Apache Spark Streaming

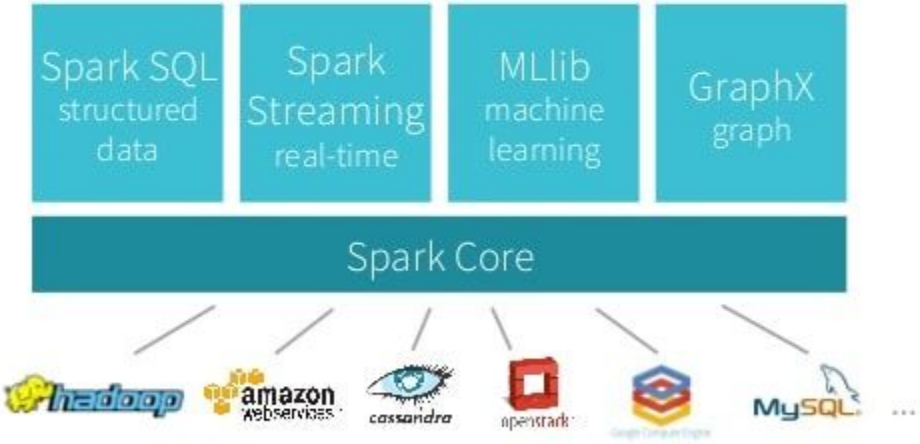
Daniel Monteiro Valério
Gabriel Mace dos Santos Ferreira
Marcus Vinícius Souza Fernandes

Spark

Spark

Ferramenta unificada para o processamento de dados em larga escala. A partir de suas ferramentas principais é possível realizar o processamento de grafos, machine learning e computação incremental e processamento em stream.

