****

**DocShifter 7.1**

**High Availability Guide**

How to set up high availability with DocShifter 7.1



**Legal notice**

Copyright © 2021 Docshifter NV. All Rights Reserved.

**Disclaimer**

This documentation has been created for software version DocShifter 7.1 It is also valid for subsequent software versions as long as no new document version is published.

Docshifter NV believes the information in this publication is accurate as of its publication date.

All information in this publication is provided “as is” and is subject to change without notice. Docshifter NV makes no representations or warranties of any kind with respect to the information in this publication.

Use of any DocShifter software described in this publication requires an applicable software license.

For the most up-to-date listing of Docshifter solutions, visit [www.docshifter.com](http://www.docshifter.com).

All trademarks of other products used herein are the property of their respective owners.

DocShifter NV

Eedverbondkaai 242/003, B-9000, Gent, Belgium

[info@docshifter.com](mailto:info@docshifter.com)

**Table of Contents**

[1 PREFACE 4](#_Toc93962015)

[2 Configuration on WINDOWS 5](#_Toc93962016)

[2.1 Configuration of shared folders 5](#_Toc93962017)

[2.1.1 Introduction 5](#_Toc93962018)

[2.1.1.1 Configuration 5](#_Toc93962019)

[2.2 Configuration of DocShifterDB (PostgreSQL) 6](#_Toc93962020)

[2.2.1 Introduction 6](#_Toc93962021)

[2.2.2 Configuration of Primary Server Machine A 7](#_Toc93962022)

[2.2.2.1 Configuring replication 7](#_Toc93962023)

[2.2.3 Configuration of Replica Machine B (optionally C, D…) 8](#_Toc93962024)

[2.2.3.1 Configuring replication 8](#_Toc93962025)

[2.3 Starting the services 9](#_Toc93962026)

[3 Configuration on LINUX 10](#_Toc93962027)

[3.1 Configuration of Shared folders 10](#_Toc93962028)

[3.1.1 Introduction 10](#_Toc93962029)

[3.1.2 Configuration 10](#_Toc93962030)

[3.2 Configuration of DocShifterDB ( PostgreSQL) 11](#_Toc93962031)

[3.2.1 Introduction 11](#_Toc93962032)

[3.2.2 Configuration of Primary Server Machine A 12](#_Toc93962033)

[3.2.2.1 Configuring replication 12](#_Toc93962034)

[3.2.3 Configuration of Replica Machine B (optionally C, D…) 13](#_Toc93962035)

[3.2.3.1 Configuring replication 13](#_Toc93962036)

[3.2.4 Starting the daemons 14](#_Toc93962037)

[4 Configuration of DocShifter DB Healthy Checker (auto-failover) 15](#_Toc93962038)

[4.1 Introduction 15](#_Toc93962039)

[4.2 Precautions before proceeding 15](#_Toc93962040)

[4.3 Configuration of Machines A 16](#_Toc93962041)

[4.4 Configuration of Machines B 16](#_Toc93962042)

[5 Configuration of DocShifterMQ 17](#_Toc93962043)

[5.1.1 Introduction 17](#_Toc93962044)

[5.1.2 Precautions before proceeding 17](#_Toc93962045)

[5.1.3 Configuration of Machine A 18](#_Toc93962046)

[5.1.4 Configuration of Machine B 18](#_Toc93962047)

[5.1.5 Components Configuration 19](#_Toc93962048)

[6 Validating the distributed system 20](#_Toc93962049)

[6.1 Testing DocShifter DB & DocShifter DB Healthy Checker 20](#_Toc93962050)

[6.1.1 Sanity checks 20](#_Toc93962051)

[6.2 Testing DocShifterMQ 20](#_Toc93962052)

[6.2.1 Sanity checks 20](#_Toc93962053)

[6.2.2 Failover testing 21](#_Toc93962054)

[7 Hints & Tips 22](#_Toc93962055)

[7.1.1.1 Upgrading or Installing a module 22](#_Toc93962056)

[7.1.1.2 Importing a Workflow 22](#_Toc93962057)

# PREFACE

DocShifter is a multi-platform server application for the automation of document conversion into a wide variety of formats. DocShifter integrates with a large number of content management systems to provide optimum service in every circumstance.

