Foi utilizado o Ubuntu 22.04 LTS para este desafio.

Após a instalação em uma máquina virtual, foram utilizados os comandos:

Sudo apt update && apt upgrade

POSTGRESQL:

Para instalação do PostgresSQL:

Sudo apt install postgresql

Já realizei a alteração da senha do usuário postgres:

Sudo -u postgres psql

ALTER USER postgres WITH PASSWORD ‘123’

Não estava no desafio, mas preferi utilizar o pgAdmin4 para administrar o PostgreSQL:

pgAdmin4

As instruções foram tiradas do site pgadmin.org

# Install the public key for the repository (if not done previously):

curl -fsS https://www.pgadmin.org/static/packages\_pgadmin\_org.pub | sudo gpg --dearmor -o /usr/share/keyrings/packages-pgadmin-org.gpg

# Create the repository configuration file:

sudo sh -c 'echo "deb [signed-by=/usr/share/keyrings/packages-pgadmin-org.gpg] https://ftp.postgresql.org/pub/pgadmin/pgadmin4/apt/$(lsb\_release -cs) pgadmin4 main" > /etc/apt/sources.list.d/pgadmin4.list && apt update'

# Install for desktop mode only:

sudo apt install pgadmin4-desktop

Preferi utilizar a versão desktop para não ter que fazer configurações no Apache.

Após isso fiz a configuração do server no pgAdmin4:

Server: localhost

User: postgres

Senha: 123

Com isso já é o suficiente para que eu tenha acesso total pelo pgAdmin.

Agora será feito a instalação do git, para versionamento e documentação.

Pentaho PDI

Foi baixado do site Hitachi ventara o pdi-ce-9.4.0.0-343, está versão é necessários alguns pacotes para fazer a instalação, segue o que foi feito para realizar a instalação dos pacotes:

Foi adicionado na sources.list este repositório

deb <http://cz.archive.ubuntu.com/ubuntu> bionic main universe

após isso:

sudo apt install libwebkitgtk-1.0-0 && libcanberra-gtk-module && libcanberra-gtk3-module

Também foi instalado o java:

sudo apt install default-jdk

E configurado a variável de ambiente do java:

nano ~/.bashrc

export JAVA\_HOME=/usr/lib/jvm/default-java

export PATH=$PATH:$JAVA\_HOME

também já aproveitei para fazer o unzip do pdi:

unzip /home/luhf/Downloads/pdi-ce-9.4.0.0.zip -d /home/luhf/pdi

Configurando a variável de ambiente do pdi:

export PDI\_HOME=/home/luhf/pdi/data-integration

export PATH=$PATH:$PDI\_HOME

Também fiz uma configuração para que o PDI fosse mostrado nos meus aplicativos:

nano ~/.local/share/applications/spoon.desktop

Dentro do arquivo aberto:

[Desktop Entry]

Type=Application

Name=Pentaho Data Integration

GenericName=ETL tool

Comment=Pentaho Data Integration (a.k.a Kettle)

Exec=/home/luhf/pdi/data-integration/spoon.sh

Icon=/home/luhf/pdi/data-integration/spoon.png

Terminal=false

Categories=Development;DataProcessing;

Com isso, toda a preparação do ambiente do desafio foi feita.

O processo de criação dos bancos de dados solicitados:

Sudo -i -u postgres psql

CREATE DATABASE banco\_a;

CREATE DATABASE banco\_b;

Fiz a transferência dos arquivos backup\_a.dmp e backup\_b.dmp para a pasta /tmp, para não ter problemas de permissão ao tentar fazer o restore.

Para o restore foi utilizado os comandos:

sudo -u postgres pg\_restore banco\_a /tmp/backup\_a.dmp

sudo -u postgres pg\_restore banco\_b /tmp/backup\_b.dmp

Também foi feito os arquivos .sql para o versionamento no github:

sudo -u postgres pg\_dump banco\_a /home/luhf/systeck/systeck.pdi/backup\_a.sql

sudo -u postgres pg\_dump banco\_b /home/luhf/systeck/systeck.pdi/backup\_b.sql

Após isso, todo o processo é documentado no github.

Com isso fica finalizado a configuração dos bancos de dados e vamos partir para o PDI:

No PDI, vou fazer uma transformation, e em input table, selecionar o banco\_a de onde eu quero tirar os dados da tabela consumos\_a.

Agora em output table, vou fazer a seleção do consumos\_b, para receber os dados do comsumos\_a.

Após executar a transformation, me gerou um log me informando que na tabela consumos\_a existe uma coluna chamada valor\_total e na consumos\_b está apenas total.

Alteração do nome da coluna:

Psql -u postgres -d banco\_b -h localhost

ALTER TABLE consumos\_b RENAME COLUMN valor TO valor\_total;

Feito a alteração, fui novamente executar a transformation, onde tive um novo log de erro me informando que a coluna status no consumos\_b estava com chacarter varying(1), porem na consumos\_a estava character varying (10).

Alteração do type da coluna:

Psql -u postgres -d banco\_b -h localhost

ALTER TABLE consumos\_b ALTER COLUMN status character varying(10);

Feito a alteração, novamente fui executar a transformation, onde tive sucesso na transferência de dados da tabela consumos\_a para consumos\_b.