

H
A
R
B
O
R

What,
Why
and how!

Harbor with Postgres
with AWS Bucket S3





Adonai Costa

@adonaicosta

adonai@getup.io

SRE

Getup

João Brito

@juniorjbn

joao@getup.io

CTO

Getup



GETUP

WE MAKE INFRASTRUCTURE INVISIBLE

O quê?

- CNCF;
- Repositório de imagens;
- Seguro;
- Performático;
- Escalável;
- Modularizado;
- Integrações:
 - RBAC
 - AD/LDAP
 - Vault
 - Audit
 - Notary
 - ChartMuseum (Helm)

Por quê?

- Na faixa;
- Controle de imagem base;
- Alta disponibilidade;
- Escalabilidade;
- Configurações personalizadas;
- Self-Hosted;
- Unlimited rate (Bye Docker).

Por que utilizar o Harbor?



Docker hub deixou de ser gratuito

 **RED HAT**® Quay.io Quay nunca foi gratuita, não é barata e é em USD.



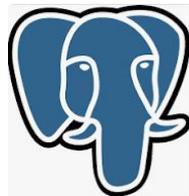
Nossa solução:

Construa seu próprio Registry em seus clusters Kubernetes e tenha a mesma eficiência, maior controle, segurança e níveis de disponibilidade.



HARBOR

&



PostgreSQL

Por que utilizar o Harbor?

 **RED HAT**® Quay.io Quay ou o Hub Docker às vezes estão indisponíveis.

Evite isso:

 **RED HAT**® Quay.io

SUBSCRIBE TO UPDATES

Quay.io image pulls failing

Subscribe

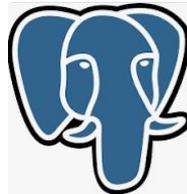
Investigating - Image pulls from quay.io are currently failing with the code "502 Bad Gateway". Engineers are currently investigating.

Mar 22, 16:30 EDT



HARBOR

&



PostgreSQL

Harbor por si só, em , já é resiliente ... é mais gostoso 😊

Harbor is a [Cloud Native Computing Foundation](#) Graduated project



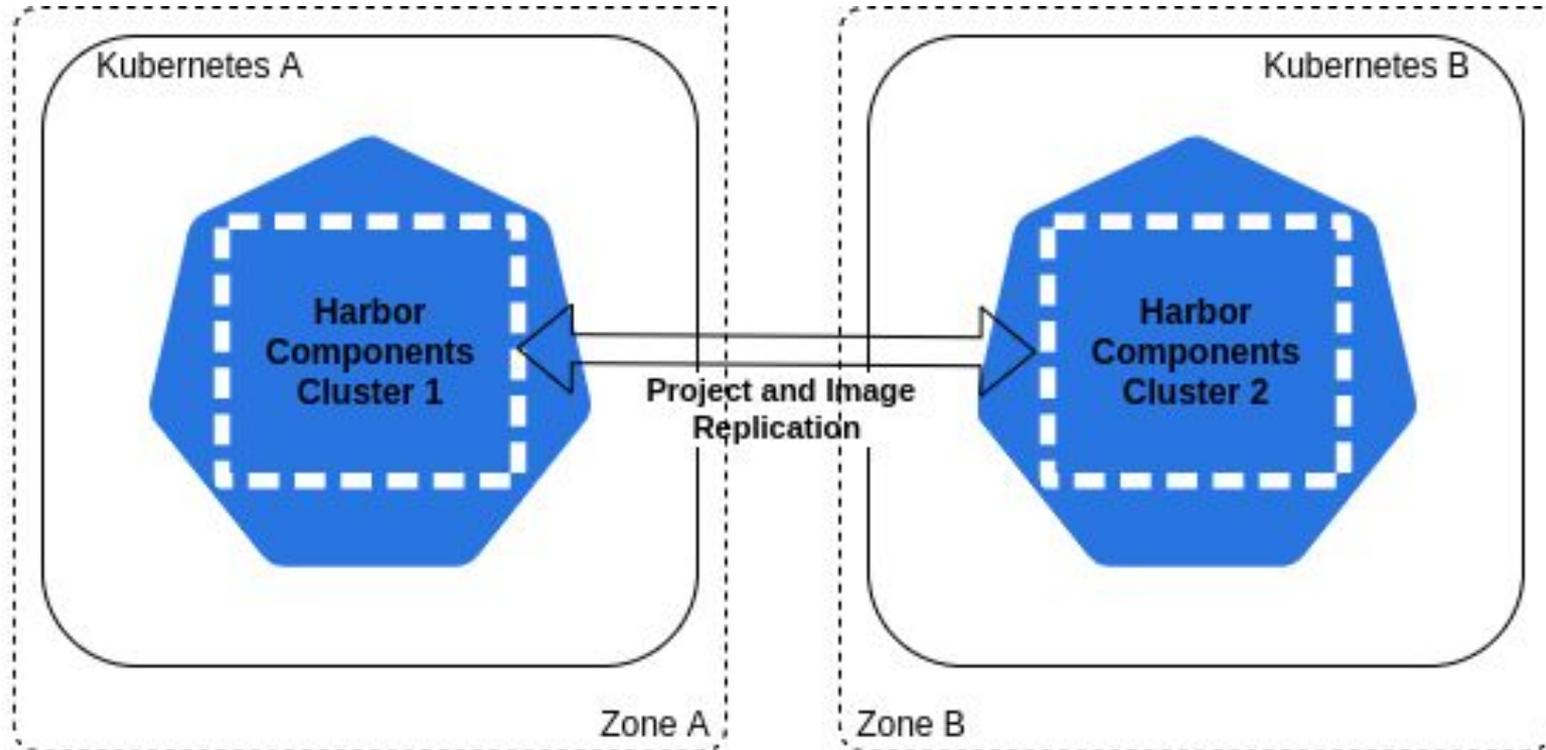
- Pods, por natureza são efêmeros
- Os dados podem ser mantidos em disco (pv)
- Há possibilidade de snapshot/backup desses discos
- Você pode escalar a sua capacidade
 - Em volume
 - Em quantidade de pods

