

INDEX

[VitalPBX High Availability 3](#_Toc48887842)

[1.- Introduction 3](#_Toc48887843)

[1.1.- VitalPBX High Availability 3](#_Toc48887844)

[1.2.- Prerequisites 3](#_Toc48887845)

[2.- Configurations 4](#_Toc48887846)

[2.1- IP Configuration and Hostname. 4](#_Toc48887847)

[2.2.- Installing the necessary software dependencies 5](#_Toc48887848)

[2.3.- Create authorization 6](#_Toc48887849)

[2.4.- Installing from Scripts 6](#_Toc48887850)

[2.5.- Pairing both Servers (Continuing with the step by step) 7](#_Toc48887851)

[2.6.- Firewall 8](#_Toc48887852)

[2.7.- Hostname 9](#_Toc48887853)

[2.8.- Configure MariaDB Master-Master Replication. 9](#_Toc48887854)

[2.9.- Configure lsync 11](#_Toc48887855)

[2.10.- Configure the HA Cluster 15](#_Toc48887856)

[2.11.- Create useful commands for the maintenance of our cluster 16](#_Toc48887857)

[2.11.- Test 19](#_Toc48887858)

[3.- Resources troubleshooting 20](#_Toc48887859)

[3.1.- Very useful commands 20](#_Toc48887860)

[3.2.- Mariadb does not start correctly 20](#_Toc48887861)

[3.3.- I can't get into the VitalPBX interface, it gives me a sql error 21](#_Toc48887862)

[3.4.- The Two server show Master state and the remote server show Offline status 21](#_Toc48887863)

[3.5.- Restore from No High Availability VitalPBX Backup 22](#_Toc48887864)

[3.6.- Cluster Destroy 22](#_Toc48887865)

[3.7.- Create the Cluster again 22](#_Toc48887866)

[3.8.- Destroy the Cluster and Normalize both servers 24](#_Toc48887867)

[3.9.- Recommendations 26](#_Toc48887868)

[4.-0.- Some sources of information 26](#_Toc48887869)

# VitalPBX High Availability

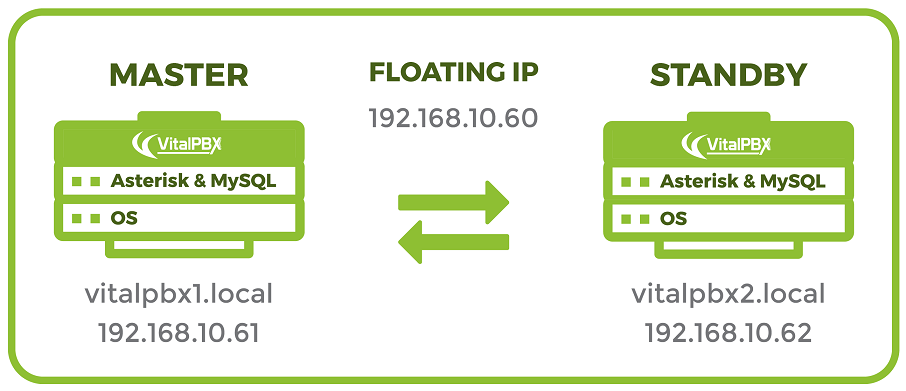
# 1.- Introduction

## 1.1.- VitalPBX High Availability

High availability is a characteristic of a system which aims to ensure an agreed level of operational performance, usually uptime, for a higher than normal period.

Make a high-availability cluster out of any pair of VitalPBX servers. VitalPBX can detect a range of failures on one VitalPBX server and automatically transfer control to the other server, resulting in a telephony environment with minimal down time.

Example



## 1.2.- Prerequisites

In order to install VitalPBX in high availability you need the following:

a.- 3 IP addresses.

b.- Clean installation of VitalPBX Version 3.0 in two servers with similar characteristics.

c.- MariaDB 10 (include in VitalPBX 3)

d.- Corosync, Pacemaker, PCS and lsyncd.

# 2.- Configurations

## 2.1- IP Configuration and Hostname.

We will configure in each server the IP address and the host name.

First, we will go to the web interface under:

**Admin>System Settings>Network Settings**

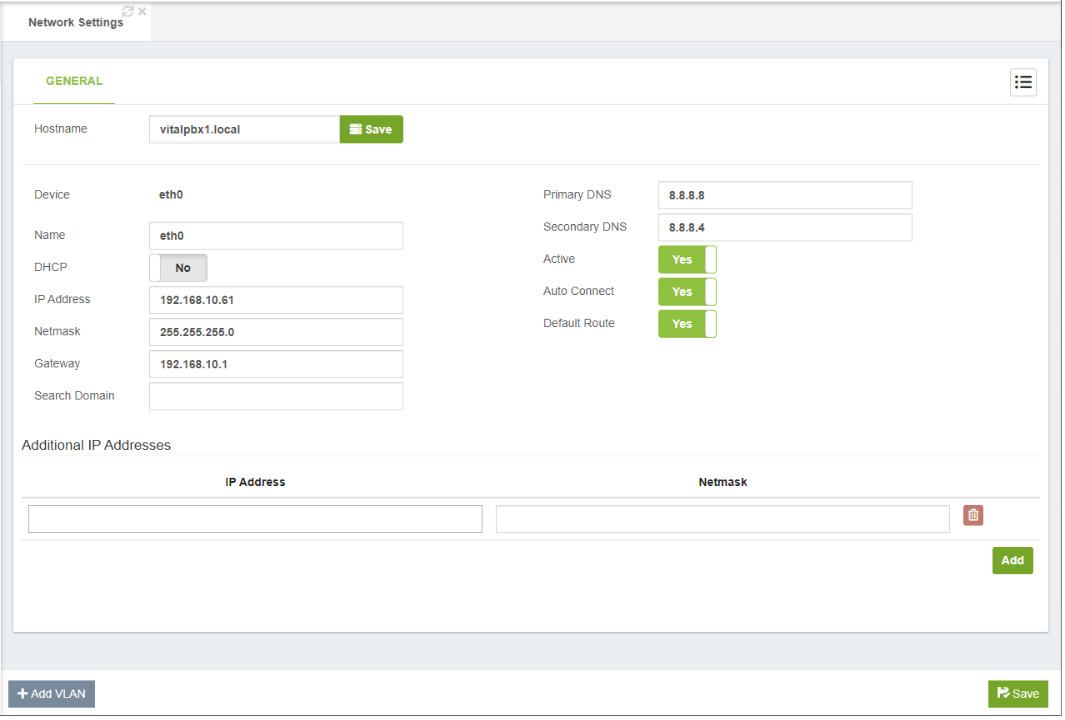
Disable DHCP and configure the selected IP and hostname. In our

example we will use the following values.

