VitalPBX

High Availability

CONTENTS

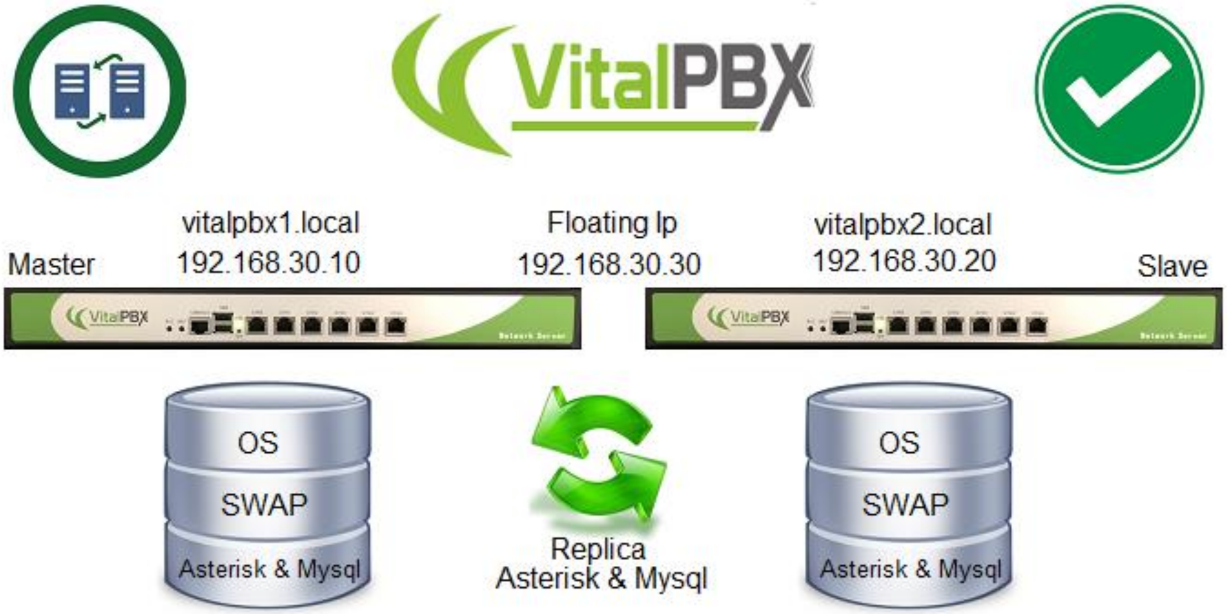
Contents

VitalPBX High Availability

1.- Introduction

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



2.- Prerequisites

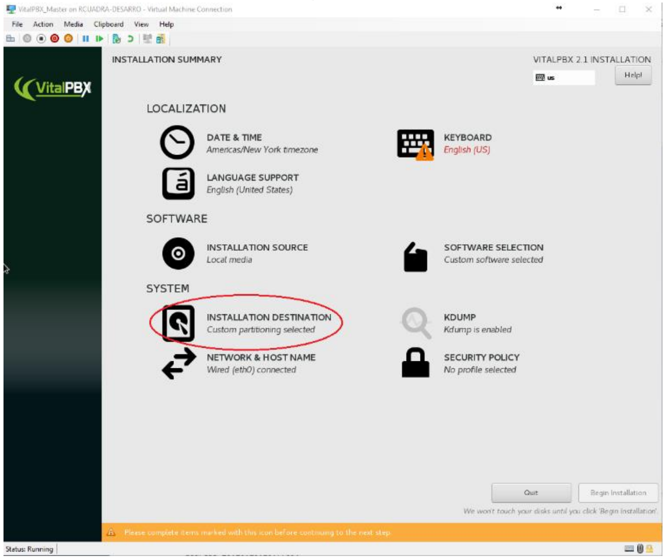
In order to install VitalPBX in high availability you need the following:

a.- 3 IP addresses.
b.- Install VitalPBX on two servers with similar characteristics.
c.- At the time of installation leave the largest amount of space on the hard drive to store the variable data on both servers.

3.- Installation

We are going to start by installing VitalPBX on two servers

a.- When starting the installation go to:
INSTALLATION DESTINATION (Custom partitioning selected)



INSTALLATION SUMMARY

VITALPBX 2.1 INSTALLATION

LOCALIZATION

DATE & TIME
Americas/New York timezone

KEYBOARD
English (US)

LANGUAGE SUPPORT
English (United States)

