

# How to develop your first cloud-native Applications with Java

Niklas Heidloff  
Developer Advocate, IBM  
@nheidloff

Harald Uebel  
Developer Advocate, IBM  
@Harald\_U

# Once upon a time ...



# “Never not be afraid”

Grug Crood  
Beginning of ‘The Croods’



@nheidloff @Harald\_U

#IBMDeveloper [github.com/nheidloff/cloud-native-starter](https://github.com/nheidloff/cloud-native-starter)

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

Wikipedia

# What are cloud-native Applications?

Elasticity

→ App stays responsive

Continuous delivery

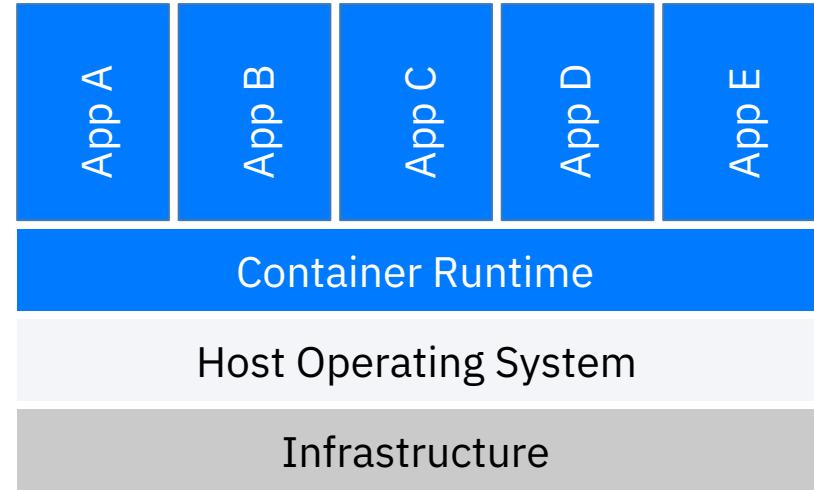
→ DevOps

# New Options → New Challenges



“A container image is a lightweight, standalone, executable package of software that includes everything needed to run an application.”

[docker.com](https://docker.com)



“I’m a caveman. Can you explain containers so that I understand it?”

Grug Crood  
Caveman learning microservices



# Portable Containers



# 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 openliberty/open-liberty:microProfile2-java8-openj9
```

```
COPY liberty/server.xml /config/
```

```
ADD target/articles.war /config/dropins/
```

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

[kubernetes.io](https://kubernetes.io)



# kubernetes

“When a container contains everything to run a microservice, why do I need Kubernetes?”

Grug Crood  
Caveman learning microservices



# Example Application

Cloud Native Starter

user@demo.email ▾

## Articles



Title

[Debugging Microservices running in Kubernetes](#)

[Dockerizing Java MicroProfile Applications](#)

[Install Istio and Kiali on IBM Cloud or Minikube](#)

[Three awesome TensorFlow.js Models for Visual Recognition](#)

[Blue Cloud Mirror Architecture Diagrams](#)



Author

Niklas Heidloff

Niklas Heidloff

Harald Uebele

Niklas Heidloff

Niklas Heidloff



Twitter

[@nheidloff](#)

[@nheidloff](#)

[@harald\\_u](#)

[@nheidloff](#)

[@nheidloff](#)



Blog

[Blog](#)

[Blog](#)

[Blog](#)

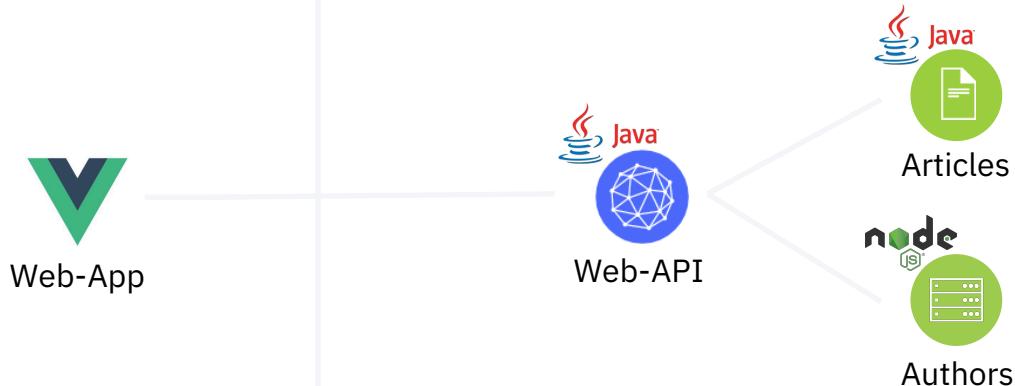
[Blog](#)

[Blog](#)

# Example Application – REST APIs

Browser

Kubernetes with Istio



# Exposing REST APIs

## JAX-RS

Java API for RESTful Web Services

### 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

## 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

The screenshot shows the Open Liberty Web-API Service interface. At the top, there's a logo and the text "Web-API Service 1.0 OAS3". Below that, it says "Web-API Service APIs". A "Server" dropdown is set to "http://192.168.99.100:31380/web-api".

The main area is titled "default". It lists two endpoints:

- POST /v1/create** Create a new article
- GET /v1/getmultiple** Get most recently added articles

For the GET endpoint, there's a description: "Get most recently added articles". Under "Parameters", it says "No parameters". There are "Execute" and "Clear" buttons at the bottom.

Below that, under "Responses", is a "Curl" section with the command: `curl -X GET "http://192.168.99.100:31380/web-api/v1/getmultiple" -H "accept: application/json"`. There's also a "Request URL" field containing `http://192.168.99.100:31380/web-api/v1/getmultiple`.

Under "Server response", there's a table with one row:

Code	Details
200	Response body

The "Response body" section shows a JSON array:

