TIBCO DataSynapse Federator™

User Guide

Software Release 3.0 SP1
February 2015

Important Information

SOME TIBCO SOFTWARE EMBEDS OR BUNDLES OTHER TIBCO SOFTWARE. USE OF SUCH EMBEDDED OR BUNDLED TIBCO SOFTWARE IS SOLELY TO ENABLE THE FUNCTIONALITY (OR PROVIDE LIMITED ADD-ON FUNCTIONALITY) OF THE LICENSED TIBCO SOFTWARE. THE EMBEDDED OR BUNDLED SOFTWARE IS NOT LICENSED TO BE USED OR ACCESSED BY ANY OTHER TIBCO SOFTWARE OR FOR ANY OTHER PURPOSE.

USE OF TIBCO SOFTWARE AND THIS DOCUMENT IS SUBJECT TO THE TERMS AND CONDITIONS OF A LICENSE AGREEMENT FOUND IN EITHER A SEPARATELY EXECUTED SOFTWARE LICENSE AGREEMENT, OR, IF THERE IS NO SUCH SEPARATE AGREEMENT, THE CLICKWRAP END USER LICENSE AGREEMENT WHICH IS DISPLAYED DURING DOWNLOAD OR INSTALLATION OF THE SOFTWARE (AND WHICH IS DUPLICATED IN THE LICENSE FILE) OR IF THERE IS NO SUCH SOFTWARE LICENSE AGREEMENT OR CLICKWRAP END USER LICENSE AGREEMENT, THE LICENSE(S) LOCATED IN THE "LICENSE" FILE(S) OF THE SOFTWARE. USE OF THIS DOCUMENT IS SUBJECT TO THOSE TERMS AND CONDITIONS, AND YOUR USE HEREOF SHALL CONSTITUTE ACCEPTANCE OF AND AN AGREEMENT TO BE BOUND BY THE SAME.

TIBCO, The Power of Now, DataSynapse Federator, GridServer, FabricServer, GridClient, GridBroker, FabricBroker, LiveCluster, VersaUtility, VersaVision, SpeedLink, Federator, and RTI Design are either registered trademarks or trademarks of TIBCO Software Inc. in the United States and/or other countries.

EJB, Java EE, J2EE, and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

TIBCO products may include some or all of the following:

Software developed by the Apache Software Foundation.

Software developed by the OpenSSL Project for use in the OpenSSL Toolkit.

Software developed by Sam Stephenson.

Software developed by Yahoo! Inc.

Code licensed from RSA Data Security.

Spring Framework software.

Code licensed under the Lesser Gnu Public License (LGPL)

Repackaged JBoss Enterprise Middleware open source software.

Product integration with the BEA WebLogic® Server software developed and copyrighted by BEA Systems, Inc. Certain third party software is distributed by the BEA WebLogic Server.

WebLogic® is a registered trademark of BEA Systems, Inc.

WebSphere® is a registered trademark and CloudScape™ is a trademark of International Business Machines Corporation in the United States, other countries, or both.

Software developed using the Eclipse open source development platform.

Software developed by the Massachusetts Institute of Technology.

VMware, the VMware "boxes" logo and design, Virtual SMP and VMotion are registered trademarks or trademarks of VMware, Inc. in the United States and/or other jurisdictions.

Portions copyright 2009 John Resig.

PuTTY, copyright 1997-2009 Simon Tatham.

ANTLR 1989-2004 Developed by Terence Parr.

This product includes software licensed under the Eclipse Public License. The source code for such software licensed under the Eclipse Public License is available upon request to TIBCO and additionally may be obtained from http://eclipse.org/.

This product includes HTML Parser software which is licensed under the Common Public License, v1.0. For a period of time not to exceed three years from the Purchase Date, TIBCO also offers to provide Customer, upon written request of Customer, a copy of the source code for this software.

This product includes Javassist licensed under the Mozilla Public License, v1.1. You may obtain a copy of the source code from http://www.jboss.org/javassist/

This product uses Open Thundergraph. Open Thundergraph is distributed pursuant to the terms of the Lesser General Public License. The source code for Open Thundergraph may be obtained from http://sourceforge.net/projects/thundergraph/. For a period of time not to exceed three years from the Purchase Date, TIBCO also offers to provide Customer, upon written request of Customer, a copy of the source code for Open Thundergraph.

This product includes software developed by the Indiana University Extreme! Lab (http://www.extreme.indiana.edu/).

Contents

Preface

TIBCO DataSynapse Federator™ offers the flexibility and scalability of the public cloud with the security and control of your own data center. It brings the elasticity of cloud computing to your organization – supporting existing solutions within your current infrastructure while automatically scaling resources to meet demand.

Topics

Related Documentation

This section lists documentation resources you may find useful.

TIBCO Federator Documentation

The following documentation is included with Federator in Adobe Acrobat (PDF) format. To view the guides, click on the book icon at the upper right of the interface. The PDF files are also on the Broker at `webapps/federator/doc`. The following documents form the Federator documentation set:

- *TIBCO DataSynapse Federator™ User Guide* Covers general usage and administration of Federator.
- *TIBCO DataSynapse Federator™ Installation Guide* Covers installation of Federator for Windows and Unix.
- *TIBCO DataSynapse Federator™ Release Notes* Read the release notes for a list of new and changed features. This manual also contains lists of known issues and closed issues for this release.

Typographical Conventions

The following typographical conventions are used in this manual.

Table 1 General Typographical Conventions

Convention	Use
TIBCO_HOME *DS_HOME*	Many TIBCO products must be installed within the same home directory. This directory is referenced in documentation as *TIBCO_HOME*. The default value of *TIBCO_HOME* depends on the operating system. For example, on Windows systems, the default value is `C:\tibco`. TIBCO DataSynapse Federator™ installs into a directory within *TIBCO_HOME*. This directory is referenced in documentation as *DS_HOME*. The default value of *DS_HOME* depends on the operating system. For example on Windows systems, the default value is `C:\tibco\datasynapse`.
`code font`	Code font identifies commands, code examples, filenames, pathnames, and output displayed in a command window. For example: Use `MyCommand` to start the foo process.
`bold code font`	Bold code font is used in the following ways: • In procedures, to indicate what a user types. For example: Type `admin`. • In large code samples, to indicate the parts of the sample that are of particular interest. • In command syntax, to indicate the default parameter for a command. For example, if no parameter is specified, `MyCommand` is enabled: `MyCommand [enable | disable]`
italic font	Italic font is used in the following ways: • To indicate a document title. For example: See *TIBCO ActiveMatrix BusinessWorks Concepts*. • To introduce new terms. For example: A portal page may contain several portlets. *Portlets* are mini-applications that run in a portal. • To indicate a variable in a command or code syntax that you must replace. For example: `MyCommand` *PathName*

Table 1 General Typographical Conventions (Continued)

Convention	Use
Key combinations	Key names separated by a plus sign indicates keys pressed simultaneously. For example: Ctrl+C. Key names separated by a comma and space indicate keys pressed one after the other. For example: Esc, Ctrl+Q.
	The note icon indicates information that is of special interest or importance, for example, an additional action required only in certain circumstances.
	The tip icon indicates an idea that could be useful, for example, a way to apply the information provided in the current section to achieve a specific result.
	The warning icon indicates the potential for a damaging situation, for example, data loss or corruption if certain steps are taken or not taken.

Table 2 Syntax Typographical Conventions

Convention	Use
`[ ]`	An optional item in a command or code syntax. For example: `MyCommand [optional_parameter] required_parameter`
`|`	A logical `OR` that separates multiple items of which only one may be chosen. For example, you can select only one of the following parameters: `MyCommand para1 | param2 | param3`

Table 2 Syntax Typographical Conventions (Continued)

Convention	Use

`{ }`	A logical group of items in a command. Other syntax notations may appear within each logical group.
	
	For example, the following command requires two parameters, which can be either the pair `param1` and `param2`, or the pair `param3` and `param4`.
	
	`MyCommand {param1 param2} | {param3 param4}`
	
	In the next example, the command requires two parameters. The first parameter can be either `param1` or `param2` and the second can be either `param3` or `param4`:
	
	`MyCommand {param1 | param2} {param3 | param4}`
	
	In the next example, the command can accept either two or three parameters. The first parameter must be `param1`. You can optionally include `param2` as the second parameter. And the last parameter is either `param3` or `param4`.
	
	`MyCommand param1 [param2] {param3 | param4}`

Connecting with TIBCO Resources

How to Join TIBCOmmunity

TIBCOmmunity is an online destination for TIBCO customers, partners, and resident experts, a place to share and access the collective experience of the TIBCO community. TIBCOmmunity offers forums, blogs, and access to a variety of resources. To register, go to http://www.tibcommunity.com.

How to Access All TIBCO Documentation

After you join TIBCOmmunity, you can access the documentation for all supported product versions here:

http://docs.tibco.com/TibcoDoc

How to Contact TIBCO Support

For comments or problems with this manual or the software it addresses, please contact TIBCO Support as follows.

- For an overview of TIBCO Support, and information about getting started with TIBCO Support, visit this site:

  http://www.tibco.com/services/support

- If you already have a valid maintenance or support contract, visit this site:

  https://support.tibco.com

  Entry to this site requires a user name and password. If you do not have a user name, you can request one.

Chapter 1 Introduction

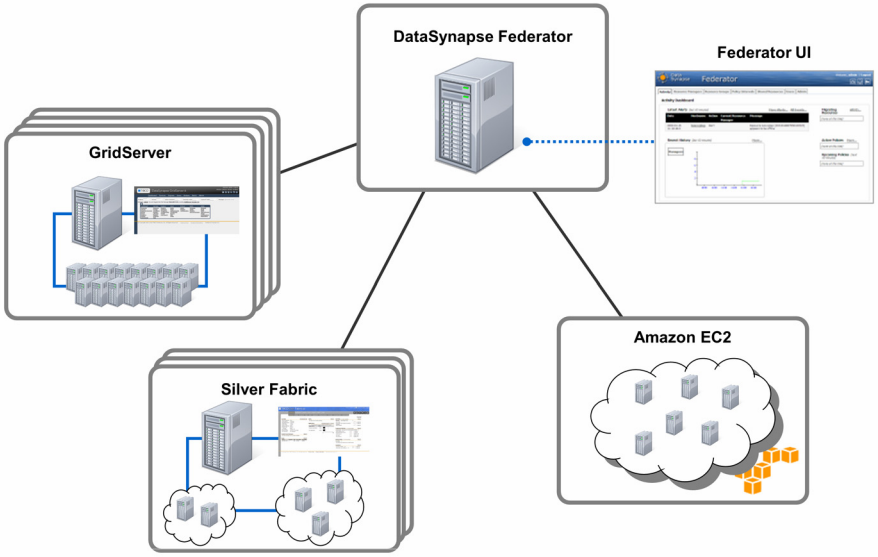
This chapter provides a product overview of Federator and a list of definitions to common terms.

Topics

Product Overview

Federator extends the power of your existing DataSynapse installations by enabling you to move groups of *resources* among your Silver Fabric clouds and GridServer grids. You can do this automatically through a schedule, through a schedule with demand-based rules, or manually.

In addition to using resources running within your datacenter or organization, you can access *virtual resources* running in an external infrastructure such as Amazon's Elastic Compute Cloud (EC2) or VMware.



First, a few definitions:

- **Resource** In Federator, a resource is a DataSynapse Engine Daemon and all of its Engine instances.

- **Resource Manager** Resource Managers represent sources and destinations for resources. There are three types of Resource Manager:
  - **DataSynapse Resource Manager** represents Silver Fabric Brokers and GridServer Directors. They are able to be both the source and target for resources.
  - **EC2 Resource Manager** represents the source of resources from Amazon EC2.
  - **VMware Resource Manager** represents the source of resources from VMware instances.
- **Resource Group** A resource group specifies multiple resources (Engine Daemons) and enables them to be manipulated as a single unit.
- **Static Resource Group** represents resources being explicitly shared by a DataSynapse Resource Manager for use in policies.
- **Dynamic Resource Group** represents resources used in policies, but they are comprised of resources determined at policy startup time by a set of criteria.
- **Virtual Resource Group** represents resources coming from virtual Resource Managers, such as Amazon EC2 or VMware.
- **Policy Interval** A policy interval is the basic building block for scheduling resources in Federator. It defines which resource groups will be assigned to which target DataSynapse Resource Manager at what time of the day.
- **Placement** A policy interval that moves a resource group based upon the schedule.
- **Demand Based** A policy interval that moves a number of resources based upon the demand-based rules.

Chapter 2 Administration

This chapter discusses common Federator administrative tasks such as managing user accounts and modifying Federator's configuration and licensing.

Topics

Managing User Accounts

Federator gives you the option of managing all of your user accounts within Federator itself or of using an external LDAP server to provide account validation. Regardless of which method you configure, user accounts in Federator fall into one or more of the following permission categories (an individual user account may be assigned to one of the following categories):

- *View-Only User* accounts have permission to view most of Federator's functionality.
- *Federator User* accounts have permission to access most of Federator's functionality. They can create, modify, and use Resource Managers, Resource Groups, and Policy Intervals. They cannot manage user accounts or configure parameters on the Admin tab of the Federator Administration Tool.
- *Administrator* accounts have permission to do anything any of the other types of accounts can do. In addition, they can administer user accounts and perform all configurations of Federator on the Admin tab.

LDAP Configuration

If your organization has an LDAP server that you wish to use for authentication of users in Federator, you can configure this on the LDAP page of the Admin tab in the Federator Administration Tool.

Enabling LDAP means that you will not be able to create and edit non-LDAP accounts other than the initial Administrator account. Also, since Federator accesses the LDAP server in a read-only manner, you will need to use other tools to modify users and their attributes on the LDAP server.

If you select the "Enable LDAP Authentication" box, the remaining fields on the page are enabled. You configure LDAP as follows:

Table 3 LDAP Configuration Parameters

Parameter	Description	Default Value
LDAP Server	The URL for the LDAP server. It must include the port number and start with `ldap://`.	`ldap://localhost:389`
Manager Username	The username that Federator uses to log into the LDAP server.	none

Table 3 LDAP Configuration Parameters (Continued)

Parameter	Description	Default Value
Manager Password	The password that Federator uses to log into the LDAP server.	none
Search Subtrees	Check this if you want to search the LDAP trees deeply as opposed to just at the search bases.	selected
User Search Base	The search base to use when locating users in the LDAP directory.	none
User Search Filter	A filter to use when locating users in the LDAP directory.	`(cn={0})`
User Password Attribute	The attribute on the user record which indicates the user's password.	`userPassword`
Group Search Base	The search base to use when locating groups in the LDAP directory.	none
Group Search Filter	A filter to use when locating groups in the LDAP directory.	`uniquemember={0}`
Group Role Attribute	The attribute on the group record which indicates the role name.	`cn`
E-mail Attribute	The attribute on the user record which indicates the user's email address.	`email`
Federator User Group(s)	The group or groups whose members should have Federator User access to the system. (A comma-delimited list if more than one.)	none
Federator Admin Group(s)	The group or groups whose members should have full Federator Admin access to the system. (A comma-delimited list if more than one.)	none
View-Only User Group(s)	The group or groups whose members should have View-Only access to the system. (A comma-delimited list if more than one.)	none

Non-LDAP Configuration

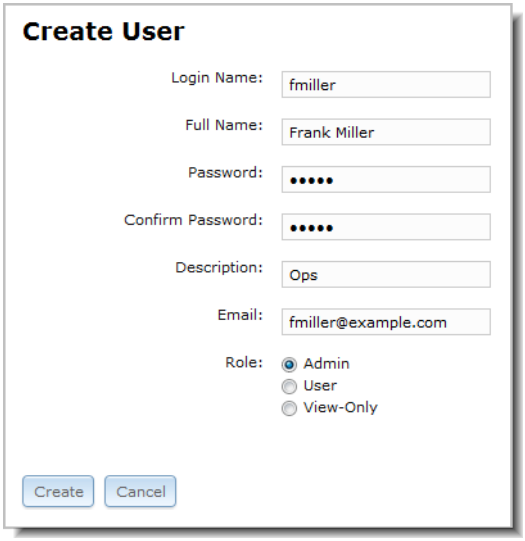
If you are not using an LDAP server to manage Federator's user authentication, you will need to create accounts for the users in your system.

