How to develop your first cloud-native Applications with Java

Niklas Heidloff
Developer Advocate, IBM
@nheidloff

Harald Uebele
Developer Advocate, IBM
@Harald_U

IBM **Developer**

IBM

What are cloud-native Applications?

Horizontal scalability

→ Distributed systems

Elasticity

→ App stays responsive

Continues delivery

→ DevOps

Polyglot microservices

→ Interactions via APIs

“Microservices are a software development technique [...] that structures an application as a collection of loosely coupled services.”

Wikipedia

When to use Microservices?

Full control

→ Canary rollouts

Fast response times

→ No cold start

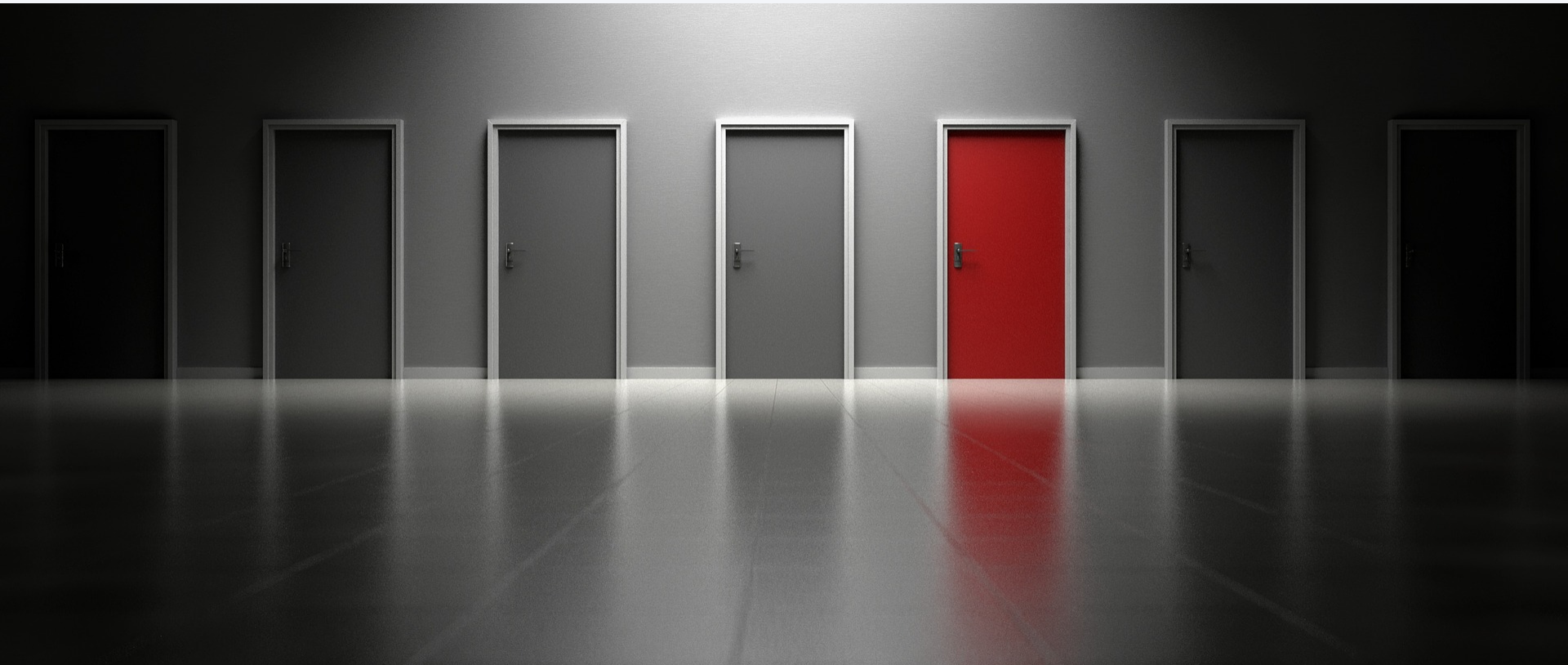
Run anywhere

→ Kubernetes

Empowered teams

→ Ownership

New Options → New Challenges



Cloud-native applications provide new capabilities, which challenge developers

- New tasks
- Old tasks in new context

Service discovery and traffic management

Distributed monitoring, logging and tracing

Authentication and authorization between services

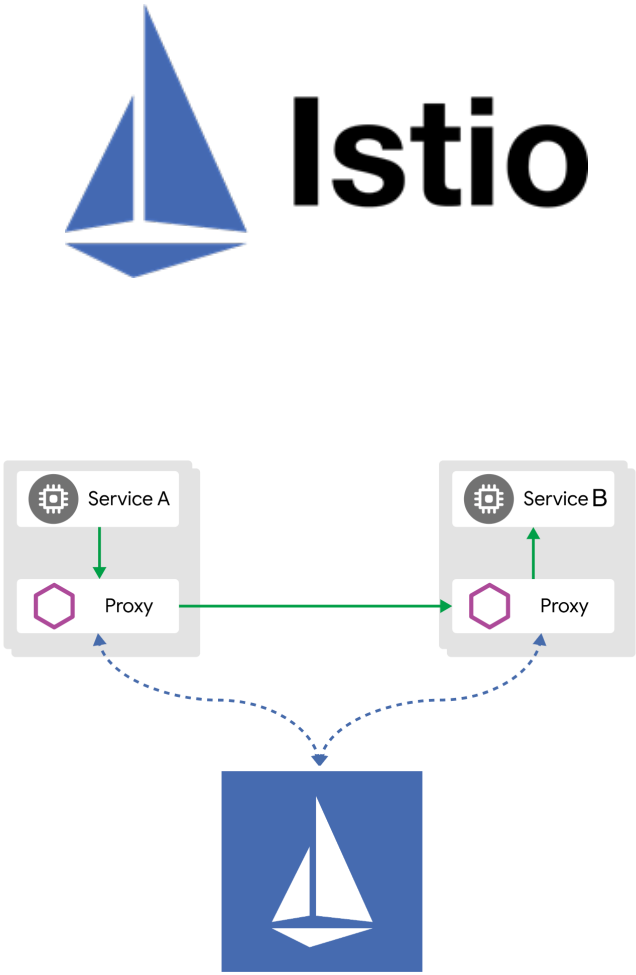
“Kubernetes (K8s) is an open-source system for automating deployment, scaling, and management of containerized applications.”

kubernetes.io

kubernetes

“Istio is an open platform for providing a uniform way to integrate microservices, manage traffic flow across microservices, enforce policies and aggregate telemetry data.”

`github.com/istio/istio`

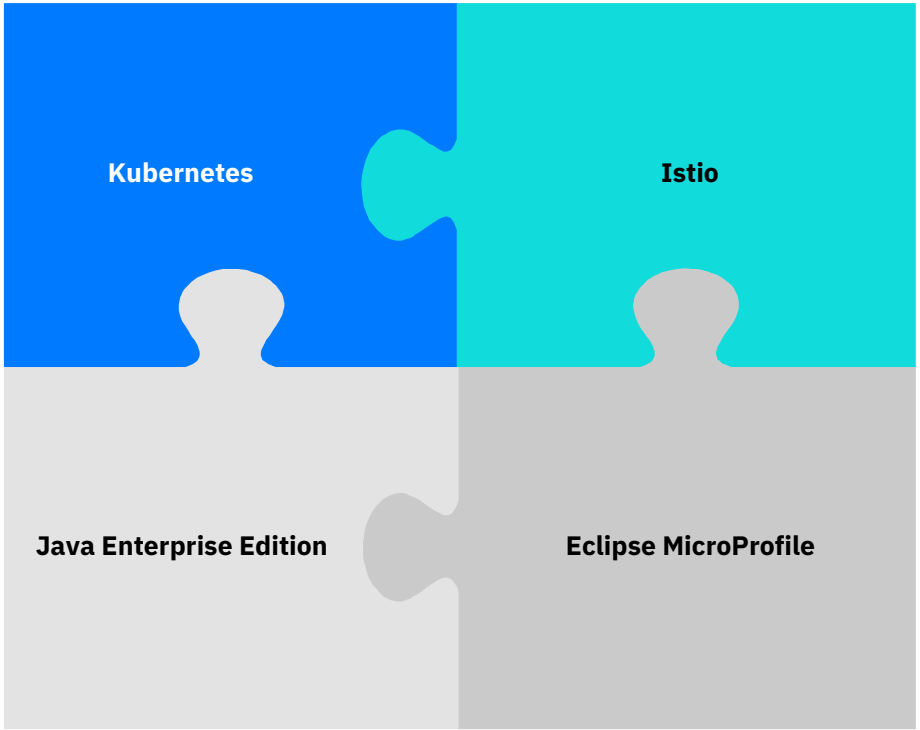


“Optimizing Enterprise Java for a Microservices Architecture.

