Tutorial Docker e Kubernetes

avanade

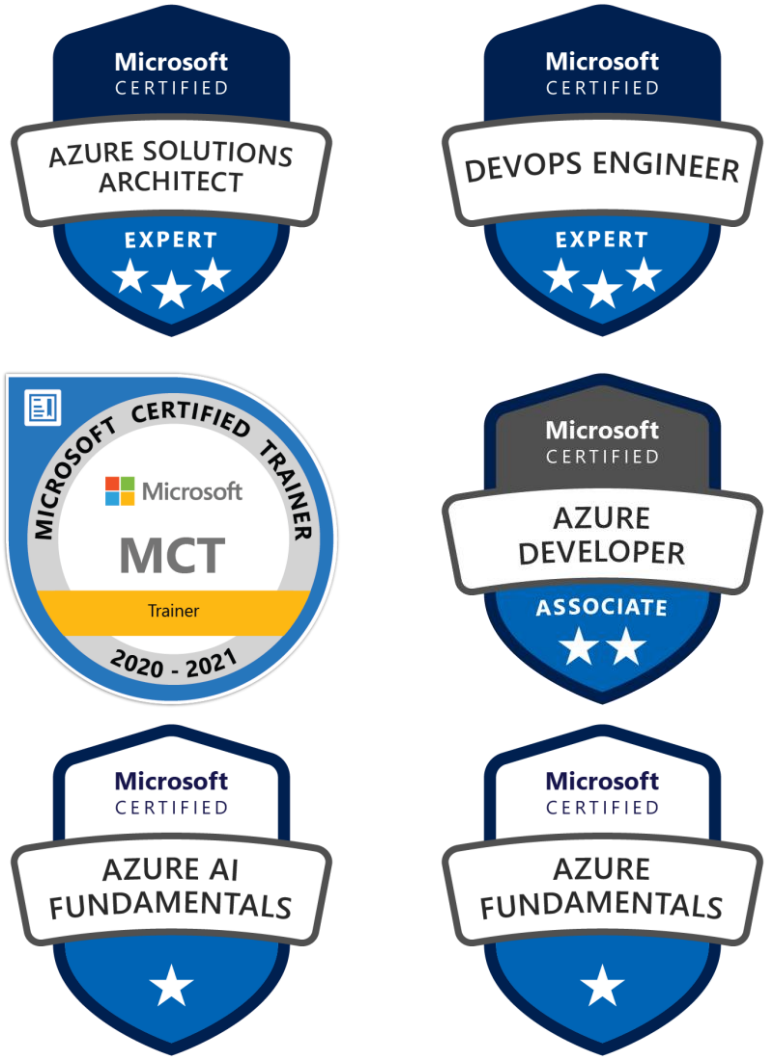
Sobre mim

Diretor – Enterprise Tech Arch Leader
https://www.linkedin.com/in/aracz/
andre.racz@avanade.com

- + de 20 anos de carreira
- + de 15 anos em Arquitetura
- 2,5 anos de Avanade

Projetos em grandes empresas dos segmentos: Bancos / Seguradoras / Setor Público / Industrias / Farmacêutico / Hospitais / Energia

Tecnologias: Java, .Net, NodeJS, Kubernetes, Cloud, Devops, IA

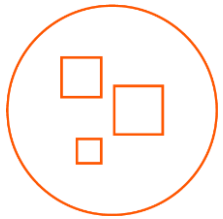


avanade

Containers 101 - 5 cool things

Containers são uma grande mudança de VMs

Elimine a proliferação de VMs, seu overhead e diminua o custo de infra.



Qualquer aplicação, qualquer SO, qualquer Plataforma)

Windows, Linux, Cloud

.Net, Java, Node, etc..