Creating User Accounts

If you are an admin-level user, you can create new user accounts.

To create new accounts:

1. Click the **Users** tab.
2. Click **New**.
3. Enter the log-in credentials and other information to create the user account.
4. Click **Create**. The new account is created and appears in the list of users.



Editing Accounts

To edit an existing account:

1. Click the **Users** tab. If you are not an admin user, you are brought directly to the Show User page for your account and you can skip step 2.
2. Click the login name of the user you wish to edit.
3. Click **Edit**. (As a shortcut, you can get to the Edit page by clicking the edit (pencil) icon for the user on the list page.)
4. Make the desired changes.

5. Click **Update**. The account is updated with the new information.

If you are logged on as a non-admin-level user, you can only edit the password, full name, and description fields, and only of your own account.

Deleting Accounts

To delete an account, click the delete (trash can) icon next to its name in the list.

Only admin users can delete user accounts.

Enabling and Disabling Accounts

To change whether a user account is enabled:

1. Click the **Users** tab.
2. Click the login name of the user you wish to edit.
3. Click **Enable** or **Disable**. The button that appears depends on the current state of the account—if the account is currently enabled, the button reads "Disable," and vice-versa. After you click the button, the account's status changes. (As a shortcut, you can click the enable or disable icon for the user on the list page.)

Only admin users can enable or disable accounts. Admin users cannot disable the account they're currently using.

Modifying Federator's Configuration and Licensing

If you are an admin user, you can change the configuration parameters Federator uses in the user interface as well as upload license files. These are done using the Admin tab.

Configuring Federator

On the General page of the Admin tab you can specify global parameters affecting how Federator runs. To edit them, simply change one or more values and click the **Update** button (click **Revert** to undo any changes you have made).

General Configuration Parameters

The general parameters you can configure are:

Table 4 Configuration Parameters

Parameter	Description	Default Value
Logging Level	The level of granularity of the information contained in Federator's logs. The default setting, Info, provides standard informational output and should be appropriate most of the time. Changing this setting to Debug, Trace, or All increases the amount of information written to the logs and can be helpful when troubleshooting problems. See Viewing Federator Logs, page 14 for more details about Federator's logging.	Info

Table 4 Configuration Parameters (Continued)

Parameter	Description	Default Value
Scheduler Interval	The interval between runs of Federator's scheduler, in seconds. The shorter this interval, the more often Federator reallocates resources among your DataSynapse Resource Managers. While resources are being migrated among Resource Managers, the short scheduler interval (see below) is used instead, to provide more responsive migration. The scheduler interval must be greater than or equal to the minimum interval (see below).	60 seconds
Scheduler Minimum Interval	The interval between runs of Federator's scheduler *while resources are being migrated*, in seconds. The two intervals work in tandem to allow Federator to only poll frequently for resource status from the managers while migration is occurring. The minimum interval must be at least 5 seconds and less than or equal to the scheduler interval (see above).	5 seconds
Migration Timeout	The time, in minutes, that migrating Engine Daemons have to log into their new target Resource Manager. If they fail to do so within in this time frame, Federator will generate an alert.	3 minutes
Alert Timeout	The time, in minutes, that the alert panel on the dashboard will keep alerts visible. Alerts are visible on the drill-down pages until they end of the Statistics Expiration period (see below).	60 minutes

Table 4 Configuration Parameters (Continued)

Parameter	Description	Default Value
Statistics Expiration	The time, in days, that statistics, scheduler events, and alerts are maintained in Federator's database. Information older than this will be removed.	30 days
Policy Interval Minimum	The minimum duration, in minutes, that policy intervals must have when created. If you try to create a policy interval shorter than this, Federator will show an error and force you to lengthen the interval.	15 minutes
License Expiration Warning	The amount of time, in days, before your license expires that Federator will begin warning you of the impending expiration. During this period, Federator will show a warning when you log on and the license (key) icon at the top of the pages will flash red.	30 days

EC2 Configuration Parameters

The following parameters are used for EC2 configuration:

Table 5 EC2 Configuration Parameters

Parameter	Description	Default Value
Enable Proxy	Enables the proxy server which communicates with the cloud-side proxy when using SSH for EC2 tunneling. This starts the proxy when either Windows or UNIX versions of Federator are started. See Appendix B, Using EC2 Resources Without VPC, on page 99 for details.	Checked
Local Proxy Hostname	The hostname for the local proxy server. Change this value if you want to host your own proxy server and have Federator manage the SSH connection.	localhost

Table 5 EC2 Configuration Parameters (Continued)

Parameter	Description	Default Value
Startup Timeout	The time, in minutes, that Engine Daemons on EC2 instances have to log into their target Resource Manager. If they fail to do so within this time frame, Federator will generate an alert.	20 minutes
Shutdown Threshold	The time, in minutes, that an EC2 resource group must be idle (not specified in a current policy) before Federator will terminate its instances. For example, if you have two policies that use a particular EC2 resource group and the second one starts within this threshold from the time the first one ends, Federator will migrate the already running EC2 instances to the new target rather than terminating the instances and starting new ones. This can diminish the time spent waiting for instances to start.	20 minutes
Terminate instances when shutdown	If EC2 instances are terminated when shut down by Federator. You can change this to **False** to stop instances; when EBS instances are stopped instead of terminated, they will restart more quickly. For more information, see AMIs in EBS on page 31.	True

VMware Configuration Parameters

The following parameters are used for VMware configuration:

Table 6 VMWare Configuration Parameters

Parameter	Description	Default Value
Startup Timeout	The time, in minutes, that Engine Daemons on VMware instances have to log into their target Resource Manager. If they fail to do so within this time frame, Federator will generate an alert.	20 minutes

Table 6 VMWare Configuration Parameters (Continued)

Parameter	Description	Default Value
Shutdown Threshold	The time, in minutes, that a VMware resource group must be idle (not specified in a current policy) before Federator will terminate its instances. For example, if you have two policies that use a particular VMware resource group and the second one starts within this threshold from the time the first one ends, Federator will migrate the already running VMware instances to the new target rather than terminating the instances and starting new ones. This can diminish the time spent waiting for instances to start.	20 minutes
Destroy virtual machines when shutdown	If VMware instances are terminated when shut down by Federator. You can change this to **False** to stop instances.	True

Federator Licensing

On the Licensing page of the Admin tab you can see information about the currently installed license and upload a new license file. The License Details section shows when your license will expire, the version of Federator you are running, how many DataSynapse Resource Managers you are allowed to have, and whether the license restricts you to running on particular hosts.

To upload a new license file:

1. Click the **Admin** tab.
2. Click the **Licensing** sub-navigation link.
3. Click the **Browse…** or **Choose File** button (the name depends on the web browser you are using) in the License Upload section and locate the new license file you want to upload (usually named `license.ser`).
4. Click the **Upload…** button to upload the license file to Federator. Federator validates the license and updates the page to show the new details.

Viewing Federator Logs

Federator generates logs in the *DS_HOME*/`logs` directory. In addition to its `federator.log` log file, logs are also generated by the Tomcat application server.

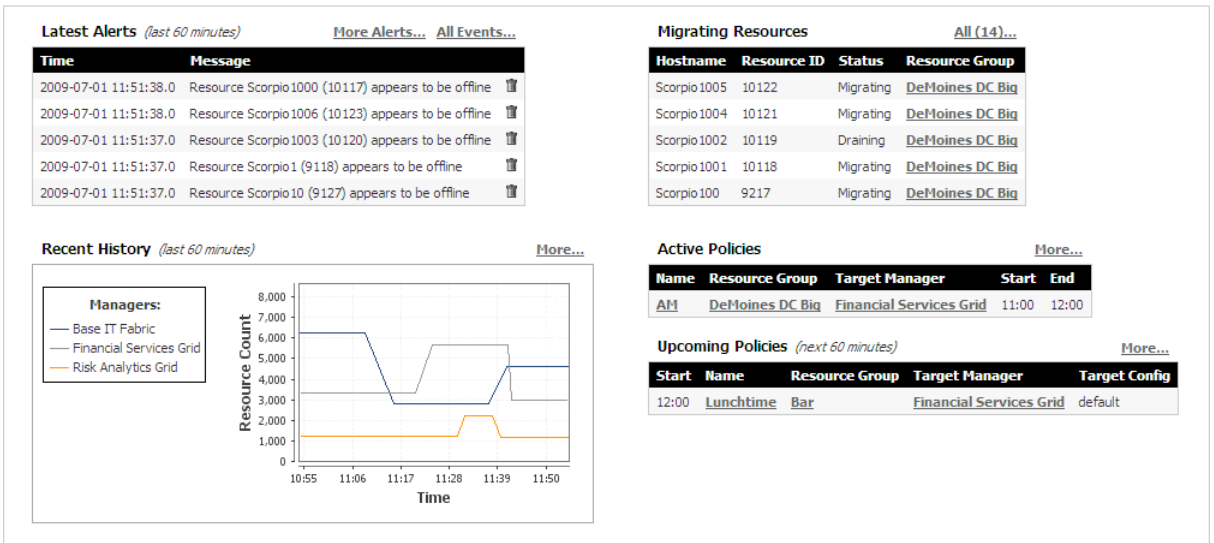
Monitoring Federator Activity

Click Federator's **Activity** tab at any time to monitor current Federator activity. The Activity Dashboard is divided up into sections that show:

- The most recent alerts regarding problems migrating resources (within the alert timeout specified on the configuration page; the default is within the last hour)
- A chart showing the history of resource migration (within the past hour)
- The status of resources that are currently in the process of migrating
- A list of active policies as well as any policies that are scheduled to become active within the next hour

Each of these sections provides links that let you drill down and find out more detail about its contents. The activity dashboard automatically updates itself periodically.

Activity Dashboard



Alerts and Scheduler Events

When Federator's scheduler detects resource changes, it creates and logs change events. Some important change events that merit the administrator's attention are also shown as alerts. The Latest Alerts panel has two links (**More Alerts…** and **All Events…**) that take you to detail pages (Alerts and Scheduler Events,

respectively) that show all the alerts and events currently on record in the Federator database. If you want to acknowledge an alert and allow room for others to be displayed on the panel abbreviated list, click the delete button next to the alert and it will be removed from the panel (it will remain on the drill-down pages). Alerts and other events are kept for a period determined by the configuration property named Statistics Expiration (thirty days by default), though the panel on the dashboard only shows alerts for the Alert Timeout period (one hour by default). See Configuring Federator on page 9 for details.

Recent History

The Recent History chart shows the number of resources allocated to various DataSynapse Resource Managers over the past hour. Clicking the **More…** button will show the Resource History page with a more-detailed chart. Here you can specify which managers' resources to include, and the exact time frame you'd like to see.

Migrating Resources

The Migrating Resources panel shows resources which are in the process of being migrated to a new target DataSynapse Resource Manager. These are resources which have been told to move but which haven't yet logged into the target's Resource Manager. The list is limited to the five most recently migrated resources. To see the full list, click the **All…** link to go to the Migrating Resources page. (Note that the link tells you how many total migrating resources there are.)

Policies

The Active Policies panel shows you the policy intervals which are currently active. It will show up to five policies—click the **More…** link to see them all. The Upcoming Policies panel shows any policies which are scheduled to start within the next hour.

Additional Monitoring Pages

In addition to the Activity Dashboard and its drill-down pages, you can see information about the state of Federator using pages available throughout Federator's user interface. For details, see:

- Viewing or Editing a DataSynapse Resource Manager on page 21
- Viewing or Editing a Static Resource Group on page 51
- Creating Dynamic Resource Groups on page 53

- Viewing or Editing a Policy Interval on page 64
- Reclaiming Abandoned Resources Manually on page 65
- Viewing or Editing an EC2 Resource Manager on page 36

Chapter 3 Using DataSynapse Resources

DataSynapse Resource Managers represent your GridServer and Silver Fabric clouds. Each Resource Manager is associated with a GridServer Director or Silver Fabric Broker, and contains the URLs and credentials that allow Federator to communicate with the Director or Broker.

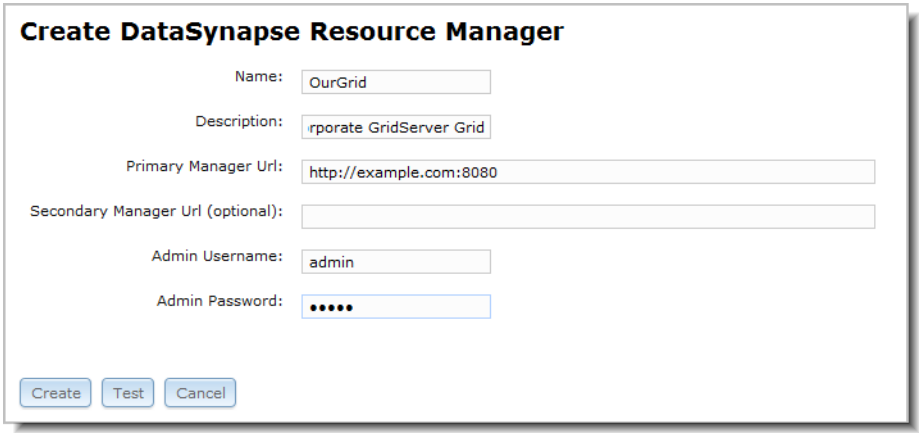
Federator communicates with DataSynapse Resource Managers to find out about logged in resources, to tell resources to migrate to a different Resource Manager, and to see the status of migration. You will create a DataSynapse Resource Manager for each GridServer or Silver Fabric cloud you want to federate.

Topics

Creating a New DataSynapse Resource Manager

To create a DataSynapse Resource Manager:

1. Click the **DataSynapse** link under the **Resource Managers** tab.
2. Click **New**.



3. Enter information about your cloud:

Table 7 DataSynapse Resource Manager Properties

Property	Description
Name	The name of the Resource Manager. Resource Managers are referred to by name when specified in resource groups and policy intervals.
Description	An optional description for the Resource Manager.
Primary Manager URL	The URL to your cloud—the Director for a GridServer grid or Broker for a Silver Fabric cloud. This is generally in the form `http://grid1.example.com:8000/`.
Secondary Manager URL	The URL of a secondary, or failover, Manager or Broker. If there is no secondary Manager, leave this field blank.

Table 7 DataSynapse Resource Manager Properties (Continued)

Property	Description
Admin Username	The username that Federator uses when connecting to the Resource Manager. This user must have administrative privileges on the Resource Manager.
Admin Password	The password that Federator uses when connecting to the Resource Manager with the given username. (Note that all passwords are stored encrypted within Federator.)

4. Click **Test** to make sure you entered the information correctly. Test can reveal the following types of problems:
   - If you've typed an incorrect URL
   - If you've typed incorrect login credentials
   - If the Resource Manager hasn't yet been enabled for Federator
5. Correct any errors that appear and test again until the errors are resolved.
6. Click **Create**. The DataSynapse Resource Manager is created for the specified cloud and appears in the list of Resource Managers.

Viewing or Editing a DataSynapse Resource Manager

To view or edit an existing Resource Manager:

1. Click the **DataSynapse** link under the **Resource Managers** tab.
2. Click the name of the Resource Manager you want to view or edit.
3. If you want to edit the values, click the **Edit** button. (As a shortcut, you can get to the Edit page by clicking the edit (pencil) icon for the Resource Manager on the list page.)
4. Make any changes you want and click **Update** to save them. Note that you can use the **Test** button when editing a Resource Manager to validate your changes just as when you created it.
5. The DataSynapse Resource Managers page contains the following columns:

Table 8 DataSynapse Resource Manager Properties

Property	Description
Name	The name of the Resource Manager.
Description	An optional description for the Resource Manager.
Status	If the status of the Resource Manager is online, offline, or disabled.
Type	The type and version of GridServer Director or Silver Fabric Broker.
Primary URL	The Manager URL to the Director for a GridServer grid or Broker for a Silver Fabric cloud.
Secondary URL	The URL of a secondary, or failover, Manager or Broker.
Resources	The total number of resources for the Resource Manager.
Active	The number of demand resources plus the number of shared resources currently in the active policy.

Table 8 DataSynapse Resource Manager Properties (Continued)

Property	Description
External	The number of resources for this Resource Manager that are currently at other Resource Managers.
CPUs	The number of CPUs for the Resource Manager.

