Cloudera Runtime 7.3.1

Configuring Streams Messaging Manager

Date published: 2023-09-25
Date modified: 2024-12-10

https://docs.cloudera.com/

Legal Notice

The documentation is and contains Cloudera proprietary information protected by copyright and other intellectual property rights. No license under copyright or any other intellectual property right is granted herein.

Unless otherwise noted, scripts and sample code are licensed under the Apache License, Version 2.0.

Copyright information for Cloudera software may be found within the documentation accompanying each component in a particular release.

Cloudera software includes software from various open source or other third party projects, and may be released under the Apache Software License 2.0 (“ASLv2”), the Affero General Public License version 3 (AGPLv3), or other license terms. Other software included may be released under the terms of alternative open source licenses. Please review the license and notice files accompanying the software for additional licensing information.

Please visit the Cloudera software product page for more information on Cloudera software. For more information on Cloudera support services, please visit either the Support or Sales page. Feel free to contact us directly to discuss your specific needs.

Cloudera reserves the right to change any products at any time, and without notice. Cloudera assumes no responsibility nor liability arising from the use of products, except as expressly agreed to in writing by Cloudera.

Cloudera, Cloudera Altus, HUE, Impala, Cloudera Impala, and other Cloudera marks are registered or unregistered trademarks in the United States and other countries. All other trademarks are the property of their respective owners.

Disclaimer: EXCEPT AS EXPRESSLY PROVIDED IN A WRITTEN AGREEMENT WITH CLOUDERA, CLOUDERA DOES NOT MAKE NOR GIVE ANY REPRESENTATION, WARRANTY, NOR COVENANT OF ANY KIND, WHETHER EXPRESS OR IMPLIED, IN CONNECTION WITH CLOUDERA TECHNOLOGY OR RELATED SUPPORT PROVIDED IN CONNECTION THEREWITH. CLOUDERA DOES NOT WARRANT THAT CLOUDERA PRODUCTS NOR SOFTWARE WILL OPERATE UNINTERRUPTED NOR THAT IT WILL BE FREE FROM DEFECTS NOR ERRORS, THAT IT WILL PROTECT YOUR DATA FROM LOSS, CORRUPTION NOR UNAVAILABILITY, NOR THAT IT WILL MEET ALL OF CUSTOMER’S BUSINESS REQUIREMENTS. WITHOUT LIMITING THE FOREGOING, AND TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, CLOUDERA EXPRESSLY DISCLAIMS ANY AND ALL IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, QUALITY, NON-INFRINGEMENT, TITLE, AND FITNESS FOR A PARTICULAR PURPOSE AND ANY REPRESENTATION, WARRANTY, OR COVENANT BASED ON COURSE OF DEALING OR USAGE IN TRADE.

Contents

Installing Streams Messaging Manager on an existing CDP Private Cloud Base cluster

Learn how to install Streams Messaging Manager (SMM) on an already running CDP Private Cloud Base cluster.

Overview

The following sections guide you through the process of installing SMM on an already running cluster. Installing SMM involves adding the service to the cluster and enabling optional integration between SMM and other services like Schema Registry, Streams Replication Manager (SRM) and Cruise Control, or service roles like Kafka Connect.

Integration with other services

SMM is capable of integrating with other services. Enabling integration enables additional features and functions on the SMM UI. For example, enabling integration with SRM enables the **Cluster Replications** section in SMM, which you can use to monitor SRM replications.

If other services that SMM integrates with are located on the same cluster as SMM, enabling the integration happens when you add SMM to your cluster. However, SMM can integrate with some services, like SRM, even if the service is located on a remote cluster. In this case, the integration between SMM and the remote service is configured after you add SMM to your cluster. Additionally, the integration with other services can be enabled, disabled, or further configured separately following installation. As a result, the service integration steps are also documented as separate tasks.

Deployment recommendations

The SMM service is comprised of two service roles, which are as follows:

- Streams Messaging Manager Rest Admin Server: This role is responsible for processing Rest API requests from the SMM UI and collecting metrics information from Cloudera Manager and Kafka.
- Streams Messaging Manager UI Server: This role provides the SMM UI. You use the UI to manage and monitor Kafka and related services.

Cloudera recommends that you add the SMM service to a management host on your cluster. Cloudera does not recommend adding the SMM service to hosts where Kafka broker roles are running. Each cluster can only have a single instance of the SMM service deployed on it. In addition, the two SMM roles must be installed on the same host.

Cloudera Manager and Cloudera Service Monitor requirements

SMM requires high levels of Cloudera Manager and Service Monitor Service (SMON) memory usage. Ensure that both are tuned according to Cloudera recommendations.

Prometheus

By default SMM uses Cloudera Manager as its metrics store. This means that the metrics that are available for monitoring on the SMM UI are sourced from Cloudera Manager. If required, you can create a deployment where instead of Cloudera Manager, SMM uses Prometheus as its metrics store. Prometheus is an open-source systems monitoring and alerting toolkit that stores its metrics as time series data.

Prometheus supports a significantly larger number of time series entities compared to Cloudera Manager. As a result, Cloudera recommends that you use Prometheus instead of Cloudera Manager if you need to monitor a large number of Kafka entities (topics, partitions and so on). Additionally, using Prometheus enables you to configure the roll up policy, delete specific time series entities, and configure scrape intervals and metrics retention periods. Prometheus setup for SMM is done after adding SMM service to your cluster.

Related Information

Cloudera Manager hardware requirements

Setting up Prometheus for Streams Messaging Manager

Adding the Streams Messaging Manager service to an existing cluster

Learn how to add the Streams Messaging Manager (SMM) service to an already running cluster.

About this task

The following list of steps walk you through how you can add SMM on an already existing CDP Private Cloud Base Cluster. The following steps are aimed to provide you with the basic process of installation and do not go into detail how you can set up or configure security for the service.

