

# ZABBIX HA CLUSTER SETUPS



Edmunds Vesmanis  
**ZABBIX** Senior Trainer

**ZABBIX** '19  
SUMMIT

# WHAT AND WHY?

ZABBIX  
HA CLUSTER SETUPS

# HA is a must.

**but...**

**Keep things simple**


**Tested, common and time proven solution**

**Open source components used**


**True HA starts with 3**

# A PLAN...


### Server nodes



zabbix-ha-srv1  
192.168.7.93  
CPU: 0  
RAM: 423.17 MB/991.35 MB  
Uptime: 22:39:16  
OK




zabbix-ha-srv2  
192.168.7.94  
CPU: 0  
RAM: 440.88 MB/991.35 MB  
Uptime: 22:39:58  
OK




zabbix-ha-srv3  
192.168.7.95  
CPU: 0  
RAM: 457.72 MB/991.35 MB  
Uptime: 22:40:39  
OK


### Database nodes



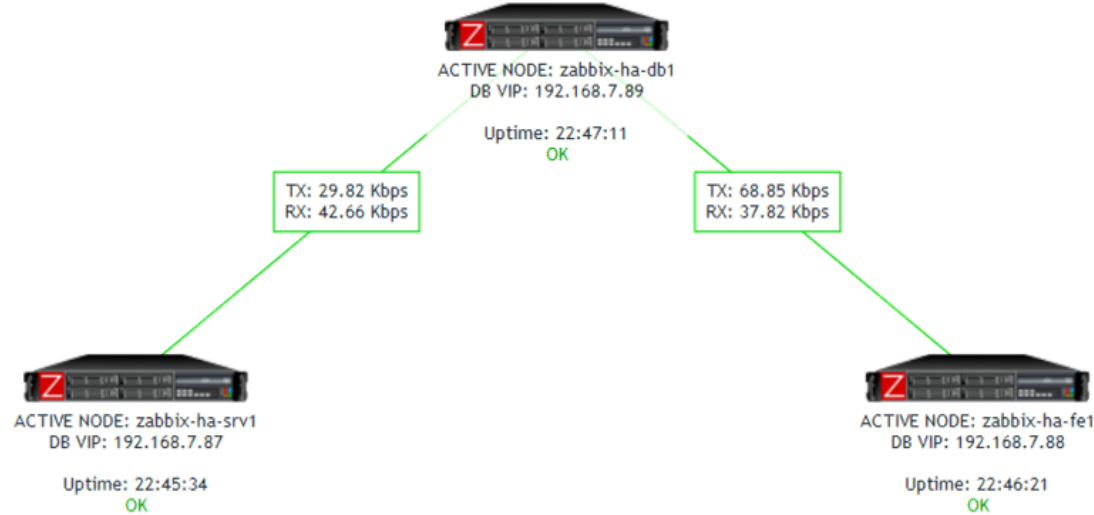
zabbix-ha-db1  
192.168.7.96  
CPU: 0.01  
RAM: 1.05 GB/1.8 GB  
Uptime: 22:47:18  
OK




zabbix-ha-db2  
192.168.7.97  
CPU: 0.005  
RAM: 1.07 GB/1.8 GB  
Uptime: 22:47:58  
OK




zabbix-ha-db3  
192.168.7.99  
CPU: 0.005  
RAM: 1.09 GB/1.8 GB  
Uptime: 22:38:40  
OK




### Frontend nodes



zabbix-ha-fe1  
192.168.7.90  
CPU: 0.01  
RAM: 356.61 MB/991.35 MB  
Uptime: 22:41:29  
OK



zabbix-ha-fe2  
192.168.7.91  
CPU: 0  
RAM: 415.71 MB/991.35 MB  
Uptime: 22:42:11  
OK



zabbix-ha-fe3  
192.168.7.92  
CPU: 0  
RAM: 430.77 MB/991.35 MB  
Uptime: 22:42:52  
OK

# A PLAN...

**bare minimum**

**no complex automation at first**

**understand the basics**

**nodes will switch automatically if basic resources die  
or connectivity problems**

**at first: manual control/override in case of problems**

# A PLAN...

The image shows a screenshot of the Zabbix HA management console. On the left, there are two server icons with their respective status information:

- Top server: zabbix-h, 192.168..., CPU: ..., RAM: 442.23 M, Uptime: 2, OK
- Bottom server: zabbix-h, 192.168..., CPU: ..., RAM: 456.71 M, Uptime: 2, OK

