IBM Security Verify Privilege

Distributed Engine Guide

Contents

Last updated: September 21, 2020

Revision History

Date	**Change**
15 June 2020	Added note about site entitlement ‘Privilege Vault Distributed Engine’ section.
12 November 2019	Version update.
30 April 2019	Initial.

Privilege Vault Distributed Engine

Distributed Engine is a rewrite of the Privilege Vault agent and has been designed for scalability. Released in version 8.9.000000, Distributed Engine supports Active Directory Synchronization, Authentication, Heartbeat, Password Changing, Discovery, and SSH Proxying. Distributed Engine has three components – **Engines**, **Sites**, and **Site Connectors**.

SITES

“Sites” are logical groupings of work items. They can be thought of as labeled buckets. If your network has a DMZ and 2 remote offices that Privilege Vault cannot directly access, you would have 4 sites – “Local” (the default site representing the local network), “DMZ”, “Remote Office 1”, and “Remote Office 2”. Every Secret must now have a site (“Local” by default). This replaces the previous optional Agent assignment. Multiple engines can be assigned to the same site, significantly increasing throughput (2 or more Engines can simultaneously process Synchronizations / Authentications / Heartbeats / Password Changes / Discovery scans for the same site).

Note: IBM Security Privilege Vault provides entitlement for 10 sites and 10 engines per site.

ENGINES

“Engines” are the components that do the actual work. They are the component most similar to what the Privilege Vault agent was previously. Each engine can send data directly to Privilege Vault, as well as the ability to pull work items from the site.

SITE CONNECTORS

“Site Connectors” are the Windows services that hold the data for sites. Their job is to maintain the queue of work items. Two site connector types are supported by Privilege Vault: RabbitMq and MemoryMq. MemoryMq is a simple, in-memory site connector designed to be easy to get up and running. For best scalability and reliability, we recommend that customers use RabbitMq. A MemoryMq site connector can handle up to 100 sites (however, we recommend using no more than 50-60). A single RabbitMq site connector can handle up to 200 sites. Whether using RabbitMq or MemoryMq, Thycotic recommends installing Site Connectors on 64 bit Windows Server operating systems.

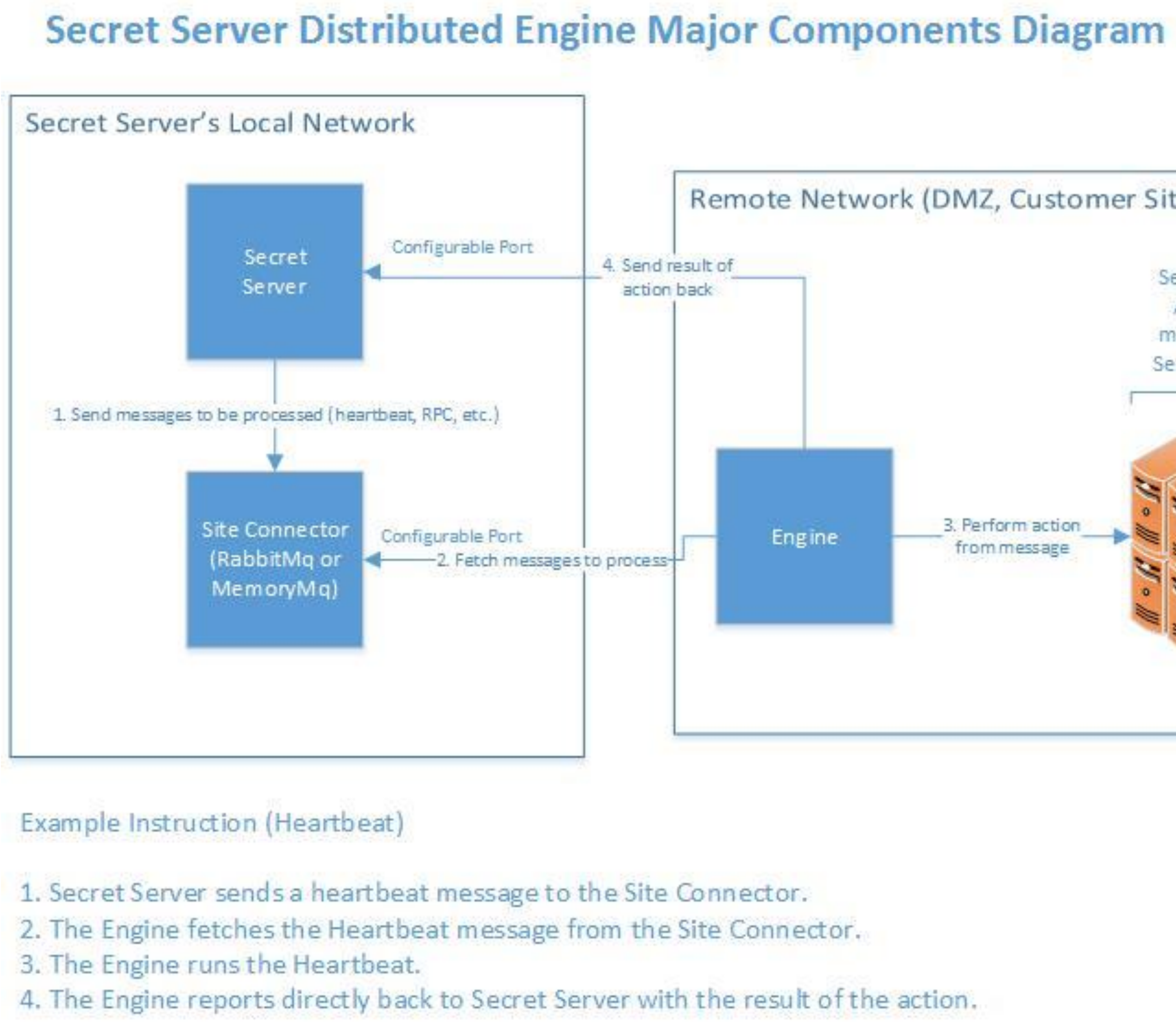
NOTE: Site Connectors are only required for on premise editions of Privilege Vault. Privilege Vault Cloud automatically uses an Azure Service Bus and this cannot be modified.

Ports

The major components diagram on the following page shows two configurable ports: one for connecting to the site connector (the site connector port), and one for sending responses back to Privilege Vault (the engine callback port). In Privilege Vault Cloud the Privilege Vault all callbacks from the engine are on port 443 and are not configurable.

The site connector port is the port on which the site connector listens for requests. These requests may be from Privilege Vault or from an engine. Privilege Vault contacts the site connector to place tasks into a site bucket, while engines contact the site connector to retrieve tasks to work on.

The engine callback port is the port on which Privilege Vault listens for responses from the engine. For example, “Secret 23 ran heartbeat and it was successful.” If the port field is blank, responses from the engine will be sent to the Privilege Vault website using whatever port the website is running on. If a specific port is specified, Privilege Vault will listen on that port for the responses.



Secret Server Distributed Engine Major Components Diagram

Secret Server's Local Network
Secret Server
Configurable Port
4. Send result of action back
1. Send messages to be processed (heartbeat, RPC, etc.)
Site Connector (RabbitMq or MemoryMq)
Configurable Port
2. Fetch messages to process
Remote Network (DMZ, Customer Site, etc.)
Servers with Accounts managed by Secret Server
Engine
3. Perform action from message

Example Instruction (Heartbeat)

1. Secret Server sends a heartbeat message to the Site Connector.
2. The Engine fetches the Heartbeat message from the Site Connector.
3. The Engine runs the Heartbeat.
4. The Engine reports directly back to Secret Server with the result of the action.

