FIXEdge Failover Cluster installation (based on Logs replicator)

- Stimulus
- Input
  - System requirements
  - Virtual IP
  - Cluster credentials
  - Nodes layout
  - Additional artifacts
- Per node configuration
  - FIXEdge setup
  - Pacemaker related
  - Additional configuration
- Cluster setting up
  - Nodes registration
  - Resources configuration
  - Resources constraints
  - Ending setup
  - Health check
- Notes
  - What next

Stimulus

The purpose of this document is to describe setting up demo failover cluster of **FIXEdge** isntances, with state replication implemented via **b2b_replication (aka Logs replicator).**

Input

System requirements

- At least CentOS Linux release 7.0.1406
- At least FIXEdge-5.10.0.70626-FA-2.13.2.70568-Linux-2.6.32
- epel enabled
- internet connection

Virtual IP

10.11.132.199 ip address is an access point. Configured FIXEdge instance will be available through this IP.

Such resource will be laid out in **eth0** interface of each node.

Cluster credentials

The following credentials were used for cluster authorization:

login: hacluster

password: epm-bfix

Nodes layout

There are two nodes (the following two nodes will be used in the instructions below):

- ECSE00100034.epam.com
- ECSE00100035.epam.com

Additional artifacts

Modified version of script anything(https://github.com/ClusterLabs/resource-agents/blob/master/heartbeat/anything) should be used. Modified script anything attached,

it's compatible with standard version, but contains modifications which allow to use services pid file instead automatically created by script anything.

Per node configuration

Steps from this part should be performed on **both** nodes.

FIXEdge setup

FIXEdge setup

```
$ sudo yum install -y java
$ mv FIXEdge-5.10.0.70626-FA-2.13.2.70568-Linux-2.6.32-gcc447-x86_64.tar.gz /home/user/FIXEdge.tgz
$ cd /home/user
$ tar -xzf FIXEdge.tgz
```

Upload licenses (engine.license and fixaj2-license.bin) to /home/user/FIXEdge folder

Start fixicc agent

```
$ cd /home/user/FIXEdge/fixicc-agent/bin
$ ./installDaemon.sh
$ ./startDaemon.sh
$ sudo iptables -I INPUT -p tcp -m state --state NEW -m tcp --dport 8005 -j ACCEPT
$ sudo iptables -I INPUT -p tcp -m state --state NEW -m tcp --dport 8901 -j ACCEPT
$ sudo service iptables save
```

Edit replication.client.host to #Virtual IP in /home/user/FIXEdge/FixEdge1/conf/replication.client.properties :

replication.client.properties

```
# Client configuration
# server ip to connect to
replication.client.host=10.11.132.199
```

Pacemaker related

Pacemaker configuration

```
# install pacemaker
$ sudo yum install -y corosync pcs pacemaker

# configure firewall
$ sudo iptables -I INPUT -m state --state NEW -p udp -m multiport --dports 5404,5405 -j ACCEPT
$ sudo iptables -I INPUT -p tcp -m state --state NEW -m tcp --dport 2224 -j ACCEPT
$ sudo iptables -I INPUT -p igmp -j ACCEPT
$ sudo iptables -I INPUT -m addrtype --dst-type MULTICAST -j ACCEPT
$ sudo service iptables save

# configure password for cluster login (epm-bfix is used)
$ sudo passwd hacluster

# enable and start service
$ sudo systemctl enable pcsd.service
$ sudo systemctl start pcsd.service

#make cluster autospawning with node spawn
sudo pcs cluster enable
```

Additional configuration

Attention: it's mandatory step.

Modified anything, should be installed (it's missed in current base and epel repositories anyway).

Anything installation

```
$ wget <anything url>
$ sudo cp anything /usr/lib/ocf/resource.d/heartbeat/
$ sudo chmod a+rwx /usr/lib/ocf/resource.d/heartbeat/anything
```

Cluster setting up

Steps from this part are related to cluster setting up, so should be ran in **any** node, and only **once** (except #Health check, which can be run at any moment after initial cluster setup)

Nodes registration

Authorization and configuration

```
# auth both nodes in cluster
$ export nodes="ECSE00100034.epam.com ECSE00100035.epam.com"
$ sudo pcs cluster auth -u hacluster -p epm-bfix $nodes
# create and start cluster on all nodes
$ sudo pcs cluster setup --name fixedge_cluster $nodes && sudo pcs cluster start --all

# we wouldn't need to STONITH nodes so
$ sudo pcs property set stonith-enabled=false

# no need in quorum in two node cluster
$ sudo pcs property set no-quorum-policy=ignore
```

