How HP set up secure and wise platform with Istio

John Zheng/ john.zheng@hp.com

#IstioCon

Agenda

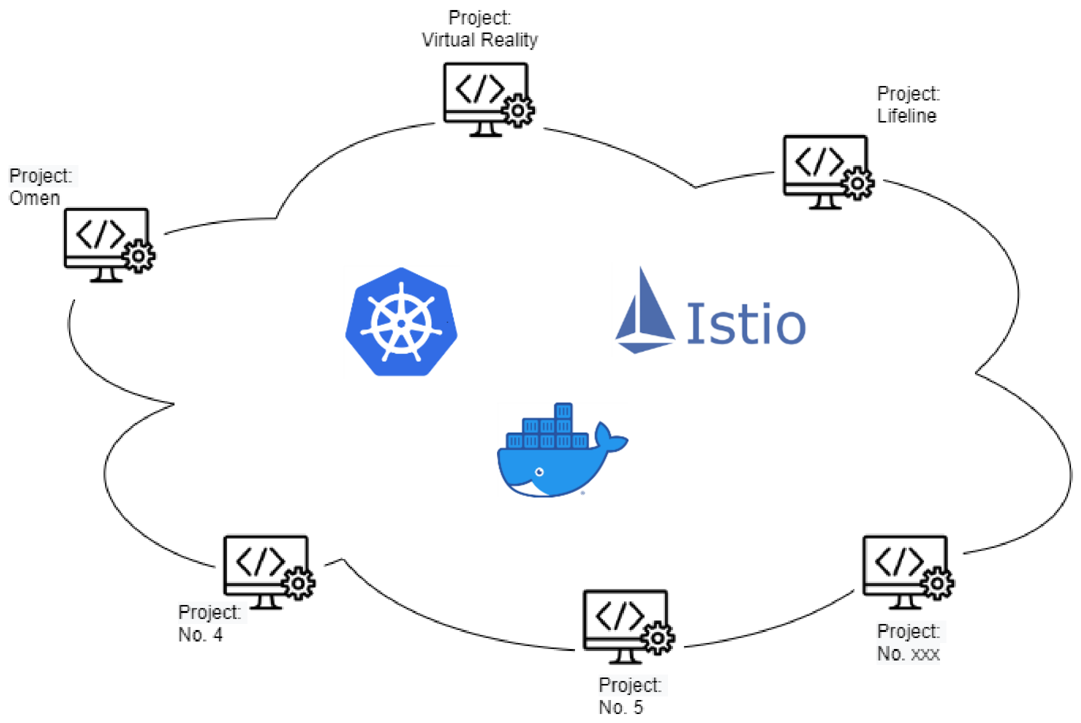
- *HP Horizon platform design with Istio*
- *Secure Platform*
- *Wise Platform*
- *Excellent Observability*
- *Q & A*

HP Horizon Platform design with Istio

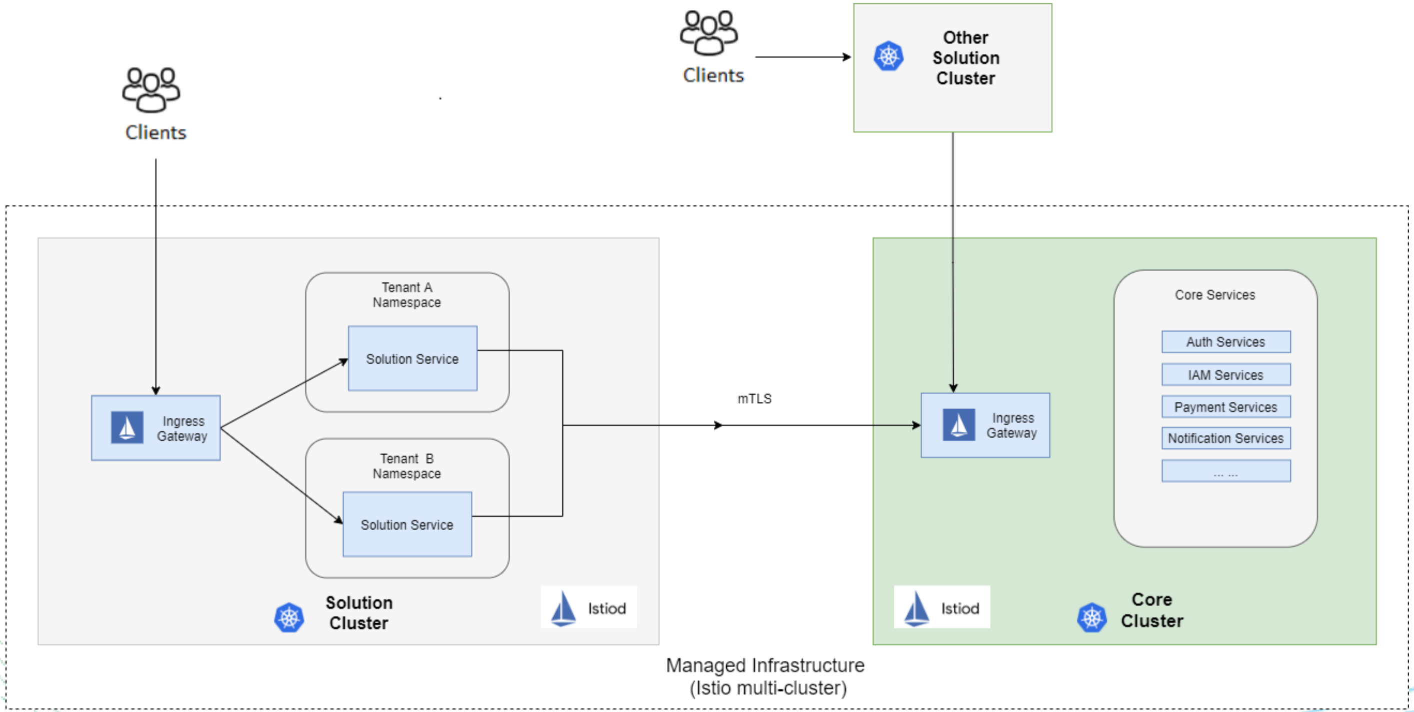
HP Horizon Platform

HP has lots of projects, deployed on cloud. They have common features, also have project specified feature.

We provide a common platform includes all common features, connect all projects with istio.