A General Engine



databricks

Spark Streaming

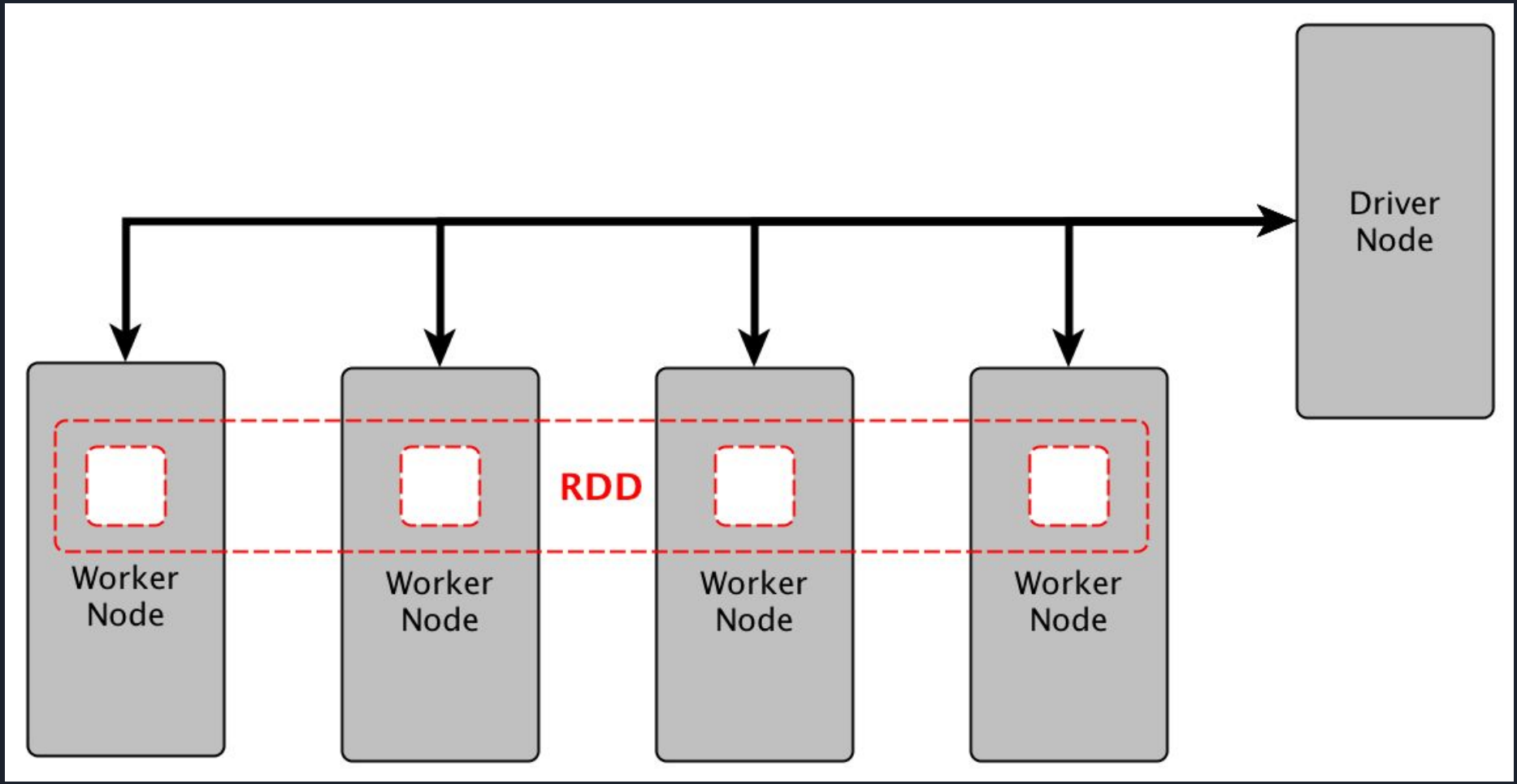
Spark Streaming

Spark Streaming permite o processamento de dados em stream à partir de diversas fontes. É possível processar os dados com funções de alto nível como: map, reduce, join e window.

Por sua vez, os dados podem ser armazenados em sistemas de arquivo, bancos de dado e tabelas ao vivo.

Arquitetura Spark


input data stream
Spark Streaming
batches of input data
Spark Engine
batches of processed data


Driver
Node
RDD
Worker
Node
Worker
Node
Worker
Node
Worker
Node

Casos de Uso

Tentativas

Diferentes API’s

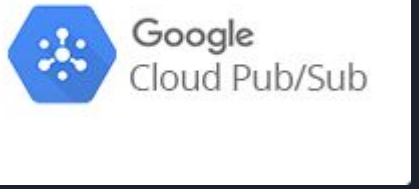
1°

2°

Aplicação

Tecnologias

Google Cloud



Tecnologias

1 - Dataproc: O Dataproc é um serviço totalmente gerenciado para executar o Apache Spark.

2 - Pub/Sub: Fornece uma vasta análise de streaming e pipelines de integração de dados para gerir e distribuir dados.

3 - Datastore: O Datastore é um banco de dados NoSQL.

4 - Cloud Functions: É uma solução de computação sem servidor do Google para criar aplicativos com base em eventos.

Overview

Visão geral do código

O código a seguir configura o stream do Pub/Sub para processar os dados de entrada.

```
33      // [START stream_setup]
34      val sparkConf = new SparkConf().setAppName("TrendingHashtags")
35      val ssc = new StreamingContext(sparkConf, Seconds(slidingInterval.toInt))
36
37      // Set the checkpoint directory
38      val yarnTags = sparkConf.get("spark.yarn.tags")
39      val jobId = yarnTags.split(",").filter(_.startsWith("dataproc_job")).head
40      ssc.checkpoint(checkpointDirectory + '/' + jobId)
41
42      // Create stream
43      val messagesStream: DStream[String] = PubsubUtils
44        .createStream(
45          ssc,
46          projectID,
47          None,
48          "tweets-subscription",  // Cloud Pub/Sub subscription for incoming tweets
49          SparkGCPCredentials.builder.build(), StorageLevel.MEMORY_AND_DISK_SER_2)
50        .map(message => new String(message.getData(), StandardCharsets.UTF_8))
```

Visão geral do código

O app do Spark extrai e conta todas as hashtags usando este canal simples.

```
24  // [START extract]
25  private[demo] def extractTrendingTags(input: RDD[String]): RDD[Popularity] =
26    input.flatMap(_.split("\\s+")) // Split on any white character
27      .filter(_.startsWith("#")) // Keep only the hashtags
28      // Remove punctuation, force to lowercase
29      .map(_.replaceAll("[,.!?:;]", "").toLowerCase)
30      // Remove the first #
31      .map(_.replaceFirst("^#", ""))
32      .filter(!_.isEmpty) // Remove any non-words
33      .map((_, 1)) // Create word count pairs
34      .reduceByKey(_ + _) // Count occurrences
35      .map(r => Popularity(r._1, r._2))
36      // Sort hashtags by descending number of occurrences
37      .sortBy(r => (-r.amount, r.tag), ascending = true)
```