Tip: Although the following steps explicitly walk you through installation on an existing cluster using the **Add a service** wizard, you can also use the following steps as reference when installing a new cluster that will include SMM.

Before you begin

- Ensure that a Kafka service is installed on your cluster. Additionally, ensure that the Enable Producer Metrics Kafka service property is selected. This property is enabled by default.
- SMM requires a backend database to function. The database is used to store Kafka metadata, metrics, and alert definitions. As a result, a database and database user must be set up for SMM before adding the service. The following SQL snippet gives an example of the setup that you are required to do.

  ```
  create database streamsmsgmgr;
  CREATE USER 'streamsmsgmgr'@'localhost' IDENTIFIED BY 'streamsmsgmgr';
  GRANT ALL PRIVILEGES ON streamsmsgmgr.* TO 'streamsmsgmgr'@'localhost' WIT
  H GRANT OPTION;
  CREATE USER 'streamsmsgmgr'@'%' IDENTIFIED BY 'streamsmsgmgr';
  GRANT ALL PRIVILEGES ON streamsmsgmgr.* TO 'streamsmsgmgr'@'%' WITH GRANT
   OPTION;
  ```

  For more information on database setup as well as supported databases. Review the following resources:

  - Install and Configure Databases
  - Configuring Databases for Streaming Components
  - Cloudera Support Matrix

  Note: SMM supports the databases and database versions that are supported by the version CDP Private Cloud Base you are using. This means that any database version that is highlighted as supported for CDP Private Cloud Base in the Cloudera Support Matrix is also supported by SMM.
- SMM is capable of establishing a secure (encrypted) connection with its database if the database has TLS 1.2/TCPS enabled. This is done by specifying the database connection using a JDBC URL. If you plan on using an encrypted connection, ensure that you have a valid JDBC URL on hand and have completed additional prerequisites. For more information and example JDBC URLs, see Database setup details for Streams Messaging Manager for TLS 1.2/TCPS-enabled databases.

Procedure

1. In Cloudera Manager, select the cluster that you want to install SMM on.
2. Click Actions Add Service .
3. Select Streams Messaging Manager from the list of services and click Continue.

4. Select service dependencies.

   The options on the **Select Dependencies** page differ depending on what other services are available on your cluster. Review the following and select the dependencies based on your requirements.

   Schema Registry

   If selected, integration between SMM and Schema Registry will be enabled. Schema Registry related features become available on the SMM UI. This is a recommended optional dependency.

   Streams Replication Manager

   If selected, integration between SMM and SRM will be enabled. SRM integration enables the **Cluster Replications** section on the UI, which you use to monitor replications. This is a recommended optional dependency.

   Kafka

   SMM requires a Kafka service as a dependency. Kafka services will only be visible on this page if there are multiple Kafka services available on the cluster. Otherwise, if only a single Kafka is available, SMM automatically selects it as a dependency as Kafka is a mandatory dependency.

   Note: If there are no optional dependencies available, Cloudera Manager automatically selects the required service dependencies and skips this step.

5. Click Continue.
6. Assign role instances to hosts.

   On the **Assign Roles** page you select which host you want to install SMM roles on. Because both SMM roles must be installed on the same host, you can only select a host for the Rest Admin Server role. The UI Server role is installed on the same host.

   A host is selected by default. If required you can change the host by doing the following:

   a) Click the field below **Streams Messaging Manager Rest Admin Server** to display a dialog containing a list of hosts.
   b) Select a single host and click Ok.

7. Click Continue.
8. Set up the database connection.

   The database that you specify here is the one that you already set up for SMM prior to installation.

   a) Select the database type and enter the database name, username, and password.
   b) Click Test Connection.

      If the connection test fails, review your configuration, fix any errors, and rerun the connection test. You can only continue if the validation check is successful.

   Tip: If your database has TLS 1.2/TCPS enabled and you want SMM to connect to the database using an encrypted connection, select yes from the Use JDBC URL Override drop-down list and enter a valid JDBC URL.

9. Click Continue.
10. Review changes.

    The **Review Changes** pages lists default and suggested settings for several configuration properties. Review and if required configure these properties.

11. Click Finish.

Results

The SMM service is added to your cluster, however, it is not yet started.

What to do next

Depending on your requirements, you have multiple options on how to continue. Choose one of the following.

- Start the SMM service. To do this, click Actions Start on the homepage of the SMM service.
- Learn more about or enable integration with other services and service roles. Continue with service specific integration steps.
- Set up Prometheus as the metrics store. Continue with *Setting up Prometheus for Streams Messaging Manager*.

Related Information

CDP Private Cloud Base Installation Guide

Setting up Prometheus for Streams Messaging Manager

Integrating Schema Registry with Streams Messaging Manager

Learn how to configure Streams Messaging Manager (SMM) to connect and integrate with Schema Registry. Enabling integration between the two services enables Schema Registry related functionality within the SMM UI.

About this task

SMM is capable of connecting to and integrating with a Schema Registry service. Enabling integration makes it possible for the SMM UI Data Explorer to use schemas stored in Schema Registry to deserialize Kafka messages. Additionally, the Data Explorer will include a quick access link to the schema associated with the topic you are viewing.

Before you begin

- In most cases Schema Registry integration is enabled when you add the SMM service to the cluster. As a result, it is highly likely that integration is already enabled. If integration is enabled, you do not need to complete the following steps.
- SMM can only integrate with a Schema Registry service if the service is co-located with SMM. Specifying remote Schema Registry services is not possible.

Procedure

1. In Cloudera Manager, select the SMM service.
2. Go to Configuration.
3. Find and select the Schema Registry Service property.
4. Click Save Changes.
5. Restart SMM.

Results

Integration between SMM and Schema Registry is enabled. Schema Registry related functionality is now available in the Data Explorer of the SMM UI.

