

????????? KUMA ?

???????????????????? -

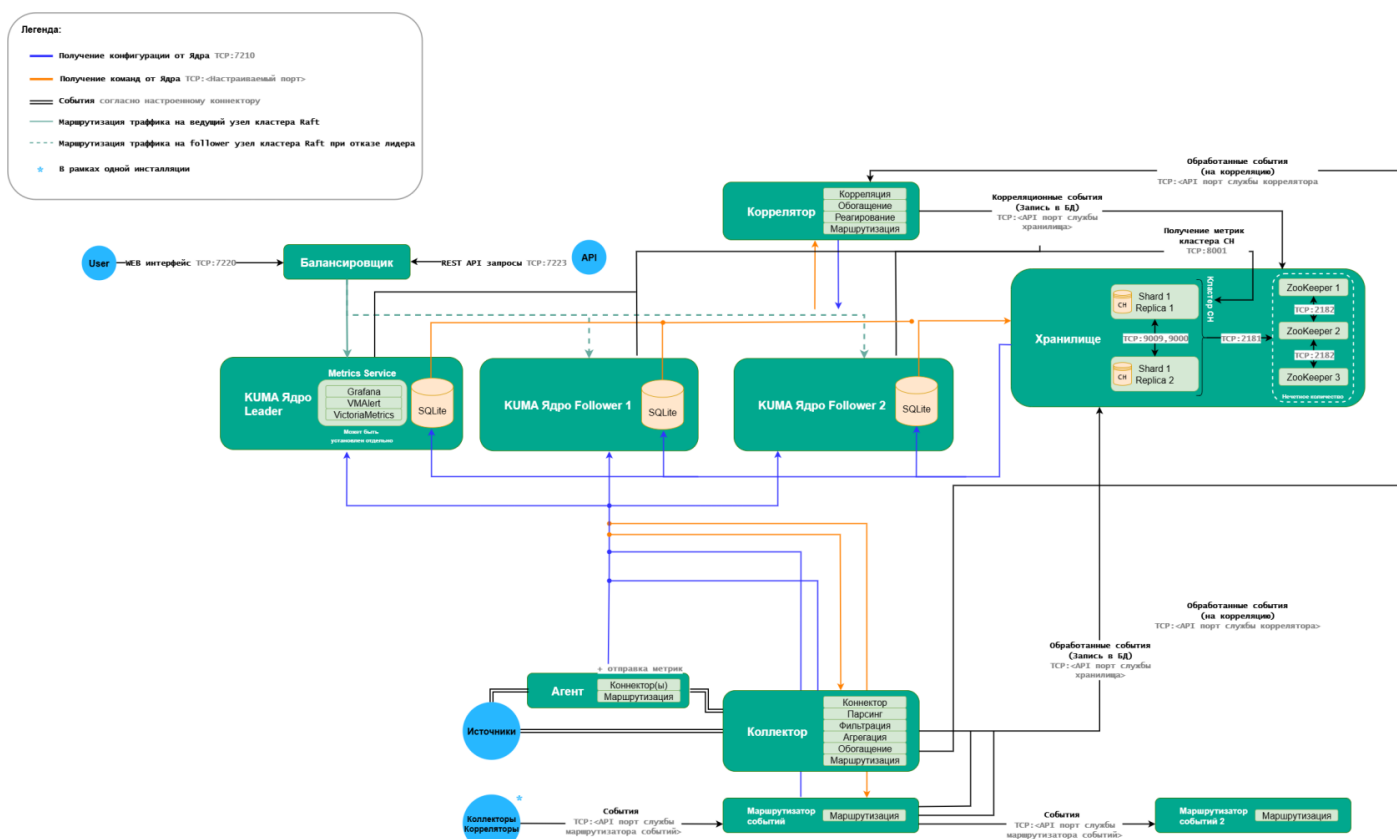
RAFT

Информация, приведенная на данной странице, является разработкой команды pre-sales и/или community KUMA и **НЕ** является официальной рекомендацией вендора.

Официальная документация по данному разделу приведена в [Онлайн-справке](#) на продукт.

Кластер RAFT **поддерживается в KUMA с версии 4.0** и позволяет обеспечить доступность компонентов ядра KUMA, в случае сбоя работы одного из компонентов. Допустимо разворачивать только нечетное количество сервисов Ядра KUMA.

Схема ядра в RAFT (хранилище представлено с двумя репликами, т.е. с копией данных)



??????