DocShifter’s internal Message Queue supports the use of broker clusters: groups of brokers working together to provide message delivery services to clients. Clusters enable a message service to scale its operations to meet an increasing volume of message traffic by distributing client connections among multiple brokers. If a broker fails, clients connected to that broker reconnect to a failover broker that takes over the pending work of the failed broker, delivering messages without interruption of service. That way, no messages will be lost when a broker fails.

Another aspect is the synchronization of DocShifter settings across instances of a cluster. DocShifter uses PostgreSQL database to store this information: workflow definitions, queue connection information, preferences, ...

DocShifter recommends using PostgreSQL built-in replication mechanism to keep these databases synchronized across the cluster and DocShifter Database Healthy Checker as auto-failover

Moving to a high availability architecture will entail a dependence of DocShifter on external resources such as network, storage and database. This may affect overall DocShifter performance

# Configuration on WINDOWS

## Configuration of shared folders

### Introduction

DocShifter uses a temporary work- and error folder. Items that are being processed are stored in the work folder. If a failure occurs, these files are copied to an error folder.

These folders have to be shared in a high availability setup, making it possible for all instances of DocShifter to refer to the same work- and error folder.

In this section we’ll use the following machine configurations:

* **Machine A** (DocShifter instance 1): MACHINEA
* **Machine B** (DocShifter instance 2): MACHINEB
* **Sharedworkfolder**: \\HOST\sharedworkfolder
* **Sharederrorfolder**: \\ HOST\sharederrorfolder

#### Configuration

* Access the DocShifter web admin console on [http://localhost:8080/](http://localhost:8080/docshifter/static) .

*Default login and user name values are both* ***admin****.*

Graphical user interface, application

Description automatically generated

**Figure 1 DocShifter Dashboard**

* Click **Settings**.
* In **General Settings**, configure the location of both the work and error folder to the shared alternatives (always duplicating the slash character, e.g.: \\\\fs-0e21b910c09ac1060.com.docshifter\\share\\work)

Graphical user interface, text, application, email

Description automatically generated

**Figure 2 DocShifter Settings**

* Click **Save** to finish the process.

## Configuration of DocShifterDB (PostgreSQL)

### Introduction

For the replication of DocShifter's database, we recommend **PostgresSQL** **Streaming Replication Protocol**. Other replication solutions may be used, but have not been tested.

Replication of data will be unidirectional. Standby servers will connect to primary server to ask for updates from time to time.

The following configurations are used in this section:

**Machine A** (Primary instance): MACHINEA

**Machine B** (Standby replica of Machine A): MACHINEB

**[Optional]**

**Machine C** (Standby replica of Machine A): MACHINEC

**Machine D** (Standby replica of Machine A): MACHINED

Since Machine B (and optionally Machine C, D, …) will replicate the data of Machine A, preparation of Machine B (and C, D, …) will be our first step.

**Precautions before proceeding**

Make sure all machines can find each other based on hostname and IP.

Make sure the following ports are opened on the firewalls:

**5432** – used by DocShifter DB.de www.docshifter

### Configuration of Primary Server Machine A

#### Configuring replication

1. Stop the following DocShifter services in order: DocShifter Sender, DocShifter Console, DocShifter Receiver, DocShifter MQ, DocShifter Metrics, DocShifterDBHealthyChecker, DocShifter DB.
2. Add the following line in pg\_hba.conf file located at **{DS Install Path}\docshifter-db\data\pg\_hba.conf**

