

はじめてみよう GCP ハンズオンセミナー

<u>https://goo.gle/startgcp1</u> (本資料のリンク)



Agenda

- 1. <u>サービス共通編</u>
- 2. <u>Compute Engine 編</u>
- 3. <u>地域をまたがる負荷分散編</u>
- 4. Kubernetes Engine 編
- 5. <u>App Engine 編</u>
- 6. <u>Cloud Storage 編</u>
- 7. <u>Cloud SQL 編</u>
- 8. <u>BigQuery</u> 編
- 9. Cloud Dataflow 編





はじめてみようGCP ハンズオンセミナー サービス共通編



トライアルを申し込む

- <u>無料トライアル</u>にアクセスし、Google アカウントでログイン
- フォームの必須フィールドを入力
- 課金情報を入力
 - 名前、住所、支払い方法(クレジットカードまたはデビットカード)
- Google Cloud Console にログイン

 (12 か月間 \$300 の無料クレジットで、あらゆる GCP プロダクトを試用できます。)

トライアル単語のが残わっても、Alweys Free プロダクトは引き続き 2010にいただけます TRY FOR FREE その他の設定	日度 新香酸のお知らせ、パフォーマンスに関するプドバイス、フィードバ ック男素、発発に買する農業情報をメールで受け取ります。 ④ はい ○ ドリス 学べてのワービスと使用 AFI について、適用される引用用約を通守し で利用することに利用 4日の たんごうえて内容に発意します。 BHT 64に当 ILL4を意訳するの意があります BHT 64に当 ILL4を意訳するの意があります ④ はい ○ ドリス	アブリや、ウュブサイト、ウーズスの構築と新行さ 数を Freibare や Doogle Maps API などがすべて 使用できます。 3300 相当のクレジットを無料でご提供 ご意識いただくと、Google Guad Pistform でうゆ はか用回ご利用しただける 300 ドル3ののレジットを要用できます。 無料トライアル期間が終了しても、自 動的に請求されることはありません ロボットによる意識ではないことを確認するため、 ロボットによる意識ではないことを確認するため、
---	--	--

プロジェクトとは

GCPのリソースが必ず紐づく単位 例:開発プロジェクトの下にインスタンスが存在

ダッシュボード	דסדרבדו		
	Cloud Platform	🗣 My First Project 👻	
	アクティビテ・	т	
		● プロジェクト情報	
	0 リソース	I → API Ø	概要に移動



リソース階層

組織・フォルダ・プロジェクト・ リソースの関係は右図の通り



https://cloud.google.com/resource-manager/d ocs/cloud-platform-resource-hierarchy?hl=ja







必要に応じてクーポンコードを有効にする

- 1. Google Cloud Console にログオンする
- 2. 下記 URL にアクセス https://cloud.google.com/redeem
- 3. プロモーションコードの入力
- 4. コンソールより、[お支払い] [請求アカウント]をクリックし、 プロモーションコードが有効になっているか確認

お支払い 株式グカウントを発発 - ポニンカウント マイブロビュウト ウンノトを用成 単素なブカウントのを発表:- またプロジントを、 業系式アカウント - ステーシス プロジンクト後 後市 株式ガリカウント	
8877070+ VKJDULSH 1979-EME ERG7039-646EE 19857091-6 - 255709-6 - 27-93 JDULSHE MM = 88577091-6 - 25	
カンソトを作成 を取むアカンメトロトを参加 - 家家の大学ないトモー 高学を見てみつントモー ステータス プロジェクト巻 後日 様式アスワント	
■##57Aつンド&- ■#587Aつンド4 37-93 プロジェクト機 体作 ii ##5720ンド ■ 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	
###22224	
プロモーション コードの利用 プロモーション コード デバ ログボジェントを発音 ソンクできる第8末アカウントが10 す 課業大アカウント	008878
*	*>#1 10 XA

	Google Cloud Platform	٩
	お支払い	概要 請求先アカウント - / 請求先アカウント名を変更 O 請求先アカウントを閉鎖
51	積要	請求先アカウントの編要 お支払いの概要
iĥ	予算とアラート	請求先アカウント ID: 00 01
O	料金の履歴	AL STUL
杰	課金データのエクスポート	プロモーションID 有効規模へ プロモーションの金額 残額
1	お支払い設定	Free Trial 2018/08/16 ¥33,177.00 ¥23,647.61
٥	お支払い方法	Marketing Credit - Enterprise hands on session 2019/02/02 ¥32,699.00 ¥32,699.00
		この請求先アカウントにリンクされているプロジェクト プロジェクト名 プロジェクト lo kimi dev project kimi-dev project :



注: クーポンコードを利用するには有料アカウントに アップグレードする必要があります。 有料アカウントでも、適用したクレジット分は無料で 利用できます。

課金情報を確認する

課金情報を確認します。

	Google Cloud Platform	🐌 My First Project 👻	9	
	お支払い	概要 請求先アカウント 👻	✔ 請求先アカウント名を変更 ③ 請求先アカウントを閉鎖	
58	概要	讀求先アカウントの概要 お支払いの概要		
66	予算とアラート	請求先アカウント ID: 123 XXX-XXXXXX-XXXXXX		4月21日 1000 1000 1000 1000 1000 1000 1000 1
\odot	料金の履歴	AL STWL	_ クレンツトにノロモーションの。] ねていてこした 変烈したた	ゆ200 い豆球ら
击	課金データのエクスポート		れていることを確認しまり	
÷	お支払い設定	¥33,668.00	365 残り日数	
۵	お支払い方法	テの残高:¥33,668.00	201B/11/21 に期限切れ になります	
	L	この請求先アカウントにリンクされている	5プロジェクト	
	Γ	プロジェクト名 プロジェクト ID	まままでもしいについいよう	レが即声づいて
		My First Project xxxxx-xxxxx-186716	inることを確認します	「い」実達 しいし



課金額の確認方法

[お支払いの概要]をクリックすると課金額を確認することができます。





参考: プロジェクトにアカウントを追加





GCP へのアクセス方法





Google Cloud Console へのアクセス

Google Cloud Platform を利用するには、Cloud Platform コンソールに アクセスします。

Cloud Platform コンソール:

https://console.cloud.google.com/





コマンドラインツールの選択肢

用途に応じた選択肢

- Google Cloud Shell を利用
 - ブラウザから手軽にコマンドラインツールを実行
 - ブラウザからインスタンスに SSH
- gcloud SDK をローカルにインストール
 - ローカルのマシンのターミナルから GCP リソースを操作
 - 開発を行う際、手元のターミナルを使う際に用いる



Cloud Shell をつかってみよう

Google Cloud Shell とは

- ブラウザから直接クラウドリソースにアクセスできるシェル
- gcloud など SDK、CLI ツールにインストール済み





Cloud Shell をつかってみよう

Cloud Shell へのアクセス

コンソールで1クリックするだけ

			Cloud Shell をアクティブにす
Google Cloud Platform 🛟 My First Project 👻	٩		• B 0 0 0
ジ リソース Compute Engine 1個のインスタンス	CPU (%)	0.25	 Google Cloud Platform のステータス : 金サービス正常
= トレース	1	0.15	→ クラウド ステータス ダッシュボードに移動



Cloud Shell をつかってみよう

Cloud Shell へのアクセス

- シェルが立ち上がったら以下のコマンドを実行
- プロジェクトー覧が表示される

\$ gcloud projects list







コマンドラインツールをインストールする

Google Cloud Platform の各サービスは、GUI の管理コンソールとともに コマンドラインツールを提供しています。

インストール方法は以下をご参照ください。 https://cloud.google.com/sdk/

		Υ.	5 9 9 3 :	
			設定 キーボード ショートカット	イズ
	1	 Google Cloud 全サービス正常 	ダウンロード ゼ クラウド パートナー ゼ]
	7e-3 6e-3	クラウドステータ:	利用規約 プライバシー 2	
Mm	5e-3 4e-3 3e-3	 高求期間 2019/06/ 	プロジェクトの設定 インタラクティブなチュートリアルを試す	





コマンドラインツールをインストールする



CLOUD SDK

Google Cloud Platform のプロダクトとサービスのためのコマンドライン インターフェース



Linux/Mac OS X または Windows を選択



Skip

コマンドラインツールをインストールする

正しくインストールされていることを確認してください。

\$ gcloud version

	000	Guest — bash — 80×24
	naoyas-mba:~ Guest\$ gcloud Google Cloud SDK 0.9.64	version
	bq 2.0.18 bq-nix 2.0.18 compute 2015.06.09 core 2015.06.09 core-nix 2015.06.02 dns 2015.06.02 gcloud 2015.06.09 gcutil-msg 2015.06.09	注: gcloud が実行出来ない場合は、パスが設定されている ことを確認してください。 初回実行時はターミナル or コマンドプロンプトを再起動する必 要があります
ogle	gsutil 4.13 gsutil-nix 4.12 sql 2015.06.09	

コマンドラインツールをインストールする

Google

OAuth 認証を行います。

\$ gcloud auth login

.

Guest - Python - 80×24

naoyas-mba:~ Guest\$ gcloud auth login
Your browser has been opened to visit:

https://accounts.google.com/o/oauth2/auth?redirect_ur t%3A8085%2F&prompt=select_account&response_type=code&clie googleusercontent.com&scope=https%3A%2F%2Fwww.googleapis. mail+https%3A%2F%2Fwww.googleapis.com%2Fauth%2Fcloud-plat googleapis.com%2Fauth%2Fappengine.admin+https%3A%2F%2Fwww 2Fcompute&access_type=offline

- Goo	gle Cloud SDK が次の許可をリクエスト ます:
3	Google でのユーザーの把握
۲	メール アドレスの表示
8	Google Compute Engine のリゾースの表示と管理
8	Google App Engine に導入されたアプリケーションの 表示と管理

Skip





コマンドラインツールをインストールする

デフォルトプロジェクトを設定します

\$ gcloud config set project <project id>

正しく設定されていることを確認

\$ gcloud config list



```
Guest — bash — 80×24
naoyas-mba:~ Guest$ gcloud config set project gcpbeginner-project01
naoyas-mba:~ Guest$ gcloud config list
[core]
account = gcpbeginner@gmail.com
disable_usage_reporting = True
project = gcpbeginner-project01
naoyas-mba:~ Guest$
```

参考: gcloud コマンド

gcloud コマンド Tips 集

- インストールされているコンポーネントを確認する
 - \$ gcloud components list
- コンポーネントをアップデートする
 - \$ gcloud components update
- 新規にコンポーネントをインストール(アップデート)する
 - \$ gcloud components update app-engine-python
- コンポーネントを削除する
 - \$ gcloud components remove bq



参考: gcloud コマンド

- 設定情報を確認する
 - \$ gcloud config list
- プロジェクト ID を確認する
 - \$ gcloud config list core/project
- プロジェクトを変更する
 - \$ gcloud config set project <project id>
- Google アカウントを変更する
 - \$ gcloud auth login
- 構成プロパティを設定する
 - o \$ gcloud config set <property> <value>
 - 例:\$ gcloud config set compute/zone asia-east1-b
- 構成プロパティを設定解除する
 - \$ gcloud config unset <property>





gcloud コマンド活用ビデオ http://youtu.be/4y4-xn4Vi04





参考: 言語設定を変更するには

[設定] > [言語と地域] > [言語]



参考: 言語設定を変更するには

\$ ⊐-t	ザー設定	言語と地域
≜ ⊐ミュ:	ニケーション	言語
⑤ 言語と ¹	地域	日本語
	- 1 -	日付の表示形式
	×1 X	言語のデフォルト(2015年3月14日) -
	ł	特刻の表示形式
	[言語のデフォルト(15:45) 🔹
	2	数字の表示形式
		言語のデフォルト (1234.56)



本ハンズオン後に不用な課金を避けるため

- プロジェクトを削除できる場合は削除してください。作成したリソースは全て削除されます。
 - [IAMと管理] > [設定] > [シャットダウン]
- プロジェクトを削除できない場合、利用したリソースを個別に削除してください。





はじめてみよう GCP ハンズオンセミナー Compute Engine 編



ハンズオンのシステム構成





Web サーバを立ち上げる

- 手順概要
 - VM インスタンスを作成する
 - ログインする
 - Web ページを提供する
 - VM インスタンスを削除する



Compute Engine ページへのアクセス

• Cloud Platform コンソール左上でサービス一覧を表示

	★ ホーム	
😑 🛛 Google Cloud Platform 🔹 My First Project 👻	■ ここに固定されたプロダクトが表	
	・ 市 サポート >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>	1
ナビゲーションメニュー アクティビティ	● IAM と管理 >>	
	🤝 スタートガイド	
	🔮 セキュリティ 💦 👌	VM インスタンス
	コンピューティング	インスタンス グループ
● ノロシェクト 情報	ِنُ App Engine >	キシスタンス テシブレート 単一テナントノード
プロジェクト名	Compute Engine	ディスク スナップショット
My First Project	Kubernetes Engine	イメージ
	(···) Cloud Functions	TPU 確約利用割引

• Cloud コンソールより

[コンピューティング] > [Compute Engine] > [VMインスタンス]を選択 (初回は少し時間が掛かります)

• [作成]または[インスタンスを作成]ボタンをクリックする





- インスタンス名を入力(例:www1)
- リージョンを選択(例: asia-northeast1)
- ゾーンを選択(例: asia-northeast1-a)

名前 🕜

www1				
リージョン 🕜		ゾーン 🕐		
asia-northeast1(東京)	•	asia-northeast1-a	•	



- マシンタイプを選択
 - micro(共有vCPU x1)
- イメージを選択
 - Debian GNU/Linux 9
- ファイアウォールを設定
 - HTTPトラフィックを許可
 - HTTPSトラフィックを許可

マシンタイプ カスタマイズしてコア、メモリ、GPUを選択します。

micro(共有 vC *	メモリ 0.6 GB	カスタマイズ
アカウントをアップグレー 成します	- ドし、コア数が最大で 96 1	個のインスタンスを作

コンテナ 🍘

この VM インスタンスにコンテナ イメージをデプロイします。詳細

プートディスク 🛞



IDとAPIへのアクセス 🛞



Google Cloud

チェックボックスに チェックする

ックインフィール ルールを追加して、インターネットからの特定のネットワーク トラフィックを許可します

✓ HTTP トラフィックを許可する
✓ HTTPS トラフィックを許可する

ブートディスクの設定を確認

ファイアウォール ② タグとファイアウォール ルールを追加して、インターネットからの特定のネットワーク トラフィックを許可します

✓ HTTP トラフィックを許可する
 ✓ HTTPS トラフィックを許可する

※ 管理、セキュリティ、ディスク、ネットワーク、単一テナンシー

折り畳まれている詳細設定を開く

削り	^{余ルール} インスタンスを削除する際にブートディスクを削除する
暗 9 デ-	号化 −夕は自動的に暗号化されます。暗号鍵管理ソリューションを選択してください
0	Google が管理する鍵 設定は必要ありません
	お客様が管理する鍵
0	Google Cloud Key Management Service により管理します お客様が指定する鍵
	Google Cloud の外部で管理します
追加	ロディスク 🜍 (Optional)
	+ 新しいディスクを追加 + 既存のディスクを接続

管理 セキュリティ ディスク ネットワーキング 単一テナンシー

デフォルトで「インスタンスを削除する際にブートディスク を削除する」が有効になっていることを確認


VM インスタンスを作成する

• 作成ボタンをクリック

このインスタンスについては課金が発生します。Compute Engineの料金 🖸





VM インスタンスの作成確認

Google Cloud

[Compute Engine] > [VM インスタンス] より
 VM インスタンスが作成されたことを確認してください。

💲 My First Project 👻	٩			5 0 4
VM インスタンス	🛃 インスタンスを作成	▼ 古 VM のインポート	C 更新	通知
	.411×1 <i>1</i> 4		別を選択	VM インスタンス「www1」とそのブートディ たった今 スク「www1」を作成 My First Project
□ 2前 2 V-V	が (1) (1) (1) (1) (1) (1) (1) (1) (1) (1)	Alan 运航	71.6.251	プロジェクト My First Project の Compute Engine 8 分前 を初期化しています My First Project
🦳 🏈 www1 asia-northea	st1-a 10.146.0.2 (nic	0) 35.200.121.214 년 ⁷ SSH ·	•	My First Project ジェクト「My First Project」を作成 23 分前
				すべてのアクティビティを表示

Cloud Shell からのログイン

 インスタンスー覧画面から対象インスタンスの行を探し、 右端の SSH ボタンより gcloud コマンドを表示 をクリック





Cloud Shell からのログイン

• gcloud コマンドを Cloud Shell で実行

►	開始	■ 停止	心 リセット	育 削除
	② 列,	-		
接続				
SSH	• :			
	ブラウ	ザウィンドウで	開く	
	ブラウ	ザウィンドウで	カスタムポートを購	1<
	指定され	れた秘密 SSH 謬	R証鍵を使用してブラ	ラウザ ウィンドウで開く
	gcloud	コマンドを表示	ŧ	
			大使田	

gcloud コマンドライン

このインスタンスへの SSH 通信には次の gcloud コマンドラインを使用できます。



☑ 行の折り返し

geloud リファレンス



閉じる CLOUD SHELL で実行する

Cloud Shell からのログイン

Cloud Shell が起動するので、適宜質問に答えていきます。

kkitase@cloudshell: ~ (kimi-handson-project) \$ gcloud compute --project "kimi-handson-project" ssh --zone "asia-northeast1-a" "www1" ENTER

This tool needs to create the directory [/home/kkitase/.ssh] before being able to generate SSH keys. Do you want to continue (Y/n)?

