

12+ factor
databases

@sebaWebber



- Super fã do PostgreSQL
- Aprendiz de ninja
- Gosta dos Golang e Ruby
- Organiza uns evento por aí
- Tende a falar bobagem nos eventos
- Faz uns rango que fica bonito
- Humildão que não toma IPA

ONGRES

Professional Services Products Com

ONGRES

Products and services
to boost your Postgres

Always focused on Postgres, specialized in Infrastructure,
Consulting, Professional Services and R&D.

CONTACT US



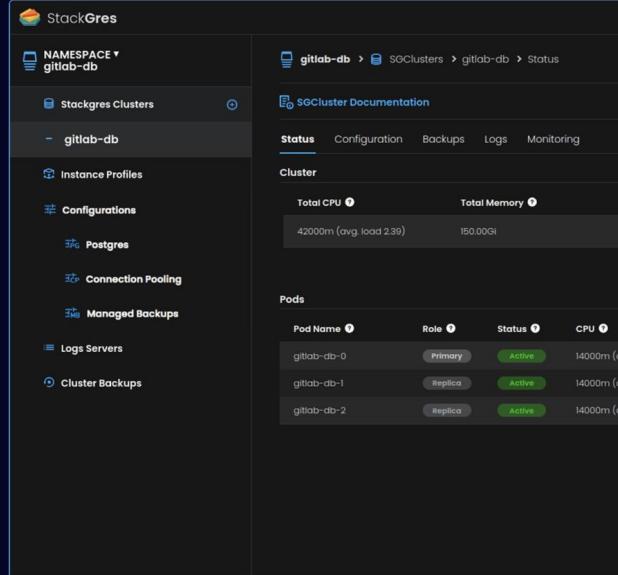
SQL
Ruby
o por aí
em nos
fica bonito
toma IPA



**Enterprise Postgres
made easy.
On Kubernetes**

StackGres is a fully-featured platform for running PostgreSQL on Kubernetes. Deploy in minutes an Enterprise-Grade Postgres-as-a-Service, in your infrastructure. 100% Open Source.

[Learn More](#) [Install →](#)



C.BOX A



Os Doze Fatores

I. Base de Código

Uma base de código com rastreamento utilizando controle de revisão, muitos deploys

II. Dependências

Declare e isole as dependências

III. Configurações

Armazene as configurações no ambiente

IV. Serviços de Apoio

Trate os serviços de apoio, como recursos ligados

V. Construa, lance, execute

Separe estritamente os builds e execute em estágios

VI. Processos

Execute a aplicação como um ou mais processos que não armazenam estado

VII. Vínculo de porta

Exporte serviços por ligação de porta

VIII. Concorrência

Dimensione por um modelo de processo

IX. Descartabilidade

Maximizar a robustez com inicialização e desligamento rápido

X. Dev/prod semelhantes

Mantenha o desenvolvimento, teste, produção o mais semelhante possível

XI. Logs

Trate logs como fluxo de eventos

XII. Processos de Admin

Executar tarefas de administração/gerenciamento como processos pontuais

Minha app

Dentro da minha super app!

Métricas de monitoramento

Gerencia de logs

Gestão de inicialização/vida

Gestão de Configuração

Funcionalidades

Runtime e dependencias

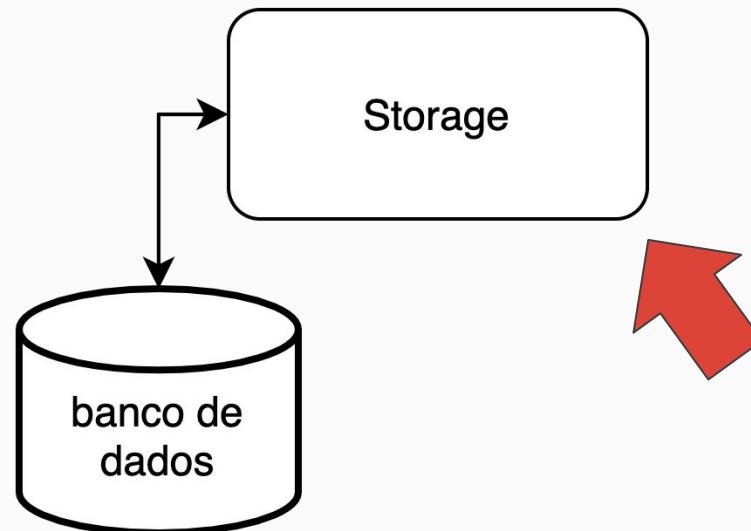
0 - Persistência

On Premises: Storage dedicado
com discos, rede e energia
redundantes tolerantes a falhas

Na cloud: EBS ou similar



WE LOVE STORAGE!

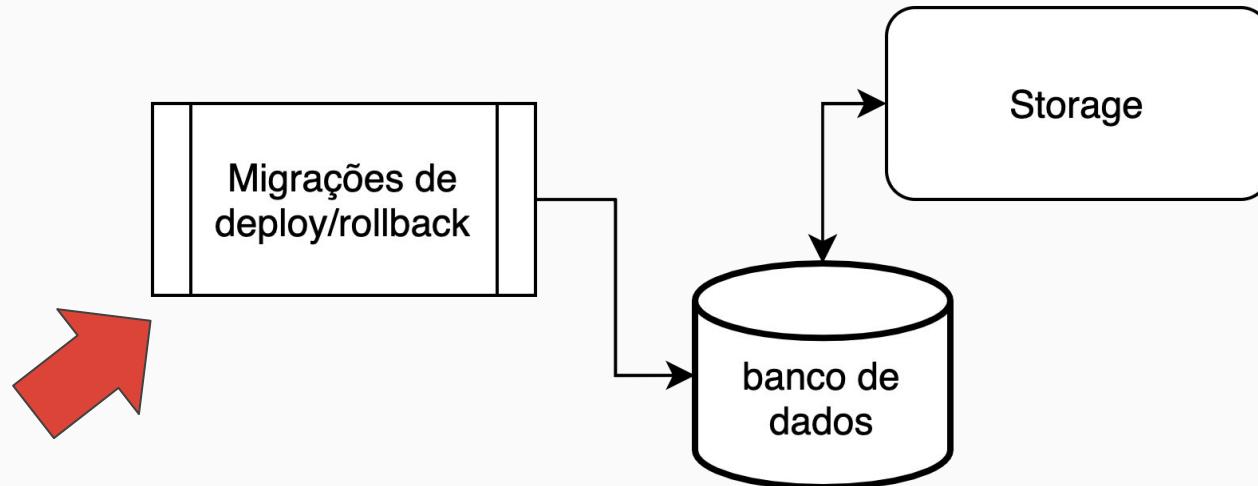


1 - Base de código

GIT MERGE

Automação do deployment ou rollback das mudanças no banco armazenadas no git - também conhecido como migrations.

Capricha no `git log`.

