

Cloud-Native Legacy Applications

Robert Munteanu, Adobe

Slides revision: 20190922-478c9cc

APACHECON

EUROPE Oct. 22nd - 24th

2019

Welcome

About me



@rombert

Outline


- Welcome
- Cloud-native vs legacy
- Managing cloud-native applications
- Cloud-native transformation

Cloud-native vs legacy

Cloud-native applications



Defining cloud-native

-  Using cloud-native services
- Application-centric design
- Automation

*Cloud Native Architectures -
Kamal Arora, Erik Farr, Tom Laszewski, Piyum Zonooz*

Cloud-native services

- Logging: StackDriver, Centralised Logging, Logging and Auditing
- Routing: ELB, Azure Load Balancer, Cloud Load Balancing
- Block storage: Azure Blob Store, S3, Cloud Storage
- Databases: RDS, Cloud SQL, Azure SQL Database

Application-centric design



Automation

1. Building
2. Testing
3. Integration
4. Deployment
5. Monitoring
6. Capacity adjustment

A young boy with light brown hair, wearing blue sunglasses, a dark blue suit jacket over a white t-shirt, and blue jeans, is sitting in a black tufted chair. He is holding a large stack of various banknotes, including US dollars and Euro banknotes, in his right hand. The background is a textured, grey wall. The text "Legacy for everyone" is overlaid in white, bold, sans-serif font across the middle of the image.

Legacy for everyone

A wide-angle photograph of a massive landfill of plastic waste. The foreground and middle ground are filled with a sea of discarded plastic items, including bottles, containers, and fragments, stretching towards a horizon under a dramatic sunset sky with golden light and scattered clouds. The text "Legacy for programmers" is overlaid in the center of the image.

Legacy for programmers

Legacy applications

legacy /ˈlɛɡəsi/

1. an amount of money or property left to someone in a will.
2. software or hardware that has been superseded but is difficult to replace because of its wide use

<https://www.lexico.com/en/definition/legacy>

Managing cloud-native applications

Containers, container, containers

```
FROM openjdk:8-jre-alpine
MAINTAINER dev@sling.apache.org

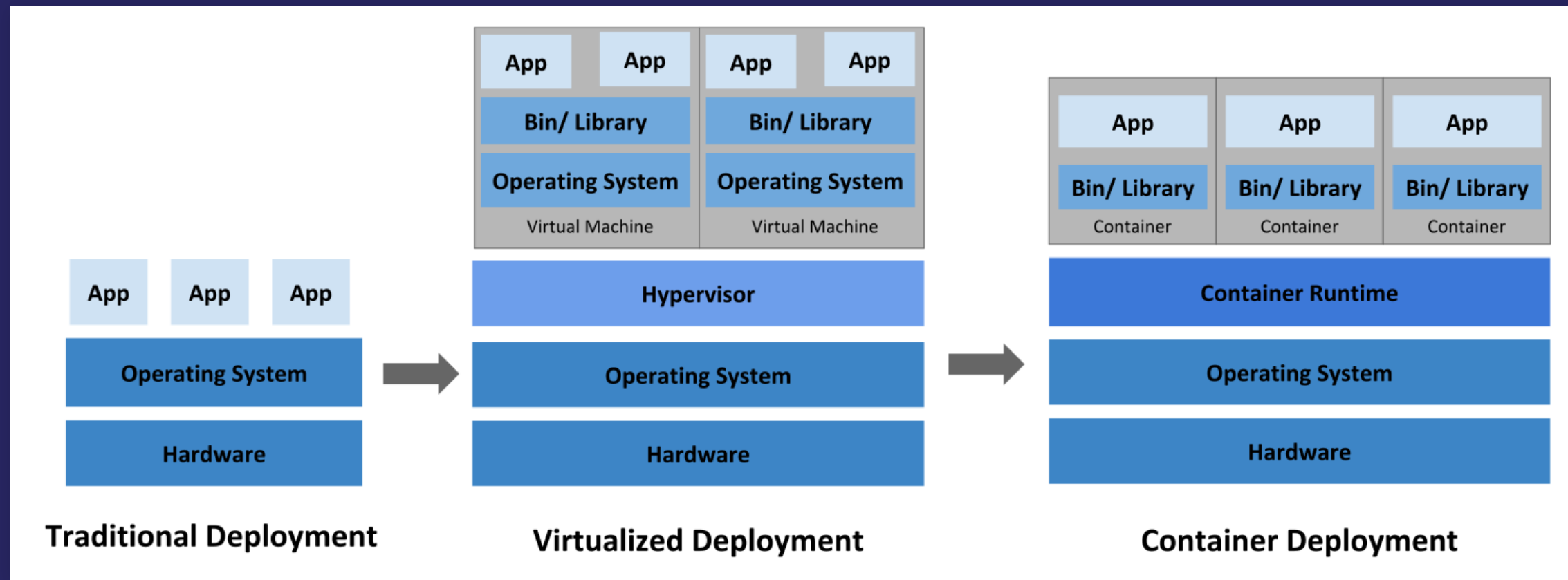
RUN mkdir -p /opt/sling
COPY target/sling-cloud-ready-*.jar /opt/sling/sling.jar

WORKDIR /opt/sling
EXPOSE 8080
VOLUME /opt/sling/sling

ENV JAVA_OPTS -Xmx512m
ENV SLING_OPTS ''

CMD exec java $JAVA_OPTS -jar sling.jar $SLING_OPTS
```

