

# Dell EMC VxFlex OS High Availability Technical Notes

Version 3.x

P/N 302-005-647

Rev 01

May 2019

Copyright © 2019 Dell Inc. or its subsidiaries. All rights reserved.

Dell believes the information in this publication is accurate as of its publication date. The information is subject to change without notice.

THE INFORMATION IN THIS PUBLICATION IS PROVIDED "AS-IS." DELL MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WITH RESPECT TO THE INFORMATION IN THIS PUBLICATION, AND SPECIFICALLY DISCLAIMS IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. USE, COPYING, AND DISTRIBUTION OF ANY DELL SOFTWARE DESCRIBED IN THIS PUBLICATION REQUIRES AN APPLICABLE SOFTWARE LICENSE.

Dell Technologies, Dell, EMC, Dell EMC and other trademarks are trademarks of Dell Inc. or its subsidiaries. Other trademarks may be the property of their respective owners. Published in the USA.

Dell EMC  
Hopkinton, Massachusetts 01748-9103  
1-508-435-1000 In North America 1-866-464-7381  
[www.DellEMC.com](http://www.DellEMC.com)

# CONTENTS

	<b>Preface</b>	<b>5</b>
<b>Chapter 1</b>	<b>Introduction</b>	<b>7</b>
	Overview of the high-availability solution.....	8
<b>Chapter 2</b>	<b>Example using Apache httpd and Keepalived on Ubuntu</b>	<b>11</b>
	Configuration workflow - Ubuntu environment.....	12
	Update server.xml.....	12
	Create certificates.....	13
	Configure the virtual IP address.....	15
	Enable mod_ssl.....	16
	Configure Keepalived.....	16
	Finish the configuration.....	17
<b>Chapter 3</b>	<b>Example using HAProxy and Keepalived on CentOS 7.3</b>	<b>19</b>
	Configure the firewall.....	20
	Prepare the VxFlex OS Gateway.....	21
	Configure HAProxy systems.....	22
	Create self-signed certificates for HAProxy.....	27



# Preface

As part of an effort to improve its product lines, Dell EMC periodically releases revisions of its software and hardware. Therefore, some functions described in this document might not be supported by all versions of the software or hardware currently in use. The product release notes provide the most up-to-date information on product features.

Contact your Dell EMC technical support professional if a product does not function properly or does not function as described in this document.

**Note:** This document was accurate at publication time. Go to Dell EMC Online Support (<https://support.emc.com>) to ensure that you are using the latest version of this document.

Previous versions of Dell EMC VxFlex OS were marketed under the name Dell EMC ScaleIO.

Similarly, previous versions of Dell EMC VxFlex Ready Node were marketed under the name Dell EMC ScaleIO Ready Node.

References to the old names in the product, documentation, or software, etc. will change over time.

**Note:** Software and technical aspects apply equally, regardless of the branding of the product.

## Related documentation

The release notes for your version includes the latest information for your product.

The following Dell EMC publication sets provide information about your VxFlex OS or VxFlex Ready Node product:

- VxFlex OS software (downloadable as VxFlex OS Software <version> Documentation set)
- VxFlex Ready Node with AMS (downloadable as VxFlex Ready Node with AMS Documentation set)
- VxFlex Ready Node no AMS (downloadable as VxFlex Ready Node no AMS Documentation set)
- VxRack Node 100 Series (downloadable as VxRack Node 100 Series Documentation set)

You can download the release notes, the document sets, and other related documentation from Dell EMC Online Support.

## Typographical conventions

Dell EMC uses the following type style conventions in this document:

<b>Bold</b>	Used for names of interface elements, such as names of windows, dialog boxes, buttons, fields, tab names, key names, and menu paths (what the user specifically selects or clicks)
<i>Italic</i>	Used for full titles of publications referenced in text
Monospace	Used for: <ul style="list-style-type: none"><li>• System code</li></ul>

- System output, such as an error message or script
- Pathnames, filenames, prompts, and syntax
- Commands and options

<i>Monospace italic</i>	Used for variables
<b>Monospace bold</b>	Used for user input
[ ]	Square brackets enclose optional values
	Vertical bar indicates alternate selections - the bar means “or”
{ }	Braces enclose content that the user must specify, such as x or y or z
...	Ellipses indicate nonessential information omitted from the example

### Where to get help

Dell EMC support, product, and licensing information can be obtained as follows:

#### Product information

For documentation, release notes, software updates, or information about Dell EMC products, go to Dell EMC Online Support at <https://support.emc.com>.

