

ANÁLISE DE DADOS UTILIZANDO *CLUSTER* DE BAIXO CUSTO

Tendências de consumo da azitromicina no Brasil antes e durante a
pandemia da COVID-19

Felipe Fonseca Rocha

Orientador: Ítalo Fernando Scotá Cunha

Universidade Federal de Minas Gerais

09 de Fevereiro de 2022

Sumário

1 Contexto e Motivação

2 Justificativa

3 Objetivo

4 Revisão de literatura

5 Método

6 Resultados

7 Conclusão

1 Contexto e Motivação

2 Justificativa

3 Objetivo

4 Revisão de literatura

5 Método

6 Resultados

7 Conclusão

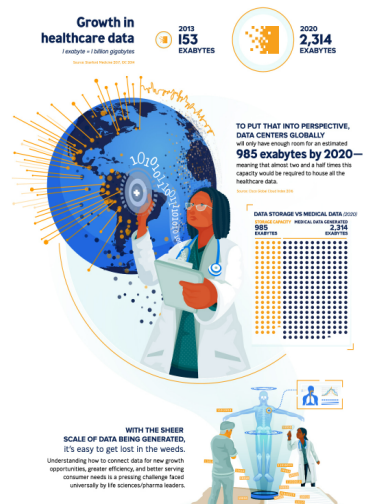
Contexto de Dados – produção e uso

A todo momento nós geramos milhões de dados que são coletados por diferentes meios

Existem várias ferramentas disponíveis para transformá-los em informações e embasar decisões



Contexto de Dados – Área da Saúde



Isso também acontece na área da saúde

Porém o uso de ferramentas de *big data* em saúde ainda é pouco significativo

Boa parte dessas ferramentas implica processamento distribuído

Contexto de Dados – Desafios

Potencial de melhora do sistema de saúde através de análise de dados

Integrar times com trabalho interdisciplinar

Uso de ferramentas e recursos já disponíveis de maneira correta



1 Contexto e Motivação

2 Justificativa

- Justificativa Social
- Justificativa Econômica
- Justificativa Técnica

3 Objetivo

4 Revisão de literatura

5 Método

6 Resultados

7 Conclusão

Apoio a melhores decisões

- Tomada de decisão em saúde
- Escala: **152 milhões** dependem exclusivamente do SUS
- Restrição: Gasto de **R\$3.83** por pessoa por dia
- Transformar dados em informação
- **Assertividade**
 - Ações em saúde
 - políticas públicas

1 Contexto e Motivação

2 Justificativa

- Justificativa Social
- Justificativa Econômica
- Justificativa Técnica

3 Objetivo

4 Revisão de literatura

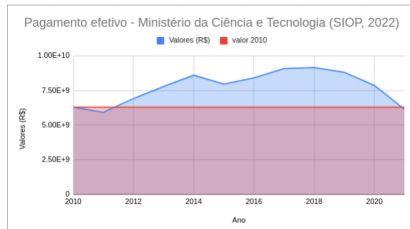
5 Método

6 Resultados

7 Conclusão

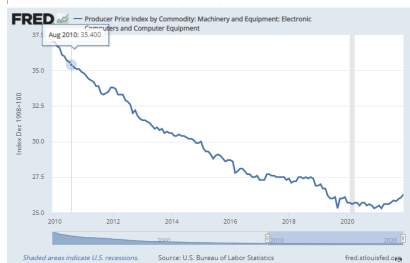
Restrições de orçamento a ciência

- Diminuição de verbas para ciência e tecnologia -2,32%



Alterações de cenário econômico

- Aumento do dólar em mais de 327% diminuindo o poder de compra
- Aumento do custo de hardware e máquinas



1 Contexto e Motivação

2 Justificativa

- Justificativa Social
- Justificativa Econômica
- Justificativa Técnica

3 Objetivo

4 Revisão de literatura

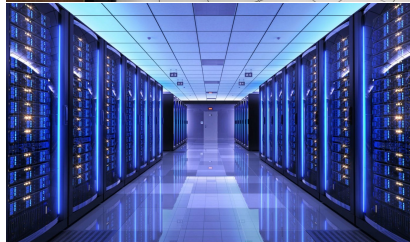
5 Método

6 Resultados

7 Conclusão

Viabilização de alternativas

- Necessário ser interdisciplinar
- Avaliar alternativas de processamento de dados
- Amenizar questões orçamentárias
- Melhorar uso dos recursos já existentes (e.g. inventário de universidades)



1 Contexto e Motivação

2 Justificativa

3 Objetivo

4 Revisão de literatura

5 Método

6 Resultados

7 Conclusão

Objetivo

Objetivos Geral:

Avaliar a viabilidade de orquestração de recursos em *cluster* de baixo custo em ambientes containerizados, para o processamento e a análise dos dados.

Objetivos Específicos:

- Realizar a orquestração de recursos em *cluster* de baixo custo;
- Avaliar tempo de provisionamento, tempo de execução e disponibilidade do cluster;
- Validar o uso de um *cluster* de utilização compartilhada para processamento de dados distribuídos;
- Propor um método de análise em *cluster* Kubernetes com uso de computadores desktops;
- Disponibilizar um cluster pronto para uso para UFMG, bem como ferramentas de auxílio no provisionamento;

- 1 Contexto e Motivação
- 2 Justificativa
- 3 Objetivo
- 4 Revisão de literatura
 - Alternativas *open source*
 - Cluster orquestrador de container
- 5 Método
- 6 Resultados
- 7 Conclusão

Alternativas *open source*

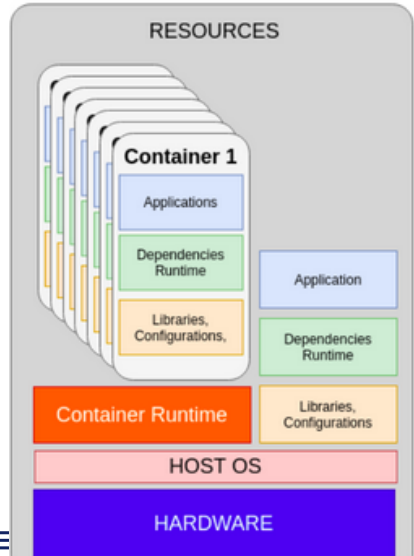
- Considerando
 - O escopo deste trabalho
 - Limitações de hardware
 - As estratégias para processamento
 - Ferramentas de análise de dados disponíveis no mercado

As soluções encontradas no mercado foram agrupadas em dois grupos:

- Soluções de Computação em nuvem privada:
 - ▶ Se estendem para além do propósito desse trabalho
 - ▶ Requisitos de hardware elevados
 - ▶ Complexidade de configuração devido a sua abrangência

Alternativas *open source*

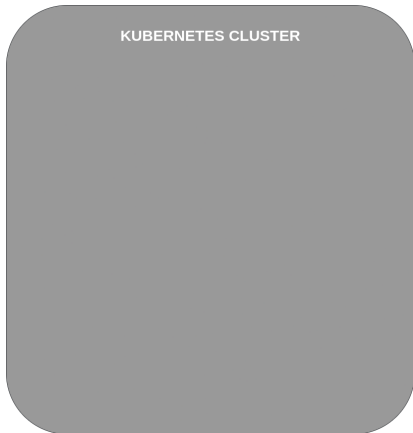
- Soluções de Orquestração de Containers:
 - Kubernetes®
 - Apache Mesos®
 - Hashicorp Nomad®
 - Docker Swarm®



- 1 Contexto e Motivação
- 2 Justificativa
- 3 Objetivo
- 4 Revisão de literatura
 - Alternativas *open source*
 - Cluster orquestrador de container
- 5 Método
- 6 Resultados
- 7 Conclusão

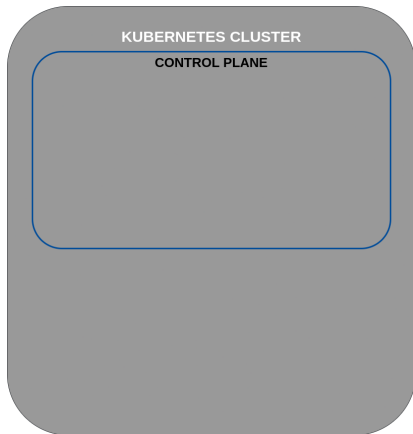
Cluster orquestrador de container

- Kubernetes®:
 - Origem de 15 anos de trabalho da Google (Borg)
 - Estrutura de objetos componentizados
 - ▶ Kube-apiserver
 - ▶ Kube-scheduler
 - ▶ Kube-controller-manager
 - ▶ Kubelet
 - ▶ Kube-proxy
 - ▶ Pod



Cluster orquestrador de container

- Kubernetes®:
 - Origem de 15 anos de trabalho da Google (Borg)
 - Estrutura de objetos componentizados
 - ▶ Kube-apiserver
 - ▶ Kube-scheduler
 - ▶ Kube-controller-manager
 - ▶ Kubelet
 - ▶ Kube-proxy
 - ▶ Pod



