Entradas Laterais (Python)

**visão global**

Neste laboratório, você aprende como carregar dados no BigQuery e executar consultas complexas. Em seguida, você executará um pipeline do Dataflow que poderá realizar operações de Mapeamento e Redução, usar entradas laterais e fluxo no BigQuery.

**Objetivo**

Neste laboratório, você aprende como usar o BigQuery como uma fonte de dados no Dataflow e como usar os resultados de um pipeline como entrada secundária para outro pipeline.

* Leia dados do BigQuery no Dataflow
* Use a saída de um pipeline como uma entrada lateral para outro pipeline

**Configuração**

**O que você precisará**

Para concluir este laboratório, você precisará de:

* Acesso a um navegador de internet padrão (navegador Chrome recomendado).
* Tempo. Observe o tempo de **conclusão** do laboratório no Qwiklabs. Essa é uma estimativa do tempo necessário para concluir todas as etapas. Planeje sua programação para que você tenha tempo de concluir o laboratório. Depois de iniciar o laboratório, você não poderá pausar e retornar mais tarde (começará na etapa 1 toda vez que iniciar um laboratório).
* O tempo de **acesso** do laboratório é o tempo que seus recursos de laboratório estarão disponíveis. Se você concluir seu laboratório com o tempo de acesso ainda disponível, poderá explorar o Google Cloud Platform ou trabalhar em qualquer seção do laboratório marcada "se tiver tempo". Quando o tempo de acesso acabar, seu laboratório terminará e todos os recursos serão encerrados.
* Você **NÃO** precisa de uma conta ou projeto do Google Cloud Platform. Uma conta, projeto e recursos associados são fornecidos a você como parte deste laboratório.
* Se você já tiver sua própria conta do GCP, não a use para este laboratório.
* Se o seu laboratório solicitar que você faça login no console, **use apenas a conta do aluno fornecida pelo laboratório** . Isso impede que você incorra em cobranças por atividades de laboratório em sua conta pessoal do GCP.

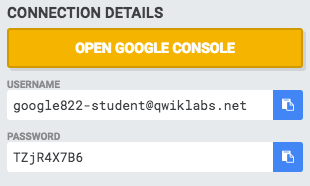
**Comece seu laboratório**

Quando estiver pronto, clique em **Iniciar Laboratório** . Você pode acompanhar o progresso do seu laboratório com a barra de status na parte superior da tela.

**Importante** O que está acontecendo durante esse tempo? Seu laboratório está gerando recursos do GCP para você nos bastidores, incluindo uma conta, um projeto, recursos dentro do projeto e permissão para você controlar os recursos necessários para executar o laboratório. Isso significa que, em vez de gastar tempo manualmente configurando um projeto e construindo recursos do zero como parte de seu laboratório, você pode começar a aprender mais rapidamente.

**Encontre o nome de usuário e a senha do GCP do seu laboratório**

Para acessar os recursos e o console deste laboratório, localize o painel Detalhes da Conexão no Qwiklabs. Aqui você encontrará o ID da conta e a senha da conta que você usará para fazer login no Google Cloud Platform:



Se o seu laboratório fornecer outros identificadores de recursos ou informações relacionadas à conexão, ele também aparecerá nesse painel.