#### Technical support

Go to Dell EMC Online Support and click Service Center. You will see several options for contacting Dell EMC Technical Support. Note that to open a service request, you must have a valid support agreement. Contact your Dell EMC sales representative for details about obtaining a valid support agreement or with questions about your account.

### Your comments

Your suggestions will help us continue to improve the accuracy, organization, and overall quality of the user publications. Send your opinions of this document to [techpubcomments@emc.com](mailto:techpubcomments@emc.com).


# CHAPTER 1

## Introduction

VxFlex OS is a software-only solution that uses existing servers' local disks and LAN to create a virtual SAN that has all the benefits of external storage—but at a fraction of the cost and complexity.

VxFlex OS utilizes the existing local internal storage and turns it into internal shared block storage. For many workloads, VxFlex OS storage is comparable to, or better than, external shared block storage.

The VxFlex OS Gateway, which runs on Apache Tomcat, hosts a number of VxFlex OS features, including a REST gateway, VxFlex OS Installer, and SNMP trap sender. For high availability when two Apache Tomcat instances (VxFlex OS Gateway servers) are configured, use Apache httpd.

 **Note:** This document discusses procedures needed only in environments in which the REST API commands are used.

- [Overview of the high-availability solution](#).....8

## Overview of the high-availability solution

To prevent a single point of failure in the Apache httpd, you can use a failover httpd instance that is clustered using one of the many available clustering stacks.

**Note:** When the high-availability solution is used for VxFlex OS Gateway, ESRS is not supported.

These operating system solutions are described in this document:

- Ubuntu, using Keepalived, a lightweight open source clustering stack for Linux.
- CentOS, using Keepalived and HAProxy

A shared (virtual) IP address is used for both Apache httpd servers. Both of the Apache httpd servers have the same configuration, except that one is defined in the Keepalived configuration as the master, while the other is defined as the slave. If the master fails, the slave becomes the master, which means the users do not notice any loss of service.

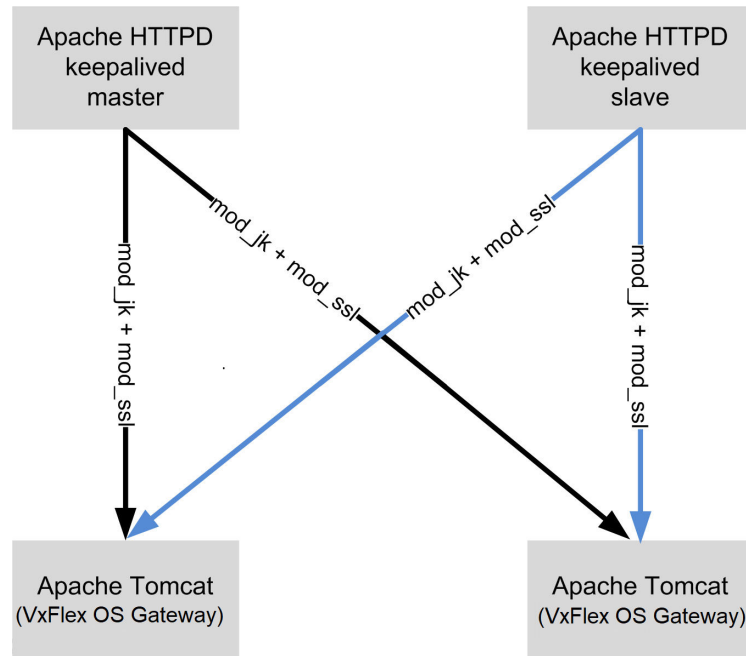
The VxFlex OS Gateway servers are configured the same way. According to the configuration in the Apache httpd, one of the VxFlex OS Gateway servers (machine2) is the master and the other (machine1) is the slave.

