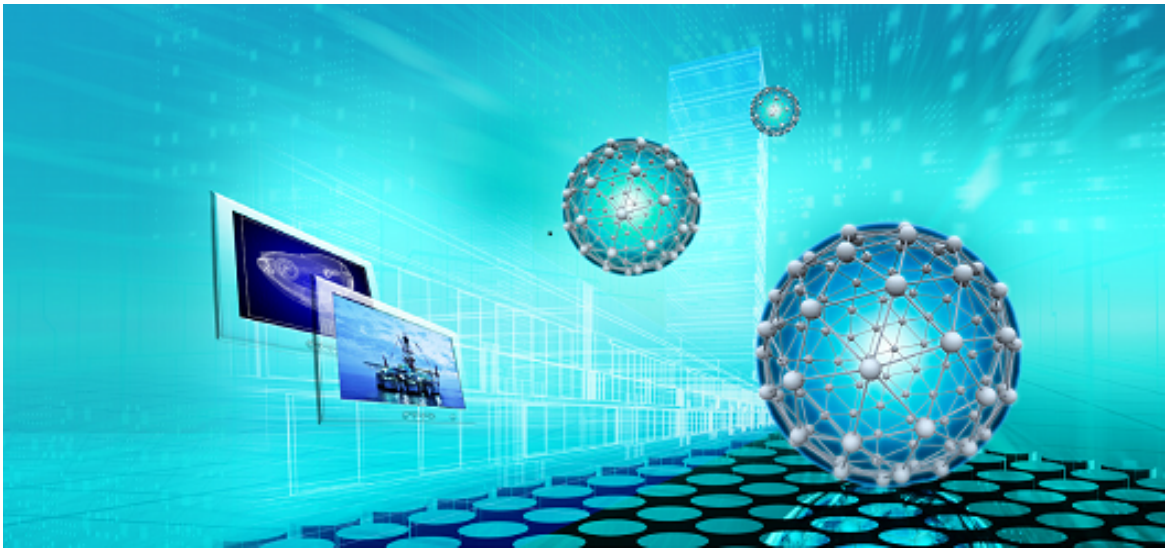
Moab HPC Suite

Installation and Configuration Guide 10.1.0 for Ubuntu-Based Systems (18.04, 20.04, 22.04)

March 2025



Adaptive COMPUTING

Legal Notices

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

Welcome to the *Moab HPC Suite Installation and Configuration Guide 10.1.0* for Ubuntu-Based Systems (18.04, 20.04, 22.04)

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 10.1.0 Moab HPC Suite contains the following components:

- Torque Resource Manager 7.1.0
- Moab Workload Manager 10.1.0
- Moab Accounting Manager 10.1.0
- Moab Web Services 10.1.0

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 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

In this section:

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.

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 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 for the various components.

   The Manual 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 as applicable.
5. Refer to Chapter 3: Troubleshooting for assistance in addressing common problems during installation and configuration.

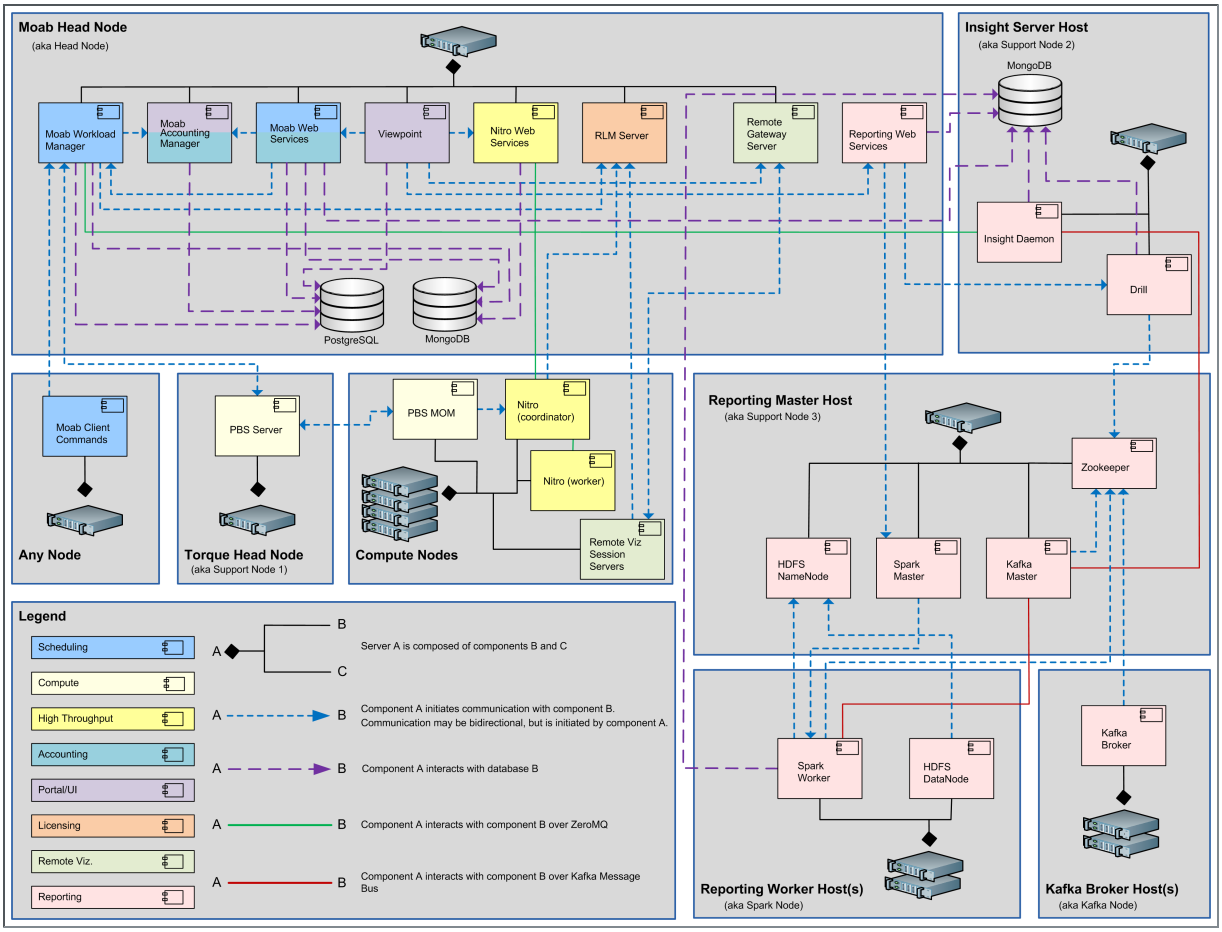
1.2 Server Hardware Requirements

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

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 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-server) 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, and accounts, 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, an 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

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.

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** • 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)
			disk space **Insight Server Host** • 8 Intel/AMD x86-64 cores • At least 16 GB RAM • At least 512 GB dedicated disk space	
Medium	500	<100k	**Moab Server+Torque Server Host** • 8 Intel/AMD x86-64 cores • At least 16 GB RAM • At least 512 GB dedicated disk space **Insight Server Host** • 8 Intel/AMD x86-64 cores • At least 16 GB of RAM • At least 1024 GB disk	**Moab Server+Torque Server Host** • 16 Intel/AMD x86-64 cores • At least 32 GB RAM • At least 1 TB dedicated disk space **Insight Server Host** • 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	**Moab Server Host** • 8 Intel/AMD x86-64 cores • At least 16 GB RAM • At least 512 GB dedicated disk space **Torque Server Host** • 8 Intel/AMD x86-64 cores • At least 16 GB RAM • At least 512 GB dedicated disk space **Insight Server Host** • 8 Intel/AMD x86-64 cores • At least 16 GB of RAM • At least 2048 GB disk	The Moab Server should *not* reside on the same host as the Torque Server. MWS Server *must* reside on the same host as the Moab Server (Moab Server Host). The MAM Server can reside on its own host, on the Moab Host (preferred), or another server's host (except for the Insight Host). Databases can reside on the same or a different host from its server component.

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	**Reporting Master Host** • 4 Intel/AMD x86-64 cores • At least 8 GB RAM • At least 512 GB dedicated disk space **Reporting Worker Host** • 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)
	Kafka Broker Host • 4 Intel/AMD x86-64 cores • At least 6 GB RAM • At least 512 GB dedicated disk space	
Medium	**Reporting Master Host** • 4 Intel/AMD x86-64 cores • At least 8 GB RAM • At least 1024 GB dedicated disk space **Reporting Worker Host** • 8 Intel/AMD x86-64 cores • At least 16 GB RAM • At least 512 GB dedicated disk space **Kafka Broker Host** • 4 Intel/AMD x86-64 cores • At least 6 GB RAM • At least 1024 GB dedicated disk space	**Reporting Master Host** • 4 Intel/AMD x86-64 cores • At least 16 GB RAM • At least 1024 GB dedicated disk space **Reporting Worker Host** • 8 Intel/AMD x86-64 cores • At least 32 GB RAM • At least 512 GB dedicated disk space **Kafka Broker Host** • 4 Intel/AMD x86-64 cores • At least 6 GB RAM • At least 1024 GB dedicated disk space
Medium with High Throughput or Larger	**Reporting Master Host** • 4 Intel/AMD x86-64 cores • At least 16 GB RAM • At least 2048 GB dedicated disk space **Reporting Worker Host**	More than one Reporting Worker host is recommended.

