

Deployment, Pod e Container



O que é e para que serve cada um deles?



BL²



TI direto ao
ponto!

Imperativo

- Preocupado em “**como fazer**”
- Instruções passo-a-passo
- Maior controle
- Necessário conhecimento de como fazer



Declarativo

- Preocupado em “**o que fazer**”
- Declaração da intenção, ou seja, do resultado esperado
- Configuração
- Mais abstrato
- Não exige conhecimento de como fazer



Deployment [1]

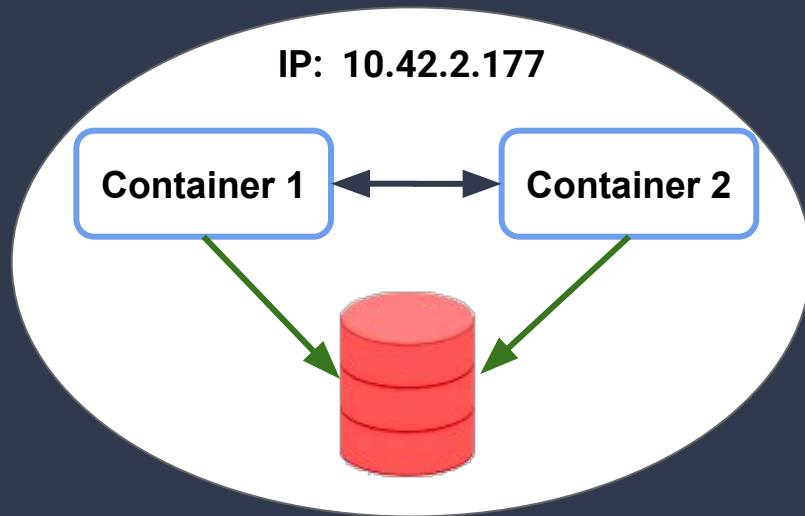


Sonho a ser atingido!



- Definição do estado esperado / desejado:
 - Imagem utilizada
 - Número de instâncias
 - Portas abertas
 - Variáveis de ambiente
- Utilizado para:
 - Implantar aplicações no *cluster*
 - Escalar aplicações
 - Realizar *rollback*
 - Definir modo de atualização da aplicação
 - Atualizar a aplicação

Pod [2]



Unidos, venceremos! Separados, não subiremos.

- Menor unidade de implantação
- Grupo de *containers* que compartilham:
 - armazenamento (volumes)
 - recursos de rede
 - namespace de processos
- Representam um serviço (acoplamento)
- Exemplos: *proxies*, obtenção de dados, envio de dados



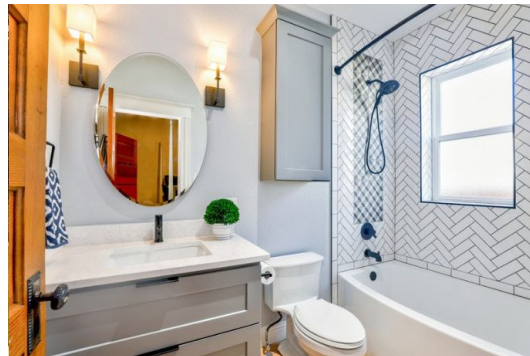
Container [3]



Cada aplicação no seu quadrado...
Ops, container!



- Encapsula uma determinada aplicação e suas dependências
- Cada container pode ter necessidades únicas para ser executado, como variáveis de ambiente, volumes e portas



ReplicaSet [4]

Nem mais, nem menos!



- Responsável por garantir que o número desejado de *pods* esteja rodando
- Cria ou remove *pods* conforme necessário



Deployment

Declara como serão nossos *pods*:
containers, imagens, portas, volumes, variáveis de ambiente e número de réplicas



ReplicaSet

Responsável por garantir que o número de réplicas existentes (*pods*) seja igual ao número planejado



Pod

Unidade de implantação (serviço), isola os *containers* existentes nele permitindo que se comuniquem



Container

Encapsula a aplicação e suas dependências

