Cloudera DataFlow for Data Hub 7.2.12

# Planning Your Streams Messaging Deployment

**Date published: 2019-12-16**
**Date modified: 2021-10-25**

**https://docs.cloudera.com/**

# Legal Notice

# Contents

# Deployment scenarios

Before you get started with a Cloudera Data Flow deployment, it is useful to understand which software version is right for your platform and operational objectives. This helps you additionally understand the documentation you need to review to get started.

| If you want ... | Running with the following additional components | Review this documentation |
|---|---|---|
| Kafka for CDP Public Cloud – Streams Messaging clusters | • Schema Registry<br>• Streams Messaging Manager<br>• Streams Replication Manager<br>• Cruise Control | Cloudera Data Flow for Data Hub documentation |
| Kafka for CDP Private Cloud Base | | Streaming libraries in Cloudera Runtime CDP Private Cloud Base |
| CDK powered by Apache Kafka for CDH 5.x | | CDK 4.1 Powered by Apache Kafka documentation |
| | • Schema Registry | Cloudera Streams Processing 1.0.0 documentation |
| | • Schema Registry<br>• Streams Messaging Manager<br>• Streams Replication Manager | Cloudera Streams Processing 2.0.x documentation |
| Kafka in CDH 6.x | | CDH 6.x Kafka documentation |
| | • Schema Registry<br>• Streams Messaging Manager<br>• Streams Replication Manager | Cloudera Streams Processing documentation |
| Kafka for HDF and HDP | | HDF documentation |

# Data Hub cluster definitions

The Streams Messaging templates include Kafka, Schema Registry, Streams Messaging Manager, Streams Replication Manager and ZooKeeper. You may choose from the following template options, depending on your operational objectives:

- Streams Messaging Heavy Duty for AWS
- Streams Messaging Heavy Duty for Azure
- Streams Messaging Heavy Duty for GCP
- Streams Messaging Light Duty for AWS
- Streams Messaging Light Duty for Azure
- Streams Messaging Light Duty for GCP

Streams Messaging provides advanced messaging and real-time processing on streaming data using Apache Kafka, centralized schema management using Schema Registry, management and monitoring capabilities powered by Streams Messaging Manager, as well as cross-cluster Kafka topic replication using Streams Replication Manger and Kafka partition rebalancing with Cruise Control.

These templates set up fault-tolerant standalone deployments of Apache Kafka and supporting Cloudera components (Schema Registry. Streams Messaging Manager, Streams Replication Manager and Cruise Control), which can be used for Kafka workloads in the cloud or as a disaster recovery instance for on-premises Kafka clusters.

**Note:**

Streams Messaging cluster definitions and templates available in CDP Public Cloud do not support the following Cloudera Runtime Streaming components and features.