SOFTWARE

INSTALLATION SOURCE
Local media

SOFTWARE SELECTION
Custom software selected

SYSTEM

INSTALLATION DESTINATION
Custom partitioning selected

KDUMP
Kdump is enabled

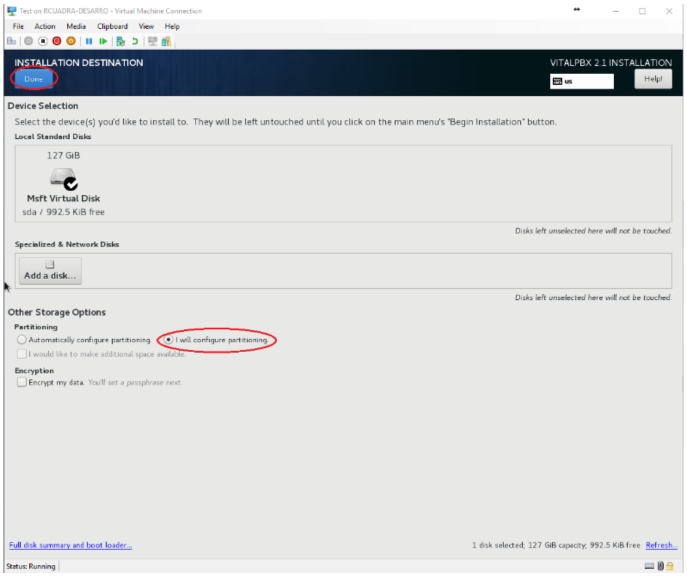
NETWORK & HOST NAME
Wired (eth0) connected

SECURITY POLICY
No profile selected

We won't touch your disks until you click 'Begin Installation'.

Please complete items marked with this icon before continuing to the next step.

b.- Select:
I will configure partitioning
And press the button
Done


Test on RCUADRA-DESARRO - Virtual Machine Connection
File
Action
Media
Clipboard
View
Help
Done
us
Help!
Add a disk...
Full disk summary and boot loader...
Refresh...
Status: Running

INSTALLATION DESTINATION

VITALPBX 2.1 INSTALLATION

Device Selection

Select the device(s) you'd like to install to. They will be left untouched until you click on the main menu's "Begin Installation" button.

Local Standard Disks

127 GiB
Msft Virtual Disk
sda / 992.5 KiB free

Disks left unselected here will not be touched.

Specialized & Network Disks

Disks left unselected here will not be touched.

Other Storage Options

Partitioning
○ Automatically configure partitioning. ● I will configure partitioning.
□ I would like to make additional space available.

Encryption
□ Encrypt my data. You'll set a passphrase next.

1 disk selected; 127 GiB capacity; 992.5 KiB free

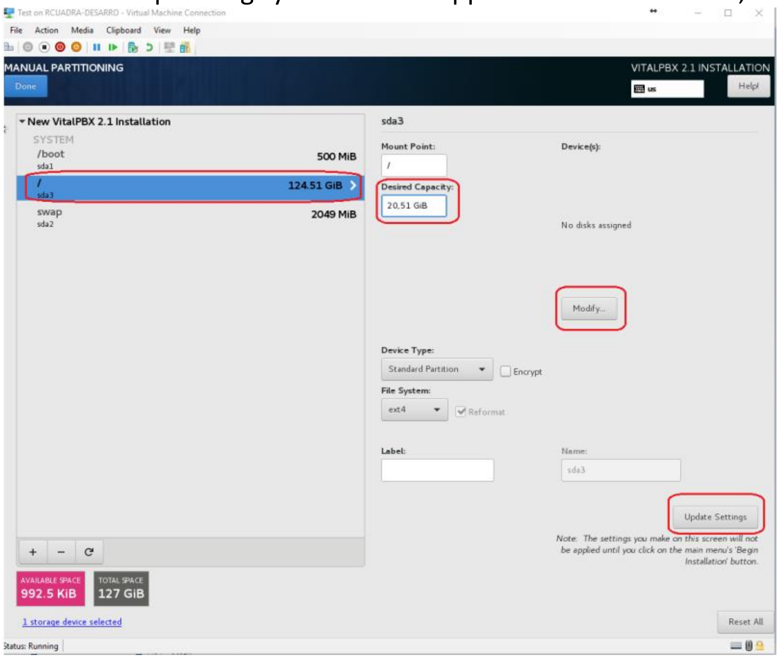
c.- Select the root partition:

/

Change the capacity to:

Desired Capacity: 20GB

We need enough space for the operating system and its applications in the future; then click


Test on RCUADRA-DESARRO - Virtual Machine Connection
File
Action
Media
Clipboard
View
Help
MANUAL PARTITIONING
Done
VITALPBX 2.1 INSTALLATION
us
Help!
+
−
Reset All

New VitalPBX 2.1 Installation

SYSTEM

/boot sda1	500 MiB
/ sda3	124.51 GiB
swap sda2	2049 MiB

sda3

Mount Point: /

Desired Capacity: 20,51 GiB

Device(s): No disks assigned

Modify...

Device Type: Standard Partition □ Encrypt

File System: ext4 ☑ Reformat

Label:

Name: sda3

Update Settings

Note: The settings you make on this screen will not be applied until you click on the main menu's 'Begin Installation' button.

