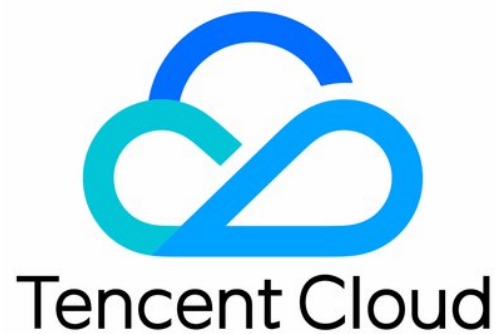


Tencent Kubernetes Engine

시작하기

제품 문서



Copyright Notice

©2013-2023 Tencent Cloud. All rights reserved.

Copyright in this document is exclusively owned by Tencent Cloud. You must not reproduce, modify, copy or distribute in any way, in whole or in part, the contents of this document without Tencent Cloud's the prior written consent.

Trademark Notice



All trademarks associated with Tencent Cloud and its services are owned by Tencent Cloud Computing (Beijing) Company Limited and its affiliated companies. Trademarks of third parties referred to in this document are owned by their respective proprietors.

Service Statement

This document is intended to provide users with general information about Tencent Cloud's products and services only and does not form part of Tencent Cloud's terms and conditions. Tencent Cloud's products or services are subject to change. Specific products and services and the standards applicable to them are exclusively provided for in Tencent Cloud's applicable terms and conditions.

목록:

시작하기

신규 사용자 가이드

표준 클러스터를 빠르게 생성

Demo

간단한 Nginx 서비스 생성

Hello World 서비스 수동 구축

단일 포드 WordPress

TencentDB를 사용하는 WordPress 서비스

간단한 Web 애플리케이션 구축

시작하기

신규 사용자 가이드

최종 업데이트 날짜: : 2023-04-26 18:43:22

본 문서는 지침에 따라 Tencent Kubernetes Engine(TKE)을 빠르게 이해하고 시작하는 데 도움이 됩니다.

1. TKE란 무엇입니까?

기본 Kubernetes 시스템을 기반으로 하는 TKE(Tencent Kubernetes Engine)는 컨테이너 중심의 확장성이 뛰어난 고성능 컨테이너 관리 서비스를 제공합니다. Tencent Cloud IaaS 제품과 긴밀하게 작동하여 비즈니스 컨테이너화를 신속하게 구현할 수 있도록 지원합니다. 자세한 내용은 [제품 개요](#)를 참고하십시오.

TKE는 [TKE 콘솔](#)을 통한 클러스터 및 서비스 작업을 지원합니다.

2. TKE 결제

TKE를 사용하면 과금 항목과 과금 기준이 다른 다양한 유형의 Kubernetes 클러스터를 생성할 수 있습니다. 과금 방식 및 가격에 대한 자세한 내용은 [구매 가이드](#)를 참고하십시오.

3. TKE 사용

3.1 가입 및 인증

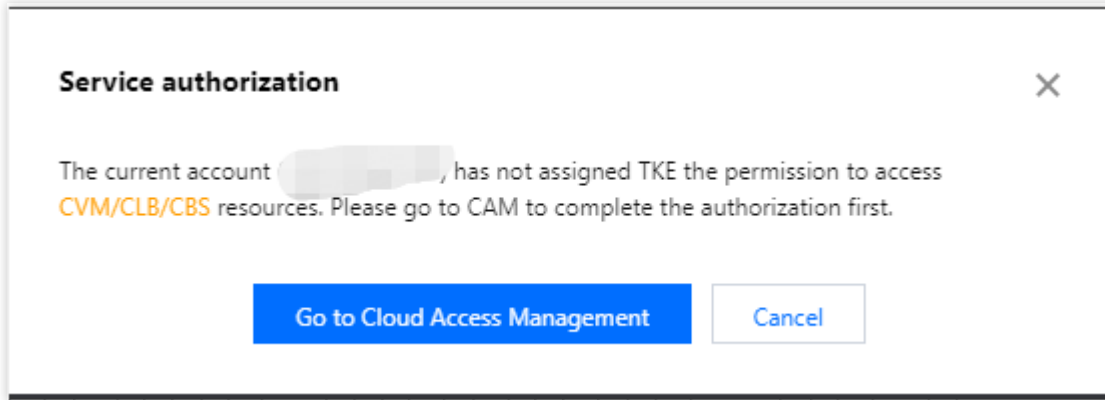
TKE를 사용하기 전에 [Tencent Cloud 계정](#)을 등록하고 [실명 인증](#)을 완료해야 합니다.

3.2 역할 인증

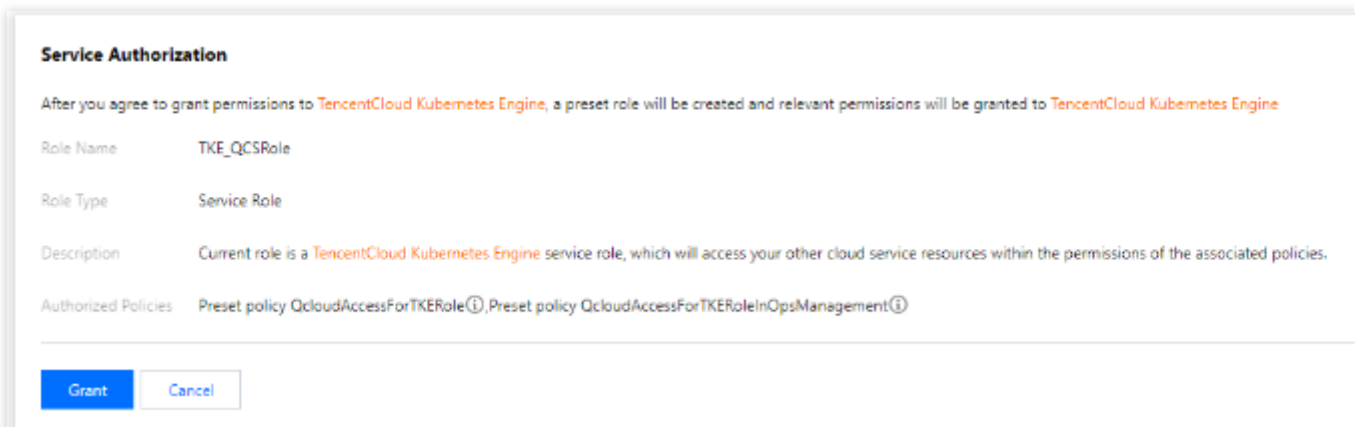
다른 Tencent Cloud 서비스 리소스에 액세스하기 전에 현재 서비스 역할을 인증하고 TKE에 대한 작업 권한을 부여해야 합니다.

Tencent Cloud 콘솔을 열고 [제품 > Tencent Kubernetes Engine](#)을 선택하여 [TKE 콘솔](#)로 들어가 프롬프트에 따라 TKE를 인증합니다. 그 다음 관련 리소스 작업 권한을 얻고 클러스터 생성을 시작할 수 있습니다. 단계는 다음과 같습니다.

1. 표시된 '서비스 인증' 대화 상자에서 정보를 보고 다음 이미지와 같이 **CAM으로 이동**을 클릭합니다.



2. '역할 관리' 페이지에서 다음 이미지와 같이 역할과 관련된 정보를 자세히 읽으십시오.



3. 권한 부여 동의를 클릭하여 권한을 부여합니다. 이제 **TKE 콘솔**로 이동하여 클러스터를 생성하고 관련 제품을 구매할 수 있습니다.

3.3 클러스터 생성

표준 관리형 클러스터를 빠르게 생성하는 방법에 대한 자세한 내용은 [표준 클러스터를 빠르게 생성](#)을 참고하십시오.

표준 관리 클러스터를 생성하는 전체 프로세스에 대한 자세한 내용은 [클러스터 생성](#)을 참고하십시오.

더 많은 종류의 클러스터가 필요한 경우 [Serverless 클러스터 생성](#), [컨테이너 인스턴스 생성](#), [엣지 클러스터 생성](#)을 참고하십시오.

3.4 워크로드 배포

이미지를 배포하거나 YAML 파일을 조정하여 워크로드를 배포할 수 있습니다.

이미지 템플릿을 통해 상태 비저장 워크로드를 배포(Deployment)하려는 경우 단일 포드로 [간단한 Nginx 서비스 생성](#) 또는 [단일 포드 WordPress 생성](#)을 참고하십시오.

사용자 지정 이미지를 통해 워크로드를 배포하려는 경우 [Hello World 서비스 수동 구축](#) 지침을 참고하십시오.

3.5 클러스터 작업

TKE는 클러스터, 애플리케이션, 스토리지 및 네트워크를 위한 관리 플랫폼입니다. 더 자세한 정보나 지침은 아래 표를 참고하십시오.

다음은 원하시는 경우	다음은 참고하시기 바랍니다
-------------	----------------

Kubernetes 명령 라인 툴인 kubectl을 사용하여 로컬 클라이언트에서 TKE 클러스터에 연결	클러스터에 연결
실행 중인 Kubernetes 클러스터 업그레이드	클러스터 업그레이드
생성된 Kubernetes 클러스터에 인스턴스 추가	노드 추가
Kubernetes 클러스터에서 노드 관리	노드 풀 생성
콘솔에서 네이티브 Kubernetes 객체 조작	Kubernetes 객체 관리
Service를 통해 일련의 컨테이너에 대한 고정 액세스 항목 제공	Service 기본 기능
Ingress 리소스를 통해 다른 포워딩 규칙 구성	Ingress 관리
TKE 스토리지 기능 활용	스토리지 관리 개요
컨테이너 네트워크 주소 범위 내의 IP 주소를 클러스터의 컨테이너에 할당	컨테이너 네트워크 개요
Kubernetes 클러스터에서 서비스 로그를 저장 및 분석	로그 수집
클러스터 모니터링 진행	TPS 인스턴스 관리
TCR(Tencent Container Registry)에서 호스팅되는 개인 이미지를 사용하여 애플리케이션 배포	TCR 엔터프라이즈 인스턴스에서 컨테이너 이미지를 사용하여 워크로드 생성

4. 초보자 가이드

클래식 네트워크에서 TKE를 사용할 수 있습니까?

아니요. 현재 VPC에서는 TKE를 사용할 수 있지만 클래식 네트워크에서는 사용할 수 없습니다.

클러스터에 기존 CVM을 추가할 수 있습니까?

네. 클러스터를 생성한 후 기존 CVM을 클러스터에 추가할 수 있습니다. 자세한 내용은 [노드 추가](#)를 참고하십시오.

내 서비스가 계속 시작 중인 이유는 무엇입니까?

컨테이너에서 실행 중인 프로세스가 없으면 서비스가 계속 시작될 수 있습니다. 서비스 시작에 대한 자세한 내용은 [이벤트 FAQ](#)를 참고하십시오.

클러스터를 생성하기 전에 네트워크 계획을 수행하려면 어떻게 해야 합니까?

클러스터 생성 시 클러스터 네트워크와 컨테이너 네트워크의 IP 범위가 겹치지 않도록 주의하십시오. 일반적으로 VPC 인스턴스의 서브넷을 클러스터의 노드 네트워크로 선택할 수 있습니다. 자세한 내용은 [컨테이너 네트워크 및 클러스터 네트워크 설명](#)을 참고하십시오.

생성된 서비스에 어떻게 액세스합니까?

다른 액세스 방법에는 다른 액세스 항목이 있습니다. 자세한 내용은 [서비스 관리 개요](#)의 서비스 액세스 섹션을 참고

하십시오.

컨테이너는 공중망에 어떻게 액세스합니까?

컨테이너가 상주하는 호스트에 공중망 IP 주소와 공중망 대역폭이 있는 경우 컨테이너는 공중망에 직접 액세스할 수 있습니다. 그렇지 않으면 공중망에 액세스하기 위해 NAT 게이트웨이가 필요합니다.

이미지 생성 방법을 모르는 경우 TKE를 사용할 수 있습니까?

TKE에 통합된 Helm 3.0과 관련된 기능을 사용하면 helm chart, TCR 및 소프트웨어 서비스와 같은 제품 및 서비스를 만들 수 있습니다. 생성된 애플리케이션은 해당 기능을 제공하기 위해 지정된 클러스터에서 실행됩니다. 자세한 내용은 [애플리케이션 관리](#)를 참고하십시오.

내 서비스에 대한 구성 파일 또는 환경 변수를 어떻게 관리합니까?

[구성 항목](#)을 편집하여 구성 파일을 관리할 수 있습니다.

서비스는 서로 어떻게 액세스합니까?

클러스터에서 Namespace가 동일한 서비스는 서로 직접 액세스할 수 있지만 Namespace가 다른 서비스는

```
<service-name>.<namespace-name>.svc.cluster.local
```

을 사용하여 액세스합니다.

5. 피드백 및 의견

TKE 제품 및 서비스를 사용하면서 의문점이나 제안 사항이 있는 경우 다음 채널을 통해 피드백을 제출할 수 있습니다. 전담 직원이 귀하의 문제를 해결하기 위해 연락을 드릴 것입니다.

링크, 콘텐츠 또는 API와 같은 제품 설명서에 대한 질문은 문서 페이지 오른쪽에 있는 [피드백 보내기](#)를 클릭하십시오. 제품 관련 문제가 발생한 경우 [티켓 제출](#)을 통해 도움을 요청할 수 있습니다.

표준 클러스터를 빠르게 생성

최종 업데이트 날짜: : 2022-04-14 16:57:28

본 문서에서는 TKE를 사용하여 컨테이너 클러스터를 빠르게 생성하는 방법을 설명합니다.

1단계: Tencent Cloud 계정 생성

이미 Tencent Cloud 계정이 있는 경우는 이 단계를 생략할 수 있습니다.

가입하려면 여기를 클릭

2단계: 온라인 충전

TKE 서비스 자체는 현재 무료이지만 실제로 사용하는 클라우드 리소스에 대해서는 요금이 부과됩니다. 본 문서에서 생성할 것은 '관리 클러스터'이며 클러스터의 작업자 노드, 영구 저장소 및 서비스에 바인딩된 CLB 등에 대해 비용을 지불해야 합니다. 관련 작업에 대한 자세한 내용은 [결제 방법](#)을 참고하십시오.

3단계: TKE 승인

[Tencent Cloud 콘솔](#)을 열고 **Tencent Cloud 서비스 > TKE**를 선택하여 TKE 콘솔로 들어가 프롬프트에 따라 TKE를 인증합니다. 이미 TKE를 승인했다면 이 단계를 건너뛰십시오.

승인하려면 여기를 클릭

4단계: 클러스터 생성

입력하려면 여기를 클릭

클러스터 정보

'클러스터 정보' 페이지에서 클러스터 이름을 입력하고 클러스터 지역, 클러스터 네트워크, 컨테이너 네트워크를 선택합니다. 다른 기본 옵션을 변경하지 않은 상태로 유지하고 아래와 같이 **다음**을 클릭합니다.

Cluster Name

Region

Cluster Network

Container Network

Up to 60 characters

Kubernetes Version: 1.18.4

Guangzhou	Shenzhen Finance	Shanghai	Shanghai Finance	Beijing	Chengdu	Hong Kong, China	Singapore	Bangkok	Mumbai		
Tokyo	Silicon Valley	Virginia	Frankfurt	Moscow	Nanjing	Jakarta	Jinan	Hangzhou	Wuhan	Changsha	Shenyang
Shijiazhuang	Hefei	Beijing finance	Toronto	São Paulo							

Tencent Cloud resources in different regions cannot communicate via private network. The region cannot be changed after purchase. Please choose a region close to your end-users to minimize download speed.

Cluster Network: kafuttest

If the current networks are not suitable, please go to the console to [create a VPC](#).

<input type="checkbox"/> Subnet ID	Subnet Name	Availability Zone	Remaining IPs	
<input type="checkbox"/> subnet-259h6vi5	1	Shanghai Zone 1	250	Elastic clusters are not available in t...
<input type="checkbox"/> subnet-6uxmas5v	3	Shanghai Zone 1	253	Elastic clusters are not available in t...
<input type="checkbox"/> subnet-2hjsswez	8	Shanghai Zone 1	253	Elastic clusters are not available in t...
<input type="checkbox"/> subnet-qt6x8emp	4	Shanghai Zone 1	253	Elastic clusters are not available in t...
<input type="checkbox"/> subnet-0ev2c4u3	6	Shanghai Zone 1	253	Elastic clusters are not available in t...

- **클러스터 이름:** 클러스터 이름을 입력합니다. 이 문서에서는 'test'를 클러스터 이름으로 사용합니다.
- **리전:** 가장 가까운 리전을 선택합니다.
- **클러스터 네트워크:** 노드 네트워크 주소 범위 내의 IP 주소를 클러스터의 서버에 할당합니다. 여기에서 VPC를 선택합니다.
- **컨테이너 네트워크:** 컨테이너 네트워크 주소 범위 내의 IP 주소를 클러스터의 컨테이너에 할당합니다. 여기에서 사용 가능한 컨테이너 네트워크를 선택합니다.

모델 선택

'모델 선택' 페이지에서 청구 모드를 확인하고, 가용존과 해당 서브넷을 선택하고, 노드 모델을 확인하고 아래와 같이 다음을 클릭합니다.

Node Source

Master Node

The default cluster's Master, Etc and other components are maintained and managed by Tencent Cloud. For the convenience of management, you can also purchase CVM to deploy the Master. For details, please refer to ["Cluster Hosting Mode Instruction"](#) [🔗](#)

Billing Mode

Worker Configurations

Availability Zone

Node Network 250/253 subnet IPs available
 CIDR:10.0.0.0/16
 If the current networks are not suitable, please go to the console to [create a VPC](#) [🔗](#) or [create a subnet](#) [🔗](#).

Model S2.SMALL1(Standard S2,1 core1GB) [🔗](#)

System Disk Premium Cloud Disk 50GB [🔗](#)

Data Disk Purchase Later [🔗](#)

Public Network Bandwidth Bandwidth pack Bandwidth cap: 1Mbps [🔗](#)

Node Name Auto-generated [🔗](#)

CVM Quantity
 VPC network limit: Up to 250 IPs available for current node network

[▶ Advanced Settings](#)

- **노드 소스:** 노드 추가 또는 기존 노드를 선택할 수 있습니다. 여기에서 '노드 추가'를 선택합니다.
- **Master 노드:** 관리 및 자체 배포의 두 가지 클러스터 모드 중에서 선택할 수 있습니다. 여기에서 '관리'를 선택합니다.
- **청구 모드:** 종량제만 선택할 수 있습니다. 여기에서 '종량제'를 선택합니다.
- **Worker 구성:** 가용존과 해당 서브넷을 선택하고 노드 모델을 확인하기만 하면 됩니다. 다른 기본 설정은 변경하지 마십시오.
 - **가용존:** 여기에서 '광저우 영역3'을 선택합니다.
 - **노드 네트워크:** 여기에서 현재 VPC의 서브넷을 선택합니다.
 - **모델:** 여기에서 'S1.SMALL1(표준 S1, 1코어1GB)'을 선택합니다.

CVM 설정

‘CVM 구성’ 페이지에서 로그인 방법을 선택하고 다른 기본 설정을 변경하지 않고 다음을 클릭합니다.

The screenshot shows the configuration page for a Tencent Cloud CVM. The 'Login Method' section is highlighted with a red box. It includes options for 'SSH Key Pair', 'Random Password', and 'Custom Password'. Other sections include 'Container Directory', 'Security Group', 'Security Services', and 'Cloud Monitor', each with checkboxes and links for details.

