Microservices, Websphere Liberty and Kubernetes - a fully buzzword-compliant devOps pipeline

Mark Nuttall
mnuttall@uk.ibm.com

InterConnect
2017

IBM

Please note

IBM’s statements regarding its plans, directions, and intent are subject to change or withdrawal without notice at IBM’s sole discretion.

Information regarding potential future products is intended to outline our general product direction and it should not be relied on in making a purchasing decision.

The information mentioned regarding potential future products is not a commitment, promise, or legal obligation to deliver any material, code or functionality. Information about potential future products may not be incorporated into any contract.

The development, release, and timing of any future features or functionality described for our products remains at our sole discretion.

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. The actual throughput or performance that any user will experience will vary depending upon many factors, including considerations such as the amount of multiprogramming in the user’s job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve results similar to those stated here.

Agenda

Introducing Microservice Builder (Beta)

On-Prem DevOps Pipeline

Demo

Introducing Microservice Builder (Beta)

Better software delivery through devOps

- As an IT business I want to
  - Make money: reduce time to market
  - Save money: reduce faults, outages, manual steps, unused capacity
- Buzzwords to the rescue!
  - devOps
  - Cloud computing
  - Automation, continuous delivery, pipelines
  - Microservices
  - Kubernetes
  - Docker

The Microservices Revolution

Connecting people and digital apps better than ever before

A microservices architecture is gaining traction for developing and delivering cloud-native workloads across public, private, and hybrid application environments

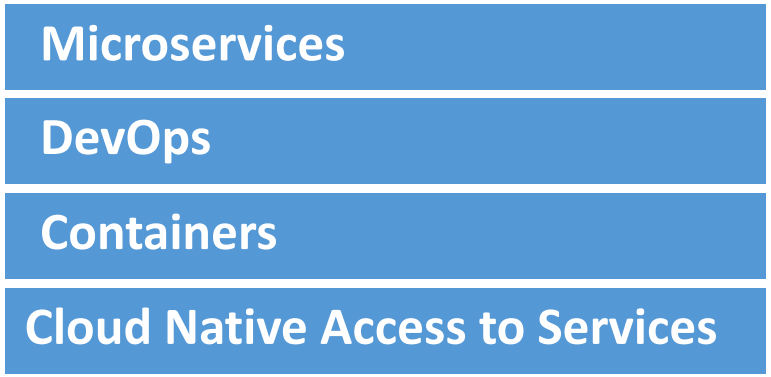
***60% of new apps will use cloud-enabled continuous delivery and cloud-native application architectures to enable faster innovation and business agility** (Bluemine)*

Why?

- ✓ Decomposed into small pieces
- ✓ Loosely coupled
- ✓ Easier to scale development
- ✓ Improved fault isolation
- ✓ Each service can be developed and deployed independently
- ✓ Eliminates any long-term commitment to a technology stack

Cloud-Native

An **application architecture** designed to leverage the **strengths** and accommodate the **challenges** of a **standardized** cloud environment, including concepts such as **elastic** scaling, **immutable** deployment, **disposable** instances, and **less predictable infrastructure**.



Cloud Native and Microservices: The journey

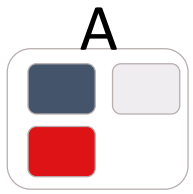
Stages of Maturity

Single Developer (experimentation)

Single Container

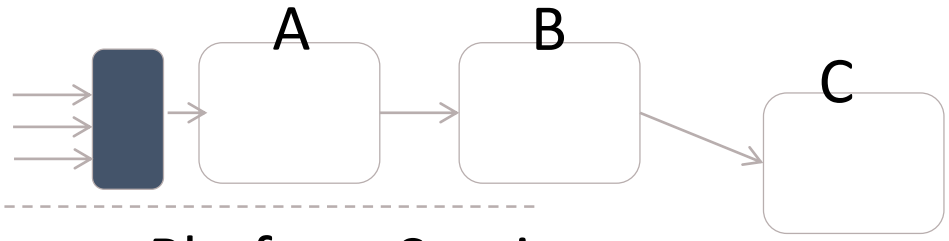
A Single Team

Single Service



Multiple Teams Collaborating

Microservices App



Technology /Ecosystem

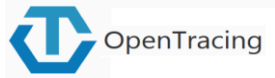
Docker Containers

Pods , Clusters

k8

swarm

Platform Services



Problem Solved

Portability

Orchestration
Clustering

Integration
Collaboration
Innovation

The Java programming model for microservices

Define a Java programming model in order to:

- Provide guidance on developing microservices
- Consistent qualities of service (scaling, health, analytics) across all envs
- Make Java microservices portable and interoperable
- Ensure applications have overall control over the environment provided similar checks

Java programming model:

Collaborate in MicroProfile.io: evolution of Java EE focused on microservices. Open collaboration led by IBM & Red Hat

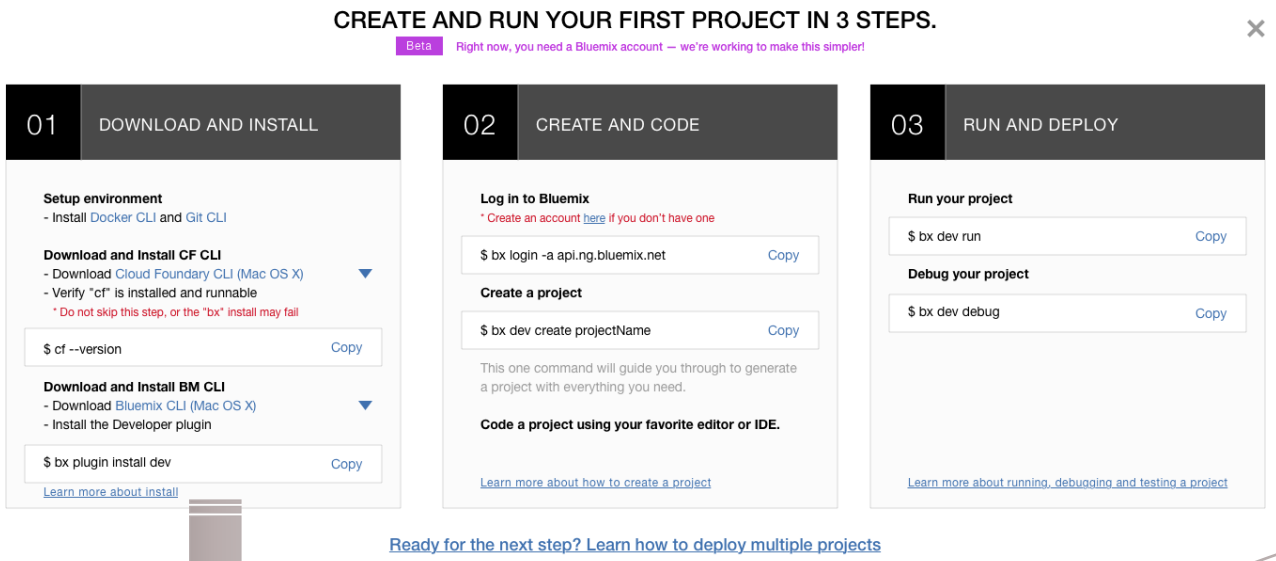
CDI:	A loose-coupling model for developing microservices
Config:	A model of isolating config source from microsvcs code
Fault Tolerance:	A model to build a robust microservices
Monitoring:	A model to provide vendor and application specific data
Health Check:	A model to provide health check data
Trace Correlation:	A model to correlate trace
Security:	A model to secure microservices

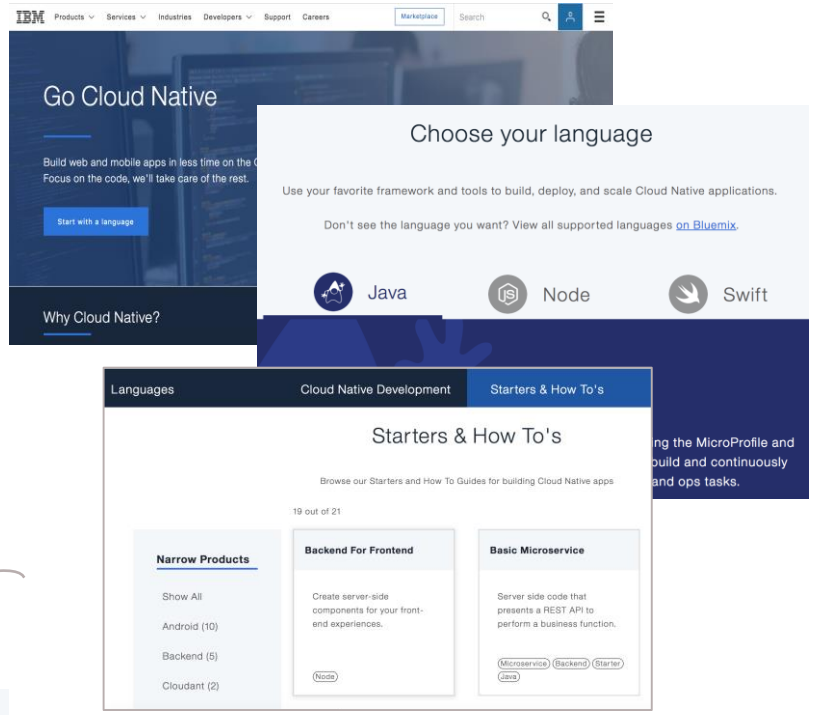
Adding resilience to the Java microservices programming model

As a developer, ….

