# Руководство по настройке keepalived для nginx на CentOS 8

Отказоустойчивый кластер для nginx можно сделать на основе VRRP. Для этого используем службу keepalived.

Пусть имеются 2 сервера nginx (установку и настройку см. в «Руководство по установке Nginx под CentOS 8») c IP (обязательно в одной сети):

192.168.1.165

192.168.1.167

Будем считать 192.168.1.167 виртуальным IP.

Устанавливаем keepalived (можно установить локально из пакетов, можно через репозиторий из интернет):

# yum install keepalived

# systemctl enable keepalived  
# systemctl start keepalived

На каждом сервере редактируем keepalived.conf:

# vim /etc/keepalived/keepalived.conf

Блок global\_defs нужен только для настройки рассылки, его можно удалить. eth0 если отличается – заменить на свой.

На первом сервере:

! nginx1

global\_defs {

notification\_email {

primo.rpa@mail.ru

}

notification\_email\_from primo.rpa@mail.ru

smtp\_server localhost

smtp\_connect\_timeout 30

! enable\_script\_security

router\_id nginx1.primo

}

vrrp\_script check\_httpstatus {

script "/opt/scripts/check\_httpstatus.sh 127.0.0.1[[1]](#footnote-1)"

interval 5

timeout 150

weight 100

}

vrrp\_instance NGINX\_1 {

state BACKUP

interface eth0

virtual\_router\_id 26

priority 27

advert\_int 1

smtp\_alert

authentication {

auth\_type PASS

auth\_pass nginx

}

virtual\_ipaddress {

192.168.1.167

}

track\_script {

check\_httpstatus

}

}

На втором сервере:

! nginx2

global\_defs {

notification\_email {

primo.rpa@mail.ru

}

notification\_email\_from primo.rpa@mail.ru

smtp\_server localhost

smtp\_connect\_timeout 30

! enable\_script\_security

router\_id nginx2.primo

}

vrrp\_script check\_httpstatus {

script "/opt/scripts/check\_httpstatus.sh 127.0.0.1"

interval 5

timeout 150

weight 100

}

vrrp\_instance NGINX\_2 {

state MASTER

interface eth0

virtual\_router\_id 26

priority 28

advert\_int 1

smtp\_alert

authentication {

auth\_type PASS

auth\_pass nginx

}

virtual\_ipaddress {

192.168.1.167

}

track\_script {

check\_httpstatus

}

}

На каждом сервере добавляем разрешающие правила для multicast траффика и VRRP протокола с помощью iptables:

# iptables -A INPUT -i eth0 -d 224.0.0.0/8 -j ACCEPT  
# iptables -A INPUT -p vrrp -i eth0 -j ACCEPT

На каждом сервере создаем скрипт проверки доступности nginx:

# mkdir /opt/scripts

# vim /opt/scripts/check\_httpstatus.sh

#!/bin/bash

#

ch=$(curl -m 1 -I @${1} -o /dev/null -w '%{http\_code}\n' -s)

if [ $ch == 200 ]

then

exit 0

else

exit 1

fi

Делаем скрипт исполняемым:

# sudo chmod +x /opt/scripts/check\_httpstatus.sh

Перезапускаем keepalived:

# systemctl restart keepalived

# systemctl status keepalived

Смотрим текущие IP адреса на интерфейсе eth0 серверов:

# ip a show eth0

Чтобы разрешить nginx слушать на не существующих IP, включаем в ядре net.ipv4.ip\_nonlocal\_bind:

# sudo sysctl -w net.ipv4.ip\_nonlocal\_bind=1

# sysctl net.ipv4.ip\_nonlocal\_bind

В секции server конфигурационного файла nginx прописываем слушать на обоих IP:

# vim /etc/nginx/nginx.conf

server {

listen 192.168.1.165:80;

listen 192.168.1.167:80;

…

Перезапускаем nginx:

# sudo systemctl restart nginx

# sudo systemctl status nginx

Проверка кластера сводится к тому, что nginx должен отвечать по адресу 192.168.1.167 (виртуальному IP) при поочередной недоступности серверов 1 и 2.

1. Тут должен быть адрес, который вернет 200 [↑](#footnote-ref-1)