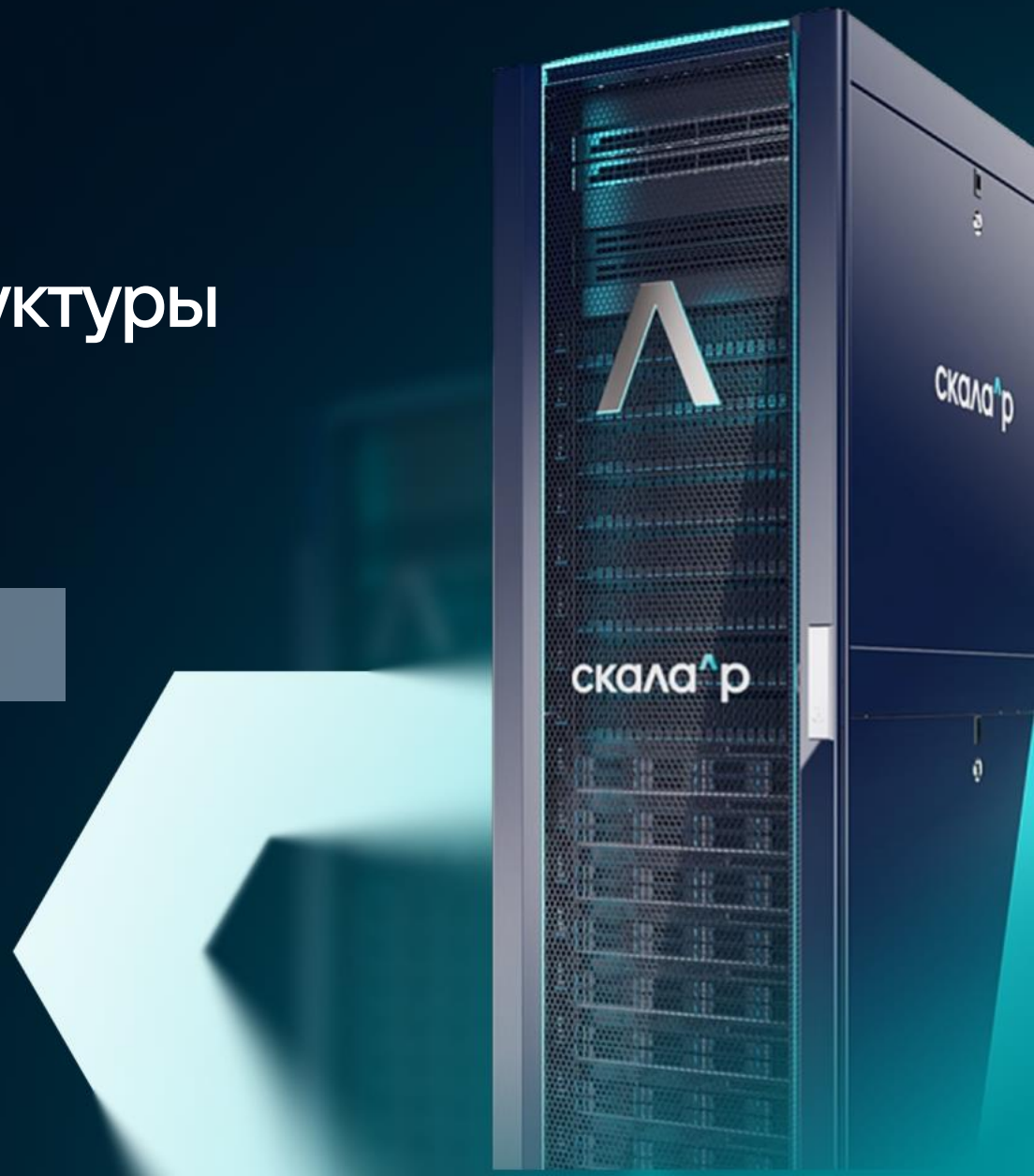
скала^р | ГРУППА RUBYTECH

Построение доверенной инфраструктуры высоконагруженных систем модульными ПАК Скала^р

