HYPERIC | All Systems Go.

Hyperic HQ for Amazon Web Services

Hyperic, Inc., 609 Mission Street, 4th Floor, San Francisco, CA 94105

Table of Contents

About This Guide

This document has instructions for launching HQ Enterprise for Amazon Web Services (HQE for AWS) in the Amazon cloud, and deploying HQ Agents to EC2 instances you want to manage with HQ.

Where to Start

If you are new to Hyperic HQ, start by reading Overview of Hyperic HQ on page 5.

Otherwise, proceed to Launch HQE for AWS Instance on page 9.

Related Hyperic Documentation

To access online documentation for HQ, see HQ Documentation.

To download a guide for the first-time HQ evaluator or user, click Hyperic HQ 4.0 Product Tour.

Amazon EC2 Information

A broad set of information about EC2 is available on the Amazon EC2 Resource Center.

The Getting Started Guide on that site is a good introduction to EC2.

The EC2 Developer Guide documents the APIs and commands that are referenced in the instructions below.

Getting Help

If you have problems getting the product up and running, see HQ Documentation or visit the forums at HQ Community. Once you are up and running, click **Help** in Hyperic HQ's user interface for more information.

Overview of Hyperic HQ

This section is a brief overview of Hyperic HQ functionality and architecture.

- *Hyperic Products*
- *Hyperic HQ Functionality in a Nutshell*
- *HQ for AWS Architecture and Deployment*
- *HQ Inventory and Access Model*
- *Hyperic HQ Functionality in a Nutshell*

Hyperic Products

Hyperic HQ provides monitoring and management for your web infrastructure. You can use it to streamline operations, manage infrastructure complexity, and drive service level improvements. Three versions are available:

- Hyperic HQ 4.0—Hyperic's open source offering is licensed under GNU GPL v2.
- Hyperic HQ 4.0 Enterprise—Hyperic's enterprise edition, available under a commercial license, has all the capabilities of the open source edition, plus advanced automation and control features for managing web applications at scale.
- Hyperic HQ for Amazon Web Services—HQ for AWS is the enterprise version of HQ, delivered as an Amazon Machine Instance. Hyperic offers several HQ for AWS AMIs, sized by how many machine instances can be managed.
    - **HQ for AWS-Developer** is free of charge from Hyperic, and limited to managing four machine instances.
    - **HQ for AWS-25** is available on a subscription basis through Amazon DevPay, and can manage up to 25 machine instances.
    - **HQ for AWS-100** is available on a subscription basis through Amazon DevPay, and can manage up to 100 machine instances.
    - **HQ for AWS-200** is available on a subscription basis through Amazon DevPay, and can manage up to 200 machine instances.

Each HQ AMI includes the HQ Server, the HQ Agent, and a pre-configured HQ database. As described in the procedures that follow, you should move the HQ database to Amazon Elastic Block Storage (EBS) to avoid loss of HQ data in the event that your HQ instance terminates.

Once you have deployed an HQ Server instance in the cloud, you can deploy HQ Agents to other AMI instances that you wish to manage with HQ and configure those agents to report to your cloud-resident HQ Server.