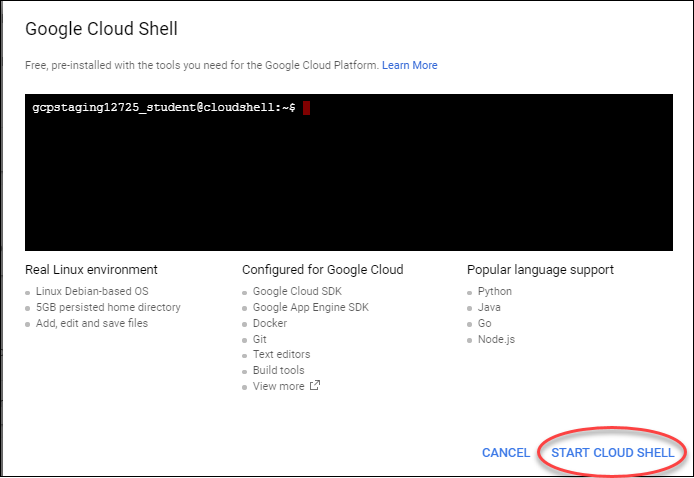
Ativar o Google Cloud Shell

O Google Cloud Shell fornece acesso via linha de comando aos seus recursos do GCP.

No console do GCP, clique no ícone do **Cloud Shell** na barra de ferramentas superior direita:

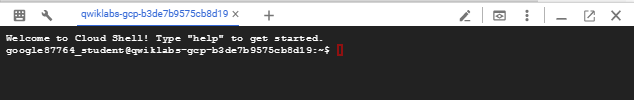


Em seguida, clique em **START CLOUD SHELL** :



Você pode clicar em **START CLOUD SHELL** imediatamente quando a caixa de diálogo aparecer, em vez de esperar na caixa de diálogo até as disposições do Cloud Shell.

Demora alguns momentos para provisionar e se conecta ao ambiente:



O Cloud Shell é uma máquina virtual carregada com todas as ferramentas de desenvolvimento de que você precisa. Ele oferece um diretório pessoal de 5 GB persistente e é executado no Google Cloud, aprimorando muito o desempenho e a autenticação da rede.

Uma vez conectado ao shell da nuvem, você verá que já está autenticado e o projeto está definido como seu *PROJECT\_ID* :

gcloud auth list

Saída:

Credentialed accounts:

- <myaccount>@<mydomain>.com (active)

**Observação:** gcloud é a ferramenta de linha de comando poderosa e unificada do Google Cloud Platform. A documentação completa está disponível no [Google Cloud gcloud Overview](https://cloud.google.com/sdk/gcloud) . Ele vem pré-instalado no Cloud Shell e suporta a conclusão de guias.

gcloud config list project

Saída:

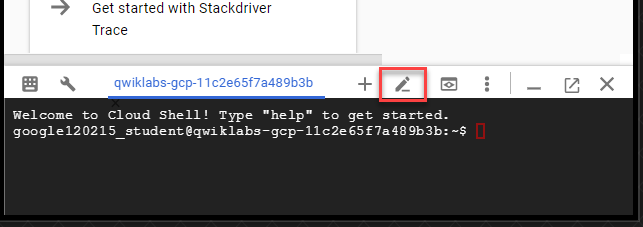
[core]

