Infrastructure testing using Kubernetes

@rantav

OR:

Testing **Kafka replication** over the Atlantic using **Golang, Kubernetes** and friends

Hello!

I am Ran Tavory

I work on **multi-region at appsflyer**

You can find me **@**rantav (twitter/gmail/facebook)

What is AppsFlyer?

Data collection and analytics system

The Goal: Test Kafka Replication

- Replicate 100MB per second
- Validate **correctness**
- Measure **latency**
- Run **test cases** (failure scenarios)
  - Broker crash, cluster resize, packet loss…

Does this remind anyone anything?

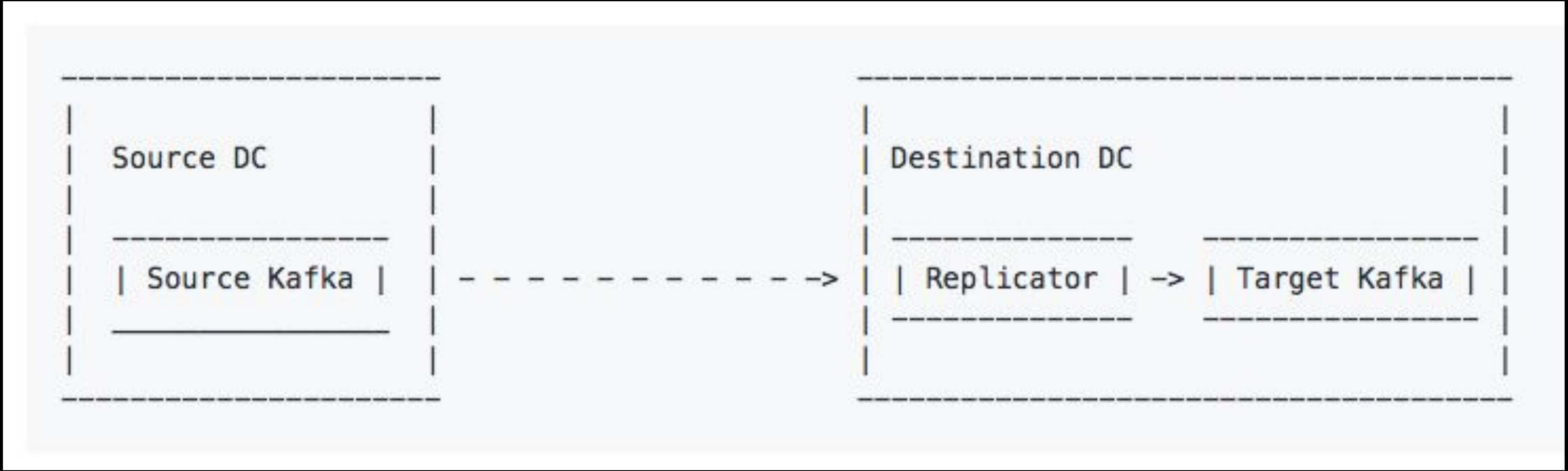
- Run test cases (failure scenarios)
  - Broker crash
  - cluster resize
  - packet loss
  - etc...

“ Jepsen: Call me maybe

// Kyle Kingsbury

HOW ?

High level design



```
 ----------------------                                 --------------------------------------
|                     |                                |                                      |
|  Source DC          |                                | Destination DC                       |
|                     |                                |                                      |
|  ----------------   |                                |  --------------    ----------------  |
|  | Source Kafka |   | - - - - - - - - - - ->         |  | Replicator | -> | Target Kafka |  |
|  ________________   |                                |  --------------    ----------------  |
|                     |                                |                                      |
 ----------------------                                 --------------------------------------
```

Kafka Replication

From https://github.com/AppsFlyer/kafka-mirror-tester

```
----------------------                        --------------------------------------
|                    |                        |                                    |
|  Source DC         |                        | Destination DC                     |
|                    |                        |                                    |
|  ----------------  |                        | --------------    ---------------- |
|  | Source Kafka |  | - - - - - - - - - - -> | | Replicator | -> | Target Kafka | |
|  ----------------  |                        | --------------    ---------------- |
|    ↑               |                        |                            |       |
|    |               |                        |                            |       |
|    |               |                        |                            ↓       |
|  ------------      |                        |                        ------------|
|  | producer |      |                        |                        | consumer ||
|  ------------      |                        |                        ------------|
----------------------                        --------------------------------------
```

With a home-grown **producer** and **consumer**
written in Golang

WHAT DO WE WANT?

DOCUMENTATION

HOW DO WE DO IT?

With Code

WHAT DO WE WANT?

REPRODUCIBILITY

HOW DO WE DO IT?

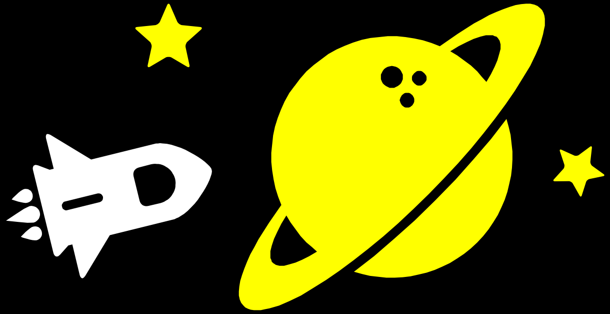
With Code

WHAT DO WE WANT?

MONITORING

HOW DO WE DO IT?

With Code



KUBERNETES

A brief introduction to **Kubernetes**

Kubernetes is a

CLUSTER MANAGER

It manages

NODES

The **host** running a kubelet and a set of Pods

Nodes contain

PODS

Pods consist of 1 or more **containers**

Pods run

DEPLOYMENTS

A set of stateless pods

Pods also run

STATEFUL SET

A set of **stateful** pods

You control k8s using

KUBECTL

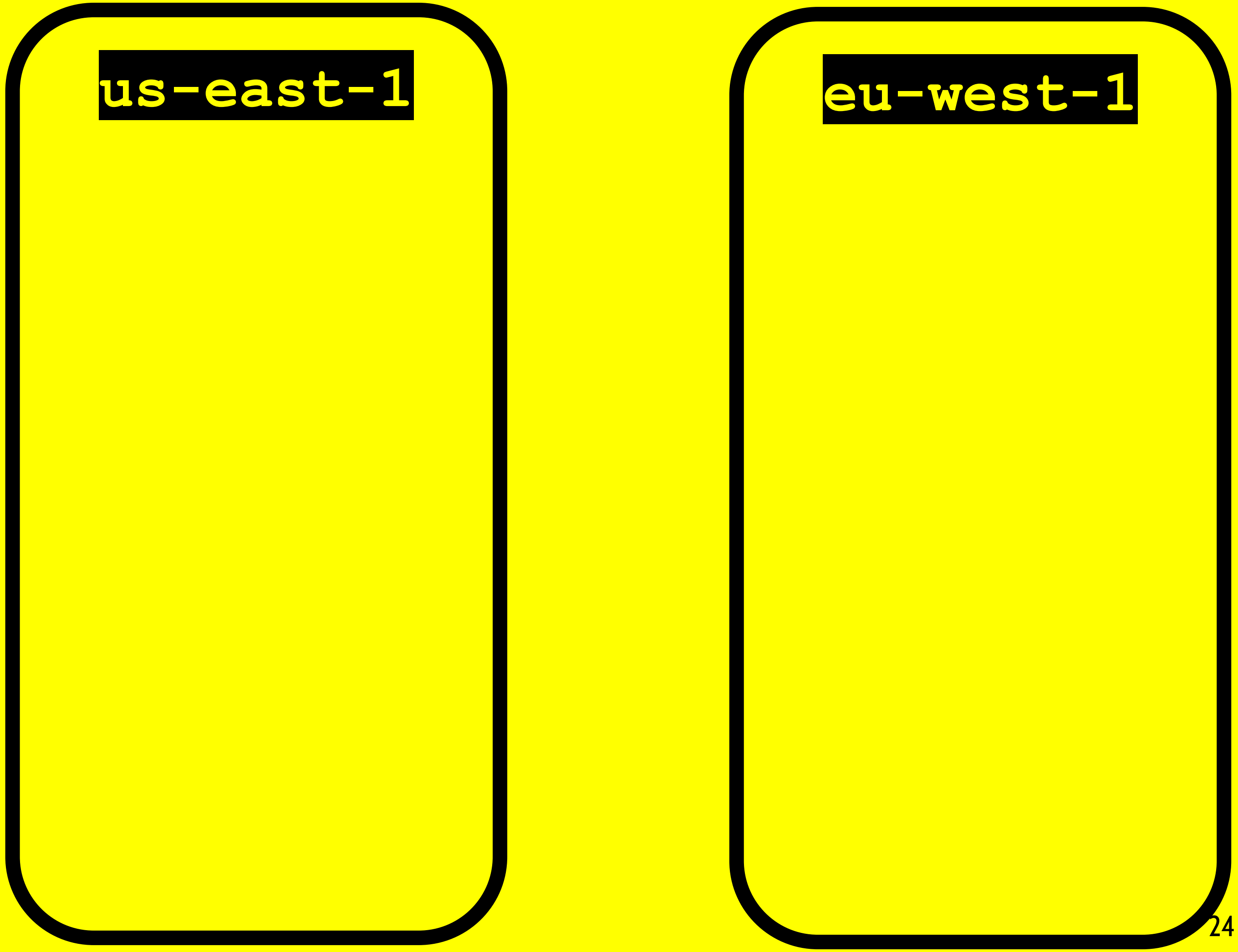
The k8s CLI

```
$ make k8s-all
```

What does that do?

- Provision VMs in AWS
- In two regions. Ireland and Virginia.
- Setup VPCs, Subnets, Routing Tables
- Create Security Groups
- Setup Load Balancers
- Install Kubernetes (etcd, masters, nodes)
- Setup Monitoring (Prometheus & Grafana and install dashboard)
- Install Weave Scope
- Install Kafkas (and test them)
- Install **uReplicator / Brooklin** (and test them)
- Install test programs (written by my in Go)
- And more… (ASGs, DHCP etc)

What are we building?



What are we building?

`us-east-1`

40 nodes
k8s cluster

`eu-west-1`

48 nodes
k8s cluster

What are we building?

`us-east-1`

40 nodes
k8s cluster

30 brokers
kafka cluster

`eu-west-1`

48 nodes
k8s cluster

30 brokers
kafka cluster

What are we building?

`us-east-1`

40 nodes
k8s cluster

30 brokers
kafka cluster

`eu-west-1`

48 nodes
k8s cluster

30 brokers
kafka cluster

8 workers
uReplicator
/
32 Brooklin

What are we building?

