modules within the Attribute Repository. Optionally, you can also implement file and workunit access control by enabling that setting in the Dali server.

Establish permissions by group or by user and define them by association with a particular feature of the HPCC Systems platform. Permissions can be defined for each unique combination of group and feature. Permissions are separated into the following categories:

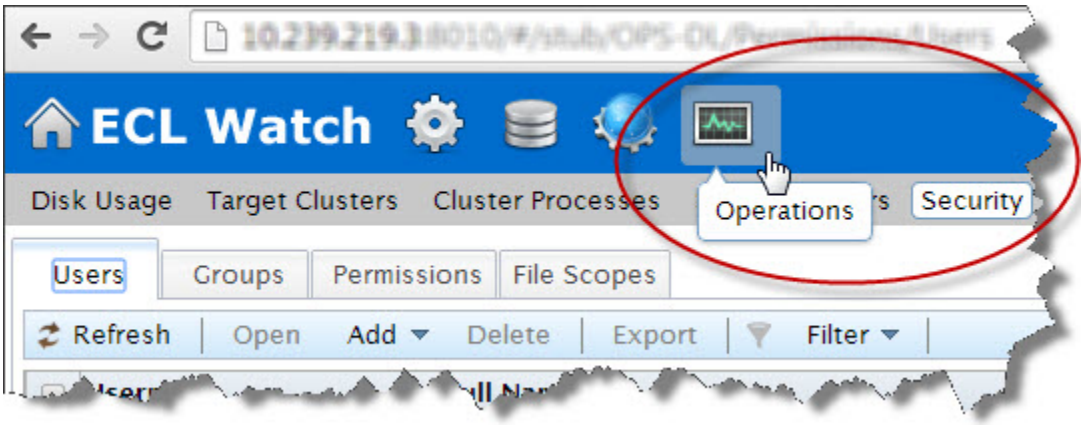
Esp Features for SMC	Controls access to features in ECLWatch and similar features accessed from ECL IDE.
Esp Features for WsEclAccess	Controls access to the WS-ECL web service
Esp Features for EclDirectAccess	Controls access to the ECLDirect web service
File Scopes	Controls access to data files by applying permissions to File scopes
Workunit Scopes	Controls access to Workunits by applying permissions to Workunit scopes
Repository Modules	Controls access to the Attribute Repository and Modules in the repository (legacy)

Security Administration using ECL Watch

Administrator rights are needed to manage permissions. Once you have administrator access rights, open ECL Watch in your browser using the following URL:

- **http://nnn.nnn.nnn.nnn:pppp (where nnn.nnn.nnn.nnn is your ESP Server's IP Address and pppp is the port. The default port is 8010).**

Security administration is controlled using the **Security** area of ECL Watch. To access the Security are click on the**Operations** icon, then click the **Security** link from the navigation sub-menu.



There are three areas where permissions may be set:

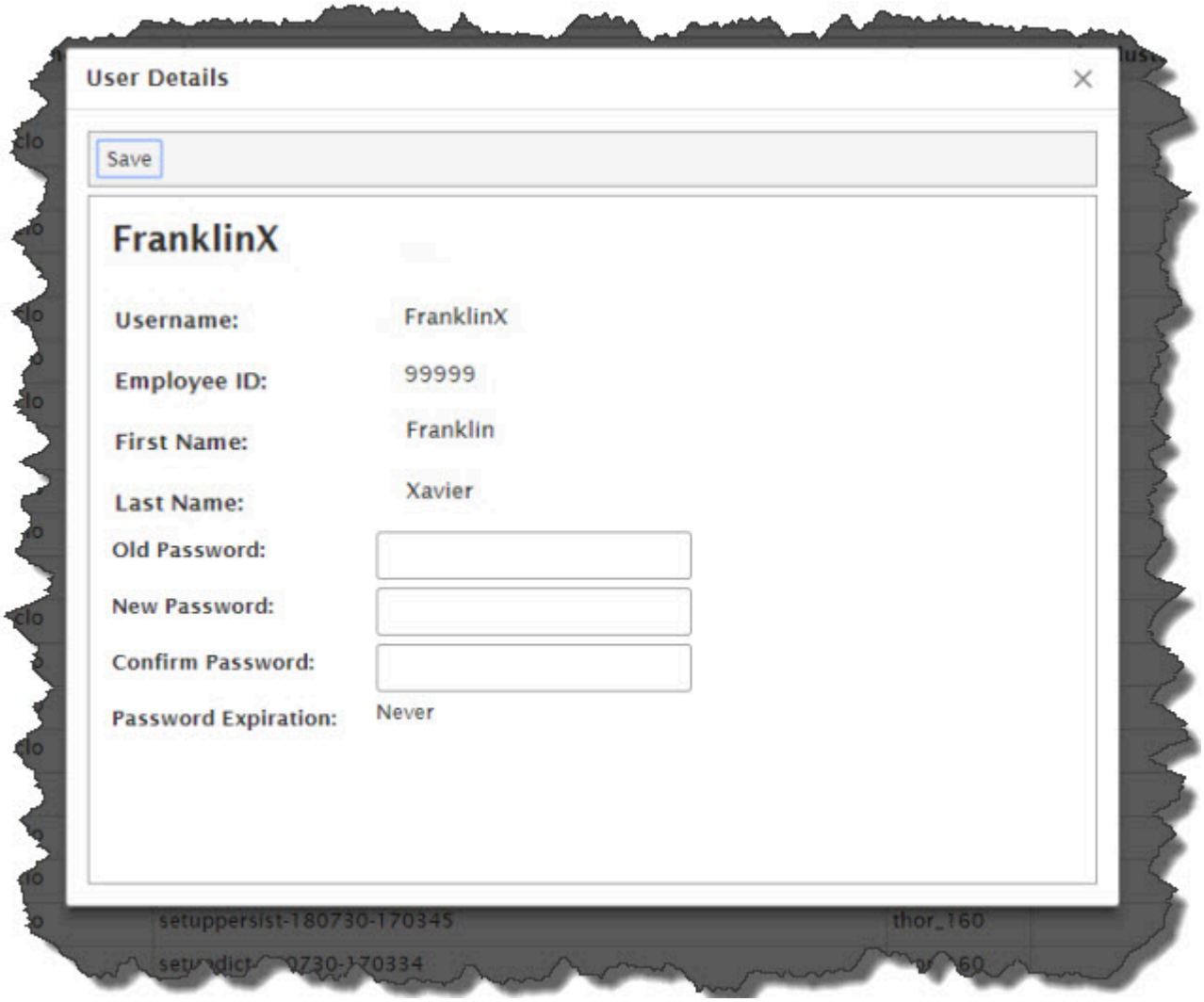
- **Users**. Shows all the users currently setup. Use this area to add or delete a user, edit a user's details, set/reset a user's password and view the permissions currently assigned to a user.
- **Groups**. Shows all the groups currently setup. Use this area to add or delete a group, view and edit the members of a group, view and edit the permissions that have been set for a group.
- **Permissions**. Shows the features of the HPCC Systems where permissions may be set. Use this area to view the permissions currently set for any area of HPCC Systems, or to add groups and users and set/modify their permission for a specific feature

	**NOTE:** Use caution when setting any explicit **deny** permission setting. The most restrictive permission always applies.

Information about your account

To find out more information about your account, in ECL Watch click on your username link under **Logged In As:** at the top of the ECL Watch page.

- A **User Details** tab with your account information displays.



- You can change your password here, if desired.
- You can also verify the password expiration date, if your password is set to expire.

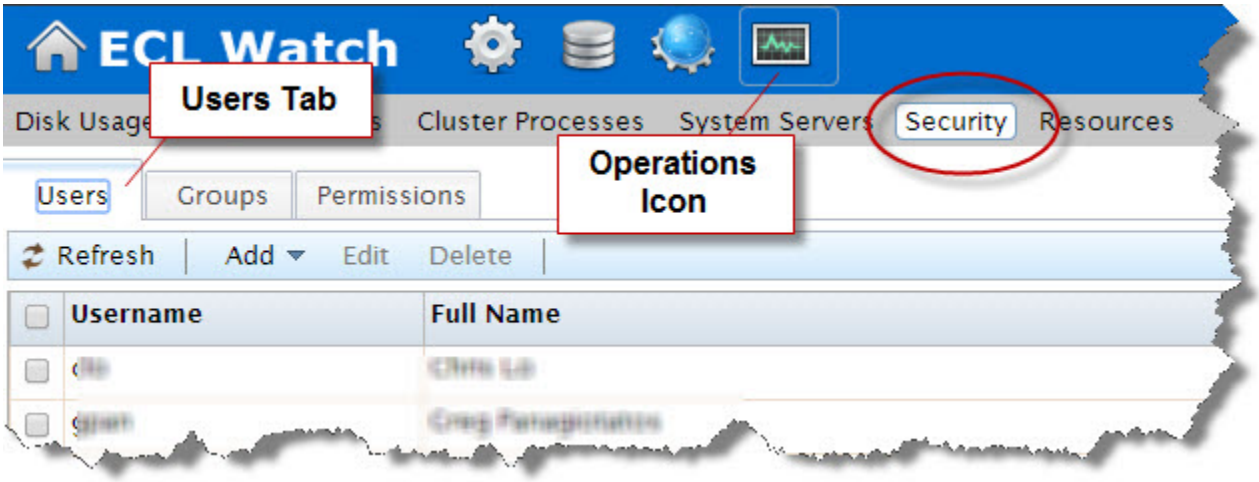
Setting and modifying user permissions

In a security-enabled environment, access to ECL Watch and its features is controlled using a login and password. The **Users** area enables you to control who has access to ECL Watch and the features of your HPCC Systems to which they have access. Permissions can be set for users based on their individual needs and users can also be added to groups which have already been set up. Use the **Users** menu item to:

- Add a new user (**note**: the Username cannot be changed)
- Delete a user
- Add a user to a group
- Change a user's password
- Modify the details/permissions of an individual user

Adding and editing users

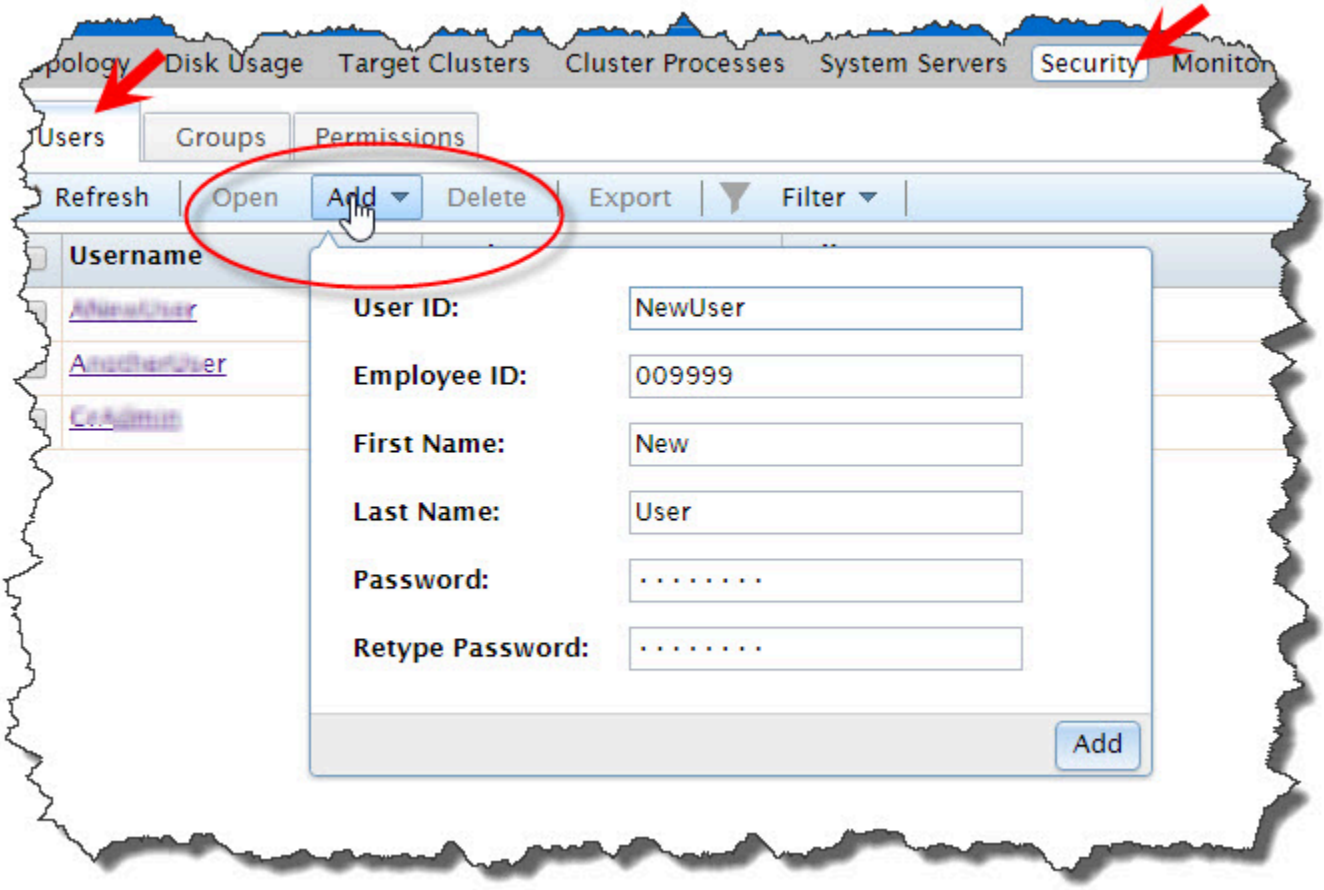
To access the user administration sections click on the **Operations** icon, then click the **Security** link from the navigation sub-menu. Click on the **Users** tab to add or edit users.



All current users are identified in the list by their Username and Full Name.

To add a new user to the list of authenticated users:

To add a new user you must have Administrator level access.



1. Press the **Add** button.

   The add user dialog displays.

2. Enter a **Username**.

   This is the login name to use ECL Watch, ECL IDE, WsECL, etc.

3. Enter the **First Name** and **Last Name** of the user.

   This information helps to easily identify the user and is displayed in the **Full Name** field on the main **Users** window.

4. Enter a **Password** for the user and then confirm it in the **Retype Password** field.

   NOTE: The password must conform to the policy of your security manager server.

5. Press the **Add** button.

   A successful addition opens a new tab where you can verify the new user's information.

6. Press the **Save** button.

Once added, the new user displays in the list and you can modify details and set permissions as required.

To modify a user's details:

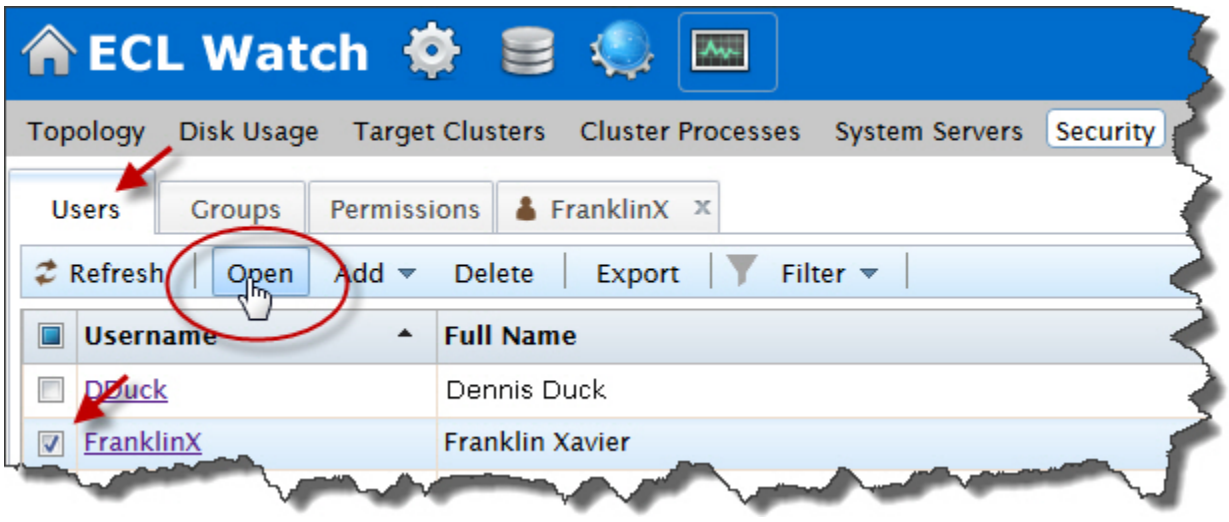
Click on the **Operations** icon, then click the **Security** link from the navigation sub-menu.

1. Click on the **Users tab**.

   The users display in a list.

2. Select the user (or users) to modify. Click on the **Username** link to open the users' details tab.

   To select multiple users, check the box next to the Username to select. This enables the Users action buttons. Press the **Open** action button.



A tab opens for each user selected. On each user's tab there are several sub-tabs.

The user's details are on the **Summary** tab.

3. Modify the user's details as required (if more than one user selected, repeat for each user).

   Note: The **Username** cannot be changed.

4. Press the **Save** button.

   A confirmation message displays.

To add a user to a group:

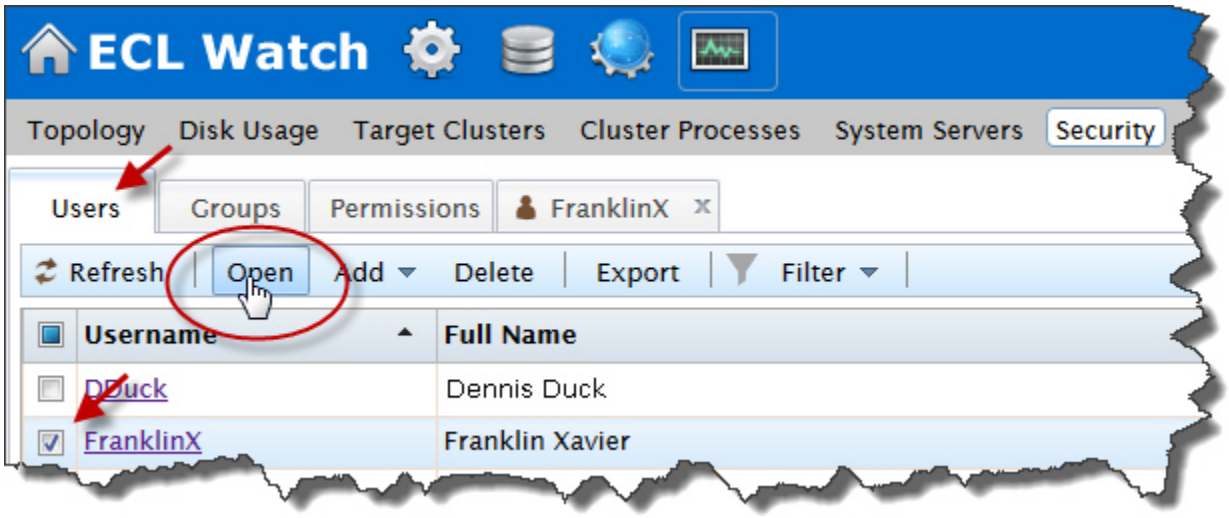
Click on the **Operations** icon, then click the **Security** link from the navigation sub-menu.

1. Click on the **Users tab**.

   The users display in a list.

2. Select the user (or users) to modify. Click on the **Username** link to open the users' details tab.

   To select multiple users, check the box next to the Username to select. This enables the Users action buttons. Press the **Open** action button.

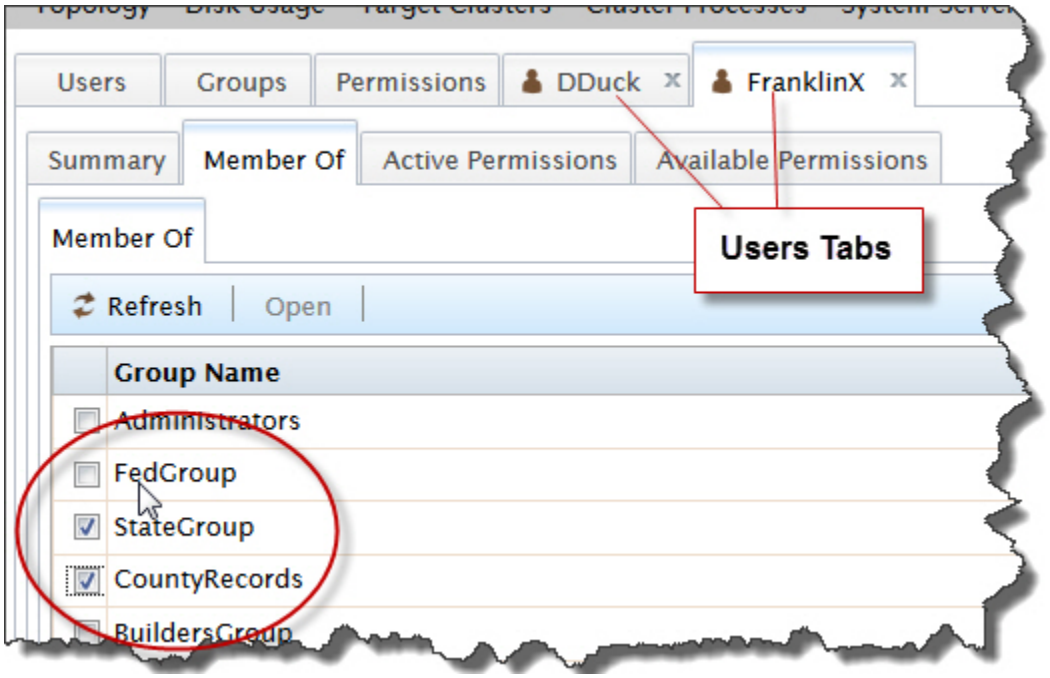


   A tab opens for each user selected. On each user's tab there are several sub-tabs.

   The user's details are on the **Summary** tab.

3. Click on the tab for the user to modify (if more than one user selected, repeat for each user).

   On the user's tab there are several sub-tabs.



Click on the **Member Of** sub-tab to modify that user's groups.

4. On the **Member Of** tab for that user, a list of the available groups display.

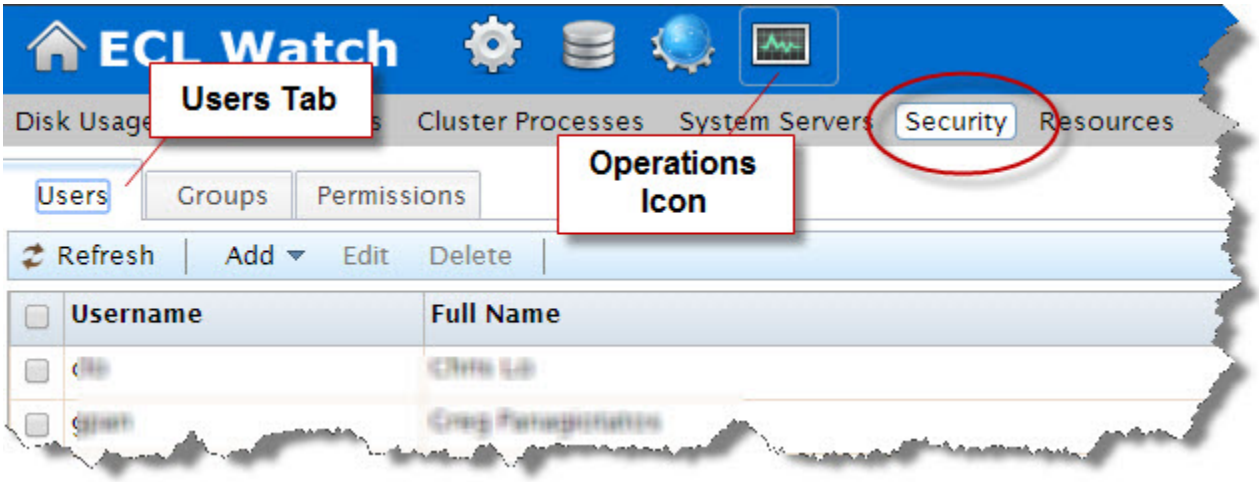
   To add the user to the group, check the box next to the desired group.

