

VESTÍCIOS DE GUERRA

Igor Duarte



Minha Experiência Com Docker & OpenShift

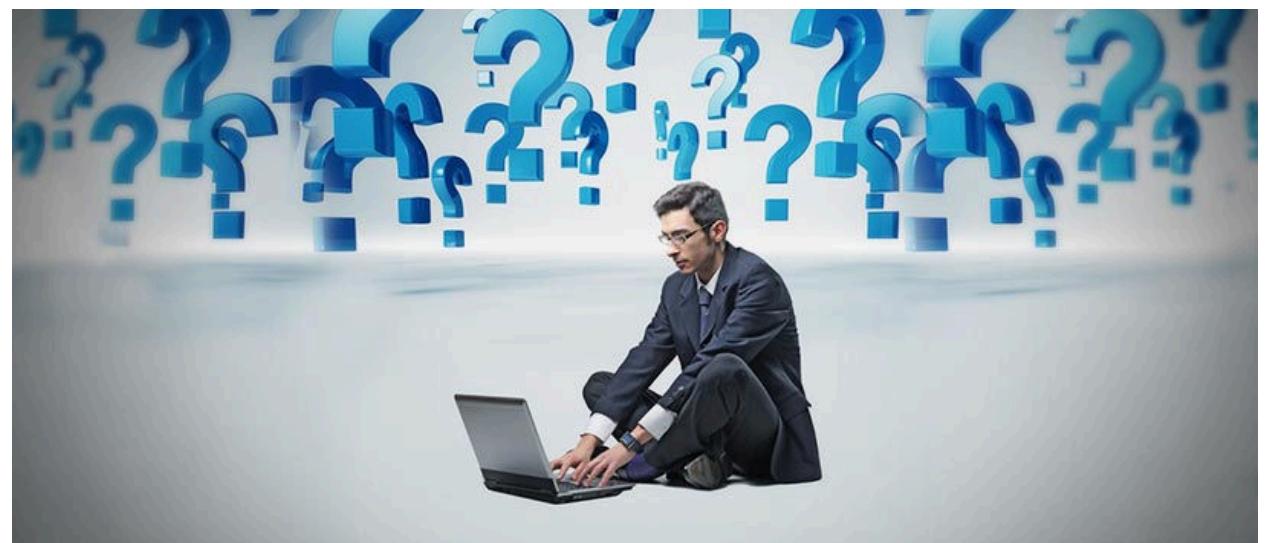
Caso de Uso

Disponibilizar Aplicação desenvolvida com JavaScript, chamada de Prontuário CAF para o usuário final, através do OpenShift.

Solução implementada no grupo RBS.

Como ...

1. Um pouco sobre nossa aplicação
2. Entendendo o tal do Docker
3. Publicando Container e a batalha com o OpenShift
4. Uma Nova batalha chamada MySQL
5. Estatísticas para Nerds



Aplicação Prontuário CAF



Stack



Aplicação desenvolvida para consultar histórico médico do EMED, através de uma API construída com Node.

Tecnologias

- HTML5
- CSS3
- Bootstrap
- JavaScript
- JQuery
- AngularJS
- Node.js
- Express.js
- Gulp
- Bower
- MySQL BD

Pré-Requisitos

- ✓ Projeto já no GitLab
- ✓ Chave SSH já criada e configurada
- ✓ Dump SQL do Banco de Dados



Gerando Chave no Git

No GitBash, executar o comando para gerar as chaves:

```
$ ssh-keygen -t rsa -b 4096 -C seu-email@seu-servidor.com.br
```

No diretório C:\Users\SEU_USER\.ssh
irá gerar os arquivos:

- id_rsa [chave privada]
- id_rsa.pub [chave pública]



Pegar o conteúdo do arquivo id_rsa.pub (chave pública) e colar no GitLab para adicionar a chave.

Profile Settings

Profile Account Applications Chat Access Tokens Emails Notifications **SSH Keys** Preferences Audit Log

SSH Keys

SSH keys allow you to establish a secure connection between your computer and GitLab.

Add an SSH key

Before you can add an SSH key you need to generate it.

Key

Don't paste the private part of the SSH key. Paste the public part, which is usually contained in the file '~/.ssh/id_rsa.pub' and begins with 'ssh-rsa'.

Title

Add key

Your SSH keys (2)

created a week ago

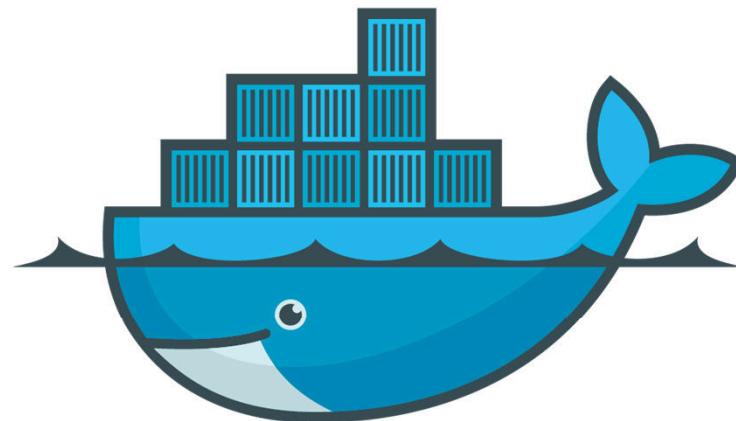
Git Key created a month ago



GitLab

Embarcando aplicação Node.js em um Container





docker

Docker

Neste caso de uso não usaremos o Dockerfile!

