vFabric Hyperic Overview

VMware vFabric Hyperic 5.0

This document supports the version of each product listed and supports all subsequent versions until the document is replaced by a new edition. To check for more recent editions of this document, see http://www.vmware.com/support/pubs.
You can find the most up-to-date technical documentation on the VMware Web site at:
http://www.vmware.com/support/
The VMware Web site also provides the latest product updates.
If you have comments about this documentation, submit your feedback to:
docfeedback@vmware.com

Copyright © 2013 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at http://www.vmware.com/go/patents.
VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com
# Table of Contents

1 About vFabric Hyperic Overview 6
   Intended Audience 6

2 Use of "Hyperic," "vFabric Hyperic," and "Hyperic HQ" 7

3 Introduction to Hyperic Monitoring 8
   Main Components 8
   Hyperic Monitoring and Management Features 9
   Key Hyperic Facts for the New User 10

4 Resources, Resource Types and Inventory Types 11
   Resources and Resource Categories in Hyperic 11
   About Platforms, Servers, and Services 14
   About Groups in Hyperic 17
   About Applications in Hyperic 22

5 User Accounts and Roles in Hyperic 25
   Authentication 26
   User Accounts 26
   Roles in vFabric Hyperic 27

6 Resource Auto-Discovery Processes 33
   Hyperic Auto-Discovery Processes 33
   About Auto-Inventory IDs and InstallPath 37
   What the Agent Can and Cannot Discover 37
   How Discovered Resources Get into Hyperic Inventory 37
   What to Do After Adding New Resources to Inventory 37

7 Metrics and Metric Collection 39
   Metric Categories 40
   Metric Value Types 41
   Baselines 42
   Default Metric Collection Settings 43

8 Log and Configuration Event Tracking 45
   Hyperic Log Tracking Overview 45
   Hyperic Configuration Tracking Overview 46
   How to View Event Data 47

9 Resource Control in Hyperic 49
   Control Action Overview 49
10  Alerts and Alert Definitions  50
    Alerts  50
    Functionality of a Resource Alert  50
    Alert Definition Process  50
    Alerts in the Hyperic User Interface  51
    Fixing and Acknowledging Alerts  51
    Enabling and Disabling Alert Definitions  51
    Introduction to Escalation Schemes  51
    Options for Controlling Alert and Notification Volume  52
    Responding to Alert and Notification Storms  52
    Advanced Alert Functionality in vFabric Hyperic  53

11  SNMP Functionality in Hyperic  55
    Simple SNMP Agent Availability Checks  55
    Monitor SNMP Devices and Hosts with Built-In Plugins  55
    Build Vendor-Specific SNMP Plugins  55
    Send SNMP Notifications for Alerts  56
    Integrate Hyperic with OpenNMS  56
- About vFabric Hyperic Overview (see page 6)
- Use of "Hyperic," "vFabric Hyperic," and "Hyperic HQ" (see page 7)
- Introduction to Hyperic Monitoring (see page 8)
- Resources, Resource Types and Inventory Types (see page 11)
- User Accounts and Roles in Hyperic (see page 25)
- Resource Auto-Discovery Processes (see page 33)
- Metrics and Metric Collection (see page 39)
- Log and Configuration Event Tracking (see page 45)
- Resource Control in Hyperic (see page 49)
- Alerts and Alert Definitions (see page 50)
- SNMP Functionality in Hyperic (see page 55)
About vFabric Hyperic Overview

vFabric Hyperic Overview is an introduction to VMware® vFabric™ Hyperic® functionality. It covers basic concepts such as monitoring, resources and resource types, autodiscovery, alerts, and metric collection.

Intended Audience

vFabric Hyperic Overview is intended for anyone who will establish, manage, or use a vFabric Hyperic environment.

Last updated October 9, 2012.
Use of "Hyperic," "vFabric Hyperic," and "Hyperic HQ"

vFabric Hyperic is the commercial version of the Hyperic product, available as part of vFabric Suite (previously known as "vFabric Platform") and as a standalone product. Hyperic HQ is the open-source version of Hyperic. The products have many similarities, although vFabric Hyperic has a number of features that the open source offering does not.

As a general rule in documentation text, "Hyperic" is used in contexts that apply to both vFabric Hyperic and Hyperic HQ. "vFabric Hyperic" and "Hyperic HQ" (or "HQ") are used in contexts that apply specifically to one product or the other. Exceptions include titles of Hyperic guides, which refer to vFabric Hyperic, and some "HQ" references in the user interface and code that apply to both products.
Introduction to Hyperic Monitoring

Topics marked with * relate to features available only in vFabric Hyperic.

Hyperic provides proactive performance management with complete and constant visibility into applications and infrastructure. It produces more than 50,000 performance metrics on more than 75 technologies at every layer of the stack. At startup, Hyperic automatically discovers and adds new servers and VMs to inventory; configures monitoring parameters; and collects performance metrics and events. Hyperic helps you reduce operations workload, increase your company's IT management maturity level, and drive improvements in availability and infrastructure health.

- Main Components (see page 8)
  - Hyperic Agent (see page 8)
  - Hyperic Server and Hyperic Database (see page 8)
  - Hyperic User Interface (Hyperic Portal) (see page 8)
- Hyperic Monitoring and Management Features (see page 9)
- Key Hyperic Facts for the New User (see page 10)

Main Components

Main components of Hyperic include Hyperic Server, Agent, Database, and the Hyperic User Interface, also known as the Hyperic Portal.

Hyperic Agent

You run a Hyperic Agent on each physical or virtual machine that you want to manage with Hyperic. Agents auto-discover the software components running on the machine, and periodically re-scan the platform for changes in its configuration. Hyperic Agents gather performance and availability metrics; perform log and event tracking; and allow you to perform control functions, such as starting and stopping servers. Agents send the inventory and performance data they collect to a central Hyperic Server.

Hyperic Server and Hyperic Database

The Hyperic Server receives inventory and metric data from Hyperic Agents and stores it in the Hyperic database. The server provides facilities for managing your software inventory. It implements the Hyperic inventory and access model, which allows you group your software assets in useful ways that ease the process of monitoring and management. The Hyperic Server detects when alerts fire, and performs the notifications or escalation processes you define. It also processes actions that you initiate through the Hyperic Portal or Hyperic Web Services API. Hyperic Server also provides authentication services, using an internal engine or an external authentication service.
Hyperic User Interface (Hyperic Portal)

The Hyperic browser-based user interface, sometimes referred to as the Hyperic Portal, is a configurable, extendable user interface for monitoring and analyzing performance and availability. Key features of the user interface include:

- **Dashboard.** The Dashboard is the first page you see when you open the Hyperic user interface. The Dashboard contains multiple portlets — Recent Alerts, Availability, Problem Resources, and so on — each of which presents a summary view of events and resources of interest. You can tailor the Dashboard on a per-user basis. You can remove and rearrange portlets on your Dashboard, and tailor what data a portlet presents.

- **Resource Hub.** You go to the Resources tab, often referred to as the Resource Hub, to browse to specific resources, view resource properties, view metric data and charts, and initiate resource control actions. Hyperic administrators use the features of the Resource Hub to configure resources for monitoring and set up alert definitions.

- **Global Monitoring Views.** The Hyperic user interface contains the following pages that present deployment-wide resource monitoring results:
  - **Operations Center.** Broad view of deployment health, including alerts, events, and currently down resources.
  - **Alert Center.** Deployment-wide view of alert activity and alert definitions.
  - **Event Center.** Deployment-wide view of log events, configuration events, and alerts.
  - **Nagios Data.** Available in deployments that have integrated Hyperic with Nagios.
  - **Currently Down.** List of currently unavailable resources.

- **Resource Type Views.** Some pages in the Hyperic user interface are specific to a particular resource type; for example, the vSphere View for vCenter and vCenter-managed resources, and the GemFire View for monitoring components of a vFabric GemFire distributed caching environment.

Hyperic Monitoring and Management Features

Hyperic includes the following key monitoring and management capabilities:

- **Resource discovery.** The Hyperic Agent managing a platform automatically discovers the resources and software on the platform. The agent discovers key platform properties, such as architecture, RAM, CPU speed, IP address, and domain name. The agent uses resource plug-ins to discover software products — for example, Web servers, application servers, database servers, and so on — running on the platform. Hyperic categorizes the resources it discovers into inventory types. Hyperic's inventory model is fundamental to how Hyperic makes sense of a large number of software resources and presents information about software resources components in a useful way. Discovered resources are presented in the Hyperic user interface; an authorized user explicitly imports them into inventory.

