Powering Real-time Data at Intuit: A Look at Golden Signals powered by Beam

By Dunja Panic, Nick Hwang,
Omkar Deshpande, & Nagaraja Tantry

BEAM SUMMIT

Agenda

Intuit's Stream Processing Platform

Developer Experience

Platform Architecture

Featured Use Case

Q&A

Who we serve

Consumers

Small businesses

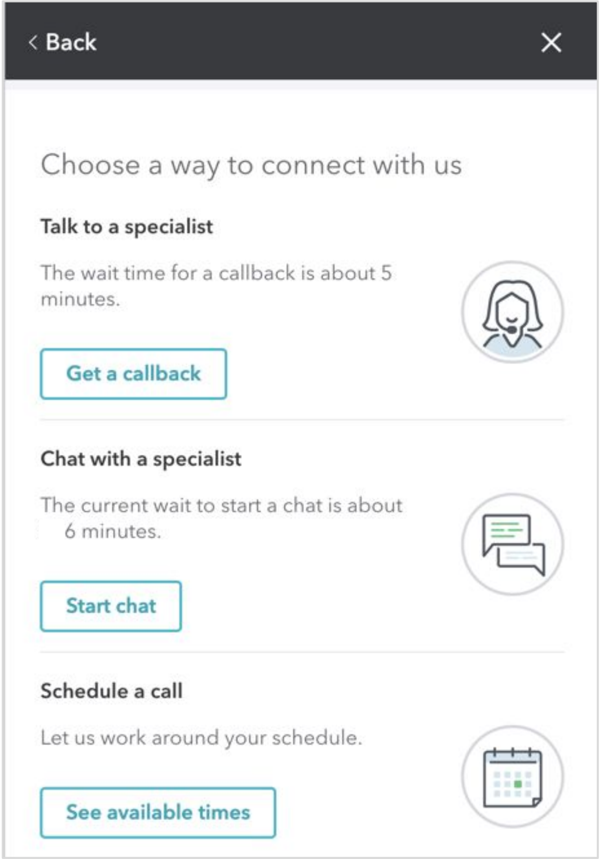
Self-employed

intuit

turbotax quickbooks mint

credit karma mailchimp

Powering Prosperity with AI and Data-driven platforms



Choose a way to connect with us

Talk to a specialist

The wait time for a callback is about 5 minutes.

Chat with a specialist

The current wait to start a chat is about 6 minutes.

Schedule a call

Let us work around your schedule.

Intuit customers want to get their issues resolved in the most efficient way possible to feel confident in their outcomes. We want to Intelligently route users to the right expert to quickly resolve their issue

Origins of Stream Processing Platform

Before

- Adhoc workloads with low data freshness
- High infrastructure costs across teams
- Custom integrations handled on an adhoc basis, team by team
- Team focused on operating streaming infrastructure

O'REILLY
Kafka
The Definitive Guide
REAL-TIME DATA AND STREAM PROCESSING AT SCALE
Gwen Shapira & Todd
O'REILLY
Stream Processing with Apache Flink
Fundamentals, Implementation, and Operation of Streaming Applications
O'REILLY
Kubernetes
Up & Running
Dive into the Future of Infrastructure
Second Edition
Joe Beda & Kelsey Hightower
Brendan Burns,

After

- High data availability and data freshness
- Cost savings due to shared infrastructure
- Standardized integrations with Intuit developer ecosystem
- Teams focus on producing/consuming clickstream or application events

3x	Improvement in Speed to Market
5x	Reduction in cost
240x	Improvement in data availability

Data Engineer

"I want to focus on rapidly developing streaming applications so that I can provide real-time personalized user experiences in my product"

Data Exploration

Deployment

Monitoring

Access & Governance

Data Security, Compliance

Metadata & Lineage

Asset Lifecycle

Cost & Billing

Operationalization

Cloud resources

Eventing

Scalability

Containers

Data Lake

Reliability

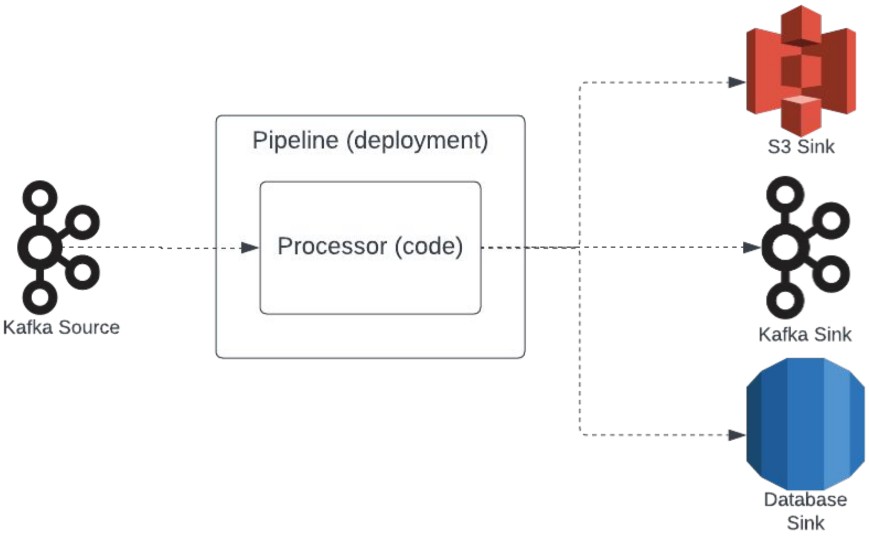
Infrastructure Management

Key Features

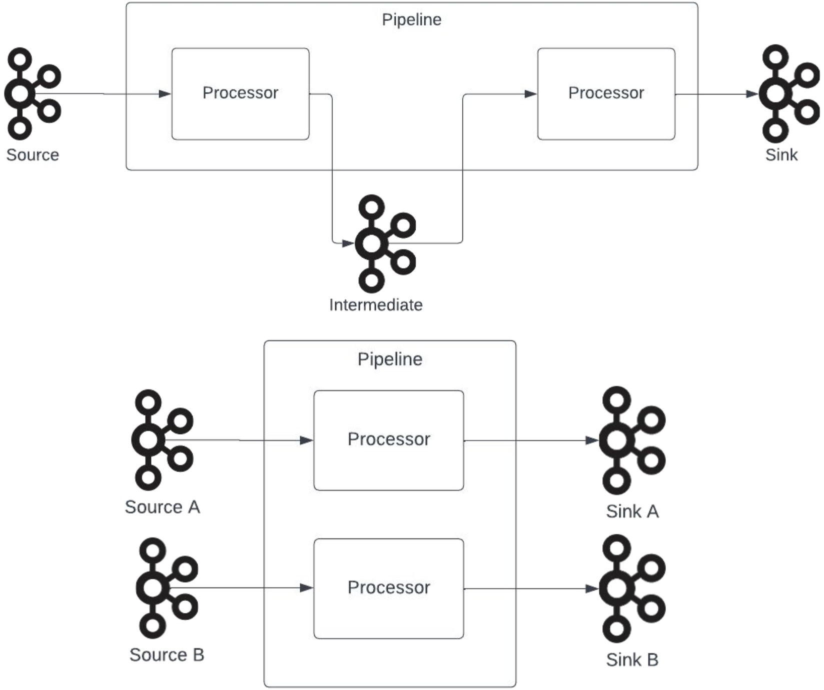
- **Push-button** pipeline management
- Completely **managed** infrastructure
- **Out-of-the-box** starter code and dashboards
- Programming language and processing execution engine **flexibility**
- Rich **discoverability** and exploration of Intuit data ecosystem

Developer Experience on SPP

Processors and Pipelines



- Processor = Business Logic & Code
- Pipeline = Deployment & Infrastructure



- Serial processors (e.g., reusable intermediate topic)
- Parallel processors (e.g., fleet deployment)

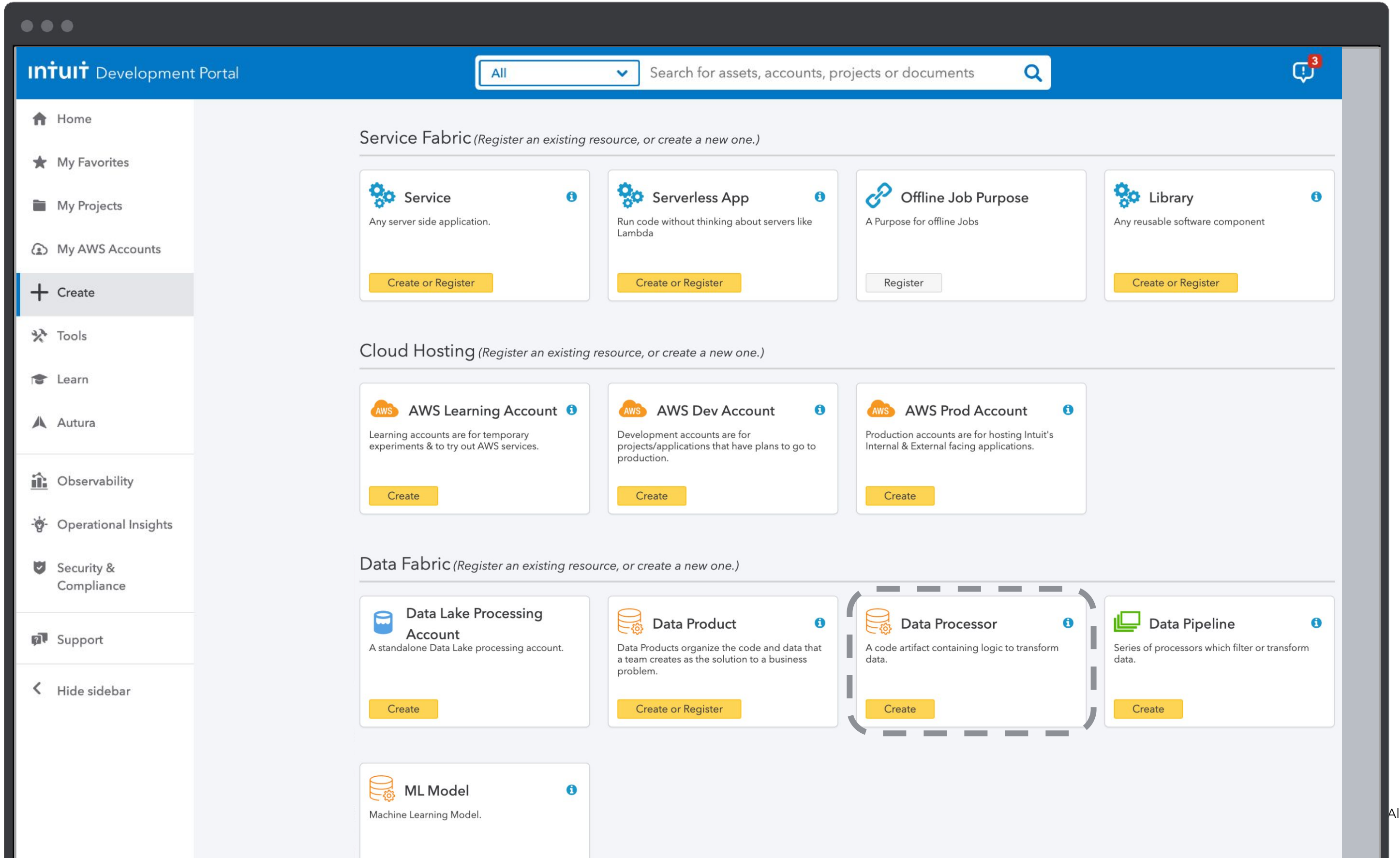
Author

Compose

Deploy

Author → Compose → Deploy

DevPortal: Intuit's home for "self-serve paved roads"



Service Fabric *(Register an existing resource, or create a new one.)*

Service
Any server side application.
Create or Register

Serverless App
Run code without thinking about servers like Lambda
Create or Register

Offline Job Purpose
A Purpose for offline Jobs
Register

Library
Any reusable software component
Create or Register

Cloud Hosting *(Register an existing resource, or create a new one.)*

AWS Learning Account
Learning accounts are for temporary experiments & to try out AWS services.
Create

AWS Dev Account
Development accounts are for projects/applications that have plans to go to production.
Create

AWS Prod Account
Production accounts are for hosting Intuit's Internal & External facing applications.
Create

Data Fabric *(Register an existing resource, or create a new one.)*

Data Lake Processing Account
A standalone Data Lake processing account.
Create

Data Product
Data Products organize the code and data that a team creates as the solution to a business problem.
Create or Register

