Руководство по эксплуатации

Компонента Сервисный прокси (SVPX)

Продукта Platform V Synapse Service Mesh (SSM)

ОГЛАВЛЕНИЕ

Руководство по эксплуатации

Руководство по системному администрированию

Термины и определения

Термин/аббревиатура	Определение
TLS	Transport Layer Security, протокол защиты транспортного уровня
Платформа	Платформа оркестрации приложений с средствами автоматизации и управления на основе политик, например Kubernetes
Дамп конфигурации	Снимок информации о состоянии конфигурации
Istio SE	Настраиваемая сервисная сетка с открытым исходным кодом, служащая для взаимодействия, мониторинга и обеспечения безопасности контейнеров в кластере Kubernetes
Контрольная панель	Проект, где запущены управляющие приложения Synapse Service Mesh (компонент POLM)
Управление политиками / POLM	Компонент Управление политиками из состава продукта Platform V Synapse Service Mesh
Platform V Synapse Service Mesh / SSM	Программный продукт на базе Istio SE, обеспечивающий возможность создания сервисной сети поверх Платформенной в Kubernetes

Термин/аббревиатура	Определение
Сервисный прокси / SVPX	Компонент Сервисный прокси Platform V Synapse Service Mesh
VirtualService	Конфигурация, изменяющая направление трафика

Сценарии администрирования

Для администрирования программного компонента Сервисный прокси из состава программного продукта Platform V Synapse Service Mesh (далее — Synapse) используются:

- конфигурационные файлы загружаемые в прикладной неймспейс (см. подраздел "Конфигурационные файлы");
- журнал сервисного прокси SVPX (см. раздел "События системного журнала");
- конфигурационный файл компонента POLM (см. раздел "Сценарии администрирования" Руководства по системному администрированию компонента POLM)

Конфигурационные файлы

Базовая конфигурация сервисного прокси осуществляется путем автоматического получения конфигурационного файла от компонента POLM. Администрирование компонента POLM описано в соответствующем разделе руководства по системному администрированию POLM.

1. Для подключения через веб-интерфейс Платформы

Шаг	Действие
Авторизуйтесь в веб-консоли Платформы	Перейдите по ссылке URL в веб-консоли нужного кластера Платформы. Введите в окне ввода учетных данных логин и пароль
Перейдите в проект	Выберите пункт меню Home/Projects и выберите из списка проект

Для подключения через консоль Платформы:

Шаг	Действие

Шаг	Действие
Войдите в консольного клиента платформы	В окне командной строки в приглашении введите команды: kubectl config set-credentials /<host-alias-без-точек>: --token= kubectl config set-cluster <host-alias-без-точек>: --insecure-skip-tls-verify=true --server=https://: kubectl config set-context /<host-alias-без-точек>:/ --user=/<host-alias-без-точек>: --namespace=maximov-test --cluster=<host-alias-без-точек>: kubectl config use-context /<host-alias-без-точек>:/

1. Установите конфигурационные файлы

Для установки через веб-интерфейс Платформы:

Шаг	Действие
Загрузите артефакты из файла <конфигурационный файл>.yaml	В правом верхнем углу окна проекта нажмите иконку с изображением значка + Import YAML. Перенесите, используя механизм drag and drop, файл с секретом из директории, содержащей конфигурационные артефакты. В открывшемся окне редактирования нажмите Create

Для подключения через консоль Платформы:

Шаг	Действие
Загрузите артефакты из файла <конфигурационный файл>.yaml	В консоли выполните команду: kubectl apply -f <Конфигурационный файл>.yaml

Типы конфигурационных файлов

Для настройки сервиса SVPX могут быть применены следующие типы конфигурационных файлов:

- DestinationRule;
- VirtualService;
- ServiceEntry.

Destination Rule

DestinationRule определяет политики применяемые к сервису, после перенаправления на него трафика. Эти правила конфигурируют тип балансировки трафика между подами сервиса, максимальное количество соединений обрабатываемых сервисным прокси, настройки определения отклонений для выявления проблемных подов и исключения их из балансировки.