```
[{"id": "1555051929394", "title": "Example Java App running in the Cloud via Kubernetes", "url": "http://heidloff.net/article/example-java-app-cloud-kubernetes", "authorName": "Niklas Heidloff", "authorBlog": "http://heidloff.net", "authorTwitter": "@heidloff"}, {"id": "1555051929395", "title": "Java Application Containerization with Docker", "url": "http://heidloff.net/article/java-application-containerization-docker", "authorName": "Niklas Heidloff", "authorBlog": "http://heidloff.net", "authorTwitter": "@heidloff"}]
```

# Consuming REST APIs

## MicroProfile Rest Client

Type-safe approach to invoke RESTful services

### 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);
        }
    }
}
```

# Consuming REST APIs

## MicroProfile Rest Client

Type-safe approach to invoke RESTful services

### 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() {}

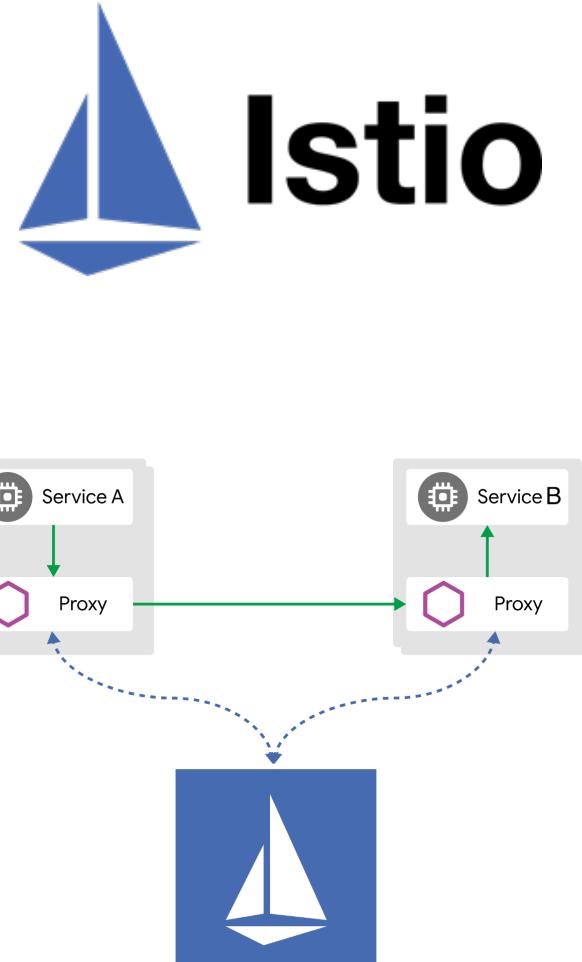
    static final String BASE_URL = "http://authors:3000/api/v1/";