Скала^р Модуль балансировки нагрузки



скала^р | ГРУППА RUBYTECH

Скала^р — модульная платформа

продукт Группы Rubytech

для построения инфраструктуры высоконагруженных корпоративных и государственных информационных систем

11 лет серийного выпуска

800+ комплексов в промышленной эксплуатации

13 тыс. вычислительных узлов

Продуктовые направления Скала^р

скала^р | ГРУППА RUBYTECH

Динамическая инфраструктура

Машины динамической инфраструктуры Скала^р МДИ

- на основе решений BASIS для создания динамической конвергентной и гиперконвергентной инфраструктуры серверной виртуализации ЦОД и виртуальных рабочих мест пользователей

Инфраструктура ИИ

Машина искусственного интеллекта Скала^р

- на основе оптимизированного программно-аппаратного стека для максимальной производительности при работе с моделями ИИ

Управление данными

Машины баз данных Скала^р МБД.П

- на основе решений Postgres Pro для замены Oracle Exadata в высоконагруженных системах с обеспечением высокой доступности и сохранности критически важных данных

Машины больших данных Скала^р МБД.8

- на основе решений ARENADATA и PICODATA для создания инфраструктуры хранения, преобразования, аналитической, статистической обработки данных, а также распределенных вычислений

Машины хранения данных Скала^р МХД

- на основе технологии объектного хранения S3 для георасспределенных катастрофоустойчивых систем с миллиардами объектов различного типа и обеспечения быстрого доступа к ним
- решения на основе платформы S3 и российского ПО для комплексных задач резервного копирования и восстановления крупных массивов данных со встроенной иерархией хранения и обеспечением высокой доступности копий

Специализированные решения

Машина управления технологическими процессами Скала^р МСП.ТП

- высоконадежная инфраструктура для различных АСУ ТП промышленных предприятий с высокими требованиям к отказоустойчивости и информационной безопасности. Соответствует требованиям ЗОКИИ, в том числе критериям к доверенным ПАК

Машина специализированных банковских систем Скала^р МСП.БС

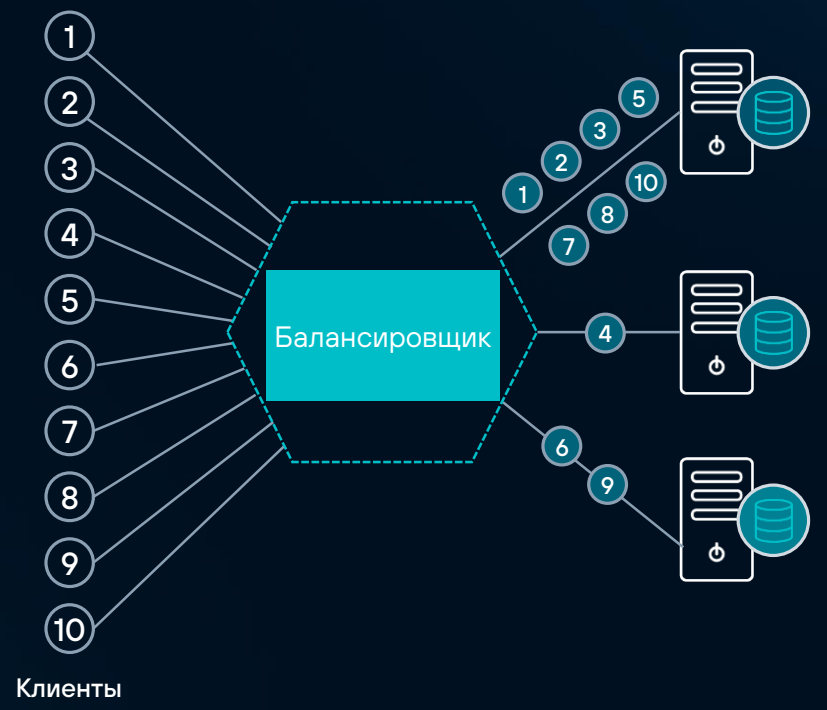
- на платформе Машин Скала^р для задач класса АБС и процессинговых решений с поддержкой высокой транзакционной и аналитической нагрузки, сегментирования баз данных и обеспечения ИБ

