Телеметрия баз данных

Телеметрия базы данных предоставляет общую информацию о настроенных движках секретов и базах данных.

1. Метрики базы данных секретов

Метрики, связанные с настроенными механизмами секретов, включая метрики, специфичные для базы данных, для каждого именованного механизма секретов. Например, если вы включили механизм секретов PostgreSQL под названием `postgresql-prod`, связанная с ним метрика `CreateUser.error` будет `database.postgresql-prod.CreateUser.error`.

Название метрики	Тип метрики	Единицы измерения	Описание	Примечание
database.Close	сводная	мс	Время, необходимое для закрытия хранилища секретов баз данных (по всем хранилищам секретов баз данных)	---
database.Close.error	счетчик	число	Количество ошибок, возникших при закрытии соединений с базой данных во всех хранилищах секретов баз данных	---
database.NewUser	сводная	мс	Время, необходимое для создания пользователя во всех хранилищах секретов баз данных	---

Название метрики	Тип метрики	Единицы измерения	Описание	Примечание
database.New-User.error	счетчик	число	Количество ошибок, возникших при создании пользователей во всех хранилищах секретов баз данных	---
database.Initialize	сводная	мс	Время, необходимое для инициализации хранилища секретов баз данных (по всем хранилищам секретов баз данных)	---
database.Initialize.error	счетчик	число	Количество ошибок, возникших при инициализации базы данных во всех хранилищах секретов баз данных.	---
database.{NAME}.Close	сводная	мс	Время, необходимое для закрытия хранилища секретов базы данных с именем {NAME}	---
database.{NAME}.Close.error	счетчик	число	Количество ошибок, возникших при закрытии соединений с базой данных в хранилище секретов базы данных с именем {NAME}	---

Название метрики	Тип метрики	Единицы измерения	Описание	Примечание
database.{NAME}.NewUser	сводная	мс	Время, необходимое для создания пользователя в хранилище секретов базы данных с именем {NAME}	---
database.{NAME}.NewUser.error	счетчик	число	Количество ошибок, возникших при создании пользователей в хранилище секретов базы данных с именем {NAME}	---
database.{NAME}.Initialize	сводная	мс	Время, необходимое для инициализации хранилища секретов базы данных для базы данных с именем {NAME}	---
database.{NAME}.Initialize.error	счетчик	число	Количество ошибок, возникших при инициализации базы данных в хранилище секретов с именем {NAME}	---
database.{NAME}.UpdateUser	сводная	мс	Время, необходимое для обновления пользователя в хранилище секретов базы данных с именем {NAME}	---

Название метрики	Тип метрики	Единицы измерения	Описание	Примечание
database.{NAME}.Update-User.error	database.{NAME}.Update-User.error	число	Количество ошибок, возникших при обновлении пользователей в хранилище секретов базы данных с именем {NAME}	---
database.{NAME}.DeleteUser	сводная	мс	Время, необходимое для аннулирования пользователя в хранилище секретов базы данных с именем {NAME}	---
database.{NAME}.Delete-User.error	счетчик	число	Количество ошибок, возникших при аннулировании пользователей в хранилище секретов базы данных с именем {NAME}	---
database.Update-User	сводная	сводная	Время, необходимое для обновления пользователя во всех хранилищах секретов баз данных	---
database.Update-User.error	счетчик	число	Количество ошибок, возникших при обновлении пользователей во всех хранилищах секретов баз данных	---

Название метрики	Тип метрики	Единицы измерения	Описание	Примечание
database.DeleteUser	сводная	сводная	Время, необходимое для аннулирования пользователя во всех хранилищах секретов баз данных	---
database.Delete-User.error	счетчик	число	Количество ошибок, возникших при аннулировании пользователей во всех хранилищах секретов баз данных	---

2. База данных Cockroach

Далее в таблице представлены метрики, связанные с бэкендом хранения базы данных Cockroach:

Название метрики	Тип метрики	Единицы измерения	Описание	Примечание
vault.cockroachd-b.delete	сводная	сводная	Время, необходимое для завершения операции DELETE в бэкенде хранения CockroachDB	---
vault.cockroachd-b.get	сводная	мс	Время, необходимое для завершения операции GET в бэкенде хранения CockroachDB	---
vault.cockroachd-b.list	сводная	мс	Время, необходимое для завершения операции LIST в бэкенде хранения CockroachDB	---

Название метрики	Тип метрики	Единицы измерения	Описание	Примечание
vault.cockroachd-b.put	сводная	мс	Время, необходимое для завершения операции PUT в бэкенде хранения CockroachDB	---

3. Базы данных Couch

Метрики, связанные с бэкендом хранения базы данных Couch представлены далее в таблице:

Название метрики	Тип метрики	Единицы измерения	Описание	Примечание
vault.couchdb.delete	сводная	мс	Время, необходимое для выполнения операции DELETE в бэкенде хранения CouchDB	---
vault.couchdb.get	сводная	мс	Время, необходимое для выполнения операции GET в бэкенде хранения CouchDB	---
vault.couchdb.list	сводная	мс	Время, необходимое для выполнения операции LIST в бэкенде хранения CouchDB	---
vault.couchdb.put	сводная	мс	Время, необходимое для выполнения операции PUT в бэкенде хранения CouchDB	---