Deleting DataSynapse Resource Managers

To delete a Resource Manager, click the delete (trash can) icon next to its name in the list.

Enabling or Disabling a DataSynapse Resource Manager

Federator can expend considerable resources (CPU, network, and time) attempting to contact offline resources. If a Resource Manager is going to be unavailable for an extended period, you can disable it so that Federator will ignore it, which will improve Federator performance.

If a Resource Manager is offline temporarily due to a network issue or short maintenance operation, it will come back online if it is left enabled. If a Resource Manager will be offline for more than a few minutes, or if it no longer exists, disable it to avoid performance issues.

To enable or disable a Resource Manager:

1. Click the **DataSynapse** link under the **Resource Managers** tab to show the list of DataSynapse resource mangers.
2. Click the name of the Resource Manager you want to modify. It appears on the Show DataSynapse Resource Manager page.
3. Click **Enable** or **Disable** to modify the Resource Manager (if the Resource Manager is currently enabled, the button says “Disable,” and vice-versa). (As a shortcut, you can click the enable or disable icon for the Resource Manager on the list page.)

Chapter 4 Using Amazon EC2 With Federator

Federator enables you to use Amazon's Elastic Compute Cloud (EC2) to launch Engine Daemons in the Amazon compute cloud for your clouds to use. You can use these resources on a continuing basis, to account for temporary needs, for regular, periodic demand, or whatever makes the most sense for your business.

Topics

Configuring EC2 for Federator

Before you can create EC2 Resource Managers in Federator, there are several configuration tasks you must complete.

To use EC2 resources:

1. Obtain an EC2 account. See Obtaining an EC2 Account on page 26.
2. By default, AWS accounts enabled for EC2 allow a maximum of twenty instances to be launched simultaneously. If you need to launch more Engines, you can request an increase by visiting http://aws.amazon.com/contact-us/ec2-request. Note that it may take several days for Amazon to process your request
3. Get Access Identifiers and a Certificate Pair. See Getting Access Identifiers and a Certificate Pair on page 27.
4. Set up the Virtual Private Cloud (VPC). See Setting Up the Virtual Private Cloud on page 27.
5. Create Security Groups. See Creating Security Groups on page 28.
6. Create Key Pairs. See Creating Key Pairs on page 30.

Obtaining an EC2 Account

The first step in getting Federator working with Amazon EC2 is to create and set up an account with Amazon Web Services (AWS). Visit http://aws.amazon.com for details on getting started.

You will initially set up your AWS account using a credit card for billing purposes. You can contact Amazon to arrange for other billing options available to corporate customers.

For more information on EC2, visit the Amazon AWS web site at http://aws.amazon.com. There you'll find details about AWS in general, and EC2 more specifically. There are "Getting Started" guides, developer guides, and reference sheets for the various services. In addition, there are links for technical support and account administration.

Getting Access Identifiers and a Certificate Pair

Access identifiers are used by Federator to connect with the EC2 server on your behalf. There are two, the AWS Access Key ID and the AWS Secret Access Key. You can think of them as a name and password for connecting to EC2. You can see these from the Access Identifiers page of the AWS web site while logged on with your account (click the Show link to make your secret access key visible).

Amazon uses an X.509 certificate to secure the connection to the server when using the command-line tools. You can have Amazon generate one for you or you can upload your own to the server. You can do either on the same Access Identifiers page of the AWS site. See the command-line tool documentation for details on how to use the certificate.

If you generate a new secret access key you will need to update the EC2 Resource Managers and groups in Federator to use it or they will no longer be able to communicate with the EC2 server. Similarly for the X.509 certificate.

Setting Up the Virtual Private Cloud

Because EC2 resources aren't hosted on your corporate network, there must be a method for Engines in the Amazon cloud to securely communicate with Directors and Brokers in your corporate network.

To accomplish this, Federator supports the use of a Virtual Private Cloud (VPC), which enables you to provision a virtual network within the Amazon Web Services cloud. This enables you to host EC2 instances in Amazon's cloud as if they were local instances.

Federator will also start a proxy machine instance in the cloud that runs a proxy server. The proxy server caches the grid resources that are transferred when Engine Daemons synchronize with their Brokers. This helps to minimize the network traffic into and out of the Amazon cloud. Depending on the quantity of data your clouds need to synchronize, you may wish to alter the instance type used for the proxy server to provide more storage.

Previous Federator versions supported another method of tunneling, using a Squid proxy and SSH tunneling. This legacy method is still supported for evaluation purposes, but the VPC method should be used for production environments. See Appendix B, Using EC2 Resources Without VPC, page 99 for more details.

Configuring Amazon VPC

Amazon Virtual Private Cloud (VPC) is a service that enables you to provision a virtual network within the Amazon Web Services cloud. You can define the network's topology, how it is secured, what subnets are public-facing, and bridge the network to your existing networking environment to serve as a virtual extension of your own network. For more information, see http://aws.amazon.com/vpc/.

To create a VPC, log in to the AWS Management Console and click the **Get Started Creating a VPC** button. This will open a wizard that enables you to create and configure a VPC. This involves attaching an internet gateway, creating a subnet, and setting up the VPN link between your network and the Amazon cloud.

For more information on setting up VPC, see Amazon's documentation at http://docs.amazonwebservices.com/AmazonVPC/latest/GettingStartedGuide and http://docs.amazonwebservices.com/AmazonVPC/latest/UserGuide/.

Configuring Your Network

The Engines in the VPC must be able to communicate port 22 (the SSH port), port 3389 (the RDP port), and port 27159 (used by GridServer and Silver Fabric) with machines on your corporate network. You must ensure that your corporate network allows inbound communication on these ports.

Also, Engines must be able to resolve the DNS names of Managers and Drivers so that DDT will work.

Creating Security Groups

Before you can launch a machine instance in EC2, you must associate the instance with a security group. Security groups specify access rules such as which network ports will be open to the instance and which IP addresses the instance can connect to via these ports.

You should create a group for the proxy instance and at least one additional group for your Engines. The proxy's group will be specified when creating the EC2 Resource Manager and the Engine group will be used by the EC2 resource groups.

To create new security groups, log in to the AWS Management Console, go to **EC2 > Network & Security > Security Groups**. Then click **Create Security Group**.

For each security group, you must add rules to allow communication for several ports.

To add a new rule:

1. Select the security group.
2. Click the **Inbound** tab.
3. Select **Custom TCP rule**.
4. For **Port range**, enter a port to allow. See below for the ports you need to open for each security group.
5. For Source, enter your corporate subnet and subnet mask bits, such as 192.168.2.0/24.
6. Click **Add Rule**.
7. After adding all rules, click **Apply Rule Changes**.

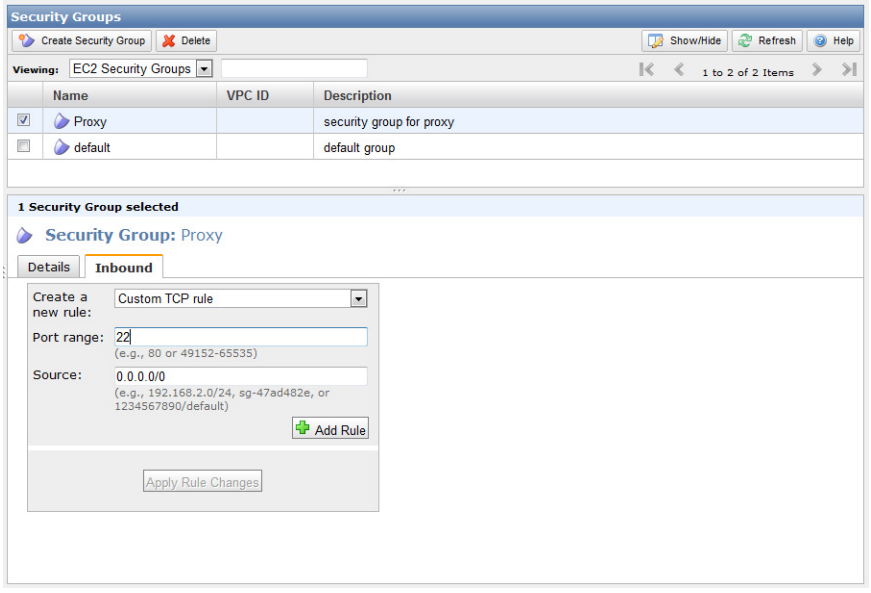


Figure 1 Adding a port to the proxy security group.

You can also add and edit security groups using the EC2 command-line tools. The `ec2-add-group` command can be used to add groups, and the `ec2-authorize` command can be used to add TCP rules to open ports.

Proxy Group

The proxy must have permission to communicate on port 22 (the SSH port) and 80 (the HTTP port) with machines on your corporate network. It must also have permission to communicate on port 3128 from instances in the cloud (specified based on their security group).

Engine Group

The Engines must have permission to communicate on port 22 (the SSH port), port 3389 (the RDP port), and port 27159 (used by GridServer and Silver Fabric) with machines on your corporate network. It must also have permission to communicate on port 3128 from instances in the cloud (specified based on their security group).

Creating Key Pairs

Amazon lets you generate X.509 certificates for use by the instances you launch. Each instance is launched using a specific key pair. Knowing this key pair allows you to make connections via SSL into these virtual machines. Federator automatically creates and uses a key pair for the SSH-based tunnel that the proxy server uses to communicate between the Amazon cloud and your corporate network.

You may wish to create additional key pairs for use by the Engine instances you create. Each EC2 resource group you define lets you choose a specific key pair. You can either use the same one that Federator creates for the proxy to use (named `fed-`*Proxy IP Address*`-keypair`) or you can create your own. Be aware that you can only download the key pair when you first generate it.

Amazon Machine Images

Amazon Machine Images (AMIs) are the templates that Amazon uses to start instances in the cloud. They contain the operating system software for the virtual machine as well as any other software that may have been pre-installed. Federator requires two types of AMI, one for the proxy server and one for DataSynapse Engine Daemons. Each of these is available in different flavors depending on your OS needs and preferences. When you create an EC2 Resource Manager in Federator, you tell it which AMI to use when instantiating the proxy server; when you create an EC2 resource group, you tell it which AMI to use for instantiating engine machines.

AMIs in EBS

TIBCO provides the necessary AMIs with Amazon's Elastic Block Store (EBS) product. They are publicly available and the AMI names use TIBCO's owner ID, which is 575373498564. Filtering the owner to 575373498564 will restrict the list to just the TIBCO AMIs.

EBS-backed instances are persistent, meaning changes to a running instance are saved so updates to Engines and resources (like Grid Libraries) can be done once rather than every time a cloud Engine is launched.

Instances can also be stopped instead of terminated when they are shut down by Federator. A stopped instance can be restarted faster than starting a new instance. Federator will consider stopped instances before starting a new instance.

By default, instances are terminated when shut down. To change this, go to **Admin > General**, and set **Terminate instances when shutdown** to false.

Custom AMIs

For most customers, the AMIs provided by TIBCO will be sufficient. Some customers may wish to create their own images with additional software installed. Amazon provides tools for working with AMIs to enable you to manipulate their contents.

To create a custom EBS:

1. Log into the AWS Management Console and launch an AMI.
2. Make whatever changes you want to it, such as pre-installing data files for use by your grid tasks.
3. Right-click on the instance in the AWS Management Console and select "Create Image (EBS AMI)".

4. Give the new AMI a name.

After a few minutes, your new AMI will be available.

EC2 Resource Managers

An EC2 Resource Manager represents a connection to the Amazon Web Services (AWS) Elastic Compute Cloud (EC2) service. Using an EC2 Resource Manager, Federator can launch Engine Daemons in the Amazon cloud which can be assigned to your DataSynapse Resource Managers in order to do work as part of your clouds. In order to use EC2 Resource Managers, you must set up an account with Amazon and properly configure it. See Configuring EC2 for Federator on page 26 for more details. Many of the fields used to specify an EC2 Resource Manager govern the manager's proxy server. For more about proxy servers, see Setting Up the Virtual Private Cloud on page 27.

Creating a New EC2 Resource Manager

To create an EC2 Resource Manager:

1. Go to **Resource Managers > Amazon EC2**.
2. Click **New**.
3. Enter the information about your EC2 account:

Table 9 EC2 Resource Manager Properties

Property	Description
Name	The name for the Resource Manager. Resource Managers are referred to by name when specified in resource groups.
Description	An optional description for the Resource Manager.
Engine Instance Limit	The maximum number of instances you want to run on this account. This does not include stopped instances, or the proxy server instance. The Amazon limit for your account (default of 20) allows for that many stopped as well as that many running. In the default case of 20, you could have up to 20 running and up to 20 stopped at the same time.
AWS Access Key ID	The AWS access key ID for your AWS account.

Table 9 EC2 Resource Manager Properties (Continued)

Property	Description
Secret AWS Access Key	The secret access key for your AWS account. The two keys are used to allow Federator to log onto EC2 on your behalf.
Region	The EC2 region your Resource Manager will run in. If you want to run Engines in multiple regions, you will need to create one EC2 Resource Manager for each region. (Note that Amazon may charge different fees based on regions.)
EC2 Tunnel Type	The method used by instances in Amazon's cloud to reach machines on your network. Select **VPC Tunnel**, **SSH Tunnel**, or **No (manual) Tunnel**. See Setting Up the Virtual Private Cloud on page 27 for more details
Elastic IP	The IP address to use for the proxy instance. This value is only used if the **EC2 Tunnel Type** is **SSH Tunnel**. The list shows the elastic IP addresses configured for your account and indicates which ones aren't currently bound to a running instance. See , Creating an Elastic IP Address, page 101 for more about elastic IP addresses.
VPC	The ID of the Virtual Private Cloud that will be used for the instances.This value is only used if the **EC2 Tunnel Type** is **VPC Tunnel**.
AWS Image	The Amazon Machine Image (AMI) to use for the proxy server associated with this EC2 Resource Manager. (Note that this list is filtered to only show images with "proxy" in their manifest name.)

Table 9 EC2 Resource Manager Properties (Continued)

Property	Description
Instance Type	The instance type to use when running the proxy server. The instance type determines how much processing power and disk space the instance will have. Standard instances may be small, large, or extra large. High-CPU instances may be medium or extra large. Note that the values in this list change to reflect the types appropriate for the image you have chosen. For the proxy server, you will want to use a standard type—the size will be determined by the size of the grid resources you need to cache. For specific details on the instance types and their pricing, see the EC2 web site.
Security Groups	You can choose one or more of the EC2 security groups defined in your account for the proxy server to run in. Generally, you would have defined a security group specifically for the proxy server and would choose that group here (see Creating Security Groups on page 28).
Spot Price	An optional spot price you are willing to pay for this instance. See About Spot Instances on page 38 for more information on Spot Instances.
Availability Zone	The availability zone for the proxy server. Choose **Automatic** to let Amazon choose for you at start-up time.
VPC Subnet ID	The subnet used for the Virtual Private Network. This value is only used if the **EC2 Tunnel Type** is **VPC Tunnel**.
User Data	Optional user data; only used when you are using a custom AMI for the proxy server and it requires this field.

Table 9 EC2 Resource Manager Properties (Continued)

Property	Description
Kernel ID	An optional kernel ID; only used when you are using a custom AMI for the proxy server and it requires this field.
Ram Disk ID	An optional Ram Disk ID; only used when you are using a custom AMI for the proxy server and it requires this field.
Block Device Mappings	Optional block device mappings; only used when you are using a custom AMI for the proxy server and it requires this field. This field maps block devices from Amazon Elastic Block Store (EBS) to devices visible to the instance. The format is the same as for the `run-instances` command in the command-line interface—see Amazon's documentation for details.

4. Click **Test** to make sure you entered the information correctly. Test can reveal the following types of problems:
   - — If any of the required fields are missing
   - — If any of the fields' values are incompatible with one another
   - — If you've typed incorrect access keys
   - — If there are communication issues connecting to the EC2 server
5. Correct any errors that appear and test again until the errors are resolved.
6. Click **Create**. The EC2 Resource Manager is created and appears in Federator's list.