Environment Type	Minimum Requirements (per Host Distribution)	Recommended Requirements (targeting minimum number of hosts)
	• 8 Intel/AMD x86-64 cores • At least 32 GB RAM • At least 512 GB dedicated disk space **Kafka Broker Host** • 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

In this section:

1.3.1 Torque

If you intend to use Torque 7.1.0 with Moab Workload Manager, you must run Moab version 8.0 or later. However, some Torque functionality may not be available. See the *Moab HPC Suite Release Notes* for more information.

Supported Operating Systems

- Red Hat 7, 8
- SUSE 12, 15
- Ubuntu 18.04, 20.04, 22.04

Software Requirements

- libxml2-devel package (package name may vary)
- openssl-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
- cgroupv1:
  - cgroupv1 is recommended for the tarball install
  - cgroupv2 is not yet supported by Torque
- If you build Torque from source, the following additional software is required:
  - gcc
  - gcc-c++
  - posix-compatible version of make
  - libtool 1.5.22 or later
  - boost-devel 1.36.0 or later

1.3.2 Moab Workload Manager

Supported Operating Systems

- Red Hat 7, 8
- SUSE 12, 15
- Ubuntu 18.04, 20.04, 22.04

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

1.3.3 Moab Accounting Manager

Moab Accounting Manager (MAM) is commonly installed on the same host as Moab Workload Manager; however, in some cases you might obtain better performance by installing them on different hosts.

Supported Operating Systems

- Red Hat 7, 8
- SUSE 12, 15
- Ubuntu 18.04, 20.04, 22.04

Software Requirements

- gcc
- perl-suidperl
- httpd
- mod_ssl
- rrdtool
- Moab Workload Manager 10.1.0
- Perl modules; see 2.1.4 Installing Moab Accounting Manager (Manual Installation) 1.1 Installing Moab Accounting Manager for more details

Depends On (not necessarily on the same host)

MAM uses an RDBMS as a back end. We recommend that the database used by MAM does *not* reside on the same host as the database used by Insight.

- PostgreSQL 7.2 or later

1.3.4 Moab Web Services

MWS Server *must* reside on the same host as Moab HPC Suite Server (Moab Server Host).

Supported Operating Systems

- Red Hat 7, 8
- SUSE 12, 15
- Ubuntu 18.04, 20.04, 22.04

Software Requirements

- Moab Workload Manager 10.1.0
- Oracle® Java® 8 Runtime Environment

  Oracle Java 8 Runtime Environment is the recommended Java environment, but Oracle Java 7 is also supported. All other versions of Java, including Java 9, OpenJDK/IcedTea, GNU Compiler for Java, and so on, cannot run Moab Web Services.

- Apache Tomcat™ 7, 8

Depends On (not necessarily on the same host)

- LDAP *or* PAM; see 2.1.5 Installing Moab Web Services (Manual Installation) 1.1 Installing Moab Web Services for more details
- MongoDB® 4.2.x

Chapter 2: Manual Installation

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

Be aware of the following:

- Manual Installation is not available for Insight.
- 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.

In this chapter:

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

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

> 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.

> 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

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 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]# apt upgrade
```

2.1.1.C 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.D 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. Moab Workload Manager. See 2.1.3 Installing Moab Workload Manager.
3. Moab Accounting Manager. See 2.1.4 Installing Moab Accounting Manager.
4. Moab Web Services. See 2.1.5 Installing Moab Web Services.

2.1.2 Installing Torque Resource Manager

If you intend to use Torque Resource Manager 7.1.0 with Moab Workload Manager, you must run Moab HPC Suite version 8.0 or later. However, some Torque functionality may not be available. See the *Moab HPC Suite Release Notes* for more information.

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

In this topic:

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 Packages

On the Torque Server Host, run the following command to install the `libxml2-devel`, `openssl-devel`, and `boost-devel` packages:

```
[root]# apt install build-essential libxml2-dev libssl-dev libboost-dev libtool pkg-
config zlib1g-dev
```

Install hwloc

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

```
[root]# apt install libhwloc-dev
```

2.1.2.C Install Torque Server

> 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 -xzvf torque-7.1.0.tar.gz
   [root]# cd torque-7.1.0/
   ```

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

   > These instructions assume you are using cgroups. When cgroups are supported, cpusets are handled by the cgroup cpuset subsystem. If you are not using cgroups, use `--enable-cpusets` instead.

   > If --enable-gui is part of your configuration, run the following commands:
   >
   > ```
   > $ cd /usr/lib64
   > $ ln -s libXext.so.6.4.0 libXext.so
   > $ ln -s libXss.so.1 libXss.so
   > ```
   >
   > When finished, cd back to your install directory.

   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.

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:

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

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

   a. If installing Torque with cgroups support enabled (recommended), install cgroup-tools:

```
[root]# apt install cgroup-tools
```

> Torque currently only supports cgroupv1. If you are using a version of Ubuntu that uses cgroupv2 by default (e.g., 22.04), switch to use cgroupv1:
>
> ```
> [root]# mount | grep cgroup # If you see cgroup2 you will need to switch
> to cgroupv1
> [root]# vim /etc/default/grub
> GRUB_CMDLINE_LINUX_DEFAULT="systemd.unified_cgroup_hierarchy=0"
> [root]# update-grub
> [root]# reboot
> ```

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.

   > 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>:/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

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 *or* 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]# apt install build-essential libxml2-dev pkg-config
```

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 Install Moab Server

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-10.1.0-<OS>.tar.gz
[root]# cd moab-10.1.0-<OS>
```

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>
```

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

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 '10.1.0' License Information:
 Current License:  Max Procs   = 10000
 Current License:  Valid Until - Jul 13 19:42:10 2025
```

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.D 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.E 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.F (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.

If your site needs secure communication and authentication between 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

2.1.4 Installing Moab Accounting Manager

This topic contains instructions on how to install and start Moab Accounting Manager (MAM).

In this topic:

2.1.4.A Plan Your Installation

The first step is determining the number of different hosts (physical machines) required for your MAM installation.

Your MAM installation includes:

- MAM Server
- MAM Database

- MAM Clients (possibly several hosts)
- MAM Web Server (optional, for the MAM GUI and/or MAM Web Services)

Each of these components can be installed on their own hosts (meaning the actual physical machine) or can be combined on the same hosts. For example, the MAM Database can be installed on the same *host* as the MAM Server. Or the MAM Server can be installed on the same host on which you installed the Moab Server.

Once you have determined which components are installed on which hosts, complete the rest of the instructions for the MAM installation.

The instructions that follow in this topic will use the term *host* after each component to indicate the physical machine on which the component is installed (for example, MAM Server Host and MAM Database Host). Depending on your configuration, the *host* may refer to the component installed on its own machine or installed on the same machine as another component.

2.1.4.B 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
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 *or* 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

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

2.1.4.C Install and Initialize PostgreSQL Server

MAM uses a database for transactions and data persistence. The PostgreSQL database can be installed on a different host from the MAM Server; however, it is often convenient to install them on the same host. For example, the PostgreSQL database can be installed on:

- The same host as the MAM Server.
- A separate PostgreSQL database host.
- A separate *shared* PostgreSQL database host.

On the host where the MAM PostgreSQL database will reside, do the following.

1. Install and initialize the PostgreSQL Server:

```
[root]# apt install postgresql
```

2. If the MAM Database Host is installed on a *different* host from where you will install the MAM Server, configure PostgreSQL to accept connections from the MAM Server Host:

```
[root]# vi /etc/postgresql/*/main/postgresql.conf

# Replace <mam-server-host> with the TCP/IP address on which the database server is
to listen for connections
# from the MAM server. This will normally be the hostname or ip address of the MAM
Database Host.
listen_addresses = '<mam-database-host>'
```

3. If your PostgreSQL database version is prior to version 9.1, configure postgresql to avoid interpreting backslashes as escape characters:

```
[root]# vi /var/lib/pgsql/data/postgresql.conf

standard_conforming_strings = on
```

4. Start or restart the database:

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

2.1.4.D Install Dependencies, Packages, or Clients

Use the following instructions to install the required MAM dependencies, packages, or clients.

Depending on your configuration, the MAM Server Host and the MAM Web Server Host may be installed on the same host. The MAM Client Host is automatically installed on the same host as the MAM Server Host; however, you can also install the MAM Client Host on any other hosts on which you want to have the MAM client commands available to users or admins.

