



**Руководство по системному администрированию
(клиентская часть)
компонента Объединенный мониторинг Unimon (код
компонента: MONA)
продукта Platform V Monitor (код продукта: OPM)**

ОГЛАВЛЕНИЕ

Руководство по системному администрированию	2
Сценарии администрирования	2
События системного журнала	4
События мониторинга	5
Конфигурирование	59
Установка топика Kafka по умолчанию	60
Создание подключения и объектов хранилища.....	60
Квотирование	60
Создание и редактирование правил выключения метрик	60
Сбор метрик с узлов и виртуальных машин по IP(протокол HTTP).....	60
Сбор метрик с узлов и виртуальных машин по хосту(протокол HTTPs).....	61
Часто встречающиеся проблемы и пути их устранения	62

Руководство по системному администрированию

Здесь и далее поддерживаемой системой приложений-контейнеров является Kubernetes (использование OSE – опционально). В переменных, именах и параметрах системы могут встречаться названия систем контейнеризации, которые применимы для различных сред контейнеризации, указанных в Руководстве по установке.

Термины и определения

Общие термины и определения, используемые в данном документе, представлены в общей документации продукта Platform V Monitor (OPM) в документе «Общее описание продукта Platform V Monitor (OPM)».

Сценарии администрирования

1. Администратору рекомендуется регулярно выполнять:
 - контроль состояния работы системы;
 - мониторинг производительности системы;
 - контроль свободного места на жестких дисках всех серверов системы, а также в файловой системе.

Сделать это можно с помощью системных метрик (описаны ниже) и метрики доступности. Также контролировать работу сервиса можно с помощью логов Unimon-sender и Unimon-agent, проверять их на наличие ошибок.

2. При выявлении нештатных ситуаций необходимо:
 - a. Проверить поступают ли метрики в топик Kafka;
 - b. Проверить логи Unimon-agent на наличии ошибок;
 - c. Проверить логи Unimon-sender на наличии ошибок.
3. В рамках выполнения требований безопасной работы системы Администратор выполняет следующие функции:
 - осуществляет контроль использования организационных и технических средств защиты информации (т.е. контролирует выполнение правил и использование ПО должностными лицами в целях обеспечения защиты информации);
 - осуществляет контроль доступа к обрабатываемым данным пользователями, согласно с их правами доступа к АС;
 - несет ответственность за качество проводимых им работ.

Доступ к АС должны иметь только те сотрудники, которым он необходим в соответствии с их должностными обязанностями. Доступ должен ограничиваться минимально необходимым объемом данных. Должны разделяться среды разработки, тестирования и эксплуатации. При этом производится разделение обязанностей между разработчиками АС, тестирующим персоналом и сотрудниками, непосредственно эксплуатирующими уже введенные в промышленную эксплуатацию системы.

Используемые порты:

Наименование файла	Порт	Описание
--------------------	------	----------

deployment unimon-sender	8080	порт для приема метрик
	8081	порт для публикации метрик
service unimon-sender	8080	порт для приема метрик
	8081	порт для публикации метрик
	10098	alias порта 8080
deployment egress	8443	трафик в prometheus federate
	15021	liveness/readiness probe
	7072	внутренний порт egress для доступа в Abyss
service egress	443 (8443)	трафик в prometheus federate
	27070	порт для подключения к сервису Журналирование
	7071	внутренний порт для подключения к Unimon-server
	7073	внутренний порт egress для доступа в сервис Audit
	7075	внутренний порт egress для подключения к виртуальным машинам
	7077	внутренний порт egress для подключения к сервису аутентификации
	7079	внутренний порт egress для подключения к сервису авторизации
	7072	внутренний порт egress для доступа в Abyss

	15021	liveness/readiness probe
deployment ingress	15021	liveness/readiness probe
	15012	используется для control-plane Istio
service ingress	15021	liveness/readiness probe
	11443	шлюз ingress

Описание парольной политики не применимо, так как Unimon не предполагает создание учетных записей/паролей пользователей, это действие происходит в рамках сервисов IAM Proxu в составе продукта Platform V IAM SE (IAM) или Abyss .

События системного журнала

Если в системном журнале Unimon-agent нет сообщений об ошибках, значит все работает штатно.

1. В системном журнале Unimon-sender должны публиковаться события о передаче метрик в хранилище вида:

```
2020-10-23 07:42:04,194 [http-nio-0.0.0.0-8080-exec-23] [INFO]
(com.sbt.opsmon.unimon.api.MonitoringController)
[com.sbt.opsmon.unimon.api.MonitoringController::sendMetrics:50] mdc:{} Metrics received: 90
```

2. Если нет соединения Unimon-sender с Unimon-server, выдается ошибка в логе:

```
[2021-09-20T10:47:30.929Z] "GET /mona/v1/instance HTTP/1.1" 502 - "-" "-" 0 0 0 - "-"
"Java/11.0.4" "101d3e8b-0d94-9f9d-8ea8-e5d96a5940ad" "r3-1.ingress.unimon.ci01976100-
edevgen2-unimon-dev.apps.dev-gen2.sigma.sbrf.ru" "-" - XX.XX.XXX.XXX:XX
XX.XXX.XXX.XX:XXXX - block_all
```

3. Если метрики не были отправлены в Unimon-sender, выдается ошибка в логе Unimon-agent:

```
{"caller":"dedupe.go:112","component":"remote","lastSendTimestamp":1650872278,"level":"war
n","minSendTimestamp":1650872282,"msg":"Skipping resharding, last successful send was
beyond threshold","remote_name":"2f7932","ts":"2022-04-
25T07:38:12.930Z","url":"http://unimon-sender:10098/monitoring/v1/metrics/send" }
```

4. При возникновении проблемы получения списка endpoint сервисов с метриками, выдается ошибка в логе Unimon-agent:

```
{"caller":"klog.go:94","component":"k8s_client_runtime","func":"ErrorDepth","level":"error","ms
g":"/app/discovery/kubernetes/kubernetes.go:362: Failed to list *v1.Service: Get
```

```
https://172.30.0.1:443/api/v1/namespaces/ci01976100-edevgen2-unimon-  
dev/services?limit=500&resourceVersion=0: dial tcp 172.30.0.1:443: connect: connection  
refused", "ts": "2022-04-25T07:37:13.049Z" }
```

События мониторинга

В таблице представлены варианты метрик и запросов для мониторинга модуля Unimon-agent. Под `sample` понимается объект, состоящий из значения `float64` и метки времени с точностью до миллисекунды. `Sample` формируют фактические данные временных рядов.

Название метрики	Краткое описание	Тип на дашборде	Подробное описание	Пример запроса
Доступность				
up	Доступность Unimon-agent	Graph	В метрике должно быть указано "labels.app": "unimon-agent", значение 1-доступен	

process_start_time_seconds	Доступность	Graph	<p>Подсчет количества уникальных значений по лейблу "labels.pod" за момент времени с гранулярностью \$time. Условно верная метрика, не отражает фактического статуса недоступности, а строит предположение на основе отсутствия значения в окне гранулярности \$time, так при условии, что Abyss в одно окно гранулярности не забрал метрику, например сфейлилась аналитическая задача фактически сервис доступен для потребителя и функционирует, а на графике зафиксировано сокращение количества реплик. Снижение количества реплик, если оно не ожидается, это основание посмотреть, что, что-то идет не так</p>	<p>Query: select \$time as "time", count(distinct("labels.pod")) as "pod" from "\$druidtable" \$where and name = 'process_start_time_seconds' and "labels.namespace" = '\$namespace' and "labels.pod" in ([[pod])group by \$time</p>
----------------------------	-------------	-------	--	--

process_start_time_seconds	Uptime подов / Версия приложения	Table	<p>Отображение uptime контейнера с приложением на основе метрики process_start_time_seconds.</p> <p>Неожиданное уменьшение времени без изменения имени пода свидетельствует о перезапуске контейнера</p>	<p>Query: select "labels.app" as "App", "labels.distribVersion" as "Version", "labels.pod" as "Pod", TIMESTAMP_TO_MILLIS(CURRENT_TIMESTAMP) / 1000 - LATEST(CAST("value" AS FLOAT)) as "Uptime"from "\$druidtable" \$where and name = 'process_start_time_seconds' and "labels.namespace" = '\$namespace' and "labels.pod" in ([pod]group by "labels.app", "labels.pod", "labels.distribVersion"</p>
Производительность				

