Microsoft

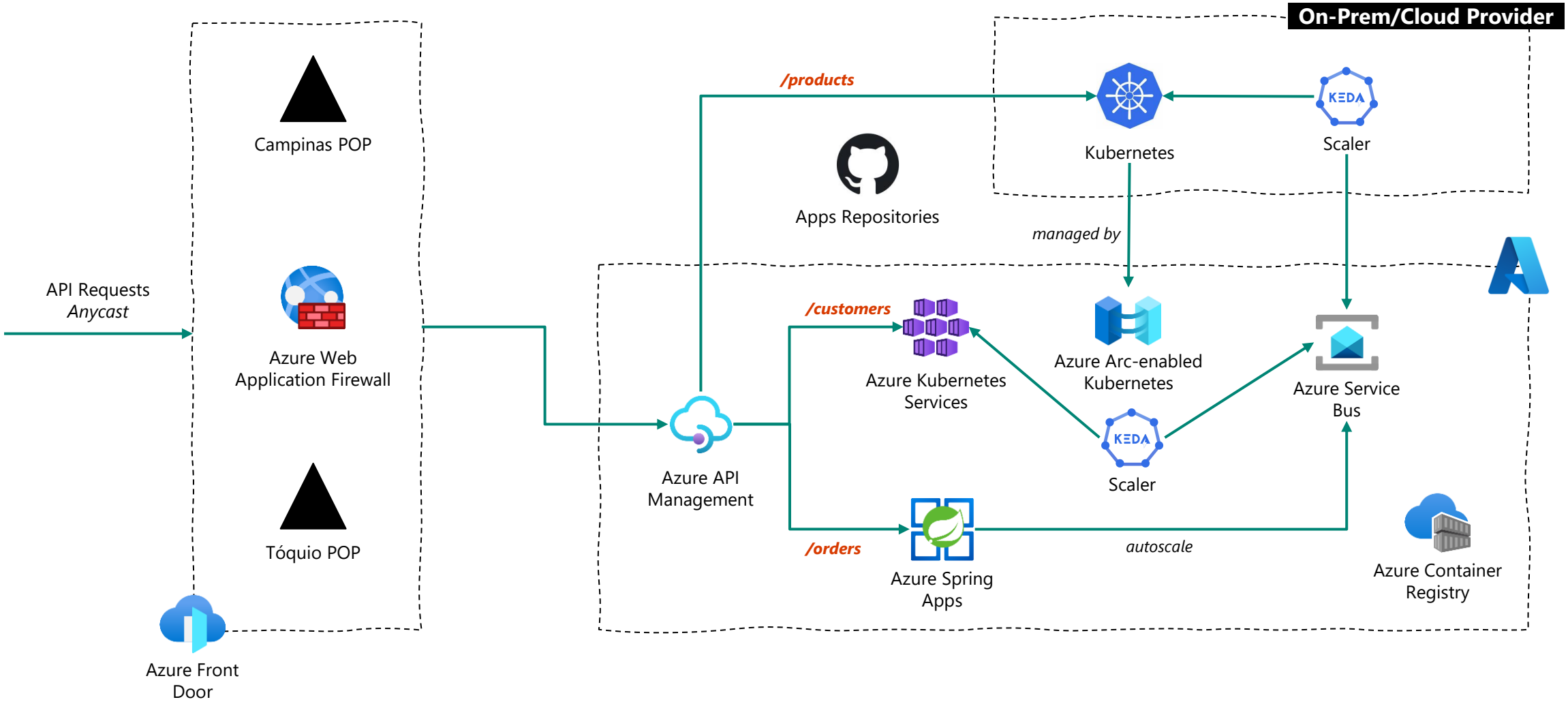
Aplicações nativas com AKS e Azure Arc-enabled Kubernetes

Quem sou eu?