    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);
        }
    }
}
```

“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](https://github.com/istio/istio)



# “Why do I need a service mesh? Can’t I just use Kubernetes?”

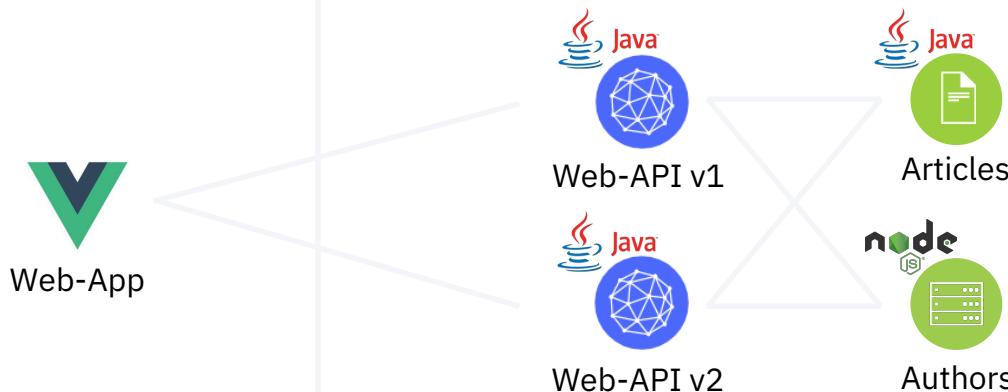
Grug Crood  
Caveman learning microservices



# Example Application – Traffic Management

Browser

Kubernetes with Istio



# Traffic Management

**80% / 20% splitting**

## ingress.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

**kiali**

Namespace: default

Graph ?

Display Edge Labels Graph Type Versioned app Find... Hide... Fetching Last min Every 5 sec

Apr 11, 11:25:29 ... Apr 11, 11:26:29

The graph illustrates the traffic flow between three services: web-app, web-api, and articles/authors. The web-app service has two versions, v1 and v1, with 100% traffic to v1. The web-api service also has two versions, v1 and v2, with 80% traffic to v1 and 20% to v2. The articles/authors service has two versions, v1 and v1, with 100% traffic to v1. There are two gateway nodes: istio-ingressgateway (istio-system) and web-gateway. The istio-ingressgateway receives 14.3% traffic from the web-app and 85.7% from the web-api. The web-gateway receives 100% traffic from the web-app. The web-api gateway receives 80% traffic from version v1 and 20% from version v2. The articles/authors gateway receives 16.7% traffic from the web-api and 83.3% from the web-gateway.

Namespace: default  
applications, services, workloads

Current Graph:  
6 apps  
4 services  
11 edges

HTTP Traffic (requests per second):  
Total %Success %Error  
6.65 100.00 0.00

0 25 50 75 100 %

OK 3xx 4xx 5xx

# “Optimizing Enterprise Java for a Microservices Architecture.

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

[micrometer.io](https://micrometer.io)



“Dude, I just learned Istio and  
Kubernetes. Can you show  
me a simple MicroProfile  
example?”

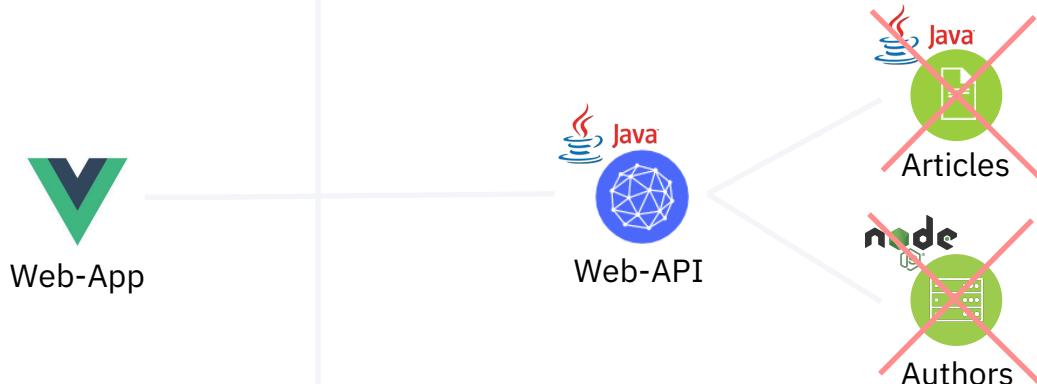
Grug Crood  
Caveman learning microservices



# Example Application – Resiliency

Browser

Kubernetes with Istio



# Resiliency

Authors service not available

## Usage of default values

### 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

Articles service not available

## MicroProfile Fallback annotation

### 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

[Debugging Microservices running in Kubernetes](#)

[Dockerizing Java MicroProfile Applications](#)

[Install Istio and Kiali on IBM Cloud or Minikube](#)

[Three awesome TensorFlow.js Models for Visual Recognition](#)

[Blue Cloud Mirror Architecture Diagrams](#)



Author

Niklas Heidloff

Niklas Heidloff

Harald Uebele

Niklas Heidloff

Niklas Heidloff



Twitter



Blog

# Authentication and Authorization

## OpenID Connect

Identity layer on top of the OAuth 2.0 protocol

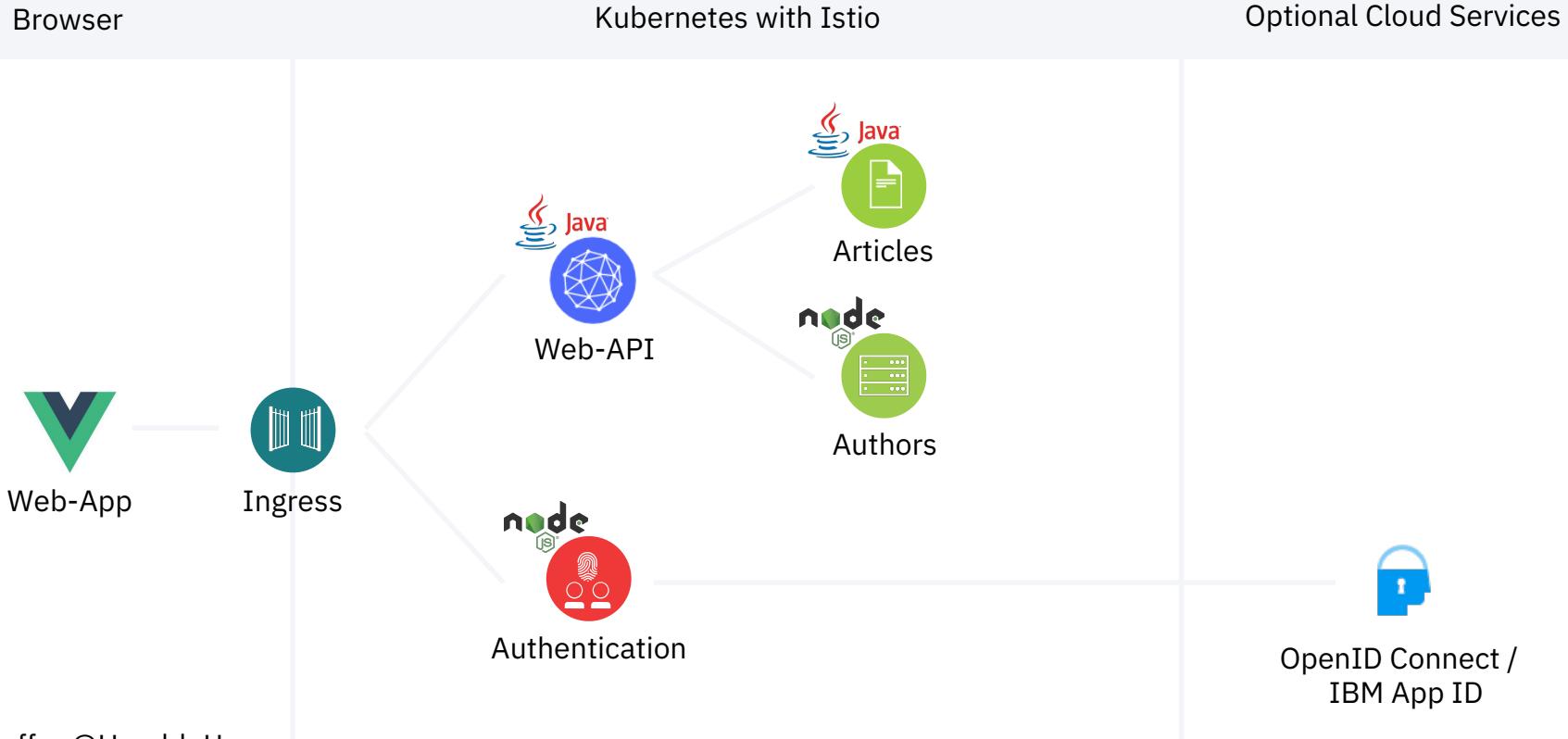
## IBM App ID

IBM service to authenticate users and protect APIs

## policy.yaml