project = <PROJECT\_ID>

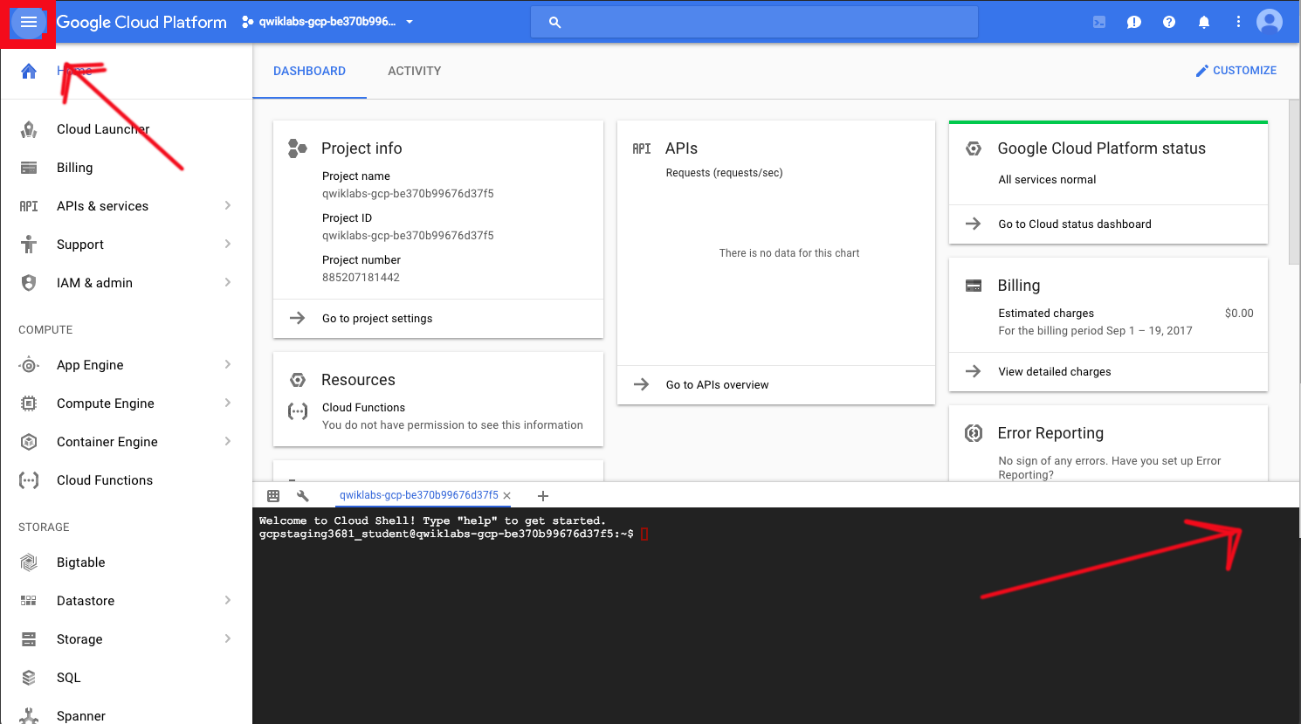
Inicie o Editor de códigos do Google Cloud Shell

Use o Editor de códigos do Google Cloud Shell para criar e editar facilmente diretórios e arquivos na instância do Cloud Shell.

Depois de ativar o Google Cloud Shell, clique no botão **Iniciar editor de códigos**(parece um lápis) para abrir o Cloud Shell Code Editor.

