

Spring Cloud Data Flow for VMware GemFire Documentation

Spring Cloud Data Flow for VMware GemFire 1.0

You can find the most up-to-date technical documentation on the VMware by Broadcom website at:

<https://docs.vmware.com/>

VMware by Broadcom
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

Copyright © 2024 Broadcom. All Rights Reserved. The term “Broadcom” refers to Broadcom Inc. and/or its subsidiaries. For more information, go to <https://www.broadcom.com>. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies. [Copyright and trademark information](#).

Contents

Spring Cloud Data Flow for VMware GemFire Documentation	5
Release Notes	6
1.0.0	6
Compatibility and Versions	7
Compatibility	7
Getting Started	8
Installing Into Spring Cloud Data Flow	8
GemFire Sink Rabbit	9
Docker Hub Images	9
Commercial Maven Repository Artifacts	9
Properties:	10
gemfire.consumer	10
gemfire.pool	10
gemfire.region	10
gemfire.security	10
gemfire.security.ssl	10
GemFire Source Rabbit	12
Docker Hub Images	12
Commercial Maven Repository Artifacts	12
Properties:	13
gemfire.client	13
gemfire.pool	13
gemfire.region	13
gemfire.security	13
gemfire.security.ssl	13
gemfire.supplier	14
GemFire Sink Kafka	15
Docker Hub Images	15
Commercial Maven Repository Artifacts	15

Properties:	16
gemfire.consumer	16
gemfire.pool	16
gemfire.region	16
gemfire.security	16
gemfire.security.ssl	16
GemFire Source Kafka	18
Docker Hub Images	18
Commercial Maven Repository Artifacts	18
Properties:	19
gemfire.client	19
gemfire.pool	19
gemfire.region	19
gemfire.security	19
gemfire.security.ssl	19
gemfire.supplier	20

Spring Cloud Data Flow for VMware GemFire Documentation

Spring Cloud Dataflow for VMware Tanzu GemFire is a project that defines integration with the [Spring Cloud Stream](#) and [Spring Cloud Stream Applications](#) projects.

The published artifacts are:

- `gemfire-sink-rabbit`
- `gemfire-source-rabbit`
- `gemfire-sink-kafka`
- `gemfire-source-kafka`

These artifacts are then “installed” into a [Spring Cloud Dataflow Server](#) in order to receive or send data from VMware Tanzu GemFire instances.

Release Notes

This topic contains the release notes for Spring Cloud Dataflow for VMware GemFire.

1.0.0

- Initial release of Spring Cloud Stream AppsFor VMware Tanzu GemFire, for Spring Cloud Dataflow Server 2.10.x and 2.11.1 and VMware GemFire 9.15.x and 10.0.x.

This includes the following bindings for RabbitMQ and Kafka:

- gemfire-source-rabbit
- gemfire-sink-rabbit
- gemfire-source-kafka
- gemfire-sink-kafka

Compatibility and Versions

This topic list Spring Cloud Dataflow for VMware GemFire compatibility and versions.

Compatibility

Spring Cloud Stream App Artifact	Latest Versions	Compatible GemFire Versions	Compatible Spring Cloud Dataflow Server
gemfire-source-rabbit	1.0.0	9.15.x, 10.0.x, 10.1.x	2.10.x , 2.11.x
gemfire-source-kafka	1.0.0	9.15.x, 10.0.x, 10.1.x	2.10.x, 2.11.x
gemfire-sink-rabbit	1.0.0	9.15.x, 10.0.x, 10.1.x	2.10.x, 2.11.x
gemfire-sink-kafka	1.0.0	9.15.x, 10.0.x, 10.1.x	2.10.x, 2.11.x

Getting Started

This guide explains how to download and install the Spring Cloud Dataflow for VMware GemFire libraries onto a Spring Cloud Data Flow server.

You can download the Spring Cloud Dataflow for VMware GemFire libraries from either [Docker Hub](#) or the [Pivotal Commercial Maven Repository](#). Note that access to the Pivotal Commercial Maven Repository requires a one-time registration to create an account.