George Luiz Bittencourt

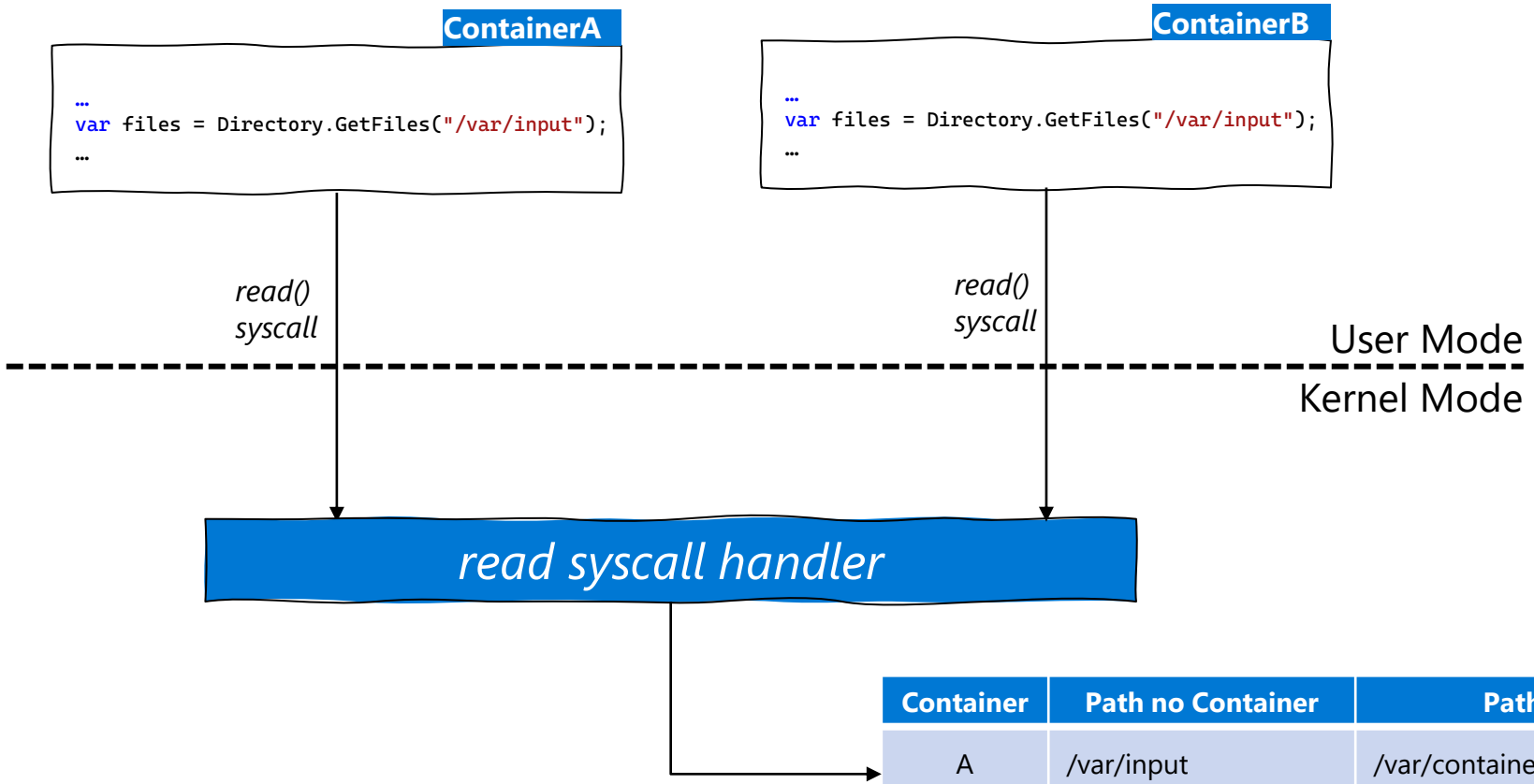
- CSA na Microsoft focado em Apps & Infra.
- Mais de 20 anos de experiência com desenvolvimento e infraestrutura.
- LinkedIn: https://www.linkedin.com/in/glzbcrt/
- E-mail: george.bittencourt@microsoft.com

