Телеметрия

1. Общие сведения

Процесс сервера StarVault собирает различные рабочие метрики о производительности различных библиотек и подсистем. Эти метрики агрегируются с интервалом в 10 секунд и хранятся в памяти 1 минуту. Метрики с высокой кардинальностью, такие как vault.`secret.kv.count`, сообщаются каждые 10 минут или с интервалом, указанным в разделе телеметрии.

Телеметрию из StarVault необходимо передавать и сохранять в программное обеспечение для агрегации метрик, чтобы контролировать StarVault и собирать надежные метрики.

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

Примечание

> Начните с ключевых метрик состояния.

Документация по архитектурной структуре Well-Architected включает руководство по актуальным рекомендациям по ключевым метрикам StarVault для стандартных проверок состояния. Мы рекомендуем ознакомиться с рекомендациями по ключевым метрикам, чтобы определить метрики, которые вы, возможно, захотите начать отслеживать немедленно.

2. Работа с необработанными данными телеметрии

Вы можете просмотреть необработанные данные телеметрии с целью отладки, прервав процесс StarVault сигналом USR1 (на *nix) или BREAK (на Windows). Когда процесс StarVault получает этот сигнал, он выводит данные телеметрии за последние 10 секунд в stderr.

Необработанные данные телеметрии маркируются префиксом, обозначающим тип метрики:

- `[C]` указывает, что метрика является **счетчиком**;
- `[G]` указывает, что метрика является **датчиком**;
- `[S]` указывает, что метрика является **сводной**.

3. Пример вывода необработанной телеметрии

```
[2017-12-19 20:37:50 +0000 UTC][G] 'vault.7f320e57f9fe.expire.num_leases':
5100.000
[2017-12-19 20:37:50 +0000 UTC][G] 'vault.7f320e57f9fe.runtime.num_goroutines':
39.000
[2017-12-19 20:37:50 +0000 UTC][G] 'vault.7f320e57f9fe.runtime.sys_bytes':
222746880.000
[2017-12-19 20:37:50 +0000 UTC][G] 'vault.7f320e57f9fe.runtime.malloc_count':
109189192.000
[2017-12-19 20:37:50 +0000 UTC][G] 'vault.7f320e57f9fe.runtime.free_count':
108408240.000
[2017-12-19 20:37:50 +0000 UTC][G] 'vault.7f320e57f9fe.runtime.heap_objects':
780953.000
[2017-12-19 20:37:50 +0000 UTC][G] 'vault.7f320e57f9fe.runtime.total_gc_runs':
232.000
[2017-12-19 20:37:50 +0000 UTC][G] 'vault.7f320e57f9fe.runtime.alloc_bytes':
72954392.000
[2017-12-19 20:37:50 +0000 UTC][G]
'vault.7f320e57f9fe.runtime.total_gc_pause_ns': 150293024.000
[2017-12-19 20:37:50 +0000 UTC][S] 'vault.merkle.flushDirty': Count: 100 Min:
0.008 Mean: 0.027 Max: 0.183 Stddev: 0.024 Sum: 2.681 LastUpdated: 2017-12-19
20:37:59.848733035 +0000 UTC m=+10463.692105920
[2017-12-19 20:37:50 +0000 UTC][S] 'vault.merkle.saveCheckpoint': Count: 4 Min:
0.021 Mean: 0.054 Max: 0.110 Stddev: 0.039 Sum: 0.217 LastUpdated: 2017-12-19
20:37:57.048458148 +0000 UTC m=+10460.891835029
[2017-12-19 20:38:00 +0000 UTC][G] 'vault.7f320e57f9fe.runtime.alloc_bytes':
73326136.000
[2017-12-19 20:38:00 +0000 UTC][G] 'vault.7f320e57f9fe.runtime.sys_bytes':
222746880.000
[2017-12-19 20:38:00 +0000 UTC][G] 'vault.7f320e57f9fe.runtime.malloc_count':
109195904.000
[2017-12-19 20:38:00 +0000 UTC][G] 'vault.7f320e57f9fe.runtime.free_count':
108409568.000
[2017-12-19 20:38:00 +0000 UTC][G] 'vault.7f320e57f9fe.runtime.heap_objects':
786342.000
[2017-12-19 20:38:00 +0000 UTC][G]
```

```
'vault.7f320e57f9fe.runtime.total_gc_pause_ns': 150293024.000
[2017-12-19 20:38:00 +0000 UTC][G] 'vault.7f320e57f9fe.expire.num_leases':
5100.000
[2017-12-19 20:38:00 +0000 UTC][G] 'vault.7f320e57f9fe.runtime.num_goroutines':
39.000
[2017-12-19 20:38:00 +0000 UTC][G] 'vault.7f320e57f9fe.runtime.total_gc_runs':
232.000
[2017-12-19 20:38:00 +0000 UTC][S] 'vault.rollback.attempt.pki-': Count: 1 Sum:
0.070 LastUpdated: 2017-12-19 20:38:01.96867005 +0000 UTC m=+10465.812041936
[2017-12-19 20:38:00 +0000 UTC][S] 'vault.route.rollback.auth-app-id-': Count: 1
Sum: 0.012 LastUpdated: 2017-12-19 20:38:01.969146401 +0000 UTC
m=+10465.812516689
[2017-12-19 20:38:00 +0000 UTC][S] 'vault.rollback.attempt.identity-': Count: 1
Sum: 0.063 LastUpdated: 2017-12-19 20:38:01.968029888 +0000 UTC
m=+10465.811400276
[2017-12-19 20:38:00 +0000 UTC][S] 'vault.rollback.attempt.database-': Count: 1
Sum: 0.066 LastUpdated: 2017-12-19 20:38:01.969394215 +0000 UTC
m=+10465.812764603
[2017-12-19 20:38:00 +0000 UTC][S] 'vault.barrier.get': Count: 16 Min: 0.010
Mean: 0.015 Max: 0.031 Stddev: 0.005 Sum: 0.237 LastUpdated: 2017-12-19
20:38:01.983268118 +0000 UTC m=+10465.826637008
[2017-12-19 20:38:00 +0000 UTC][S] 'vault.merkle.flushDirty': Count: 100 Min:
0.006 Mean: 0.024 Max: 0.098 Stddev: 0.019 Sum: 2.386 LastUpdated: 2017-12-19
20:38:09.848158309 +0000 UTC m=+10473.691527099
```