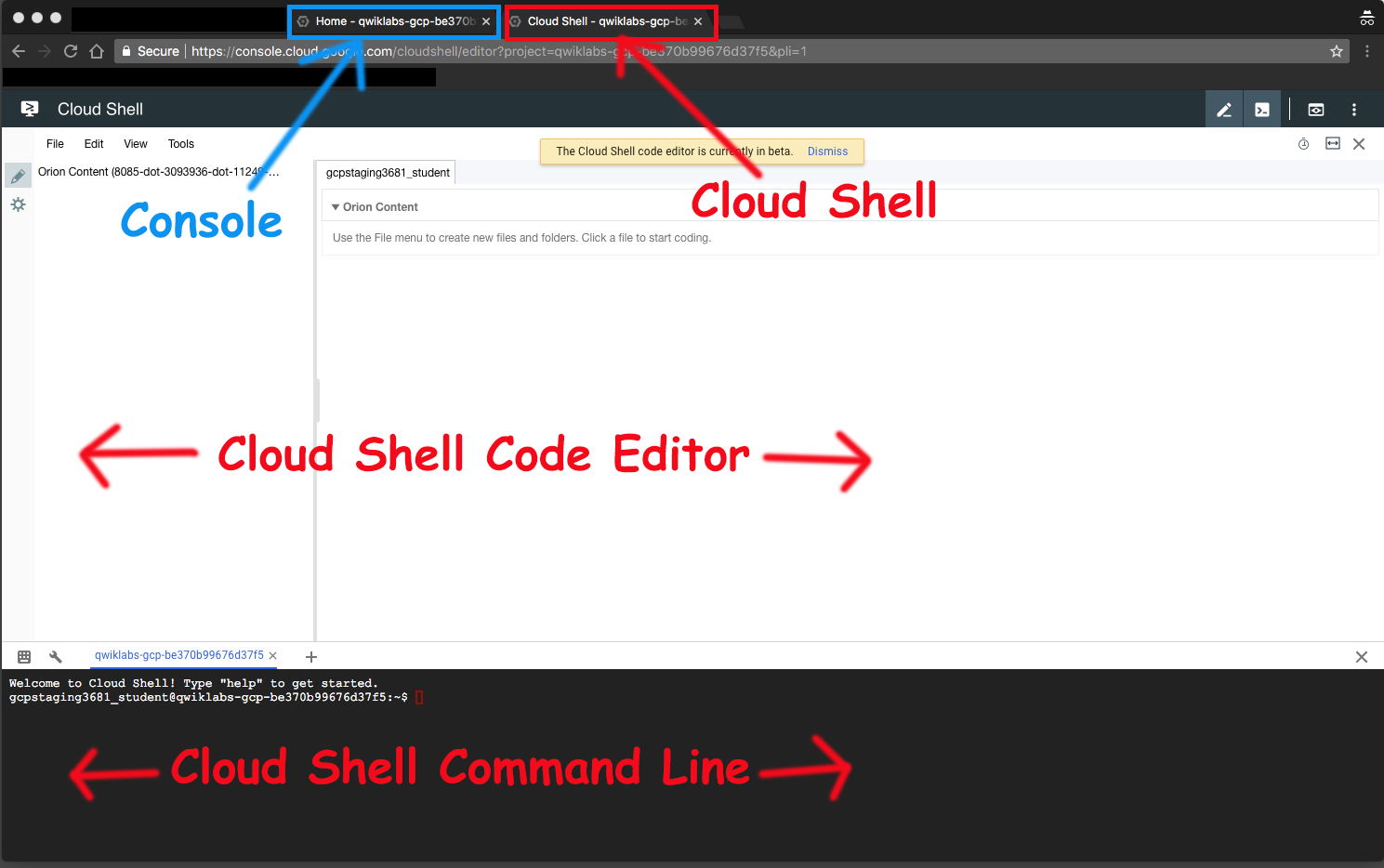


**Nota** : O botão **Launch code editor** pode estar fora da tela à direita. Pode ser necessário clicar no botão do **menu Navegação** para fechar o menu para ver os botões.



Agora você tem três interfaces disponíveis:

* O editor de código do Cloud Shell
* A linha de comando do Cloud Shell
* Console (clicando na guia). Você pode alternar entre o console e o Cloud Shell clicando na guia.



**Tarefa 1. Preparação**

Para este laboratório, você precisará dos arquivos de análise de dados de treinamento.

Verifique se os arquivos do repositório estão no Cloud Shell

1. Clone o repositório na linha de comando do Cloud Shell:

git clone https://github.com/GoogleCloudPlatform/training-data-analyst

1. Clique em **Arquivo> Atualizar** no painel esquerdo do navegador. Você deve ver o diretório de **análise de dados de treinamento** .

Verifique se você tem um intervalo do Cloud Storage

Se você não tiver um intervalo, siga estas instruções para criar um intervalo.

1. No console, no **menu Navegação** ( 7a91d354499ac9f1.png), clique em **Início** .
2. **Selecione e copie** o ID do projeto. Para simplificar, você usará o ID do projeto Qwiklabs, que já é globalmente exclusivo, como o nome do intervalo.
3. No console, no **menu Navegação** ( 7a91d354499ac9f1.png), clique em **Armazenamento** > **Navegador** .
4. Clique em **Criar Balde** .
5. Especifique o seguinte e deixe as configurações restantes como seus padrões:

|  |  |
| --- | --- |
| **Propriedade** | **Valor**  (digite o valor ou selecione a opção conforme especificado) |
| **Nome** | <your unique bucket name (Project ID)> |
| **Classe de armazenamento padrão** | Multi-Regional |
| **Localização** | <Your location> |

1. Clique em **Criar** .
2. Registre o nome do seu balde. Você precisará disso nas tarefas subseqüentes.
3. No Cloud Shell, digite o seguinte para criar uma variável de ambiente chamada "BUCKET" e verifique se existe com o comando echo.

