**Tarefa 1: preparação**

Você estará executando um simulador de sensores da VM de treinamento. No Lab 1 você configura manualmente os componentes Pub / Sub. Neste laboratório, vários desses processos são automatizados.

Abra o terminal SSH e conecte-se à VM de treinamento

1. No Console, no **menu Navegação** ( ), clique em **Compute Engine** > **instâncias de VM** .
2. Localize a linha com a instância chamada **training\_vm** .
3. No canto direito, em "Conectar", clique em **SSH** para abrir uma janela de terminal.
4. Neste laboratório, você entrará comandos CLI no **training\_vm** .

Verificar se a inicialização está completa

1. O **training\_vm** está instalando o software em segundo plano. Verifique se a configuração está concluída, verificando se o diretório a seguir existe. Se não existir, aguarde alguns minutos e tente novamente.

ls /training

Aguarde até que a configuração esteja concluída antes de prosseguir. Você pode verificar a instalação do maven com o **mvn -version** e o JDK com **java -version** .

Copie arquivos

1. Um repositório foi baixado para a VM. Copie o repositório para o seu diretório inicial.

cp -r /training/training-data-analyst/ .

Definir variáveis ​​de ambiente

1. No terminal SSH **training\_vm** , digite o seguinte:

source /training/project\_env.sh

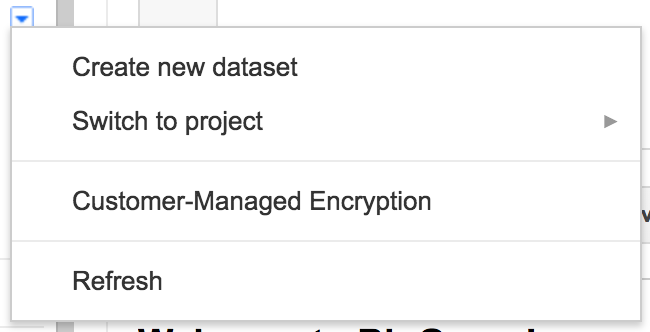
Este script define as variáveis ​​de ambiente **$ DEVSHELL\_PROJECT\_ID** e **$ BUCKET** .

**Tarefa 2: criar um conjunto de dados do BigQuery e um repositório do Cloud Storage**

O pipeline do Dataflow será criado posteriormente e será gravado em uma tabela neste conjunto de dados.

Crie um conjunto de dados do BigQuery

1. Abra a interface da web do BigQuery. No **menu Navegação** ( ), clique em **BigQuery** . Clique em **Ir para a interface clássica** . O console do BigQuery será aberto em uma nova guia do navegador. Agora, selecione seu projeto Qwiklabs clicando na seta para baixo ao lado de Recursos Qwiklabs, selecionando **Alternar para projeto> Seu projeto Qwiklab** .
2. À direita do nome do projeto, clique na **seta azul** . Escolha **Criar novo conjunto de dados** .



1. Na caixa de diálogo "Criar Conjunto de Dados", para **ID do Conjunto** de **Dados** , digite **demonstrações** e clique em **OK** .

Verifique o balde do Cloud Storage

Um bucket já deve existir com o mesmo nome que o ID do projeto.

1. No console, no **menu Navegação** ( 7a91d354499ac9f1.png), clique em **Armazenamento** > **Navegador** .
2. Observe os seguintes valores:

|  |  |
| --- | --- |
| **Propriedade** | **Valor**  (digite o valor ou selecione a opção conforme especificado) |
| **Nome** | **<O mesmo que o ID do projeto>** |
| **Classe de armazenamento padrão** | **[x] Regional** |
| **Localização** | **<Sua localização>** |

