УДК 004.056

Пестов Игорь Евгеньевич кандидат технических наук, доцент
Санкт-Петербургский государственный университет телекоммуникаций имени профессора М. А. Бонч-Бруевича Россия, г. Санкт-Петербург
Жвакин Алексей Владимирович студент
5 курс, Санкт-Петербургский государственный университет телекоммуникаций имени профессора М. А. Бонч-Бруевича
Россия, г. Санкт-Петербург
Чезлов Кирилл Сергеевич студент
5 курс, Санкт-Петербургский государственный университет телекоммуникаций имени профессора М. А. Бонч-Бруевича
Россия, г. Санкт-Петербург

ПОСТРОЕНИЕ ОТКАЗОУСТОЙЧИВОГО КЛАСТЕРА БАЛАНСИРОВЩИКОВ НАГРУЗКИ НА ОСНОВЕ HAPROXY И KEEPALIVED

Аннотация: Статья посвящена построению отказоустойчивого кластера балансировщиков нагрузки с использованием связки программных решений HAProxy и Keepalived. Рассматриваются архитектура и принципы совместной работы данных технологий для обеспечения высокой доступности и автоматического переключения при сбоях. Подробно описан процесс развертывания и настройки компонентов кластера. Предложенное решение позволяет минимизировать время простоя системы, эффективно распределять нагрузку между серверами и автоматически восстанавливать работоспособность при отказах отдельных компонентов.

Ключевые слова: отказоустойчивость, кластер, балансировка нагрузки, высокая доступность, HAProxy, Keepalived, автоматическое переключение, распределенные системы.

Annotation: The article is devoted to building a fault-tolerant load balancer cluster using a combination of the HAProxy and Keepalived software solutions. It examines

the architecture and principles of interaction between these technologies to ensure high availability and automatic failover. The process of deploying and configuring the clusteыr components is described in detail. The proposed solution minimizes system downtime, efficiently distributes load among servers, and automatically restores functionality in case of failures of individual components.

Key words: fault tolerance, cluster, load balancing, high availability, HAProxy, Keepalived, automatic failover, distributed systems.

Введение. В современных высоконагруженных системах обеспечение отказоустойчивости и распределения нагрузки является критически важной задачей. Особую сложность представляет организация бесперебойной работы кластеров баз данных и приложений, где простои могут привести к значительным репутационным и финансовым потерям. В данной статье рассматривается решение этой задачи с использованием связки технологий HAProxy и Keepalived. HAProxy выполняет балансировку запросов между узлами, а Keepalived гарантирует отказоустойчивость самих балансировщиков. В работе подробно описывается процесс настройки каждого компонента, включая конфигурацию слушателей HAProxy, проверку доступности узлов и настройку виртуального IP-адреса с помощью Keepalived. Актуальность темы обусловлена необходимостью создания надежных и масштабируемых систем, способных выдерживать высокие нагрузки и автоматически восстанавливаться после сбоев.

Архитектура кластера балансировщиков нагрузки. В современных распределённых системах балансировка нагрузки и обеспечение отказоустойчивости являются ключевыми компонентами для поддержания высокой доступности сервисов. Для решения этих задач используются специализированные программные решения, такие как HAProxy и Keepalived.

HAProxy — это высокопроизводительный балансировщик нагрузки, работающий на 4-м (транспортном) и 7-м (прикладном) уровнях модели OSI. Основные возможности:

- Распределение запросов между серверами с использованием различных алгоритмов (round-robin, leastconn и др.);
- Проверка работоспособности узлов;
- Поддержка SSL/TLS терминации;
- Гибкая система мониторинга и статистики.

Keepalived реализует механизм отказоустойчивости на основе протокола VRRP (Virtual Router Redundancy Protocol). Его основные функции:

- Мониторинг состояния сервисов;
- Автоматическое переключение виртуального IP-адреса;
- Обеспечение высокой доступности за счёт резервирования.

Принципы совместной работы:

1. Распределение нагрузки: HAProxy равномерно распределяет входящие запросы между рабочими серверами, используя выбранный алгоритм балансировки;
2. Мониторинг состояния: регулярные проверки работоспособности узлов позволяют исключать неработоспособные узлы из балансировки;
3. Обеспечение отказоустойчивости: Keepalived отслеживает состояние HAProxy и при его сбое автоматически переключает виртуальный IP-адрес на резервный узел;
4. Минимизация времени простоя: комбинация этих технологий позволяет сократить время восстановления после сбоев до минимума.