5. The changes are automatically saved. Close the tab.

To promote a user to an Administrator

To modify a users credentials you must have Administrator level access. You can designate the HPCC Systems Administrator account to have limited permissions only relating to HPCC Systems elements and not LDAP administrator's rights. To promote a user to an HPCC Systems Administrator, add the user to the configured **Administrators** group.

Click on the **Operations** icon, then click the **Security** link from the navigation sub-menu.

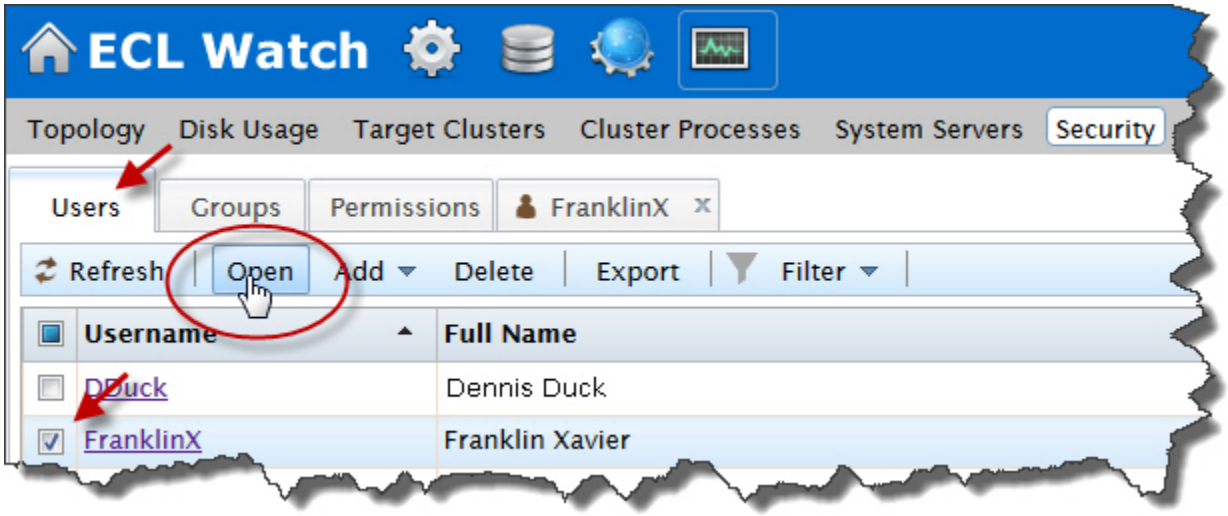


1. Click on the **Users tab**.

The users display in a list.

2. Select the user (or users) to promote. Click on the **Username** link to open the users' details tab.

   To select multiple users, check the box next to the Username to select. This enables the Users action buttons. Press the **Open** action button.



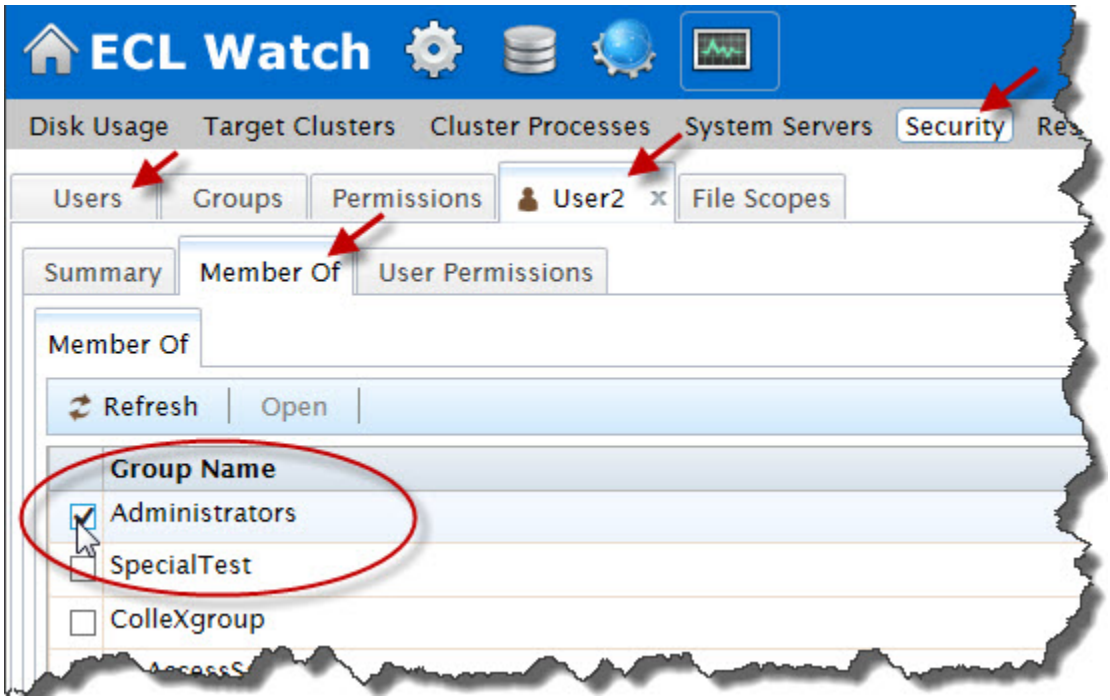
   A tab opens for each user selected. On each user's tab there are several sub-tabs.

   The user's details are on the **Summary** tab.

3. Click on the tab for the user to modify (if more than one user selected, repeat for each user).

   On the user's tab there are several sub-tabs.

   Click on the **Member Of** sub-tab.



4. Select **Administrators** by placing a check in box.

   NOTE: The name of the default Administrators group could vary. It is a configurable value defined as the value of **adminGroupName** in the configuration. For example, if you set the adminGroupName to "HPCCAdministrators", in the environment then HPCCAdministrators would display in the list.

5. The changes are automatically saved. Close the tab(s).

To delete a user from a group:

To delete a user from a group you must have Administrator level access.

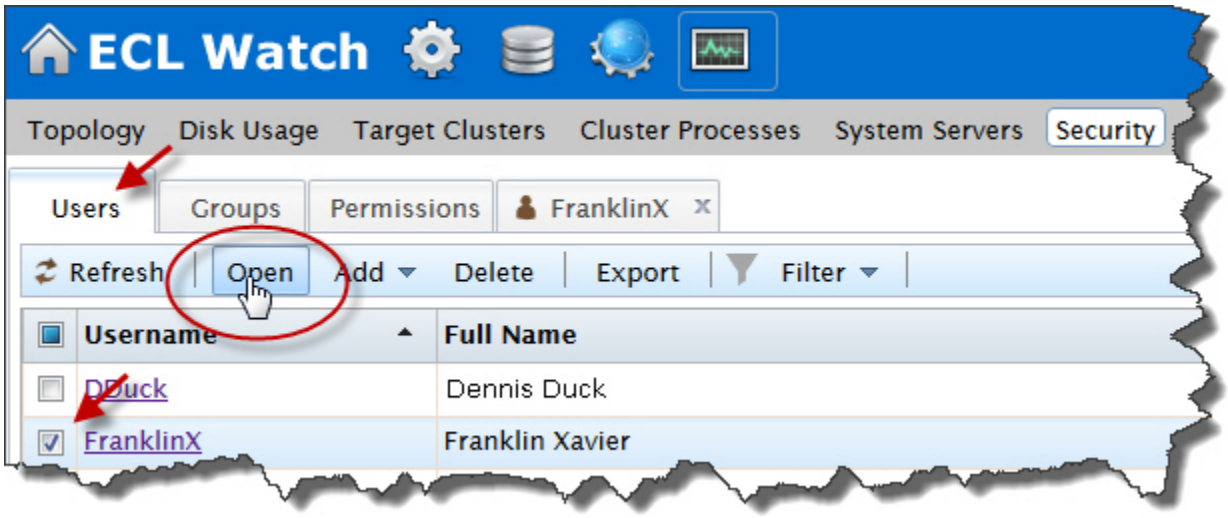
Click on the **Operations** icon, then click the **Security** link from the navigation sub-menu.

1. Click on the **Users tab**.

   The users display in a list.

2. Select the user (or users) to remove. Click on the **Username** link to open the users' details tabs.

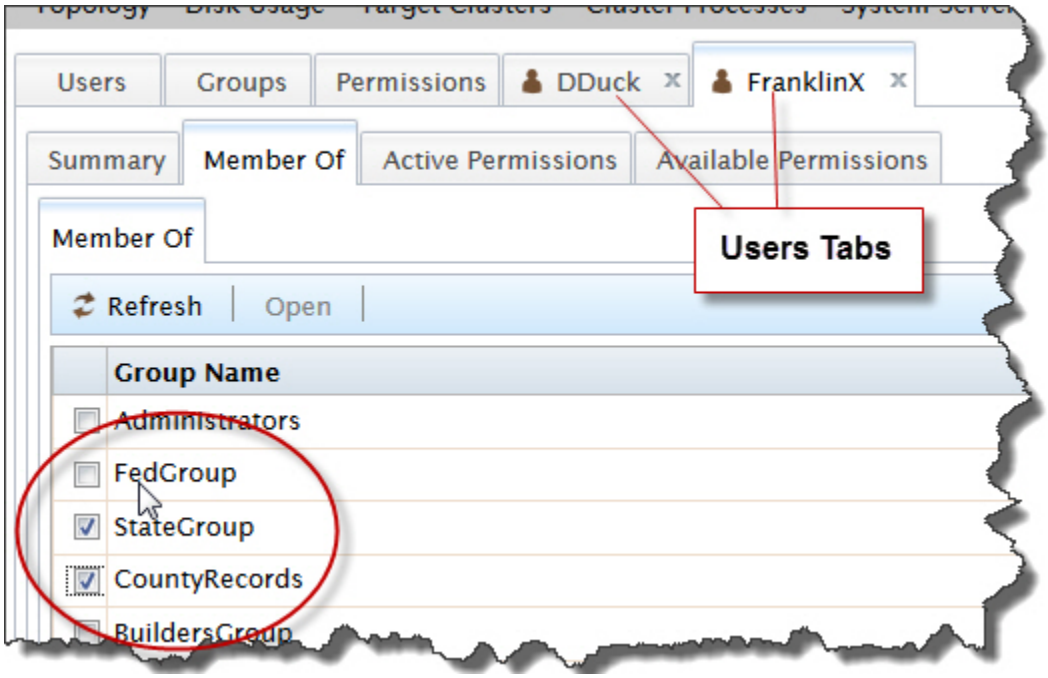
   To select multiple users, check the box next to the Username to select. This enables the Users action buttons. Press the **Open** action button.



   A tab opens for each user selected. On each user's tab there are several sub-tabs.

3. Click on the tab of the user to modify (if multiple users selected, repeat for each user).

   On the user's tab there are several sub-tabs.



Click on the **Member Of** sub-tab to modify that user's groups.

4. On the **Member Of** tab for that user, there is a list of the available groups.

   There is a check in the box next to each group that user belongs to.

   To remove that user from a group, uncheck the box next to the desired group.

5. The changes are automatically saved. Close the tab.

To change a user's password:

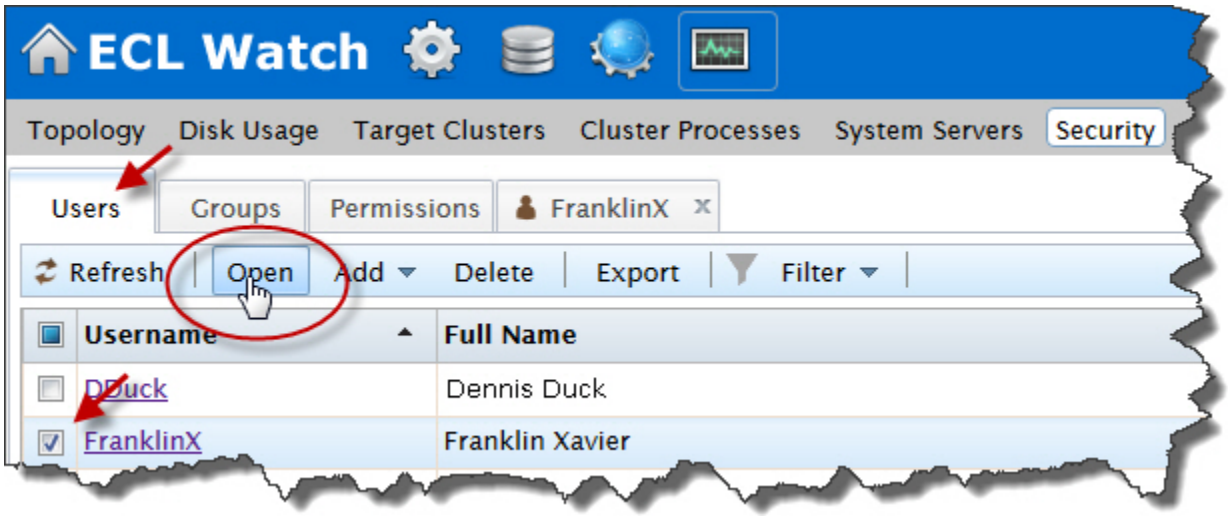
Click on the **Operations** icon, then click the **Security** link from the navigation sub-menu.

1. Click on the **Users tab**.

   The users display in a list.

2. Select the user (or users) to modify. Click on the **Username** link to open the users' details tab.

   To select multiple users, check the box next to the Username to select. This enables the Users action buttons. Press the **Open** action button.



A tab opens for each user selected. On that tab there are several sub-tabs.

The user details are on the **Summary** tab.

3. Select the Summary tab.
4. Change the password in the **Password** and **Retype New Password** fields as required on the User details summary tab (if multiple users selected, repeat for each user).

   Note: The **Username** cannot be changed.
5. Press the **Save** button.

   A confirmation message displays.

To delete a user from the list of authenticated users:

Click on the **Operations** icon, then click the **Security** link from the navigation sub-menu.

1. Click on the **Users** tab.

   The users display in a list.
2. Check the box to the left of the user(s) you want to remove.

   Note: These users will no longer have access to ECL Watch.
3. Press the **Delete** action button.

   Confirmation displays.

Setting permissions for an individual user

There may be occasions when you need to modify the permissions for individual users. For example, users may have individual security needs that are not completely covered in any group or, there may be occasions when a user requires temporary access to an HPCC Systems feature. Permissions set in this area of ECL Watch only affect the user you choose. Most individual permissions you set here overwrite ones set in any group to which the user belongs, except in the case of an explicit deny.

To set permissions for an individual user:

Click on the **Operations** icon, then click the **Security** link from the navigation sub-menu.

1. Click on the **Users tab**.

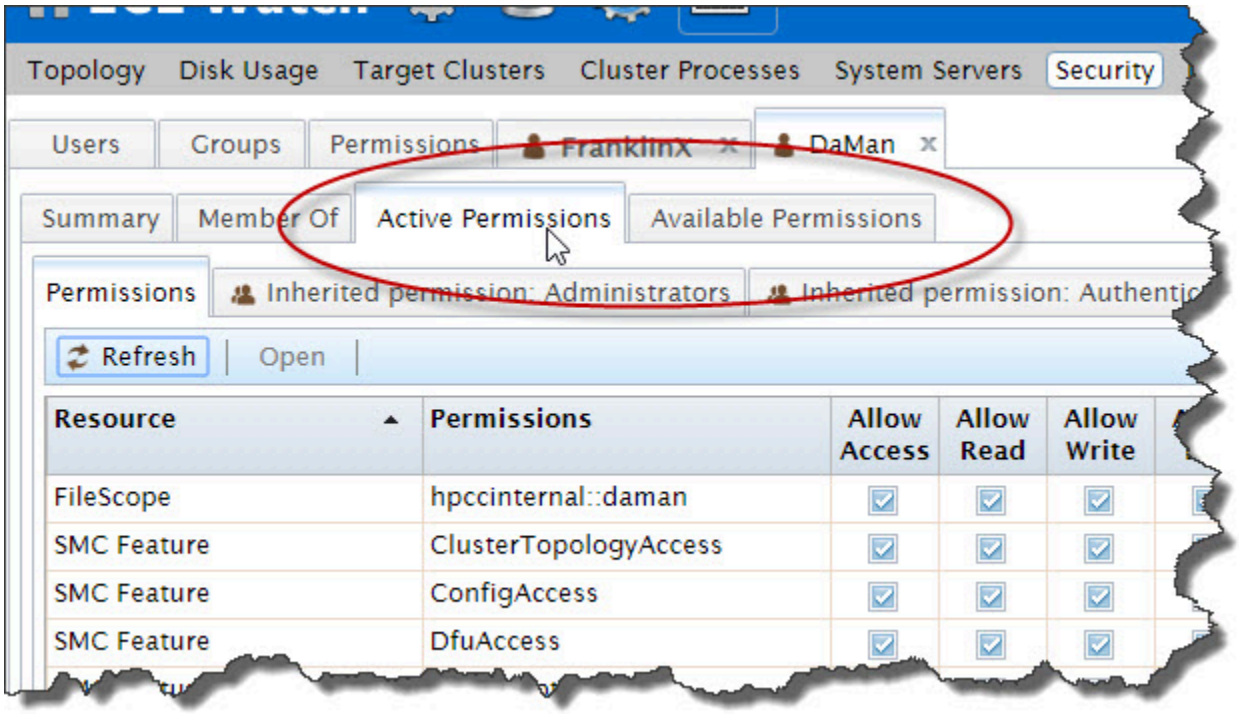
   The users display in a list.

2. Select the user (or users) to modify. Click on the **Username** link to open the users' details tab.

   To select multiple users, check the box next to the Username to select. This enables the Users action buttons. Press the **Open** action button.

3. Click on the tab of the username to modify (if multiple users selected, repeat for each user).

   On the user's tab there are several sub-tabs.

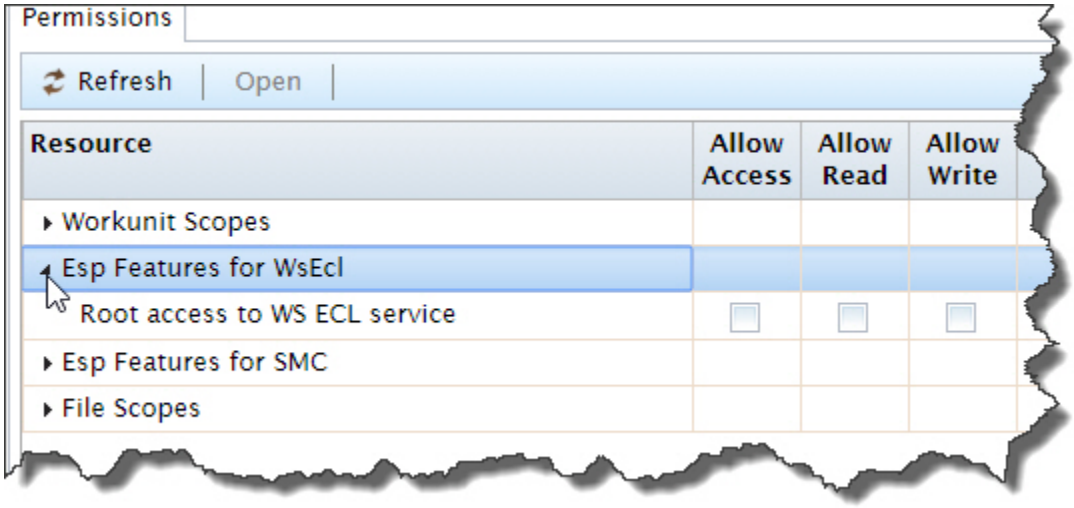


   Click on the **Active Permissions** sub-tab to view the user's current permissions.

4. Click on the **Available Permissions** tab to see all the sets of permissions that are available to apply to that user.

   When you select permissions from the Available Permissions tab, they display and can be set in the Active Permissions tab.

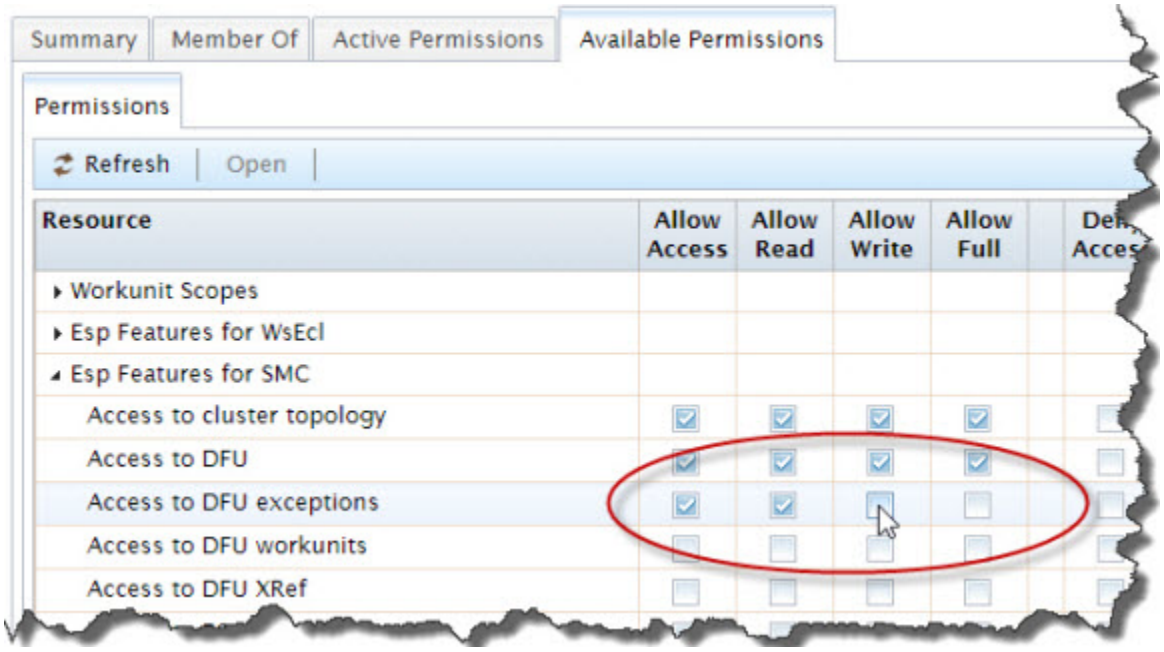
5. Click on the arrow next to the resource to display the permissions that can be set for that resource.



The list of permission groups currently set for this user and the ones the user has inherited are also listed. Click the arrow to allow setting the individual resource settings.

6. There may be more than one resource setting available in each group, be sure to set the permissions for each setting as required.

7. Check the boxes that **allow** and **deny** access as required for the user.



	**NOTE:** Use caution when setting any explicit **deny** permission setting. The most restrictive permission always applies.

8. The changes are automatically saved. Close the tab.

Setting and modifying group permissions

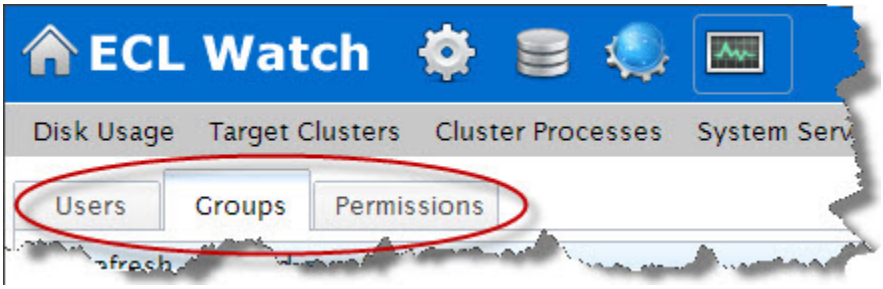
Setting up groups ensures that all users with the same permission needs have the same permission settings. You can give users the access they require to the feature areas of HPCC Systems that they need. There is no limit to the number of groups you can create. You can create as many groups as you need to control access for all your users regardless of their tasks.