**Tarefa 3: Simular dados do sensor de tráfego no Pub / Sub**

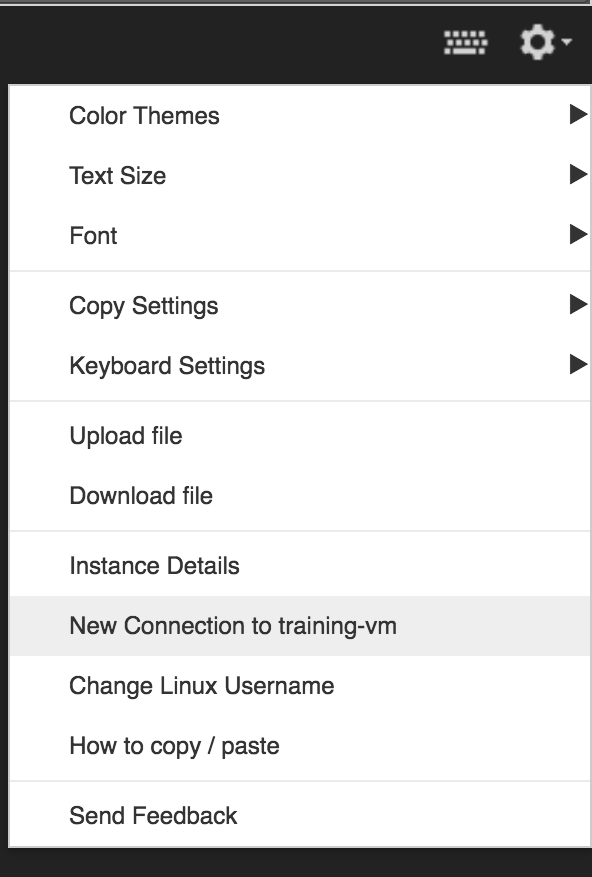
1. No terminal SSH **training\_vm** , inicie o simulador do sensor. O script lê dados de amostra de um arquivo csv e os publica no Pub / Sub.

/training/sensor\_magic.sh

Este comando irá enviar 1 hora de dados em 1 minuto. Deixe o script continuar a rodar no terminal atual.

Abra um segundo terminal SSH e conecte-se à VM de treinamento

1. No canto superior direito do terminal SSH **training\_vm** , clique no botão em forma de engrenagem ( 9649d58acf1c4e06.png) e selecione **New Connection to training-vm** no menu suspenso. Uma nova janela de terminal será aberta.



1. A nova sessão de terminal não terá as variáveis ​​de ambiente necessárias. Execute o seguinte comando para defini-los.
2. No novo terminal SSH **training\_vm** , digite o seguinte:

source /training/project\_env.sh

Visualizar inscrições

1. No console, no **menu Navegação** ( 7a91d354499ac9f1.png), clique em **Pub / Sub** > **Tópicos** .
2. Examine a linha para o **nome** do tópico para o tópico **sandiego** . Observe que as **Assinaturas** estão atualmente em 0.

**Tarefa 4: iniciar o pipeline do Dataflow**

Verificar se a API do Google Cloud Dataflow está ativada para este projeto

1. Volte para a guia do navegador para o Console. Na barra de pesquisa superior, insira a **API do Dataflow** . Isso levará você à página, **menu "Navegação"> APIs e serviços> Painel de controle> Google Dataflow API** . Ele mostrará uma informação de status ou lhe dará a opção de **ativar** a API.
2. Se necessário, **ative** a API.
3. Volte para o segundo terminal SSH **training\_vm** . Mude para o diretório deste laboratório.

cd ~/training-data-analyst/courses/streaming/process/sandiego

1. Identifique o script que cria e executa o pipeline do Dataflow.

cat run\_oncloud.sh

1. Copie e cole o seguinte URL em uma nova guia do navegador para visualizar o código-fonte no Github.

https://github.com/GoogleCloudPlatform/training-data-analyst/blob/master/courses/streaming/process/sandiego/run\_oncloud.sh

1. O script requer três argumentos: **id do projeto** , **nome do intervalo** , **nome da classe**

Um quarto argumento opcional é **opções** . O argumento de **opções** discutido posteriormente neste laboratório.

|  |  |
| --- | --- |
| **id do projeto** | <seu ID do projeto> |
| **nome do balde** | <seu nome de balde> |
| **nome da classe** | <arquivo java que executa agregações> |
| **opções** | <opções> |

Existem 4 arquivos java que você pode escolher para **classname** . Cada um lê os dados de tráfego do Pub / Sub e executa diferentes agregações / cálculos.

1. Vá para o diretório java. Identifique o arquivo de origem **AverageSpeeds.java** .

cd ~/training-data-analyst/courses/streaming/process/sandiego/src/main/java/com/google/cloud/training/dataanalyst/sandiego

cat AverageSpeeds.java

O que o script faz?

Feche o arquivo para continuar. Você vai querer se referir a este código-fonte durante a execução do aplicativo. Assim, para facilitar o acesso, você abrirá uma nova guia do navegador e **exibirá** o arquivo **AverageSpeeds.java** no Github.

