

Основные параметры Дополнительные параметры


*Название

*Тенант

Выключено

*Тип

*URL X

URL  X

Далее необходимо перейти во вкладку дополнительные параметры, нас интересует параметр **Политика выбора URL (URL selection policy)**, ниже описаны детали каждого метода:

1. **Любой (Any)** - выбор любого доступного узла для отправки событий. Если в начале был выбран узел URL1, то события будут лететь в него до тех пор, пока он доступен, в случае выхода его из строя - будет выбран узел URL2, который тоже будет получать события пока сам не выйдет из строя. И так пока не закончатся активные узлы.
2. **Сначала первый (Prefer First) (рекомендуется для корреляторов)** - события отправляются в URL1 до тех пор, пока сервис (URL1) живой. Если он выйдет из строя, события полетят в URL2. Когда URL1 снова станет активным, поток снова направится в него.
3. **По очереди (Round robin) (рекомендуется для хранилищ и агента KUMA для балансировки коллекторов)** - отправляются события не по одному, а пачками. Количество событий в пачке почти всегда будет уникальным - пачка формируется, либо когда достигнут лимит ее размера, либо когда сработает таймер. Каждый URL получит равное количество пачек.

Параметр **Ожидание проверки работоспособности (Health check timeout)** - задает периодичность запросов Health check.

Настроим балансировку потока событий для двух коллекторов:

<input type="checkbox"/>	<input checked="" type="checkbox"/>	Collector	TEST (TCP/5577)	2.0.0.306	Main
<input type="checkbox"/>	<input checked="" type="checkbox"/>	Collector	TEST HA (TCP/5578)	2.0.0.306	Main

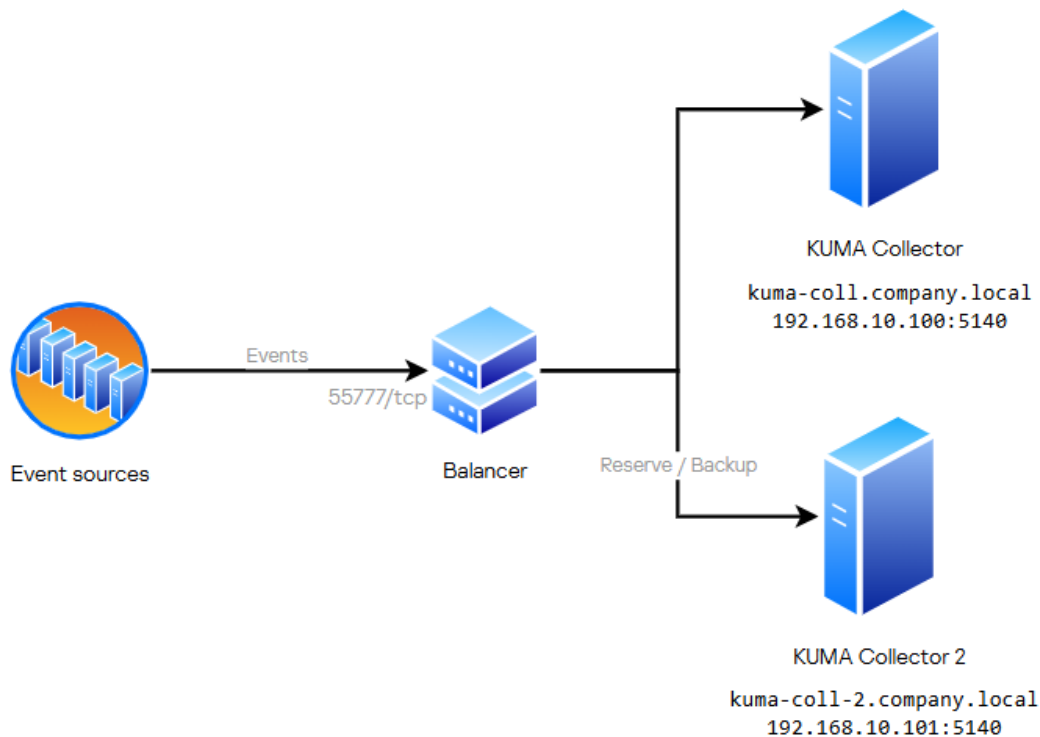
Конфигурация HAпроху будет следующая (путь файла конфигураций по умолчанию - `/etc/haproxy/haproxy.cfg`):

```
defaults
    timeout connect 5s
    timeout client 1m
    timeout server 1m

listen collectorTEST
    bind *:55777
    mode tcp
    balance roundrobin
    server tcp1 <IP_КОЛЛЕКТОРА_1>:5577 check
    server tcp2 <IP_КОЛЛЕКТОРА_2>:5577 backup check
```

Балансировщик прослушивает порт 55777, вы можете поменять конфигурацию порта на свое.

Опция backup используется для того, чтобы использовать этот (сервер с этой опцией) сервер в случае отказа первого.



Для запуска HAпроху:

```
haproxy -D -f /etc/haproxy/haproxy.cfg -p /var/run/haproxy.pid -sf $(cat /var/run/haproxy.pid)
```

Для остановки HAпроху:

```
killall haproxy
```

Для типа коннектора SQL прокси не подойдет и потребуется написание скрипта для реализации следующего:

- создано 2 микросервиса коллектора SQL с одним конфигом
- установлены на сервера Coll-A и Coll-B
- один микросервис включен, второй - выключен
- у каждого из них сервисов в точке монтирования какая-то сетевая папка /opt/kaspersky/kuma/collector/<ID>/sql/ т.е. стейт (состояния) с позициями, которые уже были загружены из базы будут общими
- скрипт произаодит мониторинг за событиями аудита KUMA и если "микросервис перешел из зеленого в красный" и как только один SQL коллектор на сервере A стал красным - запускаем скриптом микросервис на сервере B и он начнет читать из базы из того же места (где остановился сервис A)

Отказоустойчивость за счет Rsyslog: [Управление потоком событий с помощью rsyslog \(kaspersky.com\)](https://kaspersky.com/management-events-with-rsyslog)

- Ручное резервирование LB (Load Balancer). можно создать клон виртуальной машины с LB или скопировать конфигурацию на резервный "железный" сервер и в случае необходимости переключить трафик на резерв руками или с помощью какой-либо автоматизации. Возможны длительные задержки с момента обнаружения проблемы с LB до момента переключения трафика на резервный LB. FQDN и IP LB при этом переносятся на резервный хост;
- Можно использовать службу **keepalived**, позволяющая использовать VRRP, пример настройки: https://docs.oracle.com/en/operating-systems/oracle-linux/6/admin/section_sm3_svy_4r.html. В качестве tcp балансировщиков могут выступать машины с nginx или haproxy. Переключение на резервный балансировщик при этом осуществляется автоматически. Такой подход может оказаться неприменим при размещении хостов с tcp балансировщиками в разных data-центрах, т.к. они использует VRRP;
- Для Nginx можно использовать коммерческую версию, пример - <https://www.nginx.com/products/nginx/high-availability/>;
- Отказоустойчивость компонентов также можно обеспечить функционалом систем виртуализации;
- Использование GSLB или других DNS-based балансировщиков "поверх" tcp LB, при такой схеме можно не покупать коммерческий Nginx.

Revision #26

Created 2023-08-11 10:08:51 UTC by Boris RZR

Updated 2026-05-21 08:31:45 UTC by Boris RZR