`us-east-1`

40 nodes
k8s cluster

30 brokers
kafka cluster

10 producers

`eu-west-1`

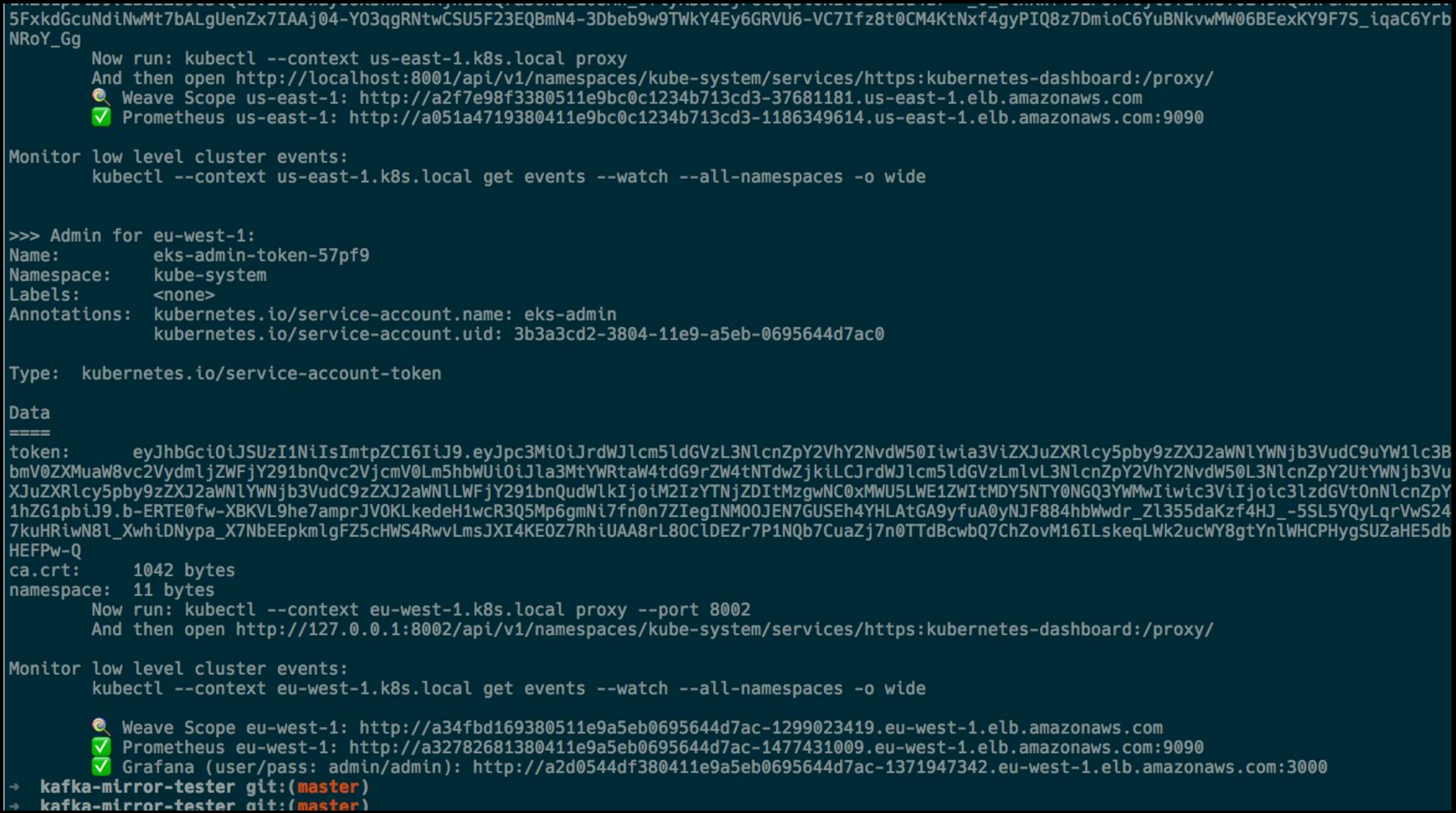
48 nodes
k8s cluster

30 brokers
kafka cluster

uReplicator /
Brooklin

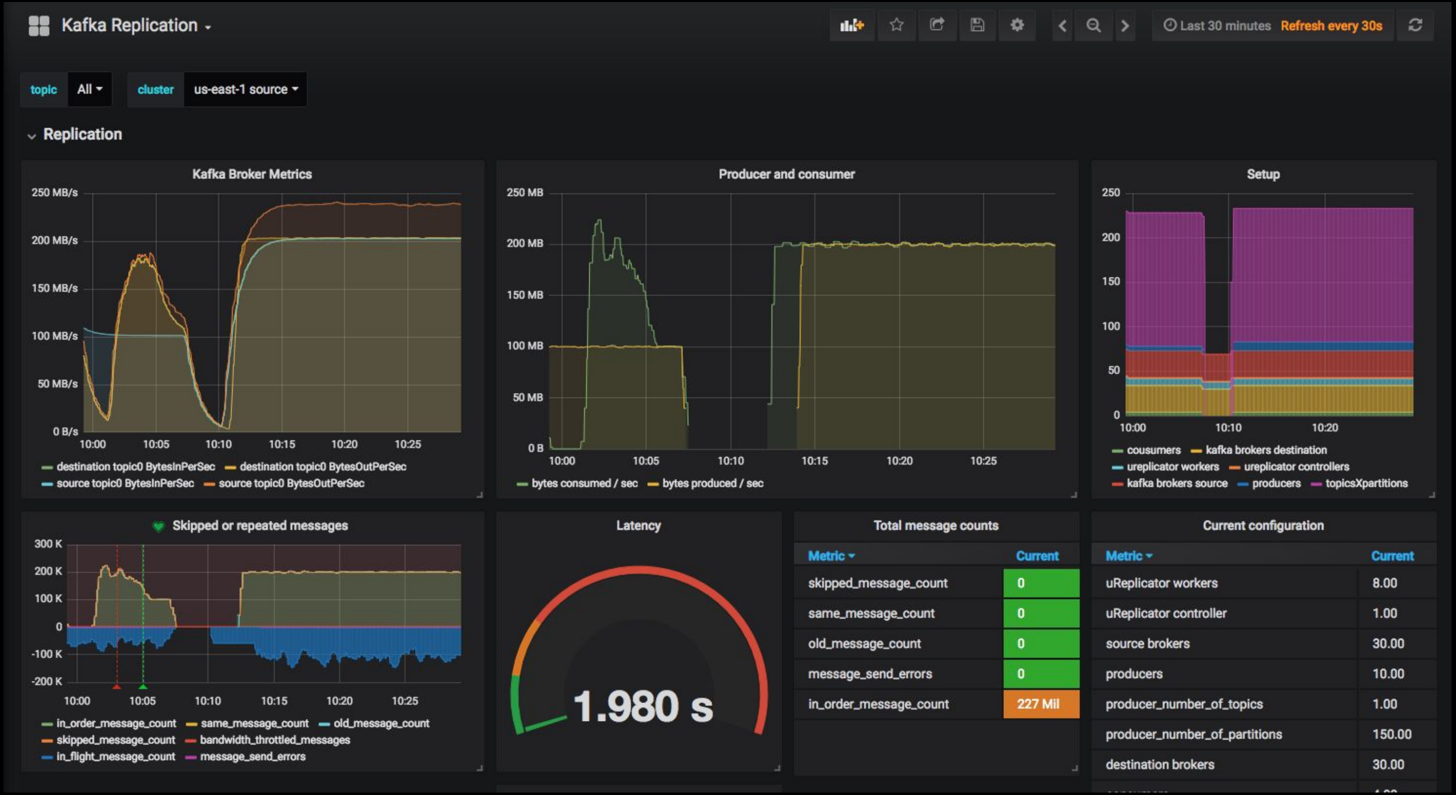
4 consumers

End Result



```
5FxkdGcuNdiNwMt7bALgUenZx7IAAj04-YO3qgRNtwCSU5F23EQBmN4-3Dbeb9w9TWkY4Ey6GRVU6-VC7Ifz8t0CM4KtNxf4gyPIQ8z7DmioC6YuBNkvwMW06BEexKY9F7S_iqaC6Yrb
NRoY_Gg
        Now run: kubectl --context us-east-1.k8s.local proxy
        And then open http://localhost:8001/api/v1/namespaces/kube-system/services/https:kubernetes-dashboard:/proxy/
        🍭 Weave Scope us-east-1: http://a2f7e98f3380511e9bc0c1234b713cd3-37681181.us-east-1.elb.amazonaws.com
        ✅ Prometheus us-east-1: http://a051a4719380411e9bc0c1234b713cd3-1186349614.us-east-1.elb.amazonaws.com:9090

Monitor low level cluster events:
        kubectl --context us-east-1.k8s.local get events --watch --all-namespaces -o wide


>>> Admin for eu-west-1:
Name:         eks-admin-token-57pf9
Namespace:    kube-system
Labels:       <none>
Annotations:  kubernetes.io/service-account.name: eks-admin
              kubernetes.io/service-account.uid: 3b3a3cd2-3804-11e9-a5eb-0695644d7ac0

Type:  kubernetes.io/service-account-token

Data
====
token:      eyJhbGciOiJSUzI1NiIsImtpZCI6IiJ9.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2NvdW50Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9uYW1lc3B
bmV0ZXMuaW8vc2VydmljZWFjY291bnQvc2VjcmV0Lm5hbWUiOiJla3MtYWRtaW4tdG9rZW4tNTdwZjkiLCJrdWJlcm5ldGVzLmlvL3NlcnZpY2VhY2NvdW50L3NlcnZpY2UtYWNjb3Vu
XJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9zZXJ2aWNlLWFjY291bnQudWlkIjoiM2IzYTNjZDItMzgwNC0xMWU5LWE1ZWItMDY5NTY0NGQ3YWMwIiwic3ViIjoic3lzdGVtOnNlcnZpY
1hZG1pbiJ9.b-ERTE0fw-XBKVL9he7amprJVOKLkedeH1wcR3Q5Mp6gmNi7fn0n7ZIegINMOOJEN7GUSEh4YHLAtGA9yfuA0yNJF884hbWwdr_Zl355daKzf4HJ_-5SL5YQyLqrVwS24
7kuHRiwN8l_XwhiDNypa_X7NbEEpkmlgFZ5cHWS4RwvLmsJXI4KEOZ7RhiUAA8rL8OClDEZr7P1NQb7CuaZj7n0TTdBcwbQ7ChZovM16ILskeqLWk2ucWY8gtYnlWHCPHygSUZaHE5db
HEFPw-Q
ca.crt:     1042 bytes
namespace:  11 bytes
        Now run: kubectl --context eu-west-1.k8s.local proxy --port 8002
        And then open http://127.0.0.1:8002/api/v1/namespaces/kube-system/services/https:kubernetes-dashboard:/proxy/

Monitor low level cluster events:
        kubectl --context eu-west-1.k8s.local get events --watch --all-namespaces -o wide

        🍭 Weave Scope eu-west-1: http://a34fbd169380511e9a5eb0695644d7ac-1299023419.eu-west-1.elb.amazonaws.com
        ✅ Prometheus eu-west-1: http://a32782681380411e9a5eb0695644d7ac-1477431009.eu-west-1.elb.amazonaws.com:9090
        ✅ Grafana (user/pass: admin/admin): http://a2d0544df380411e9a5eb0695644d7ac-1371947342.eu-west-1.elb.amazonaws.com:3000
➜  kafka-mirror-tester git:(master)
➜  kafka-mirror-tester git:(master)
```

Grafana Dashboard



topic: All cluster: us-east-1 source

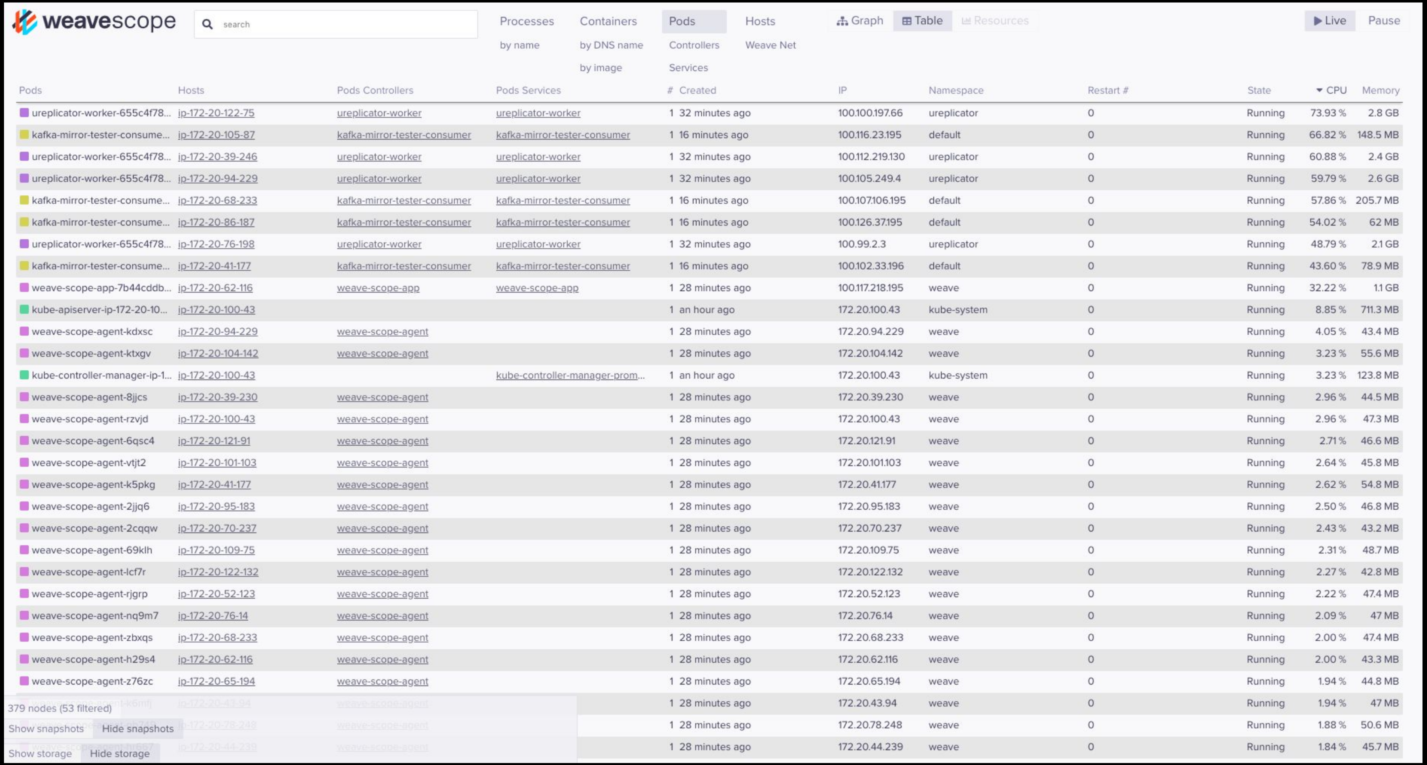
Total message counts

Metric	Current
skipped_message_count	0
same_message_count	0
old_message_count	0
message_send_errors	0
in_order_message_count	227 Mil

Current configuration

Metric	Current
uReplicator workers	8.00
uReplicator controller	1.00
source brokers	30.00
producers	10.00
producer_number_of_topics	1.00
producer_number_of_partitions	150.00
destination brokers	30.00

Weave Scope



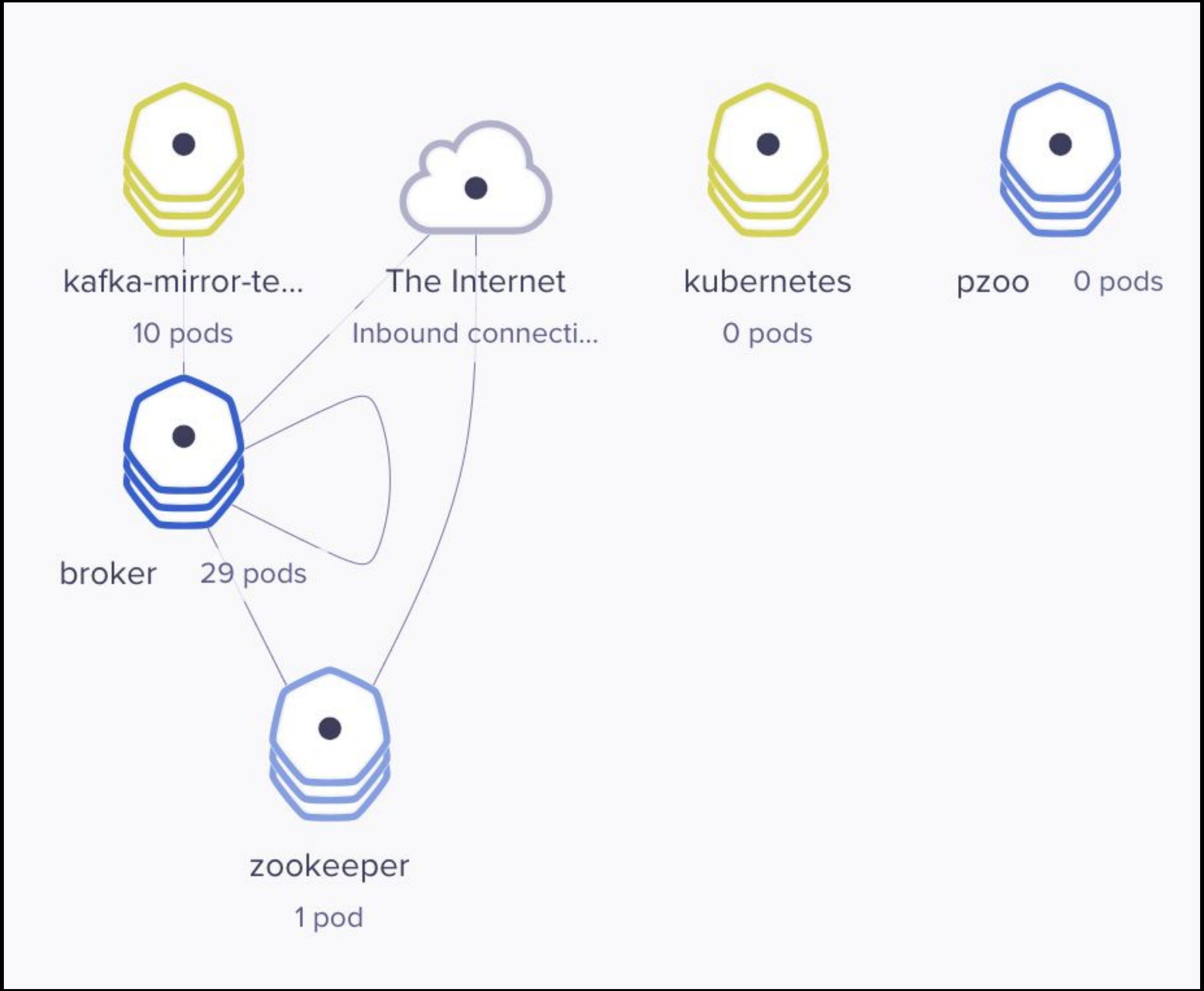
Pods	Hosts	Pods Controllers	Pods Services	#	Created	IP	Namespace	Restart #	State	▾ CPU	Memory
🟪 ureplicator-worker-655c4f78...	ip-172-20-122-75	ureplicator-worker	ureplicator-worker	1	32 minutes ago	100.100.197.66	ureplicator	0	Running	73.93 %	2.8 GB
🟨 kafka-mirror-tester-consume...	ip-172-20-105-87	kafka-mirror-tester-consumer	kafka-mirror-tester-consumer	1	16 minutes ago	100.116.23.195	default	0	Running	66.82 %	148.5 MB
🟪 ureplicator-worker-655c4f78...	ip-172-20-39-246	ureplicator-worker	ureplicator-worker	1	32 minutes ago	100.112.219.130	ureplicator	0	Running	60.88 %	2.4 GB
🟪 ureplicator-worker-655c4f78...	ip-172-20-94-229	ureplicator-worker	ureplicator-worker	1	32 minutes ago	100.105.249.4	ureplicator	0	Running	59.79 %	2.6 GB
🟨 kafka-mirror-tester-consume...	ip-172-20-68-233	kafka-mirror-tester-consumer	kafka-mirror-tester-consumer	1	16 minutes ago	100.107.106.195	default	0	Running	57.86 %	205.7 MB
🟨 kafka-mirror-tester-consume...	ip-172-20-86-187	kafka-mirror-tester-consumer	kafka-mirror-tester-consumer	1	16 minutes ago	100.126.37.195	default	0	Running	54.02 %	62 MB
🟪 ureplicator-worker-655c4f78...	ip-172-20-76-198	ureplicator-worker	ureplicator-worker	1	32 minutes ago	100.99.2.3	ureplicator	0	Running	48.79 %	2.1 GB
🟨 kafka-mirror-tester-consume...	ip-172-20-41-177	kafka-mirror-tester-consumer	kafka-mirror-tester-consumer	1	16 minutes ago	100.102.33.196	default	0	Running	43.60 %	78.9 MB
🟪 weave-scope-app-7b44cddb...	ip-172-20-62-116	weave-scope-app	weave-scope-app	1	28 minutes ago	100.117.218.195	weave	0	Running	32.22 %	1.1 GB
🟩 kube-apiserver-ip-172-20-10...	ip-172-20-100-43			1	an hour ago	172.20.100.43	kube-system	0	Running	8.85 %	711.3 MB
🟪 weave-scope-agent-kdxsc	ip-172-20-94-229	weave-scope-agent		1	28 minutes ago	172.20.94.229	weave	0	Running	4.05 %	43.4 MB
🟪 weave-scope-agent-ktxgv	ip-172-20-104-142	weave-scope-agent		1	28 minutes ago	172.20.104.142	weave	0	Running	3.23 %	55.6 MB
🟩 kube-controller-manager-ip-1...	ip-172-20-100-43		kube-controller-manager-prom...	1	an hour ago	172.20.100.43	kube-system	0	Running	3.23 %	123.8 MB
🟪 weave-scope-agent-8jjcs	ip-172-20-39-230	weave-scope-agent		1	28 minutes ago	172.20.39.230	weave	0	Running	2.96 %	44.5 MB
🟪 weave-scope-agent-rzvjd	ip-172-20-100-43	weave-scope-agent		1	28 minutes ago	172.20.100.43	weave	0	Running	2.96 %	47.3 MB
🟪 weave-scope-agent-6qsc4	ip-172-20-121-91	weave-scope-agent		1	28 minutes ago	172.20.121.91	weave	0	Running	2.71 %	46.6 MB
🟪 weave-scope-agent-vtjt2	ip-172-20-101-103	weave-scope-agent		1	28 minutes ago	172.20.101.103	weave	0	Running	2.64 %	45.8 MB
🟪 weave-scope-agent-k5pkg	ip-172-20-41-177	weave-scope-agent		1	28 minutes ago	172.20.41.177	weave	0	Running	2.62 %	54.8 MB
🟪 weave-scope-agent-2jjq6	ip-172-20-95-183	weave-scope-agent		1	28 minutes ago	172.20.95.183	weave	0	Running	2.50 %	46.8 MB
🟪 weave-scope-agent-2cqqw	ip-172-20-70-237	weave-scope-agent		1	28 minutes ago	172.20.70.237	weave	0	Running	2.43 %	43.2 MB
🟪 weave-scope-agent-69klh	ip-172-20-109-75	weave-scope-agent		1	28 minutes ago	172.20.109.75	weave	0	Running	2.31 %	48.7 MB
🟪 weave-scope-agent-lcf7r	ip-172-20-122-132	weave-scope-agent		1	28 minutes ago	172.20.122.132	weave	0	Running	2.27 %	42.8 MB
🟪 weave-scope-agent-rjgrp	ip-172-20-52-123	weave-scope-agent		1	28 minutes ago	172.20.52.123	weave	0	Running	2.22 %	47.4 MB
🟪 weave-scope-agent-nq9m7	ip-172-20-76-14	weave-scope-agent		1	28 minutes ago	172.20.76.14	weave	0	Running	2.09 %	47 MB
🟪 weave-scope-agent-zbxqs	ip-172-20-68-233	weave-scope-agent		1	28 minutes ago	172.20.68.233	weave	0	Running	2.00 %	47.4 MB
🟪 weave-scope-agent-h29s4	ip-172-20-62-116	weave-scope-agent		1	28 minutes ago	172.20.62.116	weave	0	Running	2.00 %	43.3 MB
🟪 weave-scope-agent-z76zc	ip-172-20-65-194	weave-scope-agent		1	28 minutes ago	172.20.65.194	weave	0	Running	1.94 %	44.8 MB
weave-scope-agent-k6mfj	ip-172-20-43-94	weave-scope-agent		1	28 minutes ago	172.20.43.94	weave	0	Running	1.94 %	47 MB
[illegible]	ip-172-20-78-248	weave-scope-agent		1	28 minutes ago	172.20.78.248	weave	0	Running	1.88 %	50.6 MB
[illegible]	ip-172-20-44-239	weave-scope-agent		1	28 minutes ago	172.20.44.239	weave	0	Running	1.84 %	45.7 MB