AVAILABLE SPACE 992.5 KiB

TOTAL SPACE 127 GiB

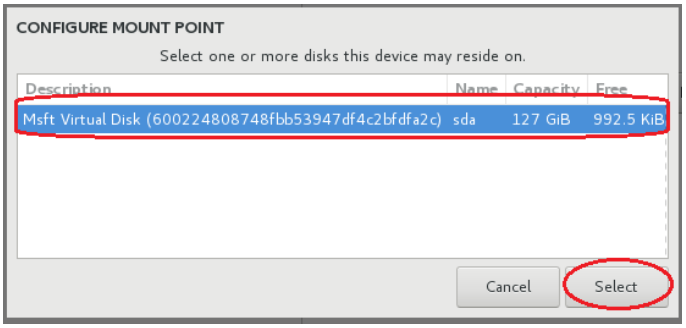
1 storage device selected

Status: Running

“Modify…” button

Select the disk and press

Select

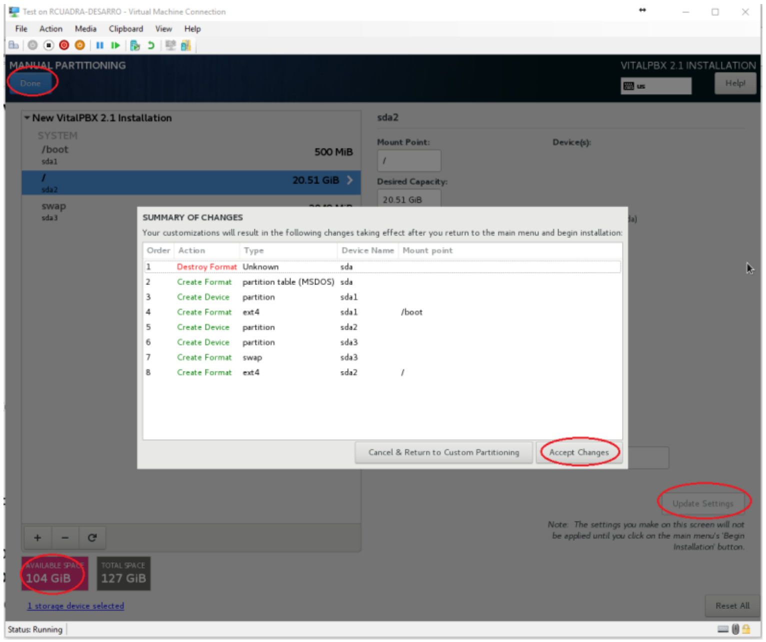


CONFIGURE MOUNT POINT

Select one or more disks this device may reside on.

Description	Name	Capacity	Free
Msft Virtual Disk (600224808748fbb53947df4c2bfdfa2c)	sda	127 GiB	992.5 KiB

Update Settings



SUMMARY OF CHANGES

Your customizations will result in the following changes taking effect after you return to the main menu and begin installation:

Order	Action	Type	Device Name	Mount point
1	Destroy Format	Unknown	sda	
2	Create Format	partition table (MSDOS)	sda	
3	Create Device	partition	sda1	
4	Create Format	ext4	sda1	/boot
5	Create Device	partition	sda2	
6	Create Device	partition	sda3	
7	Create Format	swap	sda3	
8	Create Format	ext4	sda2	/

Note: The settings you make on this screen will not be applied until you click on the main menu's 'Begin Installation' button.

Finally, we press:
Done
And press:
Accept Changes
And continue with the installation.

4.- Configurations

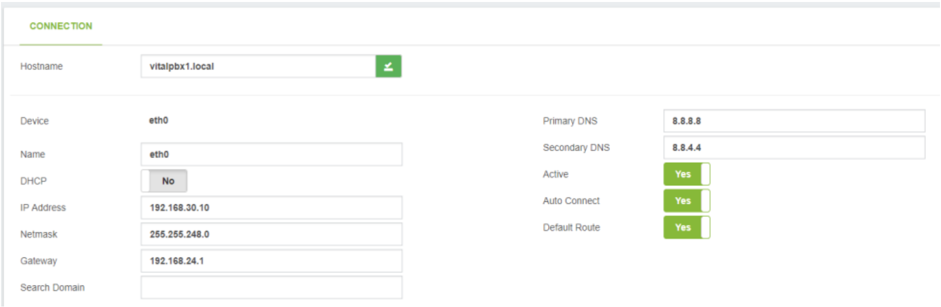
4.1- IP and Hostname Configuration.

We will configure in each server the IP address and the host name.
Go to the web interface to:
Admin>System Settings>Network Settings

First, change the Hostname, and remember to press the **Check button**.
Disable the DHCP option and we will set these values

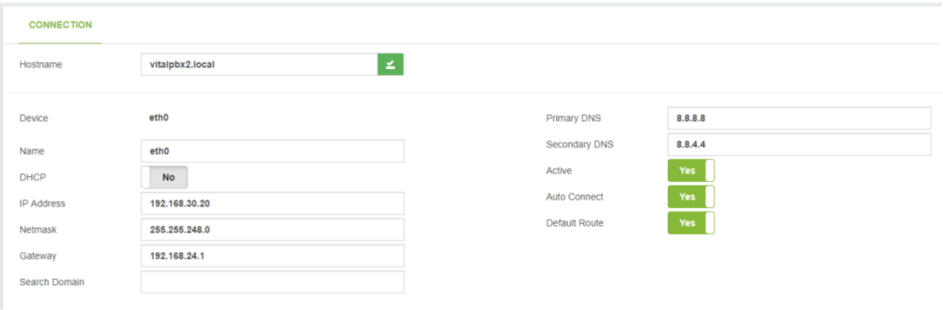
Name	Master	Slave
Hostname	vitalpbx1.local	vitalpbx2.local
IP Address	192.168.30.10	192.168.30.20
Netmask	255.255.248.0	255.255.248.0
Gateway	192.168.24.1	192.168.24.1
Primary DNS	8.8.8.8	8.8.8.8
Secondary DNS	8.8.4.4	8.8.4.4

