THE DEVELOPER'S CONFERENCE

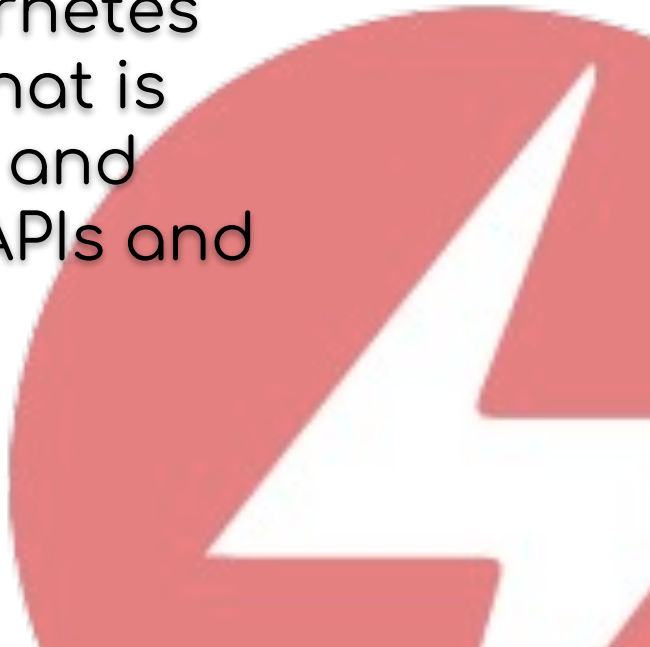
Desenvolvimento para Kubernetes

Alexandre Mioranza

TDC - 2019

Kubernetes is an open source orchestrator for deploying containerized applications. It was originally developed by Google, inspired by a decade of experience deploying scalable, reliable systems in containers via application-oriented APIs.

An **Operator** is a method of packaging, deploying and managing a Kubernetes application. A Kubernetes application is an application that is both deployed on Kubernetes and managed using the Kubernetes APIs and kubectl tooling.