В кластере Raft каждый из серверов в каждый момент времени находится в одном из трёх состояний:

- Leader (лидер) – обрабатывает все клиентские запросы, является source of truth всех данных в логе, поддерживает лог фоловеров.
- Follower (фоловер) – пассивный сервер, который только «слушает» новые записи в лог от лидера и редиректит все входящие запросы от клиентов на лидера. По сути, является hot-standby репликой лидера.
- Candidate (кандидат) – специальное состояние сервера, возможное только во время выбора нового лидера.

Во время нормальной работы в кластере только один сервер является лидером, все остальные – его фоловеры. Например, в случае 3 компонентов ядра из строя может выйти только 1 компонент, для иного количества компонентов - пока работает формула $(n+1)/2$. Полезная ссылка для понимания механизма работы: <https://thesecretlivesofdata.com/raft/>

Сетевые задержки не должны превышать 150 мс.

?????????? ???? ?? ????????? ? Raft

Сценарии установки с установкой Ядра в кластере Raft смотреть можно [тут](#)

Данный сценарий реализуется, если KUMA находится в инсталляции, где одно ядро (standalone). Для этого необходимо [подготовить](#) дополнительные целевые машины для установки сервисов, а также на контрольной машинке настроить обмен SSH [ключами](#). После создать копию файла на контрольной машине (с которой предполагается развертывание) **expand.inventory.yml.template** из распакованного архива KUMA

```
cp expand.inventory.yml.template expand.inventory.yml
```

Далее отредактировать файл **expand.inventory.yml** (разворачиваем дополнительные ядра 2 и 3):

```
GNU nano 2.9.8 expand.inventory.yml
kuma:
  vars:
    ansible_connection: ssh
    ansible_user: root
    no_firewall_actions: false
  children:
    kuma_core_peers:
      hosts:
        kuma-core-2.local:
        kuma-core-3.local:
```

В хостах указываем FQDN и проверяем их доступность с контрольной машины, а также на всех компонентах ядра должно быть единое время, настройте на всех машинах NTP.

Далее на контрольной машине с доступом root из папки с распакованным установщиком (скриншот выше) выполните следующую команду для начала установки:

```
./expand.sh expand.inventory.yml
```

В результате выполнения команды на каждой целевой машине, указанной в файле инвентаря expand.inventory.yml, появятся файлы для создания и установки дополнительных сервисов Ядра KUMA, для этого проверьте **наличие директории /opt/kaspersky/kuma/kuma**

В разделе **Ресурсы → Активные сервисы** нажмите **Добавить** и в раскрывающемся списке выберите **Добавить сервис Ядро**.

Ресурсы и сервисы / Сервисы

Сервисы

2

+ Добавить ▾ | Обновить параметры | Перезагрузить

Добавить сервис

Добавить сервис 'Ядро'

Добавить сервис 'Метрики'

	Тип
☐ ☒ Вкл	Ядро
☐ ☒ Вкл	Агент
☐ ☒ Вкл	Ядро
☐ ☒ Вкл	Ядро
☐ ☒ Вкл	Хранилище
☐ ☒ Вкл	Коррелятор
☐ ☒ Вкл	Коллектор
☐ ☒ Вкл	Коллектор
☐ ☒ Вкл	Коллектор
☐ ☒ Вкл	Коллектор
☐ ☒ Вкл	Коллектор

В открывшемся окне Добавить сервис Ядро укажите имя сервиса в поле Имя сервиса Ядро и нажмите Добавить. Добавленный сервис появится в списке Сервисы.

Сервисы

+ Добавить

Обновить параметры

Перезапустить

Перейти к событиям

Смотреть активные листы

Смотреть контекстные таблицы

Поиск

🔍

🔄

⚙️

☐ Статус	Тип	Сервис	Версия	Тенант	Полное доменное имя	IP-адрес	Порт API	Время работы	Создан
☐ Выкл	Ядро	kuma-core-2		Main					08.08.2025 11...
☐ Выкл	Ядро	kuma-core-3		Main					08.08.2025 11...

Установите флажок рядом с добавленным сервисом Ядра KUMA и на панели инструментов нажмите `more_vertical`, в открывшемся меню нажмите Копировать идентификатор. Идентификатор сервиса Ядра KUMA понадобится для установки сервиса на сервере.

