

Provisionamento, Deployment e monitorização da aplicação

GitLab

GRUPO 4

João Alves
Gonçalo Raposo
Alexandre Dias
Hugo Oliveira



APRESENTAÇÃO

1

COMPONENTES DO GITLAB

Apresentados os diferentes componentes existentes na arquitetura do GitLab.

ARQUITETURA ADOTADA

Apresentada a arquitetura adotada bem como soluções para o aumento da disponibilidade e escalabilidade do serviço através do reconhecimento dos componentes críticos.

MONITORIZAÇÃO E *BENCHMARKING*

Apresentadas ferramentas que permitem a monitorização e avaliação da infraestrutura.

FERRAMENTAS DE AUTOMATIZAÇÃO

Apresentadas as ferramentas que permitem o *deployment* e provisionamento automático da aplicação.

- Plataforma grátis que permite hospedar projetos em servidores.
- Utiliza o git para controlo de versões.

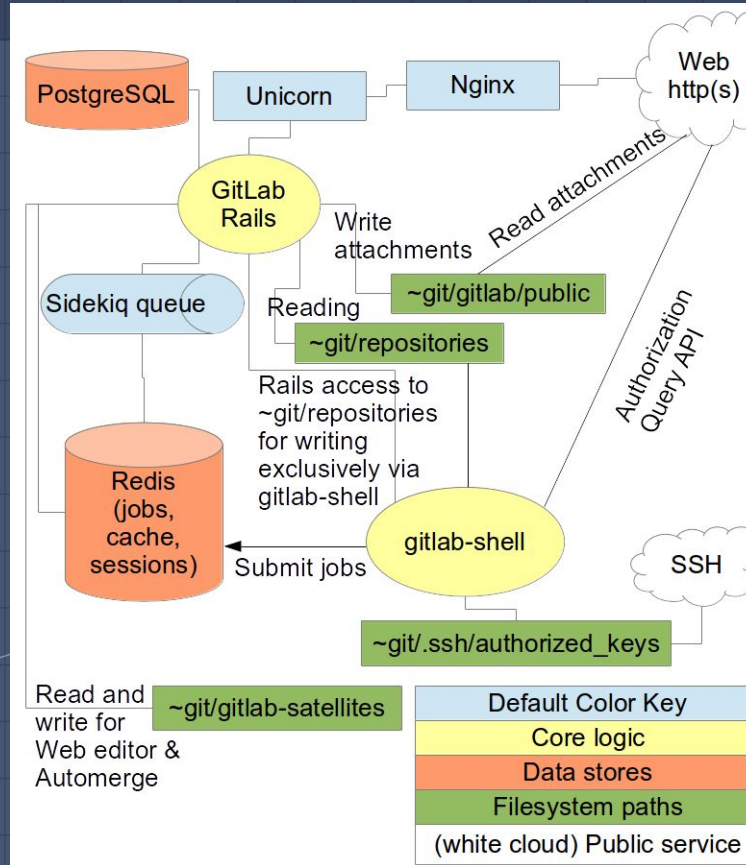
GitLab

e os seus
componentes



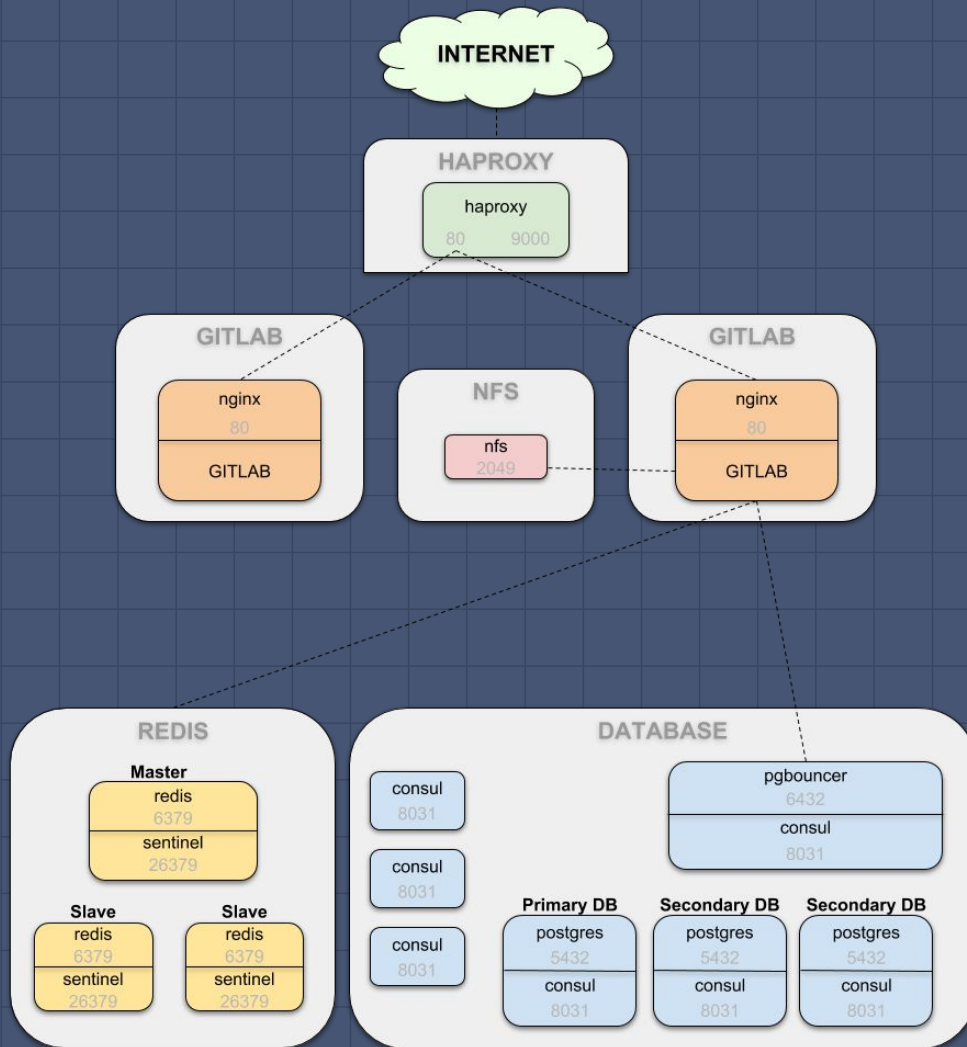
COMPONENTES

3



Arquitetura
adotada





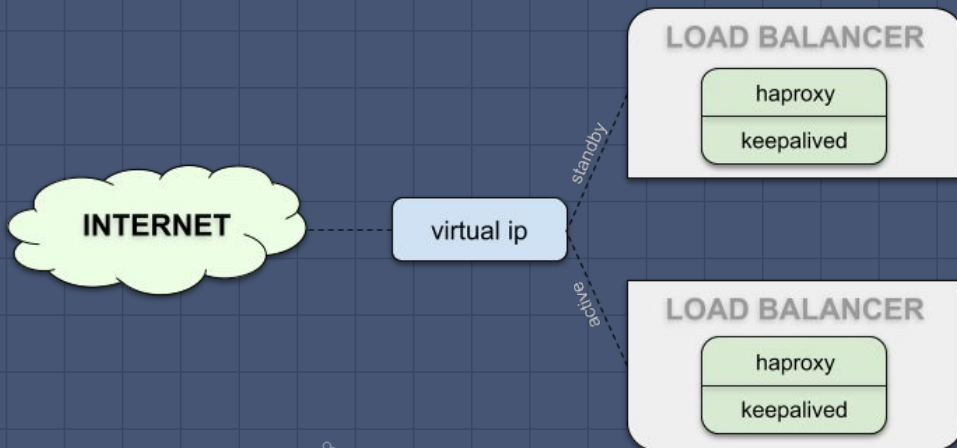
COMPONENTES CRÍTICOS

5

HA Proxy

Balanceamento de carga.
Ponto de falha único.

- Keepalived
- Pacemaker Cluster
- DNS round robin



COMPONENTES CRÍTICOS

6

Redis

Caching de dados como tarefas para o *sidekiq*, armazenamento de sessões.

- Master/Slave
- Redis Sentinel



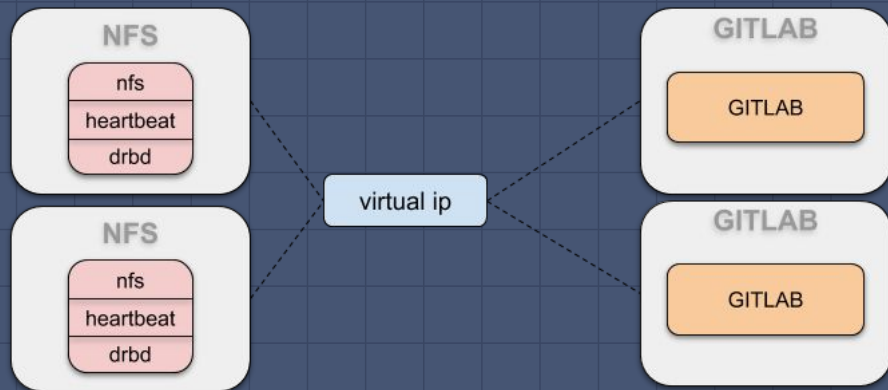
COMPONENTES CRÍTICOS

7

NFS

Armazenamento,
replicação e partilha de
dados pela rede.