Pets v/s Cattle

Imutável: reconstrua e redeploye x atualização

Infra-estrutura de próxima geração

Plataforma única com único plano de controle para aplicações legadas ou modernas

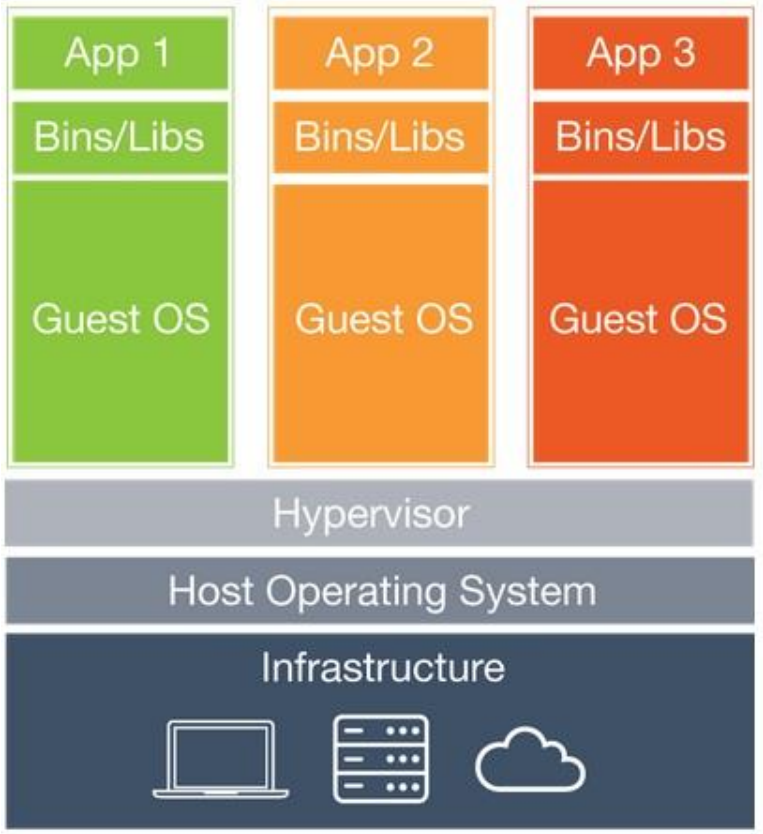
Docker e Kubernetes

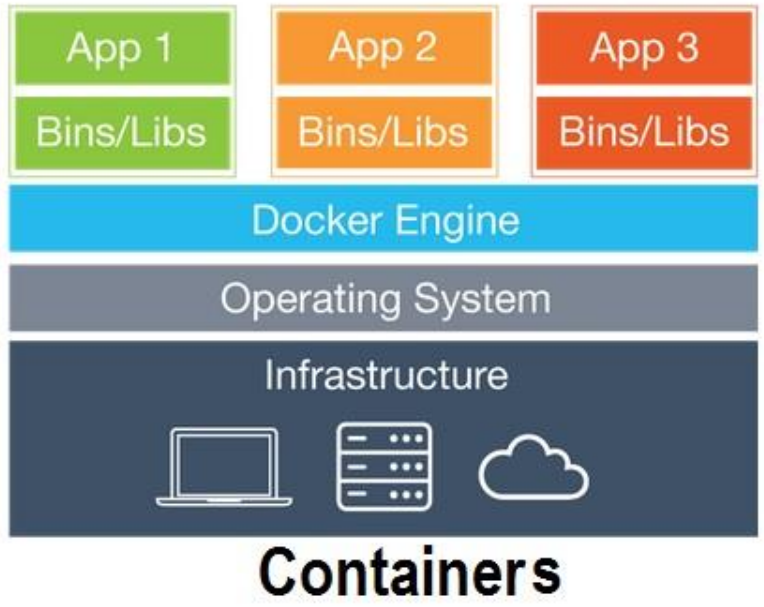
Docker Containers e Kubernetes são padrões de fato.

