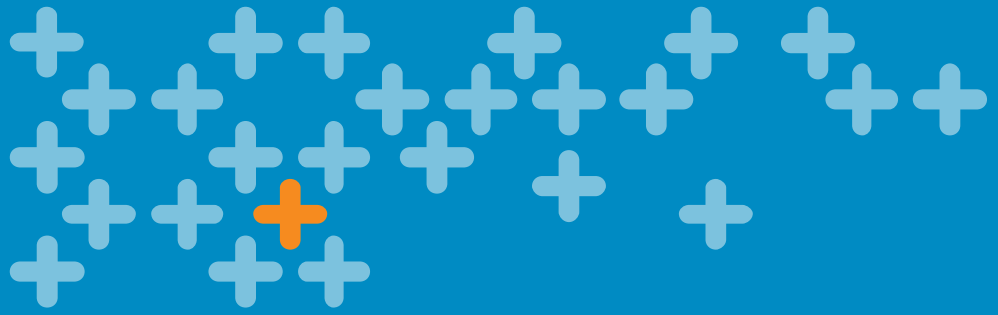


Event-driven Computing with Kubernetes

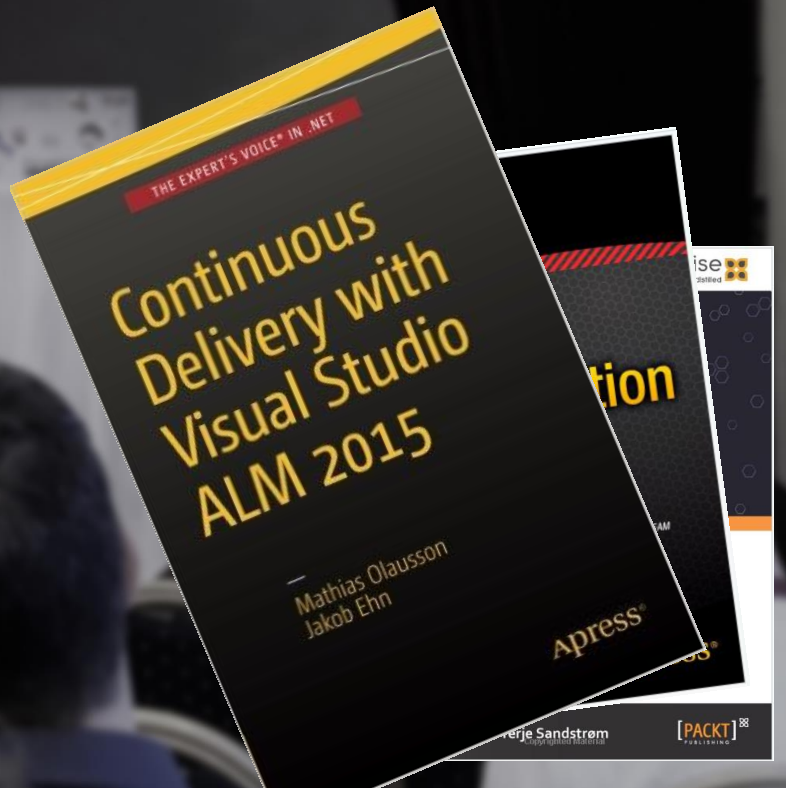
Jakob Ehn
@jakobehn
https://blog.ehn.nu

active SOLUTION

Microsoft Azure MVP

https://blog.ehn.nu

@jakobehn



Event-Based Automation

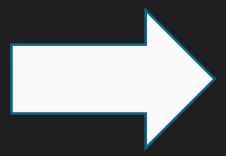
ServiceBus message sent

Push to GitHub

Container image pushed

Pull Request approved

.....



Trigger build

Create/Tear down environments

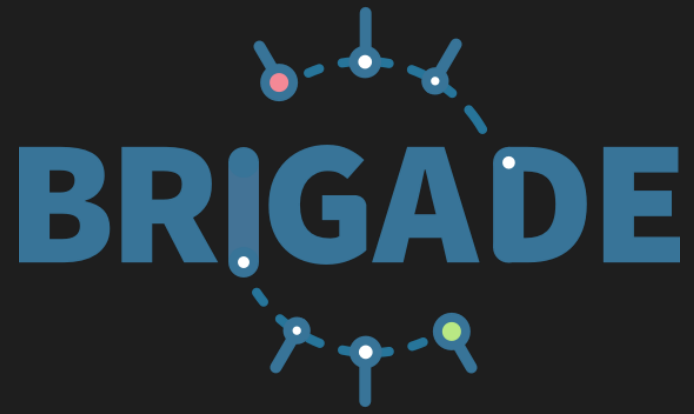
Run smoke tests

Scale up/down

....

