NETFLIX


# Beaming Flink to the Cloud @ Netflix

Monal Daxini

Engineering Manager

Stream Processing, Real Time Data Infrastructure

@monaldax @Netflix #keystone

Flink Forward

Sep 12 2016

# World's Leading Internet Streaming Service

(Global launch Jan 6, 2016)

# Netflix Service Scale

- 83+ Million Members, 190+ Countries
- 1000+ device types
- 35% of downstream Internet traffic

# Netflix Service Scale - Daily viewing hours

**125,000,000,000+**

Whoa!

PIPELINE Keystone Events Processed / day

# 1,000,000,000,000+

## 1.4 PB

That's a huge number!

# Event Scale

## Peak

- 1T unique events ingested/day
- 16M / sec
- 43GB / sec
- 10MB / message

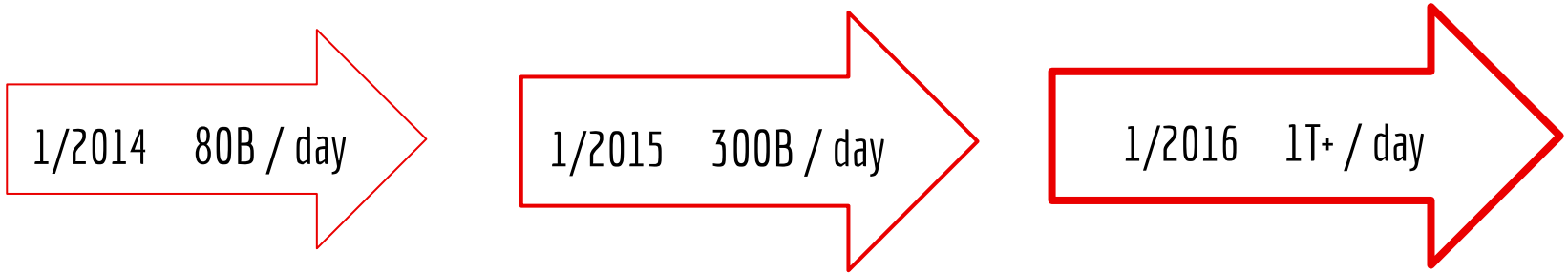
## Daily Averages

- 1T+ events processed
- 600B unique events ingested
- 1.4 PB / day
- 4K / event

# Keystone Scale

**99.99% +** Availability / **Four 9s**

# Keystone Events Trend

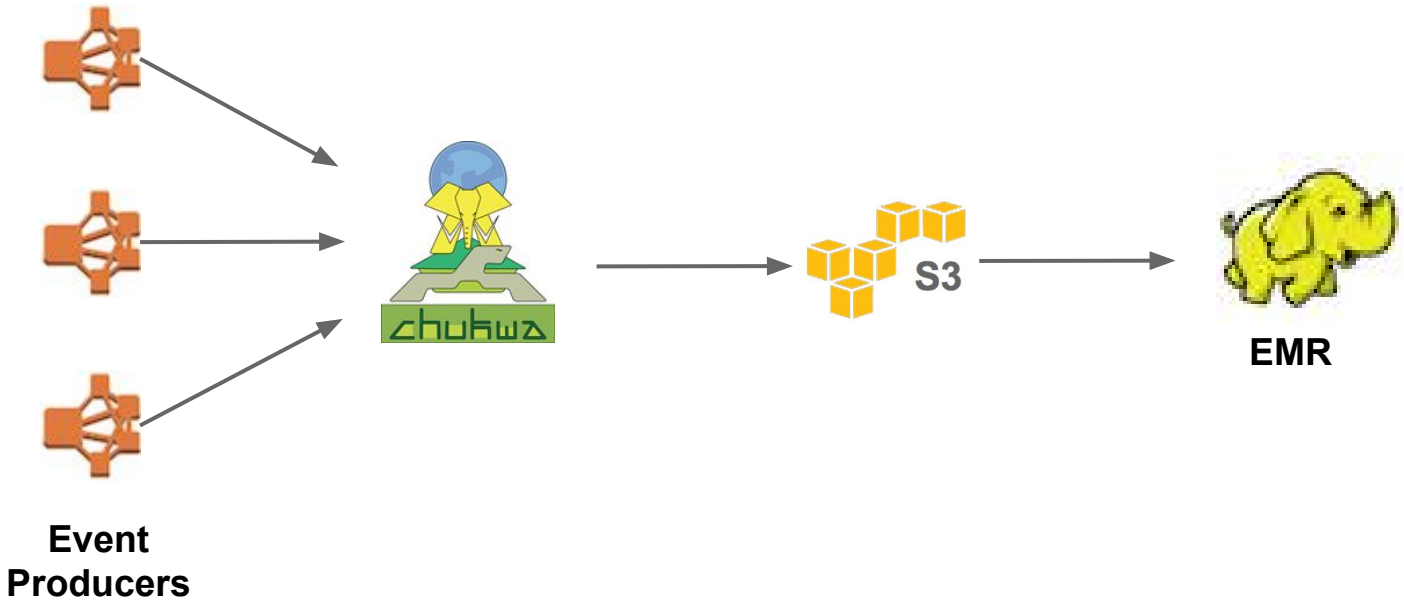


# Evolution

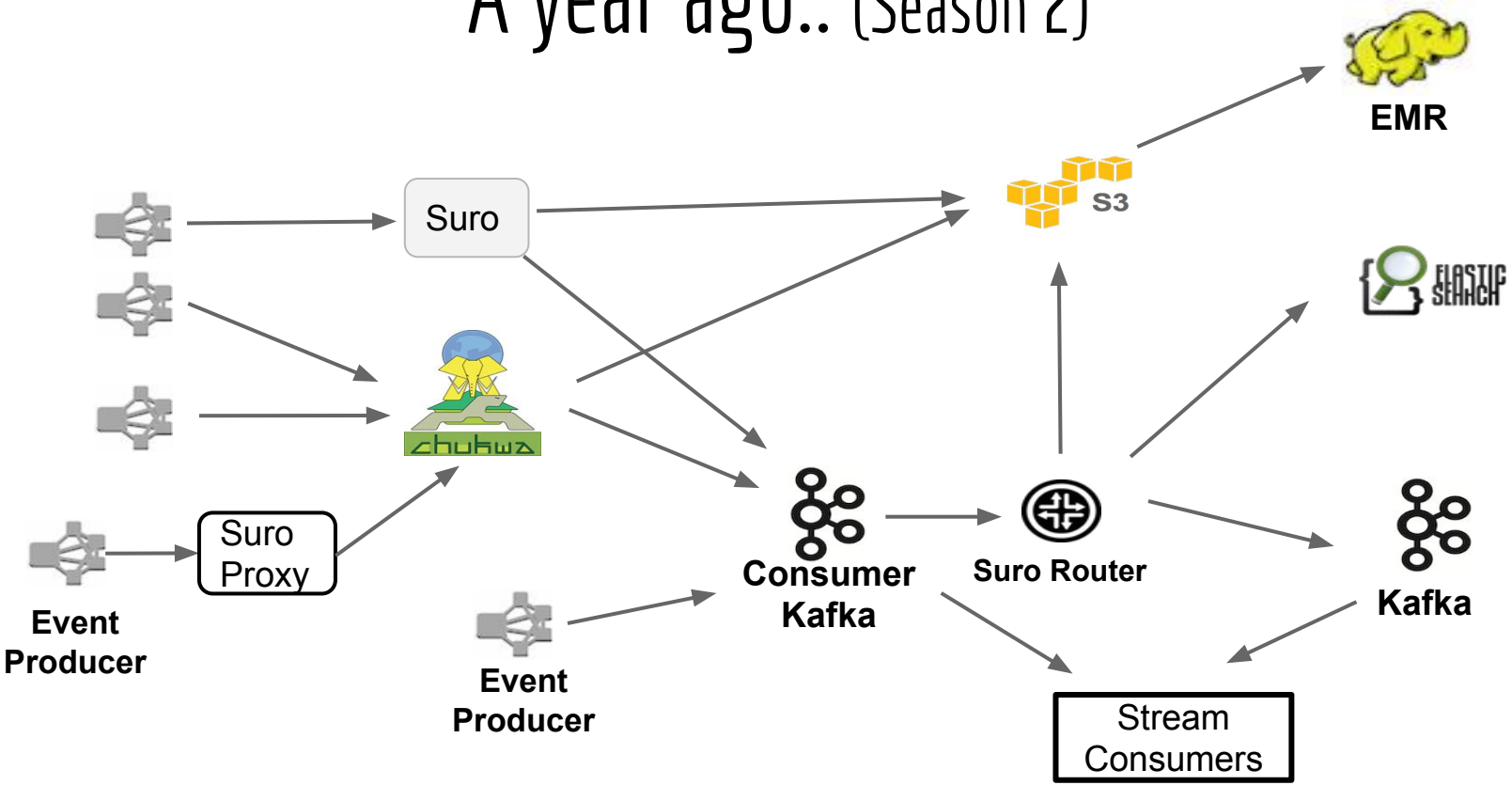
SPaaS

# Goal - Migrate 1.3 PB of event data to a new Pipeline in flight, while ensuring data diff < 0.1%

# Few years ago (Season 1)



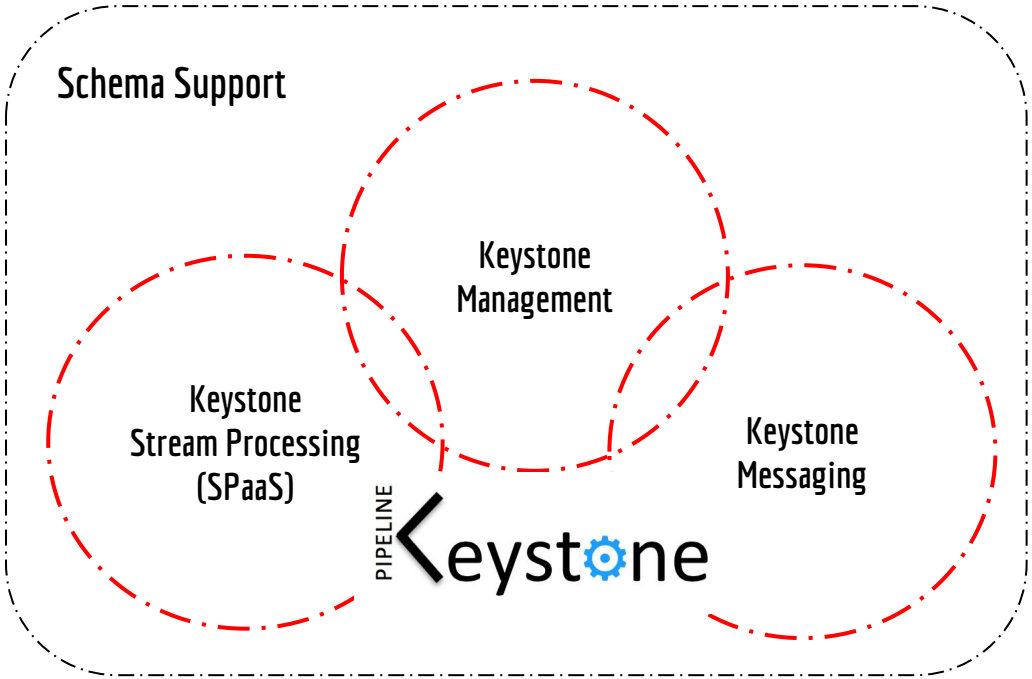
# A year ago.. (Season 2)



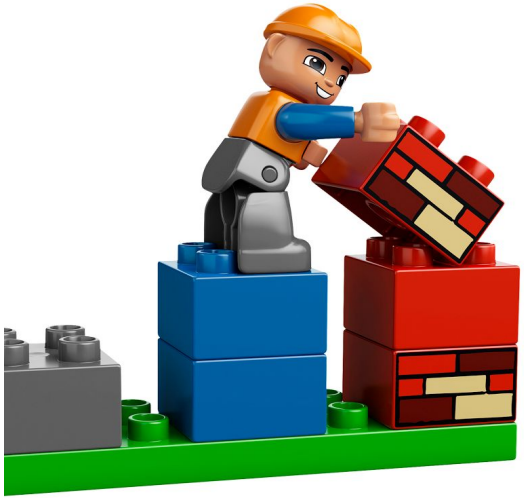
# Season 3

# Keystone

100% in AWS



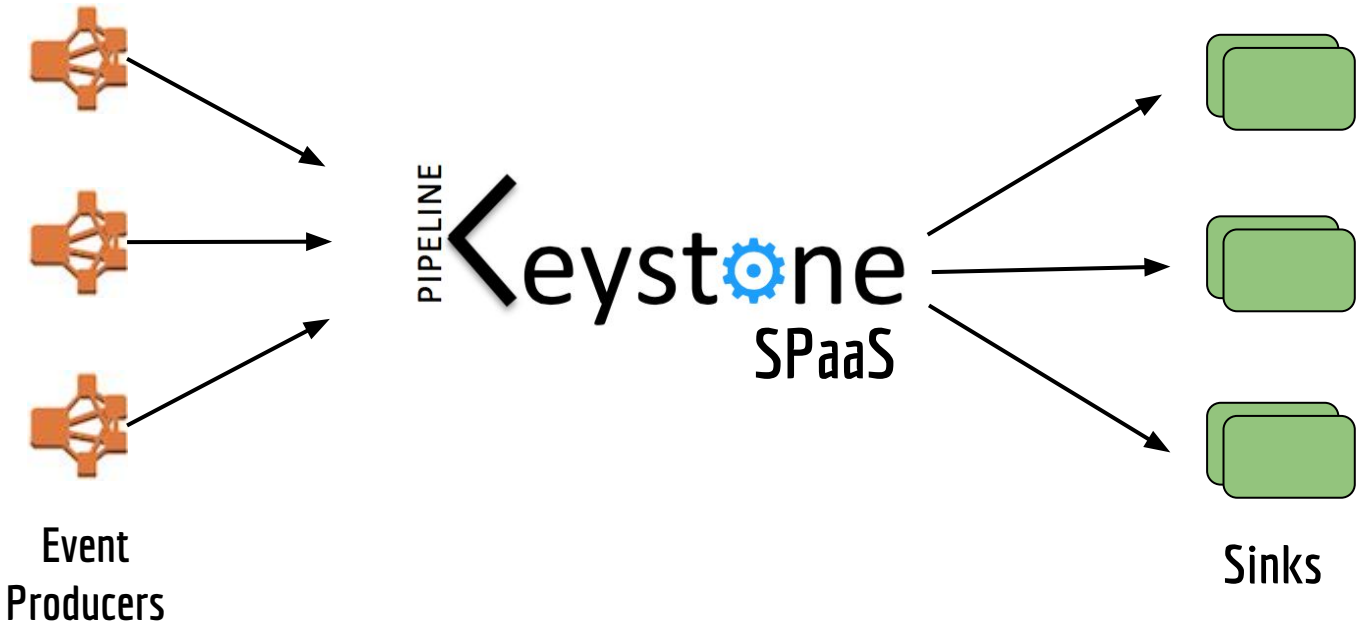
# Our Philosophy



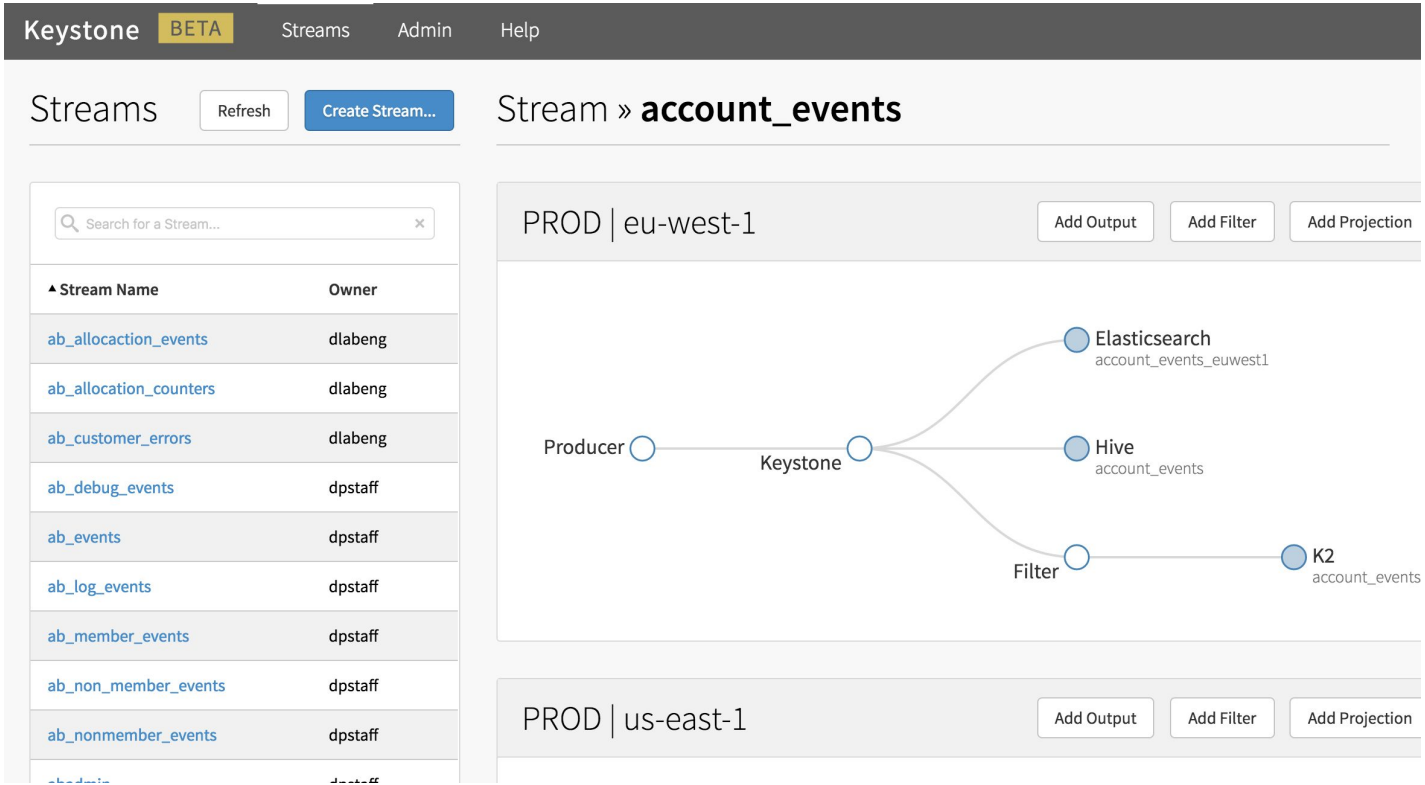
Create Duplo®  Blocks:

