

# Tencent Kubernetes Engine

クイックスタート

製品ドキュメント



Tencent Cloud

## 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.

## カタログ：

クイックスタート

初心者ガイド

標準クラスターのクイック作成

ビギナー向け事例

簡易Nginxサービスの構築

Hello Worldサービスの手動構築

単一インスタンス版のWordPress

TencentDBを用いたWordPress

簡易Webアプリケーションの構築

# クイックスタート 初心者ガイド

最終更新日：：2023-05-05 20:02:24

本文では、Tencent Kubernetes Engine (TKE) についてわかりやすく説明します。ユーザーがこのガイドに従ってTKEを簡単に開始することができます。

## 1. テンセントクバネティスエンジンとは？

Tencent Kubernetes Engine(TKE)は、ネイティブのkubernetesをベースとしてコンテナを中心に据えた、拡張性が高くハイパフォーマンスなコンテナ管理サービスを提供します。Tencent CloudのIaaS製品との密接な連携により、お客様のビジネスの速やかなコンテナ化をお手伝いします。詳細については、[製品概要](#)をご参照ください。TKEは、[TKEコンソール](#)によってクラスターやサービスの運用をサポートします。

## 2. TKE料金について

TKEは、さまざまな種類のKubernetesクラスターの作成をサポートしており、クラスターの種類によって課金項目と課金基準が異なります。課金モデルや具大的な価格については、[購入ガイド](#)をご参照ください。