Integrating Streams Replication Manager with Streams Messaging Manager

Learn how to configure Streams Messaging Manager (SMM) to connect and integrate with Streams Replication Manager (SRM). Enabling integration between the two services enables SRM related functionality within the SMM UI.

About this task

SMM is capable of integrating with SRM. Enabling integration between the two services enables the Cluster Replications section within the SMM UI. You can use this section of the UI to monitor the replication flows you create with SRM.

Service integration is enabled in either of two ways. If SRM is located on the same cluster as SMM, you integrate the two services by enabling a service dependency. If the two services are located remote to each other (on different clusters), you need to manually specify the host, port, and protocol of one or multiple SRM Service role instances that SMM should interface with.

Before you begin

- Ensure that a co-located or remote SRM service is available.
- If SRM is co-located with SMM, it is likely that integration between the two services was enabled when you added SMM to your cluster. Check if the Cluster Replications section is available in the SMM UI. If it is, you do not need to complete the following steps.

Procedure

1. In Cloudera Manager, select the SMM service.
2. Go to Configuration.
3. Enable integration between the two services.

   How you complete this step depends on whether SRM is colocated with SMM. Use a service dependency if the two services are colocated. Otherwise, specify the connection using dedicated properties.

   For Using a service dependency

   a. Find and select the STREAMS_REPLICATION_MANAGER Service property.

   For Using dedicated properties

   a. Find and select the Configure Streams Replication Manager Manually property.
   b. Find and configure the Streams Replication Manager Rest Protocol property.

      Select http or https depending on configuration of SRM. If Enable TLS/SSL for SRM Service is selected, select https, otherwise, select http.
   c. Find and configure the Streams Replication Manager Rest HostPorts property.

      Enter the host and port pairs of one or multiple the SRM Service roles. You can find the hostname by going to SRM Instances . For the port, use the value set in the SRM Service Https Port or SRM Service Port properties. If the SRM Service roles are TLS/SSL enabled (Enable TLS/SSL for SRM Service is selected), use the port in SRM Service Https Port. Otherwise, use the port in SRM Service Port.
4. Click Save Changes.
5. Restart SMM.

Results

Integration between SMM and SRM is enabled. The Cluster Replications tab is now available on the SMM UI.

What to do next

Go to the SMM UI and access the Cluster Replications section from the navigation sidebar.

Related Information

Monitoring Streams Replication Manager

Integrating Kafka Connect with Streams Messaging Manager

Learn how to set up Streams Messaging Manager (SMM) for monitoring and managing Kafka Connect.

About this task

Cloudera recommends that you use SMM to manage and monitor Kafka Connect. In order for SMM to be able to interact with Kafka Connect, it must be configured and provided with the host, port, and protocol of the Kafka Connect role instances. This is done by configuring the Kafka Connect Rest HostPort and Kafka Connect Protocol SMM configuration properties. Configuring these properties enables the **Connect** section in SMM, which allows you to interact with Kafka Connect through a UI interface. Additionally, configuring these properties also enables you to use the SMM REST API to interact with Kafka Connect.

If the Kafka Connect Rest HostPort property is left empty (default), SMM is automatically configured with the host, port, and protocol of all Kafka Connect role instances belonging to the Kafka service selected with the Kafka Service SMM property. This means that if you want SMM to monitor and manage the Kafka Connect instance that belongs to the Kafka service that SMM depends on, you do not need to complete the following steps.

Procedure

1. Select the Streams Messaging Manager service.
2. Go to Configuration.
3. Find and configure the following properties.
   - Kafka Connect Rest HostPort

     Enter the host and port pairs of the Kafka Connect roles. If you have multiple instances of the Kafka Connect role, Cloudera recommends that you add all of them for high availability. If there are multiple host and port pairs specified, SMM will automatically failover between the specified endpoints. For the port, use the value set in the Kafka Connect Rest Port or Secure Kafka Connect Rest Port Kafka Connect properties. If the Kafka Connect role is not TLS/SSL enabled, use the port specified in Kafka Connect Rest Port. If TLS/SSL is enabled, use the port specified in Secure Kafka Connect Rest Port.
   - Kafka Connect Protocol

     Select http if SSL/TLS is not enabled for the Kafka Connect roles. Select https if SSL/TLS is enabled for the Kafka Connect roles.
4. Click Save.
5. Restart the service.

Results

SMM is configured and is able to manage and interact with Kafka Connect.

What to do next

Go to the SMM UI and access the **Connect** section from the left column. Alternatively, use the SMM REST API to interact with Kafka Connect.

Related Information

Monitoring Kafka Connect

Authorizing users to access Cruise Control in SMM

Learn how to authorize users to access the Cruise Control UI in Streams Messaging Manager (SMM). This is required to enable users access the UI or manage rebalancing. You can set access by assigning users the required authorization levels.

About this task

The integration between SMM and Cruise Control is automatic. The Cruise Control User Interface (UI) can be accessed through an SMM instance running on the same cluster as Cruise Control. In environments where security is disabled, anyone will be able to access Cruise Control in SMM. In secured environments, users need to be added to the authorization level groups before they can see the Cruise Control UI. The group a user belongs to determines what they are allowed to do in the UI.

Procedure

1. Select the Cruise Control service in Cloudera Manager.
2. Click Configuration.
3. Search for auth_admins, auth_users or auth_viewers based on which authorization level is needed.
4. Assign users to the required authorization level.

   Based on the authorization level, users have the following access to Cruise Control in SMM:

   - Viewer level users: can only view the metrics
   - User level users: includes Viewer level permissions and manage goals, estimations
   - Admin level users: includes Viewer and User level permissions, starting and stopping rebalance

   Note: The assigned authorization level for a user can be checked with the permissions endpoint:

   ```
   curl -X GET "http://<cruise_control_hostname>:8899/kafkacruisecontro
   l/kafkacruisecontrol/permissions”
   ```
5. Click Save changes.