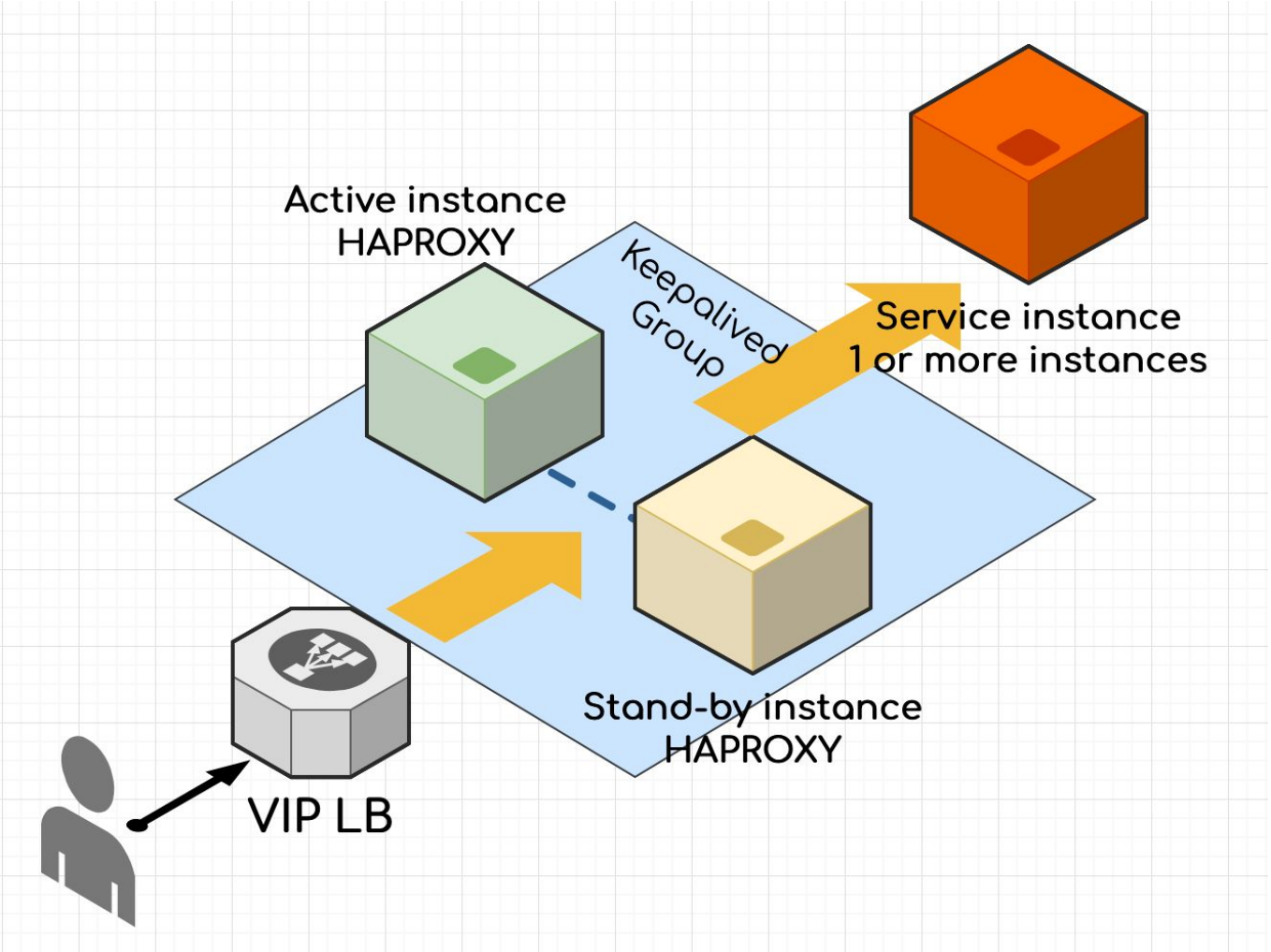
Alex Garzão muito obrigado!