Kubernetes



Kubernetes

```
apiVersion: v1
kind: Pod
metadata:
  name: nginx-pod
  labels:
    name: wildfly-pod
spec:
  containers:
  - name: nginx
    image: nginx:1.17.1
    ports:
    - containerPort: 80
```

Kubernetes interface

```
$ kubectl get pods
```

NAME	READY	STATUS	AGE
grafana-78c6877b7f-ghmwp	1/1	Running	142m
prometheus-alertmanager-5cfb66f7f7-mtrx2	2/2	Running	18d
prometheus-kube-state-metrics-57d85676c5-bvh5j	1/1	Running	18d
prometheus-node-exporter-5pdnf	1/1	Running	18d
prometheus-node-exporter-6qf72	1/1	Running	18d
prometheus-node-exporter-h7bjx	1/1	Running	18d
prometheus-node-exporter-wrxfl	1/1	Running	18d
prometheus-pushgateway-86cf78b9f4-4m4fc	1/1	Running	18d
prometheus-server-747fc94b5d-fx5qf	2/2	Running	18d

What does Kubernetes manage?

- Applications (Pods, Services, Deployments)
- DNS services
- Volume management (NFS, Ceph, Azure/GCE disks, AWS EBS, etc)
- Software-defined networking
- Resource quotas (CPU, Memory, etc)
- Rollouts, rollbacks and scaling
- Secrets