BUCKET="<your unique bucket name (Project ID)>"

echo $BUCKET

Você pode usar o $ BUCKET nos comandos do Cloud Shell. E se você precisar inserir o nome do intervalo <your-bucket> em um campo de texto no Console, poderá recuperar rapidamente o nome com "echo $ BUCKET".

Verificar a variável de ambiente para seu ID do projeto

1. O Cloud Shell cria uma variável de ambiente padrão que contém o ID do projeto atual.

echo $DEVSHELL\_PROJECT\_ID

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 Google Dataflow** . Isso levará você à página, **menu Navegação> APIs e serviços> Painel de controle> API de fluxo de dados** . Ele mostrará uma informação de status ou lhe dará a opção de **ativar** a API.
2. Se necessário, **ative** a API.

Verifique se o Apache Beam está instalado no Cloud Shell

1. Retornar ao Cloud Shell. Verifique se o Apache Beam está instalado no Cloud Shell. Se o Cloud Shell expirou e foi reconectado, ele pode ter perdido os componentes de memória do Apache Beam. Não há nenhum dano na reinstalação. Ele vai tomar as medidas necessárias.

cd ~/training-data-analyst/courses/data\_analysis/lab2/python

sudo ./install\_packages.sh

**Tarefa 2. Tente usar a consulta do BigQuery**

1. Retornar à interface da web do BigQuery. Se ainda não estiver aberto, abra o[Console](http://console.cloud.google.com/) . No **menu Navegação** ( 7a91d354499ac9f1.png), clique em **BigQuery** .
2. Quando o console abrir, você vai alterar a interface clássica. Clique em **Ir para a interface clássica** .
3. 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** .
4. Clique em **Compor Consulta** e digite a consulta a seguir.

SELECT

content

FROM

`fh-bigquery.github\_extracts.contents\_java\_2016`

LIMIT

10

1. Clique em **Mostrar opções** e verifique se você está usando o SQL padrão. Você está usando o SQL Padrão se a caixa de seleção **Usar Legacy SQL** estiver desmarcada. Clique em **Ocultar opções** .
2. Clique em **Executar Consulta** .

O que está sendo devolvido?

A tabela do BigQuery fh-bigquery.github\_extracts.contents\_java\_2016contém o conteúdo (e alguns metadados) de todos os arquivos Java presentes no GitHub em 2016.

Para descobrir quantos arquivos Java esta tabela possui, digite a seguinte consulta e clique em **Executar Consulta** :

SELECT

COUNT(\*)

FROM

`fh-bigquery.github\_extracts.contents\_java\_2016`

A razão pela qual zero bytes são processados ​​é que isso é um metadado da tabela.

Quantos arquivos existem neste conjunto de dados?

Este é um conjunto de dados que você deseja processar localmente ou na nuvem?

