

Moab HPC Suite

Installation and Configuration Guide 10.2.0 for
Red Hat-Based Systems (8, 9, 10)

March 2026



Legal Notices

© 2015, 2026 Adaptive Computing Enterprises, Inc. All rights reserved.

Distribution of this document for commercial purposes in either hard or soft copy form is strictly prohibited without prior written consent from Adaptive Computing Enterprises, Inc.

This documentation and related software are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

This documentation and related software may provide access to or information about content, products, and services from third-parties. Adaptive Computing is not responsible for and expressly disclaims all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Adaptive Computing. Adaptive Computing will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Adaptive Computing.

Adaptive Computing, Cluster Resources, Moab, Moab Workload Manager, Moab Viewpoint, Moab Cluster Manager, Moab Cluster Suite, Moab Grid Scheduler, Moab Grid Suite, Moab Access Portal, NODUS Cloud OS™, On-Demand Data Center™, and other Adaptive Computing products are either registered trademarks or trademarks of Adaptive Computing Enterprises, Inc. The Adaptive Computing logo and the Cluster Resources logo are trademarks of Adaptive Computing Enterprises, Inc. All other company and product names may be trademarks of their respective companies.

Adaptive Computing Enterprises, Inc.
1100 5th Avenue South, Suite #201
Naples, FL 34102
+1 (239) 330-6093
www.adaptivecomputing.com

Contents

Moab HPC Suite Installation and Configuration Guide Overview	5
Chapter 1: Planning Your Installation	6
1.1 Getting Started	6
1.1.1 Installation Terminology	7
1.1.2 Where to Start	7
1.2 Server Hardware Requirements	8
1.2.1 Topology	9
1.2.2 Hardware Requirements	12
1.2.3 Virtual Machines and Containers	17
1.3 Component Requirements	17
1.3.1 Torque	18
1.3.2 Moab Workload Manager	18
1.4 Identify the Installation Methods	19
1.4.1 Manual Installation	19
1.4.2 RPM Installation	19
Chapter 2: Manual Installation	21
2.1 Manual Installation Steps	21
2.1.1 Preparing for Manual Installation	22
2.1.2 Installing Torque Resource Manager	23
2.1.3 Installing Moab Workload Manager	28
2.2 Additional Configuration	34
2.2.1 Opening Ports in a Firewall	34
2.2.2 Configuring SSL in Tomcat	34
2.2.3 Moab Workload Manager Configuration Options	34
2.2.4 Trusting Servers in Java	36
2.3 Manual Upgrade Steps	38
2.3.1 Upgrading Torque Resource Manager	38
2.3.2 Upgrading Moab Workload Manager	43
Chapter 3: RPM Installation Method	46
3.1 About RPM Installations and Upgrades	46
3.1.1 RPM Installation and Upgrade Methods	46
3.1.2 Special Considerations	47
3.1.3 Installation and Upgrade Process	47
3.2 RPM Installations	47
3.2.1 Preparing the Host (RPM)	48


3.2.2 Installing Torque Resource Manager (RPM)	50
3.2.3 Installing Moab Workload Manager (RPM)	53
3.2.4 Disabling the Adaptive Repository after Installs (RPM)	56
3.3 Additional Configuration	56
3.3.1 Opening Ports in a Firewall (RPM)	56
3.3.2 Configuring SSL in Tomcat (RPM)	57
3.3.3 Trusting Servers in Java (RPM)	57
3.3.4 Updating the Reporting Application Configuration (RPM)	58
3.4 RPM Upgrades	59
3.4.1 Preparing the Host (RPM)	59
3.4.2 Upgrading Torque Resource Manager (RPM)	61
3.4.3 Upgrading Moab Workload Manager (RPM)	64
3.4.4 Disabling the Adaptive Repository after Upgrades (RPM)	67
Chapter 4: Troubleshooting	68
4.1 General Issues	68
4.1.1 Where do I set credentials and what are the default values?	68
4.1.2 FastX Error Message: Logins are disabled on this system	72
4.2 Port Reference	73
4.3 Moab Workload Manager Issues	76
4.3.1 Moab error: cannot determine local hostname	76
4.3.2 Moab error: Moab will now exit due to license file not found	76
4.4 Moab Web Services Issues	77
4.4.1 MongoDB: Errors during MWS startup	77
4.4.2 MongoDB: Out of semaphores to get db connection	79
4.4.3 MongoDB: Connection wait timeout after 120000 ms	80
4.4.4 java.lang.OutOfMemoryError: Java heap space	80
4.4.5 java.lang.OutOfMemoryError: PermGen space	80
4.4.6 SEVERE: Context [/mws] startup failed due to previous errors	81
4.4.7 Moab HPC Suite Reached Maximum Number of Concurrent Connections	81
4.4.8 MongoDB Service Does Not Start	81
4.5 Moab Viewpoint Issues	82
4.5.1 General Configuration Issues	83
4.5.2 Only the Configuration Page is Displayed in Viewpoint	84
4.5.3 Viewpoint Does Not Report Any of My Jobs or Nodes	85
4.5.4 viewpoint-query-helper Plugin Does Not Connect	86
4.5.5 Job's Processor Count Changes After Submission	88

Moab HPC Suite Installation and Configuration Guide Overview

Welcome to the *Moab HPC Suite Installation and Configuration Guide 10.2.0* for Red Hat-Based Systems (8, 9, 10)

This guide includes detailed instructions for installing each component of the suite so that you can quickly get up and running.

This guide is intended for system administrators who are responsible for installing the Moab HPC Suite components.


 Depending on your system configuration and license, not all of the HPC Suite components may be available.

The 10.2.0 Moab HPC Suite contains the following components:


- Torque Resource Manager 7.2.0
- Moab Workload Manager 10.2.0

Before commencing the installation or upgrade, see [Chapter 1: Planning Your Installation](#) to verify your system conforms to minimum prerequisites.

Chapter 1: Planning Your Installation

 We highly recommend that you *first* perform installations and upgrades in a *test environment*. Standard installation and upgrade procedures and use cases are tested prior to release. However, due to the wide range of possible configurations and customizations, it is important to exercise caution when deploying new versions of software into your production environments. This is especially true when the workload has a vital bearing on your organization's day-to-day operations. We recommend that you test in an environment that mirrors your production environment's configuration, workflow, and load as closely as possible. Contact your Adaptive Computing account manager for suggestions and options for installing/upgrading to newer versions.

There are many different ways to install and configure the Moab HPC Suite. Each environment has its own set of requirements and preferences. This chapter is intended to help an administrator understand how each of the Moab HPC Suite components interact, basic requirements, and configuration information to prepare for the installation.

 Code samples have been provided for convenience. Some code samples provide sample passwords (i.e., changeme!). We strongly recommend that you do not use these passwords during installation, as using the documented passwords could introduce unnecessary security vulnerabilities into your system.

In this chapter:

- [1.1 Getting Started](#)
- [1.2 Server Hardware Requirements](#)
- [1.3 Component Requirements](#)
- [1.4 Identify the Installation Methods](#)

1.1 Getting Started

In this section:

1.1.1 Installation Terminology

1.1.2 Where to Start

1.1.1 Installation Terminology

To aid in documentation clarity, Adaptive Computing uses the following terms in this Installation and Configuration Guide:

- **Components** – The different 'products' included in the Moab HPC Suite. For example, Moab Workload Manager, Moab Web Services.
- **Servers** – Also known as components, but specifically relating to the actual services. For example, the Moab Workload Manager component is referred to as the Moab Server for non-client services.
- **Host** – The actual box where a Moab HPC Suite component (server or client) is installed.

i Previous documentation typically used Head Node to designate a host or a Server.

1.1.2 Where to Start

You need to plan your environment and determine how many hosts you will need and for which components you will install using the Manual Installation or the RPM Installation method. The following are suggested steps to help you in your planning and installing process.

1. Determine whether you have a small, medium, High-Throughput, or large environment, including an example and required and recommended hardware requirements. See [1.2 Server Hardware Requirements](#).
2. Decide whether you will perform a Manual Installation or an RPM Installation for the various components. See [1.4 Identify the Installation Methods](#).

i The Manual Installation and the RPM Installation sections include 'Additional Configuration' that provides additional information and instructions for optional but recommended configurations (for example, Configuring SSL in Tomcat).

3. Review the software requirements for your components and set up your hosts accordingly. See [1.3 Component Requirements](#).

4. Install the individual components on their respective host(s). See [2.1.1 Preparing for Manual Installation](#) or [3.1 About RPM Installations and Upgrades](#), as applicable.
5. Refer to [Chapter 4: Troubleshooting](#) for assistance in addressing common problems during installation and configuration.

1.2 Server Hardware Requirements

The Moab HPC Suite is installed and configured differently for small, medium, or large environment types. This topic provides a general topology of the Moab HPC Suite and the server hardware requirements depending on your environment size.

In this section:

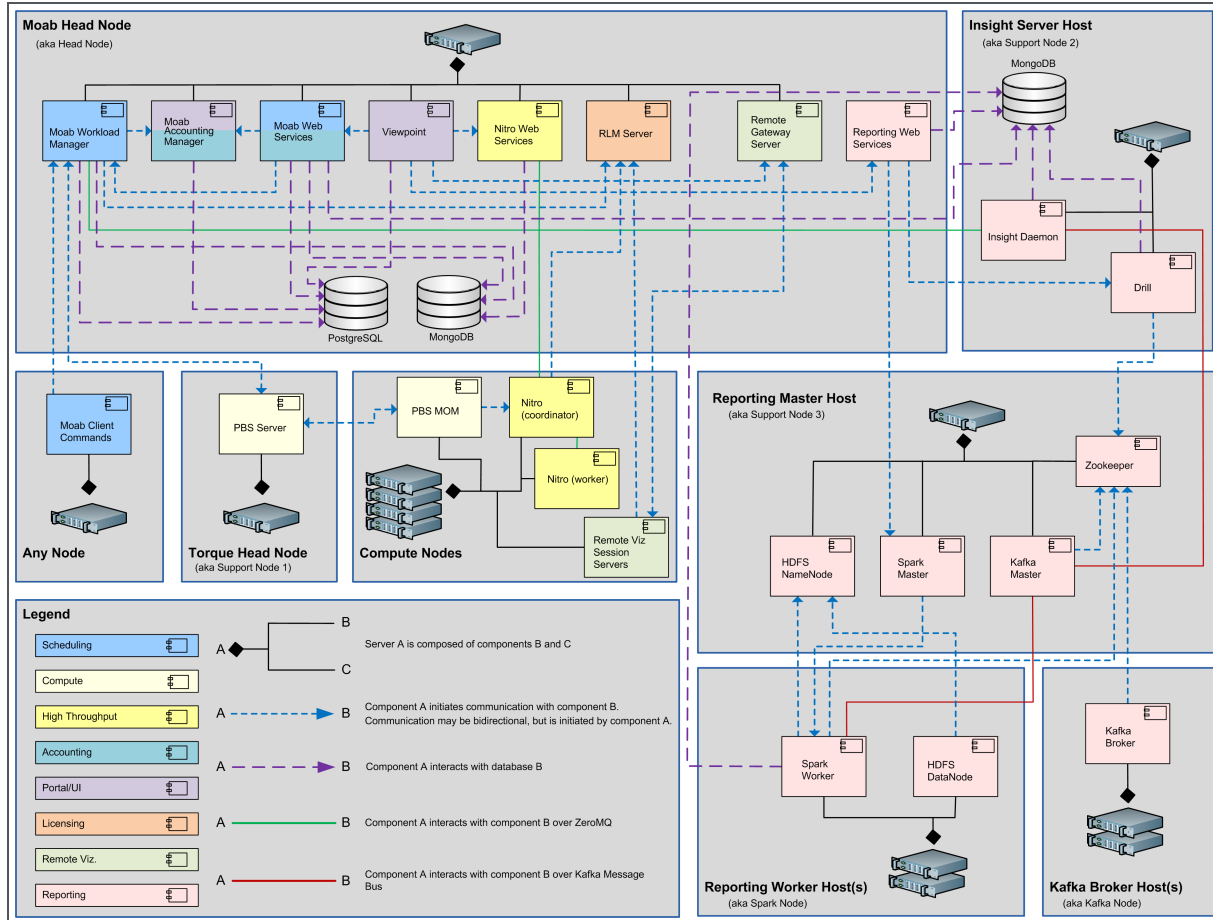
[1.2.1 Topology](#)

[1.2.2 Hardware Requirements](#)

[1.2.3 Virtual Machines and Containers](#)

1.2.1 Topology

The following diagram provides a general topology of the Moab HPC Suite for a medium (with high throughput) or a large environment:



Note the following:

- Smaller environments may elect to consolidate the Torque Server with the Moab Server on the same host, including the PBS Server in the list of components installed on the same host.
- Although Moab Workload Manager and Moab Accounting Manager can share the same database instance, it is not a requirement. Two database instances can be used, one for each component.
- Larger systems will require more dedicated resources for each component, in which case it may be necessary to move individual components from the Moab Server Host (i.e., databases, Moab Accounting Manager) to their own respective servers.

Software components that may be included in a Moab HPC Suite installation are described in the table below:

Component	Description
Moab Workload Manager	A scheduling and management system designed for clusters and grids.
Torque Resource Manager - PBS Server	A resource manager for Moab HPC Suite. Torque provides the low-level functionality to discover and report cluster resources/features, and to start, hold, cancel, and monitor jobs. Required by Moab Workload Manager.
Torque Resource Manager - PBS MOM	Torque MOMs (Machine Oriented Mini-servers) are agents installed on each compute node that complete tasks assigned to them by the Torque Server. When a multi-node job runs, one of the Torque MOMs is assigned the role of Mother Superior, and all other nodes assigned to the job are sister nodes. Mother Superior manages the job across all the sister nodes by communicating with each of them and updating the Torque Server. Required by Torque.
Moab Accounting Manager	An accounting management system that allows for usage tracking, charge accounting, and allocation enforcements for resource usage in technical computing environments. Required by Moab Workload Manager and Moab Web Services.
Moab Web Services (MWS)	A component of the Moab HPC Suite that enables programmatic interaction with Moab Workload Manager via a RESTful interface. MWS lets you create and interact with Moab objects and properties such as jobs, nodes, virtual machines, and reservations. MWS is the preferred method for those wanting to create custom user interfaces for Moab.
Moab Insight	A component of the Moab HPC Suite that collects the data that Moab HPC Suite emits on its message queue and stores it in a database. The message queue is efficient, can be encrypted, and tolerates disconnections and restarts on either side. Required by Kafka Master.
Reporting Web Services (RWS)	A component of Adaptive Computing Suites that enables programmatic interaction with Moab Reporting and Analytics via a RESTful interface. RWS is the preferred method for those wanting to create custom user interfaces for Moab Reporting and Analytics.
Reporting and Analytics	Streams in massive amounts of workload and resource usage data from your High-Performance Computing (HPC), High-Throughput Computing (HTC), and Grid Computing environments, and then correlates that information against users, groups, accounts, and organizations so you can gain insights into exactly how your investment is being used and how well it aligns with your goals.

Component	Description
MongoDB	A free and open-source cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with schemas. Required by Moab Workload Manager, Moab Web Services, Reporting Web Services, and Spark Worker.
PostgreSQL	An object-relational database (ORDBMS). That is, RDBMS, with additional (optional use) object features – with an emphasis on extensibility and standards compliance. Required by Moab Workload Manager, Moab Accounting Manager, and Moab Web Services.
Drill	Apache Drill is an open-source software framework that supports data-intensive distributed applications for interactive analysis of large-scale datasets. Required by Reporting Web Services.
Hadoop	The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly available service on top of a cluster of computers, each of which may be prone to failures. Required by Spark Worker.
Spark Master	Apache Spark is a fast and general engine for large-scale data processing. Spark Streaming is an extension of the core Spark API that enables scalable, high-throughput, fault-tolerant stream processing of live data streams. The Spark Master uses one or more Spark Workers when processing live data streams. Data can be ingested from many sources like Kafka, Flume, Kinesis, or TCP sockets and can be processed using complex algorithms expressed with high-level functions like map, reduce, join, and window. Finally, processed data can be pushed out to file systems, databases, and live dashboards. Required by Reporting Web Services.
Spark Worker	The Spark Worker is used by a Spark Master when processing live data streams. Required by Spark Master.
Kafka Master	Apache Kafka is used for building real-time data pipelines and streaming apps. It is horizontally scalable, fault-tolerant, wicked fast, and runs in production in thousands of companies. Kafka Master uses one or more Kafka Brokers when pipelining and processing live data streams. Required by Spark Worker and Insight.
Kafka Broker	Kafka Broker is used by a Kafka Master to pipeline and process live data streams. Apache Kafka is used for building real-time data pipelines and

Component	Description
	streaming apps. It is horizontally scalable, fault-tolerant, wicked fast, and runs in production in thousands of companies. Required by Kafka Master.