Hyperic HQ Functionality in a Nutshell

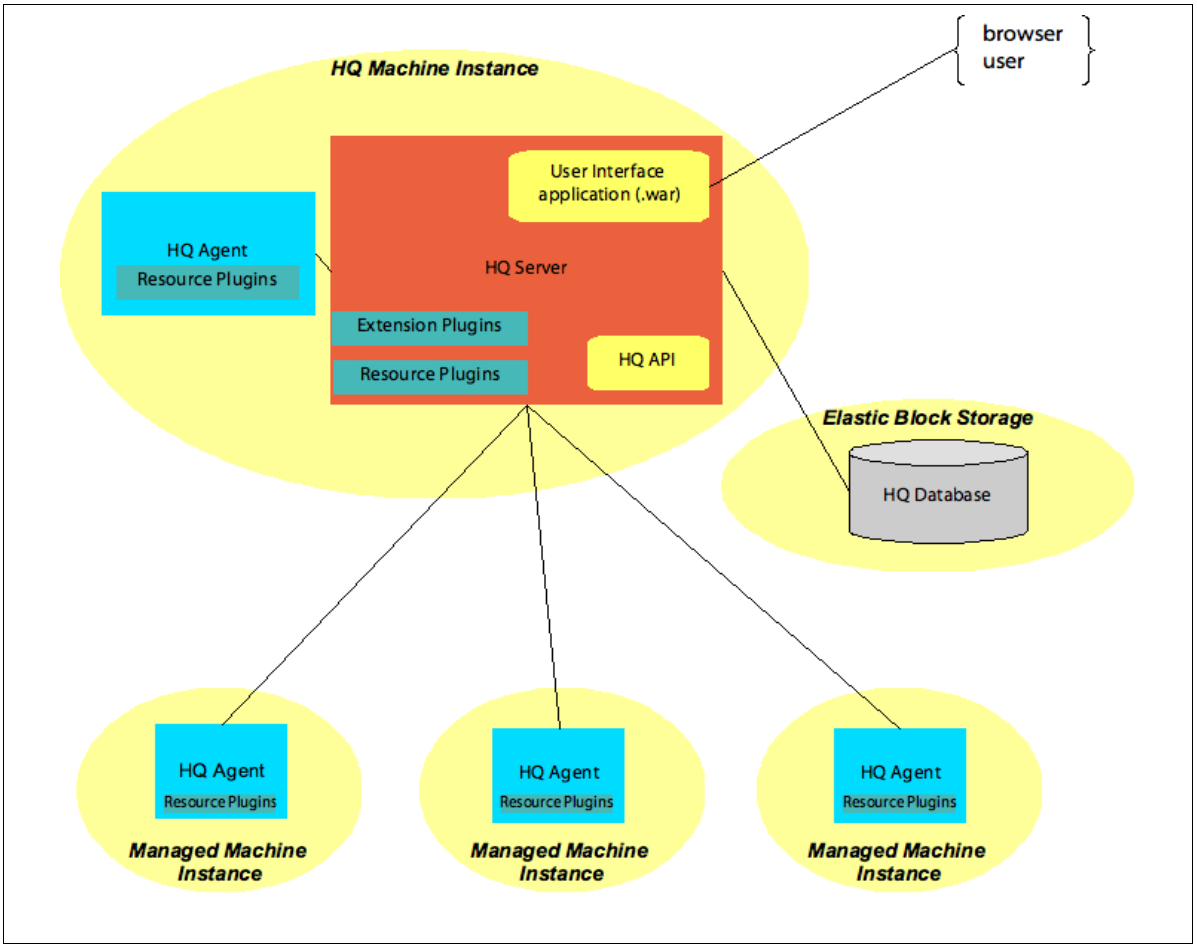
Hyperic HQ monitors and manages web applications across a broad range platforms and technologies, including on-premise and cloud-hosted. HQ provides these core management functions for your software and network resources:

- **Discover**—HQ Agents *auto-discover* your software resources, and populate a database with key information about your software inventory. Agents collect basic facts about each software resource, like its type, vendor, version, and location. Depending on the resource type, the agent collects information, like a machine or platform's architecture, RAM, CPU speed, IP address, and domain name. An agent uses built-in *resource plugins* to auto-discover commonly used operating systems, application servers, HTTP servers, databases, and other software and network resources. You can build your own resource plugins to manage software that HQ does not support out-of-the box, and take advantage of resource plugins contributed by the Hyperic community. For a list of resources you can manage with HQ, see Hyperic Community.

- **Organize**—The resources that HQ Agents discover are stored in the HQ database according to a hierarchical *inventory model*. The inventory model is fundamental to how HQ makes sense of a large number of software resources and the relationships among them.
- **Monitor**—HQ Agents collect metrics that reflect availability, performance, utilization, and throughput. Agents collect a standard set of metrics for each supported resource type. You can tailor which metrics you collect from the web-based user interface, and choose which metrics to spotlight in the Dashboard, the home page of the user interface. In addition to metrics, agents monitor logs, events, and configuration changes.
- **Control**—You can use HQ for remote control and administration of your software resources. Available control actions vary by resource type. For an application server, you can do tasks like starting, stopping, and garbage collection. For a database server, you can perform analysis or housekeeping functions.
- **Alert, notify, escalate**—You can set alerts on metrics and configure actions for HQ to perform when an alert fires. When an alert fires, HQ can respond in a variety of ways: it can issue email notifications, set SNMP traps, or communicate with another management system. You can define a sequence of responses to a fired alert—an *escalation scheme*—so problems don't fall through a crack.
- **Present, visualize, analyze**—The web-based user interface is a highly configurable window into performance and availability. The Dashboard is made up of portlets—you can add, remove, rearrange them, and control what information they present.