1. Copie e cole o seguinte URL em uma guia do navegador para visualizar o código-fonte no Github.

https://github.com/GoogleCloudPlatform/training-data-analyst/blob/master/courses/streaming/process/sandiego/src/main/java/com/google/cloud/training/dataanalyst/sandiego/AverageSpeeds.java

Deixe esta aba do navegador aberta. Você estará se referindo ao código-fonte em uma etapa posterior deste laboratório.

1. Volte para o terminal SSH **training\_vm** . Execute o pipeline do Dataflow para ler o PubSub e gravar no BigQuery.

cd ~/training-data-analyst/courses/streaming/process/sandiego

./run\_oncloud.sh $DEVSHELL\_PROJECT\_ID $BUCKET AverageSpeeds

Esse script usa o maven para criar um pipeline de streaming do Dataflow em Java.

Exemplo de conclusão bem-sucedida:

[INFO] ------------------------------------------------------------------------

[INFO] BUILD SUCCESS

[INFO] ------------------------------------------------------------------------

[INFO] Total time: 45.542 s

[INFO] Finished at: 2018-06-08T16:51:30+00:00

[INFO] Final Memory: 56M/216M

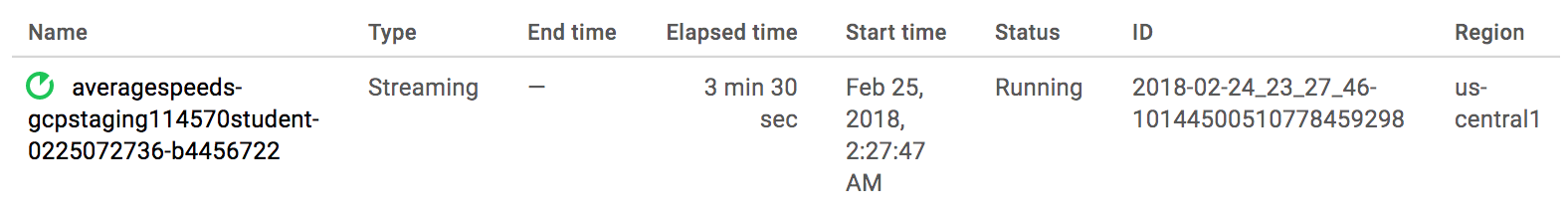
[INFO] ------------------------------------------------------------------------

**Tarefa 5: Explorar o pipeline**

Este pipeline do Dataflow lê mensagens de um tópico do Pub / Sub, analisa o JSON da mensagem de entrada, produz uma saída principal e grava no BigQuery.

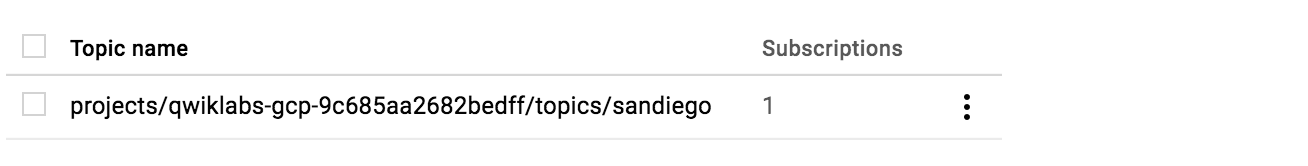
1. Volte para a guia do navegador para o Console. No **menu Navegação** ( 7a91d354499ac9f1.png), clique em **Fluxo de dados** e clique em seu trabalho para monitorar o progresso.

Exemplo:



1. Depois que o pipeline estiver em execução, clique no **menu Navegação** ( 7a91d354499ac9f1.png), clique em **Pub / Sub** > **Tópicos** .
2. Examine a linha para o **nome** do tópico para o tópico **sandiego** . Observe que o campo **Subscrições** está agora em 1.

Exemplo:



1. Volte ao **menu de navegação** ( 7a91d354499ac9f1.png), clique em **Dataflow** e clique no seu trabalho.
2. Compare o código na guia do navegador do Github, **AverageSpeeds.java** e o gráfico de pipeline na página do seu trabalho do Dataflow.
3. Encontre a etapa do pipeline **GetMessages** no gráfico e localize o código correspondente no arquivo **AverageSpeeds.java** . Esta é a etapa do pipeline que lê o tópico Pub / Sub. Cria uma coleção de Strings - que corresponde às mensagens Pub / Sub que foram lidas.