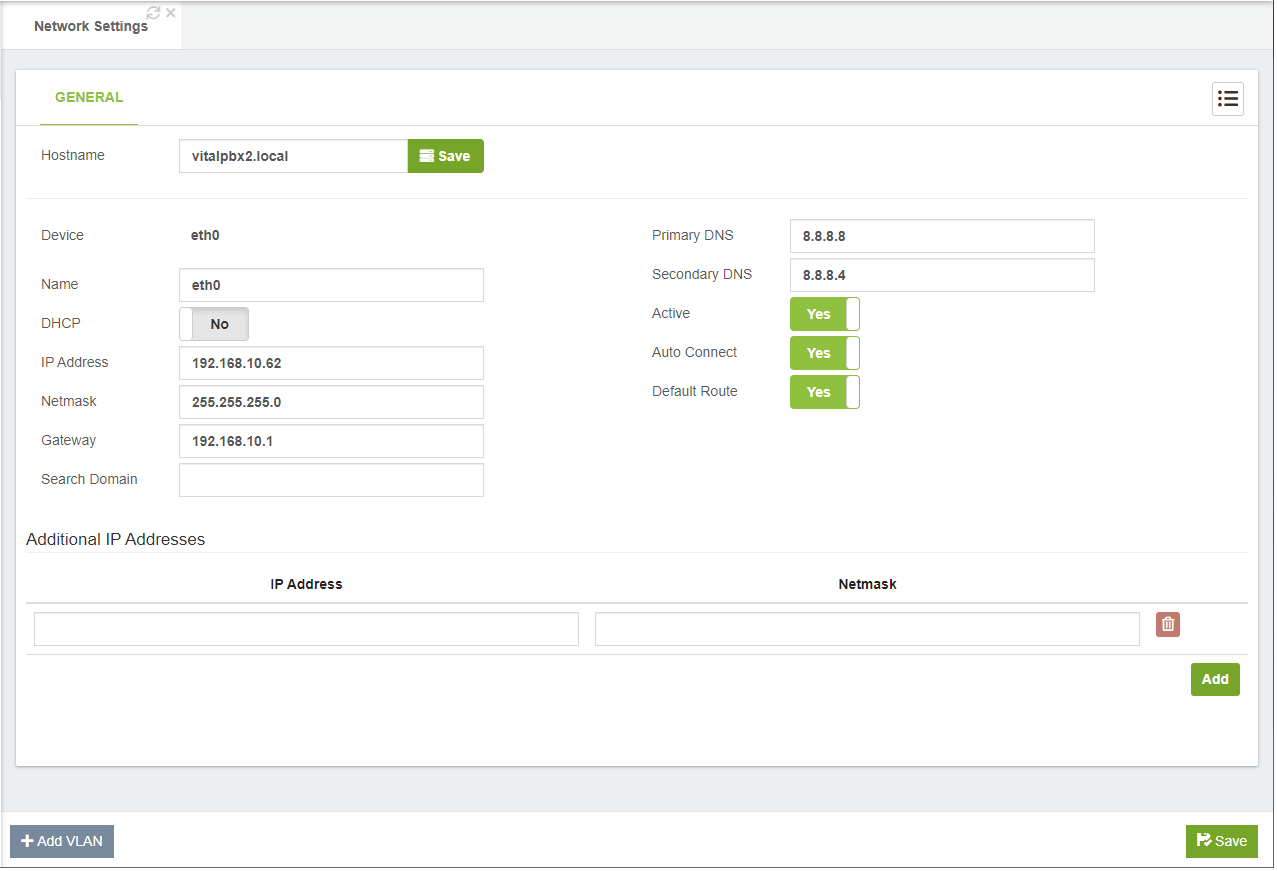
| **Name** | **Master** | **Standby** |
| --- | --- | --- |
| Hostname | vitalpbx1.local | vitalpbx2.local |
| IP Address | 192.168.10.61 | 192.168.10.62 |
| Netmask | 255.255.255.0 | 255.255.255.0 |
| Gateway | 192.168.10.1 | 192.168.10.1 |
| Primary DNS | 8.8.8.8 | 8.8.8.8 |
| Secondary DNS | 8.8.4.4 | 8.8.4.4 |

First change the Hostname, remember press the **Check button ()** next to it to apply the new hostname.

Server 1 (Master)



Server 2 (Standby)



You can also change the hostname from the console using the following command:

Server 1

|  |
| --- |
| [root@ vitalpbx~]# hostnamectl set-hostname vitalpbx**1**.local |

Server 2

|  |
| --- |
| [root@ vitalpbx~]# hostnamectl set-hostname vitalpbx**2**.local |

## 2.2.- Installing the necessary software dependencies

For High Availability services we need to install in both servers Corosync and Pacemaker

|  |
| --- |
| [root@ vitalpbx**1-2** ~]# yum -y install corosync pacemaker pcs |

We are going to synchronize some directories in both servers. For this we need to install lsync in both Server

|  |
| --- |
| [root@ vitalpbx**1-2** ~]# yum install lsyncd -y |

## 2.3.- Create authorization

Create authorization key for the Access between the two servers without credentials

Create key in Server 1

|  |
| --- |
| [root@ vitalpbx**1** ~]# ssh-keygen -f /root/.ssh/id\_rsa -t rsa -N '' >/dev/null  [root@ vitalpbx**1** ~]# ssh-copy-id root@**192.168.10.62**  Are you sure you want to continue connecting (yes/no)? **yes**  root@192.168.10.62's password: **(remote server root’s password)**  Number of key(s) added: 1  Now try logging into the machine, with: "ssh 'root@192.168.10.62'"  and check to make sure that only the key(s) you wanted were added.  [root@vitalpbx**1** ~]# |

Create key in Server 2

|  |
| --- |
| [root@ vitalpbx**2** ~]# ssh-keygen -f /root/.ssh/id\_rsa -t rsa -N '' >/dev/null  [root@ vitalpbx**2** ~]# ssh-copy-id root@**192.168.10.61**  Are you sure you want to continue connecting (yes/no)? **yes**  root@192.168.10.61's password: **(remote server root’s password)**  Number of key(s) added: 1  Now try logging into the machine, with: "ssh 'root@192.168.10.61'"  and check to make sure that only the key(s) you wanted were added.  [root@vitalpbx**2** ~]# |

## 2.4.- Installing from Scripts

If you want to continue step by step go to step 2.5, but if you want to create the configuration automatically, run the following script in Server 1:

|  |
| --- |
| [root@ vitalpbx**1** ~]# mkdir /usr/share/vitalpbx/ha  [root@ vitalpbx**1** ~]# cd /usr/share/vitalpbx/ha  [root@ vitalpbx**1** ~]# wget https://raw.githubusercontent.com/VitalPBX/vitalpbx\_ha/master/vpbxha.sh  [root@ vitalpbx**1** ~]# chmod +x vpbxha.sh  [root@ vitalpbx**1** ~]# ./vpbxha.sh  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Welcome to the VitalPBX high availability installation \*  \* All options are mandatory \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  IP Master................ > **192.168.10.61**  IP Standby............... > **192.168.10.62**  Floating IP.............. > **192.168.10.60**  Floating IP Mask (SIDR).. > **24**  hacluster password....... > **MyPassword (any password)**  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Check Information \*  \* Make sure you have internet on both servers \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Are you sure to continue with this settings? (yes,no) > **yes** |

This process may take a couple of minutes, and once it is done, VitalPBX High Availability will be ready to use.