Enter passphrase (empty for no passphrase): ENTER Enter same passphrase again: ENTER

kkitase@www1:~\$



Web ページを提供する

- Apache Web サーバをインストールする
- 以下の2つのコマンドを実行する

- \$ sudo apt update
- \$ sudo apt install -y apache2



Web ページを作成する

- Web ページ(HTML ファイル)を作成する
- 以下の echo コマンドをコピー & ペーストして実行する

\$ echo "<html><body><h1>Hello World! running on `hostname`</h1></body></html>" |
sudo tee /var/www/html/index.html



Web ページを作成する

- VM インスタンスの一覧画面で、外部IPアドレスを確認
- IP アドレスのリンクをクリックし、ブラウザでアクセス
- アドレスバーの https を http に変更

名前 ^	ゾーン	おすすめ	使用中	内部 IP	外部 IP	接続
🥝 www1	asia-northeast1-a			10.146.0.2 (nic0)	35.200.100.65 🖒	ssh 🔹 🚦

Hello World! running on www1



(演習 1)リージョン跨ぎの通信

以下の構成でVM インスタンスを作成してください





(演習1)リージョン跨ぎの通信

- ローカル端末から ASIA リージョン、US リージョンにある VM インスタンス(外部 IP アドレス)に ping を発行して、レイテンシを確認してみてください
- www1 に SSH ログインして、www2 に向けて ping コマンドを実行してく ださい(ping [www2の内部IPアドレス] を実行)
 ✓ リージョンをまたいでも、プライベート IP アドレスでアクセスできることを確認してください



(演習 2) プライベートネットワークを作成

プライベートネットワークを作成します。



mynetwork



(演習 2)プライベートネットワークを作成

H	VPC ネットワーク				
8	VPC ネットワーク	VPC ネットワーク	+ VPC ネットワーク	7を作成 C 更調	新
Ľ	外部 IP アドレス			J	
	ファイアウォール ルール	名前 ヘ リージョン	サブネット モード	IP アドレス範囲	ゲートウェイ
X	ルート	default	18 自動、	•	
ං	VPC ネットワーク ピアリング	us-central1	default	10.128.0.0/20	10.128.0.1
×	共有 VPC	europe-west1	default	10.132.0.0/20	10.132.0.1



(演習 2)プライベートネットワークを作成

ネットワークの設定を行う

- ネットワーク名を入力
 - mynetwork
- サブネット作成モードは[自動]を選択
- ファイアウォールルールをチェック
- [作成]をクリック





(演習 2) プライベートネットワークを作成

• Mynetwork が作成されたことを確認する

VPC A	989-9	1 VPC 4 9	F7-74	TERS C EN		
名前 へ	リージョン	サブネット	Ξ-K	IP アドレス範囲	ゲートウェイ	ファイアウォール ルール
default		18	自動 🕶			б
	us-central1	default		10.128.0.0/20	10.128.0.1	
	europe-west1	default		10.132.0.0/20	10.132.0.1	
	us-west1	default		10.138.0.0/20	10.138.0.1	
	asia-east1	default		10.140.0.0/20	10.140.0.1	

asia-east2	default	10.170.0.0/20	10.170.0.1	
mynetwork	18	自動 👻		4
us-central1	mynetwork	10.128.0.0/20	10.128.0.1	
europe-west1	mynetwork	10.132.0.0/20	10.132.0.1	
us-west1	mynetwork	10.138.0.0/20	10.138.0.1	
asia-east1	mynetwork	10.140.0.0/20	10.140.0.1	
us-east1	mynetwork	10.142.0.0/20	10.142.0.1	



(演習 2) プライベートネットワークを作成

新しいインスタンス www3 を作成して mynetwork に接続します





(演習 2) プライベートネットワークを作成

VM インスタンス作成時にネットワークを選択することができます

管理 セキュリティ ディスク ネットワーキング	単一テナンシー	
ネットワーク タグ 🥹 (首略可)		
ネットワーク インターフェース 🔞		
ネットワーク インターフェース	~	
ネットワーク 🕜		新しく作成した
mynetwork	*	mynetwork ≁ ৲ख़ ₊⊓
サブネットワーク 💿		を選択
mynetwork (10.132.0.0/20)	•	
プライマリ内部 IP 📀		
エフェメラル(自動)	*	
※ エイリアス IP 範囲を表示		
外部 IP 🥥		
エフェメラル	•	



(演習 2)通信確認

1. www3 にログインして、www1 に対して ping を実行してください ✓ ネットワークを跨いでプライベート IPでアクセスできないことを確認してください

2. www3 にログインして、www1 の Public IP アドレスに対して ping を実行 してください ✓ パブリックIPではアクセスできることを確認してください



500GB のディスクを作成して、VM インスタンスに追加して使えるようにしま しょう。

手順概要は以下のとおりです。

- 永続ディスクを作成
- VM インスタンスにディスクをアタッチ
- フォーマット&マウント
 - <u>https://cloud.google.com/compute/docs/disks/add-persistent-disk</u>



- 対象インスタンスを選択(インスタンス名をクリック)し、[編集]をクリック
- 追加ディスクセクションで [+新しいディスクの追加] をクリック (新しいディスクの設定を入れるセクションがオープンする)

追加ディスク 🔞 (省略可)

+新しいディスクを追加

+ 既存のディスクを接続

名前 📦 disk-2 説明(省路可) タイプ 😡 標準の永続ディスク Ŧ スナップショット スケジュール スナップショットスケジュールを使用してディスクバックアップを自動化します。 スケジュールされたスナップショット ビ スケジュールなし -データを定期的にバックアップするには、スナップショッ 閉じる トのスケジュールを作成します。 スナップショットのスケジュールの作成に関する詳細 [7] ソースの種類

1

空のディスク イメージ スナップショット

新しいディスク(disk-2, ブランク, 500 GB)



永続ディスクの情報を入力します。

- 名前: ディスクの名前
- ディスクタイプ:標準の永続ディスク
- ソースの種類:なし(空のディスク)
- サイズ: 500

完了ボタンをクリック



名前 1/2			
disk-2			
説明(省略可)			
タイプ 😡			
標準の永続ディスク			
スナップショット スケジュール スナップショット スケジュールを使用してデ- スケジュールさわた スナップショット 「 ³	ィスクバックフ	アップを自動化	しま
スケジュールなし			
データを定期的にパックアップする トのスケジュールを作成します。 スナップショットのスケジュールの	には、スナッ 作成に関する	プショッ 詳細 ピ	閉じ
ソースの種類 🚱			
空のディスク イメージ スナップショ	マト		
モード ● 読み取り / 書き込み			
○ 読み取り専用			
削除ルール インスタンスを削除したときの動作			
 ディスクを維持 ディスクを削除 			
# 1 (CP) 0			
サイズ (GB) () 500			
サイズ (GB) ② 500 #FFパフォーマンス ②			
サイズ (GB) ② 500 推定パフォーマンス ③	-112.町M	##1] 7	
サイズ (68) @ 500 推定パフォーマンス @ オペレーションのタイプ 持続的なランダム IOPS の上眼	読み取り 375 00	書き込み 750.00	
サイズ (GB) ④ 500 推定パフォーマンス ④ オペレーションのタイプ 持続的なランダム IOPS の上限 特徴的なランダム IOPS の上限	読み取り 375.00	書き込み 750.00	
サイズ (BB) ④ 500 増定パフォーマンス ④ オペレーションのタイプ 持続的なランダム IOPS の上限 持続的なスループットの上限 (MB/s)	読み取り 375.00 60.00	書한込み 750.00 60.00	
サイズ (cg) ◎ 500 増走パフォーマンス ◎ オペレーションのタイプ 持続的なフンダム IOPS の上限 持続的なスループットの上限 (MB/s) 明合化 デークタ目動物に暗号化されます。暗号健智 ● Google が増増する鍵 あ定なが得速する鍵 の 50号 Cloud Say Namagement Service (C → 25号が指定する鍵 の Google できなば Say	読み取り 375.00 60.00 星ソリューショ より管理します	 書き込み 750.00 60.00 シを選択してく 	、た
サイズ (GB) ◎ 500 推定パフォーマンス ◎ オペレーションのタイプ 持続的なランダム IOPS の上限 持続的なスループットの上限 (MB/s) 曜行化 データは目的に暗号化されます。暗号鍵管理 認定は必要的生せん わ客様が指定する鍵 Google Chud 的外間が管理します 物理プロッグサイズ (CG) ◎ 激症は後で変更できません。	読み取り 375.00 60.00 星ソリュージョ より管理します	 書き込み 750.00 60.00 シを選択してく 	(<i>t</i>
 サイズ (GB) ● 500 推定パフォーマンス ● オペレーションのタイプ 持続的なランダム IOPS の上間 持続的なスループットの上限 (MB/s) 暗号化 デーダは目的的に暗号化されます。暗号鍵音音 ● Google が管理する鍵 Google たじょび 保y Management Service に む客様が管理する鍵 Google Cloud の外部で管理します 物理プロックサイズ (KC) ● 設定は後で変更できません。 4 (デフォルト) 	読み取り 375.00 60.00 星ソリュージョ より管理しま1	書き込み 750.00 60.00 ンを選択してく す	(だ)
サイズ (GB) ● 500 増定パフォーマンス ● オペレーションのタイプ 特統的なランダム IOPS の上限 特統的なスループットの上限 (MB/s) 暗号化 データは目的に暗号化されます。暗号健智 電Google Coloid の外部で管理します 個での回家 Coloid Key Management Service に あ客様が管理する鍵 Google Coloid の外部で管理します 物理プロックサイズ (KB) ● 設定は後で変更できません。 4 (デフォルト) デバイスを ● マンントやサイズ変更でデバイスを参照する)	読み取り 375.00 60.00 星ソリューショ より管理します ために使用しま	書き込み 750.00 60.00 シンを選択してぐ す	(た)
サイズ (GB) ● 500 増定パフォーマンス ● オペレーションのタイプ 持続的なランダム IOPS の上限 持続的なスループットの上限 (MB/s) 増行化 データは目勤的に暗号化されます。暗号鍵管部 必定は必要のりません わ変体が管理する鍵 のcoogle Colue の外部で管理します 物理プロックサイズ (KQ) ● 設定は変で変更できません。 4 (デフォルト) デパスク名(に基づく (デフォルト)	読み取り 375.00 60.00 星ソリューショ より管理します ために使用しま	書き込み 750.00 60.00 シを選択してく す。	(た)
サイズ (GB) ● 500 増定パフォーマンス ● オペレーションのタイプ 特統的なランダム IOPS の上限 特統的なスループットの上限 (MB/s) 暗守化 データは自動的に暗号化されます。暗号健智士 整定は必要のりません あ客様が管理する鍵 Google Coled の外部で管理します 個子イズ (KB) ● 設定は没て変更できません。 4 (デフォルト) デバイス名 ● アンシトモサイズ変更でデバイスを参照する) ディスク名に基づく (デフォルト) dsk-2	読み取り 375.00 60.00 望ソリューショ より管理しま1	 書き込み 750.00 60.00 シを選択してく す 	() E

インスタンスにディスクが追加されたことを確認し、保存をクリック

新しいディスク	(disk-2 ブランク 500 0	(B)	
810× / 1 / /	(0.00 2, 7 7 7 7 7,000 0		





(演習3)ディスクを追加する

VM インスタンスにログインする

►	開始	■ 停止	心 リセット	會 削除
	Ø 列 ▼			
接続				
SSH	• :			
	ブラウザ	ウィンドウで	開く	
	ブラウザ	ウィンドウで	カスタムポートを開	I<
	指定され	た秘密 SSH ii	R証鍵を使用してブラ	ラウザ ウィンドウで開く
	gcloud I	コマンドを表示	ŧ	
			+ + m	

gcloud コマンドライン

このインスタンスへの SSH 通信には次の gcloud コマンドラインを使用できます。



■行の折り返し

geloud リファレンス



閉じる CLOUD SHELL で実行する

利用可能なディスク容量を確認します

\$ sudo df -h

fukudak@www1:~\$ sudo df -h						
Filesystem	Size	Used	Avail	Use%	Mounted	on
rootfs	9.9G	655M	8.8G	7%	1	
udev	10M	0	10M	0%	/dev	
tmpfs	372M	116K	372M	1%	/run	
/dev/disk/by-uuid/a3864f53-b3b7-4a6d-9a27-548305aa6594	9.9G	655M	8.8G	7%	1	
tmpfs	5.0M	0	5.0M	0%	/run/loc	ck
tmpfs	743M	0	743M	0%	/run/shn	n

アタッチしたディスクが、まだ利用可能でないことを確認してください



マウントポイントを作成します

\$ sudo mkdir -p <mount point>

例:\$ sudo mkdir -p /mnt/disk1



(演習 3)ディスクを追加する

ディスク名を確認します

\$ ls -l /dev/disk/by-id/google-*

今回追加するディスク。ディスク名で識別できるようになっている

fukudak@www1:~\$ ls -l /dev/disk/by-id/google-*
lrwxrwxrwx 1 root root 9 Jun 2 00:14 /dev/disk/by-id/google-disk-1 -> ../../sdb
lrwxrwxrwx 1 root root 9 Jun 1 23:56 /dev/disk/by-id/google-wwwl-> ../../sda
lrwxrwxrwx 1 root root 10 Jun 1 23:56 /dev/disk/by-id/google-wwwl-part1 -> ../../sda1



(演習 3) ディスクを追加する

フォーマットを実行します

\$ sudo mkfs.ext4 -F -E lazy_itable_init=0,lazy_journal_init=0,discard <disk name>

例: \$ sudo mkfs.ext4 -F -E lazy_itable_init=0,lazy_journal_init=0,discard /dev/disk/by-id/google-disk-1