A menu is open over the top server, showing the following options:

- SCRIPTS
- Cluster CTRL (highlighted) ▶
- Ping
- GO TO
- Host inventory
- Latest data
- Problems
- Graphs
- Host screens
- URLS
- HA management console

The 'Cluster CTRL' menu is further expanded to show:

- DATABASE ▶
- FRONTEND ▶
- SERVER (highlighted) ▶

The 'SERVER' sub-menu is expanded to show:

- Move SRV Resource (highlighted)

On the right side of the console, the status of the active node is shown:

- ACTIVE NODE: zabbix-ha-fe1
- DB VIP: 192.168.7.88
- Uptime: 22:46:21
- OK

Green lines connect the 'Cluster CTRL' menu item to the 'SERVER' sub-menu, and the 'SERVER' sub-menu to the 'Move SRV Resource' option. Another green line connects the 'Move SRV Resource' option to the 'ACTIVE NODE' status information on the right.

# A PLAN...

## Server nodes



zabbix-ha-srv1  
192.168.7.93  
CPU: 0  
RAM: 453.74 MB/991.35 MB  
Uptime: 22:59:16  
OK



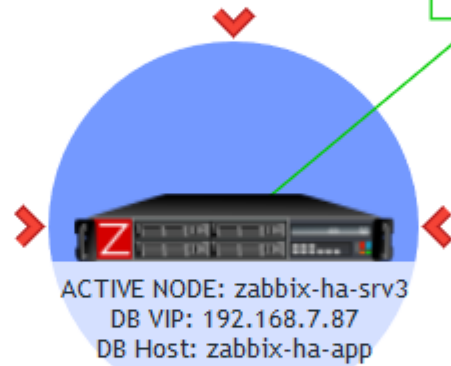
zabbix-ha-srv2  
192.168.7.94  
CPU: 0.01  
RAM: 412.81 MB/991.35 MB  
Uptime: 22:59:58  
OK



zabbix-ha-srv3  
192.168.7.95  
CPU: 0  
RAM: 428.06 MB/991.35 MB  
Uptime: 23:00:39  
OK