- **Retry** – execute the same operation multiple times on failures
  - ….I would like to retry the same operation on some failure
- **Fallback** – provide an alternative result for a failed execution. Can be used to suppress exceptions and provide a default result.
  - …..I would like to fall back to a secondary service if the primary service is down.
- **Timeout** – When making a service invocation, the request must have a timeout in order to prevent from indefinite wait.
  - ….I would like to get timeout on request instead of waiting indefinitely.
- **CircuitBreaker** – provide a way to avoid repeating timeouts.
  - ….I would like to fail fast after a repeating failure instead of waiting for timeout to occur.
- **Bulkhead** – failure in one part doesn't take down the whole app.
  - ….I would like to make sure one service failure does not cause the whole application to fail.

Cloud Native Developer Experience





IBM Developer Hub

IBM Project Generators

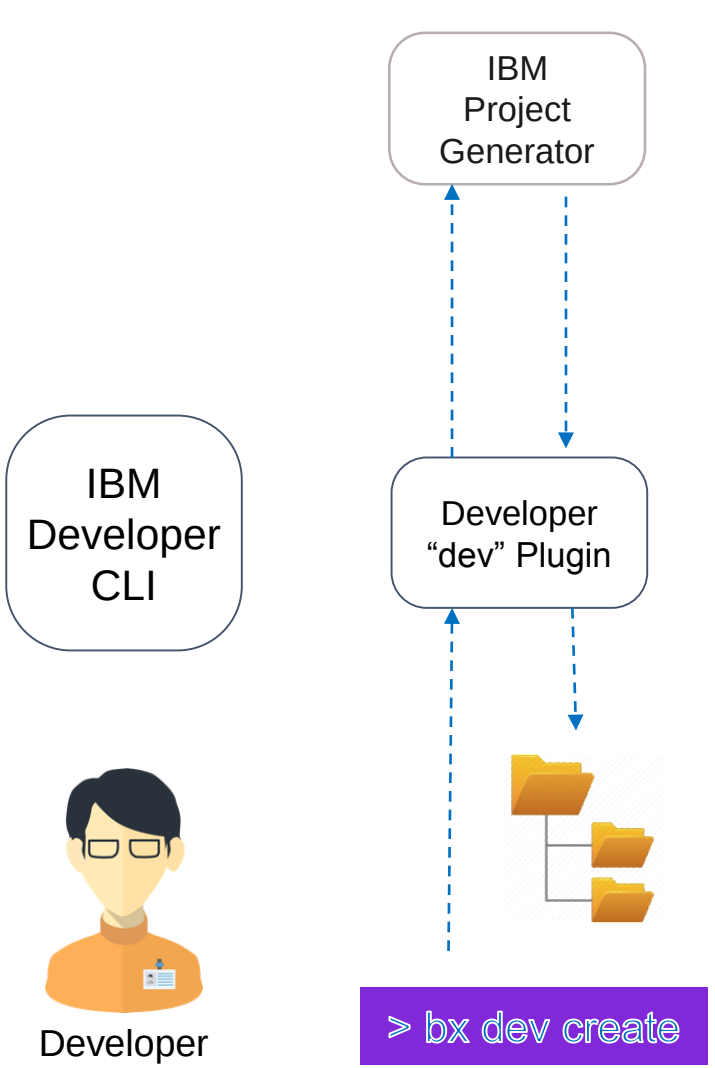
IBM Developer CLI:
create, run, test, deploy

Downloaded Projects

Application Patterns:

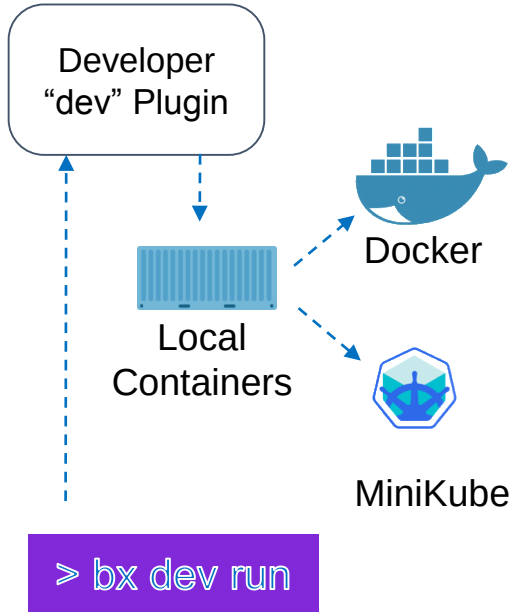
1) Web
2) Mobile
3) Backend-4-Frontend
4) Microservice

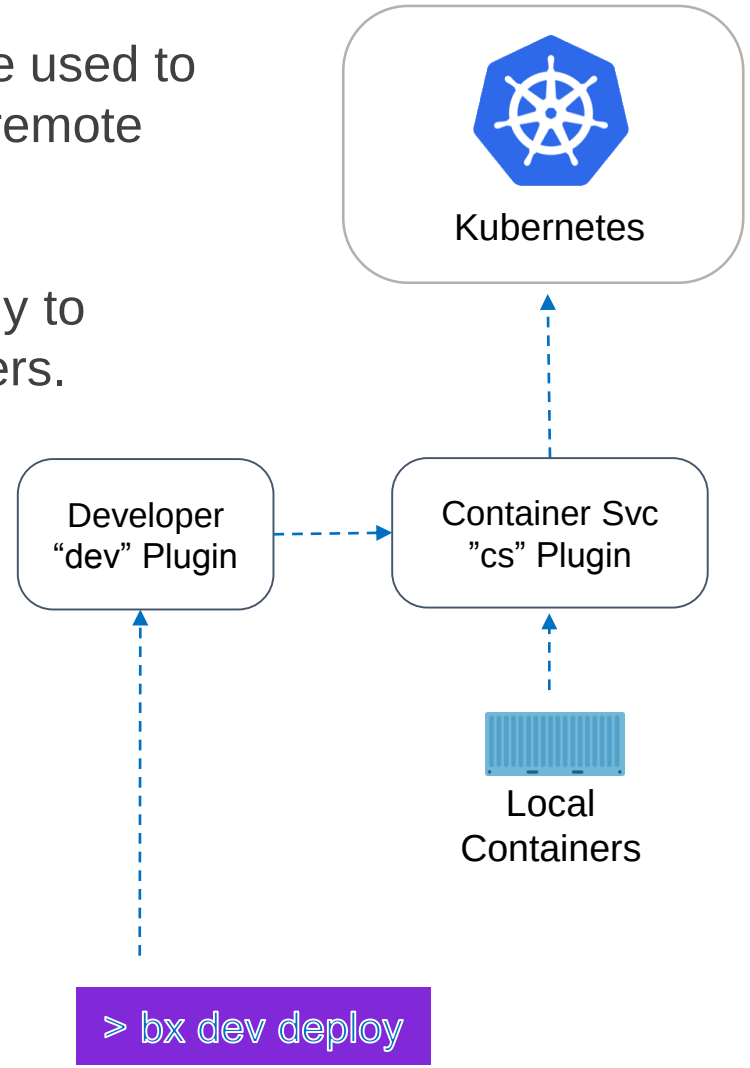
Developer CLI Workflow



In a basic environment Docker can be used to manage a single container, using all remote services.