Results

Cruise Control is visible in SMM. Users can access the Cruise Control UI using the 🖥 on the navigation sidebar.

Setting up Prometheus for Streams Messaging Manager

Get started with setting up and using Prometheus for Streams Messaging Manager (SMM). You can use Prometheus as the metrics store for SMM. Its use is recommended over Cloudera Manager if your deployment includes a significantly large number Kafka entities that you want to monitor.

Cloudera Manager metrics overview

A metric is a property that can be measured to quantify the state of an entity or activity. They include properties such as the number of open file descriptors or CPU utilization percentage across your cluster.

Cloudera Manager monitors a number of performance metrics for services and role instances running on your clusters. These metrics are monitored against configurable thresholds and can be used to indicate whether a host is functioning as expected or not. You can view these metrics in the Cloudera Manager Admin Console.

Cloudera agents collect the metrics from individual brokers and report them to Cloudera Manager once in a minute (through Heartbeat). The collected metrics are stored in Cloudera Manager's database to query or search for historical data.

Cloudera Manager provides a default metric store for SMM. SMM fetches the required metrics from Cloudera Manager whenever required and caches the subset of metrics in the SMM server for which the historical values appear in the SMM UI. The cache refresh is configurable, the default is 50 seconds.

Prometheus overview

Prometheus is a metrics store that pulls metrics from different endpoints which you configure. Prometheus is not the default metric store for SMM. If you want to use Prometheus as the metrics store for SMM, you need to download and configure it. Prometheus supports a larger number of time series entities compared to the Cloudera Manager metrics store.

If you use Prometheus, you can configure the roll up policy, delete specific time series entities, and configure scrape interval and metrics retention period.

For SMM, you need to configure the following endpoints for Prometheus to pull metrics from:

- Kafka

  Kafka exposes a Prometheus metrics endpoint for Kafka metrics to be pulled.
- Kafka Connect

  Kafka Connect, through configuration, exposes a Prometheus endpoint for Kafka connect metrics to be pulled.
- Prometheus Node Exporter

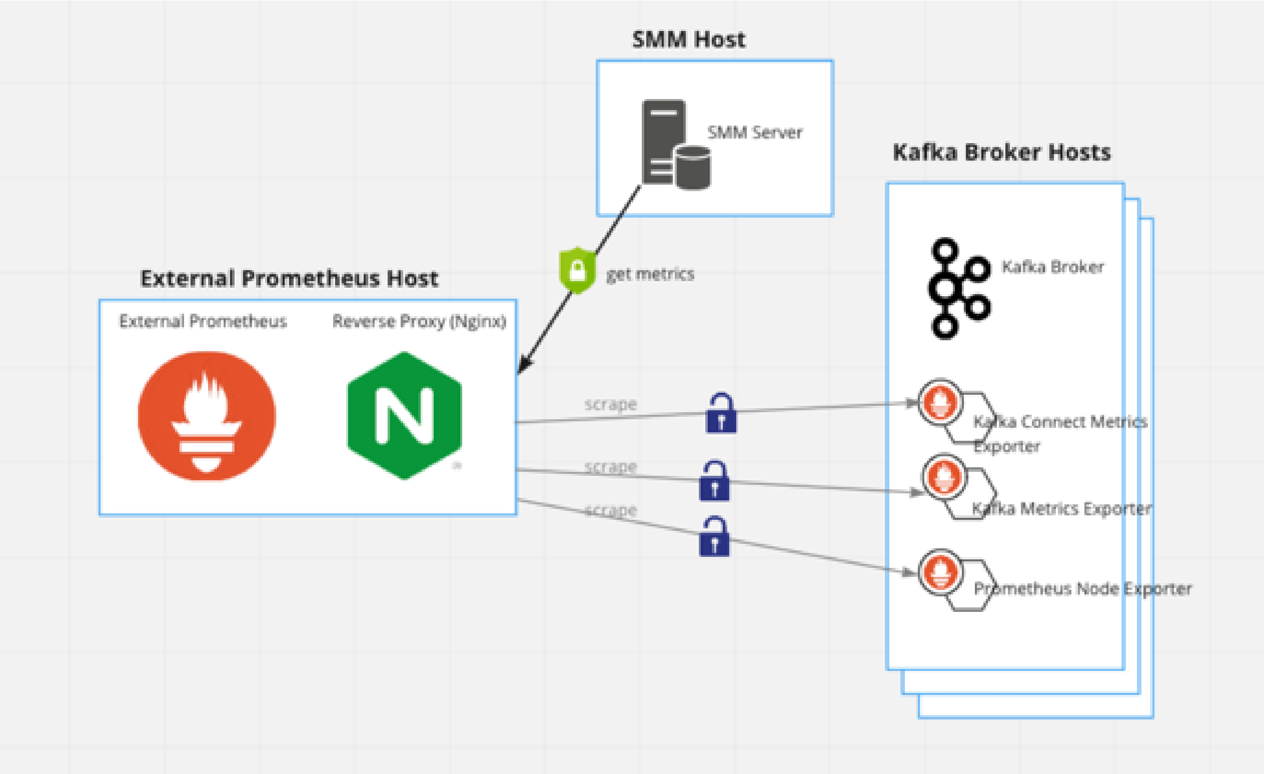
  You need to configure a separate Node Exporter on each Kafka broker host and enable Prometheus to pull the system metrics.

SMM queries Prometheus for metrics over time. Prometheus fetches the metrics from the endpoints.

Prometheus relies on external tools for security. For example, you can secure your Prometheus with Nginx in the following scenarios:

- TLS
- TLS with basic authentication
- mTLS

The following image shows the architecture of Prometheus configured for SMM and secured with Nginx:



Prometheus configuration for SMM

Prometheus is not the default metric store for Streams Messaging Manager (SMM). If you want to use Prometheus as the metric store for SMM, you need to download and configure it.