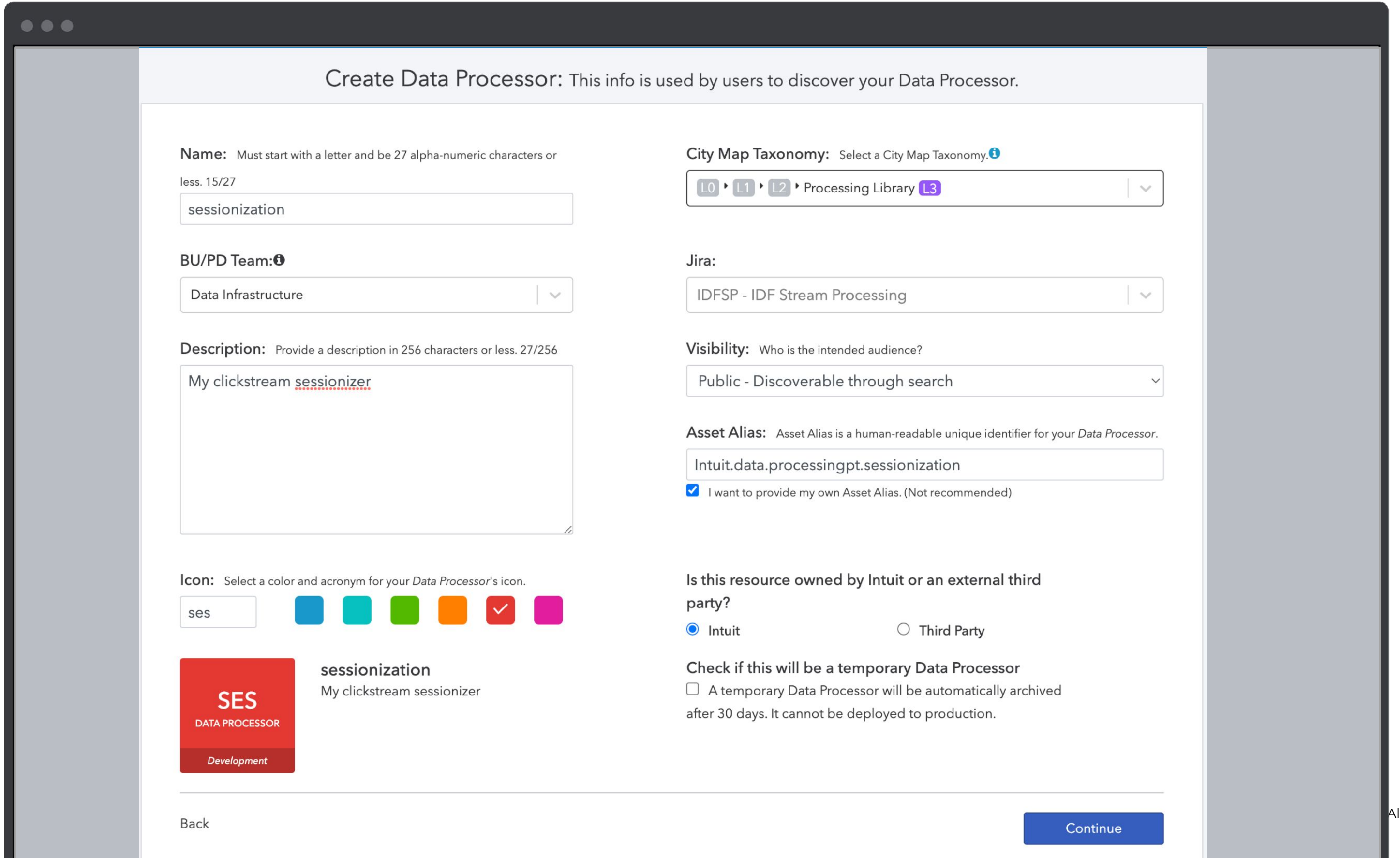
Data Processor
A code artifact containing logic to transform data.
Create

Data Pipeline
Series of processors which filter or transform data.
Create

ML Model
Machine Learning Model.

Author → Compose → Deploy

With a couple clicks…



Create Data Processor: This info is used by users to discover your Data Processor.

Name: Must start with a letter and be 27 alpha-numeric characters or less. 15/27

sessionization

BU/PD Team:

Data Infrastructure

Description: Provide a description in 256 characters or less. 27/256

My clickstream sessionizer

Icon: Select a color and acronym for your *Data Processor*'s icon.

ses

SES
DATA PROCESSOR
Development

sessionization
My clickstream sessionizer

City Map Taxonomy: Select a City Map Taxonomy.

L0 ▸ L1 ▸ L2 ▸ Processing Library L3

Jira:

IDFSP - IDF Stream Processing

Visibility: Who is the intended audience?

Public - Discoverable through search

Asset Alias: Asset Alias is a human-readable unique identifier for your *Data Processor*.

Intuit.data.processingpt.sessionization

☑ I want to provide my own Asset Alias. (Not recommended)

Is this resource owned by Intuit or an external third party?

● Intuit ○ Third Party

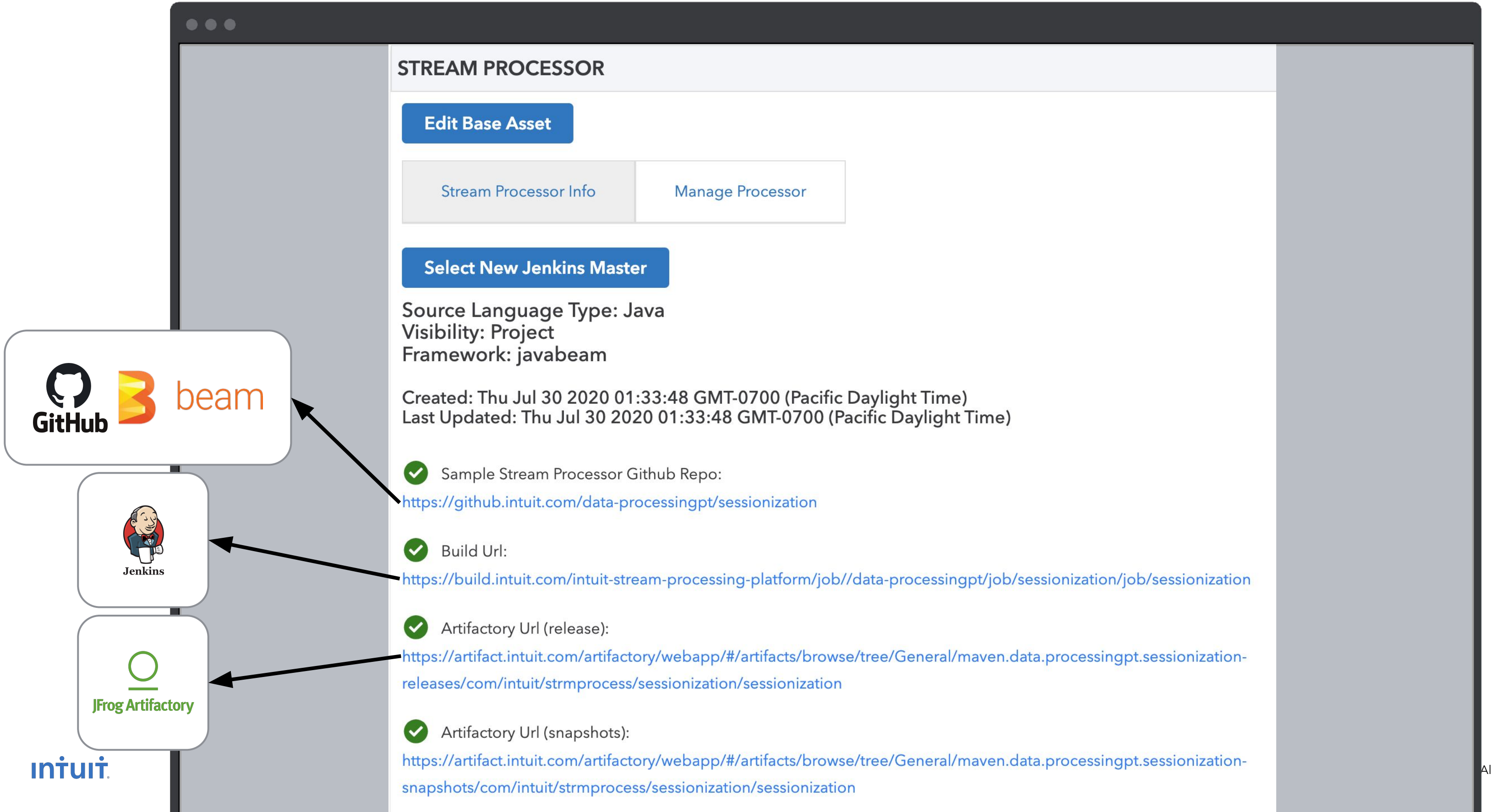
Check if this will be a temporary Data Processor

□ A temporary Data Processor will be automatically archived after 30 days. It cannot be deployed to production.

Back

Continue

Everything you need to start coding



STREAM PROCESSOR

Edit Base Asset

Stream Processor Info | Manage Processor

Select New Jenkins Master

Source Language Type: Java
Visibility: Project
Framework: javabeam

Created: Thu Jul 30 2020 01:33:48 GMT-0700 (Pacific Daylight Time)
Last Updated: Thu Jul 30 2020 01:33:48 GMT-0700 (Pacific Daylight Time)

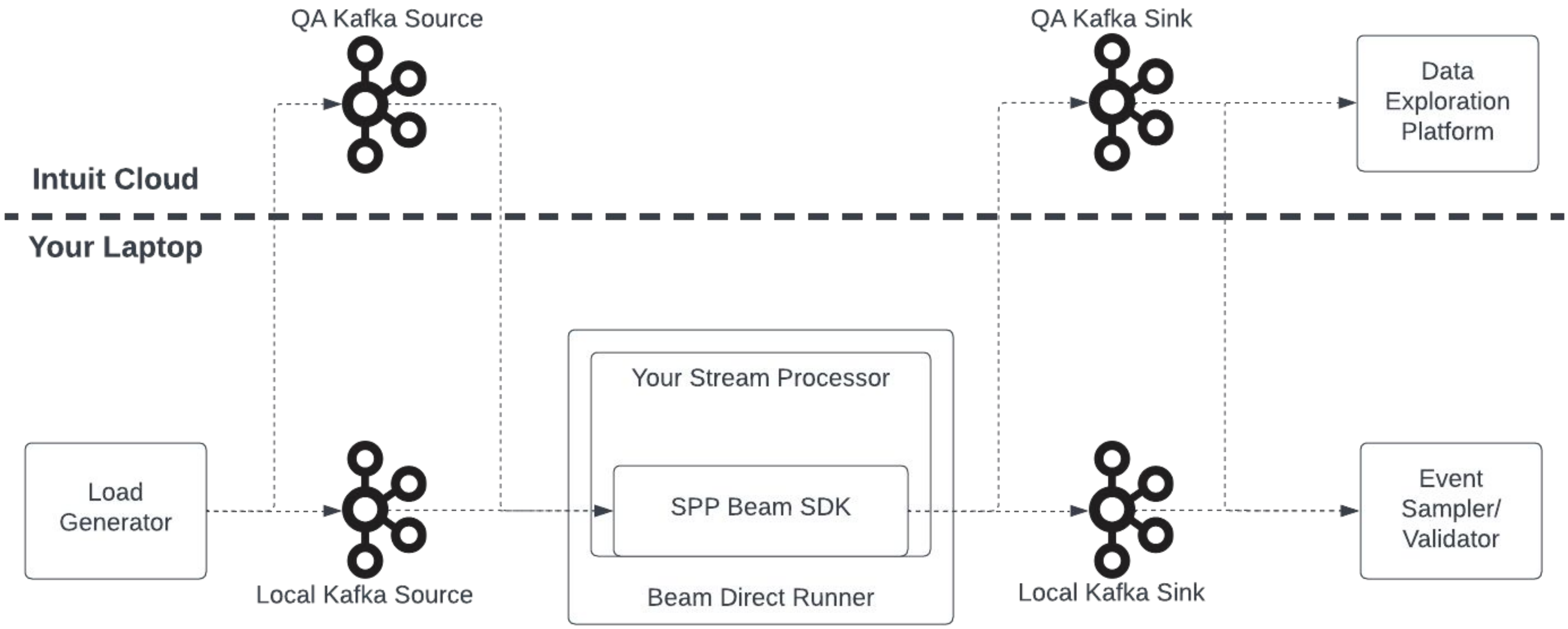
✓ Sample Stream Processor Github Repo:
https://github.intuit.com/data-processingpt/sessionization

✓ Build Url:
https://build.intuit.com/intuit-stream-processing-platform/job//data-processingpt/job/sessionization/job/sessionization

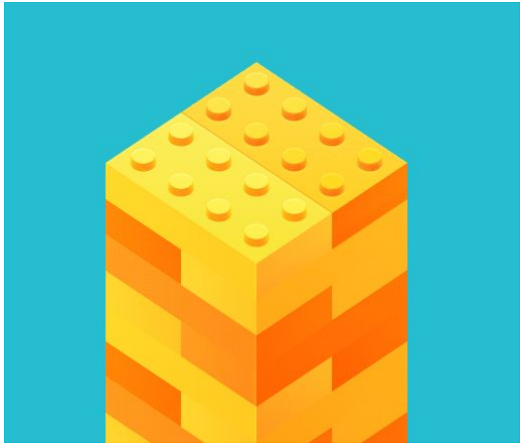
✓ Artifactory Url (release):
https://artifact.intuit.com/artifactory/webapp/#/artifacts/browse/tree/General/maven.data.processingpt.sessionization-releases/com/intuit/strmprocess/sessionization/sessionization

✓ Artifactory Url (snapshots):
https://artifact.intuit.com/artifactory/webapp/#/artifacts/browse/tree/General/maven.data.processingpt.sessionization-snapshots/com/intuit/strmprocess/sessionization/sessionization