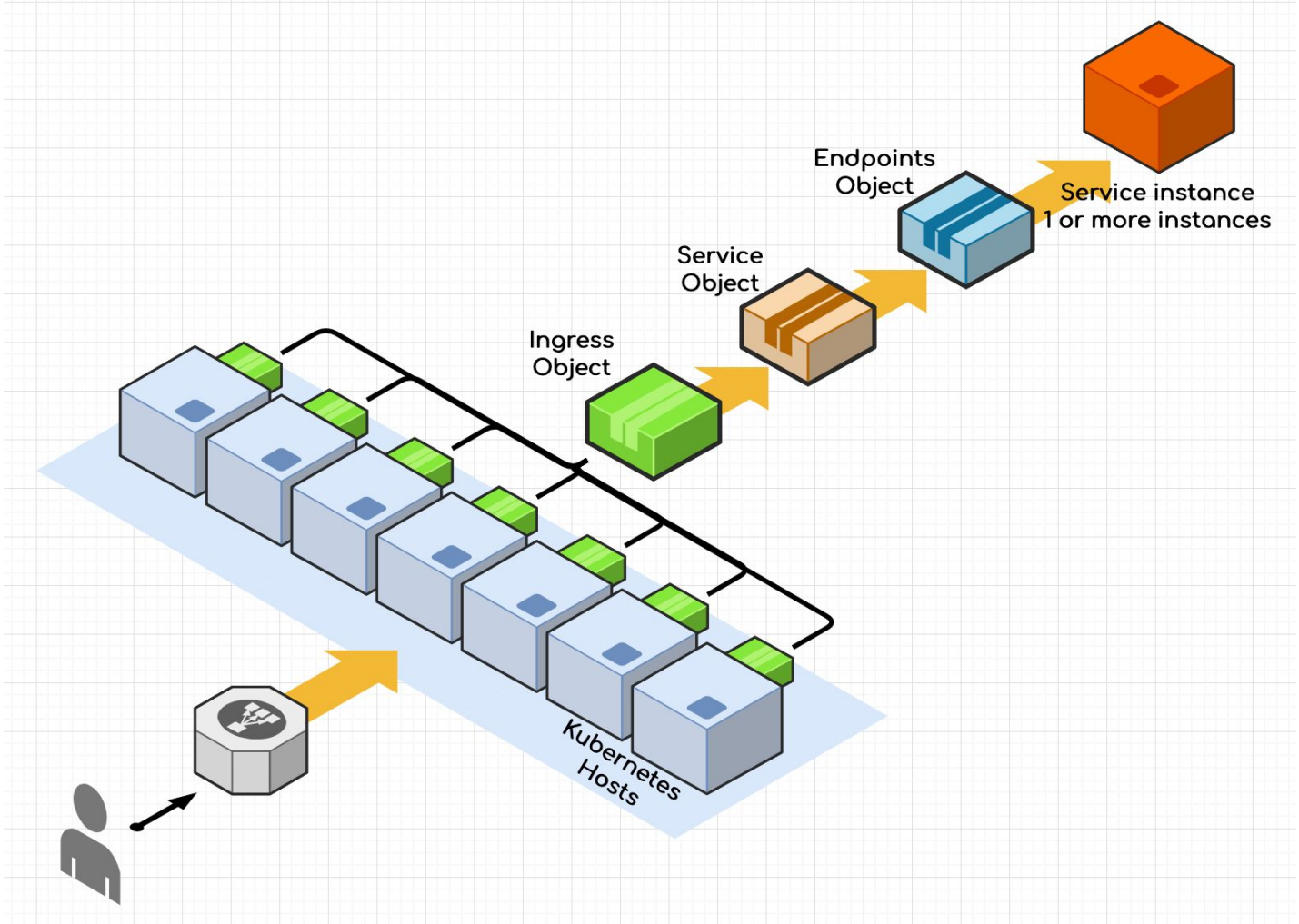
THE DEVELOPER'S CONFERENCE

THE DEVELOPER'S CONFERENCE

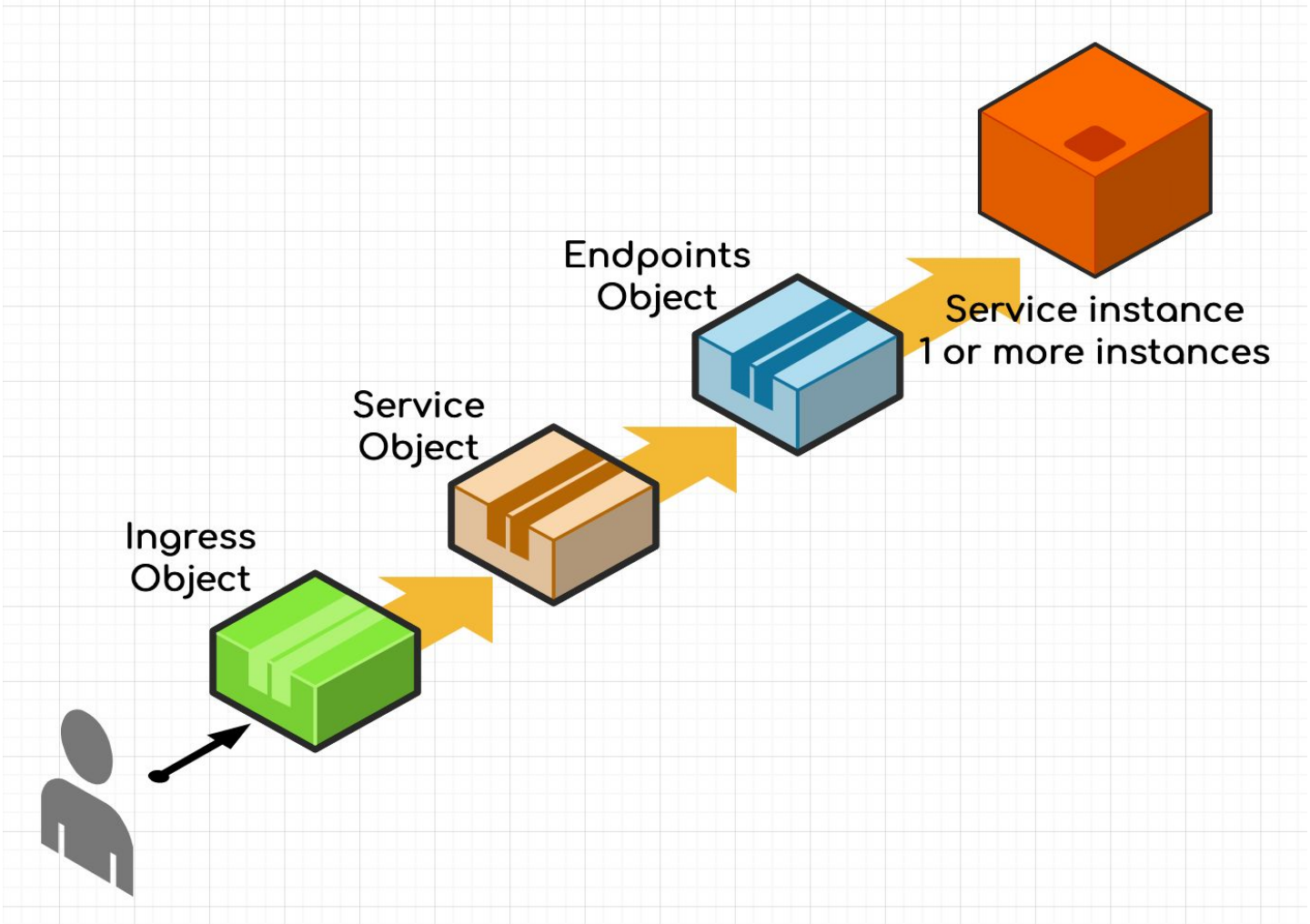