1.2.2 Hardware Requirements

The following tables show hardware requirements for Moab HPC Suite, Torque, and Reporting Framework environments of various deployment sizes.

In this topic:

[1.2.2.A Moab HPC Suite and Torque Requirements](#)

[1.2.2.B Reporting Framework Requirements](#)

1.2.2.A Moab HPC Suite and Torque Requirements

The following table identifies the minimum and recommended hardware requirements for the different environment types. Use this table as a guide when planning out your suite topology.

i Software requirements are listed per-component rather than suite-wide as the suite components reside on different hosts. See [1.3 Component Requirements](#)

Environment Type	# of Compute Nodes	Jobs/Week	Minimum Requirements (per Host Distribution)	Recommended Requirements (targeting minimum number of hosts)
Proof of Concept / Small Demo	50	<1k	Moab Server+Torque Server Host <ul style="list-style-type: none"> • 4 Intel/AMD x86-64 cores • At least 8 GB RAM • At least 100 GB dedicated 	Same as minimum

Environment Type	# of Compute Nodes	Jobs/Week	Minimum Requirements (per Host Distribution)	Recommended Requirements (targeting minimum number of hosts)
			<p>disk space</p> <p>Insight Server Host</p> <ul style="list-style-type: none"> • 8 Intel/AMD x86-64 cores • At least 16 GB RAM • At least 512 GB dedicated disk space 	
Medium	500	<100k	<p>Moab Server+Torque Server Host</p> <ul style="list-style-type: none"> • 8 Intel/AMD x86-64 cores • At least 16 GB RAM • At least 512 GB dedicated disk space <p>Insight Server Host</p> <ul style="list-style-type: none"> • 8 Intel/AMD x86-64 cores • At least 16 GB of RAM • At least 1024 GB disk 	<p>Moab Server+Torque Server Host</p> <ul style="list-style-type: none"> • 16 Intel/AMD x86-64 cores • At least 32 GB RAM • At least 1 TB dedicated disk space <p>Insight Server Host</p> <ul style="list-style-type: none"> • 8 Intel/AMD x86-64 cores • At least 16 GB of RAM dedicated 1 Gbit channel between Insight and Moab • 128 GB local SSD for swap • At least 1024 GB disk

Environment Type	# of Compute Nodes	Jobs/Week	Minimum Requirements (per Host Distribution)	Recommended Requirements (targeting minimum number of hosts)
Medium with High Throughput or Larger	>500	>100k	<p>Moab Server Host</p> <ul style="list-style-type: none"> • 8 Intel/AMD x86-64 cores • At least 16 GB RAM • At least 512 GB dedicated disk space <p>Torque Server Host</p> <ul style="list-style-type: none"> • 8 Intel/AMD x86-64 cores • At least 16 GB RAM • At least 512 GB dedicated disk space <p>Insight Server Host</p> <ul style="list-style-type: none"> • 8 Intel/AMD x86-64 cores • At least 16 GB of RAM • At least 2048 GB disk 	<p>The Moab Server should <i>not</i> reside on the same host as the Torque Server.</p> <p>MWS Server <i>must</i> reside on the same host as the Moab Server (Moab Server Host).</p> <p>The MAM Server can reside on its own host, on the Moab Host (preferred), or another server's host (except for the Insight Host).</p> <p>Databases can reside on the same or a different host from its server component.</p>

Note the following:

- All requirements above (minimum and recommended) target a minimum number of management servers. Admins are encouraged to separate the Torque Server and the Moab Server onto different hosts where possible for better results, especially when High Throughput is enabled.
- Although many factors may have an impact on performance (network bandwidth, intended use and configuration, etc.), we consider High Throughput as something that makes a significant enough difference between minimum and recommended hardware requirements to merit mention in the table above.
- Moab and Torque are both multi-threaded and perform better with more processors.
- Due to the large amount of data that Moab must send to Insight, Moab performs better without Insight enabled (for environments that do not use Crystal Reporting).
- Regarding disk space, consideration should be given to requirements related to log files, log depth, number of jobs/nodes/reservations (more objects impact database journal size), average number of events generated (more events take more space), etc.

1.2.2.B Reporting Framework Requirements

The following table shows hardware requirements for the Reporting and Kafka hosts needed to support the addition of the Reporting Framework to a Moab HPC Suite environment. These requirements are *in addition* to the requirements shown in the table above.

Environment Type	Minimum Requirements (per Host Distribution)	Recommended Requirements (targeting minimum number of hosts)
Proof of Concept / Small Demo	<p>Reporting Master Host</p> <ul style="list-style-type: none"> • 4 Intel/AMD x86-64 cores • At least 8 GB RAM • At least 512 GB dedicated disk space <p>Reporting Worker Host</p> <ul style="list-style-type: none"> • 8 Intel/AMD x86-64 cores • At least 16 GB RAM • At least 512 GB dedicated disk space 	Same as minimum

Environment Type	Minimum Requirements (per Host Distribution)	Recommended Requirements (targeting minimum number of hosts)
	<p>Kafka Broker Host</p> <ul style="list-style-type: none"> • 4 Intel/AMD x86-64 cores • At least 6 GB RAM • At least 512 GB dedicated disk space 	
Medium	<p>Reporting Master Host</p> <ul style="list-style-type: none"> • 4 Intel/AMD x86-64 cores • At least 8 GB RAM • At least 1024 GB dedicated disk space <p>Reporting Worker Host</p> <ul style="list-style-type: none"> • 8 Intel/AMD x86-64 cores • At least 16 GB RAM • At least 512 GB dedicated disk space <p>Kafka Broker Host</p> <ul style="list-style-type: none"> • 4 Intel/AMD x86-64 cores • At least 6 GB RAM • At least 1024 GB dedicated disk space 	<p>Reporting Master Host</p> <ul style="list-style-type: none"> • 4 Intel/AMD x86-64 cores • At least 16 GB RAM • At least 1024 GB dedicated disk space <p>Reporting Worker Host</p> <ul style="list-style-type: none"> • 8 Intel/AMD x86-64 cores • At least 32 GB RAM • At least 512 GB dedicated disk space <p>Kafka Broker Host</p> <ul style="list-style-type: none"> • 4 Intel/AMD x86-64 cores • At least 6 GB RAM • At least 1024 GB dedicated disk space
Medium with High Throughput or Larger	<p>Reporting Master Host</p> <ul style="list-style-type: none"> • 4 Intel/AMD x86-64 cores • At least 16 GB RAM • At least 2048 GB dedicated disk space <p>Reporting Worker Host</p>	More than one Reporting Worker host is recommended.

Environment Type	Minimum Requirements (per Host Distribution)	Recommended Requirements (targeting minimum number of hosts)
	<ul style="list-style-type: none"> • 8 Intel/AMD x86-64 cores • At least 32 GB RAM • At least 512 GB dedicated disk space <p>Kafka Broker Host</p> <ul style="list-style-type: none"> • 4 Intel/AMD x86-64 cores • At least 6 GB RAM • At least 2048 GB dedicated disk space 	

1.2.3 Virtual Machines and Containers

The Moab HPC suite can be installed on virtual machines. There are both free and commercial virtual machine platforms available. However, installing the Moab HPC Suite on a container (e.g., Docker, Singularity, or LXC) has been known to be problematic and is not officially supported.

1.3 Component Requirements

i On RHEL systems, you must be registered for a Red Hat subscription in order to have access to required RPM package dependencies.

In this section:

[1.3.1 Torque](#)

[1.3.2 Moab Workload Manager](#)

1.3.1 Torque

Supported Operating Systems

- Red Hat 8, 9, 10

Software Requirements

- libxml2-devel package (package name may vary)
- Tcl/Tk version 8 or later if you plan to build the GUI portion of Torque, or use a Tcl-based scheduler
- cgroups:
 - cgroups v1 or v2 is recommended for the tarball install and required by the RPM install
 - dbus-libs is required if using cgroups v2
- If using NVIDIA/NVML configuration flags:
 - NVIDIA driver version $\geq 450.80.02$
 - CUDA driver ≥ 11.0
- If you build Torque from source, the following additional software is required:
 - gcc-c++
 - posix-compatible version of make
 - libtool
 - flex
 - yacc
 - zlib-devel
 - hwloc-devel (if built with cgroups enabled)

1.3.2 Moab Workload Manager

Supported Operating Systems

- Red Hat 8, 9, 10

Software Requirements

- libcurl (<https://curl.haxx.se/libcurl/>)
- Perl 5.8.8 or later
- perl-CPAN (package name may vary)
- libxml2-devel (package name may vary)
- *(Optional)* Moab Accounting Manager
- *(Optional)* MySQL, PostgreSQL, or Oracle with ODBC driver (see 'Database Configuration' in the *Moab Workload Manager Administrator Guide* for details)

Supported Resource Managers

- Torque

1.4 Identify the Installation Methods

Adaptive Computing provides different methods for installing the Moab HPC Suite components: Manual Installation, RPM Installation (uses RPM methodology).

Depending on your environment and which components you are installing (and on which host), you may need to use a combination of Manual Installation and RPM Installation.

1.4.1 Manual Installation

This method provides both advantages and disadvantages for admins who want non-standard configuration options.

This method has more supported operating systems than the RPM Installation method. However, some components cannot be installed using the Manual Installation method.

See [Chapter 2: Manual Installation](#) for more information on the Manual Installation method.

1.4.2 RPM Installation

This method provides advantages for admins who want a standard installation with little customization.

Whether you are installing RPMs on one host or on several hosts, each host must have the Adaptive Computing Package Repository enabled.

Some customization options are available for Moab Workload Manager and Moab Accounting Manager by building custom RPMs. See [2.1.3.C \(Optional\) Build a Custom RPM](#) for Moab Workload Manager.

See [Chapter 3: RPM Installation Method](#) for more information on the RPM Installation method.

Chapter 2: Manual Installation

This chapter provides installation, configuration, and upgrading information using the Manual Installation method.

Be aware of the following:

- On RHEL systems, you must be registered for a Red Hat subscription in order to have access to required RPM package dependencies.
- Manual Installation is not available for Insight.
- During the installation, you will access many system-level files and directories, so you should execute the instructions in this guide with root privileges. You will see that the instructions execute commands as the root user. Also be aware that the same commands will work for a non-root user with the `sudo` command.

In this chapter:

- [2.1 Manual Installation Steps](#)
- [2.2 Additional Configuration](#)
- [2.3 Manual Upgrade Steps](#)

Related Topics

- [Chapter 1: Planning Your Installation](#)

2.1 Manual Installation Steps

This section provides instructions and other information for installing your Moab HPC Suite components using the Manual installation method.

In this section:

- [2.1.1 Preparing for Manual Installation](#)
- [2.1.2 Installing Torque Resource Manager](#)
- [2.1.3 Installing Moab Workload Manager](#)

2.1.1 Preparing for Manual Installation

The manual installation process of the Moab HPC Suite includes installing the different components in the suite.

i Many individual components have dependencies on other components (see [Chapter 1: Planning Your Installation](#)). However, if you do not require a certain component, you do not have to install it.

The install instructions for each component include information about system requirements and dependencies. Some include instructions that you need to complete before you begin the install. Read this information carefully, and make sure you have installed all the dependencies and packages that are necessary in order to avoid errors during the Moab HPC Suite install process.

i Because many system-level files and directories are accessed during the installation, the instructions in this guide should be executed with root privileges.

You will see that the instructions execute commands as the root user. Note that the same commands will work for a non-root user with the `sudo` command.

This topic contains instructions that you need to complete before you begin the installations.

In this topic:

[2.1.1.A Set Up Proxies](#)

[2.1.1.B Enable Extra Packages for the Repository](#)

[2.1.1.C Update Your System Software to the Latest Version](#)

[2.1.1.D Ensure Hostname Resolution for All Hosts](#)

[2.1.1.E Install the Moab HPC Suite Software Components](#)

2.1.1.A Set Up Proxies

If your site uses a proxy to connect to the internet, configure `dnf` to use a proxy by editing the `/etc/dnf.conf` file as follows:

```
proxy=https://<proxy_server_id>:<port>
```

If your site uses an external repository to install Python dependencies, set up `pip` to use a proxy. Do the following:

```
export https_proxy=https://<proxy_server_id>:<port>
```

2.1.1.B Enable Extra Packages for the Repository

Many individual components have dependencies that are found in the optional add-on repositories for the distribution. You must enable the respective repository for your distribution on all hosts upon which you install Adaptive Computing software components. Do the following:

On non-RHEL Red Hat-based systems (e.g., CentOS, Rocky Linux, AlmaLinux, and Scientific Linux), install the EPEL release package in order to have access to required RPM package dependencies:

```
[root]# dnf install epel-release
```

On RHEL systems, you must be registered for a Red Hat subscription in order to have access to required RPM package dependencies.

2.1.1.C Update Your System Software to the Latest Version

We recommend that you update your system software to the latest version before installing Moab HPC Suite components.

On *each* host where you will install the Moab HPC Suite components, run the following:

```
[root]# dnf update
```

2.1.1.D Ensure Hostname Resolution for All Hosts

Each host should be resolvable from all other hosts in the cluster. Usually this is implemented by having all hosts in DNS. Alternatively, each host can include all other hosts (with the correct IP address) in its `/etc/hosts` file.

2.1.1.E Install the Moab HPC Suite Software Components

To install the Moab HPC Suite, install the packages in the following order:

1. Torque. See [2.1.2 Installing Torque Resource Manager](#).

2.1.2 Installing Torque Resource Manager

This topic contains instructions on how to install and start Torque Resource Manager (Torque).

In this topic:

- [2.1.2.A Open Necessary Ports](#)
- [2.1.2.B Install Dependencies, Packages, or Clients](#)
- [2.1.2.C Install Torque Server](#)
- [2.1.2.D Install Torque MOMs](#)
- [2.1.2.E Install Torque Clients](#)
- [2.1.2.F Configure Data Management](#)

2.1.2.A Open Necessary Ports

Torque requires certain ports to be open for essential communication.

If your site is running firewall software on its hosts, configure the firewall to allow connections to the necessary ports:

Location	Port	Function	When Needed
Torque Server Host	15001	Torque Client and MOM communication to Torque Server	Always
Torque MOM Host (Compute Nodes)	15002	Torque Server communication to Torque MOMs	Always
Torque MOM Host (Compute Nodes)	15003	Torque MOM communication to other Torque MOMs	Always

If using the MOM hierarchy (documented in 'Setting Up the MOM Hierarchy' in the *Torque Resource Manager Administrator Guide*), you must also open port 15003 from the server to the nodes.

See also:

- [2.2.1 Opening Ports in a Firewall](#) for general instructions and an example of how to open ports in the firewall.
- 'Configuring Ports' in the *Torque Resource Manager Administrator Guide* for more information on how to configure the ports that Torque uses for communication.

