Moab HPC Suite

Installation and Configuration Guide 9.1.3 for Red Hat 6-Based Systems

May 2020

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

Scan to open online help

Contents

Welcome

Welcome to the 9.1.3 Moab HPC Suite Installation and Configuration Guide for Red Hat 6-Based Systems.

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

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

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

The 9.1.3 Moab HPC Suite contains the following components for Red Hat 6-based systems:

- Torque Resource Manager 6.1.3
- Moab Workload Manager 9.1.3
- Moab Accounting Manager 9.1.3
- Moab Web Services 9.1.3
- Moab Insight 9.1.3
- Moab Viewpoint 9.1.3
- Remote Visualization uses FastX 2.2-77.3
- Nitro 2.1.1
- Nitro Web Services 2.1.1
- Reprise License Manager 12.1.2

Before commencing the installation or upgrade, please see Chapter 1: Planning Your Installation - 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.

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

In this chapter:

1.1 Getting Started

In this topic:

1.1.1 Installation Terminology

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

- Components – The different "products" included in the Moab. 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 component (server or client) is installed.

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

1.1.2 Where to Start

You 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 1.2 Server Hardware Requirements - page 7.

2. Decide whether you will perform a Manual Installation or an RPM Installation for the various components. See 1.5 Identify the Installation Methods - page 37.

> **ⓘ** 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 1.3 Component Requirements - page 18.

4. Install the individual components on their respective host(s). See 2.2 Preparing for Manual Installation - page 44 or 3.1 About RPM Installations and Upgrades - page 166 as applicable.

5. Refer to Chapter 5: Troubleshooting - page 403 for assistance in addressing common problems during installation and configuration.

1.2 Server Hardware Requirements

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

1.2.1 Topology

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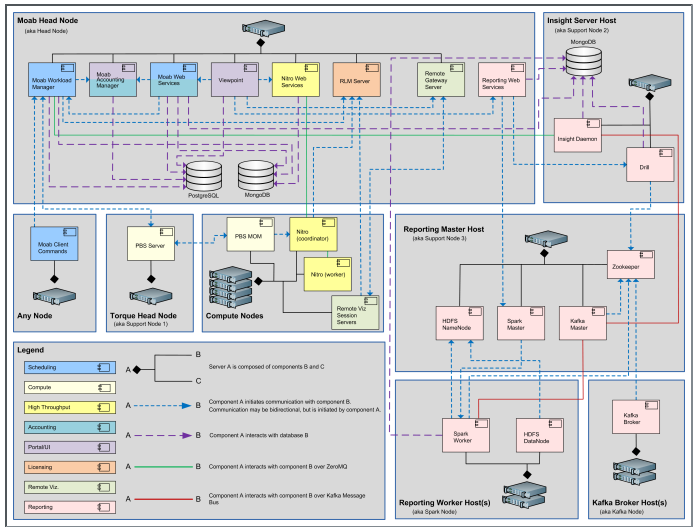
Please note the following:

- Smaller environments may elect to consolidate the Torque Server with the Moab Server on the same host, including PBS Server in the list of components installed on the same host.

- Although Moab Workload Manager and Moab Accounting Manager 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 Server Host (i.e. databases, Moab Accounting Manager, and/or Viewpoint) to their own respective servers.

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

Component	Description
Moab Workload Manager	A scheduling and management system designed for clusters and grids.
Moab Elastic Computing	Manages resource expansion and contraction of bursty workloads utilizing additional resources from private clouds or other data centers.
Torque Resource Manager - PBS Server	A resource manager for Moab. Torque provides the low-level functionality to discover and report cluster resources/features, and to start, hold, cancel, and monitor jobs. Required by Moab Workload Manager.
Torque Resource Manager - PBS MOM	Torque MOMs are agents installed on each compute node that complete tasks assigned to them by the Torque Server. When a multi-node job runs, one of the Torque MOMs is assigned the role of Mother Superior and all other nodes assigned to the job are sister nodes. Mother Superior manages the job across all the sister nodes by communicating with each of them and updating the Torque Server. Required by Torque.
Moab Passthrough	Enables job submission and monitoring with Slurm.
Slurmd	The compute node daemon of Slurm. It monitors all tasks running on the compute node, accepts work (tasks), launches tasks, and kills running tasks upon request. The Automated Installer does not install slurmd at this time. Slurmd is assumed to already be installed.
Moab Accounting Manager	An accounting management system that allows for usage tracking, charge accounting, and allocation enforcements for resource usage in technical computing environments. Required by Moab Workload Manager and Moab Web Services.
Moab Web Services (MWS)	A component of the Moab HPC Suite that enables programmatic interaction with Moab Workload Manager via a RESTful interface. MWS lets you create and interact with Moab objects and properties such as jobs, nodes, virtual machines, and reservations. MWS is the preferred method for those wishing to create custom user interfaces for Moab and is the primary method by which Moab Viewpoint communicates with Moab. Required by Moab Viewpoint.

Component	Description
Reprise License Manager Server (RLM)	A flexible and easy-to-use license manager with the power to serve enterprise users. Required by Moab Elastic Computing, Nitro, and Remote Visualization.
Moab Insight	A component of the Moab HPC Suite that collects the data that Moab emits on its message queue and stores it in a database. The message queue is efficient, can be encrypted, and tolerates disconnections and restarts on either side. Required by Moab Viewpoint and Kafka Master.
Nitro	A highly powerful, yet simple task launching solution which operates as an independent product but can also integrate seamlessly with any HPC scheduler. In the Moab HPC Suite, Nitro is fully integrated with Viewpoint for seamless high-throughput job submission and monitoring.
Nitro Web Services	Enables programmatic interaction with Nitro for obtaining Nitro job status information within Viewpoint. Required by Moab Viewpoint.
Moab Viewpoint	A rich, easy-to-use portal for end-users and administrators, designed to increase productivity through its visual web-based interface, powerful job management features, and other workload functions. The portal provides greater self-sufficiency for end-users while reducing administrator overhead in High-Performance Computing. Nitro, Remote Visualization, Elastic Computing, Moab Passthrough, and Reporting and Analytics features are also licensable for use with Viewpoint. Required by Remote Visualization.
Remote Visualization Gateway	Manages Remote Visualization sessions on the Remote Visualization Session servers. Remote Visualization is an extension of Viewpoint. Required by Viewpoint and Remote Visualization.
Remote Visualization Session	Remote Visualization sessions provide access into remote applications, rendering remotely and transfering the pixels to the local browser. Required by Viewpoint and Remote Visualization Gateway.
Reporting Web Services (RWS)	A component of Adaptive Computing Suites that enables programmatic interaction with Moab Reporting and Analytics via a RESTful interface. RWS is the preferred method for those wishing to create custom user interfaces for Moab Reporting and Analytics and is the primary method by which Moab Viewpoint communicates with Moab Reporting and Analytics.

Component	Description
Reporting and Analytics	Streams in massive amounts of workload and resource usage data from your High-Performance Computing (HPC), High-Throughput Computing (HTC) and Grid Computing environments, and then correlates that information against users, groups, and accounts, organizations so you can gain insights into exactly how your investment is being used and how well it aligns with your goals.
MongoDB	A free and open-source cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with schemas. Required by Moab Workload Manager, Moab Passthrough, Moab Web Services, Nitro Web Services, Reporting Web Services, and Spark Worker.
PostgreSQL	An object-relational database (ORDBMS) – i.e. an RDBMS, with additional (optional use) object features – with an emphasis on extensibility and standards compliance. Required by Moab Workload Manager, Moab Passthrough, Moab Accounting Manager, Moab Web Services, and Moab Viewpoint.
Drill	Apache Drill is an open-source software framework that supports data-intensive distributed applications for interactive analysis of large-scale datasets. Required by Reporting Web Services.
Hadoop	The Apache Hadoop software library is a framework that allows for the distributed processing of large data sets across clusters of computers using simple programming models. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly available service on top of a cluster of computers, each of which may be prone to failures. Required by Spark Worker.
Spark Master	Apache Spark is a fast and general engine for large-scale data processing. Spark Streaming is an extension of the core Spark API that enables scalable, high-throughput, fault-tolerant stream processing of live data streams. The Spark Master uses one or more Spark Workers when processing live data streams. Data can be ingested from many sources like Kafka, Flume, Kinesis, or TCP sockets, and can be processed using complex algorithms expressed with high-level functions like map, reduce, join and window. Finally, processed data can be pushed out to filesystems, databases, and live dashboards. Required by Reporting Web Services.
Spark Worker	The Spark Worker is used by a Spark Master when processing live data streams. Required by Spark Master.
Kafka Master	Apache Kafka is used for building real-time data pipelines and streaming apps. It is horizontally scalable, fault-tolerant, wicked fast, and runs in production in thousands of companies. Kafka Master uses one or more Kafka Brokers when pipelining and processing live data streams. Required by Spark Worker, and Insight.

Component	Description
Kafka Broker	Kafka Broker is used by a Kafka Master to pipeline and process live data streams. Apache Kafka is used for building real-time data pipelines and streaming apps. It is horizontally scalable, fault-tolerant, wicked fast, and runs in production in thousands of companies. Required by Kafka Master.

1.2.2 Hardware Requirements

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

1.2.2.A Moab and Torque Requirements

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

> ℹ️ Software requirements are listed per-component rather than suite-wide as the suite components reside on different hosts. See 1.3 Component Requirements - page 18

Environment Type	# of Compute Nodes	Jobs/ Week	Minimum Requirements (per Host Distribution)	Recommended Requirements (targeting minimum number of hosts)
Proof of Concept / Small Demo	50	<1k	**Moab Server- +Torque Server Host** - 4 Intel/AMD x86-64 cores - At least 8 GB RAM - At least 100 GB dedicated disk space **Insight Server Host** - 8 Intel/AMD x86-64 cores - At least 16 GB RAM - At least 512 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 Server +Torque Server Host** • 8 Intel/AMD x86-64 cores • At least 16 GB RAM • At least 512 GB dedicated disk space **Insight Server Host** • 8 Intel/AMD x86-64 cores • At least 16 GB of RAM • At least 1024 GB disk	**Moab Server+Torque Server Host** • 16 Intel/AMD x86-64 cores • At least 32 GB RAM • At least 1 TB dedicated disk space **Insight Server Host** • 8 Intel/AMD x86-64 cores • At least 16 GB of RAM dedicated 1 Gbit channel between Insight and Moab • 128 GB local SSD for swap • At least 1024 GB disk

Environment Type	# of Compute Nodes	Jobs/ Week	Minimum Requirements (per Host Distribution)	Recommended Requirements (targeting minimum number of hosts)
Medium with High Throughput or Larger	>500	>100k	**Moab Server Host** • 8 Intel/AMD x86-64 cores • At least 16 GB RAM • At least 512 GB dedicated disk space **Torque Server Host** • 8 Intel/AMD x86-64 cores • At least 16 GB RAM • At least 512 GB dedicated disk space **Insight Server Host** • 8 Intel/AMD x86-64 cores • At least 16 GB of RAM • At least 2048 GB disk	The Moab Server should *not* reside on the same host as the Torque Server. MWS Server *must* reside on the same host as the Moab Server (Moab Server Host). The MAM Server 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 Server Host (preferred), or another server's host (except for the Insight Server Host). Databases may also reside on the same or a different host from 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 different hosts where possible for better results; especially when High

Throughput is enabled.

- Although many factors may have an impact on performance (network bandwidth, intended use and configuration, etc.), we consider High Throughput as something that makes a significant enough difference between minimum and recommended hardware requirements to merit mention in the table above.

- Moab and Torque are both multi-threaded and perform better with more processors.

- Due to the large amount of data 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.

1.2.2.B Reporting Framework Requirements

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

Environment Type	Minimum Requirements (per Host Distribution)	Recommended Requirements (targeting minimum number of hosts)
Proof of Concept / Small Demo	**Reporting Master Host** - 4 Intel/AMD x86-64 cores - At least 8 GB RAM - At least 512 GB dedicated disk space **Reporting Worker Host** - 8 Intel/AMD x86-64 cores - At least 16 GB RAM - At least 512 GB dedicated disk space **Kafka Broker Host** - 4 Intel/AMD x86-64 cores - At least 6 GB RAM - At least 512 GB dedicated disk space	Same as minimum

Environment Type	Minimum Requirements (per Host Distribution)	Recommended Requirements (targeting minimum number of hosts)
Medium	**Reporting Master Host** • 4 Intel/AMD x86-64 cores • At least 8 GB RAM • At least 1024 GB dedicated disk space **Reporting Worker Host** • 8 Intel/AMD x86-64 cores • At least 16 GB RAM • At least 512 GB dedicated disk space **Kafka Broker Host** • 4 Intel/AMD x86-64 cores • At least 6 GB RAM • At least 1024 GB dedicated disk space	**Reporting Master Host** • 4 Intel/AMD x86-64 cores • At least 16 GB RAM • At least 1024 GB dedicated disk space **Reporting Worker Host** • 8 Intel/AMD x86-64 cores • At least 32 GB RAM • At least 512 GB dedicated disk space **Kafka Broker Host** • 4 Intel/AMD x86-64 cores • At least 6 GB RAM • At least 1024 GB dedicated disk space
Medium with High Throughput or Larger	**Reporting Master Host** • 4 Intel/AMD x86-64 cores • At least 16 GB RAM • At least 2048 GB dedicated disk space **Reporting Worker Host** • 8 Intel/AMD x86-64 cores • At least 32 GB RAM • At least 512 GB dedicated disk space **Kafka Broker Host** • 4 Intel/AMD x86-64 cores • At least 6 GB RAM • At least 2048 GB dedicated disk space	More than one Reporting Worker hosts is recommended.

1.2.3 Cray Systems

For installing Moab HPC Suite on a Cray system, refer to the Installation Notes for Moab and Torque for Cray in Appendix G of the *Moab Workload ManagerAdministrator Guide*.

1.2.4 Virtual Machines and Containers

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

1.3 Component Requirements

This topic provides the various software requirements and dependencies for the suite components (servers) for Red Hat 6-based systems.

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

1.3.1 Torque

⚠️ If you intend to use Torque 6.1 with Moab Workload Manager, you must run Moab version 8.0 or later. However, some Torque functionality may not be available. See Compatibility Requirements in the Moab HPC Suite Release Notes for more information.

1.3.1.A Supported Operating Systems

- CentOS 6, 7
- RHEL 6, 7
- Scientific Linux 6, 7
- SUSE Linux Enterprise Server 12, 12-SP1

1.3.1.B 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

  cgroups are supported and cpusets are handled by the cgroup cpuset subsystem.

  > ℹ️ It is recommended that you use --enable-cgroups instead of --enable-cpuset. --enable-cpuset is deprecated and no new features will be added to it.

  - boost version: 1.41 or later
  - libcgroup version: Red Hat-based systems must use libcgroup version 0.40.rc1-16.el6 or later; SUSE-based systems need to use a comparative libcgroup version.
  - libhwloc version: 1.9.1 is the minimum supported, however NVIDIA K80 requires libhwloc 1.11.0. Instructions for installing hwloc are provided as part of the Torque Resource Manager install or upgrade instructions.
- 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

> ℹ️ 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 1.1 Customizing the Install in the *Torque Resource Manager Administrator Guide* for more information.

1.3.2 Moab Workload Manager

1.3.2.A Supported Operating Systems

- CentOS 6, 7
- RHEL 6, 7
- Scientific Linux 6, 7
- SUSE Linux Enterprise Server 12, 12-SP1

> ⚠️ A SUSE 11-based OS is *only* supported for Moab Server if your configuration does *not* include Elastic Computing or Moab Web Services (MWS).

1.3.2.B 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)

1.3.2.C Supported Resource Managers

- Torque 4.2.9 or later
- Slurm

1.3.3 Moab Accounting Manager

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

1.3.3.A Supported Operating Systems

- CentOS 6, 7
- RHEL 6, 7
- Scientific Linux 6, 7
- SUSE Linux Enterprise Server 12, 12-SP1

1.3.3.B Software Requirements

- gcc
- perl-suidperl
- httpd
- mod_ssl
- rrdtool
- Moab Workload Manager 9.1.3
- Perl modules; see 2.5 Installing Moab Accounting Manager - page 62 (Manual Installation) 3.9 Installing Moab Accounting Manager - page 187 (RPM Installation) for more details

1.3.3.C Depends On (not necessarily on the same host)

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

- PostgreSQL 7.2 or later

1.3.4 Moab Web Services

⚠ MWS Server *must* reside same host as Moab Server (Moab Server Host).

1.3.4.A Supported Operating Systems

- CentOS 6, 7
- RHEL 6, 7
- Scientific Linux 6, 7
- SUSE Linux Enterprise Server 12, 12-SP1

1.3.4.B Software Requirements

- Moab Workload Manager 9.1.3
- Oracle® Java® 8 Runtime Environment

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

- Apache Tomcat™ 7, 8

1.3.4.C Depends On (not necessarily on the same host)

- LDAP *or* PAM; see 2.6 Installing Moab Web Services - page 75 (Manual Installation) 3.10 Installing Moab Web Services - page 198 (RPM Installation) for more details
- MongoDB® 3.2.x

1.3.5 Moab Insight

> ⓘ Only an RPM-based installation is supported for installing Moab Insight.

1.3.5.A Supported Operating Systems

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

1.3.5.B Software Requirements

- Oracle® Java® 8 Runtime Environment

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

1.3.5.C Depends On

- Moab Workload Manager 9.1.3

  > ⚠️ Moab Workload Manager and Insight both tend to heavily consume system resources. It is strongly recommended that the Insight Server and the Moab Server must run on *different* hosts.

- MongoDB 3.2.x

  > ⚠️ It is strongly recommended that the Insight MongoDB reside on the Insight Server Host.

1.3.5.D Performance Benchmarks

Adaptive Computing has tested and certified Insight's scale and performance under the following server configuration and load scenarios.

Server Configuration

Host hardware: 8 core AMD Opteron 6320 2.8 GHz servers, with 32GB of RAM and a 500GB WD Blue hard drive

Installed services: Moab Workload Manager, Moab Web Services, Moab Insight, Moab Viewpoint (all at version 9.0.0 and running on the same host)

> ℹ️ The benchmarks were ran with multiple services on a single host to benchmark Insight under very aggressive working conditions. Moab Insight must be installed on its own host.

Load Scenarios

Jobs in queue	Avg Job Duration	Avg job Size (ppn)	Number of Nodes	Procs per Node	Avg Jobs per Week
1000	200	32	500	32	25200
1000	60	32	500	32	84000
1000	10	32	500	32	504000
1000	200	16	6384	16	321754
1000	60	16	6384	16	1072512
1000	10	16	6384	16	6435072
10000	200	32	500	32	25200
10000	60	32	500	32	84000
10000	10	32	500	32	504000
10000	200	16	6384	16	321754
10000	60	16	6384	16	1072512
25000	200	32	500	32	25200
25000	60	32	500	32	84000
25000	10	32	500	32	504000

1.3.6 Moab Viewpoint

> ℹ Only an RPM-based installation is supported for installing Moab Viewpoint.

1.3.6.A Supported Operating Systems

- CentOS 6, 7

- RHEL 6, 7

- Scientific Linux 6, 7

> There is a known issue in RHEL/CentOS 7.2.1511 and 7.3.1611 where the pip generates a Traceback in the pyparsing package.
>
> ```
> Traceback (most recent call last):
>   File "/usr/bin/pip", line 5, in <module>
>     from pkg_resources import load_entry_point
>   File "/usr/lib/python2.7/site-packages/pkg_resources/__init__.py", line 72,
> in <module>
>     import packaging.requirements
>   File "/usr/lib/python2.7/site-packages/packaging/requirements.py", line 59,
> in <module>
>     MARKER_EXPR = originalTextFor(MARKER_EXPR())("marker")
> TypeError: __call__() takes exactly 2 arguments (1 given)
> ```
>
> To avoid this issue, check the version of pyparsing that is installed. Install pyparsing 2.0.3 or later. Either run 'yum update', 'yum update pyparsing', or locate and install pyparsing 2.0.3 or later.

1.3.6.B Software Requirements

> The new user interface was built on Django, a forward-thinking web framework, which relies heavily on Python; thus, HPC administrators should install Viewpoint only on systems with standard system level Python installed. The system you select for Viewpoint should not have any modifications made to its default Python installation.

- httpd

- mod_wsgi

- python-anyjson

- python-crypto2.6

- python-httplib2

- python-mako

- python-markupsafe

- python-paramiko

- python-pip

- python-requests

- python-simplejson

- python-six

- python-unittest2

1.3.6.C Depends On (not necessarily on the same host)

- Moab Web Services 9.1.3

- Moab Insight 9.1.3

1.3.6.D Supported Browsers

- Mozilla Firefox 25+

- Internet Explorer 10+

- Chrome 35+

1.3.7 RLM Server

Moab's Elastic Computing Feature, Viewpoint's Remote Visualization Feature, and Nitro require access to a centralized Reprise License Manager (RLM) server.

Adaptive Computing *strongly* recommends that your RLM Server is version 12.1.2.

This server is not load-extensive so it may be installed on any host within your Moab HPC Suite environment. It may also be installed on its own host.

> ℹ If your company already utilizes an RLM Server, you do not have to install another as long as the Moab HPC Suite components can access it.

> ⚠ The host on which you install RLM Server must always be on and should have High Availability (uptime).

1.3.7.A Supported Operating Systems

- CentOS 6, 7

- RHEL 6, 7

- Scientific Linux 6, 7

- SUSE Linux Enterprise Server 12

1.3.8 Remote Visualization

⚠️ Remote Visualization comes packaged with FastX 2.2. FastX 2.2 requires reverse DNS to be set up on your network in order for the Gateway Server and Session Servers to resolve each other's IP addresses and hostnames. Without it, Session Servers will not be able to register correctly with the Gateway Server and authentication to the Gateway Server will fail.

ℹ️ Only an RPM-based installation is supported for installing Remote Visualization.

1.3.8.A Supported Operating Systems

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

1.3.8.B License Requirements

Remote Visualization requires access to a centralized Reprise License Manager (RLM) server. See 1.3.7 RLM Server - page 27 for more information.

1.3.8.C Software Requirements

The following software packages are also required. The installation of these packages are included in the Install Remote Visualization procedure.

The following software packages are also required. The installation of these packages are included in the Install Remote Visualization procedure.

- ImageMagick
- ImageMagick-perl
- perl-Crypt-SSLeay
- perl-X11-Protocol

ℹ️ The installation of these packages is included in the Install Remote Visualization procedure.

In addition, *each* Session Server must include the graphical applications (resources) you will have Moab schedule. For example, desktop (gnome-session), xterm, firefox, chrome.

1.3.8.D Depends On (not on the same host)

- Torque Resource Manager 6.1.3

- Moab Workload Manager 9.1.3

- Moab Web Services 9.1.3

- Moab Insight 9.1.3

- Moab Viewpoint 9.1.3

1.3.9 Nitro

ⓘ When integrated with the Moab HPC Suite, Nitro resides on the Torque compute nodes.

1.3.9.A Hardware Requirements

- Nitro requires one or more multi-core processors per host. Generally the more processors (sockets) and/or OS cores a host has, the more tasks Nitro can execute simultaneously on each host; although this will be application-dependent.

- It is recommended that hosts should have sufficient memory to execute as many applications as possible so that Nitro can run them at a rate of one application instance per OS core (especially if they are not multi-threaded). This eliminates the need for users to have to request memory in their Nitro task definitions.

ⓘ See the *Nitro Administrator Guide* for information on specifying memory requirements.

1.3.9.B Supported Operating Systems

- CentOS 6, 7

- Red Hat 6, 7

- Scientific Linux 6, 7

- SUSE Linux Enterprise Server 12, 12-SP1

1.3.9.C License Requirements

Nitro requires access to a centralized Reprise License Manager (RLM) server. See for more information.

1.3.9.D Software Requirements

Nitro is built with all needed libraries statically linked. This provides for a quick and simple install-ation and helps avoid troublesome library mismatches. No additional packages need to be installed on the compute nodes.

However, users running nitrosub and/or the nitrostat utility require Python 2.6.6 or later on the system from which they are running it.

1.3.10 Nitro Web Services

> ℹ Nitro Web Services is commonly installed on the Moab Server Host.

1.3.10.A Supported Operating Systems

- CentOS 6, 7
- Red Hat 6, 7
- Scientific Linux 6, 7
- SUSE Linux Enterprise Server 11

1.3.10.B Depends On (not necessarily on the same host)

- Nitro 2.1.1 – Installed on Torque compute nodes
- Viewpoint 9.1.3
- MongoDB 3.2.x

1.3.11 Reporting Framework

1.3.11.A Supported Operating Systems

- CentOS 6, 7
- Red Hat 6, 7
- Scientific Linux 6, 7
- SUSE Linux Enterprise Server 12, 12-SP1

1.3.11.B Software Requirements

- Oracle® Java® 8 Runtime Environment

  > ⓘ Oracle Java 8 Runtime Environment is the recommended Java environment. Other versions of Java, including Java 9, OpenJDK/IcedTea, GNU Compiler for Java, are not officially supported with the Reporting Framework.

- Drill 1.8.0
- Hadoop 2.6.4
- Kafka 2.11-0.10.0.1
- Spark 2.1.2
- Zookeeper 3.4.6

1.3.11.C Depends On (not necessarily on the same host)

- Moab Insight 9.1.3
- MongoDB® 3.2.x

1.4 RLM Licensing Model

The Moab HPC Suite uses Reprise License Manager (RLM) to enforce nodelocked and floating licenses.

1.4.1 Definitions

Term	Definition
Nodelocked License	A nodelocked license is a license grant which allows software to be used on a particular computer, and on that computer only. A license server is not required to enforce a nodelocked license. However, a unique unmodifiable ID unique to the computer is required.
Floating License	Floating licensing is a software licensing approach in which a limited number of licenses for a software application are shared among a larger number of users/clients over time. When an authorized user wishes to run the application they request a license from a central license server. A license server is used for floating licenses and logging of usage data.
RLM HostID	The host ID of the RLM server. Adaptive Computing requests that you provide the 'ether' RLM HostID when the RLM HostID is required. RLM's `rlmhostid` command can be used to obtain the 'ether' RLM HostID. However, the `rlmhostid` command may not be available on every host (typically only available where RLM server is installed). The `rlmhostid` command takes an optional parameter `hostid type`, which defaults to `ether` when not given. When the `rlmhostid` command is unavailable, provide the Ethernet MAC address of one of the host's Network Interface Cards (NICs).

1.4.2 License File Names and Installation Locations

You will be provided license files with file names depicted below. Please install each file in the designated Destination Directory and rename the files to the Installed File Name. When issued RLM-signed licenses, technically, the only Installed File Name requirement is that the file name end in `.lic`. However, we suggest you use the Installed File Name.

> 🛈 Do not install more than one of each of the following licenses in the respective Destination Directory. Having more than one `moab-rlm--xxxxx.lic` file in `/opt/moab/etc`, for example, will cause problems. Thus, the suggested Installed File Names.

File Description	File Name	Destination Directory	Installed File Name	Host
Moab Workload Manager (version >= 9.1.0)	`moab-rlm--xxxxx.lic`	`/opt/moab/etc`	`moab-rlm.lic`	Moab Server Host
Moab Workload Manager (version < 9.1.0)	`moab_xxxx.lic moabl_ xxxx.lic moabc_ xxxx.lic`	`/opt/moab/etc`	`moab.lic`	Moab Server Host
Elastic Computing (Moab side)	`moab-rlm-elastic--xxxxx.lic`	`/opt/moab/etc`	`moab-rlm-elastic.lic`	Moab Server Host
Elastic Computing (RLM side)	`moab-rlm-elastic-track-ing--xxxxx.lic`	`/opt/rlm`	`moab-rlm-elastic-track-ing.lic`	RLM Host
Viewpoint	`moab-vp--xxxxx.lic`	`/op-t/viewpoint/etc/license.d`	`license.lic`	Viewpoint Host
Nitro	`nitro-rlm--xxxxx.lic`	`/opt/rlm/`	`nitro.lic`	RLM Host
Nitro	`nitro-rlm--xxxxx.lic`	`/opt/nitro/etc`	`nitro.lic`	Nitro Coordinator Host - All Nitro Compute Nodes

File Description	File Name	Destination Directory	Installed File Name	Host
Adaptive Computing ISV RLM	`adapt-iveco.set` `adapt-iveco.opt`	`/opt/rlm`	`adapt-iveco.set` `adapt-iveco.opt`	RLM Host
StarNet ISV RLM	`starnet.set`	`/opt/rlm`	`starnet.set`	RLM Host

1.4.3 Licenses Issued by Adaptive Computing

Instructions for obtaining licenses from Adaptive Computing are given below.

1.4.3.A Nodelocked/Uncounted Licenses

Moab License

This license is limited by the number of sockets the scheduler will schedule. The socket count includes physical sockets, GPUs, and Xeon Phis in your cluster.

Moab does not delegate license enforcement by socket to an external licensing server. Doing so allows Moab to give you visibility into all resources reported by your resource managers, while scheduling within the limits of your licensing agreement.

Please send the following to `licenses@adaptivecomputing.com` to obtain a Moab License:

- RLM Hostid of the Moab Server host
- Total number of physical sockets, GPUs, and Xeon Phis in your cluster

Install this license as `/opt/moab/etc/moab-rlm.lic` on the Moab Server host.

Viewpoint License

Please send the following to `licenses@adaptivecomputing.com` to obtain a Viewpoint License:

- RLM Hostid of the Viewpoint Server host

Install this license as `/opt/viewpoint/etc/license.d/license.lic` on the Viewpoint Server host.

1.4.3.B Floating/Counted Licenses

Remote Visualization License

Please send the following to `licenses@adaptivecomputing.com` to obtain a Remote License:

- RLM Server Hostname
- RLM Hostid of the RLM Server host
- Max number of concurrent Remote Viz sessions

Install this license as `/opt/rlm/fastx.lic` on the RLM Server host.

Elastic Computing License

> ℹ As of Moab version 9.1.2, dynamic node procs are no longer counted against the total procs listed in the Moab license. This allows you to do as many bursts as you desire without exceeding the total procs used for on-premises nodes. If your version of Moab is before 9.1.2, please contact your Adaptive Computing sales representative.

Note that the Elastic Computing feature is also part of the Moab and Viewpoint nodelocked licenses. Adaptive Computing will issue you new Moab/Viewpoint nodelocked licenses when needed.

Please send the following to licenses@adaptivecomputing.com to obtain all the necessary licenses to enable Elastic Computing on your cluster:

- RLM Server Hostname
- RLM Hostid of the RLM server

Required to generate/regenerate the Moab License:

- RLM Hostid of the Moab Server host
- Total number of physical sockets, GPUs, and Xeon Phis in your cluster

If using Viewpoint, required to generate/regenerate the Viewpoint License:

- RLM Hostid of the Viewpoint Server host

Three Moab licenses will be generated:

1. Moab license - A nodelocked license with the Elastic Computing feature enabled
2. Moab Elastic license - A file that tells Moab where the RLM server is located. RLM will be used to track Elastic Computing usage statistics.
3. Moab Elastic Tracking license - A file that tells RLM to allow unlimited Elastic Computing usage and to track when Elastic nodes are added and removed from the cluster.

On the Moab Server Host:

- Install the Moab nodelocked license as `/opt/moab/etc/moab-rlm.lic`.

- Install the Moab Elastic license as `/opt/moab/etc/moab-elastic.lic`.

On the RLM Server Host:

- Install the Elastic Tracking licenses as `/opt/moab/etc/moab-elastic-tracking.lic`.

If using Viewpoint, on the Viewpoint Server Host:

- Install the Viewpoint nodelocked license as `/opt/viewpoint/etc/license.d/license.lic`.

Nitro License

Nitro can be licensed based on number of concurrent Nitro Workers or number of cores.

Please send the following to `licenses@adaptivecomputing.com` to obtain a Nitro License:

- RLM Server Hostname

- RLM Hostid of the RLM server

- Number of Nitro Workers or cores.

1.5 Identify the Installation Methods

Adaptive Computing provides different methods for installing the Moab components, Manual Installation, RPM Installation or the new Automated Installation (uses RPM methodology).

Depending on your environment and which components you are installing (and on which host), you may need to use a combination of Manual Installation and RPM Installation. However, the automated installer is only available for systems that support RPM installations. See for more information on the automated installer.

In this topic:

1.5.1 Manual Installation

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

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

See Chapter 2: Manual Installation - page 39 for more information on the Manual Installation method.

1.5.2 RPM Installation

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

- Whether you are installing RPMs on one host or on several hosts, each host must have the Adaptive Computing Package Repository enabled. See 3.3 Preparing for RPM Installs - page 169 for more information.
- Some customization options are available for Moab Workload Manager and Moab Accounting Manager by building custom RPMs. See 2.4.5 (Optional) Build a Custom RPM - page 57 for Moab Workload Manager and 2.5.5 (Optional) Build a Custom RPM - page 66 for Moab Accounting Manager.

1.5.3 Automated Installation

This method provides advantages for systems who do not want the complexity of the Manual Installation or a RPM Typical or Offline Installation.

- This method leverages the RPM functionality.

- This method requires you to answer some configuration questions (for example, how many servers, which Moab HPC Suite products) and then launches the installation across all the hosts in your system in less than an hour.

See Chapter 4: Automated Installation Method - page 367 for more information on the Automated Installation method.

<div style="background:#1f4e79;color:white">

Chapter 2: Manual Installation

</div>

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

Be aware of the following:

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

- Manual Installation is not available for Insight, Viewpoint, or Remote Visualization.

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

Related Topics

-
-

2.1 Manual Installation

This section provides instructions and other information for installing your Moab components for Red Hat 6-based systems using the Manual installation method.

2.2 Preparing for Manual Installation

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

> ℹ Many individual components have dependencies on other components (see Chapter 1: Planning Your Installation - page 3). However, if you do not require a certain component, you do not have to install it.

The install instructions for each component include information about system requirements and dependencies. Some include 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 install process.

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

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

2.2.1 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>
```

If your site uses an external repository to install python dependencies (for example, the host where you install Viewpoint might need to download extra packages), you will need to set up pip to use a proxy. Do the following:

```
export http_proxy=http://<proxy_server_id>:<port>
```

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

2.2.2 Enable Extra Packages for the Repository

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

Do the following:

- On non-RHEL Red Hat-based systems (e.g. CentOS and Scientific Linux), you will need to install the epel release package in order to have access to required rpm package dependencies.

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

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

2.2.3 Update Your System Software to the Latest Version

It is recommended that you update your system software to the latest version before installing Moab HPC Suite components.

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

```
[root]# yum update
```

2.2.4 Ensure Hostname Resolution for all Hosts

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

2.2.5 Install the Moab HPC Suite Software Components

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

1. Torque. See 2.3 Installing Torque Resource Manager - page 47.

2. Moab Workload Manager. See 2.4 Installing Moab Workload Manager - page 54.

3. Moab Accounting Manager. See 2.5 Installing Moab Accounting Manager - page 62.

4. Moab Web Services. See 2.6 Installing Moab Web Services - page 75.

5. Moab Insight (RPM install method only). See 3.11 Installing Moab Insight - page 206.

6. Moab Viewpoint (RPM install method only). See 3.12 Installing Moab Viewpoint - page 214.

7. RLM Server. See 2.7 Installing RLM Server - page 86.

8. Remote Visualization (RPM install method only). See 3.15 Installing Remote Visualization - page 257

9. Integrate Nitro with your Moab HPC Suite. See 2.8 Nitro Integration - page 89.

2.3 Installing Torque Resource Manager

> ⚠️ If you intend to use Torque Resource Manager 6.1.3 with Moab Workload Manager, you must run Moab version 8.0 or later. However, some Torque functionality may not be available. See "Compatibility Requirements" in the *Moab HPC Suite Release Notes* for more information.

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

> ℹ️ For Cray systems, Adaptive Computing recommends that you install Moab and Torque Servers (head nodes) on commodity hardware (*not* on Cray compute/service/login nodes).
>
> *However*, you *must* install the Torque pbs_mom daemon and Torque client commands on Cray login and "mom" service nodes since the pbs_mom *must* run on a Cray service node within the Cray system so it has access to the Cray ALPS subsystem.
>
> See Installation Notes for Moab and Torque for Cray" in the *Moab Workload Manager Administrator Guide* for instructions on installing Moab and Torque on a non-Cray server.

In this topic:

2.3.1 Open Necessary Ports

Torque requires certain ports to be open for essential communication.

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

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

If using the MOM hierarchy (documented in 1.1 Setting Up the MOM Hierarchy (Optional) in the *Torque Resource ManagerAdministrator Guide*) you must also open port 15003 from the server to the nodes.

See also:

- for general instructions and an example of how to open ports in the firewall.
- "Configuring Ports" in the *Torque Resource ManagerAdministrator Guide* for more information on how to configure the ports that Torque uses for communication.

2.3.2 Install Dependencies, Packages, or Clients

2.3.2.A Install Packages

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

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

2.3.2.B Install hwloc

⚠ Using "yum install hwloc" may install an older, non-supported version.

When cgroups are enabled (recommended), hwloc version 1.9.1 or later is required. NVIDIA K80 requires libhwloc 1.11.0. If cgroups are to be enabled, check the Torque Server Host to see if the required version of hwloc is installed. You can check the version number by running the following command:

```
[root]# hwloc-info --version
```

The following instructions are for installing version 1.9.1.

If hwloc is not installed or needs to be upgraded to the required version, do the following:

1. On the Torque Server Host, *each* Torque MOM Host, and *each* Torque Client Host, do the following:

   a. Download hwloc-1.9.1.tar.gz from https://www.open-mpi.org/software/hwloc/v1.9.

   b. Run each of the following commands in order.

   ```
   [root]# yum install gcc make
   [root]# tar -xzvf hwloc-1.9.1.tar.gz
   [root]# cd hwloc-1.9.1
   [root]# ./configure
   [root]# make
   [root]# make install
   ```

2. Run the following commands on the Torque Server Host *only*.

   ```
   [root]# echo /usr/local/lib >/etc/ld.so.conf.d/hwloc.conf
   [root]# ldconfig
   ```

2.3.3 Install Torque Server

> You `must` complete the tasks to install the dependencies, packages, or clients before installing Torque Server. See 2.3.2 Install Dependencies, Packages, or Clients - page 48.
>
> If your configuration uses firewalls, you *must also* open the necessary ports before installing the Torque Server. See 2.3.1 Open Necessary Ports - page 47.

On the Torque Server Host, do the following:

1. Download the latest 6.1.3 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:
   > ```
   > [root]# yum install git
   > ```

   ```
   [root]# git clone https://github.com/adaptivecomputing/torque.git -b 6.1.3 6.1.3
   [root]# cd 6.1.3
   [root]# ./autogen.sh
   ```

   - Download the latest Torque build from Adaptive Computing Torque Downloads.

```
[root]# tar -xzvf torque-6.1.3.tar.gz
[root]# cd torque-6.1.3/
```

2. Determine which ./configure command options you need to add, based on your system configuration.

    At a minimum, you add:

    - --enable-cgroups

    - --with-hwloc-path=/usr/local See 1.3.1 Torque - page 19 for more information.

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

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

3. Run each of the following commands in order.

```
[root]# ./configure --enable-cgroups --with-hwloc-path=/usr/local # add any other
specified options
[root]# make
[root]# make install
```

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

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

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

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

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

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

```
[root]# chkconfig --add pbs_server
[root]# service pbs_server restart
```

2.3.4 Install Torque MOMs

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

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

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-x86_64.sh ...
   Building ./torque-package-mom-linux-x86_64.sh ...
   Building ./torque-package-server-linux-x86_64.sh ...
   Building ./torque-package-gui-linux-x86_64.sh ...
   Building ./torque-package-devel-linux-x86_64.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 MOM 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-x86_64.sh <mom-node>:
   ```

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

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

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

2. On *each* Torque MOM Host, confirm that cgroups have been mounted; if not, mount them.

   a. Run `lssubsys -am`.

   b. If the command is not found, or you do not see something similar to the following, then cgroups are *not* mounted, continue with these instructions.

   ```
   ns
   perf_event
   net_prio
   cpuset /cgroup/cpuset
   cpu /cgroup/cpu
   cpuacct /cgroup/cpuacct
   memory /cgroup/memory
   devices /cgroup/devices
   freezer /cgroup/freezer
   net_cls /cgroup/net_cls
   blkio /cgroup/blkio
   ```

    c. Install the cgroup library package and mount cgroups.

```
[root]# yum install libcgroup
[root]# service cgconfig start
[root]# chkconfig cgconfig on
```

    d. Run *lssubsys -am* again and confirm cgroups are mounted.

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

    a. Install the self-extracting MOM package.

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

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

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

2.3.5 Install Torque Clients

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

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

    a. Copy the self-extracting client package to *each* Torque Client Host.

> ℹ 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 Client Host.

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

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

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

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

    a. Install the self-extracting client package.

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

    b. Enable and start the trqauthd service.

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

2.3.6 Configure Data Management

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

Related Topics

2.4 Installing Moab Workload Manager

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

ℹ️ For Cray systems, Adaptive Computing recommends that you install Moab and Torque Servers (head nodes) on commodity hardware (*not* on Cray compute/service/login nodes).

However, you *must* install the Torque pbs_mom daemon and Torque client commands on Cray login and "mom" service nodes since the pbs_mom *must* run on a Cray service node within the Cray system so it has access to the Cray ALPS subsystem.

See Installation Notes for Moab and Torque for Cray" in the *Moab Workload Manager Administrator Guide* for instructions on installing Moab and Torque on a non-Cray server.

In this topic:

2.4.1 Understand Licenses

With the 9.1.0 release, Moab now uses an RLM Server to manage licenses. For the Moab core and for most Moab features, an RLM Server is not required. The new Moab "core" license will have a new name to reflect the RLM generation. Do *not* rename this license to moab.lic. See 1.4 RLM Licensing Model - page 32 for more information about RLM licensing.

Elastic Computing, beginning with 9.1.0, requires an RLM Server as part of your configuration.

ℹ️ The 9.1.0 licensing change does not affect legacy licenses; however, a module-based license may be required to use newer functionality.

2.4.2 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.

Location	Ports	Functions	When Needed
Moab Server Host	42559	Moab Server Port	If you intend to run client commands on a host different from the Moab Server Host *or* if you will be using Moab in a grid

See 2.13 Opening Ports in a Firewall - page 106 for general instructions and an example of how to open ports in the firewall.

2.4.3 Install Dependencies, Packages, or Clients

2.4.3.A Dependencies and Packages

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

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

2.4.3.B Torque Client

If you are using Torque and are installing the Torque Server on a different host (Torque Server Host) from the Moab Server (Moab Server Host), 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 2.3.2.B Install hwloc - page 48 and 2.3.5 Install Torque Clients - page 52 using the Moab Server Host as the Torque Client Host; with the exception that you must copy and install the torque-package-devel-linux-<arch>.sh self-extracting package in addition to the torque-package-clients-linux-<arch>.sh package.

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

2.4.4 Obtain and Install the Elastic Computing License

If using Elastic Computing, Moab requires access to an RLM license server to record usage.

> ⚠️ These instructions assume you already have access to an RLM Server. See 2.7 Installing RLM Server - page 86 for instructions on how to set up a new RLM Server.

Do the following:

1. On the RLM server, obtain the hostid and hostname.

    - hostid

    ```
    [root]# /opt/rlm/rlmhostid
    ```

    You should see output similar to the following.

    ```
    rlmhostid v12.1
    Copyright (C) 2006-2016, Reprise Software, Inc. All rights reserved.

    Hostid of this machine: 00259096f004
    ```

    - hostname

    ```
    [root]# /opt/rlm/rlmhostid host
    ```

    You should see output similar to the following.

    ```
    rlmhostid v12.1
    Copyright (C) 2006-2016, Reprise Software, Inc. All rights reserved.

    Hostid of this machine: host=<your-host-name>
    ```

2. Email licenses@adaptivecomputing.com for a license and include the hostid and hostname you just obtained.

3. Adaptive Computing will generate the license and send you the Elastic Computing license file (.lic) file in a return email.

4. On the RLM server, do the following:

    a. Download and install the license file.

    ```
    [root]# cd /opt/rlm
    [root]# chown rlm:rlm <licenseFileName>.lic
    ```

    b. If the RLM Server in your configuration uses a firewall, edit the license file to reference the ISV adaptiveco port for the Adaptive license-enabled products. This is the same port number you opened during the RLM Server installation. See the instructions to open necessary ports in the 2.7 Installing RLM Server - page 86 (manual installation method) or 3.14 Installing RLM Server - page 255 (RPM installation method) for more information.

    ```
    [root]# vi /opt/rlm/moab_elastic_tracking.lic


    ISV adaptiveco port=5135
    ```

    The license file already references the RLM Server port (5053 by default).

> ⓘ If the RLM Server in your configuration uses different ports, you will need to modify the license file to reflect the actual ports. See the instructions to open necessary ports in the 2.7 Installing RLM Server - page 86 (manual installation method) or 3.14 Installing RLM Server - page 255 (RPM installation method) for more information.

   c. If you did *not* install an RLM Server using the file available from Adaptive Computing (for example, because your system configuration already uses one), do the following:

      i. Download the 'adaptiveco.set' file from the Adaptive Computing Moab HPC Suite Download page (http://www.adaptivecomputing.com/support/download-center/moab-hpc-suite-download/).

     ii. Install the 'adaptiveco.set' file.

```
[root]# chown rlm:rlm adaptiveco.set
```

    iii. Place the 'adaptiveco.set' file in the *same* directory where the Elastic Computing license resides. Typically, this is the RLM Server base directory (/opt/rlm); but may be different depending on your configuration

   d. Perform a reread on the RLM Server base directory to update the RLM Server with your license. For example:

```
[root]# /opt/rlm/rlmreread
```

2.4.5 (Optional) Build a Custom RPM

Do the following:

1. Install `rpm-build`.

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

2. Download the latest Moab build from the Adaptive Computing Moab HPC Suite Download Center (https://www.adaptivecomputing.com/support/download-center/moab-hpc-suite-download/).

3. Untar the downloaded package.

4. Change directories into the untarred directory.

5. Edit the ./moab.spec file for RPM customization.

6. Run ./rpm-build.

7. Locate the custom RPM in rpm/RPMS/x86_64.

2.4.6 Install Moab Server

> ℹ️ You `must` complete the tasks to install the dependencies, packages, or clients before installing Moab Server. See 2.4.3 Install Dependencies, Packages, or Clients - page 55.
>
> If your configuration uses firewalls, you *must also* open the necessary ports before installing the Moab Server. See 2.4.2 Open Necessary Ports - page 55.

On the Moab Server Host, do the following:

1. Download the latest Moab build from the Adaptive Computing Moab HPC Suite Download Center (https://www.adaptivecomputing.com/support/download-center/moab-hpc-suite-download/).

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

```
[root]# tar -xzvf moab-9.1.3-<OS>.tar.gz
[root]# cd moab-9.1.3-<OS>
```

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

If Elastic Computing is part of your Moab Workload Manager configuration, install `deps/acpython-base*`.

```
[root]# yum install deps/acpython-base*
```

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

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

> ℹ️ See 2.16 Moab Workload Manager Configuration Options - page 115 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 one of the following:

a. If using Torque Resource Manager:

- 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 installed the Torque Server on a different host (Torque Server Host), configure the RMCFG HOST parameter to tell Moab the host on which Torque Server is running.

```
RMCFG[torque] HOST=<torque_server_hostname>
```

b. If using the Moab Passthrough feature for Viewpoint on Slurm, see Moab Passthrough for additional information and installation instructions.

c. If using a NATIVE resource manager, see Managing Resources Directly with the Native Interface in the *Moab Workload Manager Administrator Guide* for configuration information.

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).

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

To verify the current status of your license, run the following command:

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

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

- New RLM-Based License (version 9.1.0 or after)

```
$ moab --about | grep License
Moab Workload Manager Version 'master' License Information:
  Current License:    (moab_license) Valid Until - 15-jan-2019
  Current License:    Max Sockets    = 1000000
  Current License:    (moab_grid) Valid Until - 15-jan-2019
  Current License:    (moab_green) Valid Until - 15-jan-2019
  Current License:    (moab_provision) Valid Until - 15-jan-2019
  Current License:    (moab_vms) Valid Until - 15-jan-2019
  Current License:    Max VMs    = 1000000
  Current License:    (moab_elastic) Valid Until - 15-jan-2019
```

```
   Current License:    (moab_groupsharing) Valid Until - 15-jan-2019
   Current License:    (moab_advancedrm) Valid Until - 15-jan-2019
   Current License:    (moab_workflow) Valid Until - 15-jan-2019
   Current License:    (moab_accounting) Valid Until - 15-jan-2019
```

- Legacy License Format

```
Moab Workload Manager Version '9.1.3' License Information:
 Current License:  Max Procs   = 10000
 Current License:  Valid Until - Jul 13 19:42:10 2019
```

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

9. Start Moab.

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

2.4.7 Configure Torque to Trust Moab

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

2.4.8 Verify the Installation

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

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

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

2.4.9 (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

- 2.2 Preparing for Manual Installation - page 44

2.5 Installing Moab Accounting Manager

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

Perform the following steps:

1. Plan Your Installation
2. Open Necessary Ports
3. Install and Initialize PostgreSQL Server
4. Install Dependencies, Packages, or Clients
5. (Optional) Build a Custom RPM
6. Install MAM Server
7. Configure the MAM GUI
8. Configure MAM Web Services
9. Access the MAM GUI
10. Access MAM Web Services
11. Configure Moab Workload Manager to Use Moab Accounting Manager
12. Initialize Moab Accounting Manager

2.5.1 Plan Your Installation

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

Your MAM installation includes:

- MAM Server
- MAM Database
- MAM Clients (possibly several hosts)
- MAM GUI (optional)
- MAM Web Services (optional)

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

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

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

2.5.2 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.

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

See for general instructions and an example of how to open ports in the firewall.

2.5.3 Install and Initialize PostgreSQL Server

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

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

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

1. Install and initialize the PostreSQL Server.

```
[root]# yum install postgresql-server
[root]# service postgresql initdb
```

2. Configure trusted connections.

   Edit or add a "host" line in the pg_hba.conf file for the interface from which the MAM Server will be connecting to the database and ensure that it specifies a secure password-based authentication method (for example, md5).

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

# Replace 127.0.0.1 with the IP address of the MAM Server Host if the
# MAM PostgreSQL server is on a separate host from the MAM server.
host    all             all             127.0.0.1/32            md5
host    all             all             ::1/128                 md5
```

> **ⓘ** Note that the last column of your entry may contain `ident sameuser`. If so, change the authentication method to `md5` as shown above.

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

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

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

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

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

standard_conforming_strings = on
```

5. Start or restart the database.

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

2.5.4 Install Dependencies, Packages, or Clients

Use the following instructions to install the required Moab Accounting Manager dependencies, packages, or clients.

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

1. On the MAM Server Host, the MAM GUI Host, the MAM Web Services Host, and the MAM Client Hosts, do the following:

```
[root]# yum install gcc redhat-lsb-core perl rrdtool perl-Config-Tiny perl-Crypt-
CBC perl-Crypt-DES perl-Crypt-DES_EDE3 perl-Digest-HMAC perl-Digest-SHA perl-Error
perl-JSON perl-Log-Dispatch-FileRotate perl-Log-Log4perl perl-XML-LibXML
```

> ⓘ If installing on RHEL, some packages may not be found in the standard RHEL distribution repositories.
>
> - One way to overcome this problem is to install the missing dependencies from EPEL or other reputable repositories. For example:
>
> ```
> [root]# rpm -Uvh http://dl.fedoraproject.org/pub/epel/epel-release-latest-
> 6.noarch.rpm
> [root]# yum install yum-utils
> [root]# yum-config-manager --disable epel
> [root]# yum install --enablerepo=epel,rhel-6-server-optional-rpms gcc
> redhat-lsb-core perl rrdtool perl-Config-Tiny perl-Crypt-CBC perl-Crypt-DES
> perl-Crypt-DES_EDE3 perl-Digest-HMAC perl-Digest-SHA perl-Error perl-JSON
> perl-Log-Dispatch-FileRotate perl-Log-Log4perl perl-XML-LibXML
> ```
>
> - *Alternatively*, you can install the available packages in the RHEL repository and then install the missing modules from CPAN.
>
> ```
> [root]# yum install --skip-broken gcc redhat-lsb-core perl rrdtool perl-
> Config-Tiny perl-Crypt-CBC perl-Crypt-DES perl-Crypt-DES_EDE3 perl-Digest-
> HMAC perl-Digest-SHA perl-Error perl-JSON perl-Log-Dispatch-FileRotate perl-
> Log-Log4perl perl-XML-LibXML perl-CPAN
> [root]# cpan YAML Config::Tiny Log::Log4perl Log::Dispatch::FileRotate
> Compress::Zlib
> ```
>
> You may need to run the cpan command more than once for it to complete successfully.

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

```
[root]# yum install postgresql postgresql-libs perl-DBD-Pg perl-Date-Manip perl-
Time-HiRes perl-DBI
```

3. On the MAM GUI Host, do the following:

```
[root]# yum install httpd mod_ssl perl-CGI perl-CGI-Session
```

4. On the MAM Web Services Host, do the following:

```
[root]# yum install httpd mod_perl mod_ssl
```

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

```
[root]# yum install perl-suidperl perl-Term-ReadLine-Gnu perl-TermReadKey
```

> ℹ️ If installing on RHEL, some packages may not be found in the standard RHEL distribution repositories. One way to overcome this problem is to install the missing dependencies from EPEL or other reputable repositories. For example:
>
> ```
> [root]# rpm -Uvh http://dl.fedoraproject.org/pub/epel/epel-release-latest-
> 6.noarch.rpm
> [root]# yum install yum-utils
> [root]# yum-config-manager --disable epel
> [root]# yum install --enablerepo=epel perl-suidperl perl-Term-ReadLine-Gnu
> perl-TermReadKey
> ```

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

2.5.5 (Optional) Build a Custom RPM

Do the following:

1. Install `rpm-build`.

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

2. Download the latest MAM build from the Adaptive Computing Moab HPC Suite Download Center (https://www.adaptivecomputing.com/support/download-center/moab-hpc-suite-download/).

3. Untar the downloaded package.

4. Change directories into the untarred directory.

5. Edit the `rpm/SPECS/mam.spec` file for RPM customization.

6. Run `build/rpm-build`.

7. Locate the custom RPM in `rpm/RPMS/x86_64`.

2.5.6 Install MAM Server

On the MAM Server Host, do the following:

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

```
[root]# useradd -m mam
[root]# su - mam
[mam]$ mkdir src
[mam]$ cd src
```

2. Download the latest MAM build from the Adaptive Computing Moab HPC Suite Download Center (https://www.adaptivecomputing.com/support/download-center/moab-hpc-suite-download/).

3. As the mam user, run each of the following commands in order.

```
[mam]$ tar -zxvf mam-9.1.3.tar.gz
[mam]$ cd mam-9.1.3
```

4. Configure the software. For a list of all the configuration options, see 2.17 Moab Accounting Manager Configuration Options - page 117.

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

5. Compile the software.

```
[mam]$ make
```

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

6. Install the software.

```
[mam]$ exit
[root]# cd ~mam/src/mam-9.1.3
[root]# make install
```

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

7. As the database user, create a database called `mam` and grant database privileges to the `mam` user.

> PostgreSQL should have previously been installed using the instructions in 2.2 Preparing for Manual Installation - page 44.

```
[root]# su - postgres
[postgres]$ psql

create database mam;
create user mam with password 'changeme!';
\q

[postgres]$ exit
```

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

```
[root]# vi /opt/mam/etc/mam-server.conf

database.password = changeme!
```

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

```
[root]# su - mam
[mam]$ cd src/mam-9.1.3
[mam]$ psql mam < hpc.sql
[mam]$ exit
```

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

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

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

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

2.5.7 Configure the MAM GUI

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

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

```
[root]# vi /etc/httpd/conf.d/ssl.conf

<Directory "/var/www/cgi-bin">
## Add these lines
     Options ExecCGI
     AddHandler cgi-script .cgi
     AllowOverride All
     Order allow,deny
     Allow from all
</Directory>

# Aliases for /cgi-bin
Alias /cgi-bin/ /var/www/cgi-bin/
Alias /mam /var/www/cgi-bin/mam/

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

2. For Red Hat-based systems where Security Enhanced Linux (SELinux) is enforced, you may need to customize SELinux to allow the web server to make network connections, use `setuid` for authentication, and write to the log file.

   a. Determine the current mode of SELinux.

   ```
   [root]# getenforce

   Enforcing
   ```

   - If the command returns a mode of `Disabled` or `Permissive`, or if the `getenforce` command is not found, you can skip the rest of this step.

   - If the command returns a mode of `Enforcing`, you can choose between options of customizing SELinux to allow the web GUI to perform its required functions or disabling SELinux on your system.

   b. If you choose to customize SELinux, do the following:

   > ℹ SELinux can vary by version and architecture and that these instructions may not work in all possible environments.

   > ℹ If you used the `--prefix=<prefix>` configuration option when you configured Moab Accounting Manager, you must replace references to `/opt/mam` in the example below with the `<prefix>` you specified. See 2.17 Moab Accounting Manager Configuration Options - page 117 for more information.

   ```
   [root]# cat > mamgui.te <<EOF
   module mamgui 1.0;
   require {
        type httpd_sys_script_t;
        type port_t;
        type user_home_t;
        class capability setuid;
   ```

```
      class tcp_socket name_connect;
      class file read;
      class file open;
}
allow httpd_sys_script_t port_t:tcp_socket name_connect;
allow httpd_sys_script_t self:capability setuid;
allow httpd_sys_script_t user_home_t:file read;
allow httpd_sys_script_t user_home_t:file open;
EOF
[root]# checkmodule -M -m -o mamgui.mod mamgui.te
[root]# semodule_package -m mamgui.mod -o mamgui.pp
[root]# semodule -i mamgui.pp
[root]# setenforce 0
[root]# chcon -v -t httpd_sys_content_t /opt/mam/log
[root]# setenforce 1
```

3. For the highest security, it is recommended that you install a public key certificate that has been signed by a certificate authority. The exact steps to do this are specific to your distribution and the chosen certificate authority. An overview of this process for CentOS 7 is documented at https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/System_Administrators_Guide/ch-Web_Servers.html#s2-apache-mod_ssl.

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

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

4. Start or restart the HTTP server daemon.

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

2.5.8 Configure MAM Web Services

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

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

```
[root]# vi /etc/httpd/conf.d/ssl.conf
# Place the following within the 443 VirtualHost definition
PerlOptions +Parent
PerlSwitches -Mlib=/opt/mam/lib
PerlModule MAM::WSResponseHandler
PerlModule MAM::WSAuthenHandler
<Location /mamws>
     SetHandler perl-script
     PerlResponseHandler MAM::WSResponseHandler
     Options +ExecCGI

     AuthName MAM
```

```
        PerlAuthenHandler MAM::WSAuthenHandler
        Require valid-user

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

2. For Red Hat-based systems where Security Enhanced Linux (SELinux) is enforced, you may need to customize SELinux to allow the web server to make network connections and write to the log file.

   a. Determine the current mode of SELinux.

