Moab HPC Suite - Basic Edition

Installation and Configuration Guide 9.0.0

October 2015

Adaptive Computing Enterprises, Inc.
1712 S. East Bay Blvd., Suite 300
Provo, UT 84606
+1 (801) 717-3700
www.adaptivecomputing.com

Scan to open online help

Welcome

Welcome to the Moab HPC Suite – Basic Edition 9.0.0 Installation and Configuration Guide, which will help you install or upgrade and configure your Moab HPC Suite. 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 – Basic Edition. The Moab HPC Suite – Basic Edition 9.0.0 contains the following components:

- Moab Workload Manager 9.0.0
- Moab Web Services 9.0.0
- Torque 6.0.0

Before commencing the installation or upgrade, please see Chapter 1 Planning your Installation on page 3 to verify your system conforms to minimum prerequisites.

Chapter 1 Planning your Installation

It is highly recommended 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. Please 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:

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 an Moab HPC Suite component (server or client) is installed.

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

Where to Start

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

1. Determine whether you have a small, medium, High-Throughput or large environment; including an example, and required and recommended hardware requirements. See Suite Topology on page 4.
2. Decide whether you will perform a Manual Installation or an RPM Installation for the various components. See Identify the Manual or RPM Installation Methods on page 9.

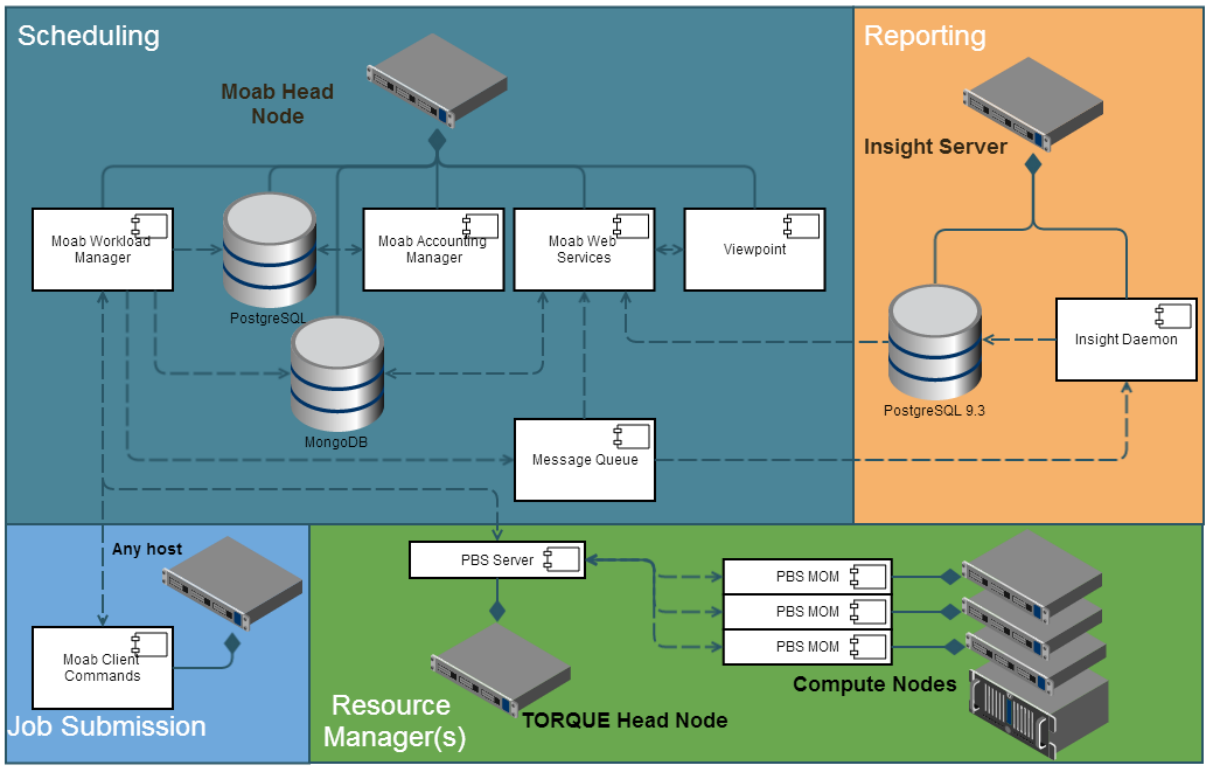
   > The Manual Installation and the RPM Installation chapters each have an "Additional Configuration" section that provides additional information and instructions for optional, but recommended configurations (for example, Configuring SSL in Tomcat).

3. Review the software requirements for your components and set up your hosts accordingly. See Component Requirements on page 10.
4. Install the individual components on their respective host(s). See Preparing for Manual Installation on page 16 or Preparing the Host for RPM Installations on page 78 as applicable.
5. Refer to Chapter 4 Troubleshooting on page 119 for assistance in addressing common problems during installation and configuration.

Suite Topology

The Moab HPC Suite is installed and configured differently for small, medium or large environment types. See Environment Requirements on page 6.

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



Please note the following:

- Moab Accounting Manager is available only with the Moab HPC Enterprise Suite.
- Moab Insight is available only with the Moab HPC Enterprise Suite.
- Moab Viewpoint is available only with the Moab HPC Enterprise Suite.
- 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 may share the same database instance, it is not a requirement. Two database instances may 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 Host (i.e. databases, Moab Accounting Manager, and/or Viewpoint) to their own respective servers.
- The Message Queue component is fulfilled by ZeroMQ™. The libraries are provided with the components that use the message queue and are enabled via configuration; no special installation is necessary.

Environment Requirements

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

Software requirements are listed per-component rather than suite-wide as the suite components reside on different hosts. See Component Requirements on page 10

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+Torque Host** • 4 Intel/AMD x86-64 cores • At least 8 GB RAM • At least 100 GB dedicated disk space **Insight Host** • 4 Intel/AMD x86-64 cores • At least 8 GB RAM • At least 256 GB dedicated disk space	Same as minimum

Environment Type	# of Compute Nodes	Jobs/ Week	Minimum Requirements (per Host Distribution)	Recommended Requirements (targeting minimum number of hosts)
Medium	500	<100k	**Moab+Torque Host** • 8 Intel/AMD x86-64 cores • At least 16 GB RAM • At least 512 GB dedicated disk space **Insight Host** • 8 Intel/AMD x86-64 cores • At least 8 GB of RAM; a dedicated 1 Gbit channel between Insight and Moab • 128 GB local SSD for swapAt least 512 GB disk	**Moab+Torque Host** • 16 Intel/AMD x86-64 cores • At least 32 GB RAM • At least 1 TB dedicated disk space **Insight Host** • 8 Intel/AMD x86-64 cores • At least 8 GB of RAMA dedicated 1 Gbit channel between Insight and Moab • 128 GB local SSD for swapAt least 512 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 Host** • 8 Intel/AMD x86-64 cores • At least 16 GB RAM • At least 512 GB dedicated disk space **Torque Host** • 8 Intel/AMD x86-64 cores • At least 16 GB RAM • At least 512 GB dedicated disk space **Insight Host** • 8 Intel/AMD x86-64 cores • At least 16 GB of RAM; a dedicated 1 Gbit channel between Insight and Moab • 128 GB local SSD for swap • At least 512 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 Host). The MAM Server may reside on its own host, on the Moab Host (preferred), or another server's host (except for the Insight Host). The Viewpoint Server may reside on its own host, on the Moab Host (preferred), or another server's host (except for the Insight Host). Databases may also reside one the same or a different host than its server component.

Please note the following:

- All requirements above (minimum and recommended) target a minimum number of management servers. Administrators are encouraged to separate the Torque Server and the Moab Server onto separate 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 Moab must send to Insight, Moab performs better without Insight enabled (for environments that do not require Viewpoint, or 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.

Identify the Manual or RPM Installation Methods

Adaptive Computing provides two methods for installing the Moab HPC Suite components, Manual Installation and RPM Installation.

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

Most components can be installed using either method. Please choose one method for each component.

Manual Installation

This method provides advantages for administrators who want non-standard configure options.

- This method has more supported operating systems than the RPM Installation method.

RPM Installation

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

- This method only supports CentOS, RHEL, or Scientific Linux operating systems.
- Whether you are installing RPMs on one host or on several hosts, each host must have the Adaptive Computing Package Repository enabled. See Preparing the Host for RPM Installations on page 78

Component Requirements

This topic provides the various software requirements and dependencies for the suite components (servers).

In this topic:

Torque

If you intend to use Torque 6.0.0 with Moab Workload Manager, you must run Moab version 9.0.0 or 8.0 or later. Torque 6.0.0 will not work with versions earlier than Moab 8.0.

In this section:

Supported Operating Systems

- CentOS 6.x, 7.x
- RHEL 6.x, 7.x
- Scientific Linux 6.x, 7.x
- SUSE Linux Enterprise Server 11, 12

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
- cpusets and cgroups
    - NUMA-awareness uses cgroups, which include cpusets. Red Hat systems must use libcgroup version 0.40.rc1-16.el6 or later; SUSE systems need to use a comparative libcgroup version.
    - cpusets: libhwloc 1.9.1 is the minimum supported, however NVIDIA K80 requires libhwloc 1.11.0. If you need to install libhwloc and the

corresponding hwloc-devel package, see Linux Cpuset Support in the *Torque Resource ManagerAdministrator Guide*.

If `--enable-cgroups` is specified, `--enable-cpuset` is ignored.

- if you build Torque from source (i.e. clone from github), 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

    Red Hat 6-based systems come packaged with 1.41.0 and Red Hat 7-based systems come packaged with 1.53.0. If needed, use the --with-boost-path=DIR option to change the packaged boost version. See Customizing the Install in the *Torque Resource Manager Adminstrator Guide*.

Moab Workload Manager

In this section:

Supported Operating Systems

- CentOS 6.x, 7.x
- RHEL 6.x, 7.x
- Scientific Linux 6.x, 7.x
- SUSE Linux Enterprise Server 11, 12;

If installing MWS on the same host, SLES 11 is not supported. If your configuratation includes Viewpoint, only Red Hat-based Oss are supported for the Moab Server and MWS Server shared host.

Software Requirements

- libcurl (http://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 8.1
- *(Optional)* MySQL, PostgreSQL, or Oracle with ODBC driver (see Database Configuration in the *Moab Workload Manager Administrator Guide* for details)

Supported Resource Managers

- Torque 5.1
- SLURM

Moab Web Services

MWS Server *must* reside same host as Moab Server (Moab Host). If using Viewpoint, the shared Moab Server and MWS Server must a Red-Hat based OS; regardless of whether Viewpoint is installed on that host.

In this topic:

Supported Operating Systems

- CentOS 6.x, 7.x
- RHEL 6.x, 7.x
- Scientific Linux 6.x, 7.x
- SUSE Linux Enterprise Server 12

Software Requirements

- Moab Workload Manager 9.0.0
- 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 OpenJDK/IcedTea, GNU Compiler for Java, and so on cannot run Moab Web Services.

- MongoDB® 2.4.x

Depends On (not necessarily on the same host)

- OpenLDAP *or* PAM; see Installing Moab Web Services on page 31 (Manual Installation) Installing Moab Web Services on page 87 (RPM Installation) for more details

Chapter 2 Manual Installation

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

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. Please note that the same commands will work for a non-root user with the `sudo` command.

Related Topics

-
- Preparing for Manual Installation on page 16
- Additional Configuration on page 43
- Upgrade on page 54

Installation

In this section:

Preparing for Manual Installation

The manual installation process of the Moab HPC Suite includes installing the separate components in the suite. This guide contains detailed instructions for installing each component.

Many individual components have dependencies on other components (see Chapter 1 Planning your Installation on page 3). However, if you do not require a certain component (Moab Web Services, for example), you do not have to install it.

The install instructions for each component include information about system requirements and dependencies. Some include prerequisite instructions that you will need to complete before you begin the install. Please 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. Please note that the same commands will work for a non-root user with the `sudo` command.

Set Up Proxies

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

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

Enable Extra Packages for the Repository

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

- Red Hat-based systems

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

> On RHEL systems, if your system is not registered to Red Hat Subscription Management, you will need to install the epel release rpm from the Fedora repo. The exact epel release version can vary over time, but the command should be similar to the following:
>
> RHEL 6
>
> ```
> [root]# rpm -Uvh http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm
> ```
>
> RHEL 7
>
> ```
> [root]# rpm -Uvh http://dl.fedoraproject.org/pub/epel/7/x86_64/e/epel-release-7-5.noarch.rpm
> ```

- SUSE-based systems

1. Verify that you have a licensed installation of SLES 11 or 12.
2. Download the SUSE Linux Enterprise 11 or 12 Software Development Kit e-Media Kit and add the ISO to the repository.
3. Add the `devel:languages:perl` and the `devel:languages:python` repos for your SLES version to the your software repositories. For example, if you are using SLES 12, you might use the following commands:

```
[root]# zypper addrepo
http://download.opensuse.org/repositories/devel:languages:perl/SLE_
12/devel:languages:perl.repo
[root]# zypper addrepo
http://download.opensuse.org/repositories/devel:languages:python/SLE_
12/devel:languages:python.repo
[root]# zypper refresh
```

Install the Moab HPC Suite Software Components

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

1. Install Torque. See Installing Torque Resource Manager on page 18.
2. Install Moab Workload Manager. See Installing Moab Workload Manager on page 25.
3. Install Moab Web Services. See Installing Moab Web Services on page 31.

Installing Torque Resource Manager

If you intend to use Torque Resource Manager 6.0.0 with Moab Workload Manager, you must run Moab version 8.0 or later. However, some Torque 6.0 functionality requires Moab 9.0 or later.

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

In this topic:

Requirements

In this section:

Supported Operating Systems

- CentOS 6.x, 7.x
- RHEL 6.x, 7.x
- Scientific Linux 6.x, 7.x
- SUSE Linux Enterprise Server 11, 12

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
- cpusets and cgroups
  - NUMA-awareness uses cgroups, which include cpusets. Red Hat systems must use libcgroup version 0.40.rc1-16.el6 or later; SUSE systems need to use a comparative libcgroup version.
  - cpusets: libhwloc 1.9.1 is the minimum supported, however NVIDIA K80 requires libhwloc 1.11.0. If you need to install libhwloc and the corresponding hwloc-devel package, see Linux Cpuset Support in the *Torque Resource Manager Administrator Guide*.

  If using `--enable-cgroups` is specified, `--enable-cpuset` is ignored.
- if you build Torque from source (i.e. clone from github), 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

    Red Hat 6-based systems come packaged with 1.41.0 and Red Hat 7-based systems come packaged with 1.53.0. If needed, use the --with-boost-path=DIR option to change the packaged boost version. See Customizing the Install in the *Torque Resource Manager Adminstrator Guide* for more information.

Prerequisites

In this section:

- Open Necessary Ports on page 19
- Verify the hostname on page 21

Open Necessary Ports

Torque requires certain ports to be open for essential communication.

- For client and pbs_mom communication to pbs_server, the default port is 15001.
- For pbs_server communication to pbs_mom, the default port is 15002.
- For pbs_mom communication to pbs_mom, the default port is 15003.

For more information on how to configure the ports that Torque uses for communication, see Configuring Ports in the *Torque Resource Manager Administrator Guide* for more information.