H
A
R
B
O
R

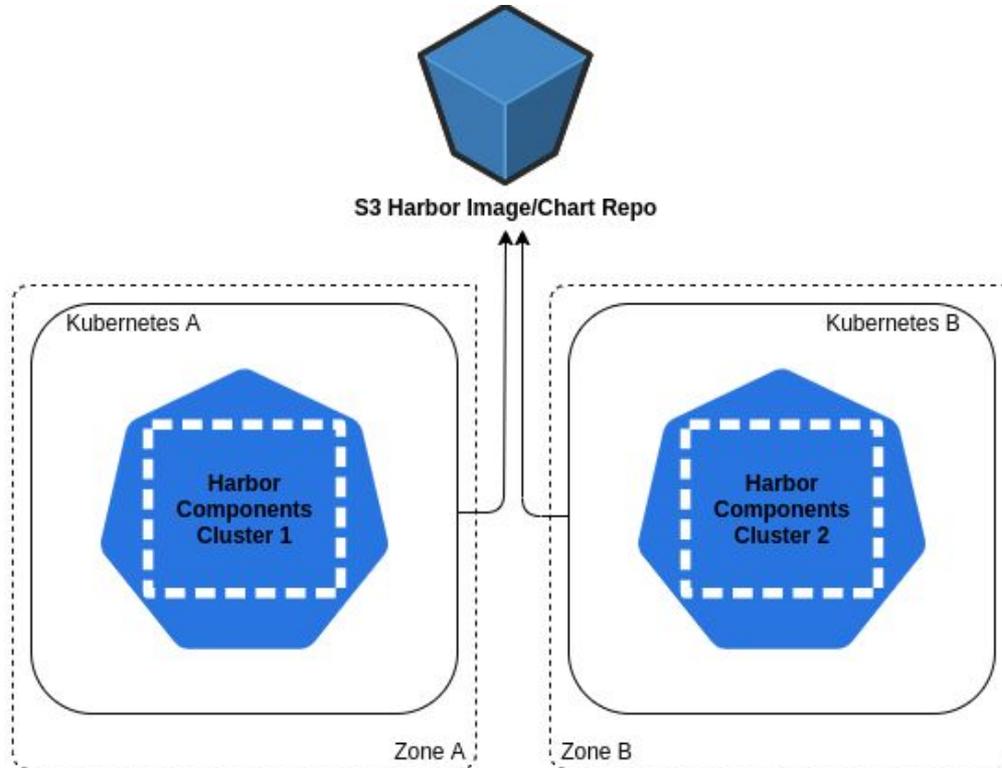
Arquiteturas
&
Resiliência



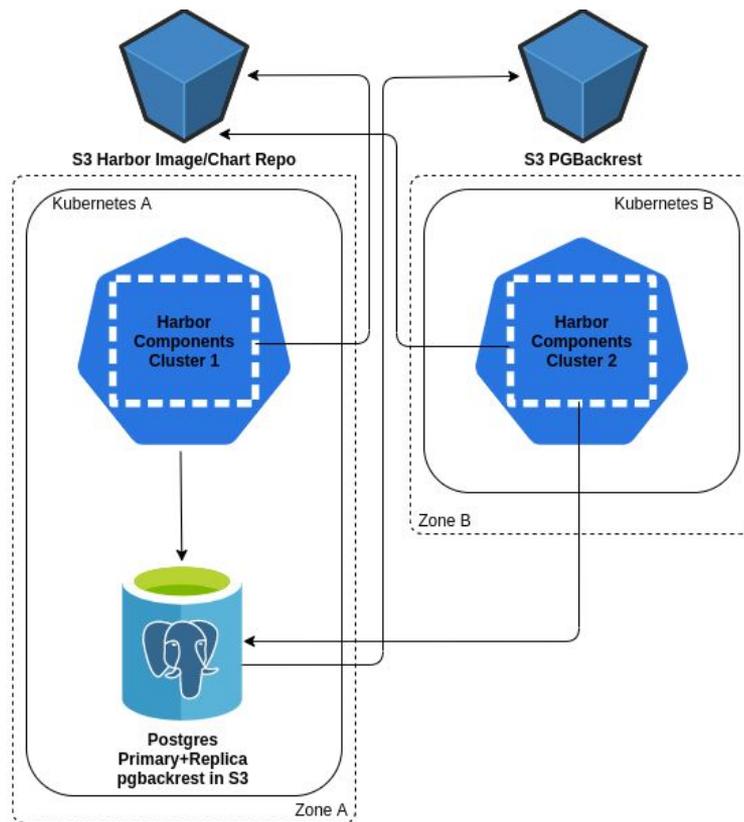