<p>prometheus_tsdb_head_samples_appended_total</p>	<p>Скорость добавления sample в локальную базу</p>	<p>Graph</p>	<p>Интенсивность считывания и добавления sample в БД Prometheus на основе метрики prometheus_tsdb_head_samples_appended_total (Total number of appended samples. TYPE prometheus_tsdb_head_samples_appended_total counter). Показатель отражает текущую интенсивность входящего потока sample</p>	<p>Query: select TIME_FLOOR(__time, 'PT30S') as "time", LATEST("value") as "value", "labels.pod" as "pod" from "\$druidtable" \$where and name = 'prometheus_tsdb_head_samples_appended_total' and "labels.namespace" = '\$namespace' and "labels.pod" in ([pod]) group by TIME_FLOOR(__time, 'PT30S'), "labels.pod"</p>
--	--	--------------	---	--

prometheus_tsdh_head_series	Количество временных рядов	Graph	Количество активных временных рядов в Prometheus на основе метрики prometheus_tsdh_head_series (Total number of series in the head block). От количества активных временных рядов зависит величина потребления памяти, что при неконтрольном росте может приводить к исчерпанию памяти. При наличии большого значения данной метрики и низком количестве endpoint сбора метрик рекомендуется обратить внимание на публикуемые метрики и рассмотреть возможные пути оптимизации количества публикуемых метрик, например исключить дублирование метрик или уменьшение вариативности значений лейблов с целью сокращения временных рядов	Query: select TIME_FLOOR(__time, 'PT30S') as "time", LATEST("value") as "value", "labels.pod" as "pod" from "\$druidtable" \$where and name = 'prometheus_tsdh_head_series' and "labels.namespace" = '\$namespace' and "labels.pod" in ([pod]) group by TIME_FLOOR(__time, 'PT30S'), "labels.pod"
-----------------------------	----------------------------	-------	---	---

<p>prometheus_target_scrapes_sample_out_of_order_total</p>	<p>Количество sample отклоненных из-за несоответствия ожидаемому порядку</p>	<p>Table</p>	<p>Количество sample с неверным порядком на основе метрики prometheus_target_scrapes_sample_out_of_order_total (Total number of samples rejected due to not being out of the expected order. TYPE prometheus_target_scrapes_sample_out_of_order_total counter). Может возникать при федерировании, например с корневого Прометейя, свидетельствует о получении sample с более ранней датой, чем уже получено</p>	<p>Query: select (LATEST("value") - EARLIEST("value")) as "value", "labels.pod" as "pod" from "\$druidtable" \$where and name = 'prometheus_target_scrapes_sample_out_of_order_total' and "labels.namespace" = '\$namespace' and "labels.pod" in ([[pod])group by "labels.pod"</p>
--	--	--------------	--	--

scrape_samples_scraped	Количество собранных sample	Graph	Количество собранных sample на основе метрики scrape_samples_scraped (the number of samples the target exposed). Отражает количество sample, которое было собрано с endpoint	Query: select TIME_FLOOR(__time, 'PT30S') as "time", LATEST("value"), "labels.pod"from "\$druidtable" \$where and name = 'scrape_samples_scraped' and "labels.namespace" = '\$namespace' -- and "labels.pod" in ([[pod]group by TIME_FLOOR(__time, 'PT30S'), "labels.pod" select TIME_FLOOR(__time, 'PT30S') as "time", LATEST("value") as "system-federate-metrics"from "\$druidtable" \$where and name = 'scrape_samples_scraped' and "labels.job"='system-federate-metrics' -- and "labels.pod" in ([[pod]group by TIME_FLOOR(__time, 'PT30S')
------------------------	-----------------------------	-------	---	---

scrape_duration_seconds	Длительность сбора sample	Graph	Длительность сбора метрик на основе метрики scrape_duration_seconds. Если длительность сбора превышает timeout сбора, сбор не будет осуществлен	Query: select TIME_FLOOR(__time, 'PT15S') as "time", AVG("value") as "AVG", MIN("value") as "MIN", MAX("value") as "MAX" -- "labels.pod"from "\$druidtable" \$where and name = 'scrape_duration_seconds' and "labels.namespace" = '\$namespace' -- and "labels.pod" in ([pod]group by TIME_FLOOR(__time, 'PT15S') -- "labels.pod"
-------------------------	---------------------------	-------	---	---

prometheus_sd_discovered_targets	Количество целей сбора	Graph	Количество целей сбора на основе метрики prometheus_sd_discovered_targets (Current number of discovered targets) в разрезе конфигурации сбора	Query: select TIME_FLOOR(__time, 'PT30S') as "time", LATEST("value") as " ", "labels.pod", "labels.config" from "\$druidtable" \$where and name = 'prometheus_sd_discovered_targets' and "labels.namespace" = '\$namespace' and "labels.pod" in ([pod] and "labels.name" = 'scrape' and "value" > 0 group by TIME_FLOOR(__time, 'PT30S'), "labels.pod", "labels.config"
----------------------------------	------------------------	-------	---	---

<p>prometheus_remote_storage_succeeded_samples_total</p>	<p>Скорость отправки sample в удаленное хранилище</p>	<p>Graph</p>	<p>Интенсивность успешной отправки sample при настроенной конфигурации remote_write в endpoint для приема метрик на основе метрики prometheus_remote_storage_succeeded_samples_total (Total number of samples successfully sent to remote storage). Стоит учитывать, что отправка производится пакетами sample, по умолчанию 100 sample в пакете. В нормальном случае должна коррелировать с метрикой считывания и добавления sample. При заметной разнице, есть вероятность, что endpoint приема метрик не справляется с нагрузкой, вследствие чего может расти очередь отправки метрик и запаздывание появления метрик в хранилище</p>	<p>Query: select TIME_FLOOR(__time, 'PT30S') as "time", LATEST("value") as " ", "labels.pod", "labels.url" from "\$druidtable" \$where and (name = 'prometheus_remote_storage_succeeded_samples_total' or name = 'prometheus_remote_storage_samples_total') and "labels.namespace" = '\$namespace' and "labels.pod" in ([pod]group by TIME_FLOOR(__time, 'PT30S'), "labels.pod", "labels.url"</p>
--	---	--------------	--	---

<p>prometheus_remote_storage_pending_samples</p>	<p>Количество ожидающих sample в очередях shard для отправки в удаленное хранилище</p>	<p>Graph</p>	<p>Показывает количество метрик ожидающих отправки в очереди на основе метрики prometheus_remote_storage_pending_samples (The number of samples pending in the queues shards to be sent to the remote storage). Рост данного показателя может означать медленную обработку метрик endpoint приема метрик или другие проблемы с данным endpoint. Возможно следует переконфигурирование размера пакета отправки. На основе этого показателя рассчитывается количество потоков отправки пакетов метрик. Не должно быть постоянного роста, но некоторое колебание считать нормой</p>	<p>Query: select TIME_FLOOR(__time, 'PT30S') as "time", LATEST("value") as "pending samples", -- (LATEST("value") - EARLIEST("value"))/15 as "pending samples per sec", "labels.pod", "labels.url" from "\$druidtable" \$where and name = 'prometheus_remote_storage_pending_samples' and "labels.namespace" = '\$namespace' and "labels.pod" in ([pod]group by TIME_FLOOR(__time, 'PT30S'), "labels.pod", "labels.url"</p>
--	--	--------------	--	---

<p>prometheus_remote_storage_sent_batch_duration_seconds_count</p>	<p>Количество пакетной отправки sample в удаленное хранилище в секунду</p>	<p>Graph</p>	<p>Интенсивность отправки пакетов метрик в удаленное хранилище на основе метрики prometheus_remote_storage_sent_batch_duration_seconds (Duration of sample batch send calls to the remote storage). Интенсивность вызова REST API сервиса приема метрик, отражает нагрузку на сервис приема метрик</p>	<p>Query: select TIME_FLOOR(__time, 'PT30S') as "time", LATEST("value") as " ", "labels.pod", "labels.url" from "\$druidtable" \$where and name = 'prometheus_remote_storage_sent_batch_duration_seconds_count' and "labels.namespace" = '\$namespace' and "labels.pod" in ([pod]) group by TIME_FLOOR(__time, 'PT30S'), "labels.pod", "labels.url"</p>
--	--	--------------	--	---

<p>prometheus_remote_storage_queue_highest_sent_timestamp_seconds, prometheus_remote_storage_queue_highest_timestamp_in_seconds</p>	<p>Отставание метки времени метрики в очереди</p>	<p>Graph</p>	<p>На основе метрик prometheus_remote_storage_queue_highest_sent_timestamp_in_seconds (Timestamp from a WAL sample, the highest timestamp successfully sent by this queue, in seconds since epoch) и prometheus_remote_storage_highest_timestamp_in_seconds (Highest timestamp that has come into the remote storage via the Appender interface, in seconds since epoch) вычисляем фактическое отставание метрик в секундах. Должно быть приемлемым</p>	<p>Query: select "time", "time_in_queue" - "lag", "labels.pod" from (select TIME_FLOOR(_time, 'PT15S') as "time", MAX("value") - MIN("value") as "value", "labels.pod", LATEST("value") FILTER (where name = 'prometheus_remote_storage_queue_highest_sent_timestamp_in_seconds') as "time_in_queue", LATEST("value") FILTER (where name = 'prometheus_remote_storage_highest_timestamp_in_seconds') as "time" from "\$druidtable" \$where and "labels.namespace" = '\$namespace' and "labels.pod" in ([pod] group by TIME_FLOOR(_time, 'PT15S'), "labels.pod")</p>
---	---	--------------	---	---