Use the **Groups** menu item to:

- Add a new group.
- Delete a group.
- Add members to a group.
- Modify the permissions for a group.

Adding and editing groups

When adding or changing the permissions for a group, all members of that group are given those permission settings. So it is important to be sure that you are giving or denying access to features appropriate for the members of that group. If you need to make a change for a single user (or small number of users), it is probably better to make that change for each individual user as illustrated in the previous sections.

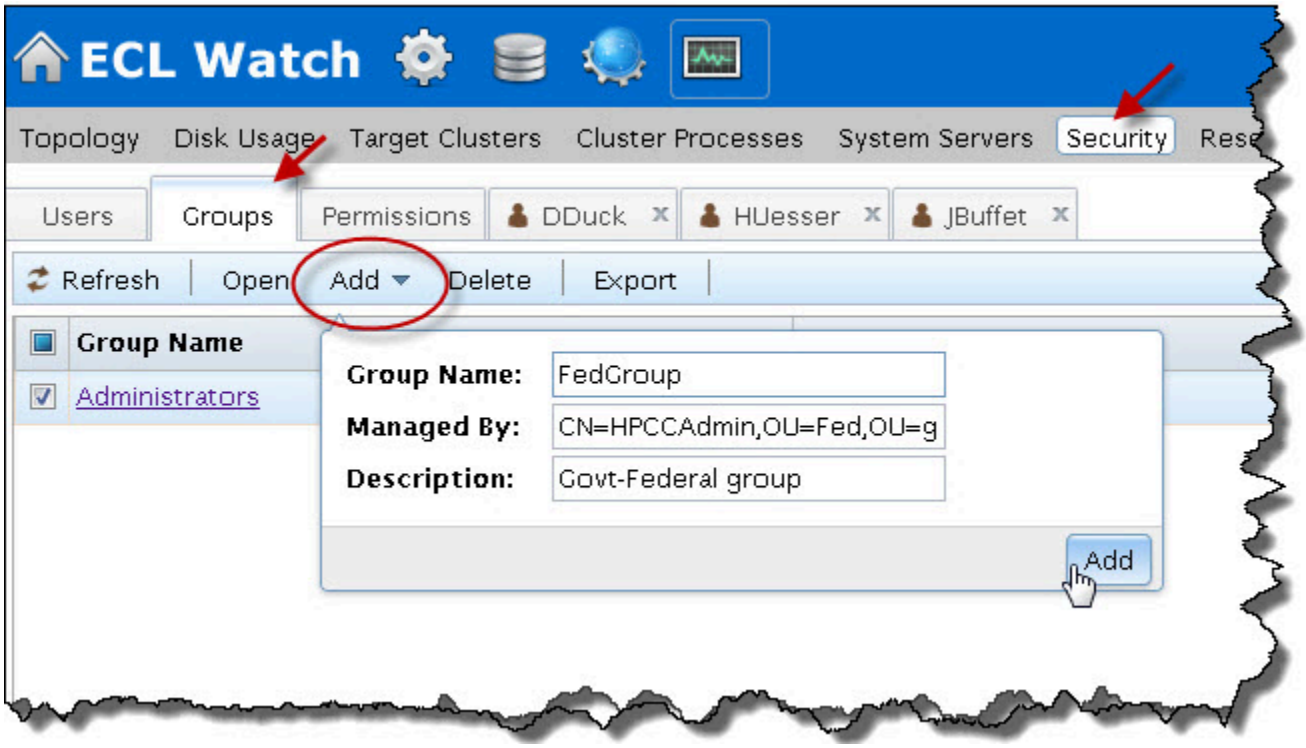


To modify groups, click on the **Operations** icon, then click the **Security** link from the navigation sub-menu. Click on the **Groups** tab.

To add a new group:

Click on the **Operations** icon, then click the **Security** link from the navigation sub-menu.

1. Click on the **Groups** tab.
2. Press the **Add** action button button.



This opens a dialog where you can enter the name for the group.

3. Enter a **Group Name**.
4. Enter the fully qualified Distinguished Name for the owner of the group **Managed By** field.
5. Enter a description of the group. (optional)
6. Press the **Add** button.

   This opens a new tab for the group and several sub tabs

   The **Summary** sub-tab displays the group name.

   The **Members** tab displays the list of users, check the box next to each user to add to the group.

   The **Active Group Permissions** tab displays the permissions applied to the group.

   The **Available Group Permissions** tab displays all the available permissions, selecting from the Available Permissions applies them to the Active Group Permissions.

   You can set the permissions and add members to this group from the respective sub-tabs on that group tab.

To delete a group:

Click on the **Operations** icon, then click the **Security** link from the navigation sub-menu.

1. Click on the **Groups** tab.
2. Locate the group in the list and check the checkbox next to it.

3. Press the **Delete** action button.
4. Press the **OK** confirmation button.

   The group no longer displays in the list.

To add new members to a group:

Click on the **Operations** icon, then click the **Security** link from the navigation sub-menu.

1. Click on the **Groups** tab.
2. Locate the group in the list and check the box next to it.
3. Press **Open** action button.

   This opens a new tab for the group.

   The sub-tabs display: **Summary**, **Members**, **Active Group Permissions**, and **Available Group Permissions**.
4. Select the **Members** tab.

   The members tab displays a list of all users on the system. The users that belong to the selected group have a check in the box next to them.
5. Check the box(es) to the left of the users you want to add to the group.
6. The changes are automatically saved. Close the tab.

To delete members from a group:

Click on the **Operations** icon, then click the **Security** link from the navigation sub-menu.

1. Click on the **Groups** tab.
2. Locate the group in the list and check the box next to it.
3. Press the **Open** action button.

   This opens a new tab for the group.

   The Groups tab has several sub-tabs: **Summary**, **Members**, **Active Group Permissions** and **Available Group Permissions**.
4. Select the **Members** tab.

   The Members tab displays a list of all users on the system. The users that belong to the selected group have a check in the box next to them.
5. Uncheck the box(es) to the left for all users you want to delete from the group.
6. The changes are automatically saved. Close the tab.

Setting permissions for a group

By default, all users are members of the **Authenticated Users** group. The **Authenticated Users** group has access rights to almost all resources. To set up more restricted controls, you should create specific groups with more restricted permissions.

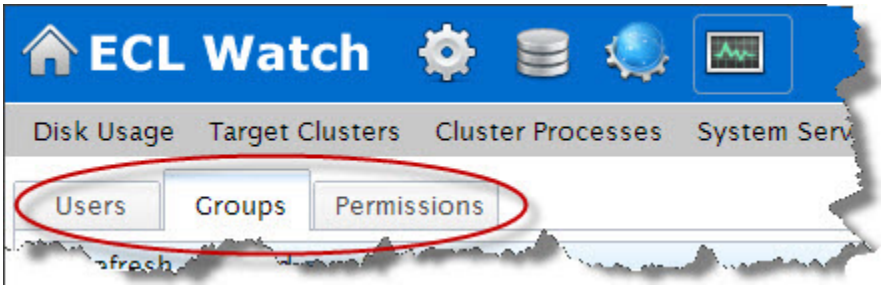
You can then create groups with only those access rights you wish to grant. This approach allows the most flexibility since a single User ID can have multiple group memberships.

As a best practice, you should use **Allow** instead of **Deny** to control access. Denies should be used only as an exception, when possible. If you wish to deny a user access to some specific control, a good practice would be to create a group for that, place the user(s) in that group, then you can deny access to that group.

Remember the most restrictive control takes precedence. For example, if a user is in a group that has deny permission to file access, and the user is in another group where file access is allowed, that user will still not have file access.

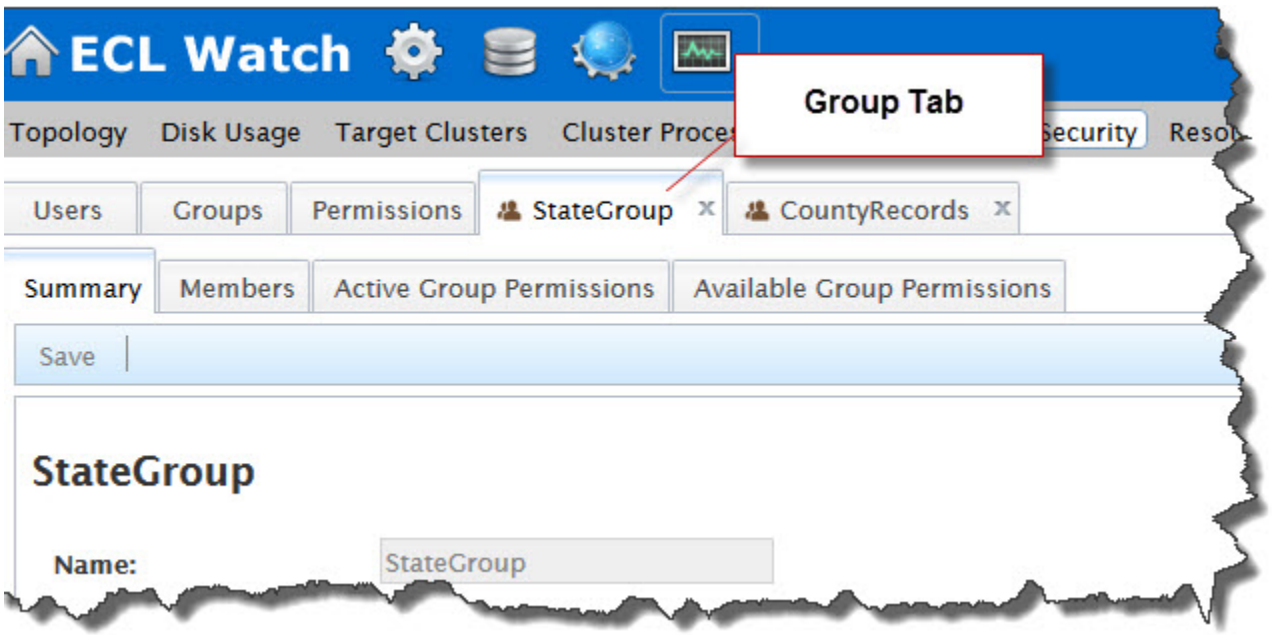
To set permissions for a group:

Click on the **Operations** icon, then click the **Security** link from the navigation sub-menu.



1. Click the **Groups** tab.
2. Locate the group in the list and check the box next to it.
3. Press the **Open** action button.

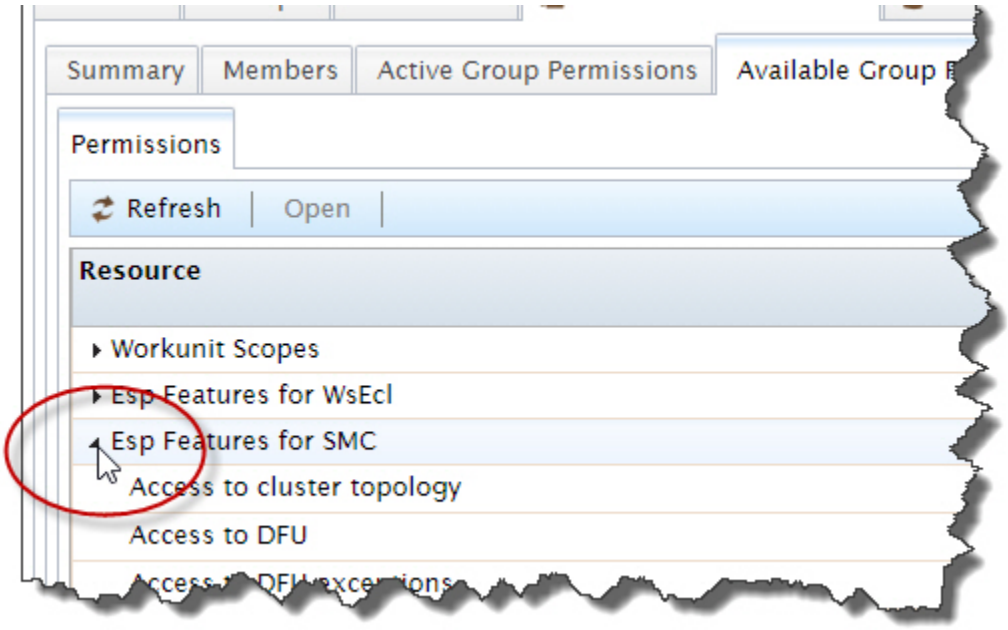
   This opens a new tab for the group.



The group tab displays the sub-tabs: **Summary**, **Members**, **Active Group Permissions** and **Available Group Permissions**.

4. Select the **Available Group Permissions** sub-tab. This displays all the available permission resources.

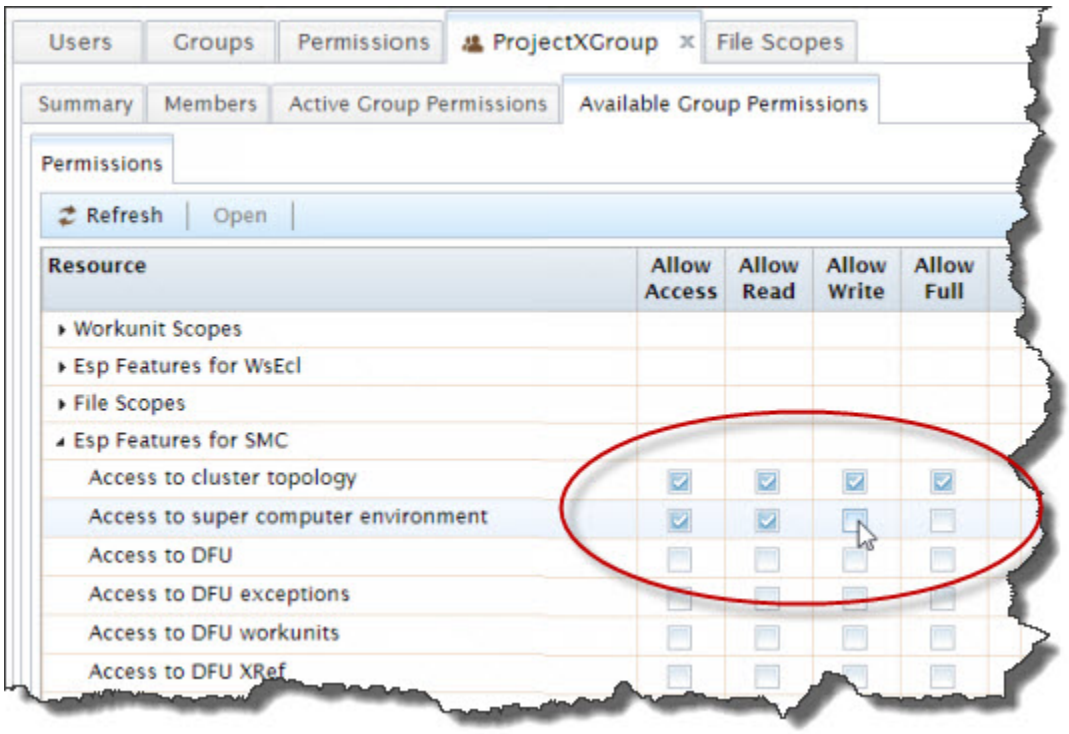
5. Click on the arrow to the left of the **Resource** to expand and expose the permission sets for the resources.



The groups permission resources display.

6. There may be more than one resource setting available in each group, be sure to set the permissions for each setting as required.

7. Check the boxes for **allow** and **deny** as required for the group.



	**NOTE:** Use caution when setting any explicit **deny** permission setting. The most restrictive permission always applies.

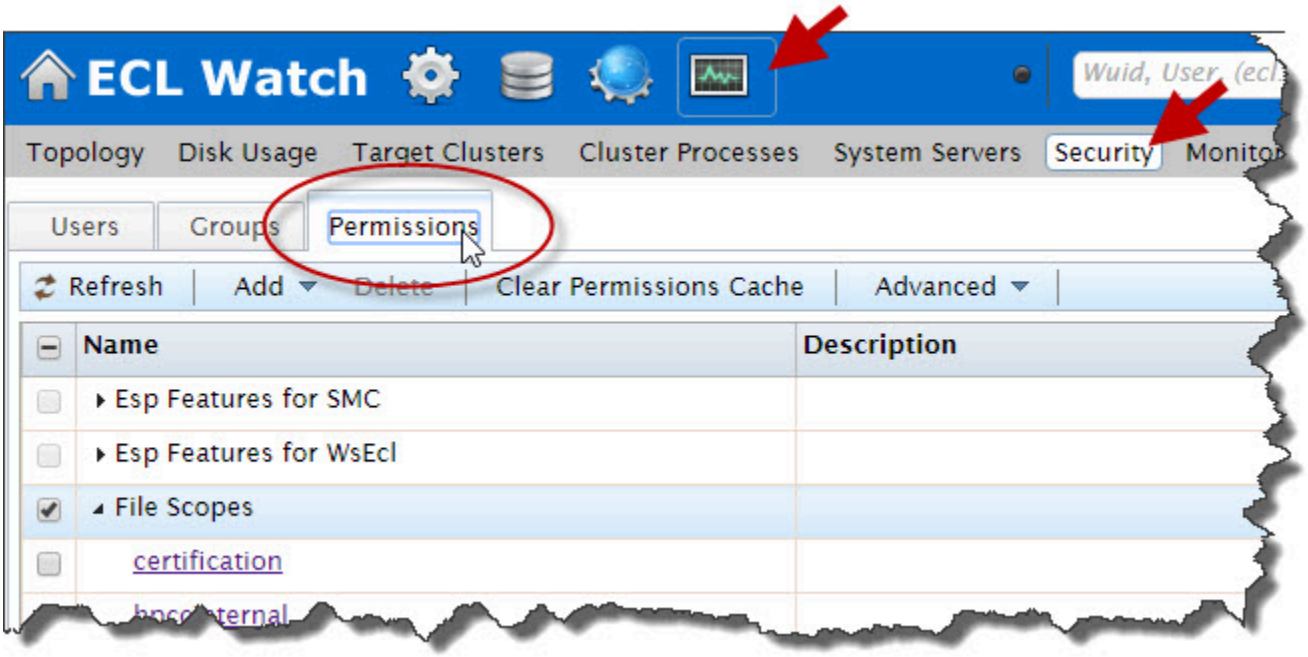
8. There may be more than one resource setting available, select the resource(s) you require from the drop list.

   Repeat for each applicable resource.

9. The changes are automatically saved. Close the tab.

Feature level access control

Access to the feature permissions is available through ECL Watch. In order to modify feature permissions you must have Administrator level access. To access the feature permissions click on the **Operations** icon, then click the **Security** link from the navigation sub-menu.

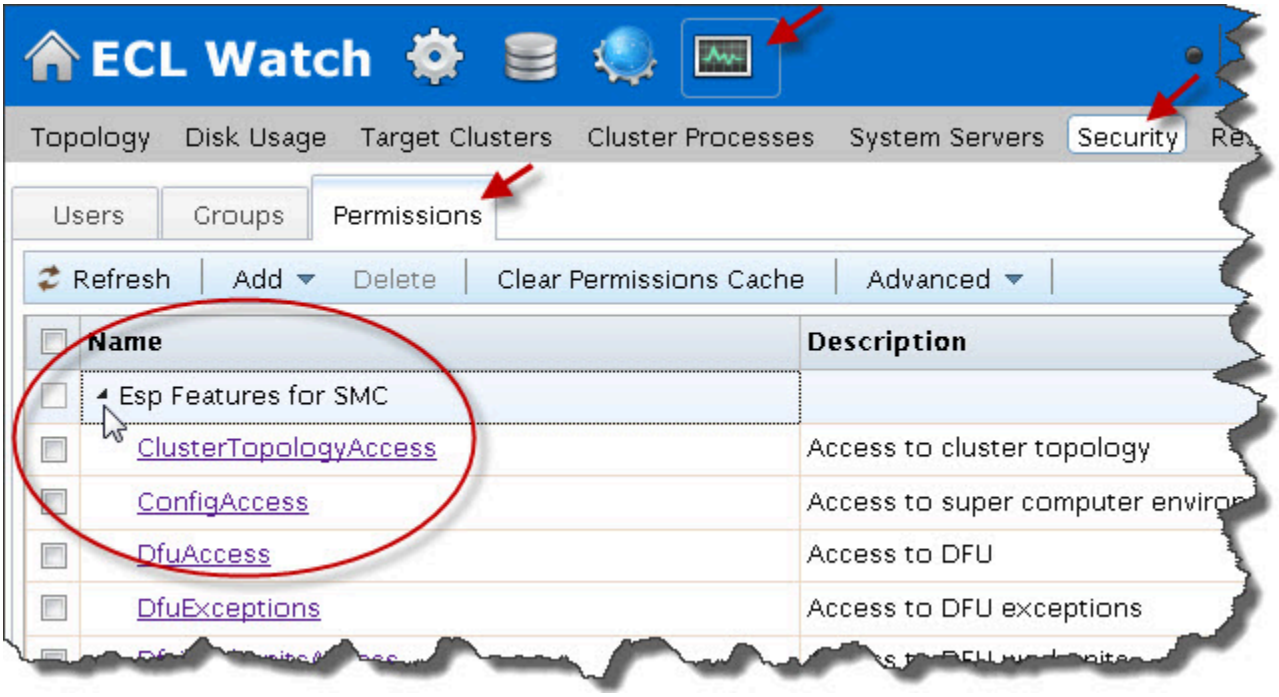


To use the feature level controls, apply the feature resource from the **Available Permissions** tab to the **Active Permissions** for users and groups. Using the feature level controls allow you to:

- View the features and permissions for any resource
- Edit the permissions for any feature
- Update the permissions for users and groups for a specific resource

Feature resources

There are several features for which you can set up access control in HPCC Systems. Access to features of the HPCC Systems platform is controlled by via the **ESP Features for SMC** category.



The available features are listed under the **Permissions** tab. You can view and gain access to the feature controls from here. However, the feature controls must be applied to users, or to groups. If you click on the feature name link, a tab opens that displays the users and groups where those feature permissions are applied.

ECL Watch feature permission settings that are not listed are not relevant and should not be used.

Apply permissions for a feature resource:

To use the feature permissions, you must apply them to a user or group(s). To access the feature permissions click on the **Operations** icon, then click the **Security** link from the navigation sub-menu.