Installing Into Spring Cloud Data Flow

This project offers implementations for VMware Tanzu GemFire that are compatible with the Kafka and RabbitMQ bindings.

Follow the instructions provided in the links below to install the appropriate binder for your Spring Cloud Data Flow server:

- [GemFire Sink RabbitMQ](#)
- [GemFire Source RabbitMQ](#)
- [GemFire Sink Kafka](#)
- [GemFire Source Kafka](#)

GemFire Sink Rabbit

There are two methods to deploy the artifacts into the Spring Cloud Dataflow Server: 1. Using the images from Docker Hub 2. Download the artifacts from the Pivotal Commercial Maven repository.

Choose one of the following options.

Docker Hub Images

1. Log in to your Spring Cloud Dataflow Server.
2. Add the Application using the **Add Application** button.
3. Select the first option **Register one or more applications** and enter the following information:
 - **Name:** `gemfire` or `gemfire-sink-rabbit`.
 - **Type:** `sink`.
 - **Spring Boot version:** `Spring Boot 2.x`.
 - **URI:** `docker://docker.io/gemfire/gemfire-sink-rabbit:1.0.0`
4. Click **Import Application** to import the GemFire Sink for Rabbit Stream Application.

Commercial Maven Repository Artifacts

1. In a browser, navigate to the [Pivotal Commercial Maven Repository](#).
2. Click the **Create Account** link.
3. Complete the information in the registration page.
4. Click **Register**.
5. After registering, you will receive a confirmation email. Follow the instruction in this email to activate your account.
6. Once you have a Pivotal Commercial Maven Repository account created, you can now download the GemFire Sink Rabbit artifact (replace `{commercialMavenRepoUsername}` with your commercial maven repo username/email):

```
wget https://commercial-repo.pivotal.io/data3/gemfire-release-repo/gemfire/com/
vmware/gemfire/spring/cloud/stream/app/gemfire-sink-rabbit/1.0.0/gemfire-sink-r
abbit-1.0.0.jar --user {commercialMavenRepoUsername} --ask-password
```

7. Log in to your Spring Cloud Dataflow Server.
8. Add the Application using the **Add Application** button.

9. Select the third option, **Import application coordinates from properties file**.
10. Add the `sink` properties, replacing `{artifactFileName}` with the location of the downloaded artifacts from step 2. You can change the application name (`gemfire`) as needed.

```
sink.gemfire=file://{artifactFileName}
```

11. Click **Import Application** to import the GemFire Sink for Rabbit Stream Application. It may take a minute for this application to appear in the SCDF Applications list.

Properties:

gemfire.consumer

Property Name	Description	Type	Defaults
json	Indicates if the GemFire region stores json objects as PdxInstance.	Boolean	false
key-expression	SpEL expression to use as a cache key.	String	

gemfire.pool

Property Name	Description	Type	Defaults
connect-type	Specifies connection type: 'server' or 'locator'.	ConnectType	
host-addresses	Specifies one or more GemFire locator or server addresses formatted as [host]:[port].	InetSocketAddress[]	
subscription-enabled	Set to true to enable subscriptions for the client pool. Required to sync updates to the client cache.	Boolean	false

gemfire.region

Property Name	Description	Type	Defaults
region-name	The region name.	String	

gemfire.security

Property Name	Description	Type	Defaults
password	The cache password.	String	
username	The cache username.	String	

gemfire.security.ssl

Property Name	Description	Type	Defaults
ciphers	Configures the SSL ciphers used for secure Socket connections as an array of valid cipher names.	String	any

keystore-type	Identifies the type of Keystore used for SSL communications (e.g. JKS, PKCS11, etc.).	String	JKS
keystore-uri	Location of the pre-created Keystore URI to be used for connecting to the GemFire cluster.	Resource	
ssl-keystore-password	Password for accessing the keys truststore.	String	
ssl-truststore-password	Password for accessing the trust store.	String	
truststore-type	Identifies the type of truststore used for SSL communications (e.g. JKS, PKCS11, etc.).	String	JKS
truststore-uri	Location of the pre-created truststore URI to be used for connecting to the GemFire cluster.	Resource	
user-home-directory	Local directory to cache the truststore and keystore files downloaded from the truststoreUri and keystoreUri locations.	String	user.home

GemFire Source Rabbit

There are two methods to deploy the artifacts into the Spring Cloud Dataflow Server: 1. Using the images from Docker Hub 2. Download the artifacts from the Pivotal Commercial Maven repository.