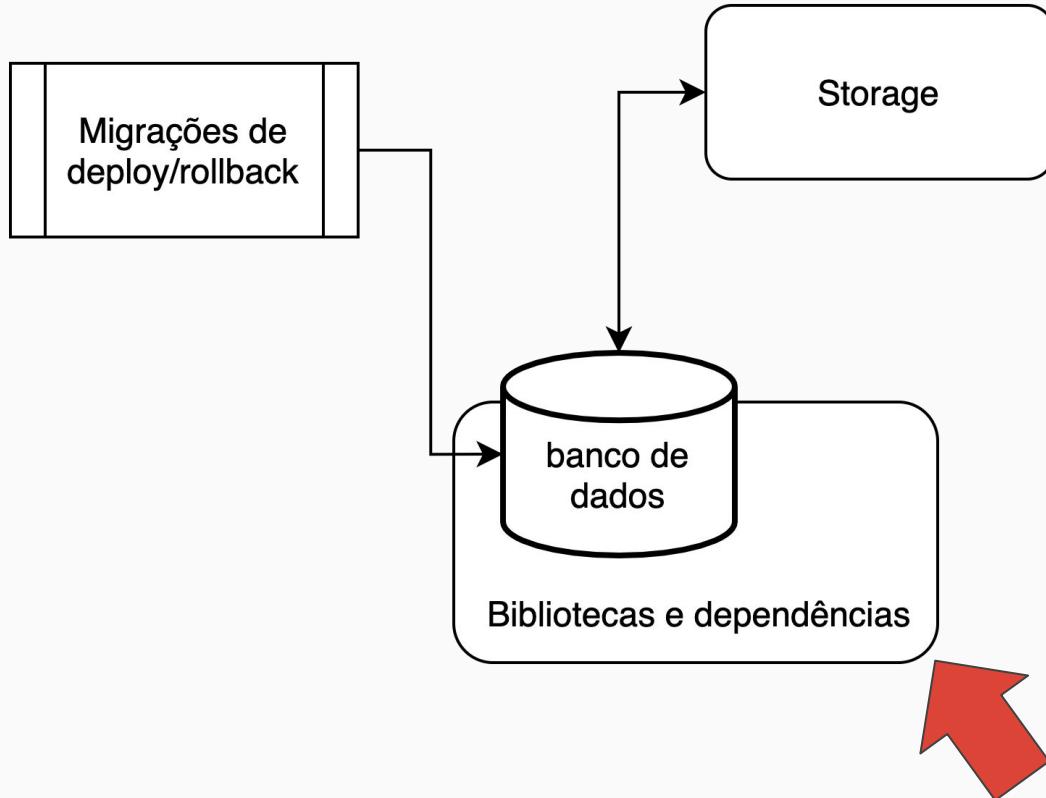


2 - Dependências

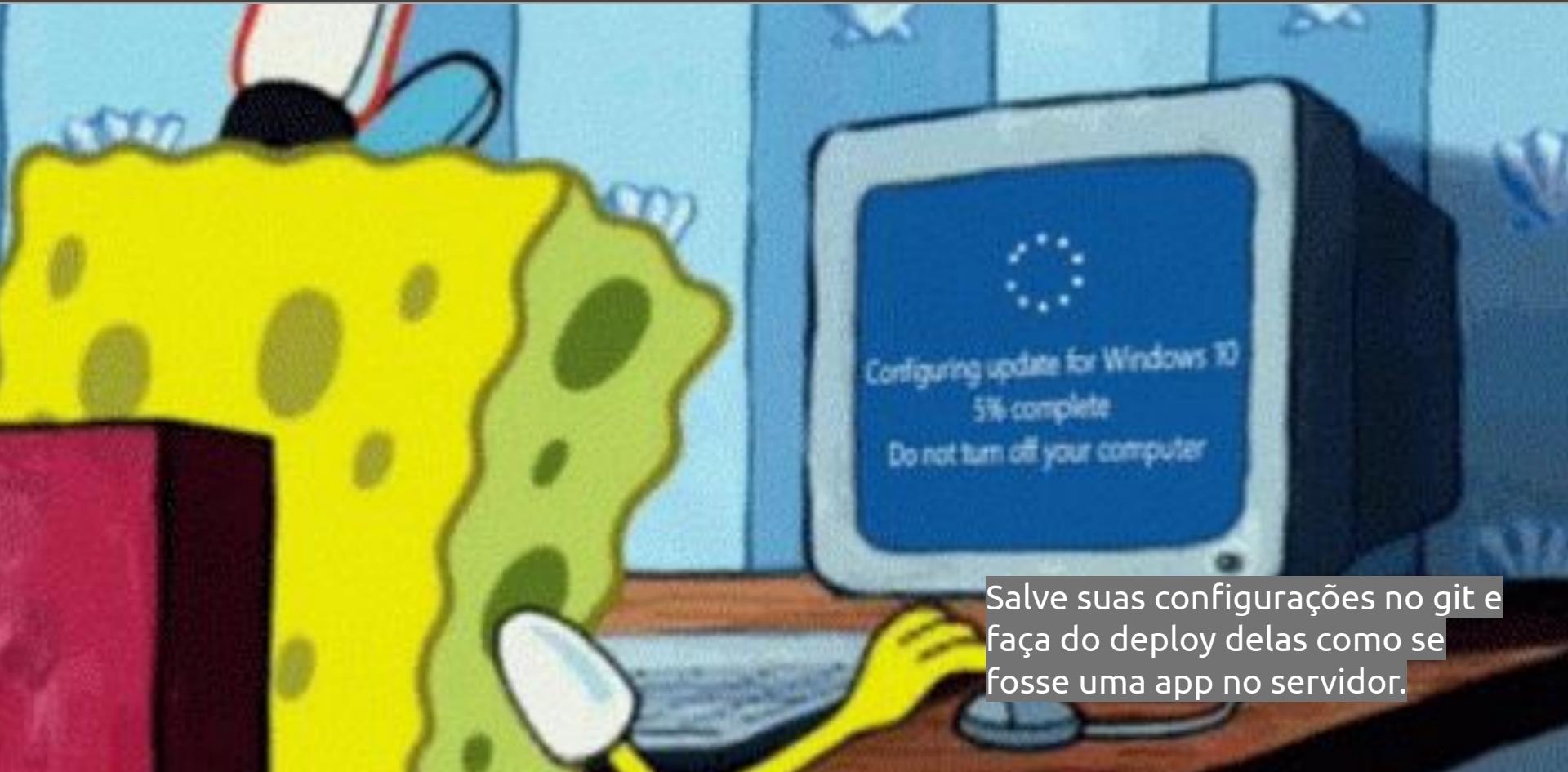
A blurry, motion-blurred image of Spider-Man in his iconic red and blue suit, running towards the right. He is positioned in front of a white computer monitor which displays a dark screen. The background is a dark, indistinct room.

Setup do ambiente que o banco vai rodar seja vm/equipamento físico ou containers.