Usaremos uma imagem pronta do próprio OpenShift para Node.js
Veremos essa abordagem mais a frente.



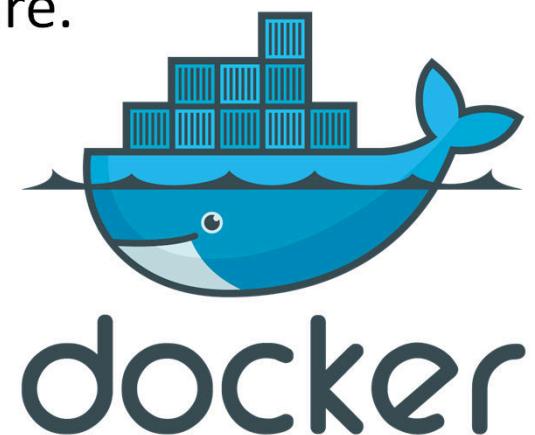
Sim! Um Pouco de Conceitos Entendendo o tal do Docker



O que é Docker?

Docker é uma ferramenta para criar e manter containers, sendo responsável por armazenar vários serviços de forma isolada do SO, como: aplicação, web services, banco de dados, etc..

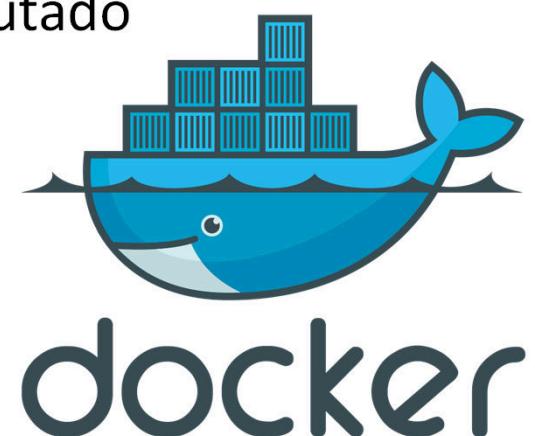
O Docker é uma alternativa para virtualização completa e leve se comparado a outras ferramentas como o VMware.

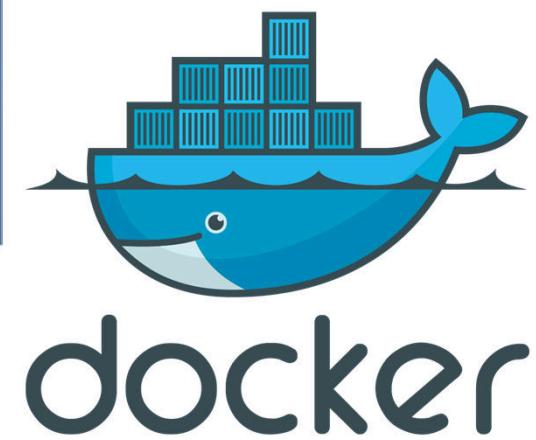
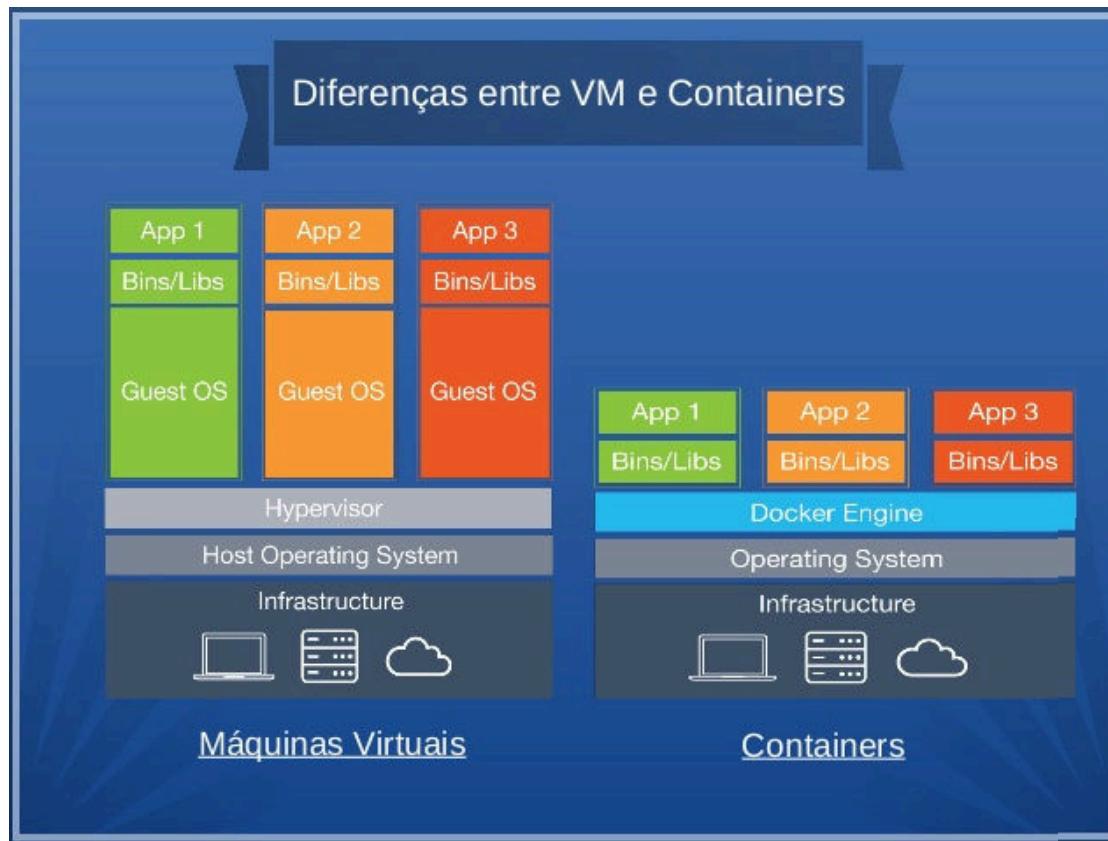