マウントを実行します

\$ sudo mount -o discard, defaults <disk name> <mount point>

例: \$ sudo mount -o discard, defaults /dev/disk/by-id/google-disk-1 /mnt/disk1



新しいディスクがマウントされ、利用可能になったことを確認します

\$ sudo df -h

fukudak@www1:~\$ sudo df -h				
Filesystem	Size	Used	Avail	Use% Mounted on
rootfs	9.9G	656M	8.8G	7% /
udev	10M	0	10M	0% /dev
tmpfs	372M	116K	372M	1% /run
/dev/disk/by-uuid/a3864f53-b3b7-4a6d-9a27-548305aa6594	9.9G	656M	8.8G	7% /
tmpfs	5.0M	0	5.0M	0% /run/lock
tmpfs	743M	0	743M	0% /run/shm
/dev/sdb	493G	198M	467G	1% /data



(演習 4)Stackdriver

Stackdriverをセットアップして、ログ、メトリックを確認する。

先程作成した www 1 インスタンスに、Stackdriver Monitoring エージェントをインストールします。

\$ curl -sSO https://dl.google.com/cloudagents/install-monitoring-agent.sh \$ sudo bash install-monitoring-agent.sh

• Stackdriver Logging エージェントをインストールします。

\$ curl -sSO https://dl.google.com/cloudagents/install-logging-agent.sh \$ sudo bash install-logging-agent.sh --structured



(演習4)稼働時間チェックの作成

	Google Cloud Platform	3.
♠	ホーム	
Emil	お支払い	
プロタ	ブクト 🔨	
STAC	KDRIVER	
	モニタリング	
st	デバッグ	

Monitoring Overview	Monitoring Overview			
Resources Alerting Uptime Checks Groups	Welcome to Sta Complete the Stackdriver get ⊕ Add GCP Projects ■ Email Re	ting started checkl	1onitoring ist to gain deeper	g! r insights into your system.
Dashboards	Install Stackdriver age Collect enhanced metrics	nts from your	Create u Verify the	ptime checks availability of your services and
Debug	system, services, and app	lications	Infrastruc CREATE C	ture worldwide CHECK
Trace				
Logging				
Error Reporting	Resource dashboards	0	^	Incidents @
Profiler	RESOURCE TYPE	TOTAL RESOURCE	s	
	Instances	1		Charts



(演習4)稼働時間チェックの作成

稼働時間チェックで使用するア ラート ポリシーを作成する

Create uptime checks から CREATE CHECK をクリックし、下 記設定で作成。 Title: My uptime Check Check Type: HTTP Resource Type: Instance Applies To: Single, www1

•	Create uptime chec	ks
	Verify the availability o infrastructure worldwi	f your services and ie
ſ	CREATE CHECK	

÷
7



Uptime Check Created

Would you like to create an Alert Policy for this check? Alert policies allow you to be notified via email, SMS, or any of our notification options when your check is unresponsive.



No thanks Create Alert Policy

(演習 4)アラート ポリシーの作成

- Untitled Condition フィールドにアラート ポリ シー条件のタイトル Uptime Health Check on My Uptime Check と入力
- Notification (Optional) で、Email を選択
- メールアドレスを入力してから、Add Notification Channel をクリック
- Name this policy パネルに My Uptime Check Policy と入力
- Save をクリックすると、ポリシーの概要が表示 されます

Sugge	ested title: Uptime Health	Check on My Uptime Check		
	METRIC	UPTIME CHECK	F	>
Та	rget 🗇			
Me	etric: check passed			
Re	source Type			
A	All .			
Up	otime check id			
N	/ly Uptime Check		*	I

Notifications (optional)



Name this policy

A policy's name is used in identifying which policies were triggered, as well as managing configurations of different policies.

My Uptime Check Policy



(演習 4)ダッシュボードとグラフの作成

- Stackdriver Monitoring Console で、[Dashboards]
 > [Create dashboard]の順に選択します。
- 右上隅にある [Add Chart] をクリックします。
- [Add Chart] ウィンドウで、[Metric] タブをクリックし ます。
- [Find resource type and metric] 見出しの下の [instance, cpu, usage, etc.] フィールドに「CPU」と 入力し、プルダウン リストから [CPU load (1m)] を 選択
- パネルの [Preview] セクションに新しいグラフが表示されたら、[Save] をクリックします。

Debug METRIC VIEW OPTIONS No dashboards have I Create Dashboard Public Charts	been created.
Create Dashboard Public Charts	
METRIC VIEW OPTIONS	
METRIC VIEW OPTIONS	
Find resource type and metric ③	
cpu	
apert oppleads cum/cou/load 15m	
CPU load (1m) gee instance _ Di	
agent googleapis.com/cpu/load_1m Ri	
CPU load (5m) oce instance UI	
Add Chart	
SPUload (Im) Like •	th sh ta to the sum
CPUIdad (Im)	in sh 1d Tw Tw the sum
DPUlicad (Im) Live MILTINE VARW OFFICIALS (THE VARWA (INVES)	th sh td tw th the sum
DPUlicad (Im) Live MILTINE View controles (The server streng)	th th Td Te Ter tw man
DRUTING (Int) Live MICTOR verw opmous	th th Td Te Ter tw sum
Devices (Int) Live	th sh to to two sum
Professional (Imp) Live Transmitting Find resource type and metric (O finosurce type (GE VM Instance O inter (De VM	Un one tod two two more more than the today of
Professional (Imp) Live Interview (Photos Find resource type and metric © Resource type (Interview) Metric (CPU Med (Imil@) Metric (CPU Med (Imil@)	00 mb 1d te 1m no no no 01 01 02
CRUIted (Inte) Live Transverse presed) Find resource type and metric ©	00 mb 1d te 1m for nor 00 00
CPUTIos: (Imp) Lie MICTINE value certools Find resource type and metric ()	00 00 14 10 10 10 00 00 01 01
CRUIted (Im) Lie	0) 00 14 14 19 19 00 00 0) 0)
CPUIded (Im) Life MCTRE VERV CPTIONS A A Find resource type and metric () A A anti- (DPUIdeDTING) A Fild a state A A anti- (DPUIdeDTING) A	01 10 10 1d 10 10 10 00 00 01 01 01 01 01 01 01 01 01 01 01 01 0
Produces (Inn) I Ue Find resource type and metric © Resource type (Inn) Find resource type (Inn) Resource type (Inn) Find resource type (Inn)	01 mb 1d 1w 1w mm mm 01 01 01 01 01 01 01 01 01 01 01 01 01



(演習 4)ダッシュボードとグラフの作成

- 2つ目のグラフを作成するには、[Add Chart] をクリックします。
- [Metric] タブをクリックします。
- [Find resource type and metric] プルダウ ンリストで [Received bytes] を選択しま す。他のフィールドはデフォルト値のままに します。
- [Preview] セクションに新しいグラフが表示 されたら、[Save] をクリックします。
- 新しいダッシュボードで [Untitled Dashboard] を Stackdriver quickstart dashboard に変更します。

Add Chart





(演習 4)ログの表示

- GCP コンソールより、Stackdriver Logging をク リック
- ログビューアの設定を変更して、目的のログを表示します。
 - 最初のプルダウン リストで、[GCE VM Instance, www1]を選択します。
 - 2番目のプルダウン リストで [syslog] を選 択し、[OK] をクリックします。
 - 他のフィールドはデフォルト値のままにします。VM インスタンスのログが表示されます。



キャンセル



インスタンスを削除する(最後に実行)

- インスタンスー覧画面より、削除するインスタンス名の左側の チェックボックにチェックを入れる
- インスタンスー覧画面の上部にある削除ボタンをクリック


Compute Engine 参考 URL(1/2)

- オンラインマニュアル
 - <u>https://developers.google.com/compute/</u>
- Quick Start Guide
 - <u>https://developers.google.com/compute/docs/signup</u>
- ロードバランサーの設定
 - <u>ネットワークロードバランサー</u>
 - <u>HTTPロードバランサー(リージョンまたぎ)</u>
 - <u>HTTPロードバランサー(コンテンツベース)</u>
- Docker on GCE
 - <u>https://developers.google.com/compute/docs/containers/container_vms</u>



Compute Engine 参考 URL(2/2)

- オートスケーラー
 - <u>オートスケーラー</u>
- サポート
 - → オペレーション情報(障害、計画停止等) gce-operations
 - ディスカッション <u>gce-discussion</u>
 - <u>Security Bulletins</u>
 - コミュニティ(<u>Stack Overflow</u>)
- 費用
 - <u>価格表</u>
 - <u>料金計算ツール</u>





はじめてみようGCP ハンズオンセミナー 地域をまたがる負荷分散編





- Google Cloud Platformプロジェクトを作成していること
 - 作成してない場合は<u>こちら</u>
- 課金が有効になっていること
 - 有効になってない場合は<u>こちら</u>
- コマンドツールのインストールが完了していること
 - 完了していない場合は<u>こちら</u>



ハンズオンのシステム構成(今回は HTTP のみ)





インスタンスの設定

次の流れでトラフィックを処理するインスタンスを設定します。

- 1. 2つのリージョンに2つのインスタンスを構築
- 2. Apacheのインストールと設定
- 3. ファイヤーウォールの設定
- 4. 動作確認



VM インスタンスを作成する

- 1. Google Cloud Platform コンソールより [Compute Engine] > [VM インスタンス]を選択
- 2. [インスタンスを作成]ボタンをクリックする

https://console.cloud.google.com/





インスタンスの作成

- 1. [名前] に www-1、[ゾーン] を us-central1-b に設定
- 2. [管理、ディスク、ネットワーキング、SSH キー]をクリックすると、拡張設定が表示されます。
 [ネットワーキング]タブ下の [ネットワークタグ] フィールドに、http-tagを指定します。
- 3. [作成]をクリックします。

www-1	管理 セキュリティ ディスク ネットワーキング 単一テナンシー
リージョン 📀 ジーン 📀	
us-central1 (アイオワ) 🔹 us-central1-b	* * * * * * * * * * * * * * * * * * *
マシンタイプ カスタマイズしてコア、メモリ、GPUを選択します。	http-tag 🔞
vCPU x 1 ▼ メモリ 3.75 GB カスタマイズ	ホスト名 ② このインスタンスのカスタムホスト名を設定するか、デフォルトのままにします



夕前 🕗

インスタンスの追加

- 1. 同じ設定で www-2を追加します (www-1の詳細を開き、同様のものを作 成ボタンを使うと簡単に2つ目を作成できます)。
- 2. **www-3**と**www-4**をゾーン **europe-west1-b**に作成します(他の設定 は同じです)。

4	VM インスタンスの詳細	✔ 編集	ひ リセット	🔋 同様のものを作成	■ 停止	盲 削除
S 1	www-1					
詳細	モニタリング					
リモ- SSF	-ト アクセス - ト アクセス					
3	シリアルポート接続を有効化 📀					
ログ Stacl	kdriver Logging					
シリ ※ さ	アルポート1(コンソール) 5らに表示					



Apache のインストールと設定

- 1. [接続] 列の [SSH] ボタンをクリックして www-xに接続します。(xは1~4 で置き換えてください)
- 2. 次のコマンドでApacheのインストールと設定をします。
- 3. 4台全てに設定をしてください。

※付録の演習で実施しますが、イメージを利用してVMの設定をコピーすることもできます。

\$ sudo apt update \$ sudo apt install -y apache2 \$ hostname=`hostname` \$ echo '<!doctype html><html><body><h1>'\$hostname'</h1></body></html>' | sudo tee /var/www/html/index.html



Google Cloud Shell を起動

<u>管理コンソール</u>から[Google Cloud Shell] を起動する。右上のアイコンをクリッ クする。

gcloud コマンドなどGCPの操作に必要なツールが予めインストールされている。







ファイヤーウォールの設定

- 1. [ネットワーキング] > [VPC ネットワーク] > [ファイアウォール ルール] を選択し ます。
- 2. [ファイアウォール ルールを作成]をクリックします。
- 3. [名前] として www-firewall と入力します。
- 4. [ネットワーク] に default を選択します。
- 5. [ターゲットタグ]を http-tag に設定します。
- 6. [ソースIPの範囲]を 0.0.0.0/0 に設定します。
- 7. [プロトコルとポート]を tcp:80 に設定します。
- 8. [作成]をクリックします。





インスタンスの動作確認

- 1. [VM インスタンス] に移動し、[外部IP] 列のインスタンスのIPアドレスをコ ピーします。
- 2. 以下のいずれかの方法で動作を確認します。
 - a. ブラウザのアドレス欄にIPアドレスをペーストして確認
 - b. curl [外部IP] コマンドで確認

名前へ	ゾーン	マシンタイプ	アドバイス	使用中	内部 IP	外部 IP	接続		
🥑 www-1	us-central1-b	vCPU1個、3.75	GB		10.128.0.2	146.148.37.111	SSH	•	
🥑 www-2	us-central1-b	vCPU1個、3.75	GB		10.128.0.4	104.197.241.31	SSH	3 9 8	
🥑 www-3	europe-west1-b	vCPU1個、3.75	GB		10.132.0.2	104.155.59.102	SSH		•
🥑 www-4	europe-west1-b	vCPU1個、3.75	GB		10.132.0.3	104.199.11.16	SSH	390	



ロードバランサーの準備

次の流れで作成します。

- 1. グローバル静的外部IPアドレスの予約
- 2. インスタンスグループの作成
- 3. ヘルスチェックの設定



グローバル静的外部 IP アドレスの予約

- 1. [ネットワーキング] > [VPC ネットワーク] > [外部IPアドレス] を選択
- 2. [静的 IP を予約]をクリックします。
- 3. [名前] として **Ib-ip-cr** を割り当てます。
- 4. [タイプ]を[グローバル]に設定します。
- 5. [予約]をクリックします。





インスタンスグループの作成

- 1. [Compute Engine] > [インスタンス グループ] を選択します。
- 2. [インスタンス グループを作成]をクリックします。
- 3. [新しい非マネージド インスタンス グループ]を選択します。
- 4. [名前]をus-resources-wに設定します。
- 5. [ゾーン]をus-central1-bに設定します。
- 6. [ネットワーク] で、**default** を
- 7. [VM インスタンス] で、www-1 と www-2 を選択します。
- 8. [作成]をクリックします。

Google Cloud



default	
サブネットワーク 📀	
default (10.128.0.0/20)	
VM インスタンス	

www-1

インスタンスグループの作成

- 1. 同様の要領で、ヨーロッパのインスタンスグループを作成します。
 - a. [名前]: europe-resources-w
 - b. [ゾーン]: **europe-west1-b**
 - c. [インスタンス]: www-3 と www-4
- 2. 2 つのインスタンスにそれぞれ 2 つのインスタンス グループが作成されたことを確認してください。

名前へ	ゾーン	インスタンス	テンプレート	自動スケーリング	使用中
🥝 europe-resources-w	europe-west1-b	2	22		
🥝 us-resources-w	us-central1-b	2	855		



ヘルスチェックの作成

- 1. [Compute Engine] > [ヘルスチェック] を選択
- 2. [ヘルスチェックを作成]をクリックします。
- 3. [名前] に、http-basic-check を入力します。
- 4. [プロトコル] フィールドを [HTTP] に設定します。
- 5. 残りのフィールドはデフォルト値をそのまま使用します。
- 6. [作成]をクリックします。

名前 😰	
http-basic-check	
説明 (省略可)	
	7
プロトコル	
HTTP	×
ポート 🔞	
80	
リクエストパス 🛞	
1	



負荷分散サービスの設定

- 1. [ネットワーキング] [ネットワークサービス] > [負荷分散] を選択
- 2. [ロードバランサ]を作成をクリック
- 3. [HTTP(S)負荷分散]の[設定を開始]をクリック
- 4. ロードバランサの [名前] に web-map を入力します。