HP Horizon Platform Connect With Istio



Common services are in core cluster

Projects shared solution cluster

- Different namespace
- Project runs as tenant, need control rights

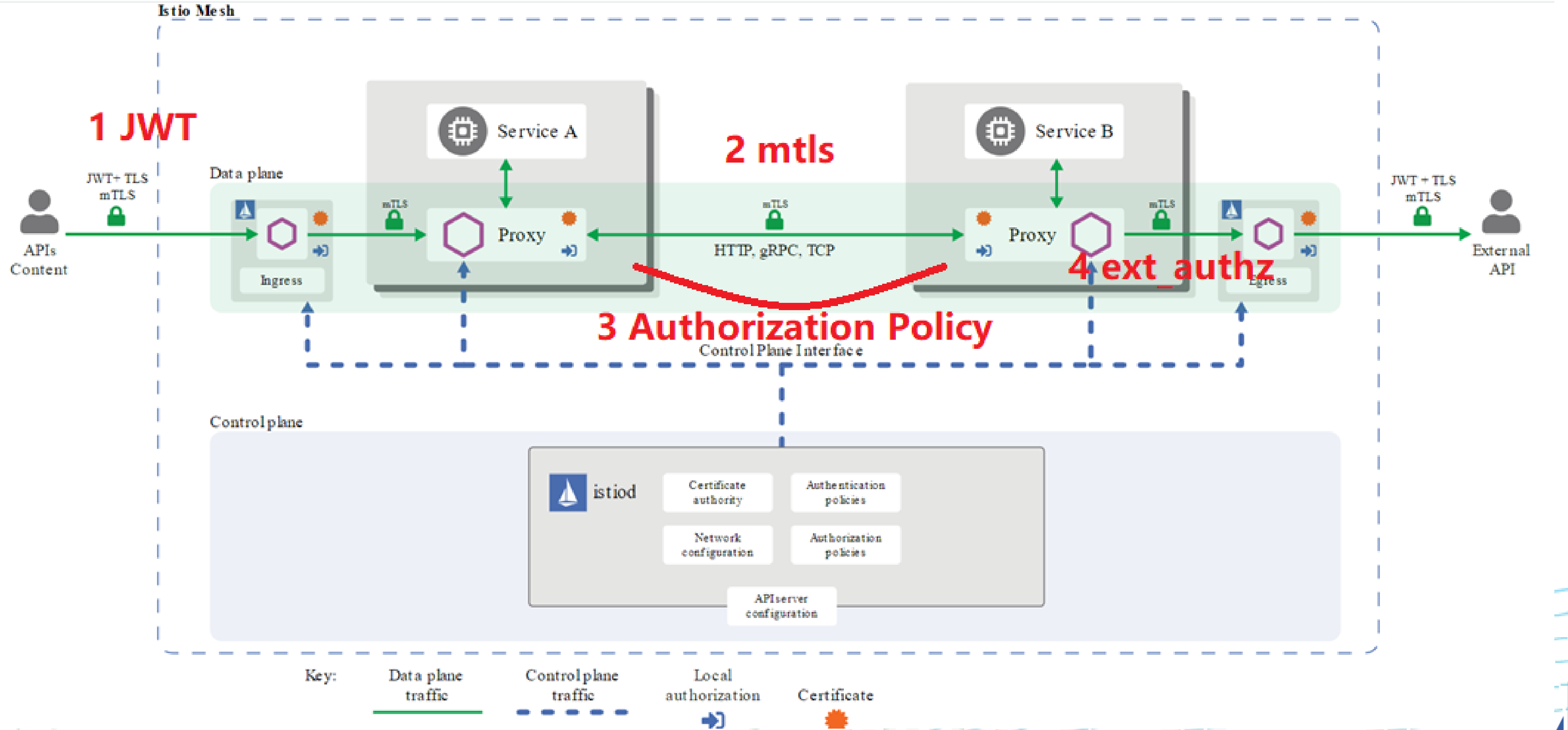
Solution cluster connect core cluster with Istio multi-cluster - Replicated control planes

Some standalone cluster without Istio can access core cluster also, as tenant.

Secure Platform

- *JWT Verify*
- *Mutual TLS*
- *Authorization Policy*
- *Envoy External Authorization*

Secure Platform

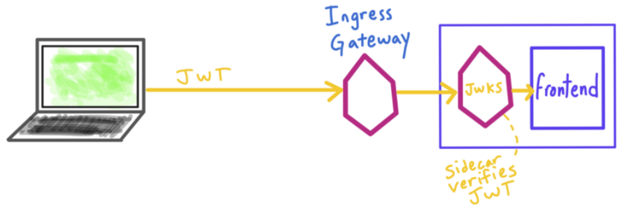


#IstioCon

Secure Platform – JWT Verify

Using request authentication policy to

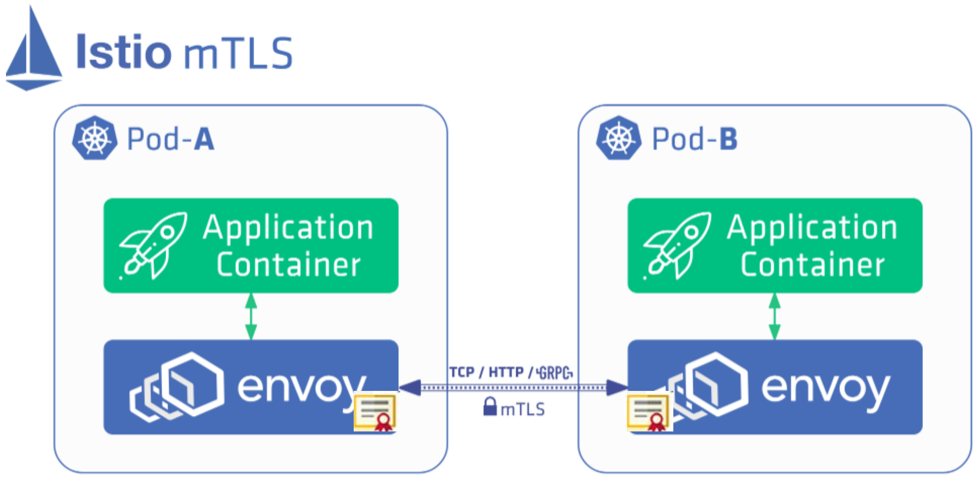
Verify end-user JWT easily



```
$ kubectl apply -f - <<EOF
apiVersion: "security.istio.io/v1beta1"
kind: "RequestAuthentication"
metadata:
  name: "jwt-example"
  namespace: istio-system
spec:
  selector:
    matchLabels:
      istio: ingressgateway
  jwtRules:
  - issuer: "testing@secure.istio.io"
    jwksUri: "https://raw.githubusercontent.com/istio/istio/release-1.6/security/tools/jwt/samples/jwks.json"
EOF
```

Secure Platform – mutual TLS

Using **mutual TLS** for service-to-service authentication.