Architecture



Containers = namespaces + cgroups + overlay filesystems

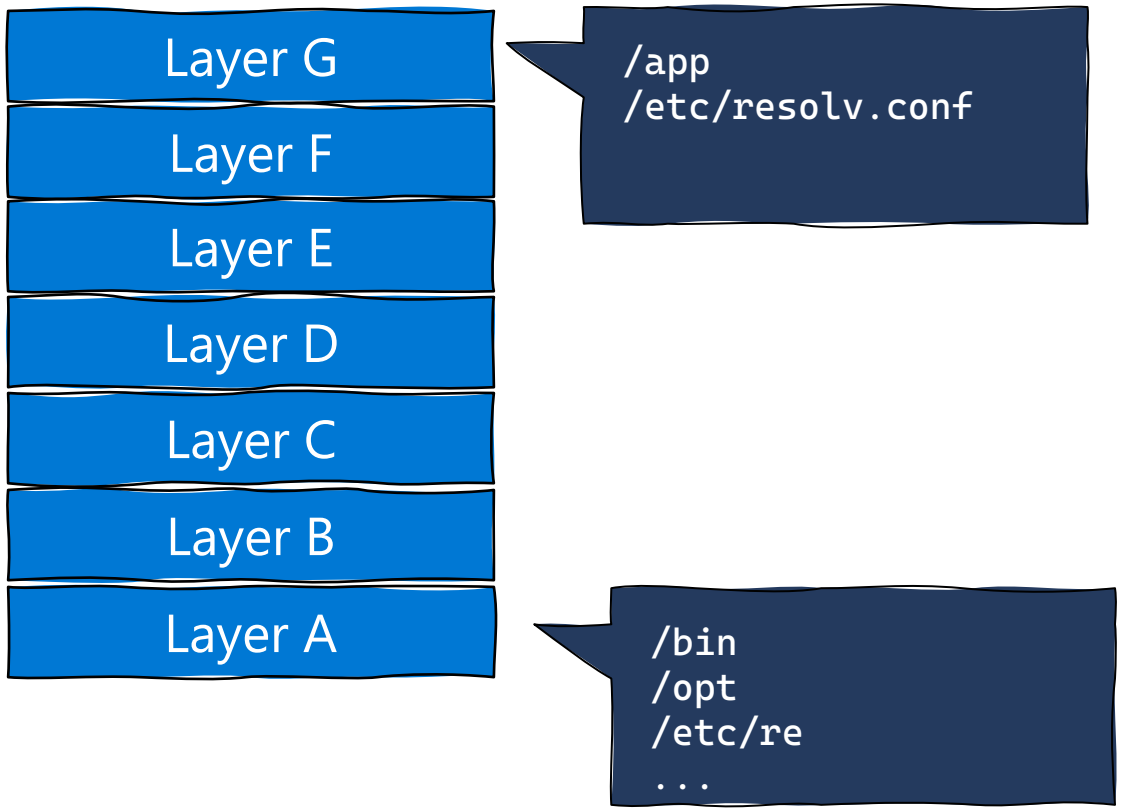
Linux Namespaces



Container	Path no Container	Path no Host
A	/var/input	/var/containerA/input
B	/var/input	/var/containerB/input

Linux Union/Overlay Filesystems

[opencontainers/image-spec: OCI Image Format]