## 3. TKEの使用

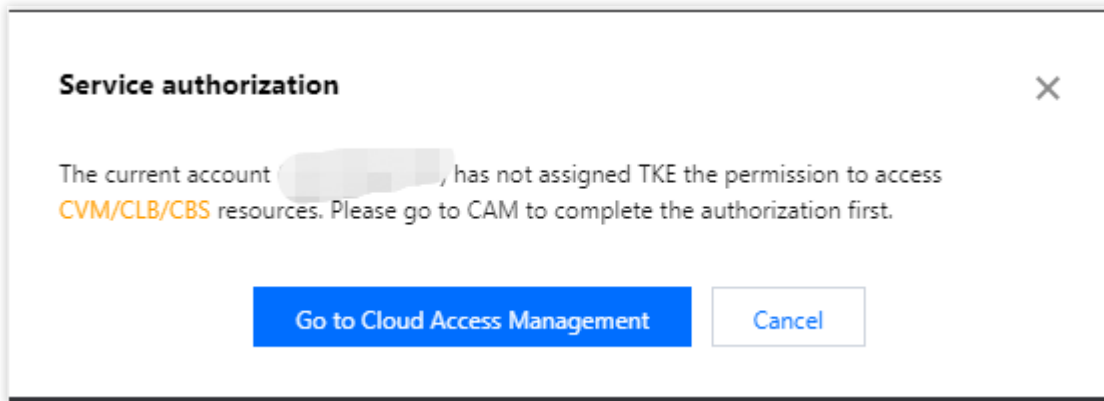
### 3.1 登録および認証

TKEを使用する前に、[Tencent Cloudアカウントの登録](#)と[実名認証](#)を完了する必要があります。

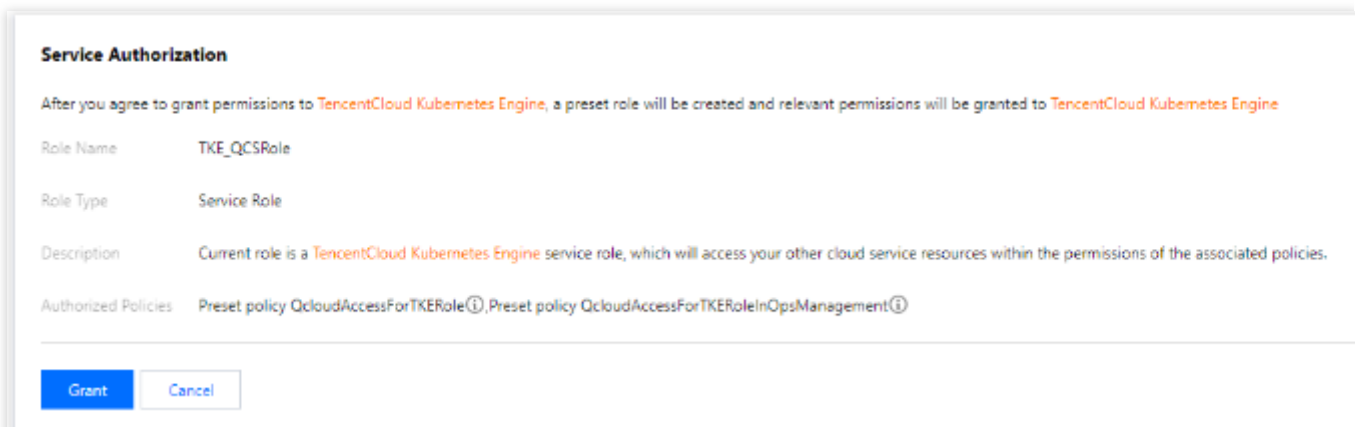
### 3.2 ロール権限授与

他のクラウドサービスリソースに正常にアクセスするには、現在のサービスロールに権限を授与してください。Tencent Cloudコンソールにおいて、[クラウド製品](#) > [TKE](#)を選択し、[TKEコンソール](#)に進み、インターフェースのプロンプトに従ってTKEを権限承認します。サービスの承認を完了して関連のリソース操作権限を取得した時点から、クラスターの作成を開始できます。操作手順は次のとおりです。

1. 「サービスの権限承認」というポップアップウィンドウの情報を確認し、[CAM](#)へをクリックします。下の図に示すとおりです。



- 「ロール管理」ページで、ロール関連情報の詳細を確認します。下の図に示すとおりです。



- 承認に同意するをクリックすると承認操作が完了し、直ちに[Tencent Kubernetes Engine購入ページ](#)に移動して関連製品を購入できます。

### 3.3 クラスターの作成

標準マネージド版クラスターをクイック作成する方法については、[標準クラスターのクイック作成](#)のドキュメントをご参照ください。標準マネージド版クラスターを作成する全プロセスについては、[クラスターの作成](#)のドキュメントをご参照ください。

より多くのタイプのクラスターを使用する必要がある場合は、[Serverlessクラスターの作成](#)、[コンテナインスタンスの作成](#)および[エッジクラスターの作成](#)のドキュメントをご参照ください。

### 3.4 ワークロードのデプロイ

TKEは、イメージデプロイおよびYAMLファイルオーケストレーションの両方の方法でワークロードをデプロイできます。

イメージテンプレートを介してステートレスロードをデプロイ (Deployment) したい場合、操作の詳細については、[シンプルなNginxサービスの作成](#)または[シングルインスタンス版WordPressの作成](#)をご参照ください。

カスタムイメージを介してワークロードをデプロイしたい場合、操作の詳細については、[Hello Worldサービスの手動ビルド](#)をご参照ください。

### 3.5 クラスターの運用保守

コンテナサービスTKEは、クラスター、アプリケーション、ストレージ、ネットワークなどのモジュールの管理プラットフォームとして機能します。詳細な情報や実践的な作業が必要な場合は、以下を参照してさらに理解するうえで使用してください。

次のことをしたい場合	以下をご参照ください
KubernetesのコマンドラインツールKubectlを介して、ローカルのクライアントマシンからTKEクラスターに接続します	<a href="#">クラスターとの接続</a>
実行中のKubernetesクラスターをアップグレードします	<a href="#">クラスターのアップグレード</a>
作成済みのKubernetesクラスターにインスタンスを追加します	<a href="#">ノードの追加</a>
Kubernetesクラスター内ノードを管理します	<a href="#">ノードプールの作成</a>
コンソールからKubernetesのネイティブオブジェクトを直接操作します	<a href="#">Kubernetesオブジェクトの管理</a>
Serviceメソッドを使用して、コンテナ群に固定アクセスポータルを提供します	<a href="#">Service基本機能</a>
Ingressリソースによって異なる転送ルールを設定します	<a href="#">Ingress管理</a>
TKEのストレージ機能を使用します	<a href="#">ストレージ管理の概要</a>
コンテナネットワークのアドレス範囲内のIPアドレスをクラスター内コンテナに割り当てます	<a href="#">コンテナネットワークの概要</a>
Kubernetesクラスター内サービスログのストレージと分析を行います	<a href="#">ログキャプチャ</a>
クラスターのモニタリングを行います	<a href="#">インスタンス管理のモニタリング</a>
Tencent Container Registry内でホストされるプライベートイメージを使用してアプリケーションのデプロイを行います	<a href="#">TCRエンタープライズ版インスタンス内のコンテナイメージを使用したワークロードの作成</a>