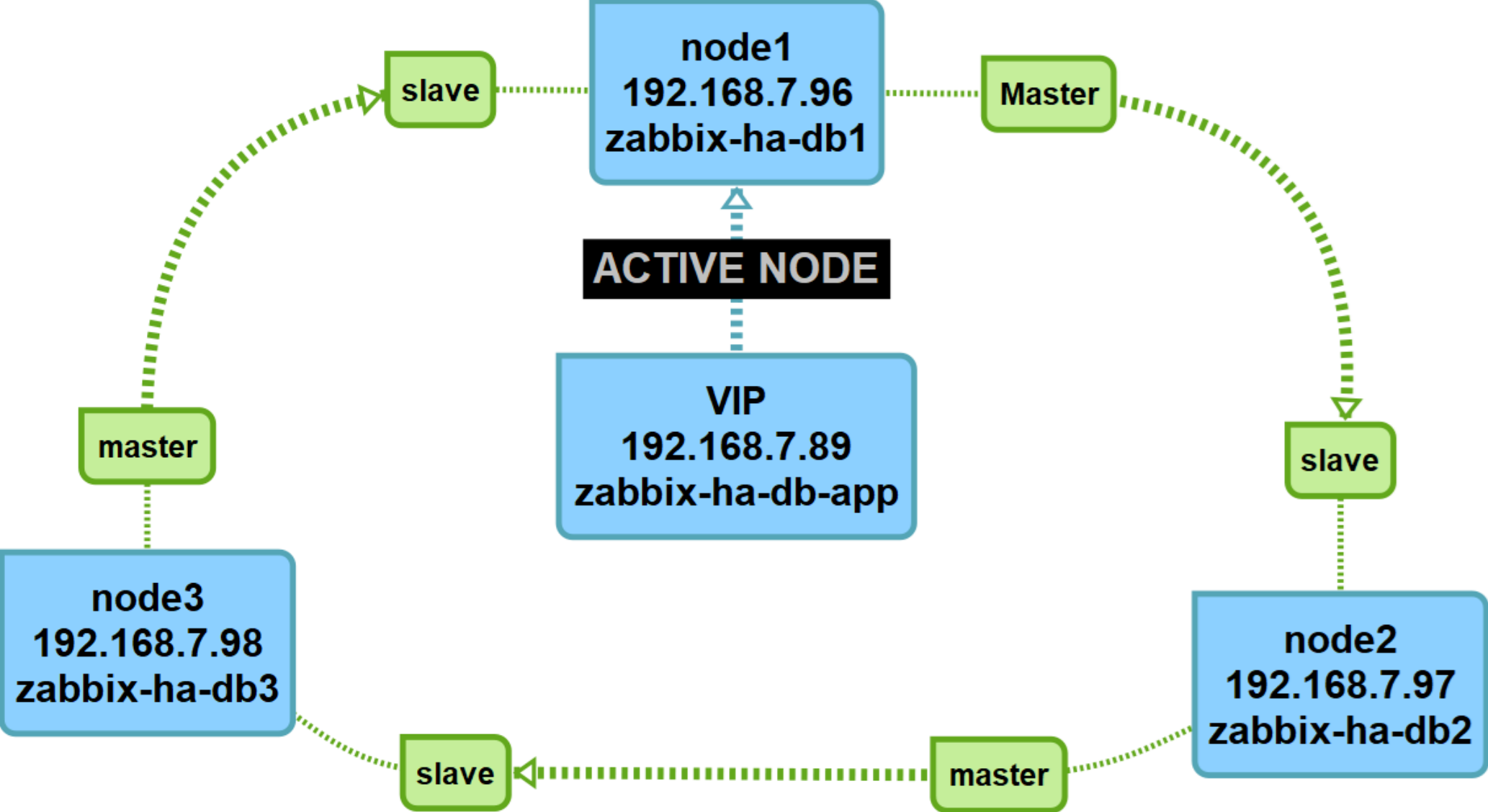


ACTIVE NODE: zabbix-ha-db1  
DB VIP: 192.168.7.89  
DB Host: zabbix-ha-db-app  
Uptime: 22:57:12  
OK



Uptime: 23:05:33  
Hostname was changed on Zabbix server VIP

# DB cluster setup





# all cluster ip's and hostnames

# VIPs for cluster:

192.168.7.87 zabbix-ha-app  
192.168.7.89 zabbix-ha-db-app  
192.168.7.88 zabbix-ha-fe-app

# IP's for nodes:

# DB nodes:

192.168.7.96 zabbix-ha-db1  
192.168.7.97 zabbix-ha-db2  
192.168.7.99 zabbix-ha-db3

# zabbix server nodes:

192.168.7.93 zabbix-ha-srv1  
192.168.7.94 zabbix-ha-srv2  
192.168.7.95 zabbix-ha-srv3

# Front-end nodes:

192.168.7.90 zabbix-ha-fe1  
192.168.7.91 zabbix-ha-fe2  
192.168.7.92 zabbix-ha-fe3

# VM preparations

**ntp (Time settings same on all nodes)**

**localization**

**firewall**

**selinux ... :-/**

**/etc/hosts: don't relay on DNS**

**Storage: Separate block devices for DB, logs, apps and configs)**

**Zabbix Agent on all nodes**

# DATABASE CLUSTER

ZABBIX  
HA CLUSTER SETUPS

# On all DB nodes:

```
## Install HA components:  
yum groupinstall 'High Availability' -y  
## OR:  
yum groupinstall ha -y  
  
## Create user for cluster:  
echo <CLUSTER_PASSWORD> | passwd --stdin hacluster
```

# On node1: cluster setup

```
# Authenticate cluster nodes:  
pcs cluster auth zabbix-ha-db1 zabbix-ha-db2 zabbix-ha-db3 \  
username: hacluster  
password: <CLUSTER_PASSWORD>
```

```
zabbix-ha-db1: Authorized  
zabbix-ha-db2: Authorized  
zabbix-ha-db3: Authorized
```

# On node1: cluster setup

```
# Create zabbix-db-cluster:  
pcs cluster setup --name zabbix_db_cluster \  
zabbix-ha-db1 zabbix-ha-db2 zabbix-ha-db3 -force  
  
## Create resource for cluster virtual IP (VIP)  
pcs resource create virtual_ip ocf:heartbeat:IPaddr2 \  
ip=192.168.7.89 op monitor interval=5s --group zabbix_db_cluster
```