Viewing or Editing an EC2 Resource Manager

To view or edit an existing Resource Manager:

1. Click the **Amazon EC2** link under the **Resource Managers** tab.
2. Click the name of the Resource Manager you want to view or edit. It appears on the Show Amazon EC2 Resource Manager page.
3. If you want to edit the values, first disable the Resource Manager (described below), then click the **Edit** button to show the Edit Amazon EC2 Resource

Manager page. (As a shortcut, you can get to the Edit page by clicking the edit (pencil) icon for the Resource Manager on the list page.)

4. Make any changes you want and click **Update** to save them. The fields are the same as when creating a new EC2 Resource Manager. Note that you can use the **Test** button when editing a Resource Manager to validate your changes just as when you created it.

Enabling or Disabling an EC2 Resource Manager

You can enable and disable resource mangers. Federator uses the enabled state of EC2 Resource Managers to know when to start and stop the associated proxy server instance (see About Spot Instances on page 38 for more about proxy servers).

To enable or disable a Resource Manager:

1. Click the **Amazon EC2** link under the **Resource Managers** tab.
2. Click the name of the Resource Manager you want to modify. It appears on the Show Amazon EC2 Resource Manager page.
3. Click **Enable** or **Disable** to modify the Resource Manager (if the Resource Manager is currently enabled, the button says "Disable," and vice-versa). As mentioned above, the proxy server will be started or stopped by this action. (As a shortcut, you can click the enable or disable icon for the Resource Manager on the list page.)

Deleting EC2 Resource Managers

To delete an EC2 Resource Manager, click the delete (trash can) icon next to its name in the list.

About Spot Instances

Spot instances enable you to bid a spot price on unused Amazon EC2 instances. Instead of paying a fixed price for on-demand instances, you can bid a maximum price you are willing to pay for instances. Amazon's current spot price changes periodically due to supply and demand, which means instances will become available when demand drops. This offers the opportunity of a potentially lower instance price, but with the tradeoff that instances may not always be available.

When you specify a spot price, instances are only started when your spot price exceeds Amazon's current spot price. If your spot price is lower than Amazon's current spot price, new instances won't start. Amazon also limits the amount of resources available at the spot price, so even if your price exceeds the spot price there may still not be any resources available. Also, when Amazon's spot price drops lower than your offered spot price, instances will be terminated.

Spot instances cannot be used when using Amazon VPC for your EC2 instances. Because of this, spot instances should not be used in production environments. They are best used in situations such as development or QA, where lower-cost instances are desired, and reliability is not an issue.

Note that the instance startup timeout does not apply to spot instances.

To enable spot pricing, enter your price in the **Spot Price** field when creating or editing an EC2 Resource Manager, EC2 resource group, or EC2 demandable resources.

It's also possible to set a spot price on the proxy, but no Engine spot requests will be entered until the proxy instance is started, because Engines need a proxy to function. For this reason, a best practice would be to launch a regular proxy instance at the normal price and then use spot Engines with it.

For more information on Amazon EC2 Spot Instances, including updated current spot prices, see http://aws.amazon.com/ec2/spot-instances/.

Chapter 5 Using VMware With Federator

Federator enables you to use VMware vSphere to launch Engine Daemons in the vSphere environment for your clouds to use. You can use these resources on a continuing basis, to account for temporary needs, for regular, periodic demand, or whatever makes the most sense for your business.

Topics

Configuring VMware for Federator

Federator supports Engines hosted on virtual machines running VMware through a Resource Manager for VMware. It can only start and stop existing virtual machines and clone new virtual machines from existing templates. Federator does not create new virtual machines from scratch nor does it install the Engine on pre-existing virtual machines or templates.

Federator supports VMware vCenter Server 4.0 on the following platforms.

Platform	Version
UNIX/Linux	RHEL 4, 5, 6; SLES 10, 11; Solaris 10 x86 (64-bit only), 11
Windows	XP Professional; 2003 Server; 2008 Server, Windows 7

Creating Templates

Before you create a VMWare Resource Manager, you must create templates, which are virtual machines that run Engines.

To create templates:

1. Create a base location for Federator related activity such as `/Datasynapse/federator`
2. Create a folder for templates such as `/Datasynapse/federator/templates`
3. Create a folder for cloned virtual machines, such as: `/Datasynapse/federator/engines`
4. Optionally, create a folder called `init` for the initial virtual machine pool under the base location, such as: `/Datasynapse/federator/engines/init`. If the virtual machine exists in the `init` folder you create, then the virtual machine in `init` is started.

5. Create templates as follows:
   a. Create a virtual machine with the desired OS.
   b. Install VMware Tools. Federator relies on functionality provided by VMware Tools.
   c. Install the GridServer/Silver Fabric Engine in the usual way.
   d. Run `configure.sh`, start the Engine, and ensure that the Engine can communicate with the Broker.
   e. Shut down the Engine and remove the contents of the profiles directory.
   f. Create a file in the Engine directory called `engine-session.properties`, and add a property with the name `vimTemplate` with the value set to the name of the template.
   g. Configure the Engine to start up automatically upon boot, using the OS-appropriate procedure. See "Example Startup Scripts" for details.
   h. Shut down the VM cleanly.
   i. Convert the VM to a template.
6. Ensure that all templates reside in the templates folder you created above.

Configuring SSL

VMware uses SSL by default for client communication.

To use SSL to communicate with vCenter:

1. Copy the default `cacerts` file from your Java installation (*JAVA_HOME*`/jre/lib/security/cacerts`) to a location in your Federator installation, such as `DS_HOME/webapps/federator/WEB-INF/certs/ssl.keystore`.
2. Import the certificate used by vCenter to this `cacerts` file. The vCenter certificate is at: `C:\Documents and Settings\All Users\Application Data\VMware\VMwareVirtualCenter\SSL\rui.crt`. For example:

   ```
   keytool -importcert -file rui.crt -keystore ssl.keystore \
   -alias   [cert alias]
   ```

3. Edit the `server.bat` or `server.sh` file and change the JAVA_OPTS to use the keystore. For example:

   ```
   set SSL_KEYSTORE="path\to\certs\ssl.keystore
   set JAVA_OPTS=%JAVA_OPTS% \
   -Djavax.net.ssl.keystore=$SSL_KEYSTORE
   ```

Example Startup Scripts

Federator passes several properties to cloned virtual machines that can be retrieved from VMware Tools. See the “Using VMware Resource Groups” for a list of properties available to the virtual machine to automatically configure the Engine at operating system startup. The following is an example startup script for Red Hat Linux:

```
#!/bin/sh
# Startup script for DataSynapse Engine
#
# Expects the existence of a user 'ds' with exec/write permissions
# in the root Engine directory
#
DSBASE=/opt/TIBCO/datasynapse/engine
JAVA_HOME=/usr/local/java
VMWARE_TOOLS=/usr/sbin/vmware-guestd
VMWARE_TOOLS_CMD='info-get guestinfo.broker_url'
VMWARE_MUSTCONFIG_CMD='info-get guestinfo.must_configure'

export JAVA_HOME
case "$1" in
    start)
        cd $DSBASE
        BROKER_URL=
        while [ $SECONDS -lt 120 ]; do
            echo $VMWARE_TOOLS --cmd "$VMWARE_TOOLS_CMD"
            BROKER_URL=`$VMWARE_TOOLS --cmd "$VMWARE_TOOLS_CMD"`
            if [ "$BROKER_URL" == "" -o "$BROKER_URL" == "No value
found" ]; then
                echo sleeping
                sleep 5
            else
                echo $BROKER_URL
                ENGINE_OPTIONS=`echo $BROKER_URL | sed
's/https\?:\/\/\(.*\)/\1/'`
                if [[ $BROKER_URL == https* ]]; then
                    ENGINE_OPTIONS=$ENGINE_OPTIONS -l y
                fi
                MUST_CONFIGURE=`$VMWARE_TOOLS --cmd
"$VMWARE_MUSTCONFIG_CMD"`
       if [ ! -d profiles ] || [ $MUST_CONFIGURE" == "true" ]; then
            sudo -u ds ./configure.sh $BROKER_OPTIONS
                fi
                sudo -u ds ./engine.sh start
                exit 0
            fi
        done
        echo "Error: no Broker URL found"
        exit 6
        ;;
    stop)
        cd $DSBASE
        sudo -u ds ./engine.sh stop
        ;;
    restart)
```

```
        cd $DSBASE
        sudo -u ds ./engine.sh stop
        sudo -u ds ./engine.sh start
        ;;
    *)
        echo $"Usage: $0 {start|stop|restart}"
        exit 1
        ;;
esac
```

The following is an example RunOnce script for Windows:

```
@REM Copyright 2012 TIBCO Software Inc. All Rights Reserved.
@echo off
setlocal
set COMMAND_PATH=C:\Program Files\VMware\VMware Tools
set COMMAND_NAME=VMwareService.exe
set COMMAND_ARGS="info-get guestinfo.broker_url"
set BROKER_URL=No Value Found
set DS_ENGINE_SERVICE=DataSynapse Engine
set ENGINE_BASE_DIR=C:\Program Files\DataSynapse\Engine
set INTRANET_DAT="%ENGINE_BASE_DIR%\intranet.dat"
set COUNT=1
cd %COMMAND_PATH%
:checkTools
if %COUNT% gtr 20 (
    echo Failed to get Broker URL
    goto:EOF
) else (
    set /a COUNT+=1
)
for /F "usebackq tokens=*" %%i in (`%COMMAND_NAME% -cmd
%COMMAND_ARGS%`) do @set BROKER_URL=%%i
if /i "%BROKER_URL%" == "No Value Found" (
    @REM Sleep for 5 seconds using the Windows PowerShell
    call C:\WINDOWS\system32\WindowsPowerShell\v1.0\powershell.exe
sleep 5
) else (
    goto:configureEngine
)
goto:checkTools

:configureEngine
echo Configuring Engine with Broker URL %BROKER_URL%
net stop "%DS_ENGINE_SERVICE%"

if EXIST %ENGINE_BASE_DIR%\profiles (
    del /S /Q %ENGINE_BASE_DIR%\profiles
)
set
NEW_BROKER_URL=%BROKER_URL%/livecluster/public_html/register/regis
ter.jsp
echo %NEW_BROKER_URL% > %INTRANET_DAT%
net start "%DS_ENGINE_SERVICE%"
```

Federator VMware Configuration Parameters

Federator has several optional parameters used to affect behavior of VMware Engines. For more information, see VMware Configuration Parameters on page 12.

VMware Behavior During Failover

When the cloud transitions to the Failover Broker, Engines residing on running virtual machines will move to the Failover Broker, but the number of running virtual machines will not change — the Failover Broker will neither start new virtual machines nor stop existing ones.

VMware Resource Managers

A VMware Resource Manager represents a connection to a VMware vSphere environment. Using a VMware Resource Manager, Federator can launch Engine Daemons in a vSphere environment which can be assigned to your DataSynapse Resource Managers in order to do work as part of your clouds.

Creating a New VMware Resource Manager

To create a VMware Resource Manager:

1. Click the **VMware** link under the **Resource Managers** tab.
2. Click **New**.
3. Enter the information about your VMware environment:

Table 10 VMware Resource Manager Properties

Property	Description
Name	The name for the Resource Manager. Resource Managers are referred to by name when specified in resource groups.
Description	An optional description for the Resource Manager.
Engine Instance Limit	The maximum number of instances you want to run on this account. This does not include stopped instances, or the proxy server instance.
VMware Server Url	The URL of the VMware server.
Admin Username	The username used to log into the VMware server.
Admin Password	The password used to log into the VMware server.

4. Click **Test** to make sure you entered the information correctly. Test can reveal the following types of problems:
   - If any of the required fields are missing
   - If any of the fields' values are incompatible with one another
   - If there are communication issues connecting to VMware
5. Correct any errors that appear and test again until the errors are resolved.
6. Click **Create**. The VMware Resource Manager is created and appears in Federator's list.

Viewing or Editing a VMware Resource Manager

To view or edit an existing Resource Manager:

1. Click the **VMware** link under the **Resource Managers** tab.
2. Click the name of the Resource Manager you want to view or edit. It appears on the Show VMware Resource Manager page.
3. If you want to edit the values, first disable the Resource Manager (described below), then click the **Edit** button to show the Edit VMware Resource Manager page. (As a shortcut, you can get to the Edit page by clicking the edit (pencil) icon for the Resource Manager on the list page.)
4. Make any changes you want and click **Update** to save them. The fields are the same as when creating a new VMware Resource Manager. Note that you can use the **Test** button when editing a Resource Manager to validate your changes just as when you created it.

Enabling or Disabling a VMware Resource Manager

You can enable and disable resource mangers.

To enable or disable a Resource Manager:

1. Click the **VMware** link under the **Resource Managers** tab.
2. Click the name of the Resource Manager you want to modify. It appears on the Show VMware Resource Manager page.
3. Click **Enable** or **Disable** to modify the Resource Manager (if the Resource Manager is currently enabled, the button says "Disable," and vice-versa). As a shortcut, you can click the enable or disable icon for the Resource Manager on the list page.

Deleting VMware Resource Managers

To delete a VMware Resource Manager, click the delete (trash can) icon next to its name in the list.

Chapter 6 Working with Resource Groups

Like Resource Managers, resource groups come in three types: DataSynapse resource groups, EC2 resource groups, and VMware resource groups. DataSynapse resource groups represent resources shared into Federator by DataSynapse Resource Managers (Silver Fabric Brokers and GridServer Directors) ; EC2 and VMware resource groups represent resources shared by EC2 and VMware Resource Managers.

Topics

Creating and Populating DataSynapse Resource Groups

GridServer and Silver Fabric resources are not assigned to resource groups by default. To use Placement policies to control migration of resources, you must assign them to DataSynapse resource groups. DataSynapse resource groups are divided into static and dynamic groups. Note that resources that are not assigned to resources groups may still be migrated to satisfy Demand-Based policy intervals.

Static resource groups contain, as their name implies, a static set of resources. Resources in a static resource group are migrated to the target Resource Manager while they're in use by an active policy interval. When not in use, resources in a static group return to the Resource Manager (and configuration) designated as the default for that group. You add resources to a static resource group by selecting them on one of the resource pages in the Federator Administration Tool and then explicitly adding them to the group.

Dynamic resource groups, on the other hand, contain potentially a different set of resources every time they are used. Dynamic groups contain a set of rules which specify their resources. When a dynamic resource group comes into effect as part of an active policy interval, its set of rules is used to find available resources. These resources are added to the group only for the duration of the current policy interval, after which time they revert back to the Resource Managers they had been assigned to prior to being grouped.

The sections that follow discuss creating, populating, and using these types of DataSynapse resource groups in more detail.

Creating Static Resource Groups

To create a static DataSynapse resource group:

1. Click the **Static** link under the **Resource Groups** tab.
2. Click **New**.

3. Enter the details for the resource group:
   - **Name:** Give the resource group a name. Resource groups are referred to by name when specified in policy intervals. All resource group names must be unique, even across different types of groups.
   - **Description:** Optionally, enter a description for the resource group.
   - **Default Manager:** Choose the default DataSynapse Resource Manager for this group. When the resources in the group are not involved in a policy interval, they will be returned to this Resource Manager.
   - **Initial Configuration:** You can optionally specify the name of the configuration to use when the Engine Daemons first start up. Use this field, for example, if you want them to synchronize with a particular Resource Manager but then connect to their "real" Resource Manager (as defined by the default configuration) afterwards. This is in effect when the group is not part of a policy interval; when in a policy interval, the policy interval's settings have precedence.
   - **Default Target Configuration:** Specify the name of the configuration to use when the Engine Daemons are running. This is in effect when the group is not part of a policy interval; when in a policy interval, the policy interval's settings have precedence.
   - **Default Busy Action:** Specify the action that should be taken when Federator wants to migrate resources in this group but they are busy. Choosing "Force" causes the resources to be migrated regardless of their busy status. Choosing "Abandon" causes Federator to give up on migrating resources that are still busy (taking them out of draining mode as well). This is in effect when the group is not part of a policy interval; when in a policy interval, the policy interval's settings have precedence.
   - **Default Warning Period:** Specify the warning period for resources in this group (in minutes). This is the amount of time before the resources are to be migrated that they will be put into draining mode (see Resource Draining on page 65). This is in effect when the group is not part of a policy interval; when in a policy interval, the policy interval's settings have precedence.
   - **Default Success Threshold:** Optionally, specify a percentage of the resources in the group (an integer between 0 and 100) that must be successfully migrated; if this threshold is not reached, Federator will generate an alert. Leaving this blank means Federator will never generate alerts when resources fail to migrate. This is in effect when the group is not part of a policy interval; when in a policy interval, the policy interval's settings have precedence.

