TECH CONFERENCE
DotNet 2020

#DotNet2020

Desplegando a AKS tus aplicaciones .NET Core con Azure DevOps YAML

ORGANIZATION

plain concepts

PLATINUM SPONSORS

Microsoft

COLLABORATORS

CEU | Universidad San Pablo

Thank you!

Juan Ramírez

Software Engineer

@jramgar

jramgar@plainconcepts.com

Luis Fraile

Help development teams

@lfraile

lfraile@plainconcepts.com

DISCLAIMER !!!

Lo que vamos a ver

Introducción a despliegues YAML

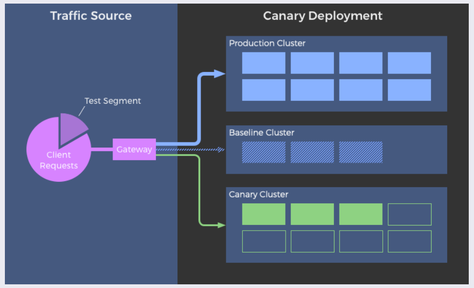
Estrategias de ~~despliegue~~ entrega

Canary deployments

Cambios en paralelo (*parallel changes*)

¿Qué vamos a usar?

- AKS + Service Mesh
- YAML con estrategia canary
- Kubernetes Manifest Task



Azure DevOps Pipelines

Flujos de ejecución para compilación y despliegue (CI/CD)

Nos permite la automatización de todos nuestros procesos

YAML versionados en repositorio

Multiplataforma:

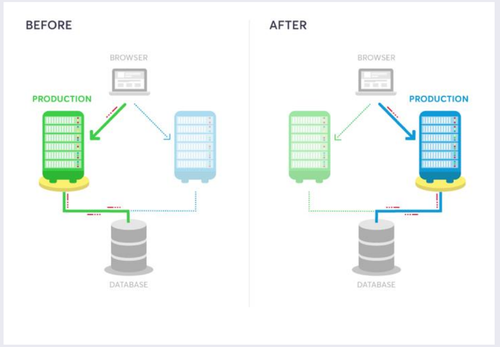
- Cualquier tipo de proyecto (.NET , Java, NodeJS, Go, Xcode, …)
- Cualquier tipo de entorno (Azure, Aws, Linux, Mac, …)

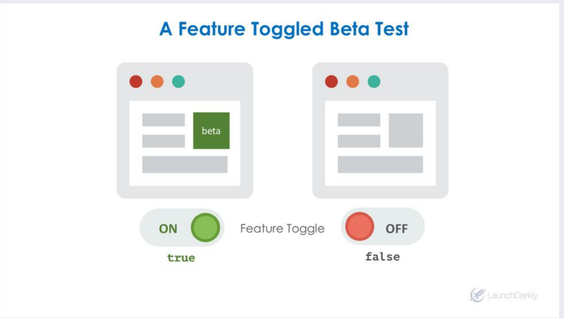
Estrategias de ~~despliegue~~ entrega

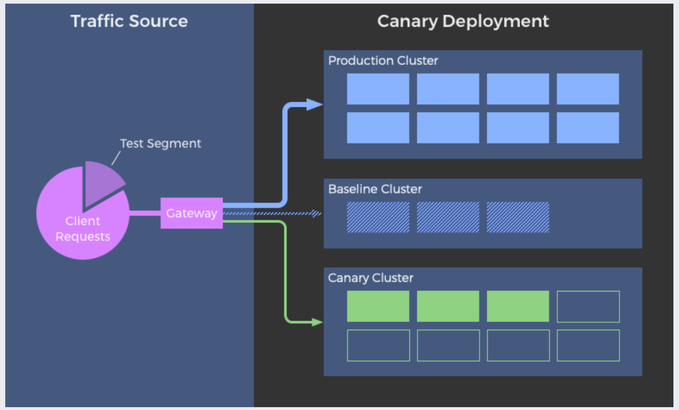
Unas cuantas

- Blue/Green
- Toggles
- Canary deployment
- Por entornos …

O todas a la vez …







Canary deployment

1. Se despliega lo nuevo a “canarios”
    - Desplegamos a nodos
    - Redirigimos parte del tráfico a esos nodos
2. Se mantiene en pruebas comparando métricas con línea base
3. Se incrementa el tráfico de usuarios a lo nuevo (nuevos nodos)
4. Se sigue midiendo y comparando
5. Repetir del 2 al 4 hasta llegar al 100% ☺