HQ for AWS Architecture and Deployment

The diagram illustrates the components of HQ for AWS and how they fit into an AWS deployment.



The HQ Machine Instance Hosts the HQ Server and an HQ Agent

The HQ machine instance you launch from the HQ for AWS AMI is where you run the HQ Server and an HQ Agent. The HQ database is packaged in the HQ for AWS AMI; during the deployment process described in *Launch HQE for AWS Instance* on page *9* you move the HQ database to Amazon Elastic Block Storage, to ensure persistence of monitoring data in the event of instance failure.

- HQ Server is the central server for the HQ for AWS deployment. It receives inventory and metric data from HQ Agents and stores it in the HQ database. The server provides facilities for managing your software inventory—it implements the HQ inventory and access model, grouping your software assets in useful ways to ease the process of monitoring and management. The HQ Server detects when alerts fire, and performs the notifications or escalation processes you define. It also processes actions that you initiate from the user interface or Hyperic HQ's web services API. The server provides authentication services, using an internal engine or an external authentication service.

  The HQ Server contains these other HQ components:

  - HQ user interface application—Hyperic HQ has a rich, web-based user interface, implemented as a web application that runs within the HQ Server. The home page of the user interface is the HQ Dashboard, a one page overview of software inventory changes, problem resources, recent alerts, and metric charts for important resources. Beyond the Dashboard are tabbed views for browsing inventory, viewing and visualizing metrics, and managing your monitoring and alerting logic.
  - HQ API—The HQ Server has a web services API that provides programmatic access to all HQ Server data and functionality.
  - Extension plugins—The architecture of the HQ Server supports pluggable extensions, which you can develop to extend the HQ user interface, develop scripts for automating commonly performed processes, and develop web services interfaces with other management systems.
- HQ Agent—In a typical HQ deployment, you run an HQ Agent on the machine that hosts the HQ Server as well as on on each machine you want to manage with Hyperic HQ. This allows you to monitor the health and well-being of the HQ Server, in addition to the managed applications throughout your AWS environment. Note however that there is no requirement to run an HQ Agent on the machine that hosts the HQ Server. Agent functionality is described in *An HQ Agent Runs on each Managed Machine Instance* below.

  The HQ Agent uses *resource plugins* to discover, monitor, and control software resources. HQ has many built-in resource plugins. You can build your own plugins for resources that HQ doesn't support off-the-shelf; community-contributed plugins are also available.

The HQ Database Runs on Elastic Block Storage

Any data stored on an EC2 machine instance is permanently lost if the instance is terminated. For this reason, the HQ database runs on Amazon's Elastic Block Storage (EBS). For ease of delivery, the HQ database is packaged with the HQ Server and HQ Agent in the HQ for AWS AMI. As a part of the deployment process, you move the HQ database to EBS.

The HQ database is the repository for the metrics, events, and log data collected by HQ Agents, and managed by the HQ Server. The database bundled with HQ for AWS is MySQL 5.0.45.