## 4. 入門必読

**基幹ネットワークでTKEを使いたいのですが、可能ですか**

いいえ、TKEは現時点ではプライベートネットワークのみをサポートしています。

**既存のCloud Virtual Machine(CVM)を選択してクラスターに加えることは可能ですか**

可能です。クラスター作成が完了した後は、既存CVMを追加できます。詳細については、[既存ノードの追加](#)をご

参照ください。

#### サービスが起動し続けた状態なのはなぜですか

サービス内コンテナに継続的に実行されているプロセスがない場合、サービスは起動し続けた状態になります。サービスの起動に関するご質問については、[イベントに関するよくあるご質問](#)をご参照ください。

#### クラスターを作成する前、どのようにネットワークを計画すればよいですか

クラスターを作成する場合、クラスターネットワークセグメントとコンテナネットワークは重複できません。通常、Virtual Private Cloud(VPC)内のサブネットを選択して、クラスターのノードネットワークに用います。詳細については、[コンテナネットワークとクラスターネットワークの説明](#)をご参照ください。

#### 作成済みのサービスには、どのようにアクセスするのですか

提供されるアクセスポータルはアクセス方法によって異なります。詳細については、[サービスアクセス方法](#)。

#### コンテナはどのようにパブリックネットワークにアクセスするのですか

コンテナは存在するノードにブリックネットワークIP帯域幅がある場合、コンテナはパブリックネットワークに直接アクセスできます、コンテナは存在するノードにブリックネットワークIP帯域幅がない場合、パブリックネットワークにアクセスするにはNATゲートウェイを使ってください。

#### イメージの作り方がわからないのですが、TKEを使えばよいのでしょうか

TKEに統合されたHelm 3.0関連機能により、helm chart、コンテナイメージ、ソフトウェアサービスなど、さまざまな製品やサービスの提供が可能です。作成済みのアプリケーションは、指定したクラスター内で実行され、対応する機能が提供されます。詳細については、[アプリケーション管理](#)をご参照ください。

私の業務では、たくさんのテキストや環境変数を設定する必要がありますが、どのように管理すればよいですか  
[設定項目](#)から設定ファイルを管理できます。

#### サービス同士で相互アクセスしたい場合は、どうすればよいですか

クラスター下の同じNamespaceのサービス同士であれば直接相互アクセスができますが、異なるNamespaceのサービスは、`<service-name>.<namespace-name>.svc.cluster.local` という形式によってアクセスする必要があります。

## 5. フィードバックおよびアドバイス

テンセントクバネティスエンジン製品およびサービスを使用する上で何らかの質問や助言がある場合は、以下の方法でフィードバックしてください。後ほど専門スタッフがお客様のお問い合わせに回答いたします：

製品ドキュメントの問題などを発見した場合（リンク、内容、APIエラーなど）は、ドキュメントのページ右側の **ドキュメントに関するフィードバック** をクリックするか、または問題が存在するコンテンツを選択してフィードバックしてください。

製品に関する問題が発生した場合、[チケットを提出](#)からサポートを求めることができます。

# 標準クラスターのクイック作成

最終更新日：2022-04-14 16:56:29

本文では、TKEを使用してコンテナクラスターを速やかに作成する方法について説明します。

## ステップ1：Tencent Cloudアカウントの登録

Tencent Cloudのアカウント登録が済んでいる場合は、このステップを無視してかまいません。

[ここをクリックしてTencent](#)

