



# Usando AppCDS e Jlink para gerar imagens mais eficientes do Java para Containers



# SERGIO LOPES

Especialista no Banco Itaú S/A

@ sergiolopessp@gmail.com

https://twitter.com/sergiolopessp

https://www.linkedin.com/in/sergio-lopes-20131a33/

https://deviniciative.wordpress.com/

# Agenda

- Entendendo o Class Data Sharing (CDS)
- Entendendo o Application Class Data Sharing (AppCDS)
- Como usar o CDS e AppCDS
- Entendo Jdeps e Jlink para gerar uma Runtime
- Juntando tudo para gerar Runtimes mais eficientes para sua aplicação
- Pontos de Atenção

# Entendendo CDS e AppCDS

- CDS – Existe desde a JDK 8 (1.8), mas precisa gerar.
- JDK 10 (JEP 310) – Visando melhorar o startup e o footprint das aplicações, veio o AppCDS (Application CDS)
- JDK 12 (JEP 341) – Default CDS Archives – A JDK passa a gerar o CDS usando um lista de classes default.
- JDK 13 (JEP 350) – Dynamic CDS Archives – Extensão para dinamicamente adicionar as classes que não estão no default

# Hands On – Como funciona o CDS

Acima do JDK 13 já é ligado por Default

```
sergiolopes@sergiolopes-Nitro-AN515-51:~/Documentos/dev/demoCDS/target/classes$ time java demoCDS
Olá TDC, estamos aprendendo sobre AppCDS

real    0m0,082s
user    0m0,054s
sys     0m0,027s
```

Mas você pode desligar

```
sergiolopes@sergiolopes-Nitro-AN515-51:~/Documentos/dev/demoCDS/target/classes$ time java -Xshare:off demoCDS
Olá TDC, estamos aprendendo sobre AppCDS

real    0m0,115s
user    0m0,094s
sys     0m0,021s
```

# Trabalhando com o Application CDS

< JDK13

Cria a lista de classes da aplicação

```
$ java
  -XX:DumpLoadedClassList=classes.lst
  -jar app.jar
```

Cria o Arquivo do AppCDS

```
$ java
  -Xshare:dump
  -XX:SharedClassListFile=classes.lst
  -XX:SharedArchiveFile=app-cds.jsa
  --class-path app.jar
```

> JDK13

Cria o arquivo na saída da JVM

```
$ java
  -XX:ArchiveClassesAtExit=app-cds.jsa
  -jar app.jar
```

Agora é só usar o arquivo do AppCDS

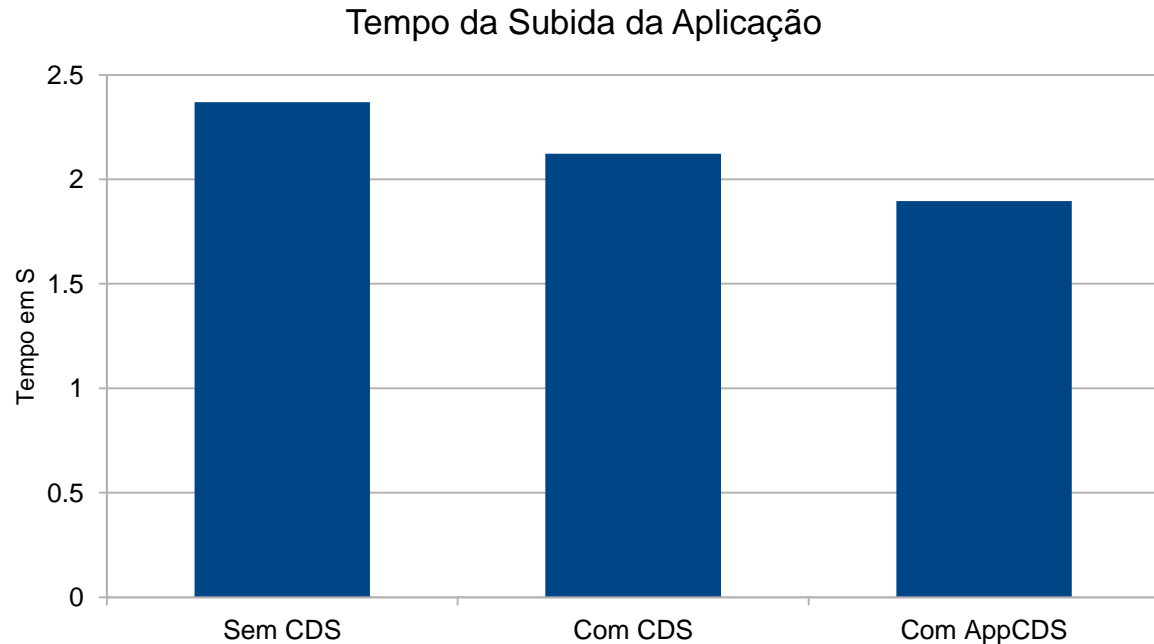
```
$ java
  -XX:SharedArchiveFile=app-cds.jsa
  -jar app.jar
```

# Hands On -Usando na sua aplicação



# Resultado do Exemplo

Aplicação Spring Boot rodando  
com a JDK 15.0.2\*



\* Tempos podem variar conforme o hardware e warmups.



# Melhorando a Imagem da Runtime



1º Passo: Jdeps – comando para analisar as dependências e módulos das aplicações

2º Passo: Jlink – Ferramenta para criar uma runtime customizada da JDK a partir das dependências necessárias

# Hands On – Juntando tudo na Prática



# Pontos de Atenção



Jdeps – Não é recursivo, é preciso entrar nas dependências da aplicação para extrair os módulos.

AppCDS – É específico da JDK que você está rodando, então precisa ser gerado para aquela imagem.

Ainda tem bastante pedágios na JVM.

É um processo que tem muito espaço de melhoria, mas da pra resolver e automatizar muita coisa.



Obrigado Pessoal!