Let reusability drive new value

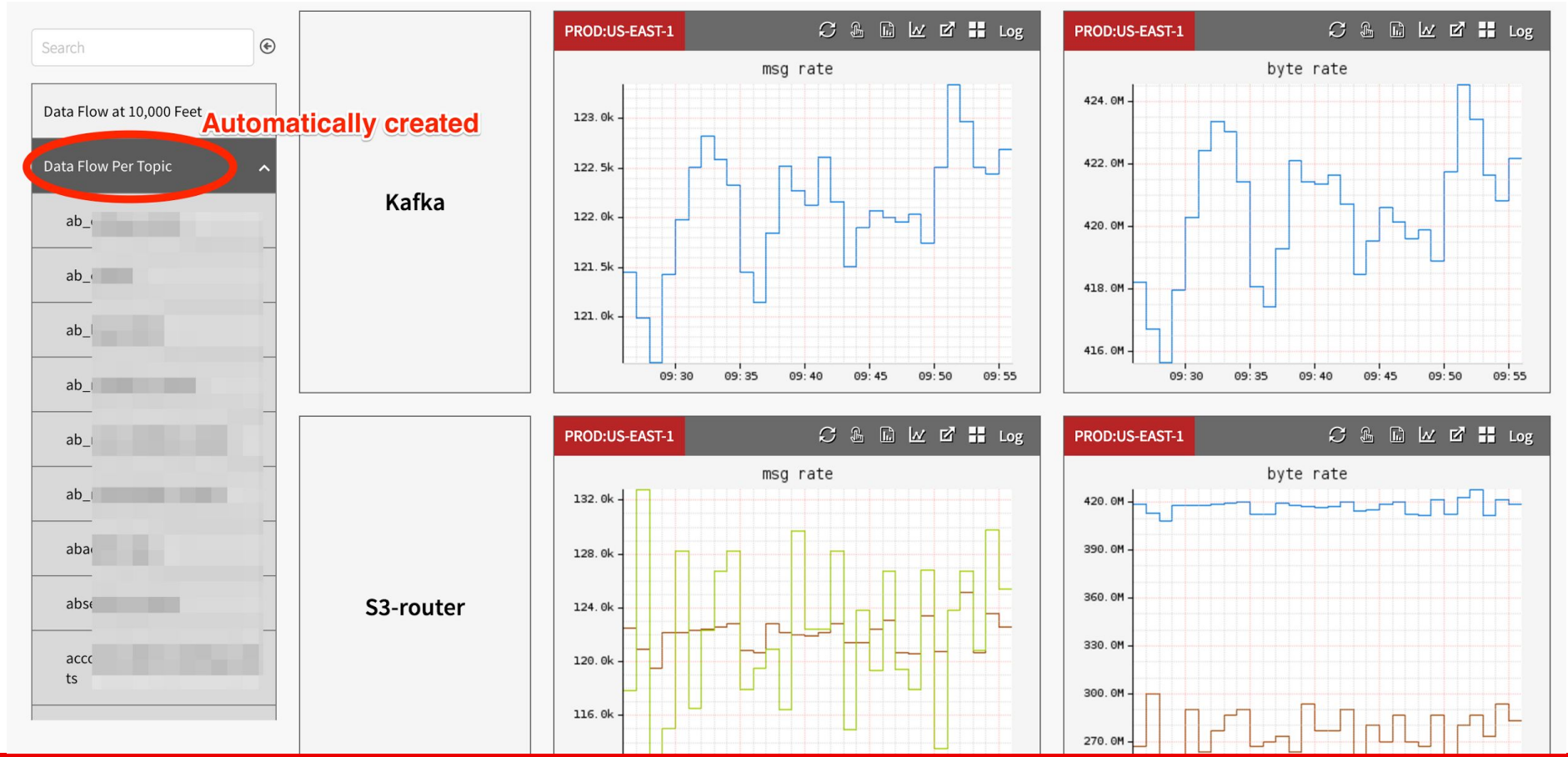
# Q4 2015



# Keystone Management



# Per Stream Auto Dashboard

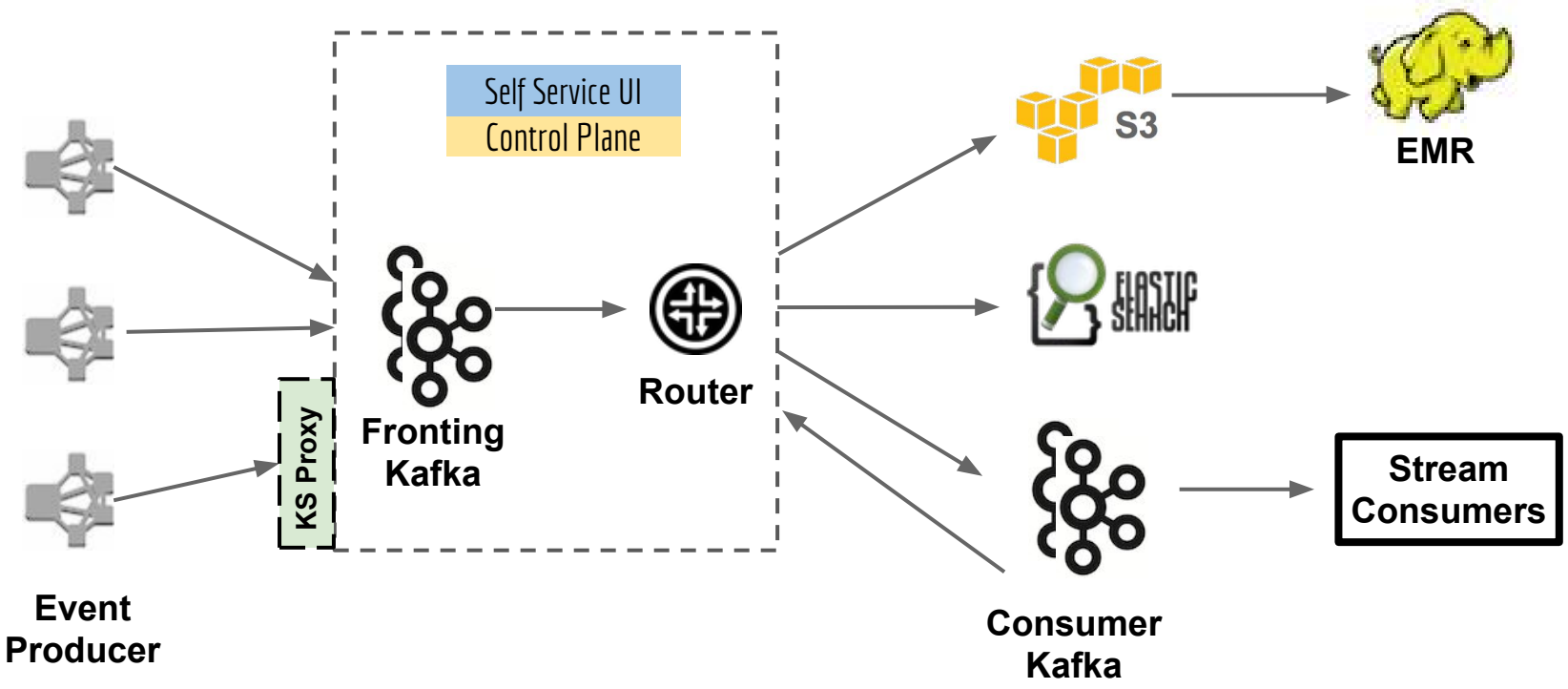


100% in AWS

24 x 7

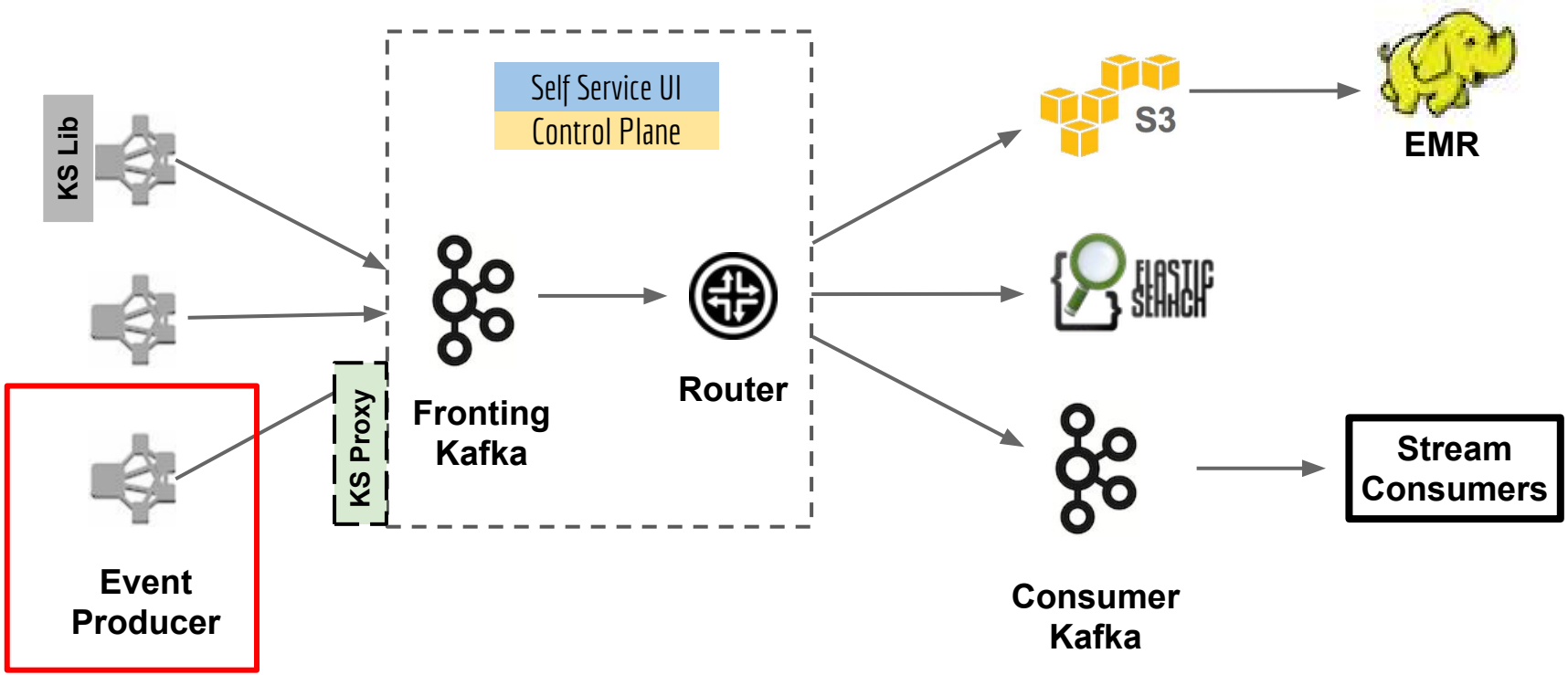
Region failover

# Keystone SPaaS

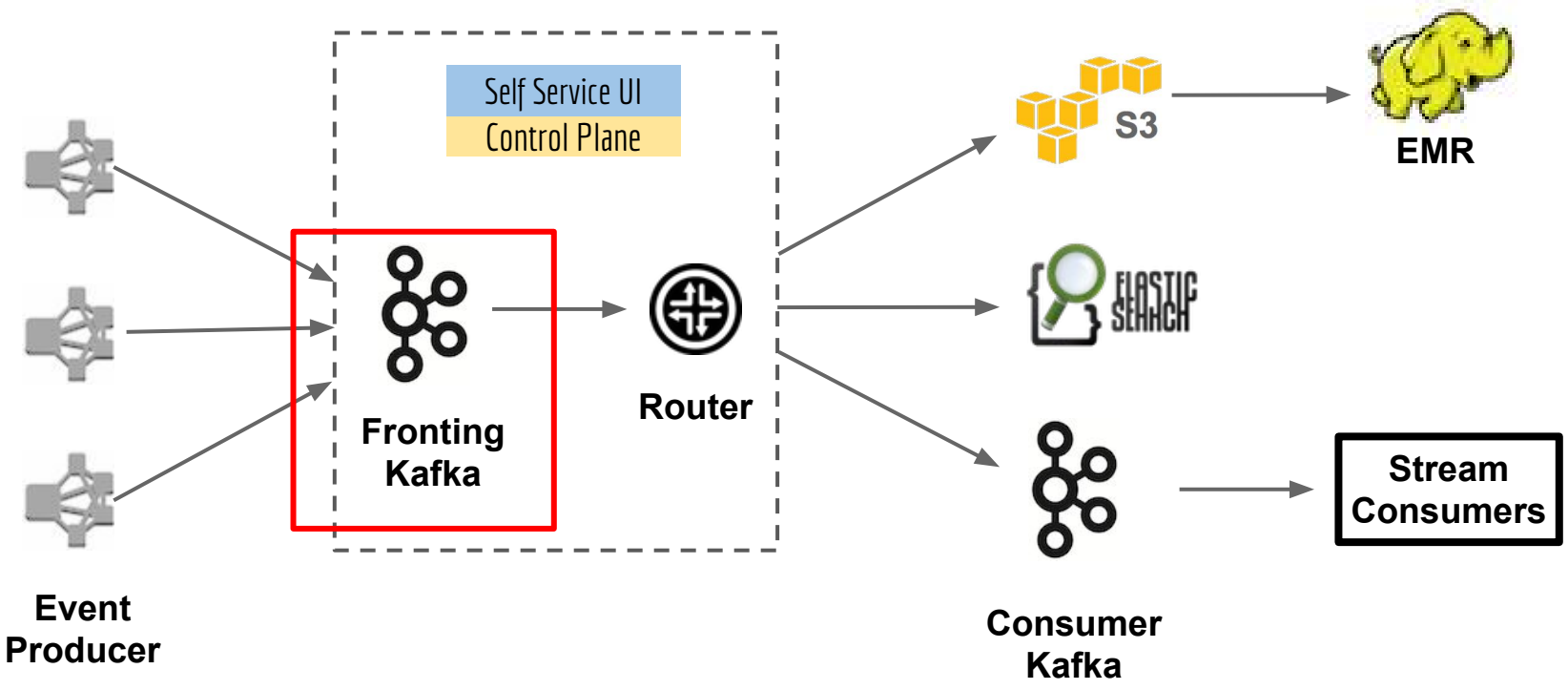


# Event flow
## Keystone Pipeline As a Service

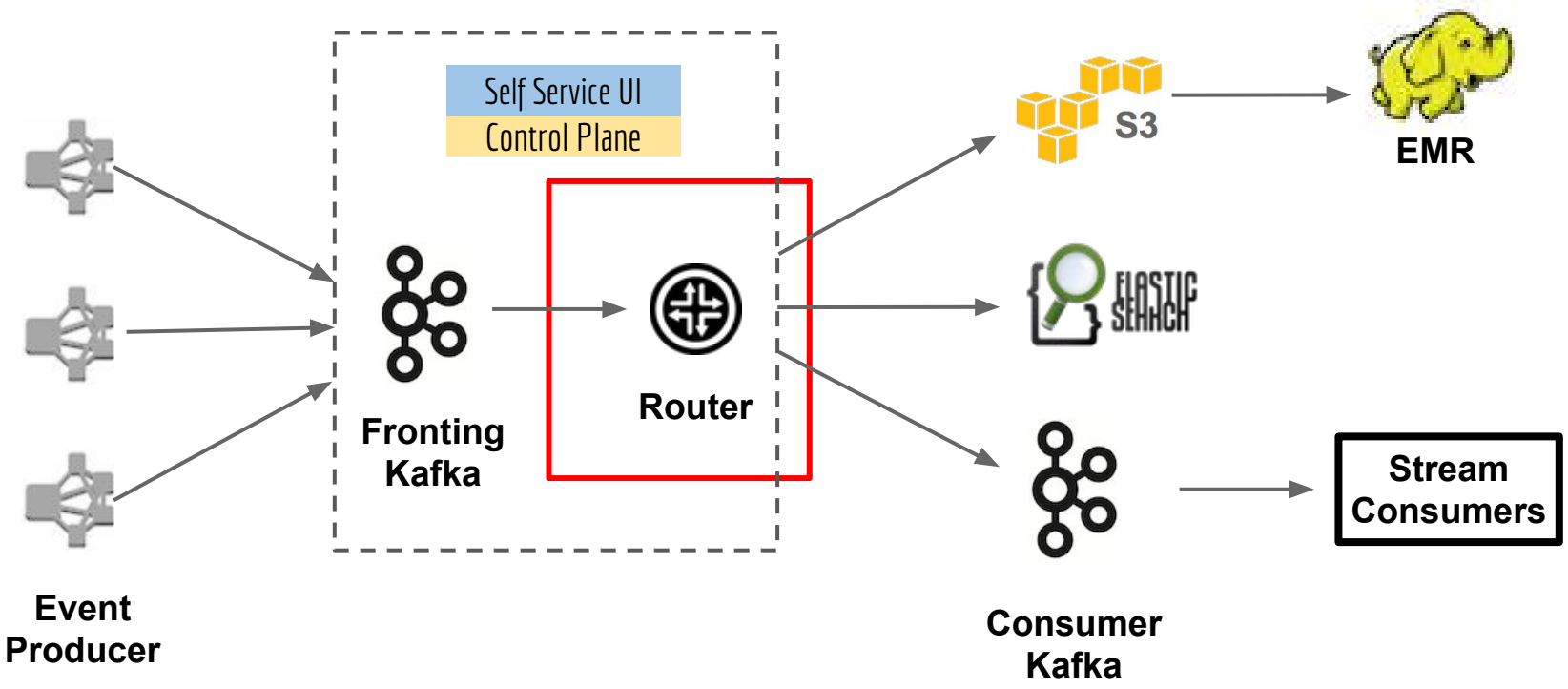
# Keystone


KS Lib
Self Service UI
Control Plane
S3
EMR
Fronting Kafka
Router
KS Proxy
Event Producer
Consumer Kafka
Stream Consumers

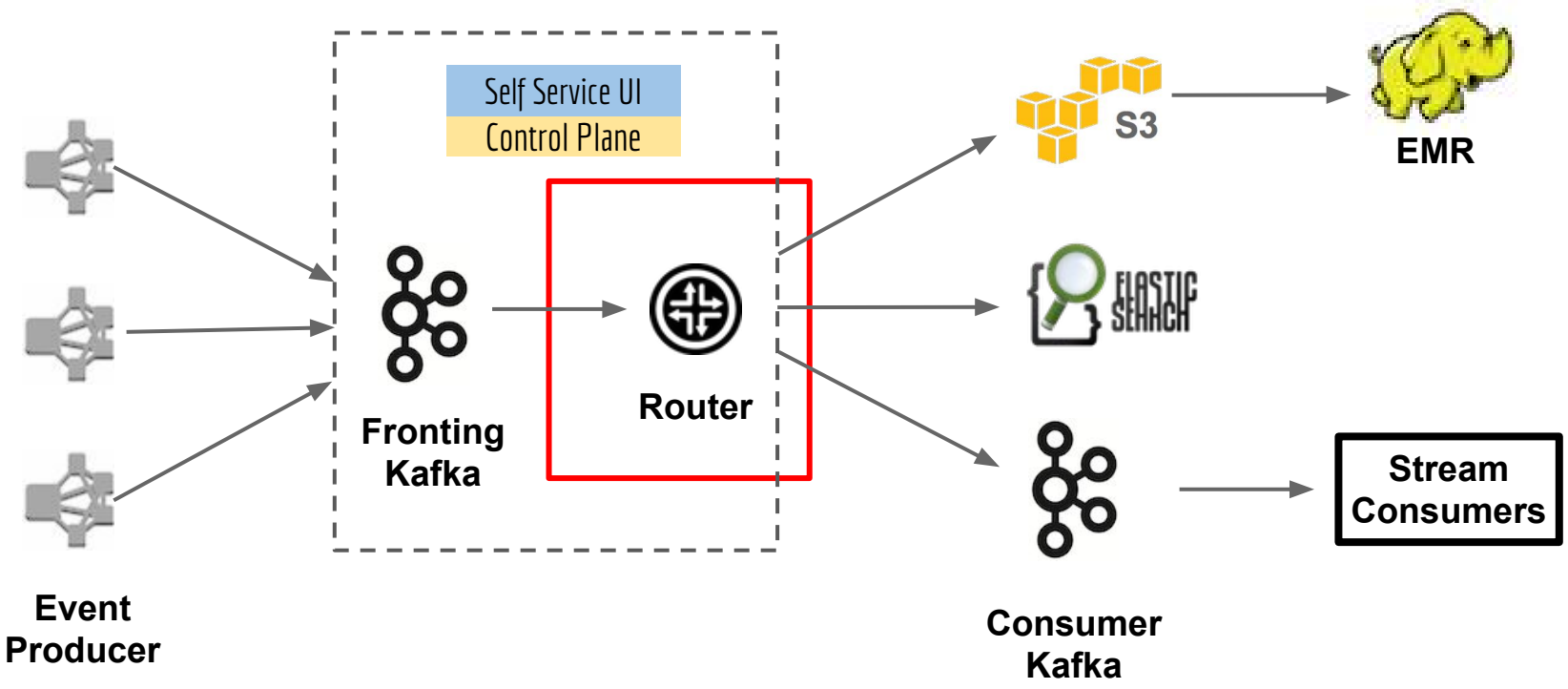
# Keystone


Self Service UI
Control Plane
Fronting Kafka
Router
S3
EMR
Stream Consumers
Event Producer
Consumer Kafka

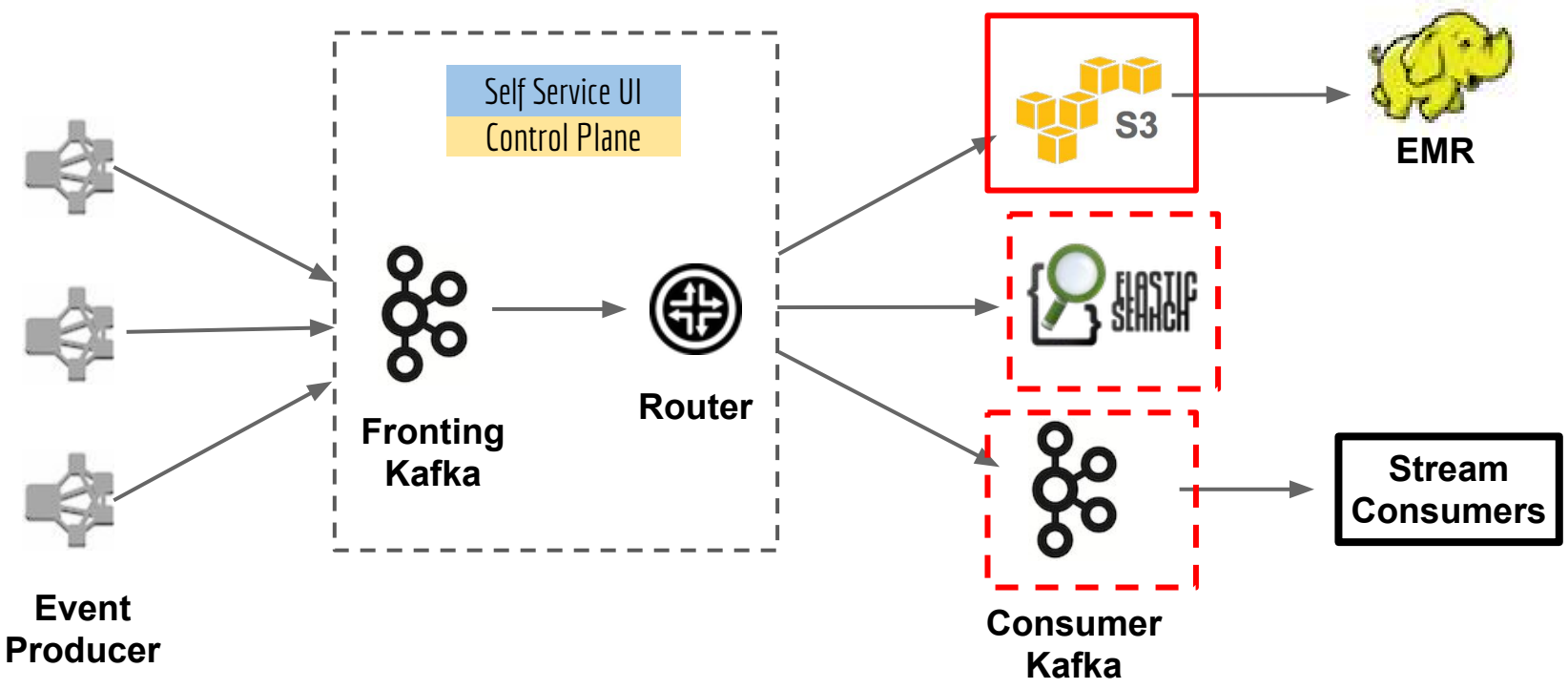
# Keystone


Self Service UI