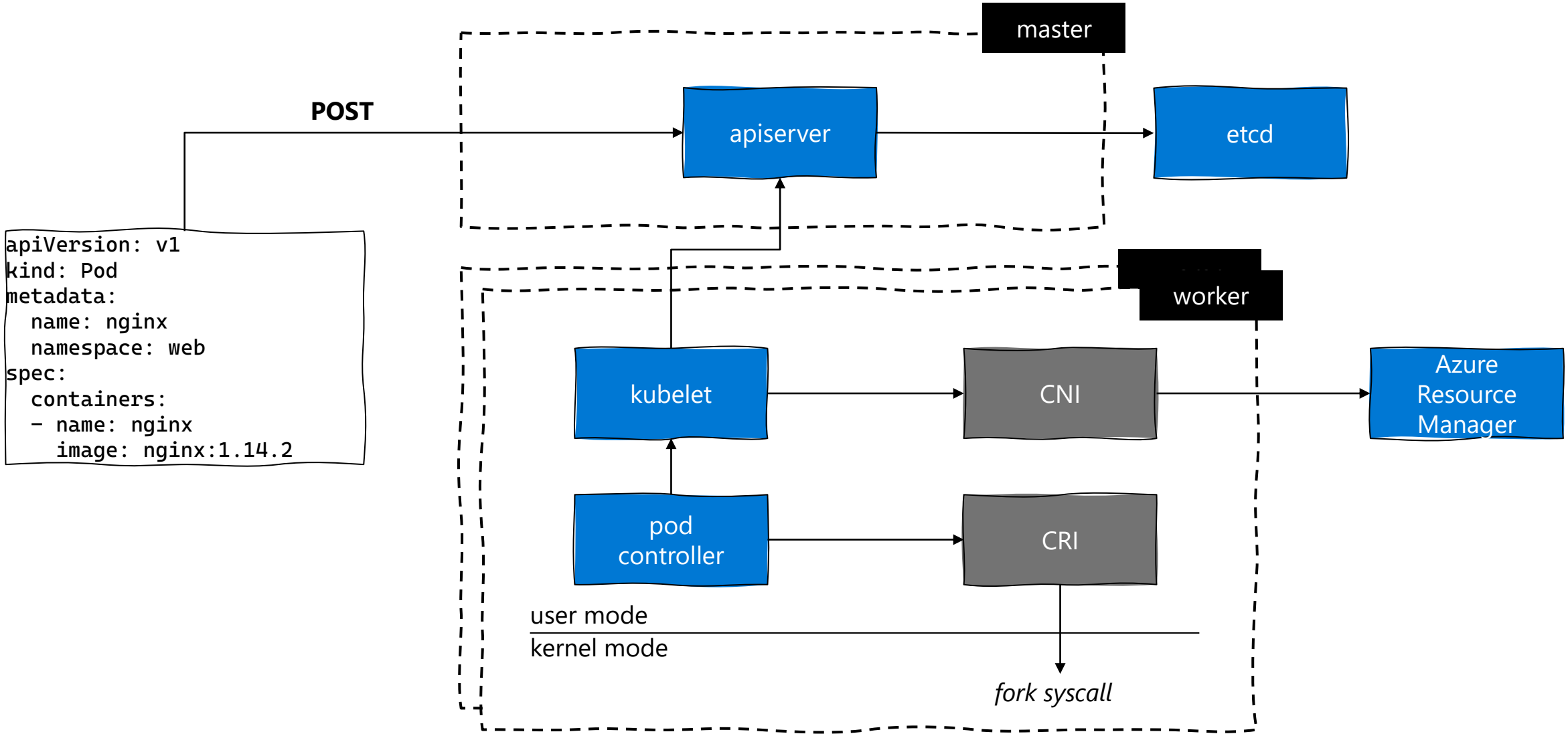

```
FROM mcr.microsoft.com/dotnet/sdk:7.0 AS build
WORKDIR /source

# copy csproj and restore as distinct layers
COPY *.csproj .
RUN dotnet restore --use-current-runtime

# copy and publish app and libraries
COPY . .
RUN dotnet publish -c Release -o /app --use-current-
runtime --self-contained false --no-restore

# final stage/image
FROM mcr.microsoft.com/dotnet/runtime:7.0
WORKDIR /app
COPY --from=build /app .
ENTRYPOINT ["dotnet", "dotnetapp.dll"]
```