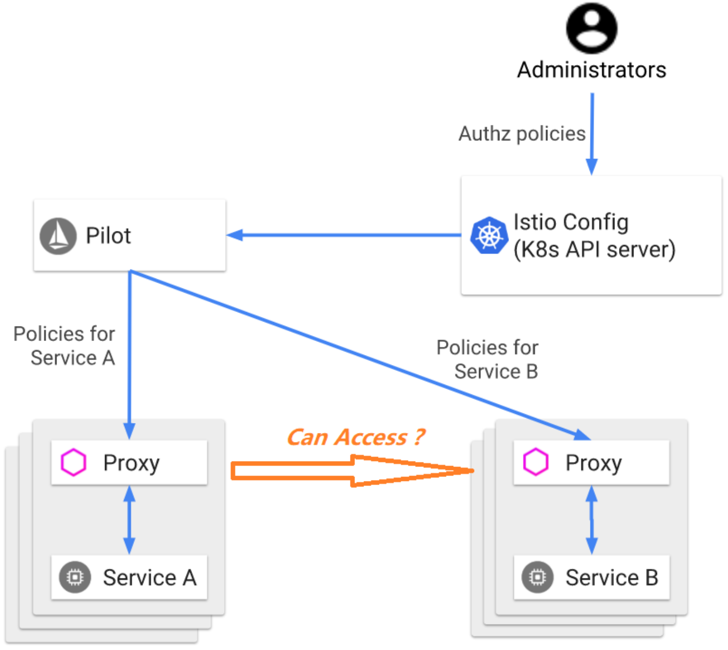


- When a service receives or sends network traffic, the traffic always goes through the Envoy proxies first.
- When mTLS is enabled between two services, the client side and server side's "envoy proxies" verify each other's identities before sending requests.
- If the verification is successful, then the client-side proxy encrypts the traffic, and sends it to the server-side proxy.
- The server-side proxy decrypts the traffic and forwards it locally to the actual destination service.

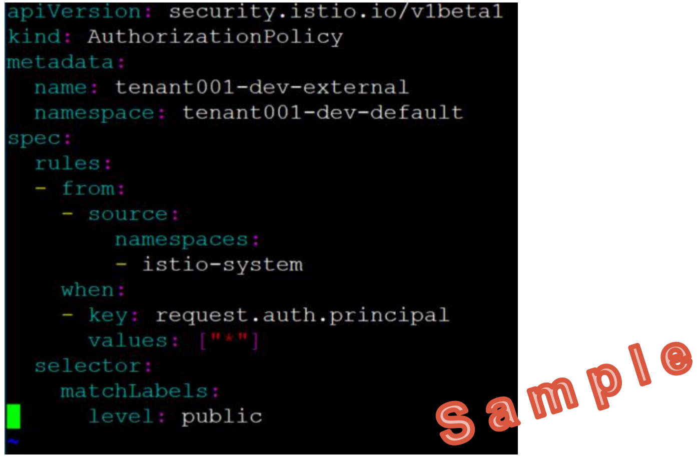
Secure Platform – Authorization Policy

Using **Authorization Policy**

enables access control on workloads in the mesh.

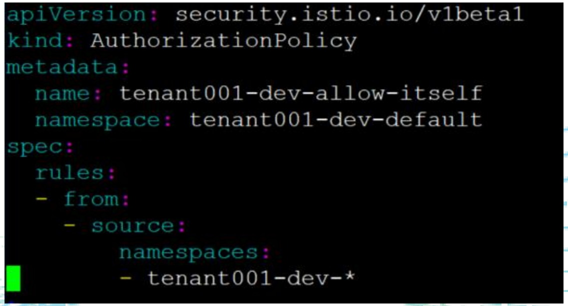


For request from ingressgateway, need verify token



```
apiVersion: security.istio.io/v1beta1
kind: AuthorizationPolicy
metadata:
  name: tenant001-dev-external
  namespace: tenant001-dev-default
spec:
  rules:
  - from:
    - source:
        namespaces:
        - istio-system
    when:
    - key: request.auth.principal
      values: ["*"]
  selector:
    matchLabels:
      level: public
```

