

# Trabalho Fase 4 | Containerization Strategy

## 1. URLs usadas nas aulas

Openshift 1 (blog + banco de dados + hpa): <https://console-openshift-console.apps.na46a.prod.ole.redhat.com/topology/ns/rm343626/graph>

Openshift 2 (CI/CD com Github): <https://console-openshift-console.apps.na46a.prod.ole.redhat.com/topology/ns/rm343626-express-ts/graph>

Blog: <http://blog-django-py-git-rm343626.apps.na46a.prod.ole.redhat.com>

Imagem (Quay.io): <https://quay.io/repository/zarantonello/do180-custom-httpd?tab=history>

Github: <https://github.com/zarantonello/express-ts>

Snyk: <https://app.snyk.io/org/zarantonello/>

## 2. Console criada no ambiente do Openshift

Delete

Stop

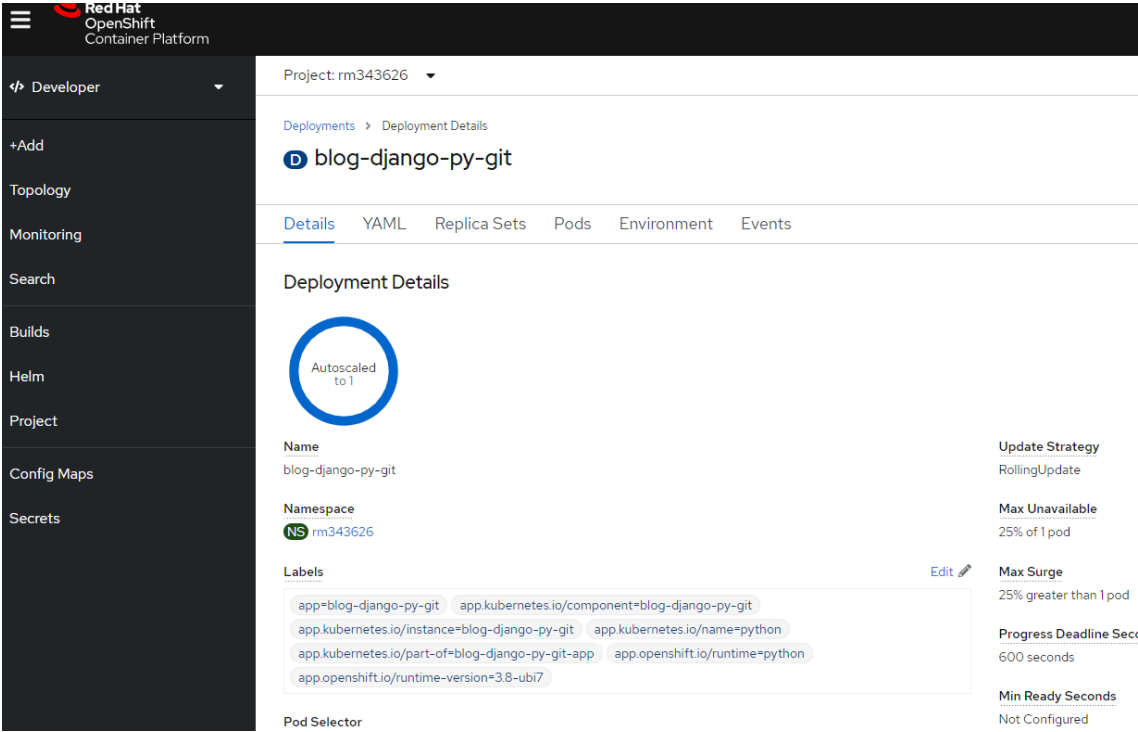
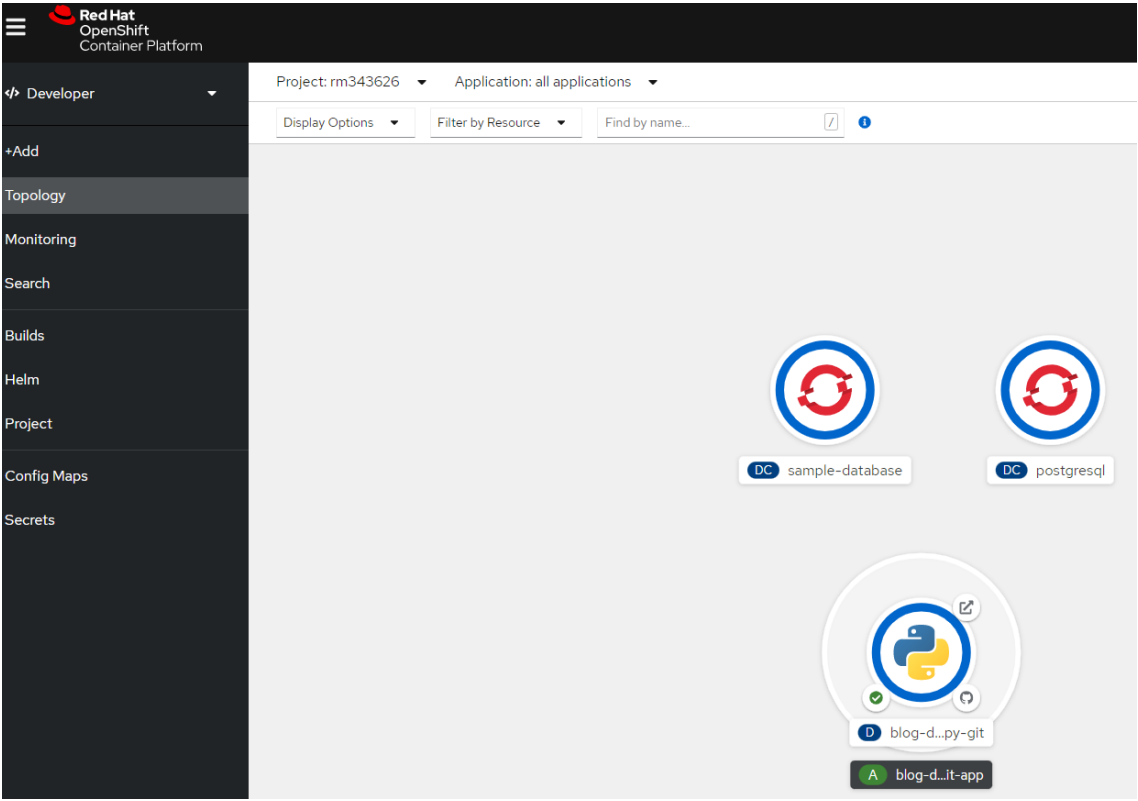
Username	RHT_OCP4_DEV_USER	
Password	RHT_OCP4_DEV_PASSWORD	
API Endpoint	RHT_OCP4_MASTER_API	<a href="https://api.na46a.prod.ole.redhat.com:6443">https://api.na46a.prod.ole.redhat.com:6443</a>
Console Web Application		<a href="https://console-openshift-console.apps.na46a.prod.ole.redhat.com">https://console-openshift-console.apps.na46a.prod.ole.redhat.com</a>
Cluster Id	19394d51-9c1f-40f9-ae07-9a8bdae032a9	