Kubernetes + AKS



Microsoft

DEMO

Kubernetes + AKS – Principais Recursos

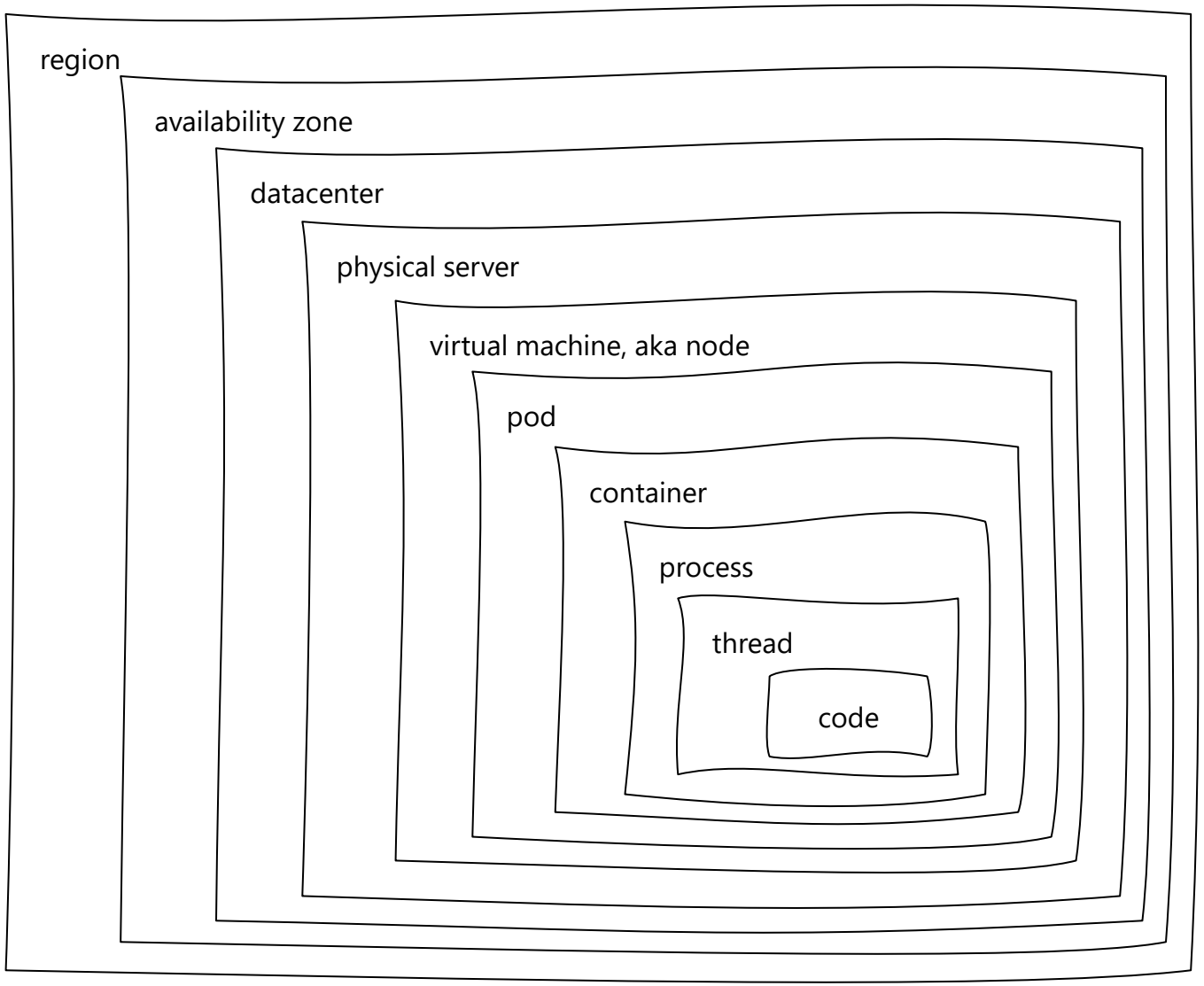
- **Workload**
  - Pod
  - Deployment
  - DaemonSet
  - ReplicaSet
  - StatefulSet
- **Networking**
  - Service
  - Endpoint
  - Ingress
  - Ingress Controller
  - Network Policy
- **Configuration**
  - ConfigMap
  - Secret
  - HPA
- **Storage**
  - Persistent Volume
  - Persistent Volume Claim
  - Storage Class