```
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
```

# Example Application – Authentication and Authorization



# Authentication



Email:

A text input field for entering an email address. It features a small icon of a person in a grey box to its left. The placeholder text "user@demo.email" is visible inside the input field.

Password:

A password input field represented by a text input with a lock icon in a grey box on the left and a series of dots representing the password characters.

[Forgot Password?](#)

**Login**

Don't have an account? [Sign up](#)

# Authorization

Cloud Native Starter

user@demo.email ▾

## Create new Article

Title:

Example Java App running in the Cloud via Kubernetes

URL:

<http://heidloff.net/article/example-java-app-cloud-kubernetes>

Author:

Niklas Heidloff

Submit

Show Articles

# Authorization

Cloud Native Starter

user@demo.email ▾

## Articles

Title	Author	Twitter	Blog
<a href="#">Example Java App running in the Cloud via Kubernetes</a>	Niklas Heidloff	@nheidloff	<a href="#">Blog</a>
<a href="#">Debugging Microservices running in Kubernetes</a>	Niklas Heidloff	@nheidloff	<a href="#">Blog</a>
<a href="#">Dockerizing Java MicroProfile Applications</a>	Niklas Heidloff	@nheidloff	<a href="#">Blog</a>
<a href="#">Install Istio and Kiali on IBM Cloud or Minikube</a>	Harald Uebele	@harald_u	<a href="#">Blog</a>
<a href="#">Three awesome TensorFlow.js Models for Visual Recognition</a>	Niklas Heidloff	@nheidloff	<a href="#">Blog</a>

Network

Filter Hide data URLs

All XHR JS CSS Img Media Font Doc WS Manifest Other

Name	Headers	Preview	Response	Timing
auth	General			
callback?code=w...				
loginwithtoken?na...				
app.9cd06f5a.css				
chunk-vendors.73...				
app.bfe2e0c6.js				
chunk-vendors.3e...				
getmultiple				
create	Response Headers			
getmultiple				

Request URL: http://192.168.99.100:31380/web-api/v1/create  
Request Method: POST  
Status Code: 201 Created  
Remote Address: 192.168.99.100:31380  
Referrer Policy: no-referrer-when-downgrade

access-control-allow-credentials: true  
access-control-allow-headers: origin, content-type, accept, aut...  
access-control-allow-methods: GET, POST, PUT, DELETE, OPTIONS,  
access-control-allow-origin: \*  
content-language: en-US  
content-length: 182  
content-type: application/json  
date: Fri, 12 Apr 2019 06:52:09 GMT  
server: istio-envoy  
x-envoy-upstream-service-time: 346  
x-powered-by: Servlet/4.0

Request Headers

⚠ Provisional headers are shown

Accept: application/json, text/plain, \*/\*  
Authorization: Bearer eyJhbGci...  
KLTrMTQ2NGZnLmNKOTM1NDM5Ii...  
Iy0jM1ljk5NCi...InZlcii6NH0.e...  
3VKLmlibS5jb20vb2f1dGvdjQvN...  
ZTY0liw1iXVkiobImVUOYxNDc2L...

# Authorization

## Via MicroProfile

### Manage.java