bastion	active	Action	Open Console
classroom	active	Action	Open Console
workstation	active	Action	Open Console

Auto-stop in an hour.

Auto-destroy in 14 days.

3. Criação/provisionamento do projeto blog-django-py-git no Openshift



[View all](#)

#### 4 Services

Active

1 Hour ▾

Resource	Usage	9:00	9:15	9:30	9:45
CPU	6,61m	40m 20m			
Memory	141,6 MiB	300 MiB 200 MiB 100 MiB			
Filesystem	732 KiB	2 MiB 1 MiB			
Network Transfer	0 Bps in 0 Bps out	30 KBps 20 KBps 10 KBps			
Pod count	5	5			

[View event](#)

There are no ongoing activities.

|| Pause

00:38  Add etho [10:125.1.255/25]

## 4. Criação do banco de dados do blog

The screenshot shows the Red Hat OpenShift Container Platform interface. The left sidebar contains navigation links: Developer, +Add, Topology, Monitoring, Search, Builds, Helm, Project, Config Maps, and Secrets. The main area displays the 'sample-database' resource in the 'Resources' tab. The top bar shows 'Project: rm343626' and 'Application: all applications'. The resource is a Deployment Config (DC) named 'sample-database'. The 'Pods' section shows one pod 'sample-database-1-2pgtf' in a 'Running' state. The 'Builds' section indicates 'No Build Configs found for this resource'. The 'Services' section shows a service named 'sample-database' with 'Service port: postgresql' and 'Pod Port: 5432'.