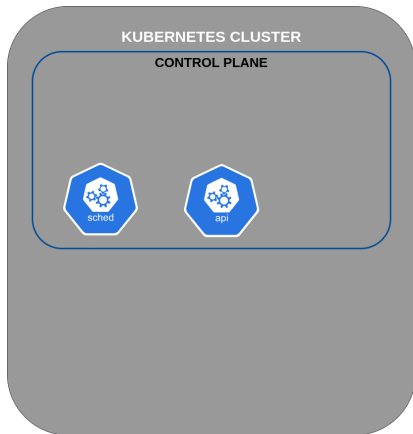
Cluster orquestrador de container

- Kubernetes®:
 - Origem de 15 anos de trabalho da Google (Borg)
 - Estrutura de objetos componentizados
 - ▶ Kube-apiserver
 - ▶ Kube-scheduler
 - ▶ Kube-controller-manager
 - ▶ Kubelet
 - ▶ Kube-proxy
 - ▶ Pod



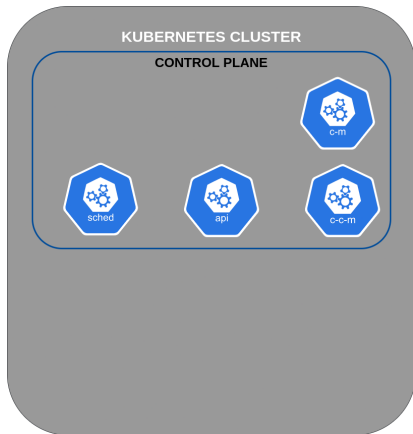
Cluster orquestrador de container

- Kubernetes®:
 - Origem de 15 anos de trabalho da Google (Borg)
 - Estrutura de objetos componentizados
 - ▶ Kube-apiserver
 - ▶ Kube-scheduler
 - ▶ Kube-controller-manager
 - ▶ Kubelet
 - ▶ Kube-proxy
 - ▶ Pod



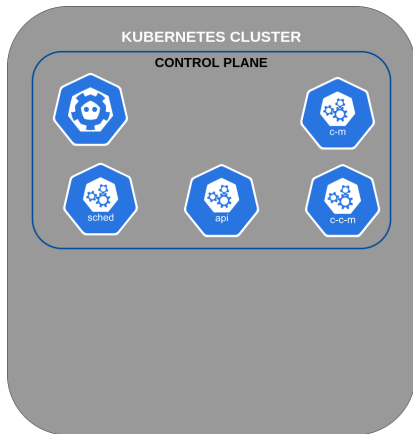
Cluster orquestrador de container

- Kubernetes®:
 - Origem de 15 anos de trabalho da Google (Borg)
 - Estrutura de objetos componentizados
 - ▶ Kube-apiserver
 - ▶ Kube-scheduler
 - ▶ Kube-controller-manager
 - ▶ Kubelet
 - ▶ Kube-proxy
 - ▶ Pod



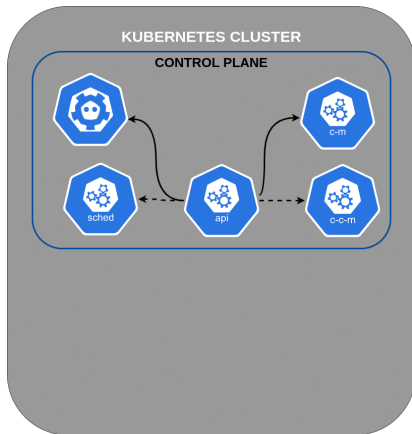
Cluster orquestrador de container

- Kubernetes®:
 - Origem de 15 anos de trabalho da Google (Borg)
 - Estrutura de objetos componentizados
 - ▶ Kube-apiserver
 - ▶ Kube-scheduler
 - ▶ Kube-controller-manager
 - ▶ Kubelet
 - ▶ Kube-proxy
 - ▶ Pod



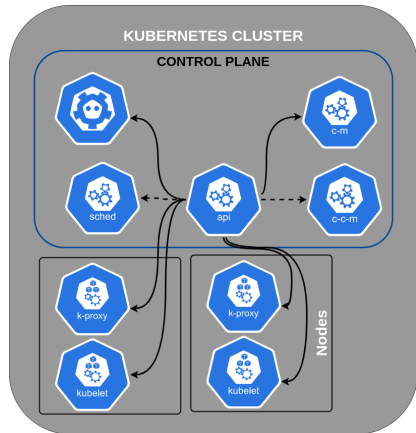
Cluster orquestrador de container

- Kubernetes®:
 - Origem de 15 anos de trabalho da Google (Borg)
 - Estrutura de objetos componentizados
 - ▶ Kube-apiserver
 - ▶ Kube-scheduler
 - ▶ Kube-controller-manager
 - ▶ Kubelet
 - ▶ Kube-proxy
 - ▶ Pod



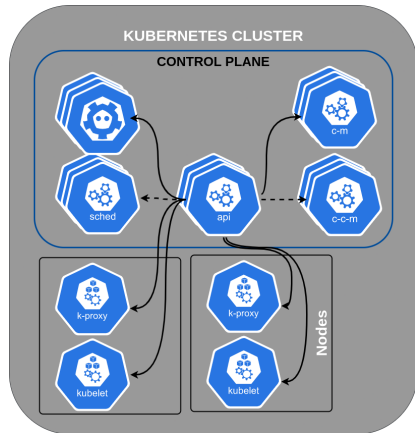
Cluster orquestrador de container

- Kubernetes®:
 - Origem de 15 anos de trabalho da Google (Borg)
 - Estrutura de objetos componentizados
 - ▶ Kube-apiserver
 - ▶ Kube-scheduler
 - ▶ Kube-controller-manager
 - ▶ Kubelet
 - ▶ Kube-proxy
 - ▶ Pod



Cluster orquestrador de container

- Kubernetes®:
 - Origem de 15 anos de trabalho da Google (Borg)
 - Estrutura de objetos componentizados
 - ▶ Kube-apiserver
 - ▶ Kube-scheduler
 - ▶ Kube-controller-manager
 - ▶ Kubelet
 - ▶ Kube-proxy
 - ▶ Pod



1 Contexto e Motivação

2 Justificativa

3 Objetivo

4 Revisão de literatura

5 Método

- Abordagem
- Especificações
- Arquitetura Orquestrador
- Gerenciamento de configuração
- Monitoramento
- Avaliação viabilidade
- Análise de dados

6 Resultados

Abordagem – Cluster e Análise

Utilizar um *Cluster* Kubernetes® como plataforma de orquestração de cargas de trabalho em computadores desktops.

- Cargas de trabalho:
 - Análise de tendência de uso de azitromicina entre 2014 e 2021
- Composição do cluster com computadores *desktops* reaproveitados
- Minimizar trabalho local e priorizar a possibilidade de provisionamento remoto
- redução do CAPEX e otimizar utilização de hardware ocioso ou subutilizado
- reaproveitamento de máquinas

Abordagem – Condução do projeto

O uso de conceitos e metodologias de DevOps:

- Versionamento (Git)
- CI (integração contínua) **make build**
- CD (entrega contínua) **make deploy**
- Monitoramento
 - método USE, parâmetros de utilização, saturação e erro
 - avaliação de utilização dos nós durante processamento

1 Contexto e Motivação

2 Justificativa

3 Objetivo

4 Revisão de literatura

5 Método

- Abordagem
- Especificações
- Arquitetura Orquestrador
- Gerenciamento de configuração
- Monitoramento
- Avaliação viabilidade
- Análise de dados

6 Resultados

Especificações I

- Cluster:
 - Composição:
 - ▶ 1 computadores com 6 CPUs e 8GB de RAM (*load balancer*)
 - ▶ 3 computadores com 6 CPUs e 8GB de RAM (*control-plane*)
 - ▶ 4 computadores com 6 CPUs e 16GB de RAM (*workers*)
 - Containers para processamento e análise:
 - ▶ 90 containers (1/mês de análise) [procesamento]
 - ▶ 1 container / usuário [análise]
 - ▶ arquitetura: **amd64**
 - ▶ 1 vCPU
 - ▶ 2 GB de RAM

Especificações II

- Orquestração do processamento dos dados originias:
 - Apache Airflow®
 - ▶ Kubernetes executor (onde)
 - ▶ Python Operators (como)
- Consumo e análise de dados tratados:
 - JupyterHub - Notebooks para multi-usuários (gerenciamento)
 - Jupyter Lab - Notebooks para análise dos dados (execução)

1 Contexto e Motivação

2 Justificativa

3 Objetivo

4 Revisão de literatura

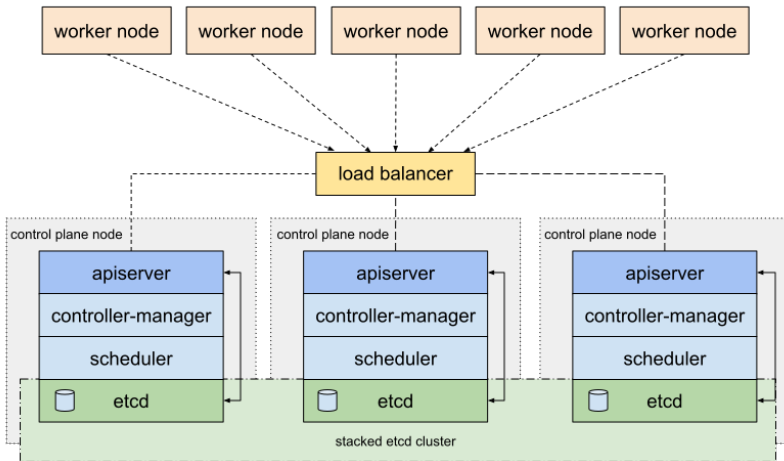
5 Método

- Abordagem
- Especificações
- Arquitetura Orquestrador
- Gerenciamento de configuração
- Monitoramento
- Avaliação viabilidade
- Análise de dados