Master



Hostname	vitalpbx1.local		
Device	eth0	Primary DNS	8.8.8.8
Name	eth0	Secondary DNS	8.8.4.4
DHCP	No	Active	Yes
IP Address	192.168.30.10	Auto Connect	Yes
Netmask	255.255.248.0	Default Route	Yes
Gateway	192.168.24.1		
Search Domain			

Slave



Hostname	vitalpbx2.local		
Device	eth0	Primary DNS	8.8.8.8
Name	eth0	Secondary DNS	8.8.4.4
DHCP	No	Active	Yes
IP Address	192.168.30.20	Auto Connect	Yes
Netmask	255.255.248.0	Default Route	Yes
Gateway	192.168.24.1		
Search Domain			

4.2.- Hostname

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

```
[root@vitalpbx1-2 ~]# vi /etc/hosts
192.168.30.10 vitalpbx1.local
192.168.30.20 vitalpbx2.local
```

4.3.- Create the partition on both servers

Initialize the partition to allocate the available space on the hard disk. Do these on both servers.

```
[root@vitalpbx1-2 ~]# fdisk /dev/sda
Command (m for help): n
Partition type:
  p   primary (3 primary, 0 extended, 1 free)
  e   extended
Select (default e): p
Selected partition 4 (take note of the assigned partition number as we will need it later)
First sector (35155968-266338303, default 35155968): [Enter]
Last sector, +sectors or +size{K,M,G} (35155968-266338303, default 266338303): [Enter]
Using default value 266338303
Partition 4 of type Linux and of size 110.2 GiB is set
Command (m for help): w
```

Then, restart the servers so that the new table is available.

```
[root@vitalpbx1-2 ~]# reboot
```

4.4.- Format the partition

Now, we will proceed to format the new partition in both servers with the following command:

```
[root@vitalpbx1-2 ~]# mke2fs -j /dev/sda4
[root@vitalpbx1-2 ~]# dd if=/dev/zero bs=1M count=500 of=/dev/sda4; sync
```

4.5.- Firewall

Adjust the firewall using the following commands:

```
[root@ vitalpbx1-2 ~]# firewall-cmd --permanent --add-service=high-availability
```

Just in Server1

```
[root@ vitalpbx1 ~]# firewall-cmd --permanent --add-rich-rule='rule family="ipv4" source
address="192.168.30.20" port port="7789" protocol="tcp" accept'
```

Just in Server2

```
[root@ vitalpbx2 ~]# firewall-cmd --permanent --add-rich-rule='rule family="ipv4" source
address="192.168.30.10" port port="7789" protocol="tcp" accept'
```

In both Servers

```
[root@ vitalpbx1-2 ~]# firewall-cmd --reload
```

4.-6.- Install Dependencies

Install the necessary dependencies on both servers

```
[root@ vitalpbx1-2 ~]# yum -y install drbd90-utils kmod-drbd90 corosync pacemaker pcs
```

4.7.- Configuring DRBD

Load the module and enable the service on both nodes, using the follow command:

```
[root@vitalpbx1-2 ~]# modprobe drbd
[root@vitalpbx1-2 ~]# systemctl enable drbd.service
```

Create a new global_common.conf file on both nodes with the following contents:

```
[root@ vitalpbx1-2 ~]# mv /etc/drbd.d/global_common.conf /etc/drbd.d/global_common.conf.orig
[root@ vitalpbx1-2 ~]# vi /etc/drbd.d/global_common.conf
global {
  usage-count no;
}
  common {
net {
  protocol C;
   }
}
```

Next, we will need to create a new configuration file called /etc/drbd.d/drbd0.res for the new resource named drbd0, with the following contents:

```
[root@ vitalpbx1-2 ~]# vi /etc/drbd.d/drbd0.res
resource drbd0 {
protocol C;
     on vitalpbx1.local {
          device /dev/drbd0;
          disk /dev/sda4;
          address 192.168.30.10:7789;
          meta-disk internal;
     }
     on vitalpbx2.local {
          device /dev/drbd0;
          disk /dev/sda4;
          address 192.168.30.20:7789;
          meta-disk internal;
     }
}
```

Initialize the meta data storage on each nodes by executing the following command on both nodes

```
[root@vitalpbx1-2 ~]#  drbdadm create-md drbd0
Writing meta data...
New drbd meta data block successfully created.
```

Let’s define the DRBD Primary node as first node “vitalpbx1”

```
[root@vitalpbx1 ~]# drbdadm up drbd0
[root@vitalpbx1 ~]# drbdadm primary drbd0 --force
```

On the Secondary node “vitalpbx2” run the following command to start the drbd0

```
[root@vitalpbx2 ~]# drbdadm up drbd0
```