Пример настройки политик балансировки:

```
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: bookinfo-ratings
spec:
  host: ratings.prod.svc.cluster.local
  trafficPolicy:
    loadBalancer:
      simple: LEAST_CONN
```

Можно настроить отдельные политики, для разных версий приложений на которые ссылается общий сервис, в примере показано правило, применяющее тип балансировки "round robin" для всего трафика, направленного на подгруппу "testversion", объединяющую все эндпойнты (поды) с метками "version:v3":

```
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: bookinfo-ratings
spec:
  host: ratings.prod.svc.cluster.local
  trafficPolicy:
    loadBalancer:
      simple: LEAST_CONN
  subsets:
  - name: testversion
    labels:
      version: v3
    trafficPolicy:
      loadBalancer:
        simple: ROUND_ROBIN
```

Можно определить балансировку в зависимости от портов на которые поступает трафик:

```
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: bookinfo-ratings-port
spec:
  host: ratings.prod.svc.cluster.local
  trafficPolicy: # Apply to all ports
    portLevelSettings:
    - port:
        number: 80
      loadBalancer:
        simple: LEAST_CONN
    - port:
        number: 9080
      loadBalancer:
        simple: ROUND_ROBIN
```

Virtual Service

Несколько полезных терминов для понимания контекста перенаправления трафика:

Сервис - единица поведения приложения, связанная с уникальным именем в реестре служб платформы. Сервис содержит некоторое количество сетевых конечных точек, привязанных к инстансам деплоймента (подам).

Версии сервиса (subsets) описанные в деплойментах различные версии приложения относящиеся к одному сервису.

Источник - клиент вызывающий сервис.

Хост - адрес используемый клиентом при попытке соединения с сервисом.

Модель доступа - Приложения обращаются только к целевому сервису (хосту) без знания отдельных версий (подмножеств) сервисов. Фактический выбор версии определяется сервисным прокси.

VirtualService - определяет набор правил для маршрутизации трафика, адресованного на определенный хост. Каждое правило имет критерии соответствия, по которым трафик будет отправлен в нужном направлении. Источник трафика так же может быть одним из критериев соответствия в правилах маршрутизации.

Следующий пример маршрутизирует HTTP трафик по умолчанию на поды сервиса reviews версии "version: v1", но для трафика содержащего URI с /wpcatalog/ или /consumercatalog/ будет выполнена перезапись значения URI на /newcatalog и трафик будет перенаправлен на поды с метками “version: v2”.

```
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: reviews-route
spec:
  hosts:
  - reviews.prod.svc.cluster.local
  http:
  - name: "reviews-v2-routes"
    match:
    - uri:
        prefix: "/wpcatalog"
    - uri:
        prefix: "/consumercatalog"
    rewrite:
      uri: "/newcatalog"
    route:
    - destination:
        host: reviews.prod.svc.cluster.local
        subset: v2
  - name: "reviews-v1-route"
    route:
    - destination:
        host: reviews.prod.svc.cluster.local
        subset: v1
```

Подмножества направления маршрутизации описаны в связанном с сервисом DestinationRule:

```
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: reviews-destination
spec:
  host: reviews.prod.svc.cluster.local
  subsets:
  - name: v1
```

```
    labels:
      version: v1
  - name: v2
    labels:
      version: v2
```

Service Entry

ServiceEntry включает дополнительные записи во внутренний реестр сервисов Service Mesh, так что бы система обнаружения сервисов имела доступ к дополнительно описанным сервисам.

ServiceEntry описывает свойство сервиса (DNS имя, порты, протоколы). Эти сервисы могут быть внешними как для Service Mesh так и внутренними.

Следующий пример описывает несколько внешних сервисов для доступа к ним от внутренних приложений по HTTPS. Сервисный прокси проверяет значение SNI в сообщении ClientHello что бы направить трафик на внешний сервис.

```
apiVersion: networking.istio.io/v1alpha3
kind: ServiceEntry
metadata:
  name: external-svc-https
spec:
  hosts:
  - api.dropboxapi.com
  - www.googleapis.com
  - api.facebook.com
  location: MESH_EXTERNAL
  ports:
  - number: 443
    name: https
    protocol: TLS
  resolution: DNS
```

События системного журнала

Системный журнал компонента Сервисный прокси содержит описание следующих событий:

- Access logs - поступление входящего запроса и отправка исходящего запроса
- Application logs - прикладные события сервисного прокси (ошибки соединения с компонентом POLM и т.п.)

