

Pivotal™ RabbitMQ®

Version 3.4

Getting Started with Pivotal RabbitMQ

Rev: 01

© 2014 Pivotal Software, Inc.

Notice

Copyright

Copyright © 2014 Pivotal Software, Inc. All rights reserved.

Pivotal Software, Inc. believes the information in this publication is accurate as of its publication date. The information is subject to change without notice. THE INFORMATION IN THIS PUBLICATION IS PROVIDED "AS IS." PIVOTAL SOFTWARE, INC. ("Pivotal") MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

Use, copying, and distribution of any Pivotal software described in this publication requires an applicable software license.

All trademarks used herein are the property of Pivotal or their respective owners.

Contents

- Preface: Contacting Pivotal.....iv**
 - Current Pivotal Customers..... v
 - Public Inquiries About Application Fabric Products.....vi

- Chapter 1: About Getting Started with Pivotal RabbitMQ..... 7**
 - Intended Audience.....8
 - About Pivotal RabbitMQ.....9
 - Main Components.....9
 - What You Can Do with Pivotal RabbitMQ..... 10
 - Installing Pivotal RabbitMQ..... 13
 - RHEL: Install Pivotal RabbitMQ from a RPM.....13
 - RHEL: Start and Stop the RabbitMQ Server..... 15
 - RHEL: Install Pivotal RabbitMQ Java Client Libraries from RPM..... 16
 - SLES: Install Pivotal RabbitMQ.....17
 - Ubuntu: Install Pivotal RabbitMQ from the Pivotal Debian Package Repository..... 18
 - Ubuntu: Start and Stop the RabbitMQ Server..... 20
 - Ubuntu: Install Pivotal RabbitMQ Java Client Libraries from the Pivotal Debian Repository...21
 - Windows: Install Pivotal RabbitMQ Server.....21
 - Windows: Start and Stop the RabbitMQ Server..... 23
 - Windows: Install Pivotal RabbitMQ Client Libraries.....24
 - Upgrade Pivotal RabbitMQ.....24
 - Troubleshooting Installation Problems.....25
 - Next Steps.....26
 - Perform General Server Configuration Tasks.....26
 - Implement SSL.....26
 - Set Up Distributed Messaging and High Availability.....27
 - Set Up and Work with Pivotal RabbitMQ Clients.....27

Contacting Pivotal

Current Pivotal Customers

Submit a ticket from the [Help & Support Page](#).

Public Inquiries About Application Fabric Products

Email the appropriate group for the Application Fabric product:

- RabbitMQ@goPivotal.com
- tcServer@goPivotal.com
- WebServer@goPivotal.com

Chapter 1

About Getting Started with Pivotal RabbitMQ

Getting Started with Pivotal RabbitMQ describes how to install Pivotal RabbitMQ cloud-ready messaging, a highly scalable messaging solution that is portable across many operating systems and developer platforms. This guide also provides a functional overview of the product.

Intended Audience

About Pivotal RabbitMQ

Installing Pivotal RabbitMQ

Next Steps

Intended Audience

Getting Started with Pivotal RabbitMQ is intended for anyone who needs to download and install Pivotal RabbitMQ server software and client libraries.

About Pivotal RabbitMQ

Pivotal RabbitMQ is a protocol-based messaging solution designed for cloud computing and modern, widely distributed web applications. It is the de facto standard for cloud messaging and the leading implementer of Advanced Message Queuing Protocol (AMQP), an open-standards alternative to costly, proprietary commercial messaging technologies.

Main Components

What You Can Do with Pivotal RabbitMQ

Main Components

Pivotal RabbitMQ consists of a server (including plug-ins) and client libraries that implement AMQP.

Pivotal RabbitMQ Server

Pivotal RabbitMQ Plug-ins

Pivotal RabbitMQ Client Libraries

Pivotal RabbitMQ Server

The RabbitMQ server is an implementation of an AMQP broker. It is written on the widely-used Erlang/Open Telecom Platform (OTP), an "always available" platform that has been in production use in carrier grade telcos since the late 1990s. The core server is unusually compact, with only 12,000 lines of code. Because RabbitMQ implements the open AMQP protocol, it is not necessary to know Erlang, and many users prefer to manage the broker using Java or Ruby. For details about Pivotal RabbitMQ server usage and functionality, see [Server Documentation](#). For AMQP information, see [Protocol Documentation](#).

Pivotal RabbitMQ Plug-ins

Pivotal RabbitMQ server is shipped with the following plug-ins. You enable, disable, and display them with the `rabbitmq-plugins` tool. See [Plugins](#).

- `amqp_client`
- `eldap`
- `erlando`
- `mochiweb`
- `rabbitmq_auth_backend_ldap`
- `rabbitmq_auth_mechanism_ssl`
- `rabbitmq_consistent_hash_exchange`
- `rabbitmq_federation`
- `rabbitmq_federation_management`
- `rabbitmq_jsonrpc`
- `rabbitmq_jsonrpc_channel`
- `rabbitmq_jsonrpc_channel_examples`
- `rabbitmq_management`
- `rabbitmq_management_agent`
- `rabbitmq_management_visualiser`
- `rabbitmq_mochiweb`
- `rabbitmq_shovel`

- `rabbitmq_shovel_management`
- `rabbitmq_stomp`
- `rabbitmq_tracing`
- `rabbitmq_web_stomp`
- `rabbitmq_mqtt`
- `rfc4627_jsonrpc`
- `webmachine`

Pivotal RabbitMQ Client Libraries

Pivotal RabbitMQ provides and commercially supports .NET and Java client libraries. The Erlang AMQP client is also available to all AMQP-compliant brokers, which includes RabbitMQ server.

The RabbitMQ .NET client is an implementation of an AMQP client library for C# (and, implicitly, other .NET languages), and a binding that exposes AMQP services through Microsoft's Windows Communication Foundation (WCF). The library is open-source, and is dual-licensed under the Apache License v2 and the Mozilla Public License v1.1. For details about .NET client usage and functionality, under *Client Documentation*, see ".NET Client Library User Guide" and ".NET Client API Documentation."