Control Plane
S3
EMR
Fronting Kafka
Router
Stream Consumers
Event Producer
Consumer Kafka

# Keystone

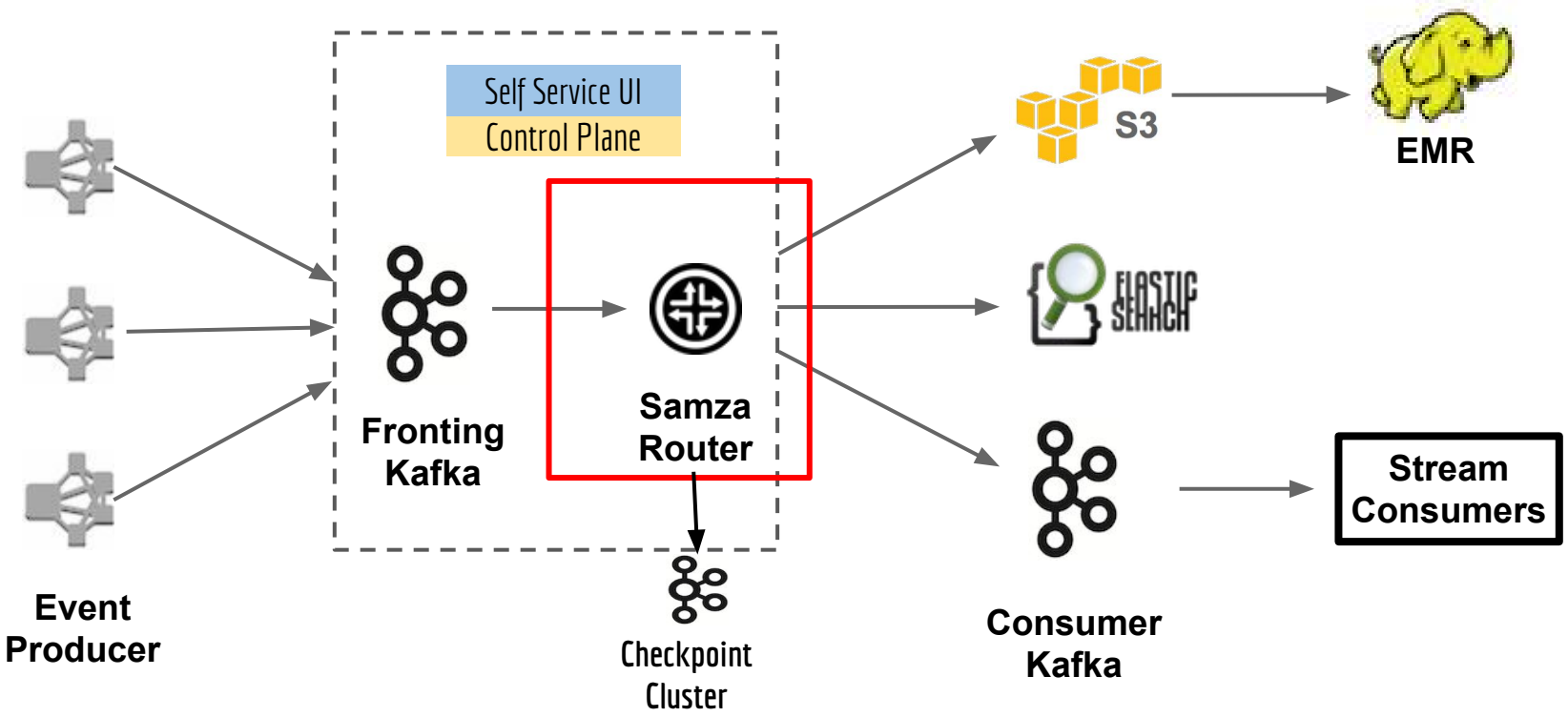

Self Service UI
Control Plane
S3
EMR
Fronting Kafka
Router
Stream Consumers
Event Producer
Consumer Kafka

# Keystone


Self Service UI
Control Plane
S3
EMR
ELASTIC SEARCH
Fronting Kafka
Router
Stream Consumers
Event Producer
Consumer Kafka


# Routing

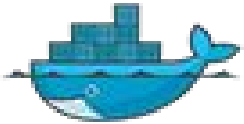
# Keystone


Self Service UI
Control Plane
S3
EMR
ELASTIC SEARCH
Fronting Kafka
Samza Router
Stream Consumers
Event Producer
Checkpoint Cluster
Consumer Kafka

# Details..

- Massively parallel use-case
  - Per element processing - declarative filtering & projection
- Stateless except Kafka offset checkpointing state