В кооперации с технологическими лидерами российского рынка в каждом из сегментов

О технологии балансировки нагрузки

Балансировка нагрузки или выравнивание нагрузки (англ. load balancing) — метод распределения заданий между несколькими сетевыми устройствами (например, серверами) с целью оптимизации использования ресурсов, сокращения времени обслуживания запросов, горизонтального масштабирования кластера (динамическое добавление/удаление устройств), а также обеспечения отказоустойчивости (резервирования). Процедура балансировки осуществляется при помощи целого комплекса алгоритмов и методов.

- **DNS балансировка** (Сетевой уровень) — Обеспечивает балансировку нагрузки и отказоустойчивость в нескольких центрах обработки данных.
- **L4 балансировка** (Транспортный уровень) — Распределение трафика идет на основе информации из протоколов TCP или UDP. Данный метод популярен в сетевых сервисах, где важно учитывать порты и протоколы, но не требуется анализировать содержимое трафика.
- **L7 балансировка** (Прикладной уровень) — Распределение идет
на базе содержимого приложения (HTTP/HTTPS). Идеален для сложных веб-приложений, микросервисных архитектур, API-шлюзов, где нужно учитывать контекст запроса
и принимать решения на основе содержимого HTTP-запросов.



Какие бывают балансировщики нагрузки?

скала^р | ГРУППА RUBYTECH

- **Аппаратные балансировщики**
  Это специализированные устройства, которые используются в центрах обработки данных для управления сетевым трафиком. Они предоставляют высокую производительность, безопасность и масштабируемость.
  Примеры: F5 Networks, Citrix ADC (ранее NetScaler), A10 Networks.

- **Программные балансировщики**
  Это программные решения, которые работают на стандартных серверах заказчика.
  Они гибкие и часто более доступные по сравнению с аппаратными балансировщиками.
  Примеры Open Source решений: HA Proxy, Nginx, Traefik, Apache HTTP Server

Балансировщики — ключевой компонент современных распределённых систем, который помогает им работать эффективно даже во время пиковых нагрузок.

Балансировщики применяются для:

- Повышения надежности системы
- Уменьшения времени ответа на запрос
- Оптимизации использования ресурсов
- Предотвращения перегрузки какого-либо одного ресурса
- Максимизации пропускной способности

Модуль балансировки нагрузки Скала^р

скала^р | ГРУППА RUBYTECH

Модуль балансировки нагрузки Скала^р (ПАК) представляет собой аппаратный отказоустойчивый кластер балансировки нагрузки с использованием OEM компонентов ПО Angie ADC

Основные возможности:

- Функциональность локальной (LTM) и глобальной (GTM на основе DNS) балансировки
- Балансировка нагрузки на уровнях L3/L4/L7 модели OSI
- Поддержка протоколов динамической маршрутизации BGP, OSPF
- Двухузловой кластер Active-Active / Active-Passive
- Система управления и мониторинга
- Управление через API, CLI, GUI

Дополнительные преимущества:

- Единая техническая поддержка от вендора **Скала^р**
- Возможность доработки функционала по требованиям заказчика

Производительность в зависимости от лицензии и конфигурации до 20 Гбит/с или до 50 Гбит/с на узел кластера в составе Модуля