4. База данных Dynamo

Далее а таблице представлены метрики, связанные с бэкендом хранения базы данных Dynamo:

Название метрики	Тип метрики	Единицы измерения	Описание	Примечание
vault.dynamod-b.delete	сводная	мс	Время, необходимое для выполнения операции DELETE в бэкенде хранения DynamoDB	---
vault.dynamodb.get	сводная	мс	Время, необходимое для выполнения операции GET в бэкенде хранения DynamoDB	---
vault.dynamodb.list	сводная	мс	Время, необходимое для выполнения операции LIST в бэкенде хранения DynamoDB	---
vault.dynamodb.put	сводная	мс	Время, необходимое для выполнения операции PUT в бэкенде хранения DynamoDB	---

5. Облако Google - Spanner

Далее а таблице представлены метрики, связанные с бэкендом хранения Spanner.

Название метрики	Тип метрики	Единицы измерения	Описание	Примечание
vault.spanner.delete	сводная	мс	Время, необходимое для выполнения операции DELETE в бэкенде хранения Google Cloud Spanner	

Название метрики	Тип метрики	Единицы измерения	Описание	Примечание
vault.spanner.get	сводная	мс	Время, необходимое для выполнения операции GET в бэкенде хранения Google Cloud Spanner	---
vault.spanner.list	сводная	мс	Время, необходимое для выполнения операции LIST в бэкенде хранения Google Cloud Spanner	---
vault.spanner.lock.lock	сводная	мс	Время, необходимое для выполнения операции LOCK в бэкенде хранения Google Cloud Spanner в режиме высокой доступности	---
vault.spanner.lock.unlock	сводная	мс	Время, необходимое для выполнения операции UNLOCK в бэкенде хранения Google Cloud Spanner в режиме высокой доступности	---
vault.spanner.lock.value	сводная	мс	Время, необходимое для выполнения операции VALUE в бэкенде хранения Google Cloud Spanner в режиме высокой доступности	---

Название метрики	Тип метрики	Единицы измерения	Описание	Примечание
vault.spanner.put	сводная	мс	Время, необходимое для выполнения операции PUT в бэкенде хранения Google Cloud Spanner	---

6. Microsoft SQL Server (MSSQL)

Далее в таблице представлены метрики, связанные с бэкендом хранения SQL Server.

Название метрики	Тип метрики	Единицы измерения	Описание	Примечание
vault.mssql.delete	сводная	мс	Время, необходимое для выполнения операции DELETE в бэкенде хранения Microsoft SQL Server	---
vault.mssql.get	сводная	мс	Время, необходимое для выполнения операции GET в бэкенде хранения Microsoft SQL Server	---
vault.mssql.list	сводная	мс	Время, необходимое для выполнения операции LIST в бэкенде хранения Microsoft SQL Server	---
vault.mssql.put	сводная	мс	Время, необходимое для выполнения операции PUT в бэкенде хранения Microsoft SQL Server	---

7. MySQL

Далее в таблице представлены метрики, связанные с бэкендом хранения MySQL.

Название метрики	Тип метрики	Единицы измерения	Описание	Примечание
vault.mysql.delete	сводная	мс	Время, необходимое для выполнения операции DELETE в бэкенде хранения MySQL	---
vault.mysql.get	сводная	мс	Время, необходимое для выполнения операции GET в бэкенде хранения MySQL	---
vault.mysql.list	сводная	мс	Время, необходимое для выполнения операции LIST в бэкенде хранения MySQL	---
vault.mysql.put	сводная	мс	Время, необходимое для выполнения операции PUT в бэкенде хранения MySQL	---

8. PostgreSQL

Далее в таблице представлены метрики, связанные с бэкендом хранения PostgreSQL:

Название метрики	Тип метрики	Единицы измерения	Описание	Примечание
vault.postgres.delete	сводная	мс	Время, необходимое для выполнения операции DELETE в бэкенде хранения PostgreSQL	---

Название метрики	Тип метрики	Единицы измерения	Описание	Примечание
vault.postgres.get	сводная	мс	Время, необходимое для выполнения операции GET в бэкенде хранения PostgreSQL	---
vault.postgres.list	сводная	мс	Время, необходимое для выполнения операции LIST в бэкенде хранения PostgreSQL	---
vault.postgres.put	сводная	сводная	Время, необходимое для выполнения операции PUT в бэкенде хранения PostgreSQL	---

Включение сбора телеметрии StarVault

Далее описан процесс сбора данных телеметрии из установленного StarVault.

1. Перед началом

Перед началом работы убедитесь, что выполнены требования:

- должен быть установлен и запущен StarVault.
- должен быть доступ к файлу [конфигурации StarVault].

