Fabric Controller
Installation Manual

Version 1.0

October. 2017

Table of Contents

Revision History

Version	Date	Contents
1.0	Oct. 20, 2017	First edition

1. Overview

This manual describes the installation method and the redundancy settings of fabric controller (FC) (hereinafter it is referred to as “FC”).

1.1 Relevant manuals

Please refer to the following manuals or information found online as necessary when installing or using FC.

- Linux “CentOS(http://www.centos.org/)”.
- RDBMS “PostgreSQL (http://www.postgresql.org/)”.
- Redundancy “Pacemaker (http://clusterlabs.org/)”.

1.2 Trademark Notice

2. Execution environment

2.1 FC configuration

Configuration of FC is shown in Figure 2-1 Configuration of FC

.

FC configuration is illustrated in blue box.

FC can be executed both in stand-alone configuration and redundancy configuration.

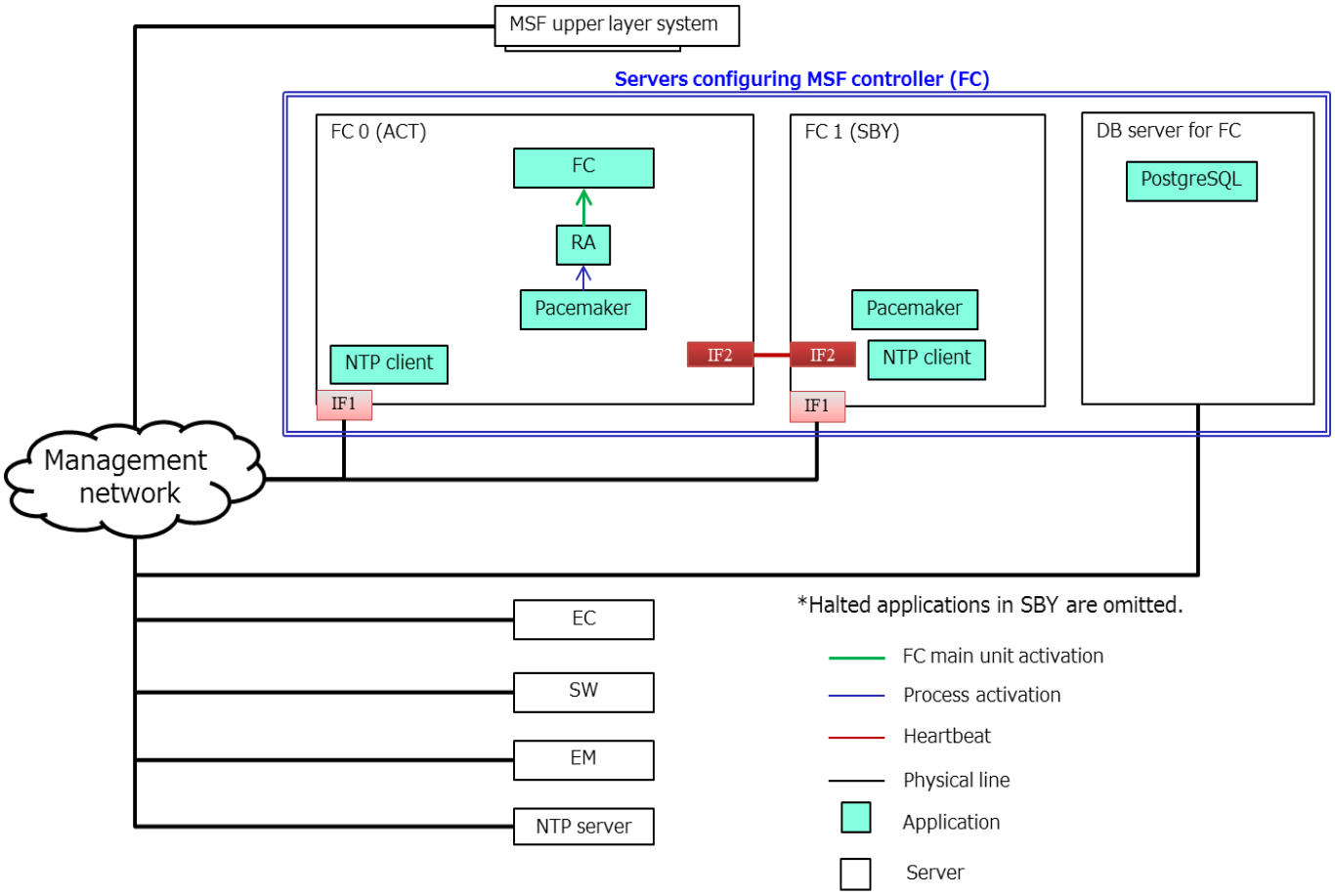

Figure 2-1 Configuration of FC

2.2 Hardware requirements

FC is executed on PC which mounts x86 CPU. Minimum hardware requirements are shown in Table 2-1 Minimum hardware requirements.

Table 2-1 Minimum hardware requirements

No.	Item	Specification
1	CPU	2 Cores or more
2	Memory	1 GB or more
3	Hard disk	Available disk size 20 GB or more
4	NIC	1 port or more (2 ports or more when FC is redundancy configuration)

Hardware configuration when operability confirmed on FC are shown in Table 2-2 Hardware configuration when operability confirmed as reference.