For request from same tenant, allow
For request from another tenant, not allow

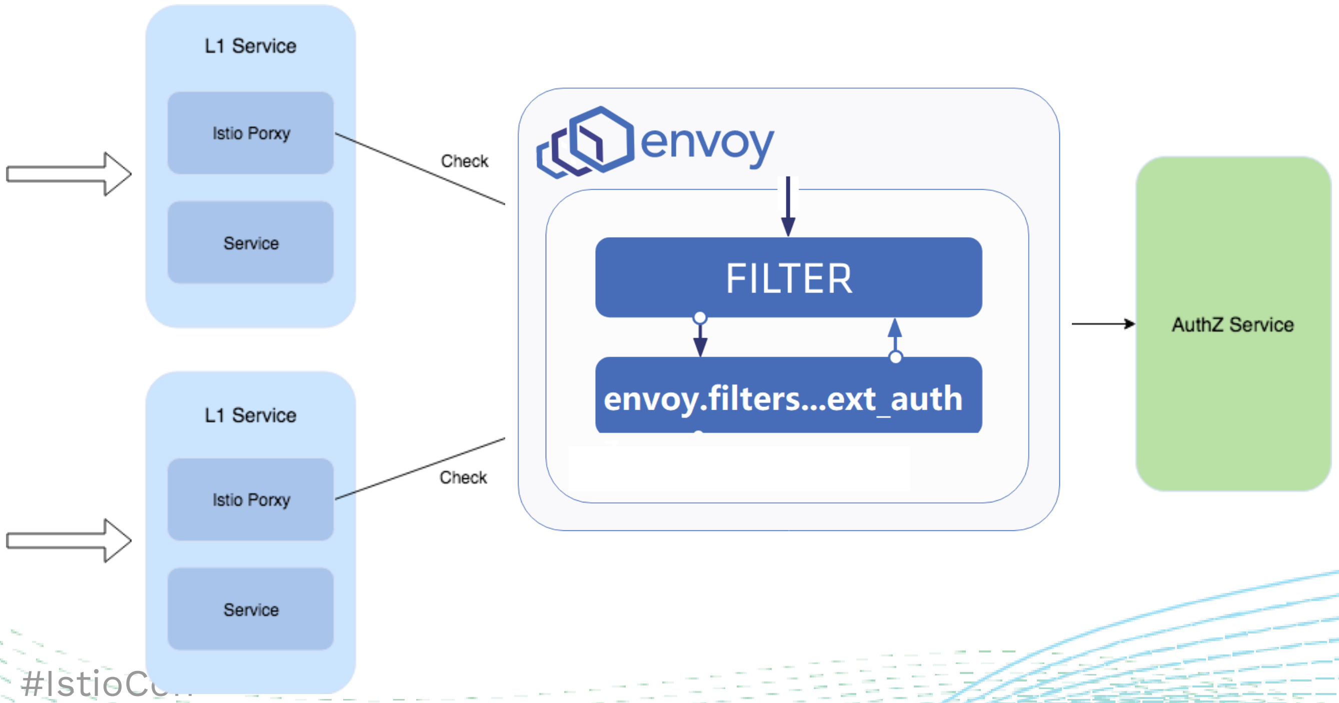


```
apiVersion: security.istio.io/v1beta1
kind: AuthorizationPolicy
metadata:
  name: tenant001-dev-allow-itself
  namespace: tenant001-dev-default
spec:
  rules:
  - from:
    - source:
        namespaces:
        - tenant001-dev-*
```

Secure Platform – Extra Authorization

Version 1 : Istio Mixer authz adapt => Version 2: Envoyfilter ext_authz

Implement role-based authorization – whether this user can access this api based on its role

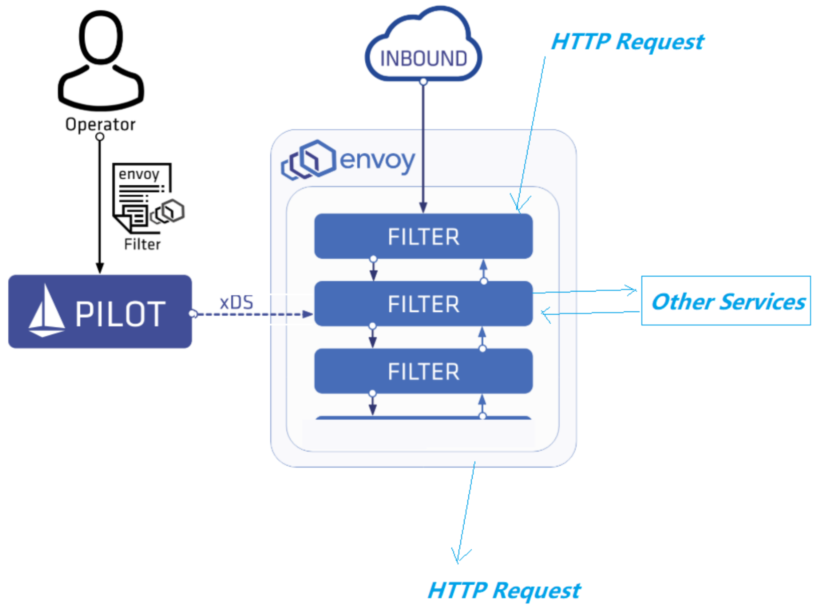


Wise Platform

Wise Platform

Using **envoy filter** to handle things from platform level, reduce workload of developers**.**

EnvoyFilter provides a mechanism to customize the Envoy configuration generated by Istio Pilot.
Use EnvoyFilter to modify values for certain fields, add specific filters, or even add entirely new listeners, clusters, etc.



Wise Platform

K8s custom resource definition

HTTP filters

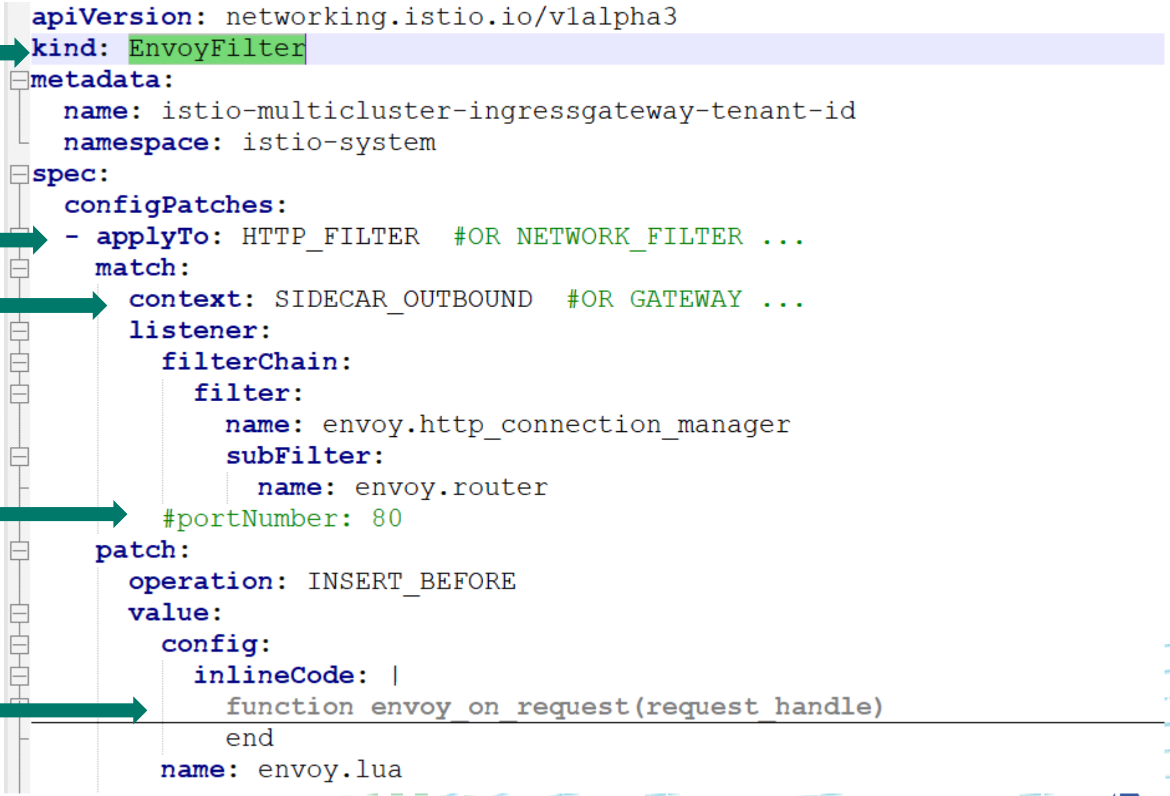
Network filters

UDP listener filters ...