Engine Internal Workflow

The following list demonstrates the operations that occur in a typical engine workflow:

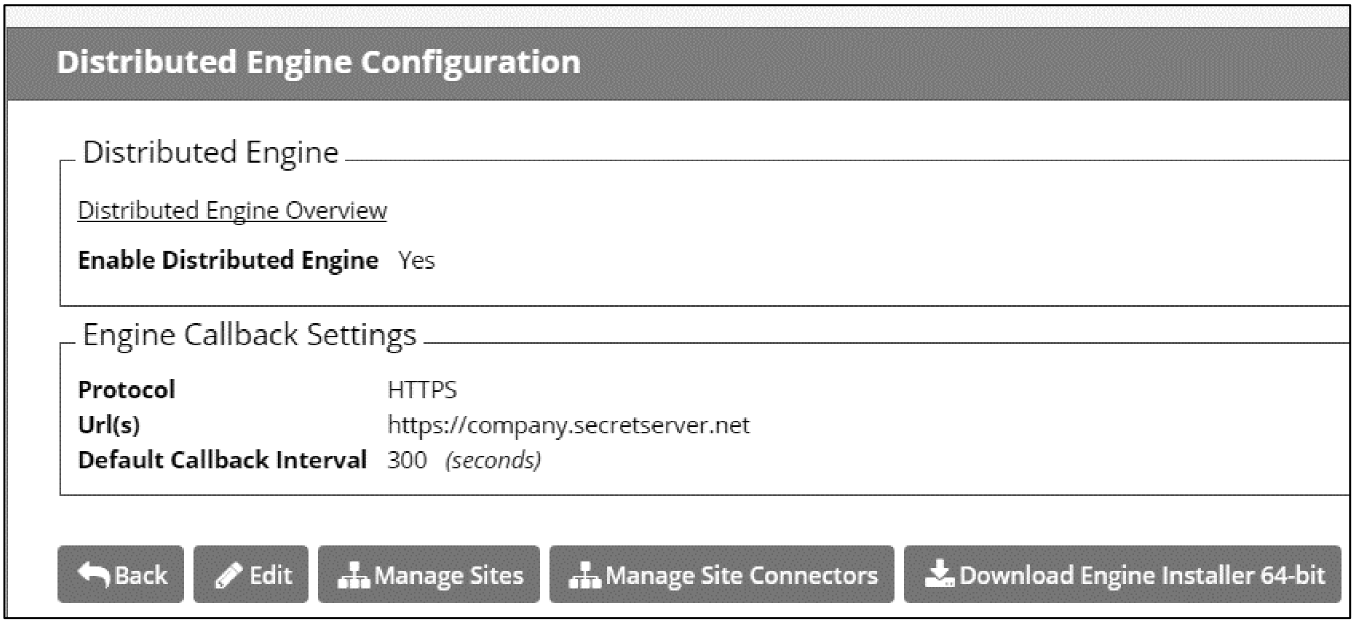
1. The engine starts up.
2. Connect to Privilege Vault using the engine callback port.
    a. Privilege Vault verifies that the engine is authorized.
    b. Privilege Vault sends back site connector & site information for the engine.
3. The engine connects to the site connector via the site connector port and looks for work in its site bucket.
4. The engine picks up as many work items as it can at one time and works on them.
5. When a work item is complete, the engine sends the result of the work item back to Privilege Vault using the engine callback port.
6. The engine picks up more work items when it has capacity.

Security

Security within Distributed Engine has multiple layers. Each site has its own symmetric key which is used to encrypt every message placed within the site. This means that all messages in the queue are encrypted at rest. Second, SSL is available when communicating with the site connector and back to Privilege Vault using the engine callback port, providing transport layer security. Third, when a Site Connector is configured, a username and password are created as part of the configuration. These are required in order to connect to the Site Connector and retrieve data.

Installing & Configuring Site Connectors

To configure Distributed Engine, first select **Distributed Engine** from the **ADMIN** menu within Privilege Vault. Click **Edit** and select **Enable Distributed Engine** to configure the Engine Callback Settings. **Default Callback Interval** is the callback period that each created site will require from its engines.



Distributed Engine Configuration

Distributed Engine

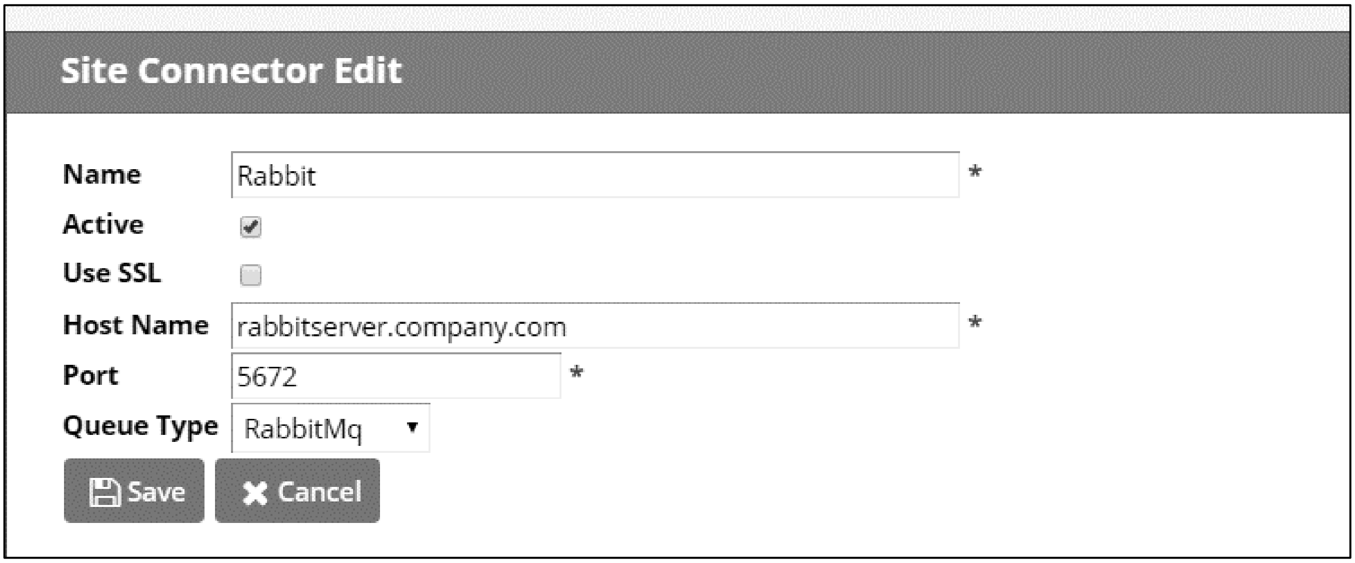
Distributed Engine Overview

Enable Distributed Engine Yes

Engine Callback Settings

Protocol	HTTPS
Url(s)	https://company.secretserver.net
Default Callback Interval	300 *(seconds)*

Once you've saved these settings, click **Manage Site Connectors** and then **New Site Connector**. Fill in the fields, then click **Save**.



Site Connector Edit

Name	Rabbit *
Active	☑
Use SSL	☐
Host Name	rabbitserver.company.com *
Port	5672 *
Queue Type	RabbitMq

- ✓ **Name** The name of the site connector (used in logs and within the Privilege Vault UI).
- ✓ **Use SSL** Whether connections to the site connector will use SSL for transport security or not.
- ✓ **Host Name** The name of the machine where the site connector will run.

- ✓ **Port** The port to listen for connections on (the host name and port are used by Privilege Vault and engines in order to connect to the site connector).
- ✓ **Queue Type** MemoryMq or RabbitMq. We recommend that RabbitMq be used in enterprise situations.

If you are using MemoryMq, click the **Download Site Connector Installer** button and run the .msi on the intended host. View the log to confirm that the MemoryMq service is listening for connections.

If you are using RabbitMq, we offer an installation helper console application to assist with the manual installation. The following section, **Using RabbitMq with Distributed Engine**, will walk you through the installation and configuration of RabbitMq.