1. Identify the user(s) or group(s) which you want to modify the feature permissions.

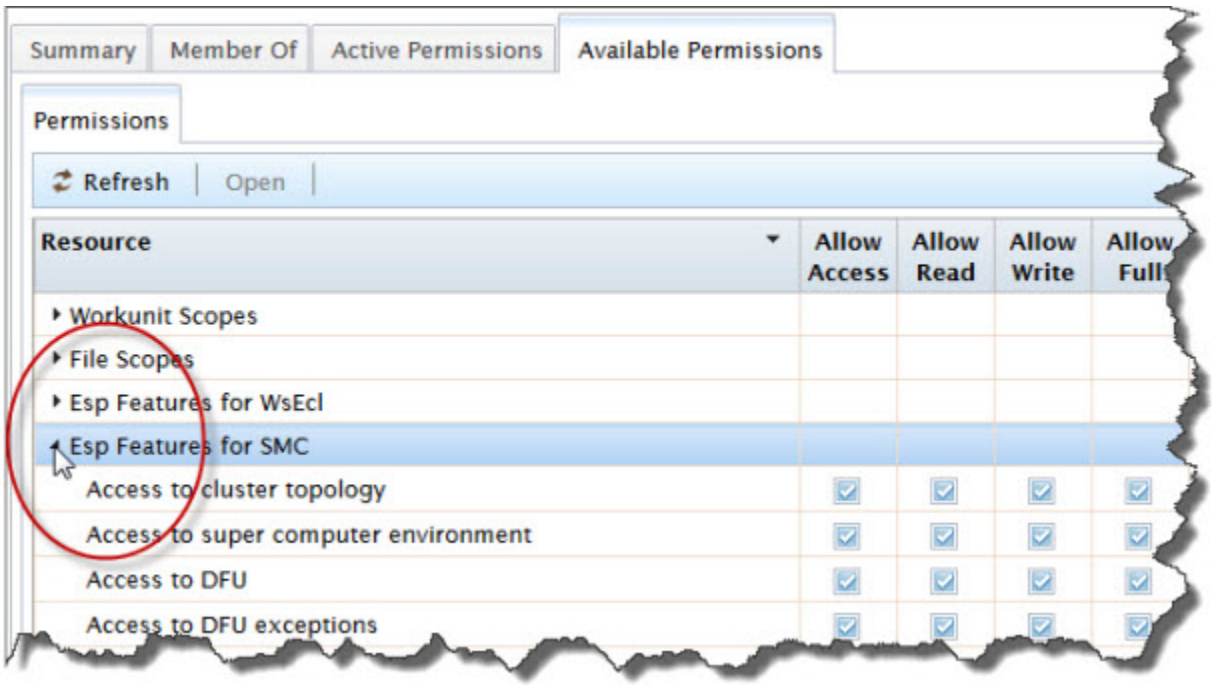
   Select the appropriate tab. (Users or Groups)

2. Check the checkbox(es) next to the user(s) or group(s) to modify.

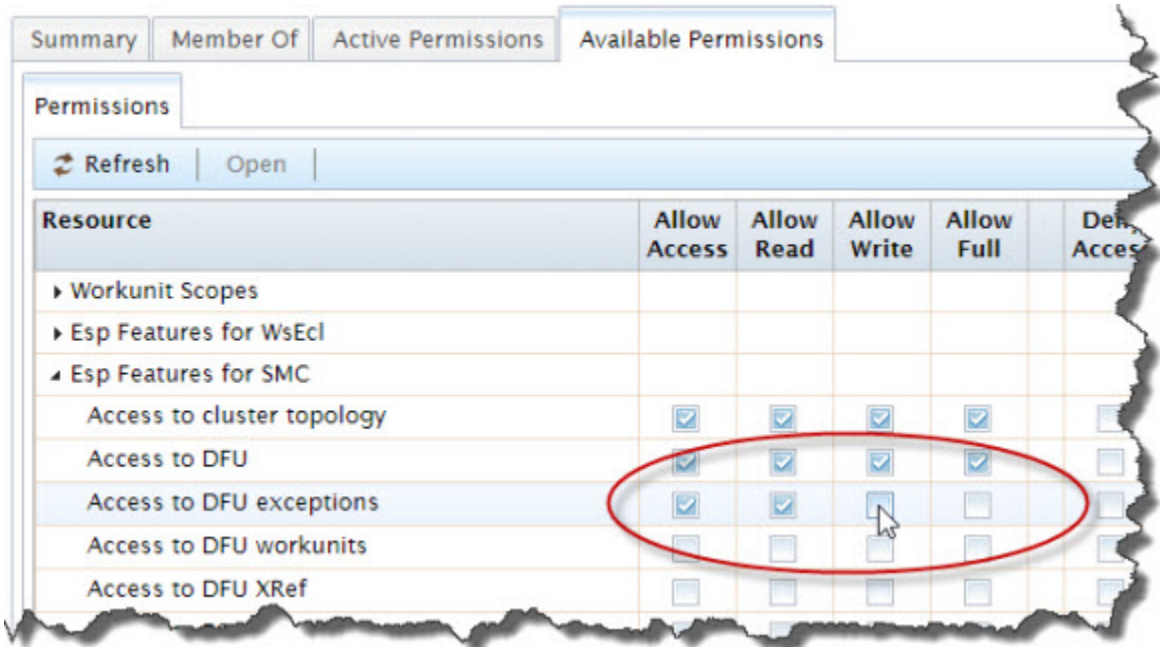
3. Press the **Open** action button. A tab for each user or group selected opens.

4. Click the **Available Permissions** sub-tab.

5. Click on the arrow to the left of the resource to display the features of that resource.



6. Locate the feature resource(s) you want to update.



7. Click the checkbox(es) in the **allow** and **deny** columns as appropriate.

8. The changes are automatically saved. Close the tab(s).

   Note: You must follow this process for each user or group(s) separately.

SMC Feature Permissions

The following table describes the level of access required to be able to use these HPCC Systems ECL Watch features.

Name	Description	Access
ClusterTopologyAccess	Access to Cluster Topology	Read
	Access to log files.	Full
DfuAccess	Access to DFU Logical Files	Read
	Delete Files, add to superfiles, and remove from superfiles	Write
	Erase file history metadata	Full
DfuExceptions	Access to DFU Exceptions	Read
DfuWorkunitsAccess	Access to View DFU Workunits	Read
	Access to Create, Delete, Update, Submit, and Abort DFU Workunits	Write
DfuXrefAccess	Access to DFU XREF	Read
	Clean directory	Write
	Make changes and generate XREF Reports	Full
EclDirectAccess	Access to ECL direct service.	Full
ESDLConfigAccess	ESDL Config Access	Read
	Publish ESDL definition and ESDL binding, configure ESDL binding method.	Write
	Delete ESDL definitions, delete ESDL bindings.	Full
FileDesprayAccess	Allows a user to despray logical files.	Write
FileIOAccess	Access to read files in Drop zone	Read
	Access to write to files in Drop zone	Write
PackageMapAccess	Access to ListPackage, ListPackages, GetPackage, GetPackageMapById, ValidatePackage, GetQueryFileMapping, GetPackageMapSelectOptions, GetPartFromPackageMap	Read
	Access to AddPackage, CopyPackageMap, ActivatePackage, DeActivatePackage, AddPartToPackageMap, RemovePartFromPackageMap	Write
	DeletePackage	Full
FileScopeAccess	Allows access to query, set, modify, and delete File Scope Permissions	Full
FileSprayAccess	Access to Spraying and Copying	Read
	Rename, spray, copy, and replicate files	Write
	Download or delete file on a landing zone	Full
MachineInfoAccess	Access to machine/Preflight Information	Read
MetricsAccess	Access to SNMP Metrics Information (Roxie Metrics)	Read
OthersWorkunitsAccess	Access to View Other User's Workunits	Read

Name	Description	Access
	Access to Modify or Resubmit User's Workunits	Write
	Access to Delete Other User's Workunits	Full
OwnWorkunitsAccess	Access to View Own Workunit	Read
	Access to Create or Modify Own Workunit	Write
	Access to Delete Own Workunits	Full
RoxieControlAccess	Access to Roxie control commands	Read
SmcAccess	Access to ECL Watch (SMC Service)	Read
ThorQueueAccess	Access to Thor Job Queue Control	Full
CodeSignAccess	Access to Code Signing service ListUserIDs	Read
	Sign code	Full
WsELKAccess	Access to ELK integration service	Access
	Read the ELK configuration	Read
WsStoreAccess	Access to WsStore service	Access
	List stores, fetch key-value pairs, listkeys, listnamespaces	Read
	Set key-value pairs	Write
	Delete keys, delete namespaces, fetch keymetadata	Full
WsEclAccess	Access to WS ECL service	Full
WsLogAccess	Allows ability to read component logs	Read
SashaAccess	Access to WsSasha service	Access
	List Workunits	Read
	Archive Workunits, restore archived Workunits	Full

Some Feature Permissions Notes

- SMCAccess is required to be able to successfully login to ECL Watch.
- ThorQueueAccess allows you to manipulate the queue by promoting/demoting queued workunits according to priority.
- ThorQueueAccess also allows you to pause or clear the Thor queue. You can also view Thor usage statistics.
- Depending on the level of access the user has, they can view, modify, and delete their own, or others workunits. This is OwnWorkunitsAccess, and OthersWorkunitsAccess respectively.
- DfuWorkunitsAccess permissions allow users to view and/or manipulate DFU Workunits.
- Users need permission to see files on the dropzone and also to put files there. They need further permissions to be able to spray and copy files from the dropzone to their cluster and also to despray files from the cluster back to the dropzone.
- The WsStore service uses **namespaces** (similar to a database in a DBMS system), **stores** (similar to tables in a database), and **key-value pairs** (similar to fields).

DFU Xref

XREF is used for monitoring files on the cluster(s). Reports generated show where housekeeping is required on the cluster(s) and users require additional permission to use this feature.

	On a large system, we suggest limiting the number of users who can Generate XREF reports by setting DfuXrefAccess access to FULL for only those users.

Users/Permissions

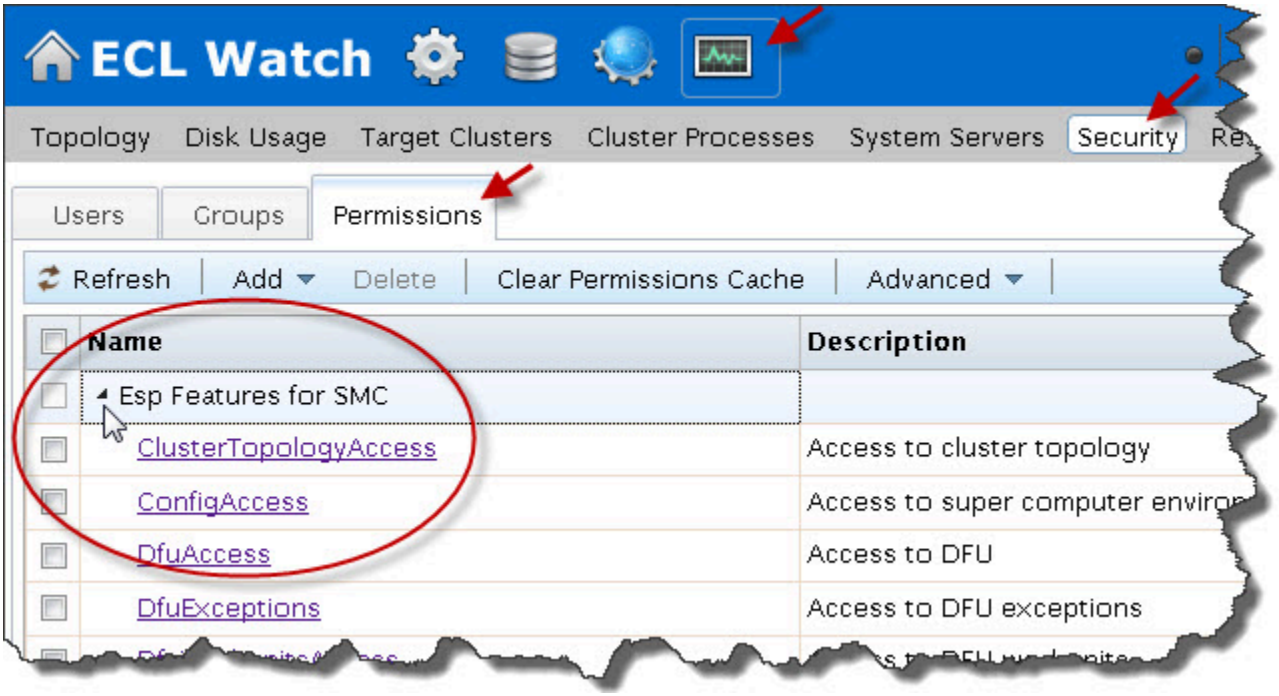
To be able to view the **Users/Permissions** area in ECL Watch, a user must be a member of the Administrators (or similarly named) group with the appropriate permissions on the LDAP or Active Directory server.

File Access Control

The HPCC Systems LDAP **Dali Server** technology provides the ability to set secure access permissions to data file folders (or file scopes). This is controlled by the use of file scope resources.

An OU called **Files** is automatically created when the Dali server starts. To secure data folders, create a file scope for that folder and apply rights to each scope.

Figure 29. File Scopes Permissions



For example, below **Files** there is a unit (OU) representing the cluster, such as **thor** (or the name that you set up for your cluster). Furthering the example, below that could be a unit named **collectionx** which contains two units, **publicdata** and **securedata.** The **publicdata** folder has rights granted to a large group of users and the **securedata** folder has limited access granted. This allows you to prevent unauthorized users from any access to files in the **securedata** folder.

The structure described above corresponds to this logical structure:

collectionx::securedata

Which corresponds to this physical structure:

/var/lib/HPCCSystems/hpcc-data/thor/collectionx/securedata

All HPCC Systems components and tools respect LDAP file access security. The following exceptions are assumed to be system level or for administrative users:

- Network file access using UNCs, Terminal Services, or SSH.
- Administrative utilities

Attempting to access a file in a folder for which access is not granted will result in one of the following errors:

```
DFS Exception: 4 Create access denied for scope <filepath>
```

or

```
DFS Exception: 3 Lookup access denied for scope <filepath>
```

(where <filepath> is the full logical file scope path)

Creating File Scopes

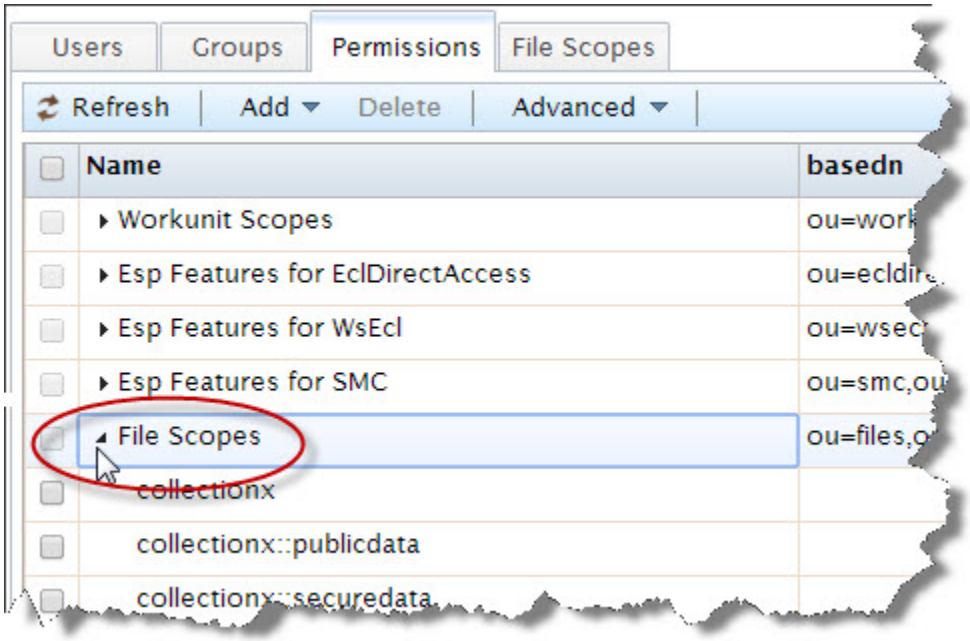
To apply permissions to a file scope, you must first create the file scope(s).

To create file scope(s) click on the **Operations** icon, then click the **Security** link from the navigation sub-menu.

1. Click the **Permissions** tab.

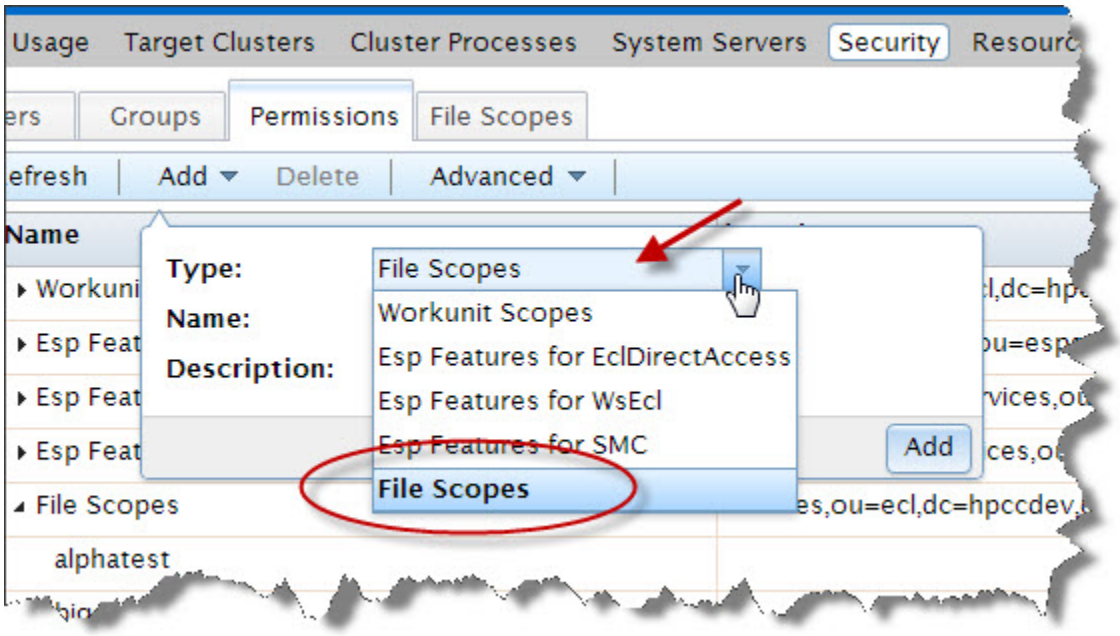
   The feature resources display.

2. Click on the arrow to the left of the **File Scopes** resource to display the file scopes.

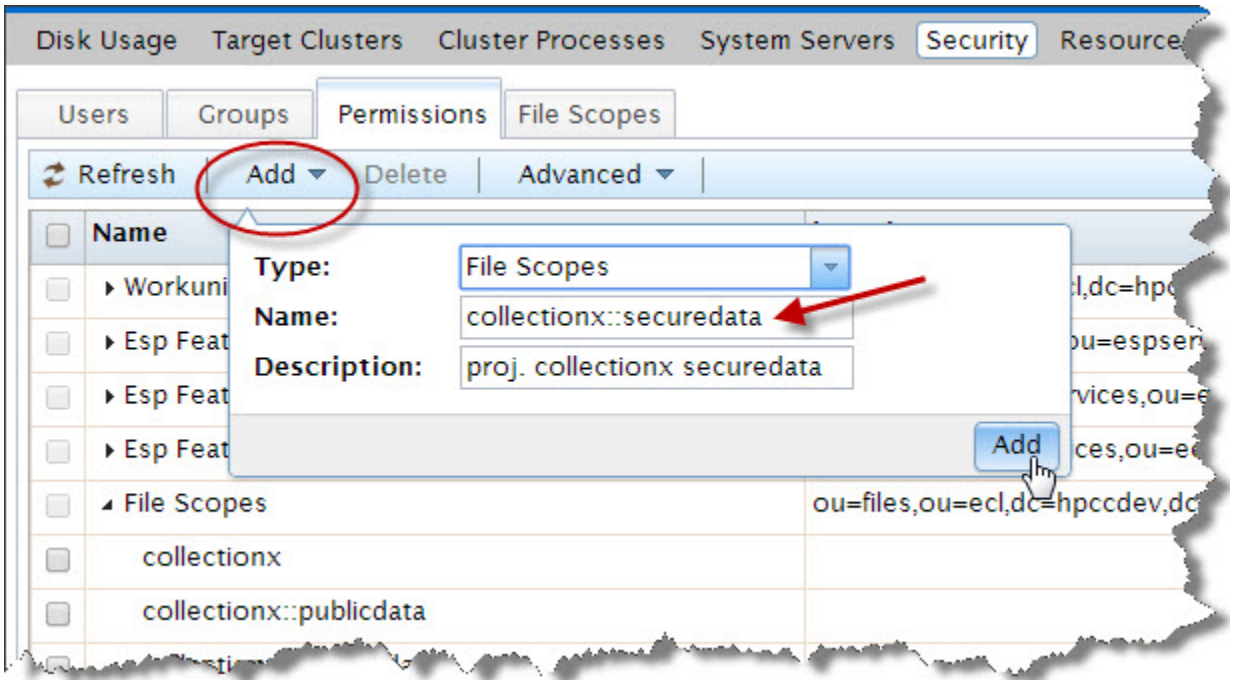


3. Press the **Add** button.

4. Choose **File Scopes** from the drop list.



5. Enter the exact name of the scope you want to add in the **Name** field.



Enter a short description in the **Description field.**

6. Press the **Add** button.

The new scope displays in the list.

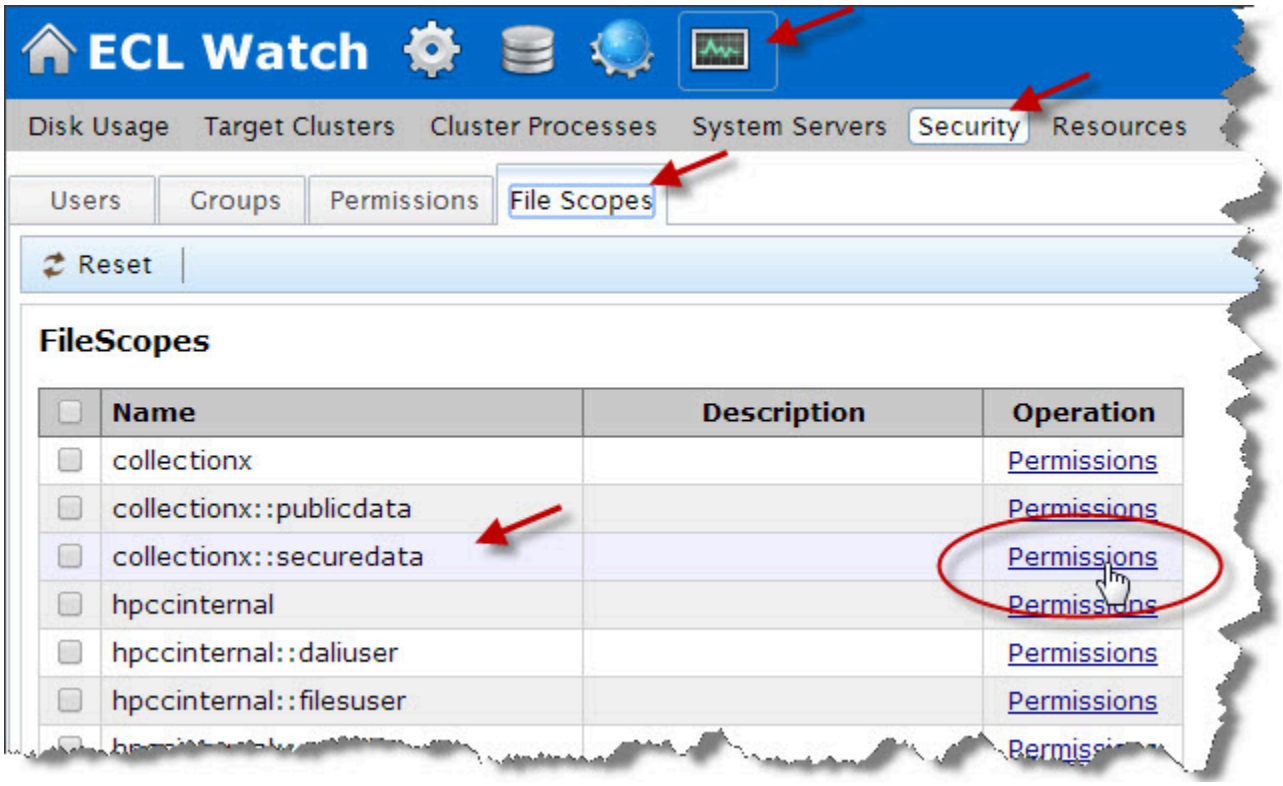
Setting permissions for file scopes

You must apply permissions for file scopes to users or group(s). If you want to apply the scope to a new group, create the group(s) as required.

