Event Based Computing

The Next Frontier

Jon McDonald

Red Hat Senior Account SA

Red Hat User Group

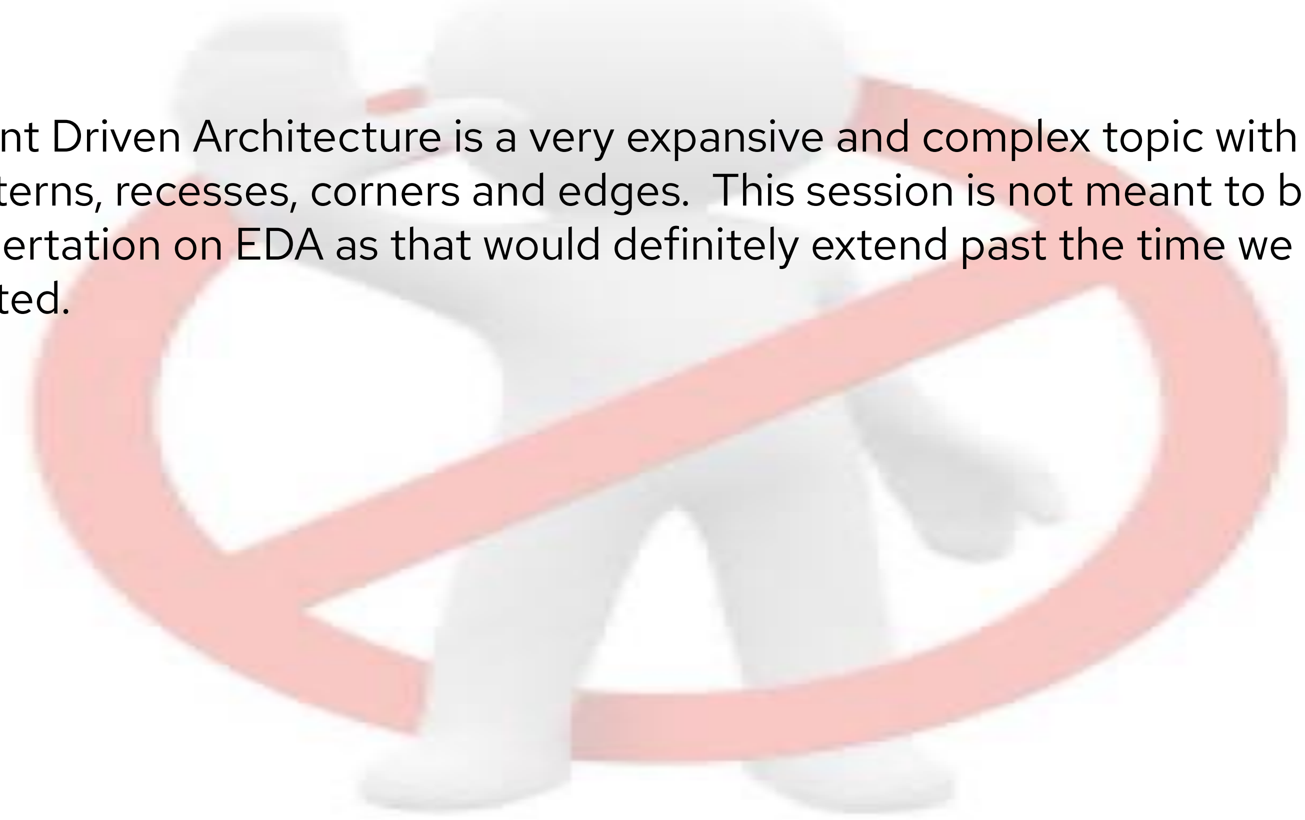
November 17, 2020

Red Hat

Objectives For This Session

- Put some high level definition and context behind event driven architecture (EDA)
- Identify the trends driving change making EDA a widely adopted pattern for applications
- Outline both non-technical and technical examples and real world applications for EDA
- Outline some of the common key technical components of an event driven architecture
- Outline the premise of the demonstration and demonstration architecture
- Demonstrate of how easy it can be to deploy EDA services and applications leveraging them when using the right platform
- Wrap up of what demo covered and links to resources and material to recreate demo and learn more

What this session is not meant to be..



Event Driven Architecture is a very expansive and complex topic with many patterns, recesses, corners and edges. This session is not meant to be a deep dissertation on EDA as that would definitely extend past the time we have alloted.

The What & Why

...Remind me again what this is about

The Challenge

All businesses across all industries today are undergoing “digital” transformation in an effort to enhance customer experience and differentiate themselves from their competition

The transformation efforts are highly reliant on technology solutions and capabilities, what some may refer to as the trend “making every business a technology company”

The pace at which organizations need to transform and differentiate is putting enormous pressure on their IT organizations to keep up, driving them to rethink how they deliver technology backed capabilities to the business

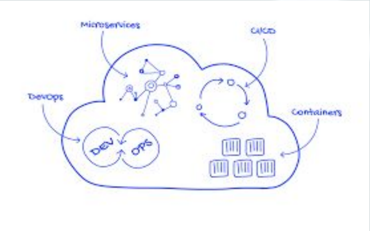
Without the ability to deliver new capabilities to the business quickly the business will suffer market loss and revenue decline to those who figured out how to tackle the challenge first.

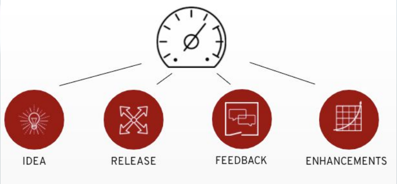
The Trends Driving Change...

Cloud Native….

...How applications are built, delivered, and deployed

Cloud native is an application development and deployment model where applications are developed in small reusable and distributed service components often referred to as micro-services which expose their interfaces as REST endpoints and are often deployed in containers. Their distributed nature makes them highly dependent on network stability. Their container deployment allows them to elastically scale but also requires them to be designed to sustain unplanned restarts. These dependency and deployment characteristics can subject them to loss of data when either network outage or service disruption occurs





Cloud/Hybrid Cloud….

...Where applications run

Cloud computing is a model of running application workloads on cloud based compute infrastructure in a utilization based cost model. This means that when applications are consuming compute resources that the meter is spinning and the more resource an application consumes the faster the meter spins and the more it is costing you.

aws

Azure

Google Cloud

IBM Cloud

Red Hat

What is an Event?

