Solace PubSub+ Distributed Tracing using self-managed OTEL Collector with Jaeger, Dynatrace, New Relic and more…

Table of Contents

1 Purpose

Describe how to get OpenTelemetry (OTEL) traces from a simple application chain to Jaeger and/or other observability Cloud solutions using standard self-managed versions of OTEL collector and Jaeger. This document and accompanying code can be found online at https://github.com/SolaceLabs/solace-demo-observability

1.1 Application chain

Solace SDKPerf ***publisher*** – Solace Broker ***topic*** – Solace Broker ***queue1*** and ***queue2*** – Solace SDKPerf ***consumer***

1.2 Reference

Solace PubSub+ Distributed Tracing is an additional option for Solace PubSub+ and SAP AEM event brokers. The Distributed Tracing part of this demo is based on work from my colleague Daniel Brunold (https://github.com/dabgmx, also well-known for his work on the Solace Prometheus Exporter, see https://github.com/solacecommunity/solace-prometheus-exporter)
For this demo Daniel got some good inspiration from the Solace Codelabs 'Getting Started with Solace Distributed Tracing and Context Propagation' at https://codelabs.solace.dev/codelabs/dt-otel/
In addition, you can also use Solace Syslog Forwarding to ingest Event, System and Command logs information.

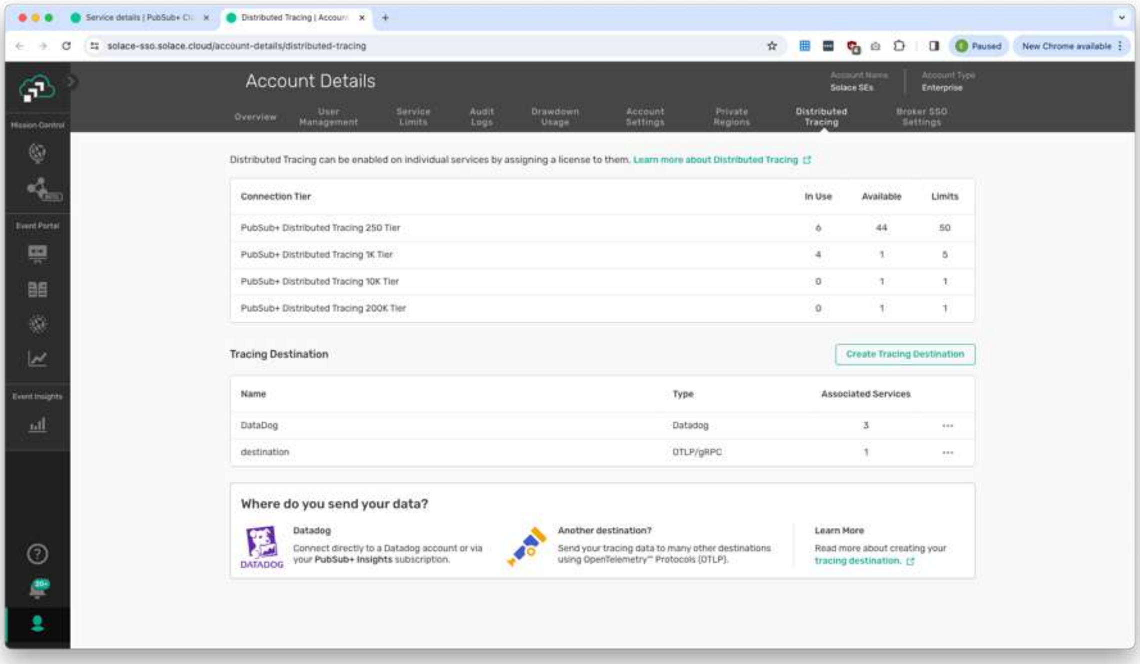
2 Setup

2.1 Prerequisites

A Solace broker running in the PubSub+ Cloud platform or self-managed.
Java installed if you want to use direct installations of SDKPerf and OpenSearch.

2.2 Configuration

For Solace PubSub+ Cloud brokers Distributed Tracing can be enabled on individual services by assigning a license to them. This license is available for multiple Connection Tiers.



Distributed Tracing can be enabled on individual services by assigning a license to them. Learn more about Distributed Tracing

Connection Tier	In Use	Available	Limits
PubSub+ Distributed Tracing 250 Tier	6	44	50
PubSub+ Distributed Tracing 1K Tier	4	1	5
PubSub+ Distributed Tracing 10K Tier	0	1	1
PubSub+ Distributed Tracing 200K Tier	0	1	1

Tracing Destination

Name	Type	Associated Services	
DataDog	Datadog	3	•••
destination	OTLP/gRPC	1	•••

Where do you send your data?

Datadog
Connect directly to a Datadog account or via your **PubSub+ Insights** subscription.

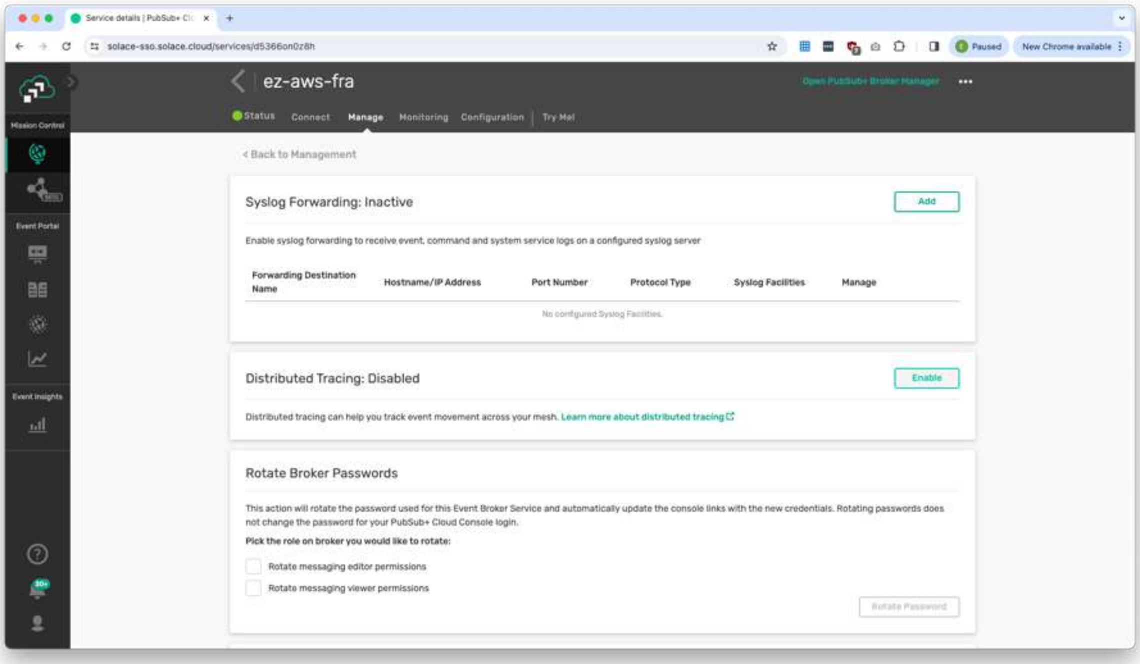
Another destination?
Send your tracing data to many other destinations using OpenTelemetry™ Protocols (OTLP).

Learn More
Read more about creating your tracing destination.

In this demo you will enable Distributed Tracing but you will not use the [Deploy Configuration] option to deploy a managed OTEL collector as you will use a self-managed OTEL collector with local Jaeger and optional other destinations.

In Mission Control go to Cluster Manager and select the service where you want to use Distributed Tracing. In the service overview click Manage then Advanced Options. In the

Distributed Tracing section click [Enable]



Syslog Forwarding: Inactive

Enable syslog forwarding to receive event, command and system service logs on a configured syslog server

Forwarding Destination Name	Hostname/IP Address	Port Number	Protocol Type	Syslog Facilities	Manage
No configured Syslog Facilities.					

Distributed Tracing: Disabled

Distributed tracing can help you track event movement across your mesh. Learn more about distributed tracing

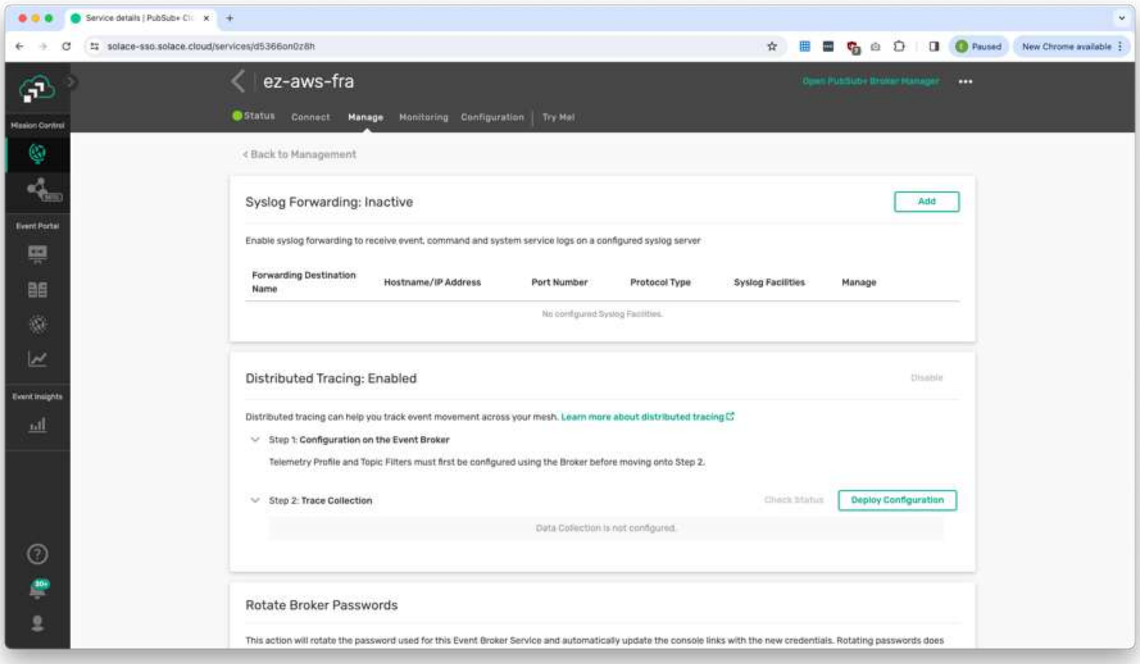
Rotate Broker Passwords

This action will rotate the password used for this Event Broker Service and automatically update the console links with the new credentials. Rotating passwords does not change the password for your PubSub+ Cloud Console login.

Pick the role on broker you would like to rotate:

- [ ] Rotate messaging editor permissions
- [ ] Rotate messaging viewer permissions

Do not click [Deploy Configuration] as you will setup your own OTEL collector in this demo.



Syslog Forwarding: Inactive

Enable syslog forwarding to receive event, command and system service logs on a configured syslog server

Forwarding Destination Name	Hostname/IP Address	Port Number	Protocol Type	Syslog Facilities	Manage
No configured Syslog Facilities.					

Distributed Tracing: Enabled

Distributed tracing can help you track event movement across your mesh. Learn more about distributed tracing

Step 1: **Configuration on the Event Broker**

Telemetry Profile and Topic Filters must first be configured using the Broker before moving onto Step 2.

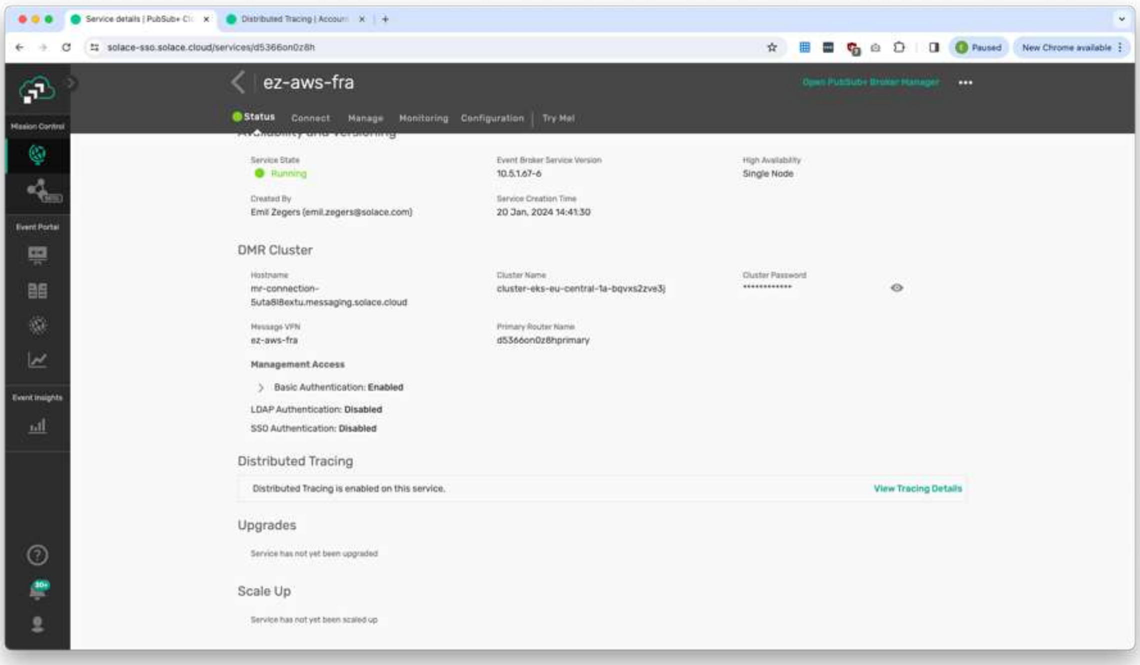
Step 2: **Trace Collection**

Data Collection is not configured.

Rotate Broker Passwords

This action will rotate the password used for this Event Broker Service and automatically update the console links with the new credentials. Rotating passwords does

When enabled you'll find a section Distributed Tracing in the Status overview.



Service State: Running
Event Broker Service Version: 10.5.1.67-6
High Availability: Single Node
Created By: Emil Zegers (emil.zegers@solace.com)
Service Creation Time: 20 Jan, 2024 14:41:30

DMR Cluster

Hostname: mr-connection-5uta8i8extu.messaging.solace.cloud
Cluster Name: cluster-eks-eu-central-1a-bqvxs2zve3j
Cluster Password: *************
Message VPN: ez-aws-fra
Primary Router Name: d5366on0z8hprimary

Management Access
Basic Authentication: **Enabled**
LDAP Authentication: **Disabled**
SSO Authentication: **Disabled**

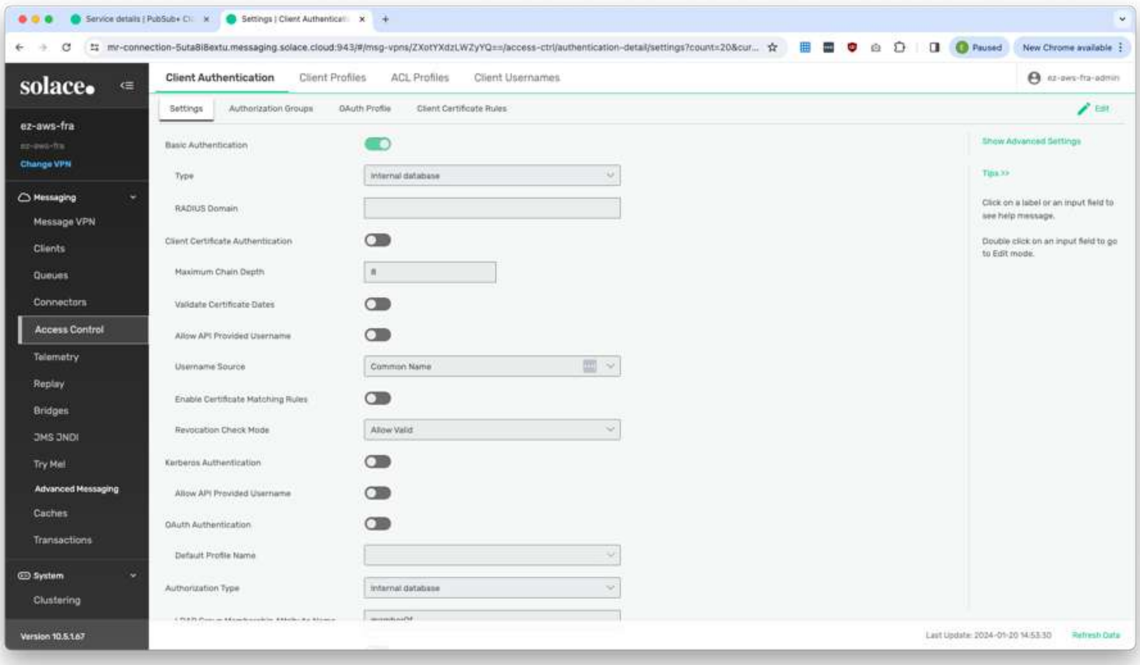
Distributed Tracing
Distributed Tracing is enabled on this service. View Tracing Details

Upgrades
Service has not yet been upgraded

Scale Up
Service has not yet been scaled up

2.3 Configure Broker

Open PubSub+ Broker Manager and verify under Access Control that Basic Authentication is enabled and set to Type Internal database.



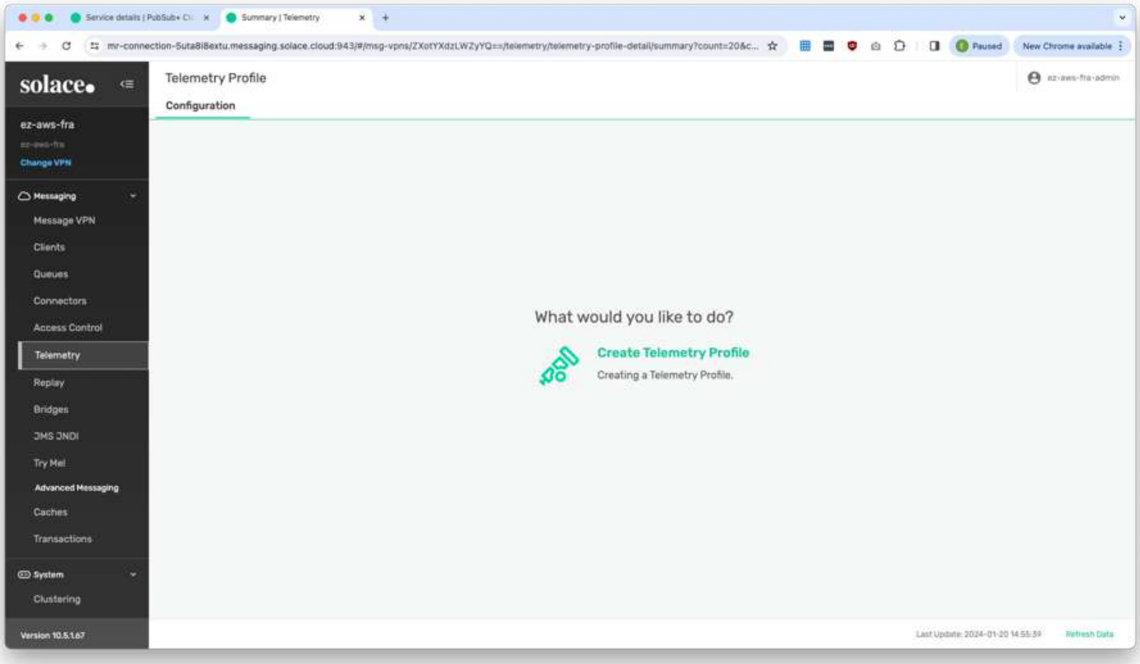
Basic Authentication	On
Type	Internal database
RADIUS Domain	
Client Certificate Authentication	Off
Maximum Chain Depth	8
Validate Certificate Dates	Off
Allow API Provided Username	Off
Username Source	Common Name
Enable Certificate Matching Rules	Off
Revocation Check Mode	Allow Valid
Kerberos Authentication	Off
Allow API Provided Username	Off
OAuth Authentication	Off
Default Profile Name	
Authorization Type	Internal database

Click on a label or an input field to see help message.

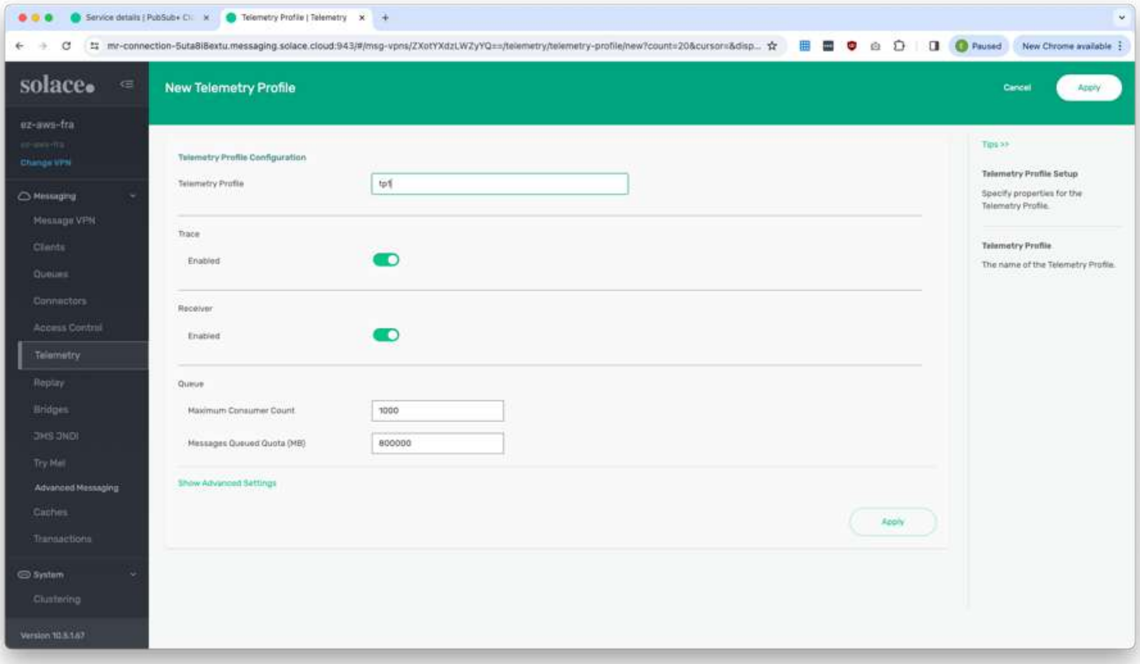
Double click on an input field to go to Edit mode.

Last Update: 2024-01-20 14:53:30

At Telemetry click Create a Telemetry Profile.

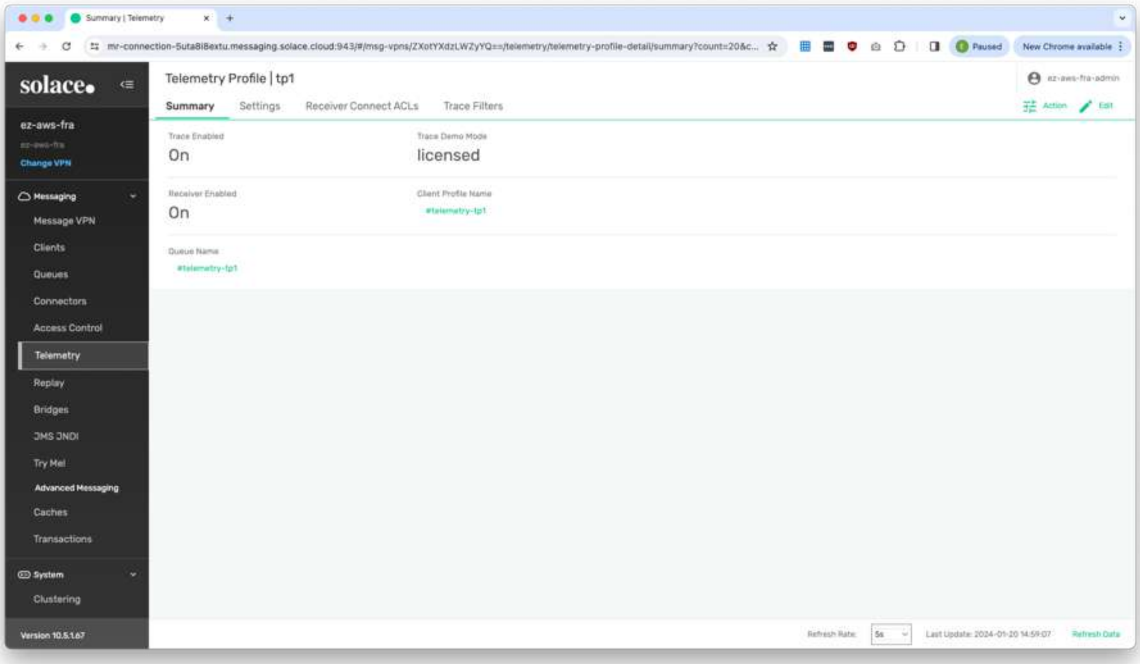


Use name "tp1" and check if Trace and Receiver are enabled then click [Apply]


New Telemetry Profile
Cancel
Apply
Telemetry Profile Configuration
Show Advanced Settings
Apply
Tips >>
Telemetry Profile Setup
Specify properties for the Telemetry Profile.
Telemetry Profile
The name of the Telemetry Profile.