Escalável, Autom-reparável, extensível e portável

avanade

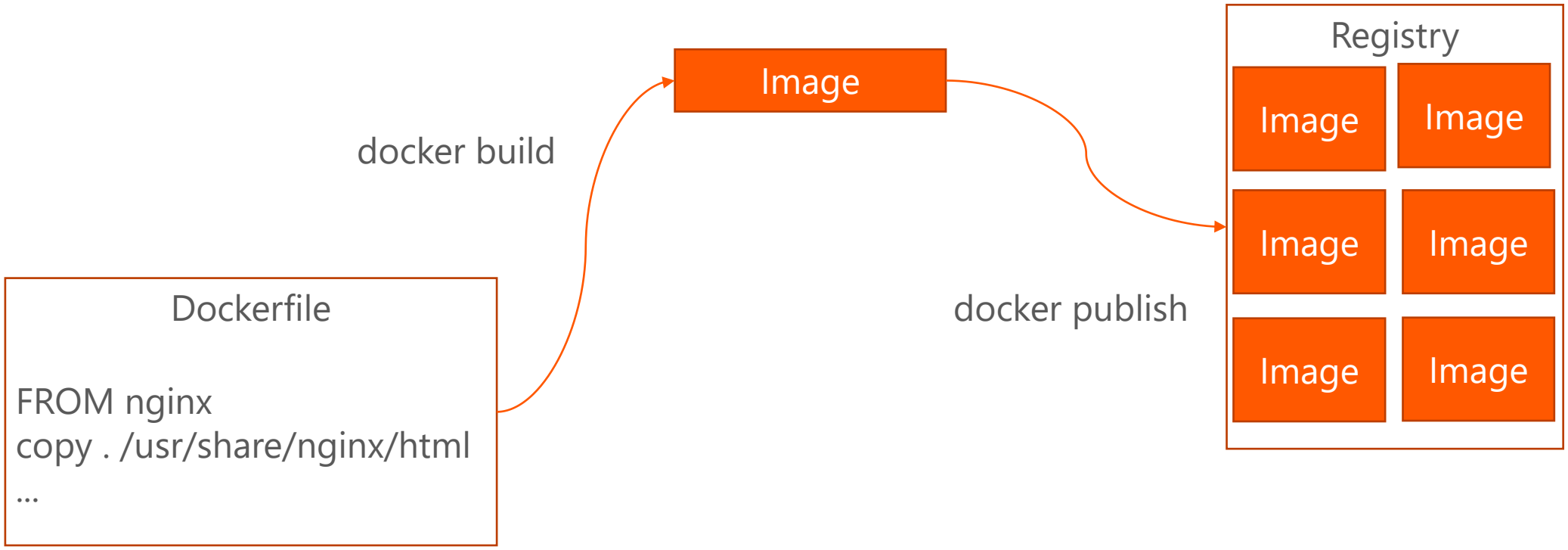
VMs x Containers





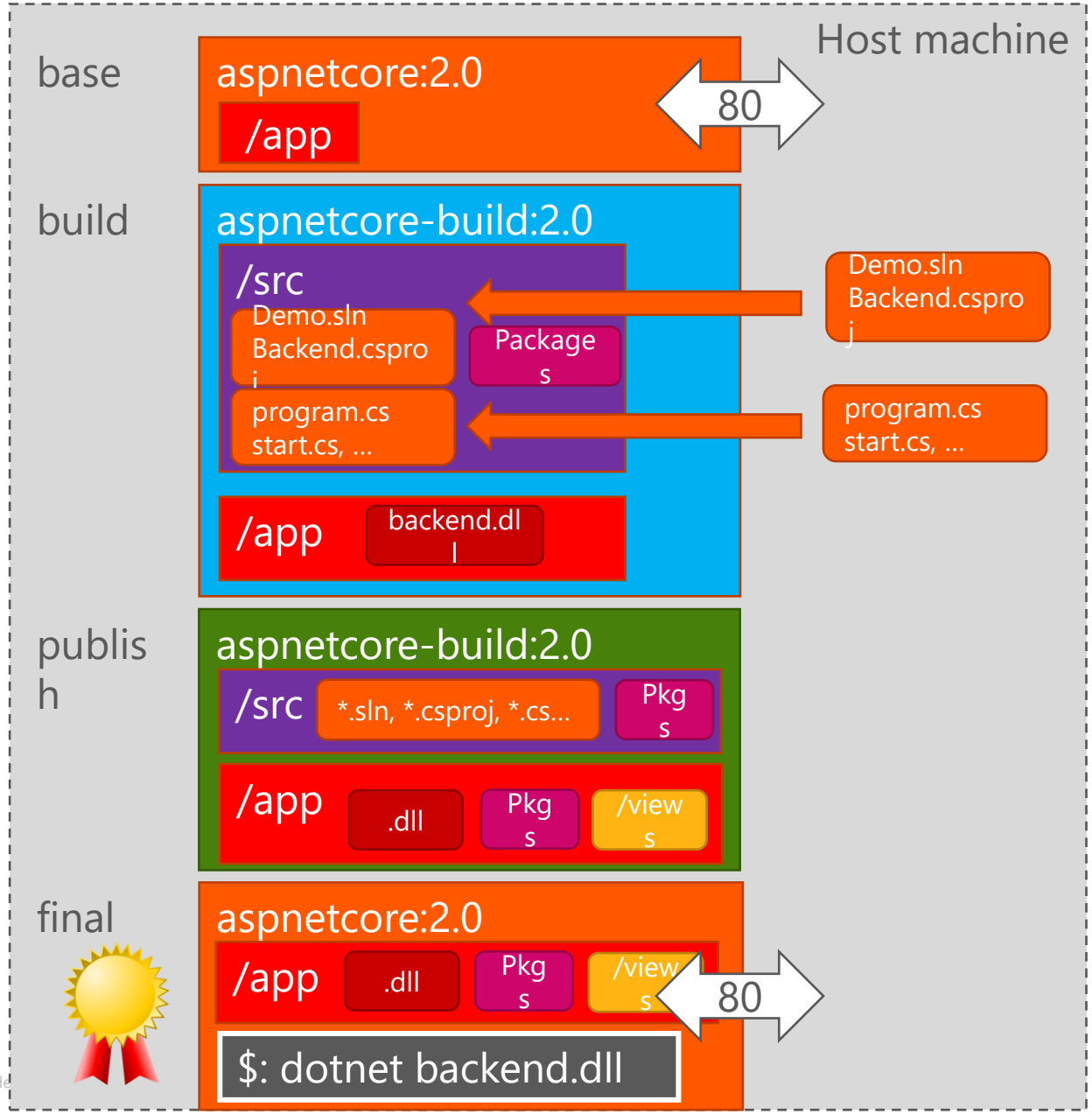
avanade

Docker - Conceitos



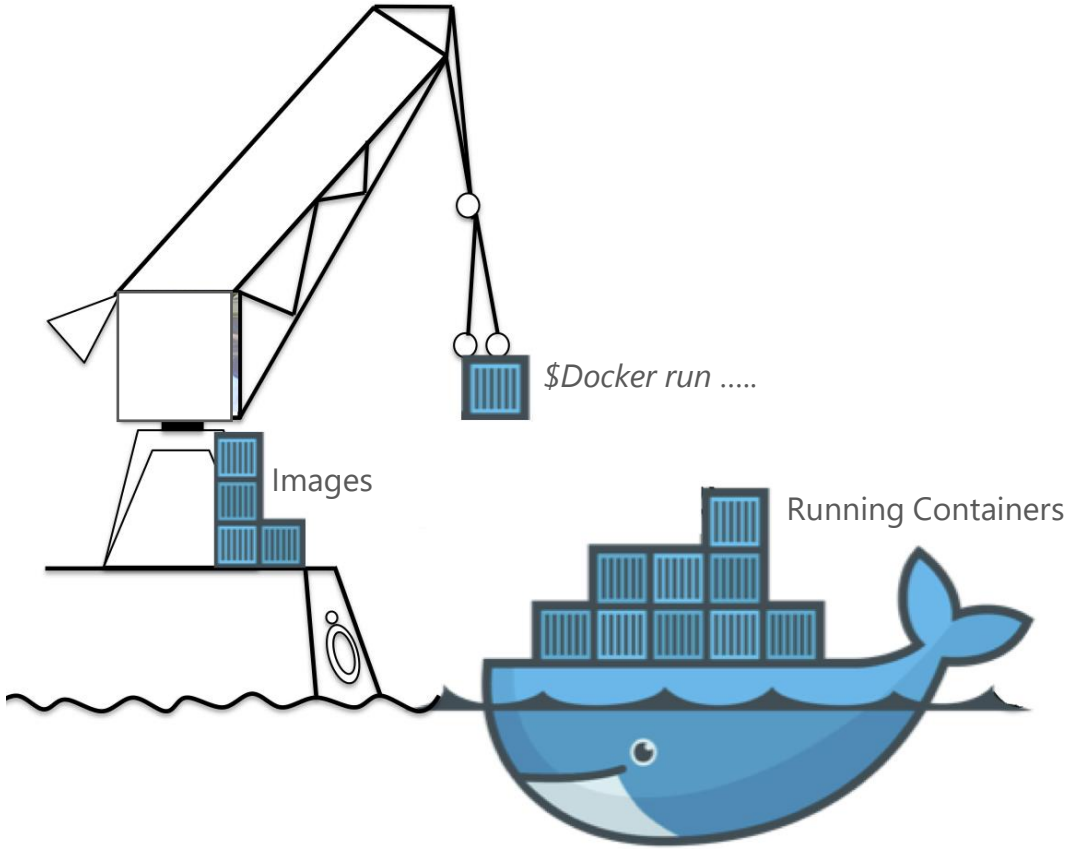
Container 101

```
FROM microsoft/aspnetcore:2.0 AS base
WORKDIR /app
EXPOSE 80

FROM microsoft/aspnetcore-build:2.0 AS build
WORKDIR /src
COPY *.sln ./
COPY backend/backend.csproj backend/
RUN dotnet restore

COPY . .
WORKDIR /src/backend