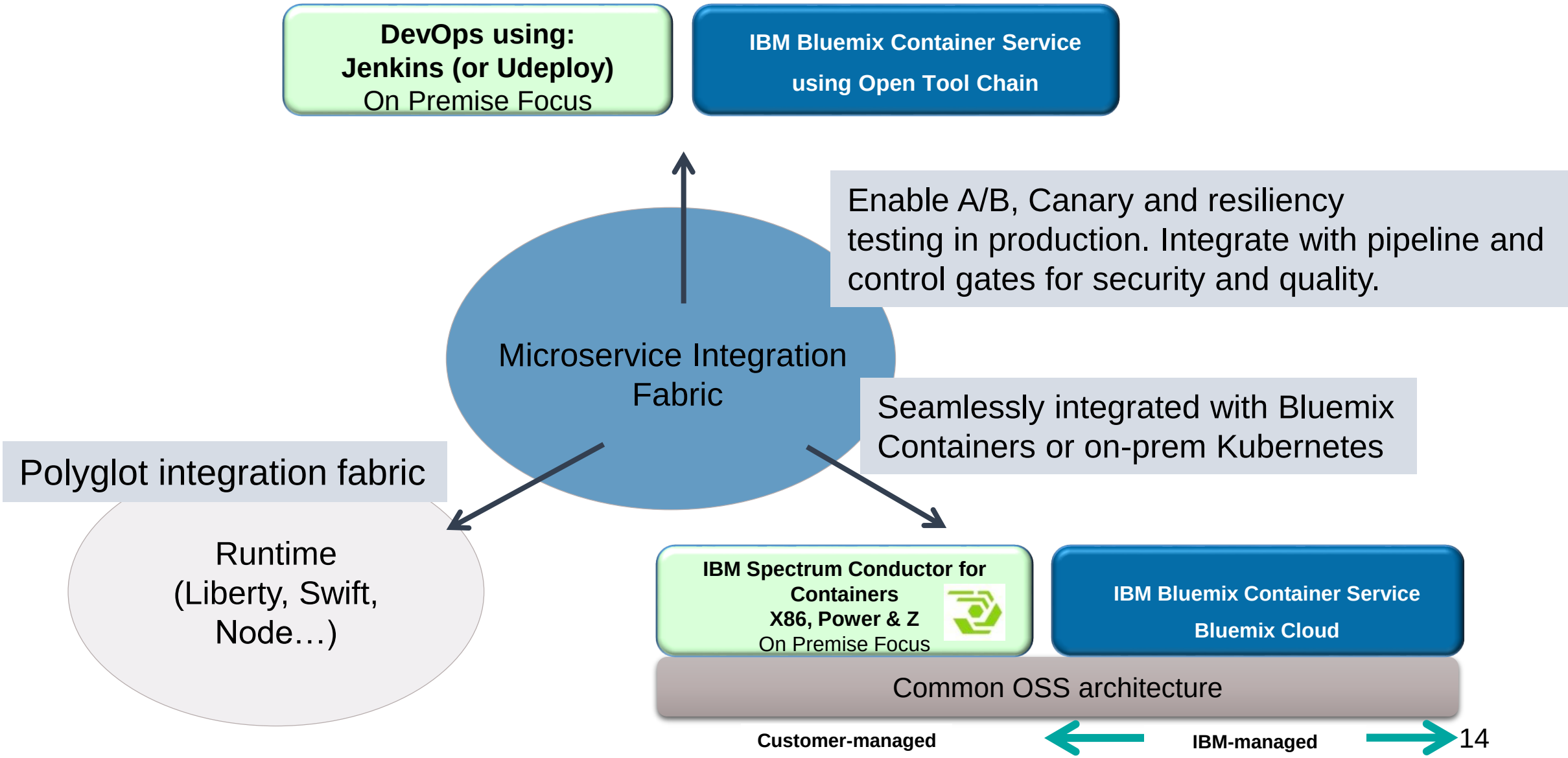
More realistically, use MiniKube locally to manage multiple interrelated containers.





Note: DevOps flow not shown

Cloud-native and Microservices Run Environments



IBM's approach: Providing a A Turnkey Solution to Build Microservices

Application

MicroProfile

Swift

node

Standardization of programming model for microservices

Dev Env

Dev Tools

Innovate through ease of connecting w/Watson & Connect Services

Fabric

elastic +

logstash +

kibana

Monitoring services for real-time visibility

Pipeline

urban{code} Deploy

Automation through IBM DevOps

On-Off premise

docker

kubernetes

collectives

IBM Spectrum Conductor

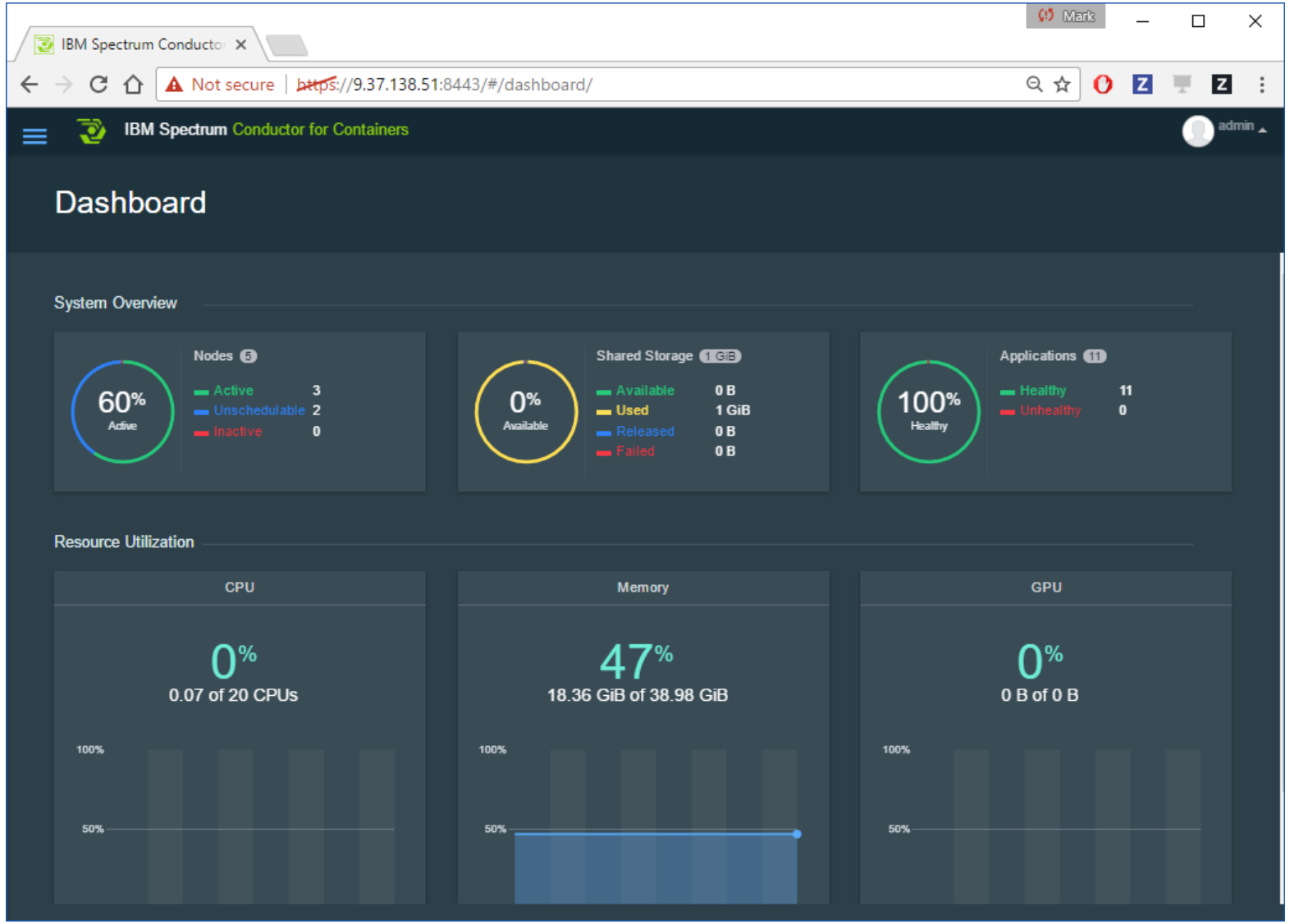
Virtualization through abstraction of service discovery & scaling across infra. service offerings

Business benefits:

1. **Frictionless application lifecycle** and user experience from **laptop through production**
2. **Accelerate software delivery** leveraging Continuous Delivery Pipeline
3. **Easily** connect existing assets with **Watson Cognitive services** to discover actionable insights
4. **Real-time diagnosis and resolution** of app infrastructure: minimizing downtime and maintaining SLAs
5. **Achieve cost optimization** in any cloud with **"seamless"** portability across popular cloud providers, public, dedicated, private, and hybrid

End-to-end user experience for development and deployment of Microservice and hybrid apps!