Telemetry Profile: tp1
Trace
Enabled: on
Receiver
Enabled: on
Queue
Maximum Consumer Count: 1000
Messages Queued Quota (MB): 800000

Telemetry Profile is displayed with Client Profile Name and Queue Name #telemetry-tp1



Telemetry Profile | tp1

Trace Enabled
On

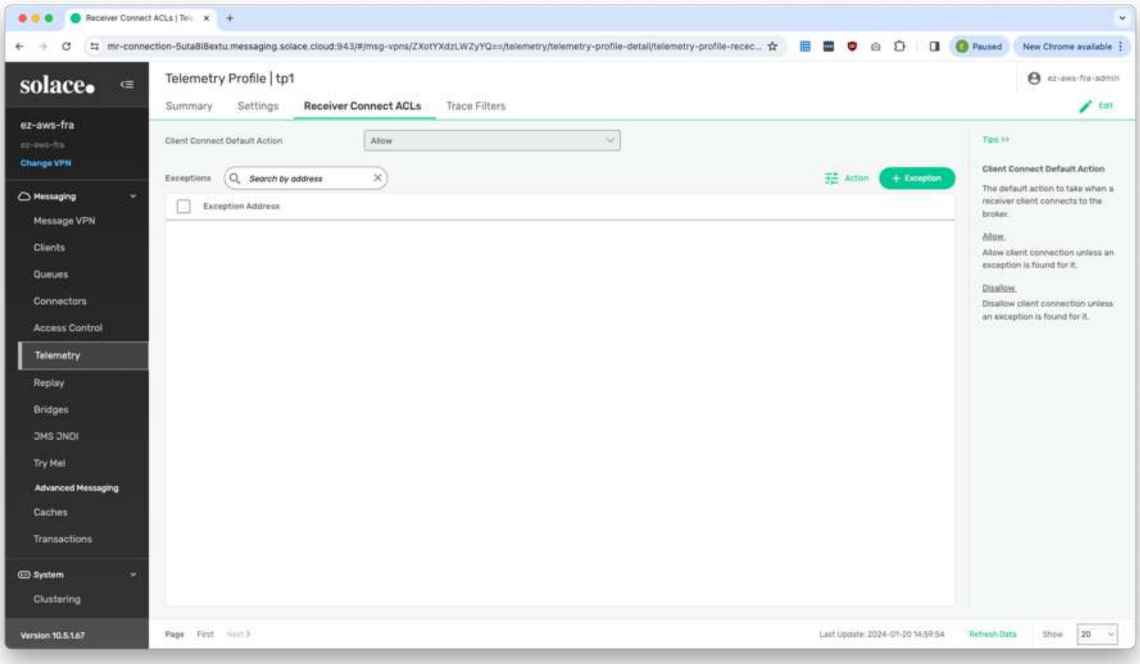
Trace Demo Mode
licensed

Receiver Enabled
On

Client Profile Name
#telemetry-tp1

Queue Name
#telemetry-tp1

Click Edit at Receiver Connect ACLs to switch from Disallow to Allow and click [Apply]



Telemetry Profile | tp1

Client Connect Default Action: Allow

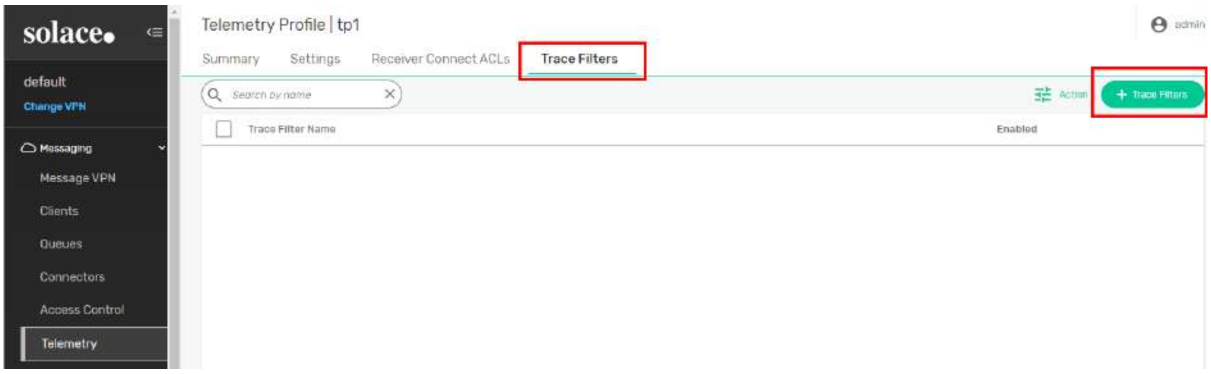
Exceptions

Client Connect Default Action

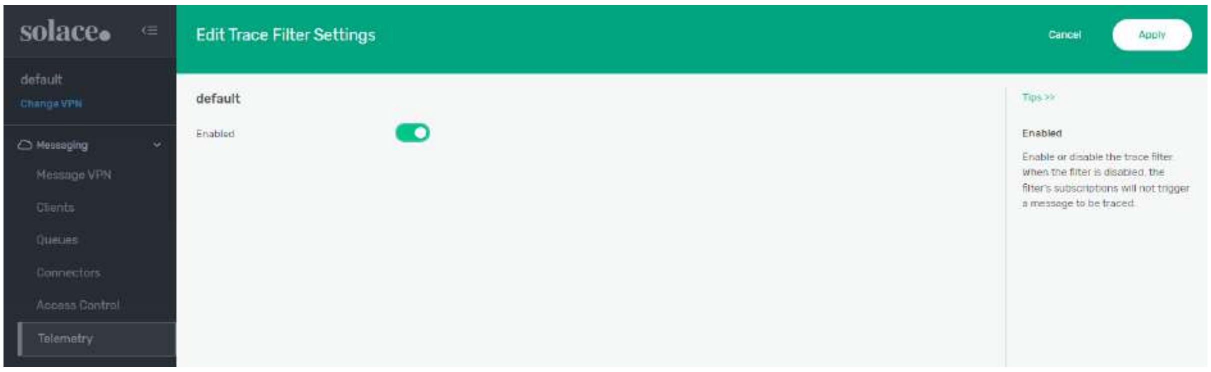
The default action to take when a receiver client connects to the broker.

Allow.
Allow client connection unless an exception is found for it.

Disallow.
Disallow client connection unless an exception is found for it.

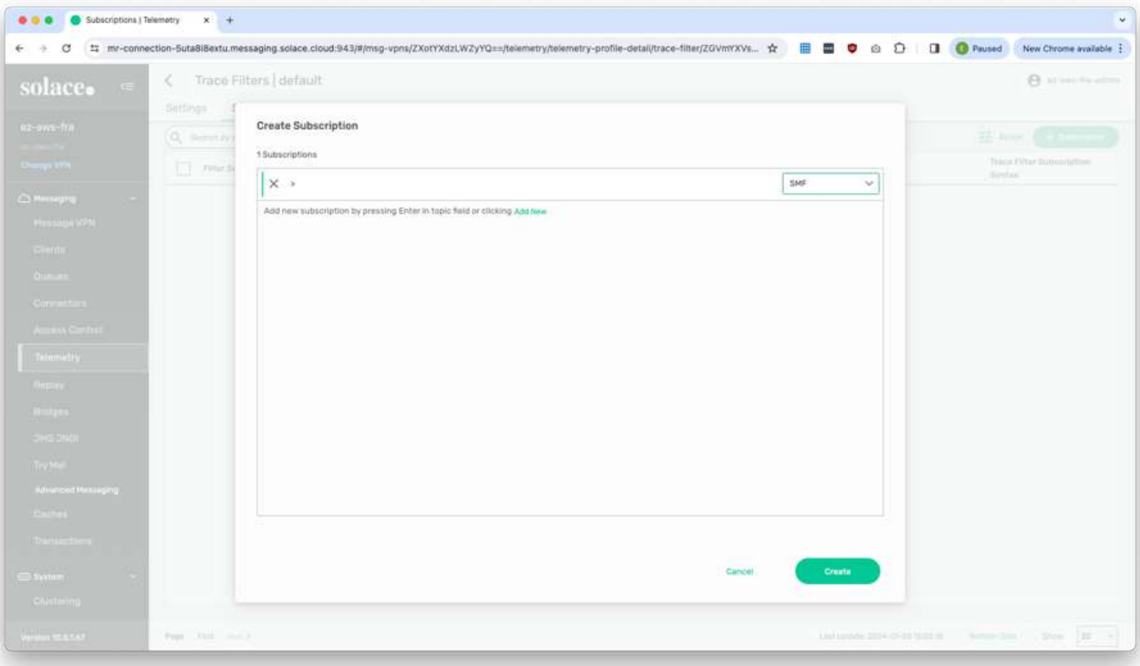


At Trace Filters click [+ Trace Filters] to add a filter with name "default" and set it to Enabled.

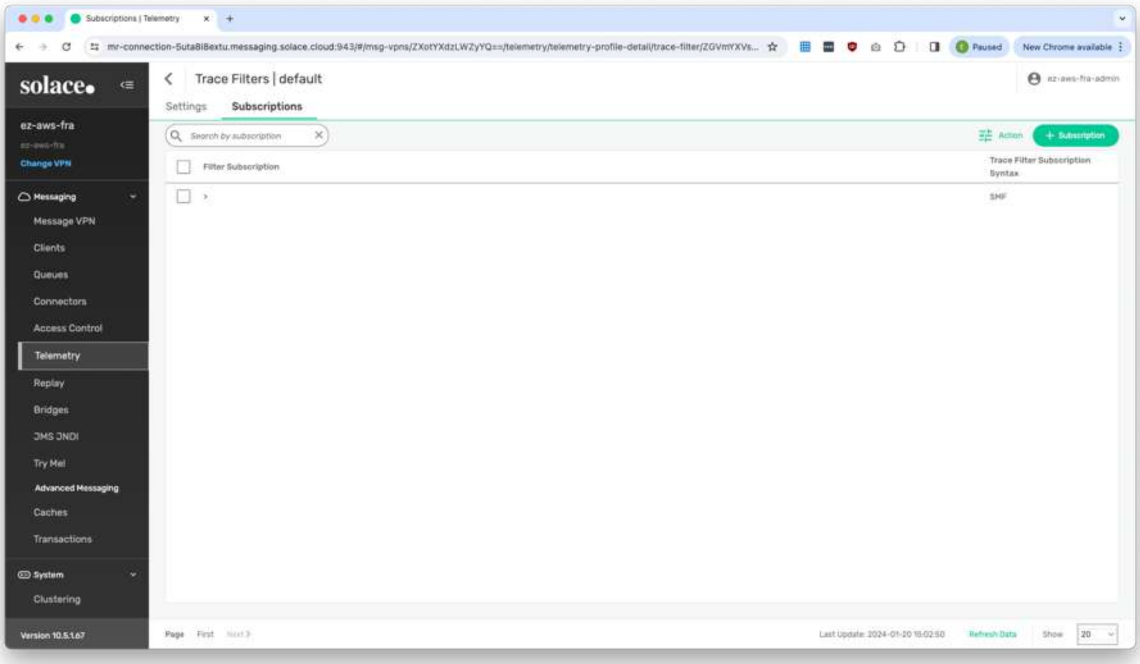


Enable or disable the trace filter. When the filter is disabled, the filter's subscriptions will not trigger a message to be traced.

Open this filter and create a subscription using the *greater than wildcard sign* ">"

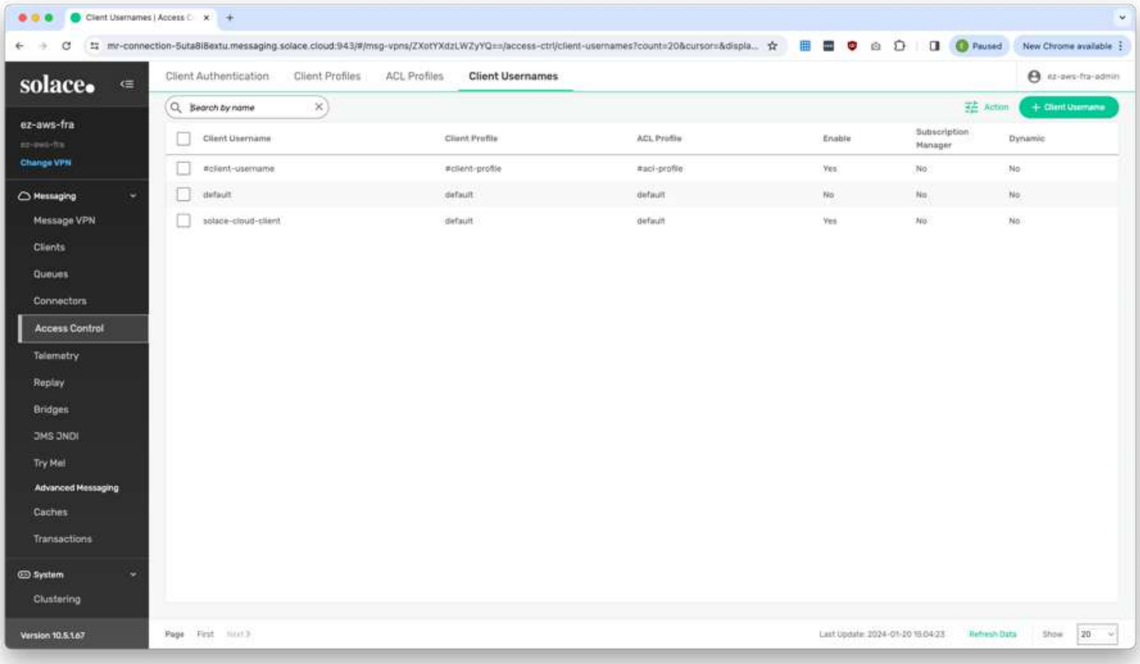

Subscriptions | Telemetry
mr-connection-5uta8i8extu.messaging.solace.cloud:943/#/msg-vpns/ZXotYXdzLWZyYQ==/telemetry/telemetry-profile-detail/trace-filter/ZGVmYXVs...
Paused
New Chrome available
solace
Trace Filters | default
Settings
Create Subscription
1 Subscriptions
>
SMF
Cancel
Create

Add new subscription by pressing Enter in topic field or clicking Add New

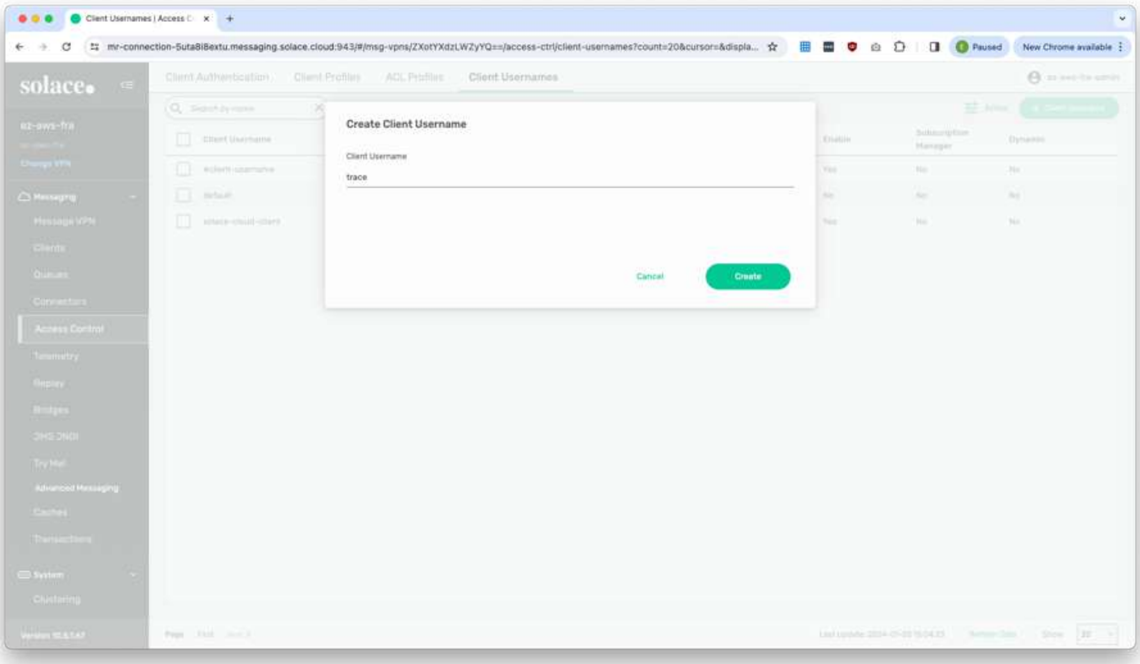


Filter Subscription	Trace Filter Subscription Syntax
>	SMF

At Access Control go to Client Usernames and click [+ Client Username] to create a client username "trace" with password "trace123" and "#telemetry-tp1" profiles:

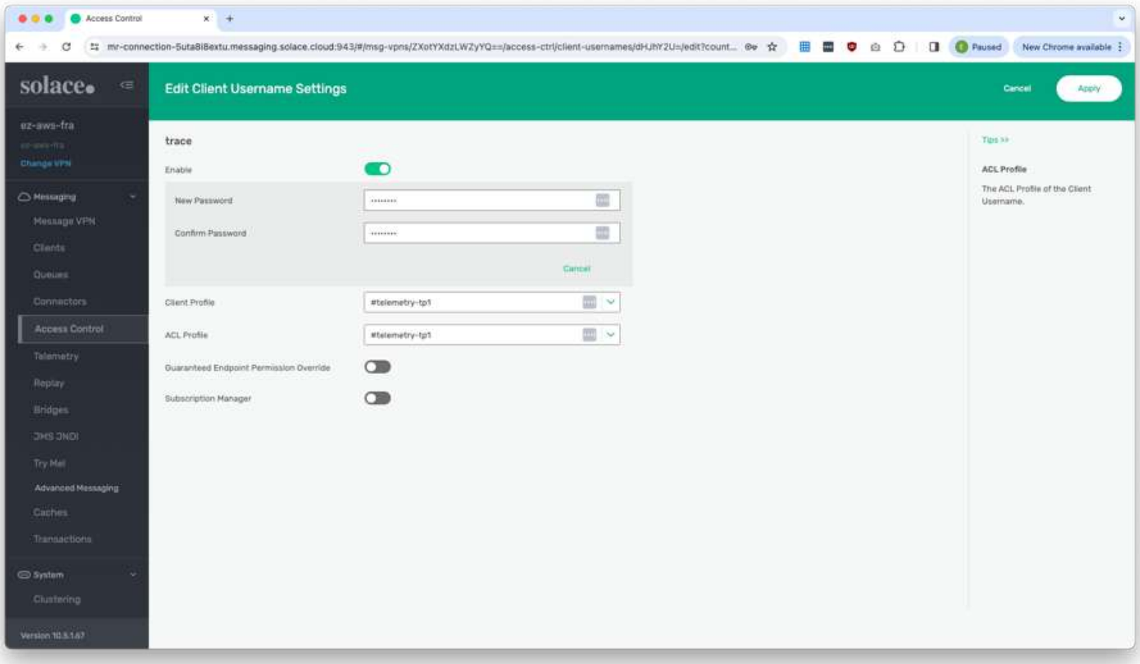


Client Username	Client Profile	ACL Profile	Enable	Subscription Manager	Dynamic
#client-username	#client-profile	#acl-profile	Yes	No	No
default	default	default	No	No	No
solace-cloud-client	default	default	Yes	No	No


Client Usernames | Access C
mr-connection-5uta8i8extu.messaging.solace.cloud:943/#/msg-vpns/ZXotYXdzLWZyYQ==/access-ctrl/client-usernames?count=20&cursor=&displa...
Client Authentication
Client Profiles
ACL Profiles
Client Usernames
Cancel
Create

Create Client Username

Client Username: trace



Edit Client Username Settings

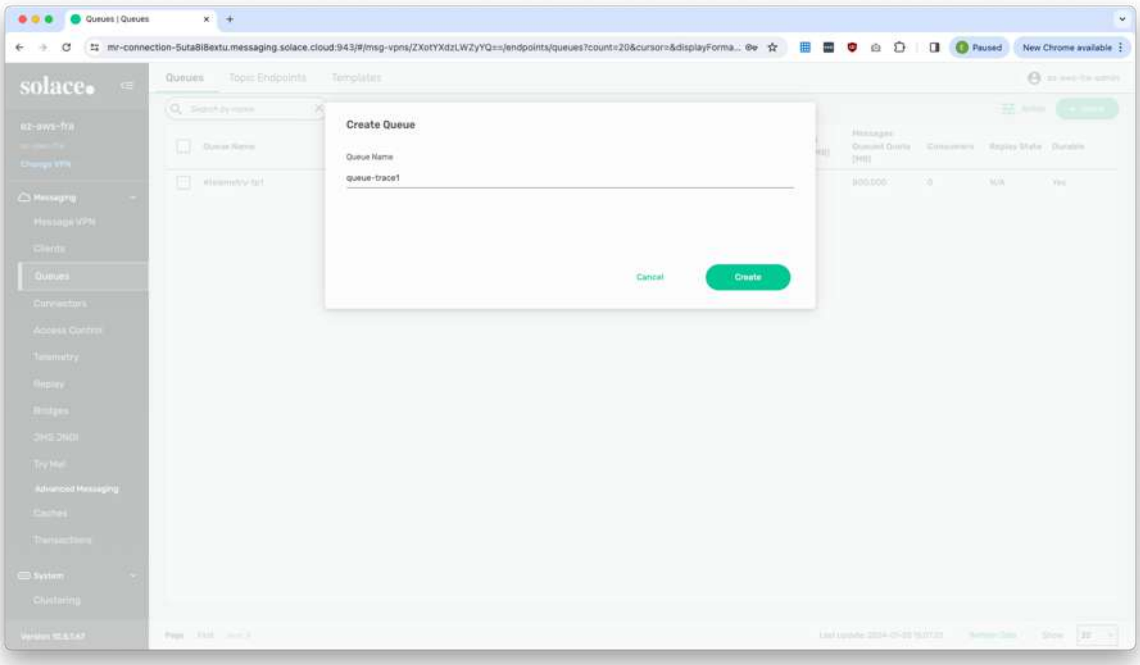
trace

Enable	On
New Password	········
Confirm Password	········
Client Profile	#telemetry-tp1
ACL Profile	#telemetry-tp1
Guaranteed Endpoint Permission Override	Off
Subscription Manager	Off

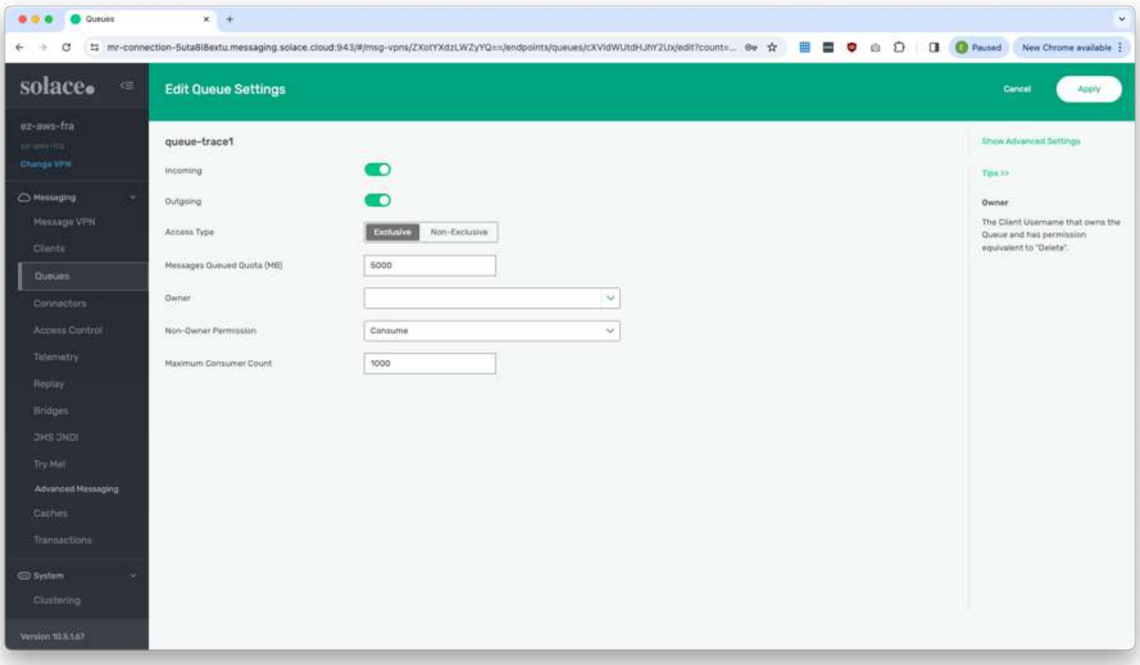
ACL Profile

The ACL Profile of the Client Username.

Create two or more queues (queue-trace1, queue-trace2, ...) with subscription "demo/trace". These are the messaging queues.



Queue Name: queue-trace1



queue-trace1

Incoming: on

Outgoing: on

Access Type: [✓ Exclusive] Non-Exclusive

Messages Queued Quota (MB): 5000

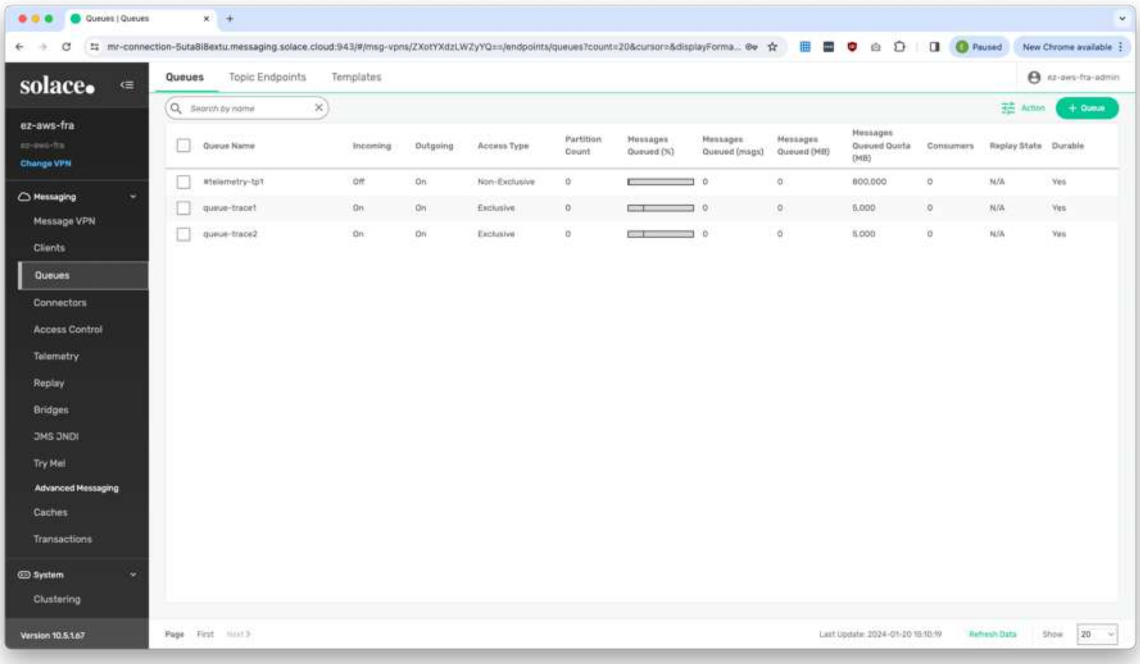
Owner:

Non-Owner Permission: Consume

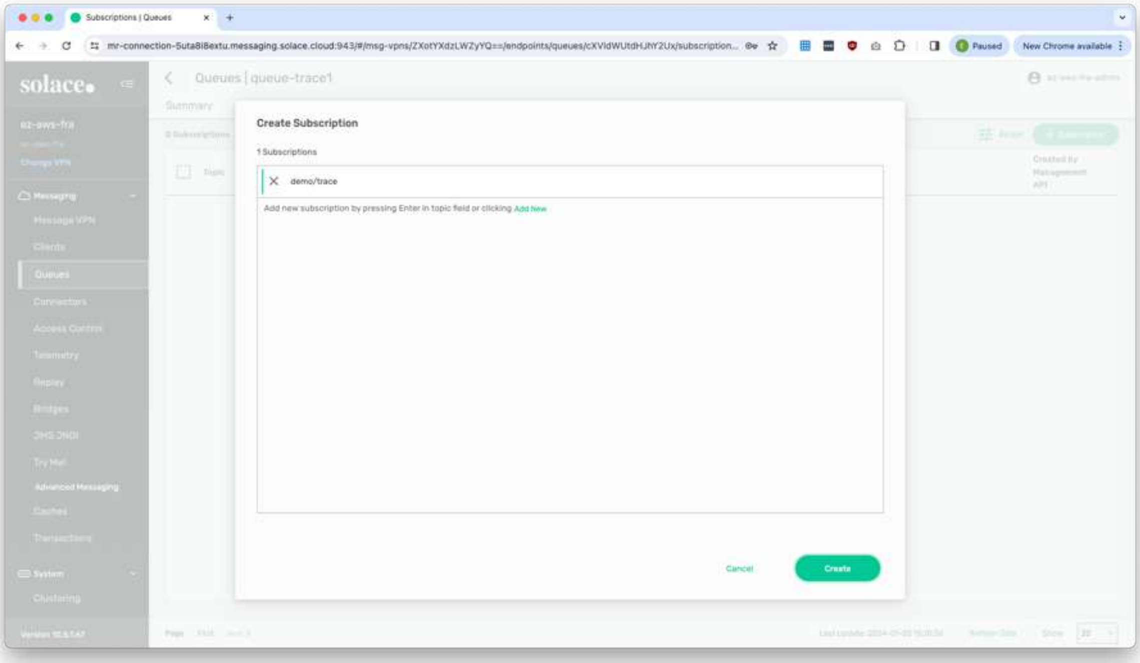
Maximum Consumer Count: 1000

Owner

The Client Username that owns the Queue and has permission equivalent to "Delete".