## ステップ2：オンラインチャージ

TKEは当面、サービス料金がかかりません。ユーザーが実際に使用したクラウドリソースに応じて課金されます。ここで「ホスティングクラスター」を作成します。このモードでは、クラスターのワーカーノード、永続的なストレージおよびサービスにバインディングされたCLBなどのサービスについて料金を支払ってください。購入前に、アカウントにチャージしてください。詳細については、[オンラインチャージドキュメント](#)をご参照ください。

## ステップ3：サービス認証

[Tencent Cloudコンソール](#)でクラウド製品>TKEを選択し、[TKEコンソール]に移動し、画面の提示に従って、TKE認証を行います。（TKE認証を行った場合は、このステップを省略してください。）

[ここをクリックしてサー](#)

## ステップ4：クラスターの新規作成

[ここをクリックしてクラスタ](#)

### クラスターの情報

「クラスター情報」ページで、クラスター名を入力し、クラスターが存在する地域、クラスターネットワークおよびコンテナネットワークを選択します。その他のデフォルト・オプションはそのままにして、「**次へ**」をクリック



クします。下図の通りです：

The screenshot shows the configuration page for a Tencent Kubernetes Engine cluster. Key elements include:

- Cluster Name:** A text input field with a placeholder "Up to 60 characters".
- Kubernetes Version:** A dropdown menu set to "1.18.4".
- Region:** A grid of region buttons. "Shanghai" is selected and highlighted in blue. Other regions include Guangzhou, Shenzhen Finance, 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, and São Paulo.
- Cluster Network:** A dropdown menu set to "kafuttest".
- Container Network:** A table listing available subnets for container networking.

Below the region selection, there is a note: "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."

Below the cluster network selection, there is a note: "If the current networks are not suitable, please go to the console to [create a VPC](#)."

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-2hjswez	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...

- **クラスター名：**作成するクラスターの名前を入力します。ここでは、「test」を例に挙げます。
- **所在地域：**あなたに最も近い地域を選択してください。
- **クラスターネットワーク：**ノードのネットワークアドレス範囲内のIPアドレスをクラスター内のCVMに割り当てます。ここでは、既存のVPCネットワークを選択します。
- **コンテナネットワーク：**コンテナネットワークアドレス範囲内のIPアドレスをクラスターのコンテナに割り当てます。ここで利用可能なコンテナネットワークを選択します。

## モデルの選択

[モデルの選択]ページで、課金モードを確認し、アベイラビリティゾーンとそれに対応するサブネットを選択し、ノードのモデルを確認して、[次へ]をクリックします。下図の通りです：

Node Source  Existing Nodes

Master Node  Self-deployed Cluster

The default cluster's Master, Etcd 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  Shanghai Zone 2 Shanghai Zone 3 Shanghai Zone 4 Shanghai Zone 5

Node Network  1 250/253 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 [🔗](#)

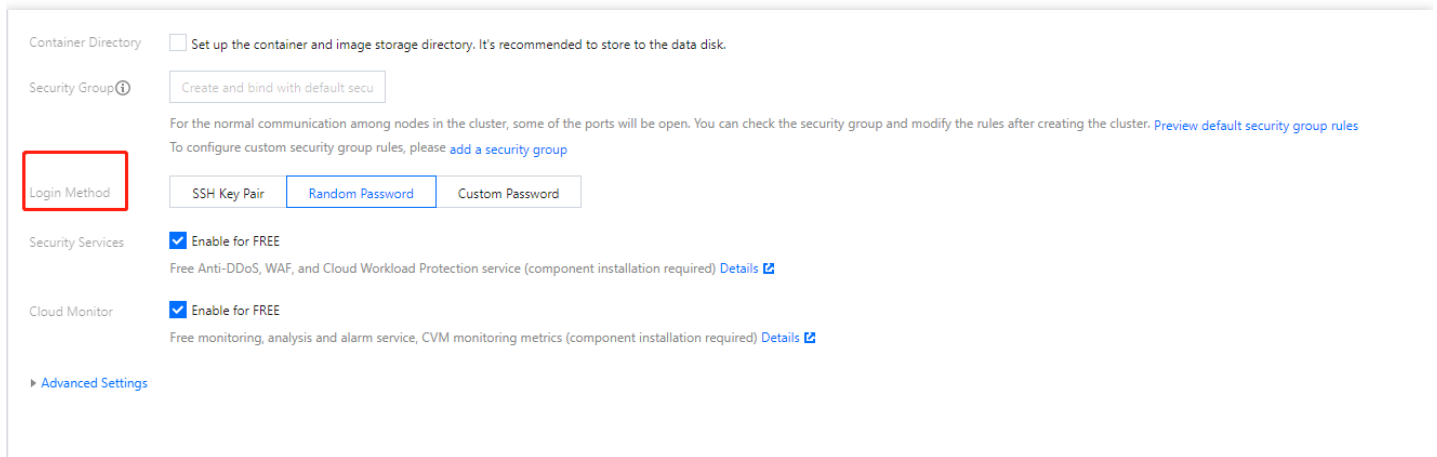
CVM Quantity  1   
VPC network limit: Up to 250 IPs available for current node network