Locally iterating until cloud-ready



Author

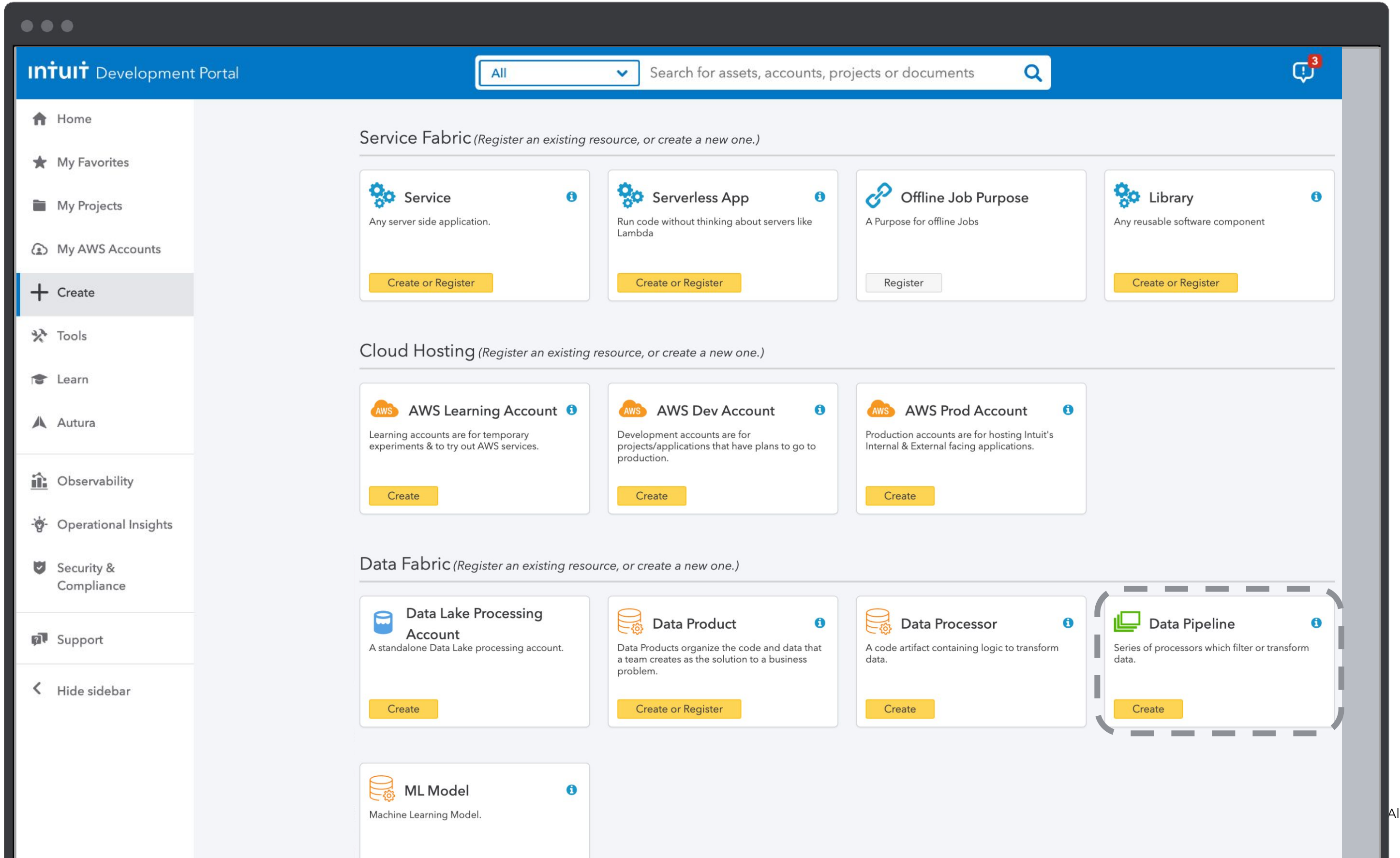


Compose

Deploy

Author → **Compose** → Deploy

Processor = Code
Pipeline = Cloud Deployment



intuit Development Portal

All ⌄ | Search for assets, accounts, projects or documents

- Home
- My Favorites
- My Projects
- My AWS Accounts
- Create
- Tools
- Learn
- Autura
- Observability
- Operational Insights
- Security & Compliance
- Support
- Hide sidebar

Service Fabric *(Register an existing resource, or create a new one.)*

Service
Any server side application.
Create or Register

Serverless App
Run code without thinking about servers like Lambda
Create or Register

Offline Job Purpose
A Purpose for offline Jobs
Register

Library
Any reusable software component
Create or Register

Cloud Hosting *(Register an existing resource, or create a new one.)*

AWS Learning Account
Learning accounts are for temporary experiments & to try out AWS services.
Create

AWS Dev Account
Development accounts are for projects/applications that have plans to go to production.
Create

AWS Prod Account
Production accounts are for hosting Intuit's Internal & External facing applications.
Create

Data Fabric *(Register an existing resource, or create a new one.)*

Data Lake Processing Account
A standalone Data Lake processing account.
Create

Data Product
Data Products organize the code and data that a team creates as the solution to a business problem.
Create or Register

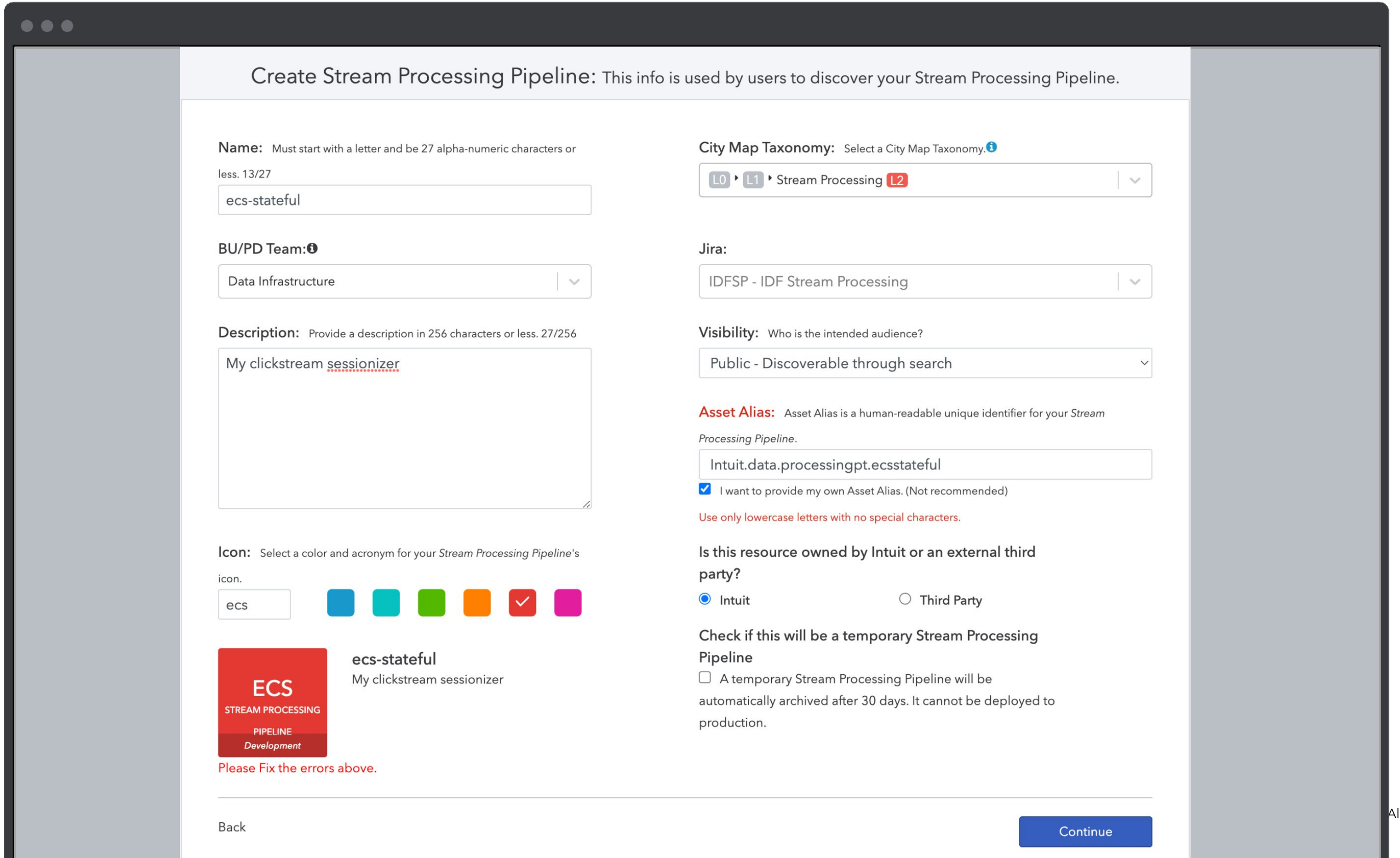
Data Processor
A code artifact containing logic to transform data.
Create

Data Pipeline
Series of processors which filter or transform data.
Create

ML Model
Machine Learning Model.

Author → **Compose** → Deploy

With a couple clicks…



Create Stream Processing Pipeline: This info is used by users to discover your Stream Processing Pipeline.

Name: Must start with a letter and be 27 alpha-numeric characters or less. 13/27

ecs-stateful

BU/PD Team:

Data Infrastructure

Description: Provide a description in 256 characters or less. 27/256

My clickstream sessionizer

Icon: Select a color and acronym for your *Stream Processing Pipeline*'s icon.

ecs

ECS
STREAM PROCESSING
PIPELINE
Development

ecs-stateful
My clickstream sessionizer

Please Fix the errors above.

City Map Taxonomy: Select a City Map Taxonomy.

L0 ‣ L1 ‣ Stream Processing L2

Jira:

IDFSP - IDF Stream Processing

Visibility: Who is the intended audience?

Public - Discoverable through search

Asset Alias: Asset Alias is a human-readable unique identifier for your *Stream Processing Pipeline*.

Intuit.data.processingpt.ecsstateful

☑ I want to provide my own Asset Alias. (Not recommended)

Use only lowercase letters with no special characters.

Is this resource owned by Intuit or an external third party?

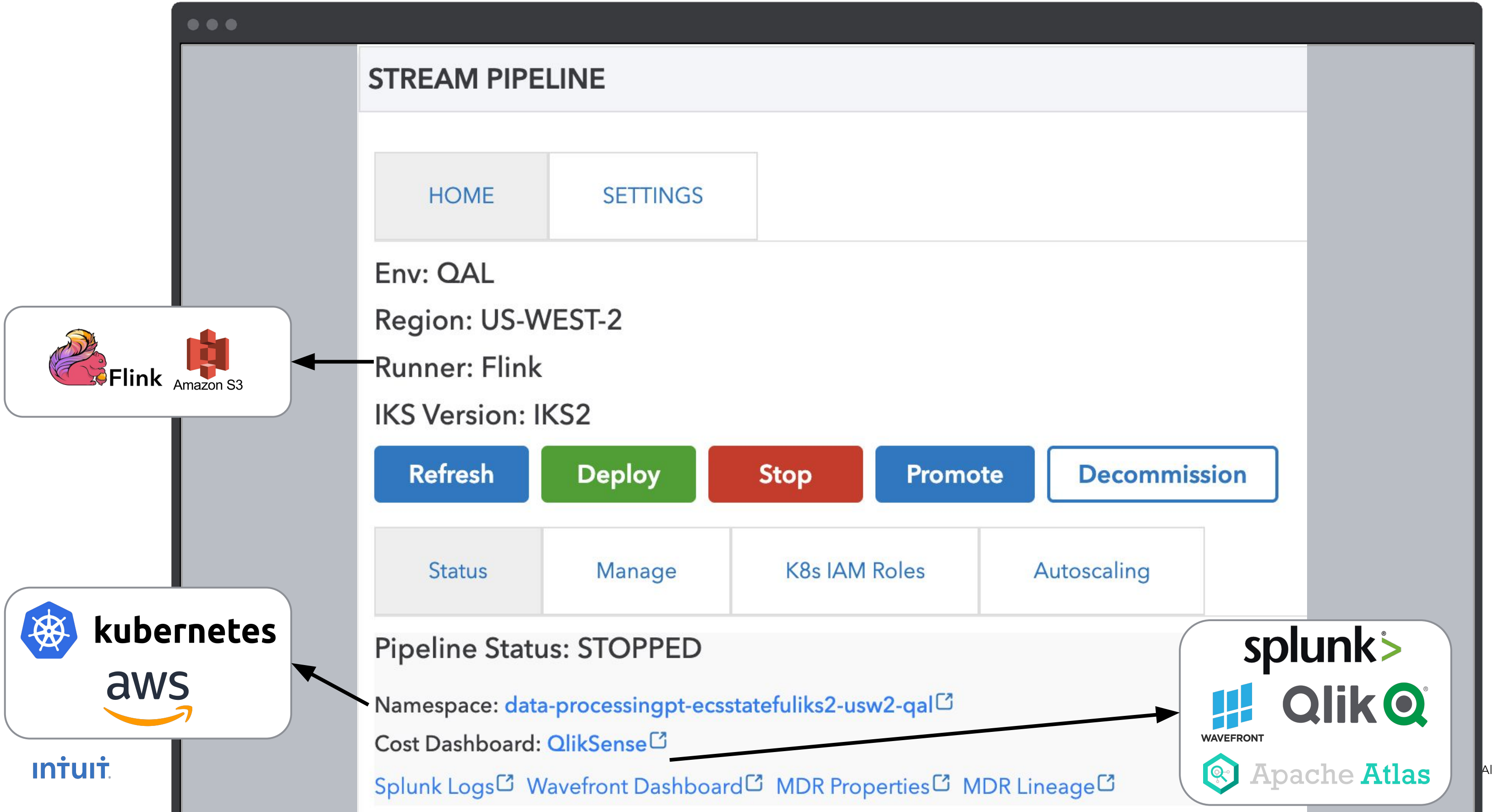
● Intuit ○ Third Party

Check if this will be a temporary Stream Processing Pipeline

□ A temporary Stream Processing Pipeline will be automatically archived after 30 days. It cannot be deployed to production.