   ```
   [root]# getenforce

   Enforcing
   ```

   - If the command returns a mode of `Disabled` or `Permissive`, or if the `getenforce` command is not found, you can skip the rest of this step.

   - If the command returns a mode of `Enforcing`, you can choose between options of customizing SELinux to allow MAM Web Services to perform its required functions or disabling SELinux on your system.

   b. If you choose to customize SELinux, do the following:

   > ℹ SELinux can vary by version and architecture and that these instructions may not work in all possible environments.

   > ℹ If you used the `--prefix=<prefix>` configuration option when you configured Moab Accounting Manager, you must replace references to `/opt/mam` in the example below with the `<prefix>` you specified. See 2.17 Moab Accounting Manager Configuration Options - page 117 for more information.

   ```
   [root]# cat > mamws.te <<EOF
   module mamws 1.0;
   require {
       type httpd_t;
       type port_t;
       type usr_t;
       class tcp_socket name_connect;
       class file { create append };
   }
   allow httpd_t port_t:tcp_socket name_connect;
   allow httpd_t usr_t:file { create append };
   EOF
   [root]# checkmodule -M -m -o mamws.mod mamws.te
   [root]# semodule_package -m mamws.mod -o mamws.pp
   [root]# semodule -i mamws.pp
   [root]# setenforce 0
   [root]# chcon -v -t httpd_sys_content_t /opt/mam/log
   [root]# setenforce 1
   ```

3. For the highest security, it is recommended that you install a public key certificate that has been signed by a certificate authority. The exact steps to do this are specific to your distribution and the chosen certificate authority. An overview of this process for CentOS 7 is documented at https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/System_Administrators_Guide/ch-Web_Servers.html#s2-apache-mod_ssl.

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

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

4. Start or restart the HTTP server daemon.

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

2.5.9 Access the MAM GUI

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

1. Create a password for the `mam` user to be used with the MAM Web GUI.

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

2. Verify the connection.

   a. Open a web browser and navigate to `https://<mam-server-host>/cgi-bin/mam`.

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

2.5.10 Access MAM Web Services

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

1. Create a password for the mam user that you wish to access MAM Web Services.

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

2. Make a call to web services.

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

   Alternatively, for queries, you can use the browser to access the URL. For example: 'https://<mam-web-services-host>/mamws/system'.

2.5.11 Configure Moab Workload Manager to Use Moab Accounting Manager

If integrating with Moab Workload Manager, do the following:

1. Configure Moab to talk to MAM

   Do *one* of the following:

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

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

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

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

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

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

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

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

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

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

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

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

On the *Moab* Server Host, follow the instructions in and ; with the following exceptions:

- Install only the dependent packages applicable to MAM Client Hosts.

- Use the configure option `--without-init`.

- Instead of running make, use `make clients-only`.

- Instead of running make install, use `make install-clients-only`.

- Omit the step to create the database and all of the steps thereafter.

2. On the Moab Server Host, restart Moab.

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

2.5.12 Initialize Moab Accounting Manager

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

Related Topics

-

2.6 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 Server Host.

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

2.6.1 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.

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

See 2.13 Opening Ports in a Firewall - page 106 for general instructions and an example of how to open ports in the firewall.

2.6.2 Adjust Security Enhanced Linux

For Red Hat-based systems where Security Enhanced Linux (SELinux) is enforced, you may need to customize SELinux to allow Tomcat to perform operations like making network connections, reading the MWS configuration files, copying MWS plugin jar files, and writing to the MWS log files.

> ℹ️ SELinux can vary by version and architecture and these instructions may not work in all possible environments.

First, determine the current mode of SELinux.

```
[root]# getenforce
Enforcing
```

If the command returns a mode of `Disabled` or `Permissive`, or if the `getenforce` command is not found, you can skip the rest of this step.

If the command returns a mode of `Enforcing`, you can choose between options of customizing SELinux to allow Tomcat to perform its required functions or disabling SELinux on your system.

2.6.2.A Customizing SELinux

If you choose to customize SELinux, do the following:

```
[root]# yum install checkpolicy policycoreutils-python
[root]# cat > mws.te <<EOF
module mws 1.0;
require {
        type tomcat_t;
        type tomcat_var_lib_t;
        type usr_t;
        type tomcat_cache_t;
        type mongod_port_t;
        class process setsched;
        class tcp_socket { name_bind name_connect };
        class dir { add_name create write };
        class file { append create execute setattr write };
}
allow tomcat_t mongod_port_t:tcp_socket name_connect;
allow tomcat_t self:process setsched;
allow tomcat_t tomcat_cache_t:file execute;
allow tomcat_t tomcat_var_lib_t:file execute;
allow tomcat_t usr_t:dir { add_name create write };
allow tomcat_t usr_t:file { append create setattr write };
EOF
[root]# checkmodule -M -m -o mws.mod mws.te
[root]# semodule_package -m mws.mod -o mws.pp
[root]# semodule -i mws.pp
```

2.6.2.B Disabling SELinux

If you choose to disable SELinux:

```
[root]# vi /etc/sysconfig/selinux
SELINUX=disabled

[root]# setenforce 0
```

2.6.3 Install Dependencies, Packages, or Clients

2.6.3.A Install Java

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

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

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>
    ```

2.6.3.B Install Tomcat

Install Tomcat 7.

> ⚠️ Tomcat 7 is required to run MWS 9.0 and after. MWS 9.0 will not run on Tomcat 6.

On the MWS Server Host, do the following:

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

> ℹ️ If installing on RHEL 6, tomcat may not be found in the standard RHEL distribution repositories.
>
> One way to overcome this problem is to install the missing dependencies from EPEL or other reputable repositories. For example (for the current RHEL 6 repositories):
>
> ```
> [root]# rpm -Uvh http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-
> 8.noarch.rpm
> [root]# yum install yum-utils
> [root]# yum-config-manager --disable epel
> [root]# yum install --enablerepo=epel tomcat
> ```

2.6.3.C Install MongoDB

> ℹ️ Setting per-user limits on various resources can prevent MongoDB from closing connections if the number of connections grows to high. See Review and Set Resource Limits (https://docs.mongodb.com/manual/reference/ulimit/#review-and-set-resource-limits) for more information about using the $ulimit$ command to review and set resource limits.

On the MWS MongoDB Database Host, do the following:

1. Add the MongoDB Repository.

```
[root]# cat > /etc/yum.repos.d/mongodb-org-3.2.repo <<'EOF'
[mongodb-org-3.2]
name=MongoDB Repository
baseurl=https://repo.mongodb.org/yum/redhat/$releasever/mongodb-org/3.2/x86_64/
gpgcheck=1
enabled=1
gpgkey=https://www.mongodb.org/static/pgp/server-3.2.asc
EOF
```

2. Install MongoDB.

```
[root]# yum install -y mongodb-org
```

3. Enable and start MongoDB.

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

4. Add the required MongoDB users.

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

```
[root]# mongo
```

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

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

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

> exit
```

> 🛈 Because the `admin_user` has read and write rights to the `admin` database, it also has read and write rights to all other databases. See Control Access to MongoDB Instances with Authentication (at http://docs.mongodb.org/manual/tutorial/control-access-to-mongodb-with-authentication) for more information.

5. Set MongoDB Configuration Options.

   - The configuration file for MongoDB is `/etc/mongod.conf`. See https://docs.mongodb.com/manual/reference/configuration-options for information.

   - Adaptive Computing recommends that you set `security.authorization` to `enabled`. See https://docs.mongodb.com/manual/reference/configuration-options/#security-options for more information.

> 🛈 By default, `/etc/mongod.conf` sets `net.bindIp` to 127.0.0.1. You will need to change this setting if the MongoDB server needs to be accessible from other hosts or from other interfaces besides loopback. See https://docs.mongodb.com/manual/reference/configuration-options/#net-options for more information.

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

6. Restart MongoDB.

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

2.6.4 Install MWS Server

> ℹ️ You `must` complete the tasks to install the dependencies, packages, or clients before installing MWS Server. See 2.6.3 Install Dependencies, Packages, or Clients - page 77.
>
> If your configuration uses firewalls, you *must also* open the necessary ports before installing the MWS Server. See 2.6.1 Open Necessary Ports - page 75.

On the MWS Server Host, do the following:

1. Verify Moab Server is installed and configured as desired (for details, see 2.4 Installing Moab Workload Manager - page 54).

2. Start Moab.

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

3. Create the MWS home directory and subdirectories.

    For more information, see Configuration in the *Moab Web Services Reference Guide*.

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

    Do the following:

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

4. Download the latest MWS build from the Adaptive Computing website.

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-9.1.3.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 mws-9.1.3/utils
[root]# cp * /opt/mws/utils
[root]# chown tomcat:tomcat /opt/mws/utils/*
```

7. Connect Moab to MongoDB.

> ⓘ The USEDATABASE parameter is unrelated to the MongoDB configuration.

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

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

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

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

b. In the /opt/moab/etc/moab-private.cfg file, set the `MONGOUSER` and `MONGOPASSWORD` parameters to the MongoDB moab_user credentials you set. See 2.6.3.C Install MongoDB - page 78.

```
MONGOUSER     moab_user
MONGOPASSWORD secret2
```

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

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

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

8. Secure communication using secret keys.

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

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

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

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

```
[root]# 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
```

   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-9.1.3
[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
```

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


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


...

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

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

> ℹ You can configure only one authentication method in `/opt/mws/etc/mws-config.groovy`—LDAP or PAM, but not both. If you have configured both LDAP and PAM, MWS defaults to using LDAP.
>
> If you need multiple authentication methods, you must add them to your local PAM configuration. See your distribution documentation for details.

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

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

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

> ℹ 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 /opt/mws/etc/mws-config.groovy.

> ℹ️ 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 `/opt/mws/etc/mws-config.groovy` file. For example:

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

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

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

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

d. Add the `grails.mongo.username` and `grails.mongo.password` parameters to
the `/opt/mws/etc/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"
```

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

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

```
[root]# chkconfig tomcat on
[root]# service tomcat stop
[root]# cp /tmp/mws-install/mws-9.1.3/mws.war /usr/share/tomcat/webapps
[root]# service tomcat start
```

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.)

ℹ️ If you encounter problems, or if the application does not seem to be running, see the steps in 5.5 Moab Web Services Issues - page 418.

Related Topics

- 2.2 Preparing for Manual Installation - page 44

2.7 Installing RLM Server

The RLM Server can run multiple licenses.

Access to a Reprise License Manager (RLM) server is required when using Moab's Elastic Computing Feature, Viewpoint's Remote Visualization Feature, or Nitro.

As the RLM Server can run multiple licenses, it is recommended that you install *one* RLM Server for your configuration. If your company already uses an RLM Server, you do not need to install a new one for Adaptive Computing products. However, Adaptive Computing *strongly* recommends that your RLM Server is version 12.1.2 and the Adaptive Computing products may use a different port than the default RLM Server port (5053).

> ⚠️ If your system configuration requires more than one RLM Server, additional configuration may be needed. See 2.18 Using Multiple RLM Servers - page 119 for more information.

> ℹ️ Expired RLM license files must be removed and the RLM service restarted whenever a new license is installed.

This topic contains instructions on how to install an RLM Server.

> In this topic:
>

2.7.1 Open Necessary Ports

> ℹ️ These instructions assume you are using the default ports. If your configuration will use other ports, then substitute your port numbers when opening the 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.

Location	Ports	Functions	When Needed
RLM Server Host	5053	RLM Server Port	Always
RLM Server Host	5054	RLM Web Interface Port	Always

Location	Ports	Functions	When Needed
RLM Server Host	57889	Remote Visualization Port	If Remote Visualization is part of your configuration
RLM Server Host	5135	ISV adaptiveco Port (for the Adaptive license-enabled products)	For Moab Workload Manager *and* if Nitro is part of your configuration.

See 2.13 Opening Ports in a Firewall - page 106 for general instructions and an example of how to open ports in the firewall.

2.7.2 Install the RLM Server

On the host where the RLM Server will reside, do the following:

1. Download the latest RLM build from the Adaptive Computing Moab HPC Suite Download Center (https://www.adaptivecomputing.com/support/download-center/moab-hpc-suite-download/).

2. Create a non-root user and group (rlm is used in the example).

```
[root]# groupadd -r rlm
[root]# useradd -r -g rlm -d /opt/rlm -c "A non-root user under which to run
Reprise License Manager" rlm
```

3. Create a directory and install the tarball files in that location (we are using /opt/rlm as the install location in the example).

```
[root]# mkdir -p -m 0744 /opt/rlm
[root]# cd /opt/rlm
[root]# tar -xzvf /tmp/ac-rlm-12.1.2.tar.gz --strip-components=1
[root]# chown -R rlm:rlm /opt/rlm
```

> ℹ The `--strip-components=1` removes the "ac-rlm-12.1/" from the relative path so that they are extracted into the current directory.

4. Install the startup scripts.

> ℹ If you are using a user:group other than rlm:rlm or a location other than /opt/rlm, then edit the following files to reflect those changes after copying them.

```
[root]# cp init.d/rlm /etc/init.d
```

5. Start the services and configure the RLM Server to start automatically at system reboot.

```
[root]# chkconfig --add rlm
[root]# chkconfig rlm on
[root]# service rlm start
```

2.7.3 Change the Default Passwords

The RLM Web interface includes two usernames (admin and user) by default. These usernames have the default password "changeme!".

> ⚠️ If you do not change this password, RLM, and Remote Visualization, will not be secure. For tips on choosing a good password, see https://www.us-cert.gov/ncas/tips/ST04-002.

Do the following for *both* the user and the admin usernames:

1. Using a web browser, navigate to your RLM instance. (http://<RLM_host>:5054; where <RLM_host> is the IP address or name of the RLM Server Host).

2. Log in.

3. Select `Change Password` and change the password according to your password security process.

> ℹ️ The password for "user" will be needed as part of the Remote Visualization installation.

2.8 Nitro Integration

This section provides instructions on integrating Nitro as part of your Moab configuration.

2.9 Preparing for Nitro Manual Installation

This topic contains instructions on how to download and unpack the Nitro Tarball Bundle for all the hosts in your configuration.

> ⓘ Whether you are installing tarballs on one host or on several hosts, each host (physical machine) on which a server is installed (Nitro, Nitro Web Services) *must* have the Nitro Tarball Bundle.

2.9.1 Set Up Proxies

If your site uses a proxy to connect to the Internet, do the following:

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

2.9.2 Download and Unpack the Nitro Tarball Bundle

The Nitro Tarball Bundle contains all the tarballs available for Nitro. However, not every tarball may be installed on the same host.

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

1. Using a web browser, navigate to the Adaptive Computing Nitro Download Center (http://www.adaptivecomputing.com/support/download-center/nitro/).

2. Download the Nitro Tarball Bundle `nitro-tarball-bundle-<version>-<OS>.tar.gz`.

   > ⓘ 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. Unpack the Nitro Tarball Bundle.

```
[root]# tar xzvf nitro-tarball-bundle-<version>-<OS>.tar.gz
```

Related Topics

2.10 Installing Nitro

This topic contains instructions on how to install Nitro.

Nitro:

- needs to be available to all of the nodes that will be used as part of the Nitro job.
- can be installed either to each node individually *or* to a shared file system that each node can access.

  ⚠ Some Nitro functionality, such as using the nitrosub command, is not available unless you are using a shared file system.

- can be installed to integrate with a scheduler, such as Moab, or without (Nitro standalone). The instructions are the same.

A shared file system allows Nitro task files to be copied to the compute nodes, which is where they will be needed when the job runs. Without a shared file system the Nitro task file will have to be copied to the location where Nitro expects it to be on the compute node manually. This is challenging if it is not known in advance on which compute node the job will run. In addition, a shared file system allows Nitro logs to be seen and opened on the head node, which allows Viewpoint to access them.

In this topic:

2.10.1 Obtain a Nitro License

The Nitro license file is installed on an RLM Server. See for more information about RLM licensing.

⚠ These instructions assume you already have access to an RLM Server. See for instructions on how to set up a new RLM Server.

Do the following:

1. On the RLM server, obtain the host ID and hostname.

   - host ID

   ```
   [root]# /opt/rlm/rlmhostid
   ```

   You should see output similar to the following.

   ```
   rlmhostid v12.1
   Copyright (C) 2006-2016, Reprise Software, Inc. All rights reserved.

   Hostid of this machine: 00259096f004
   ```

   - hostname

   ```
   [root]# /opt/rlm/rlmhostid host
   ```

   You should see output similar to the following.

   ```
   rlmhostid v12.1
   Copyright (C) 2006-2016, Reprise Software, Inc. All rights reserved.

   Hostid of this machine: host=<your-host-name>
   ```

2. Email `licenses@adaptivecomputing.com` for a license and include the host ID and hostname you just obtained.

3. Adaptive Computing will generate the license and send you the Nitro license file (typically, `nitro.lic`) file in a return email.

4. On the RLM server, do the following:

   a. Download and install the license file.

   ```
   [root]# cd /opt/rlm
   [root]# chown rlm:rlm nitro.lic
   ```

   b. If the RLM Server in your configuration uses a firewall, edit the license file to reference the ISV `adaptiveco` port for the Adaptive license-enabled products. This is the same port number you opened during the RLM Server installation. See the instructions to open necessary ports in the 2.7 Installing RLM Server - page 86 (manual installation method) or 3.14 Installing RLM Server - page 255 (RPM installation method) for more information.

   ```
   [root]# vi /opt/rlm/nitro.lic

   ISV adaptiveco port=5135
   ```

   The license file already references the RLM Server port (5053 by default).

   > ℹ️ If the RLM Server in your configuration uses different ports, you will need to modify the license file to reflect the actual ports. See the instructions to open necessary ports in the 2.7 Installing RLM Server - page 86 (manual installation method) or 3.14 Installing RLM Server - page 255 (RPM installation method) for more information.

c. If you did *not* install an RLM Server using the file available from Adaptive Computing (for example, because your system configuration already uses one), do the following:

   i. Download the `adaptiveco.set` file from the Adaptive Computing Nitro Download Center (https://www.adaptivecomputing.com/support/download-center/nitro/).

   ii. Copy the `adaptiveco.set` file into the same directory where the Nitro license resides (`/opt/rlm`).

d. Perform a reread to update the RLM Server with your license.

```
[root]# /opt/rlm/rlmreread
```

2.10.2 Open Necessary Ports

Nitro uses several ports for communication between the workers and the coordinator.

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

> ℹ The listed ports are for configurations that have only one coordinator. If multiple coordinators are run on a single compute host, then sets of ports (range of 4) must be opened for the number of expected simultaneous coordinators.

Location	Ports	Functions	When Needed
Compute Hosts (Nitro Coordinator)	47000	Coordinator/Worker communication	Always
Compute Hosts (Nitro Coordinator)	47001	Coordinator PUB/SUB channel - publishes status information	Always
Compute Hosts (Nitro Coordinator)	47002	Reserved for future functionality	
Compute Hosts (Nitro Coordinator)	47003	API communication channel	Always

See for general instructions and an example of how to open ports in the firewall.

2.10.3 Install Nitro

> ℹ You `must` complete the tasks to obtain a Nitro license before installing Nitro. See 2.10.1 Obtain a Nitro License - page 92.
>
> If your configuration uses firewalls, you *must also* open the necessary ports before installing Nitro. See 2.10.2 Open Necessary Ports - page 94.

On the host where Nitro will reside, do the following:

1. If you have not already done so, complete the steps to prepare the host. See 2.9 Preparing for Nitro Manual Installation - page 90.

2. Change the directory to the root of the unpacked Nitro tarball bundle.

   ```
   [root]# cd nitro-tarball-bundle-<version>-<OS>
   ```

3. Identify the Nitro product tarball (nitro-`<version>`-`<OS>`.tar.gz).

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

   ```
   [root]# mkdir /opt/nitro
   [root]# tar xzvpf nitro-<version>-<OS>.tar.gz -C /opt/nitro --strip-components=1
   ```

5. Copy the license file you generated earlier in this topic to each compute node (coordinator). On each compute node, *or* on the shared file system, do the following:

   ```
   [root]# cp nitro.lic /opt/nitro/bin/
   ```

6. Copy the provided scripts and the *nitrosub* command from the `/opt/nitro/scripts` directory.

   > ℹ This is a "copy" file operation and not a "move" operation. This allows you to customize your version and always have the factory version available for consultation and/or comparison.

   a. Copy the *launch_nitro.sh* and *launch_worker.sh* scripts for your resource manager to the `bin` directory. Each resource manager has a subdirectory with the `scripts` directory that contains the scripts. This example uses Torque as the resource manager.

   ```
   [root]# cp /opt/nitro/scripts/torque/launch_nitro.sh /opt/nitro/bin/
   [root]# cp /opt/nitro/scripts/torque/launch_worker.sh /opt/nitro/bin/
   ```

   b. Copy the *nitrosub* command to the `bin` directory.

   ```
   [root]# cp /opt/nitro/scripts/nitrosub /opt/nitro/bin/
   ```

    c. Copy the `nitro_job.sh` and the `worker_job.sh` scripts to the `etc` directory.

```
[root]# cp /opt/nitro/scripts/nitro_job.sh /opt/nitro/etc/
[root]# cp /opt/nitro/scripts/worker_job.sh /opt/nitro/etc/
```

7. Now that you have copied the scripts and the *nitrosub* command, edit the copies for your site's administrative policies.

   - `bin/nitrosub` command (applicable only if using a shared file system). At a *minimum*, do the following:

     a. Uncomment the `_resource_manager` line for your resource manager.

     b. Uncomment the `resource_type` line for your licensing model's allocation (nodes or cores).

     c. If your system will be using dynamic jobs, set the `_dynamic_size` value to the number of resources to allocate to a dynamic job.

     See 1.1 nitrosub Command in the *Nitro Administrator Guide* for more information.

   - `bin/launch_nitro.sh` and `bin/launch.worker.sh` scripts. See 1.1 Launch Scripts in the *Nitro Administrator Guide* for more information.

8. If your system configuration allows multiple coordinators on the same node, additional configuration may be needed. See for more information.

9. If you are *not* using a shared file system, copy the Nitro installation directory to *all* hosts.

```
[root]# scp -r /opt/nitro root@host002:/opt
```

> ℹ️ If you are not using a shared file system, you may not be able to use the *nitrosub* client command.

2.10.4 Verify Network Communication

Verify that the nodes that will be running Nitro are able to communicate with the Nitro ports *and* that the nodes are able to communicate with one another.

Related Topics

- 2.8 Nitro Integration - page 89

2.11 Installing Nitro Web Services

This topic contains instructions on how to install Nitro Web Services.

Perform the following steps:

1. Open Necessary Ports
2. Install MongoDB
3. Install and Configure Nitro Web Services
4. Configure Viewpoint for Nitro Web Services
5. Grant Users Nitro Permissions in Viewpoint
6. Publish Nitro Events to Nitro Web Services

2.11.1 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.

Location	Ports	Functions	When Needed
Nitro Web Services Host	9443	Tornado Web Port	Always
Nitro Web Services Host	47100	ZMQ Port	Always
Nitro Web Services Database Host	27017	Nitro Web Services MongoDB Server Port	If you will be installing the Nitro Web Services Database on a different host from Nitro Web Services

See 2.13 Opening Ports in a Firewall - page 106 for general instructions and an example of how to open ports in the firewall.

2.11.2 Install MongoDB

> ℹ Setting per-user limits on various resources can prevent MongoDB from closing connections if the number of connections grows to high. See Review and Set Resource Limits (https://docs.mongodb.com/manual/reference/ulimit/#review-and-set-resource-limits) for more information about using the `ulimit` command to review and set resource limits.

On the Nitro Web Services MongoDB Database Host, do the following:

1. Add the MongoDB Repository.

```
[root]# cat > /etc/yum.repos.d/mongodb-org-3.2.repo <<'EOF'
[mongodb-org-3.2]
name=MongoDB Repository
baseurl=https://repo.mongodb.org/yum/redhat/$releasever/mongodb-org/3.2/x86_64/
gpgcheck=1
enabled=1
gpgkey=https://www.mongodb.org/static/pgp/server-3.2.asc
EOF
```

2. Install MongoDB.

```
[root]# yum install -y mongodb-org
```

3. Enable and start MongoDB.

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

4. Add the required MongoDB users.

> ℹ The passwords used below (`secret1` and `secret5`) are examples. Choose your own passwords for these users.

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

> use nitro-db
> db.createUser({"user": "nitro_user", "pwd": "secret5", "roles": ["dbOwner"]})

> exit
```

> ℹ 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.

5. Set MongoDB Configuration Options.

   - The configuration file for MongoDB is `/etc/mongod.conf`. See
     https://docs.mongodb.com/manual/reference/configuration-options for information.

   - Adaptive Computing recommends that you set `security.authorization` to `enabled`.
     See https://docs.mongodb.com/manual/reference/configuration-options/#security-options
     for more information.

> **ⓘ** By default, `/etc/mongod.conf` sets `net.bindIp` to 127.0.0.1. You will need to
> change this setting if the MongoDB server needs to be accessible from other hosts or from
> other interfaces besides loopback. See
> https://docs.mongodb.com/manual/reference/configuration-options/#net-options for
> more information.

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

6. Restart MongoDB.

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

2.11.3 Install and Configure Nitro Web Services

> **ⓘ** You `must` complete the tasks earlier in this topic before installing Nitro Web Services.

On the host where Nitro Web Services will reside, do the following:

1. If you have not already done so, complete the steps to prepare the host. See .

2. Change the directory to the root of the unpacked Nitro tarball bundle.

```
[root]# cd nitro-tarball-bundle-<version>-<OS>
```

3. Identify and unpack the Nitro Web Services tarball (nitro-web-services-
   <version>.<OS>.tar.gz).

```
[root]# tar -xzvpf nitro-web-services-<version>.<OS>.tar.gz
```

4. Install Nitro Web Services.

```
[root]# cd /opt/nitro-web-services-<version>.<OS>
[root]# ./install <directory>
# <directory> is where you want to install Nitro Web Services (defaults to /opt)
```

5. Understand and edit the the Nitro Web Services web application configuration file
   (/opt/nitro-web-services/etc/nitro.cfg). This includes clarifying what the
   configuration file is for and what to expect the first time the NWS service is started vs. each
   subsequent start.

> ℹ The nitro_user with dbOwner permissions was set up earlier in the procedure (see
> 2.11.2 Install MongoDB - page 98).
>
> When you first start nitro-web-services, the nitro-db Mongo database
> (including its collections and indexes) is created. The nitro-db 'user' collection is also
> populated with the default Nitro Web Services API users/passwords. Several of the
> options defined in the configuration files influence this process.

MongoDB user, table, and index creation is performed at initial startup. Many of the options
defined in the Nitro Web Service configuration files influence Mongo user/password and index
creation.

> ⚠ Usernames and passwords are created *only* if they do not yet exist. Changing a password
> in the configuration file after initial startup will not update the password.

The /opt/nitro-web-services/etc/nitro.cfg configuration file includes default
password values. If you want to use passwords other than the default value, change the lines in
the file as shown below.

- Before initial startup, set the db_password to be the nitro_user password. It is also
  recommended that you change all other default passwords before starting Nitro Web
  Services. If you do not change the passwords at this point, it will be more difficult to change
  them later.

```
db_password = <password goes here>
admin_password = <admin_password goes here>
ws_readonly_password = <ws_readonly_password goes here>
ws_writeonly_password = <ws_writeonly_password goes here>
```

- By default, NWS uses an auto-generated self-signed SSL certificate to encrypt the link
  between the web server and the browser clients. The auto-generated self-signed SSL
  certificate is created at service start up; not during the installation process.

However, you can use your own `certfile`, `keyfile`, and `ca_certs` files if you wish.

> 🛈 If you choose to use your own `ssl_certfile` and `ssl_keyfile`, `ssl_create_self_signed_cert=true` is ignored.

- By default, NWS does *not* encrypt network traffic with MongoDB. You should set the `db_ssl_*` properties if you choose to enable TLS/SSL when installing MongoDB earlier in this topic.

6. Understand and edit the Nitro ZMQ Job Status Adapter configuration file (`/opt/nitro-web-services/etc/zmq_job_status_adapter.cfg`). This includes clarifying what the configuration file is for and what to expect the first time the NWS service is started vs. each subsequent start.

   - The Nitro ZMQ Job Status Adapter listens to job status updates on the ZMQ bus and publishes them to MongoDB using the Nitro Web Services REST API.

   - The username and password must be set to a Nitro Web Services API user with write permissions. At minimum, set the password for `nitro-writeonly-user` to the password defined in `/opt/nitro-web-services/etc/nitro.cfg` and make sure the SSL options are set correctly based on SSL settings in `/opt/nitro-web-services/etc/nitro.cfg`.

   ```
   password = <ws_writeonly_password goes here>
   ```

7. If you did not need to install the Nitro Web Services MongoDB database earlier in this topic, verify that the mongodb_hostlist in `/opt/nitro-web-services/etc/nitro.cfg` is set correctly (localhost:27017 is the default).

8. Start the services and configure Nitro Web Services to start automatically at system boot.

   ```
   [root]# chkconfig --add nitro-web-services
   [root]# chkconfig --add nitro-zmq-job-status-adapter
   [root]# service nitro-web-services start
   [root]# service nitro-zmq-job-status-adapter start
   ```

2.11.4 Configure Viewpoint for Nitro Web Services

Do the following:

1. Using a web browser, navigate to your Viewpoint instance (`http://<server>:8081`) and then log in as the MWS administrative user (moab-admin, by default).

2. Click `Configuration` from the menu and then click `Nitro Services` from the left pane. The following is an example of the Nitro Services Configuration page.

3. Enter the configuration information. The following table describes the required information.

Field	Description
Nitro WS URL	Hostname (or IP address) and port number for the host on which you installed Nitro Web Services. For example, https://\<hostname>:9443
Username	Name of the user. This typically nitro-readonly-user.
Password	The user's password.
Trust Self Signed	Indicates whether Nitro Web Services was set up using self-signed certificates.

4. Click TEST to confirm the settings are correct. This confirms whether Nitro Web Services is up and receiving connections.

5. Click SAVE to submit your settings.

6. (Recommended) Use curl to test Nitro Web Services connectivity.

```
[root]# curl --insecure --data '{"username": "nitro-admin", "password":
"ChangeMe2!"}' \
 https://<hostname>:9443/auth
```

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

```
{
  "status": 200,
  "data": {
```

```
    "nitro-key": "3e0fb95e9a0e44ae91daef4deb500dcc67a3714880e851d781512a49",
    "user": {
      "username": "nitro-admin",
      "last_updated": "2016-02-26 23:34:55.604000",
      "name": "Nitro Admin",
      "created": "2016-02-26 23:34:55.604000",
      "auth": {
        "job": [
          "read",
          "write",
          "delete"
        ],
        "user": [
          "read",
          "write",
          "delete"
        ]
      }
    }
  }
}
```

2.11.5 Grant Users Nitro Permissions in Viewpoint

Viewpoint comes packed with base (default) roles for Nitro jobs. Any user who will be working with Nitro Web Services, must have the appropriate role added to the Viewpoint user principal.

These are the Viewpoint roles for Nitro:

- NitroAdmin – Administrative user, with permission to create Nitro application templates and manage other user's Nitro jobs.

- NitroUser – Basic user, with permission to create and manage their own Nitro jobs.

See "Creating or Editing Principals" in the *Moab Viewpoint Reference Guide* for instructions on setting up principals.

2.11.6 Publish Nitro Events to Nitro Web Services

You need to configure the Nitro coordinators to send job status updates to the Nitro Web Services's ZMQ Job Status Adapter. The ZMQ Job Status Adapter is responsible for reading job status updates off of the ZMQ bus and persisting them to Mongo. Nitro Web Services can then be used to access Nitro job status.

Each Nitro job has a Nitro Coordinator. Nitro Coordinators can be configured to publish job status updates to ZMQ by setting the "nws-connector-address" configuration option in Nitro's nitro.cfg file. Each compute node allocated/scheduled to a Nitro Job can play the role of a Nitro coordinator. Therefore, you must update the "nws-connector-address" in each compute node's nitro.cfg file.

> ℹ Configuring nws-connector-address is simplified if each node is sharing Nitro's configuration over a shared filesystem. If you are not using a shared filesystem, update the Nitro configuration on each compute node.

Do the following:

1. If you have not already done so, on the Nitro Web Services host, locate the msg_port number in the `/opt/nitro-web-services/etc/zmq_job_status_adapter.cfg` file. This is the port number you need to specify for the nws-connector-address.

2. On *each* Nitro compute note (Torque MOM Host), specify the nws-connector-address in the `/opt/nitro/etc/nitro.cfg` file.

   ```
   ...
   # Nitro Web Services connection allows Nitro to communicate job status information
   # to the Nitro Web Services connector. This option indicates name and port of the
   # remote server in the form: <host>:<port>
   nws-connector-address <nitro-web-services-hostname>:47100
   ...
   ```

> ℹ You must uncomment the nws-connector-address line, even if it is already correct.

Related Topics

- 2.8 Nitro Integration - page 89

2.12 Additional Configuration

2.13 Opening Ports in a Firewall

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

This topic provides an example and general instructions for how to open ports in your firewall. See 5.3 Port Reference - page 411 for the actual port numbers for the various products.

Red Hat 6-based systems use iptables as the default firewall software. For the ip6tables service, replace all occurrences of iptables with ip6tables in the example. If you use different firewall software, refer to your firewall documentation for opening ports in your firewall.

The following is an example of adding port 1234 when using 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 1234 -j ACCEPT

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

2.14   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).

2.15 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.

2.15.1 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.

ℹ️ 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.

> ℹ️ 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
```

> ℹ 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 2.15.2 Adding an Organizational Unit (OU) - page 112.

2.15.2 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
```

2.15.3 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
```

2.15.4 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
```

2.15.5 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.

> ℹ Before you add a user to an LDAP group, the user must first be added to LDAP. For more information, see 2.15.3 Adding a User - page 113.

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
```

2.16 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.

Optio-n	Description	Example
--prefix	Specifies the location of the binaries and lib-raries 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.	```[root]# ./configure --with-am```
--with-am-dir	Uses the specified prefix directory for the accounting manager if installed in a non-default location.	```[root]# ./configure --with-am-dir=/opt/mam-9.1.3```
--with-flexlm	Causes Moab to install the `license.-mon.flexLM.pl` script in the `/op-t/moab/tools` directory. For more information about this script, see the Inter-facing to FlexNet (formerly FLEXlm) section in the *Moab Workload Manager Administrator Guide*.	```[root]# ./configure --with-flexlm```
--with-homedi-r	Specifies the location of the Moab con-figuration directory and the MOABHOMEDIR environment variable. The default location is `/opt/moab`. ⓘ By default, MOABHOMEDIR is auto-matically set during installation. Use the --without-profile option to disable installed scripts.	```[root]# ./configure --with-homedir=/var/moab``` *The Moab home directory will be `/var/-moab` instead of the default `/opt/moab`.*

Optio-n	Description	Example
--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-pro-file	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`

2.17 Moab Accounting Manager Configuration Options

The following table comprises commonly-used configure options.

Option	Description
-h,--help	Run `./configure --help` to see the list of configure options.
--localstatedir=DIR	Home directory where per-configuration subdirectories (such as `etc`, `log`, `data`) will be installed (defaults to `PREFIX`).
--prefix=PREFIX	Base installation directory where all subdirectories will be installed unless otherwise designated (defaults to `/opt/mam`).
--with-cgi-bin=DIR	If you intend to use the web GUI, use `--with-cgi-bin` to specify the directory where you want the Moab Accounting Manager CGI files to reside (defaults to `/var/www/cgi-bin/mam`).
--with-db-name=NAME	Name of the SQL database that the server will sync with (defaults to `mam`).
--with-legacy-links	Creates symbolic links allowing the use of the old client and server command names (for example, *mam-list-users* would be created as symbolic link to *mam-list-users*). When running a command under its old name, the command will issue a deprecation warning. This warning can be disabled by setting `client.deprecationwarning = false` in the `mam-client.conf` file. The default is not to install the legacy links.
--with-mam-libs=local|site	Use `--with-mam-libs` to indicate whether you want to install the Perl MAM modules in a local directory (`${exec_prefix}/lib`) or in the default system site-perl directory (defaults to `local`).
--with-pro-motion=mamauth|suidperl	Command-line clients and scripts using the API need to use a security promotion method to authenticate and encrypt the communication using the symmetric key. The default is `suidperl` if it is installed on the system, otherwise the default is `mamauth`. See the description for the `security.promotion` configuration parameter in the Client Configuration section for more information about the two security promotion methods.
--with-user=USER	Use `--with-user` to specify the accounting admin userid that the server will run under and who will have full administrative privileges (defaults to `mam`). It is recommended that this be a non-privileged user for the highest security.

Option	Description
--without-gui	Specifies whether to install the CGI web GUI. If you do not intend to use the CGI web GUI, you can specify `--without-gui` to not install the CGI scripts. Otherwise, the default is to install the GUI CGI scripts.
--without-init	If you do not intend to use the mam `init.d` service, you can use `--without-init` to specify that Moab should not install the mam `init.d` script. Otherwise, the script is installed by default.
--without-profile	If you do not intend to use the mam `profile.d` environment scripts, you can use `--without-profile` to specify that Moab should not install the mam `profile.d` scripts. Otherwise, the scripts are installed by default.

2.18 Using Multiple RLM Servers

As the RLM Server can run multiple licenses, it is recommended that you install *one* RLM Server for your configuration.

However, if your configuration requires more than one RLM Server, you will *need* to configure the Adaptive Computing products to connect to a specific RLM Server. If not configured to connect to a specific RLM Server, the Adaptive Computing product will scan the network and connect to the first RLM Server it finds listening to request the license. If the first RLM Server does *not* have the product's license, the RLM connection will fail.

If you are using multiple RLM Servers, do the following to configure the an Adaptive Computing product to connect to a specific RLM Server:

1. Modify the RLM Server not to accept the network search connections.

    - Edit the init script in `/opt/rlm/` to add -noudp.

      ```
      start() {
      su -l $rlmuser -s /bin/bash -c "$rlmdir/rlm -l -dlog $debuglog -noudp &"
      }
      ```

2. Enable the Adaptive Computing product to connect to a specific RLM.

    On the host where the Adapative Computing product resides, do the following:

    a. Create a new text file and name it with the .lic extension (typically, remote.lic) and save it in the same location as the other Adaptive Computing licenses. Be careful not to override an existing license.

    b. Edit the new remote.lic file to point to the specific RLM Server hostname and port. Port 5053 is the default. If you use a different port number for the RLM Server, specify that port number in the remote.lic file.

      ```
      HOST <hostname> ANY 5053
      ```

Repeat as needed for each Adaptive Computing product that you want to connect to a specific RLM Server.

2.19 Running Multiple Coordinators on the Same Node

Nitro provides the ability to run multiple coordinators on the same node.

> ℹ️ Running multiple coordinators on the same node is not available if your system configuration uses a policy to limit nodes to a single job (i.e., NODEACCESSPOLICY=SINGLEJOB on Moab).

If your system is configured to allow multiple coordinators on the node:

- It is recommended that you instruct your uses to submit Nitro jobs using the nitrosub command. See nitrosub Command in the *Nitro User Guide* for more information.

- If you prefer that your users do *not* use the nitrosub command, and instead you prefer that they submit the Nitro jobs directly to your scheduler/resource manager, then you will need to add the --port-file option to the bin/launch_nitro.sh *and* bin/launch_worker.sh scripts to ensure that all coordinators will be able to run.

```
NITRO_OPTIONS="--port-file --job-id ${NITROJOBID} ${NITRO_OPTIONS}"
```

Add the --port-file option `before` *the --job-id information.*

2.20 Trusting Servers in Java

In this topic:

2.20.1 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/tomcat/tomcat.conf
```

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

2.20.2 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.

2.20.3 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.

> ℹ 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.

2.21 Manual Upgrade

This section provides instructions and other information when upgrading your for installing your Moab components for Red Hat 6-based systems using the Manual upgrade method.

> ⚠️ 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.

> ℹ️ 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.

2.22 Upgrading to MongoDB 3.2.x

Moab HPC Suite 9.1.0 and after requires MongoDB 3.2.x. On each host on which MongoDB is installed, check the installed version of MongoDB to see if it needs to be upgraded. You can check the version of MongoDB installed by running the following command:

```
[root]# rpm -qa | grep mongo
```

> ⚠ In order to upgrade the MongoDB databases, you must stop all services *first*. These instructions assume that you have all the MongoDB databases on the same host (for example, the Database Host). If you have installed the MongoDB databases on *separate* hosts (for example, the Insight MongoDB on the Insight Server Host), you will have to go to *each* host to stop the services before you can upgrade any of the MongoDB databases.

If MongoDB is at a version prior to 3.2, do the following:

1. Stop *all* the services that use MongoDB. See the warning at the beginning of this topic.

```
[root]# service nitro-web-services stop # If Nitro Web Services is part of your
configuration
[root]# service tomcat stop # If MWS is part of your configuration
[root]# service insight stop # If Insight is part of your configuration
[root]# service moab stop
```

2. Confirm that nothing is connected to MongoDB.

```
[root]# netstat -antp | egrep '(27017|28017).*ESTABLISHED'
```

3. Dump the database.

```
[root]# cd /root
[root]# mongodump -u admin_user -p secret1
[root]# cp -a dump dump.save
[root]# rm -rf dump/admin/system.users.* # Cannot restore users.
```

4. Install MongoDB 3.2.x.

```
[root]# service mongod stop
[root]# chkconfig mongod off
[root]# cat > /etc/yum.repos.d/mongodb-org-3.2.repo <<'EOF'
[mongodb-org-3.2]
name=MongoDB Repository
baseurl=https://repo.mongodb.org/yum/redhat/$releasever/mongodb-org/3.2/x86_64/
gpgcheck=1
enabled=1
gpgkey=https://www.mongodb.org/static/pgp/server-3.2.asc
EOF
[root]# rpm -e --nodeps $(rpm -qa 'mongo*')
[root]# rm -rf /tmp/mongo*.sock /var/run/mongo* /var/lib/mongo* /var/log/mongo*
[root]# yum install mongodb-org
[root]# chkconfig mongod on
[root]# service mongod start
```

5. Restore the database.

```
[root]# cd /root
[root]# mongorestore
```

6. Create the users.

> ℹ The `admin_user` is required. All other users are required only for the products that are part of your system configuration. For example, if Nitro Web Services is not part of your confirmation, you do not need to add the `nitro_user`.

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

        use moab
        db.createUser({"user": "moab_user", "pwd": "secret2", "roles":
["dbOwner"]})
        db.createUser({"user": "mws_user", "pwd": "secret3", "roles": ["read"]})
        db.createUser({"user": "insight_user", "pwd": "secret4", "roles":
["read"]})

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

        use insight
        db.createUser({"user": "insight_user", "pwd": "secret4", "roles":
["dbOwner"]})
        db.createUser({"user": "mws_user", "pwd": "secret3", "roles": ["read"]})

        use nitro-db
        db.createUser({"user": "nitro_user", "pwd": "secret5", "roles":
["dbOwner"]})

        exit
```

7. Set MongoDB Configuration Options.

   - The configuration file for MongoDB is `/etc/mongod.conf`. See https://docs.mongodb.com/manual/reference/configuration-options for information.
   - Adaptive Computing recommends that you set `security.authorization` to `enabled`. See https://docs.mongodb.com/manual/reference/configuration-options/#security-options for more information.

> ℹ By default, `/etc/mongod.conf` sets `net.bindIp` to 127.0.0.1. You will need to change this setting if the MongoDB server needs to be accessible from other hosts or from other interfaces besides loopback. See https://docs.mongodb.com/manual/reference/configuration-options/#net-options for more information.

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

8. Restart MongoDB.

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

9. Follow the instructions to upgrade your Moab HPC Suite components.

2.23 Upgrading Torque Resource Manager

Torque 6.1 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 6.1.3 from versions prior to 5.0, all MOM and server daemons must be upgraded at the same time.

The job format is compatible between 6.1 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.

> ⓘ See 1.1 Considerations Before Upgrading in the *Torque Resource Manager Administrator Guide* for additional important information, including about how to handle running jobs during an upgrade, mixed server/MOM versions, and the possibility of upgrading the MOMs without having to take compute nodes offline.

In this topic:

2.23.1 Before You Upgrade

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

2.23.1.A Running Jobs

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

2.23.1.B Cray Systems

For upgrading Torque to 6.1.3 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*.

2.23.1.C hwloc

> ⚠️ Using "yum install hwloc" may install an older, non-supported version.

When cgroups are enabled (recommended), hwloc version 1.9.1 or later is required. NVIDIA K80 requires libhwloc 1.11.0. If cgroups are to be enabled, check the Torque Server Host to see if the required version of hwloc is installed. You can check the version number by running the following command:

```
[root]# hwloc-info --version
```

The following instructions are for installing version 1.9.1.

If hwloc is not installed or needs to be upgraded to the required version, do the following:

1. On the Torque Server Host, *each* Torque MOM Host, and *each* Torque Client Host, do the following:

   a. Download hwloc-1.9.1.tar.gz from https://www.open-mpi.org/software/hwloc/v1.9.

   b. Run each of the following commands in order.

   ```
   [root]# yum install gcc make
   [root]# tar -xzvf hwloc-1.9.1.tar.gz
   [root]# cd hwloc-1.9.1
   [root]# ./configure
   [root]# make
   [root]# make install
   ```

2. Run the following commands on the Torque Server Host *only*.

   ```
   [root]# echo /usr/local/lib >/etc/ld.so.conf.d/hwloc.conf
   [root]# ldconfig
   ```

2.23.1.D GPU Support

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

2.23.2 Stop Torque Services

Do the following:

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

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

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

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

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

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

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

2.23.3 Upgrade the Torque Server

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

On the Torque Server Host, do the following:

1. Back up your `server_priv` directory.

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

2. If not already installed, install the Boost C++ headers.

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

3. Download the latest Torque build from the Adaptive Computing website.

4. Depending on your system configuration, you will need to add `./configure` command options.

   At a minimum, you add:

   - `--enable-cgroups`
   - `--with-hwloc-path=/usr/local` See 1.3.1 Torque - page 19 for more information.

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

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

5. Install the latest Torque tarball.

```
[root]# cd /tmp
[root]# tar xzvf torque-6.1.3.tar.gz
[root]# cd torque-6.1.3
[root]# ./configure --enable-cgroups --with-hwloc-path=/usr/local # add any other
specified options
[root]# make
[root]# make install
```

2.23.4 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-x86_64.sh ...
Building ./torque-package-mom-linux-x86_64.sh ...
Building ./torque-package-server-linux-x86_64.sh ...
Building ./torque-package-gui-linux-x86_64.sh ...
Building ./torque-package-devel-linux-x86_64.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 mom package 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-x86_64.sh <torque-mom-host>:
```

> ⓘ If you have already installed init scripts, you may want to skip this step after comparing your existing scripts to the ones provided in the `contrib` directory. If you decide to use the new ones, be sure to check for manual alterations (e.g., to ulimits) in the old ones. The script includes a check for the presence of `/etc/sysconfig/pbs_mom`, so the natural place to put ulimits would be there, if not directly in the init script itself.

2. On *each* Torque MOM Host, confirm that cgroups have been mounted; if not, mount them.

a. Run *lssubsys -am*.

b. If the command is not found, or you do not see something similar to the following, then cgroups are *not* mounted, continue with these instructions.

```
ns
perf_event
net_prio
cpuset /cgroup/cpuset
cpu /cgroup/cpu
cpuacct /cgroup/cpuacct
memory /cgroup/memory
devices /cgroup/devices
freezer /cgroup/freezer
net_cls /cgroup/net_cls
blkio /cgroup/blkio
```

c. Install the cgroup library package and mount cgroups.

```
[root]# yum install libcgroup
[root]# service cgconfig start
[root]# chkconfig cgconfig on
```

d. Run *lssubsys -am* again and confirm cgroups are mounted.

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

a. Install the self-extracting MOM package.

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

2.23.5 Update the Torque Clients

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

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

    a. Copy the self-extracting client package to *each* Torque Client Host.

        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-x86_64.sh <torque-client-host>:
        ```

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

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

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

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

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

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

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

2.23.6 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.

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

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

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

3. On the Torque Server Host, start up the Torque server.

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

2.23.7 Perform Status and Error Checks

On the Torque Server Host, do the following:

- Verify that the status of the nodes and jobs are as expected.

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

2.24 Upgrading Moab Workload Manager

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

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

> In this topic:
>
> 2.24.1 Recommendations - page 134
> 2.24.2 Upgrade Moab Workload Manager - page 134

2.24.1 Recommendations

It is highly recommended that you *first* perform upgrades in a *test environment*. See the warning in 1.1 Preparing for Upgrade. 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* for more information.

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.

2.24.2 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 2.2.2 Enable Extra Packages for the Repository - page 45 for more information.

2. Download the latest Moab build from the Adaptive Computing website.

3. Untar the distribution file. For example:

```
[root]# tar -xzvf moab-9.1.3-<OS>.tar.gz
```

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

4. Change directory into the extracted directory.

```
[root]# cd moab-9.1.3-<OS>
```

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]# service moab stop
```

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

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.

> ℹ 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
```

> ℹ 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 confirm the database schema compatibility. See Migrating Your Database to Newer Versions of Moab in the *Moab Workload Manager Administrator Guide* for more information.

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

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

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

```
Defaults:   server=:42559  cfgdir=/opt/moab (env)  vardir=/opt/moab
Build dir:  /tmp/jenkins/workspace/MWM-9.1.3/label/build-<OS>
Build host: us-devops-build10
Build date: Oct 09 13:00:00 MST 2018
Build args: NA
Compiler Flags:  -D__M64 -D_BUILDDATETIME="2018100913" -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 9.1.3 (revision 2018100913, changeset
14dee972ebcee919207e48054e9f285db9f6a555)
```

12. If you are upgrading from a version of Moab Workload Manager prior to 8.1 and you are using Moab Accounting Manager with the native interface (TYPE=native), remove all entries in moab.cfg with the form (AMCFG[*] *URL=exec://*), except for those that you have customized. See AMCFG Parameters and Flags in the *Moab Workload Manager Administrator Guide* for more information.

> ⓘ In Moab Workload Manager 8.1 and after, Moab defaults to using a set of stock scripts that no longer need to be explicitly configured in the server configuration file.

13. Start Moab.

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

2.25 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 from a version 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 from a version at or after 9.0, run `mam-shell System Query`.

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

   b. If the current version is lower than 9.1, 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.

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

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

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

> **ⓘ** MySQL is no longer a supported database for MAM. If you are using MySQL for your MAM database, follow the instructions in 2.32 Migrating the MAM Database from MySQL to PostgreSQL - page 159 to convert your database.

5. If your PostgreSQL database version is prior to version 9.1, update the postgresql configuration to avoid interpreting backslashes as escape characters.

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

standard_conforming_strings = on

[root]# service postgresql restart
```

6. Verify that each of the prerequisites listed in 2.5 Installing Moab Accounting Manager - page 62 have been satisfied.

7. Download the latest MAM build from the Adaptive Computing website.

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

```
[mam]$ tar -zxvf mam-9.1.3.tar.gz
[mam]$ cd mam-9.1.3
```

9. Configure Moab Accounting Manager by running `configure` 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.

> **⚠** Client and server command names 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.

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

10. Run `make` to compile the program.

```
[mam]$ make
```

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

11. Run *make install* as root to install Moab Accounting Manager.

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

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

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 from a version 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
    ```

    - If you are upgrading MAM from a version at or after 9.1, and you are using MAM Web Services, merge and change any of the new options supplied in the new default MAM Web

Services configuration file (saved in mam-ws.conf.dist) into your existing MAM Web Services configuration file (mam-ws.conf).

```
[mam]$ diff /opt/mam/etc/mam-ws.conf /opt/mam/etc/mam-ws.conf.dist
[mam]$ vi /opt/mam/etc/mam-ws.conf
```

13. Start the server daemon.

```
[mam]$ su -c "service mam start"
```

14. If you are migrating your database to 9.1, 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 (9.1). These scripts are designed to be rerunnable, so if you encounter a failure, resolve the failure and rerun the migration script. If you are unable to resolve the failure and complete the migration, contact Support.

For example, if you are migrating from Moab Accounting Manager version 7.2, you must run six 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, the fifth to migrate the database schema from 8.1 to 9.0, and the sixth to migrate the database schema from 9.0 to 9.1.

```
[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
[mam]$ sbin/migrate_9.0-9.1.pl
```

15. Verify that the resulting database schema version is 9.1.

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

Name                   Version Description
---------------------- ------- -----------------
Moab Accounting Manager 9.1    Commercial Release
```

16. Verify that the executables have been upgraded to 9.1.3.

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

Moab Accounting Manager version 9.1.3
```

17. If you are upgrading MAM from a version prior to 9.1.0, and you wish to use MAM Web Services, perform the following procedures (provided in the Installing Moab Accounting Manager topic):

- 2.5.8 Configure MAM Web Services - page 70
- 2.5.10 Access MAM Web Services - page 72

2.26 Upgrading Moab Web Services

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

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

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

2.26.1 Before You Upgrade

This section provides instructions for tasks that need to be performed before you upgrade Moab Web Services.

2.26.1.A Upgrade to Tomcat 7

Tomcat 7, 8 is required to run MWS 9.0 and later.

On the MWS Server Host, do the following to upgrade Tomcat to version 7:

1. Check your Tomcat version.

```
[root]# rpm -qa tomcat
tomcat-7.0.33-4.el6.noarch
```

2. If your Tomcat version is prior to 7, upgrade Tomcat.

```
[root]# service tomcat6 stop
[root]# chkconfig tomcat6 off
[root]# yum install tomcat
```

2.26.1.B Upgrade to Java 8

> ℹ️ 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 2.6.3.A Install Java - page 77 instructions.

2.26.1.C Upgrade to MongoDB 3.2.x

Beginning with version 9.1.0, Moab Web Services requires MongoDB 3.2.x.

> ℹ️ Setting per-user limits on various resources can prevent MongoDB from closing connections if the number of connections grows to high. See Review and Set Resource Limits (https://-docs.mongodb.com/manual/reference/ulimit/#review-and-set-resource-limits) for more information about using the *ulimit* command to review and set resource limits.

On the MWS MongoDB host, do the following:

1. Check your MongoDB server version.

```
[root]# rpm -qa | grep mongo
```

2. If the MongoDB server version is older than 3.2.x, then you will need to upgrade the databases. See 2.22 Upgrading to MongoDB 3.2.x - page 124 for more information.

2.26.2 Back up the MongoDB Databases

> ℹ️ This procedure is only applicable if you did not have to upgrade the MongoDB databases earlier in this topic. See 2.26.1.C Upgrade to MongoDB 3.2.x - page 142 for more information.

On the MWS MongoDB server host, do the following:

1. Stop all services that are using the MongoDB databases.

2. Back up the MongoDB databases.

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

3. Restart the services.

2.26.3 Upgrade Moab Web Services

> ℹ You `must` complete the tasks in 2.26.1 Before You Upgrade - page 141 before upgrading MWS.

On the MWS Server Host, do the following:

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

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

2. Download the latest MWS build from the Adaptive Computing website.

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-9.1.3.tar.gz
[root]# cd mws-9.1.3
```

4. Deploy the updated `mws.war` to Tomcat.

> ℹ If your prior MWS version had tomcat 6, you should have stopped the tomcat6 service when you upgraded to Tomcat 7 (required). See 2.26.1.A Upgrade to Tomcat 7 - page 141 for more information.

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

5. Back up the MWS home directory and create the required destination directory structure.

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

Where `<version>` if the product version being backed up.

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

```
[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-9.1.3/`mws-config.groovy` file into your existing `/opt/mws/etc/mws-config.groovy`.

   a. Depending on your current MWS version, do the following as needed:

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

    > ℹ Version 9.1 removed the Insight PostreSQL database.

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

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

- 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

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

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

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

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

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

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

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

// 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 f
                                maxBackupIndex: 10, //Retain only the 10 most recent log files,
logs to save space
                                layout: new com.ace.mws.logging.ACPatternLayout(), //Configures
format of each log entry
                                threshold: org.apache.log4j.Level.ERROR //Ignore any logging ent
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.gapiParsers',
                        'grails.app.gapiRequests',
                        'grails.app.gapiSerializers',
                        'grails.app.translators',
                        '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. Remove all plugins from /opt/mws/plugins except for those that you may have created. The
    presence of obsolete plugins can prevent MWS from starting up. Out-of-the-box plugins will be
    recreated when MWS is restarted.

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

10. 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.

11. Verify the following lines are added 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"
    ```

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

12. Start Tomcat.

    > ⚠️ You will need to start the "tomcat" service. Starting the "tomcat6" service will install the
    > wrong version of Tomcat.

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

13. 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.

14. 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`.)

> ℹ️ If you encounter problems, or if MWS does not seem to be running, see the steps in 5.5 Moab Web Services Issues - page 418.