379 nodes (53 filtered)

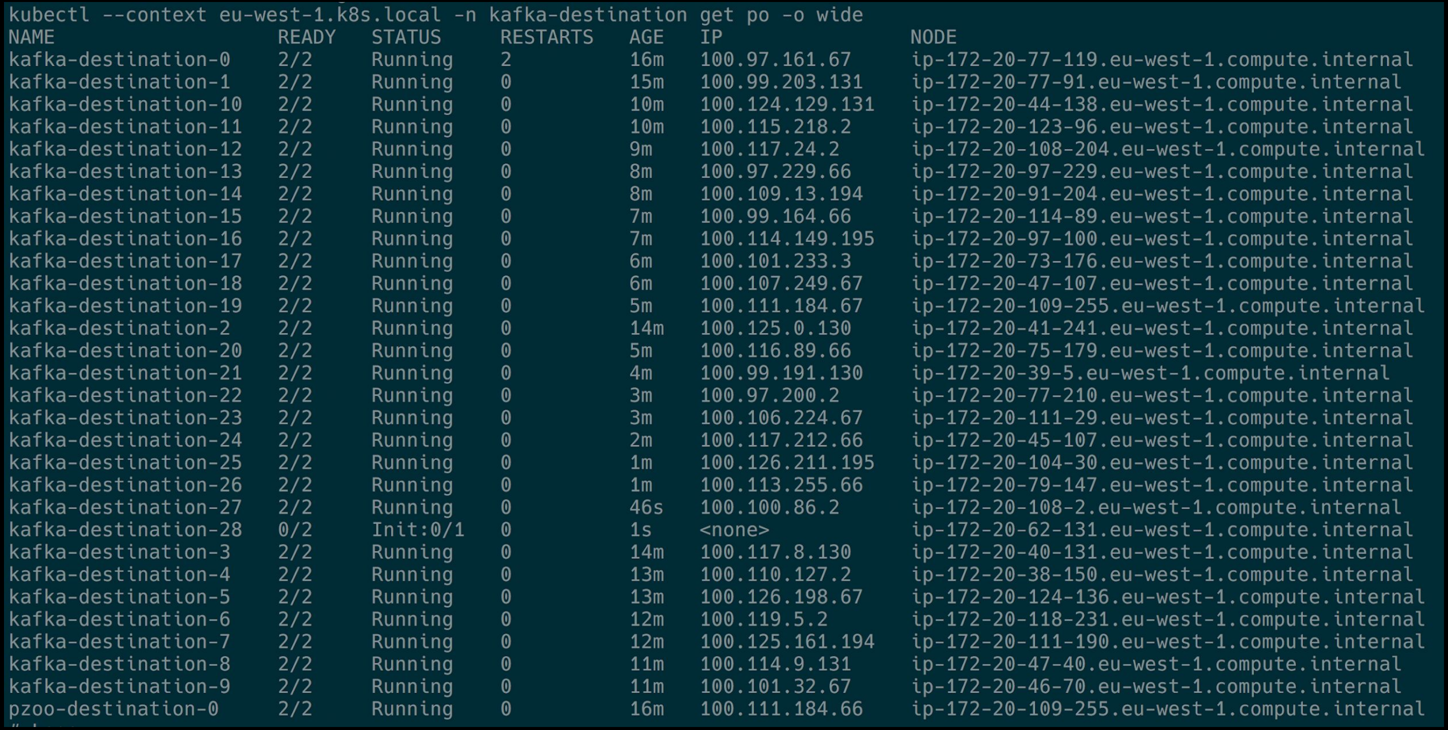
Show snapshots Hide snapshots

Show storage Hide storage

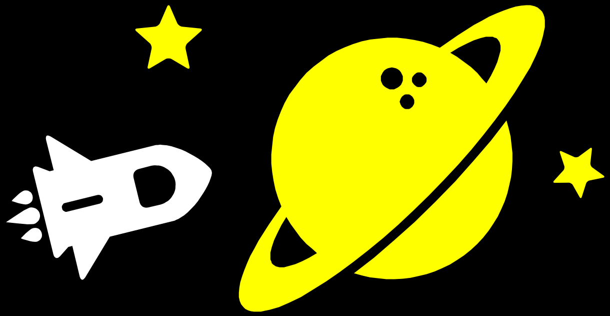
Weave Scope


kafka-mirror-te...
10 pods
The Internet
Inbound connecti...
kubernetes
0 pods
pzoo
0 pods
broker
29 pods
zookeeper
1 pod

kubectl example



```
kubectl --context eu-west-1.k8s.local -n kafka-destination get po -o wide
NAME                   READY   STATUS     RESTARTS   AGE   IP                NODE
kafka-destination-0    2/2     Running    2          16m   100.97.161.67     ip-172-20-77-119.eu-west-1.compute.internal
kafka-destination-1    2/2     Running    0          15m   100.99.203.131    ip-172-20-77-91.eu-west-1.compute.internal
kafka-destination-10   2/2     Running    0          10m   100.124.129.131   ip-172-20-44-138.eu-west-1.compute.internal
kafka-destination-11   2/2     Running    0          10m   100.115.218.2     ip-172-20-123-96.eu-west-1.compute.internal
kafka-destination-12   2/2     Running    0          9m    100.117.24.2      ip-172-20-108-204.eu-west-1.compute.internal
kafka-destination-13   2/2     Running    0          8m    100.97.229.66     ip-172-20-97-229.eu-west-1.compute.internal
kafka-destination-14   2/2     Running    0          8m    100.109.13.194    ip-172-20-91-204.eu-west-1.compute.internal
kafka-destination-15   2/2     Running    0          7m    100.99.164.66     ip-172-20-114-89.eu-west-1.compute.internal
kafka-destination-16   2/2     Running    0          7m    100.114.149.195   ip-172-20-97-100.eu-west-1.compute.internal
kafka-destination-17   2/2     Running    0          6m    100.101.233.3     ip-172-20-73-176.eu-west-1.compute.internal
kafka-destination-18   2/2     Running    0          6m    100.107.249.67    ip-172-20-47-107.eu-west-1.compute.internal
kafka-destination-19   2/2     Running    0          5m    100.111.184.67    ip-172-20-109-255.eu-west-1.compute.internal
kafka-destination-2    2/2     Running    0          14m   100.125.0.130     ip-172-20-41-241.eu-west-1.compute.internal
kafka-destination-20   2/2     Running    0          5m    100.116.89.66     ip-172-20-75-179.eu-west-1.compute.internal
kafka-destination-21   2/2     Running    0          4m    100.99.191.130    ip-172-20-39-5.eu-west-1.compute.internal
kafka-destination-22   2/2     Running    0          3m    100.97.200.2      ip-172-20-77-210.eu-west-1.compute.internal
kafka-destination-23   2/2     Running    0          3m    100.106.224.67    ip-172-20-111-29.eu-west-1.compute.internal
kafka-destination-24   2/2     Running    0          2m    100.117.212.66    ip-172-20-45-107.eu-west-1.compute.internal
kafka-destination-25   2/2     Running    0          1m    100.126.211.195   ip-172-20-104-30.eu-west-1.compute.internal
kafka-destination-26   2/2     Running    0          1m    100.113.255.66    ip-172-20-79-147.eu-west-1.compute.internal
kafka-destination-27   2/2     Running    0          46s   100.100.86.2      ip-172-20-108-2.eu-west-1.compute.internal
kafka-destination-28   0/2     Init:0/1   0          1s    <none>            ip-172-20-62-131.eu-west-1.compute.internal
kafka-destination-3    2/2     Running    0          14m   100.117.8.130     ip-172-20-40-131.eu-west-1.compute.internal
kafka-destination-4    2/2     Running    0          13m   100.110.127.2     ip-172-20-38-150.eu-west-1.compute.internal
kafka-destination-5    2/2     Running    0          13m   100.126.198.67    ip-172-20-124-136.eu-west-1.compute.internal
kafka-destination-6    2/2     Running    0          12m   100.119.5.2       ip-172-20-118-231.eu-west-1.compute.internal
kafka-destination-7    2/2     Running    0          12m   100.125.161.194   ip-172-20-111-190.eu-west-1.compute.internal
kafka-destination-8    2/2     Running    0          11m   100.114.9.131     ip-172-20-47-40.eu-west-1.compute.internal
kafka-destination-9    2/2     Running    0          11m   100.101.32.67     ip-172-20-46-70.eu-west-1.compute.internal
pzoo-destination-0     2/2     Running    0          16m   100.111.184.66    ip-172-20-109-255.eu-west-1.compute.internal
```



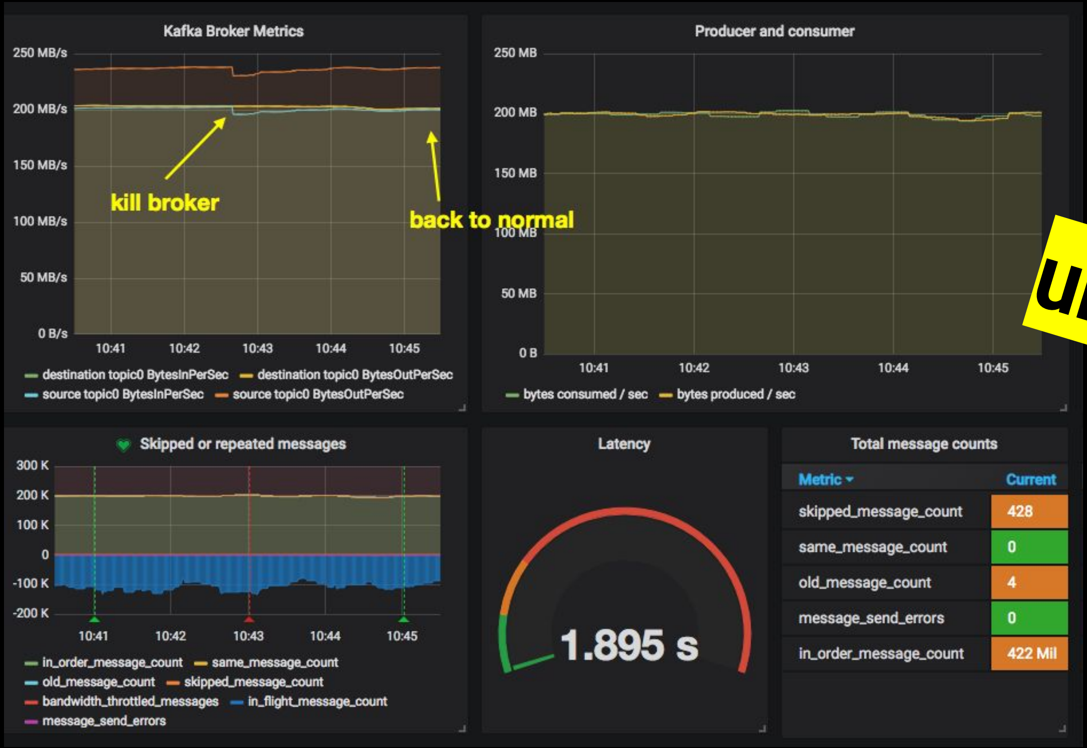
TEST IT

Let’s put all this to test now

KILL A BROKER

```
$ kubectl \
  --context us-east-1.k8s.local \
  -n kafka-source \
  delete pod kafka-source-2
```