BRIGADE

KEDA

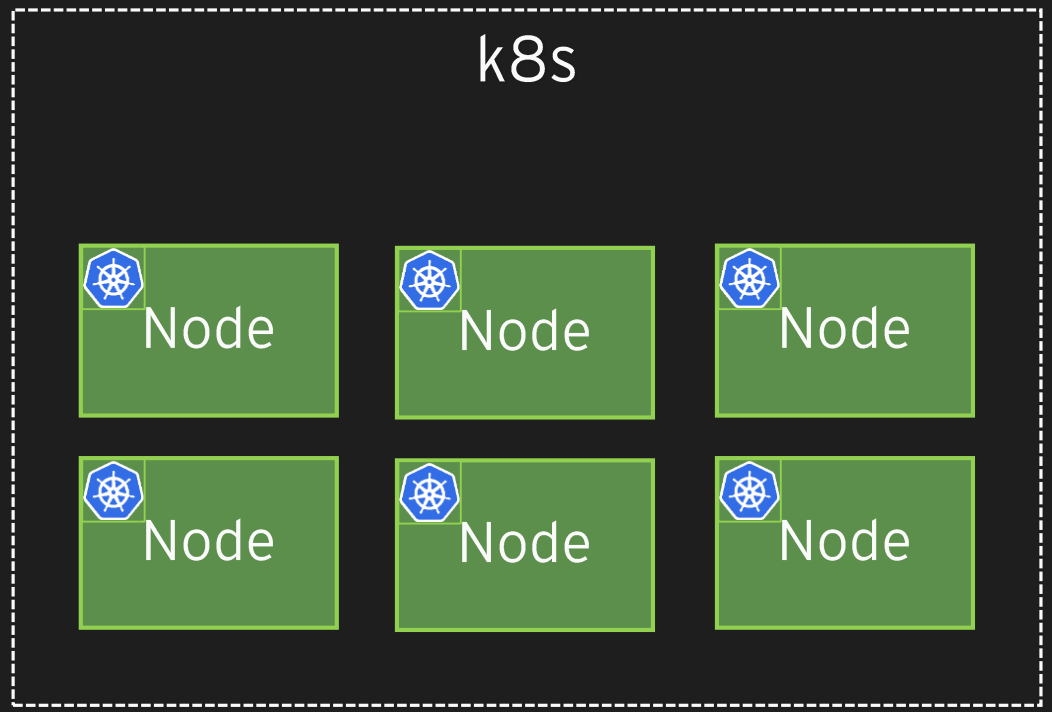


Event-driven scripting
for Kubernetes

Kubernetes-based
Event Driven Autoscaling

Chain together containers to create workflows

Runs inside your Kubernetes cluster



Use JavaScript to create pipelines

```
var job = new Job("say-hello", "alpine:3.8");
job.tasks = [
  "echo Hello",
  "echo World"
];

job.run();
```

Integrates with various event providers
(or roll your own)