* Você vê uma assinatura criada?
* Como o código puxa mensagens do Pub / Sub?

1. Encontre a etapa do pipeline da **janela de tempo** no gráfico e no código. Nesta etapa do pipeline, criamos uma janela com duração especificada nos parâmetros do pipeline (janela deslizante, neste caso). Essa janela acumulará os dados de tráfego da etapa anterior até o final da janela e passará para as próximas etapas para transformações posteriores.

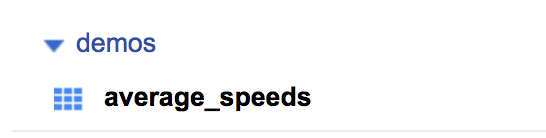
* Qual é o intervalo da janela?
* Com que frequência uma nova janela é criada?

1. Encontre os **BySensor** e **AvgBySensor** passos de dutos no gráfico, e em seguida, encontrar o trecho de código correspondente no arquivo AverageSpeeds.java. Esse **BySensor** faz um agrupamento de todos os eventos na janela pelo ID do sensor, enquanto o **AvgBySensor** calcula a velocidade média para cada agrupamento.
2. Encontre a etapa do pipeline **ToBQRow** no gráfico e no código. Esta etapa cria simplesmente uma "linha" com a média computada da etapa anterior junto com as informações da faixa.

Na prática, outras ações podem ser tomadas na etapa **ToBQRow** . Por exemplo, pode comparar a média calculada com um limite predefinido e registrar os resultados da comparação no Stackdriver Logging.

1. Encontre o **BigQueryIO.Write** no gráfico de pipeline e no código-fonte. Esta etapa grava a linha do pipeline em uma tabela do BigQuery. Como escolhemos a disposição de gravação **WriteDisposition.WRITE\_APPEND** , novos registros serão anexados à tabela.
2. Volte para a guia da web do BigQuery. Atualize seu navegador.
3. Na coluna da esquerda, abaixo da caixa de texto, encontre o nome do seu projeto e o conjunto de dados de demos que você criou. A pequena seta azul à esquerda deve estar ativa e clicar nela revelará a tabela **average\_speeds** .

Exemplo:



**Tarefa 6: Determinar taxas de transferência**

Uma atividade comum ao monitorar e aprimorar os pipelines do Dataflow é descobrir quantos elementos o pipeline processa por segundo, qual é o atraso do sistema e quantos elementos de dados foram processados ​​até o momento. Nesta atividade, você aprenderá onde, no Cloud Console, é possível encontrar informações sobre os elementos processados ​​e o tempo.

1. Volte para a guia do navegador para o Console. No **menu Navegação** ( 7a91d354499ac9f1.png), clique em **Fluxo de dados** e clique em seu trabalho para monitorar o progresso (ele terá seu nome de usuário no nome do pipeline).
2. Selecione o nó de pipeline **GetMessages** no gráfico e observe as métricas da etapa à direita.

* **O System Lag** é uma métrica importante para pipelines de streaming. Ele representa a quantidade de tempo que os elementos de dados estão aguardando para serem processados ​​desde que "chegaram" na entrada da etapa de transformação.
* **Elementos A** métrica **adicionada** sob coleções de saída informa quantos elementos de dados saíram desta etapa (para a etapa **Ler PubSub Msg** do pipeline, ela também representa o número de mensagens Pub / Sub lidas do tópico pelo conector I / O do Pub / Sub).

1. Selecione o nó da **janela de tempo** no gráfico. Observe como a métrica "Elementos adicionados" na etapa " Coleções de entrada" da **janela de tempo**corresponde à métrica "Elementos adicionados" nas coleções de saída da etapa anterior " **GetMessages"** .

**Tarefa 7: revisar a saída do BigQuery**

1. Retornar à interface da web do BigQuery.

Os dados e as tabelas de fluxo contínuo podem não aparecer imediatamente e o recurso Visualizar pode não estar disponível para os dados que ainda estão no buffer de fluxo. Se você clicar em **Visualizar** , verá a mensagem "Esta tabela possui registros no buffer de fluxo que podem não estar visíveis na visualização." Você ainda pode executar consultas para visualizar os dados.