バックエンドの設定

- 1. [バックエンドの設定]をクリックします。
- 2. [バックエンド サービスを作成または選択] プルダウン メニューで [バックエンド サービスを作成]を選択します。
- 3. バックエンド サービスの [名前] を web-map-backend-service に設定します。
- 4. [インスタンス グループ]を us-resources-w に設定します。
- 5. 残りのフィールドはデフォルト値をそのまま使用します。
- 6. [バックエンドを追加]をクリックします
- 7. 同様に[インスタンスグループ]を europe-resources-w に設定します。
- 8. [ヘルスチェック]を http-basic-check に設定します。
- 9. [作成]をクリックします。



ホストとパスのルール

1. [ホストとパスのルール]をクリックします。

ここでは、すべてのトラフィックをデフォルトのルールで実行するため、ホストやパスのルールは設定しません。

ホストとパスのルール

ホストとパスのルールはトラフィックを振り向ける方法を決定します。トラフィックはパックエンド サービスに振り向け ることができます。ホストとパスのルールに明確に一致しないトラフィックはすべて、最初の行で選択された既定のサー ビスに送信されます。

一致しないすべてのURLバス		web-map-backend-service	•
	+ ホストとパスのルールを追加		

※ 設定テストを表示する



フロントエンドの設定

- 1. [フロンドエンドの設定]をクリックします。
- 2. [プロトコル] フィールドで **HTTP** を選択します
- 3. [IPアドレス] で、以前に作成した **Ib-ip-cr** を選択します。
- 4. [ポート] に 80 を選択します。

新しいフロントエンドの)Pとポート 🗊	^
名前 (吉路可) 🔞		
小文字、空白なし		
説明を追加		
プロトコル 💮		
HTTP		-
す。標準階層が機	がインシュントンではなジランジョンに成しているパック 線するのは、1 つのリージョンに成しているパック)みです。	2
IPパージョン	IPアドレス	
iPv4 -	lb-ip-cr (34.98.104.109)	*
ポート		
80		*
完了 キャンセル		





1. [確認と完了]をクリックします。

HITP

130.211.46.248.80

- 2. 現在の設定と作成しようとしている内容を比較します。
- 3. すべて問題なければ、[作成] をクリックして HTTP(S)ロードバランサを作 成します。

数分かかります。	確認と完了		負荷分散	= -	-ドバランサを作成	℃ 更新	盲 削除	
	バックエンド							
	パックエンド サービス			ロードバランサ	バックエンド	フロントエンド		
	1. web-map-backend-service エンドポイント プロトコル: HTTP 名前付きオ	ート:http タイムアウト:30秒 ヘルスチェック:1	http-basic-check	Act Martine	1992.00	a 1 1		
	♡ 詳細設地							
	インスタンスグルーフィ ダーン	日間スケーリング 分散モード	13 🖩	🚍 名前また(まプロトコルでフ	ィルタ		
	europe-resources-w europe-west1-b	オフ CPU 最大使用率:80%	100%					
	us-resources-w us-central1-b	7.7 CPU 最大使用率: 80%	100%					
	ホストとパスのルール			名前	プロトコル ^	バックエンド		
	ホスト ヘ パス	パックエンド		web-map	HTTP	🥝 1個のバックエ	ンドサービス(2	2 個のインスタンス
	不一致すべて (原定) 不一致すべて (現定	() web map-backend service.		and a second sec		NTO NEWSFILM		
	フロントエンド							
	プロトコル ヘー IP:ボート 証明	*						



インスタンスへのトラフィックの送信

- 1. web-map をクリックして作成したロードバランサを展開します。
- 2. [バックエンド] セクションの [正常] 列を確認し、インスタンスが正常である ことを確認します。
- 3. 4つとも正常になったら、[フロントエンド] セクションから [IP:ポート] をコ ピーし、ブラウザに貼り付けます。(正常にならなくてもつながります。)

バックエンド

バックエンド サービス

1. web-map-backend-service

エンドポイント プロトコル: HTTP 名前付きボート: http タイムアウト: 30 秒 ヘルスチェック: http-basic-check

> 詳細設定

インスタンス グループ 🔺	ゾーン	正常	自動スケーリング	分散モード	容量
europe-resources-w	europe-west1-b	2/2	オフ	CPU 最大使用率: 80%	100%
us-resources-w	us-central1-b	2/2	オフ	CPU 最大使用率: 80%	100%



ブラウザからの動作確認

1. ブラウザに貼り付けた後リロードすると、応答するバックエンドのインスタ ンスが切り替わることが確認できます。





地理的に近い場所からの動作確認

最も近い地域のインスタンスから応答があります。

- us-central1-bにあるインスタンスにsshでログインして、curl 外部IP を実行して、結果を確認します。
- 2. europe-west1-bにあるインスタンスにsshでログインして、curl 外部IP を 実行して、結果を確認します。

<!doctype html><html><body><h1>instance-1</h1></body></html>

\$ curl http://130.211.46.248/

<!doctype html><html><body><h1>instance-2</h1></body></html>

\$ curl http://130.211.46.248/
<!doctype html><html><body><h1>instance-3</h1></body></html>
\$ curl http://130.211.46.248/
<!doctype html><html><body><h1>instance-4</h1></body></html>



^{\$} curl http://130.211.46.248/

TIPS: RPSの確認

web-map ロードバランサの [モニタリング] をクリックするとバックエンドのRPS を確認できます。バックエンドからバックエンドを指定するとどのリージョンから どのリージョンに接続しているか確認できます。





付録:東京リージョンにインスタンスを移動

- 1. europe-west1-b にある www-3 と www-4 のイメージを作成
 - a. インスタンスを停止
 - b. インスタンスの編集
 - c. [Boot Disk] > [インスタンスを削除したときの動作] をディスクを維持 に変更
 - d. インスタンスの削除
 - e. [Compute Engine] > [イメージ]より www-3, 4 のソースディスクからイメージを作成



付録:東京リージョンにインスタンスを移動

- 1. イメージからインスタンスを asia-northeast1-b に作成
 - a. イメージを選択
 - b. インスタンスの作成を選択
 - c. www3,4 と同じ設定で作成 (インスタンス名は www3-tokyo, www4-tokyo、タグに http-tag の設定を忘れずに)



付録:東京リージョンにインスタンスを移動

- 1. インスタンスグループの設定やロードバランサへの組み込みを実施
- 2. 手元のブラウザからアクセスをして instance-3, instance-4 から結果が 帰ってくることを確認



Clean-up

- [Compute Engine] > [VM インスタンス]より作成した4つのVMをすべて削除 します。
- 2. [ネットワークサービス] > [負荷分散] から作成したロードバランサを削除しま す。同時にバックエンド サービス web-map-backend-service にもチェック を入れて一緒に削除します。
- 3. [外部IPアドレス] より **Ib-ip-cr** を解放します。
- 4. [ファイアウォールルール] www-firewall を削除します。
- 5. [インスタンス グループ] europe-resouces-w と us-resources-w を削除します。
- 6. [Compute Engine] > [ヘルスチェック] より http-basic-check を削除します。



Clean-up:付録

- 1. [Compute Engine] > [ディスク] よりイメージ作成のために使った2つの ディスクをすべて削除します。
- 2. [イメージ]より作成したイメージを削除します。





はじめてみよう GCP ハンズオンセミナー Kubernetes Engine 編



準備

- Google Cloud Platformプロジェクトを作成していること
 - 作成してない場合は<u>こちら</u>
- 課金が有効になっていること
 - 有効になってない場合は<u>こちら</u>
- "Google Container Engine API", "Google Compute Engine" およびその関連APIが有効になっていること
 - 有効になっていない場合は<u>こちら</u>をクリック







文化:開発と運用を高速に繰り返し行うための組織や考え方 技術: DevOps 文化を支えるための技術やツール

C Google Cloud

DevOps における技術要件の変化

- 1. 手動から自動化へ
 - a. オペレーション数削減
 - b. 再現性の向上
- 2. 自動化からアーキテクチャ最適化へ
 - a. オペレーションしやすいアーキテクチャへの変更


DevOps を支える技術

- 1. マイクロサービスアーキテクチャ
- 2. コンテナとオーケストレーション
- 3. CI / CD
- 4. 高度な監視(フィードバックループの実現)



マイクロサービスアーキテクチャとは?

従来のモノリシック(一枚岩)なアプリケーションとは異なり、 役割や責務ごとに機能を小さなサービスに分割し、それらのサービスが 一つのシステムとして動作する状態を目指したアーキテクチャ。







マイクロサービスアーキテクチャのメリット

- ビルド・デプロイプロセスがサービス単位でシンプルに
- コードベースが小さく、エンジニアの初期学習コスト低
- サービス間が疎結合になっているので新たな技術を導入しやすい

DevOps 的に開発と運用を高速に繰り返すには最適なアーキテクチャ



コンテナとは?

コンテナはアプリケーションコードとその依存性を一つのユニットとしてまとめる

これにより、アプリケーションとインフラを疎結合にすることができる コンテナはアプリケーションとその依存性がまとまっているので、例えば、開発 環境、テスト環境、本番環境をまたいだデプロイが容易になる オンプレミス、プライベートクラウド、パブリッククラウド等ことなる実行環境間 の移動が容易になる





VM vs コンテナ

VM







DevOpsとも親和性が高いコンテナ

軽量	ポータブル	効率性

仮想マシンに比べて 軽量でシンプル。数十ミリ秒 で起動

様々な実行環境に対応し、 デプロイメントが容易

リソース使用量が少なく、コ

ンピュートリソースを細分化 して効率的に利用可能



コンテナ管理の課題



- 複数のノードに対するコンテナのデプロイは?
- ノード障害が発生した場合は?
- **コンテナ障害**が発生した場合は?
- アプリケーションのアップグレードはどうやって管理する?

Google Cloud

Kubernetes(k8s)



OSS のコンテナオーケストレーションシステム Google 内部で使われている Borg をインスパイア オンプレでもクラウドでも運用可能

kubernetesの特徴

- 複数のコンテナに対する管理機構
- manifest による宣言的な定義/管理
- オートスケール
- ローリングアップデート・自己修復



Scheduled and packed dynamically onto nodes



Google Kubernetes Engine (a.k.a GKE)



- GCP 上で動作する kubernetes のマネージドサービス
- マスターノードの管理は全て Google が行う
- 自動アップグレード
- GCP の各種サービスとのインテグレーション



CI/CDとは?

<u>継続的インテグレーション(CI: Continuous Integration)</u>

ー日に何度もビルドを実行し、ソフトウェアをインテグレーションした時に発生する様々な問題を早期に検出する。フィードバックループを短くし、ソフトウェア開発の品質と生産性を向上させる

<u>継続的デリバリー(CD: Continuous Delivery)</u>

ソフトウェア全体のライフサイクルを通じて、常に本番環境にリリースできる状態を保つ リリースまでの期間を短縮することで、フィードバックループを短くできる







GCPでCI/CD



Cloud Build



デベロッパーフレンドリー

開発者が使い慣れた Git のイベントに応じたトリガー

柔軟なビルドステップ

あらゆるサーバーレス CLI ツールをビルドステップとして 組み込むことが可能

Spinnaker との連携

GKE 環境での B/G デプロイ、カナリーリリースへの対応

サーバーレスな CI/CD プラットフォーム

お客様が VM を用意したりキャパシティの管理をする必要はない



Cloud Build によるワークフロー





ハンズオンのシステム構成



• 管理コンソール > [Kubernetes Engine] を選択

٢	Kubernetes Engine	>	Q	Kubernetes engine
<pre></pre>	Deployment Manager	>	٢	Kubernetes クラスタ Kubernetes Engine
-@.	App Engine	>	٨	Kubernetes Engine
007		~~	~	



クラスタを作成する

• [コンテナ クラスタを作成] ボタンをクリック





クラスタテンプレートから "標準クラスタ"を 選択し、コンテナクラスタの情報を入力する

- 名前:"cluster-1"
- ロケーションタイプ:"ゾーン"
- ゾーン: "asia-northeast1-a"
- マシンタイプ: "vCPU x 1"
- サイズ(クラスタ内のノード数): "3" "可用性、ネットワーキング、

セキュリティ、その他の機能"をクリック

クラスタ テンプレート 「標準クラスタ」テンプレート #前欄成を含むテンプレートを選択するか、ニーズに合わ 離聴的インテグレーション、ウェブサービス、バックエンド。さらにカスタ せてテンプレートをカスタマイズします。 マイズする場合や、何を選択したらよいかわからない場合に最適な選択肢で 4. 名前 💮 既存クラスタのクローンを作成 cluster-1 する フィールドの入力に使用する股存クラスタ ロケーションタイプ ① を1つ選択してください y-y リージョン 標準クラスタ y-> 0 継続的インテグレーション、ウェブサービ asia-northeast1-a ス、パックエンド。さらにカスタマイズす る場合や、何を選択したらよいかわからな マスターのバージョン い場合に最適な選択肢です。 1.12.8-gke.6 (デフォルト) 最初のクラスタ ノードプール Kubernetes Engine をテストして、最初の ノードプールは、クラスタ内で Kubernetes を実行している独立したイ アプリケーションをデプロイします。作業 ンスタンス グループです。複数のゾーンにノードプールを追加して可 を開始するのに手頃な選択肢です。 用性を高めたり、さまざまなタイプのマシンで構成されたノードプー ルを追加したりできます。ノードプールを追加するには、[編集]をク リックします。詳細 CPU 使用率の高いアプリケーシ ヨン default-pool ウェブクローリングなど、多くの CPU を ノード数 必要とする処理。 3 メモリ使用量の多いアプリケー マシンタイプ 〇 カスタマイズしてコア、メモリ、GPUを選択します ション vCPU x 1 * メモリ 3.75 GB データベース、分析、Hadoon、Soark、 ETLなど、多くのメモリを必要とする処 自動アップグレード:オン その他の設定項目 GPU による高速コンピューティ 24 + ノードブールを追加 機械学習、動画のコード変換、科学計算な ど、多くのコンピューティングを必要とす る、GPUを利用可能な処理。 ※可用性。ネットワーキング、セキュリティ。その他の機

カスタマイズ



続きのパラメータを入力し、"作成"をクリック

サブネットワーク: "default"

cluster-1			
説明 (省略可)			
ゾーン 💿			
asia-east1-a			
マシンタイプ			
vCPU x 1	· ¥	メモリ 3.75 GB	カスタマイズ
サイズ 💮			
3			
合計コア数	vCPU 3 個		
合計メモリ	11.25 GB		

クラスタ インスタンスはエフェメラル ローカル ディスクを使用します。必要で あれば、ポッドに永続ディスクを接続できます。

*

サプネットワーク 🕥

default

Logging and monitoring 💿

Turn on Stackdriver Monitoring To use Stackdriver Monitoring for Instances, enable Stackdriver Monitoring for your project 12th

Turn on Stackdriver Logging



 コンテナクラスタが作成されたことを確認 (作成されるまで数分かかります)

コンテナ クラスタ

□ 名前	ゾーン	クラスタサイズ	合計コア数	合計メモリ	ノードのバージョン	
🗌 🥑 cluster-1	asia-east1-a	3	vCPU 3 個	11.25 GB	1.3.3	× 1





Kubernetes Cluster

Google Kubernetes Engine (GKE)



Google Cloud Shell を起動

<u>管理コンソール</u>から[Google Cloud Shell] を起動する。右上のアイコンをクリッ クする。

gcloud コマンドなどGCPの操作に必要なツールが予めインストールされている。







kubectl コマンドをインストールする

• [Google Cloud Shell] にて以下のコマンドを実行

\$ sudo gcloud components update kubectl

TIPS: gcloud components list コマンドを実行するとインストールされたコン ポーネントの一覧を取得することができます。



コマンド環境を設定する

以下のコマンドを実行して、コマンドの実行環境を設定します。

\$ export PROJECT ID="\$(gcloud config get-value project -q)"

- \$ gcloud config set project \${PROJECT_ID}
- \$ gcloud config set compute/zone asia-northeast1-a
- \$ gcloud config set container/cluster cluster-1
- \$ gcloud container clusters get-credentials cluster-1









アプリケーションを作成

アプリケーション用のディレクトリを作成します

\$ mkdir helloNode

\$ cd helloNode



アプリケーションを作成

 以下のコマンドを実行して、 server.jsというファイルをダウンロードし確認します。

\$ wget

https://raw.githubusercontent.com/GoogleCloudPlatform/gcp-getting-started-lab-jp/
master/fundamental/gke/helloNode/server.js

\$ cloudshell edit server.js

#「Hello World!」とホスト名を出力する単純な Webアプリ。



アプリケーションの実行確認

- 以下のコマンドを実行
- \$ node server.js

別のコンソールより curl localhost:8080 をするとメッセージが出力されること を確認 (もしくは「プレビューのポート」)

Ctrl-C にてサーバプロセスを終了



Dockerコンテナイメージを作成

以下のコマンドを実行して、
 Dockerfile というファイルをダウンロードし確認します。

\$ wget

https://raw.githubusercontent.com/GoogleCloudPlatform/gcp-getting-started-lab-jp/
master/fundamental/gke/helloNode/Dockerfile

\$ cloudshell edit Dockerfile



Dockerコンテナイメージを作成

\$ docker build -t asia.gcr.io/\${PROJECT_ID}/hello-node:v1.