Harbor em HA com Project ou Harbor Replication



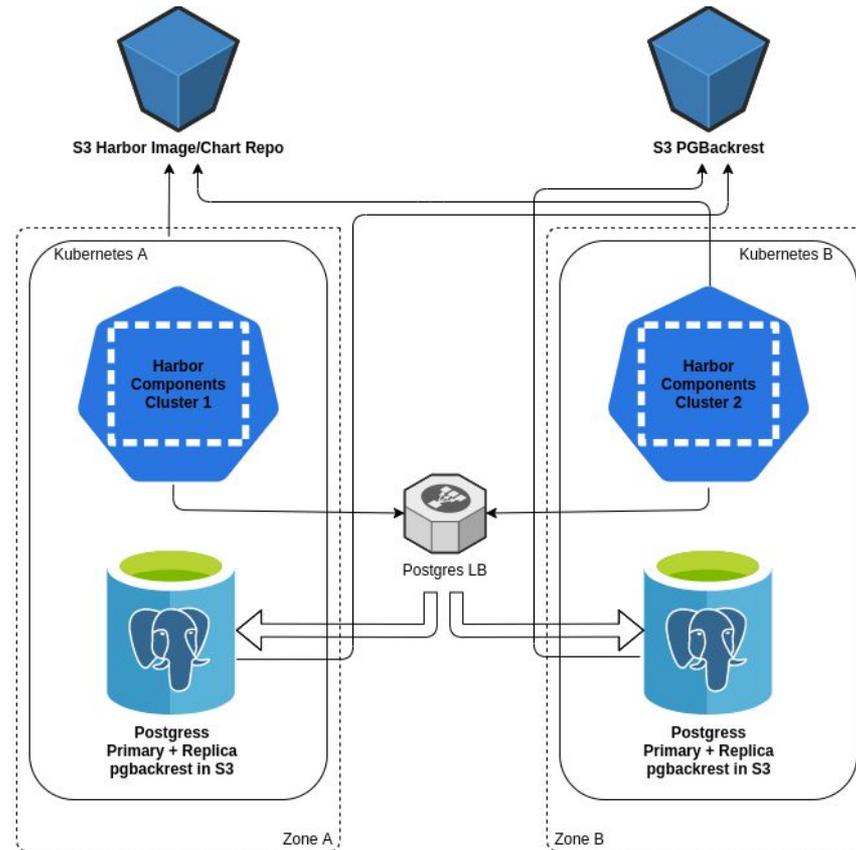
Harbor com Image e Chart Repository em s3