If you have a firewall enabled, do the following:

1. On the Torque Server Host:
   - Red Hat 6-based systems using iptables

```
[root]# iptables-save > /tmp/iptables.mod
[root]# vi /tmp/iptables.mod

# Add the following line immediately *before* the line matching
# "-A INPUT -j REJECT --reject-with icmp-host-prohibited"
-A INPUT -p tcp --dport 15001 -j ACCEPT

[root]# iptables-restore < /tmp/iptables.mod
[root]# service iptables save
```

   - Red Hat 7-based systems using firewalld

```
[root]# firewall-cmd --add-port=15001/tcp --permanent
[root]# firewall-cmd --reload
```

   - SUSE 11-based and SUSE 12-based systems using SuSEfirewall2

```
[root]# vi /etc/sysconfig/SuSEfirewall2

# Add the following port to the FW_SERVICES_EXT_TCP parameter
FW_SERVICES_EXT_TCP="15001"

[root]# service SuSEfirewall2_setup restart
```

2. On the Torque MOM Hosts (Compute Hosts):
   - Red Hat 6-based systems using iptables

```
[root]# iptables-save > /tmp/iptables.mod
[root]# vi /tmp/iptables.mod

# Add the following lines immediately *before* the line matching
# "-A INPUT -j REJECT --reject-with icmp-host-prohibited"
-A INPUT -p tcp --dport 15002 -j ACCEPT
-A INPUT -p tcp --dport 15003 -j ACCEPT

[root]# iptables-restore < /tmp/iptables.mod
[root]# service iptables save
```

   - Red Hat 7-based systems using firewalld

```
[root]# firewall-cmd --add-port=15002/tcp --permanent
[root]# firewall-cmd --add-port=15003/tcp --permanent
[root]# firewall-cmd --reload
```

- SUSE 11-based and SUSE 12-based systems using SuSEfirewall2

```
[root]# vi /etc/sysconfig/SuSEfirewall2

# Add the following ports to the FW_SERVICES_EXT_TCP parameter
FW_SERVICES_EXT_TCP="15002 15003"

[root]# service SuSEfirewall2_setup restart
```

Verify the hostname

On the Torque Server Host, confirm your host (with the correct IP address) is in your `/etc/hosts` file. To verify that the hostname resolves correctly, make sure that hostname and hostname -f report the correct name for the host.

Install Dependencies, Packages, or Clients

Install Packages

On the Torque Server Host, use the following commands to install the `libxml2-devel`, `openssl-devel`, and `boost-devel` packages.

- Red Hat 6-based and Red Hat 7-based systems

```
[root]# yum install libtool openssl-devel libxml2-devel boost-devel gcc gcc-c++
```

- SUSE 11-based and SUSE -12 based systems

```
[root]# zypper install libopenssl-devel libtool libxml2-devel boost-devel gcc
gcc-c++ make gmake
```

Install Torque Server

On the Torque Server Host, do the following:

1. Download the latest 6.0.0 build from the Adaptive Computing website. It can also be downloaded via command line (github method or the tarball distribution).
   - Clone the source from github.

If git is not installed:

```
# Red Hat-based systems
[root]# yum install git

# SUSE-based systems
[root]# zypper install git
```

```
[root]# git clone https://github.com/adaptivecomputing/torque.git -b 6.0.0 6.0.0
[root]# cd 6.0.0
[root]# ./autogen.sh
```

- Get the tarball source distribution.
  - Red Hat-based systems

```
[root]# yum install wget
[root]# wget http://www.adaptivecomputing.com/download/torque/torque-
6.0.0.tar.gz -O torque-6.0.0.tar.gz
[root]# tar -xzvf torque-6.0.0.tar.gz
[root]# cd torque-6.0.0/
```

  - SUSE-based systems

```
[root]# zypper install wget
[root]# wget http://www.adaptivecomputing.com/download/torque/torque-
6.0.0.tar.gz -O torque-6.0.0.tar.gz
[root]# tar -xzvf torque-6.0.0.tar.gz
[root]# cd torque-6.0.0/
```

2. Run each of the following commands in order.

```
[root]# ./configure
[root]# make
[root]# make install
```

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

3. Verify that the `/var/spool/torque/server_name` file exists and contains the correct name of the server.

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

4. Configure the `trqauthd` daemon to start automatically at system boot.
   - Red Hat 6-based systems

```
[root]# cp contrib/init.d/trqauthd /etc/init.d/
[root]# chkconfig --add trqauthd
[root]# echo /usr/local/lib > /etc/ld.so.conf.d/torque.conf
[root]# ldconfig
[root]# service trqauthd start
```

   - SUSE 11-based systems

```
[root]# cp contrib/init.d/suse.trqauthd /etc/init.d/trqauthd
[root]# chkconfig --add trqauthd
[root]# echo /usr/local/lib > /etc/ld.so.conf.d/torque.conf
[root]# ldconfig
[root]# service trqauthd start
```

- Red Hat 7-based and SUSE 12-based systems

```
[root]# cp contrib/systemd/trqauthd.service /usr/lib/systemd/system/
[root]# systemctl enable trqauthd.service
[root]# echo /usr/local/lib > /etc/ld.so.conf.d/torque.conf
[root]# ldconfig
[root]# systemctl start trqauthd.service
```

5. By default, Torque installs all binary files to `/usr/local/bin` and `/usr/local/sbin`. Make sure the path environment variable includes these directories for both the installation user and the root user.

```
[root]# export PATH=/usr/local/bin/:/usr/local/sbin/:$PATH
```

6. Initialize `serverdb` by executing the `torque.setup` script.

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

7. Add nodes to the `/var/spool/torque/server_priv/nodes` file. See Specifying Compute Nodes in the *Torque Resource Manager Adminstrator Guide* for information on syntax and options for specifying compute nodes.
8. Configure pbs_server to start automatically at system boot, and then start the daemon.
   - Red Hat 6-based systems

```
[root]# cp contrib/init.d/pbs_server /etc/init.d
[root]# chkconfig --add pbs_server
[root]# service pbs_server restart
```

   - SUSE 11-based systems

```
[root]# cp contrib/init.d/suse.pbs_server /etc/init.d/pbs_server
[root]# chkconfig --add pbs_server
[root]# service pbs_server restart
```

   - Red Hat 7-based and SUSE 12-based systems

```
[root]# qterm
[root]# cp contrib/systemd/pbs_server.service /usr/lib/systemd/system/
[root]# systemctl enable pbs_server.service
[root]# systemctl start pbs_server.service
```

Install Torque MOMs

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

See Specifying Compute Nodes or Configuring on Compute Nodes in the Torque Resource Manager Adminstrator Guide for more information.

Do the following:

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
Building ./torque-package-clients-linux-i686.sh ...
Building ./torque-package-mom-linux-i686.sh ...
Building ./torque-package-server-linux-i686.sh ...
Building ./torque-package-gui-linux-i686.sh ...
Building ./torque-package-devel-linux-i686.sh ...
Done.

The package files are self-extracting packages that can be copied and executed
on your production machines.  Use --help for options.
```

   b. Copy the self-extracting packages to each Torque MOM Host.

      Adaptive Computing recommends 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.

      The only required package for the compute node is mom-linux. Additional packages are recommended so you can use client commands and submit jobs from compute nodes.

```
[root]# scp torque-package-mom-linux-i686.sh <mom-node>:
[root]# scp torque-package-clients-linux-i686.sh <mom-node>:
```

   c. Copy the pbs_mom startup script to each Torque MOM Host.
      - Red Hat 6-based systems

```
[root]# scp contrib/init.d/pbs_mom <mom-node>:/etc/init.d
```

      - SUSE 11-based systems

```
[root]# scp contrib/init.d/suse.pbs_mom <mom-node>:/etc/init.d/pbs_mom
```

      - Red Hat 7-based and SUSE 12-based systems

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

2. On each Torque MOM Host, do the following:
   a. Install the self-extracting packages and run ldconfig.

```
[root]# ssh root@<mom-node>
[root]# ./torque-package-mom-linux-i686.sh --install
[root]# ./torque-package-clients-linux-i686.sh --install
[root]# ldconfig
```

   b. Configure pbs_mom to start at system boot, and then start the daemon.

- Red Hat 6-based and SUSE 11-based systems

```
[root]# chkconfig --add pbs_mom
[root]# service pbs_mom start
```

- Red Hat 7-based and SUSE-12 based systems

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

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 Adminstrator Guide* for more information.

Related Topics

Preparing for Manual Installation on page 16

Installing Moab Workload Manager

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

In this topic:

Open Necessary Ports

Moab uses a configurable server port (default 42559) for client-server communication. If you intend to run client commands on a host other than the Moab Host, or if you will be using Moab in a grid, and if you have a firewall enabled, you will need to configure the firewall to allow the server port.

On the Moab Server Host, do the following:

- Red Hat 6-based systems using iptables

```
[root]# iptables-save > /tmp/iptables.mod
[root]# vi /tmp/iptables.mod

# Add the following lines immediately *before* the line matching
# "-A INPUT -j REJECT --reject-with icmp-host-prohibited"

# Needed on the Moab server for off-host client communication
-A INPUT -p tcp --dport 42559 -j ACCEPT

[root]# iptables-restore < /tmp/iptables.mod
[root]# service iptables save
```

- Red Hat 7-based systems using firewalld

```
firewall-cmd --add-port=42559/tcp --permanent
firewall-cmd --reload
```

- SUSE 11-based and SUSE 12-based systems

```
[root]# vi /etc/sysconfig/SuSEfirewall2

# Add the following ports to the FW_SERVICES_EXT_TCP parameter as required

# Needed on the Moab server for off-host client communication
FW_SERVICES_EXT_TCP="42559"

[root]# service SuSEfirewall2_setup restart
```

Install Dependencies, Packages, or Clients

In this section:

- Dependencies and Packages on page 26
- Torque Client on page 26
- Installing Moab Workload Manager on page 25

Dependencies and Packages

On the Moab Server Host, use the following commands to install the required Moab dependencies and packages.

- Red Hat-based systems

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

- SUSE-based systems

```
[root]# zypper update
[root]# zypper install make curl libxml2-devel
```

Torque Client

If you are using Torque and are installing the Torque Server on a separate host (Torque Server Host) from the Moab Server (Moab Server Host), you will need

to install the Torque client on the Moab Server Host in order for Moab to interact with Torque.

Follow the instructions in Installing Torque Resource Manager on page 18 with these exceptions:

- Use the configure options --disable-server, --disable-mom and --disable-gui
- Omit the step initializing the serverdb and all of the steps thereafter

(Optional) Build a Custom RPM

If you want to build a custom RPM, do the following:

1. Install `rpm-build`.

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

2. Download the latest Moab build (`moab-<version>-<OS>.tar.gz`) from the Adaptive Computing website.

> The variable marked <*version*> is the desired version of the suite; for example, 8.1.0-2015010514-ed0c40a would be Moab 8.1.0 revision 2015010514 at changeset ed0c40a. The variable marked <*OS*> indicates the OS for which the build was designed.

3. Untar the downloaded package.
4. Change directories into the untarred directory.
5. Edit the ./moab.spec file for RPM customization.
6. Run ./rpm-build.
7. Locate the custom RPM in rpm/RPMS/x86_64.

Install Moab Server

On the Moab Server Host, do the following:

1. Download the latest Moab build (`moab-<version>-<OS>.tar.gz`) from the Adaptive Computing website.

> The variable marked <*version*> indicates the build's version, revision, and changeset information. The variable marked <*OS*> indicates the OS for which the build was designed.

2. As the root user, run each of the following commands in order.

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

3. Configure Moab.

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

See Moab Workload Manager Configuration Options on page 50 for a list of commonly used options or use `./configure --help` for a complete list of available options.

4. *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. It is recommended 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
```

5. Install Moab.

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

6. Modify the Moab configuration file.

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

   Do the following:

   a. 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
```

      If you use a SLURM resource manager, see Moab-SLURM Integration Guide in the *Moab Workload Manager Administrator Guide* for configuration information. If you use a NATIVE resource manager, see Managing Resources Directly with the Native Interface in the *Moab Workload Manager Administrator Guide* for configuration information.

   b. If you are using Torque as a resource manager and you installed the Torque Server on a separate 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>
```

7. Source the appropriate profile script to add the Moab executable directories to your current shell *$PATH* environment.

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

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

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

   To verify the current status of your license, use `moab --about`.

   Moab checks the status of the license every day just after midnight. At 60 and 45 days before, and daily from 30 days before license expiration to and including the license expiration date, Moab sends an e-mail to all level 1 administrators informing them of the pending Moab license expiration. A log record is also made of the upcoming expiration event. For the notifications to occur correctly, you must enable administrator email notification (see Notifying Administrators of Failures in the *Moab Workload Manager Administrator Guide*) and `moab.cfg` must contain email addresses for level 1 administrators. For example:

```
ADMINCFG[1] USERS=u1,u2,u3[,...]

USERCFG[u1] EMAILADDRESS=u1@company.com
USERCFG[u2] EMAILADDRESS=u2@company.com
USERCFG[u3] EMAILADDRESS=u3@company.com

MAILPROGRAM DEFAULT
```

   Moab will not run without a license. For information about obtaining a trial license, please contact Adaptive Computing.

9. Start Moab.

   - Red Hat 6-based or SUSE 11-based systems

```
[root]# chkconfig moab on
[root]# service moab start
```

   - Red Hat 7-based or SUSE 12-based systems

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

Configure Torque to Trust Moab

If you are using Torque as a resource manager and you installed the Torque Server on a separate host (Torque Server Host); recommended, 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
```

Verify the Installation

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

- 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
```

(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.

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. See Mauth Authentication in the *Moab Workload Manager Administrator Guide* for more information.

Do the following:

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. On the Moab Client Host, run the tarball to install the Moab client commands.

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

Related Topics

Preparing for Manual Installation on page 16

Installing Moab Web Services

You must deploy Moab Web Services on the *same* host as Moab Server (Moab Server Host). If using Viewpoint, this shared host must have a Red Hat-based OS; regardless of whether Viewpoint is also installed on that host. For documentation clarity, these instructions refer to the shared host for Moab Server and MWS as the MWS Host.

This topic contains instructions on how to install Moab Web Services (MWS).

In this topic:

Open Necessary Ports

If your site is running firewall software on its hosts, you will need to configure the firewall to allow connections to the necessary ports.

In this section:

Open the Tomcat Port (8080)

On the MWS Server Host, do the following:

- Red Hat 6-based systems using iptables

```
[root]# iptables-save > /tmp/iptables.mod
[root]# vi /tmp/iptables.mod

# Add the following lines immediately *before* the line matching
# "-A INPUT -j REJECT --reject-with icmp-host-prohibited"

-A INPUT -p tcp --dport 8080 -j ACCEPT

[root]# iptables-restore < /tmp/iptables.mod
[root]# service iptables save
```

- Red Hat 7-based systems using firewalld

```
[root]# firewall-cmd --add-port=8080/tcp --permanent
[root]# firewall-cmd --reload
```

- SUSE 12-based systems using SuSEfirewall2

```
[root]# vi /etc/sysconfig/SuSEfirewall2

FW_SERVICES_EXT_TCP="8080"

[root]# service SuSEfirewall2_setup restart
```

Open the MWS MongoDB Database Port (27017)

Depending on your system configuration, your MongoDB databases may not be installed on the same host as their corresponding component servers. For example, you may choose to install the MWS MongoDB database on the same host where you have installed other MongoDB databases instead of on the MWS Server Host.

Do the following, as needed:

- If you have chosen to install the MWS MongoDB database on the *same* host you installed other MongoDB databases (for example, the same host you installed the Moab MongoDB database), confirm the firewall port (27017) is already opened on that host.
- If you have chosen to install the MWS MongoDB database on a *separate* host from other MongoDB databases, you will need to open the MWS MongoDB database port in firewall for that host. To open the port in the firewall, do the following:
  - Red Hat 6-based systems using iptables

```
[root]# iptables-save > /tmp/iptables.mod
[root]# vi /tmp/iptables.mod

Add the following line immediately *before* the line matching
# "-A INPUT -j REJECT --reject-with icmp-host-prohibited"

-A INPUT -p tcp --dport 27017 -j ACCEPT

[root]# iptables-restore < /tmp/iptables.mod
[root]# service iptables save
```

  - Red Hat 7-based systems using firewalld

```
[root]# firewall-cmd --add-port=27017/tcp --permanent
[root]# firewall-cmd --reload
```

  - SUSE 11-based or SUSE 12-based systems using SuSEfirewall2 (SUSE 11 is not supported on the MWS Server Host)

```
[root]# vi /etc/sysconfig/SuSEfirewall2

FW_SERVICES_EXT_TCP="27017"

[root]# service SuSEfirewall2_setup restart
```

Install Dependencies, Packages, or Clients

In this section:

Install Java

Install the Linux x64 RPM version of 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 OpenJDK/IcedTea, GNU Compiler for Java, and so on cannot run MWS.

