

# Publicação

---

IIS / NGINX / DOCKER

# IIS

---

## Vantagens do Asp Net Core

- Velocidade e facilidade no desenvolvimento!
- Velocidade porque o Asp.Net Core é muitas vezes mais rápido que o Asp.Net convencional.
- Facilidade pois você tem maior controle sobre toda a aplicação e como ela irá funcionar!
- Não podemos esquecer também que uma aplicação Asp.Net Core é multiplataforma: Windows, Linux e Mac.

# Fazendo a publicação no IIS

---

dotnet new mvc

```
C:\Core
λ dotnet new mvc
The template "ASP.NET Core Web App (Model-View-Controller)" was created successfully.
This template contains technologies from parties other than Microsoft, see https://aka.ms/template-3pn
for details.

Processing post-creation actions...
Running 'dotnet restore' on C:\Core\Core.csproj...
Restore succeeded.
```

# Publish

---

Este comando irá criar uma aplicação padrão Asp.Net Core MVC. Você pode também criar a aplicação usando o Visual Studio 2017!

Agora vamos publicar esta aplicação em um diretório, que depois será adicionado ao IIS:

`dotnet publish -o c:\temp\core`

```
C:\Core
λ dotnet publish -o c:\temp\core
Microsoft (R) Build Engine version 15.3.117.23532
Copyright (C) Microsoft Corporation. All rights reserved.

Core -> C:\Core\bin\Debug\netcoreapp2.0\Core.dll
Razor view compilation for Core -> obj\Debug\netcoreapp2.0\Core.PrecompiledViews.dll
Core -> c:\temp\core\
```

---

Agora temos os arquivos da publicação no diretório c:\temp\core:

```
Directory of c:\temp\core

06/08/2017  17:19    <DIR>          .
06/08/2017  17:19    <DIR>          ..
06/08/2017  17:17             366 appsettings.Development.json
06/08/2017  17:17             228 appsettings.json
06/08/2017  17:17             207 bower.json
06/08/2017  17:17             628 bundleconfig.json
06/08/2017  17:19          333.726 Core.deps.json
06/08/2017  17:19             8.192 Core.dll
06/08/2017  17:19             1.604 Core.pdb
06/08/2017  17:19          76.288 Core.PrecompiledViews.dll
06/08/2017  17:19             240 Core.runtimeconfig.json
06/08/2017  17:19             378 web.config
06/08/2017  17:19    <DIR>          wwwroot
06/08/2017  17:19    <DIR>          x64
06/08/2017  17:19    <DIR>          x86
                10 File(s)          421.857 bytes
```

# Deploy

---

Vamos então publicar no IIS, mas antes disto, precisamos instalar o **Windows Server Hosting** do Asp.Net Core, que você encontra no site do [DotNet Core](#), vamos baixar e instalar:



# Deploy

---

O Windows Server Hosting é quem irá realizar a execução das aplicações.

Agora vamos para o IIS!

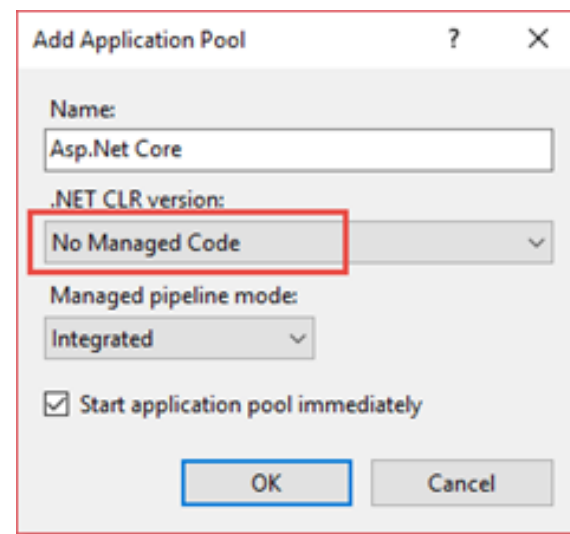
Primeiro vamos criar um Application Pool para nossos sites em Asp.Net Core, para isto abra o IIS, vá em Application Pools e clique em “Add Application Pool...”