[...] by innovating [...] with a goal of standardization.”

`microprofile.io`

How to use all Pieces together?



Leverage platforms as much as possible.

Use frameworks for app specific logic.

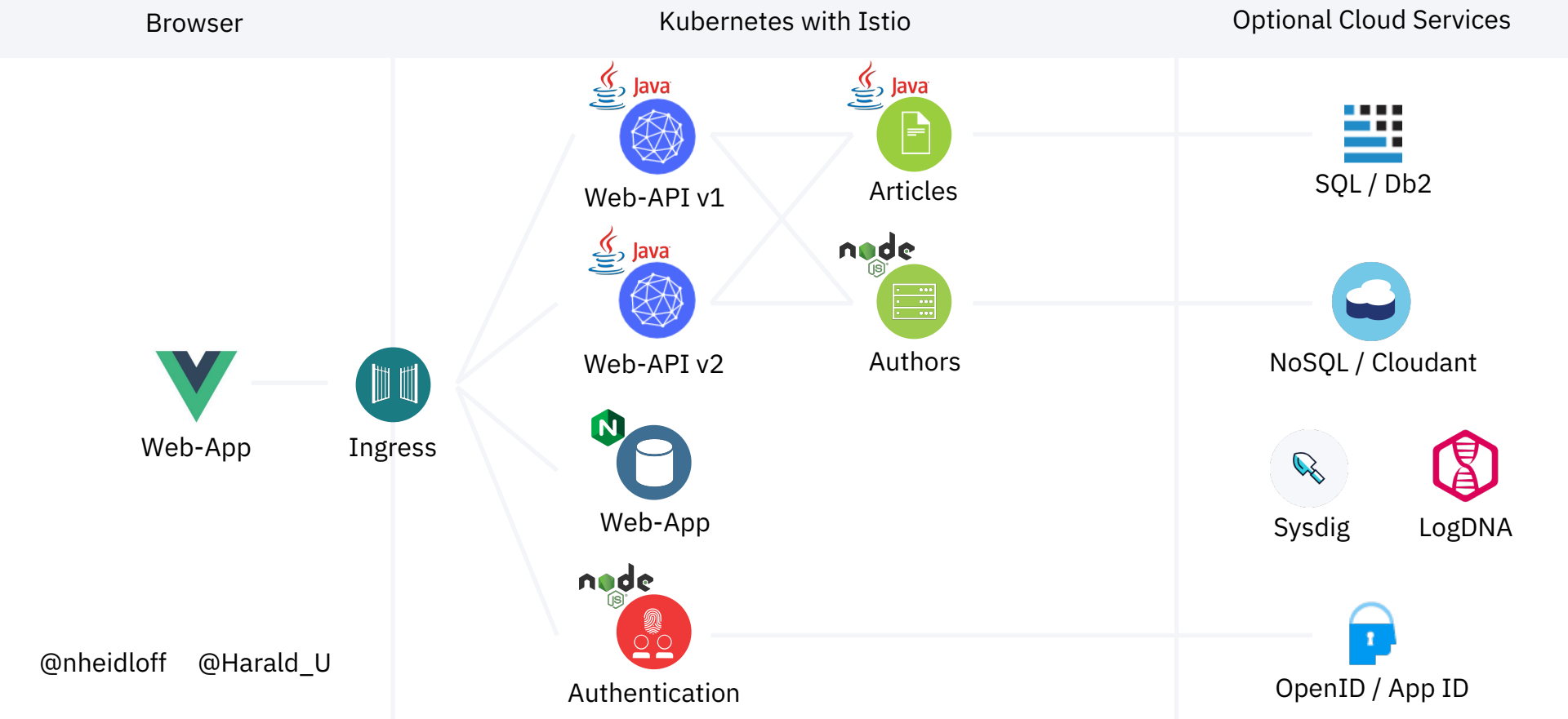
Design principles for the end-to-end example ‘cloud-native-starter’

Use only open-source components for the core services of the application

Make the first time experience as simple as possible

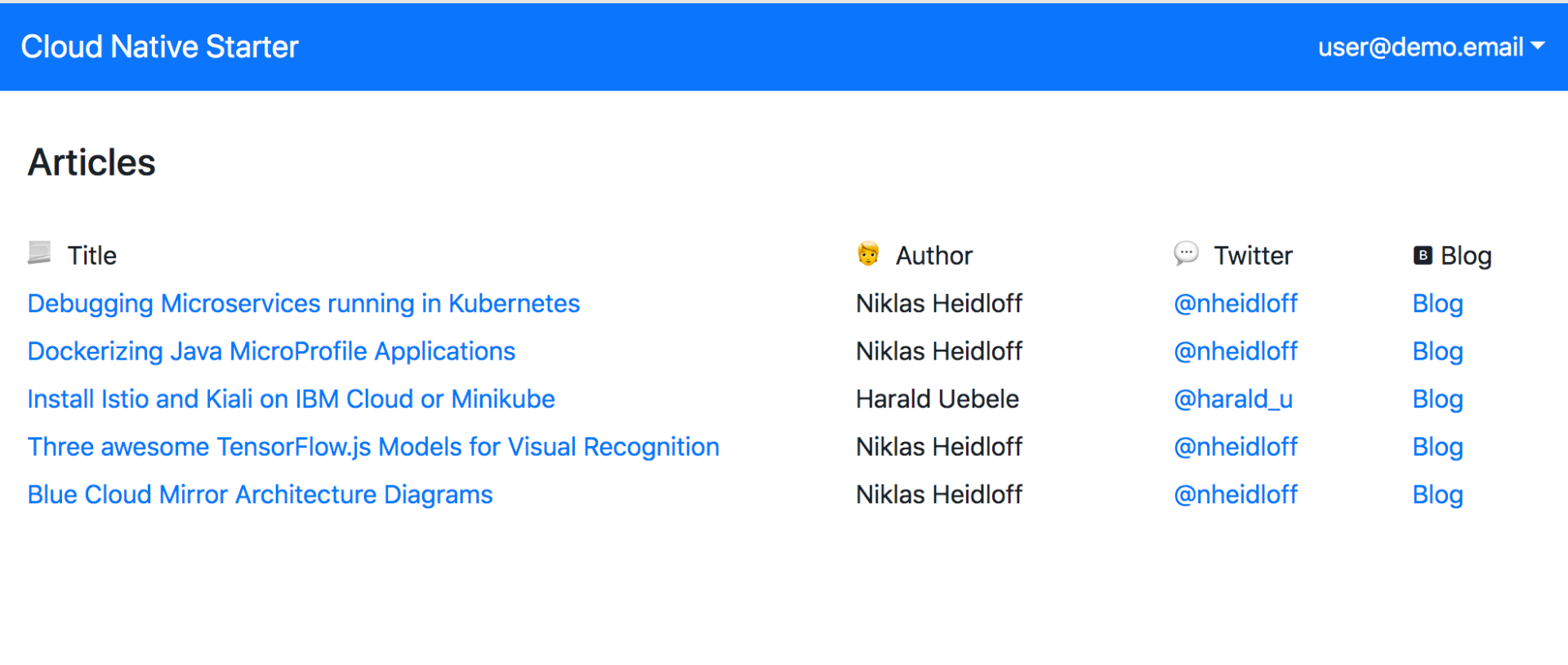
Be able to run the application in different environments

Architecture: End-to-End Example ‘cloud-native-starter’



@nheidloff @Harald_U

Web Experience in ‘cloud-native-starter’



Java Image

Open source stack

OpenJ9 0.12.1

OpenJDK 8u202-b08 from AdoptOpenJDK

Open Liberty 18.0.0.4

MicroProfile 2.1

Dockerfile

```
FROM maven:3.5-jdk-8 as BUILD

COPY src /usr/src/app/src
COPY pom.xml /usr/src/app
RUN mvn -f /usr/src/app/pom.xml clean package


FROM openliberty/open-liberty:microProfile2-java8-openj9

ADD liberty-opentracing-zipkintracer-1.2-sample.zip /

RUN unzip liberty-opentracing-zipkintracer-1.2-sample.zip -d /opt/ol/wlp/usr/ \
 && rm liberty-opentracing-zipkintracer-1.2-sample.zip

COPY liberty/server.xml /config/

COPY --from=BUILD /usr/src/app/target/articles.war /config/dropins/
```