GitHub

Azure Event Grid

Azure Container Registry

Kubernetes

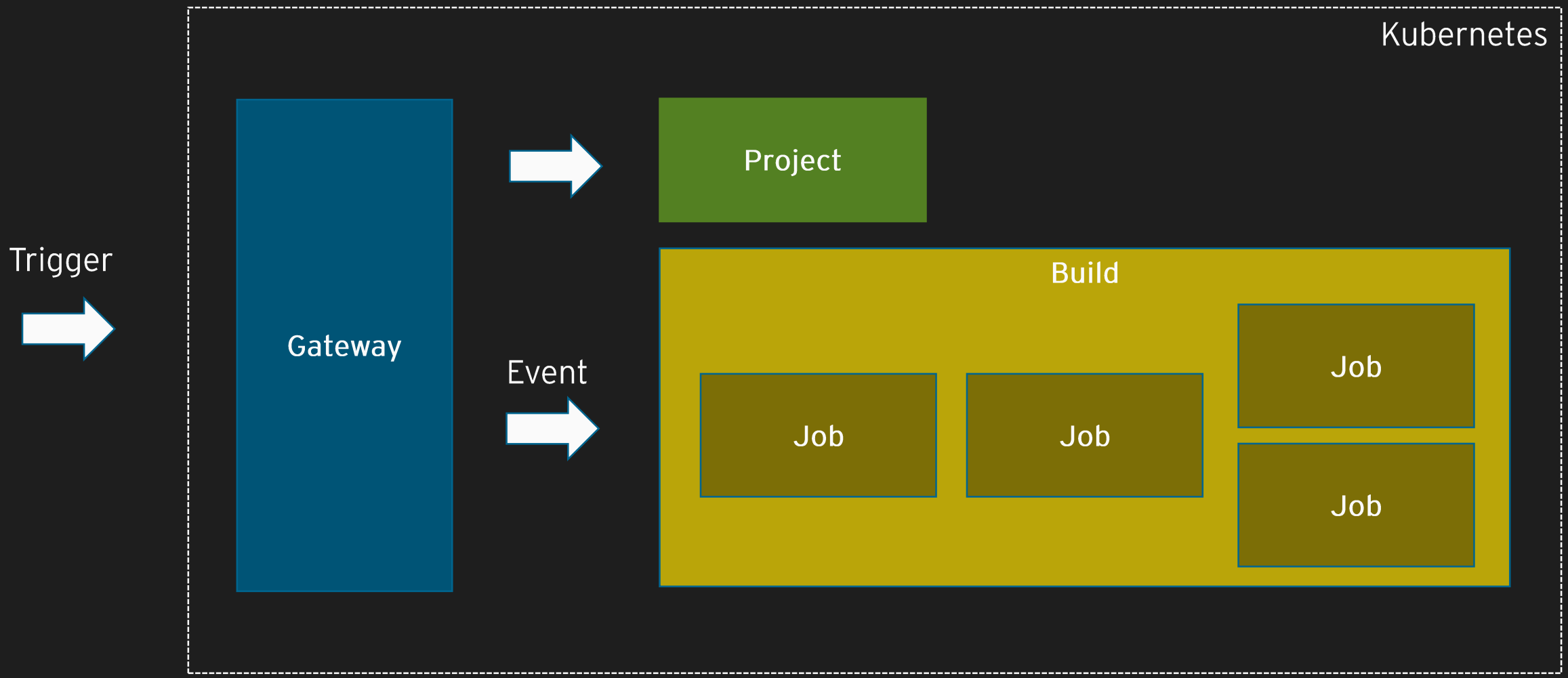
Supports CloudEvents

cloudevents

https://cloudevents.io/

A specification for describing event data in a common way

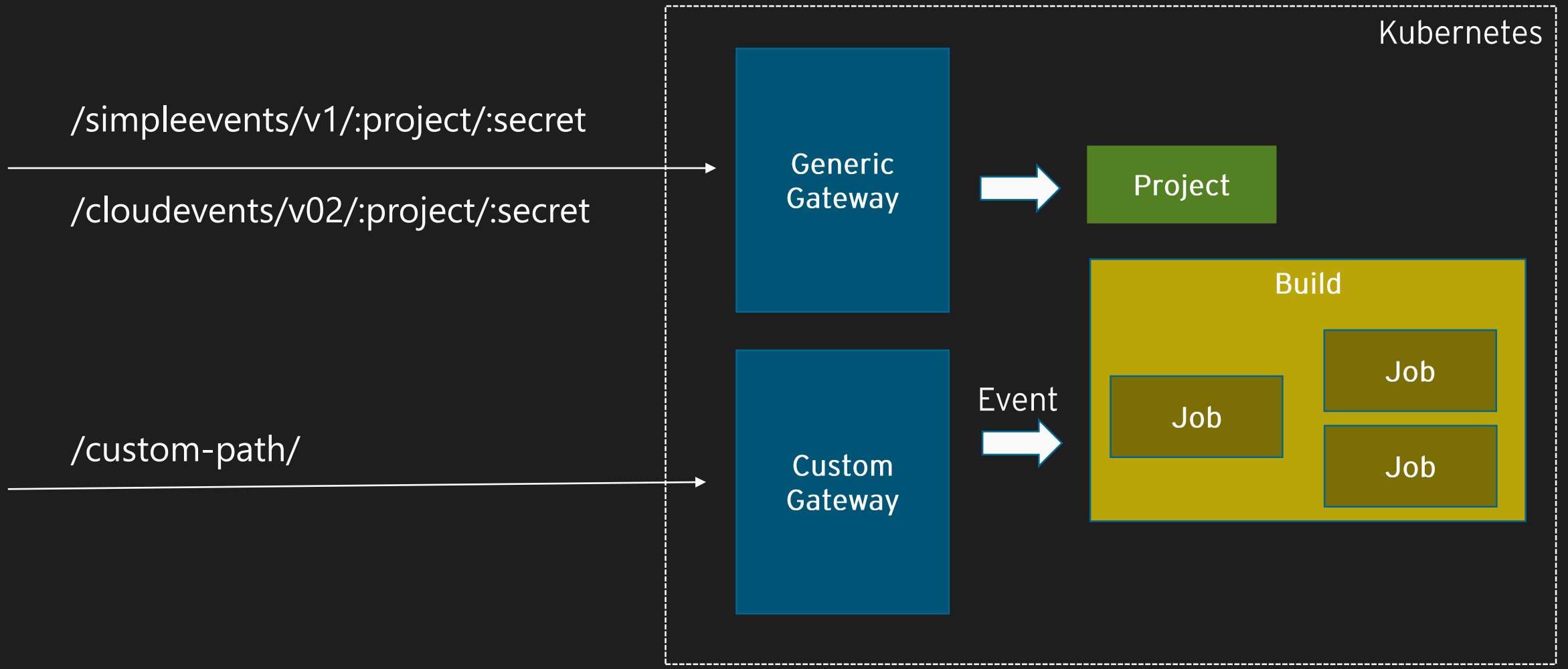
Brigade Architecture



Brigade script

```
const { events, Job } = require("brigadier");

//Handler for exec event
events.on("exec", () => {

  var job = new Job("say-hello", "alpine:3.8");
  job.tasks = [
    "echo Hello",
    "echo World"
  ];

  job.run();

});
```