```
@RequestScoped  
@Path("/v1")  
public class Manage {  
  
    @Inject  
    private JsonWebToken jwtPrincipal;  
  
    @POST  
    @Path("/manage")  
    @Produces(MediaType.APPLICATION_JSON)  
    @Operation(summary = "Manage app", description = "Manage app")  
    public Response manage() {  
        System.out.println("com.ibm.web-api.apis.Manage.manage");  
        System.out.println(this.jwtPrincipal);  
  
        String principalEmail = this.jwtPrincipal.getClaim("email");  
        if (principalEmail.equalsIgnoreCase("admin@demo.email")) {  
            JsonObject output = Json.createObjectBuilder()  
                .add("message", "success").build();  
            return Response.ok(output).build();  
        }  
    }  
}
```

# “Microservices sound great, but where is the log file?”

Grug Crood  
Caveman learning microservices



# 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

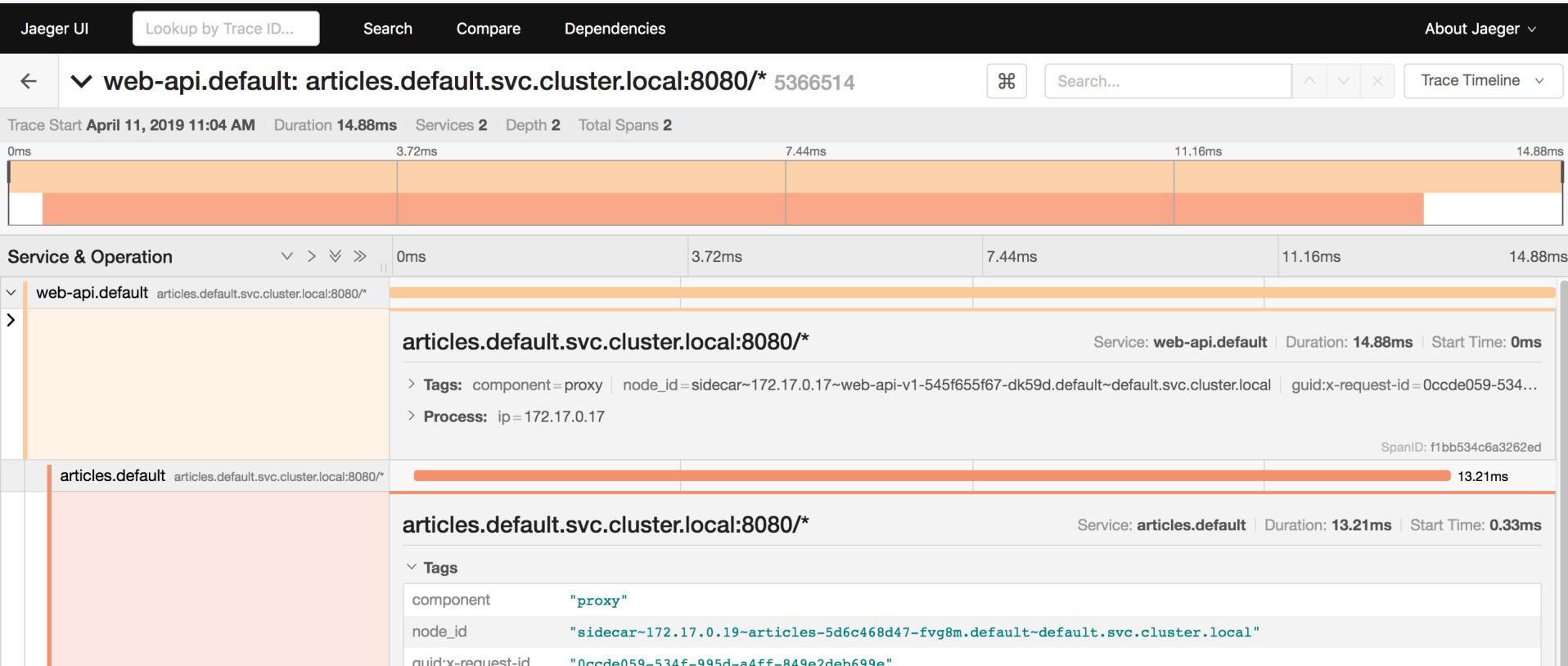
```
<?xml version="1.0" encoding="UTF-8"?>
<server description="OpenLiberty Server">