2.1.2.B Install Dependencies, Packages, or Clients

Install Build Packages

On the Torque Server Host, run the following command to install the `dbus-devel`, `libxml2-devel`, and build dependency packages:

```
[root]# dnf install dbus-devel gcc-c++ libtool libxml2-devel make
```

Install hwloc

On the Torque Server Host, each Torque MOM Host, and each Torque Client Host, install the `hwloc` development package:

```
[root]# dnf install hwloc-devel
```

- If using RHEL 8, first enable the CodeReady Linux Builder repository before installing the `hwloc` development package

```
[root]# subscription-manager repos --enable codeready-builder-for-rhel-8-x86_64-rpms
```

- If using (non-RHEL) 8, first enable the PowerTools repository before installing the `hwloc` development package:

```
[root]# dnf config-manager --set-enabled powertools
```

2.1.2.C Install Torque Server

i You must complete the tasks to install the dependencies, packages, or clients before installing Torque Server. See [2.1.2.B Install Dependencies, Packages, or Clients](#).

If your configuration uses firewalls, you *must also* open the necessary ports before installing the Torque Server. See [2.1.2.A Open Necessary Ports](#).

On the Torque Server Host, do the following.

1. Download the latest Torque tarball from the [Adaptive Computing](#) website:

```
[root]# tar -xzf torque-7.2.0.tar.gz
[root]# cd torque-7.2.0/
```

2. Determine which `./configure` command options you need to add, based on your system configuration. At a minimum, you add `--enable-cgroups`

i These instructions assume you are using cgroups.

See 'Customizing the Install' in the *Torque Resource Manager Administrator Guide* for more information on which options are available to customize the `./configure` command.

3. Run the following commands:

```
[root]# ./configure --enable-cgroups # add any other specified options
[root]# make
[root]# make install
```

4. Source the appropriate profile file to add `/usr/local/bin` and `/usr/local/sbin` to your path:

```
[root]# ./etc/profile.d/torque.sh
```

5. Initialize `serverdb` by executing the `torque.setup` script:

```
[root]# ./torque.setup root
```

6. Add nodes to the `/var/spool/torque/server_priv/nodes` file. See 'Specifying Compute Nodes' in the *Torque Resource Manager Administrator Guide* for information on syntax and options for specifying compute nodes.

7. Configure `pbs_server` to start automatically at system boot, and then start the daemon:

```
[root]# qterm
[root]# systemctl enable pbs_server.service
[root]# systemctl start pbs_server.service
```

2.1.2.D Install Torque MOMs

In most installations, you will install a Torque MOM on each of your compute nodes.

i See 'Specifying Compute Nodes' or 'Configuring Torque on Compute Nodes' in the *Torque Resource Manager Administrator Guide* for more information.

1. On the Torque Server Host, do the following:
 - a. Create the self-extracting packages that are copied and executed on your nodes:

```
[root]# make packages
```

- b. Copy the self-extracting MOM packages to each Torque MOM Host. We recommend that you use a remote shell, such as SSH, to install packages on remote systems. Set up shared SSH keys if you do not want to supply a password for each Torque MOM Host.

```
[root]# scp torque-package-mom-linux-x86_64.sh <mom-node>:
```

- c. Copy the `pbs_mom` startup script to each Torque MOM Host.

i Not all sites see an inherited `ulimit`, but those that do can change it in the `pbs_mom` init script, which is responsible for starting and stopping the `pbs_mom` process. The script includes a check for the presence of `/etc/sysconfig/pbs_mom`, so the natural place to put `ulimits` would be there (or directly in the init script itself).

```
[root]# scp contrib/systemd/pbs_mom.service <mom-node>:/usr/lib/systemd/system/
```

2. On *each* Torque MOM Host, do the following:

- a. If using `cgroups v1` (e.g., Red Hat 8), install `libcgroup-tools`:

```
[root]# dnf install libcgroup-tools
```

i If you enabled `cgroups` in the configure step, you can see the detected version of `cgroups` at the end of the configure output.

- b. Install the self-extracting MOM package:

```
[root]# ./torque-package-mom-linux-x86_64.sh --install
```

- c. (Optional) If you expect your jobs to require more than the default 12 MB of stack space, increase the stack limit by editing the `LimitSTACK` setting in `/usr/lib/systemd/system/pbs_mom.service`:

```
LimitSTACK=infinity
```

- d. Configure `pbs_mom` to start at system boot, and then start the daemon:

```
[root]# systemctl enable pbs_mom.service
[root]# systemctl start pbs_mom.service
```

2.1.2.E Install Torque Clients

If you want to have the Torque client commands installed on hosts other than the Torque Server Host (such as the compute nodes or separate login nodes), do the following.

1. On the Torque Server Host, do the following:
 - a. Copy the self-extracting client package to *each* Torque Client Host.

i We recommend that you use a remote shell, such as SSH, to install packages on remote systems. Set up shared SSH keys if you do not want to supply a password for each Torque Client Host.

```
[root]# scp torque-package-clients-linux-x86_64.sh <torque-client-host>:
```

- b. Copy the trqauthd startup script to *each* Torque Client Host:

```
[root]# scp contrib/systemd/trqauthd.service <torque-client-host>:/usr/lib/systemd/system/
```

2. On *each* Torque Client Host, do the following:

- a. Install the self-extracting client package:

```
[root]# ./torque-package-clients-linux-x86_64.sh --install
```

- b. Enable and start the trqauthd service:

```
[root]# systemctl enable trqauthd.service
[root]# systemctl start trqauthd.service
```

2.1.2.F Configure Data Management

When a batch job completes, stdout and stderr files are generated and placed in the spool directory on the master Torque MOM Host for the job instead of the submit host. You can configure the Torque batch environment to copy the stdout and stderr files back to the submit host. See 'Configuring Data Management' in the *Torque Resource Manager Administrator Guide* for more information.

Related Topics

- [2.1.1 Preparing for Manual Installation](#)

2.1.3 Installing Moab Workload Manager

This topic contains instructions on how to install and start Moab Workload Manager (Moab HPC Suite).

In this topic:

- [2.1.3.A Open Necessary Ports](#)
- [2.1.3.B Install Dependencies, Packages, or Clients](#)
- [2.1.3.C \(Optional\) Build a Custom RPM](#)
- [2.1.3.D Install Moab Server](#)
- [2.1.3.E Configure Torque to Trust Moab](#)
- [2.1.3.F Verify the Installation](#)
- [2.1.3.G \(Optional\) Install Moab Client](#)

2.1.3.A Open Necessary Ports

If your site is running firewall software on its hosts, configure the firewall to allow connections to the necessary port:

Location	Port	Function	When Needed
Moab Server Host	42559	Moab Server Port	If you intend to run client commands on a host different from the Moab Server Host <i>or</i> if you will be using Moab in a grid

See [2.2.1 Opening Ports in a Firewall](#) for general instructions and an example of how to open ports in the firewall.

2.1.3.B Install Dependencies, Packages, or Clients

Dependencies and Packages

On the Moab Server Host, run the following command to install the required Moab HPC Suite dependencies and packages:

```
[root]# dnf install make libcurl perl-CPAN libxml2-devel gcc
```

Torque Client

If you are using Torque and are installing the Torque Server on a different host (Torque Server Host) from the Moab Server (Moab Server Host), install the Torque client on the Moab Server Host in order for Moab to interact with Torque.

Follow the instructions in [Install hwloc](#) and [2.1.2.E Install Torque Clients](#) using the Moab Server Host as the Torque Client Host, with the exception that you must copy and install the `torque-package-devel-linux-<arch>.sh` self-extracting package in addition to the `torque-package-clients-linux-<arch>.sh` package:

```
[root]# scp torque-package-devel-linux-x86_64.sh <torque-client-host>:
[root]# ./torque-package-devel-linux-x86_64.sh --install
```

2.1.3.C (Optional) Build a Custom RPM

1. Install `rpm-build`:

```
[root]# dnf install rpm-build
```

2. Download the latest tarball from the [Adaptive Computing](#) website.
3. Untar the downloaded package.
4. Change directories into the untarred directory.
5. Edit the `./moab.spec` file for RPM customization.
6. Run `./rpm-build`.
7. Locate the custom RPM in `rpm/RPMS/x86_64`.

2.1.3.D Install Moab Server

i You *must* complete the tasks to install the dependencies, packages, or clients before installing Moab Server. See [2.1.3.B Install Dependencies, Packages, or Clients](#).

If your configuration uses firewalls, you *must also* open the necessary ports before installing the Moab Server. See [2.1.3.A Open Necessary Ports](#).

On the Moab Server Host, do the following.

1. Download the latest Moab Workload Manager tarball from the [Adaptive Computing](#) website.
2. As the root user, run the following commands:

```
[root]# tar -xzvf moab-10.2.0-<OS>.tar.gz
[root]# cd moab-10.2.0-<OS>
```

i The variable marked `<OS>` indicates the OS for which the build was designed.

4. Configure Moab. If you are installing Moab Accounting Manager, configure Moab with the `--with-am` option.

```
[root]# ./configure <options>
```

i See [2.2.3 Moab Workload Manager Configuration Options](#) for a list of commonly used options, or use `./configure --help` for a complete list of available options.

5. *ONLY* if you are using green computing *or* if you are using a resource manager other than Torque, run the `make perldeps` command to install the necessary Perl modules using CPAN. When first running CPAN, you will be asked for configuration information. We recommend that you choose an automatic configuration. You will be prompted to provide input during module installation; running the `make perldeps` command with a script is not recommended.

```
[root]# make perldeps
```

6. Install Moab:

```
[root]# make install
```

7. Modify the Moab configuration file:

```
[root]# vi /opt/moab/etc/moab.cfg
```

Do one of the following:

- If using Torque Resource Manager:
 - i. Verify that `SUBMITCMD` is set up for your Torque resource manager and that it points to a valid `qsub` executable, for example:


```
RMCFG[torque] SUBMITCMD=/usr/local/bin/qsub
```
 - ii. If you installed the Torque Server on a different host (Torque Server Host), configure the `RMCFG HOST` parameter to tell Moab the host on which the Torque Server is running:


```
RMCFG[torque] HOST=<torque_server_hostname>
```
- If using a NATIVE resource manager, see 'Managing Resources Directly with the Native Interface' in the *Moab Workload Manager Administrator Guide* for configuration information.

8. Source the appropriate profile script to add the Moab HPC Suite executable directories to your current shell `$PATH` environment:

```
[root]# . /etc/profile.d/moab.sh
```

9. Copy your license file into the same directory as `moab.cfg` (`/opt/moab/etc/` by default):

```
[root]# cp moab.lic $MOABHOMEDIR/moab.lic
```

- a. To verify the current status of your license, run the following:

```
[root]# moab --about 2>&1 | grep License
```

You should get something similar to the following in the response:

```
Moab Workload Manager Version '10.2.0' License Information:
Current License:  Max Procs    = 10000
Current License:  Valid Until  - Jul 13 19:42:10 2025
```

i A license is required for Moab. A trial license may be included in your Moab installation, enabling you to run Moab for a limited time and with limited features. Email licenses@adaptivecomputing.com for information on obtaining licenses.

10. Start Moab:

```
[root]# systemctl start moab.service
```

2.1.3.E Configure Torque to Trust Moab

If you are using Torque as a resource manager and you installed the Torque Server on a different host (Torque Server Host), which we recommend, do the following.

On the Torque Server Host, add the name of the Moab Server Host (where Moab Server is installed) as a manager and as a submit host:

```
[root]# qmgr
Qmgr: set server managers += root@<moab_server_hostname>
Qmgr: set server submit_hosts += <moab_server_hostname>
Qmgr: exit
```

2.1.3.F Verify the Installation

If you have a resource manager configured, verify that the scheduler is able to schedule a job. Do the following.

Submit a sleep job as a non-root user (adaptive is used in this example) and verify the job is running:

```
[root]# su - adaptive
[adaptive]$ echo sleep 150 | msub
[adaptive]$ showq
[adaptive]$ exit
```

2.1.3.G (Optional) Install Moab Client

After you have installed Moab Server, you can create a client tarball to install just the Moab client commands on a login/client host. This tarball uses a single `tar` command to install the binary Moab client command files and their man pages. The tarball also contains a `moab.cfg` file configured with the Moab Server host name and port number so you do not have to manually configure this information on the login/client node. To create the Moab client package, the Moab service must first be stopped. Also, you need to `cd` into the `moab<build-info>` directory from the original Moab install.

i If your site needs secure communication and authentication between the Moab Client Host and the Moab Server Host, create a site-specific key and place it in the same directory as your `moab.cfg` file. By default, this would be `$MOABHOMEDIR/etc/.moab.key`. When the Moab server and client commands detect the presence of those two files, they will use the key in those files to authenticate and communicate instead of the default key. If `.moab.key` is used, this protected file needs to be on each host that is authorized to run Moab client commands. See 'Mauth Authentication' in the *Moab Workload Manager Administrator Guide* for more information.

1. On the Moab Server Host, create the client tarball:

```
[root]# make client-pkg
```

2. Copy the tarball to the root directory of the Moab Client Host.
3. Copy the `/etc/profile.d/moab.sh` to the Moab Client Host.
4. On the Moab Client Host, run the tarball to install the Moab client commands:

```
[root]# tar xvf client.tgz
```

This creates an `opt/moab/` directory in the CWD; it does not create `/opt/moab/`. To use the current `client.tgz`, you must `cd` to `/`, then `untar` the package.

5. Copy the `/opt/moab/etc/.moab.key` file to the same location on the Moab Client Host.

Related Topics

- [2.1.1 Preparing for Manual Installation](#)

2.2 Additional Configuration

In this section:

[2.2.1 Opening Ports in a Firewall](#)

[2.2.2 Configuring SSL in Tomcat](#)

[2.2.3 Moab Workload Manager Configuration Options](#)

[2.2.4 Trusting Servers in Java](#)

2.2.1 Opening Ports in a Firewall

If your site is running firewall software on its hosts, configure the firewall to allow connections to the products in your installation.

Below is an example and general instructions for how to open ports in your firewall. See [4.2 Port Reference](#) for the actual port numbers for the various products.

Red Hat-based systems use firewalld as the default firewall software. If you use different firewall software, refer to your firewall documentation for opening ports in your firewall.

The following is an example of adding port 1234 when using firewalld:

```
[root]# firewall-cmd --add-port=1234/tcp --permanent
[root]# firewall-cmd --reload
```

2.2.2 Configuring SSL in Tomcat

To configure SSL in Tomcat, refer to the Apache Tomcat [documentation](#).

2.2.3 Moab Workload Manager Configuration Options

The following is a list of commonly used configuration options. For a complete list, use `./configure --help` when configuring Moab HPC Suite.

Option	Description	Example
<code>--prefix</code>	Specifies the location of the binaries and libraries of the Moab install. The default	<pre>[root]# ./configure --prefix=/usr/local</pre>

Option	Description	Example
	location is /opt/moab.	
--with-am	Specifies that you want to configure Moab with Moab Accounting Manager.	<pre>[root]# ./configure --with-am</pre>
--with-am-dir	Uses the specified prefix directory for the accounting manager if installed in a non-default location.	<pre>[root]# ./configure --with-am-dir=/opt/mam-10.1.0</pre>
--with-flexlm	Causes Moab to install the <code>license.mon.flexLM.pl</code> script in the <code>/opt/moab/tools</code> directory. For more information about this script, see the section 'Interfacing with FlexNet (formerly FLEXlm)' in the <i>Moab Workload Manager Administrator Guide</i> .	<pre>[root]# ./configure --with-flexlm</pre>
--with-homedir	Specifies the location of the Moab configuration directory and the MOABHOMEDIR environment variable. The default location is <code>/opt/moab</code> . i By default, MOABHOMEDIR is automatically set during installation. Use the <code>--without-profile</code> option to disable installed scripts.	<pre>[root]# ./configure --with-homedir=/var/moab</pre> <i>The Moab HPC Suite home directory will be /var/moab instead of the default /opt/moab.</i>
--without-init	Disables the installation of a distribution-specific Moab service startup file. By default, 'make install' will install an <code>init.d</code> or <code>systemd</code> service startup file as appropriate for	<pre>[root]# ./configure --without-init</pre>

Option	Description	Example
	<p>your distribution. The installed file (/etc/init.d/moab or /usr/lib/systemd/system/moab.service) can be customized to your needs. If you do not want this file to be installed, use this option to exclude it.</p>	
<p>--without-profile</p>	<p>Disables the installation of a distribution-specific shell profile for Bash and C shell. By default, 'make install' will install the Moab shell initialization scripts as appropriate for your operating system. These scripts help to establish the MOABHOMEDIR, PERL5LIB, PATH, and MANPATH environment variables to specify where the new Moab configuration, scripts, binaries, and man pages reside. The installed scripts (/etc/profile.d/moab.{csh,sh}) can be customized to your needs. If you do not want these scripts to be installed, use this option to exclude them.</p>	<pre>[root]# ./configure --without-profile</pre>

2.2.4 Trusting Servers in Java

In this topic:

[2.2.4.A Prerequisites](#)

[2.2.4.B Retrieve the Server's X.509 Public Certificate](#)

[2.2.4.C Add the Server's Certificate to Java's Keystore](#)

2.2.4.A Prerequisites

Some of these instructions refer to `JAVA_HOME`, which must point to the same directory that Tomcat uses. To set `JAVA_HOME`, use the following command:

```
[root]# source /etc/tomcat/tomcat.conf
```

Your system administrator might have defined Tomcat's `JAVA_HOME` in a different file.