6 Resultados

Arquitetura Orquestrador

kubeadm HA topology - stacked etcd



1 Contexto e Motivação

2 Justificativa

3 Objetivo

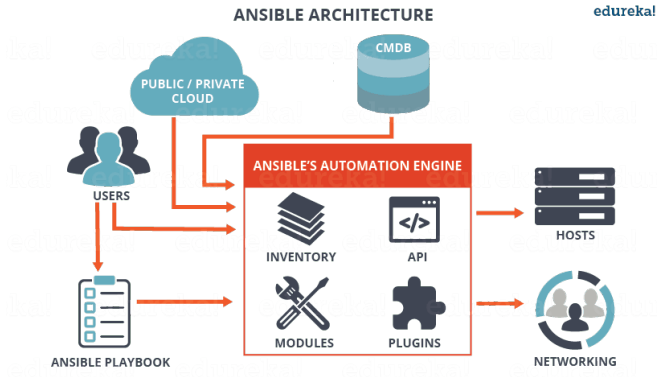
4 Revisão de literatura

5 Método

- Abordagem
- Especificações
- Arquitetura Orquestrador
- Gerenciamento de configuração
- Monitoramento
- Avaliação viabilidade
- Análise de dados

6 Resultados

Gerenciamento de configuração



1 Contexto e Motivação

2 Justificativa

3 Objetivo

4 Revisão de literatura

5 Método

- Abordagem
- Especificações
- Arquitetura Orquestrador
- Gerenciamento de configuração
- Monitoramento
- Avaliação viabilidade
- Análise de dados

6 Resultados

Monitoramento

- *Node Exporter* Expõe métricas de Host
- *Prometheus* - Monitoramento de sistemas e Banco de dados de series temporais
- *Grafana* - Dashboard e observabilidade
- *Airflow* - Relatório de tempo de execução, falhas, tentativas

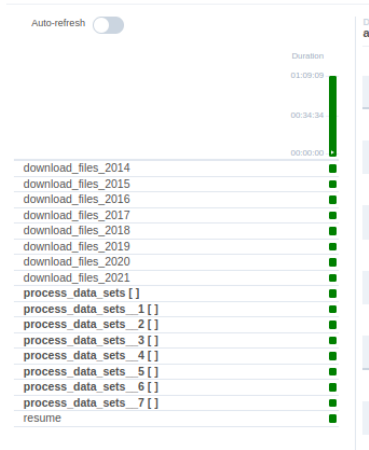


Figura: Airflow - Relatório de execução

1 Contexto e Motivação

2 Justificativa

3 Objetivo

4 Revisão de literatura

5 Método

- Abordagem
- Especificações
- Arquitetura Orquestrador
- Gerenciamento de configuração
- Monitoramento
- Avaliação viabilidade
- Análise de dados

6 Resultados

Avaliação de utilização do cluster I

- macrobenchmark (system level benchmark) - Teste utilizando uma solução avaliando tempo de execução
métricas de Desempenho (nós do cluster, *guests*):
- Taxa de Utilização de CPU e Memória
- Taxa de saturação de CPU e Memória
- Tempo de Implementação:
- Tempo de configuração do cluster
- Método base utilizado para coleta de informações:
- Metodo USE de avaliação (Checklist Linux)

1 Contexto e Motivação

2 Justificativa

3 Objetivo

4 Revisão de literatura

5 Método

- Abordagem
- Especificações
- Arquitetura Orquestrador
- Gerenciamento de configuração
- Monitoramento
- Avaliação viabilidade
- Análise de dados

6 Resultados

Exemplo da Análise de dados

- Vendas de Medicamentos Controlados e Antimicrobianos – Medicamentos Industrializados
- $530 \cdot 10^6$ linhas com mais de 70 GB
- Análise de tendência do consumo de azitromicina por região
- Análise de tendência do consumo de azitromicina no país
- Avaliação comparativa de 2 anos anteriores ao COVID-19

Disponibilidade dos recursos

Todos os componentes definidos neste trabalho estarão contidos em um repositório público <https://github.com/felipefrocha/esufmg-tcc>, sob a licença pública geral GNU versão 3, para livre acesso.

1 Contexto e Motivação

2 Justificativa

3 Objetivo

4 Revisão de literatura

5 Método

6 Resultados

- Provisionamento
- Configuração
- Resultados do Monitoramento
- Resultados das Análises

7 Conclusão

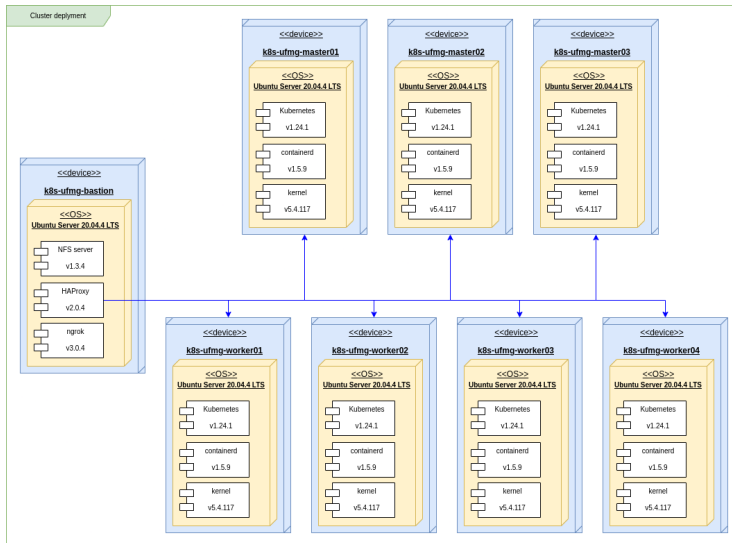


Figura: Diagrama de deploy – OS e versões

NgRok - Acesso Remoto

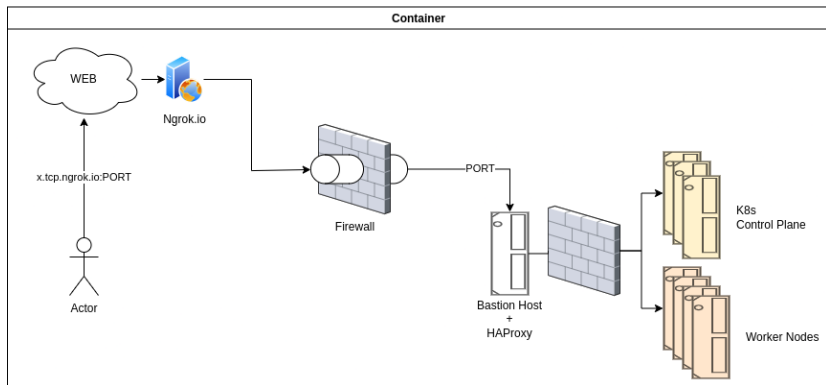



Figura: Funcionamento NgRok

Endpoints

An Endpoint is the access point for anything you use with ngrok.

🔍 Filter endpoints...

ID ↕	Region ↕	URL ↕
ep_... 	US	tcp://...tcp.ngrok.io:2222

1 Contexto e Motivação

2 Justificativa

3 Objetivo

4 Revisão de literatura

5 Método

6 Resultados

- Provisionamento
- Configuração
- Resultados do Monitoramento
- Resultados das Análises