Visão geral do código

O app do Spark salva as 10 principais hashtags em uma nova linha do banco de dados no Datastore.

```
29  // [START convert_identity]
30  private[demo] def convertToEntity(hashtags: Array[Popularity],
31                                    keyFactory: String => KeyFactory): FullEntity[IncompleteKey] = {
32    val hashtagKeyFactory: KeyFactory = keyFactory("Hashtag")
33
34    val listValue = hashtags.foldLeft[ListValue.Builder](ListValue.newBuilder())(
35      (listValue, hashTag) => listValue.addValue(convertToDatastore(hashtagKeyFactory, hashTag))
36    )
37
38    val rowKeyFactory: KeyFactory = keyFactory("TrendingHashtags")
39
40    FullEntity.newBuilder(rowKeyFactory.newKey())
41      .set("datetime", Timestamp.now())
42      .set("hashtags", listValue.build())
43      .build()
44  }
```

Implantação

Passos

1 - Ativação dos serviços (Pub/Sub, Dataproc e Cloud functions) e do banco de dados (Datastore).

2 - Clonar projeto.

3 - Criar tópico e assinatura do Pub/Sub.

4 - Criar conta no dataproc e fornecer acesso aos serviços.

5 - Criar o cluster.

6 - Fazer o download e ou atualizar as dependências.

7 - Criar e iniciar o Job.

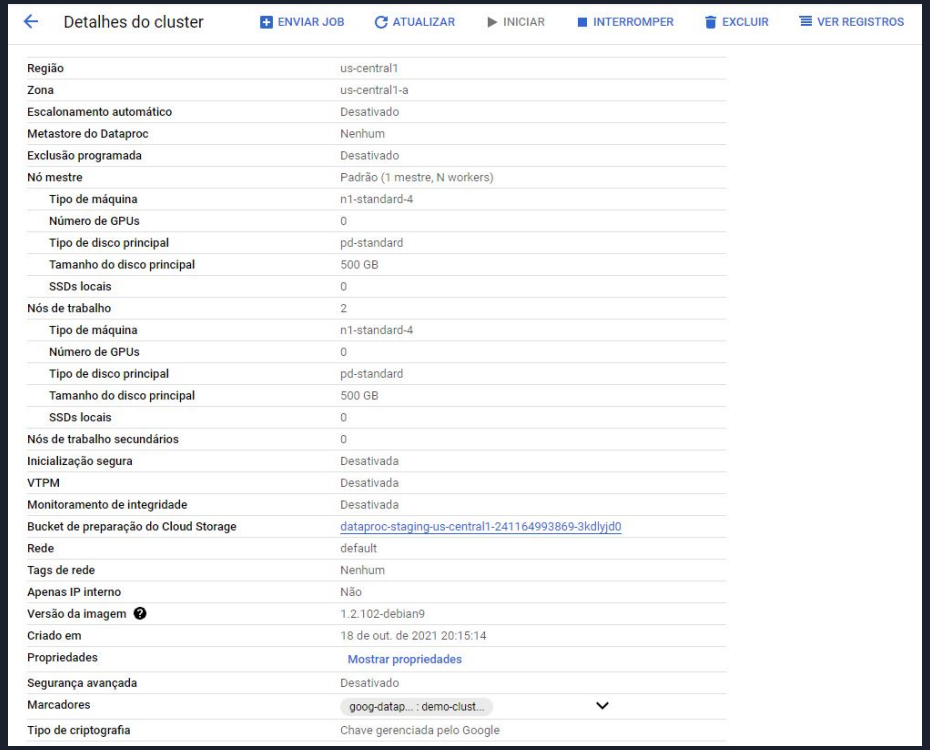
8 - Criar ambiente em python, instalar dependências e ativar o tweet-generator.

9 - Visualizar página web com os top trendings.