<p>prometheus_remote_storage_sent_bytes_total</p>	<p>Скорость отправки sample в удаленное хранилище, байт в секунду</p>	<p>Graph</p>	<p>Description: Количество байт отправленных в сервис удаленного приема метрик в секунду на основе метрики prometheus_remote_storage_sent_bytes_total (The total number of bytes sent by the queue)</p>	<p>Query: select TIME_FLOOR(__time, 'PT30S') as "time", LATEST("value") as " ", "labels.pod", "labels.url" from "\$druidtable" \$where and name = 'prometheus_remote_storage_sent_bytes_total' and "labels.namespace" = '\$namespace' and "labels.pod" in ([pod]) group by TIME_FLOOR(__time, 'PT30S'), "labels.pod", "labels.url"</p>
---	---	--------------	---	--

<p>prometheus_remote_storage_shards, prometheus_remote_storage_shards_desired</p>	<p>Рекомендуемое/текущее количество shard, параллельность отправки</p>	<p>Graph</p>	<p>Description: Количество активных потоков отправки пакетов метрик. удаленное хранилище и расчетнотребуется от интенсивности обработки метрик агентом на основе метрик prometheus_remote_storage_shards (The number of shards used for parallel sending to the remote storage) и prometheus_remote_storage_shards_desired (The number of shards that the queues shard calculation wants to run based on the rate of samples in vs. samples ou)</p>	<p>Query: select TIME_FLOOR(__time, 'PT1M') as "time", MAX("value") as "shards", "labels.pod", "labels.url"from "\$druidtable" \$where and name = 'prometheus_remote_storage_shards' and "labels.namespace" = '\$namespace' and "labels.pod" in ([pod]group by TIME_FLOOR(__time, 'PT1M'), "labels.pod", "labels.url" select TIME_FLOOR(__time, 'PT1M') as "time", MAX("value") as "desired", "labels.pod", "labels.url"from "\$druidtable" \$where and name = 'prometheus_remote_storage_shards_desired' and "labels.namespace" = '\$namespace' and "labels.pod" in ([pod]group by TIME_FLOOR(__time, 'PT1M'), "labels.pod", "labels.url"</p>
---	--	--------------	---	--

<p>prometheus_remote_storage_sent_batch_duration_seconds_sum</p>	<p>Средняя длительность отправки пакета sample в удаленное хранилище</p>	<p>graph</p>	<p>Средняя длительность отправки пакета sample в сервис приема метрик на основе метрики prometheus_remote_storage_sent_batch_duration_seconds (Duration of sample batch send calls to the remote storage). Длительность отражает задержку отработки запроса сервисом удаленного приема метрик, и зависит от загруженности данного сервиса или других сетевых задержек. Следует анализировать с временем отработки запроса самого сервиса для определения на какой этап уходит время при зафиксированном его росте</p>	<p>Query: select "time", "sum" / "cnt" as "duration", "labels.pod" from (select TIME_FLOOR(__time, 'PT30S') as "time", "labels.pod", LATEST(CAST("value" AS FLOAT)) FILTER (where name = 'prometheus_remote_storage_sent_batch_duration_seconds_sum') - EARLIEST(CAST("value" AS FLOAT)) FILTER (where name = 'prometheus_remote_storage_sent_batch_duration_seconds_sum') as "sum", LATEST(CAST("value" AS INTEGER)) FILTER (where name = 'prometheus_remote_storage_sent_batch_count') - EARLIEST(CAST("value" AS INTEGER)) FILTER (where name = 'prometheus_remote_storage_sent_batch_count') as "cnt" from "\$druidtable" \$where and "labels.namespace" = '\$namespace' and</p>
--	--	--------------	---	---

				<pre>"labels.pod" in ([pod) group by TIME_FLOOR(__ti me, 'PT30S'), "labels.pod")</pre>
--	--	--	--	--

<p>prometheus_remote_storage_sent_batch_duration_seconds_sum</p>	<p>Длительность отправки пакетов sample в удаленное хранилище</p>	<p>graph</p>	<p>Длительность отправки определенного количества пакетов sample на основе метрики prometheus_remote_storage_sent_batch_duration_seconds (Duration of sample batch send calls to the remote storage)</p>	<p>Query: select TIME_FLOOR(__time, 'PT30S') as "time", LATEST("value") as "value", "labels.pod", name from "\$druidtable" \$where and (name = 'prometheus_remote_storage_sent_batch_duration_seconds_sum' or name = 'prometheus_remote_storage_sent_batch_duration_seconds_count') and "labels.namespace" = '\$namespace'group by TIME_FLOOR(__time, 'PT30S'), "labels.pod", name</p>
<p>Ошибки и сбои</p>				

<p>prometheus_remote_storage_enqueue_retries_total</p>	<p>Неудачные попытки постановки в очередь из-за переполнения очереди shard</p>	<p>graph</p>	<p>Показывает переполнение очереди на отправку метрик на основе метрики prometheus_remote_storage_enqueue_retries_total (Total number of times enqueue has failed because ashards queue was full). В норме должно быть 0</p>	<p>Query: select TIME_FLOOR(__time, 'PT30S') as "time", LATEST("value") as " ", "labels.pod", "labels.url" from "\$druidtable" \$where and name = 'prometheus_remote_storage_enqueue_retries_total' and "labels.namespace" = '\$namespace' group by TIME_FLOOR(__time, 'PT30S'), "labels.pod", "labels.url"</p>
--	--	--------------	--	---

<p>prometheus_remote_storage_retried_samples_total</p>	<p>Количество повторно отправленных sample после ошибки</p>	<p>graph</p>	<p>Количество sample, которые не были отправлены в удаленное хранилище, но были повторены, поскольку ошибка отправки была исправима на основе метрики prometheus_remote_storage_retried_samples_total (Total number of samples which failed on send to remote storage but were retried because the send error was recoverable). В норме должно тримится к 0</p>	<p>Query: select TIME_FLOOR(__time, 'PT30S') as "time", LATEST("value") as " ", "labels.pod", "labels.url" from "\$druidtable" \$where and name = 'prometheus_remote_storage_retried_samples_total' and "labels.namespace" = '\$namespace' group by TIME_FLOOR(__time, 'PT30S'), "labels.pod", "labels.url"</p>
--	---	--------------	---	---

<p>prometheus_remote_storage_dropped_samples_total</p>	<p>Количество sample, которые были отброшены после чтения из WAL перед отправкой в удаленное хранилище</p>	<p>graph</p>	<p>Количество sample, которые были отброшены из-за переполнения очереди на основе метрики prometheus_remote_storage_dropped_samples_total (Total number of samples which were dropped after being read from the WAL before being sent via remote write)</p>	<p>Query: select TIME_FLOOR(__time, 'PT30S') as "time", LATEST("value") as " ", "labels.pod", "labels.url"from "\$druidtable" \$where and name = 'prometheus_remote_storage_dropped_samples_total' and "labels.namespace" = '\$namespace'group by TIME_FLOOR(__time, 'PT30S'), "labels.pod", "labels.url"</p>
<p>prometheus_remote_storage_failed_samples_total</p>	<p>Количество неотправленных sample</p>	<p>graph</p>	<p>Количество sample, которые не были отправлены в сервис приема метрик на основе метрики prometheus_remote_storage_failed_samples_total (Total number of samples which failed on send to remote storage, non-recoverable errors). В норме должно быть равно 0</p>	<p>Query: select TIME_FLOOR(__time, 'PT30S') as "time", LATEST("value") as " ", "labels.pod", "labels.url"from "\$druidtable" \$where and name = 'prometheus_remote_storage_failed_samples_total' and "labels.namespace" = '\$namespace'group by TIME_FLOOR(__time, 'PT30S'), "labels.pod", "labels.url"</p>

<p>prometheus_target_scrapes_exceeded_sample_limit_total, prometheus_target_scrapes_sample_duplicate_timestamp_total, prometheus_target_scrapes_sample_out_of_bounds_total, prometheus_target_scrapes_sample_out_of_order_total</p>	<p>Ошибки сбора</p>	<p>graph</p>	<p>На графике отражено наличие ошибок на основе метрик: prometheus_target_scrapes_exceeded_sample_limit_total (Total number of scrapes that hit the sample limit and were rejected), prometheus_target_scrapes_sample_duplicate_timestamp_total (Total number of samples rejected due to duplicate timestamps but different values), prometheus_target_scrapes_sample_out_of_bounds_total (Total number of samples rejected due to timestamp falling outside of the time bounds), prometheus_target_scrapes_sample_out_of_order_total (Total number of samples rejected due to not being out of the expected order)</p>	<p>Query: select TIME_FLOOR(__time, 'PT30S') as "time", (LATEST("value") - EARLIEST("value")) as " ", "labels.pod", name from "\$druidtable" \$where and (name = 'prometheus_target_scrapes_exceeded_sample_limit_total' or name like 'prometheus_target_scrapes_sample%' _total') and "labels.namespace" = '\$namespace' and "labels.pod" in ([[pod]group by TIME_FLOOR(__time, 'PT30S'), "labels.pod", name</p>
--	---------------------	--------------	---	--

prometheus_target_scrapes_sample_duplicate_timestamp_total	Количество sample отклоненных из-за дублирования временных меток, но с разными значениями	table	Количество sample отклоненных из-за дублирования временных меток, но с разными значениями за выбранный период на основе метрики prometheus_target_scrapes_sample_duplicate_timestamp_total (Total number of samples rejected due to duplicate timestamps but different values)	Query: select (LATEST("value") - EARLIEST("value")) as "value", "labels.pod" as "pod" from "\$druidtable" \$where and name = 'prometheus_target_scrapes_sample_duplicate_timestamp_total' and "labels.namespace" = '\$namespace' and "labels.pod" in ([[pod])group by "labels.pod"
--	---	-------	--	--

В таблице представлены варианты метрик и запросов для мониторинга модуля Unimon-sender.