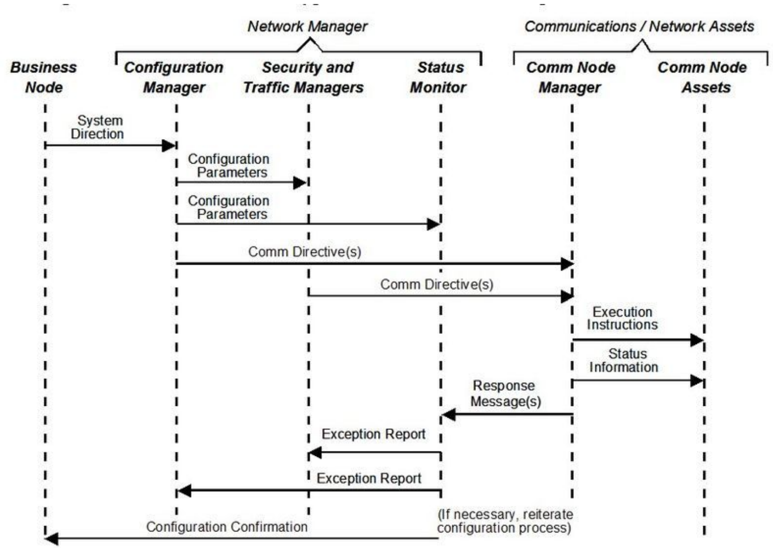
Event: An action or occurrence recognized by software, often originating asynchronously from the external environment, that may be handled by the software

CloudEvents: A specification for describing event data in a common way. CloudEvents seeks to dramatically simplify event declaration and delivery across services, platforms, and beyond! CloudEvents is a new effort and it's still under active development. The specification is now under the Cloud Native Computing Foundation.

Image: https://foto.wuestenigel.com/hand-ready-to-push-domino-pieces/

What Is Event Driven Architecture?

“Event-Driven Architecture (EDA) is a way of designing applications and services to ***respond to real-time information*** *based on the sending and receiving of information about individual events. EDA is* ***based on asynchronous non-blocking communication between event producers and event consumers*** *that are able to* ***release the resource consumption*** *while waiting for the next event to occur. Events enhance the* ***decoupling*** *of now well-defined bounded context (DDD) services technically and at runtime* ***becoming the first architectural consideration for cloud and container-native distributed systems****. Because more communication patterns are available, multiple consumers can receive events, simultaneously* ***lowering the latency and increasing the throughput.****”*



Why Event Driven Architecture?

Mirrors real world

The real world is event-driven. Systems generate and respond to events in everyday life (for example, the human central nervous system).

Reduced coupling

Traditional RPC-style service architecture results in tightly bound services. Changes to the application flow typically require service code changes. EDA allows new functionality to be added by adding services that consume existing event streams.

Encapsulation

Microservices concepts have grown in popularity due to the ability for service teams to develop services in isolation. EDA means that service designers need not be aware of how events are consumed.

Fine-grained scaling

Services can be independently scaled up and down to meet the event volume.

Near real-time latency

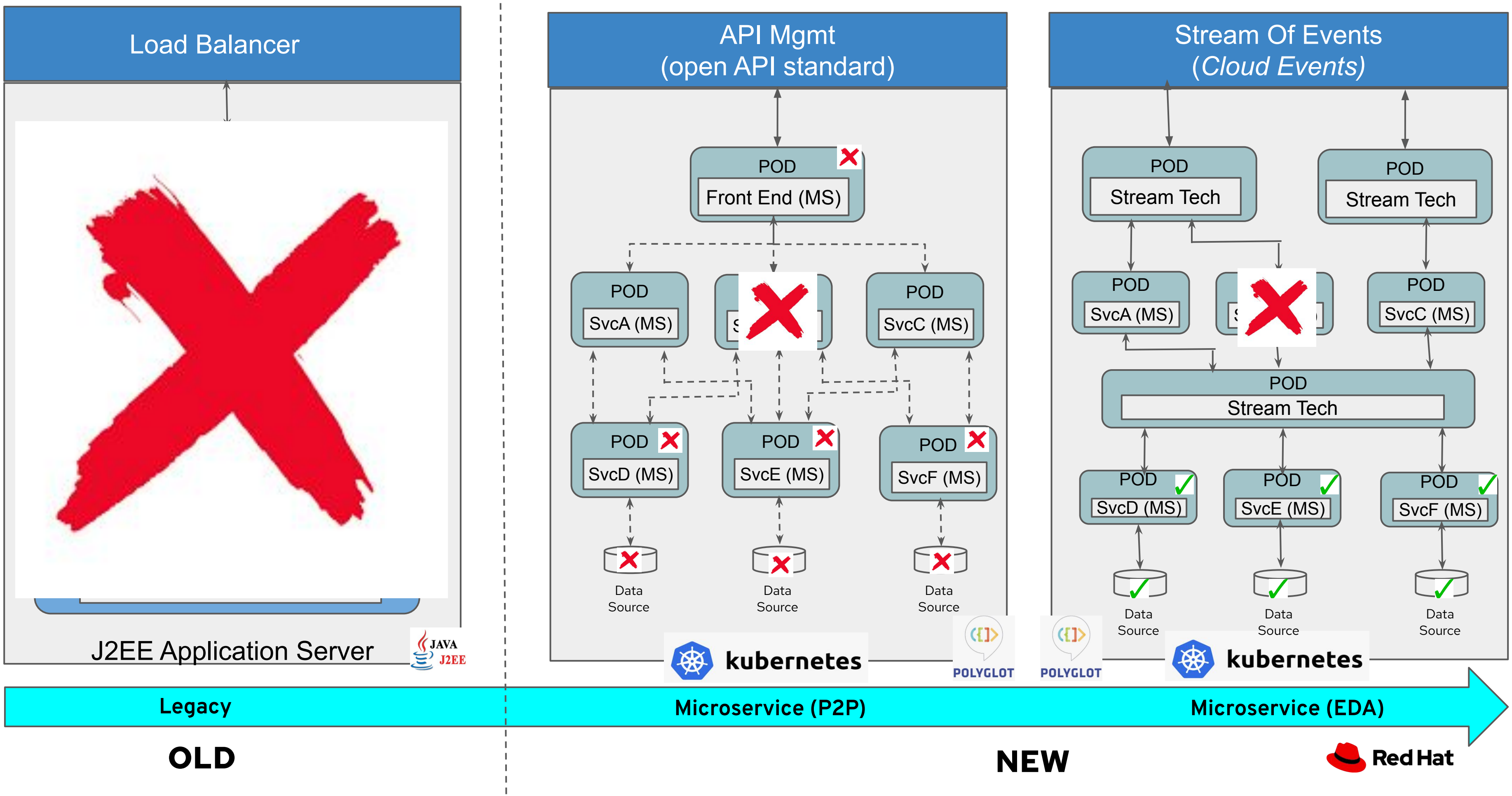
Customers increasingly expect a near real-time experience. Polling on APIs is a delicate trade-off between responsiveness and load. EDA allow apps to react in near real-time without compromise.

Cloud-native nature

EDA are designed to work perfectly in the distributed world, and cloud-native services are running separately on nodes/servers in the cloud.