Brigade Events



Demo: Brigade

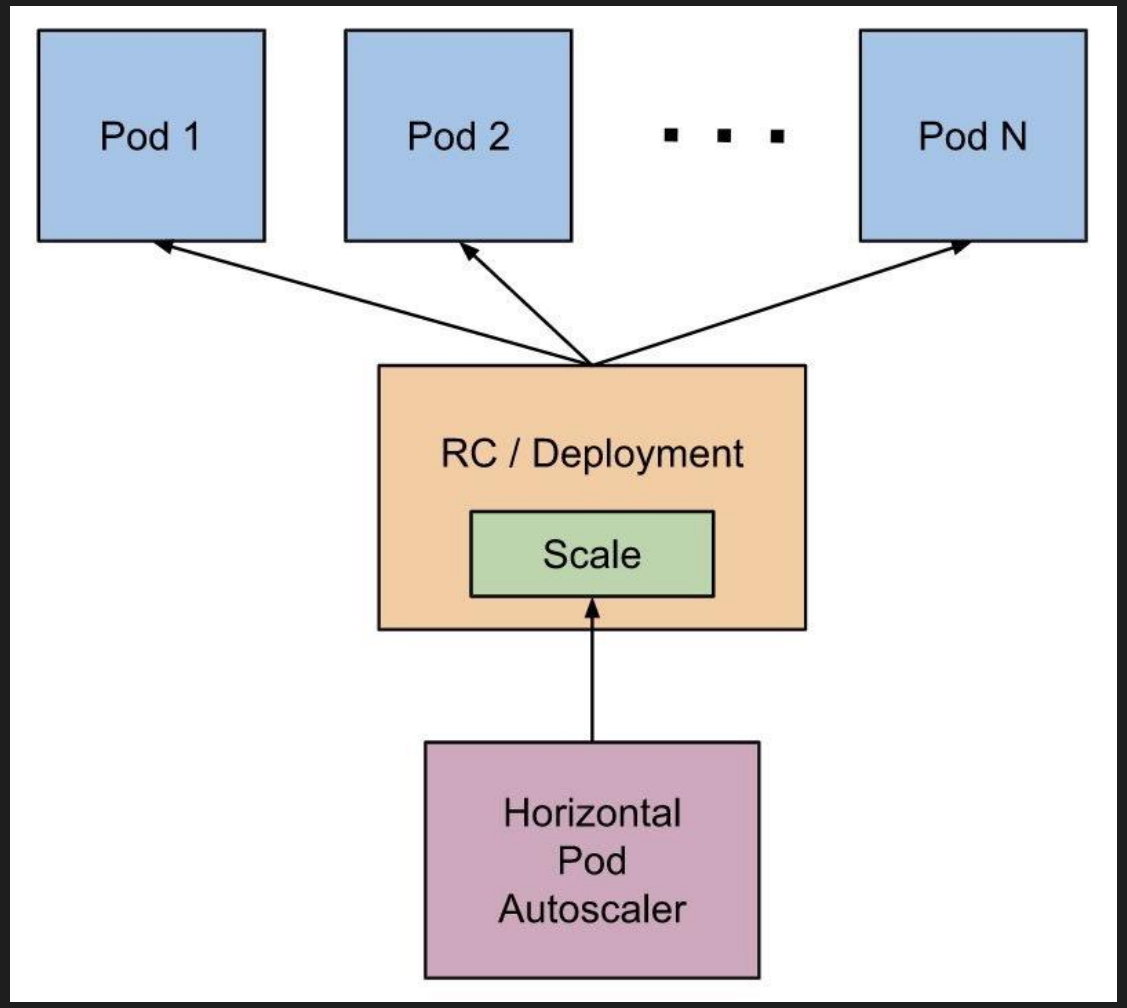
- Create and run Brigade pipelines
- Trigger from GitHub pushes