Далее на целевой машине необходимо установить службу специальным образом:

```
sudo /opt/kaspersky/kuma/kuma core --raft.join <FQDN хоста из секции kuma_core, где запущен первый сервис Ядра KUMA>:7210 --id <идентификатор сервиса Ядра KUMA, скопированный в веб-интерфейсе> --install
```

Повторите создание клиентской и серверной части сервиса Ядра KUMA для каждого хоста из группы `kuma_core_peers`.

На каждом хосте может быть установлен только один сервис Ядра KUMA.

После успешной установки в веб-интерфейсе KUMA, вы увидите статус сервисов **Вкл**

Статус	Тип	Сервис	Версия	Тенант	Полное доменное имя	IP-адрес	Порт API	Время работы	Создан
Вкл	Ядро	kuma-core-3	4.0.0.389	Main	kuma-core-3.local	10.10.10.12	7210	1 минута 44 секунды	08.08.2025 11...
Вкл	Ядро	kuma-core-2	4.0.0.389	Main	kuma-core-2.sales.lab	10.10.10.14	7210	6 минуты 12 секунды	08.08.2025 11...
Вкл	Ядро	core-1	4.0.0.389	Main	kuma-test.local	10.10.10.11	7210	17 дня 20 часа 50 м...	20.06.2025 17...

Если ведущий узел выходит из строя, сервер будет следовать за новым ведущим, обеспечивая непрерывный доступ к интерфейсу SIEM или API.

Журналы Ядра KUMA хранятся в директории `/opt/kaspersky/kuma/core/<идентификатор сервиса Ядра KUMA>/log/core`.

????????? ?????????????????? - nginx

Установим балансировщик нагрузки nginx для удобства работы с кластером Raft, а именно обращение по единому hostname / IP-адресу.

Когда одна из нод кластера отказывает, балансировщик перенаправляет запросы на активные ноды

Для установки балансировщика поднимем отдельный сервер и на него установим **nginx**, далее необходимо отредактировать файл `/etc/nginx/nginx.conf` и добавить конфиг в конец файла

```
stream {
    upstream raft_backend {
        server kuma-test.local:7220 max_fails=1 fail_timeout=5s;
        server kuma-core-2.local:7220 backup;
        server kuma-core-3.local:7220 backup;
    }

    server {
        listen 443;
        proxy_pass raft_backend;
        ssl_preread on;
    }
}
```

Далее перезапустить сервис `systemctl restart nginx.service`

Для проверки работоспособности отключаем сервис ядра на **kuma-test** командой `systemctl stop kuma-core-<идентификатор-ядра>.service`. В браузере вводим `https://<ip-балансировщика>`, появится форма входа в интерфейс.

Далее нужно перейти в "**Ресурсы**" -> "**Активные сервисы**". Увидим, что сервис ядра отключен, но независимые компоненты имеют Статус: "**Вкл**"

Kaspersky
Unified Monitoring and Analysis Platform

Выбрано тенантов: 1

- Панели мониторинга
- Алерты
- Инциденты
- События
- Активы
- Отчеты
- Ресурсы 1**
- CyberTrace
- Диспетчер задач
- Параметры
- Состояние источников
- Метрики

Administrator

Ресурсы и сервисы / Сервисы 2

Сервисы

+ Добавить
Обновить параметры
Перезапустить
Перейти к событиям
Смотреть активные листы
Смол...