Author → **Compose** → Deploy

All the infrastructure you need



STREAM PIPELINE

HOME | SETTINGS

Env: QAL

Region: US-WEST-2

Runner: Flink

IKS Version: IKS2

Refresh | Deploy | Stop | Promote | Decommission

Status | Manage | K8s IAM Roles | Autoscaling

Pipeline Status: STOPPED

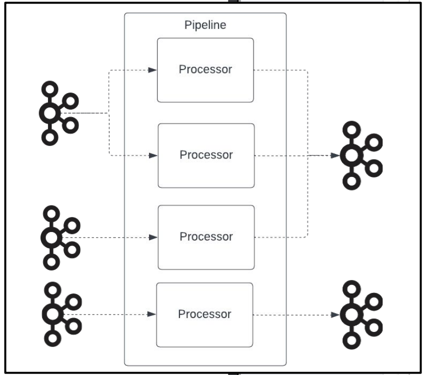
Namespace: data-processingpt-ecsstatefuliks2-usw2-qal

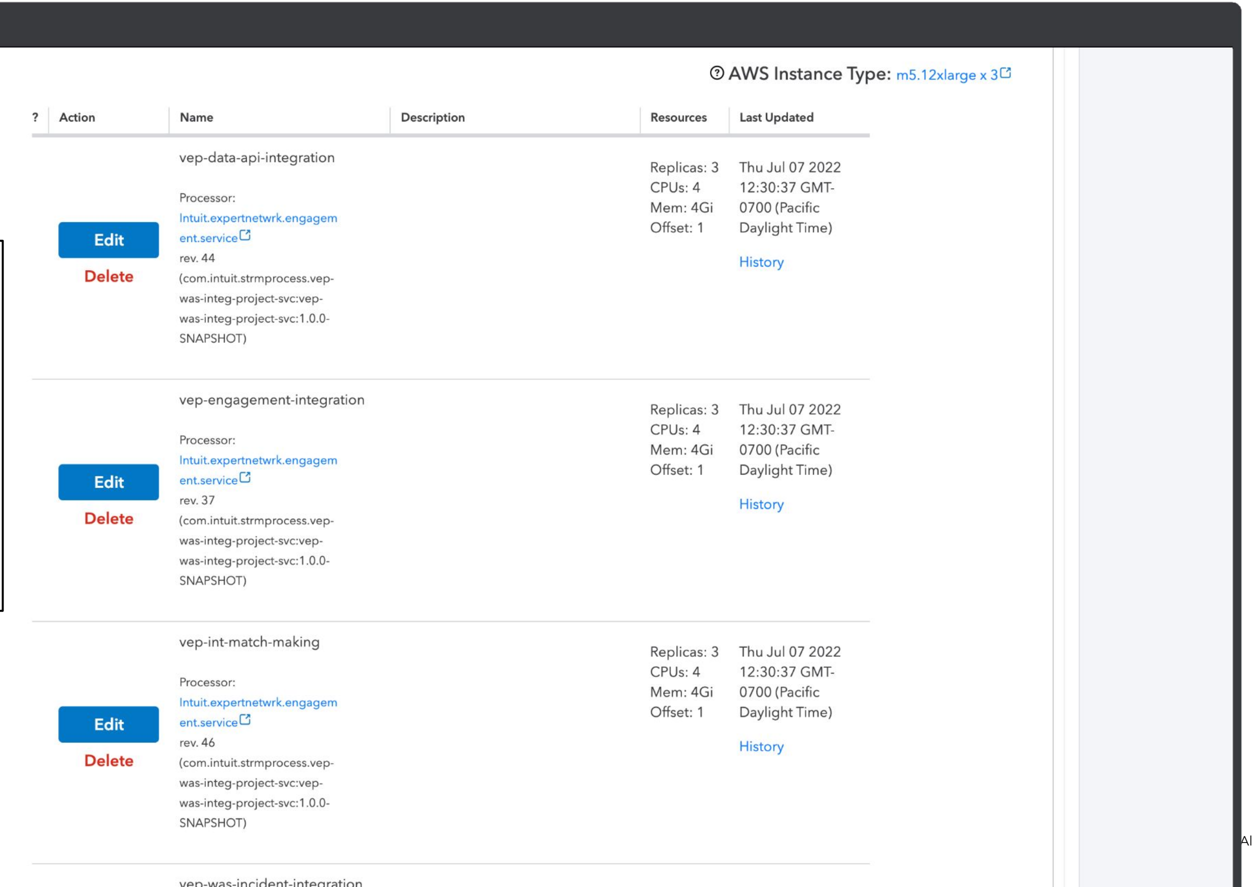
Cost Dashboard: QlikSense

Splunk Logs Wavefront Dashboard MDR Properties MDR Lineage

Author → **Compose** → Deploy

Compose your processors…

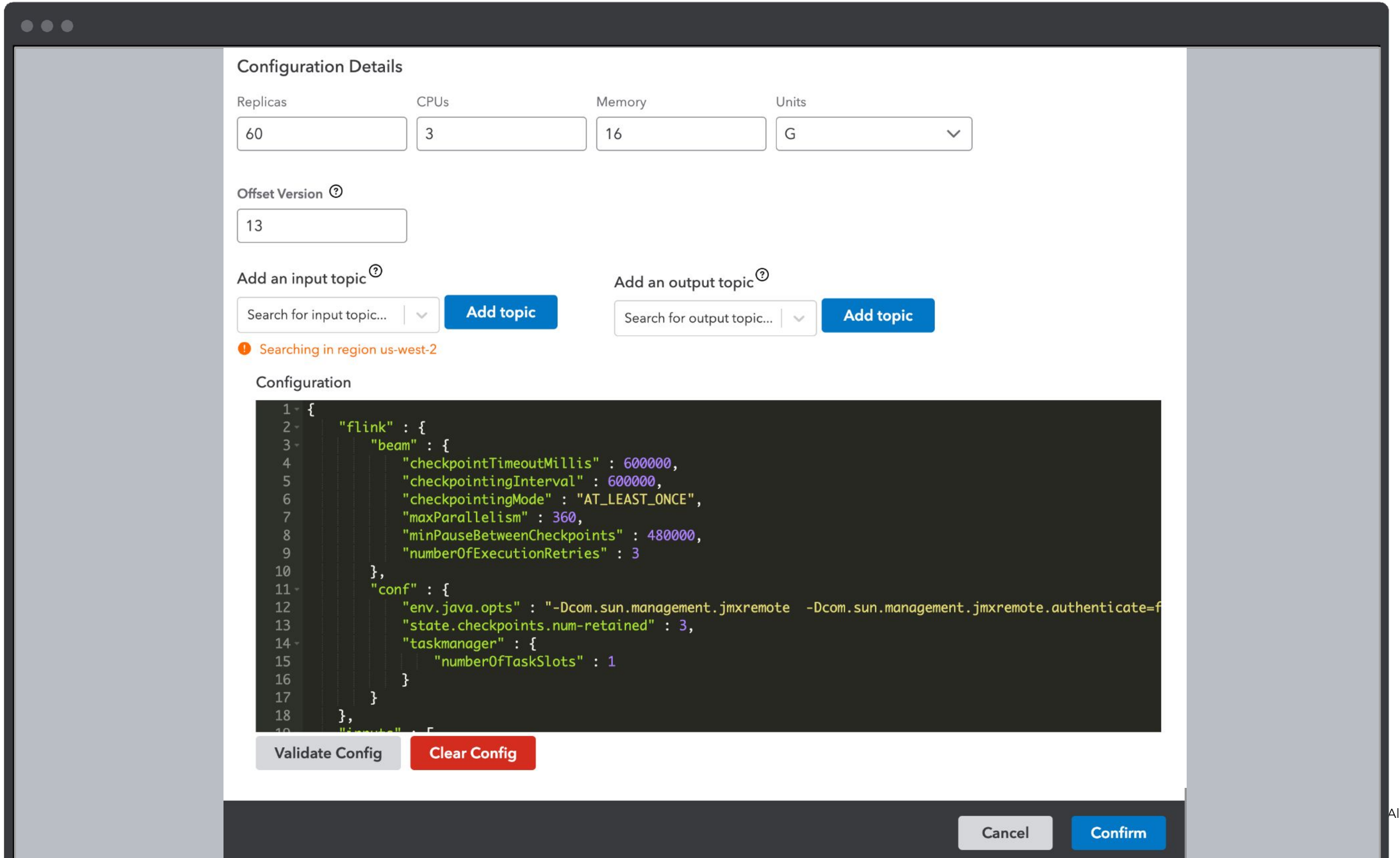




ⓘ AWS Instance Type: m5.12xlarge x 3

?	Action	Name	Description	Resources	Last Updated
	Edit Delete	vep-data-api-integration Processor: Intuit.expertnetwrk.engagement.service rev. 44 (com.intuit.strmprocess.vep-was-integ-project-svc:vep-was-integ-project-svc:1.0.0-SNAPSHOT)		Replicas: 3 CPUs: 4 Mem: 4Gi Offset: 1	Thu Jul 07 2022 12:30:37 GMT-0700 (Pacific Daylight Time) History
	Edit Delete	vep-engagement-integration Processor: Intuit.expertnetwrk.engagement.service rev. 37 (com.intuit.strmprocess.vep-was-integ-project-svc:vep-was-integ-project-svc:1.0.0-SNAPSHOT)		Replicas: 3 CPUs: 4 Mem: 4Gi Offset: 1	Thu Jul 07 2022 12:30:37 GMT-0700 (Pacific Daylight Time) History
	Edit Delete	vep-int-match-making Processor: Intuit.expertnetwrk.engagement.service rev. 46 (com.intuit.strmprocess.vep-was-integ-project-svc:vep-was-integ-project-svc:1.0.0-SNAPSHOT)		Replicas: 3 CPUs: 4 Mem: 4Gi Offset: 1	Thu Jul 07 2022 12:30:37 GMT-0700 (Pacific Daylight Time) History
		vep-was-incident-integration			

…fine-tune to your heart's content…



Configuration Details

Replicas: 60

CPUs: 3

Memory: 16

Units: G

Offset Version: 13

Add an input topic

Add an output topic

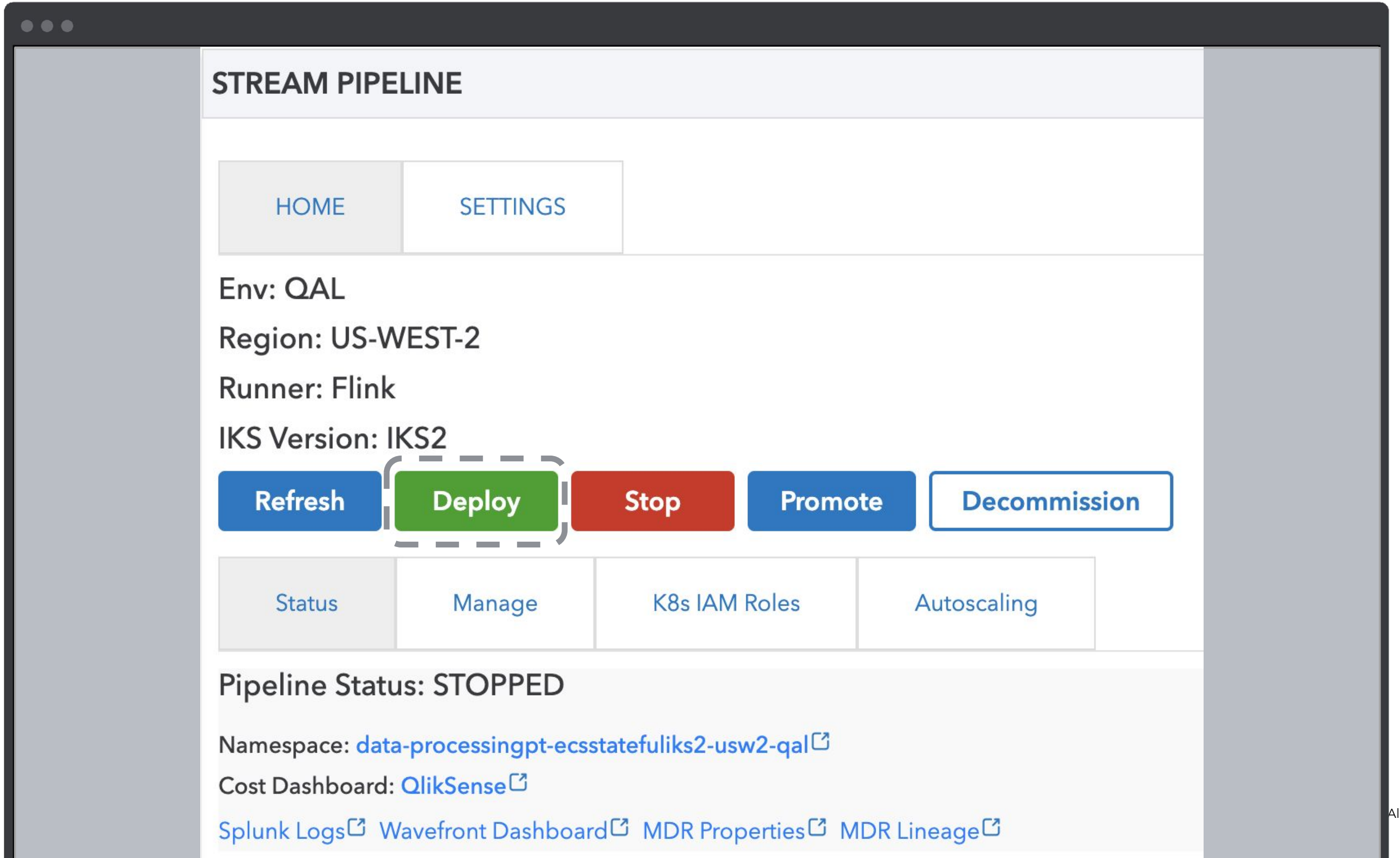
Searching in region us-west-2

Configuration

```
{
    "flink" : {
        "beam" : {
            "checkpointTimeoutMillis" : 600000,
            "checkpointingInterval" : 600000,
            "checkpointingMode" : "AT_LEAST_ONCE",
            "maxParallelism" : 360,
            "minPauseBetweenCheckpoints" : 480000,
            "numberOfExecutionRetries" : 3
        },
        "conf" : {
            "env.java.opts" : "-Dcom.sun.management.jmxremote  -Dcom.sun.management.jmxremote.authenticate=f
            "state.checkpoints.num-retained" : 3,
            "taskmanager" : {
                "numberOfTaskSlots" : 1
            }
        }
    },
    "inputs" : [
```

Author → **Compose** → Deploy

…and get ready to click the big green button!