2.27 Upgrading RLM Server

Adaptive Computing *strongly* recommends that your RLM Server is version 12.1.2.

2.27.1 Confirm if an Upgrade is Needed

Run the following command to determine your current version of RLM Server.

```
[root]# /opt/rlm/rlm -v
```

If the version reported is less than 12.1BL2, continue with the section to Upgrade the RLM Server later in this topic.

2.27.2 Upgrade the RLM Server

ⓘ These instructions assume you used /opt/rlm as the install location.

On the RLM Server Host, do the following:

1. Download the latest RLM build (`ac-rlm-12.1.2.tar.gz`) from the Adaptive Computing Moab HPC Suite Download Center (https://www.adaptivecomputing.com/support/download-center/moab-hpc-suite-download/).

2. Stop the RLM service.

```
[root]# service rlm stop
```

3. Archive the existing RLM installation, including the license file(s).

```
[root]# mv /opt/rlm/ /opt/rlm-<archive_version>/
```

4. Install the new tarball files.

```
[root]# mkdir -p -m 0744 /opt/rlm
[root]# cd /opt/rlm
[root]# tar -xzvf /<unpack-directory>/ac-rlm-12.1.2.tar.gz --strip-components=1
[root]# chown -R rlm:rlm /opt/rlm
```

ⓘ The `--strip-components=1` removes the "ac-rlm-12.1.2/" from the relative path so that they are extracted into the current directory.

5. Install the startup scripts.

> ℹ️ If you are using a user:group other than rlm:rlm or a location other than /opt/rlm, then edit the following files to reflect those changes after copying them.

```
[root]# cp init.d/rlm /etc/init.d
```

6. Restore the license file(s).

```
[root]# cp /opt/rlm-<archive_version>/*.lic /opt/rlm/
```

7. Restart the RLM service.

```
[root]# service rlm restart
```

2.28 Upgrading Your Nitro Integration

This section provides instructions on upgrading your Nitro Integration as part of your Moab configuration.

2.29 Preparing for Nitro Manual Upgrade

This topic contains instructions on how to download and unpack the Nitro Tarball Bundle for all the hosts in your configuration.

> ℹ Whether you are installing tarballs on one host or on several hosts, each host (physical machine) on which a server is installed (Nitro, Nitro Web Services) *must* have the Nitro Tarball Bundle.

2.29.1 Set Up Proxies

If your site uses a proxy to connect to the Internet, do the following:

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

2.29.2 Download and Unpack the Nitro Tarball Bundle

The Nitro Tarball Bundle contains all the tarballs available for Nitro. However, not every tarball may be installed on the same host.

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

1. Using a web browser, navigate to the Adaptive Computing Nitro Download Center (http://www.adaptivecomputing.com/support/download-center/nitro/).

2. Download the Nitro Tarball Bundle `nitro-tarball-bundle-<version>-<OS>.tar.gz`.

   > ℹ 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. Unpack the Nitro Tarball Bundle.

```
[root]# tar xzvf nitro-tarball-bundle-<version>-<OS>.tar.gz
```

Related Topics

2.30 Upgrading Nitro

This topic contains instructions on how to upgrade Nitro.

2.30.1 Upgrade Nitro

On the Nitro Host, do the following:

1. If you have not already done so, complete the steps to prepare the host. See 2.9 Preparing for Nitro Manual Installation - page 90.

2. Back up your existing launch script in `/opt/nitro/bin/`.

3. Change the directory to the root of the unpacked Nitro tarball bundle.

   ```
   [root]# cd nitro-tarball-bundle-<version>-<OS>
   ```

4. Identify the Nitro product tarball (nitro-`<version>`-`<OS>`.tar.gz) and unpack the tarball into the same directory you created when you first installed Nitro (for example, `/opt/nitro`.

   ```
   [root]# tar xzvpf nitro-<version>-<OS>.tar.gz -C /opt/nitro --strip-components=1
   ```

5. Copy the provided scripts and the *nitrosub* command from the `/opt/nitro/scripts` directory.

   > **ⓘ** This is a "copy" file operation and not a "move" operation. This allows you to customize your version and always have the factory version available for consultation and/or comparison.

   a. Copy the `launch_nitro.sh` and `launch_worker.sh` scripts for your resource manager to the `bin` directory. Each resource manager has a subdirectory with the scripts directory that contains the scripts. This example uses Torque as the resource manager.

      ```
      [root]# cp /opt/nitro/scripts/torque/launch_nitro.sh /opt/nitro/bin/
      [root]# cp /opt/nitro/scripts/torque/launch_worker.sh /opt/nitro/bin/
      ```

   b. Copy the *nitrosub* command to the `bin` directory.

      ```
      [root]# cp /opt/nitro/scripts/nitrosub /opt/nitro/bin/
      ```

c. Copy the `nitro_job.sh` and the `worker_job.sh` scripts to the `etc` directory.

```
[root]# cp /opt/nitro/scripts/nitro_job.sh /opt/nitro/etc/
[root]# cp /opt/nitro/scripts/worker_job.sh /opt/nitro/etc/
```

6. Merge any customizations from your existing launch scripts, job scripts, and the *nitrosub* command (if applicable) into the new launch scripts, job scripts, and the *nitrosub* command that you copied from the scripts directory.

7. If your system configuration allows multiple coordinators on the same node, additional configuration may be needed. See 2.19 Running Multiple Coordinators on the Same Node - page 120 for more information.

8. If you are not using a shared file system, copy the updated Nitro installation directory to *all* hosts.

```
[root]# scp -r /opt/nitro root@host002:/opt
```

> 🛈 If you are not using a shared file system, you may not be able to use the *nitrosub* client command.

2.30.2 Verify Network Communication

Verify that the nodes that will be running Nitro are able to communicate with the Nitro ports *and* that the nodes are able to communicate with one another.

Related Topics

- 2.28 Upgrading Your Nitro Integration - page 151

2.31 Upgrading Nitro Web Services

This topic contains instructions on how to upgrade Nitro Web Services.

2.31.1 Upgrade to MongoDB 3.2.x

Beginning with version 2.1.0, Nitro Web Services requires MongoDB 3.2.x.

On the Nitro Web Services MongoDB host, do the following:

1. Check your MongoDB server version.

```
[root]# rpm -qa | grep mongo
```

2. If the MongoDB server version is older than 3.2.x, then you will need to upgrade the databases. See 2.22 Upgrading to MongoDB 3.2.x - page 124 for more information.

2.31.2 Back up the MongoDB Databases

> ℹ This procedure is only applicable if you did not have to upgrade the MongoDB databases earlier in this topic. See 2.31.1 Upgrade to MongoDB 3.2.x - page 156 for more information.

On the Nitro Web Services MongoDB serverhost, do the following:

1. Stop all services that are using the MongoDB databases.

2. Back up the MongoDB databases.

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

3. Restart the services.

2.31.3 Upgrade Nitro Web Services

On the Nitro Web Services Host, do the following:

1. If you have not already done so, complete the steps to prepare the host. See 2.9 Preparing for Nitro Manual Installation - page 90 for more information.

2. Stop the services.

```
[root]# service nitro-web-services stop
[root]# service nitro-zmq-job-status-adapter stop
```

3. Back up the contents of the `/opt/nitro-web-services/etc` directory (contains the `nitro.cfg` and the `zmq_job_status_adapter.cfg` files).

4. Remove the `/opt/nitro-web-services` directory.

```
[root]# rm -rf /opt/nitro-web-services
```

5. Change the directory to the root of the unpacked Nitro tarball bundle.

```
[root]# cd nitro-tarball-bundle-<version>-<OS>
```

6. Create the `/opt/nitro-web-services` directory.

```
[root]# mkdir -p /opt/nitro-web-services
```

7. Identify the Nitro Web Services tarball (nitro-web-services-`<version>`-`<OS>`.tar.gz) and unpack the tarball into `/opt/nitro-web-services`.

```
[root]# tar -xzvpf  nitro-web-services-<version>-<OS>.tar.gz -C /opt/nitro-web-
services --strip-components=1
```

8. Install Nitro Web Services. This step assumes the installation directory is `/opt/nitro-web-services`.

```
[root]# cd /opt/nitro-web-services
[root]# ./install.sh
```

9. Merge any customizations from the `nitro.cfg` and the `zmq_job_status_adapter.cfg` files (and any other files) you backed up earlier in this procedure into the new files.

> ⓘ See the steps in 2.11.3 Install and Configure Nitro Web Services - page 99 for more information on the configuration files.

10. Restart the services.

```
[root]# service nitro-web-services restart
[root]# service nitro-zmq-job-status-adapter restart
```

2.31.4 Grant Users Nitro Permissions in Viewpoint

Verify that the users who work with Nitro Web Services have the appropriate role in their Viewpoint user principal.

These are the Viewpoint roles for Nitro:

- NitroAdmin – Administrative user, with permission to create Nitro application templates and manage other user's Nitro jobs.

- NitroUser – Basic user, with permission to create and manage their own Nitro jobs.

See "Creating or Editing Principals" in the *Moab Viewpoint Reference Guide* for instructions on setting up principals.

Related Topics

2.32 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.

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

To Migrate the MAM Database

This procedure was successfully tested on an actual customer MySQL database with millions of transactions 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** - 2.6 Installing Moab Web Services - page 75
   - **RPM Install** - 3.10 Installing Moab Web Services - page 198

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 Method

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

3.1 About RPM Installations and Upgrades

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

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

3.1.1 RPM Installation and Upgrade Methods

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

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

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

Adaptive Computing provides two different types of RPM installation or upgrade methods.

- The typical method is the original RPM method in which you download the Moab HPC Suite RPM bundle to each host in your Moab environment.

- The offline method is available for configurations where the hosts in your Moab environment do *not* have internet access in order to download the Moab HPC Suite RPM dependencies. This method requires an authorized user to download the Moab HPC Suite RPM bundle and other related dependencies and create a moab-offline tarball. That tarball is then copied (using scp, DVD, USB drive, or similar) to each host in your Moab environment. See 1.1 Creating the moab-offline Tarball for instructions on how to create the tarball.

3.1.2 Special Considerations

Be aware of the following:

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

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

- If using the offline method, the internet-enabled host *must* have the *exact* same OS as the hosts within your Moab environment. As the Moab can have several hosts, and each host may not use the same OS, you may need to repeat this procedure for each OS used.

3.1.3 Installation and Upgrade Process

Each host (physical machine) will need to have the Moab HPC Suite RPM bundle and the Adaptive Computing repository enabled. This is referred to as preparing the host. Again this can be done using the typical or the offline method.

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

3.2 RPM Installations

This section provides instructions and other information for installing your Moab components for Red Hat 6-based systems using the RPM installation method.

3.3 Preparing for RPM Installs

Depending on the RPM installation method (typical or offline) you choose, you will need to prepare your system for the RPM installations.

- If you are using the *typical* RPM installation method, continue with the topic: 3.4 Preparing the Host – Typical Method - page 170.

- If you are using the *offline* RPM installation method, continue with the topics: 3.5 Creating the moab-offline Tarball - page 173 *and* 3.6 Preparing the Host – Offline Method - page 176.

Related Topics

- 3.1.1 RPM Installation and Upgrade Methods - page 166

3.4 Preparing the Host – Typical Method

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

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

> ℹ️ 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. If Remote Visualization is part of your configuration, the Adaptive Computing Package Repository must also be enabled on the Torque MOM Hosts (compute nodes); otherwise is not necessary to enable the Adaptive Computing repository on the Torque MOM Hosts or client hosts.

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

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

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

2. Update your system software to the latest version.

```
[root]# yum update
```

3. Ensure hostname resolution for all hosts.

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

4. Download the latest Moab HPC Suite RPM bundle from the Adaptive Computing Moab HPC Suite Download Center (https://www.adaptivecomputing.com/support/download-center/moab-hpc-suite-download/).

5. Untar the RPM bundle.

```
[root]# tar zxf moab-hpc-suite-9.1.3-<OS>.tar.gz
```

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

6. Change directories into the untarred directory.

```
[root]# cd moab-hpc-suite-9.1.3-<OS>
```

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

7. 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`" to view usage information and identify the default directory location. If the [`<repository-directory>`] already exists, RPMs will be added to the existing directory. No files are overwritten in [`<repository-directory>`].

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

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

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

8. Test the repository.

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

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

```
...
moab-accounting-manager.x86_64 : Moab Accounting Manager for Moab HPC Suite
moab-hpc-enterprise-suite.noarch : Moab HPC Suite virtual package
moab-insight.x86_64 : Moab Insight
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                            |
| ...                                                                  |
 ---------------------------------------------------------------------
```

9. Continue with instructions to install the Moab HPC Suite components. See 3.2 RPM Installations - page 168.

3.5 Creating the moab-offline Tarball

⚠️ The Moab Offline Tarball is *only* created if you are using the RPM Installation – Offline Method. See 3.1.1 RPM Installation and Upgrade Methods - page 166 for more information.

This topic contains instructions on how to create a moab-offline tarball on a web-enabled host outside of your Moab HPC Suite environment. This is the tarball that is then copied (using either by scp, DVD, USB or similar) to each host within your Moab HPC Suite environment.

⚠️ The internet-enabled host *must* have the *exact* same OS as the hosts within your Moab HPC Suite environment. As the Moab HPC Suite can have several hosts, and each host may not use the same OS, you may need to repeat this procedure for each OS used.

These instructions assume the user is non-root, but has sudo rights.

On a web-enabled host, do the following:

1. If the host uses a proxy to connect to the Internet, do the following:

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

2. Download the Moab HPC Suite RPM bundle from the Adaptive Computing Moab HPC Suite Download Center (https://www.adaptivecomputing.com/support/download-center/moab-hpc-suite-download/).

3. Untar the RPM bundle.

```
[root]# tar zxf moab-hpc-suite-9.1.3-<OS>.tar.gz
```

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

4. Change directories into the untarred directory.

```
[root]# cd moab-hpc-suite-9.1.3-<OS>
```

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

5. Install the suite repositories.

```
sudo ./install-rpm-repos.sh -y
```

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

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

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

6. Confirm you own /opt.

```
sudo chown <user>:<user> /opt
```

7. Create the moab-offline directory in which to store the RPMs.

```
mkdir /opt/moab-offline
```

8. Download the Moab HPC Suite RPMs into the moab-offline directory.

Do the following:

a. Symlink all the Moab HPC Suite RPMs to your moab-offline directory. This enables the repotrack utility to copy them.

```
ln -s /opt/adaptive-rpm-repository/rpm/*.rpm /opt/moab-offline/
```

b. Use repotrack to download all dependency RPMs.

```
repotrack -a x86_64 -p /opt/moab-offline moab-hpc-suite
```

9. Download the Java RPM into the moab-offline directory.

> ℹ️ The Java version may vary depending on the Moab HPC Suite components in your configuration. See 1.3 Component Requirements - page 18 for more information.

```
cd /opt/moab-offline
wget <java_url>
```

10. Create a repository file for the moab-offline directory.

The `createrepo` package and its dependencies should have been installed when you ran `./install-rpm-repos.sh -y`.

```
echo "[moab-offline]
name=moab-offline
baseurl=file:///opt/moab-offline
failovermethod=priority
enabled=1
gpgcheck=0" > moab-offline.repo
```

11. Create the moab-offline tarball. The "h" option ensures the symlinked targets will be copied, instead of just the links.

```
tar hczvf moab-offline.tgz moab-offline
```

This tarball can now be copied (using scp, DVD, USB drive, or similar) to *each* host within your Moab HPC Suite environment.

3.6 Preparing the Host – Offline Method

The offline method is available for configurations where the hosts in your environment do not have internet access in order to download the Moab HPC Suite RPM dependencies.

This topic describes how to deploy the moab-offline tarball so that you can install various Moab HPC Suite components and their dependencies on all the hosts in your environment.

On *each* host (physical machine), do the following:

1. Update your system software to the latest version.

   ```
   [root]# yum update
   ```

2. Ensure hostname resolution for all hosts.

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

3. If you have not already done so, copy the moab-offline tarball to the host. For example, copy it from a CD, USB drive, or Shared network drive. See 3.5 Creating the moab-offline Tarball - page 173 for instructions on how to create the tarball.

4. Place the moab-offline tarball in the /opt directory and enter that directory.

   ```
   mv moab-offline.tgz /opt

   cd /opt
   ```

5. Untar the moab-offline directory.

   ```
   tar xvzf moab-offline.tgz
   ```

6. Copy the moab-offline.repo into place.

   - Copy to yum.repos.d.

     ```
     cp moab-offline/moab-offline.repo /etc/yum.repos.d/
     ```

   - Update the cache.

     ```
     yum clean all
     ```

7. Continue with instructions to install the Moab components. See 3.2 RPM Installations - page 168.

3.7 Installing Torque Resource Manager

⚠️ If you intend to use Torque Resource Manager 6.1.3 with Moab Workload Manager, you must run Moab version 8.0 or later. However, some Torque functionality may not be available. See Compatibility Requirements in the Moab HPC Suite Release Notes for more information.

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

ℹ️ For Cray systems, Adaptive Computing recommends that you install Moab and Torque Servers (head nodes) on commodity hardware (*not* on Cray compute/service/login nodes).

However, you *must* install the Torque pbs_mom daemon and Torque client commands on Cray login and "mom" service nodes since the pbs_mom *must* run on a Cray service node within the Cray system so it has access to the Cray ALPS subsystem.

See Installation Notes for Moab and Torque for Cray" in the *Moab Workload Manager Administrator Guide* for instructions on installing Moab and Torque on a non-Cray server.

In this topic:

3.7.1 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.

Location	Ports	Functions	When Needed
Torque Server Host	15001	Torque Client and MOM communication to Torque Server	Always
Torque MOM Host (Compute Nodes)	15002	Torque Server communication to Torque MOMs	Always

Location	Ports	Functions	When Needed
Torque MOM Host (Compute Nodes)	15003	Torque MOM communication to other Torque MOMs	Always

If using the MOM hierarchy (documented in 1.1 Setting Up the MOM Hierarchy (Optional) in the Torque Resource ManagerAdministrator Guide) you must also open port 15003 from the server to the nodes.

See also:

- for general instructions and an example of how to open ports in the firewall.
- Configuring Ports in the *Torque Resource ManagerAdministrator Guide* for more information on how to configure the ports that Torque uses for communication.

3.7.2 Install Torque Server

> ⓘ If your configuration uses firewalls, you *must* open the necessary ports before installing the Torque Server. See .

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 for more information.

2. Install the Torque Server RPM.

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

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

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

4. Add the hostnames of your Torque MOMs (which 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:

```
[root]# vi /var/spool/torque/server_priv/nodes

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

5. Start the Torque server.

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

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

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

2. On *each* Torque MOM Host, confirm that cgroups have been mounted; if not, mount them.

   a. Run *lssubsys -am*.

   b. If the command is not found, or you do not see something similar to the following, then cgroups are *not* mounted, continue with these instructions.

```
ns
perf_event
net_prio
cpuset /cgroup/cpuset
cpu /cgroup/cpu
cpuacct /cgroup/cpuacct
memory /cgroup/memory
devices /cgroup/devices
freezer /cgroup/freezer
net_cls /cgroup/net_cls
blkio /cgroup/blkio
```

   c. Install the cgroup library package and mount cgroups.

```
[root]# yum install libcgroup
[root]# service cgconfig start
[root]# chkconfig cgconfig on
```

   d. Run *lssubsys -am* again and confirm cgroups are mounted.

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

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

4. 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
```

5. On each Torque MOM Host, start the pbs_mom daemon.

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

6. If you installed the Torque Client RPM on the MOMs, then on each Torque MOM Host, start the trqauthd daemon.

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

3.7.4 Configure Data Management

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

Related Topics

- Chapter 3: RPM Installation Method - page 161

3.8 Installing Moab Workload Manager

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

> ⓘ For Cray systems, Adaptive Computing recommends that you install Moab and Torque Servers (head nodes) on commodity hardware (*not* on Cray compute/service/login nodes).
>
> *However*, you *must* install the Torque pbs_mom daemon and Torque client commands on Cray login and "mom" service nodes since the pbs_mom *must* run on a Cray service node within the Cray system so it has access to the Cray ALPS subsystem.
>
> See Installation Notes for Moab and Torque for Cray" in the *Moab Workload Manager Administrator Guide* for instructions on installing Moab and Torque on a non-Cray server.

In this topic:

3.8.1 Understand Licenses

With the 9.1.0 release, Moab now uses an RLM Server to manage licenses. For the Moab core and for most Moab features, an RLM Server is not required. The new Moab "core" license will have a new name to reflect the RLM generation. Do *not* rename this license to `moab.lic`. See 1.4 RLM Licensing Model - page 32 for more information about RLM licensing.

Elastic Computing, beginning with 9.1.0, requires an RLM Server as part of your configuration.

> ⓘ The 9.1.0 licensing change does not affect legacy licenses; however, a module-based license may be required to use newer functionality.

3.8.2 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.

Location	Ports	Functions	When Needed
Moab Server Host	42559	Moab Server Port	If you intend to run client commands on a host different from the Moab Server Host *or* if you will be using Moab in a grid

See 2.13 Opening Ports in a Firewall - page 106 for general instructions and an example of how to open ports in the firewall.

3.8.3 Obtain and Install the Elastic Computing License

If using Elastic Computing, Moab requires access to an RLM license server to record usage.

> ⚠️ These instructions assume you already have access to an RLM Server. See 3.14 Installing RLM Server - page 255 for instructions on how to set up a new RLM Server.

Do the following:

1. On the RLM server, obtain the hostid and hostname.

   - hostid

   ```
   [root]# /opt/rlm/rlmhostid
   ```

   You should see output similar to the following.

   ```
   rlmhostid v12.1
   Copyright (C) 2006-2016, Reprise Software, Inc. All rights reserved.

   Hostid of this machine: 00259096f004
   ```

   - hostname

   ```
   [root]# /opt/rlm/rlmhostid host
   ```

   You should see output similar to the following.

   ```
   rlmhostid v12.1
   Copyright (C) 2006-2016, Reprise Software, Inc. All rights reserved.

   Hostid of this machine: host=<your-host-name>
   ```

2. Email licenses@adaptivecomputing.com for a license and include the hostid and hostname you just obtained.

3. Adaptive Computing will generate the license and send you the Elastic Computing license file (.lic) file in a return email.

4. On the RLM server, do the following:

    a. Download and install the license file.

    ```
    [root]# cd /opt/rlm
    [root]# chown rlm:rlm <licenseFileName>.lic
    ```

    b. If the RLM Server in your configuration uses a firewall, edit the license file to reference the ISV adaptiveco port for the Adaptive license-enabled products. This is the same port number you opened during the RLM Server installation. See the instructions to open necessary ports in the 2.7 Installing RLM Server - page 86 (manual installation method) or 3.14 Installing RLM Server - page 255 (RPM installation method) for more information.

    ```
    [root]# vi /opt/rlm/moab_elastic_tracking.lic


    ISV adaptiveco port=5135
    ```

    The license file already references the RLM Server port (5053 by default).

    > ⓘ If the RLM Server in your configuration uses different ports, you will need to modify the license file to reflect the actual ports. See the instructions to open necessary ports in the 2.7 Installing RLM Server - page 86 (manual installation method) or 3.14 Installing RLM Server - page 255 (RPM installation method) for more information.

    c. If you did *not* install an RLM Server using the file available from Adaptive Computing (for example, because your system configuration already uses one), do the following:

        i. Download the 'adaptiveco.set' file from the Adaptive Computing Moab HPC Suite Download page (http://www.adaptivecomputing.com/support/download-center/moab-hpc-suite-download/).

        ii. Install the 'adaptiveco.set' file.

        ```
        [root]# chown rlm:rlm adaptiveco.set
        ```

        iii. Place the 'adaptiveco.set' file in the *same* directory where the Elastic Computing license resides. Typically, this is the RLM Server base directory (/opt/rlm); but may be different depending on your configuration

    d. Perform a reread on the RLM Server base directory to update the RLM Server with your license. For example:

    ```
    [root]# /opt/rlm/rlmreread
    ```

3.8.4 Install Moab Server

On the Moab Server Host do the following:

1. If you configuration uses firewalls, confirm you have opened the necessary ports. See 3.8.2 Open Necessary Ports - page 181.

2. If you have not already done so, complete the steps to prepare the Moab Server Host. See 3.3 Preparing for RPM Installs - page 169 for more information.

3. Install RPM packages.

   a. Install the Moab Server RPMs.

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

   > **ℹ** If installing on RHEL, you may need to enable optional RHEL repositories in order to find some of the dependent packages.
   >
   > ```
   > [root]# yum install --enablerepo=rhel-6-server-optional-rpms moab-
   > workload-manager moab-workload-manager-hpc-configuration
   > ```

   b. If you are using Torque as a resource manager and installed the Torque Server on a different host (Torque Server Host; 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
   ```

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

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

4. Source the following file to add the Moab executable directories to your current shell $PATH environment.

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

5. 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, run the following command:

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

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

- New RLM-Based License (version 9.1.0 or after)

```
$ moab --about | grep License
Moab Workload Manager Version 'master' License Information:
  Current License:   (moab_license) Valid Until - 15-jan-2019
  Current License:   Max Sockets  =  1000000
  Current License:   (moab_grid) Valid Until - 15-jan-2019
  Current License:   (moab_green) Valid Until - 15-jan-2019
  Current License:   (moab_provision) Valid Until - 15-jan-2019
  Current License:   (moab_vms) Valid Until - 15-jan-2019
  Current License:   Max VMs   =  1000000
  Current License:   (moab_elastic) Valid Until - 15-jan-2019
  Current License:   (moab_groupsharing) Valid Until - 15-jan-2019
  Current License:   (moab_advancedrm) Valid Until - 15-jan-2019
  Current License:   (moab_workflow) Valid Until - 15-jan-2019
  Current License:   (moab_accounting) Valid Until - 15-jan-2019
```

- Legacy License Format

```
Moab Workload Manager Version '9.1.3' License Information:
 Current License:  Max Procs   = 10000
 Current License:  Valid Until - Jul 13 19:42:10 2019
```

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

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

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

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

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

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

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

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

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

c. Start the trqauthd daemon.

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

7. If using the Moab Passthrough feature for Viewpoint on Slurm, see Moab Passthrough for additional information and installation instructions.

8. Start Moab.

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

3.8.5 Configure Torque to Trust Moab

If you are using Torque as a resource manager and you installed the Torque Server on a different host (Torque Host); 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 as a submit host.

```
[root]# qmgr

Qmgr: set server managers += root@<moab_server_hostname>
Qmgr: set server submit_hosts += <moab_server_hostname>
Qmgr: exit
```

3.8.6 Verify the Installation

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

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

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

Related Topics

-

3.9 Installing Moab Accounting Manager

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

Perform the following steps:

1. Plan Your Installation
2. Confirm Requirements
   a. Hardware Requirements
   b. Supported Operating Systems
   c. Supported Databases
3. Open Necessary Ports
4. Install Dependencies, Packages, or Clients
   a. Install and Initialize PostgreSQL Server
   b. Install Perl ReadLine (Optional)
5. Install MAM Server
6. Configure the MAM GUI
7. Configure MAM Web Services
8. Access the MAM GUI
9. Access MAM Web Services
10. Configure Moab Workload Manager to use Moab Accounting Manager
11. Initialize Moab Accounting Manager

3.9.1 Plan Your Installation

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

Your MAM installation includes:

- MAM Server
- MAM Database
- MAM Clients (possibly several hosts)
- MAM GUI (optional)
- MAM Web Services (optional)

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

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

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

3.9.2 Confirm Requirements

3.9.2.A Hardware Requirements

- Dual or Quad core Intel/AMD x86-64 processor
- At least 8 GB of RAM
- 1-2 TB disk space

> **ⓘ** MAM is commonly installed on the same host as Moab; however, in some cases you might obtain better performance by installing them on different hosts.

3.9.2.B Supported Operating Systems

MAM has been tested on the following variants of Linux:

- CentOS (6, 7)
- RHEL (6, 7)
- Scientific Linux (6, 7)
- SLES (12)

3.9.2.C Supported Databases

MAM uses an RDBMS as a back end. If this is a new installation, use the following database:

- PostgreSQL 7.2 or higher

3.9.3 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.

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

See 2.13 Opening Ports in a Firewall - page 106 for general instructions and an example of how to open ports in the firewall.

3.9.4 Install Dependencies, Packages, or Clients

3.9.4.A Install and Initialize PostgreSQL Server

Moab Accounting Manager uses a database for transactions and data persistence. The PostgreSQL database may be installed on a different host from the MAM Server; however, it is often convenient to install them on the same host.

On the MAM Database Host, do the following:

1. Install and initialize PostgreSQL.

```
[root]# yum install postgresql-server
[root]# service postgresql initdb
```

2. Configure trusted connections.

Edit or add a "host" line in the pg_hba.conf file for the interface from which the MAM Server will be connecting to the database and ensure that it specifies a secure password-based authentication method (for example, md5).

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

# Replace 127.0.0.1 with the IP address of the MAM Server Host if the
# MAM PostgreSQL server is on a separate host from the MAM server.
host    all             all             127.0.0.1/32            md5
host    all             all             ::1/128                 md5
```

> ℹ️ Note that the last column of your entry may contain `ident sameuser`. If so, change the authentication method to `md5` as shown above.

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

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

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

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

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

standard_conforming_strings = on
```

5. Start or restart the database.

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

3.9.4.B Install Perl ReadLine (Optional)

Moab Accounting Manager can be optionally configured to provide command history editing functionality in the mam-shell command.

The perl-Term-ReadLine-Gnu package is recommended and is typically included in the standard repositories for the OS.

To install the perl-Term-ReadLine-Gnu package:

```
[root]# yum install perl-Term-ReadLine-Gnu
```

> ⓘ If installing on RHEL, this package may not be found in the standard RHEL distribution repositories. You will need to install the missing dependencies from EPEL or other reputable repositories.
>
> ```
> [root]# rpm -Uvh http://dl.fedoraproject.org/pub/epel/epel-release-latest-
> 6.noarch.rpm
> [root]# yum install yum-utils
> [root]# yum-config-manager --disable epel
> [root]# yum install --enablerepo=epel perl-Term-ReadLine-Gnu
> ```

3.9.5 Install MAM Server

> ⓘ You *must* complete all the previous sections in this topic before installing MAM server. See the list of steps at the beginning of this topic.

On the MAM Server Host do the following:

1. If you are installing the MAM Server on its own host and *not* on the same host where you installed another server (such as Moab Server), verify you completed the steps to prepare the host. See 3.3 Preparing for RPM Installs - page 169 for more information.

2. Install the MAM Server RPM.

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

3. As the database user, create a database called mam and grant database privileges to the mam user.

   > ⓘ PostgreSQL was installed and initialized earlier in this topic. See 3.9.4.A Install and Initialize PostgreSQL Server - page 189.

   ```
   [root]# su - postgres
   [postgres]$ psql

   create database mam;
   create user mam with password 'changeme!';
   \q

   [postgres]$ exit
   ```

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

   ```
   [root]# vi /opt/mam/etc/mam-server.conf

   database.password = changeme!
   ```

4. Run the `hpc.sql` script to populate the Moab Accounting Manager database with objects, actions, and attributes necessary to function as an Accounting Manager.

```
[root]# su - mam

[mam]$ psql mam < /usr/share/moab-accounting-manager/hpc.sql
[mam]$ exit
```

5. Start the `mam` service.

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

3.9.6 Configure the MAM GUI

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

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

```
[root]# vi /etc/httpd/conf.d/ssl.conf

<Directory "/var/www/cgi-bin">
## Add these lines
     Options ExecCGI
     AddHandler cgi-script .cgi
     AllowOverride All
     Order allow,deny
     Allow from all
</Directory>

# Aliases for /cgi-bin
Alias /cgi-bin/ /var/www/cgi-bin/
Alias /mam /var/www/cgi-bin/mam/

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

2. For Red Hat-based systems where Security Enhanced Linux (SELinux) is enforced, you may need to customize SELinux to allow the web server to make network connections, use `setuid` for authentication, and write to the log file.

   a. Determine the current mode of SELinux.

```
[root]# getenforce

Enforcing
```

   - If the command returns a mode of `Disabled` or `Permissive`, or if the `getenforce` command is not found, you can skip the rest of this step.

- If the command returns a mode of `Enforcing`, you can choose between options of customizing SELinux to allow the web GUI to perform its required functions or disabling SELinux on your system.

b. If you choose to customize SELinux, do the following:

> ⓘ SELinux can vary by version and architecture and that these instructions may not work in all possible environments.

> ⓘ If you used the `--prefix=<prefix>` configuration option when you configured Moab Accounting Manager, you must replace references to `/opt/mam` in the example below with the `<prefix>` you specified. See 2.17 Moab Accounting Manager Configuration Options - page 117.

```
[root]# cat > mamgui.te <<EOF
module mamgui 1.0;
require {
     type httpd_sys_script_t;
     type port_t;
     type user_home_t;
     class capability setuid;
     class tcp_socket name_connect;
     class file read;
     class file open;
}
allow httpd_sys_script_t port_t:tcp_socket name_connect;
allow httpd_sys_script_t self:capability setuid;
allow httpd_sys_script_t user_home_t:file read;
allow httpd_sys_script_t user_home_t:file open;
EOF
[root]# checkmodule -M -m -o mamgui.mod mamgui.te
[root]# semodule_package -m mamgui.mod -o mamgui.pp
[root]# semodule -i mamgui.pp
[root]# setenforce 0
[root]# chcon -v -t httpd_sys_content_t /opt/mam/log
[root]# setenforce 1
```

3. For the highest security, it is recommended that you install a public key certificate that has been signed by a certificate authority. The exact steps to do this are specific to your distribution and the chosen certificate authority. An overview of this process for CentOS 7 is documented at https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/System_Administrators_Guide/ch-Web_Servers.html#s2-apache-mod_ssl.

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

If your configuration uses self-signed certificates, no action is required. RedHat 6 ships with ready-made certificates.

4. Start or restart the HTTP server daemon.

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

3.9.7 Configure MAM Web Services

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

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

```
[root]# vi /etc/httpd/conf.d/ssl.conf
# Place the following within the 443 VirtualHost definition
PerlOptions +Parent
PerlSwitches -Mlib=/opt/mam/lib
PerlModule MAM::WSResponseHandler
PerlModule MAM::WSAuthenHandler
<Location /mamws>
      SetHandler perl-script
      PerlResponseHandler MAM::WSResponseHandler
      Options +ExecCGI

      AuthName MAM
      PerlAuthenHandler MAM::WSAuthenHandler
      Require valid-user

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

2. For Red Hat-based systems where Security Enhanced Linux (SELinux) is enforced, you may need to customize SELinux to allow the web server to make network connections and write to the log file.

   a. Determine the current mode of SELinux.

   ```
   [root]# getenforce

   Enforcing
   ```

   - If the command returns a mode of `Disabled` or `Permissive`, or if the `getenforce` command is not found, you can skip the rest of this step.

   - If the command returns a mode of `Enforcing`, you can choose between options of customizing SELinux to allow MAM Web Services to perform its required functions or disabling SELinux on your system.

   b. If you choose to customize SELinux, do the following:

   > ⓘ SELinux can vary by version and architecture and that these instructions may not work in all possible environments.

> **ℹ** If you used the `--prefix=<prefix>` configuration option when you configured Moab Accounting Manager, you must replace references to `/opt/mam` in the example below with the `<prefix>` you specified. See 2.17 Moab Accounting Manager Configuration Options - page 117 for more information.

```
[root]# cat > mamws.te <<EOF
module mamws 1.0;
require {
     type httpd_t;
     type port_t;
     type usr_t;
     class tcp_socket name_connect;
     class file { create append };
}
allow httpd_t port_t:tcp_socket name_connect;
allow httpd_t usr_t:file { create append };
EOF
[root]# checkmodule -M -m -o mamws.mod mamws.te
[root]# semodule_package -m mamws.mod -o mamws.pp
[root]# semodule -i mamws.pp
[root]# setenforce 0
[root]# chcon -v -t httpd_sys_content_t /opt/mam/log
[root]# setenforce 1
```

3. For the highest security, it is recommended that you install a public key certificate that has been signed by a certificate authority. The exact steps to do this are specific to your distribution and the chosen certificate authority. An overview of this process for CentOS 7 is documented at https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/System_Administrators_Guide/ch-Web_Servers.html#s2-apache-mod_ssl.

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

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

4. Start or restart the HTTP server daemon.

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

3.9.8 Access the MAM GUI

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

1. Create a password for the `mam` user to be used with the MAM Web GUI.

```
[root]# su - mam
```

```
[mam]$ mam-set-password
[mam]$ exit
```

2. Verify the connection.

    a. Open a web browser and navigate to `https://<mam-server-host>/mam`.

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

3.9.9 Access MAM Web Services

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

1. Create a password for the mam user that you wish to access MAM Web Services.

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

2. Make a call to web services.

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

    Alternatively, for queries, you can use the browser to access the URL. For example:
    'https://`<mam-web-services-host>`/mamws/system'.

3.9.10 Configure Moab Workload Manager to use Moab Accounting Manager

If integrating with Moab Workload Manager, do the following, as applicable:

1. On the *Moab* Server Host, edit the Moab configuration file.

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

    a. Uncomment the AMCFG lines and customize as needed. See Accounting, Charging, and Allocation Management in the *Moab Workload Manager Administrator Guide*.

    b. If the Moab Server and the MAM Server are on the *same* host, set HOST to 'localhost'; otherwise, set HOST to the host name for the MAM Server (MAM Server Host).

2. Configure Moab to authenticate with MAM using the MAM secret key.

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

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

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

CLIENTCFG[AM:mam] KEY=<MAMSecretKey>
```

3. Restart Moab

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

3.9.11 Initialize Moab Accounting Manager

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

Related Topics

<div style="background:#1f5fa8;color:white;padding:8px 16px;font-size:1.6em;font-weight:bold;">3.10 Installing Moab Web Services</div>

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

3.10.1 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.

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

See for general instructions and an example of how to open ports in the firewall.

3.10.2 Install Dependencies, Packages, or Clients

3.10.2.A 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 Java 9, OpenJDK/IcedTea, GNU Compiler for Java, and so on cannot run Moab Web Services.

On the MWS Server Host, do the following:

1. Install the Linux x64 RPM version of Oracle Java SE 8 JRE.

   a. Go to the 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 commands:

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

3.10.2.B Install and Configure MongoDB

> 🛈 Setting per-user limits on various resources can prevent MongoDB from closing connections if the number of connections grows to high. See Review and Set Resource Limits (https://docs.mongodb.com/manual/reference/ulimit/#review-and-set-resource-limits) for more information about using the `ulimit` command to review and set resource limits.

On the MWS MongoDB Database Host, do the following:

1. Install MongoDB.

   ```
   [root]# yum install -y mongodb-org
   ```

2. Enable and start MongoDB.

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

3. Add the required MongoDB users.

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

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

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

   > use mws
   ```

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

> exit
```

> ℹ 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.

4. Set MongoDB Configuration Options.

- The configuration file for MongoDB is `/etc/mongod.conf`. See https://docs.mongodb.com/manual/reference/configuration-options for information.

- Adaptive Computing recommends that you set `security.authorization` to `enabled`. See https://docs.mongodb.com/manual/reference/configuration-options/#security-options for more information.

> ℹ By default, `/etc/mongod.conf` sets `net.bindIp` to 127.0.0.1. You will need to change this setting if the MongoDB server needs to be accessible from other hosts or from other interfaces besides loopback. See https://docs.mongodb.com/manual/reference/configuration-options/#net-options for more information.

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

5. Restart MongoDB.

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

3.10.3 Install MWS Server

> ℹ You `must` complete the tasks to install the dependencies, packages, or clients before installing MWS Server. See 3.10.2 Install Dependencies, Packages, or Clients - page 198.
>
> If your configuration uses firewalls, you *must also* open the necessary ports before installing MWS Server. See 3.10.1 Open Necessary Ports - page 198.

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

> ℹ 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.

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

   b. In the `/opt/moab/etc/moab-private.cfg` file, set the `MONGOUSER` and `MONGOPASSWORD` parameters to the MongoDB moab_user credentials you set. See 3.10.2.B Install and Configure MongoDB - page 199 earlier in this topic.

```
MONGOUSER     moab_user
MONGOPASSWORD secret2
```

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

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

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

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`.

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

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`.

```
[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
```

> ⓘ 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

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


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


...

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

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

b. Do *one* of the following:

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

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

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

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

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

ℹ 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 /opt/mws/etc/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 `/opt/mws/etc/mws-config.groovy` file. For example:

```
pam.configuration.service = "login"
```
> *This is just an example PAM configuration file name. Make sure you specify the name of the configuration file you want MWS to use.*

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

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

c. Add the `grails.mongo.username` and `grails.mongo.password` parameters to the `/opt/mws/etc/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.

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

3.10.4 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 5.5 Moab Web Services Issues - page 418.

Related Topics

- Chapter 3: RPM Installation Method - page 161
- 3.8 Installing Moab Workload Manager - page 181

3.11 Installing Moab Insight

This topic contains instructions on how to install Moab Insight (Insight).

Because Insight accumulates data for one cluster at a time, one Insight Server (daemon) should service one Moab instance.

> ⓘ Moab Workload Manager and Insight both tend to heavily consume system resources. Therefore, Adaptive Computing *requires* that the Insight Server and the Moab Workload Manager Server run on different hosts. For these installation instructions, the "Moab Server Host" refers to one host and the "Insight Server Host" refers to another host.

In this topic:

3.11.1 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.

Location	Ports	Functions	When Needed
Insight Server Host	5568	Insight Server Port	Always
Moab MongoDB Database Host	27017	Moab MongoDB Server Port	Always
Moab Server Host	5574	Moab Data Port	Always
Moab Server Host	5575	Moab Reliability Port	Always

See for general instructions and an example of how to open ports in the firewall.

3.11.2 Dependencies, Packages, or Client Installations

3.11.2.A Install Java

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

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

On the Insight 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>
    ```

3.11.2.B Install and Configure MongoDB

> ℹ️ Setting per-user limits on various resources can prevent MongoDB from closing connections if the number of connections grows to high. See Review and Set Resource Limits (https://docs.mongodb.com/manual/reference/ulimit/#review-and-set-resource-limits) for more information about using the *ulimit* command to review and set resource limits.

On the Insight MongoDB Database Host, do the following:

1. Install MongoDB.

    ```
    [root]# yum install -y mongodb-org
    ```

2. Enable and start MongoDB.

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

3. Add the required MongoDB users to the Insight MongoDB *and* Moab MongoDB; regardless of whether they share a host.

> 🛈 The passwords used below (`secret1`, `secret3`, and `secret4` are examples. Choose your own passwords for these users.

- Insight MongoDB (on the Insight MongoDB Database host)

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

> use insight
> db.createUser({"user": "insight_user", "pwd": "secret4", "roles":
["dbOwner"]})
> db.createUser({"user": "mws_user", "pwd": "secret3", "roles": ["read"]})

> exit
```

- Moab MongoDB (on the Moab MongoDB Database host)

```
[root]# mongo
> use admin
> db.auth("admin_user", "secret1")

> use moab
> db.createUser({"user": "insight_user", "pwd": "secret4", "roles": ["read"]})

> 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.

4. Set MongoDB Configuration Options.

- The configuration file for MongoDB is /etc/mongod.conf. See https://docs.mongodb.com/manual/reference/configuration-options for information.

- Adaptive Computing recommends that you set security.authorization to enabled. See https://docs.mongodb.com/manual/reference/configuration-options/#security-options for more information.

> 🛈 By default, /etc/mongod.conf sets net.bindIp to 127.0.0.1. You will need to change this setting if the MongoDB server needs to be accessible from other hosts or from other interfaces besides loopback. See https://docs.mongodb.com/manual/reference/configuration-options/#net-options for more information.

Edit the /etc/mongod.conf configuration file on *both* the Insight node and Moab Head node as follows.

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

5. Restart MongoDB.

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

3.11.3 Install Insight

You `must` complete the tasks to install the dependencies, packages, or clients before installing Insight Server. See 3.11.2 Dependencies, Packages, or Client Installations - page 207.

If your configuration uses firewalls, you *must also* open the necessary ports before installing Insight Server. See 3.11.1 Open Necessary Ports - page 206.

These instructions contain steps to edit the `/opt/insight/etc/config.groovy` file.

Commented out values in the config.groovy file are not necessarily the default values.

It is recommended that anytime you edit the config.groovy file that you first stop Insight, edit the file and then restart Insight.

1. If you have not already done so, complete the steps to prepare the Insight Server Host. See 3.3 Preparing for RPM Installs - page 169 for more information.

2. On the Insight Server Host, install the Insight RPM.

```
[root]# yum install moab-insight
```

> **ⓘ** If the installation returns the following warning line:
>
> ```
> warning: rpmts_HdrFromFdno: Header V4 RSA/SHA1 Signature, key
> ID 952741e1: NOKEY
>
> Retrieving key from file:///opt/adaptive-rpm-
> repository/key/GPG_ADAPTIVE_COMPUTING_INC_EL_6_KEY
>
> Importing GPG key 0x952741E1:
>
> Userid: "Adaptive Computing Enterprises, Inc. (EL 6 key)
> <info@adaptivecomputing.com>"
>
> From : /opt/adaptive-rpm-repository/key/GPG_ADAPTIVE_COMPUTING_
> INC_EL_6_KEY
> ```
>
> This is normal. You can safely input y and continue.

3. If you are using MWS, on the MWS Server Host, do the following:

    a. Add or edit the following parameters in the `/opt/mws/etc/mws-config.groovy` file to specify connection information for the Insight Server.

    ```
    insight.server = "<insight_server_ip_address>"
    insight.command.port = 5568
    insight.command.timeout.seconds = 5
    ```

    In this example,

    - `<insight_server_ip_address>` represents the DNS name for the host on which the Insight Server is running.

    - the default Insight command port number (5568) is used.

    See Configuration in the *Moab Web Services Reference Guide* for more information on the MWS configuration properties.

    b. Restart Tomcat.

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

4. Configure Insight's connection to the Insight MongoDB database *and* the Moab MongoDB database. On the Insight Server Host, edit `/opt/insight/etc/config.groovy` as follows:

    ```
    mongo.host="<insight mongo host>"
    mongo.port=<insight mongo port>
    mongo.username="insight_user"
    mongo.password="secret4"

    moab.mongo.host="<moab mongo host>"
    moab.mongo.port=<moab mongo port>
    moab.mongo.username="insight_user"
    ```

```
moab.mongo.password="secret4"
```

> ℹ️ Use mongo.host="localhost" when the Insight MongoDB resides on the Insight Server Host (strongly recommended).
>
> "secret4" is the password you specified when installing the mongoDB. See 3.11.2.B Install and Configure MongoDB - page 207.

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

5. On the Insight Server Host, verify that Insight runs on startup.

```
[root]# chkconfig insight on
```

6. On the Moab Server Host, configure Moab's connection to Insight.

   a. In /opt/moab/etc/moab.cfg, configure the INSIGHTENDPOINT parameter so that Moab can connect to Insight. See Moab Parameters in the *Moab Workload Manager Administrator Guide* for parameter information.

   ```
   INSIGHTENDPOINT <hostname>[:<port>]
   ```

   <hostname> is the server where Insight is located. <hostname> is required, <port> is optional.

   b. If you have not done so already when installing MWS, in /opt/moab/etc/moab-private.cfg file, configure the MESSAGEQUEUESECRETKEY parameter so that Moab can connect to Insight. See Secure communication using secret keys - page 201

   ```
   MESSAGEQUEUESECRETKEY <secret key>
   ```

   The <secret key> is required when updating the Insight configuration file later in this procedure.

   c. Check (and possibly remove) the contents of /opt/moab/spool/insight_store directory.

   ```
   [root]# ls -lh /opt/moab/spool/insight_store/
   total 146M
   -rw------- 1 root root 129M Mar 14 13:27 mwm_rmd_CABhYI.ps
   -rw------- 1 root root 2.8M Mar 13 16:28 mwm_rmd_l3F967.ps
   -rw------- 1 root root  15M Mar 14 15:14 mwm_rmd_MIaZqI.ps
   -rw------- 1 root root 183K Mar 14 15:14 mwm_rmd_MoK9w0.ps
   -rw-r--r-- 1 root root  925 Mar 14 15:14 mwm_rmd.ps
   ```

If you see files prefixed with mws_rmd this means most likely Moab was previously configured to send messages to Insight and has stored these old messages in files. If this is the first time you have started Insight then Moab will attempt to send all old messages to Insight before it sends current messages. If you have a lot of messages it can take Insight a long time to process them all. Currently running jobs will not show up in the Insight database until all the old messages are processed. If you do not care about the old messages you can simply stop Moab and delete the files in this directory.

```
[root]# service moab stop
[root]# rm /opt/moab/spool/insight_store/*.ps
```

> ⓘ If you are concerned you may have deleted messages you did not intend, be aware that Moab has a database containing information on all current jobs, and you can easily sync Insight with Moab's database. See 1.1.2 Sync Insight Database with Moab Database for more information.

d. Restart Moab in order for the new configuration parameters to take effect.

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

e. Verify that Moab is properly configured to connect to Insight.

```
[root]# mdiag -S | grep Insight
```

You should see something similar to the following:

```
ZeroMQ Insight connection is bound on port 5574 (reliability port 5575) on host
* using Insight endpoint <the insight hostname displays here>:5568
encryption is on)
ZeroMQ Insight reliable message delivery is using store file(s) up to 1024 MB in
/opt/moab/spool/insight_store/
```

7. On the Insight Server Host, configure the moab.host and messageQueue.secretKey parameters in the Insight configuration file /opt/insight/etc/config.groovy.

```
moab.host = "<moab server>"
messageQueue.secretKey = "<secret key>"
```

The <secret key> must match the secret key configured in moab-private.cfg on the Moab server for the MESSAGEQUEUESECRETKEY configuration parameter.

8. On the Insight Server Host, start Insight.

```
[root]# service insight start
```

⚠️ The first time you start Insight it will take a minute or two to create the database schema. Although 'service insight start' will quickly return OK, it is not safe to terminate Insight while this initialization is taking place. Rebooting or terminating Insight during this initialization may cause the database to not be initialized correctly.

You will know it is safe to reboot or terminate Insight if you see the following line in /opt/insight/log/insight.log.

```
2014-12-11T18:36:08.059-0700    main    INFO
com.ace.insight.app.Application 0         Started Application in 89.502
seconds (JVM running for 89.882)
```

Related Topics

- Chapter 3: RPM Installation Method - page 161

3.12 Installing Moab Viewpoint

This topic contains instructions on how to install Moab Viewpoint (Viewpoint).

Viewpoint requires a connection to Moab Server and MWS installed on the shared host. Viewpoint may also be installed on that shared host or on a different host. For documentation clarity, the instructions refer to the shared Moab Server and MWS Server host as the Moab Server Host and the host on which you install Viewpoint Server as the Viewpoint Server Host.

3.12.1 Prerequisites

3.12.1.A Adjust Security Enhanced Linux

For Red Hat-based systems where Security Enhanced Linux (SELinux) is enforced, you need to adjust SELinux to allow the web server to make network connections and create and write to the log file.

On the Viewpoint Server Host, do the following:

1. To determine the current mode of SELinux, run getenforce.

```
[root]# getenforce
```

2. If the command returns a mode of Disabled or Permissive, or if the getenforce command is not found, you can skip the rest of this procedure.

3. If the command returns a mode of Enforcing, you can choose between options of customizing SELinux to allow the web GUI to perform its required functions or disabling SELinux on your system.

- If you choose to customize SELinux:

> ℹ SELinux can vary by version and architecture and these instructions may not work in all possible environments.

```
[root]# yum install policycoreutils-python
[root]# semanage permissive -a httpd_t
```

- If you choose to disable SELinux:

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

SELINUX=disabled

[root]# setenforce 0
```

3.12.1.B 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.

Location	Ports	Functions	When Needed
Viewpoint Server Host	8081	Viewpoint Web Server Port	Always
Moab Server Host	8443	Viewpoint File Manager Port	Always
Viewpoint Database Host	5432	Viewpoint PostgreSQL Database Port	If you will be installing the Viewpoint Database on a different host from the Viewpoint Server

See 2.13 Opening Ports in a Firewall - page 106 for general instructions and an example of how to open ports in the firewall.

3.12.1.C Install and Initialize PostgreSQL Server

> ℹ The Viewpoint PostgreSQL database may be installed on the Viewpoint Server Host or on different host. If you will install on a different host, and your configuration uses firewalls, open the necessary port. See 3.12.1.B Open Necessary Ports - page 215 for more information.

On the host you have chosen to install the Viewpoint PostgreSQL database, do the following:

1. Install and initialize PostgreSQL.

```
[root]# yum install postgresql-server
[root]# service postgresql initdb
```

2. Configure trusted connections.

   Edit or add a "host" line in the pg_hba.conf file for the interface from which the Viewpoint Server will be connecting to the database and ensure that it specifies a secure password-based authentication method (for example, md5).

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

# Replace 127.0.0.1 with the IP address of the Viewpoint Server Host if the
# Viewpoint PostgreSQL server is on a separate host from the Viewpoint server.
host    all             all             127.0.0.1/32            md5
host    all             all             ::1/128                 md5
```

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

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

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

4. Start or restart the database.

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

3.12.1.D Configure the ViewpointQueryHelper Plugin

You will need to configure the MWS ViewpointQueryHelper plugin to allow Viewpoint to query the Insight MongoDB (MongoDB host, database, port, and user information).

Do the following:

1. Using a web browser, navigate to your MWS instance (`http://<server>:8080/mws/`) and then log in as the MWS administrative user (moab-admin, by default).

2. Click `Plugins` and then from the drop-down click `Plugins` to display the list of MWS plugins (displays Plugin List page).

3. Click the viewpoint-query-helper plugin to view this plugin's information (displays Show Plugin page).

4. Click `Edit` to modify the Configuration table fields (displays Edit Plugin page). The following is an example of the Edit Plugin page.

5. Modify the values as needed. The following table describes the required information.

Key	Value Description
host	Name or IP address of the host on which Insight MongoDB resides.
database	Name of the MongoDB database to which Insight writes.
port	Port number for Insight MongoDB (typically 27017).
user	User name with which MWS connects to Insight MongoDB.
password	Password used by the user listed in the value for the "user" key.

> This is the user name and password you specified when installing the Insight MongoDB. See 3.11.2.B Install and Configure MongoDB - page 207 for the user and password information.

6. When finished, click `Update` to save your changes. If you see error messages at the top of the screen (for example: `Invalid configuration for plugin viewpoint-query-helper`), go back and correct the plugin's configuration values. See Step 4 and Step 5 for more information.

7. Navigate to Plugins/Plugin Monitoring, and start the plugin using the green start button.

8. Log out of your MWS instance and close the web browser.

See also About Moab Web Services Plugins in the *Moab Web Services Reference Guide* for more information.

3.12.2 Install Viewpoint Server

> ℹ You *must* complete the prerequisite tasks earlier in this topic before installing the Viewpoint Server. See 3.12.1 Prerequisites - page 214.

Do the following:

1. If you are installing Viewpoint on its own host *or* on a host that does not have another RPM installation, complete the steps to prepare the host. See 3.3 Preparing for RPM Installs - page 169 for more information.

2. Set up PostgreSQL for Viewpoint.

> ℹ These instructions assume you will install the Viewpoint PostgreSQL database on a host which already has a PostgreSQL database installed (e.g. your Moab Server host). Depending on your system confirmation, this may be on the Moab Database Host or on some other PostgreSQL Database Host.
>
> *If* you choose to install the Viewpoint PostgreSQL database on a host that does not already have a PostgreSQL database, you will need to install the Viewpoint PostgreSQL database. See 3.12.1.C Install and Initialize PostgreSQL Server - page 215 for more information.

On the host containing the Viewpoint PostgreSQL, do the following:

```
[root]# su – postgres
[postgres]$ psql
CREATE USER moab_viewpoint WITH PASSWORD 'changeme!';
CREATE DATABASE moab_viewpoint WITH OWNER=moab_viewpoint;
\q
[postgres]$ exit
```

3. On the Moab Server Host, install the moab-viewpoint-filemanager package.

   a. Install the package.

   ```
   [root]# yum install moab-viewpoint-filemanager
   [root]# yum install python-setuptools
   ```

   b. Using the instructions in `/opt/acfileman/utils/certs-handling/Readme.txt`, follow these steps:

Step 1. Create CA (Certificate Authority).

Step 2. Create server (WebDav server) certificate and key.

Step 3. Create client certificate and key.

Step 4. Configure WebDav server.

For example:

```
[root]# cd /opt/acfileman/utils/certs-handling
[root]# ./ac-cert-tool.sh create-ca
[root]# ./ac-cert-tool.sh create-server-cert --altnames 127.0.0.1,localhost
<moab_host>
[root]# ./ac-cert-tool.sh create-client-cert
[root]# bash certs/servers/<moab_host>/install-server-certs.sh -u root:root -p
600 /opt/acfileman/etc/
[root]# vi /opt/acfileman/etc/uwsgi.ini
```

Provided you followed the above steps, your key files will have been installed in
`/opt/acfileman/etc/server-cert.pem` and `/opt/acfileman/etc/server-key.pem`. To change the location where your certificates are stored, edit the
`/opt/acfileman/etc/uwsgi.ini` file accordingly.

c. Configure the moab-viewpoint-filemanager package to start up at system boot and start the
moab-viewpoint-filemanager.

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

4. On the Moab Server Host, enable negative job priority and remote visualization features.

a. Set the ENABLENEGJOBPRIORITY parameter in `/opt/moab/etc/moab.cfg`.

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

> 🛈 You must set this Moab parameter to support Viewpoint features that enable users to
> specify user priorities for their jobs. See Advanced Settings in the *Viewpoint Reference
> Guide* for more information on enabling user priorities for jobs.

b. If using the Remote Visualization features, set the USEMOABJOBID parameter in
`/opt/moab/etc/moab.cfg`.

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

c. Restart Moab.