VM x Containers

VM: o software que proporciona a virtualização é instalado diretamente sobre o hardware. Esse tipo de virtualização proporciona um isolamento maior e, ao mesmo tempo, uma sobrecarga, pois cada máquina virtual que é criada executará seu próprio kernel e instância do SO.

Container: o software que proporciona a virtualização é executado sobre um sistema operacional. A virtualização por container ocorre de forma menos isolada, pois compartilha algumas partes do kernel do host, fazendo assim com que a sobrecarga seja menor.





Container

O Container é uma instância de uma Imagem em execução naquele momento.



Publicando Container com Node Embarcado



Publicando no OpenShift

0. Instalando e Configurando OC
1. Criando o Projeto
2. Visualizando App
3. Habilitando Acesso ao GitLab
4. Criando Rota
5. Novos Build
6. Resolvendo Erros
7. Finalmente, Funcionou!!!



OC (Origin Client)

Antes de tudo, vamos baixar e configurar o OC!!



Instalando OC

Download

<https://github.com/openshift/origin/releases/tag/v1.4.1>

➤ **openshift-origin-client-tools-v1.4.1-3f9807a-windows.zip**

Instalado em: C:\oc\

Downloads

CHECKSUM	749 Bytes
openshift-origin-client-tools-v1.4.1-3f9807a-linux-32bit.tar.gz	20.4 MB
openshift-origin-client-tools-v1.4.1-3f9807a-linux-64bit.tar.gz	22.4 MB
openshift-origin-client-tools-v1.4.1-3f9807a-mac.zip	21.3 MB
openshift-origin-client-tools-v1.4.1-3f9807a-windows.zip	21.5 MB
openshift-origin-server-v1.4.1-3f9807a-linux-64bit.tar.gz	66 MB
Source code (zip)	
Source code (tar.gz)	



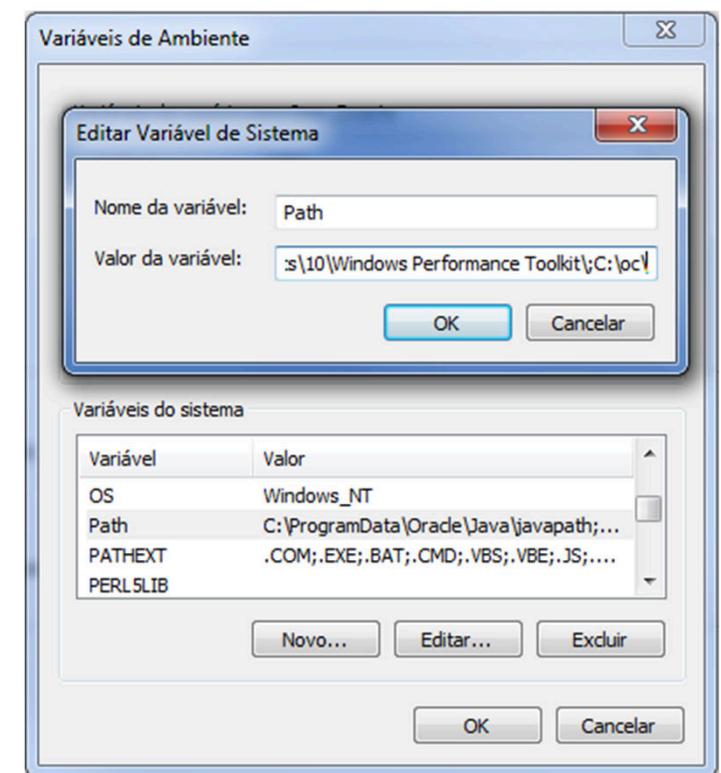
Instalando OC

1 – Acessando via path

\$ cd c:\oc\ (Diretório onde está o oc)

2 – Configurando Variável de ambiente

Acessível direto no terminal



Acessando OC

```
$ oc login https://seu.dominio.com.br:8443
```

```
$ oc project project-name
```

```
$ oc status
```

```
C:\Users\igor.duarte
$ oc login https://openshift.seu.dominio.com.br:8443
Authentication required for https://openshift.seu.dominio.con.br:8443 (openshift)
Username:|
```