You can check the current status of the synchronization while it’s being performed. The ***cat /proc/drbd*** command displays the creation and synchronization progress of the resource, as shown here:

4.7.1.- Test the DRBD

In order to test the DRBD functionality we need to Create a file system, mount the volume and write some data on primary node “vitalpbx1” and finally switch the primary node to “vitalpbx2”

Run the following command on the primary node to create an xfs filesystem on /dev/drbd0 and mount it to the mnt directory, using the following commands

```
[root@ vitalpbx1 ~]# mkfs.xfs /dev/drbd0
[root@ vitalpbx1 ~]# mount /dev/drbd0 /mnt
```

Create some data using the following command:

```
[root@ vitalpbx1 ~]# touch /mnt/file{1..5}
[root@ vitalpbx1 ~]# ls -l /mnt/
total 0
-rw-r--r-- 1 root root 0 Nov 17 11:28 file1
-rw-r--r-- 1 root root 0 Nov 17 11:28 file2
-rw-r--r-- 1 root root 0 Nov 17 11:28 file3
-rw-r--r-- 1 root root 0 Nov 17 11:28 file4
-rw-r--r-- 1 root root 0 Nov 17 11:28 file5
```

Let’s now switch primary mode “vitalpbx1” to second node “vitalpbx2” to check the data replication works or not.
First, we have to unmount the volume drbd0 on the first drbd cluster node “vitalpbx1” and change the primary node to secondary node on the first drbd cluster node “vitalpbx1”

```
[root@ vitalpbx1 ~]# umount /mnt
[root@ vitalpbx1 ~]# drbdadm secondary drbd0
```

Change the secondary node to primary node on the second drbd cluster node “vitalpbx2”

```
[root@ vitalpbx2 ~]# drbdadm primary drbd0 --force
```

Mount the volume and check the data available or not.

```
[root@ vitalpbx2 ~]# mount /dev/drbd0 /mnt
[root@ vitalpbx2 ~]# ls -l  /mnt
total 0
-rw-r--r-- 1 root root 0 Nov 17 11:28 file1
-rw-r--r-- 1 root root 0 Nov 17 11:28 file2
-rw-r--r-- 1 root root 0 Nov 17 11:28 file3
-rw-r--r-- 1 root root 0 Nov 17 11:28 file4
-rw-r--r-- 1 root root 0 Nov 17 11:28 file5
```

Normalize
Server1

```
[root@ vitalpbx2 ~]# umount /mnt
[root@ vitalpbx2 ~]# drbdadm secondary drbd0
```

Server2

```
[root@ vitalpbx1 ~]# drbdadm primary drbd0
[root@ vitalpbx1 ~]# mount /dev/drbd0 /mnt
```

4.8.- Configure Cluster

Create the password of the hacluster user on both nodes

```
[root@ vitalpbx1-2 ~]# echo MyPassword | passwd --stdin hacluster
```

Start PCS on both servers

```
[root@ vitalpbx1-2 ~]# systemctl start pcsd
```

Configure the start of services on both nodes

```
[root@ vitalpbx1-2 ~]# systemctl enable pcsd.service
[root@ vitalpbx1-2 ~]# systemctl enable corosync.service
[root@ vitalpbx1-2 ~]# systemctl enable pacemaker.service
```

Server Authenticate in Master

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

Create the cluster and configure parameters, perform only on the server1

```
[root@ vitalpbx1 ~]# pcs cluster setup --name cluster_voip vitalpbx1.local vitalpbx2.local
```

Starting Cluster in Master

```
[root@ vitalpbx1 ~]# pcs cluster start --all
[root@ vitalpbx1 ~]# pcs cluster enable --all
[root@ vitalpbx1 ~]# pcs property set stonith-enabled=false
[root@ vitalpbx1 ~]# pcs property set no-quorum-policy=ignore
```

Create resource for the use of Floating IP

```
[root@ vitalpbx1 ~]# pcs resource create virtual_ip ocf:heartbeat:IPaddr2 ip=192.168.30.30
cidr_netmask=21 op monitor interval=30s on-fail=restart
[root@ vitalpbx1 ~]# pcs cluster cib drbd_cfg
[root@ vitalpbx1 ~]# pcs cluster cib-push drbd_cfg
```

Create resource for the use of DRBD

```
[root@ vitalpbx1 ~]# pcs -f drbd_cfg resource create DrbdData ocf:linbit:drbd
drbd_resource=drbd0 op monitor interval=60s
[root@ vitalpbx1 ~]# pcs -f drbd_cfg resource master DrbdDataClone DrbdData master-max=1
master-node-max=1 clone-max=2 clone-node-max=1 notify=true
[root@ vitalpbx1 ~]# pcs cluster cib-push drbd_cfg
```

Create FILESYSTEM resource for the automated mount point

```
[root@ vitalpbx1 ~]# pcs  -f fs_cfg resource create DrbdFS Filesystem device="/dev/drbd0"
directory="/mnt" fstype="xfs"
[root@ vitalpbx1 ~]# pcs  -f fs_cfg constraint colocation add DrbdFS with DrbdDataClone
INFINITY with-rsc-role=Master
[root@ vitalpbx1 ~]# pcs  -f fs_cfg constraint order promote DrbdDataClone then start DrbdFS
[root@ vitalpbx1 ~]# pcs -f fs_cfg constraint colocation add DrbdFS with virtual_ip INFINITY
[root@ vitalpbx1 ~]# pcs -f fs_cfg constraint order virtual_ip then DrbdFS
[root@ vitalpbx1 ~]# pcs cluster cib-push fs_cfg
```