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

5. On the Moab Server Host, register Viewpoint as a client in MWS.

a. Edit the grails.plugin.springsecurity.oauthProvider.clients array in
   `/opt/mws/etc/mws-config.groovy` and specify a client id and a client secret.
   Leave the authorizedGrantTypes field unchanged.

> ℹ The following is a suggested script for generating the client secret:
>
> ```
> [root]# dd if=/dev/urandom count=24 bs=1 2>/dev/null | base64
> ```

```
[root]# vi /opt/mws/etc/mws-config.groovy
        grails.plugin.springsecurity.oauthProvider.clients = [
                [
                        clientId: "viewpoint",
                        clientSecret: "<ENTER-CLIENTSECRET-HERE>",
                        authorizedGrantTypes: ["password"]
                ]
        ]
```

b. Restart Tomcat.

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

6. On the Viewpoint Server Host, do the following:

   a. Install the moab-viewpoint package.

   ```
   [root]# yum install moab-viewpoint
   ```

   b. (Optional) Configure virtual hosts. The moab-viewpoint package installs a file for Apache.

   `/etc/httpd/conf.d/viewpoint.conf`

   Virtual host configurations should be made within this file. See
   http://httpd.apache.org/docs/2.2/vhosts/ for more information.

   c. Edit the `/opt/viewpoint/etc/viewpoint.cfg` values as needed. The following is an
      example of the viewpoint.cfg file with the default values.

   ```
   [admin]
   username = viewpoint-admin
   password = pbkdf2_
   sha256$20000$ZHeToCJgrSUH$+xmzYdhpqZCJokxO9eGzyr2B6jrfCgLlBT+pBgMis4w=

   [environment]
   VIEWPOINT_DATABASE_NAME = moab_viewpoint
   VIEWPOINT_DATABASE_USER = moab_viewpoint
   VIEWPOINT_DATABASE_PASSWORD = changeme!
   VIEWPOINT_DATABASE_HOST = localhost
   VIEWPOINT_DATABASE_PORT = 5432


   [settings]
   past_hours = 24
   future_hours = 4
   ```

   Be aware of the following:

- **[admin] username**: The admin username must not be the same as the auth.defaultUser.username in mws-config.groovy.

- **[admin] password**: For security purposes, the admin password is encrypted. In the example, the default is the encrypted equivalent to "changeme!", which is the default for the Viewpoint instance. Change this default password to a different encrypted password.

  To encrypt the password, do the following (substituting "changeme!" with your password):

  ```
  [root]# echo -n 'changeme!' | /opt/viewpoint/bin/viewpoint makehash
  Using default hasher
  pbkdf2_sha256$20000$ZHeToCJgrSUH$+xmzYdhpqZCJokxO9eGzyr2B6jrfCgLlBT+pBgMis4w=
  ```

  > ℹ The default hashing algorithm is pbkdf2_sha256. To show the other available algorithms, run `/opt/viewpoint/bin/viewpoint makehash --help`
  >
  > bcrypt_sha256 and bcrypt are *not* supported on Red Hat 7-based systems.

- **[environment]**: "changeme!", although unencrypted, is the default for the Viewpoint database password. If you do not change this password, your Viewpoint database will not be secure. For tips on choosing a good password, see https://www.us-cert.gov/ncas/tips/ST04-002.

- **[settings]**: These values are used to limit the threshold for the Resource Job Timeline. See Resource Job Timeline Page in the *Moab Viewpoint Reference Guide*.

> ℹ Viewpoint has a number of environment variables used to configure a Viewpoint installation and troubleshoot operating issues. See 5.6.1 General Configuration Issues - page 423 for more information about Viewpoint environment variables.

d. Initialize Viewpoint's PostgreSQL database.

```
[root]# /opt/viewpoint/bin/viewpoint migrate
```

> ℹ When running `viewpoint migrate`, `max_user_instances` should be set to at least 128. To set `max_user_instances`, execute the following command.
>
> ```
> [root]# echo "128" >  /proc/sys/fs/inotify/max_user_instances
> ```

e. Start (or restart) the Apache service.

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

3.12.3 Enable Access to the Viewpoint File Manager

This section finishes the SSL authentication steps you began when you installed moab-viewpoint-file-manager -- that is, Step 5 of `/opt/acfileman/utils/certs-handling/Readme.txt` that you skipped earlier.

Do the following:

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

```
[root]# cd /opt/acfileman/utils/certs-handling/certs
[root]# scp ca/ca-cert.pem client/client-cert.pem client/client-key.pem
root@<viewpoint_host>:/opt/viewpoint/lib/viewpoint/webdav_client
```

2. On the Viewpoint Server Host, set the mode, owner, and group of the files you copied over.

```
[root]# cd /opt/viewpoint/lib/viewpoint/webdav_client
[root]# chmod 600 ca-cert.pem client-key.pem client-cert.pem
[root]# chown apache:apache ca-cert.pem client-key.pem client-cert.pem
[root]# service httpd restart
```

3.12.4 License Viewpoint

Viewpoint uses an RLM server to validate licensed features. See 1.4 RLM Licensing Model - page 32 for more information about RLM licensing.

Do the following:

1. Using a web browser, navigate to your Viewpoint instance. (http://`<viewpoint_host>`:8081; where `<viewpoint_host>` is the IP address or name of the Viewpoint Server Host).

2. Log in as the Viewpoint administrative user (viewpoint-admin, by default) using the password you set in the Viewpoint installation instructions.

3. Click the Licensed Features link on the left side of the page.

> ⓘ The Viewpoint administrative user has very limited rights.

The Licensed Features page displays with the Viewpoint License information. For example:

4. On the Licensed Features page, locate the Viewpoint Host ID (under the Browse button).

5. Email licenses@adaptivecomputing.com with that hostid.

6. Adaptive Computing will generate the license and send you the Viewpoint license (.lic) file in a return email.

7. Save the Viewpoint license in a safe location.

8. Return to the Licensed Features page.

9. Click `Browse`, navigate to where you saved the Viewpoint License file, and then click `Open`.

10. Click `Upload`.

11. Once the license file has uploaded, the Viewpoint License information shows green check boxes for your licensed features and displays the path to your uploaded license file under the Viewpoint Host ID information. For example:
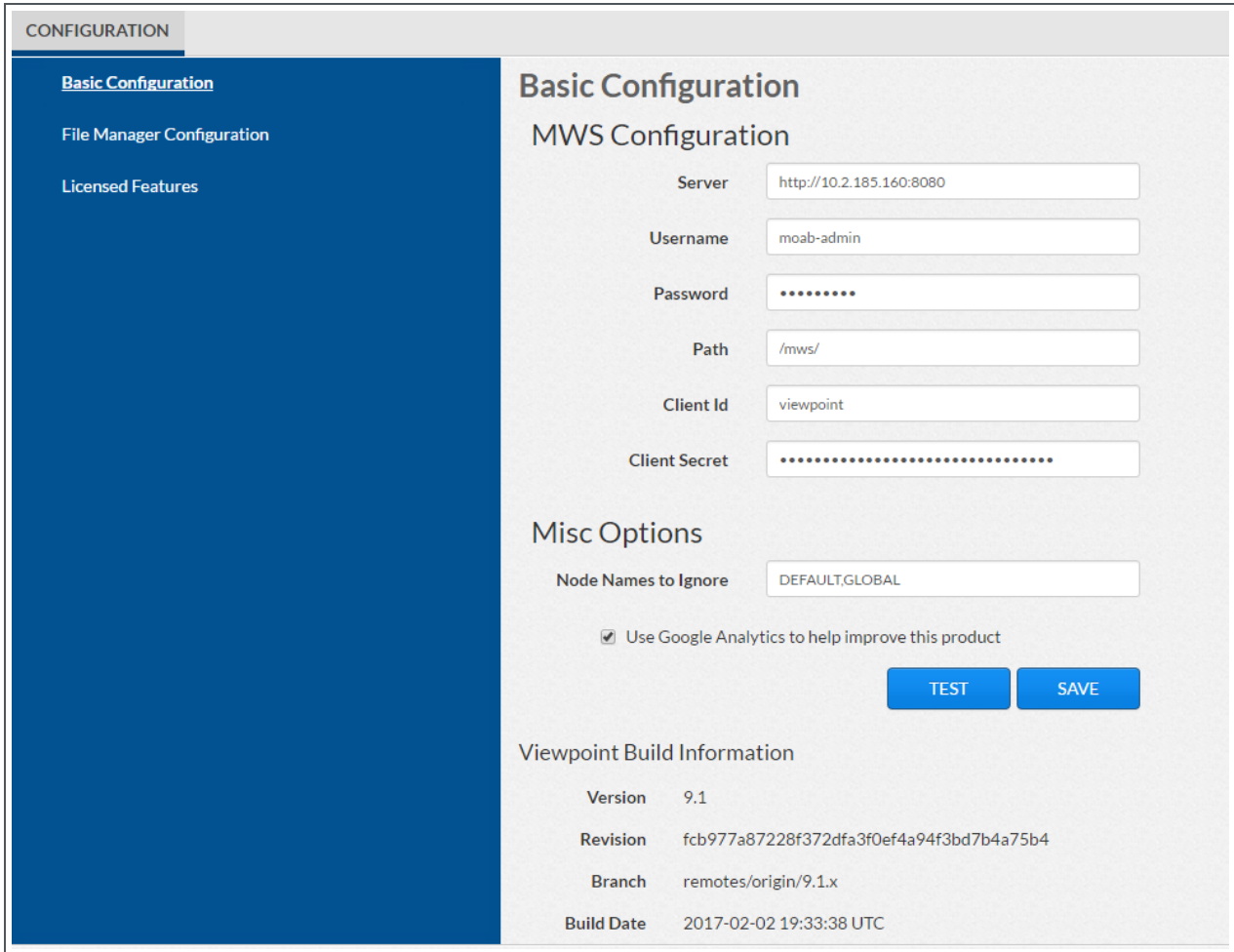
- Click `Preview` to view the contents of the license file you uploaded
- You can expand the `More Information` section to see expiration information.

3.12.5 Configure Viewpoint

Do the following:

1. While still logged in as the Viewpoint administrative user, click `Basic Configuration` from the left pane. The Basic Configuration page displays. For example:

2. In the MWS Configuration area, do the following:

   a. In the Server field, enter the URL for MWS on the Moab Server Host. For example: http://server:8080

      > ℹ️ If your configuration uses a secure connection between Viewpoint and MWS, the URL must contain "https" and the secure port.

   b. In the Username and Password fields, enter the MWS administrator credentials. You can find these credentials in `/opt/mws/etc/mws-config.groovy` on the Moab Server Host. Look for auth.defaultUser.username and auth.defaultUser.password.

   c. In the Path field, the default value (/mws/) is already filled in. Leave it as is unless you have installed MWS with a non-default path.

   d. In the Client Id and Client Secret fields, enter the values that you set during the Viewpoint installation. Refer back to the step (On the Moab Server Host, register Viewpoint as a client in MWS.) earlier in this topic.

3. In the Misc Options area, do the following:

   a. In the Node Names to Ignore field, enter the nodes that you want Viewpoint to ignore. Separate node names with a comma (,).

   b. Choose whether you wish to use Google Analytics to help improve this product.

4. Click `TEST` to confirm the settings are correct.

5. Click `SAVE` to submit your settings.

3.12.6 Configure File Manager

Do the following:

1. While still logged in as the Viewpoint administrative user, click `File Manager` from the left pane. The File Manager Configuration page displays. For example:
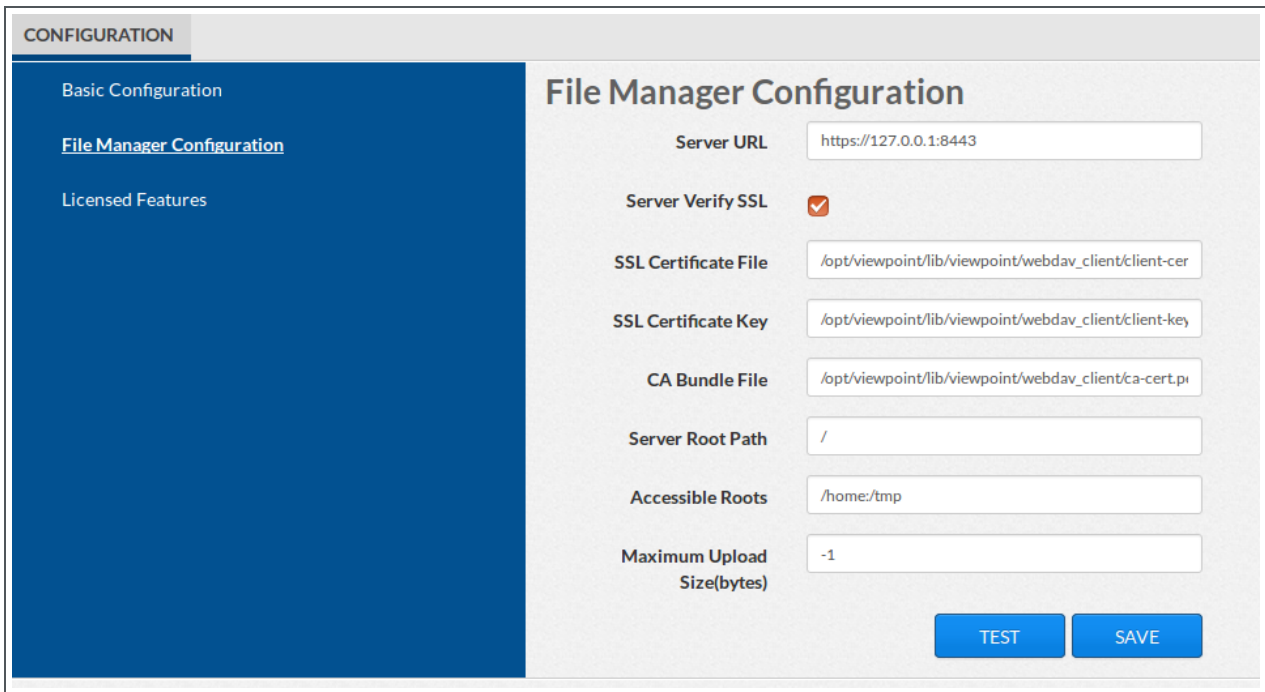
2. Modify the values as needed. The following table describes the required information.

Field	Description
Server URL	The name of the Moab Server host on which you installed the File Manager Service and the port number for the File Manager Service (for example, "https://<host name>:8443").

Field	Description
Server Verify SSL	When enabled: - The client SSL certificate will be verified. - Viewpoint will use the given certificate when connecting to File Manager Service.
SSL Certificate File	The location of the SSL certificate file on the Viewpoint Server. Usually, `/opt/viewpoint/lib/viewpoint/webdav_client/client-cert.pem`.
SSL Certificate Key	The location of the SSL certificate key on the Viewpoint Server. Usually, `/opt/viewpoint/lib/viewpoint/webdav_client/client-key.pem`.
CA Bundle File	The location of the CA bundle file on the Viewpoint Server. Usually, `/opt/viewpoint/lib/viewpoint/webdav_client/ca-cert.pem`.
Server Root Path	The root URL path where File Manager Service publishes its API (usually it is simply "/").

Accessible Roots	The root folders that users can access from the File Manager page. This can be used to limit users' access to certain directories, without giving them access to the "/" folder on the remote file system (RFS). Separate root folders with a colon (for example, /home:/usr/share/groups).
	
	For example, if you define /home and /usr/share/groups as accessible roots, although users will be able to see a tree similar to the following, the users will not be able to see (access) anything inside /usr other than "share" and anything inside "share" other than "groups".
	
	<pre>|
	- /home/
	     |
	      - user1/
	      - user2/
	      - youruser/
	- /usr/
	     |
	      - share/
	          |
	           - groups/</pre>

Maximum Upload Size (bytes)	Total amount of data that can be uploaded in a single file. A value of '-1' means unlimited.

3. Click TEST to confirm the settings are correct.
4. Click SAVE to submit your settings.

3.12.7 Verify Base Roles are Present

Viewpoint comes configured with several default (base) roles. See Differences in version 9.1 of the *Moab HPC Suite Release Notes* for more information.

As part of the Viewpoint installation, you will need to verify that the base roles are present.

Do the following:

1. Using a web browser, navigate to your Viewpoint instance. (http://<viewpoint_host>:8081; where <viewpoint_host> is the IP address or name of the Viewpoint Server Host).

2. Log in as the MWS administrative user (moab-admin, by default).

3. Click `Configuration` from the menu. The Basic Configuration page displays with additional options in the left pane. For example:

4. Click `Roles` from the left pane. The Role Management page displays.

5. If all the roles *are* there, continue with the procedure in .

   However, if one of the base roles is not present, you will need to recreate (restore) the base roles.

6. If you need to recreate the base roles, the Recreate Base Roles button displays on the Role Management page. For example:

- Click `Recreate Base Roles`. Viewpoint will restore the roles.

> ℹ️ You can also modify the default roles and create new roles as needed. See "About Roles" in the *Moab ViewpointReference Guide* for more information.

3.12.8 Grant Users Access to Viewpoint

For a user to be able to access Viewpoint, he or she must be a member of a principal.

Do the following:

1. While still logged in as the MWS administrative user, click `Configuration` from the menu. The Basic Configuration page displays with additional options in the left pane. For example:

HOME	WORKLOAD	TEMPLATES	NODES	SESSIONS	CONFIGURATION

Basic Configuration

File Manager Configuration

Roles

Principals

Remote Visualization Services

Nitro Services

Reporting Configuration

Application Templates

Licensed Features

Basic Configuration
MWS Configuration

Server	http://10.2.185.160:8080
Username	moab-admin
Password	•••••••••
Path	/mws/
Client Id	viewpoint
Client Secret	••••••••••••••••••••••••••••••

☐ Reset Permissions

Misc Options

Node Names to Ignore	DEFAULT,GLOBAL

☑ Use Google Analytics to help improve this product

[TEST] [SAVE]

Viewpoint Build Information

Version	9.1
Revision	fcb977a87228f372dfa3f0ef4a94f3bd7b4a75b4
Branch	remotes/origin/9.1.x
Build Date	2017-02-02 19:33:38 UTC

2. Click `Principals` from the left pane. The Principal Management page displays. For example:
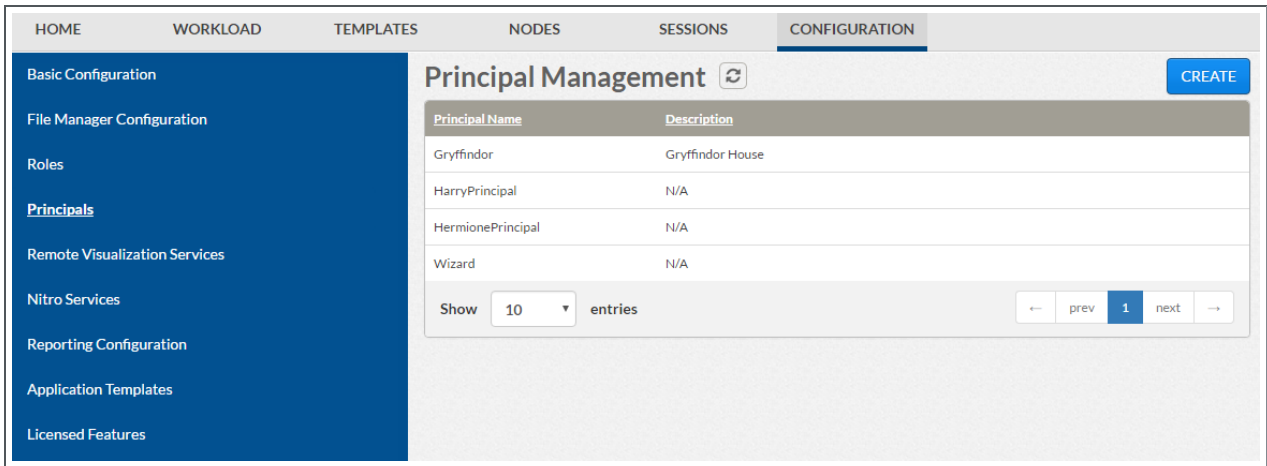
3. Click the Create button (upper right). The Create Principal page displays. For example:
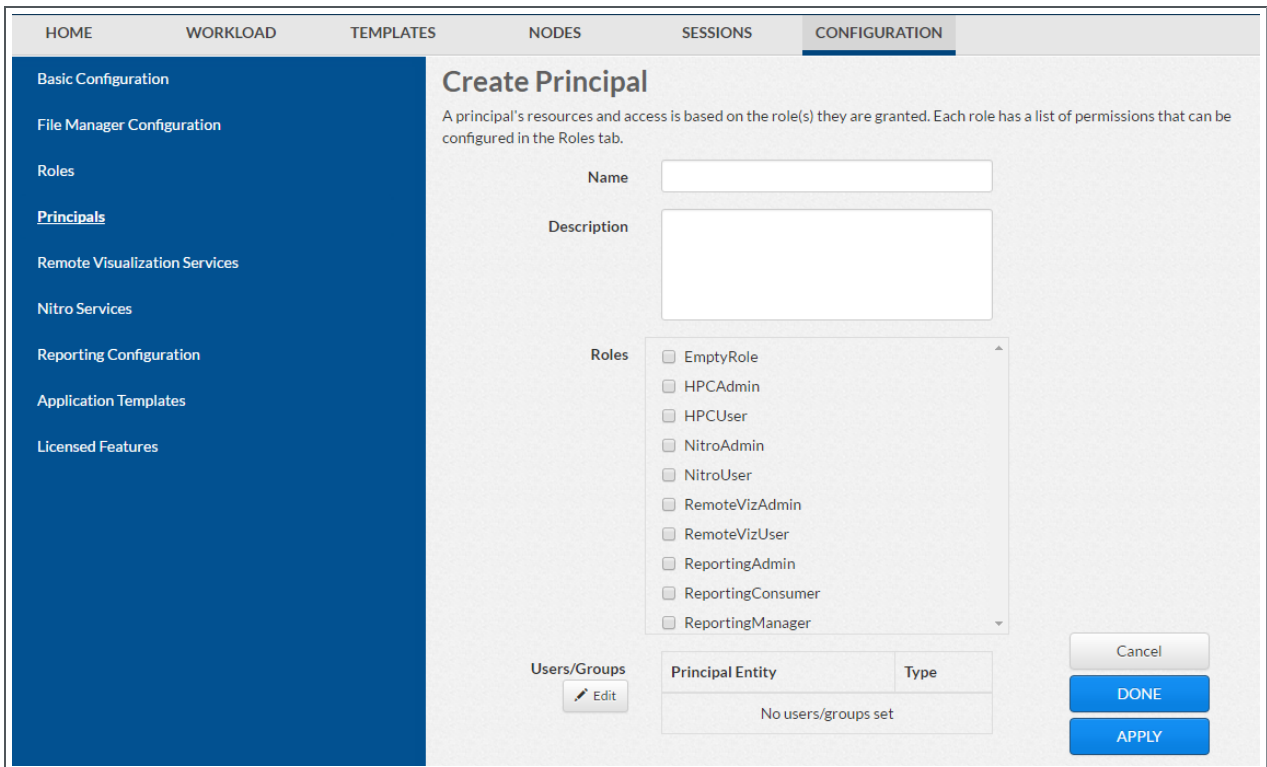
4. Create one or more principals. See Creating or Editing Principals in the *Moab Viewpoint Reference Guide* for instructions on setting up principals.

Related Topics

- Chapter 3: RPM Installation Method - page 161

3.13 Installing the Reporting Framework

This topic contains instructions on how to install the Reporting Framework.

3.13.1 Planning Your Reporting Installation

You will need to allocate one or more hosts to the Reporting framework that are separate from the hosts on which you have already installed other components of the Moab stack (e.g. Torque, Moab, MWS). A typical Reporting installation will have a dedicated master host and one or more worker hosts.

Reporting installations require the installation of many third party components across multiple hosts. New components include:

- **Apache Spark** — Consists of one Spark Master and one or more Spark Workers.
- **Apache Hadoop File System (HDFS)** — Consists of one HDFS name node and one or more HDFS data node services.
- **Apache Zookeeper** — Required to run Apache Drill and Apache Kafka.
- **Apache Drill** — For typical reporting installations, Drill is typically installed on a single host.
- **Apache Kafka** — Consists of one Kafka Master host and zero or more Kafka Broker hosts.

In addition, these components require components that should have been installed previously as part of your MWS installation, including Insight, Tomcat, MongoDB.

A common installation would have a single Reporting Master host, several Reporting Worker hosts, and several Kafka Broker hosts. Here is what will typically be installed on each host:

- **Reporting Master Host** — Spark Master, HDFS name node, Kafka Master, and Zookeeper.
- **Reporting Worker Host(s)** — Spark Worker, HDFS data node. Note that Spark Workers require large amounts of memory (32 GB is typical).
- **Kafka Broker Host(s)** — Kafka

Once you have determined which hosts will be used for the Reporting framework, follow the steps described in this document to prepare the hosts.

> ⓘ Each host must be resolvable via DNS.

Note that during the installation you will also need to modify hosts where the following Moab components are located (these components should have been installed and configured before starting the Reporting framework installation):

- **Moab/Tomcat/MWS** — In the sections below you will install Reporting Web Services into the same host where Tomcat and MWS are running.
- **Insight** — In the sections below you will configure Insight to send messages to the Kafa master service. You will also install Drill. The most typical location for Drill is on the Insight Server Host.

3.13.2 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.

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

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

See 3.20 Opening Ports in a Firewall - page 289 for general instructions and an example of how to open ports in the firewall.

3.13.3 Disable IPv6 on All Reporting Hosts

Reporting has never been tested with IPv6 and the presence of IPv6 has been shown to cause problems during installation even if IPv4 is also available. If IPv6 is available, some Reporting components will only listen on IPv6 and will ignore IPv4 connections. Therefore it is highly recommended that you disable IPv6 on hosts used for the Reporting framework. To see if IPv6 is enabled, check *ifconfig -a* and see if any interfaces support inet6.

```
[root]# ifconfig -a | grep inet6
     inet6 addr: fe80::f816:3eff:fe72:cb7f/64 Scope:Link     #means IPv6 is enabled
     inet6 addr: ::1/128 Scope:Host
```

If *ifconfig -a* returns at least one line containing inet6 this means IPv6 is enabled. To disable IPv6, add the following to /etc/sysctl.conf and run *sysctl -p*.

```
[root]# vi /etc/sysctl.conf
net.ipv6.conf.all.disable_ipv6 = 1
net.ipv6.conf.default.disable_ipv6 = 1

[root]# sysctl -p
```

Also, make sure all IPv6 entries are commented out in your /etc/hosts file. IPv6 entries start with **::1**.

```
[root]# vi /etc/hosts
#### Comment out lines starting with ::1
###::1 my-reporting-host.acme.com my-reporting-host
###::1 localhost.localdomain localhost
###::1 localhost6.localdomain6 localhost6
```

3.13.4 Install HDFS Name and Data Nodes

Spark depends on Hadoop Distributed File System (HDFS). In Reporting installations, HDFS consists of a single name node and one or more data nodes. The HDFS name node service is typically installed on the Reporting Master host. HDFS data node services are typically installed on each of the Reporting Worker nodes.

3.13.4.A Set up HDFS Name Node

On the Reporting Master host, do the following:

1. Install Hadoop.

    ```
    [root]# yum install hadoop
    ```

2. Configure HDFS.

    ```
    [root]# source /etc/profile.d/hadoop.sh
    [root]# hdfs --config $HADOOP_CONF_DIR namenode -format -force $CLUSTER_NAME
    ```

3. Start the name node service.

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

> ℹ️ You can ignore error messages that say `chown: missing operand after "/opt/hadoop-2.6.4/logs"`

4. Create the folders that the Reporting framework expects to be in HDFS.

    ```
    [root]# hadoop fs -mkdir -p /checkpoints/processingApp
    [root]# hadoop fs -chown -R spark:spark /checkpoints/
    ```

> ℹ️ On Red Hat 6-based systems you may see warnings like the following:
>
> ```
> 16/12/21 13:18:43 WARN util.NativeCodeLoader: Unable to load
> native-hadoop library for your platform... using builtin-java
> classes where applicable
> ```
>
> You can ignore these warnings.

3.13.4.B Set up HDFS Data Node(s)

You can have one or many HDFS data nodes. For Reporting installations it is typical to configure each of the Reporting Worker hosts as HDFS data nodes. However, it is possible to install the HDFS data node service on the same node as the HDFS name node service.

On each Reporting Worker host, do the following:

1. Install Hadoop.

    ```
    [root]# yum install hadoop
    ```

2. Make sure the data node knows the URI of the HDFS name node.

    ```
    [root]# vi /etc/hadoop/core-site.xml
    <property><name>fs.defaultFS/<name><value>hdfs://<hdfs_name_node_
    hostname>:8020/<value></property>
    ```

3. Start the HDFS data node service.

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

3.13.4.C Verify Hadoop Installation

To verify that Hadoop was installed successfully, do the following:

1. Open the following in a browser:

```
http://<hdfs_name_node_host>:50070
```

2. Click on the Datanodes tab and verify that your data nodes appear.

3. Run the following on the HDFS name node and verify that you see at least one data node.

```
[root]# hdfs dfsadmin -printTopology
192.168.1.3:50010 (<hdfs_data_host1>)
192.168.1.4:50010 (<hdfs_data_host2>)
```

3.13.5 Install Apache Spark

To install Apache Spark, you need to select a host to be the Spark Master and one or more hosts to be Spark Workers. Typically you will install the Spark Master service on the Reporting Master host and the Spark Worker service on the Reporting Worker hosts. However, it is possible to have a node that is both a Spark Master and a Spark Worker.

3.13.5.A Set Up the Spark Master Host

On the Reporting Master host, do the following:

1. Install Spark.

```
[root]# yum install spark
```

2. Configure the Spark Master URL to be the hostname of the Reporting Master host.

> ⚠ Use the actual hostname (which should be DNS resolvable) when specifying the Reporting Server host. Do not use localhost.

```
[root]# vi /etc/profile.d/spark.sh
export SPARK_MASTER_URL=spark://<spark_master_host>:7077

[root]# vi /etc/spark/spark-env.sh
export SPARK_MASTER_WEBUI_PORT=8082
```

```
[root]# mkdir -p /var/log/spark/
[root]# chown -R spark:spark /var/log/spark/
```

3. Increase user resource limits.

```
[root]# vi /etc/security/limits.conf
```

Add the following lines.

```
root soft core unlimited
root soft stack unlimited
root soft nproc unlimited

root hard core unlimited
root hard stack unlimited
root hard nproc unlimited

spark soft core unlimited
spark soft stack unlimited
spark soft nproc unlimited

spark hard core unlimited
spark hard stack unlimited
spark hard nproc unlimited
```

4. Exit and re-login from the terminal for the changes to take effect. Verify the changes work before starting spark-master.

```
[root]# exit
[user]$ ssh root@<reporting-master-host>
[root]# ulimit -a | grep -P "core|stack|processes"
core file size          (blocks, -c) unlimited
stack size              (kbytes, -s) unlimited
max user processes           (-u) unlimited
```

5. Start the Spark Master service.

```
[root]# chkconfig spark-master on
[root]# service spark-master start
```

6. Verify the Spark Master web interface is running by opening http://**<spark_master_host>:**8082 in a web browser.

3.13.5.B Set Up the Spark Worker Host

Spark Workers need to be installed on hosts with a lot of memory and processors. A typical installation includes the Spark Worker service running on one or more Reporting Worker hosts that are separate from the Reporting Master host. However, it is possible make the Spark Master host a Spark Worker as well. At least 32 GB of memory and 8 processors are recommended for a worker. Each Spark Worker host should have been configured as an HDFS data node, as explained in the section above.

On each host you have chosen to be a Spark Worker host, do the following:

1. Install Spark.

```
[root]# yum install spark
```

2. Configure the worker host to find the Spark Master by setting the Spark Master URL.

> ⚠ If the Spark Master and Spark Worker are on the same host, use the actual hostname where Spark Master is running (which should be DNS resolvable). Do not use localhost.

```
[root]# vi /etc/profile.d/spark.sh
export SPARK_MASTER_URL=spark://<spark_master_host>:7077

[root]# vi /etc/spark/spark-env.sh
export SPARK_WORKER_WEBUI_PORT=8083
```

3. Increase user resource limits.

```
[root]# vi /etc/security/limits.conf
```

Add the following lines.

```
root soft core unlimited
root soft stack unlimited
root soft nproc unlimited

root hard core unlimited
root hard stack unlimited
root hard nproc unlimited

spark soft core unlimited
spark soft stack unlimited
spark soft nproc unlimited

spark hard core unlimited
spark hard stack unlimited
spark hard nproc unlimited
```

4. Exit and re-login from the terminal for the changes to take effect. Verify the changes work before starting spark-worker.

```
[root]# exit
[user]$ ssh root@<reporting-worker-host>
[root]# ulimit -a | grep -P "core|stack|processes"
core file size          (blocks, -c) unlimited
stack size              (kbytes, -s) unlimited
max user processes             (-u) unlimited
```

5. Start the Spark Worker service.

```
[root]# chkconfig spark-worker on
[root]# service spark-worker start
```

3.13.5.C Verify Spark Installation

To verify that Spark has been configured correctly, do the following:

1. Verify that the Spark Master found the Spark Worker by opening `http://`**`<spark_master_`** **`host>:`**`8082` in a web browser. Make sure there is at least one worker listed and that the state of the worker is ALIVE.
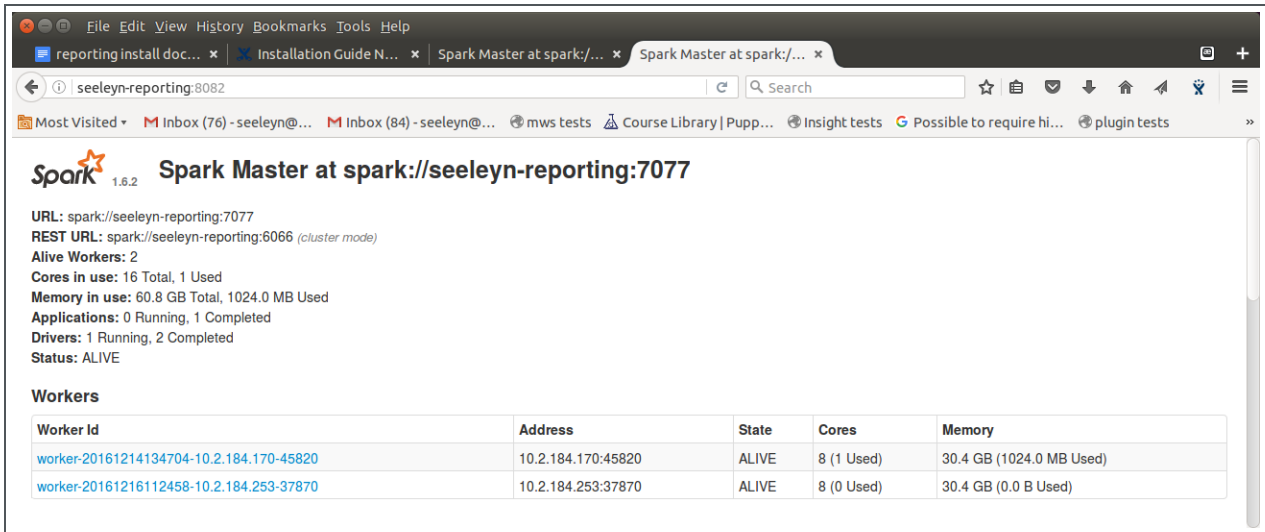
2. Verify the Spark Worker web interface is running by opening `http://`**`<spark_worker_`** **`host>:`**`8083` in a web browser.
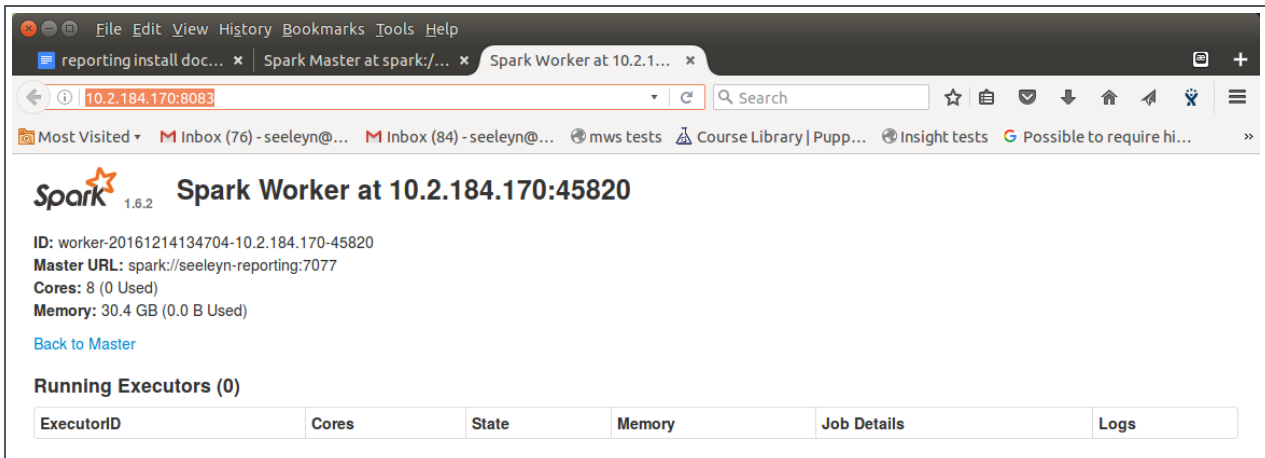
3.13.6 Install Apache Kafka

Apache Kafka can be installed on one or many hosts. At least one must be the Kafka Master. The master Kafka service is typically installed on the Reporting Master host. Verify that the IPv6 is disabled on all Kafka hosts. See 3.13.3 Disable IPv6 on All Reporting Hosts - page 235 for more information.

3.13.6.A Set Up the Kafka Master

On the Reporting Master host, do the following:

1. Install Kafka.

```
[root]# yum install kafka
```

2. Start Zookeeper

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

3. Make sure Zookeeper is listening on port 2181 for external connections, not just connections from the local machine.

    To do this run *netstat* and make sure that the local address that Zookeeper is listening on is either the machine's external IP address or 0.0.0.0, *not 127.0.0.1 or ::*. Using 127.0.0.1 or :: means external connections will likely be ignored.

```
[root]# netstat -ant | grep LISTEN | grep 2181

Bad
tcp    0      0 127.0.0.1:2181      0.0.0.0:*       LISTEN
tcp    0      0 :::2181              0.0.0.0:*       LISTEN

Good
tcp    0      0 192.168.0.4:2181    0.0.0.0:*       LISTEN
tcp    0      0 0.0.0.0:2181        0.0.0.0:*       LISTEN
```

4. Start Kafka.

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

5. Make sure Kafka is listening on port 9092 for external connections, not just connections from the local machine.

    To do this, run *netstat* and make sure that the local address Kafka is listening on is either the machine's external IP address or 0.0.0.0, *not 127.0.0.1 or ::*. Using 127.0.0.1 or :: means external connections will be ignored. In the examples below, 192.168.0.4 is the external IP address of the host.

```
[root]# netstat -ant | grep LISTEN | grep 9092

Bad
tcp    0      0 127.0.0.1:9092      0.0.0.0:*       LISTEN
tcp    0      0 :::9092              0.0.0.0:*       LISTEN

Good
tcp    0      0 192.168.0.4:9092    0.0.0.0:*       LISTEN
tcp    0      0 0.0.0.0:9092        0.0.0.0:*       LISTEN
```

3.13.6.B Set Up the Kafka Broker Hosts

Additional Kafka Broker hosts (besides the Kafka Master) are optional. If you choose to create additional Kafka Brokers you will first need to install Kafka on each of the Kafka Broker hosts.

On each host you have selected to be a Kafka Broker host, do the following:

1. Install Kafka.

```
[root]# yum install kafka
```

2. Configure Kafka to connect to Zookeeper on the Kafka Master host. Give each Kafka Broker a unique ID.

```
[root]# vi /etc/kafka/server.properties

broker.id=-1 # Allow auto-generation of a unique ID
zookeeper.connect=<kafka master host>:2181
```

3. Start Kafka.

```
[root]# service kafka start
```

3.13.6.C Determine the Number of Kafka Partitions Needed

You will first need to figure out how many partitions will be needed for the Insight topic. Determining the number of partitions does not need to be exact, but there should be at least as many partitions as Spark executors. If you have more executors than partitions, then the excess executors will not process any messages. Note that you can have more partitions than executors, because multiple partitions can be mapped to a single executor.

For example, say you have 10 partitions. You can have anywhere from 1 to 10 executors and each executor will receive messages. If you have only one executor, then all 10 partitions will be serviced by the single executor. Likewise if you have only five executors, then each executor will process two partitions. However, if you add an 11th executor, then only 10 of your 11 executors will process messages, because you only have 10 partitions. The 11th executor will essentially be wasted. See the Kafka documentation for more details about Kafka partitions

You can estimate the maximum number of Spark executors on each reporting worker host by dividing the number of cores by the number of cores per executor. For example, if you have configured `reporting.cluster.executorCores=4` in your reporting web services `application.properties` (see 3.13.10 Install and Configure Reporting Web Services - page 250), and you have 16 cores on each your three reporting worker hosts, then the max number of executors can be calculated as shown below.

```
(3 workers * 16 cores/worker) / (4 cores/executor) = 12 executors
```

You may have fewer executors if, say, memory is the constraining factor rather than cores, but you cannot have more than 12. So you will need *at least* 12 partitions. Once you have completed the installation, you will be able to count the number of executors by visiting the web interface of each

Spark worker (`http://<spark_worker_host>:8083`) and viewing the "Running Executors" section. If you estimated incorrectly or need to add partitions after installation, see 3.13.6.E Adding Kafka Partitions - page 243.

3.13.6.D Create the Insight Topic

On the Reporting Master host, create the Insight topic.

```
[root]# source /etc/profile.d/kafka.sh
[root]# kafka-topics.sh --create --zookeeper localhost:2181 --replication-factor
<number of kafka hosts> --partitions <number of partitions> --topic insight
```

> **ℹ** For the purpose of setting the replication factor, the number of Kafka hosts referred to here should include the Kafka master. For example, if you have installed one Kafka broker in addition to the Kafka master, use a value of 2 as the argument to --replication-factor.

3.13.6.E Adding Kafka Partitions

You should have at least as many partitions as you have Spark executors. If you add Spark worker nodes (and thus add Spark executors), you may need to increase the number of partitions on the Insight Kafka topic. First, calculate the number of partitions you need as explained in 3.13.6.C Determine the Number of Kafka Partitions Needed - page 242. Then check to see how many partitions you already have, as shown below.

```
[root]# source /etc/profile.d/kafka.sh
[root]# kafka-topics.sh --zookeeper localhost:2181 --describe --topic insight | grep -
i PartitionCount
Topic:insight   PartitionCount:15
```

To modify the number of partitions run the following on the Kafka master node.

```
[root]# kafka-topics.sh --zookeeper localhost:2181 --topic insight --alter --
partitions <desired number of partitions>
```

3.13.6.F Configure Insight to Send Messages to Kafka

On the Insight host, do the following:

1. Edit the Insight configuration file and enable Kafka.

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

   kafka.enabled=true
   kafka.bootstrap.servers="<kafka_master_host>:9092"
   ```

2. Restart Insight.

   ```
   [root]# service insight restart
   ```

3.13.6.G Verify Kafka Installation

To verify that Kafka is configured correctly, do the following:

1. Verify that messages are getting sent to Kafka. On the Kafka Master host (Assuming Zookeeper is running on the Kafka Master host) run the following.

```
[root]# source /etc/profile.d/kafka.sh
[root]# kafka-console-consumer.sh --zookeeper localhost:2181 --topic insight
```

2. Wait at least two minutes. After this time you should see JSON messages occasionally streaming to your console. If you do not, debug by logging into your Insight host and running the following to send messages to Kafka from the command line.

```
[root]# source /etc/profile.d/kafka.sh
[root]# kafka-console-producer.sh --broker-list <kafka_master_host>:9092 --topic insight
> Here is a sample kafka message
```

3. If the sample Kafka message does not show up in the console running the kafka-console-consumer command, then Kafka on the Insight host is unable to send messages to the Kafka host. Be sure to check that port 9092 is opened on the Kafka host and is listening for external connections. Use <Ctrl-C> to end the Kafka console consumer command after verification is completed.

3.13.7 Install and Configure MongoDB

For the Reporting framework, you can either use the same MongoDB instance that Insight is using or you can install MongoDB on a different host.

> ℹ️ Setting per-user limits on various resources can prevent MongoDB from closing connections if the number of connections grows to high. See Review and Set Resource Limits (https://-docs.mongodb.com/manual/reference/ulimit/#review-and-set-resource-limits) for more information about using the *ulimit* command to review and set resource limits.

On the Reporting MongoDB Database host, do the following:

1. Install MongoDB.

```
[root]# yum install mongodb-org
```

2. Enable and start MongoDB.

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

3. Create an admin user.

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

4. Create the Reporting database and Reporting user.

```
[root]# mongo admin
mongo> db.auth("admin_user", "secret1")
mongo> use reporting
mongo> db.createUser({user:"reporting_user", pwd:"secret6", roles:["dbOwner"]})
mongo> exit
```

5. Verify that MongoDB can accept connections from other hosts. To do this, you must configure MongoDB to listen on all interfaces by changing or commenting out the bindIp parameter in the MongoDB configuration file.

```
[root]# vi /etc/mongod.conf

#bindIp: 127.0.0.1   #Comment this line out to tell Mongo to listen on all
inferfaces
```

6. Restart MongoDB.

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

3.13.8 Install and Configure Apache Drill

Although Apache Drill can make use of many hosts in a clustered environment, for typical reporting installations Drill is installed on a single host. We recommend installing Drill on the Insight Server host.

On the Insight Server host, do the following:

1. Install Drill.

```
[root]# yum install drill
```

2. Configure the zk.connect option with the name of the host running zookeeper (which is also most likely the name of the Kafka Master host), so that Drill can communicate with Zookeeper. (Zookeeper should have been installed with Kafka.)

```
[root]# vi /etc/drill/drill-override.conf

drill.exec: {
    cluster-id: "reporting-drill"
    zk.connect: "<zookeeper_host>:2181"
}
```

3. Start the Drill service.

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

4. Verify that Drill started correctly by opening the following in a web browser:

```
http://<drill_host>:8047
```

If Drill is not running:

a. Check its log file.

```
[root]# vi /opt/drill-1.8.0/log/drillbit.out
```

b. Search for an error similar to:

```
Caused by: org.apache.drill.exec.exception.DrillbitStartupException: Drillbit is
disallowed to bind to loopback address in distributed mode
```

If found, make sure the hostname is resolvable via DNS. Make sure /etc/hosts isn't setting the IP address of the hostname to 127.0.0.1.

```
[root]# hostname
myreportinghost

[root]# vi /etc/hosts
###127.0.0.1 myreportinghost  ##Comment out or drill will not start

[root]# hostname --ip
192.168.0.3     #Should NOT be 127.0.0.1
```

c. Restart Drill.

```
[root]# service drill restart
```

5. Create the necessary Linux groups and users.

```
[root]# groupadd drill
[root]# useradd  -G drill -d /home/drilluser -m drilluser
[root]# useradd  -G drill -d /home/drilladmin -m drilladmin
```

6. Set the user's passwords.

```
[root]# passwd drilluser
changeme!

[root]# passwd drilladmin
changeme!
```

ⓘ changeme! is an example. You can set the password to any value you want.

7. Configure Drill.

```
[root]# curl -X POST -H "Content-Type: application/json" -d '
{"name":"mongo","config":
{"type":"mongo","enabled":true,"connection":"mongodb://reporting_
user:secret6@<reporting_mongodb_database_host>:27017/reporting"}}'
http://localhost:8047/storage/mongoPlugin.json

[root]# curl -X POST -H "Content-Type: application/json" -d '{ "queryType" : "SQL",
"query" : "ALTER SYSTEM SET `security.admin.users` = '"'"'drilladmin'"'"'" }'
http://localhost:8047/query.json

[root]# curl -X POST -H "Content-Type: application/json" -d '{ "queryType" : "SQL",
"query" : "ALTER SYSTEM SET `store.mongo.read_numbers_as_double` = true" }'
http://localhost:8047/query.json
```

8. Install and configure JPam. JPam allows you to log into the Drill web UI as the drilladmin and drilluser Linux users you created earlier.

   a. Install JPam.

   ```
   [root]# mkdir /opt/pam
   [root]# curl http://vorboss.dl.sourceforge.net/project/jpam/jpam/jpam-1.1/JPam-
   Linux_amd64-1.1.tgz | \
       tar -xz -C /opt/pam --strip-components=1 JPam-1.1/libjpam.so
   ```

   b. Enable Drillbit to find and use JPam.

   ```
   [root]# vi /etc/drill/drill-env.sh
   export DRILLBIT_JAVA_OPTS="-Djava.library.path=/opt/pam/"
   ```

   c. Enable security authentication by appending the following to the Drill configuration file.

   ```
   [root]# vi /etc/drill/drill-override.conf
   drill.exec: { security.user.auth { enabled: true, packages +=
   org.apache.drill.exec.rpc.user.security, impl: pam, pam_profiles: [ sudo, login
   ] } }
   ```

   You may have multiple drill.exec stanzas or they may be combined as in the following example.

   ```
   drill.exec: {
       cluster-id: "reporting-drill"
       zk.connect: "install-reporting:2181"
       security.user.auth {
           enabled: true
           packages += org.apache.drill.exec.rpc.user.security
           impl: pam
           pam_profiles: [ sudo, login ]
       }
   }
   ```

9. Restart Drill.

   ```
   [root]# service drill restart
   ```

10. Verify that Drill is running and is able to query MongoDB. To do this, first create a test MongoDB collection that you can query with Drill to verify you have configured Drill correctly.

On the MongoDB host with the Reporting database do the following:

```
[root]# mongo reporting
> db.auth("reporting_user","secret6")
> db.drilltest.insert({"test":"you can delete this collection once drill is
working"})
> exit
```

11. Use a web browser to navigate to http://**<drill_host>**:8047 and verify you can see the Drill web interface. Log in as the drilladmin user you configured above.

```
Username: drilladmin
Password: changeme!
```

12. Click Query and Select SQL. In the Query text box enter:

```
SELECT * FROM mongo.reporting.drilltest limit 10
```

13. Click Submit. You should see the sample data you submitted to MongoDB.
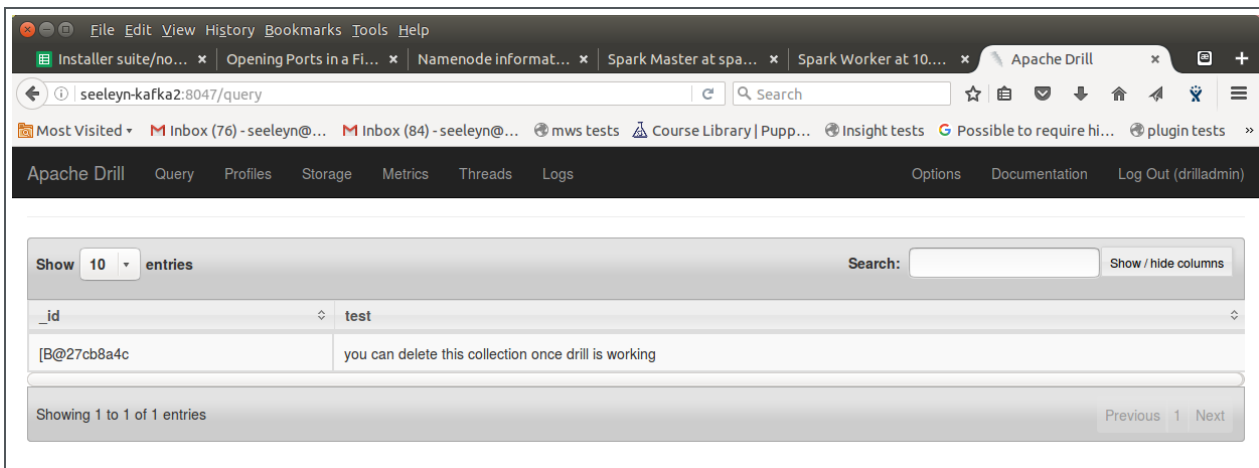
3.13.9 Install and Configure the Reporting Application

The Viewpoint Reporting application should be installed on the Spark Master host, where the HDFS name node service was previously installed.

On the Reporting Master host, do the following.

1. Install the Reporting application.

```
[root]# yum install moab-viewpoint-reporting
```

2. Edit the Reporting application.conf file and replace the references to localhost with the hostnames corresponding to the host running the service the reference is for.

> ⓘ Assume **<kafka_master_host>** refers to the hostname of the host where Kafka Master is running, **<zookeeper_host>** is the hostname of the host where Zookeeper is running, **<hdfs_name_node>** is the host where the HDFS name node service is running, **<mongo_host>** is the host where MongoDB is running, and **secret6** is the password reporting_user uses to authenticate to MongoDB.

> ⓘ In order to change the `application.conf` file after the Reporting Spark application has started, you will need to follow the steps described in 3.26 Updating the Reporting Application Configuration - page 302.

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

kafka {
    brokers = "<kafka_master_host>:9092"
}

...

zookeeper {
    hosts = "<zookeeper_host>:2181"
    ...
}

application {
    namenodeURI = "hdfs://<hdfs_name_node>:8020"
}

...

database {
    host = "<mongo_host>"
    ...
    username = "reporting_user"

    password = "secret6"
}
```

3. Upload the Reporting jar and configuration file to the Reporting folder on HDFS.

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

4. Verify that the files were uploaded successfully to HDFS.

```
[root]# hadoop fs -ls /reporting
-rw-r--r--   3 spark spark      1061 2016-12-13 15:32 /reporting/application.conf
-rw-r--r--   3 spark spark 122941004 2016-12-13 15:32 /reporting/reporting-
framework.jar
```

> ⓘ At this point, the Reporting application will not have been deployed to Spark. This will happen when Reporting Web Services starts. (See next section.)

3.13.10 Install and Configure Reporting Web Services

Reporting Web Services must be deployed on a host where Tomcat is running. These instructions assume you are installing Reporting Web Services to the same host where MWS is running, since MWS (which should have already been installed) is also running in Tomcat.

On the RWS Server host (typically the same as the MWS Server host), do the following:

1. Install Reporting Web Services.

```
[root]# yum install moab-reporting-web-services
```

2. Configure Reporting Web Services.

   a. Configure hostnames and passwords.

```
[root]# vi /opt/reporting-web-services/etc/application.properties

reporting.cluster.restUrl=http://<spark_master_host>:6066
reporting.cluster.masterUrl=spark://<spark_master_host>:7077
reporting.cluster.namenodeUri=hdfs://<hdfs_name_node_host>:8020

…
reporting.rest.drill.hostname=<drill_host>
reporting.rest.drill.username=drilluser
reporting.rest.drill.password=changeme!

mws.url=http://<mws_host>:8080/mws

spring.data.mongodb.host=<reporting_mongodb_database_host>
spring.data.mongodb.port=27017
spring.data.mongodb.database=reporting
spring.data.mongodb.username=reporting_user
spring.data.mongodb.password=secret6
```

   b. Configure cores/memory for the Spark driver and executors in the Reporting Web Services configuration file.

   The reporting application requires a single Spark driver and one or more Spark executors. It is recommended that you allocate at least 2 cores and 2 GB of memory for the driver and at least 4 cores and 6 GB of memory per executor. The following is a typical configuration:

```
[root]# vi /opt/reporting-web-services/etc/application.properties

reporting.cluster.driverMemory=2g
reporting.cluster.driverCores=2
reporting.cluster.executorCores=4
reporting.cluster.executorMemory=6g
```

   There must be enough cores and memory available in the Spark worker nodes to run the driver and at least one executor. For example, if you have only one reporting worker host and this host has 8 processors and 32 GB of memory. The sum of driver cores plus executor cores must not exceed 8. Likewise if your lone reporting worker host has 32 GB of memory then the sum of driver memory plus executor memory must not exceed 32 GB.

3. Restart Tomcat.

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

4. Verify that Reporting Web Services is running.

```
[root]# curl -X GET -v localhost:8080/rws/ping
< HTTP/1.1 200 OK
```

If the ping was not successful, check the logs at `/var/log/tomcat` for error messages.

When Reporting Web Services starts, it deploys the Reporting Data Processing application (described in the previous section) to Spark. Verify this has happened by opening `http://<spark_master_host>:8082` in a web browser. In the Running Applications section, you should see ReportingDataProcessing is in RUNNING state.

3.13.11 Configure Spark Submission Parameters (Optional)

You can override Spark submission parameters (like `spark.driver.memory` or `spark.cleaner.ttl`) by specifying parameter settings in the `application.properties` file. Doing so allows you to change Spark submission parameters from their default values without rebuilding RWS.

> ⚠️ Be extremely careful changing Spark settings since an incorrect value can cause the Reporting Framework to malfunction. There are dozens of Spark settings for many possible cases, so make sure you understand exactly what you are doing before making changes and be sure to test your reports afterward.

To set a Spark configuration parameter, edit the `application.properties` file and add a `reporting.cluster.sparkOptions` setting set to a JSON string containing the parameter setting(s). An example is shown below.

```
[root]# vi /opt/reporting-web-services/etc/application.properties

reporting.cluster.sparkOptions={"spark.streaming.kafka.maxRatePerPartition":"10",
"spark.streaming.backpressure.enabled": "true"}
```

After changing the `application.properties` file, you must restart Tomcat for the changes to take effect.

3.13.12 Connect Viewpoint to Reporting

Once the Reporting application is running, the final step is to connect Viewpoint to Reporting.