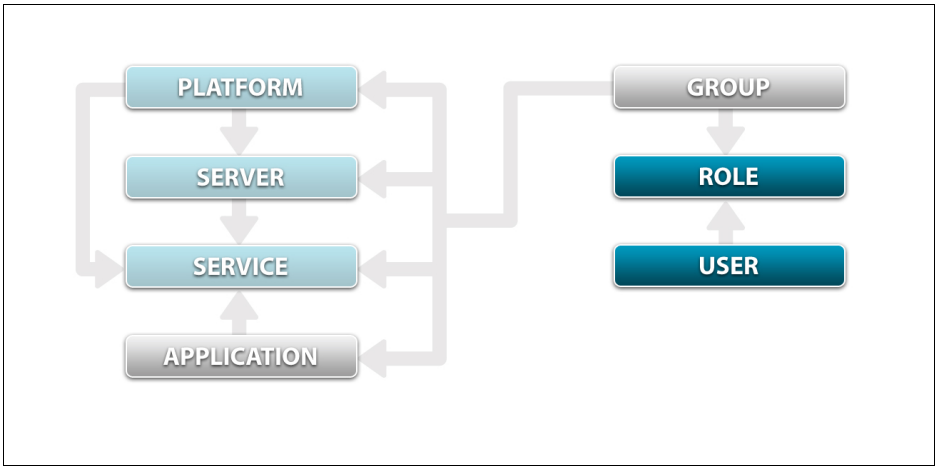
An HQ Agent Runs on each Managed Machine Instance

You run an HQ Agent on each machine instance you want to manage with Hyperic HQ. Upon first startup, the agent auto-discovers the software running on the machine, and periodically re-scans for configuration changes. HQ Agents gather availability, utilization, performance, and throughput metrics; perform log and event tracking; and allow you to perform software control actions, for instance, starting and stopping web and application servers. Agents send the inventory data and metrics they collect to the central HQ Server.

HQ Inventory and Access Model

In Hyperic HQ, *inventory* refers to the managed resources that make up your infrastructure: hosts, operating systems, application servers, application components, and other software components. HQ classifies individual resources into a hierarchy of inventory types: *platform*, *server*, and *service*. A platform is a hosting system, a server is a software product that runs on a platform, and a service is a component that runs in a server, or is associated with a platform.

Figure 2. HQ Inventory and Access Model



Hyperic HQ auto-discovers platforms, servers, and services. There are also two inventory types that you can configure yourself: *applications* and *groups*.

An application is a set of services that satisfy a related set of business requirements; monitoring performance and availability at the application level allows you to track service levels as experienced by real users.

A group is a collection of other resources. Groups serve multiple purposes in HQ. For example, if you have a group of resources of the same type (a *compatible group*), you can perform a control action, like a restart, on all its members with a single command. Groups are also fundamental to HQ's access model. You group inventory items that have the same access requirements together. By mapping groups and users to *roles* that define permission sets, you control what resources a user can access, and what actions they can perform on those resources.

Launch HQE for AWS Instance

This section has instructions for running an instance of the Hyperic HQ for AWS AMI,in the Amazon Elastic Compute Cloud (EC2). The AMI contains the HQ Server, the HQ Agent, and a pre-configured copy of MySQL. After launching the HQ for AWS instance, you create a storage volume on Amazon Elastic Block Storage (EBS) and move the HQ database to it.

Before You Start

To perform the steps described below, you need:

- an EC2 account, and the associated command-line tools
- familiarity with AMIs and experience launching application instances in the cloud
- an SSH keypair that you have generated—you'll need this to launch the HQ instance
- an SSH client

To ensure that your environment is properly configured, perform the verification steps described in the "Running an Instance" section of the EC2 Getting Started Guide.

Note: The configuration instructions assume that all steps are performed from a shell, rather than using the web-based AWS management console, which does not support all of the commands required to setup of HQE for AWS.

Create Security Group

The commands shown how below create a security group for your HQ instance, and specify that the HQ Server can be accessed only on its SSH port (22), plain text listen port (7080), and SSL listen port (7443).

```
$ ec2-add-group hq-server -d "Used for the Hyperic HQ AMI"
$ ec2-authorize hq-server -p 22
$ ec2-authorize hq-server -p 7080
$ ec2-authorize hq-server -p 7443
```

You can use other `ec2-authorize` command options to further secure your HQ instance. For example, you can specify a rule that the HQ Server will accept only requests from specific source addresses or subnets.

Note: Hyperic recommends that you secure the HQ instance you deploy in the cloud. If you have an existing security procedure for cloud-resident resources, extend it to your HQ deployment.