The RabbitMQ Java client library allows Java code to interface to AMQP servers. The library is platform neutral; the binary distributions differ only in the version of Java they are intended for use with. The library is open-source, and is dual-licensed under the Mozilla Public License v1.1 and the GNU General Public License, v2. For details about Java client usage and functionality, under *Client Documentation*, see *Java Client API Guide* and "Javadoc Documentation."

JMS Client for Pivotal RabbitMQ is an add-on library for this version of Pivotal RabbitMQ. It allows applications written for the JMS 1.1 API to use Pivotal RabbitMQ messaging. The JMS Client for Pivotal RabbitMQ implements the JMS API on top of the RabbitMQ Java client API, so that new and existing JMS applications can connect with RabbitMQ brokers.

What You Can Do with Pivotal RabbitMQ

The following sections highlight typical use cases and describe how Pivotal RabbitMQ is optimized for modern messaging.

Main Features

How Pivotal RabbitMQ Is Engineered for Modern Messaging

Protocols Supported by Pivotal RabbitMQ

Main Features

Pivotal RabbitMQ lets you implement a range of *message patterns* across many platforms and protocols.

- *Reliability* . Pivotal RabbitMQ offers a variety of features to let you trade off performance with reliability, including persistence, delivery acknowledgements, publisher confirms, and high availability.
- *Flexible routing* . Messages are routed through exchanges before arriving at queues. Pivotal RabbitMQ features several built-in exchange types for typical routing logic. For more complex routing you can bind exchanges together or even write your own exchange type as a plugin.
- *Clustering* . Several Pivotal RabbitMQ servers on a local network can be clustered to form a single logical broker.
- *Federation* . Federation offers an alternative distributed messaging model for servers that need to be more loosely and unreliably connected than clustering allows.
- *Highly Available Queues* . You can mirror queues across several machines in a cluster, ensuring that even in the event of hardware failure your messages are safe.
- *Multi-protocol* . Pivotal RabbitMQ supports messaging over a variety of messaging protocols.

- *Many Clients* . There are Pivotal RabbitMQ clients for almost any language you can think of.
- *Management UI* . Pivotal RabbitMQ ships with an easy-to use management user interface that allows you to monitor and control every aspect of your message broker.
- *Tracing* . If your messaging system is not working properly, you can use the tracing support to find out what's going wrong.
- *Plugin system* . Pivotal RabbitMQ ships with a variety of plugins, extending it in different ways, and you can also write your own.
- Commercial support. Commercial support, training, and consulting are available from Pivotal.
- *Large Community* . There is a large community around open-source RabbitMQ, producing many clients, plugins, guides, and so on. Join the mailing list to get involved.

How Pivotal RabbitMQ Is Engineered for Modern Messaging

Messaging is critical to widely distributed, cloud-based applications that leverage pools of shared infrastructure. Modern web applications use messaging both within their own application components and to connect to other services.

This approach requires messaging software that is lightweight enough to embed throughout an application; easily adoptable by many developers across many teams; and flexible enough to integrate heterogeneous application components and systems. Pivotal RabbitMQ leverages the advantages and requirements of modern messaging by implementing:

- A small disk and memory footprint.
- An easily understood API with drivers available for multiple languages.
- Support on all major operating systems and developer platforms.
- A protocol-based approach that let you integrate clients built in Java, .NET, Python, PHP, Ruby, and other languages, or which are implemented as web services.
- AMQP, the industry standard protocol for business messaging.

About AMQP

AMQP has widespread industry backing, especially among financial services and large-scale web application providers. It is a complete solution for all categories of messaging, from publish/subscribe and content-based routing to reliable and transactional transfer of large data sets.

Unlike JMS, AMQP is a binary wire protocol that enables testable interoperability across any language and platform. This capability translates into lower integration costs because, like TCP or HTTP, messaging becomes a simple matter of wiring. AMQP is a modern design, more like a switching exchange than like a database, and is not affected by technology decisions made in the 1990s.

To learn how Pivotal RabbitMQ implements and extends AMQP, see *Compatibility and Conformance* and *Protocol Extensions*.

Protocols Supported by Pivotal RabbitMQ

Pivotal RabbitMQ supports the following messaging protocols, either directly or through the use of plugins:

- **AMQP 0-9-1, 0-9 and 0-8, and extensions:** RabbitMQ was originally developed to support AMQP and is thus the "core" protocol supported by the broker. All of these variants are fairly similar to each other, with later versions tidying up unclear or unhelpful parts of earlier versions. RabbitMQ has extended AMQP 0-9-1 in various ways.

AMQP 0-9-1 is a binary protocol, and defines quite strong messaging semantics. For clients it's a reasonably easy protocol to implement, and there are a large number of implementations available for many different programming languages and environments.

It is recommended that you use AMQP 0-9-1.

- **STOMP:** STOMP is a text-based messaging protocol emphasising (protocol) simplicity. It defines little in the way of messaging semantics, but is easy to implement and very easy to implement partially (it's the only protocol that can be used by hand over telnet).

RabbitMQ supports STOMP (all current versions) via a plugin.

- **MQTT:** MQTT is a binary protocol emphasising lightweight publish/subscribe messaging, targeted to clients in constrained devices. It has well defined messaging semantics for publish/subscribe, but not for other messaging idioms.

RabbitMQ supports MQTT 3.1 via a plugin.

- **AMQP 1.0:** Despite the name, AMQP 1.0 is a radically different protocol from AMQP 0-9-1 / 0-9 / 0-8, sharing essentially nothing at the wire level. AMQP 1.0 imposes far fewer semantic requirements; it is therefore easier to add support for AMQP 1.0 to existing brokers. The protocol is substantially more complex than AMQP 0-9-1, and there are fewer client implementations.

RabbitMQ supports AMQP 1.0 via a plugin.

- **HTTP:** HTTP is not a messaging protocol. However, RabbitMQ can transmit messages over HTTP in three ways:
 - The management plugin supports a simple HTTP API to send and receive messages. This is primarily intended for diagnostic purposes but can be used for low volume messaging without reliable delivery.
 - The Web-STOMP plugin supports STOMP messaging to the browser, using WebSockets or one of the fallback mechanisms supported by SockJS.
 - The JSON-RPC channel plugin supports AMQP 0-9-1 messaging over JSON-RPC to the browser. Note that because JSON RPC is a synchronous protocol, some parts of AMQP that depend on asynchronous delivery to the client are emulated by polling.