Choose one of the following options.

Docker Hub Images

1. Log in to your Spring Cloud Dataflow Server.
2. Add the Application using the **Add Application** button.
3. Select the first option **Register one or more applications** and enter the following information:
 - **Name:** `gemfire` or `gemfire-source-rabbit`.
 - **Type:** `source`.
 - **Spring Boot version:** `Spring Boot 2.x`.
 - **URI:** `docker://docker.io/gemfire/gemfire-source-rabbit:1.0.0`
4. Click **Import Application** to import the GemFire Source for Rabbit Stream Application.

Commercial Maven Repository Artifacts

1. In a browser, navigate to the [Pivotal Commercial Maven Repository](#).
2. Click the **Create Account** link.
3. Complete the information in the registration page.
4. Click **Register**.
5. After registering, you will receive a confirmation email. Follow the instruction in this email to activate your account.
6. Once you have a Pivotal Commercial Maven Repository account created, you can now download the GemFire Source Rabbit artifact (replace `{commercialMavenRepoUsername}` with your commercial maven repo username/email):

```
wget https://commercial-repo.pivotal.io/data3/gemfire-release-repo/gemfire/com/
vmware/gemfire/spring/cloud/stream/app/gemfire-source-rabbit/1.0.0/gemfire-sour
ce-rabbit-1.0.0.jar --user {commercialMavenRepoUsername} --ask-password
```

7. Log in to your Spring Cloud Dataflow Server.
8. Add the Application using the **Add Application** button.

9. Select the third option, **Import application coordinates from properties file**.
10. Add the `source` properties, replacing `{artifactFileName}` with the location of the downloaded artifacts from step 2. You can change the application name (`gemfire`) as needed.

```
source.gemfire=file://{artifactFileName}
```

11. Click **Import Application** to import the GemFire Source for Rabbit Stream Application. It may take a minute for this application to appear in the SCDF Applications list.

Properties:

gemfire.client

Property Name	Description	Type	Defaults
pdx-read-serialized	Deserialize the GemFire objects into PdxInstance instead of the domain class.	Boolean	false

gemfire.pool

Property Name	Description	Type	Defaults
connect-type	Specifies connection type: 'server' or 'locator'.	ConnectType	
host-addresses	Specifies one or more GemFire locator or server addresses formatted as [host]:[port].	InetSocketAddress[]	
subscription-enabled	Set to true to enable subscriptions for the client pool. Required to sync updates to the client cache.	Boolean	false

gemfire.region

Property Name	Description	Type	Defaults
region-name	The region name.	String	

gemfire.security

Property Name	Description	Type	Defaults
password	The cache password.	String	
username	The cache username.	String	

gemfire.security.ssl

Property Name	Description	Type	Defaults
---------------	-------------	------	----------

ciphers	Configures the SSL ciphers used for secure Socket connections as an array of valid cipher names.	String	any
keystore-type	Identifies the type of Keystore used for SSL communications (e.g. JKS, PKCS11, etc.).	String	JKS
keystore-uri	Location of the pre-created Keystore URI to be used for connecting to the GemFire cluster.	Resource	
ssl-keystore-password	Password for accessing the keys truststore.	String	
ssl-truststore-password	Password for accessing the trust store.	String	
truststore-type	Identifies the type of truststore used for SSL communications (e.g. JKS, PKCS11, etc.).	String	JKS
truststore-uri	Location of the pre-created truststore URI to be used for connecting to the GemFire cluster.	Resource	
user-home-directory	Local directory to cache the truststore and keystore files downloaded from the truststoreUri and keystoreUri locations.	String	user.home

gemfire.supplier

Property Name	Description	Type	Defaults
event-expression	SpEL expression to extract data from an {@link org.apache.geode.cache.EntryEvent} or {@link org.apache.geode.cache.query.CqEvent}.	Expression	
query	An OQL query. This will enable continuous query if provided.	String	

GemFire Sink Kafka

There are two methods to deploy the artifacts into the Spring Cloud Dataflow Server: 1. Using the images from Docker Hub 2. Download the artifacts from the Pivotal Commercial Maven repository.