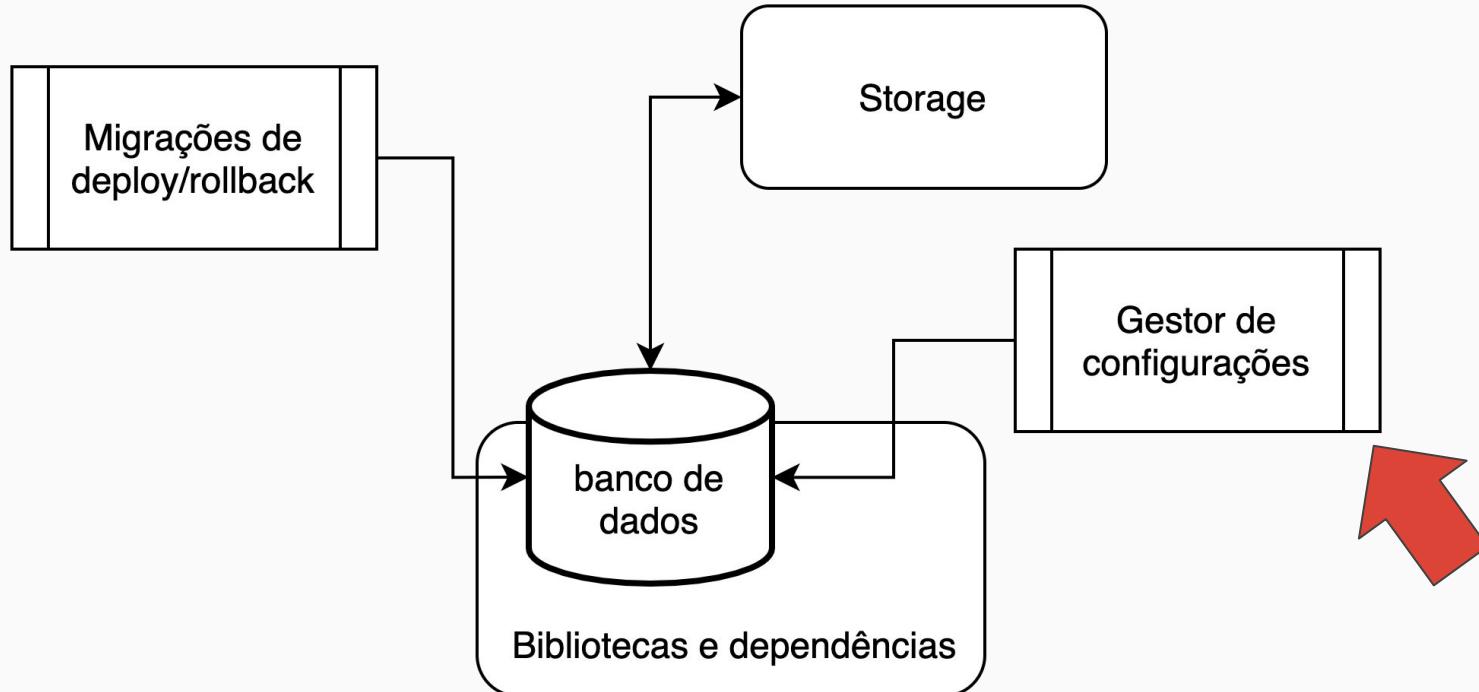
Lembre-se do drama do Alpine e a GLib C.



3 - Configurações



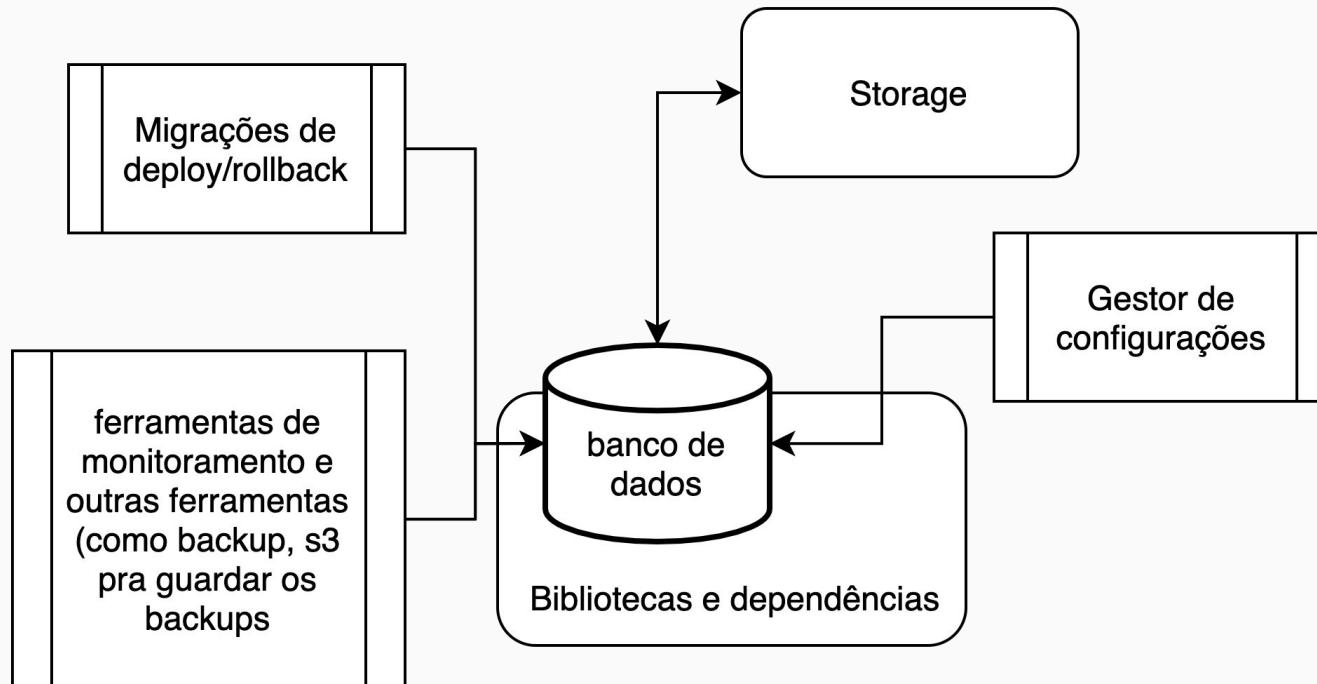
Salve suas configurações no git e
faça do deploy delas como se
fosse uma app no servidor.



4 - Serviços de apoio

A blurry, low-quality photograph showing two men from the waist up. One man on the left is wearing a red t-shirt and has his arm around the other man. The man on the right is wearing a grey t-shirt and is looking towards the camera. They appear to be standing in a hallway with a door visible in the background.

Não esqueça tratar as ferramentas
externas, como backup e
monitoramento, como parte do
próprio banco.



5 - Construa, lance e Execute



Use IaC para criar os servers/containers , instalar e configurar seus banco de dados.

Objetivo final da infra *deve* server imutável.