# On node1: cluster setup

```
## check:  
pcs status
```

```
## Restart cluster services in case of:  
## "cluster is not currently running on this node" error  
pcs cluster stop --all && pcs cluster start --all
```

```
# in case you have a firewalld:  
firewall-cmd --permanent --add-service=high-availability && firewall-cmd --reload
```

# On node1: cluster setup

```
## Prevent Resources from Moving after Recovery
pcs resource defaults resource-stickiness=100

## if you are not using fencing disable STONITH:
pcs property set stonith-enabled=false

## otherwise resource won't start
## STONITH = Shoot the other node in the head!
```



# Did you know there is a GUI?!

The image shows a browser window with the URL `https://zabbix-ha-db1:2224/login`. The page header includes the Zabbix logo and the text "HIGH AVAILABILITY MANAGEMENT". A navigation menu contains the following items: "NODES", "RESOURCES", "FENCE DEVICES", "ACLS", "CLUSTER PROPERTIES", and "MANAGE CLUSTERS".

The main content area features a login form with the following fields and controls:

- Username:
- Password:
- Login button

A sub-menu is shown below the main page, featuring the Zabbix logo and "HIGH AVAILABILITY MANAGEMENT" text. The sub-menu has two items: "MANAGE CLUSTERS" and "PERMISSIONS".

Below the sub-menu, a "MANAGE CLUSTERS" section contains the following controls:

- MANAGE CLUSTERS**
- [✕ Remove](#)
- [+ Add Existing](#)
- [+ Create New](#)

# Cluster creation via pscd GUI

**Create Cluster** ✕

Enter the hostnames and ports of the nodes you would like to use to create a cluster:

Cluster Name:

Node 1:	<input type="text" value="zabbix-ha-db1"/>	:	<input type="text" value="2224"/>
Node 2:	<input type="text" value="zabbix-ha-db2"/>	:	<input type="text" value="2224"/>
Node 3:	<input type="text" value="zabbix-ha-db3"/>	:	<input type="text" value="2224"/>

[More nodes...](#)

▶ **Advanced Options:**

# Cluster creation via pscd GUI

**Add Existing Cluster** ✕

Enter the hostname/IP of a node in a cluster that you would like to manage:

Node Name/IP:

PCSD port:

# Cluster creation via pcsd GUI

The screenshot displays the pcsd web interface for managing a cluster. At the top, the header includes the Red Hat logo and 'HIGH AVAILABILITY MANAGEMENT'. The current cluster is identified as 'zabbix\_db\_cluster' and the user is logged in as 'hacluster'. A navigation menu contains tabs for 'NODES', 'RESOURCES', 'FENCE DEVICES', 'ACLS', 'CLUSTER PROPERTIES', and 'MANAGE CLUSTERS'. The 'NODES' tab is active, showing a list of three nodes: 'zabbix-ha-db1', 'zabbix-ha-db2', and 'zabbix-ha-db3'. The 'zabbix-ha-db1' node is selected, and its details are shown in a right-hand pane. This pane includes the node name 'zabbix-ha-db1', status indicators for 'Pacemaker Connected' and 'Corosync Connected', and control buttons for 'Start', 'Stop', 'Restart', 'Standby', 'Maintenance', and 'Configure Fencing'. The node's ID is 1 and its uptime is 0 days, 20:26:49.

[Red Hat Enterprise Linux7 >> High Availability Add-On Reference >> Chapter 2. The pcsd Web UI](#)

# MariaDB install and replication setup

```
## install MariaDB server on all 3 DB nodes:
```

```
yum install mariadb-server -y
```

```
## tune/configure db settings:
```

```
cp ./zabbixdb.cnf /etc/my.cnf.d/
```

```
## Start and enable to start on boot:
```

```
systemctl start mariadb
```

```
systemctl enable mariadb
```

```
## secure your installation and create <MYSQL_ROOT_PASSWORD>:
```

```
mysql_secure_installation
```

# MariaDB install and replication setup

```
cat zabbixdb.cnf
[mysqld]
# ZABBIX specific settings and tuning
default-storage-engine      = InnoDB
innodb                      = FORCE
innodb_file_per_table       = 1
innodb_buffer_pool_size     = 512M           # 50-75% of total RAM
innodb_buffer_pool_instances = 8           # For MySQL 5.5 - 4, for 5.6+ - 8
innodb_flush_log_at_trx_commit = 2
innodb_flush_method         = O_DIRECT
innodb_io_capacity          = 800           # HDD disks 500-800,    SSD disks - 2000
sync-binlog                 = 0
query-cache-size            = 0
server_id                   = 96           # for id settings IPs last number used
report_host                 = zabbix-ha-db1
log-slave-updates
log_bin                     = /var/lib/mysql/log-bin
log_bin_index               = /var/lib/mysql/log-bin.index
relay_log                   = /var/lib/mysql/relay-bin
relay_log_index             = /var/lib/mysql/relay-bin.index
binlog_format               = mixed
binlog_cache_size          = 64M
max_binlog_size             = 1G
expire_logs_days           = 5
binlog_checksum            = crc32
max_allowed_packet         = 500M
```