*host all all <ip-machine-A>/32 md5*

*host all all <ip-machine-B>/32 md5*

*host replication docshifter\_cluster <ip-machine-A>/32 md5*

*host replication docshifter\_cluster <ip-machine-B>/32 md5*

***\*Replace placeholders <ip-machine-\*> with the hosts local ip (e.g.: 10.110.0.90)***

1. Edit the following line in postgresql.conf file located at **{DS Install Path}\docshifter-db\data\postgresql.conf**

*Replace #listen\_address with listen\_addresses = '\*'*

*Replace #max\_connections with max\_connections = 1000*

*Replace # primary\_conninfo with primary\_conninfo=user=docshifter\_cluster password=docshifter\_cluster channel\_binding=prefer host=<ip-machine-B> port=5432'*

***\*Replace placeholders <ip-machine-\*> with the hosts local ip (e.g.: 10.110.0.90)***

*Replace #promote\_trigger\_file with promote\_trigger\_file = '<shared drive DB folder>\unhealthy.signal'* (duplicate the slash characters)

***\*Replace placeholder <shared driver> with the docshifter-db shared folder path (e.g.: \\fs-0e21b910c09ac1060.com.docshifter\share\docshifter\docshifter-db)***

*Replace #logging\_collector with logging\_collector = on*

*Replace #log\_filename with log\_filename = 'postgresql-%Y-%m-%d.log'*

1. Create a backup of primary server located

Start the DocShifter DB service.

Run the command **{DS Install Path}\docshifter-db\windows\bin\**pg\_basebackup.exe -D <shared drive DB folder>\server1 -U postgres

Provide the DB password when requested. (The same defined during the DocShifter installation process)

Run the command **{DS Install Path}\docshifter-db\windows\bin\**pg\_basebackup.exe -D <shared drive DB folder>\server2 -U postgres

***\*Replace placeholder <shared driver DB folder> with the docshifter-db shared folder path (e.g.: \\fs-0e21b910c09ac1060.com.docshifter\share\docshifter\docshifter-db)***

Provide the DB password when requested. (The same defined during the DocShifter installation process)



Run the **{DS Install Path}\docshifter-db\windows**\**unregister**.**bat** to remove the service

Text

Description automatically generated

Run **{DS Install Path}\docshifter-db\windows**\**register.bat <shared drive DB folder>\server1** (e.g.: \\fs-0e21b910c09ac1060.com.docshifter\share\docshifter\docshifter-db\server1) to create a new service pointing to the data folder located at a network drive

1. Configuration of DocShifter Components

Replace in all components (console,message-broker,metrics,receiver,sender) the database URL located at the application.properties as following.

spring.datasource.url=jdbc:postgresql://*MACHINEA*:5432,*MACHINEB*:5432/docshifter?targetServerType=primary

spring.datasource.metrics.url=jdbc:postgresql://*MACHINEA*:5432,*MACHINEB*:5432/metrics?targetServerType=primary

*# You should replace the placeholders (MACHINEA/MACHINEB) with the local machines IP address.*

### Configuration of Replica Machine B (optionally C, D…)

#### Configuring replication

1. Stop the following DocShifter services in order: DocShifter Sender, DocShifter Console, DocShifter Receiver, DocShifter MQ, DocShifter Metrics, DocShifterDBHealthyChecker, DocShifter DB.
2. Edit the following line in postgresql.conf file located at **{network-datatabase-server2-folder}\postgresql.conf**

*Replace # primary\_conninfo with primary\_conninfo=user=docshifter\_cluster password=docshifter\_cluster channel\_binding=prefer host=<ip-machine-A> port=5432'*

***\*Replace placeholders <ip-machine-\*> with the hosts local ip (e.g.: 10.110.0.90)***

1. Start the Docshifter DB service.
2. Run the **{DS Install Path}\docshifter-db\windows**\**unregister**.**bat** to remove the service

Text

Description automatically generated

1. Run **{DS Install Path}\docshifter-db\windows**\**register.bat <shared drive DB folder>\server2** (e.g.: \\fs-0e21b910c09ac1060.com.docshifter\share\docshifter-db\server2) to create a new service pointing to the data folder located at a network drive.
2. Create a "standby.signal" file on the **<shared drive DB folder>\server2** folder.