2.2.4.B Retrieve the Server's X.509 Public Certificate

To retrieve the server's certificate, use the following command:

```
[root]# $JAVA_HOME/bin/keytool -printcert -rfc -sslserver <servername>:<port>
/tmp/public.cert.pem
```

Replace `<servername>` with the server's host name and `<port>` with the secure port number. The default port for HTTPS is 443. The default port for LDAP is 636. If successful, `/tmp/public.cert.pem` contains the server's public certificate. Otherwise, `/tmp/public.cert.pem` contains an error message. This message is typical:
`keytool error: java.lang.Exception: No certificate from the SSL server.` This message suggests that the server name or port is incorrect. Consult your IT department to determine the correct server name and port.

2.2.4.C Add the Server's Certificate to Java's Keystore

Java stores trusted certificates in a database known as the keystore. Because each new version of Java has its own keystore, you need to add the server certificate to the Java keystore (using the steps below) every time you install a new version of Java.

Java's keystore is located at `$JAVA_HOME/lib/security/cacerts`. If Tomcat's `JAVA_HOME` points to a JDK, then the keystore is located at `$JAVA_HOME/jre/lib/security/cacerts`. To add the server certificate to the keystore, run the following:

```
[root]# $JAVA_HOME/bin/keytool -import -trustcacerts -file /tmp/public.cert.pem -alias
<servername> -keystore $JAVA_HOME/lib/security/cacerts
```

You will be prompted for the keystore password, which is 'changeit' by default.

 Your system administrator might have changed this password.


After you have entered the keystore password, you will see the description of the server's certificate. At the end of the description, it prompts you to trust the certificate:


```
Trust this certificate? [no]:
```

Type `yes` and press `Enter` to add the certificate to the keystore.

2.3 Manual Upgrade Steps

This section provides instructions and other information when upgrading your Moab HPC Suite components using the Manual upgrade method.

 We highly recommend that you *first* perform upgrades in a *test environment*. Installation and upgrade procedures are tested prior to release; however, due to customizable variations that may be utilized by your configuration, it is not recommended to drop new versions of software directly into production environments. This is especially true when the workload has a vital bearing. Contact Adaptive Computing for more information.


 Because many system-level files and directories are accessed during the upgrade, the upgrade instructions in this guide should be executed with root privileges. You will see that the instructions execute commands as the root user. Note that the same commands will work for a non-root user with the `sudo` command.

In this section:

[2.3.1 Upgrading Torque Resource Manager](#)

[2.3.2 Upgrading Moab Workload Manager](#)

2.3.1 Upgrading Torque Resource Manager

 See 'Considerations Before Upgrading' in the *Torque Resource Manager Administrator Guide* for additional important information, including about how to handle running jobs during an upgrade, mixed server/MOM versions, and the possibility of upgrading the MOMs without having to take compute nodes offline.

In this topic:

[2.3.1.A Before You Upgrade](#)

[2.3.1.B Stop Torque Services](#)

- 2.3.1.C Upgrade the Torque Server
- 2.3.1.D Update the Torque MOMs
- 2.3.1.E Update the Torque Clients
- 2.3.1.F Start Torque Services
- 2.3.1.G Perform Status and Error Checks

2.3.1.A Before You Upgrade

This section contains information you should be aware of before upgrading.

Running Jobs

Before upgrading the system, all running jobs must complete. To prevent queued jobs from starting, nodes can be set to offline, or all queues can be disabled (using the `started` queue attribute). See 'pbsnodes' or 'Queue Attributes' in the *Torque Resource Manager Administrator Guide* for more information.

hwloc

Torque version 7.0.1 and later permit and recommend the use of the OS vendor-provided version of hwloc.

1. If you are upgrading from a version of Torque earlier than 7.0.1, on the Torque Server Host, each Torque MOM Host, and each Torque Client Host, run the following:

```
[root]# dnf install hwloc-devel
[root]# cd hwloc-1.9.1
[root]# make uninstall
```

i If using RHEL 8, first enable the CodeReady Linux Builder repository before installing the hwloc development package:

```
[root]# subscription-manager repos --enable codeready-builder-for-rhel-8-
x86_64-rpms
```

If using (non-RHEL) 8, first enable the PowerTools repository before installing the hwloc development package:

```
[root]# dnf config-manager --set-enabled powertools
```

2. On the Torque Server Host, run the following commands:

```
[root]# rm /etc/ld.so.conf.d/hwloc.conf
[root]# ldconfig
```

2.3.1.B Stop Torque Services

1. On the Torque Server Host, shut down the Torque server:

```
[root]# systemctl stop pbs_server.service
```

2. On *each* host where the Torque MOM Host resides (regardless of whether it resides on the Torque Server Host), shut down the Torque MOM service.



Confirm all jobs have completed before stopping `pbs_mom`. You can do this by typing `momctl -d3`. If there are no jobs running, you will see the message 'NOTE: no local jobs detected' towards the bottom of the output. If jobs are still running and the MOM is shut down, you will only be able to track when the job completes, and you will not be able to get completion codes or statistics.

```
[root]# systemctl stop pbs_mom.service
```

3. On *each* host where the Torque Client Host resides (regardless of whether it resides on the Moab Server Host, the Torque Server Host, or the Torque MOM Hosts), shut down the `trqauthd` service:

```
[root]# systemctl stop trqauthd.service
```

2.3.1.C Upgrade the Torque Server



You *must* complete all the previous upgrade steps in this topic before upgrading the Torque server. See the list of steps at the beginning of this topic.

On the Torque Server Host, do the following.

1. Back up your `server_priv` directory:

```
[root]# tar -cvf backup.tar.gz TORQUE_HOME/server_priv
```

2. Download the latest Torque tarball from the [Adaptive Computing](#) website.
3. Depending on your system configuration, you will need to add `./configure` command options. At a minimum, you add `--enable-cgroups`.



These instructions assume you are using `cgroups`.

See 'Customizing the Install' in the *Torque Resource Manager Administrator Guide* for more information on which options are available to customize the `./configure` command.

4. Install the latest Torque tarball:

```
[root]# tar xzvf torque-7.2.0.tar.gz
[root]# cd torque-7.2.0
[root]# ./configure --enable-cgroups # add any other required options
[root]# make
[root]# make install
```

2.3.1.D Update the Torque MOMs

1. On the Torque Server Host, do the following:

- a. Create the self-extracting packages that are copied and executed on your nodes:

```
[root]# make packages
```

- b. Copy the self-extracting MOM package to *each* Torque MOM Host. We recommend that you use a remote shell, such as SSH, to install packages on remote systems. Set up shared SSH keys if you do not want to supply a password for each Torque MOM Host.

```
[root]# scp torque-package-mom-linux-x86_64.sh <torque-mom-host>:
```

- c. Copy the pbs_mom startup script to each Torque MOM Host:

```
[root]# scp contrib/systemd/pbs_mom.service <mom-node>:/usr/lib/systemd/system/
```

2. On *each* Torque MOM Host, install the self-extracting MOM package:

```
[root]# ./torque-package-mom-linux-x86_64.sh --install
```

2.3.1.E Update the Torque Clients

This section contains instructions on updating the Torque clients on the Torque Client Hosts (including the Moab Server Host and Torque MOM Hosts, if applicable).

1. On the Torque Server Host, do the following:

- a. Copy the self-extracting client package to *each* Torque Client Host. We recommend that you use a remote shell, such as SSH, to install packages on remote systems. Set up shared SSH keys if you do not want to supply a password for each Torque MOM Host.

```
[root]# scp torque-package-clients-linux-x86_64.sh <torque-client-host>:
```

- b. If Moab Workload Manager is part of your configuration, copy the self-extracting devel package to the Moab Server Host:

```
[root]# scp torque-package-devel-linux-x86_64.sh <moab-server-host>:
```

c. Copy the `trqauthd` startup script to each Torque Client Host:

```
[root]# scp contrib/systemd/trqauthd.service <torque-client-host>:/usr/lib/systemd/system/
```

2. On *each* Torque Client Host, do the following.

i This step can be done from the Torque server from a remote shell, such as SSH. Set up shared SSH keys if you do not want to supply a password for each Torque Client Host.

```
[root]# ./torque-package-clients-linux-x86_64.sh --install
```

3. If Moab Workload Manager is part of your configuration, run the following command on the Moab Server Host:

```
[root]# ./torque-package-devel-linux-x86_64.sh --install
```

2.3.1.F Start Torque Services

1. On the Torque Server Host, start up the Torque server:

```
[root]# systemctl daemon-reload
[root]# systemctl start pbs_server.service
```

2. On *each* Torque MOM Host, start up the Torque MOM service:

```
[root]# systemctl daemon-reload
[root]# systemctl start pbs_mom.service
```

3. On *each* Torque Client Host (including the Moab Server Host, Torque Server Host, and Torque MOM Hosts, if applicable), start up the `trqauthd` service:

```
[root]# systemctl daemon-reload
[root]# systemctl start trqauthd.service
```

2.3.1.G Perform Status and Error Checks

On the Torque Server Host, verify that the status of the nodes and jobs are as expected:

```
[root]# pbsnodes
[root]# qstat
```

2.3.2 Upgrading Moab Workload Manager

This topic provides instructions to upgrade Moab Workload Manager to the latest release version. Depending on which version of Moab you are presently running, upgrade instructions may vary.

Moab Workload Manager uses the standard configure, make, and make install steps for upgrades. This topic provides several sample steps referenced to a particular installation on a Linux platform using the Bash shell. These steps indicate the user ID in brackets performing the step. The exact commands to be performed and the user that issues them will vary based on the platform, shell, installation preferences, and other factors.

In this topic:

[2.3.2.A Recommendations](#)

[2.3.2.B Upgrade Moab Workload Manager](#)

2.3.2.A Recommendations



We highly recommend that you *first* perform upgrades in a *test environment*. We also recommend that you verify the policies, scripts, and queues work the way you want them to in this test environment. See 'Testing New Releases and Policies' in the *Moab Workload Manager Administrator Guide* for more information.

2.3.2.B Upgrade Moab Workload Manager

On the Moab Server Host, do the following.

1. Download the latest Moab Workload Manager tarball from the [Adaptive Computing](#) website.
2. Untar the distribution file, for example:

```
[root]# tar -xzvf moab-10.2.0-<OS>.tar.gz
```



The variable marked <OS> indicates the OS for which the build was designed.

3. Change directory into the extracted directory:

```
[root]# cd moab-10.2.0-<OS>
```

- Configure the installation package. Use the same configuration options as when Moab was installed previously. If you cannot remember which options were used previously, check the `config.log` file in the directory where the previous version of Moab was installed from. For a complete list of configuration options, use `./configure --help`.

- Stop Moab:

```
[root]# systemctl stop moab.service
```

i While Moab is down, all currently running jobs continue to run on the nodes, the job queue remains intact, and new jobs cannot be submitted to Moab.

- Back up your Moab Workload Manager home directory (`/opt/moab/` by default) before continuing.

- Install Moab:

```
[root]# make install
```

i Default configuration files are installed during `make install`. Existing configuration files are not overwritten, and the new files are given a `.dist` extension.

- If you use ODBC, you must confirm the database schema compatibility. See 'Migrating Your Database to Newer Versions of Moab' in the *Moab Workload Manager Administrator Guide* for more information.

- Verify the version number is correct before starting the new server version:

```
[root]# moab --about
```

You should get something similar to the following in the response:

```
Defaults:  server=:42559  cfgdir=/opt/moab (env)  vardir=/opt/moab
Build dir:  /home/autobuild/jenkins/workspace/MWM-10.2.0/label/build-<OS>
Build host: us-devops-build10
Build date: Thu Feb 26 12:00:00 MST 2026
Build args: NA
Compiler Flags:  -D_M64 -D_BUILDDATETIME="2026100913" -DMUSEZEROMQ -
DMUSEWEBSERVICES -DMUSEMONGODB -DMMAX_GRES=512 -DMMAX_RANGE=2048 -DMMAX_TASK=32768
-fPIC -gdwarf-3 -Wall -Wextra -DVALGRIND -Og -x c++ -std=c++11 -DDMAX_PJOB=512 -D_
GNU_SOURCE
Compiled as little endian.
Version: moab server 10.2.0 (revision 2026100913, changeset
84fb128e55ff5acf247c1159f4ff98c749ddda4a)
```

- Start Moab:

```
[root]# systemctl daemon-reload  
[root]# systemctl start moab.service
```

Chapter 3: RPM Installation Method

This chapter contains an introduction to the RPM Installation method and explains how to prepare your component hosts (physical machines in your cluster) for the RPM installations and upgrades. Information and configuration information for each Moab HPC Suite product or module using the RPM Installation method, is also provided.

In this chapter:

- [3.1 About RPM Installations and Upgrades](#)
- [3.2 RPM Installations](#)
- [3.3 Additional Configuration](#)
- [3.4 RPM Upgrades](#)

3.1 About RPM Installations and Upgrades

This section contains information useful to know and understand when using RPMs for installation and upgrading.

Adaptive Computing provides RPMs to install or upgrade the various component servers (such as Moab HPC Suite Server, MWS Server, and Torque Server). The Moab HPC Suite RPM bundle contains all the RPMs for the Moab HPC Suite components and modules. However, not every component may be installed or upgraded on the same host (for example, we recommend that you install the Torque Server on a different host from the Moab Server).

In this section:

- [3.1.1 RPM Installation and Upgrade Methods](#)
- [3.1.2 Special Considerations](#)
- [3.1.3 Installation and Upgrade Process](#)

3.1.1 RPM Installation and Upgrade Methods

Depending on your configuration, you may install many servers on a single host or a single server on its own host. In addition, you can install various clients and GUIs on the same host you installed the server on or on another host. For example, you have the Moab HPC

Suite Server and the MWS Server on the same host (required), and you install the Torque Server on a different host (recommended).

i Be aware that the same host may be called by different names. For example, even though the Moab Server and the MWS Server are installed on the same host, the MWS instructions will call it the MWS Server Host, not the Moab Server Host.

i The RPM install process assumes that the Moab HPC Suite components are the only software on the Moab Server host and all components will install on that one server. We strongly recommend that all other software be removed from the Moab HPC Suite host. If you have a more complex configuration of Moab, we recommend that you contact support to discuss considerations before using the RPM installation to upgrade.

3.1.2 Special Considerations

Be aware of the following:

- On RHEL systems, you must be registered for a Red Hat subscription in order to have access to required RPM package dependencies.
- Because many system-level files and directories are accessed during the installation, the instructions in this guide should be executed with root privileges. You will see that the instructions execute commands as the root user. Also be aware that the same commands will work for a non-root user with the `sudo` command.

3.1.3 Installation and Upgrade Process

Each host (physical machine) will need to have the Moab HPC Suite RPM bundle and the Adaptive Computing repository enabled. This is referred to as preparing the host.

Once each host has been prepared, you can install or upgrade the individual components on the designated hosts.