Sending build Step 1 : FROM	context to Docker daemon 3.072 kB node:0.12		
0.12: Pulling	from library/node		
523ef1d23f22:	Extracting [====================================	1	34.6 MB/51.35 MB
140f9bdfeb97:	Download complete		
5c63804eac90:	Download complete		
ce2b29af7753:	Download complete		
5c2bdca41b86:	Downloading [====================================]	114.6 MB/128.6 MB
f417df1119e6:	Download complete		
d36821cb651a:	Download complete		
48d9fce985a8:	Download complete		
d09c6f7639f7:	Download complete		

ビルドが完了するまで数分かかります。



Dockerコンテナイメージを作成

 Google Container Registry にPushします。 (初回は時間がかかります)

\$ docker push asia.gcr.io/\${PROJECT_ID}/hello-node:v1

※ docker push [Push 先ホスト名]/[プロジェクト名] / [イメージ] : [タグ]

• Google Container Registry にPush されたことを確認します。

\$ gcloud container images list-tags asia.gcr.io/\${PROJECT_ID}/hello-node



Docker コンテナイメージを作成 (Cloud Build)

 Google Container Registry にPushします。 (初回は時間がかかります)

\$ gcloud builds submit --tag asia.gcr.io/\${PROJECT_ID}/hello-node:v1 .



Dockerコンテナレジストリを確認

• 管理コンソール > [ツール] > [Container Registry]

	Google Cloud Platform	🗫 My First Project 👻	٩				
$\begin{pmatrix} 0\\ 0 \\ 0 \\ 0 \end{pmatrix}$	Container Registry	← イメージ C 更新 前除					
Ħ	イメージ	hello-node					
۵	設定	asia.gcr.io / my-first-project-244405 / hello-node Г 一 名前またはタグでフィルタ					
		名前 タグ					
		□ 🕹 81d2ce53df72 v1					





演習1. 作成したDocker コンテナを Cloud Shell から実行する

\$ docker images #ローカルに登録されているDockerイメージー覧を取得する

\$ docker run -d -p 8080:8080 asia.gcr.io/\${PROJECT_ID}/hello-node:v1 # ローカルの Dockerイメージからコンテナを起動する

\$ curl localhost:8080 #ローカルで起動したWebアプリケーションに接続する

\$ docker ps #ローカルで起動しているコンテナー覧を取得する

\$ docker kill <container id> #container idを指定してコンテナを停止する

\$ docker rm <container id> #container idを指定してコンテナを削除する







Deploymentを作成

• プロジェクト ID を確認

```
$ gcloud config list core/project
[core]
project = myproject-0505
Your active configuration is: [cloudshell-9999]
```



Deploymentを作成

 以下のコマンドを実行して、hello-node-deployment.yaml というファイル をダウンロードします。

\$ wget

https://raw.githubusercontent.com/GoogleCloudPlatform/gcp-getting-started-lab-jp/
master/fundamental/gke/helloNode/hello-node-deployment.yaml

Cloud Shell コードエディタより、hello-node-deployment.yaml を選択し、
 <project id>を自分の project id に変更

\$ cloudshell edit hello-node-deployment.yaml




Deploymentを作成

• kubectl apply コマンドを使ってDeploymentを作成します。

\$ kubectl apply -f hello-node-deployment.yaml

作成されたDeployment、Pod を確認します。

\$ kubectl get deployment

\$ kubectl get pods -o wide







演習

演習1. Pod が動作しているNodeにログインしてDockerコンテナが動いている ことを確認しましょう。

演習2.動作中のDockerコンテナを削除してください。どのような動きになるかを確認しましょう。



Service を作成

以下のコマンドを実行して、hello-node-service.yaml というファイルをダウン ロードします。

\$ wget

https://raw.githubusercontent.com/GoogleCloudPlatform/gcp-getting-started-lab-jp/
master/fundamental/gke/helloNode/hello-node-service.yaml



Service を作成

• kubectl apply コマンドを使ってServiceを作成します。

\$ kubectl apply -f hello-node-service.yaml

• 作成されたサービスを確認します。

\$ kubectl get svc

ネットワークロードバランサーを作成するため、外部 IPアドレスが割り 当てられるまでにしばらく時間がかかります。



ここまでの構成







サービスの外部IPアドレスを確認します。 (外部 IP アドレスが割り当てられるまで、少し時間がかかります。)

\$ kubectl get svc							
NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE		
hello-node	LoadBalancer	10.0.12.198	12.34.567.89	8080:30189/TCP	58s		
kubernetes	ClusterIP	10.0.1	<none></none>	443/TCP	47m		

curl コマンドでアクセスして出力を確認します。 (Webブラウザでもアクセス可能)

\$ curl <EXTERNAL IP>:8080 # <EXTERNAL IP> を上記で確認した IP アドレスに変更



ポッドをスケールアウト

• kubectl scale コマンドを使うとポッドのレプリカ数を変更できます。

\$ kubectl scale deployment/hello-node --replicas=3

ポッドの数を確認します。

\$ kubectl get pods -o wide



ここまでの構成





(演習)バージョン変更

- 1. アプリケーション(server.js)が出力するメッセージを変更
- 2. Docker イメージをビルドして v2 とする
- 3. Docker push で新しいバージョンをレジストリに登録
- 4. kubectl edit deployments/hello-node でイメージを変更
- 5. kubectl describe deployments/hello-node でバージョンが変わっていく 様子を確認



Cleanup - Kubernetesコンポーネント

サービスを削除します。

\$ kubectl delete svc/hello-node

デプロイメントを削除します。

\$ kubectl delete deployment/hello-node



Cleanup - GKE クラスタ

• クラスタを削除します。

コンテナ クラスタ 🚹 クラスタを作成 💼 削除

コンテナ クラスタ

□ 名前	ゾーン	クラスタサイズ	合計コア数	合計メモリ	ノードのバージョン	_	
🗌 🥝 cluster-1	asia-east1-a	3	vCPU 3 個	11.25 GB	1.3.3	/ ī	î

削除アイコンをクリック



Cleanup - Container Registry

- 管理コンソール > [Storage] を選択します。
- バケット名 asia.artifacts.<project id>.appspot.com を選択して削除ボ タンをクリック

	Storage	ブラウザ	🕞 パケットを作成	C更新	∎ 削除
	ブラウザ	パケット		2. 削隊	ネボタンをクリック
#	Transfer	三名前 図 asia.artifacts.fukud	ak-ocean.appspot.com		
		し 1. バケット	· を選択		





- Pod のstdout/stderr を確認する
- \$ kubectl logs <POD_NAME>

• クラスタのメタ情報を確認する

- \$ kubectl cluster-info
- Pod の状態を確認する
- \$ kubectl describe pods/<pod name>





はじめてみよう GCP ハンズオンセミナー App Engine 編



準備

- Google Cloud Platformプロジェクトを作成していること
 - 作成していない場合は<u>こちら</u>
- 作業用の仮想マシンを一台準備します
 - 作成していない場合は<u>こちら</u>
- Cloud Source Repositories APIを有効にします
 - メニューで「APIとサービス」=>「ライブラリ」と進み、Cloud Source Repositories APIを検索して有効にします



Google Cloud Shell を起動

<u>管理コンソール</u>から[Google Cloud Shell] を起動する。右上のアイコンをクリッ クする。

gcloud コマンドなどGCPの操作に必要なツールが予めインストールされている。







コードを作成する(gitコマンドで取得)

githubからコードを取得する (gitコマンドを利用する場合)

• Cloud Shell 上で、以下のコマンドを実行

\$ git clone <u>https://github.com/GoogleCloudPlatform/python-docs-samples</u>

\$ cd python-docs-samples/appengine/standard/hello_world \$ ls app.yaml main.py main_test.py



コードを作成する

app.yaml (設定ファイル)の中身を確認します

 Cloud Shellで以下のコマンドを実行し、 コードエディタでファイルを開きます

\$ cloudshell edit app.yaml



コードを作成する

• [File]タブから[Save]を選択するか、Ctrl+Sで保存

₹	Cloud Shell	
D 2	File Edit Selection View Go I EXPLORER You have not yet opened a workspace. Open Workspace Open Workspace	Help app.yaml × 1 runtime: python27 2 api_version: 1 3 threadsafe: true 4 5 handlers: 6 - url: /.* 7 script: main.app 8



コードを作成する

\$ cloudshell edit main.py





ローカルテストを行う

• dev_appserver.pyコマンドを使ってローカル環境でコードをテスト

\$ dev_appserver.py .

• default moduleがlocalhostの8080ポートで起動したことを確認

```
$ dev_appserver.py .
INFO 2019-06-22 14:06:04,144 devappserver2.py:278] Skipping SDK update check.
INFO 2019-06-22 14:06:14,741 dispatcher.py:256] Starting module "default"
running at: http://0.0.0.0:8080
INFO 2019-06-22 14:06:14,743 admin_server.py:150] Starting admin server at:
http://0.0.0:8000
```





Cloud ShellのWeb Preview機能を使い、ローカル(Cloud Shell)の8080
 ポートへアクセスし、Hello Worldが表示されることを確認する





App Engine にデプロイする

- (Devサーバを動かしている場合は、"Ctrl + C" で停止)
- gcloud app deploy コマンドを使って、App Engineにデプロイする

\$ pwd

~/python-docs-samples/App Engine/standard/hello_world

\$ gcloud app deploy



App Engine にデプロイする

- Console > App Engine > バージョン を参照する
- 先ほどデプロイしたバージョン名をクリックする
- Webブラウザ上に"Hello World"が表示されることを確認する

	Google Cloud Platform	💲 My First Project 👻		٩					
- ()-	App Engine	バージョン	C 更新	∎ 削除	■ 停止	▶ 起重) 🍂 Þ. 5	シフィー	ックを移行
999 X	ダッシュボード サービス	盲 バージョンをフィル	<i>ッ</i> タ						
0	パージョン	□ バージョン	ステータス	トラフィック害	りり当て		インスタンス	0	ランタイム
B	インスタンス	□ 20180823t000323 ▷	提供中			100%	0		python27



コードを作成する

main.py のコードを確認します。

\$ cat main.py

もしくは、

\$ cloudshell edit main.py



(演習 1)バージョンコントロール

アプリケーションが出力するメッセージを変更して、新しいバージョンとしてデ プロイしてみてください。

- 1. main.py が出力するメッセージを変更
- 2. gcloud app deploy --no-promote コマンドにてデプロイ
- 3. コンソール バージョンにて新しいバージョンがデプロイされていることを 確認
- 4. アプリケーションを確認



(演習 1)バージョンコントロール

アプリケーションに負荷をかけながらバージョンコントロールしてみてください

1. Cloud Shell より以下のコマンドを実行

\$ while true; do curl <project-id>.appspot.com; echo; done

"トラフィック分割の基準"を "Cookie" にする



(演習 1)バージョンコントロール

۰Ô»	App Engine	← トラフィックを分割
5	ダッシュボード	受信トラフィックを複数のバージョンのアプリに分割することができます。トラフィックの分割は、新しいバージョンを時間をかけて公開する場合や、デザイン
*	サービス	や機能の A/B テストを行う場合に役立ちます。詳細 忆
0	バージョン	トラフィック分割の基準
8	インスタンス	
i	タスクキュー	トラフィック割り当て
Ō	cron ジョブ	20190622t231351 マク量: 20100622t231044 -
0	セキュリティ スキャン	201906221231944 - 30 %
	ファイアウォール ルール	
	割り当て	保存

Soogle Cloud

(演習 2)オートスケールを確認

アプリケーションに負荷をかけてサーバインスタンスが自動的にスケールする ことを確認

\$ sudo apt install apache2-utils

\$ ab -n 1000 -c 100 -t 30 http://<project-id>.appspot.com/

コンソール > [App Engine] > [インスタンス]よりインスタンス数を確認



(演習 2)オートスケールを確認

インスタンス C 更新 盲 削除





(演習 2)オートスケールを確認

インスタンス(自動的に追加/削除) 💿

QPS 過j	S 去1分	レイテンシ 過去 1 分	リクエスト数	エラー数	メモリ	開始時間	ログ	可用性
ø	16.133	2 ミリ秒	7221	0	30 MB	2016/10/07 5:46:02	表示	ダイナミック
0	16	1ミリ秒	2553	0	21.9 MB	2016/10/07 6:06:46	表示	ダイナミック
0	16.35	4ミリ秒	2536	0	29,8 MB	2016/10/07 6:06:46	表示	ダイナミック
0	15.817	1ミリ秒	2528	0	27.8 MB	2016/10/07 6:06:46	表示	ダイナミック
ø	15.883	2 ミリ秒	2520	0	13.8 MB	2016/10/07 6:06:46	表示	ダイナミック
0	0.1	8 ミリ秒	2451	0	27.9 MB	2016/10/07 6:06:32	表示	ダイナミック
ø	0.033	5ミリ秒	3	0	21.4 MB	2016/10/07 6:06:47	表示	ダイナミック
•	0	0ミリ秒	2	0	13,7 MB	2016/10/07 6:06:47	表示	ダイナミック



(演習 3)guestbook アプリを作成

https://cloud.google.com/App

Engine/docs/python/getting-started/creating-guestbook

※このアプリケーションをデプロイする場合、インデックス定義ファイルもデプ ロイする必要があります。またインデックスが作成されるまでしばらく時間がか かることがあります。

\$ gcloud app deploy app.yaml index.yaml



App Engine 参考URL

- オンラインマニュアル
 - <u>https://developers.google.com/App Engine/</u>
- Getting Started Guide
 - o <u>Java</u>
 - <u>Python</u>
 - <u>PHP</u>
 - <u>Go</u>



App Engine 参考URL

- サポート
 - コミュニティサポート <u>Stackoverflow</u>
 - システムステータス
 - <u>Downtimenotify</u> (障害情報)





はじめてみようGCP ハンズオンセミナー Cloud Storage 編


準備

- Google Cloud Platformプロジェクトを作成していること
 - 作成してない場合は<u>こちら</u>
- 課金が有効になっていること
 - 有効になってない場合は<u>こちら</u>
- "Google Cloud Storage", "Google Cloud Storage JSON API"サービス が有効になっていること
 - 有効になっていない場合は<u>こちら</u>
- コマンドツールのインストールが完了していること
 - 完了していない場合は<u>こちら</u>