    <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

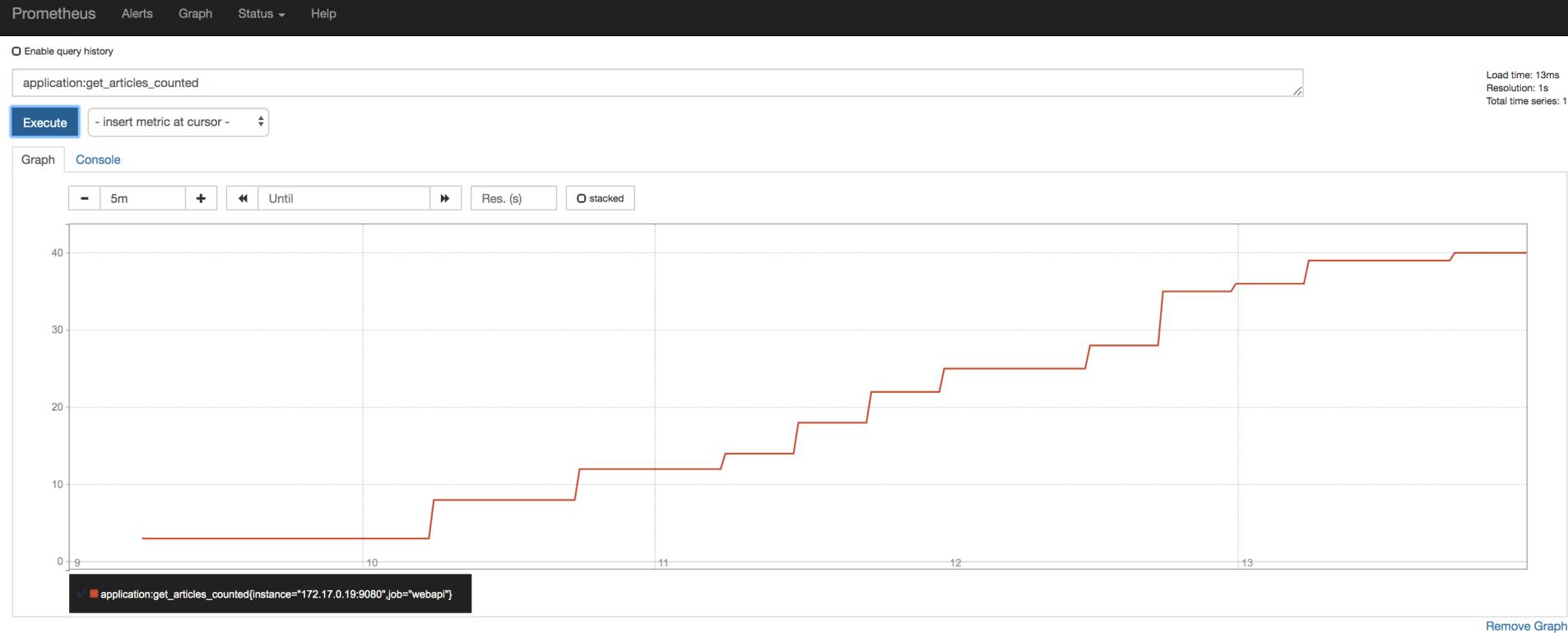
## Prometheus

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



# LogDNA

Find a View

EVERYTHING

VIEWS

cloud-native-starter

Error cloud-native-sta...

Niklas Heidloff

AuthorsService -

{ "name": "Niklas Heidloff", "twitter": "@nheidloff", "blog": "http://heidloff.net" }

AuthorsService -

::ffff:127.0.0.1 - - "GET /api/v1/getauthor?name=Niklas%20Heidloff HTTP/1.1" 200 - "" "Apache-CXF/3.2.6"

AuthorsService - Query for:

Harald Uebele

AuthorsService -

{ "name": "Harald Uebele", "twitter": "@harald\_u", "blog": "https://haralduebele.blog" }

AuthorsService -

::ffff:127.0.0.1 - - "GET /api/v1/getauthor?name=Harald%20Uebele HTTP/1.1" 200 - "" "Apache-CXF/3.2.6"

AuthorsService - Query for:

Niklas Heidloff

AuthorsService -

::ffff:127.0.0.1 - - "GET /api/v1/getauthor?name=Niklas%20Heidloff HTTP/1.1" 200 - "" "Apache-CXF/3.2.6"

web-api-v2-5b4d66d87-96ml4 web-api com.ibm.web-api.apis.GetArticles.getArticles

err com.ibm.webapi.business.getArticles: Cannot connect to articles service

web-api-v2-5b4d66d87-96ml4 web-api err com.ibm.webapi.business.fallbackNoArticlesService: Cannot connect to articles service

articles-76678b7787-k9rbg articles com.ibm.articles.apis.GetArticles.getArticles

web-api-v1-567b8cf4f-8zw58 web-api com.ibm.web-api.apis.GetArticles.getArticles

^ 3859e59a-c631-406e-b0d2...