On the MWS Host, do the following:

1. Install the Linux x64 RPM version of Oracle Java SE 8 JRE.
   a. Go to the to the Oracle Java download page (http://java.com/en/download/linux_manual.jsp).
   b. Copy the URL for the Linux x64 RPM version, and run the following command:
      - Red Hat 6-based and Red Hat 7-based systems

```
[root]# rpm -Uh <URL>
```

      - SUSE 12-based systems

```
[root]# ln -s /usr/sbin/update-alternatives /usr/sbin/alternatives
[root]# rpm -Uh <URL>
```

Install Tomcat

On the MWS Host, do the following:

- Red Hat 6-based and Red Hat 7-based systems

```
[root]# yum install tomcat
```

- SUSE 12-based systems

```
[root]# zypper install tomcat
```

Install MongoDB

To install and enable MongoDB, on the MWS Host, do the following:

1. Install MongoDB.
   - Red Hat 6-based and Red Hat 7-based systems

```
[root]# cat > /etc/yum.repos.d/mongodb.repo <<End-of-file
[mongodb]
name=MongoDB Repository
baseurl=http://downloads-distro.mongodb.org/repo/redhat/os/x86_64
gpgcheck=0
enabled=1
exclude=mongodb-org mongodb-org-server
End-of-file
[root]# yum install mongo-10gen-server
```

   - SUSE-12 based systems

```
[root]# zypper ar --refresh -r
http://download.opensuse.org/repositories/server:/database/SLE_
12/server:database.repo
[root]# zypper install mongodb
```

2. Start MongoDB.

> There may be a short delay (approximately three minutes) for Mongo to start the first time.

   - Red Hat 6-based systems

```
[root]# chkconfig mongod on
[root]# service mongod start
```

   - Red Hat 7-based systems

```
[root]# cat > /usr/lib/systemd/system/mongodb.service <<End-of-file
[Unit]
Description=High-performance, schema-free document-oriented database
After=syslog.target network.target

[Service]
Type=forking
User=mongod
Group=mongod
Environment=CONFIG=/etc/mongod.conf
Environment=OPTIONS=
EnvironmentFile=-/etc/sysconfig/mongod
ExecStart=/usr/bin/mongod -f \$CONFIG \$OPTIONS
PrivateTmp=true
LimitNOFILE=65536
TimeoutStartSec=180
StandardOutput=syslog
StandardError=syslog

[Install]
WantedBy=multi-user.target
End-of-file
[root]# rm -f /etc/init.d/mongod
[root]# systemctl enable mongodb.service
[root]# systemctl start mongodb.service
```

- SUSE 12-based systems

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

3. Prepare the MongoDB database by doing the following:
   a. 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.addUser("admin_user", "secret1");
> db.auth ("admin_user", "secret1");

> use moab;
> db.addUser("moab_user", "secret2");
> db.addUser("mws_user", "secret3", true);

> use mws;
> db.addUser("mws_user", "secret3");
> 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 (at http://docs.mongodb.org/manual/tutorial/control-access-to-mongodb-with-authentication) for more information.

   d. Enable authentication in MongoDB.
      - Red Hat 6-based systems

```
[root]# vi /etc/mongod.conf
auth = true
[root]# service mongod restart
```

      - Red Hat 7-based systems

```
[root]# vi /etc/mongod.conf
auth = true
[root]# systemctl restart mongodb.service
```

      - SUSE 12-based systems

        MongoDB authentication is enabled (`auth = true`) by default. No further action is needed.

Install MWS Server

On the MWS Host, do the following:

1. Verify Moab Server is installed and configured as desired (for details, see Installing Moab Workload Manager on page 25).
2. Start Moab.
   - Red Hat 6-based systems
     ```
     [root]# service moab start
     ```
   - Red Hat 7-based and SUSE 12-based systems
     ```
     [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.

   Here is a sample script for this setup:

   ```
   [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 # Depending on your OS, the Tomcat username
   might be tomcat6.
   [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 build (`mws-<version>.tar.gz`) from the Adaptive Computing website.

   > The variable marked <*version*> is the desired version of the suite; for example, 9.0.0.

5. Extract the contents of the MWS download tarball into a temporary directory. For example:

   ```
   [root]# mkdir /tmp/mws-install
   [root]# cd /tmp/mws-install
   [root]# tar xvzf $HOME/Downloads/mws-9.0.0.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.

```
[root]# cd /tmp/mws-install/mws-9.0.0/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.

   ```
   # Listen to local interface only. Comment out to listen on all interfaces.
   #bind_ip=127.0.0.1
   ```

   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 on page 33.

   ```
   MONGOUSER     moab_user
   MONGOPASSWORD secret2
   ```

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

   - Red Hat 6-based systems

   ```
   [root]# service moab restart
   [root]# mdiag -S

   Mongo connection (localhost) is up (credentials are set)
   ```

   - Red Hat 7-based and SUSE 12-based systems

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

   Mongo connection (localhost) is up (credentials are set)
   ```

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.

- Red Hat 6-based systems

```
[root]# service moab stop
[root]# dd if=/dev/urandom count=18 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]# service moab start
```

- Red Hat 7-based and SUSE 12-based systems

```
[root]# systemctl stop moab.service
[root]# dd if=/dev/urandom count=18 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`

- Red Hat 6-based systems

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

- Red Hat 7-based and SUSE 12-based systems

```
[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 messsages 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.

```
[root]# cd /tmp/mws-install/mws-9.0.0
[root]# cp mws-config.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/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 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

// Replace <ENTER-KEY-HERE> with the contents of /opt/moab/etc/.moab.key.

moab.secretKey = "<ENTER-KEY-HERE>"
moab.server = "localhost"
moab.port = 42559

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

moab.messageQueue.secretKey = "<ENTER-KEY-HERE>"

// Change these to be whatever you like.

auth.defaultUser.username = "moab-admin"
auth.defaultUser.password = "changeme!"
```

> If you do not change **auth.defaultUser.password**, your MWS will not be secure (because anyone reading these instructions would be able to log into your MWS). Here are some tips (http://www.us-cert.gov/cas/tips/ST04-002.html) for choosing a good password.

c. Do *one* of the following:

> You can configure only one authentication method in `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 an MWS connection to your LDAP server, add the following parameters to the `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.

> If you followed the Adaptive Computing tutorial, Setting Up OpenLDAP on CentOS 6, your **ldap.directory.type** should be set to "OpenLDAP Using InetOrgPerson Schema." However, the use of other schemas is supported. For more information see LDAP Configuration Using mws-config.groovy.

> To see how to configure a secure connection to the LDAP server, see Securing the LDAP Connection.

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

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

> For more information about PAM configuration with MWS, see PAM (Pluggable Authentication Module) Configuration Using mws-config.groovy.

> There is a security risk when authenticating local users through your PAM configuration. This behavior is highly discouraged and not supported by Adaptive Computing.

d. Add the **grails.mongo.username** and **grails.mongo.password** parameters to the `mws-config.groovy` file. Use the MWS credentials you added to MongoDB in the Preparing for Manual Installation section.

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

e. Make the MWS configuration files read-only.

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

10. Configure Tomcat

    Add the following lines to the end of `/etc/tomcat/tomcat.conf`.

```
CATALINA_OPTS="-DMWS_HOME=/opt/mws -Xms256m -Xmx3g -XX:MaxPermSize=384m -
Dfile.encoding=UTF8"
JAVA_HOME="/usr/java/latest"
```

> MaxPermSize is ignored using Java 8; and therefore can be omitted.

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

    - Red Hat 6-based systems

```
[root]# chkconfig tomcat on
[root]# service tomcat stop
[root]# cp /tmp/mws-install/mws-9.0.0/mws.war /usr/share/tomcat/webapps
[root]# service tomcat start
```

    - Red Hat 7-based and SUSE 12-based systems

```
[root]# systemctl enable tomcat.service
[root]# systemctl stop tomcat.service
[root]# cp /tmp/mws-install/mws-9.0.0/mws.war /usr/share/tomcat/webapps
[root]# systemctl start tomcat.service
```

12. Navigate to `http://<server>:8080/mws/` in a web 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 Moab Web Services Issues on page 122.

Related Topics

Preparing for Manual Installation on page 16

Additional Configuration

In this section:

Configuring SSL in Tomcat

To configure SSL in Tomcat, please refer to the Apache Tomcat documentation (http://tomcat.apache.org/tomcat-7.0-doc/ssl-howto.html).

Setting Up OpenLDAP on CentOS 6

These instructions are intended to help first-time LDAP administrators get up and running. The following procedures contain instructions for getting started using OpenLDAP on a CentOS 6 system. For more complete information on how to set up OpenLDAP see the OpenLDAP documentation (http://www.openldap.org/doc/admin24/).

In this topic:

Adaptive Computing is not responsible for creating, maintaining, or supporting customer LDAP or Active Directory configurations.

Installing and Configuring OpenLDAP on Centos 6

First, you will need to install OpenLDAP. These instructions explain how you can do this on a CentOS 6 system.

To install and configure OpenLDAP on Centos 6

1. Run the following command:

```
[root]# yum -y install openldap openldap-clients openldap-servers
```

2. Generate a password hash to be used as the admin password. This password hash will be used when you create the root user for your LDAP installation. For example:

```
[root]# slappasswd
New password : p@ssw0rd
Re-enter new password : p@ssw0rd
{SSHA}5lPFVw19zeh7LT53hQH69znzj8TuBrLv
```

3. Add the root user and the root user's password hash to the OpenLDAP configuration in the `olcDatabase={2}bdb.ldif` file. The root user will have permissions to add other users, groups, organizational units, etc. Do the following:

   a. Run this command:

```
[root]# cd /etc/openldap/slapd.d/cn\=config
[root]# vi olcDatabase\=\{2\}bdb.ldif
```

   b. If the **olcRootPW** attribute does not already exist, create it. Then set the value to be the hash you created from `slappasswd`. For example:

```
olcRootPW: {SSHA}5lPFVw19zeh7LT53hQH69znzj8TuBrLv
...
```

4. While editing this file, change the distinguished name (DN) of the **olcSuffix** to something appropriate. The suffix typically corresponds to your DNS domain name, and it will be appended to the DN of every other LDAP entry in your LDAP tree.

   For example, let's say your company is called Acme Corporation, and that your domain name is "acme.com". You might make the following changes to the `olcDatabase={2}bdb.ldif` file:

```
olcSuffix: dc=acme,dc=com
...
olcRootDN: cn=Manager,dc=acme,dc=com
...
olcRootPW: {SSHA}5lPFVw19zeh7LT53hQH69znzj8TuBrLv
...
```

Do not set the cn of your root user to "root" (`cn=root,dc=acme,dc=com`), or OpenLDAP will have problems.

> Throughout the following examples in this topic, you will see `dc=acme,dc=com`. "acme" is only used as an example to illustrate what you would use as your own domain controller if your domain name was "acme.com". You should replace any references to "acme" with your own organization's domain name.

5. Modify the DN of the root user in the `olcDatabase={1}monitor.ldif` file to match the **olcRootDN** line in the `olcDatabase={2}bdb.ldif` file. Do the following:
   a. Run this command to edit the `olcDatabase={2}bdb.ldif` file:

```
[root]# vi olcDatabase\=\{1\}monitor.ldif
```

   b. Modify the **olcAccess** line so that the **dn.base** matches the **olcRootDN** from the `olcDatabase={2}bdb.ldif` file. (In this example, **dn.base** should be "cn=Manager,dc=acme,dc=com".)

```
olcAccess: {0}to * by
dn.base="gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth" read by
dn.base="cn=Manager,dc=acme,dc=com" read by * none
```

   c. Now the root user for your LDAP is `cn=Manager,dc=acme,dc=com`. The root user's password is the password that you entered using `slappasswd` earlier in this procedure, which, in this example, is **p@ssw0rd**
6. Hide the password hashes from users who should not have permission to view them.

> A full discussion on configuring access control in OpenLDAP is beyond the scope of this tutorial. For help, see the OpenLDAP Access Control documentation (http://www.openldap.org/doc/admin24/access-control.html).

   a. Run this command to edit the `oclDatabase\=\{2\}bdb.ldif` file:

```
[root]# vi olcDatabase\=\{2\}bdb.ldif
```

   b. Add the following two lines to the end of the file to restrict users from viewing other users' password hashes.

```
olcAccess: {0}to attrs=userPassword by self write by
dn.base="cn=Manager,dc=acme,dc=com" write by anonymous auth by * none
olcAccess: {1}to * by dn.base="cn=Manager,dc=acme,dc=com" write by self write by
* read
```

These lines allow a user to read and write his or her own password. It also allows a manager to read and write anyone's password. Anyone, including anonymous users, is allowed to view non-password attributes of other users.

7. Make sure that OpenLDAP is configured to start when the machine starts up, and start the OpenLDAP service.

```
[root]# chkconfig slapd on
[root]# service slapd start
```

8. Now, you must manually create the "dc=acme,dc=com" LDAP entry in your LDAP tree.

   An LDAP directory is analogous to a tree. Nodes in this tree are called LDAP "entries" and may represent users, groups, organizational units, domain controllers, or other objects. The attributes in each entry are determined by the LDAP schema. In this tutorial we will build entries based on the InetOrgPerson schema (which ships with OpenLDAP by default).

   In order to build our LDAP tree we must first create the root entry. Root entries are usually a special type of entry called a domain controller (DC). Because we are assuming that the organization is called Acme Corporation, and that the domain is "acme.com," we will create a domain controller LDAP entry called `dc=acme,dc=com`. Again, you will need to replace "acme" with your organization's domain name. Also note that `dc=acme,dc=com` is what is called an LDAP distinguished name (DN). An LDAP distinguished name uniquely identifies an LDAP entry.

   Do the following:

   a. Create a file called `acme.ldif`. (You can delete this file once its content has been added to LDAP, so in this example, we will create it in the `/tmp` folder.)

      ```
      [root]# cd /tmp
      [root]# vi acme.ldif
      ```

   b. Add the following lines in `acme.ldif`:

      ```
      dn: dc=acme,dc=com
      objectClass: dcObject
      objectClass: organization
      dc: acme
      o : acme
      ```

   c. Now add the contents of this file to LDAP. Run this command:

      ```
      [root]# ldapadd -f acme.ldif -D cn=Manager,dc=acme,dc=com -w p@ssw0rd
      ```

   d. Verify that your entry was added correctly.

      ```
      [root]# ldapsearch -x -LLL -b dc=acme,dc=com
      dn: dc=acme,dc=com
      objectClass: dcObject
      objectClass: organization
      dc: acme
      o: acme
      ```

9. Run the following:

```
[root]# sudo iptables -L
[root]# sudo service iptables save
```

10. By default, the CentOS 6 firewall will block external requests to OpenLDAP. In order to allow MWS to access LDAP, you will have to configure your firewall to allow connections on port 389. (Port 389 is the default LDAP port.)

    Configuring your firewall is beyond the scope of this tutorial; however, it may be helpful to know that the default firewall on CentOS is a service called `iptables`. For more information, see the documentation on iptables (http://wiki.centos.org/HowTos/Network/IPTables). In the most basic case, you may be able to add a rule to your firewall that accepts connections to port 389 by doing the following:

    a. Edit your `iptables` file:

```
[root]# vi /etc/sysconfig/iptables
```

    b. Add the following line *after* all the **ACCEPT** lines but *before* any of the **REJECT** lines in your `iptables` file:

```
# ... lines with ACCEPT should be above
-A INPUT -p tcp --dport 389 -j ACCEPT
# .. lines with REJECT should be below
```

    For example, here is a sample `iptables` file with this line added:

```
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
-A INPUT -p tcp --dport 389 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited

COMMIT
```

    c. Now reload `iptables`.

```
[root]# service iptables reload
```

> Although providing instructions is beyond the scope of this tutorial, it is also highly recommended that you set up OpenLDAP to use SSL or TLS security to prevent passwords and other sensitive data from being sent in plain text. For information on how to do this, see the OpenLDAP TLS documentation (http://www.openldap.org/doc/admin24/tls.html).

Now that you have installed and set up Open LDAP, you are ready to add organizational units. See Adding an Organizational Unit (OU) on page 48.