1. Configuration of DocShifter Components

Replace in all components (console, message-broker, metrics, receiver and sender) the database URL located at the application.properties as following:

spring.datasource.url=jdbc:postgresql://*MACHINEA*:5432,*MACHINEB*:5432/docshifter?targetServerType=primary

spring.datasource.metrics.url=jdbc:postgresql://*MACHINEA*:5432,*MACHINEB*:5432/metrics?targetServerType=primary

*# You should replace the placeholders (MACHINEA/MACHINEB) with the machines IP address.*

## Starting the services

Start all DocShifter services in order on both machine A and machine B:

* DocShifter MQ
* DocShifter DB
* DocShifter Receiver
* DocShifter Console
* DocShifter Sender
* DocShifter DB Healthy Checker

# Configuration on LINUX

## Configuration of Shared folders

### Introduction

DocShifter uses a temporary work- and error folder. Items that are being processed are stored in the work folder. If a failure occurs, these files are copied to an error folder.

In a high availability setup these folders have to be shared, making it possible for all instances of DocShifter to refer to the same work- and error folder. In order to make a filesystem accessible, make sure that you have mounted it.

In this section we’ll use the following machine configurations:

**Machine A** (DocShifter instance 1): MACHINEA

**Machine B** (DocShifter instance 2): MACHINEB

**Sharedworkfolder**: //MACHINEC/sharedworkfolder

**Sharederrorfolder**: //MACHINEC/sharederrorfolder

### Configuration

1. Access the DocShifter web admin console on [http://localhost:8080/](http://localhost:8080/docshifter/static).

*Default login and user name values are both* ***admin****.*

Graphical user interface, application

Description automatically generated

**Figure 3 DocShifter Dashboard**

* Click **Settings**.
* In **General Settings**, configure the location of both the work and error folder to the shared alternatives.

Graphical user interface, text, application, email

Description automatically generated

**Figure 4 DocShifter Settings**

* Click **Save** to finish the process.
* Repeat these steps on Machine B.

## Configuration of DocShifterDB ( PostgreSQL)

### Introduction

For the replication of DocShifter's database, we recommend **PostgresSQL** **Streaming Replication Protocol**. Other replication solutions may be used, but have not been tested.

Replication of data will be unidirectional. Standby servers will connect to primary server to ask for updates from time to time.

The following configurations are used in this section:

**Machine A** (Primary instance): MACHINEA

**Machine B** (Standby replica of Machine A): MACHINEB

**[Optional]**

**Machine C** (Standby replica of Machine A): MACHINEC

**Machine D** (Standby replica of Machine A): MACHINED

Since Machine B (and optionally Machine C, D, …) will replicate the data of Machine A, preparation of Machine B (and C, D, …) will be our first step.

**Precautions before proceeding**

Make sure all machines can find each other based on hostname and IP.

Make sure the following ports are opened on the firewalls:

**5432** – used by DocShifter DB. guide www.docshifter

### Configuration of Primary Server Machine A

#### Configuring replication

1. Stop the following DocShifter daemons, in order, using systemctl: docshifter-db.service, DocShifterMQ, DocShifterReceiver,DocShifterSender, DocShifterConsole, DocShifterMetrics,DocShifterDBHealthyChecker.
2. Add the following line in pg\_hba.conf file located at **{DS Install Path}/docshifter-db/** **/data/pg\_hba.conf**