Exposing REST APIs

JAX-RS

Java API for RESTful Web Services

@nheidloff @Harald_U

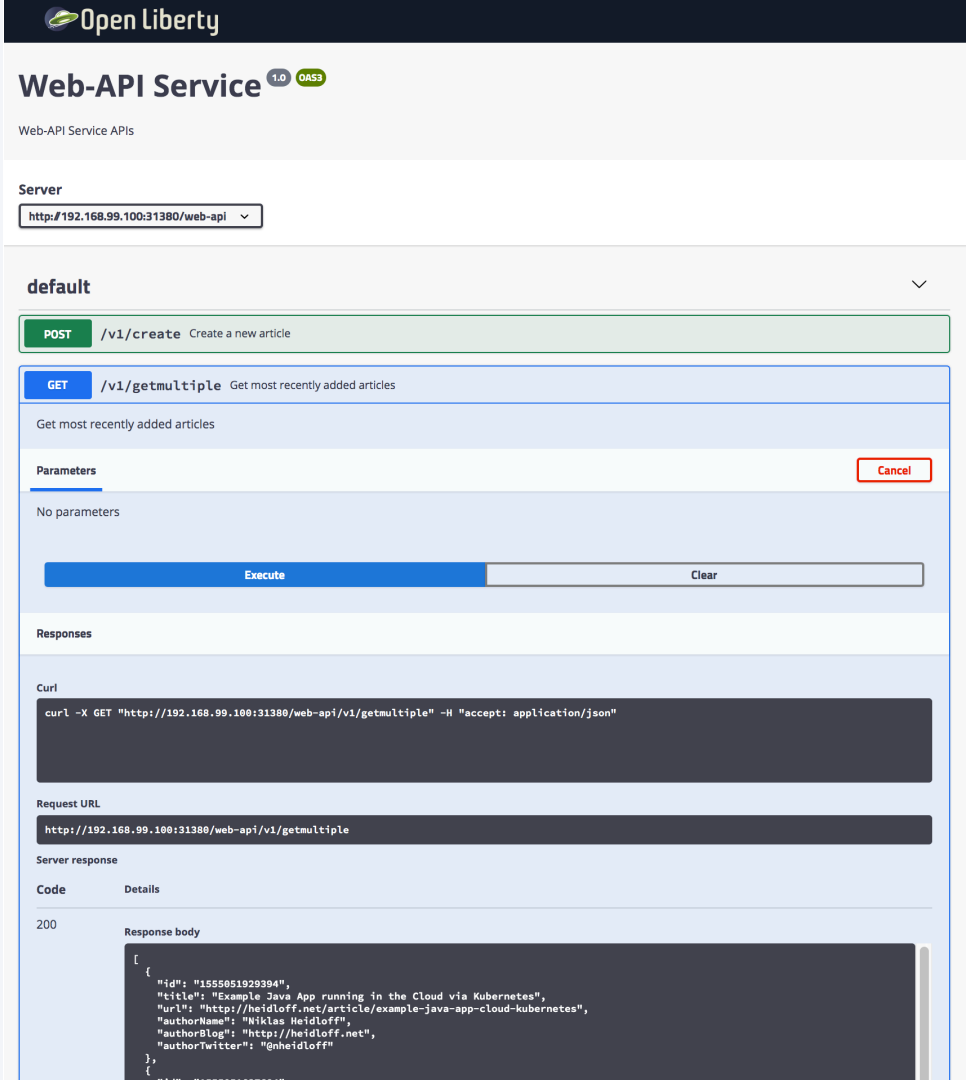
GetArticles.java

```
@RequestScoped
@Path("/v1")
@OpenAPIDefinition(info = @Info(title = "Web-API Service",
  version = "1.0", description = "Web-API Service APIs"))
public class GetArticles {

  @Inject
  com.ibm.webapi.business.Service service;

  @GET
  @Path("/getmultiple")
  @Produces(MediaType.APPLICATION_JSON)
  @APIResponses(value = {
      @APIResponse(responseCode = "200",
        description = "Get most recently added articles",
        content = @Content(mediaType = "application/json",
        schema = @Schema(type = SchemaType.ARRAY,
          implementation = Article.class))),
      @APIResponse(responseCode = "500",
        description = "Internal service error") })
  @Operation(summary = "Get most recently added articles",
    description = "Get most recently added articles")
  public Response getArticles() {
```

Exposing REST APIs

Open API (formerly Swagger)

API description format for REST APIs



@nheidloff @Harald_U

Consuming REST APIs

MicroProfile Rest Client

Type-safe approach to invoke RESTful services

@nheidloff @Harald_U

AuthorsService.java

```
@RegisterProvider(ExceptionMapperArticles.class)
public interface AuthorsService {

  @GET
  @Produces(MediaType.APPLICATION_JSON)
  public Author getAuthor(String name) throws NonexistentAuthor;
}
```

AuthorsServiceDataAccess.java

```
public class AuthorsServiceDataAccess implements AuthorsDataAccess {
  public AuthorsServiceDataAccess() {}

  static final String BASE_URL = "http://authors:3000/api/v1/";

  public Author getAuthor(String name) throws NoConnectivity, NonexistentAuthor {
    try {
      name = URLEncoder.encode(name, "UTF-8").replace("+", "%20");
      URL apiUrl = new URL(BASE_URL + "getauthor?name=" + name);
      AuthorsService customRestClient;
      customRestClient = RestClientBuilder.newBuilder().baseUrl(apiUrl)
          .register(ExceptionMapperAuthors.class).build(AuthorsService.class);

      Author output = customRestClient.getAuthor(name);
      return output;
    } catch (NonexistentAuthor e) {
      e.printStackTrace();
      throw new NonexistentAuthor(e);
    } catch (Exception e) {
      throw new NoConnectivity(e);
```

Resiliency

Timeouts and retries

Circuit breaking

Fallback

Fault injection

@nheidloff @Harald_U

Service.java

```
private List<Article> lastReadArticles;

public List<Article> fallbackNoArticlesService() {
  return lastReadArticles;
}

@Fallback(fallbackMethod = "fallbackNoArticlesService")
public List<Article> getArticles() throws NoDataAccess {

  List<Article> articles = new ArrayList<Article>();
  List<CoreArticle> coreArticles = new ArrayList<CoreArticle>();

  try {
    coreArticles = DataAccessManager.getArticlesDataAccess().getArticles(5);
  } catch (NoConnectivity e) {
    throw new NoDataAccess(e);
  }

  for (int index = 0; index < coreArticles.size(); index++) {
    CoreArticle coreArticle = coreArticles.get(index);
    Article article = new Article(coreArticle.id, coreArticle.title,
    coreArticle.title, coreArticle.author);
    try {
      Author author;
      author = DataAccessManager.getAuthorsDataAccess().getAuthor(coreArticle.author);
      article.authorBlog = author.blog;
      article.authorTwitter = author.twitter;
    } catch (Exception e) {
      article.authorBlog = "";
      article.authorTwitter = "";
    }
    articles.add(article);
  }
```

Resiliency

Cloud Native Starter user@demo.email

Articles

Title	Author	Twitter	Blog
Debugging Microservices running in Kubernetes	Niklas Heidloff		
Dockerizing Java MicroProfile Applications	Niklas Heidloff		
Install Istio and Kiali on IBM Cloud or Minikube	Harald Uebele		
Three awesome TensorFlow.js Models for Visual Recognition	Niklas Heidloff		
Blue Cloud Mirror Architecture Diagrams	Niklas Heidloff		