Название метрики	Описание, что вычисляем	Тип на дашборде	Описание	Пример запроса
Доступность unimon_sender_health_status	Доступность		Общая работоспособность (1-UP, 0-DOWN, 2-Частичная работоспособность) Если хоть один из компонентов сервиса (метрики unimon_sender_health_status{component=<имя компонента>}) DOWN, тогда общая метрика будет в состоянии 2 - частичная работоспособность. Если все компоненты в DOWN, тогда и общий статус DOWN	

unimon_sender_health_status, где component="unimon-server"	Доступность	Работоспособность (1-UP, 0-DOWN) DOWN, если число последних ошибок при взаимодействии с Unimon-server больше параметра UNIMON_SERVER_HEALTH_STATUSES_ERRORS_THRESHOLD
unimon_sender_health_status, где component="unimon-filter"	Доступность	Работоспособность (1-UP, 0-DOWN) DOWN, если число последних ошибок при взаимодействии с Unimon-filter больше параметра UNIMON_FILTER_HEALTH_STATUSES_ERRORS_THRESHOLD
unimon_sender_health_status, где component="unimon-metadata"	Доступность	Работоспособность (1-UP, 0-DOWN) DOWN, если число последних ошибок при взаимодействии с Unimon-metadata больше параметра UNIMON_METADATA_HEALTH_STATUSES_ERRORS_THRESHOLD
unimon_sender_health_status, где component="kafka-producer"	Доступность	Работоспособность (1-UP, 0-DOWN) DOWN, если число последних ошибок при взаимодействии с kafka-producer больше параметра UNIMON_KAFKA_HEALTH_STATUSES_ERRORS_THRESHOLD
unimon_sender_health_status, где component="kafka-topic"	Доступность	Работоспособность (1-UP, 0-DOWN) DOWN, если число последних ошибок при взаимодействии с kafka-topic больше параметра UNIMON_KAFKA_HEALTH_STATUSES_ERRORS_THRESHOLD

<p>process_start_time_ seconds</p>	<p>Доступность</p>	<p>graph</p>	<p>Подсчет количества уникальных значений по лейблу "labels.pod" за момент времени с гранулярностью \$time. Условно верная метрика, не отражает фактического статуса недоступности, а строит предположение на основе отсутствия значения в окне гранулярности \$time. Так при условии, что Abyss в одно окно гранулярности не забрал метрику, например, сфейлилась аналитическая задача, на графике будет зафиксировано сокращение количества реплик, а фактически сервис доступен для потребителя и функционирует. Снижение количества реплик, если оно не ожидается, основание посмотреть, что что-то идет не так</p>	<p>Query: select TIME_FLOOR("__ time", 'PT1M') as "time", count(distinct("labe ls.pod")) as "pod"from "\$druidtable" \$where and name = 'process_start_time_ seconds' and "labels.namespace" ='\$namespace' and "labels.app"='\$app' and "labels.pod" in (\$pod) --and "value"!='0'group by TIME_FLOOR("__ time", 'PT1M')</p>
--	--------------------	--------------	---	---

process_uptime_seconds	Uptime подов/Версия приложения	table	<p>Отображение uptime контейнера с приложением на основе метрики process_uptime_seconds (The uptime of the Java virtual machine).</p> <p>Неожиданное уменьшение времени без изменения имени пода свидетельствует о перезапуске контейнера</p>	<p>Query: select "labels.app" as "App", "labels.distribVersion" as "Version", "labels.pod" as "Pod", MAX("value") as "Uptime"from "\$druidtable" \$where and name = 'process_uptime_seconds' and "labels.namespace" = '\$namespace' and "labels.app" = '\$app' and "labels.pod" in ([pod) and "value" like '%.%'group by "labels.app", "labels.pod", "labels.distribVersion"order by "Uptime"</p>
Производительность				

http_server_request s_seconds_count	HTTP запросы в секунду (усредненные за 30 секунд)	graph	Интенсивность http-запросов, полученных Unimon-sender в разрезе uri, HTTP метода и HTTP кода, на основе метрики http_server_request s_seconds_count	Query: select TIME_FLOOR("__time", 'PT30S') as "time", LATEST("value") as " ", "labels.pod", "labels.status", "labels.uri", "labels.method" from "\$druidtable" \$where and name = 'http_server_request s_seconds_count' and "labels.namespace" = '\$namespace' and "labels.app" = '\$app' -- and "labels.method" = 'POST' and "labels.pod" in (\$pod) and "labels.uri" in ([API] group by TIME_FLOOR("__time", 'PT30S'), "labels.pod", "labels.status", "labels.uri", "labels.method")
--	---	-------	---	---

http_server_request s_seconds	Среднее время выполнения запроса	graph	Среднее время выполнения запроса в разрезе uri, HTTP метода и HTTP кода на основе метрики http_server_request s_seconds. Показывает за какое время в среднем сервис обрабатывает запрос	Query: select "time", "sum" / "cnt" as " ", "labels.pod", "labels.status", "labels.uri", "labels.method" fro m (select TIME_FLOOR(__ti me, 'PT30S') as "time", "labels.pod", "labels.status", "labels.uri", "labels.method", LATEST(CAST("v alue" AS FLOAT)) FILTER (where name = 'http_server_request s_seconds_sum') - EARLIEST(CAST("value" AS FLOAT)) FILTER (where name = 'http_server_request s_seconds_sum') as "sum", LATEST(CAST("v alue" AS INTEGER)) FILTER (where name = 'http_server_request s_seconds_count') - EARLIEST(CAST("value" AS INTEGER)) FILTER (where name = 'http_server_request s_seconds_count') as "cnt" from "\$druidtable" \$where and "labels.namespace" =\$namespace' and
----------------------------------	--	-------	---	---

				<pre>"labels.app"='\$app' -- and "labels.method"='P OST' and "labels.pod" in ([pod) and "labels.uri" in ([API) group by TIME_FLOOR(__ti me, 'PT30S'), "labels.pod", "labels.status", "labels.uri", "labels.method")</pre>
--	--	--	--	---

<p>http_server_request s_seconds_max</p>	<p>Максимальная длительность HTTP запроса</p>	<p>table</p>	<p>Максимальная продолжительность выполнения запроса в течении временного окна на основе метрики http_server_request s_seconds_max. Временное окно по умолчанию 2 минуты</p>	<p>Query: select TIME_FLOOR(__ti me, 'PT30S') as "time", MAX("value") as " ", "labels.pod", "labels.uri", "labels.method", "labels.status" from "\$druidtable" \$where and name = 'http_server_request s_seconds_max' and "labels.namespace" = '\$namespace' and "labels.app" = '\$app' and "labels.pod" in ([pod] and "labels.uri" in ([API] group by TIME_FLOOR(__ti me, 'PT30S'), "labels.pod", "labels.uri", "labels.method", "labels.status"</p>
--	---	--------------	--	---

<p>http_server_request s_seconds</p>	<p>Количество HTTP запросов со статусом 200</p>	<p>table</p>	<p>Количество запросов со статусом 200 за выбранный интервал на основе метрики http_server_request s_seconds</p>	<p>Query: select TIME_FLOOR("__ time", 'PT30S') as "time", LATEST("value") as " ", "labels.pod", "labels.uri", "labels.method" fro m "\$druidtable" \$where and name = 'http_server_request s_seconds_count' and "labels.namespace" =\$namespace' and "labels.app"=\$app' -- and "labels.method"='P OST' and "labels.status" = '200' and "labels.pod" in (\$pod) and "labels.uri" in ([API]group by TIME_FLOOR("__ time", 'PT30S'), "labels.pod", "labels.uri", "labels.method"</p>
--	---	--------------	--	---