- 로그인 방법: SSH 키 쌍, 임의 비밀번호 또는 사용자 정의 비밀번호를 선택할 수 있습니다. 여기서 ‘임의의 비밀번호’를 선택합니다.

컴포넌트 구성

‘컴포넌트 구성’ 페이지에서 스토리지, 모니터링, 이미지 등 필요한 컴포넌트를 선택할 수 있습니다. 이러한 컴포넌트가 필요하지 않은 경우 다음을 직접 클릭합니다. 여기서 컴포넌트를 설치하지 않고 다른 기본 설정을 변경하지 않도록 선택합니다.

정보 확인

‘정보 확인’ 페이지에서 클러스터에 대해 선택한 구성 정보 및 청구 모드를 확인하고 아래 그림과 같이 완료를 클릭합니다.

Selected Configuration

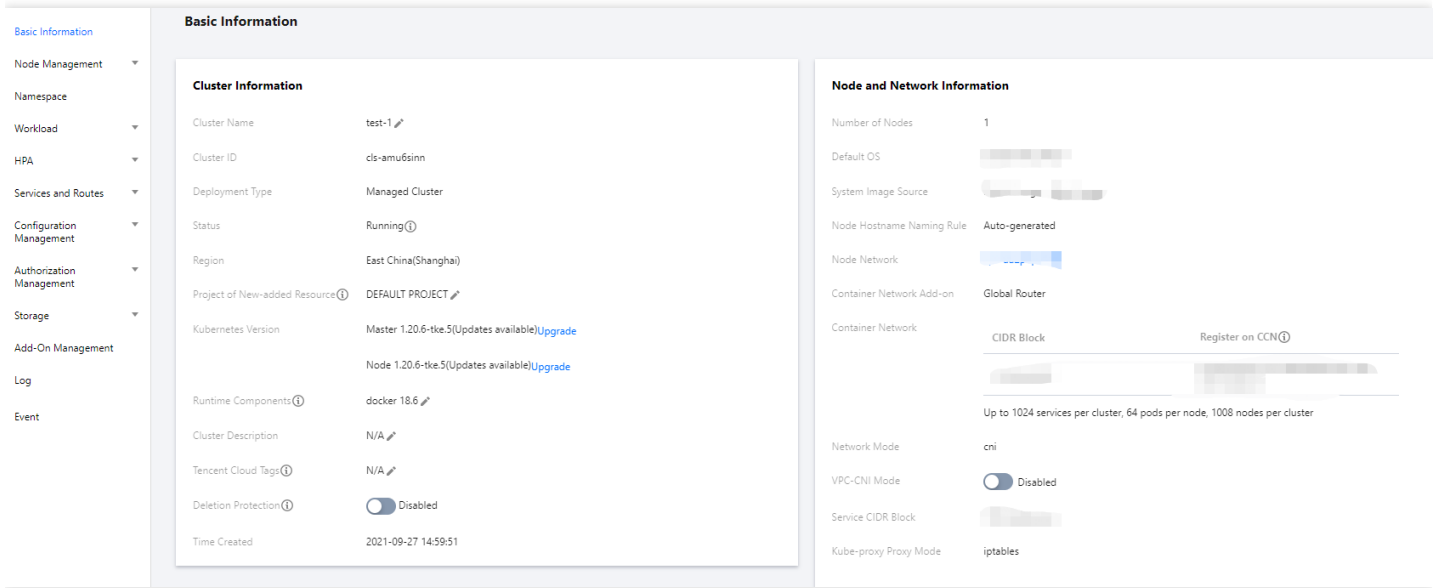
Cluster Name	lili
Region	East China(Shanghai)
Container Network	GR, 192.168.0.0/16, 1024 Service/cluster, 64 Pod/node, up to 1008 nodes
Operating System	Tencent Linux 2.4 64bit
Cluster Type	Managed Cluster
Billing Mode	Pay-as-you-go
Operating System ⓘ	Tencent Linux 2.4 64bit Public Image -Basic Image
Node	AZ: Shanghai Zone 1 Model:S2.SMALL1(Standard S2,1 core1GB) System disk: Premium Cloud Disk 50GB Data disk: purchase later Public bandwidth: bandwidth pack; bandwidth cap 1Mbps Amount:1
Add-on	CBS Tencent Cloud CBS
Cluster Auditing	Enabled

Fees 

요금을 지불한 후 첫 번째 클러스터를 생성할 수 있습니다. 그런 다음 [TKE 콘솔](#)에서 생성한 관리형 클러스터를 볼 수 있습니다.

5단계: 클러스터 보기

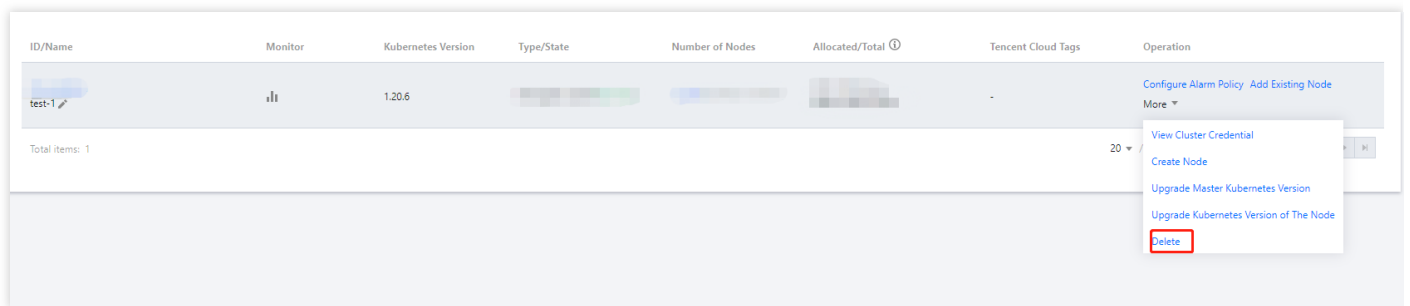
[클러스터 목록](#)에서 생성된 클러스터를 볼 수 있습니다. 클러스터 ID를 클릭하여 세부 정보 페이지로 들어가면 아래와 같이 '기본 정보' 페이지에서 클러스터, 노드 및 네트워크 정보를 볼 수 있습니다.



6단계: 클러스터 삭제

일단 시작되면 클러스터는 리소스를 소비하기 시작합니다. 불필요한 비용을 피하기 위해 아래 단계에 따라 모든 리소스를 지울 수 있습니다.

1. 왼쪽 사이드바에서 **클러스터**를 선택합니다. '클러스터 관리' 페이지에서 삭제할 클러스터가 있는 **raw**의 오른쪽에 있는 **더 보기 > 삭제**를 선택합니다.



2. '클러스터 삭제' 팝업 창에서 관련 정보를 확인하고 **확인**을 클릭하면 클러스터가 삭제됩니다.

후속 작업: 클러스터 사용

이제 본 문서를 읽은 후 TKE에서 클러스터를 생성 및 삭제하는 방법을 알게 되었습니다. 생성된 클러스터에서 워크로드를 설정하고 서비스를 생성할 수 있습니다. 일반적인 작업은 다음과 같습니다.

- [간단한 Nginx 서비스 생성](#)

- [단일 포트 WordPress](#)
- [TencentDB를 사용하는 WordPress 서비스](#)
- [수동으로 Hello World 서비스 구축](#)
- [간단한 Web 애플리케이션 구축](#)

문제 해결

전체 과정에서 문제가 발생하면 [티켓 제출](#)하여 저희에게 연락하십시오.

Demo

간단한 Nginx 서비스 생성

최종 업데이트 날짜: : 2023-05-06 19:41:07

작업 시나리오

본 문서는 컨테이너 클러스터에서 Nginx 서비스를 빠르게 생성하는 방법을 설명합니다.

전제 조건

[Tencent Cloud 계정을 등록](#)합니다.

클러스터를 생성합니다. 자세한 내용은 [클러스터 생성](#)을 참고하십시오.

작업 단계

Nginx 서비스 생성

1. TKE 콘솔에 로그인하고 왼쪽 사이드바에서 [Cluster](#)를 선택합니다.
2. [클러스터 관리](#) 페이지에서 대상 클러스터의 ID를 클릭하여 기본 정보 페이지로 이동합니다.
3. 워크로드 > [Deployment](#) 페이지에서 [생성](#)을 클릭합니다. 매개변수에 대한 자세한 내용은 [Deployment 생성](#)을 참고하십시오.
4. [Deployment 생성](#) 페이지에서 아래와 같이 워크로드의 기본 정보를 설정합니다.

Name

Up to 63 characters, including lowercase letters, numbers, and hyphens ("-"). It must begin with a lowercase letter, and end with a number or lowercase letter.

Description

Namespace

Labels =

[Add](#)

The key name cannot exceed 63 chars. It supports letters, numbers, "/" and "-". "/" cannot be placed at the beginning. A prefix is supported. [Learn more](#)

The label key value can only include letters, numbers and separators ("-", "_", "."). It must start and end with letters and numbers.

OS type

Configurations are initialized when you change the OS type for the container.

Volume (optional) [Add volume](#)

It provides storage for the container. It can be a node path, cloud disk volume, file storage NFS, config file and PVC, and must be mounted to the specified path of

워크로드 이름: 본문은 nginx를 예로 들어 설명합니다.

Description: 관련 워크로드 정보를 지정합니다.

레이블: 여기서 기본값은 `k8s-app = nginx` 입니다.

네임스페이스: 필요에 따라 네임스페이스를 선택합니다. 기본값은 default입니다.

Volume: 요구 사항에 따라 워크로드가 마운트될 볼륨을 설정합니다. 자세한 내용은 [스토리지 볼륨 사용 설명](#)을 참고하십시오.

5. 지침에 따라 'Pod의 컨테이너'를 구성합니다. 아래 이미지를 참고하십시오.

Containers in the Pod

Name

Up to 63 characters. It supports lower case letters, numbers, and hyphen ("-") and cannot start or end with "-".

Image [Select image](#)

Image tag

Pull image from remote registry

If the image pull policy is not set, when the image tag is empty or "latest", the "Always" policy is used, otherwise "IfNotPresent" is used.

Environment variable [Add variable](#)

To enter multiple key-value pairs in a batch, you can paste multiple lines of key-value pairs (key=value or key:value) in the "Variable name" field. They will be filled accordingly.

CPU/memory limit

CPU limit - -core

Memory limit - MiB

Request is used to pre-allocate resources. When the nodes in the cluster do not have the required number of resources, the container will fail to create. Limit is used to set an upper limit for resource usage for a container, so as to avoid over usage of node resources in case of exceptions.

GPU resource

Number of cards:

Configure the minimum GPU resource usage of this workload. Please make sure that the cluster has enough GPU resource.

Target port [Add container port](#)

[Advanced settings](#)

주요 매개변수는 다음과 같이 설명됩니다.

Name: 포드의 컨테이너의 이름을 입력합니다. 여기서는 `test`를 예시로 사용합니다.

이미지: 이미지 선택을 클릭하고 팝업 창에서 **Docker Hub** 이미지 > **nginx**를 선택한 후 **확인**을 클릭합니다.

Image Tag: `latest` 기본값을 사용합니다.

이미지 풀링 정책: 필요에 따라 사용 가능한 **Always**, **IfNotPresent**, **Never** 세 가지 정책 중 하나를 선택합니다. 이 예시에서는 **기본 정책**이 사용됩니다.

6. '인스턴스 수'에서 다음 정보에 따라 서비스의 인스턴스 수를 설정합니다. 본문에서는 **수동 조정**을 선택하고 인스턴스 번호를 1로 설정합니다. 아래 이미지를 참고하십시오.

Number of Pods ⓘ

Manual adjustment Set the number of pods directly

Number of Pods 1

Auto adjustment Automatically adjust the number of pods if any of the setting conditions are met [View more](#) ⓘ

7. 다음 이미지와 같이 워크로드의 액세스 모드를 설정합니다.

Access settings (Service)

Service Enable

Service access ClusterIP NodePort LoadBalancer (public network) LoadBalancer (private network) [Suggestions](#) ⓘ

A classic public CLB is automatically created for Internet access: It supports TCP and UDP protocols and is applicable to web front-end services.

If you need to forward via internet using HTTP/HTTPS protocols or by URL, you can go to Ingress page to configure Ingress for routing. [Learn more](#) ⓘ

IP version

The IP version cannot be changed later.

Availability zone

"Random AZ" is recommended to avoid the instance creation failure due to the resource shortage in the specified AZ.

ISP type

Network billing mode

Bandwidth cap 10 Mbps

Load Balancer

! Automatically create a CLB for public/private network access to the service. The lifecycle of the CLB is managed by TKE. Do not manually modify the CLB listener created by TKE. [Learn more](#) ⓘ

Protocol ⓘ	Target port ⓘ	Node port ⓘ	Port ⓘ	Secret ⓘ
TCP	Port listened by application in cor	Range: 30000-32767	Should be the same as the target	The current protocol does not support Secret.

[Add port mapping](#)

[Advanced settings](#)

Service: '활성화'를 선택합니다.

서비스 액세스: '공중망 액세스'를 선택합니다.

로드 밸런서: 필요에 따라 로드 밸런서를 생성하거나 선택합니다.

포트 매핑: TCP 프로토콜을 선택하고 컨테이너 포트와 서비스 포트를 모두 포트 80으로 설정합니다.

프로토콜: 통신할 프로토콜 유형을 선택합니다.

컨테이너 포트: 컨테이너 내 응용 프로그램 수신 포트, 포트 범위는 1~65535입니다.

호스트 포트: CVM IP + 호스트 포트를 통해 서비스 액세스가 가능하고, 포트 범위는 30000~32767이며, 입력하지 않을 경우 자동으로 할당됩니다.

서비스 포트: 서비스는 CLB 인스턴스 도메인 이름 또는 IP + 서비스 포트, 클러스터 외부에서 액세스; 또는 서비스 이름 + 서비스 포트, 클러스터 내에서 액세스할 수 있습니다.

Secret: TCP SSL 프로토콜이 선택된 경우에만 선택해야 합니다.

주의사항

서비스가 속한 클러스터의 보안 그룹에 있는 노드 네트워크, 컨테이너 네트워크 및 포트 30000 - 32768은 인터넷에 개방되어 있어야 합니다. 그렇지 않으면 TKE를 사용하지 못할 수 있습니다. 자세한 내용은 [TKE 보안 그룹 설정](#)을 참고하십시오.

8. **workload** 생성을 클릭하여 Nginx 서비스 생성을 완료합니다.

Nginx 서비스에 액세스

Nginx 서비스는 다음 두 가지 방법으로 액세스할 수 있습니다.

Cloud Load Balancer(CLB) IP를 사용하여 Nginx 서비스에 액세스

1. 왼쪽 사이드바에서 **클러스터**를 클릭하여 '클러스터 관리' 페이지로 이동합니다.
2. 대상 클러스터 ID를 클릭하고 **서비스 및 라우팅 > Service**를 선택합니다.
3. Service 목록 페이지에서 아래와 같이 Nginx 서비스의 로드 밸런서 IP를 복사합니다.

Name	Labels	Type	Selector	IP address	Time created	Operation
kubernetes	componentapiserver providerkubernetes	ClusterIP	-	(PV4) (Service IP)	2023-03-06 11:53:32	Update configuration Delete

4. 브라우저의 주소 표시줄에 CLB IP를 입력하고 **'Enter'** 키를 눌러 서비스에 액세스합니다.

서비스 이름을 사용하여 Nginx 서비스에 액세스

클러스터의 다른 서비스 또는 컨테이너는 서비스 이름을 사용하여 직접 액세스할 수 있습니다.

Nginx 서비스 확인

서비스가 성공적으로 생성되면 서비스에 액세스할 때 Nginx 서버 시작 페이지로 직접 들어갑니다. 아래 이미지를 참고하십시오.

Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

더 많은 Nginx 설정

컨테이너를 생성할 수 없는 경우 솔루션에 대한 [이벤트 FAQ](#)를 검색할 수 있습니다.

Hello World 서비스 수동 구축

최종 업데이트 날짜: : 2023-05-06 19:41:07

작업 시나리오

본 문서는 컨테이너 클러스터에서 Node.js Hello World 서비스를 빠르게 생성하는 방법을 설명합니다. Docker 이미지를 빌드하는 방법에 대한 자세한 내용은 [How to Build a Docker Image](#)를 참고하십시오.

전제 조건

클러스터를 생성합니다. 자세한 내용은 [클러스터 생성](#)을 참고하십시오.

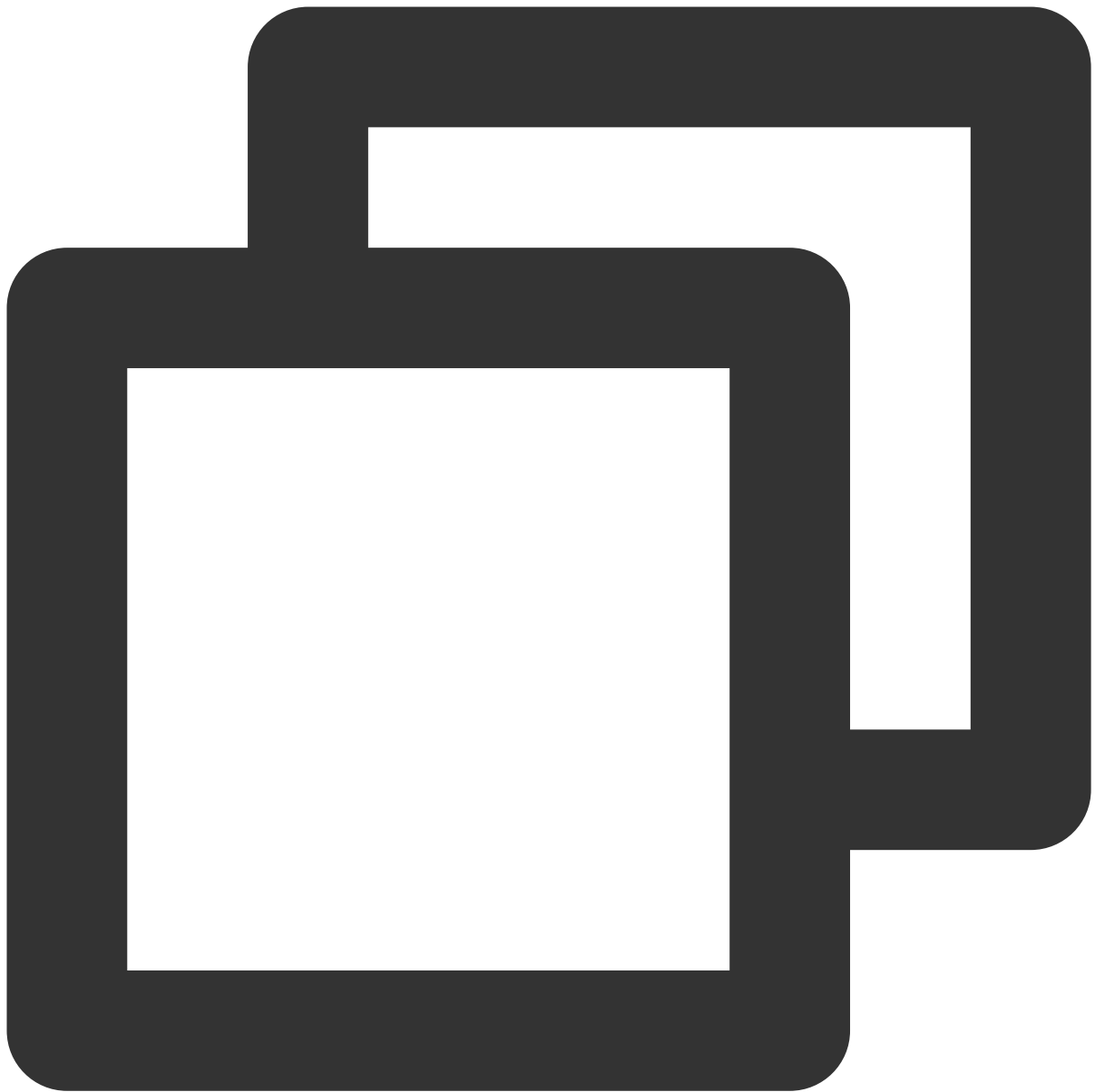
Node.js가 설치된 노드에 로그인해야 합니다.