1. Clique em **Compor Consulta** . Use a consulta a seguir para observar a saída do job do Dataflow. Substitua <PROJECTID> pelo seu ID do projeto. Ele está listado em Detalhes da Conexão no Qwiklabs.

SELECT \*

FROM [<PROJECTID>:demos.average\_speeds]

ORDER BY timestamp DESC

LIMIT 100

1. Encontre a última atualização para a tabela executando o seguinte SQL.

SELECT

MAX(timestamp)

FROM

[<PROJECTID>:demos.average\_speeds]

1. Use o Decorador de tabelas do BigQuery para ver os resultados nos últimos 10 minutos.

SELECT

\*

FROM

[<PROJECTID>:demos.average\_speeds@-600000]

ORDER BY

timestamp DESC

Use o **BigQuery Invalid Snapshot Time** , tente reduzir o 600000 para 100000.

**Tarefa 8: Observar e entender o escalonamento automático**

Observe como o Dataflow dimensiona o número de trabalhadores para processar o backlog de mensagens Pub / Sub recebidas.

1. Volte para a guia do navegador para o Console. No **menu Navegação** ( 7a91d354499ac9f1.png), clique em **Fluxo de dados** e clique no trabalho do pipeline.
2. Examine o painel de **resumo** do **trabalho** à direita e revise a seção **Escalonamento automático** . Quantos trabalhadores estão sendo usados ​​atualmente para processar mensagens no tópico Pub / Sub?
3. Clique em **Ver mais histórico** e analise quantos trabalhadores foram usados ​​em diferentes momentos no tempo durante a execução do pipeline.
4. Os dados de um simulador de sensor de tráfego iniciado no início do laboratório cria centenas de mensagens por segundo no tópico Pub / Sub. Isso fará com que o Dataflow aumente o número de funcionários para manter o atraso do sistema do pipeline em níveis ideais.
5. Clique em **Ver mais histórico** . No pop-up do **histórico** do trabalho, você pode ver como o Dataflow alterou o número de trabalhadores. Observe a coluna **Justificativa** que explica o motivo da mudança.

**Tarefa 9: Atualizar o script de simulação de dados do sensor**

O ambiente de laboratório de treinamento possui limites de cota. Se o script de simulação de dados do sensor for executado por muito tempo, ele passará um limite de cota, fazendo com que as credenciais da sessão sejam suspensas.

1. Volte para o terminal SSH **training\_vm** onde os dados do sensor estão executando.
2. Se você vir mensagens que dizem **INFO: Publishing** , o script ainda está em execução. Pressione **CRTL + C** para pará-lo. Em seguida, emita o comando para iniciar o script novamente.

cd ~/training-data-analyst/courses/streaming/publish

./send\_sensor\_data.py --speedFactor=60 --project $DEVSHELL\_PROJECT\_ID

1. Se o script ultrapassar o limite de cota, você verá mensagens de erro repetidas que "as credenciais não puderam ser atualizadas" e talvez não seja possível usar **CTRL + C** para interromper o script. Simplesmente feche o terminal SSH. Abra um novo terminal SSH. A nova sessão terá uma nova cota.
2. No Console, no **menu Navegação** ( 7a91d354499ac9f1.png), clique em **Compute Engine** > **instâncias de VM** .
3. Localize a linha com a instância chamada **training\_vm** .
4. Na extrema direita, em "Conectar", clique em **SSH** para abrir uma terceira janela de terminal.
5. No terminal SSH **training\_vm** , insira o seguinte para criar variáveis ​​de ambiente.

source /training/project\_env.sh

1. Use os seguintes comandos para iniciar um novo simulador de sensores.

cd ~/training-data-analyst/courses/streaming/publish

./send\_sensor\_data.py --speedFactor=60 --project $DEVSHELL\_PROJECT\_ID

**Tarefa 10: integração do Stackdriver**

A integração do Stackdriver Monitoring com o Dataflow permite que os usuários acessem as métricas do job do Dataflow, como Atraso do sistema (para tarefas de streaming), Status do trabalho (com falha, êxito), Contagens de elementos e Contadores de usuários no Stackdriver.

Monitorando recursos de integração do Stackdriver

* **Explore as métricas do Dataflow** : navegue pelas métricas disponíveis do pipeline do Dataflow e visualize-as nos gráficos.