Always remember to use floating ip to manage your VitalPBX. In this example it is 192.168.10.60.

Some interesting commands that we must keep in mind for any problem we have with our cluster

To destroy the cluster and start again we use the following command

|  |
| --- |
| [root@ vitalpbx**1** ~]# ./vpbxha.sh destroy |

To rebuild the cluster after destroying it

|  |
| --- |
| [root@ vitalpbx**1** ~]# ./vpbxha.sh rebuild |

And lastly to force a refresh of the cluster

|  |
| --- |
| [root@ vitalpbx**1** ~]# pcs resource refresh --full |

Remember that in all cases we must have the script on our server

**Note:**

In some cases, the destroy command does not respond because the corosync service cannot be stopped, if this happens, we recommend stopping the service forcibly with the “kill PID” command (the service PID). Do this on both servers.

|  |
| --- |
| [root@ vitalpbx**1** ~]# systemctl status corosync | grep "Main PID"  Main PID: 5702 (corosync)  [root@vitalpbx1-2 ~]# |

Kill the service in both servers

|  |
| --- |
| [root@ vitalpbx**1** ~]# kill 5702 |

The PID is different on both servers, so we recommend searching for the PID separately.

Important Note:

Please read the section on 3.- Resources Troubleshooting, this will help you solve any problem you have with your cluster.

## 2.5.- Pairing both Servers (Continuing with the step by step)

Now, we will pair the two servers because we need them to have the same main Tenant ID.

Server 1

|  |
| --- |
| [root@ vitalpbx**1**.local~]# mysql -uroot ombutel -e "select path from ombu\_tenants" | awk 'NR==2'  af739029bb237e9e |

Remember this ID.

Server 2

|  |
| --- |
| [root@ vitalpbx**2**.local~]# mysql -uroot ombutel -e "select path from ombu\_tenants" | awk 'NR==2'  633225cc70d86221 |

Next, in Server 2 Update the Tenant ID with the value of Server 1

|  |
| --- |
| [root@ vitalpbx**2**.local~]# mysql -uroot ombutel -e "update ombu\_tenants set path='af739029bb237e9e'" |

And rename the main Tenant path in Server 2

|  |
| --- |
| [root@ vitalpbx**2**.local~]# mv /var/lib/vitalpbx/static/633225cc70d86221 /var/lib/vitalpbx/static/af739029bb237e9e |

## 2.6.- Firewall

In both Servers.

In VitalPBX GUI go to Admin > Firewall > Services and add the following Services.

| **Server Name** | **Port** | **Protocol** |
| --- | --- | --- |
| MariaDB Client | 3306 | TCP |
| MariaDB Galera Traffic | 4567-4568 | TCP |
| MariaDB Galera SST | 4444 | TCP |
| HA2224 | 2224 | TCP |
| HA3121 | 3121 | TCP |
| HA5403 | 5403 | TCP |
| HA5404-5405 | 5404-5405 | UPD |
| HA21064 | 21064 | TCP |
| HA9929 | 9929 | BOTH |

Then, go to Admin > Firewall > Rules and add the Rules for the services created and in the Source for security use the local network (In my case is: 192.168.10.0/24). Then remember to apply changes in both servers. If you want more security you can only add the 3 IPs instead of the entire network (e.g. 192.168.10.61, 192.168.10.62 and 192.168.10.60).

It is also recommended to whitelist these three IPs..

## 2.7.- Hostname

Next, we will connect through ssh to each of the servers and we configure the hostname of each server in the /etc/hots file, so that both servers see each other with the hostname.

|  |
| --- |
| [root@vitalpbx**1-2**.local ~]# echo -e "**192.168.10.61** \tvitalpbx1.local" >> /etc/hosts  [root@vitalpbx**1-2**.local ~]# echo -e "**192.168.10.62** \tvitalpbx2.local" >> /etc/hosts |

Remember to change the IP addresses to your IP addresses

## 2.8.- Configure MariaDB Master-Master Replication.

##### In Server 1 add this 2 line in file /etc/my.cnf.d/vitalpbx.cnf

**server-id=1**

**log-bin=mysql-bin**

**report\_host = master1**

|  |
| --- |
| [root@ vitalpbx**1**.local~]# nano /etc/my.cnf.d/vitalpbx.cnf  [mysqld]  **server-id=1**  **log-bin=mysql-bin**  **report\_host = master1**  innodb\_buffer\_pool\_size = 64M  innodb\_flush\_log\_at\_trx\_commit = 2  innodb\_log\_file\_size = 64M  innodb\_log\_buffer\_size = 64M  bulk\_insert\_buffer\_size = 64M  max\_allowed\_packet = 64M |

##### 2.8.1.- In Server 2

**server-id=2**

**log-bin=mysql-bin**

**report\_host = master2**

|  |
| --- |
| [root@ vitalpbx**2**.local~]# nano /etc/my.cnf.d/vitalpbx.cnf  [mysqld]  **server-id=2**  **log-bin=mysql-bin**  **report\_host = master2**  innodb\_buffer\_pool\_size = 64M  innodb\_flush\_log\_at\_trx\_commit = 2  innodb\_log\_file\_size = 64M  innodb\_log\_buffer\_size = 64M  bulk\_insert\_buffer\_size = 64M  max\_allowed\_packet = 64M |

In Server 1 and 2 restart mariadb service

|  |
| --- |
| [root@ vitalpbx**1-2** ~]# systemctl restart mariadb |

##### 2.8.2.- Create User for Replication in Server 1

Afterwards, in Server 1 create a user for replication slave followed by flush privileges.

|  |
| --- |
| [root@ vitalpbx**1** ~]# mysql -uroot -e "grant replication slave on \*.\* to **vitalpbx\_replica**@'%' identified by '**vitalpbx\_replica**';"  [root@ vitalpbx**1** ~]# mysql -uroot -e "flush privileges;"  [root@ vitalpbx**1** ~]# mysql -uroot -e "flush tables with read lock;" |

##### You can change the username and password for your own

##### 2.8.3.- Get the File Info In Server 1

At this point, if you do a show master status, you need to have a file with the mysql-bin incrementing number as well a position number. **Please write these two values**.

|  |
| --- |
| [root@ vitalpbx**1** ~]# mysql -uroot -e "show master status" | awk 'NR==2 {print $1}'  [root@ vitalpbx**1** ~]# mysql -uroot -e "show master status" | awk 'NR==2 {print $2}' |

##### 2.8.4.- Make all database copy In Server 1 and restore In Server 2

|  |
| --- |
| [root@ vitalpbx**1** ~]# mysqldump -u root --all-databases > all\_databases.sql  [root@ vitalpbx**1** ~]# scp all\_databases.sql root@192.168.10.62:/tmp/all\_databases.sql  [root@ vitalpbx**2** ~]# cd /tmp  [root@ vitalpbx**2** ~]# mysql mysql -u root < /tmp/all\_databases.sql |

##### 2.8.5.- Create User for Replication in Server 2

Afterwards, in Server 2 create a user for replication between two servers

|  |
| --- |
| [root@ vitalpbx**2** ~]# mysql -uroot -e "grant replication slave on \*.\* to **vitalpbx\_replica**@'%' identified by '**vitalpbx\_replica**';"  [root@ vitalpbx**2** ~]# mysql -uroot -e "flush privileges;"  [root@ vitalpbx**2** ~]# mysql -uroot -e "flush tables with read lock;" |

##### You can change the username and password for your own

##### 2.8.6.- Create the Replication in Server 2

Now, we will temporarily run Server 2 as the slave.

|  |
| --- |
| [root@ vitalpbx**2** ~]# mysql -uroot -e "change master to master\_host='192.168.10.61', master\_user='vitalpbx\_replica', master\_password='vitalpbx\_replica', master\_log\_file='**file\_server\_1**', master\_log\_pos=**position\_server\_1**;"  [root@ vitalpbx**2** ~]# mysql -uroot -e "start slave;" |

You need to change the value of **file\_server\_1** and **position\_server\_1** for the value that you wrote in the step 2.8.3.