If any of the Perl module packages fail to install or are unavailable for your system, you can install it from CPAN by running `cpan MODULENAME` where `MODULENAME` is the respective perl module name.

1. On the MAM Server Host, the MAM Web Server Host, and the MAM Client Hosts, run the following commands:

```
[root]# apt install gcc libauthen-pam-perl libconfig-tiny-perl libcrypt-cbc-perl
libcrypt-des-perl libcrypt-des-ede3-perl libdigest-hmac-perl libdigest-sha-perl
liberror-perl libjson-perl liblog-dispatch-filerotate-perl liblog-log4perl-perl
libparams-validate-perl libxml-libxml-perl make
```

2. On the MAM Server Host, run the following:

```
[root]# apt install postgresql-common postgresql-client libdate-manip-perl libdbd-
pg-perl libdbi-perl
```

3. If you plan to use the MAM GUI, on the MAM Web Server Host, run the following:

```
[root]# apt install apache2 libcgi-session-perl
```

4. If you plan to use MAM Web Services, on the MAM Web Server Host, run the following:

```
[root]# apt install apache2 libapache2-mod-perl2
```

5. On each of the MAM Client Hosts (including the MAM Server Host), run the following:

```
[root]# apt install libssl-dev libterm-readline-gnu-perl libterm-readkey-perl
```

2.1.4.E Install MAM Server

If you will be installing the MAM GUI or MAM Web Services and you intend to use PAM authentication (see 'Integrating with PAM' in the *Moab Accounting Manager Administrator Guide*), the MAM server will need to be installed and run using the root user. If this is not required in your setup, we recommend installing and running the MAM server using a non-root user. These instructions will show examples for installing MAM using a non-root user, mam. If using root, or a different user, you will need to make appropriate substitutions for your selected user in the remaining steps and sections below.

On the MAM Server Host, do the following.

1. Create a user called `mam` and switch to that user:

```
[root]# useradd -m mam -s $(which bash)
[root]# su - mam
[mam]$ mkdir src
[mam]$ cd src
```

2. Download the latest MAM tarball from the Adaptive Computing website.

3. As the mam user, run the following commands:

```
[mam]$ tar -zxvf mam-10.1.0.tar.gz
[mam]$ cd mam-10.1.0
```

4. Configure the software. For a list of all the configuration options, see 2.2.4 Moab Accounting Manager Configuration Options.

```
[mam]$ ./configure
```

> If you are planning to use the GUI or web services and you want to use PAM for UNIX password authentication, use the --with-pam option. This will configure MAM to run as root and configure the GUI and web services to use PAM for user password authentication.

5. Compile the software:

```
[mam]$ make
```

> Depending on your configuration, you may need to replace 'make' with a `make` command that includes additional functionality. Specifically:
>
> - If you only need to install the clients on a particular system, use `make clients-only`.
> - If you only need to install the web GUI on a particular system, use `make gui-only`.
> - If you only need to install the web services on a particular system, use `make ws-only`.

6. Install the software:

```
[mam]$ exit
[root]# cd ~mam/src/mam-10.1.0
[root]# make install
```

> Depending on your configuration, you may need to replace 'make install' with a `make` command that includes additional functionality. Specifically:
>
> - If you only need to install the clients on a particular system, use `make install-clients-only`.
> - If you only need to install the web GUI on a particular system, use `make install-gui-only`.
> - If you only need to install the web services on a particular system, use `make install-ws-only`.

7. As the database user, create a mam user with the ability to create a database.

> PostgreSQL should have previously been installed using the instructions in 2.1.1 Preparing for Manual Installation.

```
[root]# su - postgres
[postgres]$ psql
CREATE USER mam WITH CREATEDB PASSWORD 'changeme!';
\q
[postgres]$ exit
```

The password you define must be synchronized with the `database.password` value in `/opt/mam/etc/mam-server.conf`:

```
[root]# vi /opt/mam/etc/mam-server.conf
database.password = changeme!
```

For systems with a separate PostgreSQL host, add `database.datasource` to `/opt/mam/etc/mam-server.conf`:

```
database.datasource=DBI:Pg:dbname=mam;host=remote-host
```

8. Create the mam database. Run the `hpc.sql` script to populate the database with objects, actions, and attributes necessary to function as an Accounting Manager:

```
[root]# su - mam
[mam]$ cd src/mam-10.1.0
[mam]$ createdb mam
[mam]$ psql mam < hpc.sql
[mam]$ exit
```

9. Configure MAM to automatically start up at system boot; start the `mam` service:

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

2.1.4.F Configure the MAM GUI

If you plan to use the web GUI, then on the MAM Web Server Host, do the following.

1. As `root`, add or edit the SSL virtual host definition as appropriate for your environment. To do so, configure the `cgi-bin` directory in `ssl.conf`. Below the `cgi-bin` directory element, create an alias for `/cgi-bin` pointing to your `cgi-bin` directory. If you chose to install to a `cgi-bin` sub-directory, you might want to create an alias for that as well. Also, add `index.cgi` to the `DirectoryIndex` so you can use the shorter sub-directory name.

```
[root]# a2enmod cgi
[root]# a2enmod ssl
[root]# a2ensite default-ssl

[root]# vi /etc/apache2/sites-enabled/default-ssl.conf

# Configure your cgi-bin directory
<Directory "/usr/lib/cgi-bin">
     Options ExecCGI
     AddHandler cgi-script .cgi
     AllowOverride All
     Require all granted
</Directory>

# Aliases for /cgi-bin
Alias /cgi-bin/ /usr/lib/cgi-bin/
Alias /mam /usr/lib/cgi-bin/mam/

# Make shorter sub-dir name available
DirectoryIndex index.cgi
```

2. For the highest security, we recommend that you install a public key certificate that has been signed by a certificate authority. The exact steps to do this are specific to your distribution and the chosen certificate authority. An overview of this process for CentOS is documented at Red Hat Products and Documentation.

   Alternatively, if your network domain can be secured from man-in-the-middle attacks, you could use a self-signed certificate. Often this does not require any additional steps because in many distributions, the Apache SSL configuration provides self-signed certificates by default.

   If your configuration uses self-signed certificates, no action is required; Ubuntu ships with ready-made certificates.

3. Configure the GUI to use the desired authentication method. Edit the GUI configuration file and ensure that the authentication.method parameter is set to the desired value. This parameter can be set to a value of mam-password to use the built-in MAM database Password table, or to a value of pam to authenticate the user using Linux PAM. If you want to use PAM for UNIX password authentication, you should have run configure with the --with-pam option (or --with-user=root) in order to configure MAM to run as root. See 'Integrating with PAM' in the *Moab Accounting Manager Administrator Guide* for more information about the steps required to configure PAM authentication.

```
[mam]$ vi /opt/mam/etc/mam-gui.conf
```

```
authentication.method = mam-password
```

4. Start or restart the HTTP server daemon:

```
[root]# systemctl restart apache2.service
```

2.1.4.G Configure MAM Web Services

If you plan to use MAM Web Services, then on the MAM Web Server Host, do the following.

1. Edit the SSL virtual host definition in ssl.conf to include the mamws location, for example:

```
[root]# a2enmod ssl
[root]# a2ensite default-ssl

[root]# vi /etc/apache2/sites-enabled/default-ssl.conf

# Place the following within the 443 VirtualHost definition
PerlOptions +Parent
PerlSwitches -Mlib=/opt/mam/lib
PerlModule MAM::WSResponseHandler
PerlModule MAM::WSAuthenHandler
<Location /mamws>
     SetHandler perl-script
     PerlResponseHandler MAM::WSResponseHandler
     Options +ExecCGI

     AuthName MAM
     PerlAuthenHandler MAM::WSAuthenHandler
     Require valid-user

     Order allow,deny
     Allow from all
</Location>
```

2. For the highest security, we recommend that you install a public key certificate that has been signed by a certificate authority. The exact steps to do this are specific to your distribution and the chosen certificate authority. An overview of this process for CentOS is documented at Red Hat Products and Documentation.

   Alternatively, if your network domain can be secured from man-in-the-middle attacks, you could use a self-signed certificate. Often this does not require any additional steps because in many distributions, the Apache SSL configuration provides self-signed certificates by default.

   If your configuration uses self-signed certificates, no action is required; Ubuntu ships with ready-made certificates.

3. Configure MAM Web Services to use the desired authentication method. Edit the web services configuration file and ensure that the `authentication.method` parameter is set to the desired value. This parameter can be set to a value of `mam-`

password to use the built-in MAM database Password table, or to a value of pam to authenticate the user using Linux PAM. If you want to use PAM for UNIX password authentication, you should have run configure with the --with-pam option (or --with-user=root) in order to configure MAM to run as root. See 'Integrating with PAM' in the *Moab Accounting Manager Administrator Guide* for more information about the steps required to configure PAM authentication.

```
[mam]$ vi /opt/mam/etc/mam-ws.conf
authentication.method = mam-password
```

4. Start or restart the HTTP server daemon:

```
[root]# systemctl restart apache2.service
```

2.1.4.H Access the MAM GUI

If you plan to use the web GUI, then on the MAM Server Host, do the following.