- DRBD
- Pacemaker Cluster



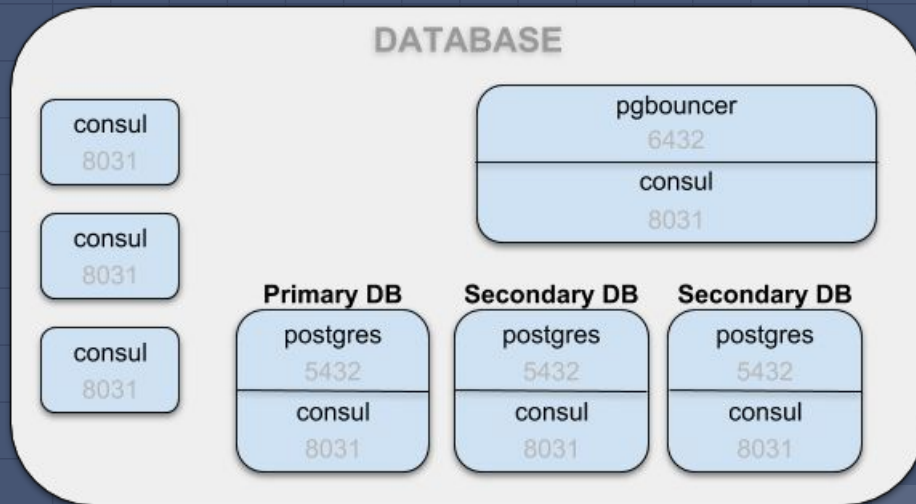
COMPONENTES CRÍTICOS

8

Postgres

Sistema de gestão de base de dados. A disponibilidade é fundamental.

- Master/Slave
- repmgr
- consul e pgbouncer



Monitorização e *benchmarking*



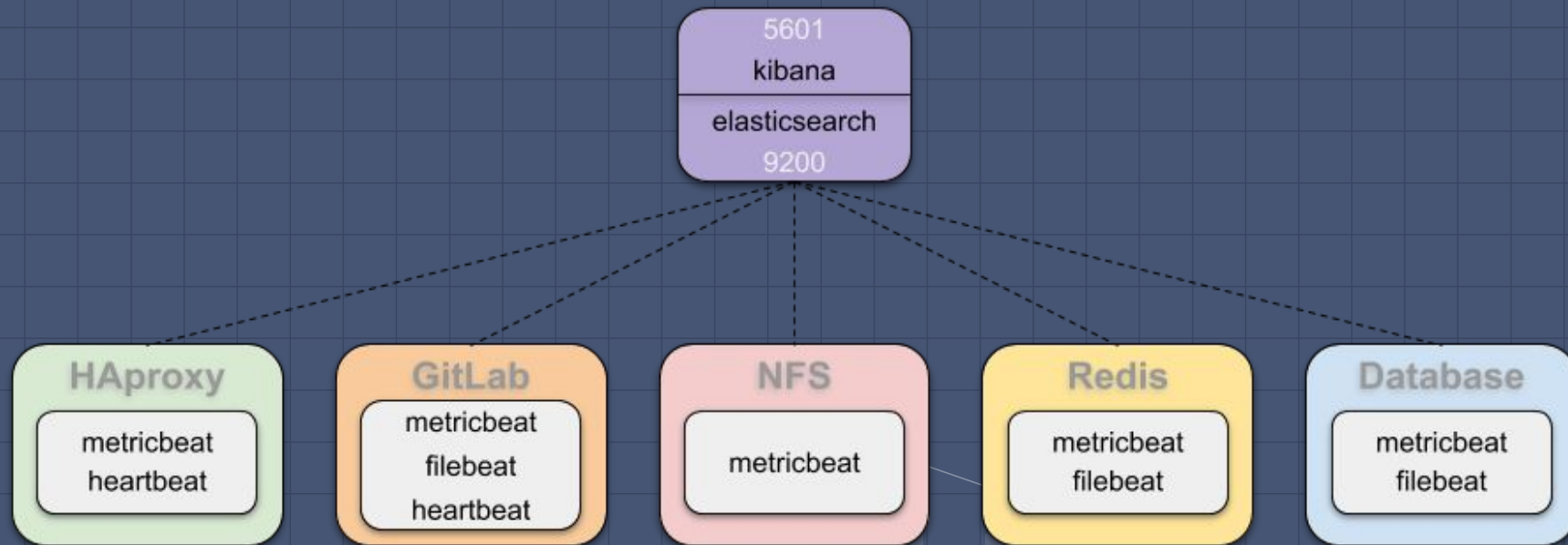
FERRAMENTAS DE MONITORIZAÇÃO

9



FERRAMENTAS DE MONITORIZAÇÃO

10



FERRAMENTA DE *BENCHMARKING*

- Ferramenta utilizada para fazer testes de carga a aplicações *web*.
- Permite gravação de cenários de teste.
- Permite simular o acesso de vários utilizadores em simultâneo.