Проверка системного журнала:

1. Авторизуйтесь в веб-консоли платформы, пользователь должен иметь права на чтение.
2. Выберете прикладное пространство имен подключенное к контрольной панели Synapse Service Mesh
3. Слева в меню Workloads выберите раздел Deployments
4. В списке выберите нужный Деплоймент приложения с подключенным сервисным прокси.
5. Перейдите на вкладку Pods
6. Перейдите на вкладку Logs

Пример Access logs

```
{"startTime": 1594730419955,
"namespace": "test.namespace",
"envoyId": "router~10.111.11.111~ingressgateway-df5755cc7-dkw9c.test.namespace~
test.namespace.svc.cluster.local",
"requestMethod": "POST",
"path": "/cardacctdinq",
"protocol": "HTTP11",
"responseCode": 200,
"userAgent": "Apache-HttpClient/4.5.8 (Java/11.0.3)",
"requestId": "14c6cfe4-8af7-94fd-b64c-c657d39a86ba",
"authority": "ingress.ca.ru",
"upstreamCluster": "outbound|8787|test-1|test.namespace.svc.cluster.local",
"upstreamHost": "10.111.11.111:8787",
"upstreamLocalAddress": "",
"upstreamTransportFailureReason": "",
"downstreamRemoteAddress": "10.111.1.1:52896",
"downstreamLocalAddress": "10.111.11.111:8080",
"routeName": "",
"responseFlags": "{}",
"xforwardedFor": "10.111.11.111,10.111.111.11,10.111.1.1"}
```


Пример Application logs


```
{
"level": "info",
"time": "1589871979",
"msg": "Envoy proxy is NOT ready: config not received from Pilot (is Pilot running?): cds
updates: 0
successful, 1 rejected; lds updates: 0 successful, 0 rejected",
"namespace": "test.namespace",
"pod": "egressgateway-8678c8797d-mxqzx"
}
```


События мониторинга

Компонент SVPX не интегрирован с компонентом "Объединенный мониторинг Unimon" Plaform V Monitor (MONA).

Для наблюдения за состоянием подов используйте существующие средства мониторинга Платформы, описанные в документации на конкретную платформу.

Часто встречающиеся проблемы и пути их устранения

Проблема	Пути решения
Сервисный прокси не может подключиться к компоненту POLM	В системном журнале видно сообщение Envoy proxy is NOT ready. Проверьте корректность подключения к контрольной панели Synapse Service Mesh. Обратитесь к системным администраторам контрольной панели Synapse Service Mesh

Проблема	Пути решения
Не доходят запросы до прикладного приложения	Проверьте Access логи сервисного прокси, возможны следующие варианты сообщений: - NR (No route configured): В конфигурации сервисного прокси отсутствует требуемый маршрут. Авторизуйтесь в прикладном namespace. Если вызов направлен на внутренний сервис, проверьте наличие сервиса с таким именем, проверьте корректность параметров хост/порт в конфигурационных файлах прикладного namespace - DestinationRule или VirtualService. Если вызов направлен на внешний хост проверьте наличие конфигурационного файла ServiceEntry для данного сочетания хост/порт. - UO (Upstream overflow with circuit breaking): Поставщик перегружен запросами. Авторизуйтесь в прикладном namespace. Проверьте корректность конфигурации раздела connectionPoolSettings в DestinationRule. - UF (Failed to connect to upstream): Поставщик сбросил соединение. Авторизуйтесь в прикладном namespace. Если вы используете автоматическую аутентификацию ISTIO_MUTUAL, проверьте наличие конфликта в разделе trafficPolicy конфигурационного файла DestinationRule относящегося к проблемному сервису, и раздела Spec/mtls конфигурационного файла peerAuthentication. В случае указания разных режимов работы в поле tls - возможны указанные ошибки. - UH (No healthy upstream): Поставщик неработоспособен. Авторизуйтесь в прикладном namespace. Проверьте наличие вызываемого сервиса - в веб-интерфейсе Home/Networking/Services/Search_by_name найдите сервис). Проверьте наличие запущенного пода на который ссылается сервис - в веб-интерфейсе кликните правой кнопкой мыши на найденный сервис, выберете вкладку pods на открывшейся странице, убедитесь что статус подов на данной странице имеет значение running