2. Шаг 1: Выбор агента агрегации

StarVault использует пакет **go-metrics** для экспорта телеметрии и поддерживает следующие агенты агрегации для мониторинга временных рядов:

Префикс конфигурации	Название	Компания
`circonus`	Circonus	Circonus
`dogstatsd`	DogStatsD	Datadog
`prometheus`	Prometheus	Prometheus / Open source
`stackdriver`	Cloud Operations	Google
`statsd`	Statsd	Open source
`statsite`	Statsite	Open source

3. Шаг 2: Включение хотя бы одного устройства аудита

Чтобы включить метрики, связанные с аудитом, вы должны активировать аудит хотя бы на одном устройстве с помощью команды `StarVault audit enable`. Например, чтобы включить аудит для файлового устройства и сохранять логи в `/var/log/starvault_audit.log`:

```bash
starvault audit enable file file_path=/var/log/starvault_audit.log
```

4. Шаг 3: Настройка сбора телеметрии

Чтобы настроить сбор телеметрии, обновите раздел телеметрии в конфигурации StarVault с вашими предпочтениями по сбору и деталями агента агрегации.

Например, следующий раздел `telemetry` настраивает StarVault с использованием стандартных настроек телеметрии и подключает его к агенту Statsite, работающему на порту по умолчанию в корпоративной интрасети по адресу `mycompany.statsite`:

```
telemetry {
  usage_gauge_period = "10m"
  maximum_gauge_cardinality = 500
  disable_hostname = false
  enable_hostname_label = false
  lease_metrics_epsilon = "1h"
  num_lease_metrics_buckets = 168
  add_lease_metrics_namespace_labels = false
  filter_default = true

  statsite_address = "mycompany.statsite:8125"
}
```

Многие решения для работы с метриками взимают плату за каждую метрику. Вы можете установить параметр filter_default в значение false и использовать параметр prefix_filter для включения и исключения определённых значений на основе названия метрики, чтобы избежать оплаты за неактуальную информацию.

Например, чтобы ограничить вашу телеметрию основными метриками токенов плюс количеством аренды, установленными на истечение:

```
telemetry {
  filter_default = false
  prefix_filter = ["+vault.token", "-vault.expire", "+vault.expire.num_leases"]
}
```

5. Шаг 4: Выбор решения для сбора/агрегации метрик

Вам необходимо сохранить или переслать ваши данные телеметрии в отдельное решение для сбора/агрегации метрик, анализа и оповещений. Какое решение вам нужно, зависит от набора функций, предоставляемых вашим агентом агрегации, и поддержки протоколов вашей платформой для отчётности.

Популярные решения сбора/агрегации метрик, совместимые с StarVault:

- Grafana
- Graphite
- InfluxData: Telegraf
- InfluxData: InfluxDB
- InfluxData: Chronograf
- InfluxData: Kapacitor
- Splunk

6. Следующие шаги

- Ознакомьтесь с руководством по ключевым метрикам для стандартных проверок состояния, чтобы определить, какие метрики вы, возможно, захотите начать отслеживать немедленно.
- Ознакомьтесь с полным списком доступных параметров телеметрии.
- Ознакомьтесь с руководством по мониторингу телеметрии и данных журналов аудита для общего руководства по мониторингу и шагов по настройке телеметрии StarVault для Splunk с использованием Telegraf и Fluentd.
- Ознакомьтесь с руководством по мониторингу телеметрии с использованием Prometheus и Grafana, чтобы настроить телеметрию StarVault для Prometheus и Grafana.

Ключевые метрики для общей проверки работоспособности

В данной статье описаны общие схемы мониторинга StarVault. Представление о работе и использовании кластера StarVault помогает в следующих аспектах:

- реагирование на инцеденты
- понимание рабочих нагрузок
- понимание сценариев использования

Этот документ состоит из **шести разделов**, посвященных метрикам:

1. **Основные метрики**: раздел с описанием базовых внутренних метрик, которые необходимо отслеживать для обеспечения работоспособности кластера StarVault
2. **Метрики использования**: раздел охватывает метрики, которые помогают подсчитать активных и исторических клиентов StarVault.
3. **Бэкэнд хранилища**: раздел, в котором выделены метрики, которые необходимо отслеживать, чтобы понимать инфраструктуру хранения, используемую кластером StarVault. Данные метрики помогают убедиться, что хранилище работает исправно.
4. **Аудит**: раздел, в котором описаны метрики аудита. Данные метрики позволяют настроить мониторинг, который помогает соответствовать нормативным требованиям.
5. **Ресурсы**: раздел, в котором описаны метрики ресурсов. Они позволяют отслеживать такие показатели, как процессор, сеть и другие ресурсы, используемые StarVault на хосте.
6. **Репликация**: в разделе описаны метрики, которые можно использовать, чтобы убедиться в том, что StarVault реплицирует данные исправно.