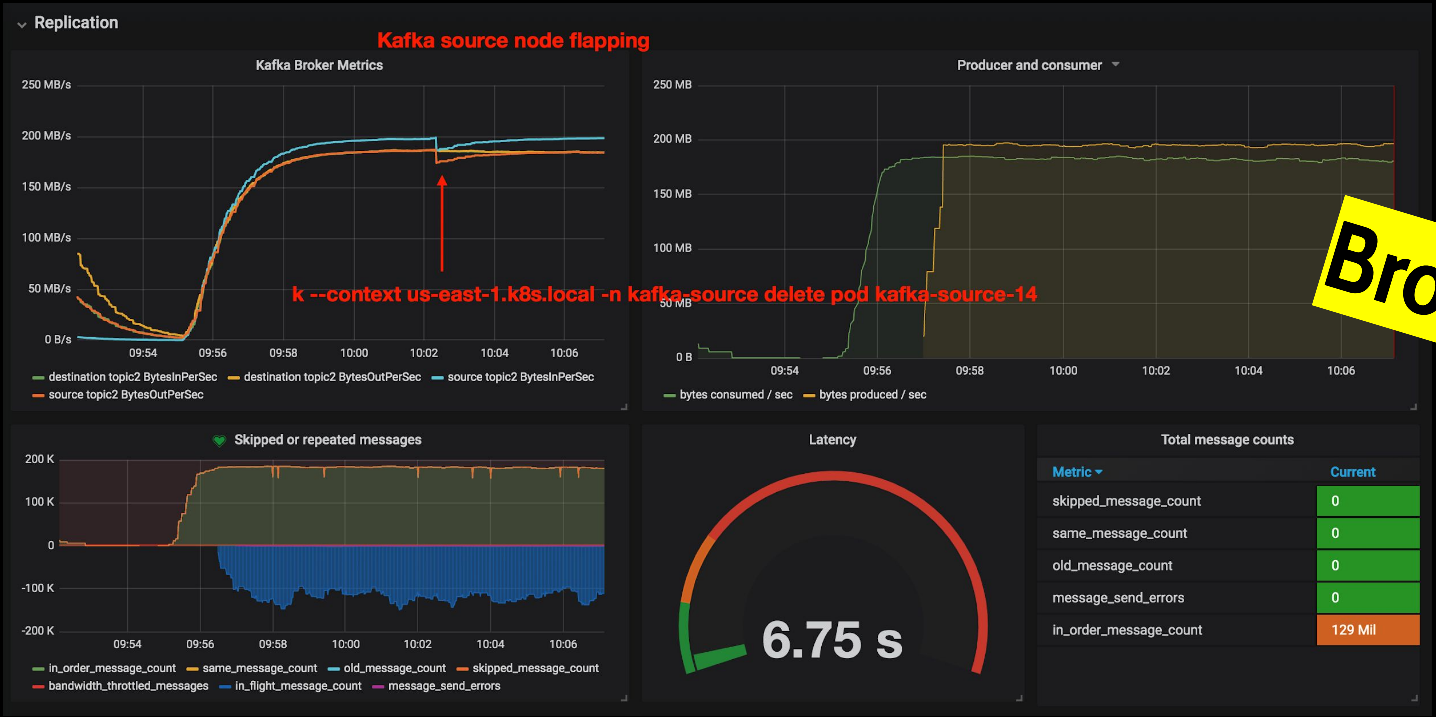
KILL A BROKER - SRC



Metric	Current
skipped_message_count	428
same_message_count	0
old_message_count	4
message_send_errors	0
in_order_message_count	422 Mil

uReplicator

KILL A BROKER - SRC



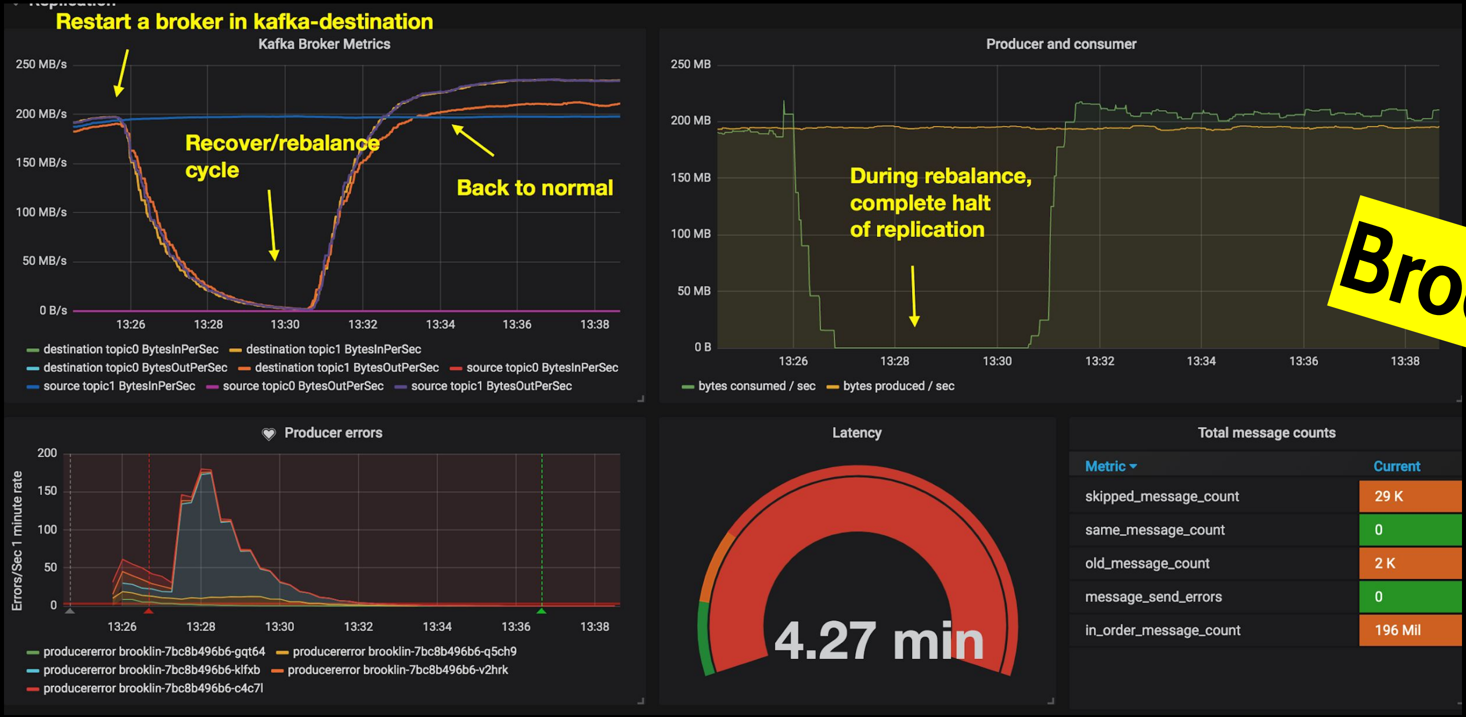
```
k --context us-east-1.k8s.local -n kafka-source delete pod kafka-source-14
```

Total message counts

Metric	Current
skipped_message_count	0
same_message_count	0
old_message_count	0
message_send_errors	0
in_order_message_count	129 Mil

Brooklin

KILL A BROKER - DST



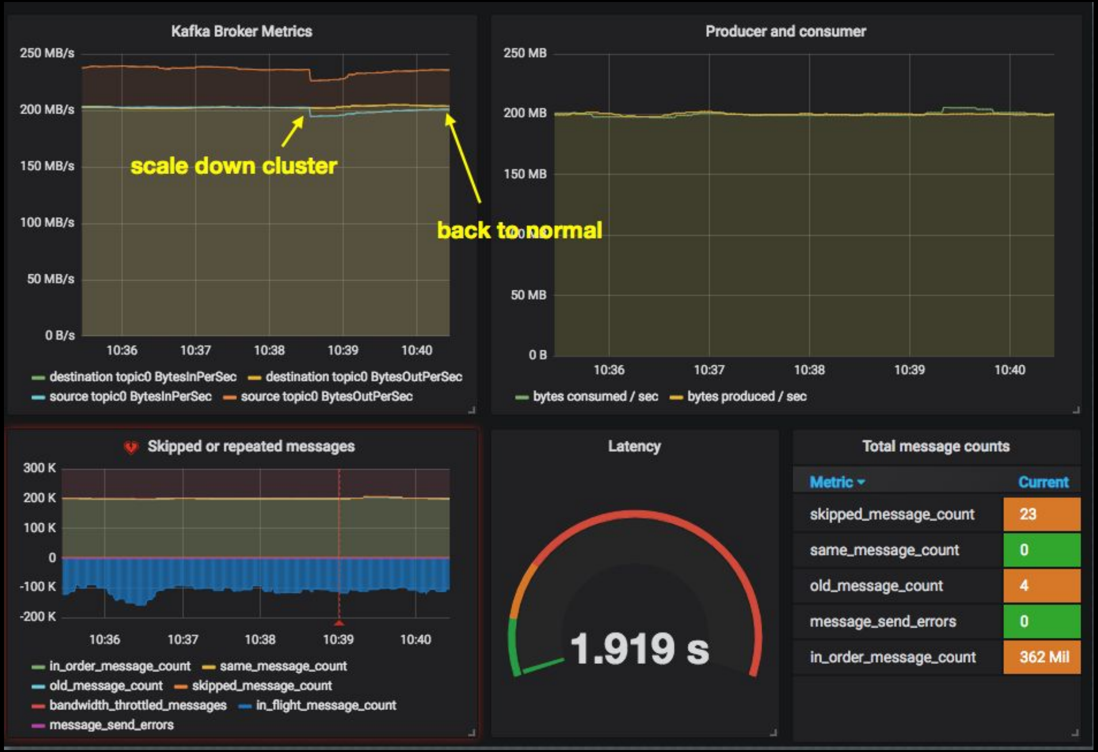
Total message counts

Metric	Current
skipped_message_count	29 K
same_message_count	0
old_message_count	2 K
message_send_errors	0
in_order_message_count	196 Mil

RESIZE KAFKA CLUSTER

```
$ kubectl \
    --context us-east-1.k8s.local \
    -n kafka-source \
    scale \
    statefulset kafka-source \
    --replicas 29
```

RESIZE KAFKA CLUSTER



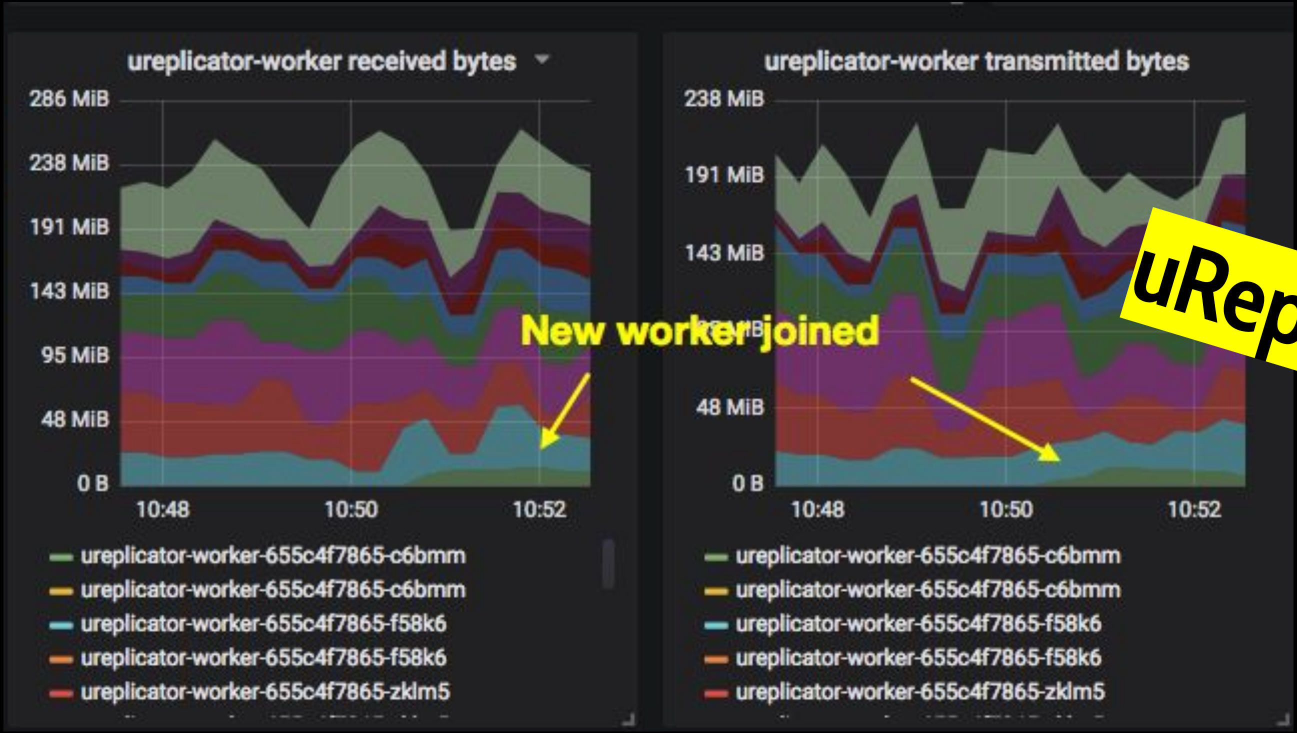
Total message counts

Metric	Current
skipped_message_count	23
same_message_count	0
old_message_count	4
message_send_errors	0
in_order_message_count	362 Mil

ADD uREPLICATOR WORKER

```
$ kubectl \
   --context eu-west-1.k8s.local \
   -n ureplicator \
   scale deployment \
   ureplicator-worker \
   --replicas 9
```

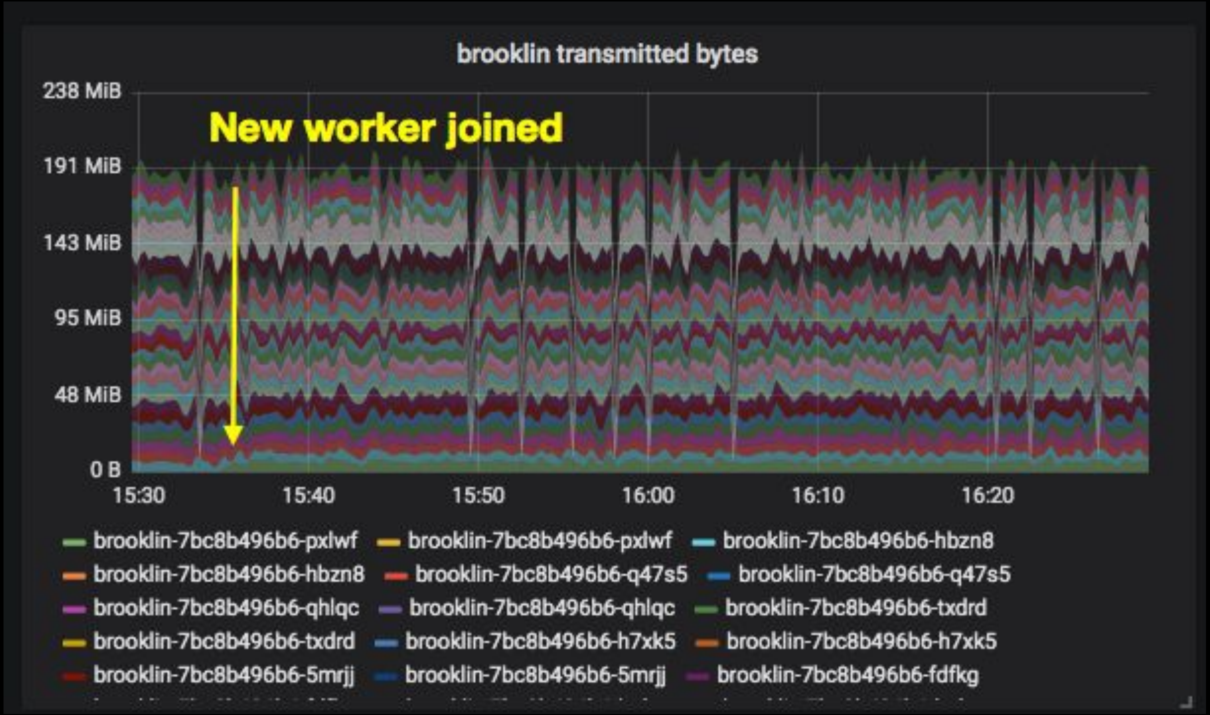
ADD uReplicator WORKER



- ureplicator-worker-655c4f7865-c6bmm
- ureplicator-worker-655c4f7865-c6bmm
- ureplicator-worker-655c4f7865-f58k6
- ureplicator-worker-655c4f7865-f58k6
- ureplicator-worker-655c4f7865-zklm5

- ureplicator-worker-655c4f7865-c6bmm
- ureplicator-worker-655c4f7865-c6bmm
- ureplicator-worker-655c4f7865-f58k6
- ureplicator-worker-655c4f7865-f58k6
- ureplicator-worker-655c4f7865-zklm5