1. If your GUI authentication method is mam-password, create a password for the mam user that you want to access the MAM GUI:

```
[root]# su - mam
[mam]$ mam-set-password
[mam]$ exit
```

2. Verify the connection:

   a. Open a browser and navigate to https://<mam_web_server_host>/cgi-bin/mam.

   b. Log in as the mam user with the password you set in step 1.

2.1.4.I Access MAM Web Services

If you plan to use MAM web services, then on a MAM Client Host, do the following.

1. If your web services authentication method is mam-password, create a password for the mam user that you want to access the MAM Web Services:

```
[root]# su - mam
[mam]$ mam-set-password
[mam]$ exit
```

2. Make a call to web services:

```
[root]# curl -k -X GET --basic -u mam:changeme! 'https://<mam_web_server_
host>/mamws/system'
```

   Alternatively, for queries, you can use the browser to access the URL, for example: 'https://<mam_web_server_host>/mamws/system'.

2.1.4.J Configure Moab Workload Manager to Use MAM

If integrating with Moab Workload Manager, do the following.

1. Configure Moab to talk to MAM. Do *one* of the following:

   - **MAM Option.** If you will be using the MAM (direct network) accounting manager interface with Moab Workload Manager (this is the default), do the following:

     a. On the Moab Server Host, edit the Moab configuration file, uncomment the AMCFG lines and set the TYPE to MAM and set the HOST. If the Moab Server and the MAM Server are on the same host, set HOST to 'localhost'; otherwise, set HOST to the host name for the MAM Server (MAM Server Host).

     ```
     [root]# vi /opt/moab/etc/moab.cfg
     AMCFG[mam] TYPE=MAM HOST=<mam_server_host>
     ```

     Customize additionally as needed. See 'Accounting, Charging, and Allocation Management' in the *Moab Workload Manager Administrator Guide*.

     b. Configure Moab to authenticate with MAM using the MAM secret key:

        i. On the MAM Server Host, copy the auto-generated secret key from the token.value value in the `/opt/mam/etc/mam-site.conf` file.

        ii. On the Moab Server Host, add the secret key to the `moab-private.cfg` file as the value of the CLIENTCFG KEY attribute:

        ```
        [root]# vi /opt/moab/etc/moab-private.cfg
        CLIENTCFG[AM:mam] KEY=<MAMSecretKey>
        ```

   - **Native Option.** If you will be using the Native (custom script) accounting manager interface with Moab Workload Manager, do the following:

     a. On the Moab Server Host, edit the Moab configuration file, uncomment the AMCFG lines and set the TYPE to NATIVE:

     ```
     [root]# vi /opt/moab/etc/moab.cfg
     AMCFG[mam] TYPE=NATIVE
     ```

     b. If you are installing MAM on a different host (MAM Server Host) from the Moab Server (Moab Server Host), install the MAM client on the Moab Server Host in order for the custom scripts to use the MAM API.

     On the Moab HPC Suite Server Host, follow the instructions in 2.1.4.D Install Dependencies, Packages, or Clients and 2.1.4.E Install MAM Server; with the following exceptions:

- Install only the dependent packages applicable to MAM Client Hosts.
- Use the configure option `--without-init`.
- Instead of running make, use `make clients-only`.
- Instead of running make install, use `make install-clients-only`.
- Omit the step to create the database and all of the steps thereafter.

2. On the Moab Server Host, restart Moab:

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

2.1.4.K Initialize Moab Accounting Manager

You will need to initialize MAM to function in the way that is most applicable to the needs of your site. See 'Initial Setup' in the *Moab Accounting Manager Administrator Guide* to set up MAM for your desired accounting mode.

Related Topics

- 2.1.1 Preparing for Manual Installation

2.1.5 Installing Moab Web Services

You must deploy Moab Web Services (MWS) on the *same* host as Moab HPC Suite Server (Moab Server Host). For documentation clarity, these instructions refer to the shared host for Moab Server and MWS as the MWS Server Host.

This topic contains instructions on how to install MWS.

In this topic:

2.1.5.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
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

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

2.1.5.B Install Dependencies, Packages, or Clients

Install Java

Install the Linux x64 RPM version of Oracle® Java® 8 Runtime Environment.

> Moab Web Services requires the Oracle Java 8 Runtime Environment. All other distributions and versions of Java, including Java 9, OpenJDK/IcedTea, GNU Compiler for Java, and so on, cannot run Moab Web Services.

On the MWS Server Host, do the following.

1. Install the Linux x64 RPM version of Oracle Java SE 8 JRE:
   a. Go to the Oracle Java download page and download the Linux x64 tarball for Version 8.
   b. Install the tarball for the particular update that you downloaded into `/usr/java`:

```
[root]# mkdir /usr/java
[root]# tar -zxvf jre-8u<update>-linux-x64.tar.gz -C /usr/java
[root]# ln -s jre1.8.0_<update> /usr/java/latest
```

Install Tomcat

1. On the MWS Server Host, do the following:
   - If you are using Ubuntu 18.04, we recommend installing Tomcat 8:

```
[root]# apt install tomcat8
```

   - If you are using Ubuntu 20.04 or 22.04, we recommend installing Tomcat 9.

```
[root]# apt install tomcat9
```

Install MongoDB

> Setting per-user limits on various resources can prevent MongoDB from closing connections if the number of connections grows too high. See Review and Set Resource Limits for more information about using the *ulimit* command to review and set resource limits.

On the MWS MongoDB Database Host, do the following.

1. Add the MongoDB Repository.

   > Moab Web Services version 10.1.0 requires MongoDB version 4.2.

   - If using Ubuntu 18.04:

     ```
     [root]# wget -qO - https://www.mongodb.org/static/pgp/server-4.2.asc | sudo
     apt-key add -
     [root]# echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu
     bionic/mongodb-org/4.2 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-
     org-4.2.list
     ```

   - If using Ubuntu 20.04 or Ubuntu 22.04:

     > Although MongoDB does not provide an official 4.2 repository option for Ubuntu 20.04 or Ubuntu 22.04, it is possible to still install it using the repository for Ubuntu 18.04 by using the following steps.

     ```
     [root]# apt-get install gnupg curl
     [root]# curl -fsSL https://www.mongodb.org/static/pgp/server-4.2.asc | sudo
     apt-key add -
     [root]# echo "deb [ arch=amd64,arm64 ] https://repo.mongodb.org/apt/ubuntu
     bionic/mongodb-org/4.2 multiverse" | sudo tee /etc/apt/sources.list.d/mongodb-
     org-4.2.list
     ```

2. Install MongoDB:

   ```
   [root]# apt update
   [root]# apt install -y mongodb-org
   ```

   > If using Ubuntu 22.04, you may get an error about unmet dependencies on libssl1.1. One way to work around this is to install libssl1.1 from the Ubuntu archive:

   ```
   [root]# wget http://archive.ubuntu.com/ubuntu/pool/main/o/openssl/libssl1.1_
   1.1.1f-1ubuntu2_amd64.deb
   [root]# dpkg -i libssl1.1_1.1.1f-1ubuntu2_amd64.deb
   ```

3. Enable and start MongoDB:

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

4. Add the required MongoDB users.

> The passwords used below (`secret1`, `secret2`, and `secret3`) are examples. Choose your own passwords for these users.

```
[root]# mongo
> use admin
> db.createUser({"user": "admin_user", "pwd": "secret1", "roles": ["root"]})

> use moab
> db.createUser({"user": "moab_user", "pwd": "secret2", "roles": ["dbOwner"]})
> db.createUser({"user": "mws_user", "pwd": "secret3", "roles": ["read"]})

> use mws
> db.createUser({"user": "mws_user", "pwd": "secret3", "roles": ["dbOwner"]})

> exit
```

> Because the `admin_user` has read and write rights to the `admin` database, it also has read and write rights to all other databases. See Control Access to MongoDB Instances with Authentication for more information.

5. Set MongoDB Configuration Options:

   - The configuration file for MongoDB is `/etc/mongod.conf`. See Self-Managed Configuration File Options for information.
   - We recommend that you set `security.authorization` to `enabled`. See security Options for more information.