##### 2.8.7.- Get the File Info In Server 2

Now In Server 2, if you do a show master status, you need to have a file with the mysql-bin incrementing number as well a position number. **Please write these two values**.

|  |
| --- |
| [root@ vitalpbx**2** ~]# mysql -uroot -e "show master status" | awk 'NR==2 {print $1}'  [root@ vitalpbx**2** ~]# mysql -uroot -e "show master status" | awk 'NR==2 {print $2}' |

##### 2.8.8.- Create the Replication in Server 1

Now make the las change in Server 1

|  |
| --- |
| [root@ vitalpbx**1** ~]# mysql -uroot -e "unlock table;"  [root@ vitalpbx**1** ~]# mysql -uroot -e "change master to master\_host='192.168.10.62', master\_user='vitalpbx\_replica', master\_password='vitalpbx\_replica', master\_log\_file='**file\_server\_2**', master\_log\_pos=**position\_server\_2**;"  [root@ vitalpbx**1** ~]# mysql -uroot -e "start slave;" |

You need to change the value of **file\_server\_2** and **position\_server\_2** for the value that you wrote in the step 2.8.6.

## 2.9.- Configure lsync

##### 2.9.1.- In Server 1

Remember to change the IP addresses to your IP addresses

|  |
| --- |
| [root@ vitalpbx**1** ~]# nano /etc/lsyncd.conf  ----  -- User configuration file for lsyncd.  --  -- Simple example for default rsync.  --  settings {  logfile = "/var/log/lsyncd/lsyncd.log",  statusFile = "/var/log/lsyncd/lsyncd-status.log",  statusInterval = 20,  nodaemon = true,  insist = true,  }  sync {  default.rsync,  source="/var/spool/asterisk/monitor",  target="**192.168.10.62**:/var/spool/asterisk/monitor",  rsync={  owner = true,  group = true  }  }  sync {  default.rsync,  source="/var/lib/asterisk/",  target="**192.168.10.62**:/var/lib/asterisk/",  rsync = {  binary = "/usr/bin/rsync",  owner = true,  group = true,  archive = "true",  \_extra = {  "--include=astdb.sqlite3",  "--exclude=\*"  }  }  }  sync {  default.rsync,  source="/var/lib/asterisk/agi-bin/",  target="**192.168.10.62**:/var/lib/asterisk/agi-bin/",  rsync={  owner = true,  group = true  }  }  sync {  default.rsync,  source="/var/lib/asterisk/priv-callerintros/",  target="**192.168.10.62**:/var/lib/asterisk/priv-callerintros",  rsync={  owner = true,  group = true  }  }  sync {  default.rsync,  source="/var/lib/asterisk/sounds/",  target="**192.168.10.62**:/var/lib/asterisk/sounds/",  rsync={  owner = true,  group = true  }  }  sync {  default.rsync,  source="/var/lib/vitalpbx",  target="**192.168.10.62**:/var/lib/vitalpbx",  rsync = {  binary = "/usr/bin/rsync",  owner = true,  group = true,  archive = "true",  \_extra = {  "--exclude=\*.lic",  "--exclude=\*.dat",  "--exclude=dbsetup-done",  "--exclude=cache"  }  }  }  sync {  default.rsync,  source="/etc/asterisk",  target="**192.168.10.62**:/etc/asterisk",  rsync={  owner = true,  group = true  }  } |

##### 2.9.2.- In Server 2

Remember to change the IP addresses to your IP addresses

|  |
| --- |
| [root@ vitalpbx**2** ~]# nano /etc/lsyncd.conf  ----  -- User configuration file for lsyncd.  --  -- Simple example for default rsync.  --  settings {  logfile = "/var/log/lsyncd/lsyncd.log",  statusFile = "/var/log/lsyncd/lsyncd-status.log",  statusInterval = 20,  nodaemon = true,  insist = true,  }  sync {  default.rsync,  source="/var/spool/asterisk/monitor",  target="**192.168.10.61**:/var/spool/asterisk/monitor",  rsync={  owner = true,  group = true  }  }  sync {  default.rsync,  source="/var/lib/asterisk/",  target="**192.168.10.61**:/var/lib/asterisk/",  rsync = {  binary = "/usr/bin/rsync",  owner = true,  group = true,  archive = "true",  \_extra = {  "--include=astdb.sqlite3",  "--exclude=\*"  }  }  }  sync {  default.rsync,  source="/var/lib/asterisk/agi-bin/",  target="**192.168.10.61**:/var/lib/asterisk/agi-bin/",  rsync={  owner = true,  group = true  }  }  sync {  default.rsync,  source="/var/lib/asterisk/priv-callerintros/",  target="**192.168.10.61**:/var/lib/asterisk/priv-callerintros",  rsync={  owner = true,  group = true  }  }  sync {  default.rsync,  source="/var/lib/asterisk/sounds/",  target="**192.168.10.61**:/var/lib/asterisk/sounds/",  rsync={  owner = true,  group = true  }  }  sync {  default.rsync,  source="/var/lib/vitalpbx",  target="**192.168.10.61**:/var/lib/vitalpbx",  rsync = {  binary = "/usr/bin/rsync",  owner = true,  group = true,  archive = "true",  \_extra = {  "--exclude=\*.lic",  "--exclude=\*.dat",  "--exclude=dbsetup-done",  "--exclude=cache"  }  }  }  sync {  default.rsync,  source="/etc/asterisk",  target="**192.168.10.61**:/etc/asterisk",  rsync={  owner = true,  group = true  }  } |

## 2.10.- Configure the HA Cluster

##### 2.10.1.- Configure the start of services on both servers

|  |
| --- |
| [root@ vitalpbx**1-2** ~]# systemctl start pcsd  [root@ vitalpbx**1-2** ~]# systemctl enable pcsd.service  [root@ vitalpbx**1-2** ~]# systemctl enable corosync.service  [root@ vitalpbx**1-2** ~]# systemctl enable pacemaker.service |

##### 2.10.2.- Create the password of the hacluster user on both nodes

|  |
| --- |
| [root@ vitalpbx**1-2** ~]# echo **MyPassword** | passwd --stdin hacluster |