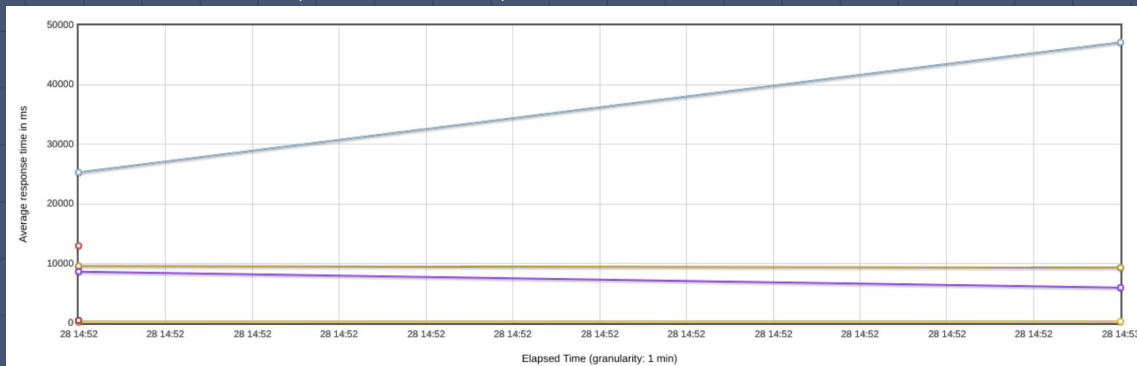


TESTES DE *BENCHMARKING*

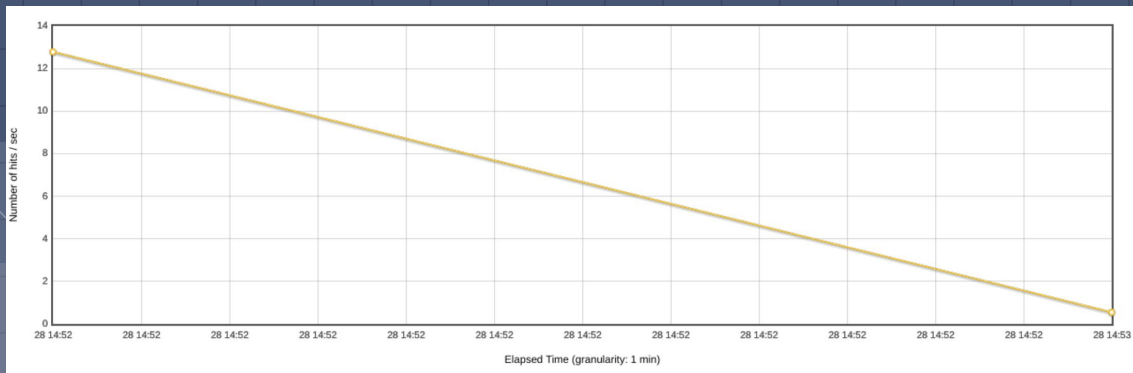
- Stress com *download* de repositórios:
 - Ficheiros de 10MB;
 - 100, 250 e 500 clientes;
- Stress com autenticação:
 - Mesmas credenciais;
 - *Login e logout*;
 - 100, 250 e 500 clientes;

RESULTADOS DO *BENCHMARK 1*

Tempo de resposta (100 clientes)

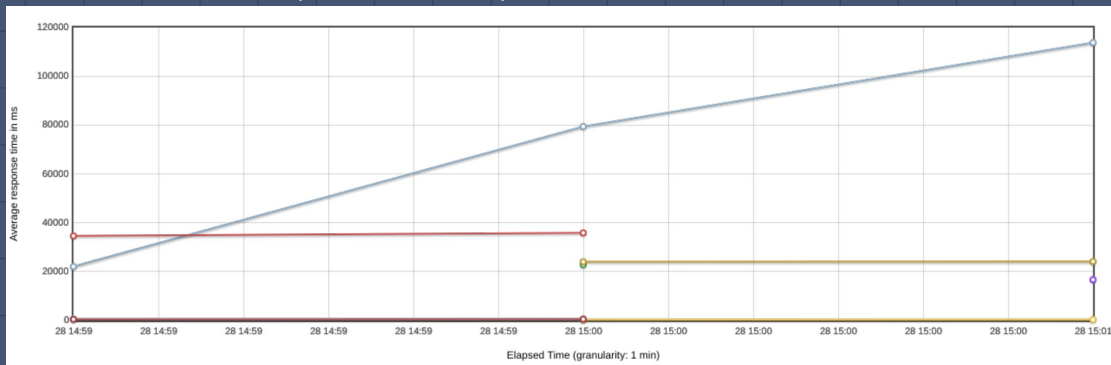


Débito (100 clientes)