7 Conclusão

Jupyter Execution

Figura: Jupyter - Diagrama de Sequencia

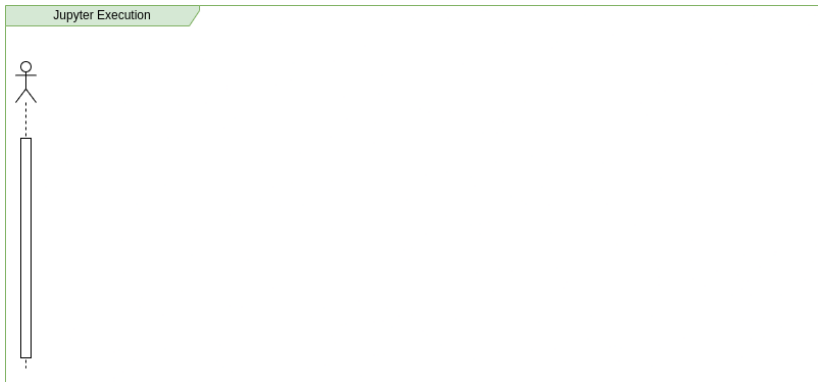


Figura: Jupyter – Diagrama de Sequencia

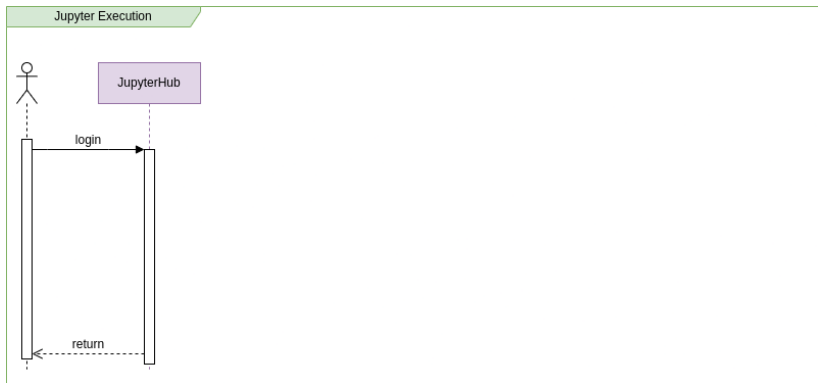


Figura: Jupyter – Diagrama de Sequencia

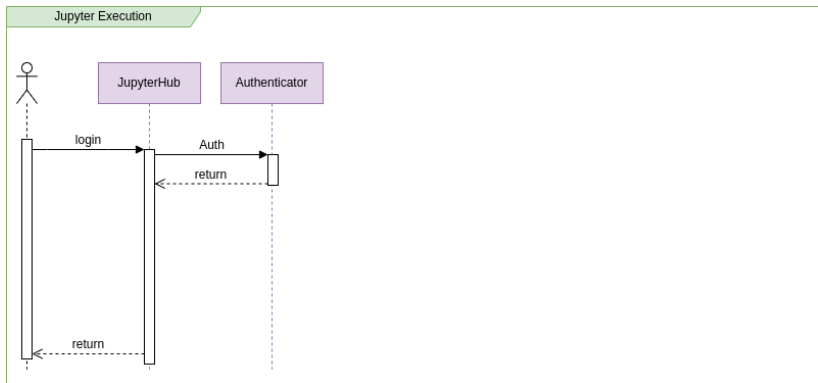


Figura: Jupyter – Diagrama de Sequencia

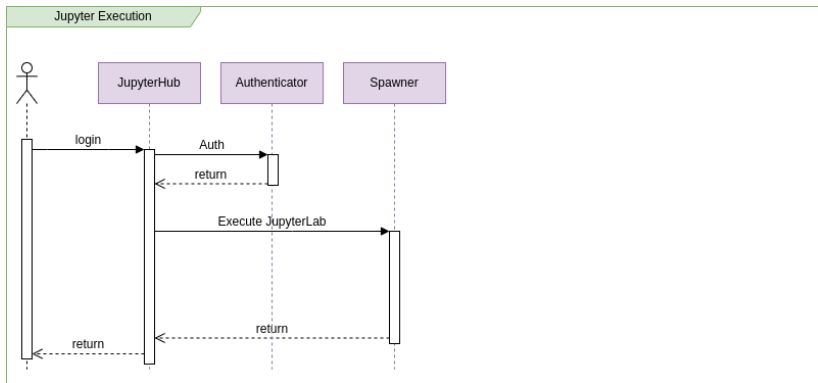


Figura: Jupyter – Diagrama de Sequencia

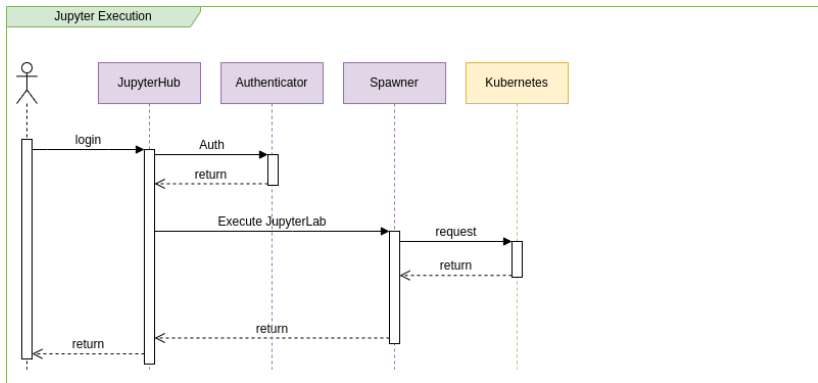


Figura: Jupyter – Diagrama de Sequencia

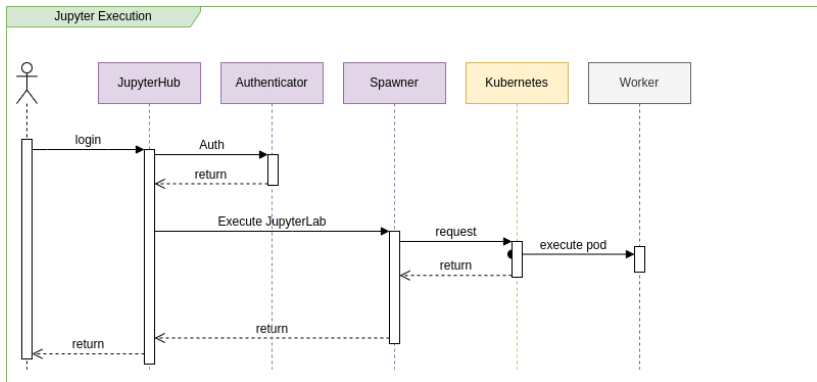


Figura: Jupyter – Diagrama de Sequencia

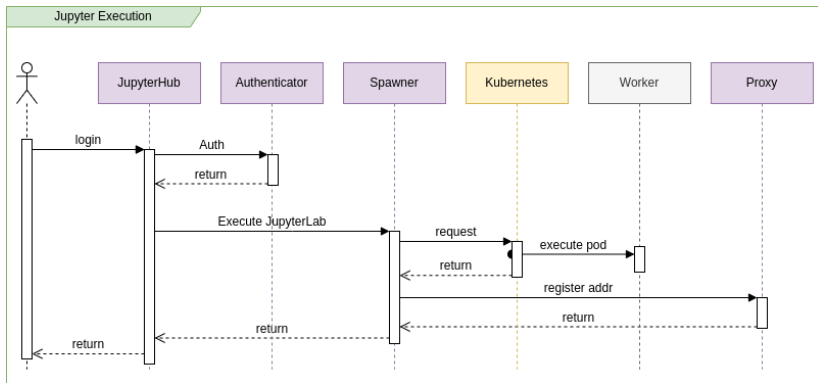


Figura: Jupyter – Diagrama de Sequencia

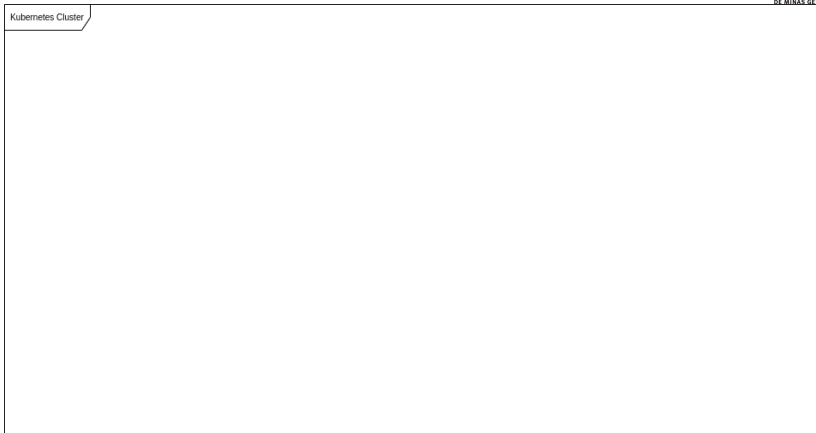


Figura: Jupyter - Diagrama de Deploy

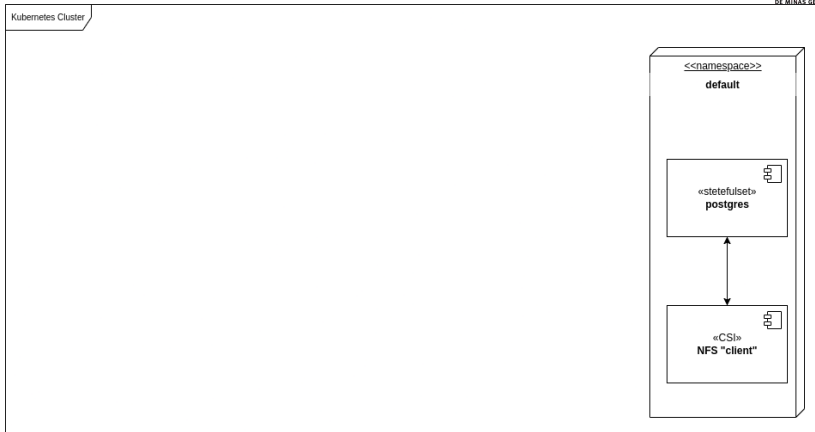


Figura: Jupyter - Diagrama de Deploy

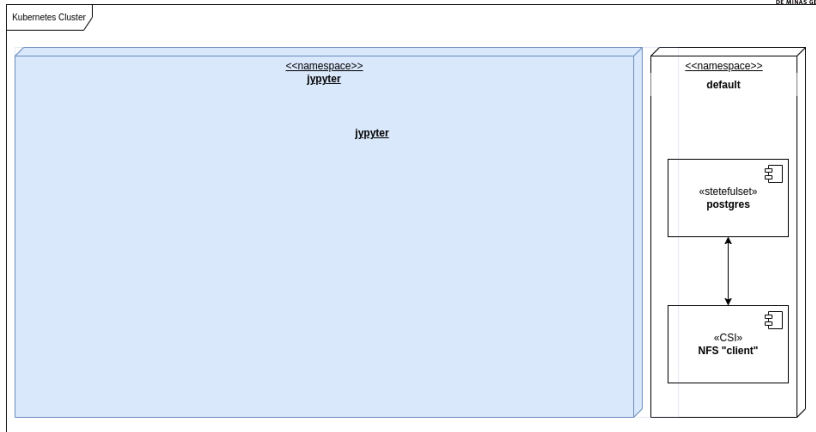