Choose one of the following options.

Docker Hub Images

1. Log in to your Spring Cloud Dataflow Server.
2. Add the Application using the **Add Application** button.
3. Select the first option **Register one or more applications** and enter the following information:
 - **Name:** `gemfire` or `gemfire-sink-kafka`.
 - **Type:** `sink`.
 - **Spring Boot version:** `Spring Boot 2.x`.
 - **URI:** `docker://docker.io/gemfire/gemfire-sink-kafka:1.0.0`
4. Click **Import Application** to import the GemFire Sink for Kafka Stream Application.

Commercial Maven Repository Artifacts

1. In a browser, navigate to the [Pivotal Commercial Maven Repository](#).
2. Click the **Create Account** link.
3. Complete the information in the registration page.
4. Click **Register**.
5. After registering, you will receive a confirmation email. Follow the instruction in this email to activate your account.
6. Once you have a Pivotal Commercial Maven Repository account created, you can now download the GemFire Sink Kafka artifact (replace `{commercialMavenRepoUsername}` with your commercial maven repo username/email):

```
wget https://commercial-repo.pivotal.io/data3/gemfire-release-repo/gemfire/com/
vmware/gemfire/spring/cloud/stream/app/gemfire-sink-kafka/1.0.0/gemfire-sink-ka
fka-1.0.0.jar --user {commercialMavenRepoUsername} --ask-password
```

7. Log in to your Spring Cloud Dataflow Server.
8. Add the Application using the **Add Application** button.

9. Select the third option, **Import application coordinates from properties file**.
10. Add the `sink` properties, replacing `{artifactFileName}` with the location of the downloaded artifacts from step 2. You can change the application name (`gemfire`) as needed.

```
sink.gemfire=file://{artifactFileName}
```

11. Click **Import Application** to import the GemFire Sink for Kafka Stream Application. It may take a minute for this application to appear in the SCDF Applications list.

Properties:

gemfire.consumer

Property Name	Description	Type	Defaults
json	Indicates if the GemFire region stores json objects as PdxInstance.	Boolean	false
key-expression	SpEL expression to use as a cache key.	String	

gemfire.pool

Property Name	Description	Type	Defaults
connect-type	Specifies connection type: 'server' or 'locator'.	ConnectType	
host-addresses	Specifies one or more GemFire locator or server addresses formatted as [host]:[port].	InetSocketAddress[]	
subscription-enabled	Set to true to enable subscriptions for the client pool. Required to sync updates to the client cache.	Boolean	false

gemfire.region

Property Name	Description	Type	Defaults
region-name	The region name.	String	

gemfire.security

Property Name	Description	Type	Defaults
password	The cache password.	String	
username	The cache username.	String	

gemfire.security.ssl

Property Name	Description	Type	Defaults
ciphers	Configures the SSL ciphers used for secure Socket connections as an array of valid cipher names.	String	any

keystore-type	Identifies the type of Keystore used for SSL communications (e.g. JKS, PKCS11, etc.).	String	JKS
keystore-uri	Location of the pre-created Keystore URI to be used for connecting to the GemFire cluster.	Resource	
ssl-keystore-password	Password for accessing the keys truststore.	String	
ssl-truststore-password	Password for accessing the trust store.	String	
truststore-type	Identifies the type of truststore used for SSL communications (e.g. JKS, PKCS11, etc.).	String	JKS
truststore-uri	Location of the pre-created truststore URI to be used for connecting to the GemFire cluster.	Resource	
user-home-directory	Local directory to cache the truststore and keystore files downloaded from the truststoreUri and keystoreUri locations.	String	user.home

GemFire Source Kafka

There are two methods to deploy the artifacts into the Spring Cloud Dataflow Server: 1. Using the images from Docker Hub 2. Download the artifacts from the Pivotal Commercial Maven repository.