# Routing Infrastructure

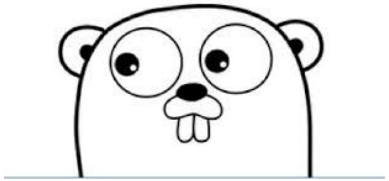



+

samza

0.9.1

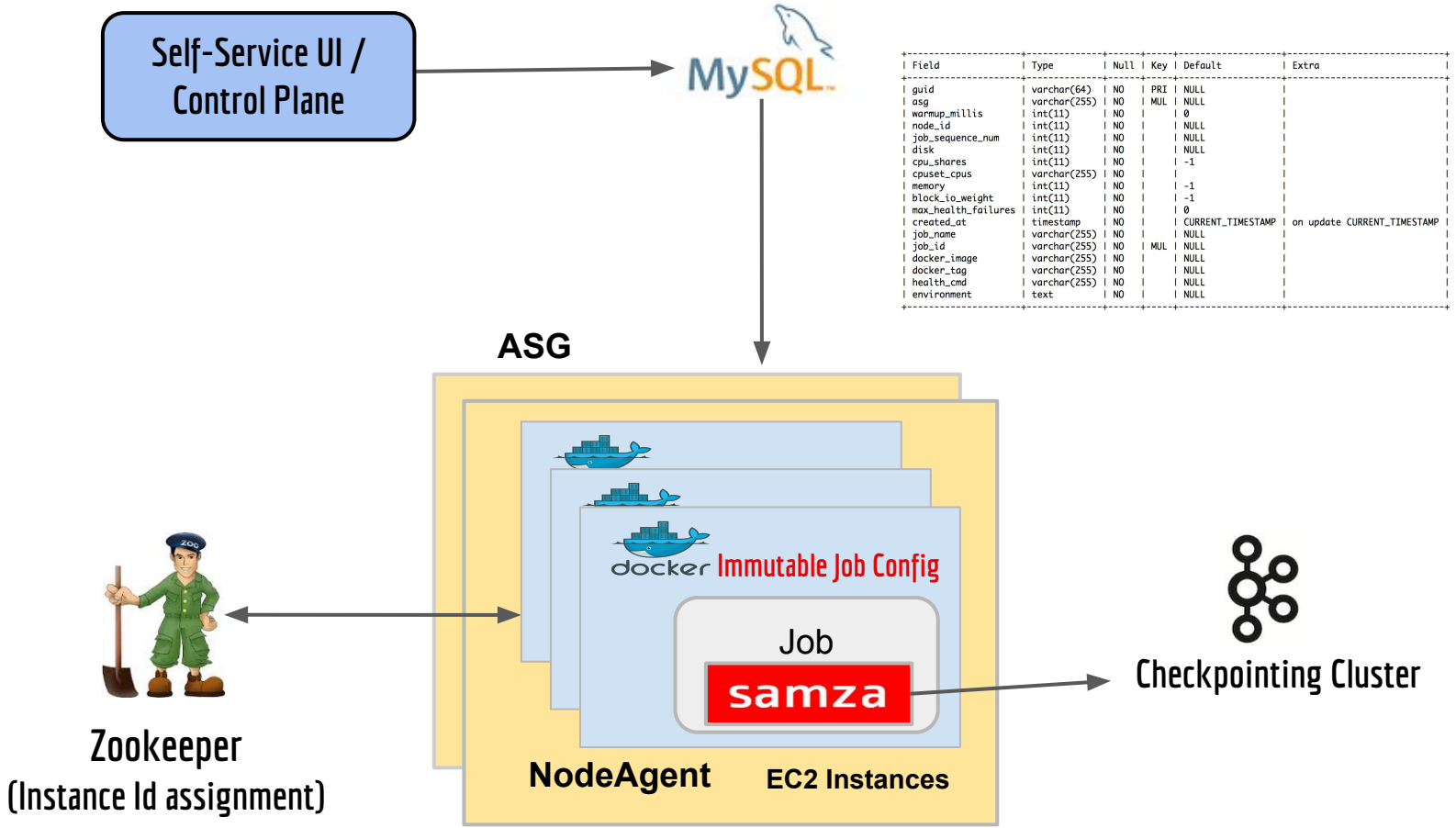
Go




+

**Checkpointing Cluster**

C language


Self-Service UI / Control Plane
MySQL
+-----------------------+--------------+------+-----+---------------------+-----------------------------+
| Field | Type | Null | Key | Default | Extra |
+-----------------------+--------------+------+-----+---------------------+-----------------------------+
| guid | varchar(64) | NO | PRI | NULL | |
| asg | varchar(255) | NO | MUL | NULL | |
| warmup_millis | int(11) | NO | | 0 | |
| node_id | int(11) | NO | | NULL | |
| job_sequence_num | int(11) | NO | | NULL | |
| disk | int(11) | NO | | NULL | |
| cpu_shares | int(11) | NO | | -1 | |
| cpuset_cpus | varchar(255) | NO | | | |
| memory | int(11) | NO | | -1 | |
| block_io_weight | int(11) | NO | | -1 | |
| max_health_failures | int(11) | NO | | 0 | |
| created_at | timestamp | NO | | CURRENT_TIMESTAMP | on update CURRENT_TIMESTAMP |
| job_name | varchar(255) | NO | | NULL | |
| job_id | varchar(255) | NO | MUL | NULL | |
| docker_image | varchar(255) | NO | | NULL | |
| docker_tag | varchar(255) | NO | | NULL | |
| health_cmd | varchar(255) | NO | | NULL | |
| environment | text | NO | | NULL | |
+-----------------------+--------------+------+-----+---------------------+-----------------------------+
ASG
docker
Immutable Job Config
Job
samza
NodeAgent
EC2 Instances
Zookeeper
(Instance Id assignment)
Checkpointing Cluster

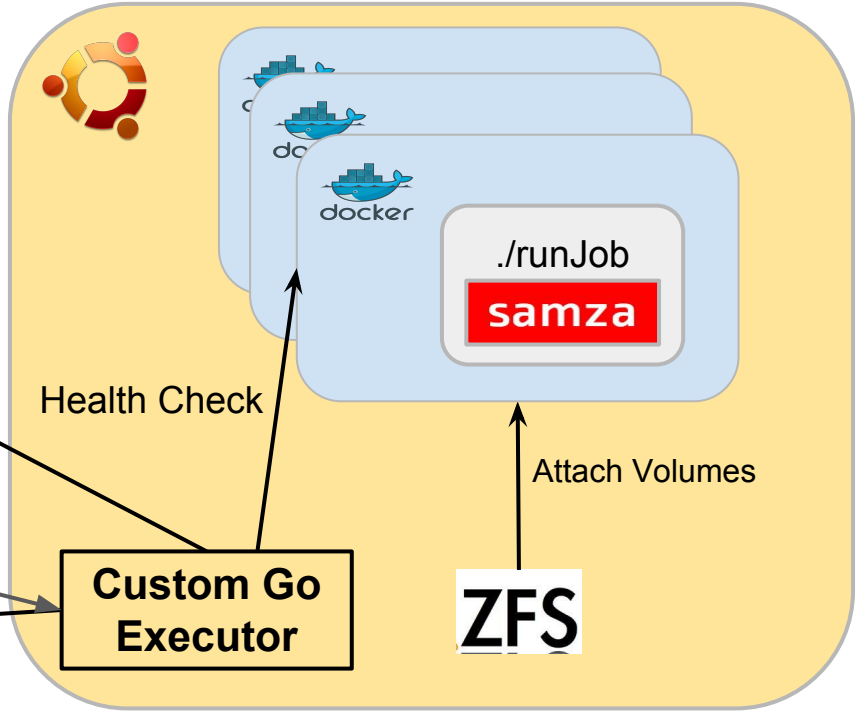
# What's running on the host?

| Field | Type | Null | Key | Default | Extra |
|---|---|---|---|---|---|
| guid | varchar(64) | NO | PRI | NULL | |
| asg | varchar(255) | NO | MUL | NULL | |
| warmup_millis | int(11) | NO | | 0 | |
| node_id | int(11) | NO | | NULL | |
| job_sequence_num | int(11) | NO | | NULL | |
| disk | int(11) | NO | | NULL | |
| cpu_shares | int(11) | NO | | -1 | |
| cpuset_cpus | varchar(255) | NO | | | |
| memory | int(11) | NO | | -1 | |
| block_io_weight | int(11) | NO | | -1 | |
| max_health_failures | int(11) | NO | | 0 | |
| created_at | timestamp | NO | | CURRENT_TIMESTAMP | on update CURRENT_TIMESTAMP |
| job_name | varchar(255) | NO | | NULL | |
| job_id | varchar(255) | NO | MUL | NULL | |
| docker_image | varchar(255) | NO | | NULL | |
| docker_tag | varchar(255) | NO | | NULL | |
| health_cmd | varchar(255) | NO | | NULL | |
| environment | text | NO | | NULL | |

Zookeeper
(Instance Id assignment)

Reconcile Loop - 1 min


Logs
Snapshots



Yes!  You inferred right!

# No Mesos  &  No Yarn

# Keystone Salient Features

# Keystone Scale

- **4000+** Kafka brokers
- **14000+** Samza Jobs
  - Running in docker containers
  - On **1600+** nodes

# Keystone Salient Features

100% in AWS

- At-least-once delivery semantics*
- Multi-Tenant
- Self Service
- Scalable
- Fault Tolerant

# Keystone Salient Features

- Event payload is immutable
- Inject event metadata
  - guid, timestamp, host, app
- Custom extensible wire protocol
- Kafka producer wrapper

# Keystone Salient Features

- Kafka Cluster failover
  - **Kafka Kong**
- Routing regional / failover
- Scales based on historical traffic
  - Externally driven

# Season 4

## Plot & Pilot...

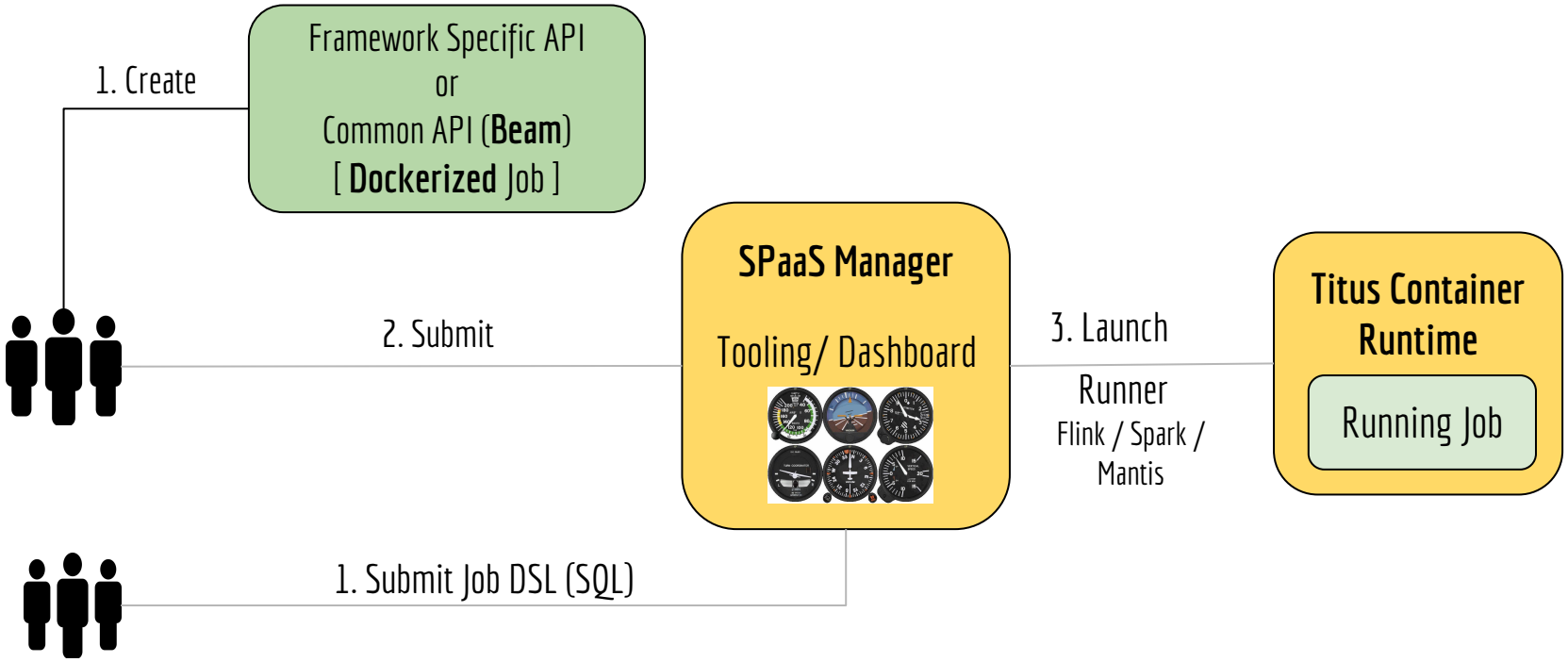

# **S**tream **P**rocessing **A**s a **S**ervice (SPaaS)

(more capable)

# SPaaS Vision (plot)

- Multi-tenant support for stateful stream processing apps
- Autoscaling managed infrastructure
- Support for schemas
- Self Service Tooling

# SPaaS Architecture (plot)



# SPaaS - *“Beam Me Up, Scotty ! ”*

Why Apache Beam?

- Portable API layer to build sophisticated data processing apps
  - Support multiple execution engines
- Unified model API over **bounded and unbounded data sources**
- Millwheel, FlumeJava, Dataflow model lineage

# SPaaS - Pilot

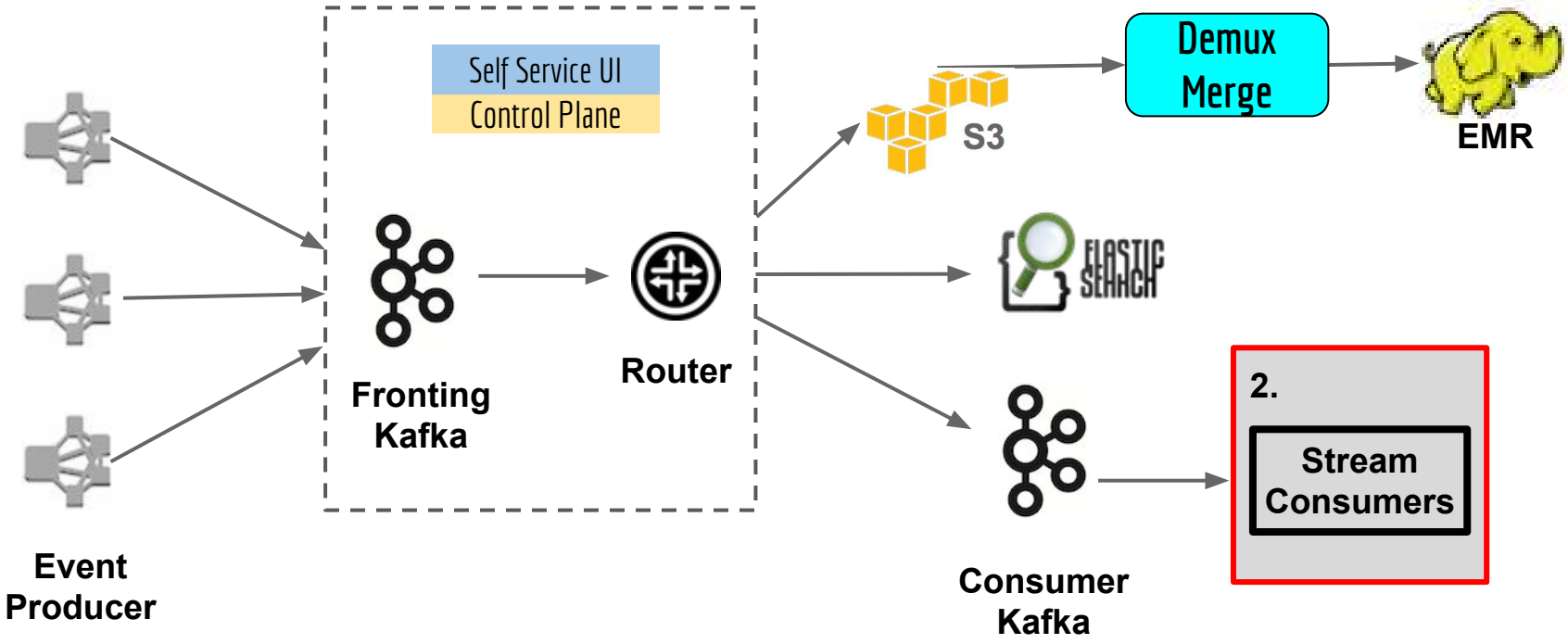
Iterative build out: 

then 

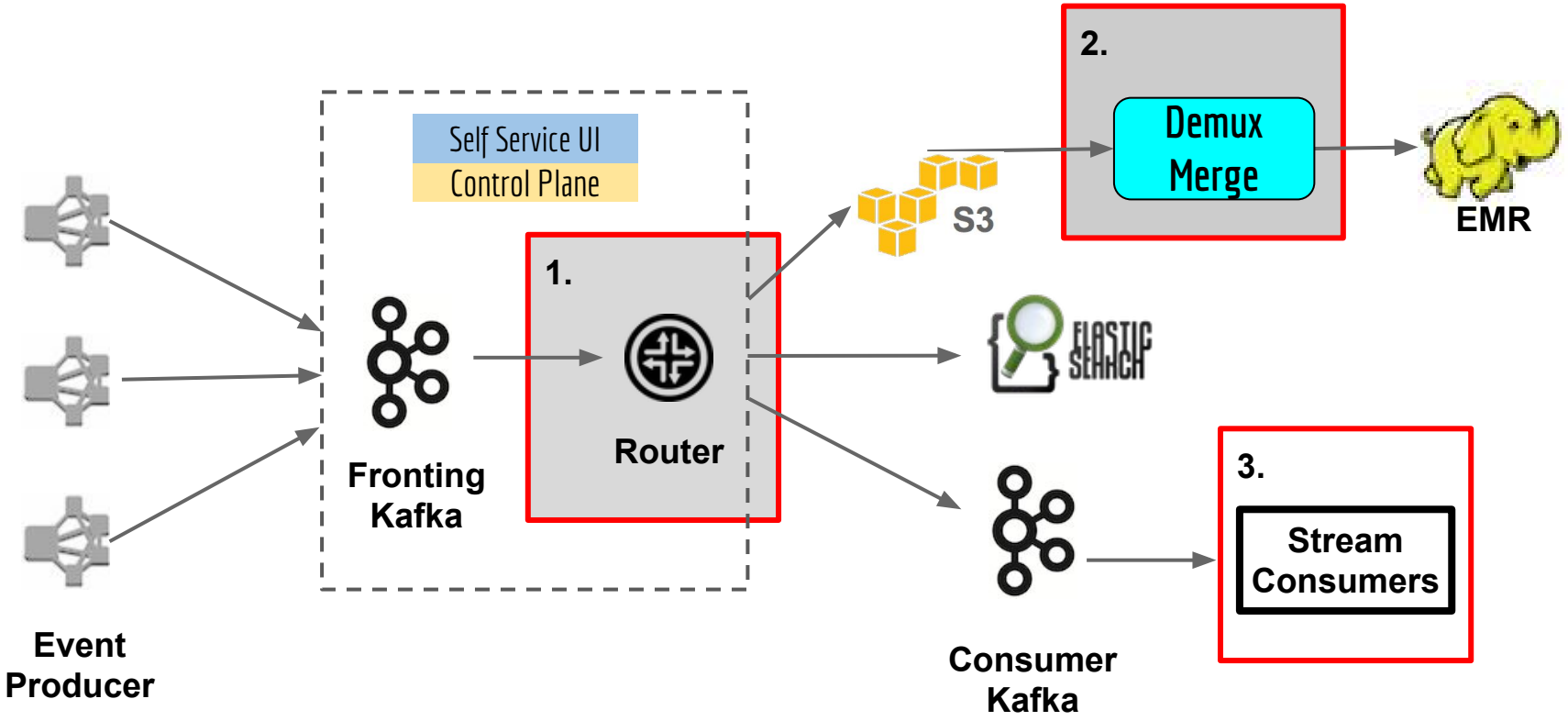
- **First -** Flink on ***Titus*** in VPC, AWS
  - ***Titus*** is a cloud runtime platform for container based jobs