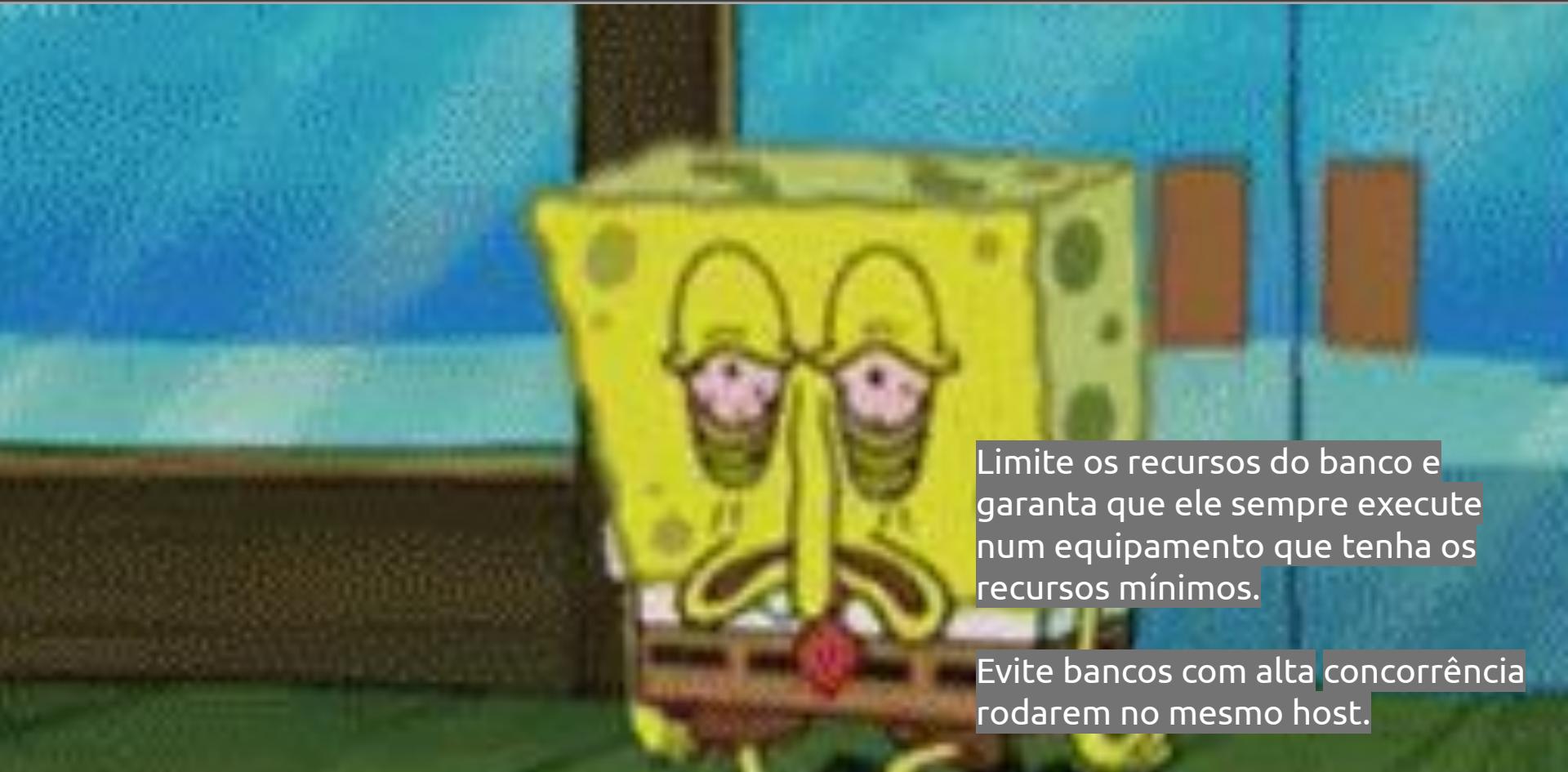
6 - Processos e 7 - Vínculo de portas



Crie suas configurações a fim de garantir que sejam independentes do ambiente, podendo rodar em portas e locais diferentes.

Cada banco vai precisar de seu disco independente.

8 - Concorrência



Limite os recursos do banco e garanta que ele sempre execute num equipamento que tenha os recursos mínimos.

Evite bancos com alta concorrência rodarem no mesmo host.

9 - Descartabilidade



Tarefas de upgrade ou mudanças de configuração considere aplicá-las em outro host, movendo os dados até ele - ou através da promoção de uma réplica.

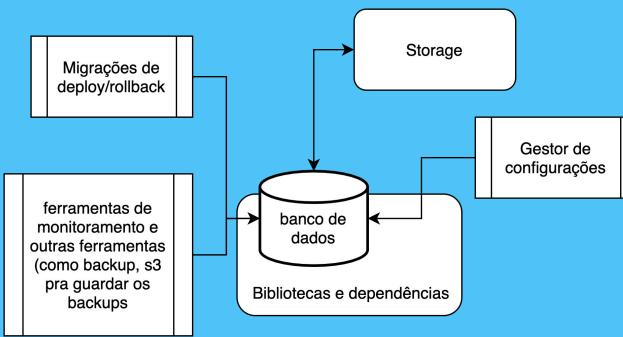
10 - Paridade entre desenvolvimento e produção



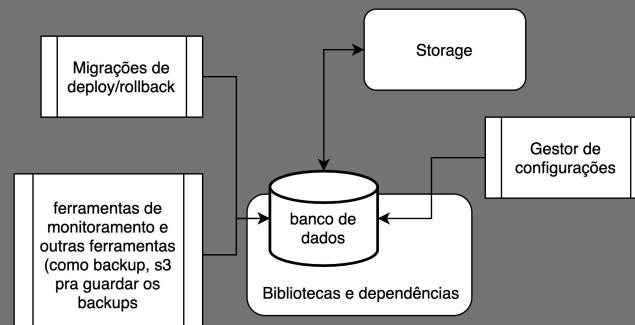
Use a mesma arquitetura pra
todos os ambientes.

Sério.