> By default, `/etc/mongod.conf` sets `net.bindIp` to 127.0.0.1. You will need to change this setting if the MongoDB server needs to be accessible from other hosts or from other interfaces besides loopback. See net Options for more information.

```
# Sample /etc/mongod.conf file
net:
  port: 27017
  # bindIp: 127.0.0.1
processManagement:
  fork: true
  pidFilePath: /var/run/mongodb/mongod.pid
security:
  authorization: enabled
storage:
  dbPath: /var/lib/mongo
```

```
  journal:
    enabled: true
systemLog:
  destination: file
  logAppend: true
  path: /var/log/mongodb/mongod.log
```

6. Restart MongoDB:

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

2.1.5.C Install MWS Server

> You must complete the tasks to install the dependencies, packages, or clients before installing MWS Server. See 2.1.5.B Install Dependencies, Packages, or Clients.
>
> If your configuration uses firewalls, you *must also* open the necessary ports before installing the MWS Server. See 2.1.5.A Open Necessary Ports .

On the MWS Server Host, do the following.

1. Verify Moab HPC Suite Server is installed and configured as desired (for details, see 2.1.3 Installing Moab Workload Manager).
2. Start Moab:

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

3. Create the MWS home directory and subdirectories. For more information, see 'Configuration' in the *Moab Web Services Administrator Guide*.

   > The default location for the MWS home directory is `/opt/mws`. These instructions assume the default location.

   Do the following:

   - If using Ubuntu 18.04:

```
[root]# mkdir -p \
    /opt/mws/etc/mws.d \
    /opt/mws/hooks \
    /opt/mws/log \
    /opt/mws/plugins \
    /opt/mws/spool/hooks \
    /opt/mws/utils
[root]# chown -R tomcat8:tomcat8 /opt/mws
[root]# chmod -R 555 /opt/mws
[root]# chmod u+w \
    /opt/mws/log \
```

```
    /opt/mws/plugins \
    /opt/mws/spool \
    /opt/mws/spool/hooks \
    /opt/mws/utils
```

- If using Ubuntu 20.04 or Ubuntu 22.04:

```
[root]# mkdir -p \
    /opt/mws/etc/mws.d \
    /opt/mws/hooks \
    /opt/mws/log \
    /opt/mws/plugins \
    /opt/mws/spool/hooks \
    /opt/mws/utils
[root]# chown -R tomcat:tomcat /opt/mws
[root]# chmod -R 555 /opt/mws
[root]# chmod u+w \
    /opt/mws/log \
    /opt/mws/plugins \
    /opt/mws/spool \
    /opt/mws/spool/hooks \
    /opt/mws/utils
```

4. Download the latest MWS tarball from the Adaptive Computing website.
5. Extract the contents into a temporary directory, for example:

```
[root]# mkdir /tmp/mws-install
[root]# cd /tmp/mws-install
[root]# tar xvzf $HOME/Downloads/mws-10.1.0.tar.gz
```

6. Copy the extracted utility files to the utility directory created in the previous step and give the Tomcat user ownership of the directory:

   - If using Ubuntu 18.04:

```
[root]# cd mws-10.1.0/utils
[root]# cp * /opt/mws/utils
[root]# chown tomcat8:tomcat8 /opt/mws/utils/*
```

   - If using Ubuntu 20.04 or Ubuntu 22.04:

```
[root]# cd mws-10.1.0/utils
[root]# cp * /opt/mws/utils
[root]# chown tomcat:tomcat /opt/mws/utils/*
```

7. Connect Moab to MongoDB.

   The USEDATABASE parameter is unrelated to the MongoDB configuration.

   a. Set the `MONGOSERVER` parameter in `/opt/moab/etc/moab.cfg` to the MongoDB server hostname. Use localhost as the hostname if Moab and MongoDB are hosted on the same server.

```
MONGOSERVER <host>[:<port>]
```

If your MONGOSERVER host is set to anything other than localhost, edit the /etc/mongod.conf file on the MongoDB server host and either comment out any bind_ip parameter or set it to the correct IP address:

```
net:
  port: 27017
  #bindIp: 127.0.0.1  # Listen to local interface only. Comment out to listen on
all interfaces.
```

b. In the /opt/moab/etc/moab-private.cfg file, set the MONGOUSER and MONGOPASSWORD parameters to the MongoDB moab_user credentials you set. See Install MongoDB above.

```
MONGOUSER     moab_user
MONGOPASSWORD secret2
```

c. Verify that Moab is able to connect to MongoDB:

```
[root]# systemctl restart moab.service
[root]# mdiag -S | grep Mongo

Mongo connection (localhost [replicaset: not set]) is up (credentials are set
and SSL is disabled)
```

8. Secure communication using secret keys.

a. (Required) Moab and MWS use Message Authentication Codes (MAC) to ensure messages have not been altered or corrupted in transit. Generate a key and store the result in /opt/moab/etc/.moab.key:

```
[root]# systemctl stop moab.service
[root]# dd if=/dev/urandom count=24 bs=1 2>/dev/null | base64 >
/opt/moab/etc/.moab.key
[root]# chown root:root /opt/moab/etc/.moab.key
[root]# chmod 400 /opt/moab/etc/.moab.key
[root]# systemctl start moab.service
```

b. (Optional) Moab supports message queue security using AES. This feature requires a Base64-encoded 16-byte (128-bit) shared secret. Do the following:

i. Generate a key and append the result to /opt/moab/etc/moab-private.cfg:

```
[root]# systemctl stop moab.service
[root]# echo "MESSAGEQUEUESECRETKEY $(dd if=/dev/urandom count=16 bs=1
2>/dev/null | base64)" >> /opt/moab/etc/moab-private.cfg
[root]# systemctl start moab.service
```

> If MWS is configured to encrypt the message queue and Moab is not (or vice versa), then MWS will ignore the messages from Moab. Furthermore, all attempts to access the MWS service resource will fail.

ii. Verify that encryption is on for the ZeroMQ connection:

```
[root]# mdiag -S|grep 'ZeroMQ MWS'

 ZeroMQ MWS connection is bound on port 5570 (encryption is on)
```

9. Set up the MWS configuration files. In the extracted directory are several configuration files.

   a. Copy the configuration files into place and grant the Tomcat user ownership.

   - If using Ubuntu 18.04:

```
[root]# cd /tmp/mws-install/mws-10.1.0
[root]# cp mws-config.groovy logback.groovy /opt/mws/etc
[root]# cp mws-config-hpc.groovy /opt/mws/etc/mws.d
[root]# chown tomcat8:tomcat8 /opt/mws/etc/mws-config.groovy
/opt/mws/etc/logback.groovy /opt/mws/etc/mws.d/mws-config-hpc.groovy
```

   - If using Ubuntu 20.04 or Ubuntu 22.04:

```
[root]# cd /tmp/mws-install/mws-10.1.0
[root]# cp mws-config.groovy logback.groovy /opt/mws/etc
[root]# cp mws-config-hpc.groovy /opt/mws/etc/mws.d
[root]# chown tomcat:tomcat /opt/mws/etc/mws-config.groovy
/opt/mws/etc/logback.groovy /opt/mws/etc/mws.d/mws-config-hpc.groovy
```

   b. In the `/opt/mws/etc/mws-config.groovy` file, change these settings:

   - `moab.secretKey`: Must match the Moab HPC Suite secret key you generated earlier (contained in `/opt/moab/etc/.moab.key`).
   - `auth.defaultUser.username`: Any value you like, or leave as is.
   - `auth.defaultUser.password`: Any value you like, but choose a strong password.
   - `moab.messageQueue.secretKey`: If you opted to configure a message queue security key in MWS, this parameter value should match exactly that key specified in `/opt/moab/etc/moab-private.cfg` for the MESSAGEQUEUESECRETKEY Moab configuration parameter you generated earlier.

> If MWS is configured to encrypt the message queue and Moab is not (or vice versa), then the messages from Moab will be ignored. Furthermore, all attempts to access the MWS service resource will fail.

```
[root]# vi /opt/mws/etc/mws-config.groovy

// Change these to be whatever you like.
auth.defaultUser.username = "moab-admin"
auth.defaultUser.password = "changeme!"


// Replace <ENTER-KEY-HERE> with the contents of /opt/moab/etc/.moab.key.
moab.secretKey = "<ENTER-KEY-HERE>"
moab.server = "localhost"
moab.port = 42559
moab.messageDigestAlgorithm = "SHA-1"


...

// Replace <ENTER-KEY-HERE> with the value of MESSAGEQUEUESECRETKEY in
/opt/moab/etc/moab-private.cfg.
moab.messageQueue.secretKey = "<ENTER-KEY-HERE>"
```

If you do not change `auth.defaultUser.password`, your MWS will not be secure (because anyone reading these instructions would be able to log in to your MWS).

c. Do *one* of the following.

You can configure only one authentication method in `/opt/mws/etc/mws-config.groovy`—LDAP or PAM, but not both. If you have configured both LDAP and PAM, MWS defaults to using LDAP.

If you need multiple authentication methods, you must add them to your local PAM configuration. See your distribution documentation for details.