- **Metric collection.** After a resource is added to inventory, and (if necessary) configured for monitoring, the Hyperic Agent starts collecting metrics for the resource. Agents collect a standard set of metrics that reflect availability, performance, utilization, and throughput for each supported resource type. An authorized user can tailor metric collection from the Hyperic user interface.

- **Event tracking.** Hyperic can monitor log and configuration files and record events of interest for most server types. For example, you can track user logins, Windows registry key changes, error logs, configuration files, and so on. You configure event tracking for an individual resource.

- **Resource control.** You can use Hyperic for remote control and administration of your software resources. Available control actions vary by resource type. For example, for an application server, you can perform tasks such as starting, stopping, and garbage collection. For a database server, you can perform analysis or housekeeping functions.
- **Alerting and notification.** You can set alerts on metrics and configure actions for Hyperic to perform when an alert fires. When an alert fires, Hyperic can respond in a variety of ways: it can issue email notifications, set SNMP traps, perform a control action, or issue a communication to another management system. You can define a sequence of responses to a fired alert — an escalation scheme — to ensure that problems do not escape notice.

- **Live data.** Hyperic provides *Live Exec* views for all platform types. You can run a variety of real-time system commands to obtain live system status.

### Key Hyperic Facts for the New User

The resources you can view and your permission levels are governed by your *role*.

Some resources need to be configured for monitoring. Although Hyperic starts monitoring most resources as soon as they are added to inventory, for certain resource types, some configuration is required to enable the agent to start collection metrics. For example, the JMX URL and credentials have to be defined in Hyperic for the agent to be able to monitor via JMX.

The metric collected for a resource are governed by the default metric collection setting for the resource type.
Resources, Resource Types and Inventory Types

An individual managed resource is classified in Hyperic inventory as an inventory type and as a resource type. An inventory type relates to a software dependency hierarchy, most notably, Hyperic's platform - server - service hierarchy. Groups and applications are examples of inventory types. A resource type relates to the "brand" or vendor associated with a resource.

- Resources and Resource Categories in Hyperic (see page 11)
- About Platforms, Servers, and Services (see page 14)
- About Groups in Hyperic (see page 17)
- About Applications in Hyperic (see page 22)

Resources and Resource Categories in Hyperic

Resources in the Hyperic Inventory are hierarchically related. A resource is classified as an inventory type (platform, server, service, group, or application) and as a resource type that identifies the brand of the inventory type (a Win32 platform, a JBoss 4.0 server, and so forth).

Inventory Type

A resource's inventory type is the first level of classification that Hyperic applies to resources. Inventory types serve two purposes:

- Resource hierarchy. Several inventory types identify where a resource fits into a resource hierarchy. All Hyperic resources are classified as one of the following inventory types.
  - platform - usually corresponds to a machine running an operating system
  - server - a software product running on an operating system, for instance a database or application server
  - service - an integral component of a platform or server, for instance, a file server mount, database table, or a connection pool.
Grouped resources. There are two inventory types that correspond to multiple individual resources. You group resources for a variety of reasons: to monitor a set of like or related resources in aggregate; to administer or control like resources at the group level instead of individually; and, in vFabric Hyperic, for resource access control. There are two inventory types that are named sets of other resources:

- group
- application

In summary, “inventory type” classifies a resource as a platform, server, service, group, or application. The term “inventory level” refers to inventory types that fit into a hierarchical structure - platforms, servers, and services.

**Resource Type**

Each individual resource (every resource that is a platform, server, or service) in Hyperic inventory has a *resource type* that indicates what kind of platform, server, or service it is. For example,

- The resource type of a Windows system (whose inventory type is “platform”) is "Win32"; the resource type of a Linux system (whose inventory type is also platform) is "Linux".
  - For clarity, Hyperic documentation refers to resource types that correspond to platforms - like "Win32" and "Linux" - as platform types.
- The resource type of a JBoss 4.0 instance (whose inventory type is "server") is "JBoss 4.0"; the resource type of a WebLogic 9.1 instance (whose inventory type is also server) is "WebLogic 9.1".
  - For clarity, Hyperic documentation refers to resource types that correspond to servers - such as "JBoss 4.0" and "WebLogic 9.1" - as server types.
- The resource type of a Jboss entity EJB (whose inventory type is "service") is "JBoss 4.0 Entity EJB"; the resource type of a WebLogic EJB (whose inventory type is also service) is "WebLogic 9.2 Entity EJB".
  - For clarity, Hyperic documentation refers to resource types that correspond to services - such as "JBoss 4.0 Entity EJB" and "WebLogic 9.1 Entity EJB" - as service types.

In summary, "resource type" classifies a resource as a particular type of platform, server, service.

**Platform Server Service Hierarchy**

In Hyperic, platforms, servers, and services are hierarchically related.

- A platform is usually a machine its operating system, with a Hyperic Agent running on it. There are also platform types for virtual and network hosts.
- A server is a software product that runs on a platform.
- A service is an resource that is integral to, or runs upon, a platform or server. Whether the resource is at the platform or server level, in Hyperic it is a "service". Note, however that services associated with a platform are usually referred to as a platform services.

Hyperic auto-discovers most platform, server, and service types and populates the Hyperic database with key information about each discovered item, and its relationship with other resources.

**Graphical View of a Resource Hierarchy**

The diagram below illustrates a specific platform-server-service hierarchy. (Only a subset of the servers and services in the hierarchy are shown.) The label for each resource indicates its type. The hierarchy consists of:

- a platform of whose type is "MacOSX"
- two platform services, whose types are "File System" and "CPU"
- two servers, whose types are "JBoss 4.2" and "Tomcat 6.0"
- four services (that run in servers) whose types are "JMS Topic", "JMS Destination", "Webapp", and "Connector".

Platform Hierarchy in Hyperic User Interface

The screenshot below is the Monitor tab for the platform whose hierarchy is partially illustrated in the previous section. Note that the Resources panel shows the currently selected resource's immediate "relatives". For the selected platform, the Resources panel lists the platform services and the servers that run on the platform.
About Platforms, Servers, and Services

Topics marked with * relate to features available only in vFabric Hyperic.

- Platforms (see page 14)
  - Operating System Platforms (see page 14)
  - Virtual and Network Platforms (see page 14)
- Servers (see page 15)
- Services and Platform Services (see page 15)

This page describes the fundamental inventory types in Hyperic: platforms, servers, and services — any individual resource instances has one of these types. For information about inventory types that are configurable collections of other resources — groups and applications — see About Groups in Hyperic (see page 17) and About Applications in Hyperic (see page 22).

Platforms

There are two major kinds of platforms in Hyperic.

Operating System Platforms

An operating system platform is a computer and the operating system that runs on it. The Hyperic Agent auto-discovers operating system platform using Hyperic's system plugin. You cannot manually add an operating system platform to inventory. Hyperic supports these operating system platform types:

- AIX
- FreeBSD
- HPUX
- Linux
- MacOSX
- Solaris
- Unix
- Win32

Virtual and Network Platforms

Hyperic supports a variety of platform types that do not map to an individual physical machine running a traditional operating system. These include:

- Resources that a Hyperic Agent monitors remotely over the network, such as for network hosts and devices,
- Virtual resources such as VMware vSphere hosts and VMs, and
- Distributed sets of resources, such as GemFire Distributed Systems.

The Hyperic Agent does not automatically discover and monitor virtual and network platforms — typically you create such platforms manually (using the New Platform command on the Tools menu in the Resource tab of the Hyperic user interface), or at a minimum, supply resource properties data that enable the agent to manage them. These are the virtual and network platform types that Hyperic supports:

- Cisco IOS
- Cisco Pixos
- GemFire Distributed System
- NetApp Filer
Network Device
- Network Host
- VMware vSphere Host
- VMware vSphere VM
- Xen Host

For information about creating an un-discoverable platform, see Create a Platform.

Servers

In Hyperic, a server is a software product that runs on a platform. Servers provide a communications interface and perform specific tasks upon request. Examples of server types include Tomcat, JBoss, and Exchange. The Monitoring Defaults page in Hyperic’s Administration tab lists all of the server types that Hyperic supports.

Most server types are auto-discovered by a server type-specific Hyperic plugin. If the plugin that manages a server does not support auto-discovery, or if auto-discover of a server fails, you may need to manually create a server, as described in Create a Server.

The screenshot below shows the Monitor tab for a server. The Resources panel for the server lists its child services and parent platform.

Services and Platform Services

In Hyperic, a service is a software component dedicated to a particular task that runs on a server or platform. A service that runs on a server is referred to as a service. A service that runs on a platform is referred to as a platform service.