Table 2-2 Hardware configuration when operability confirmed

No.	Item	Specification
1	CPU	Xeon E5-2420v2 @ 2.20 GHz 6 Cores/12 Threads
2	Memory	32 GB
3	Hard disk	600 GB
4	NIC	4 ports

2.3 Software requirements

Software requirements are shown in Table 2-3.

Use particular version of software which listed on Table 2-3 for Java libraries excluding JAXB.

Table 2-3 Software requirements

No.	Item	Software	Version	Software is available from
1	OS	CentOS	7 (1511)	http://www.centos.org/ Download file: CentOS-7-x86_64-DVD-1511.iso
2	Middleware	Java	Oracle JDK 8 Update 101 or later (Oracle JDK 9 or later is not applicable)	http://www.oracle.com/ Download file: jdk-8u101-linux-x64.rpm
		Pacemaker	1.1.14-1	http://linux-ha.osdn.jp/wp/dl Download file: pacemaker-repo-1.1.14-1.1.el7.x86_64.rpm
		Corosync	2.3.5-1	It is included in installation package of Pacemaker.
3	Java library	Jetty	9.3.11	http://archive.eclipse.org/jetty/9.3.11.v20160721/dist/ Download file: jetty-distribution-9.3.11.v20160721.tar.gz
		Gson	2.7	https://repo1.maven.org/maven2/com/google/code/gson/gson/2.7/ Download file: gson-2.7.jar

No.	Item	Software	Version	Software is available from
		JAXB	2.2.8	It is included in Java.
		Jersey	2.23.2	http://repo1.maven.org/maven2/org/glassfish/jersey/bundles/jaxrs-ri/2.23.2/ Download file: jaxrs-ri-2.23.2.tar.gz
		Hibernate	5.0.10	https://sourceforge.net/projects/hibernate/files/hibernate-orm/5.0.10.Final/hibernate-release-5.0.10.Final.tgz/download Download file: hibernate-release-5.0.10.Final.tgz
		SLF4J	1.6.1	https://sourceforge.net/projects/unirods/files/lib/slf4j-nop-1.6.1.jar/download?use_mirror=ayera&download=&failedmirror=kent.dl.sourceforge.net Download file: slf4j-nop-1.6.1.jar * Additional library for Hibernate
		Log4J	2.6.2	http://archive.apache.org/dist/logging/log4j/2.6.2 Download file: apache-log4j-2.6.2-bin.tar.gz
		Apache Commons IO	2.5	https://commons.apache.org/proper/commons-io/download_io.cgi Download file: commons-io-2.5-bin.tar.gz
		Hipster4j	1.0.1	http://www.hipster4j.org/ Download file: citiususc-hipster-v1.0.1-0-g09cf69c.zip
		JDBC	9.4.1209	https://jdbc.postgresql.org/download/ Download file: postgresql-9.4.1209.jar

3. Installation method of FC

This manual describes the settings of FC server, FC main unit and DB for FC which are illustrated in Figure 2-1.

3.1 Installation method of libraries

This manual refers to a work user as general user “msfctrl”. Replace this with appropriate user based on your environment.

3.1.1 OS settings

(1) Add a work user to wheel group.

```
$ su -
# gpasswd -a msfctrl wheel
# id msfctrl
----------------------------------------
uid=1000(msfctrl) gid=1000(msfctrl) groups=1000(msfctrl),10(wheel)
----------------------------------------
```

(2) Execute visudo command and grant sudo access to users in the group wheel.

```
# visudo
----------------------------------------
## Allows people in group wheel to run all commands
%wheel    ALL=(ALL)           ALL
----------------------------------------
# exit
```

* Login again after these settings.

(3) Disable SELinux.

```
$ sudo vi /etc/sysconfig/selinux


----------------------------------------
SELINUX=disabled
----------------------------------------
```

(4) Reflect the settings of SELinux (Reboot the OS to take effect).

```
$ sudo shutdown -r now
```

(5) Disable firewall.

```
$ sudo systemctl disable firewalld.service
$ sudo systemctl stop firewalld.service
```

(6) Synchronize time with NTP server.

* Install NTP client software to FC server to perform time synchronization.

(7) Install unzip tool.

* This is necessary to unzip Java libraries provided as zip files.

3.1.2 Java installation

This section gives the instructions for installing the JDK and deploying Java libraries.
In the example below, Java libraries are placed under "~/java_lib/".
Directory configuration under "~/java_lib" is shown below.

/home/msfctrl	Home directory for work user
└-- java_lib	Preparation directory for Java library
├-- gson	Store directory for Gson library
├-- hipster4j	Store directory for Hipster4j library
├-- jetty	Store directory for Jetty library
├-- postgresql	Store directory for JDBC library
├-- apache-commons	Store directory for Apache Commons IO library
├-- hibernate	Store directory for Hibernate library
├-- jersey	Store directory for Jersey library
└-- log4j	Store directory for Log4J library

3.1.2.1 JDK installation

(1) Download rpm file of JDK from http://www.oracle.com/.