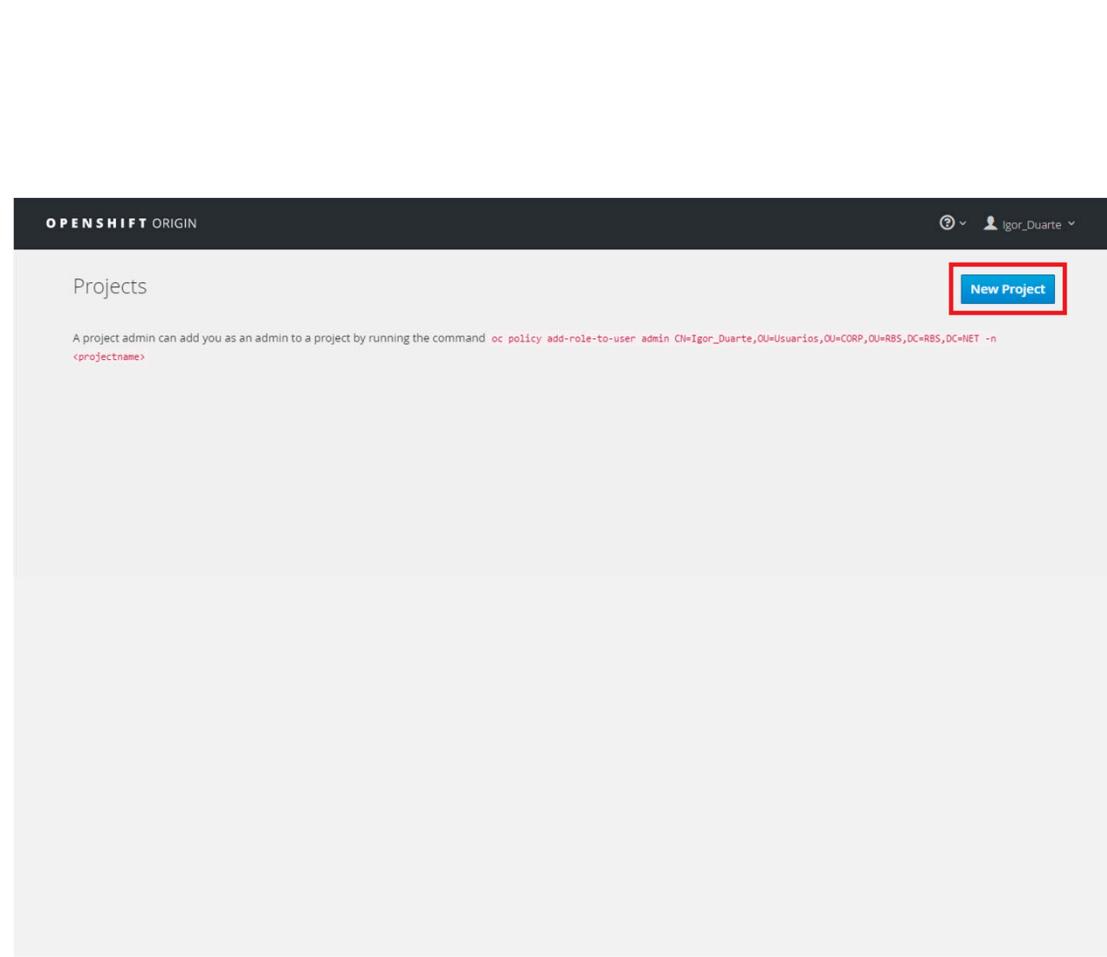


Publicando no OpenShift

1 – Criando o Projeto



The image shows the OpenShift Origin login interface. It features a dark background with a faint watermark of a person wearing a red hat. At the top, there's a header with the text "OPENSHIFT ORIGIN". Below it is a form with fields for "Username" and "Password", each accompanied by a redacted input field. A blue "Log In" button is positioned at the bottom right of the form. To the right of the form, a message reads "Welcome to OpenShift Origin.".



The image shows the OpenShift Origin project management interface. The top navigation bar includes the "OPENSHIFT ORIGIN" logo, a user icon labeled "Igor_Duarte", and a "New Project" button highlighted with a red border. The main content area is titled "Projects" and contains a message for project admins about adding users. At the bottom of the page, there's a footer with the text "A project admin can add you as an admin to a project by running the command `oc policy add-role-to-user admin CN=Igor_Duarte,OU=Usuarios,OU=CORP,OU=RBS,DC=RBS,DC=NET -n <projectname>`".



Publicando no OpenShift

1 – Criando o Projeto....

OPENSHIFT ORIGIN

New Project

* Name
historico-emed

A unique name for the project.

Display Name
historico-emed

Description
A short description.