- **Next -** Apache Beam and Flink runner

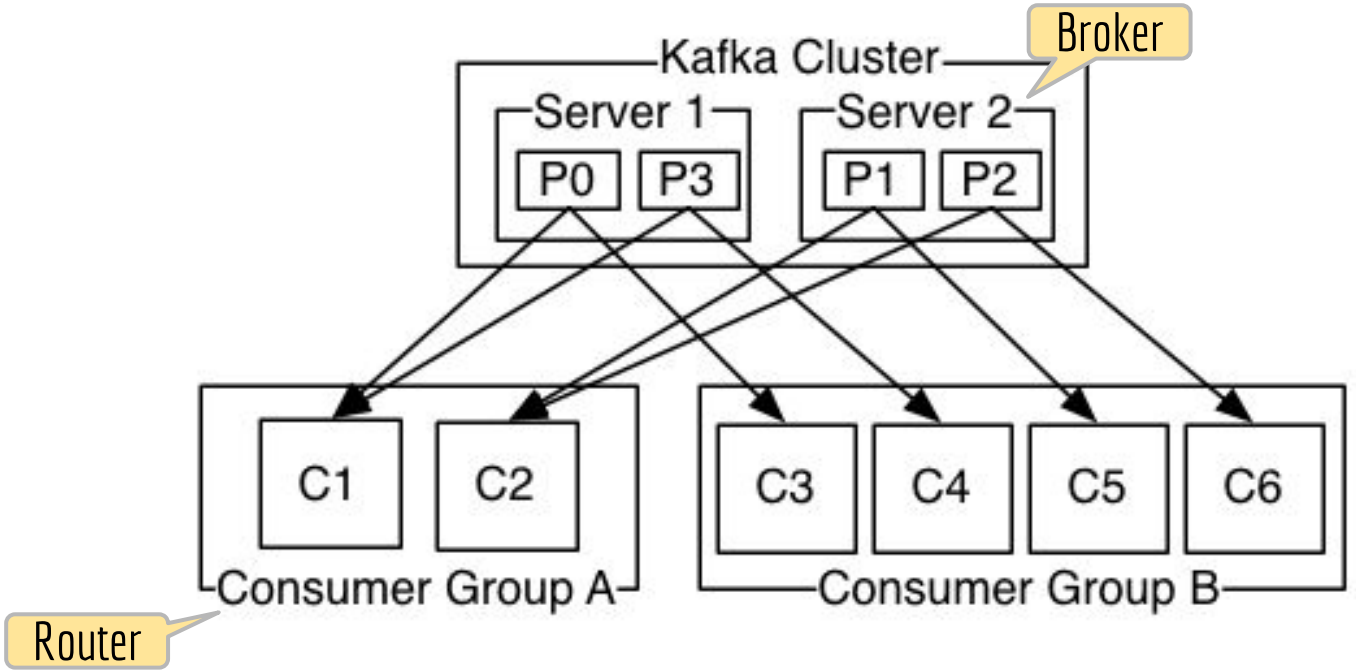
# Keystone SPaaS-Flink Pilot Use Cases



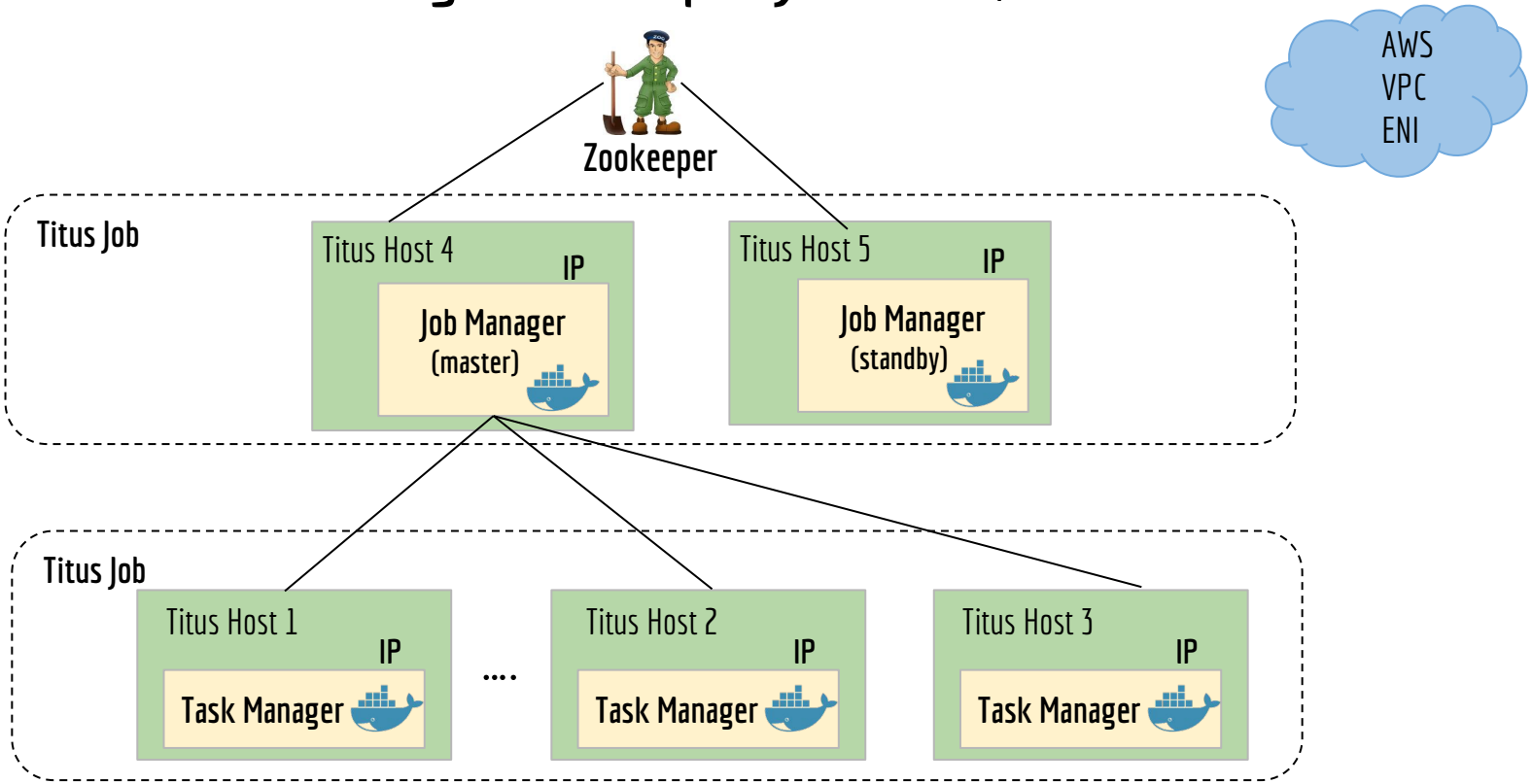
# Keystone SPaaS-Flink Pilot Use Cases



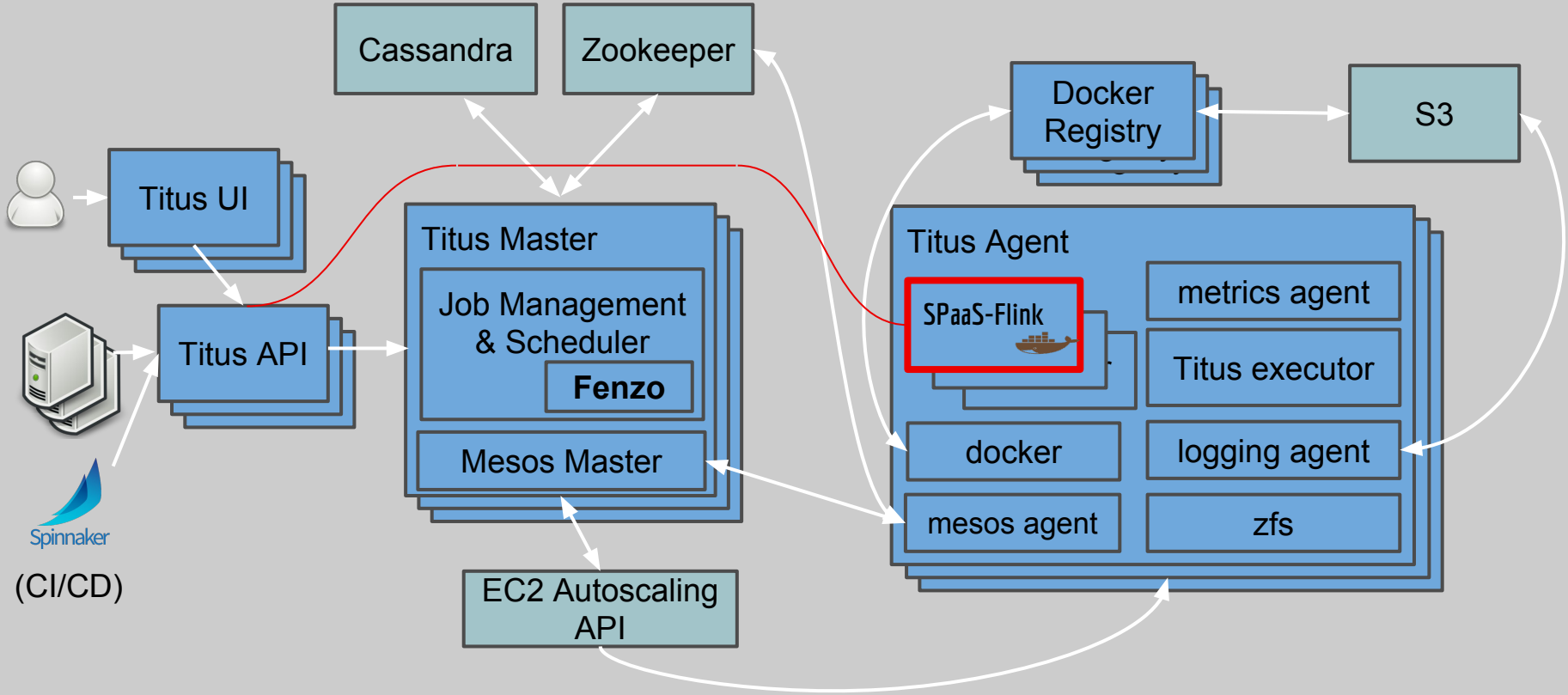
# Router - Massively parallel use case



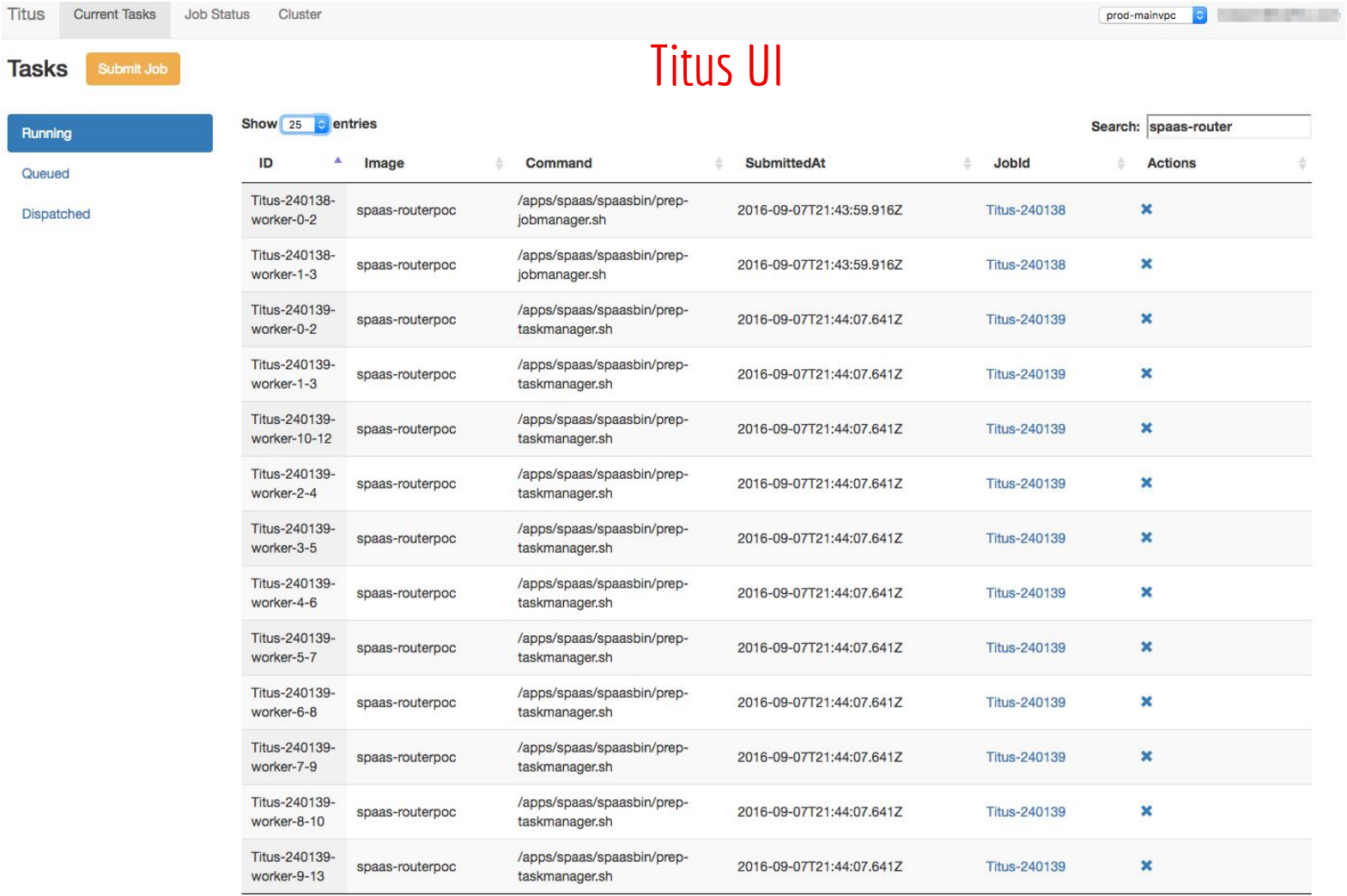
# Flink (1.2) Program Deployment (prod shadow)