1. Основные метрики

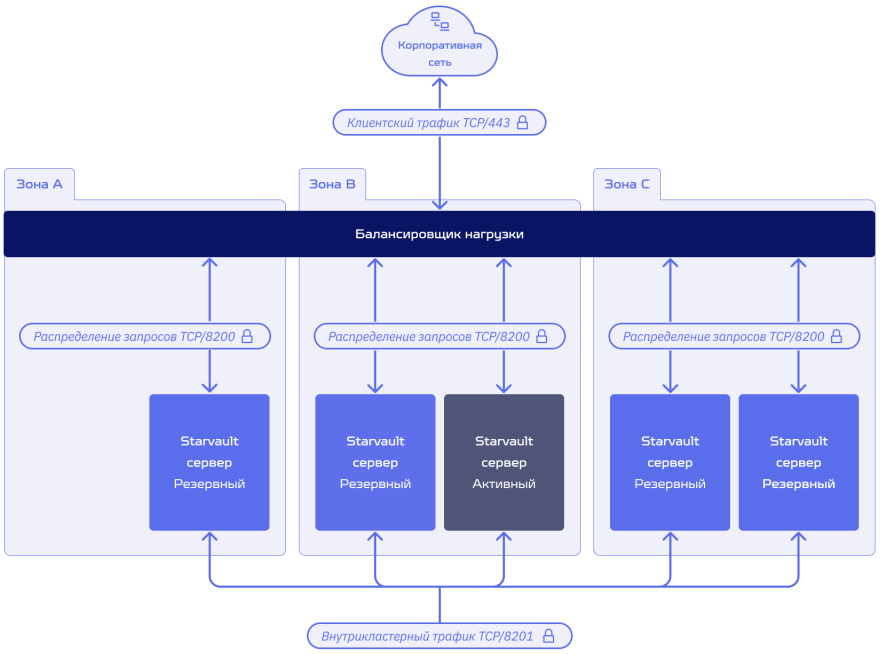
1.1. Серверы в роли лидера

1.1.1. Метрики

```
vault.core.leadership_lost

vault.core.leadership_setup_failed
```

1.1.2. Описание метрик



На диаграмме показан высокодоступный кластер StarVault с пятью узлами, распределенными между тремя зонами доступности. Интегрированное хранилище StarVault использует согласованный протокол для обеспечения синхронизации между узлами кластера. Ведущий (активный) узел отвечает за прием новых записей журнала, их репликацию на последующие (резервные) узлы и управление временем фиксации записи. Integrated Storage использует согласованный протокол для обеспечения синхронизации; поэтому, если лидер потерян, узлы выбирают нового лидера. Более подробную информацию см. в документации по Integrated Storage.

Если вы используете StarVault с интегрированным хранилищем, оно автоматически предоставляет дополнительные метрики для изменения руководства Raft.

1.1.3. Оповещение

Метрика `vault.core.leadership_lost` измеряет продолжительность, в течение которой сервер занимал лидирующую позицию, прежде чем потерять ее. Стабильно низкое значение этого показателя свидетельствует о высокой сменяемости руководства, что указывает на потенциальную нестабильность в кластере.

С другой стороны, скачки метрики `vault.core.leadership_setup_failed` указывают на то, что резервные серверы не могут успешно взять на себя роль лидера, когда это требуется. Незамедлительно расследуйте эти сбои и проверьте, нет ли проблем с получением блокировки выборов лидера. Эти метрики являются важными предупреждениями и могут указывать на риски безопасности и надежности.

Например, может возникнуть проблема со связью между StarVault и бэкендом хранилища или более широкое отключение, приводящее к отказу нескольких серверов StarVault. Мониторинг и анализ этих показателей поможет выявить и устранить все основные проблемы, обеспечив стабильность и безопасность кластера StarVault.

1.2. Высокая задержка в приложении

1.2.1. Метрики

`vault.core.handle_login_request`

`vault.core.handle_request`

1.2.2. Описание метрик

StarVault может использовать доверенные источники, такие как Kubernetes и Active Directory для проверки личности клиентов (пользователей или сервисов) перед предоставлением им доступа к секретам. Клиенты должны пройти аутентификацию, сделав запрос на вход с помощью команды `starvault login` или API. После успешной аутентификации StarVault выдает клиенту токен, который хранится локально на машине клиента и используется для авторизации последующих запросов. Если клиент предъявляет действительный токен, срок действия которого не истек, StarVault признает его аутентифицированным.

1.2.3. Оповещение

Усредненный показатель `vault.core.handle_login_request` измеряет, насколько быстро StarVault реагирует на запросы клиента на вход в систему. Если вы заметили значительное увеличение этого показателя, но не заметили существенного увеличения количества выпущенных токенов (`vault.token.creation`), важно немедленно выяснить причину этой проблемы.

Когда клиент отправляет запрос в StarVault, ему обычно необходимо пройти процедуру аутентификации, чтобы подтвердить свою личность и получить необходимые разрешения. Этот процесс аутентификации включает проверку учетных данных клиента, таких как имя пользователя и пароль или токен API, и обеспечение того, чтобы у клиента были соответствующие права доступа.