Create **Cancel**



OPENSHIFT ORIGIN

NodeJS

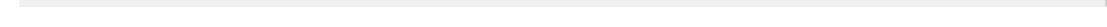
- dancer-mysql-example
QUICKSTART PERL DANCER MYSQL
- nodejs:0.10
BUILDER NODEJS
- nodejs:latest** BUILDER NODEJS
- nodejs-example
QUICKSTART NODEJS
- nodejs-mongodb-example
QUICKSTART NODEJS MONGODB

Other

- logging-deployer-account-template
INFRASTRUCTURE
- logging-deployer-template
INFRASTRUCTURE
- metrics-deployer-template
INFRASTRUCTURE

[Don't see the image you are looking for?](#)

[See all](#)



Publicando no OpenShift

2 – Visualizando App

The screenshot shows the OpenShift Origin web interface. On the left, there's a sidebar with 'Projects' and a 'New Project' button. A message indicates that a project admin can add another user as an admin via a specific command. The main area displays the 'historico-emed' project, which has a service named 'historico-emed-historico-emed.paas.rbs.com.br' on port 8080/TCP. A build log entry shows an error where build #1 failed. The right side of the screen shows a detailed view of the project, including sections for Overview, Details, and a sidebar with 'Overview', 'Browse', and 'Settings' options.



Falhou, mas tudo bem! Problema: Acesso ao GitLab!

Publicando no OpenShift

3 – Habilitando Acesso ao GitLab

1. Criar Secret e anexá-lo ao ServiceAccount
2. Vincular Secret ao BuildConfig



Publicando no OpenShift

3.1 – Criando Secret e Anexar ao Service Account

Configurando Git

➤ Criando chave no git

1. `$ oc secret new historico-emed-secret ssh-privatekey=/users/Igor_Duarte/.ssh/id_rsa`
2. `$ oc secrets add serviceaccount/builder secrets/historico-emed-secret`

```
C:\Users\igor.duarte
λ oc secret new historico-emed-secret ssh-privatekey=/users/igor.duarte/.ssh/id_rsa
secret/historico-emed-secret

C:\Users\igor.duarte
λ oc secrets add serviceaccount/builder secrets/historico-emed-secret
```



Falhou, novamente! Problema: Faltou vincular a Secret ao BuildConfig

Publicando no OpenShift

3.2 – Vinculando Secret ao BuildConfig

```
$ oc edit bc/historico-emed
```

➤ Adicionar linha:

```
sourceSecret:  
  name: historico-emed-secret
```

➤ Editar URL do gitlab:

```
git@gitlab.seu.dominio.com.br:seu-grupo/ProntuarioCAF.git
```

Para: ssh:\git@gitlab.seu.dominio.com.br/seu-grupo/ProntuarioCAF.git

```
27   source:  
28     git:  
29       ref: master  
30       uri: ssh://git@gitlab.seu.dominio.com.br/seu-grupo/ProntuarioCAF.git  
31     sourceSecret:  
32       name: historico-emed-secret  
33       type: Git  
34   strategy:  
35     sourceStrategy:  
36       from:  
37         kind: ImageStreamTag  
38         name: nodejs:latest  
39         namespace: openshift  
40         type: Source  
41     triggers:
```



Publicando no OpenShift

4 – Criando Rota

Browse > Routes > Create Route

OPENSIFT ORIGIN

historico-emed > Create Route

Create Route



Routing is a way to make your application publicly visible. Select a service to expose and enter route details.

* Service: historico-emed

Service to route to.

* Name: historico-emed

A unique name for the route within the project.

Hostname: historico-emed.paas.rbs.com.br

Public hostname for the route. If not specified, a hostname is generated.

Path: /

Path that the router watches to route traffic to the service.

Target Port: 8080 → 8080 (TCP)

Target port for traffic.

Show options for secured routes

Create | Cancel



Publicando no OpenShift

5 – Novo Build ... Novamente

```
$ oc start-build historico-emed --follow
```



Screenshot of the OpenShift web interface showing the 'historico-emed' project overview. The deployment 'HISTORICO-EMED' is shown with 1 pod. The container image is 'historico-emed/historico-emed' (96ac123) and the build is '#6'. The service 'historico-emed' is listed with IP 'historico-emed.paas.rbs.com.br' and port '8080/TCP → 8080'. A note indicates a recent update from '2 hours ago from image change'.