* In the example below, the downloaded rpm file is referred to as "jdk-8u101-linux-x64.rpm".

* Use JDK 8 update 101 or later version. However, JDK 9 or later version is not applicable for current FC server.

(2) Set the downloaded rpm file to the arbitrary location on FC server.
In the example below, the rpm file is set under "~/rpm/".

(3) Change the working directory to the rpm directory and install JDK.

```
$ cd ~/rpm/
$ sudo rpm -ivh jdk-8u101-linux-x64.rpm
```

(4) Set Java version as default.

* Select a number of /usr/java/jdk1.8.0_101/jre/bin/java.

```
$ sudo alternatives --config java
```

3.1.2.2 Set Gson library

(1) Download gson-2.7.ja from https://repo1.maven.org/maven2/com/google/code/gson/gson/2.7/.

(2) Set the downloaded gson-2.7.jar to the arbitrary location on FC server.
In the example below, the file is set under "~/download/".

Place gson-2.7.jar under "~/java_lib/gson/".

```
$ cd ~/download/
$ mv gson-2.7.jar ~/java_lib/gson/
```

3.1.2.3 Set Hipster4j library

Compile Hipster4j from source by using Apache Maven.
Therefore, install Apache Maven for compile.

(1) Download apache-maven-3.5.0-bin.tar.gz from https://maven.apache.org/download.cgi.

(2) Set the downloaded apache-maven-3.5.0-bin.tar.gz to the arbitrary location on FC server.
In the example below, the file is set under "~/download/".

(3) Unzip apache-maven-3.5.0-bin.tar.gz and set an environment variable to make it available (in the example below, the environment is configured in ~/download/apache-maven-3.5.0.)

```
$ cd ~/download/
$ tar xvfz apache-maven-3.5.0-bin.tar.gz
$ cd ~/
$ vi .bash_profile
Add the following.
-----
export M3_HOME=/home/msfctrl/download/apache-maven-3.5.0
M3=$M3_HOME/bin
export PATH=$M3:$PATH
---
```

* Logout and login again to reflect the environment variable after editing .bash_profile.

(4) Download citiususc-hipster-v1.0.1-0-g09cf69c.zip from http://www.hipster4j.org/.

(5) Set the downloaded citiususc-hipster-v1.0.1-0-g09cf69c.zip to the arbitrary location on FC server.
In the example below, the file is set under "~/download/".

(6) Unzip citiususc-hipster-v1.0.1-0-g09cf69c.zip and generate a jar file in the unzipped directory by using Apache Marven. Set generated hipster-all-1.0.1-all.jar under "~/java_lib/hipster4j/".

```
$ cd ~/download/
$ unzip citiususc-hipster-v1.0.1-0-g09cf69c.zip
$ cd citiususc-hipster-09cf69c/
$ mvn package
$ cd ~/download/citiususc-hipster-09cf69c/hipster-all/target/
$ mv hipster-all-1.0.1-all.jar ~/java_lib/hipster4j/
```

3.1.2.4 Set Jetty library

(1) Download jetty-distribution-9.3.11.v20160721.tar.gz from
http://archive.eclipse.org/jetty/9.3.11.v20160721/dist/.

(2) Set the downloaded jetty-distribution-9.3.11.v20160721.tar.gz to the arbitrary location on FC server.
In the example below, the file is set under "~/download/".

(3) Unzip jetty-distribution-9.3.11.v20160721.tar.gz and place a lib directory in the unzipped directory under "~/java_lib/jetty/".

```
$ cd ~/download/
$ tar xvfz jetty-distribution-9.3.11.v20160721.tar.gz
$ cd jetty-distribution-9.3.11.v20160721/
$ mv lib ~/java_lib/jetty/
```

3.1.2.5 Set JDBC library

(1) Download postgresql-9.4.1209.jar from https://jdbc.postgresql.org/download/postgresql-9.4-1209.jar.

(2) Set the downloaded postgresql-9.4.1209.jar to the arbitrary location on FC server.
In the example below, the file is set under "~/download/".

(3) Set postgresql-9.4.1209.jar under "~/java_lib/postgresql".

```
$ cd ~/download/
$ mv postgresql-9.4.1209.jar ~/java_lib/postgresql/
```

3.1.2.6 Set Apache Commons IO library

(1) Download commons-io-2.5-bin.tar.gz from

https://commons.apache.org/proper/commons- io/download_io.cgi.

(2) Set the downloaded commons-io-2.5-bin.tar.gz to the arbitrary location on FC server.
In the example below, the file is set under "~/download/".

(3) Unzip commons-io-2.5-bin.tar.gz and place commons-io-2.5.jar in the unzipped directory under "~/java_lib/apache-commons/".

```
$ cd ~/download/
$ tar xvfz commons-io-2.5-bin.tar.gz
$ cd commons-io-2.5/
$ mv commons-io-2.5.jar ~/java_lib/apache-commons/
```

3.1.2.7 Set Hibernate library

Prepare a main unit and additional library for Hibernate library.

(1) Download hibernate-release-5.0.10.Final.tgz from
https://sourceforge.net/projects/hibernate/files/hibernate-orm/5.0.10.Final/hibernate-release-5.0.10.Final.tgz/download.