Find Hyperic HQ AMIs

You need to know the ID of the Amazon Machine Image (AMI) that contains the version of HQE for AWS you wish to launch. Run this command to list the currently available HQ AMIs:

```
ec2-describe-images -o 890100225205
```

Explanation: In this command, the `-o owner` option limits the list displayed to those owned by Hyperic, whose ID is `890100225205`

A list of Hyperic AMIs will be returned. Each item in the list looks like this:

```
IMAGE ami-b8688cd1 hyperic-amis/hqserver-401-EE-4Agent.manifest.xml 890100225205   available
public  i386 machine aki-a71cf9ce ari-a51cf9cc
```

In the response above:

The AMI ID, shown in bold in the example above, is "`ami-b8688cd1`".

The AMI manifest path, shown in bold in the example above, is `hyperic-amis/hqserver-401-EE-4Agent.manifest.xml`. The "4Agent" portion of the manifest indicates that this is the AMI for HQ for AWS-Developer. The manifest for the other HQ for AWS AMIs will contain "25Agent", "100Agent", and "200Agent".

Launch an Instance HQE for AWS

Run this command to launch an instance of the AMI:

```
$ ec2-run-instances -k MyKeypair -t type -z AvailabilityZone -g SecurityGroup amiID
-dINITPWORD=password
```

supplying the appropriate values for the command options defined in the table below.

Option	Description
`-k`	Specifies the SSH keypair. Replace *`MyKeypair`* with the name of your keypair.
`-t`	Specifies the type of the instance. Replace *`type`* with the appropriate value: ● m1.small—Use this for small HQE instances. ● m1.large—Use for the large HQE instances.
`-g`	Specifies the security group in which to run the HQ instance. Replace *`SecurityGroup`* with the name of your security group.
`-z`	Optional; specifies the availability zone where the HQ instance will run. If you do not specify an availability zone, one will be automatically selected. U.S. availability zones include: us-east-1a, us-east-1b, and us-east-1c
`amiID`	Identifies the AMI from which to launch the instance image. Replace *`amiID`* with the ID of the appropriate AMI.
`-d`*`INITPWORD`*	Sets the password you will have to supply when you log in the HQ Server. Replace *`password`* with the desired password.

A response like this is returned:

```
INSTANCE   i-2ca00145   ami-d5ce2abc   pending  hqserver-keypair 0   m1.large
2008-09-16T18:37:58+0000   us-east-1b
```

The INSTANCE row lists:

- the instance ID
- the AMI from which it was launched,

- current instance state ("pending" indicates it is being prepared for launch)
- the keypair name, if supplied at launch,
- instance type,
- launch time,
- availability zone

Upon successful launch, the HQ Server, HQ Agent, and HQ database will start up.

Note: The HQ startup sequence that occurs automatically when you start the HQ machine instance will take longer than subsequent startups will. Expect the first startup to take up to 10 minutes. A plain restart of the HQ Server will take less time.

How To: List Instance Information

At any time, `ec2-describe-instance` command lists information about one or all of the instances you have launched:

You may optionally append an Instance ID to the command to limit the results to a specific instance.

```
$ ec2-describe-instances InstanceID
```

An INSTANCE row like the one returned when you use the ec2-run-instances command is returned.

```
INSTANCE   i-2ca00145  ami-d5ce2abc  ec2-75-101-243-183.compute-1.amazonaws.com
ip-10-251-102-239.ec2.internal  running hqserver-keypair 0  m1.large 2008-09-
16T18:37:58+0000 us-east-1b aki-b51cf9dc    ari-b31cf9da
```

For instances that have been launched, the row contains additional data:

- the instance's public DNS name
- the instance's private DNS name.
- Amazon Kernel Image (AKI)
- Amazon RAM Image (ARI).

If you run the `ec2-describe-instance` command without appending an Instance ID, all of the instances you have launched will be listed.

Obtain an Elastic IP Address

In this step, you obtain an Elastic IP Address that you will associate with the HQ Server later in this procedure. Elastic IP Addresses are static public IP addresses designed for dynamic cloud computing. An Elastic IP address is associated with your EC2 account, not a particular instance, and you control that address until you choose to explicitly release it.

Elastic IP Addresses allow you to deal with instance or availability zone failures.

When you launch an HQ instance, it is automatically assigned a private IP address. If the HQ instance were to be terminated, the replacement instance you launch will have a different private IP address, hence the clients and HQ Agents that need to connect to it would be unable to do so.