RUN dotnet build -c Release -o /app

FROM build AS publish
RUN dotnet publish -c Release -o /app


FROM base AS final
WORKDIR /app
COPY --from=publish /app .
ENTRYPOINT ["dotnet", "backend.dll"]
```



avanade

Containers 101



avanade

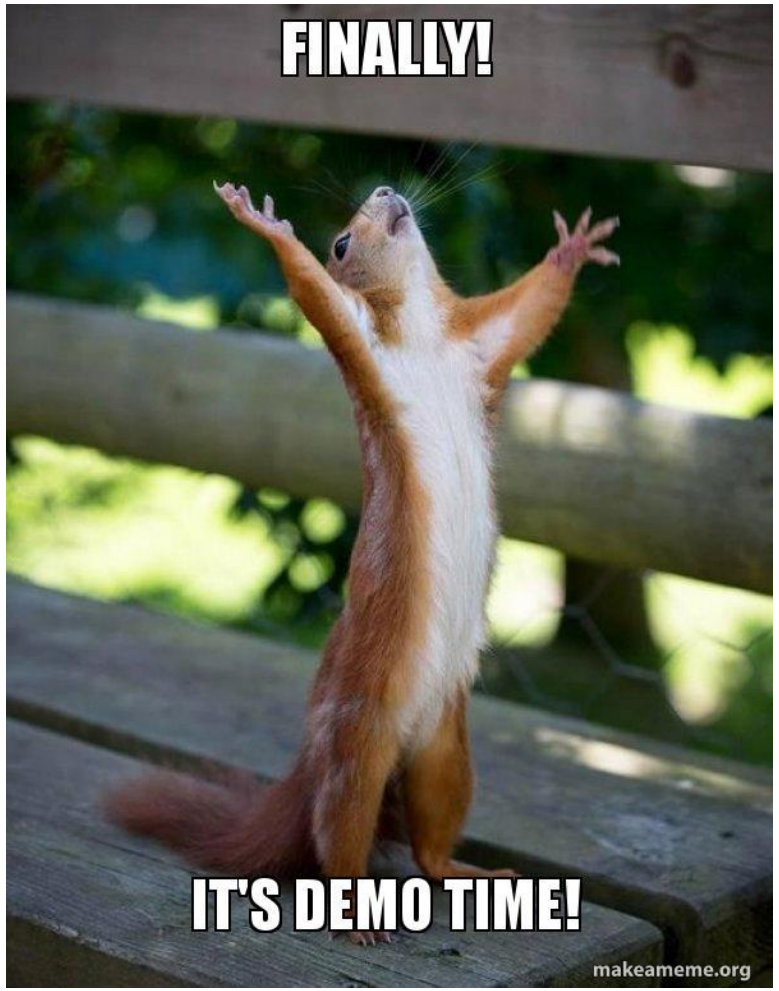
Containers 101

O conceito de containers está sendo muito utilizado em novas aplicações. É ideal para tipos de arquiteturas distribuídas, como por exemplo arquitetura em microserviços ou sistemas portáteis.