Prerequisites for Prometheus configuration

Learn the prerequisites before you configure Prometheus for Streams Messaging Manager (SMM).

- You must download Prometheus and install Prometheus and Prometheus node exporters for each Kafka broker. SMM requires system-level metrics for each Kafka broker and therefore, Prometheus node exporter must be configured for each Kafka broker. Cloudera only supports Linux node exporters.
- Cloudera recommends using a dedicated Prometheus instance for SMM because other services querying Prometheus could use the capacity causing SMM query to timeout.

Prometheus properties configuration

Learn the properties that you need to configure in the prometheus.yml file before you start using the Prometheus metric store for Streams Messaging Manager (SMM).

Configure the following properties in the prometheus.yml file:

- Set the scrape_interval property value to 60 seconds in the prometheus.yml file.

```
scrape_interval: 60s
```

- Prometheus automatically assigns the instance label to metrics. However, in case partition leader change reassignment happens, it means an excessive amount of metrics being created. This property is recommended for large clusters.

```
    metric_relabel_configs:
    - source_labels: [__name__]
      regex: ^(broker_producer_messagesinpersec_total|topic_partition_me
ssagesinpersec_total|topic_partition_bytesinpersec_total|topic_partition
_bytesoutpersec_total)$
      target_label: instance
      replacement: 'no-instance'
```

- Set Kafka host and metrics port values.

```
['luigi-1.luigi.root.hwx.site:24042','luigi-2.luigi.root.hwx.site:24042'
,'luigi-3.luigi.root.hwx.site:24042']
```

  You can find the metric port used by Kafka in Cloudera Manager Kafka service Configuration HTTP Metric Report Port .
- Set Kafka Connect host (deployed on same hosts as Kafka) and the metrics port values.

```
['luigi-1.luigi.root.hwx.site:28087','luigi-1.luigi.root.hwx.site:28087'
,'luigi-1.luigi.root.hwx.site:28087']
```

  The metrics port used by Kafka Connect is set in Cloudera Manager Kafka service Configuration Secure Jetty Metrics Port and Jetty Metrics Port. Use Secure Jetty Metrics Port if the Kafka Connect metrics reporter uses TLS/SSL, otherwise, use Jetty Metrics Port. The value of 28087 in this example is the default secure port.
- Set Prometheus node exporter host (deployed on same hosts as Kafka) and the Prometheus node exporter metrics port values.

```
['luigi-1.luigi.root.hwx.site:9100','luigi-2.luigi.root.hwx.site:9100','
luigi-3.luigi.root.hwx.site:9100']
```

- update=true parameter should be only used by Prometheus. Querying Kafka Prometheus endpoint with this flag set to true updates the internal metrics cache for stateful metrics.

```
update: ['true']
```

- If Basic Authentication is enabled for the Kafka Connect metrics reporter, add the following parameters:

```
basic_auth:
  username: 'email@username.me'
```

```
   password: 'password'
```

Ensure that you specify the username and password that is configured for the Kafka Connect metrics reporter. You can find the username and password in Cloudera Manager Kafka service Configuration Jetty Metrics User Name and Jetty Metrics Password.

Configuration example

The following is an example of a Prometheus configuration YAML. You can use this example as a template and make changes as necessary. Ensure that you replace host and port values as well as the Basic Authentication credentials with your own values. This example uses the default ports.

```
# my global config
global:
  scrape_interval: 60s
  scrape_timeout: 55s

# Alertmanager configuration
alerting:
  alertmanagers:
  - static_configs:
    - targets:
      # - alertmanager:9093

# Load rules once and periodically evaluate them according to the global '
evaluation_interval'.
rule_files:
  # - "first_rules.yml"
  # - "second_rules.yml"

# A scrape configuration containing exactly one endpoint to scrape:
# Here it's Prometheus itself.
scrape_configs:
  # The job name is added as a label `job=<job_name>` to any timeseries s
craped from this config.
  - job_name: 'prometheus'
    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.
    static_configs:
    - targets: ['localhost:9090']

  - job_name: 'kafka'

    metrics_path: '/api/prometheus-metrics'

    params:
      update: ['true']

    static_configs:
    - targets: ['luigi-1.luigi.root.hwx.site:24042','luigi-2.luigi.root.hwx.
site:24042','luigi-3.luigi.root.hwx.site:24042']
    metric_relabel_configs:
    - source_labels: [__name__]
      regex: ^(broker_producer_messagesinpersec_total|topic_partition_messa
gesinpersec_total|topic_partition_bytesinpersec_total|topic_partition_byteso
utpersec_total)$
      target_label: instance
      replacement: 'no-instance'

  - job_name: 'kafka_connect'

    metrics_path: '/prometheus-metrics'
    basic_auth:
      username: 'email@username.me'
      password: 'password'

    static_configs:
    - targets: ['luigi-1.luigi.root.hwx.site:28087','luigi-1.luigi.root.hwx.
site:28087','luigi-1.luigi.root.hwx.site:28087']

  - job_name: 'system_metrics'

    # metrics_path defaults to '/metrics'
    # scheme defaults to 'http'.

    static_configs:
    - targets: ['luigi-1.luigi.root.hwx.site:9100','luigi-2.luigi.root.hwx
.site:9100','luigi-3.luigi.root.hwx.site:9100']
```

Where:

- ['luigi-1.luigi.root.hwx.site:9100','luigi-2.luigi.root.hwx.site:9100','luigi-3.luigi.root.hwx.site:9100'] = Kafka host + Node exporter metrics port
- ['luigi-1.luigi.root.hwx.site:24042','luigi-2.luigi.root.hwx.site:24042','luigi-3.luigi.root.hwx.site:24042'] = Kafka host + HTTP Metric Report Port
- ['luigi-1.luigi.root.hwx.site:28087','luigi-1.luigi.root.hwx.site:28087','luigi-1.luigi.root.hwx.site:28087'] = Kafka Connect host + Secure Jetty Metrics Port

SMM property configuration in Cloudera Manager for Prometheus

Learn about the Streams Messaging Manager (SMM) properties that you need to configure in Cloudera Manager before you start using the Prometheus metric store.