ADD Brooklin WORKER

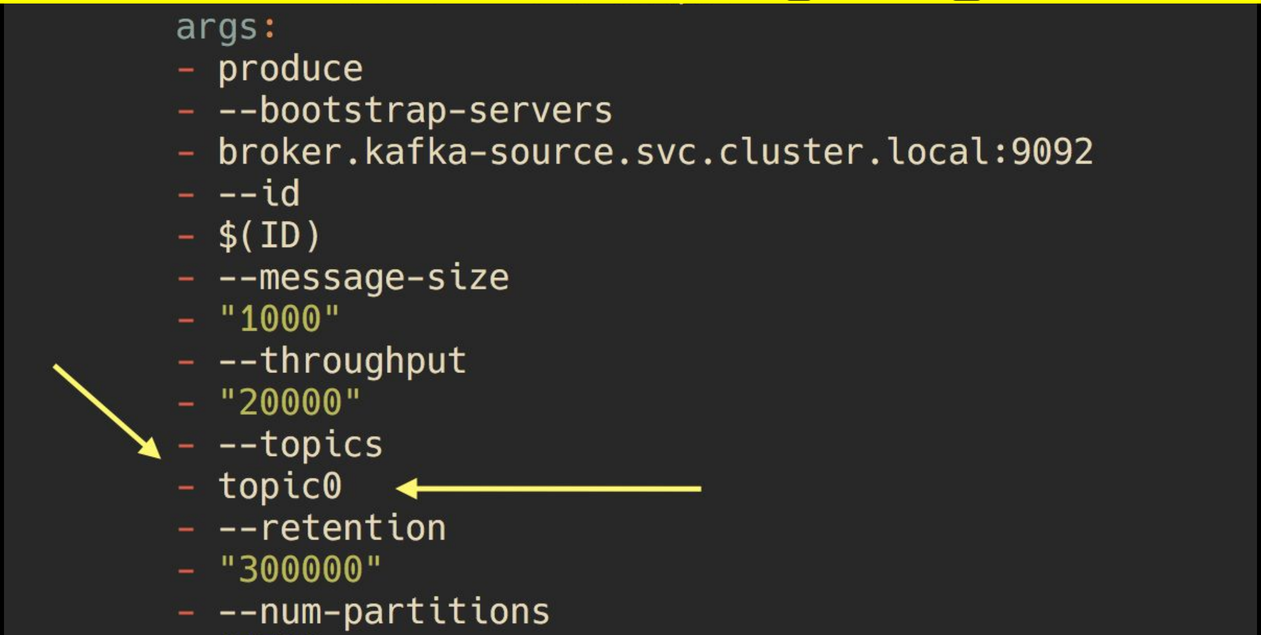


brooklin-7bc8b496b6-pxlwf brooklin-7bc8b496b6-pxlwf brooklin-7bc8b496b6-hbzn8
brooklin-7bc8b496b6-hbzn8 brooklin-7bc8b496b6-q47s5 brooklin-7bc8b496b6-q47s5
brooklin-7bc8b496b6-qhlqc brooklin-7bc8b496b6-qhlqc brooklin-7bc8b496b6-txdrd
brooklin-7bc8b496b6-txdrd brooklin-7bc8b496b6-h7xk5 brooklin-7bc8b496b6-h7xk5
brooklin-7bc8b496b6-5mrjj brooklin-7bc8b496b6-5mrjj brooklin-7bc8b496b6-fdfkg

Brooklin

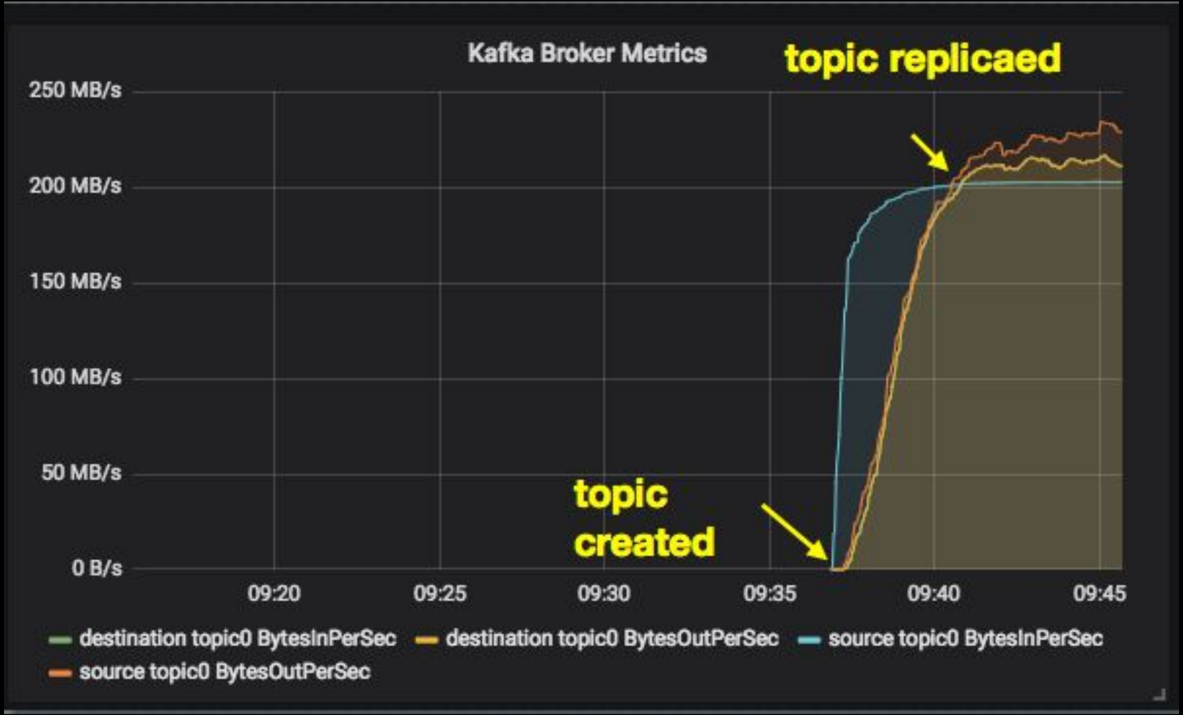
ADD NEW TOPIC

```
$ make k8s-redeploy-tests
```



```
args:
- produce
- --bootstrap-servers
- broker.kafka-source.svc.cluster.local:9092
- --id
- $(ID)
- --message-size
- "1000"
- --throughput
- "20000"
- --topics
- topic0
- --retention
- "300000"
- --num-partitions
```

ADD NEW TOPIC



ADD PARTITIONS

```
$ make k8s-kafka-shell-source

$ bin/kafka-topics.sh --zookeeper
zookeeper:2181 --alter --topic
topic5 --partitions 300
```

ADD PARTITIONS

Problem: **uReplicator** does not see the new partitions

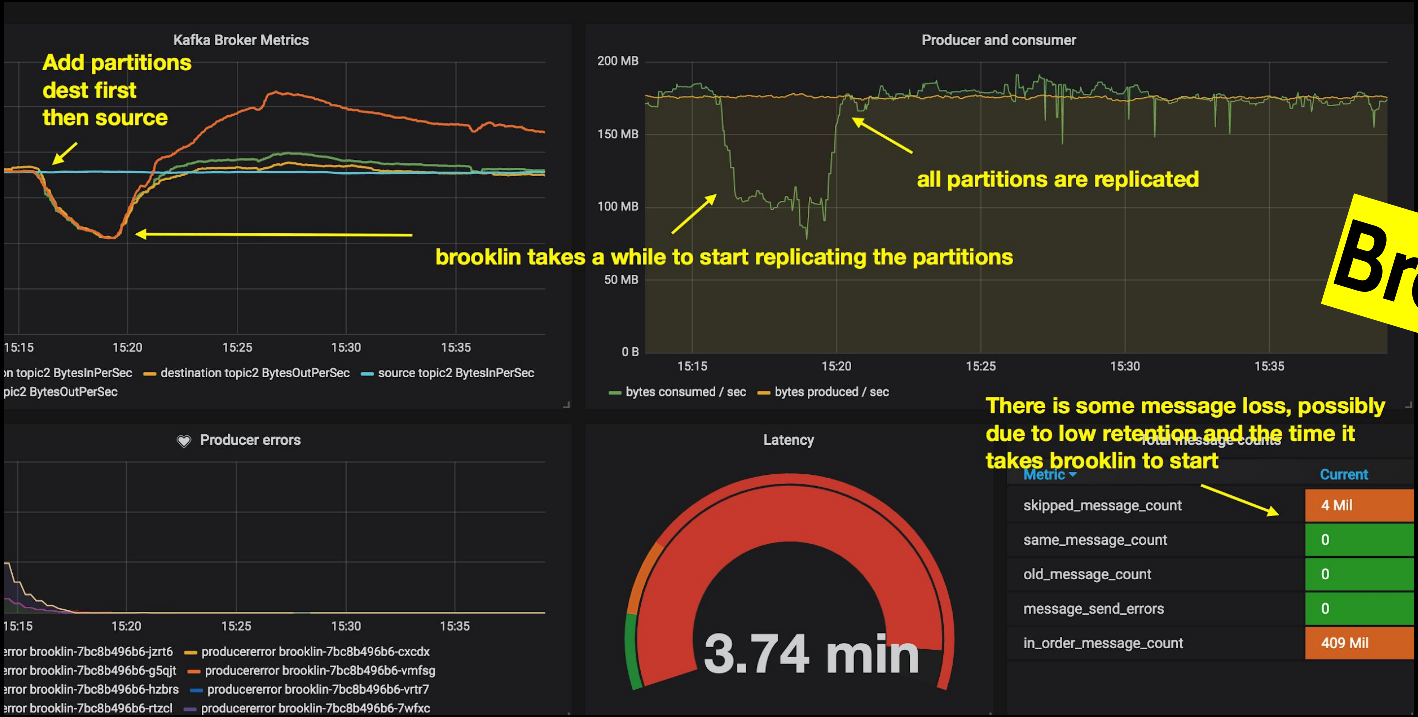
uReplicator

ADD PARTITIONS (fix)

uReplicator

```
$ kubectl --context eu-west-1.k8s.local \
   -n ureplicator port-forward \
   ureplicator-controller-76ff85b889-19mzl 9000

$ curl -X DELETE http://localhost:9000/topics/topic5

$ curl -X POST -d \
   '{"topic":"topic5", "numPartitions":"300"}' \
   http://localhost:9000/topics
```

ADD PARTITIONS



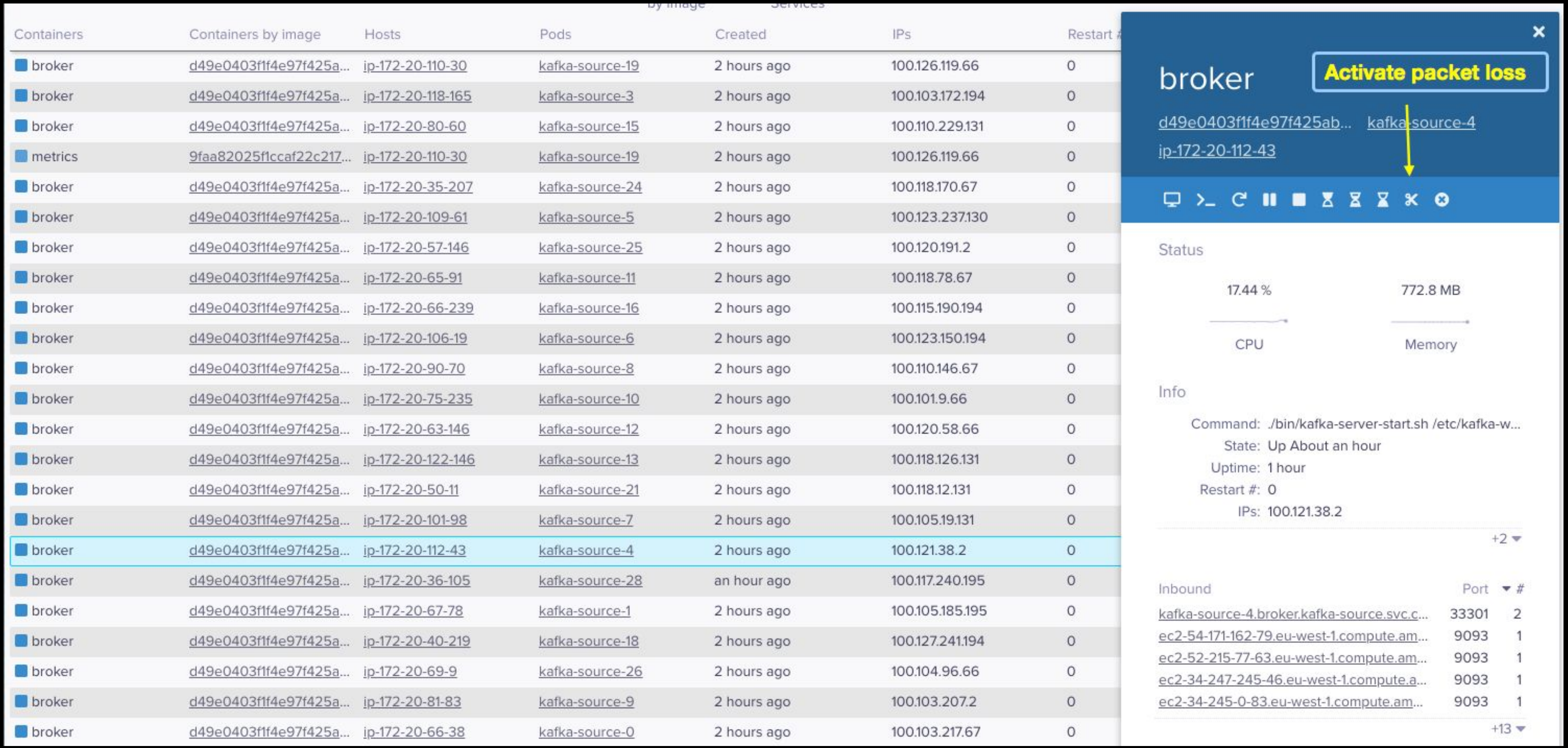
Kafka Broker Metrics
Add partitions dest first then source
15:15
15:20
15:25
15:30
15:35
n topic2 BytesInPerSec
destination topic2 BytesOutPerSec
source topic2 BytesInPerSec
pic2 BytesOutPerSec
Producer and consumer
200 MB
150 MB
100 MB
50 MB
0 B
all partitions are replicated
brooklin takes a while to start replicating the partitions
bytes consumed / sec
bytes produced / sec
Producer errors
rror brooklin-7bc8b496b6-jzrt6
producererror brooklin-7bc8b496b6-cxcdx
rror brooklin-7bc8b496b6-g5qjt
producererror brooklin-7bc8b496b6-vmfsg
rror brooklin-7bc8b496b6-hzbrs
producererror brooklin-7bc8b496b6-vrtr7
rror brooklin-7bc8b496b6-rtzcl
producererror brooklin-7bc8b496b6-7wfxc
Latency
3.74 min
Total message counts

There is some message loss, possibly due to low retention and the time it takes brooklin to start

Metric	Current
skipped_message_count	4 Mil
same_message_count	0
old_message_count	0
message_send_errors	0
in_order_message_count	409 Mil