Match outbound listeners in all sidecars
Or Istio gateway

Which port number the filter will apply to

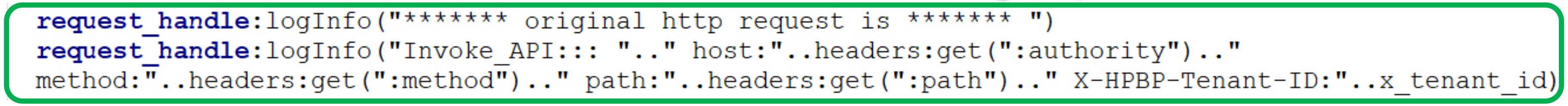
The Lua code that Envoy will execute.



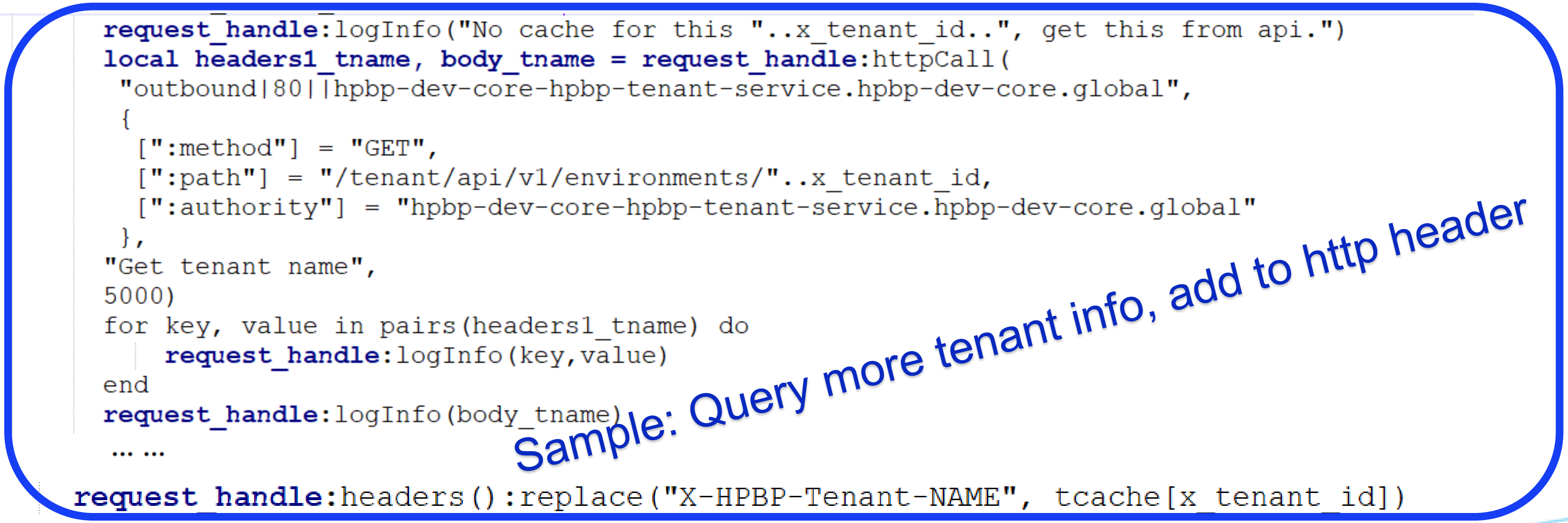
```
apiVersion: networking.istio.io/v1alpha3
kind: EnvoyFilter
metadata:
  name: istio-multicluster-ingressgateway-tenant-id
  namespace: istio-system
spec:
  configPatches:
  - applyTo: HTTP_FILTER  #OR NETWORK_FILTER ...
    match:
      context: SIDECAR_OUTBOUND  #OR GATEWAY ...
      listener:
        filterChain:
          filter:
            name: envoy.http_connection_manager
            subFilter:
              name: envoy.router
        #portNumber: 80
    patch:
      operation: INSERT_BEFORE
      value:
        config:
          inlineCode: |
            function envoy_on_request(request_handle)
            end
        name: envoy.lua
```

Wise Platform – lua

Sample: print api info into log



```lua
request_handle:logInfo("******* original http request is ******* ")
request_handle:logInfo("Invoke_API::: ".." host:"..headers:get(":authority").."
method:"..headers:get(":method").." path:"..headers:get(":path").." X-HPBP-Tenant-ID:"..x_tenant_id)
```



```lua
  request_handle:logInfo("No cache for this "..x_tenant_id..", get this from api.")
  local headers1_tname, body_tname = request_handle:httpCall(
   "outbound|80||hpbp-dev-core-hpbp-tenant-service.hpbp-dev-core.global",
   {
    [":method"] = "GET",
    [":path"] = "/tenant/api/v1/environments/"..x_tenant_id,
    [":authority"] = "hpbp-dev-core-hpbp-tenant-service.hpbp-dev-core.global"
   },
  "Get tenant name",
  5000)
  for key, value in pairs(headers1_tname) do
      request_handle:logInfo(key,value)
  end
  request_handle:logInfo(body_tname)
   … …

request_handle:headers():replace("X-HPBP-Tenant-NAME", tcache[x_tenant_id])
```

Sample: Query more tenant info, add to http header

Wise Platform

Using envoyfilter to implement requirements on platform level, reduces application workload.

Intelligence Platform for Multiple Tenant Support

- Support multi-tenants (Add extra http header/ logs wisely)
- Verify whether JWT token in blacklist or not
- Different Rate Limits for each tenant

… …

Excellent Observability

Excellent Observability

Istio generates detailed telemetry for all service communications within a mesh. This telemetry provides *observability* of service behavior, empowering operators to troubleshoot, maintain, and optimize their applications – **without imposing any additional burdens on service developers**.
Through Istio, operators gain a thorough understanding of how monitored services are interacting, both with other services and with the Istio components themselves.