Traffic Management

Blue-green deployments

Canary rollouts

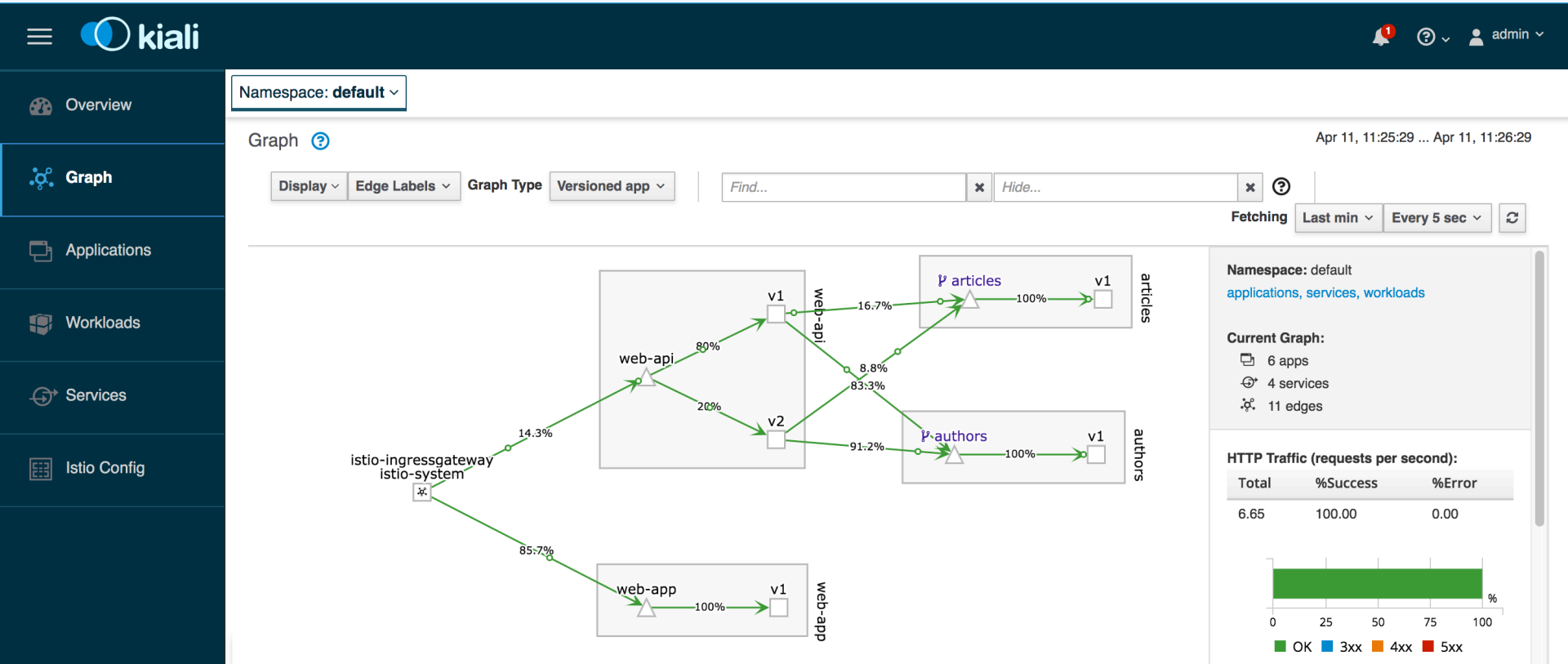
A/B testing

@nheidloff @Harald_U

ingress.yaml

```yaml
apiVersion: networking.istio.io/v1alpha3
kind: Gateway
metadata:
  name: default-gateway-ingress-http
spec:
  selector:
    istio: ingressgateway
  servers:
  - port:
      number: 80
      name: http
      protocol: HTTP
    hosts:
    - "*"
---
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
spec:
  hosts:
  - "*"
  gateways:
  - default-gateway-ingress-http
  http:
  - match:
    - uri:
        prefix: /web-api/v1/getmultiple
    route:
      - destination:
          host: web-api
          subset: v1
        weight: 80
      - destination:
          host: web-api
          subset: v2
        weight: 20
```

Traffic Management



Authentication and Authorization

OpenID Connect

Identity layer on top of the OAuth 2.0 protocol

IBM App ID

IBM service to authenticate users and protect APIs

policy.yaml

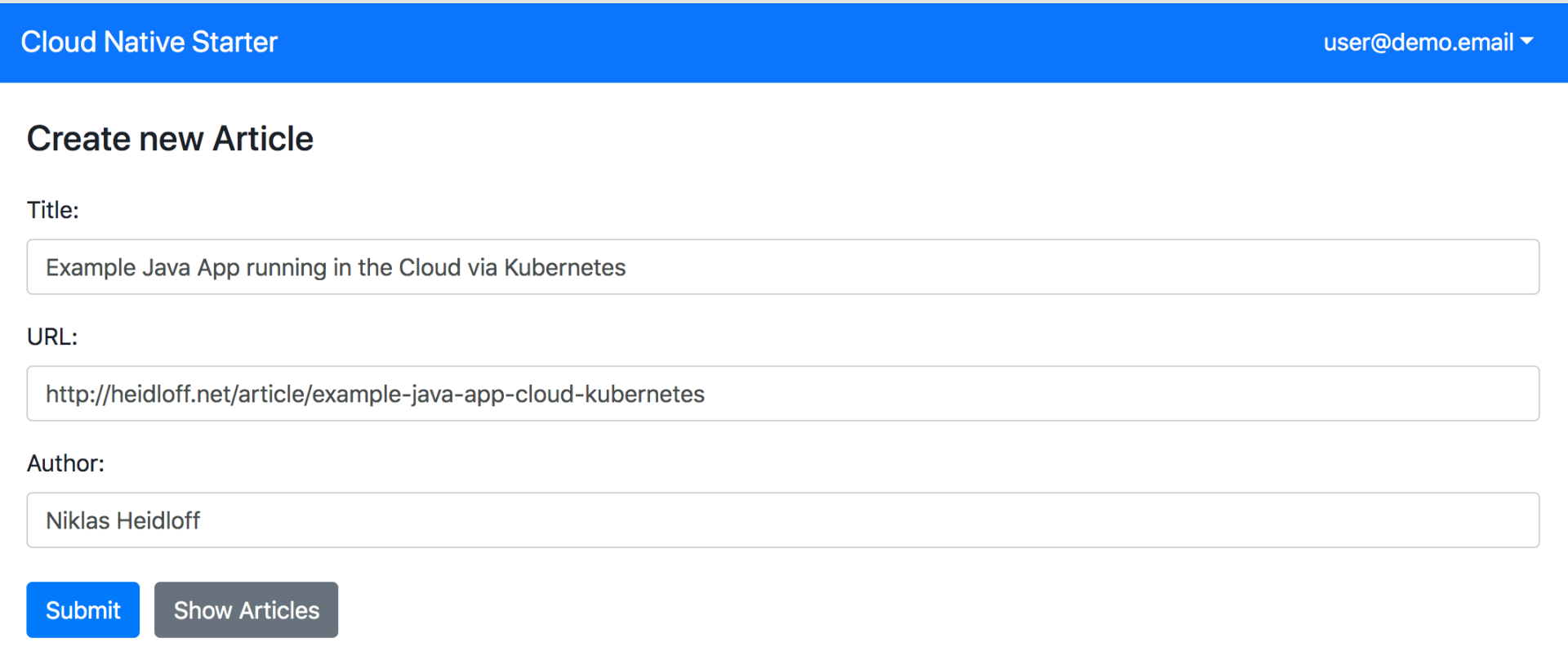
```yaml
apiVersion: "authentication.istio.io/v1alpha1"
kind: "Policy"
metadata:
  name: "protect-web-api"
spec:
  targets:
  - name: web-api
  origins:
  - jwt:
      issuer: "https://us-south.appid.cloud.ibm.com/oauth/v4/xxx"
      jwksUri: "https://us-south.appid.cloud.ibm.com/oauth/v4/xxx/publickeys"
      trigger_rules:
      - included_paths:
        - exact: /web-api/v1/create
  principalBinding: USE_ORIGIN
```

.env of authentication service

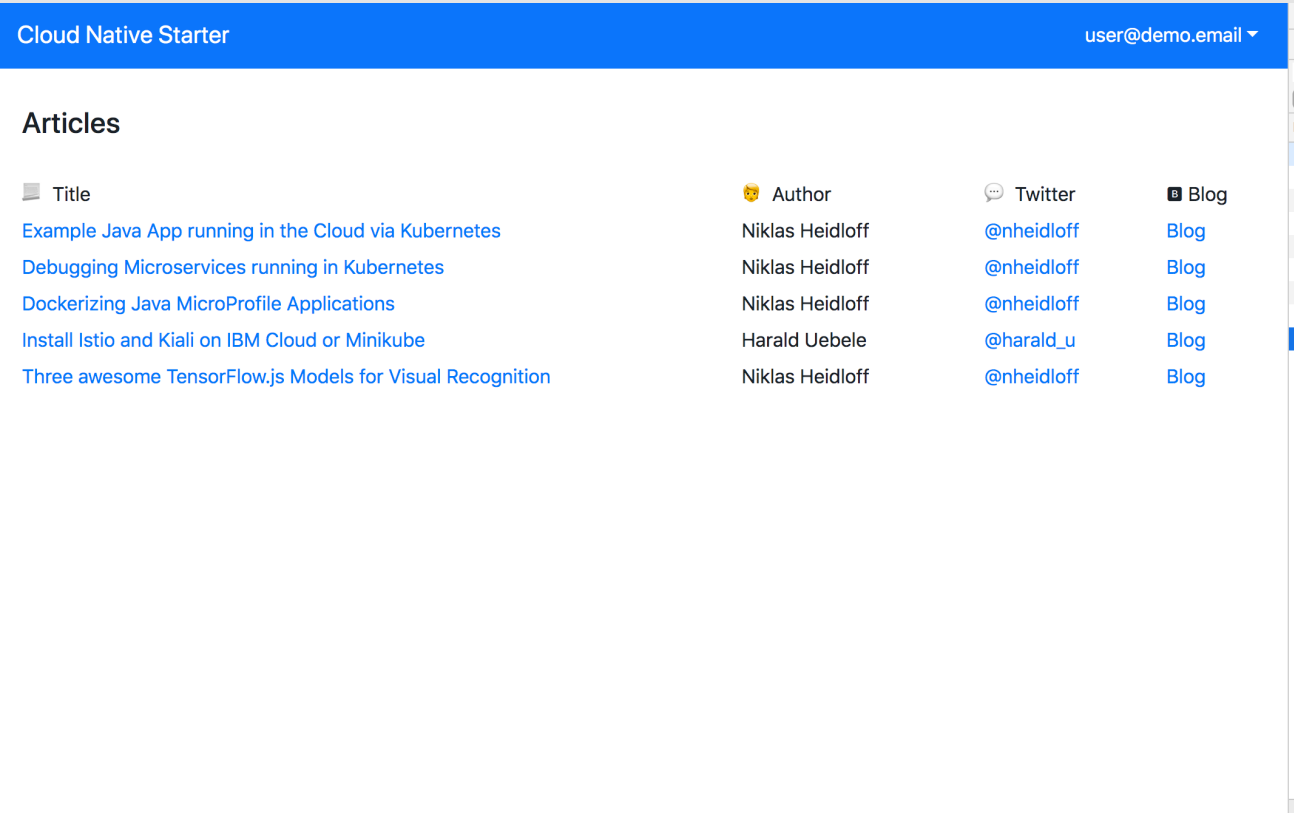
```
APPID_ISSUER=https://us-south.appid.cloud.ibm.com/oauth/v4/xxx
APPID_OPENID_CONFIG=https://us-south.appid.cloud.ibm.com/oauth/v4/xxx/.well-known/openid-configuration
APPID_AUTHORIZATION_ENDPOINT=https://us-south.appid.cloud.ibm.com/oauth/v4/xxx/authorization
APPID_TOKEN_ENDPOINT=https://us-south.appid.cloud.ibm.com/oauth/v4/xxx/token
APPID_USERINFO_ENDPOINT=https://us-south.appid.cloud.ibm.com/oauth/v4/xxx/userinfo
APPID_JWKS_URI=https://us-south.appid.cloud.ibm.com/oauth/v4/xxx/publickeys
APPID_CLIENTID=ee961476-e6c0-435d-95e4-xxx
APPID_SECRET=MTAxZjg2N2ItMmY1Mi00ZGIzLTxxx
APPID_MGMTURL=https://us-south.appid.cloud.ibm.com/management/v4/xxx
REDIRECT_URL_CALLBACK=http://192.168.99.100:31380/callback
REDIRECT_URL_WEB_APP=http://192.168.99.100:31380/loginwithtoken
```