The resource plugin that discovers a platform or server also discovers key services — such as CPUs, network interfaces, file systems, and so on — running on the platform.

In addition, an authorized user can explicitly configure a platform service on a platform to serve as a proxy for a resource the Hyperic Agent can monitor over the network, for example, a DNS or POP3 service. For more information see Create a Platform Service.
Services that run on a server can be either an internal component of the server (for instance, “Weblogic Admin 9.2 Entity EJB service”) or a deployed item (“CustomerEntityEJB”).

The Monitoring Defaults page in Hyperic’s Administration tab lists the service and platform service types that Hyperic supports.

The screenshot below shows the Monitor tab for a service. The Resources panel for the service lists its parent server.
About Groups in Hyperic

- Resource Groups in Hyperic (see page 17)
- Compatible Groups (see page 18)
- Mixed Groups (see page 19)
- Autogroups (see page 20)
  - View a List of Autogroups on a Resource (see page 20)
  - View Monitor Tab for an Autogroup (see page 20)
  - View Monitor Tab for a Member of an Autogroup (see page 21)

In Hyperic, a group is an inventory type that is a collection of other inventory resources. This page describes the purpose of groups in Hyperic and different types of groups you can create.

For information about creating groups, see Configure and Manage Resource Groups.

Resource Groups in Hyperic

In the Hyperic inventory model, a group is named set of other inventory resources. Grouping resources is useful for:

- **Monitoring a set of homogeneous or related resources in aggregate** - Groups enable role-specific monitoring views, or views that reflect the purpose or business need that the set of resources satisfy. In an environment with thousands of resources, viewing availability and performance data at the group level reduces the clutter in the user interface.

- **Automating resource operations and control** — You can perform control actions on a group of like resources with a single command.

- **Controlling access to resources** — Groups are fundamental to vFabric Hyperic's role-based access control. A Hyperic role specifies permissions to the resources in the groups associated with the role. Resources can only be associated with a role at the group level.

**Note:** When you create a group in vFabric Hyperic, you can designate it as "private". Private groups are invisible to other users, including admin users. You can share a private group by associating it with a role.
Compatible Groups

A compatible group is a user-configured set of inventory items of the same type, for example “JBoss 4.x” servers, or “Linux” platforms.

Using compatible groups, you can ease the effort for operations tasks for a large population of like resources - you can perform the same control action on all members of a compatible group with a single command, on a scheduled or ad hoc basis.

Compatible groups enable richer monitoring. For instance, you can view metrics in aggregate across some or all resources of the same type. In vFabric Hyperic, you can set alerts on compatible groups. Group alerts fire based on the percentage or number of members that meet an alert condition.

The screenshot below is the Monitor tab for a compatible group. Note:

- The Indicators panel charts the aggregate values for metrics across all group members.
- The Resource panel lists the member of the group, and the platforms that host group members.
- The Control tab is present, because the selected group supports control actions.
- The Alert tab is present, because vFabric Hyperic supports alerts on compatible groups.
Mixed Groups

Mixed groups contain inventory resources that are of different types.

Mixed groups are useful in implementing access control policies - for instance, for a set of resources from the same vendor, or that are hosted for a particular customer. Mixed groups do not have a common measurement and control profile. The metrics available naturally vary for different types of resources for instance, you monitor free memory for a CPU, but not for a database table. For similar reasons, mixed groups do not support group control actions.

There are three basic sub-types of mixed groups, which vary in terms of their membership. When you browse mixed groups in Hyperic, the “Group Type” column shows each group’s sub-type:

- **Mixed Group - Platforms, Servers, & Service.** If your service level agreements vary by customer, you could configure this sort of mixed group to contain all of the resources hosted for CustomerA, and name it accordingly. The “CustomerA” group might include multiple Linux platforms, each running Tomcat servers and a variety of deployed EJBs and servlets.

- **Mixed Group - Groups.** This type of mixed group, a kind of “supergroup”, is made up other groups. For example, a regional manager might use a mixed group that contains many customer-specific groups (like the “CustomerA” group above) to monitor availability and other metrics from a territory perspective.

- **Mixed Group - Applications.** This type of mixed group is made up of multiple applications. For example, a line-of-business manager might want to assess and monitor operations at the product line level.

The following screenshot is the **Inventory** page for a mixed group. Note that no **Monitor** or **Control** or **Alert** tab is present, because these functions are not supported for a mixed group.
**Autogroups**

An autogroup is a set of resources of the same type with the same parent resource. As the term implies, an autogroup is not explicitly configured. HQ automatically creates an autogroup to contain all of the resources of the same type on the same platform or server. An autogroup is named for the type of resources it contains. For instance, an autogroup that contains the CPUs on a platform is called “CPU”.

**View a List of Autogroups on a Resource**

You can see the autogroups on a resource on its Monitor tab. This is the only way to see and navigate to an autogroup and its member resources. An autogroup name is only unique in the context of its parent resource.

The name of the autogroup is prefixed with a blue folder-like icon 📂. In the screenshot below, there are four autogroups in the “Platform Services” section: “CPU”, “FileServer Mount”, “HTTP”, and “NetworkServer Interface”. The Availability icon for an autogroup indicates the availability of the group.

![Screenshot of Autogroup](image)

**View Monitor Tab for an Autogroup**

When you click an autogroup in its parent's Resources panel, the Monitor tab for autogroup appears. The Resources panel lists the individual resources in the group, and indicator metrics are shown for the group as a whole.
View Monitor Tab for a Member of an Autogroup

When you click an resource name in Resources panel for the parent autogroup, the Monitor tab for that resource appears, and indicator metrics are shown for that resource.
About Applications in Hyperic

Topics marked with * relate to features available only in vFabric Hyperic.

- Introduction to Applications in Hyperic (see page 22)
- Inventory Tab for an Application (see page 23)
- Monitor Tab for an Application (see page 23)

In Hyperic, an application is an inventory type that is a collection of other inventory resources. This page describes the purpose of applications in Hyperic, and key application views in the Hyperic user interface.

Note: For information about creating an application in Hyperic, see Create and Manage Applications.

Introduction to Applications in Hyperic

In Hyperic, an application is a collection of services that together fulfill a single business purpose. This concept reflects the idea that an application, from the business point of view, comprises many different pieces, and those pieces are usually distributed across different platforms and provided by different servers. Thus you can manage your infrastructure from an application — as opposed to a hardware — point of view.

In Hyperic, an application is an inventory type, configured by an authorized user. An application is a set of selected services, usually running in different servers on multiple platforms, that together fulfill a single business purpose. Configuring applications enables you to manage your infrastructure from an application — as opposed to a hardware — perspective.

Hyperic Visibility into Instrumented Java Applications

The Hyperic Agent can auto-discover and manage Java application services via Model MBeans that adhere to a specified ObjectName naming convention and expose a specified set of service data. This enables deeper visibility into application health: you can monitor application services along with the hosting application server and its internal services. For more information, see Java Applications.

Note: Although instrumentation provides deeper visibility into Java application health, it is not required for application monitoring.
Inventory Tab for an Application

The screenshot below shows the Inventory tab for the application. Note:

- This is the tab you use to add services to an application.
- The "Service Counts" section shows the total number of services in the application, and the number of each type.
- The "Services" section lists key information for each service in the application.
- You can define and view the dependencies between services by clicking View button in the "Dependencies" column.

Monitor Tab for an Application

The screenshot below show the Monitor tab for an application. Note that:
- The **Resources** panel on the left side of the page lists the services in the application, grouped by type.
- The **Indicator** panel charts the aggregated values for selected metrics that are available for the services in the application and the servers where they run. The user that configures the application can choose the metrics to display as indicators.
User Accounts and Roles in Hyperic

Topics marked with * relate to features available only in vFabric Hyperic.

- Authentication (see page 26)
- User Accounts (see page 26)
  - About User Accounts in vFabric Hyperic (see page 26)
  - About User Accounts in Hyperic HQ (see page 26)
  - Built-in Accounts (see page 26)
  - User Account Creation (see page 27)
- Roles in vFabric Hyperic (see page 27)
  - Permission Matrix: Grants Access to Types (see page 28)
    - About Permission Levels (see page 28)
    - Permission Tips (see page 29)
    - How HQ Validates Platform-Server-Service Permission Level Assignments (see page 29)
- Groups: Grant Access to Specific Resources (see page 29)
- Alert Calendar: Enable Shift-Based Notifications (see page 30)
- Built-in Roles (see page 31)
  - SuperUser (see page 31)
  - Guest Role (see page 32)
- Role Creation (see page 32)

This page is an overview of users and roles in VMware vFabric™ Hyperic®.
**Authentication**

Hyperic Server encrypts user passwords using an encryption key you supply during installation. Note however, that Hyperic Server does **not** have a strength-of-password policy, or a lockout policy for failed login attempts.