Red Hat

Impact To The Evolution Of Applications Architecture



Load Balancer

J2EE Application Server

JAVA J2EE

API Mgmt
(open API standard)

POD
Front End (MS)

POD
SvcA (MS)

POD
SvcC (MS)

POD
SvcD (MS)

POD
SvcE (MS)

POD
SvcF (MS)

Data Source

Data Source

Data Source

kubernetes

POLYGLOT

POLYGLOT

Stream Of Events
(*Cloud Events*)

POD
Stream Tech

POD
Stream Tech

POD
SvcA (MS)

POD
SvcC (MS)

POD
Stream Tech

POD
SvcD (MS) ✓

POD
SvcE (MS) ✓

POD
SvcF (MS) ✓

Data Source ✓

Data Source ✓

Data Source ✓

kubernetes

Legacy

Microservice (P2P)

Microservice (EDA)

OLD

NEW

Red Hat

Solutions To Manage Change...

Cloud Native….

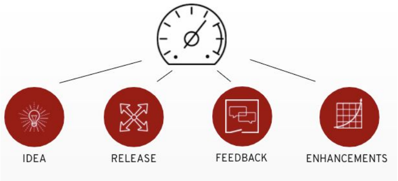
...How applications are built, delivered, and deployed

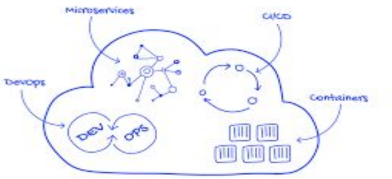
APACHE kafka®

Scalable Asynchronous High Throughput Streaming Messaging Integration

...Minimal Latency

...Near Zero Data Loss





Cloud/Hybrid Cloud….