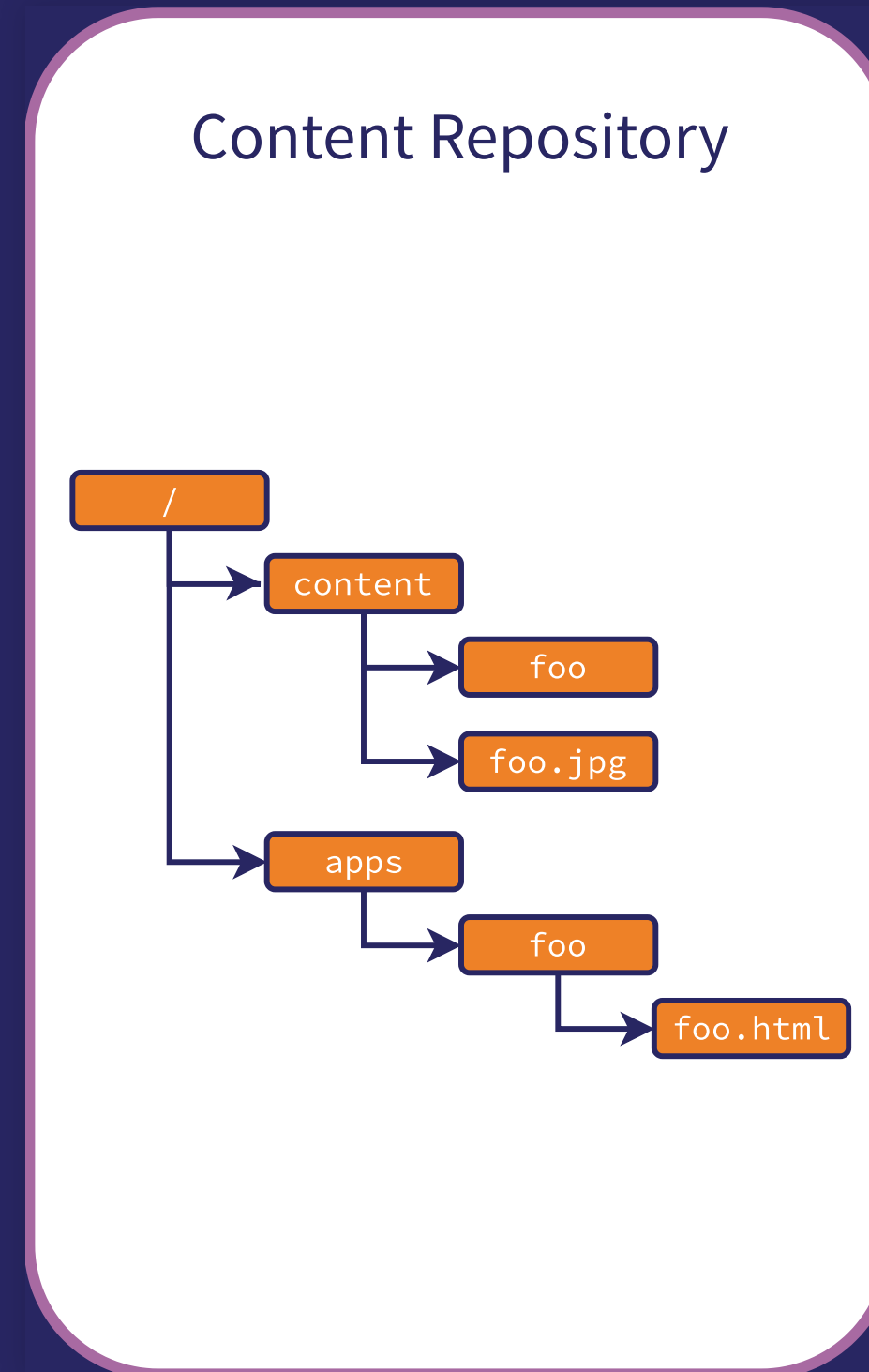
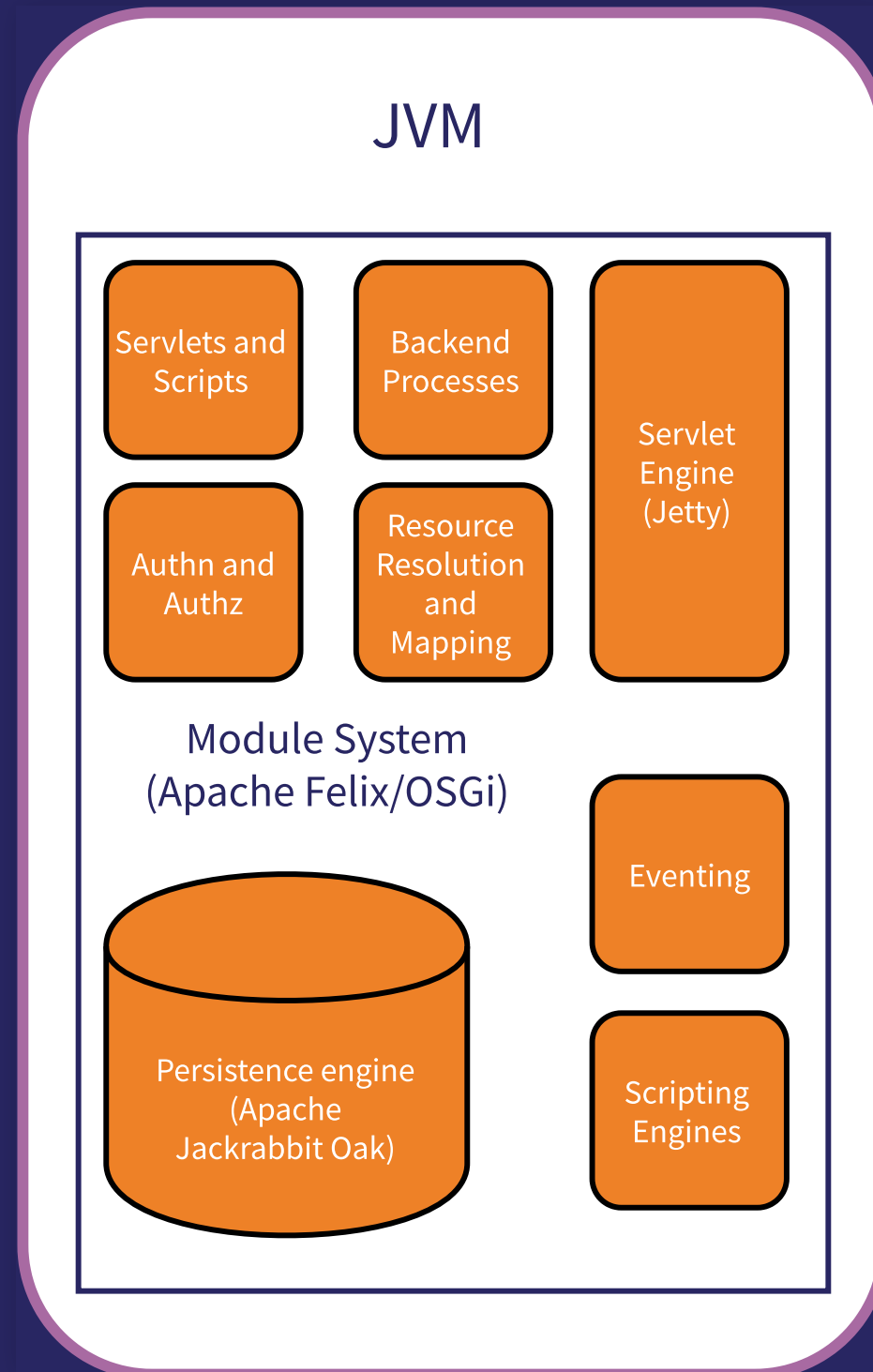
Cloud-native transformation