작업 단계

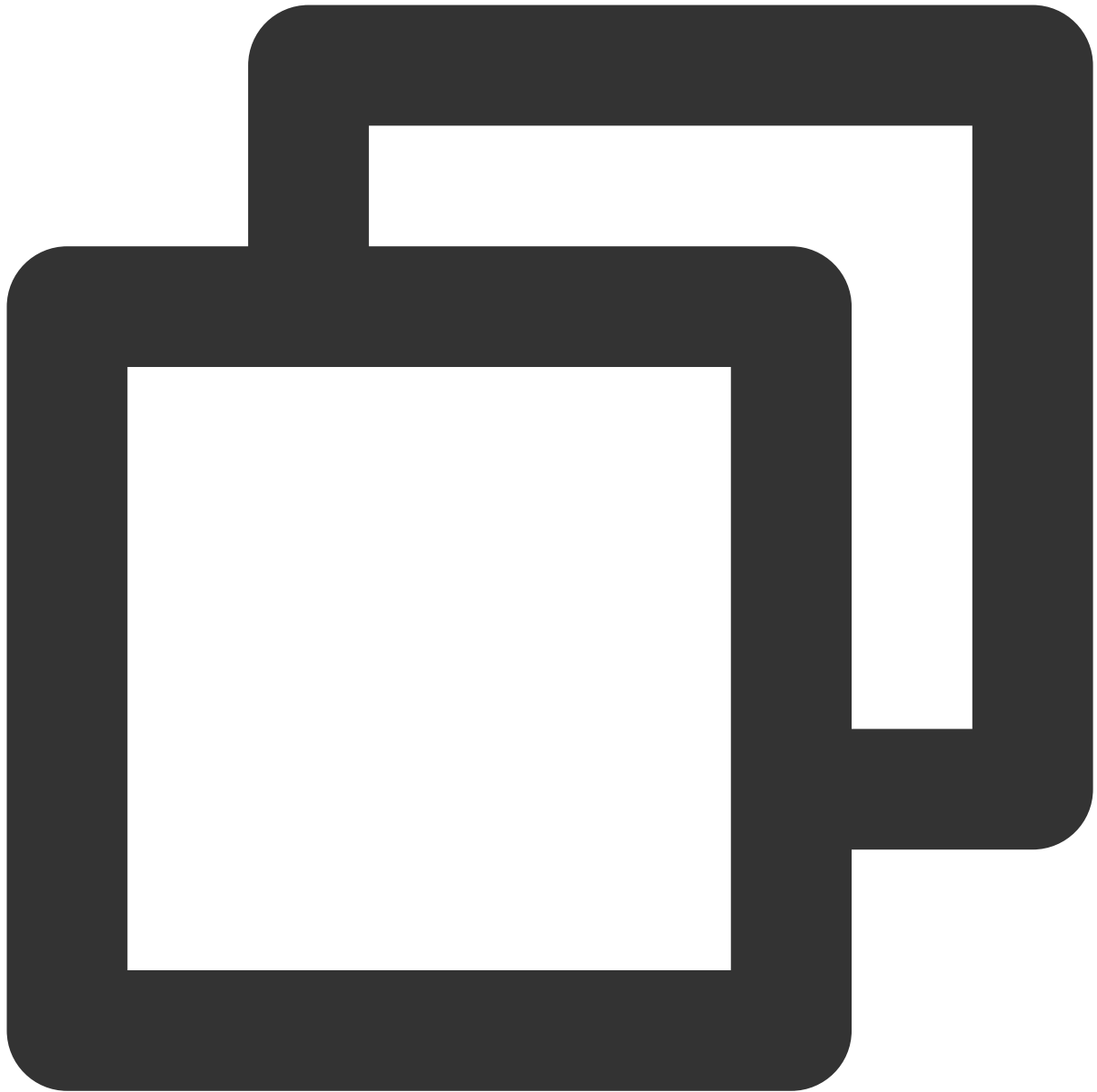
이미지 생성 코드 작성

애플리케이션 개발

1. 다음 명령어를 순서대로 실행하여 hellonode 디렉터리를 생성하고 이동합니다.

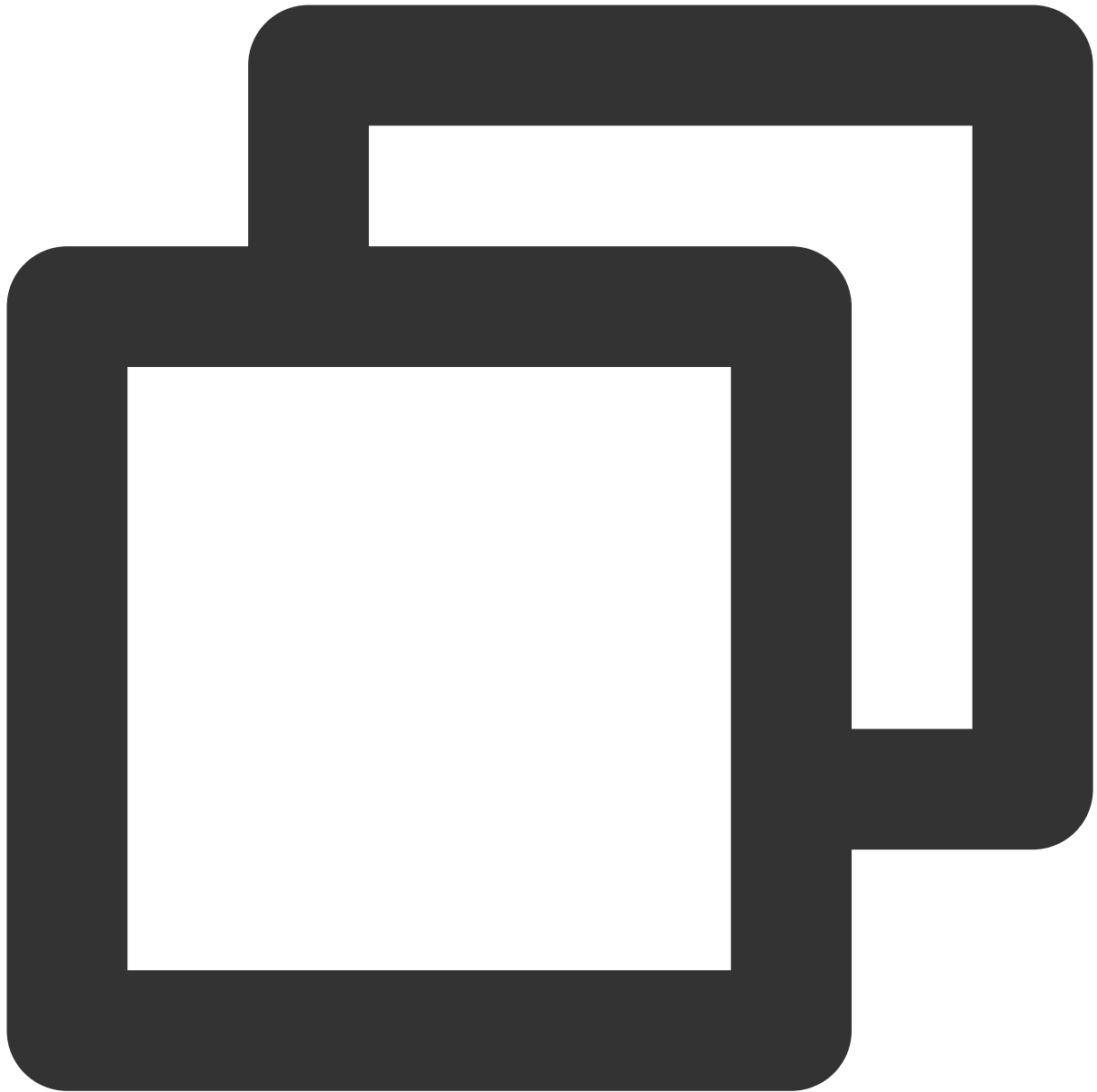


```
mkdir hellonode
```



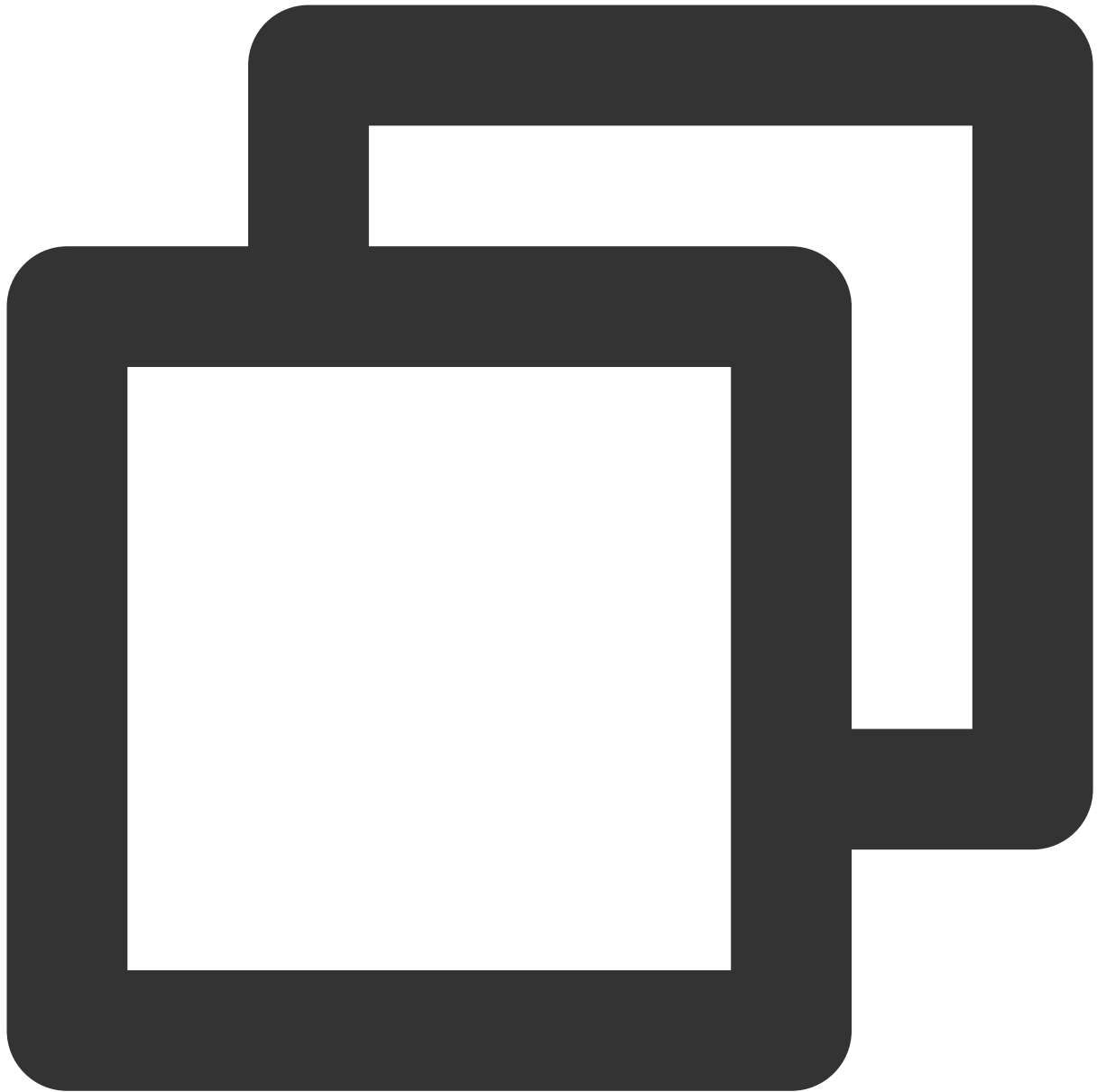
```
cd hellonode/
```

2. 다음 명령을 실행하여 `server.js` 파일을 생성하고 엽니다.



```
vim server.js
```

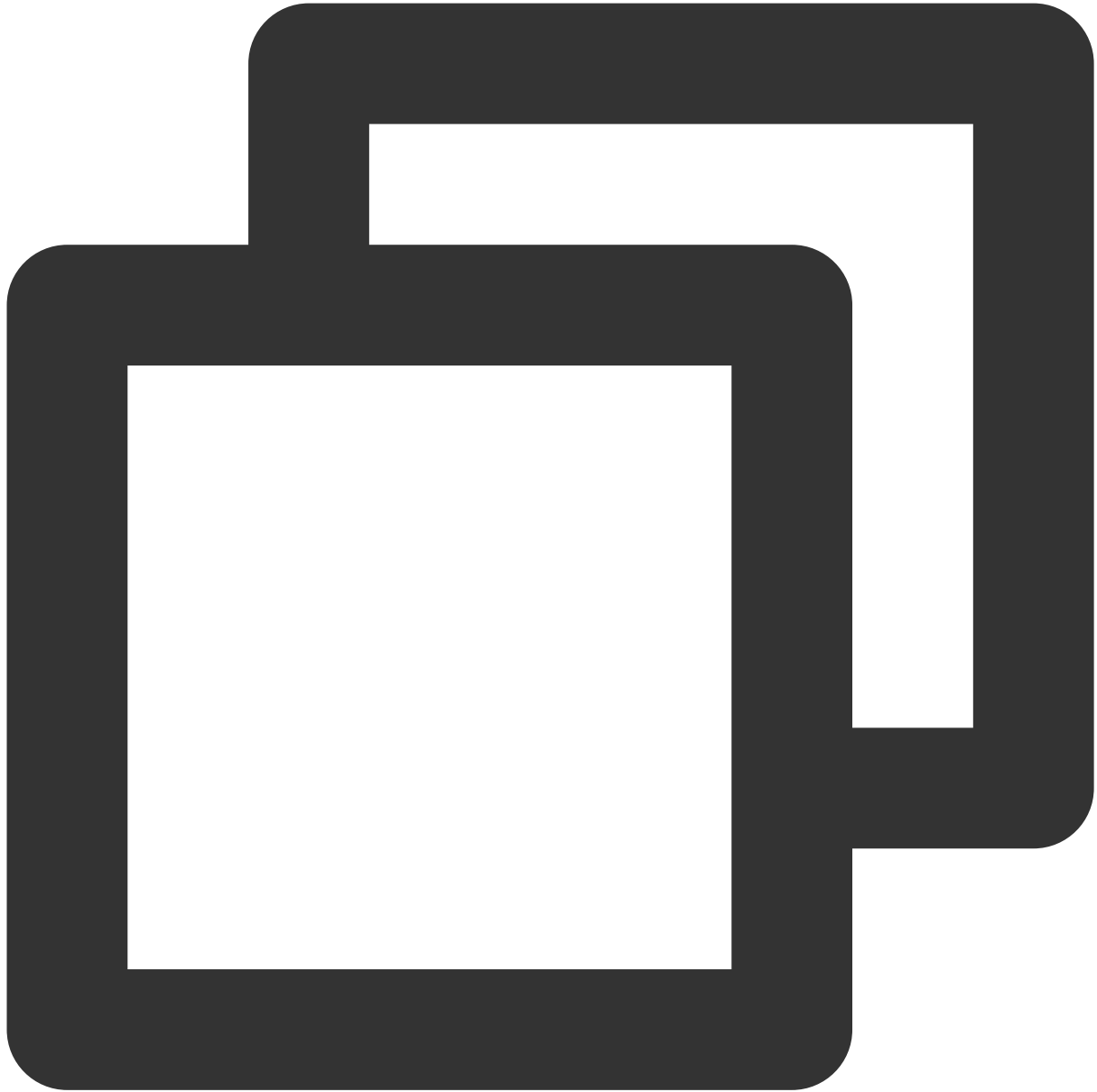
3. **i**를 눌러 편집 모드로 전환하고 다음 코드를 `server.js`에 복사합니다.



```
var http = require('http');
var handleRequest = function(request, response) {
  console.log('Received request for URL: ' + request.url);
  response.writeHead(200);
  response.end('Hello World!');
};
var www = http.createServer(handleRequest);
www.listen(80);
```

'Esc' 키를 누르고 `:wq` 를 입력하여 파일을 저장하고 반환합니다.

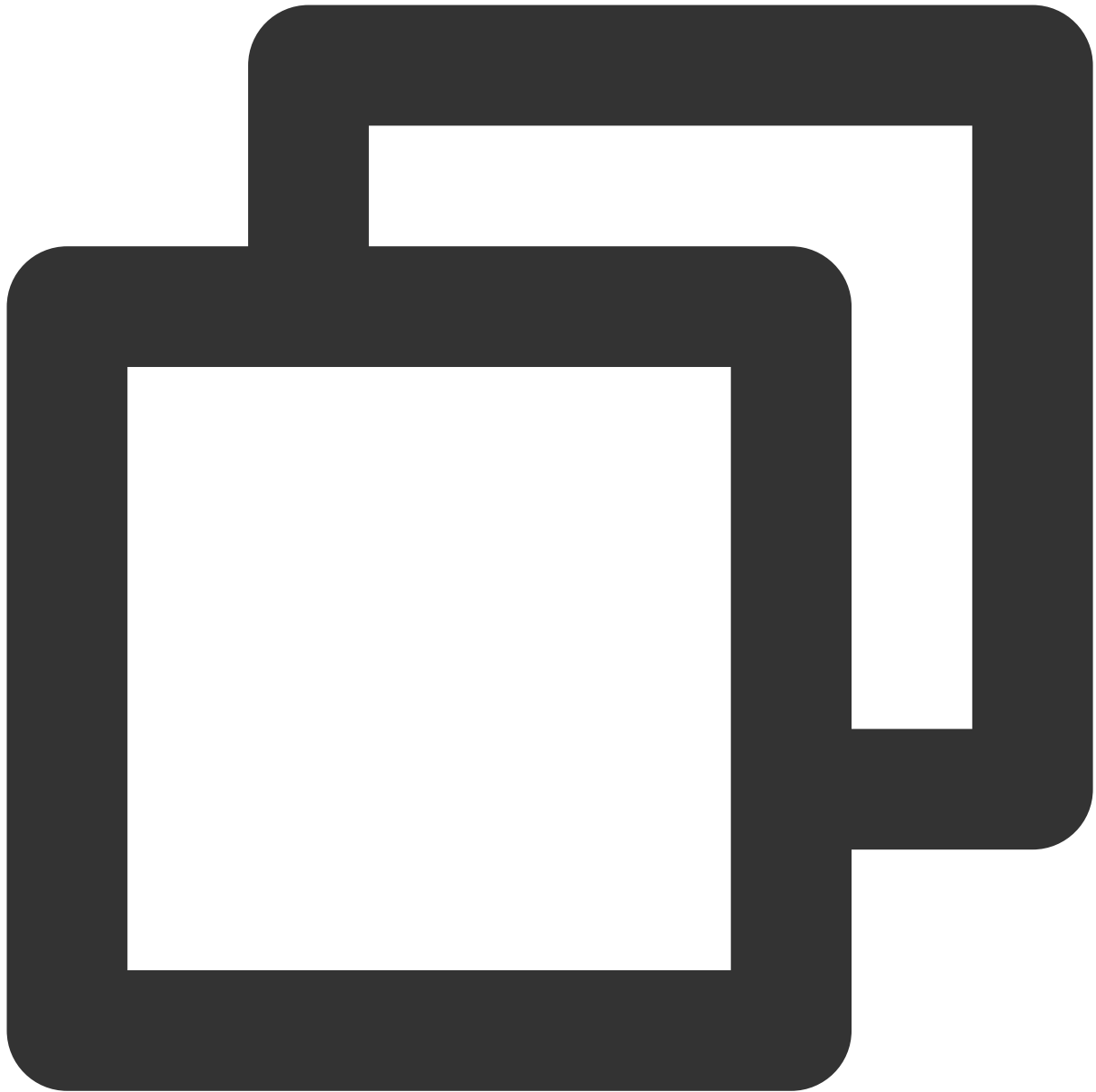
4. 다음 명령어를 실행하여 `server.js` 파일을 실행합니다.



```
node server.js
```

5. 다음과 같은 방법으로 Hello World 프로그램을 테스트할 수 있습니다.