# Deploy

---

No nosso exemplo estamos criando um Application Pool chamado “Asp.Net Core”, e o segredo aqui é colocar “No Managed Code”, isto porque o WebHost que instalamos irá fazer o gerenciamento da aplicação.

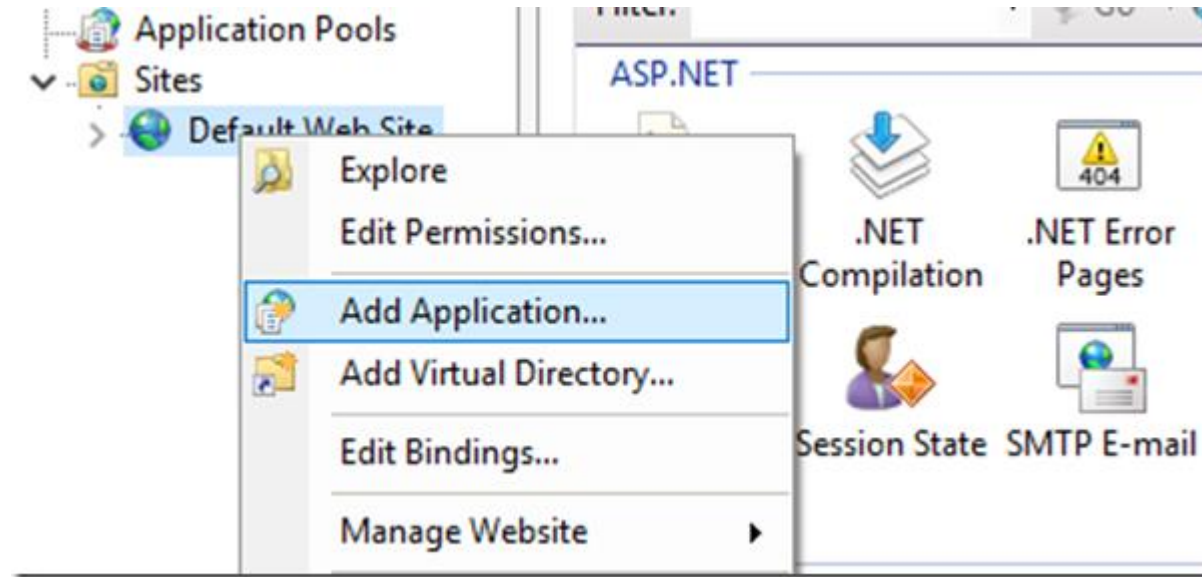
Agora é só criarmos nossa aplicação e colocarmos neste pool. Lembrando que para criar a aplicação basta clicar com o botão direito em Site e depois em Add Application:





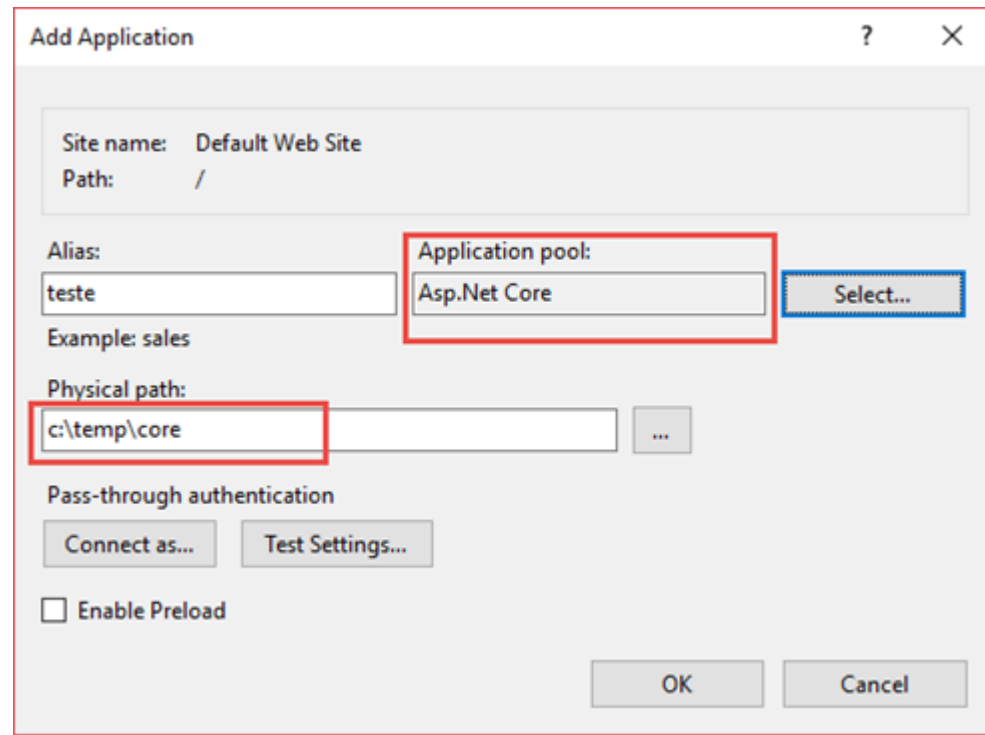
# Deploy

---



# Depois criamos a nossa aplicação:

---



Em seguida, Acessar via localhost

# NGINX

---



Baixar o .NET Runtime específico para Linux

Configurar o aplicativo como **FDD (implantação dependente de estrutura)**

Comando de publicação

`dotnet publish --configuration Release`

Testando o aplicativo

1. Na linha de comando, execute o aplicativo: `dotnet <app_assembly>.dll`.

2. Em um navegador, vá para `http://<serveraddress>:<port>`  
para verificar se o aplicativo funciona no Linux localmente.