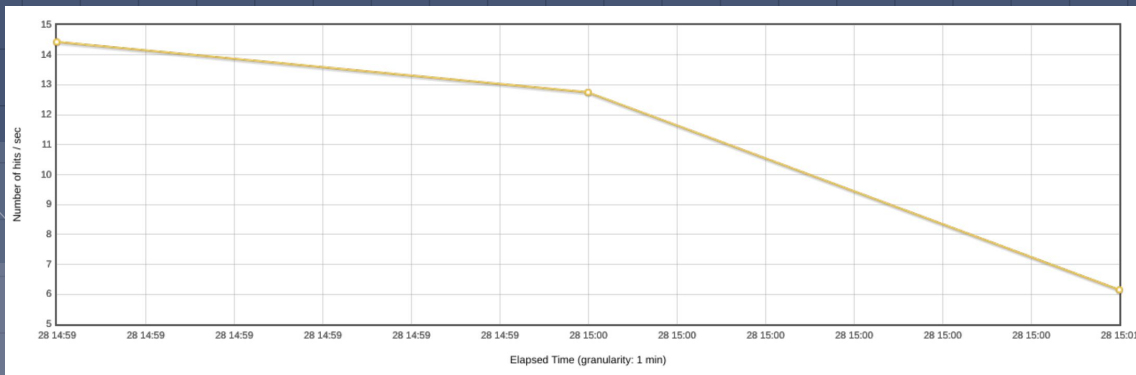


RESULTADOS DO *BENCHMARK 1*

Tempo de resposta (250 clientes)

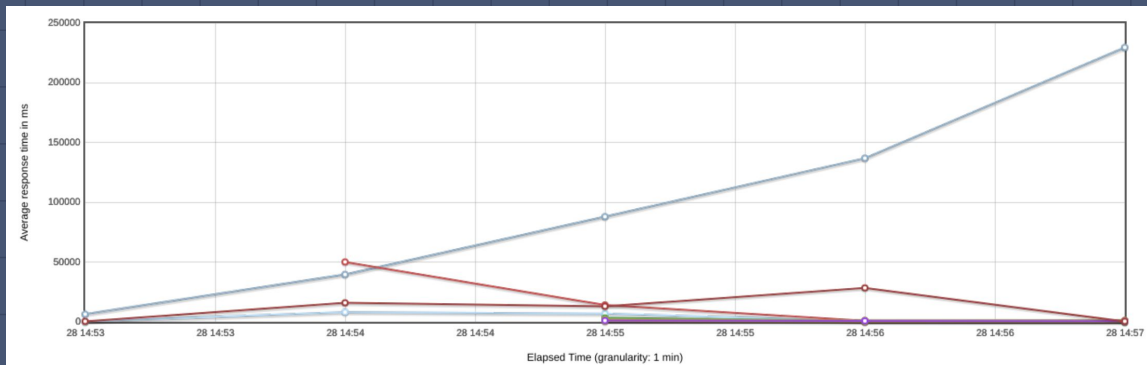


Débito (250 clientes)

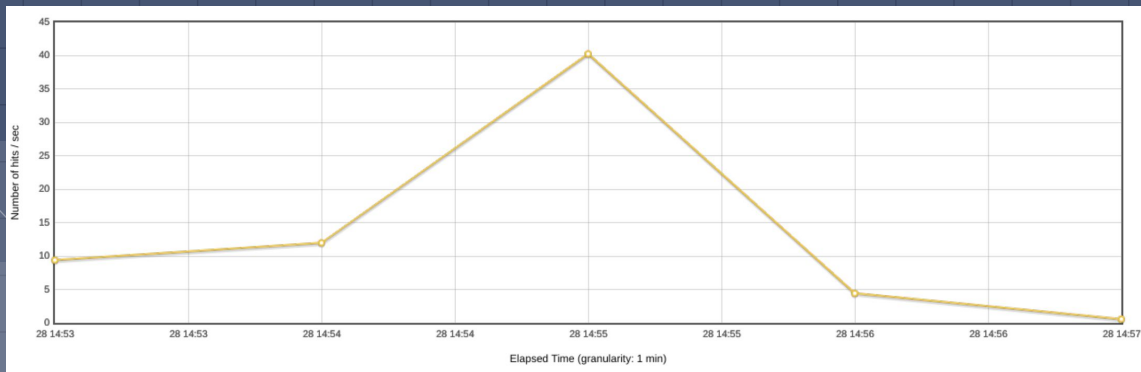


RESULTADOS DO *BENCHMARK 1*

Tempo de resposta (500 clientes)