To connect Viewpoint to the Reporting application, do the following:

1. Log in to Viewpoint as moab-admin.

2. Go to the Configuration page, click Reporting Configuration, and enter the following URL:

```
http://<reporting_web_services_host>:8080/rws
```
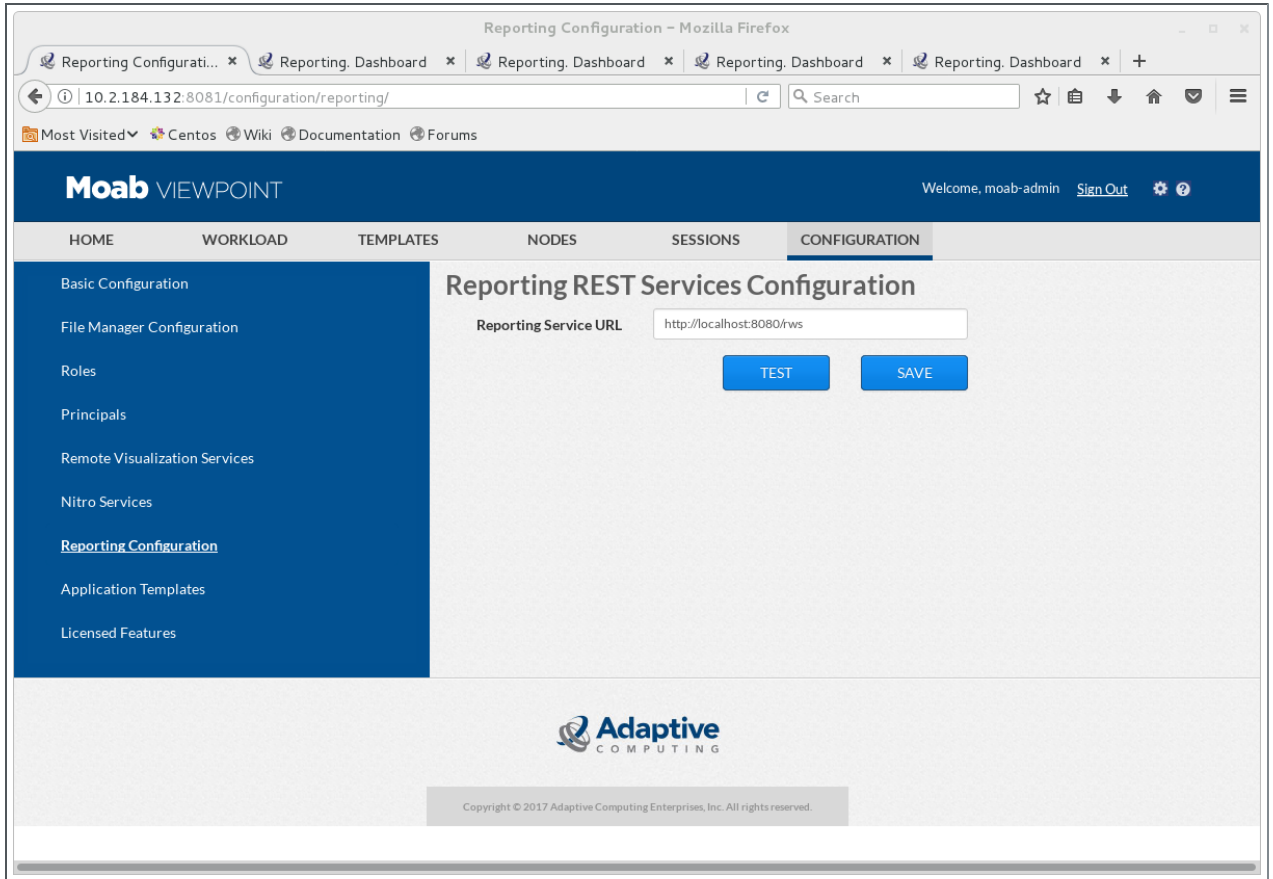
3. Click `TEST` to confirm that the setting is correct.

4. Click `SAVE` to save the setting.

5. Set up one or more reporting administrators in Viewpoint.

   a. On the Configuration page, click Principals.

   b. Hover over the principal you want to make a reporting administrator and click Modify on the pop-up menu.

   c. Verify that the principal has been granted the ReportingAdmin, ReportingManager, and ReportingConsumer roles.

   d. Click Done.

6. Log out of Viewpoint.

7. Log into Viewpoint as a user belonging to one of the principals to which you granted reporting permissions.

8. Verify that Viewpoint is communicating with the Reporting application.

   a. Click Reporting to go to the Reporting page.

   b. Click Aggregated Views.

c. Verify that the status of processing application is RUNNING.
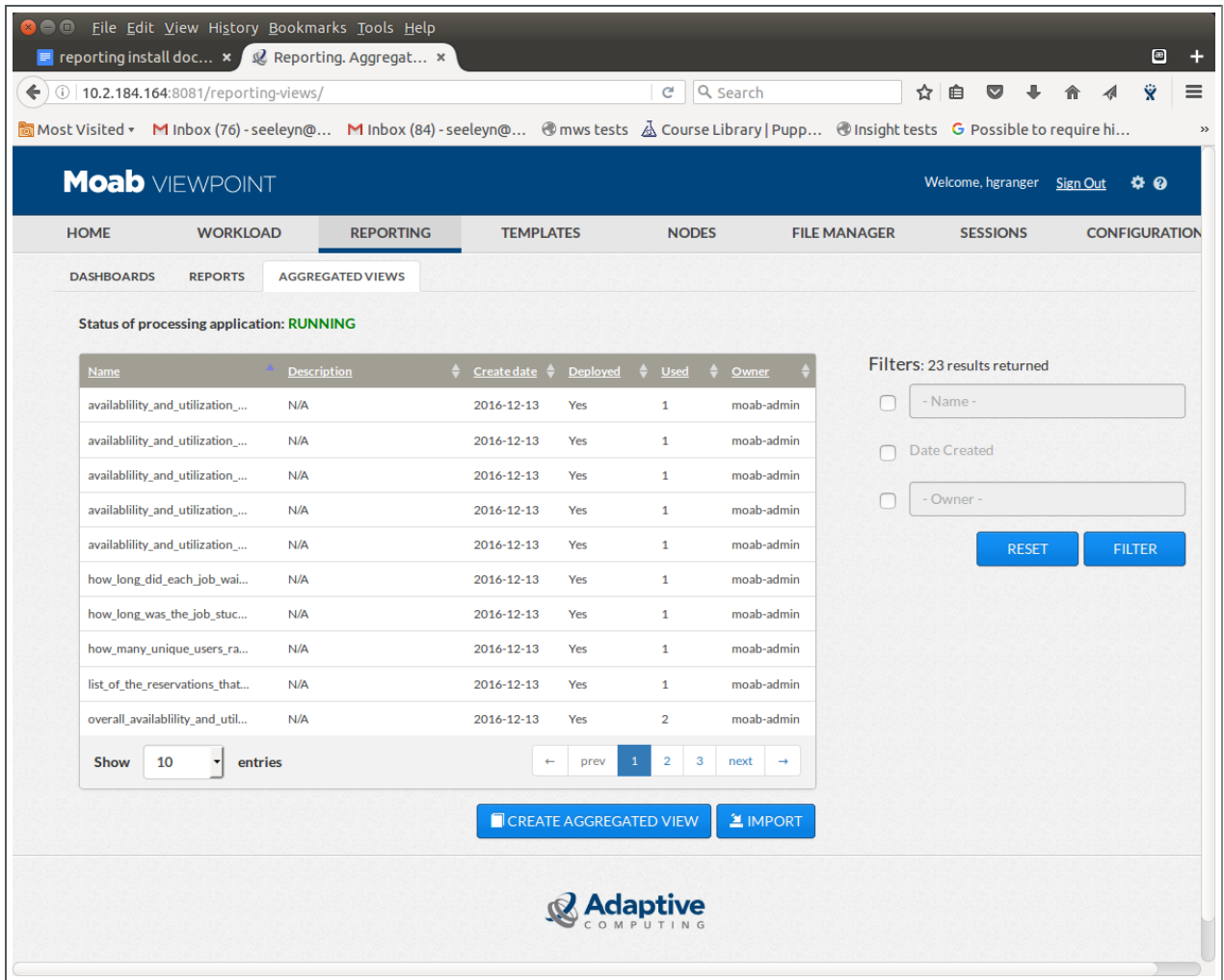
d. Click Reports.

e. At the bottom of the list of reports, click Next until you find the Node state/outage report.

f. Click on the Node state/outage report. A line graph should appear, indicating that the Reporting application is installed correctly and communicating with Viewpoint.

Related Topics

3.14 Installing RLM Server

Access to a Reprise License Manager (RLM) server is required when using Moab's Elastic Computing Feature, Viewpoint's Remote Visualization Feature, or Nitro.

As the RLM Server can run multiple licenses, it is recommended that you install *one* RLM Server for your configuration. If your company already uses an RLM Server, you do not need to install a new one for Adaptive Computing products. However, Adaptive Computing *strongly* recommends that your RLM Server is version 12.1.2 and the Adaptive Computing products may use a different port than the default RLM Server port (5053).

> ⚠ If your system configuration requires more than one RLM Server, additional configuration may be needed. See 3.23 Using Multiple RLM Servers - page 298 for more information.

This topic contains instructions on how to install an RLM Server.

> In this topic:

3.14.1 Open Necessary Ports

> ℹ These instructions assume you are using the default ports. If your configuration will use other ports, then substitute your port numbers when opening the 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.

Location	Ports	Functions	When Needed
RLM Server Host	5053	RLM Server Port	Always
RLM Server Host	5054	RLM Web Interface Port	Always
RLM Server Host	57889	Remote Visualization Port	If Remote Visualization is part of your configuration
RLM Server Host	5135	ISV adaptiveco Port (for the Adaptive license-enabled products)	For Moab Workload Manager *and* if Nitro is part of your configuration.

See 2.13 Opening Ports in a Firewall - page 106 for general instructions and an example of how to open ports in the firewall.

3.14.2 Install the RLM Server

On the host on where the RLM Server will reside, do the following:

1. If you are installing RLM Server on its own host *or* on a host that does not have another RPM installation, complete the steps to prepare the host. See 3.3 Preparing for RPM Installs - page 169 for more information.

2. If your configuration uses firewalls, you *must also* open the necessary ports before installing the RLM Server. See 3.14.1 Open Necessary Ports - page 255.

3. Install the RPM.

```
[root]# yum install ac-rlm
```

3.14.3 Change the Default Passwords

The RLM Web interface includes two usernames (admin and user) by default. These usernames have the default password "changeme!".

> ⚠ If you do not change this password, RLM, and Remote Visualization, will not be secure. For tips on choosing a good password, see https://www.us-cert.gov/ncas/tips/ST04-002.

Do the following for both the user and the admin usernames:

1. Using a web browser, navigate to your RLM instance. (http://<RLM_host>:5054; where <RLM_host> is the IP address or name of the RLM Server Host).

> ⓘ If you have problems connecting using the web browser, on the RLM server check the /opt/rlm/rlm.dl log file for error information.

2. Log in.

3. Select Change Password and change the password according to your password security process.

> ⓘ The password for "user" will be needed as part of the Remote Visualization installation.

3.15 Installing Remote Visualization

This topic contains instructions on how to install Remote Visualization, including licensing and configuration information.

ⓘ Remote Visualization uses the FastX product. The Remote Visualization installation includes installing the Remote Visualization Server (gateway server) and Remote Visualization on the Torque MOM Hosts (session servers).

⚠ Remote Visualization Server (gateway server) and the Remote Visualization Session Servers, must be configured in order for Remote Visualization to work.

In this topic:

3.15.1 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.

Location	Ports	Functions	When Needed
Remote Visualization Server Host (also known as the Gateway Server)	3443	FastX Web Server Port	Always

Location	Ports	Functions	When Needed
Remote Visualization Session Server Host (Torque MOM Host)	Add ports as required, e.g. TCP: 3443, 6000-6005, 16001, 35091 UDP: 117	Session Server Ports	Ports 16001 and 35091 are *only* needed when using gnome

See 2.13 Opening Ports in a Firewall - page 106 for general instructions and an example of how to open ports in the firewall.

3.15.2 Obtain and Install the Remote Visualization License

Remote Visualization uses an RLM Server to validate the amount of open and available sessions. See 1.4 RLM Licensing Model - page 32 for more information about RLM licensing.

> ⚠️ These instructions assume you already have access to an RLM Server. See 3.14 Installing RLM Server - page 255 for instructions on how to set up a new RLM Server.

Do the following:

1. Email licenses@adaptivecomputing.com and request an activation key. Adaptive Computing will send you the activation key in a return email.

2. Once you have your activation key, do the following on the RLM Server:

    a. Install the license activation script and dependencies.

    ```
    [root]# yum -y install perl-Crypt-SSLeay StarNetFastX2
    ```

    b. Run the license activation script.

    ```
    [root]# /usr/lib/fastx2/install/activate
    ```

    c. When prompted:

       - Enter the activation key.

       - Enter how many seats (sessions) you want for this license.

    When the license has generated you will see something similar to the following on the last line:

    ```
    License activated and saved in /usr/lib/fastx2/rlm/FastX2-<date>.lic
    ```

    d. Move the license file to the /opt/rlm directory.

    ```
    [root]# mv /usr/lib/fastx2/rlm/FastX2-<date>.lic /opt/rlm
    ```

> **ⓘ** This license file references the default RLM Server port (5053). If the RLM Server in your configuration uses a different port, you will need to modify the license file to reflect the actual port.

    e. If you did *not* install an RLM Server using the file available from Adaptive Computing (for example, because your system configuration already uses one), do the following:

        i. Download the 'starnet.set' file from the Adaptive Computing Moab HPC Suite Download Center (https://www.adaptivecomputing.com/support/download-center/moab-hpc-suite-download/).

       ii. Copy the 'starnet.set' file into the same directory where the Remote Visualization license resides (/opt/rlm).

    f. Disable the FastX RLM service.

```
[root]# chkconfig --del fastx_rlm
```

    g. Restart RLM.

```
[root]# service rlm restart
```

3.15.3 Configure the RLM Plugin

Moab can schedule available remote visualization sessions by querying the RLM server for the number of active and total available sessions.

> **ⓘ** In order for Moab to schedule remote visualization sessions, Moab also needs to be configured to use Moab Web Services as a resource manager. See Configuring Moab Workload Manager in the *Moab Web Services Reference Guide* for more information.

Do the following:

1. Using a web browser, navigate to your MWS instance (`http://<server>:8080/mws/`) and then log in as the MWS administrative user (moab-admin, by default).

2. Select `Plugins` and then from the drop-down select `Plugins` to display the list of MWS plugins (displays Plugin List page).

3. Click `Add Plugin` (displays Create Plugin page).

4. Select `RLM` from the Plugin Type drop-down.

5. Click `Continue` (displays the already built information for this plugin on the Create Plugin page).

6. In the Configuration field, select `Resource` from the drop-down and then click `Add Entry` (adds the Resource key to the table). The following is an example of what your Create Plugin

page should look like.

7. Enter the key values. The following table describes the required information.

Key	Value Description
URL	URL for the RLM Server web interface in the form: `<protocol>://<rlm_server_host>:<rlm_web_interface_port>`. For example: http://server:5054
Username	The username in the RLM Web interface; typically user.
Password	Password used by the user listed in the Username key. This is the password you set when you install the RLM. See Change the Default Passwords.
ISV	Independent software vender for Remote Visualization. This value must be starnet.
Product	Name of the licensed product for Remote Visualization. This value must be fastx2.

Key	Value Description
Resource	Name of the resource to report to Moab Workload Manager. This value must be remote_visualization.

8. When finished, click `Save` to save your changes and close this page; otherwise click `Cancel` to reset all the changes.

> ℹ️ The state should be "Started". If the state says "Errored", click Edit, modify the values as needed, click Update. Then from the Plugin Monitoring page, locate the RLM plugin and click the play icon.

9. Log out of your MWS instance and close the web browser.

3.15.4 Configure Moab to use Moab Web Services as a Resource Manager

In order for Moab to schedule remote visualization sessions, Moab also needs to be configured to use Moab Web Services as a resource manager.

On the Moab Server Host, do the following:

1. Add the following lines to `/opt/moab/etc/moab.cfg`:

```
RMCFG[mws]                        TYPE=MWS
RMCFG[mws]                        BASEURL=http://localhost:8080/mws
```
The `BASEURL` must match the configured URL of MWS.

2. Add the following line to `/opt/moab/etc/moab-private.cfg`:

```
CLIENTCFG[RM:mws] USERNAME=moab-admin PASSWORD=changeme!
```

> ℹ️ `USERNAME` and `PASSWORD` must match the values of `auth.defaultUser.username` and `auth.defaultUser.password`, respectively, found in the MWS configuration file. The MWS RM contacts MWS directly using the base URL, username, and password configured.

3. Restart Moab.

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

3.15.5 Install Remote Visualization

Remote Visualization needs to be installed on the gateway server and on *all* the session servers (Torque MOM Hosts).

> ⓘ You *must* complete all the tasks earlier in this topic before installing Remote Visualization.

Do the following:

1. Make sure that your DNS server is configured for reverse lookups. Without reverse DNS, Session Servers will fail to register with your Gateway Server. As a result, authentication requests to the Gateway Server will fail because the Gateway Server will not be able to connect to any Session Servers.

2. Prepare the hosts for RPM installation. If you will be installing Remote Visualization on a host that does not have another RPM installation, complete the steps to prepare the host. See 3.3 Preparing for RPM Installs - page 169 for more information.

3. On the Remote Visualization Gateway Server Host and each Session Server Host, do the following:

   a. Install FastX and all its dependencies.

   ```
   [root]# yum -y install ImageMagick-perl perl-Crypt-SSLeay perl-Net-SSLeay perl-X11-Protocol StarNetFastX2
   ```

   > ⓘ If installing on RHEL, some packages may not be found in the standard RHEL distribution repositories. You will need to install the missing dependencies from EPEL or other reputable repositories.
   >
   > ```
   > [root]# rpm -Uvh http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-release-6-8.noarch.rpm
   >
   > [root]# yum install yum-utils
   > [root]# yum-config-manager --disable epel
   > [root]# yum install --enablerepo=epel,rhel-6-server-eus-optional-rpms ImageMagick-perl perl-Crypt-SSLeay perl-Net-SSLeay perl-X11-Protocol StarNetFastX2
   > ```

   b. Create or use an unprivileged account to login into fastx with admin privileges. This is the `<fastxadminuser>`.

   > ⓘ The following example uses the ace user and password. You can use an existing user, as long as that user can ssh into this host with a username/password pair.

   ```
   [root]# useradd ace
   ```

```
[root]# passwd ace
```

Run the install.sh script on the Remote Visualization Gateway Server *and* on all of the Session Servers (Torque MOM Hosts).

Answer the questions raised by the install.sh script. For example:

```
[root]# /usr/lib/fastx2/install.sh
Do you have a license server on your network? [y/N] y
Enter the name (or IP address) of your license server: localhost
License file /usr/lib/fastx2/rlm/localhost.lic has been created.
Install/update the FastX web server? [Y/n] y
Creating a self-signed certificate... done.
A self-signed certificate has been created for this web server.
It will allow secure connections, but is vulnerable to a
man-in-the-middle attack. Because of this, connections will generate
warnings from the browser. These warnings (and the vulnerability) can be
eliminated later by installing a certificate from a certificate authority.
Setup initial admin user? [Y/n] y
The admin must be an existing Linux user, but not root.
The admin will be able to see and terminate any user's session,
add additional admins, and configure the server.
Enter admin account: ace
Starting FastX web service...
Starting fastx (via systemctl): [ OK ]
FastX Server listening on port 3000
FastX HTTPS Server listening on port 3443
                                        done.
```

4. Viewpoint supports either password-based authentication *or* key-based authentication for Remote Visualization.

   • For password-based authentication, do the following on the Remote Visualization Gateway Server *and* on each Session Server:

     a. Set the following parameters in `/etc/ssh/sshd_config`:

     ```
     PasswordAuthentication yes
     ChallengeResponseAuthentication no
     ```

     b. Restart the sshd service.

     ```
     [root]# service sshd restart
     ```

   • For key-based authentication, do the following:

     a. On the Remote Visualization Gateway Server, log in as the FastX admin user and generate a ssh key. Accept the default.

     > ⚠️ A passphrase is not supported by Viewpoint. Leave this field empty.

     ```
     [<fastxadminuser>@<hostname> ~]$ ssh-keygen -t rsa
     Generating public/private rsa key pair.
     Enter file in which to save the key (/home/<fastxadminuser>/.ssh/id_rsa):
     Created directory '/home/<fastxadminuser>/.ssh'.
     ```

```
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/<fastxadminuser>/.ssh/id_rsa.
Your public key has been saved in /home/<fastxadminuser>/.ssh/id_rsa.pub.
The key fingerprint is:
...
```

b. Copy the generated id_rsa private key to a location where Viewpoint has access.

c. Set the generated id_rsa public key as an authorized key for the Gateway Server.

```
[root]# cat ~<fastxadminuser>/.ssh/id_rsa.pub >>
~<fastxadminuser>.ssh/authorized_keys ; chown <fastxadminuser>.
~<fastxadminuser>/.ssh/ -R
```

d. Copy the id_rsa public key to all the Session Servers and set it as an authorized key.

> ⓘ For documentation clarity, these instructions use node00 through node09 as the names of the Session Servers; with node00 designated as the initial Session Server.

```
[root]# for i in {00..09} ; do scp ~<fastxadminuser>/.ssh/id_rsa.pub
node$i:<fastxadminuser home>:id_rsa.pub ; done
[root]# for i in {00..09} ; do ssh node$i "cat id_rsa.pub >> <fastxadminuser
home>/.ssh/authorized_keys ; rm -f id_rsa.pub ; chownfastxadminuser>.
<fastxadminuser home>/.ssh/ -R" ; done
```
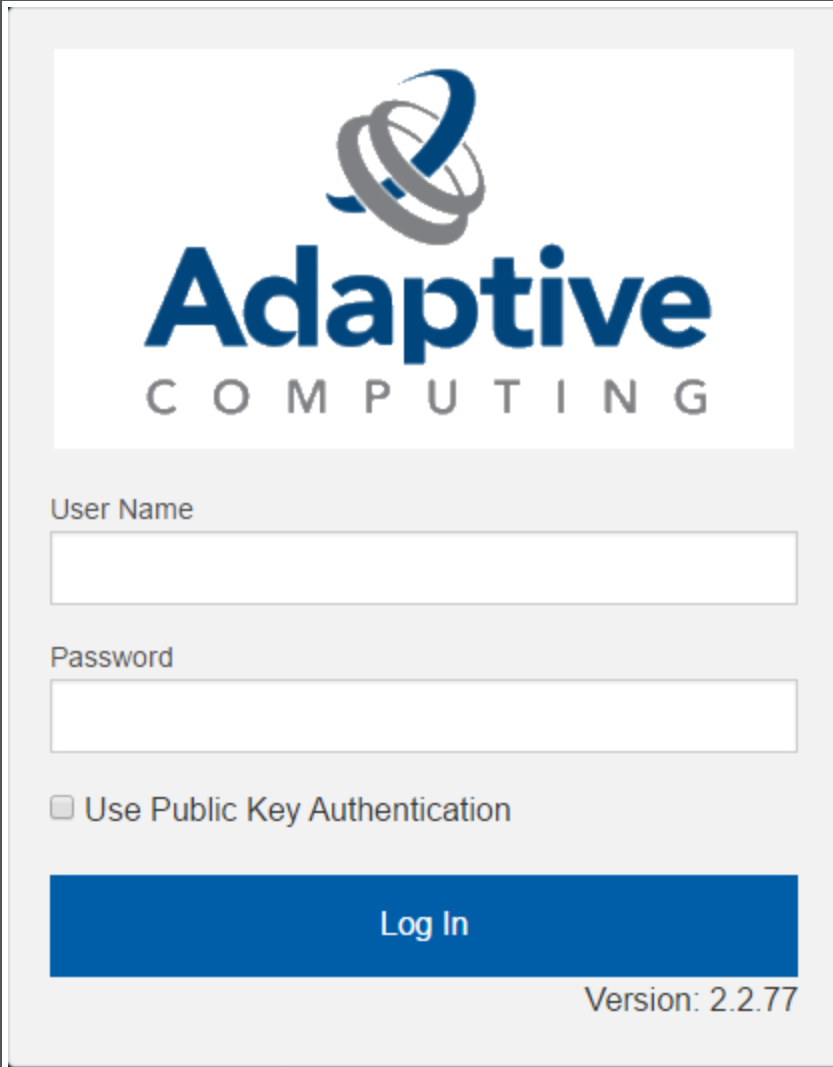
3.15.6 Configure the Gateway Server

Do the following:

1. Using a web browser, navigate to your *secure* Remote Visualization Gateway Server instance. (**https**://<gateway_host>:3443; where <gateway_host> is the IP address or name of the Gateway Server Host).

   The Log In page displays. For example:

2. Log in as the FastX admin user. Do *one* of the following:

- If your authentication method is password-based, do the following:
  a. Enter the user name (default is "ace").
  b. Enter the password (default is "ace").
  c. Make sure the "Use Public Key Authentication" checkbox is cleared.
  d. Click `Log In`.
- If your authentication method is key-based, do the following:
  a. Enter the user name (default is "ace").
  b. Select the "Use Public Key Authentication" checkbox.

    c. A prompt will display asking for you to load your private key file.

        a. Click `Upload Private Key File` and navigate to your stored key file.

          When your key file has uploaded it will be displayed in the prompt. For example:

        b. Click `Close`. The prompt closes.

    d. Click `Log In`.

3. Click the icon for Admin\System Configuration. The icon is circled in the example to assist in finding its location.

4. Select the Network tab. If it is not already selected, select the Configuration sub-tab to display the FastX Network Configuration page.

5. Do the following:

    a. In the Secret Key field is a FastX-generated key. Record this secret key (e.g. copy to your clipboard) because you will need it when configuring the Session Servers later in this topic.

This key is different from the key file used to log in as the administrative user. You can also change the generated Secret Key if needed.

b. Enable the connection to accept data from cluster member.

c. In the box to specify the log in method, select "Sessions - log in to the system running the fewest sessions".

d. Disable the Gateway Server from sending data to cluster members.

The following image is an example of the completed FastX Network Configuration page for the Gateway Server.
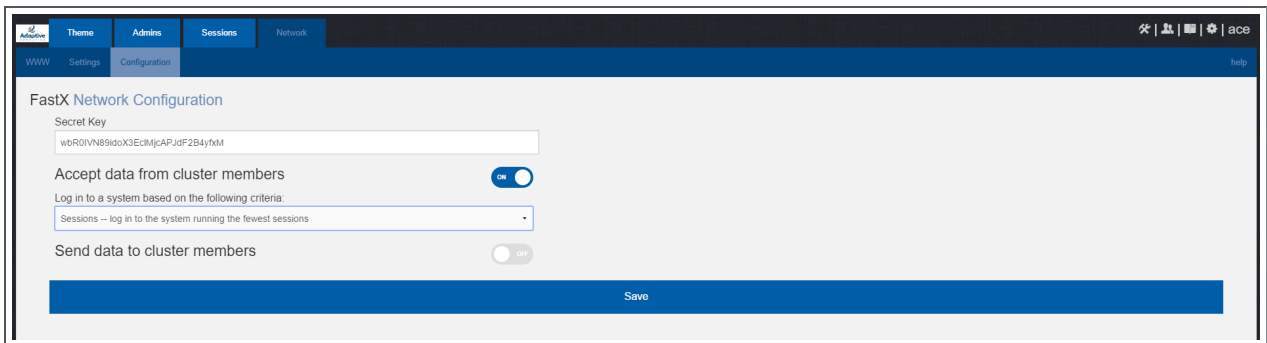
6. Click Save to submit your changes.
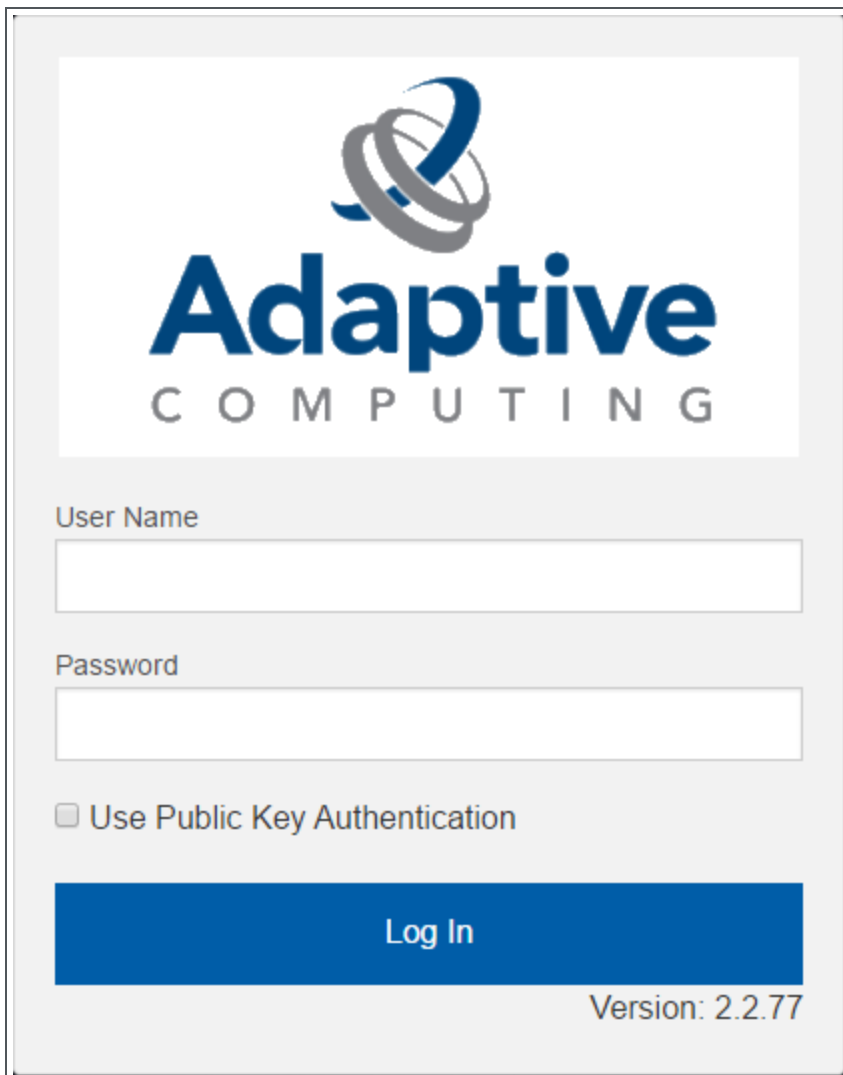
3.15.7 Configure a Session Server

This section provides instructions on how to configure *one* Session Server (referred to as the initial Session Server). The configuration will then be copied to the additional Session Servers in your environment in a later procedure.

Do the following:

1. Using a web browser, navigate to your *secure* Remote Visualization Session Server instance. (**https**://<session-host>:3443; where <session_host> is the IP address or name of the *initial* Remote Visualization Session Server Host).

The Log In page displays. For example:

2. Log in as the FastX admin user. Do *one* of the following:

- If your authentication method is password-based, do the following:

    a. Enter the user name (default is "ace").

    b. Enter the password (default is "ace").

    c. Make sure the "Use Public Key Authentication" checkbox is cleared.

    d. Click `Log In`.

- If your authentication method is key-based, do the following:

    a. Enter the user name (default is "ace").

    b. Select the "Use Public Key Authentication" checkbox. Upload the public key used when you configured the Gateway Server earlier in this topic.

    c. Click `Log In`.

    > 🛈 When you first log in, you will get a message that you have no session running. That is expected.

3. Select the icon for Admin\System Configuration. The icon is circled in the example to assist in finding its location.

4. Select the Network tab. If it is not already selected, select the Configuration sub-tab to display the FastX Network Configuration page.

5. Do the following:

    a. In the Secret Key field, enter the name of the secret key provided when configuring the Gateway Server earlier in this topic.

    > 🛈 You will not be able to login to the portal on the Gateway Server until you have completed the configuration of at least one Session Server. If you did not save it earlier, the secret key can be found in the `/usr/lib/fastx2/config/network.json` on the Gateway Server.

    b. Disable the connection to accept data from cluster members.

    c. Enable the Gateway Server to send data to cluster members.

    d. In the box to specify whether to SSL certificates, select "I am using a self-signed certificate".

e. In the Cluster member URLs box, do the following:

   i. Click the + icon.

   ii. In the box that displays, enter the IP address or name and the port number of the Gateway Server you just configured (for example: "https://mgmtnode:3443").

The following image is an example of the completed FastX Network Configuration page.
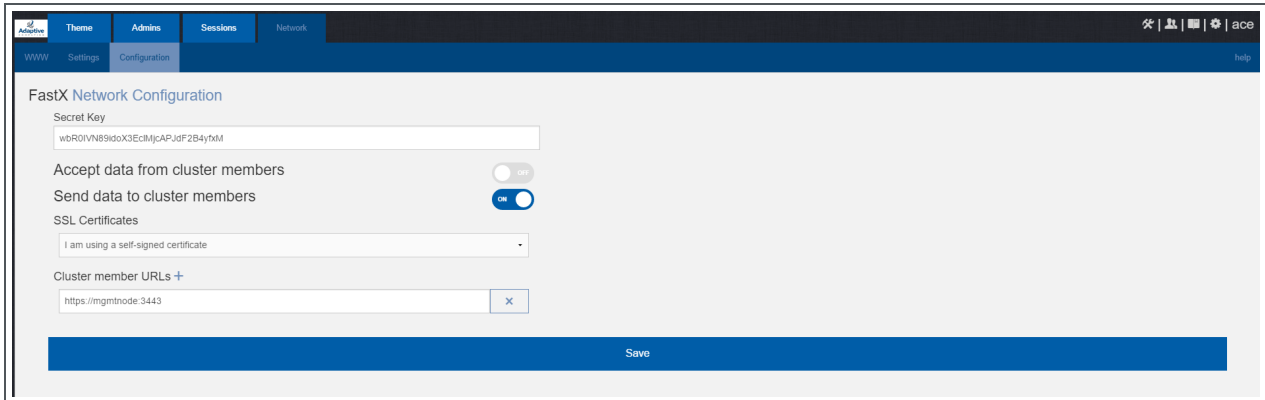
6. Click `Save` to submit your changes.

3.15.8 Copy the Session Server Configuration to the Remaining Session Servers

After you configured the initial Session Server, the settings are saved in the network.json file.

> **ℹ** For documentation clarity, these instructions use node00 through node09 as the names of the Session Servers; with node00 designated as the initial Session Server.

On the *initial* Session Server Host, copy the network.json file to the remaining Session Server Hosts in your environment, and restart the FastX service.

```
[root]# for i in {01..09} ; do scp /usr/lib/fastx2/config/network.json
root@node$i:/usr/lib/fastx2/config/network.json ; done
[root]# for i in {01..09} ; do ssh node$i "chown fastx. /usr/lib/fastx2/config/. -R" ;
done
[root]# for i in {01..09} ; do ssh node$i "service fastx restart" ; done
```

3.15.9 (Optional) Install Graphical Packages on Each Session Server

A few graphical packages are available to let you easily submit remote visualization jobs from Viewpoint (install a desktop environment).

One each Remote Visualization Session Server Host, do the following:

```
[root]# yum -y groupinstall "Desktop" "Desktop Platform" "X Window System" "Fonts"
[root]# yum -y install xterm
[root]# service messagebus start
```

3.15.10 Configure Moab for Remote Visualization

On the Moab Server Host, verify the `/opt/moab/etc/moab.cfg` file contains the following uncommented parameter:

```
JOBCFG[remote_visualization] FLAGS=usemoabjobid SELECT=TRUE
```

> ℹ This parameter configuration specifies that Moab will reference remote visualization jobs by their internal Moab job id. However, the job's output and error files will still be generated by your resource manager (for exampe, Torque). This means that, even though your job will get assigned a Moab job id, your job's output and error file names will reference the resource manager's job id (for example, job.oX).
>
> If you need the job's output files to match the same job id as your Moab job, append the following parameters to your moab.cfg:
>
> ```
> RMCFG[pbs] SYNCJOBID=TRUE FLAGS=ProxyJobSubmission
> ```
>
> ```
> RMCFG[internal] JOBIDFORMAT=integer
> ```
>
> Be advised that these appended parameters are *not* recommended for all systems; especially if your configuration includes customizations. If your system is not working as expected, contact Adaptive Computing support for assistance.

If you have made changes to the moab.cfg file, make sure you restart Moab.

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

3.15.11 Configure Viewpoint for Remote Visualization

Do the following:

1. Using a web browser, navigate to your Viewpoint instance (`http://<server>:8081`) and then log in as the MWS administrative user (moab-admin, by default).

2. Click `Configuration` from the menu and then click `Remote Visualization Services` from the left pane.

   The following is an example of the Remote Visualization Configuration page.

3. Enter the hostname (or IP address) and port number for the FastX gateway server in the Gateway Server field (do not use localhost if you intend to submit remote viz jobs from other hosts). For example, https://<server>:3443.

4. If your Remote Visualization configuration was set up using self-signed certificates, confirm the Trust Self Signed check box is selected.

5. Enter the FastX admin user you specified when you installed the Remote Visualization Server in the Username field. For example, ace.

6. If your configuration will authenticate using the *password-based* method, do the following:

   a. Select Password Based Authentication from the Authentication Method box.

   b. Enter the FastX admin user's password in the Password field.

   > ⚠ The /etc/ssh/sshd_config file on each Session server must be configured to enable password authentication. See 3.15.5 Install Remote Visualization - page 262 earlier in this topic for more information.

7. If your configuration will authenticate using the *key-based* method, do the following:

   a. Select Key Based Authentication from the Authentication Method box.

   b. Click UPLOAD KEY and navigate to the copy of the generated .ssh/id_rsa file.

8. Click TEST to confirm your settings are correct.

9. Click SAVE to submit your settings.

3.15.12 Grant Users Remote Visualization Permissions in Viewpoint

Viewpoint comes packed with base (default) roles for Remote Visualization jobs. Any user who will be working with Remote Visualization, must have the appropriate role added to the Viewpoint user principal.

These are the Viewpoint Roles for Remote Visualization:

- RemoteVizAdmin – Administrative user, with permission to create remote visualization application templates and manage other user's remote visualization jobs.

- RemoteVizUser – Basic user, with permission to create and manage their own remote visualization jobs.

See Creating or Editing Principals in the *Moab Viewpoint Reference Guide* for instructions on setting up principals.

3.16 Installing Nitro

This topic contains instructions on how to install Nitro.

Nitro:

- needs to be available to all of the nodes that will be used as part of the Nitro job.
- can be installed either to each node individually *or* to a shared file system that each node can access.

> ⚠ Some Nitro functionality, such as using the nitrosub command, is not available unless you are using a shared file system.

- can be installed to integrate with a scheduler, such as Moab, or without (Nitro standalone). The instructions are the same.

A shared file system allows Nitro task files to be copied to the compute nodes, which is where they will be needed when the job runs. Without a shared file system the Nitro task file will have to be copied to the location where Nitro expects it to be on the compute node manually. This is challenging if it is not known in advance on which compute node the job will run. In addition, a shared file system allows Nitro logs to be seen and opened on the head node, which allows Viewpoint to access them.

> In this topic:
>

3.16.1 Obtain a Nitro License

The Nitro license file is installed on an RLM Server. See 1.4 RLM Licensing Model - page 32 for more information about RLM licensing.

> ⚠ These instructions assume you already have access to an RLM Server. See 3.14 Installing RLM Server - page 255 for instructions on how to set up a new RLM Server.

Do the following:

1. On the RLM server, obtain the host ID and hostname.

    - host ID

    ```
    [root]# /opt/rlm/rlmhostid
    ```

    You should see output similar to the following.

    ```
    rlmhostid v12.1
    Copyright (C) 2006-2016, Reprise Software, Inc. All rights reserved.

    Hostid of this machine: 00259096f004
    ```

    - hostname

    ```
    [root]# /opt/rlm/rlmhostid host
    ```

    You should see output similar to the following.

    ```
    rlmhostid v12.1
    Copyright (C) 2006-2016, Reprise Software, Inc. All rights reserved.

    Hostid of this machine: host=<your-host-name>
    ```

2. Email `licenses@adaptivecomputing.com` for a license and include the host ID and hostname you just obtained.

3. Adaptive Computing will generate the license and send you the Nitro license file (typically, `nitro.lic`) file in a return email.

4. On the RLM server, do the following:

    a. Download and install the license file.

    ```
    [root]# cd /opt/rlm
    [root]# chown rlm:rlm nitro.lic
    ```

    b. If the RLM Server in your configuration uses a firewall, edit the license file to reference the ISV `adaptiveco` port for the Adaptive license-enabled products. This is the same port number you opened during the RLM Server installation. See the instructions to open necessary ports in the 2.7 Installing RLM Server - page 86 (manual installation method) or 3.14 Installing RLM Server - page 255 (RPM installation method) for more information.

    ```
    [root]# vi /opt/rlm/nitro.lic

    ISV adaptiveco port=5135
    ```

    The license file already references the RLM Server port (5053 by default).

    > ⓘ If the RLM Server in your configuration uses different ports, you will need to modify the license file to reflect the actual ports. See the instructions to open necessary ports in the 2.7 Installing RLM Server - page 86 (manual installation method) or 3.14 Installing RLM Server - page 255 (RPM installation method) for more information.

   c. If you did *not* install an RLM Server using the file available from Adaptive Computing (for example, because your system configuration already uses one), do the following:

      i. Download the `adaptiveco.set` file from the Adaptive Computing Nitro Download Center (https://www.adaptivecomputing.com/support/download-center/nitro/).

      ii. Copy the `adaptiveco.set` file into the same directory where the Nitro license resides (`/opt/rlm`).

   d. Perform a reread to update the RLM Server with your license.

```
[root]# /opt/rlm/rlmreread
```

3.16.2 Open Necessary Ports

Nitro uses several ports for communication between the workers and the coordinator.

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

> ℹ The listed ports are for configurations that have only one coordinator. If multiple coordinators are run on a single compute host, then sets of ports (range of 4) must be opened for the number of expected simultaneous coordinators.

Location	Ports	Functions	When Needed
Compute Hosts (Nitro Coordinator)	47000	Coordinator/Worker communication	Always
Compute Hosts (Nitro Coordinator)	47001	Coordinator PUB/SUB channel - publishes status information	Always
Compute Hosts (Nitro Coordinator)	47002	Reserved for future functionality	
Compute Hosts (Nitro Coordinator)	47003	API communication channel	Always

See 2.13 Opening Ports in a Firewall - page 106 for general instructions and an example of how to open ports in the firewall.

3.16.3 Install Nitro

> ℹ️ You `must` complete the tasks to obtain a Nitro license before installing Nitro. See 3.16.1 Obtain a Nitro License - page 274.
>
> If your configuration uses firewalls, you *must also* open the necessary ports before installing Nitro. See 3.16.2 Open Necessary Ports - page 276.

On the host on where Nitro will reside, do the following:

1. If you are installing Nitro on its own host *or* on a host that does not have another RPM installation, complete the steps to prepare the host. See 3.3 Preparing for RPM Installs - page 169 for more information.

2. Install the RPM.

```
[root]# yum install nitro
```

3. Copy the license file you generated earlier in this topic to each compute node (coordinator). On each compute node, *or* on the shared file system, do the following:

```
[root]# cp nitro.lic /opt/nitro/bin/
```

4. Copy the provided scripts and the *nitrosub* command from the `/opt/nitro/scripts` directory.

> ℹ️ This is a "copy" file operation and not a "move" operation. This allows you to customize your version and always have the factory version available for consultation and/or comparison.

   a. Copy the *launch_nitro.sh* and *launch_worker.sh* scripts for your resource manager to the `bin` directory. Each resource manager has a subdirectory with the `scripts` directory that contains the scripts. This example uses Torque as the resource manager.

```
[root]# cp /opt/nitro/scripts/torque/launch_nitro.sh /opt/nitro/bin/
[root]# cp /opt/nitro/scripts/torque/launch_worker.sh /opt/nitro/bin/
```

   b. Copy the *nitrosub* command to the `bin` directory.

```
[root]# cp /opt/nitro/scripts/nitrosub /opt/nitro/bin/
```

   c. Copy the `nitro_job.sh` and the `worker_job.sh` scripts to the `etc` directory.

```
[root]# cp /opt/nitro/scripts/nitro_job.sh /opt/nitro/etc/
[root]# cp /opt/nitro/scripts/worker_job.sh /opt/nitro/etc/
```

5. Now that you have copied the scripts and the *nitrosub* command, edit the copies for your site's administrative policies.

- `bin/nitrosub` command (applicable only if using a shared file system). At a *minimum*, do the following:

   a. Uncomment the `_resource_manager` line for your resource manager.

   b. Uncomment the `resource_type` line for your licensing model's allocation (nodes or cores).

   c. If your system will be using dynamic jobs, set the `_dynamic_size` value to the number of resources to allocate to a dynamic job.

   See 1.1 nitrosub Command in the *Nitro Administrator Guide* for more information.

- `bin/launch_nitro.sh` and `bin/launch.worker.sh` scripts. See 1.1 Launch Scripts in the *Nitro Administrator Guide* for more information.

6. If your system configuration allows multiple coordinators on the same node, additional configuration may be needed. See 2.19 Running Multiple Coordinators on the Same Node - page 120 for more information.

7. If you are *not* using a shared file system, copy the Nitro installation directory to *all* hosts.

```
[root]# scp -r /opt/nitro root@host002:/opt
```

> 🛈 If you are not using a shared file system, you may not be able to use the `nitrosub` client command.

3.16.4 Verify Network Communication

Verify that the nodes that will be running Nitro are able to communicate with the Nitro ports *and* that the nodes are able to communicate with one another.

Related Topics

- 1.1 Nitro Integration

3.17 Installing Nitro Web Services

This topic contains instructions on how to install Nitro Web Services.

Perform the following steps:

1. Open Necessary Ports

2. Install and Configure MongoDB

3. Install and Configure Nitro Web Services

4. Configure Viewpoint for Nitro Web Services

5. Publish Nitro Events to Nitro Web Services

3.17.1 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.

Location	Ports	Functions	When Needed
Nitro Web Services Host	9443	Tornado Web Port	Always
Nitro Web Services Host	47100	ZMQ Port	Always
Nitro Web Services Database Host	27017	Nitro Web Services MongoDB Server Port	If you will be installing the Nitro Web Services Database on a different host from Nitro Web Services

See 2.13 Opening Ports in a Firewall - page 106 for general instructions and an example of how to open ports in the firewall.

3.17.2 Install and Configure MongoDB

ⓘ If you have already installed Mongo DB (and created the admin_user), skip to step 3b.

> ⓘ Setting per-user limits on various resources can prevent MongoDB from closing connections if the number of connections grows to high. See Review and Set Resource Limits (https://-docs.mongodb.com/manual/reference/ulimit/#review-and-set-resource-limits) for more information about using the *ulimit* command to review and set resource limits.

On the Nitro Web Services MongoDB Database Host, do the following:

1. Install MongoDB.

```
[root]# yum install -y mongodb-org
```

2. Enable and start MongoDB.

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

3. Add the required MongoDB users.

> ⓘ The passwords used below (secret1 and secret5) are examples. Choose your own passwords for these users.

   a. Add admin_user.

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

> ⓘ 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.

   b. Add nitro_user.

```
[root]# mongo -u admin_user -p secret1 admin
> use nitro-db
> db.createUser({"user": "nitro_user", "pwd": "secret5", "roles": ["dbOwner"]})

> exit
```

4. Set MongoDB Configuration Options.

   - The configuration file for MongoDB is /etc/mongod.conf. See https://docs.mongodb.com/manual/reference/configuration-options for information.

- Adaptive Computing recommends that you set `security.authorization` to `enabled`. See https://docs.mongodb.com/manual/reference/configuration-options/#security-options for more information.

> ℹ By default, `/etc/mongod.conf` sets `net.bindIp` to 127.0.0.1. You will need to change this setting if the MongoDB server needs to be accessible from other hosts or from other interfaces besides loopback. See https://docs.mongodb.com/manual/reference/configuration-options/#net-options for more information.

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

5. Restart MongoDB.

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

3.17.3 Install and Configure Nitro Web Services

> ℹ You `must` complete the tasks earlier in this topic before installing Nitro Web Services.

On the host where Nitro Web Services will reside, do the following:

1. If you are installing Nitro Web Services on its own host *or* on a host that does not have another RPM installation, complete the steps to prepare the host. See 3.3 Preparing for RPM Installs - page 169 for more information.

2. Install the Nitro Web Services RPM.

```
[root]# yum install -y nitro-web-services
```

3. Understand and edit the the Nitro Web Services web application configuration file (`/opt/nitro-web-services/etc/nitro.cfg`). This includes clarifying what the

configuration file is for and what to expect the first time the NWS service is started vs. each subsequent start.

> ℹ The `nitro_user` with `dbOwner` permissions was set up earlier in the procedure (see 3.17.2 Install and Configure MongoDB - page 279).
>
> When you first start `nitro-web-services`, the `nitro-db` Mongo database (including its collections and indexes) is created. The `nitro-db` 'user' collection is also populated with the default Nitro Web Services API users/passwords. Several of the options defined in the configuration files influence this process.

MongoDB user, table, and index creation is performed at initial startup. Many of the options defined in the Nitro Web Service configuration files influence Mongo user/password and index creation.

> ⚠ Usernames and passwords are created *only* if they do not yet exist. Changing a password in the configuration file after initial startup will not update the password.

The `/opt/nitro-web-services/etc/nitro.cfg` configuration file includes default password values. If you want to use passwords other than the default value, change the lines in the file as shown below.

- Before initial startup, set the `db_password` to be the `nitro_user` password. It is also recommended that you change all other default passwords before starting Nitro Web Services. If you do not change the passwords at this point, it will be more difficult to change them later.

```
db_password = <password goes here>
admin_password = <admin_password goes here>
ws_readonly_password = <ws_readonly_password goes here>
ws_writeonly_password = <ws_writeonly_password goes here>
```

- By default, NWS uses an auto-generated self-signed SSL certificate to encrypt the link between the web server and the browser clients. The auto-generated self-signed SSL certificate is created at service start up; not during the installation process.

  However, you can use your own `certfile`, `keyfile`, and `ca_certs` files if you wish.

  > ℹ If you choose to use your own `ssl_certfile` and `ssl_keyfile`, `ssl_create_self_signed_cert=true` is ignored.

- By default, NWS does *not* encrypt network traffic with MongoDB. You should set the `db_ssl_*` properties if you choose to enable TLS/SSL when installing MongoDB earlier in this topic.

4. Understand and edit the Nitro ZMQ Job Status Adapter configuration file (`/opt/nitro-web-services/etc/zmq_job_status_adapter.cfg`). This includes clarifying what the configuration file is for and what to expect the first time the NWS service is started vs. each

subsequent start.

- The Nitro ZMQ Job Status Adapter listens to job status updates on the ZMQ bus and publishes them to MongoDB using the Nitro Web Services REST API.

- The username and password must be set to a Nitro Web Services API user with write permissions. At minimum, set the password for `nitro-writeonly-user` to the password defined in `/opt/nitro-web-services/etc/nitro.cfg` and make sure the SSL options are set correctly based on SSL settings in `/opt/nitro-web-services/etc/nitro.cfg`.

```
password = <ws_writeonly_password goes here>
```

5. If you did not need to install the Nitro Web Services MongoDB database earlier in this topic, verify that the mongodb_hostlist in `/opt/nitro-web-services/etc/nitro.cfg` is set correctly (localhost:27017 is the default).

6. Start the services and configure Nitro Web Services to start automatically at system boot.

```
[root]# chkconfig --add nitro-web-services
[root]# chkconfig --add nitro-zmq-job-status-adapter
[root]# service nitro-web-services start
[root]# service nitro-zmq-job-status-adapter start
```

3.17.4 Configure Viewpoint for Nitro Web Services

Do the following:

1. Using a web browser, navigate to your Viewpoint instance (`http://<server>:8081`) and then log in as the MWS administrative user (moab-admin, by default).

2. Click `Configuration` from the menu and then click `Nitro Services` from the left pane. The following is an example of the Nitro Services Configuration page.

3. Enter the configuration information. The following table describes the required information.

Field	Description
Nitro WS URL	Hostname (or IP address) and port number for the host on which you installed Nitro Web Services. For example, https://`<hostname>`:9443
Username	Name of the user. This typically nitro-readonly-user.
Password	The user's password.
Trust Self Signed	Indicates whether Nitro Web Services was set up using self-signed certificates.

4. Click `TEST` to confirm the settings are correct. This confirms whether Nitro Web Services is up and receiving connections.

5. Click `SAVE` to submit your settings.

6. (Recommended) Use curl to test Nitro Web Services connectivity.

```
[root]# curl --insecure --data '{"username": "nitro-admin", "password":
"ChangeMe2!"}' \
 https://<hostname>:9443/auth
```

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

```
{
  "status": 200,
  "data": {
```

```
      "nitro-key": "3e0fb95e9a0e44ae91daef4deb500dcc67a3714880e851d781512a49",
      "user": {
        "username": "nitro-admin",
        "last_updated": "2016-02-26 23:34:55.604000",
        "name": "Nitro Admin",
        "created": "2016-02-26 23:34:55.604000",
        "auth": {
          "job": [
            "read",
            "write",
            "delete"
          ],
          "user": [
            "read",
            "write",
            "delete"
          ]
        }
      }
    }
  }
}
```

3.17.5 Grant Users Nitro Permissions in Viewpoint

Viewpoint comes packed with base (default) roles for Nitro jobs. Any user who will be working with Nitro Web Services, must have the appropriate role added to the Viewpoint user principal.

These are the Viewpoint roles for Nitro:

- NitroAdmin – Administrative user, with permission to create Nitro application templates and manage other user's Nitro jobs.

- NitroUser – Basic user, with permission to create and manage their own Nitro jobs.

See "Creating or Editing Principals" in the *Moab Viewpoint Reference Guide* for instructions on setting up principals.

3.17.6 Publish Nitro Events to Nitro Web Services

You need to configure the Nitro coordinators to send job status updates to the Nitro Web Services's ZMQ Job Status Adapter. The ZMQ Job Status Adapter is responsible for reading job status updates off of the ZMQ bus and persisting them to Mongo. Nitro Web Services can then be used to access Nitro job status.

Each Nitro job has a Nitro Coordinator. Nitro Coordinators can be configured to publish job status updates to ZMQ by setting the "nws-connector-address" configuration option in Nitro's nitro.cfg file. Each compute node allocated/scheduled to a Nitro Job can play the role of a Nitro coordinator. Therefore, you must update the "nws-connector-address" in each compute node's nitro.cfg file.

> ℹ️ Configuring nws-connector-address is simplified if each node is sharing nitro's configuration over a shared filesystem. If you are not using a shared filesystem, update the nitro configuration on each compute node.

Do the following:

1. If you have not already done so, on the Nitro Web Services Host, locate the msg_port number in the `/opt/nitro-web-services/etc/zmq_job_status_adapter.cfg` file. This is the port number you need to specify for the nws-connector-address.

2. On *each* Nitro compute note (Torque MOM Host), specify the nws-connector-address in the `/opt/nitro/etc/nitro.cfg` file.

```
...
# Nitro Web Services connection allows Nitro to communicate job status information
# to the Nitro Web Services connector. This option indicates name and port of the
# remote server in the form: <host>:<port>
nws-connector-address <nitro-web-services-hostname>:47100
...
```

> ℹ️ You must uncomment the nws-connector-address line, even if it is already correct.

Related Topics

- 1.1 Nitro Integration

3.18 Disabling the Adaptive Repository after Installs

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

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

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

3.19 Additional Configuration

3.20 Opening Ports in a Firewall

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

This topic provides an example and general instructions for how to open ports in your firewall. See 5.3 Port Reference - page 411 for the actual port numbers for the various products.

Red Hat 6-based systems use iptables as the default firewall software. For the ip6tables service, replace all occurrences of iptables with ip6tables in the example. If you use different firewall software, refer to your firewall documentation for opening ports in your firewall.

The following is an example of adding port 1234 when using 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 1234 -j ACCEPT

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

3.21 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).

3.22 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.

3.22.1 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.

ℹ️ 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
```

> ℹ️ 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 3.22.2 Adding an Organizational Unit (OU) - page 295.

3.22.2 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
```

3.22.3 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
    ```

3.22.4 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
    ```

3.22.5 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.

> ℹ️ Before you add a user to an LDAP group, the user must first be added to LDAP. For more information, see 3.22.3 Adding a User - page 296.

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
```

3.23 Using Multiple RLM Servers

As the RLM Server can run multiple licenses, it is recommended that you install *one* RLM Server for your configuration.

However, if your configuration requires more than one RLM Server, you will *need* to configure the Adaptive Computing products to connect to a specific RLM Server. If not configured to connect to a specific RLM Server, the Adaptive Computing product will scan the network and connect to the first RLM Server it finds listening to request the license. If the first RLM Server does *not* have the product's license, the RLM connection will fail.

If you are using multiple RLM Servers, do the following to configure the an Adaptive Computing product to connect to a specific RLM Server:

1. Modify the RLM Server not to accept the network search connections.

   - Edit the init script in `/opt/rlm/` to add -noudp.

   ```
   start() {
   su -l $rlmuser -s /bin/bash -c "$rlmdir/rlm -l -dlog $debuglog -noudp &"
   }
   ```

2. Enable the Adaptive Computing product to connect to a specific RLM.

   On the host where the Adapative Computing product resides, do the following:

   a. Create a new text file and name it with the .lic extension (typically, remote.lic) and save it in the same location as the other Adaptive Computing licenses. Be careful not to override an existing license.

   b. Edit the new remote.lic file to point to the specific RLM Server hostname and port. Port 5053 is the default. If you use a different port number for the RLM Server, specify that port number in the remote.lic file.

   ```
   HOST <hostname> ANY 5053
   ```

Repeat as needed for each Adaptive Computing product that you want to connect to a specific RLM Server.

3.24 Running Multiple Coordinators on the Same Node

Nitro provides the ability to run multiple coordinators on the same node.

> **ℹ** Running multiple coordinators on the same node is not available if your system configuration uses a policy to limit nodes to a single job (i.e., NODEACCESSPOLICY=SINGLEJOB on Moab).

If your system is configured to allow multiple coordinators on the node:

- It is recommended that you instruct your uses to submit Nitro jobs using the nitrosub command. See nitrosub Command in the *Nitro User Guide* for more information.

- If you prefer that your users do *not* use the nitrosub command, and instead you prefer that they submit the Nitro jobs directly to your scheduler/resource manager, then you will need to add the --port-file option to the bin/launch_nitro.sh *and* bin/launch_worker.sh scripts to ensure that all coordinators will be able to run.