Если процесс аутентификации в StarVault выполняется медленно, StarVault требуется больше времени для проверки учетных данных клиента и авторизации запроса. Такая задержка в аутентификации напрямую влияет на время ответа StarVault на запрос клиента.

Необходимо отслеживать показатель `vault.core.handle_request` , который измеряет нагрузку на сервер. Этот показатель помогает определить, нужно ли расширять кластер для

удовлетворения возросшего трафика. С другой стороны, внезапное снижение пропускной способности может указывать на проблемы с подключением между StarVault и его клиентами, которые следует изучить дополнительно.

1.3. Трудности с настройкой аудита или проблемы с установкой пользовательского бэкенда плагина

1.3.1. Метрики

```
vault.core.post_unseal
```

1.3.2. Описание метрик

Серверы StarVault могут находиться в одном из двух состояний:

- запечатанном
- незапечатанном

Для обеспечения безопасности StarVault не доверяет бэкендам хранилищ и хранит данные в зашифрованном виде. После запуска StarVault должен пройти процесс распечатывания, чтобы получить root ключ в виде открытого текста, необходимый для чтения ключа расшифровки данных. После распечатывания StarVault выполняет различные операции, чтобы правильно настроить сервер, прежде чем он сможет начать отвечать на запросы.

1.3.3. Оповещение

Если вы заметили внезапное увеличение метрики `vault.core.post_unseal`, это означает, что на доступность вашего сервера во время процесса post-unseal могут влиять такие проблемы, как ошибки с аудитом или монтированием бэкенда пользовательского плагина. Чтобы диагностировать проблемы, обратитесь к журналам сервера StarVault.

2. Метрики использования

2.1. Влияние чрезмерного создания маркеров на производительность хранилища

2.1.1. Метрики

```
vault.token.creation
```

2.1.2. Описание метрик

Все аутентифицированные запросы StarVault требуют наличия действительного клиентского токена. Токены связаны с политиками, определяющими, какими возможностями обладает клиент (пользователь или система) для определенного пути. StarVault выпускает три типа токенов: **сервисные**, **пакетные** и токены **восстановления**.

Сервисные токены - это то, что пользователи обычно воспринимают как "обычные" токены StarVault. Они поддерживают все функции, такие, как обновление, отзыв, создание дочерних токенов и другие. Соответственно, их создание и отслеживание требует больших усилий.

Пакетные токены - это зашифрованные двоичные объекты большого размера (блобы), содержащие лишь достаточную информацию о клиенте. Хотя StarVault не хранит пакетные токены, он хранит токены сервисов. Объем пространства, необходимый для хранения служебного токена, зависит от метода аутентификации. Поэтому большое количество сервисных токенов может привести к проблеме нехватки памяти.

Токены восстановления используются исключительно для работы StarVault в режиме восстановления.

2.1.3. Оповещение

Отслеживая количество созданных токенов (`vault.token.creation`) и частоту запросов на вход (`vault.core.handle_login_request`, считая от общего числа), вы можете получить **представление об общей загруженности системы**. Если ваш сценарий предполагает запуск множества короткоживущих процессов, как, например, в бессерверных рабочих нагрузках, вы можете столкнуться с одновременным созданием и запросом секретов от сотен или тысяч функций. В таких случаях вы увидите коррелированные всплески в обеих метриках.

При работе с переходными рабочими нагрузками следует использовать пакетные токены, чтобы повысить производительность кластера. StarVault создает пакетный токен, который шифрует всю информацию клиента и предоставляет ее клиенту. Когда клиент использует этот токен, StarVault расшифровывает сохраненные метаданные и выполняет запрос. В отличие от сервисных токенов, пакетные не хранят информацию о клиенте и не реплицируются по кластерам. Эта особенность позволяет снизить нагрузку на бэкэнд хранилища и повысить производительность кластера.

Чтобы узнать больше о пакетных токенах, обратитесь к руководству по пакетным токенам.

2.2. Жизненный цикл аренды приводит к неожиданным скачкам трафика в Vault

2.2.1. Метрики

```
vault.expire.num_leases
```

2.2.2. Описание метрик

StarVault создает договор аренды, когда генерирует динамический секрет или служебный маркер. Эта аренда содержит важную информацию, такую, как время жизни секрета или токена (TTL), а также возможность продления или обновления. StarVault хранит договор аренды в бэкенде хранилища. Если StarVault не продлит договор аренды до истечения TTL, он истечет и будет признан недействительным, что приведет к аннулированию связанного с ним секрета или токена.

2.2.3. Оповещение

Отслеживание количества активных договоров аренды на вашем сервере StarVault (`vault.expire.num_leases`) может дать ценную информацию об уровне активности сервера. Увеличение количества договоров аренды свидетельствует о повышении объема трафика для вашего приложения. В то же время внезапное снижение может указывать на проблемы с доступом к динамическим секретам, которые могут быть получены достаточно быстро для обслуживания входящего трафика.