Brooklin

PACKET LOSS



Containers	Containers by image	Hosts	Pods	Created	IPs	Restart #
broker	d49e0403f1f4e97f425a...	ip-172-20-110-30	kafka-source-19	2 hours ago	100.126.119.66	0
broker	d49e0403f1f4e97f425a...	ip-172-20-118-165	kafka-source-3	2 hours ago	100.103.172.194	0
broker	d49e0403f1f4e97f425a...	ip-172-20-80-60	kafka-source-15	2 hours ago	100.110.229.131	0
metrics	9faa82025f1ccaf22c217...	ip-172-20-110-30	kafka-source-19	2 hours ago	100.126.119.66	0
broker	d49e0403f1f4e97f425a...	ip-172-20-35-207	kafka-source-24	2 hours ago	100.118.170.67	0
broker	d49e0403f1f4e97f425a...	ip-172-20-109-61	kafka-source-5	2 hours ago	100.123.237.130	0
broker	d49e0403f1f4e97f425a...	ip-172-20-57-146	kafka-source-25	2 hours ago	100.120.191.2	0
broker	d49e0403f1f4e97f425a...	ip-172-20-65-91	kafka-source-11	2 hours ago	100.118.78.67	0
broker	d49e0403f1f4e97f425a...	ip-172-20-66-239	kafka-source-16	2 hours ago	100.115.190.194	0
broker	d49e0403f1f4e97f425a...	ip-172-20-106-19	kafka-source-6	2 hours ago	100.123.150.194	0
broker	d49e0403f1f4e97f425a...	ip-172-20-90-70	kafka-source-8	2 hours ago	100.110.146.67	0
broker	d49e0403f1f4e97f425a...	ip-172-20-75-235	kafka-source-10	2 hours ago	100.101.9.66	0
broker	d49e0403f1f4e97f425a...	ip-172-20-63-146	kafka-source-12	2 hours ago	100.120.58.66	0
broker	d49e0403f1f4e97f425a...	ip-172-20-122-146	kafka-source-13	2 hours ago	100.118.126.131	0
broker	d49e0403f1f4e97f425a...	ip-172-20-50-11	kafka-source-21	2 hours ago	100.118.12.131	0
broker	d49e0403f1f4e97f425a...	ip-172-20-101-98	kafka-source-7	2 hours ago	100.105.19.131	0
broker	d49e0403f1f4e97f425a...	ip-172-20-112-43	kafka-source-4	2 hours ago	100.121.38.2	0
broker	d49e0403f1f4e97f425a...	ip-172-20-36-105	kafka-source-28	an hour ago	100.117.240.195	0
broker	d49e0403f1f4e97f425a...	ip-172-20-67-78	kafka-source-1	2 hours ago	100.105.185.195	0
broker	d49e0403f1f4e97f425a...	ip-172-20-40-219	kafka-source-18	2 hours ago	100.127.241.194	0
broker	d49e0403f1f4e97f425a...	ip-172-20-69-9	kafka-source-26	2 hours ago	100.104.96.66	0
broker	d49e0403f1f4e97f425a...	ip-172-20-81-83	kafka-source-9	2 hours ago	100.103.207.2	0
broker	d49e0403f1f4e97f425a...	ip-172-20-66-38	kafka-source-0	2 hours ago	100.103.217.67	0

broker

Activate packet loss

d49e0403f1f4e97f425ab... kafka-source-4
ip-172-20-112-43

Status

17.44 % CPU

772.8 MB Memory

Info

Command: ./bin/kafka-server-start.sh /etc/kafka-w...
State: Up About an hour
Uptime: 1 hour
Restart #: 0
IPs: 100.121.38.2

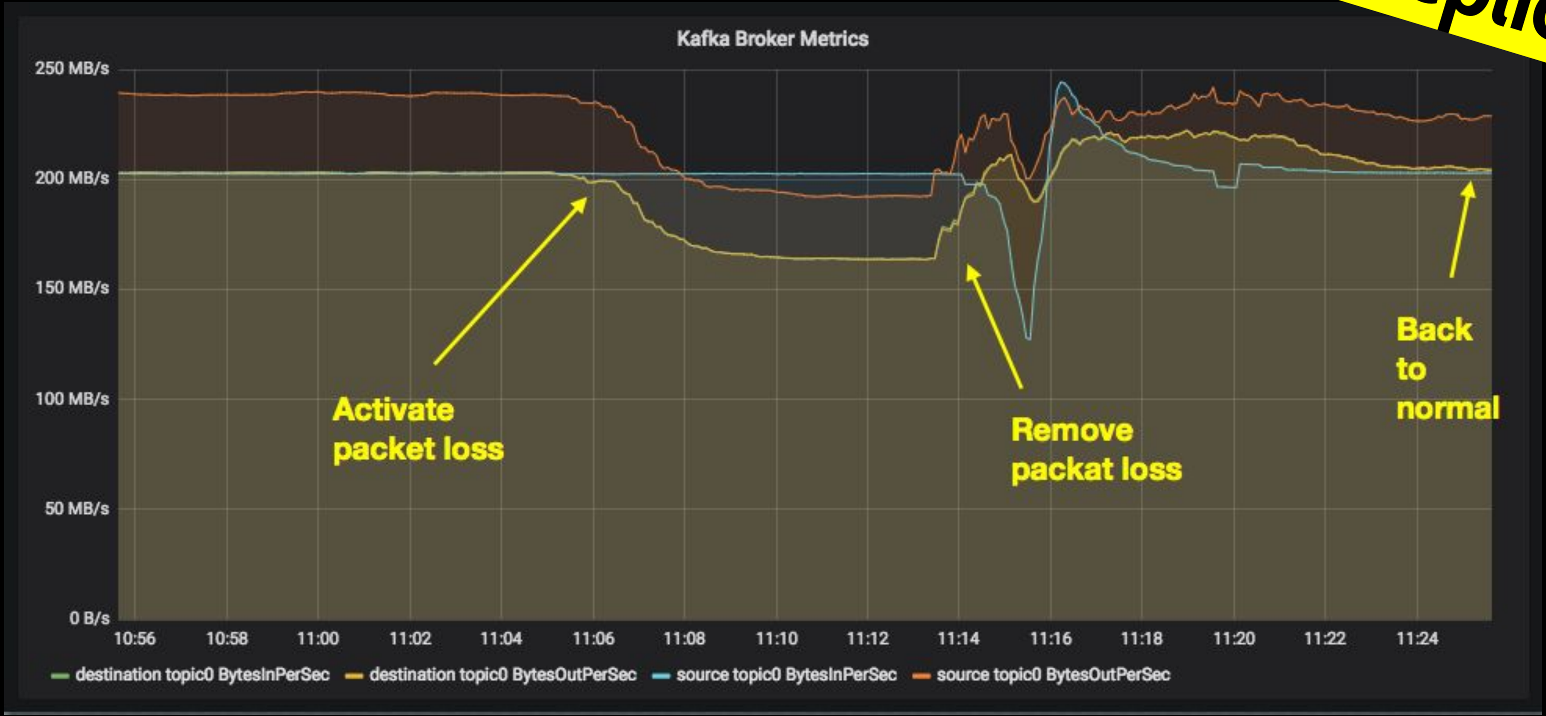
+2

Inbound	Port	#
kafka-source-4.broker.kafka-source.svc.c...	33301	2
ec2-54-171-162-79.eu-west-1.compute.am...	9093	1
ec2-52-215-77-63.eu-west-1.compute.am...	9093	1
ec2-34-247-245-46.eu-west-1.compute.a...	9093	1
ec2-34-245-0-83.eu-west-1.compute.am...	9093	1

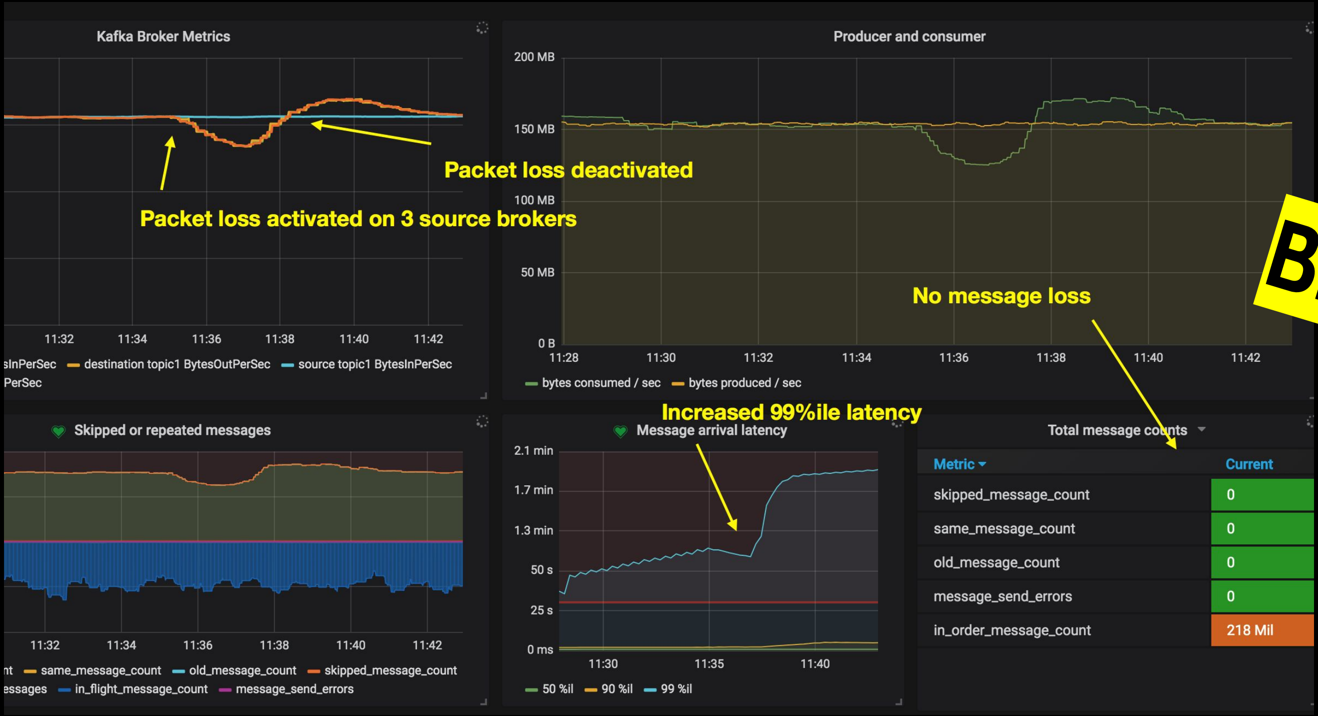
+13

PACKET LOSS

uReplicator



PACKET LOSS



Metric	Current
skipped_message_count	0
same_message_count	0
old_message_count	0
message_send_errors	0
in_order_message_count	218 Mil

Brooklin

This is so much FUN!!!

But how do we actually do that?

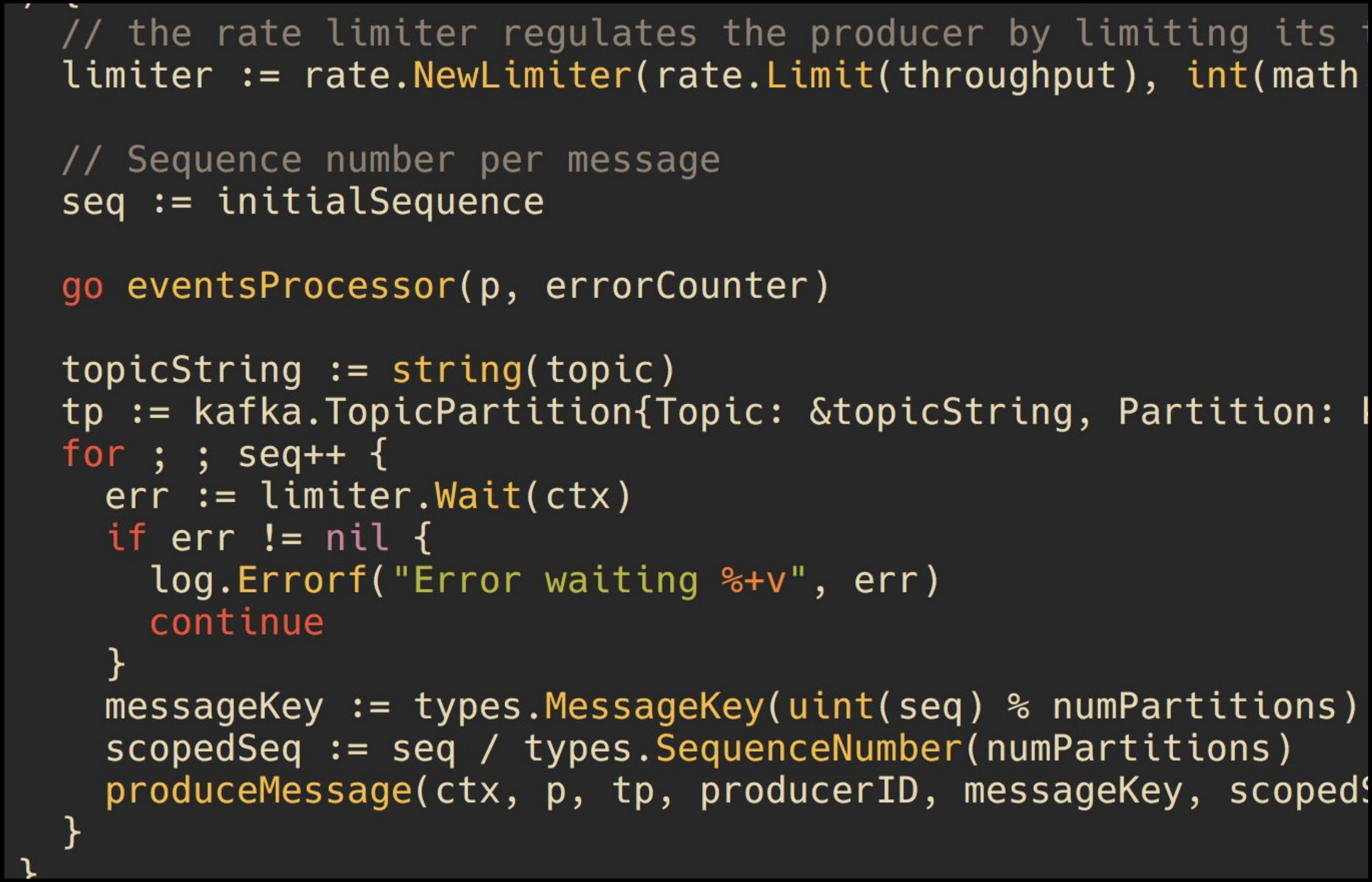
CODE DEEP DIVE

Golang Producer/Consumer

(just a few bites)

Producer

Main loop



```go
  // the rate limiter regulates the producer by limiting its
  limiter := rate.NewLimiter(rate.Limit(throughput), int(math

  // Sequence number per message
  seq := initialSequence

  go eventsProcessor(p, errorCounter)

  topicString := string(topic)
  tp := kafka.TopicPartition{Topic: &topicString, Partition:
  for ; ; seq++ {
    err := limiter.Wait(ctx)
    if err != nil {
      log.Errorf("Error waiting %+v", err)
      continue
    }
    messageKey := types.MessageKey(uint(seq) % numPartitions)
    scopedSeq := seq / types.SequenceNumber(numPartitions)
    produceMessage(ctx, p, tp, producerID, messageKey, scoped
  }
}
```