...Where applications run

Serververless With Knative

...Scale to N Scale to 0

...Reduce The Meter Spin

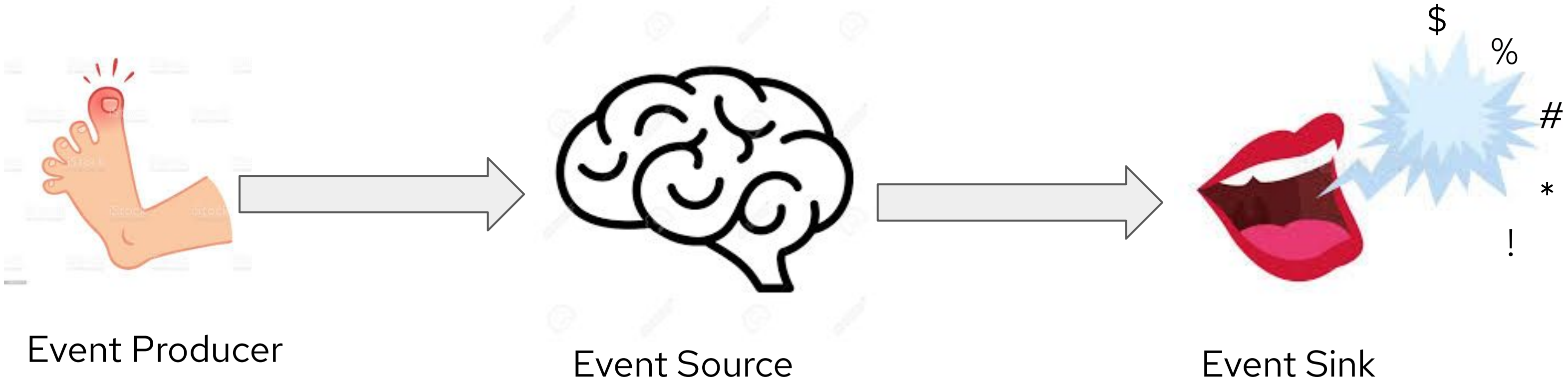
aws Azure Google Cloud IBM Cloud

Red Hat

Examples & Patterns

...What does this look like

The Human Body

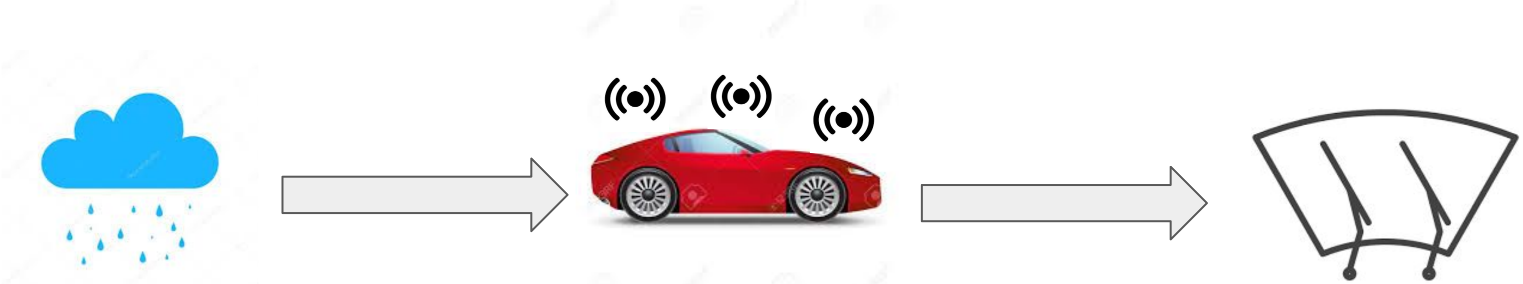


Event Producer

Event Source

Event Sink

The Automobile



Event Producer

Event Source

Event Sink

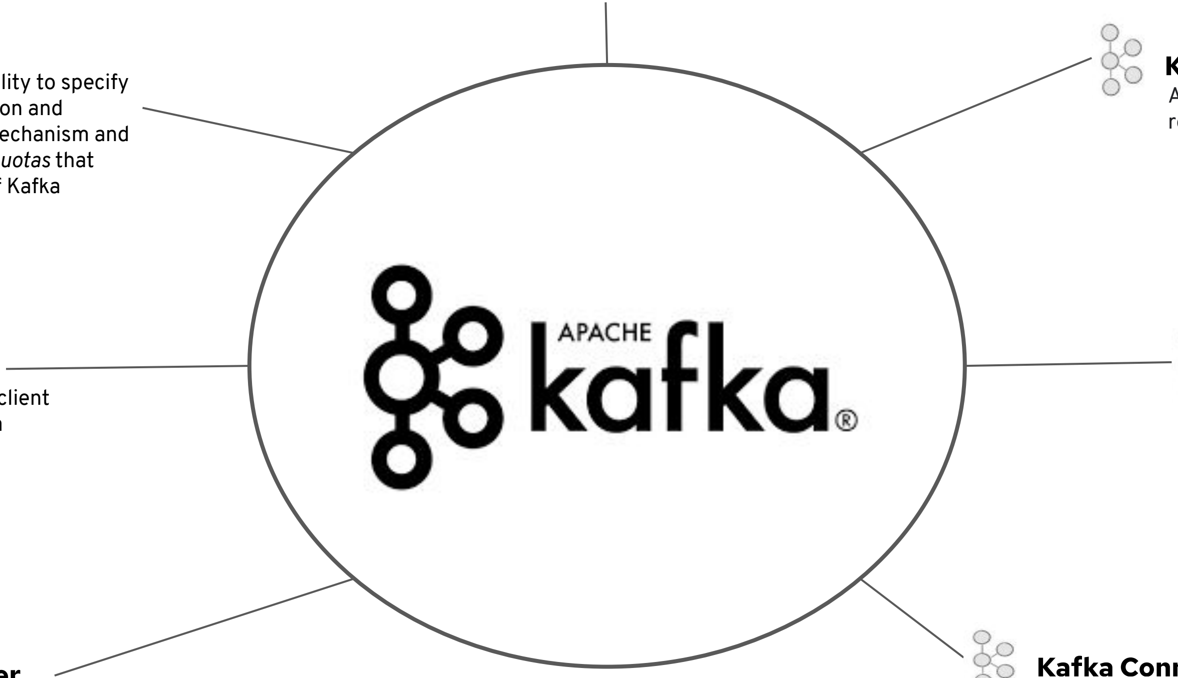
Real World Use Cases

- **Stream Processing**
  - IOT Based Environmental Monitoring In Manufacturing Plants To Proactively Respond Before Machine Failures Occur
  - Financial Fraud Detection based on historical and real-time analysis of customer behavior across multiple systems
- **Data Integration**
  - Captures streams of events or state/data changes and feeds these to other data systems.
  - Change Data Capture (CDC) replaces ETL batches with more timely and granular updates between separate databases
- **WebSite Activity Tracking**
  - Create user activity tracking pipeline as a set of real-time feeds to enable greater scalability, expose events as streams to applications for complex processing and analysis on customer interests and market trends to drive recommendations and forecast product demand.
- **Centralized Business Metrics**
  - Aggregation of statistics from distributed applications to produce centralized feeds of operational data that can be analyzed and statistics provided to distributed applications as an effective way of creating a centralized view and processing that data in different ways. Combine previously isolated silos of supply chain data to more accurately predict the timing of dependencies, and used to automatically optimize production schedules across multiple locations
- **Log Aggregation**
  - Publish log messages as events to build a simplified environment where multiple services can process log data in real time or asynchronously
- **Refactoring traditional P2P applications**
  - Provide a decoupled event driven architecture model to make them more fault tolerant, resilient, and scalable.

Key Components Of EDA

Apache Kafka

Red Hat
AMQ
Streams



Kafka Cluster

Broker and Zookeeper instances deployed in high availability and fault tolerant capacity

Kafka User

Provides the ability to specify the authentication and authorization mechanism and configure *user quotas* that control usage of Kafka resources.

Kafka Topic

A category/feed name to which records are stored and published.

Kafka Bridge

Provides a REST API for integrating HTTP based client applications with a Kafka cluster.

Kafka Connect

A tool for scalably and reliably streaming data between Apache Kafka and other systems

Kafka MirrorMaker

A tool for mirroring data between Kafka clusters. Consumes from a source cluster and produces to a destination cluster. A common use case for this kind of mirroring is to provide a replica in another datacenter

Kafka Connector

Ready-to-use components, which can help us to import data from external systems into Kafka topics and export data from Kafka topics into external systems

Asynchronous Data Streaming Integration For A Decoupled Event Driven World

Knative

OPENSHIFT Serverless

Knative Serving

Building Serverless Applications In The Language Of Your Choice with Auto-scaling and scale-to-zero capabilities

Knative Eventing

Binding Event Sources To Serverless Applications

ApiServerSource
Fires a new event each time a Kubernetes resource is created, updated or deleted

ContainerSource
Will instantiate container image(s) that can generate events until the ContainerSource is deleted

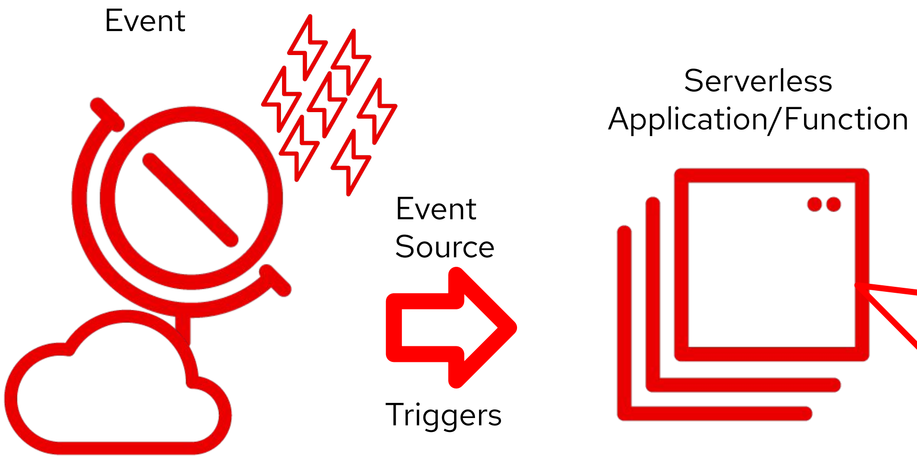
KafkaSource
Reads events from an Apache Kafka Cluster, and passes these events to a sink so that they can be consumed

PingSource
Produces events with a fixed payload on a specified Cron schedule

SinkBinding
Used to author new event sources using any of the familiar compute abstractions that Kubernetes makes available

https://knative.dev/docs/eventing/sources/

Serverless Function/Application Behaviour



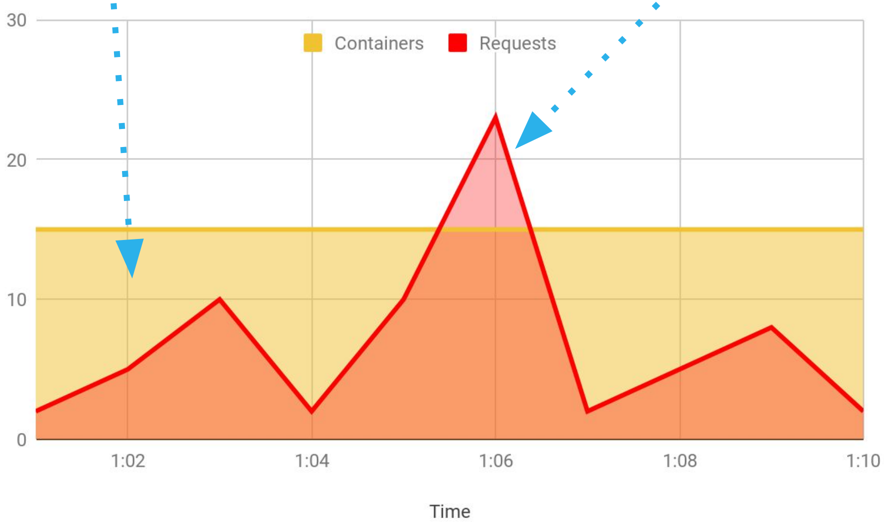
- Fast boot up time
- Stateless
- Natively consuming cloud events
- Code on the fly, no more packaging/build and auto deployment