Configurações do Cluster

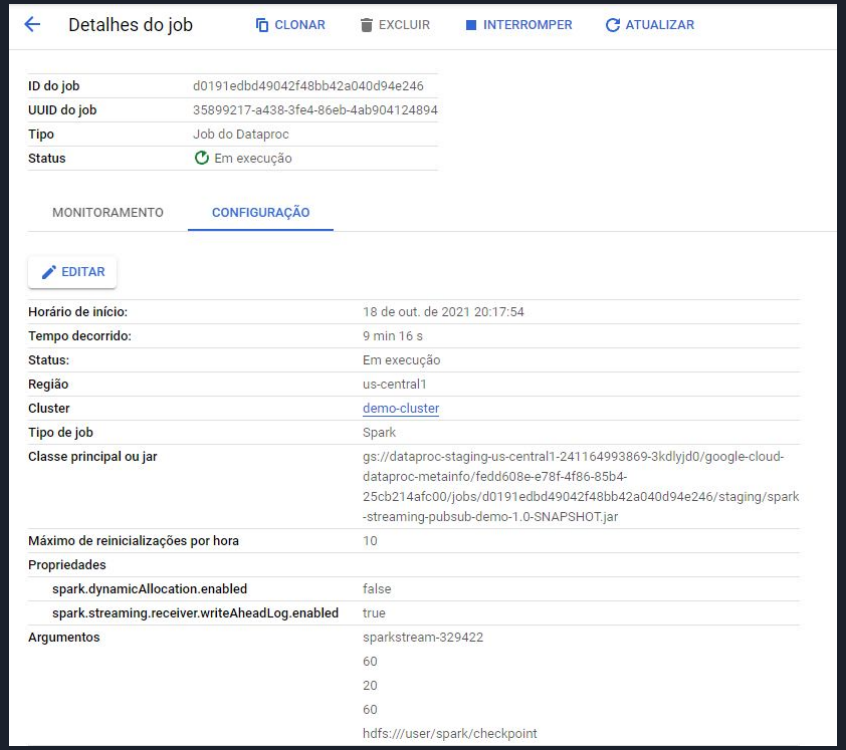
```
gcloud dataproc clusters create demo-cluster \
    --region=us-central1 \
    --zone=us-central1-a \
    --scopes=pubsub,datastore \
    --image-version=1.2 \
    --service-account="$SERVICE_ACCOUNT_NAME@$PROJECT.iam.gserviceaccount.com"
```

Configurações do Cluster



Região	us-central1
Zona	us-central1-a
Escalonamento automático	Desativado
Metastore do Dataproc	Nenhum
Exclusão programada	Desativado
Nó mestre	Padrão (1 mestre, N workers)
Tipo de máquina	n1-standard-4
Número de GPUs	0
Tipo de disco principal	pd-standard
Tamanho do disco principal	500 GB
SSDs locais	0
Nós de trabalho	2
Tipo de máquina	n1-standard-4
Número de GPUs	0
Tipo de disco principal	pd-standard
Tamanho do disco principal	500 GB
SSDs locais	0
Nós de trabalho secundários	0
Inicialização segura	Desativada
VTPM	Desativada
Monitoramento de integridade	Desativada
Bucket de preparação do Cloud Storage	dataproc-staging-us-central1-241164993869-3kdlyjd0
Rede	default
Tags de rede	Nenhum
Apenas IP interno	Não
Versão da imagem	1.2.102-debian9
Criado em	18 de out. de 2021 20:15:14
Propriedades	Mostrar propriedades
Segurança avançada	Desativado
Marcadores	goog-datap... : demo-clust...
Tipo de criptografia	Chave gerenciada pelo Google