STREAM PIPELINE

Env: QAL

Region: US-WEST-2

Runner: Flink

IKS Version: IKS2

Pipeline Status: STOPPED

Namespace: data-processingpt-ecsstatefuliks2-usw2-qal

Cost Dashboard: QlikSense

Splunk Logs Wavefront Dashboard MDR Properties MDR Lineage

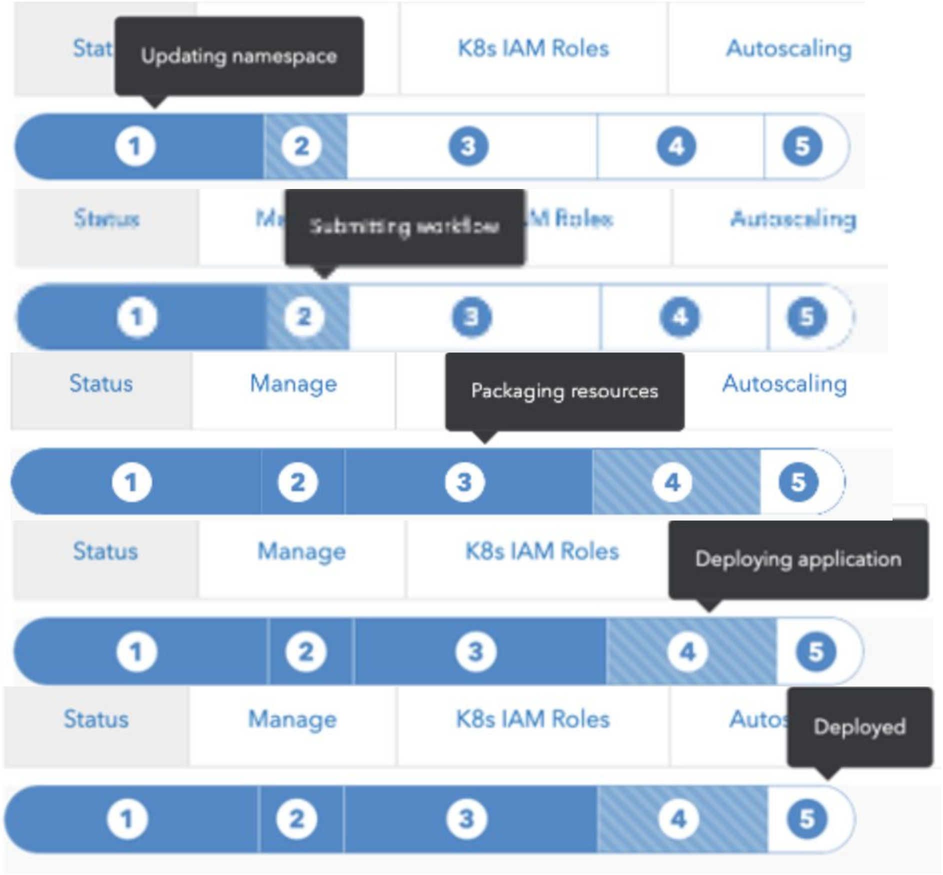
Author

Compose

Deploy

Author → Compose → Deploy

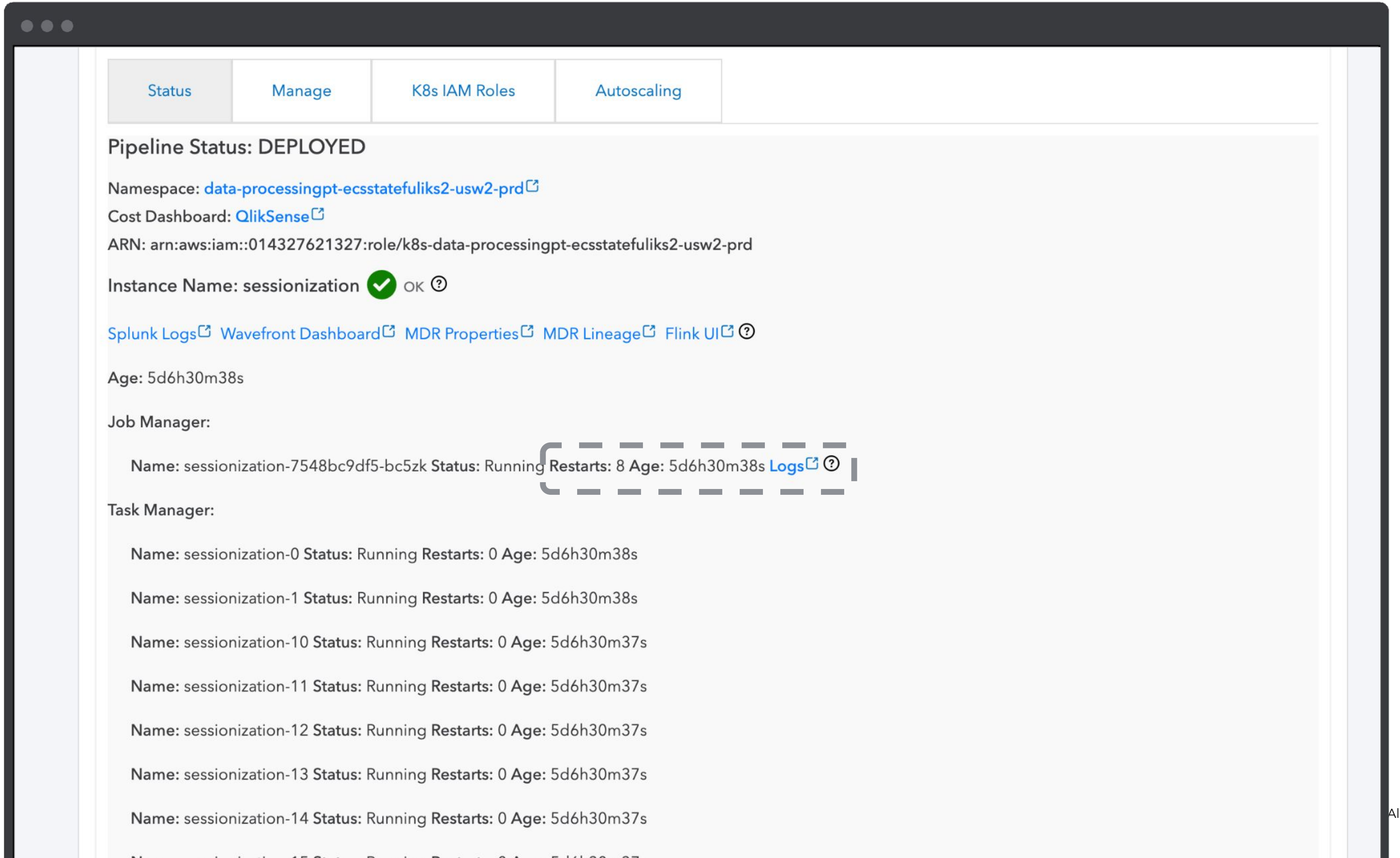
Watch your deploy complete…



intuit

Author → Compose → **Deploy**

…see your pipeline's status…



Status | Manage | K8s IAM Roles | Autoscaling

Pipeline Status: DEPLOYED

Namespace: data-processingpt-ecsstatefuliks2-usw2-prd
Cost Dashboard: QlikSense
ARN: arn:aws:iam::014327621327:role/k8s-data-processingpt-ecsstatefuliks2-usw2-prd

Instance Name: sessionization ✓ OK ?

Splunk Logs Wavefront Dashboard MDR Properties MDR Lineage Flink UI ?

Age: 5d6h30m38s

Job Manager:

Name: sessionization-7548bc9df5-bc5zk **Status:** Running **Restarts:** 8 **Age:** 5d6h30m38s Logs ?

Task Manager:

Name: sessionization-0 **Status:** Running **Restarts:** 0 **Age:** 5d6h30m38s

Name: sessionization-1 **Status:** Running **Restarts:** 0 **Age:** 5d6h30m38s

Name: sessionization-10 **Status:** Running **Restarts:** 0 **Age:** 5d6h30m37s

Name: sessionization-11 **Status:** Running **Restarts:** 0 **Age:** 5d6h30m37s

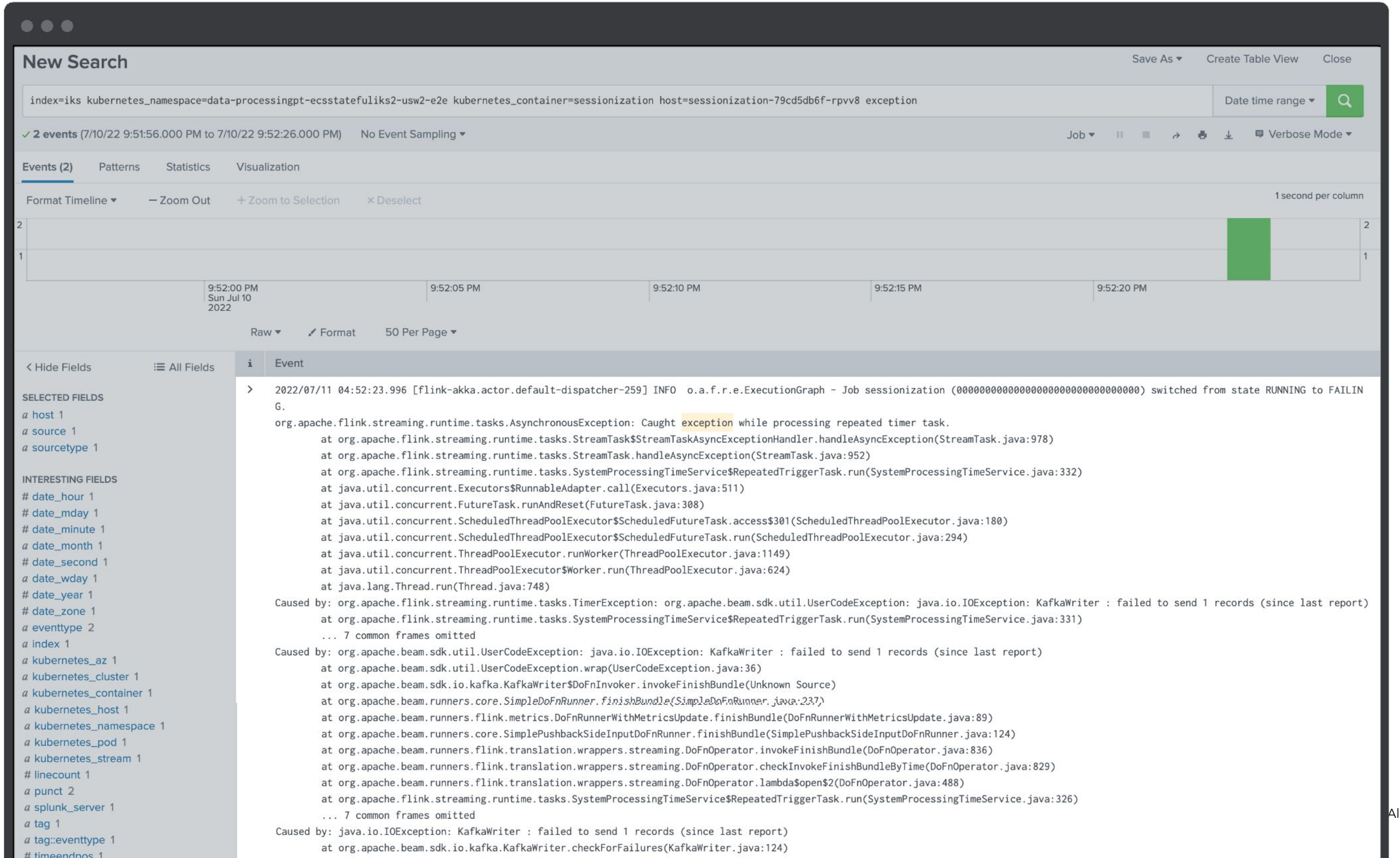
Name: sessionization-12 **Status:** Running **Restarts:** 0 **Age:** 5d6h30m37s