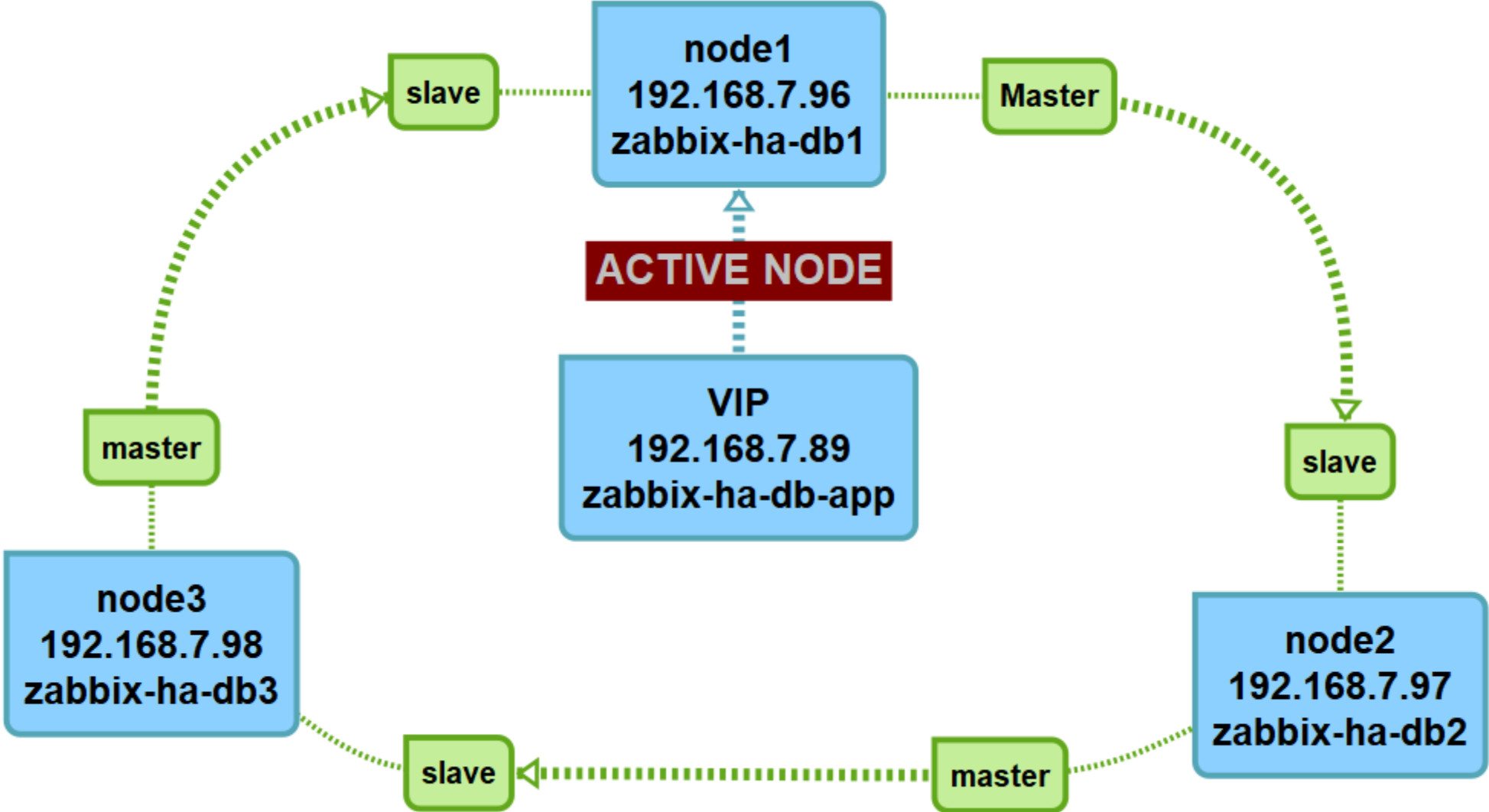
# MariaDB install and replication setup

```
## Must be set on every db node accordingly
```

```
vi /etc/my.cnf.d/zabbixdb.cnf
```

```
server_id          = 96          ## Last number of IP  
report_host       = zabbix-ha-db1 ## Hostname
```