	Queue Name	Incoming	Outgoing	Access Type	Partition Count	Messages Queued (%)	Messages Queued (msgs)	Messages Queued (MB)	Messages Queued Quota (MB)	Consumers	Replay State	Durable
	#telemetry-tp1	Off	On	Non-Exclusive	0		0	0	800,000	0	N/A	Yes
	queue-trace1	On	On	Exclusive	0		0	0	5,000	0	N/A	Yes
	queue-trace2	On	On	Exclusive	0		0	0	5,000	0	N/A	Yes



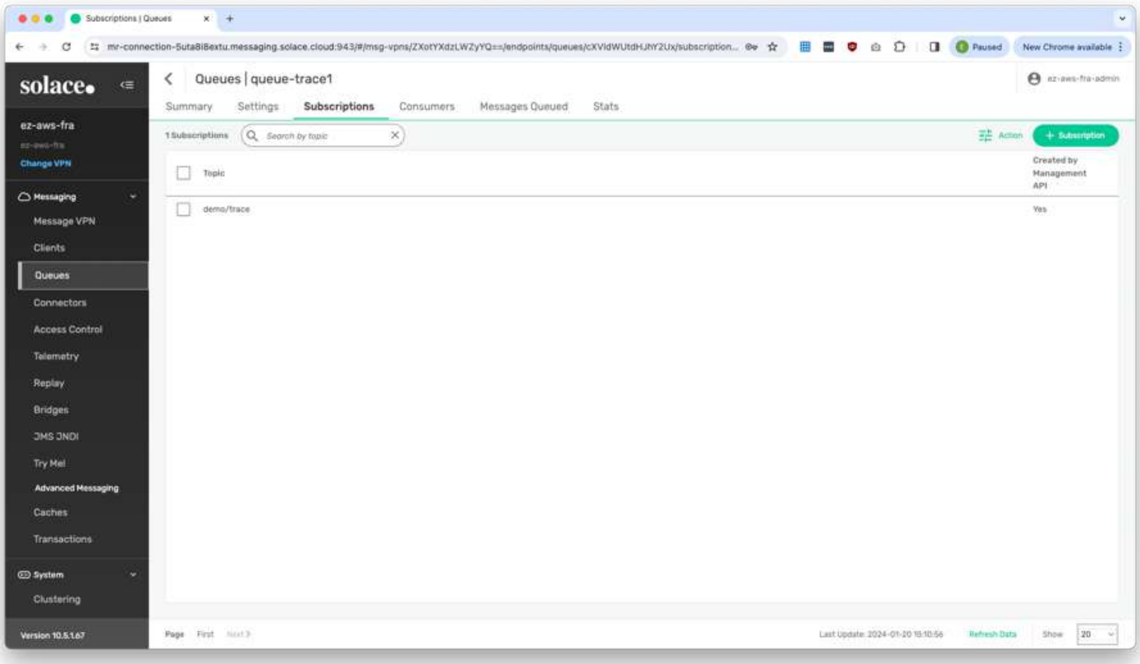
Create Subscription

1 Subscriptions

demo/trace

Add new subscription by pressing Enter in topic field or clicking Add New

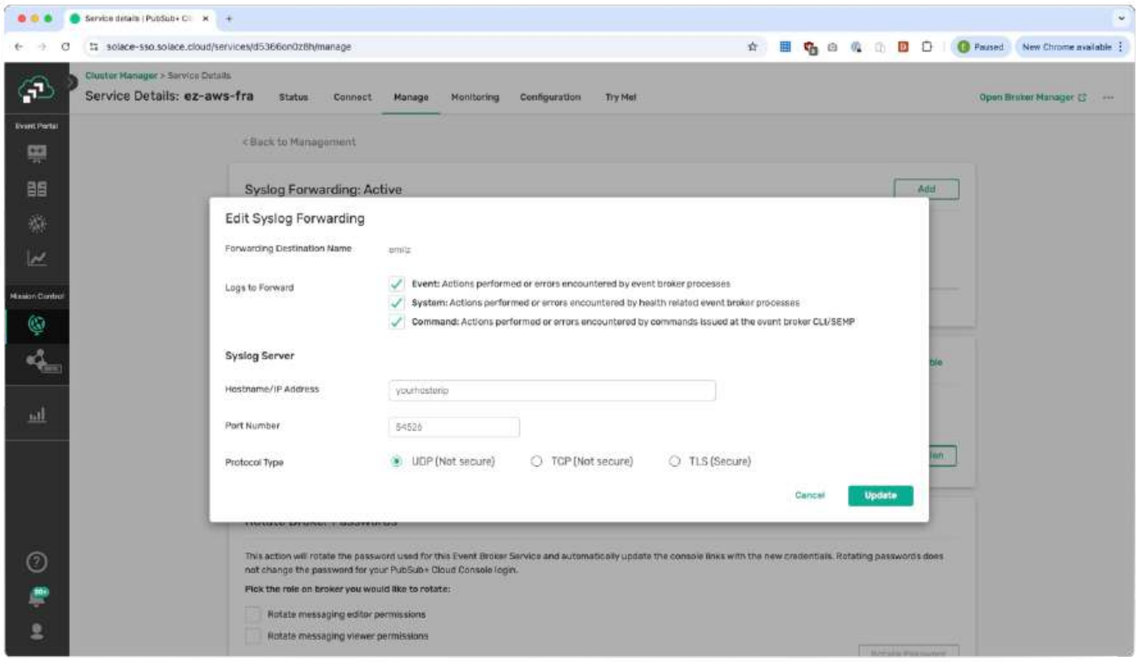
Subscriptions | Queues
mr-connection-5uta8i8extu.messaging.solace.cloud:943/#/msg-vpns/ZXotYXdzLWZyYQ==/endpoints/queues/cXVldWUtdHJhY2Ux/subscription...
Paused
New Chrome available
solace
Queues | queue-trace1
Summary
Cancel
Create



Configuration of the broker is now done.

2.4 Syslog Forwarding

See for configuration https://docs.solace.com/Cloud/cloud-syslog-forwarding.htm


Service details | PubSub+ Cl
solace-sso.solace.cloud/services/d5366on0z8h/manage
Cluster Manager > Service Details
Service Details: ez-aws-fra
Status
Connect
Manage
Monitoring
Configuration
Try Me!
Open Broker Manager
< Back to Management
Syslog Forwarding: Active
Add
Cancel
Update
Rotate messaging editor permissions
Rotate messaging viewer permissions

Edit Syslog Forwarding

Forwarding Destination Name: emilz

Logs to Forward:
- [✓] **Event:** Actions performed or errors encountered by event broker processes
- [✓] **System:** Actions performed or errors encountered by health related event broker processes
- [✓] **Command:** Actions performed or errors encountered by commands issued at the event broker CLI/SEMP

Syslog Server

Hostname/IP Address: yourhostorip

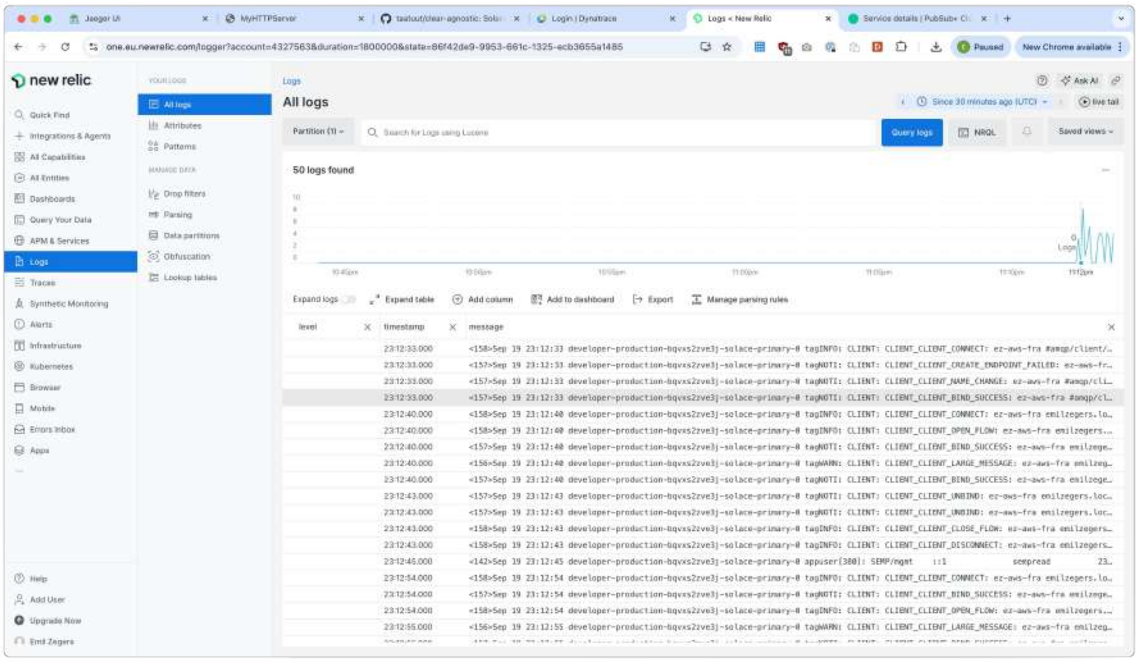
Port Number: 54526

Protocol Type: [✓ UDP (Not secure)] TCP (Not secure) TLS (Secure)

This action will rotate the password used for this Event Broker Service and automatically update the console links with the new credentials. Rotating passwords does not change the password for your PubSub+ Cloud Console login.

Pick the role on broker you would like to rotate:

Exported the log data to New Relic (you can have a permanent free account with some restrictions). Looks like Jaeger does not (directly) support ingesting log data? TODO: check, also explore Loki.



```
level   timestamp      message
        23:12:33.000   <158>Sep 19 23:12:33 developer-production-bqvxs2zve3j-solace-primary-0 tagINFO: CLIENT: CLIENT_CLIENT_CONNECT: ez-aws-fra #amqp/client/…
        23:12:33.000   <157>Sep 19 23:12:33 developer-production-bqvxs2zve3j-solace-primary-0 tagNOTI: CLIENT: CLIENT_CLIENT_CREATE_ENDPOINT_FAILED: ez-aws-fr…
        23:12:33.000   <157>Sep 19 23:12:33 developer-production-bqvxs2zve3j-solace-primary-0 tagNOTI: CLIENT: CLIENT_CLIENT_NAME_CHANGE: ez-aws-fra #amqp/cli…
        23:12:33.000   <157>Sep 19 23:12:33 developer-production-bqvxs2zve3j-solace-primary-0 tagNOTI: CLIENT: CLIENT_CLIENT_BIND_SUCCESS: ez-aws-fra #amqp/cl…
        23:12:40.000   <158>Sep 19 23:12:40 developer-production-bqvxs2zve3j-solace-primary-0 tagINFO: CLIENT: CLIENT_CLIENT_CONNECT: ez-aws-fra emilzegers.lo…
        23:12:40.000   <158>Sep 19 23:12:40 developer-production-bqvxs2zve3j-solace-primary-0 tagINFO: CLIENT: CLIENT_CLIENT_OPEN_FLOW: ez-aws-fra emilzegers.…
        23:12:40.000   <157>Sep 19 23:12:40 developer-production-bqvxs2zve3j-solace-primary-0 tagNOTI: CLIENT: CLIENT_CLIENT_BIND_SUCCESS: ez-aws-fra emilzege…
        23:12:40.000   <156>Sep 19 23:12:40 developer-production-bqvxs2zve3j-solace-primary-0 tagWARN: CLIENT: CLIENT_CLIENT_LARGE_MESSAGE: ez-aws-fra emilzeg…
        23:12:40.000   <157>Sep 19 23:12:40 developer-production-bqvxs2zve3j-solace-primary-0 tagNOTI: CLIENT: CLIENT_CLIENT_BIND_SUCCESS: ez-aws-fra emilzege…
        23:12:43.000   <157>Sep 19 23:12:43 developer-production-bqvxs2zve3j-solace-primary-0 tagNOTI: CLIENT: CLIENT_CLIENT_UNBIND: ez-aws-fra emilzegers.loc…
        23:12:43.000   <157>Sep 19 23:12:43 developer-production-bqvxs2zve3j-solace-primary-0 tagNOTI: CLIENT: CLIENT_CLIENT_UNBIND: ez-aws-fra emilzegers.loc…
        23:12:43.000   <158>Sep 19 23:12:43 developer-production-bqvxs2zve3j-solace-primary-0 tagINFO: CLIENT: CLIENT_CLIENT_CLOSE_FLOW: ez-aws-fra emilzegers…
        23:12:43.000   <158>Sep 19 23:12:43 developer-production-bqvxs2zve3j-solace-primary-0 tagINFO: CLIENT: CLIENT_CLIENT_DISCONNECT: ez-aws-fra emilzegers…
        23:12:46.000   <142>Sep 19 23:12:45 developer-production-bqvxs2zve3j-solace-primary-0 appuser[380]: SEMP/mgmt    ::1          sempread        23…
        23:12:54.000   <158>Sep 19 23:12:54 developer-production-bqvxs2zve3j-solace-primary-0 tagINFO: CLIENT: CLIENT_CLIENT_CONNECT: ez-aws-fra emilzegers.lo…
        23:12:54.000   <157>Sep 19 23:12:54 developer-production-bqvxs2zve3j-solace-primary-0 tagNOTI: CLIENT: CLIENT_CLIENT_BIND_SUCCESS: ez-aws-fra emilzege…
        23:12:54.000   <158>Sep 19 23:12:54 developer-production-bqvxs2zve3j-solace-primary-0 tagINFO: CLIENT: CLIENT_CLIENT_OPEN_FLOW: ez-aws-fra emilzegers.…
        23:12:55.000   <156>Sep 19 23:12:55 developer-production-bqvxs2zve3j-solace-primary-0 tagWARN: CLIENT: CLIENT_CLIENT_LARGE_MESSAGE: ez-aws-fra emilzeg…
```

When using the JSON export, the log data can also be displayed by the Simple OTEL endpoint Python script. Example log data:

```
{"resourceLogs":[{"resource":{},"scopeLogs":[{"scope":{},"logRecords":[{"timeUnixNano":"17266
72533000000000","observedTimeUnixNano":"1726672533570971000","severityNumber":10,"severityTex
t":"notice","body":{"stringValue":"\u003c157\u003eSep 18 15:15:33 developer-production-
bqvxs2zve3j-solace-primary-0 tagNOTI: CLIENT: CLIENT_CLIENT_UNBIND: ez-aws-fra
emilzegers.local/60503/b12f5c57cd5958cd0001/vGOdYyr7dG Client (325)
emilzegers.local/60503/b12f5c57cd5958cd0001/vGOdYyr7dG username solace-cloud-client Unbind to
Flow Id (576), ForwardingMode(StoreAndForward), final statistics - flow(0, 0, 0, 0, 0, 1, 0,
0, 1, 0), isActive(No), Reason(Client issued
unbind)"},"attributes":[{"key":"facility","value":{"intValue":"19"}},{"key":"hostname","value
":{"stringValue":"developer-production-bqvxs2zve3j-solace-primary-
0"}},{"key":"message","value":{"stringValue":"CLIENT: CLIENT_CLIENT_UNBIND: ez-aws-fra
emilzegers.local/60503/b12f5c57cd5958cd0001/vGOdYyr7dG Client (325)
emilzegers.local/60503/b12f5c57cd5958cd0001/vGOdYyr7dG username solace-cloud-client Unbind to
Flow Id (576), ForwardingMode(StoreAndForward), final statistics - flow(0, 0, 0, 0, 0, 1, 0,
0, 1, 0), isActive(No), Reason(Client issued
unbind)"}},{"key":"priority","value":{"intValue":"157"}},{"key":"appname","value":{"stringVal
ue":"tagNOTI"}}],"traceId":"","spanId":""},{"timeUnixNano":"1726672533000000000","observedTim
eUnixNano":"1726672533602244000","severityNumber":10,"severityText":"notice","body":{"stringV
alue":"\u003c157\u003eSep 18 15:15:33 developer-production-bqvxs2zve3j-solace-primary-0
tagNOTI: CLIENT: CLIENT_CLIENT_UNBIND: ez-aws-fra
emilzegers.local/60503/b12f5c57cd5958cd0001/vGOdYyr7dG Client (325)
emilzegers.local/60503/b12f5c57cd5958cd0001/vGOdYyr7dG username solace-cloud-client Unbind to
Flow Id (1074), ForwardingMode(StoreAndForward), final statistics - flow(0, 0, 0, 0, 0, 1, 0,
0, 1, 0), isActive(No), Reason(Client issued
unbind)"},"attributes":[{"key":"priority","value":{"intValue":"157"}},{"key":"facility","valu
e":{"intValue":"19"}},{"key":"hostname","value":{"stringValue":"developer-production-
bqvxs2zve3j-solace-primary-
0"}},{"key":"appname","value":{"stringValue":"tagNOTI"}},{"key":"message","value":{"stringVal
ue":"CLIENT: CLIENT_CLIENT_UNBIND: ez-aws-fra
emilzegers.local/60503/b12f5c57cd5958cd0001/vGOdYyr7dG Client (325)
emilzegers.local/60503/b12f5c57cd5958cd0001/vGOdYyr7dG username solace-cloud-client Unbind to
Flow Id (1074), ForwardingMode(StoreAndForward), final statistics - flow(0, 0, 0, 0, 0, 1, 0,
0, 1, 0), isActive(No), Reason(Client issued
unbind)"}}],"traceId":"","spanId":""},{"timeUnixNano":"1726672533000000000","observedTimeUnix
Nano":"1726672533631824000","severityNumber":9,"severityText":"info","body":{"stringValue":"\
u003c158\u003eSep 18 15:15:33 developer-production-bqvxs2zve3j-solace-primary-0 tagINFO:
CLIENT: CLIENT_CLIENT_CLOSE_FLOW: ez-aws-fra
emilzegers.local/60503/b12f5c57cd5958cd0001/vGOdYyr7dG Client (325)
emilzegers.local/60503/b12f5c57cd5958cd0001/vGOdYyr7dG username solace-cloud-client Pub flow
session flow name 50e5f354f0154822a429ca719cdb87ac (1052), transacted session id -1,
publisher id 1001, last message id 117988, window size 50, final statistics - flow(0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1)"},"attributes":[{"key":"appname","value":{"stringValue":"tagINFO"}},{"key":"priority","val
ue":{"intValue":"158"}},{"key":"facility","value":{"intValue":"19"}},{"key":"hostname","value
":{"stringValue":"developer-production-bqvxs2zve3j-solace-primary-
0"}},{"key":"message","value":{"stringValue":"CLIENT: CLIENT_CLIENT_CLOSE_FLOW: ez-aws-fra
emilzegers.local/60503/b12f5c57cd5958cd0001/vGOdYyr7dG Client (325)
emilzegers.local/60503/b12f5c57cd5958cd0001/vGOdYyr7dG username solace-cloud-client Pub flow
session flow name 50e5f354f0154822a429ca719cdb87ac (1052), transacted session id -1,
publisher id 1001, last message id 117988, window size 50, final statistics - flow(0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1)"}}],"traceId":"","spanId":""},{"timeUnixNano":"1726672533000000000","observedTimeUnixNano"
:"1726672533639511000","severityNumber":9,"severityText":"info","body":{"stringValue":"\u003c
158\u003eSep 18 15:15:33 developer-production-bqvxs2zve3j-solace-primary-0 tagINFO: CLIENT:
CLIENT_CLIENT_DISCONNECT: ez-aws-fra emilzegers.local/60503/b12f5c57cd5958cd0001/vGOdYyr7dG
Client (325) emilzegers.local/60503/b12f5c57cd5958cd0001/vGOdYyr7dG username solace-cloud-
client WebSessionId (N/A) reason(Peer TCP Closed) final statistics - dp(14, 9, 1, 2, 15, 11,
974, 884, 33001, 66116, 33975, 67000, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0) conn(0, 0,
95.97.192.6:52946, ESTAB, 0, 0, 0) zip(0, 0, 0, 0, 0.00, 0.00, 0, 0, 0, 0, 0, 0, 0, 0) web(0,
0, 0, 0, 0, 0, 0), SslVersion(TLSv1.2), SslCipher(ECDHE-RSA-AES256-SHA384 TLSv1.2 Kx=ECDH
Au=RSA Enc=AES(256)
Mac=SHA384)"},"attributes":[{"key":"facility","value":{"intValue":"19"}},{"key":"message","va
lue":{"stringValue":"CLIENT: CLIENT_CLIENT_DISCONNECT: ez-aws-fra
emilzegers.local/60503/b12f5c57cd5958cd0001/vGOdYyr7dG Client (325)
emilzegers.local/60503/b12f5c57cd5958cd0001/vGOdYyr7dG username solace-cloud-client
WebSessionId (N/A) reason(Peer TCP Closed) final statistics - dp(14, 9, 1, 2, 15, 11, 974,
884, 33001, 66116, 33975, 67000, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0) conn(0, 0, 95.97.192.6:52946,
ESTAB, 0, 0, 0) zip(0, 0, 0, 0, 0.00, 0.00, 0, 0, 0, 0, 0, 0, 0, 0) web(0, 0, 0, 0, 0, 0, 0),
SslVersion(TLSv1.2), SslCipher(ECDHE-RSA-AES256-SHA384 TLSv1.2 Kx=ECDH Au=RSA Enc=AES(256)
Mac=SHA384)"}},{"key":"priority","value":{"intValue":"158"}},{"key":"hostname","value":{"stri
ngValue":"developer-production-bqvxs2zve3j-solace-primary-
0"}},{"key":"appname","value":{"stringValue":"tagINFO"}}],"traceId":"","spanId":""}]}]}]}
```

Some yaml excerpts below.

Add New Relic exporter to exporters section:

```
 otlphttp/newrelic:
   endpoint: ${NEW_RELIC_EXPORTER_OTLP_ENDPOINT}
   tls:
     insecure: false
   headers:
     api-key: ${NEW_RELIC_LICENSE_KEY}
```

Service section:

```
service:
 telemetry:
   logs:
     level: "debug"
 pipelines:
   logs:
     receivers: [syslog]
     exporters: [logging, otlphttp/newrelic, otlphttp/jsontest]
   traces:
     receivers: [solace/broker1, otlp]
     processors: [batch]
#      exporters: [logging, otlp/jaeger, otlphttp/newrelic, otlphttp/dynatrace, otlphttp/jsontest]
     exporters: [logging, otlp/jaeger, otlphttp/newrelic, otlphttp/jsontest]
```

See section OTEL Collector for full YAML configuration information.

2.5 Jaeger

Download Jaeger from here: https://www.jaegertracing.io/download/
Examplefor for MacOS: jaeger-1.53.0-darwin-amd64.tar.gz

Installation

```
mkdir -p ~/jaeger
cd ~/jaeger
tar xzvf <your download directory>/jaeger-1.53.0-darwin-amd64.tar.gz
```

Gives output like:

```
x jaeger-1.53.0-darwin-amd64/
x jaeger-1.53.0-darwin-amd64/example-hotrod
…
x jaeger-1.53.0-darwin-amd64/jaeger-ingester
x jaeger-1.53.0-darwin-amd64/jaeger-query
```

Remove extended attributes from extracted files to avoid MacOS popup warnings on downloaded (executable) files, assuming you are allowed to administer the Mac you are working on:

```
xattr -rc '~/jaeger'
```

If necessary, add sudo.

Start

```
cd ~/jaeger/jaeger-1.53.0-darwin-amd64/
./jaeger-all-in-one
# Or detached:
#nohup ./jaeger-all-in-one > /dev/null 2>&1 &
```

To stop kill the process with Control-C.

2.6 OTEL collector

Download from here:
https://github.com/open-telemetry/opentelemetry-collector-releases/releases/
Example for MacOS ARM: otelcol-contrib_0.96.0_darwin_arm64_darwin_arm64.tar.gz

Installation

```
mkdir -p ~/otelcol/otelcol-contrib_0.96.0_darwin_arm64
cd ~/otelcol/otelcol-contrib_0.96.0_darwin_arm64
tar xzvf <your download directory>/otelcol-contrib_0.96.0_darwin_arm64_darwin_arm64.tar.gz
```

Gives output like:

```
x LICENSE
x README.md
x otelcol-contrib
```

Prepare config files here:

```
cd ~/otelcol
```

Ping for IP address (what is preferred method to obtain -static- IP address?):

```
ping mr-connection-5uta8l8extu.messaging.solace.cloud
```

Added a custom hostname ez-dt.messaging.solace.cloud

Example for one broker: otel-collector-config-single.yaml:
<TODO: add yaml>

When working with multiple brokers define additional receivers and include in service/receivers:

```
receivers:
  solace/broker1:
    broker: [1.2.3.4:5671]
    max_unacknowledged: 500
    auth:
      sasl_plain:
        username: trace
        password: trace123
    queue: queue://#telemetry-tp1
    tls:
      insecure: false
      insecure_skip_verify: true

  solace/broker2:
    broker: [4.3.2.1:5671]
    max_unacknowledged: 500
    auth:
      sasl_plain:
        username: trace
        password: trace123
    queue: queue://#telemetry-tp1
    tls:
      insecure: false
      insecure_skip_verify: true

service:
  telemetry:
    logs:
      level: "debug"
  pipelines:
    traces:
      receivers: [solace/broker1, solace/broker2]
      processors: [batch]
      exporters: [logging, otlp/jaeger]
```

To get rid of other Apple messages like "can't be opened because Apple cannot check it for malicious software." , assuming you are allowed to administer the Mac you are working on:

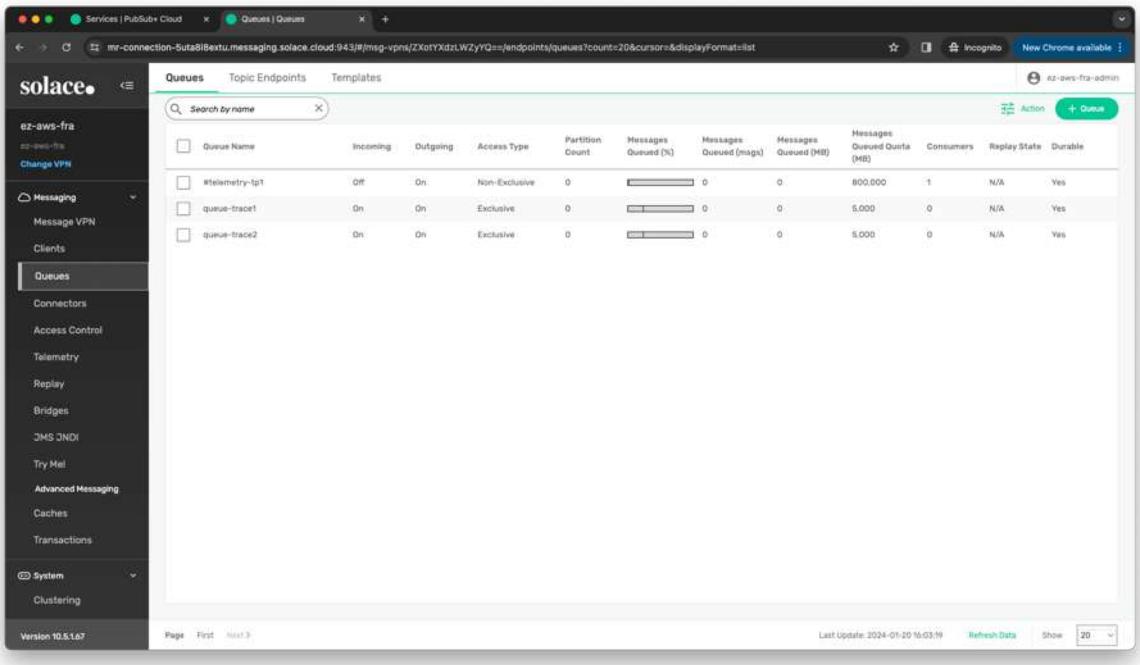
```
sudo spctl --master-disable
```

Start

```
cd ~/otelcol/otelcol-contrib_0.96.0_darwin_arm64
./otelcol-contrib --config=../otel-collector-config-single.yaml
# Or detached:
#nohup ./otelcol-contrib --config=../otel-collector-config-single.yaml > /dev/null 2>&1 &
```

To stop kill the process with Control-C.