방법1: 노드에 다시 로그인하고 다음 명령을 실행합니다.



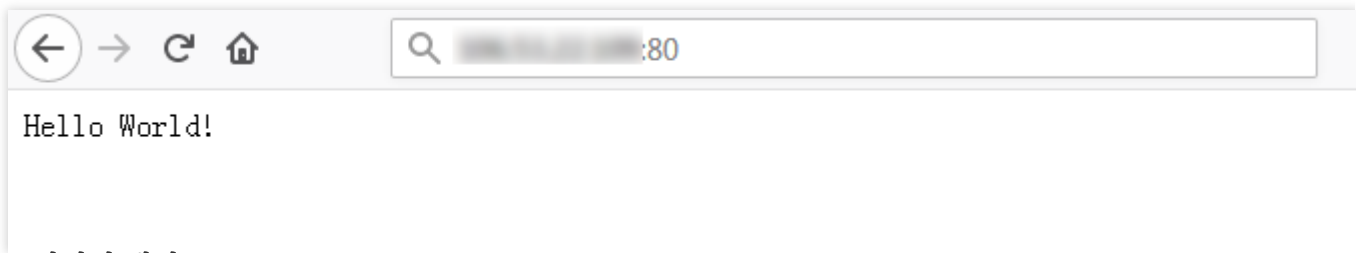
```
curl 127.0.0.1:80
```

다음과 같은 정보가 나오면 Hello World 프로그램이 정상적으로 실행되고 있는 것입니다.

```
[root@VM_2_5_centos ~]# curl 127.0.0.1:80  
Hello World! [root@VM_2_5_centos ~]#
```

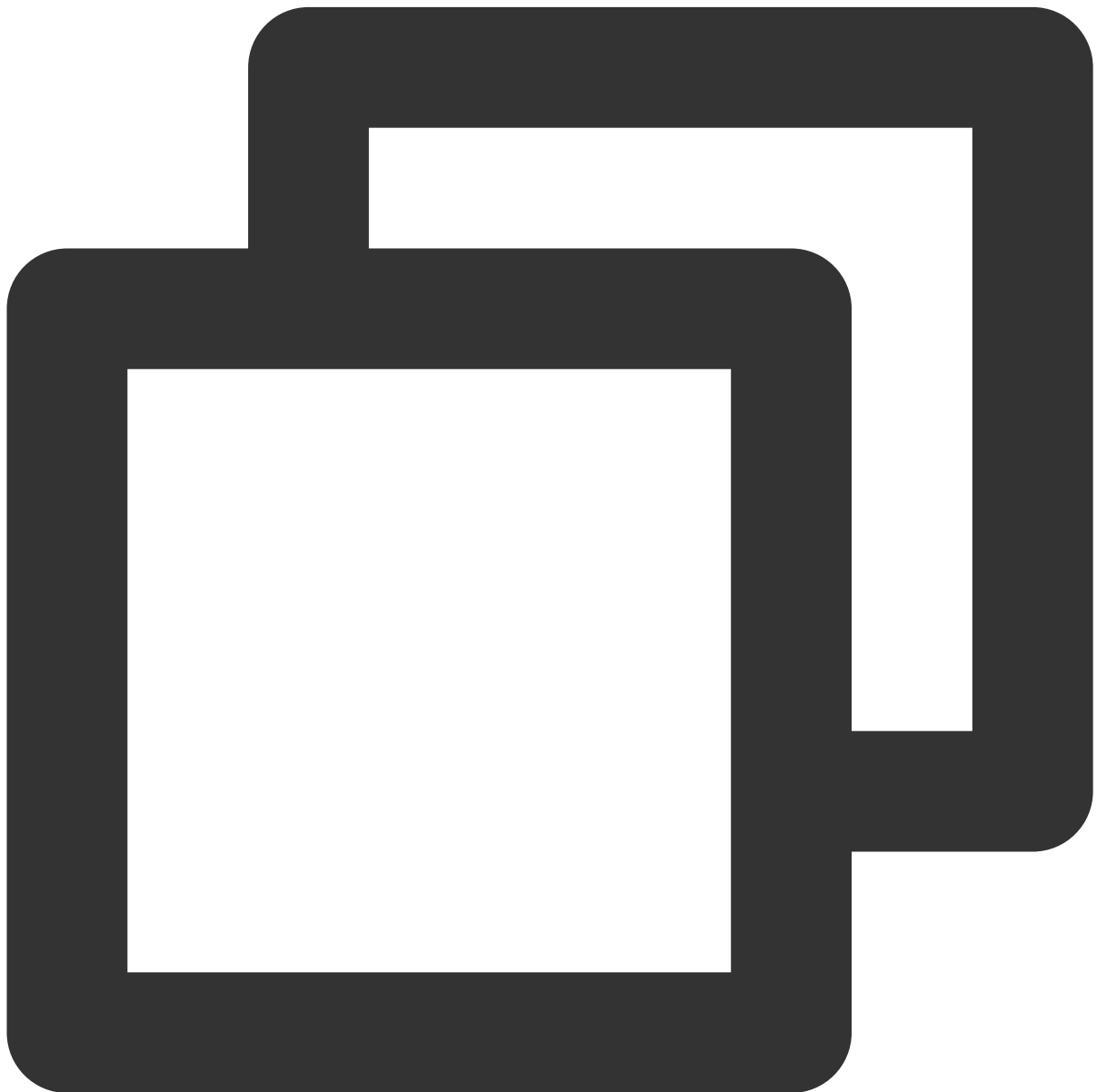
방법2: 로컬 브라우저를 열고 CVM 인스턴스의 공중망 IP 주소:구성된 포트 번호를 입력하여 프로그램에 액세스합니다. 포트 번호는 80입니다.

다음과 같은 정보가 나오면 Hello World 프로그램이 정상적으로 실행되고 있는 것입니다.

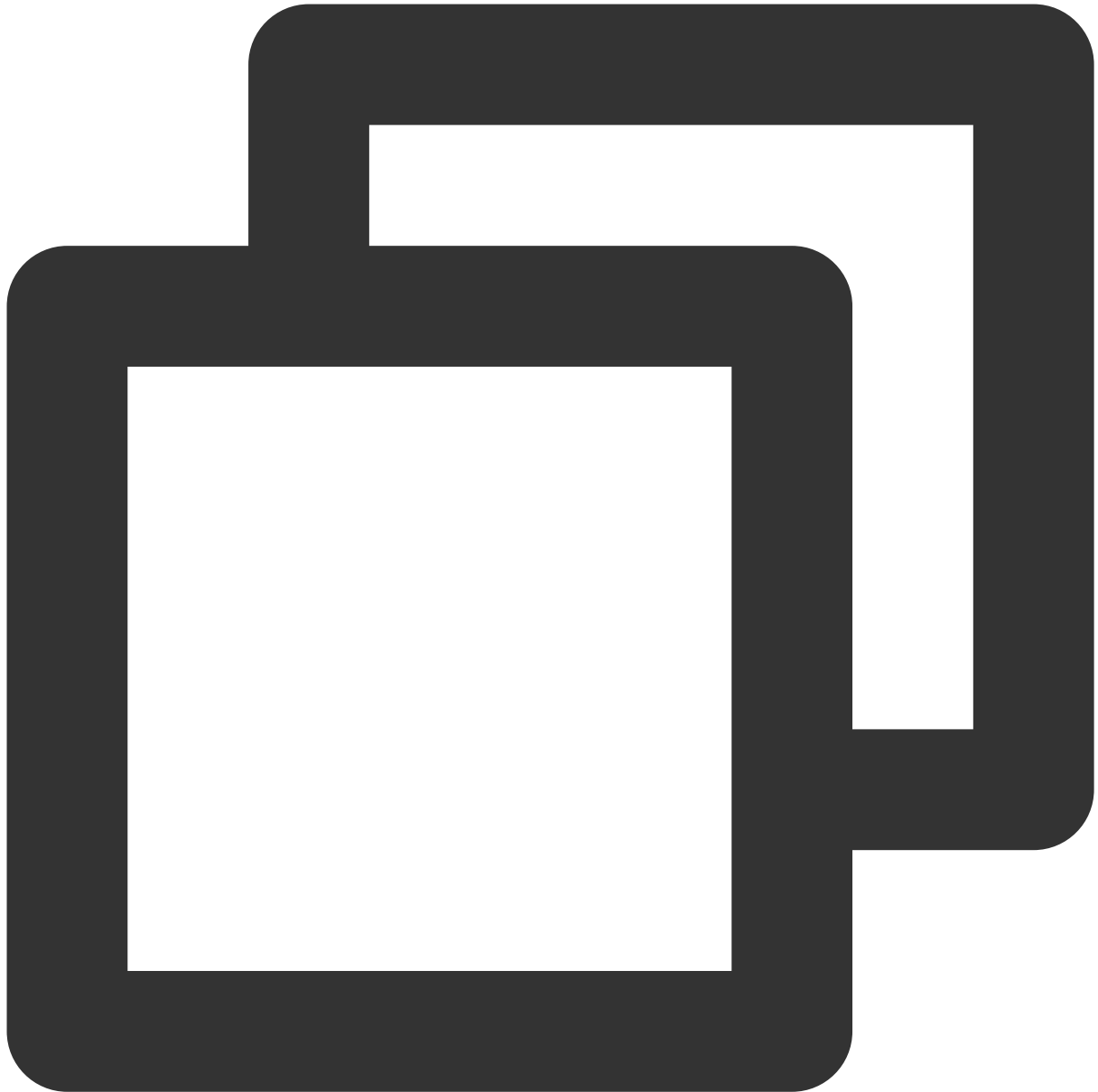


Docker 이미지 생성

1. 다음 명령어를 순차적으로 실행하여 hellonode 디렉터리에 Dockerfile 파일을 생성합니다.

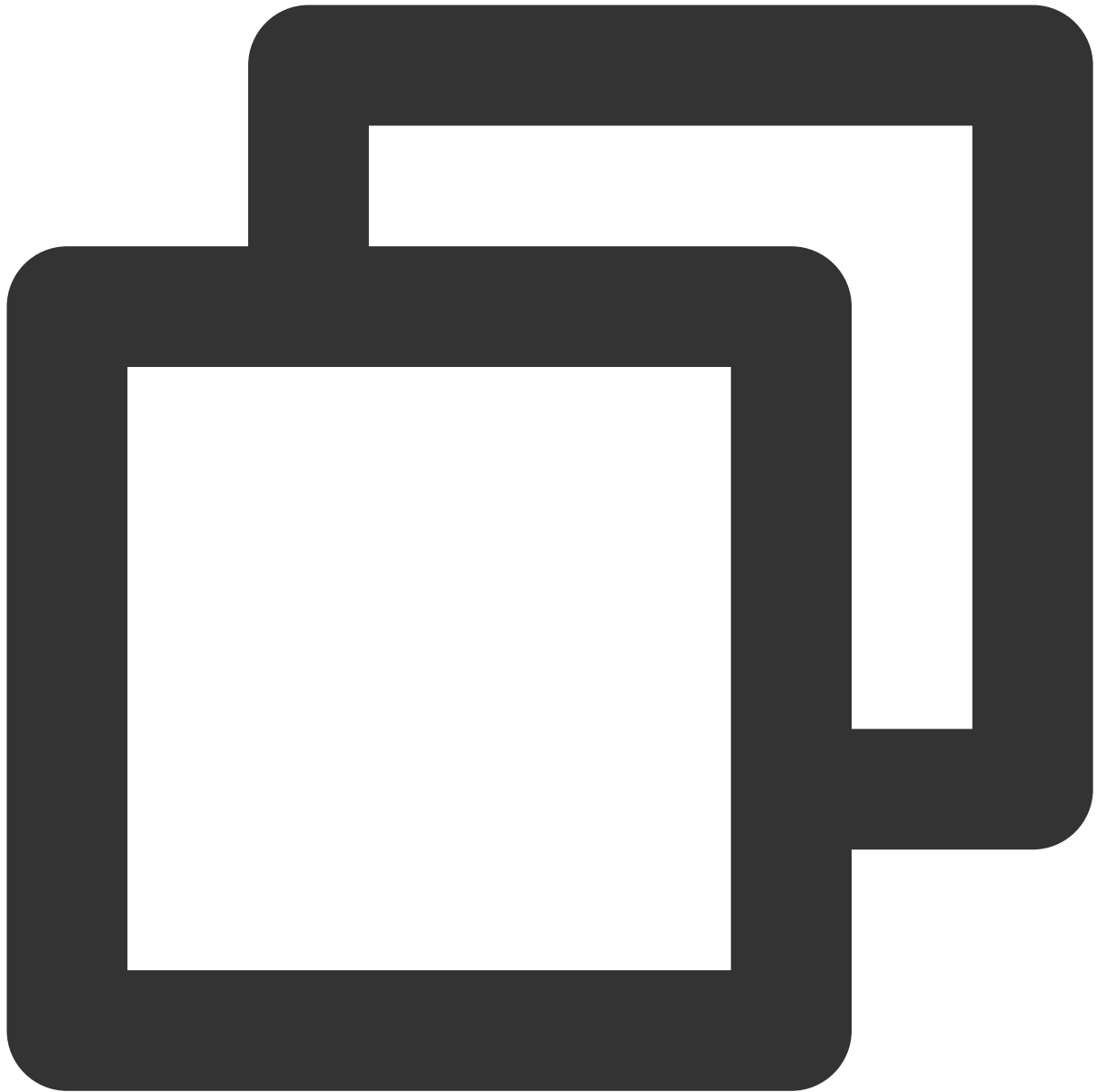


```
cd hellonode
```



```
vim Dockerfile
```

2. **i**를 눌러 편집 모드로 전환하고 다음 코드를 Dockerfile 파일에 복사합니다.



```
FROM node:4.4
EXPOSE 80
COPY server.js .
CMD node server.js
```

'Esc' 키를 누르고 `:wq` 를 입력하여 파일을 저장하고 반환합니다.

3. 노드는 docker를 설치하고 docker 서비스를 시작합니다.



```
yum install -y docker  
systemctl start docker
```

4. 다음 명령어를 실행하여 이미지를 빌드합니다.



```
docker build -t hello-node:v1 .
```

5. 다음 명령어를 실행하여 빌드된 hello-node 이미지를 확인합니다.



```
docker images
```

다음 정보가 나오면 hello-node 이미지가 성공적으로 빌드된 것입니다. 다음 이미지와 같이 나중에 사용할 수 있도록 IMAGE ID를 기록해 두십시오.

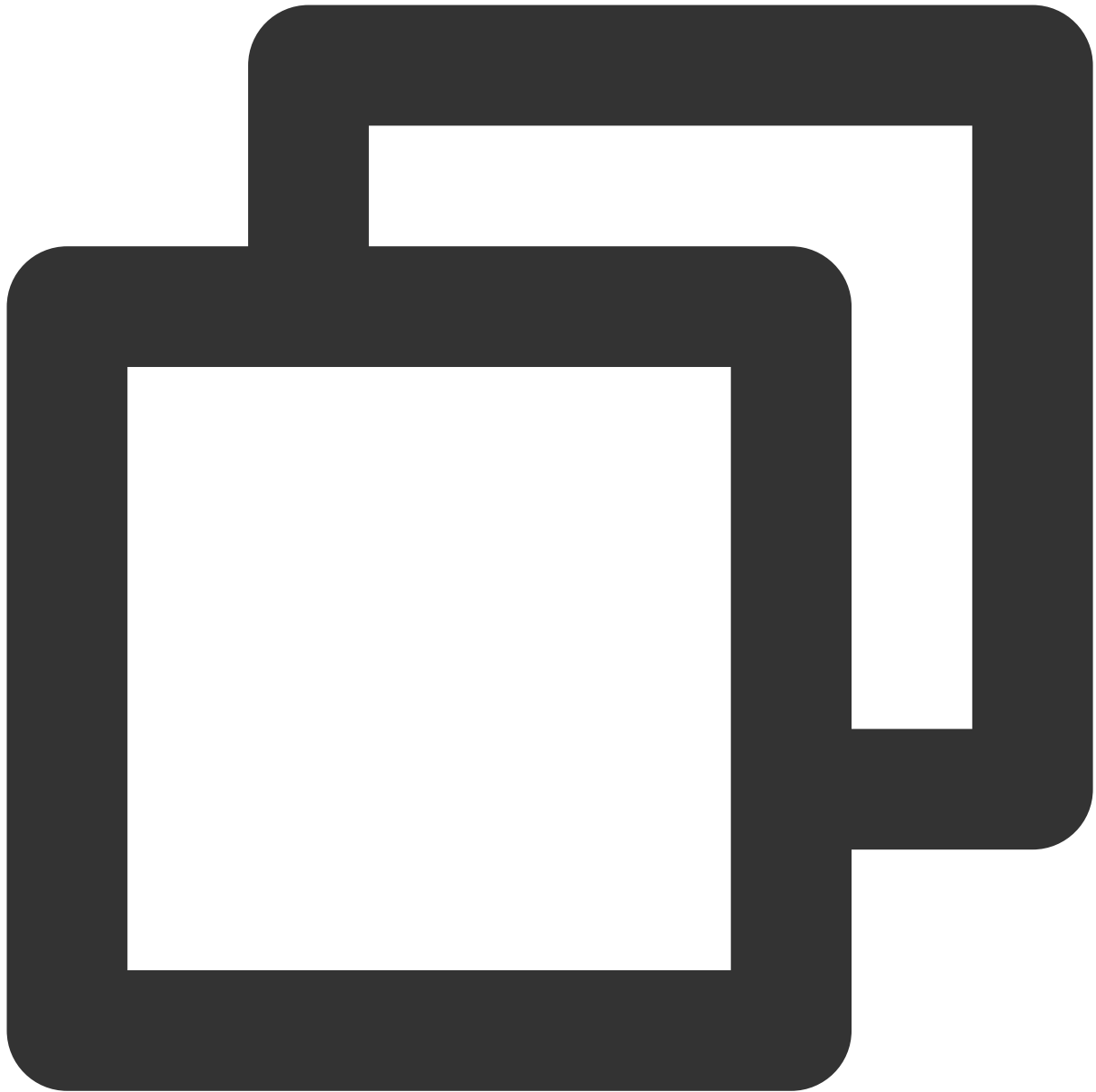
```
[root@VM_2_5_centos hellonode]# docker images
REPOSITORY          TAG          IMAGE ID          CR
hello-node          v1          5ac              2
```

Tencent Cloud 이미지 레지스트리에 이미지 업로드

다음 명령을 순서대로 실행하여 이미지를 Tencent Cloud 이미지 레지스트리에 업로드합니다.



```
docker tag IMAGEID ccr.ccs.tencentyun.com/네임스페이스/hello-node:v1
```



```
docker login ccr.ccs.tencentyun.com
docker push ccr.ccs.tencentyun.com/네임스페이스/hello-node:v1
```

설명

명령의 IMAGEID를 [이미지 조회](#)에서 기록된 IMAGEID로 바꿉니다.