<p>http_server_request s_seconds</p>	<p>Количество HTTP запросов со статусом отличным от 200</p>	<p>table</p>	<p>Количество запросов со статусом отличным от 200 за выбранный интервал на основе метрики http_server_request s_seconds</p>	<p>Query: select TIME_FLOOR("__ time", 'PT30S') as "time", LATEST("value") as " ", "labels.pod", "labels.uri", "labels.method" fro m "\$druidtable" \$where and name = 'http_server_request s_seconds_count' and "labels.namespace" ='\$namespace' and "labels.app"='\$app' -- and "labels.method"='P OST' and "labels.status" != '200' and "labels.pod" in (\$pod) and "labels.uri" in ([API]group by TIME_FLOOR("__ time", 'PT30S'), "labels.pod", "labels.uri", "labels.method"</p>
--	---	--------------	--	--

kafka_producer_top ic_record_send_tot al	Скорость отправки сообщений в Kafka	graph	Интенсивность отправки сообщений в Kafka в разрезе топика и пода на основе метрики kafka_producer_top ic_record_send_tot al (The total number of records sent for a topic)	Query: select TIME_FLOOR("__ time", 'PT30S') as "time", LATEST("value") as " ", "labels.pod", "labels. topic" from "\$druidtable"\$wher eand name = 'kafka_producer_to pic_record_send_tot al'and "labels.namespace" ='\$namespace'and "labels.app"='\$app' and "labels.pod" in (\$pod)--and "value"!='0'group by TIME_FLOOR("__ time", 'PT30S'), "labels.pod", "labels.topic"
--	---	-------	--	---

kafka_producer_request_rate	Запросы в секунду	graph	Среднее количество отправленных запросов в секунду на основе метрики kafka_producer_request_rate (The number of requests sent per second), должно идти ровно без особых всплесков, всплески могут оказать влияние на производительность брокеров или вызвать увеличение задержки в обработке запросов	Query: select \$time as "time", "value" as "value", "labels.pod" as "pod"from "\$druidtable"where \$defaultfilterand name = 'kafka_producer_request_rate'and "labels.namespace" = '\$namespace'and "labels.app" = '\$app'and "labels.pod" in (\$pod)--and "value" != '0'group by \$time, "labels.pod", "value"
-----------------------------	-------------------	-------	---	--

kafka_producer_response_rate	Ответы в секунду	graph	Среднее количество полученных ответов от брокера продюсеру в секунду на основе метрики kafka_producer_response_rate (The number of responses received per second)	Query: select \$time as "time", "value" as "value", "labels.pod" as "pod" from "\$druidtable" where \$defaultfilter and name = 'kafka_producer_response_rate' and "labels.namespace" = '\$namespace' and "labels.app" = '\$app' and "labels.pod" in (\$pod) -- and "value" != '0' group by \$time, "labels.pod", "value"
kafka_producer_request_latency_avg	Среднее время запроса	graph	Средняя задержка запроса в миллисекундах kafka_producer_request_latency_avg (The average request latency in ms). Время между отправкой продюсером и ответом брокера. Размеры сообщений могут влиять на данный показатель увеличивая его	Query: select \$time as "time", "value" as "value", "labels.pod" as "pod" from "\$druidtable" \$where and name = 'kafka_producer_request_latency_avg' and "labels.namespace" = '\$namespace' and "labels.app" = '\$app' and "labels.pod" in (\$pod) -- and "value" != '0' group by \$time, "labels.pod", "value"

kafka_producer_node_request_latency_max	Максимальное время запроса	graph	Максимальная задержка запроса в миллисекундах в разрезе по нодам Kafka на основе метрики kafka_producer_node_request_latency_max	Query: select TIME_FLOOR(__time, 'PT30S') as "time", LATEST("value") as " ", "labels.pod", "labels.node_id" from "\$druidtable"\$ where name = 'kafka_producer_node_request_latency_max' and "labels.namespace" = '\$namespace' and "labels.app" = '\$app' and "labels.pod" in (\$pod) -- and "value" != '0' group by TIME_FLOOR(__time, 'PT30S'), "labels.pod", "labels.node_id"
---	----------------------------	-------	--	---

kafka_producer_no de_request_size_max x	Максимальный размер отправленного запроса	graph	Максимальный размер отправленного запроса в разрезе по нодам Kafka на основе метрики kafka_producer_no de_request_size_max x	Query: select TIME_FLOOR(__time, 'PT30S') as "time", LATEST("value") as " ", "labels.pod", "labels.node_id" from "\$druidtable"\$where and name = 'kafka_producer_no de_request_size_max' and "labels.namespace"= '\$namespace' and "labels.app"=\$app' and "labels.pod" in (\$pod)--and "value"!='0' group by TIME_FLOOR(__time, 'PT30S'), "labels.pod", "labels.node_id"
---	--	-------	--	--

jvm_memory_used_bytes	Heap Memory	graph	<p>Величина используемой HEAP памяти jvm_memory_used_bytes (The amount of used memory)</p> <p>sum all "labels.area" = 'heap'</p>	<p>Query:</p> <pre>select TIME_FLOOR(__time, 'PT15S') as "time", SUM("value") as " ", "labels.pod" from "\$druidtable" \$where and name = 'jvm_memory_used _bytes' and "labels.area" = 'heap' and "labels.namespace" ='\$namespace' and "labels.app" = '\$app' and "labels.pod" in (\$pod) -- and "value" != '0' group by TIME_FLOOR(__ti me, 'PT15S'), "labels.pod"</pre>
-----------------------	-------------	-------	--	--

jvm_memory_used_bytes	NonHeap Memory	graph	<p>Величина используемой non-HEAP памяти jvm_memory_used_bytes (The amount of used memory) sum all "labels.area" = 'nonheap'</p>	<p>Query: select TIME_FLOOR(__time, 'PT15S') as "time", SUM("value") as "labels.pod" from "\$druidtable"\$ where name = 'jvm_memory_used_bytes' and "labels.area" = 'nonheap' and "labels.namespace" = '\$namespace' and "labels.app" = '\$app' and "labels.pod" in (\$pod) -- and "value" != '0' group by TIME_FLOOR(__time, 'PT15S'), "labels.pod"</p>
-----------------------	----------------	-------	--	--

jvm_gc_live_data_size_bytes, jvm_gc_max_data_size_bytes, jvm_gc_live_data_size_bytes_util	Утилизация heap OLD GEN после full GC	graph	Утилизация пула старшего поколения после вызова GC. 100 * "jvm_gc_live_data_size_bytes" / "jvm_gc_max_data_size_bytes" as "jvm_gc_live_data_size_bytes_util" HELP jvm_gc_max_data_size_bytes Max size of old generation memory pool. TYPE jvm_gc_max_data_size_bytes gauge HELP jvm_gc_live_data_size_bytes Size of old generation memory pool after a full GC. TYPE jvm_gc_live_data_size_bytes gauge	Query: select "time", 100 * "jvm_gc_live_data_size_bytes" / "jvm_gc_max_data_size_bytes" as "jvm_gc_live_data_size_bytes_util", "labels.pod" from (select TIME_FLOOR(__time, 'PT30S') as "time", "labels.pod", LATEST("value") FILTER (where name = 'jvm_gc_live_data_size_bytes') as "jvm_gc_live_data_size_bytes", LATEST("value") FILTER (where name = 'jvm_gc_max_data_size_bytes') as "jvm_gc_max_data_size_bytes" from "\$druidtable" \$where and "labels.namespace" = '\$namespace' and "labels.pod" in ([[pod]) group by TIME_FLOOR(__time, 'PT30S'), "labels.pod")
---	---------------------------------------	-------	--	--

jvm_gc_live_data_size_bytes	Размер heap OLD GEN после full GC	graph	Размер пула старшего поколения после вызова GC на основе метрики jvm_gc_live_data_size_bytes (Size of old generation memory pool after a full GC)	Query: select TIME_FLOOR(__time, 'PT15S') as "time", LATEST("value") as " ", "labels.pod"from "\$druidtable" \$where and name = 'jvm_gc_live_data_size_bytes' and "labels.namespace" = '\$namespace' and "labels.app" = '\$app' and "labels.pod" in ([pod])group by TIME_FLOOR(__time, 'PT15S'), "labels.pod"
-----------------------------	-----------------------------------	-------	---	---

jvm_gc_memory_allocated_bytes_total	Увеличение размера молодого поколения между вызовами GC	graph	Увеличение пула молодого поколения между вызовами GC на основе метрики jvm_gc_memory_allocated_bytes_total (Incremented for an increase in the size of the young generation memory pool after one GC to before the next)	Query: select TIME_FLOOR(__time, 'PT30S') as "time", LATEST("value") as " ", "labels.pod"from "\$druidtable" \$where and name = 'jvm_gc_memory_allocated_bytes_total' and "labels.namespace" = '\$namespace' and "labels.app" = '\$app' and "labels.pod" in ([pod]group by TIME_FLOOR(__time, 'PT30S'), "labels.pod"
jvm_gc_pause_seconds_max	Максимальное время в minor GC	table	Максимальное время в GC на основе метрики jvm_gc_pause_seconds_max (Time spent in GC pause). Временное окно по умолчанию около 2х минут	Query: select MAX("value") as " ", "labels.pod"from "\$druidtable" \$where and name = 'jvm_gc_pause_seconds_max' and "labels.action" = 'end of minor GC' and "labels.namespace" = '\$namespace' and "labels.app" = '\$app' and "labels.pod" in ([pod]group by "labels.pod"

jvm_gc_pause_seconds_max	Максимальное время в major GC	table	Максимальное время в GC на основе метрики jvm_gc_pause_seconds_max (Time spent in GC pause). Временное окно по умолчанию около 2х минут	Query: select MAX("value") as " ", "labels.pod" from "\$druidtable" \$where and name = 'jvm_gc_pause_seconds_max' and "labels.action" = 'end of major GC' and "labels.namespace" = '\$namespace' and "labels.app" = '\$app' and "labels.pod" in ([pod])group by "labels.pod"
--------------------------	-------------------------------	-------	--	--