- Kafka Connect

# Streams Messaging cluster layout

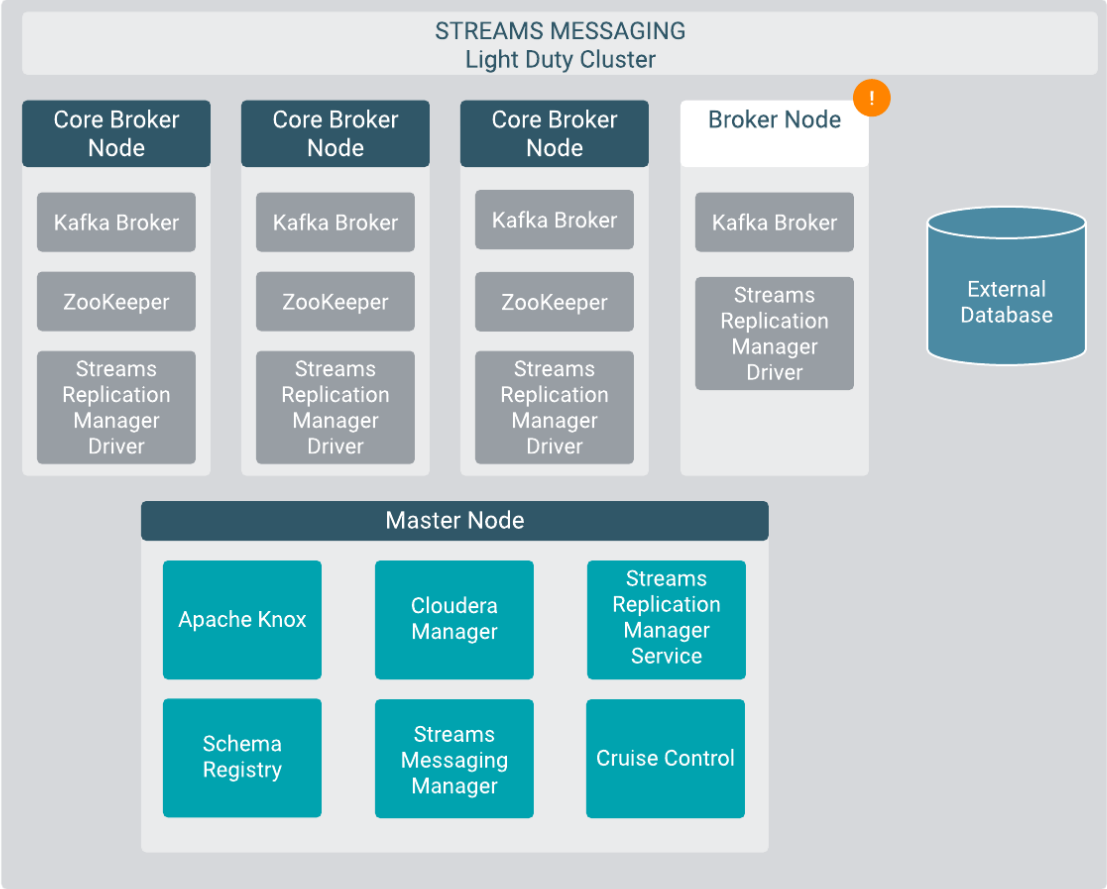
Describes the layout and capacity of the Streams Messaging: Light Duty and Streams Messaging: Heavy Duty cluster definitions

## Streams Messaging: Light Duty cluster layout

You can use a Streams Messaging: Light Duty cluster definition in development, testing, or proof of concept scenarios. Each of the cluster nodes includes:

- Kafka Core Broker, Kafka Broker, ZooKeeper, and the SRM driver are co-located on all instances
- For each node hosting Kafka Core Broker, Kafka Broker, ZooKeeper, and the SRM driver instances:
  - AWS: m5.2xlarge
  - Azure: D8_v3
  - GCP: e2-standard-8
- Storage configuration per Kafka Core Broker node:
  - AWS: 1 TB Volume EBS ST1
  - Azure: 1 TB Standard Locally-redundant SSD storage
  - GCP: 1 TB Zonal PD-SSD
- Storage configuration per Kafka Broker node:
  - AWS: 1 TB Volume EBS ST1
  - Azure: 1 TB Standard Locally-redundant SSD storage
  - GCP: 1 TB Zonal PD-SSD
- Master node instance
  - AWS: m5.2xlarge
  - Azure: Standard_D8_v3
  - GCP: e2-standard-8
- Storage configuration for Master node:
  - AWS: 100 GB Volume EBS Magnetic
  - Azure: 100 GB Standard Locally-redundant SSD storage
  - GCP: 100 GB Zonal PD-Standard

For more information about the cloud provider-specific instance and storage types, see the *Related Information* section.




The Broker node is not provisioned by default. You have the option to manually set how many Broker nodes are created when provisioning the cluster. After the cluster is provisioned, the number of Broker nodes can be changed by scaling your cluster. For more information about scaling, Core Broker and Broker nodes, see *Scaling Streams Messaging Clusters*.

**Important:** By default, the volume per instance count for Broker and Core Broker nodes is identical. If you customize your cluster during provisioning, Cloudera recommends that Attached Volume per Instances is set to the same value for both node types. Alternatively, if you want to provision a cluster where the number of volumes is not identical, ensure that you complete *Configure data directories for clusters with custom disk configurations* after the cluster is provisioned. Otherwise, Kafka does not utilize all available volumes. Additionally, scaling the cluster might also not be possible.