Using RabbitMq with Distributed Engine

RabbitMq is the enterprise ready alternative to MemoryMq. MemoryMq is usually sufficient for basic and prototyping installations. RabbitMq is the choice messaging framework when the need for greater reliability and clustering arises. For feature information, please visit:

https://www.rabbitmq.com/features.html

The basic promise of both services is to provide site connectivity in order for distributed engines to be able to fulfill workloads. Currently, there is no installer that is generated by Privilege Vault for RabbitMq. However, we offer an installation helper console application to assist with the manual installation process:

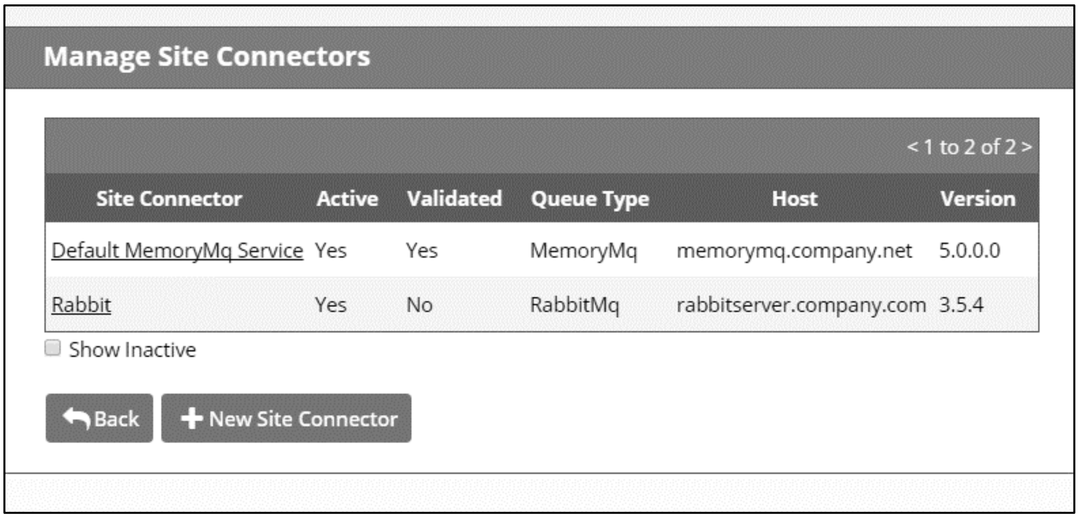
https://updates.thycotic.net/links.ashx?RabbitMqInstaller

RabbitMq uses credentials that are stored in its own database. These credentials are not local machine or domain accounts, they are specific to RabbitMq. Additionally, RabbitMq has the concept of a virtual host. It is sufficient to use the root virtual host (i.e. “/”) in most installations.

The Erlang runtime is a RabbitMq prerequisite.

PREPARING TO INSTALL

1. Navigate to Admin, Distributed Engine and click **Manage Site Connectors**. Select **New Site Connector**.



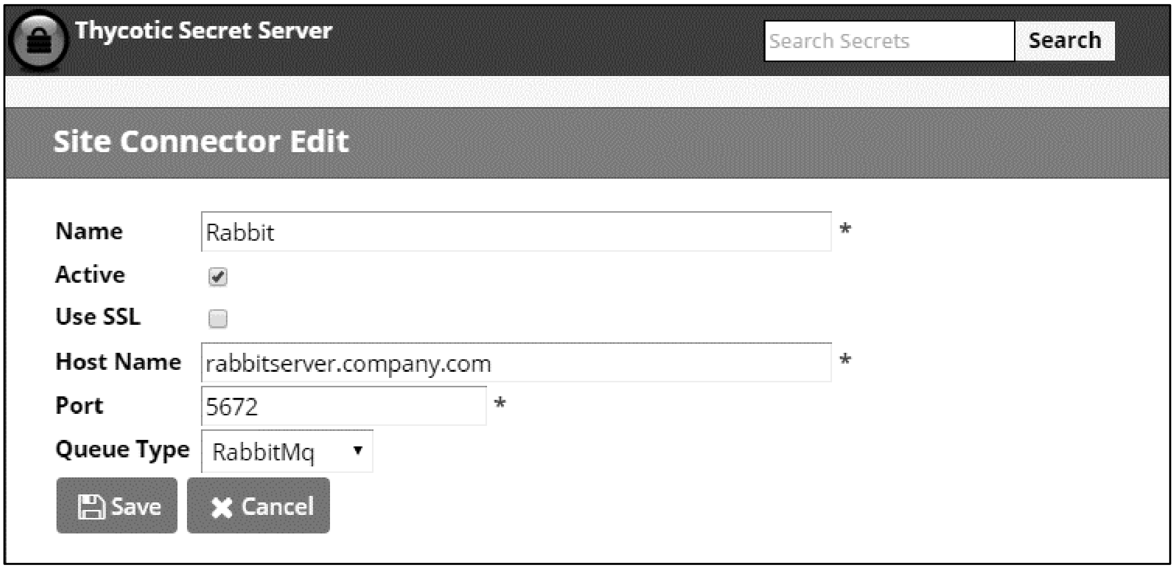
Manage Site Connectors

< 1 to 2 of 2 >

Site Connector	Active	Validated	Queue Type	Host	Version
Default MemoryMq Service	Yes	Yes	MemoryMq	memorymq.company.net	5.0.0.0
Rabbit	Yes	No	RabbitMq	rabbitserver.company.com	3.5.4

2. Enter a friendly **Name** for your new site connector, and mark it as **Active**. Since there are no sites using this connector yet, there are no problems with doing so.
3. Select the **Use SSL** box or leave it deselected according to whether or not this site connector should support SSL.
4. Enter the hostname of the machine where you just installed RabbitMq. Note that engines have to be able to resolve this hostname or connection will fail. Inbound firewall rules have to be created on the machine hosting the connector as well.

5. Enter the port for RabbitMq. The default port for non-SSL is 5672. The default port for SSL is 5671.
6. Select RabbitMq for the Queue Type.



Site Connector Edit

Name: Rabbit *
Active: ☑
Use SSL: □
Host Name: rabbitserver.company.com *
Port: 5672 *
Queue Type: RabbitMq ▾

7. After the connector is created, you can retrieve the automatically generated credentials by viewing the connector and clicking the **View Credentials** button. These credentials will be used shortly for the installation of RabbitMq.

NON-SSL INSTALLATION

Installing RabbitMq without transport level encryption (or non-SSL) simply requires an administrator to go through the steps below.

Using the Thycotic RabbitMq Helper

1. Download the Thycotic RabbitMq Helper from the following link: https://updates.thycotic.net/links.ashx?RabbitMqInstaller
2. Run the downloaded grmqh.msi from the command prompt. This installs the RabbitMq Helper in C:\Program Files\Thycotic Software Ltd\RabbitMq Helper.
3. Create a (**.bat**) file using the following code. Replace **rabbit_user** and **rabbit_pw** with the username and password from Step 7 of **Preparing to Install**.

✓

```
"C:\Program Files\Thycotic Software Ltd\RabbitMq Helper\Thycotic.RabbitMq.Helper.exe" install

-agreeErlangLicense=true ^

-agreeRabbitMqLicense=true ^

-rabbitMqUsername=rabbit_user ^

-rabbitMqPw=rabbit_pw ^

-useSsl=false
```

✓

4. Run the **.bat** file and it will will download the Erlang and RabbitMq installers. After installation completes, it will create the user and grant permissions to the root virtual host "/". Finally, it will install the RabbitMq management plug in and open a web browser to it. There should be no need to interact with the site at this time so you can minimize or close it for now. You can be find additional information about the site from a section later in this guide called **Additional Information about RabbitMq**.
5. RabbitMq will respond on the non-SSL port. You can modify this behavior with the additional options listed below.