# Remember the PLAN?!





# Replicaton setup: node1 (zabbix-ha-db1)

```
## Login to MySQL:
```

```
mysql -uroot -p<MYSQL_ROOT_PASSWORD>
```

```
MariaDB [(none)]> STOP SLAVE;
```

```
MariaDB [(none)]> GRANT REPLICATION SLAVE ON *.* TO 'replicator'@'<NODE2_IP>'
identified by '<REPLICATOR_PASSWORD>';
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
MariaDB [(none)]> SHOW MASTER STATUS\G
```

```
***** 1. row *****
```

```
File: log-bin.000001
```

```
Position: 245
```

```
Binlog_Do_DB:
```

```
Binlog_Ignore_DB:
```

```
1 row in set (0.00 sec)
```

# Replicaton setup: node2 (zabbix-ha-db2)

```
## Login to MySQL:
mysql -uroot -p<MYSQL_ROOT_PASSWORD>

STOP SLAVE;

CHANGE MASTER TO MASTER_HOST = '<NODE1_IP>', MASTER_USER = 'replicator',
MASTER_PASSWORD = '<REPLICATOR_PASSWORD>', MASTER_LOG_FILE='log-bin.000001',
MASTER_LOG_POS = 245;

GRANT REPLICATION SLAVE ON *.* TO 'replicator'@'<NODE3_IP>' identified by
'<REPLICATOR_PASSWORD>';

RESET MASTER;
START SLAVE;
```

# Replicaton setup: node2 (zabbix-ha-db2)

```
SHOW SLAVE STATUS\G
```

```
***** 1. row *****
      Slave_IO_State: Waiting for master to send event
      Master_Host: <NODE1_IP>
      Master_User: replicator
      ...
      Master_Log_File: log-bin.000001
Read_Master_Log_Pos: 245
      ...
      Slave_IO_Running: Yes
      Slave_SQL_Running: Yes
      ...
      Last_IO_Errno: 0
      Last_IO_Error:
```

# Replicaton setup: node2 (zabbix-ha-db2)

```
MariaDB [(none)]> SHOW MASTER STATUS\G
***** 1. row *****
      File: log-bin.000001
      Position: 245
      Binlog_Do_DB:
      Binlog_Ignore_DB:
```

# Replicaton setup: node3 (zabbix-ha-db3)

```
## Login to MySQL:
```

```
mysql -uroot -p<MYSQL_ROOT_PASSWORD>
```

```
STOP SLAVE;
```

```
CHANGE MASTER TO MASTER_HOST = '<NODE2_IP>', MASTER_USER = 'replicator',  
MASTER_PASSWORD = '<REPLICATOR_PASSWORD>', MASTER_LOG_FILE='log-bin.000001',  
MASTER_LOG_POS = 245;
```

```
GRANT REPLICATION SLAVE ON *.* TO 'replicator'@'<NODE1_IP>' identified by  
'<REPLICATOR_PASSWORD>';
```

```
RESET MASTER;  
START SLAVE;
```

# Replicaton setup: node3 (zabbix-ha-db3)

```
SHOW SLAVE STATUS\G
```

```
***** 1. row *****
      Slave_IO_State: Waiting for master to send event
      Master_Host: <NODE2_IP>
      Master_User: replicator
      ...
      Slave_IO_Running: Yes
      Slave_SQL_Running: Yes
      ...
      Last_IO_Errno: 0
      Last_IO_Error:
```

# Replicaton setup: node3 (zabbix-ha-db3)

```
MariaDB [(none)]> SHOW MASTER STATUS\G
***** 1. row *****
      File: log-bin.000001
      Position: 245
      Binlog_Do_DB:
      Binlog_Ignore_DB:
```