Installing Pivotal RabbitMQ

Installation procedures for Pivotal RabbitMQ server differ according to the operating system you are using.

RHEL: Install Pivotal RabbitMQ from a RPM

RHEL: Start and Stop the RabbitMQ Server

RHEL: Install Pivotal RabbitMQ Java Client Libraries from RPM

SLES: Install Pivotal RabbitMQ

Ubuntu: Install Pivotal RabbitMQ from the Pivotal Debian Package Repository

Ubuntu: Start and Stop the RabbitMQ Server

Ubuntu: Install Pivotal RabbitMQ Java Client Libraries from the Pivotal Debian Repository

Windows: Install Pivotal RabbitMQ Server

Windows: Start and Stop the RabbitMQ Server

Windows: Install Pivotal RabbitMQ Client Libraries

Upgrade Pivotal RabbitMQ

Troubleshooting Installation Problems

RHEL: Install Pivotal RabbitMQ from a RPM

You can install Pivotal RabbitMQ on Red Hat Linux Enterprise (RHEL) computers using the Pivotal RPM repository. See *Install Pivotal RabbitMQ from a Downloaded RPM*.

You can also download the RPM from the *Pivotal RabbitMQ product page* and install it on your RHEL computer using the `rpm` command, as described in *Install Pivotal RabbitMQ from a Downloaded RPM*.

Install Pivotal RabbitMQ from the RPM Repository

Install Pivotal RabbitMQ from a Downloaded RPM

Install Pivotal RabbitMQ from the RPM Repository

Prerequisites

- Install Erlang.
- Install the Pivotal repository RPM, which makes it easy to browse for the RPMs. Install the repository on each RHEL computer on which you want to install Pivotal RabbitMQ.

1. On the RHEL computer, start a terminal either as the `root` user or as an unprivileged user who has `sudo` privileges.
2. Install the Pivotal repository RPM using the following `wget` command:

RHEL 5:

```
prompt# wget -q -O - http://packages.gopivotal.com/pub/rpm/rhel5/app-suite/app-suite-installer | sh
```

RHEL 6:

```
prompt# wget -q -O - http://packages.gopivotal.com/pub/rpm/rhel6/app-suite/app-suite-installer | sh
```

The command performs the following tasks:

- Imports the Pivotal GNU Privacy Guard (GPG) key.
 - Installs the Pivotal repository RPM.
 - Launches the Pivotal End User License Agreement (EULA) acceptance and responsibility configuration script.
 - Outputs the EULA for you to read; you must answer `yes` to accept the terms and continue.
3. Enter the command `yum search pivotal` to view the list of Pivotal components that you can install from the repository. For example (output truncated for clarity):

```
prompt# yum search pivotal
===== Matched: pivotal =====
=====
...
pivotal-rabbitmq-java-client-bin.noarch : The RabbitMQ Java Client Library
pivotal-rabbitmq-server.x86_64 : The RabbitMQ server

pivotal-web-server.x86_64 : Pivotal Web Server
...
```

Procedure

1. Log in to the RHEL computer on which you will install Pivotal RabbitMQ as the `root` user (or as an unprivileged user who has `sudo` privileges).
2. Execute the following `yum` command:

```
prompt# yum install pivotal-rabbitmq-server
```

3. Respond to the prompts to complete RabbitMQ installation.

What the yum command does

The `yum` command:

- Installs Pivotal RabbitMQ into the `/opt/pivotal/rabbitmq` directory and sets the owner of the directory, along with all child directories and files, to `root:root`.
- If the user does not already exist, adds a `rabbitmq` non-interactive user (in the group `rabbitmq`). The RabbitMQ server process runs as the `rabbitmq` user.

You cannot log in directly as the `rabbitmq` user because interactive login is disabled.

What to do next

- *Start and Stop the Pivotal RabbitMQ Server* to ensure that you installed it correctly.
- Ensure that the RabbitMQ management plugin is enabled by running the following command:

```
prompt# cd /opt/pivotal/rabbitmq/bin
prompt# ./rabbitmq-plugins list
```

Check that there is an `[E]` next to `rabbitmq_management`, which indicates the plugin is enabled, as shown:

```
...
[E] rabbitmq_management          3.4. [x]
[e] rabbitmq_management_agent   3.4. [x]
[E] rabbitmq_management_visualiser 3.4. [x]
...
```

If the management plugin is not enabled, enable it with the following command:

```
prompt# ./rabbitmq-plugins enable rabbitmq_management
```

Restart the RabbitMQ server for the changes to take effect.

- To ensure RabbitMQ Server is running, launch the management console in your browser using the following URL:

```
http://host:15672
```

If your browser is on the same computer as the RabbitMQ server you can use `http://localhost:15672`.

Enter the default credentials that were created at installation: user `guest` and password `guest`.

Use the management console to manage exchanges, queues, bindings, virtual hosts, users and permissions.

- *Install the Pivotal RabbitMQ Java Client Libraries.*
- Read the *Next Steps* section for a high-level roadmap to subsequent tasks.

Install Pivotal RabbitMQ from a Downloaded RPM

You can install Pivotal RabbitMQ on RHEL by downloading the RPM from the Pivotal download center and executing the `rpm` command.

Prerequisites

- Install Erlang.

Procedure

1. Log in to the RHEL computer on which you will install Pivotal RabbitMQ as the `root` user (or as an unprivileged user who has `sudo` privileges).
2. From the *Pivotal RabbitMQ product page*, click **Downloads**.
3. Download the RabbitMQ RPM file, and save it to a directory on your computer. The RPM file is called `pivotal-web-server-version.noarch.rpm`.

Choose the architecture (32- or 64-bit) appropriate for your computer.