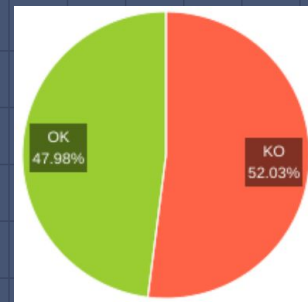
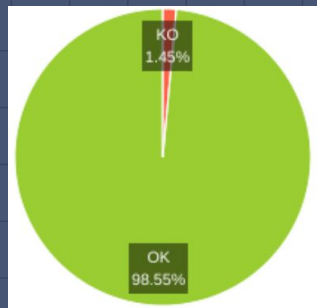
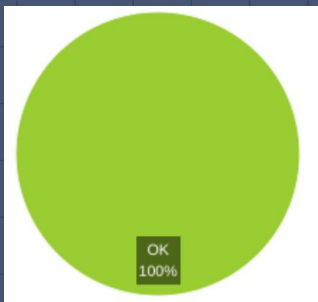
Débito (500 clientes)



RESULTADOS DO *BENCHMARK 1*

16

Percentagem de acerto e erro.



RESULTADOS DO *BENCHMARK 1*

17

250 clientes

vs

500 clientes

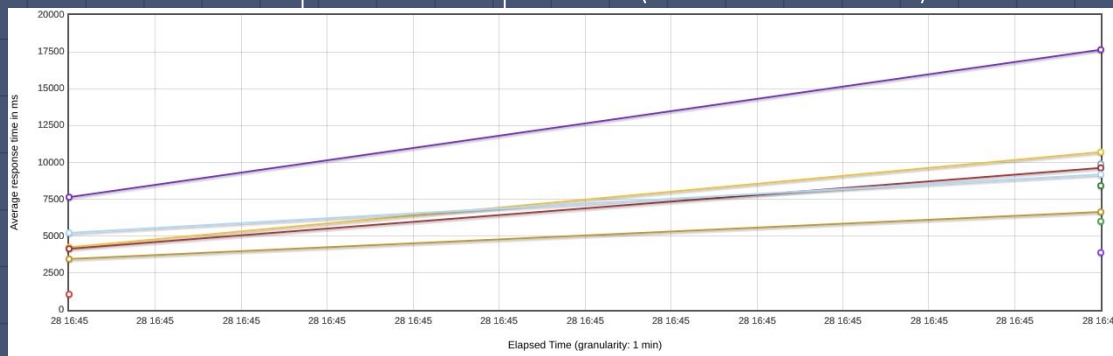
- ▣ Baixos tempos de resposta;
- ▣ Débito baixo;
- ▣ Poucos erros e timeouts.

- ▣ Tempos de resposta elevados;
- ▣ Há um pico onde o débito é máximo;
- ▣ Elevada quantidade de erros.

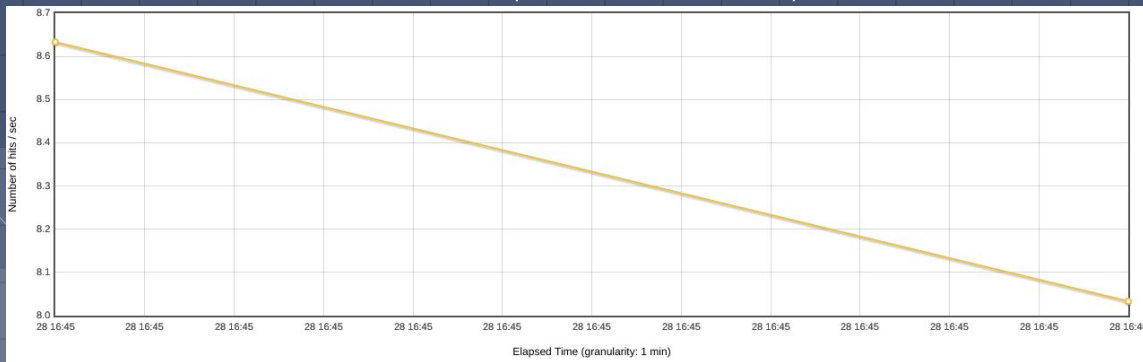


RESULTADOS DO *BENCHMARK 2*

Tempo de resposta (100 clientes)