In the Broker verify that the Telemetry queue #telemetry-tp1 has a (1) consumer at Consumers:



Queue Name	Incoming	Outgoing	Access Type	Partition Count	Messages Queued (%)	Messages Queued (msgs)	Messages Queued (MB)	Messages Queued Quota (MB)	Consumers	Replay State	Durable
#telemetry-tp1	Off	On	Non-Exclusive	0		0	0	800,000	1	N/A	Yes
queue-trace1	On	On	Exclusive	0		0	0	5,000	0	N/A	Yes
queue-trace2	On	On	Exclusive	0		0	0	5,000	0	N/A	Yes

2.7 OpenSearch

There is no direct download for OpenSearch for macOS now as it is not fully supported. Install using docker or brew works fine. OpenSearch requires Java to run.
To use a direct install without docker or brew go to https://github.com/opensearch-project/opensearch-build/issues/4670 and download from https://artifacts.opensearch.org/snapshots/core/opensearch/3.0.0-SNAPSHOT/opensearch-min-3.0.0-SNAPSHOT-darwin-arm64-latest.tar.gz This tarball includes a JDK.

2.7.1 Installation

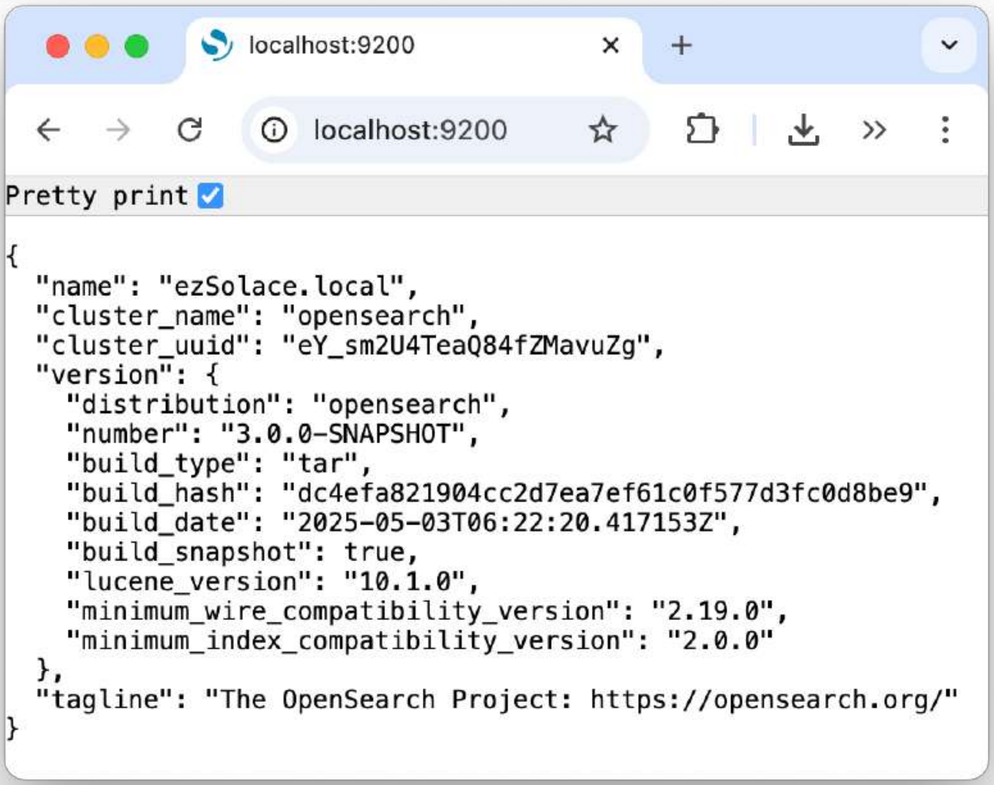
```
mkdir -p ~/opensearch
cd ~/opensearch
tar xzvf <your download directory>/opensearch-min-3.0.0-SNAPSHOT-darwin-arm64-latest.tar.gz
```

Now run OpenSearch, might need to remove Apple quarantine attribute from jdk.app.

```
xattr -d com.apple.quarantine ~/opensearch/opensearch-3.0.0-SNAPSHOT/jdk.app
cd opensearch-3.0.0-SNAPSHOT/bin
./opensearch
```

Gives output like:

```
WARNING: Using incubator modules: jdk.incubator.vector
WARNING: Unknown module: org.apache.arrow.memory.core specified to --add-opens
[2025-05-10T23:20:18,424][WARN ][stderr               ] [ezSolace.local] May 10, 2025 11:20:18 PM
org.opensearch.javaagent.bootstrap.AgentPolicy setPolicy
[2025-05-10T23:20:18,425][WARN ][stderr               ] [ezSolace.local] INFO: Policy attached successfully:
org.opensearch.bootstrap.OpenSearchPolicy@2b037cfc
[2025-05-10T23:20:18,430][INFO ][o.o.n.Node           ] [ezSolace.local] version[3.0.0-SNAPSHOT], pid[4157],
build[tar/dc4efa821904cc2d7ea7ef61c0f577d3fc0d8be9/2025-05-03T06:22:20.417153Z], OS[Mac OS
X/15.4.1/aarch64], JVM[Eclipse Adoptium/OpenJDK 64-Bit Server VM/21.0.7/21.0.7+6-LTS]
[2025-05-10T23:20:18,430][INFO ][o.o.n.Node           ] [ezSolace.local] JVM home
[/Users/emilzegers/opensearch/opensearch-3.0.0-SNAPSHOT/jdk.app/Contents/Home], using bundled
JDK/JRE [true]
…
[2025-05-10T23:20:24,341][INFO ][o.o.h.AbstractHttpServerTransport] [ezSolace.local] publish_address
{127.0.0.1:9200}, bound_addresses {[::1]:9200}, {127.0.0.1:9200}
[2025-05-10T23:20:24,344][INFO ][o.o.n.Node           ] [ezSolace.local] started
[2025-05-10T23:20:24,355][INFO ][o.o.g.GatewayService     ] [ezSolace.local] recovered [0] indices into
cluster_state
```



```
{
  "name": "ezSolace.local",
  "cluster_name": "opensearch",
  "cluster_uuid": "eY_sm2U4TeaQ84fZMavuZg",
  "version": {
    "distribution": "opensearch",
    "number": "3.0.0-SNAPSHOT",
    "build_type": "tar",
    "build_hash": "dc4efa821904cc2d7ea7ef61c0f577d3fc0d8be9",
    "build_date": "2025-05-03T06:22:20.417153Z",
    "build_snapshot": true,
    "lucene_version": "10.1.0",
    "minimum_wire_compatibility_version": "2.19.0",
    "minimum_index_compatibility_version": "2.0.0"
  },
  "tagline": "The OpenSearch Project: https://opensearch.org/"
}
```

For information about experimental support for Distributed Tracing in OpenSearch see https://docs.opensearch.org/docs/latest/observing-your-data/trace/distributed-tracing/ To enable add (uncomment) below lines to ~/opensearch/opensearch-3.0.0-SNAPSHOT/config/opensearch.yml and (re)start OpenSearch

```
opensearch.experimental.feature.extensions.enabled: true
opensearch.experimental.feature.telemetry.enabled: true
telemetry.feature.tracer.enabled: true
telemetry.tracer.enabled: true
```

https://docs.opensearch.org/docs/latest/api-reference/index-apis/stats/
Check http://localhost:9200/_stats

3 Testing the application chain

3.1 Solace SDKPerf

In this demo you will use Solace SDKPerf, a general purpose testing tool with support for OpenTelemetry. You can find information about SDKPerf at https://docs.solace.com/API/SDKPerf/SDKPerf.htm and downloads at https://solace.com/downloads/?fwp_downloads_types=other
On MacOS you can for example use the Java version: sdkperf-jcsmp-8.4.14.10.zip or sdkperf-mqtt-8.4.15.5.zip

3.1.1 Installation

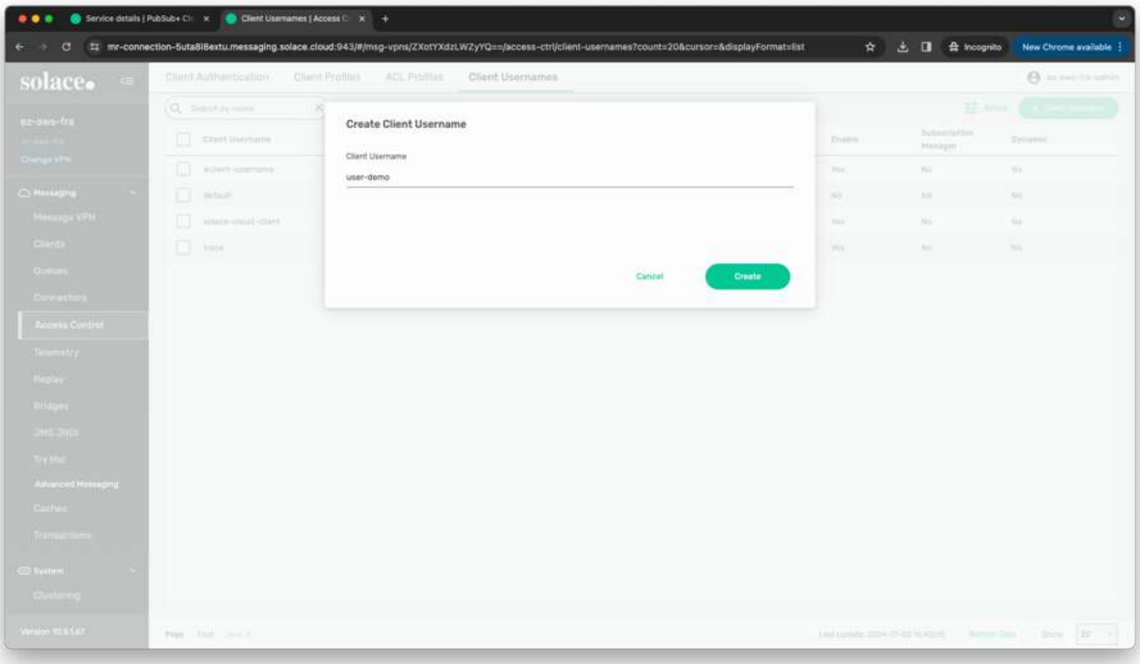
```
mkdir -p ~/sdkperf
cd ~/sdkperf
tar xzvf <your download directory>/sdkperf-jcsmp-8.4.14.10.zip
tar xzvf <your download directory>/sdkperf-mqtt-8.4.15.5.zip
```

Gives output like:

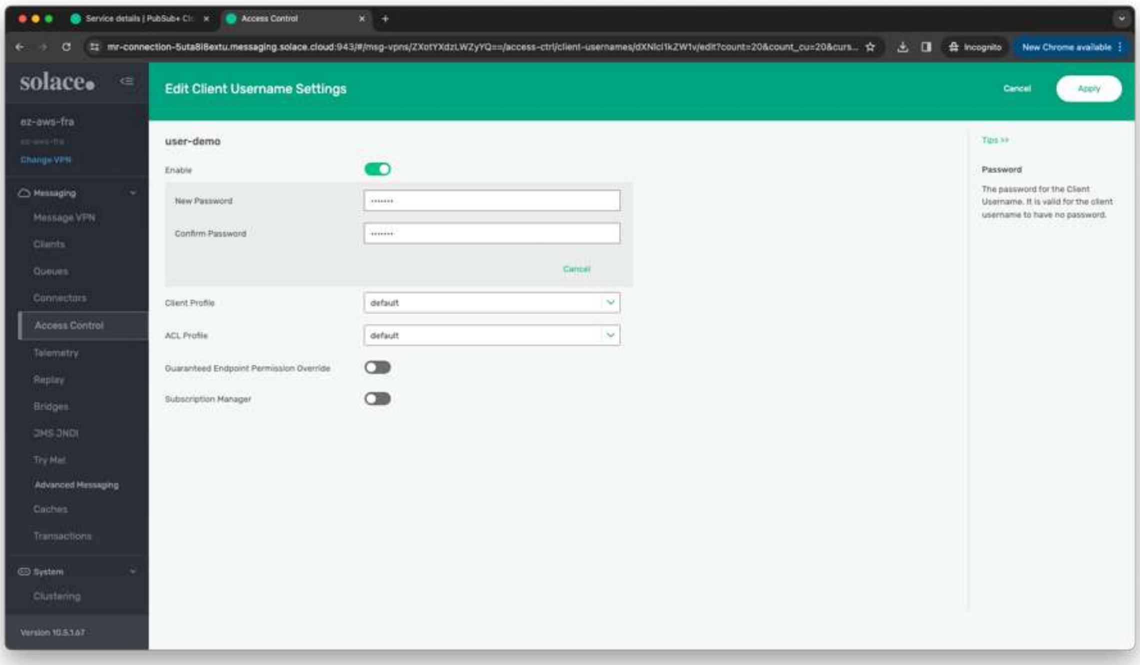
```
x sdkperf-jcsmp-8.4.14.10/
x sdkperf-jcsmp-8.4.14.10/lib/
…
x sdkperf-jcsmp-8.4.14.10/sdkperf_java.bat
x sdkperf-jcsmp-8.4.14.10/sdkperf_java.sh
```

And similar for the MQTT version

In Broker Manager > Messaging > Acces Control > Client Usernames create user user-demo with password default.


Create Client Username

Client Username: user-demo


Edit Client Username Settings

user-demo

Enable: on

New Password: •••••••

Confirm Password: •••••••

Client Profile: default

ACL Profile: default

Guaranteed Endpoint Permission Override: off

Subscription Manager: off

Password

The password for the Client Username. It is valid for the client username to have no password.

Publish a message and receive it from 2 queues:

cd ~/sdkperf/sdkperf-jcsmp-8.4.14.10

Using Distributed Tracing from/to SDKPerf
https://solace.community/discussion/1633/distributed-tracing-context-propagation
With default user solace-cloud-client and initial auto-generated hostname

Can add -md flag to dump message, or -tmd to dump trace message (sort of works does drop an error)

./sdkperf_java.sh -cip=tcps://mr-connection-5uta8l8extu.messaging.solace.cloud:55443 -cu=solace-cloud-client@ez-aws-fra -cp=deun1l905ashrflooldf1qhrfg -ptl='demo/trace' -sql='queue-trace1,queue-trace2' -mt=persistent -mn=1 -mr=1 -msa=32768 -q -tcc -tcrc -tecip="http://localhost:4317"

Run repeatedly every 10 seconds

while true; do ./sdkperf_java.sh -cip=tcps://mr-connection-5uta8l8extu.messaging.solace.cloud:55443 -cu=solace-cloud-client@ez-aws-fra -cp=deun1l905ashrflooldf1qhrfg -ptl='demo/trace' -sql='queue-trace1,queue-trace2' -mt=persistent -mn=1 -mr=1 -msa=32768 -q -tcc -tcrc -tecip="http://localhost:4317"; sleep 10; done

With default user solace-cloud-client and initial auto-generated hostname

./sdkperf_java.sh -cip=tcps://mr-connection-5uta8l8extu.messaging.solace.cloud:55443 -cu=solace-cloud-client@ez-aws-fra -cp=deun1l905ashrflooldf1qhrfg -ptl='demo/trace' -sql='queue-trace1,queue-trace2' -mt=persistent -mn=1 -mr=1 -msa=32768 -q

With default user solace-cloud-client and additional created hostname

./sdkperf_java.sh -cip=tcps://ez-dt.messaging.solace.cloud:55443 -cu=solace-cloud-client@ez-aws-fra -cp=deun1l905ashrflooldf1qhrfg -ptl='demo/trace' -sql='queue-trace1,queue-trace2' -mt=persistent -mn=1 -mr=1 -msa=32768 -q

With default user solace-cloud-client and IP address (Dynamic? Going round between 18.159.178.64, 18.153.239.155, … How to find these, and/or create static?)

./sdkperf_java.sh -cip=tcps://18.159.178.64:55443 -cu=solace-cloud-client@ez-aws-fra -cp=deun1l905ashrflooldf1qhrfg -ptl='demo/trace' -sql='queue-trace1,queue-trace2' -mt=persistent -mn=1 -mr=1 -msa=32768 -q

With created user user-demo

./sdkperf_java.sh -cip=tcps://mr-connection-5uta8l8extu.messaging.solace.cloud:55443 -cu=user-demo@ez-aws-fra -cp=default -ptl='demo/trace' -sql='queue-trace1,queue-trace2' -mt=persistent -mn=1 -mr=1 -msa=32768 -q

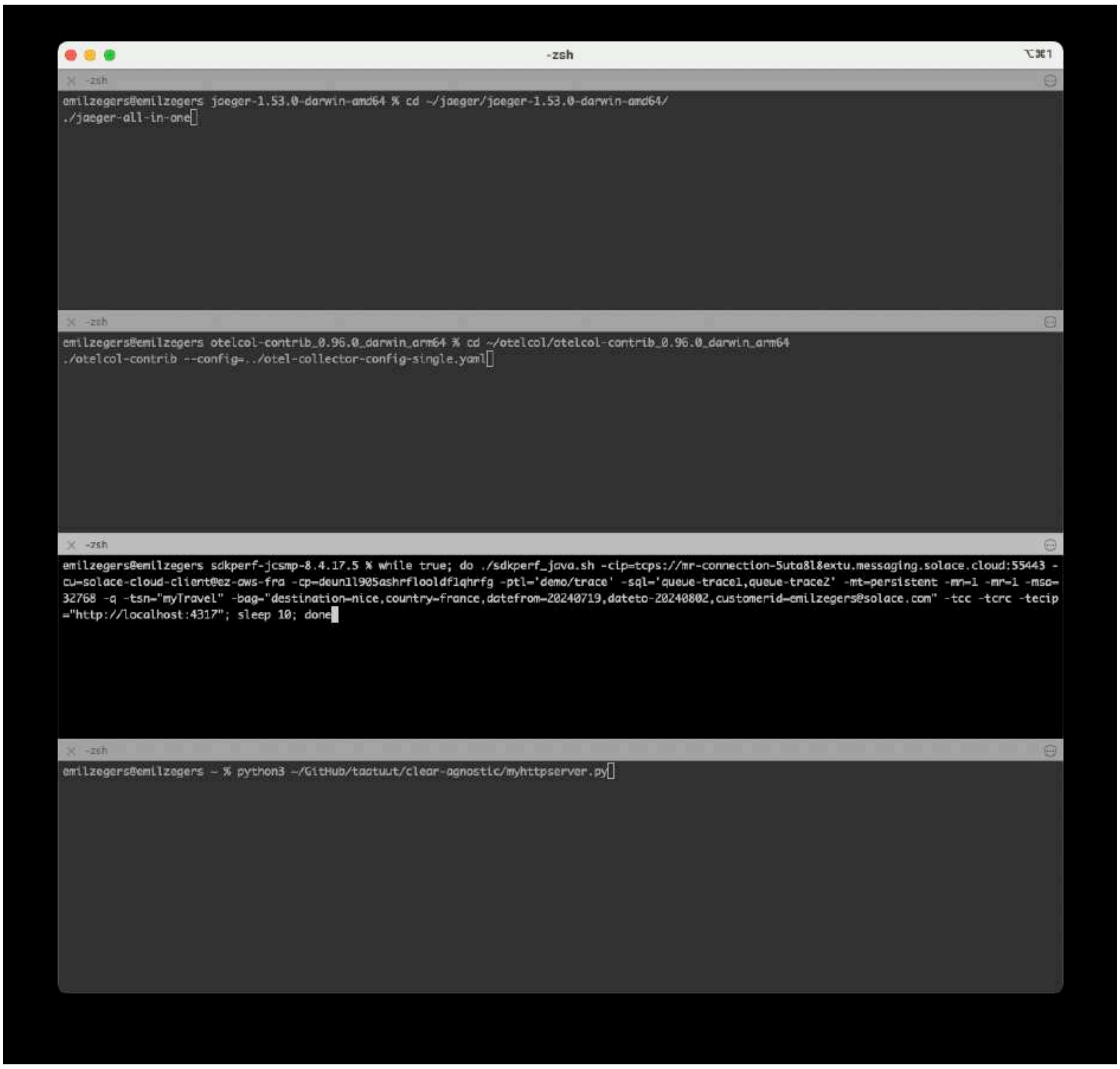
For MQTT

./sdkperf_mqtt.sh -cip=ssl://ez-dt.messaging.solace.cloud:8883 -cu=solace-cloud-client@ez-aws-fra -cp=deun1l905ashrflooldf1qhrfg -ptl='demo/trace' -sql='queue-trace1,queue-trace2' -mpq=1 -msq=1 -mn=1 -mr=1 -msa=32768 -q -tcc -tcrc -tecip="http://localhost:4317"

MQTT5

./sdkperf_mqtt5.sh -cip=ssl://ez-dt.messaging.solace.cloud:8883 -cu=solace-cloud-client@ez-aws-fra -cp=deun1l905ashrflooldf1qhrfg -ptl='demo/trace' -sql='queue-trace3' -mpq=1 -msq=1 -mn=1 -mr=1 -msa=32768 -q -tcc -tcrc -tecip="http://localhost:4317"

https://docs.solace.com/API/SDKPerf/Command-Line-Options.htm
https://docs.solace.com/API/SDKPerf/Example-Commands.htm



```
emilzegers@emilzegers jaeger-1.53.0-darwin-amd64 % cd ~/jaeger/jaeger-1.53.0-darwin-amd64/
./jaeger-all-in-one
```

```
emilzegers@emilzegers otelcol-contrib_0.96.0_darwin_arm64 % cd ~/otelcol/otelcol-contrib_0.96.0_darwin_arm64
./otelcol-contrib --config=../otel-collector-config-single.yaml
```

```
emilzegers@emilzegers sdkperf-jcsmp-8.4.17.5 % while true; do ./sdkperf_java.sh -cip=tcps://mr-connection-5uta8l8extu.messaging.solace.cloud:55443 -
cu=solace-cloud-client@ez-aws-fra -cp=deun1l905ashrflooldflqhrfg -ptl='demo/trace' -sql='queue-trace1,queue-trace2' -mt=persistent -mn=1 -mr=1 -msa=
32768 -q -tsn="myTravel" -bag="destination=nice,country=france,datefrom=20240719,dateto=20240802,customerid=emilzegers@solace.com" -tcc -tcrc -tecip
="http://localhost:4317"; sleep 10; done
```

```
emilzegers@emilzegers ~ % python3 ~/GitHub/taatuut/clear-agnostic/myhttpserver.py
```

Can use something like iTerm to have all terminals together (jaeger, otelcollector, sdkperf and Simple OTEL endpoint from top to bottom).

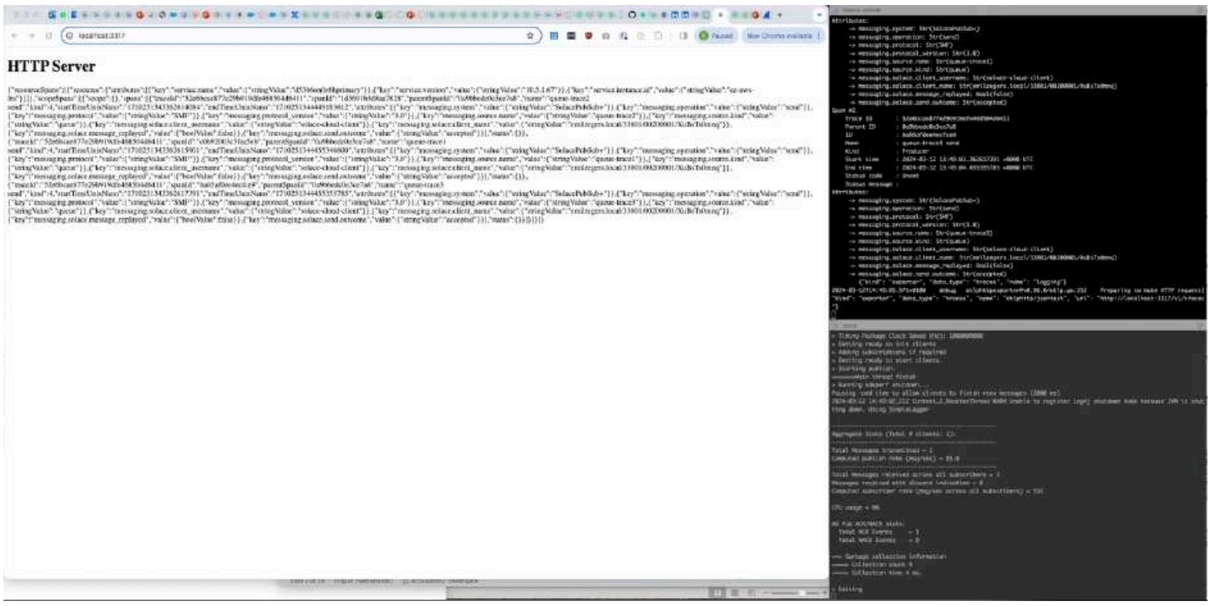
4 Results

4.1 Simple OTEL endpoint

The Simple OTEL endpoint Python script processes POST requests from OTEL collector exporter with metrics in JSON and just displays the data received. See the simpleotelendpoint.py script, and relevant configuration in OTEL collector YAML file.