Adding an Organizational Unit (OU)

These instructions will describe how to populate the LDAP tree with organizational units (OUs), groups, and users, all of which are different types of LDAP entries. The examples that follow also presume an InetOrgPerson schema, because the InetOrgPerson schema is delivered with OpenLDAP by default.

To add an organizational unit (OU) entry to the LDAP tree

In this example, we are going to add an OU called "Users".

1. Create a temporary file called `users.ldif`. (You can delete this file once its content has been added to LDAP, so in this example, we will create it in the `/tmp` folder.)

```
[root]# cd /tmp
[root]# vi users.ldif
```

2. Add these lines to `users.ldif`:

```
dn: ou=Users,dc=acme,dc=com
objectClass: organizationalUnit
ou: Users
```

3. Add the contents of `users.ldif` file to LDAP.

```
[root]# ldapadd -f users.ldif -D cn=Manager,dc=acme,dc=com -w p@ssw0rd
```

Adding a User

To add a user to LDAP

In this example, we will add a user named "Bob Jones" to LDAP inside the "Users" OU.

1. Create a temporary file called `bob.ldif`. (You can delete this file once its content has been added to LDAP, so in this example, we will create it in the `/tmp` folder.)

```
[root]# cd /tmp
[root]# vi bob.ldif
```

2. Add these lines to `bob.ldif`:

```
dn: cn=Bob Jones,ou=Users,dc=acme,dc=com
cn: Bob Jones
sn: Jones
objectClass: inetOrgPerson
userPassword: p@ssw0rd
uid: bjones
```

3. Add the contents of `bob.ldif` file to LDAP.

```
[root]# ldapadd -f bob.ldif -D cn=Manager,dc=acme,dc=com -w p@ssw0rd
```

Adding a Group

To add a group to LDAP

In this example, we will add a group called "Engineering" to LDAP inside the "Users" OU.

1. Create a temporary file called `engineering.ldif`. (You can delete this file once its content has been added to LDAP, so in this example, we will create it in the `/tmp` folder.)

```
[root]# cd /tmp
[root]# vi engineering.ldif
```

2. Add these lines to `engineering.ldif`:

```
dn: cn=Engineering,ou=Users,dc=acme,dc=com
cn: Engineering
objectClass: groupOfNames
member: cn=Bob Jones,ou=Users,dc=acme,dc=com
```

3. Add the contents of `engineering.ldif` file to LDAP.

```
[root]# ldapadd -f engineering.ldif -D cn=Manager,dc=acme,dc=com -w p@ssw0rd
```

Adding a User to a Group

To add a user to an LDAP group

In this example, we will add an LDAP member named "Al Smith" to the "Engineering" LDAP group. This example assumes that user, Al Smith, has already been added to LDAP.

> Before you add a user to an LDAP group, the user must first be added to LDAP. For more information, see Adding a User on page 48.

1. Create a temporary file called `addUserToGroup.ldif`. (You can delete this file once its content has been added to LDAP, so in this example, we will create it in the `/tmp` folder.)

```
[root]# cd /tmp
[root]# vi addUserToGroup.ldif
```

2. Add these lines to `addUserToGroup.ldif`:

```
dn: cn=Engineering,ou=Users,dc=acme,dc=com
changetype: modify
add: member
member: cn=Al Smith,ou=Users,dc=acme,dc=com
```

3. Now add the contents of `addUserToGroup.ldif` file to LDAP.

```
[root]# ldapadd -f addUserToGroup.ldif -D cn=Manager,dc=acme,dc=com -w p@ssw0rd
```

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.

Option	Description	Example
--prefix	Specifies the location of the binaries and libraries of the Moab install. The default location is `/opt/moab`.	`[root]# ./configure --prefix=/usr/local`
--with-am	Specifies that you want to configure Moab with Moab Accounting Manager. There is a similar `--with-torque` option that configures Moab with Torque, but you do not need to specify this option if you install the "torque" tarball version.	`[root]# ./configure --with-am`
--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 Interfacing to FLEXlm section in the Moab Administrator Guide.	`[root]# ./configure --with-flexlm`

Option	Description	Example
--with-homedir	Specifies the location of the Moab configuration directory and the MOABHOMEDIR environment variable. The default location is `/opt/moab`. MOABHOMEDIR is automatically set on some distributions during installation, when the --with-profile option is enabled.	`[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 your distribution. The installed file (`/etc/init.d/moab` or `/usr/lib/systemd/system/moab.service`) may be customized to your needs. If you do not want this file to be installed, use this option to exclude it.	`[root]# ./configure --without-init`
--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}`) may 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`

Trusting Servers in Java

In this topic:

Prerequisites

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

```
[root]# source /etc/tomcat6/tomcat6.conf
```

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

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 ldaps 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.

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 command:

```
[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've entered the keystore password, you'll 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.

Upgrade

In this section:

Preparing for Upgrade

The upgrade process of the Moab HPC Suite includes upgrading the database and separate components in the suite. This guide contains detailed instructions for upgrading each component.

It is highly recommended 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 Professional Services 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. Please note that the same commands will work for a non-root user with the `sudo` command.

Upgrade the Moab HPC Suite in the following order:

1. Mongo database. See Upgrading MongoDB
2. Torque. See Upgrading Torque
3. Moab Workload Manager. See Upgrading Moab Workload Manager
4. Moab Web Services. See Upgrading Moab Web Services

Related Topics

- Chapter 1 Planning your Installation on page 3

Upgrading Torque

Torque 6.0.0 binaries are backward compatible with Torque 5.0 or later. However they are not backward compatible with Torque versions prior to 5.0. When you upgrade to Torque 6.0.0 from versions prior to 5.0, all MOM and server daemons must be upgraded at the same time.

The job format is compatible between 6.0.0 and previous versions of Torque and any queued jobs will upgrade to the new version. It is not recommended to upgrade 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.

In this topic:

Before You Upgrade

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

In this section:

serverdb

The pbs_server configuration is saved in the file *TORQUE_HOME*/server_priv/serverdb. When running Torque 4.1 or later for the first time, this file converts from a binary file to an XML-like format.

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 Adminstrator Guide* for more information.

Cray Systems

For upgrading Torque to 6.0.0 on a Cray system, refer to the Installation Notes for Moab and Torque for Cray in Appendix G of the *Moab Workload Manager Administrator Guide*.

Stop Torque Services

Do the following:

1. On the Torque Server Host, shut down the Torque server.
   - Red Hat 6-based or SUSE 11-based systems

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

   - Red Hat 7-based or SUSE 12-based systems

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

2. On each Torque MOM 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.

   - Red Hat 6-based or SUSE 11-based systems

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

   - Red Hat 7-based or SUSE 12-based systems

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

3. On each Torque Client Host (including the Moab Server Host, the Torque Server Host, and the Torque MOM Hosts, if applicable), shut down the trqauthd service.
   - Red Hat 6-based of SUSE 11-based systems

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

- Red Hat 7-based or SUSE 12-based systems

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

Upgrade the Torque Server

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. If not already installed, install the Boost C++ headers.
   - Red Hat-based systems

```
[root]# yum install boost-devel
```

   - SUSE-based systems

```
[root]# zypper install boost-devel
```

3. Download the latest 6.0.0 build from the Adaptive Computing website.
4. Install the latest Torque tarball.

```
[root]# cd /tmp
[root]# tar xzvf torque-<version>-<build number>.tar.gz
[root]# cd torque-<version>-<build number>
[root]# ./configure
[root]# make
[root]# make install
```

5. Update the pbs_server service startup script.
   a. Make a backup of your current service startup script.
      - Red Hat 6-based and SUSE 11-based systems

```
[root]# cp /etc/init.d/pbs_server pbs_server.bak
```

      - Red Hat 7-based and SUSE 12-based systems

```
[root]# cp /usr/lib/systemd/system/pbs_server.service pbs_server.service.bak
```

   b. Copy in the new stock service startup script.
      - Red Hat 6-based systems

```
[root]# cp contrib/init.d/pbs_server /etc/init.d
```

      - SUSE 11-based systems

```
[root]# cp contrib/init.d/suse.pbs_server /etc/init.d/pbs_server
```

- Red Hat 7-based and SUSE 12-based systems

```
[root]# cp contrib/systemd/pbs_server.service /usr/lib/systemd/system/
```

c. Merge in any customizations.

- Red Hat 6-based and SUSE 11-based systems

```
[root]# vi /etc/init.d/pbs_server
```

- Red Hat 7-based and SUSE 12-based systems

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

Update the Torque MOMs

Do the following:

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
Building ./torque-package-clients-linux-i686.sh ...
Building ./torque-package-mom-linux-i686.sh ...
Building ./torque-package-server-linux-i686.sh ...
Building ./torque-package-gui-linux-i686.sh ...
Building ./torque-package-devel-linux-i686.sh ...
Done.

The package files are self-extracting packages that can be copied and executed
on your production machines.  Use --help for options.
```

b. Copy the self-extracting packages to each Torque MOM Host.

Adaptive Computing recommends 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-i686.sh <torque-mom-host>:
```

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

- Red Hat 6-based systems

```
[root]# scp contrib/init.d/pbs_mom <torque-mom-host>:/etc/init.d
```

- SUSE 11-based systems

```
[root]# scp contrib/init.d/suse.pbs_mom <torque-mom-host>:/etc/init.d/pbs_mom
```

- Red Hat 7-based and SUSE 12-based systems

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

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

> Many of these steps 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 MOM Host.

```
[root]# ./torque-package-mom-linux-x86_64.sh --install
[root]# echo /usr/local/lib > /etc/ld.so.conf.d/torque.conf
[root]# ldconfig
```

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 packages to each Torque Client Host.

      Adaptive Computing recommends 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-i686.sh <torque-client-host>:
      ```

   b. Copy the trqauthd startup script to each Torque Client Host.
      - Red Hat 6-based systems

        ```
        [root]# scp contrib/init.d/trqauthd <torque-client-host>:/etc/init.d
        ```

      - SUSE 11-based systems

        ```
        [root]# scp contrib/init.d/suse.trqauthd <torque-client-host>:/etc/init.d/trqauthd
        ```

      - Red Hat 7-based and SUSE 12-based systems

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

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

> Many of these steps 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
[root]# echo /usr/local/lib > /etc/ld.so.conf.d/torque.conf
[root]# ldconfig
```

Start Torque Services

Do the following:

1. On each Torque Client Host (including the Moab Server Host, Torque Server Host and Torque MOM Hosts, if applicable), start up the trqauthd service.
   - Red Hat 6-based of SUSE 11-based systems

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

   - Red Hat 7-based or SUSE 12-based systems

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

2. On each Torque MOM Host, start up the Torque MOM service.
   - Red Hat 6-based or SUSE 11-based systems

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

   - Red Hat 7-based or SUSE 12-based systems

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

3. On the Torque Server Host, start up the Torque server.
   - Red Hat 6-based or SUSE 11-based systems

     ```
     [root]# service pbs_server start
     ```

   - Red Hat 7-based or SUSE 12-based systems

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

Perform Status and Error Checks

On the Torque Server Host, do the following:

1. Check the status of the jobs in the queue.

   ```
   [root]# qstat
   ```

2. Check for errors.

   ```
   [root]# grep -i error /var/spool/torque/server_logs/*
   [root]# grep -i error /var/spool/torque/mom_logs/*
   ```

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.

It is highly recommended that you *first* perform upgrades in a *test environment*. See the warning in Preparing for Upgrade on page 54. It is 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*.

If you are also upgrading Torque from an older version (pre-4.0), contact Adaptive Computing.

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. Please note that the same commands will work for a non-root user with the `sudo` command.

Upgrade Moab Workload Manager

On the Moab Server Host, do the following:

1. If you have not already done so, install extra packages from the add-on repositories. See Enable Extra Packages for the Repository on page 17
2. Download the latest Moab build (`moab-<version>-<OS>.tar.gz`) from the Adaptive Computing website.

   The variable marked <*version*> indicates the build's version, revision, and changeset information. The variable marked <*OS*> indicates the OS for which the build was designed.

3. Untar the distribution file. For example:

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

4. Change directory into the extracted directory.

5. 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`.

6. Stop Moab.

```
[root]# mschedctl -k
moab will be shutdown immediately
```

   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.

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

8. 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.

   CPAN is installed on SUSE-based systems by default. If upgrading on a Red Hat-based system, you will need to install CPAN [root]# yum install perl-CPAN if you have not already done so. When first running CPAN, you will be asked for configuration information. It is recommended that you choose an automatic configuration.

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

9. 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.

10. If you use ODBC, you must upgrade to the 9.0.0 schema. See Migrating Your Database to Newer Versions of Moab for more information.

11. Verify the version number is correct before starting the new server version.

```
[root]# moab --about

Defaults:   server=:42559  cfgdir=/opt/moab (env)  vardir=/opt/moab
Build dir:  /tmp/jenkins/workspace/MWM-9.0.0/label/build-sles12
Build host: us-devops-build10
Build date: Fri Oct 09 13:00:00 MST 2015
Build args: NA
Compiler Flags:  -D__M64 -D_BUILDDATETIME="2015100913" -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 9.0.0 (revision 2015100913, changeset
14dee972ebcee919207e48054e9f285db9f6a555)
```

12. Start Moab.
    - Red Hat 6-based or SUSE 11-based systems

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

    - Red Hat 7-based or SUSE 12-based systems

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

Upgrading Moab Accounting Manager

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

Moab Accounting Manager 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.

Upgrade Moab Accounting Manager

On the MAM Server Host, do the following:

1. Determine the MAM Accounting admin user and change to that user.
    - If you are upgrading MAM from a version *prior* to 9.0, use glsuser.

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

- If you are upgrading MAM at or after 9.0, use mam-list-users.

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

2. Determine whether you need to migrate your database.
   a. Determine your database version.
      - If you are upgrading MAM from a version *prior* to 9.0, run `goldsh System Query`.

```
[mam]$ goldsh System Query
```

      - If you are upgrading MAM at or after 9.0, run `mam-shell System Query`.

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

   b. If the current version is lower than 9.0, you must migrate your database. The steps required to do so are incorporated in the remaining steps for this topic.
3. Stop the server daemon.
   - If you are upgrading MAM from a version *prior* to 9.0, run goldd -k.

```
[mam]$ goldd -k
```

   - If you are upgrading MAM at or after 9.0, run mam-server -k.

```
[mam]$ mam-server -k
```

4. If you determined that you must migrate your database, create a database backup.
   - PostgreSQL database.

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

   - MySQL database.

```
[mam]$ mysqldump -u <mam_database_user> -p <old_database_name> > /tmp/<old_
database_name>.sql
```

5. Verify that each of the prerequisites listed in 1.1 Installing have been satisfied.
6. Download the latest MAM build (`mam-<version>.tar.gz`) from the Adaptive Computing website.

   The variable marked <*version*> indicates the build's version.

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

distribution.

```
[mam]$ tar -zxvf mam-9.0.0.tar.gz
[mam]$ cd mam-9.0.0
```

8. Configure Moab Accounting Manager by running the `configure` script with the desired options.

   It is recommended 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.

   In addition to your previous configuration options, it is recommended that you specify the prior symmetric key used between Moab Workload Manager and Moab Accounting Manager via the `--with-key` configure option so that you do not have to change the key in Moab. You can find this key in the prior `site.conf` or `mam-site.conf` file.

   The examples in this guide demonstrate installing the new version of Moab Accounting Manager over the previous version by using the default directory, database name and port.

   IMPORTANT: client and server command names have changed beginning with 9.0. If you want to create symbolic links to enable you to continue to use the old client and server command names, use the --with-legacy-links option with configure. 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.

   If you are migrating the database, you can install the new version of Moab Accounting Manager with a new installation directory, port and database name by specifying `--prefix`, `--with-port` and `--with-db-name` configuration options that are different from the old version. This would allow you to run both the new and old versions simultaneously.

   ```
   [mam]$ ./configure --with-key=<MAMSecretKey>
   ```

9. To compile the program, type `make`.

   ```
   [mam]$ make
   ```

   If you only need to install the clients on a particular system, replace make with make clients-only. If you only need to install the web GUI on a particular system, replace make with make gui-only.