Развертывание кластера балансировщиков нагрузки. Установим HAProxy (рис. 1) и напишем его конфигурацию. Для проверки доступности всех компонентов кластера и балансировки запросов между ними будем использовать «слушатели» (рис. 2). Настроим каждый слушатель в соответствии с необходимыми адресами и портами приложений, укажем алгоритмы балансировки, опции проверки доступности. Обратим внимание на то, что для проверки доступности используются HTTP запросы на различные конечные адреса (endpoint) и в качестве критерия работоспособности

используются разные статус-коды ответов, зависящие от конкретного приложения. Также включим сбор статистики.

```
root@astra-claster1:~# apt install haproxy
Чтение списков пакетов… Готово
Построение дерева зависимостей
Чтение информации о состоянии… Готово
Уже установлен пакет haproxy самой новой версии (2.2.32-5astra14).
Обновлено 0 пакетов, установлено 0 новых пакетов, для удаления отмечено 0 пакетов, и 0 пакетов не обновлено.
```

Рисунок 1. Установка HAProxy

```
listen Loadbalancer
    mode http
    bind *:3000
    stats enable
    stats refresh 10s
    stats uri /
    option httpchk GET /
    server haproxy_10.10.110.73 10.10.110.73:3000 check
    server haproxy_10.10.110.74 10.10.110.74:3000 check

listen APP
    bind *:80
    mode http
    balance source
    option httpchk GET /ankey/info/version
    http-check expect status 401
    server ankey_10.10.110.73 10.10.110.73:8080 check
    server ankey_10.10.110.74 10.10.110.74:8080 check
    server ankey_10.10.110.75 10.10.110.75:8080 check
    server ankey_10.10.110.76 10.10.110.76:8080 check

listen OpenSearch
    bind *:8090
    mode http
    balance roundrobin
    option httpchk GET /
    http-check expect status 401
    server elasticsearch_10.10.110.75 10.10.110.75:8111 check
    server elasticsearch_10.10.110.76 10.10.110.76:9200 check


listen PostgreSQL_cluster_master
    bind *:5435
    option httpchk
    http-check expect status 200
    default-server inter 3s fall 3 rise 2 on-marked-down shutdown-sessions
    server astra-pg-cluster_10.10.110.73 10.10.110.73:5432 maxconn 10000 check port 8008
    server astra-pg-cluster_10.10.110.74 10.10.110.74:5432 maxconn 10000 check port 8008
    server astra-pg-cluster_10.10.110.75 10.10.110.75:5432 maxconn 10000 check port 8008
    server astra-pg-cluster_10.10.110.76 10.10.110.76:5432 maxconn 10000 check port 8008
```

Рисунок 2. Конфигурация слушателей HAProxy

Слушатель «PostgreSQL_cluster_master» отслеживает кто из экземпляров PostgreSQL является основным узлом кластера СУБД, поэтому дополнительно будем проверять доступность всех развернутых единиц кластера СУБД (рис. 3).

```
backend PostgreSQL_servers
    mode tcp
    server postgres_10.10.110.73 10.10.110.73:5432 check
    server postgres_10.10.110.74 10.10.110.74:5432 check
    server postgres_10.10.110.75 10.10.110.75:5432 maxconn 100 check
    server postgres_10.10.110.76 10.10.110.76:5432 check
(END)
```

Рисунок 3. Конфигурация бэкенда HAProxy

Теперь установим пакеты «keepalived» и «psmisc». Напишем конфигурацию Keepalived (рис. 4). Здесь укажем необходимый виртуальный IP-адрес, роль (для второго сервера нужно указать «BACKUP»), приоритет и интерфейс текущего настраиваемого сервера. Скрипт «chk_haproxy» будет проверять, запущен ли процесс «haproxy» и, в противном случае передаст роль мастера другому серверу.

```
vrrp_script chk_haproxy {
        script "killall -0 haproxy"
        interval 2
        weight 2
}

vrrp_instance VI_1 {
        interface eth0
        state MASTER
        virtual_router_id 51
        priority 102

        virtual_ipaddress {
                10.10.110.2
        }

        track_script {
                chk_haproxy
        }
}
```

Рисунок 4. Конфигурация Keepalived

После завершения всех настроек запустим HAProxy и Keepalived. Перейдя по настроенному адресу статистики увидим, что кластер балансировщиков нагрузки работает корректно (рис. 5).