Example configuration for JSON:

```
otlphttp/jsontest:
  endpoint: "http://localhost:3317/"
  compression: "none"
  encoding: "json"
  tls:
    insecure: true
  headers:
    Content-Type: "application/json"
```



HTTP Server

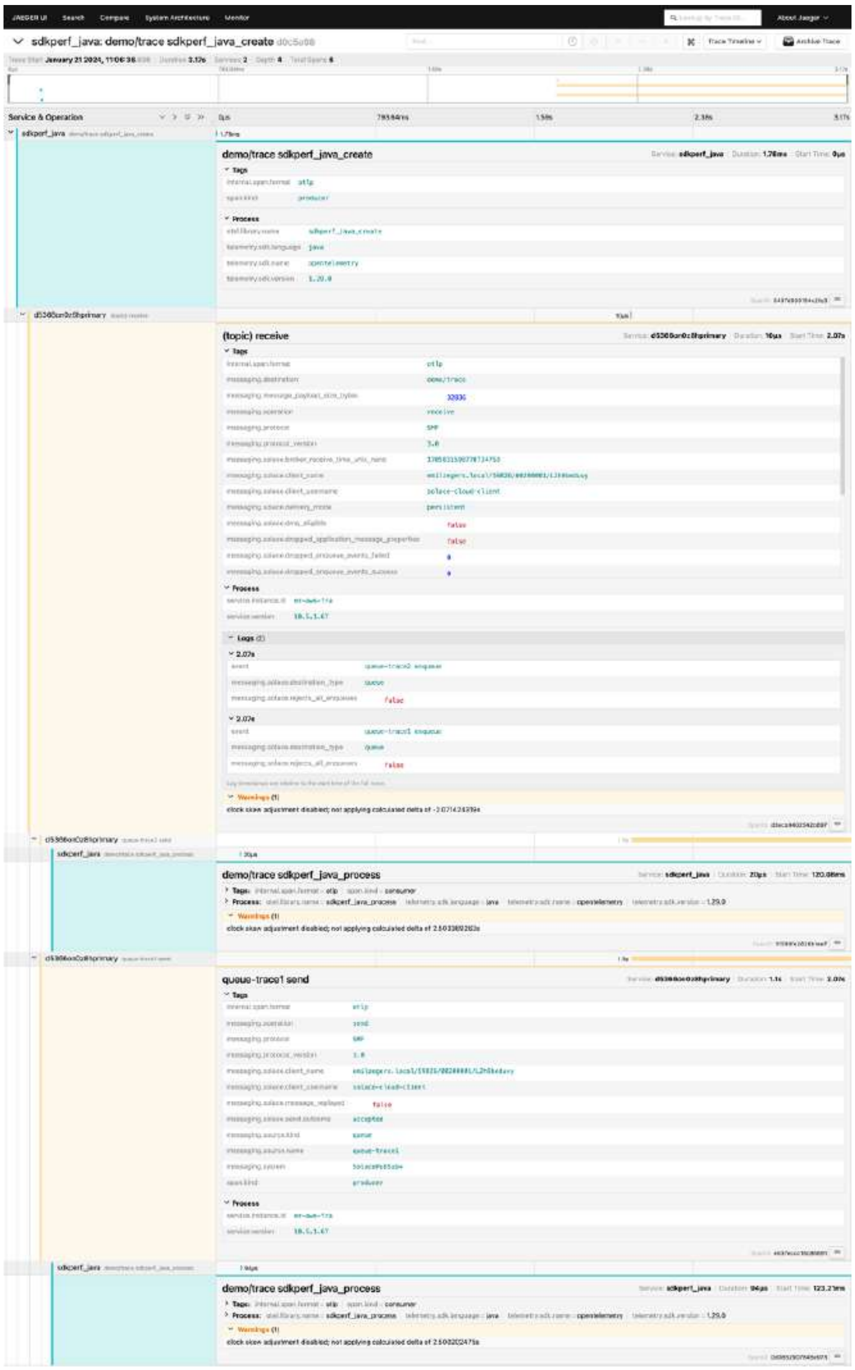
[illegible]

```
Attributes:
     -> messaging.system: Str(SolacePubSub+)
     -> messaging.operation: Str(send)
     -> messaging.protocol: Str(SMF)
     -> messaging.protocol_version: Str(3.0)
     -> messaging.source.name: Str(queue-trace1)
     -> messaging.source.kind: Str(queue)
     -> messaging.solace.client_username: Str(solace-cloud-client)
     -> messaging.solace.client_name: Str([illegible])
     -> messaging.solace.message_replayed: Bool(false)
     -> messaging.solace.send.outcome: Str(accepted)
Span #2
    Trace ID       : [illegible]
    Parent ID      : [illegible]
    ID             : [illegible]
    Name           : queue-trace3 send
    Kind           : Producer
    Start time     : 2024-03-12 [illegible] +0000 UTC
    End time       : 2024-03-12 [illegible] +0000 UTC
    Status code    : Unset
    Status message :
Attributes:
     -> messaging.system: Str(SolacePubSub+)
     -> messaging.operation: Str(send)
     -> messaging.protocol: Str(SMF)
     -> messaging.protocol_version: Str(3.0)
     -> messaging.source.name: Str(queue-trace3)
     -> messaging.source.kind: Str(queue)
     -> messaging.solace.client_username: Str(solace-cloud-client)
     -> messaging.solace.client_name: Str([illegible])
     -> messaging.solace.message_replayed: Bool(false)
     -> messaging.solace.send.outcome: Str(accepted)
        {"kind": "exporter", "data_type": "traces", "name": "logging"}
2024-03-12T[illegible]    debug    [illegible]    Preparing to make HTTP request
"kind": "exporter", "data_type": "traces", "name": "otlphttp/jsontest", "url": "http://localhost:3317/v1/traces
"}
```

```
> Timing Package Clock Speed (Hz): [illegible]
> Getting ready to init clients
> Adding subscriptions if required
> Getting ready to start clients.
> Starting pubsub:
-------Main thread finish
> Running sdkperf shutdown....
Pausing [illegible] to allow clients to finish [illegible]
[illegible]
[illegible]

-------------------------------------------------
Aggregate Stats (Total # clients: 1):
-------------------------------------------------
Total Messages transmitted = 1
Computed publish rate (msg/sec) = [illegible]
-------------------------------------------------
Total Messages received across all subscribers = 1
Messages received with discard indication = 0
Computed subscriber rate (msg/sec across all subscribers) = [illegible]

CPU usage = 0%

AD Pub ACK/NACK stats:
  Total ACK Events    = 1
  Total NACK Events   = 0

--- Garbage collection information
----- Collection count 0
----- Collection time 0 ms

> Exiting
```

4.2 Jaeger

Navigate to http://localhost:16686 to access the Jaeger UI (server status info at http://0.0.0.0:14269/, see https://www.jaegertracing.io/docs/1.53/deployment/ for more info).



JAEGER UI Search Compare System Architecture Monitor About Jaeger

sdkperf_java: demo/trace sdkperf_java_create d0c5a08

Trace Timeline Archive Trace

Trace Start January 21 2024, 11:06:36.838 Duration 3.17s Services 2 Depth 4 Total Spans 6

Service & Operation 0μs 793.64ms 1.59s 2.38s 3.17s

sdkperf_java demo/trace sdkperf_java_create 1.78ms

demo/trace sdkperf_java_create Service: sdkperf_java Duration: 1.78ms Start Time: 0μs

Tags
internal.span.format otlp
span.kind producer

Process
otel.library.name sdkperf_java_create
telemetry.sdk.language java
telemetry.sdk.name opentelemetry
telemetry.sdk.version 1.29.0

SpanID: 8497e89919b4a29a9

d5366on0c8hprimary (topic) receive 10μs

(topic) receive Service: d5366on0c8hprimary Duration: 10μs Start Time: 2.07s

Tags
internal.span.format otlp
messaging.destination demo/trace
messaging.message_payload_size_bytes 32836
messaging.operation receive
messaging.protocol SMF
messaging.protocol_version 3.0
messaging.solace.broker_receive_time_unix_nano 1705831598778734750
messaging.solace.client_name emilzegers.local/58836/00280001/LJt6bedLvy
messaging.solace.client_username solace-cloud-client
messaging.solace.delivery_mode persistent
messaging.solace.dmq_eligible false
messaging.solace.dropped_application_message_properties false
messaging.solace.dropped_enqueue_events_failed 0
messaging.solace.dropped_enqueue_events_success 0

Process
service.instance.id mr-dwm-1ra
service.version 10.5.1.67

Logs (2)

2.07s
event queue-trace2 enqueue
messaging.solace.destination_type queue
messaging.solace.rejects_all_enqueues false

2.07s
event queue-trace1 enqueue
messaging.solace.destination_type queue
messaging.solace.rejects_all_enqueues false

Log timestamps are relative to the start time of the full trace.

Warnings (1)
clock skew adjustment disabled; not applying calculated delta of -2.071424319s

SpanID: d8eca9402342c89f

d5366on0c8hprimary queue-trace2 send 1.1s

sdkperf_java demo/trace sdkperf_java_process 20μs

demo/trace sdkperf_java_process Service: sdkperf_java Duration: 20μs Start Time: 120.08ms

Tags: internal.span.format = otlp span.kind = consumer
Process: otel.library.name = sdkperf_java_process telemetry.sdk.language = java telemetry.sdk.name = opentelemetry telemetry.sdk.version = 1.29.0

Warnings (1)
clock skew adjustment disabled; not applying calculated delta of 2.503389282s

SpanID: 9f988fe2616b1eaf

d5366on0c8hprimary queue-trace1 send 1.1s

queue-trace1 send Service: d5366on0c8hprimary Duration: 1.1s Start Time: 2.07s

Tags
internal.span.format otlp
messaging.operation send
messaging.protocol SMF
messaging.protocol_version 1.0
messaging.solace.client_name emilzegers.local/55026/00280001/L2h5bedavy
messaging.solace.client_username solace-cloud-client
messaging.solace.message_replayed false
messaging.solace.send.outcome accepted
messaging.source.kind queue
messaging.source.name queue-trace1
messaging.system SolacePubSub+
span.kind producer

Process
service.instance.id mr-dwm-1ra
service.version 10.5.1.67

SpanID: e69fecc19c80001

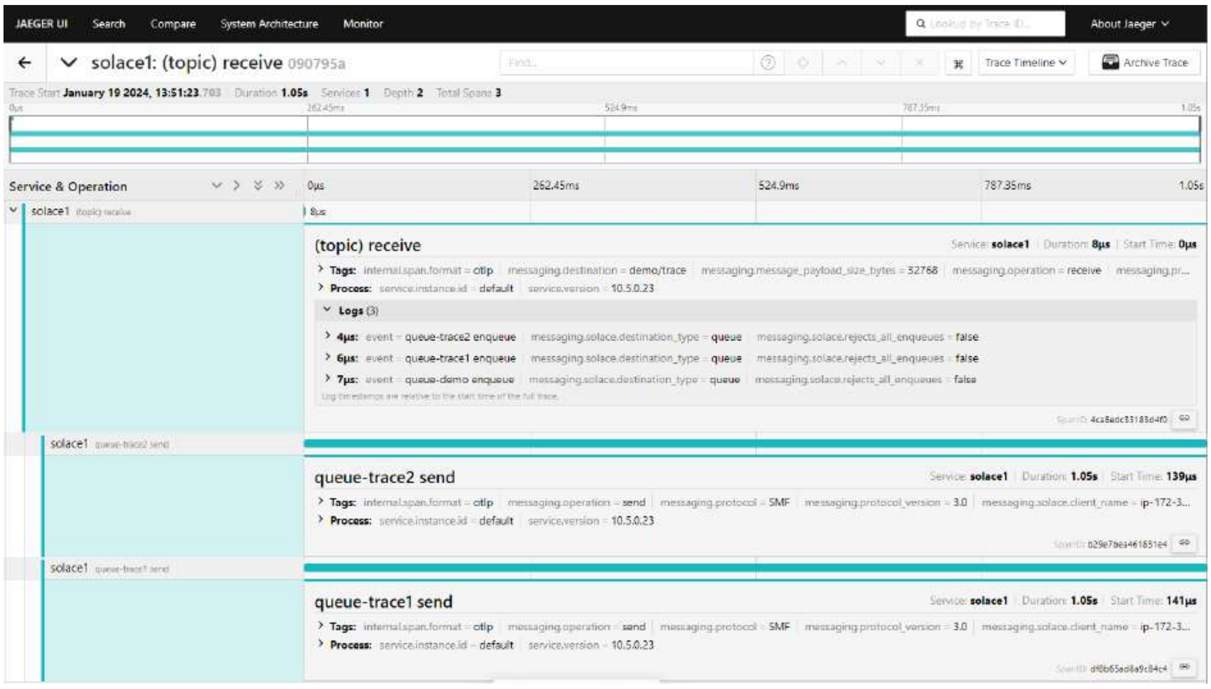
sdkperf_java demo/trace sdkperf_java_process 94μs

demo/trace sdkperf_java_process Service: sdkperf_java Duration: 94μs Start Time: 123.21ms

Tags: internal.span.format = otlp span.kind = consumer
Process: otel.library.name = sdkperf_java_process telemetry.sdk.language = java telemetry.sdk.name = opentelemetry telemetry.sdk.version = 1.29.0

Warnings (1)
clock skew adjustment disabled; not applying calculated delta of 2.500202475s

SpanID: 0d0953907945e975



solace1: (topic) receive 090795a

Trace Start **January 19 2024, 13:51:23**.703 | Duration **1.05s** | Services **1** | Depth **2** | Total Spans **3**

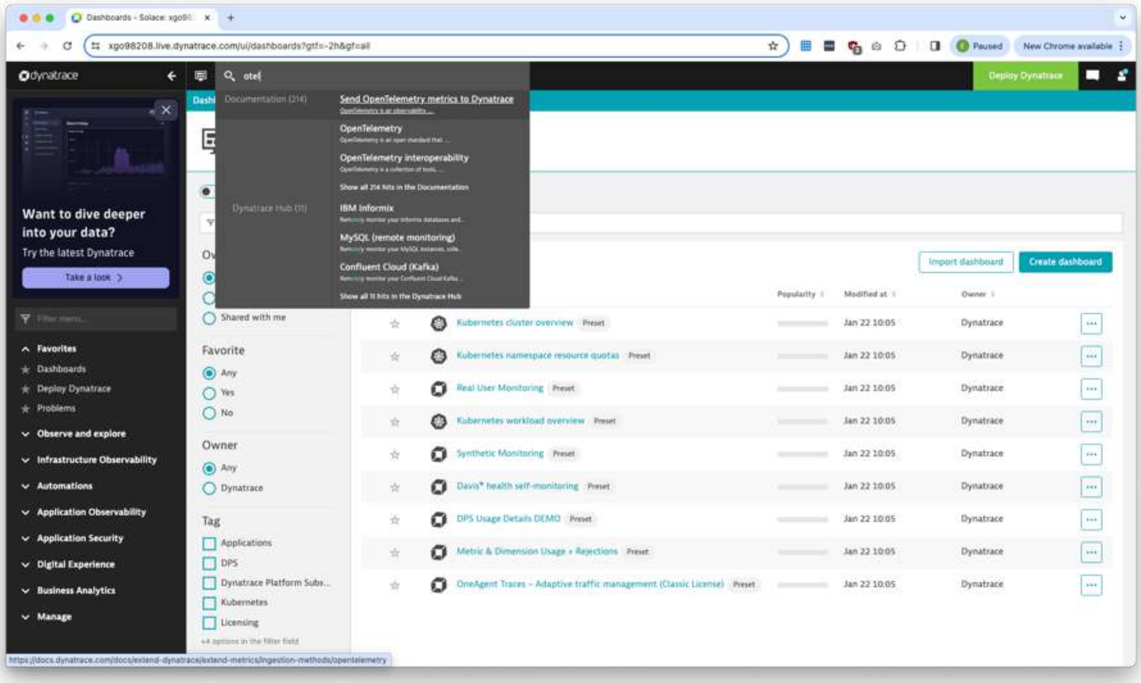
0µs | 262.45ms | 524.9ms | 787.35ms | 1.05s

```
solace1 (topic) receive                  8µs

(topic) receive                          Service: solace1 | Duration: 8µs | Start Time: 0µs
> Tags: internal.span.format = otlp | messaging.destination = demo/trace | messaging.message_payload_size_bytes = 32768 | messaging.operation = receive | messaging.pr...
> Process: service.instance.id = default | service.version = 10.5.0.23
v Logs (3)
  > 4µs: event = queue-trace2 enqueue | messaging.solace.destination_type = queue | messaging.solace.rejects_all_enqueues = false
  > 6µs: event = queue-trace1 enqueue | messaging.solace.destination_type = queue | messaging.solace.rejects_all_enqueues = false
  > 7µs: event = queue-demo enqueue | messaging.solace.destination_type = queue | messaging.solace.rejects_all_enqueues = false
  Log timestamps are relative to the start time of the full trace.
                                                                  SpanID: 4ca8edc83183d4f0

solace1 queue-trace2 send

queue-trace2 send                        Service: solace1 | Duration: 1.05s | Start Time: 139µs
> Tags: internal.span.format = otlp | messaging.operation = send | messaging.protocol = SMF | messaging.protocol_version = 3.0 | messaging.solace.client_name = ip-172-3...
> Process: service.instance.id = default | service.version = 10.5.0.23
                                                                  SpanID: b29e7bea461851e4

solace1 queue-trace1 send

queue-trace1 send                        Service: solace1 | Duration: 1.05s | Start Time: 141µs
> Tags: internal.span.format = otlp | messaging.operation = send | messaging.protocol = SMF | messaging.protocol_version = 3.0 | messaging.solace.client_name = ip-172-3...
> Process: service.instance.id = default | service.version = 10.5.0.23
                                                                  SpanID: df8b65ad8a9c84c4
```

4.2.1 Loki

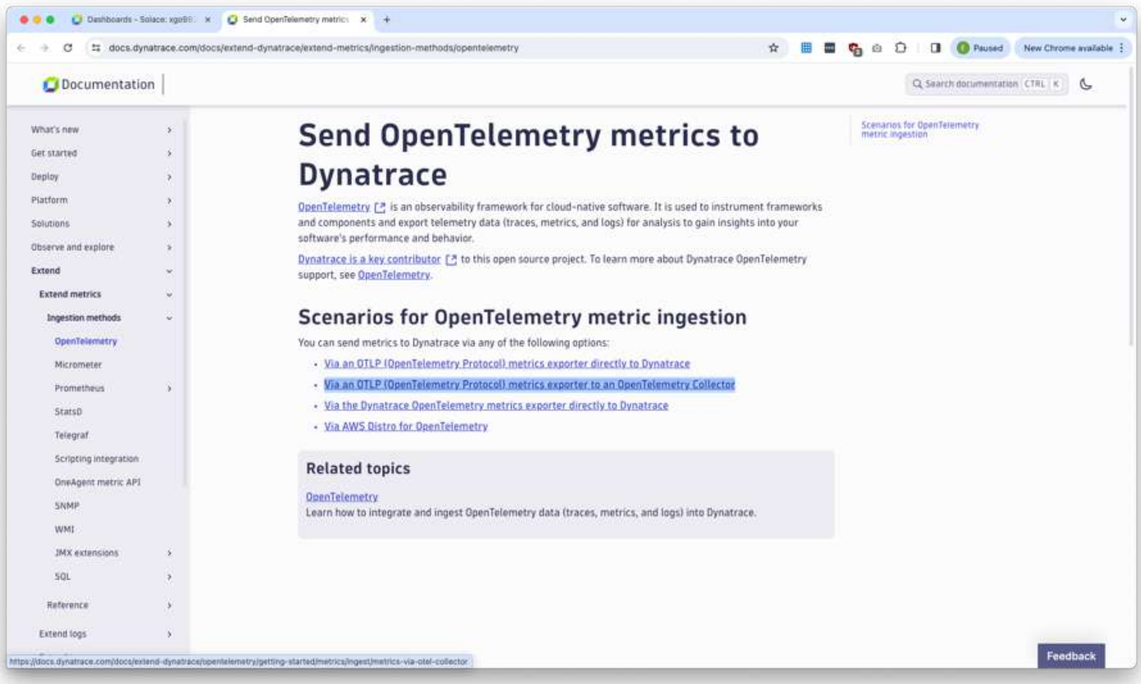
TODO

4.3 Dynatrace


Dashboards - Solace: xgo98
xgo98208.live.dynatrace.com/ui/dashboards?gtf=-2h&gf=all
Paused
New Chrome available
dynatrace
Deploy Dynatrace
otel
Documentation (214)
Send OpenTelemetry metrics to Dynatrace
OpenTelemetry
OpenTelemetry interoperability
Show all 214 hits in the Documentation
Dynatrace Hub (11)
IBM Informix
MySQL (remote monitoring)
Confluent Cloud (Kafka)
Show all 11 hits in the Dynatrace Hub
Want to dive deeper into your data?
Try the latest Dynatrace
Take a look
Filter menu...
Favorites
Dashboards
Deploy Dynatrace
Problems
Observe and explore
Infrastructure Observability
Automations
Application Observability
Application Security
Digital Experience
Business Analytics
Manage
Shared with me
Favorite
Any
Yes
No
Owner
Any
Dynatrace
Tag
Applications
DPS
Dynatrace Platform Subs...
Kubernetes
Licensing
+4 options in the filter field
Import dashboard
Create dashboard
https://docs.dynatrace.com/docs/extend-dynatrace/extend-metrics/ingestion-methods/opentelemetry

	Popularity	Modified at	Owner
Kubernetes cluster overview Preset		Jan 22 10:05	Dynatrace
Kubernetes namespace resource quotas Preset		Jan 22 10:05	Dynatrace
Real User Monitoring Preset		Jan 22 10:05	Dynatrace
Kubernetes workload overview Preset		Jan 22 10:05	Dynatrace
Synthetic Monitoring Preset		Jan 22 10:05	Dynatrace
Davis® health self-monitoring Preset		Jan 22 10:05	Dynatrace
DPS Usage Details DEMO Preset		Jan 22 10:05	Dynatrace
Metric & Dimension Usage + Rejections Preset		Jan 22 10:05	Dynatrace
OneAgent Traces – Adaptive traffic management (Classic License) Preset		Jan 22 10:05	Dynatrace

https://docs.dynatrace.com/docs/extend-dynatrace/extend-metrics/ingestion-methods/opentelemetry



Dashboards - Solace: xgo98 | Send OpenTelemetry metrics

docs.dynatrace.com/docs/extend-dynatrace/extend-metrics/ingestion-methods/opentelemetry

Documentation

What's new
Get started
Deploy
Platform
Solutions
Observe and explore
Extend
Extend metrics
Ingestion methods
OpenTelemetry
Micrometer
Prometheus
StatsD
Telegraf
Scripting integration
OneAgent metric API
SNMP
WMI
JMX extensions
SQL
Reference
Extend logs

Send OpenTelemetry metrics to Dynatrace

OpenTelemetry is an observability framework for cloud-native software. It is used to instrument frameworks and components and export telemetry data (traces, metrics, and logs) for analysis to gain insights into your software's performance and behavior.

Dynatrace is a key contributor to this open source project. To learn more about Dynatrace OpenTelemetry support, see OpenTelemetry.

Scenarios for OpenTelemetry metric ingestion

You can send metrics to Dynatrace via any of the following options:

- Via an OTLP (OpenTelemetry Protocol) metrics exporter directly to Dynatrace
- Via an OTLP (OpenTelemetry Protocol) metrics exporter to an OpenTelemetry Collector
- Via the Dynatrace OpenTelemetry metrics exporter directly to Dynatrace
- Via AWS Distro for OpenTelemetry

Related topics

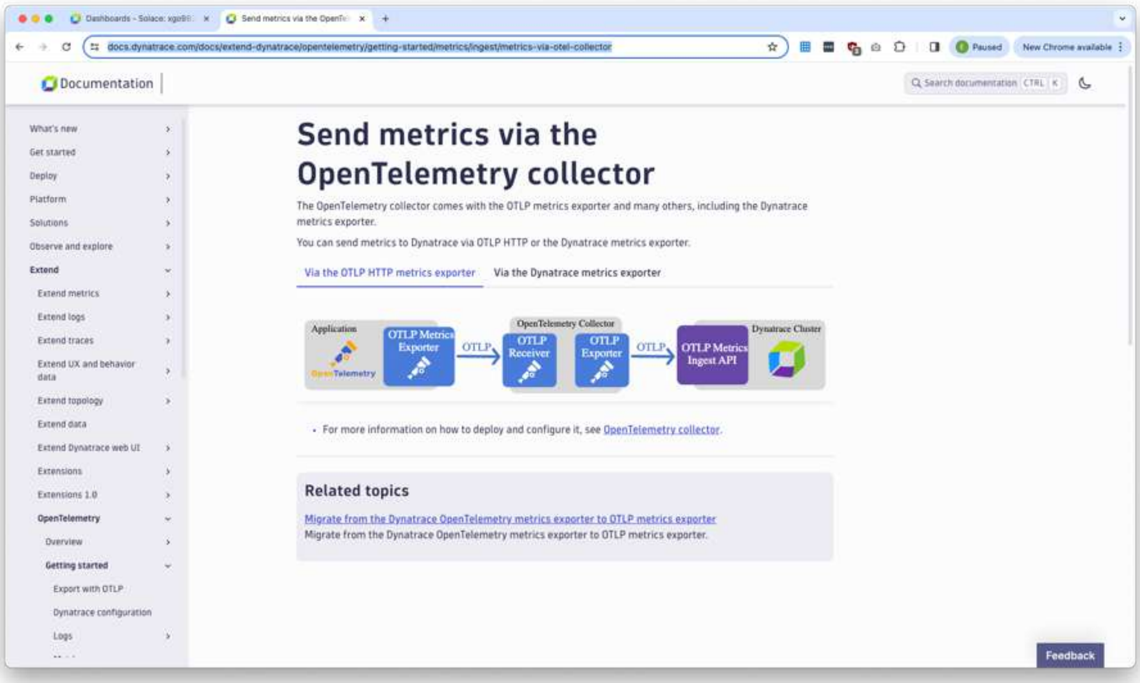
OpenTelemetry
Learn how to integrate and ingest OpenTelemetry data (traces, metrics, and logs) into Dynatrace.

Scenarios for OpenTelemetry metric ingestion

https://docs.dynatrace.com/docs/extend-dynatrace/opentelemetry/getting-started/metrics/ingest/metrics-via-otel-collector

Feedback

https://docs.dynatrace.com/docs/extend-dynatrace/opentelemetry/getting-started/metrics/ingest/metrics-via-otel-collector



Dashboards - Solace: xgo98 | Send metrics via the OpenTe

docs.dynatrace.com/docs/extend-dynatrace/opentelemetry/getting-started/metrics/ingest/metrics-via-otel-collector

Documentation

What's new
Get started
Deploy
Platform
Solutions
Observe and explore
Extend
Extend metrics
Extend logs
Extend traces
Extend UX and behavior data
Extend topology
Extend data
Extend Dynatrace web UI
Extensions
Extensions 1.0
OpenTelemetry
Overview
Getting started
Export with OTLP
Dynatrace configuration
Logs

Send metrics via the OpenTelemetry collector

The OpenTelemetry collector comes with the OTLP metrics exporter and many others, including the Dynatrace metrics exporter.

You can send metrics to Dynatrace via OTLP HTTP or the Dynatrace metrics exporter.

Via the OTLP HTTP metrics exporter | Via the Dynatrace metrics exporter

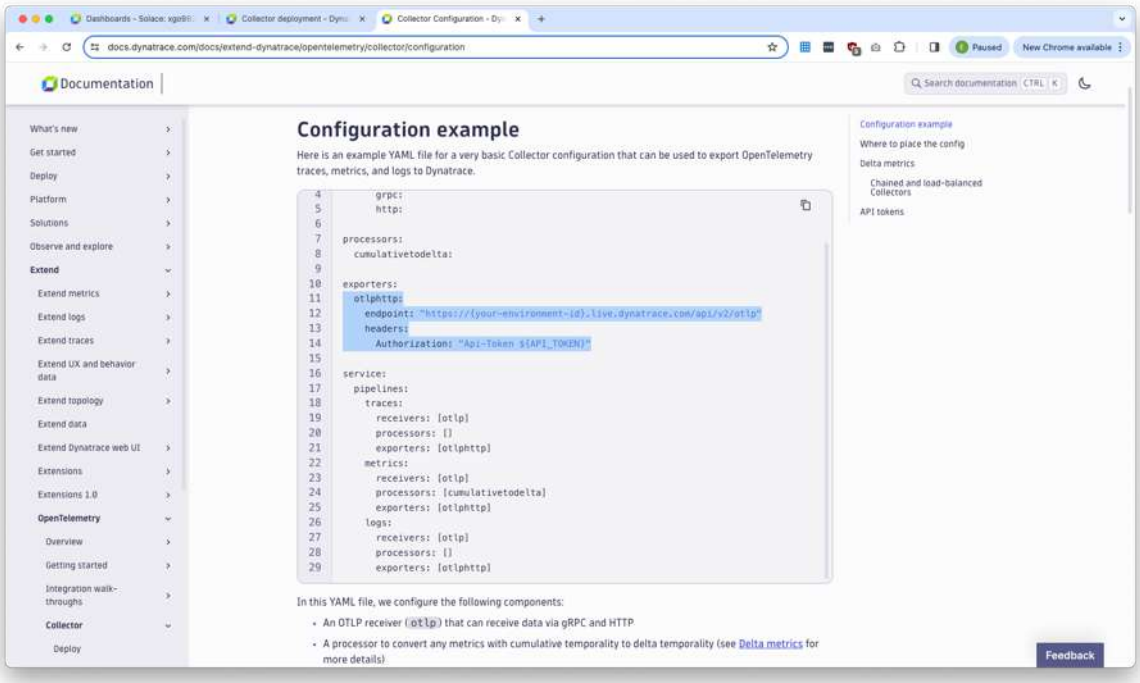
- For more information on how to deploy and configure it, see OpenTelemetry collector.