Kubernetes + AKS – Anatomia de um Recurso

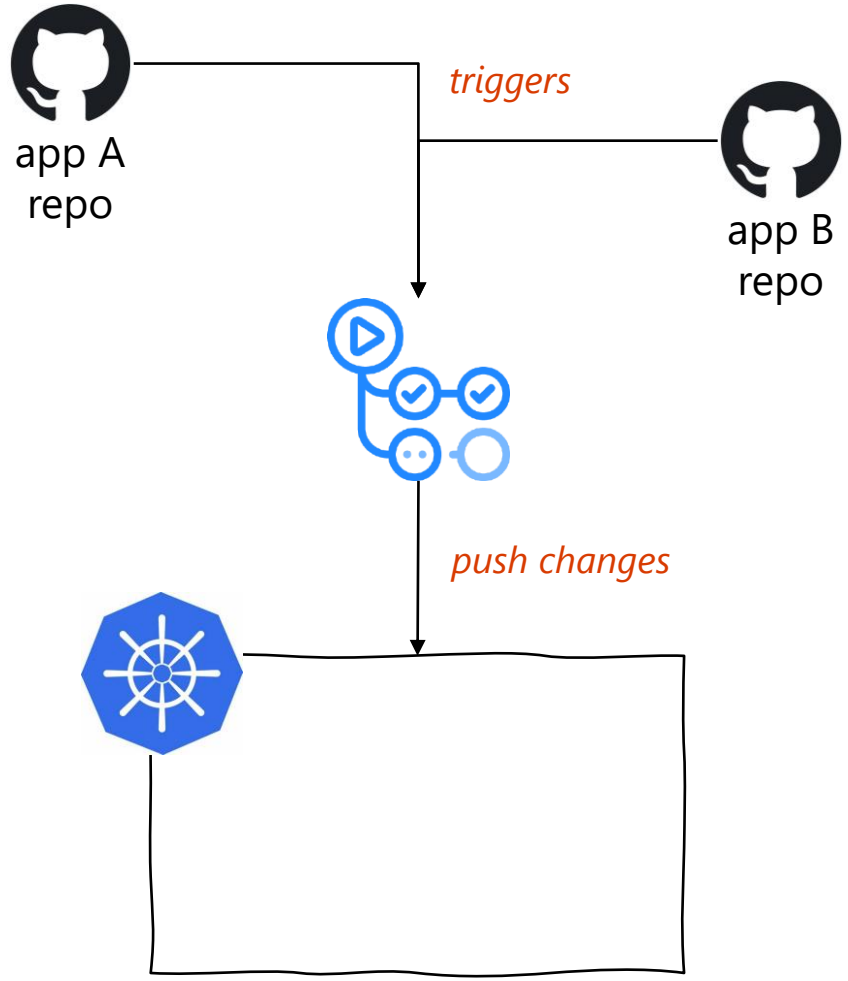
```yaml
apiVersion: v1
kind: Pod
metadata:
  name: ingress-nginx-1670504232-controller-55b9799b87-k6zpn
  namespace: default
  labels:
      pod-template-hash: 55b9799b87
spec:
  containers:
    - name: controller
      image: >-
        registry.k8s.io/ingress-nginx/controller:
      args:
        - /nginx-ingress-controller
        - '--publish-service=$(POD_NAMESPACE)/ingr
        - '--election-id=ingress-nginx-1670504232-
        - '--controller-class=k8s.io/ingress-ngin>
ports:
        - name: https
          containerPort: 443
          protocol: TCP
      env:
- name: POD_NAMESPACE
          valueFrom:
            fieldRef:
              apiVersion: v1
              fieldPath: metadata.namespace
resources:
        requests:
          cpu: 100m
          memory: 90Mi
```