IBM Spectrum Conductor for Containers



App Center – currently Helm charts

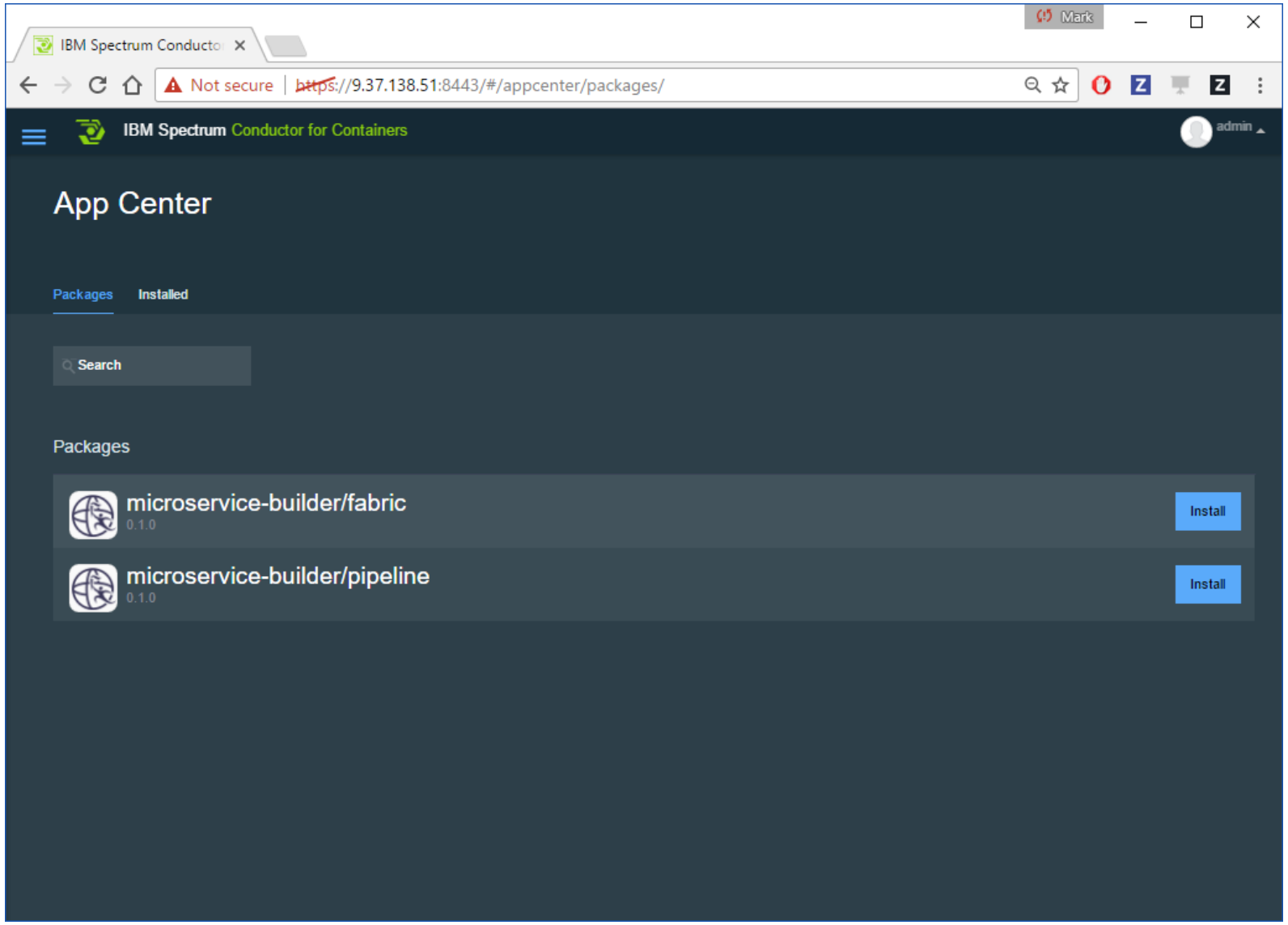
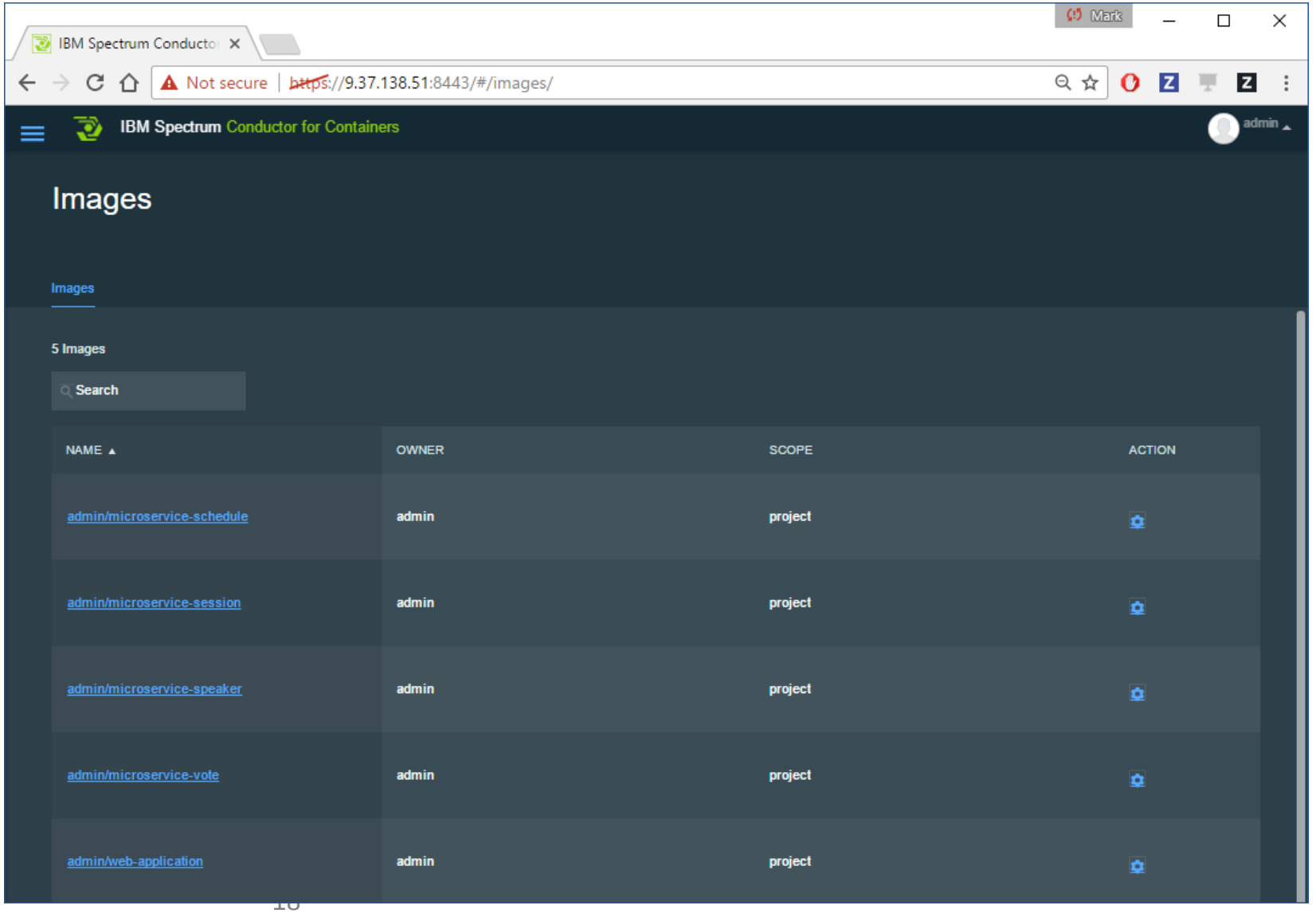
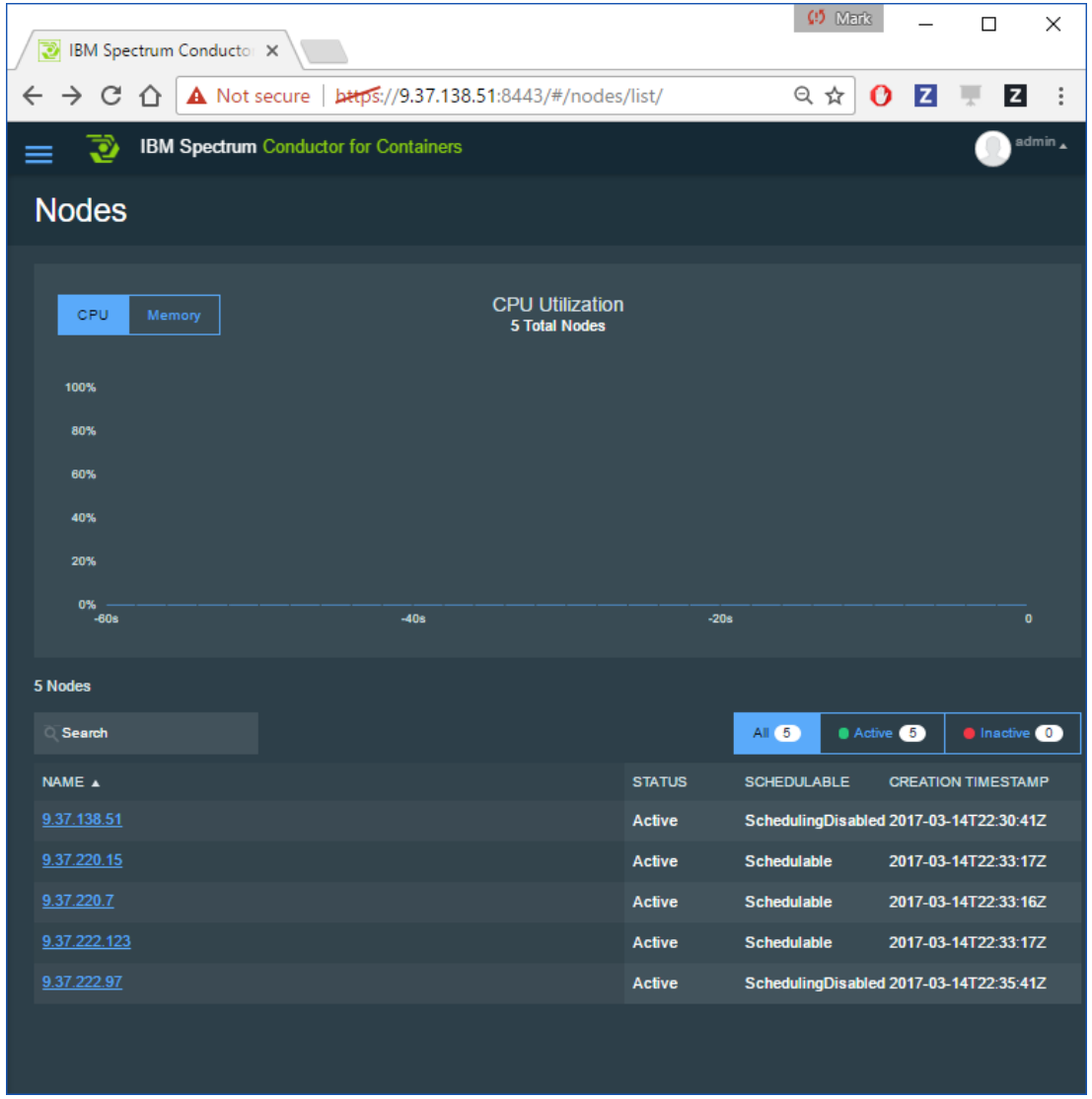