Руководство прикладного разработчика

Термины и определения

Термин/аббревиатура	Определение

Термин/аббревиатура	Определение
Istio-proxy	Sidecar-контейнер, предназначенный для маршрутизации трафика
Pod/Под	Набор контейнеров внутри узла кластера Kubernetes
Deployment/Деплоймент	Набор инструкций для запуска приложения в Kubernetes
Istio SE	Настраиваемая сервисная сетка с открытым исходным кодом, служащая для взаимодействия, мониторинга и обеспечения безопасности контейнеров в кластере Kubernetes
Контрольная панель	Проект, где запущены управляющие приложения Synapse Service Mesh (компонент POLM)
Управление политиками / POLM	Компонент Управление политиками из состава продукта Platform V Synapse Service Mesh
Platform V Synapse Service Mesh / SSM	Программный продукт на базе Istio SE, обеспечивающий возможность создания сервисной сети поверх Платформенной в Kubernetes
Граничный прокси / IGEG	Компонент Граничный прокси Platform V Synapse Service Mesh
Сервисный прокси / SVPX	Компонент Сервисный прокси Platform V Synapse Service Mesh

Системные требования

Для использования компонента Сервисный прокси из состава продукта Platform V Synapse Service Mesh необходим проект, подключенный к интеграционной платформе Synapse.

Подключение и конфигурирование

Чтобы запустить приложение с сервисным прокси, необходимо добавить его аннотацию в Деплоймент приложения.

```
apiVersion: apps/v1
kind: Deployment
spec:
  template:
    metadata:
      annotations:
    sidecar.istio.io/inject: 'true'
```

Миграция на текущую версию

После обновления контрольной панели требуется перезапуск Пода с контейнером прокси. Новая версия загрузится автоматически.

Разработка первого приложения с использованием программного продукта

Данный раздел не применим, так как SVPX используется как sidecar к пользовательскому приложению.

Использование программного продукта

Сервисный прокси используется для маршрутизации и обеспечения безопасности трафика между приложениями.

Часто встречающиеся проблемы и пути их устранения

Проблема	Пути решения
Сервисный прокси не может подключиться к компоненту POLM	В системном журнале видно сообщение Envoy proxy is NOT ready. Проверьте корректность подключения к контрольной панели Service Mesh. Обратитесь к системным администраторам контрольной панели.
Не доходят запросы до прикладного	Проверьте Access логи сервисного прокси, возможны следующие варианты сообщений: - NR (No route configured): В конфигурации сервисного прокси

Проблема	Пути решения
приложения	отсутствует требуемый маршрут. Авторизуйтесь в прикладном проекте. Если вызов направлен на внутренний сервис, проверьте наличие сервиса с таким именем, проверьте корректность параметров хост/порт в конфигурационных файлах прикладного проекта - DestinationRule или VirtualService. Если вызов направлен на внешний хост проверьте наличие конфигурационного файла ServiceEntry для данного сочетания хост/порт. - UO (Upstream overflow with circuit breaking): Поставщик перегружен запросами. Авторизуйтесь в прикладном проекте. Проверьте корректность конфигурации раздела connectionPoolSettings в DestinationRule. - UF (Failed to connect to upstream): Поставщик сбросил соединение. Авторизуйтесь в прикладном проекте. Если вы используете автоматическую аутентификацию ISTIO_MUTUAL, проверьте наличие конфликта в разделе trafficPolicy конфигурационного файла DestinationRule относящегося к проблемному сервису, и раздела Spec/mtls конфигурационного файла peerAuthentication. В случае указания разных режимов работы в поле tls - возможны указанные ошибки. - UH (No healthy upstream): Поставщик неработоспособен. Авторизуйтесь в прикладном проекте. Проверьте наличие вызываемого сервиса - в веб-интерфейсе Home/Networking/Services/Search_by_name найдите сервис). Проверьте наличие запущенного пода на который ссылается сервис - в веб-интерфейсе кликните правой кнопкой мыши на найденный сервис, выберете вкладку pods на открывшейся странице, убедитесь что статус подов на данной странице имеет значение running. Обратитесь к администраторам Synapse.
Контейнер с istio-proxy начинает часто перезагружаться из-за нехватки ресурсов	Чтобы увеличить ресурсы и переопределить настройки по умолчанию, добавьте аннотацию, устанавливающую лимит CPU и памяти в Деплоймент прикладного сервиса apiVersion: apps/v1 kind: Deployment spec: template: metadata: annotations: sidecar.istio.io/inject: 'true' sidecar.maistra.io/proxyCPULimit: 300m sidecar.maistra.io/proxyMemoryLimit: 300Mi