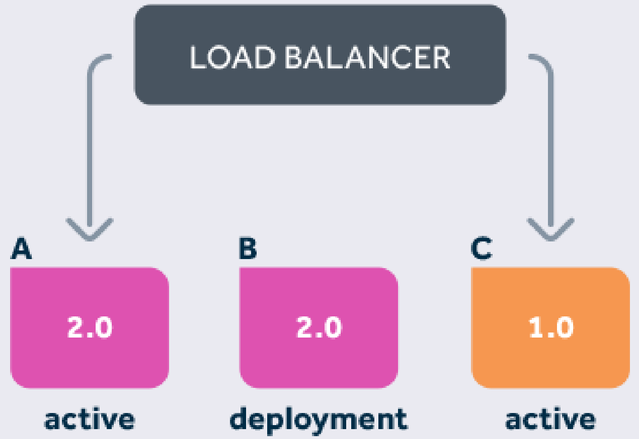
OJO: Convive durante un tiempo vieja y nueva versión

Compatibilidad de versiones

Tenemos que prepararnos para convivir con dos versiones

Parallel Changes (Danilo Sato - https://martinfowler.com/bliki/ParallelChange.html)

- Expand
- Migrate
- Contract



Tanto para almacén de datos

Como para interfaces

https://www.slideshare.net/sergio_pino/commit-conf-2019-zero-downtime-deployment

Kubernetes - AKS

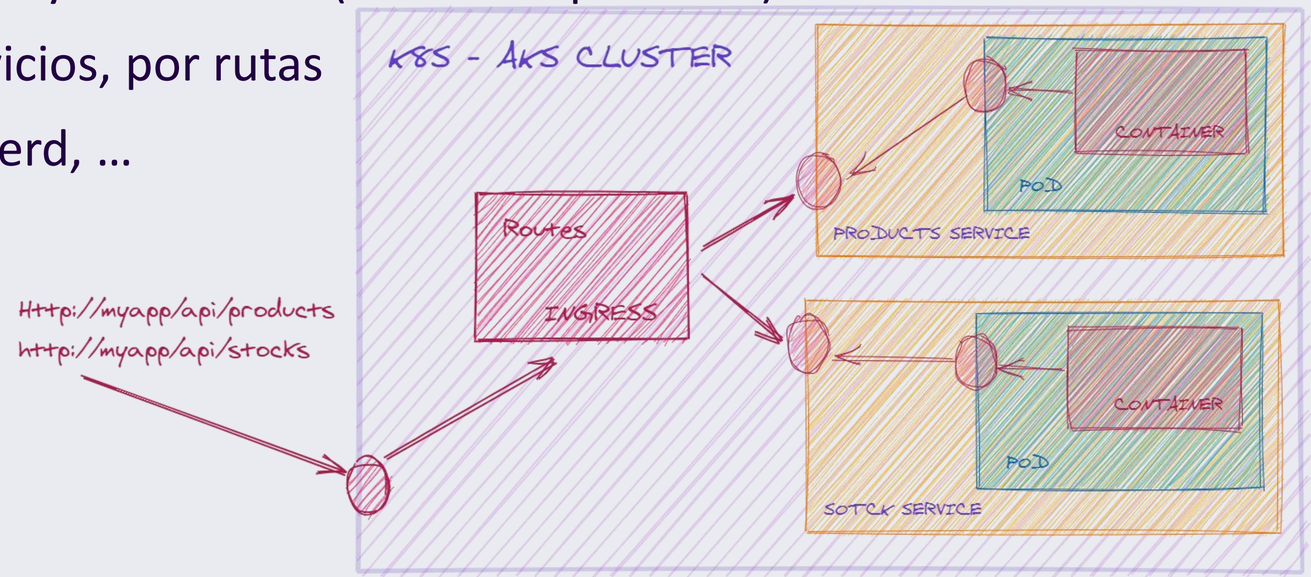
Entorno ideal microservicios o servicios distribuidos

Desplegamos nuestra aplicaciones con contenedores en Pods

Definimos servicios para exponer nuestras aplicaciones

Exponemos servicios usando un recurso Ingress,

- Abstracción en Kubernetes de Proxy nivel OSI 7 (HTTP → Aplicación)
- Redireccionar el tráfico a los servicios, por rutas
- Diferentes: Nginx, AKS, Istio, Linkerd, …



Service Mesh

¿Qué es?

Un montón de proxies que conectan nuestros servicios y que nos permite hacer cosas con ellos

¿Qué no es?

No se usa para hacer trasnformaciones de la semática o el payload de las peticiones

Esto lo hacemos con un Enterprise Service Bus o un middleware.