- If you are configuring a MWS connection to your LDAP server, add the following parameters to the `/opt/mws/etc/mws-config.groovy` file:

```
ldap.server  = "192.168.0.5"
ldap.port = 389
ldap.baseDNs = ["dc=acme,dc=com"]
ldap.bindUser = "cn=Manager,dc=acme,dc=com"
ldap.password = "*****"
ldap.directory.type = "OpenLDAP Using InetOrgPerson Schema"
```

This is just an example LDAP connection. Be sure to use the appropriate domain controllers (dc) and common names (cn) for your environment.

To see how to configure a secure connection to the LDAP server, see 'Securing the LDAP Connection' in the *Moab Web Services Administrator Guide*.

- If you are configuring MWS to use PAM, add the `pam.configuration.service` parameter to the `/opt/mws/etc/mws-config.groovy` file, for example:

```
pam.configuration.service = "login"
```

This is just an example PAM configuration file name. Make sure you specify the name of the configuration file you want MWS to use.

If you configure MWS to authenticate via PAM using local files or NIS, you need to run Tomcat as root. This configuration is highly discouraged and is not supported by Adaptive Computing. The recommended approach is to configure PAM and NSS to authenticate against LDAP.

For more information about PAM configuration with MWS, see 'PAM (Pluggable Authentication Module) Configuration Using /opt/mws/etc/mws-config.groovy' in the *Moab Web Services Administrator Guide*.

d. Add the `grails.mongodb.username` and `grails.mongo.password` parameters to the `/opt/mws/etc/mws-config.groovy` file. Use the MWS credentials you added to MongoDB.

```
...
grails.mongodb.username = "mws_user"
grails.mongodb.password = "secret3"
```

e. Make the MWS configuration files read-only:

```
[root]# chmod 400 /opt/mws/etc/mws-config.groovy /opt/mws/etc/logback.groovy
/opt/mws/etc/mws.d/mws-config-hpc.groovy
```

10. Configure Tomcat by doing the following:

- If using Ubuntu 18.04:

```
[root]# vim /etc/default/tomcat8
JAVA_HOME="/usr/java/latest"
[root]# vim /usr/share/tomcat8/bin/setenv.sh
CATALINA_OPTS="-DMWS_HOME=/opt/mws -Xms256m -Xmx3g -Dfile.encoding=UTF8"
[root]# chmod +x /usr/share/tomcat8/bin/setenv.sh
```

- If using Ubuntu 20.04 or Ubuntu 22.04:

```
[root]# vim /usr/share/tomcat9/bin/setenv.sh
JAVA_HOME="/usr/java/latest"
CATALINA_OPTS="-DMWS_HOME=/opt/mws -Xms256m -Xmx3g -Dfile.encoding=UTF8"
[root]# chmod +x /usr/share/tomcat9/bin/setenv.sh
[root]# vim /lib/systemd/system/tomcat9.service
[Service]
ReadWritePaths=/opt/mws
```

```
[root]# systemctl daemon-reload
```

11. Deploy the `mws.war` file and start Tomcat:

    - If using Ubuntu 18.04:

      ```
      [root]# systemctl stop tomcat8.service
      [root]# cp /tmp/mws-install/mws-10.1.0/mws.war /var/lib/tomcat8/webapps
      [root]# systemctl start tomcat8.service
      ```

    - If using Ubuntu 20.04 or Ubuntu 22.04:

      ```
      [root]# systemctl stop tomcat9.service
      [root]# cp /tmp/mws-install/mws-10.1.0/mws.war /var/lib/tomcat9/webapps
      [root]# systemctl start tomcat9.service
      ```

12. Navigate to `http://<server>:8080/mws/` in a browser to verify that MWS is running (you will see some sample queries and a few other actions).

13. Log in to MWS to verify that your credentials are working. (Your login credentials are the `auth.defaultUser.username` and `auth.defaultUser.password` values you set in the `/opt/mws/etc/mws-config.groovy` file.)

> If you encounter problems, or if the application does not seem to be running, see the steps in 3.4 Moab Web Services Issues.

Related Topics

- 2.1.1 Preparing for Manual Installation

2.2 Additional Configuration

In this section:

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 3.2 Port Reference for the actual port numbers for the various products.

Ubuntu-based systems use ufw 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 ufw:

```
[root]# ufw allow 1234/tcp
```

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 configure options. For a complete list, use *./configure --help* when configuring Moab HPC Suite.

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

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

Option	Description	Example
	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.	
--without-profile	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.	`[root]# ./configure --without-profile`

2.2.4 Moab Accounting Manager Configuration Options

The following table comprises commonly-used configure options:

Option	Description
-h,--help	Run `./configure --help` to see the list of configure options.
--localstatedir=DIR	Home directory where per-configuration subdirectories

Option	Description
	(such as `etc`, `log`, `data`) will be installed (defaults to `PREFIX`).
--prefix=PREFIX	Base installation directory where all subdirectories will be installed unless otherwise designated (defaults to `/opt/mam`).
--with-cgi-bin=DIR	If you intend to use the web GUI, use `--with-cgi-bin` to specify the directory where you want the Moab Accounting Manager CGI files to reside (defaults to `/var/www/cgi-bin/mam`).
--with-db-name=NAME	Name of the SQL database that the server will sync with (defaults to `mam`).
--with-legacy-links	Creates symbolic links allowing the use of the old client and server command names (for example, *`mam-list-users`* would be created as symbolic link to *`mam-list-users`*). When running a command under its old name, the command will issue a deprecation warning. This warning can be disabled by setting `client.deprecationwarning = false` in the `mam-client.conf` file. The default is not to install the legacy links.
--with-mam-libs=local|site	Use `--with-mam-libs` to indicate whether you want to install the Perl MAM modules in a local directory (`${exec_prefix}/lib`) or in the default system site-perl directory (defaults to `local`).
--with-promotion=mamauth|suidperl	Command-line clients and scripts using the API need to use a privilege promotion method to authenticate and encrypt the communication using the symmetric key. The default is suidperl if it is installed on the system, otherwise the default is `mamauth`.
--with-user=USER	Use `--with-user` to specify the accounting admin userid that the server will run under and who will have full administrative privileges (defaults to mam). We recommend that this be a non-privileged user for the highest security.
--without-gui	Specifies whether to install the CGI web GUI. If you do not intend to use the CGI web GUI, you can specify `--`

Option	Description
	`without-gui` to not install the CGI scripts. Otherwise, the default is to install the GUI CGI scripts.
--without-init	If you do not intend to use the mam `init.d` service, you can use `--without-init` to specify that Moab HPC Suite should not install the mam `init.d` script. Otherwise, the script is installed by default.
--with[out]-pam	Indicates whether to use PAM authentication for the GUI and web services. If `--with-pam` is specified, the PAM configuration file is installed and the GUI and web services will default to using `pam` as the authentication method. If `--without-pam` is specified, the PAM configuration file is not installed and the GUI and web services will default to using `mam-password` as the authentication method. If this option is not specified, the PAM configuration file is installed but the GUI and web services will default to using `mam-password` as the authentication method. When `--with-pam` option is specified, the accounting admin user will default to root, unless overridden with the `--with-user` configuration option.
--without-profile	If you do not intend to use the mam `profile.d` environment scripts, you can use `--without-profile` to specify that Moab HPC Suite should not install the mam `profile.d` scripts. Otherwise, the scripts are installed by default.

2.2.5 Trusting Servers in Java

In this topic:

2.2.5.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.

2.2.5.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.

2.2.5.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 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

Torque 7.1 is not backward compatible with Torque versions prior to 7.0. When you upgrade to Torque 7.1 from versions prior to 7.0, the server, moms, and clients must be upgraded at the same time.

The job format is compatible between 7.1 and previous versions of Torque and any queued jobs will upgrade to the new version. We do *not* recommend upgrading Torque while jobs are in a running state.

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

If you need to upgrade a Torque version prior to 4.0, contact Adaptive Computing.

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

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]# apt install libhwloc-dev pkg-config
[root]# cd hwloc-1.9.1
[root]# make uninstall
```

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

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

GPU Support

Because Torque GPU support has evolved over time, upgrading may require a re-examination of the cluster's GPU setup, especially if the upgrade will include configuration changes to take advantage of cgroups and/or NVIDIA/NVML support. See 'Scheduling GPUs' in the Accelerators chapter of the *Moab Workload Manager Administrator Guide* for an overview of currently-available options.

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 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
```

2.3.1.C Upgrade the Torque Server

You *must* complete all the previous upgrade steps in this topic before upgrading 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. When cgroups are supported, cpusets are handled by the cgroup cpuset subsystem. If you are not using cgroups, use `--enable-cpusets` instead.

> If `--enable-gui` is part of your configuration, run the following commands:
>
> ```
> $ cd /usr/lib64
> $ ln -s libXext.so.6.4.0 libXext.so
> $ ln -s libXss.so.1 libXss.so
> ```
>
> When finished, cd back to your `install` directory.