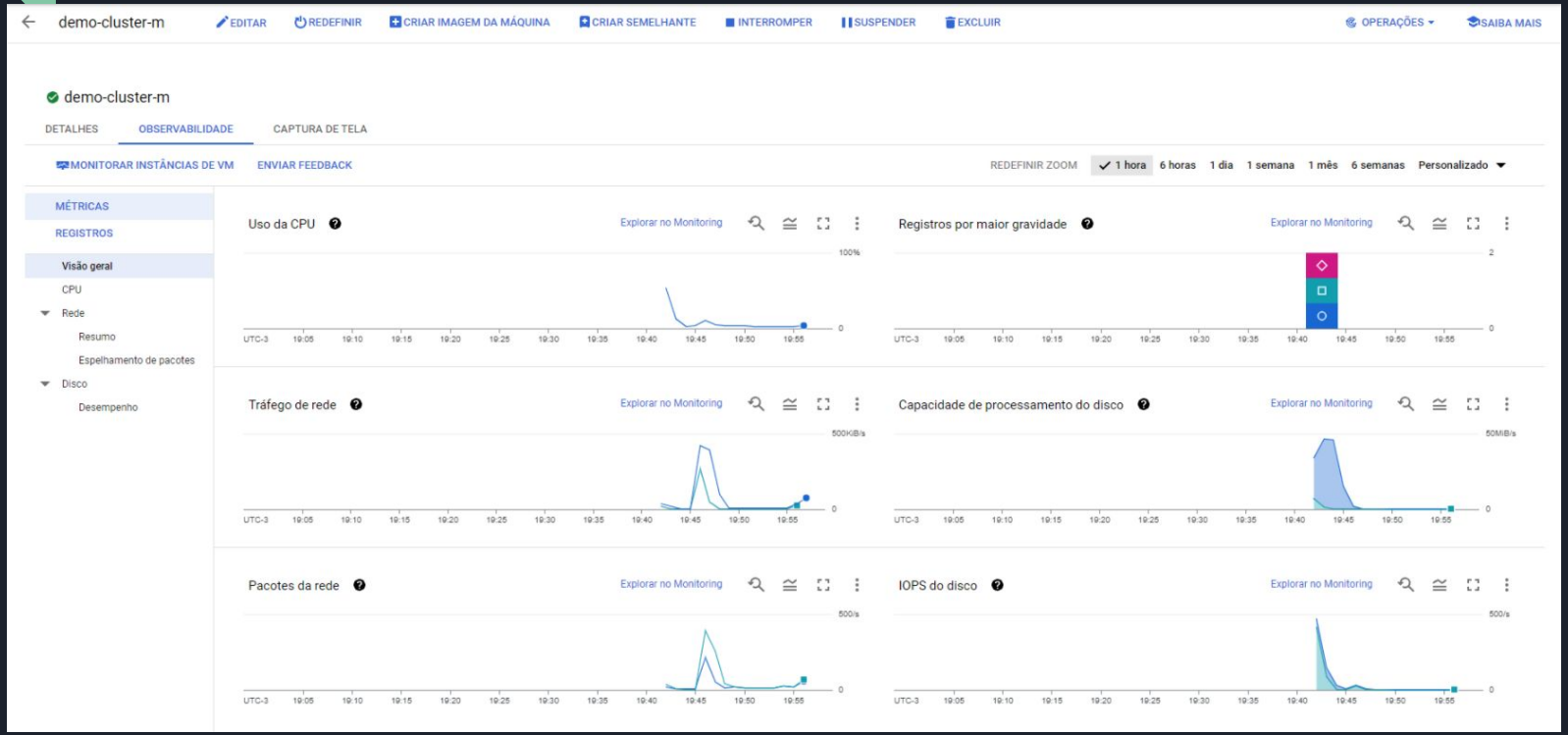
Configurações do Job



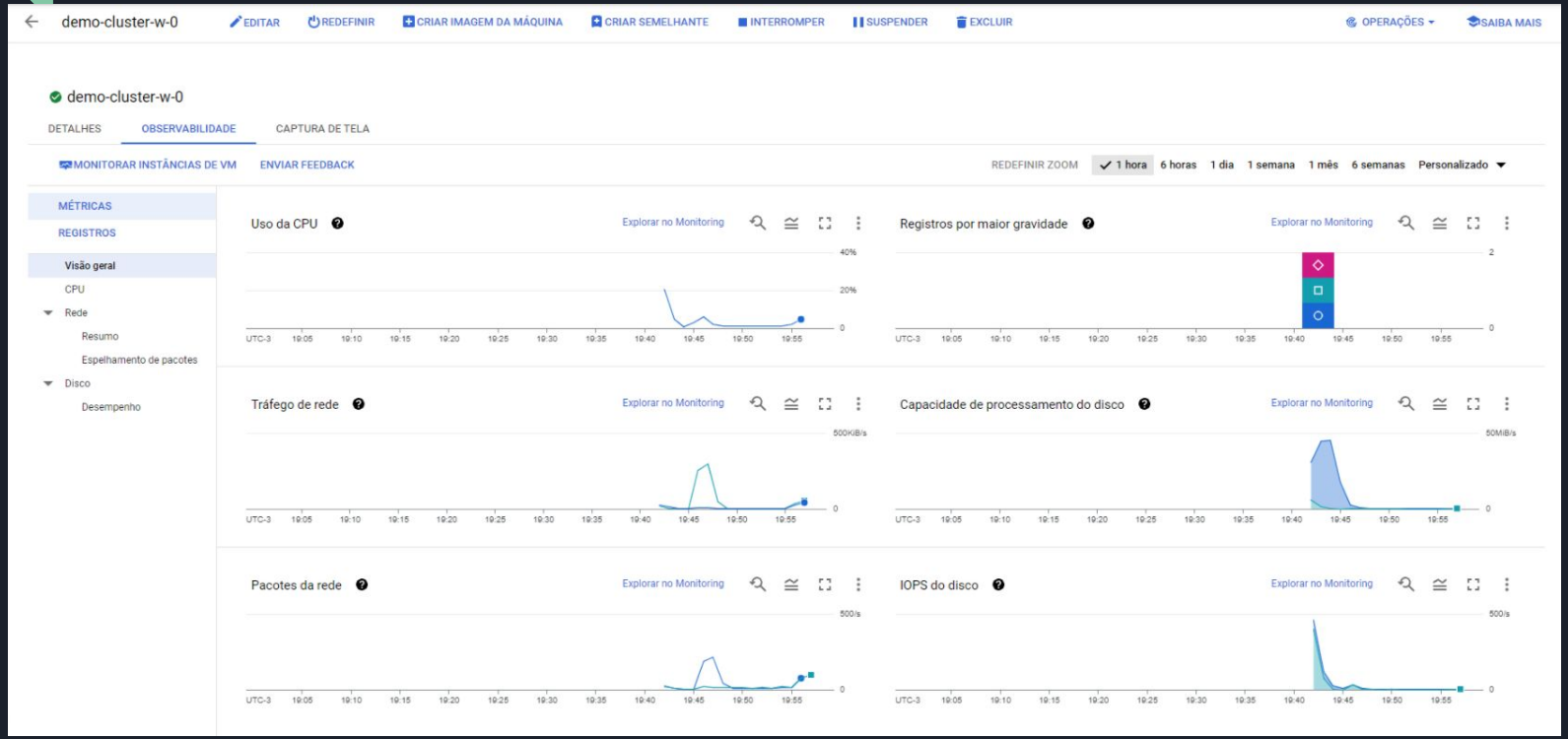
Output do Job

Funcionamento