Related topics

Migrate from the Dynatrace OpenTelemetry metrics exporter to OTLP metrics exporter
Migrate from the Dynatrace OpenTelemetry metrics exporter to OTLP metrics exporter.

Feedback

https://docs.dynatrace.com/docs/extend-dynatrace/opentelemetry/collector#example-configuration



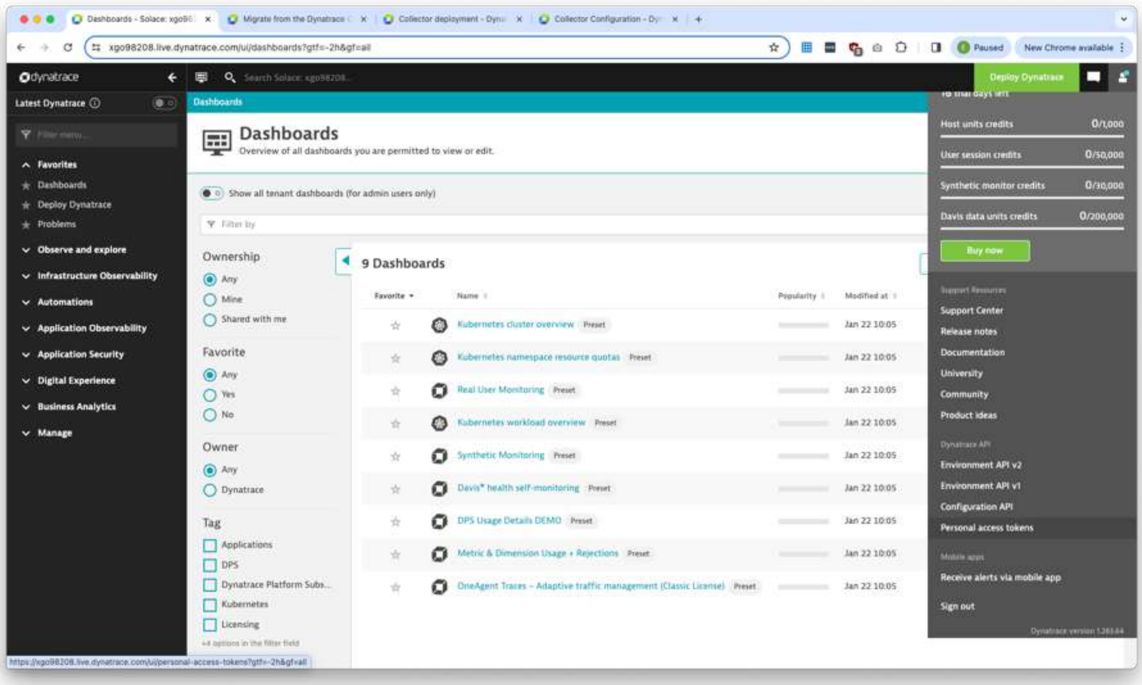
Configuration example

Here is an example YAML file for a very basic Collector configuration that can be used to export OpenTelemetry traces, metrics, and logs to Dynatrace.

```
      grpc:
      http:

processors:
  cumulativetodelta:

exporters:
  otlphttp:
    endpoint: "https://{your-environment-id}.live.dynatrace.com/api/v2/otlp"
    headers:
      Authorization: "Api-Token ${API_TOKEN}"

service:
  pipelines:
    traces:
      receivers: [otlp]
      processors: []
      exporters: [otlphttp]
    metrics:
      receivers: [otlp]
      processors: [cumulativetodelta]
      exporters: [otlphttp]
    logs:
      receivers: [otlp]
      processors: []
      exporters: [otlphttp]
```

In this YAML file, we configure the following components:

- An OTLP receiver (`otlp`) that can receive data via gRPC and HTTP
- A processor to convert any metrics with cumulative temporality to delta temporality (see Delta metrics for more details)


Dashboards - Solace: xgo98
Migrate from the Dynatrace
Collector deployment - Dyn
Collector Configuration - Dy
xgo98208.live.dynatrace.com/ui/dashboards?gtf=-2h&gf=all
Paused
New Chrome available
dynatrace
Search Solace: xgo98208...
Deploy Dynatrace
Latest Dynatrace
Filter menu...
Favorites
Dashboards
Deploy Dynatrace
Problems
Observe and explore
Infrastructure Observability
Automations
Application Observability
Application Security
Digital Experience
Business Analytics
Manage
Dashboards

Dashboards

Overview of all dashboards you are permitted to view or edit.

Show all tenant dashboards (for admin users only)

Filter by

Ownership: [✓ Any] Mine / Shared with me

Favorite: [✓ Any] Yes / No

Owner: [✓ Any] Dynatrace

Tag: Applications / DPS / Dynatrace Platform Subs... / Kubernetes / Licensing

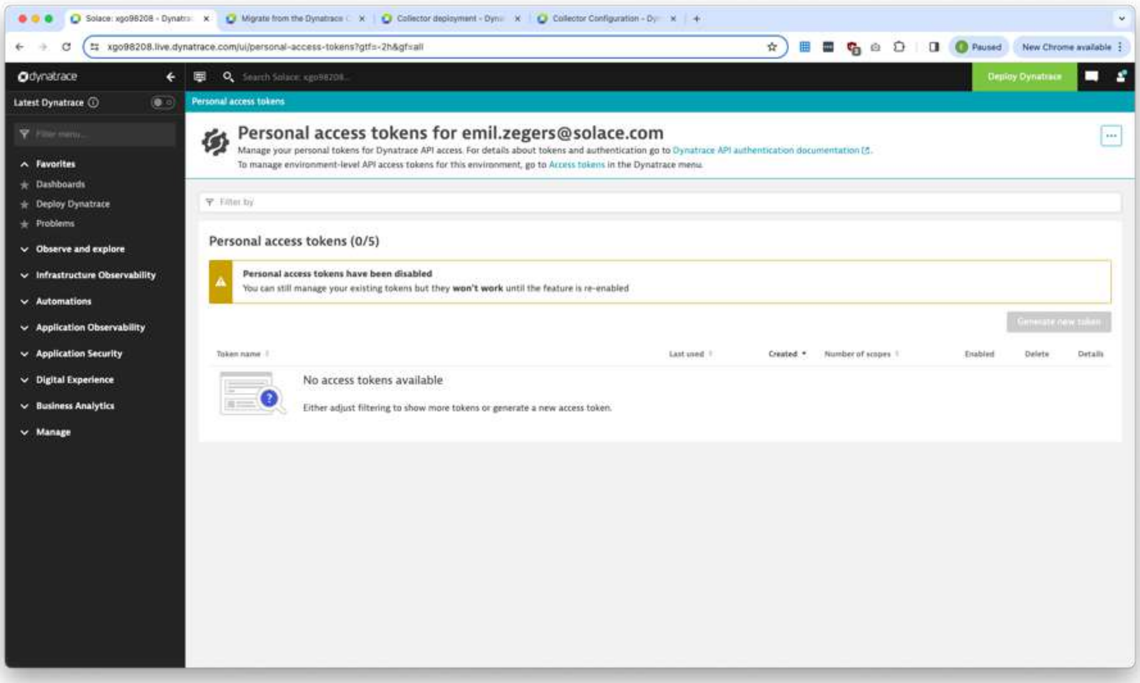
+4 options in the filter field

9 Dashboards

Favorite	Name	Popularity	Modified at
☆	Kubernetes cluster overview Preset		Jan 22 10:05
☆	Kubernetes namespace resource quotas Preset		Jan 22 10:05
☆	Real User Monitoring Preset		Jan 22 10:05
☆	Kubernetes workload overview Preset		Jan 22 10:05
☆	Synthetic Monitoring Preset		Jan 22 10:05
☆	Davis® health self-monitoring Preset		Jan 22 10:05
☆	DPS Usage Details DEMO Preset		Jan 22 10:05
☆	Metric & Dimension Usage + Rejections Preset		Jan 22 10:05
☆	OneAgent Traces – Adaptive traffic management (Classic License) Preset		Jan 22 10:05

16 trial days left

Host units credits	0/1,000
User session credits	0/50,000
Synthetic monitor credits	0/30,000
Davis data units credits	0/200,000



xgo98208.live.dynatrace.com/ui/personal-access-tokens?gtf=-2h&gf=all

Personal access tokens

Personal access tokens for emil.zegers@solace.com

Manage your personal tokens for Dynatrace API access. For details about tokens and authentication go to Dynatrace API authentication documentation.
To manage environment-level API access tokens for this environment, go to Access tokens in the Dynatrace menu.

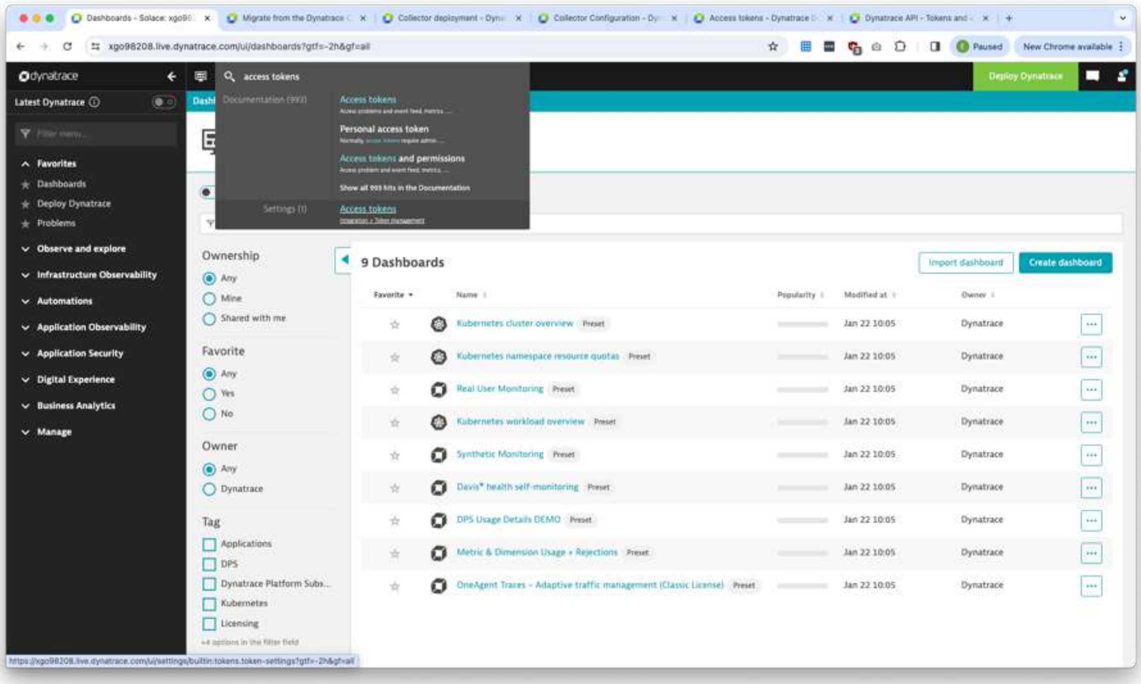
Personal access tokens (0/5)

Personal access tokens have been disabled
You can still manage your existing tokens but they **won't work** until the feature is re-enabled

Token name	Last used	Created	Number of scopes	Enabled	Delete	Details

No access tokens available

Either adjust filtering to show more tokens or generate a new access token.



xgo98208.live.dynatrace.com/ui/dashboards?gtf=-2h&gf=all

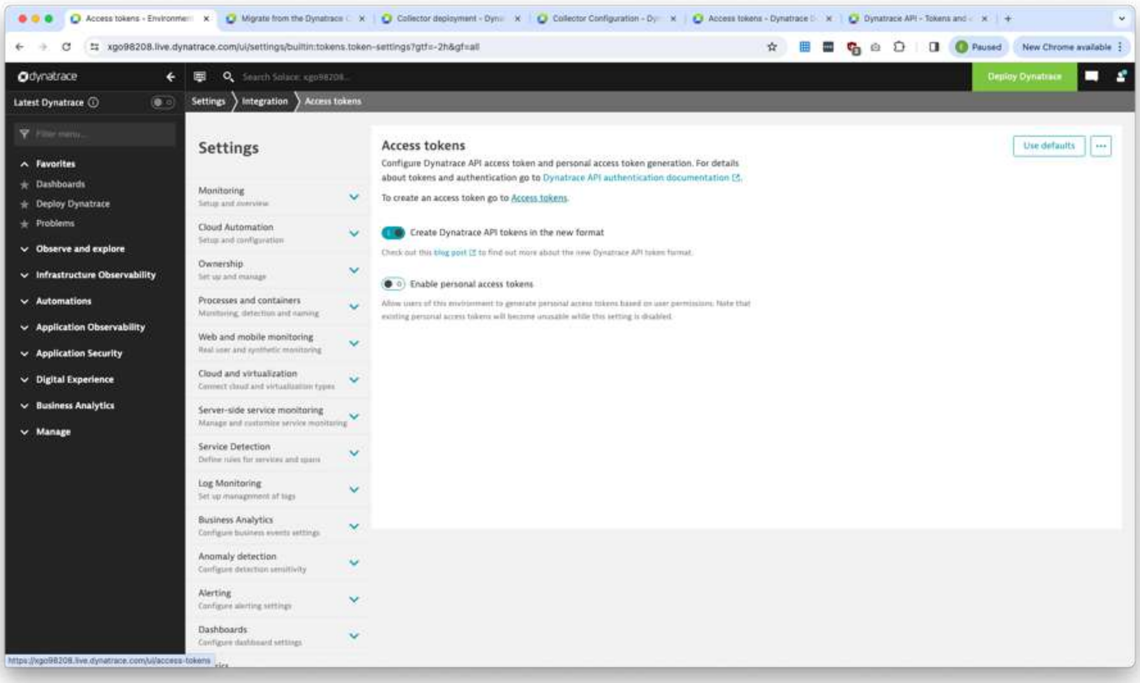
access tokens

Documentation (993)
Access tokens
Personal access token
Access tokens and permissions
Show all 993 hits in the Documentation
Settings (1)
Access tokens
Integration > Token management

9 Dashboards

Favorite	Name	Popularity	Modified at	Owner
☆	Kubernetes cluster overview Preset		Jan 22 10:05	Dynatrace
☆	Kubernetes namespace resource quotas Preset		Jan 22 10:05	Dynatrace
☆	Real User Monitoring Preset		Jan 22 10:05	Dynatrace
☆	Kubernetes workload overview Preset		Jan 22 10:05	Dynatrace
☆	Synthetic Monitoring Preset		Jan 22 10:05	Dynatrace
☆	Davis® health self-monitoring Preset		Jan 22 10:05	Dynatrace
☆	DPS Usage Details DEMO Preset		Jan 22 10:05	Dynatrace
☆	Metric & Dimension Usage + Rejections Preset		Jan 22 10:05	Dynatrace
☆	OneAgent Traces - Adaptive traffic management (Classic License) Preset		Jan 22 10:05	Dynatrace

https://xgo98208.live.dynatrace.com/ui/settings/builtin:tokens.token-settings?gtf=-2h&gf=all



Access tokens

Configure Dynatrace API access token and personal access token generation. For details about tokens and authentication go to Dynatrace API authentication documentation.

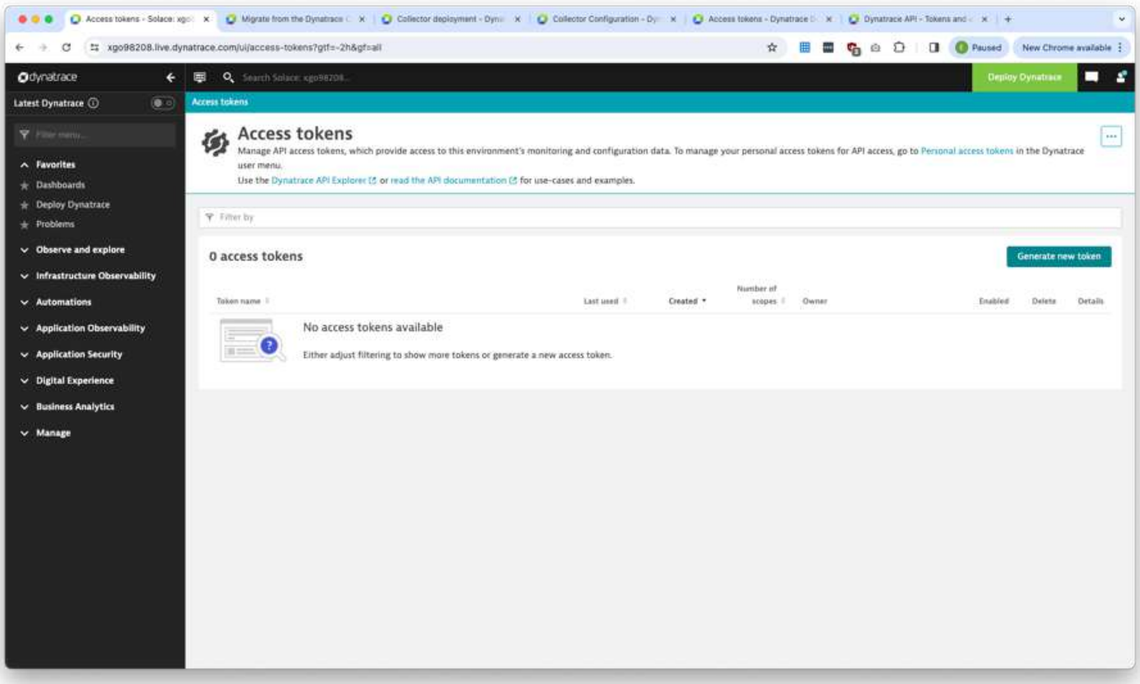
To create an access token go to Access tokens.

Create Dynatrace API tokens in the new format

Check out this blog post to find out more about the new Dynatrace API token format.

Enable personal access tokens

Allow users of this environment to generate personal access tokens based on user permissions. Note that existing personal access tokens will become unusable while this setting is disabled.



Access tokens

Manage API access tokens, which provide access to this environment's monitoring and configuration data. To manage your personal access tokens for API access, go to Personal access tokens in the Dynatrace user menu.

Use the Dynatrace API Explorer or read the API documentation for use-cases and examples.

0 access tokens

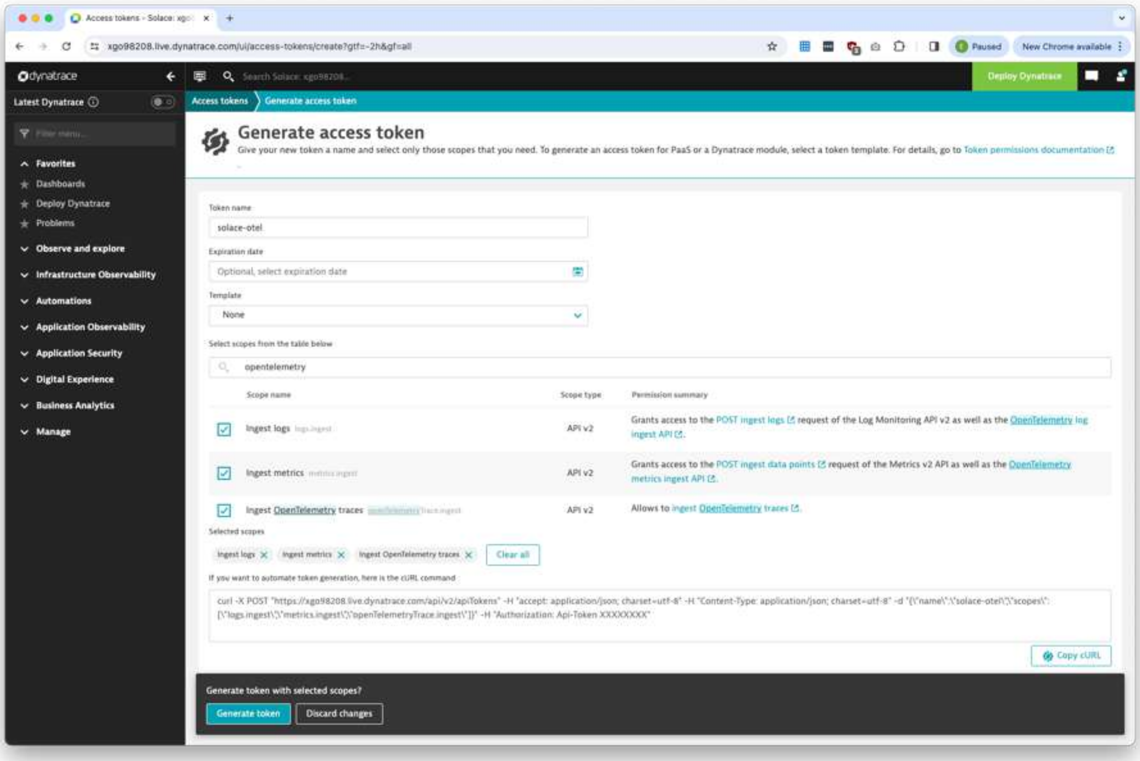
Token name	Last used	Created	Number of scopes	Owner	Enabled	Delete	Details

No access tokens available

Either adjust filtering to show more tokens or generate a new access token.

Token name: solace-otel

Search for OpenTelemetry and add Scopes



Generate access token

Give your new token a name and select only those scopes that you need. To generate an access token for PaaS or a Dynatrace module, select a token template. For details, go to Token permissions documentation

Token name: solace-otel

Expiration date: Optional, select expiration date

Template: None

Select scopes from the table below: opentelemetry

Scope name	Scope type	Permission summary
✓ Ingest logs logs.ingest	API v2	Grants access to the POST ingest logs request of the Log Monitoring API v2 as well as the OpenTelemetry log ingest API.
✓ Ingest metrics metrics.ingest	API v2	Grants access to the POST ingest data points request of the Metrics v2 API as well as the OpenTelemetry metrics ingest API.
✓ Ingest OpenTelemetry traces openTelemetryTrace.ingest	API v2	Allows to ingest OpenTelemetry traces.

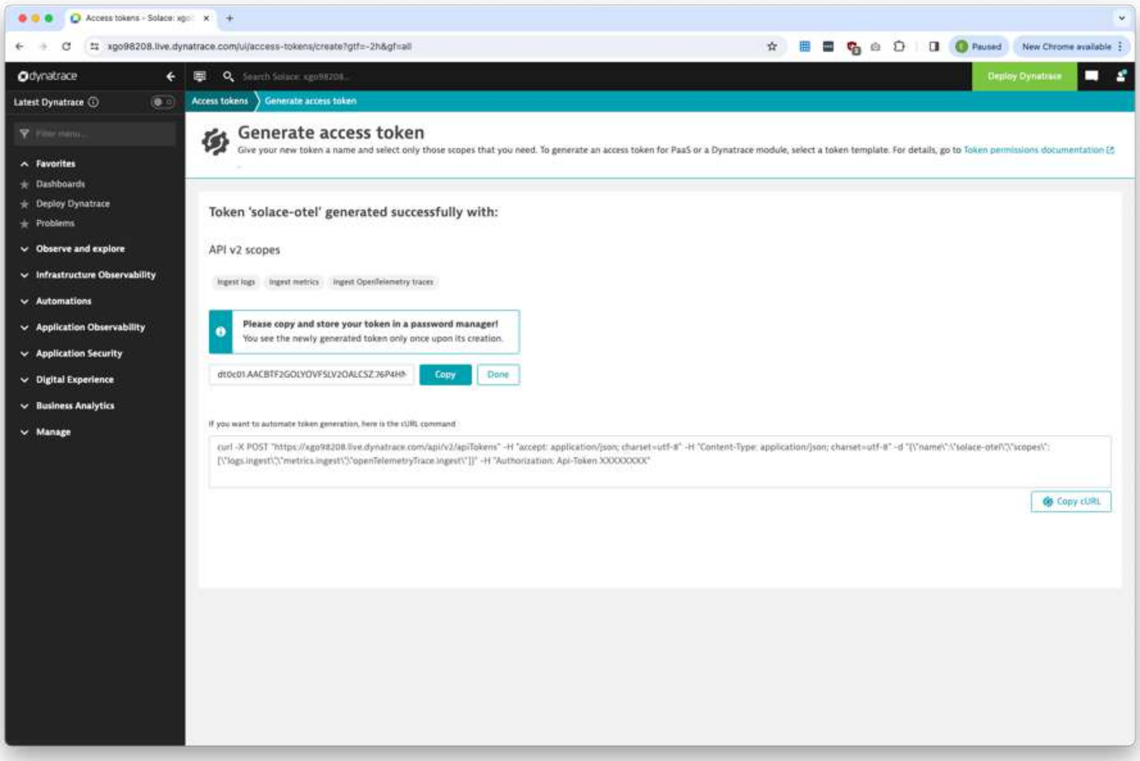
Selected scopes: Ingest logs, Ingest metrics, Ingest OpenTelemetry traces

If you want to automate token generation, here is the cURL command

```
curl -X POST "https://xgo98208.live.dynatrace.com/api/v2/apiTokens" -H "accept: application/json; charset=utf-8" -H "Content-Type: application/json; charset=utf-8" -d "{\"name\":\"solace-otel\",\"scopes\":
[\"logs.ingest\",\"metrics.ingest\",\"openTelemetryTrace.ingest\"]}" -H "Authorization: Api-Token XXXXXXXX"
```

Generate token with selected scopes?

Click [Generate token]



Generate access token

Give your new token a name and select only those scopes that you need. To generate an access token for PaaS or a Dynatrace module, select a token template. For details, go to Token permissions documentation

Token 'solace-otel' generated successfully with:

API v2 scopes

Ingest logs, Ingest metrics, Ingest OpenTelemetry traces

Please copy and store your token in a password manager!
You see the newly generated token only once upon its creation.