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-7.1.0.tar.gz
[root]# cd torque-7.1.0
[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>:/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>:/lib/systemd/system/
   ```

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

   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 a number of 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

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-10.1.0-<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-10.1.0-<OS>
```

4. Configure the installation package. Use the same configure 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 configure options, use `./configure --help`.
5. Stop Moab:

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

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.

6. Back up your Moab Workload Manager home directory (`/opt/moab/` by default) before continuing.
7. Install Moab:

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

> Default configuration files are installed during *make install*. Existing configuration files are not overwritten and the new files are given a .dist extension.

8. 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.
9. 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:  /tmp/jenkins/workspace/MWM-10.1.0/label/build-<OS>
Build host: us-devops-build10
Build date: Oct 09 13:00:00 MST 2024
Build args: NA
Compiler Flags:  -D__M64 -D_BUILDDATETIME="2024100913" -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 10.1.0 (revision 2024100913, changeset
14dee972ebcee919207e48054e9f285db9f6a555)
```

10. Start Moab:

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

2.3.3 Upgrading Moab Accounting Manager

This topic provides instructions to upgrade Moab Accounting Manager (MAM) to the latest release version. It includes instructions for migrating your database schema to a new version if necessary.

MAM uses the standard *configure*, *make*, and *make install* steps for upgrades. This document provides a number of 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.

On the MAM Server Host, do the following.

1. Determine the MAM Accounting Admin user and change to that user:

```
[root]# mam-list-users | grep 'Accounting Admin'
mam      True                                                    Accounting
Admin
[root]# su - mam
```

2. Determine whether a migration is necessary.

   a. Determine your database version:

```
[mam]$ mam-shell System Query
```

   b. If the current version is lower than 10.1, then your database and configuration files will need to be migrated. The steps required to do so are incorporated in the remaining steps for this topic.

3. Stop the server daemon:

```
[mam]$ su -c "systemctl stop mam.service"
```

4. If a migration is required, create a database backup:

```
[mam]$ pg_dump -U <mam_database_user> -W <old_database_name> > /tmp/<old_database_
name>.sql
```

> MySQL is no longer a supported database for MAM. If you are using MySQL for your MAM database, follow the instructions in 2.3.5 Migrating the MAM Database from MySQL to PostgreSQL to convert your database.

5. Verify that each of the prerequisites listed in 2.1.4 Installing Moab Accounting Manager have been satisfied.

6. Download the latest MAM tarball from the Adaptive Computing website.

7. Unpack the tar archive and change directory into the top directory of the distribution:

```
[mam]$ tar -zxvf mam-10.1.0.tar.gz
[mam]$ cd mam-10.1.0
```

8. Configure MAM by running *configure* with the desired options.

   We recommend that you use the same configure options that were used in the previous installation. You can examine the config.log file where you unpacked your previous distribution to help determine the configuration options that were used to install the prior version of MAM.

```
[mam]$ ./configure
```

9. Run *make* to compile the program:

```
[mam]$ make
```

> Depending on your configuration, you may need to replace 'make' with a make command that includes additional functionality. Specifically:
>
> - If you only need to install the clients on a particular system, use `make clients-only`.
> - If you only need to install the web GUI on a particular system, use `make gui-only`.
> - If you only need to install the web services on a particular system, use `make ws-only`.

10. Run *`make install`* as root to install MAM:

```
[mam]$ su -c "make install"
```

> Depending on your configuration, you may need to replace 'make install' with a make command that includes additional functionality. Specifically:
>
> - If you only need to install the clients on a particular system, use `make install-clients-only`.
> - If you only need to install the web GUI on a particular system, use `make install-gui-only`.
> - If you only need to install the web services on a particular system, use `make install-ws-only`.

11. Start the server daemon:

```
[mam]$ su -c "systemctl daemon-reload"
[mam]$ su -c "systemctl start mam.service"
```

12. If you need to migrate, you will do so by running one or more migration scripts. You must run every incremental migration script between the version you are currently using and the new version (10.1). These scripts are designed to be rerunnable, so if you encounter a failure, resolve the failure and rerun the migration script. If you are unable to resolve the failure and complete the migration, contact Support.

    For example, if you are migrating from MAM version 9.1, you must run two migration scripts: the first to migrate the database schema from 9.1 to 10.0, and the second to migrate the database schema from 10.0 to 10.1:

```
[mam]$ sbin/migrate_9.1-10.0.pl
[mam]$ sbin/migrate_10.0-10.1.pl
```

13. Verify that the resulting database schema version is 10.1:

```
[mam]$ mam-shell System Query

Name                    Version Description
----------------------- ------- ------------------
Moab Accounting Manager 10.1    Commercial Release
```

14. Verify that the executables have been upgraded to 10.1.0:

```
[mam]$ mam-server -v

Moab Accounting Manager version 10.1.0
```

15. If you are upgrading MAM from a version prior to 9.1.0, and you want to use MAM Web Services, perform the following procedures (provided in the Installing Moab Accounting Manager topic):
    - 2.1.4.G Configure MAM Web Services
    - 2.1.4.I Access MAM Web Services

2.3.4 Upgrading Moab Web Services

This topic provides instructions to upgrade Moab Web Services (MWS) to the latest release version.

These instructions assume you are upgrading MWS from version 8.0 or later. If you are upgrading MWS from a version prior to 8.0, contact your Adaptive Computing account manager for more information.

You must deploy MWS on the *same* host as Moab HPC Suite Server (Moab Server Host). For documentation clarity, these instructions refer to the host for Moab Server and MWS Server as the MWS Server Host.

In this topic:

2.3.4.A Before You Upgrade

This section provides instructions for tasks that need to be performed before you upgrade MWS.

Upgrade to Java 8

> Moab Web Services requires the Oracle Java 8 Runtime Environment. All other distributions and versions of Java, including Java 9, OpenJDK/IcedTea, GNU Compiler for Java, and so on, cannot run Moab Web Services.

If you want to upgrade to Java 8, refer to the Install Java instructions.

2.3.4.B Back up the MongoDB Databases

On the MWS MongoDB server host, do the following.

1. Stop all services that are using the MongoDB databases.
2. Back up the MongoDB databases:

```
[root]# cd /root
[root]# mongodump -u admin_user -p secret1
```

3. Restart the services.

2.3.4.C Upgrade Moab Web Services

> You `must` complete the tasks in 2.3.4.A Before You Upgrade before upgrading MWS.

On the MWS Server Host, do the following.

1. Create a directory for which you will extract the contents of the MWS download tarball, for example:

```
[root]# mkdir /tmp/mws-install
[root]# cd /tmp/mws-install
```

2. Download the latest MWS tarball from the Adaptive Computing website.
3. In the directory you created earlier, extract the contents and then change directory into the extracted directory, for example:

```
[root]# tar xvzf mws-10.1.0.tar.gz
[root]# cd mws-10.1.0
```

4. Deploy the updated `mws.war` to Tomcat:
   - If using Ubuntu 18.04:

```
[root]# systemctl stop tomcat8.service
[root]# rm -rf /var/lib/tomcat8/webapps/mws /var/lib/tomcat8/webapps/mws.war
[root]# cp mws.war /var/lib/tomcat8/webapps
[root]# chown tomcat8:tomcat8 /var/lib/tomcat8/webapps/mws.war
```

   - If using Ubuntu 20.04 or Ubuntu 22.04:

```
[root]# systemctl stop tomcat9.service
[root]# rm -rf /var/lib/tomcat9/webapps/mws /var/lib/tomcat9/webapps/mws.war
[root]# cp mws.war /var/lib/tomcat9/webapps
[root]# chown tomcat:tomcat /var/lib/tomcat8/webapps/mws.war
```

5. Back up the MWS home directory:

```
[root]# cp -rp /opt/mws /opt/mws-<version>-backup
```

   Where <version> is the product version being backed up.

6. Copy the extracted utility files to the utility directory created above and give the Tomcat user ownership of the directory:
   - If using Ubuntu 18.04:

```
[root]# cd utils
[root]# \cp * /opt/mws/utils
[root]# chown tomcat8:tomcat8 /opt/mws/utils/*
[root]# cd ..
```

   - If using Ubuntu 20.04 or Ubuntu 22.04:

```
[root]# cd utils
[root]# \cp * /opt/mws/utils
[root]# chown tomcat:tomcat /opt/mws/utils/*
[root]# cd ..
```

7. Merge the changes in the /tmp/mws-install/mws-10.1.0/`mws-config.groovy` file into your existing `/opt/mws/etc/mws-config.groovy`.
   a. Depending on your current MWS version, do the following as needed:
      - Replace parameters starting with "grails.mongo" with "grails.mongodb"; prior to version 10.1.
      - Remove the log4j configuration; prior to version 10.1.
      - If Viewpoint is part of your configuration, replace the grails.plugin.springsecurity.oauthProvider.clients configuration with viewpoint.clientSecret in the form:

```
viewpoint.clientSecret = "<ENTER-CLIENTSECRET-HERE>"
```

replacing <ENTER-CLIENTSECRET-HERE> with your client secret (password) for Viewpoint; prior to version 10.1.

- If Insight is part of your configuration:
  - **remove** the Insight PostgreSQL information (dataSource_insight.username, dataSource_insight.password, dataSource_insight.url); prior to version 9.1.

    > Version 9.1 removed the Insight PostreSQL database.

  - add the health check information for the Insight Server (insight.server, insight.command.port, insight.command.timeout.seconds); prior to version 9.0.2.

    > insight.server is the DNS name of the host on which the Insight Server is running.

b. Confirm the value for moab.messageQueue.secretKey matches the value located in /opt/moab/etc/moab-private.cfg; if you have not yet configured a secret key, see Secure communication using secret keys.

Example of the merged `/opt/mws/etc/mws-config.groovy` file for MWS 10.1.0:

```
// Any settings in this file may be overridden by any
// file in the mws.d directory.