In the Ubuntu environment, the mod\_jk module in the Apache httpd monitors the VxFlex OS Gateway servers and decides to which server to forward the request from the client.

In the CentOS environment, the Keepalived service monitors the VxFlex OS Gateway servers and determines to which server to forward the client request.

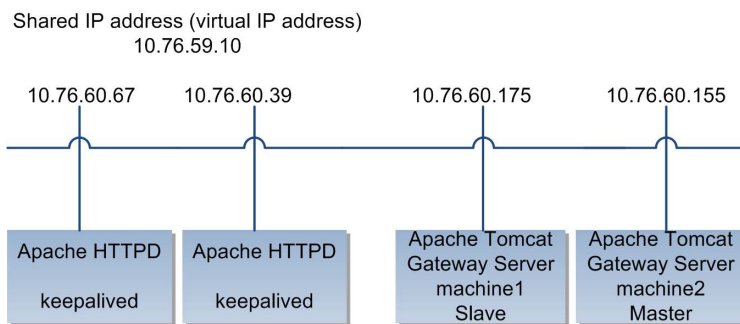
The following figures illustrate the configuration and topology for the Ubuntu solution:

**Figure 1** High availability Gateway cluster topology - Ubuntu





**Figure 2** High availability networking example





# CHAPTER 2

## Example using Apache httpd and Keepalived on Ubuntu

The following is a full configuration example for a system running on Ubuntu 14.04 with Apache httpd 2.4.7, with the Keepalived open source tool used for the keep-alive component.

For more information about Keepalived, see <http://www.keepalived.org>. For more information about Ubuntu and the Apache HTTP Server, see their respective product documentation.

 **Note:** The following procedures involve VxFlex OS Gateway service restarts.

- [Configuration workflow - Ubuntu environment](#)..... 12
- [Update server.xml](#)..... 12
- [Create certificates](#)..... 13
- [Configure the virtual IP address](#)..... 15
- [Enable mod\\_ssl](#)..... 16
- [Configure Keepalived](#)..... 16
- [Finish the configuration](#)..... 17

## Configuration workflow - Ubuntu environment

Overview of the configuration workflow to enable high-availability on VxFlex OS Gateway servers in the Ubuntu environment.

1. Install Apache httpd.
2. Install mod\_jk.
3. Install and enable mod\_ssl.
4. Configure the Apache httpd virtual host to use the virtual IP address that is defined for both Apache httpd servers, instead of the default IP addresses.
5. Update the `server.xml` file on the Tomcat servers (VxFlex OS Gateway servers) to listen on port 8009 + `jvmRoute` according to the configuration of the `workers.properties` file used by mod\_jk.
6. Install and configure the clustering solution.

## Update server.xml

Update the `server.xml` file on the Tomcat servers (VxFlex OS Gateway) to listen on port 8009 + `jvmRoute` according to the configuration of the `workers.properties` file used by the mod\_jk module:

### Procedure

1. Edit the `server.xml` file, located at `<gateway installation directory>/conf/server.xml`:
  - a. Uncomment and edit the following line:

```
<!-- <Connector port="8009" protocol="AJP/1.3"
redirectPort="443" /> -->
```

After uncommenting and editing, it should look like this:

```
<Connector port="8009" protocol="AJP/1.3" redirectPort="$
{ssl.port}" />
```

- b. Add `jvmRoute="machine1"` to one gateway and `jvmRoute="machine2"` to the other.

Make sure that the name you used in the `workers.properties` file is the same as the `jvmRoute` value:

SIO-GW #1:

```
<Engine name="Catalina" defaultHost="localhost"
jvmRoute="machine1">
```

SIO-GW #2:

```
<Engine name="Catalina" defaultHost="localhost"
jvmRoute="machine2">
```

**Note:** The `worker.properties` file is configured in the "create certificates" step.

- c. If you intend to run Apache httpd on the same machine as the VxFlex OS Gateway (not recommended), ensure that the following is commented out to disabled the HTTP connector:

```
<Connector port="${http.port}" protocol="HTTP/1.1"
           connectionTimeout="20000"
           redirectPort="${ssl.port}"
           compression="force"
           compressableMimeType="application/json,application/octet-
stream" />
```

**Note:** To prevent port collision with the Apache default SSL port, change the default SSL port to something other than 443 at either Apache or Tomcat.