### Streams Messaging: Heavy Duty cluster layout

You can use the Streams Messaging: Heavy Duty cluster definition in production scenarios. The cluster definition includes:

Azure

- Master Node – Containing Knox, Cloudera Manager, ZooKeeper
  - Instance type – Standard_D8_v3
  - Storage configuration – 100 GB Standard Locally-redundant SSD storage

- Registry Nodes – Containing Schema Registry, ZooKeeper
  - Instance type – Standard_D8_v3
  - Storage configuration – 100 GB Standard Locally-redundant HDD storage
- SMM Nodes – Containing SMM, Schema Registry, ZooKeeper
  - Instance type – Standard_D8_v3
  - Storage configuration – 100 GB Standard Locally-redundant HDD storage
- Core Broker Nodes – Containing a Kafka Broker
  - Instance type – Standard_D8s_v3
  - Storage configuration – 1 TB Premium Locally-redundant SSD storage
- Broker Nodes – Containing a Kafka Broker
  - Instance type – Standard_D8s_v3
  - Storage configuration – 1 TB Premium Locally-redundant SSD storage
- SRM Nodes – Containing the SRM Driver and Service
  - Instance type – Standard_D8_v3
  - Storage configuration – 100 GB Standard Locally-redundant HDD storage

AWS

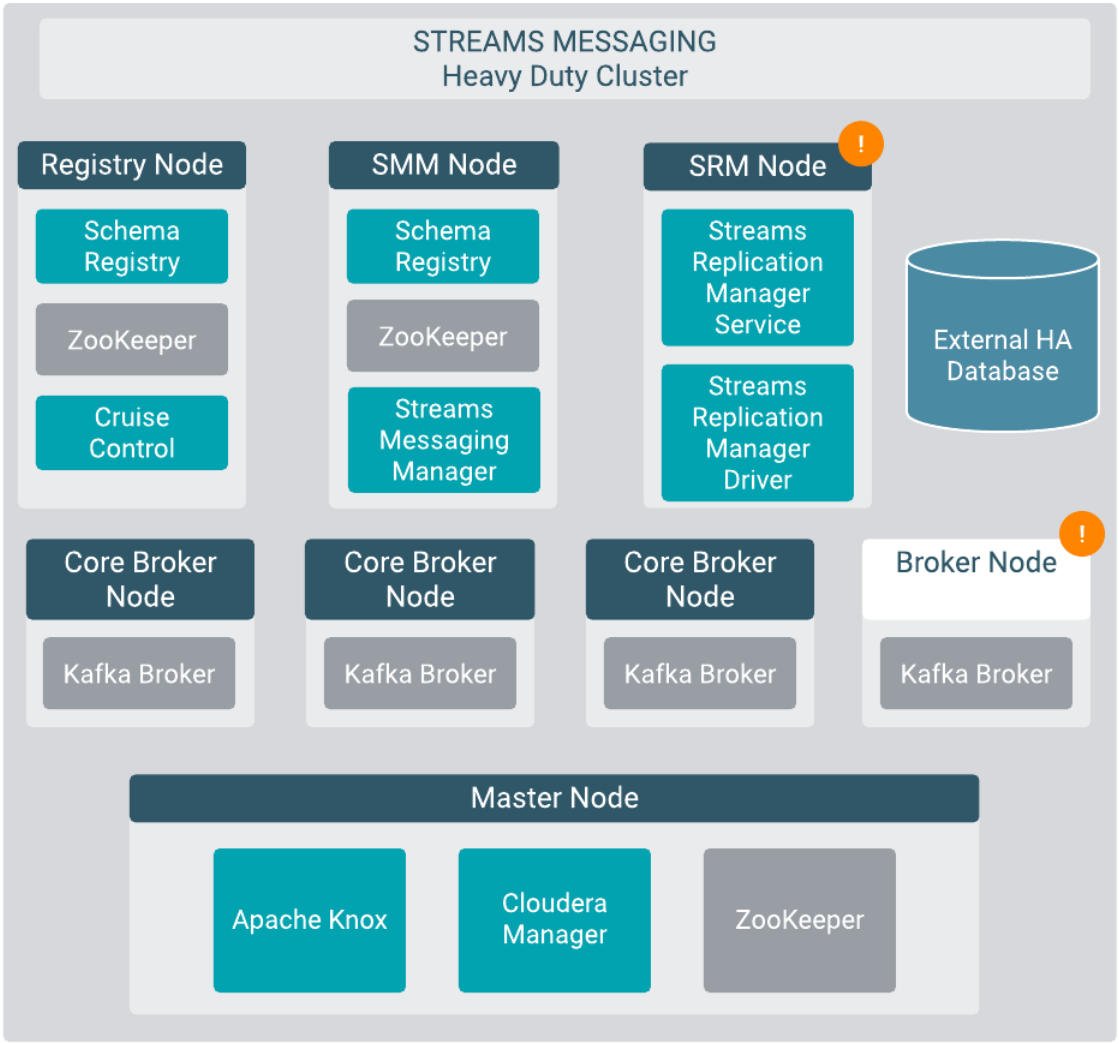
- Master Node – Containing Knox, Cloudera Manager, ZooKeeper
  - Instance type – r5.2xlarge
  - Storage configuration – 100 GB EBS SC1
- Registry Nodes – Containing Schema Registry, ZooKeeper
  - Instance type – m5.2xlarge
  - Storage configuration – 100 GB EBS SC1
- SMM Nodes – Containing SMM, Schema Registry, ZooKeeper
  - Instance type – m5.2xlarge
  - Storage configuration – 100 GB EBS SC1
- Core Broker Nodes – Containing a Kafka Broker
  - Instance type – m5.2xlarge
  - Storage configuration – 1 TB GP2 SSD
- Broker Nodes – Containing a Kafka Broker
  - Instance type – m5.2xlarge
  - Storage configuration – 1 TB GP2 SSD
- SRM Nodes - Containing the SRM Driver and Service
  - Instance type – m5.2xlarge
  - Storage configuration – 100 GB EBS SC1