Руководство оператора

Термины и определения

Термин/аббревиатура	Определение
TLS	Transport Layer Security, протокол защиты транспортного уровня
istio-proxy	Сервисный прокси - Sidecar-контейнер, предназначенный для маршрутизации трафика
Pod/Под	Набор контейнеров внутри узла кластера Kubernetes
Платформа	Платформа оркестрации приложений с средствами автоматизации и управления на основе политик, например Kubernetes
Istio SE	Настраиваемая сервисная сетка с открытым исходным кодом, служащая для взаимодействия, мониторинга и обеспечения безопасности контейнеров в кластере Kubernetes
Контрольная панель	Проект, где запущены управляющие приложения Synapse Service Mesh (компонент POLM)
Управление политиками / POLM	Компонент Управление политиками из состава продукта Platform V Synapse Service Mesh
Platform V Synapse Service Mesh / SSM	Программный продукт на базе Istio SE, обеспечивающий возможность создания сервисной сети поверх Платформенной в Kubernetes
Граничный прокси / IGEG	Компонент Граничный прокси Platform V Synapse Service Mesh

Термин/аббревиатура	Определение
Сервисный прокси / SVPX	Компонент Сервисный прокси Platform V Synapse Service Mesh

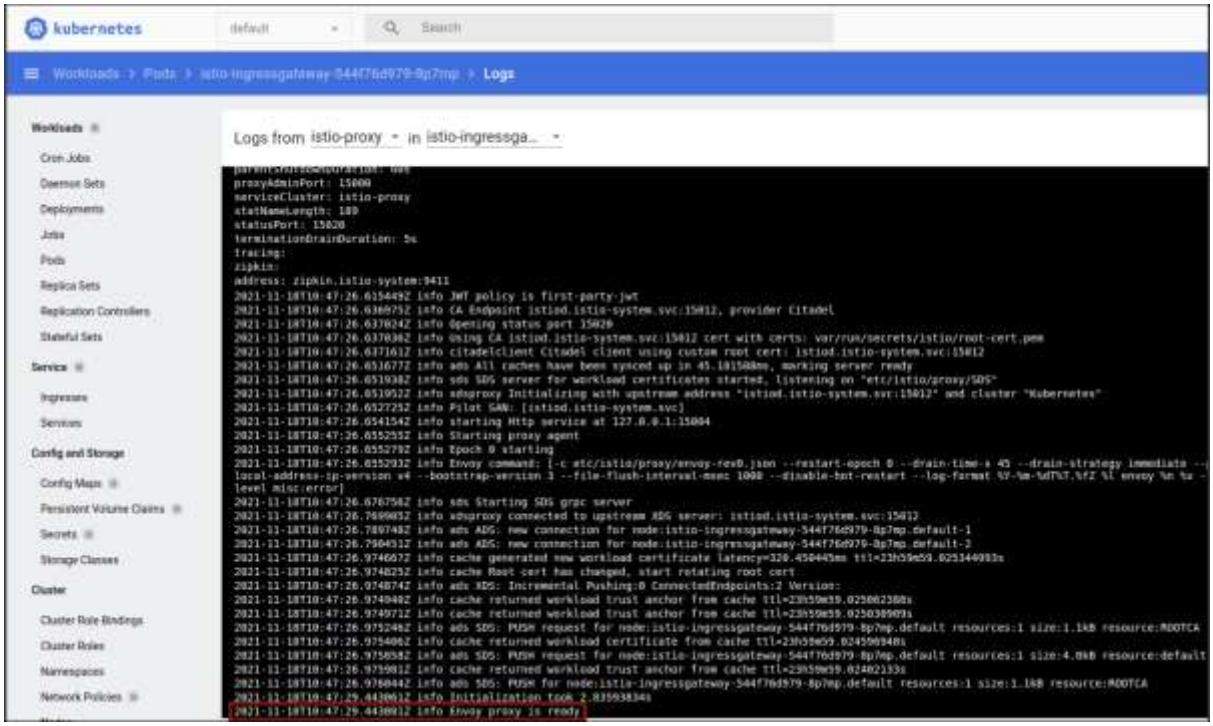
Доступ к приложению

Компонент Сервисный прокси не имеет пользовательского интерфейса. Он размещается в виде Sidecar-контейнера в каждом поде прикладного проекта, подключенного к Контрольной панели Synapse Service Mesh. В процессе эксплуатации оператор с правами на чтение не имеет доступа к управлению и настройке сервисного прокси. Оператор с правами на просмотр имеет доступ к записям системного журнала сервисного прокси.