Desejo a ser atendido:

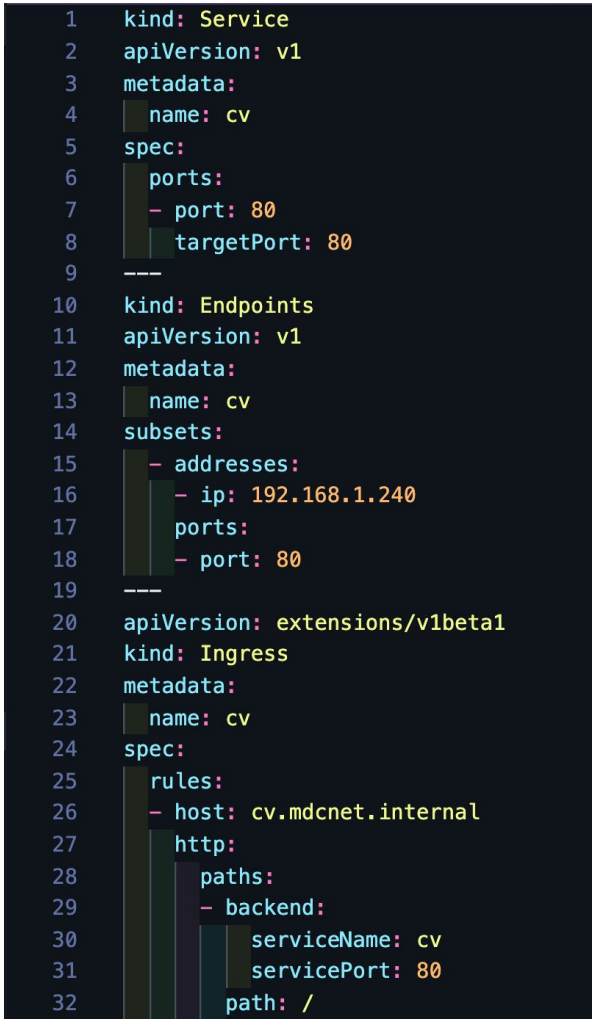
Remover instâncias de HAProxy do ambiente de Flow para aplicações que não estão no Kubernetes.