Скала^р Модуль балансировки нагрузки

Функциональность

- **DNS-балансировка** — распределяются запросы, возвращая разные IP-адреса одного доменного имени для разных клиентов. Решения принимаются на основе географического расположения пользователей, загруженности серверов, политик маршрутизации
- **Балансировка нагрузки на уровне L4** — принимаются решения на основе алгоритма балансировки и информации о TCP/UDP-сессиях, включая IP-адреса, порты источника и назначения
- **Балансировка нагрузки на уровне L7** — анализ содержимое HTTP/HTTPS запросов (например, URL, заголовки, куки, методы запроса) и принятие решения о маршрутизации на основе этих данных
- **Удержание клиентских сессий** — возможность определения всех запросов от одного клиента, и направление их на один бэкенд
- **Обработка и ускорение TLS** — Балансировщик может терминировать на себе шифрованные соединения, снимая таким образом нагрузку с серверов
- **Проверка доступности серверов** — используются гибкие механизмы проверки доступности серверов
- **Ролевая модель доступа** — набор полномочий, привязанных к должностям или рабочим задачам
- **Управление и мониторинг** — через API/GUI/CLI

Скала^р Модуль балансировки нагрузки

Поддерживаемые режимы балансировки

- Round Robin — запросы распределяются по серверам последовательно (используется по умолчанию)
- Балансировка на основе веса (weight) — запросы распределяются по серверам с учетом веса каждого сервера
- Балансировка ip-hash — сервер выбирается с помощью хэш-функции на основе IP-адреса клиента, что обеспечивает «sticky session». Гарантирует, что запросы от одного клиента будут попадать на один сервер, если этот сервер доступен
- least_conn — новый запрос направляется на сервер с минимальным количеством активных соединений (наименее загруженный)
- least_bandwidth — балансировка на основе наименьшего потребления полосы пропускания. Периодически вычисляется среднее использование полосы пропускания для каждого апстрим-сервера, используя формулу скользящего среднего для сглаживания колебаний со временем
- least_packets — балансировка на основе наименьшего числа пакетов в единицу времени
- least_time — балансировка на основе наименьшего времени ответа. Вероятность передачи соединения активному серверу обратно пропорциональна среднему времени его ответа
- feedback — механизм балансировки нагрузки по обратной связи (по произвольному параметру из ответа на основной или проверочный запрос)

Скала^р Модуль балансировки нагрузки

Поддерживаемые проверки доступности серверов

Балансировщик **Скала^р** использует пассивные и активные проверки состояния бэкендов. Если бэкенд начинает отвечать с ошибками, то он временно исключается из пула доступных бэкендов.

- Пассивные проверки доступности определяют состояние бэкенд узлов на основе пользовательских ответов. Позволяет мгновенно принимать решение.
- Активные периодические проверки устанавливают доступность бэкенд узлов, в том числе со сложной логикой. Узел проходит проверку, если запрос к нему успешно выполняется с учетом всех заданных параметров.
- Механизмы определения доступности сервера: tcp-check, http-check, icmp-check.
- Slow start позволяет постепенно вводить в строй восстановленный бэкенд узел. Тем самым «непрогретый» бэкенд узел не будет немедленно испытывать критическую нагрузку.

Скала^р Модуль балансировки нагрузки

Распределение нагрузки

Основные методы балансировки можно использовать как по отдельности, так и в различных сочетаниях:

- **Round Robin** — запросы распределяются последовательно по каждому бэкенду
- **Weighted Round Robin** — аналогично, с учётом весов бэкендов
- **Least Connections** — новый запрос отправляется на бэкенд с наименьшим количеством активных соединений
- **Weighted Least Connections** — аналогично, но учитывается вес бэкендов

Весовые коэффициенты позволяют задать долю (вес) запросов, обрабатываемых каждым сервером приложений.