- d. Restart the VxFlex OS-gateway service on both VxFlex OS Gateway servers

```
service scaleio-gateway restart
```

2. Install Apache httpd:

```
- sudo apt-get install apache2
```

## Create certificates

### Procedure

1. Create a working directory called `certs` at `etc/apache2/certs`.
2. Create `.pem` files:

```
"C:\Program Files\Java\jdk1.7.0_25\bin\keytool.exe" -genkey -
alias scaleio -keyalg RSA -validity 360 -keysize 2048 -dname
"CN=<Virtual ip>, OU=ASD, O=EMC, L=Hopkinton,
S=Massachusetts, C=US" -storepass <keystore password> -
keypass <keystore password> -keystore .keystore
```

- a. Replace the values for OU, O, L, S, and C with your credentials.
- b. Replace `<keystore password>` with the password you want to use. For example:

```
b2aa330a-e7c7-4e89-86c3-6f2565973cab
```

3. Run the following commands:

 **Note:** A password is required.

```
keytool -importkeystore -storepass <keystore password> -  
keypass <keystore password> -srckeystore .keystore -  
destkeystore  
intermediate.p12 -deststoretype PKCS12
```

```
openssl pkcs12 -in intermediate.p12 -nocerts -nodes -out  
filename-key.pem
```

```
openssl pkcs12 -in intermediate.p12 -clcerts -nokeys -out  
filename-cert.pem
```

4. Copy the `filename-key.pem` and `filename-cert.pem` files you created in the previous step to `/etc/apache2/certs`.
5. Install the `mod_jk` module:

```
apt-get install libapache2-mod-jk
```

6. Create a `workers.properties` file:

```
touch /etc/apache2/workers.properties
```

7. Add the following content to the `workers.properties` file:

```
worker.list=balancel  
worker.machine1.type=ajp13  
worker.machine1.host=<machine1_IP>  
worker.machine1.port=8009  
worker.machine1.lbfactor=1  
worker.machine1.activation=disabled  
worker.machine2.type=ajp13  
worker.machine2.host=<machine2_IP>  
worker.machine2.port=8009  
worker.machine2.lbfactor=1  
worker.machine2.redirect=machine1  
worker.balancel.type=lb  
worker.balancel.balance_workers=machine1,machine2
```

Where:

- `<machine1_IP>` and `<machine2_IP>` are the IP addresses of the VxFlex OS Gateway servers.
- `worker.machine2` is the master gateway,
- `worker.machine1` is the slave gateway

8. Update the `/etc/apache2/mods-available/jk.conf` file:

- a. Change `JkWorkersFile` to `JkWorkersFile /etc/apache2/workers.properties`.

**b. Add the following:**

```
# Map a worker to a namespace.
JkMount /api/* balancer1
JkOptions +ForwardSSLCertChain
```

**c. Delete the following:**

```
<Location /jk-status>
  # Inside Location we can omit the URL in JkMount
  JkMount jk-status
  Order deny,allow
  Deny from all
  Allow from 127.0.0.1
</Location>
<Location /jk-manager>
  # Inside Location we can omit the URL in JkMount
  JkMount jk-manager
  Order deny,allow
  Deny from all
  Allow from 127.0.0.1
</Location>
```

## Configure the virtual IP address

Configure the virtual IP address for the VxFlex OS Gateway servers.

**Procedure**

1. Add the virtual IP address to the `/etc/hosts` file.

 **Note:** Add only the hostname related to the virtual IP.

2. Edit `/etc/apache2/sites-available/default-ssl.conf`:

**a. Find this line in the file:**

```
Update <VirtualHost _default_:443>
```

**b. Replace `_default_` with the virtual IP address.****c. Under the `VirtualHost _default_` section, add the following lines:**

```
SSLCertificateFile /etc/apache2/certs/filename-cert.pem
SSLCertificateKeyFile /etc/apache2/certs/filename-key.pem
JkMountCopy On
JkMount /* balancer1
```

## Enable mod\_ssl

Enable mod\_ssl, that was installed previously.