<p>process_files_open_files,process_files_max_files</p>	<p>Утилизация файловых дескрипторов для процесса</p>	<p>graph</p>	<p>Утилизация файловых дескрипторов в относительной величине на основе метрик process_files_open_files (The open file descriptor count) и process_files_max_files (The maximum file descriptor count) 100 *"process_files_open_files" / "process_files_max_files" as "process_files_util"</p>	<p>Query: select "time", 100 *"process_files_open_files" / "process_files_max_files" as "process_files_util", "process_files_util", "labels.pod" from (select TIME_FLOOR(__time, 'PT30S') as "time", "labels.pod", LATEST("value") FILTER (where name = 'process_files_open_files') as "process_files_open_files", LATEST("value") FILTER (where name = 'process_files_max_files') as "process_files_max_files" from "\$druidtable" \$where and "labels.namespace" = '\$namespace' and "labels.pod" in ([pod] group by TIME_FLOOR(__time, 'PT30S'), "labels.pod")</p>
---	--	--------------	---	--

process_files_max_files	Максимальное количество файловых дескрипторов для процесса	table	Максимальное значение файловых дескрипторов process_files_max_file descriptor count)	Query: select MAX("value") as " ", "labels.pod" from "\$druidtable" \$where and name = 'process_files_max_files' and "labels.namespace" = '\$namespace' and "labels.app" = '\$app' and "labels.pod" in ([pod] group by "labels.pod"
process_files_open_file	Количество открытых файловых дескрипторов	table	Абсолютная величина открытых файловых дескрипторов process_files_open_files (The open file descriptor count)	Query: select LATEST("value") as " ", "labels.pod" from "\$druidtable" \$where and name = 'process_files_open_files' and "labels.namespace" = '\$namespace' and "labels.app" = '\$app' and "labels.pod" in (\$pod) group by "labels.pod"

jvm_memory_used_bytes	Heap Memory Pools	graph	<p>Величина используемой памяти HEAP в разрезе memory pools на основе метрики jvm_memory_used_bytes (The amount of used memory)</p>	<pre>Query: select TIME_FLOOR(__time, 'PT30S') as "time", avg("value") as " ", "labels.pod", "labels.id" from "\$druidtable" \$where and name = 'jvm_memory_used_bytes' and "labels.area" = 'heap' and "labels.namespace" ='\$namespace' and "labels.app"='\$app' and "labels.pod" in (\$pod)group by TIME_FLOOR(__time, 'PT30S'), "labels.pod", "labels.id"</pre>
-----------------------	-------------------	-------	---	--

jvm_memory_used_bytes	NonHeap Memory Pools	graph	<p>Величина используемой памяти non-HEAP в разрезе memory pools на основе метрики jvm_memory_used_bytes (The amount of used memory)</p>	<p>Query: select TIME_FLOOR(__time, 'PT30S') as "time", avg("value") as " ", "labels.pod", "labels.id" from "\$druidtable" \$where and name = 'jvm_memory_used_bytes' and "labels.area" = 'nonheap' and "labels.namespace" = '\$namespace' and "labels.app" = '\$app' and "labels.pod" in (\$pod)group by TIME_FLOOR(__time, 'PT30S'), "labels.pod", "labels.id"</p>
-----------------------	----------------------	-------	---	--

jvm_threads_live_threads,jvm_threads_daemon_threads	Количество потоков	graph	Текущее количество потоков на основе метрик jvm_threads_live_threads (The current number of live threads including both daemon and non-daemon threads) и jvm_threads_daemon_threads (The current number of live daemon threads). Показывает общее количество "живых" потоков и количество потоков демонов, выполняющих фоновые задачи	Query: select TIME_FLOOR(__time, 'PT30S') as "time", LATEST("value") as " ", "labels.pod", namefrom "\$druidtable" \$where and (name = 'jvm_threads_live_threads' or name = 'jvm_threads_daemon_threads') and "labels.namespace" = '\$namespace' and "labels.app" = '\$app' and "labels.pod" in (\$pod)group by TIME_FLOOR(__time, 'PT30S'), "labels.pod", name
---	--------------------	-------	---	---

jvm_threads_states_threads	Число потоков в разрезе статусов	graph	Количество потоков по их состояниям на основе метрики jvm_threads_states_threads. Наличие потоков с состоянием BLOCKED показывает количество заблокированных потоков, что может сказываться на производительности	Query: select TIME_FLOOR(__time, 'PT30S') as "time", LATEST("value") as " ", "labels.pod", "labels.state" from "\$druidtable" \$where and name = 'jvm_threads_states_threads' and "labels.namespace" = '\$namespace' and "labels.app" = '\$app' and "labels.pod" in ([pod]) group by TIME_FLOOR(__time, 'PT30S'), "labels.pod", "labels.state"
----------------------------	----------------------------------	-------	---	--

jvm_gc_pause_seconds	Garbage Collector per second	graph	Интенсивность вызова GC в секунду на основе метрики jvm_gc_pause_seconds (Time spent in GC pause)	Query: select TIME_FLOOR(__time, 'PT1M') as "time", LATEST("value") as " ", "labels.pod", name, "labels.action", "labels.cause" from "\$druidtable" \$where and (name = 'jvm_gc_pause_seconds_count') and "labels.namespace" = '\$namespace' and "labels.app" = '\$app' and "labels.pod" in (\$pod) group by TIME_FLOOR(__time, 'PT1M'), "labels.pod", name, "labels.action", "labels.cause"
----------------------	------------------------------	-------	---	--

jvm_gc_pause_seconds	Garbage Collector average time per collect	graph	Средняя длительность цикла вызова GC на основе метрики jvm_gc_pause_seconds (Time spent in GC pause)	Query: select "time", "sum" / "cnt" as "duration", "labels.pod", "labels.action", "labels.cause" from (select TIME_FLOOR(__time, 'PT30S') as "time", "labels.pod", "labels.action", "labels.cause", LATEST(CAST("value" AS FLOAT)) FILTER (where name = 'jvm_gc_pause_seconds_sum') - EARLIEST(CAST("value" AS FLOAT)) FILTER (where name = 'jvm_gc_pause_seconds_sum') as "sum", LATEST(CAST("value" AS INTEGER)) FILTER (where name = 'jvm_gc_pause_seconds_count') - EARLIEST(CAST("value" AS INTEGER)) FILTER (where name = 'jvm_gc_pause_seconds_count') as "cnt" from "\$druidtable" \$where and "labels.namespace" = '\$namespace' and "labels.pod" in (\$pod) group by
----------------------	--	-------	--	--

				<pre>TIME_FLOOR(_ti me, 'PT30S'), "labels.pod", "labels.action", "labels.cause")</pre>
--	--	--	--	--

