# How to Setup Highly Available NGINX with KeepAlived in Linux

Pradeep Kumar / November 17, 2020

As we know NGINX is a highly rated web server which can also be used as reverse proxy, load balancer and HTTP cache. In this article, we will demonstrate how to setup highly available (HA) NGINX web server with keepalived in Linux. Keepalived works on VRRP (Virtual Router Redundancy Protocol) which allows one static IP to be fail-over between two Linux systems.

Following are my lab details for NGINX HA:

- Node 1 - 192.168.1.130 - nginx1.example.com - minimal CentOS 8 / RHEL 8
- Node 2 - 192.168.1.140 - nginx2.example.com - minimal CentOS 8 / RHEL 8
- Virtual IP (VIP) - 192.168.1.150
- sudo user pkumar
- Firewalld enbled
- SELinux Running

Let's jump into the Installation and configuration steps,

## Step 1) Install NGINX Web Server from command line

NGINX package is available in the default CentOS 8 / RHEL 8 repositories, so run below dnf command on both the nodes to install nginx web sever

```
$ sudo dnf install -y nginx
```

## For CentOS 7 / RHEL 7

NGINX package is not available in default CentOS 7 / RHEL 7 repositories, so to install it first we have to enable epel repository. Run the following command on both the nodes

```
$ sudo yum install epel-release -y
$ sudo yum install -y nginx
```

### For Ubuntu / Debian

For Debian based Linux distributions, nginx web server package is available in default package repositories, so to install nginx, run

```
$ sudo apt update
$ sudo apt install -y nginx
```

### Step 2) Configure Custom index.html file for both nodes

Let's create custom index.html file for both the nodes so that we can easily identify which server is serving the web site while accessing via virtual ip.

For node 1, run following echo command,

```
[pkumar@nginx1 ~]$ echo "<h1>This is NGINX Web Server from Node
1</h1>" | sudo tee /usr/share/nginx/html/index.html
```

For node 2, run

```
[pkumar@nginx2 ~]$ echo "<h1>This is NGINX Web Server from Node
2</h1>" | sudo tee /usr/share/nginx/html/index.html
```

### Step 3) Allow NGINX port in firewall and start its service

In case firewall is enabled and running on both the nodes then allow port 80 by executing following commands,

### For CentOS / RHEL System

```
$ sudo firewall-cmd --permanent --add-service=http
$ sudo firewall-cmd –reload
```

For Ubuntu / Debian System

```
$ sudo ufw allow 'Nginx HTTP'
```

Start and enable nginx service by running beneath command commands on both the nodes,

```
$ sudo systemctl start nginx
$ sudo systemctl enable nginx
```

Test NGINX Web server of both the nodes by running following curl command from outside,

```
$ curl http://192.168.1.130
<h1>This is NGINX Web Server from Node 1</h1>
$ curl http://192.168.1.140
<h1>This is NGINX Web Server from Node 2</h1>
```

Perfect, above command's output confirm that nginx is running and accessible from outside with system's ip address.

## Step 4) Install and Configure Keepalived

For CentOS / RHEL systems, keepalived package and its dependencies are available in the default package repositories, so its installation is straight forward, just run below command on both the nodes.

```
$ sudo dnf install -y keepalived       // CentOS 8/ RHEL 8
$ sudo yum install -y keepalived      // CentOS 7 / RHEL 7
```

For Ubuntu / Debian System,

```
$ apt install -y keepalived
```

Once the keepalived is installed then configure it by editing its configuration file '/etc/keepalived/keepalived.conf'. We will keep node 1 as master node and node 2 as backup node.