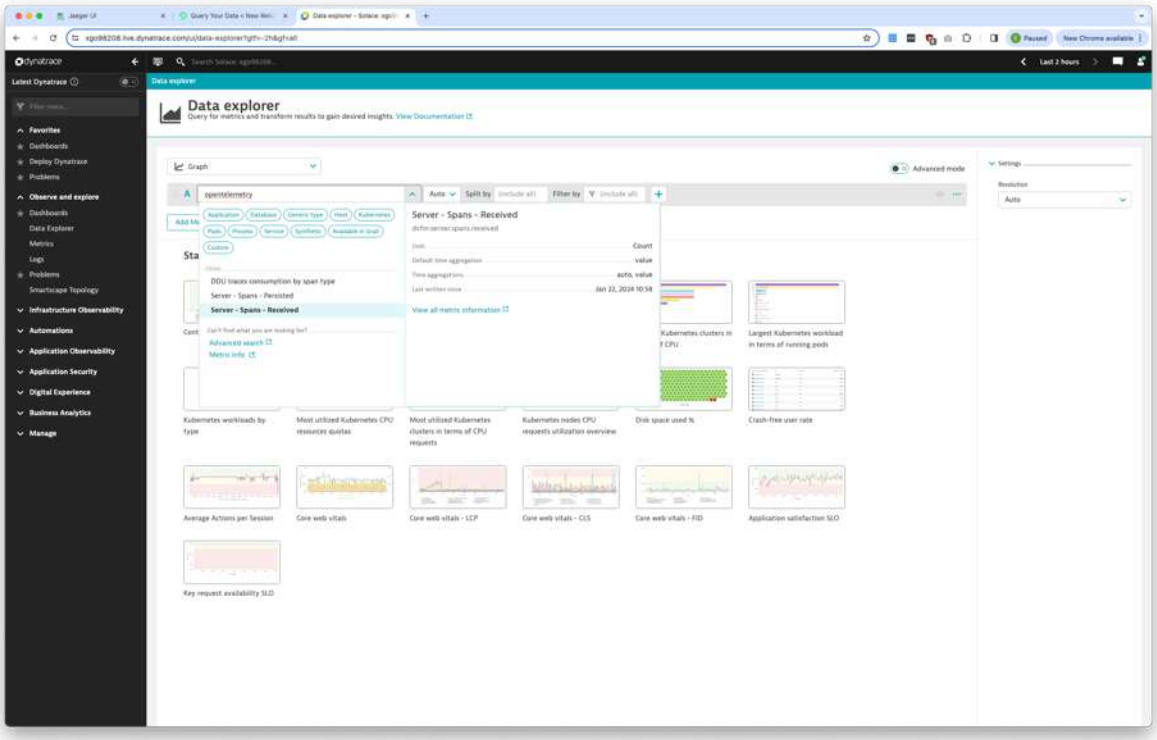
dt0c01.AACBTF2GOLYOVFSLV2OALCSZ.76P4HN

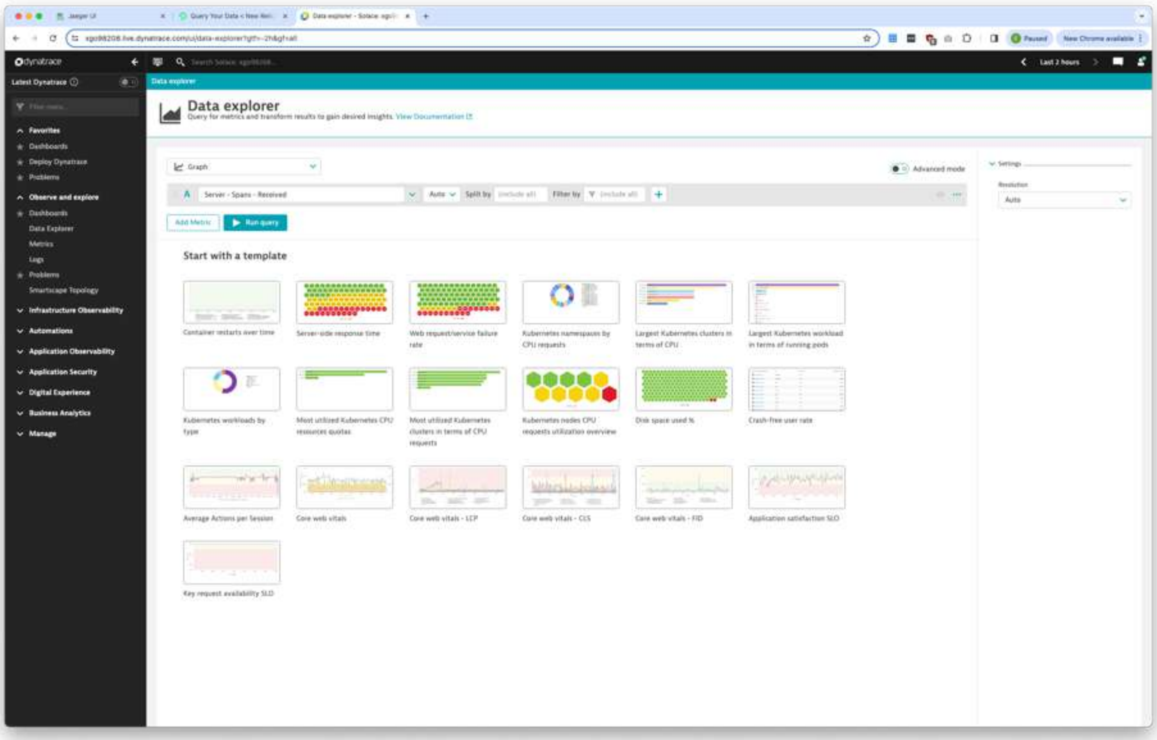
If you want to automate token generation, here is the cURL command

```
curl -X POST "https://xgo98208.live.dynatrace.com/api/v2/apiTokens" -H "accept: application/json; charset=utf-8" -H "Content-Type: application/json; charset=utf-8" -d "{\"name\":\"solace-otel\",\"scopes\":
[\"logs.ingest\",\"metrics.ingest\",\"openTelemetryTrace.ingest\"]}" -H "Authorization: Api-Token XXXXXXXX"
```

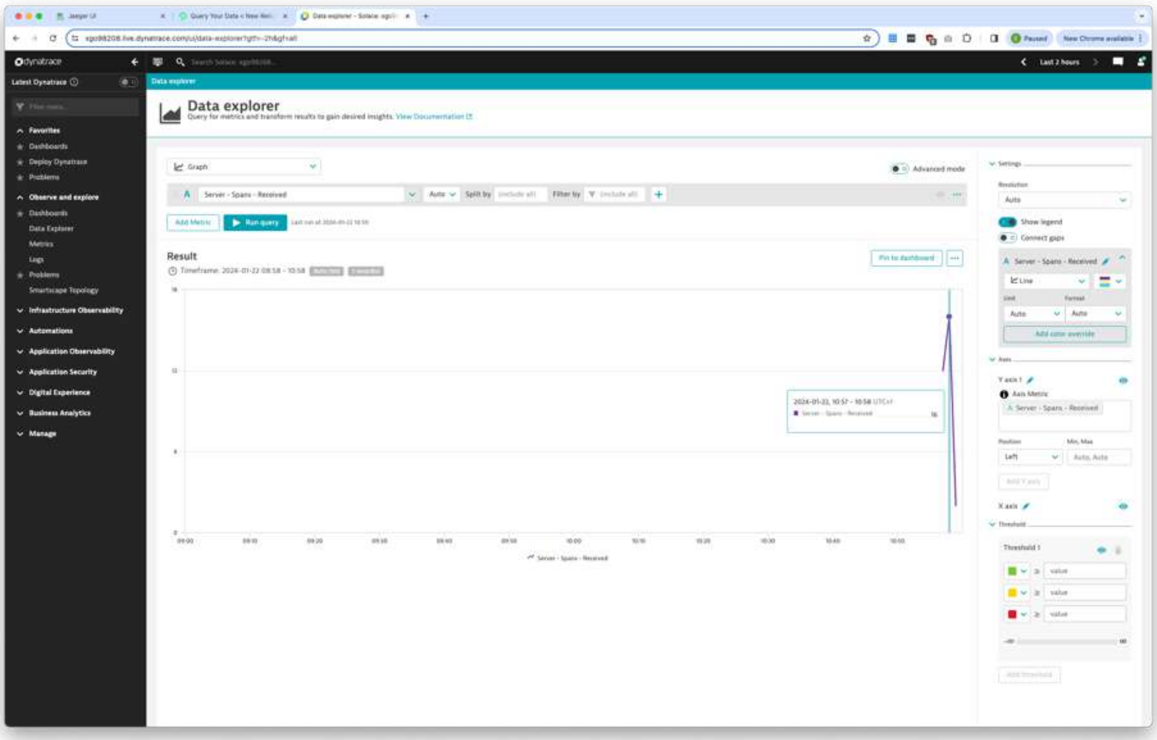
Copy token and add to YAML file.

Now send some events with SDKPerf resulting in traces in Dynatrace.

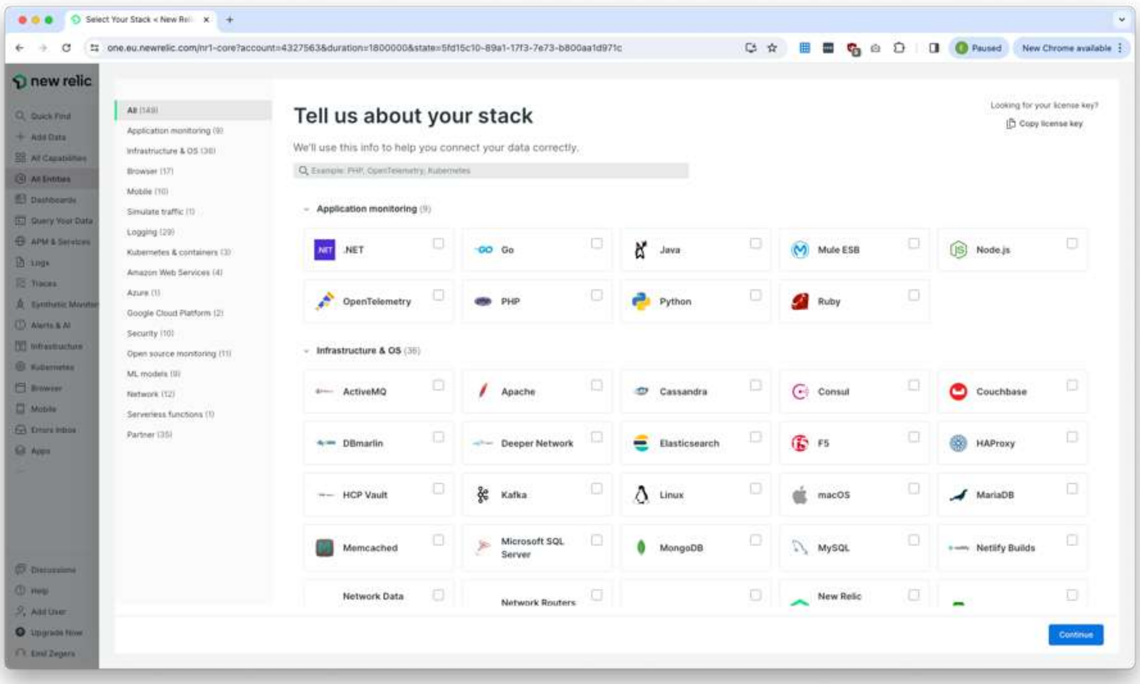

Data explorer
Query for metrics and transform results to gain desired insights. View Documentation
opentelemetry
Server - Spans - Received
DDU traces consumption by span type
Server - Spans - Persisted
View all metric information

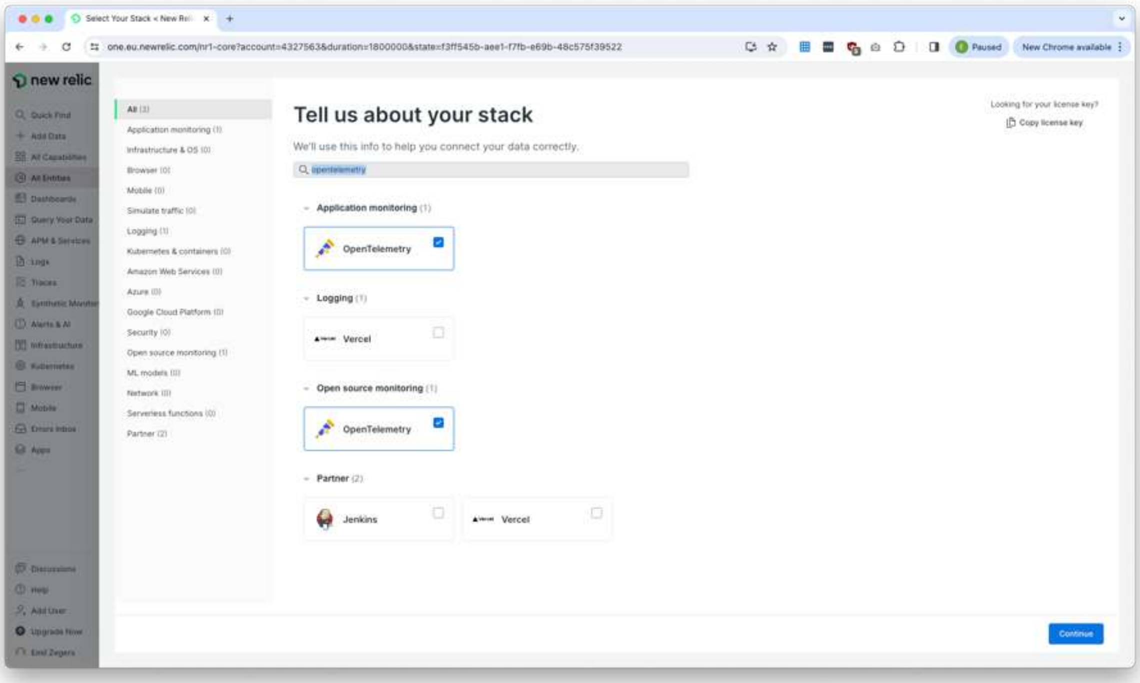


Click [Run query]


Data explorer
Query for metrics and transform results to gain desired insights. View Documentation
Graph
Advanced mode
Server - Spans - Received
Add Metric
Run query
Result
Pin to dashboard
Settings
Show legend
Connect gaps
Add color override
Threshold
Favorites
Dashboards
Problems
Observe and explore
Data Explorer
Metrics
Logs
Smartscape Topology
Infrastructure Observability
Automations
Application Observability
Application Security
Digital Experience
Business Analytics
Manage
Last 2 hours

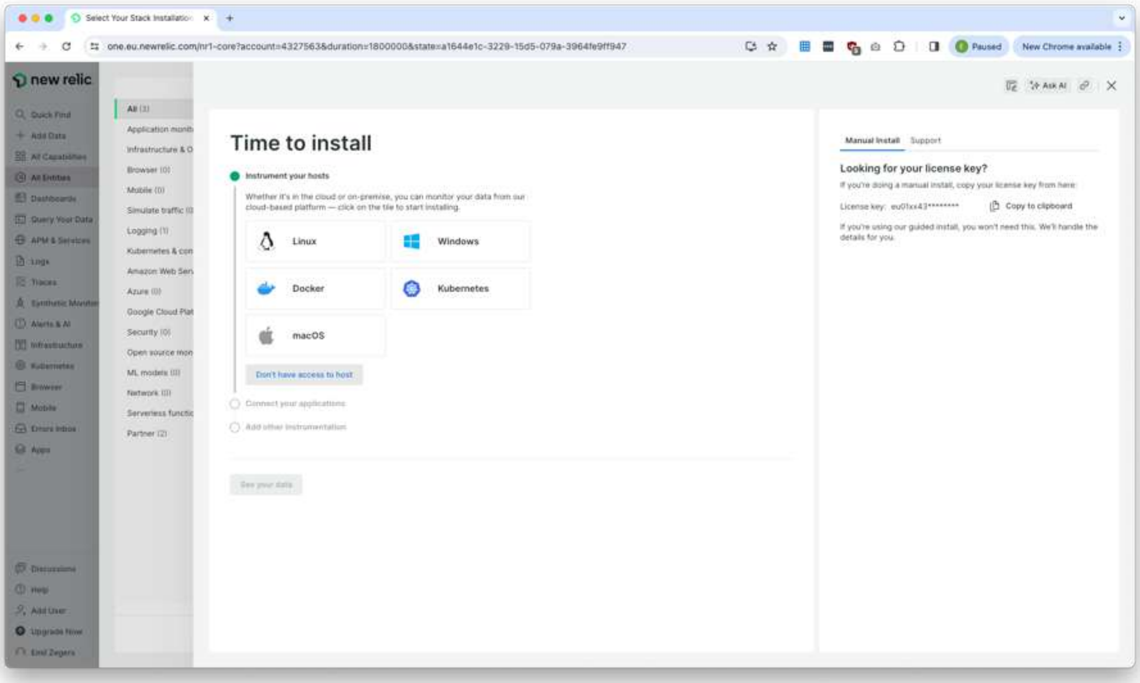
4.4 New Relic


one.eu.newrelic.com/nr1-core?account=4327563&duration=1800000&state=5fd15c10-89a1-17f3-7e73-b800aa1d971c
new relic
Tell us about your stack
We'll use this info to help you connect your data correctly.
Looking for your license key?
Copy license key
Application monitoring (9)
.NET
Go
Java
Mule ESB
Node.js
OpenTelemetry
PHP
Python
Ruby
Infrastructure & OS (38)
ActiveMQ
Apache
Cassandra
Consul
Couchbase
DBmarlin
Deeper Network
Elasticsearch
F5
HAProxy
HCP Vault
Kafka
Linux
macOS
MariaDB
Memcached
Microsoft SQL Server
MongoDB
MySQL
Netlify Builds
Network Data
Network Routers
New Relic
Continue



Tell us about your stack

We'll use this info to help you connect your data correctly.


Select Your Stack Installation
one.eu.newrelic.com/nr1-core?account=4327563&duration=1800000&state=a1644e1c-3229-15d5-079a-3964fe9ff947
Paused
New Chrome available
new relic
Ask AI
Manual install
Support
Instrument your hosts
Linux
Windows
Docker
Kubernetes
macOS
Don't have access to host
Connect your applications
Add other instrumentation
See your data
Copy to clipboard

Time to install

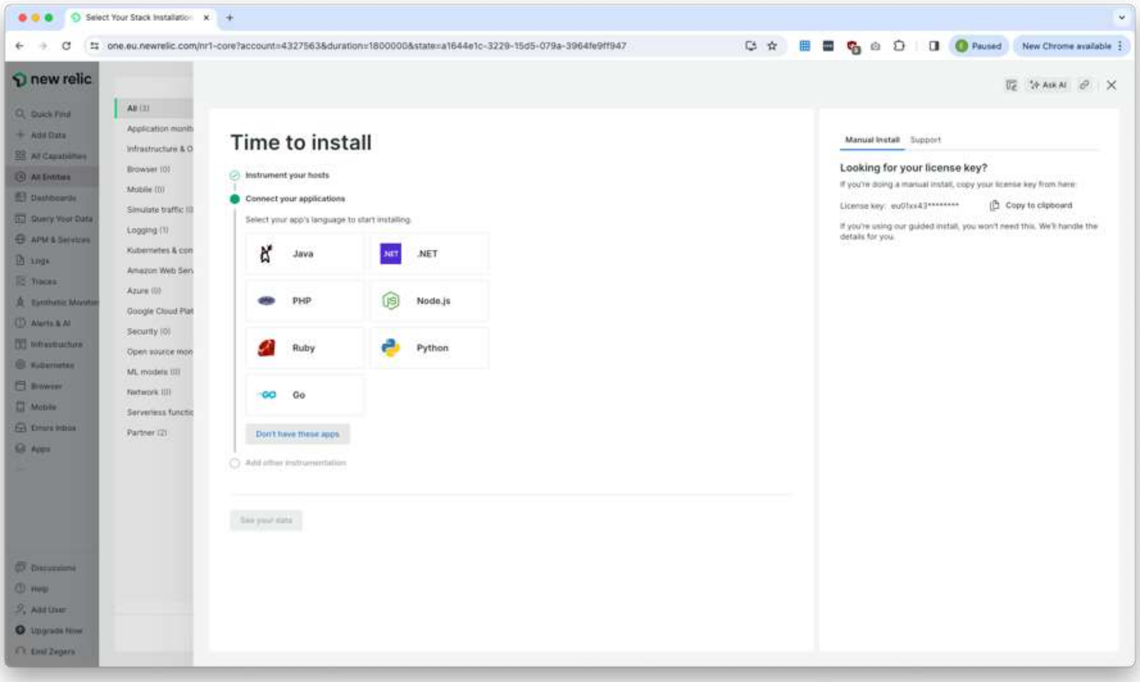
Whether it's in the cloud or on-premise, you can monitor your data from our cloud-based platform — click on the tile to start installing.

Looking for your license key?

If you're doing a manual install, copy your license key from here:

License key: eu01xx43********

If you're using our guided install, you won't need this. We'll handle the details for you.



Time to install

- Instrument your hosts
- Connect your applications

Select your app's language to start installing.

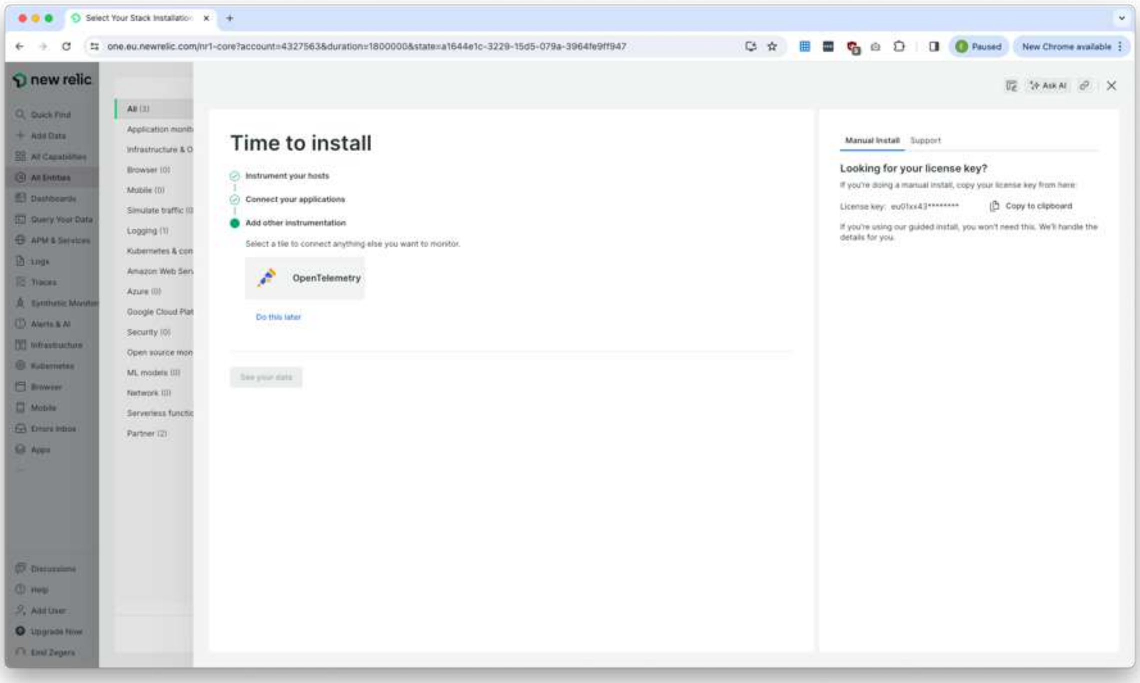
- Add other instrumentation

Looking for your license key?

If you're doing a manual install, copy your license key from here:

License key: eu01xx43********

If you're using our guided install, you won't need this. We'll handle the details for you.


Select Your Stack Installation
one.eu.newrelic.com/nr1-core?account=4327563&duration=1800000&state=a1644e1c-3229-15d5-079a-3964fe9ff947
new relic
Ask AI
OpenTelemetry
Do this later
See your data
Manual Install
Support
Copy to clipboard

Time to install

- Instrument your hosts
- Connect your applications
- Add other instrumentation

Select a tile to connect anything else you want to monitor.

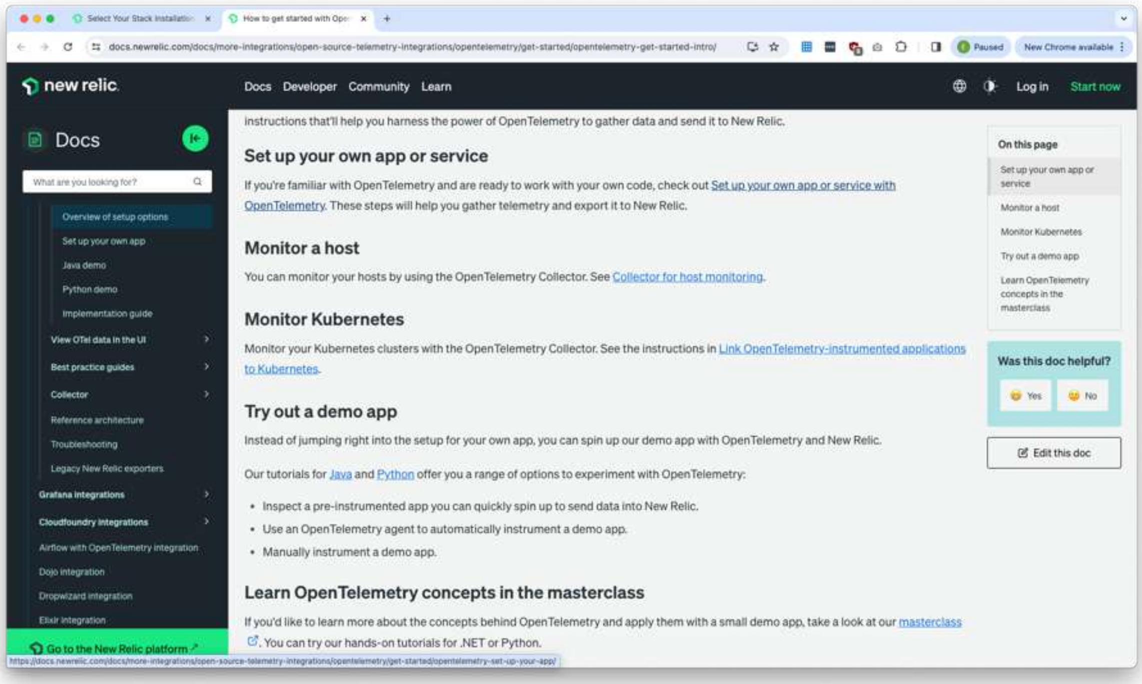
Looking for your license key?

If you're doing a manual install, copy your license key from here:

License key: eu01xx43********

If you're using our guided install, you won't need this. We'll handle the details for you.

https://docs.newrelic.com/docs/more-integrations/open-source-telemetry-integrations/opentelemetry/get-started/opentelemetry-get-started-intro/



instructions that'll help you harness the power of OpenTelemetry to gather data and send it to New Relic.

Set up your own app or service

If you're familiar with OpenTelemetry and are ready to work with your own code, check out Set up your own app or service with OpenTelemetry. These steps will help you gather telemetry and export it to New Relic.

Monitor a host

You can monitor your hosts by using the OpenTelemetry Collector. See Collector for host monitoring.

Monitor Kubernetes

Monitor your Kubernetes clusters with the OpenTelemetry Collector. See the instructions in Link OpenTelemetry-instrumented applications to Kubernetes.

Try out a demo app

Instead of jumping right into the setup for your own app, you can spin up our demo app with OpenTelemetry and New Relic.