https://quarkus.io/guides/funqy

Serverless Operational Benefits

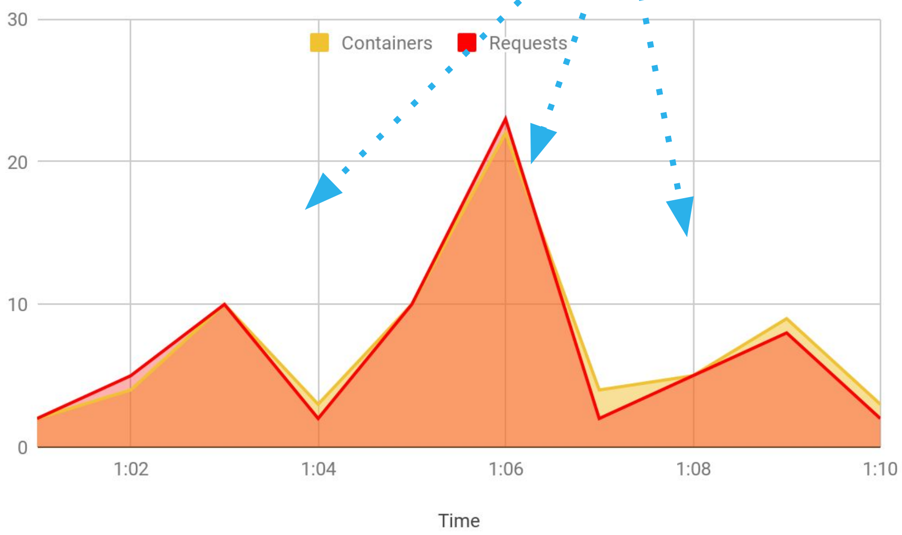
Over provisioning
Time in capacity planning
IT cost of idle resources

Under provisioning
Lost business revenue
Poor quality of service



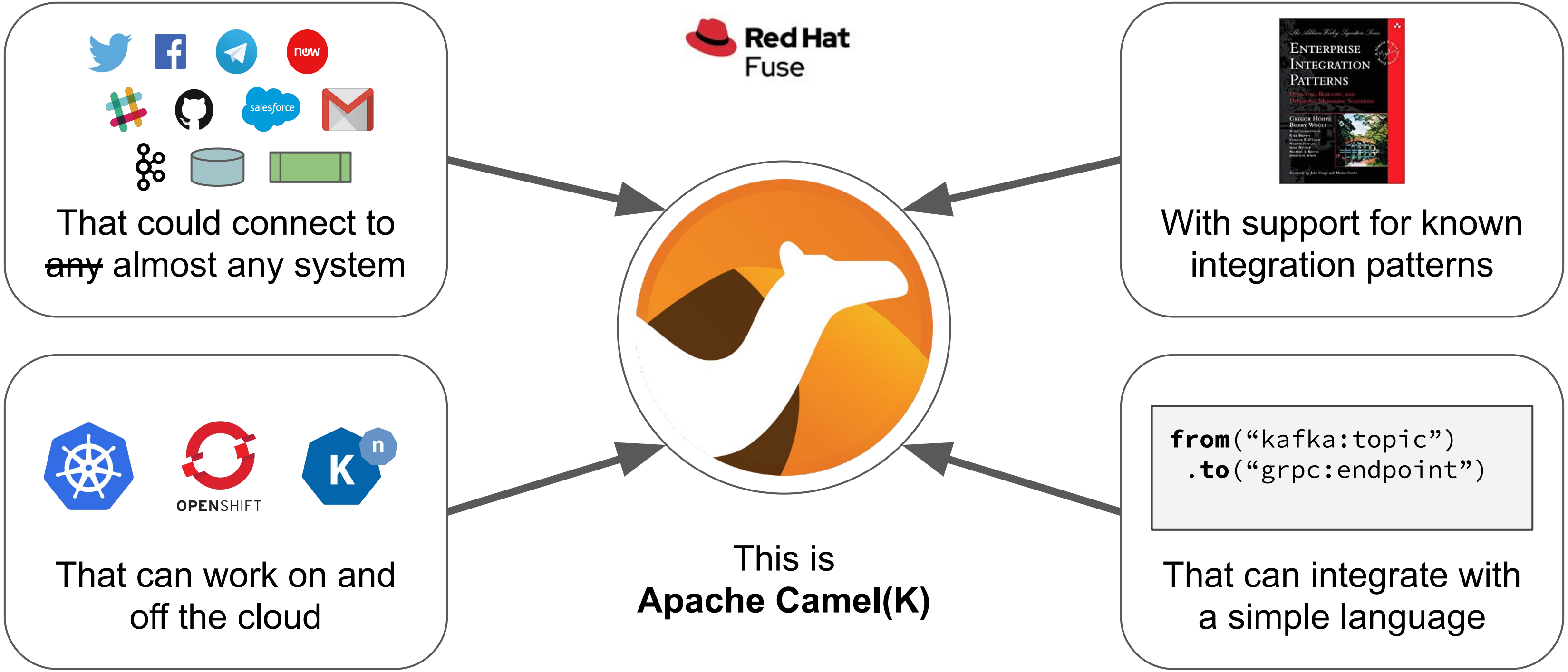
NOT Serverless

More applications
Direct line between IT
costs & business revenue



with Serverless

If we had the perfect tool to deal with integration...



That could connect to ~~any~~ almost any system

With support for known integration patterns

That can work on and off the cloud

This is
Apache Camel(K)

```
from("kafka:topic")
 .to("grpc:endpoint")
```

That can integrate with a simple language

Camel K For Developers

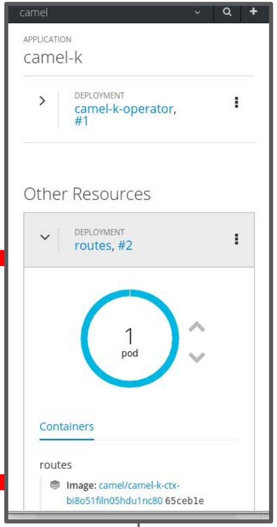
```
from("knative:channel/xxxx")
  .transform()...
  .to("kafka:topic")

from("kafka:topic")

.to("http:my-host/api/path")
```