##### 2.10.3.- Server Authenticate in Server 1

|  |
| --- |
| [root@ vitalpbx**1** ~]# pcs cluster auth vitalpbx1.local vitalpbx2.local -u hacluster -p **MyPassword**  vitalpbx1.local: Authorized  vitalpbx2.local: Authorized |

##### 2.10.4.- Create the cluster and configure parameters, perform only on the Server 1

|  |
| --- |
| [root@ vitalpbx**1** ~]# pcs cluster setup --name cluster\_vitalpbx vitalpbx1.local vitalpbx2.local |

##### 2.10.5.- Starting Cluster in Server 1

|  |
| --- |
| [root@ vitalpbx**1** ~]# pcs cluster start --all  [root@ vitalpbx**1** ~]# pcs cluster enable --all  [root@ vitalpbx**1** ~]# pcs property set stonith-enabled=false  [root@ vitalpbx**1** ~]# pcs property set no-quorum-policy=ignore |

##### 2.10.6.- Stop services and disable in both servers

|  |
| --- |
| [root@ vitalpbx**1-2** ~]# systemctl stop asterisk  [root@ vitalpbx**1-2** ~]# systemctl disable asterisk |

##### 2.10.7.- Create resource for the use of the Floating IP

|  |
| --- |
| [root@ vitalpbx**1** ~]# pcs resource create virtual\_ip ocf:heartbeat:IPaddr2 ip=192.168.10.60 cidr\_netmask=24 op monitor interval=30s on-fail=restart  [root@ vitalpbx**1** ~]# pcs cluster cib drbd\_cfg  [root@ vitalpbx**1** ~]# pcs cluster cib-push drbd\_cfg |

##### 2.10.8.- Create asterisk Service in Server 1

|  |
| --- |
| [root@ vitalpbx**1** ~]# pcs resource create asterisk service:asterisk op monitor interval=30s  [root@ vitalpbx**1** ~]# pcs cluster cib fs\_cfg  [root@ vitalpbx**1** ~]# pcs cluster cib-push fs\_cfg --config  [root@ vitalpbx**1** ~]# pcs -f fs\_cfg constraint colocation add asterisk with virtual\_ip INFINITY  [root@ vitalpbx**1** ~]# pcs -f fs\_cfg constraint order virtual\_ip then asterisk  [root@ vitalpbx**1** ~]# pcs cluster cib-push fs\_cfg –config  [root@ vitalpbx**1** ~]# pcs resource update asterisk op stop timeout=120s  [root@ vitalpbx**1** ~]# pcs resource update asterisk op start timeout=120s  [root@ vitalpbx**1** ~]# pcs resource update asterisk op restart timeout=120s |

Note:

Changing these values from 15s (default) to 120s is very important since depending on the server and the number of extensions the Asterisk can take more than 15s to start

##### 2.10.9.- Create lsyncd Service in Server 1

|  |
| --- |
| [root@ vitalpbx**1** ~]# pcs resource create lsyncd service:lsyncd.service op monitor interval=30s  [root@ vitalpbx**1** ~]# pcs cluster cib fs\_cfg  [root@ vitalpbx**1** ~]# pcs cluster cib-push fs\_cfg --config  [root@ vitalpbx**1** ~]# pcs -f fs\_cfg constraint colocation add lsyncd with virtual\_ip INFINITY  [root@ vitalpbx**1** ~]# pcs -f fs\_cfg constraint order asterisk then lsyncd  [root@ vitalpbx**1** ~]# pcs cluster cib-push fs\_cfg --config |

## 2.11.- Create useful commands for the maintenance of our cluster

##### 2.11.1.- Create “bascul” command in both servers

|  |
| --- |
| [root@ vitalpbx**1-2** ~]# nano /usr/local/bin/bascul  #!/bin/bash  set -e  # Authors: Rodrigo Cuadra  # with Collaboration of Jose Miguel Rivera  # 4-Jul-2020  # Support: rcuadra@aplitel.com  #  #funtion for draw a progress bar  #You must pass as argument the amount of secconds that the progress bar will run  #progress-bar 10 --> it will generate a progress bar that will run per 10 seconds  progress-bar() {  local duration=${1}  already\_done() { for ((done=0; done<$elapsed; done++)); do printf ">"; done }  remaining() { for ((remain=$elapsed; remain<$duration; remain++)); do printf " "; done }  percentage() { printf "| %s%%" $(( (($elapsed)\*100)/($duration)\*100/100 )); }  clean\_line() { printf "\r"; }  for (( elapsed=1; elapsed<=$duration; elapsed++ )); do  already\_done; remaining; percentage  sleep 1  clean\_line  done  clean\_line  }  server\_a=`pcs status | awk 'NR==10 {print $3}'`  server\_b=`pcs status | awk 'NR==10 {print $4}'`  server\_master=`pcs status resources | awk 'NR==1 {print $4}'`  #Perform some validations  if [ "${server\_a}" = "" ] || [ "${server\_b}" = "" ]  then  echo -e "\e[41m There are problems with high availability, please check with the command \*pcs status\* (we recommend applying the command \*pcs cluster unstandby\* in both servers) \e[0m"  exit;  fi  if [[ "${server\_master}" = "${server\_a}" ]]; then  host\_master=$server\_a  host\_standby=$server\_b  else  host\_master=$server\_b  host\_standby=$server\_a  fi  # Print a warning message and ask to the user if he wants to continue  echo -e "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"  echo -e "\* Change the roles of servers in high availability \*"  echo -e "\*\e[41m WARNING-WARNING-WARNING-WARNING-WARNING-WARNING-WARNING \e[0m\*"  echo -e "\*All calls in progress will be lost and the system will be \*"  echo -e "\* be in an unavailable state for a few seconds. \*"  echo -e "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"  #Perform a loop until the users confirm if wants to proceed or not  while [[ $perform\_bascul != yes && $perform\_bascul != no ]]; do  read -p "Are you sure to switch from $host\_master to $host\_standby? (yes,no) > " perform\_bascul  done  if [[ "${perform\_bascul}" = "yes" ]]; then  #Unstandby both nodes  pcs cluster unstandby $host\_master  pcs cluster unstandby $host\_standby  #Do a loop per resource  pcs status resources | grep "^s.\*s(.\*):s.\*" | awk '{print $1}' | while read -r resource ; do  #Skip moving the virutal\_ip resource, it will be moved at the end  if [[ "${resource}" != "virtual\_ip" ]]; then  echo "Moving ${resource} from ${host\_master} to ${host\_standby}"  pcs resource move ${host\_standby}  fi  done  sleep 5 && pcs cluster standby $host\_master & #Standby current Master node after five seconds  sleep 20 && pcs cluster unstandby $host\_master & #Automatically Unstandby current Master node after$  #Move the Virtual IP resource to standby node  echo "Moving virutal\_ip from ${host\_master} to ${host\_standby}"  pcs resource move virtual\_ip ${host\_standby}  #End the script  echo "Becoming ${host\_standby} to Master"  progress-bar 10  echo "Done"  else  echo "Nothing to do, bye, bye"  fi  sleep 5  role |

