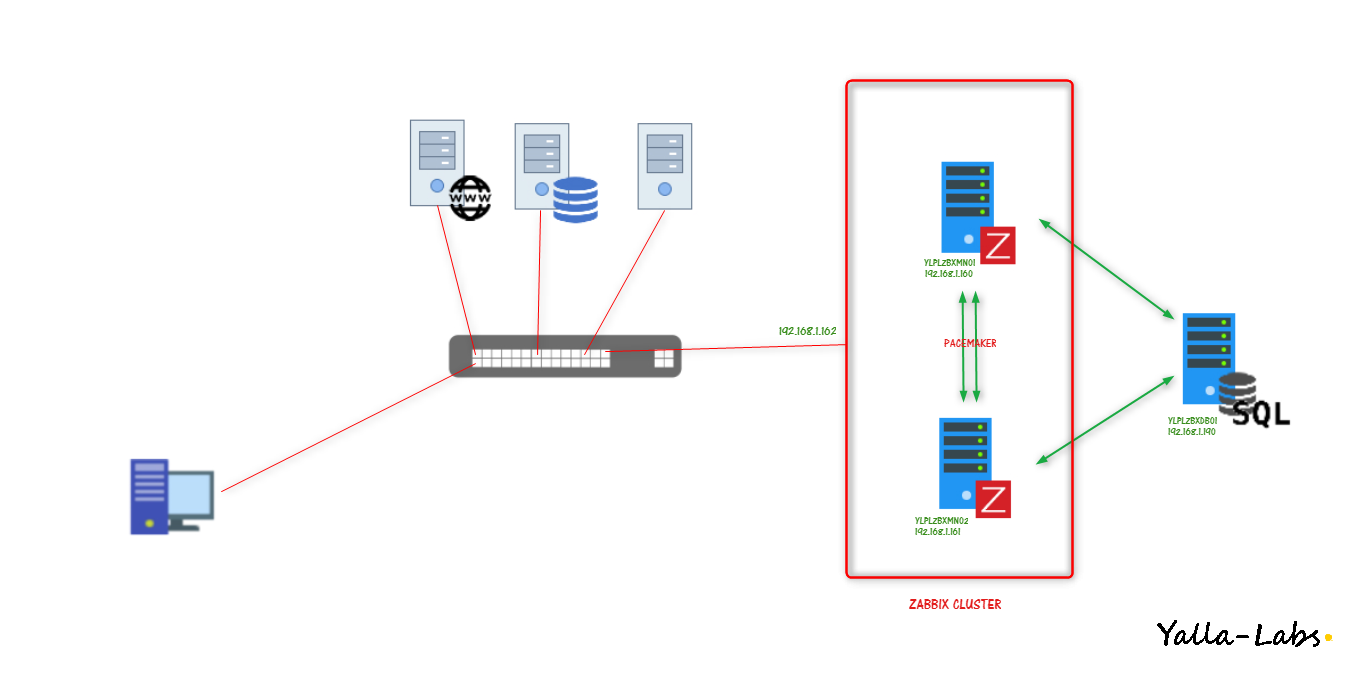
**About Zabbix**

**Zabbix** is a free and open source **network monitoring Software tool**which is used to monitor and track the availability and performance of your IT infrastracture: servers, network devices and other IT assets.

A **high availability** architecture is one of the key requirements for any Enterprise Deployment network. In this tutorial we will cover how to set up **high availability Zabbix server**. We are going to create a High Availability Active/Passive Cluster that will consist of two CentOS 7 / RHEL 7 servers with a Floating IP and using a shared database. In order to achieve the high availability we will use the **Corosync** cluster engine, and the **Pacemaker** resource manager.