Active instance
HAPROXY
Keepalived
Group
Service instance
1 or more instances
Stand-by instance
HAPROXY
VIP LB



THE DEVELOPER'S CONFERENCE


Ingress Object
Service Object
Endpoints Object
Service instance
1 or more instances



```yaml
kind: Service
apiVersion: v1
metadata:
  name: cv
spec:
  ports:
  - port: 80
    targetPort: 80
---
kind: Endpoints
apiVersion: v1
metadata:
  name: cv
subsets:
  - addresses:
    - ip: 192.168.1.240
    ports:
    - port: 80
---
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: cv
spec:
  rules:
  - host: cv.mdcnet.internal
    http:
      paths:
      - backend:
          serviceName: cv
          servicePort: 80
        path: /
```

THE DEVELOPER'S CONFERENCE

Desejo a ser atendido:

Remover instâncias de HAProxy do ambiente de Flow para aplicações que não estão no Kubernetes.

ATENDIDO!

Novos problemas introduzidos:

1 - Cada serviço necessita de 3 objetos para serem configurados: ingress, service e endpoints

2 - Endpoints são configurações fixas e não removem endpoints com falha

THE DEVELOPER'S CONFERENCE

Desejos novos a serem atendidos:

1 - Configurar um único objeto no Kubernetes

2 - Atualizar os endpoints de forma dinâmica de acordo com a saúde do serviço

Controllers	Operators
Atua sobre os recursos core tais como deployments, services, endpoints. São tipicamente parte do Kubernetes controller-manager no control plane ou podem observar e manipular recursos customizados criados pelo usuário	São controllers que codificam algum conhecimento operacional, por exemplo, o ciclo de vida de uma aplicação em conjunto com recursos definidos pelo usuário

Endpoints Operator

1 - Configurar um único objeto no Kubernetes

```yaml
kind: Service
apiVersion: v1
metadata:
  name: cv
spec:
  ports:
  - port: 80
    targetPort: 80
---
kind: Endpoints
apiVersion: v1
metadata:
  name: cv
subsets:
  - addresses:
    - ip: 192.168.1.240
    ports:
    - port: 80
---
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: cv
spec:
  rules:
  - host: cv.mdcnet.internal
    http:
      paths:
      - backend:
          serviceName: cv
          servicePort: 80
        path: /
```