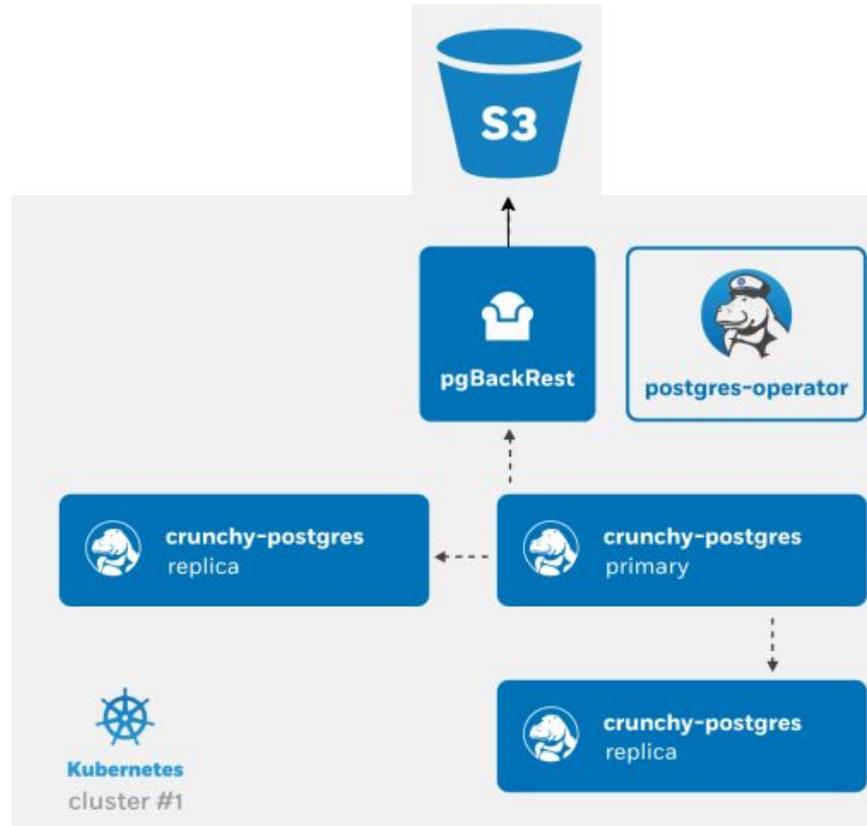
Harbor em HA com Postgres



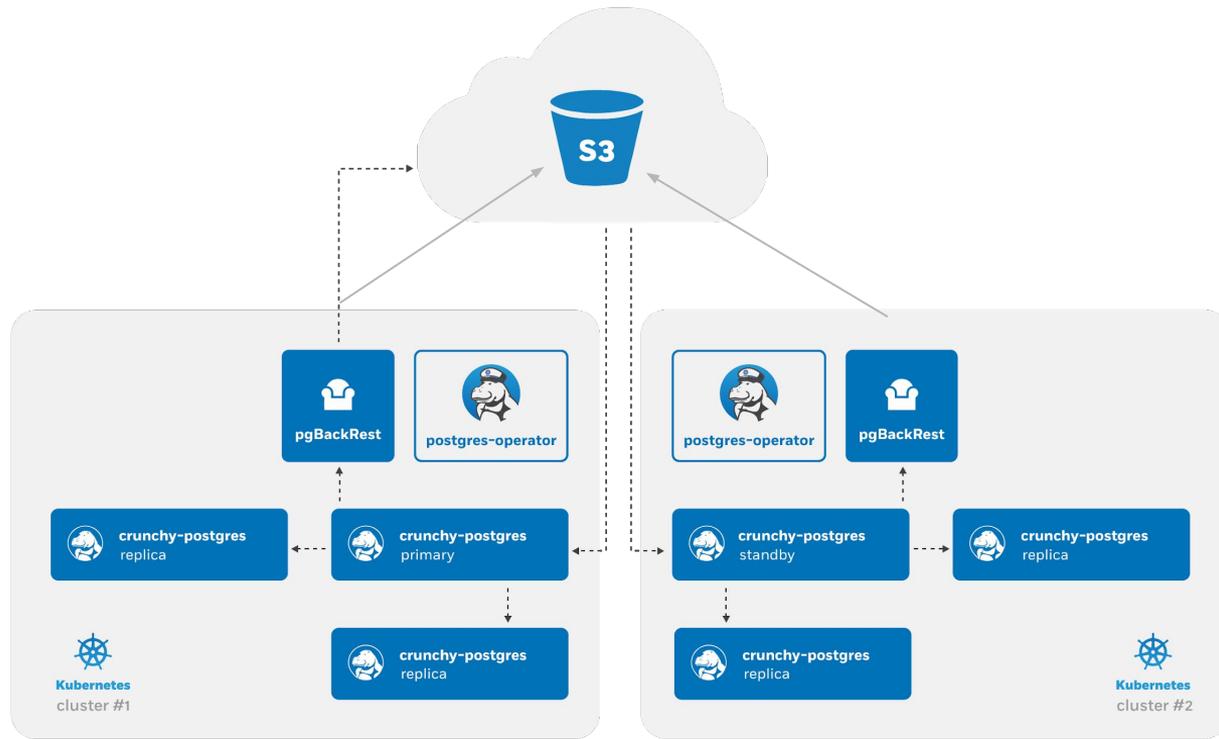
Harbor em HA com Postgres em HA



Topologia de Postgres com réplica



Topologia de um Postgres com réplica – em HA



Criando, no primeiro cluster, a primeira instancia + réplica de Postgres

```
pgo create cluster harbor1
--pgbouncer
--replica-count=1
--pgbackrest-storage-type=local,s3
--pgbackrest-s3-key=$AWS_ACCESS_KEY_ID
--pgbackrest-s3-key-secret=$AWS_SECRET_ACCESS_KEY
--pgbackrest-s3-bucket=pgo-hole
--pgbackrest-s3-endpoint=s3.amazonaws.com
--pgbackrest-s3-region=us-east-1
--password-superuser=harbor
--password-replication=somewhatsecrethippo
--password=harbor
--pvc-size=10Gi
--service-type=LoadBalancer
```

No segundo Cluster, criando a segunda instância Postgres + réplica, baseado na primeira

```
pgo create cluster harbor2
--standby
--pgbouncer
--replica-count=1
--pgbackrest-storage-type=local,s3