Take backup of configuration file,

```
[pkumar@nginx1 ~]$ sudo cp /etc/keepalived/keepalived.conf
/etc/keepalived/keepalived.conf-org
```

Replace the content of keepalived.conf with below:

```
[pkumar@nginx1 ~]$ echo -n | sudo tee /etc/keepalived/keepalived.conf
[pkumar@nginx1 ~]$ sudo vi /etc/keepalived/keepalived.conf
```

Paste the following contents

```
global_defs {
  # Keepalived process identifier
  router_id nginx
}

# Script to check whether Nginx is running or not
vrrp_script check_nginx {
  script "/bin/check_nginx.sh"
  interval 2
  weight 50
}

# Virtual interface - The priority specifies the order in which the
assigned interface to take over in a failover
```

```
  vrrp_instance VI_01 {
    state MASTER
    interface enp0s3
    virtual_router_id 151
    priority 110

    # The virtual ip address shared between the two NGINX Web Server
  which will float
    virtual_ipaddress {
      192.168.1.150/24
    }
    track_script {
      check_nginx
    }
    authentication {
      auth_type AH
      auth_pass secret
    }
  }
```

```
global_defs {
  # Keepalived process identifier
  router_id nginx
}
# Script to check whether Nginx is running or not
vrrp_script check_nginx {
  script "/bin/check_nginx.sh"
  interval 2
  weight 50
}
# Virtual interface - The priority specifies the order in which the assigned interface to take over in a failover
vrrp_instance VI_01 {
  state MASTER
  interface enp0s3
  virtual_router_id 151
  priority 110
  # The virtual ip address shared between the two NGINX Web Server which will float
  virtual_ipaddress {
    192.168.1.150/24
  }
  track_script {
    check_nginx
  }
  authentication {
    auth_type AH
    auth_pass secret
  }
}
```

Now create a script with the following contents which will check whether nginx service is running or not. Keepalived will always check the output of check_nginx.sh script, if it finds that

nginx service is stopped or not responding then it will move virtual ip address on backup node.

```
[pkumar@nginx1 ~]$ sudo vi /bin/check_nginx.sh
#!/bin/sh
if [ -z "`pidof nginx`" ]; then
  exit 1
fi
```

save & close the file and set the required permission with chmod command,

```
[pkumar@nginx1 ~]$ sudo chmod 755 /bin/check_nginx.sh
```

Now copy the keepalived.conf and check_nginx.sh files from node 1 to node 2 using following scp command.

```
[pkumar@nginx1 ~]$ scp /etc/keepalived/keepalived.conf
root@192.168.1.140:/etc/keepalived/
[pkumar@nginx1 ~]$ scp /bin/check_nginx.sh root@192.168.1.140:/bin/
```

Once files are copied then login to Node 2 and make couple of changes in keepalived.conf file. Change State from MASTER to BACKUP and lower the priority by setting it as 100. After making the changes, keepalived.conf on Node 2 would look like below,

```
global_defs {
  # Keepalived process identifier
  router_id nginx
}
# Script to check whether Nginx is running or not
vrrp_script check_nginx {
  script "/bin/check_nginx.sh"
  interval 2
  weight 50
}
# Virtual interface - The priority specifies the order in which the assigned interface to take over in a failover
vrrp_instance VI_01 {
  state BACKUP
  interface enp0s3
  virtual_router_id 151
  priority 100
  # The virtual ip address shared between the two NGINX Web Server which will float
  virtual_ipaddress {
    192.168.1.150/24
  }
  track_script {
    check_nginx
  }
  authentication {
    auth_type AH
    auth_pass secret
  }
}
```

In Case OS firewall is running then allow VRRP by the running following commands,

**Note** - Execute these commands on both the nodes

For CentOS / RHEL Systems

```
$ sudo firewall-cmd --add-rich-rule='rule protocol value="vrrp"
accept' --permanent
$ sudo firewall-cmd --reload
```

For Ubuntu / Debian Systems

Allow VRRP by executing followings, from the master node (Node 1), run

```
$ sudo ufw allow to 224.0.0.18 comment 'VRRP Broadcast'
$ sudo ufw allow from 192.168.1.140 comment 'VRRP Router'
```

From the Backup / Slave Node (Node 2)

```
$ sudo ufw allow to 224.0.0.18 comment 'VRRP Broadcast'
$ sudo ufw allow from 192.168.1.130 comment 'VRRP Router'
```