Apache Sling in 2 slides (1/2)

Apache Sling™ is a framework for RESTful web-applications based on an extensible content tree.

- maps (...) requests to content resources (...)
- requests are processed by scripts and servlets , dynamically selected based on the current resource
- (...) modular nature (...)

Apache Sling in 2 slides (2/2)



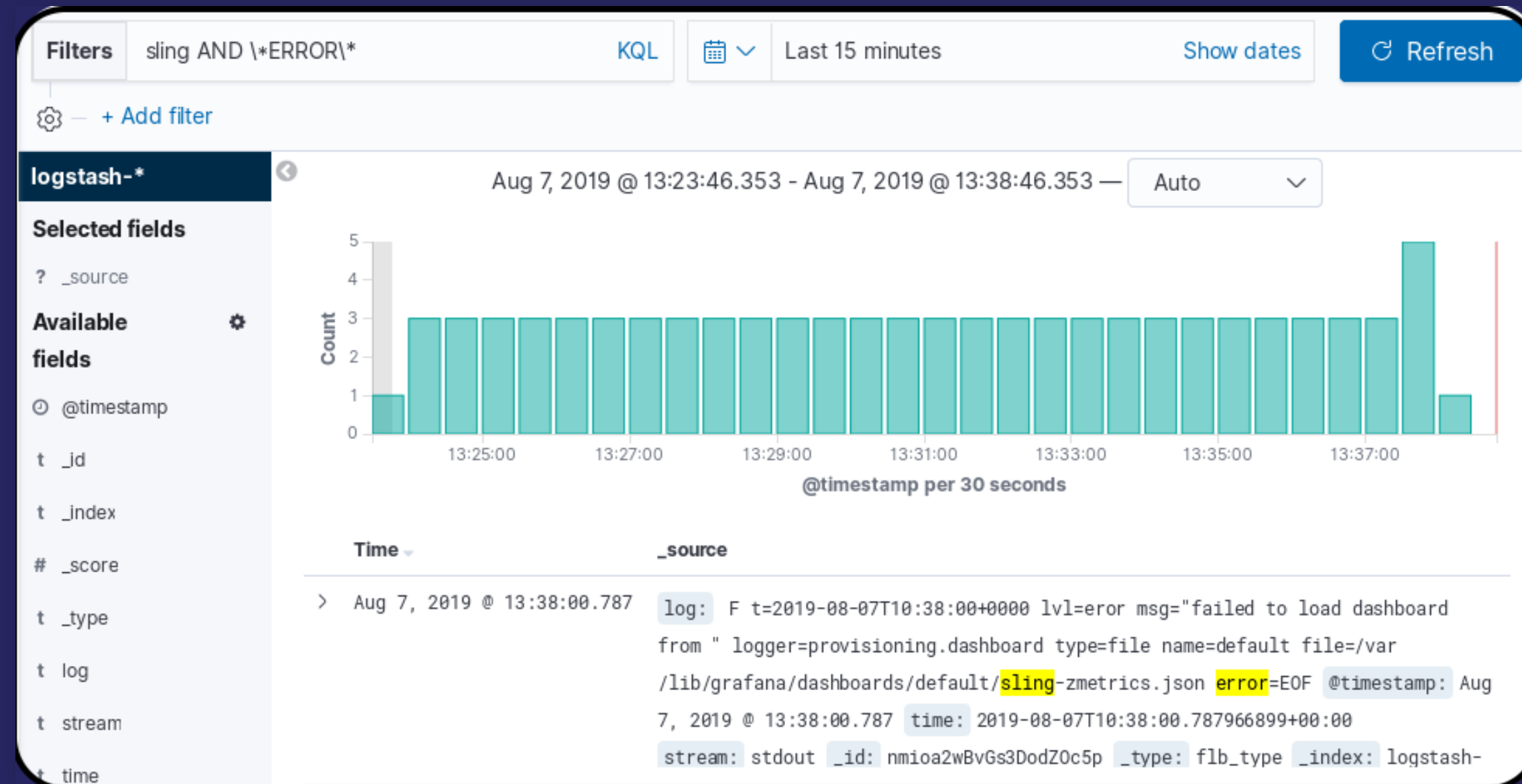
Logging

```
[configurations]
org.apache.sling.commons.log.LogManager
org.apache.sling.commons.log.pattern=↵
    "%level*\ [%thread]\ %logger\ %msg%n"
org.apache.sling.commons.log.file=""
org.apache.sling.commons.log.level="info"
org.apache.sling.commons.log.packagingDataEnabled=B"true"
```