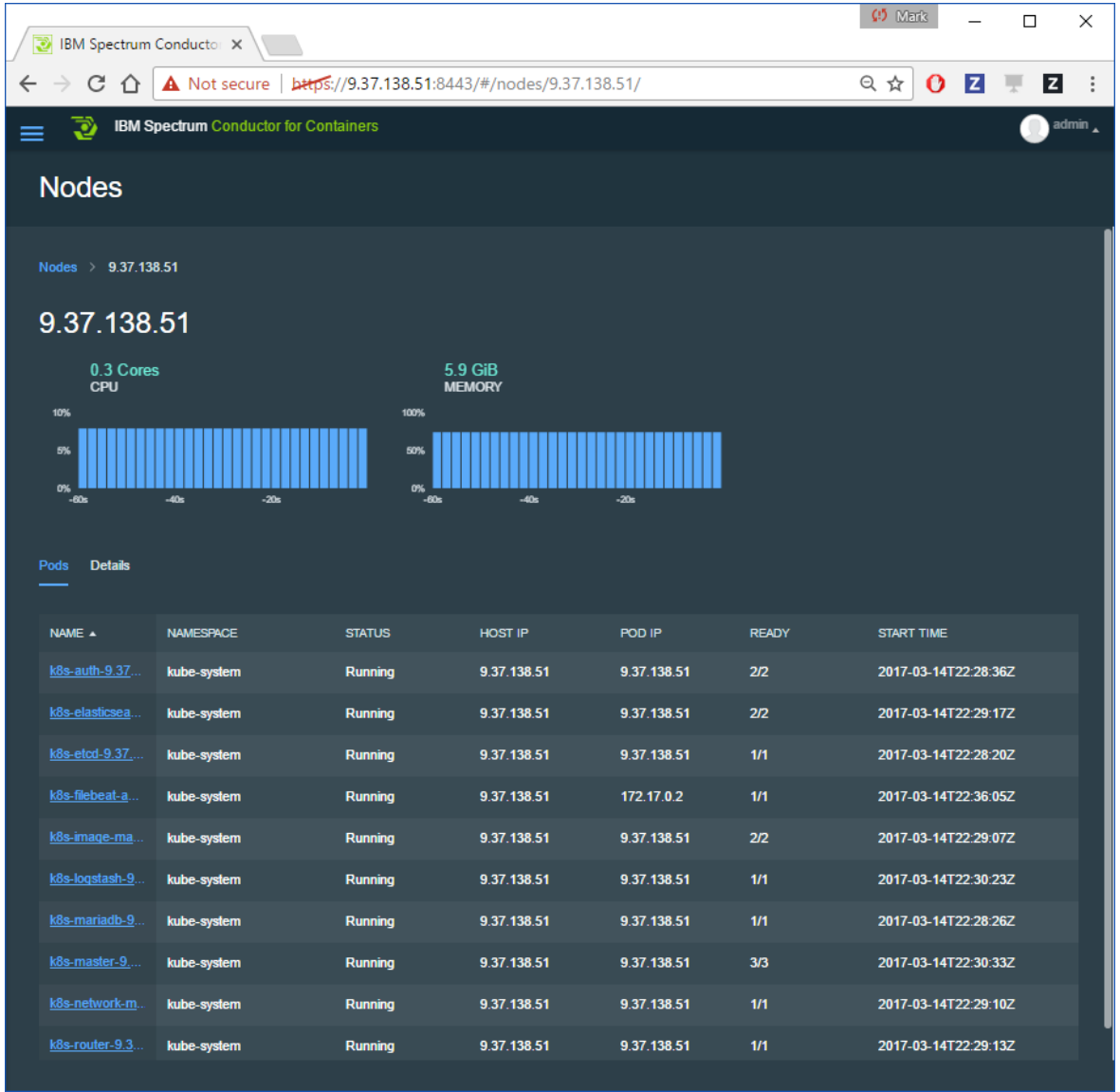
Image management



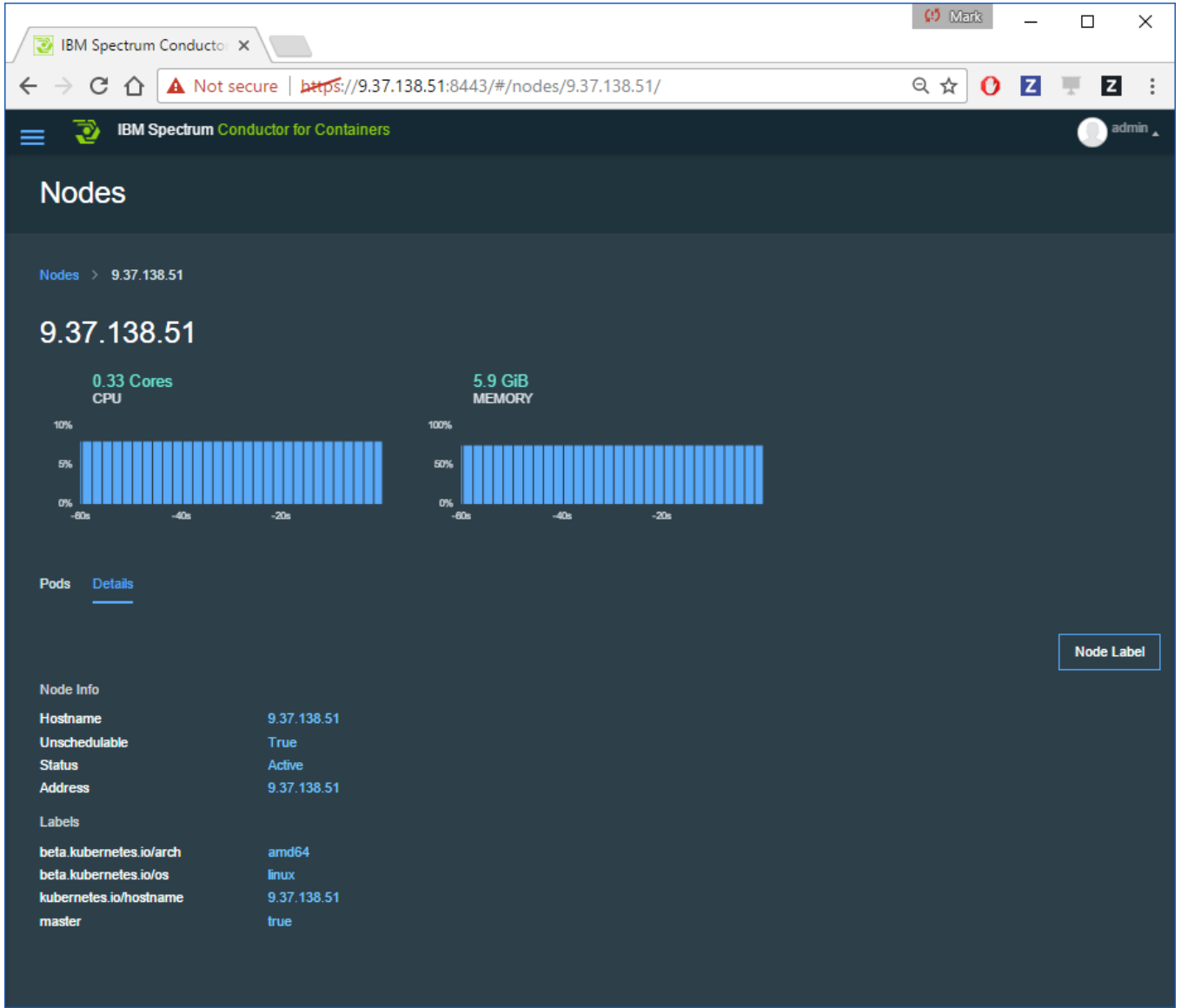
Node management



Pods on a node



Node details



On-Prem DevOps Pipeline

Deployed as Helm Chart

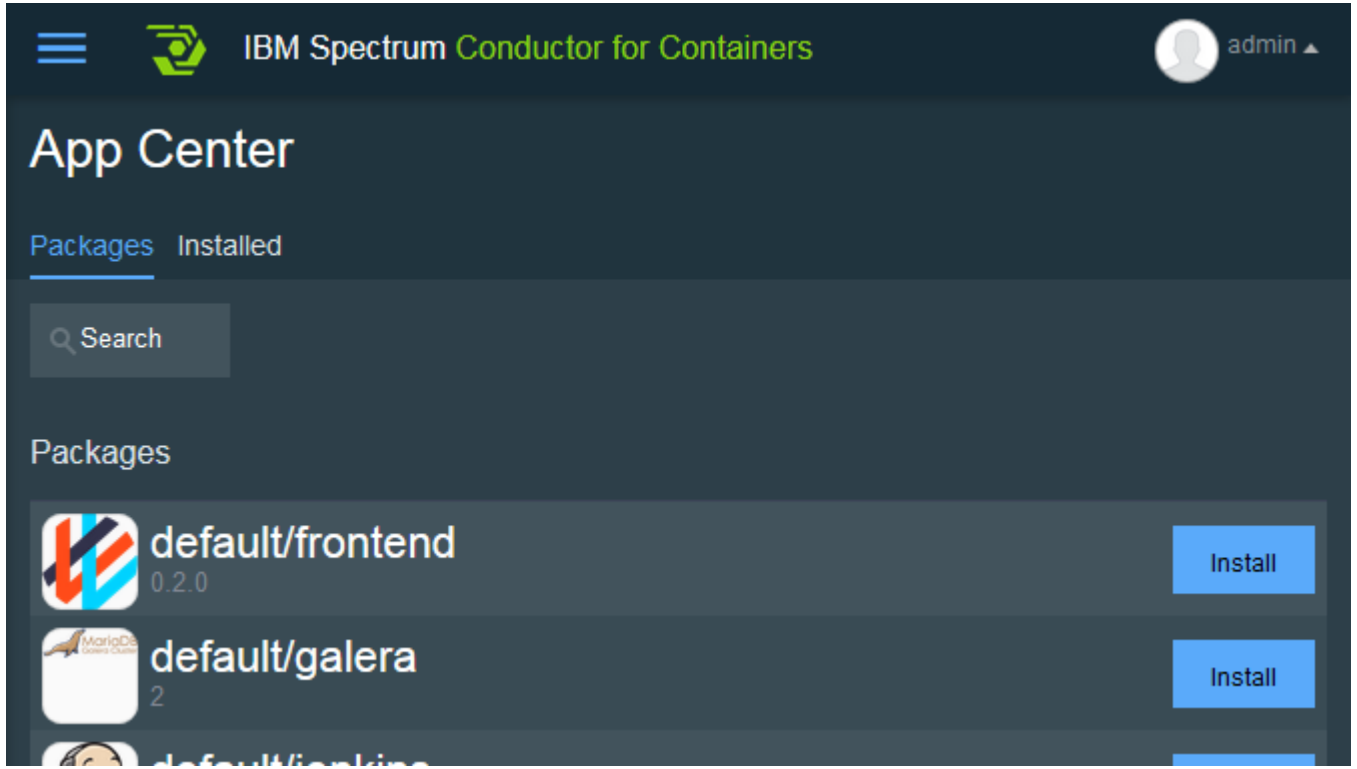
Deploy via Helm CLI or Conductor for Containers App Center

Required values:

- GitHub organization(s)
- GitHub OAuth User/Token
- GitHub App ID and Secret
- GitHub IDs for admin users
- Jenkins email address

Optionally:

- Regex for repositories



Pipeline – Deploy on Green

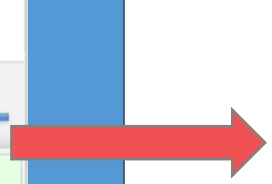
GitHub/GitHub Enterprise

- my-org
  - ▸ service-a
  - ▾ service-b
    - ▸ src
    - Dockerfile
    - Jenkinsfile
    - ▾ manifests
      - deploy.yaml
    - pom.xml

Jenkins

▸ my-org ▸ service-b ▸ master

Extract	Build	Test	Deploy
6s	1min 18s	27ms	902ms
6s	1min 18s	27ms	902ms