(2) Set the downloaded hibernate-release-5.0.10.Final.tgz to the arbitrary location on FC server.
In the example below, the file is set under "~/download/".

(3) Unzip hibernate-release-5.0.10.Final.tgz and place a lib directory in the unzipped directory under "~/java_lib/hibernate/".

```
$ cd ~/download/
$ tar xvfz hibernate-release-5.0.10.Final.tgz
$ cd hibernate-release-5.0.10.Final/
$ mv lib ~/java_lib/hibernate/
```

(4) Add a necessary library to Hibernate.
Download slf4j-nop-1.6.1.jar from https://sourceforge.net/projects/unirods/files/lib/slf4j-nop-1.6.1.jar/download?use_mirror=ayera&download=&failedmirror=kent.dl.sourceforge.net.

(5) Set the downloaded file to the arbitrary location on FC server.
In the example below, the file is set under "~/download/".

(6) Set slf4j-nop-1.6.1.jar under "~/java_lib/hibernate/lib/optional/ehcache".

```
$ cd ~/download/
$ mv slf4j-nop-1.6.1.jar ~/java_lib/hibernate/lib/optional/ehcache/
```

3.1.2.8 Set Jersey library

(1) Download jaxrs-ri-2.23.2.tar.gz from
http://repo1.maven.org/maven2/org/glassfish/jersey/bundles/jaxrs-ri/2.23.2/.

(2) Set the downloaded jaxrs-ri-2.23.2.tar.gz to the arbitrary location on FC server.
In the example below, the file is set under "~/download/".

(3) Unzip jaxrs-ri-2.23.2.tar.gz and place all files and directories in the unzipped directory under "~/java_lib/jersey/".

```
$ cd ~/download/
$ tar xvfz jaxrs-ri-2.23.2.tar.gz
$ cd jaxrs-ri/
$ mv * ~/java_lib/jersey/
```

3.1.2.9 Set Log4J library

(1) Download apache-log4j-2.6.2-bin.tar.gz from http://archive.apache.org/dist/logging/log4j/2.6.2.

(2) Set the downloaded apache-log4j-2.6.2-bin.tar.gz to the arbitrary location on FC server.
In the example below, the file is set under "~/download/".

(3) Unzip apache-log4j-2.6.2-bin.tar.gz and place log4j-1.2-api-2.6.2.jar, log4j-api-2.6.2.jar, log4j-core-2.6.2.jar in the unzipped directory under "~/java_lib/log4j/".

```
$ cd ~/download/
$ tar xvfz apache-log4j-2.6.2-bin.tar.gz
$ cd apache-log4j-2.6.2-bin/
$ mv log4j-1.2-api-2.6.2.jar ~/java_lib/log4j/
$ mv log4j-api-2.6.2.jar ~/java_lib/log4j/
$ mv log4j-core-2.6.2.jar ~/java_lib/log4j/
```

3.2 FC installation

FC installation methods are described below.

3.2.1 Directory configuration on FC server

Directory configuration of FC (main files and store directories for libraries) is shown below.
Installation directory is referred to as "~/msf-controller/" in this manual.

/home/msfctrl	Home directory for work user
└-- msf-controller	Directory for installation
| -- bin	Script store directory for FC activation and shutdown
| ├--fc_ctl.sh	Script for FC shutdown
| └--fc	Resource agent for FC
├-- lib	FC main unit and store directories for libraries
| ├-- FabricController.jar	FC main unit
| ├-- gson	Store directory for Gson library
| ├-- hipster4j	Store directory for Hipster4j library
| ├-- jetty	Store directory for Jetty library
| ├-- postgresql	Store directory for JDBC library
| ├-- apache-commons	Store directory for Apache Commons IO library
| ├-- hibernate	Store directory for Hibernate library
| ├-- jersey	Store directory for Jersey library
| └-- log4j	Store directory for Log4J library
├-- logs	Log directory *1
└--conf	Configuration directory
├-- fc_system.xml	FC system setting configuration
├-- fc_data.xml	FC initial setting configuration
├-- fc_develop.xml	FC internal operability setting configuration
├-- log4j2.xml	Log4J configuration
└-- hibernate.cfg.xml	Hibernate configuration

*1. Log directory is automatically generated when FC starts running. It does not exist at the time of FC installation.

3.2.2 FC main unit installation

(1) Set tar file (msf-controller.tar.gz) of FC to the installation directory.

(2) Unzip tar file of FC.

* To reinstall the FC server, first, delete all directories in the installed directory extracted from the archive before. Then extract the archive to it again.

```
$ cd ~/msf-controller/
$ tar xvfz ~/msf-controller/msf-controller.tar.gz
```

(3) Copy Java libraries under “lib” of FC.

* For FC reinstallation, leave the original directory which includes Java libraries.

```
$ cp -r ~/java_lib/* ~/msf-controller/lib/
```

(4) Grant a permission to the script file (fc_ctl.sh) in the unzipped directory.

```
$ cd ~/msf-controller/bin/
$ chmod 755 fc_ctl.sh
```

3.3 FC settings

This section describes the settings of FC.

