Informatica® Multidomain MDM
10.4

# Infrastructure Planning Guide

# Table of Contents

# Preface

Read the Informatica® *Multidomain MDM Infrastructure Planning Guide* before you plan the infrastructure and the architecture of the Multidomain MDM environment. The guide also includes sample installation topologies.

## Informatica Resources

Informatica provides you with a range of product resources through the Informatica Network and other online portals. Use the resources to get the most from your Informatica products and solutions and to learn from other Informatica users and subject matter experts.

### Informatica Network

The Informatica Network is the gateway to many resources, including the Informatica Knowledge Base and Informatica Global Customer Support. To enter the Informatica Network, visit https://network.informatica.com.

As an Informatica Network member, you have the following options:

- Search the Knowledge Base for product resources.
- View product availability information.
- Create and review your support cases.
- Find your local Informatica User Group Network and collaborate with your peers.

### Informatica Knowledge Base

Use the Informatica Knowledge Base to find product resources such as how-to articles, best practices, video tutorials, and answers to frequently asked questions.

To search the Knowledge Base, visit https://search.informatica.com. If you have questions, comments, or ideas about the Knowledge Base, contact the Informatica Knowledge Base team at KB_Feedback@informatica.com.

### Informatica Documentation

Use the Informatica Documentation Portal to explore an extensive library of documentation for current and recent product releases. To explore the Documentation Portal, visit https://docs.informatica.com.

If you have questions, comments, or ideas about the product documentation, contact the Informatica Documentation team at infa_documentation@informatica.com.

## Informatica Product Availability Matrices

Product Availability Matrices (PAMs) indicate the versions of the operating systems, databases, and types of data sources and targets that a product release supports. You can browse the Informatica PAMs at https://network.informatica.com/community/informatica-network/product-availability-matrices.

## Informatica Velocity

Informatica Velocity is a collection of tips and best practices developed by Informatica Professional Services and based on real-world experiences from hundreds of data management projects. Informatica Velocity represents the collective knowledge of Informatica consultants who work with organizations around the world to plan, develop, deploy, and maintain successful data management solutions.

You can find Informatica Velocity resources at http://velocity.informatica.com. If you have questions, comments, or ideas about Informatica Velocity, contact Informatica Professional Services at ips@informatica.com.

## Informatica Marketplace

The Informatica Marketplace is a forum where you can find solutions that extend and enhance your Informatica implementations. Leverage any of the hundreds of solutions from Informatica developers and partners on the Marketplace to improve your productivity and speed up time to implementation on your projects. You can find the Informatica Marketplace at https://marketplace.informatica.com.

## Informatica Global Customer Support

You can contact a Global Support Center by telephone or through the Informatica Network.

To find your local Informatica Global Customer Support telephone number, visit the Informatica website at the following link:
https://www.informatica.com/services-and-training/customer-success-services/contact-us.html.

To find online support resources on the Informatica Network, visit https://network.informatica.com and select the eSupport option.

CHAPTER 1

# Introduction to Infrastructure Planning

This chapter includes the following topics:

## Introduction to Infrastructure Planning Overview

Master Data Management (MDM) is the controlled process to enhance data reliability and data maintenance procedures for an organization. Multidomain MDM is also referred to as the MDM Hub. The MDM Hub is deployed as part of a broader Data Governance program that includes a combination of technology, people, policy, and process. Plan the infrastructure for the MDM Hub deployment based on the data policy definitions, strategies, and objectives of your organization.

The MDM Hub has core and optional components. Decide on the components that you want in the MDM Hub environment. Also, you need to decide on the infrastructure components such as the operating system, database system, application server, and load balancers. Before you install and deploy the MDM Hub, be clear about your installation and deployment objectives. Also, you need to decide on an installation topology that meets your objectives.

To ensure successful implementation of the MDM Hub, collate the information that the MDM Hub implementers require in an installation requirements form.

## Installation Requirements Form

Create an installation requirements form to provide information that the MDM Hub implementers require for a successful MDM Hub implementation. Base the installation requirements on your business and technical requirements. Also, take into consideration the installation and deployment objectives.

You can add the following installation requirement information to the installation requirements form:

- Detailed installation topology
- Optional MDM Hub installation components to deploy
- Timeline granularity

- External cleanse engines
- Operating system locale for the MDM Hub components
- HTTPS protocol
- Database type
- Application server type
- Security configuration for password hashing
- Search configuration with Elasticsearch
- Detailed installation topology

# CHAPTER 2

# Business and Technical Requirements

This chapter includes the following topics:

## Business and Technical Requirements Overview

When you plan the infrastructure for the MDM Hub environment, consider your business and technical requirements. You might need to identify the business and technical requirement in consultation with other stakeholders in the organization that have an interest in the MDM Hub environment.

Before the MDM Hub is implemented, the implementers need to know the MDM Hub components that need to be deployed. Also, the implementers need to know the business and technical requirements such as the Timeline granularity, the operating system locale, and the need for secure communication.

# Identify the Installation Components

The MDM Hub enhances data reliability and data maintenance procedures. You can access the MDM Hub features through the Hub Console. The MDM Hub has core and optional installation components. Based on your business requirements, decide on the optional components that you want to install.

## Core Components

The following table describes the core installation components:

| Component | Description |
| --- | --- |
| MDM Hub Master Database | A schema that stores and consolidates business data for the MDM Hub. Contains the MDM Hub environment configuration settings, such as user accounts, security configuration, Operational Reference Store registry, and message queue settings. You can access and manage an Operational Reference Store from an MDM Hub Master Database. The default name of an MDM Hub Master Database is CMX_SYSTEM, but you can use a custom name. |
| | You can create multiple MDM Hub Master Databases, each with its own set of Operational Reference Stores, in the same database instance. |
| Operational Reference Store | A schema that stores and consolidates business data for the MDM Hub. Contains the master data, content metadata, and the rules to process and manage the master data. You can configure separate Operational Reference Store databases for different geographies, different organizational departments, and for the development and production environments. You can distribute Operational Reference Store databases across multiple server machines. The default name of an Operational Reference Store is CMX_ORS. |
| Hub Server | A J2EE application that you deploy on an application server. The Hub Server processes data stored within the MDM Hub and integrates the MDM Hub with external applications. The Hub Server manages core and common services for the MDM Hub. |
| Process Server | A J2EE application that you deploy on an application server. The Process Server processes batch jobs such as load, recalculate BVT, and revalidate, and performs data cleansing and match operations. The Process Server interfaces with the cleanse engine that you configure to standardize and optimize data for match and consolidation. |
| Provisioning tool | A tool to build business entity models, and to configure the Entity 360 framework for Data Director. After you build business entity models, you can publish the configuration to the MDM Hub. |
| Informatica ActiveVOS ® | A business process management (BPM) tool that is required internally by the MDM Hub for processing data. Informatica ActiveVOS supports automated business processes, including change-approval processes for data. You can also use Informatica ActiveVOS to ensure that changes to master data undergo a review-and-approval process before inclusion in the best version of the truth (BVT) records. |
| | When you install ActiveVOS Server as part of the Hub Server installation, you install the ActiveVOS Server, ActiveVOS Console, and Process Central. Also, you install predefined MDM workflows, tasks, and roles. |
| Data Director (IDD) | A user interface to master and manage the data that is stored in the MDM Hub. In IDD, data is organized by business entities, such as customers, suppliers, and employees. Business entities are data groups that have significance for organizations. |