4. Start a terminal and change to the directory in which you downloaded the RPM.
5. Execute the following `rpm` command to install RabbitMQ:

```
prompt# rpm -ivhf pivotal-rabbitmq-server-version.noarch.rpm
```

6. In the previous section, see *What the yum install command does* for post-installation information, such as the installation directory. (The `yum install` command corresponds to the `rpm` command in this procedure.)

See *What to do next* for the suggested next steps.

RHEL: Start and Stop the RabbitMQ Server

You must first start the RabbitMQ server to use it. The default configuration for the server is usually adequate to start it.

Prerequisites

- *Install Pivotal RabbitMQ Server.*

Procedure

1. Log in as the `root` user and open a terminal window.
2. Start the RabbitMQ server using the `/sbin/service rabbitmq-server` command, passing it the `start` option. For example:

```
prompt# /sbin/service rabbitmq-server start
```

The RabbitMQ server process runs as the `rabbitmq` user, which is non-interactive.

3. To stop the server:

```
prompt# /sbin/service rabbitmq-server stop
```

4. To get status about the server (only partial output shown):

```
prompt# /sbin/service rabbitmq-server status
Status of node rabbit@localhost ...
[{pid,6184},
 {running_applications,
  [{rabbitmq_management,"RabbitMQ Management Console","3.4.[x]"},
   {rabbitmq_web_dispatch,"RabbitMQ Web Dispatcher","3.4.[x]"},
   {webmachine,"webmachine","1.10.3-rmq3.4.[x]-gite9359c7"},
   {mochiweb,"MochiMedia Web Server","2.7.0-rmq3.4.[x]-git680dba8"},
   {rabbitmq_management_agent,"RabbitMQ Management Agent","3.4.[x]"},
   {rabbit,"RabbitMQ","3.4.[x]"},
   {os_mon,"CPO CXC 138 46","2.2.10"},
   {inetets,"INETETS CXC 138 49","5.9.1"},
   {xmerl,"XML parser","1.3.2"},
   {mnesia,"MNESIA CXC 138 12","4.7.1"},
   {amqp_client,"RabbitMQ AMQP Client","3.4.[x]"},
   {sasl,"SASL CXC 138 11","2.2.1"},
   {stdlib,"ERTS CXC 138 10","1.18.2"},
   {kernel,"ERTS CXC 138 10","2.15.2"}]},
 {os,{unix,linux}},
 {erlang_version,
  "Erlang R15B02 (erts-5.9.2) [source] [64-bit] [async-threads:30] [kernel-
poll:true]\n"},
 {memory,
  [{total,32983488},
   {connection_procs,5296}],
 ...
```

What to do next

- The RabbitMQ server does not start as a daemon by default when your system starts. If, however, you want it to start by default, then run the following command as the `root` user or a user with `sudo` privileges:

```
prompt# chkconfig rabbitmq-server on
```

- You can also use the `rabbitmqctl` command to stop and start the RabbitMQ server, as well as perform a number of other tasks, such as get the status of the server, rotate the logs, and so on. The `rabbitmqctl` command is located in the `/opt/pivotal/rabbitmq/bin` directory. Run the command as the `root` user or as a user with `sudo` privileges. For example, to get the status of the RabbitMQ server (full output removed for simplicity):

```
prompt# cd /opt/pivotal/rabbitmq/bin
prompt# ./rabbitmqctl status
Status of node 'rabbit@rhel-5-64-svr' ...
```

For complete information about this command, see the `rabbitmqctl` manual page.

RHEL: Install Pivotal RabbitMQ Java Client Libraries from RPM

Pivotal RabbitMQ provides client libraries for Java. The Java client library allows Java code to interface to AMQP servers, such as Pivotal RabbitMQ Server. The library is platform-neutral. The Java client library is distributed as an RPM for RHEL computers.

Procedure

1. Log in to the RHEL computer on which you will install Pivotal RabbitMQ as the `root` user (or as an unprivileged user who has `sudo` privileges).
2. From the *Pivotal RabbitMQ product page*, click **Downloads**.
3. Download the RabbitMQ Java Library RPM file, and save it to a directory on your computer. The RPM file is called `pivotal-rabbitmq-java-client-bin-version.noarch.rpm`.
4. Start a terminal and change to the directory in which you downloaded the RPM.

- Execute the following `rpm` command to install the RabbitMQ Java client libraries:

```
prompt# rpm -ivhf pivotal-rabbitmq-java-client-bin-version.noarch.rpm
```

What to do next

- Read the client documentation for programming information and code samples.

SLES: Install Pivotal RabbitMQ

You can install Pivotal RabbitMQ on SUSE Linux Enterprise Server (SLES) computers from a tar.gz file.

Install Pivotal RabbitMQ from a tar.gz file

Install Pivotal RabbitMQ from a tar.gz file

Prerequisites

- Install Erlang.

Procedure

- Install the server.

- Download the `pivotal-rabbitmq-server-3.4.x.tar.gz` file.
- Contained in the tarball is a directory named `rabbitmq_server-3.4.[x]`. Extract this directory into an appropriate directory for application binaries on your system. The `sbin` folder is located in this directory.

- Start the server.

- Invoke the `sbin/rabbitmq-server` shell script. The script displays a short banner message, concluding with the message "completed with [n] plugins.", indicating that the RabbitMQ broker has been started successfully.

You can also start the server in "detached" mode with the following command:`rabbitmq-server -detached`In this case, the server process runs in the background.

- Configure the server.

Customise the RabbitMQ environment by setting environment variables in the `$RABBITMQ_HOME/etc/rabbitmq/rabbitmq-env.conf` file.

Configure server components in the RabbitMQ configuration file located at `$RABBITMQ_HOME/etc/rabbitmq/rabbitmq.config`.

Neither of these files exists after installation.

File locations

The generic Unix tarball is designed as far as possible to run without requiring configuration changes or special permissions. The directories and files used by default are all held under the installation directory `rabbitmq_server-3.4.[x]`, which is in the `$RABBITMQ_HOME` variable in the scripts. Should you want to install RabbitMQ such that it uses the conventional system directories for configuration, database, log files, plugins etc, it is possible to do this. Find the line:`SYS_PREFIX=${RABBITMQ_HOME}`in the `sbin/rabbitmq-defaults` script and change this line to:`SYS_PREFIX=`but do not modify any other line in this script.