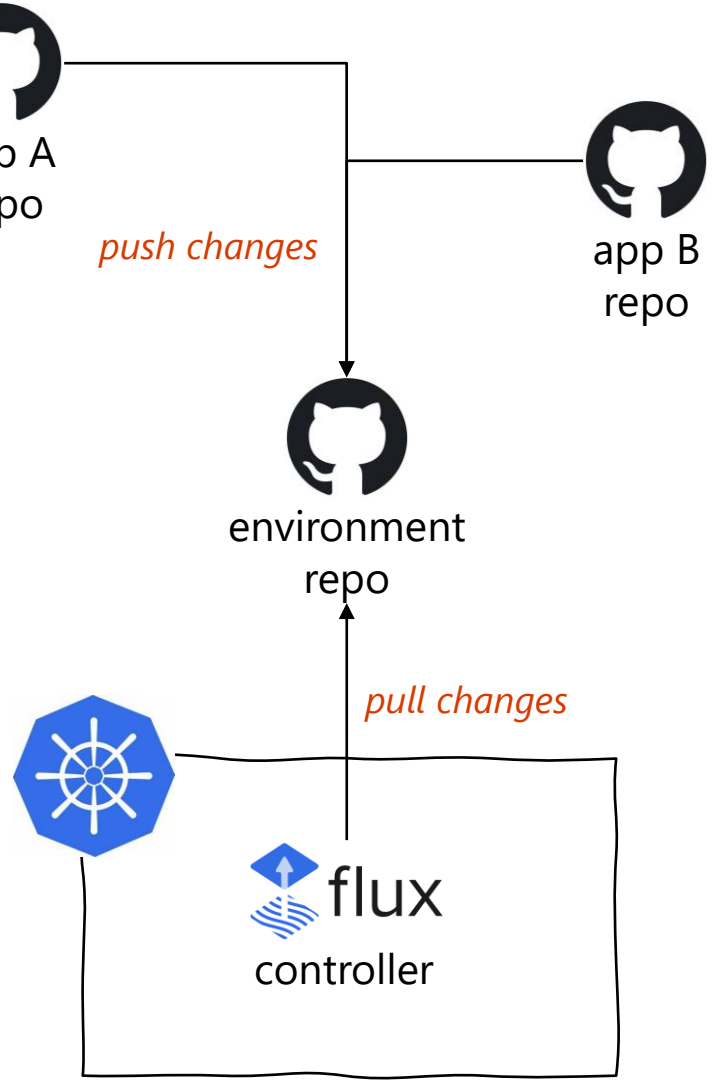
```yaml
apiVersion: v1
kind: Service
metadata:
  name: customers
  namespace: default
  labels:
    app.kubernetes.io/managed-by: Helm
  annotations:
    meta.helm.sh/release-name: customers
    meta.helm.sh/release-namespace: default
spec:
  ports:
    - protocol: TCP
      port: 80
      targetPort: 8080
  selector:
    app: customers
```

Kubernetes - Camadas



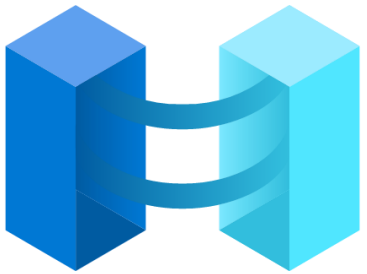
Push (Pipelines) VS Pull (GitOps)





Azure Arc-enabled Kubernetes

- É uma funcionalidade do Azure Arc, que é um produto para nuvems híbridas.
- Permite gerenciar clusters Kubernetes em outros provedores cloud ou ainda em ambiente on-premisses.
- Um agente é instalado via Helm no cluster que se comunica com o Azure.
- Não é necessário nenhuma liberação de firewall para a conexão. A comunicação é somente de saída.
- O cluster externo torna-se um recurso padrão no Azure.
- Uma vez o cluster conectado no Azure é possível:
  - Ponto central de administração de todos os clusters Kubernetes para inventário, tagging, agrupamento, etc.
  - Visualizar e monitorar com o Azure Monitor.
  - Proteger contra ataques com o Microsoft Defender for Kubernetes.
  - Garantir a governança com o Azure Policy.
  - Acessar de forma segura qualquer cluster.



Microsoft

DEMO

Kubernetes Event-drive Autoscaling

- Projeto open-source desenvolvido inicialmente em parceria com a Red Hat.
- Encontra-se em incubação CNCF.
- Complementa o uso do Horizontal Pod Autoscaler (HPA) existente no Kubernetes.
- É configurado através de Custom Resource Definitions.
- Permite escalar horizontalmente containers utilizando *scalers*.
- Possui um conjunto rico de *scalers*, como:
  - Azure Application Insights
  - Azure Log Analytics
  - Azure Service Bus
  - Apache Kafka
  - Microsoft SQL Server
  - RabbitMQ