Principais vantagens

- O tamanho das imagens é bem menor ao tamanho de uma vm. Enquanto uma vm tem o tamanho de Gb, a média de tamanho de um container é em Mb. Com isso uma máquina host consegue suportar muito mais imagens.
- A execução de containers consome muito menos recursos, permitindo que as máquinas reais executem muito mais processos.
- Provisionar e executar um novo container é muito rápido, a execução leva poucos segundos.
- Containers pode ser um grande aliado na montagem de novos ambientes.
- Com o container, testes podem ser menos complicados, pois o velho ditado “mas funciona na minha máquina...” é eliminado. Como a imagem é única, a mesma pode ser provisionada em ambiente de desenvolvimento, teste e produção.

Containers na Prática

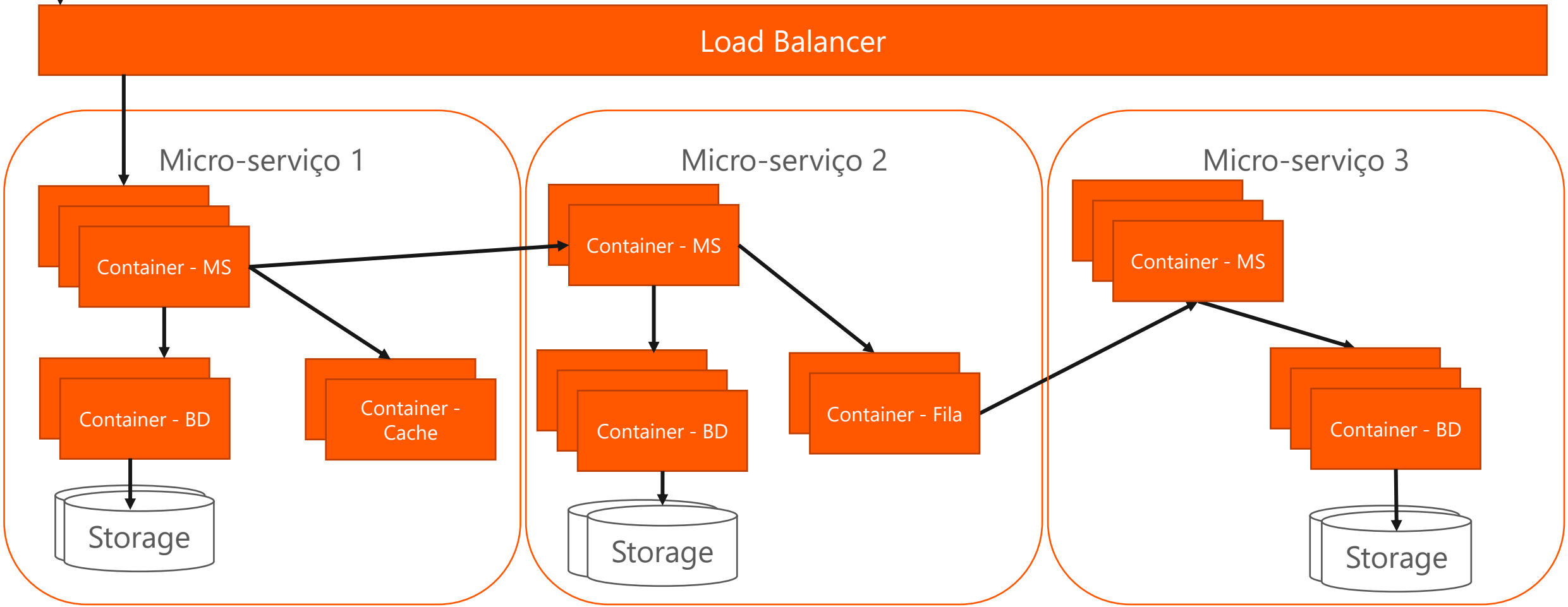


avanade

Orquestração de Containers

avanade

Por que Orquestração de Containers?



avanade

Por que Orquestração de Containers?

- Como conectar os containers? Aplicação, banco de dados, etc...
- Como fazer o balanceamento de carga?
- Como reiniciar um container quando ele cair?
- Como escalar os containers automaticamente?
- Como gerenciar armazenamento dos containers?
- Como gerenciar senhas?
- Como controlar segurança (qual serviço pode acessar o que)?
- Como dar acesso seguro aos desenvolvedores?

avanade

Por que Orquestração de Containers?

Para que seus containers dentro de sua infra estrutura não vire essa bagunça, se faz necessário o uso de orquestradores de containers. Os orquestradores otimizam a organização e o gerenciamento de suas aplicações distribuídas em containers. O orquestrador mais famoso é o Kubernetes

avanade

Azure Kubernetes Service (AKS)

Kubernetes made easy

Kubernetes é uma plataforma open-source para automatizar o deployment, escalação e operações de containers de aplicação entre um cluster de hosts. Você é responsável pela Arquitetura, construção e gestão.

AKS é a plataforma de Kubernetes gerenciada da Microsoft; A Azure é responsável pela gestão, escala e operação dos nós de controle e os usuários escalam os nós de trabalho e deployam os workloads.

avanade

Fundamentos de Orquestração de Contêineres

Os orquestradores de contêineres são os sistemas que gerenciam toda a plataforma de contêineres. Os orquestradores permitem que aplicativos contêinerizados sejam executados em escala.