Logging

```
$ kubectl logs -f sling
07.08.2019 10:50:45.212 *INFO* [FelixStartLevel] ↵
    org.apache.sling.commons.logservice BundleEvent STARTED
07.08.2019 10:50:45.216 *INFO* [FelixStartLevel] ↵
    org.apache.geronimo.specs.geronimo-atinject_1.0_spec BundleEvent ↵
    RESOLVED
07.08.2019 10:50:45.216 *INFO* [FelixStartLevel] ↵
    org.apache.geronimo.specs.geronimo-atinject_1.0_spec BundleEvent ↵
    STARTING
07.08.2019 10:50:45.219 *INFO* [FelixStartLevel] ↵
    org.apache.geronimo.specs.geronimo-atinject_1.0_spec BundleEvent ↵
    STARTED
07.08.2019 10:50:45.221 *INFO* [FelixStartLevel] ↵
    org.apache.felix.eventadmin BundleEvent RESOLVED
07.08.2019 10:50:45.225 *INFO* [FelixStartLevel] ↵
    org.apache.felix.eventadmin BundleEvent STARTING
```

Logging



Startup readiness

```
readinessProbe:  
  httpGet:  
    path: /  
    port: 8080  
  initialDelaySeconds: 15  
  periodSeconds: 5
```

Startup readiness

```
$ kubectl get pods -l=app=sling
NAME                                READY   STATUS    AGE
sling-59c6d6c656-857vq             0/1    Running   23s
sling-59c6d6c656-9wlzm             0/1    Running   23s
sling-59c6d6c656-cbbwc             1/1    Running   13m
```

Persistence

```
env:  
- name: SLING_OPTS  
  value: -Dsling.run.modes=oak_mongo"
```