# Replicaton setup: node1 (zabbix-ha-db1)

```
STOP SLAVE;  
CHANGE MASTER TO MASTER_HOST = '<NODE3_IP>', MASTER_USER = 'replicator',  
MASTER_PASSWORD = '<REPLICATOR_PASSWORD>', MASTER_LOG_FILE='log-bin.000001',  
MASTER_LOG_POS =245;  
START SLAVE;
```

```
SHOW SLAVE STATUS\G
```

```
***** 1. row *****  
Slave_IO_State: Waiting for master to send event  
Master_Host: <NODE3_IP>  
Master_User: replicator  
...  
Last_IO_Errno: 0  
Last_IO_Error:
```



# Prepare zabbix DB: node1 (zabbix-ha-db1)

```
## From this point forward all MySQL queries can be executed on any node
## All queries will be replicated to other nodes!
## We will use <NODE1>,
...
```

```
## Login to mysql and create zabbix db/user:
create database zabbix character set utf8 collate utf8_bin;
grant all privileges on zabbix.* to zabbix@'%' identified by '<DB_ZABBIX_PASS>';
quit

## upload db scema and basic conf:
## create.sql.gz copied from main zabbix server
## located in /usr/share/doc/zabbix-server-mysql-*/create.sql.gz
zcat create.sql.gz | mysql -uzabbix -p<DB_ZABBIX_PASS> zabbix
```

# Prepare zabbix DB: node1 (zabbix-ha-db1)

```
## And this is the moment you would setup Partitioning  
## But that's the other story... :)  
## SO, We proceed to server setup.
```

# Mysql replication Debug commands

```
SHOW BINARY LOGS;
```

```
SHOW SLAVE STATUS;
```

```
SHOW MASTER STATUS\g
```

```
RESET MASTER; ## removes all binary log files that are listed in the index file, leaving  
## only a single, empty binary log file with a numeric suffix of .000001
```

```
RESET MASTER TO 1234; ## reset to specific binary log position
```

```
PURGE BINARY LOGS BEFORE '2019-10-11 00:20:00';  
## Numbering is not reset, may be safely used while replication  
## slaves are running.
```

```
FLUSH BINARY LOGS; ## Will reset state of binary logs and restarts numbering
```

# SERVER CLUSTER

ZABBIX  
HA CLUSTER SETUPS

# server cluster

```
## Install HA components:  
yum groupinstall ha -y
```

```
## Create user for cluster:  
echo zabbix123 | passwd --stdin hacluster
```

```
## install zabbix server:  
yum install -y zabbix-server
```

```
## DON'T START OR ENABLE - that will be managed by HA components
```

# server cluster

```
## Copy default zabbix_server.conf file:
cp zabbix_server.conf /etc/zabbix/zabbix_server.conf

## and modify accordingly
vi zabbix_server.conf
...
SourceIP=192.168.7.87 #VIP for zabbix-server cluster
...
DBHost=192.168.7.89 # VIP of the DB's
DBName=zabbix
DBUser=zabbix
DBPassword=<DB_ZABBIX_PASS>
...

## Deploy to all server nodes
```

# server cluster

```
## Authenticate cluster nodes:  
pcs cluster auth zabbix-ha-db1 zabbix-ha-db2 zabbix-ha-db3  
username: hacluster  
password: <CLUSTER_PASSWORD>  
  
## Create zabbix_server_cluster:  
pcs cluster setup --name zabbix_server_cluster \  
zabbix-ha-srv1 zabbix-ha-srv2 zabbix-ha-srv3 --force  
  
## Disable fencing as for now we will not use it:  
pcs property set stonith-enabled=false  
  
## Restart:  
pcs cluster stop --all && pcs cluster start --all
```

# server cluster: resources

```
## Prevent Resources from Moving after Recovery  
pcs resource defaults resource-stickiness=100
```

```
## VIP for zabbix server app:
```

```
pcs resource create virtual_ip_server ocf:heartbeat:IPaddr2 ip=192.168.7.87 op  
monitor interval=5s --group zabbix_server_cluster
```

```
## control zabbix-server daemon:
```

```
pcs resource create ZabbixServer systemd:zabbix-server op monitor interval=10s --  
group zabbix_server_cluster
```



# server cluster: resources

```
## Add colocation: resources must run on same node:
```

```
pcs constraint colocation add virtual_ip_server ZabbixServer INFINITY -force
```

```
## in specific order:
```

```
pcs constraint order virtual_ip_server then ZabbixServer
```

```
## Set start/stop timeout operations
```

```
pcs resource op add ZabbixServer start interval=0s timeout=60s
```

```
pcs resource op add ZabbixServer stop interval=0s timeout=120s
```

