



인터페이스설계서

쿠버네티스 기반을 사용한 마이크로서비스 구축

[Version 2.1.5]

인터페이스 정의서

API명 : 스프링부트와 쿠버네티스 기반의 마이크로서비스 구축

설명 : 스프링부트를 사용해서 쿠버네티스를 구축하고 배포하여

마이크로서비스가 구축되도록 하는 것!

목적 : 스프링부트를 사용해서 GIS개발이 되도록 쿠버네티스를 구축하고

배포하여 마이크로서비스가 구축되게끔 함!

사용 API : 쿠버네티스(google cloudplatform API), GIS(google map API

(구글 API), Kokao map API(카카오Developer API))

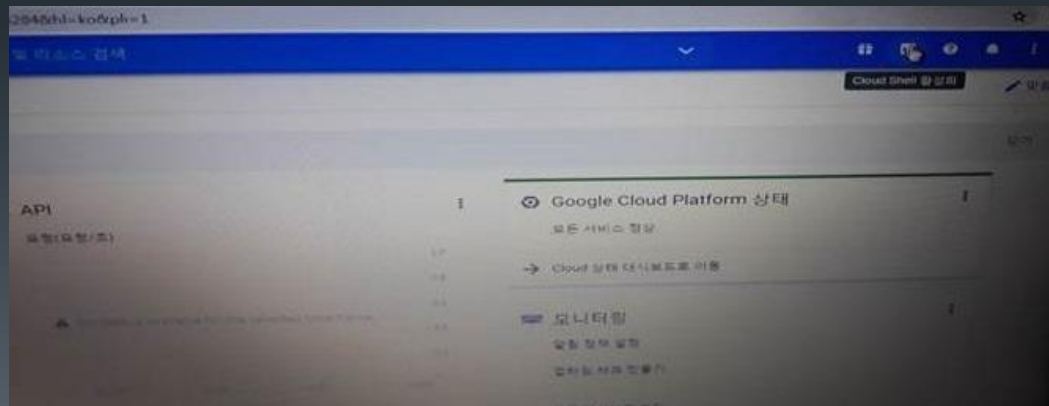
인터페이스 정의서

API명 : 쿠버네티스(google cloudplatform API)

사용하는 방법 : **Google Cloud Shell** 활성화하기

- Google Cloud Shell은 다양한 개발 도구가 탑재된 가상 머신
- 5GB의 영구 홈 디렉토리를 제공하며 Google Cloud에서 실행
- Google Cloud Shell을 사용하면 명령줄을 통해 GCP 리소스에 액세스 가능

1. GCP Console의 오른쪽 상단 툴바에서 **Cloud Shell 열기** 버튼을 클릭

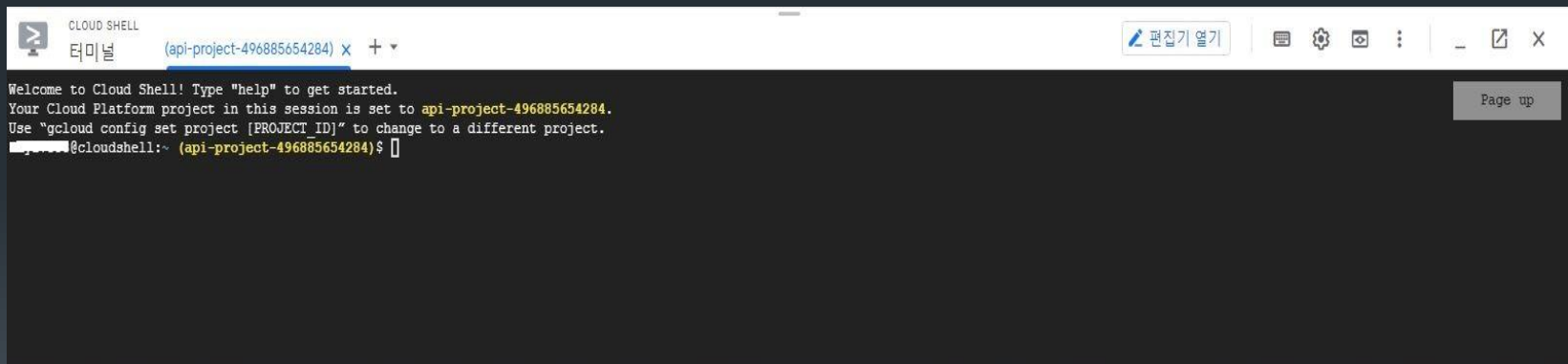


인터페이스 정의서

API명 : 쿠버네티스(google cloudplatform API)