*host all all <ip-machine-A>/32 md5*

*host all all <ip-machine-B>/32 md5*

*host replication docshifter\_cluster <ip-machine-A>/32 md5*

*host replication docshifter\_cluster <ip-machine-B>/32 md5*

1. Edit the following line in postgresql.conf file located at **{DS Install Path}/docshifter-db/data/postgresql.conf**

*Replace #listen\_address with listen\_addresses = '\*'*

*Replace #max\_connections with max\_connections = 1000*

*Replace # primary\_conninfo with primary\_conninfo=user=docshifter\_cluster password=docshifter\_cluster channel\_binding=prefer host=<ip-machine-B> port=5432'*

*Replace #promote\_trigger\_file with promote\_trigger\_file = '<shared drive DB folder>*[*\\unhealthy.signal*](file:///\\unhealthy.signal)*'* (duplicate the slash characters)

*Replace #logging\_collector with logging\_collector = on*

*Replace #log\_filename with log\_filename = 'postgresql-%Y-%m-%d.log'*

1. Create a backup of primary server located

Run the command **{DS Install Path}/docshifter-db/linux/bin/**pg\_basebackup -D <//networkdrive/server1> -U postgres

Provide the DB password when requested. (The same defined during the DocShifter installation process)

Run the command **{DS Install Path}/docshifter-db/linux/bin/**pg\_basebackup -D <//networkdrive/server2> -U postgres

Provide the DB password when requested. (The same defined during the DocShifter installation process)

1. Edit the file /etc/systemd/system/docshifter-db.service

Replace the old old database datafolder set in ExecStart property with the new database data folder.

Run sudo systemctl daemon-reload

1. Configuration of DocShifter Components

Replace in all components (console,message-broker,metrics,receiver,sender) the database URL located at the application.properties as following.

spring.datasource.url=jdbc:postgresql://*MACHINEA*:5432,*MACHINEB*:5432/docshifter?targetServerType=primary

spring.datasource.metrics.url=jdbc:postgresql://*MACHINEA*:5432,*MACHINEB*:5432/metrics?targetServerType=primary

*# You should replace the placeholders (MACHINEA/MACHINEB) with the machines IP address.*

### Configuration of Replica Machine B (optionally C, D…)

#### Configuring replication

1. Stop the following DocShifter daemons, in order, using systemctl: docshifter-db.service, DocShifterMQ, DocShifterReceiver,DocShifterSender, DocShifterConsole, DocShifterMetrics,DocShifterDBHealthyChecker.
2. Edit the following line in postgresql.conf file located at **{network-datatabase-data-folder} /postgresql.conf**

*Replace # primary\_conninfo with primary\_conninfo=user=docshifter\_cluster password=docshifter\_cluster channel\_binding=prefer host=<ip-machine-A> port=5432'*

1. Edit the file /etc/systemd/system/docshifter-db.service
2. Replace the old old database datafolder set in ExecStart property with the new database data folder.
3. Create a "standby.signal" file on the standby servers data folder.



1. Run sudo systemctl daemon-reload
2. Configuration of DocShifter Components

Replace in all components (console,message-broker,metrics,receiver,sender) the database URL located at the application.properties as following.

spring.datasource.url=jdbc:postgresql://*MACHINEA*:5432,*MACHINEB*:5432/docshifter?targetServerType=primary

spring.datasource.metrics.url=jdbc:postgresql://*MACHINEA*:5432,*MACHINEB*:5432/metrics?targetServerType=primary

*# You should replace the placeholders (MACHINEA/MACHINEB) with the machines IP address.*

### Starting the daemons

Start the following DocShifter daemons, in order, on Machine A and Machine B (and optionally additional Replica machines as needed) using systemctl: docshifter-db.service, DocShifterMQ, DocShifterReceiver,DocShifterSender, DocShifterConsole, DocShifterMetrics,DocShifterDBHealthyChecker

# Configuration of DocShifter DB Healthy Checker (auto-failover)

## Introduction

We define high availability as the ability for the system to continue functioning after failure of one or more of the servers.

A part of high availability is failover which we define as the ability for client connections to migrate from one server to another in event of server failure so DocShifter components can continue to operate.