1 **Create** integration file

2 **EXECUTE** CLI Tools

```
$ kamel run integration.java
```

3 **RUNNING Serverless** on OpenShift/Kubernetes



TEKTON CI/CD

Pipeline Resource

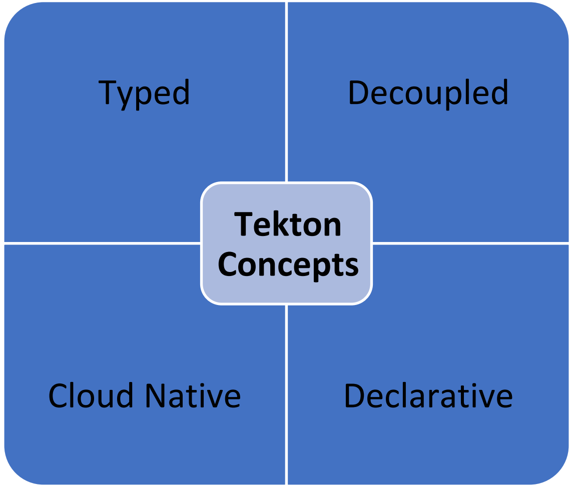
- Defines referable resources such as source code repo, container images

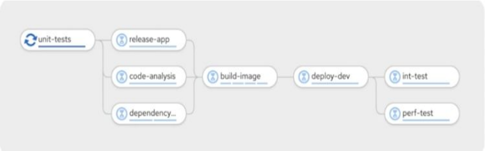
Step

- Runs commands within container(builder image)
- Mounts volumes, uses env vars

Task

- A list of steps that are executed in sequential order
- Takes inputs, outputs parameters





Task Run

- Runs a individual Task

Pipeline

- List of tasks defined to run in a certain order
- Takes inputs, outputs parameters

Pipeline Run

- Runs a Pipeline

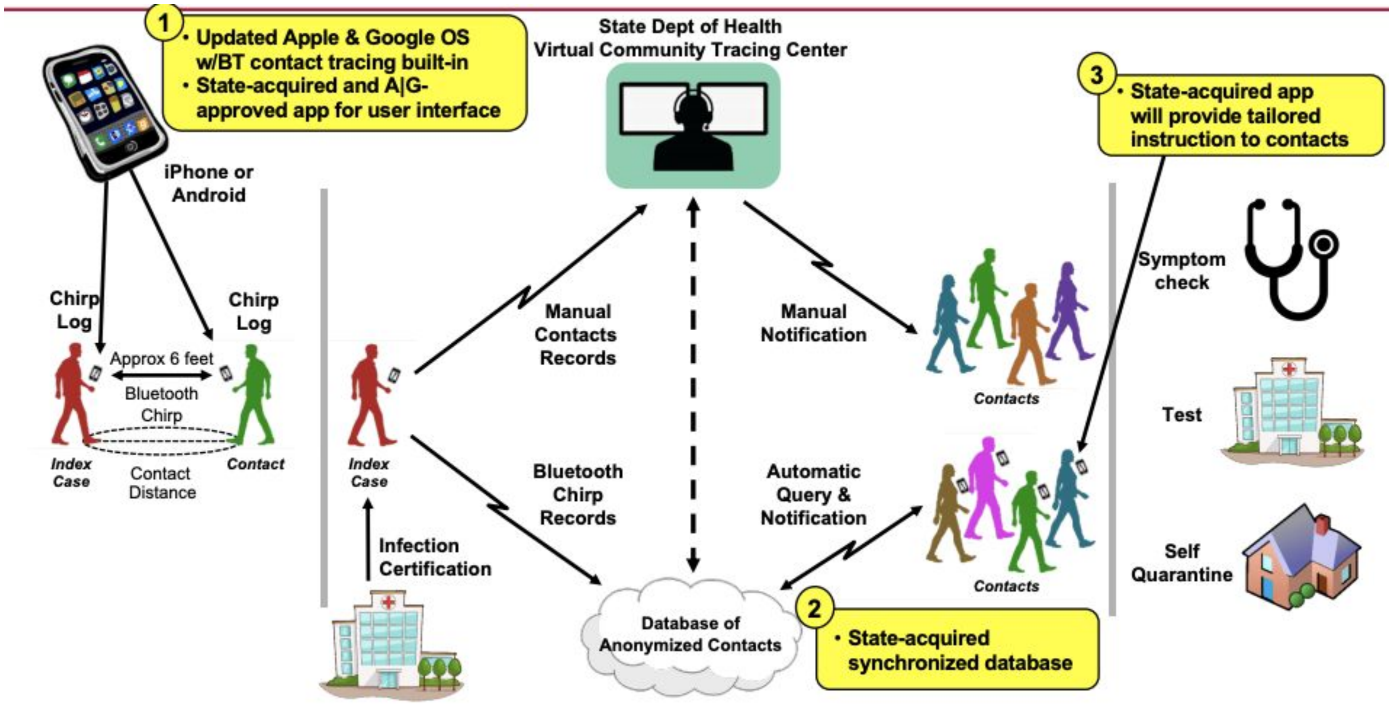
https://developers.redhat.com/blog/2020/11/03/argo-cd-and-tekton-match-made-in-kubernetes-heaven/?sc_cid=7013a0000026BcQAAU

Demo Pre-Flight Check

...Get Comfortable and Strap Yourself In The Plane Is Getting Ready To Take Off

Massachusetts
Institute of
Technology

Private Automated Contact Tracing



1
• Updated Apple & Google OS w/BT contact tracing built-in
• State-acquired and A|G-approved app for user interface
iPhone or Android
Chirp Log
Chirp Log
Approx 6 feet
Bluetooth Chirp
Index Case
Contact
Contact Distance
State Dept of Health
Virtual Community Tracing Center
Index Case
Infection Certification
Manual Contacts Records
Bluetooth Chirp Records
Database of Anonymized Contacts
Manual Notification
Automatic Query & Notification
Contacts
Contacts
2
• State-acquired synchronized database
3
• State-acquired app will provide tailored instruction to contacts
Symptom check
Test
Self Quarantine