### Optional Components

The following table describes the optional installation components:

| Component | Description |
|---|---|
| Resource Kit | Set of samples, applications, and utilities to integrate the MDM Hub into your applications and workflows. You can select the Resource Kit components that you want to install. |
| Informatica platform | An environment that comprises the Informatica services and Informatica clients that you use to cleanse and transfer source data to the MDM Hub. You can use Informatica platform instead of the cleanse functions available in the MDM Hub to cleanse data. <br><br> When you install the Informatica platform as part of the Hub Server installation, you install the Data Integration Service, Model Repository Service, and Informatica Developer (the Developer tool). |
| Dynamic Data Masking | A data security tool that operates between the MDM Hub and databases to prevent unauthorized access to sensitive information. Dynamic Data Masking intercepts requests sent to databases and applies data masking rules to the request to mask the data before it is sent back to the MDM Hub. |
| Informatica Data Controls (IDC) | Applicable to Informatica Data Director (IDD) based on the subject area data model only. <br><br> IDC is a set of user interface controls that expose the MDM Hub data in third-party applications that are used by business users. |
| Zero Downtime (ZDT) module | A module to ensure that applications have access to data in the MDM Hub during the MDM Hub upgrade. In a ZDT environment, you duplicate the databases: source databases and target databases. During the MDM Hub upgrade, the ZDT module replicates the data changes in the source databases to the target databases. <br><br> To buy the ZDT module, contact your Informatica sales representative. For information about installing a zero downtime environment, see the *Multidomain MDM Zero Downtime Installation Guide* for the database. |

# Determine the Database Environment Requirements

You can store the MDM Hub data in the following database environments:

- Oracle Database
- IBM Db2
- Microsoft SQL Server

Based on your business requirements, decide on the database environment that you want to set up. For more information about the supported database environments, see the Product Availability Matrix (PAM). You can access PAMs at
https://network.informatica.com/community/informatica-network/product-availability-matrices.

Also, decide on how you want to set up the database environment to include the MDM Hub Master Database and Operational Reference Stores.

## Oracle RAC

The Oracle RAC environment increases performance, improves fault tolerance, and provides scalability. Determine whether your business requires the benefit of using Oracle RAC with the MDM Hub.

For information about Oracle RAC, see the Oracle documentation.

### Considerations for using Oracle RAC

Based on the following considerations, decide on the Oracle RAC environment setup:

- Use Oracle service names instead of Oracle SIDs for Oracle RAC installations. Provides flexibility to specify the connection and to dynamically reallocate database servers.

- Configure all the Oracle RAC nodes in the `tnsnames.ora` file.

- Use Oracle RAC load-balanced connections. Distributes the workload among all the available nodes in a cluster. If a node becomes unavailable, the MDM Hub batch job that runs on the node fails. The batch job can be re-triggered and the job will run on one of the available nodes in the cluster.

The MDM Hub supports load balancing for the following operations and components:

- Hub Console operations
- All batch jobs except the Generate Match Tokens job

  **Note:** The MDM Hub supports load balancing for the fuzzy matching portion of the Match jobs and for the cleanse process portion of the Stage jobs.

  Batch jobs cannot use load balancing when called from the Hub Console.

- Services Integration Framework (SIF)
- Outbound JMS message queues
- Hub Server
- Process Server
- Provisioning tool
- Repository Manager

  **Note:** Repository Manager cannot use load balancing when DDL is required because DDL uses a direct JDBC connection.

The MDM Hub does not support load balancing for Data Director.

## MDM Hub Master Databases and Operational Reference Stores

Based on your business requirements, decide how to set up the MDM Hub Master Database and the Operational Reference Stores.

An MDM Hub Master Database is associated with a set of Operational Reference Stores. Decide how many Operational Reference Stores to create. Also, decide whether you want to distribute the Operational Reference Stores across multiple machines. Your decision determines the installation topology.

If you have a large database management system, you can set up multiple MDM Hub environments. For each MDM Hub environment, you must set up an MDM Hub Master Database with its own tablespaces, a unique ActiveVOS schema, and a set of Operational Reference Stores. You can set up multiple MDM Hub environments in a single database instance or multiple database instances.

# Identify the Application Server Environment

You can deploy the MDM Hub in the following application server environments:

- JBoss
- WebLogic

- WebSphere

Based on your business requirements, decide on the application server environment that you want to set up. For more information about the supported application server environments, see the Product Availability Matrix (PAM). You can access PAMs at
https://network.informatica.com/community/informatica-network/product-availability-matrices.

## Logical Grouping of Java Virtual Machines

Determine whether your business needs to create logical Java Virtual Machine (JVM) groups. When you deploy the Hub Server and Process Server applications in a logical JVM group, all communication between the applications stays within the group. To group JVMs, you assign a group ID to each JVM in the MDM Hub environment.

You can use logical JVM groups for the following scenarios:

- You need multiple JVMs and want to use some JVMs as primary nodes and some as secondary nodes. You can use the secondary nodes for specific operations or when the primary nodes are overloaded. For example, you can configure a logical JVM group for primary processes and another group for secondary or backup processes.

- You want to run multiple batch group jobs in parallel and want control over the available resources, such as threads. For example, you can create logical JVM groups for batch group jobs that run on the Party table and on the Address table.

- You want to group similar processes. For example, you can create a logical JVM group for SIF API calls and another group for batch jobs.

# Determine the Timeline Granularity

Timeline granularity is the time measurement unit that you want to use to define effective periods for versions of records. For example, you can choose the effective periods to be in years, months, or seconds. Decide on the timeline granularity and provide the information to the MDM Hub implementers.

You can configure the timeline granularity of year, month, day, hour, minute, or seconds to specify effective periods of data in the MDM Hub implementation. You can configure the timeline granularity that you need when you create or update an Operational Reference Store.