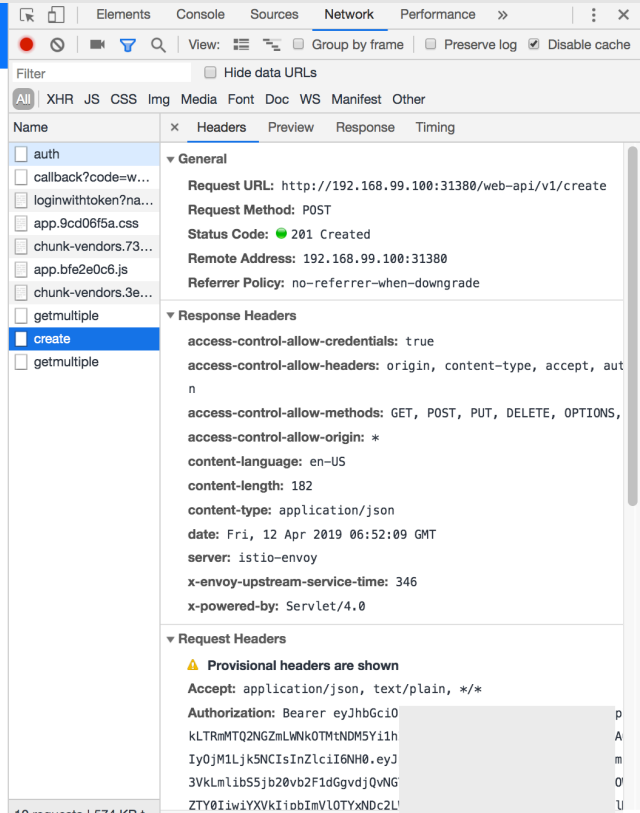
@nheidloff @Harald_U

Authorization

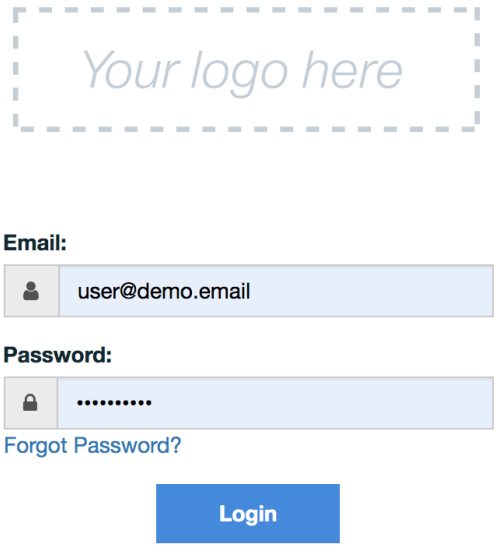


Authorization





Authentication



Configuration

MicroProfile Configuration

External configuration of microservices

@nheidloff @Harald_U

Service.yaml

```yaml
kind: Deployment
spec:
  template:
    spec:
      containers:
      - name: articles
        ...
        env:
        - name: samplescreation
          valueFrom:
            configMapKeyRef:
              name: articles-config
              key: samplescreation
        restartPolicy: Always
---
kind: ConfigMap
apiVersion: v1
metadata:
  name: articles-config
data:
  samplescreation: CREATE
```

CoreService.java

```java
@ApplicationScoped
public class CoreService {

  private static final String CREATE_SAMPLES = "CREATE";

  @Inject
  @ConfigProperty(name = "samplescreation", defaultValue = "dontcreate")
  private String samplescreation;

  @PostConstruct
  private void addArticle() {
    if (samplescreation.equalsIgnoreCase(CREATE_SAMPLES))
      addSampleArticles();
  }
```

Observability

Tracing

Logging

Monitoring

Metrics

Healthchecks

Microservices vs monolith

→ Higher complexity

→ Ephemeral

Chained invocations

Kubernetes

→ 1 service = N pods

Tracing

OpenTracing

Vendor-neutral APIs and instrumentation for distributed tracing

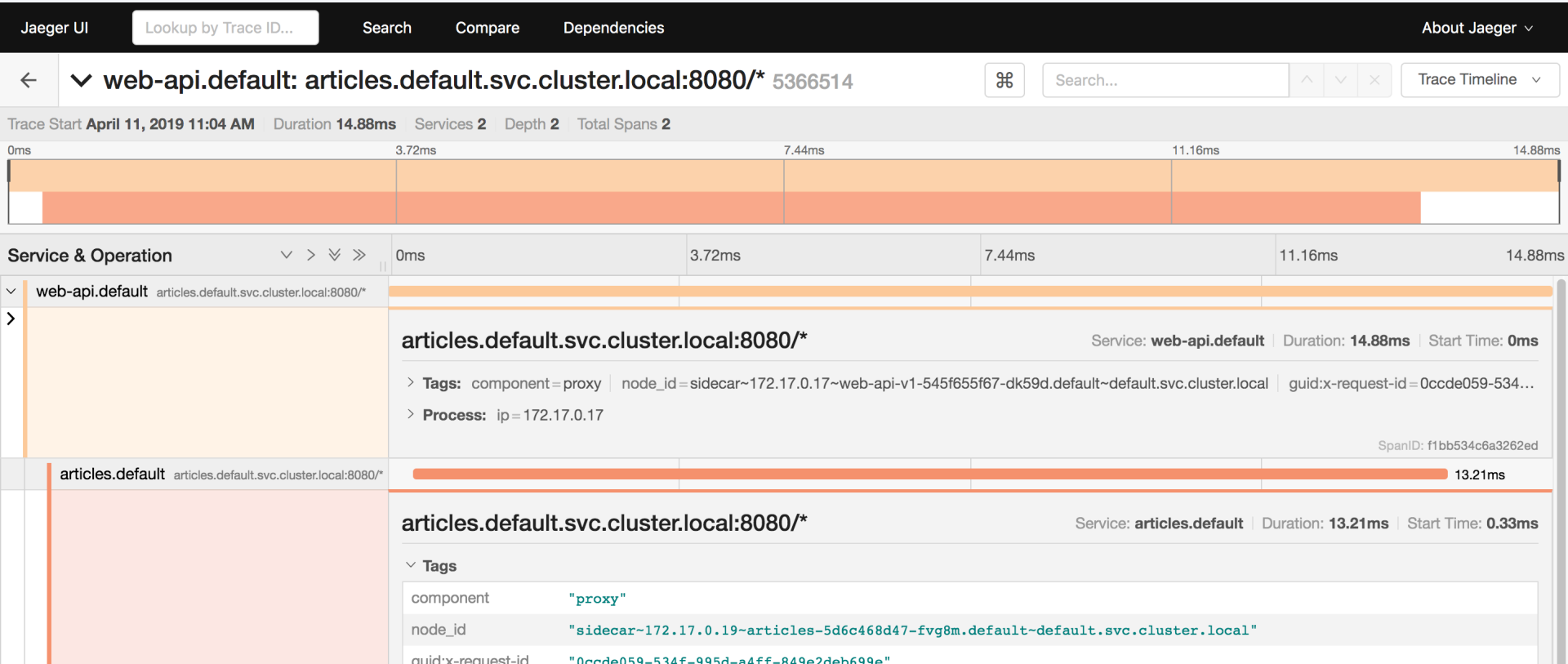
Jaeger and Zipkin

Open source distributed tracing systems

server.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<server description="OpenLiberty Server">

