**Диагностика istio-proxy (sidecar)**

Источник: <https://github.com/istio/istio/wiki/Troubleshooting-Istio>

Изменить уровень логирования envoy

* В рантайм: istioctl proxy-config log $POD --level=info|debug|trace
* При деплои пода, аннотацией: sidecar.istio.io/logLevel: "debug"
* Для всего service mesh, параметр в cfg Contr.plane: Spec.global.proxy.logLevel=debug

Получить конфигурацию и статистику с envoy

* Статистика по сервисам: oc exec $POD -c istio-proxy -- curl 'localhost:15000/stats' > stats
* Config Dump: oc exec $POD -c istio-proxy -- curl 'localhost:15000/config\_dump' > config\_dump.json
* Clusters Dump: oc exec $POD -c istio-proxy -- curl 'localhost:15000/clusters' > clusters
* Выгрузить логи envoy: oc logs $POD -c istio-proxy > proxy.log
* Просмотр логов в рантайм: oc logs -f $POD -c istio-proxy -n $NAMESPASE

Получить статус синхронизации конфигурации envoy

* istioctl proxy-status (сокращенно istioctl ps) – статус всех envoy в mesh
* istioctl proxy-status $POD.$NAMESPASE – статус одного envoy (например: istioctl proxy-status -n istio-system istio-egressgateway-59585c5b9c-ndc59.istio-system)

Пример вывода:

**PROXY CDS LDS EDS RDS PILOT VERSION**

details-v1-6dcc6fbb9d-wsjz4.default SYNCED SYNCED SYNCED SYNCED istio-pilot-75bdf98789-tfdvh 1.1.2

…

Получить текущую конфигурацию xDS envoy

* istioctl proxy-config cluster|listener|endpoint|route $POD -n $NAMESPASE

Пример вывода

**SERVICE FQDN PORT SUBSET DIRECTION TYPE**

BlackHoleCluster - - - STATIC

details.default.svc.cluster.local 9080 - outbound EDS

heapster.kube-system.svc.cluster.local 80 - outbound EDS

Получить конфигурацию авторизации Envoy envoy

* istioctl exp authz check $POD

Пример вывода:

**LISTENER[FilterChain] CERTIFICATE mTLS (MODE) JWT (ISSUERS) AuthZ (RULES)**

0.0.0.0\_80[0] none no (none) no (none) no (none)

0.0.0.0\_80[1] none no (none) no (none) no (none)

…

virtualInbound[3] /etc/certs/cert-chain.pem yes (STRICT) no (none) no (none)

virtualInbound[4] /etc/certs/cert-chain.pem yes (STRICT) no (none) no (none)

0.0.0.0\_15010[0] none no (none) no (none) no (none)

0.0.0.0\_15010[1] none no (none) no (none) no (none)

0.0.0.0\_15014[0] none no (none) no (none) no (none)

0.0.0.0\_15014[1] none no (none) no (none) no (none)

10.128.0.31\_8081 /etc/certs/cert-chain.pem yes (STRICT) no (none) no (none)

**Статус пода, включая настройки mTLS**

istioctl x describe pod rest-app-a-7cb6cff446-rdjfx

Пример вывода команды:

Pod: rest-app-a-7cb6cff446-rdjfx

Pod Ports: 8081 (jdbc-app-1), 15090 (istio-proxy)

--------------------

Service: rest-app-a

Port: http 8080/HTTP targets pod port 8081

DestinationRule: default.istio-system for "\*.local"

Traffic Policy TLS Mode: ISTIO\_MUTUAL

Pod is STRICT and clients are ISTIO\_MUTUAL

Exposed on Ingress Gateway http://192.168.130.11:8080

VirtualService: tutorial-rest-app-a-vs

/citiesCount\*, /ping\*

**Анализ производительности Istio**

Источник: <https://github.com/istio/istio/wiki/Analyzing-Istio-Performance>

**Возможные ошибки**

* Если применение конфигурации в Service mesh, не приводит к фактическим изменениям (например машрутизации), и команда istioctl proxy-status показывает, что все прокси синхронизированны, то возможно причина в конфликте примененных правил маршрутизации. Необходимо проверить логи пилота на наличие не пустых блоков ProxyStatus, которые свидетельствуют о том, что Pilot не может согласовать или применить конфигурацию для конкретного Envoy.
  + oc logs -l app=pilot -n istio-system -c discovery
* If Pilot doesn’t report any conflicts or other configuration issues, the proxies may be having a connection issue. You can check the log of the istio-proxy container in the source and destination pods for issues. If you don’t see anything helpful, you can increase the logging verbosity of the istio-proxy sidecar, which listens on port 15000 of the pod. (You may have to use kubectl port-forward to be able to connect to the sidecar.) Use a POST request against the proxy port to update the logging level: curl -s -XPOST http://localhost:15000/logging?level=debug
* Istio telemetry also collects the Envoy access logs, which include the connection response flags. ([See the list here](https://www.envoyproxy.io/docs/envoy/latest/configuration/observability/access_log.html" \l "command-operators) for RESPONSE\_FLAGS.) Use the command kubectl logs -l app=telemetry -n istio-system -c mixer to see the log entries if you’re using Mixer telemetry. If your cluster has a Prometheus instance configured to scrape Istio’s metrics, you can query that.

**Используемые ресурсы:**

- <https://www.stackrox.com/post/2019/11/how-to-make-istio-work-with-your-apps/>

- <https://github.com/istio/istio/wiki/Troubleshooting-Istio>