4. Click **Create**. The new resource group is created and appears in the list of static resource groups.

Viewing or Editing a Static Resource Group

To view or edit an existing resource group:

1. Click the **Static** link under the **Resource Groups** tab.
2. Click the name of the resource group you want to view or edit. It appears on the Show Static Resource Group page.
3. If you want to edit the values, click the **Edit** button to show the Edit Static Resource Group page. (As a shortcut, you can get to the Edit page by clicking the edit (pencil) icon for the group on the list page.)
4. Make any changes you want and click **Update** to save them.

Deleting Static Resource Groups

To delete a static resource group, click the delete (trash can) icon next to its name in the list.

Assigning Resources to Static Resource Groups

Newly created DataSynapse resource groups have no resources associated with them. To make them useful, you need to add resources to them from your Resource Managers. You can do this in two ways: using the Resource Manager-based resources page, or using the rule-based resources page. For details on removing shared resources from resource groups, see Working with Static Resources on page 58.

Finding Resources by Resource Manager

To find resources based on the Resource Manager to which they are currently assigned:

1. Click the **DataSynapse Resources** link under the **Resources** tab. The list of all DataSynapse resources appears.
2. Choose which Resource Manager's resources you want to see. The Resource Manager list shows your DataSynapse Resource Managers. Choose "All" to show resources assigned to all Resource Managers; choose a specific manager to show only resources assigned to that manager.

3. You can further filter the list by using the Match button and its related controls (such as “Match default in configuration” to match resources with configuration named “default”).
4. Select the resources you want to add to the group by clicking their checkboxes. You can click the checkbox in the table header to select all of the resources on the page.
5. Select the resource group you would like to add the resources to from the Add Checked To pop-up list.
6. Click the **Add Checked To** button to add the selected resources to the that group.
7. Repeat these steps to add additional resources to static resource groups. You can add resources from multiple managers to the same group and you can add resources from any given manager to multiple groups; you just cannot add the same resource to multiple groups.

When you add resources to a static resource group, they will be immediately migrated to the current target (manager and configuration) of the group, if they are not already there. If the group is in use by an active policy interval, they will be migrated to that policy interval’s target; if not, they will be migrated to the default manager for the group.

Finding Resources Using Rules

To find resources using resource rules:

1. Go to **Resources > Rule-Based Search**.
2. Specify the criteria for your search as follows:
   - **Target Resource Count:** Specifies the maximum number of resources you want to find. If this many resources do not meet your criteria, only those that do will be shown; otherwise this many matches will be shown.
   - **Resource Managers:** Specify which DataSynapse Resource Managers’ resources should be searched. To add a Resource Manager to the search, select it in the left list and click the right arrow button to move it to the right. To remove one, select it and click the left arrow button. To change the order of the Resource Managers on the right, use the up and down arrow buttons. You must specify at least one Resource Manager for the search.
   - **Search Resource Managers in Order:** Check this box to make the order of the Resource Managers in the list above be significant; if it is checked, then all resources from the first manager in the list will be included in the search results before any resources from the second manager are chosen, and so on. If this is not checked, the resources from all selected Resource Managers

will be searched in a random order to help distribute the results across all Resource Managers. Note that while this is designed to help distribute the burden over the list of Resource Managers, it does not necessarily guarantee an even distribution of that burden.

— **Rules:** Choose a type of rule from the Add a New Rule pop-up menu to add it to the criteria. To remove a rule, click its delete button. You can add only one instance of each rule type, except for custom rules (where you can add as many as you need). The specific rule types are described in detail in Rules for Finding Resources on page 56.

3. When you are satisfied with the criteria, click the Search button. Federator performs your search and takes you to the Rules Search Results page where you can see the list of resources that it found.
4. Select the resources you want to add to the group by clicking their checkboxes. You can click the checkbox in the table header to select all of the resources on the page.
5. Select the resource group you would like to add the resources to from the **Add Checked To** pop-up list.
6. Click the **Add Checked To** button to add the selected resources to the that group.
7. You can also click the **Add All To** button to add all of the search results to a group without having to page through the results.
8. If you want to do a new search, click the **Search Again** button to go back to the search page.
9. Repeat these steps to add additional resources to static resource groups. You can add resources from multiple managers to the same group and you can add resources from any given manager to multiple groups; you just cannot add the same resource to multiple groups.

When you add resources to a static resource group, they will be immediately migrated to the current target (manager and configuration) of the group, if they are not already there. If the group is in use by an active policy interval, they will be migrated to that policy interval's target; if not, they will be migrated to the default manager for the group.

Creating Dynamic Resource Groups

Unlike static resource groups, dynamic resource groups use rules to find available resources at the point when they are being activated. These resources remain part of the dynamic group for the duration of the policy interval. At the end of the policy interval, the resources are returned to their source Resource Managers.

To create a dynamic resource group:

1. Go to **Resource Groups > Dynamic**.
2. Click **New**.
3. Enter the details for the resource group:
   - **Name:** Give the resource group a name. Resource groups are referred to by name when specified in policy intervals. All resource group names must be unique, even across different types of groups.
   - **Description:** Optionally, enter a description for the resource group.
   - **Target Resource Count:** Specifies the maximum number of resources you want the group to include.
   - **Minimum Resource Count:** Specifies the minimum number of resources you want the group to include. If this number cannot be met, then the group will fail to activate.
   - **Resource Managers:** Specify which DataSynapse Resource Managers' resources should be searched. To add a Resource Manager to the search, select it in the left list and click the right arrow button to move it to the right. To remove one, select it and click the left arrow button. To change the order of the Resource Managers on the right, use the up and down arrow buttons. You must specify at least one Resource Manager for the group.
   - **Search Resource Managers in Order:** Check this box to make the order of the Resource Managers in the list above be significant; if it is checked, then all resources from the first manager in the list will be included in the search results before any resources from the second manager are chosen, and so on. If this is not checked, the resources from all selected Resource Managers will be searched in a random order to help distribute the results across all Resource Managers. Note that while this is designed to help distribute the burden over the list of Resource Managers, it does not necessarily guarantee an even distribution of that burden.
   - **Rules:** Choose a type of rule from the Add a New Rule pop-up menu to add it to the criteria. To remove a rule, click its delete button. You can add one instance of each rule type, except for custom rules (where you can add as many as you need). The specific rule types are described in detail in Rules for Finding Resources on page 56.
   - **Default Busy Action:** Specify the action that should be taken when Federator wants to migrate resources in this group but they are busy. Choosing "Force" causes the resources to be migrated regardless of their busy status. Choosing "Abandon" causes Federator to give up on migrating resources that are still busy (taking them out of draining mode as well). This has effect when a policy interval using this group ends and

resources are being migrated back to their original managers; when in a policy interval, the policy interval's settings have precedence.

— **Default Warning Period:** Specify the warning period for resources in this group (in minutes). This is the amount of time before the resources are to be migrated that they will be put into draining mode (see Resource Draining on page 65). This has effect when a policy interval using this group ends and resources are being migrated back to their original managers; when in a policy interval, the policy interval's settings have precedence.

— **Default Success Threshold:** Optionally, specify a percentage of the resources in the group (an integer between 0 and 100) that must be successfully migrated—if this threshold is not reached, Federator will generate an alert. Leaving this blank means Federator will never generate alerts when resources fail to migrate. This has effect when the group is not part of a policy interval; when in a policy interval, the policy interval's settings have precedence.

Click **Create**. The new resource group is created and appears in the list of dynamic resource groups.

Deleting Dynamic Resource Groups

To delete a dynamic resource group, click the delete (trash can) icon next to its name in the list.

Rules for Finding Resources

The Search Resources Using Rules and the Create Dynamic Resource Group pages use lists of rules for defining the search criteria that select resources. To add a rule to the list, choose a type of rule from the **Add a New Rule** pop-up. To remove one, click the delete icon for that rule. (Note that except for custom rules, you can only add one of each type.) Various types of rules are described in the following table:

Table 11 Resource Search Rules

Rule Type	Operators	Description
IP Address	Matches Does not match	Searches using the IP addresses of the resources. You can choose whether the resource's address must match or not match the value you specify. The value is a comma-separated list of one or more IP addresses or IP address ranges. IP addresses can be specified using the wildcard character (*), which matches any sequence of characters. IP address ranges can be specified using CIDR notation. (CIDR, or Classless Inter-Domain Routing, notation, specifies a range of IP addresses using a base address and a subnet mask. The base address is specified in standard "dot" notation followed by a forward slash and then by an integer specifying the number of bits in the subnet mask. For example, 192.169.0.0/22 would specify the range of IPv4 addresses from 192.168.0.0 to 192.168.3.255.)
Operating System	Is one of Is not one of	Searches using the resources' operating system. Select one or more operating systems to match (or exclude) from the list.
Total Memory	Is exactly equal to Is not equal to Is less than or equal to Is less than Is greater than or equal to Is greater than Is between (inclusive) Is not between (exclusive)	Compares the value you supply against the actual value of total memory on the resource's machine. Values specified should be in megabytes. If you choose either of the "between" operators, you must supply two values, for the upper and lower limits.

Table 11 Resource Search Rules (Continued)

Rule Type	Operators	Description
CPU Count	Is exactly equal to Is not equal to Is less than or equal to Is less than Is greater than or equal to Is greater than Is between (inclusive) Is not between (exclusive)	Compares the value you supply against the actual number of CPUs on the resource's machine. If you choose either of the "between" operators, you must supply two values, for the upper and lower limits.
Configuration	Matches Does not match	Compares the resources' configuration names against the value or values specified. You specify one or more values in a comma-separated list. Each value may include the wildcard character (*), which matches any sequence of characters.
Busy State	Is busy Is not busy	Compares the resources' busy state. (Note that if any of the Engines on the resource is busy, then the entire resource is deemed busy for the purposes of this rule; therefore resources with several Engine instances will seldom appear to Federator as not busy.)
Other Property	Matches Does not match Is exactly equal to Is not equal to Is less than or equal to Is less than Is greater than or equal to Is greater than Is between (inclusive) Is not between (exclusive) Is present Is not present	Compares the Engine Daemon property whose name you specify against the value or values you provide. The standard properties available from GridServer and Silver Fabric Engines can be specified, as well as any custom properties that have been defined on your grids. If you choose either of the "matches" operators, the value you supply should be a comma-separated list of one or more possible values. Each value may include the wildcard character (*), which matches any sequence of characters. If you choose the numeric operators, the value should be an appropriate number (integer or floating point) and the property's value must also be numeric. If you choose either of the "present" operators, there is no value to specify—the rule will match any resource whose list of properties includes the one you name, regardless of its value.

Working with Static Resources

Once resources have been shared and made part of a resource group, you can view and manipulate them using the Static Resources page.

To see the shared resources in your system, go to **Resources > Static Resources**.

If you would like to remove resources from a group, effectively un-sharing them, select them in the list and click the **Ungroup Checked** button. If resources have been abandoned during a migration, you can see this in the Status column. To reclaim them, select them and click the **Reclaim Abandoned** button. For more about abandoned resources, see Reclaiming Abandoned Resources Manually on page 65.

Scheduling Resource Migration

The main purpose of Federator is to enable movement of resources (Engine Daemons) associated with your Resource Managers. You can move resources from one Resource Manager to another (including between GridServer and Silver Fabric and between different versions of each product) and you can add virtual resources into the pool available to your clouds. This movement is generally done using a daily schedule, although ad hoc movement is also possible.

About Policy Intervals

Federator's schedule consists of a collection of named *policy intervals*. There are two types of policy intervals: **Placement**, and **Demand-Based**.

Placement policy intervals consist of a time range, the resource group to migrate during that time, and the target Resource Manager to migrate to. Based on a policy interval, Federator will make the resources in the resource group available to the target manager, moving them to that manager from another if necessary or creating them in the case of virtual resources. At any given time, there can be a policy interval in effect for each of Federator's resource groups (whether DataSynapse and virtual), but policy intervals involving the same resource group cannot overlap in time.

Demand-based policy intervals consist of a time range, a priority, a set of rules, and the target Resource Manager to which to migrate. Demand policies do not use resource groups, but instead choose resources for the target manager based upon the Service with the most pending tasks. When a condition is satisfied and an action is required in the policy rule, resources are chosen to migrate that satisfy the discriminators from the Service with the most pending tasks. These resources are not chosen from static resource groups or currently assigned dynamic resources. Demand policies will only pick unassigned resources or resources from a lower priority demand policy interval.

Demand-based policies only work on Engines that are not part of a resource group. If you have Engines which are part of a static or dynamic group, they are ineligible for consideration by a demand-based policy. Use a placement policy to move groups and a demand policy to move loose Engines.

When the DataSynapse resource group is not referenced by an active policy interval, Federator attaches the group to its default manager.

If a second policy interval using the same resource group starts at the end of the first policy interval, the second policy interval's settings take precedence over the resource group's defaults; the resources are migrated as the second policy interval dictates and don't return to the group's default Resource Manager.

Creating Policy Intervals

To create a policy interval:

1. Click the **Policy Intervals** tab.
2. Click **New**.
3. Enter values for the policy interval:

Table 12 Policy Interval Parameters

Value	Description
Name	The name of the policy interval.
Description	An optional description of the policy interval.
Type	Select **Placement** or **Demand-Based**.
Resource Group	The group whose resources are affected by this policy interval. Each policy interval affects exactly one group's resources. This may be either a DataSynapse resource group (static or dynamic) or a virtual one. This option is only available for Placement policy intervals.
Target Manager	The DataSynapse Resource Manager that should own the resources for this policy interval.

Table 12 Policy Interval Parameters (Continued)

Value	Description
Start	This contains parameters that affect the way the policy interval behaves when coming into effect: • **Time:** When the policy interval takes effect. It can be set in increments of five minutes. • **Warning Period:** The start warning period for this policy interval (in minutes). When resources are migrated for the policy interval, they are put into draining mode, preventing them from accepting new jobs from their current Resource Manager for this period of time (see Resource Draining on page 65 for details). If, after this time, they are still busy, then they are dealt with according to the If Still Busy parameter. • **If Still Busy:** The action taken when Federator wants to migrate resources targeted by this policy interval but they are busy processing tasks for their current Resource Manager. **Force** causes the resources to be migrated regardless of their busy status. **Abandon** causes Federator to give up on migrating resources that are still busy (taking them out of draining mode as well). Note that this action only occurs if the resources are still busy after the start warning period has expired. • **Initial Configuration:** You can optionally specify the name of the configuration to use when the Engine Daemons first start up. Use this field, for example, if you want them to synchronize with a particular Resource Manager but then connect to their "real" Resource Manager (as defined by the target configuration) afterwards. • **Target Configuration:** Specify the name of the configuration to use when the Engine Daemons are running. • **Success Threshold:** Optionally, specify a percentage of the resources in the group (an integer between 0 and 100) that must be successfully migrated—if this threshold is not reached, Federator will generate an alert. Leaving this blank means Federator will never generate alerts when resources fail to migrate.

Table 12 Policy Interval Parameters (Continued)

Value	Description
End	The End group contains fields that affect the way the policy interval behaves when ending: • **Time:** This is the time of day the policy interval ends. It can be set in increments of five minutes. • **Warning Period:** Specify the end warning period for this policy interval (in minutes). Resources will be put into draining mode, preventing them from accepting new jobs from their current Resource Manager, this many minutes before the policy interval ends (see Resource Draining on page 65 for details). If they are still busy after this time and if the resource group is not in a new policy interval, they are either returned to their default manager (if the default busy action for the group is set to "Force") or taken out of draining mode and left connected to the current Resource Manager (if the default busy action for the group is "Abandon").
Allocation Priority	The priority of this policy interval over other policy intervals: "Critical", "High", "Medium" or "Low". Higher priority intervals will be evaluated before lower priority intervals. This option is only available for Demand-based policy intervals.