Использование приложения

Сценарий 1. Просмотр журнала событий

1. Журнал сервисного прокси выводится в консоль контейнера, хранение его обеспечивается средствами Платформы.
2. Для просмотра журнала сервисного прокси необходимо в прикладном неймспейсе создать пользователя с правами не ниже view.
3. Слева в меню Workloads выберите раздел Pods.
4. В списке выберите нужный модуль и перейдите в него.
5. Перейдите на вкладку Logs.
6. Справа от надписи «Log from...» из выпадающего списка выберите контейнер istio-proxy.



Также можно сделать это через консоль.

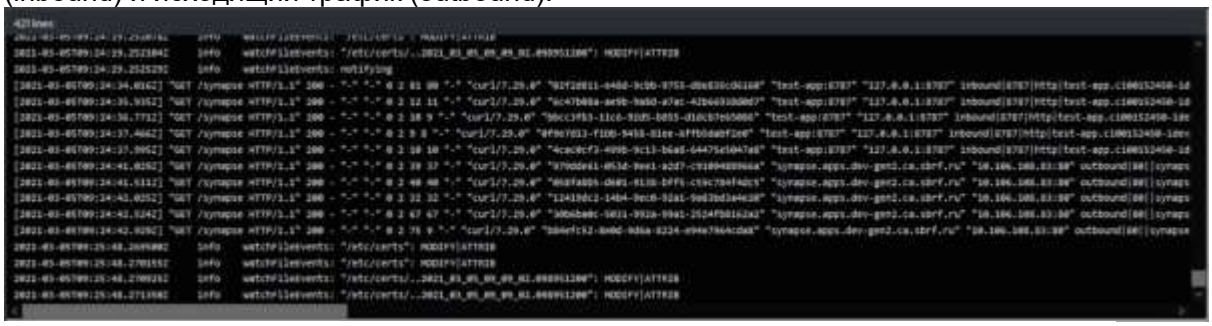
Для этого достаточно выполнить команду kubectl --kubeconfig=<путь_к_kubeconfig> -n <имя_проекта> logs <имя пода>

Пример:

```
user@myhost:istio$kubectl --kubeconfig=kubeconfig.json -n default logs istio-ingressgateway-544f76d979-8p7mp --tail=10
2021-11-18T10:47:26.974874Z     info    ads     XDS: Incremental Pushing:0 ConnectedEndpoints:2 Version:
2021-11-18T10:47:26.974940Z     info    cache   returned workload trust anchor from cache       ttl=23h59m59.025062388s
2021-11-18T10:47:26.974971Z     info    cache   returned workload trust anchor from cache       ttl=23h59m59.025030909s
2021-11-18T10:47:26.975246Z     info    ads     SDS: PUSH request for node:istio-ingressgateway-544f76d979-8p7mp.default resource
s:1 size:1.1kB resource:ROOTCA
2021-11-18T10:47:26.975406Z     info    cache   returned workload certificate from cache        ttl=23h59m59.024596948s
2021-11-18T10:47:26.975858Z     info    ads     SDS: PUSH request for node:istio-ingressgateway-544f76d979-8p7mp.default resource
s:1 size:4.0kB resource:default
2021-11-18T10:47:26.975981Z     info    cache   returned workload trust anchor from cache       ttl=23h59m59.02402133s
2021-11-18T10:47:26.976044Z     info    ads     SDS: PUSH for node:istio-ingressgateway-544f76d979-8p7mp.default resources:1 size
:1.1kB resource:ROOTCA
2021-11-18T10:47:29.443061Z     info    Initialization took 2.83593834s
2021-11-18T10:47:29.443081Z     info    Envoy proxy is ready
```

Сценарий 2. Выгрузка журнала событий

1. В правой части вкладки Logs нажмите Download, чтобы скачать журнал для анализа.
2. В журнале можно увидеть входящий в приложение с контейнером istio-proxy трафик (inbound) и исходящий трафик (outbound):