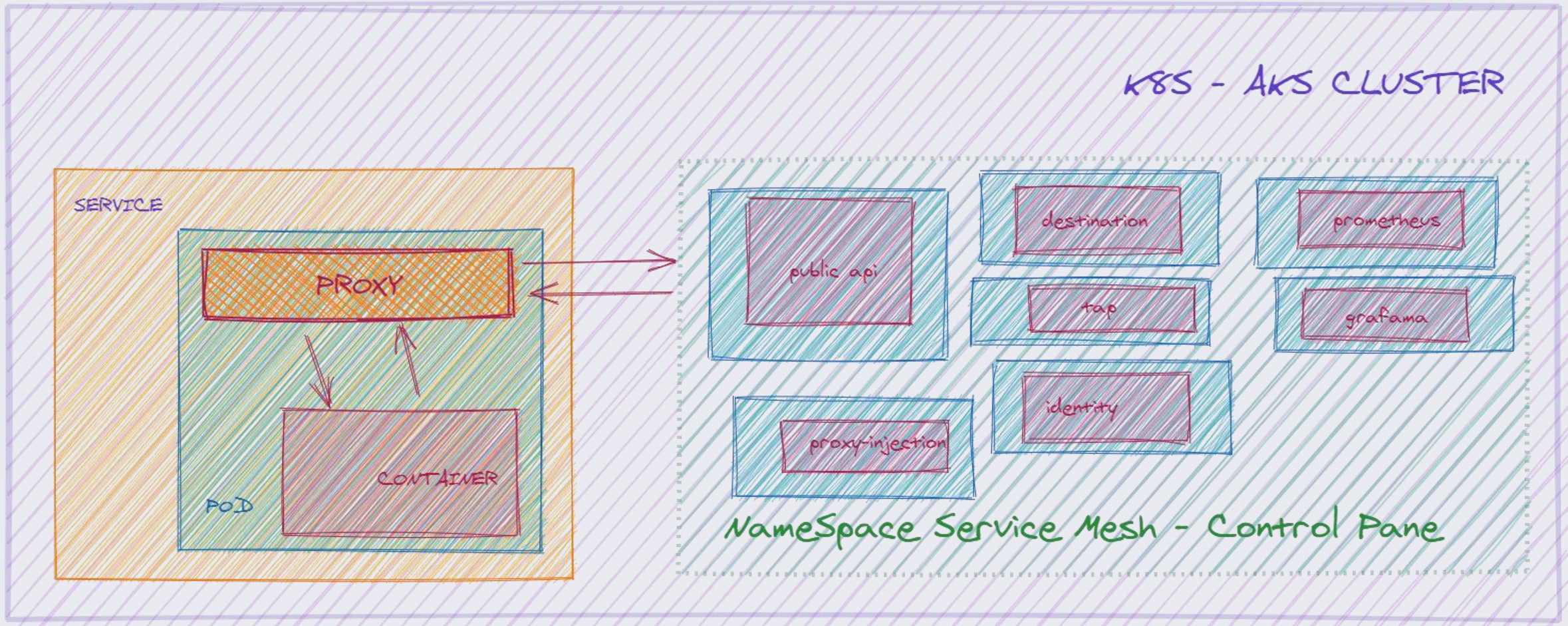
En nuestro caso usaremos **Linkerd**, muy sencillo y rápido de instalar