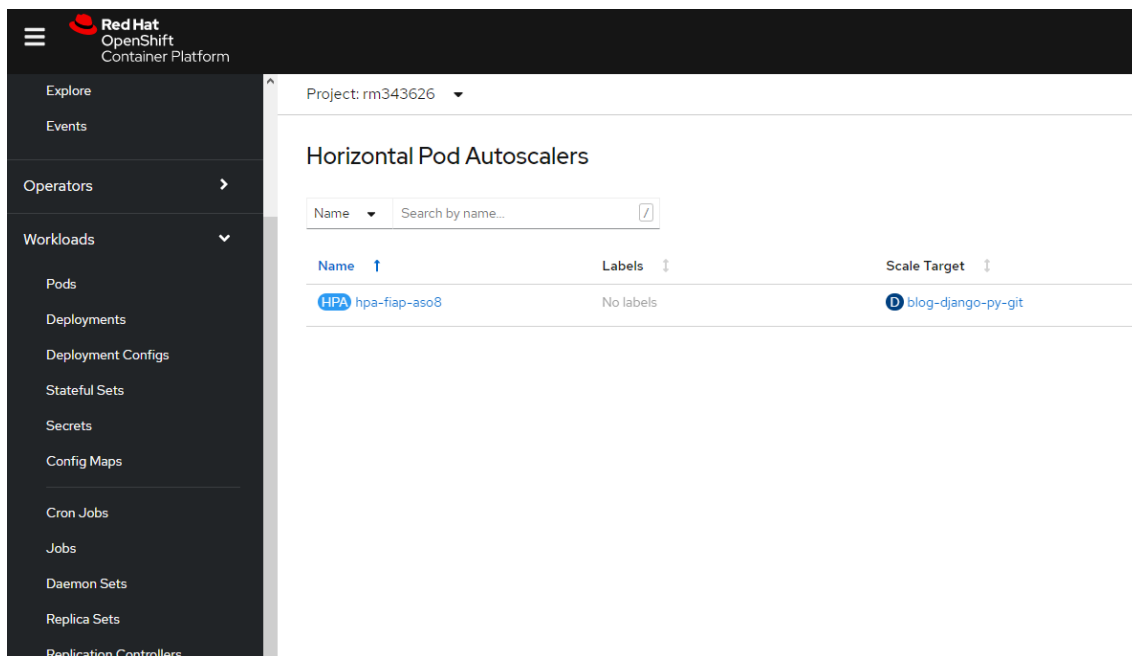
The screenshot shows the 'sample-database' resource in the 'Details' tab. The top bar shows 'Project: rm343626'. The breadcrumb navigation is 'Deployment Configs > Deployment Config Details'. The resource is a Deployment Config (DC) named 'sample-database'. The 'Details' tab is selected, showing 'Deployment Config Details'. A circular diagram indicates '1 pod'. The 'Name' is 'sample-database'. The 'Namespace' is 'rm343626'. The 'Labels' are 'app=sample-database', 'app.kubernetes.io/component=sample-database', 'app.kubernetes.io/instance=sample-database', and 'template=postgresql-persistent-template'. The 'Pod Selector' is 'name=sample-database'. The 'Node Selector' is empty. The 'Latest Version' is '1'. The 'Message' is 'config change'. The 'Update Strategy' is 'Recreate'. The 'Min Ready Seconds' is 'Not Configured'. The 'Triggers' are 'ImageChange, ConfigChange'.