Примеры записей в журнале:

```
[2021-03-05T09:24:37.466Z] "GET /synapse HTTP/1.1" 200 - "-" "-" 0 2 9 8 "-"
"curl/7.29.0" "0f9e7d13-f1bb-9458-81ee-6ffb5dabf2e0" "test-app:8787"
"127.0.0.1:8787" inbound|8787|http|test-app.test.svc.cluster.local -
10.128.93.9:8787 10.128.214.19:51794 outbound_.8787_._.test-
app.test.svc.cluster.local default
[2021-03-05T09:24:41.511Z] "GET /synapse HTTP/1.1" 200 - "-" "-" 0 2 40 40 "-"
"curl/7.29.0" "058fa8b5-d601-9138-bff5-c59c784f4dc5" "synapse.test.ru"
"10.106.108.83:80" outbound|80||synapse.test.ru - 10.106.108.83:80
10.128.93.9:46110 - default
```

3. Пользователь с правами администратора при необходимости поиска проблем и определения неисправностей может воспользоваться расширенным функционалом сервисного прокси.
   Чтобы изменить уровень журналирования (например, установить trace), выполните команду:
   ```
   kubectl exec test-app-d568d4dc-v4nk7 -c istio-proxy --bash -c 'curl -X POST
   localhost:15000/logging?level=trace'
   ```
   Если нужно изменить уровень журналирования конкретного поля (например, rbac), выполните команду:
   ```
   kubectl exec test-app-d568d4dc-v4nk7 -c istio-proxy --bash -c 'curl -X POST
   localhost:15000/logging?rbac=debug'
   ```
   При необходимости можно скачать журнал в локальный файл. Для этого выполните команду:
   ```
   kubectl logs test-app-d568d4dc-v4nk7 -c istio-proxy > file.log
   ```

Имя test-app-d568d4dc-v4nk7 — это название Пода с сервисным прокси.

Часто встречающиеся проблемы и пути их устранения

В системном журнале можно увидеть различные флаги. Общие флаги ошибок сервисного прокси:

- NR (No route configured): В конфигурации сервисного прокси отсутствует требуемый маршрут. Авторизуйтесь в прикладном проекте. Если вызов направлен на внутренний сервис, проверьте наличие сервиса с таким именем, проверьте корректность параметров хост/порт в конфигурационных файлах прикладного проекта - DestinationRule или VirtualService. Если вызов направлен на внешний хост проверьте наличие конфигурационного файла ServiceEntry для данного сочетания хост/порт.
- UO (Upstream overflow with circuit breaking): Поставщик перегружен запросами. Авторизуйтесь в прикладном проекте. Проверьте корректность конфигурации раздела connectionPoolSettings в DestinationRule. UF (Failed to connect to upstream): Поставщик сбросил соединение. Авторизуйтесь в прикладном проекте. Если вы используете автоматическую аутентификацию ISTIO_MUTUAL, проверьте наличие конфликта в разделе trafficPolicy конфигурационного файла DestinationRule относящегося к проблемному сервису, и раздела Spec/mtls конфигурационного файла peerAuthentication. В случае указания разных режимов работы в поле tls - возможны указанные ошибки.
- UH (No healthy upstream): Поставщик неработоспособен. Авторизуйтесь в прикладном проекте. Проверьте наличие вызываемого сервиса - в веб-интерфейсе Home/Networking/Services/Search_by_name найдите сервис). Проверьте наличие запущенного пода на который ссылается сервис - в веб-интерфейсе кликните правой кнопкой мыши на найденный сервис, выберете вкладку pods на открывшейся странице, убедитесь что статус подов на данной странице имеет значение running. Обратитесь к администраторам Synapse Service Mesh.

Параметры настройки

В процессе эксплуатации пользователь с правами на чтение (оператор) не имеет доступа к управлению и настройке сервисного прокси.

Параметры настройки описаны в документации SVPX. Руководство администратора

Правила эксплуатации

Компонент SVPX, конфигурируется и эксплуатируется в соответствии с эксплуатационной документацией (Руководство по установке, Руководство по системному администрированию, Руководство оператора).

Оператор имеет доступ к записям системного журнала сервисного прокси.