https://buoyant.io/service-mesh-manifesto/

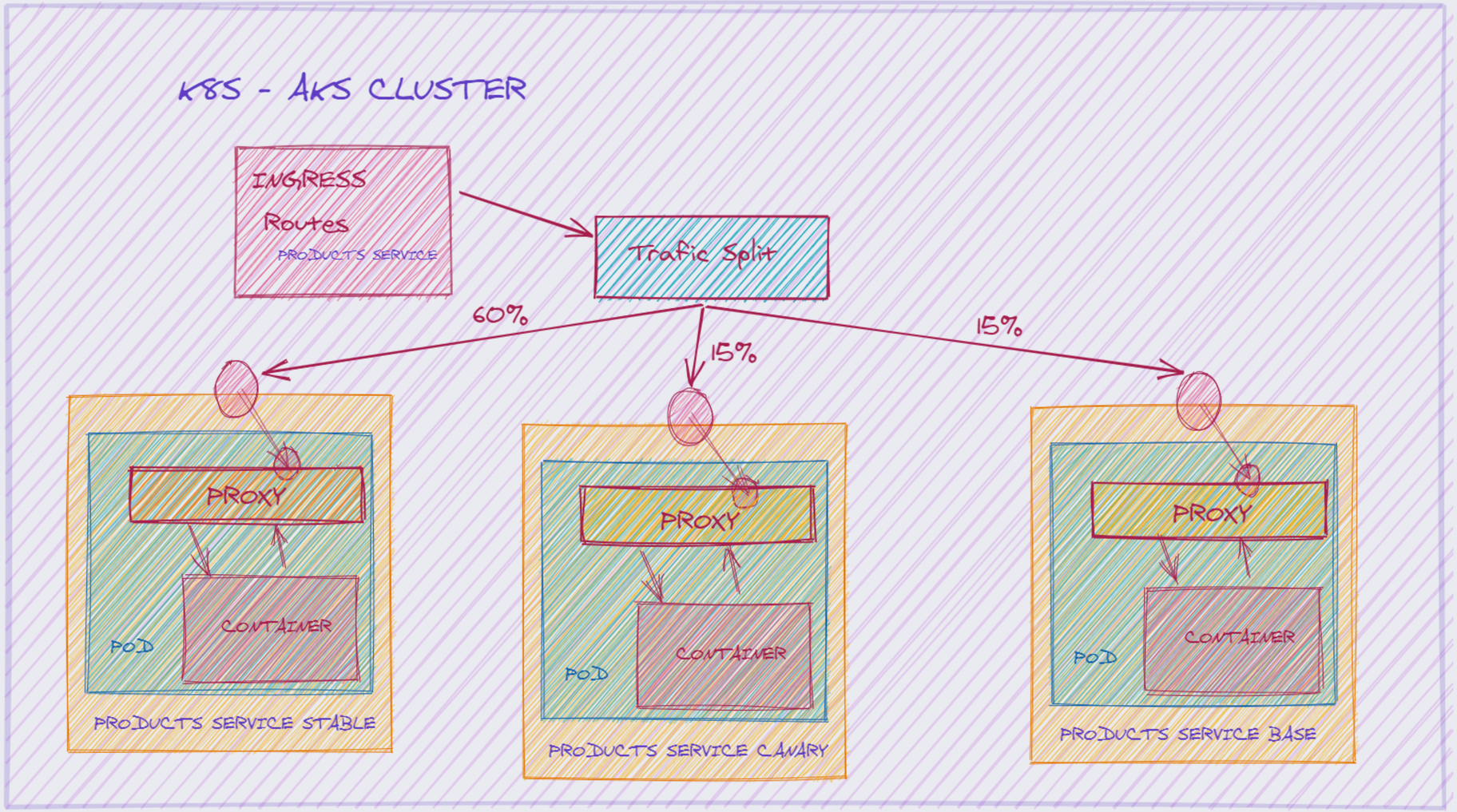
¿Por qué un service Mesh?

- Funciones de confiabilidad.
  - Solicitar reintentos,
  - Tiempos de espera
  - **Canaries (división / cambio de tráfico), etc.**
- Características de observabilidad.
  - Agregación de tasas de éxito, latencias y volúmenes de solicitudes para cada servicio
  - Rutas individuales
  - Dibujo de mapas de topología de servicios; etc.
- Características de seguridad.
  - TLS mutuo,
  - Control de acceso,
  - etc.

Qué se crea con Linkerd



¿Y en un canary depoyment?