Рекомендуется устанавливать **минимально возможный TTL** для аренд, чтобы повысить безопасность и производительность. Для этого есть две основные причины:

1. Более короткий TTL снижает влияние потенциальных атак.
2. Это предотвращает бесконечное накопление договоров аренды и использование избыточного пространства в серверной части хранилища.

Если вы не укажете TTL в явном виде, срок аренды по умолчанию составит 32 дня. Однако если произойдет резкий скачок нагрузки и будет сгенерировано множество аренд с таким длинным TTL по умолчанию, бэкэнд хранилища может быстро достичь своей максимальной емкости и выйти из строя, что приведет к его недоступности.

В зависимости от конкретного случая использования вам может потребоваться токен или секрет не на полные 32 дня, а на несколько минут или часов. Установив соответствующий TTL, вы сможете освободить место в хранилище для хранения новых секретов и маркеров. В случае StarVault Enterprise можно установить квоту на количество аренд, чтобы ограничить количество генерируемых аренд ниже определенного порога. Когда порог достигнут, StarVault ограничит создание новых аренд до тех пор, пока срок действия существующей аренды не истечет или она не будет отозвана. Это помогает управлять общим количеством аренд и предотвращает чрезмерное использование ресурсов.

Ознакомьтесь с руководством "Защита StarVault с помощью квот на ресурсы", чтобы узнать, как установить квоты на количество аренд.

Также можно использовать кэширование агента StarVault, чтобы передать управление жизненным циклом аренды агенту StarVault Agent.

Жизненным циклом аренды управляет менеджер истечения срока действия, который обрабатывает отзыв аренды по достижении значения времени жизни, связанного с арендой. Обратитесь к руководству по устранению проблем с безотзывными арендами, если вы столкнулись с безотзывными арендами, отслеживаемыми метрикой `vault.expire.num_irrevocable_leases.`

2.3. Среднее время, необходимое для обновления или отзыва клиентских токенов

2.3.1. Метрики

```
vault.expire.renew-token

vault.expire.revoke
```

2.3.2. Описание метрик

StarVault автоматически отзывает доступ к секретам, предоставленный токеном, когда истекает время его жизни (TTL). Вы можете вручную отозвать токен, если есть признаки возможного нарушения безопасности. Когда токен становится недействительным (истек срок действия или отозван), клиент теряет доступ к секретам, управляемым StarVault. Поэтому клиент должен либо обновить токен до истечения срока его действия, либо запросить новый.

2.3.3. Оповещение

Контроль за своевременным выполнением операций отзыва (`vault.expire.revoke`) и обновления (`vault.expire.renew-token`) крайне важен для обеспечения валидности и доступности секретов. Некоторые приложения, работающие длительное время, могут потребовать продления срока действия токена вместо получения нового. В этом случае время, необходимое для обновления токена, может потенциально повлиять на доступ приложения к секретам. Кроме того, важно отслеживать время, необходимое для выполнения операции отзыва, чтобы обнаружить несанкционированный доступ к секретам, поскольку злоумышленники, получившие доступ, могут проникнуть в вашу систему и нанести вред.

Если вы заметили значительные задержки в процессе отзыва, вам следует изучить журналы сервера на предмет проблем с бэкендом, которые могли помешать процессу отзыва.

3. Бэкэнд хранилища

3.1. Обнаружение проблем с производительностью бэкэнд-хранилища StarVault

3.1.1. Метрики

```
vault.<STORAGE>.get

vault.<STORAGE>.put

vault.<STORAGE>.list

vault.<STORAGE>.delete
```

3.1.2. Описание метрик

Производительность бэкэнда хранилища влияет на общую производительность StarVault. Поэтому очень важно отслеживать производительность бэкэнда хранилища, чтобы обнаружить и отреагировать на любую аномалию. Мониторинг бэкэнда позволяет убедиться в том, что инфраструктура хранения функционирует исправно. Отслеживание производительности позволяет собрать подробную информацию о работе бэкэнда и выявить области, требующие улучшения или оптимизации.

3.1.3. Оповещение

Если StarVault требует больше времени для доступа к бэкенду при выполнении таких операций, как извлечение (`vault.<STORAGE>.get`), хранение (`vault.<STORAGE>.put`), создание списка (`vault.<STORAGE>.list`) или удаление (`vault.<STORAGE>.delete`) элементов, клиенты StarVault могут наблюдать задержки, вызванные ограничениями хранения. Чтобы решить эту проблему, вы можете настроить оповещения, которые будут автоматически уведомлять вашу команду, когда доступ StarVault к бэкенду хранилища замедляется. Это позволит принять меры, например обновить диски с более высокой производительностью ввода-вывода (I/O), до того, как увеличение задержек негативно скажется на работе пользователей приложения.