**Environment:**

[](http://yallalabs.com/wp-content/uploads/2018/04/ZABBIX-HA.png)

– Before we start make sure to disable Selinux on the both nodes in the cluster

# sed -i 's/^SELINUX=.\*/SELINUX=disabled/g' /etc/sysconfig/selinux && reboot

– On both Zabbix Server nodes **ZabbixApp1** e **ZabbixApp2**, we need to install the Zabbix repository using the below commands:

[root@ZabbixApp1 ~]# rpm --import http://repo.zabbix.com/RPM-GPG-KEY-ZABBIX

[root@ZabbixApp1 ~]# rpm -ivh http://repo.zabbix.com/zabbix/3.4/rhel/7/x86\_64/zabbix-release-3.4-2.el7.noarch.rpm

– Now use the below command to install Zabbix and necessary packages:

[root@ZabbixApp1 ~]# yum update

[root@ZabbixApp1 ~]# yum install zabbix-server-mysql zabbix-web-mysql zabbix-agent zabbix-get -y

– On the Database Server **ylplzbxdb01** Install the **MariaDB Server** package using the following Command:

[root@ylplzbxdb01 ~]# yum install mariadb-server -y

[root@ylplzbxdb01 ~]# systemctl enable mariadb && systemctl start mariadb

– Don’t forget to set a password for the root using mysql\_secure\_installtion, take a look to this tutorial: [Securing MySQL server / Mariadb with mysql\_secure\_installation](http://yallalabs.com/linux/securing-mysql-server-mariadb-with-mysql_secure_installation/)

– First we need to create zabbix database **(zabbixdb)** and create a zabbix user **(zabbixuser)**.

[root@ylplzbxdb01 ~]# mysql -u root -p

Enter password:

Welcome to the MariaDB monitor. Commands end with ; or \g.

Your MariaDB connection id is 10

Server version: 5.5.47-MariaDB MariaDB Server

Copyright (c) 2000, 2015, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> CREATE DATABASE zabbixdb CHARACTER SET utf8 COLLATE utf8\_bin;

Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON zabbixdb.\* TO zabbixuser@'%' IDENTIFIED BY "Password";

Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> FLUSH PRIVILEGES;

Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> exit

Bye

[root@ylplzbxdb01 ~]#

– On one of the nodes, use the **scp** tool to copy the **create.sql.gz** to your Database Server **ylplzbxdb01**:

[root@ZabbixApp1 ~]# scp /usr/share/doc/zabbix-server-mysql-3.4.8/create.sql.gz root@192.168.1.190:/root/

– After creating the zabbix database and user we need to import the zabbix initial database using the below commands:

[root@ylplzbxdb01 ~]# zcat create.sql.gz | mysql -u zabbixuser -p zabbixdb

Enter password:

[root@ylplzbxdb01 ~]#

– On each node in the cluster, we need to edit database configuration in the zabbix server configuration file **zabbix\_server.conf**by specifing the Database Host IP, Zabbix database name for Zabbix , Zabbix database user name and the password. We need also to specify the SourceIP that should be the Virtual IP Address of our cluster.

# vi /etc/zabbix/zabbix\_server.conf

[...]

SourceIP=Virtual\_IP\_Address\_Cluster

DBHost=IP\_Address\_DataBase\_Server

DBName=zabbixdb

DBUser=zabbixuser

DBPassword=Password

– On each node in the cluster, we have to configure **PHP timezone**, open the file **/etc/httpd/conf.d/zabbix.conf** and uncomment the “date.timezone” line and change it to your timezone.

# vi /etc/httpd/conf.d/zabbix.conf

[...]

php\_value date.timezone Europe/Rome

– On each node in the cluster, we need to make sure cluster nodes can communicate with eachother by their name. If you have a DNS server, add additional entries for the two machines. Otherwise, you’ll need to add the nodes to **/etc/hosts** file. Below are the entries for our cluster nodes:

# cat >> /etc/hosts << END

> 192.168.1.160 ZabbixApp1 ZabbixApp1.yallalabs.local

> 192.168.1.161 ZabbixApp2 ZabbixApp2.yallalabs.local

> END

– If you are running the firewalld daemon, On each node in the cluster execute the following commands to enable the necessary ports

# firewall-cmd --permanent --add-service=high-availability

success

# firewall-cmd --permanent --add-service=http

success

# firewall-cmd --permanent --add-port=10051/tcp

success

# firewall-cmd --permanent --add-port=10050/tcp

success

# firewall-cmd --reload

success

– On each node in the cluster, install the High Availability packages as below

[root@ZabbixApp1 ~]# yum install pacemaker pcs -y

– In order to use pcs to configure the cluster and communicate among the nodes, you must set a password on each node for the user ID **hacluster**, which is the pcs administration account.