	Статус	Тип	Сервис	Версия	Тенант	Полное доменное имя
☐	Выкл	Агент	Windows WEC	3.4.2.22	Main	winsrv-test-kuma.truecomp...
☐	Выкл ⓘ	Коллектор	Collector/KSMG -6776	4.0.0.389	Main	kuma-test.local
☐	Выкл ⓘ	Коллектор	KSC CEF TCP/5163	4.0.0.389	Main	kuma-test.local
☐	Выкл ⓘ	Коллектор	Linux Auditd rsyslog TCP/5152	4.0.0.389	Main	kuma-test.local
☐	Выкл ⓘ	Коллектор	Linux KESL/5171	4.0.0.389	Main	kuma-test.local
☐	Выкл ⓘ	Коллектор	Windows WEC TCP/5151	4.0.0.389	Main	kuma-test.local
☐	Вкл	Коллектор	[OOTB] CEF	4.0.0.389	Main	kuma-test.local
☐	Вкл	Коллектор	[OOTB] KSC	4.0.0.389	Main	kuma-test.local
☐	Вкл	Коллектор	[OOTB] Syslog	4.0.0.389	Main	kuma-test.local
☐	Вкл	Коллектор	[OOTB] Syslog-CEF	4.0.0.389	Main	kuma-test.local
☐	Выкл ⓘ	Коллектор	sqlite-KSC/ sql	4.0.0.389	Main	kuma-test.local
☐	Выкл ⓘ	Коллектор	test-events/1111	4.0.0.389	Main	kuma-test.local
☐	Выкл ⓘ	Коллектор	test-test/7777	4.0.0.389	Main	kuma-test.local
☐	Выкл ⓘ	Коллектор	События windows via KES/51535	4.0.0.389	Main	kuma-test.local
☐	Выкл ⓘ	Ядро	core-1	4.0.0.389	Main	kuma-test.local
☐	Вкл	Ядро	kuma-core-2	4.0.0.389	Main	kuma-core-2.sales.lab
☐	Вкл	Ядро	kuma-core-3	4.0.0.389	Main	kuma-core-3.local
☐	Вкл	Коррелятор	[OOTB] Correlator	4.0.0.389	Main	kuma-test.local
☐	Вкл	Метрики	metrics	4.0.0.389	Main	kuma-test.local

Не забудьте запустить службу отключенного компонента, ядра `systemctl start kuma-core-
<идентификатор-ядра>.service`

В данном случае в кластере всего три ноды, и больше одного выведенного из строя компонента ядра нельзя себе позволить, поскольку с менее чем 2 рабочими нодами теряется доступ к системе и перестает работать кластер

В основу алгоритма Raft заложено понятие кворум - минимальное количество нод, необходимое для принятия решений, или большинство. При большинстве рабочих нод в кластере можно проводить голосование, выбирать лидера и принимать изменения

В нашем случае, если останется 1 живая нода - не сработает. Нода не образует кворум, а если захочет стать лидером, это никто не подтвердит.

?????????? ?????? ?????????? ?????:

Если вы столкнулись с ошибками после установки балансировщика, примеры:

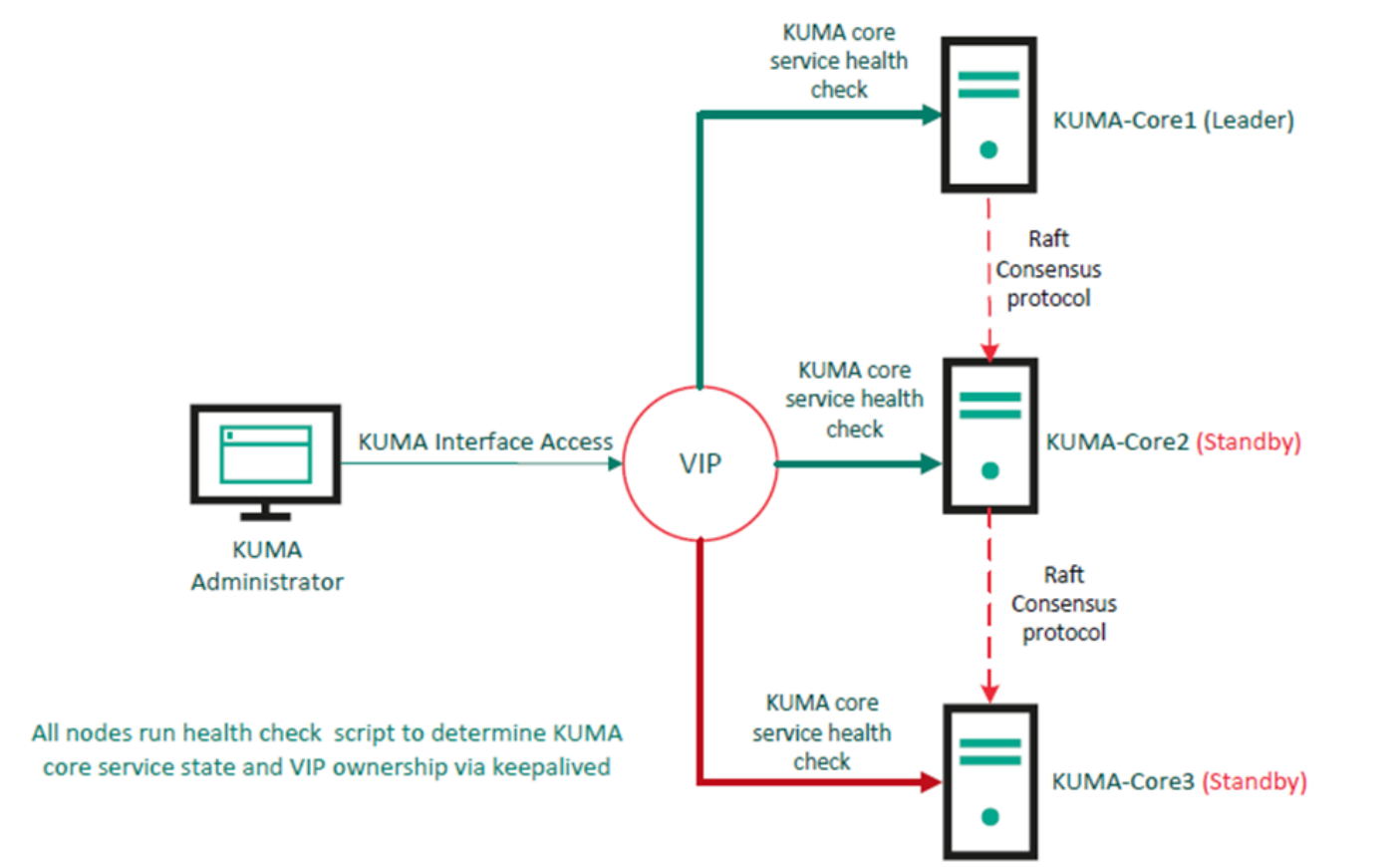
```
request dropped as the cluster is not ready
failed to lookup: can't serve requests, not enough healthy nodes
```

```
request dropped as the cluster is not ready
failed to lookup: can't serve requests, not enough healthy nodes
```