Configure the following SMM properties in Cloudera Manager:

- metrics.fetcher.class

  Configures SMM to fetch metrics from Prometheus. Set it to com.hortonworks.smm.kafka.services.metric.prometheus.PrometheusMetricsFetcher.
- prometheus.metrics.url

  Prometheus metrics URL should be configured here in the format of scheme://prometheus_host:prometheus_port. If HTTPS is configured for Prometheus, HTTPS should be specified as the scheme.
- prometheus.metrics.user

  Should be set if Prometheus is configured for TLS with basic authentication.
- prometheus.metrics.password

  Should be set if Prometheus is configured for TLS with basic authentication.

Kafka property configuration in Cloudera Manager for Prometheus

Learn about the Kafka properties that you need to configure in Cloudera Manager before you start using the Prometheus metric store.

Configure the following Kafka property in Cloudera Manager:

- kafka.http.metrics.port

  Used for exposing Cloudera Manager metrics. This port is now also shared to expose Prometheus metrics for Kafka.

Kafka Connect property configuration in Cloudera Manager for Prometheus

Learn about the Kafka Connect properties that you need to configure in Cloudera Manager before you start using the Prometheus metric store.

In order to set up Prometheus as the SMM metrics store, you need to configure the metrics reporter of Kafka Connect. Configuration is done in Cloudera Manager by setting Kafka Connect properties. Configuration differs depending on whether you want to enable security for the Kafka Connect metrics reporter. All of the following properties are found in Cloudera Manager Kafka service Configuration . Completing the following makes Prometheus-compatible metrics available on the /prometheus-metrics API path on each Kafka Connect worker host.

Do the following to configure an unsecured metrics-scraping endpoint:

1. Add metrics.jetty.server.prometheus.metrics.enable=true to the Kafka Connect Advanced Configuration Snippet (Safety Valve) for connect-distributed.properties advanced configuration snippet.
2. Look up or configure the value of Jetty Metrics Port.

   This is the port where Kafka Connect exposes its metrics when TLS/SSL is not enabled. This is the port that you need to add to prometheus.yml. Only configure this property if you want to change the port.

Do the following to configure a secure metrics scraping endpoint:

1. Add metrics.jetty.server.prometheus.metrics.enable=true to the Kafka Connect Advanced Configuration Snippet (Safety Valve) for connect-distributed.properties advanced configuration snippet.
2. Enable TLS/SSL, Basic Authentication (BA), or both for the metrics reporter by configuring the following properties:

   - Enable TLS/SSL for Kafka Connect
   - Enable Basic Authentication for Metrics Reporter
   - Jetty Metrics User Name
   - Jetty Metrics User Password

   Note the following about these properties:

   - The Enable TLS/SSL for Kafka Connect property is not specific to the metrics reporter. It enables TLS/SSL for Kafka Connect roles including their metrics reporter.
   - You can enable both TLS/SSL and BA on their own, however, Cloudera recommends that you enable both.
   - Jetty Metrics User Name and Jetty Metrics User Password set the username and password that you need to add to prometheus.yml if you enable BA.
3. Look up or configure the value of Secure Jetty Metrics Port or Jetty Metrics Port

   These are the ports where Kafka Connect exposes its metrics. Secure Jetty Metrics Port is only used if TLS/SSL is enabled, otherwise Kafka Connect uses Jetty Metrics Port. You need to add the port being used to prometheus.yml. Only configure this property if you want to change the port.

Start Prometheus

You should start Prometheus with the startup flag options that you need.

For example,

```
./prometheus --storage.tsdb.retention.time 30d --web.enable-admin-api --quer
y.max-samples 5000000000
```

- --storage.tsdb.retention.time 30d

  Prometheus defaults to 15 days. SMM monitors metrics up to 30 days. Therefore, this flag must be configured.
- --web.enable-admin-api

  Optional. If Admin APIs are needed, such as deleting time series entities, this flag should be set.

- --query.max-samples

  Optional. When many entities or samples exist, bulk query might reach the maximum sample limit.

Secure Prometheus for SMM

Streams Messaging Manager (SMM) supports connecting to Prometheus servers behind a TLS proxy (for example, Nginx).

You can connect to Prometheus servers behind a TLS proxy in the following scenarios:

- TLS

  SMM verifies the proxy’s certificate.
- TLS with basic authentication

  SMM verifies the proxy’s certificate and authenticates itself with username and password.
- mTLS

  SMM verifies the proxy’s certificate and authenticates itself with its TLS Certificate (TLS proxy should recognize it).

Nginx proxy configuration over Prometheus

You need to install and configure Nginx before using it to configure proxy over Prometheus.

Nginx installtion

To use Nginx, you need to install it.

For information about how to install Nginx, see Nginx documentation.

Nginx configuration for Prometheus

Prometheus does not, by default, support TLS encryption for connections to Prometheus instances. If you want to enforce TLS encryption for the connections, you can use Prometheus in conjunction with a reverse proxy and apply TLS at the proxy layer. You can use any reverse proxy, but in this guide you see an Nginx example.

For details about securing Prometheus using TLS encryption, see Prometheus documentation.