사용하는 방법 : 2. 계속(Continue) 클릭

- 연결되면 인증된 상태로 **PROJECT_ID** 가 설정됨



The screenshot shows a Cloud Shell terminal window. The title bar includes the Cloud Shell logo, the text 'CLOUD SHELL', and a tab labeled '터미널 (api-project-496885654284)'. The terminal content displays a welcome message: 'Welcome to Cloud Shell! Type "help" to get started. Your Cloud Platform project in this session is set to api-project-496885654284. Use "gcloud config set project [PROJECT_ID]" to change to a different project.' The prompt is '@cloudshell:~ (api-project-496885654284)\$'. The terminal interface includes a '편집기 열기' (Open Editor) button in the top right and a 'Page up' button on the right side of the terminal area.

```
WELCOME TO CLOUD SHELL! Type "help" to get started.
Your Cloud Platform project in this session is set to api-project-496885654284.
Use "gcloud config set project [PROJECT_ID]" to change to a different project.
@cloudshell:~ (api-project-496885654284)$
```

- Google Cloud Platform의 명령줄 도구 gcloud는 Cloud Shell에 사전 설치되어 있으며 탭 자동 완성을 지원

인터페이스 정의서

API명 : 쿠버네티스(google cloudplatform API)

사용하는 방법 : 사용 중인 계정 이름 목록 표시
gcloud auth list

출력: Credentialed accounts: - <myaccount>@<mydomain>.com (active)

출력 예: Credentialed accounts: - abcdefg@abcdefg.net

프로젝트 ID 목록을 표시

gcloud config list project

출력: [core] project = <project_ID>

출력 예: [core] project = qwiklabs-gcp-12345a67dea890a1

gcloud 관련 전체 설명서는 [Google Cloud gcloud 개요](#)에서 확인 가능

인터페이스 정의서

API명 : 쿠버네티스(google cloudplatform API)

사용하는 방법 : 기본 컴퓨팅 영역 설정

- 컴퓨팅 영역 : 클러스터와 리소스가 존재하는 대략적인 지역적 위치. (예: us-central1-a는 us-central1 리전의 영역)

기본 컴퓨팅 영역을 us-central1-a로 설정

gcloud config set compute/zone us-central1-a출력
Updated property [compute/zone].

Kubernetes Engine 클러스터 만들기

- 클러스터는 1개 이상의 클러스터 마스터 머신과 노드라는 다수의 작업자 머신으로 구성
- 노드란 클러스터를 구성하기 위해 필요한 Kubernetes 프로세스를 실행하는 Compute Engine VM(가상 머신) 인스턴스

클러스터 생성

※ [CLUSTER-NAME]을 내가 선택한 클러스터 이름으로 대체 (예: my-cluster)

※ 클러스터 이름은 문자로 시작하고 영숫자로 끝나야 하며 40자를 초과할 수 없음

gcloud container clusters create [CLUSTER-NAME]

인터페이스 정의서

API명 : 쿠버네티스(google cloudplatform API)

사용하는 방법 : 클러스터의 사용자 인증 정보 얻기

- 클러스터를 만든 후 클러스터와 interaction 하려면 사용자 인증 정보를 얻어야 함

클러스터 인증

※ [CLUSTER-NAME]을 내 클러스터 이름으로 대체

gcloud container clusters get-credentials [CLUSTER-NAME]출력 :

Fetching cluster endpoint and auth data. kubeconfig entry generated for my-cluster.

클러스터에 애플리케이션 배포하기

- 클러스터 생성 후, [컨테이너식 애플리케이션](#) 배포 가능.