10. Run `make install` as root to install Moab Accounting Manager.

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

> If you only need to install the clients on a particular system, replace "make install" with "make install-clients-only". If you only need to install the web GUI on a particular system, replace "make install" with "make install-gui-only".

11. If you are migrating your database and used the `--with-db-name` option to specify a new database name that does not already exist, you must create and populate the database from the dump.
    a. Create the new database.
       - PostgreSQL database.
         ```
         [postgres]$ psql
         create database "<new_db_name>";
         ```
       - MySQL database.
         ```
         [root]# mysql
         create database '<new_db_name>';
         ```
    b. Import the old data into the new database.
       - PostgreSQL database.
         ```
         [mam]$ psql -U <mam_database_user> -W <new_db_name> < /tmp/<old_db_name>.sql
         ```
       - MySQL database.
         ```
         [mam]$ mysql -u <mam_database_user> -p <new_db_name> < /tmp/<old_db_name>.sql
         ```
12. Edit the configuration files as necessary. You may want to compare your existing configuration files with those distributed with the new release to determine if you want to merge and change any of the new options within your configuration files.
    - If you are upgrading MAM from a version *prior* to 9.0, the install process will have saved your prior configuration files to {goldd,gold,goldg}.conf.pre-9.0 and written new default server configuration file as mam-{server,client,gui}.conf. You will need to merge any non-default parameters from your prior config files to the new default config files.
      ```
      [mam]$ diff /opt/mam/etc/goldd.conf.pre-9.0 /opt/mam/etc/mam-server.conf
      [mam]$ vi /opt/mam/etc/mam-server.conf
      [mam]$ diff /opt/mam/etc/gold.conf.pre-9.0 /opt/mam/etc/mam-client.conf
      [mam]$ vi /opt/mam/etc/mam-client.conf
      [mam]$ diff /opt/mam/etc/goldg.conf.pre-9.0 /opt/mam/etc/mam-gui.conf
      [mam]$ vi /opt/mam/etc/mam-gui.conf
      ```
    - If you are upgrading MAM at or after 9.0, merge and change any of the new options supplied in the new default configuration files (saved in mam-{server,client,gui}.conf.dist) into your existing configuration files (mam-{server,client,gui}.conf).

```
[mam]$ diff /opt/mam/etc/mam-server.conf /opt/mam/etc/mam-server.conf.dist
[mam]$ vi /opt/mam/etc/mam-server.conf
[mam]$ diff /opt/mam/etc/mam-client.conf /opt/mam/etc/mam-client.conf.dist
[mam]$ vi /opt/mam/etc/mam-client.conf
[mam]$ diff /opt/mam/etc/mam-gui.conf /opt/mam/etc/mam-gui.conf.dist
[mam]$ vi /opt/mam/etc/mam-gui.conf
```

Verify that your current path points to your newly installed clients and server.

```
[mam]$ which mam-server
/opt/mam/sbin/mam-server
```

13. Start the server daemon back up.

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

14. If you are migrating your database to 9.0, 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 (9.0). 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 Moab Accounting Manager version 7.2, you must run five migration scripts: the first to migrate the database schema from 7.2 to 7.3, the second to migrate from 7.3 to 7.5, the third to migrate the database schema from 7.5 to 8.0, the fourth to migrate the database schema from 8.0 to 8.1, and the fifth to migrate the database schema from 8.1 to 9.0.

```
[mam]$ sbin/migrate_7.2-7.3.pl
[mam]$ sbin/migrate_7.3-7.5.pl
[mam]$ sbin/migrate_7.5-8.0.pl
[mam]$ sbin/migrate_8.0-8.1.pl
[mam]$ sbin/migrate_8.1-9.0.pl
```

15. Verify that the resulting database schema version is 9.0.

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

Name                    Version Description
----------------------- ------- -----------------
Moab Accounting Manager 9.0     Commercial Release
```

16. Verify that the executables have been upgraded to 9.0.0.

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

Moab Accounting Manager version 9.0.0
```

Upgrading Moab Web Services

This topic provides instructions to upgrade Moab Web Services to the latest release version. Depending on which version of MWS you are presently

running, upgrade instructions may vary.

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

Before You Upgrade

Before upgrading MWS, Adaptive Computing recommends you upgrade to Java 8 and MongoDB 2.4.x.

Upgrade to Java 8

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

If you wish to upgrade to Java 8, refer to the Install Java on page 33 instructions.

Upgrade the MongoDB

> It is highly recommended that you perform a full database backup before updating your database. This can be done using the `mongodump` utility documented in the MongoDB documentation (http://www.mongodb.org/display/DOCS/Backups).

On the host where the MWS MongoDB database resides, do the following:

1. Check your MongoDB version.

```
[root]# mongo --version
```

2. If your MongoDB version is prior to 2.4, upgrade the database. When upgrading, you must add 'exclude=mongodb-org mongodb-org-server' to the /etc/yum.repos.d/mongodb.repo file to maintain 2.4.x. Depending on your MongoDB version, the file path may be /etc/yum.repos.d/10gen.repo.

```
[mongodb]
name=MongoDB Repository
baseurl=http://downloads-distro.mongodb.org/repo/redhat/os/x86_64/
gpgcheck=0
enabled=1
exclude=mongodb-org mongodb-org-server
```

Upgrade Moab Web Services

These instructions also include instructions for Tomcat. Depending on your OS, the Tomcat file or user names may not contain the version distinction (i.e. tomcat6).

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 build (`mws-<version>.tar.gz`) from the Adaptive Computing website.

   The variable marked <*version*> is the desired version of the suite; for example, 9.0.0.

3. In the directory you created earlier, extract the contents of the MWS download tarball and then change directory into the extracted directory. For example:

```
[root]# tar xvzf mws-9.0.0.tar.gz
[root]# cd /tmp/mws-install/mws-9.0.0
```

4. Stop Tomcat, re-deploy `mws.war`, and remove the exploded `mws` directory.
   - Red Hat 6-based systems

```
[root]# service tomcat6 stop
[root]# cp mws.war /usr/share/tomcat6/webapps
[root]# rm -rf /usr/share/tomcat6/webapps/mws
```

   - Red Hat 7-based or SUSE 12-based systems

```
[root]# systemctl stop tomcat.service
[root]# cp mws.war /usr/share/tomcat/webapps
[root]# rm -rf /usr/share/tomcat/webapps/mws
```

5. Create the MWS home directory and subdirectories. See Configuration in the *Moab Web Services Reference Guide* for more information.

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

   Here is a sample script for this setup:

```
[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 # Depending on your OS, the Tomcat username
might be tomcat6.
[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
```

6. Copy the extracted utility files to the utility directory created above and give the tomcat user ownership of the directory.

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

7. Merge the changes in the /tmp/mws-install/mws-9.0.0/ `mws-config.groovy` file into your existing `/opt/mws/etc/mws-config.groovy`. Depending on your current MWS version, do the following as needed:
   - If Insight is part of your configuration, add the Insight configuration information (insight.username, insight.password, insight.url); prior version to 9.0
   - If Viewpoint is part of your configuration, register Viewpoint as client; prior to version 9.0
   - Change the moab.messageQueue.port to 5570; prior to version 8.0
   - Configure and appender for the audit log; prior to version 8.0
   - Change the layout to "new com.ace.mws.logging.ACPatternLayout()" for the output format of each log entry; prior to version 8.0
   - Remove the mws.suite parameter and the mam.* parameters (they have been moved to /opt/mws/etc/mws.d/); prior to version 8.0
   - 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.

   The following is an example of the merged mws-config.groovy file for MWS 9.0:

```
// 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

// MongoDB configuration.
// grails.mongo.username = "mws_user"
// grails.mongo.password = "<ENTER-KEY-HERE>"

// Insight configuration.
// dataSource_insight.username = "mws"
// dataSource_insight.password = "changeme!"
// dataSource_insight.url = "jdbc:postgresql://127.0.0.1:5432/moab_insight"

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


// Sample OAuth Configuration
grails.plugin.springsecurity.oauthProvider.clients = [
       [
               clientId:"viewpoint",
               clientSecret:"<ENTER-CLIENTSECRET-HERE>",
                authorizedGrantTypes:["password"]
        ]
]


// 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"

log4j = {
        // Configure an appender for the events log.
        def eventAppender = new org.apache.log4j.rolling.RollingFileAppender(
                name: 'events', layout: pattern(conversionPattern: "%m%n"))
        def rollingPolicy = new org.apache.log4j.rolling.TimeBasedRollingPolicy(
                fileNamePattern: '/opt/mws/log/events.%d{yyyy-MM-dd}',
                activeFileName: '/opt/mws/log/events.log')
        rollingPolicy.activateOptions()
```

```
        eventAppender.setRollingPolicy(rollingPolicy)

        // Configure an appender for the audit log.
        def auditAppender = new org.apache.log4j.rolling.RollingFileAppender(
                        name: 'audit',
                        layout: new com.ace.mws.logging.ACPatternLayout("%j\t\t\t%c{1}\t\t\t%m%n")
        def auditRollingPolicy = new org.apache.log4j.rolling.TimeBasedRollingPolicy(
                        fileNamePattern: '/opt/mws/log/audit.%d{yyyy-MM-dd}',
                        activeFileName: '/opt/mws/log/audit.log')
        auditRollingPolicy.activateOptions()
        auditAppender.setRollingPolicy(auditRollingPolicy)

        appenders {
                rollingFile name: 'stacktrace',
                                file: '/opt/mws/log/stacktrace.log',
                                maxFileSize: '100MB'
                rollingFile name: 'rootLog',
                                file: '/opt/mws/log/mws.log',
                                maxFileSize: '100MB', //The maximum file size for a single log file
                                maxBackupIndex: 10, //Retain only the 10 most recent log files, del
logs to save space
                                layout: new com.ace.mws.logging.ACPatternLayout(), //Configures the
format of each log entry
                                threshold: org.apache.log4j.Level.ERROR //Ignore any logging entrie
verbose than this threshold

                appender eventAppender
                appender auditAppender
        }

        // NOTE: This definition is a catch-all for any logger not defined below
        root {
                error 'rootLog'
        }

        // Individual logger configurations
        debug   'com.ace.mws',
                        'grails.app.conf.BootStrap',
                        'grails.app.controllers.com.ace.mws',
                        'grails.app.domain.com.ace.mws',
                        'grails.app.filters.com.ace.mws',
                        'grails.app.services.com.ace.mws',
                        'grails.app.tagLib.com.ace.mws',
                        'grails.app.jobs.com.ace.mws',
                        'grails.app.gapiParser',
                        'grails.app.gapiRequest',
                        'grails.app.gapiSerializer',
                        'grails.app.translator',
                        'plugins'       // MWS plugins

        info    'com.ace.mws.gapi.Connection',
                        'com.ace.mws.gapi.parsers',
                        'grails.app.service.grails.plugins.reloadconfig',
                        'com.ace.mws.gapi.serializers'

        off             'org.codehaus.groovy.grails.web.errors'

        // Logs event information to the events log, not the rootLog
        trace additivity:false, events:'com.ace.mws.events.EventFlatFileWriter'
        // Logs audit information to the audit log, not the rootLog
        trace additivity:false, audit:'mws.audit'
}
```

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. 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.
10. Upgrade the schema of the `mws` database in MongoDB.

    > **Warning:** You *must* perform this step, regardless of whether you upgraded MongoDB to version 2.4.x.

    Run the database migration script provided with MWS. (It is safe to run this script more than once. If for any reason, errors occur during the execution of the script, run it again.)

    ```
    [root]# mongo -u mws_user mws /opt/mws/utils/db-migrate.js -p
    ```

    > **Note:** The script might take several minutes to execute.

11. Start Tomcat.
    - Red Hat 6-based systems

      ```
      [root]# service tomcat6 start
      ```

    - Red Hat 7-based and SUSE 12-based systems

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

12. Visit http://localhost:8080/mws/ in a web browser to verify that MWS is running again.

    You will see some sample queries and a few other actions.

13. Log into 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`.)

    > **Note:** If you encounter problems, or if MWS does not seem to be running, see the steps in Moab Web Services Issues on page 122.

Migrating the MAM Database from MySQL to PostgreSQL

PostgreSQL is the preferred DBMS for MAM. Customers who have already installed MySQL as the DBMS for MAM 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-mysql2pgsq for additional details.

To Migrate the MAM Database

This procedure was successfully tested on an actual customer MySQL database with millions of transactions on CentOS 6.4. 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** - Installing Moab Web Services on page 31
   - **RPM Install** - Installing Moab Web Services on page 87
3. Install the prerequisite packages.

```
[root]# yum 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 Moab Accounting Manager.

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

Chapter 3 RPM installation

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

The RPM Installation method only supports installation on CentOS 6.x, 7.x, RHEL 6.x, 7.x, or Scientific Linux 6.x, 7.x. See Chapter 2 Manual Installation on page 15 if installing on other supported operating systems.

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. Please note that the same commands will work for a non-root user with the `sudo` command.

Related Topics

- Chapter 1 Planning your Installation on page 3
- Preparing the Host for RPM Installations on page 78

Installation

In this section:

Preparing the Host for RPM Installations

Adaptive Computing provides RPMs to install the various component servers (such as Moab Server, MWS Server, Torque Server).

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

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

Set Up Proxies

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

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

Enable the Adaptive Computing Package Repository

> Whether you are installing RPMs on one host or on several hosts, each host (physical machine) on which a server is installed (Torque Server Host, Moab Server Host, etc) *must* have the Adaptive Computing Package Repository enabled. It is not necessary to enable the Adaptive Computing repository on the Torque MOM Hosts or client hosts.

Do the following:

1. Download the latest 9.0.0 RPM suite tarball (`moab-hpc-basic-suite-<version>-<timestamp>-<OS>.tar.gz`, for example) from the Adaptive Computing website.
2. Untar the downloaded package.

```
[root]# tar zxf moab-hpc-basic-suite-<version>-<timestamp>-<OS>.tar.gz
```

3. Change directories into the untarred directory.

> Consider reviewing the README file for additional details on using the RPM distribution tarball.

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

> For a description of the options of the repository installer script, run:
>
> ```
> [root]# ./install-rpm-repos.sh -h
> ```

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

> If the installation returns the following warning line:
>
> `Warning: RPMDB altered outside of yum.`
>
> This is normal and can safely be ignored.

The [*<repository-directory>*] option is the directory where you want to copy the RPMs. If no argument is given, run "`install-rpm-repos.sh -h`" and note the default directory location. If the [*<repository-directory>*] already exists, RPMs will be added to the existing directory. No files are overwritten in [*<repository-directory>*]. A repository file is also created in `/etc/yum.repos.d/` and points to the [*<repository-directory>*] location.

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

The script installs the `createrepo` package and its dependencies. You must answer "y" to all the questions in order for the RPM install of the suite to work. Additionally, the script installs the EPEL and 10gen repositories.

5. Test the repository.