Cloud Storage 管理コンソールにアクセス

• 管理コンソール > [ストレージ] > [Storage] を選択



Google Cloud

バケットの作成

- "バケットを作成" ボタンをクリック
- バケット名を入力 (グローバルで一意になっている必要あり)

バケット			
Cloud Storage では	、非構造化才	ブジェクトを「バケ	ット」という
コンテナに格納でき 信することや、他の Cloud Storage に格	fます。静的う) Google Clou 納することも	⁻ ータを Cloud Stora d Platform サービス できます。	ge から直接配 のデータを

← バケットの作成

🥝 バケットに名前を付ける

グローバルに一意の永続的な名前を選択します。 命名ガイドライン

my-first-project-244405

注意:機密情報を含めないでください

続行



- ストレージクラスを指定
- 場所を指定 (ここでは"アジア")
- "作成"をクリック

C Google Cloud

• デフォルトのストレージ クラスを選択する

Objects added to the bucket are assigned the selected storage class by default. Choose based on where and how often your objects will be accessed. Learn more

Multi-Regional 複数のリージョンで頻繁にアクセスされるデータに最適

Regional

単一リージョンで頻繁にアクセスされるデータに最適

O Nearline

アクセス頻度が月に1回以下のデータやバックアップに最適

O Coldline

アクセス頻度が年に1回以下のデータや障害復旧に最適

場所

asia (アジアの複数のリージョン)

•



ファイルをアップロードします

- "ファイルをアップロード"ボタンをクリック
- ローカルファイルシステムのファイルを選択

← バケットの詳	細 🧪 バケッ	バケットを編集 C		巨新
my-first-project-2444	05			
オブジェクト数 概要	権限 バケットロック			
ファイルをアップロード	フォルタをアップロード	フォルタを作成	保留を管理	削版
Q 前方一致でフィルタ				
パケット / my-first-project-	244405			

このパケット内にライブ オブジェクトはありません。オブジェクトのパージョニ ングを有効にしている場合は、このパケットにオブジェクトのアーカイブ パージ ョンが含まれている可能性がありますが、コンソールには表示されません。 gsutil または API を使用して、オブジェクトのアーカイブ パージョンを一覧表示 することができます。



ファイルのアップロードの確認

ファイルをアップロード フォルダをアップロード フォルダを作成 保留を管理 削除

Q、前方一致でフィルタ...

パケット / my-first-project-244405

名前	サイズ	タイプ	ストレージクラス	最終更新日
2.5_month.csv	228.12 KB	text/csv	Multi-Regional	2019/06/22, 23:44:16 UTC+9

1/1 のアップロードが完了しました		_ ×
2.5_month.csv	完了	×



Cloud Storage 参考 URL

- オンラインマニュアル
 - <u>https://cloud.google.com/storage/docs/overview</u>
- Getting Started
 - <u>https://cloud.google.com/storage/docs/getting-started-gsutil</u>
- コミュニティ
 - <u>StackOverflow</u>





はじめてみようGCP ハンズオンセミナー Cloud SQL 編



準備

- Google Cloud Platform プロジェクトを作成していること
 - 作成してない場合は<u>こちら</u>
- 課金が有効になっていること
 - 有効になってない場合は<u>こちら</u>
- "Google Cloud SQL"サービスが有効になっていること
 - 有効になっていない場合は<u>こちら</u>
- コマンドツールのインストールが完了していること
 - 完了していない場合は<u>こちら</u>



Cloud SQL インスタンスを作成する

- 管理コンソール > [ストレージ] > [SQL]を選択
- [インスタンスを作成する] ボタンをクリックする





Cloud SQL インスタンスを作成する

MySQLを選択し、インスタンス名とリージョンを指定

- [インスタンスID]: db1
- [リージョン]: asia-northeast1
 [作成] ボタンをクリック







Cloud SQL インスタンスを作成する

インスタンスが作成されたことを確認する

😂 SQL	インスタンス	🛨 インスタンスを作成	≢ データを移行
〒 インスタン	スのフィルタ		
1	ID 🚱	タイプ	
🗌 🥝 db1		MySQL 第2世代5	.7





ここでは、Compute EngineのLinuxインスタンスを使って、 Cloud SQLへの接続を確認します (接続用のGCEインスタンスを作成してください)

仮想マシンの外部IPアドレスを取得する

\$ gcloud compute instances list

Linuxインスタンスにsshログインする

\$ gcloud compute ssh <your-instance-name> --zone=<zone-name>





Compute Engineの仮想マシンにMySQL Clientをインストールします

\$ sudo apt update

\$ sudo apt install mysql-client





GCEのインスタンスがCloud SQLにアクセスできるようにします。

• [VCP ネットワーク] > [外部IP アドレス]から静的IPアドレスを予約

Ц	VPC ネットワーク	外部 IP アドレス	静的アドレ	_{レスを予約} C	更新 盲	静的アドレスを解放
82	VPC ネットワーク					
C	外部IPアドレス	〒 リソースをフィルタしま	ます			
RA	ファイアウォール ルール	名前 外部アドレス	リージョン	静的	バージョン	使用リソース
		- 35.243.95.9	asia-northeast1	エフェメラル	IPv4	VM インスタンス <u>instance-1</u> (ゾーン asia-northeast1-b)
N	ルート					
\$	VPC ネットワーク ピアリング					
×	共有 VPC					
\otimes	サーバーレス VPC アクセス					





承認済みネットワークを追加します

- [ストレージ] > [SQL] からSQLインスタンスを選択
- [接続]タブ下の[ネットワークを追加]をクリック
- VMインスタンスの外部IPアドレスを入力
- [完了]、[保存]を順にクリック



接続

データベース インスタンスへの接続方法を選択します

🗌 プライベート IP

プライベート IP 接続に追加の API と権限が必要です。この機能を有効にする、また は使用するには、組織の管理者への依頼が必要になる場合があります。現在は、プ ライベート IP 接続を有効にすると、削除できなくなります。

🗹 パブリック IP

承認済みネットワーク

変更を破棄

インスタンスに接続するネットワークを承認するか、プロキシを使用します。ネットワー クは、これらのアドレスを使用してのみ承認されます。詳細







VMインスタンスより、接続を確認します。

- SQLインスタンスの詳細からIPアドレスを確認
- VMインスタンスよりmysqlコマンドで接続

\$ mysql --host=<sql-instance-IP> --user=root --password

- Э -	このインスタンスに接続	
	パブリック IP アドレス	
	35.243.101.70	Ū
	インスタンス接続名	
	<pre>my-first-project-244405;asia-northeast1;db1</pre>	Ū
>.	Cloud Shell を使用して接続	
۲	Compute Engine VM インスタンスから接続してくた	ぎさい
Ø	すべての接続方法を表示	





データベース、テーブルを作成します。

CREATE DATABASE guestbook; USE guestbook;

CREATE TABLE entries (guestName VARCHAR(255), content VARCHAR(255), entryID INT NOT NULL AUTO INCREMENT, PRIMARY KEY(entryID));

INSERT INTO entries (guestName, content) values ("first guest", "hello world!");

SELECT * FROM entries;



Cloud SQL 参考 URL

- オンラインマニュアル
 - o <u>https://developers.google.com/cloud-sql/</u>
- Getting Started
 - <u>https://developers.google.com/cloud-sql/docs/getting-started</u>
- レプリケーション設定
 - <u>https://developers.google.com/cloud-sql/docs/replication?hl=j</u>
- <u>a</u> ● サポート
 - □ミュニティ Stack Overflow
 - 障害情報 / アナウンス google-cloud-sql-announce



(補足)レプリケーションの設定

レプリケーションの設定を行う

https://developers.google.com/cloud-sql/docs/replication?hl=ja

注意事項

- CHANGE MASTER TO MASTER_LOG_FILE...コマンド実行時にマスター サーバでSHOW MASTER STATUSを実行して、ログファイル、ログポジ ションを確認する
- SLAVE STATUSを確認しながら構成する
- MySQLのトラブルシュートができることが望ましい





はじめてみようGCP ハンズオンセミナー BigQuery 編



23FFC5

Google BigQuery 複雑なデータ分析をシンプルに

文→A

SQL

-

(O)

API

 \bigcirc

(—)

•

フルマネージド DWH 自動でスケールアウト 設定や管理は不要 最大 100,000 行 / 秒のストリーム サードパーティソフトウェアとの容易な統合 ペタバイトクラスのクエリも高速処理

クラウド型 データウェアハウス BigQuery

- BI ツールを使って非技術者でも操作可能
 - Looker, Google データポータル, Tableau, QlikView, Google
 Spreadsheet / Microsoft Excel
- ETL(データの再編集)
 - クエリー結果をテーブルに出力可能
 - JSON 関数/正規表現関数
- データ処理エンジンとの連携
 - Cloud Dataflow, Cloud Dataproc, Hadoop





17億件のフルスキャンを 6.7秒で実行

1	SELECT	
2	language,	
3	SUM(views) AS views	
4	FROM	
5	[bigquery-samples:wikimedia_pageviews.20]	1201]
6	WHERE	
7	NOT title CONTAINS ':'	
8	AND wikimedia_project='wp'	
9	GROUP BY	
10	language	
11	ORDER BY	
12	views DESC	/2 GBのノルスキャン
13	LIMIT	クエリのコスト:0072 TB * \$5/TB = 約36円
14	100	

Valid: This query will process 72.7 GB when run.





準備

- Google Cloud Platform プロジェクトを作成していること
 - 作成してない場合は<u>こちら</u>
- 課金が有効になっていること
 - 有効になってない場合は<u>こちら</u>
- "BigQuery API", "Google Cloud Storage", "Google Cloud Storage JSON API"サービスが有効になっていること
 - 有効になっていない場合は<u>こちら</u>
- コマンドツールのインストールが完了していること
 - 完了していない場合は<u>こちら</u>



BigQuery を使ってみる

<u>管理コンソール</u>から[ビッグデータ] > [BigQuery] のリンクをクリック

ビッグデータ

¢	BigQuery	Ŧ
**** ****	Pub/Sub	>
ß	Dataproc	>
Φ	Dataflow	
M	IoT Core	
Н	Composer	
** **	Genomics	>
	Dataprep	



BigQueryを使ってみる(サンプルの確認)

以下のURLにアクセスしてサンプルのデータセットを確認します。 https://console.cloud.google.com/bigguery?p=bigguery-public-data

※表示されていない場合には データを追加 -> 一般公開データセットを調べる -> 任意のデータを選択 -> データセットを表示 を選択してください

😑 Google Cloud Platform 💲 My First	Project 🔻
😡 BigQuery ベータ版 従来の UI に移動	
クエリ履歴	クエリエディタ
保存したクエリ	T
ジョブ履歴	
転送 🖸	
リソース 早 プロジェクトを固定	
Q、テーブルとデータセットを検索します 😡	
macro-mile-214114	
 bigquery-public-data 	
	● クエリを実行 -



BigQueryを使ってみる(サンプルの確認)

- 画面左側に表示される[リソース] > [bigquery-public-data]のsamples データセットをクリックする。
- その下にある [natality] テーブルを選択し、画面の右側に表示される テーブル構造(スキーマ)を確認する。

保存したクエリ	natality	Q D	テーブルをコヒ	-	
ジョブ履歴					
転送 🖸	スキーマ 詳細	プレビュー			
リソース 軍 プロジェクトを固定	フィールド名	タイプ	$\mathbf{t} - \mathbf{F}$	說明	
9. sample X	source_year	INTEGER	REQUIRED	Four-digit year	
	year	INTEGER	NULLABLE	Four-digit year	
- bigquery-public-data	month	INTEGER	NULLABLE	Month Index of	
google_analytics_sample	day	INTEGER	NULLABLE	Day of birth, st	
- 📰 samples	wday	INTEGER	NULLABLE	Day of the wee	
github_nested	state	STRING	NULLABLE	The two charac 2004 do not ini	
📰 gs od	is_male	BOOLEAN	REQUIRED	TRUE if the chi	
model	child_race	INTEGER	NULLABLE	The race of the White 2 - Black Japanese 6 - H	
shakespeare	weight_pounds	FLOAT	NULLABLE	Weight of the c	



BigQueryを使ってみる(クエリ実行)

[テーブルをクエリ]をクリックすると、テーブルに対してクエリを実行できます。





BigQueryを使ってみる(ハンズオン)

Natality テーブルに対してクエリーを実行し

以下の情報を調べてみましょう

問1: 何人分のデータが含まれているか?

問2: 何年から何年のデータが含まれているか?

問3:最も出生数が多い年は?

問4: 最も出生数が多い州は?

問5:年ごとの平均体重の推移は?

a. ヒント: AVG()関数で平均を算出する

※どのようなクエリや関数が実行できるかは、このマニュアルを参照してみてください。



BigQueryを使ってみる(問1:答え)

1. 何人分のデータが含まれているか?

SELECT COUNT(*)
FROM `bigquery-public-data.samples.natality`



BigQueryを使ってみる(問 2: 答え)

2. 何年から何年までのデータが含まれているか

SELECT year FROM `bigquery-public-data.samples.natality` GROUP BY year ORDER BY year



BigQuery を使ってみる(問 3: 答え)

3. 最も出生数が多い年は?

SELECT year, COUNT(*) as cnt FROM `bigquery-public-data.samples.natality` GROUP BY year ORDER BY cnt DESC



BigQueryを使ってみる(問 4: 答え)

4. 最も出生数が多い州は?

SELECT state, COUNT(*) as cnt FROM `bigquery-public-data.samples.natality` WHERE state IS NOT null GROUP BY state ORDER BY cnt DESC



BigQuery を使ってみる(問 5: 答え)

5. 年ごとの平均体重の推移は?

