PIETRO REY DE CAMARGO

**RELATÓRIO TÉCNICO ENGENHEIRO DE SOFTWARE**

**LISTA DE IMAGENS**

[Imagem 1 - Plano de trabalho proposto. Fonte: Autor 3](#_bookmark1)

[Imagem 2 - Exemplo uso do Zuul com Eureka. Fonte: Emmanuelneri 5](#_bookmark7)

[Imagem 3 - Diagrama de implantação. Fonte: Autor 7](#_bookmark15)

# SUMÁRIO

1. [PLANO DE TRABALHO 4](#_bookmark0)
2. [TECNOLOGIAS UTILIZADAS 6](#_bookmark2)
   1. [Docker e Docker Compose 6](#_bookmark3)
   2. [Java e Maven 6](#_bookmark4)
   3. [Apache Zookeeper 6](#_bookmark5)
   4. [Apache Kafka 6](#_bookmark6)
   5. [IntelliJ IDEA 7](#_bookmark8)
   6. [Prometheus 7](#_bookmark10)
   7. [Grafana 7](#_bookmark11)
3. [ARQUITETURA 8](#_bookmark12)
   1. [Diagrama de implantação 8](#_bookmark14)
4. [CÓDIGOS E IMAGENS DOS AMBIENTES . 9](#_bookmark19)
5. [REFERÊNCIAS 1](#_bookmark20)0

# PLANO DE TRABALHO

Tabela

Descrição gerada automaticamenteO plano de trabalho foi elaborado utilizando da metodologia 5W2H . A mesma é bastante utilizada como formato de checklist das atividades, prazos e responsabilidades que devem ser desenvolvidas além de possuir a função de definir o que será feito, porque, onde, quando e quanto tempo custará.Tornando assim uma maneria prática e eficiente de entender e acompanhar o andamento de uma tarefa.

Imagem 1 - Plano de trabalho do projeto. Fonte: Autor

Os três primeiros steps foram executados sem maiores problemas, porém com adicional de tempo no primeiro e terceiro, uma vez que em algumas partes tive de efetuar pesquisas de como efetuar da melhor maneira.

O step 4 foi feito com base na documentação do próprio Kafka, com isso não ocorreram problemas e foi possível testar outros comandos de tópicos.

A partir do step 5 até o step 7, foi necessário assistir alguns videos pois optei pelo desenvolvimento Java que é mais aderente ao cenário que a empresa tem, porém a aplicação para utilização com o Kafka exigia alguns conhecimentos fora do que eu tinha. Tive alguma dificuldade com relação aos parâmetros de conexão de rede dentro das aplicações que com algumas pesquisas foi superado.

No Step 8, após ter concluido todo o cenário basico do case, efetuei testes produzindo e consumindo mensagens no tópico para avaliar se ocorreriam problemas ou bugs, os mesmos não ocorreram.

Partindo para os adicionais, o Step 9 apresentou um desafio maior, pois não havia conhecimento sobre como eram utilizados as ferramentas bem como a maneira que elas iriam se comunicar com o Kafka, por isso, tive de efetuar diversas pesquisas para que encontrace uma maneira eficaz de aplicar a solução, o que demandou bastante tempo. O mesmo foi concluído com sucesso.

# TECNOLOGIAS UTILIZADAS

Segue as tecnologias utilizadas durante a criação da solução.

# Docker e Docker Compose

Para criação dos ambientes foram criadas imagens Linux CentOS utilizando o Docker e para funcionamento em conjunto buscando evitar erros e interligando os containers foi utilizado a ferramenta Docker Compose.

# Java e Maven

A linguagem utilizada foi o Java e Maven para desenvolver as aplicações de Consumidor e Produtor para melhor gerenciar as dependências e uma automatização de builds mais fácil.

# Apache Zookeeper

O Apache Zookeeper é um serviço centralizado que serve para manter informações de configurações e nommenclaturas entre serviços distribuídos, usualmente é usado em conjunto de outras plataformas para que possa ser efetuado a sincronização de configurações de diferentes clusters.

# Apache Kafka

Diagrama

Descrição gerada automaticamenteO Apache Kafka, ou apenas Kafka, é uma plataforma de streaming distribuído, e pode manipular milhares de eventos por dia, inicialmente utilizado como fila de mensagens, evoluiu para uma plataforma completa de streaming, servindo não apenas como message broker mas também podendo armazenar e procesar dados. (Imagem 2).

Imagem 2 - Exemplo uso do Zookeeper e Kafka. Fonte: Rodrigo Martins

# IntelliJ IDEA

Para desenvolvimento dessa aplicação foi utilizada o Intelli J que fornece total suporte para o desenvolvimento de aplicações Java e com um grande suporte a Maven, possibilitando assim melhor agilidade e praticidade durante o desenvolvimento.

# Prometheus

O Prometheus um sistema de monitoramento para serviços e um toolkit de alertas open-source, ele efetua suas funções coletando métricas em deteminados intervalos,avalia expressões de regras e exibe os seus resultados.

# Grafana

Grafana por sua vez, é uma plataforma que visualiza e analisa métricas por meio de gráficos, nele é possível criar dashboards dinamicos de acordo com a necessidade de monitoramento.

# ARQUITETURA

A solução desenvolvida consiste em reproduzir um cenário onde é possível produzir uma mensagem via aplicação a ser enviada para um tópico Kafka que será consumido via aplicação e obter monitoramento do fluxo de requisições e outros dados por dashboard.

# Diagrama de implantação

Diagrama

Descrição gerada automaticamente

Imagem 3 - Diagrama de implantação. Fonte: Autor

1. **CÓDIGOS E IMAGENS DOS AMBIENTES**

Os códigos das aplicações podem ser encontrados diretamente no github através da url: <https://github.com/PietroRey/case-kafka>

Quanto a aquisição do container docker pode ser encontrado no perfil pietrorey do Docker Hub ou no link abaixo:

* Kafka: <https://hub.docker.com/r/pietrorey/kafka>
* Zookeeper: <https://hub.docker.com/r/pietrorey/zookeeper>
* Prometheus: <https://hub.docker.com/r/pietrorey/prometheus>
* Procons: [https://hub.docker.com/r/pietrorey/procons](https://hub.docker.com/r/pietrorey/prometheus)

# REFERÊNCIAS

APACHE KAFKA QUICKSTART EM:

<https://kafka.apache.org/quickstart>

APACHE KAFKA : INTRODUÇÃO EM :

<https://atitudereflexiva.wordpress.com/2020/03/05/apache-kafka-introducao/>

KAFKA MONITORING USING PROMETHEUS, MADHUR AHUJA, EM:

[https://www.metricfire.com/blog/kafka-monitoring-using-prometheus/#Plotting-the-monitoring-visualization-on-Grafana](https://www.metricfire.com/blog/kafka-monitoring-using-prometheus/%23Plotting-the-monitoring-visualization-on-Grafana)

MONITORANDO UM CLUSTER KAFKA COM FERRAMENTAS OPEN SOURCE, ÁLVARO BACELAR, EM:

<https://medium.com/@alvarobacelar/monitorando-um-cluster-kafka-com-ferramentas-open-source-a4032836dc79>

HOW TO INSTALL AND CONFIGURE AN APACHE ZOOKEEPER CLUSTER ON UBUNTU 18.04, SHASHANK TIWARI, EM:

<https://www.digitalocean.com/community/tutorials/how-to-install-and-configure-an-apache-zookeeper-cluster-on-ubuntu-18-04>