# passwd hacluster

Changing password for user hacluster.

New password:

Retype new password:

passwd: all authentication tokens updated successfully.

– On each node in the cluster, execute the following commands to start and enable the **pcsd** service at system boot.

# systemctl start pcsd

# systemctl enable pcsd

Created symlink from /etc/systemd/system/multi-user.target.wants/pcsd.service to /usr/lib/systemd/system/pcsd.service.

– on the first node **ZabbixApp1**, execute the following command to authenticate the pcs user **hacluster** for each node in the cluster

[root@ZabbixApp1 ~]# pcs cluster auth ZabbixApp1 ZabbixApp2

Username: hacluster

Password:

ZabbixApp1: Authorized

ZabbixApp2: Authorized

– On the first node **ZabbixApp1**, execute the following command to create the cluster named **zabbixserver** that consists of the nodes **ZabbixApp1** and **ZabbixApp2**:

[root@ZabbixApp1 ~]# pcs cluster setup --name zabbixserver ZabbixApp1 ZabbixApp2

Destroying cluster on nodes: ZabbixApp1, ZabbixApp2...

ZabbixApp1: Stopping Cluster (pacemaker)...

ZabbixApp2: Stopping Cluster (pacemaker)...

ZabbixApp1: Successfully destroyed cluster

ZabbixApp2: Successfully destroyed cluster

Sending 'pacemaker\_remote authkey' to 'ZabbixApp1', 'ZabbixApp2'

ZabbixApp2: successful distribution of the file 'pacemaker\_remote authkey'

ZabbixApp1: successful distribution of the file 'pacemaker\_remote authkey'

Sending cluster config files to the nodes...

ZabbixApp1: Succeeded

ZabbixApp2: Succeeded

Synchronizing pcsd certificates on nodes ZabbixApp1, ZabbixApp2...

ZabbixApp1: Success

ZabbixApp2: Success

Restarting pcsd on the nodes in order to reload the certificates...

ZabbixApp1: Success

ZabbixApp2: Success

– To start the cluster services on both nodes of the cluster, it’s engouh to execute the below command on the first node **ZabbixApp1**:

[root@ZabbixApp1 ~]# pcs cluster start --all

ZabbixApp1: Starting Cluster...

ZabbixApp2: Starting Cluster...

– On each node in the cluster, run the following commands to enable the corosync and the Pacemaker daemon to start at boot using:

# systemctl enable corosync

# systemctl enable pacemaker

– To display the current status of the cluster, execute the following command:

[root@ZabbixApp1 ~]# pcs status cluster

Cluster Status:

Stack: corosync

Current DC: ZabbixApp2 (version 1.1.16-12.el7\_4.8-94ff4df) - partition with quorum

Last updated: Tue Apr 17 21:20:40 2018

Last change: Tue Apr 17 21:17:44 2018 by hacluster via crmd on ZabbixApp2

2 nodes configured

0 resources configured

PCSD Status:

ZabbixApp1: Online

ZabbixApp2: Online

– To display just the current status of the cluster nodes, run the following command:

[root@ZabbixApp1 ~]# pcs status nodes

Pacemaker Nodes:

Online: ZabbixApp1 ZabbixApp2

Standby:

Maintenance:

Offline:

Pacemaker Remote Nodes:

Online:

Standby:

Maintenance:

Offline:

– We are not going to use the Stonith, let’s disable it, on the first node in the cluster, run the following command:

[root@ZabbixApp1 ~]# pcs property set stonith-enabled=false

– Since we have only two nodes, it’s not necessary to enable the Quorum, on the first node in the cluster, use the following command to disable it :

[root@ZabbixApp1 ~]# pcs property set no-quorum-policy=ignore

– Our first resource will be an **IPaddr2** resource, which is a floating IP address to access the zabbix Frontend, the resource will be named **cluster\_virtual\_ip** and we will assign the floating IP address **192.168.1.162** with a netmask **24**

