

CHIP arquia:

<https://plataformadegestao.arqia.com.br/>

login: luanmw0@gmail.com

senha: Nuphi@123

banco de dados:

influxdb: <http://179.106.217.150:8086/>

admin/admin12345

grafana: <http://179.106.217.150:3000/>

admin/Nuphi@123

minio s3: <http://179.106.217.150:9001/>

admin/Nuphi@123

servidor do projeto:

ssh chdt@179.106.217.150

Nuphi@123

arquiterura das pastas

-> como funciona o processo de coleta de dados ate o influx

- o que o arquivo "extract.py" faz:

basicamente puxa o arquivo ".MF4" do bucket do minio, esse arquivo converte o arquivo .MF4 em um decimal enviando para a pasta "convertido-decimal", e depois faz a concatenação de todos os arquivos baixados, e converditos para decimal e envia a concatenação para a pasta "asammdf"

- o que o arquivo "dataframe.py" faz:

ele é responsável de enviar os dados concatenados da pasta asammdf para o influx, basicamente é isso, nele eu configuro o bucket do influx que eu desejo armazenar os dados.

Após o extract.py e dataframe.py rodar, ele limpa os arquivos gerados nas pastas "convertido-decimal" e "asammdf"

obs: importante lembrar que ele não realiza o download dos .MF4 que já foram feitos, pois implementei um checkpoint no arquivo extract.py onde ele chega se aquele arquivo decimal já foi concatenado, e no arquivo "dataframe.py" ele chega se o arquivo concatenado também já foi enviado. E com isso eu elimino o retrabalho de envio de dados, e não fico com arquivos gigantes que iriam demorar muito para ser enviados para o bucket no influx.

foi otimizado todo o processo no crontab -e

o arquivo dessa automatização é

/home/chdt/sniffer/automacao-sniffer/run_pipeline.sh

e possível ver se o cron está fazendo o processo corretamente nesse log:

```
tail -f /home/chdt/sniffer/logs/cron.log
```

```
tail -f /home/chdt/sniffer/logs/extract.log
```

```
tail -f /home/chdt/sniffer/logs/dataframe.log
```