명령의 네임스페이스를 생성한 네임스페이스로 바꿉니다. 네임스페이스가 없다면 먼저 네임스페이스를 생성하십시오. 자세한 내용은 [네임스페이스 생성](#)을 참고하십시오.

다음 정보가 나타나면 이미지가 성공적으로 업로드된 것입니다.

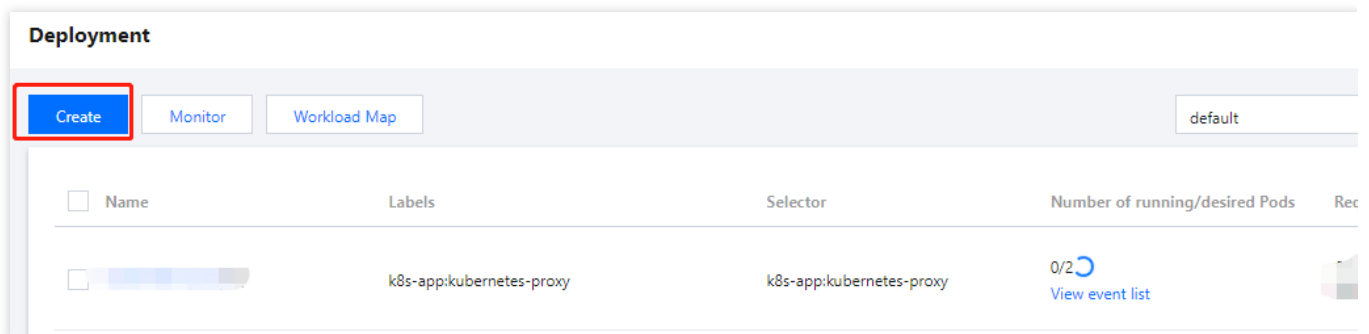
```
[root@VM_2_5_centos hellonode]# sudo docker tag ccr.ccs.tencentyun.com/
[root@VM_2_5_centos hellonode]# sudo docker push ccr.ccs.tencentyun.com/: test/hello
The push refers to repository [ccr.ccs.tencentyun.com/ test/helloworld]
7357e3a21b1f: Pushed
20a6f9d228c0: Pushed
80c332ac5101: Pushed
04dc8c446a38: Pushed
1050aff7cfff: Pushed
66d8e5ee400c: Pushed
2f71b45e4e25: Pushed
v1: digest: sha256:9c139ecbb29c49f25e02d7906b9e78c6e2e274827a75603ef48fa5547ff8a620 s
```

이 이미지를 사용하여 Hello World 서비스 생성

주의사항

Hello World 서비스를 생성 및 사용하기 전에 클러스터를 보유하고 있어야 합니다. 클러스터가 없다면 먼저 클러스터를 생성하십시오. 자세한 내용은 [클러스터 생성](#)을 참고하십시오.

1. TKE 콘솔에 로그인하고 왼쪽 사이드바에서 클러스터를 선택합니다.
2. 클러스터 관리 페이지에서 대상 클러스터의 ID를 클릭하여 서비스를 생성하고 클러스터 세부 정보 페이지로 이동합니다.
3. 워크로드 > Deployment를 선택하여 Deployment 페이지에서 다음 이미지와 같이 생성을 클릭합니다.



4. Deployment 생성 페이지에서 지시에 따라 워크로드의 기본 정보를 지정합니다. 아래 이미지를 참고하십시오.

The screenshot shows the configuration page for a new workload. The fields are as follows:

- Name:** helloworld
- Description:** helloworld serve
- Namespace:** default
- Labels:** k8s-app = helloworld
- OS type:** Linux
- Volume (optional):** Add volume

Below the form, there are several explanatory notes:

- Name:** Up to 63 characters, including lowercase letters, numbers, and hyphens ("-"). It must begin with a lowercase letter, and end with a number.
- Labels:** The key name cannot exceed 63 chars. It supports letters, numbers, "/" and "-". "/" cannot be placed at the beginning. A prefix is supported. The label key value can only include letters, numbers and separators ("-", "_", "."). It must start and end with letters and numbers.
- OS type:** Configurations are initialized when you change the OS type for the container.
- Volume:** It provides storage for the container. It can be a node path, cloud disk volume, file storage NFS, config file and PVC, and must be mounted.

Workload Name: 워크로드의 이름을 입력합니다. 이 예시에서 이름은 **helloworld**입니다.

Description: 관련 워크로드 정보를 지정합니다.

Namespace: 필요에 따라 네임스페이스를 선택합니다.

Tag: key = value 키 값 쌍을 지정합니다. 이 예시에서 기본값은 **k8s-app = helloworld**입니다.

OS 유형: 실제 필요에 따라 선택하십시오. 본문은 **Linux**를 예로 들어 설명합니다.

Volume: 요구 사항에 따라 마운트된 워크로드 볼륨을 설정합니다. 자세한 내용은 [Volume 관리](#)를 참고하십시오.

5. 지침에 따라 'Pod의 컨테이너'를 구성합니다.

5.1 컨테이너 이름을 입력합니다. 이 예시에서 이름은 **helloworld**입니다.

5.2 **이미지 선택**을 클릭하고 나타나는 대화 상자에서 **내 이미지**를 클릭합니다. 검색 상자를 사용하여 helloworld 이미지를 찾은 다음 **확인**을 클릭합니다.

주요 매개변수 정보는 다음과 같습니다.

Image Tag: 기본값인 **latest**를 사용합니다.

이미지 풀링 정책: 필요에 따라 사용 가능한 **Always, IfNotPresent** 및 **Never** 세 가지 정책 중 하나를 선택합니다. 이 예시에서는 기본 정책이 적용됩니다.

6. 'Number of Pod' 섹션에서 지침에 따라 서비스의 Pod 수를 설정합니다. 아래 이미지를 참고하십시오.

The screenshot shows the 'Number of instances' configuration section. It has two radio buttons: **Manual adjustment** (selected) and **Auto adjustment**. Below the radio buttons, it says 'Set the number of Pods directly'. There is a numeric input field with the value '1' and minus/plus buttons.

Manual adjustment: 포드 수를 지정합니다. 이 예에서는 1로 설정되어 있습니다. '+' 또는 '-'를 클릭하여 포드 수를 변경할 수 있습니다.

Auto Adjustment: 설정된 조건 중 하나라도 충족되면 자동으로 pod 수를 조정합니다. 자세한 내용은 [Automatic Scaling Basic Operation](#)을 참고하십시오.

7. 지침에 따라 워크로드에 대한 액세스 설정(서비스)을 구성합니다.

Service: '활성화'를 선택합니다.

서비스 액세스: '공중망 액세스'를 선택합니다.

로드 밸런서: 필요에 따라 로드 밸런서를 생성하거나 선택합니다.

포트 매핑: TCP 프로토콜을 선택하고 컨테이너 포트와 서비스 포트를 모두 포트 80으로 설정합니다.

주의사항

서비스가 속한 클러스터의 보안 그룹에 있는 노드 네트워크, 컨테이너 네트워크 및 포트 30000 - 32768은 인터넷에 개방되어 있어야 합니다. 그렇지 않으면 TKE를 사용하지 못할 수 있습니다. 자세한 내용은 [TKE 보안 그룹 설정](#)을 참고하십시오.

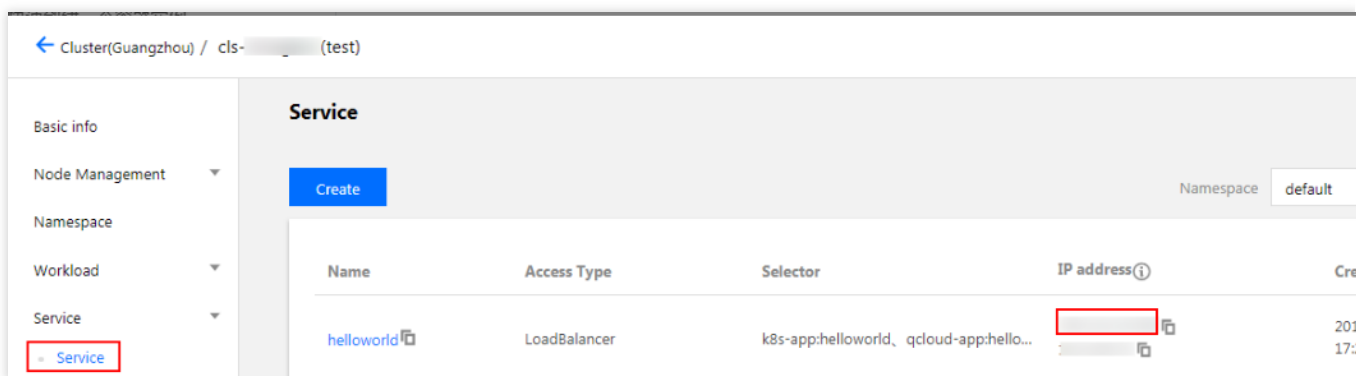
8. **Deployment 생성**을 클릭하여 Hello World 서비스를 생성합니다.

Hello World 서비스에 액세스

두 가지 방법 중 하나로 Hello World 서비스에 액세스할 수 있습니다.

CLB IP 주소로 Hello World 서비스에 액세스

1. [TKE 콘솔](#)에 로그인하고 왼쪽 사이드바에서 **클러스터**를 선택합니다.
2. **클러스터 관리** 페이지에서 Hello World 서비스가 있는 클러스터 ID를 클릭하여 **클러스터 세부 정보 페이지**로 이동합니다.
3. **Service** 페이지에서 **서비스 및 라우팅 > Service**를 선택합니다.
4. 서비스 관리 페이지에서 다음 이미지와 같이 helloworld 서비스의 CLB IP 주소를 복사합니다.



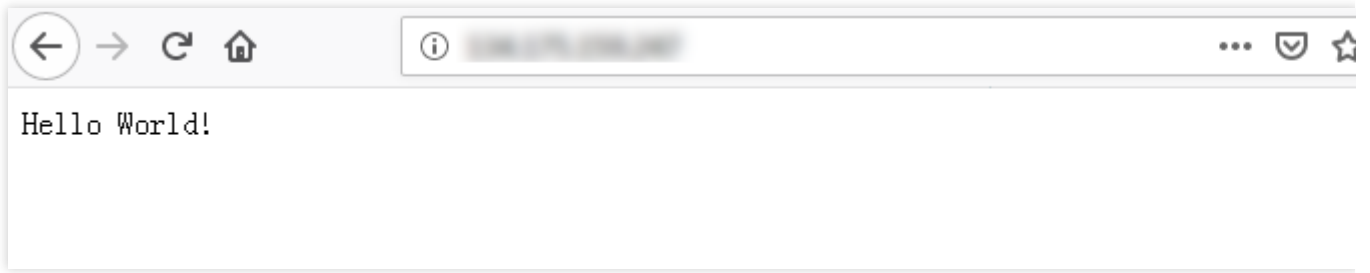
5. 브라우저에 helloworld 서비스의 CLB IP를 붙여넣습니다.

서비스 이름으로 Hello World 서비스에 액세스

클러스터의 다른 서비스 또는 컨테이너는 **서비스 이름**을 사용하여 직접 액세스할 수 있습니다.

Hello World 서비스 인증

서비스 액세스 시 아래와 같은 정보가 나오면 Hello World 서비스가 성공적으로 생성된 것입니다.



컨테이너를 생성할 수 없는 경우 솔루션에 대한 [이벤트 FAQ](#)를 검색할 수 있습니다.

단일 포드 WordPress

최종 업데이트 날짜: : 2022-04-20 10:33:01

작업 시나리오

WordPress는 PHP로 개발된 블로그 플랫폼입니다. 콘텐츠 관리 시스템으로 사용하거나 PHP 및 MySQL 데이터베이스를 지원하는 서비스에서 웹사이트를 만드는 데 사용할 수 있습니다.

본 문서에서는 Docker Hub의 공식 `wordpress` 이미지를 사용하여 공개적으로 액세스 가능한 WordPress 웹 사이트를 만드는 방법을 설명합니다.

전제 조건

주의 :

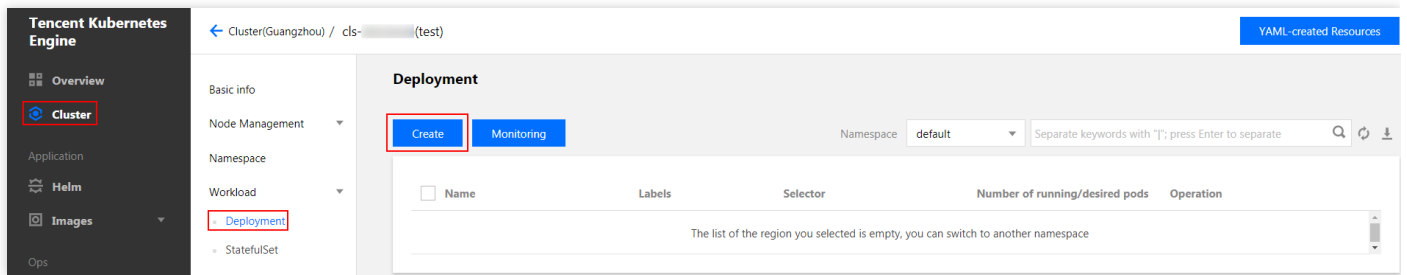
- `wordpress` 이미지에는 WordPress의 모든 운영 환경이 포함되어 있어 서비스를 직접 풀링해서 생성할 수 있습니다.
- 단일 Pod가 있는 WordPress는 테스트 목적으로만 사용되므로 영구 데이터 저장을 보장할 수 없습니다. 자체 구축한 MySQL 또는 TencentDB를 사용하여 데이터를 저장하는 것이 좋습니다. 자세한 내용은 [TencentDB를 사용하는 WordPress 서비스](#)를 참고하십시오.

- [Tencent Cloud 계정 등록](#)을 완료합니다.
- 클러스터를 생성했습니다. 클러스터 생성 방법에 대한 자세한 내용은 [Creating a cluster](#)를 참고하십시오.

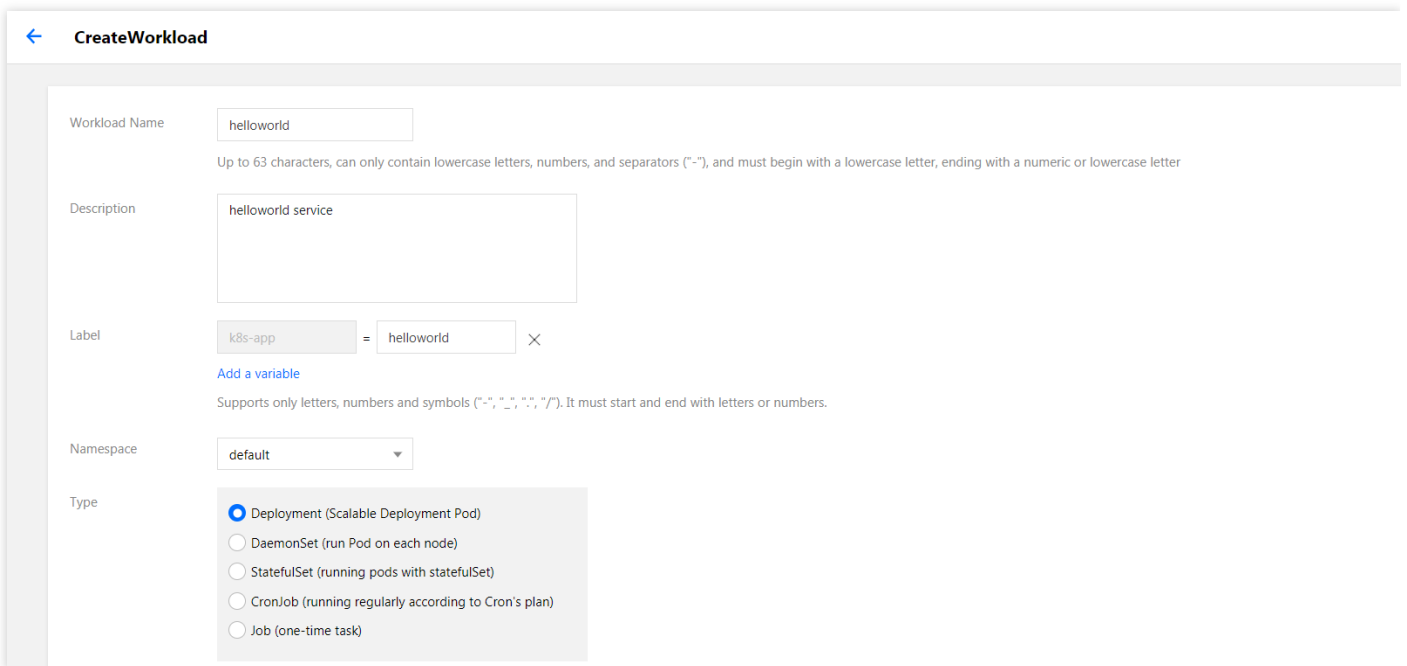
작업 단계

WordPress 서비스 생성

- TKE 콘솔에 로그인하고 왼쪽 사이드바에서 **Cluster**를 선택합니다.
- 'Cluster Management' 페이지에서 서비스를 생성할 클러스터의 ID를 클릭하여 아래 그림과 같이 클러스터의 'Deployment' 페이지로 이동한 후 **Create**를 클릭합니다.



3. 'Create Workload' 페이지에서 아래 그림과 같이 워크로드의 기본 정보를 지정합니다.



- **Workload Name:** 생성할 워크로드의 이름을 입력합니다. 이 예시에서는 wordpress를 사용합니다.
- **Description:** 관련 워크로드 정보를 지정합니다.
- **Tag:** key = value 키 값 쌍을 지정합니다. 기본값은 k8s-app = **wordpress**입니다.
- **Namespace:** 요구 사항에 따라 네임스페이스를 선택합니다.
- **Type:** 요구 사항에 따라 유형을 선택합니다.
- **Volume:** 요구 사항에 따라 워크로드가 탑재되는 볼륨을 설정합니다. 자세한 내용은 [Volume Management](#)를 참고하십시오.

4. 아래 지침에 따라 **Containers in the Pod**를 설정합니다.

주요 매개변수는 다음과 같이 설명됩니다. 다른 매개변수의 기본값을 유지할 수 있습니다.