[root@ZabbixApp1 ~]# pcs resource create cluster\_virtual\_ip ocf:heartbeat:IPaddr2 ip=192.168.1.162 cidr\_netmask=24 op monitor interval=20s

– Let’s now create a new systemd resource called **zabbix\_server** for the zabbix Server deamon and be monitored operations every 10 seconds:

[root@ZabbixApp1 ~]# pcs resource create zabbix\_server systemd:zabbix-server op monitor interval=10s

– We need to create an Apache resource called **httpd**, it’s going to be monitored evey secondes:

[root@ZabbixApp1 ~]# pcs resource create httpd systemd:httpd op monitor interval=10s

– To facilate the management the cluster, we are going to create a group resource called **groupp\_zabbix\_httpd**  where be composed by **zabbix\_server** and **httpd** resource

[root@ZabbixApp1 ~]# pcs resource group add groupp\_zabbix\_httpd zabbix\_server httpd

– To ensure resources run on the same node, to achieve that we are going to use **colocation constraint**, execute the following command to ensure resource group **groupp\_zabbix\_httpd**  and the float IP Address resource **cluster\_virtual\_ip** are started on the same node, The **INFINITY** score also means that if cluster\_virtual\_ip is not active anywhere, groupp\_zabbix\_httpd will not be permitted to run.

[root@ZabbixApp1 ~]# pcs constraint colocation add groupp\_zabbix\_httpd cluster\_virtual\_ip INFINITY

– To make sure that cluster resources start and stop in order, we will use **order constraint**, we need to make sure the float IP Address resource **cluster\_virtual\_ip** not only runs on the same node, but starts before the resource group **groupp\_zabbix\_httpd**  use the following command:

[root@ZabbixApp1 ~]# pcs constraint order cluster\_virtual\_ip then groupp\_zabbix\_httpd

Adding cluster\_virtual\_ip groupp\_zabbix\_httpd (kind: Mandatory) (Options: first-action=start then-action=start)

– After creating our resources, let’s check the status of the cluster.

[root@ZabbixApp1 ~]# pcs status

Cluster name: zabbixserver

Stack: corosync

Current DC: ZabbixApp2 (version 1.1.16-12.el7\_4.8-94ff4df) - partition with quorum

Last updated: Tue Apr 17 21:30:34 2018

Last change: Tue Apr 17 21:28:15 2018 by root via cibadmin on ZabbixApp1

2 nodes configured

3 resources configured

Online: [ ZabbixApp1 ZabbixApp2 ]

Full list of resources:

cluster\_virtual\_ip (ocf::heartbeat:IPaddr2): Started ZabbixApp1

Resource Group: groupp\_zabbix\_httpd

zabbix\_server (systemd:zabbix-server): Started ZabbixApp1

httpd (systemd:httpd): Started ZabbixApp1

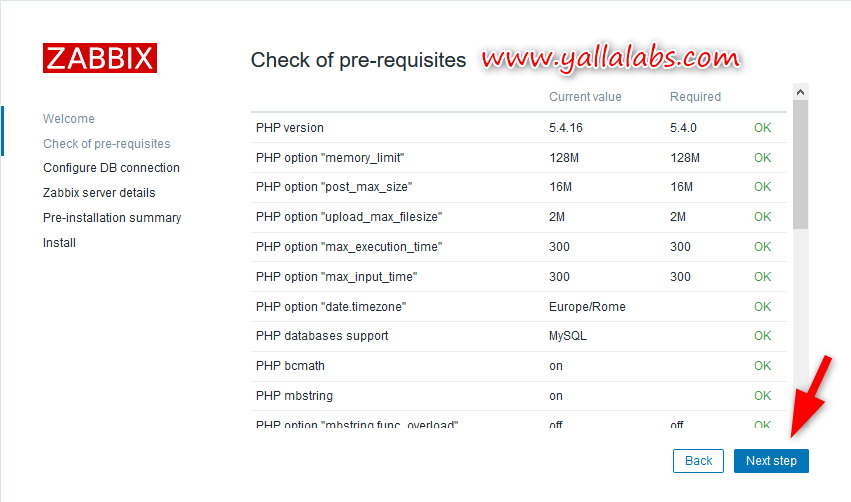
Daemon Status:

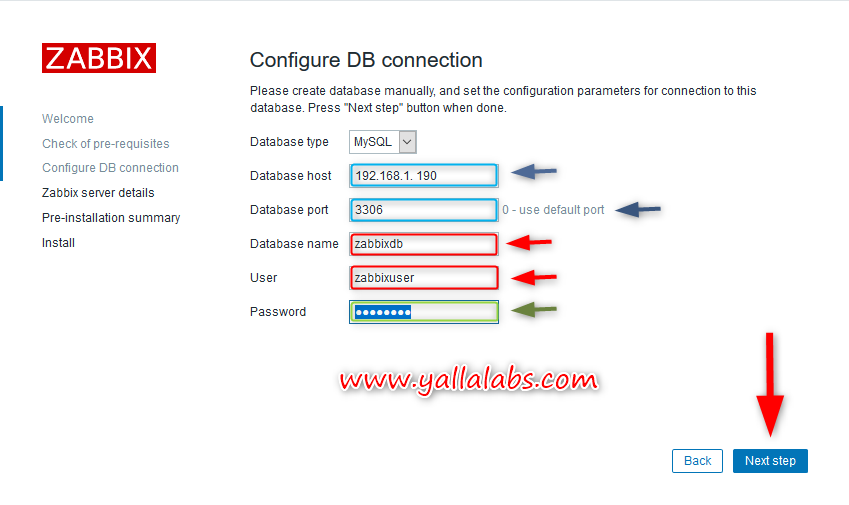
corosync: active/enabled

pacemaker: active/enabled

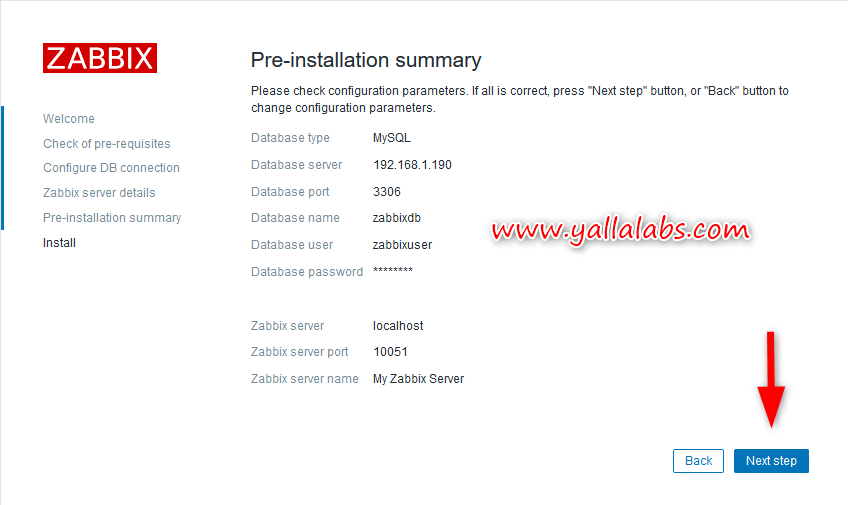
pcsd: active/enabled

– Note that all the resources are running on the same node **ZabbixApp1**, now we need to finish the installation of the zabbix Frontend, open the browser and point to the **Float IP address** of our cluster http://Float\_IP\_ADDRESS/zabbix/  
   
[](http://yallalabs.com/wp-content/uploads/2018/04/01-configuring-high-availability-ha-zabbix-server-on-centos-7.png)