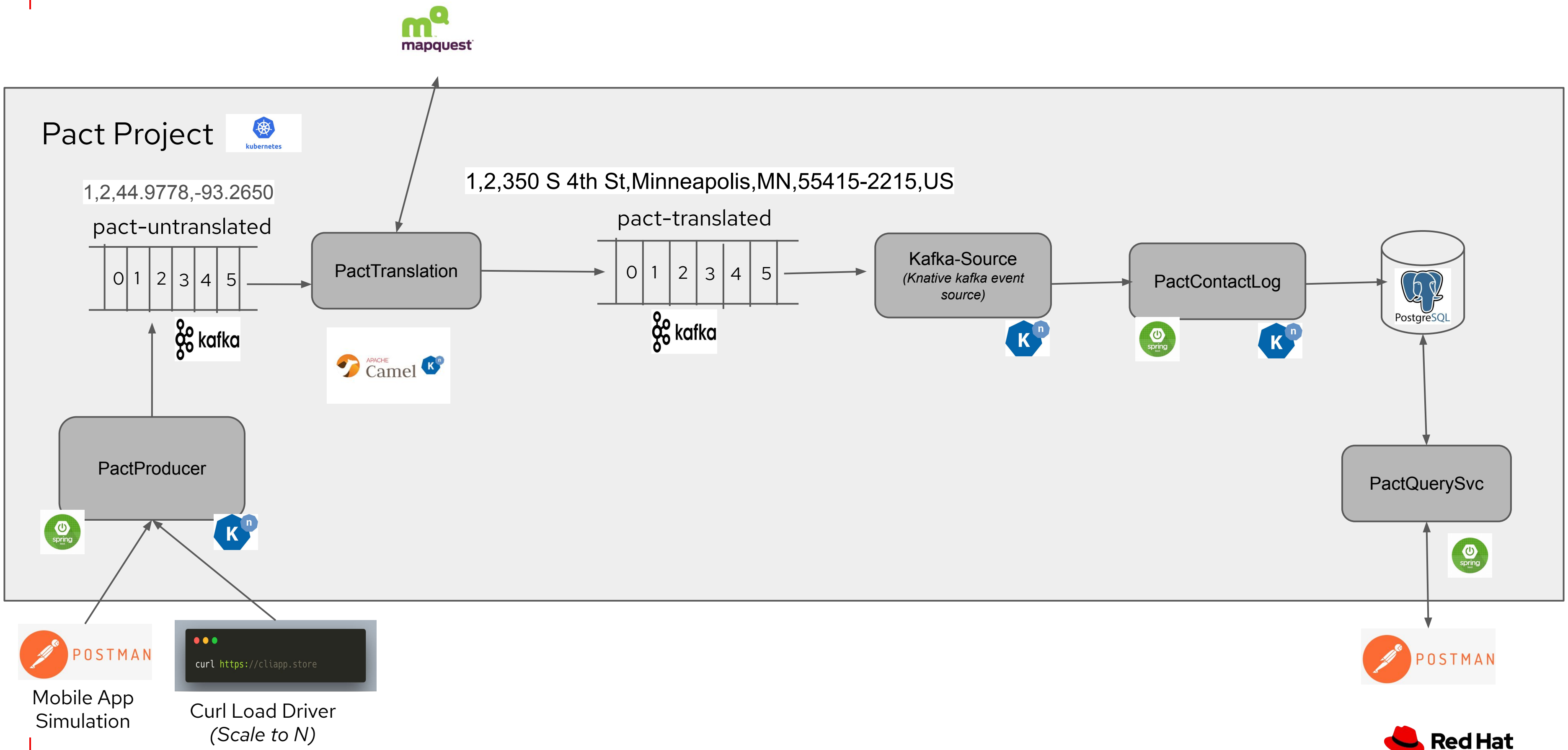
1 Contact Detection

2 COVID-19 Test Confirmed Positive

3 Exposure Query & Notification

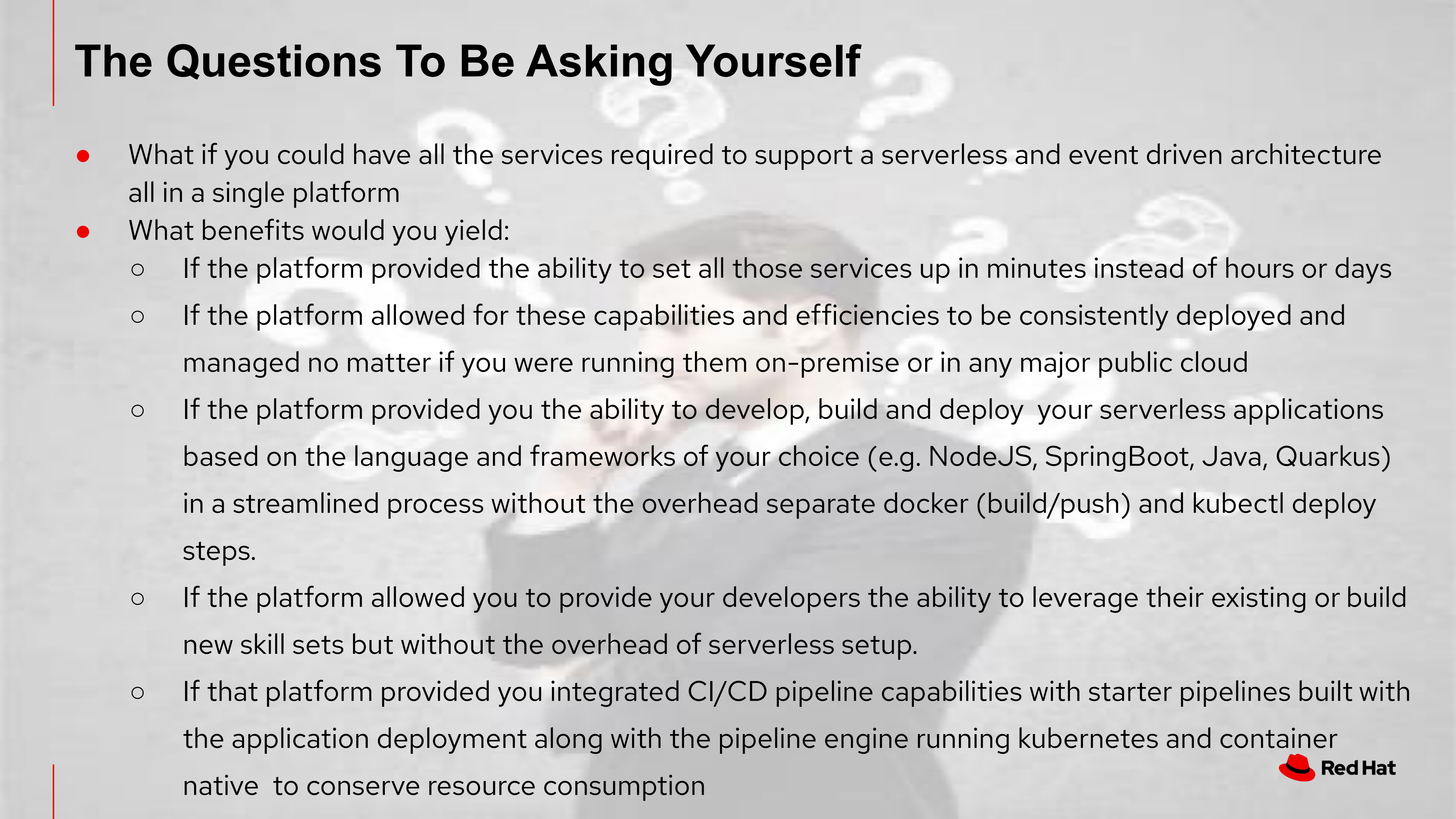
4 Contact Actions

Demo Architecture



```
curl https://cliapp.store
```

The Questions To Be Asking Yourself



- What if you could have all the services required to support a serverless and event driven architecture all in a single platform
- What benefits would you yield:
    - If the platform provided the ability to set all those services up in minutes instead of hours or days
    - If the platform allowed for these capabilities and efficiencies to be consistently deployed and managed no matter if you were running them on-premise or in any major public cloud
    - If the platform provided you the ability to develop, build and deploy your serverless applications based on the language and frameworks of your choice (e.g. NodeJS, SpringBoot, Java, Quarkus) in a streamlined process without the overhead separate docker (build/push) and kubectl deploy steps.
    - If the platform allowed you to provide your developers the ability to leverage their existing or build new skill sets but without the overhead of serverless setup.
    - If that platform provided you integrated CI/CD pipeline capabilities with starter pipelines built with the application deployment along with the pipeline engine running kubernetes and container native to conserve resource consumption

Should Lead You To Thinking About…..