3.3.1 Preparation of initial data

In order to initialize FC, initial data needs to be prepared for registration to the DB. Config of collection of data for initial registration to the DB is FC initial setting Config. Basically, FC initial setting Config can be changed in the following cases: when starting the FC server for the first time or initializing the FC. Edit FC initial setting Config while FC is not running. Note that any changes of FC initial setting Config while FC is running cannot be reflected in the behavior of FC.

3.3.1.1 Brief description of FC initial setting Config

FC initial setting Config is shown briefly in Table 3-1.

Table 3-1 Brief description of FC initial setting Config

No.	Config file name	Config Name	Location
1	fc_data.xml	FC initial configuration Config	In "*FC install directory*/conf/"

3.3.1.2 Contents of FC initial setting Config

This section describes the parameters in FC initial setting Config.

FC initial setting Config file is in xml format.

An example of FC initial setting Config is shown below.

Parameters in FC initial setting Config are summarized in Table 3-2.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<dataConf xmlns="http://fc.msf/common/config/type/data">
    <swClustersData>
        <swClusterData>
            <swCluster>
                <swClusterId>1</swClusterId>
                <maxLeafNum>45</maxLeafNum>
                <maxSpineNum>4</maxSpineNum>
                <asNum>65000</asNum>
                <rpLoopbackAddress>1.1.1.1</rpLoopbackAddress>
                <interfaceStartAddress>1.1.1.1</interfaceStartAddress>
                <loopbackStartAddress>1.1.1.1</loopbackStartAddress>
                <managementStartAddress>1.1.1.1</managementStartAddress>
                <managementAddressPrefix>24</managementAddressPrefix>
            </swCluster>
            <rrs>
                <rr>
                    <rrNodeId>1</rrNodeId>
                    <rrRouterId>1.1.1.1</rrRouterId>
                </rr>
            </rrs>
        </swClusterData>
    </swClustersData>
    <slice>
        <ipv4MulticastAddressBase>239.0.1.0</ipv4MulticastAddressBase>
    </slice>
</dataConf>
```

Table 3-2 List of parameters in FC initial setting Config

Element	Type	Setting range	Number of elements	Description
dataConf	-	-	1	-
swClustersData	-	-	1	-
swClusterData	-	-	1	Equivalent to information of a SW cluster per an element
swCluster	-	-	1	Information of SW cluster
swClusterId	integer	1 - 100	1	ID of SW cluster
maxLeafNum	integer	1 - 1000	1	Maximum number of Leaves in SW cluster
maxSpineNum	integer	1 - 1000	1	Maximum number of Spines in SW cluster
asNum	integer	0 - 65535	1	AS number of SW cluster
rpLoopbackAddress	string	-	1	RP loopback address of SW cluster (only in IPv4 format)
interfaceStartAddress	string	-	1	Start IP address of SW cluster interface address (only in IPv4 format) A prefix of assigned interface address is fixed to 30. Interface address of SW cluster is an IP address set to an internal link IF in a node (Leaf and Spine), which is managed over FC. Initiating with start IP address, when interface (node) is added an suitable IP address is calculated automatically within FC and is assigned.
loopbackStartAddress	string	-	1	Start IP address of SW cluster loopback address (only in IPv4 format) A prefix of assigned loopback address is fixed to 32. Loopback address of SW cluster is an loopback address of a node (Leaf and Spine) , which is managed over FC.

Element					Type	Setting range	Number of elements	Description
								Initiating with start IP address, when interface (node) is added an suitable IP address is calculated automatically within FC and is assigned.
				managementStartAddress	string	-	1	Management start IP address of SW cluster Management IP address is an IP address of management IF in a node, which is managed over FC. Initiating with start IP address, when interface (node) is added an suitable IP address is calculated automatically within FC and is assigned.
				managementAddressPrefix	integer	1 - 32	1	Prefix for management address of SW cluster
			rrs		-	-	0 and over	Equivalent to a record of RR information of DB
				rr	-	-	1	-
				rrNodeId	integer	1 - 65535	1	RR node ID of SW cluster
				rrRouterId	string	-	1	RR router ID of SW cluster (only in IPv4 format)
		slice			-	-	1	-
			ipv4MulticastAddressBase		string	-	1	Base address of IPv4 multicast address for L2VPN (only in IPv4 format)

3.3.2 Settings during operation

This section describes the settings during FC operation. Edit FC system setting Config while FC is not running. Note that any changes of FC system setting Config while FC is running cannot be reflected in the behavior of FC.

3.3.2.1 Brief description of FC system setting Config

The brief description of FC system setting Config is shown in Table 3-3.

Table 3-3 Brief description of system setting Config

No.	Config file name	Name of Config	Location
1	fc_system.xml	FC system setting Config	In "*FC install directory*/conf/"

3.3.2.2 Contents of FC system setting Config

This section describes the parameters in FC system setting Config.

FC system setting Config file is in xml format.

An example of FC system setting Config is shown below.

Parameters in FC system setting Config are summarized in Table 3-4.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<systemConf xmlns="http://fc.msf/common/config/type/system">
    <rest>
        <server>
            <listeningAddress>0.0.0.0</listeningAddress>
            <listeningPort>18080</listeningPort>
        </server>
        <client>
            <waitConnectionTimeout>30</waitConnectionTimeout>
            <requestTimeout>600</requestTimeout>
        </client>
        <json>
            <isPrettyPrinting>true</isPrettyPrinting>
            <isSerializeNulls>false</isSerializeNulls>
        </json>
    </rest>
    <slice>
        <l2SlicesMagnificationNum>1</l2SlicesMagnificationNum>
        <l3SlicesMagnificationNum>1</l3SlicesMagnificationNum>
        <l2MaxSlicesNum>100</l2MaxSlicesNum>
        <l3MaxSlicesNum>100</l3MaxSlicesNum>
    </slice>
    <swClustersData>
        <swClusterData>
            <swCluster>
                <swClusterId>1</swClusterId>
                <ecControlAddress>0.0.0.0</ecControlAddress>
                <ecControlPort>18080</ecControlPort>
            </swCluster>
        </swClusterData>
    </swClustersData>
    <traffic>
        <interval>0</interval>
        <dataRetentionPeriod>7</dataRetentionPeriod>
        <tmToolPath>/opt/fc/tm/</tmToolPath>
        <tmInputFilePath>/opt/fc/tm/input/</tmInputFilePath>
        <tmOutputFilePath>/opt/fc/tm/output/</tmOutputFilePath>
    </traffic>
</systemConf>
```