SELECT AVG(weight_pounds), year FROM `bigquery-public-data.samples.natality` GROUP BY year ORDER BY year



テーブルの作成

パブリックデータをダウンロードして、
 BigQueryのテーブルにロードし、テーブルの作成を体験します。

地震データ

http://earthquake.usgs.gov/earthquakes/map/


テーブルの作成(データのダウンロード)



🔼 Google Cloud

新規にデータセットを作成し、その下にテーブルを作成します。

- 1. <u>BigQuery GUIコンソール</u>を開く
- 2. プロジェクト名右横の"データセットを作成"を選択
- 3. データセット名を入力

例: earthquake

☞ エディタを非表示	[] 全面的表示	データセットを作成 ^{データセット ID}	
		earthquake	
		データのロケーション (Optional) 📀	
		デフォルト	•
	•	デフォルトのテーブルの有効期限 🕥	
■ データセットを作成 単プ	ロジェクトを間定	 無期限 日数: 	
♪が利用可能 →ルを使用して新しいデータセットを作成します			



 作成したデータセットを選択し、データセット名右横の[テーブルを作成]を 選択します。

テーブルの作成



Y¥1	=	4
达	言	尢

プロジェクト名	データセット名		テーブルタイプ 🕜
Tomoo Kubo's Demo Project 🔹 👻	earthquake	*	ネイティブ テーブル 📼
テープル名			
英数字とアンダースコアのみ使用できます			



- 先程ダウンロードしたデータをソースデータとしてアップロードします。
- [空のテーブル]を[アップロード]に変更し、ファイルを指定します。

ソース

テーブルの作成元:		ファイルを選択: 🕜		ファイル形	式:
アップロード	•		参照	Avro	•

1. アップロードを選択 2. ローカルにダウンロードしたファイルを指定 3. CSVを選択



• 宛先テーブル名を入力





テーブルの作成(自動スキーマ定義)

- ここでは、自動でテーブルのスキーマを定義します。
- [自動検出]にチェックを入れ、テーブルを作成をクリックします。





詳細オプション 🗸





参考:テーブルの作成(手動スキーマ定義)

- 定義を手動で定義することも出来ます。
- スキーマの自動検出をチェックせずに、テキストとして編集に、
 以下のような、カラム名とデータ型を記載します。

time:string,latitude:float,longitude:float,depth:float,mag:float,magType:string,n
st:integer,gap:float,dmin:float,rms:float,net:string,id:string,updated:string,pla
ce:string,type:string,horizontalError:string,depthError:string,magError:string,ma
gNst:string,status:string,locationSource:string,magSource:string



参考:テーブルの作成(手動スキーマ定義)

- 自動でスキーマを定義する場合、CSVの1行目にカラム名が入っているこ とがあります。
- オプションの設定にて、特定行を除外することが出来ます。

書き込み設定: 空の場合に書き込む ▼ 許可されているエラー数: ● 不明な値: ● 0 「不明な値を無視する フィールド区切り文字: ● 1 ソース・ファイルの一行日がカラムタになっているので 一行スキ	
○ ^{¬¬¬¬¬¬¬¬¬¬¬¬¬¬¬¬¬¬¬¬¬¬¬¬¬¬¬¬¬¬¬¬¬¬¬¬}	
フィールド区切り文字: 0 1 ソース・ファイルの一行日がカラム名になっているので、一行スキ	
	
	ツノ
スキップするヘッダー行: 🕖 Quoted newlines 🛞 Jagged rows 🚱	
1 引用符で囲まれた改行を許可する ジャグ行を許可する	
路号化 データは自動的に暗号化されます。暗号鍵管理ソリューションを選択してください。	
Google が管理する鍵	
山 お客様が管理する鍵	
Google Cloud Key Management Service (こより管理します	



参考:テーブルの作成(ジョブ履歴の確認)

• [ジョブ履歴]をクリックしてロードジョブの状況を確認する

保存したクエリ	
ジョブ履歴	
転送 🛛	
リソース	■ プロジェクトを固定

アップロードされたファイル を macro-mile-214114 earthquake.eq_201808 に読み込みます

アップロードされたファイル を macro-mile-214114:earthquake.eq_201808 に読み込みます 19:32

macro-mile-214114:asia-northeast1.bquxjob_8074cf5_1656b7e412b
東京 (asia-northeast1)
2018/08/24 19:32:42
2018/08/24 19:32:42
2018/08/24 19:32:44
1.138 秒
macro-mile-214114:earthquake.eq_201808
false
false
CSV
0



テーブルの作成

クエリを投げて、テーブルが作成できたか確認します。





- 結果を視覚化したい場合、Googleのデータポータルサービスを利用する ことが出来ます。
- クエリ結果のデータポータルで調べるを選択します

クエリ結果 📩 結果を保存する 👻	⋒ データポータルで調べる
-------------------	---------------

クエリ完了(経過時間: 1.707 秒、処理されたバイト数: 256.92 KB)

ジョ	ブ情報	結果	JSON	実行の詳細

行	time	latitude	longitude	depth	mag	magType
1	2018-11-14 00:10:33.100 UTC	19.6256	-65.084	85.0	4.9	mb
2	2018-11-06 19:28:11.480 UTC	37.7171667	-122.1216667	5.38	2.53	md
3	2018-10-30 00:12:29.600 UTC	37.3595	-121.7225	6.51	2.5	md
4	2018-11-10 22:21:57.080 UTC	38.8458333	-122.836	2.84	2.47	md



Data Studio Explorerにクエリ結果が連携される ディメンション、指標は自動的に選択されてグラフが生成される

.	無題のエクスプローラ - 2019/03/22 14:52	Explorer の詳細 C	10	G,	0	保存		8
k	 <u>山</u>、グラフを追加 							
= フィル	フィルタを作成する器様やディメンションの項目をここにドロップします 🔇		7	57	〉表			^
9				EI		EF .	Tetal 1.168	Sessions 69.3K
	net	Record Count *			4		1,100	-
1,	us	801		~	myn	~	thh	thit
2	ak	251		111	=			
З.	pr	205	1 4		_	-		
4	ne	43		0	2,99	Niti	1411	300
5,	<u>uu</u>	24		-	-	-	A	
6	ci	23	0	000	22	2	M	÷ .
7.	hv	18				==:		Real Property lies
8.	nn	14	1	1			-	
9.	dm	3	7	-9	ス	タイル		
10.	58	а	-			-		
11.	uw	а	7-9	Dialar		36.8	明瞭破現間	1. Ta 1. 11
12.	nm	2	6	out vier	10.0	0	14	100,0003
				= / * *	teres tre	· · · · · ·	locationS	



ディメンション、指標とは?

ディメンション: データのカテゴリのこと。ディメンションの各値は、名前や説明など、カテゴリの 特性を指す。

指標: 測定値のこと。合計、カウント、比率などの値を示す

https://support.google.com/datastudio/answer/6402048?hl=ja



結果をデータポータルに連携

グラフ種類を[期間]に変えてみる





データソース名横の鉛筆マークをクリックし、[フィールドを追加]を選択





緯度、経度フィールドを作成する(項目名:latlon)

CONCAT(latitude, ",", longitude)

调目名	項目 10	
lation	calc_q7d5qzqjyb	
数式 ②		式の書式を設定
CONCAT(latitude , ",", longitude)		
0		キャンセル 保存



すべてのフィールドをクリックし、latlonのタイプを変更する

• [地域] > [緯度、経度]

		BigQuery	- 2018/08/26 18:44				ートのフィールド論	国コード 「 大陸	
体田可能から日								大陸コード	
使用可能な項目	~	デッ	フィールド		タイプ		集計方法	亜大陸	
	Q	18	place	:				亜大陸コード	
Record Count		19	magNst	:	123 数值	1	,	地域	
		20	net	:	ABC テキ	ネト		地域コード	
		21	updated	:	🖬 日付	けと時刻	•	都市	
		22	longitude	1	x - 7-	-ル値		都市コード	
		23	status	:	⑤ 地切	ŧ	•	大都市圈	
		24	lation	fx ÷	123 通貨	t.	*	大都市圏コード	





[グラフ]を地図に、[ディメンション]をlatlon、[指標]をmagに変更











Soogle Cloud

参考: テーブルを作成する(bqコマンド)

bqコマンド: <u>https://cloud.google.com/bigquery/quickstart-command-line</u>

bqコマンドでテーブルを作成します

- 1. Cloud Storage でバケットを作成(未作成の場合)
- 2. Cloud Storage に 地震データ 4.5_month.csv をアップロード
- 3. Cloud Shell でschema.txt を作成(以前作成したearthquakeと同じ)

time:string,latitude:float,longitude:float,depth:float,mag:float,magType:string,n
st:integer,gap:float,dmin:float,rms:float,net:string,id:string,updated:string,pla
ce:string,type:string,horizontalError:string,depthError:string,magError:string,ma
gNst:string,status:string,locationSource:string,magSource:string



参考: テーブルを作成する(bqコマンド)

4. 以下のコマンドを実行

- \$ GSDIR=gs://my-first-project-244405 # 自分のCloud Storageのバケットを指定
- \$ DATAFILE=\$GSDIR/4.5_month.csv # バケットにCSVデータをアップロードしておく
- \$ BQ TABLE=earthquake.eq 2019 # <データセットID>.<テーブル名> のフォーマット
- \$ SCHEMA=`cat ./schema.txt` # スキーマファイルを作成しておき、展開
- \$ GLOBAL_OPTIONS="--nosync"
- \$ CMD_OPTIONS="--skip_leading_rows=1"

\$ bq \$GLOBAL OPTIONS load \$CMD OPTIOHNS \$BQ TABLE \$DATAFILE \$SCHEMA



参考: パーティションテーブルを作成する

パーティションテーブル:

https://cloud.google.com/bigquery/docs/partitioned-tables

\$ bq mk --time_partitioning_type=DAY earthquake.eq

\$ bq load \$CMD_OPTIONS 'earthquake.eq\$20190601' \$DATAFILE \$SCHEMA



参考: パーティションテーブルを作成する

• 作成されたパーティションを確認する

\$ bq query 'SELECT partition_id FROM [earthquake.eq\$__PARTITIONS_SUMMARY__]'

\$20190602、\$20190603も同様にデータをロードして、
 以下SQLの表示結果を確認する

\$ bq query 'SELECT partition_id FROM [earthquake.eq\$__PARTITIONS_SUMMARY__]'

```
$ bq query 'SELECT count(*) FROM [earthquake.eq] where _PARTITION_LOAD_TIME
between timestamp('20190601') and timestamp('20190603')'
$ bq query 'SELECT count(*) FROM [earthquake.eq$20190601]'
```



データポータルにアクセスする https://datastudio.google.com/

J Google データスタジオ ベータ版 ホーム

.... 1





- 1. レポートのタイトルを変更する
- 2. [新しいデータソースを作成]をクリック





- 1. [BigQuery]を選択
- 2. (初めての場合のみ)承認ボタンをクリックしてアクセス権を許可

Q 検索							
Google コネクタ (18) データボータルで作成されサポートされているコ:	ネクタです。	詳細					
BigQuery 開発者: Google BigQuery テーブルとカスタムクエリに接続します。 詳細	:	Google	Cloud Spanner 開発者: Google Cloud Spanner データベースに接続します。	1	Google す。詳	Cloud SQL for MySQL 開発者: Google Cloud SQL for MySQL データペースに接続	: 続しま
Google アナリティクス 開発者: Google Google アナリティクスのレポートビューに接続しま す。詳細	:	Google	Google スプレッドシート 開発者: Google スプレッドシートに接続します。 詳細	:	く Google す。詳	Google 広告 開発者: Google 広告の掲載結果レポートのデータに接続 個	いま



データポータルでレポートを作成する

データコネクションを作成する。 [一般公開データセット] > [プロジェクト名] > [sample] > [natality]





データポータルでレポートを作成する

1. データコネクション名を任意の名前に変更

2. [レポートに追加]をクリック

atality								オーナーの認証情報を使用	中 キャンセル レポートに追加
← コネク	ションを編集								0
ィン デッ	フィールド	•	タイプ			集計方法		說明	Q、 フィールドを検索
1	weight_pounds	1	123	数値	Ŧ	승하	*	Weight of the child, in pounds.	
2	apgar_1min	:	123	数値	v	合計	*	Apgar scores measure the health of a newborn of	hild on a scale from 0-10. Value after 1 min.
3	weight_gain_pounds		123	数値	Ŧ	合計	Ŧ	Number of pounds gained by the mother during	pregnancy.
4	year			年 (YYYY)	*	なし		Four-digit year of the birth. Example: 1975.	
5	apgar_5min):	123	数值	+	合計	-	Apgar scores measure the health of a newborn of	hild on a scale from 0-10. Value after 5 min.
6	mother_residence_state		ABC	テキスト	*	なし		The two-letter postal code of the mother's state	of residence when the child was born.
7	mother_married		×I×	ブール値		なし		True if the mother was married when she gave b	rth.
8	mother_age		123	数值	v	合計		Reported age of the mother when giving birth.	
9	Imp	1	ABC	テキスト	-	なし		Date of the last menstrual period in the format N	MDDYYYY. Unknown values are recorded a

C Google Cloud

- 1. [挿入ボタン]をクリック
- 2. [棒グラフ]をクリック

米国出生情報		
ファイル 編集 表示	挿入 ページ 配置 リソース ヘルプ	
	✓ 期間	• • •
	山 棒グラフ	
	● 円グラフ	
	畫 表	
	⑤ 地図	
	2.1 スコアカード	
	● 散布図	



データポータルでレポートを作成する

ディメンション、指標は自動的に選択されてグラフが生成される



ディメンション、指標とは?

ディメンション: **データのカテゴリのこと**。ディメンションの各値は、名前や説明など、カテゴリの 特性を指す。 <mark>指標</mark>:

測定値のこと。合計、カウント、比率などの値を示す

https://support.google.com/datastudio/answer/6402048?hl=ja



データポータルでレポートを作成する

1. [ディメンション]を[year]に変更(使用可能な項目からドラッグする)





- 1. 指標の[weight_pounds]をクリック
- 2. フィールドを作成







カスタムフィールドを作成する

- 1. 項目名を入力:birth_count
- 2. 数式を入力:COUNT(is_male)
- 3. [適用]ボタンをクリック

π 新規フィールド	
名前	
鼓式	
1	
タイプ	
数值	•
表示形式	
なし	•
関数を適用	
なし	• ⑦









グラフのスタイルを変更する

- 1. グラフを選択 > [スタイル]タブ
 - a. 棒の数を10に設定
 - b. [データラベルを表示]にチェック

┃┃┃┃ グラフ > 棒	~
データスタイル	
棒グラフ	
<u>Lu.</u>	E
棒の数 10 データラベルを表示	


データポータルでレポートを作成する

グラフのソート順を変更する

- 1. グラフを選択 > [データ]タブ
 - a. 並べ替え: [year] を選択
 - b. 並べ替え:[昇順]を選択





参考:データポータル活用例 GCP 課金

GCP の課金状況をデータポータルを使って可視化 プロジェクトごと、プロダクトごとで整理済み [How to Copy]から自分の環境にも適用可能





Google SpreadsheetからBigQueryを呼び出し、結果を表やグラフに描画できるアプリケーションです

	Α	В	entertainm		
			notwithsta		
1	entertainment	20	understand		
2	notwithstanding	17	contemplat		
3	understanding	11	commenda		
4	contemplation	11	circumstan		
5	commendations	10	Northumbe		
6	circumstances	9	transgressi		
7	Northumberland	8	consideration		
8	transgression	7	transforma		
9	consideration	7	4	8 12	
10	transformation	6		Count	fil.
		857.97		word_c	oun

参考: Tutorial: Update Spreadsheets with data from Google BigQuery

16

20

🙆 Google Cloud

- 新しいスプレッドシートを作成します
 - drive.google.com にアクセス
 - [新規] > [Googleスプレッドシート]
 - ドキュメント名を変更(任意)

	ドライブ	Q 1	ドライ
		マイド	ライ・
	フォルダ		-
Ð	ファイルのアップロード		
10	フォルダのアップロード		y Fol
8	Google ドキュメント	>	
	Google スプレッドシート	>	
	Google スライド	>	1
	その他	>	
0	保存容量		
	• 32 GB 中 831.6 MB を使用		
	<u> </u>		



- スクリプトエディタを起動します
 - [ツール]>[スクリプトエディタ]

Ħ

BQ Apps Script Demo ファイル 編集 表示 挿入 表示形式 データ ツール アドオン ヘルプ 変更内容をすべてドライブに保存しました

5	4 6 7	100% - ¥	% .0 .00 123▼		フォームを作成	-	è .	H 33	-	≣.	Ŧ
fx	A	В	C	0	スクリプト エディタ		 G		1	н	
1				0	マクロ ト						
2 3 4					スペルチェック 🕨						
5 6 7				~	オートコンプリートを有効にする						
8 9					通知ルール シートを保護						
10											
12				Ť	ユーザー補助設定						
13 14				-	アクティビティ ダッシュボード						
15						r,					



- Spreadsheetから利用できるリソースを設定します
 - [リソース] > [Googleの拡張サービス...]