3.2 RPM Installations

This section provides instructions and other information for installing your Moab HPC Suite components using the RPM installation method.

In this section:

- [3.2.1 Preparing the Host \(RPM\)](#)
- [3.2.2 Installing Torque Resource Manager \(RPM\)](#)
- [3.2.3 Installing Moab Workload Manager \(RPM\)](#)
- [3.2.4 Disabling the Adaptive Repository after Installs \(RPM\)](#)

3.2.1 Preparing the Host (RPM)

This topic contains instructions on how to download the Moab HPC Suite RPM bundle and enable the Adaptive Computing repository for all the hosts in your configuration.

The Moab HPC Suite RPM bundle contains all the RPMs for the Moab HPC Suite components and modules. However, not every component may be installed on the same host (for example, we recommend that you install the Torque Server on a different host from the Moab Server).

i Whether you are installing RPMs on one host or on several hosts, each host (physical machine) on which a server is installed (Torque Server Host, Moab Server Host, etc.) *must* have the Adaptive Computing Package Repository enabled.

On each host (physical machine), do the following.

1. If your site uses a proxy to connect to the Internet, run the following commands:

```
export http_proxy=https://<proxy_server_id>:<port>
export https_proxy=https://<proxy_server_id>:<port>
```

2. Many individual components have dependencies that are found in the optional add-on repositories for the distribution. You must enable the respective repository for your distribution on all hosts upon which you install Adaptive Computing software components. Do the following:

- On RHEL systems, you must be registered for a Red Hat subscription, enable the codeready-builder repository and temporarily enable the EPEL repository for your major version:

```
[root]# subscription-manager repos --enable codeready-builder-for-rhel-9-x86_64-rpms
[root]# rpm -Uvh https://dl.fedoraproject.org/pub/epel/epel-release-latest-9.noarch.rpm
```

- On non-RHEL Red Hat-based systems (e.g., Rocky Linux, CentOS, AlmaLinux, and

Scientific Linux), install the EPEL release package and enable the crb or powertools repository for your release:

```
[root]# dnf install epel-release
[root]# dnf config-manager --set-enabled crb # use powertools or PowerTools for Red Hat 8
```

3. Update your system software to the latest version:

```
[root]# dnf update
```

4. Ensure hostname resolution for all hosts. Each host should be resolvable from all other hosts in the cluster. Usually this is implemented by having all hosts in DNS.

Alternatively, each host can include all other hosts (with the correct IP address) in its `/etc/hosts` file.

5. Download the Moab HPC Basic Suite RPM bundle from the [Adaptive Computing](#) website.

6. Untar the RPM bundle:

```
[root]# tar -zxvf moab-hpc-basic-suite-10.2.0-<OS>.tar.gz
```

i The variable marked `<OS>` indicates the OS for which the build was designed.

7. Change directories into the untarred directory:

```
[root]# cd moab-hpc-basic-suite-10.2.0-<OS>
```

8. Install the suite repositories. The `-y` option installs with the default settings for the RPM suite.

i For a description of the options of the repository installer script, run:

```
[root]# ./install-rpm-repos.sh -h
```

```
[root]# ./install-rpm-repos.sh [<repository-directory>] [-y]
```

The `[<repository-directory>]` option is the directory where you want to copy the RPMs. If no argument is given, run `install-rpm-repos.sh -h` to view usage information and identify the default directory location. If the `[<repository-directory>]` already exists, RPMs will be added to the existing directory. No files are overwritten in `[<repository-directory>]`.

A repository file is also created and points to the `[<repository-directory>]` location.

The repository file is created in `/etc/dnf.repos.d/`.

For ease in repository maintenance, the install script fails if Adaptive Computing RPMs are copied to different directories. If a non-default [`<repository-directory>`] is specified, use the same directory for future updates.

The script installs the `createrepo` package and its dependencies. You must answer 'y' to all the questions in order for the RPM install of the suite to work.

Additionally, the script installs the EPEL and 10gen repositories.

9. Test the repository:

```
[root]# dnf search moab
```

If no error is given, the repository is correctly installed. The following is an example of the output after verifying the repository:

```
...
moab-hpc-basic-suite.noarch : Moab HPC Basic Suite virtual package
moab-workload-manager.x86_64 : Moab Workload Manager
moab-workload-manager-client.x86_64 : Moab Workload Manager Client
moab-workload-manager-common.x86_64 : Moab Workload Manager Common Files
moab-torque-client.x86_64 : TORQUE Client
moab-torque-common.x86_64 : TORQUE Common Files
moab-torque-devel.x86_64 : TORQUE Development Files
moab-torque-mom.x86_64 : TORQUE MOM agent
moab-torque-server.x86_64 : TORQUE Server
moab-workload-manager-hpc-configuration.x86_64 : MWM configuration for HPC
...
```

10. Continue with instructions to install the Moab HPC Suite components. See [3.2 RPM Installations](#).

3.2.2 Installing Torque Resource Manager (RPM)

This topic contains instructions on how to install, configure, and start Torque Resource Manager (Torque).

In this topic:

[3.2.2.A Open Necessary Ports](#)

[3.2.2.B Install Torque Server](#)

[3.2.2.C Install Torque MOMs](#)

[3.2.2.D Configure Data Management](#)

3.2.2.A Open Necessary Ports

If your site is running firewall software on its hosts, configure the firewall to allow connections to the necessary ports:

Location	Port	Function	When Needed
Torque Server Host	15001	Torque Client and MOM communication to Torque Server	Always
Torque MOM Host (Compute Nodes)	15002	Torque Server communication to Torque MOMs	Always
Torque MOM Host (Compute Nodes)	15003	Torque MOM communication to other Torque MOMs	Always

If using the MOM hierarchy (documented in 'Setting Up the MOM Hierarchy' in the *Torque Resource Manager Administrator Guide*), you must also open port 15003 from the server to the nodes.

See also:

- [2.2.1 Opening Ports in a Firewall](#) for general instructions and an example of how to open ports in the firewall.
- 'Configuring Ports' in the *Torque Resource Manager Administrator Guide* for more information on how to configure the ports that Torque uses for communication.

3.2.2.B Install Torque Server

i If your configuration uses firewalls, you *must* open the necessary ports before installing the Torque Server. See [3.2.2.A Open Necessary Ports](#).

On the Torque Server Host, do the following.

1. If you are installing the Torque Server on its own host (recommended) and *not* on the same host where you installed another server (such as Moab Server), verify you completed the steps to prepare the host. See [3.2.1 Preparing the Host \(RPM\)](#).

2. Install the Torque Server RPM:

```
[root]# dnf install moab-torque-server
```

3. Source the following file to add the Torque executable directories to your current shell \$PATH environment:

```
[root]# ./etc/profile.d/torque.sh
```

4. Add the hostnames of your Torque MOMs (which are commonly all of your compute nodes) to the `/var/spool/torque/server_priv/nodes` file. You can remove the hostname entry for the Torque server node *unless* you will be running a Torque MOM daemon on this host. See 'Managing Nodes' in the *Torque Resource Manager Administrator Guide* for information on syntax and options for specifying compute nodes.

Example:

```
[root]# vi /var/spool/torque/server_priv/nodes
node01 np=16
node02 np=16
...
```

5. Start the Torque server:

```
[root]# systemctl start pbs_server.service
[root]# systemctl start trqauthd.service
```

3.2.2.C Install Torque MOMs

In most installations, you will install a Torque MOM on each of your compute nodes.

1. From the Torque Server Host, copy the `moab-torque-common`, and `moab-torque-mom` RPM files to each MOM node. We also recommend that you install the `moab-torque-client` RPM so you can use client commands and submit jobs from compute nodes.

```
[root]# scp <dir>/RPMs/moab-torque-common-*.rpm <torque-mom-host>:
[root]# scp <dir>/RPMs/moab-torque-mom-*.rpm <torque-mom-host>:
[root]# scp <dir>/RPMs/moab-torque-client-*.rpm <torque-mom-host>:
```

2. On *each* Torque MOM Host, install the RPMs in the order shown:

```
[root]# dnf install moab-torque-common-*.rpm moab-torque-mom-*.rpm moab-torque-client-*.rpm
```

3. On *each* Torque MOM Host, create or edit the `/var/spool/torque/server_name` file to contain the hostname of the Torque server:

```
[root]# echo <torque_server_hostname> > /var/spool/torque/server_name
```

4. On each Torque MOM Host, start the `pbs_mom` daemon:

```
[root]# systemctl start pbs_mom.service
```

5. If you installed the Torque Client RPM on the MOMs, then on each Torque MOM Host,

start the trqauthd daemon:

```
[root]# systemctl start trqauthd.service
```

3.2.2.D Configure Data Management

When a batch job completes, stdout and stderr files are generated and placed in the spool directory on the master Torque MOM Host for the job instead of the submit host. You can configure the Torque batch environment to copy the stdout and stderr files back to the submit host. See 'Configuring Data Management' in the *Torque Resource Manager Administrator Guide* for more information.

3.2.3 Installing Moab Workload Manager (RPM)

This topic contains instructions on how to install, configure, and start Moab Workload Manager (Moab HPC Suite).

In this topic:

- [3.2.3.A Open Necessary Ports](#)
- [3.2.3.B Install Moab Server](#)
- [3.2.3.C Configure Torque to Trust Moab](#)
- [3.2.3.D Verify the Installation](#)

3.2.3.A Open Necessary Ports

If your site is running firewall software on its hosts, configure the firewall to allow connections to the necessary port:

Location	Port	Function	When Needed
Moab Server Host	42559	Moab Server Port	If you intend to run client commands on a host different from the Moab Server Host <i>or</i> if you will be using Moab in a grid.

See [2.2.1 Opening Ports in a Firewall](#) for general instructions and an example of how to open ports in the firewall.

3.2.3.B Install Moab Server

On the Moab HPC Suite Server Host, do the following.

1. If you have not already done so, complete the steps to prepare the Moab Server Host. See [3.2.1 Preparing the Host \(RPM\)](#).
2. Install RPM packages.

- a. Install the Moab Server RPMs:

```
[root]# dnf install moab-workload-manager moab-workload-manager-hpc-configuration
```

- b. If you are using Torque as a resource manager and installed the Torque Server on a different host (Torque Server Host, which we recommend) from the Moab Server (Moab Server Host), install the Torque client RPM on the Moab Server Host in order for Moab to interact with Torque:

```
[root]# dnf install moab-torque-client
```

- c. If you are using Moab Accounting Manager and will be using the Native (custom script) accounting manager interface, and are installing the Moab Accounting Manager Server on a different host from the Moab Server (Moab Server Host), you will need to install Moab Accounting Manager client on the Moab Server Host in order for the custom scripts to use the MAM API:

```
[root]# dnf install moab-accounting-manager
```

3. Source the following file to add the Moab HPC Suite executable directories to your current shell `$PATH` environment:

```
[root]# . /etc/profile.d/moab.sh
```

4. Copy your license file into the same directory as `moab.cfg` (`/opt/moab/etc/` by default), for example:

```
[root]# cp moab.lic $MOABHOMEDIR/etc/moab.lic
```

- a. To verify the current status of your license, run the following:

```
[root]# moab --about 2>&1 | grep License
```

You should get something similar to the following in the response:

```
Moab Workload Manager Version '10.2.0' License Information:
Current License: Max Procs    = 10000
Current License: Valid Until - Jul 13 19:42:10 2025
```

i A license is required for Moab. A trial license may be included in your Moab installation, enabling you to run Moab for a limited time and with limited features. Email licenses@adaptivecomputing.com for information on obtaining licenses.

5. If you are using Torque as your resource manager and you installed the Torque Server on a different host (Torque Server Host) from the Moab Server (Moab Server Host), do the following:

- a. Create or edit the `/var/spool/torque/server_name` file to contain the hostname of the Torque Server:

```
[root]# echo <Torque_server_hostname> > /var/spool/torque/server_name
```

- b. Verify that the Torque Server hostname used is *exactly* the name returned by a reverse hostname lookup:

```
[root]# cat /var/spool/torque/server_name | perl -lpe '$_(gethostbyname($_))
[0]'
```

If different, take the necessary steps to make them match. For example, it may be necessary to add the Torque Server hostname to the `/etc/hosts` file on the Moab Server Host:

```
[root]# vi /etc/hosts
<Torque_server_ip_address><Torque_server_FQDN><Torque_server_hostname>
```

- c. Start the `trqauthd` daemon:

```
[root]# systemctl start trqauthd.service
```

6. Start Moab:

```
[root]# systemctl start moab.service
```

3.2.3.C Configure Torque to Trust Moab

If you are using Torque as a resource manager and you installed the Torque Server on a different host (Torque Host), which we recommend, do the following.

On the Torque Host, add the name of the Moab Server Host (where Moab Server is installed) as a manager and as a submit host:

```
[root]# qmgr
Qmgr: set server managers += root@<moab_server_hostname>
Qmgr: set server submit_hosts += <moab_server_hostname>
Qmgr: exit
```

3.2.3.D Verify the Installation

If you have a resource manager configured, verify that the scheduler is able to schedule a job. Do the following.

Submit a sleep job as a non-root user (adaptive is used in this example) and verify the job is running:

```
[root]# su - adaptive
[adaptive]$ echo sleep 150 | msub
[adaptive]$ showq
[adaptive]$ exit
```

3.2.4 Disabling the Adaptive Repository after Installs (RPM)

After you have completed the installation of your Moab HPC Suite components, we recommend that you disable the adaptive repository so that subsequent general system software updates do not inadvertently upgrade your Moab HPC Suite components.

On *each* host where you have enabled the adaptive repository, run the following:

```
[root]# dnf config-manager --set-disabled adaptive
```

If installing on RHEL and you enabled the EPEL repository as specified in [Preparing the Host \(RPM\)](#), we recommend that you disable it after installing all of your Adaptive software packages:

```
[root]# dnf config-manager --set-disabled epel
```

3.3 Additional Configuration

In this section:

- [3.3.1 Opening Ports in a Firewall \(RPM\)](#)
- [3.3.2 Configuring SSL in Tomcat \(RPM\)](#)
- [3.3.3 Trusting Servers in Java \(RPM\)](#)
- [3.3.4 Updating the Reporting Application Configuration \(RPM\)](#)

3.3.1 Opening Ports in a Firewall (RPM)

If your site is running firewall software on its hosts, configure the firewall to allow connections to the products in your installation.

Below is an example and general instructions for how to open ports in your firewall. See [4.2 Port Reference](#) for the actual port numbers for the various products.

Red Hat-based systems use firewalld as the default firewall software. If you use different firewall software, refer to your firewall documentation for opening ports in your firewall.

The following is an example of adding port 1234 when using firewalld:

```
[root]# firewall-cmd --add-port=1234/tcp --permanent
[root]# firewall-cmd --reload
```

3.3.2 Configuring SSL in Tomcat (RPM)

To configure SSL in Tomcat, refer to the Apache Tomcat [documentation](#).

3.3.3 Trusting Servers in Java (RPM)

In this topic:

[3.3.3.A Prerequisites](#)

[3.3.3.B Retrieve the Server's X.509 Public Certificate](#)

[3.3.3.C Add the Server's Certificate to Java's Keystore](#)

3.3.3.A Prerequisites

Some of these instructions refer to `JAVA_HOME`, which must point to the same directory that Tomcat uses. To set `JAVA_HOME`, use the following command:

```
[root]# source /etc/tomcat/tomcat.conf
```

Your system administrator might have defined Tomcat's `JAVA_HOME` in a different file.

3.3.3.B Retrieve the Server's X.509 Public Certificate

To retrieve the server's certificate, use the following command:

```
[root]# $JAVA_HOME/bin/keytool -printcert -rfc -sslserver <servername>:<port>
/tmp/public.cert.pem
```

Replace `<servername>` with the server's host name and `<port>` with the secure port number. The default port for HTTPS is 443. The default port for LDAP is 636. If successful, `/tmp/public.cert.pem` contains the server's public certificate. Otherwise, `/tmp/public.cert.pem` contains an error message. This message is typical:
`keytool error: java.lang.Exception: No certificate from the SSL server.` This message suggests that the server name or port is incorrect. Consult your IT department to determine the correct server name and port.