# Novas possibilidades

---

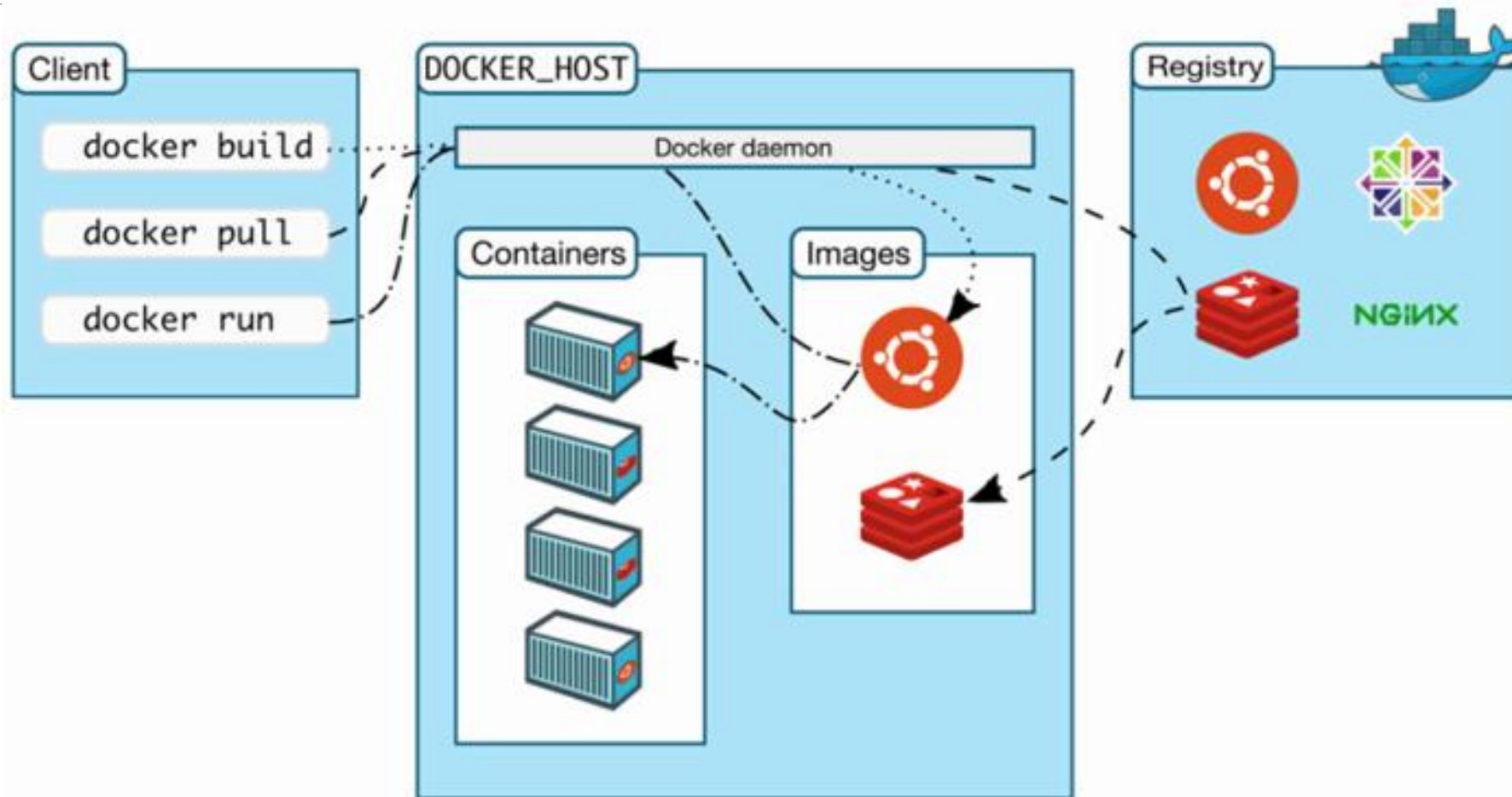


# Docker - Conceito

---

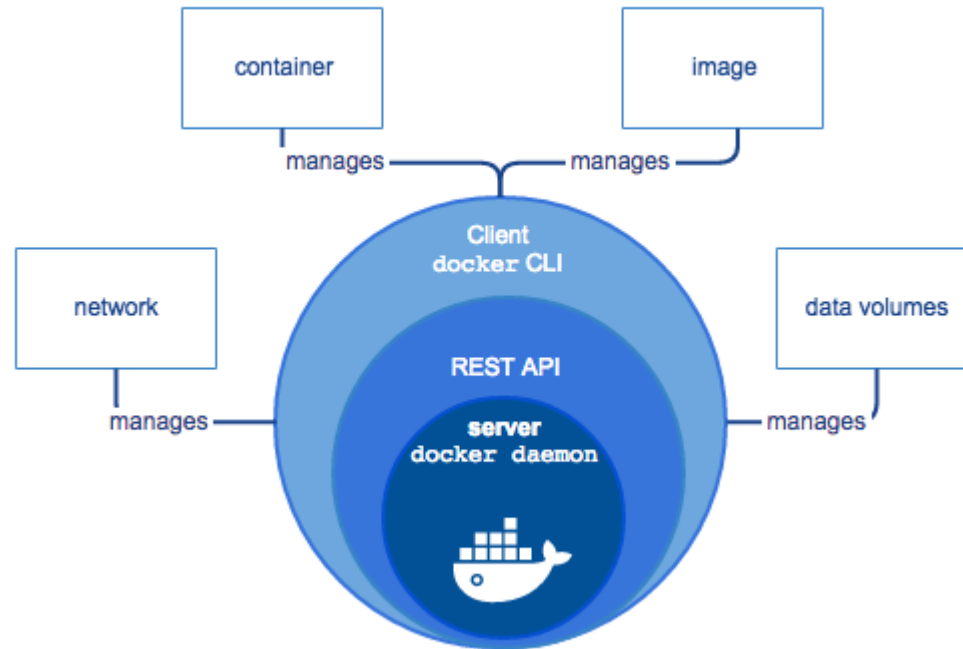
- Docker é uma plataforma de containers de software. É uma solução inovadora que elimina o grande problema do “na minha máquina funciona”.
- Docker elimina a necessidade de criação de ambientes de desenvolvimento e testes
- Docker é uma ótima alternativa ao uso de Virtual Machines, pois é muito mais leve, rápido, prático e simples!