Custom Resource Definition

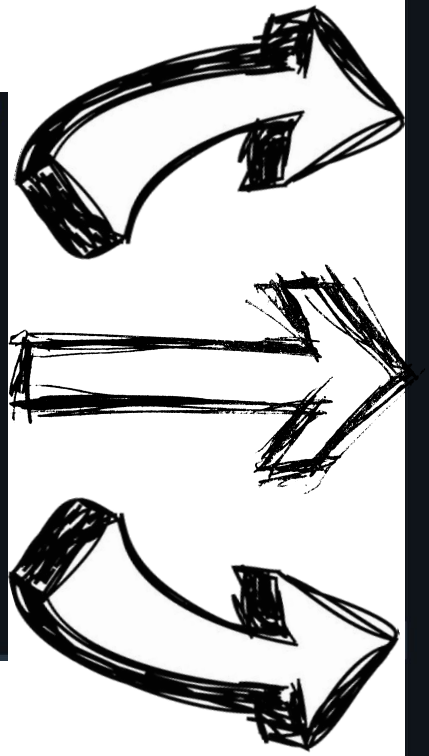
Custom Resource

ETCD

```yaml
kind: Service
apiVersion: v1
metadata:
  name: cv
spec:
  ports:
  - port: 80
    targetPort: 80
---
kind: Endpoints
apiVersion: v1
metadata:
  name: cv
subsets:
  - addresses:
    - ip: 192.168.1.240
    ports:
    - port: 80
---
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: cv
spec:
  rules:
  - host: cv.mdcnet.internal
    http:
      paths:
      - backend:
          serviceName: cv
          servicePort: 80
        path: /
```

```yaml
apiVersion: endpoints.mdcnet.ninja/v1alpha1
kind: EndpointService
metadata:
  name: cv
spec:
  url: cv.mdcnet.internal
  targetPort: 8000
  port: 80
  path: "/"
  scheme: "http"
  endpoints:
    - "192.168.1.240"
  healthcheck:
    protocol: "HTTP"
    path: "/"
```

```yaml
apiVersion: endpoints.mdcnet.ninja/v1alpha1
kind: EndpointService
metadata:
  name: cv
spec:
  url: cv.mdcnet.internal
  targetPort: 8000
  port: 80
  path: "/"
  scheme: "http"
  endpoints:
    - "192.168.1.240"
  healthcheck:
    protocol: "HTTP"
    path: "/"
```



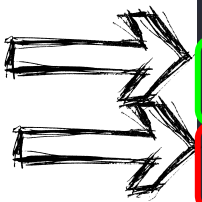
```yaml
kind: Service
apiVersion: v1
metadata:
  name: cv
spec:
  ports:
  - port: 80
    targetPort: 80
---
kind: Endpoints
apiVersion: v1
metadata:
  name: cv
subsets:
  - addresses:
    - ip: 192.168.1.240
    ports:
    - port: 80
---
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: cv
spec:
  rules:
  - host: cv.mdcnet.internal
    http:
      paths:
      - backend:
          serviceName: cv
          servicePort: 80
        path: /
```

TALK IS CHEAP

SHOW ME THE CODE

Linus Torvalds

2 - Atualizar os endpoints de forma dinâmica de acordo com a saúde do serviço



```
apiVersion: v1
kind: Endpoints
metadata:
  creationTimestamp: "2019-11-18T23:53:13Z"
  labels:
    endpointservice: cv
  name: cv
  namespace: default
  ownerReferences:
  - apiVersion: endpoints.mdcnet.ninja/v1alpha1
    blockOwnerDeletion: true
    controller: true
    kind: EndpointService
    name: cv
    uid: ba2817d1-621a-489c-9b0f-13477d5ad14f
  resourceVersion: "289025"
  selfLink: /api/v1/namespaces/default/endpoints/cv
  uid: 70442d00-b083-4733-8c76-b88d478819f0
subsets:
- addresses:
  - ip: 192.168.1.188
  notReadyAddresses:
  - ip: 192.168.50.82
  ports:
  - port: 8000
    protocol: TCP
```

THE DEVELOPER'S CONFERENCE

TALK IS CHEAP

SHOW ME THE CODE

Linus Torvalds

THE DEVELOPER'S CONFERENCE

Desejos novos a serem atendidos:

1 - Configurar um único objeto no Kubernetes

2 - Atualizar os endpoints de forma dinâmica de acordo com a saúde do serviço

ATENDIDOS!

THE DEVELOPER'S CONFERENCE

Melhorias e funcionalidades próximas versões:

- Suporte TLS (ingress e endpoints);
- Paralelização dos testes dos endpoints;
- Múltiplos subsets para definir mais de uma porta no service;
- Gerar dinamicamente o ingress e o service em caso de update do EndpointService.


SCAN ME
SCAN ME
SCAN ME
SCAN ME
THE
DEVELOPER'S
CONFERENCE