Убедитесь, что у вас достаточное количество рабочих нод в кластере для его работы.

???????????? ???? - ????? keepalived

В основу данного способа заложена интеграция виртуального IP-адреса (VIP) с использованием функции `keepalived` для предоставления единой точки доступа. Такая конфигурация обеспечивает автоматическое переключение между узлами при отказе.
 ?????? ????????????? ?????? ??? ????????????? ?????????? ????????????????????? ?????? ??????. Все узлы обслуживают веб-интерфейс через собственные полные доменные имена (FQDN), но VIP обеспечивает унифицированный и стабильный доступ, обеспечивая доступность без вмешательства пользователя.



Детализация демонстрационных машин

Hostname	IP address	OS	Role
ksiem.demo.lab	192.168.100.200	N/A	VIP
ksiem-core1.demo.lab	192.168.100.201	Oracle 9.2	Core1-Leader, Storage, collector, Correlator
ksiem-core2.demo.lab	192.168.100.202	Oracle 9.2	Core2-Follower, Storage replica2
ksiem-core3.demo.lab	192.168.100.203	Oracle 9.2	Core3- <u>quorum</u> , Storage keeper3

Мы также рекомендуем предоставлять отдельный сервера для каждого компонента.

После подготовки файла `distributed.inventory.yml`, необходимо сконфигурировать VIP для всех нод

Установка Keepalived на все основные узлы: `dnf install -y keepalived`

Для каждого узла требуется собственный скрипт для мониторинга локальной службы KUMA Core. Сохраните скрипт как `/etc/keepalived/check_kuma.sh`.

Пример для Core1 (ksiem-core1):

```
#!/bin/bash
if systemctl is-active --quiet kuma-core-000000000-0000-0000-0000-000000000000; then
    exit 0
else
    exit 1
fi
```

Настройте имя службы для каждого узла (каждый имеет уникальное имя). Вы можете найти основную службу KUMA, выполнив команду на каждом узле: `systemctl list-units --type=service | grep kuma-core`

Сделайте скрипт исполняемым: `chmod +x /etc/keepalived/check_kuma.sh`

Далее настройте keepalived. Создайте или отредактируйте файл `/etc/keepalived/keepalived.conf` на каждом узле. Ниже представлена ??адаптированная конфигурация.