[▶ Advanced Settings](#)

- **ノードのソース**：新規ノードと既存ノードの2つのオプションが用意されています。ここでは、「新規ノード」を選択します。
- **Masterノード**：プラットフォームホスティングと個別デプロイの2つのクラスターモードから選択できます。ここでは、「プラットフォームホスティング」を選択します。
- **課金モード**：従量課金1つの課金モードを提供します。ここでは「従量課金」を選択します。
- **Worker設定**：このモジュールでは、アベイラビリティゾーンとそれに対応するサブネットを選択し、ノードのモデルを確認するだけで済みます。その他の設定項目はデフォルトのままです。
  - **アベイラビリティゾーン**：最も近い地域を選択します。
  - **ノードネットワーク**：現在のVPCネットワークのサブネットを選択します。
  - **モデル**：ここでは、「S1.SMALL1（標準型S1、1コア1GB）」を選択します。

## CVMの設定

「CVMの設定」ページで、ログイン方法を選択し、その他の設定項目はデフォルトのままにして、「次へ」をクリックします。下図の通りです：



The screenshot shows the configuration page for Tencent Kubernetes Engine. The 'Login Method' section is highlighted with a red box. It contains three options: 'SSH Key Pair', 'Random Password' (which is selected), and 'Custom Password'. Other sections include 'Container Directory', 'Security Group', 'Security Services', and 'Cloud Monitor', each with a checkbox and a 'Details' link.

- **ログイン方式**：キーの即時関連付け、パスワードの自動生成、およびパスワードの設定の3つのログイン方法があります。ここでは「パスワードの自動生成」を選択します。

## コンポーネント構成

「コンポーネントの構成」ページでは、必要に応じてストレージ、監視、イメージなどのコンポーネントを選択できます。インストールが不要な場合は、**次へ**をクリックします。ここでは、コンポーネントをインストールせず、その他の設定項目はデフォルトのままにします。

## 情報の確認

「情報の確認」ページで、クラスターで選択した設定情報および料金を確認し、「**完了**」をクリックします。下図の通りです：

**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 

支払いが完了すると、最初のクラスターを作成できます。次に、[TKEコンソール](#)で、作成したホスティングクラスターを表示できます。

## ステップ5：クラスターの表示

作成されたクラスターが[クラスターリスト](#)に表示されます。クラスターIDをクリックすると、クラスターの詳細ページに移動できます。クラスターの「基本情報」ページでは、クラスター情報、ノードおよびネットワーク情報な

どを表示できます。下図の通りです：

The screenshot displays the 'Basic Information' page for a Kubernetes cluster named 'test-1'. The page is divided into two main sections: 'Cluster Information' and 'Node and Network Information'.