Algumas métricas comuns do Dataflow.

|  |  |
| --- | --- |
| **Status de trabalho** | Status do trabalho (com falha, bem-sucedido), relatado como um enum a cada 30 segundos e na atualização. |
| **Tempo gasto** | Tempo decorrido de trabalho (medido em segundos), relatado a cada 30 segundos. |
| **Atraso do sistema** | Máximo atraso em todo o pipeline, relatado em segundos. |
| **Contagem atual de vCPU** | Número atual de CPUs virtuais usadas pelo trabalho e atualizadas na alteração de valor. |
| **Contagem estimada de bytes** | Número de bytes processados ​​por PCollection. |

* **Métricas do Chart Dataflow nos Painéis do Stackdriver** : Crie Painéis e gráficos de séries temporais de métricas do Dataflow.
* **Configurar alertas** : defina limites em métricas no nível de grupo de tarefas ou de recursos e em alertas quando essas métricas atingirem valores especificados. O sistema de alerta do Stackdriver pode notificar uma variedade de condições, como atrasos no sistema de transmissão de longa duração ou falhas nos trabalhos.
* **Monitore as métricas definidas pelo usuário** : Além das métricas do Dataflow, o Dataflow expõe as métricas definidas pelo usuário (Agregadores do SDK) como contadores personalizados do Stackdriver na interface do usuário do Monitoring, disponíveis para gráficos e alertas. Qualquer Agregador definido em um pipeline do Dataflow será relatado ao Stackdriver como uma métrica personalizada. O Dataflow definirá uma nova métrica personalizada em nome do usuário e relatará atualizações incrementais para o Stackdriver aproximadamente a cada 30 segundos.

**Tarefa 11: Explorar métricas**

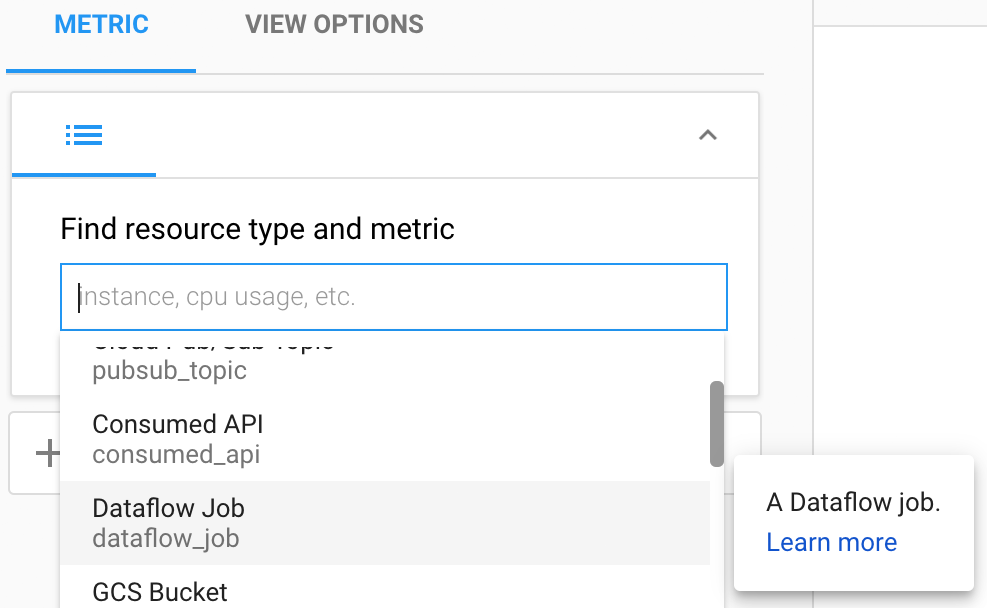
O monitoramento do Stackdriver é um serviço separado no Google Cloud Platform. Portanto, você precisará passar por algumas etapas de configuração para inicializar o serviço para sua conta de laboratório.

**Configurar conta do Stackdriver**

1. Volte para a guia do navegador para o Console. No **menu Navegação** ( 7a91d354499ac9f1.png), clique em **Stackdriver> Monitoramento** .
2. Se necessário, clique **em Fazer login com o Google** .
3. Clique em **Criar espaço de trabalho** .
4. Clique em **Continue** .
5. Clique em **Ignorar a configuração da AWS** .
6. Clique em **Continue** .
7. Selecione **Nenhum relatório** e clique em **Continuar** .
8. Pode levar alguns minutos para o Stackdriver importar informações do projeto sobre sua conta de laboratório e os recursos que já estão sendo usados. Quando o botão Iniciar monitoramento estiver ativo, clique em **Ativar monitoramento** .