Add permissions

|  |
| --- |
| [root@ vitalpbx**1-2** ~]# chmod +x /usr/local/bin/bascul |

##### 2.11.2.- Create “role” command in both servers

|  |
| --- |
| [root@ vitalpbx**1-2** ~]# nano /usr/local/bin/role  #Bash Colour Codes  green="\033[00;32m"  txtrst="\033[00;0m"  if [ -f /etc/redhat-release ]; then  linux\_ver=`cat /etc/redhat-release`  vitalpbx\_ver=`rpm -qi vitalpbx |awk -F: '/^Version/ {print $2}'`  vitalpbx\_release=`rpm -qi vitalpbx |awk -F: '/^Release/ {print $2}'`  elif [ -f /etc/debian\_version ]; then  linux\_ver="Debian "`cat /etc/debian\_version`  vitalpbx\_ver=`dpkg -l vitalpbx |awk '/ombutel/ {print $3}'`  else  linux\_ver=""  vitalpbx\_ver=""  vitalpbx\_release=""  fi  vpbx\_version="${vitalpbx\_ver}-${vitalpbx\_release}"  asterisk\_version=`rpm -q --qf "%{VERSION}" asterisk`  server\_master=`pcs status resources | awk 'NR==1 {print $4}'`  host=`hostname`  if [[ "${server\_master}" = "${host}" ]]; then  server\_mode="Master"  else  server\_mode="Standby"  fi  logo='  \_ \_ \_ \_ \_\_\_\_\_\_ \_\_\_\_\_\_ \_ \_  | | | (\_)\_ | (\_\_\_\_\_ (\_\_\_\_ \ \ / /  | | | |\_| |\_ \_\_\_\_| |\_\_\_\_\_) )\_\_\_) ) \/ /  \ \/ /| | \_)/ \_ | | \_\_\_\_/ \_\_ ( ) (  \ / | | |\_( ( | | | | | |\_\_) ) /\ \  \/ |\_|\\_\_\_)\_||\_|\_|\_| |\_\_\_\_\_\_/\_/ \\_\  '  echo -e "  ${green}  ${logo}  ${txtrst}  Role : $server\_mode  Version : ${vpbx\_version//[[:space:]]}  Asterisk : ${asterisk\_version}  Linux Version : ${linux\_ver}  Welcome to : `hostname`  Uptime : `uptime | grep -ohe 'up .\*' | sed 's/up //g' | awk -F "," '{print $1}'`  Load : `uptime | grep -ohe 'load average[s:][: ].\*' | awk '{ print "Last Minute: " $3" Last 5 Minutes: "$4" Last 15 Minutes: "$5 }'`  Users : `uptime | grep -ohe '[0-9.\*] user[s,]'`  IP Address : ${green}`ip addr | sed -En 's/127.0.0.1//;s/.\*inet (addr:)?(([0-9]\*\.){3}[0-9]\*).\*/\2/p' | xargs`${txtrst}  Clock :`timedatectl | sed -n '/Local time/ s/^[ \t]\*Local time:\(.\*$\)/\1/p'`  NTP Sync. :`timedatectl |awk -F: '/NTP sync/ {print $2}'`  "  echo -e ""  echo -e "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"  echo -e "\* Servers Status \*"  echo -e "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"  echo -e "Master"  pcs status resources  echo -e ""  echo -e "Servers Status"  pcs cluster pcsd-status |

Add permissions and make a copy to /etc/profile.d/vitalwelcome.sh

|  |
| --- |
| [root@ vitalpbx**1-2** ~]# chmod +x /usr/local/bin/role  [root@ vitalpbx**1-2** ~]# cp -rf /usr/local/bin/role /etc/profile.d/vitalwelcome.sh |

## 2.11.- Test

##### 2.11.1.- Test bascul

|  |
| --- |
| [root@vitalpbx1 ~]# bascul  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Change the roles of servers in high availability \*  \* WARNING-WARNING-WARNING-WARNING-WARNING-WARNING-WARNING \*  \*All calls in progress will be lost and the system will be \*  \* be in an unavailable state for a few seconds. \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Are you sure to switch from vitalpbx1.local to vitalpbx2.local? (yes,no) > yes  Moving virutal\_ip from vitalpbx1.local to vitalpbx2.local  Becoming vitalpbx2.local to Master  Done>>>>>>| 100%  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Servers Status \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Master  virtual\_ip (ocf::heartbeat:IPaddr2): Started vitalpbx2.local  asterisk (service:asterisk): Started vitalpbx2.local  lsyncd (service:lsyncd.service): Started vitalpbx2.local  Servers Status  vitalpbx1.local: Online  vitalpbx2.local: Online  [root@vitalpbx1 ~]# |

##### 2.11.2.- Show the Cluster status

|  |
| --- |
| [root@vitalpbx1 ~]# role  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Servers Status \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Master  virtual\_ip (ocf::heartbeat:IPaddr2): Started vitalpbx2.local  asterisk (service:asterisk): Started vitalpbx2.local  lsyncd (service:lsyncd.service): Started vitalpbx2.local  Servers Status  vitalpbx1.local: Online  vitalpbx2.local: Online  [root@vitalpbx1 ~]# |

# 3.- Resources troubleshooting

## 3.1.- Very useful commands

If a resource has failed, a failure message appears when you display the cluster status. If you resolve that resource, you can clear that failure status with the pcs resource cleanup command. This command resets the resource status and fail count, telling the cluster to forget the operation history of a resource and re-detect its current state.

The following command cleans up the resource specified by resource\_id.

|  |
| --- |
| [root@vitalpbx**1-2** ~]# pcs resource cleanup resource\_id |

If you do not specify a resource\_id, this command resets the resource status and fail count for all resources.

As of Red Hat Enterprise Linux 7.5, the pcs resource cleanup command probes only the resources that display as a failed action. To probe all resources on all nodes you can enter the following command:

|  |
| --- |
| [root@vitalpbx**1-2** ~]# pcs resource refresh |

By default, the pcs resource refresh command probes only the nodes where a resource's state is known. To probe all resources even if the state is not known, enter the following command:

|  |
| --- |
| [root@vitalpbx**1-2** ~]# pcs resource refresh --full |

Show all the parameters associate with resources

|  |
| --- |
| [root@vitalpbx**1-2** ~]# pcs resource show (asterisk, lsyncd or virtual\_ip) |

Is possible update a parameter after finish to create the cluster with the fallowing command. For change the timeout in start for the resource asterisk do this:

|  |
| --- |
| [root@vitalpbx**1-2** ~]# pcs resource update asterisk op start timeout=120s |

## 3.2.- Mariadb does not start correctly

If you shut down all nodes at the same time, then you have effectively terminated the cluster. Of course, the cluster's data still exists, but the running cluster no longer exists. When this happens, you'll need to bootstrap the cluster again.

If the cluster is not bootstrapped and mysqld on the first node is just started normally, then the node willl try to connect to at least one of the nodes listed in the wsrep\_cluster\_address option. If no nodes are currently running, then this will fail. Bootstrapping the first node solves this problem.

In some cases Galera will refuse to bootstrap a node if it detects that it might not be the most advanced node in the cluster. Galera makes this determination if the node was not the last one in the cluster to be shut down or if the node crashed. In those cases, manual intervention is needed.