Figura: Jupyter - Diagrama de Deploy

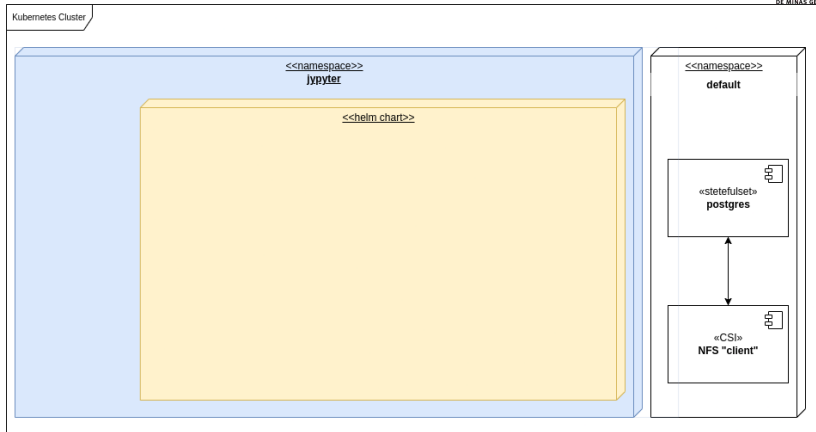


Figura: Jupyter - Diagrama de Deploy

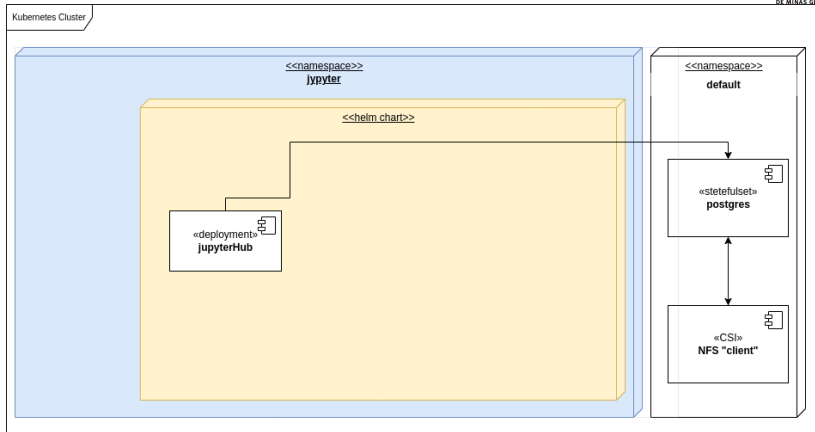


Figura: Jupyter - Diagrama de Deploy

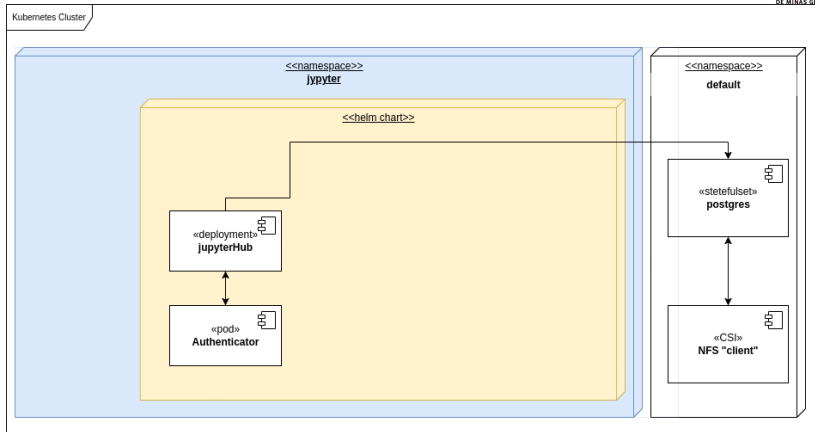


Figura: Jupyter - Diagrama de Deploy

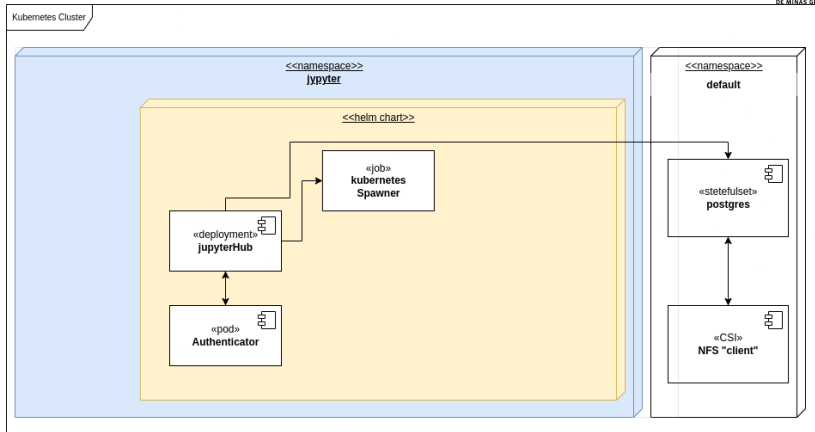


Figura: Jupyter - Diagrama de Deploy

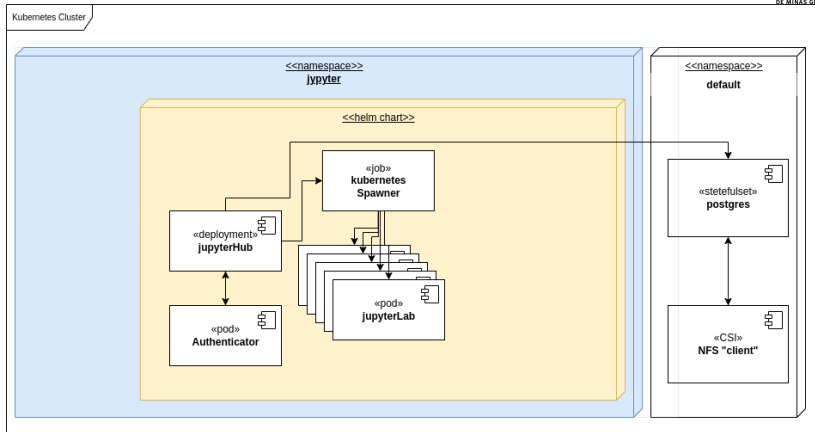


Figura: Jupyter - Diagrama de Deploy

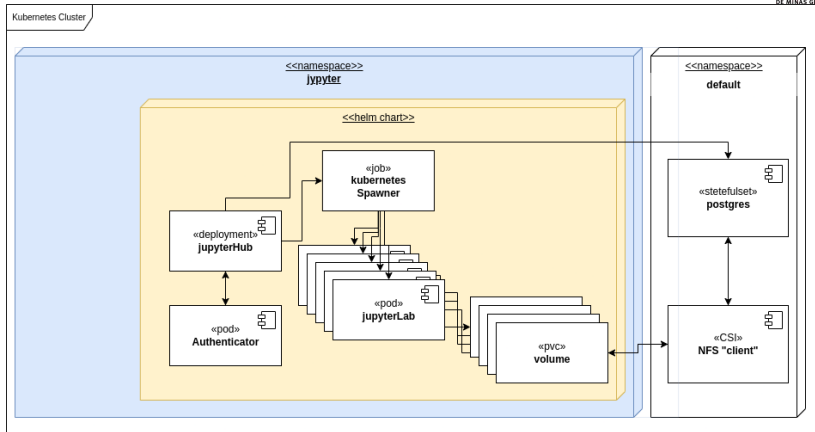


Figura: Jupyter - Diagrama de Deploy

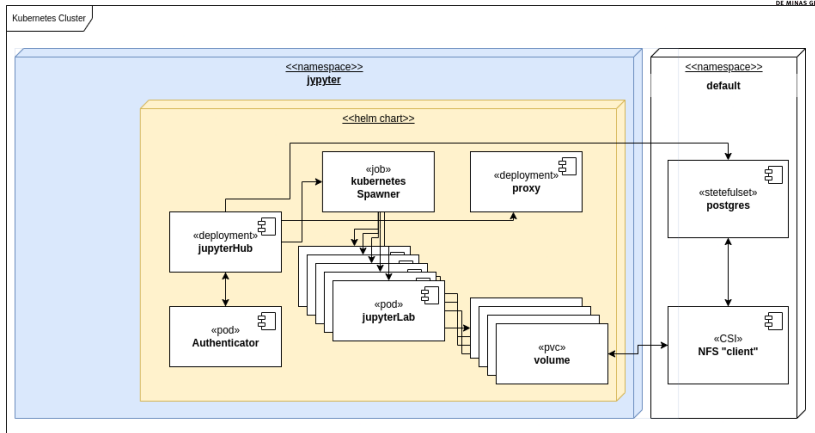


Figura: Jupyter - Diagrama de Deploy

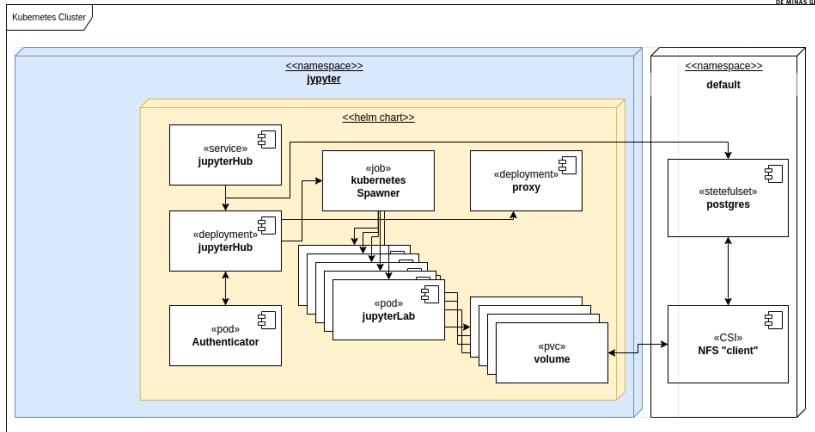


Figura: Jupyter - Diagrama de Deploy

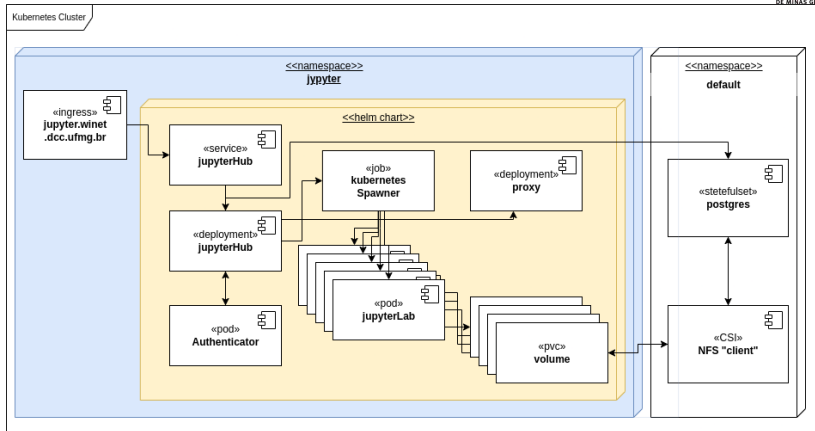


Figura: Jupyter - Diagrama de Deploy

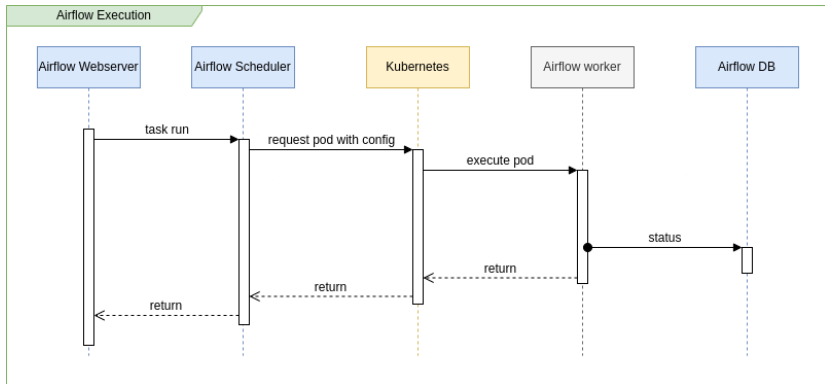


Figura: Airflow - Diagrama de Sequencia