### Procedure

1. Run the following commands:

```
a2enmod ssl
service apache2 restart
a2ensite default default-ssl
service apache2 reload
```

## Configure Keepalived

Install Keepalived on every server on which Apache httpd is installed.

### Procedure

1. Install Keepalived:

```
apt-get install keepalived
```

2. Clear the example in the keepalived.conf file:

```
cat /dev/null > /etc/keepalived/keepalived.conf
```

3. Update the /etc/keepalived/keepalived.conf file:

- a. Edit /etc/keepalived/keepalived.conf on the Apache HTTPS master server:

```
vrrp_script chk_apache_httpd {
    script "systemctl --no-pager status apache2" # verify
the pid is exist or not
    interval 2 # check every 2 seconds
    weight 2 # add 2 points of prio if
OK
}

vrrp_instance VI_1 {
    interface eth0 # interface to monitor
    state MASTER
    virtual_router_id 51 # Assign one ID for this
route
    priority 101 # 101 on master, 100 on
slave
    virtual_ipaddress {
        <virtual_ip> # the virtual IP
    }
    track_script {
        chk_apache_httpd
    }
}
```



**b. Edit the /etc/keepalived/keepalived.conf file on the Apache HTTPS slave server:**

```

vrrp_script chk_apache_httpd {
    script "systemctl --no-pager status apache2" # verify
    the pid is exist or not
    interval 2 # check every 2 seconds
    weight 2 # add 2 points of prio if
    OK
}

vrrp_instance VI_1 {
    interface eth0 # interface to monitor
    state MASTER
    virtual_router_id 51 # Assign one ID for this
    route
    priority 101 # 101 on master, 100 on
    slave
    virtual_ipaddress {
        <virtual_ip> # the virtual IP
    }
    track_script {
        chk_apache_httpd
    }
}

```

## Finish the configuration


### Procedure

1. Add this line to the /etc/sysctl.conf file:

```
net.ipv4.ip_nonlocal_bind = 1
```

2. Apply the changes:

```
sysctl -p
```

 **Note:** This instructs the kernel to bind the non-local IP to the Apache httpd service.

3. Create a symlink to the /etc directory:

```
/usr/sbin/keepalived -f /etc/keepalived/keepalived.conf
```

4. Stop, then start, Keepalived:

```
service keepalived stop
service keepalived start
```

Example using Apache httpd and Keepalived on Ubuntu

# CHAPTER 3

## Example using HAProxy and Keepalived on CentOS 7.3

Configure HAProxy and Keepalived on CentOS 7.3 to provide VxFlex OS Gateway high availability.

This section describes how to configure a system running on CentOS 7.3, with the Keepalived open source tool used for the keep-alive component.

For more information about Keepalived, see <http://www.keepalived.org>. For more information about CentOS, see the product documentation.

- [Configure the firewall](#).....20
- [Prepare the VxFlex OS Gateway](#).....21
- [Configure HAProxy systems](#)..... 22
- [Create self-signed certificates for HAProxy](#).....27

## Configure the firewall

If the firewall is active, you must allow ports. If `firewalld` and `selinux` are disabled, skip this section.

### Before you begin

Consult the firewall vendor documentation on how to open firewall ports.

### About this task

These ports must be opened on the firewall: 80, 8080, 443, 28080, 28443, and `vrp` traffic.