Microsoft

DEMO

Azure Spring Apps

- Desenvolvido em parceria com a VMware.
- Permite a execução de aplicações desenvolvidas em Spring com Java e também em Steeltoe com .NET.
- Agrega valor ao Kubernetes criando um produto dedicado para execução de aplicações.
- Abstrai detalhes de gerenciamento do Kubernetes.
- Possui três *tiers*, sendo que no *tier* Enterprise é possível utilizar os componentes do Tanzu.
- Os componenes do Tanzu incluem:
  - Service Registry
  - Build
  - Configuration Service
  - Cloud Gateway

Spring

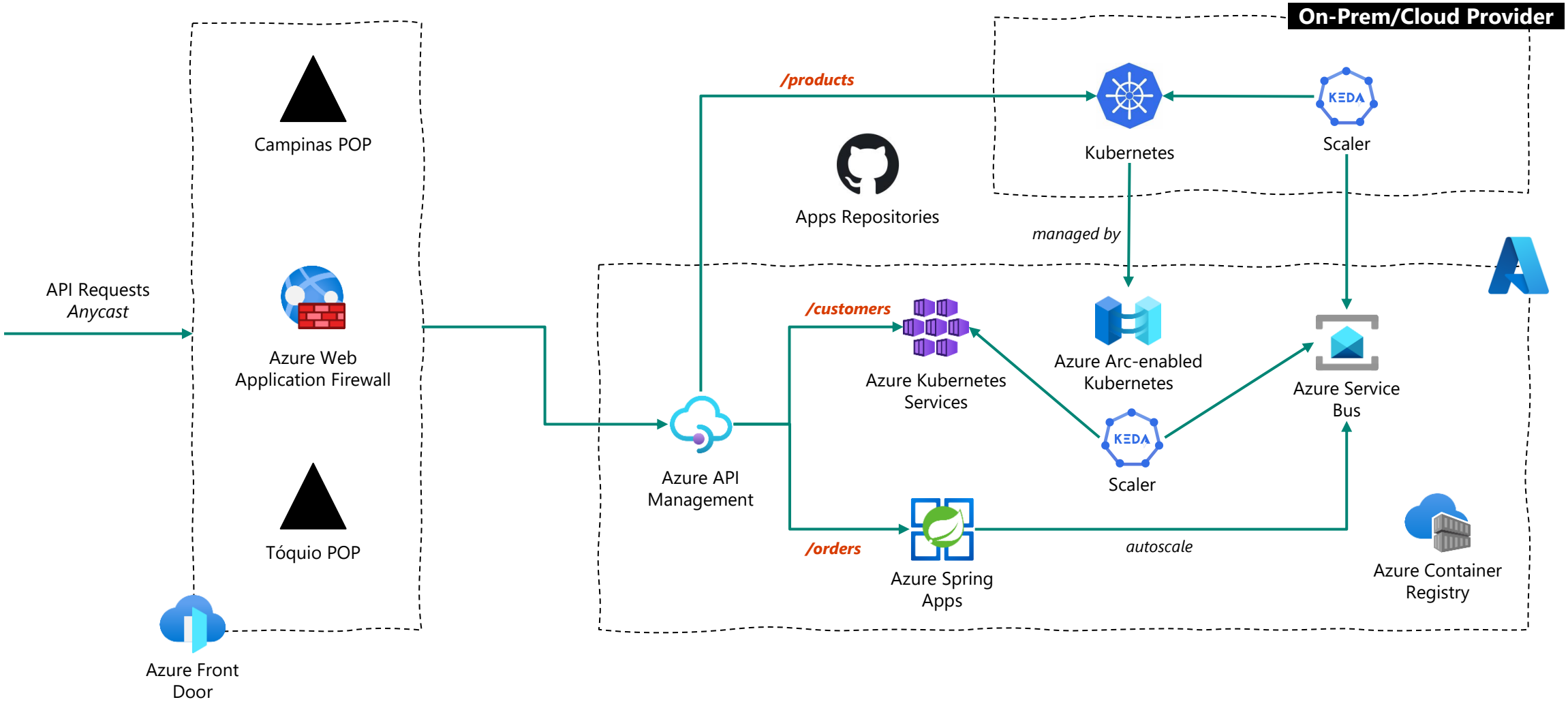
Steeltoe

Tanzu

Microsoft

DEMO

Architecture



Microsoft

DEMO

Microsoft

Obrigado!