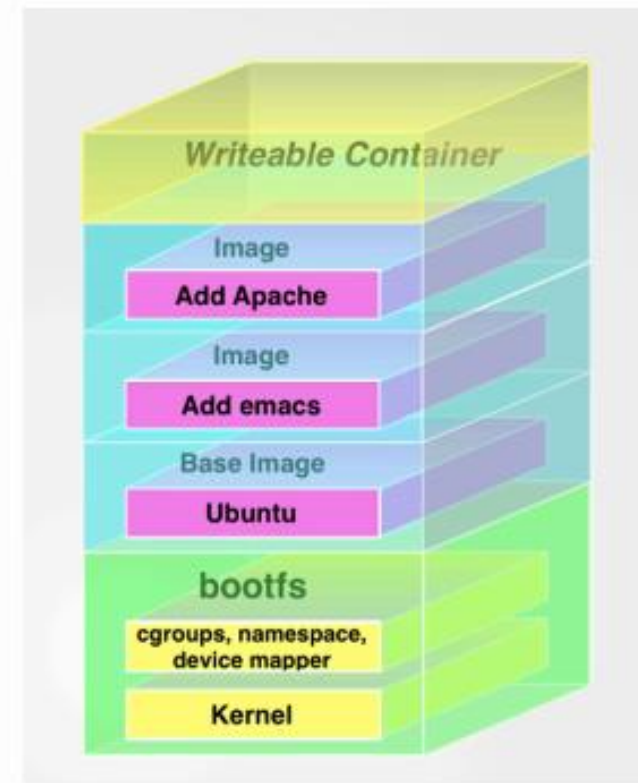
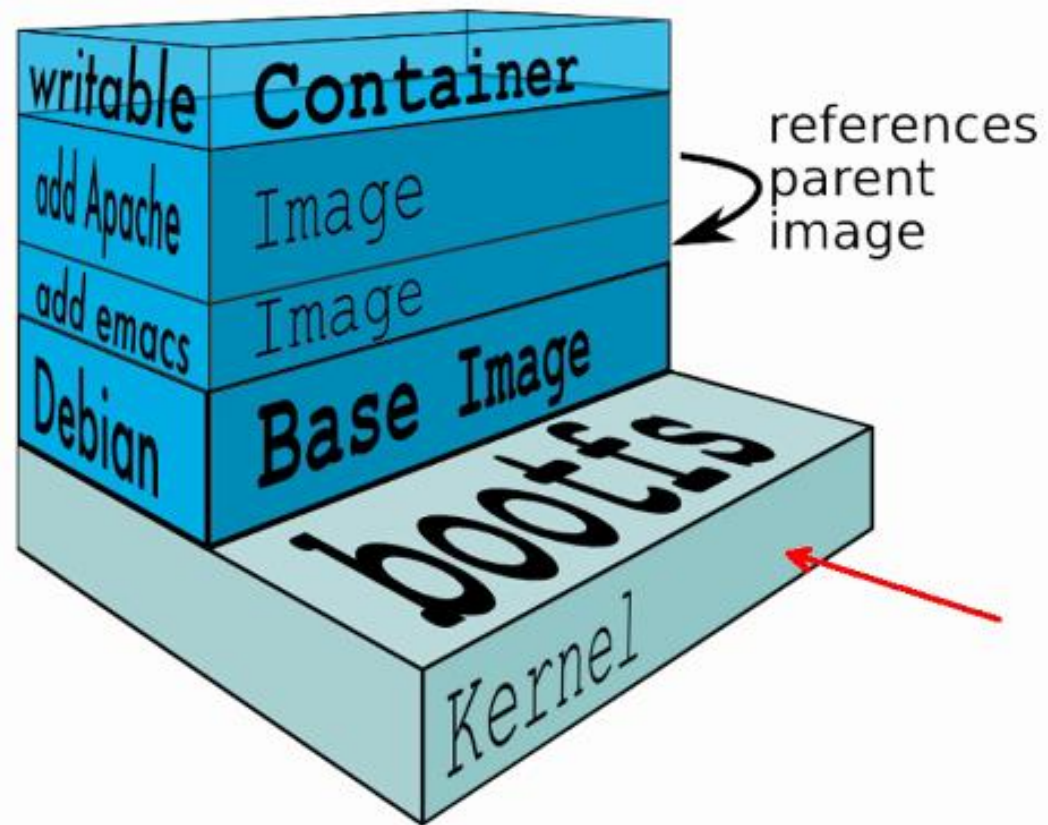
# Básico sobre Docker - Containers