- Kubernetes Engine에서는 Kubernetes 객체를 사용해 클러스터의 리소스를 생성하고 관리

- 웹 서버와 같은 상태 비추적 애플리케이션을 배포할 때는 Kubernetes에서 [배포](#) 객체를 사

- [서비스](#) 객체는 인터넷에서 애플리케이션에 액세스하기 위한 규칙과 부하 분산을 정의

Cloud Shell에서 다음 [kubectl create](#) 명령어를 실행하여 hello-app 컨테이너 이미지에서 새 포 hello-server를 생성

kubectl create deployment hello-server --image=gcr.io/google-samples/hello-app:1.0

인터페이스 정의서

API명 : 쿠버네티스(google cloudplatform API)

사용하는 방법 : - --image는 배포할 컨테이너 이미지를 지정, 해당 명령어는 [Google Container Registry](#) 버킷에서 예시 이미지를 가져옴

- gcr.io/google-samples/hello-app:1.0은 가져올 특정 이미지 버전을 나타냄. 버전이 지정되지 않은 경우 최신 버전 사용

[kubectl expose](#) 명령어를 실행하여 Kubernetes 서비스를 생성

- 이 서비스는 애플리케이션을 외부 트래픽에 노출할 수 있게 해주는 Kubernetes 리소스임

`kubectl expose deployment hello-server --type=LoadBalancer --port 8001`

--port를 통해 컨테이너가 노출될 포트 지정

type="LoadBalancer" : 컨테이너의 Compute Engine 부하 분산기 생성

출력 : service/hello-server exposed

외부 IP 주소와 노출된 포트를 사용해 웹브라우저에서 애플리케이션을 확인

`http://[EXTERNAL-IP]:8001`

클러스터 삭제

인터페이스 정의서

API명 : GIS 화면(Google map API(구글 API))

사용하는 방법 : <script async

defer src="<https://maps.googleapis.com/maps/api/js?key=자신의API키를넣으세요&callback=initMap>">

</script>

- 구글 API Console에서 받은 자신의 API키를 넣으면 구글맵 사용가능

*사용 파라미터 정보

필수 여부	파라미터	파라미터 설명	Length	비고
o	key	Api 인증키	39	관리자한테 인증받은 구글 api키
x	프로젝트이름	Project 이름	random	관리자한테 받은 프로젝트 이름
x	프로젝트 ID	Project ID	random	관리자한테 받은 프로젝트 ID
x	프로젝트번호	Project 번호	random	관리자한테 받은 프로젝트 번호

인터페이스 정의서(샘플코드)

API명 : GIS 화면(Google map API(구글 API))

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Google Map</title>
  </head>
  <body>
    <div id="map" style="width:100%; height: 100vh;"></div>
    <script async defer
src="https://maps.googleapis.com/maps/api/js?key=YOUR_KEY&callback=initMap&r
egion=kr"></script>
    <script> function initMap() { var seoul = { lat: 37.5642135 ,lng: 127.0016985 }; var
map = new google.maps.Map( document.getElementById('map'), {
  zoom: 12, center: seoul });
} </script>
  </body>
</html>
```

인터페이스 정의서

API명 : GIS 화면(Kakao map API(카카오Developer API))

사용하는 방법 : <script type="text/javascript"

src="//dapi.kakao.com/v2/maps/sdk.js?appkey=발급받은 APP KEY를 넣으시면 됩니다."></script>

- 카카오Developer API에서 받은 api키를 넣으면 됩니다. 여기서 사용하는 카카오맵 api는 플랫폼에 들어가서 javascript키를 사용하시면 됩니다.

*사용 파라미터 정보

필수 여부	파라미터	파라미터 설명	Length	비고
o	key	Api 인증키	32	Kakao에서 받은 javascript 키
x	key	Restapi 키	32	Kakao에서 받은 restapi 키
x	key	네이티브앱키	34	Kakao에서 받은 네이티브앱 키
x	key	Admin 키	33	Kakao에서 받은 admin 키

인터페이스 정의서(샘플코드)

API명 : (Kakao map API(카카오Developer API))

```
<html>
<head>
<meta charset="utf-8">
<title>지도 생성하기</title>
</head>
<body>
<!-- 지도를 표시할 div 입니다 -->
<div id="map" style="width:100%;height:350px;"></div>
<script type="text/javascript" src="//dapi.kakao.com/v2/maps/sdk.js?appkey=발급받은
APP KEY를 사용하세요"></script>
<script> var mapContainer = document.getElementById('map'), // 지도를 표시할 div
mapOption = { center: new kakao.maps.LatLng(33.450701, 126.570667),
var map = new kakao.maps.Map(mapContainer, mapOption);
</script>
</body>
</html>
```