    <featureManager>
        <feature>webProfile-8.0</feature>
        <feature>microProfile-2.1</feature>
        <feature>usr:opentracingZipkin-0.31</feature>
    </featureManager>

    <httpEndpoint id="defaultHttpEndpoint" host="*"
        httpPort="8080" httpsPort="9443"/>

</server>
```

Distributed Tracing



Metrics

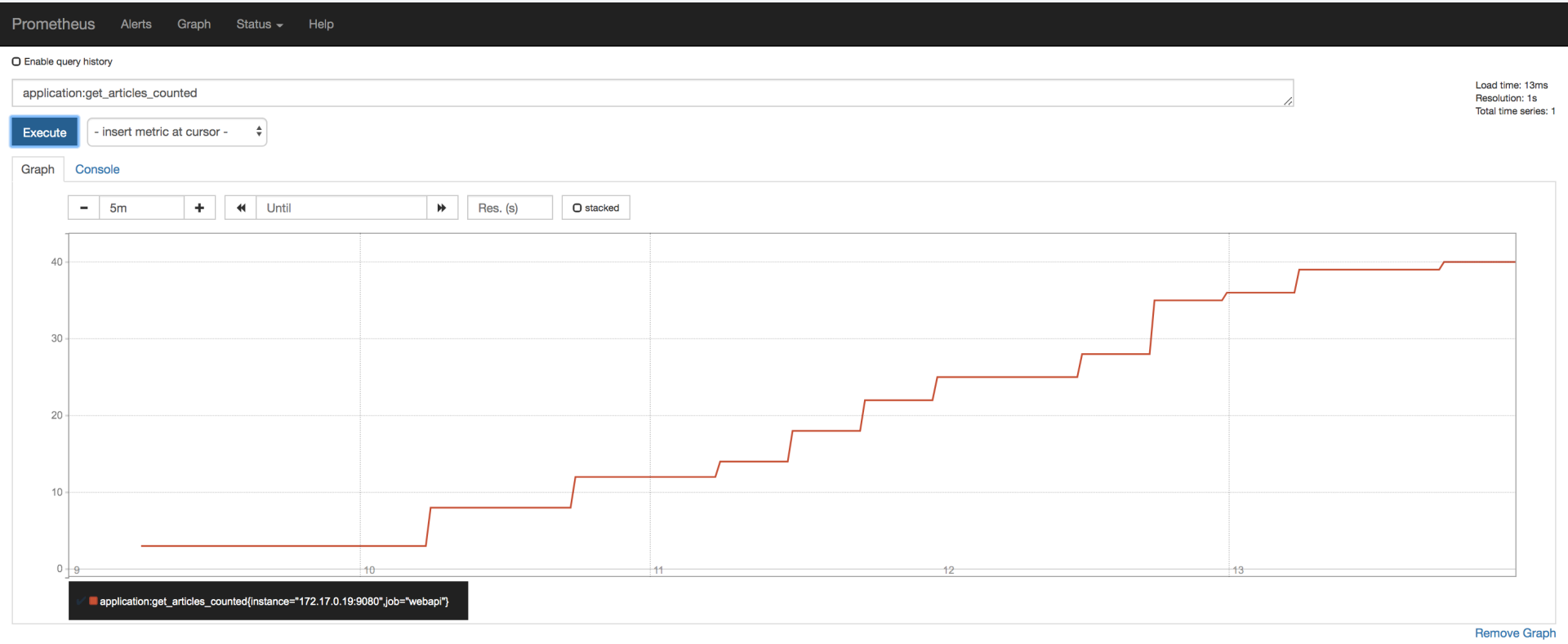
Prometheus

Monitoring system and time series database

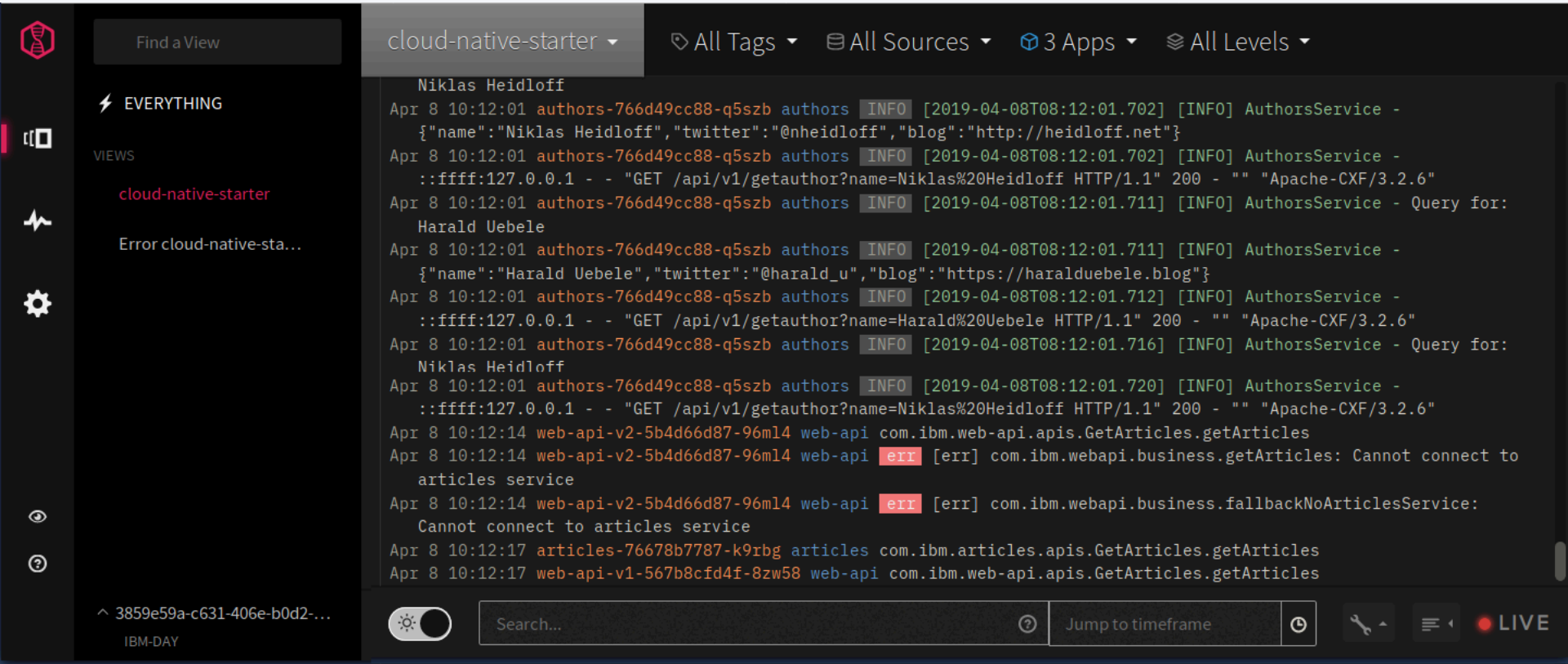
GetArticles.java