Note:

After this modification the default directory locations may require different permissions. In particular, `RABBITMQ_MNESIA_BASE` and `RABBITMQ_LOG_BASE` may need to be created. (The server will attempt to create the permissions at startup.) The `RABBITMQ_ENABLED_PLUGINS_FILE` will need to be

writable (for `rabbitmq-plugins`). The configuration files are accessed in the `/etc/rabbitmq/` directory.

What to do next

- *Start and Stop the Pivotal RabbitMQ Server* to ensure that you installed it correctly.
- Ensure that the RabbitMQ management plugin is enabled by running the following command:

```
prompt# cd <rabbitmq_dir>/bin
prompt# ./rabbitmq-plugins list
```

Check that there is an `[E]` next to `rabbitmq_management`, which indicates the plugin is enabled, as shown:

```
...
[E] rabbitmq_management          3.4. [x]
[e] rabbitmq_management_agent   3.4. [x]
[E] rabbitmq_management_visualiser 3.4. [x]
...
```

If the management plugin is not enabled, enable it with the following command:

```
prompt# ./rabbitmq-plugins enable rabbitmq_management
```

Restart the RabbitMQ server for the changes to take effect.

- To ensure RabbitMQ Server is running, launch the management console in your browser using the following URL:

```
http://host:15672
```

If your browser is on the same computer as the RabbitMQ server you can use `http://localhost:15672`.

Enter the default credentials that were created at installation: user `guest` and password `guest`.

Use the management console to manage exchanges, queues, bindings, virtual hosts, users and permissions.

- *Install the Pivotal RabbitMQ Java Client Libraries*.
- Read the *Next Steps* section for a high-level roadmap to subsequent tasks.

Ubuntu: Install Pivotal RabbitMQ from the Pivotal Debian Package Repository

It is recommended that you install Pivotal RabbitMQ on a computer running Ubuntu by first installing the Pivotal Debian package repository and then using `apt-get` to perform the RabbitMQ installation.

Prerequisites

- Install Erlang.
- Install the Pivotal Debian package repository that contains the Pivotal RabbitMQ `*.deb` packages.
 1. On the Ubuntu computer, start a terminal either as the `root` user or as an unprivileged user who has `sudo` privileges.
 2. Install the Pivotal Debian package repository using the following `wget` command:

Important: Run the entire `wget` command on a single line. Be sure you include the `| sh` at the end, or the installation fails.

```
prompt# wget -q -O - http://packages.gopivotal.com/ | sh
```

If necessary, use `sudo` to run the preceding commands if you are not logged in as the `root` user. Use the following `sudo` command:

```
prompt$ wget -q -O - packages.pivotal.io/ | sudo sh
```

The command performs the following tasks:

- Imports the Pivotal GNU Privacy Guard (GPG) key.
 - Installs the Pivotal Debian package repository.
 - Launches the Pivotal End User License Agreement (EULA) acceptance and repository configuration script.
 - Outputs the EULA for you to read; you must answer `yes` to accept the terms and continue.
3. Use the `apt-cache search pivotal` command to view the list of Pivotal components that you can install from the Pivotal Debian package repository. For example:

```
prompt# apt-cache search pivotal
rabbitmq-server - Pivotal RabbitMQ Server
pivotal-rabbitmq-java-client-bin - The RabbitMQ Java Client Library
pivotal-repo-precise - Pivotal EULA and APT repo configuration utility

pivotal-web-server - Pivotal Web Server
pivotal-web-server-devel - Pivotal Web Server Libraries and Headers
...
```

Procedure

1. From the Ubuntu computer on which you will install Pivotal RabbitMQ, log in as the `root` user (or as an unprivileged user who has `sudo` privileges) and start a terminal.
2. Execute the following `apt-get` command:

```
prompt# apt-get install rabbitmq-server
```

The `apt-get` command begins the install process, resolves dependencies, and displays the packages it will install.

If necessary, use `sudo` to run the preceding command if you are not logged in as the `root` user:

```
prompt$ sudo apt-get install rabbitmq-server
```

What the `apt-get install` command does

The `apt-get install` command:

- Installs Pivotal RabbitMQ into the `/opt/pivotal/rabbitmq` directory and sets the owner of the directory, along with all child directories and files, to `root:root`.
- If the user does not already exist, adds a `rabbitmq` non-interactive user (in the group `rabbitmq`). The RabbitMQ server process runs as the `rabbitmq` user.

You cannot log in directly as the `rabbitmq` user because interactive login is disabled.

What to do next

- *Start and Stop the Pivotal RabbitMQ Server* to ensure that you installed it correctly.
- Ensure that the RabbitMQ management plugin is enabled by running the following command:

```
prompt# cd /opt/pivotal/rabbitmq/bin
prompt# ./rabbitmq-plugins list
```

Check that there is an `[E]` next to `rabbitmq_management`, which indicates the plugin is enabled, as shown:

```
...
[E] rabbitmq_management          3.4. [x]
[e] rabbitmq_management_agent   3.4. [x]
[E] rabbitmq_management_visualiser 3.4. [x]
...
```

If the management plugin is not enabled, enable it with the following command:

```
prompt# ./rabbitmq-plugins enable rabbitmq_management
```

Restart the RabbitMQ server for the changes to take effect:

```
prompt# invoke-rc.d rabbitmq-server restart
```

- To ensure RabbitMQ Server is running, launch the management console in your browser using the following URL:

```
http://host:15672
```

If your browser is on the same computer as the RabbitMQ server you can use `http://localhost:15672`.

Enter the default credentials that were created at installation: user `guest` and password `guest`.

Use the management console to manage exchanges, queues, bindings, virtual hosts, users and permissions.

- *Install the Pivotal RabbitMQ Java Client Libraries.*
- Read the *Next Steps* section for a high-level roadmap to subsequent tasks.

Ubuntu: Start and Stop the RabbitMQ Server

The RabbitMQ server is started as a daemon by default when you first installed the RabbitMQ server package. This section describes how to manually start and stop it.