GCP

- Master Node – Containing Knox, Cloudera Manager, ZooKeeper
  - Instance type – e2-standard-8
  - Storage configuration – 100 GB Zonal PD-Standard
- Registry Nodes – Containing Schema Registry, ZooKeeper
  - Instance type – e2-standard-8
  - Storage configuration – 100 GB Zonal PD-Standard
- SMM Nodes – Containing SMM, Schema Registry, ZooKeeper
  - Instance type – e2-standard-8
  - Storage configuration – 100 GB Zonal PD-Standard

- Core Broker Nodes – Containing a Kafka Broker
  - Instance type – e2-standard-8
  - Storage configuration – 1 TB Premium Locally-redundant SSD storage
- Broker Nodes – Containing a Kafka Broker
  - Instance type – e2-standard-8
  - Storage configuration – 1 TB Premium Locally-redundant SSD storage
- SRM Nodes - Containing the SRM Driver and Service
  - Instance type – e2-standard-8
  - Storage configuration – 100 GB Zonal PD-Standard

For more information about the cloud provider-specific instance and storage types, see the *Related Information* section.




The SRM and Broker nodes are not provisioned by default. When provisioning a cluster with this definition, you have to manually set the instance count of the appropriate host group to at least 1. Otherwise the host group and its nodes are not provisioned. After a cluster is provisioned, you also have the option to scale these nodes. For more information on scaling, see *Scaling Streams Messaging Clusters*.

**Important:** By default, the volume per instance count for Broker and Core Broker nodes is identical. If you customize your cluster during provisioning, Cloudera recommends that Attached Volume per Instances is set to the same value for both node types. Alternatively, if you want to provision a cluster where the number of volumes is not identical, ensure that you complete *Configure data directories for clusters with custom disk configurations* after the cluster is provisioned. Otherwise, Kafka does not utilize all available volumes. Additionally, scaling the cluster might also not be possible.

### Related Information

AWS instance types
Azure instance types
GCP instance types
AWS storage information
Azure storage information
GCP storage information
Scaling Streams Messaging Clusters
Configure data directories for clusters with custom disk configurations