Stop and disable all services in both servers

```
[root@ vitalpbx1-2 ~]# systemctl stop fail2ban
[root@ vitalpbx1-2 ~]# systemctl disable fail2ban
[root@ vitalpbx1-2 ~]# systemctl stop asterisk
[root@ vitalpbx1-2 ~]# systemctl disable asterisk
[root@ vitalpbx1-2 ~]# systemctl stop vpbx-monitor
[root@ vitalpbx1-2 ~]# systemctl disable vpbx-monitor
[root@ vitalpbx1-2 ~]# systemctl stop mariadb
[root@ vitalpbx1-2 ~]# systemctl disable mariadb
```

Create resource for the use of MariaDB in Master

```
[root@ vitalpbx1 ~]# mkdir /mnt/mysql
[root@ vitalpbx1 ~]# mkdir /mnt/mysql/data
[root@ vitalpbx1 ~]# cd /mnt/mysql
[root@ vitalpbx1 ~]# cp -aR /var/lib/mysql/* /mnt/mysql/data
[root@ vitalpbx1-2 ~]# sed -i 's/var\/lib\/mysql/mnt\/mysql\/data/g' /etc/my.cnf
[root@ vitalpbx1 ~]# mv /etc/my.cnf /mnt/mysql/
[root@ vitalpbx2 ~]# rm -rf /etc/my.cnf
[root@ vitalpbx1-2 ~]# ln -s /mnt/mysql/my.cnf /etc/
[root@ vitalpbx1 ~]# pcs resource create mysql ocf:heartbeat:mysql
binary="/usr/bin/mysqld_safe" config="/etc/my.cnf" datadir="/mnt/mysql/data"
pid="/var/lib/mysql/mysql.pid" socket="/var/lib/mysql/mysql.sock" additional_parameters="--
bind-address=0.0.0.0" op start timeout=60s op stop timeout=60s op monitor interval=20s
timeout=30s on-fail=standby
[root@ vitalpbx1 ~]# pcs cluster cib fs_cfg
[root@ vitalpbx1 ~]# pcs cluster cib-push fs_cfg --config
[root@ vitalpbx1 ~]# pcs -f fs_cfg constraint colocation add mysql with virtual_ip INFINITY
[root@ vitalpbx1 ~]# pcs -f fs_cfg constraint order DrbdFS then mysql
[root@ vitalpbx1 ~]# pcs cluster cib-push fs_cfg --config
```

Create resource for Asterisk

```
[root@ vitalpbx1 ~]# pcs resource create asterisk service:asterisk op monitor interval=30s
[root@ vitalpbx1 ~]# pcs cluster cib fs_cfg
[root@ vitalpbx1 ~]# pcs cluster cib-push fs_cfg --config
[root@ vitalpbx1 ~]# pcs -f fs_cfg constraint colocation add asterisk with virtual_ip INFINITY
[root@ vitalpbx1 ~]# pcs -f fs_cfg constraint order mysql then asterisk
[root@ vitalpbx1 ~]# pcs cluster cib-push fs_cfg --config
```

Copy folders and files the DRBD partition on the server1

```
[root@ vitalpbx1 ~]# cd /mnt/
[root@ vitalpbx1 ~]# tar -zcvf var-asterisk.tgz /var/log/asterisk
[root@ vitalpbx1 ~]# tar -zcvf var-lib-asterisk.tgz /var/lib/asterisk
[root@ vitalpbx1 ~]# tar -zcvf usr-lib64-asterisk.tgz /usr/lib64/asterisk
[root@ vitalpbx1 ~]# tar -zcvf var-spool-asterisk.tgz /var/spool/asterisk
[root@ vitalpbx1 ~]# tar -zcvf etc-asterisk.tgz /etc/asterisk

[root@ vitalpbx1 ~]# tar xvfz var-asterisk.tgz
[root@ vitalpbx1 ~]# tar xvfz var-lib-asterisk.tgz
[root@ vitalpbx1 ~]# tar xvfz usr-lib64-asterisk.tgz
[root@ vitalpbx1 ~]# tar xvfz var-spool-asterisk.tgz
[root@ vitalpbx1 ~]# tar xvfz etc-asterisk.tgz

[root@ vitalpbx1 ~]# rm -rf /var/log/asterisk
[root@ vitalpbx1 ~]# rm -rf /var/lib/asterisk
[root@ vitalpbx1 ~]# rm -rf /usr/lib64/asterisk/
[root@ vitalpbx1 ~]# rm -rf /var/spool/asterisk/
[root@ vitalpbx1 ~]# rm -rf /etc/asterisk

[root@ vitalpbx1 ~]# ln -s /mnt/var/log/asterisk /var/log/asterisk
[root@ vitalpbx1 ~]# ln -s /mnt/var/lib/asterisk /var/lib/asterisk
[root@ vitalpbx1 ~]# ln -s /mnt/usr/lib64/asterisk /usr/lib64/asterisk
[root@ vitalpbx1 ~]# ln -s /mnt/var/spool/asterisk /var/spool/asterisk
[root@ vitalpbx1 ~]# ln -s /mnt/etc/asterisk /etc/asterisk
```