Choose one of the following options.

Docker Hub Images

1. Log in to your Spring Cloud Dataflow Server.
2. Add the Application using the **Add Application** button.
3. Select the first option **Register one or more applications** and enter the following information:
 - **Name:** `gemfire` or `gemfire-source-kafka`.
 - **Type:** `source`.
 - **Spring Boot version:** `Spring Boot 2.x`.
 - **URI:** `docker://docker.io/gemfire/gemfire-source-kafka:1.0.0`
4. Click **Import Application** to import the GemFire Source for kafka Stream Application.

Commercial Maven Repository Artifacts

1. In a browser, navigate to the [Pivotal Commercial Maven Repository](#).
2. Click the **Create Account** link.
3. Complete the information in the registration page.
4. Click **Register**.
5. After registering, you will receive a confirmation email. Follow the instruction in this email to activate your account.
6. Once you have a Pivotal Commercial Maven Repository account created, you can now download the GemFire Source Kafka artifact (replace `{commercialMavenRepoUsername}` with your commercial maven repo username/email):

```
wget https://commercial-repo.pivotal.io/data3/gemfire-release-repo/gemfire/com/
vmware/gemfire/spring/cloud/stream/app/gemfire-source-kafka/1.0.0/gemfire-sourc
e-kafka-1.0.0.jar --user {commercialMavenRepoUsername} --ask-password
```

7. Log in to your Spring Cloud Dataflow Server.
8. Add the Application using the **Add Application** button.

9. Select the third option, **Import application coordinates from properties file**.
10. Add the `source` properties, replacing `{artifactFileName}` with the location of the downloaded artifacts from step 2. You can change the application name (`gemfire`) as needed.

```
source.gemfire=file://{artifactFileName}
```

11. Click **Import Application** to import the GemFire Source for Kafka Stream Application. It may take a minute for this application to appear in the SCDF Applications list.

Properties:

gemfire.client

Property Name	Description	Type	Defaults
pdx-read-serialized	Deserialize the GemFire objects into PdxInstance instead of the domain class.	Boolean	false

gemfire.pool

Property Name	Description	Type	Defaults
connect-type	Specifies connection type: 'server' or 'locator'.	ConnectType	
host-addresses	Specifies one or more GemFire locator or server addresses formatted as [host]:[port].	InetSocketAddress[]	
subscription-enabled	Set to true to enable subscriptions for the client pool. Required to sync updates to the client cache.	Boolean	false

gemfire.region

Property Name	Description	Type	Defaults
region-name	The region name.	String	

gemfire.security

Property Name	Description	Type	Defaults
password	The cache password.	String	
username	The cache username.	String	

gemfire.security.ssl

Property Name	Description	Type	Defaults
---------------	-------------	------	----------

ciphers	Configures the SSL ciphers used for secure Socket connections as an array of valid cipher names.	String	any
keystore-type	Identifies the type of Keystore used for SSL communications (e.g. JKS, PKCS11, etc.).	String	JKS
keystore-uri	Location of the pre-created Keystore URI to be used for connecting to the GemFire cluster.	Resource	
ssl-keystore-password	Password for accessing the keys truststore.	String	
ssl-truststore-password	Password for accessing the trust store.	String	
truststore-type	Identifies the type of truststore used for SSL communications (e.g. JKS, PKCS11, etc.).	String	JKS
truststore-uri	Location of the pre-created truststore URI to be used for connecting to the GemFire cluster.	Resource	
user-home-directory	Local directory to cache the truststore and keystore files downloaded from the truststoreUri and keystoreUri locations.	String	user.home

gemfire.supplier

Property Name	Description	Type	Defaults
event-expression	SpEL expression to extract data from an {@link org.apache.geode.cache.EntryEvent} or {@link org.apache.geode.cache.query.CqEvent}.	Expression	
query	An OQL query. This will enable continuous query if provided.	String	