--pgbackrest-s3-key=$AWS_ACCESS_KEY_ID
--pgbackrest-s3-key-secret=$AWS_SECRET_ACCESS_KEY
--pgbackrest-s3-bucket=pgo-hole
--pgbackrest-s3-endpoint=s3.amazonaws.com
--pgbackrest-s3-region=us-east-1
--pgbackrest-repo-path=/backrestrepo/harbor1-backrest-shared-repo
--password-superuser=harbor
--password-replication=somewhatsecrethippo
--password=harbor
--pvc-size=10Gi
--service-type=LoadBalancer
```

Exemplo de setup DNS Route53

<input checked="" type="checkbox"/>	harbor-1.getup.io	CNAME	Simple	-	a0ceb4dc0875046c995f047378d30ee3-1
<input checked="" type="checkbox"/>	harbor-2.getup.io	CNAME	Simple	-	a0f480b1740514d00806b9741667cd50-
<input checked="" type="checkbox"/>	notary-1.getup.io	CNAME	Simple		
<input checked="" type="checkbox"/>	notary-2.getup.io	CNAME	Simple		
<input checked="" type="checkbox"/>	pgharbor.getup.io	CNAME	Failover		
<input checked="" type="checkbox"/>	pgharbor.getup.io	CNAME	Failover		