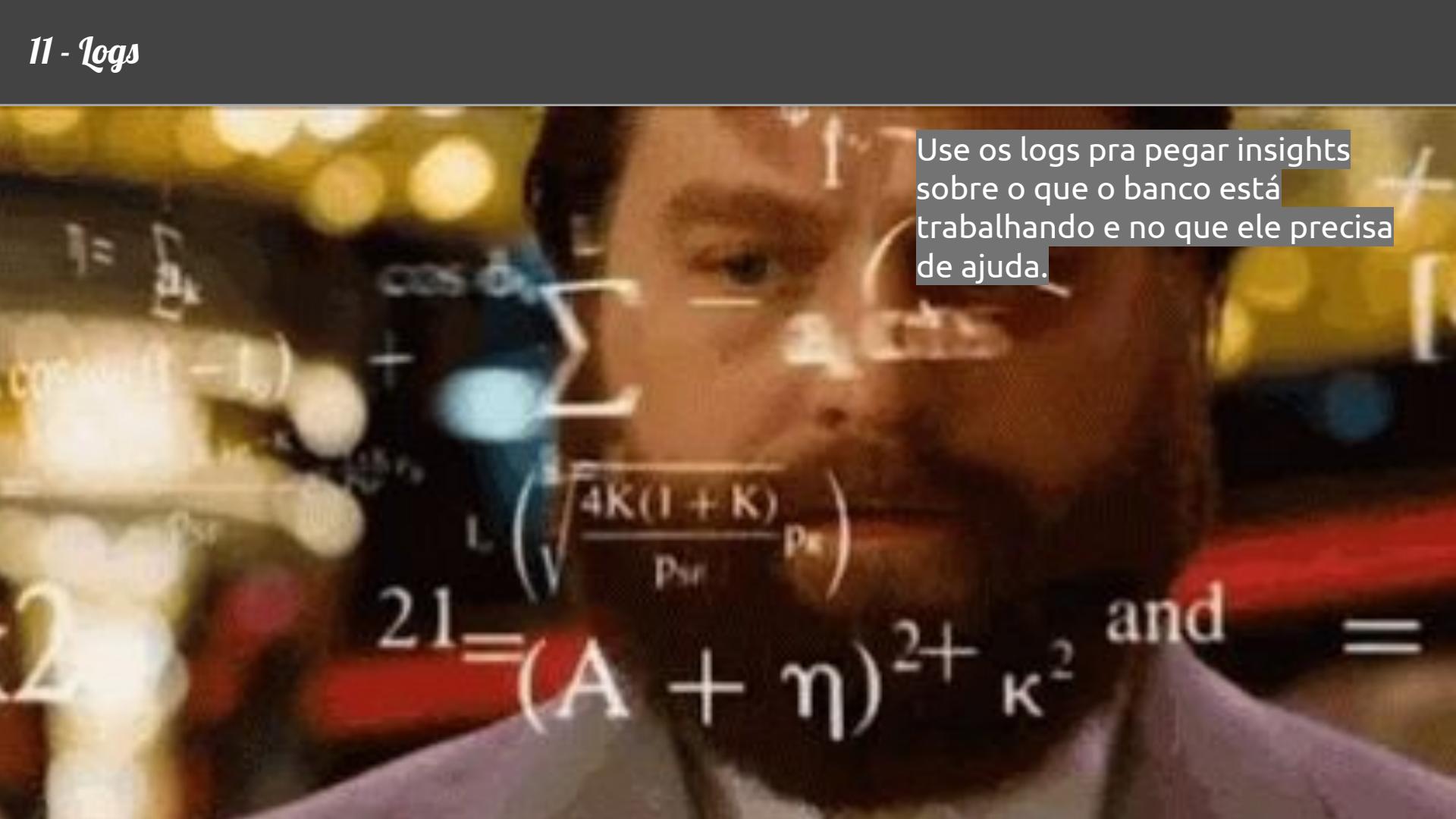
Dev



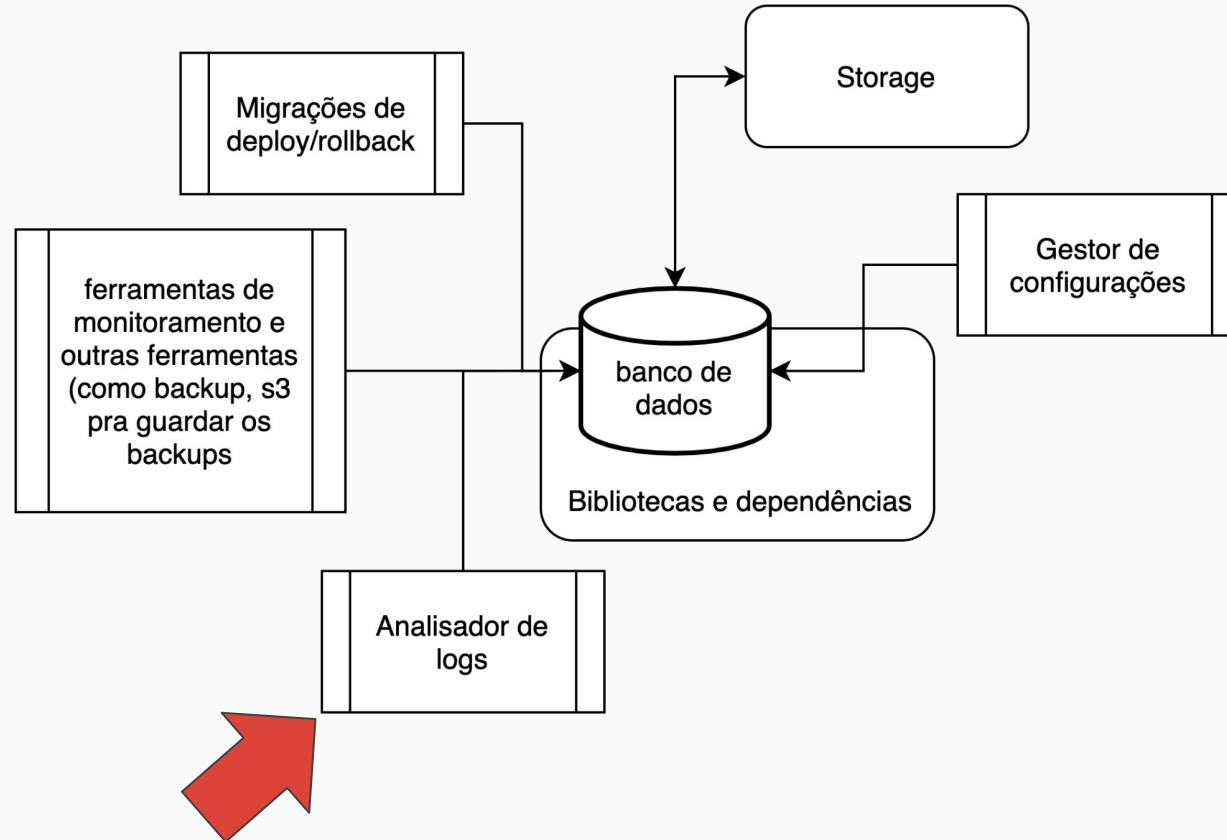
Produção



11 - Logs



Use os logs pra pegar insights sobre o que o banco está trabalhando e no que ele precisa de ajuda.

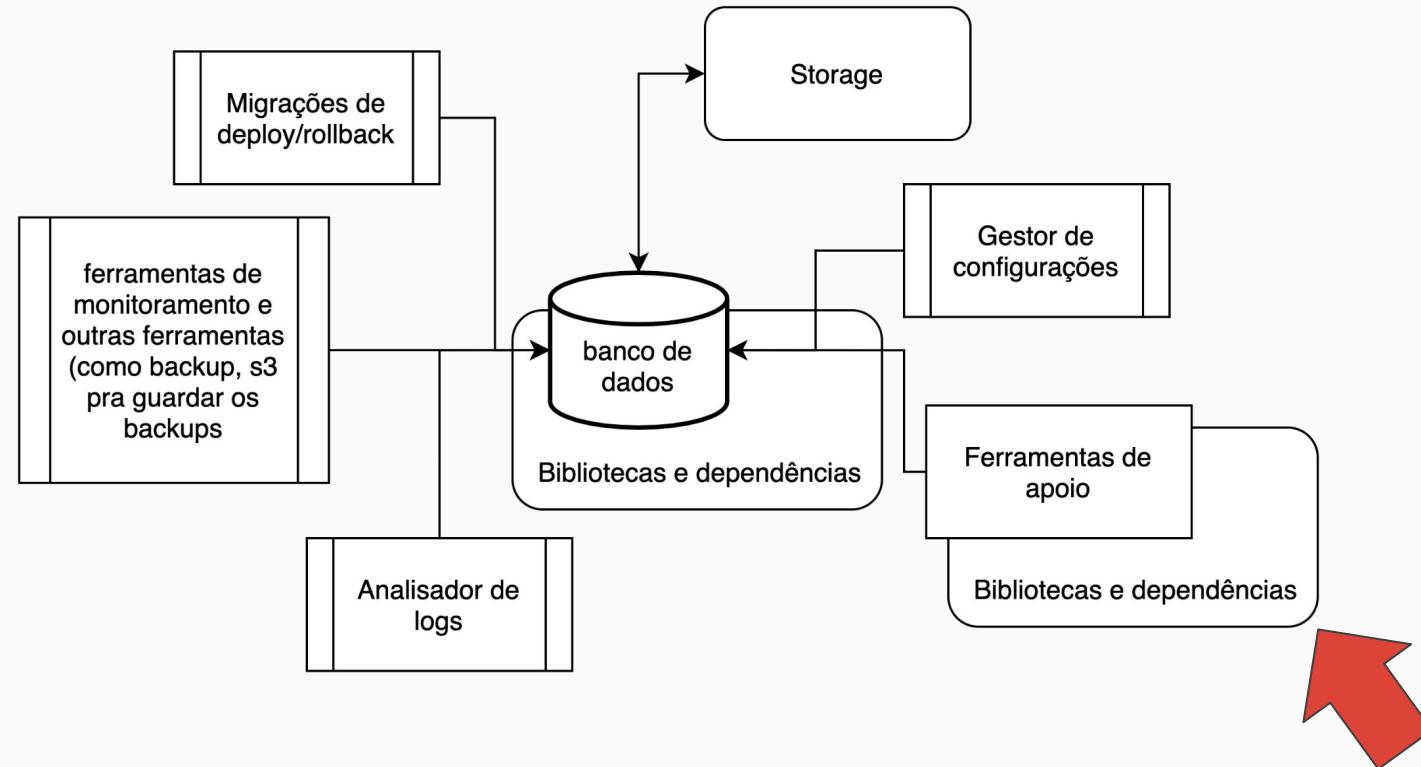


12 - Processos administrativos



Tenha um host/container configurado e pronto pra executar manutenções no banco.

Não improvise. tenha tudo instalado *antes* da manutenção.