Instalar Linkerd

az account set --subscription 00000000-0000-0000-0000-000000000000

az aks get-credentials --resource-group dotnet2020-aks-rg --name dotnet2020-aks

linkerd install > linkerd.yaml

kubectl apply -f linkerd.yaml

linkerd check

linkerd dashboard &

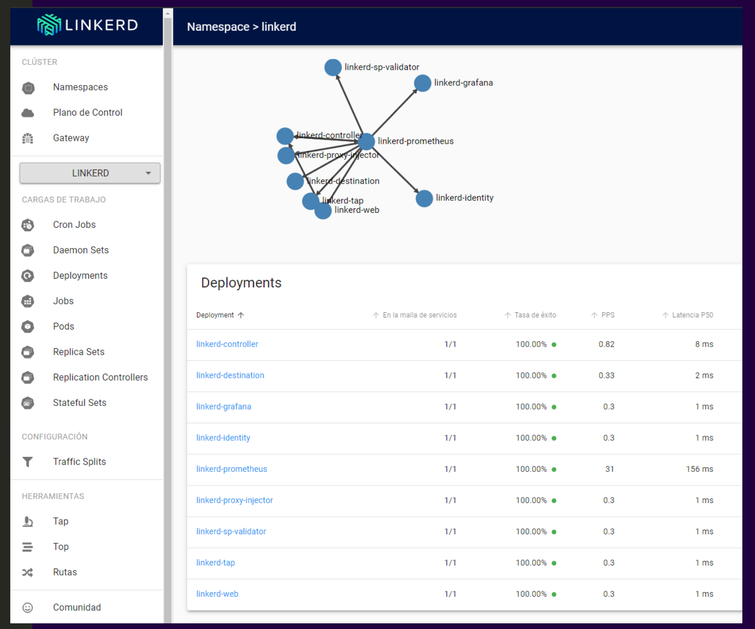
```
λ linkerd check
kubernetes-api
--------------
√ can initialize the client
√ can query the Kubernetes API

kubernetes-version
------------------
√ is running the minimum Kubernetes API version
√ is running the minimum kubectl version

linkerd-existence
-----------------
√ 'linkerd-config' config map exists
√ heartbeat ServiceAccount exist
√ control plane replica sets are ready
√ no unschedulable pods
√ controller pod is running
√ can initialize the client
√ can query the control plane API

linkerd-config
--------------
√ control plane Namespace exists
√ control plane ClusterRoles exist
√ control plane ClusterRoleBindings exist
√ control plane ServiceAccounts exist
√ control plane CustomResourceDefinitions exist
√ control plane MutatingWebhookConfigurations exist
√ control plane ValidatingWebhookConfigurations exist
√ control plane PodSecurityPolicies exist

linkerd-identity
----------------
√ certificate config is valid
√ trust anchors are using supported crypto algorithm
√ trust anchors are within their validity period
√ trust anchors are valid for at least 60 days
√ issuer cert is using supported crypto algorithm
√ issuer cert is within its validity period
√ issuer cert is valid for at least 60 days
√ issuer cert is issued by the trust anchor
```



Injectar Linkerd

Servicio a servicio:

kubectl -n dotnet2020 get deploy -o yaml | linkerd inject - | kubectl apply -f -

A todos los servicios del namespace

kubectl annotate namespace dotnet2020 linkerd.io/inject=enabled --overwrite

SMI – Service Mesh Interface

¿Cómo conectamos todo esto con nuestros Pipelines?

Es una interfaz estándar para servicios Mesh en Kubernetes

Azure DevOps se integra con K8S a través de SMI

¿Que cubre?:

- Políticas de tráfico
- Telemetría del tráfico
- Gestión del tráfico

Azure DevOps Kubernets Manifest Task, se integra con SMI

https://smi-spec.io/

Estrategia canary en YAML

Sólo disponible para despliegues YAML

Necesita de la tarea específica de Kubernetes Manifest Task

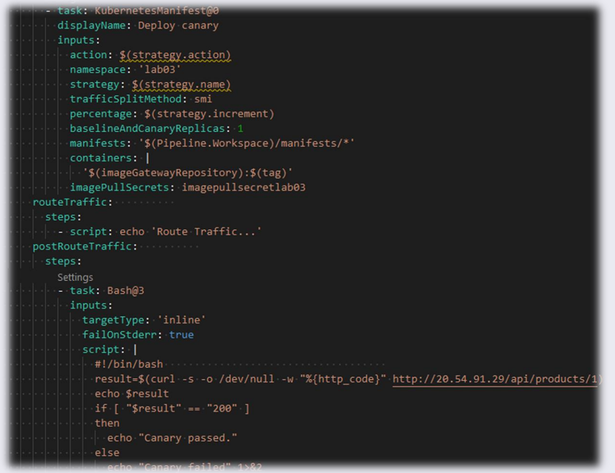
Definimos los incrementos de canarios

El llama a los distintos pasos de despliegue

Nos permite puntos de introducción de “sondas” para verificar métricas

Genera “variables”:

- Estrategia
- Acción
- Incremento



Tarea Kubernetes Manifest

```
- task: KubernetesManifest@0
  displayName: Deploy canary
  inputs:
    action: $(strategy.action)
    namespace: 'lab03'
    strategy: canary
    trafficSplitMethod: smi
    percentage: $(strategy.increment)
    baselineAndCanaryReplicas: 1
    manifests: '$(Pipeline.Workspace)/manifests/*'
    containers: |
      '$(imageGatewayRepository):$(tag)'
    imagePullSecrets: imagepullsecretlab03
```

https://docs.microsoft.com/en-us/azure/devops/pipelines/tasks/deploy/kubernetes-manifest?view=azure-devops

Talk is cheap!
Show me
the ~~code~~
YAML

Thanks and …
See you soon!

Thanks also to the sponsors.
Without whom this would not have been posible.

plain concepts

Microsoft

axazure

plasticscm

CEU | Universidad San Pablo