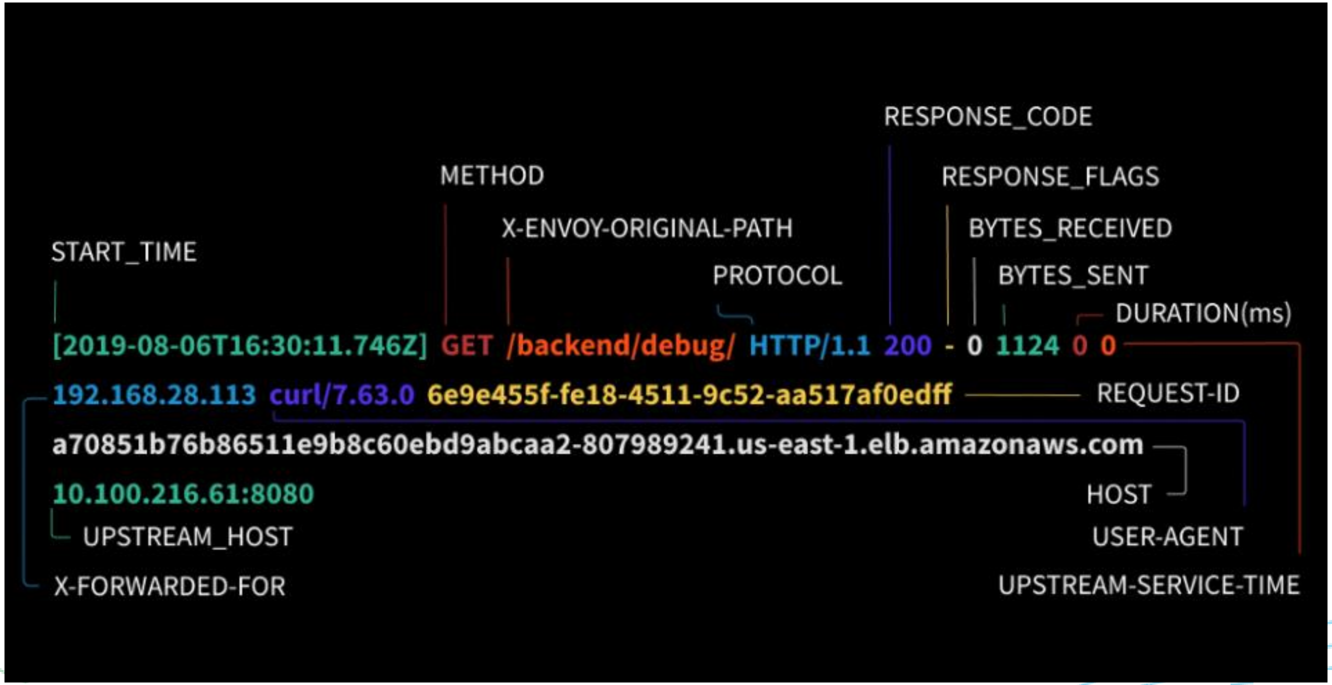
[Metrics]

[Distributed Traces]

[Access Logs]

Excellent Observability

Istio(envoy) can generate access logs for service traffic in a configurable set of formats



```
[2019-08-06T16:30:11.746Z] GET /backend/debug/ HTTP/1.1 200 - 0 1124 0 0
192.168.28.113 curl/7.63.0 6e9e455f-fe18-4511-9c52-aa517af0edff
a70851b76b86511e9b8c60ebd9abcaa2-807989241.us-east-1.elb.amazonaws.com
10.100.216.61:8080
```