Core -1

```
vrrp_script chk_kuma {
    script "/etc/keepalived/check_kuma.sh"
    interval 2
    timeout 3
    fall 2
    rise 3
}
```

```

vrrp_instance VI_1 {
    state MASTER
    interface ens34          #необходимо поменять в соответствии со своей конфигурацией
    virtual_router_id 51
    priority 110
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass secretpass
    }
    virtual_ipaddress {
        192.168.100.200
    }
    track_script {
        chk_kuma
    }
}

```

Core -2

```

vrrp_script chk_kuma {
    script "/etc/keepalived/check_kuma.sh"
    interval 2
    timeout 3
    fall 2
    rise 3
}

vrrp_instance VI_1 {
    state BACKUP
    interface ens34          #поменяйте в соответствии со своими параметрами
    virtual_router_id 51
    priority 100
    advert_int 1
    authentication {
        auth_type PASS
        auth_pass secretpass
    }
    virtual_ipaddress {
        192.168.100.200
    }
}

```

```
track_script {  
    chk_kuma  
}  
}
```

Core - 3

```
vrrp_script chk_kuma {  
    script "/etc/keepalived/check_kuma.sh"  
    interval 2  
    timeout 3  
    fall 2  
    rise 3  
}  
vrrp_instance VI_1 {  
    state BACKUP  
    interface ens34          #поменяйте в соответствии со своими сетевыми параметрами  
    virtual_router_id 51  
    priority 90  
    advert_int 1  
    authentication {  
        auth_type PASS  
        auth_pass secretpass  
    }  
    virtual_ipaddress {  
        192.168.100.200  
    }  
    track_script {  
        chk_kuma  
    }  
}
```

Далее выполните команды

```
systemctl enable keepalived  
systemctl start keepalived
```

Зайдите в веб-интерфейс KUMA через VIP FKDN (ksiem.demo.lab) активируйте KUMA и убедитесь, что все ноды сервисов установлены и их статус "зеленый"

Status	Kind	Service	Version	Tenant	FQDN	IP address	API port	Uptime	Created
On	Core	core-1	4.0.0.389	Main	ksiem-core1.demo.lab	192.168.100.201	7210	58 minutes 58 seco...	2025-07-18 14:5...
On	Core	core-ksiem-core2.demo.lab	4.0.0.389	Main	ksiem-core2.demo.lab	192.168.100.202	7210	57 minutes 35 seco...	2025-07-18 14:5...
On	Core	core-ksiem-core3.demo.lab	4.0.0.389	Main	ksiem-core3.demo.lab	192.168.100.203	7210	56 minutes 11 secon...	2025-07-18 14:5...
On	Metrics	metrics	4.0.0.389	Main	ksiem-core1.demo.lab	192.168.100.201	7227	55 minutes 58 seco...	2025-07-18 14:5...

?????????? ??????????

?????????? 1

Когда устанавливается сервис хранилища или другие ноды хранилища, необходимо указать все FQDN ядер в команде, ниже представлен пример того, как установить хранилище в Raft-е (2 replica и 3 keeper)

```

last login: Fri Jul 18 14:56:13 2025 from 192.168.100.201
[root@ksiem-core3 ~]# /opt/kaspersky/kuma/kuma collector --core https://ksiem-core1.demo.lab:7210,https://ksiem-core2.demo.lab:7210,https://ksiem-core3.demo.lab:7210 --id 96503141-925c-4ef7-8a31-ec517b339b9e ^C-install
[root@ksiem-core3 ~]# exit
logout
Connection to 192.168.100.203 closed.
[root@ksiem-core1 ~]# /opt/kaspersky/kuma/kuma storage --core https://ksiem-core1.demo.lab:7210,https://ksiem-core2.demo.lab:7210,https://ksiem-core3.demo.lab:7210 --id 63f6dd19-86c0-416c-8967-1d274d3dc7 --install
created symlink /etc/systemd/system/multi-user.target.wants/kuma-storage-63f6dd19-86c0-416c-8967-1d274d3dc7.service → /usr/lib/systemd/system/kuma-storage-63f6dd19-86c0-416c-8967-1d274d3dc7.service.
[root@ksiem-core1 ~]# ssh root@192.168.100.202
Activate the web console with: systemctl enable --now cockpit.socket

last login: Fri Jul 18 14:56:13 2025 from 192.168.100.201
[root@ksiem-core2 ~]# /opt/kaspersky/kuma/kuma storage --core https://ksiem-core1.demo.lab:7210,https://ksiem-core2.demo.lab:7210,https://ksiem-core3.demo.lab:7210 --id 8e976148-a498-4ac8-98b4-16f9d39e8af --install
created symlink /etc/systemd/system/multi-user.target.wants/kuma-storage-8e976148-a498-4ac8-98b4-16f9d39e8af.service → /usr/lib/systemd/system/kuma-storage-8e976148-a498-4ac8-98b4-16f9d39e8af.service.
[root@ksiem-core2 ~]# exit
logout
Connection to 192.168.100.202 closed.
[root@ksiem-core1 ~]# ssh root@192.168.100.203
Activate the web console with: systemctl enable --now cockpit.socket

last login: Fri Jul 18 16:08:04 2025 from 192.168.100.201
[root@ksiem-core3 ~]# /opt/kaspersky/kuma/kuma storage --core https://ksiem-core1.demo.lab:7210,https://ksiem-core2.demo.lab:7210,https://ksiem-core3.demo.lab:7210 --id 570192fe-c5dd-4b2a-8bdf-d6045d6e8743 --install
created symlink /etc/systemd/system/multi-user.target.wants/kuma-storage-570192fe-c5dd-4b2a-8bdf-d6045d6e8743.service → /usr/lib/systemd/system/kuma-storage-570192fe-c5dd-4b2a-8bdf-d6045d6e8743.service.
[root@ksiem-core3 ~]#