AGENDAMENTO

- Certifica que o número especificado de réplicas está sempre rodando
- Certifica que os recursos são usados de forma eficiente dentro de restrições

AFINIDADE

- Define restrições para controlar onde os contêineres podem ser executados
- Geografia, tipo de host, OS, etc.

MONITORAMENTO

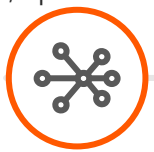
- Executa continuamente verificações de saúde para monitorar os contêineres em busca de falhas

RESILIÊNCIA

- Reequilíbrio automático dos contêineres após falhas no host ou aplicativo
-

ESCALA

- Ajusta automaticamente a capacidade para manter o desempenho estável e previsível
- Inclui containers/pods e/ou worker nodes



REDE

- Provisionamento sob demanda de redes definidas por software que se estendem por vários hosts
- Controle a comunicação entre contêineres e o mundo exterior.

SERVICE DISCOVERY

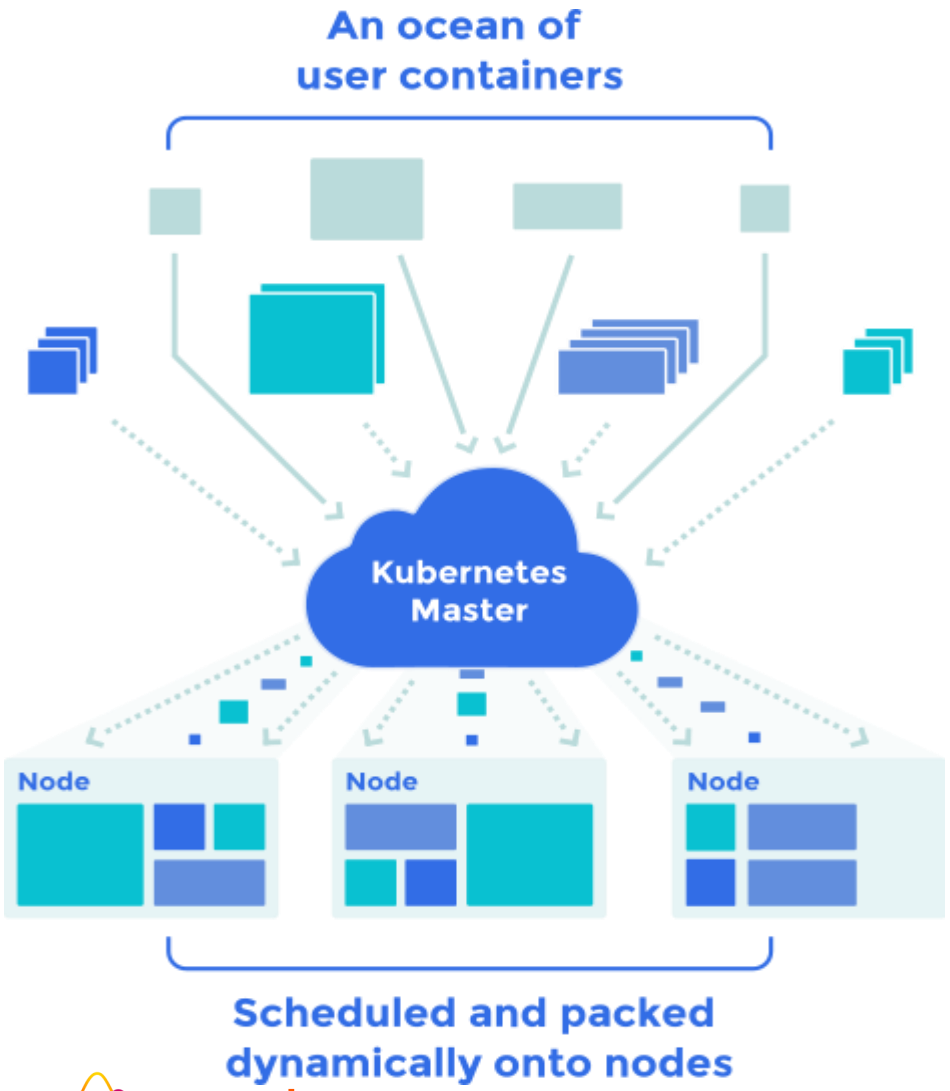
- Aplicativos em contêineres podem localizar outros serviços que fazem parte do mesmo aplicativo.
- Um sistema interno de DNS integrado com nomes de contêiner/pod

ATUALIZAÇÕES COORDENADAS

- Gerencia atualizações de contêineres para evitar o tempo de inatividade do aplicativo e habilite a reversão se ocorrerem falhas.
- Capacidade de implantação blue/green

- Infraestrutura declarativa
- Auto-cura
- Dimensionamento horizontal
- Lançamentos automatizados e reversões
- Detecção de serviço e balanceamento de carga
- Rede de contêineres e entrada de contêineres
- Gerenciamento de armazenamento
- Gerenciamento de secrets e de configuração