**Tarefa 3. Explorar o código do pipeline**

1. No editor do Cloud Shell ou no Cloud Shell, navegue até o diretório do laboratório:

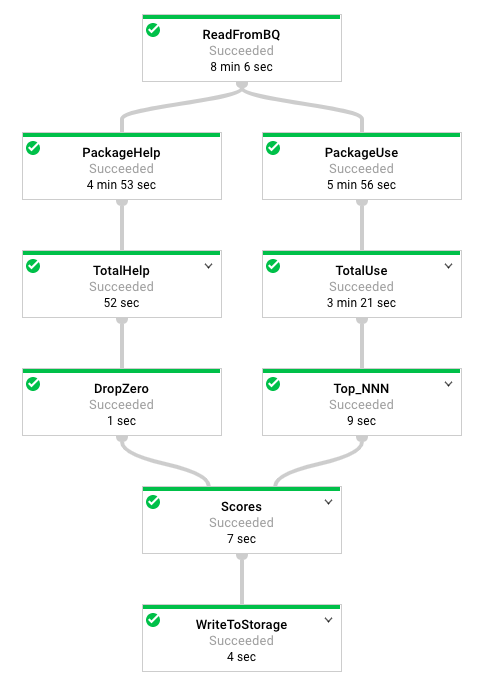
cd ~/training-data-analyst/courses/data\_analysis/lab2/python

1. Veja o código do pipeline usando o editor do Cloud Shell ou o nano. **Não faça alterações no código.**

cd ~/training-data-analyst/courses/data\_analysis/lab2/python

nano JavaProjectsThatNeedHelp.py

Consulte este diagrama enquanto você lê o código. O pipeline é assim:



1. Responda as seguintes questões:

* Olhando para a documentação da classe no topo, qual é o propósito deste pipeline?
* De onde vem o conteúdo?
* O que o lado esquerdo do duto faz?
* O que o lado direito do duto faz?
* O que o ToLines faz? (Dica: veja o campo de conteúdo do resultado do BigQuery)
* Por que o resultado de ReadFromBQ é armazenado em uma PCollection nomeada em vez de ser passado diretamente para outra etapa?
* Quais são as duas ações realizadas na PCollection geradas a partir do ReadFromBQ?
* Se um arquivo tem 3 FIXMEs e 2 TODOs em seu conteúdo (em linhas diferentes), quantas chamadas de ajuda estão associadas a ele?
* Se um arquivo estiver no pacote com.google.devtools.build, com quais pacotes ele está associado?
* popular\_packages e help\_packages são ambos chamados PCollections e ambos usados ​​na etapa Scores (entradas secundárias) do pipeline. Qual é a entrada principal e qual é a entrada lateral?
* Qual é o método usado no passo Scores?
* Qual tipo de dados Python é a entrada secundária convertida na etapa de Contagens?

A versão Java deste programa é um pouco diferente da versão do Python. O Java SDK suporta AsMap e o Python SDK não. Suporta AsDict em vez disso. Em Java, a PCollection é convertida em uma View como uma etapa preparatória antes de ser usada. Em Python, a conversão de PCollection ocorre na etapa em que é usada.

**Tarefa 4. Executar o pipeline**

1. Mude para o diretório:

cd ~/training-data-analyst/courses/data\_analysis/lab2/python

1. O programa requer valores BUCKET e PROJECT e escolhe se o pipeline deve ser executado localmente usando --DirectRunnerou na nuvem usando--DataFlowRunner
2. Execute o pipeline localmente digitando o seguinte no Cloud Shell.

python JavaProjectsThatNeedHelp.py --bucket $BUCKET --project $DEVSHELL\_PROJECT\_ID --DirectRunner

1. Quando o pipeline terminar a execução, no **menu Navegação** ( 7a91d354499ac9f1.png), clique em **Armazenamento> Navegador** e clique no seu balde. Você encontrará os resultados na pasta **javahelp** . Clique no objeto **Result** para examinar a saída.
2. Execute o pipeline na nuvem digitando o seguinte no Cloud Shell.

python JavaProjectsThatNeedHelp.py --bucket $BUCKET --project $DEVSHELL\_PROJECT\_ID --DataFlowRunner

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.
2. Quando o pipeline terminar a execução, no **menu Navegação** ( 7a91d354499ac9f1.png), clique em **Armazenamento> Navegador** e clique no seu balde. Você encontrará os resultados na pasta **javahelp** . Clique no objeto **Result** para examinar a saída.

**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: 2018-11-20

Data do último teste: 2018-11-20

© 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.