**Important:** The timeline granularity that you configure cannot be changed.

When you specify an effective period in any timeline granularity, the system uses the database time locale for the effective periods. To create a version that is effective for one timeline measurement unit, the start date and the end date must be the same.

The following table describes the timeline granularity options that you can configure:

| Timeline Granularity | Description |
|---|---|
| Year | When the timeline granularity is year, you can specify the effective period in the year format, yyyy, such as 2010. An effective start date of a record starts at the beginning of the year and the effective end date ends at the end of the year. For example, if the effective start date is 2013 and the effective end date is 2014, then the record would be effective from 01/01/2013 to 31/12/2014. |
| Month | When the timeline granularity is month, you can specify the effective period in the month format, mm/yyyy, such as 01/2013. An effective start date of a record starts on the first day of a month. The effective end date of a record ends on the last day of a month. For example, if the effective start date is 02/2013 and the effective end date is 04/2013, the record is effective from 01/02/2013 to 30/04/2013. |
| Day | When the timeline granularity is day, you can specify the effective period in the date format, dd/mm/yyyy, such as 13/01/2013. An effective start date of a record starts at the beginning of a day, that is 12:00. The effective end date of the record ends at the end of a day, which is 23:59. For example, if the effective start date is 13/01/2013 and the effective end date is 15/04/2013, the record is effective from 12:00 on 13/01/2013 to 23:59 on 15/04/2013. |
| Hour | When the timeline granularity is hour, the effective period includes the year, month, day and hour. The timeline format is dd/mm/yyyy hh, such as 13/01/2013 15. An effective start date of a record starts at the beginning of an hour of a day. The effective end date of the record ends at the end of the hour that you specify. For example, if the effective start date is 13/01/2013 15 and the effective end date is 15/04/2013 10, the record is effective from 15:00 on 13/01/2013 to 10:59 on 15/04/2013. |
| Minute | When the timeline granularity is minute, the effective period includes the year, month, day, hour, and minute. The format timeline format is dd/mm/yyyy hh:mm, such as 13/01/2013 15:30. An effective start date of a record starts at the beginning of a minute. The effective end date of the record ends at the end of the minute that you specify. For example, if the effective start date is 13/01/2013 15:30 and the effective end date is 15/04/2013 10:45, the record is effective from 15:30:00 on 13/01/2013 to 10:45:59 on 15/04/2013. |
| Second | When the timeline granularity is second, the effective period includes the year, month, day, hour, minute, and second. The timeline format is dd/mm/yyyy hh:mm:ss, such as 13/01/2013 15:30:45. An effective start date of a record starts at the beginning of a second. The effective end date ends at the end of the second that you specify. For example, if the effective start date is 13/01/2013 15:30:55 and the effective end date is 15/04/2013 10:45:15, the record is effective from 15:30:55:00 on 13/01/2013 to 10:45:15:00 on 15/04/2013. |

# Identify External Cleanse Engines

If you intend to integrate a cleanse engine with the MDM Hub, identify the cleanse engines. You can integrate cleanse engines such as Address Verification with the MDM Hub.

The following table lists the cleanse engines that the MDM Hub supports and the Informatica MDM adapters with which they work:

| Cleanse Engine | Informatica MDM Hub Adapter |
|---|---|
| IDQ | Informatica IDQ Adapter |
| Informatica Address Verification | Informatica Address Verification Adapter |
| Trillium | Trillium Director Adapter |

For more information about cleanse engines that you can integrate with the MDM Hub, see the *Multidomain MDM Cleanse Adapter Guide*.

# Determine the Operating System Locale

The operating system locale defines the language and region of users. Based on the business requirements, set the same operating system locale for the Hub Server, the Hub Store, and the Hub Console.

Choose one of the following locales for the MDM Hub components:

- en_US
- fr_FR
- de_DE
- ja_JP
- ko_KR
- zh_CN
- ES
- pt_BR

# Determine the HTTPS Protocol Requirement

You can configure the HTTPS protocol for the MDM Hub communications. Also, you might want to use the HTTPS protocol for the communication between ActiveVOS and the MDM Hub.

The decision to secure the MDM Hub communications depends on your business needs. You need to indicate to the MDM Hub implementers whether you want to secure the MDM Hub communications.

# Determine the Security Configuration for Password Hashing

Password hashing is a way to irreversibly encrypt passwords through a cryptographic hash function. The MDM Hub uses a password hashing method to protect user passwords and ensure passwords are never stored in clear text form in a database. The MDM Hub administrator configures the password hashing options, such as the algorithm and customer hashing keys, during installation of the Hub Server.

Determine the security configuration options for password hashing that the implementers need to specify during the MDM Hub installation.

The implementers need to specify the following security configuration options for password hashing:

- Whether to create a custom hash key as part of the hashing algorithm
- Whether to use the default SHA3 hashing algorithm or create a custom hashing algorithm
- Whether to use the default certificate provider or use a custom certificate provider

# Determine the Search Configuration with Elasticsearch

To provide business users and data stewards with a fast, modern, full-text search experience, you must configure search with Elasticsearch.

Elasticsearch is an open-source, full-text search engine that provides distributed indexing and search. You can install Elasticsearch on any machine where the MDM Hub components are installed or on a separate machine. Based on your MDM Hub topology and the amount of data to index, determine the number of nodes to configure for the Elasticsearch cluster. Each node can in turn have multiple indexes. You can decide to split the indexes into multiple shards.

For information about how to determine the number of nodes to configure for the Elasticsearch cluster and how to configure indexes, see the *Elasticsearch documentation*.

# CHAPTER 3

# Installation and Deployment Considerations

This chapter includes the following topics:

## Installation and Deployment Considerations Overview

Before you install and deploy the MDM Hub, first consider the installation and deployment objectives. Then, you can decide whether to install and deploy the MDM Hub on standalone application server instances or on application server clusters.

Informatica recommends that you install and deploy the MDM Hub on standalone application server instances. To achieve high availability, you can use external load balancers for real time API calls. To scale the MDM Hub implementation, you can add standalone application server instances and deploy additional MDM Hub components. You can achieve load balancing for batch jobs through the internal mechanism of the MDM Hub.

If you install and deploy the MDM Hub in application server clusters, sessions do not replicate or fail over to other nodes in the cluster. You can use SIF APIs or the Hub Console to run the MDM Hub batch jobs, but the batch jobs do not fail over because the sessions do not replicate. However, if a batch job request is through the Hub Console, the request and not the batch job fails over to the active nodes in the cluster. Also, if an application server node fails, the Informatica Data Director (IDD) session is lost and does not replicate.