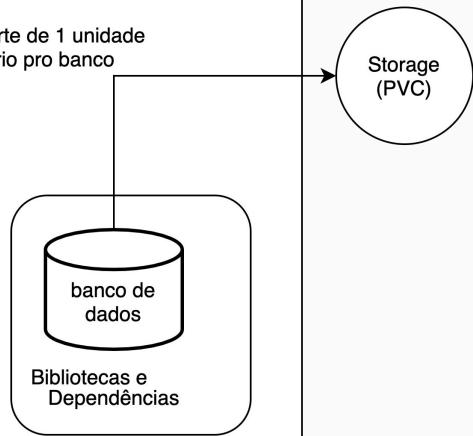
*Ok, agora me
mostra!*

NO K8S...

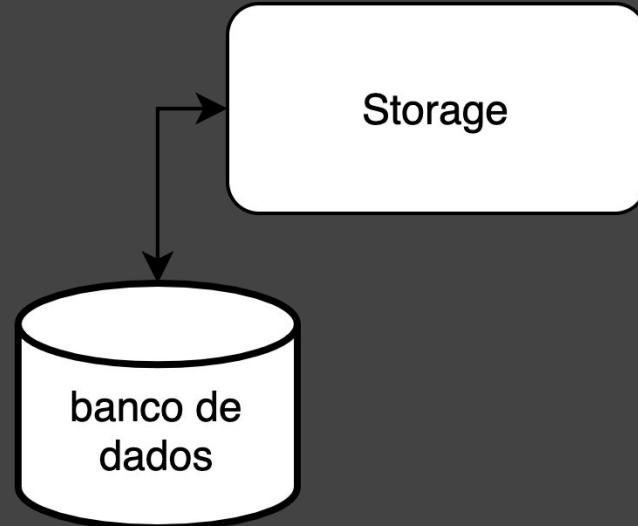
Persistência + Deps

Pod do banco de dados

Todos os seus containers fazem parte de 1 unidade apenas, que compõe tudo necessário pro banco funcionar corretamente.

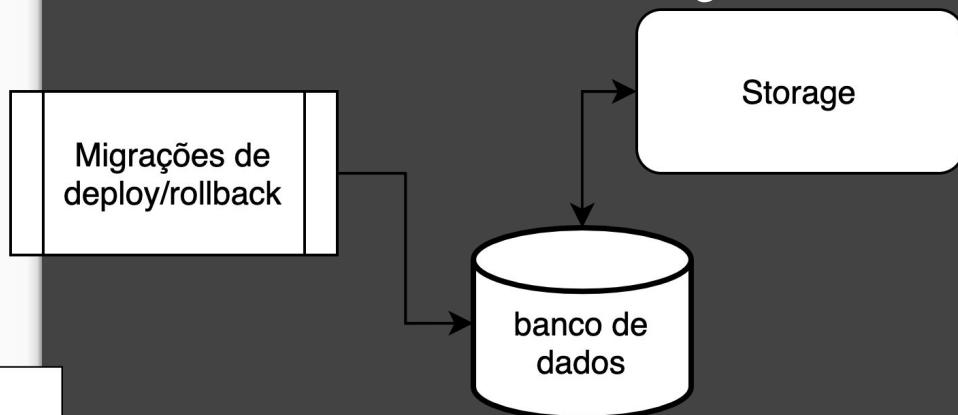


Storage



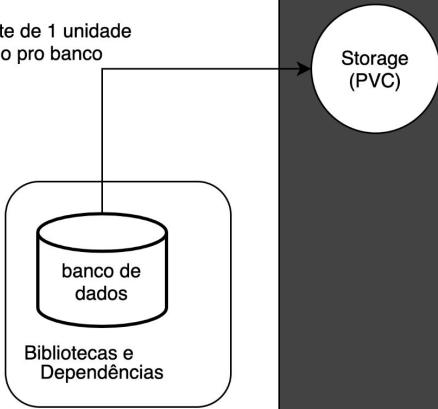
Base de código

NO K8S...



Pod do banco de dados

Todos os seus containers fazem parte de 1 unidade apenas, que compõe tudo necessário pro banco funcionar corretamente.

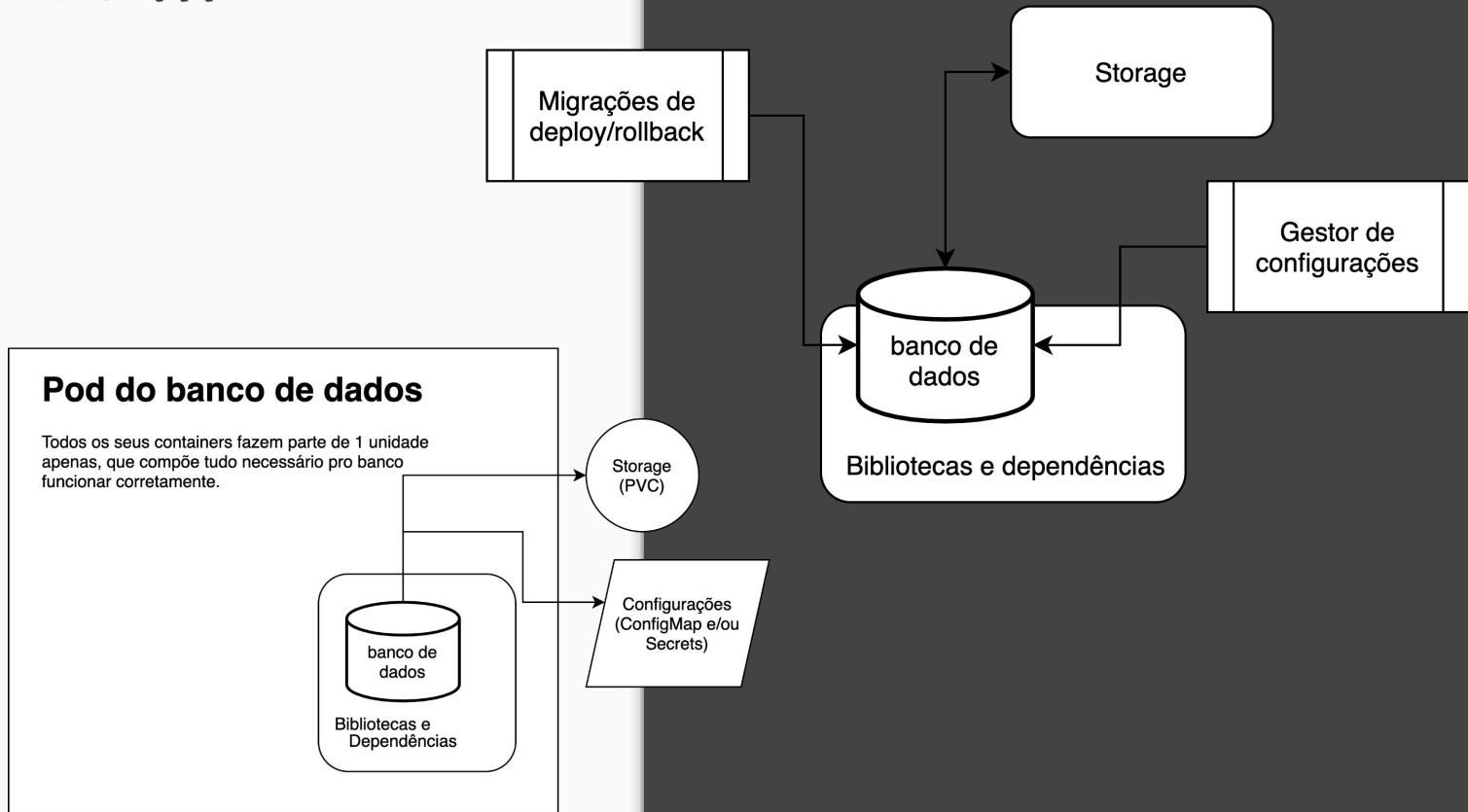


Migrações

Jobs que executam quando necessário.