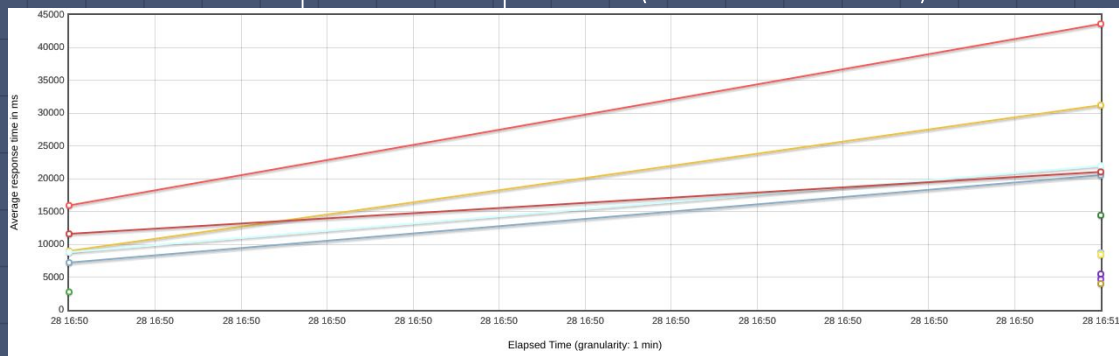


Débito (100 clientes)

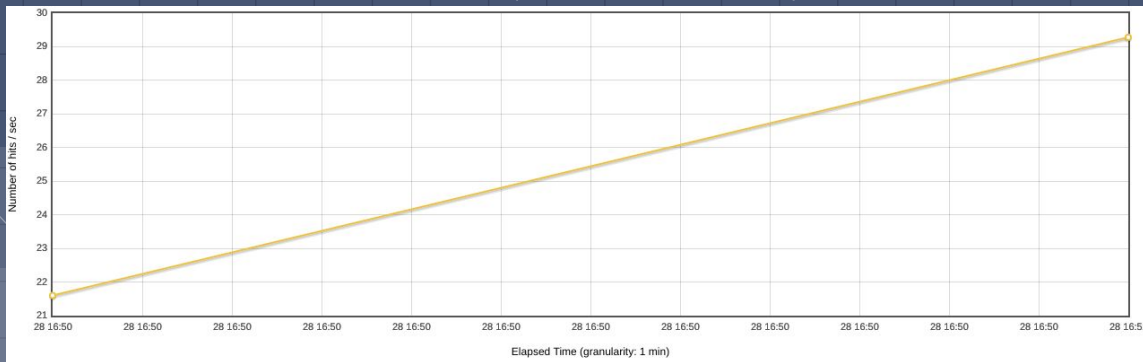


RESULTADOS DO *BENCHMARK 2*

Tempo de resposta (250 clientes)

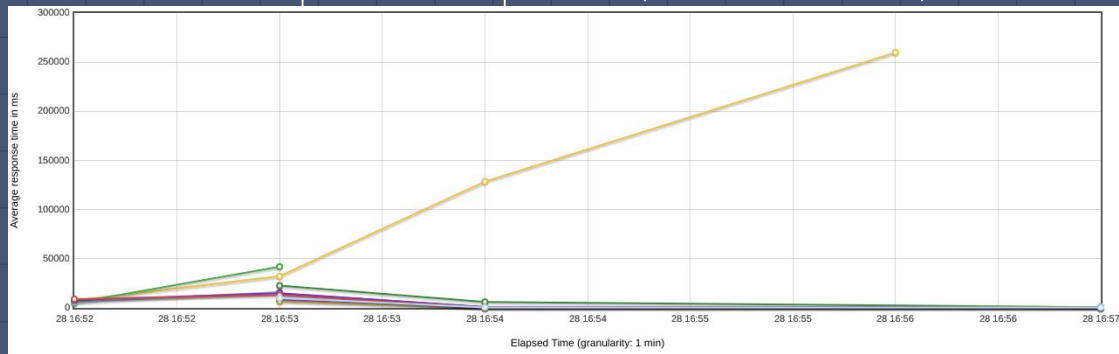


Débito (250 clientes)