### Procedure

1. Enable `vrp` traffic in the firewall:

```
firewall-cmd --permanent --zone=public --add-service=http
firewall-cmd --permanent --zone=public --add-service=https
firewall-cmd --permanent --zone=public --add-port=28443/tcp
firewall-cmd --permanent --zone=public --add-port=28080/tcp
firewall-cmd --permanent --zone=public --add-protocol=vrp
firewall-cmd --reload
```

2. Validate the configuration:

```
firewall-cmd --list-all
```

Output, similar to the following is displayed:

```
[root@A59T6290 ~]# firewall-cmd --list-all
public (active)
  target: default
  icmp-block-inversion: no
  interfaces: bond0 bond0.31 enp2s0f0 enp5s0f0 enp5s0f1
  sources:
  services: dhcpv6-client http https ssh
  ports: 33833/tcp 443/tcp 80/tcp 9011/tcp 6611/tcp 9099/tcp
 28080/tcp 28443/tcp 7072/tcp
  protocols: vrp
  masquerade: no
  forward-ports:
  sourceports:
  icmp-blocks:
  rich rules:
```

3. On both nodes where `selinux` is enabled, run this command:

```
setsebool -P haproxy_connect_any 1
```

4. Verify the new configuration is set to on:

```
/usr/sbin/getsebool -a | grep -i haproxy
```

Output, similar to the following is displayed:

```
[root@A59]# /usr/sbin/getsebool -a | grep -i haproxy
haproxy_connect_any --> on
```

## Prepare the VxFlex OS Gateway

Prepare the VxFlex OS Gateway to enable HA on CentOS 7.3 servers.

### Before you begin

Ensure that you have the following:

- Two CentOS 7.3 servers
- JRE v8 (latest) is installed on the VxFlex OS Gateway server.
- Credentials for the VxFlex OS system:
  - MDM IP address
  - MDM admin username (default is admin)
  - MDM admin password
  - LIA password

### About this task

On each CentOS node, you will install the VxFlex OS Gateway and the HA components (HAProxy and Keepalived).

### Procedure

1. Copy the VxFlex OS Gateway RPM to the node.
2. Install the RPM:

```
GATEWAY_ADMIN_PASSWORD=<admin_user_password> rpm -i
<rpm_location/rpm_name>
```

3. Add MDM IP addresses to the VxFlex OS Gateway:
  - a. On the VxFlex OS Gateway server, open this file for editing: `/opt/emc/scaleio/gateway/webapps/ROOT/WEB-INF/classes/gatewayUser.properties`
  - b. Add all MDM IP addresses to this flag: `mdm.ip.addresses=`
4. Modify the VxFlex OS Gateway ports:
  - a. On the VxFlex OS Gateway server, open this file for editing: `vi /opt/emc/scaleio/gateway/conf/catalina.properties`
  - b. Change the VxFlex OS Gateway ports:

From:

```
http.port=80
ssl.port=443
```

To:

```
http.port=28080  
ssl.port=28443
```

c. Save `catalina.properties` using `wq!`

5. Restart the VxFlex OS-gateway service:

```
/etc/init.d/scaleio-gateway restart
```

6. Approve the MDM certificates in the VxFlex OS Installer:

- a. Log in to the VxFlex OS Installer using port 28080 or 28443.
- b. Accept the certificate warning; alternatively, install your own certificate for the Tomcat server.
- c. Enter the default user name, `admin`, and the password defined when the VxFlex OS Installer was prepared, then click **Login**.
- d. Click **Maintain**.
- e. Type the MDM and LIA credentials.
- f. Click **Retrieve system topology**.

At the top of the screen, a red bar is displayed prompting you to accept certificates.

g. Click **Approve all**.

### Results

The VxFlex OS Gateway is installed on the servers, and all security certificates are confirmed.

## Configure HAProxy systems

Configure the HAProxy systems, which serve as a load balancer for the VxFlex OS Gateway, monitoring the VxFlex OS systems and redirecting traffic if a node goes down.

### Before you begin

If you are not using CA certificates, create self-signed certificates, as described in [Create self-signed certificates for HAProxy](#) on page 27.

### About this task

HAProxy1 will run both HAProxy as well as Keepalived. Keepalived is responsible for monitoring the status of HAProxy and configuring the virtual IP address as needed.

### Procedure

1. Install the HAProxy and Keepalived packages:
  - a. Run the following, from a shell:

```
$ sudo yum install keepalived haproxy
```

## 2. Configure HAProxy node1 (haproxy1):

a. Open `/etc/haproxy/haproxy.cfg` for editing.

b. Modify the value of `gateway1` and `gateway2` to match the VxFlex OS Gateway IP addresses.