3.3.3.C Add the Server's Certificate to Java's Keystore

Java stores trusted certificates in a database known as the keystore. Because each new version of Java has its own keystore, you need to add the server certificate to the Java keystore (using the steps below) every time you install a new version of Java.

Java's keystore is located at `$JAVA_HOME/lib/security/cacerts`. If Tomcat's `JAVA_HOME` points to a JDK, then the keystore is located at `$JAVA_HOME/jre/lib/security/cacerts`. To add the server certificate to the keystore, run the following:

```
[root]# $JAVA_HOME/bin/keytool -import -trustcacerts -file /tmp/public.cert.pem -alias
<servername> -keystore $JAVA_HOME/lib/security/cacerts
```

You will be prompted for the keystore password, which is 'changeit' by default.

i Your system administrator might have changed this password.

After you have entered the keystore password, you will see the description of the server's certificate. At the end of the description, it prompts you to trust the certificate:

```
Trust this certificate? [no]:
```

Type `yes` and press `Enter` to add the certificate to the keystore.

3.3.4 Updating the Reporting Application Configuration (RPM)

Once the Reporting application has been started, if you need to make changes to the Reporting configuration file (`/opt/reporting/application.conf`), you must perform the following steps for the changes to take effect.

1. Make the desired changes to `application.conf`. On the Reporting Master host (which executes the Spark Master service), open `application.conf` in `vi`:

```
[root]# vi /opt/reporting/application.conf
```

2. Kill the ReportingDataProcessing Spark application:
 - a. Open the Spark Master UI by opening `<reporting_master_host>:8082` in a browser.
 - b. Locate the Reporting Data Processing row in the Running Applications section.
 - c. Click the (kill) link to the left of the name `ReportingDataProcessing`.

3. Upload the modified script to the Hadoop file system by running the following on the Reporting Master host:

```
[root]# source /etc/profile.d/hadoop.sh
[root]# /opt/reporting/upload-reporting.sh
```

4. Restart the Tomcat service where RWS (Reporting Web Services) is deployed. On the RWS Server Host (typically the same as the MWS Server host), run the following:

```
[root]# systemctl restart tomcat
```

Note that Tomcat may take several minutes to restart.

5. Verify that the ReportingDataProcessing Spark application is running by opening `<reporting_master_host>:8082` in a browser. In the Running Applications section, you should see ReportingDataProcessing is in a RUNNING state.

3.4 RPM Upgrades

This section provides instructions and other information when upgrading your Moab HPC Suite components using the RPM upgrade method.

In this section:

- [3.4.1 Preparing the Host \(RPM\)](#)
- [3.4.2 Upgrading Torque Resource Manager \(RPM\)](#)
- [3.4.3 Upgrading Moab Workload Manager \(RPM\)](#)
- [3.4.4 Disabling the Adaptive Repository after Upgrades \(RPM\)](#)

3.4.1 Preparing the Host (RPM)

This topic contains instructions on how to download the Moab HPC Suite RPM bundle and enable the Adaptive Computing repository for all the hosts in your configuration.

The Moab HPC Suite RPM bundle contains all the RPMs for the Moab HPC Suite components and modules. However, not every component may be upgraded on the same host (for example, we recommend that you upgrade the Torque Server on a different host from the Moab Server).

i Whether you are upgrading RPMs on one host or on several hosts, each host (physical machine) on which a server is installed (Torque Server Host, Moab Server Host, etc.) *must* have the Adaptive Computing Package Repository enabled.

On each host (physical machine), do the following.

1. If your site uses a proxy to connect to the Internet, run the following commands:

```
export http_proxy=https://<proxy_server_id>:<port>
export https_proxy=https://<proxy_server_id>:<port>
```

2. Download the Moab HPC Basic Suite RPM bundle from the [Adaptive Computing](#) website.

3. Untar the RPM bundle:

```
[root]# tar -zxvf moab-hpc-basic-suite-10.2.0-<OS>.tar.gz
```

i The variable marked <OS> indicates the OS for which the build was designed.

4. Change directories into the untarred directory:

```
[root]# cd moab-hpc-basic-suite-10.2.0-<OS>
```

5. Install the suite repositories. The `-y` option installs with the default settings for the RPM suite.

i For a description of the options of the repository installer script, run:

```
[root]# ./install-rpm-repos.sh -h
```

```
[root]# ./install-rpm-repos.sh [<repository-directory>] [-y]
```

The [`<repository-directory>`] option is the directory where you want to copy the RPMs. If no argument is given, run `install-rpm-repos.sh -h` to view usage information and identify the default directory location. If the [`<repository-directory>`] already exists, RPMs will be added to the existing directory. No files are overwritten in [`<repository-directory>`].

A repository file is also created and points to the [`<repository-directory>`] location.

The repository file is created in `/etc/dnf.repos.d/`.

For ease in repository maintenance, the install script fails if Adaptive Computing RPMs are copied to different directories. If a non-default [`<repository-directory>`] is specified, use the same directory for future updates.

The script installs the `createrepo` package and its dependencies. You must answer 'y' to all the questions in order for the RPM install of the suite to work.

Additionally, the script installs the EPEL and 10gen repositories.

6. Test the repository:

```
[root]# dnf search moab
```

If no error is given, the repository is correctly installed. The following is an example of the output after verifying the repository:

```
...
moab-hpc-basic-suite.noarch : Moab HPC Basic Suite virtual package

moab-workload-manager.x86_64 : Moab Workload Manager
moab-workload-manager-client.x86_64 : Moab Workload Manager Client
moab-workload-manager-common.x86_64 : Moab Workload Manager Common Files
moab-workload-manager-hpc-configuration.x86_64 : MWM configuration for HPC
moab-torque-client.x86_64 : Torque Client
moab-torque-common.x86_64 : Torque Common Files
moab-torque-devel.x86_64 : Torque Development Files
moab-torque-mom.x86_64 : Torque MOM agent
moab-torque-server.x86_64 : Torque Server
...
```

7. Continue with instructions to upgrade the Moab HPC Suite components. See [3.1.3 Installation and Upgrade Process](#) for more information.

3.4.2 Upgrading Torque Resource Manager (RPM)

This topic provides instructions to upgrade Torque Resource Manager to the latest release version using the RPM upgrade method. It includes instructions for migrating your database schema to a new version, if necessary.

i When you upgrade to Torque 7.2 from versions prior to 7.0, the server, moms, and clients must be upgraded at the same time. The job format is compatible between Torque 7.2 and previous versions of Torque. After upgrading, any queued jobs should continue to work with the new version.

i Because many system-level files and directories are accessed during the installation, the instructions in this guide should be executed with root privileges. You will see that the instructions execute commands as the root user. Note that the same commands will work for a non-root user with the `sudo` command.

In this topic:

- [3.4.2.A Upgrade Steps](#)
- [3.4.2.B Stop Torque Services](#)
- [3.4.2.C Upgrade Torque Server, MOMs, and Clients](#)
- [3.4.2.D Start Torque Services](#)

3.4.2.A Upgrade Steps

1. If you installed Torque Server on its own host *or* if Torque Server is the first component being upgraded on a host with other RPM installations, complete the steps to prepare the host. See [3.2.1 Preparing the Host \(RPM\)](#). Do the same as needed for each Torque MOM Host (compute node).
2. Stop all Torque Server, Torque MOM, and Torque Client Services. See [3.4.2.B Stop Torque Services](#).
3. Upgrade Torque Server, Torque MOMs, and Torque Clients. See [3.4.2.C Upgrade Torque Server, MOMs, and Clients](#).
4. Start all Torque Server, Torque MOM, and Torque Client Services. See [3.4.2.D Start Torque Services](#).

3.4.2.B Stop Torque Services

1. On the Torque Server Host, shut down the Torque server:

```
[root]# systemctl stop pbs_server.service
```

2. On *each* host where the Torque MOM Host resides (regardless of whether it resides on the Torque Server Host), shut down the Torque MOM service.



Confirm all jobs have completed before stopping `pbs_mom`. You can do this by typing `momctl -d3`. If there are no jobs running, you will see the message 'Note: no local jobs detected' towards the bottom of the output. If jobs are still running and the MOM is shutdown, you will only be able to track when the job completes and you will not be able to get completion codes or statistics.

```
[root]# systemctl stop pbs_mom.service
```

3. On *each* host where the Torque Client Host resides (regardless of whether it resides on the Moab Server Host, the Torque Server Host, or the Torque MOM Hosts), shut down

the `trqauthd` service:

```
[root]# systemctl stop trqauthd.service
```

3.4.2.C Upgrade Torque Server, MOMs, and Clients

i You *must* complete all the previous upgrade steps in this topic before upgrading Torque Server, MOMs, and Clients. See the list of steps at the beginning of this topic.

1. Upgrade Torque Server. On the Torque Server Host, install the upgrade:

```
[root]# dnf update moab-torque*
```

2. Upgrade Torque MOMs.

i Repeat these instructions for each Torque MOM Host that does *not* reside on the Torque Server Host.

Do the following:

- a. On the Torque Server Host, locate the directory where the RPM distro tarball was unpacked and copy the `moab-torque-common`, `moab-torque-mom`, and `moab-torque-client` RPM files to the Torque MOM Hosts:

```
[root]# scp <dir>/RPMs/moab-torque-common-*.rpm <torque-mom-host>:
[root]# scp <dir>/RPMs/moab-torque-mom-*.rpm <torque-mom-host>:
[root]# scp <dir>/RPMs/moab-torque-client-*.rpm <torque-mom-host>:
```

- b. On *each* Torque MOM Host, use the uploaded RPMs to update the host:

```
[root]# dnf update moab-torque*
```

3. Upgrade Torque Clients.

i Repeat these instructions for any Torque Client Host that does *not* reside on the Torque Server Host *or* the Torque MOM Hosts (such as login nodes or when the Moab Server Host is different from the Torque Server Host).

- a. On the Torque Server Host, locate the directory where the RPM distro tarball was unpacked and copy the `moab-torque-common` and `moab-torque-client` RPM files to the Torque Client Hosts:

```
[root]# scp <dir>/RPMs/moab-torque-common-*.rpm <torque-client-host>:
[root]# scp <dir>/RPMs/moab-torque-client-*.rpm <torque-client-host>:
```

- b. On *each* Torque Client Host, use the uploaded RPMs to update the host:

```
[root]# dnf update moab-torque*
```

3.4.2.D Start Torque Services

1. On the Torque Server Host, start up the Torque server:

```
[root]# systemctl daemon-reload
[root]# systemctl start pbs_server.service
```

2. On *each* Torque MOM Host, start up the Torque MOM service:

```
[root]# systemctl daemon-reload
[root]# systemctl start pbs_mom.service
```

3. On *each* Torque Client Host (including the Moab Server Host, Torque Server Host, and Torque MOM Hosts, if applicable), start up the `trqauthd` service:

```
[root]# systemctl daemon-reload
[root]# systemctl start trqauthd.service
```

3.4.3 Upgrading Moab Workload Manager (RPM)

This topic provides instructions to upgrade Moab Workload Manager to the latest release version using the RPM upgrade method. It includes instructions for migrating your database schema to a new version if necessary.

i Because many system-level files and directories are accessed during the installation, the instructions in this guide should be executed with root privileges.

You will see that the instructions execute commands as the root user. Note that the same commands will work for a non-root user with the `sudo` command.

In this topic:

[3.4.3.A Upgrade Steps](#)

[3.4.3.B Upgrade Moab Server](#)

3.4.3.A Upgrade Steps

1. If you installed Moab Server on its own host *or* if Moab Server is the first component being upgraded on a host with other RPM installations, complete the steps to prepare the host. See [3.2.1 Preparing the Host \(RPM\)](#).
2. If you use ODBC, confirm the database schema compatibility. See 'Migrating Your Database to Newer Versions of Moab' in the *Moab Workload Manager Administrator Guide* for more information.
3. Upgrade Moab Server. See [Upgrade Moab Server](#) below.

3.4.3.B Upgrade Moab Server

i You *must* complete all the previous upgrade steps in this topic before upgrading Moab Server. See the list of steps above.

i The Moab HPC Suite RPM automatically creates a backup of all relevant files. These backups are stored in `/var/tmp/backup-<rpmName>-<timestamp>.tar.gz`.

If changes are detected between any existing configuration files and new configuration files, a version of the new configuration file will be saved under `<configurationFileLocation>/<fileName>.rpmnew`.

On the Moab Server Host, do the following.

1. Stop Moab:

```
[root]# systemctl stop moab.service
```

2. Install the upgrade:

```
[root]# dnf update moab-workload-manager*
```

3. Merge the configuration files.

i Decide whether to start with the old configuration file and add newer configuration options (or vice versa). Typically it depends on the amount of customization you previously made in earlier versions. In instances where you have modified very little, you should consider using the newer configuration and merging site-specific settings from the old file into the new one. Note that new configuration files may have auto-generated content for secret keys and default passwords—be careful to ensure that secret keys shared between components are configured correctly.

- a. Merge the `/opt/moab/etc/moab-private.cfg` file. Make sure that unique items in `/opt/moab/etc/moab-private.cfg.rpmnew` are added to the existing `/opt/moab/etc/moab-private.cfg` file.
- b. Merge customizations from `/opt/moab/etc/moab.cfg` and `/opt/moab/etc/moab.d/*` into `/opt/moab/etc/moab.hpc.cfg`.
 - Although there are several ways to configure and merge changes into the `/opt/moab/etc/moab.cfg` file, the following instructions outline the recommended best practices. *Deviations from these best practices may result in unexpected behavior or added difficulty in future upgrades.*
 - It is best to use the new default configuration file (`/opt/moab/etc/moab.hpc.cfg`) and merge changes from previous files into that one. You will notice that content from the `/opt/moab/etc/moab.d/` directory has been merged into `/opt/moab/etc/moab.hpc.cfg`. Ensure that custom configuration options in all files located in `/opt/moab/etc/moab.d/` directory get merged in to `/opt/moab/etc/moab.hpc.cfg`.
 - You should avoid `#include` configurations.
 - Although the upgrade should have created a backup of the `moab.cfg` file (in `/var/tmp/backup-<rpmName>-<timestamp>.tar.gz`), it is best to create your own backup until you can confirm the updated configuration behaves as expected:

```
[root]# cp /opt/moab/etc/moab.cfg /opt/moab/etc/moab.cfg.bak
```

4. Start Moab:

```
[root]# systemctl daemon-reload
[root]# systemctl start moab.service
```

3.4.4 Disabling the Adaptive Repository after Upgrades (RPM)

After you have completed the upgrade of your Moab HPC Suite components, we recommend that you disable the adaptive repository so that subsequent general system software updates do not inadvertently upgrade your Moab HPC Suite components.

On *each* host where you have enabled the adaptive repository, run the following:

```
[root]# dnf install dnf-utils  
[root]# dnf-config-manager --disable adaptive
```

Chapter 4: Troubleshooting

This chapter details some common problems and general solutions. Additional troubleshooting can be found in the individual component documentation.

Note: If you currently have a support services contract and encounter an installation problem that you can't resolve, please submit an [online support case](#), and a technical support specialist will contact you.

In this chapter:

- [4.1 General Issues](#)
- [4.2 Port Reference](#)
- [4.3 Moab Workload Manager Issues](#)
- [4.4 Moab Web Services Issues](#)
- [4.5 Moab Viewpoint Issues](#)

4.1 General Issues

This section details some common problems and general solutions.

In this section:

- [4.1.1 Where do I set credentials and what are the default values?](#)
- [4.1.2 FastX Error Message: Logins are disabled on this system](#)

4.1.1 Where do I set credentials and what are the default values?

Communication and cooperation between various components of the Moab HPC Suite requires credentials to be properly configured. For ease of use, the credential information, including where credentials are set, default values, and where they are used are grouped by database and product.