Table 3-4 List of parameters in FC system setting Config

Element	Type	Setting range *	Number of elements	Description
systemConf	-	-	1	-
rest	-	-	1	-
server	-	-	1	-
listeningAddress	string	-	1	REST listening interface address
listeningPort	integer	0 - 65535	1	REST listening port
client	-	-	1	-
waitConnectionTimeout	integer	0 and over	1	REST wait connection timeout (second)
requestTimeout	integer	0 and over	1	REST request timeout (second)
json	-	-	1	-
isPrettyPrinting	boolean	true/false	1	Whether JSON data that FC responds to REST request is formatted in such a way of line breaking and indenting or not (true/ false)
isSerializeNulls	boolean	true/false	1	In case there is a null value in JSON data that FC respondes to REST request, whether parameter is described as null or not (true/ false)
slice	-	-	1	-
l2SlicesMagnificationNum	integer	1,2	1	Parameter that controls maxmum value of assigned L2 slice ID. Assigned range of L2 slice ID is equal to l2MaxSlicesNum multiplied by l2SlicesMagnificationNum. Example 1: If l2MaxSliceNum is 100 and

Element	Type	Setting range *	Number of elements	Description
				l2SliceMagnificationNum is 1, then assigned range of L2 slice ID is from 1 to 100. Example 2: If l2MaxSliceNum is 100 and l2SliceMagnificationNum is 2, then assigned range of L2 slice ID is from 1 to 200.
l3SlicesMagnificationNum	integer	1,2	1	Parameter that controls maxmum value of assigned L3 slice ID. Assigned range of L3 slice ID is equal to l3MaxSlicesNum multiplied by l3SlicesMagnificationNum. Example 1: If l3MaxSliceNum is 100 and l3SliceMagnificationNum is 1, then assigned range of L3 slice ID is from 1 to 100. Example 2: If l3MaxSliceNum is 100 and l3SliceMagnificationNum is 2, then assigned range of L3 slice ID is from 1 to 200.
l2MaxSlicesNum	integer	1 - 1000	1	Maximum L2 slice number
l3MaxSlicesNum	integer	1 - 1000	1	Maximum L3 slice number
swClustresData	-	-	1	-
swClusterData	-	-	1	-
swCluster	-	-	1	-
swClusterId	integer	1 - 100	1	SW cluster ID
ecControlAddress	string	-	1	EC control address
ecControlPort	integer	0 - 65535	1	EC port number
traffic	-	-	1	Not covered with this version

Element	Type	Setting range *	Number of elements	Description
interval	integer	Fixed to 0	1	-
dataRetentionPeriod	integer	-	1	-
tmToolPath	string	-	1	-
tmInputFilePath	string	-	1	-
tmOutputFilePath	string	-	1	-

* If the upper limit of setting range of an attribute is not described, its upper limit is the maximum number of integer type (2147483647).

3.3.3 DB settings

The initial setting of DB is described below. It is necessary to set DB in DB server for FC. Make sure that PostgreSQL is installed on the DB server and you can login as the “postgres” user.

* The FC server and the DB server for FC are hosted in different machines.
* Any users can work on DB server for FC.
* This section describes the operations over DB server for FC.

(1) Initialize a database cluster.

```
$ sudo su - postgres
$ initdb --no-locale -E UTF-8
$ exit
```

(2) Start PostgreSQL.

```
$ sudo systemctl start postgresql-9.3.service
```

(3) Create a new role and a new database.

* The role name and the role password designated here shall be set in hibernate Config for FC.
To set hibernate Config (hibernate.cfg.xml), please refer to the official page of hibernate.

```
$ sudo -u postgres psql
postgres=# create role role name with login password 'role password';
postgres=# create database msf_fc LC_COLLATE 'en_US.UTF-8' LC_CTYPE 'en_US.UTF-8'
TEMPLATE template0;
postgres=# ¥q
```

(4) Create a table and a table schema.