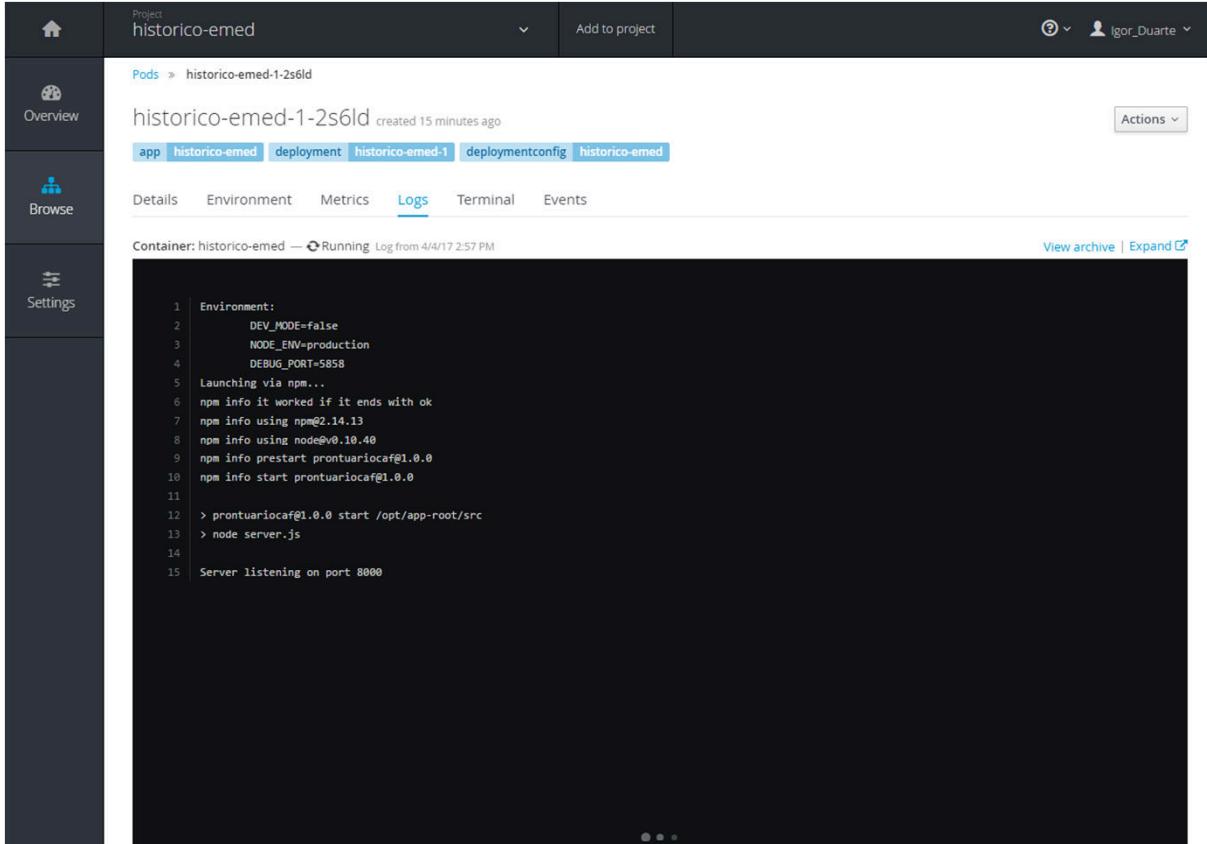
Build com sucesso!
Porém á um Problema: **503 Service Unavailable**

Publicando no OpenShift

6 – Encarando um Novo Erro!!

503 Service Unavailable

No server is available to handle this request.



The screenshot shows the OpenShift web interface for the project "historico-emed". A pod named "historico-emed-1-2s6ld" was created 15 minutes ago. The "Logs" tab is selected, displaying the following application output:

```
1 Environment:
2   DEV_MODE=false
3   NODE_ENV=production
4   DEBUG_PORT=5858
5 Launching via npm...
6 npm info it worked if it ends with ok
7 npm info using npm@2.14.13
8 npm info using node@v0.10.40
9 npm info prestart prontuariocaf@1.0.0
10 npm info start prontuariocaf@1.0.0
11
12 > prontuariocaf@1.0.0 start /opt/app-root/src
13 > node server.js
14
15 Server listening on port 8000
```



Publicando no OpenShift

6 – Encarando um Novo Erro!!

Editing Route historico-emed

```
1 apiVersion: v1
2 kind: Route
3 metadata:
4   name: historico-emed
5   namespace: historico-emed
6   selfLink: /oapi/v1/namespaces/historico-emed/routes/historico-emed
7   uid: 8a6f69f7-1643-11e7-8cda-001a4a1601c4
8   resourceVersion: '28068055'
9   creationTimestamp: '2017-03-31T18:54:59Z'
10  labels:
11    app: historico-emed
12  annotations:
13    openshift.io/generated-by: OpenShiftWebConsole
14    openshift.io/host.generated: 'true'
15  spec:
16    host: historico-emed.paas.rbs.com.br
17    to:
18      kind: Service
19      name: historico-emed
20    port:
21      targetPort: 8080-tcp
22  status:
23    ingress:
24      -
25        host: historico-emed.paas.rbs.com.br
26        routerName: router
27        conditions:
28          -
29            type: Admitted
30            status: 'True'
31            stTransitionTime: '2017-03-31T19:13:53Z'
```



Editing Service historico-emed

```
1 apiVersion: v1
2 kind: Service
3 metadata:
4   name: historico-emed
5   namespace: historico-emed
6   selfLink: /api/v1/namespaces/historico-emed/services/historico-emed
7   uid: 8a721432-1643-11e7-8cda-001a4a1601c4
8   resourceVersion: '28389622'
9   creationTimestamp: '2017-03-31T18:54:59Z'
10  labels:
11    app: historico-emed
12  annotations:
13    openshift.io/generated-by: OpenShiftWebConsole
14  spec:
15    ports:
16      -
17        name: 8080-tcp
18        protocol: TCP
19        port: 8080
20        targetPort: 8080
21    selector:
22      deploymentconfig: historico-emed
23      port1IP: seu.ip
24      clusterIP: seu.ip
25      type: ClusterIP
26      sessionAffinity: None
27    status:
28      loadBalancer: { }
```

Alterada porta de 8080 para 8000 na Rota e no Serviço

Save

Cancel

Save

Cancel

Package.json

Após executar o build do projeto, temos o postinstall no Package.json, que instalará as dependências necessárias, compilara os arquivos .js e iniciara o servidor node.

```
1  {
2    "name": "prontuariocaf",
3    "version": "1.0.0",
4    "description": "Histórico da antiga base do Sistema Emed",
5    "main": "server.js",
6    "scripts": {
7      "postinstall": "node ./node_modules/bower/bin/bower install && node ./node_modules/gulp/bin/gulp.js build",
8      "start": "node serve.js"
9    },
10   "repository": {
11     "type": "git",
12     "url": "git@gitlab.seu.dominio.com.br:seu-grupo/ProntuarioCAF.git"
13   }
```



Publicando no OpenShift

7 – E Finalmente, Funcionou!!!