// Change these to be whatever you like.
auth.defaultUser.username = "moab-admin"
auth.defaultUser.password = "changeme!"

// Moab Workload Manager configuration.
moab.secretKey = "<ENTER-KEY-HERE>"
moab.server = "localhost"
moab.port = 42559
moab.messageDigestAlgorithm = "SHA-1"

// MongoDB configuration.
// grails.mongodb.host = "127.0.0.1"
// grails.mongodb.port = 27017
grails.mongodb.username = "mws_user"
grails.mongodb.password = "<ENTER-KEY-HERE>"

// Insight configuration.
// insight.server = "localhost"
// insight.command.port = 5568
// insight.command.timeout.seconds = 5

// Message bus configuration.
moab.messageQueue.port = 5570
// moab.messageQueue.secretKey = "<ENTER-KEY-HERE>"
mws.messageQueue.address = "*"
mws.messageQueue.port = 5564

// Viewpoint Configuration
```

```
viewpoint.clientSecret = "<ENTER-CLIENTSECRET-HERE>"

// Sample LDAP Configurations

// Sample OpenLDAP Configuration
//ldap.server = "192.168.0.5"
//ldap.port = 389
//ldap.baseDNs = ["dc=acme,dc=com"]
//ldap.bindUser = "cn=Manager,dc=acme,dc=com"
//ldap.password = "*****"
//ldap.directory.type = "OpenLDAP Using InetOrgPerson Schema"

// Sample Active Directory Configuration
//ldap.server = "192.168.0.5"
//ldap.port = 389
//ldap.baseDNs = ["CN=Users,DC=acme,DC=com","OU=Europe,DC=acme,DC=com"]
//ldap.bindUser = "cn=Administrator,cn=Users,DC=acme,DC=com"
//ldap.password = "*****"
//ldap.directory.type = "Microsoft Active Directory"
```

8. Merge any changes supplied in the new `mws-config-hpc.groovy` file in to your installed `/opt/mws/etc/mws.d/mws-config-hpc.groovy`.
9. If you are upgrading from a version of MWS prior to 10.1, copy the new logback.groovy file into the MWS configuration directory:

```
[root]# cp logback.groovy /opt/mws/etc
[root]# chown tomcat:tomcat /opt/mws/etc/logback.groovy
```

10. Remove all plugins from `/opt/mws/plugins` except for those that you may have created. The presence of obsolete plugins can prevent MWS from starting up. Out-of-the-box plugins will be recreated when MWS is restarted.

```
[root]# cd /opt/mws/plugins
[root]# rm *.jar
```

11. Verify the Tomcat user has read access to the `/opt/mws/etc/mws-config.groovy` and `/opt/mws/etc/mws.d/mws-config-hpc.groovy` file.
12. Start Tomcat:
    - If using Ubuntu 18.04:

      ```
      [root]# systemctl start tomcat8.service
      ```

    - If using Ubuntu 20.04 or Ubuntu 22.04:

      ```
      [root]# systemctl start tomcat9.service
      ```

13. Visit http://localhost:8080/mws/ in a browser to verify that MWS is running again. You will see some sample queries and a few other actions.
14. Log in to MWS to verify configuration. (The credentials are the values of `auth.defaultUser.username` and `auth.defaultUser.password` set in `/opt/mws/etc/mws-config.groovy`.)

If you encounter problems, or if MWS does not seem to be running, see the steps in 3.4 Moab Web Services Issues.

2.3.5 Migrating the MAM Database from MySQL to PostgreSQL

PostgreSQL is the preferred DBMS for MAM. If you have already installed MySQL as the DBMS for MAM, you are not required to migrate their database to use PostgreSQL at this time. However, MySQL is considered deprecated and new installations will only use PostgreSQL.

PostgreSQL does not provide a standard procedure for migrating an existing database from MySQL to PostgreSQL. Adaptive Computing has had success using the py-mysql2pgsql tools for migrating/converting/exporting data from MySQL to PostgreSQL. See https://github.com/philipsoutham/py-mysql2pgsql for additional details.

To Migrate the MAM Database

This procedure was successfully tested on an actual customer MySQL database with millions of transactions. It completed in less than an hour.

1. Make a backup copy of your MySQL mam database:

```
[root]# mysqldump mam > /archive/mam.mysql
```

2. Follow the instructions to install PostgreSQL:
   - **Manual Install** - 2.1.4.C Install and Initialize PostgreSQL Server
3. Install the prerequisite packages:

```
[root]# zypper install git postgresql-devel gcc MySQL-python python-psycopg2 PyYAML
termcolor python-devel
```

4. Install pg-mysql2pgsql (from source):

```
[root]# cd /software
[root]# git clone git://github.com/philipsoutham/py-mysql2pgsql.git
[root]# cd py-mysql2pgsql
[root]# python setup.py install
```

5. Run pg-mysql2pgsql once to create a template yaml config file:

```
[root]# py-mysql2pgsql -v
```

6. Edit the config file to specify the MySQL database connection information and a file to output the result:

```
[root]# vi mysql2pgsql.yml
```

```
mysql:
hostname: localhost
port: 3306
socket:
username: mam
password: changeme
database: mam
compress: false
destination:
# if file is given, output goes to file, else postgres
file: /archive/mam.pgsql
postgres:
hostname: localhost
port: 5432
username:
password:
database:
```

7. Run the pg-mysql2pgsql program again to convert the database:

```
[root]# py-mysql2pgsql -v
```

8. Create the mam database in PostgreSQL:

```
[root]# su - postgres
[postgres]$ psql
postgres=# create database "mam";
postgres=# create user mam with password 'changeme!';
postgres=# \q
[postgres]$ exit
```

9. Import the converted data into the PostgreSQL database:

```
[root]# su - mam
[mam]$ psql mam < /archive/mam.pgsql
```

10. Point MAM to use the new postgresql database:

```
[mam]$ cd /software/mam-latest
[mam]$ ./configure                  # This will generate an etc/mam-
server.conf.dist file
[mam]$ vi /opt/mam/etc/mam-server.conf   # Merge in the database.datasource from
etc/mam-server.conf.dist
```

11. Restart MAM:

```
[mam]$ mam-server -r
```

Chapter 3: Troubleshooting

This chapter details some common problems and general solutions. Additional troubleshooting can be found in the individual Moab HPC Suite 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:

3.1 General Issues

This section details some common problems and general solutions.

In this section:

3.1.1 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:

3.1.1.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
reporting	reporting_user	secret6	/opt/reporting/application.conf	database.username, database.password

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!

3.1.1.B Product Credentials

Moab Workload Manager

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

Declared Parameter		Used By		Default Value
File	Parameter Name	File	Parameter Name	
/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:mws] USERNAME	
/opt/mws/etc/mws-config.groovy	auth.defaultUser.password	https://<viewpoint_server>:8081/configuration/	Password	changeme!
		/opt/moab/etc/moab-private.cfg	CLIENTCFG	

Declared Parameter		Used By		Default Value
File	Parameter Name	File	Parameter Name	
			[RM:mws] PASSWORD	
/opt/mws/etc/mws-config.groovy	grails.plugin.springsecurity.oauthProvider.clients[0].clientSecret	https://<viewpoint_server>:8081/configuration/	Client Secret	N/A

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.

3.2 Port Reference

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

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 *or* 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 *or* 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

3.3 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:

3.3.1 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
...
```

3.3.2 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
...
```

3.4 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`.

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:

3.4.1 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) 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.

3.4.2 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.

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.

3.4.3 MongoDB: Connection wait timeout after 120000 ms

See MongoDB: Out of semaphores to get db connection above.

3.4.4 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.

Beginning with Java 8, the MaxPermSize option is ignored.

3.4.5 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"
```

3.4.6 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
```

3.4.7 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
```

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