**Cluster Information:**

- Cluster Name: test-1
- Cluster ID: cls-amu6sinn
- Deployment Type: Managed Cluster
- Status: Running
- Region: East China(Shanghai)
- Project of New-added Resource: DEFAULT PROJECT
- Kubernetes Version: Master 1.20.6-tke.5(Updates available)Upgrade; Node 1.20.6-tke.5(Updates available)Upgrade
- Runtime Components: docker 18.6
- Cluster Description: N/A
- Tencent Cloud Tags: N/A
- Deletion Protection: Disabled
- Time Created: 2021-09-27 14:59:51

**Node and Network Information:**

- Number of Nodes: 1
- Default OS: [Redacted]
- System Image Source: [Redacted]
- Node Hostname Naming Rule: Auto-generated
- Node Network: [Redacted]
- Container Network Add-on: Global Router
- Container Network: CIDR Block; Register on CCN
- Up to 1024 services per cluster, 64 pods per node, 1008 nodes per cluster
- Network Mode: cni
- VPC-CNI Mode: Disabled
- Service CIDR Block: [Redacted]
- Kube-proxy Proxy Mode: iptables

## ステップ6：クラスターの削除

クラスターの起動時にリソースの消費が開始されます。このステップでは、不要なコストを回避するために、すべてのリソースをパージする方法を説明します。

1. 左側のナビゲーションバーで「**クラスター**」を選択し、「**クラスターの管理**」ページでクラスター削除行の右側の**その他** > **削除**を選択します。下図の通りです：

The screenshot shows a table of Kubernetes clusters. The first row is for cluster 'test-1'. The 'Operation' column has a 'More' dropdown menu open, showing several options. The 'Delete' option is highlighted with a red box.

ID/Name	Monitor	Kubernetes Version	Type/State	Number of Nodes	Allocated/Total	Tencent Cloud Tags	Operation
test-1	[Redacted]	1.20.6	[Redacted]	[Redacted]	[Redacted]	[Redacted]	<a href="#">Configure Alarm Policy</a> <a href="#">Add Existing Node</a> More ▾ <ul style="list-style-type: none"> <li><a href="#">View Cluster Credential</a></li> <li><a href="#">Create Node</a></li> <li><a href="#">Upgrade Master Kubernetes Version</a></li> <li><a href="#">Upgrade Kubernetes Version of The Node</a></li> <li><b>Delete</b></li> </ul>

2. 「**クラスターの削除**」ウィンドウで情報を確認したら、**OK**をクリックしてクラスターを削除します。

## 次の手順：クラスターの使用

本文では、TKEでクラスターを作成および削除する方法について学習しました。作成したクラスターでは、ワークロードを設定し、サービスを作成できます。一般的なタスクは次のとおりです：

- [簡易Nginxサービスの構築](#)
- [単一インスタンス版のWordPress](#)
- [TencentDBを用いたWordPress](#)
- [Hello Worldサービスの手動構築](#)
- [簡易Webアプリケーションの構築](#)

## 問題が発生した場合

ご迷惑をおかけしまして、誠に申し訳ございません。[チケットを提出(<https://console.tencentcloud.com/workorder>)]  
してサポートを依頼してください。

# ビギナー向け事例

## 簡易Nginxサービスの構築

最終更新日：：2023-05-06 19:41:07

### 概要

このドキュメントは、コンテナクラスター内のNginxサービスを素早く理解、作成できるようにするためのものです。

### 前提条件

[Tencent Cloudアカウント](#)の登録済みであること。

すでにクラスターを作成済みであること。詳細については、[クラスターの作成](#)をご参照ください。

### 操作手順

#### Nginxサービス作成

1. TKEコンソールにログインし、左側ナビゲーションバーの[クラスター](#)を選択します。
2. [クラスター管理](#)ページで、作成したいサービスのクラスターIDを選択し、クラスター基本情報ページに進みます。
3. [ワークロード > Deployment](#)ページで、[新規作成](#)をクリックします。パラメータの詳細な内容は、[Deploymentを作成](#)をご参照ください。
4. [Deploymentの新規作成](#)ページで、以下の情報に基づき、ワークロードの基本情報を設定します。下図に示すとおりです。

Name: Please enter a 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: Up to 1000 characters

Namespace: default

Labels: k8s-app = nginx

OS type: Linux

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

**ワークロードの名称:**ここではnginxを例にとります。

**説明:**ワークロードの関連情報を入力します。

**タグ:**この例でのタグデフォルト値は、`k8s-app = nginx`です。

**ネームスペース:**実際の要求に従い選択します。デフォルトはdefaultです。

**データボリューム:**実際のニーズに応じてワークロードのマウントボリュームを設定します。詳細については、[ストレージボリュームの使用説明](#)をご参照ください。

5. 次の情報設定「インスタンス内のコンテナ」をご参照ください。下図に示すとおりです。

Containers in the Pod: test + Add container

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

Image: nginx Select image

Image tag: "latest" is used if it's left empty.