23-17V M	* 12/			
		en () ►	現在のプロジェクトのトリガー	
ם ⊐ א.gs -		コード.gs ×	すべてのトリガー	
	1 2 3 4	function my	yF ライプラリ Developers Console プロジェクト Google の拡張サービス	



プロジェクト名を変更します

		>
プロジェ	クト名を変更	

新しいノロ	Dェクト名を入力してくたさい BQ Apps	Script Demo
OK	キャンセル	



- BigQuery APIを有効にします
 - [BigQuery API] > [無効ボタン]をクリックして有効にする

AdSense Management API	v1.4 😒	AdSense	無効
Admin Directory API	directory_v1 📀	AdminDirectory	無効
Admin Reports API	reports_v1 📀	AdminReports	無効
Analytics Reporting API	v4 😂	AnalyticsReporting	無効
BigQuery API	v2 😂	BigQuery	ON
Calendar API	v3 😒	Calendar	無効
Content API for Shopping	v2 📀	ShoppingContent	無効
これらのサービスを Google Clou	id Platform API ダッショ	1.ボード でも有効にする。	必要があります。



- デベロッパーコンソールでBigQueryサービスを有効にする
 - Cloud ConsoleからAPIライブラリページへ移動 <u>https://console.cloud.google.com/apis/library</u>
 - Developerコンソールが開くので、BigQuery APIを有効にする





- GCPコンソールでプロジェクト番号を確認します
 - [IAMと管理] > [設定]
 - スクリプトの中で使用するので、番号を控えておく

設定	▶ 移動	盲 シャットダウン
プロジェク	/ト名	- 4
プロジェク	ト ID	



• 以下のコードをスクリプトエディタにコピー&ペーストする

(1/3)

// 先ほど控えたプロジェクト番号に置き換える

```
var projectNumber = 'XXXXXXXX';
```

```
var ss = SpreadsheetApp.getActiveSpreadsheet();
```

```
function onOpen() {
  var menuEntries = [ {name: 'Run Query', functionName: 'runQuery'} ];
  ss.addMenu('BigQuery', menuEntries);
}
```

(次ページに続く)



```
function runQuery() {
                                                                                             (2/3)
 var sheet = SpreadsheetApp.getActiveSheet();
 var sql = 'select TOP(word, 10), COUNT(*) as word count from publicdata:samples.shakespeare WHERE
LENGTH (word) > 12;';
 var queryResults;
  // Inserts a Query Job with an optional timeoutMs parameter.
  try {
   var resource = {query: sql, timeoutMs: 1000, userQueryCache: false};
    queryResults = BigQuery.Jobs.query(resource, projectNumber);
  catch (err) {
   Logger.log(err);
    Browser.msqBox(err);
   return;
```



```
while (queryResults.getJobComplete() == false) {
    try {
      queryResults = BigOuery.Jobs.getOueryResults(projectNumber,
queryResults.getJobReference().getJobId());
      // If the Job is still not complete, sleep script for
      // 3 seconds before checking result status again
      if (queryResults.getJobComplete() == false) {
        Utilities.sleep(3000);
    catch (err) {
      Logger.log(err);
      Browser.msqBox(err);
      return;
// Update the amount of results
 var resultCount = queryResults.getTotalRows();
 var resultSchema = queryResults.getSchema();
```

🙆 Google Cloud

(3/3)

- スクリプトを保存する
 - エディタ上で保存ボタンかCtrl+S
- BigQueryを実行する
 - スプレッドシートに戻ってリロードする
 - BigQueryメニューが表示されるのでクリック

ファイル 編集	表示 挿入	表示形式	データ	ツール	アドオン	ヘルプ	BigQuery	1
5 ~ ~ 5	¥ % .0	.00 123 -	Arial	*	10 -	BI	÷ <u>A</u> -	*
A	В	С		D		E	F	



- OAuth認証を行う
- この段階ではエラーが出なければ成功です。







以下のコードをrunQuery関数の最後に追加する

```
var resultValues = new Array(resultCount);
var tableRows = queryResults.getRows();
```

```
// Iterate through query results
for (var i = 0; i < tableRows.length; i++) {
  var cols = tableRows[i].getF();
  resultValues[i] = new Array(cols.length);
  // For each column, add values to the result array
  for (var j = 0; j < cols.length; j++) {
    resultValues[i][j] = cols[j].getV();
  }
}</pre>
```

// Update the Spreadsheet with data from the resultValues array
sheet.getRange(1, 1, resultCount, tableRows[0].getF().length).setValues(resultValues);



- 追加したコードの動きを実行して確認する
 - [BigQuery] > [Run Query]

	ファイル 編集	表示 挿入	表示形式	データ	ツール	アドオン	ヘルプ	BigQuery	最終編集
	ēr a ī	b ¥ % .0₊	.0 <u>0</u> , 123 -	Arial	7	10 -	BI	5 <u>A</u> .	♦₀ - ⊞
fx									
	A	В	С		D		Е	F	
1	entertainment	20							
2	notwithstanding	17							
3	understanding	11							
4	contemplation	11							
5	commendations	10							
6	circumstances	9							
7	Northumberland	8							
8	transgression	7							
9	consideration	7							
10	transformation	6							
11									
12									



以下の関数(buildDataTable)を最後に追加する

// Returns a Data Table from a query's result values and schema
function buildDataTable(resultValues, resultSchema) {

```
dataTable.addColumn(columnType, columnName);
```

```
for (var j = 0; j < resultValues.length; j++) {
  dataTable.addRow(resultValues[j]);</pre>
```

```
dataTable.build();
return dataTable;
```

buildDataTable関数を利用するコードをrunQuery関数の最後に追加

```
var bigqueryDataTable = buildDataTable(resultValues, resultSchema);
  var chart = Charts.newBarChart()
    .setDataTable(bigqueryDataTable)
    .setColors(['green'])
    .setDimensions(500, 500)
    .setYAxisTitle('Word')
    .setXAxisTitle('Count')
    .setLegendPosition(Charts.Position.BOTTOM)
    .build();
  var chart panel = UiApp.createApplication()
    .setTitle('BigQuery Results')
    .setWidth('600')
    .setHeight('600');
  chart panel.add(chart)
```





実行する

fx	entertainment			
	A	В	BinOuery Results	~
1	entertainment	20	Dig study hoodito	
2	notwithstanding	17		
3	understanding	11		
4	contemplation	11		
5	commendations	10		
6	circumstances	9		
7	Northumberland	8	entertainm	
8	transgression	7	not vilhete	
9	consideration	7	notwithsta	
10	transformation	6	understand	
11			antomalat.	
12			contemplat	
13			g commenda	
14			o circumstan	
15				
16			Northumbe	
17			transgrassi	
18			uansyresen	
19			consideration	
20			transforma	
21				
22			4 8 12 16 20	
			2	



BigQuery SQLの特徴

- SQL標準に類似した構文
 - 行単位の更新、削除は未サポート
 - テーブル毎削除するか、新しいテーブルに書き出すことで対応
- UNION
 - FROMの後にテーブル名をカンマで区切ると UNION することができる
- JOIN
 - 右側のテーブルが圧縮後 8MBを超える場合は、Big Join (JOIN EACH構文)を利用する
- GROUP BY
 - 多数のグループを生成するクエリは失敗する可能性がある (Resources exceeded during query execution)。その場合はGROUP EACH BYを使用する



BigQuery SQLの特徴

- COUNT DISTINCTは統計的近似値を返す
 - COUNT([DISTINCT field [,n])
 - nを大きくすることで厳密性を向上させることができる
 - nのデフォルト値は1000
 - 性能と正確さのトレードオフとなる
 - EXACT_COUNT_DISTINCT() 関数を使う
- アウトプットサイズの制限
 - 圧縮後128MB。これを超える場合は、テーブルに出力することで回 避可能







BigQuery 参考URL

- オンラインマニュアル
 - <u>https://cloud.google.com/bigquery/what-is-bigquery</u>
- クエリーリファレンス
 - <u>https://cloud.google.com/bigquery/query-reference?hl=ja</u>
- BigQuery Client Library
 - <u>https://cloud.google.com/bigquery/client-libraries</u>
- サポート
 - <u>StackOverflow</u>
 - 障害時通知のメールグループ (<u>bigquery-downtime-notify</u>)
- 関連ビデオ集
 - <u>http://www.youtube.com/results?search_query=BigQuery&sm=3</u>





はじめてみよう GCP ハンズオンセミナー Cloud Dataflow 編 (Python SDK 編)





- Google Cloud Platformプロジェクトを作成していること
 - 作成してない場合は<u>こちら</u>
- 課金が有効になっていること
 - 有効になってない場合は<u>こちら</u>
- "Google Compute Engine, CloudDataflow"サービスが有効になっていること
 - 有効になっていない場合は<u>こちら</u>



インストールする

Cloud Shell を起動して以下を実行

=	Google Cloud Platform 💲 fukudak-playground 👻	<u>.</u>	
,	fuluidale alauneaund authan una	D	Activate Google Cloud Shell

\$ pip install google-cloud-dataflow --user



Hello World

Python をインタラクティブモードで起動

\$ python



Hello World

以下を実行する

```
# 必要モジュールをインポート
import apache_beam as beam
# パイプラインを作成
p = beam.Pipeline()
# PCollection を作成してファイルに書き込む
(p
| beam.Create(['hello', 'world'])
| beam.io.WriteToText('./test'))
# パイプラインを実行する
p.run()
# Python インタラクティブモードを抜ける
exit()
```



Hello World

出力されたファイルを確認する

\$ more ./test*



Dataflow モデルの基本概念

以下の4つの主要概念を理解する

- パイプライン
- PCollection
- 変換 (PTransform)
- I/0ソース およびシンク



Dataflowモデル: パイプライン

- パイプラインはデータ処理の一連の流れを記述したもの。実行時はジョブに相当する。
- パイプラインは入力データのソースを読み取り、データを変換し、結果の出力を書き込む、一連の操作で構成される
- 一つのプログラムで複数のパイプラインを定義することが可能
- パイプラインに定義されるデータと変換は、そのパイプライン固有のものであり、複数のパイプラインで共有することはできない





Dataflow モデル: PCollection

- PCollection はパイプライン内の一連のデータを表すコンテナ
- Dataflow の PCollection クラスは、事実上サイズ制限がないデータセットを表すことができる特殊コンテナクラス
- Bounded データ / Unbounded データ 両方共保持することができる
 - Bounded データ: テキストファイルやBigQueryテーブルのような境界 が定義できるデータセット
 - Unbounded データ: Pub/Sub に入力されるストリームのように継続して入力されるデータセット



変換 (PTransform)

- 変換は、パイプライン内のデータ処理操作を表す
- 1つまたは複数のPCollection を入力として、そのPCollectionの要素に 対して処理関数を実行し、1つまたは複数のPCollectionを出力する
- 汎用処理はコア変換として提供される
 - ParDo: 汎用的な並列処理
 - GroupByKey: キー / 値ペアをキーに基づいてグループ
 - Combine: コレクションまたはグループ化された値を結
 - Flatten: コレクションをマージ
- 変換はネストすることが可能(複合変換)



I/O ソースおよびシンク

- <u>パイプラインI/O</u>としてデータソースとデータシンクのAPIが提供
- **ソースAPI**を使ってパイプラインにデータを読み込む
- **シンクAPI**を使ってパイプラインからデータを出力する
- パイプラインはソースAPIによりデータ入力から始まって、シンクAPIの出力によって終了する



Wordcount サンプル

Cloud Shell コードエディタでファイルを作成

\$ touch wordcount.py

\$ cloudshell edit wordcount.py

以下のコードを入力


Dataflow モデル





Wordcount サンプル

プログラムを実行

\$ python wordcount.py

出力されたファイルを確認

```
$ cat word_count-00000-of-00001
(u'pomp', 1)
(u'bounty', 2)
(u'th', 1)
(u'to', 440)
...
```



クラウド上で実行する

wordcount.py を少し修正

\$ cloudshell edit wordcount.py

修正するポイント:

• 実行時の出力ファイルをargとして指定できるようにする



クラウド上で実行する

```
import ssl
import re
import argparse
import apache beam as beam
argv = None
parser = argparse.ArgumentParser()
parser.add argument('--output', required=True, help='Output file must be specified')
known args, pipeline args = parser.parse known args(argv)
p = beam.Pipeline (argv=pipeline args)
(p
 'read' >> beam.io.ReadFromText('qs://dataflow-samples/shakespeare/kinglear.txt')
 'split' >> beam.FlatMap(lambda x: re.findall(r'\w+', x))
 'count words' >> beam.combiners.Count.PerElement()
 | 'save' >> beam.io.WriteToTextknown args.output))
p.run()
```



クラウド上で実行する

環境変数のセット

\$ export PROJECT=\$(gcloud config list project --format "value(core.project)")

バケットの作成

- \$ gsutil mb gs://\$PROJECT-\$USER
- # 実行 (エラーを吐く場合、Dataflow APIが有効か確認してください)
- $\$ python wordcount.py --runner DataflowRunner $\$

```
--project $PROJECT \
```

--staging_location gs://\$PROJECT-\$USER/staging $\$

```
--temp_location gs://$PROJECT-$USER/staging \
```

--job_name \$PROJECT-python-wc-\$USER \

--output gs://\$PROJECT-\$USER/output/word_count



コンソールで確認する

[ビッグデータ] > [Dataflow] で確認する

Google Cloud Plat	form 1	Google Cloud Platform In fukudak playmound +		899
A Home		← fukudakhon-wc ≡ Loss	Summary	
Manitoring				
4.0020000020000		X	Job Name	fukudak playground python-wc
Compute Engine	3	read x	Job ID	2017-01-16_03_51_01-795429494614785770
A Maturation	2		STM Version	Dende Cond Detative SDK for Exten 0.4.4
* Networking	2.	· · · · · · · · · · · · · · · · · · ·	Job Type	Batch
Container Engine	Q.		Start Time	Jun 16, 2017, 8:51:02 PM
- container Englise		spie	Elapsed Time	4 min 7 sec
ODUCTS A			Total Worker Time 🔱	5 min 56 sec
E Logging	2	T	Autoscaling	
Error Departing			Workstein	P
, chernepereng		Turning	Carrent State	Worker pool started
OLS		T	Worker History	
] Container Registry	2	S200	3	1
Development	>		2	1
Deployment Manager				
Taetiah			Jan 18	. 8 51 PM Jan 16, 8 52 PM Jan 1
Tuor Low			Current Workers: 3	Targat Workaric II
Endpoints			See More History	
			Resource Metrics	
LD.CH.			Current «CPU»	D
T Training			Total vCPU Time	360 vCPU sec
BigQuery			Current Memory	DB
			Total Memory Time 🌐	1,362.408 MB sec
Pub/Sub			Current PD 🛞	0.8
			Total P0 Time 🐵	90,000 GB sec
Detaproc			Current 880 PD	0.0
			Total \$50.00 Time (1)	D DB dat



生成されたファイルを確認する

CloudShellより以下を確認

\$ gsutil ls gs://\$PROJECT-\$USER/output gs://fukudak-playground-fukudak/output/word_count-00000-of-00009 gs://fukudak-playground-fukudak/output/word_count-00002-of-00009 gs://fukudak-playground-fukudak/output/word_count-00003-of-00009 gs://fukudak-playground-fukudak/output/word_count-00004-of-00009 gs://fukudak-playground-fukudak/output/word_count-00005-of-00009 gs://fukudak-playground-fukudak/output/word_count-00005-of-00009 gs://fukudak-playground-fukudak/output/word_count-00005-of-00009 gs://fukudak-playground-fukudak/output/word_count-00006-of-00009 gs://fukudak-playground-fukudak/output/word_count-00007-of-00009 gs://fukudak-playground-fukudak/output/word_count-00007-of-00009





- Apache Beam Python SDK
 - <u>https://github.com/apache/beam/tree/python-sdk/sdks/python</u>
- Stackoverflow Google Cloud Dataflow
 - <u>http://stackoverflow.com/questions/tagged/google-cloud-data</u>
 <u>flow</u>



Thank you