In this topic:

4.1.1.A Database Credentials

4.1.1.B Product Credentials

4.1.1.A Database Credentials

MongoDB

Database	User	Default Password	Used By	Parameter
admin	admin_user	secret1	system admins	N/A
moab	moab_user	secret2	/opt/moab/etc/moab-private.cfg	MONGOUSER, MONGOPASSWORD
moab	mws_user	secret3	/opt/mws/etc/mws-config.groovy	grails.mongo.username, grails.mongo.password
moab	insight_user	secret4	/opt/insight/etc/config.groovy	moab.mongo.username, moab.mongo.password
mws	mws_user	secret3	/opt/mws/etc/mws-config.groovy	grails.mongo.username, grails.mongo.password
insight	insight_user	secret4	/opt/insight/etc/config.groovy	mongo.username, mongo.password
insight	mws_user	secret3	https://<mws_server>:8080/mws/admin/plugins/edit/viewpoint-query-helper	user, password
nitro-db	nitro_user	secret5	/opt/nitro-web-services/etc/nitro.cfg	db_username, db_password

Database	User	Default Password	Used By	Parameter
reporting	reporting_user	secret6	/opt/reporting/application.conf	database.username, database.password

i The following characters must be escaped in strings in the `/opt/insight/etc/config.groovy` and `/opt/mws/etc/mws-config.groovy` files (such as when used in a password): `\` (backslash), `"` (double quote), `'` (single quote), `$` (dollar sign). Example: `mongo.password="my\$cool\$password"`. We recommend that you avoid using these characters.

PostgreSQL

Database	User	Default Password	Used By	Parameter
mam	mam	changeme!	/opt/mam/etc/mam-server.cfg	database.user, database.password

Apache Drill

The Drill host should have a user that Reporting Web Services can use to authenticate to Drill.

Host	User	File	Parameter Name	Default Value
Drill host	drilluser	/opt/reporting-web-services/etc/application.properties	reporting.rest.drill.username	changeme!

4.1.1.B Product Credentials

Moab Workload Manager

Declared Parameter		Used By		Default Value
File	Parameter Name	File	Parameter Name	
/opt/moab/etc/moab-private.cfg	MESSAGEQUEUES SECRETKEY	/opt/mws/etc/mws-config.groovy	moab.messageQueue.secretKey	N/A
		/opt/insight/etc/config.groovy	messageQueue.secretKey	
/opt/moab/etc/.moab.key	N/A	/opt/mws/etc/mws-config.groovy	moab.secretKey	N/A

Moab Accounting Manager

Declared Parameter		Used By		Default Value
File	Parameter Name	File	Parameter Name	
/opt/mam/etc/mam-site.conf	token.value	/opt/moab/etc/moab-private.cfg	CLIENTCFG [AM:mam] KEY	N/A

Moab Web Services

Declared Parameter		Used By		Default Value
File	Parameter Name	File	Parameter Name	
/opt/mws/etc/mws-config.groovy	auth.defaultUser.username	https://<viewpoint_server>:8081/configuration/	Username	moab-admin
		/opt/moab/etc/moab-private.cfg	CLIENTCFG [RM:m	

Declared Parameter		Used By		Default Value
File	Parameter Name	File	Parameter Name	
			ws] USERNAME	
/opt/mws/etc/mws-config.groovy	auth.defaultUser.password	https://<viewpoint_server>:8081/configuration/	Password	change!
		/opt/moab/etc/moab-private.cfg	CLIENTCFG [RM:mws] PASSWORD	
/opt/mws/etc/mws-config.groovy	grails.plugin.springsecurity.oauth.Provider.clients[0].clientSecret	https://<viewpoint_server>:8081/configuration/	Client Secret	N/A

i The following characters must be escaped in strings in the `/opt/insight/etc/config.groovy` and `/opt/mws/etc/mws-config.groovy` files (such as when used in a password): `\` (backslash), `"` (double quote), `'` (single quote), `$` (dollar sign). Example: `mongo.password="my\$cool\$password"`. We recommend that you avoid using these characters.

4.1.2 FastX Error Message: Logins are disabled on this system

Once you set up FastX as a gateway, you must quickly configure the session servers. If you don't, the gateway server will be left in a bad state, the gateway's UI will be unavailable, and you will have to manually edit the gateway's configuration files. Do not set up the gateway server, log out, and then try to log back into the gateway server before the session servers are set up.

If you see the error message 'Logins are disabled on this system', do the following.

1. Re-enable logins by editing `/usr/lib/fastx2/var/config/network.json` and change `allowIncomingConnections` from `true` to `false`:

```
"allowIncomingConnections": false
```

2. Restart FastX:

```
systemctl restart fastx
```

3. Now log in again. This time the login will be successful.

Other reasons this error message might appear:

1. The FastX server is not running on the gateway server.
2. The FastX server is not running on the session server.
3. Your browser is not configured to allow pop-ups (when needing to open an exception for trusting the SSL connection).

4.2 Port Reference

The following tables contains the port numbers for the various products in the Moab HPC Suite.

Adaptive Computing Local RPM Repository

Location	Port	Function	When Needed
Deployment Host	80, 443	Adaptive Computing Local RPM repository	The duration of the install when using the RPM method

Torque Resource Manager

Location	Port	Function	When Needed
Torque Server Host	15001	Torque Client and MOM communication to Torque Server	Always
Torque MOM Host (Compute Nodes)	15002	Torque Server communication to Torque MOMs	Always
Torque MOM Host (Compute Nodes)	15003	Torque MOM communication to other Torque MOMs	Always

Moab Workload Manager

Location	Port	Function	When Needed
Moab Server Host	42559	Moab Server Port	If you intend to run client commands on a host different from the Moab Server Host <i>or</i> if you will be using Moab in a grid

Moab Accounting Manager

Location	Port	Function	When Needed
MAM Server Host	7112	MAM Server Port	If you will be installing the MAM Server on a different host from where you installed the Moab Server <i>or</i> you will be installing the MAM Clients on other hosts
MAM Web Server Host	443	HTTPS Port	If using the MAM GUI or MAM Web Services
MAM Database Host	5432	MAM PostgreSQL Server Port	If you will be installing the MAM Database on a different host from the MAM Server

Moab Web Services

Location	Port	Function	When Needed
MWS Server Host	8080	Tomcat Server Port	Always
MWS Database Host	27017	MWS MongoDB Server Port	If you will be installing the MWS Database on a different host from the MWS Server

Moab Insight

Location	Port	Function	When Needed
Insight Server Host	5568	Insight Server Port	Always
Moab MongoDB Database Host	27017	Moab MongoDB Server Port	Always

Location	Port	Function	When Needed
Moab Server Host	5574	Moab Data Port	Always
Moab Server Host	5575	Moab Reliability Port	Always

Reporting

Suggested Host	Service	Port	Function	When Needed
Reporting Master	HDFS name node	8020	HDFS communication	Always
Reporting Master	HDFS name node	50070	HDFS web interface	Always
Reporting Master	Spark Master	6066, 7077	Spark communication	Always
Reporting Master	Spark Master	8082	Spark Master web interface	Always
Reporting Master	Apache Kafka	9092	Kafka communication	Always
Reporting Master	Apache Zookeeper	2181	Zookeeper communication with Kafka and Drill	Always
Insight Server	Apache Drill	8047	Drill HTTP interface	Always
Reporting Worker	HDFS data node	50075, 50010, 50020	HDFS communication	Always
Reporting Worker	Spark Worker	4040	Spark communication	Always
Reporting Worker	Spark worker	8083	Spark worker web interface	Always
MWS Host	Tomcat	8080	Reporting Web Services HTTP interface	Always

Suggested Host	Service	Port	Function	When Needed
MWS Host	MongoDB	27017	MongoDB communication	Always

4.3 Moab Workload Manager Issues

This section details some common problems and general solutions for Moab Workload Manager. See also 'Troubleshooting and System Maintenance' in the *Moab Workload Manager Administrator Guide*.

In this section:

[4.3.1 Moab error: cannot determine local hostname](#)

[4.3.2 Moab error: Moab will now exit due to license file not found](#)

4.3.1 Moab error: cannot determine local hostname

```
# systemctl start moab.service
Starting moab: ERROR: cannot determine local hostname - node is misconfigured
                        [FAILED]
```

```
...
SCHEDCFG [Moab]                SERVER=<moab-hostname>:42559
...
```

Also check `/etc/hosts` to be sure the host name resolves, at least with localhost:

```
...
127.0.0.1 <moab-hostname> localhost localhost.localdomain localhost4
localhost4.localdomain4
...
```

4.3.2 Moab error: Moab will now exit due to license file not found

```
# systemctl start moab.service
Starting moab: Moab will now exit due to license file not found
Please contact Adaptive Computing (sales@adaptivecomputing.com) to get a license for
your system
                        [FAILED]
```

If you encounter this error when starting Moab HPC Suite, make sure your Moab HPC Suite license file is named `moab.lic` and is located in the `/opt/moab/etc/` directory.

Also make sure the license is not expired. The expiration date is listed in the license file, for example:

```
# cat /opt/moab/etc/moab.lic
...
# Expires after Tue Dec 31 10:43:46 2024
...
```

4.4 Moab Web Services Issues

This section details some common problems and general solutions for Moab Web Services (MWS).

If something goes wrong with MWS, look in the following files:

- The MWS log file. By default, this is `/opt/mws/log/mws.log`.
- The Tomcat `catalina.out` file, usually in `/var/log/tomcat` or `$CATALINA_HOME/logs`.

i If you remove the `log4j` configuration from `/opt/mws/etc/mws-config.groovy`, MWS writes its log files to `java.io.tmpdir`. For Tomcat, `java.io.tmpdir` is generally set to `$CATALINA_BASE/temp` or `CATALINA_TMPDIR`.

In this section:

- [4.4.1 MongoDB: Errors during MWS startup](#)
- [4.4.2 MongoDB: Out of semaphores to get db connection](#)
- [4.4.3 MongoDB: Connection wait timeout after 120000 ms](#)
- [4.4.4 java.lang.OutOfMemoryError: Java heap space](#)
- [4.4.5 java.lang.OutOfMemoryError: PermGen space](#)
- [4.4.6 SEVERE: Context \[/mws\] startup failed due to previous errors](#)
- [4.4.7 Moab HPC Suite Reached Maximum Number of Concurrent Connections](#)
- [4.4.8 MongoDB Service Does Not Start](#)

4.4.1 MongoDB: Errors during MWS startup

If the application fails to start and gives error messages such as these:

```
Error creating bean with name 'mongoDatastore'
can't say something; nested exception is com.mongodb.MongoException
```

```
ERROR    grails.app.services.com.ace.mws.ErrorService    0
         Error encountered while attempting to authenticate account or query database; the
         MongoDB server is not available.
         Please verify connection to server '/127.0.0.1:27017' and that MongoDB is running.
```

MongoDB is most likely not running, or the MongoDB host and port are misconfigured.

In this case, there are a few things to verify:

- (*Not relevant if MongoDB is installed on a different host*) Is MongoDB installed?

Run the following commands to assess whether MongoDB is installed on the current host:

```
$ mongo
-bash: mongo: command not found
```

To remedy, install MongoDB, start the `mongod` service and then restart the `tomcat` service. See [Install MongoDB \(Manual Installation\)](#) or [Install and Configure MongoDB \(RPM Installation\)](#) for more information on how to install and configure MongoDB.

- (*Only relevant if MongoDB is installed on a different host*) Is MWS configured to connect to the remote MongoDB host?

Run the following commands to assess whether MongoDB is installed on the current host:

```
[root]# cat /opt/mws/etc/mws-config.groovy | grep 'grails.mongo'
// grails.mongo.username = "mws_user"
// grails.mongo.password = "<ENTER-KEY-HERE>"
// grails.mongo.host = "127.0.0.1"
// grails.mongo.port = 27017
```

Make sure that the `grails.mongo.*` options are configured in `/opt/mws/etc/mws-config.groovy` for the remote MongoDB server and then restart the `tomcat` service:

```
[root]# systemctl restart tomcat.service
```

- Is MWS configured to authenticate with MongoDB, and is MongoDB configured to enforce authentication?

Run the following commands to assess the relevant MWS and MongoDB configurations:

```
[root]# cat /opt/mws/etc/mws-config.groovy | grep 'grails.mongo'
// grails.mongo.username = "mws_user"
// grails.mongo.password = "<ENTER-KEY-HERE>"

[root]# cat /etc/mongod.conf | grep 'auth'
```

```
#noauth = true
auth = true
```

The configuration above is problematic because the `grails.mongo` credentials are commented out in the `/opt/mws/etc/mws-config.groovy` file while MongoDB is configured to enforce authentication ("`auth = true`"). Similar connection issues will exist if the `grails.mongo` parameters do not match the credentials configured for the "mws_user" on both the `mws` and `moab` databases in MongoDB.

(For upgrade scenarios only) If the application fails to start and gives the following message in `/opt/mws/etc/log/mws.log`:

```
java.lang.Exception: The db-migrate.js script has not yet been run. Please see the
upgrade section of the installation guide for instructions.
```

Then the `db-migrate.js` script must be run to update the schema of the `mws` database in MongoDB.

4.4.2 MongoDB: Out of semaphores to get db connection

To resolve this error, adjust the values of `connectionsPerHost` or `threadsAllowedToBlockForConnectionMultiplier` by adding them to `/opt/mws/etc/mws-config.groovy`, for example:

```
grails.mongo.options.connectionsPerHost = 60
grails.mongo.options.threadsAllowedToBlockForConnectionMultiplier = 10
```

For more information on these options, refer to these documents:

- 'Configuring Moab Web Services' in the *Moab Web Services Administrator Guide*, which briefly discusses a few MongoDB driver options.
- The [MongoOptions](#) documentation, which contains full details on all MongoDB driver options.

i You must restart Tomcat after adding, removing, or changing `grails.mongo.options` parameters.

As shipped, `/opt/mws/etc/mws-config.groovy` does not contain any `grails.mongo.options` parameters. To adjust their values, you need to add them to `/opt/mws/etc/mws-config.groovy`.

The default value of `connectionsPerHost` is normally 10, but MWS sets it internally to 50.

The default value of `threadsAllowedToBlockForConnectionMultiplier` is 5.

Any of the options listed in `MongoOptions` can be specified in `/opt/mws/etc/mws-config.groovy`. Just use the prefix `grails.mongo.options` as shown above.

4.4.3 MongoDB: Connection wait timeout after 120000 ms

See [MongoDB: Out of semaphores to get db connection](#) above.

4.4.4 java.lang.OutOfMemoryError: Java heap space

Increase the size of the heap using JVM options `-Xms` and `-Xmx`. Here are the suggested values:

```
CATALINA_OPTS="-DMWS_HOME=/opt/mws -Xms256m -Xmx3g -XX:MaxPermSize=384m"
```

- **-Xms**: Set initial Java heap size.
- **-Xmx**: Set maximum Java heap size.

i Beginning with Java 8, the `MaxPermSize` option is ignored.

4.4.5 java.lang.OutOfMemoryError: PermGen space

(Recommended) Upgrade to Java. Java 8 has completely removed PermGen space and the `MaxPermSize` option is ignored.

For Java version prior to 8, you can increase the size of the permanent generation using JVM option `-XX:MaxPermSize`. Here are the suggested values:

```
CATALINA_OPTS="-DMWS_HOME=/opt/mws -Xms256m -Xmx3g -XX:MaxPermSize=384m"
```

4.4.6 SEVERE: Context [/mws] startup failed due to previous errors

If `catalina.out` contains this error, look in `/opt/mws/log/mws.log` and `/opt/mws/log/stacktrace.log` for more details on the error.

Also ensure that the `/opt/mws/etc/mws-config.groovy` file can be read by the Tomcat user. The permissions should appear as follows:

```
$ ls -al /opt/mws/etc/mws-config.groovy
-r----- 1 tomcat tomcat 4056 Dec  4 12:07 mws-config.groovy
```

4.4.7 Moab HPC Suite Reached Maximum Number of Concurrent Connections

When this error message is encountered, simply add a new line to the `moab.cfg` file:

```
CLIENTMAXCONNECTIONS 256
```

This will change the Moab HPC Suite configuration when Moab HPC Suite is restarted. Run the following command to immediately use the new setting:

```
[root]# changeparam CLIENTMAXCONNECTIONS 256
```

i The number 256 above can be substituted for the desired maximum number of Moab HPC Suite client connections.