- **Name:** 사용자 정의 컨테이너 이름을 입력합니다. 이 예시에서는 test가 사용됩니다.
- **Image:** `wordpress` 를 입력합니다.
- **Image Tag:** latest를 입력합니다.

- **Image Pull Policy:** 세 가지 정책을 사용할 수 있습니다. 필요에 따라 정책을 선택합니다. 본 예시에서는 기본 정책이 사용됩니다.

이미지 가져오기 정책을 설정하지 않고 이미지 태그가 비어 있거나 `latest` 으로 설정된 경우 **Always** 정책이 사용됩니다. 그렇지 않으면 **IfNotPresent** 정책이 사용됩니다.

- **Always:** 항상 원격 끝에서 이미지를 가져옵니다.
- **IfNotPresent:** 기본적으로 로컬 이미지가 사용됩니다. 사용 가능한 로컬 이미지가 없으면 이미지를 원격으로 가져옵니다.
- **Never:** 로컬 이미지만 사용합니다. 사용 가능한 로컬 이미지가 없으면 예외가 발생합니다.

6. 지시에 따라 서비스의 Pod 수를 설정합니다. 아래 이미지를 참고하십시오.

The screenshot shows a configuration panel for 'Number of Pods'. It has two radio button options: 'Manual adjustment' (selected) and 'Auto adjustment'. Under 'Manual adjustment', there is a 'Number of Pods' input field with a value of '1' and minus/plus buttons. Under 'Auto adjustment', there is a description: 'Automatically adjust the number of pods if any of the setting conditions are met' and a 'View more' link.

- **Manual adjustment:** 포드 수를 설정합니다. 이 예시에서 포드 수는 1로 설정되어 있습니다. '+' 또는 '-'를 클릭하여 포드 수를 변경할 수 있습니다.
- **Auto Adjustment:** 설정 조건 중 하나라도 충족되면 pod 수가 자동으로 조정됩니다. 자세한 내용은 [Automatic Scaling Basic Operation](#) 기본 작업을 참고하십시오.

8. 아래 지침에 따라 워크로드에 대한 ****Access Setting (Service)****를 구성합니다.

- **Service:** 활성화를 선택합니다.
- **Service Access:** 'LoadBalancer(public network)'를 선택합니다.
- **Load Balancer:** 요구 사항에 따라 선택합니다.
- **Port Mapping:** TCP를 선택하고 컨테이너 포트와 서비스 포트를 모두 80으로 설정합니다.

주의 :

서비스가 속한 클러스터의 보안 그룹은 노드 네트워크와 컨테이너 네트워크를 인터넷에 개방해야 합니다. 또한 인터넷에 포트 30000 - 32768을 개방해야 합니다. 그렇지 않으면 TKE를 사용하지 못할 수 있습니다. 자세한 내용은 [TKE 보안 그룹 설정](#)을 참고하십시오.

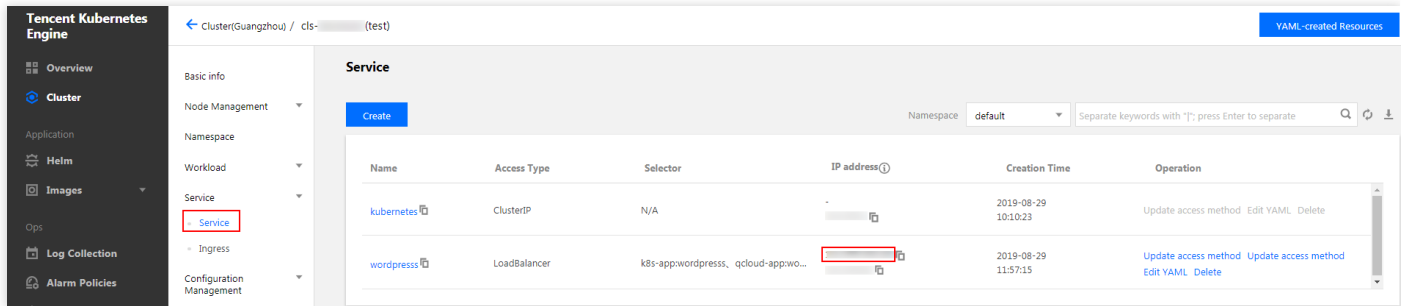
9. **Create Workload**을 클릭하여 WordPress 서비스를 생성합니다.

WordPress 서비스에 액세스

다음 두 가지 방법 중 하나를 사용하여 WordPress 서비스에 액세스할 수 있습니다.

CLB IP 주소를 사용하여 액세스

1. 왼쪽 사이드바에서 **Cluster**를 클릭하여 'Cluster Management' 페이지로 이동합니다.
2. WordPress 서비스가 속한 클러스터의 ID를 클릭하고 **Service and Route > Service**를 선택합니다.
3. 서비스 관리 페이지에서 아래 그림과 같이 WordPress 서비스의 CLB IP 주소를 복사합니다.



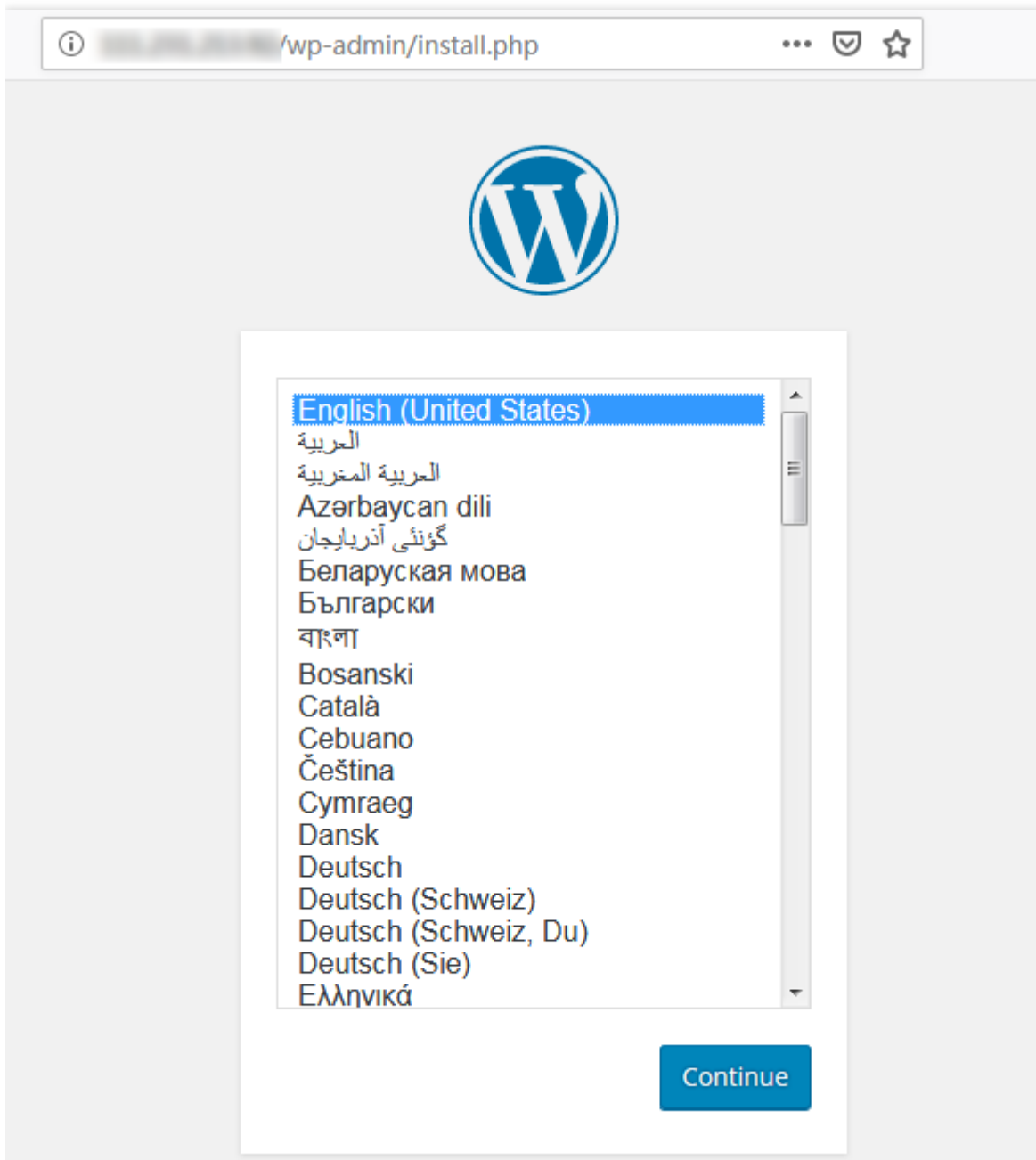
4. 브라우저에 CLB IP 주소를 붙여넣고 'Enter' 키를 눌러 서비스에 액세스합니다.

서비스 이름을 사용하여 WordPress 서비스에 액세스

클러스터의 다른 서비스 또는 컨테이너는 서비스 이름을 사용하여 WordPress 서비스에 액세스할 수 있습니다.

WordPress 서비스 확인

서비스 생성 후 서비스에 접속하면 아래 그림과 같이 WordPress 서버 설정 페이지가 나타납니다.



더 많은 WordPress 설정

컨테이너 생성에 실패한 경우 [Event FAQ](#)를 참고하십시오.

TencentDB를 사용하는 WordPress 서비스

최종 업데이트 날짜: : 2022-04-20 10:33:01

작업 시나리오

WordPress 서비스를 빠르게 만드는 방법에 대해 알아보려면 [단일 포트 WordPress](#)를 참고하십시오. 이러한 방식으로 생성된 WordPress 서비스에는 다음과 같은 기능이 있습니다.

- 데이터는 동일한 컨테이너에서 실행되는 MySQL 데이터베이스에 기록됩니다.
- 서비스를 빠르게 실행할 수 있습니다.
- 특정 이유로 컨테이너가 중지되면 데이터베이스 및 스토리지 유형 파일이 손실됩니다.

MySQL 데이터베이스를 사용하면 데이터를 영구적으로 저장할 수 있습니다. 데이터베이스는 포트/컨테이너가 다시 시작될 때 계속 실행됩니다. 본 문서에서는 [TencentDB](#)를 사용하여 MySQL 데이터베이스를 구성하는 방법과 TencentDB를 사용하는 WordPress 서비스를 만드는 방법을 설명합니다.

전제 조건

- [Tencent Cloud 계정 등록](#)을 완료합니다.
- 클러스터를 생성 완료해야 합니다. 클러스터 생성 방법에 대한 자세한 내용은 [Creating a Cluster](#)를 참고하십시오.

설명 :

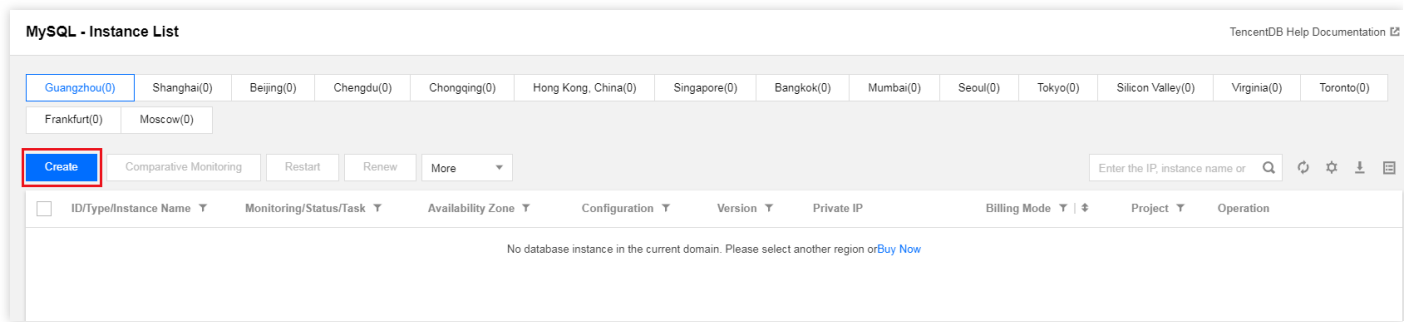
문서에 사용된 데이터베이스는 [TencentDB for MySQL](#)입니다.

작업 단계

WordPress 서비스 생성

TencentDB 인스턴스 생성

1. [TencentDB for MySQL 콘솔](#)에 로그인하고 아래 이미지와 같이 데이터베이스 인스턴스 목록에서 [Create]를 클릭합니다.



2. 구매할 구성을 선택합니다. 자세한 내용은 [TencentDB for MySQL](#)을 참고하십시오.

주의 :

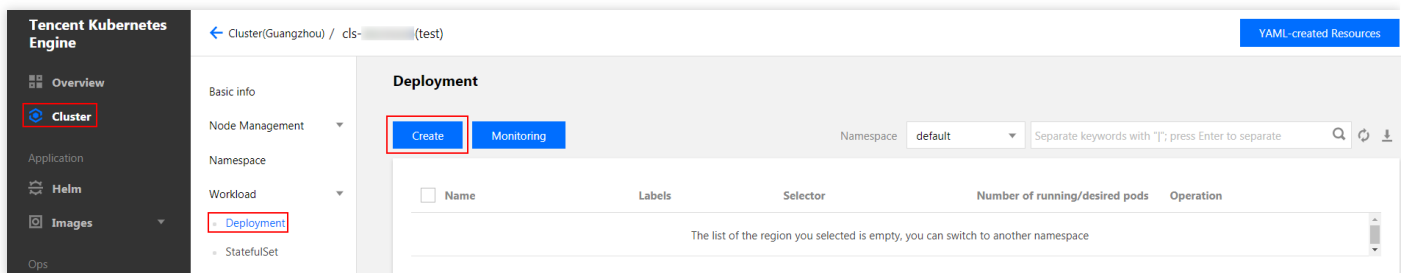
데이터베이스는 클러스터와 동일한 지역에 있어야 합니다. 그렇지 않으면 데이터베이스에 연결할 수 없습니다.

3. 데이터베이스를 생성한 후 [MySQL instance list](#)에서 볼 수 있습니다.

4. 데이터베이스를 초기화합니다. 자세한 내용은 [MySQL 인스턴스 초기화](#)를 참고하십시오.

Tencent DB를 사용하는 WordPress 서비스 생성

1. TKE 콘솔에 로그인하고 왼쪽 사이드바에서 [\[Cluster\]](#)를 선택합니다.
2. 'Cluster Management' 페이지에서 서비스가 생성될 클러스터의 ID를 클릭하여 아래 이미지와 같이 클러스터의 'Deployment' 페이지로 이동한 후 [\[Create\]](#)를 클릭합니다.



3. 'Create Workload' 페이지에서 아래 이미지와 같이 워크로드의 기본 정보를 지정합니다.

The screenshot shows the 'CreateWorkload' form with the following fields and values:

- Workload Name:** wordpress
- Description:** Wordpress TencentDB version
- Label:** k8s-app = wordpress
- Namespace:** default
- Type:** Deployment (Scalable Deployment Pod) (selected)

- **Workload Name:** 생성할 워크로드의 이름입니다. 여기서는 wordpress를 예시로 사용합니다.
- **Description:** 관련 워크로드 정보를 지정합니다.
- **Tag:** key = value 키 값 쌍을 지정합니다. 기본값은 k8s-app = **wordpress**입니다.
- **Namespace:** 요구 사항에 따라 네임스페이스를 선택합니다.
- **Type:** 요구 사항에 따라 유형을 선택합니다.
- **Volume:** 요구 사항에 따라 워크로드가 탑재되는 볼륨을 설정합니다. 자세한 내용은 [Volume Management](#)를 참고하십시오.

4. 아래 그림의 지침에 따라 'Containers in the pod'를 설정합니다.

Containers in the pod

Name: Up to 63 characters. It supports lower case letters, number, and hyphen ("-") and cannot start or end with ("-")

Image: [Select an image](#)

Image Tag:

CPU/memory limit

CPU Limit: - Memory Limit: -

Request is used to pre-allocate resources. When the nodes in the cluster do not have the required number of resources, the container will fail to be created. Limit specifies the maximum usage of resources of container to avoid abnormal excessive consumption of node resources.

GPU limit:

Environment Variable

=

=

[Add a variable Reference ConfigMap/Secret](#)

Supports only letters, numbers and symbols ("-", "_", "."). It must start and end with letters or numbers.

[Advanced Settings](#)

Note: After Workload is created, the container configuration information can be modified by updating YAML.

[Add Container](#)

주요 매개변수는 다음과 같습니다. 다른 매개변수에 대한 기본 설정을 유지합니다.

- **Name:** 사용자 정의 컨테이너 이름을 입력합니다. 이 예시에서는 `test`가 사용됩니다.
- **Image:** `wordpress` 를 입력하십시오.
- **Image Tag:** `latest`를 입력하십시오.
- **Image Pull Policy:** 세 가지 정책을 사용할 수 있습니다. 필요에 따라 정책을 선택합니다. 이 예시에서는 기본 정책이 사용됩니다.
이미지 가져오기 정책을 지정하지 않으면 이미지 태그가 비어 있거나 `latest`일 때 `Always` 정책이 사용됩니다. 그렇지 않으면 `IfNotPresent` 정책이 사용됩니다.
 - **Always:** 이미지는 항상 원격 위치에서 가져옵니다.
 - **IfNotPresent:** 기본적으로 로컬 이미지를 사용합니다. 이미지를 사용할 수 없는 경우 원격 위치에서 가져옵니다.
 - **Never:** 로컬 이미지만 사용합니다. 이미지를 사용할 수 없으면 예외가 발생합니다.
- **Environment variable:** 다음 설정 정보를 순서대로 입력합니다.
`WORDPRESS_DB_HOST` = TencentDB for MySQL의 내부 IP 주소
`WORDPRESS_DB_PASSWORD` = 초기화 시 입력한 비밀번호

5. 아래 이미지와 같이 서비스의 Pod 수를 설정합니다.

Number of Pods ⓘ

Manual adjustment Set the number of pods directly

Number of Pods 1

Auto adjustment Automatically adjust the number of pods if any of the setting conditions are met [View more](#) ⓘ

- **Manual adjustment:** 포드 수를 설정합니다. 이 예시에서 포드 수는 1로 설정되어 있습니다. ‘+’ 또는 ‘-’를 클릭하여 포드 수를 변경할 수 있습니다.
- **Auto Adjustment:** 설정 조건 중 하나라도 충족되면 pod 수가 자동으로 조정됩니다. 자세한 내용은 [Automatic Scaling Basic Operations](#)를 참고하십시오.

6. 아래 그림의 지침에 따라 워크로드에 대한 액세스 설정(Service)을 구성합니다.

Access Settings (Service)

Service Enable

Service Access Via Internet Intra-cluster Via VPC Node Port Access [How to select](#) ⓘ

Automatically create a classic public CLB (0.02 CNY/hour) to provide Internet access. It supports TCP/UDP protocol. Public network access is applicable to web front-end services.

If you need forward via internet using HTTP/HTTPS protocols or by URL, you can go to Ingress page to configure Ingress for routing. [Learn More](#) ⓘ

Load Balancer

Automatically create a traditional CLB for public/internet access service. Do not manually modify the CLB listener created by TKE. [Learn more](#) ⓘ

Port Mapping

Protocol ⓘ	Target Port ⓘ	Port ⓘ	
TCP	80	80	×

[Add Port Mapping](#)

[Advanced Settings](#)

- **Service:** ‘Enable’을 선택합니다.
- **Service Access:** ‘Via Internet’를 선택합니다.
- **Load Balancer:** 필요에 따라 로드 밸런서를 생성하거나 선택합니다.
- **Port Mapping:** TCP를 선택하고 컨테이너 포트와 서비스 포트를 모두 80으로 설정합니다.