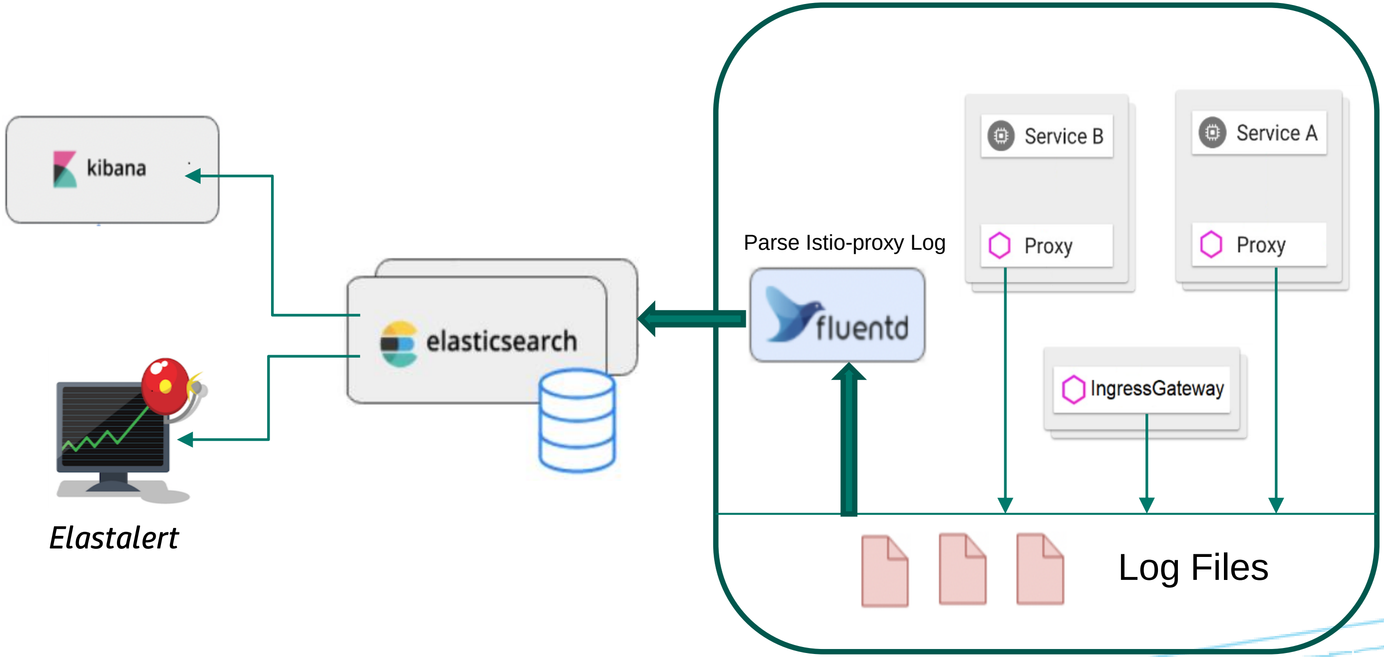
Excellent Observability - Access logs

- Each API Access Count
- Each API Fail Rate
- Each API Latency

Easy to debug

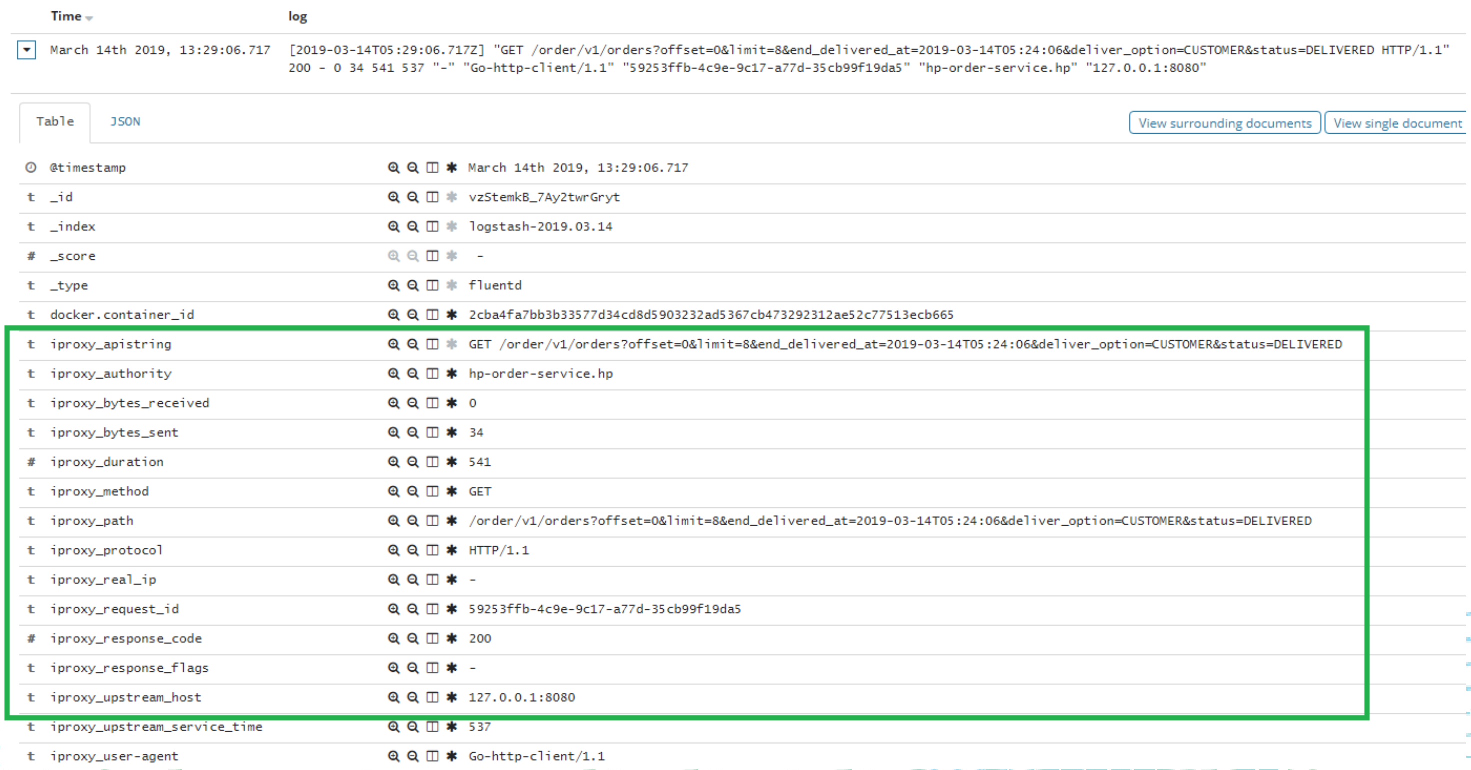
Easy to report

Easy to alert



Excellent Observability - Access logs

Istio-proxy log showed in kibana after parse

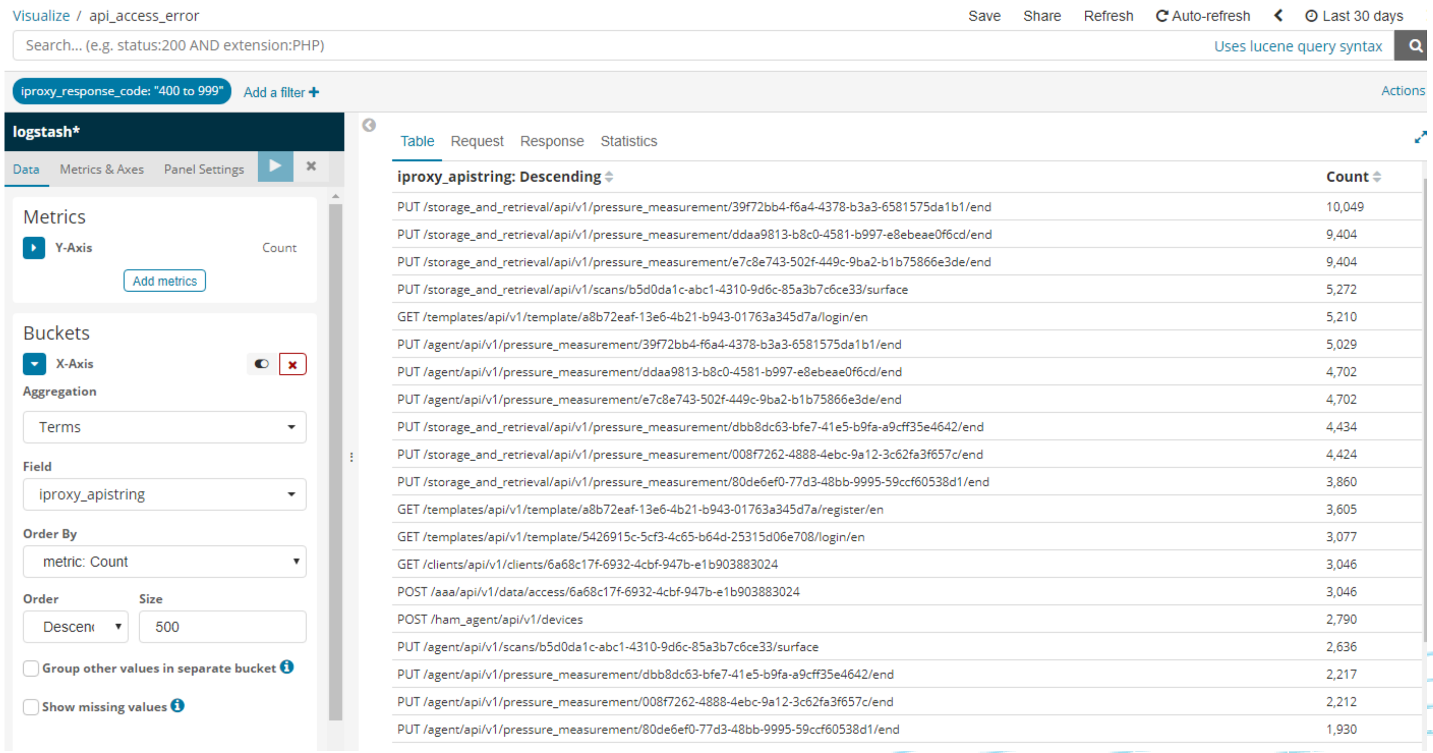


Time	log
March 14th 2019, 13:29:06.717	[2019-03-14T05:29:06.717Z] "GET /order/v1/orders?offset=0&limit=8&end_delivered_at=2019-03-14T05:24:06&deliver_option=CUSTOMER&status=DELIVERED HTTP/1.1" 200 - 0 34 541 537 "-" "Go-http-client/1.1" "59253ffb-4c9e-9c17-a77d-35cb99f19da5" "hp-order-service.hp" "127.0.0.1:8080"

	@timestamp	March 14th 2019, 13:29:06.717
t	_id	vzStemkB_7Ay2twrGryt
t	_index	logstash-2019.03.14
#	_score	-
t	_type	fluentd
t	docker.container_id	2cba4fa7bb3b33577d34cd8d5903232ad5367cb473292312ae52c77513ecb665
t	iproxy_apistring	GET /order/v1/orders?offset=0&limit=8&end_delivered_at=2019-03-14T05:24:06&deliver_option=CUSTOMER&status=DELIVERED
t	iproxy_authority	hp-order-service.hp
t	iproxy_bytes_received	0
t	iproxy_bytes_sent	34
#	iproxy_duration	541
t	iproxy_method	GET
t	iproxy_path	/order/v1/orders?offset=0&limit=8&end_delivered_at=2019-03-14T05:24:06&deliver_option=CUSTOMER&status=DELIVERED
t	iproxy_protocol	HTTP/1.1
t	iproxy_real_ip	-
t	iproxy_request_id	59253ffb-4c9e-9c17-a77d-35cb99f19da5
#	iproxy_response_code	200
t	iproxy_response_flags	-
t	iproxy_upstream_host	127.0.0.1:8080
t	iproxy_upstream_service_time	537
t	iproxy_user-agent	Go-http-client/1.1

Excellent Observability - Access logs

API Error
In last 30 days



iproxy_response_code: "400 to 999"

logstash*

Metrics

Y-Axis Count

Buckets

X-Axis

Aggregation

Terms

Field

iproxy_apistring

Order By

metric: Count

Order Size

Descend 500

Group other values in separate bucket