Table 12 Policy Interval Parameters (Continued)

Value	Description
Allocation Rules	Defined demand-based rules, which are evaluated in order until the first condition is satisfied and the corresponding action is taken. This option is only available for Demand-based policy intervals. Click **Add Rule** to add more rules to the policy interval. The following rules are available: • **Statistic**: The value reported by the Resource Manager examined by this rule: — **Pending Invocation Count**: The number of pending tasks. — **Average Invocation Duration**: The average time it takes to complete a task. — **Predicted Time to Complete**: The estimated time to complete all the pending tasks. The value is (Average Task Duration * Pending Task Count) / Engine Count. • **Operator**: The relationship between the Statistic and Value. The following operators are available: “<“, “<=”, “=”, “>=”, “>”. • **Value**: The value against which the statistic will be compared. It must be an integer greater than or equal to zero. • **Action**: The action to take on the provided number of resources: — **Return**: causes the number of specified resources to be migrated back to their original Resource Managers immediately. — **Release**: causes the number of specified resources to be released from the current Resource Manager, making them eligible for other Resource Managers to take (though they will remain on their current Manager until another Manager needs them). — **Do nothing**: causes no resource migration for the current cycle. — **Migrate (abandon if busy)**: migrates the number of specified resources to the policy’s Resource Manager using the Abandon heuristic. — **Migrate (forcing if busy)**: migrates he number of resources specified by the Resource Count field to the policy’s Resource Manager using the Force heuristic. • **Resource Count**: the number of resources that should be affected by the action. It must be a positive integer. If the action is "do nothing," then the Resource Count field is not shown.

4. Click **Create** to create the policy interval.

It is possible to create Allocation Rules that conflict with each other or cause unintended behavior. For example, you can create an on-demand policy that migrates resources when the pending invocation count is over a given number, and returns resources when pending invocation count is equal to zero. If some tasks are resubmitted, the second rule could have already started to occur. This means the pending invocation count would go up from 0, but possibly not high enough to trigger migration of more resources.

Viewing or Editing a Policy Interval

To view or edit an existing policy interval:

1. Click the **Policy Intervals** tab.
2. Click the name of the policy interval you want to view or edit. It appears on the Show Policy Interval page.
3. If you want to edit the values, click the **Edit** button to show the Edit Policy Interval page. (As a shortcut, you can get to the Edit page by clicking the edit (pencil) icon for the policy interval on the list page.)
4. Make any changes you want and click **Update** to save them. The fields are the same as when creating a new policy interval.

Deleting Policy Intervals

To delete a policy interval, click the delete (trash can) icon next to its name in the list.

Enabling or Disabling a Policy Interval

You can enable and disable policy intervals. Federator only schedules enabled policy intervals. If a currently active policy interval is disabled, it behaves the same way as if it had reached its end time.

To enable or disable a policy interval:

1. Click the **Policy Intervals** tab.
2. Click the name of the policy interval you want to modify. It appears on the Show Policy Interval page.
3. Click **Enable** or **Disable** to modify the policy interval (if the policy interval is currently enabled, the button says “Disable” and vice-versa). As mentioned above, disabling a policy interval that is currently in effect will stop it. (As a

shortcut, you can click the enable or disable icon for the policy interval on the list page.)

Resource Draining

Because Engines can be busy with important work when a move is scheduled or otherwise indicated, it is not always desirable to move Engines immediately when a policy interval begins. Federator cooperates with your Resource Managers to let Engines complete their current tasks before assigning them further work. This process is called draining. Draining occurs during a warning period that begins at the time of any resource group-related move. Idle resources, and those that become idle during the warning period, always move. The "if still busy" property of policy intervals and resource groups lets you determine what happens to resources that are not idle by the end of a warning period.

There are two choices:

- **Force** directs Federator to move the resource even though it's still busy; any tasks it may have been working on are discarded (the Broker that assigned them will reassign them to another Engine automatically).
- **Abandon** directs Federator to leave the busy resource in place and tells its current Resource Manager that it no longer needs to drain, and can once again accept new tasks. Note that abandonment is not permanent. It lasts only until the next policy transition or other Federator-directed move.

FabricServer 2.6 does not implement resource draining. This makes the Busy Action especially important when moving resources away from FabricServer 2.6 grids. In such moves, Federator monitors the resources scheduled for movement during the warning period to see if any of them go idle, but in most cases moving those resources away from that FabricServer requires the Force busy action. (FabricServer 3.0 and Silver Fabric 4.0 and newer includes this feature and resource draining there works as described above.)

Reclaiming Abandoned Resources Manually

There may be occasions where Federator abandons a resource that stays busy through a warning period, but business necessity requires that you override the general abandon setting. You can try again to move abandoned resources manually.

To move abandoned resources manually:

1. Click the **Static Resources** tab.
2. Locate the Static Resources with status of Abandoned. (You can filter the list, matching the value "Abandoned" for the Status field.)

3. Select the check boxes that correspond to the abandoned resources you would like to retry.
4. Click **Reclaim Abandoned**. The resources begin draining again and, when they become idle, Federator moves them according to the policy interval or resource group settings currently in effect.

Moving Resources Manually

There may be cases where you want to move selected resources on a one-time basis from one cloud to another. Manually moving enables you to migrate resources from one Resource Manager to another without making those resources part of a shared resource group and without scheduling them to move on a regular basis.

When you designate a resource for a manual move, Federator removes it from the resource group it belongs to, if any. It remains ungrouped after the move until you add it back into a resource group.

To manually move resources between DataSynapse resource mangers:

1. Click the **DataSynapse** link under the **Resource Managers** tab.
2. Find the DataSynapse Resource Manager that owns the resources you'd like to share. Click the number in the Resources column for that row. The Resources page appears. (You can also navigate to the All Resources or Static Resources page on the Resources tab.)
3. Select the checkboxes corresponding to the resources you want to move. You can click the checkbox in the table header to select all of the resources on the page.
4. Click **Move Checked**.
5. On the Resources to Move page, specify the following parameters about the destination:
   - **Target Manager:** Choose the DataSynapse Resource Manager to which you want to move the resources.
   - **Initial Configuration:** You can optionally specify the name of the configuration to use when the Engine Daemons first start up. Use this field, for example, if you want them to synchronize with a particular Broker but then connect to their "real" Broker (as defined by the target configuration) afterwards.
   - **Target Configuration:** Specify the name of the configuration to use when the Engine Daemons are running.
   - **Warning Period:** Specify the warning period for the move, in minutes. Resources will be put into draining mode, preventing them from accepting new jobs from their current Resource Manager for this period of time (see Resource Draining on page 65 for details). If, after this time, they are still

busy, then they are dealt with according to the “if still busy” action (see below).

— **If Still Busy:** Specify the action that should be taken resources are still busy processing tasks for their previous Resource Manager after the warning period is over. Choosing “Force” causes the resources to be migrated regardless of their busy status. Choosing “Abandon” causes Federator to give up on migrating resources that are still busy (taking them out of draining mode as well).

6. Click **Move**. Federator begins migrating the selected resources.

Resource Migration Details

Depending on what is changing for the resource as it is migrated, the specifics of what Federator does can vary. The parameters that affect what happens during migration are the target Resource Manager, the target configuration, and the initial configuration, if any. Different scenarios are discussed below, based on which of these parameters are changing.

Target Configuration Changes, Target Manager Doesn't

Federator:

1. Puts the resource into draining mode and waits until the resource is idle or the warning period has expired
2. Sets the new target configuration for the resource, causing it to restart (if the resource is idle or the busy action is "Force")
3. Takes the resource out of draining mode (if the resource has been abandoned)

Note that the initial configuration is ignored here, since the resource will have already synchronized with this Resource Manager.

Target Manager Changes, Configuration Doesn't, No Initial Configuration

Federator:

1. Puts the resource into draining mode and waits until the resource is idle or the warning period has expired
2. Sets the new target Resource Manager for the resource, causing it to restart (if the resource is idle or the busy action is "Force")
3. Takes the resource out of draining mode (if the resource has been abandoned)

Target Manager Changes, Configuration Changes, No Initial Configuration

Federator:

1. Puts the resource into draining mode and waits until the resource is idle or the warning period has expired
2. Disables the resource (if the resource is still busy and the busy action is "Abandon," the resource is abandoned—skip to Step 6)
3. Sets the new target Resource Manager for the resource, causing the Daemon to restart
4. Sets the new configuration for the resource, causing the Daemon to restart

5. Enables the resource
6. Takes the resource out of draining mode (if the resource has been abandoned)

Note that the Engine Daemon needs to restart multiple times for this type of migration.

Target Manager Changes, Configuration Changes, Initial Configuration Set

Federator:

1. Puts the resource into draining mode and waits until the resource is idle or the warning period has expired
2. Disables the resource (if the resource is still busy and the busy action is "Abandon," the resource is abandoned—skip to Step 7)
3. Sets the new target Resource Manager for the resource, causing the Daemon to restart
4. Sets the configuration for the resource to the initial configuration, causing the Daemon to restart
5. Enables the resource for synchronization
6. After synchronization, sets the configuration for the resource to the target configuration, causing the Daemon to restart
7. Takes the resource out of draining mode (if the resource has been abandoned)

Note that the Engine Daemon needs to restart multiple times for this type of migration.

Using EC2 Resource Groups

Like DataSynapse resource groups, EC2 resource groups provide resources that Federator can assign to the clouds it manages. Unlike DataSynapse resource groups, the resources don't actually exist, except when Federator requests them.

Creating EC2 Resource Groups

To create an EC2 resource group:

1. Click the **Amazon EC2** link under the **Resource Groups** tab.
2. Click **New**.
3. Enter the details for the resource group:

Table 13 Resource Group Properties

Property	Description
Name	The resource group's name. Resource groups are referred to by name when specified in policy intervals.
Description	An optional description for the resource group.
Default Warning Period	The warning period for resources in this group (in minutes). This is the amount of time before the resources are to be migrated that they will be put into draining mode (see Resource Draining on page 65). This can be overridden when you create policy intervals.
Amazon EC2 Manager	The EC2 Resource Manager that will be used to create resource instances for this group. (Federator uses the credentials—AWS access key, secret access key, region—of this manager to communicate with the EC2 server when dealing with this resource group.)
Resource Count	The number of resources (Amazon EC2 instances running Engine Daemons) that the group contains. When this group is activated in a policy interval, this many instances will be started.
AWS Image	The Amazon Machine Image (AMI) to use for the Engine Daemons launched as part of this group.

Table 13 Resource Group Properties (Continued)

Property	Description
Instance Type	The instance type to use when running the proxy server. The instance type determines how much processing power the instance will have. Standard instances may be small, large, or extra large. High-CPU instances may be medium or extra large. (For specific details on the instance types and their pricing, see the EC2 web site.) Note that the values in this list change to reflect the types appropriate for the image you have chosen.
Security Groups	You can choose one or more of the EC2 security groups defined in your account for the instances to run in. See Creating Security Groups, page 28 for more.
Key Pair Name	The key pairs associated with your account. Instances started as part of this resource group will use this key pair to secure themselves.
Spot Price	An optional spot price you are willing to pay for this instance. See About Spot Instances on page 38 for more information on Spot Instances.
Availability Zone	The availability zone for the proxy server. Choose "Automatic" to let Amazon choose for you at start-up time. Note that you choosing to run the Engine instances in the same availability zone as the EC2 Resource Manager's proxy server can avoid inter-zone networking fees. See Amazon's fee schedule for details.
VPC Subnet ID	The subnet used for the Virtual Private Network.
User Data	You should only enter user data here if you are using a custom AMI for the engines and it requires this field.
Kernel ID	You should only enter a kernel ID here if you are using a custom AMI for the engines and it requires this field.
Ram Disk ID	You should only enter a Ram Disk ID here if you are using a custom AMI for the engines and it requires this field.

Table 13 Resource Group Properties (Continued)

Property	Description
Block Device Mappings	This field maps block devices from Amazon Elastic Block Store (EBS) to devices visible to the instance. The format is the same as for the `run-instances` command in the command-line interface—see Amazon's documentation for details.

4. Click **Create**. The new resource group is created and appears in the list of EC2 resource groups.

Viewing or Editing an EC2 Resource Group

To view or edit an existing resource group:

1. Go to **Resource Groups > Amazon EC2**.
2. Click the name of the resource group you want to view or edit.
3. If you want to edit the values, click the **Edit** button to show the Edit Amazon EC2 Resource Group page. (As a shortcut, you can get to the Edit page by clicking the edit (pencil) icon for the group on the list page.)
4. Make any changes you want and click **Update** to save them.

Deleting EC2 Resource Groups

To delete an EC2 resource group, click the delete (trash can) icon next to its name in the list.

Using EC2 Demandable Resources

Defining an EC2 resource group is similar to defining a static resource group, in that the resources are not demand allocated. To use EC2 resources on demand, you can define *EC2 Demandable Resources*. Demandable EC2 resources are associated with a defined EC2 Resource Manager, but not in a group, so they can participate in demand-based allocation policies just like any other un-grouped resource.

When more resources are required based on the current runtime environment and settings defined for the EC2 Demand-Based policy, the Federator will first attempt to use EC2 Demandable Resources that are already running but which have not been assigned to a Manager instance. If more resources are needed more EC2 Demandable Resources will be started until either the policy requirements are met, or the EC2 instance quota reaches a limit.

When EC2 Demandable Resources are no longer needed, those excess resources will be shut down after a configurable timeout period. EC2 Demandable Resources can also be assigned (by specific policy setting) to another Resource Manager, at which time their timeout is reset.

EC2 demandable resources are not considered for dynamic groups and cannot be added to static groups.

To define EC2 Demandable Resources:

1. Define an EC2 Resource Manager, if you have not done so already.
2. Go to **Resources > EC2 Demandable**.
3. Click **New**.
4. Enter the following values:

Table 14 EC2 Demandable Resources Properties

Value	Description
Amazon EC2 Manager	The EC2 Resource Manager used for the resources.
AWS Image	The Amazon Machine Image (AMI) to use for the resources. (Note that this list is filtered to only show images with “proxy” in their manifest name.)

Table 14 EC2 Demandable Resources Properties (Continued)

Value	Description
Instance Type	The instance type to use when running the resources. The instance type determines how much processing power and disk space the instance will have. Standard instances may be small, large, or extra large. High-CPU instances may be medium or extra large. Note that the values in this list change to reflect the types appropriate for the image you have chosen. For specific details on the instance types and their pricing, see the EC2 web site.
Security Groups	You can choose one or more of the EC2 security groups defined in your account for the resources to run in. (See Creating Security Groups on page 28)
Key Pair Name	The key pairs associated with your account. Instances started as part of this resource group will use this key pair to secure themselves.
Spot Price	An optional spot price you are willing to pay for this instance. See About Spot Instances on page 38 for more information on Spot Instances.
Availability Zone	The availability zone for the resources. Choose **Automatic** to let Amazon choose for you at start-up time.
VPC Subnet ID	The subnet used for the Virtual Private Network.
User Data	Optional user data; only used when you are using a custom AMI for the instance and it requires this field.
Kernel ID	An optional kernel ID; only used when you are using a custom AMI for the instance and it requires this field.

Table 14 EC2 Demandable Resources Properties (Continued)

Value	Description
RAM Disk ID	An optional Ram Disk ID; only used when you are using a custom AMI for the instance and it requires this field.
Block Device Mappings	Optional block device mappings; only used when you are using a custom AMI for the instance and it requires this field. This field maps block devices from Amazon Elastic Block Store (EBS) to devices visible to the instance. The format is the same as for the `run-instances` command in the command-line interface—see Amazon's documentation for details.

5. EC2 Demandable Resources will use the Startup Timeout and Shutdown Threshold settings as defined on the **Admin > General** page.
6. Click **Create**.

Using VMware Resource Groups

Like DataSynapse resource groups, VMware resource groups provide resources that Federator can assign to the clouds it manages. Unlike DataSynapse resource groups, the resources don't actually exist, except when Federator requests them.