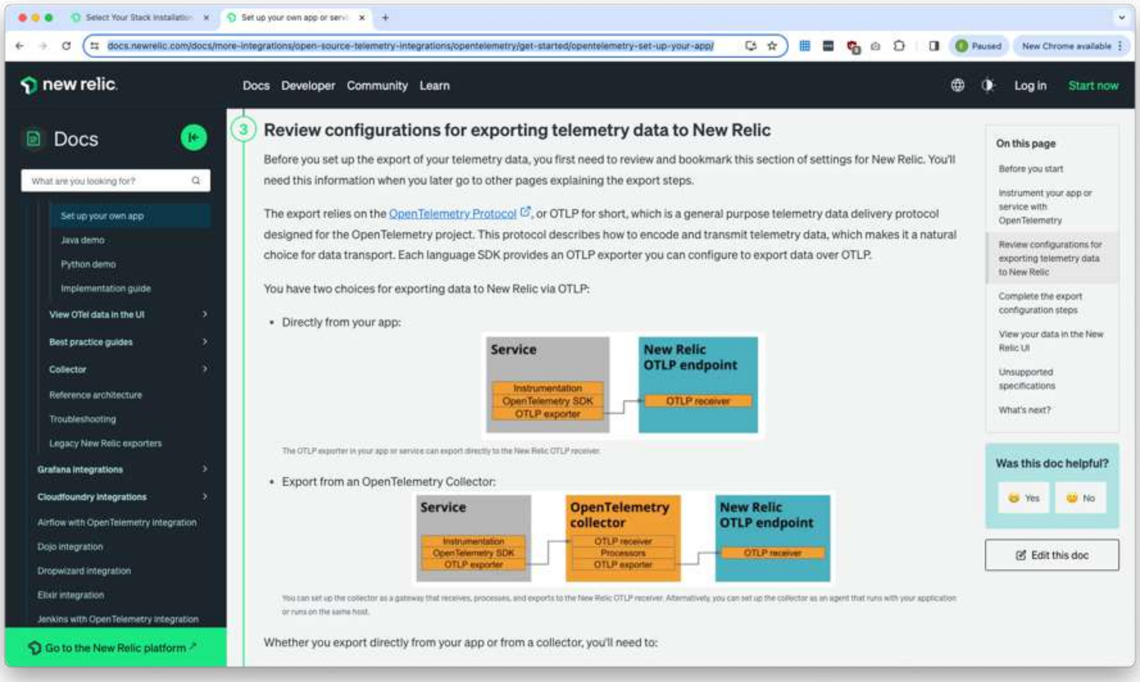
Our tutorials for Java and Python offer you a range of options to experiment with OpenTelemetry:

- Inspect a pre-instrumented app you can quickly spin up to send data into New Relic.
- Use an OpenTelemetry agent to automatically instrument a demo app.
- Manually instrument a demo app.

Learn OpenTelemetry concepts in the masterclass

If you'd like to learn more about the concepts behind OpenTelemetry and apply them with a small demo app, take a look at our masterclass. You can try our hands-on tutorials for .NET or Python.

https://docs.newrelic.com/docs/more-integrations/open-source-telemetry-integrations/opentelemetry/get-started/opentelemetry-set-up-your-app/



3 Review configurations for exporting telemetry data to New Relic

Before you set up the export of your telemetry data, you first need to review and bookmark this section of settings for New Relic. You'll need this information when you later go to other pages explaining the export steps.

The export relies on the OpenTelemetry Protocol, or OTLP for short, which is a general purpose telemetry data delivery protocol designed for the OpenTelemetry project. This protocol describes how to encode and transmit telemetry data, which makes it a natural choice for data transport. Each language SDK provides an OTLP exporter you can configure to export data over OTLP.

You have two choices for exporting data to New Relic via OTLP:

- Directly from your app:

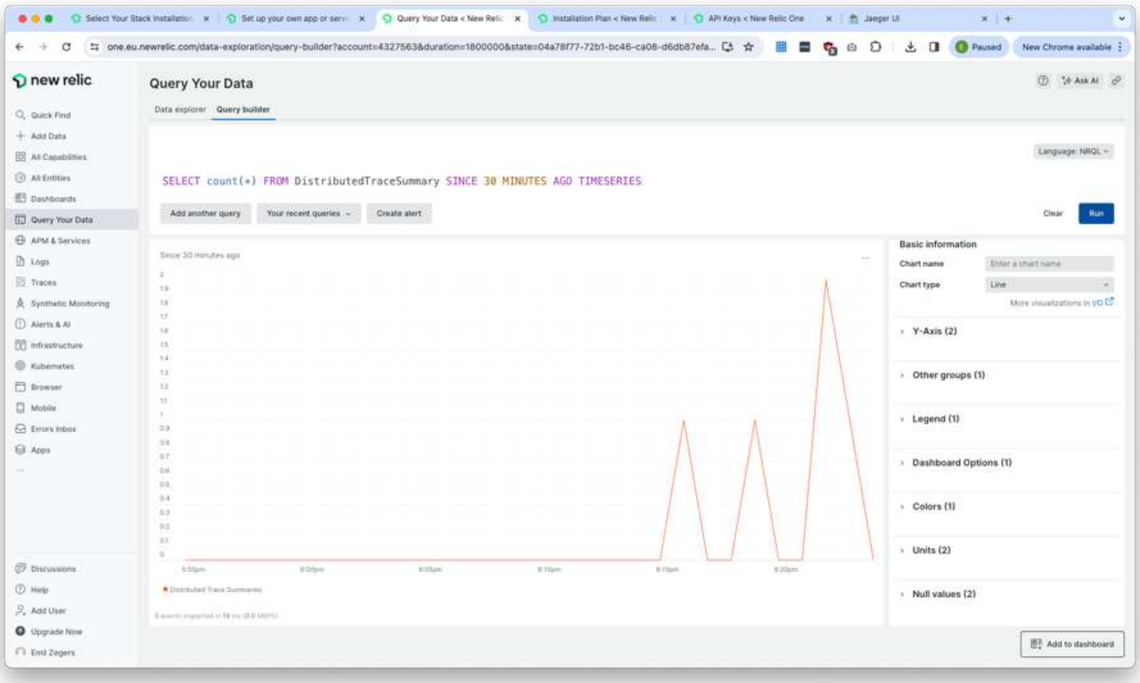
The OTLP exporter in your app or service can export directly to the New Relic OTLP receiver.

- Export from an OpenTelemetry Collector:

You can set up the collector as a gateway that receives, processes, and exports to the New Relic OTLP receiver. Alternatively, you can set up the collector as an agent that runs with your application or runs on the same host.

Whether you export directly from your app or from a collector, you'll need to:

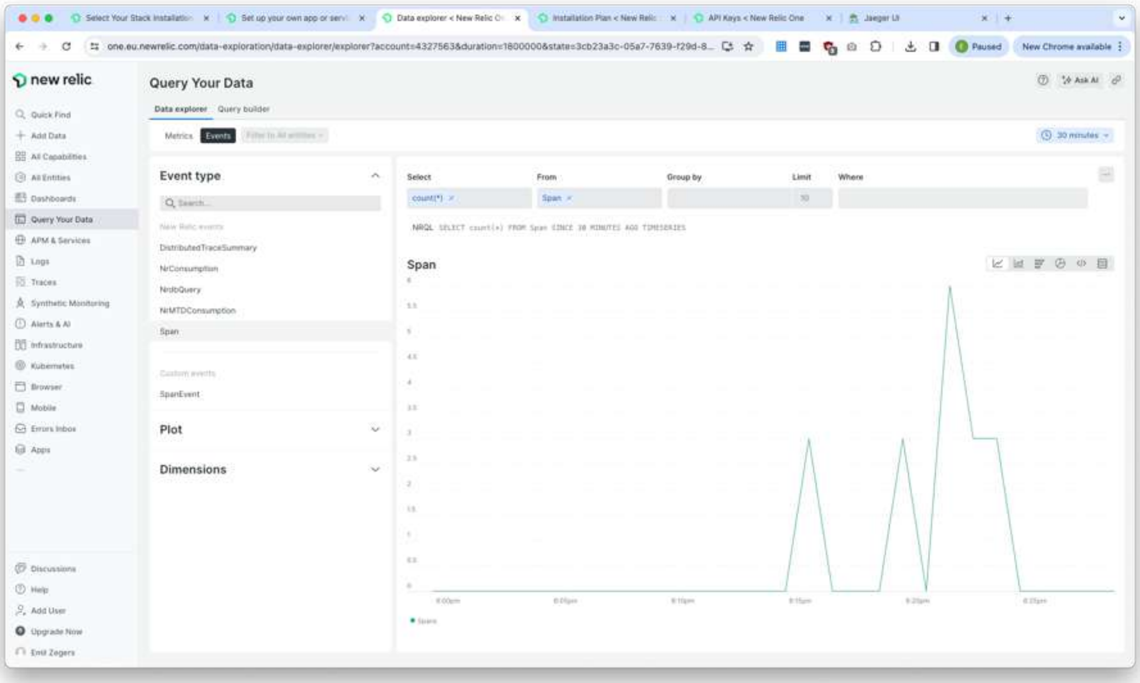
See YAML example at https://docs.newrelic.com/docs/more-integrations/open-source-telemetry-integrations/opentelemetry/collector/opentelemetry-collector-basic/


Query Your Data

```
SELECT count(*) FROM DistributedTraceSummary SINCE 30 MINUTES AGO TIMESERIES
```

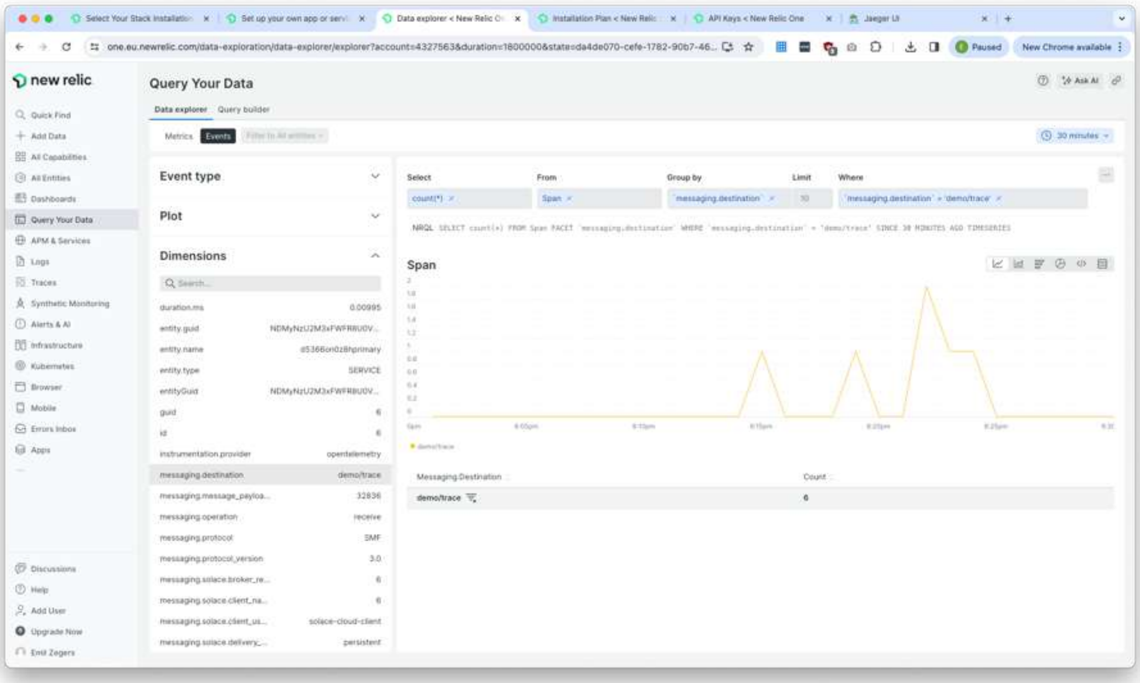
Chart type: Line

In **Query Your Data** select **Event Type Span**


Query Your Data

Select: count(*) From: Span Limit: 10

```
NRQL SELECT count(*) FROM Span SINCE 30 MINUTES AGO TIMESERIES
```


Select: count(*) | From: Span | Group by: messaging.destination | Limit: 10 | Where: messaging.destination = 'demo/trace'

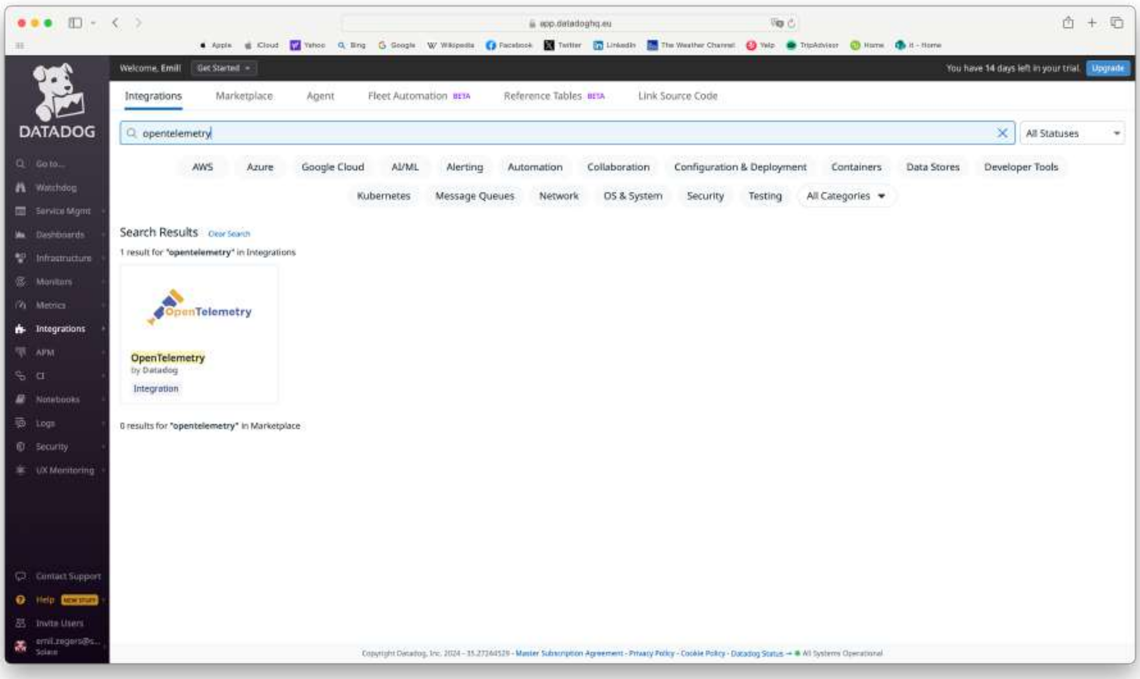
```
NRQL SELECT count(*) FROM Span FACET `messaging.destination` WHERE `messaging.destination` = 'demo/trace' SINCE 30 MINUTES AGO TIMESERIES
```

duration.ms	0.00995
entity.guid	NDMyNzU2M3xFWFR8U0V...
entity.name	d5366on0z8hprimary
entity.type	SERVICE
entityGuid	NDMyNzU2M3xFWFR8U0V...
guid	6
id	6
instrumentation.provider	opentelemetry
messaging.destination	demo/trace
messaging.message_payloa...	32836
messaging.operation	receive
messaging.protocol	SMF
messaging.protocol_version	3.0
messaging.solace.broker_re...	6
messaging.solace.client_na...	6
messaging.solace.client_us...	solace-cloud-client
messaging.solace.delivery_...	persistent

Messaging Destination	Count
demo/trace	6

4.5 DataDog

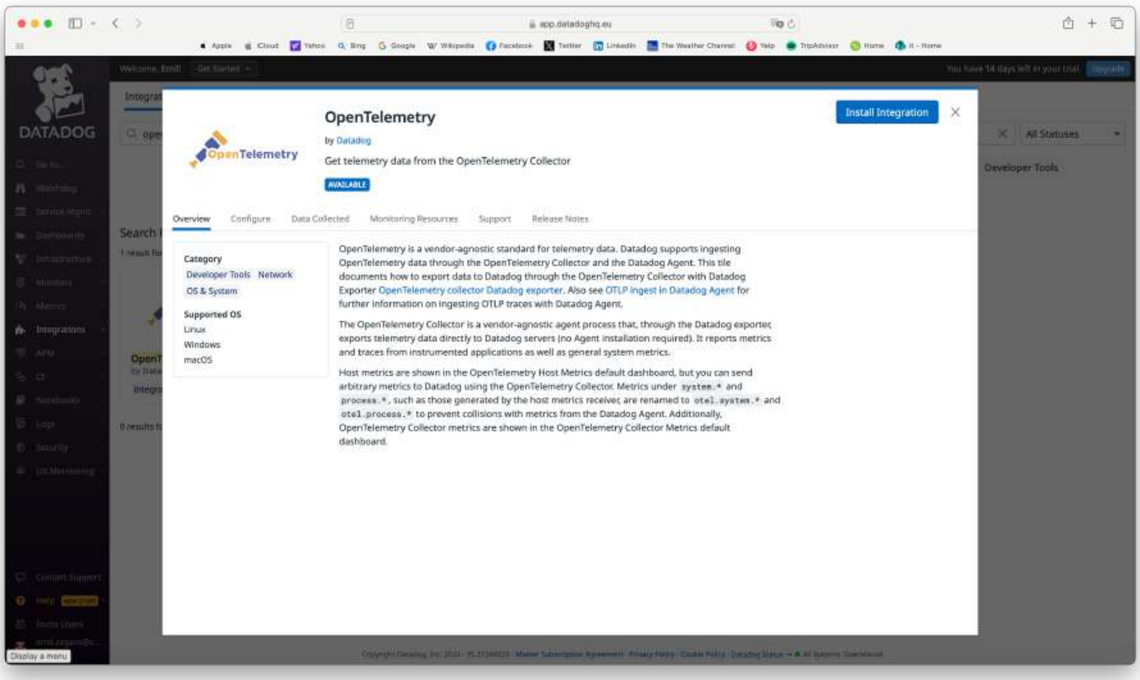
NOTE: no time yet to add… Keep an eye on updates.


Search Results

1 result for "opentelemetry" in Integrations

OpenTelemetry by Datadog — Integration

0 results for "opentelemetry" in Marketplace



OpenTelemetry

by Datadog

Get telemetry data from the OpenTelemetry Collector

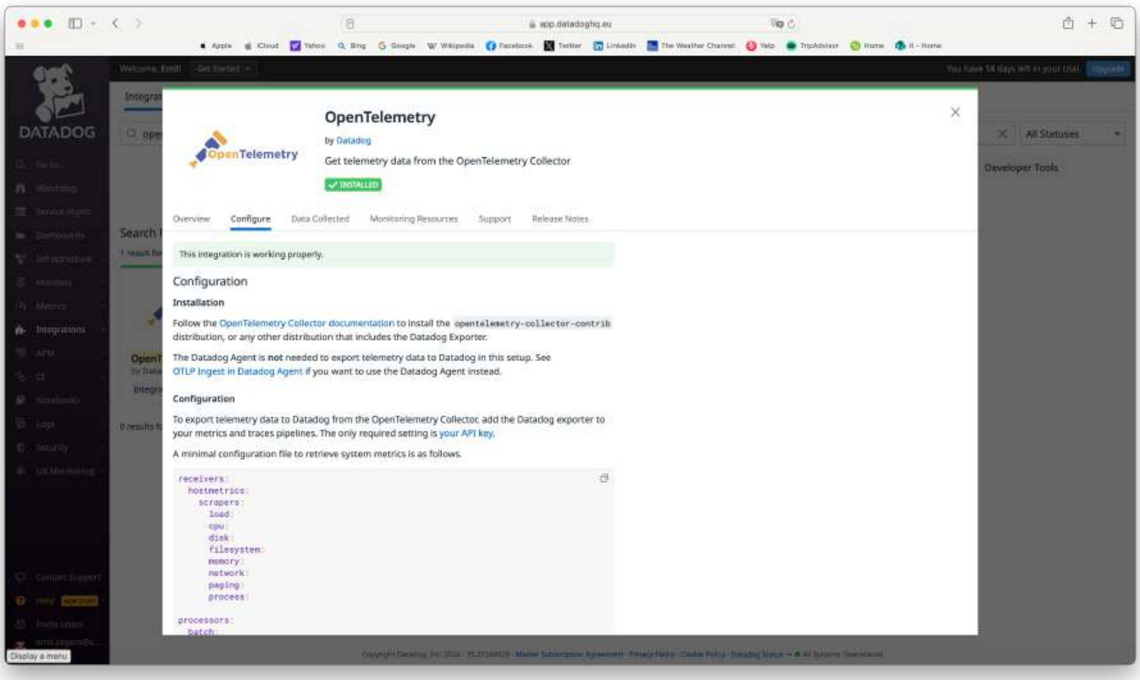
Category
Developer Tools Network
OS & System

Supported OS
Linux
Windows
macOS

OpenTelemetry is a vendor-agnostic standard for telemetry data. Datadog supports ingesting OpenTelemetry data through the OpenTelemetry Collector and the Datadog Agent. This tile documents how to export data to Datadog through the OpenTelemetry Collector with Datadog Exporter OpenTelemetry collector Datadog exporter. Also see OTLP ingest in Datadog Agent for further information on ingesting OTLP traces with Datadog Agent.

The OpenTelemetry Collector is a vendor-agnostic agent process that, through the Datadog exporter, exports telemetry data directly to Datadog servers (no Agent installation required). It reports metrics and traces from instrumented applications as well as general system metrics.

Host metrics are shown in the OpenTelemetry Host Metrics default dashboard, but you can send arbitrary metrics to Datadog using the OpenTelemetry Collector. Metrics under `system.*` and `process.*`, such as those generated by the host metrics receiver, are renamed to `otel.system.*` and `otel.process.*` to prevent collisions with metrics from the Datadog Agent. Additionally, OpenTelemetry Collector metrics are shown in the OpenTelemetry Collector Metrics default dashboard.



OpenTelemetry

by Datadog

Get telemetry data from the OpenTelemetry Collector

This integration is working properly.

Configuration

Installation

Follow the OpenTelemetry Collector documentation to install the `opentelemetry-collector-contrib` distribution, or any other distribution that includes the Datadog Exporter.

The Datadog Agent is **not** needed to export telemetry data to Datadog in this setup. See OTLP Ingest in Datadog Agent if you want to use the Datadog Agent instead.

Configuration

To export telemetry data to Datadog from the OpenTelemetry Collector, add the Datadog exporter to your metrics and traces pipelines. The only required setting is your API key.

A minimal configuration file to retrieve system metrics is as follows.

```
receivers:
  hostmetrics:
    scrapers:
      load:
      cpu:
      disk:
      filesystem:
      memory:
      network:
      paging:
      process:

processors:
  batch:
```

```
exporters:
  datadog:
    api:
      key: "<Your API key goes here>"

service:
  pipelines:
    metrics:
      receivers: [hostmetrics]
      processors: [batch]
      exporters: [datadog]
```

4.6 Splunk

NOTE: no time yet to add… Keep an eye on updates.

TODO: need for enterprise cloud package with Splunk?

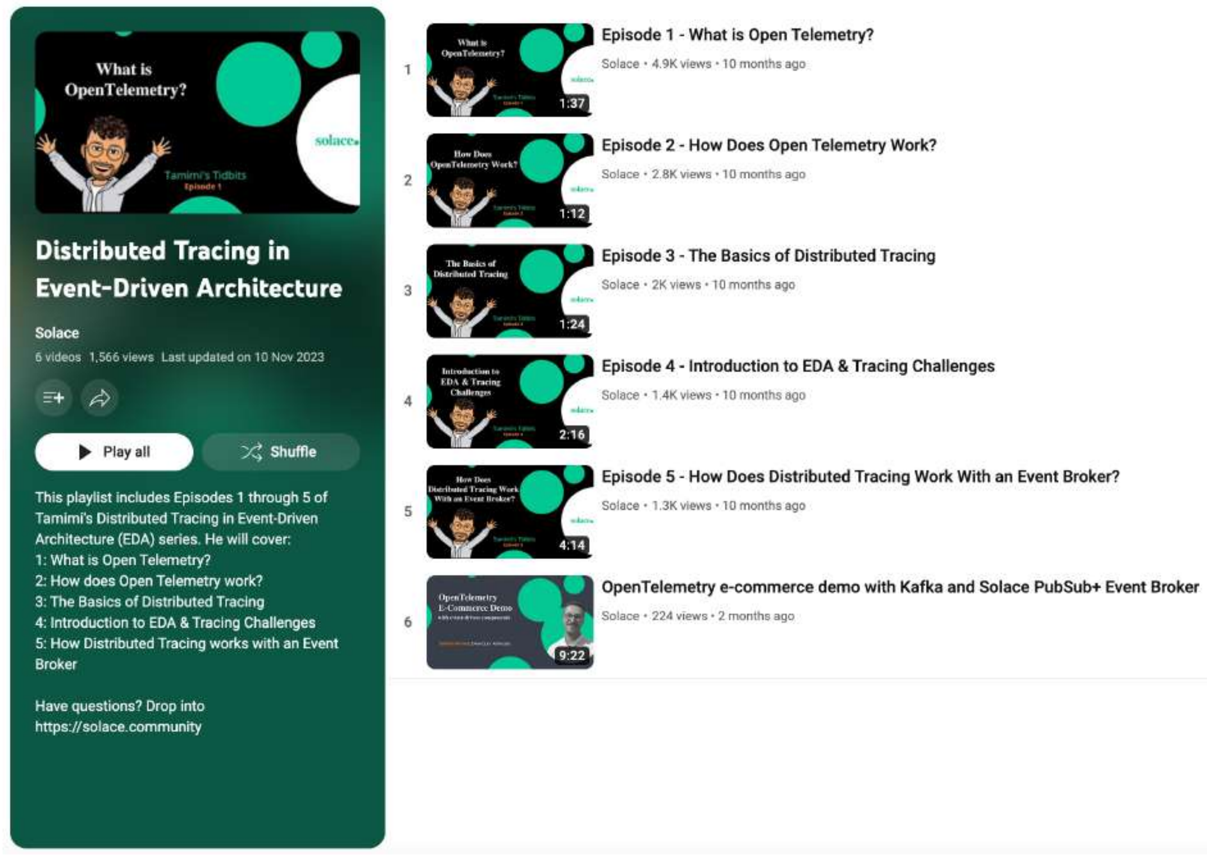
4.7 OpenSearch

https://opensearch.org/blog/opentelemetry-metrics-visualization/

5 Resources

https://github.com/SolaceLabs/solace-demo-observability
The repo also contains this document and the deck on Distributed Tracing used at the Solace Connect 2024 user group in Amsterdam https://solace.com/event/solace-connect-user-group-benelux-2024/

https://www.youtube.com/playlist?list=PLY1Ks8JEfJR7jWm3aafht9cou2oleB_Ef


Distributed Tracing in Event-Driven Architecture

Solace
6 videos 1,566 views Last updated on 10 Nov 2023

This playlist includes Episodes 1 through 5 of Tamimi's Distributed Tracing in Event-Driven Architecture (EDA) series. He will cover:
1: What is Open Telemetry?
2: How does Open Telemetry work?
3: The Basics of Distributed Tracing
4: Introduction to EDA & Tracing Challenges
5: How Distributed Tracing works with an Event Broker

Have questions? Drop into https://solace.community

1 Episode 1 - What is Open Telemetry? — Solace • 4.9K views • 10 months ago — 1:37
2 Episode 2 - How Does Open Telemetry Work? — Solace • 2.8K views • 10 months ago — 1:12
3 Episode 3 - The Basics of Distributed Tracing — Solace • 2K views • 10 months ago — 1:24
4 Episode 4 - Introduction to EDA & Tracing Challenges — Solace • 1.4K views • 10 months ago — 2:16
5 Episode 5 - How Does Distributed Tracing Work With an Event Broker? — Solace • 1.3K views • 10 months ago — 4:14
6 OpenTelemetry e-commerce demo with Kafka and Solace PubSub+ Event Broker — Solace • 224 views • 2 months ago — 9:22

OpenTelemetry e-commerce demo with Kafka and Solace PubSub+ Event Broker
https://www.youtube.com/watch?v=RIHQGVS5KNM