# Docker

---







# Docker – Básico sobre imagens

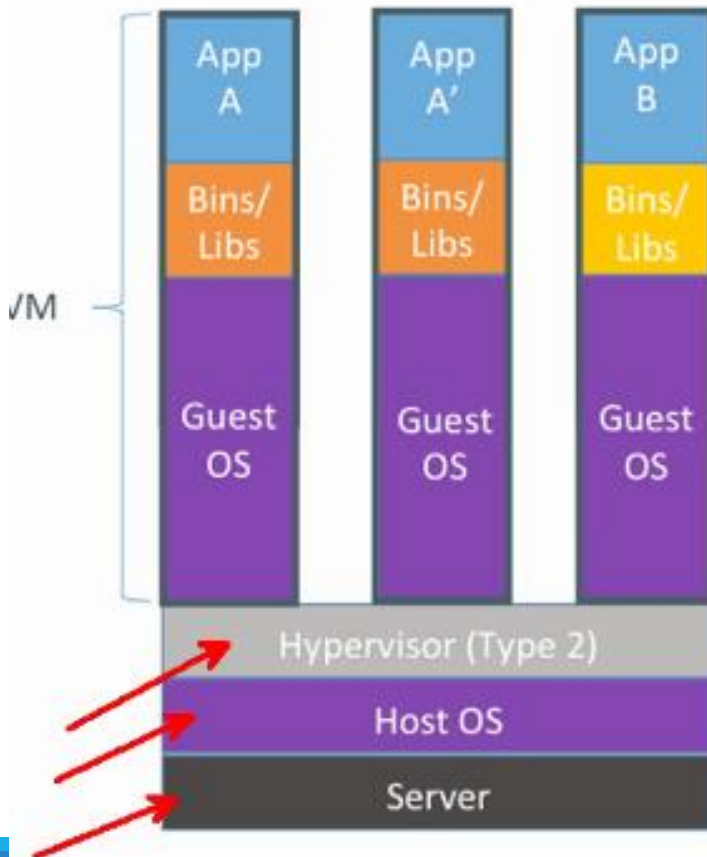
---

Uma imagem é estruturada por camadas, podemos obter uma imagem base Ubuntu por exemplo e adicionar o Apache transformando esta junção em uma outra imagem

Quando criamos uma imagem podemos definir algumas configurações de ambiente, diretórios e etc. Podemos copiar nossa aplicação para dentro dela e distribuí-la

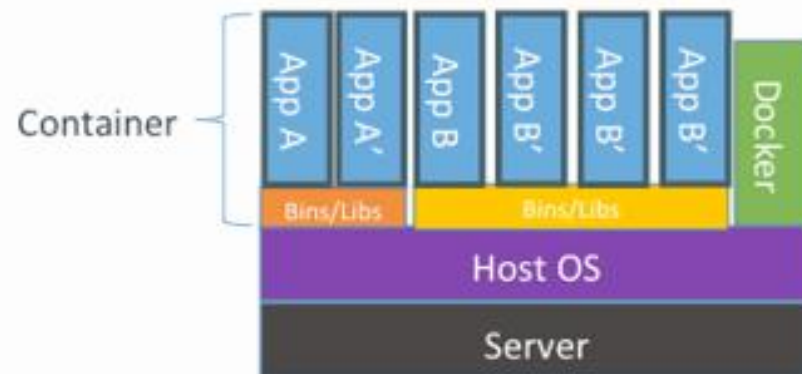
Uma imagem possui camadas somente leitura e uma camada que permite gravação (esta camada é a final e permite que um container modifique seus arquivos e diretórios)

# Básico sobre Docker - Containers



Containers are isolated, but share OS and, where appropriate, bins/libraries

...result is significantly faster deployment, much less overhead, easier migration, faster restart



# Básico sobre Docker

---

Uma imagem está para uma classe, assim como um container está para a instância desta classe

Um container é tratado como uma imagem em execução, pode ser modificado porém ele não persiste as informações quando é parado. O conceito de um container é “stateless”

Você pode criar uma imagem e compartilhar com qualquer pessoa, é o mesmo que fornecer a aplicação e o ambiente configurado (assim como numa VM)

# Instalar Docker no Windows

---

DEMO

# Possíveis problemas

---

- É necessário habilitar o suporte a virtualização na BIOS
- Procure ter o ambiente 100% atualizado
- Docker não funciona bem em uma VM no Azure