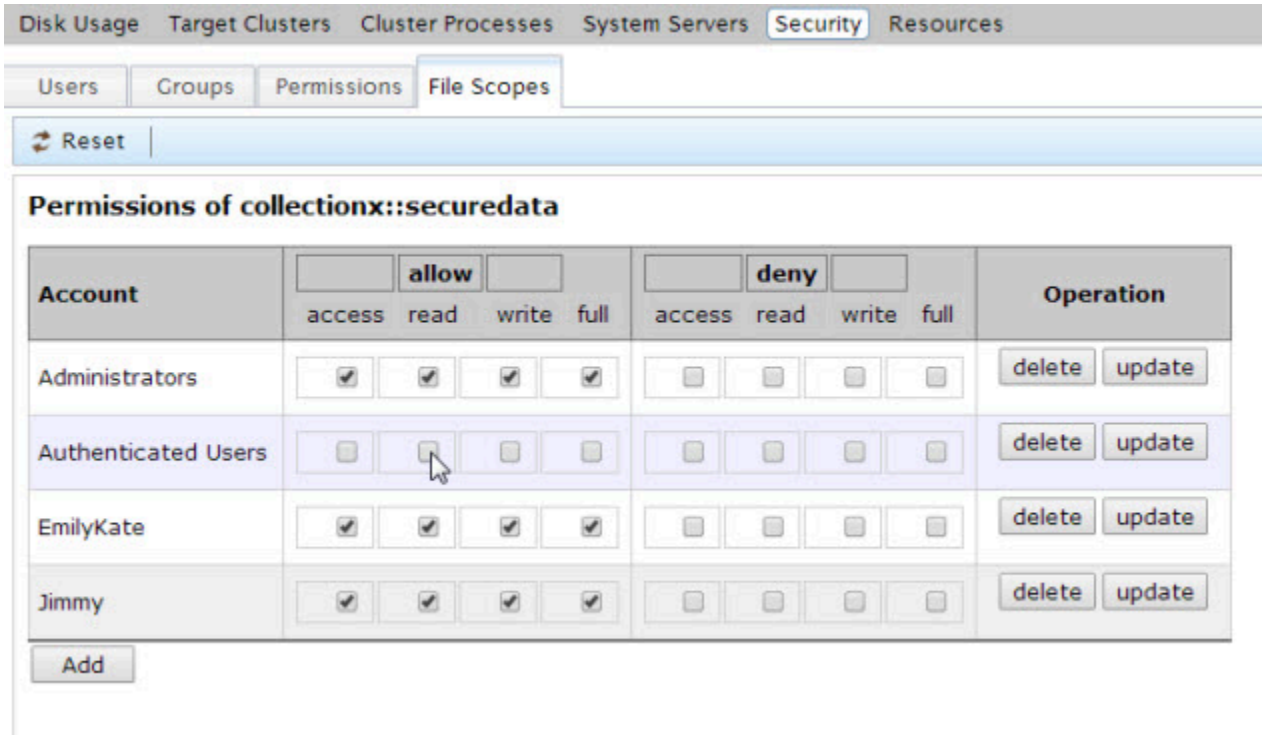
To set the file scope permissions click on the **Operations** icon, then click the **Security** link from the navigation sub-menu.

1. Select the **File Scopes** tab.

2. Choose the scope to modify. Click the **Permissions** link for that scope.



3. The permissions defined for users and groups for that scope display.

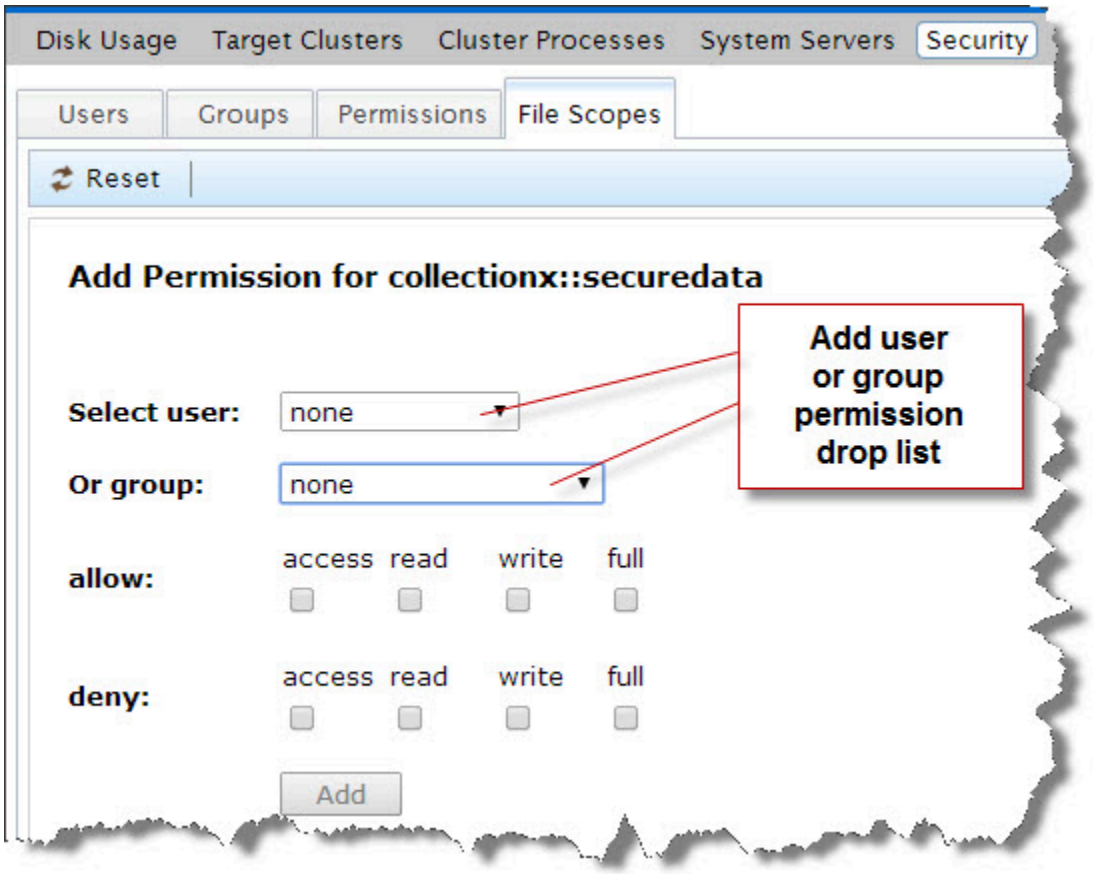


4. Check (or clear) the checkbox(es) in the **allow** and **deny** columns as appropriate for the users or groups displayed.

5. To add users or groups to the scope, press the **Add** button.

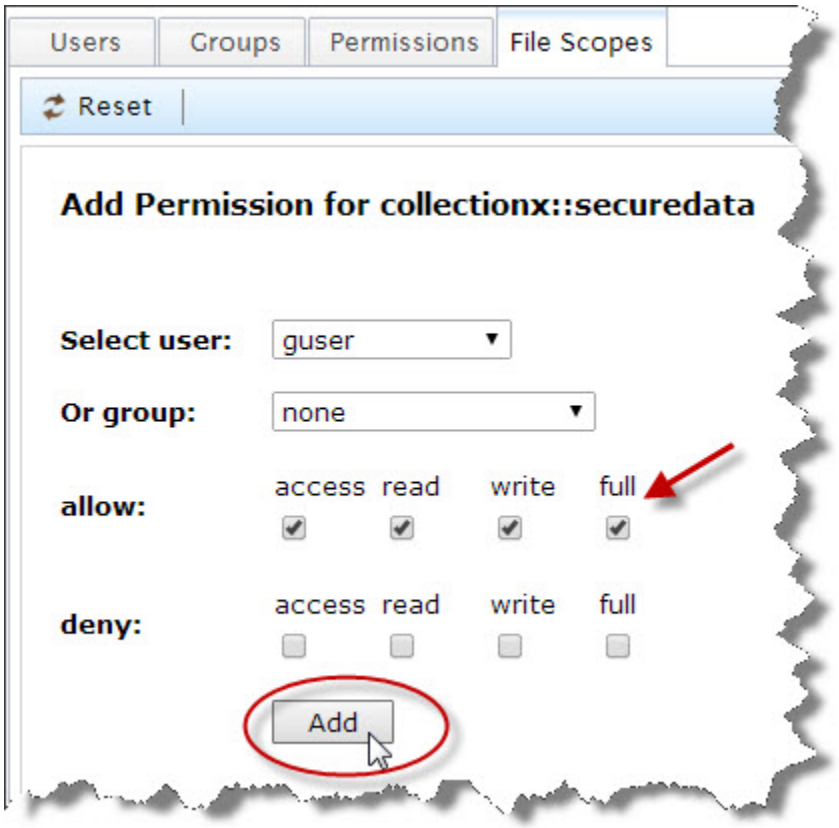
   The Add Permission dialog displays.

6. Select the user or the group to add from the drop list(s).



Once a user or group is selected, the Add button and the allow and deny checkboxes are active

7. Check the boxes for allow and deny as appropriate to set the permissions for this scope.



8. Press the **Add** button.

9. The changes are automatically saved. Close the tab(s).

File scope features

Below the List of File Scopes, there are buttons that allow you to:

- Reset **Default Permissions** to selected file(s)

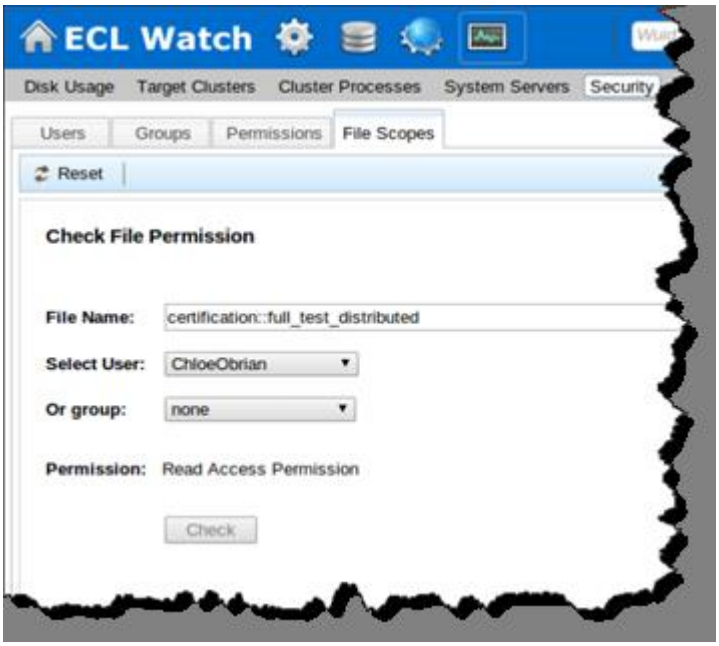
  This allows you to quickly remove any added permission settings for a file and reset to the default access.

- Allow or Deny Access to physical files on Landing Zone

  This provides a way to grant or deny access to the top level file scope. By default, only administrators have access to this scope.

- Check File Permissions for a user or group

  This provides a way to check a user or group's access to a logical file.



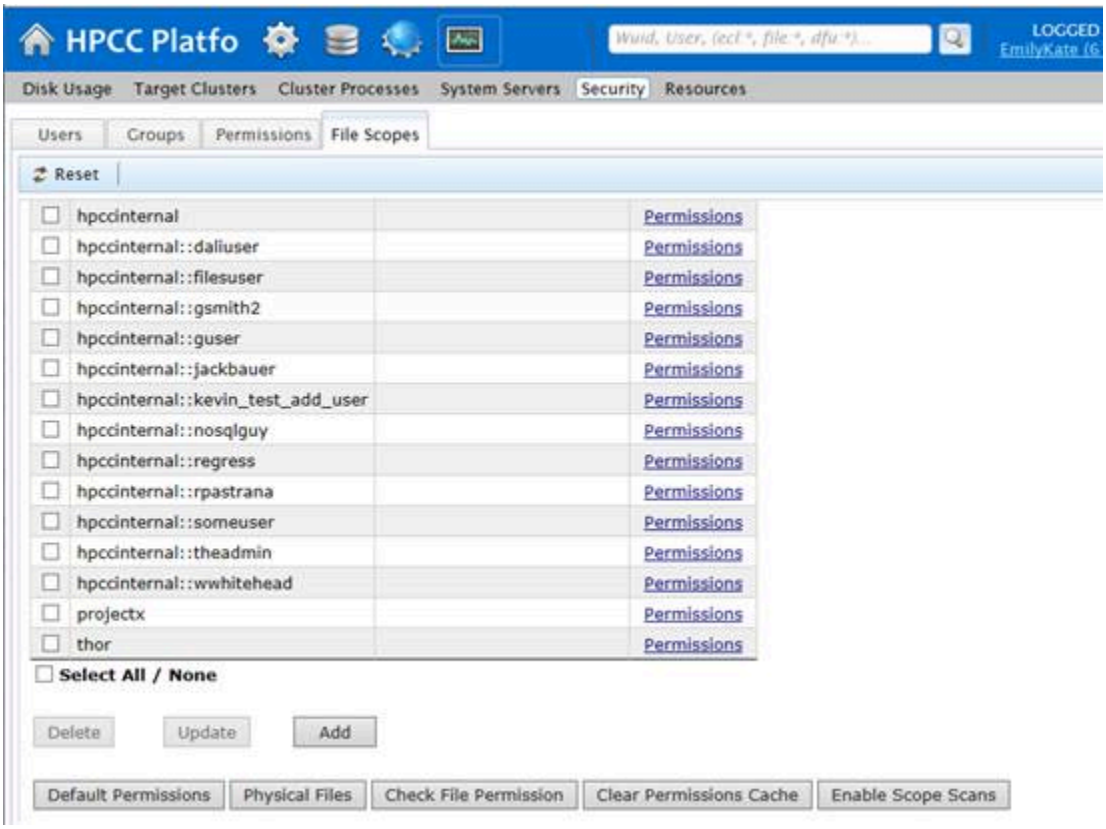
- Clear the Permissions Cache

  This clears the permissions cache and allowing any new permission settings to take effect immediately.

- Enable/Disable Scope Scans

  This provides a means to enable or disable Scope scans. Enable scope scans to check permissions for users to access scopes. This will impact performance. Disable scope scans ignores any scope permissions and removes all access control, but improves performance. Disabling access control is not recommended.

  Changing this setting through ECL Watch, as described here, is only a temporary override. When Dali restarts this setting will revert to what is defined in the configuration environment.xml.



Landing Zone Security

You can set additional security options on Landing Zone(s). Feature level security allows you to set permissions on access to your Landing Zone and what users or groups can do there. Landing Zone Scope Security allows you to set permissions on sub-folders in a Landing Zone. This provides a means to grant and deny users permission to areas within a Landing Zone.

Landing Zone Feature Authorization

This lists the HPCC Systems Landing Zone using Feature Level Authorization:

List/search Dropzone files	FileSprayAccess - SecAccess_Read
Spray a file from a Dropzone	FileSprayAccess - SecAccess_Write
Despray a file to a Dropzone	FileDesprayAccess - SecAccess_Write
Read the content of a Dropzone file	FileIOAccess - SecAccess_Read
Write the content of a Dropzone file	FileIOAccess - SecAccess_Write
Upload a file to a Dropzone using ECLWatch:	FileUploadAccess - SecAccess_Full
Download a file from a Dropzone using ECLWatch	FileSprayAccess - SecAccess_Full

To enable access to a feature, set the permission accordingly.

This may be sufficient level security in some cases, however, additional restrictions may be needed to secure certain files, from certain users or groups. You can use Landing Zone File Scope security to accomplish this. .

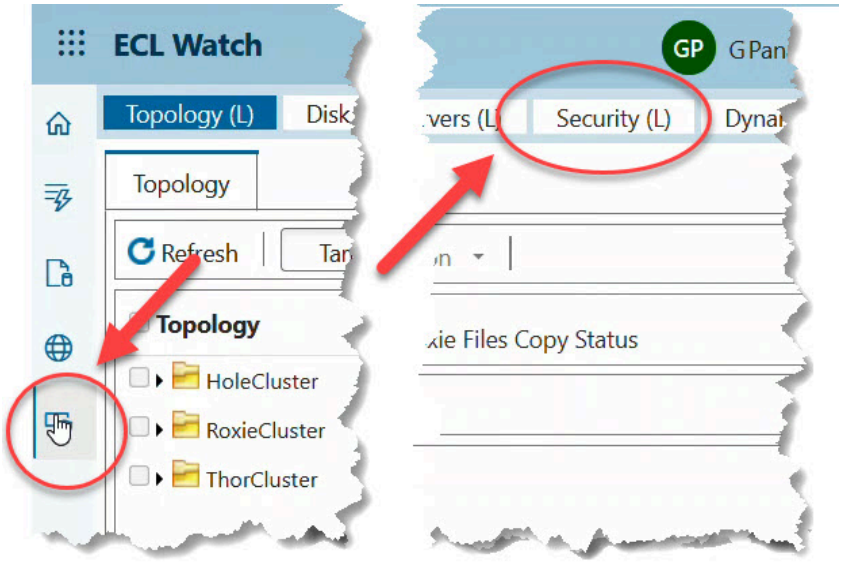
Landing Zone File Scopes

File Scope Level Authorization provides a means to secure access to folders within a Landing Zone.

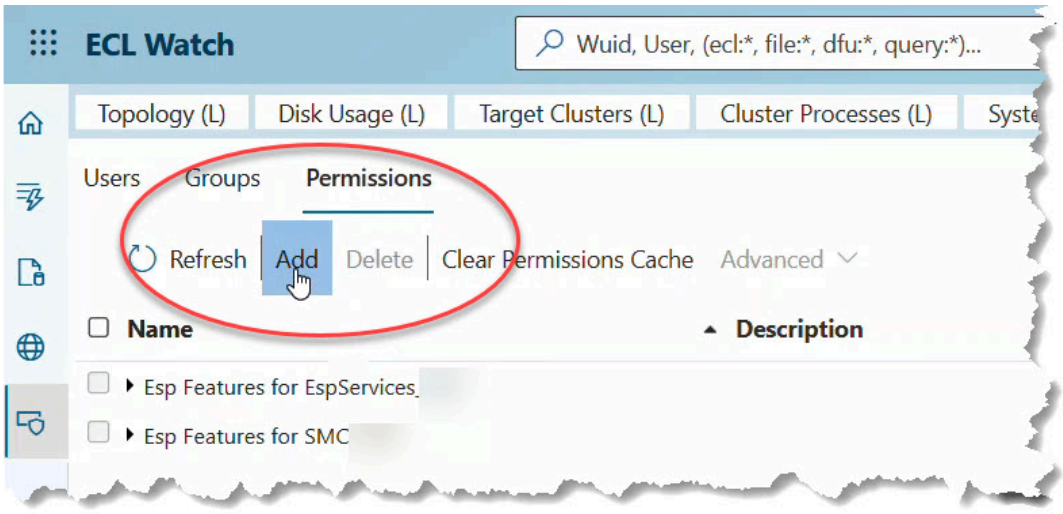
An HPCC Administrator can define the Landing Zone scopes for each folder in an HPCC Landing Zone.

Each scope is a file folder of an HPCC Landing Zone. Each Landing Zone scope is one HPCC file scope.

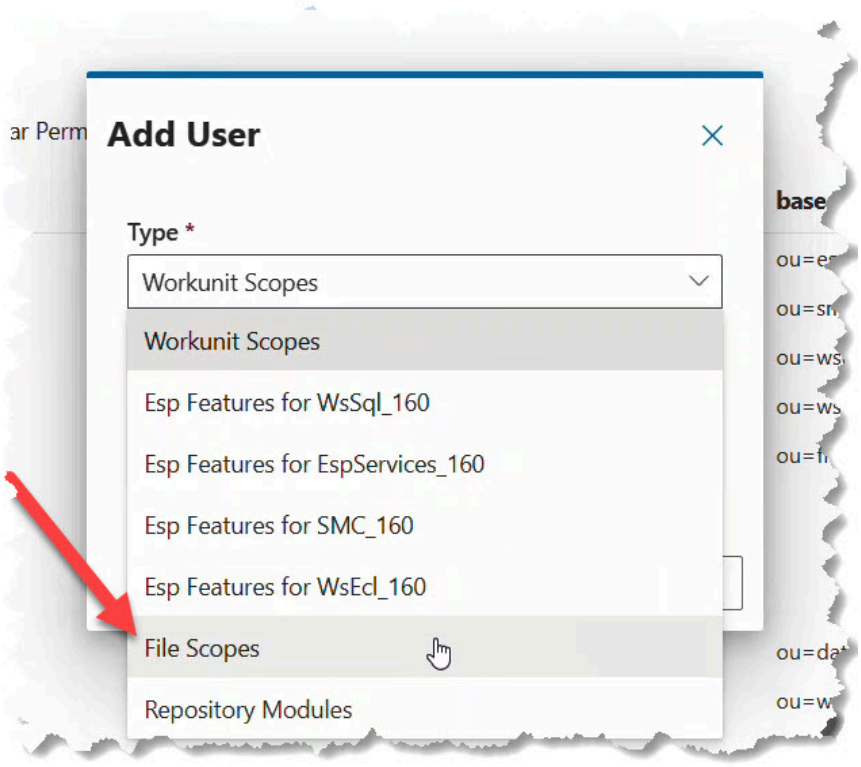
The Landing Zone file scopes can be defined using ECLWatch for security enabled systems.



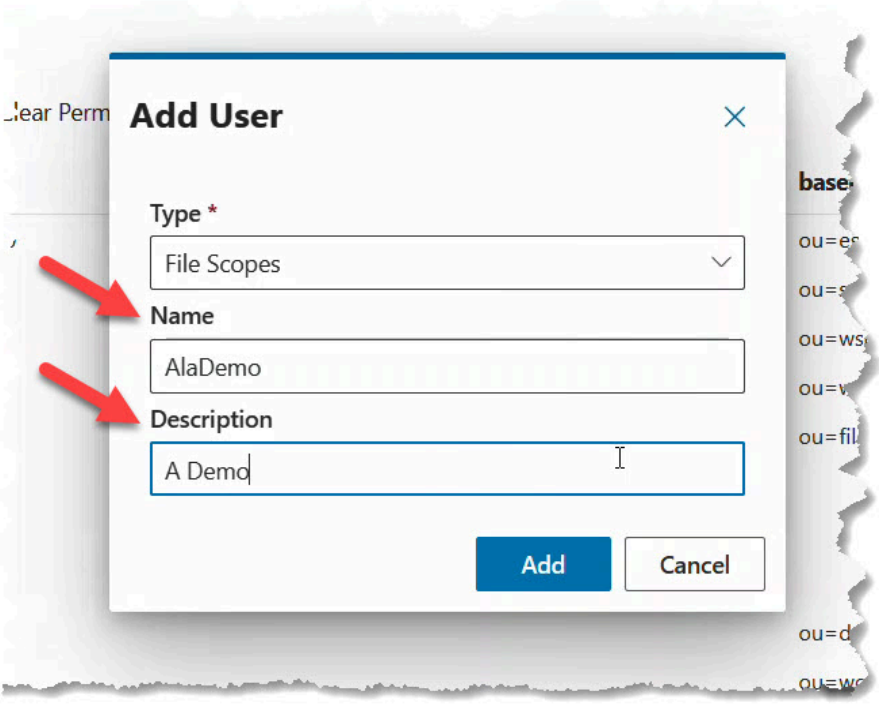
To create a new Landing Zone scope, go to the Security page of ECL Watch, and click on Permissions.