```
[root]# yum search moab
```

If no error is given, the repository is correctly installed. The output will look similar to the following (varying slightly depending on the suite and build type):

```
...

moab-hpc-basic-suite.noarch : Moab HPC Basic Suite virtual package
moab-perl-RRDs.noarch : Moab RRDs
moab-tomcat-config.x86_64 : Tomcat Configuration for Web Services
moab-web-services.x86_64 : Moab Web Services
moab-workload-manager.x86_64 : Moab Workload Manager
moab-workload-manager-client.x86_64 : Moab Workload Manager Client
moab-workload-manager-common.x86_64 : Moab Workload Manager Common Files
moab-perl-data.noarch : Perl Configuration for perl packages by Adaptive Computing
moab-torque-client.x86_64 : Torque Client
moab-torque-common.x86_64 : Torque Common Files
moab-torque-devel.x86_64 : Torque Development Files
moab-torque-mom.x86_64 : Torque MOM agent
moab-torque-server.x86_64 : Torque Server
...
```

Related Topics

- Chapter 1 Planning your Installation on page 3
- Chapter 3 RPM installation on page 77

Installing Torque Resource Manager

If you intend to use Torque Resource Manager 6.0.0 with Moab Workload Manager, you must run Moab version 8.0 or later. However, some Torque 6.0 functionality requires Moab 9.0 or later.

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

In this topic:

Prerequisites

In this section:

Open Necessary Ports

Torque requires certain ports to be open for essential communication.

- For client and pbs_mom communication to pbs_server, the default port is 15001.
- For pbs_server communication to pbs_mom, the default port is 15002.
- For pbs_mom communication to pbs_mom, the default port is 15003.

For more information on how to configure the ports that Torque uses for communication, see Configuring Ports in the *Torque Resource Manager Administrator Guide* for more information.

If you have a firewall enabled, do the following:

1. On the Torque Server Host:
   - Red Hat 6-based systems using iptables

```
[root]# iptables-save > /tmp/iptables.mod
[root]# vi /tmp/iptables.mod

# Add the following line immediately *before* the line matching
# "-A INPUT -j REJECT --reject-with icmp-host-prohibited"
-A INPUT -p tcp --dport 15001 -j ACCEPT

[root]# iptables-restore < /tmp/iptables.mod
[root]# service iptables save
```

   - Red Hat 7-based systems using firewalld

```
[root]# firewall-cmd --add-port=15001/tcp --permanent
[root]# firewall-cmd --reload
```

2. On each Torque MOM Host (Compute Hosts):
   - Red Hat 6-based systems using iptables

```
[root]# iptables-save > /tmp/iptables.mod
[root]# vi /tmp/iptables.mod

# Add the following lines immediately *before* the line matching
# "-A INPUT -j REJECT --reject-with icmp-host-prohibited"
-A INPUT -p tcp --dport 15002 -j ACCEPT
-A INPUT -p tcp --dport 15003 -j ACCEPT

[root]# iptables-restore < /tmp/iptables.mod
[root]# service iptables save
```

   - Red Hat 7-based systems using firewalld

```
[root]# firewall-cmd --add-port=15002/tcp --permanent
[root]# firewall-cmd --add-port=15003/tcp --permanent
[root]# firewall-cmd --reload
```

Verify the hostname

On the Torque Server Host, confirm your host (with the correct IP address) is in your `/etc/hosts` file. To verify that the hostname resolves correctly, make sure that hostname and hostname -f report the correct name for the host.

Install Torque Server

On the Torque Server Host, do the following:

1. If you are installing the Torque Server on its own host (recommend) and *not* on the same host where you installed another server (such as Moab Server), verify you completed the steps to prepare the host. See Preparing the Host for RPM Installations on page 78 for more information.
2. Install the Torque Server RPM.

   ```
   yum install moab-torque-server
   ```

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

   Example:

   ```
   vi /var/spool/torque/server_priv/nodes

   node01 np=16
   node02 np=16
   ...
   ```

4. Start the Torque server.
   - Red Hat 6-based systems

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

   - Red Hat 7-based systems

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

Install Torque MOMs

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

Do the following:

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

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

2. On each Torque MOM Host, install the RPMs; `moab-torque-common` is installed *first*.

```
[root]# ssh root@<torque-mom-host>
[root]# yum install moab-torque-common-*.rpm moab-torque-mom-*.rpm moab-torque-
client-*.rpm
```

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

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

4. On each Torque MOM Host, edit the `/var/spool/torque/mom_priv/config` file. This file is identical for all compute nodes and can be created on the Torque Server and distributed in parallel to all systems.

```
[root]# vi /var/spool/torque/mom_priv/config

$pbsserver     <torque_server_hostname>   # hostname running pbs server
$logevent      225                        # bitmap of which events to log
```

5. On each Torque MOM Host, start the pbs_mom daemon.
   - Red Hat 6-based systems

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

   - Red Hat 7-based systems

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

6. If you installed the Torque Client RPM on the MOMs, then on each Torque MOM Host, start the trqauthd daemon.
   - Red Hat 6-based systems.

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

   - Red Hat 7-based system.

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

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 Adminstrator Guide* for more information.

Related Topics

Chapter 3 RPM installation on page 77

Installing Moab Workload Manager

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

In this topic:

Open Necessary Ports

Moab uses a configurable server port (default 42559) for client-server communication. If you intend to run client commands on a host other than the Moab Host, or if you will be using Moab in a grid, and if you have a firewall enabled, then you will need to configure the firewall to allow the server port.

On the Moab Server Host, do the following:

- Red Hat 6-based systems using iptables

```
[root]# iptables-save > /tmp/iptables.mod
[root]# vi /tmp/iptables.mod

# Add the following lines immediately *before* the line matching
# "-A INPUT -j REJECT --reject-with icmp-host-prohibited"

# Needed on the Moab server for off-host client communication
-A INPUT -p tcp --dport 42559 -j ACCEPT

[root]# iptables-restore < /tmp/iptables.mod
[root]# service iptables save
```

- Red Hat 7-based systems using firewalld

```
[root]# firewall-cmd --add-port=42559/tcp --permanent
[root]# firewall-cmd --reload
```

Install Moab Server

On the *Moab* Server Host do the following:

1. If you have not already done so, complete the steps to prepare the Moab Server Host. See Preparing the Host for RPM Installations on page 78.
2. Install RPM packages.
   a. Install the Moab Server RPMs.
   ```
   [root]# yum install moab-workload-manager moab-workload-manager-hpc-
   configuration
   ```
   b. If you are using Torque as a resource manager and installed the Torque Server on a separate host (Torque Server Host; recommended) from the Moab Server (Moab Server Host), you will need to install the Torque client RPM on the Moab Server Host in order for Moab to interact with Torque.
   ```
   [root]# yum install moab-torque-client
   ```
3. Source the following file to add the Moab executable directories to your current shell *$PATH* environment.
   ```
   [root]# . /etc/profile.d/moab.sh
   ```
4. Copy your license file into the same directory as `moab.cfg` (`/opt/moab/etc/` by default). For example:
   ```
   [root]# cp moab.lic $MOABHOMEDIR/etc/moab.lic
   ```
   To verify the current status of your license, use `moab --about`.

   Moab checks the status of the license every day just after midnight. At 60 and 45 days before, and daily from 30 days before license expiration to and including the license expiration date, Moab sends an e-mail to all level 1 administrators informing them of the pending Moab license expiration. A log record is also made of the upcoming expiration event. For the notifications to occur correctly, you must enable administrator email notification (see Notifying Administrators of Failures in the *Moab Workload Manager Administrator Guide*) and `moab.cfg` must contain email addresses for level 1 administrators. For example:
   ```
   ADMINCFG[1] USERS=u1,u2,u3[,...]

   USERCFG[u1] EMAILADDRESS=u1@company.com
   USERCFG[u2] EMAILADDRESS=u2@company.com
   USERCFG[u3] EMAILADDRESS=u3@company.com

   MAILPROGRAM DEFAULT
   ```
   Moab will not run without a license. For information about obtaining a trial license, please contact Adaptive Computing.
5. If you are using Torque as your resource manager and you installed the

Torque Server on a separate host (Torque Server Host) from the Moab Server (Moab Server Host), do the following:

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

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

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

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

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

```
[root]# vi /etc/hosts

<Torque_server_ip_address> <Torque_server_hostname> <Torque_server_FQDN>
```

c. Start the trqauthd daemon.

- Red Hat 6-based systems

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

- Red Hat 7-based systems

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

6. Start Moab (assumes Moab configured with the `--with-int` option).

- Red Hat 6-based systems

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

- Red Hat 7-based systems

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

Configure Torque to Trust Moab

If you are using Torque as a resource manager and you installed the Torque Server on a separate host (Torque Host); recommended, do the following:

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

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

Verify the Installation

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

- 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
```

Related Topics

Chapter 3 RPM installation on page 77

Installing Moab Web Services

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

This topic contains instructions on how to install, configure, and start Moab Web Services (MWS).

In this topic:

Open Necessary Ports

If your site is running firewall software on its hosts, you will need to configure the firewall to allow connections to the necessary ports.

In this section:

Open the Tomcat Port (8080)

On the MWS Server Host, do the following:

- Red Hat 6-based systems using iptables

```
[root]# iptables-save > /tmp/iptables.mod
[root]# vi /tmp/iptables.mod

# Add the following lines immediately *before* the line matching
# "-A INPUT -j REJECT --reject-with icmp-host-prohibited"

-A INPUT -p tcp --dport 8080 -j ACCEPT

[root]# iptables-restore < /tmp/iptables.mod
[root]# service iptables save
```

- Red Hat 7-based systems using firewalld

```
[root]# firewall-cmd --add-port=8080/tcp --permanent
[root]# firewall-cmd --reload
```

- SUSE 12-based systems using SuSEfirewall2

```
[root]# vi /etc/sysconfig/SuSEfirewall2

FW_SERVICES_EXT_TCP="8080"

[root]# service SuSEfirewall2_setup restart
```

Open the MWS MongoDB Database Port (27017)

Depending on your system configuration, your MongoDB databases may not be installed on the same host as their corresponding component servers. For example, you may choose to install the MWS MongoDB database on the same host where you have installed other MongoDB databases instead of on the MWS Server Host.

Do the following, as needed:

- If you have chosen to install the MWS MongoDB database on the *same* host you installed other MongoDB databases (for example, the same host you installed the Moab MongoDB database), confirm the firewall port (27017) is already opened on that host.
- If you have chosen to install the MWS MongoDB database on a *separate* host from other MongoDB databases, you will need to open the MWS MongoDB database port in firewall for that host. To open the port in the firewall, do the following:

- Red Hat 6-based systems using iptables

```
[root]# iptables-save > /tmp/iptables.mod
[root]# vi /tmp/iptables.mod

Add the following line immediately *before* the line matching
# "-A INPUT -j REJECT --reject-with icmp-host-prohibited"

-A INPUT -p tcp --dport 27017 -j ACCEPT

[root]# iptables-restore < /tmp/iptables.mod
[root]# service iptables save
```

- Red Hat 7-based systems using firewalld

```
[root]# firewall-cmd --add-port=27017/tcp --permanent
[root]# firewall-cmd --reload
```

- SUSE 11-based or SUSE 12-based systems using SuSEfirewall2 (SUSE 11 is not supported on the MWS Server Host)

```
[root]# vi /etc/sysconfig/SuSEfirewall2

FW_SERVICES_EXT_TCP="27017"

[root]# service SuSEfirewall2_setup restart
```

Install Dependencies, Packages, or Clients

In this section:

Install Java

Install the Linux x64 RPM version of 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 OpenJDK/IcedTea, GNU Compiler for Java, and so on cannot run MWS.

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 to the Oracle Java download page (http://java.com/en/download/linux_manual.jsp).
   b. Copy the URL for the Linux x64 RPM version, and run the following command:

```
[root]# rpm -Uh <URL>
```

Install MongoDB

To install and enable MongoDB, on the MWS Host, do the following:

1. Install `mongo-10gen-server`.

```
[root]# yum install mongo-10gen-server
```

2. Start MongoDB.

> There may be a short delay (approximately 3 minutes) for Mongo to start the first time.

- Red Hat 6-based systems

```
[root]# chkconfig mongod on
[root]# service mongod start
```

- Red Hat 7-based systems

```
[root]# cat > /usr/lib/systemd/system/mongodb.service <<End-of-file
[Unit]
Description=High-performance, schema-free document-oriented database
After=syslog.target network.target

[Service]
Type=forking
User=mongod
Group=mongod
Environment=CONFIG=/etc/mongod.conf
Environment=OPTIONS=
EnvironmentFile=-/etc/sysconfig/mongod
ExecStart=/usr/bin/mongod -f \$CONFIG \$OPTIONS
PrivateTmp=true
LimitNOFILE=65536
TimeoutStartSec=180
StandardOutput=syslog
StandardError=syslog

[Install]
WantedBy=multi-user.target
End-of-file
[root]# rm -f /etc/init.d/mongod
[root]# systemctl enable mongodb.service
[root]# systemctl start mongodb.service
```

3. Prepare the MongoDB database by doing the following:
   a. 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.addUser("admin_user", "secret1");
> db.auth ("admin_user", "secret1");

> use moab;
> db.addUser("moab_user", "secret2");
> db.addUser("mws_user", "secret3", true);

> use mws;
> db.addUser("mws_user", "secret3");
> 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 (http://docs.mongodb.org/manual/tutorial/control-access-to-mongodb-with-authentication/) for more information.

b. Enable authentication in MongoDB.

- Red Hat 6-based systems

```
[root]# vi /etc/mongod.conf
auth = true
[root]# service mongod restart
```

- Red Hat 7-based systems

```
[root]# vi /etc/mongod.conf
auth = true
[root]# systemctl restart mongodb.service
```

Install MWS Server

On the MWS Host, do the following:

1. Install the MWS RPMs.

```
[root]# yum install moab-web-services moab-web-services-hpc-configuration
```

2. 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 on the same host.

```
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.

```
# Listen to local interface only. Comment out to listen on all interfaces.
#bind_ip=127.0.0.1
```

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 on page 90 earlier in this topic.

```
MONGOUSER    moab_user
MONGOPASSWORD secret2
```

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

- Red Hat 6-based systems

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

Mongo connection (localhost) is up (credentials are set)
```

- Red Hat 7-based systems

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

Mongo connection (localhost) is up (credentials are set)
```

3. 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`.

- Red Hat 6-based systems

```
[root]# service moab stop
[root]# dd if=/dev/urandom count=18 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]# service moab start
```

- Red Hat 7-based systems

```
[root]# systemctl stop moab.service
[root]# dd if=/dev/urandom count=18 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.

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

- Red Hat 6-based systems

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

- Red Hat 7-based systems

```
[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 messsages from Moab. Furthermore, all attempts to access the MWS service resource will fail.

b. 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)
```

4. Set up the MWS configuration file.

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

- **moab.secretKey**: Must match the Moab 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

// Replace <ENTER-KEY-HERE> with the contents of /opt/moab/etc/.moab.key.

moab.secretKey = "<ENTER-KEY-HERE>"
moab.server = "localhost"
moab.port = 42559

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

moab.messageQueue.secretKey = "<ENTER-KEY-HERE>"

// Change these to be whatever you like.

auth.defaultUser.username = "moab-admin"
auth.defaultUser.password = "changeme!"
```

If you do not change **auth.defaultUser.password**, your MWS will not be secure (because anyone reading these instructions would be able to log into your MWS). Here are some tips for choosing a good password.

b. Do *one* of the following:

- If you are configuring an MWS connection to your LDAP server, add the following parameters to the `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.

If you followed the Adaptive Computing tutorial, Setting Up OpenLDAP on CentOS 6 on page 97, your **ldap.directory.type** should be set to "OpenLDAP Using InetOrgPerson Schema." However, the use of other schemas is supported. For more information see LDAP Configuration Using mws-config.groovy.

To see how to configure a secure connection to the LDAP server, see Securing the LDAP Connection.

- If you are configuring MWS to use PAM, add the the **pam.configuration.service** parameter to the `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.

There is a security risk when authenticating local users through your PAM configuration. This behavior is highly discouraged and not supported by Adaptive Computing.

For more information about PAM configuration with MWS, see PAM (Pluggable Authentication Module) Configuration Using mws-config.groovy.

You can configure only one authentication method in `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.

c. Add the **grails.mongo.username** and **grails.mongo.password** parameters to the `mws-config.groovy` file. Use the MWS credentials you added to MongoDB.

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

5. Start or restart Tomcat.
   - Red Hat 6-based systems

```
[root]# chkconfig tomcat on
[root]# service tomcat restart
```

   - Red Hat 7-based systems

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

Verify the Installation

1. Open a web browser.
2. Navigate to `http://<server>:8080/mws/`. You will see some sample queries and a few other actions.
3. 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 Chapter 4 Troubleshooting on page 119.

Related Topics

- Chapter 3 RPM installation on page 77
- Installing Moab Workload Manager on page 84

Additional Configuration

In this section:

Configuring SSL in Tomcat

To configure SSL in Tomcat, please refer to the Apache Tomcat documentation (http://tomcat.apache.org/tomcat-7.0-doc/ssl-howto.html).

Setting Up OpenLDAP on CentOS 6

These instructions are intended to help first-time LDAP administrators get up and running. The following procedures contain instructions for getting started using OpenLDAP on a CentOS 6 system. For more complete information on how to set up OpenLDAP see the OpenLDAP documentation (http://www.openldap.org/doc/admin24/).

In this topic:

Adaptive Computing is not responsible for creating, maintaining, or supporting customer LDAP or Active Directory configurations.

Installing and Configuring OpenLDAP on Centos 6

First, you will need to install OpenLDAP. These instructions explain how you can do this on a CentOS 6 system.

To install and configure OpenLDAP on Centos 6

1. Run the following command:

```
[root]# yum -y install openldap openldap-clients openldap-servers
```

2. Generate a password hash to be used as the admin password. This password hash will be used when you create the root user for your LDAP installation. For example:

```
[root]# slappasswd
New password : p@ssw0rd
Re-enter new password : p@ssw0rd
{SSHA}5lPFVw19zeh7LT53hQH69znzj8TuBrLv
```

3. Add the root user and the root user's password hash to the OpenLDAP configuration in the `olcDatabase={2}bdb.ldif` file. The root user will have permissions to add other users, groups, organizational units, etc. Do the following:

   a. Run this command:

```
[root]# cd /etc/openldap/slapd.d/cn\=config
[root]# vi olcDatabase\=\{2\}bdb.ldif
```

   b. If the **olcRootPW** attribute does not already exist, create it. Then set the value to be the hash you created from `slappasswd`. For example:

```
olcRootPW: {SSHA}5lPFVw19zeh7LT53hQH69znzj8TuBrLv
...
```

4. While editing this file, change the distinguished name (DN) of the **olcSuffix** to something appropriate. The suffix typically corresponds to your DNS domain name, and it will be appended to the DN of every other LDAP entry in your LDAP tree.

   For example, let's say your company is called Acme Corporation, and that your domain name is "acme.com". You might make the following changes to the `olcDatabase={2}bdb.ldif` file:

```
olcSuffix: dc=acme,dc=com
...
olcRootDN: cn=Manager,dc=acme,dc=com
...
olcRootPW: {SSHA}5lPFVw19zeh7LT53hQH69znzj8TuBrLv
...
```

Do not set the cn of your root user to "root" (`cn=root,dc=acme,dc=com`), or OpenLDAP will have problems.

Throughout the following examples in this topic, you will see `dc=acme,dc=com`. "acme" is only used as an example to illustrate what you would use as your own domain controller if your domain name was "acme.com". You should replace any references to "acme" with your own organization's domain name.

5. Modify the DN of the root user in the `olcDatabase={1}monitor.ldif` file to match the **olcRootDN** line in the `olcDatabase={2}bdb.ldif` file. Do the following:

   a. Run this command to edit the `olcDatabase={2}bdb.ldif` file:

   ```
   [root]# vi olcDatabase\=\{1\}monitor.ldif
   ```

   b. Modify the **olcAccess** line so that the **dn.base** matches the **olcRootDN** from the `olcDatabase={2}bdb.ldif` file. (In this example, **dn.base** should be "cn=Manager,dc=acme,dc=com".)

   ```
   olcAccess: {0}to * by
   dn.base="gidNumber=0+uidNumber=0,cn=peercred,cn=external,cn=auth" read by
   dn.base="cn=Manager,dc=acme,dc=com" read by * none
   ```

   c. Now the root user for your LDAP is `cn=Manager,dc=acme,dc=com`. The root user's password is the password that you entered using `slappasswd` earlier in this procedure, which, in this example, is **p@ssw0rd**

6. Hide the password hashes from users who should not have permission to view them.

   > A full discussion on configuring access control in OpenLDAP is beyond the scope of this tutorial. For help, see the OpenLDAP Access Control documentation (http://www.openldap.org/doc/admin24/access-control.html).

   a. Run this command to edit the `oclDatabase\=\{2\}bdb.ldif` file:

   ```
   [root]# vi olcDatabase\=\{2\}bdb.ldif
   ```

   b. Add the following two lines to the end of the file to restrict users from viewing other users' password hashes.

   ```
   olcAccess: {0}to attrs=userPassword by self write by
   dn.base="cn=Manager,dc=acme,dc=com" write by anonymous auth by * none
   olcAccess: {1}to * by dn.base="cn=Manager,dc=acme,dc=com" write by self write by
   * read
   ```

   These lines allow a user to read and write his or her own password. It also allows a manager to read and write anyone's password. Anyone, including anonymous users, is allowed to view non-password attributes of other users.

7. Make sure that OpenLDAP is configured to start when the machine starts up, and start the OpenLDAP service.

   ```
   [root]# chkconfig slapd on
   [root]# service slapd start
   ```

8. Now, you must manually create the "dc=acme,dc=com" LDAP entry in your LDAP tree.

An LDAP directory is analogous to a tree. Nodes in this tree are called LDAP "entries" and may represent users, groups, organizational units, domain controllers, or other objects. The attributes in each entry are determined by the LDAP schema. In this tutorial we will build entries based on the InetOrgPerson schema (which ships with OpenLDAP by default).

In order to build our LDAP tree we must first create the root entry. Root entries are usually a special type of entry called a domain controller (DC). Because we are assuming that the organization is called Acme Corporation, and that the domain is "acme.com," we will create a domain controller LDAP entry called `dc=acme,dc=com`. Again, you will need to replace "acme" with your organization's domain name. Also note that `dc=acme,dc=com` is what is called an LDAP distinguished name (DN). An LDAP distinguished name uniquely identifies an LDAP entry.

Do the following:

a. Create a file called `acme.ldif`. (You can delete this file once its content has been added to LDAP, so in this example, we will create it in the `/tmp` folder.)

```
[root]# cd /tmp
[root]# vi acme.ldif
```

b. Add the following lines in `acme.ldif`:

```
dn: dc=acme,dc=com
objectClass: dcObject
objectClass: organization
dc: acme
o : acme
```

c. Now add the contents of this file to LDAP. Run this command:

```
[root]# ldapadd -f acme.ldif -D cn=Manager,dc=acme,dc=com -w p@ssw0rd
```

d. Verify that your entry was added correctly.

```
[root]# ldapsearch -x -LLL -b dc=acme,dc=com
dn: dc=acme,dc=com
objectClass: dcObject
objectClass: organization
dc: acme
o: acme
```

9. Run the following:

```
[root]# sudo iptables -L
[root]# sudo service iptables save
```

10. By default, the CentOS 6 firewall will block external requests to OpenLDAP. In order to allow MWS to access LDAP, you will have to configure your firewall to allow connections on port 389. (Port 389 is the default LDAP port.)

Configuring your firewall is beyond the scope of this tutorial; however, it may be helpful to know that the default firewall on CentOS is a service called `iptables`. For more information, see the documentation on iptables (http://wiki.centos.org/HowTos/Network/IPTables). In the most basic case, you may be able to add a rule to your firewall that accepts connections to port 389 by doing the following:

a. Edit your `iptables` file:

```
[root]# vi /etc/sysconfig/iptables
```

b. Add the following line *after* all the **ACCEPT** lines but *before* any of the **REJECT** lines in your `iptables` file:

```
# ... lines with ACCEPT should be above
-A INPUT -p tcp --dport 389 -j ACCEPT
# .. lines with REJECT should be below
```

For example, here is a sample `iptables` file with this line added:

```
*filter
:INPUT ACCEPT [0:0]
:FORWARD ACCEPT [0:0]
:OUTPUT ACCEPT [0:0]
-A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
-A INPUT -p icmp -j ACCEPT
-A INPUT -i lo -j ACCEPT
-A INPUT -m state --state NEW -m tcp -p tcp --dport 22 -j ACCEPT
-A INPUT -p tcp --dport 389 -j ACCEPT
-A INPUT -j REJECT --reject-with icmp-host-prohibited
-A FORWARD -j REJECT --reject-with icmp-host-prohibited

COMMIT
```

c. Now reload `iptables`.

```
[root]# service iptables reload
```

Although providing instructions is beyond the scope of this tutorial, it is also highly recommended that you set up OpenLDAP to use SSL or TLS security to prevent passwords and other sensitive data from being sent in plain text. For information on how to do this, see the OpenLDAP TLS documentation (http://www.openldap.org/doc/admin24/tls.html).

Now that you have installed and set up Open LDAP, you are ready to add organizational units. See Adding an Organizational Unit (OU) on page 101.

Adding an Organizational Unit (OU)

These instructions will describe how to populate the LDAP tree with organizational units (OUs), groups, and users, all of which are different types of LDAP entries. The examples that follow also presume an InetOrgPerson

schema, because the InetOrgPerson schema is delivered with OpenLDAP by default.

To add an organizational unit (OU) entry to the LDAP tree

In this example, we are going to add an OU called "Users".

1. Create a temporary file called `users.ldif`. (You can delete this file once its content has been added to LDAP, so in this example, we will create it in the `/tmp` folder.)

```
[root]# cd /tmp
[root]# vi users.ldif
```

2. Add these lines to `users.ldif`:

```
dn: ou=Users,dc=acme,dc=com
objectClass: organizationalUnit
ou: Users
```

3. Add the contents of `users.ldif` file to LDAP.

```
[root]# ldapadd -f users.ldif -D cn=Manager,dc=acme,dc=com -w p@ssw0rd
```

Adding a User

To add a user to LDAP

In this example, we will add a user named "Bob Jones" to LDAP inside the "Users" OU.

1. Create a temporary file called `bob.ldif`. (You can delete this file once its content has been added to LDAP, so in this example, we will create it in the `/tmp` folder.)

```
[root]# cd /tmp
[root]# vi bob.ldif
```

2. Add these lines to `bob.ldif`:

```
dn: cn=Bob Jones,ou=Users,dc=acme,dc=com
cn: Bob Jones
sn: Jones
objectClass: inetOrgPerson
userPassword: p@ssw0rd
uid: bjones
```

3. Add the contents of `bob.ldif` file to LDAP.

```
[root]# ldapadd -f bob.ldif -D cn=Manager,dc=acme,dc=com -w p@ssw0rd
```

Adding a Group

To add a group to LDAP

In this example, we will add a group called "Engineering" to LDAP inside the "Users" OU.

1. Create a temporary file called `engineering.ldif`. (You can delete this file once its content has been added to LDAP, so in this example, we will create it in the `/tmp` folder.)

```
[root]# cd /tmp
[root]# vi engineering.ldif
```

2. Add these lines to `engineering.ldif`:

```
dn: cn=Engineering,ou=Users,dc=acme,dc=com
cn: Engineering
objectClass: groupOfNames
member: cn=Bob Jones,ou=Users,dc=acme,dc=com
```

3. Add the contents of `engineering.ldif` file to LDAP.

```
[root]# ldapadd -f engineering.ldif -D cn=Manager,dc=acme,dc=com -w p@ssw0rd
```

Adding a User to a Group

To add a user to an LDAP group

In this example, we will add an LDAP member named "Al Smith" to the "Engineering" LDAP group. This example assumes that user, Al Smith, has already been added to LDAP.

1. Create a temporary file called `addUserToGroup.ldif`. (You can delete this file once its content has been added to LDAP, so in this example, we will create it in the `/tmp` folder.)

```
[root]# cd /tmp
[root]# vi addUserToGroup.ldif
```

2. Add these lines to `addUserToGroup.ldif`:

```
dn: cn=Engineering,ou=Users,dc=acme,dc=com
changetype: modify
add: member
member: cn=Al Smith,ou=Users,dc=acme,dc=com
```

3. Now add the contents of `addUserToGroup.ldif` file to LDAP.

```
[root]# ldapadd -f addUserToGroup.ldif -D cn=Manager,dc=acme,dc=com -w p@ssw0rd
```

Trusting Servers in Java

In this topic:

Prerequisites

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

```
[root]# source /etc/tomcat6/tomcat6.conf
```

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

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 ldaps 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.

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 command:

```
[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've entered the keystore password, you'll 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.

Upgrade

In this section:

Upgrading the Moab HPC Suite RPMs

This topic provides instructions to upgrade the Moab HPC Suite RPMs to the latest release version. It includes instructions for migrating your database schema to a new version if necessary.

If upgrading Moab Web Services from a version prior to 8.0, this upgrade removes all MWS roles and permissions and recreates the default roles. If you have modified any MWS permissions or roles, you will need to recreate them after the upgrade is complete.

Upgrade the RPM Suite

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. Please note that the same commands will work for a non-root user with the `sudo` command.

1. Shut down all Adaptive services.
   - Red Hat 6-based systems

```
[root]# service moab stop        # you can also run mschedctl -k
[root]# service tomcat6 stop
[root]# service pbs_server stop
[root]# service pbs_mom stop     # if running pbs_mom on the same host
[root]# service trqauthd stop
```

   - Red Hat 7-based systems

```
[root]# systemctl stop moab        # you can also run mschedctl -k
[root]# systemctl stop tomcat6
[root]# systemctl stop pbs_server
[root]# systemctl stop pbs_mom     # if running pbs_mom on the same host
[root]# systemctl stop trqauthd
```

2. Download the latest 9.0.0 build executable (`moab-hpc-basic-suite-<version>-<timestamp>-<OS>.tar.gz`, for example) from the Adaptive Computing website.
3. Untar the package.

```
[root]# tar xzf moab-hpc-basic-suite-<version>-<timestamp>-<OS>.tar.gz
```

4. Change directories into the root directory of the untarred directory.

> Consider reviewing the README file for additional details on using the RPM distribution tarball.

5. Install the suite repositories.

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

> The `-y` option will install with the default settings for the RPM suite.

> The installation returns the following warning line:
>
> `Warning: RPMDB altered outside of yum.`
>
> This is normal and can safely be ignored.

- The *[repository-directory]* option is the directory where you want to copy the RPMs. If no argument is given, *[repository-directory]* defaults to `/opt/adaptive-rpm-repository/rpm`. If the *[repository-directory]* already exists, RPMs will be added to the existing directory. No files are overwritten in *[repository-directory]*. A repository file is also created in `/etc/yum.repos.d/` and points to the *[repository-directory]* location.
- For ease in repository maintenance, the install script fails if Adaptive Computing RPMs are copied to different directories. If a non-default *[repository-directory]* is specified, please use the same directory for future updates.
- The script installs the `createrepo` package and its dependencies. You must answer "y" to all the questions in order for the RPM install to work. Additionally, the script installs the EPEL and 10gen repositories.

6. Merge the new `.repo` files in `/etc/yum.repos.d/` with the existing ones.

> The `install-rpm-repos.sh` script will not overwrite existing RPM, GPG key or `.repo` files. Because some `.repo` files may have changed from previous releases, some merging of the `.repo` files is necessary. The newest files will have the `.new` extension.

a. Compare older `.repo` files with the newer ones to ensure that the latest changes are reflected. In some cases, there is no change, and you can remove the new file. In most cases, however, it is safe to overwrite the old `.repo` file with the new one. For example:

```
[root]# mv /etc/yum.repos.d/AC.repo.new /etc/yum.repos.d/AC.repo
```

b. After making changes in the `/etc/yum.repos.d` directory, run the following command to update the `yum` cache.

```
[root]# yum clean all
```

7. Update the 9.0.0 suite packages.

> The Moab and MWS RPMs automatically create a backup of all relevant files. These backups are stored in `/var/tmp/backup-<rpmName>-<timestamp>.tar.gz`.
>
> If changes are detected between any existing configuration files and new configuration files, a version of the new configuration file will be saved under `<configurationFileLocation>/<fileName>.rpmnew`.

a. If you are upgrading Torque, then on the Torque Server Host, each Torque MOM Host, and each Torque Client Host (including the Moab Server Host if applicable), do the following:

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

b. If you are upgrading Moab Workload Manager, then on the Moab Server Host, do the following:

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

c. If you are upgrading Moab Accounting Manager, then on the MAM Server Host, do the following as needed:

> The MAM RPM name has changed between version 8.1 and 9.0. The RPM obsoletion process removes the old RPM and installs the new RPM separately; this results in removing the mam user and not preserving the customized configuration files. A special process must be followed when upgrading from an RPM version prior to 9.0.