DocShifter DB Healthy Checker keeps checking if the primary server is alive every 10 seconds. If the primary is not alive, it broadcast an "unhealthy.signal" one of the standby servers will receive the broadcast and became the new live server and will mark the old primary server as a standby. So, if the old primary is brought back it starts as standby.

Before failover, only the live server is serving the DocShifter components while the standby servers remain passive or awaiting to become the live server. When a live server crashes or is brought down in the correct mode, the backup server currently in passive mode will become live.

The following configurations are used in this section:

**Primary Server** (Live Server)

**Secondary Server** (Standby Server)

**[Optional]**

**Tertiary Server** (Standby server)

**Quaternary Server** (Standby server)

## Precautions before proceeding

* Make sure that all machines can find each other based on their hostname and IP addresses.
* Make sure that there is a proper shared folder.

## Configuration of Machines A

* Edit DocShifter DB Healthy Checker application.properties
* Open the application.properties located at **{DS Install Path}\db-healthy-checker**
* data.folder=*=***<shared drive DB folder>\\server2** (duplicate the slash characters) (e.g.: \\\\fs-0e21b910c09ac1060.com.docshifter\\share\\docshifter\\docshifter-db\\server1)

*# data folder property should point to the database data folder. The healthy checker will keep watching that folder to check if he is still the primary machine.*

* unhealthy.folder=*=***<shared drive DB folder>** (duplicate the slash characters) (e.g.: \\\\fs- e21b910c09ac1060.com.docshifter\\share\\docshifter\\docshifter-db)

*# unhealthy folder property should point to a network driver folder. The healthy checker will create a unhealthy.signal on that folder when he notice the primary server is not running.*

* cluster.machines=**<MACHINE-B>**

*# cluster machines property you need to specify the servers that will be part of the cluster separating by comma, excluding itself and must be the ip address*

## Configuration of Machines B

* Edit DocShifter DB Healthy Checker application.properties
* *Open the application.properties located at* ***{DS Install Path}\db-healthy-checker***
* *data.folder=***<shared drive DB folder>\\server1** (duplicate the slash characters) (e.g.: \\\\fs-0e21b910c09ac1060.com.docshifter\\share\\docshifter\\docshifter-db\\server1)

*# data folder property should point to the database data folder. The healthy checker will keep watching that folder to check if he is still the primary machine.*

* *unhealthy.folder==***<shared drive DB folder>** (duplicate the slash characters) (e.g.: \\\\fs-0e21b910c09ac1060.com.docshifter\\share\\docshifter\\docshifter-db)

*# unhealthy folder property should point to a network driver folder. The healthy checker will create a unhealthy.signal on that folder when he notice the primary server is not running.*

* *cluster.machines=*=**<MACHINE-A>**

*# cluster machines property you need to specify the servers that will be part of the cluster separating by comma, excluding itself and must be the ip address.*

# Configuration of DocShifterMQ

### Introduction

We define high availability as the ability for the system to continue functioning after failure of one or more of the servers.

A part of high availability is failover which we define as the ability for client connections to migrate from one server to another in event of server failure so DocShifter components can continue to operate.

DocShifter MessageMQ allows servers to be linked together as *live - backup* groups where each live server can have 1 or more backup servers. A backup server is owned by only one live server. Backup servers are not operational until failover occurs, however 1 chosen backup, which will be in passive mode, announces its status and waits to take over the live servers’ work

Before failover, only the live server is serving the DocShifter MessageMQ components while the backup servers remain passive or awaiting to become a backup server. When a live server crashes or is brought down in the correct mode, the backup server currently in passive mode will become live and another backup server will become passive. If a live server restarts after a failover then the backup server will detect the live server coming back up and automatically stop.