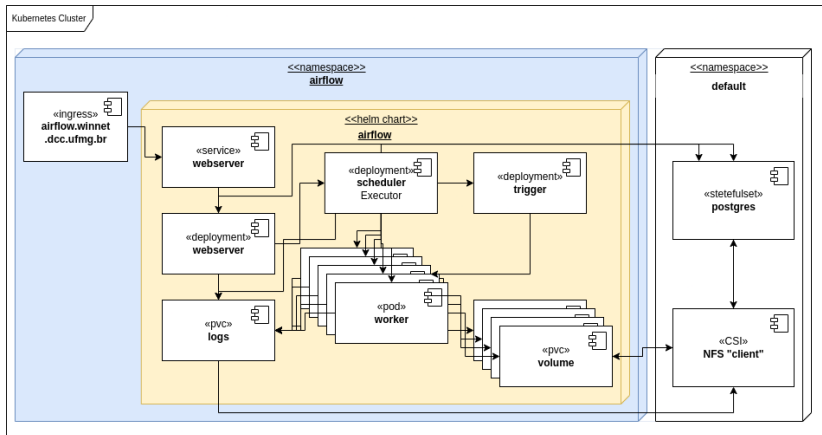


Figura: Ariflow - Diagrama de Deploy

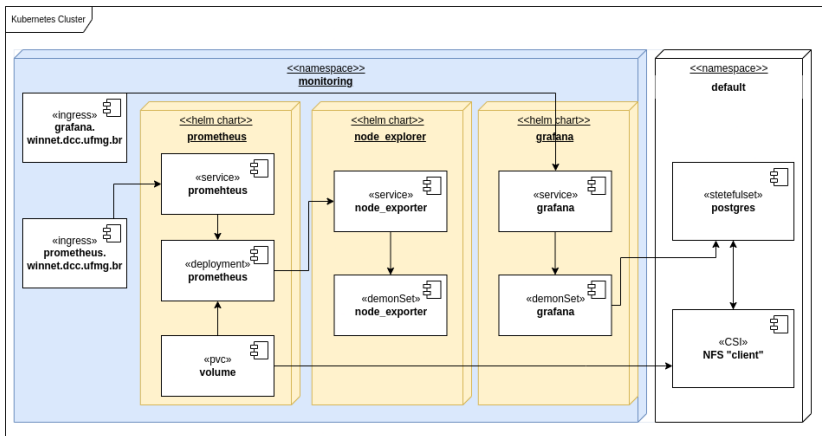


Figura: Monitoramento - Diagrama de Deploy

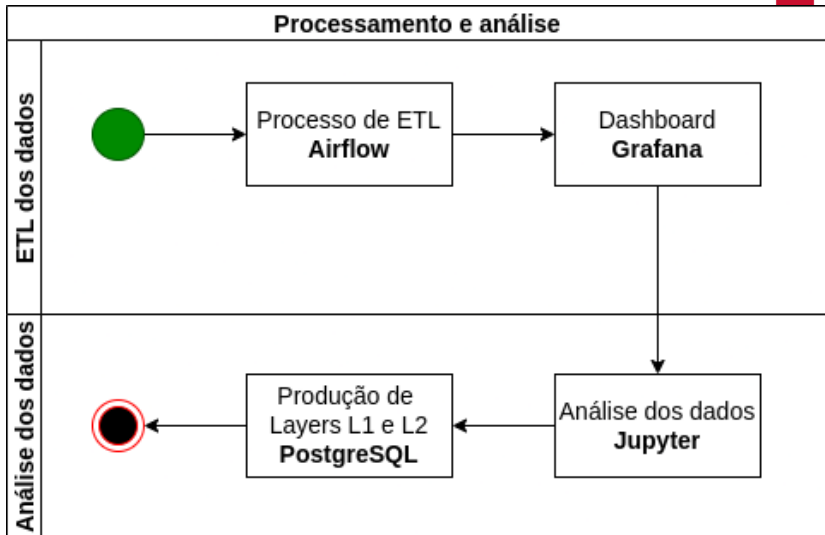


Figura: Fluxo de Process - Diagrama de Fluxo

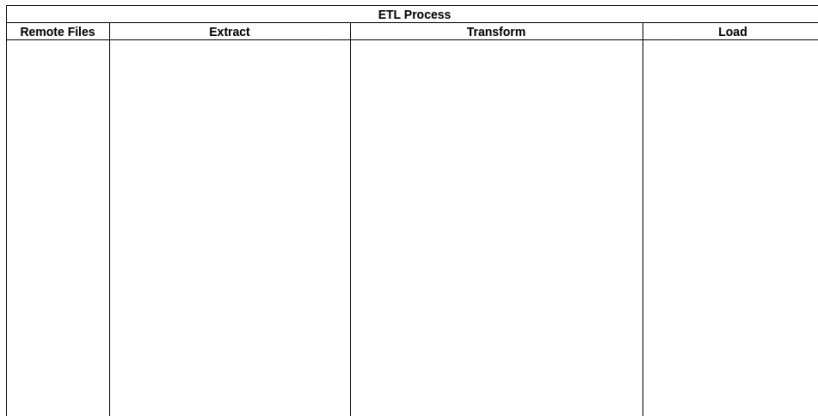


Figura: Processo ETL - Diagrama de Fluxo

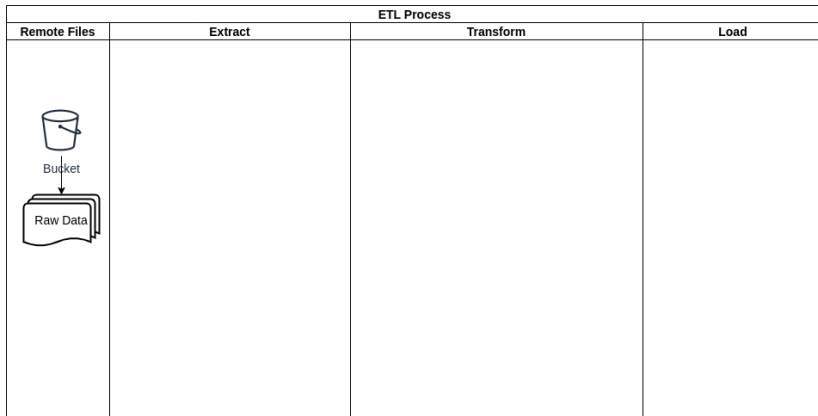


Figura: Processo ETL - Diagrama de Fluxo

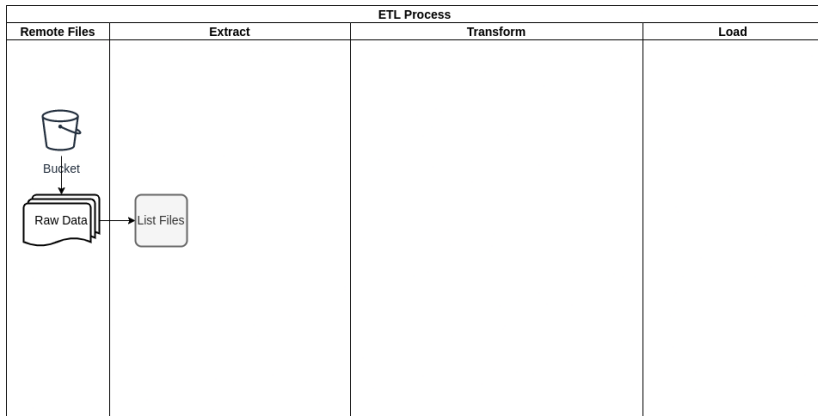


Figura: Processo ETL - Diagrama de Fluxo

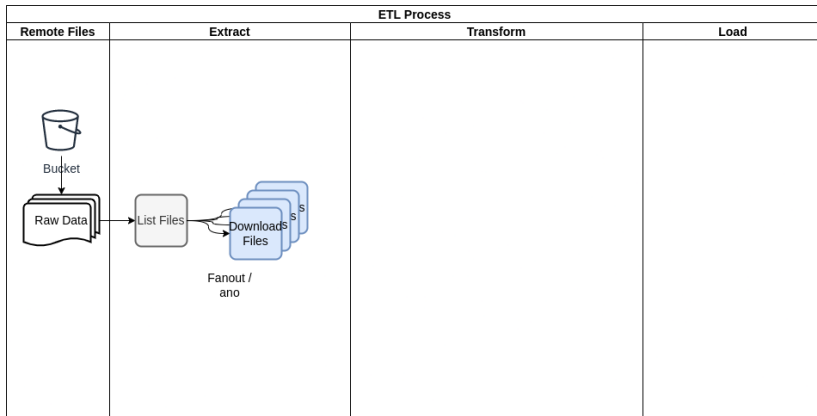


Figura: Processo ETL - Diagrama de Fluxo

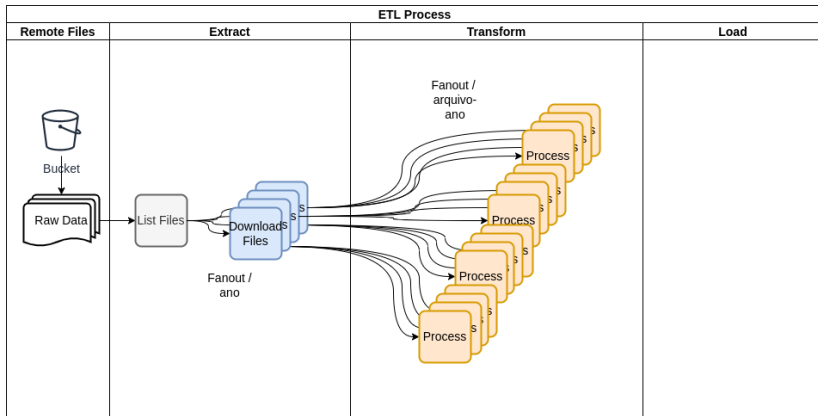


Figura: Processo ETL - Diagrama de Fluxo

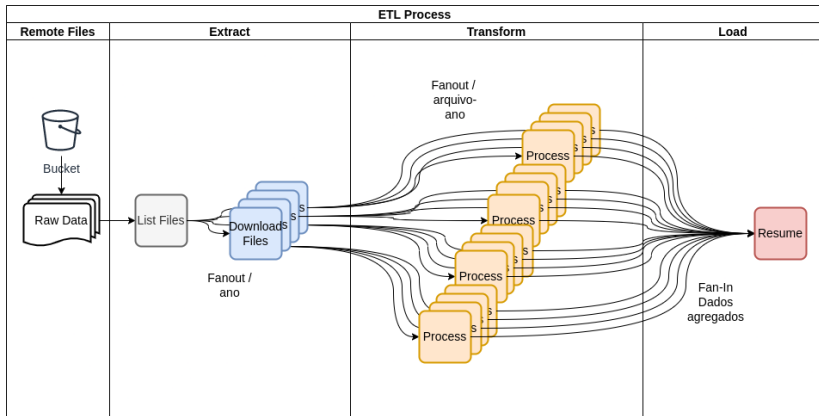


Figura: Processo ETL - Diagrama de Fluxo

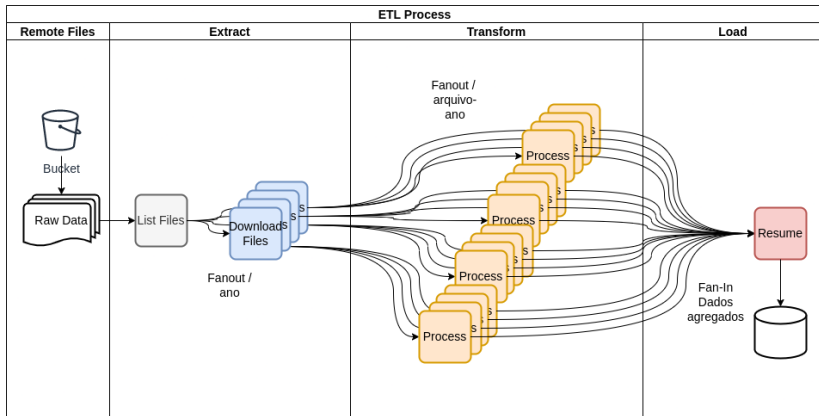


Figura: Processo ETL - Diagrama de Fluxo

Resultados e discussões

- Provisionamento
 - Tempo de configuração inicial
 - ▶ cloud-init: 2h (possível redução se utilizado imagens em rede)
 - Tempo de configuração cluster
 - ▶ automação de configuração Ansible: **20 – 35m**
 - ▶ Helm (deploy aplicação) + Terraform (orquestração de deploy): **10 – 25m**
- Execução dos job (**2GB** de RAM e **1CPU**, 90 pods):
 - Tempo ingestão dos dados: **≈ 53m**








<input type="checkbox"/>	 EDA_Industrializados_201401.csv	653.0 MB
<input type="checkbox"/>	 EDA_Industrializados_201402.csv	623.3 MB
<input type="checkbox"/>	 EDA_Industrializados_201403.csv	666.2 MB
<input type="checkbox"/>	 EDA_Industrializados_201404.csv	693.3 MB
<input type="checkbox"/>	 EDA_Industrializados_201405.csv	737.0 MB
<input type="checkbox"/>	 EDA_Industrializados_201406.csv	701.4 MB
<input type="checkbox"/>	 EDA_Industrializados_201407.csv	720.6 MB