✓

SSL INSTALLATION

Prior to installation, a valid server .pfx file and Certificate Authority .cer file needs to be available. The .pfx contains the certificate (self-signed or issued by an authority) to be used by RabbitMq and its private key. .pfx files are usually password-protected. The Certificate Authority .cer is used to establish the trust chain when connections are made RabbitMq. If using a self-signed certificate, use a .cer version of the server .pfx.

Installing RabbitMq with transport level encryption (or SSL) requires an administrator to go through the following steps:

1. Download the Thycotic RabbitMq Helper from the following link: https://updates.thycotic.net/links.ashx?RabbitMqInstaller
2. Run the downloaded grmqh.msi from the command prompt. This installs the RabbitMq Helper in C:\Program Files\Thycotic Software Ltd\RabbitMq Helper.
3. Create a (**.bat**) file using the following code. Replace **rabbit_user** and **rabbit_pw** with the username and password from Step 7 of **Preparing to Install**.

✓

```
"C:\Program Files\Thycotic Software Ltd\RabbitMq
Helper\Thycotic.RabbitMq.Helper.exe" installConnector ^

-agreeErlangLicense=true ^

-agreeRabbitMqLicense=true ^

-rabbitMqUsername=rabbit_user ^

-rabbitMqPw=rabbit_pw ^

-useSsl=false ^

-cacertpath="[your certificate directory]\SSLcert.cer" ^

-pfxPath="[your certificate directory]\SSLcert.pfx" ^

-pfxPw=password1
```

4. Run the **.bat** file and it will will download the Erlang and RabbitMq installers. After installation completes, it will create the user and grant permissions to the root virtual host “/”. Finally, it will install the RabbitMq management plug in and open a web browser to it. There should be no need to interact with the site at this time so you can minimize or close it for now. You can be find additional information about the site from a section later in this guide called **Additional Information about RabbitMq**.
5. RabbitMq will respond on the non-SSL and SSL ports. You can modify this behavior with the additional options listed below.

ADDITIONAL OPTIONS

agreeErlangLicense [true/false] Avoids prompting if you agree to the Erlang license

agreeRabbitMqLicense [true/false] Avoids prompting if you agree to the RabbitMq license

skipUserCreate [true/false] Skips creation of the initial user

offlineErlangInstallerPath [path] Uses an installer already on the machine at the specified path Supported installer located HERE.

offlineRabbitMqInstallerPath [path] Uses an installer already on the machine at the specified path Supported installer located HERE

Example 1

installConnector -agreeErlangLicense=true -agreeRabbitMqLicense=true -forceDownload=false -rabbitMqUsername=rabbit_user -rabbitMqPw=password1

Example 2

```
installConnector -agreeErlangLicense=true -agreeRabbitMqLicense=true -forceDownload=false -useSsl=true -
cacertpath=%USERPROFILE%\Desktop\sc.cer -pfxPath=%USERPROFILE%\Desktop\sc.pfx -
pfxPw=password1 -rabbitMqUsername=rabbit_user -rabbitMqPw=password1
```

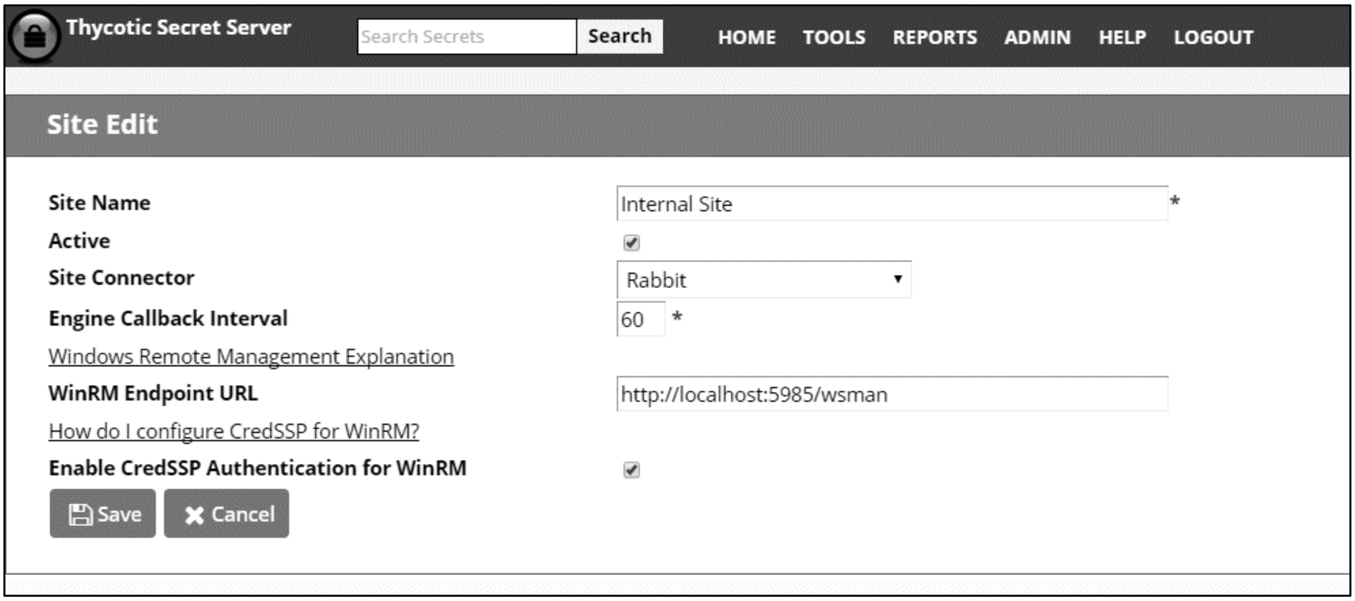
Example 3

```
installConnector -offlineErlangInstallerPath=%USERPROFILE%\Desktop\offline\erlang.exe -
offlineRabbitMqInstallerPath=%USERPROFILE%\Desktop\offline\rabbitMq.exe -agreeErlangLicense=true -
agreeRabbitMqLicense=true -forceDownload=false -useSsl=true -
cacertpath==%USERPROFILE%\Desktop\sc.cer -pfxPath=%USERPROFILE%\Desktop\sc.pfx -
pfxPw=password1 -rabbitMqUsername=rabbit_user  -rabbitMqPw=password1
```

Installing & Configuring Engines

CONFIGURING SITES (5 OR FEWER SITES)

Select **Distributed Engine** from the **ADMIN** menu within Privilege Vault. Click **Manage Sites** then select **New Site**, fill out the fields, and click **Save**.



Site Name	Internal Site *
Active	☑
Site Connector	Rabbit
Engine Callback Interval	60 *
Windows Remote Management Explanation	
WinRM Endpoint URL	http://localhost:5985/wsman
How do I configure CredSSP for WinRM?	
Enable CredSSP Authentication for WinRM	☑

- **Site Name** The name of the site.
- **Active** Active status of the site.
- **Site Connector** The site connector that will hold the work items for this site.
- **Engine Callback Interval** The interval (in seconds) for the engine to call back to Privilege Vault.
- **WinRM Endpoint URL** The WinRM endpoint URL to use when the engine makes Windows Remote Management calls (such as executing Powershell script dependencies).
- Enable CredSSP Authentication for WinRM Turns on CredSSP authentication.

Now that we have a site configured, we will download the engine installer for this site. From the View page for the internal site we just created, click **Download Engine Installer**. Install the Privilege Vault Distributed Engine .msi file on a server that has access to the accounts you want this site to manage. View the log for the engine to confirm that it starts successfully and can connect to the site connector.

CONFIGURING SITES (MANY SITES)

Select **Distributed Engine** from the **ADMIN** menu, then click **Download Engine Installer**. Run this installer in each location where you want an engine. For simplicity, we recommend that you install the engine in only a few places first.