The best practice is to integrate Hyperic with your existing enterprise directory.

---

**Benefits of External Authentication**

In addition to security benefits, integrating vFabric Hyperic with LDAP or Active Directory can streamline user setup: Hyperic automatically creates a Hyperic account for a user authenticated via LDAP and (if LDAP groups and Hyperic roles are named appropriately) assigns the user to any Hyperic roles based on the user's LDAP group assignments.

---

For information about integrating Hyperic with LDAP or Active Directory, or Kerberos, see Configure LDAP Properties and Configure Kerberos Properties.

**User Accounts**

**About User Accounts in vFabric Hyperic**

In vFabric Hyperic, a user account specifies the user's name, username, and contact information, including the email and SMS addresses for receiving alert notifications.

A vFabric Hyperic user account is associated with one or more **roles**, the mechanism by which resource access and associated permissions are granted to users. Note that in vFabric Hyperic, to have access to resources, a user account must be assigned at least one role to which resources are assigned.

For information about roles, see Roles in vFabric Hyperic (see page 27).

**About User Accounts in Hyperic HQ**

In Hyperic HQ, a user account specifies the user's name, username, and contact information, including the email and SMS addresses for receiving alert notifications.

Roles are not supported in Hyperic HQ — all users have all permissions to all resources in inventory. Similarly, any Hyperic HQ user has the permission to create other users.

**Built-in Accounts**

There is one built-in user account in Hyperic HQ, and two in vFabric Hyperic.

- **hqadmin** — Both Hyperic HQ and vFabric Hyperic have a built-in hqadmin account, which can administer the Hyperic Server. In vFabric Hyperic, the hqadmin account has the **superuser** role, and in addition to permission to administer the Hyperic Server, can:
  - Assign alert definitions to resource types
  - Modify role-based dashboards
- **guest** — vFabric Hyperic has a built-in guest account, which, when enabled, allows anonymous, view-only access to the Hyperic user interface. The guest user has the built-in guest role. Note that the guest role provides View permissions for all types, but unless resource groups are assigned to the role, anonymous users will not be able to view any resources. To allow anonymous users to view resources you must enable the guest account and assign the groups of resources you wish to expose to the guest role.

If desired, you can grant the guest role access to all resources without assigning groups to the role. To grant the guest role full access to all resources, insert the following row into the database.

```
INSERT INTO EAM_ROLE_RESOURCE_GROUP_MAP VALUES (2, 1);
```

### User Account Creation

In Hyperic HQ, you create an account for each user, as described on Create and Manage User Accounts. All Hyperic HQ user account data is stored in Hyperic database.

If you integrate vFabric Hyperic with LDAP or Active directory, you do not have to pre-configure Hyperic user accounts. Instead, a first time user can log on to vFabric Hyperic with LDAP credentials and Hyperic will prompt for required account information and create the user account. The username for an automatically created user account is the user's LDAP username, prefixed by org/.

If you do not configure vFabric Hyperic to work with an external authentication system, you must manually create user accounts, as described in Create and Manage User Accounts.

Each user account in vFabric Hyperic must have one or more roles. Note that if you integrate vFabric Hyperic with your authentication system, Hyperic can automatically assign Hyperic roles to LDAP-authenticated users: if there is a Hyperic role with the same name as an LDAP group to which a user is assigned, Hyperic will assign the matching role in Hyperic to the Hyperic user account.

### Roles in vFabric Hyperic

*Available only in vFabric Hyperic*

In vFabric Hyperic, every user is assigned one or more roles. Roles enable:

- **Access control** - A role defines what resources the users added to the role - *role users* - can access, and the types of operations - view, edit, create, and so on - they can perform on those resources.

- **Alert notification** - A role with users but no resource groups assigned to it can serve simply as a distribution list for alert notifications. Role-based notification makes it easier to maintain alert definitions, and enables shift-based alert notifications. For around-the-clock operations, you can define multiple roles, with complementary alert calendars that specify when role users are on duty. If you assign the several complementary roles as recipients for the same alert, when the alert fires, HQ will send notifications only to the role with currently active calendar.

- **Role-Specific Dashboards** - When you create a role, HQ creates a new Dashboard for the role, which you can customize to meet the needs of role users.
The sections below describe the information you define for a role in vFabric Hyperic.

Permission Matrix: Grants Access to Types

The permission matrix for a role defines the level of access that role users have to configurable items in vFabric Hyperic. There are several types of targets to which you can define a permission level:

- User management types - The permission levels to Users and Roles determines what level of access, if any, role users have to view and manage HQ user accounts and HQ roles.

- Inventory resource types - The permission level to inventory types - Platforms, Servers, Services, Groups, and Applications - controls the level of access, if any, role users have to that inventory type. Note: Granting access to an inventory type does not grant access to specific resource instances.

- Escalations - The permission level for Escalations controls the level of access, if any, role users have to view or manage escalations defined for use in alert definitions.

The screenshot below shows the permission matrix you define for a role.

<table>
<thead>
<tr>
<th>Permissions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource Type</td>
</tr>
<tr>
<td>Users</td>
</tr>
<tr>
<td>Roles</td>
</tr>
<tr>
<td>Groups *</td>
</tr>
<tr>
<td>Platforms</td>
</tr>
<tr>
<td>Servers</td>
</tr>
<tr>
<td>Services</td>
</tr>
<tr>
<td>Applications</td>
</tr>
<tr>
<td>Escalations</td>
</tr>
</tbody>
</table>

* Regardless of permissions selected, all users have the ability to create groups in the system.

About Permission Levels

You assign one of the following permission levels to each type.

- **None** - No access at all to instances of the type.

- **Read-Only** - Allows role users to view instances of the type, but not create, edit, or delete them. For Platforms, Servers, Services, Groups, also enables:
  - **Read-Only** access to alert definitions for the inventory type.

A role with Read-Only permission level does not have permissions to enable/disable/fix/ack alerts or control resources - these capabilities must be explicitly granted.
- **Read-Write** - Allows role users to view and edit instances of the type, but not create or delete them. For **Platforms, Servers, Services, Groups**, also gives:
  - **Full** access to alert definitions for the inventory type,
  - Permission to manage alerts (enable/disable, fix, acknowledge) for the inventory type.
  - Permission to perform supported control operations on resources of the inventory type.

- **Full** - Allows role users to create, edit, delete, and view instance of the type. For **Platforms, Servers, Services, Groups**, also gives:
  - **Full** access to alert definitions for the inventory type.
  - Permission to manage alerts (enable/disable, fix, acknowledge) for the inventory type.
  - Permission to perform supported control operations on resources of the inventory type.

**Permission Tips**

Define a Role's Permission Matrix

For roles that:

- **Add resources to inventory and create alert definitions** - use **Full** or Read-Write permission levels. These permission levels enable a role to also process fired alerts and control resources.

- **Monitor resources, respond to alerts and control resources** - use the **Read** permission level, and then grant Fix/Ack and Control capability, or both. This allows operations staff to respond to alerts, see the details of alert definitions, and perform routine or as-needed resource control tasks but **not** create/modify/delete resources and alert definitions.

- **Need visibility only** - Use **Read** permission level for roles that view and monitor resources, but do not (1) create/modify/delete resources and alert definitions, or (2) response to alerts.

**How HQ Validates Platform-Server-Service Permission Level Assignments**

vFabric Hyperic does a bottom-up validation of the permission levels a role grants to Platforms, Servers, and Services.

A role with **Full** access (which enables resource deletion) to an inventory type must have at least **Read-Only** access to the parent type (if there is one) and Full to the child type (if there is one).

For example, **Full** access to Servers requires at least Read access to Platforms and Full access to Services.

**Groups: Grant Access to Specific Resources**

In addition to defining a permission matrix for a role, you assign one or more groups that contain individual resource to the role. (Assigning a group that contains other groups or a group of applications will **not** grant the role permissions to the resources in the groups or applications.) Together, the permission levels and groups defined in the role determine the specific inventory resources that role users can work with.

> If you create a role simply for use in role-based alert notifications, you do not have to assign any resource groups to the role.
Permission levels to **Platforms**, **Servers**, **Services**, **Groups**, and **Applications** define the level of access role users have to each of those inventory types. The operations that a role enables for an inventory type apply only to resources that belong to a group assigned to the role. (You cannot assign individual resources to a role, you must create groups of resources, and assign groups to roles.)