Figura: S3 Lista de arquivos

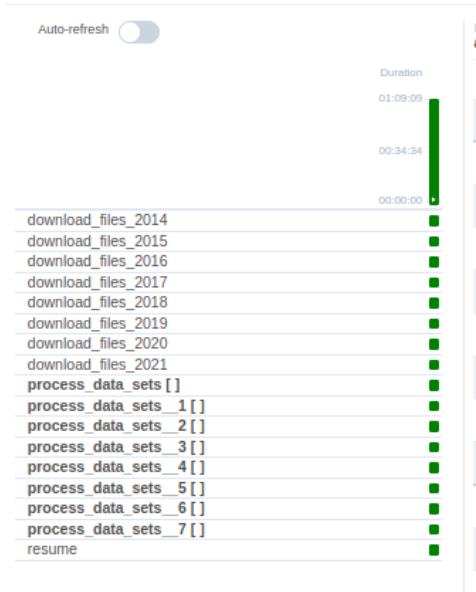


Figura: Relatório de orquestração

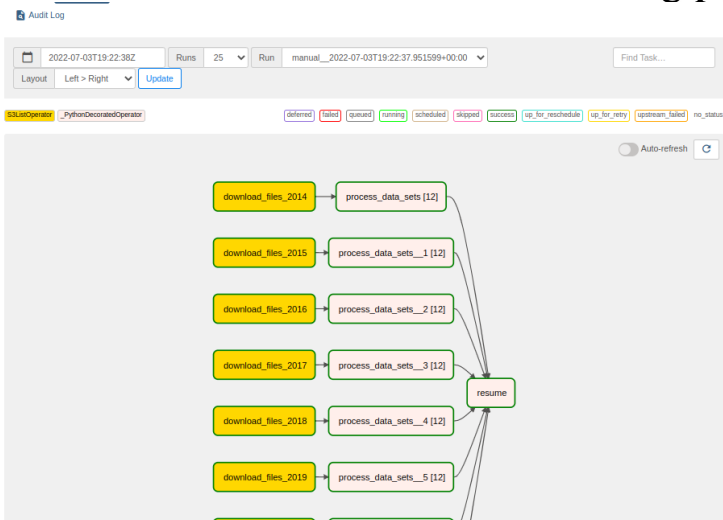


Figura: grafo DAG

1 Contexto e Motivação

2 Justificativa

3 Objetivo

4 Revisão de literatura

5 Método

6 Resultados

- Provisionamento
- Configuração
- Resultados do Monitoramento
- Resultados das Análises

7 Conclusão

DAG

azitromicina_consumption

DAG Details

DAG Runs Summary

Total Runs Displayed	10
■ Total success	2
■ Total failed	8
First Run Start	2022-07-02, 23:45:08 UTC
Last Run Start	2022-07-03, 04:15:45 UTC
Max Run Duration	00:53:11
Mean Run Duration	00:13:27
Min Run Duration	00:01:20

DAG Summary

Total Tasks	17
S3ListOperators	8
_PythonDecoratedOperators	9

Figura: Relatório de Orquestração

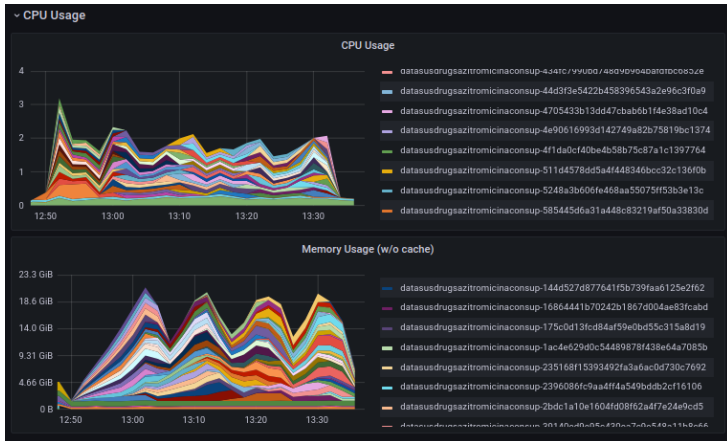


Figura: Monitoramento execução

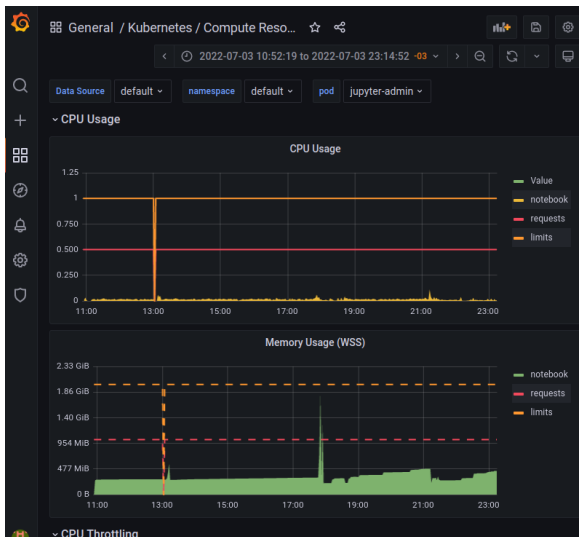


Figura: Monitoramento Jupyter

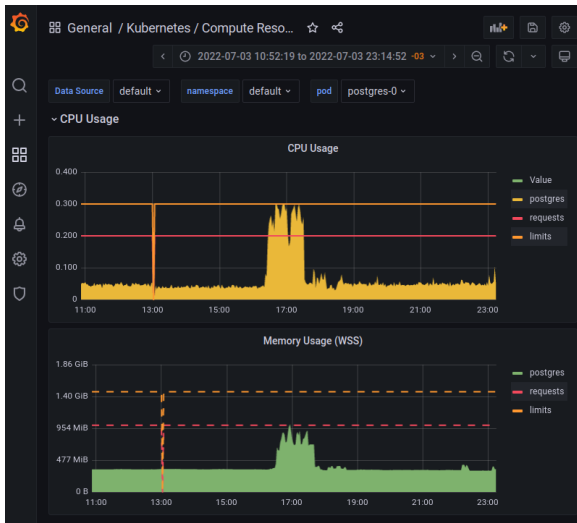


Figura: Monitoramento Postgres

1 Contexto e Motivação

2 Justificativa

3 Objetivo

4 Revisão de literatura

5 Método

6 Resultados

- Provisionamento
- Configuração
- Resultados do Monitoramento
- Resultados das Análises

7 Conclusão

Resultados das Análises

- Total de prescrições 95.345.640
- Aumento de prescrições de +32, 1% (2014-2020)
- Prescrições por 1000*hab*: +26, 59% (2014-2020)
- Regiões:
 - Sudeste: 47, 44% { *MG* : 13, 17, *SP* : 24, 76% }
 - Sul: 22, 47% { *RS* : 12, 49% }
- Estados destaque para aumento: Minas Gerais (+125, 38%), Rondônia (+191, 73% e Roraima (+168, 27%))

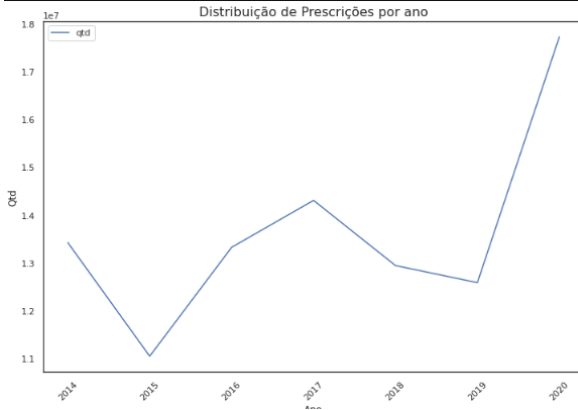


Figura: Prescrição por ano

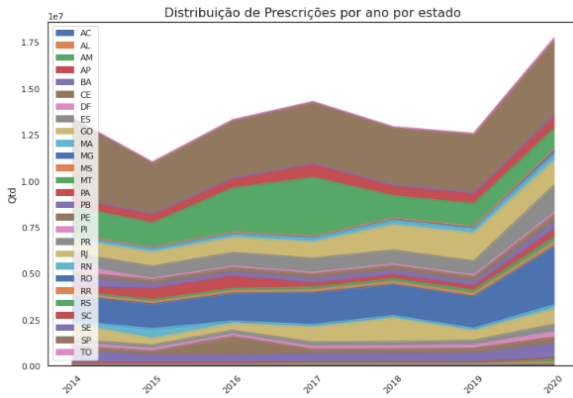


Figura: Prescrições por por ano por UF

Resultados das Análises

Relevância estatística (p valor) para correlação por τ de Kendall:

UF	τ	p valor
MT	1.0	0.003
RJ	1.0	0.003
RN	1.0	0.003
RO	1.0	0.003
TO	0.87	0.017

Conclusão

- Entendimento da complexidade dos fatores considerados no processo de decisão em saúde
- Análise dos impactos sociais-econômicos relativos a restrição orçamentária na ciência
- Seleção de tecnologias com base em requisitos e restrições
- Stack de tecnologia de mercado (maior suporte e melhores práticas)
- Desenho de uma estratégia de extração de informações em saúde
- Avaliação da viabilidade de uso de clusters de baixo custo na processamento de dados
- Interdisciplinariedade, especificidade e especialidade
- Produção de conhecimento de suporte prático
- Viés político nas decisões em saúde.
- Observabilidade dos dados em saúde

Conclusão

Para trabalho futuro visa-se a otimização de estratégia de dimensionamento de recursos, avaliação comparativa de outras tecnologias e técnicas para abordar o problema de processamento paralelo e distribuído. Ainda sugere-se, baseado nos resultados desse trabalho, discutir formas de recrutamento de computadores para o *cluster* de outros laboratórios, de maneira a criar elasticidade para cargas de trabalho ainda mais extensas.

Referências I

OBRIGADO
:)