The following configurations are used in this section:

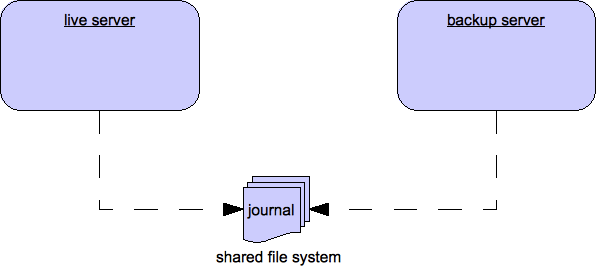
**Primary Server** (Live Server)

**Secondary Server** (Backup Server)

**[Optional]**

**Tertiary Server** (Backup server)

**Quaternary Server** (Backup server)



**Figure 1 DocShifterMQ HA Policy**

### Precautions before proceeding

* Make sure that all machines can find each other based on their hostname.
* Make sure that there is a proper fileshare.
* Make sure the following **ports** are opened on the firewalls:
  + **61617**: used by high availability and failover mechanism.
  + **61616**: used by DocShifter components.

### Configuration of Machine A

Set up the queues for High Availability by changing the following properties in the **{DS Install Path}\message-broker\application.properties**:

**The data directory property must be a shared folder.**

**Create a MQ data directory on the shared drive.** (*e.g.: \\fs-0e21b910c09ac1060.com.docshifter\share\docshifter\mq)*

***docshifter.data.directory****= <MQ shared drive data folder> (e.g.: \\fs-0e21b910c09ac1060.com.docshifter\share\docshifter\mq)*

**Set to true to enable HA mechanism.**

***docshifter.enable.cluster****=true*

**When primary server property is true that means this will be your live server.**

***docshifter.primary.server****=true*

**Specify other servers that will be part of the cluster separating by comma.**

***docshifter.cluster.machines****= <FULL MACHINE A NAME>,<FULL MACHINE B NAME>*

***\*Replace placeholders <FULL MACHINE X NAME> with the hosts full name (e.g.: ip-10-110-0-140.com.docshifter)***

### Configuration of Machine B

Set up the queues for High Availability by changing the following properties in the **{DS Install Path}\message-broker\application.properties**:

**The data directory property must be a shared folder.**

***docshifter.data.directory****=<MQ shared drive data folder> (e.g.: \\fs-0e21b910c09ac1060.com.docshifter\share\docshifter\mq)*

**Set to true to enable ha mechanism.**

***docshifter.enable.cluster****=true*

**When primary server property is false that means this will be a backup server.**

***docshifter.primary.server****=false*

**Specify other servers that will be part of the cluster separating by comma.**

***docshifter.cluster.machines****= <FULL MACHINE A NAME>,<FULL MACHINE B NAME>*

**\*Replace placeholders <FULL MACHINE X NAME> with the hosts full name *(e.g.: ip-10-110-0-140.com.docshifter)***

### Components Configuration

* Access the DocShifter web admin console on [http://localhost:8080/](http://localhost:8080/docshifter/static) from the primary server.

*Default login and user name values are both* ***admin****.*

* Click **Settings**.
* In **General Settings**, configure the MQ URL using the following pattern:

***(tcp://{live server hostname}:61616,tcp://{backup server hostname}:61616)?ha=true***

* Click **Save** to finish the process.
* **Reboot all machines.**

# Validating the distributed system

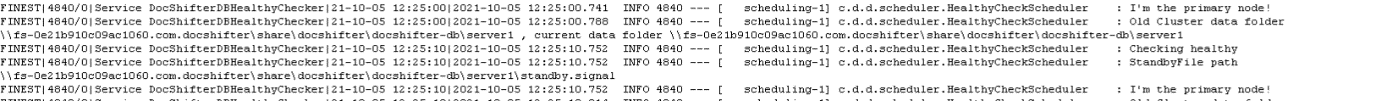
## Testing DocShifter DB & DocShifter DB Healthy Checker

### Sanity checks

DocShifter DB / Healthy Checker can be tested by following these steps:

* open the admin console on the live server
* create a new workflow
* open the admin console on the slave server
* the newly created workflow should be visible on the other servers
* **Live server** **DocShifter DB Healthy Checker logs message ({DS Install Path}\yajsw\logs\docshifter-db-healthy-checker.log)**

The following message means that live server has been started and is working correctly.