```
NITRO_OPTIONS="--port-file --job-id ${NITROJOBID} ${NITRO_OPTIONS}"
```
Add the --port-file option `before` *the --job-id information.*

3.25 Trusting Servers in Java

3.25.1 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/tomcat/tomcat.conf
```

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

3.25.2 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.

3.25.3 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.

> ℹ️ 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.

3.26 Updating the Reporting Application Configuration

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

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

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

2. Kill the ReportingDataProcessing Spark application.

    a. Open the Spark Master UI by opening `<reporting_master_host>:8082` in a browser.

    b. Locate the Reporting Data Processing row in the `Running Applications` section.

    c. Click the (kill) link to the left of the name `ReportingDataProcessing`.

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

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

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

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

Note that Tomcat may take several minutes to restart.

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

3.27 RPM Upgrades

This section provides instructions and other information when upgrading your Moab components for Red Hat 6-based systems using the RPM upgrade method.

3.28 Preparing for RPM Upgrades

Depending on the RPM upgrade method (typical or offline) you choose, you will need to prepare your system for the RPM upgrades.

- If you are using the *typical* RPM upgrade method, continue with the topic: .

- If you are using the *offline* RPM upgrade method, continue with the topics: *and* .

Related Topics

-

3.29 Preparing the Host – Typical Method

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

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

> ⓘ Whether you are upgrading RPMs on one host or on several hosts, each host (physical machine) on which a server is installed (Torque Server Host, Moab Server Host, etc) *must* have the Adaptive Computing Package Repository enabled. If Remote Visualization is part of your configuration, the Adaptive Computing Package Repository must also be enabled on the Torque MOM Hosts (compute nodes); otherwise is not necessary to enable the Adaptive Computing repository on the Torque MOM Hosts or client hosts.

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

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

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

2. Download the latest Moab HPC Suite RPM bundle from the Adaptive Computing Moab HPC Suite Download Center (https://www.adaptivecomputing.com/support/download-center/moab-hpc-suite-download/).

3. Untar the RPM bundle.

    ```
    [root]# tar zxf moab-hpc-suite-9.1.3-<OS>.tar.gz
    ```

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

4. Change directories into the untarred directory.

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

5. 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" to view usage information and identify the default directory location. If the [<repository-directory>] already exists, RPMs will be added to the existing directory. No files are overwritten in [<repository-directory>].

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

The repository file is created in /etc/yum.repos.d/.

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

6. Test the repository.

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

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

```
...
moab-accounting-manager.x86_64 : Moab Accounting Manager for Moab HPC Suite
moab-hpc-enterprise-suite.noarch : Moab HPC Suite virtual package
moab-insight.x86_64 : Moab Insight
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
...
```

7. Continue with instructions to upgrade the Moab HPC Suite components. See 3.1.3 Installation and Upgrade Process - page 167 for more information.

3.30 Creating the moab-offline Tarball

> ⚠ The Moab Offline Tarball is *only* created if you are using the RPM Installation – Offline Method. See 3.1.1 RPM Installation and Upgrade Methods - page 166 for more information.

This topic contains instructions on how to create a moab-offline tarball on a web-enabled host outside of your Moab HPC Suite environment. This is the tarball that is then copied (using either by scp, DVD, USB or similar) to each host within your Moab HPC Suite environment.

> ⚠ The internet-enabled host *must* have the *exact* same OS as the hosts within your Moab HPC Suite environment. As the Moab HPC Suite can have several hosts, and each host may not use the same OS, you may need to repeat this procedure for each OS used.

These instructions assume the user is non-root, but has sudo rights.

On a web-enabled host, do the following:

1. If the host uses a proxy to connect to the Internet, do the following:

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

2. Download the Moab HPC Suite RPM bundle from the Adaptive Computing Moab HPC Suite Download Center (https://www.adaptivecomputing.com/support/download-center/moab-hpc-suite-download/).

3. Untar the RPM bundle.

   ```
   [root]# tar zxf moab-hpc-suite-9.1.3-<OS>.tar.gz
   ```

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

4. Change directories into the untarred directory.

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

5. Install the suite repositories.

   ```
   sudo ./install-rpm-repos.sh -y
   ```

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

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

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

6. Confirm you own /opt.

```
sudo chown <user>:<user> /opt
```

7. Create the moab-offline directory in which to store the RPMs.

```
mkdir /opt/moab-offline
```

8. Download the Moab HPC Suite RPMs into the moab-offline directory.

   Do the following:

   a. Symlink all the Moab HPC Suite RPMs to your moab-offline directory. This enables the repotrack utility to copy them.

   ```
   ln -s /opt/adaptive-rpm-repository/rpm/*.rpm /opt/moab-offline/
   ```

   b. Use repotrack to download all dependency RPMs.

   ```
   repotrack -a x86_64 -p /opt/moab-offline moab-hpc-suite
   ```

9. Download the Java RPM into the moab-offline directory.

> **ℹ** The Java version may vary depending on the Moab HPC Suite components in your configuration. See 1.3 Component Requirements - page 18 for more information.

```
cd /opt/moab-offline
wget <java_url>
```

10. Create a repository file for the moab-offline directory.

    The `createrepo` package and its dependencies should have been installed when you ran `./install-rpm-repos.sh -y`.

```
echo "[moab-offline]
name=moab-offline
baseurl=file:///opt/moab-offline
failovermethod=priority
enabled=1
gpgcheck=0" > moab-offline.repo
```

11. Create the moab-offline tarball. The "h" option ensures the symlinked targets will be copied, instead of just the links.

```
tar hczvf moab-offline.tgz moab-offline
```

This tarball can now be copied (using scp, DVD, USB drive, or similar) to *each* host within your Moab HPC Suite environment.

3.31 Preparing the Host – Offline Method

The offline method is available for configurations where the hosts in your environment do not have internet access in order to download the Moab HPC Suite RPM dependencies.

This topic describes how to deploy the moab-offline tarball so that you can install various Moab HPC Suite components and their dependencies on all the hosts in your environment.

On *each* host (physical machine), do the following:

1. If you have not already done so, copy the moab-offline tarball to the host. For example, copy it from a CD, USB drive, or Shared network drive. See 3.5 Creating the moab-offline Tarball - page 173 for instructions on how to create the tarball.

2. Place the moab-offline tarball in the /opt directory and enter that directory.

    ```
    mv moab-offline.tgz /opt

    cd /opt
    ```

3. Untar the moab-offline directory.

    ```
    tar xvzf moab-offline.tgz
    ```

4. Copy the moab-offline.repo into place.

    - Copy to yum.repos.d.

      ```
      cp moab-offline/moab-offline.repo /etc/yum.repos.d/
      ```

    - Update the cache.

      ```
      yum clean all
      ```

5. Continue with instructions to install or upgrade the Moab components. See 3.1.3 Installation and Upgrade Process - page 167 for more information.

3.32 Upgrading to MongoDB 3.2.x (RPM)

Moab HPC Suite 9.1.0 and after requires MongoDB 3.2.x. On each host on which MongoDB is installed, check the installed version of MongoDB to see if it needs to be upgraded. You can check the version of MongoDB installed by running the following command:

```
[root]# rpm -qa | grep mongo
```

> ⚠️ In order to upgrade the MongoDB databases, you must stop all services *first*. These instructions assume that you have all the MongoDB databases on the same host (for example, the Database Host). If you have installed the MongoDB databases on *separate* hosts (for example, the Insight MongoDB on the Insight Server Host), you will have to go to *each* host to stop the services before you can upgrade any of the MongoDB databases.

If MongoDB is at a version prior to 3.2, do the following:

1. Stop *all* the services that use MongoDB. See the warning at the beginning of this topic.

   ```
   [root]# service nitro-web-services stop # If Nitro Web Services is part of your
   configuration
   [root]# service tomcat stop # If MWS is part of your configuration
   [root]# service insight stop # If Insight is part of your configuration
   [root]# service moab stop
   ```

2. Confirm that nothing is connected to MongoDB.

   ```
   [root]# netstat -antp | egrep '(27017|28017).*ESTABLISHED'
   ```

3. Dump the database.

   ```
   [root]# cd /root
   [root]# mongodump -u admin_user -p secret1
   [root]# cp -a dump dump.save
   [root]# rm -rf dump/admin/system.users.* # Cannot restore users.
   ```

4. Install MongoDB 3.2.x.

   ```
   [root]# service mongod stop
   [root]# chkconfig mongod off
   [root]# rpm -e --nodeps $(rpm -qa 'mongo*')
   [root]# rm -rf /tmp/mongo*.sock /var/run/mongo* /var/lib/mongo* /var/log/mongo*
   [root]# yum install mongodb-org
   [root]# chkconfig mongod on
   [root]# service mongod start
   ```

5. Restore the database.

   ```
   [root]# cd /root
   [root]# mongorestore
   ```

6. Create the users.

> ℹ The `admin_user` is required. All other users are required only for the products that are part of your system configuration. For example, if Nitro Web Services is not part of your confirmation, you do not need to add the `nitro_user`.

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

        use moab
        db.createUser({"user": "moab_user", "pwd": "secret2", "roles":
["dbOwner"]})
        db.createUser({"user": "mws_user", "pwd": "secret3", "roles": ["read"]})
        db.createUser({"user": "insight_user", "pwd": "secret4", "roles":
["read"]})

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

        use insight
        db.createUser({"user": "insight_user", "pwd": "secret4", "roles":
["dbOwner"]})
        db.createUser({"user": "mws_user", "pwd": "secret3", "roles": ["read"]})

        use nitro-db
        db.createUser({"user": "nitro_user", "pwd": "secret5", "roles":
["dbOwner"]})

        exit
```

7. Set MongoDB Configuration Options.

   - The configuration file for MongoDB is `/etc/mongod.conf`. See https://docs.mongodb.com/manual/reference/configuration-options for information.

   - Adaptive Computing recommends that you set `security.authorization` to `enabled`. See https://docs.mongodb.com/manual/reference/configuration-options/#security-options for more information.

> ℹ By default, `/etc/mongod.conf` sets `net.bindIp` to 127.0.0.1. You will need to change this setting if the MongoDB server needs to be accessible from other hosts or from other interfaces besides loopback. See https://docs.mongodb.com/manual/reference/configuration-options/#net-options for more information.

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

```
   authorization: enabled
storage:
  dbPath: /var/lib/mongo
  journal:
    enabled: true
systemLog:
  destination: file
  logAppend: true
  path: /var/log/mongodb/mongod.log
```

8. Restart MongoDB.

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

9. Follow the instructions to upgrade your Moab HPC Suite components.

3.33 Upgrading Torque Resource Manager (RPM)

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

> ⓘ 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.

3.33.1 Upgrade Steps

Do the following:

1. If you installed Torque Server on its own host *or* if Torque Server is the first component being upgraded on a host with other RPM installations, complete the steps to prepare the host.

   Do the same as needed for each Torque MOM Host (compute node).

   See 3.28 Preparing for RPM Upgrades - page 304 for more information.

2. Stop all Torque Server, Torque MOM, and Torque Client Services. See 3.33.2 Stop Torque Services - page 314.

3. Upgrade Torque Server, Torque MOMs, and Torque Clients. See 3.33.3 Upgrade Torque Server, MOMs, and Clients - page 315.

4. Start all Torque Server, Torque MOM, and Torque Client Services. See 3.33.4 Start Torque Services - page 317.

3.33.2 Stop Torque Services

Do the following:

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

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

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

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

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

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

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

3.33.3 Upgrade Torque Server, MOMs, and Clients

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

Do the following:

1. Upgrade Torque Server.

   On the Torque Server Host, install the upgrade.

```
[root]# yum update hwloc* moab-torque*
```

2. Upgrade Torque MOMs.

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

   Do the following:

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

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

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

```
[root]# yum update hwloc* moab-torque-*
```

c. On *each* Torque MOM Host, confirm that cgroups have been mounted; if not, mount them.

   a. Run *lssubsys -am*.

   b. If the command is not found, or you do not see something similar to the following, then cgroups are *not* mounted, continue with these instructions.

```
ns
perf_event
net_prio
cpuset /cgroup/cpuset
cpu /cgroup/cpu
cpuacct /cgroup/cpuacct
memory /cgroup/memory
devices /cgroup/devices
freezer /cgroup/freezer
net_cls /cgroup/net_cls
blkio /cgroup/blkio
```

   c. Install the cgroup library package and mount cgroups.

```
[root]# yum install libcgroup
[root]# service cgconfig start
[root]# chkconfig cgconfig on
```

   d. Run *lssubsys -am* again and confirm cgroups are mounted.

3. Upgrade Torque Clients.

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

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

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

b. On the Torque Client Host, use the uploaded RPMs to update the host.

```
[root]# yum update hwloc* moab-torque-*
```

3.33.4 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.

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

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

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

3. On the Torque Server Host, start up the Torque server.

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

3.34 Upgrading Moab Workload Manager (RPM)

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

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

> In this topic:
>

3.34.1 Upgrade Steps

Do the following:

1. If you installed Moab Server on its own host *or* if Moab Server is the first component being upgraded on a host with other RPM installations, complete the steps to prepare the host. See 3.28 Preparing for RPM Upgrades - page 304 for more information.

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

3. Upgrade Moab Server. See 3.34.2 Upgrade Moab Server - page 318.

3.34.2 Upgrade Moab Server

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

> ℹ The Moab RPM automatically creates a backup of all relevant files. These backups are stored in `/var/tmp/backup-<rpmName>-<timestamp>.tar.gz`.
>
> If changes are detected between any existing configuration files and new configuration files, a version of the new configuration file will be saved under `<configurationFileLocation>/<fileName>.rpmnew`.

On the Moab Server Host, do the following:

1. Stop Moab.

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

2. Install the upgrade.

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

3. Merge the configuration files.

   > ℹ️ 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 9.1.3 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.

   > ℹ️ The recommended layout for the `/opt/moab/etc/` directory appears as follows:
   >
   > ```
   > -rw-r--r--. 1 root moab 2323 Oct 25 23:33 config.moab.pl
   > -rw-r--r--. 1 root moab  989 Oct 25 23:33 config.sql.pl
   > -rw-r--r--. 1 root moab 1659 Oct 25 23:33 elastic.cfg
   > lrwxrwxrwx. 1 root root   26 Jun 22 16:29 moab.cfg ->
   > /opt/moab/etc/moab.hpc.cfg
   > drwxr-xr-x. 2 root moab 4096 Oct 25 23:33 moab.d
   > -rw-r--r--. 1 root moab 3859 Jul  6 17:14 moab.hpc.cfg
   > -rw-r--r--. 1 root root  513 Jun 22 17:27 moab.lic
   > -rw-------. 1 root moab  196 Jun 24 23:10 moab-private.cfg
   > drwxr-xr-x. 2 root moab 4096 Oct 27 23:17 power-management
   > ```

   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!
      ```

      > ℹ️ 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`.

> **ⓘ** If you are upgrading from a version prior to 9.0, the Torque RPMs will have moved the Torque binaries from /usr to /usr/local. Make sure that your RMCFG[] SUBMITCMD parameter is set to the correct path for qsub.
>
> ```
> [root]# vi /opt/moab/etc/moab.cfg
> RMCFG[pbs]        TYPE=PBS SUBMITCMD=/usr/local/bin/qsub
> ```

- 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
  ```

c. 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
```

4. Start Moab.

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

3.35 Upgrading Moab Accounting Manager (RPM)

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

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

3.35.1 Upgrade Steps

Do the following:

1. If you installed MAM Server on its own host *or* if MAM Server is the first component being upgraded on a host with other RPM installations, complete the steps to prepare the host.

    Do the same as needed for the MAM GUI Host and each MAM Client Host.

    See 3.28 Preparing for RPM Upgrades - page 304 for more information.

2. Upgrade MAM Server. See 3.35.2 Upgrade MAM Server - page 321.

3. Upgrade MAM GUI. See 3.35.2.A Upgrade Remote MAM GUI - page 324.

4. Upgrade MAM Web Services. See 3.35.2.B Upgrade Remote MAM Web Services - page 324.

5. Upgrade MAM Clients. See 3.35.2.C Upgrade Remote MAM Clients - page 324.

3.35.2 Upgrade MAM Server

> ℹ You *must* complete all the previous upgrade steps in this topic before upgrading MAM Server. See the list of steps at the beginning of this topic.

On the MAM Server Host, do the following:

1. Stop MAM.

```
[root]# service mam stop
```

2. Install the upgrade.

> ℹ The MAM RPM name has changed between version 8.1 and 9.0. The RPM obsoleted
> 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:

```
[root]# for i in /opt/mam/etc/{gold,goldd,goldg,site}.conf
do
cp -p ${i} ${i}.rpmsave
done
[root]# rpm -e --nopostun moab-hpc-enterprise-suite moab-hpc-accounting-manager
[root]# yum install moab-accounting-manager
[root]# for i in /opt/mam/etc/mam-*.conf
do
cp -p ${i} ${i}.rpmnew
done
[root]# \cp -f /opt/mam/etc/gold.conf.rpmsave /opt/mam/etc/mam-client.conf
[root]# \cp -f /opt/mam/etc/goldd.conf.rpmsave /opt/mam/etc/mam-server.conf
[root]# \cp -f /opt/mam/etc/goldg.conf.rpmsave /opt/mam/etc/mam-gui.conf
[root]# \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
```

3. Compare your existing configuration files (/opt/mam/etc/mam-*.conf) with those distributed
   with the new release (/opt/mam/etc/mam-*.conf.rpmnew) and merge the differing lines into
   your configuration files.

4. Start the mam service.

```
[root]# service mam start
```

5. If you are upgrading MAM from a version prior to 8.0, add the new mam user as a MAM
   Accounting Admin.

```
[root]# su -c "mam-create-user -u mam -d \"Accounting Admin\"" moab
[root]# su -c "mam-modify-role -r SystemAdmin --add-user mam" moab
[root]# perl -p -i -e 's/moab/mam/ if /^super.user/' /opt/mam/etc/mam-server.conf
```

6. If your PostgreSQL database version is prior to version 9.1, update the postgresql configuration
   to avoid interpreting backslashes as escape characters.

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

```
standard_conforming_strings = on

[root]# service postgresql restart
```

7. If upgrading MAM from a version prior to 9.1, migrate the Moab Accounting Manager database from your current version to 9.1.

   a. Run one or more migration scripts. You must run every incremental migration script between the version you are currently using and the new version (9.1). The migration scripts are located in the `/usr/share/moab-accounting-manager/` directory. 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.

   > 🛈 The migration scripts *must* be run as the mam user.

   For example, if you are migrating from Moab Accounting Manager version 7.2, you must run six 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, the fifth to migrate the database schema from 8.1 to 9.0, and the sixth to migrate the database schema from 9.0 to 9.1.

   ```
   [root]# su - mam

   [mam]$ /usr/share/moab-accounting-manager/migrate_7.2-7.3.pl
   [mam]$ /usr/share/moab-accounting-manager/migrate_7.3-7.5.pl
   [mam]$ /usr/share/moab-accounting-manager/migrate_7.5-8.0.pl
   [mam]$ /usr/share/moab-accounting-manager/migrate_8.0-8.1.pl
   [mam]$ /usr/share/moab-accounting-manager/migrate_8.1-9.0.pl
   [mam]$ /usr/share/moab-accounting-manager/migrate_9.0-9.1.pl
   ```

   b. Verify that the resulting database schema version is 9.1.

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

   Name                   Version Description
   ---------------------- ------- ------------------
   Moab Accounting Manager 9.1     Commercial Release
   ```

8. Verify that the executables have been upgraded to 9.1.3.

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

   Moab Accounting Manager version 9.1.3
   ```

9. If you are upgrading MAM from a version prior to 9.1.0, and you wish to use MAM Web Services, perform the following procedure (provided in the Install Moab Accounting Manager (RPM) topic):

   - 3.9.7 Configure MAM Web Services - page 194
   - 3.9.9 Access MAM Web Services - page 196

3.35.2.A Upgrade Remote MAM GUI

If you are using the MAM GUI and the MAM GUI Host is different from the MAM Server Host, then do the following on the MAM GUI Host:

1. Install the upgrade.

    - If you are upgrading the MAM RPM from a version prior to 9.0, do the following:

      ```
      cp -p /opt/mam/etc/goldg.conf /opt/mam/etc/goldg.conf.rpmsave
      rpm -e --nopostun moab-hpc-accounting-manager
      yum install moab-accounting-manager
      cp -p /opt/mam/etc/mam-gui.conf /opt/mam/etc/mam-gui.conf.rpmnew
      \cp -f /opt/mam/etc/goldg.conf.rpmsave /opt/mam/etc/mam-gui.conf
      ```

    - If you are upgrading the MAM RPM from a version at or after 9.0, do the following:

      ```
      [root]# yum update moab-accounting-manager
      ```

2. Compare your current gui configuration file (/opt/mam/etc/mam-gui.conf) with the one distributed with the new release (/opt/mam/etc/mam-gui.conf.rpmnew) and merge the differing lines into your current configuration file.

3.35.2.B Upgrade Remote MAM Web Services

If you are using MAM Web Services and the MAM Web Services Host is different from the MAM Server Host, then do the following on the MAM Web Services Host:

1. Install the upgrade.

    ```
    [root]# yum update moab-accounting-manager
    ```

2. Compare your current web services configuration file (/opt/mam/etc/mam-ws.conf) with the one distributed with the new release (/opt/mam/etc/mam-ws.conf.rpmnew) and merge the differing lines into your current configuration file.

3. Restart the HTTP server daemon.

    ```
    [root]# service httpd restart
    ```

3.35.2.C Upgrade Remote MAM Clients

If you are have any MAM Client Hosts that are different from the MAM Server Host or MAM GUI Hosts, then do the following on each MAM Client Host:

1. Install the upgrade.

   - If you are upgrading the MAM RPM from a version prior to 9.0, do the following:

   ```
   cp -p /opt/mam/etc/gold.conf /opt/mam/etc/gold.conf.rpmsave
   rpm -e --nopostun moab-hpc-accounting-manager
   yum install moab-accounting-manager
   cp -p /opt/mam/etc/mam-client.conf /opt/mam/etc/mam-client.conf.rpmnew
   \cp -f /opt/mam/etc/gold.conf.rpmsave /opt/mam/etc/mam-client.conf
   ```

   - If you are upgrading the MAM RPM from a version at or after 9.0, do the following:

   ```
   [root]# yum update moab-accounting-manager
   ```

2. Compare your current client configuration file (/opt/mam/etc/mam-client.conf) with the one distributed with the new release (/opt/mam/etc/mam-client.conf.rpmnew) and merge the differing lines into your current configuration file.

3.36 Upgrading Moab Web Services (RPM)

This topic provides instructions to upgrade Moab Web Services to the latest release version using the RPM upgrade method.

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

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

Perform the following steps:

1. Confirm Moab Workload Manager RPM Upgrade
2. Upgrade to Java 8
3. Upgrade to MongoDB 3.2.x
4. Back up the MongoDB Databases
5. Upgrade MWS Server

3.36.1 Confirm Moab Workload Manager RPM Upgrade

As Moab Web Services resides on the same host as Moab Workload Manager, you will need to confirm the Moab Server RPM upgrade has completed on their shared host. See 3.34 Upgrading Moab Workload Manager (RPM) - page 318 for more information.

3.36.2 Upgrade to Java 8

> ℹ 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 3.36.2 Upgrade to Java 8 - page 326 instructions.

3.36.3 Upgrade to MongoDB 3.2.x

Beginning with version 9.1.0, Moab Web Services requires MongoDB 3.2.x.

> ℹ Setting per-user limits on various resources can prevent MongoDB from closing connections if the number of connections grows to high. See Review and Set Resource Limits (https://-docs.mongodb.com/manual/reference/ulimit/#review-and-set-resource-limits) for more information about using the *ulimit* command to review and set resource limits.

On the MWS MongoDB host, do the following:

1. Check your MongoDB server version.

```
[root]# rpm -qa | grep mongo
```

2. If the MongoDB server version is older than 3.2.x, then you will need to upgrade the databases. See 3.32 Upgrading to MongoDB 3.2.x (RPM) - page 311 for more information.

3.36.4 Back up the MongoDB Databases

> ℹ This procedure is only applicable if you did not have to upgrade the MongoDB databases earlier in this topic. See 3.36.3 Upgrade to MongoDB 3.2.x - page 327 for more information.

On the MWS MongoDB host, do the following:

1. Stop all services that are using the MongoDB databases.

2. Back up the MongoDB databases.

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

3. Restart the services.

3.36.5 Upgrade MWS Server

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

> ℹ The MWS RPM automatically creates a backup of all relevant files. These backups are stored in `/var/tmp/backup-<rpmName>-<timestamp>.tar.gz`.
>
> If changes are detected between any existing configuration files and new configuration files, a version of the new configuration file will be saved under `<configurationFileLocation>/<fileName>.rpmnew`.

On the MWS Server Host, do the following:

1. Stop Tomcat.

   - If your prior MWS Server version used Tomcat 6, disable the tomcat 6 service.

     ```
     [root]# service tomcat6 stop
     [root]# chkconfig tomcat6 off
     ```

     > ℹ Tomcat 6 is not supported for MWS 9.0 and after. The MWS RPM will automatically install Tomcat 7.

   - If your prior MWS Server version used Tomcat 7, stop the tomcat 7 service.

     ```
     [root]# service tomcat stop
     ```

2. Install the upgrade.

   ```
   [root]# yum update moab-web-services*
   ```

3. Merge the changes in the `/opt/mws/etc/mws-config.groovy.rpmnew` file into your existing `/opt/mws/etc/mws-config.groovy` file.

   a. Depending on your current MWS version, do the following as needed:

      - If Insight is part of your configuration:

        ○ **remove** the Insight PostgreSQL information (dataSource_insight.username, dataSource_insight.password, dataSource_insight.url); prior to version 9.1.

          > ℹ Version 9.1 removed the Insight PostreSQL database.

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

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

      - If Viewpoint is part of your configuration, register Viewpoint as client; prior to version 9.0, do the following:

Edit the grails.plugin.springsecurity.oauthProvider.clients array in
`/opt/mws/etc/mws-config.groovy` and specify a client id and a client secret.
Leave the authorizedGrantTypes field unchanged.

> ⓘ The following is a suggested script for generating the client secret:
>
> ```
> dd if=/dev/urandom count=24 bs=1 2>/dev/null | base64
> ```

```
[root]# vi /opt/mws/etc/mws-config.groovy
        grails.plugin.springsecurity.oauthProvider.clients = [
                [
                        clientId: "viewpoint",
                        clientSecret: "<ENTER-CLIENTSECRET-HERE>",
                        authorizedGrantTypes: ["password"]
                ]
        ]
```

- 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

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

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

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

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

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

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

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

// 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
      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
                              maxBackupIndex: 10, //Retain only the 10 most recent log files
logs to save space
                              layout: new com.ace.mws.logging.ACPatternLayout(), //Configure
format of each log entry
                              threshold: org.apache.log4j.Level.ERROR //Ignore any logging e
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.gapiParsers',
                      'grails.app.gapiRequests',
                      'grails.app.gapiSerializers',
                      'grails.app.translators',
```

```
                          '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'
}
```

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

4. 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`.

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

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

6. 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.

7. Verify the following lines are added 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.

8. Start Tomcat.

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

3.37 Upgrading Moab Insight (RPM)

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

> ℹ️ 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 the Insight Server

> ⚠️ Insight version 9.0.2 or 9.0.3 is required to upgrade to Insight version 9.1.0 or after.

Do the following:

1. If upgrading from an Insight version prior to 9.0.2, you need to first upgrade Insight to version 9.0.2 or 9.0.3. Those upgrade instructions are available from the Adaptive Computing Documentation page.

2. On the Moab Server Host, stop Moab from sending messages to Insight.

   ```
   [root]# mschedctl -s
   ```

3. If you are upgrading Insight from a version *prior* to 9.1.0, confirm the MongoDB databases are upgraded to 3.2.x. See 3.32 Upgrading to MongoDB 3.2.x (RPM) - page 311 for more information.

4. If you are upgrading Insight from version 9.1.0 or later, do the following on the Insight MongoDB host.

   a. Stop all services that are using the MongoDB databases.

   b. Back up the MongoDB databases.

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

   c. Restart the services.

5. On the Insight Server Host, do the following:

   a. If you have not already done so, complete the steps to prepare the Insight Server Host for the upgrade. See 3.28 Preparing for RPM Upgrades - page 304 for more information.

   b. Stop Insight

   ```
   [root]# service insight stop
   ```

   c. Back up the Insight home directory.

```
[root]# cp -r /opt/insight /opt/insight-<version>-backup
```
Where `<version>` if the product version being backed up.

6. If upgrading from version 9.0.2 or 9.0.3, the Insight PostgreSQL database is no longer used. You can optionally backup the PostgreSQL database. On the host where the Insight PostgreSQL database resides, do the following:

```
[root]# su - postgres
[postgres]$ pg_dump moab_insight > /tmp/moab_insight_<version>.dump
[postgres]$ pg_dump moab_insight_reference > /tmp/moab_insight_reference_
<version>.dump
[postgres]$ exit
[root]# mv /tmp/moab_insight_<version>.dump /opt
[root]# mv /tmp/moab_insight_reference_<version>.dump /opt
```

7. On the Insight Server Host, do the following:

   a. Install the upgrade.

```
[root]# yum update moab-insight
```

   b. Merge the new configuration from `/opt/insight/etc/config.groovy.rpmnew` into `/opt/insight/etc/config.groovy`.

   c. Verify the insight user has read access to the `/opt/insight/etc/config.groovy` file.

```
[root]# ls -l /opt/insight/etc/config.groovy
-rw-------. 1 insight insight 4687 Oct 24 17:57 /opt/insight/etc/config.groovy
```

   d. Verify the following line is added to the end of `/opt/insight/etc/insight.conf`:

```
JAVA_HOME="/usr/java/latest"
```

   e. Start Insight.

```
[root]# service insight start
```

   f. Wait for and confirm the database upgrade completed. *All* data must be transferred before the upgrade is complete.

     When the upgrade is completed, you will see output similar to the following in your `/opt/insight/log/insight.log file`.

```
2016-06-28T06:25:13.120-0600    main    INFO
com.ace.insight.data.service.dbinit.DbUpgradeService 0 Database has been
upgraded to current version
```

8. On the Moab Server Host, have Moab resume sending messages to Insight.

```
[root]# mschedctl -r
```

3.38 Upgrading Moab Viewpoint (RPM)

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

> ℹ️ 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.

In this topic:

3.38.1 Upgrade the Viewpoint Server

On the Viewpoint Server Host, do the following:

1. If you installed Viewpoint Server on its own host *or* if Viewpoint Server is the first component being upgraded on a host with other RPM installations, complete the steps to prepare the host. See for more information.

2. Stop the Apache service.

   ```
   [root]# service httpd stop
   ```

3. Install the upgrade.

   - If you are upgrading from Viewpoint 9.0.0, do the following:

     > ⚠️ Beginning with the 9.0.1 release, several variables became obsolete. In addition, the configuration files were renamed and/or moved.

   a. Remove these obsolete variables from /etc/httpd/conf.d/viewpoint.conf:

       ○ IRIS_LOGS_FILENAME

       ○ IRIS_LOGS_PATH

       ○ IRIS_SESSION_FILE_PATH

       ○ IRIS_TEMPLATE_DEBUG

   The IRIS_DEBUG variable must *not* be used in production; also remove this variable from /etc/httpd/conf.d/viewpoint.conf.

   b. Back up configuration files.

```
[root]# cp -p /opt/viewpoint/config/config.json
/etc/httpd/conf.d/viewpoint.conf /tmp
```

   c. Back up certificates to connect to the file manager (if Viewpoint connects to file manager over SSL).

```
[root]# cp -p /opt/viewpoint/webdav_client/client-cert.pem
/opt/viewpoint/webdav_client/client-key.pem /opt/viewpoint/webdav_client/ca-
cert.pem /tmp
```

   d. Uninstall Viewpoint and some packages that are no longer needed.

```
[root]# rpm -e --nodeps moab-viewpoint
[root]# rpm -q --quiet python-importlib && rpm -e python-importlib
[root]# rpm -q --quiet mod_wsgi && rpm -e mod_wsgi
```

   e. Remove some leftover files.

```
[root]# rm -rf /var/log/viewpoint /opt/viewpoint
/etc/httpd/conf.d/viewpoint.conf /etc/cron.daily/viewpoint.sh
```

   f. Install the new Viewpoint RPM.

```
[root]# yum install moab-viewpoint
```

- If you are upgrading Viewpoint from 9.0.1 or 9.0.2, do the following:

   a. Back up configuration files.

```
[root]# cp -p /opt/viewpoint/lib/viewpoint/config/config.json
/opt/viewpoint/etc/viewpoint.cfg /tmp
```

   b. Back up certificates to connect to the file manager (if Viewpoint connects to file manager over SSL).

```
[root]# cp -p /opt/viewpoint/lib/viewpoint/webdav_client/client-cert.pem
/opt/viewpoint/lib/viewpoint/webdav_client/client-key.pem
/opt/viewpoint/lib/viewpoint/webdav_client/ca-cert.pem /tmp
```

    c. Uninstall Viewpoint.

```
[root]# rpm -e --nodeps moab-viewpoint
```

    d. Install the Viewpoint RPM.

```
[root]# yum install moab-viewpoint
```

- If you are upgrading Viewpoint from 9.1.0 or later, do the following:

Upgrade the Viewpoint RPM.

```
[root]# yum install moab-viewpoint
```

4. If you are upgrading from Viewpoint 9.0.0, restore certificates to their new location:

```
[root]# cp -p /tmp/client-cert.pem /tmp/client-key.pem /tmp/ca-cert.pem
/opt/viewpoint/lib/viewpoint/webdav_client/
```

5. Merge customizations into the new `viewpoint.conf` file.

If you are upgrading Viewpoint from 9.0.0, merge the customizations in the old /etc/httpd/conf.d/viewpoint.conf into the /opt/viewpoint/etc/viewpoint.cfg.

> ℹ️ All IRIS_DATABASE* SetEnv entries in /etc/httpd/conf.d/viewpoint.conf are obsolete. Database environment variables are now stored in /opt/viewpoint/etc/viewpoint.cfg. Therefore, move all your uncommented database SetEnv entries into the environment section of /opt/viewpoint/etc/viewpoint.cfg; and edit as needed to reflect the 9.0.2 renaming (see the warning later in this step for more information).
>
> Beginning with version 9.0.2, all IRIS_* variables were renamed to VIEWPOINT_*

After you are finished, your `/opt/viewpoint/etc/viewpoint.cfg` will look something like this:

```
[admin]
username = viewpoint-admin
password = pbkdf2_
sha256$20000$ZHeToCJgrSUH$+xmzYdhpqZCJokxO9eGzyr2B6jrfCgLlBT+pBgMis4w=

[environment]
VIEWPOINT_DATABASE_HOST = localhost
VIEWPOINT_DATABASE_PORT = 5432
VIEWPOINT_DATABASE_NAME = moab_viewpoint
VIEWPOINT_DATABASE_USER = moab_viewpoint
VIEWPOINT_DATABASE_PASSWORD = changeme!

[settings]
past_hours = 24
future_hours = 4
```

> ℹ Viewpoint has a number of environment variables used to configure a Viewpoint installation and troubleshoot operating issues. See 5.6.1 General Configuration Issues - page 423 for more information about Viewpoint environment variables.

6. If you are upgrading from Viewpoint 9.0.0, change the admin password in `/opt/viewpoint/etc/viewpoint.cfg`.

> ⚠ For security purposes, the admin password is encrypted. In the example above, the default is the encrypted equivalent to "changeme!", which is the default for the Viewpoint instance. Change this default password to a different encrypted password. To encrypt the password, do the following (substituting "changeme!" with your password):
>
> ```
> [root]# echo -n 'changeme!' | /opt/viewpoint/bin/viewpoint makehash
> Using default hasher
> pbkdf2_sha256$20000$ZHeToCJgrSUH$+xmzYdhpqZCJokxO9eGzyr2B6jrfCgLlBT+pBgMis4w=
> ```

> ℹ The default hashing algorithm is pbkdf2_sha256. To show the other available algorithms, run `/opt/viewpoint/bin/viewpoint makehash --help`

7. Initialize Viewpoint's PostgreSQL database.

- If you are upgrading from Viewpoint 9.0.0, do the following:

```
[root]# /opt/viewpoint/bin/viewpoint migrate --fake-initial
```

- If you are upgrading from Viewpoint 9.0.1 or later, do the following:

```
[root]# /opt/viewpoint/bin/viewpoint migrate
```

> ℹ When running `viewpoint migrate`, `max_user_instances` should be set to at least 128. To set `max_user_instances`, execute the following command.
>
> ```
> [root]# echo "128" > /proc/sys/fs/inotify/max_user_instances
> ```

8. Start the Apache service.

```
[root]# service httpd start
```

3.38.1.A Update the Permissions List

Once you have updated the Viewpoint Server, you will need to update the MWS configuration in the Viewpoint Portal to sync the permissions list.

Do the following:

1. Using a web browser, navigate to your Viewpoint instance. (http://`<viewpoint_`
`host>`:8081; where `<viewpoint_host>` is the IP address or name of the Viewpoint Server
Host).

2. Log in as the Viewpoint administrative user (viewpoint-admin, by default). The Configuration
page displays with the Basic Configuration page selected.

3. In the MWS Configuration area, click `SAVE`.

3.38.2 Upgrade the Viewpoint File Manager Service

On the Moab Server Host where the Viewpoint File Manager Service resides, do the following:

1. Install the moab-viewpoint-filemanager package.

```
[root]# yum install moab-viewpoint-filemanager
[root]# yum install python-setuptools
```

2. Restart the File Manager Service.

```
[root]# service acfileman restart
```

3.38.3 Update the Viewpoint License

- If upgrading from 9.0.1 or later, no action is needed; your existing license remains in effect.

- If upgrading from 9.0.0, you will need to license Viewpoint for the first time. Follow the
instructions in 3.12.4 License Viewpoint - page 222.

3.38.4 Verify Base Roles are Present

As part of the Viewpoint upgrade, if you are upgrading from version 9.0.2 or prior, you will need to
verify that the base roles are present.

Do the following:

1. Using a web browser, navigate to your Viewpoint instance. (http://`<viewpoint_`
`host>`:8081; where `<viewpoint_host>` is the IP address or name of the Viewpoint Server
Host).

2. Log in as the MWS administrative user (moab-admin, by default).

3. Click `Configuration` from the menu. The Basic Configuration page displays with additional
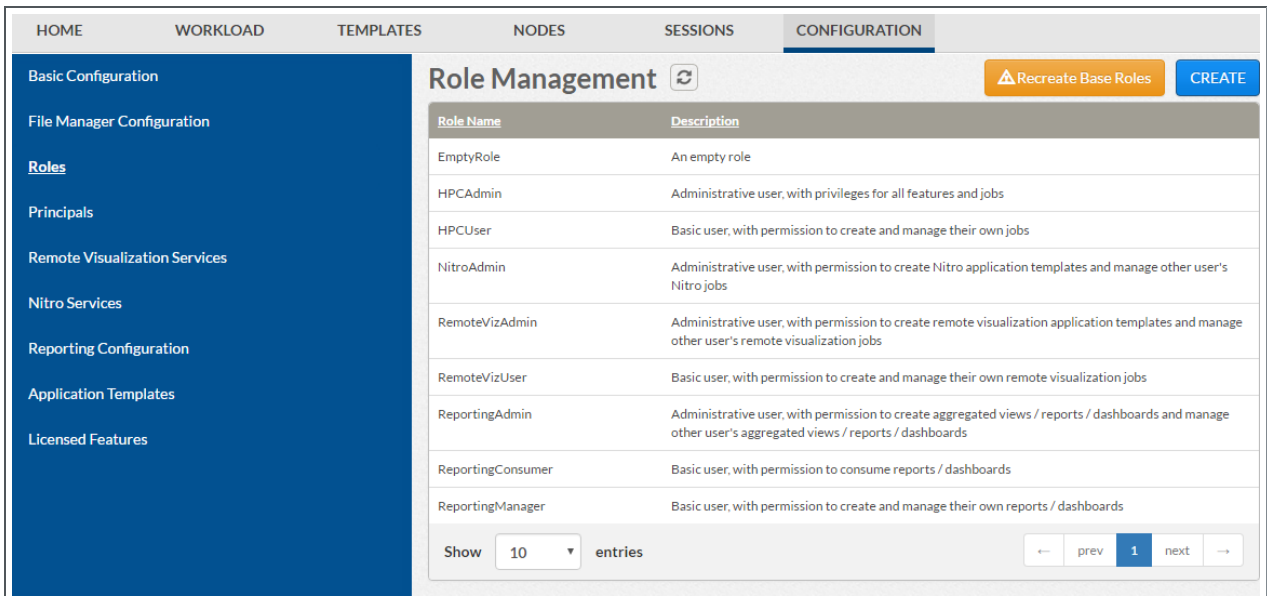options in the left pane. For example:

4. Click `Roles` from the left pane. The Role Management page displays.

5. If all the roles *are* there, continue with the procedure in 3.12.8 Grant Users Access to Viewpoint - page 230.

However, if one of the base roles is not present, you will need to recreate (restore) the base roles.

6. If you need to recreate the base roles, the Recreate Base Roles button displays on the Role Management page. For example:

- Click `Recreate Base Roles`. Viewpoint will restore the roles.

> ⓘ You can also modify the default roles and create new roles as needed. See "About Roles" in the *Moab ViewpointReference Guide* for more information.

3.39 Upgrading the Reporting Framework

This topic provides instructions to upgrade the Reporting Framework to the latest release version using the RPM upgrade method.

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

> ℹ You will need to upgrade all components in the Reporting Framework at once. Do not attempt to upgrade one component but not another, or reports may not work properly.

Perform the following steps:

1. Stop Spark
2. Upgrade Spark on the Reporting Master
3. Upgrade Spark on the Reporting Workers
4. Upgrade Reporting
5. Upgrade Reporting Web Services (RWS)
6. Ensure Viewpoint and the Reporting Framework Are Communicating

3.39.1 Stop Spark

1. On the Reporting Master host, do the following:

```
[root]# service spark-master stop
```

2. On each Reporting Worker host, do the following:

```
[root]# service spark-worker stop
```

3.39.2 Upgrade Spark on the Reporting Master

If you are upgrading the Moab components from 9.1.1.1 you will need to upgrade Spark on the Reporting Master. If you are upgrading your Moab components from 9.1.2, the proper version of Spark (2.1.2) should already be installed and you can skip the rest of this section.

You will have to uninstall the old version of Spark before installing the new.

> ⚠ Do not attempt to do a yum update on the Spark RPM.

On the Reporting Master host, do the following:

1. Uninstall Spark.

```
[root]# yum remove spark
```

2. Re-install Spark.

```
[root]# yum install spark
```

3. Make sure the SPARK_MASTER_URL variable is set correctly in the configuration files. SPARK_MASTER_URL should the hostname of the Reporting Master host.

> ℹ Use the actual hostname (which should be DNS resolvable) when specifying the Reporting Server host. Do not use localhost.

```
[root]# vi /etc/profile.d/spark.sh
export SPARK_MASTER_URL=spark://<reporting_master_host>:7077
```

4. Make sure user resource limits have been increased.

```
[root]# vi /etc/security/limits.conf
```

Add the following lines, if not already present.

```
root soft core unlimited
root soft stack unlimited
root soft nproc unlimited

root hard core unlimited
root hard stack unlimited
root hard nproc unlimited

spark soft core unlimited
spark soft stack unlimited
spark soft nproc unlimited

spark hard core unlimited
spark hard stack unlimited
spark hard nproc unlimited
```

5. Start the Spark master service.

```
[root]# chkconfig spark-master on
[root]# service spark-master start
```

6. Verify the Spark Master is running by opening the following URL in a web browser and verifying that the status is ALIVE.

```
http://<reporting_master_host>:8082
```

3.39.3 Upgrade Spark on the Reporting Workers

If you are upgrading the Moab components from 9.1.1.1 you will need to upgrade Spark on the Reporting Workers. If you are upgrading your Moab components from 9.1.2 the proper version of Spark (2.1.2) should already be installed and you can skip the rest of this section.

You will have to uninstall the old version of Spark before installing the new.

⚠ Do not attempt to do a yum update on the Spark RPM.

On each Reporting Worker host, do the following:

1. Uninstall Spark.

```
[root]# yum remove spark
```

2. Re-install Spark.

```
[root]# yum install spark
```

3. Make sure the SPARK_MASTER_URL variable is set correctly in the configuration files. SPARK_MASTER_URL should the hostname of the Reporting Master host.

```
[root]# vi /etc/profile.d/spark.sh
export SPARK_MASTER_URL=spark://<reporting_master_host>:7077
```

4. Make sure user resource limits have been increased.

```
[root]# vi /etc/security/limits.conf
```

Add the following lines, if not already present.

```
root soft core unlimited
root soft stack unlimited
root soft nproc unlimited

root hard core unlimited
root hard stack unlimited
root hard nproc unlimited

spark soft core unlimited
spark soft stack unlimited
spark soft nproc unlimited

spark hard core unlimited
spark hard stack unlimited
spark hard nproc unlimited
```

5. Start the Spark worker service.

```
[root]# chkconfig spark-worker on
[root]# service spark-worker start
```

6. Verify the Spark Master sees the Spark Worker by opening the following URL in a web browser and ensuring that each Spark Worker host is listed in the `Workers` section.

```
http://<reporting_master_host>:8082
```

3.39.4 Upgrade Reporting

On the Reporting Master host, do the following:

1. Rename the old reporting jar and configuration file to make room for the new files.

```
[root]# source /etc/profile.d/hadoop.sh
[root]# hadoop fs -mv /reporting/reporting-framework.jar /reporting/reporting-
framework.jar-<version>-backup
[root]# hadoop fs -mv /reporting/application.conf /reporting/application.conf-
<version>-backup
```

2. Upgrade the reporting RPM.

```
[root]# yum update moab-viewpoint-reporting
```

3. Look to see if a new (`/opt/reporting/application.conf.rpmnew`) file has been created with the upgrade (i.e., the file exists and has a current timestamp). If so, this indicates that the configuration file (`/opt/reporting/application.conf`) provided by the new RPM differs from that provided by the old RPM, and furthermore that the administrator has made changes to the default configuration file. In this situation, compare the two files and merge the differing lines in the new default configuration file (`/opt/reporting/application.conf.rpmnew`) into your current configuration file (`/opt/reporting/application.conf`).

4. Upload the new reporting jar and configuration file to the correct folder on HDFS.

```
[root]# /opt/reporting/upload-reporting.sh
```

5. Make sure the upload succeeded. You should see a new `reporting-framework.jar` and `application.conf` in the `/reporting` directory.

```
[root]# hadoop fs -ls /reporting
Found 4 items
-rw-r--r--   3 spark spark        915 2017-10-27 18:16 /reporting/application.conf
-rw-r--r--   3 spark spark        915 2017-10-27 18:06 /reporting/application.conf-
9.1.1.1-backup
-rw-r--r--   3 spark spark  108944403 2017-10-27 18:16 /reporting/reporting-
framework.jar
-rw-r--r--   3 spark spark  103691535 2017-10-27 18:05 /reporting/reporting-
framework.jar-9.1.1.1-backup
```

3.39.5 Upgrade Reporting Web Services (RWS)

On the RWS Server host, do the following:

1. Stop Tomcat.

```
[root]# service tomcat stop
```

2. Make a backup of the RWS configuration folder.

```
[root]# cp -r /opt/reporting-web-services /opt/reporting-web-services-<version>-
backup
```

3. Update RWS.

```
[root]# yum update moab-reporting-web-services
```

4. Compare your current RWS configuration file (`application.properties`) with the one distributed with the new release (`application.properties.rpmnew`) and consider whether to merge the differing lines into your current configuration file.

```
[root]# vi /opt/reporting-web-services/etc/application.properties
```

At a minimum you will need to change the Spark home directory from `reporting.cluster.sparkHomeDir=/opt/spark-1.6.2` to `reporting.cluster.sparkHomeDir=/opt/spark-2.1.2`.

5. Restart Tomcat.

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

6. Verify that Reporting Web Services is running.

```
[root]# curl -X GET -v localhost:8080/rws/ping
< HTTP/1.1 200 OK
```

If the ping was not successful, check the logs at `/var/log/tomcat` for error messages.

7. When Reporting Web Services starts, it deploys the latest version of the Reporting Data Processing application to Spark. Verify this has happened by opening `http://<reporting_master_host>:8082` in a web browser. In the `Running Applications` section, you should see `ReportingDataProcessing` is in `RUNNING` state.

3.39.6 Ensure Viewpoint and the Reporting Framework Are Communicating

Do the following:

1. Log into Viewpoint as a user with Reporting administrative permissions.

2. Click `Reporting` to go to the Reporting page.

3. Click `Aggregated Views`.

4. Verify that the status of processing application is `RUNNING`.

3.40 Upgrading RLM Server (RPM)

Adaptive Computing *strongly* recommends that your RLM Server is version 12.1.2.

> In this topic:
>

3.40.1 Confirm if an Upgrade is Needed

On the RLM Server Host, run the following command to determine your current installed version of Adaptive Computing's RLM RPM.

```
[root]# rpm -q --qf '%{version}\n' ac-rlm
```

If the version reported is less than 12.1.2, continue with section .

3.40.2 Upgrade the RLM Server

On the RLM Server Host, do the following:

1. If you installed the RLM Server on its own host *or* if the RLM Server is the first component being upgraded on a host with other RPM installations, complete the steps to prepare the host. See for more information.

2. Stop the RLM service.

```
[root]# service rlm stop
```

3. Install the upgrade.

```
[root]# yum update ac-rlm
```

4. Restart the RLM service.

```
[root]# service rlm restart
```

3.41 Upgrading Remote Visualization (RPM)

If using Remote Visualization with Viewpoint Server 9.1.3, your Remote Visualization installation *must* use FastX 2.2-77.3.

3.41.1 Check Your Current Version

On the Remote Visualization Gateway Server Host, run the following command to determine your current version of FastX:

```
[root]# rpm -q StarNetFastX2
```

3.41.2 Upgrade Remote Visualization

3.41.2.A Upgrade the Gateway Server

If the version and release of the FastX RPM prior to upgrade is less than 2.2-77.3, do the following:

1. Make sure that your DNS server is configured for reverse lookups. Without reverse DNS, Session Servers will fail to register with your Gateway Server. As a result, authentication requests to the Gateway Server will fail because the Gateway Server will not be able to connect to any Session Servers.

2. On the Remote Visualization Gateway Server Host, do the following

   a. If you installed Remote Visualization Gateway Server on its own host *or* if Remote Visualization Gateway Server is the first component being upgraded on a host with other RPM installations, complete the steps to prepare the host. See 3.28 Preparing for RPM Upgrades - page 304 for more information.

b. Install or update FastX and all of its dependencies.

```
[root]# yum install ImageMagick-perl perl-Crypt-SSLeay perl-Net-SSLeay perl-X11-
Protocol StarNetFastX2
```

> **ℹ** If installing on RHEL, some packages may not be found in the standard RHEL distribution repositories. You will need to install the missing dependencies from EPEL or other reputable repositories.
>
> ```
> [root]# rpm -Uvh http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-
> release-6-8.noarch.rpm
>
> [root]# yum install yum-utils
> [root]# yum-config-manager --disable epel
> [root]# yum install --enablerepo=epel,rhel-6-server-eus-optional-rpms
> ImageMagick-perl perl-Crypt-SSLeay perl-Net-SSLeay perl-X11-Protocol
> StarNetFastX2
> ```

c. Change the config directory to be owned by "fastx".

```
[root]# chown fastx. /usr/lib/fastx2/config/ -R
```

d. Remove the gateway-server.json file, if it exists.

```
[root]# rm -f /usr/lib/fastx2/config/gateway-server.json
```

e. Restart the FastX service.
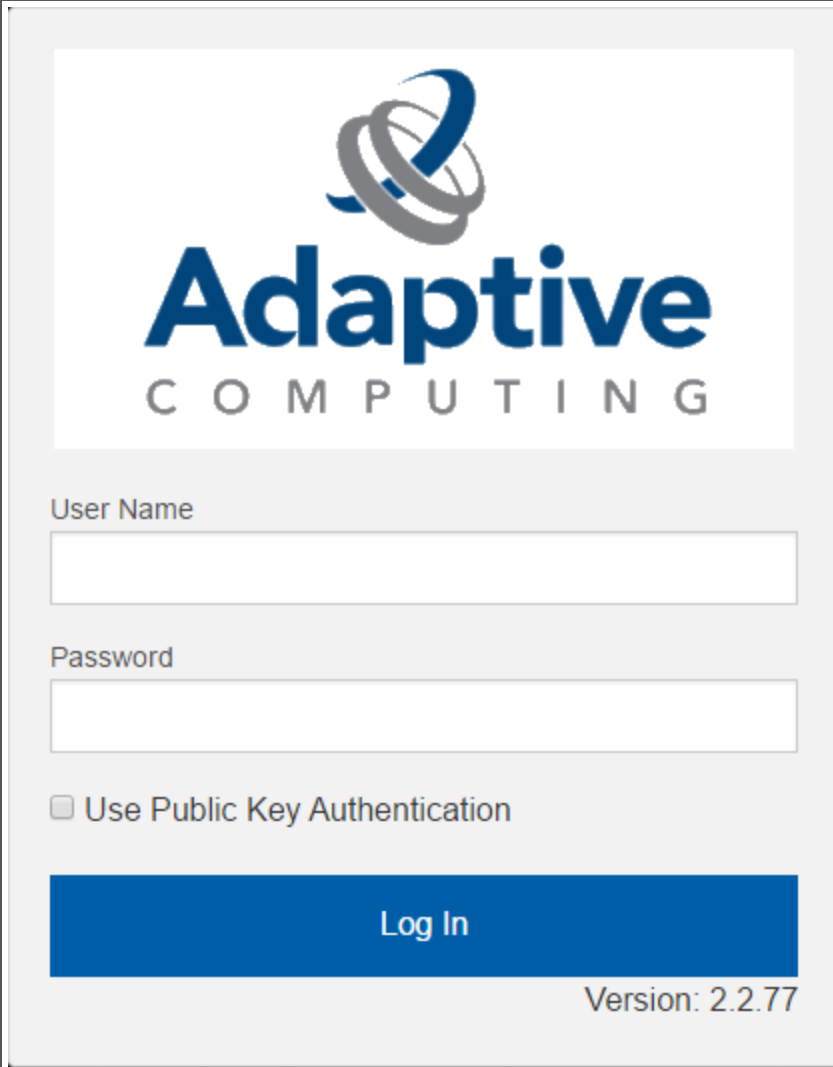
```
[root]# service fastx restart
```

3.41.2.B Configure the Gateway Server

If the version of the FastX RPM prior to upgrade was less than 2.2, do the following:

1. Using a web browser, navigate to your *secure* Remote Visualization Gateway Server instance. (**https**://<gateway_host>:3443; where <gateway_host> is the IP address or name of the Gateway Server Host).
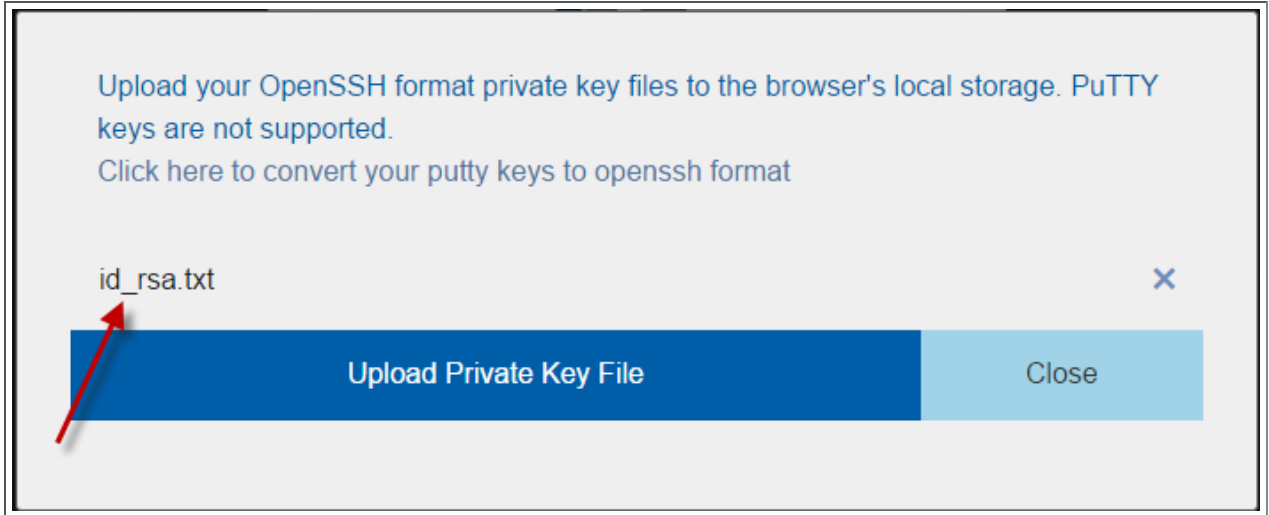
   The Log In page displays. For example:

2. Log in as the FastX admin user. Do *one* of the following:

   - If your authentication method is password-based, do the following:
     a. Enter the user name (default is "ace").
     b. Enter the password (default is "ace").
     c. Make sure the "Use Public Key Authentication" checkbox is cleared.
     d. Click `Log In`.