- Integrating with CloudEvents
- PR Workflow for helm deployments

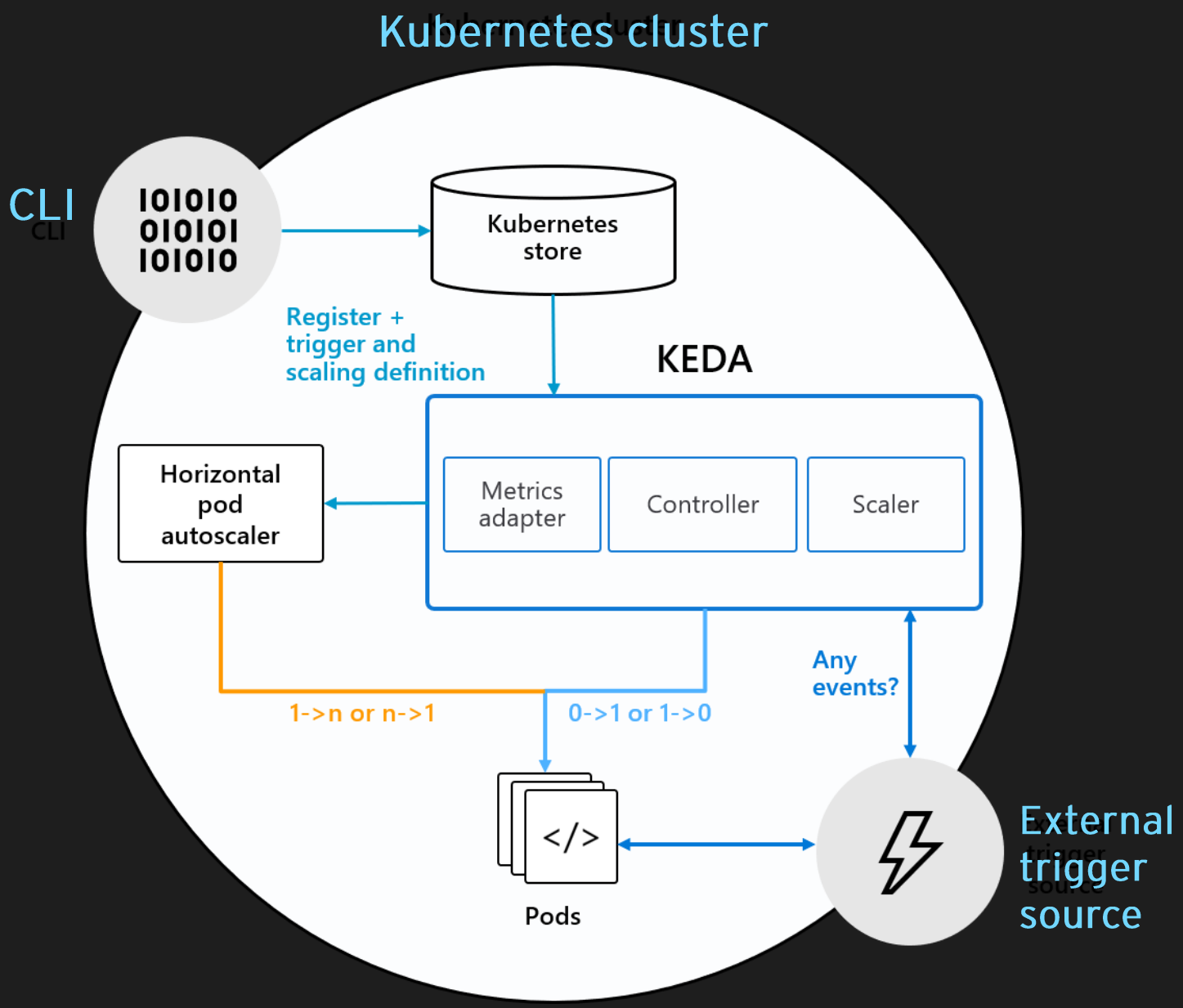
Kubernetes-based Event Driven Autoscaling