For example, the **Full** permission to **Platforms** granted by a role may only be exercised on platforms that belong to a group assigned to the role. So, a group assigned to a role may well contain resource types to which the role does not grant access.

You can assign the same resource group to multiple roles, and you can assign the same user to multiple roles. This allows for the fact that different users may need different levels of access to the same resources. For instance, you can create one role for users that need **Read-Only** access to the members of a resource group, and another for users that need **Full** permission, and assign the same resource groups to both roles.

**Alert Calendar: Enable Shift-Based Notifications**

An Alert Calendar is an optional component of a role that builds on the notion of role-based notification. In role-based alert notifications, the notification recipient is a role - notifications are sent to all users with the role. An Alert Calendar for role defines the time periods during a work week that role users are on duty. You can define multiple roles to span the week - each with a different availability calendar, and assign all of the complementary roles as the notification recipients. In this case, Hyperic Server will send alert notifications only to the role that is currently on-duty, based on the alert calendars defined in the roles.

The screenshot below shows the alert calendar you can define for a role.
Built-in Roles

vFabric Hyperic has two built-in roles, which are described in the sections below.

SuperUser

The screenshot below is the permission matrix for the Hyperic SuperUser. The built-in hqadmin account has the SuperUser role.
**Guest Role**

The screenshot below is the permission matrix for the Hyperic SuperUser. The built-in guest account has the Guest role.

**Role Creation**

For instructions on creating a role see Create and Manage Roles in vFabric Hyperic.
Resource Auto-Discovery Processes

Topics marked with * relate to features available only in vFabric Hyperic.

This page describes VMware vFabric™ Hyperic® processes for discovering resources running on a platform.

- Hyperic Auto-Discovery Processes (see page 33)
  - Default Scan (see page 33)
  - File Scan (see page 35)
  - Runtime Scan (see page 36)
- About Auto-Inventory IDs and InstallPath (see page 37)
- What the Agent Can and Cannot Discover (see page 37)
- How Discovered Resources Get into Hyperic Inventory (see page 37)
- What to Do After Adding New Resources to Inventory (see page 37)

Hyperic Auto-Discovery Processes

Most resources are automatically discovered by the Hyperic Agent running on a platform. The agent scans a platform to discover new resources, and resources whose properties have changed since the last scan. For instance, if a platform's IP address changes, the next scan detects the change.

Under the hood, an agent's auto-discovery capabilities break down to three different types of scan, described in the sections that follow.

Default Scan

A default scan discovers platforms and servers from the process table or Windows registry.

On Unix-like platforms, a default scan checks the process table for processes that match a given pattern. On Windows platforms, a simple registry scan is performed, looking for registry keys that installed products register during their installation process.

A default scan is performed upon agent startup, and every 15 minutes thereafter. A default scan typically does not take long and is not CPU-intensive.

You can also initiate a default scan on-demand for a platform to discover new servers. When you initiate a default scan, you can initiate a file scan, described below, as well.

On-Demand Default Scan Discovers all vSphere Resource Types

When you initiate a default scan on a platform where a vSphere vCenter server runs, it discovers any new vCenter instances, and also discovers all of the ESX hosts and VMs a vCenter instance manages.
File Scan

A file scan discovers servers by scanning the platform's file system for manageable products' installation directories. You can configure what server types to look for and directories to include or exclude for the search. You can initiate a file scan explicitly; when you run a default scan, you can start a file scan at the same time — the agent never runs a file scan automatically.
**Runtime Scan**

A runtime scan discovers servers and services — platform services as well as services that run in a server. The agent performs a runtime scan when a new platform or new servers on a platform are added to inventory. In addition, the agent automatically does a runtime scan once a day.

For information about how to scan a platform on demand and options for configuring and disabling auto-discovery behavior, see Options for Running and Controlling Resource Discovery in *vFabric Hyperic Administration*.

---

**Dynamic Service Type Detection**

The auto-discovery functionality described on this page is governed by the resource plugin that manages a resource type. In addition, the Hyperic Agent can auto-discover and manage Java application services via Model MBeans that adhere to a specified ObjectName naming convention and expose a specified set of service data. This enables you to monitor application services along with the hosting application server and its internal services. For more information, see Java Applications in *vFabric Hyperic Resource Configuration and Metrics*.
About Auto-Inventory IDs and InstallPath

In Hyperic, any resource that is a server type must have an installation path defined as a configuration option. Note that for most server types, the installation path is not required for auto-discovery—servers are typically discovered from a scan of the system process table or Windows registry, and the value of the installation path is set by Hyperic. In fact, the installation path shown in Hyperic for a server often does not map to the actual installation path of the resource.

The primary purpose of a server's installation path, as defined in Hyperic, is in the construction of the resource's auto-inventory identifier. A resource's auto-inventory identifier is an internal identifier for a resource that is used in two ways:

- The Hyperic Agent uses the auto-inventory identifier to ensure that it does not report duplicates of a resource to the Hyperic Server.
- The Hyperic Server uses the auto-inventory identifier to determine whether a resource in an auto-inventory report is a new resource, or an existing resource in inventory.

What the Agent Can and Cannot Discover

The data the Hyperic Agent discovers for a resource type is specified in the XML descriptor for the resource plugin that manages it. All operating system platform types are managed by HQ's system plugin. Most other Hyperic plugins discover multiple versions of a server type and the service types it contains. For example, Hyperic's tomcat plugin manages several versions of the Tomcat server and multiple services within Tomcat.

The plugins that ship with Hyperic are in the SERVER_HOME/hq-engine/hq-server/webapps/ROOT/WEB-INF/hq-plugins directory. If you want to monitor a software product for which Hyperic does not provide a plugin, you can build your own, using support classes available from Hyperic, and deploy it to this directory and to the Hyperic Agent.

There are some resources that the Hyperic Agent cannot discover:

- Non-running resources - The agent cannot discover a resource that is not currently running.
- Remote network services - The Hyperic Agent can manage remote services and devices over supported network protocols, for example HTTP, FTP, or SMTP. For obvious reasons, you explicitly configure a remotely managed resource: you manually add it to inventory, and configure the properties that enable the agent to communicate with it. For more information, see Create a Platform Service in vFabric Hyperic Administration.
- Undiscoverable types - There are a few resource types that the Hyperic Agent can manage but not discover, even on the local platform. For example, you may need to configure the location of some versions of WebLogic Server. For a resource's configuration requirements, see the Configuration help section of the documentation for the resource plugin on HyperForge, the Hyperic community site.

How Discovered Resources Get into Hyperic Inventory

After a scan is completed, new and changed platforms and servers appear in the Auto-Discovery portlet on the Hyperic Dashboard. You explicitly add new and changed platforms to inventory - using controls in the Auto-Discovery portlet, or the Auto-Discovery Results page, which provides more detailed results of what a scan detected on a platform. New and changed services do not appear in the Auto-Discovery portlet or the the Auto-Discovery Results page — you do not need to explicitly add them to inventory. When you add a platform or server to inventory, the associated services are automatically added as well.
What to Do After Adding New Resources to Inventory

After adding a new platform or server to inventory, you might need to:

- Configure the resource to enable monitoring. Configuration options are found on the resource's **Inventory** page.

- Add the resource to one or more new or existing resource groups. You can select groups to assign the resource on its **Inventory** page. Groups are useful for:
  - Resource access control, if you use vFabric Hyperic.
  - Monitoring a group of resources of the same type.

- Add new services to new or existing applications to enable monitoring at the application level.
This page describes how metrics are categorized in VMware vFabric™ Hyperic®, how metric baselines are calculated and used, and default metric collection settings.