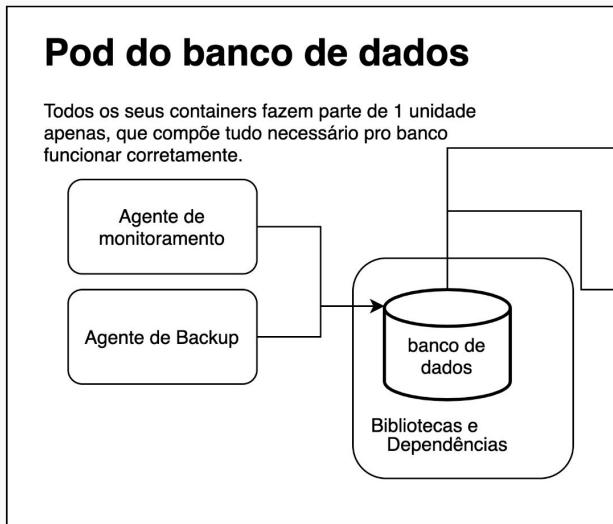
Configurações

NO K8S...



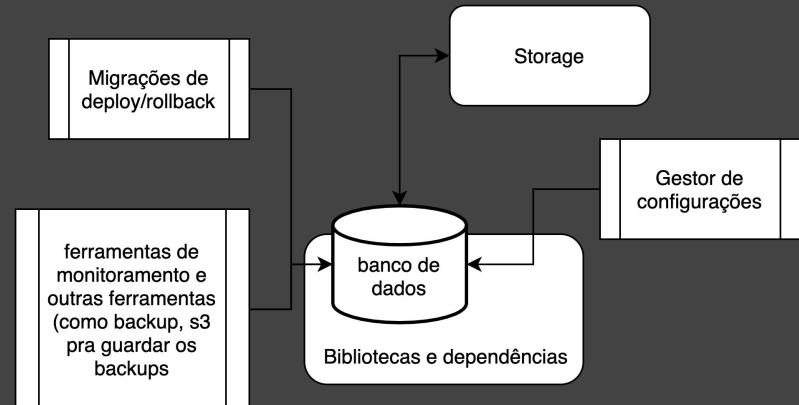
NO K8S...

Serviços de apoio



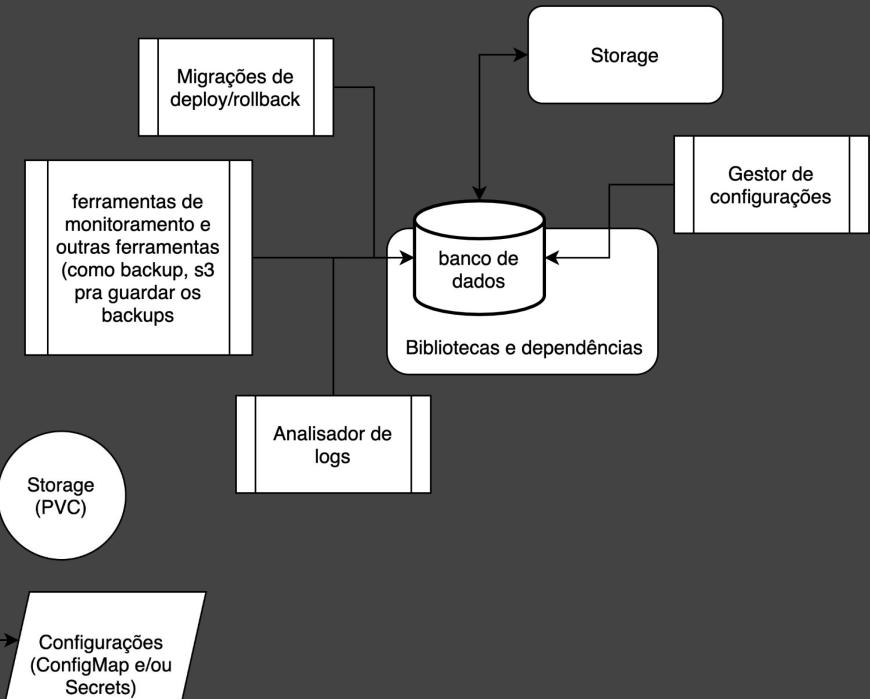
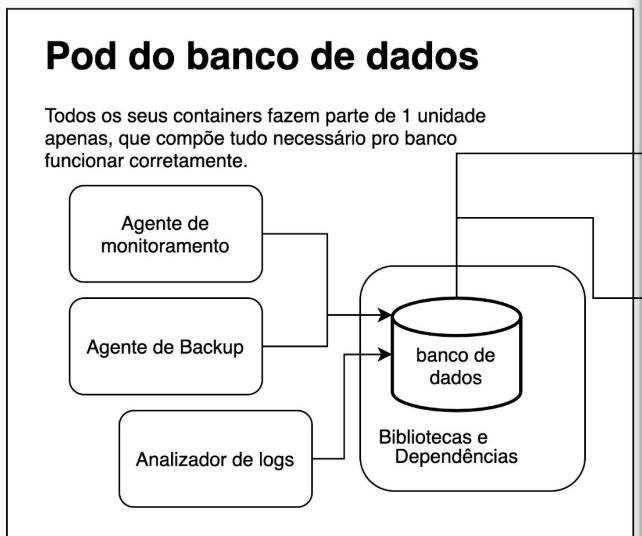
Migrações

Jobs que executam quando necessário.



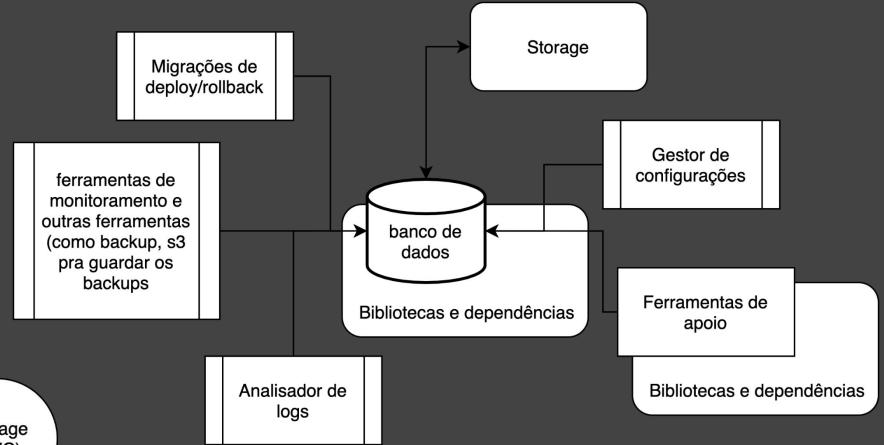
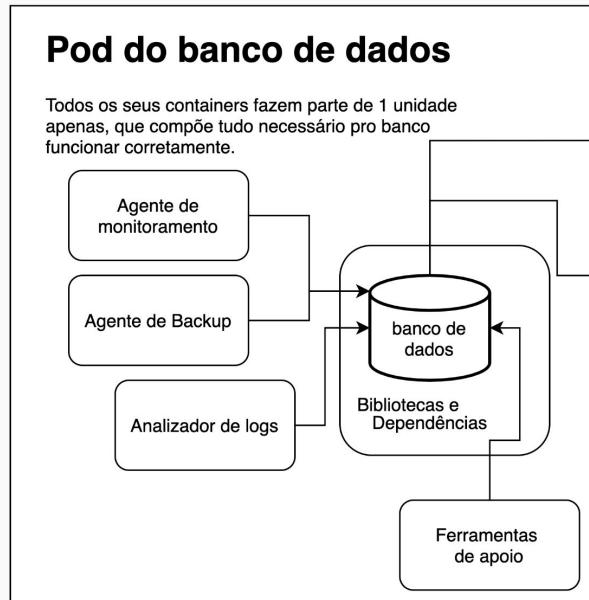
NO K8S...