## Installation and Deployment Objectives

Before you install and deploy the MDM Hub, consider the following installation and deployment objectives:

- High availability
- Scalability
- Load balancing
- Maintainability

# High Availability

High availability is the ability of a system to continue functioning after the failure of one or more of the servers. You can achieve high availability of the MDM Hub either with multiple standalone application server instances or with application server clusters.

The application servers cannot make the MDM Hub highly available because the MDM Hub uses stateless session beans. The stateless session beans do not maintain a conversational state with clients, and therefore application servers cannot synchronize the state of the beans across the application server cluster nodes.

To achieve high availability, the MDM Hub uses an internal metadata cache mechanism. The metadata caching mechanism synchronizes the metadata in the MDM Hub environment and makes it available across the MDM Hub implementation. If an application on one machine fails, the metadata is available in the cache for the applications that are online. The metadata caching mechanism uses Infinispan, which is a replicated cache that can handle metadata caching requirements in any application server environment.

Consider the following contexts that impact your decision in favor of a highly available environment:

- If the MDM Hub implementation contains multiple Hub Server instances, during a failure, the Hub Console operations do not fail over to an active node. To ensure that the Hub Console operations fail over to an active node, the Hub Server instances must be part of an application server cluster.

- If a batch job request is through the Hub Console, the request fails over to the active nodes in the cluster. If a batch job request is through a Services Integration Framework API, the request does not fail over to the active nodes in the cluster. The batch jobs do not fail over because the batch jobs do not replicate.

- If the MDM Hub implementation contains multiple Hub Server instances and you use JMS messages, you can deploy the Hub Server instances in a cluster. If you do not deploy the Hub Server instances in a cluster, the outbound JMS messages are not available to all consumers. Also, you can consider managing this scenario by using an appropriate JMS server deployment strategy.

- If you use Informatica Data Director (IDD), the IDD session binds to the application server node that services the session. If the application server node fails, the IDD session ends. The IDD session does not replicate. The IDD user needs to log in again.

# Scalability

Scalability is the ability of a system to accommodate any increase in demand for resources and processing power. You can achieve scalability of the MDM Hub either with standalone application servers or with application server clusters.

The following features of the MDM Hub implementation make it highly scalable:

**MDM Hub cache implementation**

The MDM Hub cache implementation uses a distribution mechanism that is independent of application servers.

**Multithreaded Process Server instances**

Process Server instances are multithreaded and can process multiple requests concurrently. The MDM Hub supports multithreading for the Hub Console operations, batch jobs, and Services Integration Framework (SIF) requests.

**Multiple Process Server instances**

You can run multiple Process Servers for each Operational Reference Store in the MDM Hub.

The MDM Hub does not require external components for scalability. If the volume of data increases, to scale the MDM Hub implementation, you can add more Process Server instances. To distribute the processing load across multiple CPUs and run batch jobs in parallel, deploy Process Servers on multiple hosts.

# Load Balancing

Load balancing is the ability of a system to distribute workloads to the available resources. You can achieve load balancing with Process Servers deployed on standalone application server instances.

The Process Server instances of an MDM Hub implementation use an internal load balancing mechanism. You do not need the load balancing capabilities of application server clusters. You can install and deploy the MDM Hub on standalone application servers and use the load balancing capabilities of the Process Server. When you install and deploy the MDM Hub on standalone application servers, use the load balancing capabilities of the Process Server.

**Note:** To use outbound JMS message queues or load balance Hub Console operations, deploy the Hub Server instances in an application server cluster. Do not deploy the Process Server instances in an application server cluster.

# Maintainability

Maintainability is the flexibility with which you can make changes or upgrade the MDM Hub implementation. You can maintain the MDM Hub implementation on standalone application servers or as part of application server clusters.

You can use application server clusters for coordinated multiserver management, which is not possible with standalone application server instances. The management and maintenance of changes to the configuration and deployment of the MDM Hub on an application server cluster is simpler than on multiple standalone application server instances.

Consider the frequency of the maintenance tasks, which impacts your decision in favor of a highly maintainable environment. During an MDM Hub installation or upgrade on standalone application server instances, you need to install or upgrade and deploy on each application server instance. Also, there are multiple configurations that need to be updated on each machine. On application server clusters, the installation or upgrade and deployment is relatively less tedious.

**Note:** Deploy the Process Server instances on application server clusters only if you expect great benefits of maintainability.

# CHAPTER 4

# Sample Installation Topologies

This chapter includes the following topics:

## Sample Installation Topologies

When you decide on an installation topology, aim to balance system characteristics such as high availability, scalability, and load balancing requirements. To ensure that you use an ideal installation topology, you need to understand your particular usage scenario. The sample installation topologies provide ideas to plan your installation topology.

The sample installation topologies show some ways in which the MDM Hub components can be set up in an MDM Hub implementation. You can customize them to suit your needs.

When you plan the installation topology, consider one of the following sample installation topologies as a starting point:

- Standalone application server instance topology
- Multiple application server instances topology
- Application server cluster topology

**Note:** All the components of the MDM Hub implementation must be the same version. If you have multiple versions of the MDM Hub, install each version in a separate environment.

## Standalone Application Server Instance Topology

In a standalone application server instance topology, you install all the MDM Hub components on a standalone application server instance. The standalone application server instance topology is the most basic topology. Deployment on a standalone application server instance simplifies communication among the MDM Hub components.

The standalone application server instance topology has no provision for planned or unplanned downtime. Scalability is not possible or limited to the extent of the processing power of the machine on which you

deploy the Hub Server and the Process Server. The topology is simple to maintain. Use the topology for small data volumes.

The sample installation topology contains two machines. An application server instance is installed on one machine and a database server is installed on the other machine. The Hub Server and the Process Server are deployed on the machine with the application server instance. The Hub Store is configured on the machine with the database server.