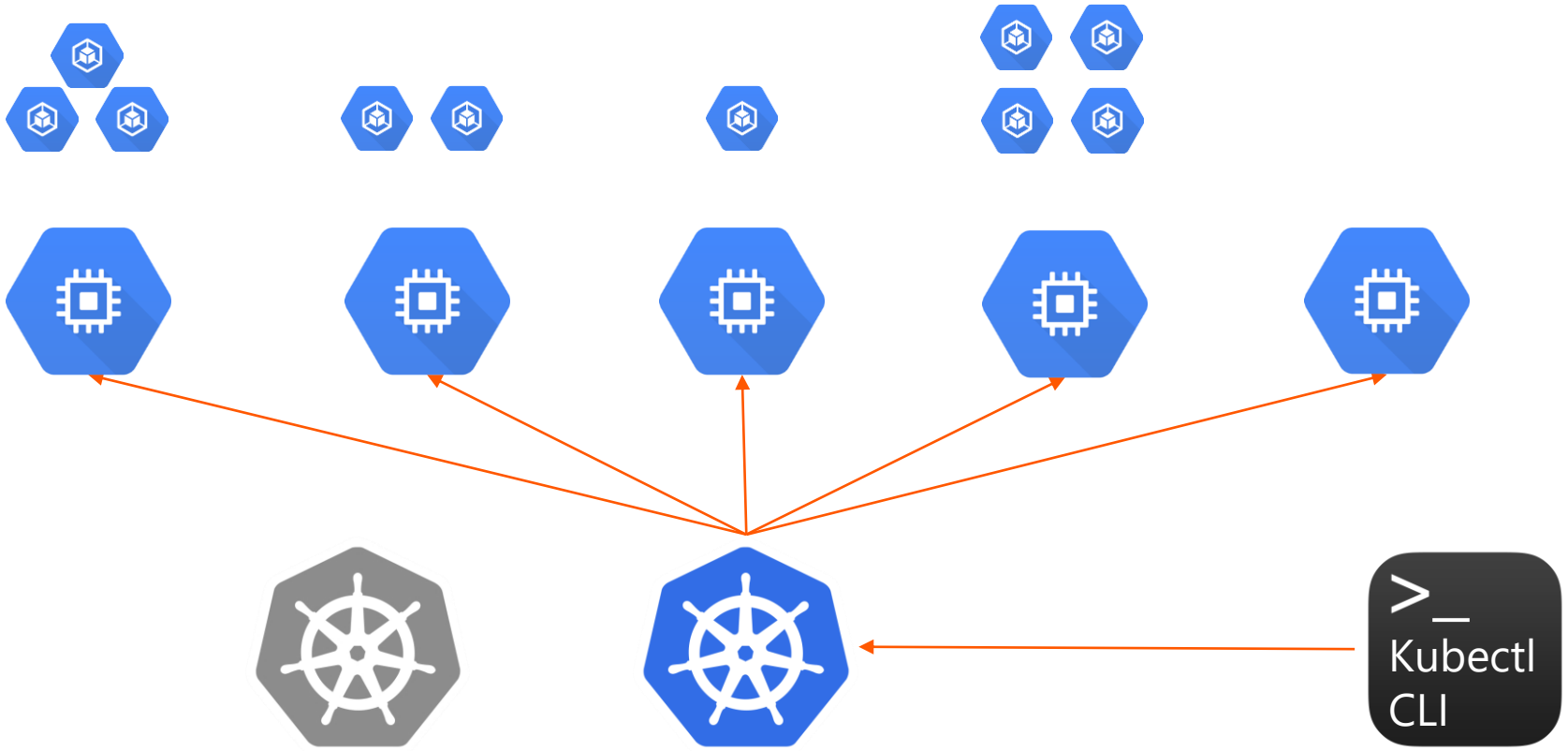
Fundamentos de Kubernetes



- Kubernetes é um projeto de código aberto para gerenciar um cluster de máquinas como um único sistema, gerenciando e executando contêineres Docker em vários hosts. É um orquestrador de contêineres, não uma "plataforma" completa por si só.
- A tecnologia Kubernetes é usada no Google há mais de 10 anos (Borg Scheduler).
- É agora um projeto de código aberto a partir de junho de 2014 com muitos mantenedores, incluindo Microsoft, Red Hat, Cisco, outros.
- Não determina rede, armazenamento persistente, registros de imagem, segurança, etc.
- É complexo instalar, configurar e usar porque muitos componentes da plataforma devem ser usados e projetados para construir uma plataforma completa de contêineres.