```
@Timed(name = "getArticlesTimed",
       absolute = true,
       displayName = "web-api /getmultiple timer",
       description = "Time taken by getArticles")
@Counted(name = "getArticlesCounted",
       absolute = true,
       displayName = "web-api /getmultiple count",
       description = "Number of times getArticles has been invoked",
       monotonic = true)
@Metered(name = "getArticlesMetered",
       displayName = "web-api /getmultiple frequency",
       description = "Rate the throughput of getArticles")
@GET
@Path("/getmultiple")
@Produces(MediaType.APPLICATION_JSON)
public Response getArticles() {
```

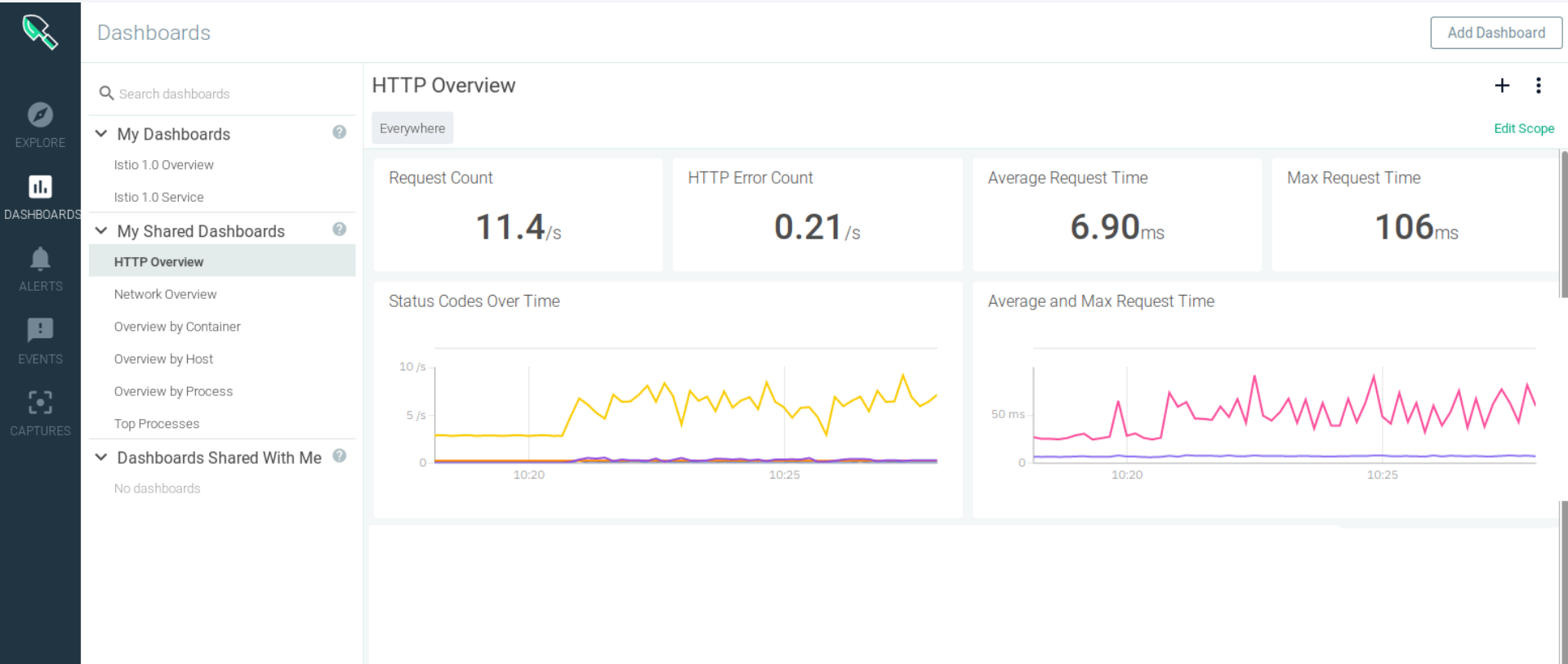
Metrics


Prometheus
Alerts
Graph
Status
Help
Enable query history
application:get_articles_counted
Load time: 13ms
Resolution: 1s
Total time series: 1
Execute
- insert metric at cursor -
Graph
Console
5m
Until
Res. (s)
stacked
40
30
20
10
0
9
10
11
12
13
application:get_articles_counted{instance="172.17.0.19:9080",job="webapi"}
Remove Graph

LogDNA


Find a View
cloud-native-starter
All Tags
All Sources
3 Apps
All Levels
EVERYTHING
VIEWS
cloud-native-starter
Error cloud-native-sta...
Niklas Heidloff
Apr 8 10:12:01 authors-766d49cc88-q5szb authors INFO [2019-04-08T08:12:01.702] [INFO] AuthorsService - {"name":"Niklas Heidloff","twitter":"@nheidloff","blog":"http://heidloff.net"}
Apr 8 10:12:01 authors-766d49cc88-q5szb authors INFO [2019-04-08T08:12:01.702] [INFO] AuthorsService - ::ffff:127.0.0.1 - - "GET /api/v1/getauthor?name=Niklas%20Heidloff HTTP/1.1" 200 - "" "Apache-CXF/3.2.6"
Apr 8 10:12:01 authors-766d49cc88-q5szb authors INFO [2019-04-08T08:12:01.711] [INFO] AuthorsService - Query for: Harald Uebele
Apr 8 10:12:01 authors-766d49cc88-q5szb authors INFO [2019-04-08T08:12:01.711] [INFO] AuthorsService - {"name":"Harald Uebele","twitter":"@harald_u","blog":"https://haralduebele.blog"}
Apr 8 10:12:01 authors-766d49cc88-q5szb authors INFO [2019-04-08T08:12:01.712] [INFO] AuthorsService - ::ffff:127.0.0.1 - - "GET /api/v1/getauthor?name=Harald%20Uebele HTTP/1.1" 200 - "" "Apache-CXF/3.2.6"
Apr 8 10:12:01 authors-766d49cc88-q5szb authors INFO [2019-04-08T08:12:01.716] [INFO] AuthorsService - Query for: Niklas Heidloff
Apr 8 10:12:01 authors-766d49cc88-q5szb authors INFO [2019-04-08T08:12:01.720] [INFO] AuthorsService - ::ffff:127.0.0.1 - - "GET /api/v1/getauthor?name=Niklas%20Heidloff HTTP/1.1" 200 - "" "Apache-CXF/3.2.6"
Apr 8 10:12:14 web-api-v2-5b4d66d87-96ml4 web-api com.ibm.web-api.apis.GetArticles.getArticles
Apr 8 10:12:14 web-api-v2-5b4d66d87-96ml4 web-api err [err] com.ibm.webapi.business.getArticles: Cannot connect to articles service
Apr 8 10:12:14 web-api-v2-5b4d66d87-96ml4 web-api err [err] com.ibm.webapi.business.fallbackNoArticlesService: Cannot connect to articles service
Apr 8 10:12:17 articles-76678b7787-k9rbg articles com.ibm.articles.apis.GetArticles.getArticles
Apr 8 10:12:17 web-api-v1-567b8cfd4f-8zw58 web-api com.ibm.web-api.apis.GetArticles.getArticles
3859e59a-c631-406e-b0d2-...
IBM-DAY
Search...
Jump to timeframe
LIVE

Sysdig


Dashboards
Add Dashboard
EXPLORE
DASHBOARDS
ALERTS
EVENTS
CAPTURES
Search dashboards
My Dashboards
Istio 1.0 Overview
Istio 1.0 Service
My Shared Dashboards
HTTP Overview
Network Overview
Overview by Container
Overview by Host
Overview by Process
Top Processes
Dashboards Shared With Me
No dashboards
HTTP Overview
Everywhere
Edit Scope
Request Count
11.4/s
HTTP Error Count
0.21/s
Average Request Time
6.90ms
Max Request Time
106ms
Status Codes Over Time
10 /s
5 /s
0
10:20
10:25
Average and Max Request Time
50 ms
0
10:20
10:25

Healthchecks

MicroProfile Health

Liveness probes and readiness probes

@nheidloff @Harald_U

HealthEndpoint.java

```java
@Health
@ApplicationScoped
public class HealthEndpoint implements HealthCheck {

    @Override
    public HealthCheckResponse call() {
        return HealthCheckResponse.named("web-api").withData("web-api", "ok").up().build();
    }
}
```

Service.yaml

```yaml
kind: Deployment
apiVersion: apps/v1beta1
metadata:
  name: web-api-v1
spec:
  replicas: 1
  template:
    metadata:
      labels:
        app: web-api
        version: v1
    spec:
      containers:
      - name: web-api
        image: web-api:1
        ports:
        - containerPort: 9080
        livenessProbe:
          exec:
            command: ["sh", "-c", "curl -s http://localhost:9080/"]
          initialDelaySeconds: 20
        readinessProbe:
          exec:
            command: ["sh", "-c", "curl -s http://localhost:9080/health | grep -q web-api"]
          initialDelaySeconds: 40
      restartPolicy: Always
```

Try out the end-to-end microservices example cloud-native-starter!

Deploying Sample Services

As easy as 'cf push'

scripts

ibm-scripts

iks-scripts

@nheidloff @Harald_U

sh scripts/deploy-authors-nodejs.sh