IBM-DAY

Search...

Jump to timeframe

LIVE

# Sysdig

 Dashboards

 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

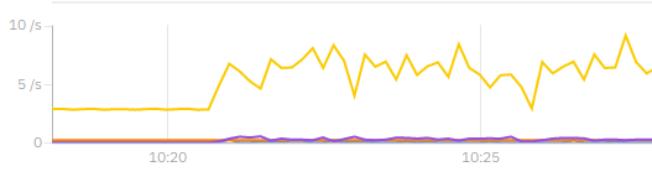
 Add Dashboard 

## HTTP Overview

Everywhere

Request Count	HTTP Error Count	Average Request Time	Max Request Time
<b>11.4</b> /s	<b>0.21</b> /s	<b>6.90</b> ms	<b>106</b> ms

Status Codes Over Time



Average and Max Request Time



# Healthchecks

## MicroProfile Health

Liveness probes and readiness probes

@nheidloff @Harald\_U

### HealthEndpoint.java

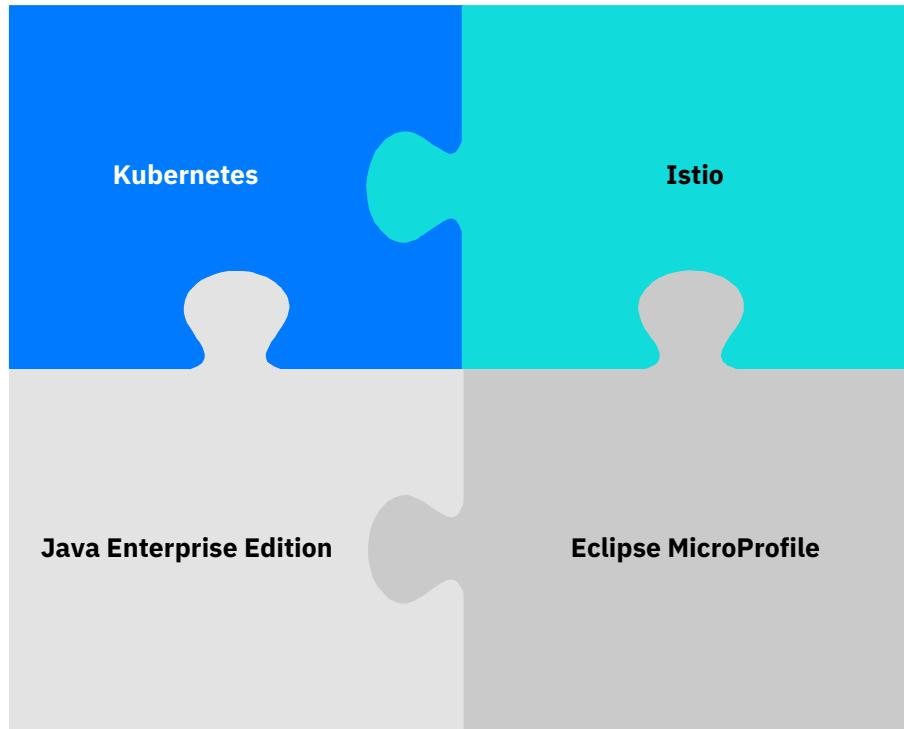
```
@Health
@ApplicationScoped
public class HealthEndpoint implements HealthCheck {

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

### Service.yaml