Prerequisites

- *Install Pivotal RabbitMQ Server.*

Procedure

1. Log in as the `root` user (or as an unprivileged user who has `sudo` privileges) and open a terminal window.
2. Start the RabbitMQ server using the `invoke-rc.d rabbitmq-server` command, passing it the `start` option. For example:

```
prompt# invoke-rc.d rabbitmq-server start
```

The RabbitMQ server process runs as the `rabbitmq` user, which is non-interactive.

3. To stop the server:

```
prompt# invoke-rc.d rabbitmq-server stop
```

4. To get status about the server (only partial output shown):

```
prompt# invoke-rc.d rabbitmq-server status
Status of node 'rabbit@vmc-ssrc-ubl056' ...
[{{pid,20045},
 {running_applications,
  [{rabbitmq_management,"RabbitMQ Management Console","3.4.[x]"},
   {rabbitmq_web_dispatch,"RabbitMQ Web Dispatcher","3.4.[x]"},
   {webmachine,"webmachine","1.10.3-rmq3.4.[x]-gite9359c7"},
   {mochiweb,"MochiMedia Web Server","2.7.0-rmq3.4.[x]-git680dba8"},
   {rabbitmq_management_agent,"RabbitMQ Management Agent","3.4.[x]"},
   {rabbit,"RabbitMQ","3.4.[x]"},
   {os_mon,"CPO CXC 138 46","2.2.7"},
   {inets,"INETS CXC 138 49","5.7.1"},
   {xmerl,"XML parser","1.2.10"},
   {mnesia,"MNESIA CXC 138 12","4.5"},
   {amqp_client,"RabbitMQ AMQP Client","3.4.[x]"},
   {sasldb,"SASL CXC 138 11","2.1.10"},
   {stdlib,"ERTS CXC 138 10","1.17.5"},
   {kernel,"ERTS CXC 138 10","2.14.5"}]},
 {os,{unix,linux}},
 {erlang_version,
  "Erlang R14B04 (erts-5.8.5) [source] [64-bit] [rq:1] [async-threads:30] [kernel-
poll:true]\n"},
 ...
```

What to do next

- You can use the `rabbitmqctl` command to stop and start the RabbitMQ server, as well as perform a number of other tasks, such as get the status of the server, rotate the logs, and so on. The `rabbitmqctl` command is located in the `/opt/pivotal/rabbitmq/bin` directory. Run the command as the `root` user or as a user with `sudo` privileges. For example, to get the status of the RabbitMQ server (full output removed for simplicity):

```
prompt# cd /opt/pivotal/rabbitmq/bin
prompt# ./rabbitmqctl status
Status of node 'rabbit@rhel-5-64-svr' ...
```

For complete information about this command, see the [rabbitmqctl manual page](#).

Ubuntu: Install Pivotal RabbitMQ Java Client Libraries from the Pivotal Debian Repository

Pivotal RabbitMQ provides client libraries for Java. The Java client library allows Java code to interface to AMQP servers, such as Pivotal RabbitMQ Server. The library is platform-neutral. The Java client library is distributed as Debian package for Ubuntu computers.

Prerequisites

If you have not already done so, install the Pivotal Debian package repository from which you will install the Java client library. See [Install the Pivotal Debian Package Repository](#).

Procedure

- From the Ubuntu computer on which you will install the Java libraries, log in as the `root` user (or as a user who has `sudo` privileges) and start a terminal.
- Execute the following `apt-get` command:

```
prompt# apt-get install pivotal-rabbitmq-java-client-bin
```

The `apt-get` command begins the install process, resolves dependencies, and displays the packages it will install.

If necessary, use `sudo` to run the preceding command if you are not logged in as the `root` user:

```
prompt$ sudo apt-get install pivotal-rabbitmq-java-client-bin
```

The `apt-get install` command installs the Pivotal RabbitMQ Java libraries into the `/opt/pivotal/rabbitmq-java-client` directory and sets the owner of all the sub-directories and files to `root:root`.

What to do next

- Read the [Client Documentation](#) for programming information and code samples.

Windows: Install Pivotal RabbitMQ Server

Prerequisites

- Install Erlang.

Procedure

- Log in to your Windows computer as the user who will install Pivotal RabbitMQ server.

Although not required, Pivotal recommends that you install and run Pivotal RabbitMQ as the Administrator user. If you decide not to do this, read "Windows Quirks" for additional information.

- From the [Pivotal RabbitMQ product page](#), click **Downloads**.

3. Download the Pivotal RabbitMQ Server for Windows *.exe installer, and save it to your Windows computer.
4. Launch the Windows installer.

The installer guides you through the installation process. It is recommended that you accept the default installation options, which include installing RabbitMQ server as a Windows service.

If the installer finds a previous version of Pivotal RabbitMQ, it prompts you to uninstall it. You must remove the old version if you want to install a newer one. When the uninstall of the old version of RabbitMQ is complete, relaunch the installer to install the new version.

Warning: Do not install Pivotal RabbitMQ in a Windows path that includes non-ASCII characters. If you do, the server fails to start. This is due to an Erlang limitation.

The installer program indicates whether the installation was successful.

By default, the server is installed in the `C:\Program Files\rabbitmq` directory, `C:\Program Files (x86)\rabbitmq` on 64-bit computers. The installation process also adds a `RabbitMQ Server` entry to the Windows Start menu that you can use to control (start, stop) the server and easily view log files, and so on.

5. Verify that the RabbitMQ service is installed by starting the Windows Services window and searching for `RabbitMQ`. You can start the Services window by typing `services.msc` from a Windows command prompt.

The Services window should show RabbitMQ in a list of services, with `Started` in the Status column.