Explore as métricas do Stackdriver

1. No painel à esquerda, clique em **Recursos> Metrics Explorer** .
2. No Metrics Explorer, localize e selecione o tipo de recurso **Dataflow\_job** . Você deve ver uma lista de métricas relacionadas ao Dataflow disponíveis.



1. Selecione o **trabalho do Dataflow** do recurso e o **atraso da marca d'água de dados**da métrica **.**
2. O Stackdriver desenha um gráfico no lado direito da página.
3. Em **Localizar tipo de recurso e métrica** , clique no **(x)** para remover a métrica de **atraso de marca d'água de dados** . Selecione uma nova métrica, **System Lag** .

As métricas que o Dataflow fornece ao Stackdriver são listadas aqui:

<https://cloud.google.com/monitoring/api/metrics_gcp>

(Pesquisar na página do Dataflow).

As métricas que você visualizou são indicadores úteis do desempenho do pipeline.

**Idade da marca d'água de dados:** A idade (tempo desde o registro de data e hora do evento) do item de dados mais recente que foi totalmente processado pelo pipeline.

**Atraso do sistema:** a duração máxima atual que um item de dados está aguardando processamento, em segundos.

**Tarefa 12: criar alertas**

Se você quiser ser notificado quando uma determinada métrica ultrapassar um limite especificado (por exemplo, quando o atraso do sistema do pipeline de fluxo de laboratório aumentar acima de um valor predefinido), você poderá usar os mecanismos de alerta do Stackdriver para fazer isso.

**Crie um alerta**

1. Na página do Stackdriver Monitoring, clique em **Alerting> Policies Overview** .
2. Clique em **Add Policy** .
3. Na página **Criar nova política de alerta** , clique em **Adicionar condição** .
4. Na linha **Limite Métrico** , clique em **Selecionar** .
5. Na seção **Destino** , defina o **TIPO DE RECURSO** para o **trabalho do Dataflow** .
6. Em **APLICAR PARA** , selecione **Único** .
7. Selecione o recurso que você usou na tarefa anterior.
8. Na seção **Configuração** , defina **IF METRIC** como **System Lag** .
9. Definir **CONDITION** para **acima** .
10. Defina **THRESHOLD** como **5** .
11. Definir **PARA** a **um minuto** .
12. Clique em **Salvar Condição** para salvar o alerta.

Adicionar uma notificação

1. Em **Notificação** , clique no menu suspenso para visualizar as opções do canal de notificação. Você pode configurar uma política de notificação, se desejar, usando seu endereço de e-mail.
2. Na seção **Nome desta política** , atribua à política um nome como **MyAlertPolicy** .
3. Clique em **Salvar Política** .

Visualizar eventos

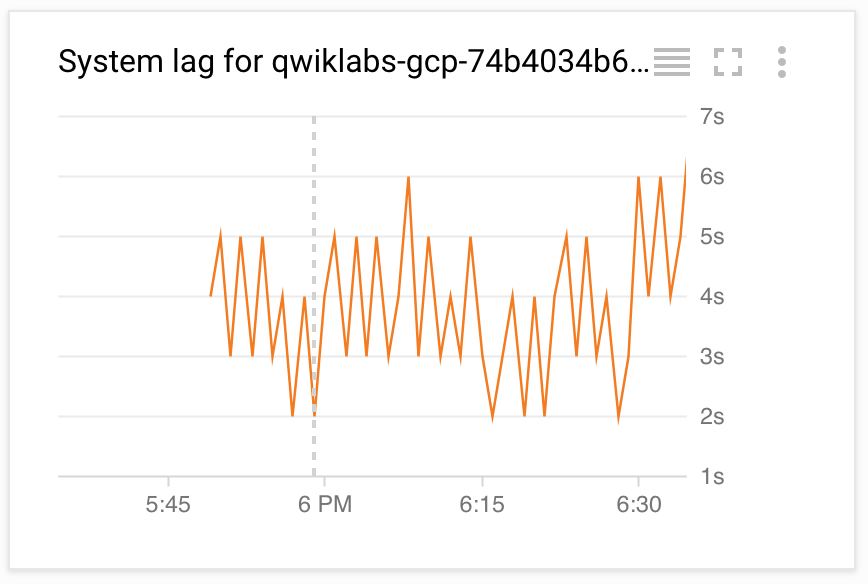
1. Na guia Stackdriver, clique em **Alerting> Events** .
2. Toda vez que um alerta é acionado por uma condição de limite métrico, um incidente e um evento correspondente são criados no Stackdriver. Se você especificou um mecanismo de notificação no alerta (e-mail, SMS, pager, etc), você também receberá uma notificação.