- If you are upgrading MAM from an RPM version prior to 9.0, do the following:

```
for i in /opt/mam/etc/{goldd,site}.conf
do
cp -p ${i} ${i}.rpmsave
done
rpm -e --nopostun moab-hpc-accounting-manager
yum install moab-accounting-manager
for i in /opt/mam/etc/mam-*.conf
do
cp -p ${i} ${i}.rpmnew
done
\cp -f /opt/mam/etc/gold.conf.rpmsave /opt/mam/etc/mam-client.conf
\cp -f /opt/mam/etc/goldd.conf.rpmsave /opt/mam/etc/mam-server.conf
\cp -f /opt/mam/etc/goldg.conf.rpmsave /opt/mam/etc/mam-gui.conf
\cp -f /opt/mam/etc/site.conf.rpmsave /opt/mam/etc/mam-site.conf
```

- If you are upgrading MAM from an RPM version at or after 9.0, do the following:

```
[root]# yum update moab-accounting-manager*
```

d. If you are upgrading Moab Web Services, then on the MWS Server Host, do the following:

```
[root]# yum update moab-web-services*
```

8. If you use ODBC, you must upgrade to the 9.0.0 schema. See Migrating Your Database to Newer Versions of Moab in the *Moab Workload Manager Administrator Guide* for more information.
9. Adaptive Computing recommends MongoDB version 2.4.x. Support for environments using 2.0 is now deprecated and will be removed in a future release. If you are running a MongoDB version less than 2.4.x, see Upgrading from MongoDB 2.0 to 2.4.x on page 114 for instructions.
10. Upgrade the schema of the `mws` database in MongoDB.

You *must* perform this step, regardless of whether you upgraded MongoDB to version 2.4.x or not. (See previous step.)

Before updating this database, you should perform a full backup. This can be done by using the `mongodump` utility documented in the MongoDB documentation (http://www.mongodb.org/display/DOCS/Backups).

- Run the database migration script provided with MWS. (It is safe to run this script more than once. If for any reason, errors occur during the execution of the script, run it again.)

```
[root]# mongo -u mws_user mws /opt/mws/utils/db-migrate.js -p
```

> You may be prompted for the mongo password. The password can be found in the /opt/mws/etc/mws-config.groovy file under the "grails.mongo.password" key.

> Depending on the number of events and services in the system, the script may take several minutes to execute.

11. (Optional, but recommended for MWS) Upgrade to the Linux x64 RPM version of 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 OpenJDK/IcedTea, GNU Compiler for Java, and so on cannot run MWS.

    Do the following:

    a. Go to the to the Oracle Java download page (http://java.com/en/download/linux_manual.jsp).

    b. Copy the URL for the Linux x64 RPM version and run the following command:

    ```
    [root]# rpm -Uh <URL>
    ```

12. Merge the configuration files.

    > You will need to decide whether to start with the old configuration file and add newer configuration options (or vice versa). Typically it depends on the amount of customization you previously made in earlier versions. In instances where you have modified very little, you should consider using the newer configuration and merging site-specific settings from the old file into the new one. The following steps highlight important changes between the 7.2.x default configuration and the 9.0.0 default configuration. Also note that new configuration files may have auto-generated content for secret keys and default passwords—be careful to ensure that secret keys shared between components are configured correctly.

The recommended layout for the `/opt/moab/etc/` directory appears as follows:

```
[root]# ls -l /opt/moab/etc
total 29
-rw-r--r--. 1 root moab  2323 Nov 13 13:41 config.moab.pl
-rw-r--r--. 1 root moab   989 Nov 13 13:41 config.sql.pl
lrwxrwxrwx. 1 root root    14 Nov 13 15:46 moab.cfg -> moab.hpc.cfg
-rw-r--r--. 1 root moab 23500 Nov 13 15:43 moab.hpc.cfg
drwxr-xr-x. 2 root moab  4096 Nov 13 15:41 moab.d
-rw-r--r--. 1 root moab   391 Nov 13 13:41 moab.dat
-r--r--r--. 1 root root   493 Nov  6 16:14 moab.lic
-rw-------. 1 root moab   288 Nov 13 15:39 moab-private.cfg
lrwxrwxrwx. 1 root root    14 Nov 13 15:46 nami.cfg -> nami.hpc.cfg
-rw-r--r--. 1 root moab   563 Nov 13 15:43 nami.hpc.cfg
```

Do the following:

a. Merge the `/opt/moab/etc/moab-private.cfg` file. Make sure that unique items in `/opt/moab/etc/moab-private.cfg.rpmnew` are added to the existing `/opt/moab/etc/moab-private.cfg` file. Include the new MWS RM credentials if you configure MWS as a resource manager:

```
CLIENTCFG[RM:mws] USERNAME=moab-admin PASSWORD=changeme!
```

The default MWS credentials in 7.2.x were *admin:adminpw*. For releases after 7.2.x, the default credentials were changed to *moab-admin:changeme!*. Use whatever credentials you have configured in `/opt/mws/etc/mws-config.groovy`.

b. Merge customizations from `/opt/moab/etc/moab.cfg` and `/opt/moab/etc/moab.d/*` into `/opt/moab/etc/moab.hpc.cfg`.

- Although there are several ways to configure and merge changes into the `/opt/moab/etc/moab.cfg` file, the following instructions outline the recommended best practices. *Deviations from these best practices may result in unexpected behavior or added difficulty in future upgrades.*
- It is best to use the new default configuration file (`/opt/moab/etc/moab.hpc.cfg`) and merge changes from previous files into that one. You will notice that content from the `/opt/moab/etc/moab.d/` directory has been merged into `/opt/moab/etc/moab.hpc.cfg`. Ensure that custom configuration options in all files located in `/opt/moab/etc/moab.d/` directory get merged in to `/opt/moab/etc/moab.hpc.cfg`.
- You should avoid `#include` configurations.

- Although the upgrade should have created a backup of the `moab.cfg` file (in `/var/tmp/backup-<rpmName>-<timestamp>.tar.gz`), it is best to create your own backup until you can confirm the updated configuration behaves as expected.

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

- If you are upgrading from a version prior to 8.0, once the changes have been merged to `/opt/moab/etc/moab.hpc.cfg`, configure Moab to use the new file. The recommended configuration is to use a symlink called `/opt/moab/etc/moab.cfg` that points to `/opt/moab/etc/moab.hpc.cfg`.

```
[root]# ln -s /opt/moab/etc/moab.hpc.cfg /opt/moab/etc/moab.cfg
```

c. Merge the `/opt/mws/etc/mws-config.groovy` file.

- Merge the `/opt/mws/etc/mws-config.groovy.rpmnew` file with the old `/opt/mws/etc/mws-config.groovy` file by editing `/opt/mws/etc/mws-config.groovy`. (Note the addition of the "auditAppender" in the default logging configuration of `/opt/mws/etc/mws-config.groovy.rpmnew`.)

```
moab.messageQueue.port = 5570

log4j = {
  // Configure an appender for the events log.
  def eventAppender = new org.apache.log4j.rolling.RollingFileAppender(
    name: 'events', layout: pattern(conversionPattern: "%m%n"))
  def rollingPolicy = new org.apache.log4j.rolling.TimeBasedRollingPolicy(
    fileNamePattern: '/opt/mws/log/events.%d{yyyy-MM-dd}',
    activeFileName: '/opt/mws/log/events.log')
  rollingPolicy.activateOptions()
  eventAppender.setRollingPolicy(rollingPolicy)

  // Configure an appender for the audit log.
  def auditAppender = new org.apache.log4j.rolling.RollingFileAppender(
      name: 'audit',
      layout: new com.ace.mws.logging.ACPatternLayout("%j\t\t\t%c{1}
\t\t\t%m%n"))
  def auditRollingPolicy = new org.apache.log4j.rolling.TimeBasedRollingPolicy(
      fileNamePattern: '/opt/mws/log/audit.%d{yyyy-MM-dd}',
      activeFileName: '/opt/mws/log/audit.log')
  auditRollingPolicy.activateOptions()
  auditAppender.setRollingPolicy(auditRollingPolicy)

  appenders {
    rollingFile name: 'stacktrace',
        file: '/opt/mws/log/stacktrace.log',
        maxFileSize: '100MB'
    rollingFile name: 'rootLog',
        file: '/opt/mws/log/mws.log',
        maxFileSize: '100MB', //The maximum file size for a single log file
        maxBackupIndex: 10, //Retain only the 10 most recent log files, delete
older logs to save space
        ~~layout:pattern(conversionPattern: '%d %p %c %m%n'), //Configures the
output format of each log entry~~
        layout: new com.ace.mws.logging.ACPatternLayout(), //Configures the
output format of each log entry
        threshold: org.apache.log4j.Level.ERROR //Ignore any logging entries
less verbose than this threshold

    appender eventAppender
    appender auditAppender
  }

  // NOTE: This definition is a catch-all for any logger not defined below
  root {
    error 'rootLog'
  }

  // Individual logger configurations
  ...

  // Logs event information to the events log, not the rootLog
  trace additivity:false, events:'com.ace.mws.events.EventFlatFileWriter'
  // Logs audit information to the audit log, not the rootLog
  trace additivity:false, audit:'mws.audit'
                                        }
```

Additions are noted in red. Removed content is stricken out.

- Note that the **mws.suite** parameter and the **mam.*** parameters have been moved to a suite-specific file in `/opt/mws/etc/mws.d/` and do not need to exist in `/opt/mws/etc/mws-config.groovy`.
- Also note the new ***messageQueue** parameters in `/opt/mws/etc/mws-config.groovy.rpmnew`. These are required and the **moab.messageQueue.secretKey** value should match the value located in `/opt/moab/etc/moab-private.cfg`.

13. Start all Adaptive services.
    - Red Hat 6-based systems

```
[root]# service pbs_server start
[root]# service moab start
[root]# service tomcat6 start
[root]# service pbs_mom start     # if running pbs_mom on the same host
[root]# service trqauthd start
```

    - Red Hat 7-based systems

```
[root]# systemctl daemon-reload
[root]# systemctl start moab
[root]# systemctl start tomcat6
[root]# systemctl start pbs_server
[root]# systemctl start pbs_mom     # if running pbs_mom on the same host
[root]# systemctl start trqauthd
```

Upgrading from MongoDB 2.0 to 2.4.x

Adaptive Computing recommends MongoDB version 2.4.x. Support for environments using 2.0 is now deprecated and will be removed in a future release.

1. Verify you can connect to the Mongo database.
   a. Obtain the Mongo username and password.

```
[root]# grep grails.mongo /opt/mws/etc/mws-config.groovy
grails.mongo.username = "mws_user"
grails.mongo.password = "secret3"
```

   b. Using the Mongo username and password (in our example, username is "mws_user" and password is "secret3"), confirm you can log in.

```
[root]# service mongod start
[root]# mongo -u mws_user -p secret3 mws
MongoDB shell version: 2.4.12
connecting to: mws
> show collections
event
mongeez
pluginInstance
...
```

2. Refer to docs.mongodb.org for instructions on how to upgrade MongoDB. Note that you must pay close attention to the information regarding instances with auth enabled (as this is the recommended setup for Moab HPC Suite).
3. Remove version 2.0 and install 2.4

```
[root]# service mongod stop
[root]# yum remove mongo20-10gen-server mongo20-10gen
[root}# yum install mongo-10gen-server
[root]# service mongod start
```

> Note that the settings in the `/etc/mongod.conf` file were saved in `/etc/mongod.conf.rpmsave` while removing MongoDB 2.0. You may need to be restore any custom settings after MongoDB 2.4.x is installed in the new `/etc/mongod.conf` file (for example, "auth = true").

4. After upgrading from 2.0 to 2.4.x, you should verify that the MongoDB credentials were preserved.

Migrating the MAM Database from MySQL to PostgreSQL

PostgreSQL is the preferred DBMS for MAM. Customers who have already installed MySQL as the DBMS for MAM 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-mysql2pgsq for additional details.

To Migrate the MAM Database

This procedure was successfully tested on an actual customer MySQL database with millions of transactions on CentOS 6.4. 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** - Installing Moab Web Services on page 31
   - **RPM Install** - Installing Moab Web Services on page 87

3. Install the prerequisite packages.

```
[root]# yum 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 Moab Accounting Manager.

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

Chapter 4 Troubleshooting

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

In this chapter:

General Issues

This topic details some common problems and general solutions.

In this topic:

Moaberror: "cannot determine local hostname"

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

If you encounter this error when starting Moab, check the `/opt/moab/etc/moab.cfg` file to make sure a valid host is configured. For example:

```
...
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
...
```

Moaberror: "Moab will now exit due to license file not found"

```
# service moab start
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, make sure your Moab 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 2013
...
```

Other Moab issues

See Troubleshooting and System Maintenance in the *Moab Workload Manager Administrator Guide*.

Where do I change my passwords?

In this section:

Moab Super User Username and Password

The default username and password for Moab are **moab-admin** and **changeme!** (respectively).

To change the username and/or the password for the Moab super user.

1. Stop the `tomcat6` and `moab` services.

   ```
   [root]# service moab stop
   [root]# service tomcat6 stop
   ```

2. Change the respective values in the following files:

   - `/opt/mws/etc/mws-config.groovy`:

     ```
     auth.defaultUser.username = "moab-admin"
     auth.defaultUser.password = "changeme!"
     ```

   - `/opt/moab/etc/moab-private.cfg`:

```
CLIENTCFG[RM:mws] USERNAME=moab-admin PASSWORD=changeme!
```

- `/opt/moab/etc/cloud.cfg`:

```
CONFIG[default]      MWS_USERNAME=moab-admin
CONFIG[default]      MWS_PASSWORD=changeme!
```

3. Start the `tomcat6` service.

```
[root]# service tomcat6 start
```

4. Start the `moab` service.

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

MongoDB Passwords

To change the passwords for MongoDB:

1. Stop the `tomcat6` and `moab` services.

```
[root]# service moab stop
[root]# service tomcat6 stop
```

2. Change the passwords for the MongoDB accounts (i.e., **moab_user** and/or **mws_user**). See the MongoDB documentation (http://docs.mongodb.org/manual/tutorial/change-user-password/) for detailed instructions.
3. Edit the password values in the following files:
   - `/opt/moab/etc/moab-private.cfg`:

```
MONGOUSER            moab_user
MONGOPASSWORD        secret2
```

   - `/opt/mws/etc/mws-config.groovy`:

```
// MongoDB configuration.
grails.mongo.username = "mws_user"
grails.mongo.password = "secret3"
```

4. Start the `tomcat6` service.

```
[root]# service tomcat6 start
```

5. Start the `moab` service.

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

Moab Web Services Issues

This topic details some common problems and general solutions for Moab Web Services.

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/tomcat6 or $CATALINA_HOME/logs`.

> If you remove the `log4j` configuration from `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 topic:

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 separate 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 `tomcat6` service. See Installing Moab Workload Manager on page 25 (Manual Installation) or Install MongoDB on page 90 (RPM Installation) for more information on how to install and configure MongoDB.

- (*Only* relevant if MongoDB is installed on a separate hostr) **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 `tomcat6` service.

  ```
  [root]# service tomcat6 restart
  ```

- **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.

MongoDB: Out of semaphores to get db connection

To resolve this error, adjust the values of `connectionsPerHost` or `threadsAllowedToBlockForConnectionMultiplier` by adding them to `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 (http://api.mongodb.org/java/current/com/mongodb/MongoOptions.html), which contains full details on all MongoDB driver options.

> You must restart Tomcat after adding, removing, or changing **grails.mongo.options** parameters.
>
> As shipped, `mws-config.groovy` does not contain any **grails.mongo.options** parameters. To adjust their values, you need to add them to `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 `mws-config.groovy`. Just use the prefix **grails.mongo.options** as shown above.

MongoDB: Connection wait timeout after 120000 ms

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

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.

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"
```

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
```

MoabReached Maximum Number of Concurrent Client Connections

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

```
CLIENTMAXCONNECTIONS 256
```

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

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

The number **256** above may be substituted for the desired maximum number of Moab client connections.

Component Documentation

The individual components of the suite have more options and allow for more configuration than can be contained in this guide. Refer to the individual component guides for more information.

Torque

- Torque 6.0.0 Administrator Guide: HTML – PDF

Moab Workload Manager

- Moab Workload Manager 9.0.0 Administrator Guide: HTML

Moab Web Services

- Moab Web Services 9.0.0 Reference Guide: HTML