Now finally start keepalived service by running beneath systemctl commands from both the nodes,

```
$ sudo systemctl start keepalived
$ sudo systemctl enable keepalived
```

Verify the keepalived service by running below:

```
$ sudo systemctl status keepalived
```

```
[pkumar@nginx1 ~]$ sudo systemctl status keepalived
● keepalived.service - LVS and VRRP High Availability Monitor
   Loaded: loaded (/usr/lib/systemd/system/keepalived.service; enabled; vendor preset: disabled)
   Active: active (running) since Wed 2020-11-11 03:35:29 GMT; 14min ago
  Process: 1231 ExecStart=/usr/sbin/keepalived $KEEPALIVED_OPTIONS (code=exited, status=0/SUCCESS)
 Main PID: 1241 (keepalived)
    Tasks: 2 (limit: 12536)
   Memory: 8.8M
   CGroup: /system.slice/keepalived.service
           ├─1241 /usr/sbin/keepalived -D
           └─1242 /usr/sbin/keepalived -D

Nov 11 03:35:32 nginx1.example.com Keepalived_vrrp[1242]: Sending gratuitous ARP on enp0s3 for 192.168.1.150
Nov 11 03:35:32 nginx1.example.com Keepalived_vrrp[1242]: Sending gratuitous ARP on enp0s3 for 192.168.1.150
Nov 11 03:35:32 nginx1.example.com Keepalived_vrrp[1242]: Sending gratuitous ARP on enp0s3 for 192.168.1.150
Nov 11 03:35:32 nginx1.example.com Keepalived_vrrp[1242]: Sending gratuitous ARP on enp0s3 for 192.168.1.150
Nov 11 03:35:37 nginx1.example.com Keepalived_vrrp[1242]: Sending gratuitous ARP on enp0s3 for 192.168.1.150
Nov 11 03:35:37 nginx1.example.com Keepalived_vrrp[1242]: (VI_01) Sending/queueing gratuitous ARPs on enp0s3 for 192.168.1.150
Nov 11 03:35:37 nginx1.example.com Keepalived_vrrp[1242]: Sending gratuitous ARP on enp0s3 for 192.168.1.150
Nov 11 03:35:37 nginx1.example.com Keepalived_vrrp[1242]: Sending gratuitous ARP on enp0s3 for 192.168.1.150
Nov 11 03:35:37 nginx1.example.com Keepalived_vrrp[1242]: Sending gratuitous ARP on enp0s3 for 192.168.1.150
```

Perfect, now verify VIP (virtual ip address) status on master node, in our case VIP is 192.168.1.130

```
$ ip add show
```