Если вы используете интегрированное хранилище, читайте следующие статьи с дополнительными рекомендациями:

- Проверка данных в интегрированном хранилище
- Проверка данных в базе данных BoltDB

4. Аудит

4.1. Заблокированные устройства аудита

4.1.1. Метрики

`vault.audit.log_request_failure`

`vault.audit.log_response_failure`

4.1.2. Описание метрик

Устройства аудита играют важную роль в обеспечении соответствия нормативным требованиям, записывая полный журнал аудита запросов и ответов StarVault. Для производственного развертывания в кластере StarVault должно быть включено как минимум одно устройство аудита, чтобы вы могли отслеживать все входящие запросы и исходящие ответы, связанные с кластером. Если вы полагаетесь только на одно устройство аудита, то столкнетесь с такими проблемами:

- потерей сетевого соединения
- проблемами с правами доступа

StarVault может перестать реагировать на запросы и их обрабатывать. Включение хотя бы одного дополнительного устройства аудита необходимо для обеспечения бесперебойной работы и оперативного реагирования StarVault.

4.1.3. Оповещение

Для обеспечения бесперебойной работы важно отслеживать любое необычное увеличение количества отказов в журнале запросов аудита (`vault.audit.log_request_failure`) и отказов в ответах (`vault.audit.log_response_failure`). Эти сбои могут указывать на блокировку устройства. При возникновении подобных проблем, изучение журналов аудита может помочь определить проблемное устройство и дать дополнительные подсказки об основной проблеме.

Если StarVault не может записать журналы аудита в syslog, сервер будет генерировать журналы ошибок, как показано в следующем примере:

```bash
2025-06-10T12:34:56.290Z [ERROR] audit: backend failed to log response:
backend=syslog/ error="write unixgram @->/test/log: write: message too long"
2025-06-10T12:34:56.291Z [ERROR] core: failed to audit response:
request_path=sys/mounts
 error="1 error occurred:
        * no audit backend succeeded in logging the response"
```

Для каждого неудачного ответа журнала следует ожидать несколько ошибок от модулей аудита и ядра. Если вы получаете сообщение об ошибке, содержащее `write: message too long` , это означает, что записи, которые StarVault пытается записать в устройство аудита syslog, превышают размер буфера отправки сокета узла syslog. В таких случаях необходимо

выяснить, что является причиной создания больших записей журнала, например обширный список групп Active Directory или LDAP.

5. Ресурсы

5.1. Проблемы с памятью хранилища, на которые указывает сборка мусора

5.1.1. Метрики

```
vault.runtime.sys_bytes

vault.runtime.gc_pause_ns
```

5.1.2. Описание метрик

Сборка мусора в среде выполнения Go временно приостанавливает все операции. Обычно эти паузы кратковременны, но при высоком уровне использования памяти сборка мусора происходит чаще. Увеличение частоты сбора мусора может привести к задержкам в работе StarVault.

5.1.3. Оповещение

Анализ взаимосвязи между использованием памяти StarVault (в процентах от общего объема доступной памяти на хосте) и временем паузы сборки мусора (измеряется `vault.runtime.gc_pause_ns`). Эта метрика может дать ценные сведения об ограничении ресурсов и помочь в эффективном распределении вычислительных ресурсов.

> Например, если `vault.runtime.sys_bytes` превышает 90 % доступной памяти на хосте, рекомендуется добавить больше памяти, чтобы предотвратить снижение производительности.

Кроме того, следует настроить оповещение, которое срабатывает, если время паузы GC превышает 5 секунд в минуту. Такое оповещение незамедлительно сообщит вам об этом, что позволит оперативно решить проблему.

5.2. Время ожидания ввода/вывода процессора

5.2.1. Описание

StarVault масштабируется горизонтально путем добавления новых экземпляров или узлов, однако практические пределы масштабируемости все же существуют. Чрезмерное время

ожидания процессора при выполнении операций ввода-вывода может указывать на то, что система достигает пределов масштабируемости или чрезмерно использует определенные ресурсы. Отслеживая эти показатели, администраторы могут оценить масштабируемость системы и предпринять соответствующие действия, например оптимизировать операции ввода-вывода или добавить дополнительные ресурсы, чтобы поддерживать производительность по мере роста системы.

5.2.2. Оповещение

Для обеспечения оптимальной производительности рекомендуется поддерживать время ожидания ввода-вывода на уровне менее 10 процентов. Если вы заметили слишком большое время ожидания, это означает, что клиенты испытывают задержки при ожидании ответа StarVault на их запросы. Такая задержка может негативно сказаться на производительности приложений, которые полагаются на StarVault. В таких ситуациях необходимо проверить, правильно ли настроены ресурсы для работы с нагрузкой и равномерно ли распределены запросы по всем процессорам. Эти шаги помогут устранить потенциальные проблемы с производительностью и обеспечить бесперебойную работу StarVault и зависимых от него приложений.

5.3. Поддержание пропускной способности сети в пределах порогового значения

5.3.1. Метрики

```
quota.rate_limit.violation
```

5.3.2. Описание метрик

Мониторинг пропускной способности сети кластеров StarVault позволяет оценить их загруженность. Внезапное снижение входящего или исходящего трафика может указывать на проблемы со связью между StarVault и его клиентами или зависимостями. И наоборот, если вы наблюдаете неожиданный всплеск сетевой активности, это может быть признаком атаки типа "отказ в обслуживании" (DoS). Знание этих сетевых закономерностей может дать ценные сведения и помочь выявить потенциальные проблемы или угрозы безопасности.