If you experience this issue the **recovery\_galera** command solves it

|  |
| --- |
| [root@vitalpbx**1** ~]# recovery\_galera |

If we cannot recover with the recovery\_galera command it means that we will have to do it manually, for which on the server we will edit the file **/var/lib/mysql/grastate.dat** and change the value of **safe\_to\_bootstrap: 0** to **safe\_to\_bootstrap: 1** on the server that we believe has the most up-to-date data from the databases.

|  |
| --- |
| [root@vitalpbx**#** ~]# nano /var/lib/mysql/grastate.dat |

Then on the same server we execute the following command:

|  |
| --- |
| [root@vitalpbx**#** ~]# galera\_new\_cluster |

And on the other server we start mariadb normally

|  |
| --- |
| [root@vitalpbx**$** ~]# systemctl restart mariadb |

With this, our mariadb cluster should be normalized.

## 3.3.- I can't get into the VitalPBX interface, it gives me a sql error

A command may fail with ER\_UNKNOWN\_COM\_ERROR producing 'WSREP has not yet prepared node for application use' (or 'Unknown command' in older versions) error message. It happens when a cluster is suspected to be split and the node is in a smaller part — for example, during a network glitch, when nodes temporarily lose each other. It can also occur during state transfer. The node takes this measure to prevent data inconsistency. Its usually a temporary state which can be detected by checking wsrep\_ready value. The node, however, allows SHOW and SET command during this period.

In the Server that have the Issue

|  |
| --- |
| [root@vitalpbx**#** ~]# mysql -uroot  [root@vitalpbx**#** ~]# SET GLOBAL wsrep\_provider\_options="pc.bootstrap=yes";  [root@vitalpbx**#** ~]# exit; |

Now on the other server

|  |
| --- |
| [root@vitalpbx**#** ~]# systemctl restart mariadb |

## 3.4.- The Two server show Master state and the remote server show Offline status

##### Some time the server lost the credential of the cluster or the firewall rules Is removed. First check the firewall rules, then re-auth you cluster nodes.

|  |
| --- |
| [root@ vitalpbx**1** ~]# pcs cluster auth vitalpbx1.local vitalpbx2.local -u hacluster -p **MyPassword**  vitalpbx1.local: Authorized  vitalpbx2.local: Authorized |

## 3.5.- Restore from No High Availability VitalPBX Backup

It is not recommended to restore from a VitalPBX that is not in High Availability. However, if you want to do so, you must follow the following steps.

1.- We recommend doing the restore before doing high availability.

2.- If you have already done high availability and you want to do it, you have to take into account that the restore will eliminate the firewall rules for high availability, so it is recommended to execute the following script to correct this problem.

|  |
| --- |
| [root@ vitalpbx**1** ~]# cd /usr/share/vitalpbx/ha  [root@ vitalpbx**1** ~]# wget https://raw.githubusercontent.com/VitalPBX/vitalpbx\_ha/master/vpbxhafirewall.sh  [root@ vitalpbx**1** ~]# chmod +x vpbxhafirewall.sh  [root@ vitalpbx**1** ~]# ./ vpbxhafirewall.sh  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Welcome to the VitalPBX high availability installation \*  \* You need three server for this implementation \*  \* All options are mandatory \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  IP Master................ > **192.168.10.61**  IP Standby............... > **192.168.10.62**  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  \* Check Information \*  \* Make sure you have internet on both servers \*  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  Are you sure to continue with this settings? (yes,no) > **yes** |

## 3.6.- Cluster Destroy

In some cases, the cluster is not working as we expect, sometimes it is necessary to make the cluster again, so the steps to follow are as follows.

To destroy the cluster on both servers run the following command

|  |
| --- |
| [root@vitalpbx**1-2** ~]# pcs cluster destroy |

If you are configuring step by step you must follow the following steps

## 3.7.- Create the Cluster again

##### 3.7.1.- Create the cluster and configure parameters, perform only on the Server 1

|  |
| --- |
| [root@ vitalpbx**1** ~]# pcs cluster setup --name cluster\_vitalpbx vitalpbx1.local vitalpbx2.local |

##### 3.7.2.- Starting Cluster in Server 1

|  |
| --- |
| [root@ vitalpbx**1** ~]# pcs cluster start --all  [root@ vitalpbx**1** ~]# pcs cluster enable --all  [root@ vitalpbx**1** ~]# pcs property set stonith-enabled=false  [root@ vitalpbx**1** ~]# pcs property set no-quorum-policy=ignore |

##### 3.7.3.- Stop services and disable in both servers

|  |
| --- |
| [root@ vitalpbx**1-2** ~]# systemctl stop asterisk  [root@ vitalpbx**1-2** ~]# systemctl disable asterisk |

##### 3.7.4.- Create resource for the use of the Floating IP

|  |
| --- |
| [root@ vitalpbx**1** ~]# pcs resource create virtual\_ip ocf:heartbeat:IPaddr2 ip=192.168.10.60 cidr\_netmask=24 op monitor interval=30s on-fail=restart  [root@ vitalpbx**1** ~]# pcs cluster cib drbd\_cfg  [root@ vitalpbx**1** ~]# pcs cluster cib-push drbd\_cfg |

##### 3.7.5.- Create asterisk Service in Server 1

|  |
| --- |
| [root@ vitalpbx**1** ~]# pcs resource create asterisk service:asterisk op monitor interval=30s  [root@ vitalpbx**1** ~]# pcs cluster cib fs\_cfg  [root@ vitalpbx**1** ~]# pcs cluster cib-push fs\_cfg --config  [root@ vitalpbx**1** ~]# pcs -f fs\_cfg constraint colocation add asterisk with virtual\_ip INFINITY  [root@ vitalpbx**1** ~]# pcs -f fs\_cfg constraint order virtual\_ip then asterisk  [root@ vitalpbx**1** ~]# pcs cluster cib-push fs\_cfg –config  [root@ vitalpbx**1** ~]# pcs resource update asterisk op stop timeout=120s  [root@ vitalpbx**1** ~]# pcs resource update asterisk op start timeout=120s  [root@ vitalpbx**1** ~]# pcs resource update asterisk op restart timeout=120s |

##### 3.7.6.- Create lsyncd Service in Server 1

|  |
| --- |
| [root@ vitalpbx**1** ~]# pcs resource create lsyncd service:lsyncd.service op monitor interval=30s  [root@ vitalpbx**1** ~]# pcs cluster cib fs\_cfg  [root@ vitalpbx**1** ~]# pcs cluster cib-push fs\_cfg --config  [root@ vitalpbx**1** ~]# pcs -f fs\_cfg constraint colocation add lsyncd with virtual\_ip INFINITY  [root@ vitalpbx**1** ~]# pcs -f fs\_cfg constraint order asterisk then lsyncd  [root@ vitalpbx**1** ~]# pcs cluster cib-push fs\_cfg --config |

##### 3.7.7.- Server 1 and 2

|  |
| --- |
| [root@vitalpbx**1-2** ~]# pcs resource refresh --full |

Wait 30 seconds and run the role command to ensure everything is fine.

|  |
| --- |
| [root@vitalpbx**1** ~]# role |

## 3.8.- Destroy the Cluster and Normalize both servers

If for any reason you want to become independent again the two servers must follow the following procedure.