The following image shows a sample standalone application server instance installation topology:
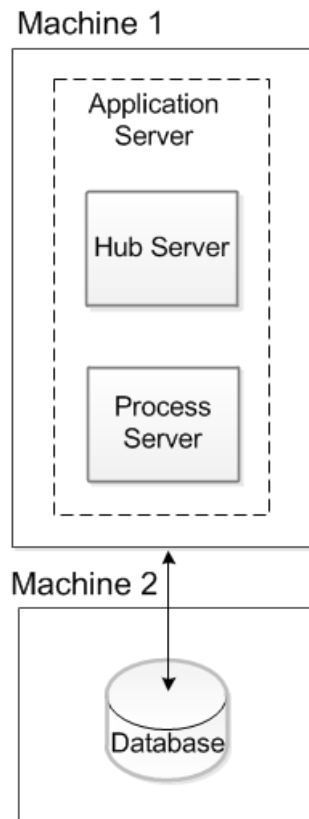


The following table describes the capabilities of the standalone application server instance topology:

| Capability | Availability |
|---|---|
| High availability | None. |
| Scalability | Yes.<br>To scale the MDM Hub to support large data volumes, configure multithreading for the Process Server. You can vertically scale the MDM Hub environment by adding more processing power.<br>The MDM Hub supports multithreading for the following operations and components:<br>- Hub Console operations<br>- Batch jobs<br>- Services Integration Framework (SIF) |
| Load balancing | None. |
| Maintainability | Easy to maintain because all the MDM Hub components are deployed on a single machine with an application server instance. |

# Multiple Application Server Instances Topology

In a multiple application server instances topology, you distribute the installation of the MDM Hub components across multiple application server instances.

To set up a multiple application server instances topology, you need multiple machines with application server instances. The topology provides scalability. To scale the processing capability of the MDM Hub implementation, deploy additional Process Server instances on additional application server instances.

If a Process Server fails, batch jobs running on the Process Server fail. The batch jobs do not fail over and complete on Process Server instances that are online. You must start the batch jobs again. The internal load-balancing mechanism of the MDM Hub distributes the batch job requests between the Process Server instances that are online.

You can use the multiple application server instances topology for high data volumes. The topology supports high volume batch jobs by distributing the load across the Process Servers that you configure.

**Note:** If the MDM Hub implementation contains multiple Hub Server instances and you use JMS message queues, to consume outbound JMS messages, deploy the Hub Server instances in a cluster. Otherwise, each application server instance will have a different outbound JMS destination.
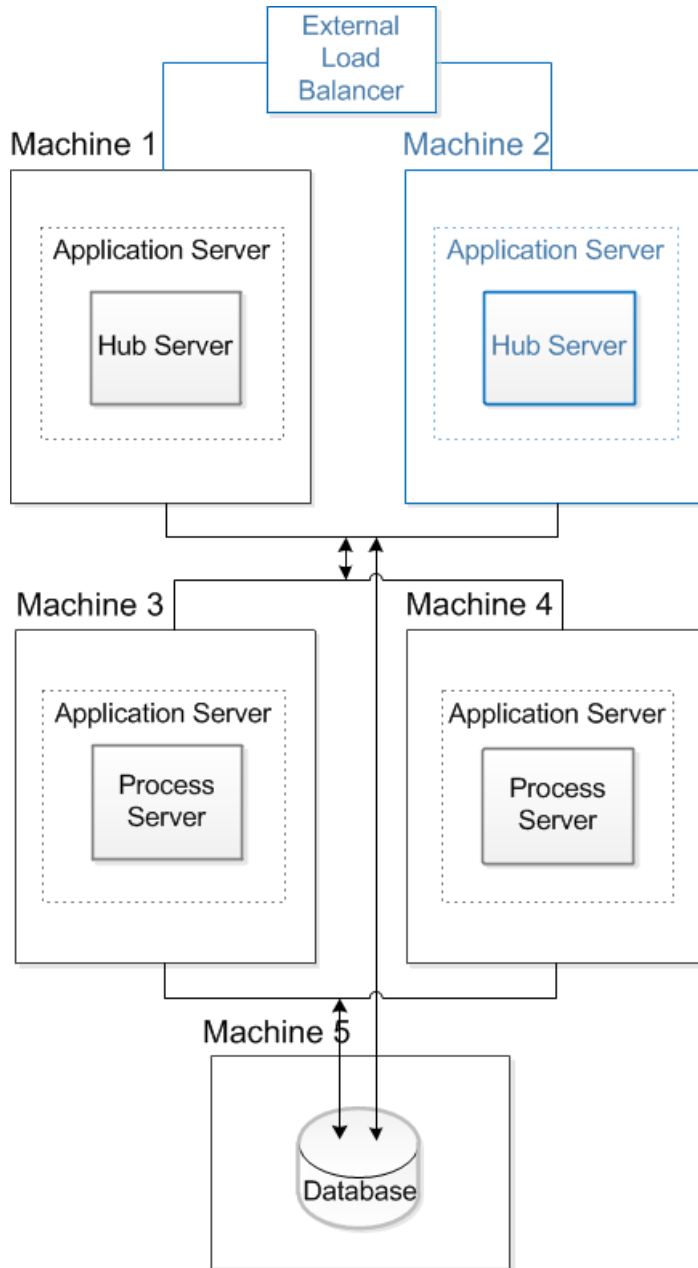
The sample installation topology contains four machines. An application server instance is installed on three of the four machines. The Hub Server is deployed on the application server instance on one of the machines. The Process Server instances are deployed on the application server instances on the other two machines. The Hub Server distributes the processing load for batch jobs between the two Process Server instances. If one of the Process Server instances fail or is offline, the Hub Server sends the processing request to the other Process Server that is online. The Hub Store is configured on the fourth machine on which a database server is installed.

The following image shows a sample multiple application server instances installation topology that is not highly available:
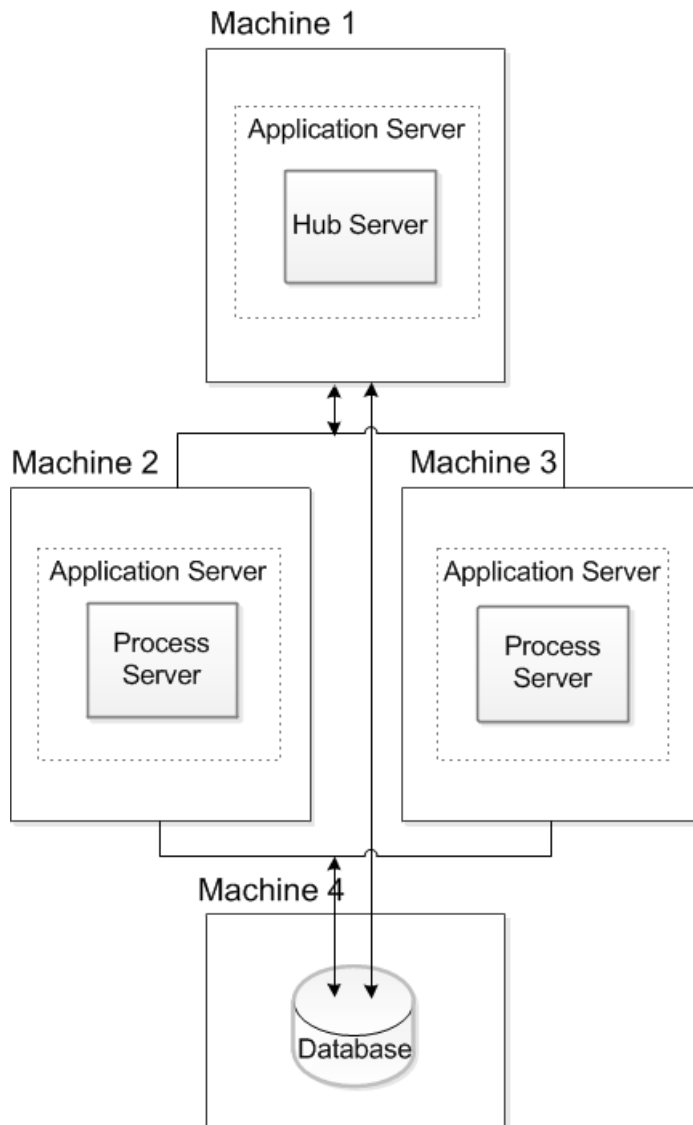