Вес может задаваться/определяться как статически, так и динамически по заданным критериям, с учётом, например:

- Производительности
- Текущей утилизации
- Времени отклика
- По настраиваемой пользователем логике

```
upstream myapp1 {
        server    192.168.56.61 weight=20;
        server    192.168.56.62 weight=30;
        server    192.168.56.63 weight=50;
}

server {
        listen    80;
        server_name        localhost;

        location / {
        proxy_pass         http://myapp1;
        }
}
```

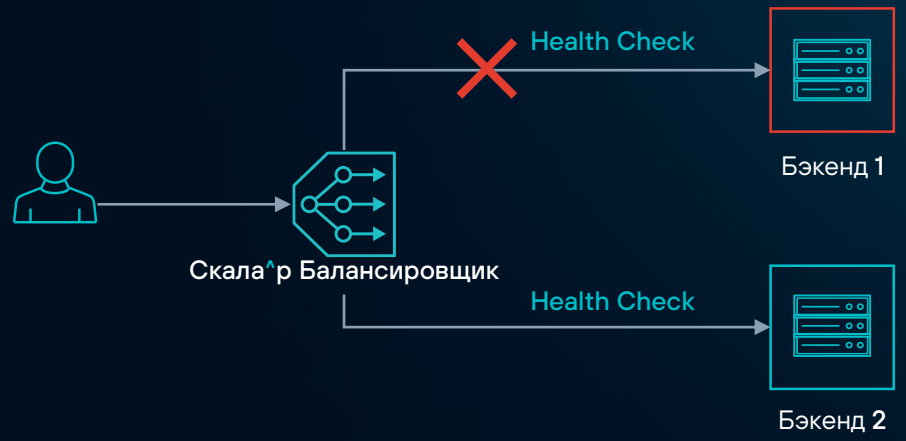
Скала^р Модуль балансировки нагрузки

Отказоустойчивость и проверки

Доступность бэкендов может контролироваться в Балансировщике динамически разными методами, отдельно или в разных сочетаниях, например:

- На уровне сети: ICMP Ping — проверка доступности через стандартный ICMP Echo Request
- На уровне транспорта: TCP Health Check (TCP Ping) — проверка доступности порта приложения
- На уровне сессии и/или приложения: HTTP(S) Health Check — проверка через запрос специального URL, и ожидание корректного кода ответа
- Кастомизированная проверка по пользовательской логике — с помощью различных скриптов
- Пассивный мониторинг — без активного опроса, Балансировщик определяет доступность бэкенда по наблюдениям над успешными/неуспешными реальными соединениями

Конечное решение может быть принято как по отдельным критериям оценки работоспособности, так и по совокупности




```
upstream myapp1 {
        server 192.168.56.61 max_fails=2
fail_timeout=10s;
        server 192.168.56.62 max_fails=2
fail_timeout=10s;
        server 192.168.56.63 max_fails=2
fail_timeout=10s;
}

server {
        listen    80;
        server_name        localhost;

        location / {
        proxy_pass         http://myapp1;
        }
}
```

Скала^р Модуль балансировки нагрузки

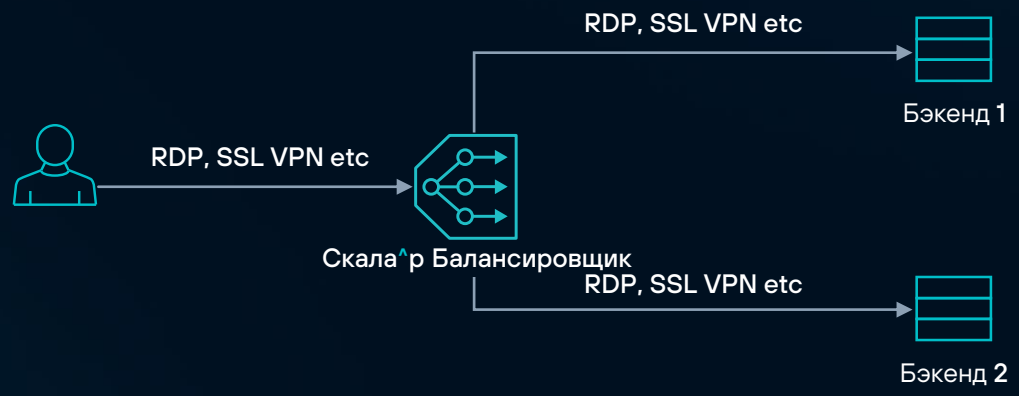
Конвертация протоколов

Балансировщик может в ряде сценариев преобразовывать протоколы с фронтенд представления сервиса на бэкенд серверы.