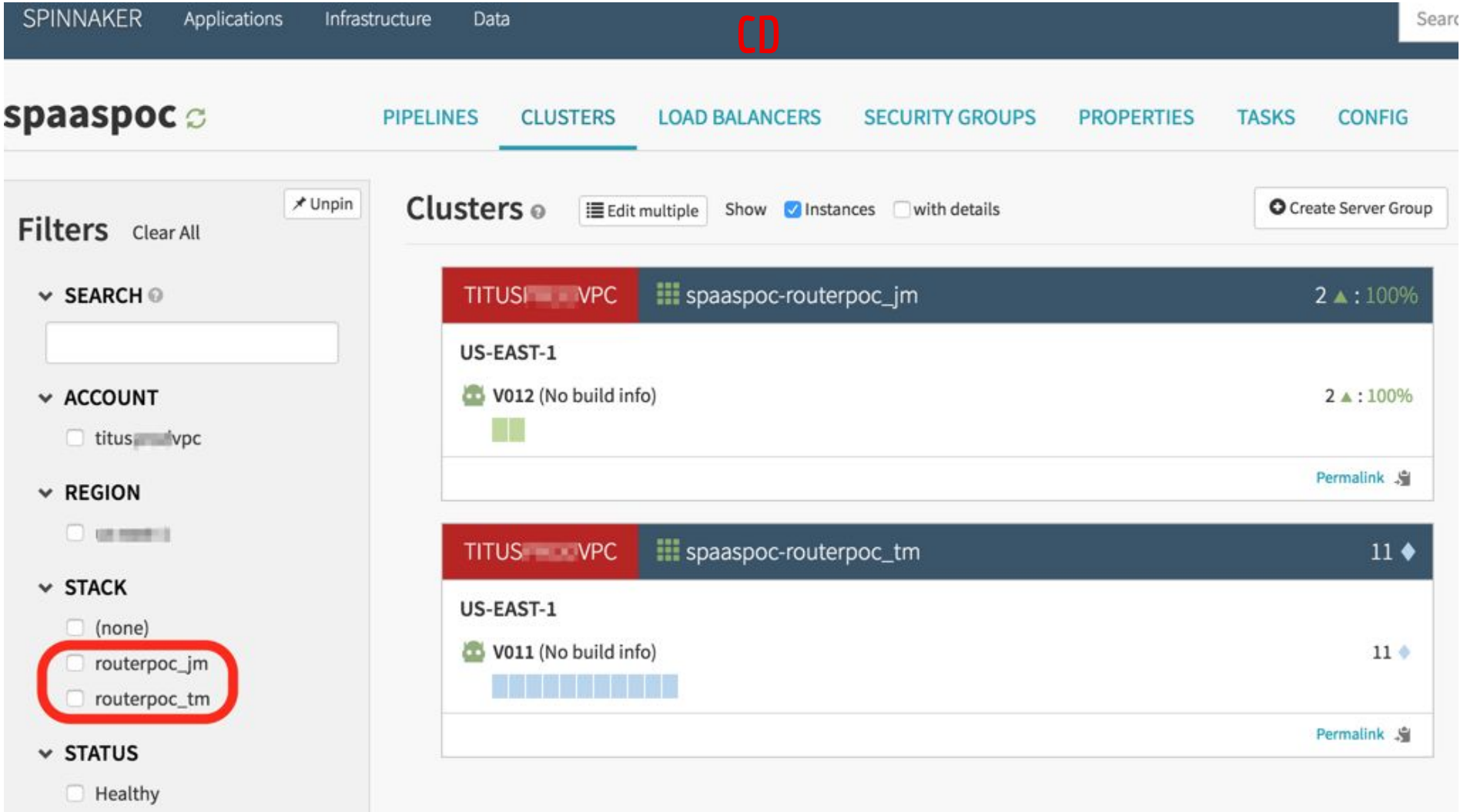


# Titus High Level Architecture

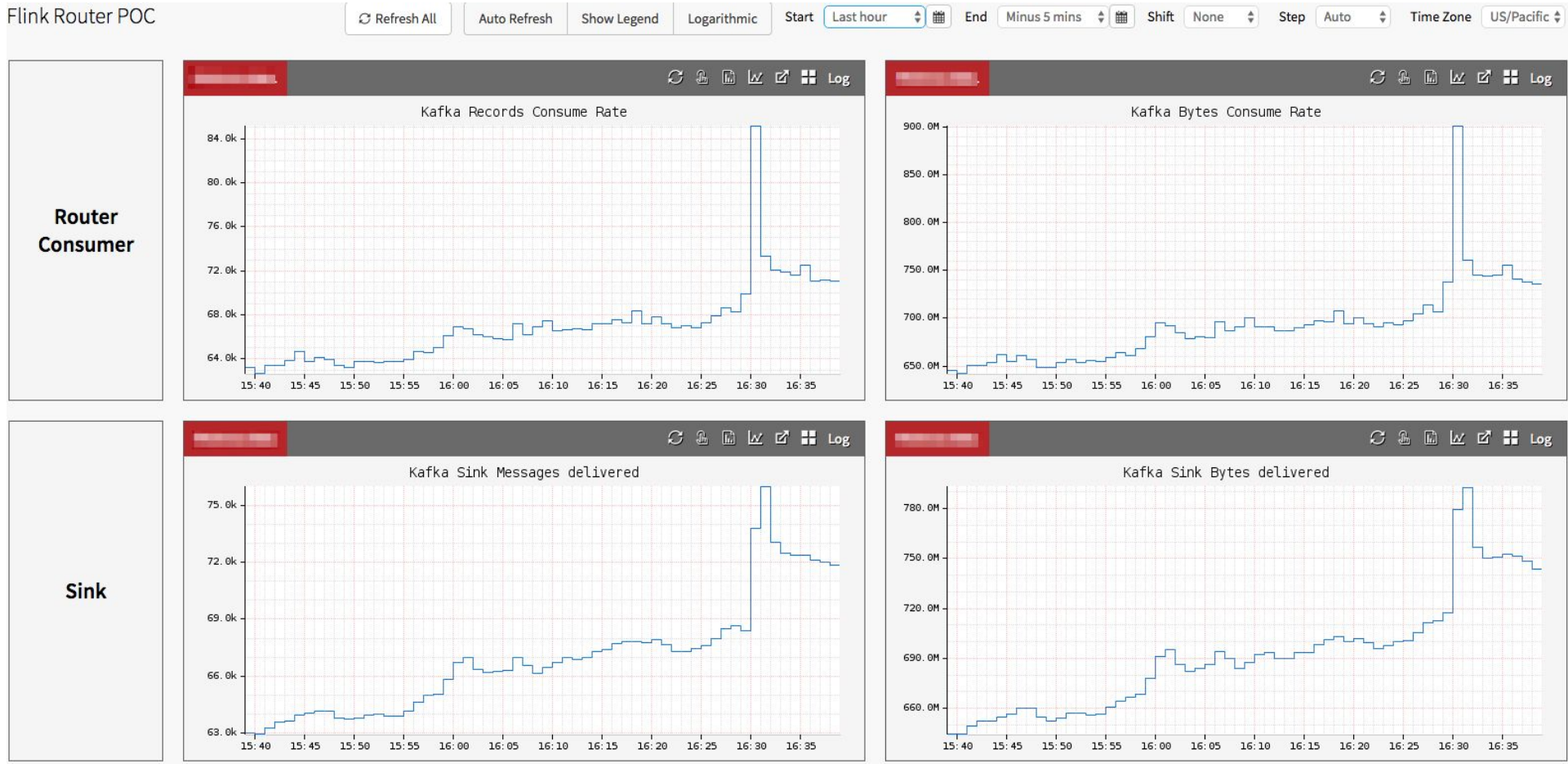


# Titus UI

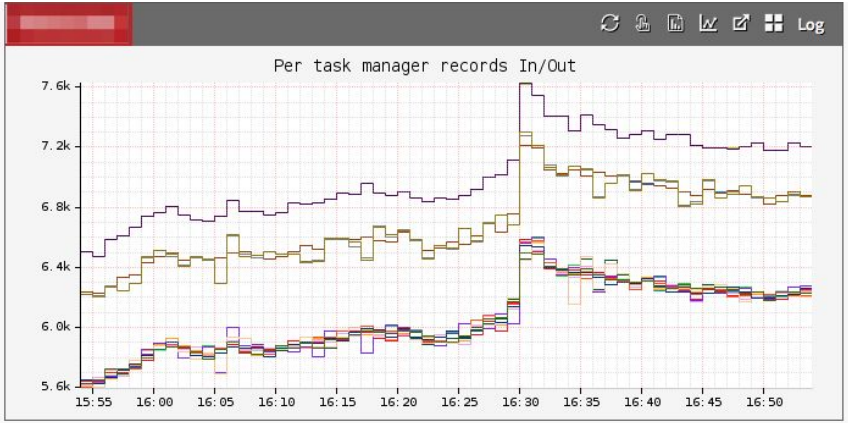

Titus
Current Tasks
Job Status
Cluster
prod-mainvpc
Tasks
Submit Job
Running
Queued
Dispatched
Show 25 entries
Search: spaas-router
ID
Image
Command
SubmittedAt
JobId
Actions
Titus-240138-worker-0-2
spaas-routerpoc
/apps/spaas/spaasbin/prep-jobmanager.sh
2016-09-07T21:43:59.916Z
Titus-240138
Titus-240138-worker-1-3
spaas-routerpoc
/apps/spaas/spaasbin/prep-jobmanager.sh
2016-09-07T21:43:59.916Z
Titus-240138
Titus-240139-worker-0-2
spaas-routerpoc
/apps/spaas/spaasbin/prep-taskmanager.sh
2016-09-07T21:44:07.641Z
Titus-240139
Titus-240139-worker-1-3
spaas-routerpoc
/apps/spaas/spaasbin/prep-taskmanager.sh
2016-09-07T21:44:07.641Z
Titus-240139
Titus-240139-worker-10-12
spaas-routerpoc
/apps/spaas/spaasbin/prep-taskmanager.sh
2016-09-07T21:44:07.641Z
Titus-240139
Titus-240139-worker-2-4
spaas-routerpoc
/apps/spaas/spaasbin/prep-taskmanager.sh
2016-09-07T21:44:07.641Z
Titus-240139
Titus-240139-worker-3-5
spaas-routerpoc
/apps/spaas/spaasbin/prep-taskmanager.sh
2016-09-07T21:44:07.641Z
Titus-240139
Titus-240139-worker-4-6
spaas-routerpoc
/apps/spaas/spaasbin/prep-taskmanager.sh
2016-09-07T21:44:07.641Z
Titus-240139
Titus-240139-worker-5-7
spaas-routerpoc
/apps/spaas/spaasbin/prep-taskmanager.sh
2016-09-07T21:44:07.641Z
Titus-240139
Titus-240139-worker-6-8
spaas-routerpoc
/apps/spaas/spaasbin/prep-taskmanager.sh
2016-09-07T21:44:07.641Z
Titus-240139
Titus-240139-worker-7-9
spaas-routerpoc
/apps/spaas/spaasbin/prep-taskmanager.sh
2016-09-07T21:44:07.641Z
Titus-240139
Titus-240139-worker-8-10
spaas-routerpoc
/apps/spaas/spaasbin/prep-taskmanager.sh
2016-09-07T21:44:07.641Z
Titus-240139
Titus-240139-worker-9-13
spaas-routerpoc
/apps/spaas/spaasbin/prep-taskmanager.sh
2016-09-07T21:44:07.641Z
Titus-240139

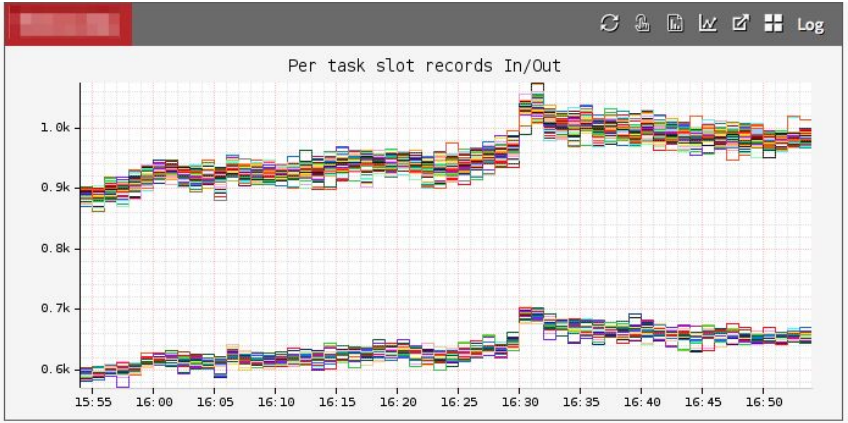

SPINNAKER
Applications
Infrastructure
Data
CD
spaaspoc
PIPELINES
CLUSTERS
LOAD BALANCERS
SECURITY GROUPS
PROPERTIES
TASKS
CONFIG
Unpin
Filters
Clear All
SEARCH
ACCOUNT
titus vpc
REGION
STACK
(none)
routerpoc_jm
routerpoc_tm
STATUS
Healthy
Clusters
Edit multiple
Show
Instances
with details
Create Server Group
TITUSI VPC
spaaspoc-routerpoc_jm
2 ▲ : 100%
US-EAST-1
V012 (No build info)
2 ▲ : 100%
Permalink
TITUS VPC
spaaspoc-routerpoc_tm
11
US-EAST-1
V011 (No build info)
11
Permalink

# Dashboard

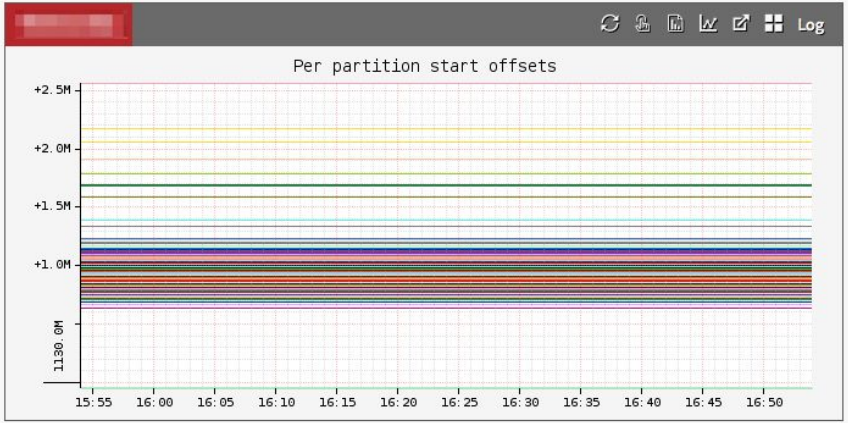

Flink Router POC
Refresh All
Auto Refresh
Show Legend
Logarithmic
Start
Last hour
End
Minus 5 mins
Shift
None
Step
Auto
Time Zone
US/Pacific
Router Consumer
Kafka Records Consume Rate
84.0k
80.0k
76.0k
72.0k
68.0k
64.0k
15:40
15:45
15:50
15:55
16:00
16:05
16:10
16:15
16:20
16:25
16:30
16:35
Kafka Bytes Consume Rate
900.0M
850.0M
800.0M
750.0M
700.0M
650.0M
Sink
Kafka Sink Messages delivered
75.0k
72.0k
69.0k
66.0k
63.0k
Kafka Sink Bytes delivered
780.0M
750.0M
720.0M
690.0M
660.0M