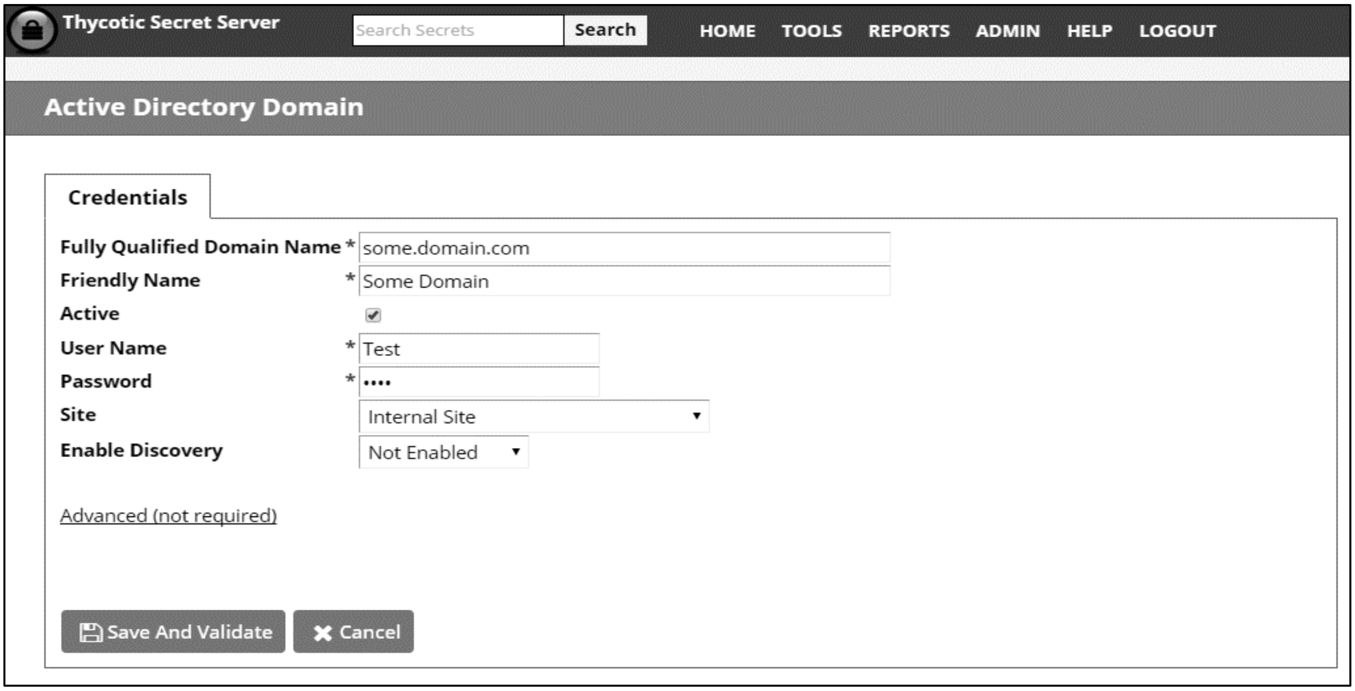
Next, click **Manage Sites** and then **Manage New Engines**. The installed engines will display here with an Activation Status of “Pending” and a default selection under Assigned Site of “Create Site”. Select each site that you wish to activate and select the **Assign and Activate Selected Engines** option from the drop-down menu. This displays a dialog where the site connector and Heartbeat interval can be specified. Click **OK**. The engines are now approved and each of them is assigned to an individual site.

ASSIGN SECRETS TO SITES

Assign one or more Secrets to the created sites. This can be done via bulk operation on Dashboard or through the Remote Password Changing tab on a Secret. Once assigned, any Heartbeat or password changing operations for that Secret will take place through the designated site.

Active Directory Synchronization & Authentication

To run Active Directory Synchronization through Distributed Engine, specify a site on the Active Directory Domain page. This will route Active Directory Synchronization operations through that site.



Active Directory Domain

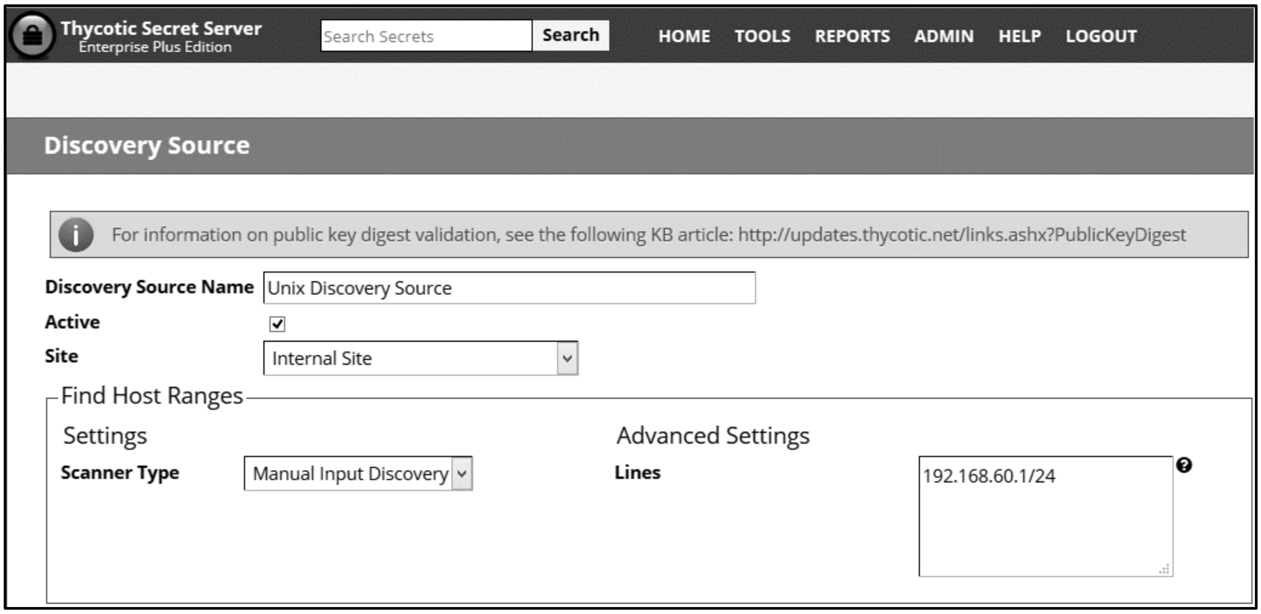
Credentials

Fully Qualified Domain Name *	some.domain.com
Friendly Name *	Some Domain
Active	☑
User Name *	Test
Password *	••••
Site	Internal Site ▾
Enable Discovery	Not Enabled ▾

When a user authenticates through a domain that is assigned to a site that has one or more active engines then the authentication will be performed through Distributed Engine.

Discovery

To run Discovery through Distributed Engine, specify a site on the Discovery Source page. This will route Discovery operations through that site.