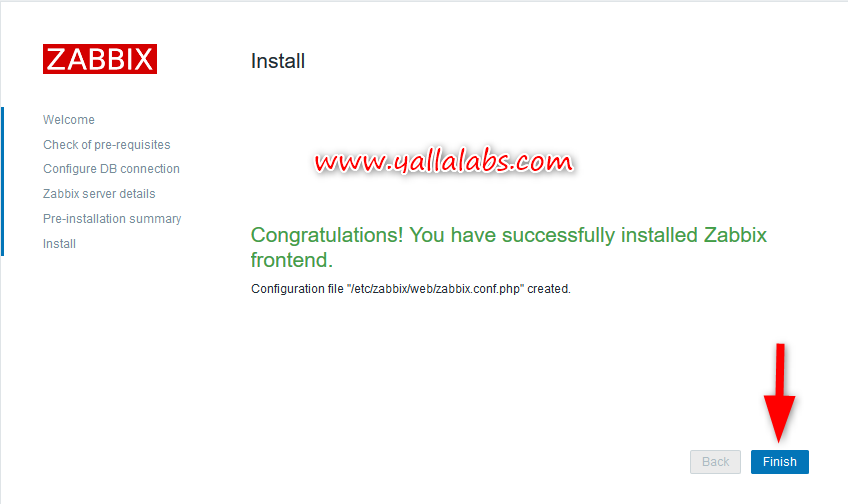
– Make sure that all software prerequisites are met and click **Next Step**  
   
[](http://yallalabs.com/wp-content/uploads/2018/04/02-configuring-high-availability-ha-zabbix-server-on-centos-7.png)

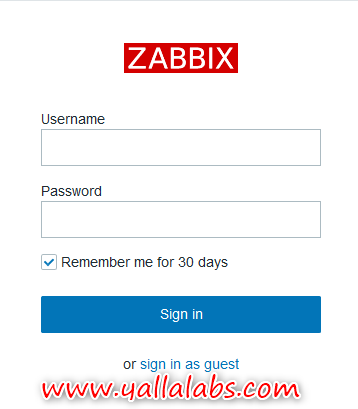
– Enter the details to connect to the zabbix database and click **Next Step**  
   
[](http://yallalabs.com/wp-content/uploads/2018/04/03-configuring-high-availability-ha-zabbix-server-on-centos-7.png)

– Enter Zabbix server details and click **Next Step** button  
   
[](http://yallalabs.com/wp-content/uploads/2018/04/04-configuring-high-availability-ha-zabbix-server-on-centos-7.png)

– Review the pre-installation summary and click **Next Step** button.  
   
[](http://yallalabs.com/wp-content/uploads/2018/04/05-configuring-high-availability-ha-zabbix-server-on-centos-7.png)

– Finish the installation by clicking the button **Finish**

[](http://yallalabs.com/wp-content/uploads/2018/04/06-configuring-high-availability-ha-zabbix-server-on-centos-7.png)

– Now you’ll be redirected to the zabbix Frontend Signin page. The default user name is **Admin** and the password is **zabbix**.  
   
[](http://yallalabs.com/wp-content/uploads/2018/04/07-configuring-high-availability-ha-zabbix-server-on-centos-7.png)

– When you are done with the installation, we have to copy the generated php file **zabbix.conf.php** to the passive node, in this case the passive node is **ZabbixApp2**

[root@ZabbixApp1 ~]# scp /etc/zabbix/web/zabbix.conf.php root@192.168.1.161:/etc/zabbix/web/zabbix.conf.php

– Finally, let’s test the fail over to the passive node **ZabbixApp2**, all the resource are active in the first node **ZabbixApp1** we should put it in standby like this all the resource well move the second node

[root@ZabbixApp1 ~]# pcs cluster standby ZabbixApp1

– After putting the first node in standby, check the status of the cluster, all the resource should run in the second node

[root@ZabbixApp1 ~]# pcs status

Cluster name: zabbixserver

Stack: corosync

Current DC: ZabbixApp2 (version 1.1.16-12.el7\_4.8-94ff4df) - partition with quorum

Last updated: Sun Apr 22 22:43:07 2018

Last change: Sun Apr 22 22:42:16 2018 by root via cibadmin on ZabbixApp1

2 nodes configured

3 resources configured

Node ZabbixApp1: standby

Online: [ ZabbixApp2 ]

Full list of resources:

cluster\_virtual\_ip (ocf::heartbeat:IPaddr2): Started ZabbixApp2

Resource Group: groupp\_zabbix\_httpd

zabbix\_server (systemd:zabbix-server): Started ZabbixApp2

httpd (systemd:httpd): Started ZabbixApp2

Daemon Status:

corosync: active/enabled

pacemaker: active/enabled

pcsd: active/enabled