The screenshot shows the 'sample-database' resource in the 'Environment' tab. The top bar shows 'Project: rm343626'. The breadcrumb navigation is 'Deployment Configs > Deployment Config Details'. The resource is a Deployment Config (DC) named 'sample-database'. The 'Environment' tab is selected, showing 'Environment'. The 'Container' is 'postgresql'. The 'Single values (env)' section shows three environment variables: 'POSTGRES\_USER', 'POSTGRES\_PASSWORD', and 'POSTGRES\_DATABASE'. The 'VALUE' column shows the values: 'sample-database/database-user', 'sample-database/database-password', and 'sample-database/database-name'. The 'Add More' and 'Add from Config Map or Secret' buttons are visible at the bottom.

```
C:\Users\wilso\Downloads\openshift-client-windows>oc get pods
```

NAME	READY	STATUS	RESTARTS	AGE
blog-django-py-git-1-build	0/1	Completed	0	3d20h
blog-django-py-git-6b4dfcd97-bxjnd	1/1	Running	0	3d19h
postgresql-1-vcx7g	1/1	Running	0	4d17h
sample-database-1-2pgtf	1/1	Running	0	3d20h
sample-database-1-deploy	0/1	Completed	0	3d20h

## 5. Configuração do HPA para o blog



```
! hpa.yaml x
C: > Users > wilso > OneDrive > Universidade > MBA > Fase 4 > Hands-on Lab > ! hpa.yaml
1  apiVersion: autoscaling/v2beta2
2  kind: HorizontalPodAutoscaler
3  metadata:
4    name: hpa-fiap-aso8
5    namespace: rm343626
6  spec:
7    scaleTargetRef:
8      apiVersion: apps/v1
9      kind: Deployment
10     name: blog-django-py-git
11    minReplicas: 1
12    maxReplicas: 3
13    metrics:
14      - type: Resource
15        resource:
16          name: cpu
17          target:
18            averageUtilization: 50
19            type: Utilization
20
```

```
C:\Users\wilso\Downloads\openshift-client-windows>oc get hpa
NAME          REFERENCE          TARGETS  MINPODS  MAXPODS  REPLICAS  AGE
hpa-fiap-aso8 Deployment/blog-django-py-git  0%/50%   1         3         1         41h
```

## 6. Teste de stress

```
{ } stress-test.json 1 X
C: > Users > wilso > Downloads > openshift-client-windows > { } stress-test.json
1  import http from 'k6/http';
2  import {check, sleep} from 'k6';
3
4  export default function() {
5      let res = http.post('http://blog-django-py-git-rm343626.apps.na46a.prod.ole.redhat.com');
6      check(res, { 'success login': (r) => r.status === 200 });
7      sleep(0.3);
8  }
9
```

```
C:\Users\wilso\Downloads\openshift-client-windows>k6 run --vus 10000 --iterations 10000 stress-test.json
```



```
execution: local
script: stress-test.json
output: -
```

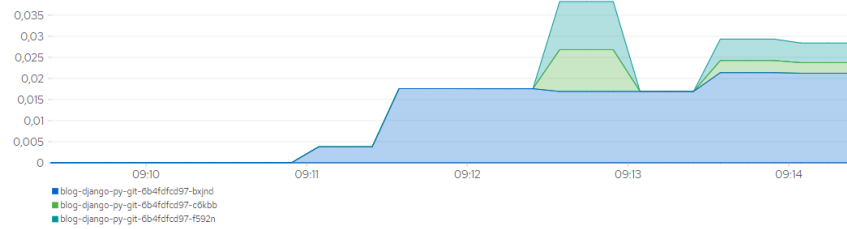
```
scenarios: (100.00%) 1 scenario, 10000 max VUs, 10m30s max duration (incl. graceful stop):
* default: 10000 iterations shared among 10000 VUs (maxDuration: 10m0s, gracefulStop: 30s)
```

```
running (00m04.9s), 09998/10000 VUs, 2 complete and 0 interrupted iterations
```

```
default [-----] 10000 VUs 00m04.9s/10m0s 00002/10000 shared iters
```

```
C:\Users\wilso\Downloads\openshift-client-windows>oc get pods
NAME                                READY   STATUS    RESTARTS   AGE
blog-django-py-git-1-build          0/1     Completed 0           3d20h
blog-django-py-git-6b4fdgcd97-bxjnd 1/1     Running   0           3d20h
blog-django-py-git-6b4fdgcd97-c6kbb 1/1     Running   0           24s
blog-django-py-git-6b4fdgcd97-f592n 1/1     Running   0           24s
postgresql-1-vcx7g                 1/1     Running   0           4d18h
sample-database-1-2pgtf             1/1     Running   0           3d20h
sample-database-1-deploy            0/1     Completed 0           3d20h
```