```
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
```

# How to use all Pieces together?

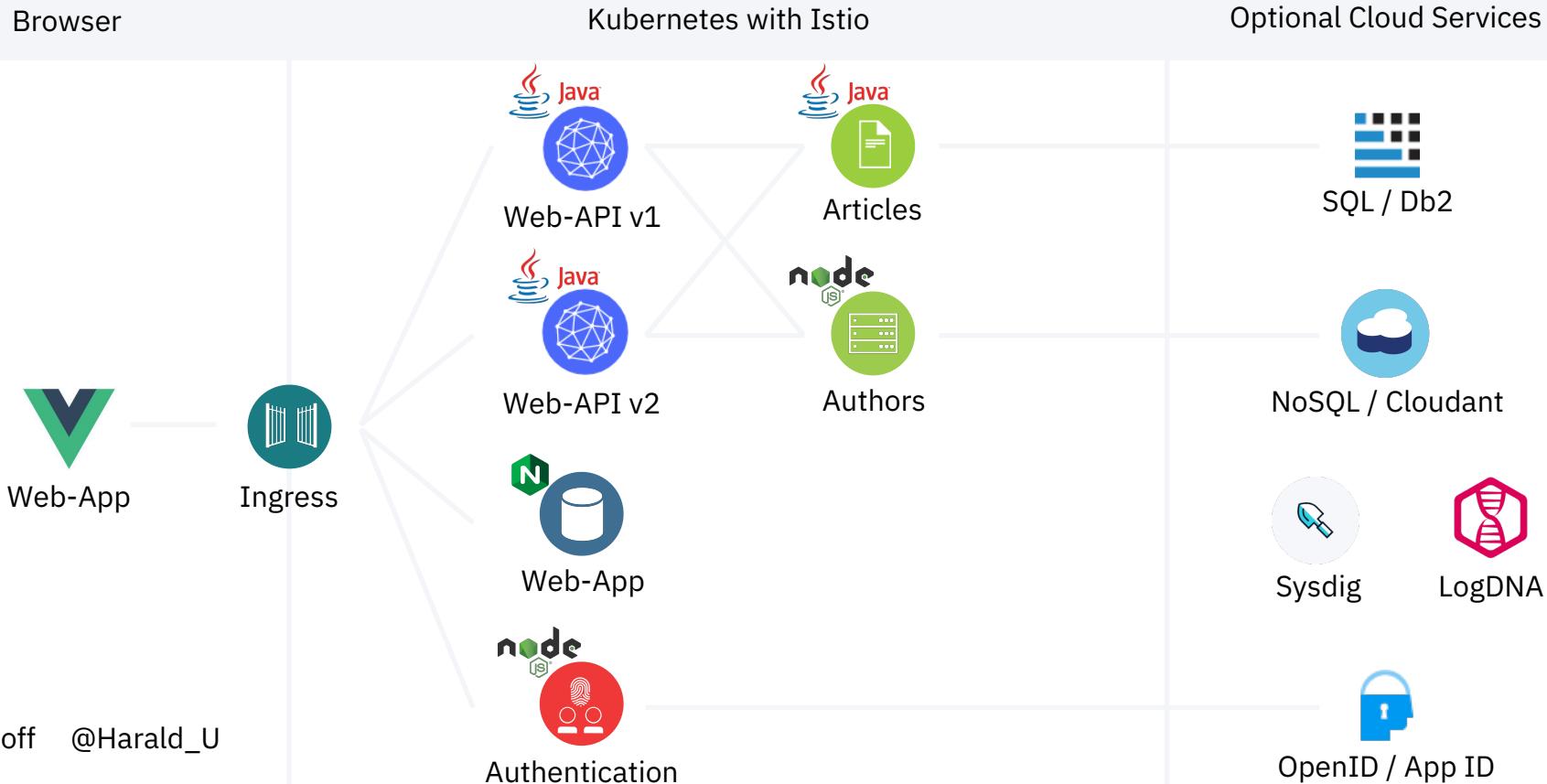


Leverage platforms as  
much as possible.

Use frameworks for app  
specific logic.

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

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



# IBM Cloud Kubernetes Service including Istio and Knative

IBM Cloud Search resources and offerings... Catalog Docs Support Manage Niklas Heidloff's Account ⚙️ 📜

Clusters / niklas-heidloff-cns

 niklas-heidloff-cns • Normal

[Web Terminal \(beta\)](#) [Kubernetes Dashboard ↗](#) [Connect via CLI](#) ⋮

[Access](#) [Overview](#) [Worker Nodes](#) [Worker Pools](#) [Add-ons](#)

**Summary**

<b>Cluster ID</b>	401c8d4144a744f6978c68a12c8335c5
<b>Master Status</b>	Ready
<b>Kubernetes version</b>	1.12.7_1548
<b>Zones</b>	hou02
<b>Owner</b>	niklas_heidloff@de.ibm.com
<b>Resource group</b>	default
<b>Key protect (Beta)</b>	<a href="#">Enable</a>
<b>IAM pullsecrets</b>	Enabled
<b>Public service endpoint URL</b>	<a href="https://c5.dal12.containers.cloud.ibm.com:31446">https://c5.dal12.containers.cloud.ibm.com:31446</a> <a href="#">Disable</a>

**Worker Nodes** 1



1	Normal
0	Warning
0	Critical
0	Pending

# “Never not be afraid”

Grug Crood  
Beginning of ‘The Croods’



@nheidloff @Harald\_U

#IBMDeveloper [github.com/nheidloff/cloud-native-starter](https://github.com/nheidloff/cloud-native-starter)

# “Never be afraid”

Grug Crood  
End of ‘The Croods’



@nheidloff @Harald\_U

#IBMDeveloper [github.com/nheidloff/cloud-native-starter](https://github.com/nheidloff/cloud-native-starter)

# Summary

Get the code →



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](https://developer.ibm.com)

IBM Cloud Lite account

[ibm.biz/nheidloff](https://ibm.biz/nheidloff)