<p>jvm_classes_loaded _classes,jvm_classes_unloaded_classes_total</p>	<p>Загруженные / выгруженные классы</p>	<p>graph</p>	<p>Текущее количество загруженных классов на основе метрики jvm_classes_loaded _classes (The number of classes that are currently loaded in the Java virtual machine). Количество выгруженных классов на основе метрики jvm_classes_unload ed_classes_total (The total number of classes unloaded since the Java virtual machine has started execution)</p>	<p>Query: select TIME_FLOOR(__ti me, 'PT30S') as "time", LATEST("value") as " ", "labels.pod", namefrom "\$druidtable" \$where and (name = 'jvm_classes_loade d_classes' or name = 'jvm_classes_unloa ded_classes_total') and "labels.namespace" ='namespace' and "labels.app"='\$app' and "labels.pod" in (\$pod)group by TIME_FLOOR(__ti me, 'PT30S'), "labels.pod", name</p>
---	---	--------------	---	---

logback_events_total	Скорость записи логов в секунду в разрезе уровня,	graph	Интенсивность вызова событий библиотеки логирования logback на основе метрики logback_events_total в разрезе уровней логирования. Позволяет увидеть наличие и частоту возникновения логов в разрезе их уровней	Query: select TIME_FLOOR("__time", 'PT30S') as "time", LATEST("value") as " ", "labels.pod", "labels.level" from "\$druidtable" \$where and name = 'logback_events_total' and "labels.namespace" = '\$namespace' and "labels.app" = '\$app' and "labels.pod" in (\$pod) group by TIME_FLOOR("__time", 'PT30S'), "labels.pod", "labels.level"
kafka_producer_request_total	Общее количество отправленных запросов	graph		
kafka_producer_response_total	Общее количество полученных ответов	graph		
kafka_producer_node_request_rate	Количество отправленных запросов в секунду в разрезе по нодам Kafka	graph		
kafka_producer_node_request_total	Общее количество отправленных запросов в разрезе по нодам Kafka	graph		

kafka_producer_request_size_max	Максимальный размер отправленного запроса	graph		
kafka_producer_request_latency_max	Максимальная задержка запроса в миллисекундах	graph		
kafka_producer_metrics_push_success_total	Количество успешно отправленных метрик	graph		
kafka_topics_get_success_total	Количество успешных попыток получения топиков Kafka	graph		
unimon_sender_filters_in_cache	Количество полученных фильтров (закешированных в текущий момент)	graph		
unimon_sender_metadata_buffer_size	Объем буфера метаданных	graph		
unimon_sender_instances_in_cache	Количество полученных действующих подключений (закешированных в текущий момент)	graph		
unimon_sender_metrics_collected_total	Количество собранных метрик	graph		
unimon_sender_metrics_not_collected_total	Количество отфильтрованных метрик	graph		
unimon_sender_instances_in_cache	Количество полученных действующих подключений (закешированных в текущий момент)	graph		

Ошибки и сбои				
kafka_producer_topic_record_error_total	Ошибки отправки сообщений в Kafka	graph	Количество отправленных записей, которые привели к ошибкам на основе метрики kafka_producer_topic_record_error_total (The total number of records sent that resulted in errors for a topic). В норме должно быть 0	Query: select TIME_FLOOR("__time", 'PT30S') as "time", LATEST("value") as " ", "labels.pod", "labels.topic" from "\$druidtable"\$ where and name = 'kafka_producer_topic_record_error_total' and "labels.namespace" = '\$namespace' and "labels.app" = '\$app' and "labels.pod" in (\$pod) -- and "value" != '0' group by TIME_FLOOR("__time", 'PT30S'), "labels.pod", "labels.topic"
kafka_producer_record_error_rate	Среднее количество отправленных записей в секунду, которые привели к ошибкам	graph		
kafka_producer_topic_record_error_rate	Среднее количество отправленных записей в секунду, которые привели к ошибкам в разрезе по топикам	graph		

unimon_sender_get_instances_error_total	Количество ошибок при получении списка подключений	graph		
unimon_sender_get_filters_errors_total	Количество ошибок при получении фильтров	graph		
unimon_sender_send_metadata_errors_total	Количество ошибок при отправке метаданных	graph		
unimon_metric_validation_failure_total	Количество метрик не прошедших валидацию	graph		
kafka_producer_metrics_push_failure_total	Количество неуспешно отправленных метрик	graph		
kafka_producer_record_error_total	Общее количество отправлено неуспешно отправленных метрик	graph		
kafka_topics_get_failure_total	Количество неуспешных попыток получения топиков Kafka	graph		

Для Unimon-sender реализован healthcheck, который помогает определить работоспособность сервиса на текущий момент, а также работоспособность всех компонентов, с которыми взаимодействует данный сервис. Healthcheck можно использовать для геобалансировки.

Есть два варианта проверки работоспособности:

1. Добавить в среде контейнеризации объект Route на Unimon-sender на порт 8081 и путь /actuator/health/components.
2. Через терминал в среде контейнеризации выполнить команду:

```
curl http://0.0.0.0:8081/actuator/health/components
```

Пример ответа для автономной работы:

```
{ "status" : "UP",
  "components" :
    { "kafka-producer" :
      { "status" : "UP", "details" : { "lastErrorsCount" : 0 } },
      "kafka-topic" : { "status" : "UP", "details" : { "lastErrorsCount" : 0 } }
    }
}
```

Пример ответа для неавтономной работы:

```
{ "status" : "UP",
  "components" :
    { "kafka-producer" : { "status" : "UP", "details" : { "lastErrorsCount" : 0 } },
      "kafka-topic" : { "status" : "UP", "details" : { "lastErrorsCount" : 0 } },
      "unimon-filter" : { "status" : "UP", "details" : { "lastErrorsCount" : 0 } },
      "unimon-metadata" : { "status" : "UP", "details" : { "lastErrorsCount" : 0 } },
      "unimon-server" : { "status" : "UP", "details" : { "lastErrorsCount" : 0 } }
    }
}
```

Конфигурирование

Все параметры конфигурации подробно описаны в «Справочнике конфигов». Изменять или задавать параметры конфигурации можно с помощью редактирования файла конфигурации необходимого модуля в среде контейнеризации или в Rasman. На текущий момент в Rasman доступно редактирование следующих конфигураций:

Название файла конфигурации в среде контейнеризации	Название файла конфигурации в Rasman
расман-unimon-common-config	common
расман-unimon-sender-config	sender-config

При редактировании параметров конфигурации в среде контейнеризации необходимо также использовать файлы указанные выше, для обеспечения синхронизации конфигурации в Rasman. Синхронизация параметров конфигурации работает в обе стороны, то есть при изменении в среде контейнеризации, будет выполнено обновление параметров в Rasman и наоборот. При редактировании в среде контейнеризации иных файлов конфигурации (для модулей перечисленных выше) заданные значения не будут применены и синхронизированы с Rasman.

С помощью параметров для Unimon-agent и Unimon-sender можно задавать необходимые значения лимитов памяти и CPU. В конфигурационных файлах указаны рекомендованные значения, которые могут быть изменены на нужные в случае разного потока метрик.

```
unimon-agent.resources.openshift.cpuRequest=500m
unimon-agent.resources.openshift.memRequest=1000Mi
unimon-agent.resources.openshift.cpuLimit=500m
unimon-agent.resources.openshift.memLimit=1000Mi
```

```
unimon-sender.openshift.cpuLimit=500m
unimon-sender.openshift.memLimit=1000Mi
unimon-sender.openshift.cpuRequest=200m
unimon-sender.openshift.memRequest=1000Mi
```

Установка топика Kafka по умолчанию

Для задания топика по умолчанию, заполняется параметр `kafka.topic`, и в случае, если в метрике будет отсутствовать `label UnimonId` (по которому должно происходить определение топика для отправки), метрики будут отправлены в топик, заданный по умолчанию.

Создание подключения и объектов хранилища

Если подразумевается использование подключения клиентской части к серверной (неавтономная работа), то необходимо выполнить подключение клиента к Unimon. После которого, клиент получит `UnimonId`, соответствующий объектам хранилища, который необходимо указывать при передаче метрик. При создании подключения также задается квота для этого подключения, которая выделяется из квоты проекта. Подробное описание квотирования находится в архитектуре Unimon. При автономной работе (без подключения к серверной части) создание подключения и `UnimonId` не требуется. Каждому `unimonId` соответствует топик Kafka, в который Unimon будет писать метрики клиента. Перед созданием подключения должен быть создан проект (в Abyss) название которого, соответствует RN. Для создания, просмотра, редактирования признака основного подключения для проекта, удаления одного или всех подключений проекта необходимо воспользоваться UI (подробное описание в инструкциях Indicator) или методами API Unimon (подробнее в описании API). Для создания, удаления, редактирования подключений пользователь должен обладать ролью `MONA_CONNECTIONS_EDITOR`. Для просмотра подключений в своем проекте необходимо обладать ролью `MONA_CONNECTIONS_EDITOR`.

Квотирование

Подробное описание механизма квотирования находится в «Детальной архитектуре». Квота на проект задается в UI Abyss. Квота на подключение задается при создании подключения, она выделяется из квоты проекта. Для задания квоты на подключение, а также для просмотра квот необходимо воспользоваться UI (подробное описание в инструкциях Indicator). Каждому уже существующему проекту будет выделена квота в размере `MAX_INT` и распределена равномерно между уже существующими подключениями по этому проекту. Свободным останется 20% квоты по проекту для возможности создания новых подключений. При указании значения квоты 0 для подключения будет задана нулевая квота. Не задавать квоту на подключение нельзя.

Создание и редактирование правил выключения метрик

Правила выключения метрик позволяют отключать передачу метрик в топик Kafka. Для создания, просмотра, редактирования правил необходимо воспользоваться UI (подробное описание в инструкциях Indicator) или методами API Unimon. Для создания, удаления, редактирования правил пользователь должен обладать ролью `MONA_FILTER_EDITOR`. Для

просмотра правил выключения метрик в своем проекте необходимо обладать ролью MONA_FILTER_VIEWER.