It is recommended to copy and paste the configuration file. You can remove the existing `frontend` and `backend` sections, replacing them with sections specific to your needs.

After editing, the file should look like this:

```
#-----
# Global settings
#-----
-----
global
# to have these messages end up in /var/log/haproxy.log
you will
# need to:
#
# 1) configure syslog to accept network log events. This
is done
#   by adding the '-r' option to the SYSLOGD_OPTIONS in
#   /etc/sysconfig/syslog
#
# 2) configure local2 events to go to the /var/log/
haproxy.log
#   file. A line like the following can be added to
#   /etc/sysconfig/syslog
#
#   local2.*                               /var/log/haproxy.log
#
log      127.0.0.1 local2

chroot   /var/lib/haproxy
pidfile  /var/run/haproxy.pid
maxconn  4000
user     haproxy
group    haproxy
# daemon

# turn on stats unix socket
stats socket /var/lib/haproxy/stats
tune.ssl.default-dh-param 2048

#-----
# common defaults that all the 'listen' and 'backend'
sections will
# use if not designated in their block
#-----
-----
defaults
mode                http
log                 global
option              httplog
option              dontlognull
option              http-server-close
option              redispatch
retries             3
timeout http-request 10s
timeout queue       1m
timeout connect     10s
timeout client      1m
```

```

        timeout server          1m
        timeout http-keep-alive 10s
        timeout check           10s
        maxconn                 3000

#-----
#-----
# main frontend which proxys to the backends
#-----
#-----
frontend scaleio
    bind *:80
    bind *:443 crt-ignore-err all ssl crt /etc/haproxy/certs/
haproxy.pem
    option tcplog
    mode tcp
    default_backend      siogateway

#-----
#-----
# max connections balancing between the various backends
# this configuration will prefer one system if available,
#   up until it has 256 connections to it
#-----
#-----
backend siogateway
    mode          tcp
    balance       first
    option        ssl-hello-chk
    server        gateway1 <machine1_IP>:28443 maxconn 256 ssl
verify none
    server        gateway2 <machine2_IP>:28443 maxconn 256 ssl
verify none

```

### 3. Configure HAProxy node2 (haproxy2):

a. Open `/etc/haproxy/haproxy.cfg` for editing.

b. Modify the value of `gateway1` and `gateway2` to match the VxFlex OS Gateway IP addresses.

It is recommended to copy and paste the configuration file. You can remove the existing `frontend` and `backend` sections, replacing them with sections specific to your needs.

After editing, the file should look like this:

```

#-----
#-----
# Global settings
#-----
#-----
global
    # to have these messages end up in /var/log/haproxy.log
you will
    # need to:
    #
    # 1) configure syslog to accept network log events. This
is done
    #   by adding the '-r' option to the SYSLOGD_OPTIONS in
    #   /etc/sysconfig/syslog
    #
    # 2) configure local2 events to go to the /var/log/
haproxy.log
    #   file. A line like the following can be added to
    #   /etc/sysconfig/syslog
    #
    #   local2.*                               /var/log/haproxy.log

```



```

#
log          127.0.0.1 local2

chroot       /var/lib/haproxy
pidfile      /var/run/haproxy.pid
maxconn      4000
user         haproxy
group        haproxy
# daemon

# turn on stats unix socket
stats socket /var/lib/haproxy/stats
tune.ssl.default-dh-param 2048

#-----
# common defaults that all the 'listen' and 'backend'
# sections will
# use if not designated in their block
#-----
defaults
    mode                http
    log                  global
    option               httplog
    option               dontlognull
    option               http-server-close
    option               redispatch
    retries              3
    timeout http-request 10s
    timeout queue        1m
    timeout connect      10s
    timeout client       1m
    timeout server       1m
    timeout http-keep-alive 10s
    timeout check        10s
    maxconn              3000

#-----
# main frontend which proxys to the backends
#-----
frontend scaleio
    bind *:80
    bind *:443 crt-ignore-err all ssl crt /etc/haproxy/certs/
haproxy.pem
    option tcplog
    mode tcp
    default_backend      siogateway

#-----
# max connections balancing between the various backends
# this configuration will prefer one system if available,
# up until it has 256 connections to it
#-----
backend siogateway
    mode                tcp
    balance              first
    option               ssl-hello-chk
    server gateway1 <machine1_IP>:28443 maxconn 256 ssl
verify none
    server gateway2 <machine2_IP>:28443 maxconn 256 ssl
verify none