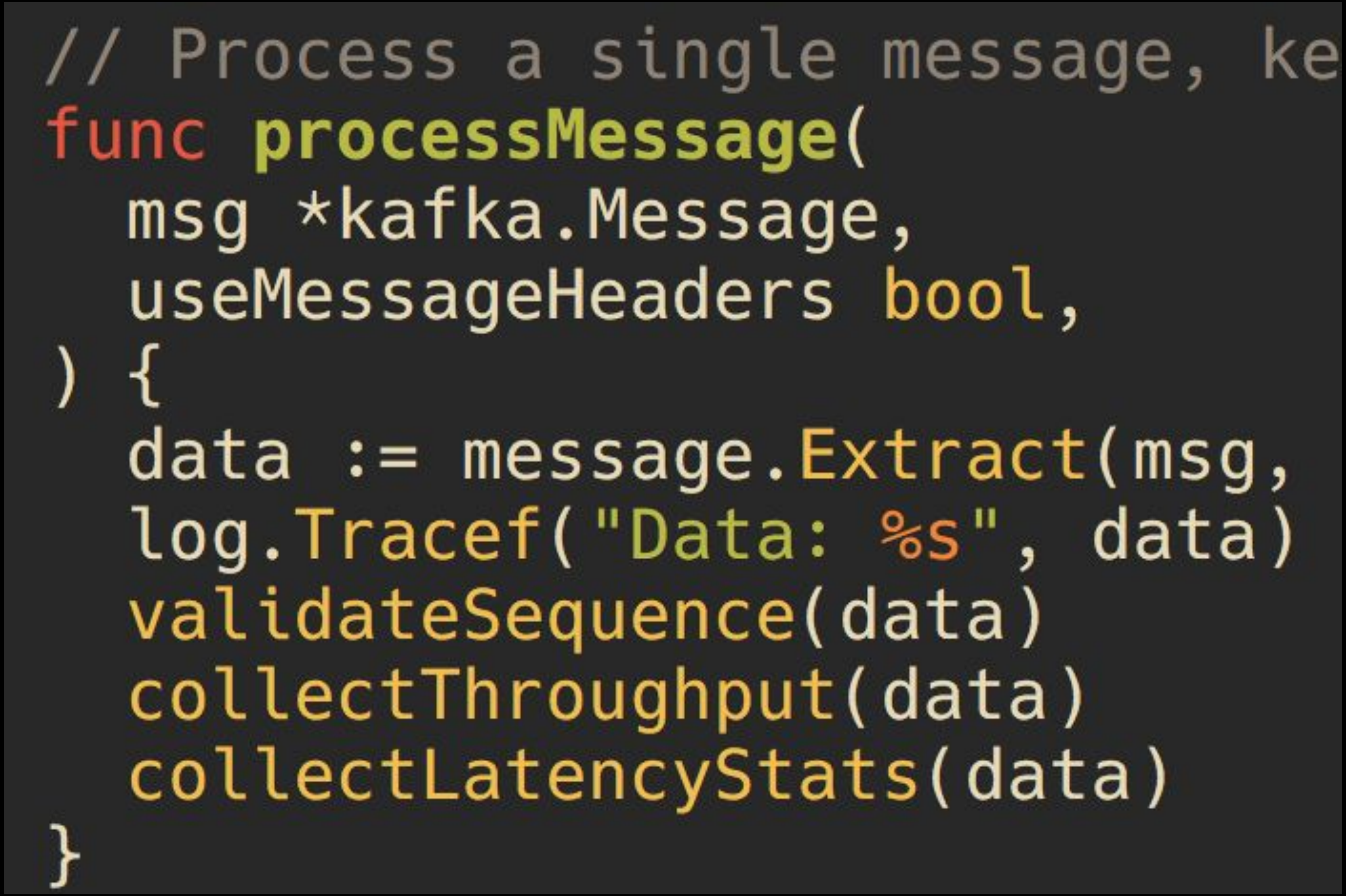
Consumer

Main loop

```go
sigchan := make(chan os.Signal, 1)
signal.Notify(sigchan, syscall.SIGINT, syscall.SIGTERM)

for {
  select {
  case sig := <-sigchan:
    log.Infof("Caught signal %v: terminating", sig)
    return
  case <-ctx.Done():
    log.Infof("Done. %s", ctx.Err())
    return
  case ev := <-c.Events():
    // Most events are typically juse messages, still w
    // Partition changes, EOF and Errors
    switch e := ev.(type) {
    case kafka.AssignedPartitions:
      log.Infof("AssignedPartitions %v", e)
      c.Assign(e.Partitions)
    case kafka.RevokedPartitions:
      log.Infof("RevokedPartitions %v", e)
      c.Unassign()
    case *kafka.Message:
      processMessage(e, useMessageHeaders)
    case kafka.PartitionEOF:
      log.Debugf("PartitionEOF Reached %v", e)
    case kafka.Error:
      // Errors should generally be considered as infor
      log.Errorf("Error: %+v", e)
    }
  }
}
```

Consumer

Process message


```
// Process a single message, ke
func processMessage(
  msg *kafka.Message,
  useMessageHeaders bool,
) {
  data := message.Extract(msg,
  log.Tracef("Data: %s", data)
  validateSequence(data)
  collectThroughput(data)
  collectLatencyStats(data)
}
```

“

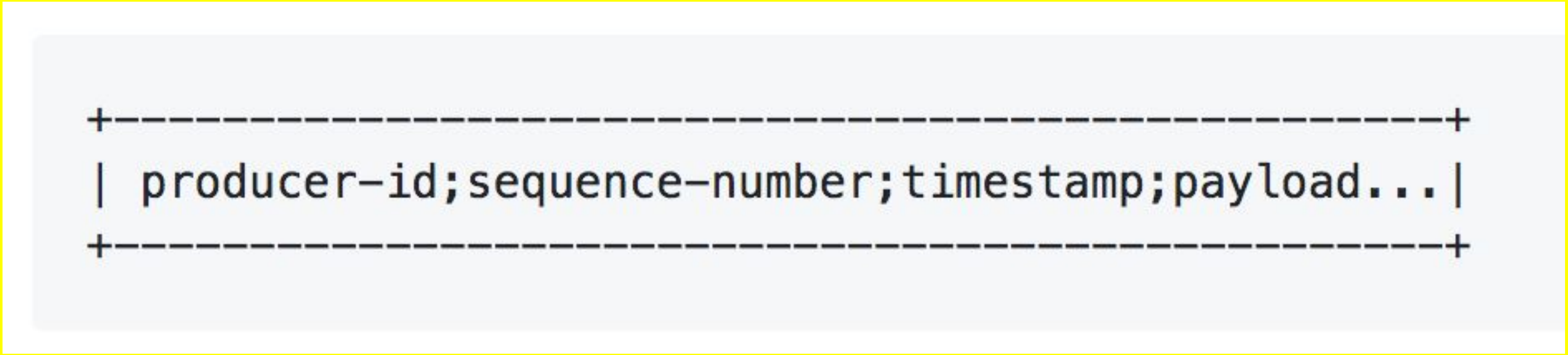
Question: How can multiple consumers validate message arrival order?

Redis?

DynamoDB?

No! There’s a trick!

Message format



```
+-------------------------------------------------+
| producer-id;sequence-number;timestamp;payload...|
+-------------------------------------------------+
```

Producer

Sequence numbers

```
messageKey = seq % partitions
perKeySeq  = seq / partitions
```

But - there is still a bug here...

Hint: 🎁 🎂 🎉

the BIRTHDAY PARADOX

Producer

Sequence numbers

Fix:

```
partitions =  partitions * 17
```

```
messageKey = seq % partitions
perKeySeq  = seq / partitions
```

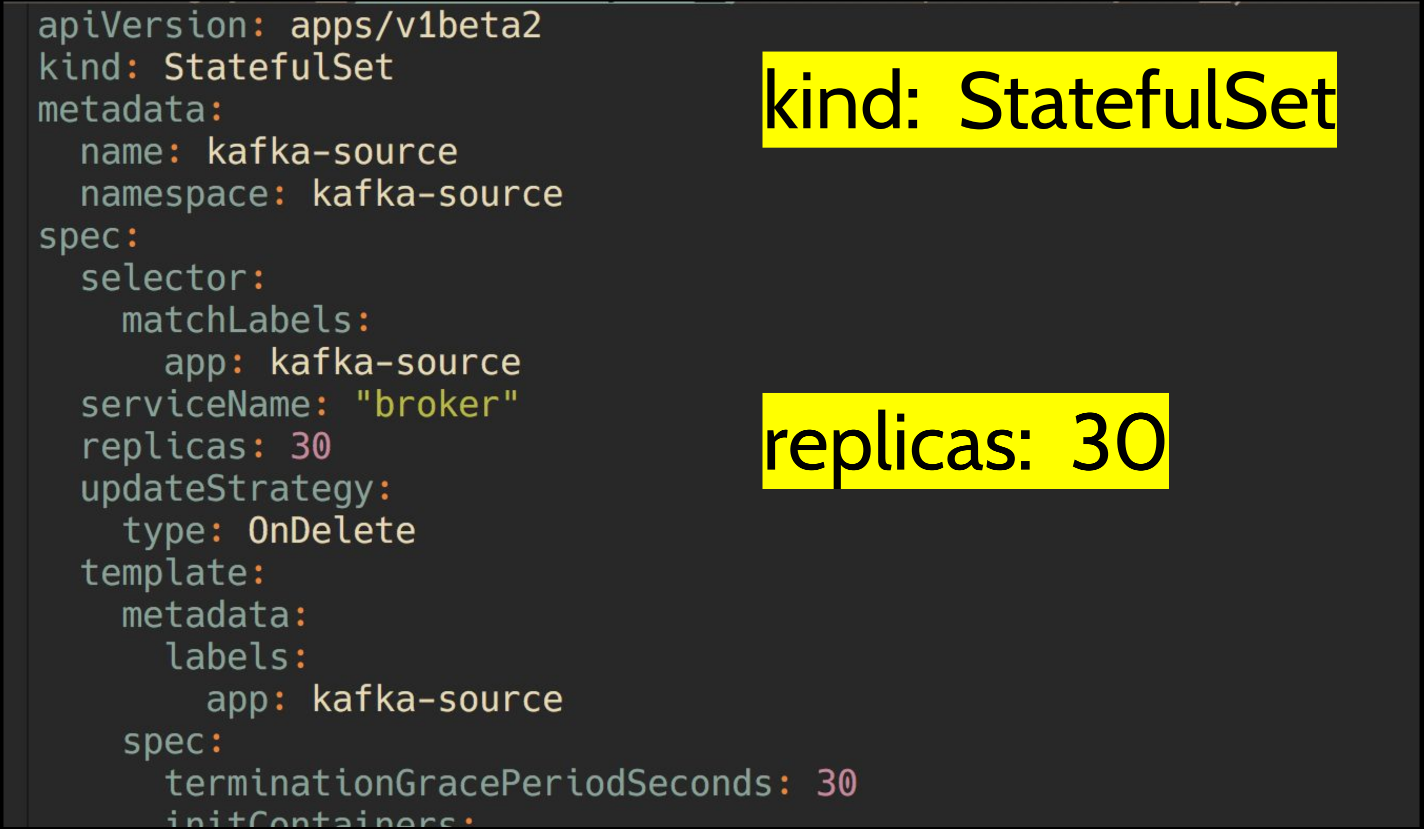
CODE DEEP DIVE

Kubernetes, head first

(just a few bites)

Kafka

There is also today a Kafka operator

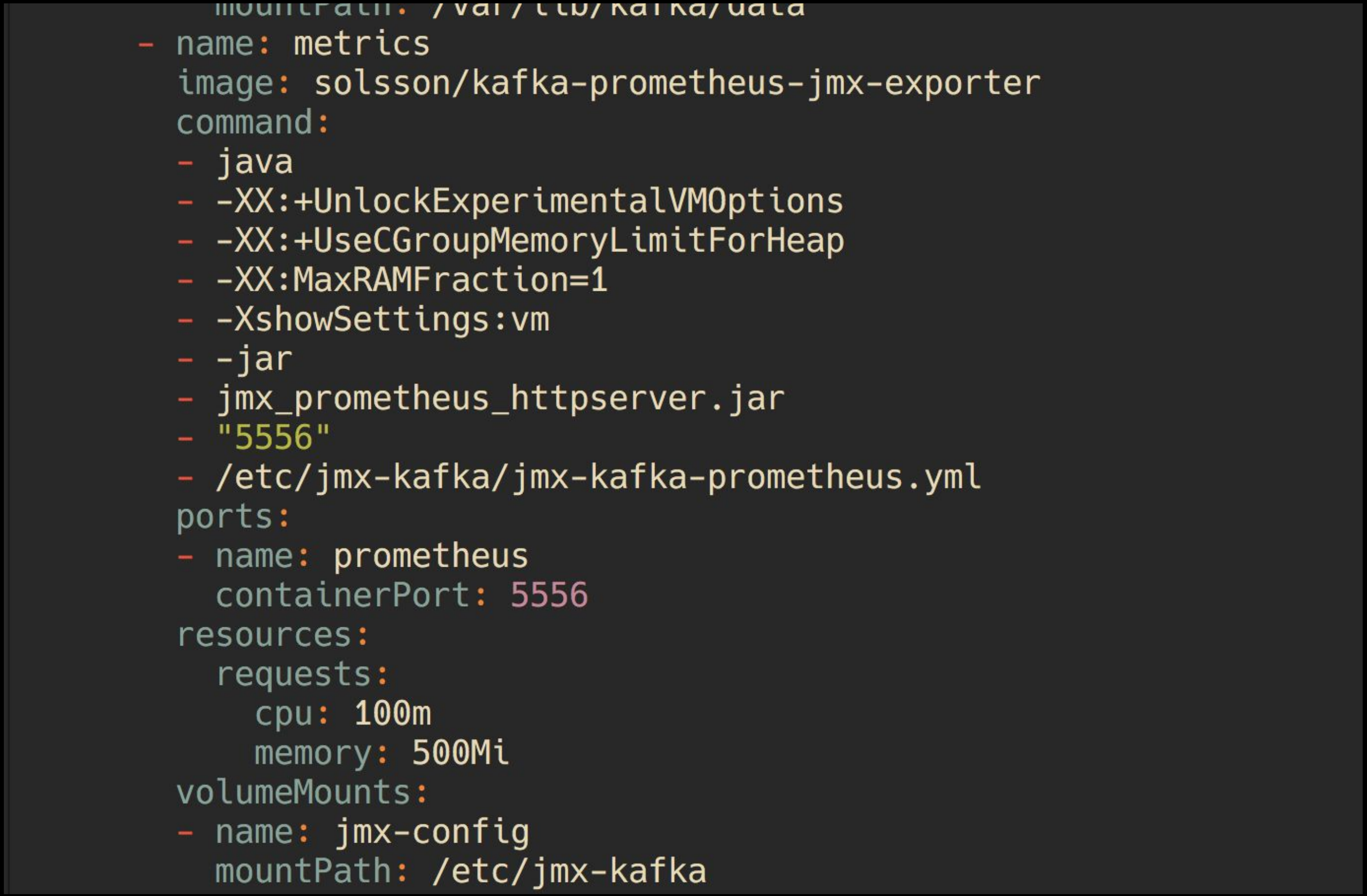


```
apiVersion: apps/v1beta2
kind: StatefulSet
metadata:
  name: kafka-source
  namespace: kafka-source
spec:
  selector:
    matchLabels:
      app: kafka-source
  serviceName: "broker"
  replicas: 30
  updateStrategy:
    type: OnDelete
  template:
    metadata:
      labels:
        app: kafka-source
    spec:
      terminationGracePeriodSeconds: 30
```