Creating VMware Resource Groups

To create a VMware resource group:

1. Go to **Resource Groups > VMWare** in the Type column.
2. Click **New**.
3. Enter the details for the resource group:

Table 15 VMware Resource Group Properties

Property	Description
Name	The resource group's name. Resource groups are referred to by name when specified in policy intervals.
Description	An optional description for the resource group.
Default Success Threshold	Optionally, specify a percentage of the resources in the group (an integer between 0 and 100) that must be successfully migrated; if this threshold is not reached, Federator will generate an alert. Leaving this blank means Federator will never generate alerts when resources fail to migrate. This is in effect when the group is not part of a policy interval; when in a policy interval, the policy interval's settings have precedence.
Default Warning Period	The warning period for resources in this group (in minutes). This is the amount of time before the resources are to be migrated that they will be put into draining mode (see Resource Draining on page 65). This can be overridden when you create policy intervals.
VMware Resource Manager	The VMware Resource Manager that will be used to create resource instances for this group. (Federator uses the URL, username, and password of this manager to communicate with the VMware server when dealing with this resource group.)

Table 15 VMware Resource Group Properties (Continued)

Property	Description
Resource Count	The number of resources (VMware instances running Engine Daemons) that the group contains. When this group is activated in a policy interval, this many instances will be started.
Datacenter	The primary container of inventory objects such as hosts and virtual machines. Typically, you can add hosts, folders, and clusters to a datacenter.
Datastore	A logical container that holds virtual machine files and other files necessary for virtual machine options. The datastore chosen is what we use for the provisioned storage for created VMs.
Template Folder Path	The directory (subtree of folders) in which Federator will look for templates to choose for the Default Template option.
Default Template	The template from which virtual machines will be created.
VM Folder Path	The directory (subtree of folders) in which Federator will create virtual machines.
Compute Resource	A specific host or a cluster in the Datacenter on which VMs will run. Clusters are simply groups of hosts.
Resource Pool	Resource Pools can be used to hierarchically partition available CPU and memory resources of a standalone host or cluster.
VM Name	An optional parameter used by Federator to help you more easily identify virtual machines that have been created for a specific resource group or demand resource.
DNS Server	The DNS server that the virtual machines will use to resolve hosts after creation.
Domain	The domain used by virtual machines.
Gateway	The gateway used by virtual machines.
Subnet	The subnet used by virtual machines.

Table 15 VMware Resource Group Properties (Continued)

Property	Description
Min Free Datastore Space	The minimum amount of available space on the chosen datastore that we require to be able to create VMs.

4. Click **Create**. The new resource group is created and appears in the list of VMware resource groups.

Viewing or Editing a VMware Resource Group

To view or edit an existing resource group:

1. Go to **Resource Groups > VMware**.
2. Click the name of the resource group you want to view or edit.
3. If you want to edit the values, click the **Edit** button to show the Edit VMware Group page. (As a shortcut, you can get to the Edit page by clicking the edit (pencil) icon for the group on the list page.)
4. Make any changes you want and click **Update** to save them.

Deleting VMware Resource Groups

To delete a VMware resource group, click the delete (trash can) icon next to its name in the list.

Using VMware Demandable Resources

Defining a VMware resource group is similar to defining a static resource group, in that the resources are not demand allocated. To use VMware resources on demand, you can define *VMware Demandable Resources*. Demandable VMware resources are associated with a defined VMware Resource Manager, but not in a group, so they can participate in demand-based allocation policies just like any other un-grouped resource.

When a VMware Demand-Based policy needs resources, it will first check for running VMware Demandable Resources that are not already assigned to a Manager. If more are needed, unstarted VMware demandable resources will be started until the policy is fully filled or the VMware instance quota is filled. When they are no longer needed, resources will be shut down after a configurable timeout period, or can be assigned by policy to another Resource Manager, at which time their timeout is reset.

VMware demandable resources are not considered for dynamic groups and cannot be added to static groups.

To define VMware Demandable Resources:

1. Define a VMware Resource Manager, if you have not done so already.
2. Go to **Resources > VMware Demandable**.
3. Click **New**.
4. Enter the following values:

Table 16 VMware Demandable Resources Properties

Value	Description
VMware Manager	The VMware Resource Manager used for the resources.
Datacenter	The primary container of inventory objects such as hosts and virtual machines. Typically, you can add hosts, folders, and clusters to a datacenter.
Datastore	A logical container that holds virtual machine files and other files necessary for virtual machine options. The datastore chosen is what we use for the provisioned storage for created VMs.

Table 16 VMware Demandable Resources Properties (Continued)

Value	Description
Template Folder Path	The directory (subtree of folders) in which Federator will look for templates to choose for the Default Template option.
Default Template	The template from which virtual machines will be created.
VM Folder Path	The directory (subtree of folders) in which Federator will create virtual machines.
Compute Resource	A specific host or a cluster in the Datacenter on which VMs will run. Clusters are simply groups of hosts.
Resource Pool	Resource Pools can be used to hierarchically partition available CPU and memory resources of a standalone host or cluster.
VM Name	An optional parameter used by Federator to help you more easily identify virtual machines that have been created for a specific resource group or demand resource.
DNS Server	The DNS server that the virtual machines will use to resolve hosts after creation.
Domain	The domain used by virtual machines.
Gateway	The gateway used by virtual machines.
Subnet	The subnet used by virtual machines.
Min Free Datastore Space	The minimum amount of available space on the chosen datastore that we require to be able to create VMs.

5. VMware Demandable Resources will use the Startup Timeout and Shutdown Threshold settings as defined on the **Admin > General** page.
6. Click **Create**.

Appendix A Web Services API Reference

This chapter describes the Federator Web Services API.

Topics

The Federator Web Services API

Most operations available in Federator's user interface are also exposed through SOAP web services. The web services API is composed of four interfaces.

- `policyIntervalAdmin`
- `resourceGroupAdmin`
- `resourceManagerAdmin`
- `sharedResourceAdmin`
- `demandableResourceAdmin`

The WSDLs for each service can be found at `http://`*host*`:`*port*`/federator/services/service-name?wsdl`. For example, if your hostname is `example` and Federator is running on port 8080, the `sharedResourceAdmin` web service is at `http://example:8080/federator/services/sharedResourceAdmin?wsdl`

Requests to the web services interface, like all other requests to Federator, require proper credentials. You must specify a valid Federator username and password both when requesting the WSDL and when executing methods on the web services interfaces.

policyIntervalAdmin

The `policyIntervalAdmin` service enables you to list, create, update and remove policy intervals. The supported methods are:

PolicyIntervalInfo[] getAllPolicyIntervalInfo()

Return information about the policies known to Federator.

PolicyIntervalInfo getPolicyIntervalInfoByName(String name)

Return information about the policy with the given name.

void addPolicyInterval(PolicyIntervalInfo info)

Create a new policy as specified by the `PolicyIntervalInfo` object passed in.

void deletePolicyInterval(PolicyIntervalInfo info)

Delete the policy whose ID is specified in `info`.

void updatePolicyInterval(PolicyIntervalInfo info)

Update the policy to match the edited `PolicyIntervalInfo` object.

void enablePolicy(String policyName)

Enable the policy with the specified name.

void disablePolicy(String policyName)

Disable the policy with the specified name.

resourceGroupAdmin

The `resourceGroupAdmin` service enables you to manage resource groups. The following methods are supported:

DatasynapseResourceGroupInfo[] getAllDatasynapseResourceGroupInfo()

Return information about the static DataSynapse resource groups known to Federator.

DatasynapseResourceGroupInfo getDatasynapseResourceGroupInfoByName(String name)

Return information about the DataSynapse resource group with the given name.

void addDatasynapseResourceGroup(DatasynapseResourceGroupInfo info)

Create a new DataSynapse resource group as specified by the `DatasynapseResourceGroupInfo` passed in.

void updateDatasynapseResourceGroup(DatasynapseResourceGroupInfo info)

Update the DataSynapse resource group to match the edited `DatasynapseResourceGroupInfo` object.

DynamicResourceGroupInfo[] getAllDynamicResourceGroupInfo()

Return information about the dynamic resource groups known to Federator.

DynamicResourceGroupInfo getDynamicResourceGroupInfoByName(String name)

Return information about the dynamic resource group with the given name.

void addDynamicResourceGroup(DynamicResourceGroupInfo info)

Create a new dynamic resource group as specified by the `DynamicResourceGroupInfo` passed in.

void updateDynamicResourceGroup(DynamicResourceGroupInfo info)

Update the dynamic resource group to match the edited `DynamicResourceGroupInfo` object.

void addResource(DatasynapseResourceGroupInfo info, String resourceId)

Add a resource with the given resource ID to the DataSynapse resource group whose ID is specified in `info`.

void removeResource(DatasynapseResourceGroupInfo info, String resourceId)

Add the resource with the given resource ID from the DataSynapse resource group whose ID is specified in `info`.

void deleteResourceGroup(long id)

Deletes the resource group with the specified ID. Applies to all types of resource groups.

VmwareResourceGroupInfo[] getAllVmwareResourceGroupInfo()

Return information about the VMware resource groups known to Federator.

VmwareResourceGroupInfo getVmwareResourceGroupInfoByName(String name)

Return information about the VMware resource group with the given name.

void addVmwareResourceGroup(VmwareResourceGroupInfo info)

Create a new VMware resource group as specified by the `VmwareResourceGroupInfo` passed in.

void updateVmwareResourceGroup(VmwareResourceGroupInfo info)

Update the VMware resource group to match the edited `VmwareResourceGroupInfo` object.

Ec2ResourceGroupInfo[] getAllEc2ResourceGroupInfo()

Return information about the Amazon EC2 resource groups known to Federator.

Ec2ResourceGroupInfo getEc2ResourceGroupInfoByName(String name)

Return information about the Amazon EC2 resource group with the given name.

void addEc2ResourceGroup(Ec2ResourceGroupInfo info)

Create a new Amazon EC2 resource group as specified by the `Ec2ResourceGroupInfo` passed in.

void updateEc2ResourceGroup(Ec2ResourceGroupInfo info)

Update the Amazon EC2 resource group to match the edited `Ec2ResourceGroupInfo` object.

resourceManagerAdmin

The `resourceManagerAdmin` service enables you to list, create, update and remove Resource Managers. The supported operations on this interface are:

DatasynapseResourceManagerInfo[] getAllDatasynapseResourceManagerInfo()

Return an array of `DatasynapseResourceManagerInfo` objects describing all of the DataSynapse Resource Managers known to Federator.

DatasynapseResourceManagerInfo getDatasynapseResourceManagerInfoByName(String name)

Return information about the named DataSynapse Resource Manager.

void addDatasynapseResourceManager(DatasynapseResourceManagerInfo

info)

Create a new DataSynapse Resource Manager as specified by the `DatasynapseResourceManagerInfo` passed in.

void updateDatasynapseResourceManager(DatasynapseResourceManagerInfo info)

Update the DataSynapse Resource Manager to match the edited `DatasynapseResourceManagerInfo` object.

void deleteResourceManager(long id)

Delete the Resource Manager with the specified ID. Applies to all types of Resource Managers.

void enableResourceManager(String name)

Enable the Resource Manager with the specified name. Applies to all types of Resource Managers.

void disableResourceManager(String name)

Disable the Resource Manager with the specified name. Applies to all types of Resource Managers.

ResourceInfo[] getAllResourceInfo()

Return information about all of the resources known to Federator.

ResourceInfo[] getResourceInfoByManager(String name)

Return information about the resources known to Federator that belong to the Resource Manager with the given name.

VmwareResourceManagerInfo[] getAllVmwareResourceManagerInfo()

Return an array of `VmwareResourceManagerInfo` objects describing all of the VMware resource managers known to Federator.

VmwareResourceManagerInfo getVmwareResourceManagerInfoByName(String name)

Return information about the named VMware resource manager.

void addVmwareResourceManager(VmwareResourceManagerInfo info)

Create a new VMware resource manager as specified by the `VmwareResourceManagerInfo` passed in.

void updateVmwareResourceManager(VmwareResourceManagerInfo info)

Update the VMware resource manager to match the edited `VmwareResourceManagerInfo` object.

Ec2ResourceManagerInfo[] getAllEc2ResourceManagerInfo()

Return an array of `Ec2ResourceManagerInfo` objects describing all of the Amazon EC2 Resource Managers known to Federator.

Ec2ResourceManagerInfo getEc2ResourceManagerInfoByName(String name)

Return information about the named Amazon EC2 Resource Manager.

void addEc2ResourceManager(Ec2ResourceManagerInfo info)

Create a new Amazon EC2 Resource Manager as specified by the `Ec2ResourceManagerInfo` passed in.

void updateEc2ResourceManager(Ec2ResourceManagerInfo info)

Update the Amazon EC2 Resource Manager to match the edited `Ec2ResourceManagerInfo` object.

sharedResourceAdmin

The `sharedResourceAdmin` service enables you to list shared resources. It also lets you manually move resources not associated with a resource group. The supported methods are:

SharedResourceInfo[] getAllSharedResourceInfo()

Return information about the shared resources known to Federator.

SharedResourceInfo[] getSharedResourceInfoByGroup(String group)

Return information about the shared resources belonging to the resource group with the given name.

SharedResourceInfo[] getSharedResourceInfoByManager(String currentManager)

Return information about the shared resources assigned to the DataSynapse Resource Manager with the given name.

void reclaimResource(SharedResourceInfo info)

Migrate the specified shared resource to its default manager.

void moveResource(String resourceId, String targetManager, String targetConfig, int warningPeriodMinutes, String busyAction)

Migrate the specified resource to the target DataSynapse Resource Manager specified by `targetManager`. The `targetConfig`, `warningPeriodMiniutes`, and `busyAction` parameters are similar to their counterparts in the `PolicyIntervalInfo` object.

demandableResourceAdmin

The `demandableResourceAdmin` service enables you to add, update, and list demandable resources. The supported methods are:

public void addDemandable(Ec2DemandableInfo info)

Adds the specified demandable resource.

public void updateDemandable(Ec2DemandableInfo info)

Updates the specified demandable resource.

public void deleteDemandable(long id)

Deletes the specified demandable resource.

public Ec2DemandableInfo[] getAllEc2DemandableInfo()

Return information about all demandable resources known to Federator.

public void addDemandable(VmwareDemandableInfo info)

Adds the specified demandable resource.

public void updateDemandable(VmwareDemandableInfo info)

Updates the specified demandable resource.

public VmwareDemandableInfo[] getAllVmwareDemandableInfo()

Return information about all demandable resources known to Federator.

Classes Used by the API

The web service API uses various objects to describe the Federator artifacts you can manipulate. These are essentially generic Java beans, with standard getter and setter methods to access their properties. Properties whose values are constrained beyond their Java type are described along with their accessors.