Сбор метрик с узлов и виртуальных машин по IP(протокол HTTP).

Для мониторинга узлов и виртуальных машин необходима отдельная настройка, после включения которой Unimon-agent начнет осуществлять опрос приложений, установленных на виртуальных машинах, а затем отправку в хранилище:

1. Скопировать исходный код job из дистрибутива клиентской части(/scripts/jenkins/devops-opsmon). Создать репозиторий в сервисе для хостинга систем управления версиями кода (можно использовать уже существующий) и добавить код в него. Удостоверится что папка scripts/jenkins/devops-opsmon/templates лежит в корне репозитория.
2. Сформировать json-файл - targets.json, содержащие IP-адреса на которых постанут метрики и положить его в репозиторий для возможного редактирования.

Внимание! Для подключения Unimon, требуется получить идентификатор подключения UnimonId и указывать его в качестве label в targets.json. Если UnimonId не указывать, отправка метрики будет производиться в топик по умолчанию (параметр kafka.topic в namespace мониторинга). Если параметр kafka.topic будет не заполнен либо отсутствовать, метрика будет игнорироваться.

Для того чтобы метрики с виртуальных машин отображались на дашбордах в Grafana, необходимо чтобы для каждого набора таргетов присутствовал лейбл «app», в котором указывается имя приложения, публикующего метрики. Пример targets.json:

```
[
  {
    "labels": {
      "app": "app1",
      "unimonId": "tenant1",
      "__metrics_path__": "/metrics"
    },
    "targets": ["XXX.XX.X.X:XXXXX", "YYY.YY.Y.Y:YYYYYY"]
  }
]
```

3. Необходимо создать job в Jenkins со следующими параметрами:
 - REPO_URL - адрес репозитория в сервисе для хостинга систем управления версиями кода, где лежит файл targets.json;
 - REPO_BRANCH - ветка репозитория в сервисе для хостинга систем управления версиями кода;
 - REPO_CREDS - credentials для подключения к сервису для хостинга систем управления версиями кода;
 - MONITORING_VERSION - версия мониторинга;
 - JSON_VM_NAME - имя json-файла, в котором лежат IP-адреса, по которым постанут метрики (по умолчанию targets.json);
 - OSE_CLUSTER - cluster в среде контейнеризации, в котором развернут мониторинг (Unimon)
 - OSE_PROJECT – namespace в среде контейнеризации, в котором развернут мониторинг (Unimon);
 - OSE_CREDS - credentials для подключения в среде контейнеризации;

В разделе Pipeline конфигурации job необходимо указать репозиторий в котором находится job (pipeline.groovy) и путь до нее.

4. Запустить созданный job.

Данный job формирует и разворачивает ConfigMap в указанном namespace, заносит туда содержимое json-файла с хостами VM и создает необходимые компоненты Istio, тем самым включая сбор метрик по http.

Сбор метрик с узлов и виртуальных машин по хосту (протокол HTTPs).

Для мониторинга узлов и виртуальных машин необходима отдельная настройка, после включения которой Unimon-agent начнет осуществлять опрос приложений, установленных на виртуальных машинах, а затем отправку в хранилище:

1. Скопировать исходный код job из дистрибутива клиентской части(/scripts/jenkins/devops-opsmon-https). Создать репозиторий в сервисе для хостинга систем управления версиями кода (можно использовать уже существующий) и добавить код в него. Удостоверится что папка scripts/jenkins/devops-opsmon-https/templates лежит в корне репозитория.
2. Сформировать json-файл - targets.json, содержащий хосты на которых построятся метрики и положить его в репозиторий для возможного редактирования.

Внимание! Для подключения Unimon, требуется получить идентификатор подключения UnimonId и указывать его в качестве label в targets.json. Если UnimonId не указывать, отправка метрики будет производиться в топик по умолчанию (параметр kafka.topic в namespace мониторинга). Если параметр kafka.topic будет не заполнен либо отсутствовать, метрика будет игнорироваться.

Для того чтобы метрики с виртуальных машин отображались на дашбордах в Grafana, необходимо чтобы для каждого набора таргетов присутствовал лейбл «app», в котором указывается имя приложения, публикующего метрики. Пример targets.json:

```
[
  {
    "labels": {
      "app": "app1",
      "unimonId": "tenant1",
      "__metrics_path__": "/metrics"
    },
    "targets": [ "some.cloud.host:9443", "some2.cloud.host:9443" ]
  }
]
```

3. Необходимо создать job в Jenkins со следующими параметрами:

- REPO_URL - адрес репозитория в сервисе для хостинга систем управления версиями кода, где лежит файл targets.json;
- REPO_BRANCH - ветка репозитория в сервисе для хостинга систем управления версиями кода;
- REPO_CREDS - credentials для подключения к сервису для хостинга систем управления версиями кода;
- MONITORING_VERSION - версия мониторинга;
- JSON_VM_NAME - имя json-файла, в котором лежат хосты, по которым построятся метрики (по умолчанию targets.json);

- OSE_CLUSTER - cluster в среде контейнеризации, в котором развернут мониторинг (Unimon);
- OSE_PROJECT - namespace в среде контейнеризации, в котором развернут мониторинг (Unimon);
- OSE_CREDS - credentials для подключения в среде контейнеризации ;

В разделе Pipeline конфигурации job-ы необходимо указать репозиторий в котором находится job (pipeline.groovy) и путь до нее.

4. Запустить созданный job.

Данный job формирует и разворачивает ConfigMap в указанном namespace, заносит туда содержимое json-файла с хостами VM и создает необходимые компоненты Istio, тем самым включая сбор метрик по https.

Часто встречающиеся проблемы и пути их устранения

1. Неправильная конфигурация ресурсов FluentBit приводит к невозможности отправки логов в Журналирование или проблемы с запуском pod Unimon. Необходимо проверить правильность настроек интеграции с Журналирование и отсутствие ошибок в логах pod.
2. Работа при недоступности Platform V One-Time-Token (ОТТ). При недоступности сервиса Platform V One-Time-Token (ОТТ) прекращают работу egress/ingress в связи с этим будет отсутствовать возможность отправки логов в Журналирование и будет недоступен сервис через API. Необходимо обратиться к администраторам стенда для восстановления работы Platform V One-Time-Token (ОТТ).
3. Не поступают метрики отправленные через API Unimon. Проверить корректность настроек ingress и наличие сертификатов у клиента-отправителя.
4. Нет соединения Unimon-sender с Unimon-server, выдается ошибка в логе:

```
[2021-09-20T10:47:30.929Z] "GET /mona/v1/instance HTTP/1.1" 502 - "-" "0 0 0 - "-"
"Java/11.0.4" "101d3e8b-0d94-9f9d-8ea8-e5d96a5940ad" "r3-1.ingress.unimon.ci01976100-
edevgen2-unimon-dev.apps.dev-gen2.sigma.sbrf.ru" "-" - - XX.XX.XXX.XXX:XX
XX.XXX.XXX.XX:XXXXXX - block_all
```

Проблемы с настройками Istio, проверить конфигурацию Istio и правильную маршрутизацию запросов.

5. Метрики не доходят до хранилища, в логе Istio и Platform V One-Time-Token (ОТТ):

```
POST /monitoring/v1/metrics/external/send HTTP/1.1" 403 UAEX
```

```
03:43:16.114 [ERROR] [c.s.o.c.g.s.EnvoyOttServiceClientImpl] [grpc-default-executor-17] -
Envoy check authz error java.lang.Exception: com.sbt.ott.base.api.OttClientException:
[expiry=2022-01-26 03:43:16.795, cause:
com.sbt.ott.api.client.exceptions.OttClientRuntimeException: Unknown TokenRequest type:
```

```
com.sbt.ott.base.api.client.impl.JwsTokenCreateReq, Unknown TokenRequest type:  
com.sbt.ott.base.api.client.impl.JwsTokenCreateReq]          at  
com.sbt.ott.client.grpc.server.EnvoyOttServiceClientImpl.verifyOttToken(EnvoyOttServiceClientI  
mpl.java:213)          at  
com.sbt.ott.client.grpc.server.EnvoyOttServiceClientImpl.check(EnvoyOttServiceClientImpl.java:  
246)
```

Ingress с Platform V One-Time-Token (ОТТ) блокирует трафик, необходимо использовать валидные сертификаты Platform V One-Time-Token (ОТТ).

6. Отсутствие ресурсов Istio или их неправильная конфигурация приводит к потере метрик. Необходимо проверить сетевое взаимодействие сервисов и отсутствие ошибок в логах.

В состав дистрибутива входят конфигурационные файлы с рекомендуемыми значениями нестендозависимых параметров для настройки продукта. Рекомендуем не вносить изменения в эти значения, так как это может нарушить безопасность продукта.