6. To check the status of RabbitMQ server, start a Windows command prompt by selecting `All Programs > RabbitMQ Server > Command Prompt (sbin dir)` from the Windows Start menu and type the following command:

```
prompt> rabbitmqctl status
```

You should see something like the following output (truncated for clarity):

```
Status of node rabbit@JSHACKELL7 ...
[{pid,3708},
 {running_applications,
  [{rabbitmq_management,"RabbitMQ Management Console","3.4.[x]"},
   {rabbitmq_web_dispatch,"RabbitMQ Web Dispatcher","3.4.[x]"},
   {webmachine,"webmachine","1.10.3-rmq3.4.[x]-gite9359c7"},
   {mochiweb,"MochiMedia Web Server","2.7.0-rmq3.4.[x]-git680dba8"},
   {rabbitmq_management_agent,"RabbitMQ Management Agent","3.4.[x]"},
   {rabbit,"RabbitMQ","3.4.[x]"},
   {os_mon,"CPO CXC 138 46","2.2.11"},
   {inet,"INETC CXC 138 49","5.9.4"},
   {mnesia,"MNESIA CXC 138 12","4.8"},
   {amqp_client,"RabbitMQ AMQP Client","3.4.[x]"},
   {xmerl,"XML parser","1.3.3"},
   {sasl,"SASL CXC 138 11","2.3.1"},
   {stdlib,"ERTS CXC 138 10","1.19.1"},
   {kernel,"ERTS CXC 138 10","2.16.1"}]},
 {os,{win32,nt}},
 {erlang_version,"Erlang R16B (erts-5.10.1) [smp:4:4] [async-threads:30]\n"},
 {memory,
  [{total,18588080},
   {connection_procs,2792},
   {queue_procs,2792},
  ...
```

What to do next

- See *Windows: Start and Stop the RabbitMQ Server* for additional information about controlling the RabbitMQ server.
- Ensure that the RabbitMQ management plugin is enabled by selecting `All Programs > RabbitMQ Server > Command Prompt (sbin dir)` from the Windows Start menu and typing the following command:

```
prompt> rabbitmq-plugins list
```

Check that there is an `[E]` next to `rabbitmq_management`, which indicates the plugin is enabled, as shown:

```
...
[E] rabbitmq_management          3.4. [x]
[e] rabbitmq_management_agent   3.4. [x]
[E] rabbitmq_management_visualiser 3.4. [x]
...
```

If the management plugin is not enabled, enable it with the following command:

```
prompt> rabbitmq-plugins enable rabbitmq_management
```

Restart the RabbitMQ server for the changes to take effect.

- To ensure that RabbitMQ Server is running, launch the management console in your browser using the following URL:

```
http://host:15672
```

If your browser is on the same computer as the RabbitMQ server you can use `http://localhost:15672`.

Enter the default credentials that were created at installation: user `guest` and password `guest`.

Use the management console to manage exchanges, queues, bindings, virtual hosts, users and permissions.

- *Install the Pivotal RabbitMQ client libraries.*
- Read the *Next Steps* section for a high-level roadmap to subsequent tasks.

Windows: Start and Stop the RabbitMQ Server

You must first start the RabbitMQ server to use it. The default configuration for the server is usually adequate to start it.

The Windows Pivotal RabbitMQ installation process installs the server as a Windows service and configures it so that it starts automatically when you start your Windows computer. The install process also added a `RabbitMQ Server` entry to the Windows Start menu that you use to control (start, stop) the server, easily view the log files, and so on. The installation process also starts a server by default.

Prerequisites

- *Install Pivotal RabbitMQ Server.*

Procedure

1. Log in to your Windows machine as the user who installed Pivotal RabbitMQ server, such as the Administrator.
2. From the Windows Start menu, select `All Programs > RabbitMQ Server > Start Service` to start the RabbitMQ server.

The service runs in the security context of the system account without requiring a user to be logged in on a console.

Use the same process for stopping, reinstalling, and removing the service.

What to do next

- You can also use the `rabbitmqctl` command to stop and start the RabbitMQ server, as well as perform a number of other tasks, such as get the status of the server, rotate the logs, and so on.

Run the `rabbitmqctl` commands from a command prompt that you start from `All Programs > RabbitMQ Server > Command Prompt (sbin dir)`, which automatically opens in the correct directory.

For example, to get the status of the RabbitMQ server (full output removed for simplicity):

```
prompt> rabbitmqctl.bat status
Status of node 'rabbit@host-E4310' ...
```

See the [rabbitmqctl manual page](#) for complete information about this command.

- For information on other RabbitMQ commands such as `rabbitmq-plugins` and `rabbitmq-service`, see the manual pages.

Windows: Install Pivotal RabbitMQ Client Libraries

Pivotal RabbitMQ provides client libraries for Java and .NET languages such as C# for Windows platforms.

The Java client library allows Java code to interface to AMQP servers, such as Pivotal RabbitMQ server. The library is platform neutral.

The RabbitMQ .NET client is an implementation of an AMQP client library for C# and other .NET languages. It is also a binding that exposes AMQP services via Microsoft's Windows Communication Foundation (WCF).

The Pivotal RabbitMQ .NET client libraries include compilations for the following versions of .NET: 2.0 and 3.0.

Prerequisites

- To compile or run the Pivotal RabbitMQ .NET/C# client library, you need an appropriate .NET SDK or runtime environment available, respectively.

Procedure

1. From the [Pivotal RabbitMQ product page](#), click **Downloads**.
2. Download the Pivotal RabbitMQ client library * .exe installer, and save it to your Windows computer. You can download Java or .NET libraries.
3. Launch the Windows installer.

The installer guides you through the installation process. It is recommended that you accept the default installation options.

The installer program displays whether the installation was successful.

By default, the client libraries are installed in the following directories, although these might differ in your environment if you did not take the default values during installation:

- **.NET:** `C:\Program Files\RabbitMQ Clients\dotnet-client-version`
- **Java:** `C:\Program Files\RabbitMQ Clients\java-client-version`

On 64-bit computers, the top-level directory is `C:\Program Files (x86)`.

What to do next

- Read the [Client Documentation](#) for programming information and code samples.

Upgrade Pivotal RabbitMQ

To upgrade a non-clustered Pivotal RabbitMQ server from a previous release to the most current, simply install the new version. All configuration and persistent message data is retained.

To upgrade a clustered Pivotal RabbitMQ server, in the [Clustering Guide](#) see "Upgrading Clusters."