Name: sessionization-13 **Status:** Running **Restarts:** 0 **Age:** 5d6h30m37s

Name: sessionization-14 **Status:** Running **Restarts:** 0 **Age:** 5d6h30m37s

Author → Compose → **Deploy**

…look for exceptions in a haystack…



```
index=iks kubernetes_namespace=data-processingpt-ecsstatefuliks2-usw2-e2e kubernetes_container=sessionization host=sessionization-79cd5db6f-rpvv8 exception
```

✓ 2 events (7/10/22 9:51:56.000 PM to 7/10/22 9:52:26.000 PM)

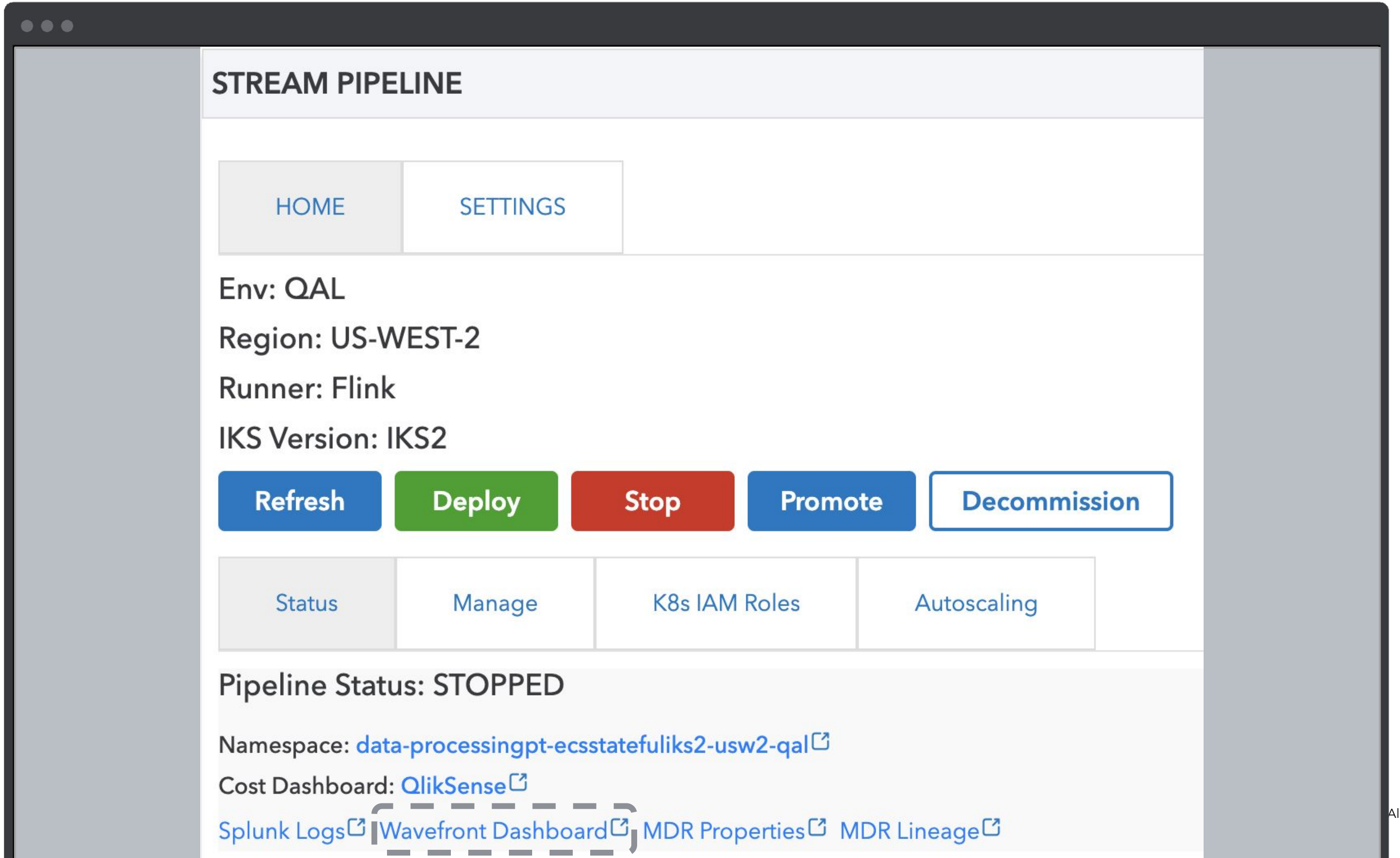
SELECTED FIELDS
a host 1
a source 1
a sourcetype 1

INTERESTING FIELDS
# date_hour 1
# date_mday 1
# date_minute 1
a date_month 1
# date_second 1
a date_wday 1
# date_year 1
# date_zone 1
a eventtype 2
a index 1
a kubernetes_az 1
a kubernetes_cluster 1
a kubernetes_container 1
a kubernetes_host 1
a kubernetes_namespace 1
a kubernetes_pod 1
a kubernetes_stream 1
# linecount 1
a punct 2
a splunk_server 1
a tag 1
a tag::eventtype 1
# timeendpos 1

```
2022/07/11 04:52:23.996 [flink-akka.actor.default-dispatcher-259] INFO  o.a.f.r.e.ExecutionGraph - Job sessionization (00000000000000000000000000000000) switched from state RUNNING to FAILIN
G.
org.apache.flink.streaming.runtime.tasks.AsynchronousException: Caught exception while processing repeated timer task.
        at org.apache.flink.streaming.runtime.tasks.StreamTask$StreamTaskAsyncExceptionHandler.handleAsyncException(StreamTask.java:978)
        at org.apache.flink.streaming.runtime.tasks.StreamTask.handleAsyncException(StreamTask.java:952)
        at org.apache.flink.streaming.runtime.tasks.SystemProcessingTimeService$RepeatedTriggerTask.run(SystemProcessingTimeService.java:332)
        at java.util.concurrent.Executors$RunnableAdapter.call(Executors.java:511)
        at java.util.concurrent.FutureTask.runAndReset(FutureTask.java:308)
        at java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.access$301(ScheduledThreadPoolExecutor.java:180)
        at java.util.concurrent.ScheduledThreadPoolExecutor$ScheduledFutureTask.run(ScheduledThreadPoolExecutor.java:294)
        at java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1149)
        at java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:624)
        at java.lang.Thread.run(Thread.java:748)
Caused by: org.apache.flink.streaming.runtime.tasks.TimerException: org.apache.beam.sdk.util.UserCodeException: java.io.IOException: KafkaWriter : failed to send 1 records (since last report)
        at org.apache.flink.streaming.runtime.tasks.SystemProcessingTimeService$RepeatedTriggerTask.run(SystemProcessingTimeService.java:331)
        ... 7 common frames omitted
Caused by: org.apache.beam.sdk.util.UserCodeException: java.io.IOException: KafkaWriter : failed to send 1 records (since last report)
        at org.apache.beam.sdk.util.UserCodeException.wrap(UserCodeException.java:36)
        at org.apache.beam.sdk.io.kafka.KafkaWriter$DoFnInvoker.invokeFinishBundle(Unknown Source)
        at org.apache.beam.runners.core.SimpleDoFnRunner.finishBundle(SimpleDoFnRunner.java:237)
        at org.apache.beam.runners.flink.metrics.DoFnRunnerWithMetricsUpdate.finishBundle(DoFnRunnerWithMetricsUpdate.java:89)
        at org.apache.beam.runners.core.SimplePushbackSideInputDoFnRunner.finishBundle(SimplePushbackSideInputDoFnRunner.java:124)
        at org.apache.beam.runners.flink.translation.wrappers.streaming.DoFnOperator.invokeFinishBundle(DoFnOperator.java:836)
        at org.apache.beam.runners.flink.translation.wrappers.streaming.DoFnOperator.checkInvokeFinishBundleByTime(DoFnOperator.java:829)
        at org.apache.beam.runners.flink.translation.wrappers.streaming.DoFnOperator.lambda$open$2(DoFnOperator.java:488)
        at org.apache.flink.streaming.runtime.tasks.SystemProcessingTimeService$RepeatedTriggerTask.run(SystemProcessingTimeService.java:326)
        ... 7 common frames omitted
Caused by: java.io.IOException: KafkaWriter : failed to send 1 records (since last report)
        at org.apache.beam.sdk.io.kafka.KafkaWriter.checkForFailures(KafkaWriter.java:124)
```

Author → Compose → **Deploy**

…monitor your pipeline's performance…



STREAM PIPELINE

Env: QAL

Region: US-WEST-2

Runner: Flink

IKS Version: IKS2

Pipeline Status: STOPPED

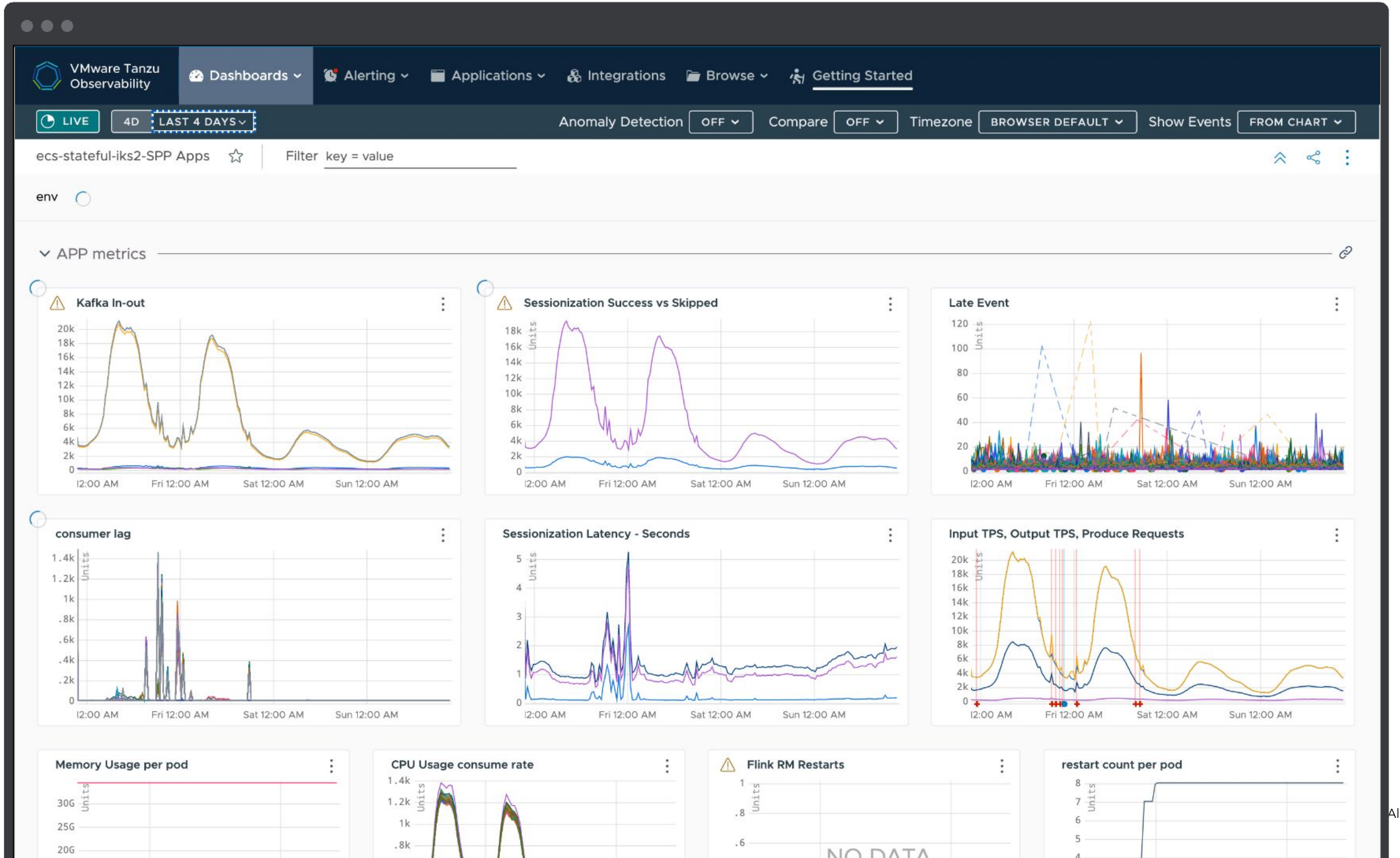
Namespace: data-processingpt-ecsstatefuliks2-usw2-qal

Cost Dashboard: QlikSense

Splunk Logs Wavefront Dashboard MDR Properties MDR Lineage

Author → Compose → **Deploy**

…and set up alerts



Anomaly Detection OFF

Compare OFF

Timezone BROWSER DEFAULT

Show Events FROM CHART

Filter key = value

Multiple entry points to the platform

Our native web experience

Our API
(e.g., Gitops, ad hoc scripts)

Third-party apps using our API
(e.g., feature processing,
stream materialization)