* Set “msf_fc.sql” under “~/” in the DB server for FC in advance, which is included in the tar file that was unzipped in section 3.2.2 .

```
$ cd ~/
$ sudo -u postgres psql -U role name -d msf_fc < msf_fc.sql
```

3.3.4 FC simple startup confirmation

This section gives the simple FC startup confirmation methods.

In the case of FC redundancy configuration, FC server shutdown is executed by Pacemaker. To configure redundancy settings, please refer to section 4. in this document, while it is advised for Pacemaker to refer to the official page of it.

(1) Change FC Config (FC initial setting Config, FC system setting Config, and Hibernate Config) appropriately.

(2) Execute the following command after changing to the directory where FC shutdown script resides.

```
$ sh fc_ctl.sh start
```

Command prompts for both FC normal startup and FC startup failed are described here.

In the case of FC normal startup, the command prompt is as follows.
(Nothing is shown except for the prompt "$".)

```
$
```

In the case of FC startup failed, the command prompt is as follows.

```
ERROR. foo
$
```

* "foo" explains either why FC can't startup correctly or what the status of FC is.
When FC cannot startup correctly, it would be likely that either the Config setting or network setting is not correctly configured.

(3) Shutdown FC.
Change the working directory to the directory where FC shutdown script resides and execute the following command.

```
$ sh fc_ctl.sh stop
```

Command prompts for both FC normal shutdown and FC shutdown failed are described here.

In the case of FC normal shutdown, the command prompt is as follows.

```
INFO. FabricController stopped.
$
```

In the case of FC shutdown failed, the command prompt is as follows.

```
ERROR. bar
$
```

* "bar" explains either why FC cannot shutdown correctly or what the status of FC is.

3.3.5 Status confirmation with FC shutdown script

This section gives the procedure for status confirmation with FC shutdown.

This procedure gives status confirmation of FC startup, whether a process is running ot not.

Change the working directory to the directory where FC shutdown script resides and then execute the following command.

```
$ sh fc_ctl.sh status
```

FC startup status can be determined based on the result of the above command (standard output log). The standard output logs are summarized in the Table 3-5.

Table 3-5 FC startup status

FC startup status	Standard output log	Remarks
Running (normal)	INFO. FabricController[pid=PID] is running.	PID shows actual pid of FC.
Running (abnormal: double activation)	WARN. FabricController is running (two or more applications are running).	More than or equal to 2 FCs are running. In order to get back normal condition, it is necessary to shutdown the FC that started up by mistake. In this case, since it is not possible to shutdown FC with FC shutdown script, kill command is required to shutdown the FC.
Not running	INFO. FabricController is not running.	-

3.3.6 FC status confirmation (normal/abnormal)

This section describes how to validate FC status.

In order to validate FC status, the REST message in Table 3-6 needs to be sent to management address of the FC.

Table 3-6 REST request for status confirmation

method	URI
GET	/v1/MSFcontroller/status

FC status is validated by the response code of the REST message sent above. However, if there is no response, either the FC isn't running or there are transmission errors.

Table 3-7 summarizes response codes.

Table 3-7 List of response codes of status confirmation

Response code	Status of FC
200	normal
500	abnormal

When the response code is 200, relevant body field of its response consists of the following information shown in Table 3-8.

Table 3-8 Response body field of status confirmation

body parameter name	Brief description of body parameter	body parameter value	Brief description of body parameter
service_status	Service status	start-up in progress	Start-up is in progress
		running	Running
		shutdown in progress	Shutdown is in progress
		system switching	System switching is in progress
blockade_status	Maintenance blockade status	blockade	Under blockade
		none	No blockade

4. Redundancy settings

FC redundancy is executed by Pacemaker. This section describes the FC redundancy settings. To install Pacemaker and make the basic settings, please refer to the official page of Pacemaker.

4.1 Placement of resource agent

Resource agent of FC is placed in a designated directory.

Resource agent of FC resides in msf-controller.tar.gz.

File name of resource agent of FC is “fc”.

(1) Resource agent of FC is placed in 0-system node and 1-system node. Note that this execution needs to be carried out with root privileges.

```
# cp -p ~/msf-controller/bin/fc /usr/lib/ocf/resource.d/heartbeat/fc
```

(2) Grant an execution right and an ownership (root) to the resource agent file of FC in 0-system node and 1-system node.

```
# cd /usr/lib/ocf/resource.d/heartbeat/
# chmod 755 fc
# chown root:root fc
```

4.2 Pacemaker settings

This section describes how to configure Pacemaker with some examples.

4.2.1 Property settings of cluster

Property settings of cluster are shown in Table 4-1.

FC invalidates the STONITH function.

Table 4-1 Settings of cluster property

Item	Value	Brief description	Remarks
no-quorum-policy	Ignore	Resource assignment based on the number of nodes	-
stonith-enabled	false	Handling of failed node (STONITH control)	STONITH is not used.

4.2.2 Default settings of resource

Default settings of resource are shown in Table 4-2.

Table 4-2 Default settings of resource

Item	Value	Brief description	Remarks
resource-stickiness	INFINITY	Resource assignment	No automatic failback
migration-threshold	1	Number of allowable failed conditions	Failover with one failed event

4.2.3 Configuration settings of resource

Configuration settings of resource are shown in Table 4-3.