**Tarefa 13: Configurar Painéis**

Você pode criar facilmente painéis com os gráficos relacionados ao Dataflow mais relevantes com os painéis de controle do Stackdriver Monitoring.

1. Na guia Stackdriver, clique em **Painéis> Criar painel** .
2. Clique em **Add Chart** .
3. Na página Adicionar gráfico:
4. Na caixa **Localizar tipo de recurso e métrica** , comece a digitar o **trabalho do Dataflow** e, em seguida, selecione-o como o tipo de recurso.
5. Depois de selecionar um Tipo de Recurso, o menu do campo Métrica será exibido. Selecione uma métrica para o gráfico, como **Atraso do sistema** .
6. No painel Filtro, selecione **projeto e** , em seguida, selecione ou insira seu projeto no campo **Valor** e clique em **Aplicar** .
7. Clique em **Salvar** .

Exemplo:



Você pode adicionar mais gráficos ao painel, se desejar, por exemplo, taxas de publicação de Pub / Sub no tópico ou um backlog de assinatura (que é um sinal para o autoescalonador do Dataflow).

**Tarefa 14: Iniciar outro pipeline de streaming**

1. No terminal SSH **training\_vm** , examine o aplicativo **CurrentConditions.java** . **Não faça alterações no código.**

cd ~/training-data-analyst/courses/streaming/process/sandiego/src/main/java/com/google/cloud/training/dataanalyst/sandiego

cat CurrentConditions.java

1. Copie e cole o seguinte URL em uma guia do navegador para visualizar o código-fonte no Github.

https://github.com/GoogleCloudPlatform/training-data-analyst/blob/master/courses/streaming/process/sandiego/src/main/java/com/google/cloud/training/dataanalyst/sandiego/CurrentConditions.java

O que o script faz?

1. Execute o código **CurrentConditions.java** em um novo pipeline do Dataflow; Esse script é mais simples no sentido de que não faz muitas transformações como **AverageSpeeds** . Os resultados serão usados ​​no próximo laboratório para criar painéis e executar algumas transformações (funções) ao recuperar dados do BigQuery.
2. No terminal SSH **training\_vm** , digite o seguinte:

cd ~/training-data-analyst/courses/streaming/process/sandiego

./run\_oncloud.sh $DEVSHELL\_PROJECT\_ID $BUCKET CurrentConditions

1. Volte para a guia do navegador para o Console. No **menu Navegação** ( 7a91d354499ac9f1.png), clique em **Fluxo de dados** e clique no novo trabalho do pipeline. Confirme se o trabalho do pipeline está listado e verifique se ele está sendo executado sem erros.
2. Levará vários minutos até que a tabela **current\_conditions seja** exibida no BigQuery.

**Termine seu laboratório**

Quando você tiver concluído seu laboratório, clique em **Finalizar Laboratório** . O Qwiklabs remove os recursos que você usou e limpa a conta para você.

Você terá a oportunidade de avaliar a experiência do laboratório. Selecione o número de estrelas aplicável, digite um comentário e clique em **Enviar** .

O número de estrelas indica o seguinte:

* 1 estrela = muito insatisfeito
* 2 estrelas = insatisfeito
* 3 estrelas = neutra
* 4 estrelas = Satisfeito
* 5 estrelas = muito satisfeito

Você pode fechar a caixa de diálogo se não quiser fornecer feedback.

Para comentários, sugestões ou correções, use a guia **Suporte** .

Data da última atualização: 2018-09-25

Data do último teste: 2018-09-20

© 2018 Google LLC Todos os direitos reservados. Google e o logotipo do Google são marcas registradas do Google LLC. Todos os outros nomes de empresas e produtos podem ser marcas registradas das respectivas empresas com as quais estão associados.