- Metric Categories (see page 40)
  - Availability (see page 40)
  - Throughput (see page 40)
  - Utilization (see page 41)
  - Performance (see page 41)
- Metric Value Types (see page 41)
- Baselines (see page 42)
  - Uses for Baselines in Hyperic (see page 42)
  - Baselines in the Hyperic User Interface (see page 42)
  - How a Baseline is Calculated (see page 43)
- Default Metric Collection Settings (see page 43)
Metric Categories

Availability

In Hyperic, a resource is “available” when it is ready to service requests. More specifically, a platform is available if the HQ Server can reach it. For other inventory types, HQ issues a query or a request to the resource to determine its availability. If a resource that is part of an application is unavailable, Hyperic considers the entire application to be unavailable. A managed resource’s availability is displayed as follows:

<table>
<thead>
<tr>
<th>Availability Icon Color</th>
<th>Availability Icon</th>
<th>State</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Green</td>
<td></td>
<td>Up</td>
<td>For an individual resource, indicates that its availability status is “up”. For a group, indicates that none of the group members has availability status of “down”.</td>
</tr>
<tr>
<td>Yellow</td>
<td></td>
<td>Warn</td>
<td>This state is valid for a group only. It indicates that (1) one or more group members, but not all members, have the availability state of “down”, and (2) the remaining members have status “green”.</td>
</tr>
<tr>
<td>Blue</td>
<td></td>
<td>Maintenance/Paused</td>
<td>This icon appears only for a VM or a group of VMs. For an individual VM indicates it is paused. For a group, indicates that all VMs in the group are paused.</td>
</tr>
<tr>
<td>Red</td>
<td></td>
<td>Down</td>
<td>For an individual resource, indicates that its availability status is “down”. For a group, indicates that all members of the group have either availability status “red” or “unknown”.</td>
</tr>
<tr>
<td>Grey</td>
<td></td>
<td>Unknown</td>
<td>For an individual resource, indicates that its availability is unknown. For a group, indicates that the availability of at least one of the members is unknown.</td>
</tr>
<tr>
<td>Blue</td>
<td></td>
<td>Suspended/Powered Off</td>
<td>This icon appears only for a VM or a group of VMs. For an single VM indicates it is suspended or powered off. For a group of VMs, indicates that all VMs in the group are suspended or powered down.</td>
</tr>
</tbody>
</table>

When HQ notifies you that an application is unavailable, you can drill down into the resources that make up that application in order to determine which resource (such as a web server, application server, or database) is causing the availability problem.
Throughput

HQ can measure throughput for each managed resource. For Web servers and application servers, throughput is typically measured as bytes served, bytes received, number of requests, and number of responses over a user-specified period of time (minutes, hours, days). For databases, throughput is typically measured as the number of requests processed or active connections over a specified period of time.

Utilization

Hyperic HQ can measure utilization rates for the platforms and servers that make up an Application. Examples of utilization include number of sessions created and destroyed, number of loaded or reloaded servlets, JVM total, used, and free memory, EJB creates, removes, loads, stores, and so on.

You can examine the capacity of an entire platform and measure individual utilization of the servers on those platforms. Using Hyperic HQ, you can pinpoint underutilized resources by establishing minimum utilization thresholds on a per platform basis. You can also determine where Application bottlenecks occur by examining utilization rates between disk, memory, CPU, and network, and then apply capacity appropriately.

Performance

A variety of metrics are categorized as performance metrics in Hyperic. Performance metrics are often measured in units of time, the milliseconds spent performing a type of operation, or the length of time that a threshold value was reached. Some performance metrics take an integer value - for instance the length of a work queue.

Metric Value Types

- Dynamic — Value may go up or down over time. CPU utilization is an example.
- Static — Value does not change over time. A time stamp is an example.
- Trends Up — Value always increases. For metrics whose values trend upwards, the rate of change is of interest, so Hyperic automatically creates a secondary metric: a per-minute rate measurement. If this rate metric has a defaultOn attribute set to true, the defaultOn attribute for the original metric is set to false (so that only the rate metric will be displayed, not the original metric). To disable an automatically generated rate metric, set its rate attribute to none.
- Trends Down — Value always decreases.
Baselines

Available only in vFabric Hyperic

Baselines — values that represent the norm for resource behavior — help you quickly identify problems with your resources. With baselines, you can automate metric analysis and configure alert conditions based on how a metric varies from baseline values. Hyperic automatically calculates the baseline values for all dynamic metrics.

Uses for Baselines in Hyperic

Baselines can help you provide:

- **Trend Analysis**: The most common use of baselining is as a performance management tool for trending analysis. Using Hyperic, you establish and retain the same metric baseline value over a specific period of time, then include the baseline when you chart the current values of the metric. You can then identify trends that will help you to estimate future performance or needs.

- **Service-Level Management**: To manage service-level agreements, you measure actual performance against agreed-upon minimum service-level values. Using Hyperic, you specify the acceptable high and low values for the metric then include this range of acceptable values when you chart the current values of the metric.

- **Exception Management**: You can monitor application health by defining an alert based on either the baseline, the high, or the low metric values. For example, you can set up an alert that triggers when the metric value is more than 25% of the baseline value.

Baselines in the Hyperic User Interface

Baseline values for a metric are indicated on a chart for the metric. A chart for a metric is displayed when you click a metric's name on the resource's Monitor page - either in the Indicators or Metric Data tab.

A metric chart, like the example shown below, provides two sets of metric statistics:

- **Low, Average, and Peak** - these lines indicate that low, average, and high values for the metric for the current display range.

- **Low Range, Baseline, and High Range** - this lines reflect baseline values that were calculated (or specified) as the expected low, normal, and high values for the metric.
How a Baseline is Calculated

Hyperic continuously and automatically calculates the baselines for dynamic metrics it collects for a resource: it averages the observed metric values over a user-specified time frame. A baseline value for a metric becomes more accurate as more data is collected. Hyperic calculates the baseline values based on the frequency of calculation, the set of metrics to consider, and the minimum number of data points to use for calculation. You can change these values and thereby change how baselines are calculated.

For more information, see Configure Metric Baselines.

Default Metric Collection Settings

Metric collection defaults are set at the resource level and apply to all resources in inventory of that type.

Although you can modify metric collection settings for an individual resource, those settings will be over-written the next time the default metric collect for that resource type is updated. Changes at the resource type level will apply to all resources of the type.
Default metric collection and alert definitions for a resource type can be viewed and edited by an authorized user on the Monitoring Defaults page for a resource type.
This page is an overview of the event tracking capabilities in VMware vFabric™ Hyperic®.

- Hyperic Log Tracking Overview (see page 45)
  - Hyperic Resource Types that Support Log Tracking (see page 45)
  - Supported Log Message Types (see page 45)
  - Log Tracking Configuration Options (see page 45)
  - How to Set Up Log Tracking (see page 46)
- Hyperic Configuration Tracking Overview (see page 46)
  - Event-Based Configuration Tracking (see page 46)
  - Polling-Based Configuration Tracking (see page 47)
  - How to Set Up Configuration Tracking (see page 47)
- How to View Event Data (see page 47)

**Hyperic Log Tracking Overview**

System problems can often be detected or diagnosed from messages generated by operating systems, application servers, network services, or middleware throughout the environment. Hyperic can monitor messages in log files and in memory, and record events in the Hyperic database based on criteria you specify.

**Hyperic Resource Types that Support Log Tracking**

Hyperic supports log tracking for operating system platforms, network services, and most server types. If a resource supports log tracking, its Configuration Properties page contains log tracking configuration options.

**Supported Log Message Types**

Hyperic can monitor and record log events for:

- Log file messages that specify log levels using log4j log levels.
- Events written to Windows Event Logs. The criteria for what events to track varies by product plugin. See plugin documentation for details. For information about the default content of the log messages that Hyperic records for Windows events and how to customize the format, see the documentation for the platform.log_track.eventfmt agent property.
- Network request results for a variety of network services.

**Log Tracking Configuration Options**

You enable and configure log tracking for a resource on its Configuration Properties page. Navigate to the resource's Inventory page, and click Edit in the Configuration Properties section to display the Configuration Properties page.
Log and configuration tracking must be enabled for a resource if you wish to log events for log messages or configuration changes. Event logging is automatic for alerts and control actions. Log tracking configuration options vary somewhat by resource type.

Log tracking options vary by resource type.

**How to Set Up Log Tracking**

For information about configuring log tracking see Set Up Log Tracking for a Resource in *vFabric Hyperic Administration*.

**Hyperic Configuration Tracking Overview**

You can configure Hyperic to log an event when specified files — usually configuration artifacts — associated with a managed resource are modified.

Configuration tracking is supported for most platform and server types; typically not for services.

The Hyperic Agent must be able to read a file to track it — ensure that file permissions are such that the Hyperic Agent can read files you wish to track.

You can base an alert definition for a resource or resource type on a configuration tracking event. For more information see Define an Alert for a Resource and Define an Alert for a Resource Group in *vFabric Hyperic Administration*.