Например, Балансировщик нагрузки может терминировать на себе **HTTP/HTTPS** (SSL) трафик в различных вариациях и использовать **HTTP без шифрования** при коммуникациях с бэкендами.

Также возможна балансировка трафика с stateful- и stateless-сессиями, по протоколам **TCP/UDP** на уровне L4 без раскрытия, с применением тех же правил и методик, как и для балансировка HTTP.

Например, можно применять Балансировщик для распределения нагрузки на **RDP/VDI** фермы, **SSL VPN** шлюзы.



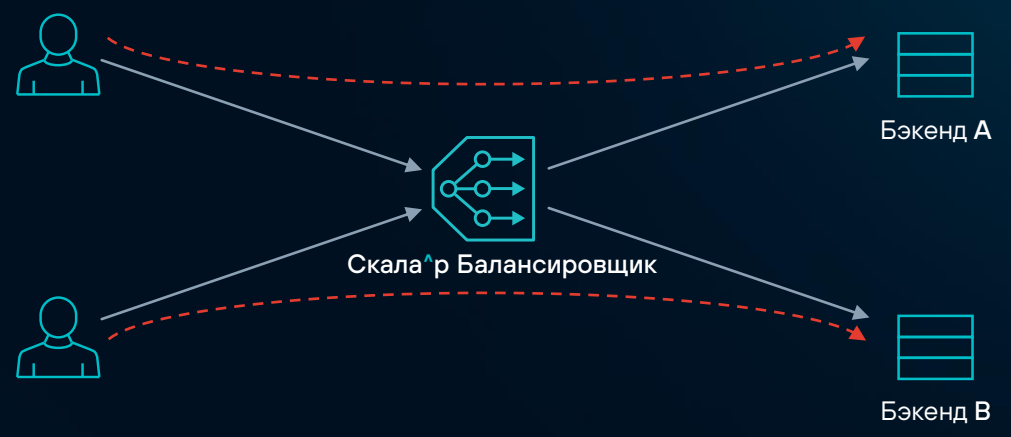
Скала^р Модуль балансировки нагрузки

Персистентность сессий и распределение

Балансировщик может распределять нагрузку с учётом персистентности клиентских сессий — чтобы трафик от одного пользователя терминировался на одном бэкенде до тех пор, пока он доступен.

Это достигается за счёт применения **хэширования** пользовательских идентификаторов (в простейшем случае — по IP) в динамическом режиме.

Кроме этого, типовые сценарии балансировки предполагают **перенаправление определённой доли** клиентского трафика на выделенный отдельный бэкенд (или группу), например, для постепенного релиза новой версии приложения или сервиса на ограниченную аудиторию.