주의 :

서비스가 속한 클러스터의 보안 그룹은 노드 네트워크와 컨테이너 네트워크를 인터넷에 개방해야 합니다. 또한 인터넷에 포트 30000-32768을 개방해야 합니다. 그렇지 않으면 TKE를 사용하지 못할 수 있습니다. 자세한 내용은 [TKE 보안 그룹 설정](#)을 참고하십시오.

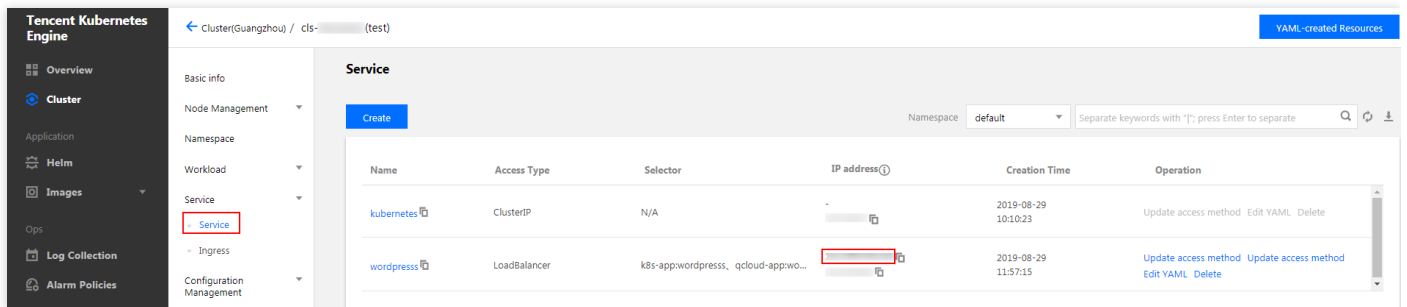
7. [Create workload]을 클릭합니다.

WordPress 서비스에 액세스

다음 두 가지 방법 중 하나를 사용하여 WordPress 서비스에 액세스할 수 있습니다.

CLB IP 주소를 사용하여 WordPress 서비스에 액세스

1. 왼쪽 사이드바에서 [Cluster]를 클릭하여 'Cluster Management' 페이지로 이동합니다.
2. WordPress 서비스가 속한 클러스터의 ID를 클릭하고 [Service]>[Service]를 선택합니다.
3. 서비스 관리 페이지에서 아래 이미지와 같이 WordPress 서비스의 CLB IP 주소를 복사합니다.



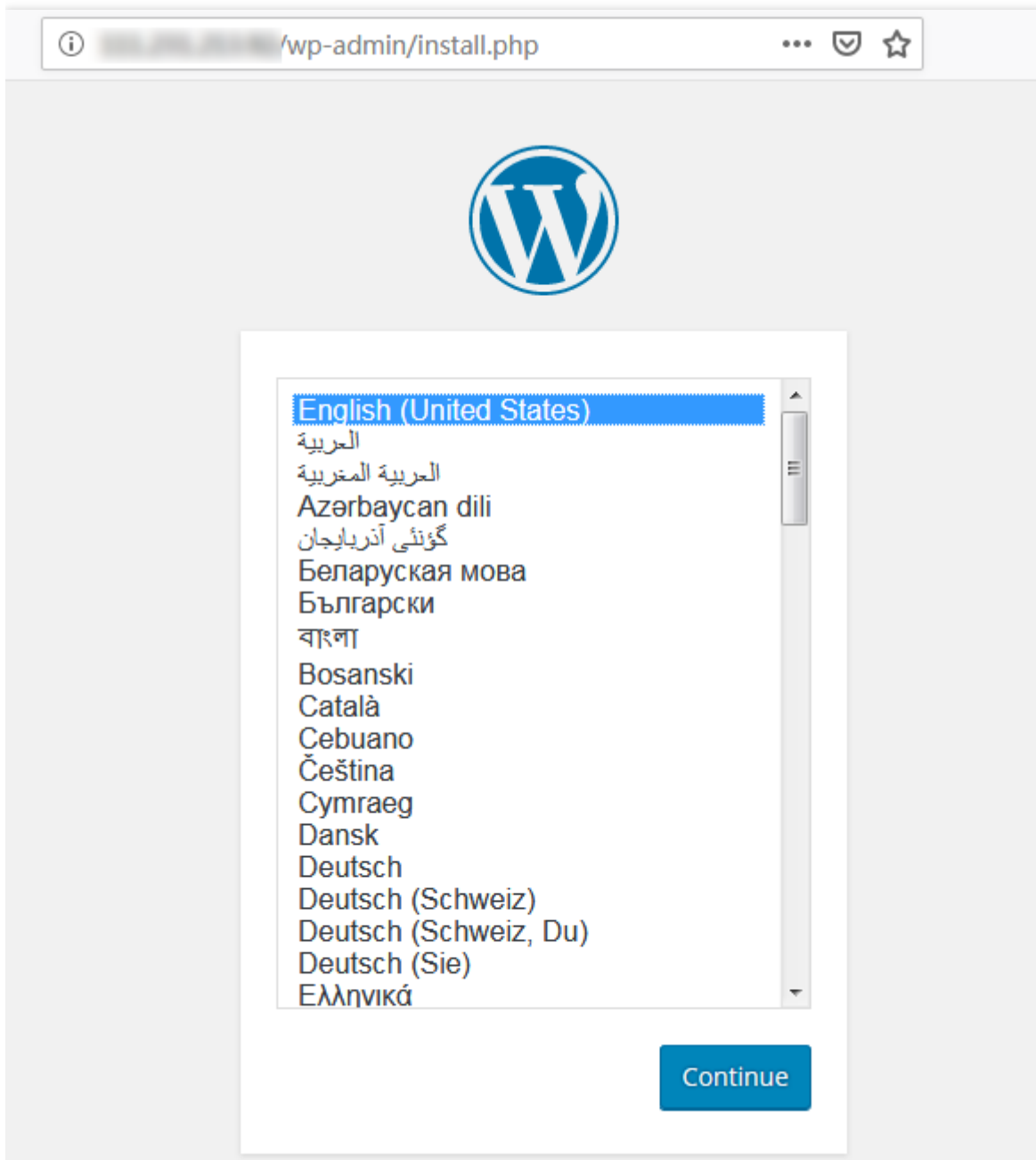
4. 브라우저 주소 상자에 CLB IP 주소를 붙여넣고 **Enter** 키를 눌러 서비스에 액세스합니다.

서비스 이름을 사용하여 WordPress 서비스에 액세스

클러스터의 다른 서비스 또는 컨테이너는 서비스 이름을 사용하여 WordPress 서비스에 액세스할 수 있습니다.

WordPress 서비스 인증

서비스 생성 후 서비스에 접속하면 아래 이미지와 같이 WordPress 서버 설정 페이지가 나타납니다.



더 많은 WordPress 설정

컨테이너 생성에 실패한 경우 [Event FAQ](#)를 참고하십시오.

간단한 Web 애플리케이션 구축

최종 업데이트 날짜: : 2022-12-14 16:43:41

작업 시나리오

본 문서에서는 TKE를 통해 간단한 Web 애플리케이션을 만드는 방법을 설명합니다.

Web 애플리케이션은 다음 부분으로 나뉩니다.

- 첫 번째 부분은 클라이언트의 쿼리 및 쓰기 요청을 처리하는 데 사용되는 프런트 엔드 서비스입니다.
- 다른 부분은 Redis를 사용하여 작성된 데이터를 redis-master에 저장하고 redis-slave에서 데이터를 읽는 데이터 저장 서비스입니다. 프라이머리/세컨더리 복제를 통해 redis-master와 redis-slave 간에 데이터를 동기화합니다.

이 애플리케이션은 Kubernetes 프로젝트와 함께 제공되는 샘플 애플리케이션입니다. 자세한 내용은 [Guestbook App](#)을 참고하십시오.

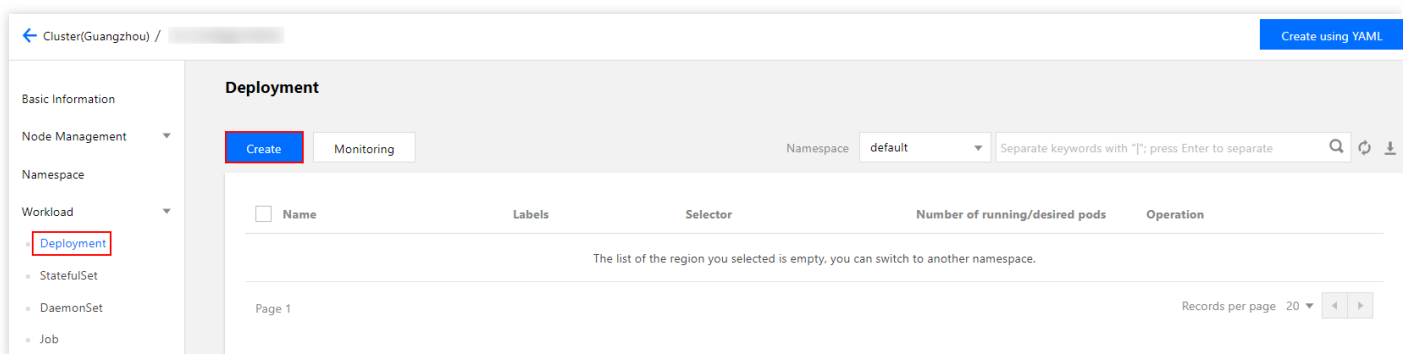
전제 조건

- [Tencent Cloud 계정 등록](#)을 완료합니다.
- 클러스터를 생성 완료해야 합니다. 자세한 내용은 [Creating a Cluster](#)를 참고하십시오.

작업 단계

redis-master 서비스 생성

1. TKE 콘솔에 로그인하고 왼쪽 사이드바에서 [\[Clusters\]](#)를 선택합니다.
2. 애플리케이션을 생성할 클러스터 ID를 클릭하고 Deployment 페이지로 이동합니다. [\[Create\]](#)를 선택합니다. 아래 그림을 참고하십시오.



3. 다음 지침에 따라 'Create Workload' 페이지에서 워크로드 기본 정보를 설정합니다. 아래 그림을 참고하십시오.

The screenshot shows the 'Create Workload' interface with the following details:

- Workload Name:** redis-master (Up to 40 chars, supports only lower-case chars, numbers and hyphens. It must start with a lower-case letter and end with a number or lower-case letter.)
- Description:** Up to 1000 characters
- Tag:** k8s-app = redis-master (Add a variable)
- Namespace:** default (Supports only letters, numbers and symbols ("-", "_", ".", "/", "/"). It must start and end with letters or numbers.)
- Type:**
 - Deployment (Scalable Deployment Pod)
 - DaemonSet (run Pod on each node)
 - StatefulSet (running pods with statefulSet)
 - CronJob (running regularly according to Cron's plan)
 - Job (one-time task)
- Volume (optional):** Add Volume (Provides storage for the container. It can be a node path, cloud disk volume, file storage NFS, config file and PVC, and must be mounted to the specified path of the container. [Instruction](#))

- **Workload Name:** 생성할 워크로드의 이름입니다. 이 예시에서는 redis-master로 설정되어 있습니다.
- **Description:** 워크로드와 관련된 정보를 입력합니다.
- **Label:** key = value 형식의 키 값 쌍입니다. 이 예시에서 기본 레이블은 k8s-app = **redis-master**입니다.
- **Namespace:** 필요에 따라 값을 선택합니다.
- **Type:** 필요에 따라 값을 선택합니다.
- **Volume:** 워크로드가 탑재될 볼륨을 선택합니다. 자세한 내용은 [Volume Management](#)를 참고하십시오.

4. 다음 지침에 따라 **Containers in Pod**를 설정합니다. 아래 그림을 참고하십시오.

Containers in the pod
✓ ✕

Name

Up to 63 characters. It supports lower case letters, number, and hyphen ("-") and cannot start or end with ("-")

Image [Select an image](#)

Image Tag

Pull Image from Remote Registry

If the image pull policy is not set, when the image tag is empty or "latest", the "Always" policy is used, otherwise "IfNotPresent" is used.

CPU/memory limit

CPU Limit

-

core

Memory Limit

-

MiB

Request is used to pre-allocate resources. When the nodes in the cluster do not have the required number of resources, the container will fail to be created.
Limit is used to set a upper limit for resource usage for a container, so as to avoid over usage of node resources in case of exceptions.

GPU Resource

Configure the minimum GPU resource usage of this workload. Please make sure that the cluster has enough GPU resource.

Environment Variable [Add a variable](#) [Reference ConfigMap/Secret](#)

Supports only letters, numbers and symbols ("-", "_", "."). It must start with a letter.

[Advanced Settings](#)

[Add Container](#)

Note: After Workload is created, the container configuration information can be modified by updating YAML.

다음은 주요 매개변수를 설명합니다. 다른 매개변수의 경우 기본값으로 그대로 두십시오.

- **Name:** 포드에 있는 컨테이너의 이름을 입력합니다. 이 예시에서 이름은 **master**입니다.
- **Image:** `ccr.ccs.tencentyun.com/library/redis` 를 입력합니다.
- **Image Tag:** `latest`를 입력하십시오.
- **Image Pull Policy:** 세 가지 정책을 사용할 수 있습니다. 필요에 따라 값을 선택합니다. 이 예에서는 이 필드를 지정할 필요가 없으며 기본 정책을 사용하기만 하면 됩니다.
이미지 풀 정책을 지정하지 않으면 이미지 태그가 비어 있거나 `latest`인 경우 **Always** 정책이 사용됩니다. 그렇지 않으면 **IfNotPresent** 정책이 사용됩니다.
 - **Always:** 항상 원격 끝에서 이미지를 가져옵니다.
 - **IfNotPresent:** 기본적으로 로컬 이미지를 사용합니다. 로컬 이미지를 사용할 수 없는 경우 원격 끝에서 이미지를 가져옵니다.

- **Never:** 로컬 이미지만 사용합니다. 로컬 이미지를 사용할 수 없으면 예외가 발생합니다.

6. 포드 수를 설정합니다.

- **Manual adjustment:** 포드 수를 설정합니다. 이 예시에서 포드 수는 1입니다. '+' 또는 '-'를 클릭하여 포드 수를 조정할 수 있습니다.
- **Auto adjustment:** 지정된 조건이 충족되면 시스템이 자동으로 pod 수를 조정합니다. 자세한 내용은 [Auto Scaling](#)을 참고하십시오.

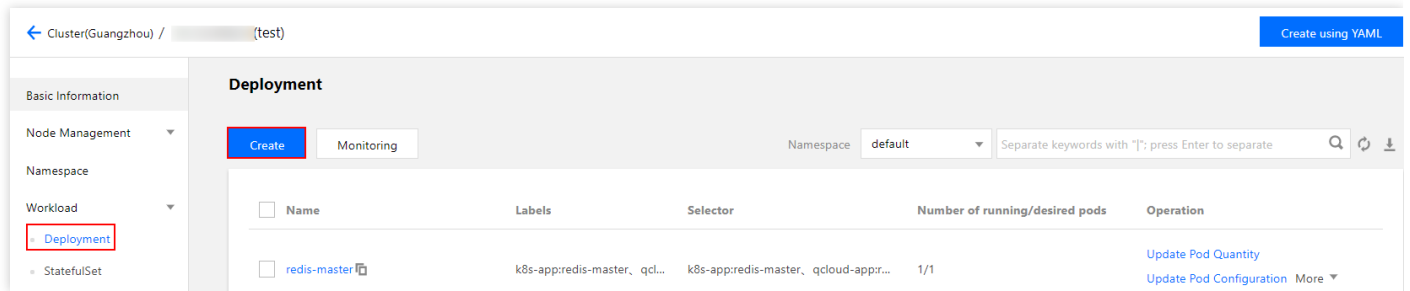
7. 다음 이미지와 같이 워크로드의 액세스 모드를 지정합니다.

- **Service:** 'Enable'을 선택합니다.
- **Service Acces:** 'Intra-cluster'를 선택합니다.
- **Load Balancer:** 필요에 따라 값을 선택합니다.
- **Port Mapping:** 프로토콜 드롭다운 목록에서 TCP를 선택하고 포트와 대상 포트를 모두 6379로 설정합니다. 이러한 방식으로 다른 서비스는 redis-master 서비스 이름과 6379 포트 번호를 사용하여 master 컨테이너에 액세스할 수 있습니다.

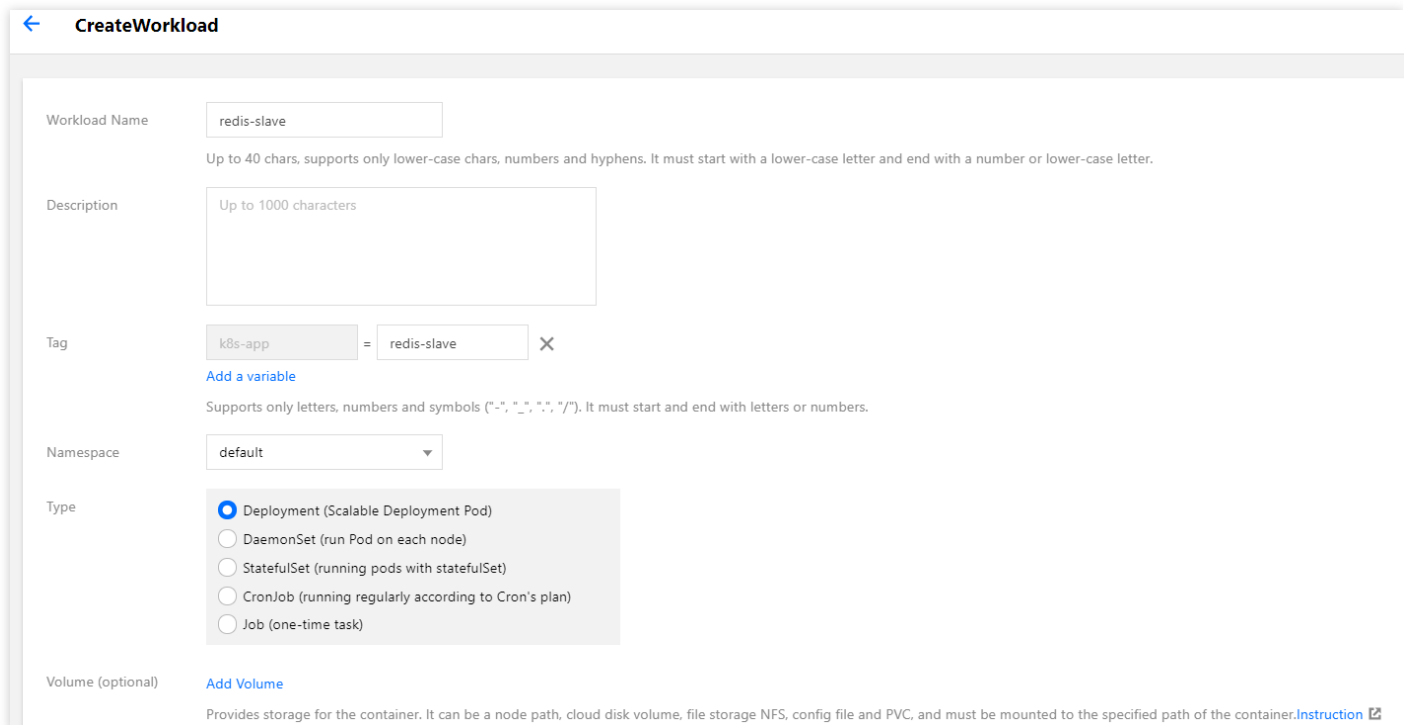
8. [Create Workload]를 클릭하여 redis-master 서비스 생성을 마칩니다.

redis-slave 서비스 생성

1. TKE 콘솔에 로그인하고 왼쪽 사이드바에서 [Clusters]를 선택합니다.
2. 서비스를 생성할 클러스터의 ID를 클릭합니다. 표시되는 Deployment 페이지에서 다음 이미지와 같이 [Create]를 클릭합니다.