Hyperic 4.6.5 provides two different configuration tracking mechanisms, described in the sections below:

**Event-Based Configuration Tracking**

In Hyperic 4.6.5, a new support class — *org.hyperic.hq.product.FileChangeTrackPlugin* — enables more detailed change tracking than available in previous versions of Hyperic. This plugin tracks the change type ("add", "delete", "modify", or "rename") and the actual changes in text files. In Hyperic 4.6.5 the following plugins have been updated to use *FileChangeTrackPlugin*:

- Tomcat
- Apache
- WebSphere
- WebLogic Server
- JBoss
- PostgreSQL
- mySQL
- Oracle

These plugins provide default Configuration File filters. Note that configuration tracking for these types is disabled by default. You can enable configuration tracking on the Configuration Properties page for a resource. The Event Center screenshot below lists configuration events for a server type managed by a plugin that uses event-based tracking.
Polling-Based Configuration Tracking

Plugins that have not been updated to use FileChangeTrackPlugin use the still-supported org.hyperic.hq.product.ConfigFileTrackPlugin class, which is polling-based, and tracks the time fact that a file was changed, but not details about the change that was made. The Event Center screenshot below lists configuration events for a server type managed by a plugin that uses polling-based tracking.

How to Set Up Configuration Tracking

For information about configuring configuration tracking see Set Up Configuration Tracking for a Resource in vFabric Hyperic Administration.
How to View Event Data

- You can view configuration event data on:
  - The Monitor page for a resource. For more information, see ui-Monitor.CurrentHealth in vFabric Hyperic User Interface.
  - The Event Center page. For more information see ui-Event.Center in vFabric Hyperic User Interface.
Resource Control in Hyperic

Topics marked with * relate to features available only in vFabric Hyperic.

Control Action Overview

In VMware vFabric™ Hyperic®, a **control action** is a resource command the agent can perform on an individual managed resource (usually a server type) or on a compatible group of resources.

Hyperic has built-in resource control functionality for a variety of resource types, generally servers and services — this functionality is implemented in the resource plugin that manages a resource type. For example, Hyperic’s apache plugin enables several tomcat control actions, including "start", "stop", and "restart". Note that a Hyperic plugin may not implement all commands supported by a resource type.

An authorized user — one with access to the resource and permission to perform a control action — can invoke a control action on-demand, schedule an action for a future time, or schedule an action for periodic execution. For more information, see Run Resource Control Actions.

An authorized Hyperic administrator can configure a control action to be initiated as the result of an alert firing. For more information, see Configure a Control Action as an Alert Action.

Hyperic control action functionality is extendable. An authorized Hyperic administrator can configure Hyperic to run custom scripts or executables. See Configure a Custom Control Action. In addition, plugin developers can leverage Hyperic’s base resource control classes to implement control functions that a target managed product supports.

You can monitor the status and history of resource control actions. For more information, see View Control Action Status and History.

HQApi control API

You can use Hyperic’s **control** API to:

- list a resource’s supported control actions,
- run control actions, and
- see control action history

For more information, see HQApi control command.
Alerts and Alert Definitions

Topics marked with * relate to features available only in vFabric Hyperic.

This page is a high level summary of alerting functionality in Hyperic HQ and VMware vFabric™ Hyperic®. See the last section, Advanced Alert Functionality in vFabric Hyperic (see page 53), for a summary of vFabric Hyperic-only features.

- Alerts (see page 50)
- Functionality of a Resource Alert (see page 50)
- Alert Definition Process (see page 50)
- Alerts in the Hyperic User Interface (see page 51)
- Fixing and Acknowledging Alerts (see page 51)
- Enabling and Disabling Alert Definitions (see page 51)
- Introduction to Escalation Schemes (see page 51)
- Options for Controlling Alert and Notification Volume (see page 52)
- Responding to Alert and Notification Storms (see page 52)
- Advanced Alert Functionality in vFabric Hyperic (see page 53)

Alerts

IT teams can use Hyperic’s alerting system to automate and manage IT problem detection and response processes. Hyperic alerting features allow you to:

- Fire and report an alert for a resource when a condition you specify occurs.
- Notify designated personnel or stakeholders of alert events.
- Execute resource control operations when an alert fires.
- Track the resolution status of problems revealed by alerts.
- Analyze alert and alert action history.

Functionality of a Resource Alert

An alert is set of rules you define that tells Hyperic, for a given resource, how to detect a problem and respond to it. You define the rules for an alert: (1) a metric value or event that signals trouble, and (2) what to do when the specified measurement or event is reported. When an alert fires, Hyperic logs it, presents it in the Hyperic user interface, and performs the actions you defined, which can include sending email and SMS notifications, generating OpenNMS traps, or kicking off an escalation - a series of scheduled notifications over a period of time. Additional alert condition and action functionality is described in Advanced Alert Functionality in vFabric Hyperic (see page 53).
Alert Definition Process

You create an alert for a resource, you define an alert definition for it. An alert definition specifies the condition that should initiate alert firing. An alert condition relates to either a metric Hyperic collects or an event Hyperic tracks for the resource. A metric condition specifies a particular metric, and the value or behavior should initiate alert firing - for example "Availability < 100%". An event condition specifies an event - a log event, a configuration file change, a control action - whose occurrence should initiates alert firing. An alert definition can also specifies actions for Hyperic to perform when an alert is fired. You set up alert definitions from the Hyperic user interface, using dialogs and selector lists to specify the condition and actions. The "minimum" alert definition simply specifies the rules for firing. Actions are optional. The alert definition process is described in Define an Alert for a Resource Type.

Note: For information about the using Hyperic's web services API for creating alert definitions, see HQApi alertdefinition command.

Alerts in the Hyperic User Interface

Any fired alert shows up immediately in Hyperic pages that present alert status and history, including the Recent Alerts portlet in the dashboard and the Alerts tab for a resource. Additional alert views are described in Advanced Alert Functionality in vFabric Hyperic (see page 53).

Fixing and Acknowledging Alerts

When an alert is fired, its status is “unfixed”, and will be indicated as such in Hyperic pages until its status is changed to "fixed". Hyperic provides several mechanisms for marking an alert fixed. You can explicitly mark an alert fixed from the Hyperic user interface. If multiple alerts have fired for the same alert definition, you can do a "fix all". Additional alert management capabilities are described in Advanced Alert Functionality in vFabric Hyperic (see page 53).

An alert with an escalation also has an "acknowledgment" status, to indicate that responsible or concerned parties are aware of the problem. When an alert with an escalation is fired, it is "unacknowledged", and remains so until it is explicitly acknowledged from the Hyperic user interface.

Enabling and Disabling Alert Definitions

At any given point in time, an alert definition is either enabled or disabled. When an alert definition is enabled, Hyperic's alerting engine evaluates the alert condition and rules, and fires alerts accordingly. Alerts will not fire for a disabled alert definition. Hyperic provides several mechanisms for enabling and disabling alert definitions.

An alert definition can be enabled:

- by a user explicitly disabling it from the Hyperic user interface
- automatically, if it configured it to disable itself each time it fires, and re-enable itself when the fired alert is marked "Fixed".
- as a result of an authorized user globally disabling all alert definitions from the HQ Server Settings page.

An alert definition can be disabled:

- temporarily, as a step in an escalation
- automatically upon firing, if it configured it to disable itself each time it fires, and re-enable itself when the fired alert is marked "Fixed".
- as a result of an authorized user globally enabling all alert definitions from the HQ Server Settings page.
Introduction to Escalation Schemes

An escalation is a type of alert action; it is a predefined sequence of notifications steps that starts automatically when alert fires. An escalation can define numerous steps to perform over whatever duration you choose. When the alert is marked "fixed" Hyperic stops the escalation. You create an escalation in the Hyperic Administration tab. You assign an escalation to an alert definition using the Escalation tab on the Alert Definition page.

There are several benefits to using escalation:

- **Prevent redundant alerts** — When an alert kicks off an escalation, Hyperic effectively disables the associated alert definition - preventing a sequence of additional alerts for the same problem. The alert definition remains inactive until the escalation ends. An escalation configured to repeat itself ensures that redundant alerts will be prevented even if the escalation ends before the triggering problem is resolved.

- **Automate issue management processes** — An escalation automates the process of monitoring and managing problem resolution processes. Thoughtfully configured escalations call attention to "long-running" or broken response processes, and make it harder for issues to fall through the crack.

- **Reduce the effort of managing alert response rules** — Unlike other types of notifications that are defined within an alert definition (for example, the Notify Hyperic Users and Notify Other Recipients actions) an escalation is defined and updated separately. When policies, procedures, or staff assignments change, it is less effort to update one escalation than many alert definitions.

- **Escalations add flexibility to automation** — An escalation has an "acknowledgement" status that enables the automated response to be more flexible and take into account whether or not someone is attending to the problem. You can specify steps to perform based on whether an alert is or is not acknowledged, or based on how long it has been unacknowledged.