Stream Processing Platform Tech Stack

Tech Stack Overview

Customer Experience

Behind-the-scenes

Application Layer

Processor CI/CD Layer

UX Layer

Control Layer

Pipeline CI/CD Layer

Runtime Layer

Infrastructure Layer

beam

GitHub Jenkins JFrog Artifactory

React splunk> WAVEFRONT Qlik

spring Amazon Aurora

Jenkins JFrog Artifactory argo podman

Flink RocksDB Amazon S3

kubernetes aws

intuit

Application Layer

beam

Guiding principle: Developer flexibility

Components

- **SDK libraries**

Core functions

- **Auto Kafka configuration**
- **Data access policy handling**
- **Metrics collection**

Runtime Layer

Application Layer

Processor CI/CD Layer

UX Layer

Control Layer

Pipeline CI/CD Layer

Runtime Layer

Infrastructure Layer

beam

GitHub Jenkins JFrog Artifactory

React splunk> WAVEFRONT Qlik

spring Amazon Aurora

Jenkins JFrog Artifactory argo podman

Flink RocksDB Amazon S3

kubernetes aws

Runtime Layer

Flink Amazon S3 RocksDB

Guiding principle: Scalability

Components

- **Flink application cluster**
- **S3 for fault tolerance**

Core functions

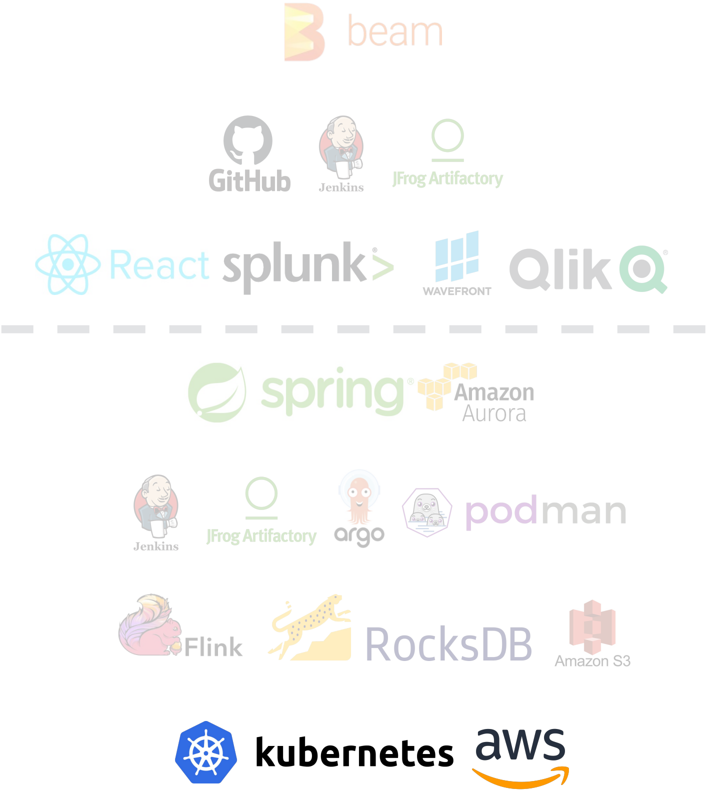
- **Stateful processing support**
- **Fault tolerance and at-least-once processing**
- **Low deploy/restart latency**
- **Health metrics**
- **Highly tunable and configurable via UX Layer**
- **Auto-scaling**

Infrastructure Layer

Customer Experience

Behind-the-scenes

- Application Layer
- Processor CI/CD Layer
- UX Layer
- Control Layer
- Pipeline CI/CD Layer
- Runtime Layer
- **Infrastructure Layer**



Infrastructure Layer

kubernetes aws

Guiding principle: Multi-tenancy

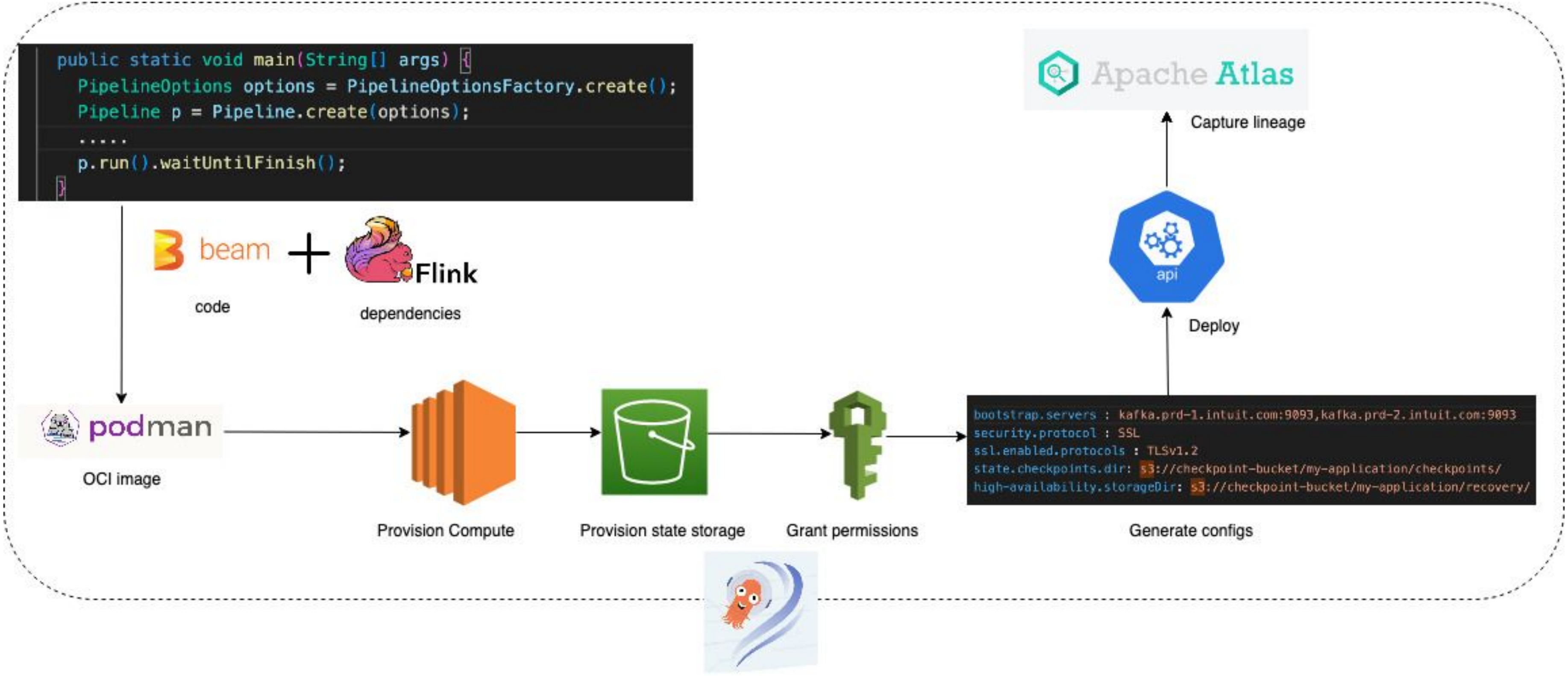
Components

- Kubernetes clusters on AWS EKS

Core functions

- Namespace isolation
- Low deploy/restart latency
- Rich operational metrics
- Fault tolerance
- Billing tags
- Multi-cluster topology

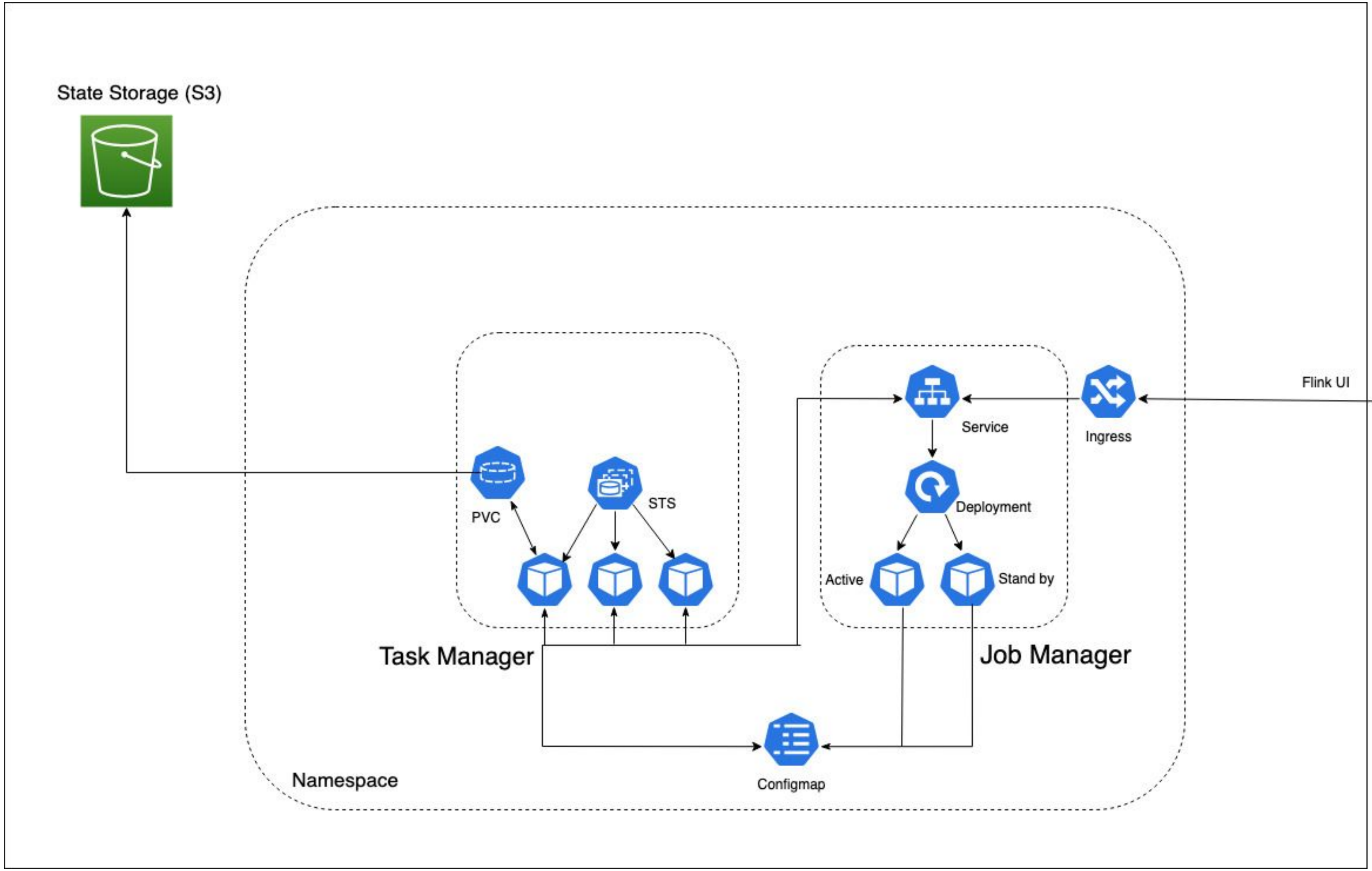
Deployment workflow



```java
public static void main(String[] args) {
  PipelineOptions options = PipelineOptionsFactory.create();
  Pipeline p = Pipeline.create(options);
  .....
  p.run().waitUntilFinish();
}
```