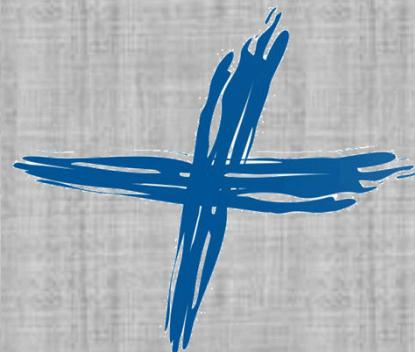


Grupo **RBS**

<https://www.historico-emed.paas.rbs.com.br>

A screenshot of a login interface. It features two input fields: 'Login' with a user icon and 'Senha' with a lock icon. Below the fields is a blue 'Acessar' button with a key icon.

Um Novo Desafio!!!



MySQL no OpenShift

1 – Subindo MySQL no OpenShift

```
$ oc login
```

```
$ oc project histórico-emed
```

```
$ oc new-app mysql MYSQL_USER=root MYSQL_PASSWORD=admin  
MYSQL_DATABASE=emed DATABASE_SERVICE_NAME=mysql -l db=mysql
```



MySQL no OpenShift

1 – Subindo MySQL no OpenShift

```
Administrator: cmd (Admin)
C:\Users\Igor_Duarte (master)

--insecure-registry=false: If true, indicates that the referenced Docker images are on insecure registries and
should bypass certificate checking
-l, --labels='': Label to set in all resources for this application.
--list=false: List all local templates and image streams that can be used to create.
--name='': Set name to use for generated application artifacts
--no-install=false: Do not attempt to run images that describe themselves as being installable
-o, --output='': Output results as yaml or json instead of executing, or use name for succinct output (resource/name).
--param-version='': The preferred API versions of the output objects
-p, --param='{}': Specify a key-value pair (e.g., -p FOO=BAR) to set/override a parameter value in the template.
--search=false: Search all templates, image streams, and Docker images that match the arguments provided.
--strategy='': Specify the build strategy to use if you don't want to detect (docker|source).
--template='': Name of a stored template to use in the app.

Use "oc options" for a list of global command-line options (applies to all commands).

C:\Users\Igor_Duarte (master) ~ oc new-app mysql MYSQL_USER='root' MYSQL_PASSWORD='root' MYSQL_DATABASE='emed' DATABASE_SERVICE_NAME='mysql' -l db=mysql
--> Found image 597621b (9 months old) in image stream "openshift/mysql" under tag "5.6" for "mysql"

MySQL 5.6
-----
MySQL is a multi-user, multi-threaded SQL database server

Tags: database, mysql, mysql56, rh-mysql56

* This image will be deployed in deployment config "mysql"
* Port 3306/tcp will be load balanced by service "mysql"
* Other containers can access this service through the hostname "mysql"
* This image declares volumes and will default to use non-persistent, host-local storage.
  You can add persistent volumes later by running 'volume dc/mysql --add ...'

--> Creating resources with label db=mysql ...
deploymentconfig "mysql" created
service "mysql" created
--> Success
Run 'oc status' to view your app.

C:\Users\Igor_Duarte (master)
```

Project: historico-emed-igor

Overview

SERVICE : HISTORICO-EMED
historico-emed.paas.rbs.com.br 8000/TCP → 8000

DEPLOYMENT: HISTORICO-EMED, #3 a day ago from image change

CONTAINER: HISTORICO-EMED

- Image: historico-emed/historico-emed (a803f34) 194.6 MiB
- Build: historico-emed, #6
- Source: Update README.md (f3f7017)
- Ports: 8080/TCP