Options for Controlling Alert and Notification Volume

The purpose of alerting is to speed the process of detecting and resolving problems. Rapid detection and response can be compromised when multiple alerts fire as a result of the same problem, or if responders are inundated by repetitive alert notifications. Excessive alert and notification are less likely when:

- A given problem or root cause results in one, rather than many, alerts.
- An alert status of "unfixed" indicates a problem that still exists and needs attention, rather than a transient issue occurred, and then went away.
- A single problem doesn't result in a firestorm of redundant notifications.

Hyperic provides a range of options for reducing the volume of alerts, and taking action when alert volume exceeds a manageable level. Prevention is the best strategy.

The best way to prevent redundant alerts is to assign a repeating escalation to every alert definition. An escalation is a series of notifications and a schedule for sending them. When the alert fires, Hyperic issues notifications according to the escalation schedule, and for the duration of the escalation, the alert will not fire again. Only after the escalation ends - because all steps are complete or the alert was marked fixed - can the alert definition fire again. You can set your escalations to repeat until the initiating alert is fixed to prevent redundant alerts for the same triggering condition.

An alternative approach for preventing redundant alerts is to configure each alert definition to disable itself upon firing. If you do, the alert will fire once, disable itself, and re-enable itself when the alert is fixed.
Responding to Alert and Notification Storms

If for some reason the volume of alerts or notifications gets out of control, you can use options on the HQ Server Settings page to immediately and globally:

- **Disable all alert definitions** — No alerts will fire for any resources. Notifications defined in escalations in progress will be completed.
- **Disable all notifications** — No alert notifications will be sent. Any escalations currently in progress stop - any remaining notification steps are not performed.

vFabric Hyperic offers additional features for managing alert and notification volume, as described in the following section.

Advanced Alert Functionality in vFabric Hyperic

vFabric Hyperic provides all the features described in the previous sections, plus these additional alert definition and management features:

- **Multi-condition resource alerts** — In vFabric Hyperic you can define up to three conditions for a resource alert.
- **Additional alert actions** — vFabric Hyperic provides additional alert actions, including
  - SNMP trap — generation
  - Script action — you can configure a script that does custom alert processing or notification, for instance, to share alert information with another management system
  - Control action — operation on a resource, either the resource where the alert fired, or a related resource,
- **Recovery alerts** — In vFabric Hyperic, you can create recovery alerts to streamline your process for responding to alerts. First you create an alert definition that is configured to fire once and then disable itself until fixed. Then you define a recovery alert that fires when the condition that fired the primary alert is no longer true. When the recovery alert fires, it sets the primary alert's status to "fixed" and re-enables the primary alert definition.
- **Resource type alerts** — In vFabric Hyperic you can create an alert definition for a resource type, that will be inherited by all resources of that type. Resource type alerts are useful if want to assign exactly the same alert rules to every resource of the same type, and to be able to enable and disable the alert definition for all of them in one fell swoop.

Best Practice for Resource Type Alert Definitions

Tailoring an inherited alert definition at the resource level is not recommended. A resource type alert definition applies to all resources of that type. If you modify the inherited alert definition for an individual resource, a subsequent update to the resource type alert definition will override the changes made at the resource level.

- **Resource group alerts** — In vFabric Hyperic you can create an alert definition for a compatible group - a group you have defined that contains selected resources, all of which have the same resource type. A resource group alert is useful when you are concerned about how many of a set of resources are having a particular problem - you want to know if 2 out of 10 platforms have high disk utilization, for instance. A resource group alert is evaluated differently than resource alerts or resource type alerts. A resource alert or resource type alert is fires for a specific resource based on monitoring results for that resource only. A resource group alert fires when a metric condition is true for a specified number or percentage of the resources in the group.
- **Hierarchical alerting** — Hierarchical alerting prevents a cascade of alerts from resulting from the same root cause, so that a single problem doesn’t result in alerts firing at multiple levels of the platform-server-service hierarchy. Hierarchical alerting is enabled by default. Hierarchical alerting functionality can be extended to network devices or virtual hosts that platforms depend on by defining dependencies. For more information, see Manage Alert and Notification Volume.

- **Notification throttling** — Notification throttling allows you limit the number of notifications that can be issued over a specified interval; when notification volume reaches the limit, Hyperic stops sending individual notifications, and instead sends periodic rollup notifications, until the volume of alerts with notification actions goes down.

- **Advanced Views for Alert Monitoring and Analysis** — In vFabric Hyperic, the Alert Center presents filterable views of alerts and alert definitions. The Operations Center and presents filterable views of unfixed alerts.
SNMP Functionality in Hyperic

Topics marked with * relate to features available only in vFabric Hyperic.

This page summarizes VMware vFabric™ Hyperic® SNMP-related capabilities.

- Simple SNMP Agent Availability Checks (see page 55)
- Monitor SNMP Devices and Hosts with Built-In Plugins (see page 55)
- Build Vendor-Specific SNMP Plugins (see page 55)
- Send SNMP Notifications for Alerts (see page 56)
- Integrate Hyperic with OpenNMS (see page 56)

Simple SNMP Agent Availability Checks

You can configure an Hyperic Agent to monitor the availability of a remote SNMP agent. You configure a platform service of type "SNMP" on the platform of your choice. The Hyperic Agent queries the remote SNMP service for sysUpTime, and reports the service available if a response is received. For more information, see SNMP Platform Service in vFabric Hyperic Resource Configuration and Metrics.

Like other platform services, an SNMP platform service is limited to availability monitoring. To collect throughput and utilization metrics for SNMP devices, see the following section, Monitor SNMP Devices and Hosts with Built-In Plugins (see page 55).

Monitor SNMP Devices and Hosts with Built-In Plugins

These are Hyperic's built-in capabilities for collecting availability, throughput, and utilization metrics for SNMP devices and hosts.


- Network Host - Hyperic has built-in support for monitoring any SNMP host that implements HOST-RESOURCES-MIB (rfc2790), in addition to IF-MIB (rfc2863) and IP-MIB (rfc4293). (A network host is an SNMP device with storage.) You configure a platform of type "Network Host".

- Cisco IOS - Hyperic has built-in support for monitoring Cisco IOS routers. You configure a platform of type "Cisco IOS". The Cisco IOS platform extends Network Device, adding metrics from CISCO-PROCESS-MIB and CISCO-MEMORY-POOL-MIB.

- Cisco Pixos - Hyperic has built-in support for monitoring Cisco Pixos routers. You configure a platform of type "Cisco Pixos". The Cisco PIXOS platform extends Cisco IOS, adding metrics from CISCO-FIREWALL-MIB.
Build Vendor-Specific SNMP Plugins

You can build your own plugin, leveraging Hyperic's SNMP plugin classes, to monitor specific SNMP device. Such a plugin is XML-only - development of custom plugin classes is not necessary. You write the plugin XML descriptor, point to the device MIB, and specify the inventory properties you wish to discover, and the metrics (OIDs) to collect. For more information see Write an SNMP Plugin in vFabric Hyperic Product Plug-in Development.

Send SNMP Notifications for Alerts

Available only in vFabric Hyperic

If you configure vFabric Hyperic to send SNMP messages to your NMS, you can use SNMP notifications in alert actions or as a step in an escalation.

You define SNMP options for HQ Server in the “SNMP Server Configuration Properties” section of the Administration > HQ Server Settings page. The properties you define specify the SNMP protocol version for communicating with the NMS (v1, v2c, or V3), the type of notification (v1 Trap, v2c Trap, or Inform), and the properties required for the SNMP version you use. After this configuration, you can select the SNMP notification type:

- As an alert action — The notification sent when the alert fires will contain three variable bindings:
  - sysUptimeOID.0 — No configuration is required for this binding.
  - snmpTrapOID.0 — This binding is configured on the HQ Server settings page.
  - A variable binding for the alert data specified in the snmp_trap.gsp alert notification template - the alert definition name and the "short reason" for firing. Note that Alert templates may be customized, as described in Tailor Alert Notification Templates in vFabric Hyperic Administration.

- As an escalation step — When you configure an SNMP notification as an escalation step, you can specify additional variable bindings. When the escalation step is performed, the trap will contain those variable bindings, along with SysUpTime.0, snmpTrapOID.0, and a variable binding for the alert data specified in the snmp_trap.gsp alert notification template. For more information and instructions, see Enable SNMP Trap Notifications in vFabric Hyperic Administration.

Integrate Hyperic with OpenNMS

You can export Hyperic platforms as OpenNMS nodes for import to OpenNMS, and send an SNMP trap to OpenNMS as an alert action. For more information see http://support.hyperic.com/display/hyperforge/HQU+OpenNMS.