If you need high availability, you can configure additional Hub Server instances and configure external load balancers between the Hub Server instances.

The following image shows a sample multiple application server instances installation topology that is highly available:



External
Load
Balancer

Machine 1

Application Server

Hub Server

Machine 2

Application Server

Hub Server

Machine 3

Application Server

Process
Server

Machine 4

Application Server

Process
Server

Machine 5

Database

The following table describes the capabilities of the multiple application server instances topology:

| Capability | Availability |
|---|---|
| High availability | None.<br>**Note:** If you need high availability, you can configure additional Hub Server instances and configure external load balancers between the Hub Server instances. |
| Scalability | Yes.<br>To scale the MDM Hub to support large data volumes, add more MDM Hub components. Also, to process multiple requests concurrently, configure multiple threads for the Process Server.<br>The MDM Hub supports multithreading for the following operations and components:<br>- Hub Console operations<br>- Batch jobs<br>- Services Integration Framework (SIF) |
| Load balancing | Yes. The MDM Hub distributes the load between the available Process Server instances by using an internal load balancing mechanism.<br>The MDM Hub supports load balancing for the following operations:<br>- Hub Console operations<br>- All batch jobs except the Generate Match Tokens job<br>    **Note:** The MDM Hub supports load balancing for the fuzzy matching portion of the Match jobs and for the cleanse process portion of the Stage jobs.<br>The MDM Hub does not support load balancing for the following components:<br>- Informatica Data Director<br>- Services Integration Framework (SIF)<br>- Outbound JMS message queues |
| Maintainability | Difficult to maintain because the MDM Hub components are deployed on multiple machines. In environments where frequent changes are required, deployments and configurations need to be performed on each machine. |

# Application Server Cluster Topology

In an application server cluster topology, you install the MDM Hub components in an application server cluster. An application server cluster topology plan can be complicated because multiple combinations are possible. The main advantage of an application server cluster topology is the ease of deployment.

To set up an application server cluster topology, you need multiple machines with application server instances that are part of an application server cluster. Deploy the Hub Server and the Process Server instances on separate application server clusters. The application server cluster topology can provision for planned or unplanned down time. You can achieve scalability by adding nodes to the cluster and by deploying additional MDM Hub components.
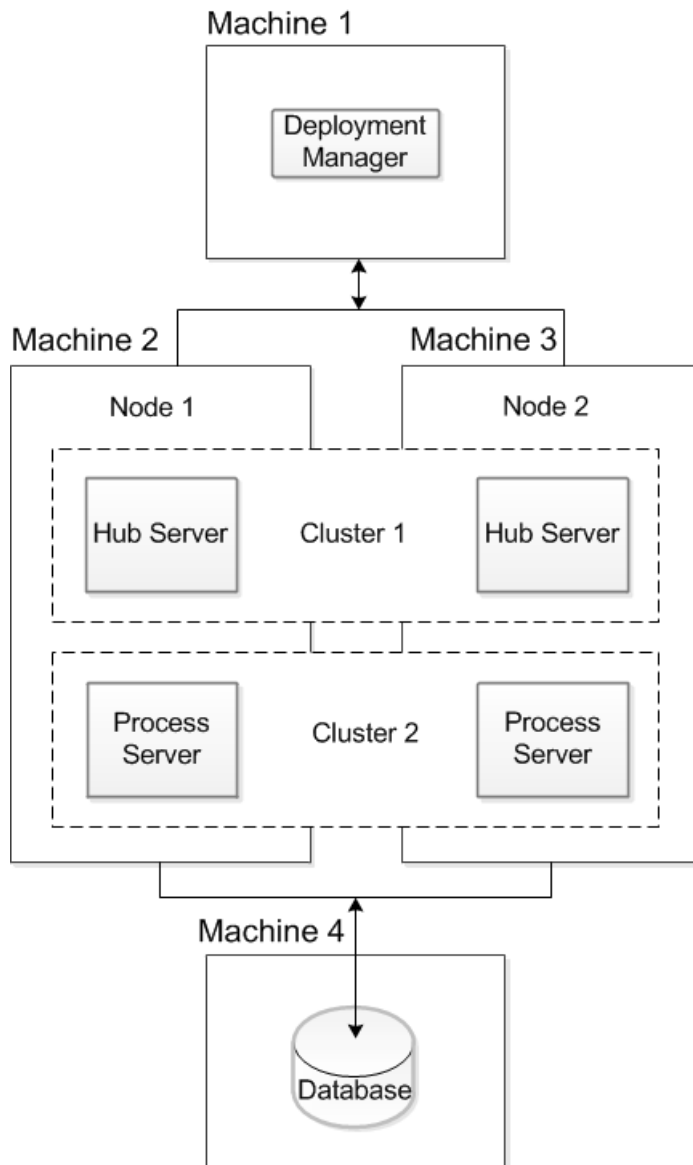
## Topology for WebSphere Clusters

The sample installation topology contains four machines with two WebSphere clusters. The WebSphere Deployment Manager can be installed on any machine, but in the sample it is on a separate machine to provide for secure WebSphere administration. Each WebSphere cluster includes the same two nodes. A Hub Server instance is deployed on each node of one cluster, so that if one node fails, the other node of the cluster can take over. A Process Server instance is deployed on each node of the second cluster, so that if one node fails, the other node of the cluster can take over.

The Hub Server distributes the processing load between the Process Server instances. If a Process Server instance fails or is offline, the Hub Server sends the processing request to the Process Server instance that is online. The Hub Store is configured on the fourth machine on which a database server is installed.