SERVICE mysql 3306/TCP → 3306 Create Route

DEPLOYMENT: MYSQL, #1 17 minutes ago from config change

CONTAINER: MYSQL

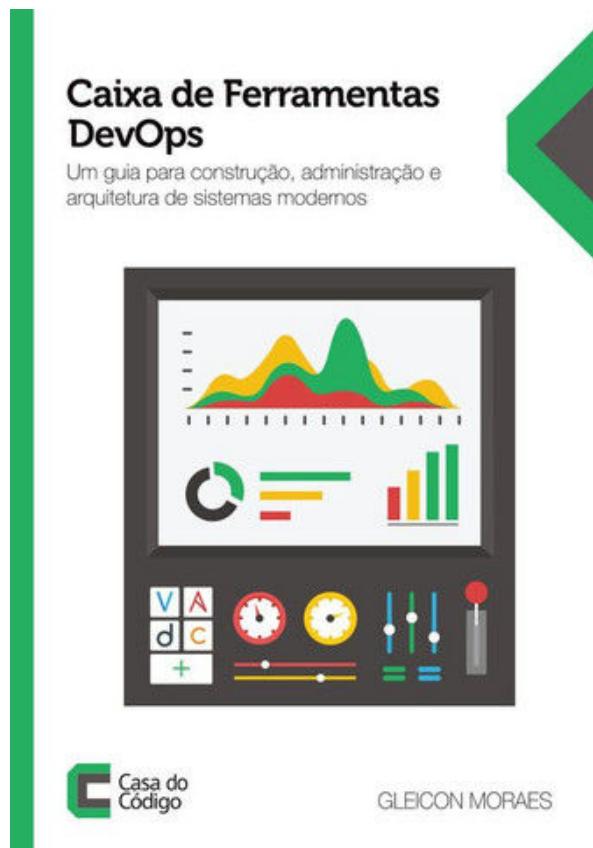
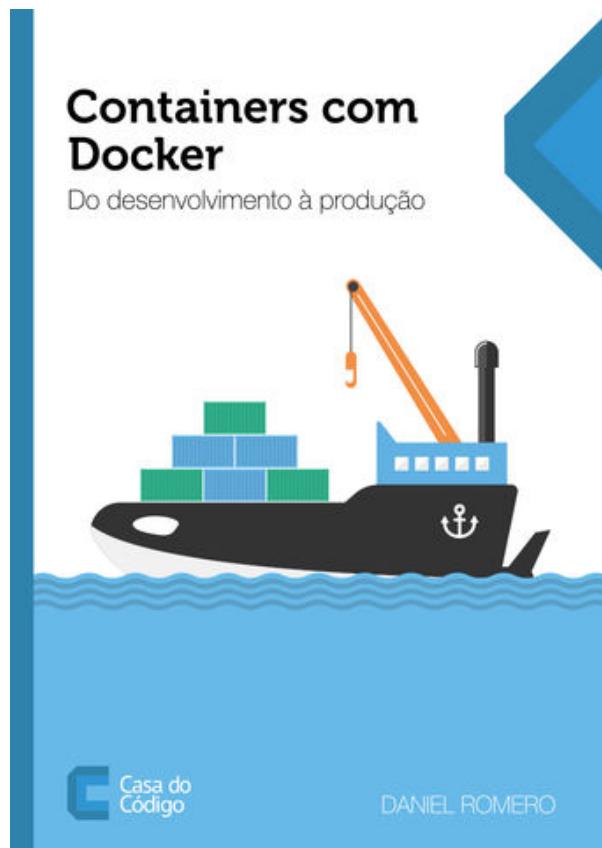
- Image: centos/mysql-56-centos7
- Ports: 3306/TCP



Me Ajudou Um Pouco

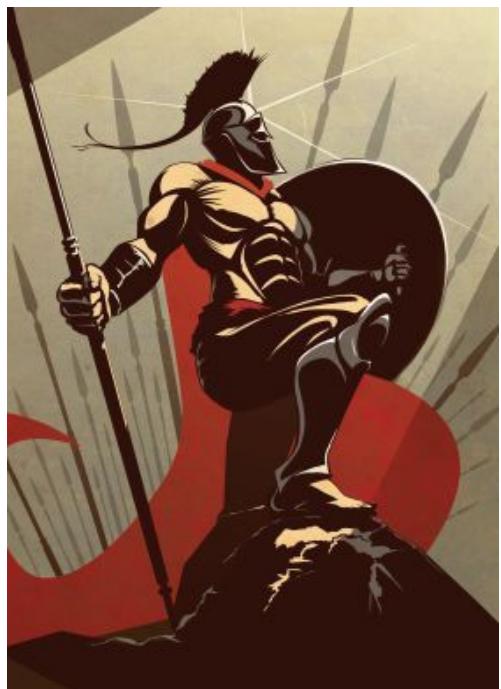
O'REILLY®

➤ <http://www.mundodocker.com.br/>



Steven Pousty & Katie J. Miller

Obrigado!



This is...

Grupo **RBS**

CONTATO



igor.duarte@gruporbs.com.br



linkedin.com/in/igorsantoos



github.com/IgorSantos17

Estatísticas para Nerds



Curiosidade! Quem usa o Node.js



DOW JONES



intuit

The New York Times

YAHOO!



NETFLIX



Kingfisher

Fonte: <https://siftery.com/nodejs>

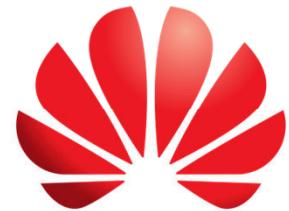


Curiosidade! Quem usa o OpenShift

Cerca de 1.400 empresas usam o OpenShift



THE UNIVERSITY
of NORTH CAROLINA
at CHAPEL HILL



HUAWEI

T-Systems.

ThoughtWorks®



Pioneer



redhat®

KeyBank



EdLogics

Wharton
UNIVERSITY OF PENNSYLVANIA



GM FINANCIAL

A.I.X Capital

ca technologies

FICO™

CISCO™



Fonte: <https://www.openshift.com/container-platform/customers.html>