Configure symbolic links on the server2

```
[root@ vitalpbx2 ~]# rm -rf /var/log/asterisk
[root@ vitalpbx2 ~]# rm -rf /var/lib/asterisk
[root@ vitalpbx2 ~]# rm -rf /usr/lib64/asterisk/
[root@ vitalpbx2 ~]# rm -rf /var/spool/asterisk/
[root@ vitalpbx2 ~]# rm -rf /etc/asterisk

[root@ vitalpbx2 ~]# ln -s /mnt/var/log/asterisk /var/log/asterisk
[root@ vitalpbx2 ~]# ln -s /mnt/var/lib/asterisk /var/lib/asterisk
[root@ vitalpbx2 ~]# ln -s /mnt/usr/lib64/asterisk /usr/lib64/asterisk [root@ vitalpbx2 ~]# ln
-s /mnt/var/spool/asterisk /var/spool/asterisk [root@ vitalpbx2 ~]# ln -s /mnt/etc/asterisk
/etc/asterisk
```

Create VitalPBX Service

```
[root@ vitalpbx1 ~]# pcs resource create vpbx-monitor service:vpbx-monitor op monitor
interval=30s
[root@ vitalpbx1 ~]# pcs cluster cib fs_cfg
[root@ vitalpbx1 ~]# pcs cluster cib-push fs_cfg
[root@ vitalpbx1 ~]# pcs -f fs_cfg constraint colocation add vpbx-monitor with virtual_ip
INFINITY
[root@ vitalpbx1 ~]# pcs -f fs_cfg constraint order asterisk then vpbx-monitor
[root@ vitalpbx1 ~]# pcs cluster cib-push fs_cfg
```

Create fail2ban Service

```
[root@ vitalpbx1 ~]# pcs resource create fail2ban service:fail2ban op monitor interval=30s
[root@ vitalpbx1 ~]# pcs cluster cib fs_cfg
[root@ vitalpbx1 ~]# pcs cluster cib-push fs_cfg
[root@ vitalpbx1 ~]# pcs -f fs_cfg constraint colocation add fail2ban with virtual_ip INFINITY
[root@ vitalpbx1 ~]# pcs -f fs_cfg constraint order asterisk then fail2ban
[root@ vitalpbx1 ~]# pcs cluster cib-push fs_cfg
```

Initialize the services of corosync and pacemaker in server2

```
[root@ vitalpbx2 ~]# systemctl restart corosync.service
[root@ vitalpbx2 ~]# systemctl restart pacemaker.service
```

Show the Cluster status

```
[root@ vitalpbx1 ~]# pcs status resources
virtual_ip     (ocf::heartbeat:IPaddr2):       Started vitalpbx1.local
 Master/Slave Set: DrbdDataClone [DrbdData]
     Masters: [ vitalpbx1.local ]
     Slaves: [ vitalpbx2.local ]
 DrbdFS (ocf::heartbeat:Filesystem):    Started vitalpbx1.local
 mysql  (ocf::heartbeat:mysql): Started vitalpbx1.local
 asterisk       (service:asterisk):     Started vitalpbx1.local
 vpbx-monitor   (service:vpbx-monitor): Started vitalpbx1.local
 fail2ban       (service:fail2ban):     Started vitalpbx1.local

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

4.9.- Create “bascul” command in both servers

```
[root@ vitalpbx1-2 ~]# vi bascul
#!/bin/bash
set -e
# Authors:      Rodrigo Cuadra
#               with Collaboration of Jose Miguel Rivera
#
# Support:      rcuadra@aplitel.com
#
host_master=`pcs status | awk '/Masters/ {print $3}'`
host_slave=`pcs status | awk '/Slaves/ {print $3}'`

        if [ "$host_master" = "" ] ;then
               echo -e "There are problems with high availability, please check with the
command "pcs status" (we recommend applying the command "pcs cluster unstandby" in both
servers)"
               exit;
       fi

        if [ "$host_slave" = "" ] ;then
                echo -e "There are problems with high availability, please check with the
command "pcs status" (we recommend applying the command "pcs cluster unstandby" in both
servers)"
                exit;
        fi