Enter this command to obtain an public Elastic IP Address:

```
ec2-allocate-address
```

The response returned contains the Elastic IP Address that is now associated with your EC2 account, like this:

```
ADDRESS 75.101.155.119
```

For more information about Elastic IP Addresses, see Amazon's Feature Guide: Amazon EC2 Elastic IP Addresses.

How To: List the Elastic IP Addresses for Account

To list the elastic IP addresses assigned to you EC2 account, enter:

```
ec2-describe-addresses
```

A response like this is returned:

```
ADDRESS 75.101.155.119
```

If multiple elastic IP addresses were assigned to your EC2 account, they would be listed.

Associate Elastic IP Address with a the HQ Instance

In this step, you assign the elastic IP address to your HQ instance using the `ec2-associate-address` command:

```
ec2-associate-address -i InstanceID  ElasticIpAddress
```

supplying the Instance ID for your HQ instance and the elastic IP address you obtained.

The command processing will take a few minutes to complete. A response like this is returned:

```
ADDRESS 75.101.157.145   i-b2e019da
```

Notes: Instances only have one Internet-routable IP address. When an Elastic IP is associated with an instance, the instance's existing Public IP address mapping is removed and is no longer valid for the instance.

There is no charge for Elastic IP addresses that are mapped to an instance. Amazon charges $0.01 per hour for Elastic IP Addresses that are not mapped to an instance.

Configure HQ Email and Password Settings

In this step, you configure the "to" and "from" email address that HQ will use for email notifications, and the password for the **hqadmin** account, the default HQ administrator account..

1. Point a browser to the HQ Server's Elastic IP Address (determined in the previous step) on its plain or SSL listen port, 7080 or 7443. For example, for the example IP address from above:

   `https://75.101.157.145:7443`

2. On the HQ login page, enter **hqadmin** as your username, and the password you configured previously.

   The HQ Dashboard appears.

3. Click **Administration** in the Masthead.
4. Click **List Users** in the "Authentication/Authorization" section.
5. In the list of users displayed, click **hqadmin**.
6. On the HQ Administrator page, click **Change...** in the password field.
7. On the Edit HQ Administrator page, enter the new password and click OK.
8. On the HQ Administrator page, click **Edit** in the "General Properties" section.
9. On the Edit User page, enter the email address to which HQ should send email notifications, and click OK.
10. Click **Administration** in the Masthead and select **HQ Server Settings**.
11. In the "Email configuration Properties" section, enter the email address from which HQ should send email notifications, and click OK.

Create Storage Volume for HQ Data

At this point, your HQ database resides on the HQ instance. If the instance gets terminated, you will lose the data in the HQ database.

In this step, you configure a volume on Amazon's Elastic Block Service (EBS) and move the HQ database to it. This will ensure that you retain the contents of the HQ database if the HQ instance is terminated

Note: You **must** move the HQ database instance to EBS to prevent data loss in the event of instance termination.

Your EBS volume must reside in the same Amazon Availability Zone as your HQ instance. You should know the zone where the HQ instance from supplying it explicitly when launch it; the zone is also shown in the INSTANCE row that was returned when you launched it.

To create a 2GiB EBS volume in us-east-1b:

```
$ ec2-create-volume -availability-zone us-east-1b -size 2
```

A row like this is returned:

```
VOLUME  VolumeID    2   us-east-1b   creating   2008-09-16T22:53:19+0000
```

Attach Storage Volume to HQ Instance

Enter this command to attach the new volume to your HQ instance, replacing *InstanceID* with your HQ instance ID and *VolumeID* with the ID returned in the VOLUME row above:

```
$ ec2-attach-volume -i InstanceID -d /dev/sdh  VolumeID
```

Explanation: The *-d* option specifies the device to attach the volume to.