Logs



NO K8S...

Processos administrativos



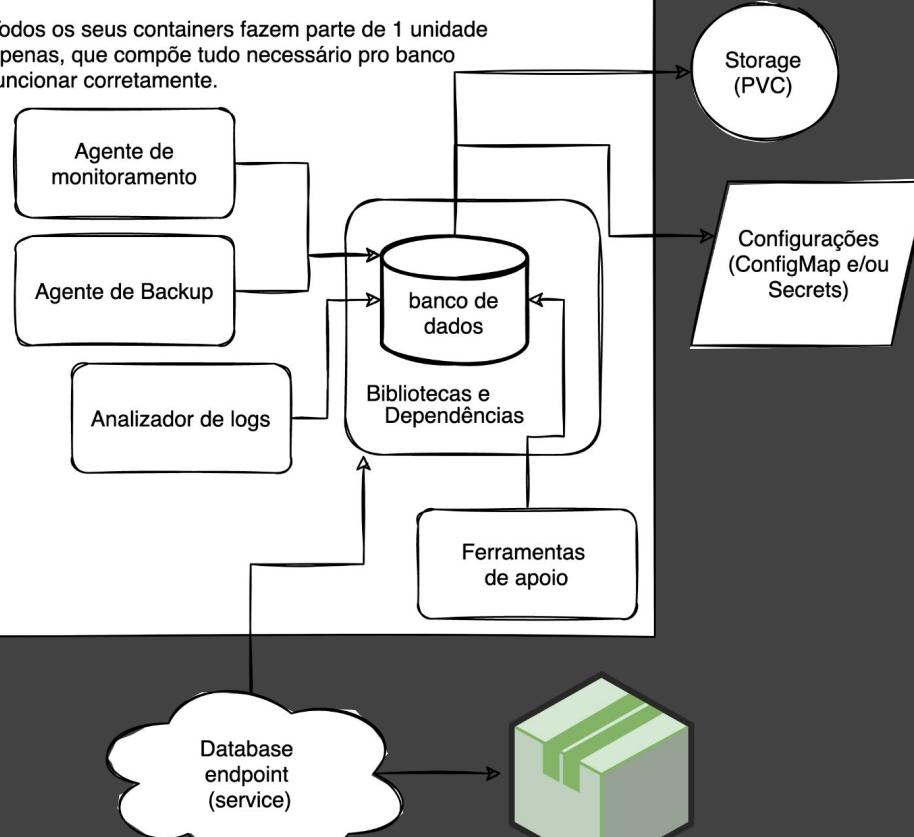
Pra resumir...

Migrações

Jobs que executam quando necessário.

Pod do banco de dados

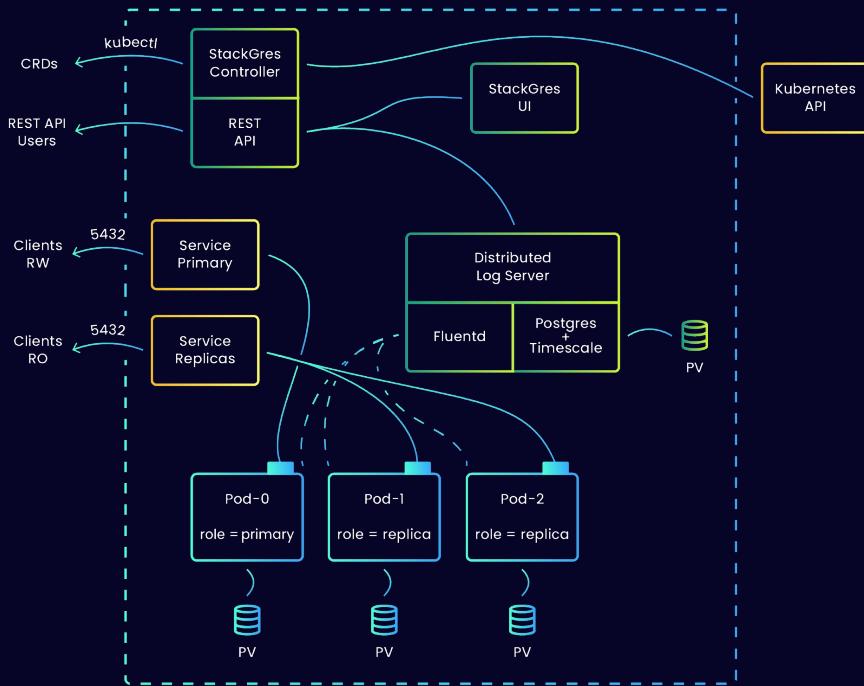
Todos os seus containers fazem parte de 1 unidade apenas, que compõe tudo necessário pro banco funcionar corretamente.



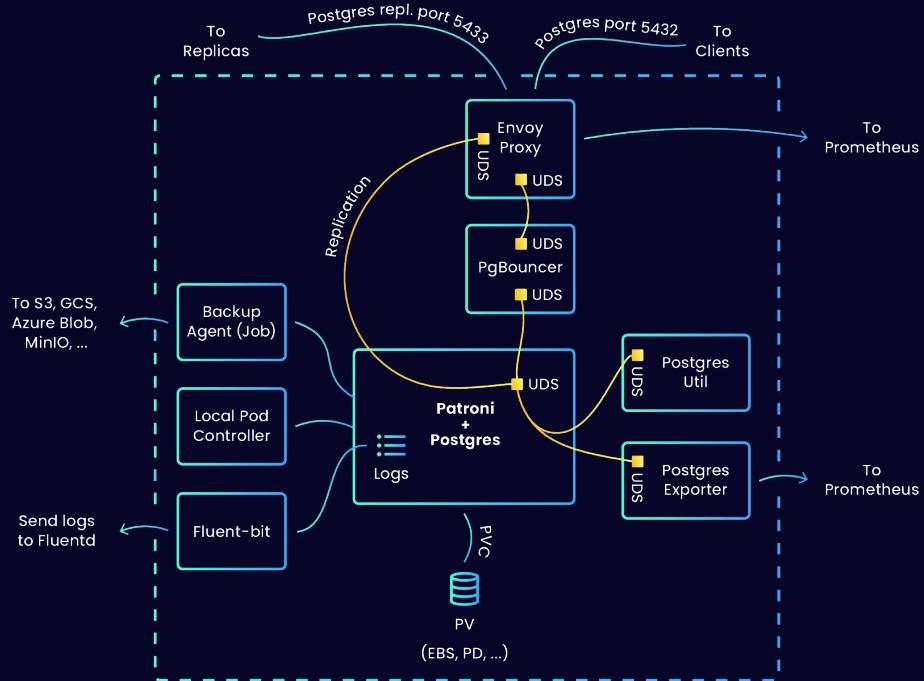
A close-up of the Muppet character Elmo, a red monster with white eyes and a nose, set against a dark, turbulent background of orange and yellow flames.

lets check
the DEMO

StackGres Architecture



Anatomy of the Pod



P