Note: In the above linked documentation it is presumed that you want to run Ningx in the 443 port. If it is not the case (for example, it runs on 9443 port), it is not necessary to run Nginx as root. However, you must pay attention to the following things:

- Ensure that the nginx user has access to the TLS certificates.
- The web.external-url parameter of the Prometheus start command must contain the port number. For example,

  ```
  --web.external-url=https://myprometheus.com:9443/prometheus
  ```

After you configure a server for Prometheus, you may want to disable the default server in Nginx configuration. The following example shows the default server commented out:

```
#    server {
#        listen       80 default_server;
#        listen       [::]:80 default_server;
#        server_name  _;
#        root         /usr/share/nginx/html;
#
#        # Load configuration files for the default server block.
#        include /etc/nginx/default.d/*.conf;
#
#        location / {
#        }
```

```
#
#        error_page 404 /404.html;
#        location = /404.html {
#        }
#
#        error_page 500 502 503 504 /50x.html;
#        location = /50x.html {
#        }
#    }
```

Setting up TLS for Prometheus

You need to configure Streams Messaging Manager (SMM) when a TLS proxy is configured over Prometheus.

You need to configure the following:

- Ensure that Cloudera Manager or SMM recognizes Nginx’s TLS certificate. For details, see *Configure SMM to recognize Prometheus’s TLS certificate*.
- Update the Prometheus URL in SMM settings. You must update the prometheus.metrics.url property to point to the TLS proxy’s endpoint (for example, https://myprometheus.com:9443/prometheus) and restart SMM.

Related Information

Configuring SMM to recognize Prometheus’s TLS certificate

Configuring SMM to recognize Prometheus’s TLS certificate

You can configure Streams Messaging Manager (SMM) either to use its own keystore or truststore, or to use the auto-TLS feature in Cloudera Manager. Cloudera recommends using the auto-TLS feature for CDP clusters.

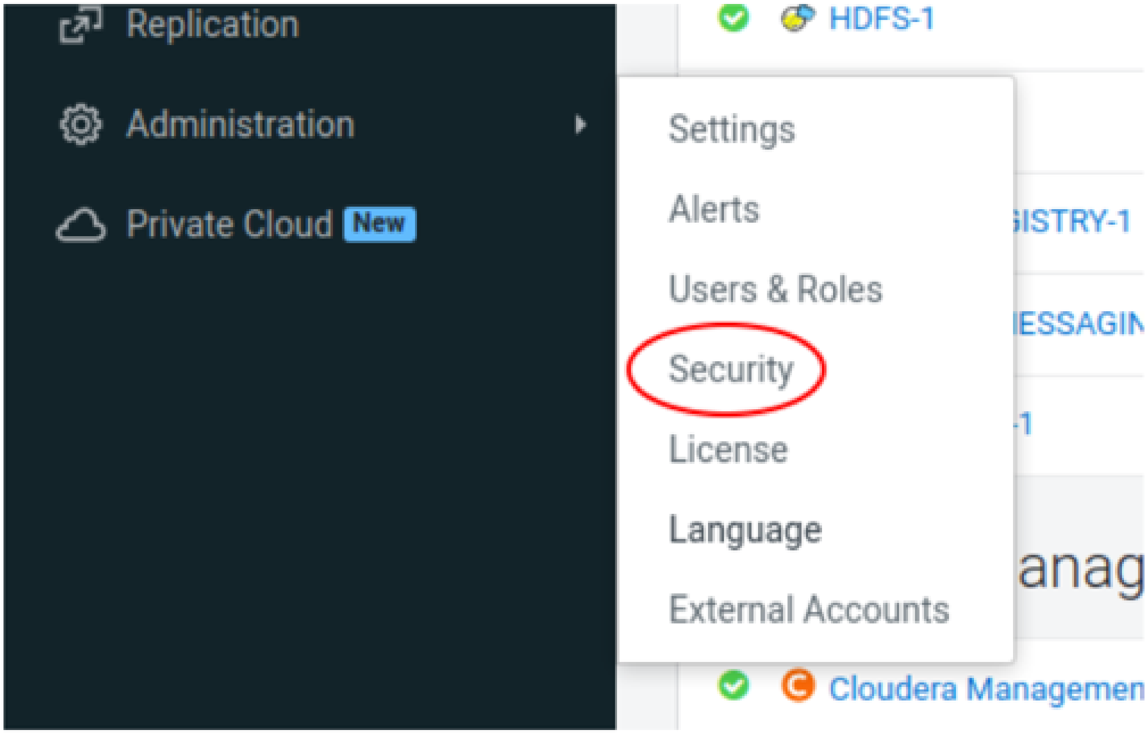
About this task

If the TLS proxy’s certificate is not recognized by SMM, it must be added to SMM truststore. The process is different for auto-TLS and the manual TLS setups.

Auto-TLS

If the TLS proxy certificate is not recognized by the cluster, you can add the TLS proxy certificate to the CA truststore of the cluster by triggering a certificate regeneration. This involves restarting the services in the cluster.

1. Go to Administration Security from the left menu bar.



2. Click Rotate Auto-TLS Certificates.
3. In the Trusted CA Certificates Location field, enter the path to the Nginx server's certificate. For example, /etc/nginx/certs/ca-certificate.pem. Ensure that the file is accessible by the cloudera-scm user.
4. Specify the authentication method with other nodes of the cluster (password or certificate).
5. Click Next and follow the instructions in the wizard.

Manual TLS

You can use the keytool command to configure the manual TLS settings.

Keytool is a tool provided by the Java Runtime Environment to manipulate JKS type keystores. You can find it in the bin folder of your JRE installation. For example, /usr/java/default/jre/bin/keytool.

1. Use the following command to add the TLS proxy's certificate to SMM's truststore:

```
keytool -import -file <TLS PROXY OR CA CERTIFICATE> -alias Nginx_for_Pro
metheus -keystore <SMM_TRUSTSTORE> -storepass <TRUSTSTORE PASSWORD>
```

   For example,

```
keytool -import -file /etc/nginx/certs/ca-certificate.pem -alias Nginx_f
or_Prometheus -keystore smm_trusstore.jks
```

   This command creates the truststore if it does not already exist.
2. Create a keystore for SMM if it does not already exist:

```
keytool -genkey -keystore smm_keystore.jks -alias smm -keyalg RSA -sigalg
 SHA256withRSA -validity 365 -keysize 3072
```

   It creates a keystore with a self-signed key.