4.4.8 MongoDB Service Does Not Start

When installing MWS on a Red Hat-based system, the MongoDB service may fail with the following error message:

```
Starting mongod (via systemctl): Job for mongod.service failed. See 'systemctl status mongod.service' and 'journalctl -xn' for details.[FAILED]
```

You may be able to work around the issue by substituting the `/etc/init.d/mongod` script with a `systemd` script. To do this first make sure the MongoDB service isn't running, then move the `mongod` startup script out of the `/etc/init.d` directory:

```
[root]# systemctl stop mongod.service
[root]# mv /etc/init.d/mongod ~
```

Create the `mongo` `systemd` unit file:

```
[root]# touch /usr/lib/systemd/system/mongod.service
[root]# chmod 664 /usr/lib/systemd/system/mongod.service
```

```
[root]# vi /usr/lib/systemd/system/mongod.service
```

The contents of the mongod unit file should be as follows:

```
[Unit]
Description=MongoDB Database Service
Wants=network.target
After=network.target

[Service]
Type=forking
PIDFile=/var/run/mongodb/mongod.pid
ExecStart=/usr/bin/mongod -f /etc/mongod.conf
ExecReload=/bin/kill -HUP $MAINPID
Restart=always
User=mongod
Group=mongod
StandardOutput=syslog
StandardError=syslog
TimeoutSec=60

[Install]
WantedBy=multi-user.target
```

After editing the file, start the MongoDB service:

```
[root]# systemctl daemon-reload
[root]# systemctl start mongod.service
```

Verify that the MongoDB service is running:

```
[root]# systemctl status mongod.service
```

4.5 Moab Viewpoint Issues

This section details some common problems and general solutions for Moab Viewpoint.

In this section:

- [4.5.1 General Configuration Issues](#)
- [4.5.2 Only the Configuration Page is Displayed in Viewpoint](#)
- [4.5.3 Viewpoint Does Not Report Any of My Jobs or Nodes](#)
- [4.5.4 viewpoint-query-helper Plugin Does Not Connect](#)
- [4.5.5 Job's Processor Count Changes After Submission](#)

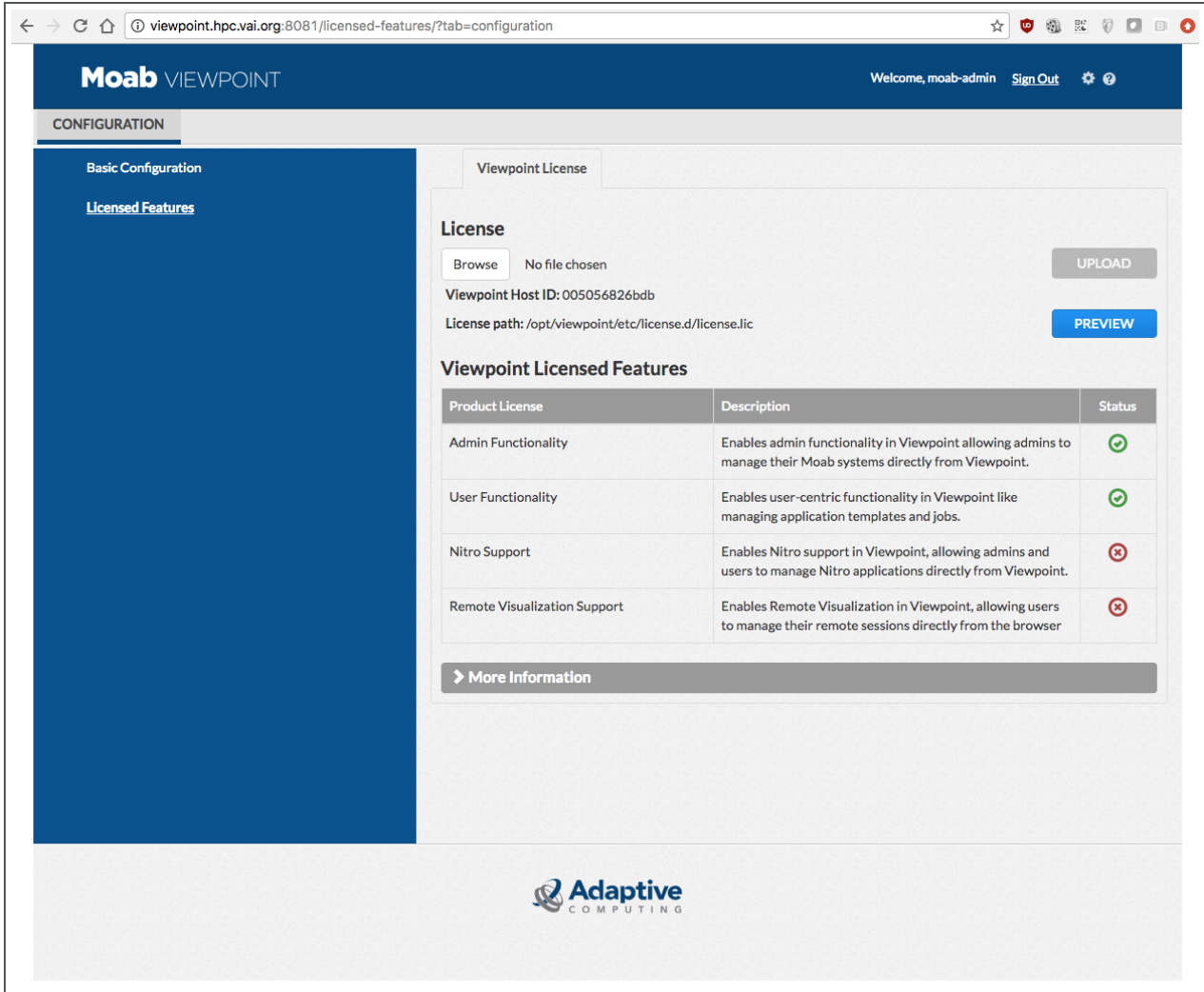
4.5.1 General Configuration Issues

The Viewpoint configuration file (`/opt/viewpoint/etc/viewpoint.cfg`) may include several environment variables used to configure a Viewpoint installation and troubleshoot Viewpoint operation issues. Viewpoint environment variables are described in the table below:

Environment Variable	Description
VIEWPOINT_CONFIG_PATH	Full path to config.json file.
VIEWPOINT_DATABASE_HOST	Database host.
VIEWPOINT_DATABASE_NAME	Database name.
VIEWPOINT_DATABASE_PASSWORD	Database user password.
VIEWPOINT_DATABASE_PORT	Database port.
VIEWPOINT_DATABASE_USER	Database user.
VIEWPOINT_DEBUG	Debug mode. Production = 0, debug = 1.
VIEWPOINT_LOG	Log file location.
VIEWPOINT_LOG_LEVEL	Log level (INFO, WARNING, ERROR, CRITICAL, or DEBUG).
VIEWPOINT_PERMISSIONS_PATH	Full path to permissions.json file.
VIEWPOINT_PREFIX	URL prefix (defaults to /).
VIEWPOINT_STATIC_ROOT	Location of compiled static files.
VIEWPOINT_STATIC_URL	URL prefix for static resources (defaults to /static/).
VIEWPOINT_TEST	TEST mode (used for UI tests only). Production = 0, test = 1.
VIEWPOINT_SESSION_AGE	Lifetime of the user session in seconds (defaults to 2 weeks).

4.5.2 Only the Configuration Page is Displayed in Viewpoint

A particular configuration problem causes Viewpoint to display only the Configuration Page with only the Viewpoint License tab (not the Moab License tab). The Viewpoint License tab includes links only to the Basic Configuration and Licensed Features pages as shown below:



The screenshot shows the Moab Viewpoint Configuration page. The browser address bar indicates the URL: `viewpoint.hpc.vai.org:8081/licensed-features/?tab=configuration`. The page header includes the Moab Viewpoint logo and a user greeting: "Welcome, moab-admin Sign Out".

The main content area is divided into two sections:

- Left Sidebar:** Contains two links: "Basic Configuration" and "Licensed Features".
- Right Panel:** Titled "Viewpoint License", it contains:
 - A "License" section with a "Browse" button, the text "No file chosen", and an "UPLOAD" button.
 - Fields for "Viewpoint Host ID: 005056826bdb" and "License path: /opt/viewpoint/etc/license.d/license.lic", with a "PREVIEW" button.
 - A "Viewpoint Licensed Features" table with the following data:

Product License	Description	Status
Admin Functionality	Enables admin functionality in Viewpoint allowing admins to manage their Moab systems directly from Viewpoint.	✓
User Functionality	Enables user-centric functionality in Viewpoint like managing application templates and jobs.	✓
Nitro Support	Enables Nitro support in Viewpoint, allowing admins and users to manage Nitro applications directly from Viewpoint.	✗
Remote Visualization Support	Enables Remote Visualization in Viewpoint, allowing users to manage their remote sessions directly from the browser	✗

At the bottom of the right panel, there is a "More Information" button with a right-pointing arrow.

The footer of the page features the Adaptive Computing logo.

This problem occurs when the Viewpoint admin user is the same as the `auth.defaultUser.username` in MWS.

To resolve this issue, do the following.

1. Change the admin user in `/opt/viewpoint/etc/viewpoint.cfg`.

For example, if the admin username was set to `moab-admin`, which is also the name of the `auth.defaultUser.username` in MWS, change the admin username in `/opt/viewpoint/etc/viewpoint.cfg` (`viewpoint-admin` in the example shown below):

```
[admin]
username = viewpoint-admin
password = pbkdf2_
sha256$20000$ZHeToCJgrSUH$+xmzYdhpqZCJokx09eGzyr2B6jrfCgLLBT+pBgMis4w=
```

2. Identify the entry for the previous admin user from the PostgreSQL database by executing the following commands as root:

```
[root]# su - postgres
[postgres]$ psql
\c moab_viewpoint
select * from auth2_user;
```

The `auth2_user` table will display, similar to the following:

```
 id | is_active | is_staff | is_superuser | last_login |
-----+-----+-----+-----+-----+
  2 | t         | t       | f           | 2024-12-19 11:49:27.765855-05 |
viewpoint-admin
  1 | t         | t       | f           | 2024-12-19 12:06:24.642922-05 | moab-
admin
(2 rows)
```

3. Delete the previous admin username from the table by executing the following command (substituting the previous admin username):

```
delete from auth2_user where username = 'moab-admin';
```

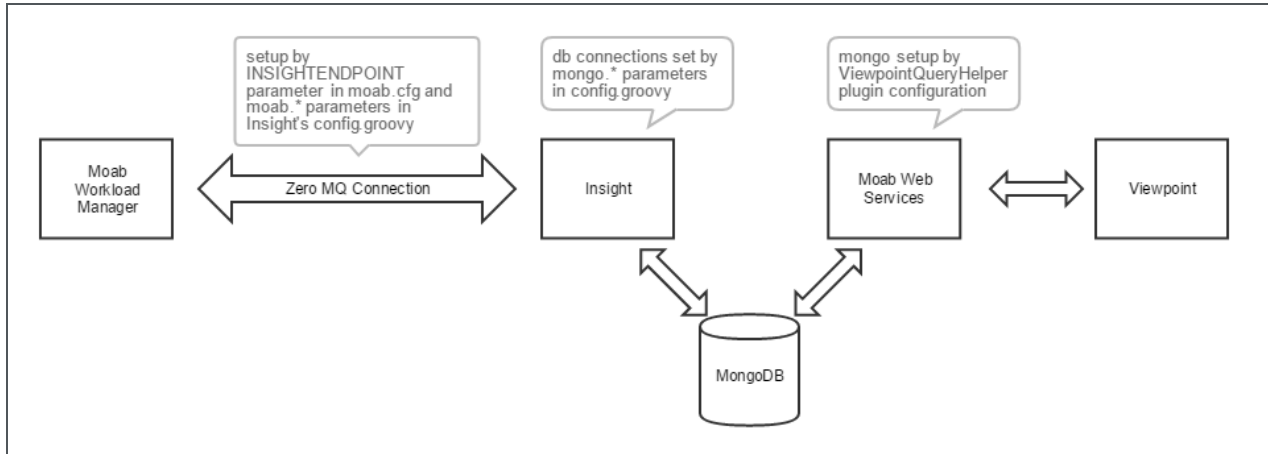
4.5.3 Viewpoint Does Not Report Any of My Jobs or Nodes

There are multiple reasons why jobs and nodes might not show up in Viewpoint. Verify the following.

1. **Moab HPC Suite Setup**

Essentially, there are many communication points in our stack from the point that jobs get submitted to the point they get displayed in Viewpoint.

Look at the following diagram describing our data flow architecture:



The Moab Workload Manager will push data into Insight using a ZeroMQ message queue pipe.

Then, Insight will parse that data and insert it into a MongoDB database.

When Viewpoint needs to query information on jobs and nodes, it will communicate with Moab Web Services, which in turn will consume the data directly from the MongoDB database where Insight recorded Moab's events.

Failure to configure the communication channels between all these components will result in Viewpoint not being able to display job or node information.

2. Hardware Specifications

Another reason why Viewpoint might not be able to show job and node information is that you installed all Moab HPC components in a single machine that is too overloaded.

See [1.2 Server Hardware Requirements](#) for more information.

3. RPM Versions

One other common problem customers can experience is that they install incompatible versions of our software components.

Make sure you are using the same major/minor version across all components (e.g., Moab Workload Manager 9.1, Moab Web Services 9.1, Insight 9.1, etc.).

4.5.4 viewpoint-query-helper Plugin Does Not Connect

viewpoint-query-helper Plugin Does Not Connect to the Insight MongoDB Database

If the user name or the password for the Insight MongoDB database was entered incorrectly, the viewpoint-query-helper plugin will not be able to connect to the database. An error message is reported to the MWS Plugin Monitoring page, for example:

Moab® Web Services

🏠 **Plugins** Admin

Plugin Monitoring

This page monitors the status of all plugins in Moab Web Services.

ⓘ Invalid configuration for plugin viewpoint-query-helper
Incorrect user name (mws_user) or password for the insight MongoDB database on host localhost

Thursday, August 18, 2022
09:51:51 AM

Reload when poll occurs

Active Plugins

ID	Plugin Type	Last Poll	Next Poll	Actions
fastx	RLM	00:00:26	00:00:03	⏹️ ⏸️ ▶️

Disabled Plugins

ID	Plugin Type	State	Actions
viewpoint-query-helper	ViewpointQueryHelper	Errored	▶️ ⏹️

To resolve this issue, do the following.

1. If you have not already done so:
 - a. Log in as an administrator to your MWS instance.
 - b. Select `Plugins`, and then select `Plugin Monitoring`. You should see a page similar to the example image displayed earlier in this section.
2. In the `Disabled Plugins` section, click on the link for the `viewpoint-query-helper` plugin.
3. When the `Show Plugin` page displays, click `Edit`.
4. Enter the correct connection information, and then click `Update` to save your changes (you are returned to the `Show Plugin` page).
5. Return to the `Plugin Monitoring` page and start the plugin using the green start button.

Alternatively, you can change the password of the `mws_user` in the insight database from the database host.

From the host on which the insight MongoDB database resides, run the following commands (substituting your password information):

```
$ mongo
> use insight;
> db.changeUserPassword("mws_user", "secret3");
> exit;
```

4.5.5 Job's Processor Count Changes After Submission

When migrating jobs to Torque from Viewpoint, Moab HPC Suite will translate the request into the equivalent *qsub* command with the proper *-l procs* syntax. In some situations, Torque's queues may have been configured with a *default_resources.nodes* setting that is incompatible with the job's *-l procs* request. In this situation, the *default_resources.nodes* setting should be removed from the queue or the job should be submitted to a queue that does not have a *default_resources.nodes* setting.