DatasynapseResourceGroupInfo

The `DatasynapseResourceGroupInfo` object is used by the `sharedResourceAdmin` service. It is defined as follows:

```
long getId()
String getName()
void setName(String name)
String getDescription()
void setDescription(String description)
String getResourceManager()
void setResourceManager(String resourceManager)
String getInitialConfig()
void setInitialConfig(String initialConfig)
String getDefaultConfig()
void setDefaultConfig(String defaultConfig)
String getBusyAction()
void setBusyAction(String busyAction)
  Note: busyAction is one of “Abandon” or “Force”
Integer getDefaultSuccessThreshold()
void setDefaultSuccessThreshold(Integer defaultSuccessThreshold)
  Note: defaultSuccessThreshold must be between 0 and 100
Integer getDefaultWarningPeriod()
void setDefaultWarningPeriod(Integer defaultWarningPeriod)
```

DynamicResourceGroupInfo

The `DynamicResourceGroupInfo` object is used by the `sharedResourceAdmin` service. It is defined as follows:

```
long getId()
String getName()
void setName(String name)
String getDescription()
void setDescription(String description)
String getBusyAction()
void setBusyAction(String busyAction)
  Note: busyAction is one of “Abandon” or “Force”
Integer getDefaultSuccessThreshold()
void setDefaultSuccessThreshold(Integer defaultSuccessThreshold)
  Note: defaultSuccessThreshold must be between 0 and 100
Integer getDefaultWarningPeriod()
void setDefaultWarningPeriod(Integer defaultWarningPeriod)
int getMinResourceCount()
```

```
void setMinResourceCount(int minResourceCount)
int getTargetResourceCount()
void setTargetResourceCount(int targetResourceCount)
boolean getResourceManagersOrdered()
void setResourceManagersOrdered(boolean resourceManagersOrdered)
DatasynapseResourceManagerInfo[] getManagers()
void setManagers(DatasynapseResourceManagerInfo[] managers)
DynamicResourceRuleInfo[] getRules()
void setRules(DynamicResourceRuleInfo[] rules)
```

DynamicResourceRuleInfo

The `DynamicResourceRuleInfo` object is used by the `DynamicResourceGroupInfo` object. It is defined as follows:

```
long getId()
String getType()
void setType(String type)
```
Note: `type` is one of `IP`, `os`, `totalMemInKB`, `cpuNo`, `configurationName`, `busy`, or `custom`
```
String getPropertyName()
void setPropertyName(String propertyName)
```
Note: propertyName is the name of the property's key in the map of values returned from GridServer or Silver Fabric for this resource (except for "custom" rules, propertyName should be the same as type)
```
String getOperator()
void setOperator(String operator)
```
Note: operator is one of matches, equals, lte (≤), gte (≥), inRange, or exists
```
String getValue()
void setValue(String value)
```
Note: value depends on the operator of the rule:
For matches, value is a comma-separated list of potential values
For equals, lte, or gte, value is a numeric value as a string
For inRange, value is a string containing two numeric values separated by .. (e.g. "4..16")
For exists, value is not used and should be set to the empty string ("")
When type is set to busy, value is not used and should be set to the empty string ("")
```
boolean getNegated()
void setNegated(boolean negated)
```

DatasynapseResourceManagerInfo

The `DatasynapseResourceManagerInfo` object is used by the `resourceGroupAdmin` and `resourceManagerAdmin` services. It is defined as follows:

```
long getId()
String getName()
void setName(String name)
String getDescription()
void setDescription(String description)
String getPrimaryManagerUrl()
void setPrimaryManagerUrl(String primaryManagerUrl)
String getSecondaryManagerUrl()
void setSecondaryManagerUrl(String secondaryManagerUrl)
```

```
String getAdminUsername()
void setAdminUsername(String adminUsername)
String getAdminPassword()
void setAdminPassword(String adminPassword)
boolean isEnabled()
void setEnabled(boolean enabled)
```

PolicyIntervalInfo

The `PolicyIntervalInfo` object is used by the `policyIntervalAdmin` service. It is defined as follows:

```
long getId()
boolean isEnabled()
void setEnabled(boolean enabled)
String getName()
void setName(String name)
String getDescription()
void setDescription(String description
String getType()
void setType(String type)
```
Note: type is one of “Placement”, “Demand”
```
String getResourceGroup()
void setResourceGroup(String resourceGroup)
```
Note: resourceGroup is only used for interval type “Placement”
```
String getTargetManager()
void setTargetManager(String targetManager)
String getBusyAction()
void setBusyAction(String busyAction)
```
Note: busyAction is one of “Abandon” or “Force”
```
String getInitialConfig()
void setInitialConfig(String initialConfig)
String getTargetConfig()
void setTargetConfig(String targetConfig)
String getStartHour()
void setStartHour(String startHour)
String getStartMinute()
void setStartMinute(String startMinute)
String getEndHour()
void setEndHour(String endHour)
String getEndMinute()
void setEndMinute(String endMinute)
Integer getStartWarningPeriod()
void setStartWarningPeriod(Integer startWarningPeriod)
Integer getEndWarningPeriod()
void setEndWarningPeriod(Integer endWarningPeriod)
Integer getStartSuccessThreshold()
void setStartSuccessThreshold(Integer startSuccessThreshold)
String getDemandPriority()
void setDemandPriority(String demandPriority)
```
Note: demandPriority is only used for interval type “Demand”
```
DemandRuleInfo[] getRules()
void setRules(DemandRuleInfo[] rules)
```
Note: rules is only used for interval type “Demand”

DemandRuleInfo

The `DemandRuleInfo` object is used indirectly by the `policyIntervalAdmin` service; it is used directly by the `PolicyIntervalInfo` class. It is defined as follows:

```
String getType()
void setType(String type)
  Note: type is one of “queue”, “time” or “completionTime”
String getOp()
void setOp(String op)
  Note: op is one of “lt”, “lte”, “eq”, “gt” or “gte”
Long getThreshold()
void setThreshold(Long threshold)
String getAction()
  Note: action is one of “nothing”, “release”, “return”, “migrate” or “force”
void setAction(String action)
Long getResourceCount()
void setResourceCount(Long resourceCount)
```

ResourceInfo

The `ResourceInfo` object is used by the `resourceGroupAdmin` service. It is defined as follows:

```
String getResourceId()
void setResourceId(String resourceId)
String getHostname()
void setHostname(String hostname)
String getIp()
void setIp(String ip)
String getOs()
void setOs(String os)
String getCurrentResourceManager()
void setCurrentResourceManager(String currentResourceManager)
```

SharedResourceInfo

The `SharedResourceInfo` object is used by the `sharedResourceAdmin` service. It is defined as follows:

```
String getResourceId()
String getHostname()
String getIp()
String getOs()
String getStatus()
void setStatus(String status)
String getCurrentResourceManager()
String getPreviousResourceManager()
```

VmwareDemandableInfo

The `VmwareDemandableInfo` object is used by the `demandableResourceAdmin` service. It is defined as follows:

```
long getId()
String getResourceManager()
void setResourceManager(String resourceManager)
String getDatacenter()
void setDatacenter(String datacenter)
String getDatastore()
void setDatastore(String datastore)
String getTemplateFolderPath()
void setTemplateFolderPath(String path)
String getVmFolderPath()
void setVmFolderPath(String path)
String getComputeResource()
void setComputeResource(String computeResource)
String getResourcePool()
void setResourcePool(String resourcePool)
String getDefaultTemplate()
void setDefaultTemplate(String defaultTemplate)
String getVirtualMachineName()
void setVirtualMachineName(String virtualMachine)
String getGateway()
void setGateway(String gateway)
String getSubnet()
void setSubnet(String subnet)
String getDnsServer()
void setDnsServer(String dnsServer)
String getDomain()
void setDomain(String domain)
```

VmwareResourceManagerInfo

The `VmwareResourceManagerInfo` object is used by the `resourceManagerAdmin` service. It is defined as follows:

```
long getId()
String getName()
void setName(String name)
String getDescription()
void setDescription(String description)
int getInstanceLimit()
void setInstanceLimit(int instanceLimit)
String getVcenterUrl()
void setVcenterUrl(String url)
String getAdminUsername()
void setAdminUsername(String adminUsername)
String getAdminPassword()
void setAdminPassword(String adminPassword)
```

VmwareResourceGroupInfo

The `VmwareResourceGroupInfo` object is used by the `resourceGroupAdmin` service. It is defined as follows:

```
long getId()
String getName()
void setName(String name)
String getDescription()
void setDescription(String description)
String getDefaultManager()
void setDefaultManager(String defaultManager)
Integer getDefaultSuccessThreshold()
void setDefaultSuccessThreshold(Integer defaultSuccessThreshold)
```

Note: defaultSuccessThreshold must be between 0 and 100

```
Integer getDefaultWarningPeriod()
void setDefaultWarningPeriod(Integer defaultWarningPeriod)
Integer getResourceCount()
void setResourceCount(Integer resourceCount)
String getDatacenter()
void setDatacenter(String datacenter)
String getDatastore()
void setDatastore(String datastore)
String getTemplateFolderPath()
void setTemplateFolderPath(String path)
String getVmFolderPath()
void setVmFolderPath(String path)
String getComputeResource()
void setComputeResource(String computeResource)
String getResourcePool()
void setResourcePool(String resourcePool)
String getDefaultTemplate()
void setDefaultTemplate(String defaultTemplate)
String getVirtualMachineName()
void setVirtualMachineName(String virtualMachine)
String getGateway()
void setGateway(String gateway)
String getSubnet()
void setSubnet(String subnet)
String getDnsServer()
void setDnsServer(String dnsServer)
String getDomain()
void setDomain(String domain)
PolicyIntervalInfo
String getType()
void setType(String type)
```

Note: type is one of "Placement", "Demand", or "VMware Demand-based"

Ec2DemandableInfo

The `Ec2DemandableInfo` object is used by the `demandableResourceAdmin` service. It is defined as follows:

```
long getId()
String GetResourceManager()
```

```
void setResourceManager(String resourceManager)
Ec2LaunchConfigInfo GetLaunchConfig
void SetLaunchConfig(Ec2LaunchConfigInfo launchConfig)
```

Ec2LaunchConfigInfo

The `Ec2LaunchConfigInfo` object is used indirectly by the `resourceGroupAdmin` and `sharedResourceAdmin` services; it is used directly by the `Ec2ResourceManagerInfo` and `Ec2ResourceGroupInfo` classes. It is defined as follows:

```
String getAwsImage()
void setAwsImage(String awsImage)
String getInstanceType()
void setInstanceType(String instanceType)
String getSecurityGroup()
void setSecurityGroup(String securityGroup)
String getAvailabilityZone()
void setAvailabilityZone(String availabilityZone)
```

Note: null indicates the “automatic” option

```
String getKeyPairName()
void setKeyPairName(String keyPairName)
String getKeyFingerprint()
void setKeyFingerprint(String keyFingerprint)
String getKeyData()
void setKeyData(String keyData)
String getUserData()
void setUserData(String userData)
String getKernel()
void setKernel(String kernel)
String getRamDisk()
void setRamDisk(String ramDisk)
String getDeviceMapping()
void setDeviceMapping(String deviceMapping)
String getSubnetId()
void setSubnetId(String subnetId)
```

Note: if an EC2 Manager is using a VPC tunnel type then subnetId must be set to a valid VPC subnet or the instances will fail to launch.

```
String getSpotPrice()
void setSpotPrice(String spotPrice)
```

Note: Spot price is a textual representation of the AWS spot price bid for your instances. It must be a valid number or null.

Ec2ResourceManagerInfo

The `Ec2ResourceManagerInfo` object is used by the `resourceGroupAdmin` service. It is defined as follows:

```
long getId()
String getName()
void setName(String name)
String getDescription()
void setDescription(String description)
String getRegionUrl()
```

```
void setRegionUrl(String regionUrl)
```
Note: regionUrl is the endpoint URL for the region you want to use, e.g. https://us-east-1.ec2.amazonaws.com
```
String getAccessKeyId()
void setAccessKeyId(String accessKeyId)
String getSecretAccessKey()
void setSecretAccessKey(String secretAccessKey)
int getInstanceLimit()
void setInstanceLimit(int instanceLimit)
String getProxy()
void setProxy(String proxy)
```
Note: proxy is the Elastic IP address that will be used for the server-side proxy
```
boolean isEnabled()
void setEnabled(boolean enabled)
Ec2LaunchConfigInfo getLaunchConfig()
void setLaunchConfig(Ec2LaunchConfigInfo info)
String getTunnelType()
void setTunnelType(String tunnelType)
```
Note: tunnelType must be one of SSH, VPC, or None. If tunnelType is VPC then vpcId must be set with a valid vpcId indicating the VPC into which instances for this Manager should be launched.
```
String getVpcId()
void setVpcId(String vpcId)
```

Ec2ResourceGroupInfo

The `Ec2ResourceGroupInfo` object is used by the `sharedResourceAdmin` service. It is defined as follows:

```
long getId()
String getName()
void setName(String name)
String getDescription()
void setDescription(String description)
String getResourceManager()
void setResourceManager(String resourceManager)
Integer getDefaultSuccessThreshold()
void setDefaultSuccessThreshold(Integer defaultSuccessThreshold)
```
Note: defaultSuccessThreshold must be between 0 and 100
```
Integer getDefaultWarningPeriod()
void setDefaultWarningPeriod(Integer defaultWarningPeriod)
Integer getResourceCount()
void setResourceCount(Integer resourceCount)
Ec2LaunchConfigInfo getLaunchConfig()
void setLaunchConfig(Ec2LaunchConfigInfo launchConfig)
```

Appendix B Using EC2 Resources Without VPC

This appendix describes how to use the SSH tunnel method for EC2 tunneling.

Topics

Overview

Because EC2 resources aren't hosted on your corporate network, there must be a method for Engines in the Amazon cloud to securely communicate with Directors and Brokers in your corporate network. Federator supports the use of a Virtual Private Cloud (VPC), which enables you to host EC2 instances in Amazon's cloud as if they were local instances.

Previous Federator versions supported another method of tunneling, in which Federator starts a proxy machine in the cloud, then starts a corresponding server on the Federator machine inside your network. Then, the servers create a secure tunnel for communication, using SSH.

Because configuring VPC typically involves coordination with your IT staff, it may be suitable to use SSH tunneling for evaluation and testing. This legacy method is still supported for evaluation purposes, but the VPC method should be used for production environments.

This appendix describes how to use the SSH tunnel method for EC2 tunneling.

Configuring SSH Tunnel

If you are using the legacy method of an SSH Tunnel to connect Amazon's cloud to your corporate network, you must configure an Elastic IP address and download the proxy's key pair.

Creating an Elastic IP Address

Federator uses a proxy server to allow the Engine Daemons running in EC2 to communicate securely with the Directors and Brokers in your corporate network. The proxy server also caches resources being downloaded by Engines, making the process faster (and less costly, since you pay Amazon for the data transferred in and out of the cloud). In order for the proxy server to work correctly, it must have a fixed IP address that is visible outside the cloud, known as an "elastic" IP address. You can create one using the AWS Management Console or the command-line interface. Take note of the address since you'll need it when setting up your EC2 Resource Manager (see EC2 Resource Managers on page 33). You won't want to bind it to another instance or the proxy won't work correctly.

Amazon charges a small fee for IP addresses which you create but don't bind to an instance, after a certain number of free ones. Check the EC2 web site for details.

Downloading the Proxy's Key Pair

Federator automatically tells Amazon to generate an X.509 key pair to use for the SSH tunnel; the key pair's name is based on the elastic IP address chosen for the proxy. (You can use this key pair when setting up EC2 resource groups so that the EC2 instances running Engines use the same encryption key. See Creating EC2 Resource Groups on page 71 for details about configuring EC2 resource groups. See Installing Squid on page 102 for more about generating key pairs in Amazon.)

You can download the proxy's key data as a PEM file if you need to securely connect to the proxy server instance yourself.

To download the proxy's key data:

1. Click the **Amazon EC2** link under the **Resource Managers** tab.
2. Click the name of the Resource Manager whose key you want to download. It appears on the Amazon EC2 Resource Managers page.
3. Click **Download Proxy Key**. Your browser should start downloading the PEM file, which will be named `fed-[Proxy IP Address]-keypair.pem`. See the documentation that came with your SSH tools for details on using the PEM file to access the cloud-based proxy machine.

Installing Squid

If you are using Amazon EC2 resources with SSH tunneling, you will also need to install the Squid proxy. You can find more information on Squid at http://www.squid-cache.org/; if you are running Federator on a Unix machine, you can also use the package management system for your OS, such as APT or YUM.

For Windows, an ssh agent must also be installed, such as the `ssh.exe` that comes with Cygwin. Edit the *DS_HOME*`/conf/federator.properties` file and set `ssh.client.path` to the location of your ssh agent. For example:

```
ssh.client.path=C:/cygwin/bin
```

or

```
ssh.client.path=C:/cygwin/bin/ssh.exe
```

Note that plink, which was previously supported as an ssh agent, is no longer supported.

Running Squid

After you have installed Squid and have it running, you must configure Federator with its location. Go to **Admin > General** and change the value of **Local Proxy Hostname** to the name of the host that is running Squid.

On Windows machines, you can also configure Federator to start and stop Squid if it is installed on the same machine. Go to **Admin > General** and check **Enable Proxy**. This assumes the Squid proxy is installed in `c:\squid`. This should only be used for test purposes; it is recommended that you run Squid on another machine and separately control when it starts.

Upgrading GridServer When Using Squid

If you are using the Squid proxy for SSH tunneling, and you upgrade GridServer on a target Resource Manager, your Engines may not upgrade properly unless you clear the Squid cache.

To clear the cache:

1. Stop the policy and EC2 Resource Manager.
2. Stop the Squid proxy. If you are running Federator on Windows, you also need to stop Federator, because it auto-restarts Squid.
3. Delete the Squid cache directory.

4. Rebuild the cache folder structure with `squid -z`.
5. Start Federator if it is stopped, and start the EC2 Resource Manager and policy.