

Kubernetesを 使うためのコマンド

宮島健太

Kubernetesを 使うステップ

1. コンテナ、スラクタを作成する
2. アプリケーションをデプロイする
3. スケールアウト（スケールアップ）
4. アプリケーションのバージョンアップ
5. クリーンアップ

※GCPのゾーンの設定等は省いてます。

はじめに

Kubernetesを使うためのコマンドを
インストールします。

kubectl

1.0

アプリケーションの コンテナイメージを作成

`docker build -t タグ名 ディレクトリパス`

Dockerfileに記載してあるコンテナイメージを作成し、
タグをつける。

Docker buildコマンドは3、4回目にやったので、そちらもご覧ください。

1.0 例

```
$docker build -t gcr.io/${PROJECT_ID}/hello-app:v1 .
```



コンテナ : gcr.io/\${プロジェクト識別子}/アプリケーション名 : バージョン

GCPのリポジトリを指定

Dockerイメージとしてのタグ



REPOSITORY	TAG	IMAGE ID	CREATED	SIZE
gcr.io/my-project/hello-app	v1	25cfadb1bf28	10 seconds ago	54 MB

1.1

GKEでコンテナイメージを 使う準備

GKE…Google Kubernetes Engine

GKEで、作成したコンテナイメージを使うため、
コンテナレジストリにアップロードする。

→Dockerイメージのデータベース

```
$ docker push gcr.io/${PROJECT_ID}/hello-app:v1
```


1.2

クラスタを作成する

```
$gcloud container clusters create クラスタ名  
--num-nodes=ノード数
```

クラスタの確認

```
$gcloud compute instances list
```


1.2 例

```
$gcloud container clusters create hello-cluster  
--num-nodes=2
```

hello-clusterを2ノード分作成する

NAME	ZONE	MACHINE_TYPE	PREEMPTIBLE	INTERNAL_IP
gke-hello-cluster-default-pool-07a63240-822n	us-central1-b	n1-standard-1		10.128.0.7
gke-hello-cluster-default-pool-07a63240-kbtq	us-central1-b	n1-standard-1		10.128.0.4

1.2

既存のクラスタを使用する場合

```
$gcloud container clusters get-credentials
```

クラスタ名

2.1

podを作成する

まずはpodを作成する

```
$ kubectl create deployment 任意のアプリケーション名  
--image=Dockerイメージ名
```



```
$ kubectl create deployment hello-web  
--image=gcr.io/${PROJECT_ID}/hello-app:v1
```


2.2

podを確認する

\$kubectl get pods

NAME	READY	STATUS	RESTARTS	AGE
<u>hello-web-4017757401-px7tx</u>	1/1	Running	0	3s

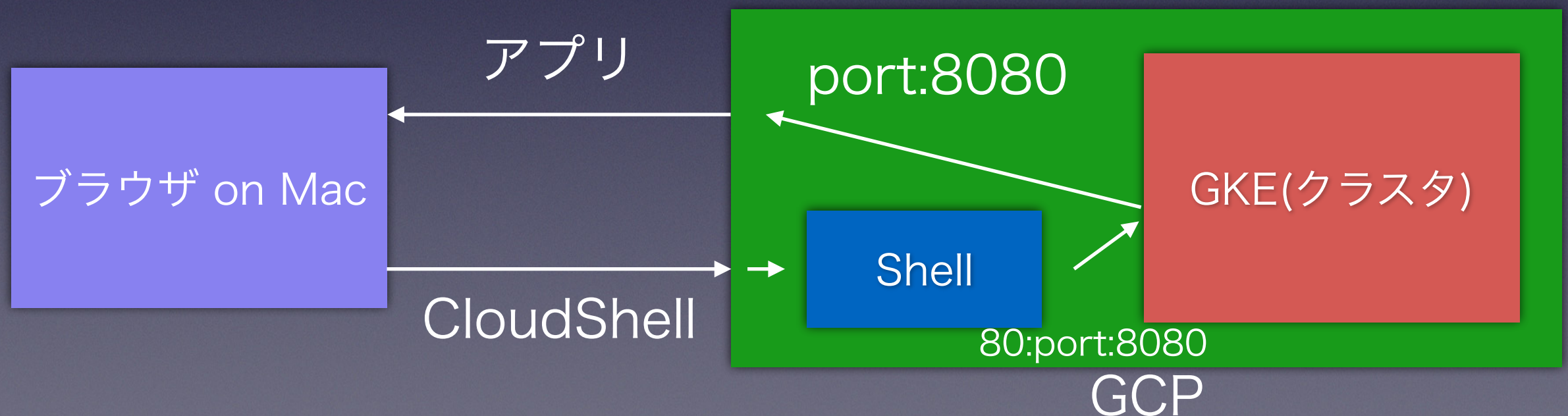


任意のアプリケーション名になっている
→Kubernetes用の識別名(pod名)

2.3

アプリケーションを インターネットに公開する

`$kubectl expose deployment pod名`
`--type=公開方法 --port CloudShellのポート`
`--target-port クラスターのポート`



2.3 補足説明

type (公開する方法)

- **ClusterIP (既定値)** … クラスター内の内部IPでServiceを公開。
- **NodePort** … NATを使用して、クラスター内の選択された各ノードの同じポートにServiceを公開。
- **LoadBalancer** … 現在のクラウドに外部ロードバランサを作成、固定の外部IPを割り当てる。NodePortのスーパーセット。
- **ExternalName** … 仕様のExternalNameで指定した名前のCNAMEレコードを返すことによって、任意の名前を使ってServiceを公開。

2.3 例

```
$ kubectl expose deployment hello-web  
--type=LoadBalancer --port 80 --target-port 8080
```

IPアドレスを確認

```
$ kubectl get service
```

NAME	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
hello-web	10.3.251.122	203.0.113.0	80:30877/TCP	3d

3.1 スケールアップ

```
$kubectl scale deployment ポッド名  
--replicas=サービス数(レプリカ)
```

確認

```
$kubectl get deployment ポッド名
```


3.1 例

```
$kubectl scale deployment hello-web  
—replicas=3
```

確認

```
$kubectl get deployment hello-web
```

NAME	DESIRED	CURRENT	UP-TO-DATE	AVAILABLE	AGE
hello-web	3	3	3	2	1m

4.1

バージョンアップ

```
$ docker build
```

```
-t gcr.io/${PROJECT_ID}/hello-app:v2 .
```



ここだけ変える

(レジストリにイメージをプッシュしましょう)

```
$ docker push gcr.io/${PROJECT_ID}/hello-app:v2
```


4.2

ローリングアップデート

```
$ kubectl set image deployment/hello-web hello-  
app=gcr.io/${PROJECT_ID}/hello-app:v2
```

実行中のクラスタ（前バージョン）に
上書きの形で実行させる

プログラムの実行を継続させたままアップデート

5.1

アプリケーションの削除

サービスを削除する

`$kubectl delete service ポッド名`

実行中のクラスターを削除する

`$ gcloud container clusters delete クラスター名`

5.1 例

```
$ kubectl delete service hello-web
```

```
$ gcloud container clusters delete hello-cluster
```