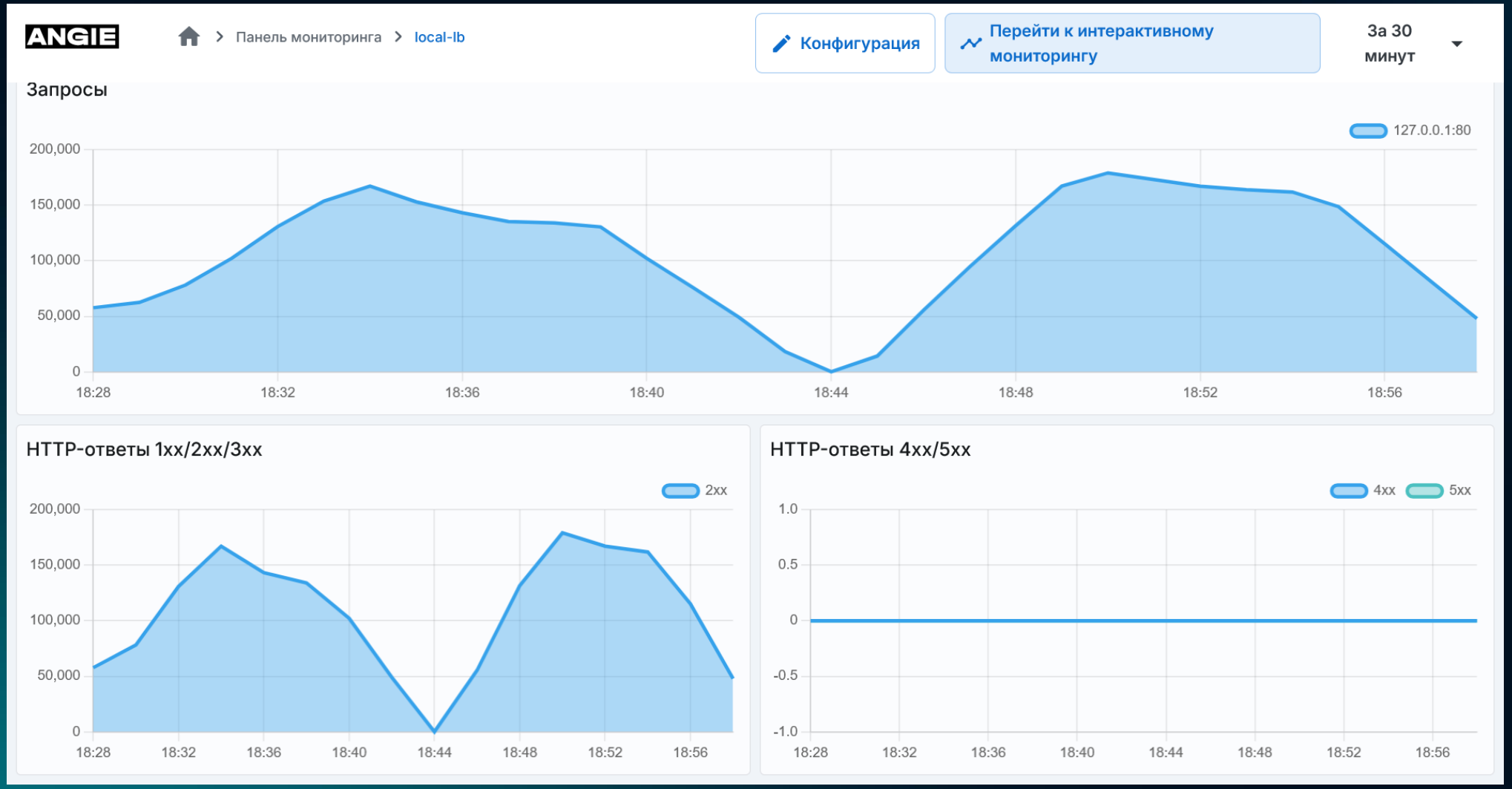

```
upstream myapp1 {
        ip_hash;
        server    192.168.56.61;
        server    192.168.56.62;
}

server {
        listen    80;
        server_name        localhost;

        location / {
        proxy_pass         http://myapp1;
        }
}
```

Скала^р Модуль балансировки нагрузки

Мониторинг и графические панели параметров работы



Скала^р Модуль балансировки нагрузки

скала^р | ГРУППА RUBYTECH

Сравнение с конкурентами (1/2)

№	Параметр	HA Proxy	Nginx	Скала^р	F5
1	Периодическая проверка доступности обслуживающих (бэкенд) серверов	✓		✓	✓
2	Несколько алгоритмов определения доступности сервера: tcp-check, http-check	✓		✓	✓
3	Балансировка HTTP / HTTPS / TCP-запросов между «живыми» серверами	✓	✓	✓	✓
4	Поддержка TLS	✓	✓	✓	✓
5	Возможность закрепления определенных клиентов за конкретными обслуживающими серверами (stick-tables)	✓	✓	✓	✓
6	Репликация sticky-сессий между нодами HA-кластера			2026	✓
7	Поддержка HTTP/2	✓	✓	✓	✓
8	Поддержка HTTP/3	✓	✓	✓	✓
9	Балансировка на основе наименьшего потребления полосы пропускания (least bandwidth)			✓	✓
10	Балансировка на основе наименьшего числа пакетов в единицу времени (least packets)			✓	✓