```

?????????? 2. ??? ??????????? ? ????????????? ?????? ??????????? ? ????? ???????????

Когда вы создаете коллектор или коррелятор, все 3 ноды автоматически добавятся в скрипт, запустите этот скрипт на сервере (коллектора или коррелятора), в этом случае, если core1 упадет, то вы все равно будете иметь доступ к настройкам коррелятора или коллектора

Connect event sources

Transport

Event parsing

Event filtering

Event aggregation

Event enrichment

Routing

Setup validation

Setup validation

Collector configuring is complete and the service is created in KUMA. For details see [Online Help](#).

To start receiving events, you must install this service on the server dedicated for the collector (see example of the install command below). Make sure network access and ports were properly configured. For details see [Online Help](#).

Services using this collector

Kind	Name
collector	KSC

Save and restart services

Save and update service configurations

Recommended command for collector installation

```
/opt/kaspersky/kuma/kuma collector --core https://ksiem-core1.demo.lab:7210,https://ksiem-core2.demo.lab:7210,https://ksiem-core3.demo.lab:7210 --id 874c35a3-4e64-4e0c-b04c-17065d4e8dc2 -  
-api.port 7231 --install
```



+ Add	Update configuration	Restart	Go to events	Go to active lists	Go to context tables	Go to partitions	Download	Copy ID	Search...	⌵	⌵	⌵
Status	Kind	Service	Version	Tenant	FQDN	IP address	API port	Uptime	Created			
☐ On	Collector	KSC	4.0.0.389	Main	ksiem-core1.demo.lab	192.168.100.201	7231	1 minute 37 seconds	2025-07-18 16:21			
☐ On	Core	core-1	4.0.0.389	Main	ksiem-core1.demo.lab	192.168.100.201	7210	17 minutes 28 secon...	2025-07-18 14:5...			
☐ On	Core	core-ksiem-core2.demo.lab	4.0.0.389	Main	ksiem-core2.demo.lab	192.168.100.202	7210	1 hour 29 minutes 47...	2025-07-18 14:5...			
☐ On	Core	core-ksiem-core3.demo.lab	4.0.0.389	Main	ksiem-core3.demo.lab	192.168.100.203	7210	1 hour 28 minutes 2...	2025-07-18 14:5...			
☐ On	Metrics	metrics	4.0.0.389	Main	ksiem-core1.demo.lab	192.168.100.201	7227	1 hour 28 minutes 10...	2025-07-18 14:5...			
☐ On	Storage	Storage	4.0.0.389	Main	ksiem-core1.demo.lab	192.168.100.201	7221	15 minutes 8 seconds	2025-07-18 16:0...			
☐ On	Storage	Storage	4.0.0.389	Main	ksiem-core2.demo.lab	192.168.100.202	7221	13 minutes 59 secon...	2025-07-18 16:0...			
☐ On	Storage	Storage	4.0.0.389	Main	ksiem-core3.demo.lab	192.168.100.203	7221	12 minutes 46 secon...	2025-07-18 16:10...			

Revision #28

Created 2025-08-04 11:44:47 UTC by Ierat

Updated 2026-04-28 10:32:51 UTC by Boris RZR