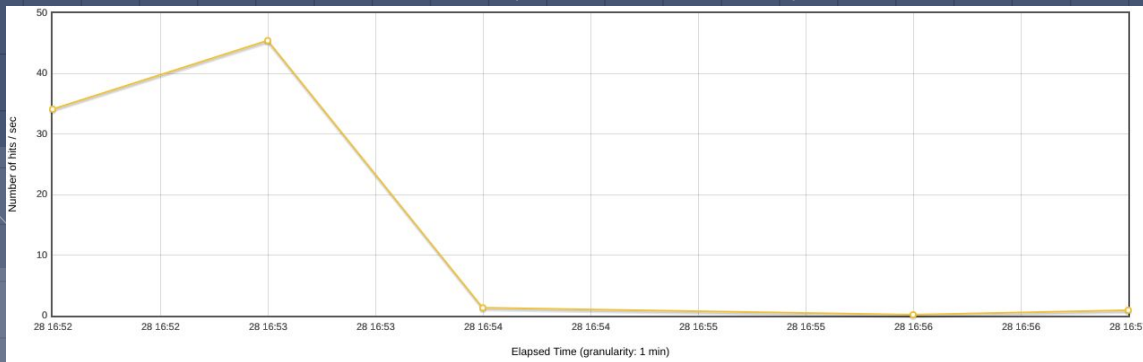


RESULTADOS DO *BENCHMARK 2*

Tempo de resposta (500 clientes)



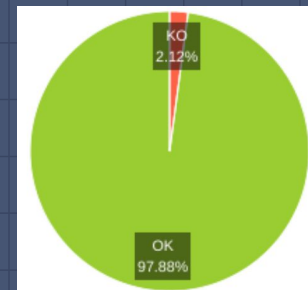
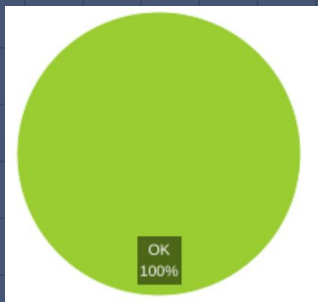
Débito (500 clientes)



RESULTADOS DO *BENCHMARK 2*

21

Percentagem de acerto e erro.



RESULTADOS DO *BENCHMARK 2*

250 clientes

vs

500 clientes

- Baixos tempos de resposta;
- Débito baixo;
- Sem erros.

- Baixos tempos de resposta;
- Há um pico onde o débito é máximo;
- Poucos erros.

Ferramentas de instalação automática



FERRAMENTAS UTILIZADAS

- “Ansible is open source software that automates software provisioning, configuration management, and application deployment.”



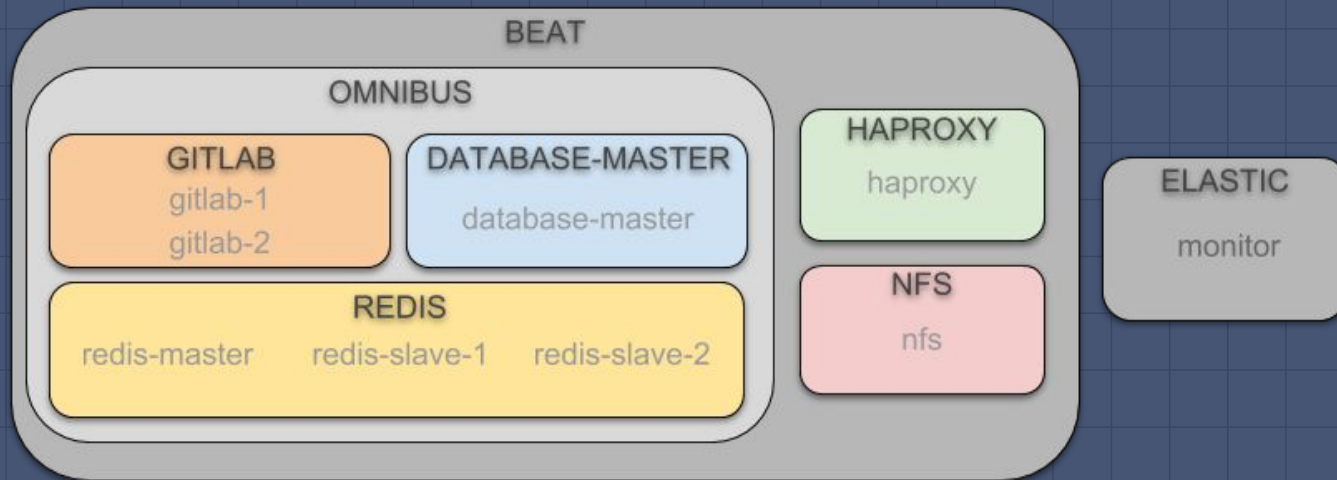
ANSIBLE

- “Omnibus is a way to package different services and tools required to run GitLab... without laborious configuration.”



FERRAMENTAS UTILIZADAS

24



Provisionamento, Deployment e monitorização da aplicação

GitLab

GRUPO 4

João Alves
Gonçalo Raposo
Alexandre Dias
Hugo Oliveira