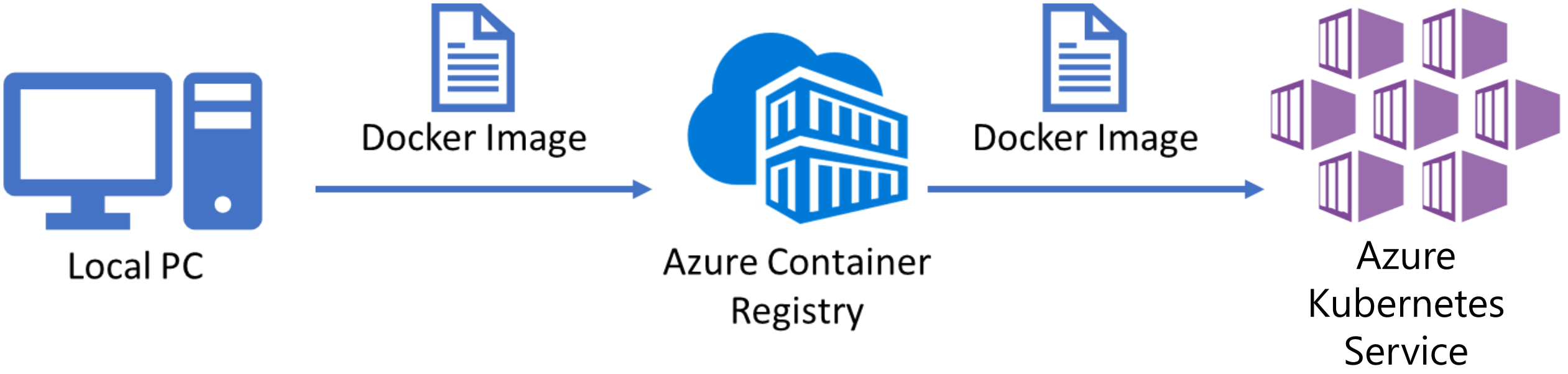
avanade

Kubernetes architecture

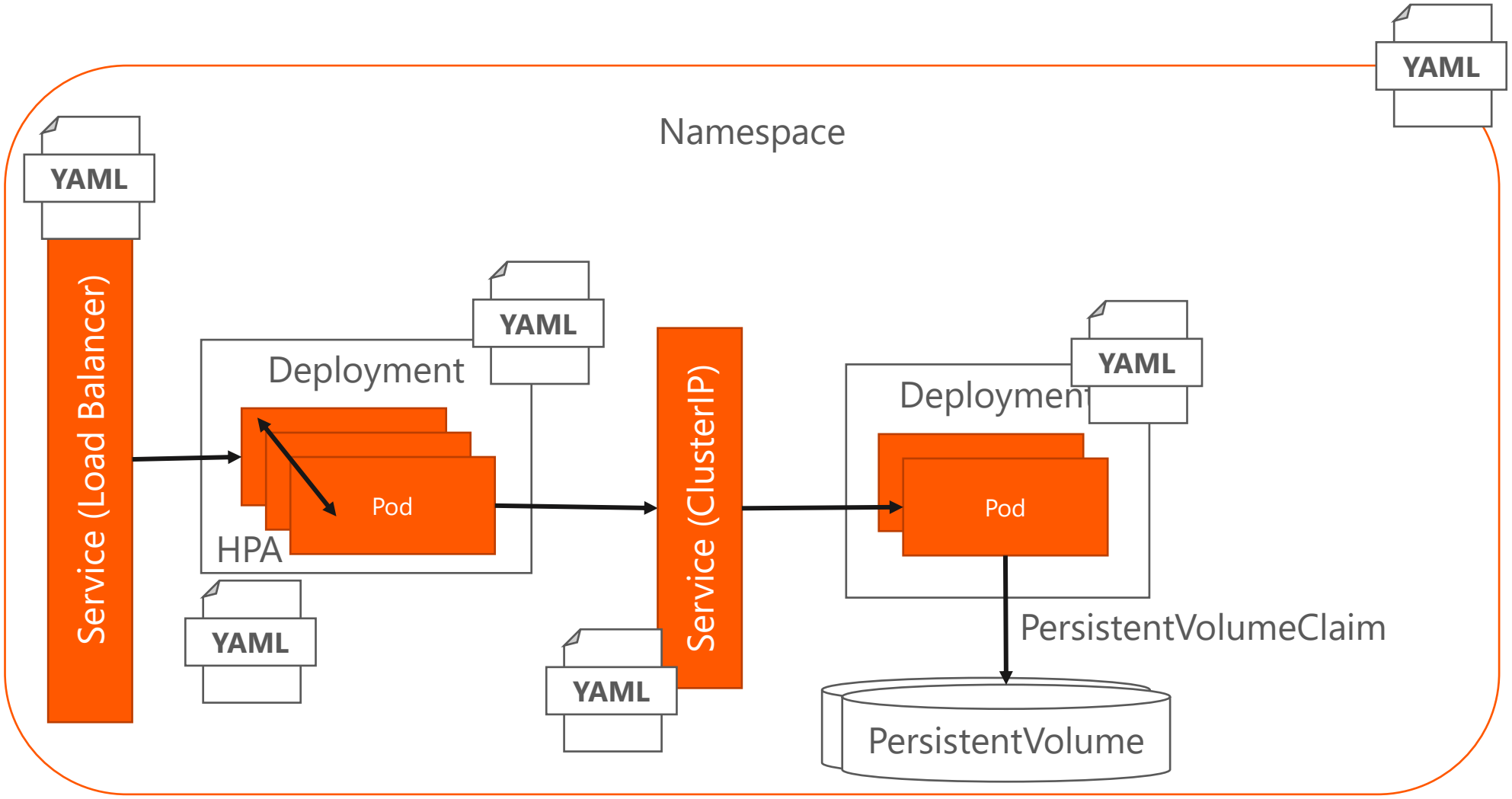


avanade

Container Registry Fundamentals



Objetos Kubernetes



avanade

Kubernetes – Demo time

avanade

Obrigado!

https://github.com/andreracz/TutorialDockerKubernetes

avanade