3. Set the following SMM properties in Cloudera Manager:

   - streams.messaging.manager.ssl.keyStorePath/ssl_server_keystore_location
   - ssl_server_keystore_password
   - ssl_server_keystore_keypassword (by default it is the same as the keystore file password)
   - streams.messaging.manager.ssl.trustStorePath/ssl_client_truststore_location
   - ssl_client_truststore_password

Related Information

Setting up TLS for Prometheus

Setting up basic authentication with TLS for Prometheus

To set up TLS with basic authentication, you need to configure Nginx and Streams Messaging Manager (SMM).

Configuring Nginx for basic authentication

To configure Nginx for basic authentication, you need to create a file for user and password, update Nginx configurations, and restart Nginx.

Procedure

1. Create an user-password file for Nginx.

```
htpasswd -c /etc/nginx/.htpasswd admin
```

   This requires the Apache HTTP package to be installed on the system.

2. Update your Nginx configuration (/etc/nginx/nginx.conf or a custom configuration file in the /etc/nginx/conf.d directory) with the highlighted portion in the code below:

```
    server {
        listen              9443 ssl;
        server_name         _;
        ssl_certificate     /<PATH TO CERTIFICATE>/nginx-server-crt.pem;
        ssl_certificate_key /<PATH TO CERTIFICATE>/nginx-server-key.pem;
        location /prometheus/ {
            auth_basic           "Prometheus";
            auth_basic_user_file /etc/nginx/.htpasswd;
            proxy_pass http://localhost:9090/;
        }
    }
```

3. Restart Nginx.

Configuring SMM for basic authentication

To configure Streams Messaging Manager (SMM), you need to configure Prometheus username and password, and restart SMM.

Procedure

1. Set the following configurations:

   - Prometheus User (admin)
   - Prometheus Password (the password you gave in the htpassword tool)

2. Restart SMM.

Setting up mTLS for Prometheus

Along with or instead of basic authentication, mTLS can also be used for client authentication.

About this task

When using mTLS, both the server and the client authenticate themselves with a TLS certificate. As Streams Messaging Manager (SMM) is configured to recognize Nginx’s certificate, it needs to be configured the other way around.

Procedure

1. Export SMM’s certificate or its CA certificate.

    - In case of Auto-TLS, it is

      ```
      /var/lib/cloudera-scm-agent/agent-cert/cm-auto-global_cacerts.pem
      ```

      or

      ```
      /var/lib/cloudera-scm-agent/agent-cert/cm-auto-in_cluster_ca_cert.pem
      ```

    - In case of manual TLS, you can use keytool to export the certificate. For example,

      ```
      keytool -exportcert -rfc -keystore /opt/cloudera/smm_keystore.jks -alias
       smm -file smm.cer
      ```

2. Add the highlighted lines to Nginx server configuration (/etc/nginx/nginx.conf or a custom configuration file in the /etc/nginx/conf.d directory).

    ```
        server {
            listen              9443 ssl;
            server_name         _;
            ssl_certificate     /<PATH TO CERTIFICATE>/nginx-server-crt.pem;
            ssl_certificate_key /<PATH TO CERTIFICATE>/nginx-server-key.pem;
            ssl_client_certificate /<PATH TO SMM’S CERTIFICATE>;
            ssl_verify_client      on;

            location /prometheus/ {
                proxy_pass http://localhost:9090/;
            }
        }
    ```

3. Restart Nginx.

Prometheus for SMM limitations

Learn about the known issues and limitations, the areas of impact, and workaround while using Prometheus for Streams Messaging Manager (SMM).

- Prometheus is not managed by Cloudera Manager. You should start Prometheus right after Kafka startup because certain metrics are stateful within Kafka and might cause the initial scrape metric values for Prometheus to be high. The stateful metrics accumulate the values from several Kafka restarts until the initial scrape, and accumulate the values between consecutive scrapes.
- You need to configure the scrape interval to 60 seconds.
- SMM supports Prometheus 2.25 and above versions.
- SMM supports Linux 1.1.2 and above versions of node exporters.
- Depending on the number of entities (topics, partitions, consumers, and producers), memory requirements for Prometheus might vary.

Troubleshooting Prometheus for SMM

Troubleshooting Prometheus for SMM requires being able to diagnose and debug problems related to performance, network connectivity, out-of-memory conditions, disk space usage, and crash or non-responsive conditions.

This section provides some troubleshooting solutions for Prometheus issues for SMM.

Issue: If no metrics are available.

Solution: Examine if all configured Prometheus targets are running.

Issue: You observe incorrect metrics in SMM.

Solution: Examine up metrics to determine if scraping failed at certain points in time.

Issue: If Prometheus does not start up or scrape right after Kafka startup, or Prometheus is down for some time, scrapped metric values might be unusually high.

Solution: This situation can be avoided by querying the endpoints manually with the update=true parameter and then starting Prometheus. If you already encountered this situation, you can delete that particular time series with the delete API of Prometheus.

Performance comparison between Cloudera Manager and Prometheus

Learn the performance comparison between Cloudera Manager and Prometheus metric stores.

- Prometheus can handle more than twice the amount of TimeSeries entities than Cloudera Manager while maintaining the consistency of the metrics.
- Metrics queries for shorter time periods (30 minutes, 1 hour) take about half the time in case of Prometheus compared to Cloudera Manager.
- Metrics queries for larger time periods (2 weeks, 1 month), Prometheus seems to be about 30% slower than Cloudera Manager.
- Metric inconsistencies are not observed while testing Prometheus.