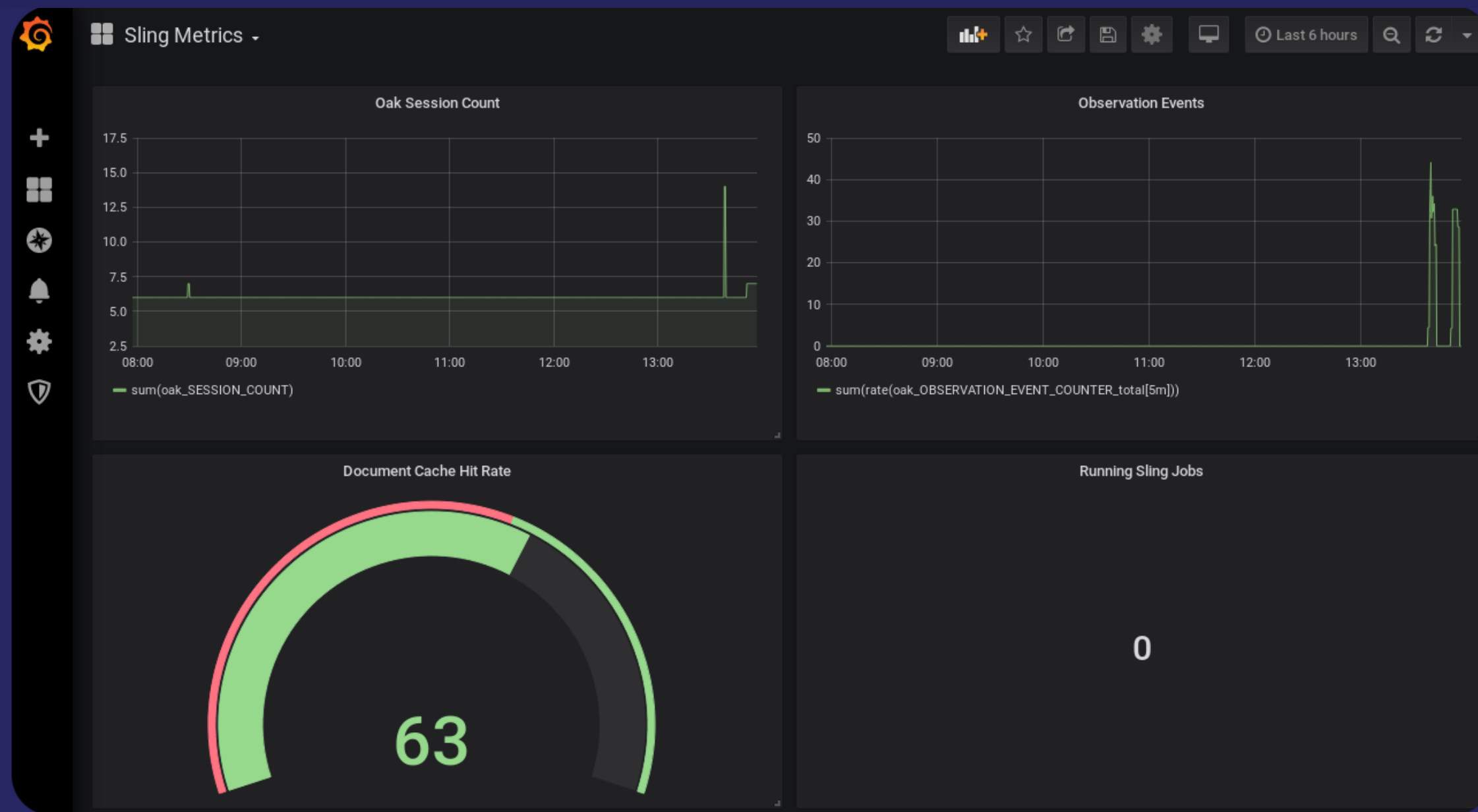

Persistence

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: sling-datastore
spec:
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 10Gi
```

Monitoring

```
annotations:  
  prometheus.io/scrape: "true"  
  prometheus.io/port: "8080"
```

Monitoring



Persistence

```
$ kubectl get svc mongo
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)
mongo	ClusterIP	10.102.248.138	<none>	27017/TCP

Scalability

```
- replicas: 1  
+ replicas: 3
```

Performance

```
org.apache.sling.dynamicinclude.Configuration-starter
include-filter.config.enabled=B"true"
include-filter.config.resource-types=↵
  ["sling/starter/hostname"]
include-filter.config.path="/content/starter"
include-filter.config.include-type="SSI"
include-filter.config.add_comment=B
```

Performance

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: ingress-sling
  annotations:
    # use the shared ingress-nginx
    kubernetes.io/ingress.class: "nginx"
    # set sticky sessions via cookie
    nginx.ingress.kubernetes.io/affinity: "cookie"
    nginx.ingress.kubernetes.io/configuration-snippet: |
      ssi on;
```

Performance

```
<!-- SDI include (↵  
  path: /content/starter/sidebar-extensions/↵  
    hostname.nocache.html, ↵  
  resourceType: sling/starter/hostname) -->  
<h2>System information</h2>  
<p id="system-information">Currently running on host ↵  
  <span class="highlight">10.47.0.34</span></p>
```


All good things...

Resources

- <https://sling.apache.org>
- <https://kubernetes.io/>
- <https://github.com/rombert/sling-cloud-native>