Destroy the cluster on both servers run the following command

|  |
| --- |
| [root@vitalpbx**1-2** ~]# pcs cluster destroy |

Note:

In some cases, the destroy command does not respond because the corosync service cannot be stopped, if this happens, we recommend stopping the service forcibly with the “kill PID” command (the service PID). Do this on both servers.

|  |
| --- |
| [root@ vitalpbx**1-2** ~]# systemctl status corosync | grep "Main PID"  Main PID: 5702 (corosync)  [root@vitalpbx**1-2** ~]# |

Kill the service in both servers

|  |
| --- |
| [root@ vitalpbx**1** ~]# kill 5702 |

The PID is different on both servers, so we recommend searching for the PID separately.

Now that we have destroyed the cluster we are going to remove the Galera cluster.

It's easy. You should gracefully stop all nodes. After this remove all Galera configuration from /etc/my.cnf.d/server.cnf and start MySQL. Better to do it on latest stopped node, to prevent data loss.

|  |
| --- |
| [root@ vitalpbx**1-2** ~]# systemctl stop mariadb |

In Server 1 remove all the yellow data, do the same in server 2.

|  |
| --- |
| [root@ vitalpbx**1**.local~]# nano /etc/my.cnf.d/server.cnf  [galera]  binlog\_format=ROW  default-storage-engine=innodb  innodb\_autoinc\_lock\_mode=2  bind-address=0.0.0.0  # Galera Provider Configuration  wsrep\_on=ON  wsrep\_provider=/usr/lib64/galera-4/libgalera\_smm.so  # Galera Cluster Configuration  wsrep\_cluster\_name="test\_cluster"  wsrep\_cluster\_address="gcomm://**192.168.10.61,192.168.10.62**"  # Galera Synchronization Configuration  wsrep\_sst\_method=rsync  # Galera Node Configuration  wsrep\_node\_address="**192.168.10.61**"  wsrep\_node\_name="Server 1" |

Now start mariadb in both servers

|  |
| --- |
| [root@ vitalpbx**1-2** ~]# systemctl start mariadb |

Now normalize the start of the asterisk service on both servers

|  |
| --- |
| [root@ vitalpbx**1-2** ~]# systemctl enable asterisk  [root@ vitalpbx**1-2** ~]# systemctl start asterisk |

Now go to the /etc/profile.d/vitalwelcome.sh file and delete the lines that are yellow on both servers

|  |
| --- |
| [root@ vitalpbx**1-2** ~]# nano /etc/profile.d/vitalwelcome.sh  #Bash Colour Codes  green="\033[00;32m"  txtrst="\033[00;0m"  if [ -f /etc/redhat-release ]; then  linux\_ver=`cat /etc/redhat-release`  vitalpbx\_ver=`rpm -qi vitalpbx |awk -F: '/^Version/ {print $2}'`  vitalpbx\_release=`rpm -qi vitalpbx |awk -F: '/^Release/ {print $2}'`  elif [ -f /etc/debian\_version ]; then  linux\_ver="Debian "`cat /etc/debian\_version`  vitalpbx\_ver=`dpkg -l vitalpbx |awk '/ombutel/ {print $3}'`  else  linux\_ver=""  vitalpbx\_ver=""  vitalpbx\_release=""  fi  vpbx\_version="${vitalpbx\_ver}-${vitalpbx\_release}"  asterisk\_version=`rpm -q --qf "%{VERSION}" asterisk`  server\_master=`pcs status resources | awk 'NR==1 {print $4}'`  host=`hostname`  if [[ "${server\_master}" = "${host}" ]]; then  server\_mode="Master"  else  server\_mode="Standby"  fi  logo='  \_ \_ \_ \_ \_\_\_\_\_\_ \_\_\_\_\_\_ \_ \_  | | | (\_)\_ | (\_\_\_\_\_ (\_\_\_\_ \ \ / /  | | | |\_| |\_ \_\_\_\_| |\_\_\_\_\_) )\_\_\_) ) \/ /  \ \/ /| | \_)/ \_ | | \_\_\_\_/ \_\_ ( ) (  \ / | | |\_( ( | | | | | |\_\_) ) /\ \  \/ |\_|\\_\_\_)\_||\_|\_|\_| |\_\_\_\_\_\_/\_/ \\_\  '  echo -e "  ${green}  ${logo}  ${txtrst}  Role : $server\_mode  Version : ${vpbx\_version//[[:space:]]}  Asterisk : ${asterisk\_version}  Linux Version : ${linux\_ver}  Welcome to : `hostname`  Uptime : `uptime | grep -ohe 'up .\*' | sed 's/up //g' | awk -F "," '{print $1}'`  Load : `uptime | grep -ohe 'load average[s:][: ].\*' | awk '{ print "Last Minute: " $3" Last 5 Minutes: "$4" Last 15 Minutes: "$5 }'`  Users : `uptime | grep -ohe '[0-9.\*] user[s,]'`  IP Address : ${green}`ip addr | sed -En 's/127.0.0.1//;s/.\*inet (addr:)?(([0-9]\*\.){3}[0-9]\*).\*/\2/p' | xargs`${txtrst}  Clock :`timedatectl | sed -n '/Local time/ s/^[ \t]\*Local time:\(.\*$\)/\1/p'`  NTP Sync. :`timedatectl |awk -F: '/NTP sync/ {print $2}'`  "  echo -e ""  echo -e "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"  echo -e "\* Servers Status \*"  echo -e "\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*"  echo -e "Master"  pcs status resources  echo -e ""  echo -e "Servers Status"  pcs cluster pcsd-status |

Remove all relative file and service in both server

|  |
| --- |
| [root@ vitalpbx**1-2** ~]# systemctl stop recovery\_galera  [root@ vitalpbx**1-2** ~]# systemctl disable recovery\_galera  [root@ vitalpbx**1-2** ~]# rm /usr/local/bin/recovery\_galera  [root@ vitalpbx**1-2** ~]# rm /usr/local/bin/role  [root@ vitalpbx**1-2** ~]# rm /usr/local/bin/bascul  [root@ vitalpbx**1-2** ~]# rm /etc/systemd/system/recovery-galera.service |

## 3.9.- Recommendations

* If you have to turn off both servers at the same time, we recommend that you start by turning off the one in Standby and then the Master.
* If the two servers stopped abruptly, always start first that you think you have the most up-to-date information and a few minutes later the other server.
* If you want to update the version of VitalPBX we recommend you do it first on Server 1, then do a bascul and do it again on Server 2.

## 4.-0.- Some sources of information

1.- https://tunnelix.com/simple-master-master-replication-on-mariadb/

2.- https://www.digitalocean.com/community/tutorials/how-to-mirror-local-and-remote-directories-on-a-vps-with-lsyncd

3.- https://clusterlabs.org/pacemaker/doc/en-US/Pacemaker/1.1/html/Clusters\_from\_Scratch/\_configuring\_corosync.html

4.- Google Search