№	Параметр	HA Proxy	Nginx	Скала^р	F5
11	Балансировка нагрузки на основе кратчайшего ответа (least time)			✓	✓
12	Динамическая балансировка нагрузки на основе произвольной метрики, исходя из значений запросов (feedback)			✓	✓
13	Постепенная подача трафика на проксируемый сервер (slow start)			✓	✓
14	Обновление списка upstream серверов по SRV-записям DNS	✓	✓	✓	✓
15	Конфигурация через API проксируемых серверов без перезапуска приложения			✓	✓
16	Балансировка по произвольному параметру из ответа на основной или проверочный запрос		✓	✓	✓
17	Поддержка балансировки GSLB с учетом активных проверок upstream серверов у каждого балансировщика	✓		✓	✓
18	Поддержка динамической фильтрации трафика через механизм BGP Flowspec			2026	✓
19	Маршрутизация шифрованных соединений на базе технологии SSL preread (балансировка ГОСТ TLS без терминирования TLS)			✓	✓
20	Интеграция с Docker API	✓		✓	✓

Скала^р Модуль балансировки нагрузки

скала^р | ГРУППА RUBYTECH

Сравнение с конкурентами (2/2)

№	Параметр	HA Proxy	Nginx	Скала^р	F5
21	Поддержка управления анонсами протоколов динамической маршрутизации на основе статуса бэкендов и групп бэкендов (RHI)			✓	✓
22	Поддержка протоколов динамической и статической маршрутизации			✓	✓
23	Встроенная поддержка ACME для перевыпуска сертификатов без использования сторонних утилит (таких как certbot)			✓	✓
24	Поддержка ГОСТ TLS на основе КриптоПро			✓	
25	API для интеграции с Prometheus для сбора метрик в реальном времени			✓	✓
26	Визуальный интерфейс управления конфигурацией и мониторинга			✓	✓
27	Ролевая модель доступа к разным конфигурационным объектам и зонам мониторинга			✓	✓
28	Наличие технической поддержки производителем на территории России			✓	
29	Возможность приоритизации дорожной карты с вендором			✓	
30	Присутствие в реестре отечественного ПО			✓	
31	Поддержка развёртывания решения в режиме высокой доступности (High Availability deployment cluster)			✓	✓
32	Поддержка развёртывания решения в режиме высокой производительности (High Performance deployment cluster)			2026	✓
33	Поддержка аппаратного ускорения обработки трафика			2026	✓

№	Параметр	HA Proxy	Nginx	Скала^р	F5
34	GSLB на основе хэша IP адреса назначения/источника, хэш порта/источника			2026	✓
35	GSLB на основе наименьшего числа соединений			2026	✓
36	GSLB на основе кратчайшего ответа			2026	✓
37	GSLB на основе RTT (round-trip time)			2026	✓
38	Возможность аутентификации администратора по TACACS/RADIUS/LDAP			2026	✓
39	Настраиваемая парольная политика локальных пользователей			✓	✓
40	Аутентификация клиентов через балансировщик нагрузки			✓	✓
41	Поддержка SNMPv3 и SNMP Trap			✓	✓
42	Детальное логирование			✓	✓
43	Возможность формирования отчетов			2026	✓
44	Откат настроек на предыдущее состояние			2026	✓
45	Резервное копирование и восстановление полной конфигурации аппаратного балансировщика			✓	✓
46	Наличие подсистемы политик обработки трафика (iRules/AppQOE)			✓*	✓

* F5 iRule = директивы Angie ADC

скала^р

ПАК Скала^р — фундамент для построения доверенной инфраструктуры

www.skala-r.ru
E-mail: info@skala-r.ru

Документ актуализирован
14.10.2026