KEDA

- Open source component for event-driven scaling in Kubernetes
- Scale to zero or to thousands
- Support various events sources
- Run and scale Azure Functions in Kubernetes

Kubernetes Horizontal Pod Autoscaler (HPA)

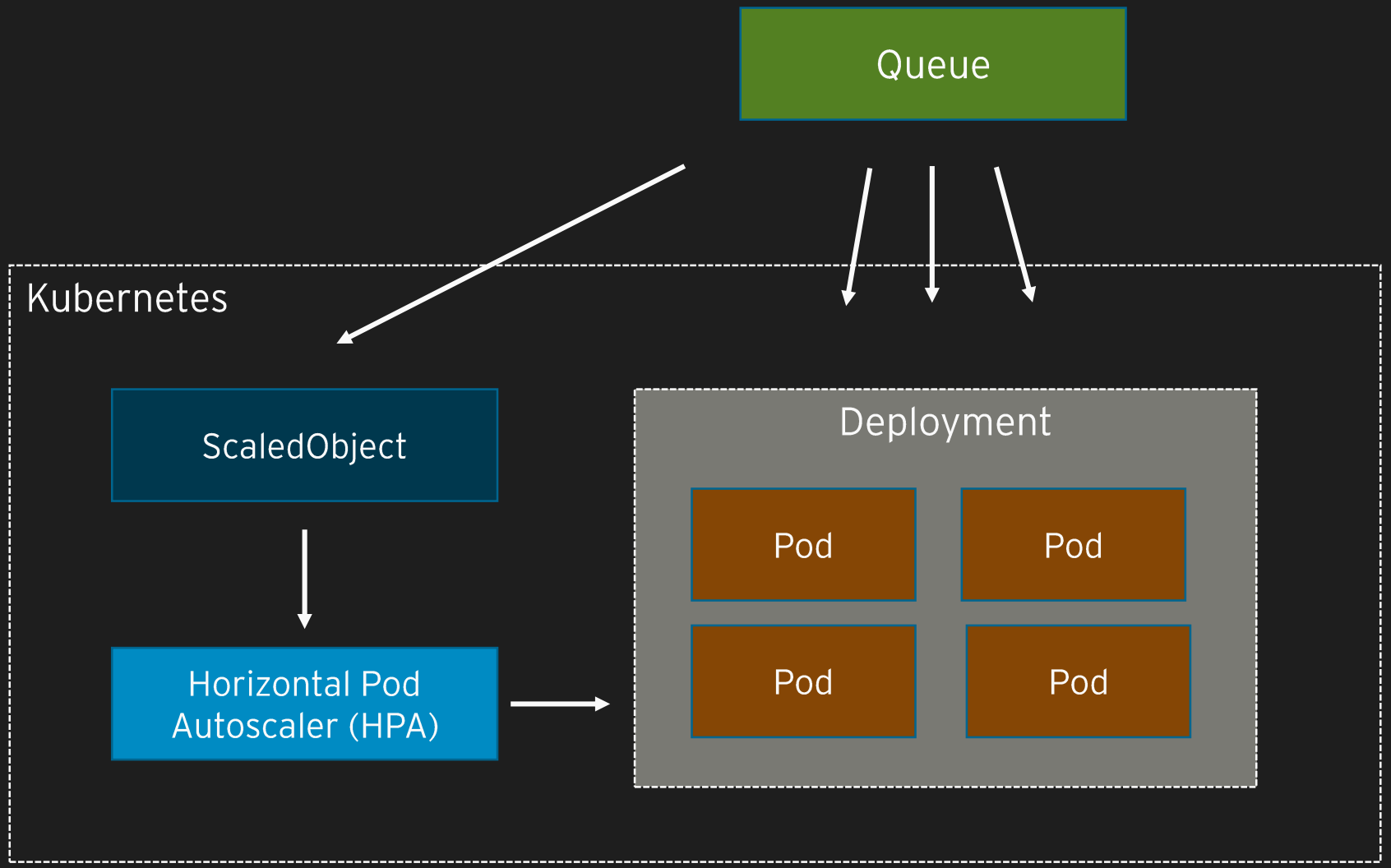




Event sources and scalars

- AWS CloudWatch
- AWS Simple Queue Service
- Azure Event Hub
- Azure Service Bus Queues and Topics
- Azure Storage Queues
- GCP PubSub
- Kafka
- Liiklus
- Nats Streaming
- Prometheus
- RabbitMQ
- Redis Lists
- ...

KEDA Scaling



ScaledObject CRD

```yaml
apiVersion: keda.k8s.io/v1alpha1
kind: ScaledObject
metadata:
  name: order-processor-scaler
  labels:
    app: order-processor
    deploymentName: order-processor
spec:
  scaleTargetRef:
    deploymentName: order-processor
  minReplicaCount: 1    #Change to define how many minimum replicas you want
  maxReplicaCount: 10
  pollingInterval: 5    # Optional. Default: 30 seconds
  cooldownPeriod:  30   # Optional. Default: 300 seconds
  triggers:
  - type: azure-servicebus
    metadata:
      queueName: orders
      connection: KEDA_SERVICEBUS_QUEUE_CONNECTIONSTRING
      queueLength: '5'
```

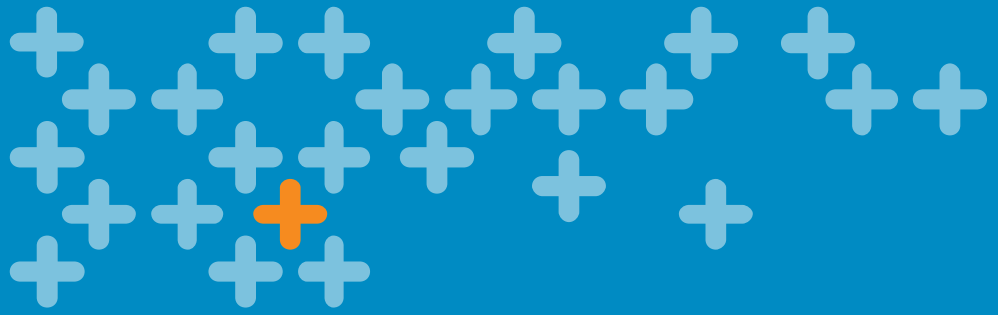
KEDA Demo

- Scaling deployment based on Azure Storage queue length
- Running & Scaling Azure Functions with Keda

Brigade
https://brigade.sh

KEDA
https://keda.sh

Demo code
https://github.com/jakobehn/brigade-keda



Thank you!

Jakob Ehn
@jakobehn
https://blog.ehn.nu