kubernetes

Pipeline – Gated Deployment

GitHub/GitHub Enterprise

- my-org
  - service-a
  - service-a-deploy
  - service-b
    - src
    - Dockerfile
    - Jenkinsfile
    - pom.xml
  - service-b-deploy
    - manifests
      - deploy.yaml
    - Jenkinsfile

Jenkins

▸ my-org ▸ service-b ▸ master

Extract	Build	Test	PushToRepo
6s	1min 18s	27ms	1s
6s	1min 18s	27ms	1s

▸ my-org ▸ service-b-deploy ▸ master

Extract	deployFromRepo
4s	902ms
4s	902ms

docker REGISTRY

kubernetes

Build Jenkinsfile

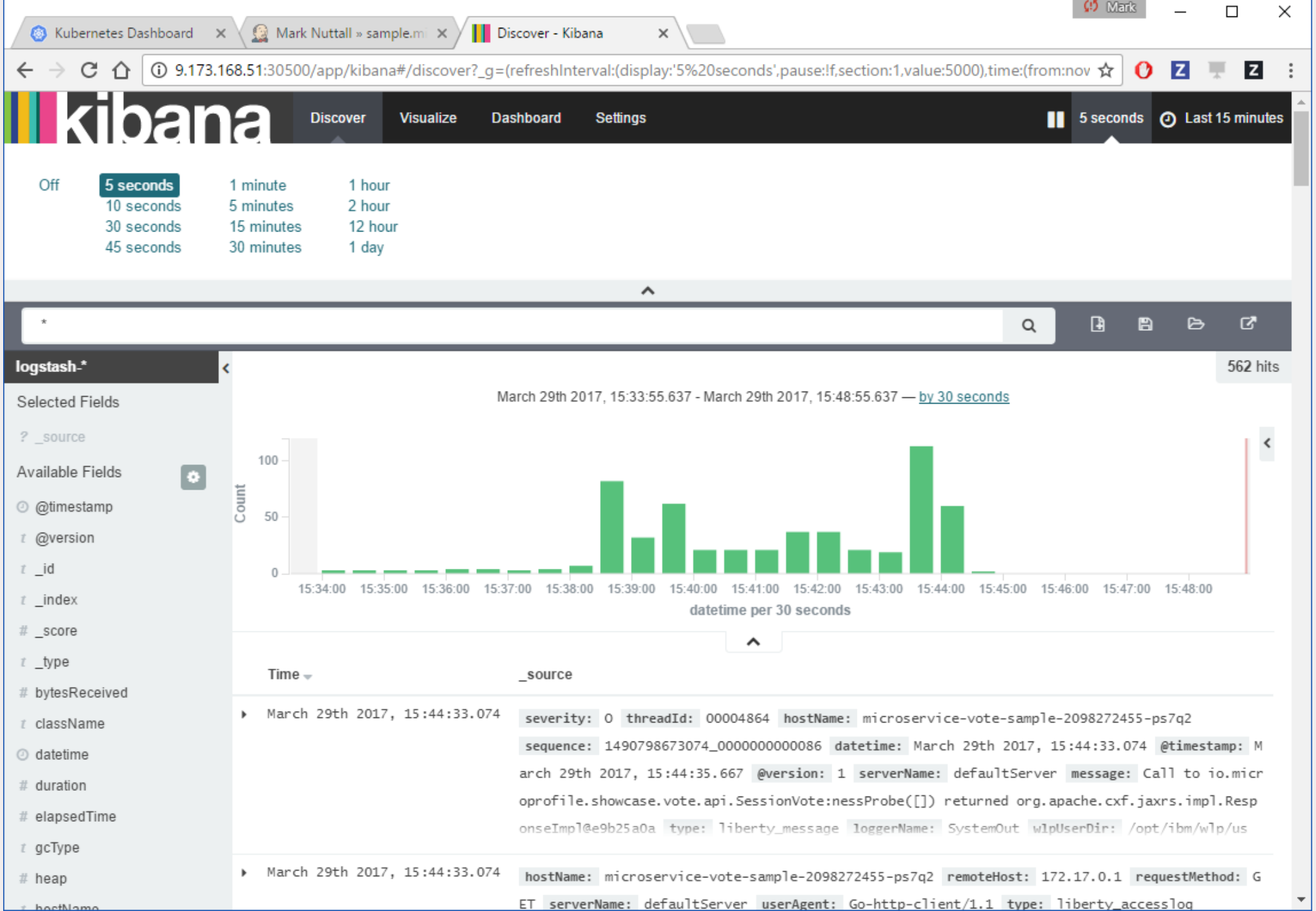
```
node('master') {
  def utils = load "/scripts/buildUtils.groovy"
  stage ('Extract') {
    checkout scm
  }
  stage ('Build') {
    utils.mavenBuild ('clean', 'package‘)
    utils.dockerBuild (‘back-end’)
  }
  stage ('PushToRepo') {
     utils.pushToRepo ('back-end')
   }
}
```