   - If your authentication method is key-based, do the following:
     a. Enter the user name (default is "ace").
     b. Select the "Use Public Key Authentication" checkbox.

   c. A prompt will display asking for you to load your private key file.

      a. Click `Upload Private Key File` and navigate to your stored key file.

        When your key file has uploaded it will be displayed in the prompt. For example:

      b. Click `Close`. The prompt closes.

   d. Click `Log In`.

3. Click the icon for Admin\System Configuration. The icon is circled in the example to assist in finding its location.

4. Select the Network tab. If it is not already selected, select the Configuration sub-tab to display the FastX Network Configuration page.

5. Do the following:

   a. In the Secret Key field, remove the auto-generated key and enter the secret key name referenced by the current (non-upgraded) Session Servers. Record this secret key (e.g. copy

to your clipboard) because you will need it when configuring the Session servers later in this topic.

b. Enable the connection to accept data from cluster member.

c. In the box to specify the log in method, select "Sessions - log in to the system running the fewest sessions".

d. Disable the Gateway Server from sending data to cluster members.

The following image is an example of the completed FastX Network Configuration page for the Gateway Server.
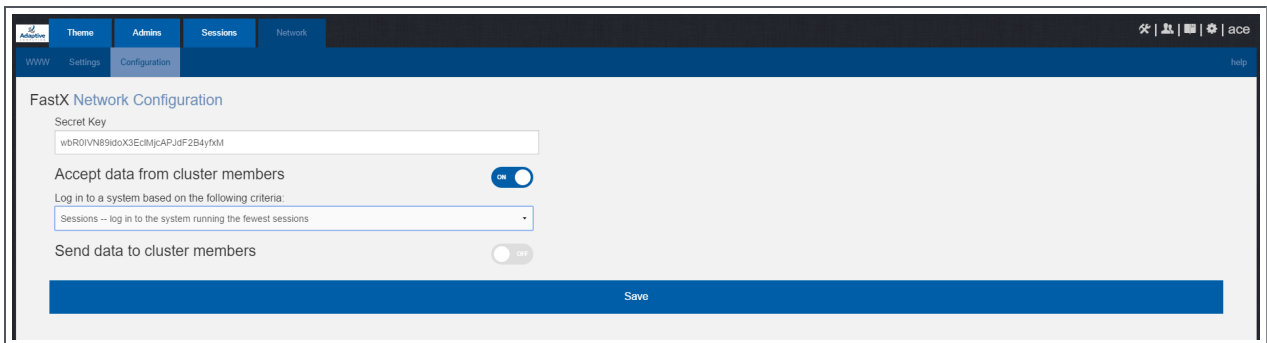
6. Click `Save` to submit your changes.

3.41.2.C Upgrade the Session Servers

> **ⓘ** These instructions assume you installed the Remote Visualization Session Servers on the same hosts on where the Torque MOM Hosts (compute nodes) were installed *and* that you have prepared those hosts for RPM upgrades.

If the version and release of the FastX rpm prior to upgrade was less than 2.2-77.3, do the following:

1. Make sure that your DNS server is configured for reverse lookups. Without reverse DNS, Session Servers will fail to register with your Gateway Server. As a result, authentication requests to the Gateway Server will fail because the Gateway Server will not be able to connect to any Session Servers.

2. On the *each* Session Server host, do the following:

a. Install or update FastX and all of its dependencies.

```
[root]# yum install ImageMagick-perl perl-Crypt-SSLeay perl-Net-SSLeay perl-X11-
Protocol StarNetFastX2
```

> **ⓘ** If installing on RHEL, some packages may not be found in the standard RHEL distribution repositories. You will need to install the missing dependencies from EPEL or other reputable repositories.
>
> ```
> [root]# rpm -Uvh http://dl.fedoraproject.org/pub/epel/6/x86_64/epel-
> release-6-8.noarch.rpm
>
> [root]# yum install yum-utils
> [root]# yum-config-manager --disable epel
> [root]# yum install --enablerepo=epel,rhel-6-server-eus-optional-rpms
> ImageMagick-perl perl-Crypt-SSLeay perl-Net-SSLeay perl-X11-Protocol
> StarNetFastX2
> ```

b. Confirm the config directory is owned by root; if it is, chown it to "fastx".

```
[root]# ls -ld /usr/lib/fastx2/config
drwxr-xr-x 2 root root 4096 Jun  6 11:11 /usr/lib/fastx2/config

[root]# chown fastx. /usr/lib/fastx2/config/ -R
```

c. Remove the existing gateway-server.json file.

```
[root]# rm /usr/lib/fastx2/config/gateway-server.json
```

d. Restart the FastX service.

```
[root]# service fastx restart
```
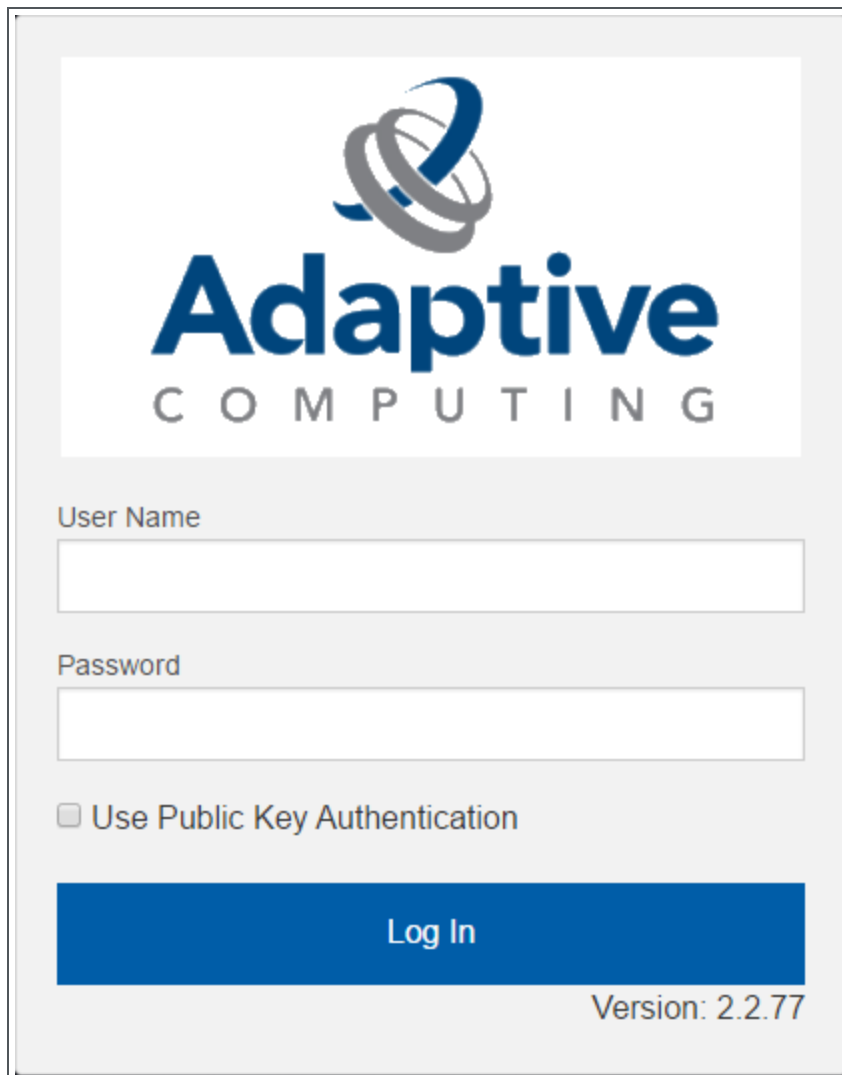
3.41.2.D Configure a Session Server

This section provides instructions on how to configure *one* Session Server (referred to as the initial Session Server). The configuration will then be copied to the additional Session Servers in your environment in a later procedure.

If the version of the FastX RPM prior to upgrade was less than 2.2, do the following:

1. Using a web browser, navigate to your *secure* Remote Visualization Session Server instance. (**https**://<session-host>:3443; where <session_host> is the IP address or name of the *initial* Remote Visualization Session Server Host).

   The Log In page displays. For example:

2. Log in as the FastX admin user. Do *one* of the following:

    -   If your authentication method is password-based, do the following:

        a. Enter the user name (default is "ace").

        b. Enter the password (default is "ace").

        c. Make sure the "Use Public Key Authentication" checkbox is cleared.
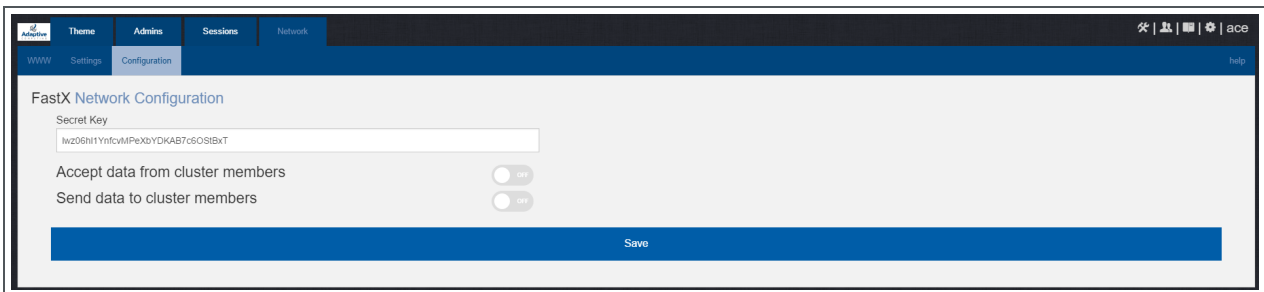
        d. Click `Log In`.

- If your authentication method is key-based, do the following:

  a. Enter the user name (default is "ace").

  b. Select the "Use Public Key Authentication" checkbox. Upload the public key used when you configured the Gateway Server earlier in this topic.

  c. Click Log In.

> 🛈 When you first log in, you will get a message that you have no session running. That is expected.

3. Select the icon for Admin\System Configuration. The icon is circled in the example to assist in finding its location.

4. Select the Network tab. If it is not already selected, select the Configuration sub-tab to display the FastX Network Configuration page.

5. Do the following:

   a. In the Secret Key field, remove the auto-generated key and enter the secret key used when configuring the Remote Visualization Gateway Server earlier in this topic.

   > 🛈 You will not be able to login to the portal on the Gateway Server until you have completed the configuration of at least one Session server. If you did not save it earlier, the secret key can be found in the /usr/lib/fastx2/config/network.json on the Gateway Server.

   b. Disable the connection to accept data from cluster members.

   c. Enable the Gateway Server to send data to cluster members.

   d. In the box to specify whether to SSL certificates, select "I am using a self-signed certificate".

e. In the Cluster member URLs area, do the following:

i. Click the + icon.

ii. In the box that displays, enter the IP address or name and the port number of the Gateway Server you just upgraded (for example: "https://mgmtnode:3443").

The following image is an example of the completed FastX Network Configuration page.
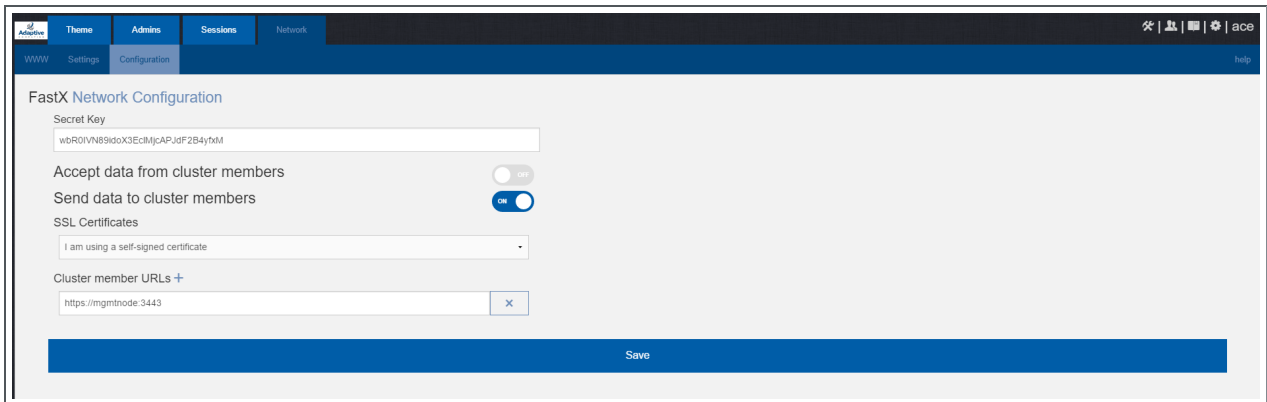
6. Click Save to submit your changes.

3.41.2.E Copy the Session Server Configuration to the Remaining Session Servers

After you configured the initial Session Server, the settings are saved in the network.json file.

> For documentation clarity, these instructions use node00 through node09 as the names of the Session Servers; with node00 designated as the initial Session Server.

If the upgrade required you to configure a session server in the prior step, on the initial Session Server Host, copy the network.json file to the remaining Session Server Hosts in your environment, and restart the FastX service.

```
[root]# for i in {01..09} ; do scp /usr/lib/fastx2/config/network.json
root@node$i:/usr/lib/fastx2/config/network.json ; done
[root]# for i in {01..09} ; do ssh node$i "chown fastx. /usr/lib/fastx2/config/. -R" ;
done
[root]# for i in {01..09} ; do ssh node$i "service fastx restart" ; done
```

3.41.3 Grant Users Remote Visualization Permissions in Viewpoint

If you upgraded Viewpoint from version 9.0.2 or prior, verify that the users who work with Remote Visualization have the appropriate role in their Viewpoint user principal.

These are the Viewpoint Roles for Remote Visualization:

- RemoteVizAdmin – Administrative user, with permission to create remote visualization application templates and manage other user's remote visualization jobs.

- RemoteVizUser – Basic user, with permission to create and manage their own remote visualization jobs.

See Creating or Editing Principals in the *Moab Viewpoint Reference Guide* for instructions on setting up principals.

3.42 Upgrading Your Nitro Integration (RPM)

This section provides instructions on upgrading your Nitro Integration as part of your Moab configuration.

3.43 Upgrading Nitro (RPM)

This topic contains instructions on how to upgrade Nitro using the RPM upgrade method.

Upgrade Nitro

On the Nitro Host, do the following:

1. If you installed Nitro on its own host *or* if Nitro is the first component being upgraded on a host with other RPM installations, complete the steps to prepare the host. See 3.28 Preparing for RPM Upgrades - page 304 for more information.

2. Back up your existing launch script in `/opt/nitro/bin/`.

3. Install the RPM.

   ```
   [root]# yum update nitro
   ```

4. Copy the provided scripts and the *nitrosub* command from the `/opt/nitro/scripts` directory.

   > ℹ This is a "copy" file operation and not a "move" operation. This allows you to customize your version and always have the factory version available for consultation and/or comparison.

   a. Copy the `launch_nitro.sh` and `launch_worker.sh` scripts for your resource manager to the `bin` directory. Each resource manager has a subdirectory with the scripts directory that contains the scripts. This example uses Torque as the resource manager.

      ```
      [root]# cp /opt/nitro/scripts/torque/launch_nitro.sh /opt/nitro/bin/
      [root]# cp /opt/nitro/scripts/torque/launch_worker.sh /opt/nitro/bin/
      ```

   b. Copy the *nitrosub* command to the `bin` directory.

      ```
      [root]# cp /opt/nitro/scripts/nitrosub /opt/nitro/bin/
      ```

   c. Copy the `nitro_job.sh` and the `worker_job.sh` scripts to the `etc` directory.

      ```
      [root]# cp /opt/nitro/scripts/nitro_job.sh /opt/nitro/etc/
      [root]# cp /opt/nitro/scripts/worker_job.sh /opt/nitro/etc/
      ```

5. Merge any customizations from your existing launch scripts, job scripts, and the *nitrosub* command (if applicable) into the new launch scripts, job scripts, and the *nitrosub* command that you copied from the scripts directory.

6. If your system configuration allows multiple coordinators on the same node, additional configuration may be needed. See 2.19 Running Multiple Coordinators on the Same Node - page 120 for more information.

7. If you are not using a shared file system, copy the updated Nitro installation directory to *all* hosts.

```
[root]# scp -r /opt/nitro root@host002:/opt
```

> **ℹ** If you are not using a shared file system, you may not be able to use the `nitrosub` client command.

Related Topics

- 3.42 Upgrading Your Nitro Integration (RPM) - page 359

3.44 Upgrading Nitro Web Services (RPM)

This topic contains instructions on how to upgrade Nitro Web Services using the RPM upgrade method.

3.44.1 Upgrade to MongoDB 3.2.x

Beginning with version 2.1.0, Nitro Web Services requires MongoDB 3.2.x.

On the Nitro Web Services MongoDB host, do the following:

1. Check your MongoDB server version.

```
[root]# rpm -qa | grep mongo
```

2. If the MongoDB server version is older than 3.2.x, then you will need to upgrade the databases. See 3.32 Upgrading to MongoDB 3.2.x (RPM) - page 311 for more information.

3.44.2 Back up the MongoDB Databases

> ⓘ This procedure is only applicable if you did not have to upgrade the MongoDB databases earlier in this topic. See 3.44.1 Upgrade to MongoDB 3.2.x - page 362 for more information.

On the Nitro Web Services MongoDB host, do the following:

1. Stop all services that are using the MongoDB databases.

2. Back up the MongoDB databases.

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

3. Restart the services.

3.44.3 Upgrade Nitro Web Services

On the Nitro Web Services Host, do the following:

1. If you installed Nitro Web Services on its own host *or* if Nitro Web Services is the first component being upgraded on a host with other RPM installations, complete the steps to prepare the host. See 3.28 Preparing for RPM Upgrades - page 304 for more information.

2. Stop the services.

```
[root]# service nitro-web-services stop
[root]# service nitro-zmq-job-status-adapter stop
```

3. Install the upgrade.

```
[root]# yum update nitro-web-services
```

4. If you are upgrading Nitro Web Services from 2.0.0, re-enable the services.

```
[root]# chkconfig nitro-web-services on
[root]# chkconfig nitro-zmq-job-status-adapter on
```

5. If you have customized your configuration files, the RPM upgrade will have copied the new configuration files into the `/opt/nitro-web-services/etc` directory with a .rpmnew extension. Merge any parameter changes in the .rpmnew files into the respective configuration files.

> ⓘ See the steps in 3.17.3 Install and Configure Nitro Web Services - page 281 for more information on the configuration files.

6. Restart the services.

```
[root]# service nitro-web-services start
[root]# service nitro-zmq-job-status-adapter start
```

3.44.4 Grant Users Nitro Permissions in Viewpoint

Verify that the users who work with Nitro Web Services have the appropriate role in their Viewpoint user principal.

These are the Viewpoint roles for Nitro:

- NitroAdmin – Administrative user, with permission to create Nitro application templates and manage other user's Nitro jobs.

- NitroUser – Basic user, with permission to create and manage their own Nitro jobs.

See "Creating or Editing Principals" in the *Moab Viewpoint Reference Guide* for instructions on setting up principals.

Related Topics

- 3.42 Upgrading Your Nitro Integration (RPM) - page 359

3.45 Disabling the Adaptive Repository after Upgrades

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

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

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

3.46 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.

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

To Migrate the MAM Database

This procedure was successfully tested on an actual customer MySQL database with millions of transactions 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** - 2.6 Installing Moab Web Services - page 75

   - **RPM Install** - 3.10 Installing Moab Web Services - page 198

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: Automated Installation Method

This chapter contains an introduction to the Automated Installer and explains how to use it to install your Moab HPC Suite components for Red Hat 6-based systems.

ⓘ The Automated Installer does not replace the current Manual Installation method and the RPM Installation methods (typical and offline); it only provides another, simpler, option to install your Moab HPC Suite components.

In this chapter:

4.1 About the Automated Installer

The Adaptive Computing Automated Installer is developed to provide an easier installation method when installing your Moab HPC Suite components. You can easily set up a production or staging system in less than an hour and with little user-interaction.

The Automated Installer uses a system management tool named Ansible. Ansible can communicate across head nodes and compute nodes to install and configure products. Using Ansible, you can start the Moab HPC Suite installation on your first head node (Moab Server Host) or on a separate deployment workstation or server.

After the initial launch of the Automated Installer, you or another approved user can access the user interface (web GUI) to specify the remaining data needed for installation. This data includes the names of the hosts in your environment, which Moab HPC Suite components you want to install, and all the usernames, passwords, and license files.

Based on the input provided through the user interface, the Automated Installer generates an inventory file and a variables file. Ansible then references these files and uses `ssh` to communicate with your Moab HPC Suite Hosts to install products and set up your environment.

> ⓘ The Automated Installer will install the Adaptive Computing products before requesting the license information.

4.2 Requirements and Prerequisites

This topic provides the requirements, prerequisites, and other useful information before using the Automated Installer.

> ⚠ This topic is for the Automated Installer and provided user interface only; the requirements for each of the hosts on which the Moab HPC Suite components will reside are available at: 1.3 Component Requirements - page 18.

4.2.1 Managing Licenses

The Moab HPC Suite uses Reprise License Manager (RLM) to manage licenses for suite components. See 1.4 RLM Licensing Model - page 32 for more information about RLM licensing.

4.2.2 Environment Setup

The Automated Installer itself can be installed on a deployment system, or the main head node. If using a deployment system, the host (physical machine) must have the same OS as the head nodes and compute nodes in your Moab HPC Suite environment. This could be a user's desktop or a head node or a compute node in your environment. A separate deployment server is recommended because once the Automated Installer has completed, Ansible is no longer needed.

For your Moab HPC environment, the Automated Installer will ask for the count of head nodes. See 1.2 Server Hardware Requirements - page 7 for more information on environment configurations.

Once the head node count is specified, the Automated Installer will ask for their hostnames and display the distribution of products across those systems accordingly. You will then need to input the compute node and job submission node information.

- Compute nodes: This can be a up to tens of thousands of systems. The Automated Installer will prompt for these host names, and alpha or numeric range expressions can be used to easily name any number of hosts. For staging or testing purposes only, a compute node can be shared with a head node. The installer calls this an "All on one node" deployment.

- Job submission nodes: These nodes can be anywhere in the cluster. For Moab and Torque, these are the client commands so that users can submit jobs from these hosts.

> ⚠ If using a separate deployment host, that host must have access to all the head nodes, compute nodes, and job submission nodes.

4.2.3 Internet Accessibility

The Automated Installer leverages the Moab HPC Suite RPMs to install your Moab HPC Suite components; therefore, the Automated Installer expects all the hosts in your environment need to have access to the internet to download the RPMs. If using the Automated Installer on a separate deployment host, that host must also have internet access.

Supported Browsers:

- Chrome (recommended)

- Edge (recommended over Internet Explorer)

- Firefox

- Internet Explorer

- Safari

An offline installation method is available for configurations without internet access. See for more information.

4.2.4 Supported Operating Systems

The Automated Installer, and the corresponding Moab HPC Suite components, can be installed on any of these operating systems:

- CentOS 6, 7; tested on 6.8 and 7.2

- RHEL 6, 7; tested on 6.8 and 7.2

- SUSE Linux Enterprise Server 12, 12 SP1

> ⚠ The Automated Installer method does *not* support running on multiple OSs. This is true if using a deployment sever, and also within the Moab HPC Suite environment.

4.2.5 Users

This section explains the different user types and permissions when using the Automated Installer.

4.2.5.A Automated Installer User

It is recommended that you run the Automated Installer as the root user. It may be run as a non-root user, but that user will need to have passwordless sudo configured on all head nodes and compute nodes so that all the necessary packages can be installed and system changes can be made.

This user also needs to have ssh key authentication setup between the deployment host and all the hosts in your Moab HPC Suite environment. Refer to the documentation for your operating system for more information on setting up passwordless sudo.

4.2.5.B Environment (Cluster) Users

It is highly recommended that you use a scalable user management system such as LDAP or NIS to synchronize your users between all your hosts. It is expected that if you are using one of these user management systems that it is completely setup before running the Automated Installer. The Automated Installer will prompt you for information about connecting the Moab HPC Suite components to these systems.

4.2.5.C Test User

The Automated Installer also requires a test user. This must be a non-root user that exists on the head nodes and compute nodes. This user will be used to submit a test job to ensure Moab and Torque are configured correctly. A test user, "testuser<Unix timestamp>", will be created with a home directory rooted in the shared file system. The test user and its home directory will be removed as part of a clean-up task on the "Finished" step.

4.2.6 SSH Keys

As mentioned earlier, the Automated Installer requires ssh key authentication between the deployment host and all the hosts in your Moab HPC Suite environment.

Tools like "ssh-copy-id" can be used to easily setup these keys.

After you run ./automated-installer.sh webui, but *before* you access the user interface, modify the `./automated-installer/ansible.cfg` file to turn on ssh host-key checking.

4.2.7 SSL

By default, the Automated Installer does *not* use SSL (https). If you want to enable SSL, you can either have the Automated Installer generate a self-signed certificate *or* use your own ssl_certfile

and ssl_keyfile.

See the "REST API web server SSL" section in the `./automated-installer-` `/webui/etc/installer.cfg` file for instructions.

4.2.8 DNS

If you do not have a DNS set up in your environment, a helper-playbook is available. This helper-playbook is *not* intended for production use, but may be useful for staging environments.

The helper-playbook uses hostname entries in /etc/hosts and dnsmasq to emulate an actual DNS server. For the helper playbook to work, you will need to fulfill these prerequisites:

1. Have entries in the /etc/hosts file on your deployment system for each host in the cluster.

    When setting up the hosts file, use this format: <IP address> <Fully Qualified Domain Name> <Short Name>

    For example: 10.0.0.2 headnode.example.com headnode

    Once this has been done, run the Automated Installer to set up the host files on the other servers as well as dnsmasq.

2. Set up the DNS.

    a. Run and use the user interface to populate your site config files; stopping when you get to the Summary page.

    b. Return to the deployment server where you launched the Automated Installer and press **Ctrl-C** to kill the user interface.

    c. Run the following:

        ```
        ./automated-installer.sh playbook helper-playbooks/dns-setup.yml
        ```

    d. Once this completes successfully, rerun the ./automated-installer.sh webui command and finish the installation.

4.2.9 Shared File System

Having a shared file system is required when using the Automated Installer. This requirement supports Torque, Nitro, and Viewpoint's File Manager feature for storing each job's output and log files. A shared file system is also needed for users to access their home directory from the Moab Viewpoint portal.

The Automated Installer provides an option to set up NFS for you automatically. If you are already using a shared file system, leave this checkbox unselected.

4.2.10 Software Repositories

As part of the Moab HPC Suite RPM process, some software repositories may be added or enabled to be able to install all necessary dependencies. However, some OSs require subscriptions in order to access the dependencies.

- CentOS 6, 7 – A subscription is not required.

- RHEL 6, 7 – You must be registered for a Red Hat subscription.

- SLES 12, 12 SP1 – You must be registered for a SUSE Linux Enterprise subscription.

4.2.11 Firewalls

If your site is running firewall software on its hosts, you will need to configure the firewall to allow connections to the Moab HPC Suite products in your installation. See 3.20 Opening Ports in a Firewall - page 289 for general instructions and an example of how to open ports in the firewall.

See 5.3 Port Reference - page 411 for the port numbers used by the various products in the Moab HPC Suite.

4.3 Using the Automated Installer

This topic contains instructions on how to configure and execute the Automated Installer to install your Moab HPC Suite components.

4.3.1 Before You Begin

Before using the Automated Installer, you must plan your topology and meet the requirements and prerequisites. See 4.2 Requirements and Prerequisites - page 369 for more information.

> 🛈 The Automated Installer does not currently support diskless installation.

> 🛈 If you are installing Moab Passthrough for use with Slurm:
>
> - Slurm must be installed before launching the Automated Installer.
> - The Slurm client commands must be installed on the host where you will be installing Moab Passthrough (Moab Head Node).
>
>   You should also test the Slurm client commands to verify that you can submit jobs from the Moab Head Node host.

4.3.2 Obtain and Launch the Automated Installer

On the host you have chosen to be your deployment host (this can be the same host as your first head node, or a stand-alone host), do the following:

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

   ```
   export http_proxy=http://<proxy_server_id>:<port>
   export https_proxy=http://<proxy_server_id>:lt;port>
   ```

2. Update your system software to the latest version.

   ```
   [root]# yum update
   ```

3. Ensure hostname resolution for all hosts.

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

4. Download the latest Moab HPC Suite RPM bundle from the Adaptive Computing Moab HPC Suite Download Center (https://www.adaptivecomputing.com/support/download-center/moab-hpc-suite-download/).

5. Untar the RPM bundle.

```
[root]# tar zxf moab-hpc-suite-9.1.3-<OS>.tar.gz
```

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

6. Change directories into the untarred directory.

```
[root]# cd moab-hpc-suite-9.1.3-<OS>
```

7. Change directories into the automated-installer directory.

```
[root]# cd automated-installer
```

8. Launch the Automated Installer.

```
[root]# ./automated-installer.sh webui
```

Once the Automated Installer has loaded the necessary files and packages, you will get a message that indicates that the user interface is available and provides the host and the port information for the user interface.

```
################################################################################
################################################################################
####
####    The Moab Automated Installer user interface is now available at:
####
####    http://127.0.0.1:7443
####
####    Leave this shell running until you are instructed to exit this process.
####    Ctrl-C will signal this process to terminate.
####
################################################################################
################################################################################
```

> ⚠ The Automated Installer must be active on the deployment host until specified later in this topic. If you terminate the Automated Installer process before you have completed user interface tasks, you will have to relaunch the Automated Installer process. You can then re-access the user interface and finish the installation.

## 4.3.3 Access and Use the User Interface

This procedure requires a web browser that can access the same network where the Automated Installer deployment server runs

> ⓘ The user interface is built with tooltips to aid you in the installation process. Hover the mouse over a field name to view additional information about the field.
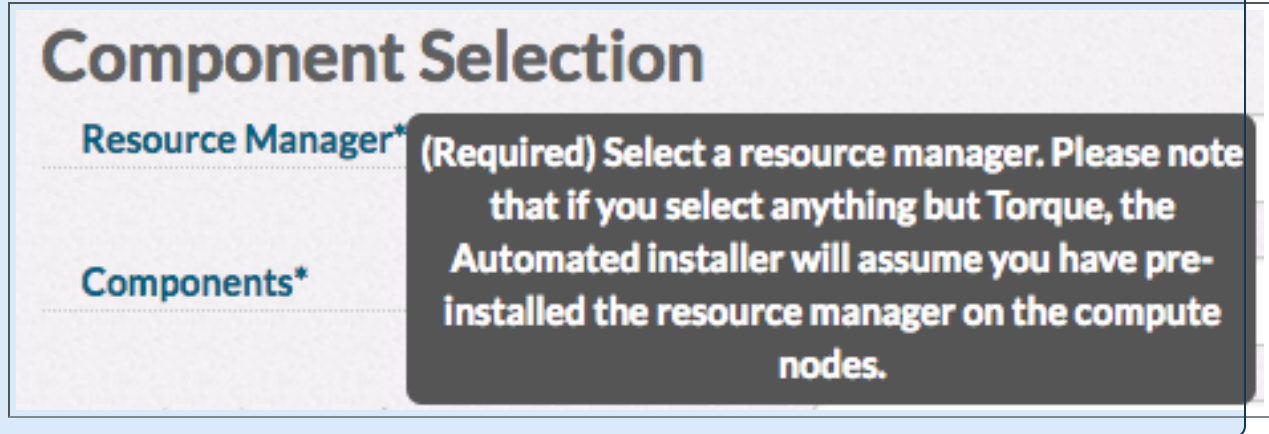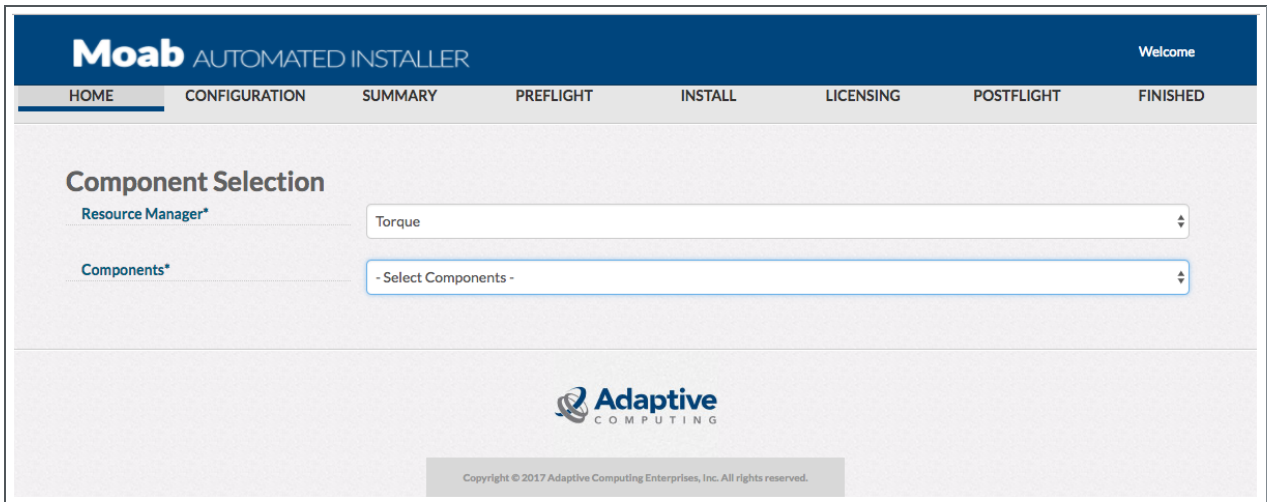

1. Using a web browser, navigate to the user interface. This is the host and port information obtained when you launched the Automated Installer.

   The Home Page displays with the `Component Selection` drop-down. For example:
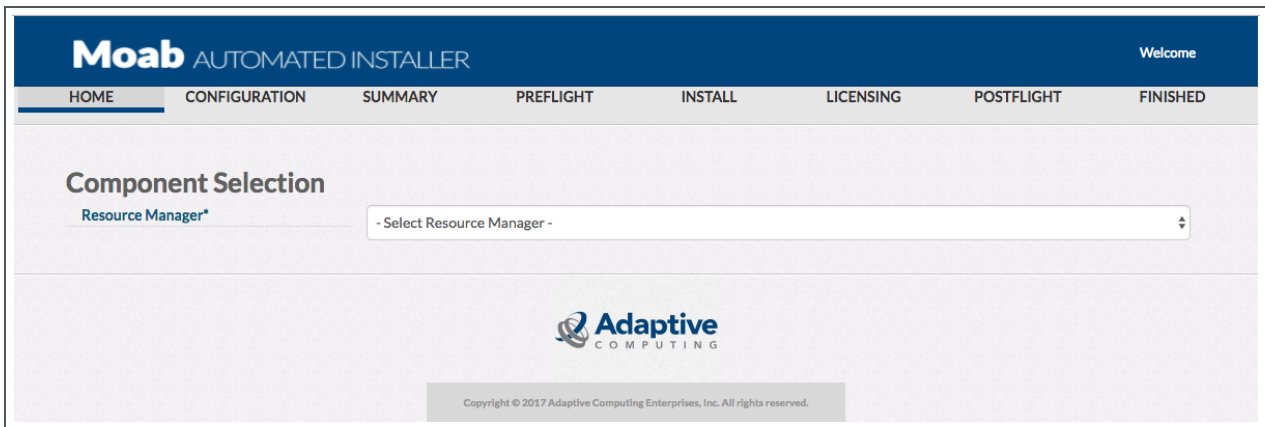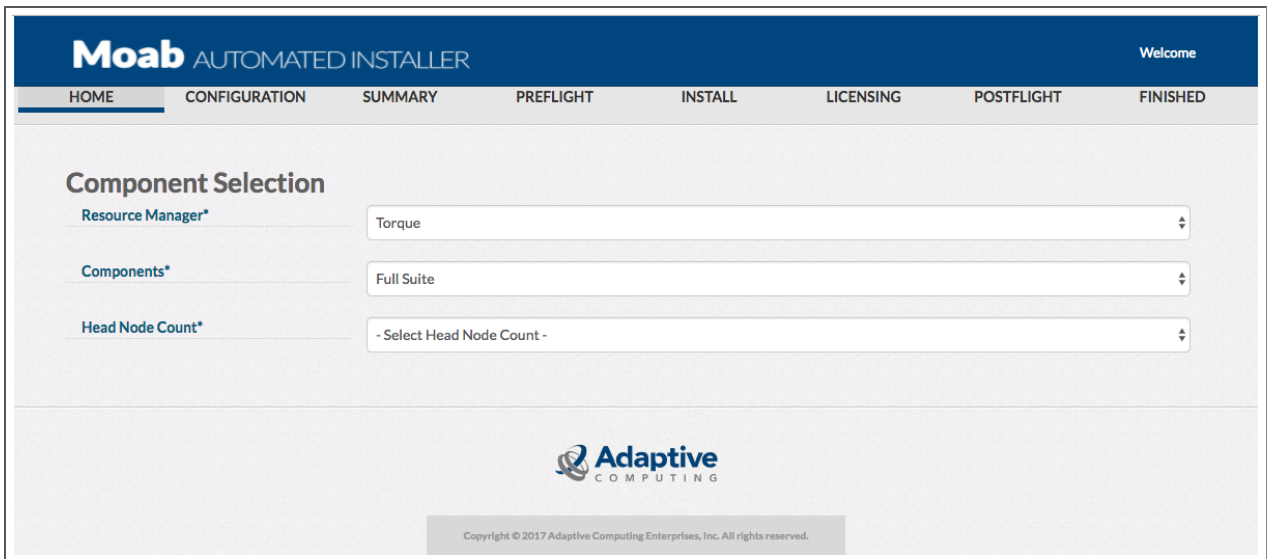
2. Select a resource manager.

3. Specify the Moab HPC Component bundle you wish to install.

4. A `Head Node Count` drop-down appears below the `Components` drop-down asking for the number of head nodes. Select the number of head nodes you wish to use.

5. Once you have specified the Resource Manager, Components, and Head Node Count, the page displays the layout of your selected configuration. For example:
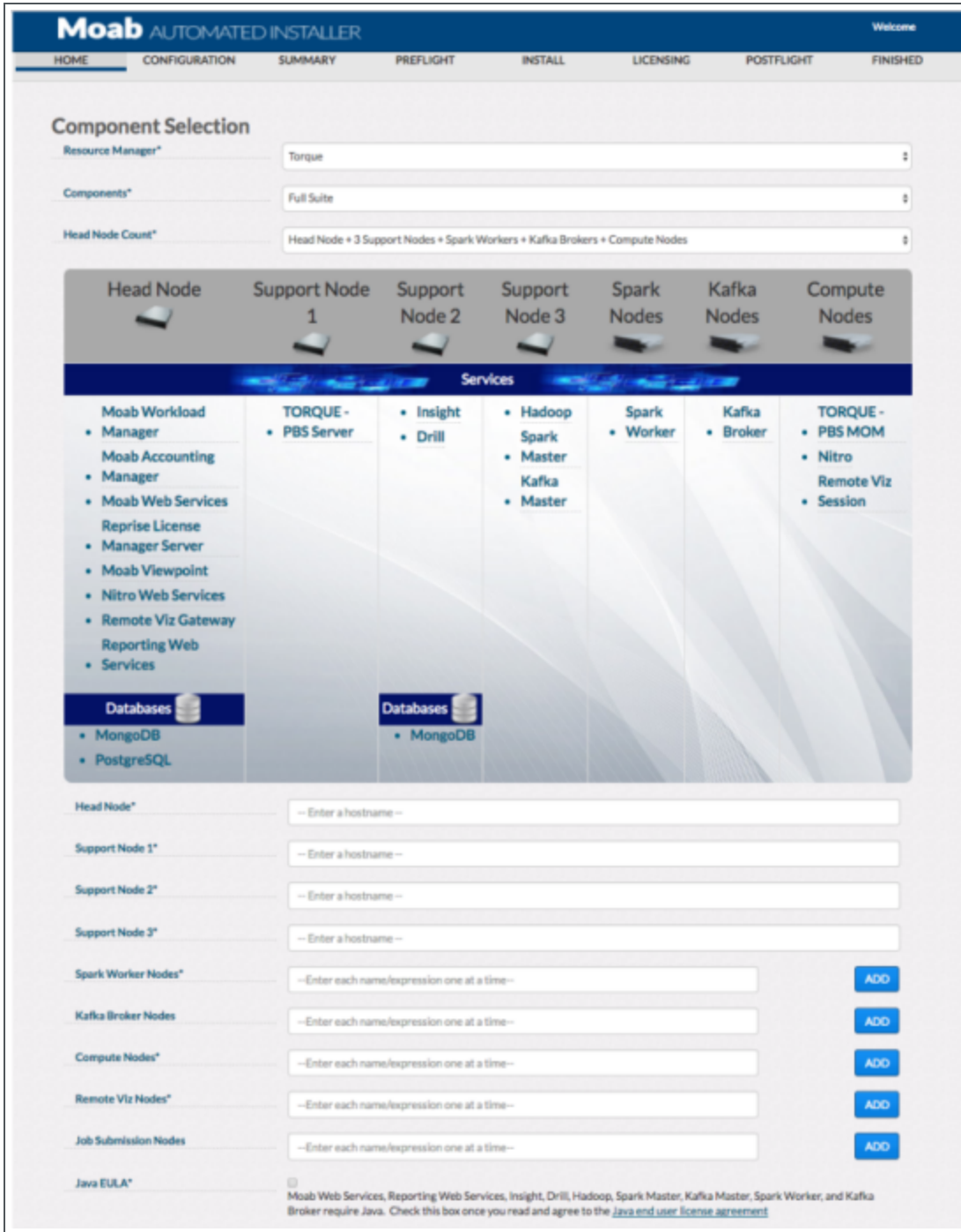
Enter the names for the different nodes in your configuration.

- In the box for the Head Node, enter the Fully Qualified Domain Name (FQDN).

- If you have specified more then one head node, enter the FQDN for each of the additional head nodes (support nodes).

- In the `Compute Nodes`, `Remote Viz Nodes`, `Job Submission Nodes`, `Spark Worker Nodes`, and `Kafka Worker Nodes` sections, enter FQDNs for each of the nodes and click `ADD`. Ansible range expressions are supported. Repeated as needed.

> ℹ️ If you selected Slurm as your resource manager, `Compute Nodes` are the nodes where `slurmd` is running.

6. If your configuration includes products that require Java, information about the Java EULA displays. Select the check box to accept the license agreement.

7. When finished, click `Configure`.

   The Configuration Page prompts for the information needed to install and set up the components selected on the previous page. For example:
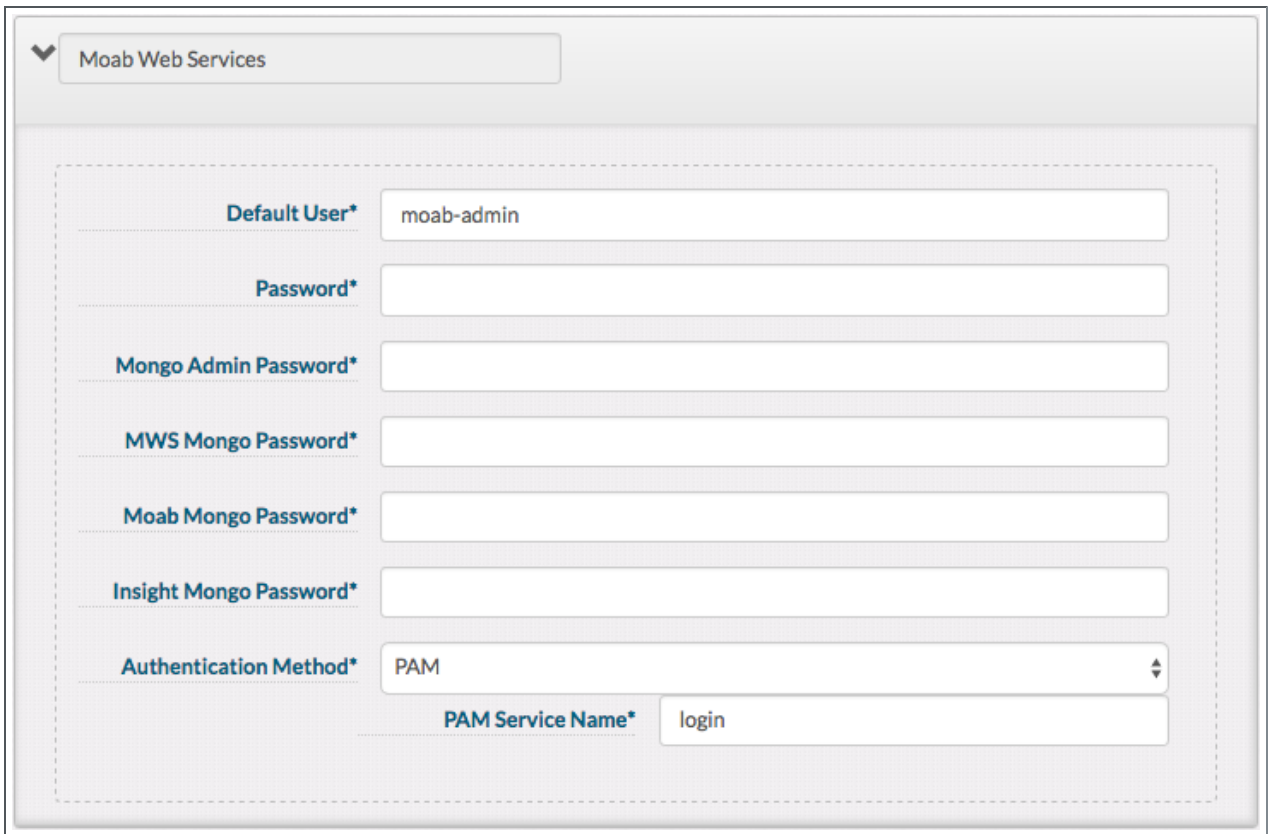
> ℹ️ Some fields are automatically populated with default values, or with information gathered at runtime. Some fields are optional. Required fields are marked with an asterisk.
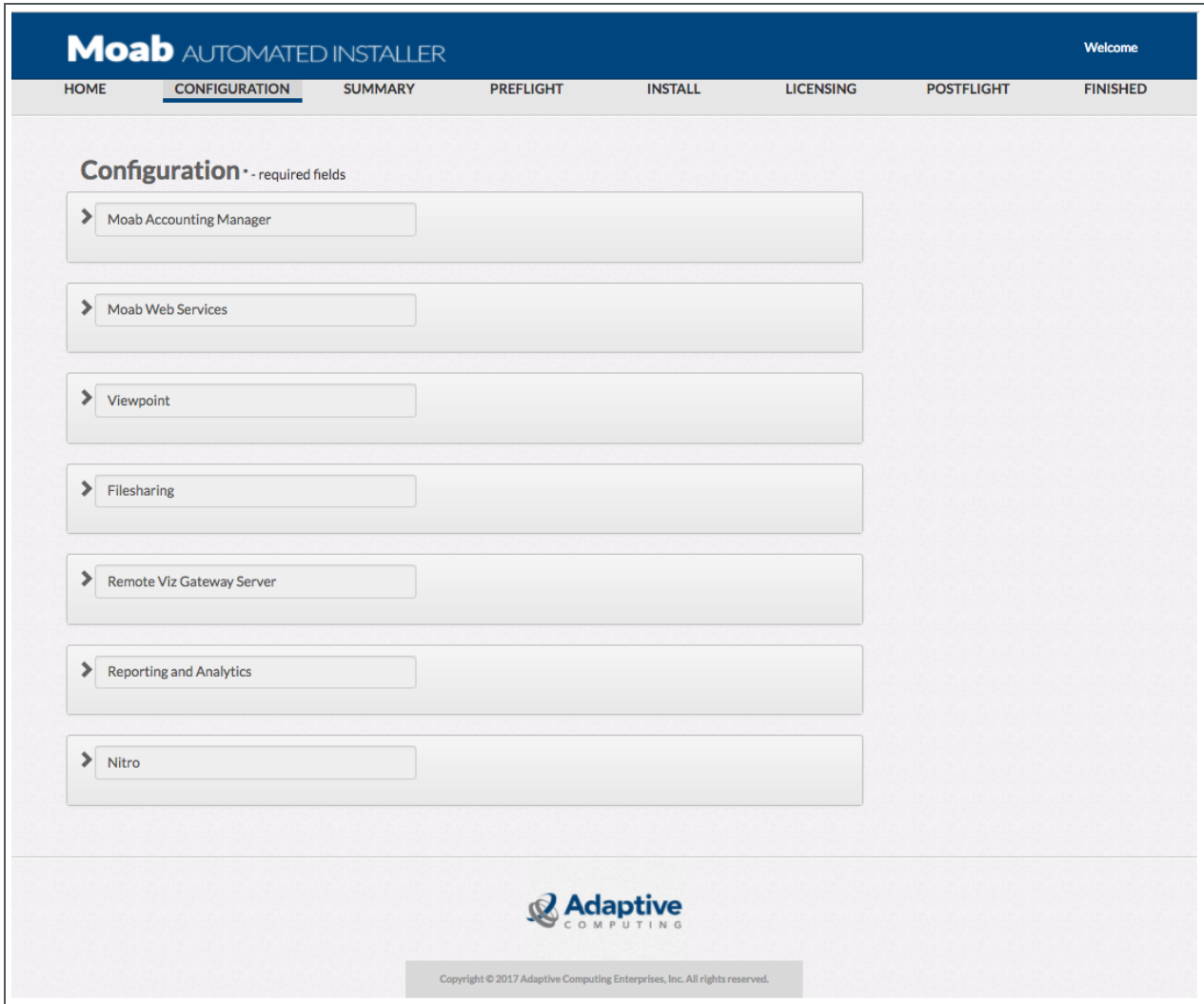
8. If Moab Accounting Manager is part of your configuration, the Automated Installer displays a section for entering Moab Accounting Manager configuration information. Enter the required information. Use the tooltips for more information. The following is an example of this section.

9. If Moab Web Services is part of your configuration, the Automated Installer displays a section for entering Moab Web Services configuration information. Enter the required information. Use the tooltips for more information. The following is an example of this section.

10. If Moab Viewpoint is part of your configuration, the Automated Installer displays a section for entering Viewpoint configuration information. Enter the required information. Use the tooltips for more information. The following is an example of this section.

11. Enter the required file sharing information in the `Filesharing` section. Check `Set up filesharing for me` if you want the Automated Installer to set up an NFS file share on all relevant nodes. The following is an example of this section.

12. If Remote Visualization is part of your configuration, the Automated Installer displays a section for entering Remote Visualization configuration information. Enter the required information. Use the tooltips for more information. The following is an example of this section.

13. If Torque is part of your configuration, the Automated Installer displays a section for entering Torque configuration information. Enter the required information. Use the tooltips for more information. The following is an example of this section.

14. If Nitro is part of your configuration, the Automated Installer displays a section for entering Nitro configuration information. Enter the required information. Use the tooltips for more information. The following is an example of this section.

15. If Reporting and Analytics is part of your configuration, the Automated Installer displays a section for entering Reporting and Analytics configuration information. Enter the required information. Use the tooltips for more information. The following is an example of this section.

16. When finished, click NEXT.

    The Summary Page displays the configuration information you just entered. For example:

Moab AUTOMATED INSTALLER   Welcome

HOME   CONFIGURATION   SUMMARY   PREFLIGHT   INSTALL   LICENSING   POSTFLIGHT   FINISHED

Resource Manager
Torque

Components
Full Suite

Inventory
Hostnames

Head + Spark + Kafka + Compute Node*   cent7screenshots.vapor2.ac

Variables
Moab Workload Manager

Moab Accounting Manager

Shared Secret   wiPCMzK33AvDmwhkYlsbJ9nbjsd64CAB

PostgreSQL Password*   changeme!

GUI Password*   changeme!

Moab Web Services

Default User*   moab-admin

Shared Secret   C92BoiInAK+6mJUT8Fhnb9fJUv8yP0y1

Moab Mongo Password*   secret3

PAM Service Name   login

Directory Type*   OpenLDAP Using InetOrgPerson Schema

Base DNs*

dc=testldap,dc=ac

Insight Mongo Password*   secret4

Server Hostname*   openldap.ac

Security Type*   None

Mongo Admin Password*   secret1

Bind Password*   secret5

Bind User DN*   cn=admin,dc=testldap,dc=ac

Server Port*   389

Password*   changeme!

MWS Mongo Password*   secret2

Message Queue AES Key   v1is67A9gBaspAGUoyNO7g==

Authentication Method*   LDAP

Viewpoint

Admin Password*   changeme!

Principals*

Principal Name	Principal Description	Rights	Object Type	Object Name
principal1	Admin	HPC Admin Nitro Admin Remote Viz Admin Reporting Admin	LDAP User	principal1
principal2	User	HPC User Nitro User Remote Viz User Reporting Consumer	LDAP User	principal2

Shared Secret   jiicw2ld63CDOUsqVj7LZA45gQnbqj19G

PostgreSQL Password*   changeme!

Filesharing

Fileshare Path*   /home

Test User*   testuser1489774050

Set up filesharing for me   Yes

NFS Network Mask*   10.2.184.0/23

NFS Options*   (rw,sync,no_root_squash,no_subtree_check)

NFS Mount Point*   /home

NFS Share*   /home

Remote Viz Gateway Server

Install Optional Graphical packages   Yes

Shared Secret   K7DRbn9sqU3HNcC7PScibxBoQtbajHmU

Reporting and Analytics

Drill Admin Password*   changeme!

Drill User Password*   changeme!

17. Confirm the information is correct for your system and then click NEXT. Note that auto-generated variables are also displayed. Displaying all variables, including auto-generated variables aids in solving problems that may occur during installation.

The Pre-Flight Checks Page displays. For example:

18. The pre-flight checks confirm your system is in order before installing the Moab HPC Suite components.

   a. To begin the checks, confirm whether you have satisfied the firewall check and then select the check box in the Status column. See 4.2.11 Firewalls - page 373 for a list of port numbers and other information.

   b. The pre-flight checks will then launch in order.

    c. If an error occurs:

- The error information will display in the description for the check.

- A retry link will display.

- For the hostname check, a `Try Fix` link displays. This link will attempt to take the information you've already given, and try to fix the hostname configuration on your systems.

If error(s) still occur, click the `BACK` button to return to the Configuration Page and adjust your settings. You may need to open another terminal on the deployment host to try to manually resolve the issues. See also Chapter 5: Troubleshooting - page 403 for information on common issues.

19. Once all the conflicts are resolved, click `NEXT`.

20. When the Install Page displays, click `INSTALL` to begin the Moab HPC Suite installation process.
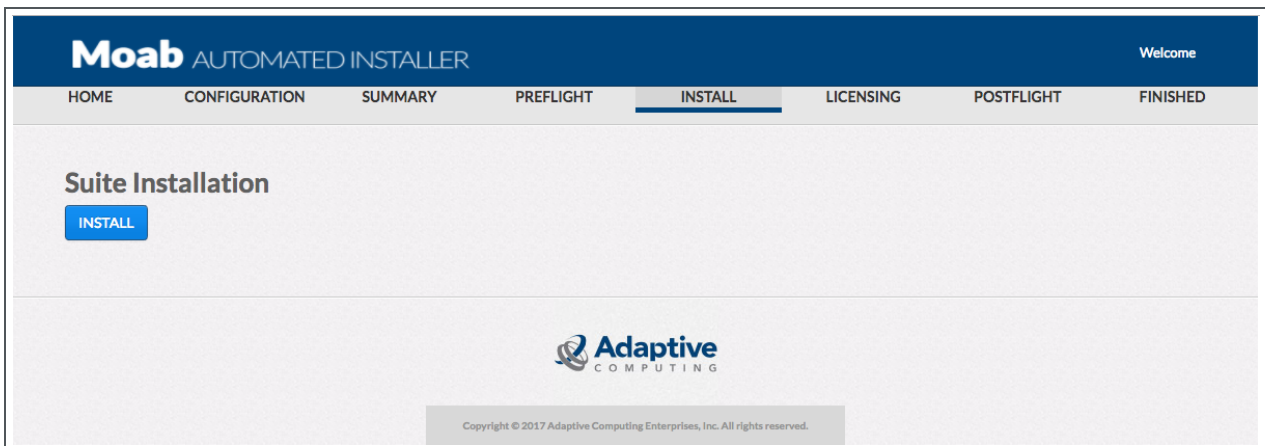
> **ⓘ** For CentOS 6 systems, if the install reports the error "Failed to set permissions on the temporary file Ansible needs to create when becoming an unpriviledged user.", then edit the `./automated-installer/ansible.cfg` file and uncomment the "allow_world_readable_tmpfiles=True" line. Rerun the install.

21. When the installation has finished, click `NEXT`.

The Licensing Page displays. For example:

Moab AUTOMATED INSTALLER

Welcome

| HOME | CONFIGURATION | SUMMARY | PREFLIGHT | INSTALL | LICENSING | POSTFLIGHT | FINISHED |

Request Licenses

To obtain licenses, send the following required(*) information to licenses@adaptivecomputing.com.

✔ **RLM Server Host ID***

The RLM Server Host ID is required for license creation.

FA163ED80C5B

↻ Rerun

Total number of physical sockets, GPUs, and Xeon Phis in your cluster.*
Required number of concurrent Remote Viz sessions.*
RLM Server Hostname* cc7h.vapor2.ac
Required number of concurrent Nitro Workers or Cores.*

Notes

When the Elastic Computing license is enabled, the following are supported:
- Dynamic Nodes (SCHEDCFG[] FLAGS=enableDynamicNodes)
- Deleting nodes from the node table
- NodeIdlePurgeTime parameter
- QOSCFG[] RequestGeometry Parameter

The following are also enabled:
- Multiple resource managers
- Triggers
- Job templates

You must configure Elastic Computing manually in Moab. The Automated Installer does not currently assist you in doing so. See "Elastic Computing" or "Elastic Computing Overview" in the documentation for details.