**Note:** You do not need to deploy the Process Server instances in a cluster. If you use JMS message queues, to consume outbound JMS messages, deploy the Hub Server instances in a cluster. Otherwise, each application server instance will have a different outbound JMS destination.

The following image shows a sample WebSphere cluster installation topology:

The following table describes the capabilities of the application server cluster topology:

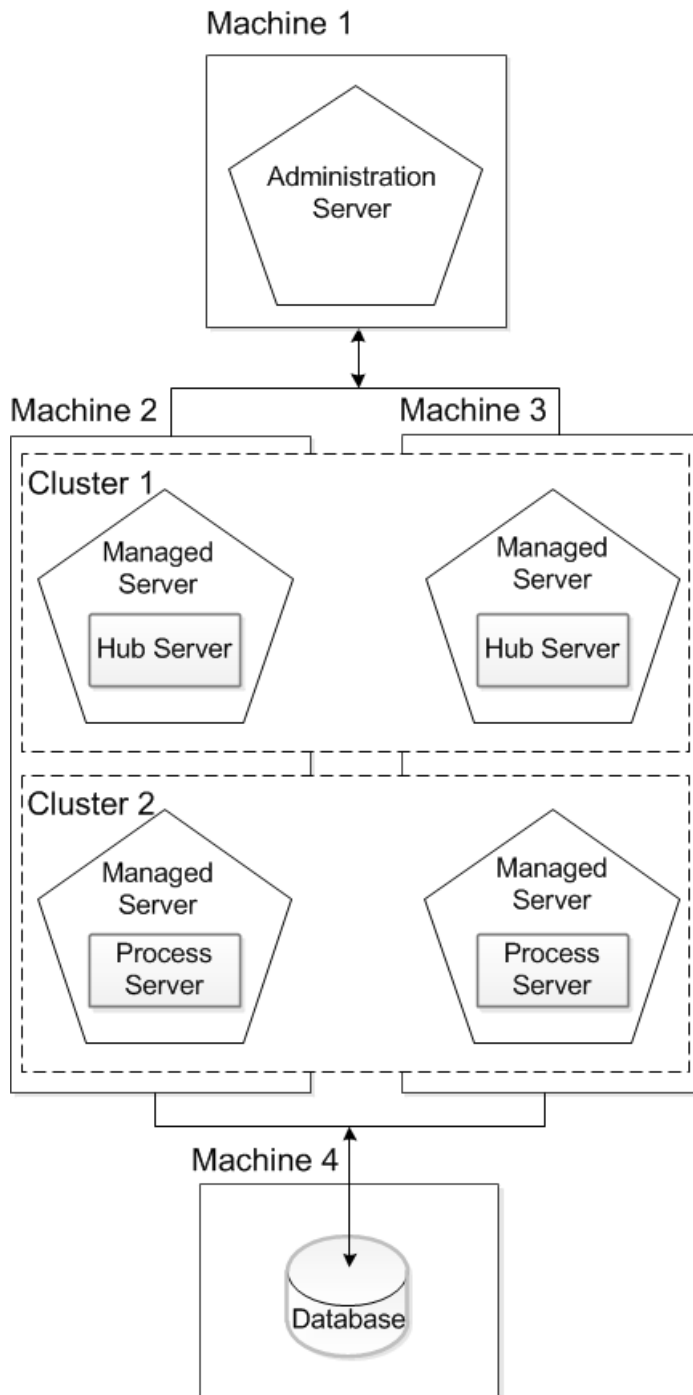| Capability | Availability |
|---|---|
| High availability | Yes.<br><br>The MDM Hub supports high availability for the following operations and components:<br>- Hub Console operations<br>- Services Integration Framework (SIF)<br>- Outbound JMS messages<br><br>The MDM Hub does not support high availability for the following operations and components:<br>- Batch jobs<br><br>**Note:** If a node in the cluster fails, the batch job requests that are initiated through the Hub Console fail over to the active node, but the batch jobs themselves do not fail over.<br>- Informatica Data Director |
| Scalability | Yes.<br><br>To scale the MDM Hub to support large data volumes, add more MDM Hub components. Also, to process multiple requests concurrently, configure multiple threads for the Process Server.<br><br>The MDM Hub supports multithreading for the following operations and components:<br>- Hub Console operations<br>- Batch jobs<br>- Services Integration Framework (SIF) |
| Load balancing | Yes. For load balancing, you do not need to deploy the Process Server instances on an application server cluster. The Process Server instances use an internal load balancing mechanism.<br><br>The MDM Hub supports load balancing for the following operations and components:<br>- Hub Console operations<br>- All batch jobs except the Generate Match Tokens job<br><br>**Note:** The MDM Hub supports load balancing for the fuzzy matching portion of the Match jobs and for the cleanse process portion of the Stage jobs.<br>- Services Integration Framework (SIF)<br>- Outbound JMS message<br>**Note:** Informatica Data Director does not support load balancing in an application server cluster. Load balancing in a clustered environment might produce unexpected results. To enhance the performance of the MDM Hub environment, you can use external load balancers. |
| Maintainability | More complicated than the standalone application server instance topology, but easier to deploy and maintain compared to the distributed application server topology. When you use the WebSphere Deployment Manager, it is easy to deploy the MDM Hub components across the nodes in a cluster. |

## Topology for WebLogic Clusters

The sample installation topology contains four machines with two WebLogic clusters. The WebLogic Administration Server can be installed on any machine, but in the sample it is on a separate machine to provide for secure WebLogic administration. Each WebLogic cluster includes the same two Managed Servers. A Hub Server instance is deployed on each Managed Server of one cluster, so that if one Managed Server fails, the other Managed Server of the cluster can take over. A Process Server instance is deployed on each Managed Server of the second cluster, so that if a Managed Server fails, the other Managed Server in the cluster can take over.

The Hub Server distributes the processing load between the two Process Server instances. If a Process Server instance fails or is offline, the Hub Server sends the processing request to the Process Server instance that is online. The Hub Store is configured on the fourth machine on which a database server is installed.

**Note:** You do not need to deploy the Process Server instances in a cluster. If you use JMS message queues, to consume outbound JMS messages, deploy the Hub Server instances in a cluster. Otherwise, each application server instance will a have different outbound JMS destination.