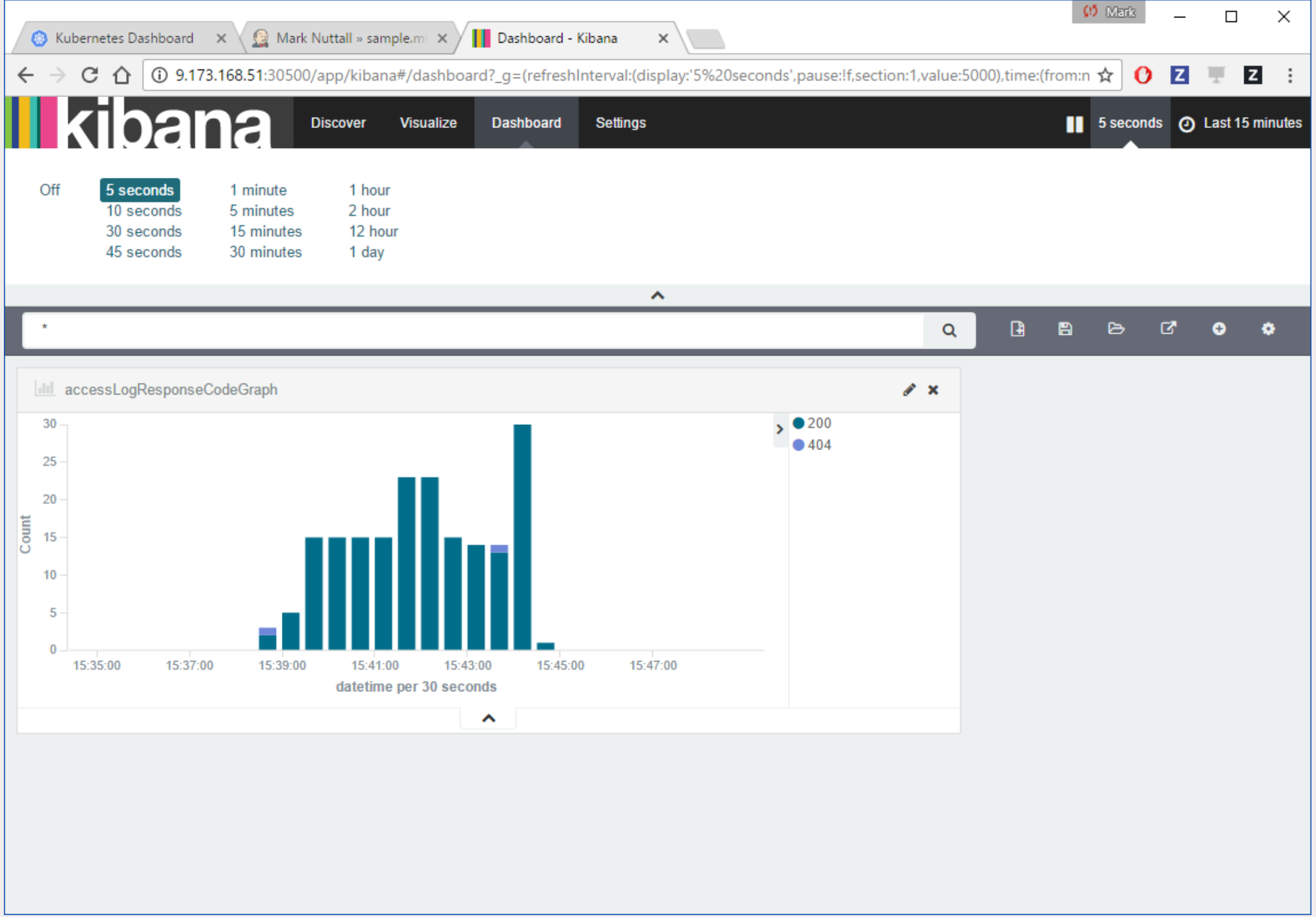
Deploy Jenkinsfile/YAML

```
node('master') {
def utils = load "/scripts/buildUtils.groovy"
  stage ('Extract') {
    checkout scm
  }
  stage ('deployFromRepo') {
    utils.deploy()
  }
}
```

```
spec:
    containers:
    - name: back-end
      image: registry/service:5e3e666
      ports:
        …
```

Demo


Kubernetes Dashboard
Mark Nuttall » sample.m
Discover - Kibana
9.173.168.51:30500/app/kibana#/discover?_g=(refreshInterval:(display:'5%20seconds',pause:!f,section:1,value:5000),time:(from:nov
kibana
Discover
Visualize
Dashboard
Settings
5 seconds
Last 15 minutes
Off
5 seconds
10 seconds
30 seconds
45 seconds
1 minute
5 minutes
15 minutes
30 minutes
1 hour
2 hour
12 hour
1 day
*
logstash-*
562 hits
Selected Fields
? _source
Available Fields
@timestamp
@version
_id
_index
_score
_type
bytesReceived
className
datetime
duration
elapsedTime
gcType
heap
March 29th 2017, 15:33:55.637 - March 29th 2017, 15:48:55.637 — by 30 seconds
Count
100
50
0
15:34:00
15:35:00
15:36:00
15:37:00
15:38:00
15:39:00
15:40:00
15:41:00
15:42:00
15:43:00
15:44:00
15:45:00
15:46:00
15:47:00
15:48:00
datetime per 30 seconds
Time
_source
March 29th 2017, 15:44:33.074
severity: 0 threadId: 00004864 hostName: microservice-vote-sample-2098272455-ps7q2
sequence: 1490798673074_0000000000086 datetime: March 29th 2017, 15:44:33.074 @timestamp: M
arch 29th 2017, 15:44:35.667 @version: 1 serverName: defaultServer message: Call to io.micr
oprofile.showcase.vote.api.SessionVote:nessProbe([]) returned org.apache.cxf.jaxrs.impl.Resp
onseImpl@e9b25a0a type: liberty_message loggerName: SystemOut wlpUserDir: /opt/ibm/wlp/us
March 29th 2017, 15:44:33.074
hostName: microservice-vote-sample-2098272455-ps7q2 remoteHost: 172.17.0.1 requestMethod: G
ET serverName: defaultServer userAgent: Go-http-client/1.1 type: liberty_accesslog


Kubernetes Dashboard
Mark Nuttall » sample.m
Dashboard - Kibana
Mark
9.173.168.51:30500/app/kibana#/dashboard?_g=(refreshInterval:(display:'5%20seconds',pause:!f,section:1,value:5000),time:(from:n
kibana
Discover
Visualize
Dashboard
Settings
5 seconds
Last 15 minutes
Off
5 seconds
10 seconds
30 seconds
45 seconds
1 minute
5 minutes
15 minutes
30 minutes
1 hour
2 hour
12 hour
1 day
*
accessLogResponseCodeGraph
200
404
Count
30
25
20
15
10
5
0
15:35:00
15:37:00
15:39:00
15:41:00
15:43:00
15:45:00
15:47:00
datetime per 30 seconds

kibana

Discover | Visualize | Dashboard | Settings

5 seconds | Last 15 minutes

Liberty-K4-20160216

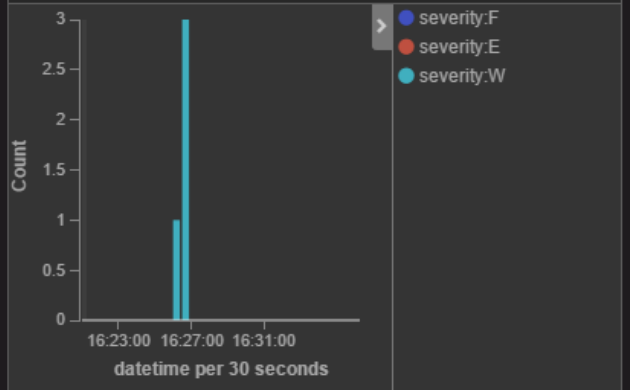
*

serversTable

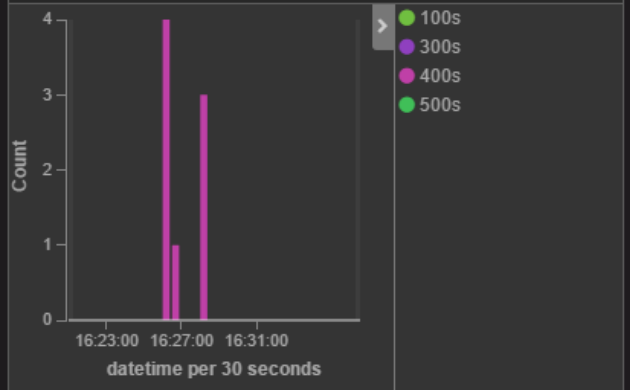
hostName: Descending	wlpUserDir: Descending	serverName: Descending	Count
microservice-test-deployment-2554010453-m6vfh	/opt/ibm/wlp/usr/	defaultServer	216
microservice-vote-sample-2098272455-r2jbd	/opt/ibm/wlp/usr/	defaultServer	193
microservice-schedule-sample-283123513-9wjkv	/opt/ibm/wlp/usr/	defaultServer	160
microservice-speaker-sample-3539148118-swpt6	/opt/ibm/wlp/usr/	defaultServer	127