Record name [Info](#)

pgharbor .getup.io

Valid characters: a-z, 0-9, ! " # \$ % & ' () * + , - / : ; < = > ?
@ [\] ^ _ ` { } . ~

TTL (seconds) [Info](#)

Record type [Info](#)

CNAME – Routes traffic to another doma... ▼

Value [Info](#)

ab004ee4ee0d4e5fbb3e67e0-

Enter multiple values on separate lines.

Failover record type

Primary

Record name [Info](#)

pgharbor .getup.io

Valid characters: a-z, 0-9, ! " # \$ % & ' () * + , - / : ; < = > ?
@ [\] ^ _ ` { } . ~

Record type [Info](#)

CNAME – Routes traffic to another doma... ▼

Value [Info](#)

af10e1111b5143d0866122732ed47e

Enter multiple values on separate lines.

Alias

TTL (seconds) [Info](#)

60

Routing policy [Info](#)

Failover ▼

Failover record type

Secondary ▼

1m

1h

1d

Recommended values: 60 to 172800 (two days)

Health check - optional [Info](#)

Choose health check ▼

Record ID [Info](#)

pgsecondary

Choose health check ▲

b2gsecouq97

Terminal... criando os postgres ...



1/1

[harbor1@bastion-br live eks-1.0 k8s-1.18] 03/12/2021 01:44:59 AM ~ \$ k get po

+ kubectl get po

NAME	READY	STATUS	RESTARTS	AGE
backrest-backup-harbor1-2frmb	0/1	Completed	0	6h15m
harbor1-666dc684f8-969x8	1/1	Running	0	6h16m
harbor1-backrest-shared-repo-67966bcd76-p4xqr	1/1	Running	0	6h16m
harbor1-bzgi-6f4ddb9f5-hr82p	1/1	Running	0	6h14m
harbor1-pgbouncer-54db88c7d5-lvpt9	1/1	Running	0	6h15m
harbor1-stanza-create-ff9x9	0/1	Completed	0	6h15m
postgres-operator-689598d795-47nzt	4/4	Running	1	10h

[harbor1@bastion-br live eks-1.0 k8s-1.18] 03/12/2021 01:45:04 AM ~ \$ k get svc

+ kubectl get svc

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
harbor1	LoadBalancer	10.100.75.19	ab894aa4acd0d4a5fbbb3567a0ab488f-1524284695.us-east-1.elb.amazonaws.com	2022:32706/TCP,5432:30005/TCP	6h17m
harbor1-backrest-shared-repo	ClusterIP	10.100.175.220	<none>	2022/TCP	6h17m
harbor1-pgbouncer	ClusterIP	10.100.52.62	<none>	5432/TCP	6h16m
harbor1-replica	LoadBalancer	10.100.136.122	a8b0b6b5d0f0242ffaacb3474d1c3502-203788904.us-east-1.elb.amazonaws.com	2022:31040/TCP,5432:30902/TCP	6h14m
postgres-operator	ClusterIP	10.100.230.174	<none>	8443/TCP,4171/TCP,4150/TCP	10h

[harbor1@bastion-br live eks-1.0 k8s-1.18] 03/12/2021 01:45:25 AM ~ \$

3: ssh br

[harbor2@bastion-br live eks-1.0 k8s-1.18] 03/12/2021 01:43:01 AM ~ \$

2: ssh br

[harbor2@bastion-br enter eks-1.0 k8s-1.18] 03/11/2021 11:12:56 PM ~ \$

Terminal...

- Instalação do Harbor em cada cluster
- Configs Principais
- Get Pods

Browser....

- Replicação
- Criação de repo e sync com o outro Harbor
- Docker login
- DNS load balance



Find us:

- getup.io
- [@getup](https://twitter.com/getup)
- [Kubicast](https://kubicast.com)
- blog.getup.io



GETUP