The following image shows a sample WebLogic cluster installation topology:



Machine 1

Administration Server

Machine 2

Machine 3

Cluster 1

Managed Server

Hub Server

Managed Server

Hub Server

Cluster 2

Managed Server

Process Server

Managed Server

Process Server

Machine 4

Database

The following table describes the capabilities of the application server cluster topology:

| Capability | Availability |
|---|---|
| High availability | Yes.<br><br>The MDM Hub supports high availability for the following operations and components:<br>- Hub Console operations<br>- Services Integration Framework (SIF)<br>- Outbound JMS messages<br><br>The MDM Hub does not support high availability for the following operations and components:<br>- Batch jobs<br><br>    **Note:** If a node in the cluster fails, the batch job requests that are initiated through the Hub Console fail over to the active node, but the batch jobs themselves do not fail over.<br>- Informatica Data Director |
| Scalability | Yes.<br><br>To scale the MDM Hub to support large data volumes, add more MDM Hub components. Also, to process multiple requests concurrently, configure multiple threads for the Process Server.<br><br>The MDM Hub supports multithreading for the following operations and components:<br>- Hub Console operations<br>- Batch jobs<br>- Services Integration Framework (SIF) |
| Load balancing | Yes. For load balancing, you do not need to deploy the Process Server instances on an application server cluster. The Process Server instances use an internal load balancing mechanism.<br><br>The MDM Hub supports load balancing for the following operations and components:<br>- Hub Console operations<br>- All batch jobs except the Generate Match Tokens job<br><br>    **Note:** The MDM Hub supports load balancing for the fuzzy matching portion of the Match jobs and for the cleanse process portion of the Stage jobs.<br>- Services Integration Framework (SIF)<br>- Outbound JMS message<br>**Note:** Informatica Data Director does not support load balancing in an application server cluster. Load balancing in a clustered environment might produce unexpected results. To enhance the performance of the MDM Hub environment, you can use external load balancers. |
| Maintainability | More complicated than the standalone application server instance topology, but easier to deploy and maintain compared to the distributed application server topology. When you use the WebLogic Administration Server, it is easy to deploy the MDM Hub components across the WebLogic Managed Servers in a cluster. |

## Topology for JBoss Clusters

The sample installation topology contains three machines with two JBoss clusters. Each JBoss cluster includes the same two nodes. A Hub Server instance is deployed on each node of one cluster, so that if one node fails, the other node of the cluster can take over. A Process Server instance is deployed on each node of the second cluster, so that if one node fails, the other node of the cluster can take over. The Hub Server distributes the processing load between the two Process Server instances. If a Process Server instance fails or is offline, the Hub Server sends the processing request to the Process Server instance that is online. The Hub Store is configured on the third machine on which a database server is installed.

**Note:** You do not need to deploy the Process Server instances in a cluster. You might want to deploy the Process Server instances in a cluster for ease of deployment, but each Process Server instance must be registered with the Hub Server. If you use JMS message queues, to consume outbound JMS messages, deploy the Hub Server instances in a cluster. Otherwise, each application server instance will have a different outbound JMS destination.

The following image shows a sample JBoss cluster installation topology:



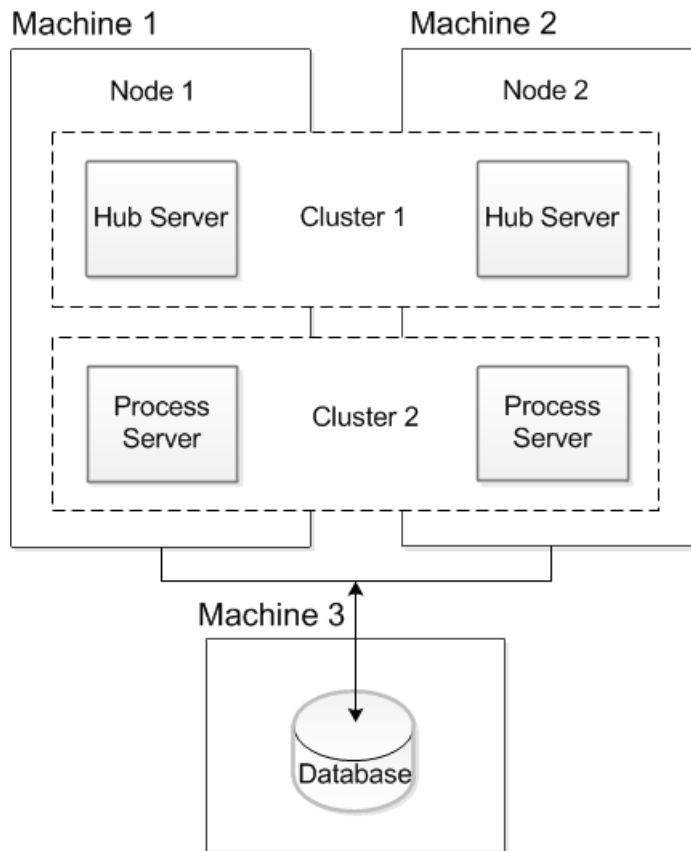The following table describes the capabilities of the application server cluster topology:

| Capability | Availability |
|---|---|
| High availability | Yes.<br><br>The MDM Hub supports high availability for the following operations and components:<br>- Hub Console operations<br>- Services Integration Framework (SIF)<br>- Outbound JMS messages<br><br>The MDM Hub does not support high availability for the following operations and components:<br>- Batch jobs<br><br>  **Note:** If a node in the cluster fails, the batch job requests that are initiated through the Hub Console fail over to the active node, but the batch jobs themselves do not fail over.<br>- Informatica Data Director |
| Scalability | Yes.<br><br>To scale the MDM Hub to support large data volumes, add more MDM Hub components. Also, to process multiple requests concurrently, configure multiple threads for the Process Server.<br><br>The MDM Hub supports multithreading for the following operations and components:<br>- Hub Console operations<br>- Batch jobs<br>- Services Integration Framework (SIF) |

| Capability | Availability |
| --- | --- |
| Load balancing | Yes. For load balancing, you do not need to deploy the Process Server instances on an application server cluster. The Process Server instances use an internal load balancing mechanism.<br><br>The MDM Hub supports load balancing for the following operations and components:<br>- Hub Console operations<br>- All batch jobs except the Generate Match Tokens job<br><br>    **Note:** The MDM Hub supports load balancing for the fuzzy matching portion of the Match jobs and for the cleanse process portion of the Stage jobs.<br>- Services Integration Framework (SIF)<br>- Outbound JMS message<br>**Note:** Informatica Data Director does not support load balancing in an application server cluster. Load balancing in a clustered environment might produce unexpected results. To enhance the performance of the MDM Hub environment, you can use external load balancers. |
| Maintainability | None.<br>**Note:** The MDM Hub supports the standalone mode for JBoss clusters. Unlike the domain mode clusters, the standalone mode clusters do not manage configuration and deployments. |