4. Содержание раздела

Устройство для файлового аудита

Устройство для файлового аудита (`file`) записывает логи аудита в файл. Это очень простое устройство аудита: оно добавляет логи в файл.

В настоящее время это устройство не помогает выполнять ротацию файлов с логами. Существуют очень стабильные и функциональные инструменты для ротации файлов с логами, поэтому рекомендуем использовать существующие инструменты.

Отправка `SIGHUP` процессу StarVault заставит устройства файлового аудита закрыть и снова открыть свой базовый файл, что может помочь в ротации файлов с логами.

1. Примеры

Включение файла аудита, указав путь по умолчанию:

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

Включение файла аудита, указав другой путь. Можно включить несколько копий устройства аудита:

```bash
starvault audit enable -path=«starvault_audit_1» file
file_path=/home/user/starvault_audit.log
```

Включение вывода логов в файл на `stdout` :

```bash
starvault audit enable file file_path=stdout
```

2. Конфигурация

Обратите внимание на разницу между параметрами команды `audit enable` и параметрами конфигурации файлового бэкенда.
Используйте команду `starvault audit enable -help` , чтобы посмотреть параметры команды.

Устройство файлового аудита поддерживает общие параметры конфигурации описанные на странице "Аудит" и также эти специфические для устройства параметры:

- `file_path` (`string: <необходимо>`) - путь к месту записи файла с логами аудита. Если по указанному пути уже существует файл, бэкенд аудита будет добавлять. Есть несколько специальных ключевых слов:
  - `stdout` — записывает файл с логами аудита в стандартный вывод
  - `discard` — отбрасывает выходные данные, вместо того чтобы записывать на устройство (полезно в сценариях тестирования)
- `mode` (`string: «0600»`) — строка, содержащая восьмеричное число, представляющее битовый шаблон для определения режима доступа к файлу, аналогично **chmod**. Например, установите значение `«0000»`, чтобы запретить StarVault изменять режим файла.

3. Ротация файлов с логами

Для правильной ротации файлов с логами StarVault File Audit Device на серверах StarVault на базе BSD, Darwin или Linux важно настроить программное обеспечение таким образом, чтобы процессу `starvault` после каждой ротации файла с логами посылался сигнал `hang up `/`SIGHUP`.

Устройство аудита Socket

Устройство аудита `socket` записывает данные в сокет `TCP`, `UDP` или `UNIX`.

> При использовании типа аудита `UDP` для устройства аудита `socket` может произойти потеря журналов аудита. Тип аудита UDP` не имеет соединений, то есть когда конечная точка UDP становится недоступной, то возможно, что любое количество отправленных к ней логов аудита может быть потеряно, хотя запрос все равно будет выполнен. StarVault не предоставляет индикации потери журналов аудита. Поэтому рекомендуем использовать ваше устройство в сочетании со вторичным «несокетным» устройством аудита, чтобы обеспечить точность и гарантировать, что логи аудита не будут потеряны.

> Если используется тип аудита `TCP` для устройства аудита `socket` и во время работы было потеряно соединения с сокетом, то одна запись аудита может быть пропущена. Запрос от устройства аудита `socket` будет успешным, несмотря на пропуск записи аудита.

1. Примеры

Устройство аудита `socket` можно включить следующей командой:

```bash
starvault audit enable socket
```

Передать параметры конфигурации можно с помощью пар **K=V** (ключ=значение):

```bash
starvault audit enable socket address=127.0.0.1:9090 socket_type=tcp
```

2. Конфигурация

Устройство аудита `socket` поддерживает общие параметры конфигурации, описанные на странице "Аудит".
Специфические параметры для устройства аудита сокет:

- `address` (`string: «»`) — Адрес сервера сокетов, который необходимо использовать. Пример `127.0.0.1:9090` или `/tmp/audit.sock`.
- `socket_type` (`string: «tcp»`) — Тип используемого сокета, допустим любой тип, совместимый с *net.Dial*. Если используется TCP и сокет назначения становится недоступным, то StarVault может стать невосприимчивым в соответствии с блокировкой устройств аудита.

- `write_timeout (string: 2s)` — время в секундах для завершения записи через сокет. Нулевое значение означает, что попытки записи не будут завершаться.