On the Permissions tab press the Add button.



Choose File Scopes on the drop down option box, then provide a name and optionally a description.



Landing Zone File Permissions

You can set the Landing Zone file permissions according to your requirements. Access your new Landing Zone using the following annotation:

```
plane::{dropzone_name}::{folder_name}::{subfolder_name}::{subfolder_name}...
```

Your HPCC Administrator can define access rights to each Landing Zone scope for each HPCC user or user group.

Workunit Access Control

There are 2 aspects of workunit (WU) security:

- Feature Authentication for workunits allows you to set permissions to control whether users can view their own WUs and/or other users' WUs.
- Workunit Scope security provides the ability to set permissions for individual WU scopes. All new workunits have a scope value.

Both methods are valid to use (either separately or together), and the strictest restriction always applys.

In other words, if someone is granted permission to see WUs in the scope *johndoe* but is denied permission to see other users' WUs in the Feature Authentication permissions, this user would be denied access to see the WUs in the *johndoe* scope.

Conversely, if the user is allowed access to see other people's WUs but is denied access to the *johndoe* WU scope, this user will be able to see other WUs in that scope.

Note: If you do not have access to a WU, you will never be able to view it or even know of its existence.

By default, a submitted WU has a scope of the user's ID. For example, a WU JohnDoe submits has *scope=johndoe* in the WU. This value in a WU allows ESP and its services to use LDAP to check for permissions and enforce those permissions.

You can override the default scope using ECL Code:

```
#workunit('scope','MyScopeValue');
```

Securing workunit scopes

ESP (on startup) automatically creates an LDAP OU called **Workunits** (unless it already exists). If this OU is automatically created, the OU is made with full permissions granted to all authenticated users. All WU scopes are below the *workunits* OU either implicitly or explicitly.

If a specific scope OU does not exist in LDAP (e.g., the scope johndoe used in earlier example), then the parent OU's permissions are used. In other words, the scope of *johndoe* is implicitly under the *workunits* OU even though it might not be explicitly listed in the LDAP structure and therefore it would use the permissions granted for the parent, *workunits* .

Workunits feature permissions

Using the **Workunit Scopes** feature in the **Permissions** area of ECL Watch the permissions for any scope can be reset to the default permissions settings for your system. Permission settings for Workunit Scopes may be set as follows:

Description	Access
View WUs in that scope	Read
Create/modify a WU in that scope	Write
Delete a WU in that scope	Full

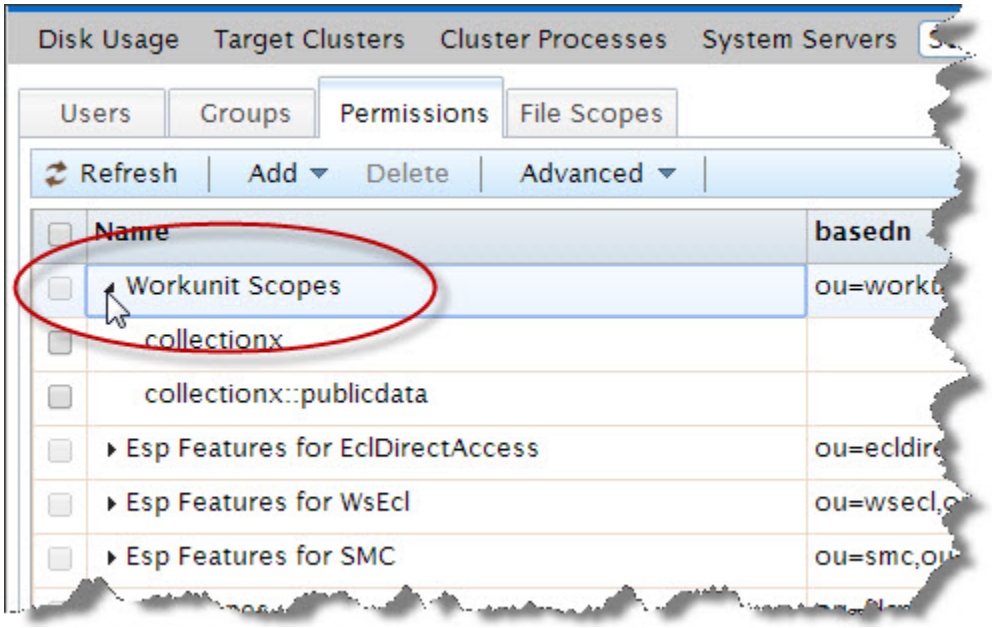
Adding workunit scopes

To add workunit scope permissions click on the **Operations** icon, then click the **Security** link from the navigation sub-menu.

1. Click the **Permissions** tab.

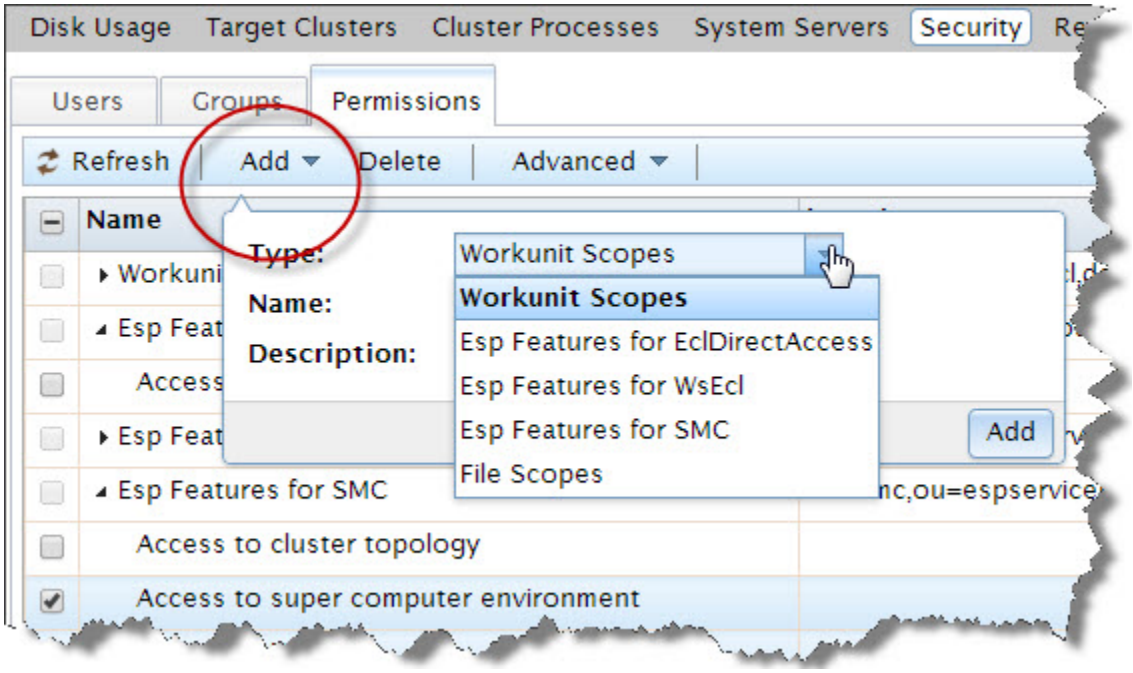
   The feature resources display.

2. Click on the arrow to the left of the **Workunit Scopes** resource to display the file scopes.

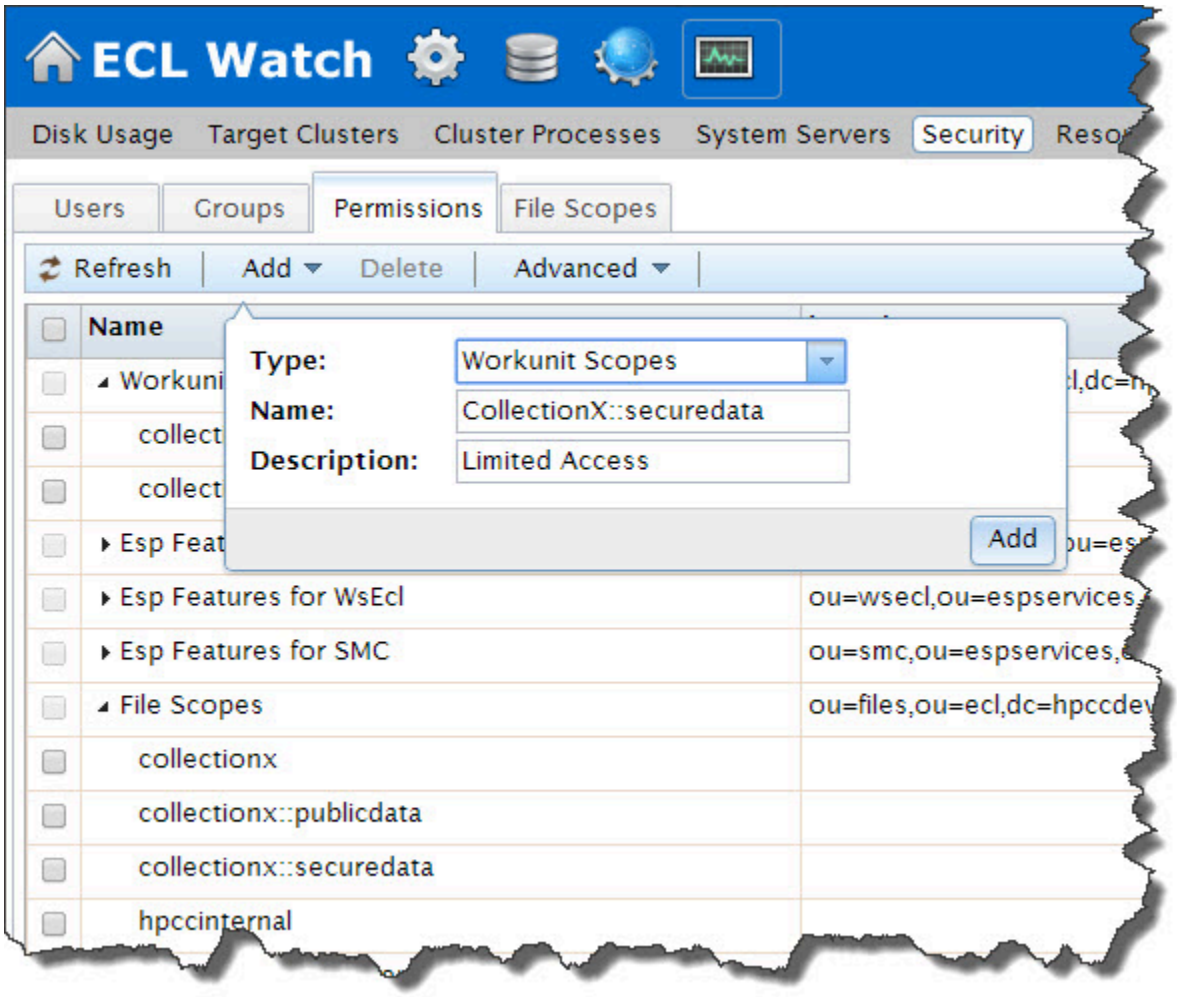


3. Press the **Add** button.

4. Choose **Workunit Scopes** from the drop list.



5. Enter the exact name of the scope you want to add in the **Name** field.



Enter a short description in the **Description field.**

6. Press the **Add** button.

   The new scope displays in the list.

Set permissions to the scope.

You apply the workunit scopes to a group. If you want to apply the scope to a new group, create the group(s) as required.

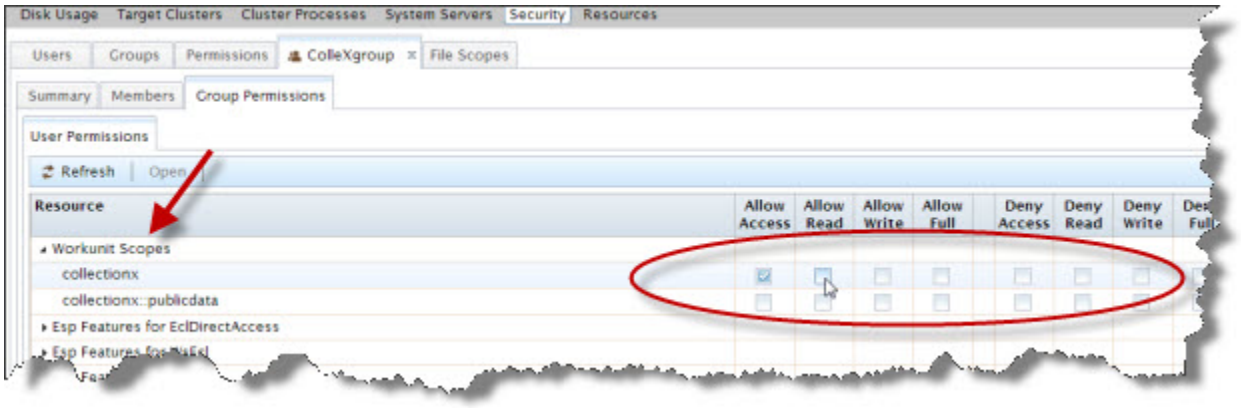
1. Go to the **Groups** tab.

2. Select a group to apply the scope to by checking the box next to the group name.

   Press the **Open** action button. You can select multiple groups, a tab opens for each group.

3. Select the **Group Permissions** tab of that group. (if multiple groups selected, you must repeat for each group)

4. Click on the arrow to the left of the Workunit Scopes to display the available scopes.



The Workunit scopes display. Check the boxes as appropriate to set the permissions for this scope.

5. To set permissions in this scope for another group, open and go to that groups tab.
6. To set permissions in this scope for a user, select the tab.
7. Select the user and press the Edit action button.

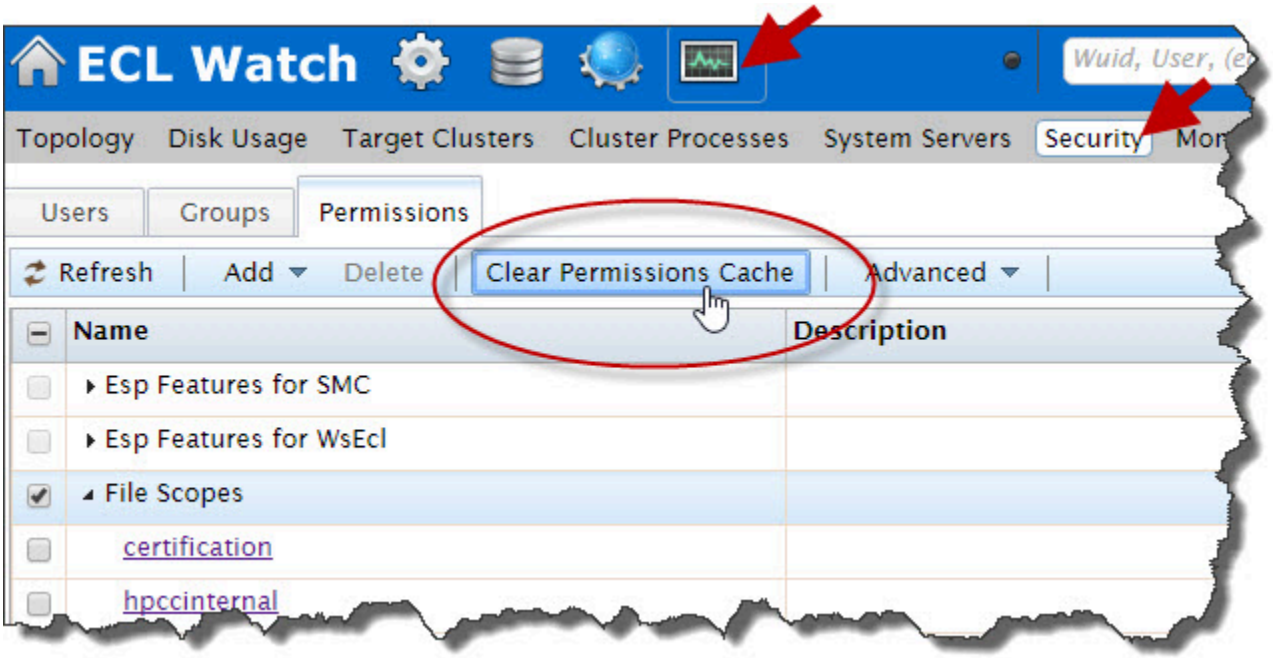
   A new tab for that user opens.
8. On that tab, click on the **User Permissions** sub-tab.
9. Locate the new scope listed under the appropriate Resource.

   Set the access permissions as appropriate for that user.
10. The changes are automatically saved. Close the tab(s).

Permission Caching

A helpful feature found on the Permissions tab is the *Clear Permissions Cache* button. The *Clear Permissions Cache* button clears the cached permissions from Dali and ESP.

When you change a permission in ECL Watch, the settings are cached in the ESP server and stored in the Dali server. The information in the cache is updated at a configurable interval. This value can be set in the Configuration Manager under the *LDAP Server settings Attributes* tab. The default cacheTimeout is 5 minutes.

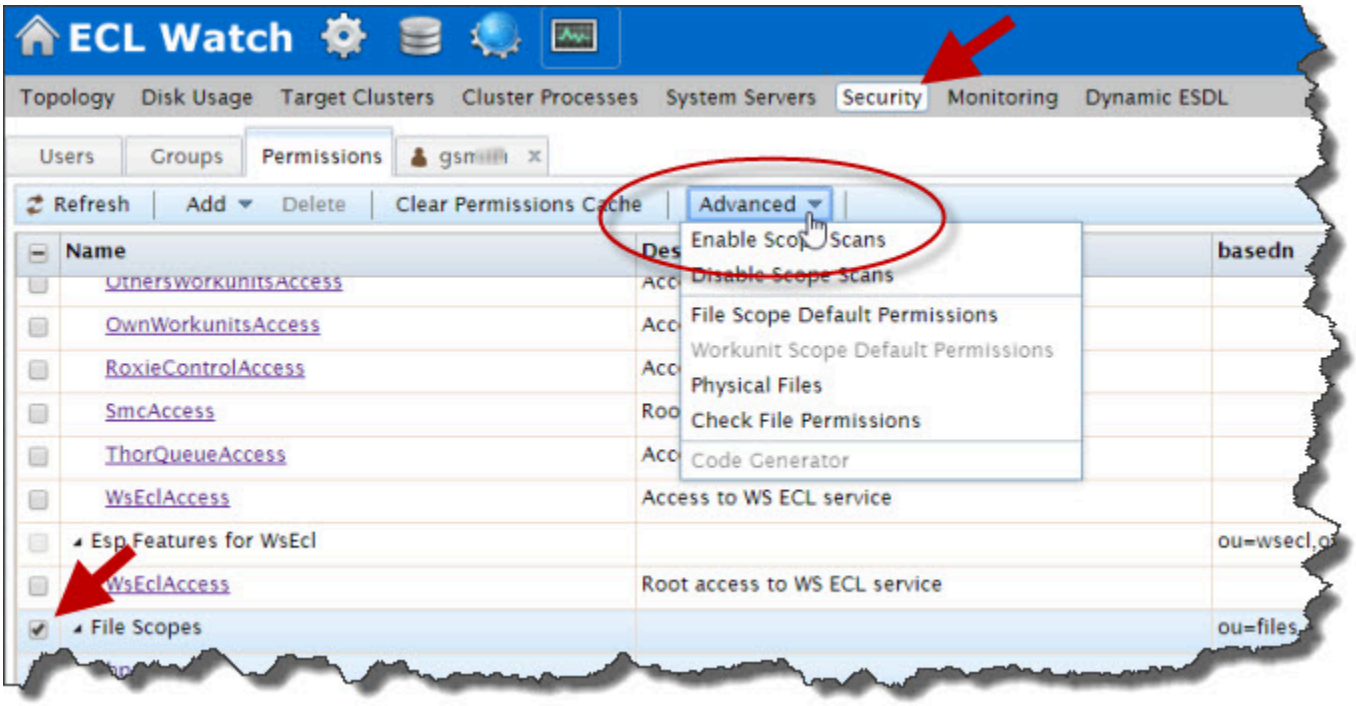


The Permissions Cache can be cleared from anywhere on the permissions tab in ECL Watch.

When you want a permission change to take effect immediately, you can clear the cache and force Dali to update the permission settings by pressing the **Clear Permissions Cache** button. This action transfers the settings when you press the button. Use this feature judiciously as overall system performance is affected temporarily while the LDAP settings in the Dali System Data Store repopulate.

Advanced Permissions

On the Permissions tab is the **Advanced** (Permissions) button. The Advanced menu/button provides access to manage file and workunit scope security. The Advanced button is only enabled when you select either File or Workunit Scopes on the Permissions tab.

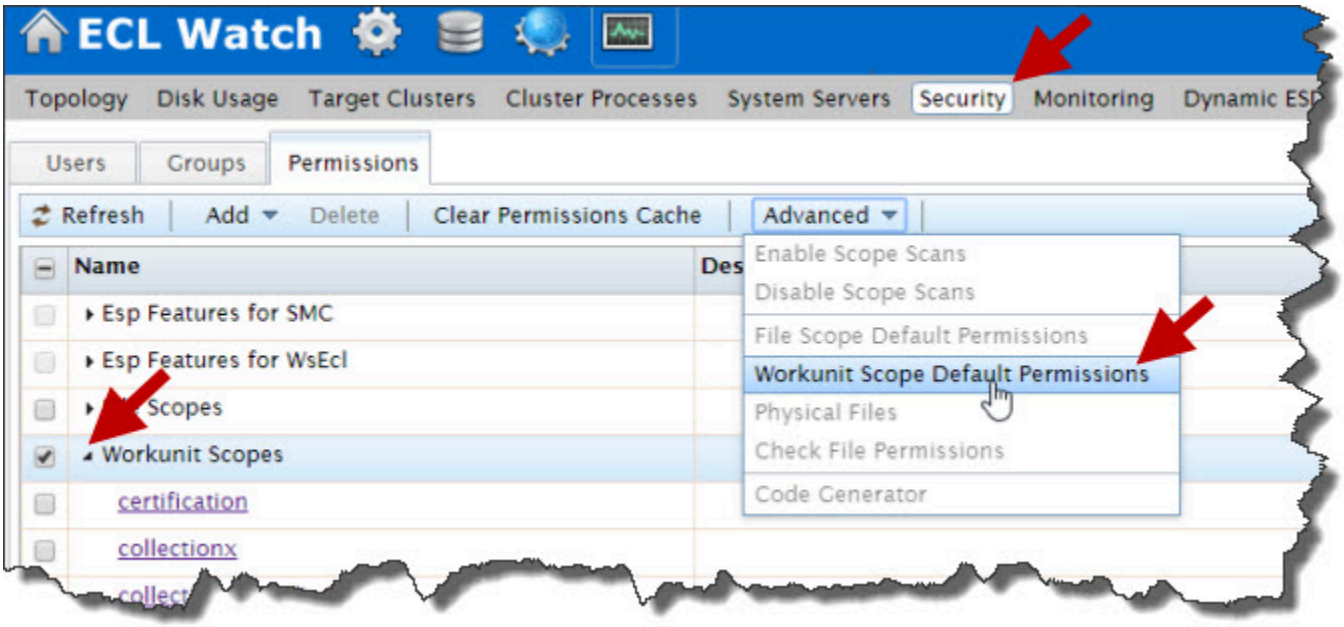


Press the Advanced button to display the Advanced permissions menu. The Advanced menu is context aware, so if you select File Scopes from the Permissions tab, then you can only choose to apply the relevant file scope permissions, likewise if you selected workunits scopes.