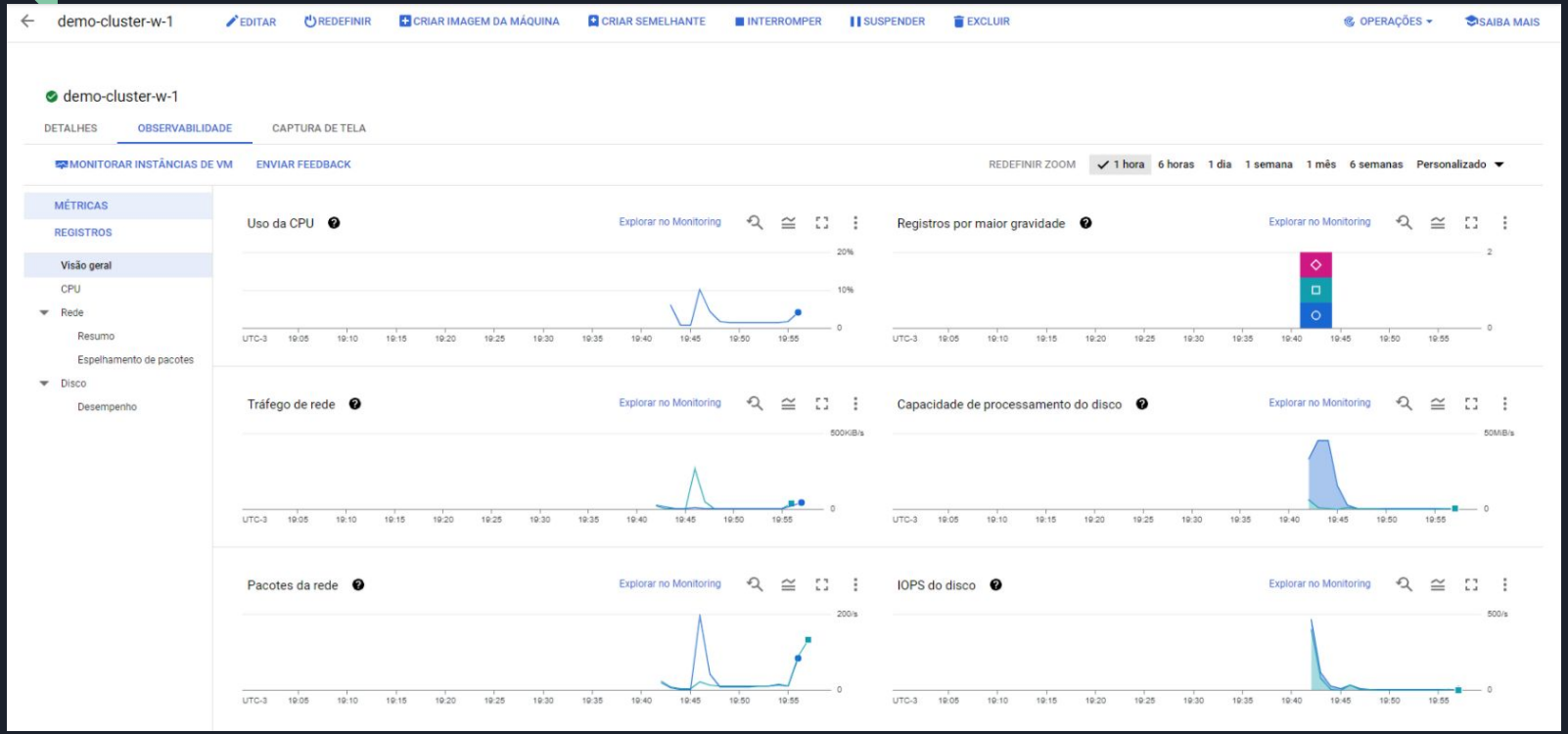
Processamento mestre



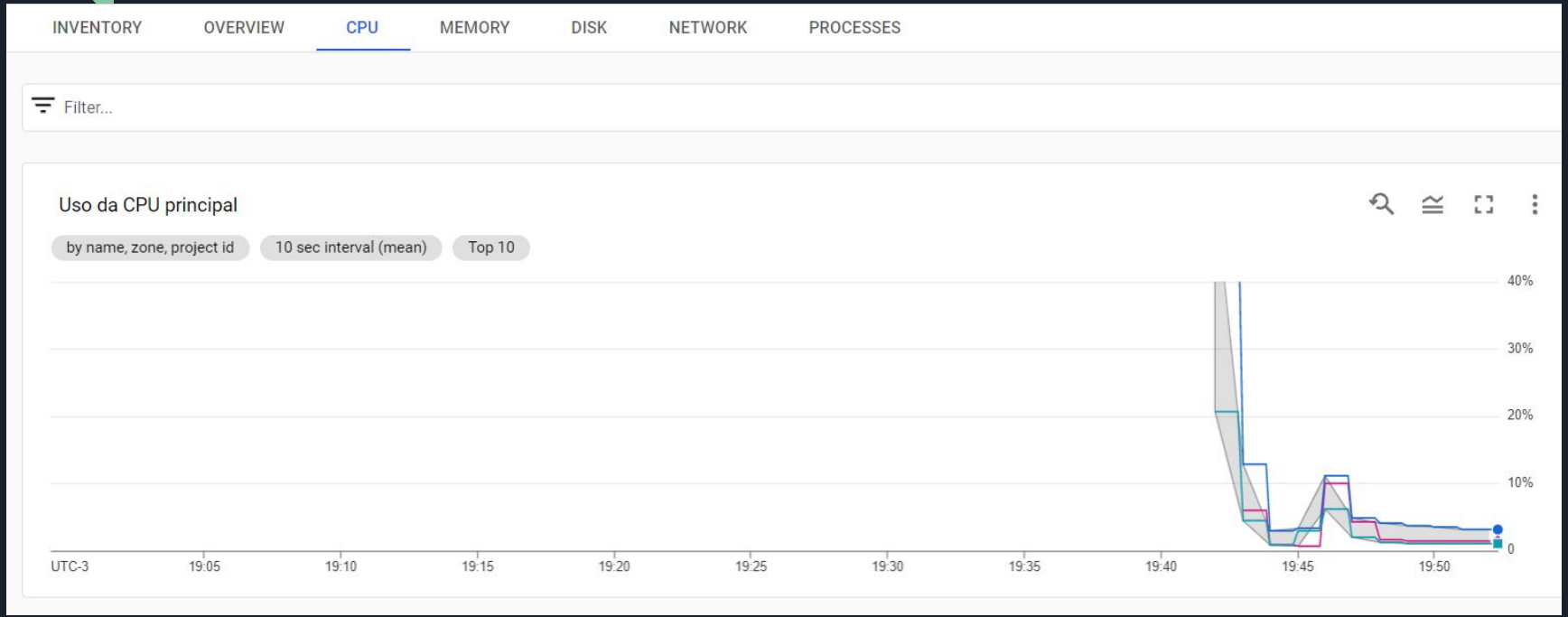
Processamento worker 0



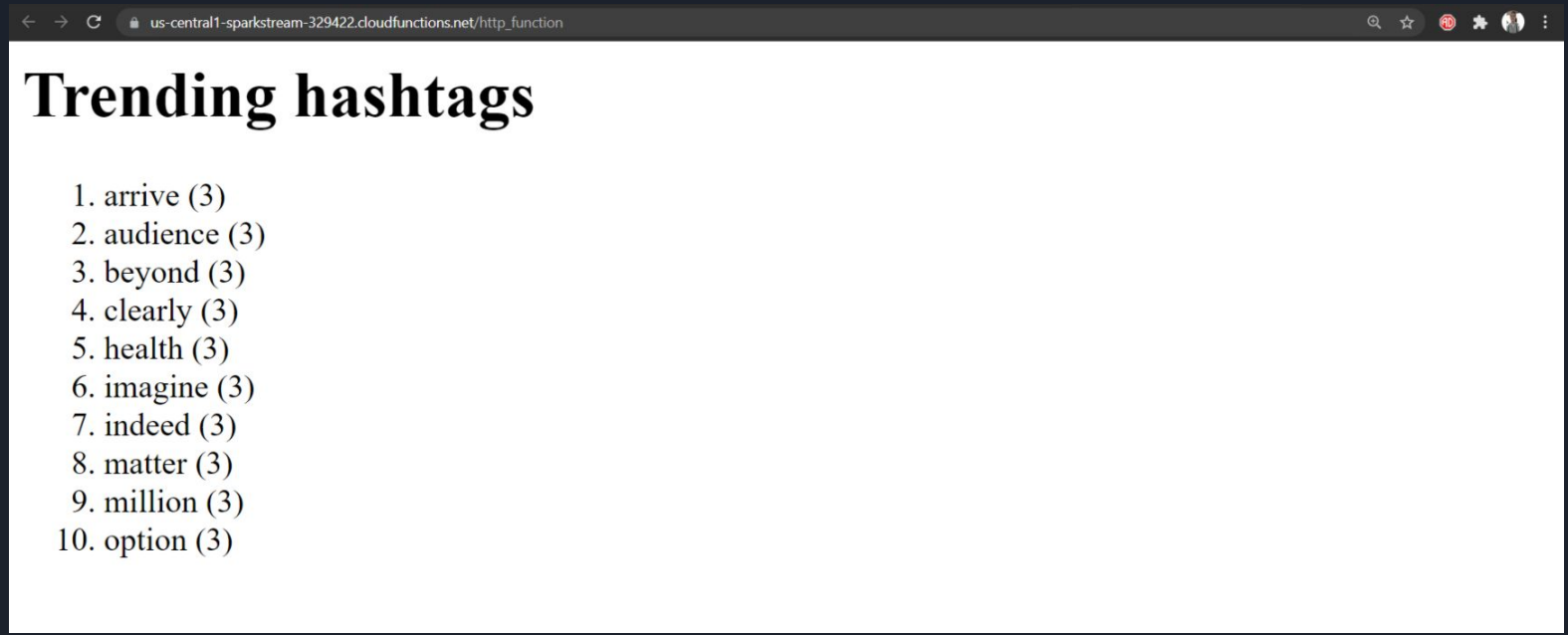
Processamento worker 1



Processamento geral



Output web


us-central1-sparkstream-329422.cloudfunctions.net/http_function
Trending hashtags
1. arrive (3)
2. audience (3)
3. beyond (3)
4. clearly (3)
5. health (3)
6. imagine (3)
7. indeed (3)
8. matter (3)
9. million (3)
10. option (3)

Muito obrigado!