Troubleshooting Installation Problems

RHEL: RabbitMQ Server Does Not Start

Problem: When you first try to start the Pivotal RabbitMQ server on RHEL after installing it, you immediately get an error that the startup `FAILED`, and you see the following in the `/var/log/rabbitmq/startup_log` file:

```
=ERROR REPORT==== 6-Apr-2012::16:08:17 ===  
failed to start TCP Listener on [::]:5672 - eaddrinuse (address already in use)
```

Solution: An existing process (such as Apache Qpid) that is running and listening on port 5672, which is the default AMQP port. You must stop this process before you can start Pivotal RabbitMQ.

For example, if Apache Qpid is running, you can stop it and make sure it does not restart by default:

```
prompt$ sudo /sbin/service qpid stop & chkconfig --del qpid
```

Check with your system administrator for the specific steps that you should perform on your particular environment.

Next Steps

Pivotal RabbitMQ provides default built-in settings for running the RabbitMQ server. Without changing these values, you can work through the *RabbitMQ Tutorials* to learn the basics of creating messaging applications with RabbitMQ and Java. The Java client library distribution is shipped with a `rabbitmq-client-tests.jar`, containing several small example programs for exercising the functionality of the RabbitMQ server and client. Under *Java Client API Guide*, see "Bundled examples."

The following sections provide a roadmap to possible post-installation tasks such as specifying non-default values; setting up multiple server nodes; configuring SSL; configuring clients; and so forth.

Perform General Server Configuration Tasks

Implement SSL

Set Up Distributed Messaging and High Availability

Set Up and Work with Pivotal RabbitMQ Clients

Perform General Server Configuration Tasks

- *Configure non-default values for RabbitMQ server.* Specify non-default file locations, ports, account or service names, environment variables, or component properties.
- *Enable plug-ins and learn about plug-ins shipped with RabbitMQ.*
- *Understand access control permissions and set up access control with the `rabbitmqctl` command.* To perform an operation on a resource, a user must have appropriate permissions for it. For convenience RabbitMQ maps the AMQP default exchange's blank name to 'amq.default' when performing permission checks.
- *Understand flow control and configure the memory threshold* at which flow control is triggered.
- *Configure the Management Plugin*, an HTTP-based API for management and monitoring of your RabbitMQ server, along with a browser-based UI and a command line tool.
- *To facilitate debugging, enable the firehose feature.* With firehose enabled, you can see every message that is published and every message that is delivered, per node and per vhost.
- *Implement protocol extensions.* Pivotal RabbitMQ enhances the AMQP specification, with extensions for publishing, consuming, message routing, and the message life cycle.

Implement SSL

- *Set up SSL.* Create a certificate authority, and generate signed certificates for the server and clients in a number of formats.
- *Enable SSL in RabbitMQ server.* Under "SSL Support," see "Enabling SSL Support in RabbitMQ."
- *Enable SSL in the .NET client.* Under "SSL Support," see "Configuring the .NET Client."
- *Enable SSL in the Erlang client.* Under "SSL Support," see "Configuring the Erlang Client."
- *Understand the levels of trust, key managers, trust managers, and key stores.* You can create an encrypted SSL connection that requires certificate verification or one that does not require verification. The Java client supports both modes of operation. Under "SSL Support," see "Levels of Trust."
- *Connect without validating certificates.* Under "Levels of Trust," see "Connecting without validating certificates." An example shows a simple client connecting to a RabbitMQ server over SSL without validating the server certificate.
- *Present and validate certificates.* Under "Levels of Trust," see "Presenting and validating certificates." Set up certificate validation and connect.

Set Up Distributed Messaging and High Availability

- *Compare approaches, and decide how you will connect RabbitMQ brokers on more than one machine.* The options include clustering, federation, and shoveling. It is possible to mix approaches. Typically you use clustering for high availability and increased throughput, with machines in a single location. You use federation or shoveling to link brokers across the Internet for pub/sub messaging. The two approaches are similar; shoveling provides more control.
- *Create and run clusters with cluster commands.* Scripts show how to set up, start, restart, add nodes to, and remove nodes from clusters.
- *Auto-configure a cluster by using a default cluster config file.* Under "Clustering Transcripts," see "Autoconfiguration of a cluster." This approach is commonly used to automatically configure nodes to join a common cluster. The same configuration can be set on all nodes. Use of the cluster command overrides this approach.
- *Learn what the federation plugin does, when to use it, how to configure it.* To configure the plugin, under "Federation Plugin," see the "Implementation," "Getting Started," and "Configuration" sections.
- *Connect brokers by configuring the shovel plugin.*
- *Understand challenges and approaches to configuring highly available queues.*
- *Learn about mirrored queues.*
- *Create a mirrored queue.* Under "Highly Available Queues," see "Creating a Mirrored Queue." Configure queues to be mirrored on other nodes within a RabbitMQ cluster. Should one node of a cluster fail, the queue can automatically switch to one of the mirrors and continue to operate, with no loss of service.

Set Up and Work with Pivotal RabbitMQ Clients

- *Work through tutorials to create messaging applications.* Learn the basics of creating messaging applications with RabbitMQ, Java, and C#.
- *Perform key tasks related to RabbitMQ .NET clients.* Refer to *NET Client Library User Guide* and *.NET Client API Documentation*.
- *Perform key tasks related to RabbitMQ Java Clients.* Refer to the appropriate sections in the *Java Client API Guide* as indicated in the list below.
 - Connect to a broker and open a channel. See "Connections and channels" and "Connecting to a broker."
 - Increase the default allocation of consumer threads. See "Advanced Connection options."
 - Publish messages to an exchange. See "Publishing Messages." Set message properties and build a message properties object.
 - Receive messages. See "Receiving messages by subscription" and/or "Retrieving Individual Messages." The most efficient way to receive messages is to set up a subscription using the Consumer interface. The messages will then be delivered automatically as they arrive, rather than having to be explicitly requested.
 - Configure notification of returned messages to the sending client. See "Handling unroutable or undelivered messages." If the client has not configured a return listener for a particular channel, the associated returned messages are silently dropped.
- *Learn how to use JMS Client for Pivotal RabbitMQ.* Set up the JMS client software and configure applications to use it.