File Scans

Using the Advanced Menu with File Scopes selected:

- Enable or disable file scope security
- Access the file scope default permission page
- Access the Physical Files permission page. The Physical Files settings specify the default permissions for files that do not have a scope explicitly specified.
- Check the file permissions - This option opens a dialog where you can input a File name and select Users and Groups for the File security scope.



Workunit scans

Using the Advanced Menu with the Workunit Scopes selected opens only the Default Permissions tab for the Workunit Scope (Default) Permissions.

NOTE: File or Workunit Scope security needs to be enabled in your system configuration in order to use File or Workunit Scope security on your system.

Dali and Security

This section contains additional information about Dali and security settings.

LDAP and Dali Security settings

There are a few Dali security settings that impact the way that Dali performs. The following configurations further explain the impact of some common security settings.

Dali without LDAP

If you configure Dali without any LDAP server bound:

- Anyone can access any file and any workunit. Essentially, you have no security.
- Without an LDAP server configured, the CheckScopeScan attribute in the configuration is ignored. This means that any user can see the entire list of logical files and can access any file.
- Anyone can see, access, or even run any workunit.

Dali with LDAP Server and CheckScopeScans setting disabled

In this scenario, you have your Dali bound to a LDAP server, and the CheckScopeScans attribute is set to false

- The CheckScopeScans setting only impacts the listing of logical files.
- All file access calls are authorized by ensuring the caller has access to the given file scope.
- FilesDefaultUser credentials are injected if none provided. The *filesDefaultUser* is an LDAP account used to access files when no user credentials are supplied. This is similar to a guest account, so it should be an account with very limited access, if used at all. To disable access without credentials, leave filesDefaultUser attribute blank.
- Since the CheckScopeScans setting is disabled, users can see the entire list of logical files, even if user does not have access to view a file.

Dali with LDAP Server and CheckScopeScans setting enabled

In this scenario, you have your Dali bound to a LDAP server, and the CheckScopeScans attribute is set to true.

- All file access calls are authorized by ensuring the caller has access to the given file scope.
- Users can see files listed for all files within file scopes for which they have read access permission.

You should keep in mind since the system must make an external LDAP call to check every level in the scope, from the top to the bottom, that the depth of file scopes can have a performance cost in systems with File Scope Security enabled.

The AllowList in Dali

The Dali server has the ability to restrict access to only those nodes that are associated with a role in the environment definition (environment.xml) or explicitly added to the *AllowList* in Dali's configuration. The AllowList is implicitly populated with the server components and their roles as defined in the environment. You can explicitly add additional nodes and their approved role(s) to a supplementary AllowList in the environment.xml file as shown in the following example:

```
<Environment>
...
 <Software>
 ...
   <DaliServerProcess>
     <AllowList>
      <Entry hosts="adminnode1,192.168.0.101" roles="DaliDiag,DaliAdmin"/>
      <Entry hosts="adminnode3" roles="DaliDiag"/>
     </AllowList>
   ...
   </DaliServerProcess>
 ...
 </Software>
...
</Environment>
```

Hosts can be specified by hostname or IP address.

Roles must be one of the following supported roles:

ThorMaster	SashaServer	DaliAdmin
EclCCServer	DfuServer	UpdateEnv
EclCC	EspServer	TreeView
EclServer	Config	DaliDiag
EclScheduler	SchedulerAdmin	Testing
EclAgent	RoxieMaster	XRef
AgentExec	BackupGen	Monitoring
DaliServer	DaFsControl	
DaliStop	SwapNode	

An entry can have one or more nodes (hosts that require access), specified in a comma separated list. Nodes can have one or more roles, specified in a comma separated list. Roles are case-sensitive.

It should never be necessary to explicitly add the following roles for nodes within the same environment :

```
AgentExec, DaliServer, EclAgent, EclCC, EclCCServer, EclServer, EspServer,
SashaServer, RoxieMaster, and ThorMaster
```

These roles correspond to components in the environment, which should always be implicitly allowlisted by inclusion in the environment. If nodes in a foreign environment need to communicate with Dali, then those nodes should be explicitly added.

In addition, the following administrative tool roles are automatically allowlisted to run on the Dali server node:

```
Config, DaFsControl, DaliAdmin, DaliDiag, ScheduleAdmin, SwapNode, Testing,
TreeView, UpdateEnv, XRef, Monitoring
```

The only roles that are likely to be added are administrative tools and utilities that are not defined components, for example:

```
Config, DaFsControl, DaliAdmin, DaliDiag, SchedulerAdmin, SwapNode, Testing,
TreeView, UpdateEnv, XRef
```

To disable the AllowList feature entirely, you can add:

```
<AllowList enabled="false"/>
```

This is not recommended for production environments.

When enabling the Dali AllowList on a cluster where there are other external environments interacting, it can be useful to initially disable the AllowList and audit the external components that are trying to connect with it by viewing the DaServer log files. Each client that would be refused access if the AllowList was enabled, creates log entries similar to this:

```
00000017 Operator 2019-11-20 16:58:39.617 17056 17074
         "AllowList is disabled, ignoring: Access denied!
         [client ip=192.168.9.12, role=DaliDiag] not AllowListed"
```

To allow legacy components that do not provide a role to connect, you can specify the allowAnonRoles option. The component MUST be on an allowlisted IP.

```
<AllowList allowAnonRoles="false"/>
```

Retrieve the AllowList

To retrieve the entire AllowList (implicit and explicit), use the **dalidiag** command line tool

(found in /opt/HPCCSystems/bin/)

```
dalidiag <dali-ip> -AllowList
```

Update Dali without restarting

To update AllowList information in Dali without restarting, use the **updtdalienv** command line tool

(found in /opt/HPCCSystems/bin/)

```
updtdalienv <environment-xml-file> -i <dali-ip>
```

Use envmod to add or remove entries in the supplementary AllowList

We recommend using the **envmod** command line tool to add AllowList entries to your environment.xml file. The envmod utility can be found in /opt/HPCCSystems/bin/.

Use a template file such as the following example:

```
{
    "name" : "AddAllowList",
    "description" : "Add AllowList to environment",
    "type" : "modification",
    "operations" : [
        {
            "action" : "find",
            "target_path" : "/Environment/Software/DaliServerProcess/AllowList",
            "data" : {
                "create_if_not_found" : true,
                "save" : {
                    "name" : "AllowListNodeId"
                }
            }
        },
        {
            "action" : "create",
            "target_nodeid" : "{{AllowListNodeId}}",
            "data" : {
                "node_type" : "Entry",
                "attributes" : [
                    {
                        "name" : "hosts",
                        "value" : "adminnode1,192.168.0.101"
                    },
                    {
                        "name" : "roles",
                        "value" : "DaliDiag,DaliAdmin"
                    }
                ]
            }
        },
{
            "action" : "create",
            "target_nodeid" : "{{AllowListNodeId}}",
            "data" : {
                "node_type" : "Entry",
                "attributes" : [
                    {
                        "name" : "hosts",
                        "value" : "adminnode3"
                    },
                    {
                        "name" : "roles",
                        "value" : "DaliDiag"
                    }
                ]
            }
        }
    ]
}
```

This example adds the DaliDiag and DaliAdmin roles to two nodes. One node is specified using its hostname (adminnode1). The other is referenced by IP address. It adds the DaliDiag role to the adminnode3. The result matches the example XML shown earlier.

Sample command line:

```
sudo /opt/HPCCSystems/bin/envmod \
     -t myAllowListTemplate.json \
     -e /etc/HPCCSystems/source/environment.xml \
     -d /opt/HPCCSystems/componentfiles/configschema/xsd \
     -o /etc/HPCCSystems/source/environmentWithAllowList.xml
```

The -t (or --template) parameter is the location of the template.

The -e (or --env) parameter is the location of the environment file. If omitted, the action validates the template.

The -d (or --schema-dir) parameter is the location of the schema files.

The -o (or --output) parameter is the location of the output file. If you specify -o without a full path to a file, it overwrites the input file.

Once you have modified your environment, you MUST copy environment.xml to every node.

To remove an entry, use a template like the following:

```
{
    “action” : “delete”,
    “target_path” : “/Environment/Software/DaliServerProcess/AllowList/Entry[@hosts=’node3’]”
}
```

Initialization under Systemd

The primary tool for starting and stopping the HPCC Systems platform on a target machine using systemd is the *hpccsystems-platform.target*. This target script is dependent on the environment.xml file, and gets generated using the *generate-hpccsystems-target.sh* script that is located in /opt/HPCCSystems/sbin by default on standard installations. This target script must be generated any time you modify the configurations environment.xml file on your local machine. This can be done manually by an administrator, or by enabling the provided *hpcc-environment-monitor.path* systemd script.

Manual regeneration of hpccsystems-platform.target

The hpccsystems-platform.target script contains a requirement list for every instance of a component declared in the environment.xml file. The hpccsystems-platform.target can be manually generated by invoking the *generate-hpccsystems-target.sh* script on a target system. It is important to note that doing so will overwrite any existing hpccsystems-platform.target currently on the system. Thus any components that were previously declared in the environment.xml, which are no longer declared, cannot be started/stopped with the new hpccsystems-platform.target. To manually (or programatically through a script) update the hpccsystems-platform.target, run the following:

```
systemctl stop hpccsystems-platform.target
```

To properly shut down hpccsystems components before the hpccsystems-platform.target changes.

Automatic regeneration of hpccsystems-platform.target

The hpccsystems-platform.target script contains a requirement list for every instance of a component which is explicitly declared in the environment.xml file. In order to generate the required components list, invoke the *generate-hpccsystems-target.sh* script. This can be done automatically by enabling the hpcc-environemnt-monitor.path systemd script. The hpcc-environemnt-monitor.path script calls a corresponding service script of the same name whenever environment.xml is modified.

When you attempt to regenerate the hpccsystems-platform.target, by modifying the environment.xml file when hpcc-environment-monitor.path is enabled, the hpcc-environment-monitor.service script will first attempt to shut down any hpccsystems-platform.target that is currently running on the target system. This is so that components previously declared and possibly no longer existing in the environment won't be orphaned on the target system. After the generate-hpccsystems-target.sh script is done invoking the hpccsystems configgen utility, the appropriate service scripts are generated from templates and a fresh hpccsystems-platform.target will exist under /opt/HPCCSystems/etc/systemd/system and symlinked to /etc/systemd/system where appropriate.

Systemd ulimits

If you wish to set custom ulimits for specific hpccsystems components, do so in the templates located in

/opt/HPCCSystems/etc/systemd/system

so that they persist across regenerations of the hpccsystems-platform.target and component service scripts. For the syntax to modify the user limits man systemd.exec.

Workunits and Active Directory

The performance of your system can vary depending on how some components interact. One area which could impact performance is the relationship with users, groups, and Active Directory. If possible, having a separate Active Directory specific to the HPCC Systems platform could be a good policy. There have been a few instances where just one Active Directory servicing many, diverse applications has been less than optimal.

HPCC Systems makes setting up your Active Directory OU's relatively easy. ESP creates all the OU's for you when it starts up, based on the settings you defined in Configuration Manager. You can then start Dali/ESP and use ECLWatch to add or modify users or groups.

You can assign permissions to each user individually, however it is more manageable to assign these permissions to groups, and then add users to these groups as appropriate. Create a group for developers and power users (people with full read/write/delete access), and another group for users that only have only read access and perhaps another group that has both read and write access. Add any other groups as appropriate for your environment. Now you can assign users to their appropriate group(s).

Active Directory, and LDAP Commonality

There are components that are common to both Active Directory and LDAP. There are a few relevant terms, that may need some further explanation.

filesBasedn Deals with restricting access to files. Also referred to as "file scoping".

groupsBasedn Controls the groups associated with the environment. For example, administrators, developers, ws_ecl only, etc.

modulesBasedn Specific to systems using a legacy central repository and controls access to specific modules. Any module you create in the application will create an entry in Eclwatch>>User/Permissions>>Repository Modules

workunitsBasedn Controls access to workunits.

System Tools and Controls

Control Commands

There are control commands which can be run through the *ecl* CLI which goes through ESP to forward the control commands to components on the backend.

Roxie Memlock Status on Individual Nodes

The *ecl roxie getmemlocked* ecl CLI command is one way to check if the Roxie memory is locked. If you wanted to check for locks on individual Roxie nodes, and if you can access those nodes. Use the command:

```
$ cat /proc/`pidof roxie`/status | grep VmLck
```

Which returns something like:

```
VmLck:   5242880 kB
```

If the VmLck value is 0 then memory is not locked.

The VmLck value should match the memsize=roxie log line:

```
00000015 PRG 2022-07-07 12:10:35.754 77841 77841 "RoxieMemMgr: 20480 Pages successfully allocated
         - memsize=5368709120 base=0x7f904fe00000 alignment=262144 bitmapSize=640
```

Which is found in the environment.xml's RoxieCluster setting or in the Helm chart -- values.yaml Roxie section for **totalMemoryLimit**:

```
    totalMemoryLimit="5368709120"
```

Resource Limits

There is also a resource limit set by default at install time that increases the amount of memory allowed to be locked:

```
hpcc soft memlock unlimited
```

and

```
hpcc hard memlock unlimited
```

Without this the limit will not be large enough to lock the memory sizes used by a typical Roxie.

You can also check this limit with a shell command:

```
ulimit -l
```

Be sure to run this command as the *hpcc* user.

Using wutool

wutool *action* **[WUID=nnn] [DALISERVER=ip] [option=value]**

The wutool is a command line utility used to maintain your Workunit store. It can be found in /opt/HPCCSystems/bin/ on any server where the platform has been installed. You can use it to import archived workunits on a Sasha server.

Actions	
list <workunits>	List workunits.
dump <workunits>	Dump xml for specified workunits.
delete <workunits>	Delete workunits.
results <workunits>	Dump results from specified workunits.
info <workunits> <filter>	This command provides filtered access to statistics and other information from a workunit. See the following table for additional info parameter information.
analyze <workunit>	Analyze the workunit to highlight potential cost savings
archive <workunits>	Archive specified Workunits to xml files. The following options are supported: [TO=<directory>] [DEL=1] [DELETERESULTS=1] [INCLUDEFILES=1]
restore <filenames>	Restore from xml files. [INCLUDEFILES=1]
importzap	Imports ZAP report to be able to recreate a workunit and replicate the reported issue. Importzap requires the following parameters. <zapreport-filename> <output-helper-directory> temporary directory to unpack the zap report into <zapreport-password> [optional]
postmortem <workunit>	<workunit> PMD=<dir> - Add post-mortem info
orphans	Delete orphaned information from store
cleanup [days=NN]	Delete workunits older than NN days
validate [FIX=1]	Check contents of workunit repository for errors. With [FIX=1] will try to repair any issues found.
clear	Delete entire workunit repository (requires entire=1 repository=1)
initialize	Initialize new workunit repository
graph <wu>	Generate an alternative representation of the graph with execution details

activity <wu>	What activities are executed between a range of times (in time order) <wu> [">scope|mintime"] ["<scope|maxtime"] [threshold=n%]
hotspot <wu> [<activity>]	Find the hotspots for workunit (or one particular activity)
critical <wu> <activity>	What activities are executed in order to execute activity
depend <wu> <activity> <activity>	Find the common paths between two activities
depend <wu> ?<activity>:startTime	Which dependencies take a large % of the start time for this activity
help <command>	More help on a command

The following table provides further information for the wutool utility issued with the action=info parameter:

info parameters	
info <workunits> <filter>	This command provides filtered access to statistics and other information from a workunit. The filter can include the following elements (those denoted by * can be repeated):
	Which scopes are matched:
scope[<scope-id>]*	scope to match
stype[<scope-type>]*	scope type to match
id[<id>]*	the id of a scope to to match **NOTE**: scope, stype and id cannot be specified in the same filter
depth[n | low..high]	range of depths to search for a match
source[global|stats|graph|all]*	which sources within the workunit to search. Defaults to the optimal sources for the rest of the filter
where[<statistickind> | <statistickind> (=|<|<=|>|>=) value | <statistickind>=low..high]	filter by statistic existence or value range
	Which scopes are include in the results:
matched[true|false]	are the matched scopes returned?
nested[<depth>|all]	what nesting of scopes within a matched scope are in the results (defaults to '0' if matched[true] and 'all' if matched[false])
includetype[<scope-type>]*	which scope types should be included?
	Which information about a scope is reported:
properties[statistics|hints|attributes|scope|all]*	
statistic[<statistic-kind>|none|all]*	
attribute[<attribute-name>|none|all]*	
hint[<hint-name>]*	
property[<statistic-kind>|	include property (category is deduced)

<attribute-name>|<hint-name>]*	
measure[<measure>]	all statistics with a particular measure
version[<version>]	minimum version to return

<workunits> can be specified on the command line or can be specified using a filter owner=XXXX. If omitted, all workunits are selected.

Example:

```
/opt/HPCCSystems/bin/wutool archive DALISERVER=. del=1
```

Redefining nodes in a Thor Cluster

To reconfigure a Thor cluster where you replace existing nodes (with new IP's) or add or remove nodes, you must take an additional step to restructure the group. Dali will not automatically restructure an existing group.

This is because existing published files reference the previous cluster group state by name and therefore changing its structure would invalidate those files and make the physical files inaccessible.

There are a couple of scenarios where you would want to redefine your Thor cluster.

Replacing faulty node(s)

If data files are replicated, replacing a node and forcing the new group to be used by existing files may be desirable. In this scenario, reading an existing file will failover to finding a part on the replicate node, when it tries to find a physical file on the new replacement node.

To force the new group to be used, use the following command:

```
updtdalienv <environment_file> -f
```

In cases where there is no replication, data loss may be unavoidable and forcing the new group may still be the best option.

Resizing the cluster

If you are adding or removing Thor cluster nodes but *all previous nodes remain part of the environment and accessible*, you must **rename** the group that is associated with the Thor cluster (or the Cluster name if there is no group name).

This will ensure all previously existing files, continue to use the old group structure, while new files use the new group structure.

In summary, if the Thor cluster changes it must be updated in the Dali.

Best Practices

This chapter outlines various forms of best practices established by long time HPCC Systems users and administrators running the HPCC Systems platform in a high availability, demanding production environment. While it is not required that you run your environment in this manner, as your specific requirements may vary. This section provides some best practice recommendations established after several years of running the HPCC Systems platform in a demanding, intense, production environment.

Cluster Redundancy

There are several aspects of cluster redundancy that should be considered when setting up your HPCC Systems platform.

Make sure you allocate ample resources to your key components. Dali is RAM intensive. ECL Agent and ECL Server are processor dependent. Thor should have a minimum of 4GB RAM per node.

Dali

Dali should be run in an active/passive configuration. Active/passive meaning you would have two Dalis running, one primary, or active, and the other passive. In this scenario all actions are run on the active Dali, but duplicated on the passive one. If the active Dali fails, then you can fail over to the passive Dali.

Another suggested best practice is to use standard clustering with a quorum and a takeover VIP (a kind of load balancer). If the primary Dali fails, you move the VIP and data directory over to the passive node and restart the Dali service.

DFU Server

You can run multiple instances of the DFU Server. You can run all instances as active, as opposed to an active/passive configuration. There is no need for a load balancer or VIP. Each instance routinely queries the Dali for workunits. Should one fail, the other(s) will continue to pull new workunits.

ECLCC Server

You can run multiple active instances of the ECLCC Server for redundancy. There is no need for a load balancer or VIP for this either. Each instance will routinely check for workunits. Should one fail, the other(s) will continue to compile.

ESP/ECL Watch/WsECL

To establish redundancy, place the ESP Servers in a VIP. For an active/active design, you must use a load balancer. For active/passive you can use pacemaker/heartbeat. If you run active/active, you should maintain a single client's connection to a single server for the life of a session for ECL Watch (port 8010). Other services, such as WsECL (port 8002) do not require a persistent connection to a single server.

ECL Agent

You can run multiple active instances of the ECL Agent. No need for a load balancer or VIP. Each instance routinely queries for workunits. Should one fail, the other(s) will continue to pull new workunits.

Sasha

Sasha should be run in an active/passive configuration. Active/passive meaning you would have two Sashas configured, one primary (active), and the other standing by.

ECL Scheduler

No need for a load balancer, runs active/active. Each instance routinely queries for workunits. Should one fail, the other(s) will continue to schdeule workunits.

Thormaster

Set up Thor in an active/passive configuration. Active/passive meaning you would have two instances running, one primary (active), and the other passive. No load balancer needed. If the active instance fails, then you can fail over to the passive. Failover then uses the VIP (a kind of load balancer) to distribute any incoming requests.

Dropzone

This is just a fileserver that runs the dafilesrv process. Configure in the same fashion as you would any active/passive file server. One primary, or active, and the other passive. No load balancer needed. If the active instance fails, then you can fail over to the passive.

High Availability

If you require high availability for your HPCC Systems platform, there are some additional considerations that you should be aware of. This is not comprehensive list, and it is not meant to be step-by-step instructions for setting up disaster recovery. Instead this section just provides some more information to consider when incorporating HPCC Systems into your disaster recovery plan.

Thor

When designing a Thor cluster for high availability, consider how it actually works -- a Thor cluster accepts jobs from a job queue. If there are two Thor clusters servicing the job queue, one will continue accepting jobs if the other one fails.

With replication enabled, the still-functioning Thor will be able to read data from the back up location of the broken Thor. Other components (such as ECL Server, or ESP) can also have multiple instances. The remaining components, such as Dali, or DFU Server, work in a traditional shared storage high availability failover model.

Another important consideration is to keep your ESP and Dali on separate nodes from your Thor master. This way if your Thor master fails, you can replace it, bring up the replacement with the same IP (address) and it should then come up. Since Thor stores no workunit data, the DALI and ESP can provide the file metadata to recover your workunits.

The Downside

Costs twice as much initially because you essentially have to have two of everything.

The Upside

Almost 100% of the time you can utilize the additional processing capacity. You can run more jobs, have more space, etc.

Disaster Recovery concerns

The important factor to consider for disaster recovery (DR) is the bandwidth required to replicate your data. Your network administrator should evaluate this aspect carefully.

If you have tens of gigabytes of delta each day then an rsync type replication or some sort of hybrid model should suffice. If you have hundreds of gigabytes to petabytes of deltas, the real limit is your budget.

A best practice is to find where the data is the smallest (at ingestion, after normalization, at Roxie) and replicate from that point and rerun the processing in both locations.

The key to getting disaster recovery right is to know your data flow. For instance, if you are ingesting 20TB of raw data daily, then taking that raw data and rolling it up, scoring it, indexing it, etc. You would be better off replicating an intermediate dataset (that we call base files), rather than replicating the large ingest. If the opposite is occurring (small daily ingest and then blow the data up in size) -- you would be better off to ingest the input and then re-run it.

Thor has the ability to do a "Thor copy" which copies data from one cluster to another. You can also do this through ECL code. Additionally, you may decide you don't want, or need to have a "hot" DR Thor. In that case, the most common minor disasters cause only a relatively brief, less than 1 day disaster. Since Thor is responsible for creating data updates it can take a day or a few to recover. The data just is not quite as

fresh but as long as the Roxies are replicated the data is still flowing. In the case of a major disaster such as, a major earthquake, a tidal wave, extended total power loss, multiple fiber cuts, where the systems will be out for a day or more. The likelihood of that occurring may not justify the costs of preventing against it.

Conclusion

Disaster recovery is a calculation. The cost of failure, times the likelihood per year of an event occurring, less than or greater than the cost to prevent against it. Taking all that into consideration can help you to put a sensible DR plan in place.

Roxie

In the case of Roxie, a best practice is to have multiple Roxie clusters and use a proxy to balance. In case of how to keep the data in sync, a pull approach is best. The Roxie automatically pulls the data it needs from the "source" listed in the package file. The data can also be pulled from another Roxie or a Thor. In most cases you would pull to your DR Roxie from the primary Roxie out of the load balancer, but it can also pull from a Thor in the primary location as well.