Settings of various resource agents are explained in section 4.2.4 , 4.2.5 , 4.2.6 , and 4.2.7 for reference.

Table 4-3 Configuration settings of resource

Resource agent	Type of resource	Brief description	Necessity of group settings	Activation sequence within group (reverse sequence for shutdown)
VIPcheck	Primitive	Virtual IP check	YES	1
Resource agent for FC	Primitive	FC process monitoring		2
IPaddr2	Primitive	Assignment of virtual IP		3
diskd	Clone	Monitoring of internal disk	NO	-

4.2.4 Settings of VIPcheck

Example of OCF parameters settings of VIPcheck are shown in Table 4-4.

Table 4-4 Examples of OCF parameters settings of VIPcheck

OCF parameter	Brief description	Settings example
target_ip	Virtual IP to be checked	192.168.10.100
count	Number of checks	1
wait	Waiting time (second)	10

Examples of settings for operation of VIPcheck are shown in Table 4-5.

Table 4-5 Examples of settings for operation of VIPcheck

Operation	Timeout value (second)	Monitoring interval (second)	Behavior with failed conditions	Start-up delay time (second)

Operation	Timeout value (second)	Monitoring interval (second)	Behavior with failed conditions	Start-up delay time (second)
start	90	0	restart	6

4.2.5 Settings of resource agent of FC

Table 4-6 explains OCF parameter of FC resource agent with setting examples.

Table 4-6 Brief description and setting examples of FC resource agent

OCF parameter	Brief description	Setting example
host_0	Host name of 0-system of FC	FC1-1
host_1	Host name of 1-system of FC	FC1-2
fc_system_xml	Location of FC system setting Config	/home/msfctrl/msf-controller /conf/fc_system.xml
fc_ctl	Location of FC shutdown script	/home/msfctrl/msf-controller /bin/fc_ctl.sh
fc_username	User name of FC	msfctrl

Configuration examples of operations of FC resource agent are described in Table 4-7.

Table 4-7 Setting examples of operations of FC resource agent

Operation	Timeout value (second)	Monitoring interval (second)	Behavior with failed conditions	Start-up delay time (second)
start	60	0	restart	No settings
monitor	60	60	restart	No settings
stop	3600	0	block	No settings

4.2.6 Settings of IPaddr2

Table 4-8 explains OCF parameter of IPaddr2 with configuration examples.

Table 4-8 Setting examples of OFC parameter of IPaddr2

OCF parameter	Brief description	Configuration example
ip	Virtual IP address	192.168.10.100
nic	IF that sets virtual IP address	enp0s99
cidr_netmask	Subnetmask of virtual IP address	24

Setting examples of operations of IPaddr2 are described in Table 4-9.

Table 4-9 Setting examples of operations of IPaddr2

Operation	Timeout value (second)	Monitoring interval (second)	Behavior with failed conditions	Start-up delay time (second)
start	60	0	restart	No settings
monitor	60	10	restart	No settings
stop	60	0	ignore	No settings

4.2.7 Settings of diskd

Table 4-10 explains OCF parameter of diskd with setting examples.

Table 4-10 Setting examples of OFC parameter of diskd

OCF parameter	Brief description	Setting example
name	Monitoring object name	diskcheck_status_internal
device	Disk name of monitoring object	/dev/sda1
interval	Monitoring interval (second)	10

Setting examples of operations of diskd are described in Table 4-11.

Table 4-11 Setting examples of operations of diskd

Operation	Timeout value (second)	Monitoring interval (second)	Behavior with failed conditions	Start-up delay time (second)
start	60	0	restart	No settings
monitor	60	10	restart	No settings
stop	60	0	ignore	No settings

4.2.8 Settings of several constraints

This manual describes the settings of resource constraints with grpFC, which is making resource ID grouped, and clnDiskd for Clone resource among resource settings shown in Table 4-3.

Settings are described based on the official Pacemaker file “pm_crmgen_env.xls”.

4.2.8.1 Settings of resource placement constraints

Setting examples of resource placement constraints are described in Table 4-12.

Table 4-12 Setting examples of resource placement constraints

Resource ID	Score	bool op (and/or)	Conditions attribute name	Conditions	Conditions value	Role
grpFC	-inf	or	diskcheck_status_internal	not_defined		No settings
				eq	ERROR	No settings

4.2.8.2 Settings of resource colocation constraints

Setting examples of resource colocation constraints are described in Table 4-13.

Table 4-13 Setting examples of resource colocation constraints

Control related Resource ID	Resource ID of Control object	Score (Weighting)	Role of constraint related resource	Role of constraint object resource
grpFC	clnDiskd	inf	No settings	No settings

4.2.8.3 Settings of resource start-up sequence constraints

Setting examples of resource start-up sequence constraints are described in Table 4-14.

Table 4-14 Setting examples of resource start-up sequence constraints

Starter resource ID	Subsequent resource ID	Score (Weighting)	Starter resource action	Subsequent resource action	Shutdown sequence
clnDiskd	grpFC	0	No settings	No settings	Same sequence as start-up