The screenshot shows the OpenShift Container Platform dashboard. On the left is a sidebar with navigation options: Developer, +Add, Topology, Monitoring, Search, Builds, Helm, Project, Config Maps, and Secrets. The main area displays a project named 'rm343626' with a filter set to 'all applications'. It shows three application icons: 'sample-database', 'postgresql', and 'blog-d.py-git'. The 'blog-d.py-git' application is highlighted with a red box. On the right, a detailed view of the 'blog-django-py-git' application is shown. It includes a 'Health Checks' section with a warning that the container does not have health checks. Below this, the 'Autoscaled to 3' status is highlighted with a red box. Further down, the 'Update Strategy' is set to 'RollingUpdate', and the 'Max Surge' is set to '25% greater than 3 pods'. The 'Labels' section shows 'app=blog-django-py-git'.





## 7. Publicação da imagem customizada no Quay.io

quay.io/repository/zarantonello/do180-custom-httpd

E-mail Finance Study Dev Smart City O que é Big O Nota...

RED HAT Quay.io EXPLORE REPOSITORIES TUTORIAL

⚠ We are currently facing technical issues with support@quay.io email. In order to contact Quay.io customer support, please use t

← Repositories zarantonello / do180-custom-httpd

Repository Activity

No

Description

Click to set repository description

RED HAT Quay.io EXPLORE REPOSITORIES TUTORIAL

search

⚠ We are currently facing technical issues with support@quay.io email. In order to contact Quay.io customer support, please use the email support@coreos.com until further notice.

← Repositories zarantonello / do180-custom-httpd ☆

Repository Tags

1 - 1 of 1

TAG	LAST MODIFIED	SECURITY SCAN	SIZE	EXPIRES
v1.0	5 days ago	110 High - 548 fixable	73.4 MB	Never

RED HAT Quay.io EXPLORE REPOSITORIES TUTORIAL

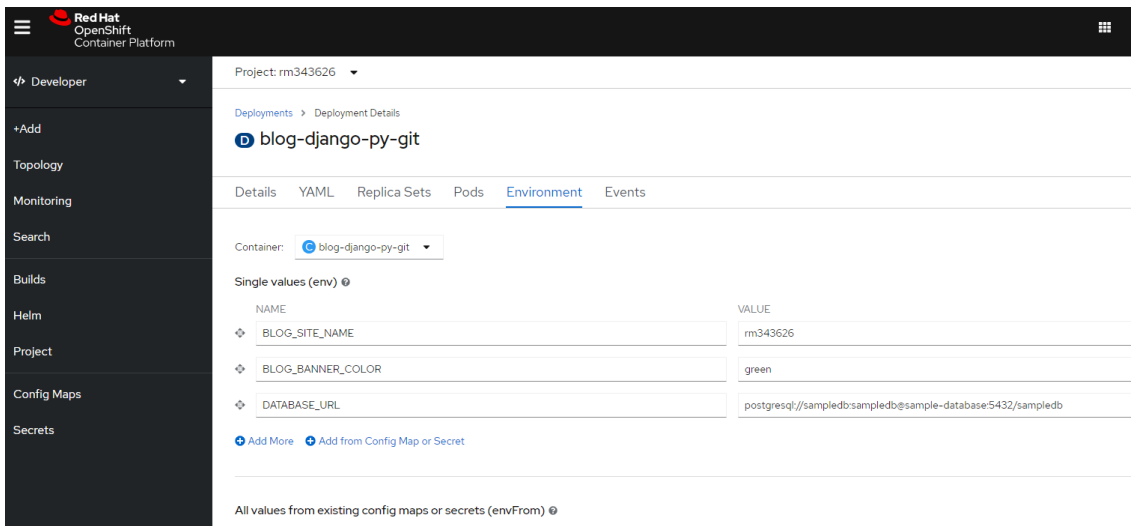
⚠ We are currently facing technical issues with support@quay.io email. In order to contact Quay