Show missing values

iproxy_apistring: Descending	Count
PUT /storage_and_retrieval/api/v1/pressure_measurement/39f72bb4-f6a4-4378-b3a3-6581575da1b1/end	10,049
PUT /storage_and_retrieval/api/v1/pressure_measurement/ddaa9813-b8c0-4581-b997-e8ebeae0f6cd/end	9,404
PUT /storage_and_retrieval/api/v1/pressure_measurement/e7c8e743-502f-449c-9ba2-b1b75866e3de/end	9,404
PUT /storage_and_retrieval/api/v1/scans/b5d0da1c-abc1-4310-9d6c-85a3b7c6ce33/surface	5,272
GET /templates/api/v1/template/a8b72eaf-13e6-4b21-b943-01763a345d7a/login/en	5,210
PUT /agent/api/v1/pressure_measurement/39f72bb4-f6a4-4378-b3a3-6581575da1b1/end	5,029
PUT /agent/api/v1/pressure_measurement/ddaa9813-b8c0-4581-b997-e8ebeae0f6cd/end	4,702
PUT /agent/api/v1/pressure_measurement/e7c8e743-502f-449c-9ba2-b1b75866e3de/end	4,702
PUT /storage_and_retrieval/api/v1/pressure_measurement/dbb8dc63-bfe7-41e5-b9fa-a9cff35e4642/end	4,434
PUT /storage_and_retrieval/api/v1/pressure_measurement/008f7262-4888-4ebc-9a12-3c62fa3f657c/end	4,424
PUT /storage_and_retrieval/api/v1/pressure_measurement/80de6ef0-77d3-48bb-9995-59ccf60538d1/end	3,860
GET /templates/api/v1/template/a8b72eaf-13e6-4b21-b943-01763a345d7a/register/en	3,605
GET /templates/api/v1/template/5426915c-5cf3-4c65-b64d-25315d06e708/login/en	3,077
GET /clients/api/v1/clients/6a68c17f-6932-4cbf-947b-e1b903883024	3,046
POST /aaa/api/v1/data/access/6a68c17f-6932-4cbf-947b-e1b903883024	3,046
POST /ham_agent/api/v1/devices	2,790
PUT /agent/api/v1/scans/b5d0da1c-abc1-4310-9d6c-85a3b7c6ce33/surface	2,636
PUT /agent/api/v1/pressure_measurement/dbb8dc63-bfe7-41e5-b9fa-a9cff35e4642/end	2,217
PUT /agent/api/v1/pressure_measurement/008f7262-4888-4ebc-9a12-3c62fa3f657c/end	2,212
PUT /agent/api/v1/pressure_measurement/80de6ef0-77d3-48bb-9995-59ccf60538d1/end	1,930

Thank you!

WeChat: johnzhengaz
Github: johnzheng1975