A row like this is returned:

```
ATTACHMENT   VolumeID    InstanceID   /dev/sdh   attaching   2008-09-16T22:54:45+0000
```

Move HQ Database to EBS Volume

1. Run the following commands:

   ```
   $ ssh -i id_rsa-hqserver-keypair hqadmin@ec2-67-202-9-216.compute-1.amazonaws.com
   $ sudo mkfs -t xfs /dev/sdh
   $ sudo /opt/hyperic/tools/mount_mysql.sh
   $ /opt/hyperic/server/bin/hq-server.sh stop
   $ /opt/hyperic/agent/bin/hq-agent.sh stop
   $ sudo mysqldump -Q -u root -p12345 --all-databases  > /mysql/full-db-dump.sql
   $ sudo /etc/init.d/mysqld stop
   ```

   Explanation: The commands above:

   a) Log in to the instance.

   b) Create the `/dev/sdh` filesystem.

   c) Run a script that mounts the filesystem on `/mysql` and sets permissions properly.

   d) Stop the HQ Server.

   e) Stop the HQ Agent.

   f) Dump the database contents to a file in `/mysql`.

   g) Stop MySQL.

2. Edit `my.cnf` to indicate new database directory. Add these lines to `my.cnf`:

   ```
   $ sudoedit /etc/my.cnf
   ```

   The default editor for the hqadmin user is set to `/usr/bin/vim` by way of the `EDITOR` environment variable in `/opt/hyperic/.bash_profile`. To change that immediately for the current session, simply do so from the command line like this:

   ```
   $ export EDITOR=/usr/bin/nano
   ```

   To make the change persistent across logins, change the value in the .bash_profile file as well.

   ```
   [mysqld]
   datadir         =/mysql
   socket          =/mysql/mysql.sock

   [mysqld_safe]
   socket          =/mysql/mysql.sock

   [client]
   port            = 3306
   socket          = /mysql/mysql.sock
   ```

3. Install the MySQL database:

   ```
   $ sudo /usr/bin/mysql_install_db
   ```

4. Start MySQL:

   ```
   $ sudo /etc/init.d/mysqld start
   ```

5. Restore your previous dump:

   ```
   $ mysql -u root < /mysql/full-db-dump.sql
   ```

6. Restart mysql:

   ```
   $ /etc/init.d/mysqld restart
   ```

7. Restart the HQ components:

   ```
   $ /opt/hyperic/server/bin/hq-server.sh start
   $ /opt/hyperic/server/bin/hq-agent.sh start
   ```

Deploy Agents to Machine Instances to be Managed

This section describes how you install an HQ Agent on an AMI, and configure the agent to report to your cloud-resident HQ instance. You can download Hyperic HQ EE Agents from the Hyperic website. Make sure that you choose the correct package for your platform. You could use a new Agent version with an older HQ Server installation, e.g. HQ EE Agent 4.0.3 and the shipped HQ EE Server 4.0.1.

1. Create an account on the target instance for running the HQ Agent. Log on as the user to perform the remaining steps.
2. Copy the HQ Agent package to the target instance, and unpack it.
3. Edit `agent.properties`, located in the top level `conf` directory of the agent, to specify setup properties. For more information see Configuring Agent Startup Settings in its Properties File in the HQ documentation at http://support.hyperic.com.
4. Start the agent. (The HQ Server must be running when you start the agent.) Enter this command to start the agent.

   AgentHome`/bin/hq-agent.sh start`

5. Check the system time offset between the HQ Server and the new agent. In the HQ user interface:

a. Go to the Administration page.

b. Click HQ Health in the plugins section.

c. Click the Agents tab.

d. Check the value in the Time Offset column for the agent. A value of less than 1000 milliseconds is acceptable. If the offset is greater, use synchronization utilities, such as ntpdate and ntpd, to correct the offset.

About HQE for AWS Lifecycle Management

Start and Stop Hyperic HQ Components in the Cloud

You can use the usual HQ lifecycle commands to start and stop the HQ Server and the HQ Agent running in the AMI.

Terminate the HQ Instance

To terminate your HQ instance, enter:

```
ec2-terminate-instances instance_id
```

Map a New HQ Instance to the HQ Database

If you upgrade or your instance is terminated, to map to a new HQ instance to the HQ database:

1. Start a new HQ instance in same availability zone as the previous instance ran.
2. Shut down the HQ Server.
3. Attach the volume to the new instance, as described in *Attach Storage Volume to HQ Instance* on page *13*.
4. Mount /mysql.
5. Stop MySQL.
6. Edit the MySQL.cnf file to point the database to the new instance.
7. Start MySQL.
8. Restart the HQ Server.