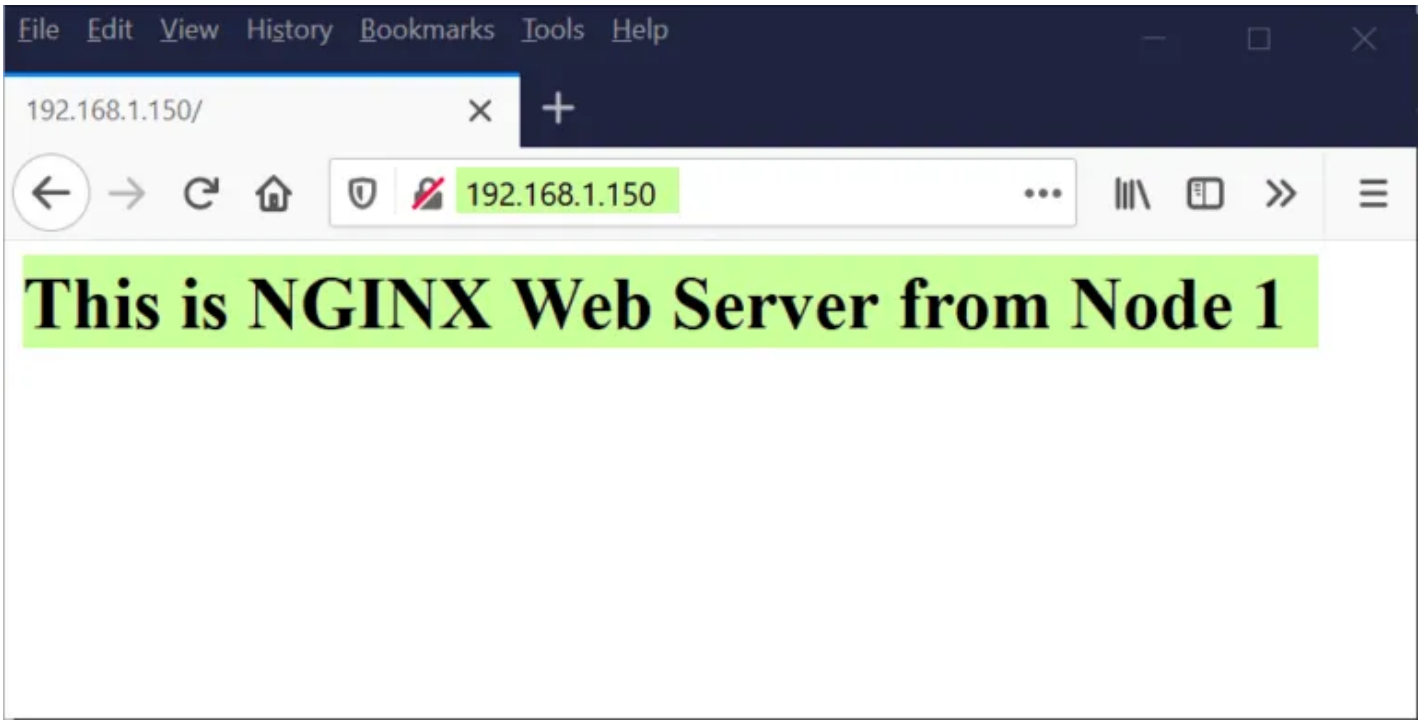
```
[pkumar@nginx1 ~]$ ip add show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:ab:d0:e6 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.130/24 brd 192.168.1.255 scope global noprefixroute enp0s3
       valid_lft forever preferred_lft forever
    inet 192.168.1.150/24 scope global secondary enp0s3
       valid_lft forever preferred_lft forever
    inet6 fe80::26f1:d797:4c6f:b761/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
[pkumar@nginx1 ~]$
```

Above output confirms VIP is configure on master node on its enp0s3 interface. So, lets do keepalived and nginx testing in the next step.

### Step 5) Keepalived and NGINX Testing

To perform the testing, try access nginx web server with virtual IP (192.168.1.150), currently it should show us node 1 nginx page.

Open the wen browser and type 'http://192.168.1.150' and hit enter,




Now try to stop the NGINX service on node 1 and see whether virtual IP is switched from Node 1 to Node 2 and then try to access nginx web page with VIP (192.168.1.150) and this time it should show us nginx page from node 2.

```
[pkumar@nginx1 ~]$ sudo systemctl stop nginx
[pkumar@nginx1 ~]$ ip add show
```