Demo

...It’s About Time McDonald

Post Demo Landing

…Put your trays and seats in upright position

…Or…

…Is This Just About Over

What Did We Accomplish

(...In A Few Clicks Of A Button)

Operations

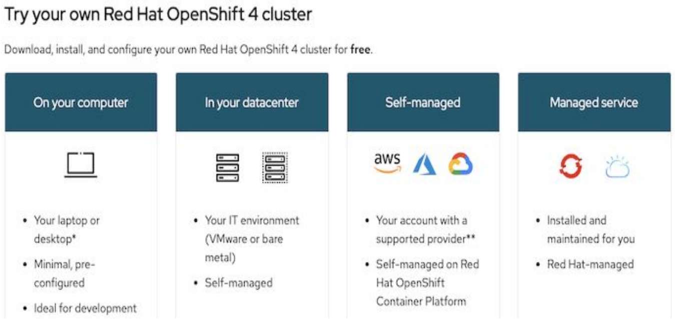
- Provided the development team compute, networking, and storage capacity in a few clicks of a button without having to go through waiting on a lengthy hardware procurement cycle, virtual machine, network, and storage setup
- Provide the development team with access to HA, Fault Tolerant Kafka services in a few clicks of a button
- Provided the development team with access to Serverless Enterprise Integration Capabilities in a few clicks of a button
- Provide the development team with access to Serverless services in a few clicks of a button
- Provided the development team with access to CI/CD Pipeline services in a few clicks of a button

Development

- Created a HA and Fault Tolerant Kafka Cluster and Kafka Topic in a few clicks...of a button
- Created a database in a self service capacity in a few clicks of a button
- Deployed containerized event driven serverless application services from source code in a few clicks of a button
- Created a starter serverless CI/CD pipeline flow as streamlined component of an application deployment
- Deployed a kubernetes native integration service from a simple command line
- Created an event source to bind Kafka topic as an event source to a serverless application as an event sink in a few clicks of a button
- Easily modified an applications source code, unit tested changes, committed them to SCM, triggered a pipeline to build and deploy changes, all in a streamlined process.

Resources

https://www.openshift.com/try



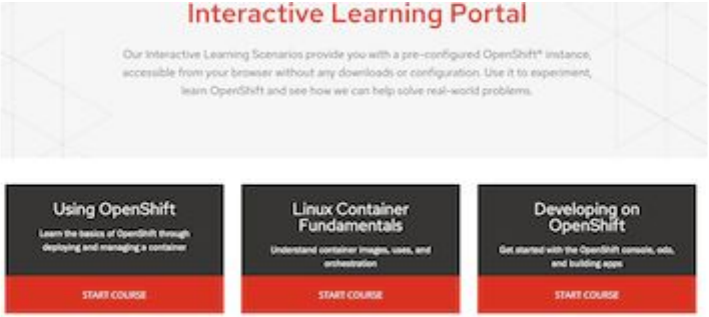
Try your own Red Hat OpenShift 4 cluster

Download, install, and configure your own Red Hat OpenShift 4 cluster for **free**.

On your computer	In your datacenter	Self-managed	Managed service
• Your laptop or desktop* • Minimal, pre-configured • Ideal for development	• Your IT environment (VMware or bare metal) • Self-managed	• Your account with a supported provider** • Self-managed on Red Hat OpenShift Container Platform	• Installed and maintained for you • Red Hat-managed

Free Access To Enterprise Kubernetes Platforms

https://learn.openshift.com/



Interactive Learning Portal

Our Interactive Learning Scenarios provide you with a pre-configured OpenShift* instance, accessible from your browser without any downloads or configuration. Use it to experiment, learn OpenShift and see how we can help solve real-world problems.

Using OpenShift

Learn the basics of OpenShift through deploying and managing a container

START COURSE

Linux Container Fundamentals

Understand container images, uses, and orchestration

START COURSE

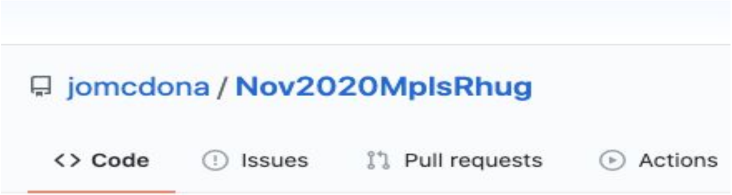
Developing on OpenShift

Get started with the OpenShift console, odo, and building apps

START COURSE

Cloud Native Interactive Guided Learning

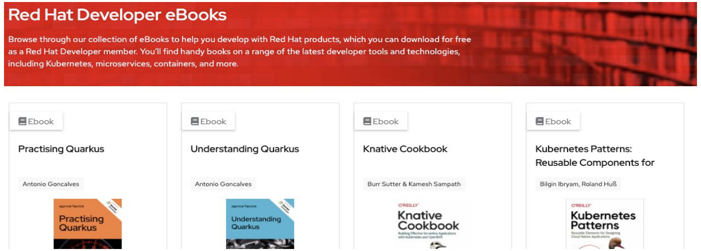
https://github.com/jomcdona/Nov2020MplsRhug



jomcdona / Nov2020MplsRhug

Demonstration Resources

https://developers.redhat.com/ebooks



Red Hat Developer eBooks

Browse through our collection of eBooks to help you develop with Red Hat products, which you can download for free as a Red Hat Developer member. You'll find handy books on a range of the latest developer tools and technologies, including Kubernetes, microservices, containers, and more.

Ebook	Ebook	Ebook	Ebook
Practising Quarkus	Understanding Quarkus	Knative Cookbook	Kubernetes Patterns: Reusable Components for
Antonio Goncalves	Antonio Goncalves	Burr Sutter & Kamesh Sampath	Bilgin Ibryam, Roland Huß

Free E-Books

Thank you

Red Hat is the world's leading provider of

enterprise open source software solutions.

Award-winning support, training, and consulting

services make

Red Hat a trusted adviser to the Fortune 500.

in linkedin.com/company/red-hat

youtube.com/user/RedHatVideos

f facebook.com/redhatinc

twitter.com/RedHat