Resources configuration

Resource configuration

```
# virtual_ip resource
# Used for accesing to FIXEdge and replication services behind cluster

$ sudo pcs resource create virtual_ip ocf:heartbeat:IPaddr2 ip=$virtual_ip nic="eth0" cidr_netmask=32 op
monitor interval=30s

# FIXEdge
# FIXEdge instance which will be available to end user
# notmanagepid="true" - extension to of standard anything resource, to make it possible using of FIXEdge's and
other services pid files

$ sudo pcs resource create FIXEdge ocf:heartbeat:anything params binfile="./FixEdge1.run.sh" \
workdir="/home/user/FIXEdge/bin/" \
pidfile="/home/user/FIXEdge/FixEdge1/log/FixEdge.pid" \
user="user" logfile="/home/user/FIXEdge_resource.log" errlogfile="/home/user/FIXEdge_resource_error.log"
notmanagepid="true"

# ReplicationServer
# Serves providing log's replicas to other cluster nodes, should be run on one node with FIXEdge

$ sudo pcs resource create ReplicationServer ocf:heartbeat:anything params binfile="./FixEdge1.replication.
server.run.sh" \
"workdir=/home/user/FIXEdge/bin/" user="user" logfile="/home/user/ReplicationServer.log" \
errlogfile="/home/user/ReplicationServerError.log" pidfile="/home/user/FIXEdge/FixEdge1/log/replication_server.
pid" notmanagepid="true"

# ReplicationClient
# Serves gather replicas to idle node. Should be run in differ of FIXEdge running node.

$ sudo pcs resource create ReplicationClient ocf:heartbeat:anything params binfile="./FixEdge1.replication.
client.run.sh" \
"workdir=/home/user/FIXEdge/bin/" user="user" logfile="/home/user/ReplicationClient.log" \
errlogfile="/home/user/ReplicationClientError.log" pidfile="/home/user/FIXEdge/FixEdge1/log/replication_client.
pid" notmanagepid="true"
```

Resources constraints

To describe resource starting order, and their placement, some constraints should be added:

Constraints

```
# Placement(colocation):
#-----------------------
# FIXEdge should be placed in one node with virtual_ip
$ sudo pcs constraint colocation add FIXEdge virtual_ip INFINITY
# ReplicationServer should be ran with FIXEdge and virtual_ip
$ sudo pcs constraint colocation add ReplicationServer virtual_ip INFINITY
# ReplicationClient should be run in differ node take attention -INFINITY is used
$ sudo pcs constraint colocation add ReplicationClient virtual_ip -INFINITY

# Ordering:
#-----------------------
# First should be started virtual_ip
# then FIXEdge
# then ReplicationServer
$ sudo pcs constraint order virtual_ip then FIXEdge INFINITY
$ sudo pcs constraint order virtual_ip then ReplicationClient INFINITY
$ sudo pcs constraint order FIXEdge then ReplicationServer INFINITY
```

Ending setup

After resources and constraints are configured, cluster should be ready to work.

To avoid any constraint violation reasons which could be produced in configuration process perfrom commands below:

Cluster restarting

```
# Restart cluster on all nodes
$ sudo pcs cluster stop --all && sudo pcs cluster start --all
```

Health check

Comands below with sample correct output will help you to check that everything is corect:

Health check

```
# Cluster status
# Command below shows all resources and nodes which are running them
# You should be able to see that ReplicationServer runned on different of ReplicationClient node
# and all resources are Started
# Starting all resources after cluster restart can take some time (about minute or two)
$ sudo pcs status
Cluster name: fixedge_cluster
Last updated: Fri Apr 17 22:41:22 2015
Last change: Fri Apr 17 20:05:21 2015
Stack: corosync
Current DC: ECSE00100034.epam.com (2) - partition with quorum
Version: 1.1.12-a14efad
2 Nodes configured
4 Resources configured


Online: [ ECSE00100034.epam.com ECSE00100035.epam.com ]

Full list of resources:

 virtual_ip     (ocf::heartbeat:IPaddr2):       Started ECSE00100034.epam.com
 FIXEdge        (ocf::heartbeat:anything):      Started ECSE00100034.epam.com
 ReplicationServer      (ocf::heartbeat:anything):      Started ECSE00100034.epam.com
 ReplicationClient      (ocf::heartbeat:anything):      Started ECSE00100035.epam.com

PCSD Status:
  ECSE00100035.epam.com: Online
  ECSE00100034.epam.com: Online

Daemon Status:
  corosync: active/disabled
  pacemaker: active/disabled
  pcsd: active/enabled
```

Also you can review /var/log/messages.

Notes

What next

This layout can be scaled to more nodes by creating ReplicationClientN resource to each node in cluster in this case one needs NodeCount - 1 resources ReplicationClient.