Discovery Source

For information on public key digest validation, see the following KB article: http://updates.thycotic.net/links.ashx?PublicKeyDigest

Discovery Source Name: Unix Discovery Source

Active: ☑

Site: Internal Site

Find Host Ranges

Settings

Scanner Type: Manual Input Discovery

Advanced Settings

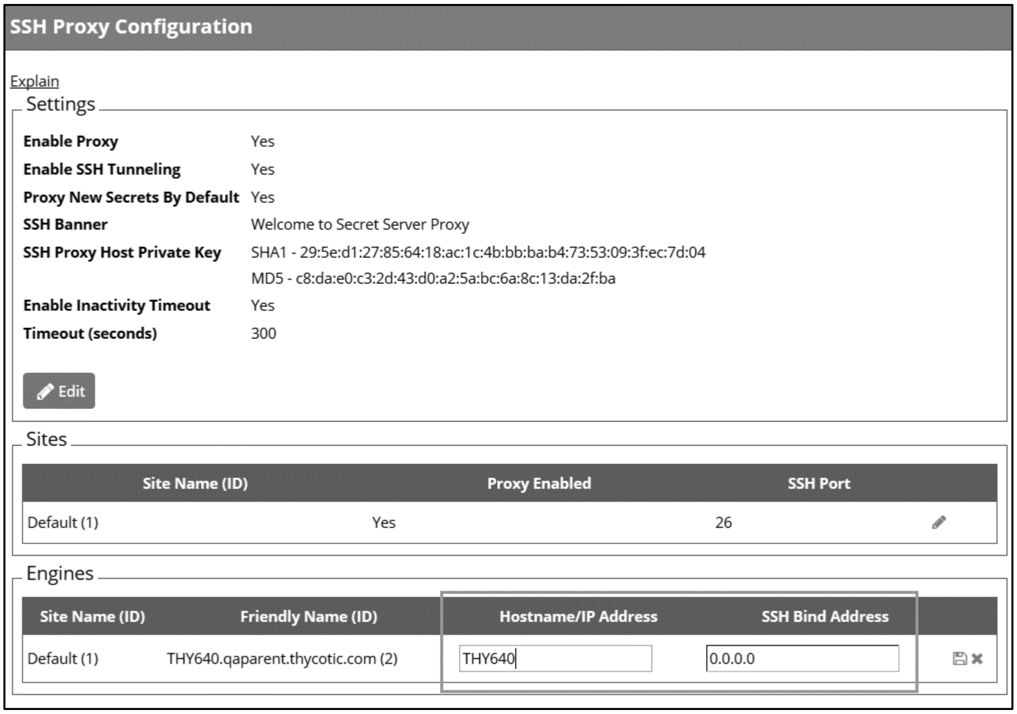
Lines: 192.168.60.1/24

Proxying

To enable Engines to serve as proxies for remote RDP and SSH Sessions, ensure that Proxying is enabled under **ADMIN | SSH Proxy.**

Edit the site you want to enable as an SSH proxy and check the box for **Enable Proxy** and optionally choose a custom port.

Under **ADMIN | SSH Proxy** you can see all engines that are allowed for proxying and set the Public Host and Bind Address for each engine. Privilege Vault will default these values to the FQDN of the machine and 0.0.0.0 which should work without modifications in many situations.



SSH Proxy Configuration

Explain

Settings

Enable Proxy	Yes
Enable SSH Tunneling	Yes
Proxy New Secrets By Default	Yes
SSH Banner	Welcome to Secret Server Proxy
SSH Proxy Host Private Key	SHA1 - 29:5e:d1:27:85:64:18:ac:1c:4b:bb:ba:b4:73:53:09:3f:ec:7d:04 MD5 - c8:da:e0:c3:2d:43:d0:a2:5a:bc:6a:8c:13:da:2f:ba
Enable Inactivity Timeout	Yes
Timeout (seconds)	300

Sites

Site Name (ID)	Proxy Enabled	SSH Port
Default (1)	Yes	26

Engines

Site Name (ID)	Friendly Name (ID)	Hostname/IP Address	SSH Bind Address
Default (1)	THY640.qaparent.thycotic.com (2)	THY640	0.0.0.0

Diagnosing Configuration Problems

PROBLEMS WITH THE MEMORYMQ SITE CONNECTOR

Validate Connectivity

There is a **Validate Connectivity** button on the Site Connector view page that tests the entire round trip through an engine. Use this to quickly deduce if Privilege Vault is connecting correctly to the Site Connector or not.

Is the service running?

If the service cannot be started, verify that the credentials on the service have appropriate permissions (Run as Service, full rights to the folder where MemoryMq is installed).

Is the site connector listening for connections?

Check the log file inside the log folder to see if any errors were reported. If the machine’s hostname does not match the host name specified on the site connector page within Privilege Vault, the site connector will not start.

PROBLEMS WITH THE ENGINE / MESSAGE PROCESSING

Confirm an Engine is assigned to the Site

Until an Engine has connected to the Site Connector for a particular Site, the work queues that hold the messages for that Site do not exist. This means that any messages sent from Privilege Vault to that Site will disappear silently. When tracking down issues, make sure that there is an Engine assigned to the Site.

Validate Connectivity

There is a **Validate Connectivity** button on the Site view page that tests the entire round trip through an engine. Use this to quickly deduce if everything is connecting correctly or not.

Is the service running?

If the service cannot be started, verify that the credentials on the service have appropriate permissions (Run as Service, full rights to the folder where the service is installed).

Is the Engine able to connect to Privilege Vault?

The first thing the engine does when it starts is connect to Privilege Vault using the engine callback port to find out what site connector/site it should connect to. Check the log files to see if it was able to connect. If this is a fresh upgrade or install of Privilege Vault installed on Server 2012, there might be missing Windows features. See the following KB article for more information.

Is the Engine active?

If the engine is inactive it will continually try to connect to Privilege Vault until it is made active.

Is the Engine able to connect to the site connector?

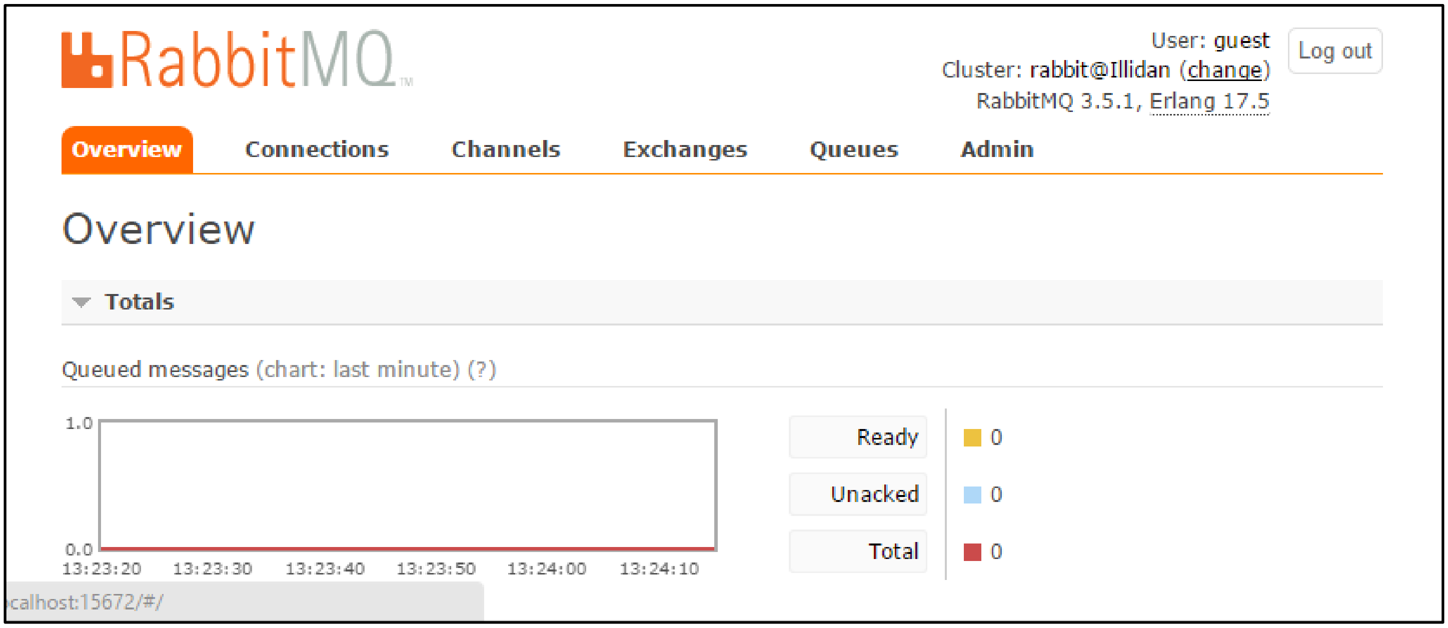
If no, ensure that:

a) the site connector is running at the correct address
b) the site connector is listening for connections
c) there is connectivity from the machine where the Engine is installed to the site connector (check firewalls, etc.)