accessLogServersPie

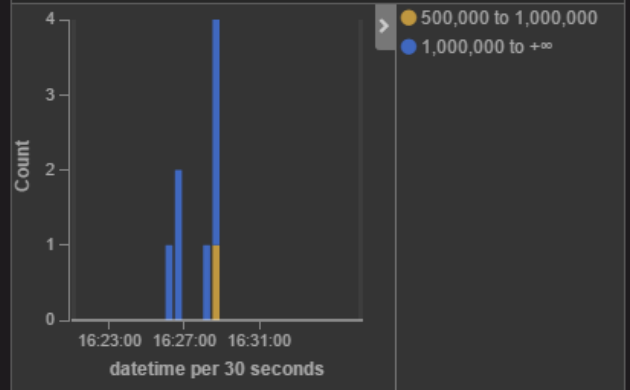
hotMessages



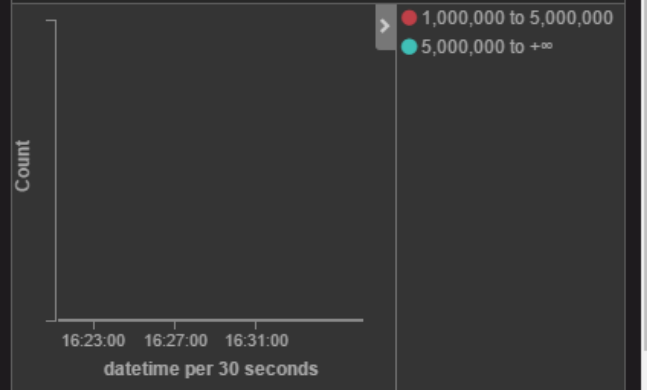
hotResponseCodes



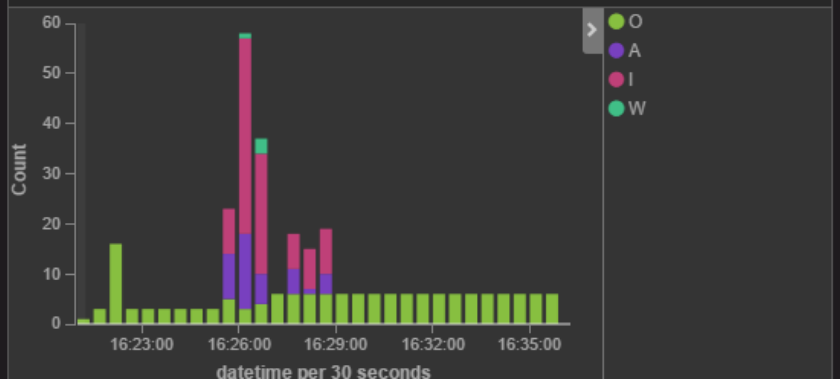
slowHits



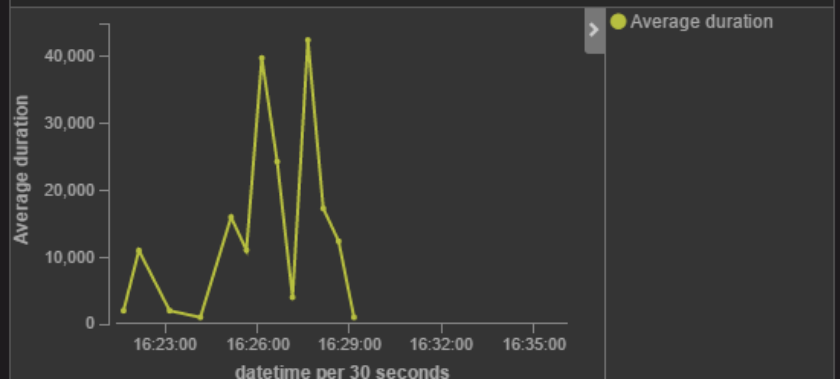
slowGC



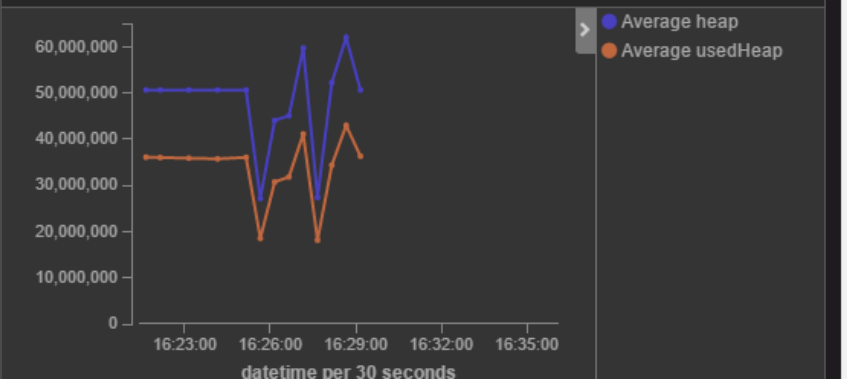
messageGraph



gcDuration

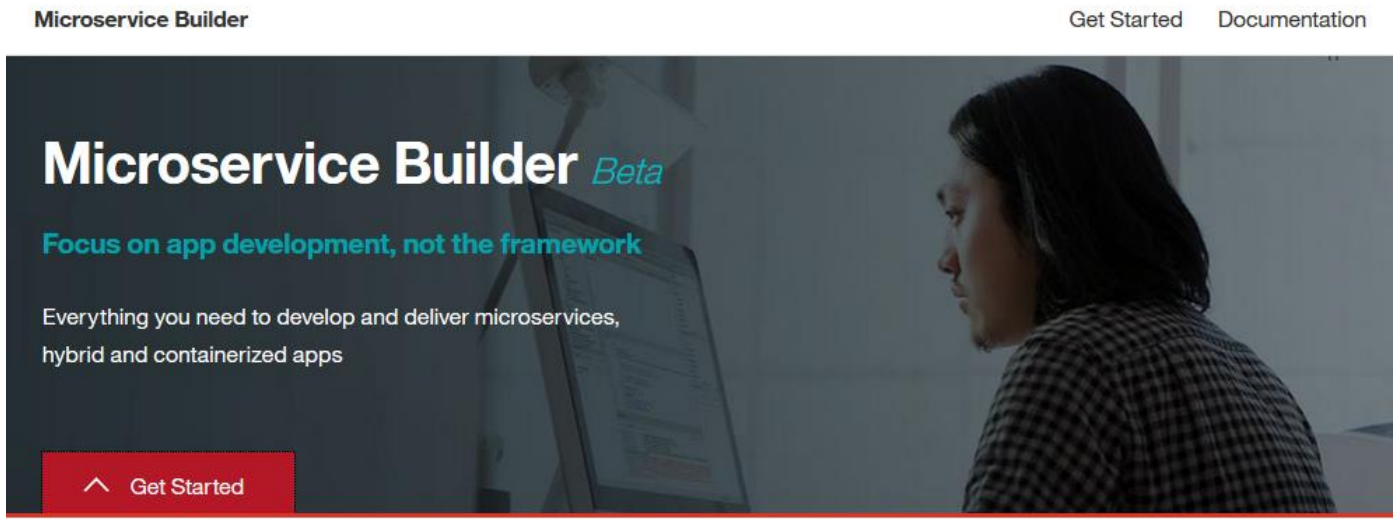


gcHeap



Microservice Builder (Beta)

End-to-End user experience to develop and deliver microservices, hybrid and containerized apps



Innovate with

SPEED

In 3 steps

Create and Run your microservices, hybrid and containerized apps

1. Download and Install
2. Create and Code
3. Run and Deploy

Set up your environment, fabric and DevOps pipeline in

Minutes

https://developer.ibm.com/microservice-builder

Questions?

Summary

Introducing Microservice Builder (Beta)

On-Prem DevOps Pipeline

Demo

Notices and disclaimers

U.S. Government Users Restricted Rights — use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM.

Information in these presentations (including information relating to products that have not yet been announced by IBM) has been reviewed for accuracy as of the date of initial publication and could include unintentional technical or typographical errors. IBM shall have no responsibility to update this information. **This document is distributed "as is" without any warranty, either express or implied. In no event shall IBM be liable for any damage arising from the use of this information, including but not limited to, loss of data, business interruption, loss of profit or loss of opportunity.** IBM products and services are warranted according to the terms and conditions of the agreements under which they are provided.

IBM products are manufactured from new parts or new and used parts. In some cases, a product may not be new and may have been previously installed. Regardless, our warranty terms apply."

Any statements regarding IBM's future direction, intent or product plans are subject to change or withdrawal without notice.

Performance data contained herein was generally obtained in a controlled, isolated environments. Customer examples are presented as illustrations of how those customers have used IBM products and the results they may have achieved. Actual performance, cost, savings or other results in other operating environments may vary.

References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business.

Workshops, sessions and associated materials may have been prepared by independent session speakers, and do not necessarily reflect the views of IBM. All materials and discussions are provided for informational purposes only, and are neither intended to, nor shall constitute legal or other guidance or advice to any individual participant or their specific situation.

It is the customer's responsibility to insure its own compliance with legal requirements and to obtain advice of competent legal counsel as to the identification and interpretation of any relevant laws and regulatory requirements that may affect the customer's business and any actions the customer may need to take to comply with such laws. IBM does not provide legal advice or represent or warrant that its services or products will ensure that the customer is in compliance with any law.

Notices and disclaimers continued

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products. IBM does not warrant the quality of any third-party products, or the ability of any such third-party products to interoperate with IBM's products. **IBM expressly disclaims all warranties, expressed or implied, including but not limited to, the implied warranties of merchantability and fitness for a particular**, **purpose.**

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents, copyrights, trademarks or other intellectual property right.

InterConnect
2017