```
Niklass-MBP-2:cloud-native-starter nheidloff$ sh scripts/deploy-authors-nodejs.sh
2019-04-23 11:10:12 Deploying authors-nodejs
2019-04-23 11:10:12 Preparing YAML files for Kubernetes Deployment
2019-04-23 11:10:12 Clean-up Minikube
No resources found
2019-04-23 11:10:12 Build Docker Image
Sending build context to Docker daemon  36.86kB
Step 1/7 : FROM node:8-alpine
 ---> ee8b4f3c67fa
Step 2/7 : WORKDIR /usr/src/app
 ---> Using cache
 ---> e49ce865071d
Step 3/7 : COPY package*.json ./
 ---> 9dd062ef89e4
Step 4/7 : RUN npm install
 ---> Running in 2c33d74f125c
added 140 packages from 117 contributors and audited 364 packages in 99.853s
found 0 vulnerabilities
Removing intermediate container 2c33d74f125c
 ---> b75bce7fdd70
Step 5/7 : COPY . .
 ---> e5cae32accc6
Step 6/7 : EXPOSE 3000
 ---> Running in 9a3cb18f493a
Removing intermediate container 9a3cb18f493a
 ---> a916d5644669
Step 7/7 : CMD ["npm", "start"]
 ---> Running in 064f5ac77dbb
Removing intermediate container 064f5ac77dbb
 ---> 293376350ca7
Successfully built 293376350ca7
Successfully tagged authors:1
2019-04-23 11:12:15 Deploy to Minikube
deployment.apps/authors created
service/authors created
virtualservice.networking.istio.io/authors created
destinationrule.networking.istio.io/authors created
2019-04-23 11:12:28 Done deploying authors-nodejs
2019-04-23 11:12:28 Wait until the pod has been started: kubectl get pod --watch | grep authors
Niklass-MBP-2:cloud-native-starter nheidloff$ kubectl get pods
NAME                              READY     STATUS    RESTARTS   AGE
articles-5d6c468d47-fvg8m         2/2       Running   15         14d
authentication-85978ff5d5-m6w54   2/2       Running   16         15d
authors-57665fb76b-kpqcc          2/2       Running   0          2m20s
web-api-v1-545f655f67-n8vcd       2/2       Running   7          11d
web-app-55d4966c68-ljqb5          2/2       Running   10         14d
```

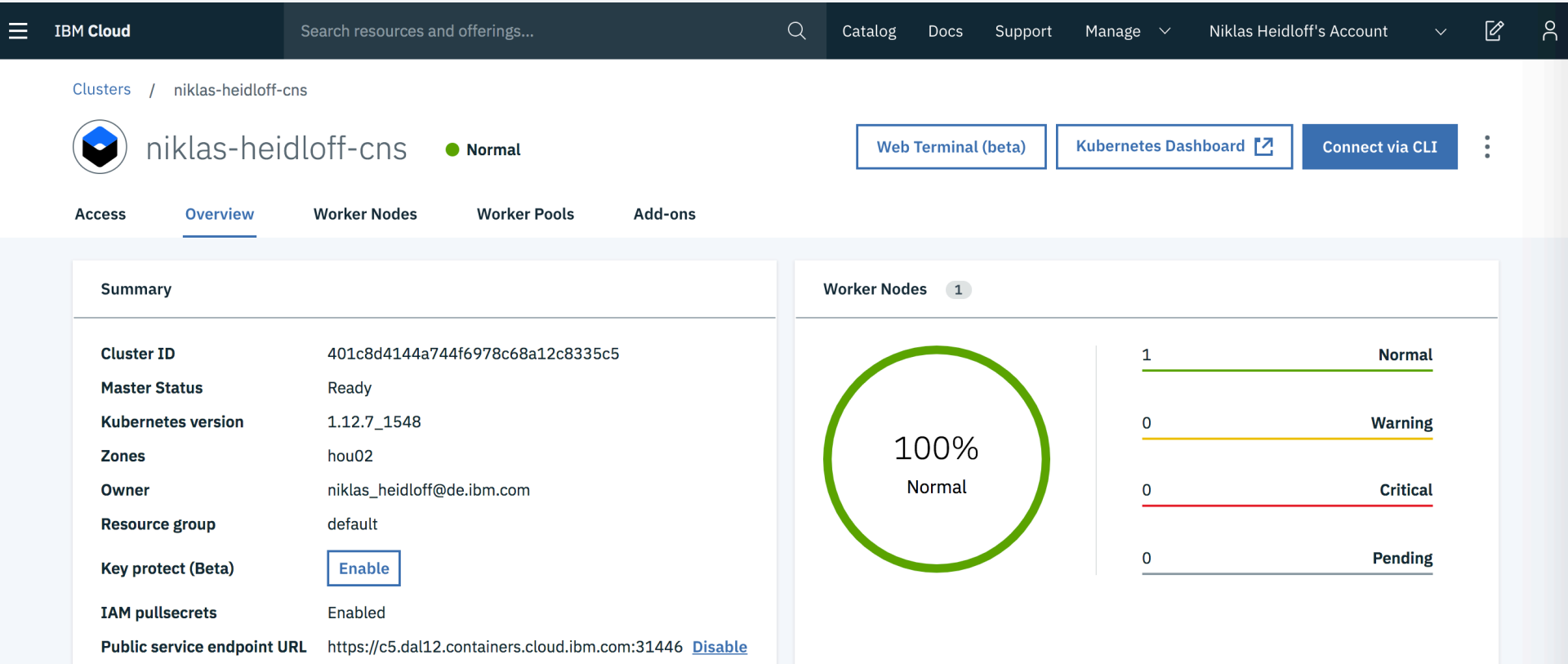
Running the Example App

Script displays all relevant URLs

sh scripts/show-urls.sh

```
Niklass-MBP-2:cloud-native-starter nheidloff$ sh scripts/show-urls.sh
2019-04-23 11:22:38 ------------------------------------------------------------------------------------------
2019-04-23 11:22:38 kiali
2019-04-23 11:22:38 Run the command: kubectl -n istio-system port-forward $(kubectl -n istio-system get
pod -l app=kiali -o jsonpath='{.items[0].metadata.name}') 20001:20001
2019-04-23 11:22:38 Then open http://localhost:20001/kiali/console with username: admin, password: admin
2019-04-23 11:22:38 ------------------------------------------------------------------------------------------
2019-04-23 11:22:38 prometheus
2019-04-23 11:22:38 Run the command: kubectl -n istio-system port-forward $(kubectl -n istio-system get
pod -l app=prometheus -o jsonpath='{.items[0].metadata.name}') 9090:9090 &
2019-04-23 11:22:38 Then open http://localhost:9090/
2019-04-23 11:22:38 ------------------------------------------------------------------------------------------
2019-04-23 11:22:38 jaeger
2019-04-23 11:22:38 Run the command: kubectl -n istio-system port-forward $(kubectl get pod -n istio-sys
tem -l app=jaeger -o jsonpath='{.items[0].metadata.name}') 16686:16686 &
2019-04-23 11:22:38 Then open http://localhost:16686
2019-04-23 11:22:38 ------------------------------------------------------------------------------------------
2019-04-23 11:22:38 grafana
2019-04-23 11:22:38 Run the command: kubectl -n istio-system port-forward $(kubectl -n istio-system get
pod -l app=grafana -o jsonpath='{.items[0].metadata.name}') 3000:3000 &
2019-04-23 11:22:38 Then open http://localhost:3000/dashboard/db/istio-mesh-dashboard
2019-04-23 11:22:38 ------------------------------------------------------------------------------------------
2019-04-23 11:22:38 articles
2019-04-23 11:22:38 API explorer: http://192.168.99.100:31035/openapi/ui/
2019-04-23 11:22:38 Sample API: curl http://192.168.99.100:31035/articles/v1/getmultiple?amount=10
2019-04-23 11:22:38 ------------------------------------------------------------------------------------------
2019-04-23 11:22:38 authors
2019-04-23 11:22:38 Sample API: curl http://192.168.99.100:32354/api/v1/getauthor?name=Niklas%20Heidloff
2019-04-23 11:22:38 ------------------------------------------------------------------------------------------
2019-04-23 11:22:38 authentication
2019-04-23 11:22:38 Login URL: http://192.168.99.100:31380/login
2019-04-23 11:22:38 User: user@demo.email / Password: verysecret
2019-04-23 11:22:38 Admin: admin@demo.email / Password: verysecret
2019-04-23 11:22:38 ------------------------------------------------------------------------------------------
2019-04-23 11:22:38 web-api
2019-04-23 11:22:38 API explorer: http://192.168.99.100:31380/openapi/ui/
2019-04-23 11:22:38 Metrics: http://192.168.99.100:31265/metrics/application
2019-04-23 11:22:38 Sample API: curl http://192.168.99.100:31380/web-api/v1/getmultiple
2019-04-23 11:22:38 ------------------------------------------------------------------------------------------
2019-04-23 11:22:38 web-app
2019-04-23 11:22:39 Web app: http://192.168.99.100:31380/
2019-04-23 11:22:39 ------------------------------------------------------------------------------------------
Niklass-MBP-2:cloud-native-starter nheidloff$
```

IBM Cloud Kubernetes Service including Istio and Knative



Summary

github.com/nheidloff/cloud-native-starter

Leverage platforms as much as possible

Use frameworks for app specific logic

IBM loves open source

Kubernetes and Istio

OpenJ9 & AdoptOpenJDK

MicroProfile

Open Liberty

IBM Developer

developer.ibm.com

IBM Cloud Lite account

ibm.biz/nheidloff


IBM