3. 표시되는 'New Workload' 페이지에서 지침에 따라 워크로드의 기본 정보를 지정합니다.



- **Workload Name:** 생성할 워크로드의 이름을 나타냅니다. 이 예시에서 이름은 redis-slave입니다.
- **Description:** 워크로드와 관련된 정보를 입력합니다.
- **Label:** key = value 형식의 키 값 쌍을 나타냅니다. 이 예시에서 기본 레이블은 k8s-app = **redis-slave**입니다.
- **Namespace:** 필요에 따라 값을 선택합니다.
- **Type:** 필요에 따라 값을 선택합니다.
- **Volume:** 워크로드가 탑재될 볼륨을 선택합니다. 자세한 내용은 [Volume Management](#)를 참고하십시오.

4. 다음 이미지와 같이 **Containers in pod** 설정을 구성합니다.

Containers in the pod
✓ ✕

Name

Up to 63 characters. It supports lower case letters, number, and hyphen ("-") and cannot start or end with ("-")

Image [Select an image](#)

Image Tag

Pull Image from Remote Registry Always IfNotPresent Never

If the image pull policy is not set, when the image tag is empty or ":latest", the "Always" policy is used, otherwise "IfNotPresent" is used.

CPU/memory limit

CPU Limit

request

0.25

-

limit

0.5

core

Memory Limit

request

256

-

limit

1024

MiB

Request is used to pre-allocate resources. When the nodes in the cluster do not have the required number of resources, the container will fail to be created.
Limit is used to set a upper limit for resource usage for a container, so as to avoid over usage of node resources in case of exceptions.

GPU Resource − +

Configure the minimum GPU resource usage of this workload. Please make sure that the cluster has enough GPU resource.

Environment Variable ⓘ

GET_HOSTS_FR

=

dns

✕

Add a variable [Reference ConfigMap/Secret](#)

Supports only letters, numbers and symbols ("-", "_", "."). It must start with a letter.

[Advanced Settings](#)

[Add Container](#)

Note: After Workload is created, the container configuration information can be modified by updating YAML.

다음은 주요 매개변수를 설명합니다. 다른 매개변수의 경우 기본 값으로 그대로 두십시오.

- **Name:** 포드에 있는 컨테이너의 이름을 입력합니다. 이 예시에서 이름은 `slave`입니다.
- **Image:** `ccr.ccs.tencentyun.com/library/gb-redisslave` 를 입력합니다.
- **Image Tag:** `latest`를 입력하십시오.
- **Image Pull Policy:** 필요에 따라 값을 선택합니다. 이 예시에서는 이 필드를 지정할 필요가 없으며 기본 정책을 사용하기만 하면 됩니다.
- **Environment Variable:** 다음 설정 정보를 입력합니다.
`GET_HOSTS_FROM = dns`

5. 포드 수를 설정합니다.

Number of Pods Manual adjustment Auto adjustment
Set the number of pods directly

Number of Pods 1

[Advanced Settings](#)

- **Manual adjustment:** 포드 수를 설정합니다. 이 예시에서 포드 수는 1입니다. '+' 또는 '-'를 클릭하여 포드 수를 조정할 수 있습니다.
- **Auto adjustment:** 지정된 조건이 충족되면 시스템이 자동으로 pod 수를 조정합니다. 자세한 내용은 [Auto Scaling](#)을 참고하십시오.

6. 다음 이미지와 같이 워크로드의 액세스 모드를 지정합니다.

Access Settings (Service)

Service Enable

Service Access Via Internet Intra-cluster Via VPC Node Port Access [How to select](#)

Provides an entrance for accesses from other services and containers in the same cluster. It supports TCP/UDP protocols, and is applicable to database services (such as MySQL)

Headless Service [?](#) (Headless Service can only be selected upon creation. **The access method cannot be changed after the creation.**)

Protocol	Target Port	Port
TCP	6379	6379

[Add Port Mapping](#)

[Advanced Settings](#)

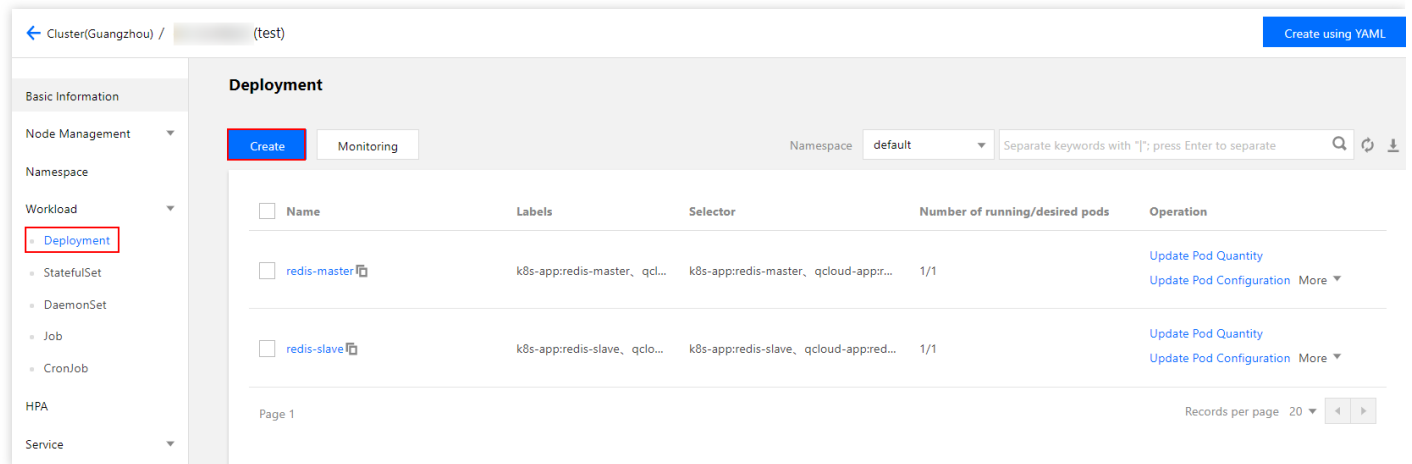
- **Service:** 'Enable'을 선택합니다.
- **Service Access:** 'Intra-cluster'를 선택합니다.
- **Load Balancer:** 필요에 따라 값을 선택합니다.
- **Port Mapping:** 프로토콜 드롭 다운 목록에서 TCP를 선택하고 포트와 대상 포트를 모두 6379로 설정합니다. 이러한 방식으로 다른 서비스는 redis-master 서비스 이름과 6379 포트 번호를 사용하여 master 컨테이너에 액세스할 수 있습니다.

7. [Create Workload]을 클릭하여 redis-slave 서비스 생성을 마칩니다.

frontend 서비스 생성

1. TKE 콘솔에 로그인하고 왼쪽 사이드바에서 [\[Clusters\]](#)를 선택합니다.

2. 애플리케이션을 생성할 클러스터의 ID를 클릭합니다. 표시되는 Deployment 페이지에서 다음 이미지와 같이 [Create]를 클릭합니다.



3. 표시되는 'New Workload' 페이지에서 지침에 따라 워크로드의 기본 정보를 지정합니다.

Workload Name: frontend
Up to 40 chars, supports only lower-case chars, numbers and hyphens. It must start with a lower-case letter and end with a number or lower-case letter.

Description: Up to 1000 characters

Tag: k8s-app = frontend X
Add a variable
Supports only letters, numbers and symbols ("-", "_", ".", "/"). It must start and end with letters or numbers.

Namespace: default

Type:

- Deployment (Scalable Deployment Pod)
- DaemonSet (run Pod on each node)
- StatefulSet (running pods with statefulSet)
- CronJob (running regularly according to Cron's plan)
- Job (one-time task)

Volume (optional): Add Volume
Provides storage for the container. It can be a node path, cloud disk volume, file storage NFS, config file and PVC, and must be mounted to the specified path of the container. [Instruction](#)

- **Workload Name:** 생성할 워크로드의 이름입니다. 이 예시에서 이름은 frontend입니다.
- **Description:** 작업 부하와 관련된 정보를 입력합니다.
- **Tag:** key = value 형식의 키 값 쌍. 이 예시에서는 k8s-app = frontend를 사용합니다.
- **Namespace:** 필요에 따라 값을 선택합니다.
- **Type:** 필요에 따라 값을 선택합니다.
- **Volume:** 워크로드가 탑재될 볼륨을 선택합니다. 자세한 내용은 [Volume Management](#)를 참고하십시오.

5. 다음 이미지와 같이 **Containers in pod** 설정을 구성합니다.

Containers in the pod
✓ ✕

Name

Up to 63 characters. It supports lower case letters, number, and hyphen ("-") and cannot start or end with ("-")

Image [Select an image](#)

Image Tag

Pull Image from Remote Registry Always IfNotPresent Never

If the image pull policy is not set, when the image tag is empty or "latest", the "Always" policy is used, otherwise "IfNotPresent" is used.

CPU/memory limit

<p>CPU Limit</p> <div style="display: flex; align-items: center; gap: 5px;"> <div style="border: 1px solid #ccc; padding: 2px 5px;">request</div> <div style="border: 1px solid #ccc; padding: 2px 5px;">0.25</div> <div style="font-size: x-small;">-</div> <div style="border: 1px solid #ccc; padding: 2px 5px;">limit</div> <div style="border: 1px solid #ccc; padding: 2px 5px;">0.5</div> </div> <p style="font-size: x-small;">core</p>	<p>Memory Limit</p> <div style="display: flex; align-items: center; gap: 5px;"> <div style="border: 1px solid #ccc; padding: 2px 5px;">request</div> <div style="border: 1px solid #ccc; padding: 2px 5px;">256</div> <div style="font-size: x-small;">-</div> <div style="border: 1px solid #ccc; padding: 2px 5px;">limit</div> <div style="border: 1px solid #ccc; padding: 2px 5px;">1024</div> </div> <p style="font-size: x-small;">MiB</p>
--	--

Request is used to pre-allocate resources. When the nodes in the cluster do not have the required number of resources, the container will fail to be created.
Limit is used to set a upper limit for resource usage for a container, so as to avoid over usage of node resources in case of exceptions.

GPU Resource − +

Configure the minimum GPU resource usage of this workload. Please make sure that the cluster has enough GPU resource.

Environment Variable ⓘ

= ✕

[Add a variable Reference ConfigMap/Secret](#)

Supports only letters, numbers and symbols ("-", "_", "."). It must start with a letter.

[Advanced Settings](#)

[Add Container](#)

Note: After Workload is created, the container configuration information can be modified by updating YAML.

다음은 주요 매개변수를 설명합니다. 다른 매개변수의 경우 기본 값으로 그대로 두십시오.

- **Name:** 포드에 있는 컨테이너의 이름을 입력합니다. 이 예시에서 이름은 frontend입니다.
- **Image:** `ccr.ccs.tencentyun.com/library/gb-frontend` 를 입력합니다.
- **Image Tag:** latest를 입력하십시오.
- **Image Pull Policy:** 필요에 따라 값을 선택합니다. 이 예시에서는 이 필드를 지정할 필요가 없으며 기본 정책을 사용하기만 하면 됩니다.
- **Environment Variable:** 다음 설정 정보를 입력합니다.
GET_HOSTS_FROM = dns

6. 포드 수를 설정합니다.

Number of Pods Manual adjustment Auto adjustment
Set the number of pods directly

Number of Pods

[Advanced Settings](#)

- **Manual adjustment:** 포드 수를 설정합니다. 이 예에서 포드 수는 1입니다. '+' 또는 '-'를 클릭하여 포드 수를 조정할 수 있습니다.
- **Auto adjustment:** 지정된 조건이 충족되면 시스템이 자동으로 pod 수를 조정합니다. 자세한 내용은 [Auto Scaling](#)을 참고하십시오.

7. 다음 이미지와 같이 워크로드의 액세스 모드를 지정합니다.

Access Settings (Service)

Service Enable

Service Access Via Internet Intra-cluster Via VPC Node Port Access [How to select](#)

Automatically create a public CLB (0.003 USD/hour) to provide Internet access. It supports TCP/UDP protocol. Public network access is applicable to web front-end services. If you need to forward via internet using HTTP/HTTPS protocols or by URL, you can go to Ingress page to configure Ingress for routing. [Learn More](#)

Load Balancer Automatic creation Use Existing

Automatically create a CLB for public/private network access to the service. Do not manually modify the CLB listener created by TKE. [Learn more](#)

Port Mapping

Protocol	Target Port	Port
TCP	80	80

[Add Port Mapping](#)

[Advanced Settings](#)

- **Service:** 'Enable'을 선택합니다.
- **Service Access:** 'Via Internet'를 선택합니다.
- **Load Balancer:** 필요에 따라 값을 선택합니다.
- **Port Mapping:** 프로토콜 드롭다운 목록에서 TCP를 선택하고 포트와 대상 포트를 모두 80으로 설정합니다. 이러한 방식으로 사용자는 로드 밸런서 IP 주소를 사용하여 브라우저에서 frontend 컨테이너에 액세스할 수 있습니다.

8. [Create Workload]을 클릭하여 frontend 서비스 생성을 완료합니다.

Web 애플리케이션 확인

1. 왼쪽 사이드바에서 [\[Cluster\]](#)를 클릭하여 'Cluster Management' 페이지로 이동합니다.

2. 생성된 서비스가 있는 클러스터의 ID를 클릭합니다. 표시되는 페이지에서 [Services]>[Service]를 선택합니다.
3. 서비스 관리 페이지에서 frontend 서비스의 로드 밸런서 IP 주소를 복사합니다.

Name	Type	Selector	IP address	Creation Time	Operation
frontend	lb-ex18knzc Load Balancer	k8s-app:frontend, qcloud-ap...	[Redacted] (Service IP)	2020-04-28 10:02:40	Update access method Edit YAML Delete
kubernetes	ClusterIP	N/A	[Redacted] (Service IP)	2020-04-27 19:36:51	Update access method Edit YAML Delete
redis-master	ClusterIP	k8s-app:redis-master, qcloud...	[Redacted] (Service IP)	2020-04-28 09:38:57	Update access method Edit YAML Delete
redis-slave	ClusterIP	k8s-app:redis-slave, qcloud-a...	[Redacted] (Service IP)	2020-04-28 09:48:30	Update access method Edit YAML Delete

설명 :

- redis-master 및 redis-slave 서비스를 생성할 때 **Intra-cluster** 액세스 모드를 선택했습니다. 따라서 이러한 서비스에는 사설 IP 주소가 하나만 있으며 클러스터 내의 다른 서비스에서만 액세스할 수 있습니다.
- frontend 서비스를 생성할 때 **Via Internet** 액세스 모드를 선택했습니다. 따라서 이 서비스는 로드 밸런서 IP 주소(공용 IP 주소)와 사설 IP 주소를 갖습니다. 결과적으로 이 서비스는 클러스터의 다른 서비스에서 액세스할 수 있을 뿐만 아니라 인터넷을 통해서도 액세스할 수 있습니다.

4. 브라우저에서 frontend 서비스의 로드 밸런서 IP 주소에 액세스합니다. 다음 페이지가 나오면 frontend 서비스에 정상적으로 접근할 수 있습니다.

5. 필드에 임의의 문자열을 입력하고 [Submit]을 클릭합니다. 입력한 문자열이 저장되어 페이지 하단에 표시되는 것을 볼 수 있습니다.

브라우저 페이지를 새로고침하여 서비스의 IP 주소에 다시 액세스하십시오. 이전에 입력한 문자열이 페이지에 남아 있어 문자열이 redis에 저장되었음을 나타냅니다.

개발 프랙티스

다음 예시 코드는 Guestbook App의 frontend 서비스에 대한 전체 코드입니다. HTTP 요청을 수신할 때 frontend 서비스는 이것이 set 명령인지 확인합니다.

- set 명령인 경우 프론트엔드 서비스는 매개변수에서 key와 value를 가져와 redis-master 서비스에 연결하고 redis-master 서비스에 key와 value를 적용합니다.
- set 명령이 아닌 경우 프론트엔드 서비스는 redis-slave 서비스에 연결하고 key 매개변수의 value를 얻은 다음 표시할 클라이언트에 값을 반환합니다.

```
<?php
error_reporting(E_ALL);
ini_set('display_errors', 1);
require 'Predis/Autoloader.php';
Predis\Autoloader::register();
if (isset($_GET['cmd']) === true) {
    $host = 'redis-master';
    if (getenv('GET_HOSTS_FROM') == 'env') {
        $host = getenv('REDIS_MASTER_SERVICE_HOST');
    }
    header('Content-Type: application/json');
    if ($_GET['cmd'] == 'set') {
        $client = new Predis\Client([
            'scheme' => 'tcp',
            'host' => $host,
            'port' => 6379,
        ]);
        $client->set($_GET['key'], $_GET['value']);
        print ('{"message": "Updated"}');
    } else {
        $host = 'redis-slave';
        if (getenv('GET_HOSTS_FROM') == 'env') {
            $host = getenv('REDIS_SLAVE_SERVICE_HOST');
        }
        $client = new Predis\Client([
            'scheme' => 'tcp',
            'host' => $host,
            'port' => 6379,
        ]);
        $value = $client->get($_GET['key']);
        print ('{"data": "' . $value . '"}');
    }
}
```

```
} else {  
phpinfo();  
} ?>
```

관련 설명

- frontend 서비스는 redis-master 및 redis-slave 서비스에 액세스할 때 **Service name and port**에 연결합니다. 클러스터는 서비스 이름을 해당 서비스 IP 주소로 해석하고 이 IP 주소에 따라 로드 밸런싱을 수행하는 dns 서비스와 함께 제공됩니다.

redis-slave 서비스가 3개의 Pod를 관리한다고 가정합니다. 프론트엔드 서비스가 redis-slave 서비스와 포트 6379에 직접 연결되어 있는 경우 dns 서비스는 redis-slave 서비스를 redis-slave 서비스의 IP 주소(로드 밸런서 IP 주소와 유사한 유동 IP 주소)로 자동 해석합니다. 그러면 dns 서비스는 redis-slave 서비스의 IP 주소를 기반으로 자동으로 로드 밸런싱을 수행하고 redis-slave 서비스의 해당 Pod에 요청을 보냅니다.

- 컨테이너에 대한 환경 변수를 지정할 때 다음 사항에 유의하십시오.
- **Default setting (recommended):** 런타임 시 frontend 컨테이너는 사전 설정된 환경 변수 GET_HOSTS_FROM = dns를 읽은 다음 서비스 이름을 사용하여 서비스에 직접 연결합니다.
- **Other setting:** redis-master 또는 redis-slave 서비스의 도메인 이름을 얻으려면 다른 환경 변수를 지정해야 합니다.