# server cluster: check

```
[root@zabbix-ha-srv1 ~]# pcs status
Cluster name: zabbix_server_cluster
Stack: corosync
Current DC: zabbix-ha-srv2 (version 1.1.20-5.el7_7.1-3c4c782f70) - partition
with quorum
...
3 nodes configured
2 resources configured

Online: [ zabbix-ha-srv1 zabbix-ha-srv2 zabbix-ha-srv3 ]

Full list of resources:
Resource Group: zabbix_server_cluster
  virtual_ip_server (ocf::heartbeat:IPaddr2):        Started zabbix-ha-srv1
  ZabbixServer      (systemd:zabbix-server):        Started zabbix-ha-srv1
```

# FRONTEND CLUSTER

ZABBIX  
HA CLUSTER SETUPS

# Frontend cluster

```
## Install HA components:  
yum groupinstall ha -y
```

```
## Create user for cluster:  
echo zabbix123 | passwd --stdin hacluster
```

```
## install zabbix frontend:  
yum install -y zabbix-web-mysql
```

```
## DON'T START OR ENABLE - that will be managed by HA components
```

# Frontend cluster

```
## Prepare zabbix-FE config:
cat /etc/zabbix/web/zabbix.conf.php
$DB['TYPE']      = 'MYSQL';
$DB['SERVER']    = '192.168.7.89';
$DB['PORT']      = '0';
$DB['DATABASE'] = 'zabbix';
$DB['USER']      = 'zabbix';
$DB['PASSWORD'] = 'zabbix123';
...
$ZBX_SERVER      = '192.168.7.87';
$ZBX_SERVER_PORT = '10051';
$ZBX_SERVER_NAME = 'ZABBIX-HA';

## Deploy to all FE nodes on same location: /etc/zabbix/web/
```

# Frontend cluster

```
## create resource for apache Enable the server-status page.
```

```
vi /etc/httpd/conf.d/serverstatus.conf
```

```
Listen 127.0.0.1:8080
<VirtualHost localhost:8080>
<Location /server-status>
RewriteEngine Off
SetHandler server-status
Allow from 127.0.0.1
Order deny,allow
Deny from all
</Location>
</VirtualHost>
```

# Frontend cluster

```
## set apache to listen only on VIP

vi /etc/httpd/conf/httpd.conf +/Listen 80

## change to:
...
Listen 192.168.7.88:80
...

## Or...
```

# Frontend cluster

```
## Authenticate cluster nodes:  
pcs cluster auth zabbix-he-fe1 zabbix-ha-fe2 zabbix-ha-fe3  
username: hacluster  
password: <CLUSTER_PASSWORD>  
  
## Create zabbix_frontend_cluster:  
pcs cluster setup --name zabbix_fe_cluster \  
zabbix-ha-fe1 zabbix-ha-fe2 zabbix-ha-fe3 --force  
  
## Restart:  
pcs cluster stop --all && pcs cluster start -all  
  
## Disable fencing as for now we will not use it:  
pcs property set stonith-enabled=false
```



# Frontend cluster: resources

```
## VIP for FE
```

```
pcs resource create virtual_ip_fe ocf:heartbeat:IPaddr2 ip=192.168.7.88 op  
monitor interval=5s --group zabbix_fe_cluster
```

```
## for Apache:
```

```
pcs resource create zabbix_fe ocf:heartbeat:apache \  
configfile=/etc/httpd/conf/httpd.conf \  
statusurl="http://localhost:8080/server-status" op \  
monitor interval=30s --group zabbix_fe_cluster
```

# Frontend cluster: resources

```
## Add colocation: resources must run on same node:
pcs constraint colocation add virtual_ip_fe zabbix_fe INFINITY

## in specific order:
pcs constraint order virtual_ip_fe then zabbix_fe

pcs resource defaults resource-stickiness=100

## Set start/stop timeout operations
pcs resource op add zabbix_fe start interval=0s timeout=60s
pcs resource op add zabbix_fe stop interval=0s timeout=120s
```

# where to get more info:

google...

```
https://access.redhat.com/documentation/en-us/  
## look for: Red Hat Enterprise Linux >> 7 >>  
High Availability Add-On Reference
```

```
https://clusterlabs.org/
```

Contact Zabbix sales :)

# THANK YOU!



Edmunds Vesmanis  
**ZABBIX** Senior Trainer

# ZABBIX '19 SUMMIT