Additional Information about RabbitMq

- **Erlang runtime** http://www.erlang.org
- **RabbitMq** https://www.rabbitmq.com
- **RabbitMq management** https://www.rabbitmq.com/management.html

The management web console is an easy way to manage and monitor the RabbitMq installation:



User: guest
Cluster: rabbit@Illidan (change)
RabbitMQ 3.5.1, Erlang 17.5

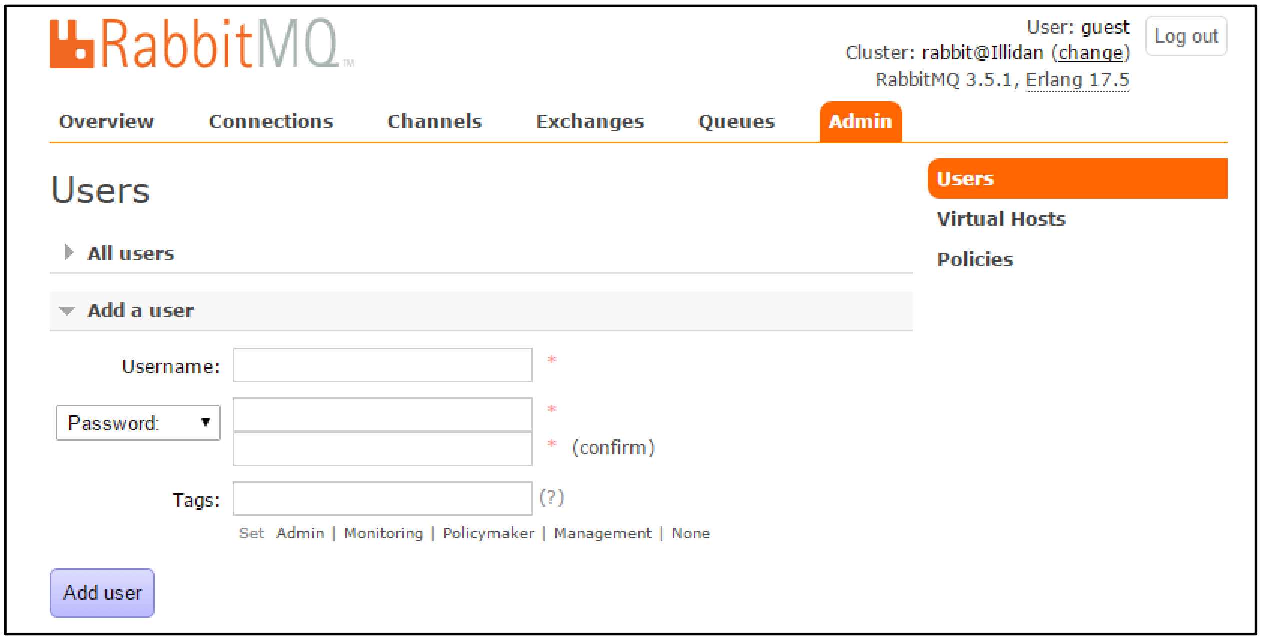
Unless already installed by the helper, you can open a command prompt with elevated privileges and navigate to the RabbitMq bin directory. For RabbitMq version 3.5.3 the path is C:\Program Files (x86)\RabbitMQ Server\rabbitmq_server-3.5.3\sbin. Run *rabbitmq-plugins enable rabbitmq_management*

USER CREATION

1. Open the RabbitMq management web site. The default address is http://localhost:15672/
2. If you are running the web site on the machine where it is installed, unless it is already deleted, you may use the default guest account (username: guest, password: guest). That account comes built in with RabbitMq but can only be used on localhost.

+ In production environments, the guest account should be deleted.

3. Navigate to the **Admin** tab. The subtab should be **Users**.
4. Expand the **Add a user** section and enter the desired credentials.



User: guest
Cluster: rabbit@Illidan (change)
RabbitMQ 3.5.1, Erlang 17.5

Users

Username: *

Password: *
* (confirm)

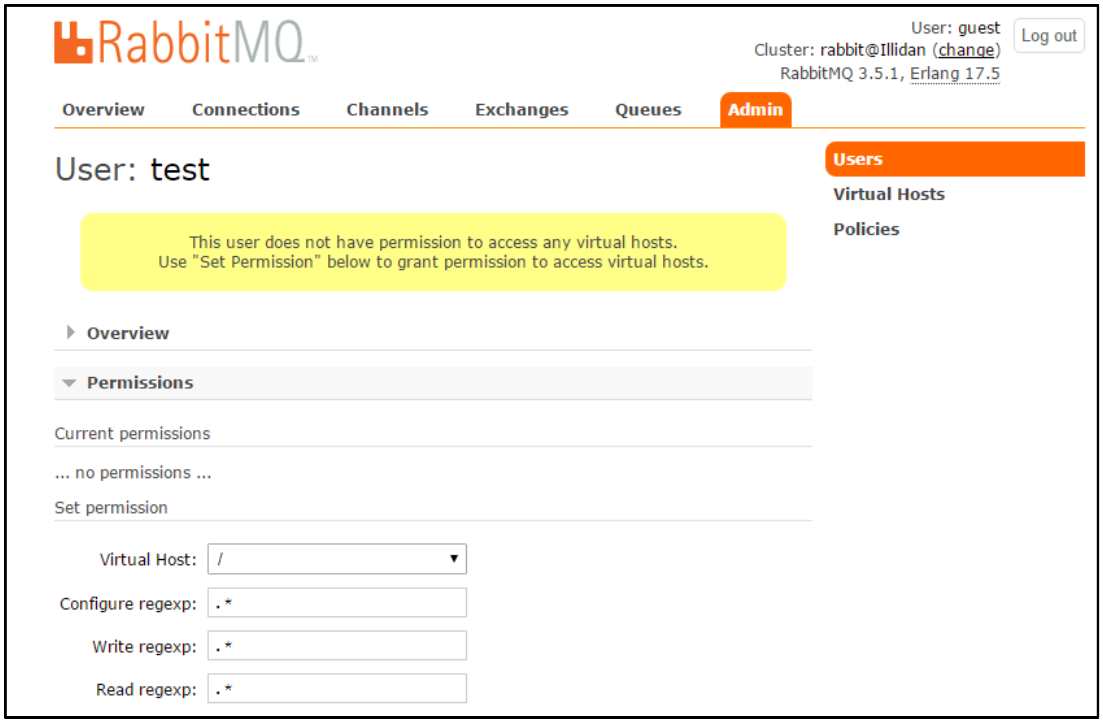
Tags: (?)

5. The user you just created should now be listed under the **All users** tab.

By default, users have access to submit and process requests in RabbitMq. However, they do not have access to the management plugin. To enable access, add the **Tag** that is applicable.

GRANTING PERMISSIONS TO USERS

Select the user you just created by clicking the user name.



User: guest
Cluster: rabbit@Illidan (change)
RabbitMQ 3.5.1, Erlang 17.5

User: test

This user does not have permission to access any virtual hosts.
Use "Set Permission" below to grant permission to access virtual hosts.

Current permissions

... no permissions ...