Apply Licenses

⌄ Moab Workload Manager

License File
Drop your License File here   **UPLOAD** 🗑

Elastic License File
Drop your Elastic License File here   **UPLOAD** 🗑

Elastic Tracking License File
Drop your Elastic Tracking License File here   **UPLOAD** 🗑

⌄ Viewpoint

License File
Drop your License File here   **UPLOAD** 🗑

⌄ Remote Viz Gateway Server

Either upload a license file, or enter a license key + maximum concurrent sessions and we will generate the license for you. Do not enter both.

License File*
Drop your License File here   **UPLOAD** 🗑

-- OR --

License Key

Maximum Concurrent Sessions

⌄ Nitro

License File*   Drop your License File here   **UPLOAD** 🗑

> **ⓘ** Moab and all components are installed; however, they are unlicensed and running in evaluation mode.
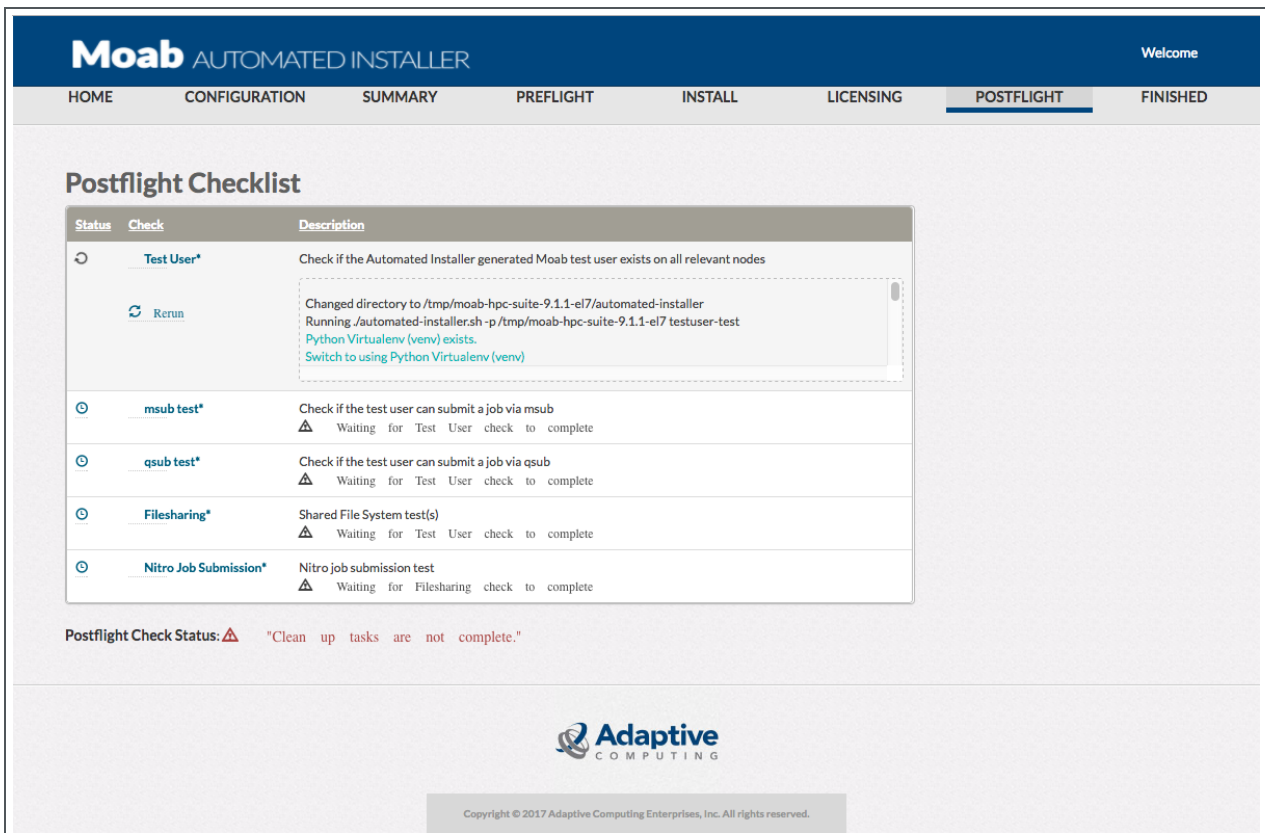
22. You can upload existing licenses, or contact licenses@adaptivecomputing.com for a new license. You will need to provide Adaptive Computing with the following information (if displayed) from the Licensing step:

    - RLM Server Host ID

    - RLM Server Hostname

    - Total number of sockets, GPUs, and Xeon Phis in your cluster

    - Required number of concurrent Remote Viz sessions

    - Required number of concurrent Nitro Workers or Cores.

    You may come back to this page to apply licenses at a later time.

23. When all the licenses are accepted, click NEXT.

24. The postflight checks are relatively quick tests that run after installation to verify the installation was successful. Postflight checks run automatically when the Postflight step loads. Each test indicates what other test(s) it is waiting on before running.

> **ⓘ** If you are installing Moab Passthrough, the Postflight Checklist will appear as shown below.

If an error occurs:

- The error information will display in the description for the check.
- A retry link will display.

If error(s) still occur, you may need to open another terminal on the deployment host to try to manually resolve the issues. See Chapter 5: Troubleshooting - page 403 for information on troubleshooting common issues.

25. Once all postflight checks pass, click `NEXT`.

26. You are now finished with the installation. One or more clean-up tasks will run during the Finished step. One or more additional links may be displayed, depending on the components you have installed. For example, `Documentation`, `Viewpoint Login`, etc.

27. Return to the host on which the Automated Installer is running, and press Ctrl-C or close the terminal.

4.4 Finishing the Installation

This topic contains additional requirements needed to finish/configure your Moab HPC Suite install-ation.

> In this topic:
>

4.4.1 Set Up Paths

Do the following:

1. On the Torque Server Host, run the following command to add the Torque binaries to the system path.

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

2. On the Moab Server Host, run the following command to add the Moab binaries to the system path.

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

> ⓘ It is recommended that you add these commands to your .bashrc so that they are auto-matically sourced at log in.

4.4.2 MWS with PAM

If you configured MWS to authenticate via PAM using local files or NIS, you need to run Tomcat as root.

> ⚠ This configuration is highly discouraged and is not supported by Adaptive Computing. The recommended approach is to configure PAM and NSS to authenticate against LDAP.

4.4.3 RLM Server

If an RLM Server is part of your configuration (for example, for Moab's Elastic Computing feature, Viewpoint's Remote Visualization feature, or if using Nitro), additional configuration is needed.

The Automated Installer uses the default password when installing the RLM Server. You *must* change the default password. See Change the Default Passwords for more information.

As the RLM Server can run multiple licenses, it is recommended that you install *one* RLM Server for your configuration. However, if your configuration requires more than one RLM Server, you will *need* to configure the Adaptive Computing products to connect to a specific RLM Server. See 2.18 Using Multiple RLM Servers - page 119 for more information.

4.5 Adding Features With the Automated Installer

The current implementation of the Automated Installer supports fresh installs of the Moab HPC Suite with a limited number of configuration options. A future implementation of the Installer may include additional configuration options and/or be designed to upgrade an existing installation. However, it is possible to use the current Installer implementation to add features to an existing installation. This topic provides an example of how to use the Automated Installer to add Reporting to an existing Moab 9.1.1/Torque 6.1.1 installation.

To add Reporting to an existing installation, do the following:

1. Provision/set up additional Head/Support Nodes in your cluster, based on what features you will be adding. For example, Reporting recommends one or more Spark and Kafka nodes. It also recommends a separate Insight node, if you don't already have Insight installed. See Chapter 1: Planning Your Installation - page 3 for details.

2. Set up SSH public/private keys (PKI) so that root or a passwordless-sudoer can ssh between all nodes.

3. Configure domain name resolution so that each host can resolve every other host using a DNS name and FQDN. If a DNS server is not in use, the `/etc/hosts` files on each node must be synchronized across all nodes. The following two files (`ssh-and-dns-setup.sh` and `etc.hosts`) have been provided as one solution to help with setting up PKI and a synchronized `/etc/hosts` file across all nodes in your cluster.

ssh-and-dns-setup.sh

Before running this script, you must:

a. Update the `hostnames` and `ips` arrays in the script to include a complete list of nodes in your cluster. The indices in each array are correlated. For example, in the sample script below, the `moab` node has an IP address of `10.2.185.1`.

b. The first host listed in the `hostnames` array is assumed to be the "deployment node" (typically the Moab Head Node.) Log in to deployment node (typically the Moab Head Node), `sudo su` - to root and run `ssh-keygen`.

c. Search and replace `ace` in the script with `root` or the name of the user given passwordless-sudoer rights on all nodes.

d. Search and replace `vapor2.ac` in the script with your domain name (i.e., if your domain name is `example.com`, replace `vapor2.ac` with `example.com`).

```
#!/bin/bash -e

echo "Define hostnames and ips..."
# Use arrays with correlated indicies.
hostnames=(
        moab
        torque-server
        compute1
        compute2
)
ips=(
        10.2.185.1  #moab
        10.2.185.2  #torque-server
        10.2.185.3  #compute1
        10.2.185.4  #compute2
)

echo "Create the installer_hosts file..."
# Create a file containing "<ip> <hostname>.vapor2.ac <hostname>" to be used to
update /etc/hosts on all nodes
rm -f installer_hosts
for i in "${!hostnames[@]}"
do
        :
        echo "${ips[$i]} ${hostnames[$i]}.vapor2.ac ${hostnames[$i]}" >> ./installer_
hosts
done
echo "--------------Begin installer_hosts--------------"
cat ./installer_hosts
echo "--------------End installer_hosts--------------"
chmod +r ./installer_hosts

echo "Create setup-deployment-node script to be uploaded to hostname ${hostnames
[0]}..."
# Write setup-deployment-node script
cat <<EOF >./setup-deployment-node
echo "Removing the contents of /root/.ssh/..."
rm -rf /root/.ssh/*

echo -e  'y\n'|ssh-keygen -q -t rsa -N "" -f /root/.ssh/id_rsa || true

cat /root/.ssh/id_rsa.pub >> /root/.ssh/authorized_keys

# Process entries from the uploaded installer_hosts file
filename=/home/ace/installer_hosts
echo "filename=\${filename}"
echo "Processing the following entries from \${filename}..."
cat \${filename}

echo "Remove current /root/.ssh/known_hosts file..."
rm -f /root/.ssh/known_hosts
while read hostent; do
        echo "Get host from \$hostent (2nd field)..."
        ip=\$(echo \$hostent | cut -d" " -f1)
        echo "ip=\${ip}"
```

```
        host=\$(echo \$hostent | cut -d" " -f2)
        echo "host=\${host}"
        shorthost=\$(echo \$hostent | cut -d" " -f3)
        echo "shorthost=\${shorthost}"
        if grep -Fq \$host /etc/hosts
        then
                echo "\"\$host\" already found in /etc/hosts. Replacing it..."
                sed -i "/\$host/d" /etc/hosts
        fi
        echo "Writing >\${hostent}< to /etc/hosts..."
        echo \${hostent} >> /etc/hosts
        echo "Writing the result of ssh-keyscan \$ip to /root/.ssh/known_hosts..."
        ssh-keyscan \$ip >> /root/.ssh/known_hosts
        echo "Writing the result of ssh-keyscan \$host to /root/.ssh/known_hosts..."
        ssh-keyscan \$host >> /root/.ssh/known_hosts
        echo "Writing the result of ssh-keyscan \$shorthost to /root/.ssh/known_
hosts..."
        ssh-keyscan \$shorthost >> /root/.ssh/known_hosts
done <\${filename}

echo "Display known_hosts file..."
cat /root/.ssh/known_hosts

# Tar up root's .ssh directory
echo "Tar up /root/.ssh..."
tar -czvf sshbootstrap.tar.gz /root/.ssh

# Copy the tarball back to /home/ace and give the ace user ownership
echo "chown ace:users /home/ace/sshbootstrap.tar.gz..."
chown ace:users /home/ace/sshbootstrap.tar.gz
EOF

echo "Make setup-deployment-node executable..."
chmod +x ./setup-deployment-node

echo "--------------Begin setup-deployment-node script--------------"
cat ./setup-deployment-node
echo "-------------End setup-deployment-node script-------------"

# Generate an RSA key-pair on hostnames[0]; ignoring errors
echo "scp setup-deployment-node file to ace@${ips[0]}..."
scp ./setup-deployment-node ace@${ips[0]}:
echo "scp installer_hosts file to ace@${ips[0]}..."
scp ./installer_hosts ace@${ips[0]}:

# Execute setup-deployment-node script on hostnames[0]
echo "Execute setup-deployment-node file on ace@${ips[0]}..."
ssh -t ace@${ips[$hostnames[0]]} sudo ./setup-deployment-node

echo "Copy sshbootstrap.tar.gz file from ace@${ips[0]}..."
scp ace@${ips[0]}:sshbootstrap.tar.gz ./

# Unpack root's .ssh directory on to each node
for i in "${!ips[@]}"
do
        :
        echo "Copying ./sshboostrap.tar.gz to ace@${ips[$i]}..."
```

```
        scp ./sshbootstrap.tar.gz ace@${ips[$i]}:
        echo "Unpacking ./sshboostrap.tar.gz on ace@${ips[$i]}..."
        ssh -t ace@${ips[$i]} sudo tar -zxvf ./sshbootstrap.tar.gz -C /
        #ssh -t ace@${ips[$i]} sudo hostname -F /etc/hostname
done
rm -f ./sshbootstrap.tar.gz

# TODO: setup /etc/hosts on each host
scp ace@${ips[0]}:/etc/hosts ./etc.hosts
for i in "${!ips[@]}"
do
        :
        echo "Seting up /etc/hosts on ace@${ips[$i]}..."
        scp ./etc.hosts ace@${ips[$i]}:hosts
        ssh -t ace@${ips[$i]} sudo mv /home/ace/hosts /etc/hosts
        #echo "Change ace user's password to ace..."
        #ssh -t ace@${ips[$i]} echo -e "ace\nace" | sudo passwd -f ace
done

echo "Run 'hostname' on each node..."
# SSH-ing to each host as too
# Unpack root's .ssh directory on to each node
for i in "${!ips[@]}"
do
        :
        ssh -t ace@${ips[$i]} hostname
done
```

etc.hosts

```
# Your system has configured 'manage_etc_hosts' as True.
# As a result, if you wish for changes to this file to persist
# then you will need to either
# a.) make changes to the master file in /etc/cloud/templates/hosts.redhat.tmpl
# b.) change or remove the value of 'manage_etc_hosts' in
#     /etc/cloud/cloud.cfg or cloud-config from user-data
# The following lines are desirable for IPv4 capable hosts
127.0.0.1 localhost.localdomain localhost
127.0.0.1 localhost4.localdomain4 localhost4

# The following lines are desirable for IPv6 capable hosts
::1 localhost.localdomain localhost
::1 localhost6.localdomain6 localhost6

10.2.185.1 moab.vapor2.ac moab
10.2.185.2 torque-server.vapor2.ac torque-server
10.2.185.3 compute1.vapor2.ac compute1
10.2.185.4 compute2.vapor2.ac compute2
```

4. On the deployment node (as root or the passwordless-sudoer):

    a. Unpack the latest Moab HPC Suite RPM Bundle.

    ```
    tar -zxvf ./moab-hpc-suite-<VERSION>.tar.gz
    cd ./moab-hpc-suite-<VERSION>/automated-installer
    ```

b. Run the Automated Installer on the deployment node (Moab Head Node is recommended)

```
./automated-installer.sh webui

################################################################################
#

################################################################################
#
####
####    The Moab Automated Installer user interface is now available at:
####
####         http://<DEPLOYMENT HOST IP DISPLAYS HERE>:7443
####
####    Leave this shell running until you are instructed to exit this process.
####
####    If, for any reason, this process is intentionally or unintentionally
####    terminated, you may simply restart it and resume the installation
process.
####
####    Ctrl-C will signal this process to terminate.
####

################################################################################
#

################################################################################
#
```

c. Open `http://<DEPLOYMENT HOST IP>:7443` in a web browser (Chrome is recommended).

If you have previously used the Automated Installer, go to `http://<DEPLOYMENT HOST IP>:7443/home`. (Be sure to include the trailing `/home`, or it may jump ahead to a later stage (e.g. `http://<DEPLOYMENT HOST IP>:7443/finished`).)

d. Select `Torque` as the resource manager.

e. Select `Full Suite`.

f. Select the Head Node Count that best describes the cluster Head/Support/Compute Node configuration. Some features, like Reporting, require additional support nodes (i.e. Spark Nodes and Kafka Nodes). You must provision/set up any additional nodes before completing this step (see step 1 above).

g. Fill out the form and click `Next`.

h. Fill out the configuration. The important part here is getting the passwords and other configuration variables for the components already installed on your cluster right. Otherwise, new components will not be able to authenticate, for example, to MongoDB/PostgreSQL/MWS, etc.

i. Skip the Summary step by changing `home` in the URL to `preflight` (e.g., `http://10.2.185.1:7443/preflight`).

j. Run the preflight checks, but do NOT click `Next`.

k. If your Moab installation is not in the default `/opt/moab` directory, add `moab_prefix` and `moab_home_dir` variables to `/root/moab-hpc-suite-<VERSION>/automated-installer/webui/vars.yml`.

```
moab_prefix: "/opt/moab"
moab_home_dir: "/opt/moab"
```

l. Run the install from the command line, as shown below. The following example assumes `moab-hpc-suite-<VERSION>` has been unpacked in `/root` and that you are installing Reporting and Viewpoint. Replace `<VERSION>` with the version you downloaded and replace the comma-separated list of `-tags` with the list of features you are installing. The complete list of features are as follows: `scheduling`, `torque_resource_manager`, `accounting`, `portal_ui`, `reporting`, `high_throughput`, and `remote_viz`. Note that if you are installing reporting you need to add the `mws` and `insight` tags if MWS and Insight are not already installed.

```
[root]# cd /root/moab-hpc-suite-<VERSION>/automated-installer
[root]# ./venv/bin/ansible-playbook \
-vvv \
-i webui/inventory \
playbooks/full-suite.yml \
-e log_path=automated_installer.log \
-e @webui/vars.yml \
-e version_dir=VERSION> \
-e suite_path=/root/moab-hpc-suite-VERSION> \
-e test_user=adaptive \
-e nitro_enabled=true \
-e fastx_enabled=true \
--tags='reporting,portal_ui,mws,insight'
```

m. Skip to the Licensing step by changing `preflight` in the URL to `licensing` (e.g., `http://10.2.185.1:7443/licensing`.

n. Install licenses. Only install licenses required by components installed on your cluster.

o. Skip to the Postflight step by changing `licensing` in the URL to `postflight` (e.g., `http://10.2.185.1:7443/postflight`).

p. Run postflight checks. Checks run on features you do not have installed can be ignored.

q. Skip to the Finished step by changing `postflight` in the URL to `finished` (e.g., `http://10.2.185.1:7443/finished`).

4.6 Offline Installation With the Automated Installer

An offline installation method is available for configurations where the hosts in the Moab environment do not have internet access. The method assumes you have an Internet-connected system where you can download everything needed for the install. All the data is then copied to the non-Internet-connected server (by scp, DVD, USB, etc.). The data will then be set up in a local yum repository. As most installations require Moab, Insight, etc. to be installed across multiple servers, other servers can install packages from the server that has the local repository. Alternatively, the data can be copied to each server into its own local yum repository.

Besides yum packages, you will also need to download Java and some Python packages.

> For this installation method to work the correctly, both the connected and offline servers should have exactly the same operating system version installed.

> Support for offline installation using the Automated Installer is supported only via an Adaptive Computing Professional Services engagement (see http://www.adaptivecomputing.com/professional-services/).

In this topic:

4.6.1 Connected Server Process

These instructions assume the user is non-root, but with sudo rights.

To download everything needed for the install to the connected server:

1. Download the Moab distro-tarball from the Adaptive Computing Moab HPC Suite Download Center (https://www.adaptivecomputing.com/support/download-center/moab-hpc-suite-download/).

2. Untar the tarball and `cd` into it.

3. Set up the Moab suite RPMs in a local yum repository.

```
# sudo ./install-rpm-repos.sh -y
```

4. Make sure your user owns `/opt`.

```
# sudo chown <user>:<user> /opt
```

5. Make an output directory to store all RPMs.

```
# mkdir /opt/moab-offline
```

6. Download all RPMs.

   a. Symlink all the Moab suite RPMs to your offline directory. Otherwise, repotrack will not copy them.

   ```
   # ln -s /opt/adaptive-rpm-repository/rpm/*.rpm /opt/moab-offline/
   ```

   b. Use repotrack to download all dependency RPMs

   ```
   # rrepotrack -a x86_64 -p /opt/moab-offline moab-hpc-suite
   ```

7. Download a Java RPM into the moab-offline directory.

```
# cd /opt/moab-offline
# wget <java_url>
```

8. Create the repo before copying the files around.

```
# createrepo .
```

9. Create a repo file.

```
# echo "[moab-offline]
name=moab-offline
baseurl=file:///opt/moab-offline
failovermethod=priority
enabled=1
gpgcheck=0" > moab-offline.repo
```

10. Tar up the directory for easy transport.

```
# tar hczvf moab-offline.tgz moab-offline
```

   The h option ensures the symlinked files will be followed instead of copied.

4.6.2 Offline Server Process

These directions will prepare a yum repo on a server with all required Moab suite dependencies. Other servers can then point to this repository to install their packages. Once this is done, you can follow the regular install documentation,

4.6.2.A Setting Up the Yum Server

Copy the tar file generated on the connected server to the offline server.

```
# mv moab-offline.tgz /opt
```

```
# cd /opt
# tar xvzf moab-offline.tgz
# cp moab-offline/moab-offline.repo /etc/yum.repos.d/
# yum clean all
```

4.6.2.B Setting Up Other Servers

1. Copy the `/opt/moab-offline/moab-offline.repo` file to `/etc/yum.repos.d/`.

    ```
    # cp moab-offline/moab-offline.repo /etc/yum.repos.d/
    ```

2. Clean all cached files from the repository.

    ```
    # yum clean all
    ```

Chapter 5: Troubleshooting

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

5.1 General Issues

This topic details some common problems and general solutions.

5.1.1 Where do I need to set credentials and what are the default values?

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

5.1.1.A Database Credentials

MongoDB

Data-base	User	Default Pass-word	Used By	Parameters
admin	admin_ user	secret1	system admins	NA
moab	moab_ user	secret2	/opt/moab/etc/moab-private.cfg	MONGOUSER, MONGOPASSWORD
moab	mws_ user	secret3	/opt/mws/etc/mws-config.groovy	grails.- mongo.username, grails.- mongo.password
moab	insight_ user	secret4	/opt/insight/etc/config.groovy	moab.- mongo.username, moab.- mongo.password

Data-base	User	Default Pass-word	Used By	Parameters
mws	mws_user	secret3	/opt/mws/etc/mws-config.groovy	grails.-mongo.username, grails.-mongo.password
insight	insight_user	secret4	/opt/insight/etc/config.groovy	mongo.username, mongo.password
insight	mws_user	secret3	http://<mws_server->:8080/mws/admin/plugins/edit/viewpoint-query-helper	user, password
nitro-db	nitro_user	secret5	/opt/nitro-web-services/etc/nitro.cfg	db_username, db_password
reporting	report-ing_user	secret6	/opt/reporting/application.conf	database.username, database.password

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

PostgreSQL

Database	User	Default Password	Used By	Parameters
moab_view-point	moab_view-point	changeme!	/opt/viewpoint/etc/viewpoint.cfg	VIEWPOINT_DATABASE_USER, VIEWPOINT_DATABASE_PASSWORD
mam	mam	changeme!	/opt/mam/etc/mam-server.cfg	database.user, database.password

Apache Drill

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

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

5.1.1.B Product Credentials

Moab Workload Manager

Declared Parameter		Used By		Defa-ult Valu-e
File	**Parameter Name**	**File**	**Parameter Name**	
/op-t/moab/etc/moab-private.cfg	MESSAGEQUEUESECR-ETKEY	/opt/mws/etc/mws-config.groovy	moab.mes-sageQueue.secretKey	NA
		/op-t/in-sight/etc/config.groovy	mes-sageQueue.secretKey	
/op-t/mo-ab/etc/.moab.key	NA	/opt/mws/etc/mws-config.groovy	moab.secretKey	NA

Moab Accounting Manager

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

Moab Web Services

Declared Parameter		Used By		Defau-lt Value
File	Parameter Name	File	Para-meter Name	
/op-t/m-ws/etc/mws-config.groovy	auth.defaultUser.username	http://<viewpoint_server->:8081/configuration/	Username	moab-admin
		/opt/moab/etc/moab-private.cfg	CLIENTCF-G [RM:mws] USERNAM-E	
/op-t/m-ws/etc/mws-config.groovy	auth.defaultUser.password	http://<viewpoint_server->:8081/configuration/	Password	change-me!
		/opt/moab/etc/moab-private.cfg	CLIENTCF-G [RM:mws] PASSWOR-D	
/op-t/m-ws/etc/mws-config.groovy	grails.plu-gin.s-pringsecurity.oauthProvider.clients [0].clientSecret	http://<viewpoint_server->:8081/configuration/	Client Secret	NA

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

Nitro Web Services

Declared Parameter		Used By	Default Value
File	**Parameter Name**		
/opt/nitro-web-services/etc/nitro.cfg	ws_admin_password	Installation - default NWS API user creation	ChangeMe2!
/opt/nitro-web-services/etc/nitro.cfg	ws_readonly_username	Installation - default NWS API user creation http://<viewpoint_server>:8081/configuration/ -> Nitro Services -> Username	nitro-readonly-user
/opt/nitro-web-services/etc/nitro.cfg	ws_readonly_password	Installation - default NWS API user creation http://<viewpoint_server>:8081/configuration/ -> Nitro Services -> Password	ChangeMe3!
/opt/nitro-web-services/etc/nitro.cfg	ws_writeonly_username	Installation - default NWS API user creation /opt/nitro-web-services/etc/zmq_job_status_adapter.cfg -> username	nitro-writeonly-user
/opt/nitro-web-services/etc/nitro.cfg	ws_writeonly_password	Installation - default NWS API user creation /opt/nitro-web-services/etc/zmq_job_status_adapter.cfg -> password	ChangeMe4!

Viewpoint

Declared Parameter		Used By	Default Value
File	**Parameter Name**		
/opt/viewpoint/etc/viewpoint.cfg	username	http://<viewpoint_server>:8081/login/	viewpoint-admin
/opt/viewpoint/etc/viewpoint.cfg	password	http://<viewpoint_server>:8081/login/	changeme!

5.2 Installation Issues

This topic details some common problems and general solutions for Moab HPC Suite installation.

5.2.1 Automated Installer Error: socket.gaierror Failure on Start-up

The Automated Installer may fail at start-up with the following error message:

```
[root]# ./automated-installer.sh webui
...
LOADER: Running aci-script.py
Traceback (most recent call last):
File "<string>", line 2, in <module>
File "aci-9.1.1.1-py2.7.egg/installer/server.py", line 129, in main
socket.gaierror: [Errno -2] Name or service not known
aci-script returned -1
```

To run the Automated Installer:

1. Use `hostnamectl` to set the hostname.

   ```
   [root]# hostnamectl set-hostname myhostname
   ```

2. Make sure the hostname and hostname -f are identical and that they match the hostname you have set.

   ```
   [root]# hostname
   myhostname
   [root]# hostname -f
   myhostname
   ```

5.2.2 Timeout or Internal Server Error When Accessing openSUSE Repository

Installation (either manual or Automated Installer) may fail when accessing an openSUSE repository. To resolve this error, change `repo-url` in the following scripts to reference a different repository (e.g., `http://mirror.uni-konstanz.de/opensuse/repositories/`):

```
install-rpm-repos.sh
automated-installer/playbooks/roles/mam_client/tasks/main.yml
automated-installer/playbooks/roles/mam_server/tasks/main.yml
automated-installer/playbooks/roles/moab_server/tasks/main.yml
```

5.3 Port Reference

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

Adaptive Computing Local RPM Repository

Location	Ports	Functions	When Needed
Deployment Host	80 443	Adaptive Computing Local RPM repository	The duration of the install when using RPM installation or Automated Installer methods.

RLM Server

Location	Ports	Functions	When Needed
RLM Server Host	5053	RLM Server Port	Always
RLM Server Host	5054	RLM Web Interface Port	Always
RLM Server Host	57889	Remote Visualization Port	If Remote Visualization is part of your configuration
RLM Server Host	5135	ISV adaptiveco Port (for the Adaptive license-enabled products)	For Moab Workload Manager *and* if Nitro is part of your configuration.

Automated Installer User Interface

Location	Ports	Functions	When Needed
Deployment Host	7443	User interface for collecting info about the install	The duration of the install using the Automated Installer method.

Torque Resource Manager

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

Moab Workload Manager

Location	Ports	Functions	When Needed
Moab Server Host	42559	Moab Server Port	If you intend to run client commands on a host different from the Moab Server Host *or* if you will be using Moab in a grid

Moab Accounting Manager

Location	Ports	Functions	When Needed
MAM Server Host	7112	MAM Server Port	If you will be installing the MAM Server on a different host from where you installed the Moab Server *or* you will be installing the MAM Clients on other hosts
MAM GUI Host	443	HTTPS Port	If using the MAM GUI
MAM Web Services Host	443	HTTPS Port	If using MAM Web Services
MAM Data-base Host	5432	MAM Post-greSQL Server Port	If you will be installing the MAM Database on a different host from the MAM Server

Moab Web Services

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

Moab Insight

Location	Ports	Functions	When Needed
Insight Server Host	5568	Insight Server Port	Always
Moab MongoDB Database Host	27017	Moab MongoDB Server Port	Always
Moab Server Host	5574	Moab Data Port	Always
Moab Server Host	5575	Moab Reliability Port	Always

Moab Viewpoint

Location	Ports	Functions	When Needed
Viewpoint Server Host	8081	Viewpoint Web Server Port	Always
Moab Server Host	8443	Viewpoint File Manager Port	Always
Viewpoint Database Host	5432	Viewpoint PostgreSQL Database Port	If you will be installing the Viewpoint Database on a different host from the Viewpoint Server

Remote Visualization

Location	Ports	Functions	When Needed
Remote Visualization Server Host (also known as the Gateway Server)	3443	FastX Web Server Port	Always
Remote Visualization Session Server Host (Torque MOM Host)	Add ports as required, e.g. TCP: 3443, 6000-6005, 16001, 35091 UDP: 117	Session Server Ports	Ports 16001 and 35091 are *only* needed when using gnome

Nitro

> ℹ The listed ports are for configurations that have only one coordinator. If multiple coordinators are run on a single compute host, then sets of ports (range of 4) must be opened for the number of expected simultaneous coordinators.

Location	Ports	Functions	When Needed
Compute Hosts (Nitro Coordinator)	47000	Coordinator/Worker communication	Always
Compute Hosts (Nitro Coordinator)	47001	Coordinator PUB/SUB channel - publishes status information	Always
Compute Hosts (Nitro Coordinator)	47002	Reserved for future functionality	
Compute Hosts (Nitro Coordinator)	47003	API communication channel	Always

Nitro Web Services

Location	Ports	Functions	When Needed
Nitro Web Services Host	9443	Tornado Web Port	Always
Nitro Web Services Host	47100	ZMQ Port	Always
Nitro Web Services Database Host	27017	Nitro Web Services MongoDB Server Port	If you will be installing the Nitro Web Services Database on a different host from Nitro Web Services

Reporting

Suggested Host	Service	Ports	Function	When Needed
Reporting Master	HDFS name node	8020	HDFS communication	Always
Reporting Master	HDFS name node	50070	HDFS web interface	Always
Reporting Master	Spark Master	6066, 7077	Spark communication	Always
Reporting Master	Spark Master	8082	Spark Master web interface	Always
Reporting Master	Apache Kafka	9092	Kafka communication	Always
Reporting Master	Apache Zookeeper	2181	Zookeeper communication with Kafka and Drill	Always

Suggested Host	Service	Ports	Function	When Needed
Insight Server	Apache Drill	8047	Drill HTTP interface	Always
Reporting Worker	HDFS data node	50075, 50010, 50020	HDFS communication	Always
Reporting Worker	Spark Worker	4040	Spark communication	Always
Reporting Worker	Spark worker	8083	Spark worker web interface	Always
MWS Host	Tomcat	8080	Reporting Web Services HTTP interface	Always
MWS Host	MongoDB	27017	MongoDB communication	Always

5.4 Moab Workload Manager Issues

This topic details some common problems and general solutions for Moab Workload Manager.

See also Troubleshooting and System Maintenance in the *Moab Workload Manager Administrator Guide.*

5.4.1 Moab error: "cannot determine local hostname"

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

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

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

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

5.4.2 Moab error: "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
...
```

5.5 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/tomcat` or `$CATALINA_ HOME/logs`.

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

5.5.1 MongoDB: Errors during MWS startup

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

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

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

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

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

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

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

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

To remedy, install MongoDB, start the `mongod` service and then restart the `tomcat` service. See 2.6.3.C Install MongoDB - page 78 (Manual Installation) or 3.10.2.B Install and Configure MongoDB - page 199 (RPM Installation) for more information on how to install and configure MongoDB.

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

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

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

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

  ```
  [root]# service tomcat 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.

5.5.2 MongoDB: Out of semaphores to get db connection

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

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

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

- Configuring Moab Web Services in the *Moab Web Services Reference 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.

> ℹ You must restart Tomcat after adding, removing, or changing `grails.mongo.options` parameters.
>
> As shipped, `/opt/mws/etc/mws-config.groovy` does not contain any `grails.mongo.options` parameters. To adjust their values, you need to add them to `/opt/mws/etc/mws-config.groovy`.
>
> The default value of `connectionsPerHost` is normally 10, but MWS sets it internally to 50.
>
> The default value of `threadsAllowedToBlockForConnectionMultiplier` is 5.
>
> Any of the options listed in MongoOptions can be specified in `/opt/mws/etc/mws-config.groovy`. Just use the prefix `grails.mongo.options` as shown above.

5.5.3 MongoDB: Connection wait timeout after 120000 ms

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

5.5.4 java.lang.OutOfMemoryError: Java heap space

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

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

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

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

> ℹ️ Beginning with Java 8, the MaxPermSize option is ignored.

5.5.5 java.lang.OutOfMemoryError: PermGen space

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

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

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

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

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

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

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

5.5.7 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
```

> ℹ️ The number `256` above may be substituted for the desired maximum number of Moab client connections.

5.5.8 MongoDB Service Does Not Start

When installing MWS on a Red Hat 7-based system, the MongoDB service may fail with the following error message:

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

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

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

Create the `mongo systemd` unit file:

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

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

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

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

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

After editing the file, start the MongoDB service:

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

Verify that the MongoDB service is running:

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

5.6 Moab Viewpoint Issues

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

5.6.1 General Configuration Issues

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

Environment Variable	Description
VIEWPOINT_CONFIG_PATH	Full path to config.json file.
VIEWPOINT_DATABASE_HOST	Database host.
VIEWPOINT_DATABASE_NAME	Database name.
VIEWPOINT_DATABASE_PASSWORD	Database user password.
VIEWPOINT_DATABASE_PORT	Database port.
VIEWPOINT_DATABASE_USER	Database user.
VIEWPOINT_DEBUG	Debug mode. Production = 0, debug = 1.
VIEWPOINT_LOG	Log file location.
VIEWPOINT_LOG_LEVEL	Log level (INFO, WARNING, ERROR, CRITICAL, or DEBUG).
VIEWPOINT_PERMISSIONS_PATH	Full path to permissions.json file.

Environment Variable	Description
VIEWPOINT_PREFIX	URL prefix (defaults to /).
VIEWPOINT_STATIC_ROOT	Location of compiled static files.
VIEWPOINT_STATIC_URL	URL prefix for static resources (defaults to /static/).
VIEWPOINT_TEST	TEST mode (used for UI tests only). Production = 0, test = 1.
VIEWPOINT_SESSION_AGE	Lifetime of the user session in seconds (defaults to 2 weeks).

5.6.2 Only the Configuration Page is Displayed in Viewpoint

A particular configuration problem causes Viewpoint to display only the Configuration Page with only the Viewpoint License tab (not the Moab License tab). The Viewpoint License tab includes links only to the Basic Configuration and Licensed Features pages as shown below.
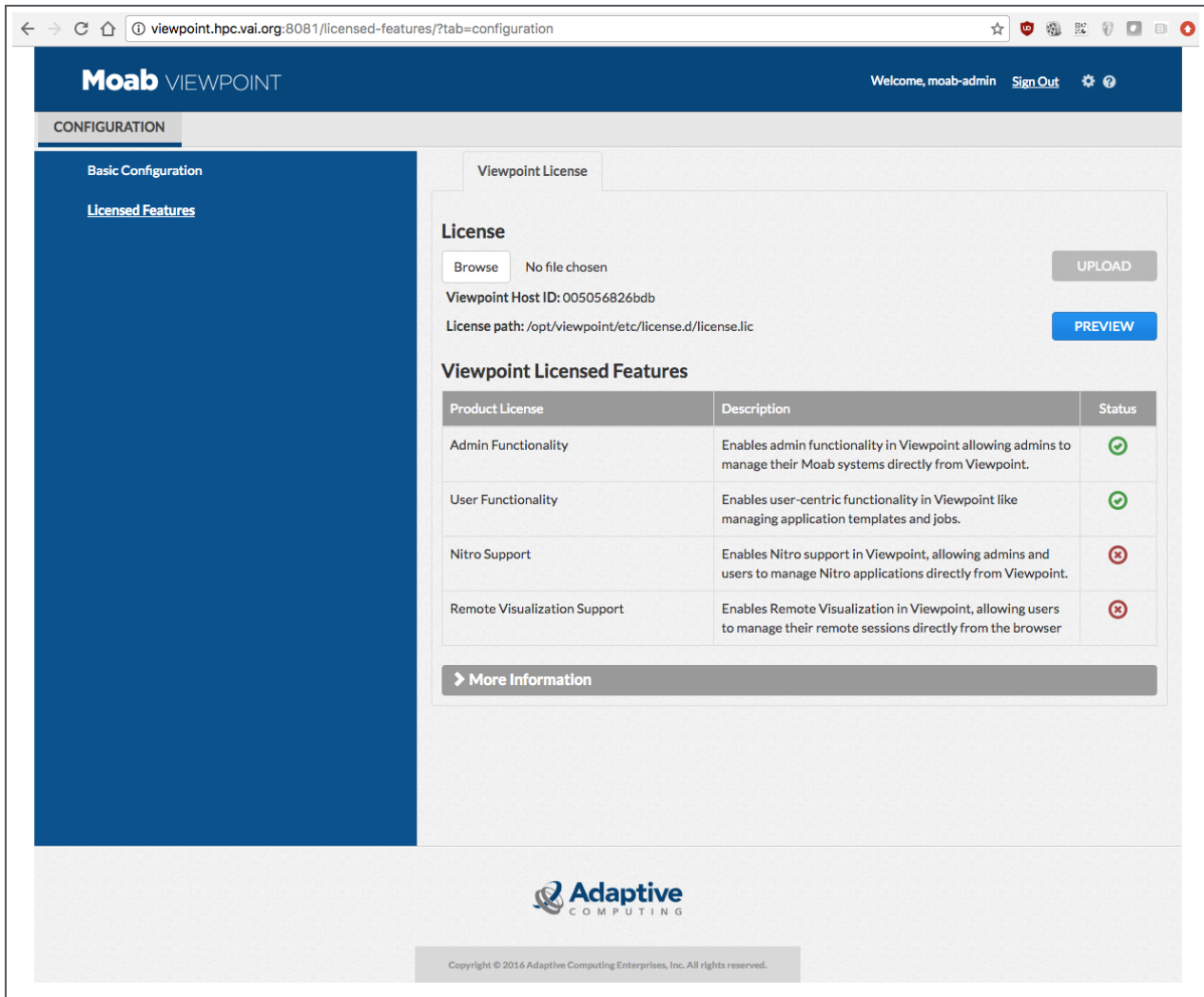
This problem occurs when the Viewpoint admin user is the same as the auth.defaultUser.username in MWS.

To resolve this issue, do the following:

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

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

   ```
   [admin]
   username = viewpoint-admin
   password = pbkdf2_
   sha256$20000$ZHeToCJgrSUH$+xmzYdhpqZCJokxO9eGzyr2B6jrfCgLlBT+pBgMis4w=
   ```

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

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

The auth2_user table will be displayed, similar to the following.

```
 id | is_active | is_staff | is_superuser |           last_login           |
username
----+-----------+----------+--------------+--------------------------------+--------
---------
  2 | t         | t        | f            | 2016-12-19 11:49:27.765855-05 |
viewpoint-admin
  1 | t         | t        | f            | 2016-12-19 12:06:24.642922-05 | moab-
admin
(2 rows)
```

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

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

5.6.3 Viewpoint Does Not Report Any of My Jobs or Nodes

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

Verify the following:

1. **Moab HPC Suite Setup**

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

   Please take a look at the following diagram describing our data flow architecture:
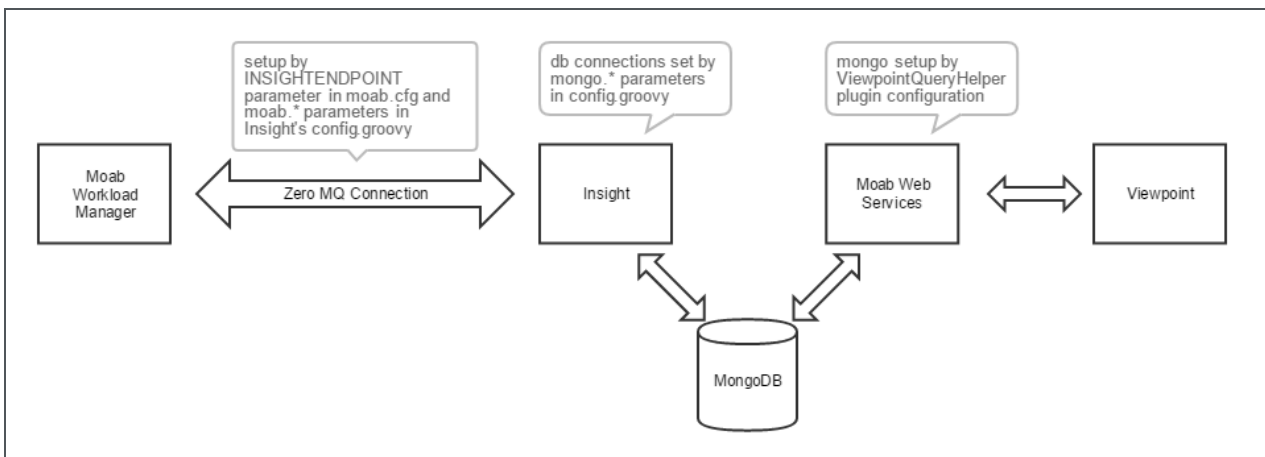
   The Moab Workload Manager will push data into Insight using a ZeroMQ message queue pipe.

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

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

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

2. **Hardware Specifications**

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

   See 1.2 Server Hardware Requirements - page 7 for more information.

3. **RPM Versions**

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

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

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

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

For example:

To resolve this issue, do the following:

1. If you have not already done so:

   a. Log in as an administrator to your MWS instance.

   b. Select `Plugins`, and then select `Plugin Monitoring`. You should see a page similar to the example image displayed earlier in this section.

2. In the Disabled Plugins section, click on the link for the viewpoint-query-helper plugin.

3. When the Show Plugin page displays, click `Edit`.

4. Enter the correct connection information, and then click `Update` to save your changes (you are returned to the Show Plugin page).

5. Return to the Plugin Monitoring page and start the plugin using the green start button.

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

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

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

5.6.5 Job's Processor Count Changes After Submission

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

5.7 Reporting Issues

This topic details some common problems and general solutions for Viewpoint reporting.

5.7.1 Reporting Log Files

If reporting is not working correctly check to see if there are error messages in the reporting logs. By default the logs are stored in `/var/log/spark/driver.log` and `/var/-log/spark/executor.log`

The location of these log files can be changed by modifying `log4j` settings in the `/etc/spark` directory on your reporting master host and reporting worker hosts.

```
[root]# vi /etc/spark/log4j-driver.properties

log4j.appender.file.File=/var/log/spark/driver.log
```

```
[root]# vi /etc/spark/log4j-executor.properties

log4j.appender.file.File=/var/log/spark/executor.log
```

Other logging information can be found in the following files:

- `/opt/spark-1.6.2/work/stderr`
- `/opt/spark-1.6.2/work/stdout`

5.7.2 Spark Unable To Create New Native Thread

When the reporting application cannot create a thread, you may receive the following error message:

```
Spark java.lang.OutOfMemoryError: Unable to create new native thread
```

Often this error can be worked around by increasing the per-user resource limits as described in 3.13.5.B Set Up the Spark Worker Host - page 238.

On the Spark Master and each Spark Worker host, do the following:

1. Increase user resource limits.

```
[root]# vi /etc/security/limits.conf
```

Add the following lines.

```
root soft core unlimited
root soft stack unlimited
root soft nproc unlimited

root hard core unlimited
root hard stack unlimited
root hard nproc unlimited

spark soft core unlimited
spark soft stack unlimited
spark soft nproc unlimited

spark hard core unlimited
spark hard stack unlimited
spark hard nproc unlimited
```

2. Exit and re-login from the terminal for the changes to take effect. Verify the changes worked.

```
[root]# exit
[user]$ ssh root@<reporting-master-host>
[root]# ulimit -a | grep -P "core|stack|processes"
core file size          (blocks, -c) unlimited
stack size              (kbytes, -s) unlimited
max user processes            (-u) unlimited
```

On the Spark Master, run:

```
[root]# service spark-master restart
```

On the Spark Worker, run:

```
[root]# service spark-worker restart
```

On the Tomcat host where Reporting Web Services is installed, run:

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

5.7.3 New Data Not Appearing in Reports Due to kafka.-common.OffsetOutOfRangeException

If no new data is being added to your reports, check to see if the following error appears in the Spark executor log (`/var/log/spark/executor.log`) on one of the reporting worker hosts:

```
[ERROR] 2017-12-11 22:22:04,318 org.apache.spark.executor.Executor logError -
Exception in task 0.0 in stage 0.0 (TID 0)
kafka.common.OffsetOutOfRangeException
at sun.reflect.NativeConstructorAccessorImpl.newInstance0(Native Method)
at sun.reflect.NativeConstructorAccessorImpl.newInstance
(NativeConstructorAccessorImpl.java:62)
```

This error usually means that messages in Kafka expired before the Reporting Framework was able to process them. This can happen if the Reporting Framework has been shut down for more than a week.

To fix this problem, tell the Reporting Framework stop trying to process the expired messages and start processing the earliest message available. To do this, do the following:

1. Stop the Spark Master and Worker services.

   a. On the Reporting Master host, run

   ```
   [root]# service spark-master stop
   ```

   b. On the Reporting Worker host, run

   ```
   [root]# service spark-worker stop
   ```

2. On the host where Zookeeper is running (usually on the Kafka Master host), start the Zookeeper shell.

   ```
   [root]# /opt/kafka_2.11-0.10.0.1/bin/zookeeper-shell.sh localhost:2181
   ```

3. Delete the `/reporting/offsets` znode.

   ```
   zookeeper> rmr /reporting/offsets
   ```

4. Start the Spark Master and Worker services.

   a. On the Reporting Master host, run

   ```
   [root]# service spark-master start
   ```

   b. On the Reporting Worker host, run

   ```
   [root]# service spark-worker start
   ```

5. On the Reporting Web Service host, restart Tomcat.

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

6. Wait a few minutes and, using the Zookeeper shell, check that the `/reporting/offsets` znode has been recreated and contains data.

   ```
   zookeeper> ls /reporting/offsets
   [5890fc1d247aa73a6c9f6f14, 5890fc1d247aa73a6c9f6f36, 5890fc1d247aa73a6c9f6f24,
   5890fc1d247aa73a6c9f6f45, 5890fc1d247aa73a6c9f6f11]
   zookeeper> quit
   ```

7. Check that reports are being updated with current data.

5.8 Nitro Web Services Issues

This topic details some common problems and general solutions for Nitro Web Services.

In this topic:

5.8.1 Logging

Logs are located in `/opt/nitro-web-services/logs/*.log`.

Logging is set to INFO (just below DEBUG) by default. Each service has it's own log_level setting. See `/opt/nitro-web-services/etc/*.cfg` for details.

If you change the log_level, you must restart the respective service.

5.8.2 Debugging

Try running the service from the command line.

The following procedure is an example of debugging the nitro-web-services service.

1. Stop the nitro-web-services service and leave the nitro-zmq-job-status-adapter service running.

```
[root]# service nitro-web-services stop
```

2. Run the nitro-web-service service from the command line.

   - Use an ampersand (nitro-web-services &) if you want to run the service in the background.

   - Fully qualify the path (i.e. /usr/bin/nitro-web-services or /bin/nitro-web-services) if nitro-web-services isn't found.

```
[root]# nitro-web-services
```

3. Exercise the service from a client/UI.

   Check for stacktraces in STDOUT/STDERR.

4. If you need to debug further, contact your Adaptive Computing account manager.

5.8.3 Viewpoint Does Not Show Job Status Updates

If you are not getting job status updates after launching your Nitro job, it is likely to be an inter-node communication problem (reachability, firewall, etc.) or an authentication/authorization issue

(incorrect username/password). For an overview of how Nitro works in the context of other Moab HPC Suite components see Using Nitro With the Moab HPC Suite in the *Nitro HTC Administrator Guide*.

Use the following steps to diagnose the problem. The order in which you follow these steps isn't required, but is recommended.

1. Check "Job Status Update (ZMQ)".

   a. Job status updates are published to the ZMQ message bus by the Nitro Coordinator (node1, step 7 in the diagram). Each node (node1, node2, ..., node(N)) can play the role of the Nitro coordinator, therefore, each node must define the "nws-connector-address" in their respective Nitro configuration file (`/opt/nitro/etc/nitro.cfg`).

      Next steps:

      - Modify the `/opt/nitro/etc/nitro.cfg` as needed.

        If you are using a shared file system, you will only have to make the modification once; otherwise, make the update on each compute node.

   b. `<nitro-web-services-hostname>` must be reachable from each Nitro coordinator and the designated ZMQ port (47100) must match the "msg_port" defined in `/opt/nitro-web-services/etc/zmq_job_status_adapter.cfg` (defaults to 47100 when not explicitly defined).

      ```
      ...
      # Nitro Web Services connection allows Nitro to communicate job status
      information
      # to the Nitro Web Services connector. This option indicates name and port of
      the
      # remote server in the form: <host>:<port>
      nws-connector-address <nitro-web-services-hostname>:47100
      ...
      ```

      > ⓘ You must uncomment the nws-connector-address line, even if it is already correct.

      Next steps:

      - If your system uses firewalls, verify the necessary ports are open. See 3.17.1 Open Necessary Ports - page 279 for more information.

        You may also be able to use telnet, netstat, etc. to check if `<nitro-web-services-hostname>` is reachable and the configured ZMQ port is open.

2. Check the ZMQ Job Status Adapter log for information. Check "log_level" in `/opt/nitro-web-services/etc/zmq_job_status_adapter.cfg` on the Nitro Web Services host (for example, the Moab Head Node). When "log_level" is not defined, the default is "INFO". The only log level more verbose is "DEBUG". Restart the zmq-job-status-adapter service if you change any configuration options.

Tail the the ZMQ Job Status Adapter log (`/opt/nitro-web-services/logs/zmq_job_status_adapter.log`) while running a Nitro job.

```
[Moab Head Node]# tail -f /opt/nitro-web-services/logs/zmq_job_status_adapter.log
```

Next steps:

- If you see any information pertaining to your job, then the Nitro Coordinators are successfully communicating with the ZMQ Job Status Adapter via ZMQ.

- If you do not see any job information, check step 1 in this procedure.

3. Check whether the ZMQ Job Status Adapter can authenticate to Nitro Web Services.

The following are the ZMQ Job Status Adapter configuration settings (`/opt/nitro-web-services/etc/zmq_job_status_adapter.cfg`).

> 🛈 After initial installation, the defaults are depicted as comments in the configuration file.

```
# DNS/IP and port where REST API (i.e. Nitro Web Services) is hosted
#http_protocol = https
#rest_api_host = localhost
#rest_api_port = 9443
#username = nitro-writeonly-user
#password = ChangeMe4!
```

Try authenticating to Nitro Web Services from the ZMQ Job Status Adapter host (Moab Head Node).

```
[root@MoabHeadNode]# curl --insecure --data '{"username": "nitro-writeonly-user", "password": "ChangeMe4!"}' \
 https://localhost:9443/auth
```

Next steps:

- If you get an HTTP status code other than 200 or 401, make sure the Nitro Web Services service is up and running.

- If you get an HTTP status code of 200, go to step 4.

- If you get an HTTP status code of 401, the configured "username" (nitro-writeonly-user) is unable to authenticate.

    Do the following:

    a. Check the value of "ws_writeonly_username" and "ws_writeonly_password" in /opt/nitro-web-services/etc/nitro.cfg.

    b. Set "username" and "password" in /opt/nitro-web-services/etc/zmq_job_status_adapter.cfg so that they match.

c. Restart the ZMQ Job Status Adapter service.

d. Retry the above curl command with the updated "username" and "password".

- If you still don't get an HTTP status code of 200, try resetting the nitro-writeonly-user's password in MongoDB.

```
# Any user can update its own password.
# The nitro-admin user can update any user's password.

# Obtain a nitro-key (session token) by authenticating as either the nitro-admin
or the nitro-writeonly-user
# Option 1: nitro-admin
[root@MoabHeadNode]# curl --insecure --data '{"username": "nitro-admin",
"password": "ChangeMe2!"}' https://localhost:9443/auth
# Option 2: nitro-writeonly-user
[root@MoabHeadNode]# curl --insecure --data '{"username": "nitro-writeonly-
user", "password": "ChangeMe4!"}' https://localhost:9443/auth

# Example nitro-admin authentication response:
> {"status": 200, "data": {"nitro-key":
"3e0fb95e9a0e44ae91daef4deb500dcc67a3714880e851d781512a49", "user": {"username":
"nitro-admin", "last_updated": "2016-08-19 16:46:17.395000", "name": "Nitro
Admin", "created": "2016-08-19 16:46:17.395000", "auth": {"job": ["read",
"write", "delete"], "user": ["read", "write", "delete"]}}}}

# Use the nitro-key from the authentication response to change nitro-writeonly-
user's password
[root@MoabHeadNode]# curl --insecure -X PUT --header "nitro-key:
3e0fb95e9a0e44ae91daef4deb500dcc67a3714880e851d781512a49" --data '{"password":
"Astr0ngPa$$!"}'  https://localhost:9443/user/nitro-writeonly-user
```

- Once you have reset the password, do the following:

a. Update the ZMQ Job Status Adapter's configuration.

b. Restart the service.

c. Update the curl command to use the new password.

d. Rerun the curl command.

4. Check if Viewpoint can authenticate to Nitro Web Services. Follow the instructions in the 3.17.4 Configure Viewpoint for Nitro Web Services - page 283.

If the TEST button indicates failure, then try the following curl command from the Nitro Web Services host, using the 'ws_readonly_username' and 'ws_readonly_password' defined in /opt/nitro-web-services/etc/nitro.cfg.

```
[root@MoabHeadNode]# curl --insecure --data '{"username": "nitro-readonly-user",
"password": "ChangeMe3!"}' \
 https://localhost:9443/auth
```

Next steps:

- If you get an HTTP status code other than 200 or 401, make sure the Nitro Web Services service is up and running.

- If you get an HTTP status code of 200, and the username and password used in the curl command match the Nitro Services Configuration in Viewpoint > Configuration, the Viewpoint server is unable to communicate with the Nitro Web Services host. Login to the Viewpoint host and check if the Nitro Web Services host and port (i.e. 9443) is reachable (i.e. ping the host and/or use telnet or netcat to test port 9443). You might need to check firewall settings.

- If you get an HTTP status code of 401, in Viewpoint, the configured "username" (i.e. nitro-readonly-user) is unable to authenticate.

  Do the following:

  a. Check the value of "ws_readonly_username" and "ws_readonly_password" in /opt/nitro-web-services/etc/nitro.cfg

  b. Set "username" and "password" in the Viewpoint "Nitro Services Configuration" so that they match.

  c. Retry the above curl command with the updated "username" and "password".

  d. If you still don't get an HTTP status code of 200, try resetting the nitro-readonly-user's password in MongoDB.

```
# Any user can update his/her/its own password.
# The nitro-admin user can update any user's password.

# Obtain a nitro-key (session token) by authenticating as either the nitro-
admin or the nitro-readonly-user
# Option 1: nitro-admin
[root@MoabHeadNode]# curl --insecure --data '{"username": "nitro-admin",
"password": "ChangeMe2!"}' https://localhost:9443/auth
# Option 2: nitro-readonly-user
[root@MoabHeadNode]# curl --insecure --data '{"username": "nitro-readonly-
user", "password": "ChangeMe4!"}' https://localhost:9443/auth

# Example nitro-admin authentication response:
> {"status": 200, "data": {"nitro-key":
"3e0fb95e9a0e44ae91daef4deb500dcc67a3714880e851d781512a49", "user":
{"username": "nitro-admin", "last_updated": "2016-08-19 16:46:17.395000",
"name": "Nitro Admin", "created": "2016-08-19 16:46:17.395000", "auth":
{"job": ["read", "write", "delete"], "user": ["read", "write", "delete"]}}}}

# Use the nitro-key from the authentication response to change nitro-
readonly-user's password
[root@MoabHeadNode]# curl --insecure -X PUT --header "nitro-key:
3e0fb95e9a0e44ae91daef4deb500dcc67a3714880e851d781512a49" --data '
{"password": "Astr0ngPa$$!"}'  https://localhost:9443/user/nitro-readonly-
user
```