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 "*************************************************************"
       while [[ $veryfy_bascul != yes && $veryfy_bascul != no ]]
       do
               read -p "Are you sure to switch from $host_master to $host_slave? (yes,no) > "
veryfy_bascul
       done
       if [ "$veryfy_bascul" = "yes" ] ;then
                BAR='........................................................'
                i=0
               pcs cluster unstandby $host_master
               pcs cluster unstandby $host_slave
               pcs cluster standby $host_master
               host_master_new=$host_master
               echo -e "Stopping services on " $host_master
               while [[ $host_master == $host_master_new ]]
               do
                        i=$i+1
                        echo -ne "\r${BAR:0:$i}"
                      sleep 1
                      host_master_new=`pcs status | awk '/Masters/ {print $3}'`
               done
                echo -e "Done"
                echo -e "Starting services on " $host_slave

               virtualip=`pcs status | awk '/virtual_ip/ {print $3}'`
                      i=0
               while [[ $virtualip != 'Started' ]]
               do
                        i=$i+1
                        echo -ne "\r${BAR:0:$i}"
                      sleep 1
                      virtualip=`pcs status | awk '/virtual_ip/ {print $3}'`
               done
               echo -e "Virtual IP: Started on " $host_slave

               Filesystem=`pcs status | awk '/DrbdFS/ {print $3}'`
               while [[ $Filesystem != 'Started' ]]
               do
```

```
                        Filesystem=`pcs status | awk '/DrbdFS/ {print $3}'`
                done
                echo -e "Filesystem: Started on " $host_slave

                mysql=`pcs status | awk '/mysql/ {print $3}'`
                while [[ $mysql != 'Started' ]]
                do
                        mysql=`pcs status | awk '/mysql/ {print $3}'`
                done
                echo -e "Mysql: Started on " $host_slave

                asterisk=`pcs status | awk '/asterisk/ {print $3}'`
                while [[ $asterisk != 'Started' ]]
                do
                        asterisk=`pcs status | awk '/asterisk/ {print $3}'`
                done
                echo -e "Asterisk: Started on " $host_slave

                vpbx_monitor=`pcs status | awk '/vpbx-monitor/ {print $3}'`
                while [[ $vpbx_monitor != 'Started' ]]
                do
                        vpbx_monitor=`pcs status | awk '/vpbx-monitor/ {print $3}'`
                done
                echo -e "VitalPBX Monitor: Started on " $host_slave

                fail2ban=`pcs status | awk '/fail2ban/ {print $3}'`
                while [[ $fail2ban != 'Started' ]]
                do
                        fail2ban=`pcs status | awk '/fail2ban/ {print $3}'`
                done
                echo -e "fail2ban: Started on " $host_slave

                pcs cluster unstandby $host_master
                sleep 3
                echo -e "Done,"
                pcs status
        else
                echo -e "Nothing to do, bye, bye"
        fi
```

Add permissions and move to folder /usr/local/bin

```
[root@ vitalpbx1-2 ~]# chmod +x bascul
[root@ vitalpbx1-2 ~]# mv bascul /usr/local/bin
```

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

Stopping services on  vitalpbx1.local
..................Done
Starting services on  vitalpbx2.local
...Virtual IP: Started on  vitalpbx2.local
Filesystem: Started on  vitalpbx2.local
Mysql: Started on  vitalpbx2.local
Asterisk: Started on  vitalpbx2.local
VitalPBX Monitor: Started on  vitalpbx2.local
fail2ban: Started on  vitalpbx2.local
Done,
Cluster name: cluster_voip
Stack: corosync
Current DC: vitalpbx1.local (version 1.1.18-11.el7_5.3-2b07d5c5a9) - partition with quorum
Last updated: Wed Nov 28 10:00:56 2018
Last change: Wed Nov 28 10:00:52 2018 by root via cibadmin on vitalpbx2.local

2 nodes configured
8 resources configured

Online: [ vitalpbx1.local vitalpbx2.local ]

Full list of resources:

virtual_ip     (ocf::heartbeat:IPaddr2):       Started vitalpbx2.local
Master/Slave Set: DrbdDataClone [DrbdData]
     Masters: [ vitalpbx2.local ]
     Slaves: [ vitalpbx1.local ]
DrbdFS (ocf::heartbeat:Filesystem):    Started vitalpbx2.local
mysql  (ocf::heartbeat:mysql): Started vitalpbx2.local
asterisk       (service:asterisk):     Started vitalpbx2.local
vpbx-monitor   (service:vpbx-monitor): Started vitalpbx2.local
fail2ban       (service:fail2ban):     Started vitalpbx2.local

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

5.- Test

To execute the process of changing the role, we recommend using the following command:

```
[root@vitalpbx1-2 /]# 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) >
```

This action convert the vitalpbx1.local to Slave and vitalpbx2.local to Master. If you want to return to default do the same again.

6.- Turn on and turn off

When you must turn off the servers, when you turn it on always start with the Master, wait for the Master to start and then turn on the Slave.

7.- Update

To update VitalPBX to the latest version just follow the following steps:
1.- From your browser, go to ip 192.168.30.30
2.- Update VitalPBX from the interface
3.- Execute the following command in Master console

```
[root@vitalpbx1 /]# bascul
```

4.- From your browser, go to ip 192.168.30.30 again
5.- Update VitalPBX from the interface
6.- Execute the following command in Master console

```
[root@vitalpbx1 /]# bascul
```

CONGRATULATIONS, you have installed and tested the high availability in VitalPBX

8.- Credits

8.1 Sources of Information

- voip-info.org
- asterisk.org
- DRBD Website (https://www.linbit.com/en/)
- Pacemaker Website (https://clusterlabs.org/pacemaker/)