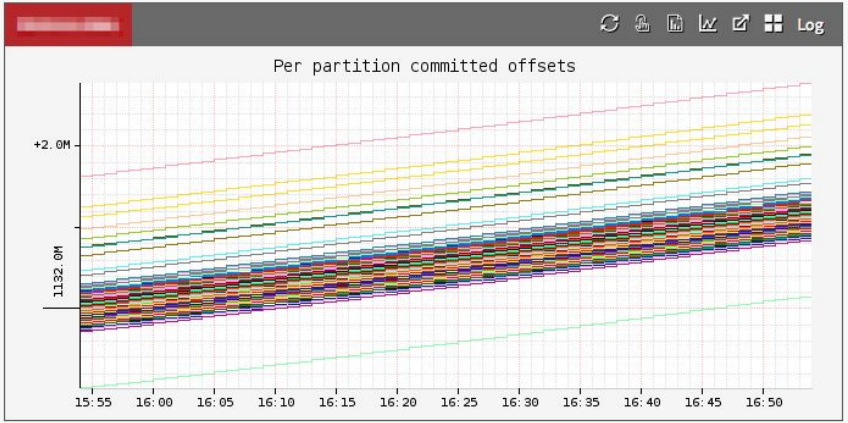
**Records In/Out**



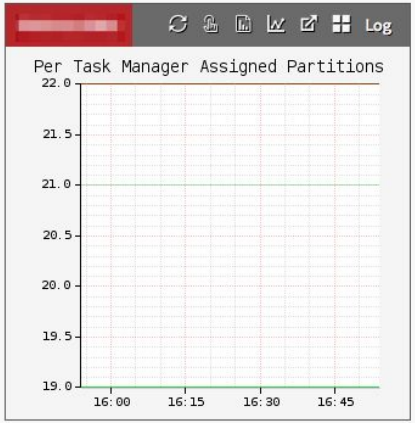



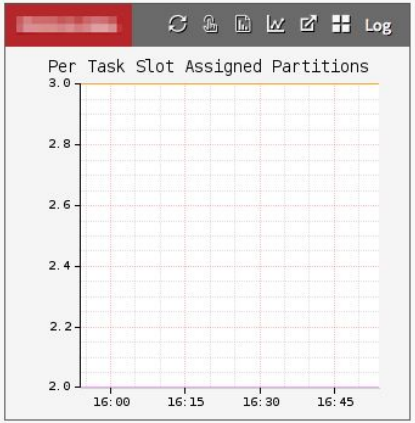

**Kafka Offsets**

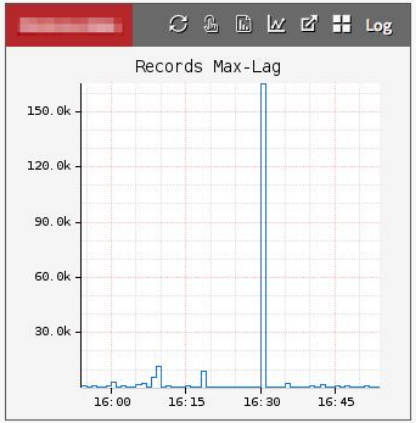


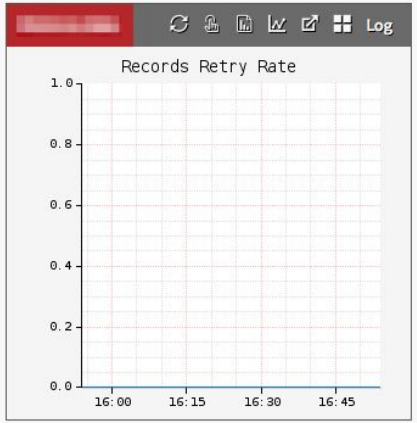



**Consumer Internals**

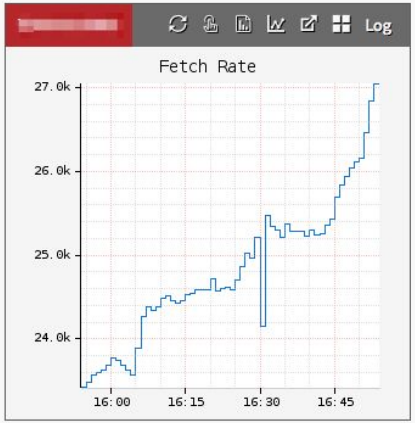


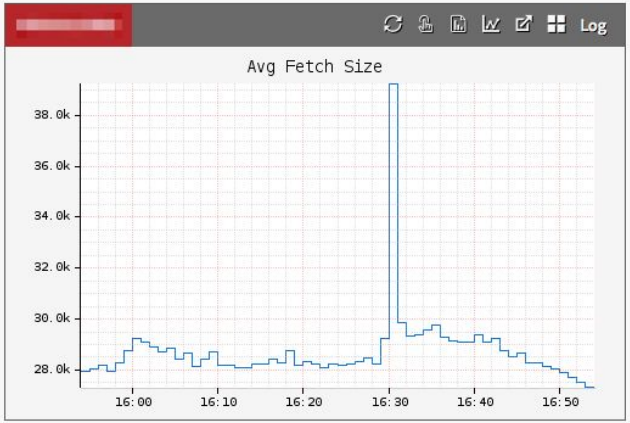



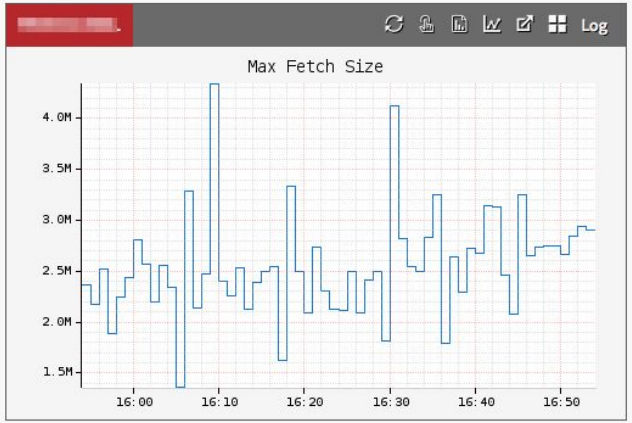



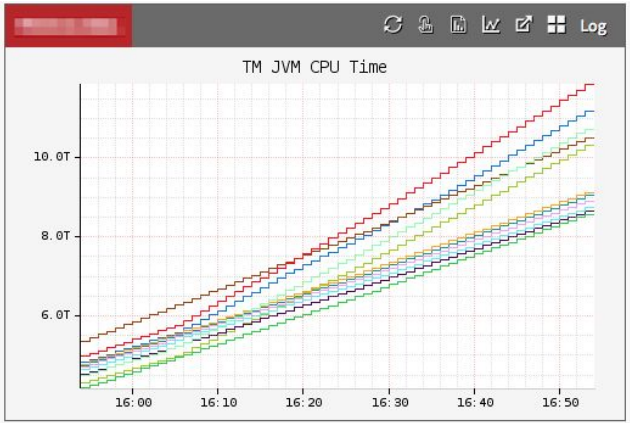

**Consumer Internals**

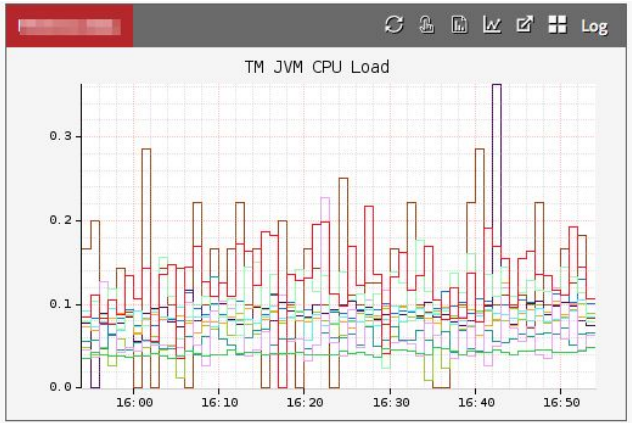


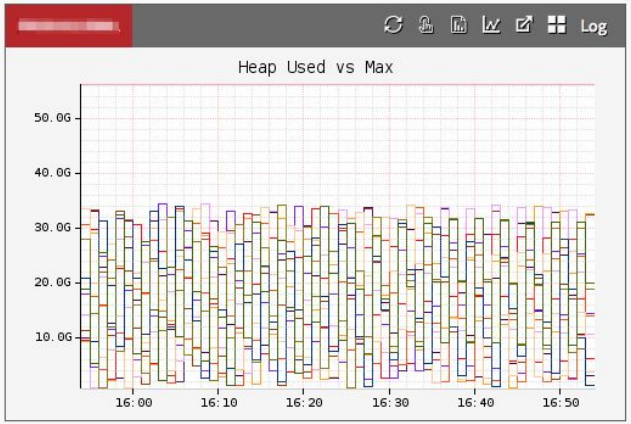



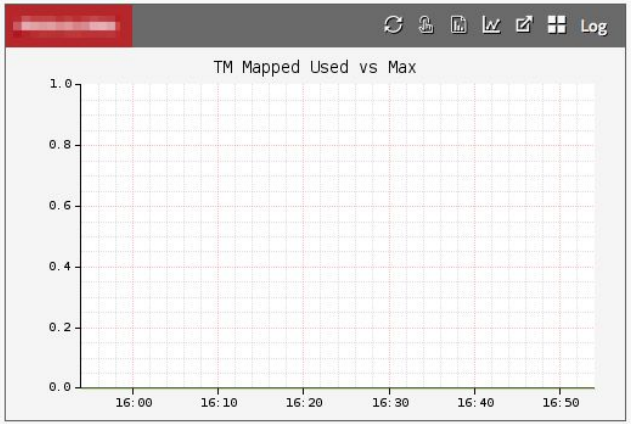
**Task Managers JVM**

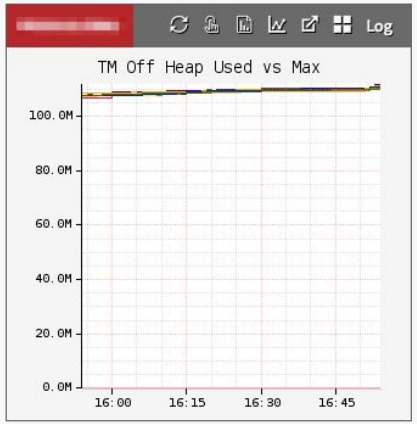



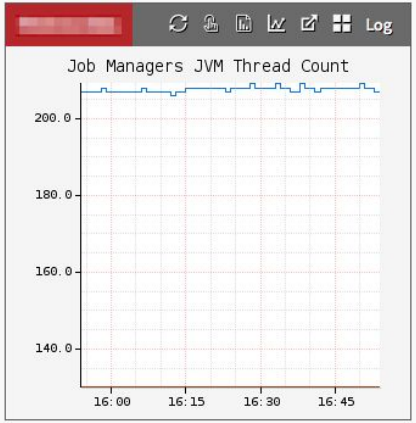

**Task Managers Memory**

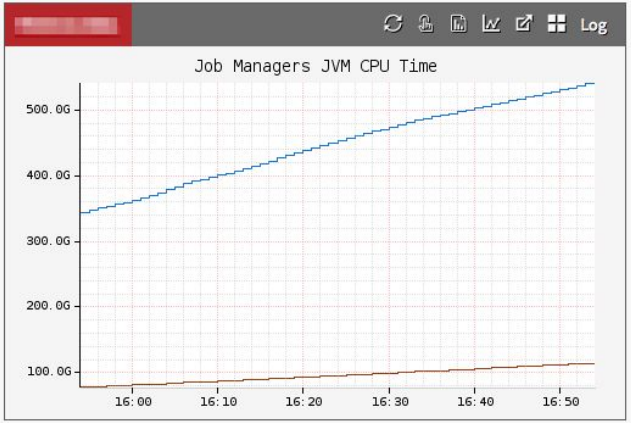


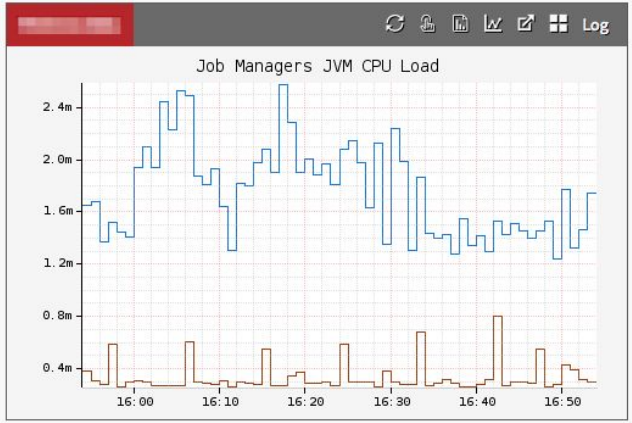



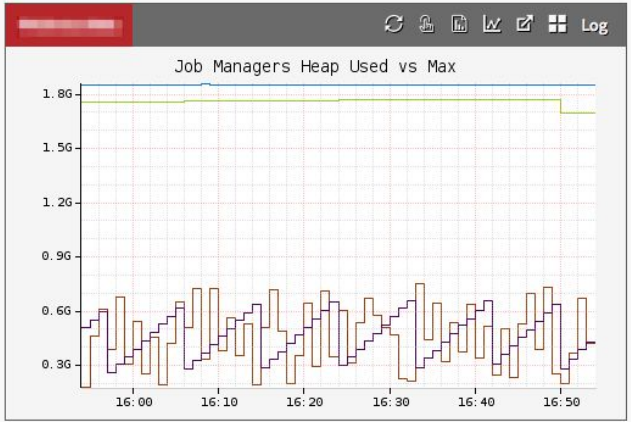
**Job Managers JVM**

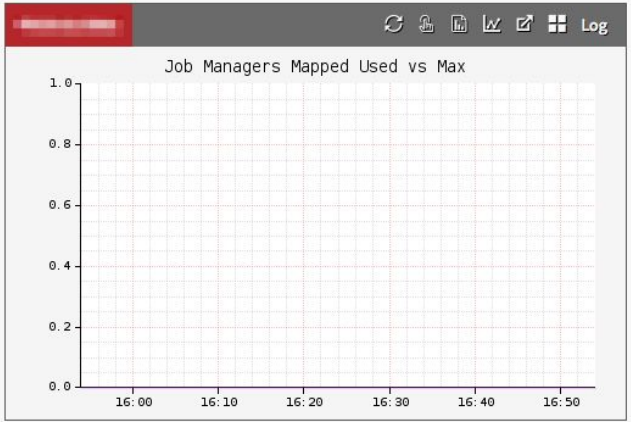


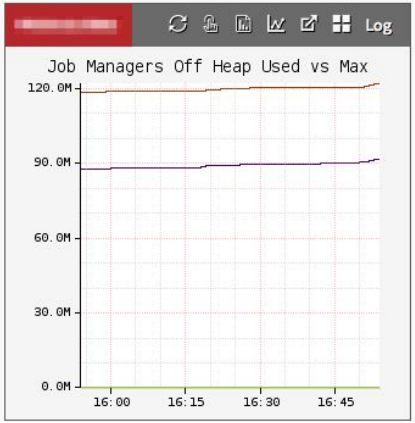



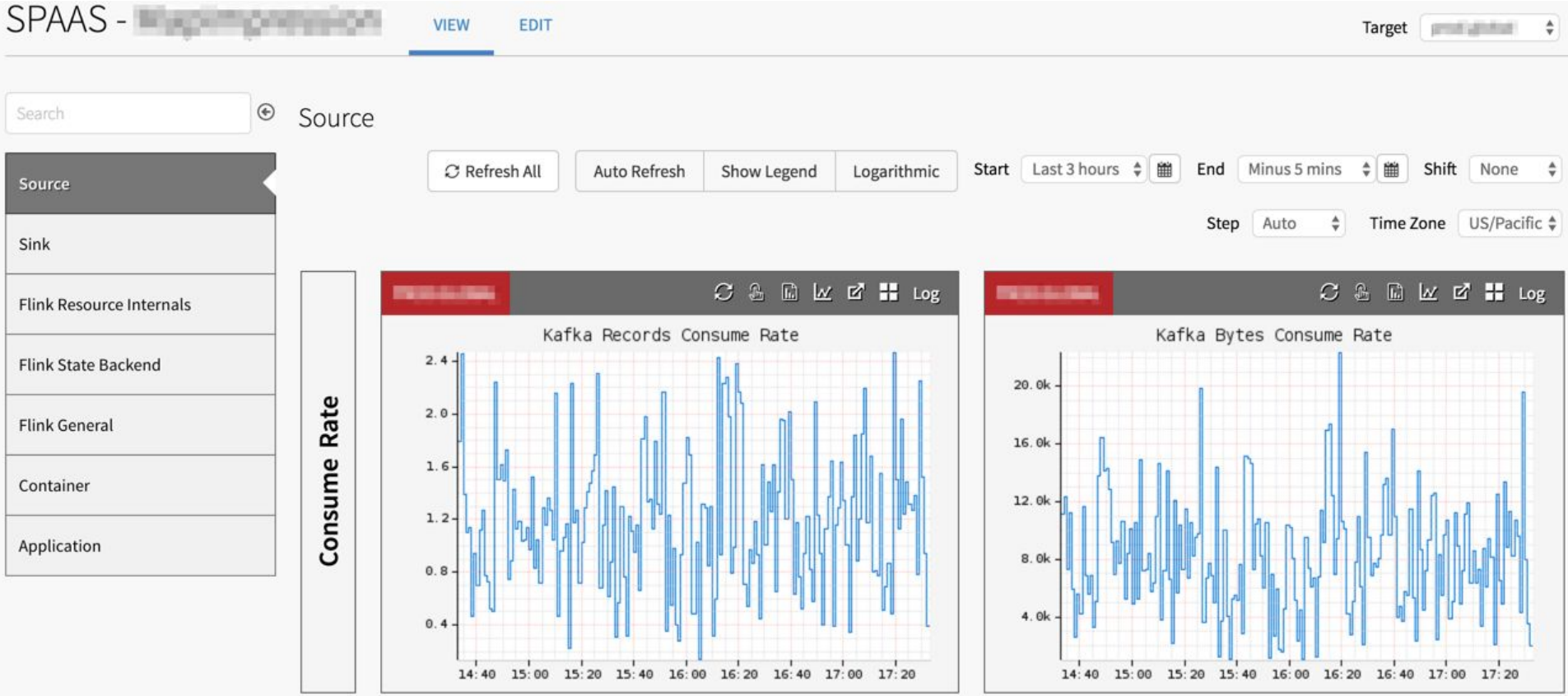

**Job Managers Memory**



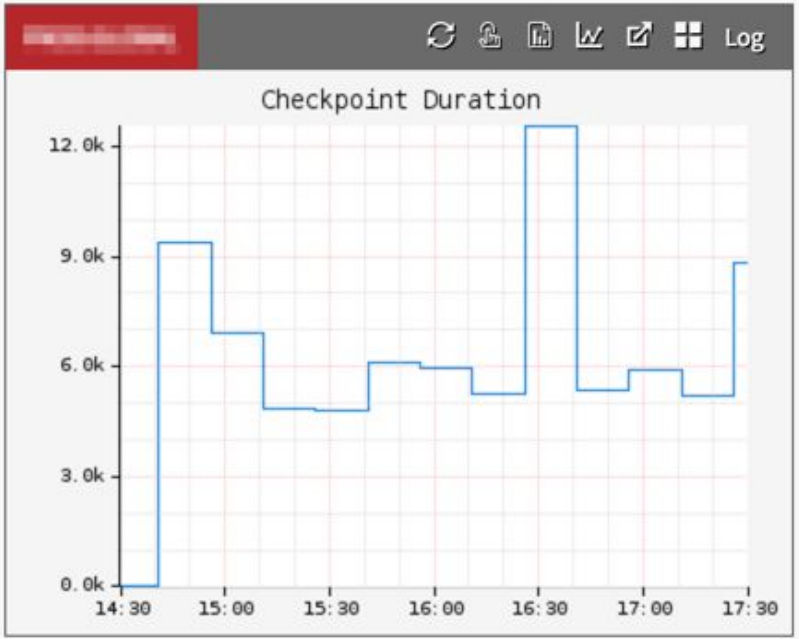



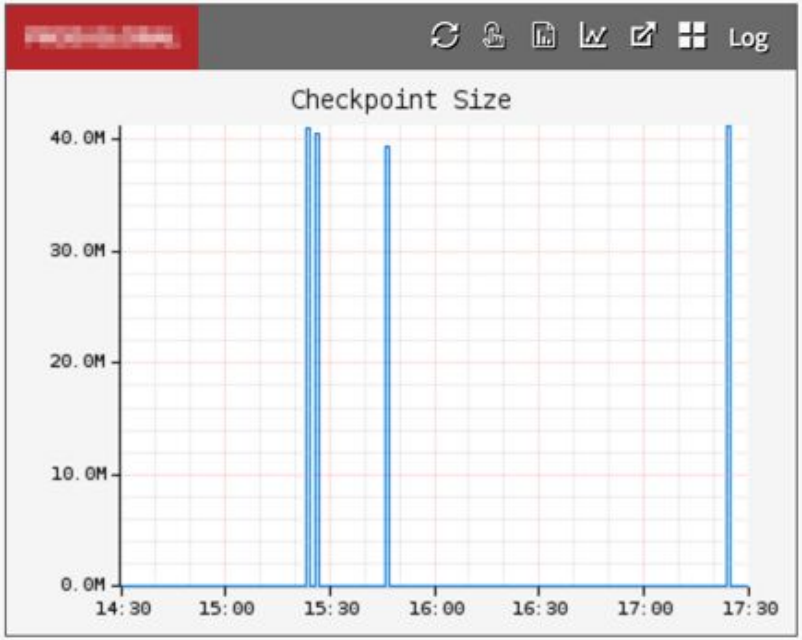
# Keystone SPaaS-Flink Pilot Use Case - 2



# Keystone SPaaS-Flink Pilot Use Case - 2

**Checkpoints**

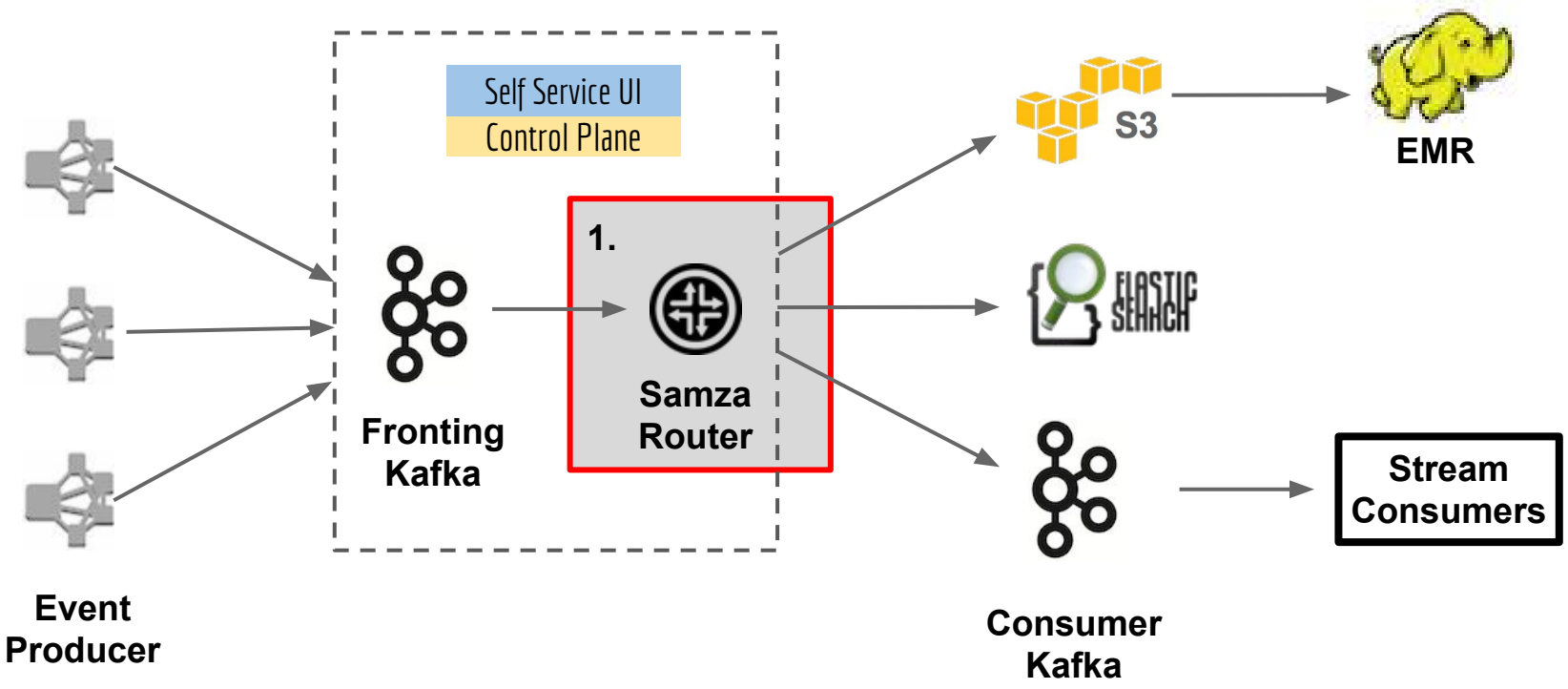


# Flink Router perf test (YMMV)

- **Note**
  - The tests were performed on a specific use case,
  - running in a specific environment, and with
  - with **one specific event stream**, and setup.

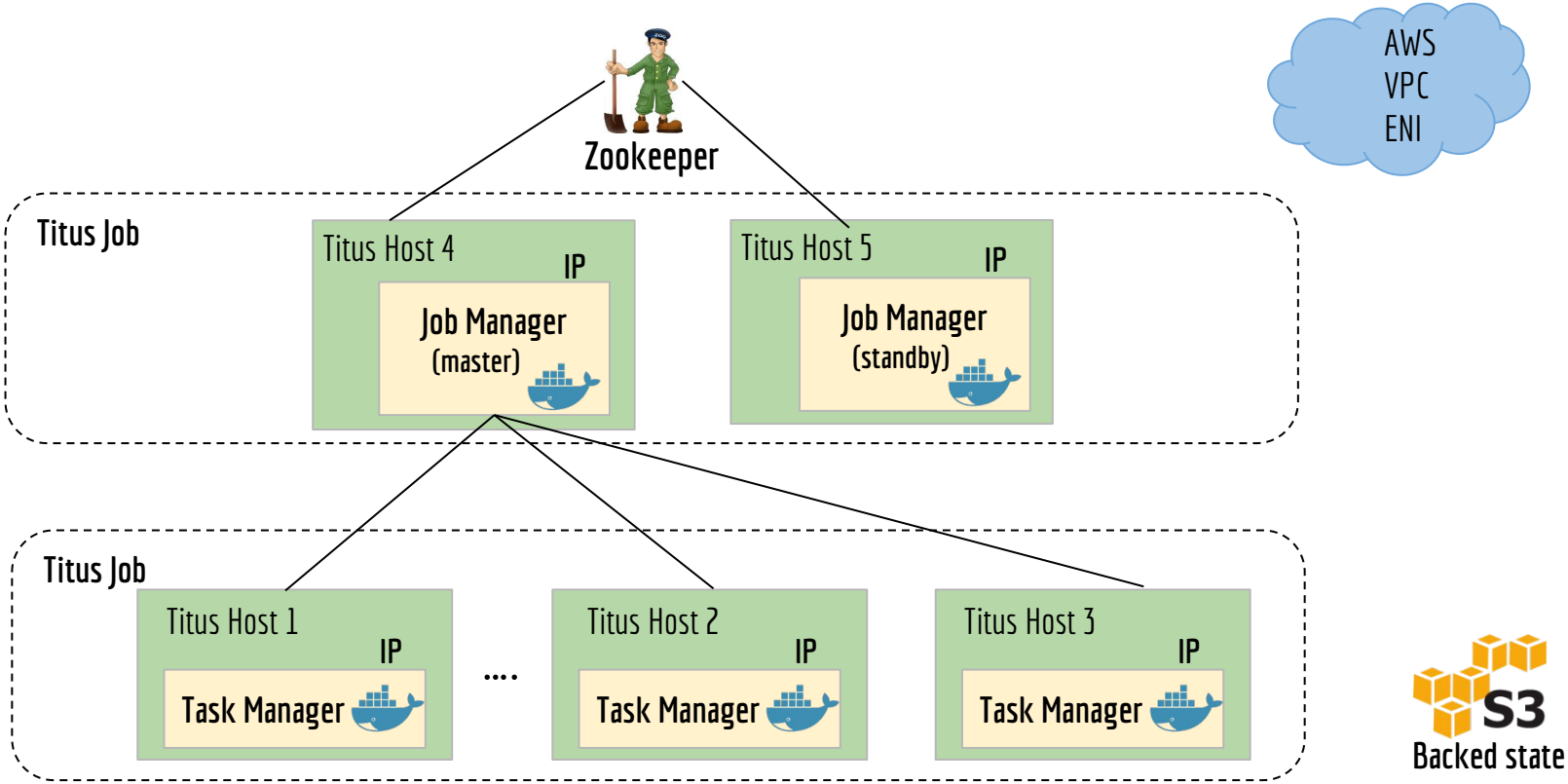
# Keystone


Self Service UI
Control Plane
1.
Samza Router
Fronting Kafka
S3
EMR