Middleware

Replication of some components (ECL Agent, ESP/Eclwatch, DFU Server, etc.) are pretty straight forward as they really don't have anything to replicate. Dali is the biggest consideration when it comes to replication. In the case of Dali, you have Sasha as the backup locally. The Dali files can be replicated using rsync. A better approach could be to use a synchronizing device (cluster WAN sync, SAN block replication, etc.), and just put the Dali stores on that and just allow it replicate as designed.

There isn't just a one size fits all approach. Special care, design, and planning are required to make an effective DR strategy that doesn't "over synchronize" across slow WAN links, but still provides you with an acceptable level of redundancy for your business needs.

Best Practice Considerations

There are several other aspects to best practice considerations, and these will change with your system requirements. The following sections are some best practice considerations for some aspects of the HPCC Systems platform. Keep in mind that suggested best practices are merely suggested and may not be appropriate for your needs. A thorough review of the considerations highlighted here can be very helpful if your needs align with the stated considerations.

Multiple Thors

You can run multiple Thors on the same physical hardware. Multiple Thors on the same hardware are independent and unaware of each other. The Thors run jobs as they receive them, regardless of what the other(s) is/are doing. The speed of a single job will never be faster with multiple Thors, but the throughput can be. You can run two Thors picking up jobs from two different queues or the same queue.

The downside of running multiple Thors on the same hardware is that the physical memory on the nodes needs to be shared among each of the Thors. This needs to be configured per Thor cluster definition.

Multiple Thors on the same cluster require them to share the same build and installation. The environment defines each Thor cluster, which can share the same machine set. There are slave and master port settings that need to be set to avoid clashing. There are also memory sharing/splitting considerations and settings that need to be made. The table below indicates settings in the environment to consider.

Setting	Description
globalMemorySize	The maximum memory a slave process can use. Typically 85 percent of the memory on the system divided by the total number of slaves running on the hardware across all Thors.
localThorPortInc	This value is the increment from the base slave port.
masterMemorySize	The maximum memory a Thor master can use. If left blank it will use the *globalMemorySize* value.
masterport	This value must be unique between Thor instances running on the same hardware.
name	The name of each Thor instance must be unique.
nodeGroup	This value is associated with files published by this Thor instance. Normally it is left blank and defaults to the same as the *name* attribute. In environments with multiple Thors sharing the same group of nodes, the *name* value of each Thor must be different. However, the *nodeGroup* value of all the Thors sharing the same physical nodes should be set to the same name. It is very important to make the *nodeGroup* value equal to one of the Thor instance name values.
slaveport	This value must be unique between Thor instances running on the same hardware.
SlavesPerNode	The number of slaves per node per Thor instance.

You must not place multiple Thors on hardware which does not have enough CPU cores to support it. You should not have more Thors than number of cores. One good rule is to use a formula where the number of cores divided by two is the maximum number of Thor clusters to use.

Separate Worker Nodes

In a bare-metal deployment, try as much as possible to keep system target cluster resources running on their own physical nodes or partitions. The idea being to avoid putting essential system components on the

same partitions as resource intensive and highly-variable cluster data. Another similar type condition could occur, if you are running some kind of active/passive high availability. In this case don't keep your active and passive managers on the same node. Try to keep Dali and ESP on separate nodes. Even if you don't have the luxury of very many physical nodes, you would still want to separate the Thor workers and the Dali (at a minimum) to be on separate disks or partitions of the physical nodes. Thor can have varying and expansive data sizes and is not uncommon to have that data take up most of the available capacity on that partition, or even all of it. If the Dali metadata store is sharing the same disk partition as that cluster's Thor data and insufficient disk space condition such as "out of disk space" occurs, the Dali's metadata could become corrupted. The best practice is to keep as many components as possible on their own nodes.

Along those same lines be wary of the Thor worker nodes and try to avoid putting any other system components on nodes with the workers. That would not be optimal and leads to an unbalanced cluster. Resulting in those workers sharing resources with less available memory/cpu taking longer than the others and dragging the whole performance of the cluster down as a result.

Thor times out

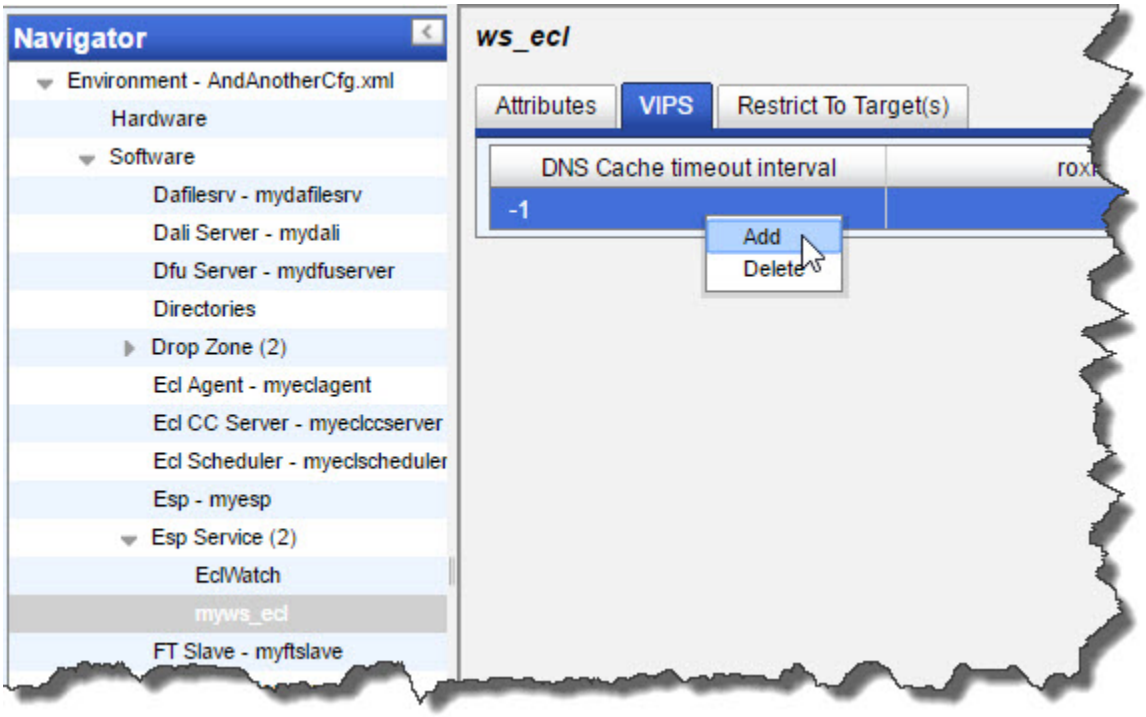
There is a case where a system policy or practice could cause an issue with Thor nodes. At startup if a Thor cluster hangs, and then eventually times out. Then if your Thormaster log shows that the master is fine, but indicates that it is waiting to connect the slaves. Then you may have an issue with the SSH daemon configuration.

There is a security feature called *"AllowUsers"* that creates an AllowList in sshd (the OpenSSH server process) that will disallow connections from anyone not declared on that list. This is not default for sshd, rather it is an option that must be enabled. If that option is enabled that can cause the Thor nodes to hang in the manner described. If that option is enabled, then you must unset the option, or add the hpcc user to the AllowUsers list.

Multiple Roxie Clusters

You can configure multiple Roxie clusters. When you have multiple Roxie clusters, it is better to use a load balancer with those Roxies. To configure multiple Roxie clusters start with adding your Roxie to the VIPS tab in the Configuration Manager.

Figure 30. Configure VIP



Open up the HPCC Systems Configuration Manager and proceed to the Advanced View option. For more information about using ConfigMgr see Using Configuration Manager.

1. Select your ESP Service (the default is the **myws_ecl**) from the Navigator panel on the left side.

2. Select the **VIPS** tab.

3. Right-click on the table, Select *Add*. (see above image)

4. Set the **Send Target To Roxie** value to *False*.

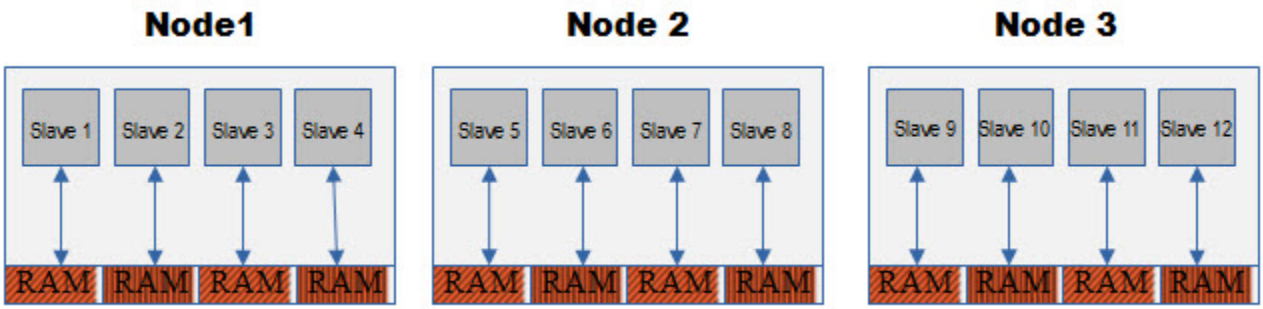
This setting (the *includeTargetInURL* setting) must be false if running multiple Roxie clusters.

Virtual Thor slaves

Beginning in version 6.0.0, Thor clusters can be configured to take full advantage of the resources available per node using Virtual Thor slaves.

In HPCC Systems platform versions prior to 6.0.0, cluster configurations were typically set to N number of **slavesPerNode** , where N equalled or approached the number of cores per machine.

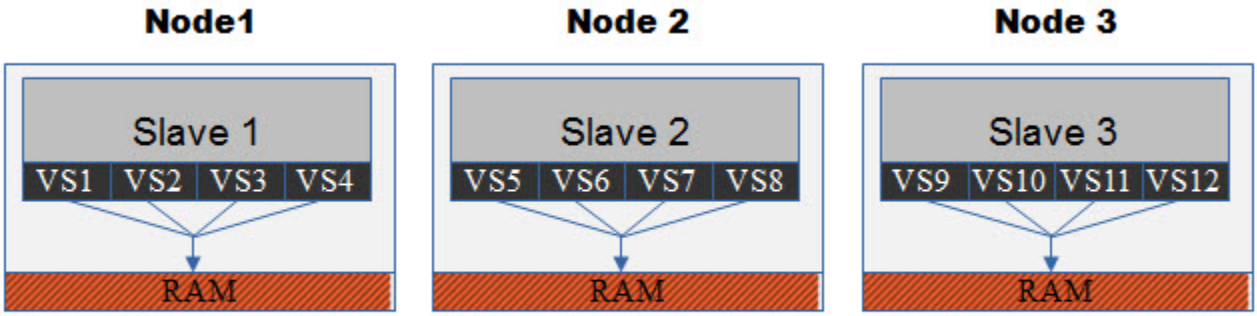
This resulted in N independent slave processes per node, as seen below:



This had several significant disadvantages:

- Each slave process in this configuration has an equal fixed split of the physical memory available to the node.
- Slaves do not share RAM or any other resources.
- Slaves use message passing via the loopback network interface for communication.

Now a new approach is used, allowing virtual slaves to be created with a single slave process, as depicted below:.



* VS = Virtual Slave (also known as a 'channel')

- In this configuration, each physical node has a single Thor slave process.
- Each slave process has N virtual slaves. This is set using a Thor configuration option called **channelsPerSlave** . Under this architecture, slaves within the same process can communicate directly with one another and share resources.

Note: The ***slavesPerNode*** setting still exists and both may be used in combination if required.

Key advantages:

- Each virtual slave shares cached resources, such as key index pages, etc.
- Slaves can request and share all available RAM.
- Startup and management of the cluster is faster and simpler.

- Allows for future enhancements to bring better management/coordination of CPU cores.

The significance of having access to all available memory becomes very significant for some activities. The clearest example is a SMART or LOOKUP JOIN.

SMART/LOOKUP JOIN example

A LOOKUP JOIN works approximately as follows:

- Streams local slave RHS dataset to all other slaves.
- All slaves gather global RHS into one table.
- A hash table based on the hard key match fields is built.
- Once all slaves are done, the LHS is streamed and matched against the hash table to produce the joined results.

Note: The complete RHS table and hash table must fit into memory; otherwise the join fails with an out of memory error.

SMART JOIN is an evolution of LOOKUP JOIN. If it cannot fit the global RHS into memory, it will HASH PARTITION the RHS and HASH DISTRIBUTE the LHS and perform a LOCAL LOOKUP JOIN.

If it cannot fit the local RHS set into memory on any given node, then it will gather and sort both local datasets and perform a standard JOIN.

The key advantage of LOOKUP JOIN is speed. If the RHS fits into memory, it can perform a very quick gather and streamed JOIN of a large LHS set, without the need of gathering and sorting anything.

The advantages of a virtual slave Thor configuration for ECL code using LOOKUP/SMART JOIN is that in effect it will have N times as much memory before it fails or fails over in the SMART JOIN case.

It is also much quicker; instead of broadcasting the local RHS to N slave process per node; it only has to communicate it to one. That one slave can share the same table and same HT with the other virtual slaves directly.

Key advantages of LOOKUP/SMART JOIN in a Virtual Slave Thor Setup:

- N times as much memory available for RHS. In other words, the RHS can be N times bigger before failing or failing over in SMART JOIN case. (In the illustrated example, the JOIN would have 4 times as much memory available)
- Significantly less communication of row data -- equals significantly faster processing for larger RHS sets.

Huge Pages

Linux uses pages as its basic units of memory. Your system may run faster and benefit from huge page support. Huge pages of the appropriate type and size need to be allocated from the operating system. Almost all current Linux systems are set up with Transparent Huge Pages (THP) available by default.

Thor, Roxie, and ECL Agent clusters all have options in the configuration to enable huge page support. The Transparent Huge Pages are enabled for Thor, Roxie, and ECL Agent clusters in the default HPCC Systems environment. Thor clusters can stand to benefit more from huge pages than can Roxie.

You can check the file /sys/kernel/mm/transparent_hugepage/enabled to see what your OS setting is. With THP you do not have to explicitly set a size. If your system is not configured to use THP, then you may want to implement Huge Pages.

Setting up Huge Pages

To set up huge page support, consult your OS documentation and determine how to enable huge page support. For example, the administrator can allocate persistent huge pages (for the appropriate OS) on the kernel boot command line by specifying the "hugepages=N" parameter at boot. With huge pages you also need to explicitly allocate the size.

In HPCC Systems, there are three places in the configuration manager to set the attributes to use Huge Pages.

There are attributes in each component, in the ECL Agent attributes, in Roxie attributes, and in Thor attributes. In each component there are two values:

```
heapUseHugePages
heapUseTransparentHugePages
```

Enable Huge Pages in your operating system, then configure HPCC Systems for the component(s) you wish.

Capacity Planning

Roxie clusters are disk-based High Performance Computing Clusters (HPCC) , typically using indexed files. A cluster is capable of storing and manipulating as much data as its combined hard drive space; however, this does not produce optimal performance.

For maximum performance, you should configure your cluster so agent nodes perform most jobs in memory.

For example, if a query uses three data files with a combined file size of 60 GB, a 40-channel cluster is a good size, while a 60-channel is probably better.

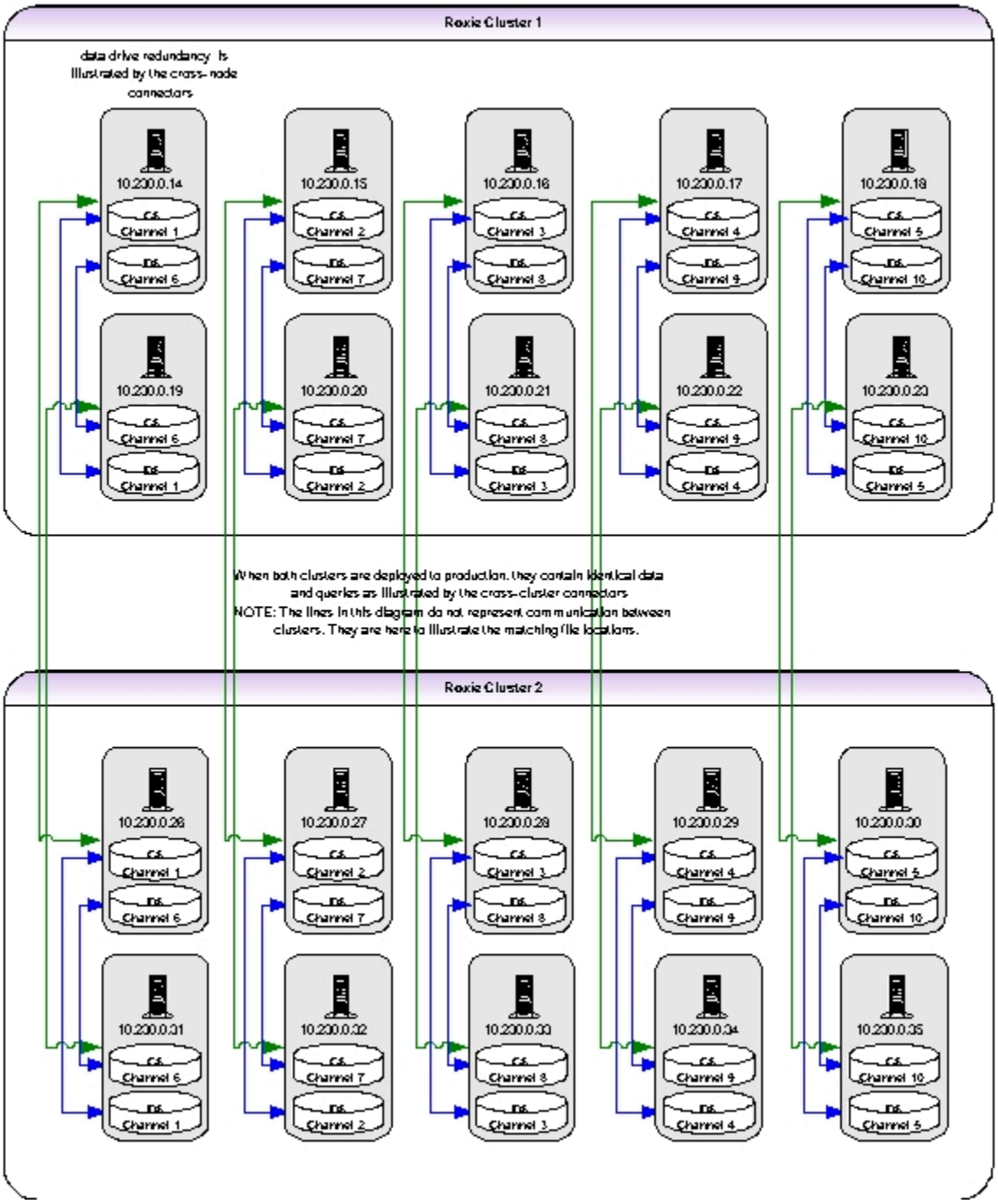
Another consideration is the size of the Thor cluster creating the data files and index files to be loaded. Your target Roxie cluster should be the same size as the Thor on which the data and index files are created or a number evenly divisible by the size of your Roxie cluster. For example, a 100-way Thor to a 20-way Roxie would be acceptable.

This is due to the manner in which data is loaded and processed by Roxie agents. If data is copied to agent nodes, the file parts are directly copied from source location to target locations. They are NOT split or resized to fit a different sized cluster. Therefore, if you load 50 file parts onto a 40-channel cluster, part one goes to channel one, part two to channel two, etc. Parts 41-50 start at the top again so that part 41 goes to channel 1, and part 42 goes to channel 2, etc. The result is an unevenly distributed workload and would result in reduced performance. A cluster will only perform as fast as its slowest node.

The final consideration is the number of Server processes in a cluster. Each agent must also be a Server, but you can dedicate additional nodes to be only Server processes. This is useful for queries that require processing on the Server after results are returned from agents. Those Server-intensive queries could be sent only to dedicated Server IP addresses so the load is removed from nodes acting as both Server and agent.

Configuring the Channels

In the illustration below, the nodes are configured using an N+5 scheme to share channels. Channels can be configured in many ways, this is one example.



In this depiction, each enclosure holds five Roxie agent blades (a row of servers in the picture). We will use this example for the rest of this manual.

Sample Sizings

This section illustrates sample system sizings for various work environments. Unlike system requirements, the following samples are suggestions for setting up you system for various operating conditions.

Sample Sizing for High Data volume (Typical)

The most typical scenario for HPCC Systems is utilizing it with a high volume of data. This suggested sample sizing would be appropriate for a site with large volumes of data. A good policy is to set the Thor size to 4 times the source data on your HPCC Systems. Typically, Roxie would be about 1/4 the size of Thor. This is because the data is compressed and the system does not hold any transient data in Roxie. Remember that you do not want the number of Roxie nodes to exceed the number of Thor nodes.

High Data Thor sizing considerations

Each Thor node can hold about 2.5 TB of data (MAX), so plan for the number of Thor nodes accordingly for your data.

If possible, SAS drives for both Thor and Roxie as they almost equal to SATA drives now. If not for both, get SAS drives at least for your Roxie cluster.

Thor replicates data and is typically configured for two copies.

High Data Roxie sizing considerations

Roxie keeps most of its data in memory, so you should allocate plenty of memory for Roxie. Calculate the approximate size of your data, and allocate appropriately. You should either increase the number of nodes, or increase the amount of memory.

A good practice is to allocate a Dali for every Roxie cluster.

Roxie should have a mirror. This is useful, when you need to update data. You update the mirror then make that primary and bring the other one down. This is a good practice but not really a necessity except in the case of high availability.

Sample Sizing for Heavy Processing on Low Data Volume

The following section provides some sample sizing for heavy processing with approximately the amount of data indicated.

750 GB of Raw Data

Thor = 3 (slaves) + 2 (management) = 5 Nodes

Roxie = 3 (agents) + 1 (Dali) = 4 Nodes (This will mean that the environment will be down during query deployment)

Spares = 2

Total = 13 nodes

1250 GB of Raw Data

Thor = 6 (slaves) + 2 (management) = 8 Nodes

Roxie = 4 (agents) + 1 (Dali) = 5 Nodes (This will mean that the environment will be down during query deployment)

Spares = 2

Total = 17 nodes

2000 GB of Raw Data

Thor = 8 (slaves) + 3 (management) = 11 Nodes

Roxie = 4 (agents) + 1 (Dali) = 5 Nodes (This will mean that the environment will be down during query deployment)

Spares = 2

Total = 20 nodes

3500 GB of Raw Data

Thor = 12 (slaves) + 5 (management) = 17 Nodes

Roxie = 6 (agents) + 1 (Dali) = 7 Nodes (This will mean that the environment will be down during query deployment)

Spares = 2

Total = 28 nodes

System Resources

There are additional resources available for the HPCC Systems platform.

HPCC Systems Resources

The resources link can be found under the Operations Icon link. The resources link in ECL Watch provides a link to the HPCC Systems® web portal. Visit the HPCC Systems® Web Portal at http://hpccsystems.com/ for software updates, plugins, support, documentation, and more. This is where you can find resources useful for running and maintaining HPCC Systems on the web portal.

ECL Watch provides a link to the HPCC Systems portal's download page: http://hpccsystems.com/download. This is the page where you can download Installation packages, virtual images, source code, documentation, and tutorials.

Additional Resources

Additional help with the HPCC Systems platform and Learning ECL is also available. There are online courses available. Go to :

https://learn.lexisnexis.com/hpcc

You may need to register for the site. There are several training videos and other very helpful information.