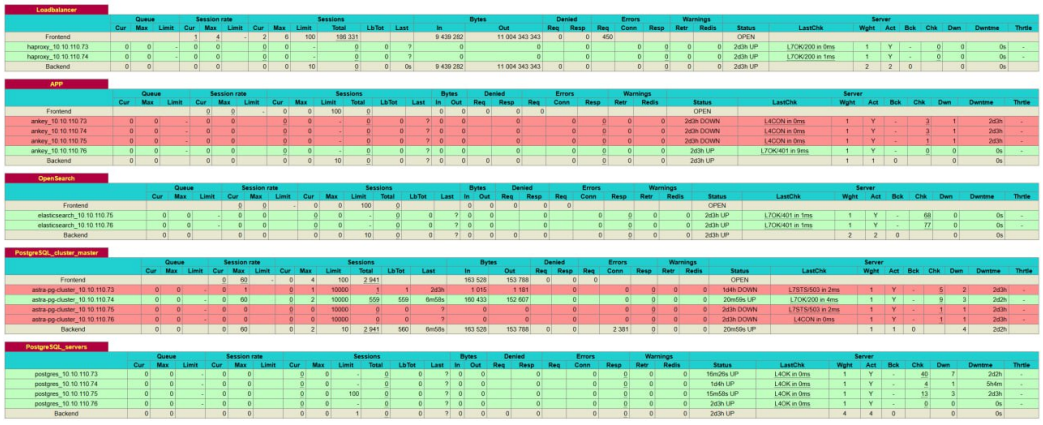

Рисунок 5. Статистика балансировщика нагрузки

Заключение. В ходе работы был успешно развернут отказоустойчивый кластер балансировщиков нагрузки HAProxy и Keepalived. Настроены балансировка нагрузки между узлами различных приложений, обеспечена проверка их доступности, а также реализована отказоустойчивость балансировщиков с помощью Keepalived. Предложенное решение эффективно распределяет нагрузку и автоматически перераспределяет нагрузку между работоспособными узлами в случае сбоев, минимизируя время простоя системы. Дальнейшее развитие проекта может включать оптимизацию алгоритмов балансировки в соответствии с возможностями отдельных приложений, расширение кластера дополнительными узлами и интеграцию с системами мониторинга для более детального анализа работы. Применение

подобных технологий позволяет создавать высоконадежные и масштабируемые инфраструктуры, отвечающие требованиям современных IT-систем.

Использованные источники:

1. HAProxy Technologies. HAProxy: Официальная документация [Электронный ресурс]. URL: https://www.haproxy.org/ (дата обращения: 12.05.2025).

2. Keepalived for Linux: Документация [Электронный ресурс]. URL: https://keepalived.readthedocs.io/ (дата обращения: 12.05.2025).

3. Защита для распределенных отказов в обслуживании в облачных вычислениях / А. М. Гельфанд, Н. А. Косов, А. В. Красов, Г. А. Орлов // Актуальные проблемы инфотелекоммуникаций в науке и образовании (АПИНО 2019) : сборник научных статей VIII Международной научно-технической и научно-методической конференцияи : в 4 т., Санкт-Петербург, 27–28 февраля 2019 года. Том 1. – Санкт-Петербург: Санкт-Петербургский государственный университет телекоммуникаций им. проф. М.А. Бонч-Бруевича, 2019. – С. 329-334. – EDN BLNOBQ.

4. Балансировка нагрузки и отказоустойчивость в современных информационных системах. / С. П. Иванов, И. Н. Морозов // М.: Солон-Пресс. – 2021. – 248 с.

5. Глушаков И. А. Высоконагруженные системы: принципы проектирования, кластеризация, отказоустойчивость. / И. А. Глушаков // СПб.: Питер – 2020. – 384 с.

6. Клюев В. В. Надежность и отказоустойчивость информационных систем. / Клюев В. В. // Горячая линия – Телеком – 2020. – 288 с.