* **Backup server logs message ({DS Install Path}\yajsw\logs\docshifter-db-healthy-checker.log)**



## Testing DocShifterMQ

### Sanity checks

* **Live server logs message ({DS Install Path}\yajsw\logs\message-broker.log)**

The following message means that live server has been started and is working correctly.

INFO [main] (FileLockNodeManager.java:241) - AMQ221034: Waiting indefinitely to obtain live lock

INFO [main] (FileLockNodeManager.java:245) - AMQ221035: Live Server Obtained live lock

INFO [main] (SharedStoreLiveActivation.java:98) - AMQ221007: Server is now live

* **Backup server logs message ({DS Install Path}\yajsw\logs\message-broker.log)**

INFO [AMQ229000: Activation for server ActiveMQServerImpl::serverUUID=f37b38e7-55e6-11eb-9684-02c55dffb87e] (SharedStoreBackupActivation.java:90) - AMQ221109: Apache ActiveMQ Artemis Backup Server version 2.16.0 [f37b38e7-55e6-11eb-9684-02c55dffb87e] **started, waiting live to fail before it gets active**

* **Extra backup servers log message**

INFO [AMQ229000: Activation for server ActiveMQServerImpl::serverUUID=f37b38e7-55e6-11eb-9684-02c55dffb87e] (FileLockNodeManager.java:223) - AMQ221032: **Waiting to become backup node**

### Failover testing

**DocShifterMQ HA** can be tested by following these steps:

1. Stop the primary server.
2. Checks if backup server became a live server and the extra backup server became the backup.
3. Send a document to be processed.
4. The document should be processed successfully.
5. Stop the backup server.
6. Checks if the extra backup server became the live server.
7. Send a document to be processed.
8. The document should be processed successfully.
9. Start Primary server.
10. Checks if the actual liver server (extra backup) became a backup server and primary server became the live server.
11. Start the backup server and checks if he becomes the backup server taking place of the extra backup.

**Do not forget to start the services again after completing these tests.**

# Hints & Tips

#### Upgrading or Installing a module

E.g : 3 DocShifter servers, HiFi module M1.2 / M1.3 version.

Before starting: Please stop DocShifterSender, DocShifterReceiver on all servers.

**On Server 1:**

* Open DocShifter Admin Console
* Login and choose Upload Module from the Dashboard.
* Click on Select a module.
* Use the file chooser to select the zip file downloaded from the DocShifter Portal.
* Click on Upload.
* Allow the system to perform the upload. Enter your DocShifter Admin Console login credentials if prompted.
* Click OK when the process is complete.
* Go to .../DocShifter/receiver/modules.
* Delete any older HiFi Module jars. In this case it will be the M1.2 version.
* Take a copy of the new HiFi Module jar (M1.3 version).
* Start DocShifterReceiver

**On Server 2:**

* Delete any older HiFi Module jars. In this case it will be the M1.2 version.
* Paste the new HiFi Module jar (M1.3) copied from Server 1 to .../DocShifter/receiver/modules.
* Start DocShifterReceiver

**On Server 3:**

* Delete any older HiFi Module jars. In this case it will be the M1.2 version.
* Paste the new HiFi Module jar (M1.3) copied from Server 1 to .../DocShifter/receiver/modules.
* Start DocShifterReceiver

**On all 3 Servers:**

* Start DocShifterSender

#### Importing a Workflow

When you need to import a workflow, just import in one of the servers and shared PostgresDB will replicate the workflow on the others servers.

**THANK YOU !**

**CONTACT US**

DocShifter NV

Eedverbondkaai 242/003

9000 Gent

Belgium

+32 (09) 242 87 30

info@docshifter.com

www.docshifter.com