kind: StatefulSet

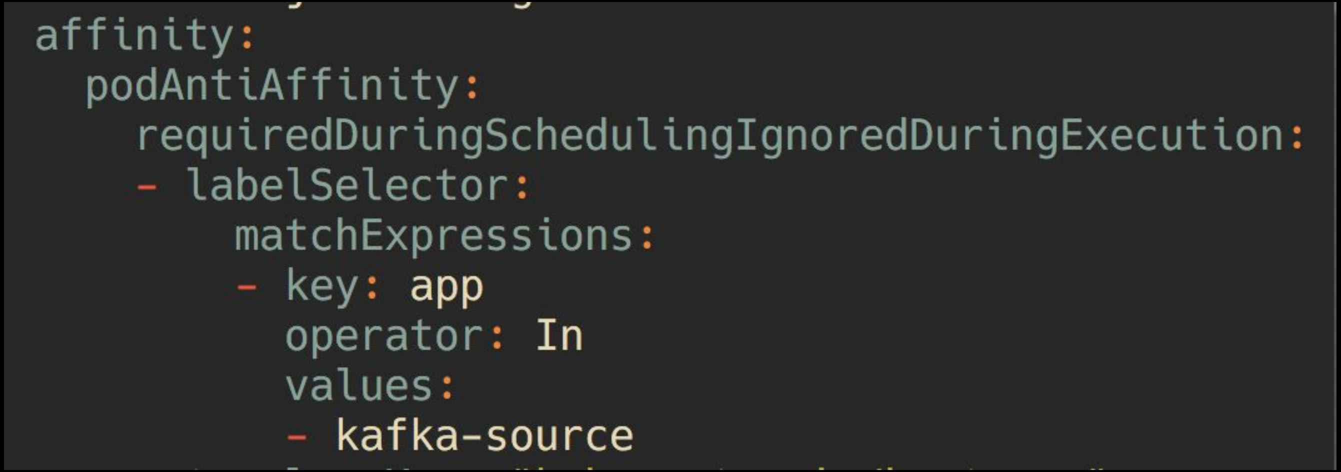
replicas: 30

Metrics sidecar container



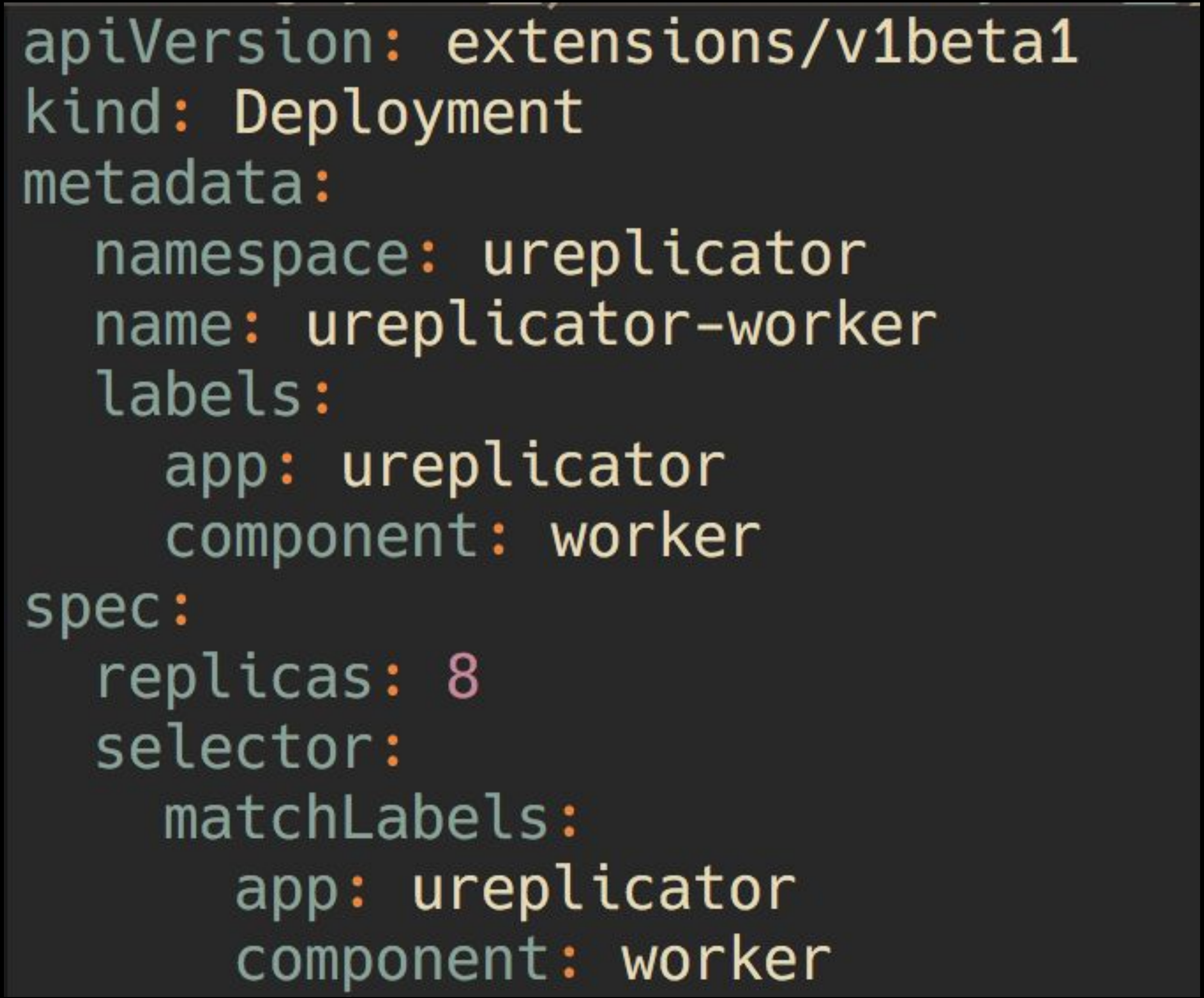
```
  mountPath: /var/lib/kafka/data
- name: metrics
  image: solsson/kafka-prometheus-jmx-exporter
  command:
  - java
  - -XX:+UnlockExperimentalVMOptions
  - -XX:+UseCGroupMemoryLimitForHeap
  - -XX:MaxRAMFraction=1
  - -XshowSettings:vm
  - -jar
  - jmx_prometheus_httpserver.jar
  - "5556"
  - /etc/jmx-kafka/jmx-kafka-prometheus.yml
  ports:
  - name: prometheus
    containerPort: 5556
  resources:
    requests:
      cpu: 100m
      memory: 500Mi
  volumeMounts:
  - name: jmx-config
    mountPath: /etc/jmx-kafka
```

Kafka anti-affinity



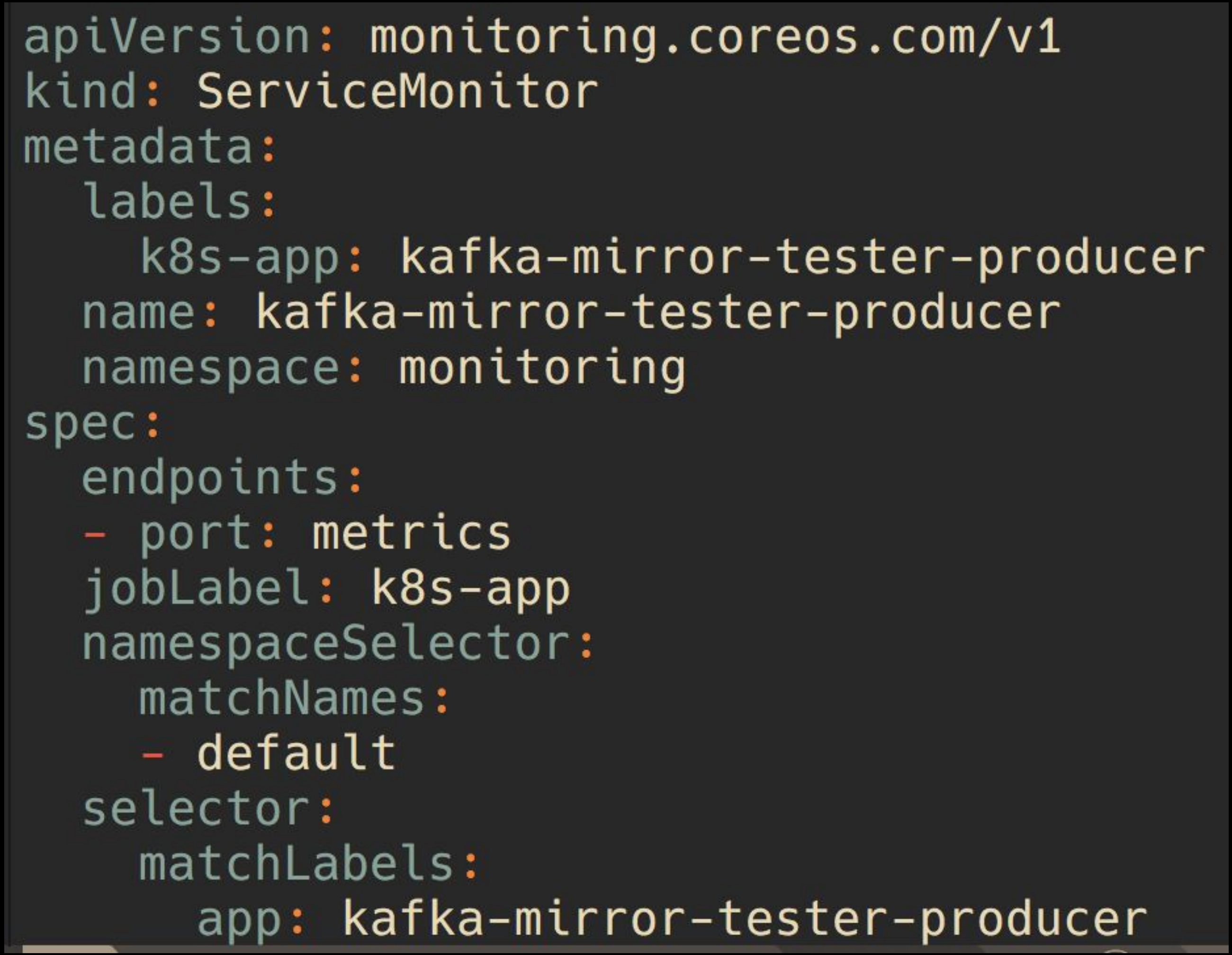
```
affinity:
  podAntiAffinity:
    requiredDuringSchedulingIgnoredDuringExecution:
    - labelSelector:
        matchExpressions:
        - key: app
          operator: In
          values:
          - kafka-source
```

uReplicator deployment



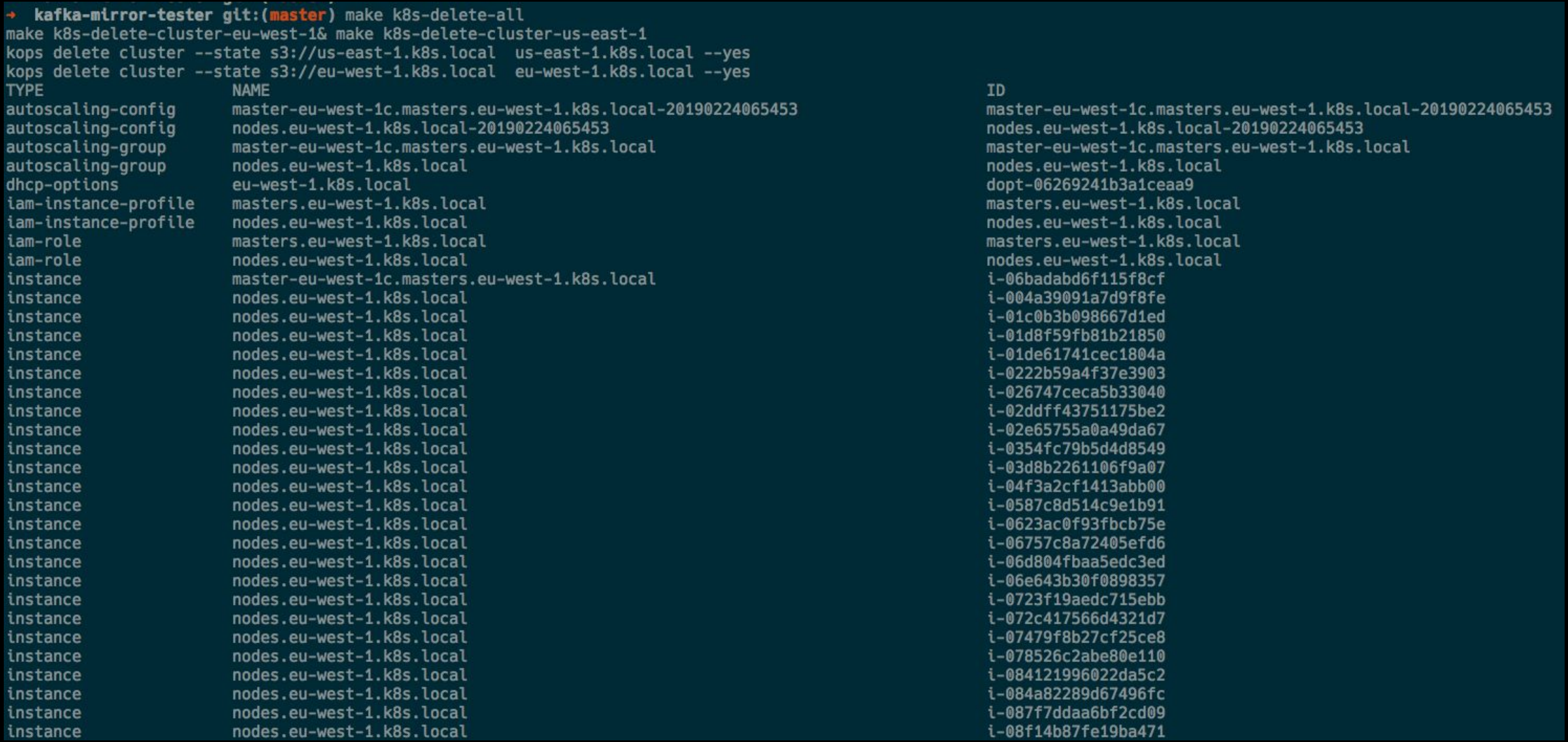
```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
  namespace: ureplicator
  name: ureplicator-worker
  labels:
    app: ureplicator
    component: worker
spec:
  replicas: 8
  selector:
    matchLabels:
      app: ureplicator
      component: worker
```

Service monitoring



```yaml
apiVersion: monitoring.coreos.com/v1
kind: ServiceMonitor
metadata:
  labels:
    k8s-app: kafka-mirror-tester-producer
  name: kafka-mirror-tester-producer
  namespace: monitoring
spec:
  endpoints:
  - port: metrics
  jobLabel: k8s-app
  namespaceSelector:
    matchNames:
    - default
  selector:
    matchLabels:
      app: kafka-mirror-tester-producer
```

```
$ make k8s-delete-all
```



```
→  kafka-mirror-tester git:(master) make k8s-delete-all
make k8s-delete-cluster-eu-west-1& make k8s-delete-cluster-us-east-1
kops delete cluster --state s3://us-east-1.k8s.local  us-east-1.k8s.local --yes
kops delete cluster --state s3://eu-west-1.k8s.local  eu-west-1.k8s.local --yes
TYPE                    NAME                                                                                    ID
autoscaling-config      master-eu-west-1c.masters.eu-west-1.k8s.local-20190224065453                            master-eu-west-1c.masters.eu-west-1.k8s.local-20190224065453
autoscaling-config      nodes.eu-west-1.k8s.local-20190224065453                                                nodes.eu-west-1.k8s.local-20190224065453
autoscaling-group       master-eu-west-1c.masters.eu-west-1.k8s.local                                           master-eu-west-1c.masters.eu-west-1.k8s.local
autoscaling-group       nodes.eu-west-1.k8s.local                                                               nodes.eu-west-1.k8s.local
dhcp-options            eu-west-1.k8s.local                                                                     dopt-06269241b3a1ceaa9
iam-instance-profile    masters.eu-west-1.k8s.local                                                             masters.eu-west-1.k8s.local
iam-instance-profile    nodes.eu-west-1.k8s.local                                                               nodes.eu-west-1.k8s.local
iam-role                masters.eu-west-1.k8s.local                                                             masters.eu-west-1.k8s.local
iam-role                nodes.eu-west-1.k8s.local                                                               nodes.eu-west-1.k8s.local
instance                master-eu-west-1c.masters.eu-west-1.k8s.local                                           i-06badabd6f115f8cf
instance                nodes.eu-west-1.k8s.local                                                               i-004a39091a7d9f8fe
instance                nodes.eu-west-1.k8s.local                                                               i-01c0b3b098667d1ed
instance                nodes.eu-west-1.k8s.local                                                               i-01d8f59fb81b21850
instance                nodes.eu-west-1.k8s.local                                                               i-01de61741cec1804a
instance                nodes.eu-west-1.k8s.local                                                               i-0222b59a4f37e3903
instance                nodes.eu-west-1.k8s.local                                                               i-026747ceca5b33040
instance                nodes.eu-west-1.k8s.local                                                               i-02ddff43751175be2
instance                nodes.eu-west-1.k8s.local                                                               i-02e65755a0a49da67
instance                nodes.eu-west-1.k8s.local                                                               i-0354fc79b5d4d8549
instance                nodes.eu-west-1.k8s.local                                                               i-03d8b2261106f9a07
instance                nodes.eu-west-1.k8s.local                                                               i-04f3a2cf1413abb00
instance                nodes.eu-west-1.k8s.local                                                               i-0587c8d514c9e1b91
instance                nodes.eu-west-1.k8s.local                                                               i-0623ac0f93fbcb75e
instance                nodes.eu-west-1.k8s.local                                                               i-06757c8a72405efd6
instance                nodes.eu-west-1.k8s.local                                                               i-06d804fbaa5edc3ed
instance                nodes.eu-west-1.k8s.local                                                               i-06e643b30f0898357
instance                nodes.eu-west-1.k8s.local                                                               i-0723f19aedc715ebb
instance                nodes.eu-west-1.k8s.local                                                               i-072c417566d4321d7
instance                nodes.eu-west-1.k8s.local                                                               i-07479f8b27cf25ce8
instance                nodes.eu-west-1.k8s.local                                                               i-078526c2abe80e110
instance                nodes.eu-west-1.k8s.local                                                               i-084121996022da5c2
instance                nodes.eu-west-1.k8s.local                                                               i-084a82289d67496fc
instance                nodes.eu-west-1.k8s.local                                                               i-087f7ddaa6bf2cd09
instance                nodes.eu-west-1.k8s.local                                                               i-08f14b87fe19ba471
```

Thanks!

Any questions?

You can find me at @rantav & rantav@appsflyer.com

This presentation: https://speakerdeck.com/rantav/infrastructure-testing-using-kubernetes-and-go

The project: https://github.com/AppsFlyer/kafka-mirror-tester