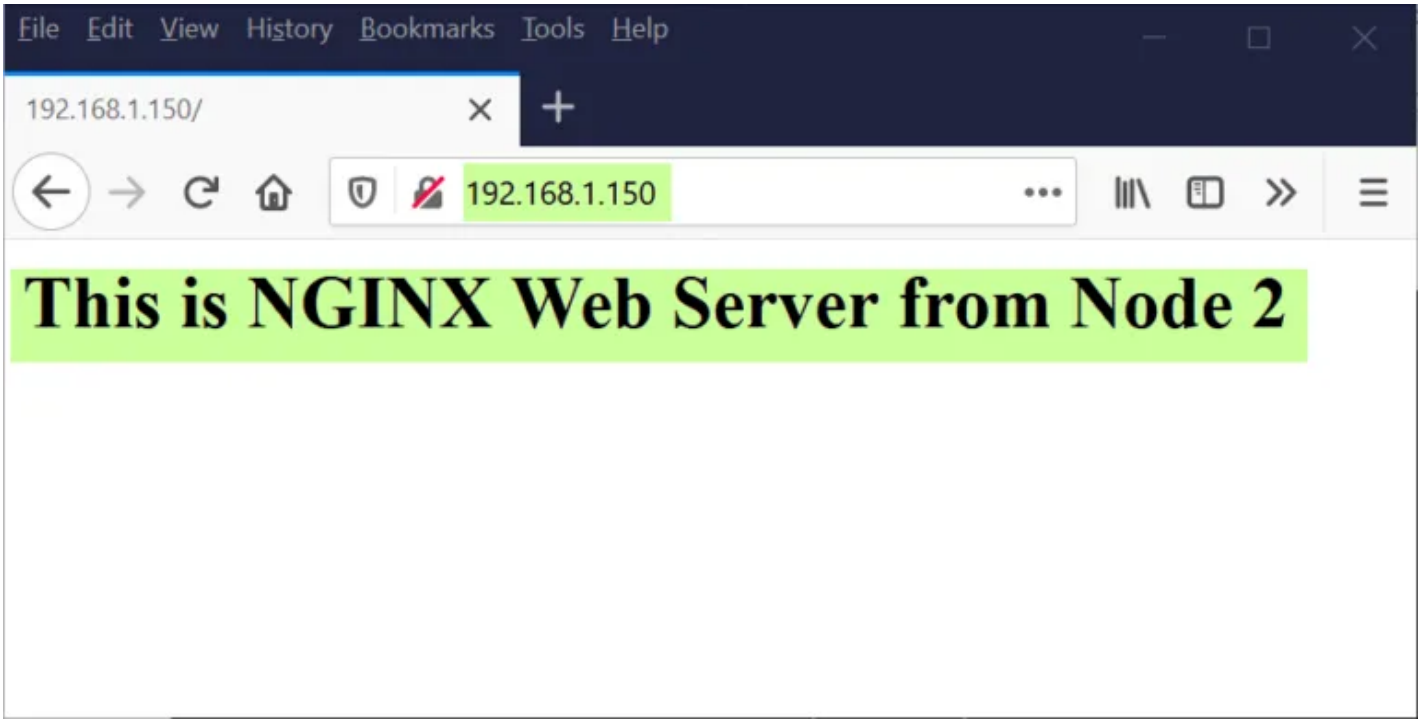
```
[pkumar@nginx1 ~]$ sudo systemctl stop nginx
[sudo] password for pkumar:
[pkumar@nginx1 ~]$ ip add show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:ab:d0:e6 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.130/24 brd 192.168.1.255 scope global noprefixroute enp0s3
       valid_lft forever preferred_lft forever
    inet6 fe80::26f1:d797:4c6f:b761/64 scope link noprefixroute
```

Login to node 2 and run ip command to see verify the virtual IP address,

```
[pkumar@nginx2 ~]$ ip add show
```

```
[pkumar@nginx2 ~]$ ip add show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:ab:d0:e6 brd ff:ff:ff:ff:ff:ff
    inet 192.168.1.140/24 brd 192.168.1.255 scope global noprefixroute enp0s3
       valid_lft forever preferred_lft forever
    inet 192.168.1.150/24 scope global secondary enp0s3
       valid_lft forever preferred_lft forever
    inet6 fe80::f9ac:3c93:d32f:927f/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
[pkumar@nginx2 ~]$
```

Now, let's try to access web page using virtual ip,




Great, above confirms that we have successfully setup highly available NGINX Web server with keepalived. That's all from this article, please do share your feedback, comments and suggestions.

Tags: NGINX HA

## 1 thought on "How to Setup Highly Available NGINX with KeepAlived in Linux"

**TAI**
March 21, 2021 at 2:12 am

Many thanks for your topic.

In my case: My master 1 have ip: 192.168.56.8. Master 2 have ip: 192.168.56.10
And VIP is: 192.168.56.22

When I do the step 5, I stopped the nginx but it still keep the VIP at the master 1: inet 192.168.56.22/24 scope global secondary enp0s8

And I can't connect to VIP