5.3.3. Оповещение

Вы можете устанавливать квоты на ограничение скорости, чтобы обеспечить общую стабильность и работоспособность StarVault. Когда сервер достигнет этого порога, StarVault будет отклонять все новые запросы клиентов и отвечать ошибкой HTTP 429, а именно "Слишком много запросов".

Эти отклоненные запросы будут записаны в журналы аудита и отобразят сообщение, как в этом примере: `"error: request path kv/app/test: rate limit quota exceeded."`

Выбор подходящего предела для квоты скорости очень важен, чтобы не блокировать легитимные запросы и не вызывать замедления работы приложений. Чтобы отслеживать частоту таких нарушений и соответствующим образом корректировать лимит, вы можете следить за метрикой `quota.rate_limit.violation`, которая увеличивается при каждом нарушении квоты лимита скорости.

Обратитесь к руководству по защите StarVault с помощью квот на ресурсы, чтобы узнать, как установить квоты на ограничение скорости для вашего StarVault.

6. Репликация

6.1. Освобождение памяти в бэкенде хранилища с помощью мониторинга журналов Write-Ahead

6.1.1. Метрики

```
vault.wal_flushready

vault.wal.persistWALs
```

6.1.2. Описание метрик

Для поддержания высокой производительности в StarVault используется сборщик мусора, который периодически удаляет старые журналы записи (WAL), чтобы освободить память на внутреннем хранилище. Однако при неожиданных скачках трафика накопление WAL может происходить быстро, что приводит к увеличению нагрузки на бэкэнд хранилища. Такие скачки могут негативно повлиять на другие процессы в StarVault, которые полагаются на тот же бэкэнд хранилища. Поэтому важно оценить влияние репликации на производительность бэкэнда хранилища. Это позволит лучше понять, как процесс репликации влияет на общую производительность системы.

6.1.3. Оповещение

Рекомендуется следить за двумя метриками: `vault.wal_flushready` и `vault.wal.persistWALs`.

Первая метрика измеряет время, необходимое для отправки готового журнала записи (WAL) в очередь persist.

Вторая - время, необходимое для персистирования WAL в бэкенд хранилища.

Для обеспечения эффективной работы рекомендуется настроить оповещения, которые будут уведомлять вас, когда метрика `vault.wal_flushready` превысит 500 миллисекунд или когда метрика `vault.wal.persistWALs` превысит 1 000 миллисекунд. Эти предупреждения служат индикаторами того, что обратное давление замедляет работу бэкенда хранилища.

Если срабатывает одно из этих предупреждений, подумайте о масштабировании бэкенда хранилища, чтобы учесть возросшую нагрузку. Масштабирование поможет снизить нагрузку и поддерживать оптимальную производительность.

6.2. Проверка работоспособности StarVault Enterprise Replication

6.2.1. Метрики

```
vault.replication.wal.last_wal
```

6.2.2. Описание метрик

Журнал StarVault с опережающей записью (WAL) - это механизм долговременного хранения и восстановления данных. WAL - это файл журнала, в который записываются все изменения, внесенные в хранилище данных StarVault, прежде чем StarVault сохранит их в базовом бэкенде хранилища. WAL обеспечивает дополнительный уровень надежности и гарантирует целостность данных в случае системных сбоев или отказов.

6.2.3. Оповещение

При развертывании StarVault Enterprise с настройками **Performance Replication** и/или **Disaster Recovery Replication** необходимо следить за тем, чтобы данные реплицировались с первичного на вторичные кластеры. Чтобы обнаружить, что первичный и вторичный кластеры теряют синхронизацию, можно сравнить последний индекс Write-Ahead Log (WAL) на обоих кластерах.

> Важно обнаружить расхождения между ними, поскольку если вторичные кластеры значительно отстают от первичного и первичный кластер становится недоступным, любые запросы к StarVault будут выдавать устаревшие данные.

Поэтому, если вы заметили недостающие значения в WAL, необходимо выяснить возможные причины, которые могут включать:

- **Сетевые проблемы между первичным и вторичным кластерами**: проблемы с сетевым подключением могут препятствовать правильной репликации данных между кластерами.

- **Ограничение ресурсов на первичной или вторичной системе**: если первичный или вторичный кластеры испытывают нехватку ресурсов, это может повлиять на их способность эффективно реплицировать данные.
- **Проблемы с определенными ключами**: иногда проблема может быть связана с определенными ключами в хранилище.

Чтобы выявить такие проблемы, изучите журналы операций и хранения хранилища, в которых содержится подробная информация о проблемных ключах, вызывающих сбои в синхронизации.

Чтобы узнать больше, обратитесь к руководству по мониторингу репликации хранилища.