```

#### 4. Set haproxy to start automatically:

a. On each node, run these commands:

```
$ sudo systemctl enable haproxy
$ sudo systemctl start haproxy
```

5. Configure Keepalived on the Master node (haproxy1):

a. Open `/etc/keepalived/keepalived.conf` for editing.

b. Modify these values:

- interface
- state (MASTER on haproxy1)
- priority (101 on haproxy1)
- virtual\_ipaddress (virtual IP address/subnet)

It is recommended to copy and paste the configuration file.

After editing, the file should look like this (comments added):

```
vrrp_script chk_haproxy {
    script "pidof_haproxy" # check the haproxy process
    interval 2 # every 2 seconds
    weight 2 # add 2 points if OK
}

vrrp_instance VI_1 {
    interface ens192 # interface to monitor
    state MASTER # MASTER on haproxy1, BACKUP on haproxy2
    virtual_router_id 51
    priority 101 # 101 on haproxy1, 100 on haproxy2
    virtual_ipaddress {
        <virtual_IP>/24 # virtual ip address and subnet
    }
    track_script {
        chk_haproxy
    }
}
```

6. Configure Keepalived on the Backup node (haproxy2):

a. Open `/etc/keepalived/keepalived.conf` for editing.

b. Modify these values:

- interface
- state (BACKUP on haproxy2)
- priority (100 on haproxy2)
- virtual\_ipaddress (virtual IP address/subnet)

It is recommended to copy and paste the configuration file.

After editing, the file should look like this (comments added):

```
vrrp_script chk_haproxy {
    script "pidof haproxy" # check the haproxy process
    interval 2 # every 2 seconds
    weight 2 # add 2 points if OK
}
```

```

vrp_instance VI_1 {
    interface ens192 # interface to monitor
    state BACKUP # MASTER on haproxy1, BACKUP on haproxy2
    virtual_router_id 51
    priority 100 # 101 on haproxy1, 100 on haproxy2
    virtual_ipaddress {
        <virtual_IP>/24 # virtual ip address and subnet
    }
    track_script {
        chk_haproxy
    }
}

```

**7. Modify the Keepalived service to start after networking:**

- a. Open `/lib/systemd/system/keepalived.service` for editing.
- b. Change the value of `After=syslog.target` from `network.target` to `network-online.target`

After editing, the file should look like this:

```

[Unit]
Description=LVS and VRRP High Availability Monitor
After=syslog.target network-online.target

[Service]
Type=forking
PIDFile=/var/run/keepalived.pid
KillMode=process
EnvironmentFile=-/etc/sysconfig/keepalived
ExecStart=/usr/sbin/keepalived $KEEPALIVED_OPTIONS
ExecReload=/bin/kill -HUP $MAINPID

[Install]
WantedBy=multi-user.target

```

**8. Set Keepalived to start automatically:**

- a. On each node, run these commands:

```

$ sudo systemctl enable keepalived
$ sudo systemctl start keepalived

```

## Create self-signed certificates for HAProxy

If not using certificates from a CA, you must generate self-signed certificates for the nodes running HAProxy.

### Procedure

1. On the first node, create the certificates:

```

$ sudo mkdir /etc/haproxy/certs
$ cd /etc/haproxy/certs
$ sudo openssl genrsa -out haproxy.key 1024
$ sudo openssl req -new \
    -key haproxy.key \
    -out haproxy.csr
$ sudo openssl x509 -req -days 365 \
    -in haproxy.csr \

```

```
-signkey haproxy.key \  
-out haproxy.crt  
$ sudo cat haproxy.crt haproxy.key > haproxy.pem
```

2. Copy the entire certificates directory to the other node.