← Repositories zarantonello / do180

Tag History

TAG CHANGE	DATE/TIME	REV
Jul 6, 2022		
v1.0 was created pointing to <a href="#">SHA256: 1cec93ad015</a>	Wed, Jul 6, 2022 9:46 AM	

## 8. Customização do blog (nome e cor do site) através das variáveis de ambiente

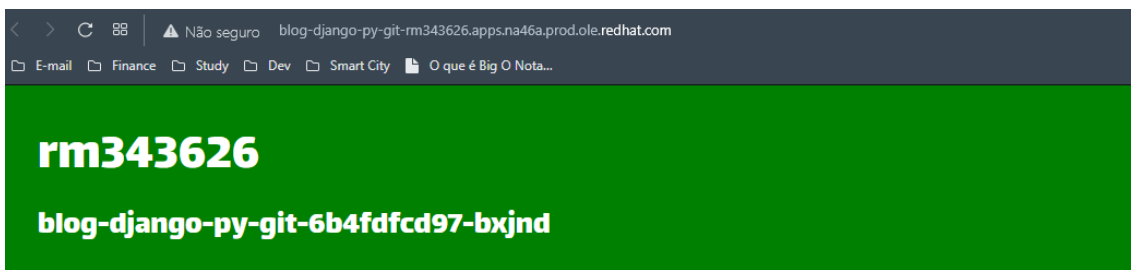


The screenshot shows the Red Hat OpenShift Container Platform console. On the left is a sidebar with navigation options: Developer, +Add, Topology, Monitoring, Search, Builds, Helm, Project, Config Maps, and Secrets. The main area displays the 'Environment' tab for the deployment 'blog-django-py-git'. It lists three environment variables:

NAME	VALUE
BLOG_SITE_NAME	rm343626
BLOG_BANNER_COLOR	green
DATABASE_URL	postgresql://sampledb:sampledb@sample-database-5432/sampledb

Below the table are links for 'Add More' and 'Add from Config Map or Secret'. At the bottom, it says 'All values from existing config maps or secrets (envFrom)'.

## 9. Postagem no blog com conectividade com o banco de dados (<http://blog-django-py-git-rm343626.apps.na46a.prod.ole.redhat.com>)



March 14, 2017, 3:26 a.m.

### What is Red Hat OpenShift?

Red Hat® OpenShift® is an enterprise-ready Kubernetes container platform with full-stack automated operations to manage hybrid cloud, multicloud, and edge deployments. Red Hat OpenShift is optimized to improve developer productivity and promote innovation. Red Hat OpenShift is available as a fully managed cloud service on leading public clouds, or as a self-managed software offering for organizations requiring more customization.

March 14, 2017, 3:27 a.m.

### What is Kubernetes?

Kubernetes is an open-source system for automating deployment, scaling, and management of containerized applications.

July 7, 2022, 3:58 p.m.

**rm343626**

Hello world!

10. Publicação do Projeto express-ts (Github)

zarantonello / express-tsPublic

<> Code

Issues

Pull requests3

Actions

Projects

Wiki

Security

Insights

Settings

main4 branches0 tags

zarantonello docker

.github/workflows

Ajustes

.vs

Ajustes

.vscode

Add files

bin

Add files

manifests

Add files

public

Add files

server

Change title

views

Add files

.dockerignore

Add files

.eslintrc.js

Add files

.gitignore

Add files

Dockerfile

docker

README.md

Add files

copy-assets.js

Add files

11. Deploy do projeto no Openshift