ELASTIC SEARCH
Stream Consumers
Consumer Kafka
Event Producer

# Details..

- Different runtimes for Flink & Samza routers
- Massively parallel use-case
  - Per element processing
- Focused on net outcomes

# Flink (1.2) Router

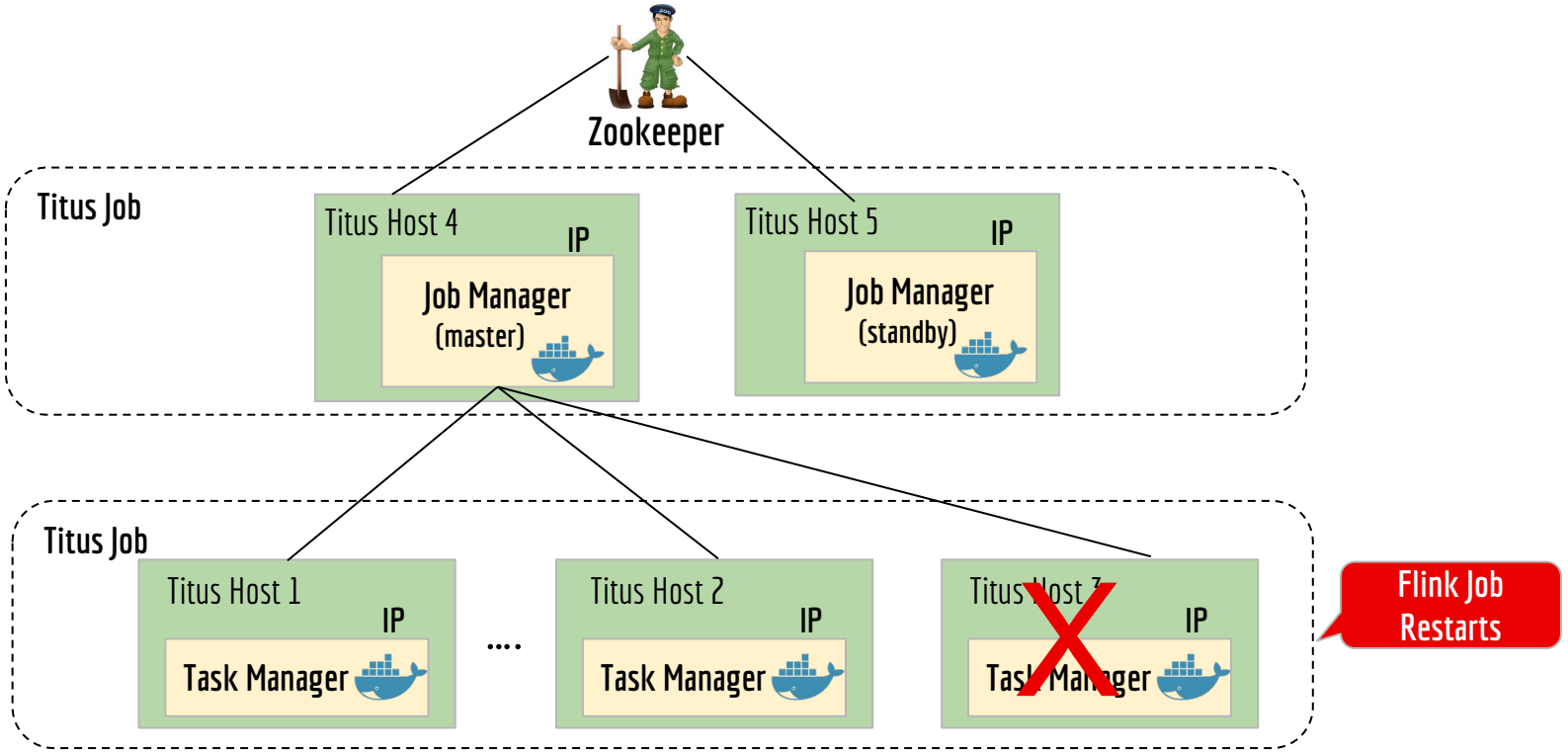


# Flink Router outcome (YMMV)

- Cost ≅17% **savings**
- Memory utilization ≅16% **better**
- Cpu utilization ≅ 40% **better**
- Network utilization ≅ 10% **better**
- Msg. throughput ≅ 1% (avg) - 4% (peak) **better**

# Awaiting Flink Features

# Fine Grained Recovery FLIP-1



# Awaiting Flink Features (now)

- Checkpoints and savepoints
  - FLINK-4484 - Unify, Persistent checkpoints, periodic savepoints
  - Compatible across Flink version upgrades
  - Inspection tool for debugging

# Awaiting Flink Features (now)

- Atomic savepoint and stop (pause) the job
- Dynamic Scaling DONE
  - Resume from savepoint with different parallelism
  - Job elasticity
- FLINK-4545 Adjust TaskManager network buffer when scaling

# Awaiting Flink Features (now)

- Cluster Runtime and Elasticity (FLIP - 6)
- Extending Window Function Metadata (FLIP-2)
- Large State support
  - Incremental checkpointing
  - hot standby

# Awaiting Flink Features

- Metric tags as key-value pairs - FLINK-4245
- FLIP-9 Window Trigger DSL DONE
- FLIP-11 Table API
- Side Inputs / Side Outputs (handle late data)

# Ponder over

Could Stream Processing Engine enable building non-analytics Applications?

# More brain food...

- Netflix Keystone Pipeline Evolution
- Netflix Kafka in Keystone Pipeline
- Samza Meetup Presentation
- Titus talk
- Netflix OSS