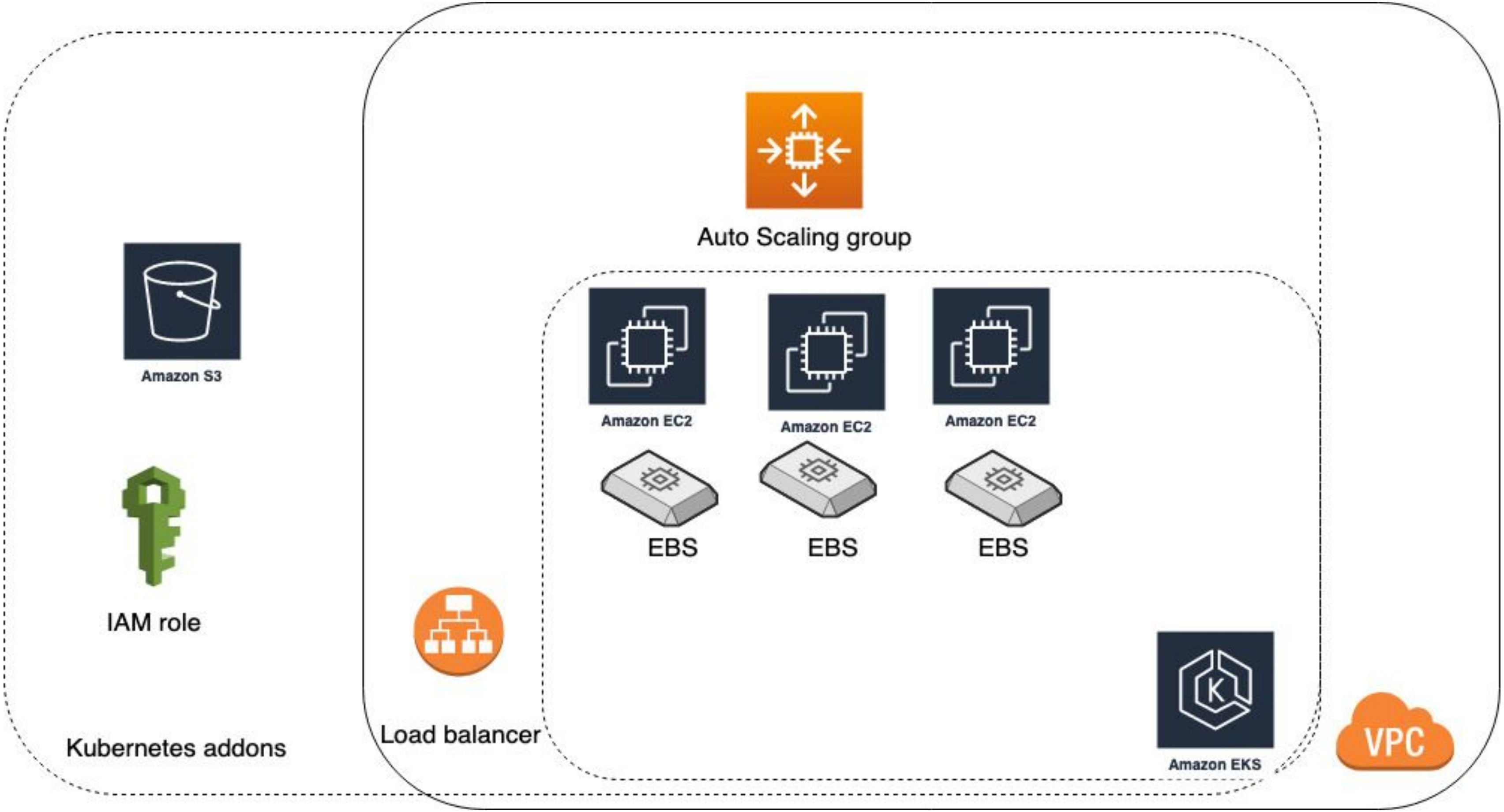
```
bootstrap.servers : kafka.prd-1.intuit.com:9093,kafka.prd-2.intuit.com:9093
security.protocol : SSL
ssl.enabled.protocols : TLSv1.2
state.checkpoints.dir: s3://checkpoint-bucket/my-application/checkpoints/
high-availability.storageDir: s3://checkpoint-bucket/my-application/recovery/
```

Anatomy of a flink application on Kubernetes



What’s underneath?

AWS components



Learnings

- **Runner migration**
  - Samza -> Flink
- **Multi tenancy model**
  - Disruptions caused by scheduler
  - Disk isolation

Summary & Learnings

Guiding Principles

beam	Developer flexibility
Flink	Scalability
kubernetes	Multi-tenancy

Lessons Learned

- **Having runner flexibility can be really nice**
  - We changed our runtime from Samza to Flink, and customers didn't have to write any new code
- **Compute isolation issues can surprise you at scale**
  - Pod disruptions caused by k8s scheduler made full multi-tenancy tricky to stabilize
  - Lack of disk isolation can become a performance bottleneck

Summary & Learnings

Tech Stack Layer	Core Technology	Guiding Principle	Lessons Learned
Application	beam	Developer flexibility	Runner flexibility allowed us to change runtime from Samza to Flink
Runtime	Flink	Scalability	Flink experiences processing disruption if k8s scheduler is overly aggressive
Infrastructure	kubernetes	Multi-tenancy	Lack of disk isolation on k8s resources can become a performance bottleneck

Featured Use Case
Golden Signal for Services

Service Golden Signals

System defined

Availability

Success Rate of Service calls.

Requests

Measure of demand / load being placed on the system..

Errors

Rate of requests that are failing. (e.g. HTTP 500s)

Latency

Time it takes to service a request. Typically measured across percentiles..

Opinionated Signals

Health

Health of an application or service in real-time. May be redefined by application teams. Typically based on aggregate availability.

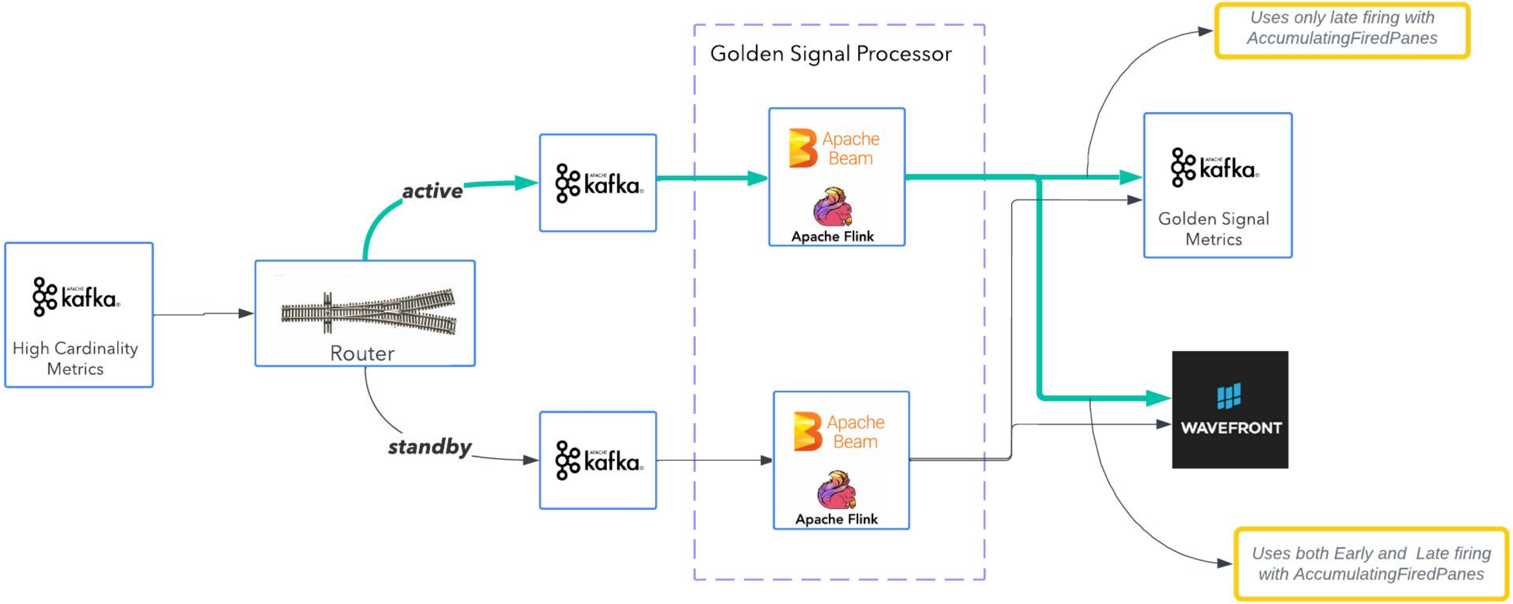
Saturation / Utilization:

How "full" a service is. Percent of “max capacity” being used. Varies by service constraints. e.g. nodes, memory, CPU, networking, auto-scaling limits, etc.

Google SRE

Golden Signals for ~2000 services at Intuit

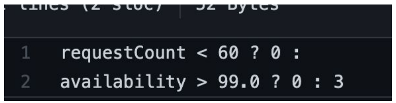
High level Design



Customization using Side Input

- An additional input that your DoFn can access each time it processes an element in the input PCollection
- Health metrics and few tags are overridable

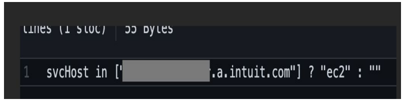
Health metrics example



lines (2 sloc) | 52 Bytes

```
requestCount < 60 ? 0 :
availability > 99.0 ? 0 : 3
```

Swimlane example

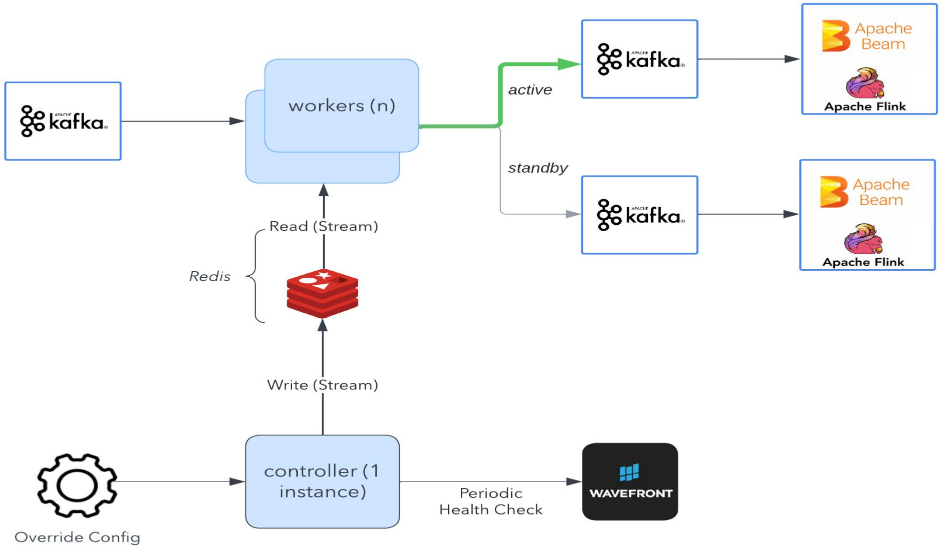


lines (1 sloc) | 55 Bytes

```
svcHost in ["[redacted].a.intuit.com"] ? "ec2" : ""
```

- Users use the GitOps model to customize their service
- Override configuration is stored in S3 (Gitops → Jenkins → Upload to S3)
- Pipeline fetches from S3 every 5 min using Beam Side Input

Router



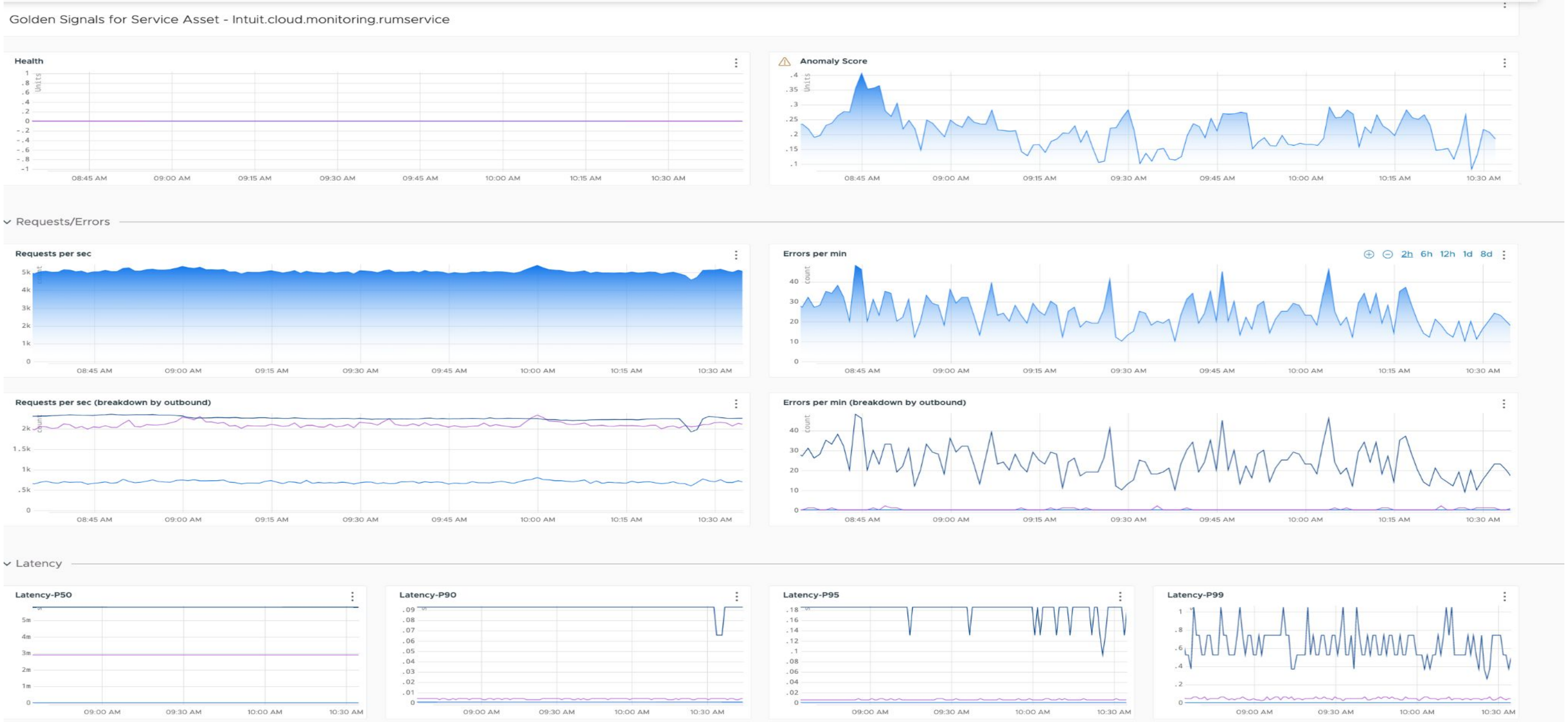
Router Component helps to

- Achieve SLA of 3 min
- Zero Downtime deployment

Controller - Sends a message to worker to flip topics when health check fails

Workers: Reads from source and publishes to destination topic

Golden Signal Dashboard



Q&A