Set permission

Virtual Host: /

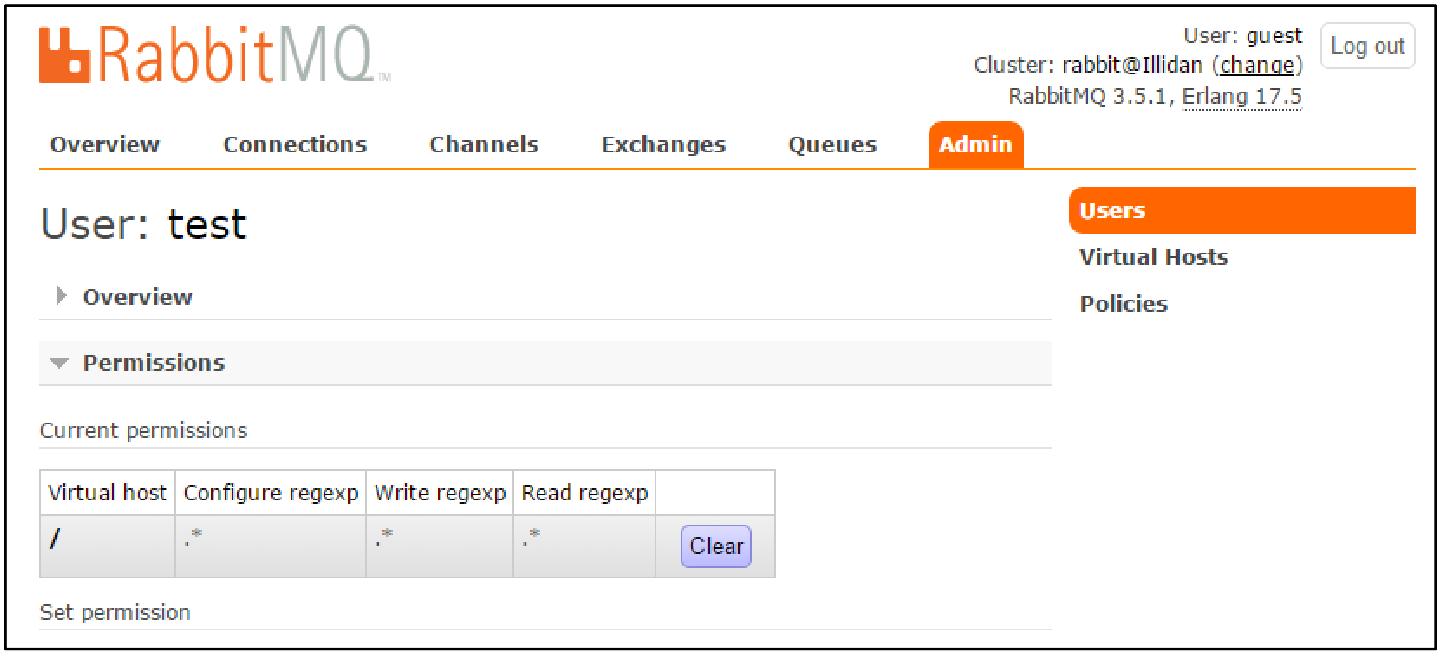
Configure regexp: .*

Write regexp: .*

Read regexp: .*

Even though the above user is in the system and has credentials, it is not yet allowed to access any virtual hosts. Under the **Permissions** tab select the root host (i.e "/"), leave the rest of the fields with their default values and click **Set permission**.

Now the user is ready to be used for a site connection.



User: test

Current permissions

Virtual host	Configure regexp	Write regexp	Read regexp	
/	.*	.*	.*	Clear

Set permission

Manual RabbitMq Installation

Thycotic does not recommend installing RabbitMq manually. However, if that is a need or the helper does not meet your requirement, follow the manual steps below.

BASIC INSTALLATION WITHOUT ENCRYPTION SUPPORT

1) Download the applicable installer for the Erlang runtime required to run RabbitMq from http://packages.erlang-solutions.com/site/esl/esl-erlang/FLAVOUR_1_general/esl-erlang_17.5-1~windows_amd64.exe or from http://www.erlang.org/download.html
2) Run the downloaded installer.
3) Download the applicable installer for RabbitMq from https://www.rabbitmq.com/releases/rabbitmq-server/v3.5.3/rabbitmq-server-3.5.3.exe or from https://www.rabbitmq.com/download.html
4) Run the downloaded installer.
5) Enable the RabbitMq Management plug-in by following the instructions in https://www.rabbitmq.com/management.html
6) After management plug-in installation completes open http://server-name:15672/ and create your site connector user.
    a. Log in and go to the **Admin** tab and click **Add User**
    b. Fill out the user name and password.
    c. Unless the user you are creating will be used for management or monitoring purposes, you do not need to add any Tags.
    d. After the user is created, navigate to the **Permissions** section for that user and grant them the default permissions to the "/" virtual host.

ADVANCED INSTALLATION WITH ENCRYPTION SUPPORT

1) Download the applicable installer for the Erlang runtime required to run RabbitMq from http://packages.erlang-solutions.com/site/esl/esl-erlang/FLAVOUR_1_general/esl-erlang_17.5-1~windows_amd64.exe or from http://www.erlang.org/download.html
2) Run the downloaded installer.
3) Download the applicable installer for RabbitMq from https://www.rabbitmq.com/releases/rabbitmq-server/v3.5.3/rabbitmq-server-3.5.3.exe or from https://www.rabbitmq.com/download.html
4) Run the downloaded installer.
5) Enable the RabbitMq Management plug-in by following the instructions in https://www.rabbitmq.com/management.html
6) After management plug-in installation completes open http://server-name:15672/
    a. On the **Overview** screen find the **Node** section and copy the configuration file path. Save the path to use later on in these instructions.
    b. If next to the path you see "not found" go to the installation folder of RabbitMq (i.e C:\Program Files (x86)\RabbitMQ Server\rabbitmq_server-3.5.3\etc), copy the rabbitmq.config.example file and place it on your desktop.

c. Rename the file to rabbitmq.config and open it for editing.
   i. Uncomment the line under network connectivity for ssl_listeners and specify the port you would like to use or leave the default of 5671

```
{ssl_listeners, [5671]},
```

   ii. Uncomment the lines under security for ssl_options and specify the certificate authority certificate, server certificate you'd like to use and the key file for that server certificate.
   iii. RabbitMq requires .PEM files and will not work with .PFX and .CER (you will need openSSL or other suitable tools to convert a .PXF to .PEM)

```
{ssl_options, [          {cacertfile, "%PATHTOCACERT%"},

                         {certfile,   "%PATHTOCERT%"},

                         {keyfile,    "%PATHTOKEY%"},

                 {verify,     verify_peer},

                         {fail_if_no_peer_cert, false}]}
```

   iv. Full SSL documentation can be found at http://www.rabbitmq.com/ssl.html

7) Unfortunately at the time of this writing, simply copying and pasting the configuration file in the location RabbitMq requires is not sufficient.
8) Ensure that you have the configuration path you copied and stored previously from the management plugin.
9) Uninstall the RabbitMq server.
10) Recreate the folder structure for the configuration path.
11) Copy the modified configuration file to that location.
12) Reinstall RabbitMq.
13) Re-Enable the management plugin.
14) Open http://server-name:15672/ and verify that under **Ports and Contexts** the port you selected is listed for SSL.

Troubleshooting

SSL issues

1) If you see the port listed for use with SSL in the management web site but you are unable to connect to RabbitMq using a SSL connection, take a look at the RabbitMq log files. The locations for those files are listed under the **Overview** tab of the management web site. In most cases the issues are with incorrect CA certificates, self-signed certificates and/or incorrect certificate format.

Management Plugin Does Not Load

The management plugin is the web page where the user logs in to manage RabbitMQ. This is usually http://localhost:15672. When installing RabbitMQ using our helper utility the management plugin should be automatically enabled and the Site Connector account specified in the helper arguments automatically created and granted the appropriate permissions. However, if any error occurs during the installation while using the helper, the management plugin may not have been enabled.

1. Open a command prompt as administrator and change to the RabbitMQ directory: "C:\Program Files (x86)\RabbitMQ Server\rabbitmq_server-3.5.3\sbin"
2. Enter the following command and press [Enter]:

```
rabbitmq-plugins enable rabbitmq_management
```

Example RabbitMq configuration file with encryption

See this Knowledge Base article for the full text of the example file.