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.

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: 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 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: 0  
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



主なパラメータ情報は下記の通りです：

**名称**：インスタンス内のコンテナ名を入力します。ここではtestを例にとります。

**イメージ**：イメージの**選択**をクリックし、ポップアップボックスで**Docker Hubイメージ** > **nginx**を選択し、**確定**をクリックします。

**イメージバージョン (Tag)**：デフォルト値 `latest` を使用します。

**イメージプルポリシー**：Always、IfNotPresent、Neverという3つのポリシーを提供しますので、必要に応じて選択してください。ここでは、**設定せずデフォルトのポリシーを使用する**を例にとります。

6. 「インスタンス数」では、次の情報に基づきサービスのインスタンス数を設定します。ここでは、**手動調節**を例にとります。インスタンス数の設定は1です。下図に示すとおりです。

Number of Pods ⓘ

**Manual adjustment** Set the number of pods directly

Number of Pods

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

7. 以下の手順に従い、ワークロードへのアクセス設定を行います。下図に示すとおりです。

### 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**：「開始」にチェックを入れます。

**サービスアクセス方法**：「パブリックネットワークLBアクセス」を選択します。

**ロードバランサ**：実際のニーズに応じて選択します。

**ポートマッピング**：TCPプロトコルを選択し、コンテナポートとサービスポートをどちらも80に設定します。

**プロトコル**：通信が必要なプロトコルタイプを選択します。

**コンテナポート**：コンテナ内のアプリケーションプログラムが監視するポートです。ポートの範囲は1~65535です。

**ホストポート**：Cloud Virtual Machine (CVM) IP+ホストポートを介してサービスにアクセスできます。ポートの範囲は30000~32767で、入力しない自動割り当てです。

**サービスポート**：クラスター外ではCloud Load Balancer (CLB) ドメイン名あるいはIP+サービスポートによってサービスにアクセスでき、クラスター内ではサービス名+サービスポートによってサービスにアクセスできます。

**Secret**：TCP SSLプロトコルを選んだ場合のみ選択が必要です。

### 注意

サービスが配置されているクラスターのセキュリティグループは、ノードネットワークとコンテナネットワークを開放している必要があり、さらにポート30000~32768も開放する必要があります。開放していないと、TKEを使用できないといった問題が発生する場合があります。詳細については、[TKEセキュリティグループの設定](#)をご参照ください。

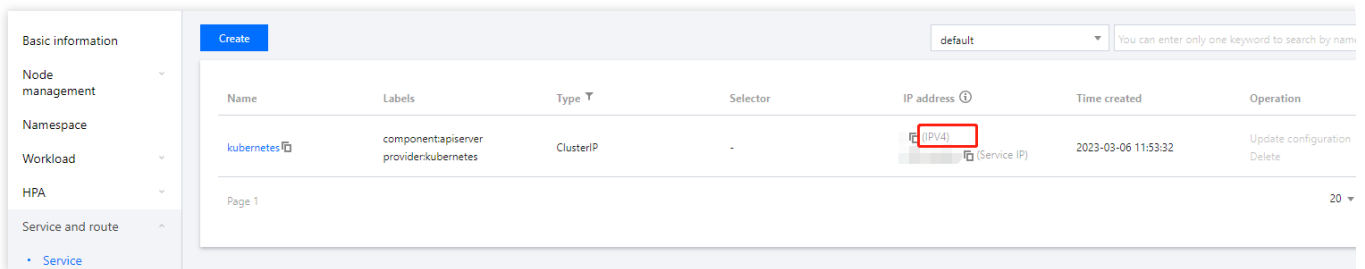
8. **Workload**を作成をクリックし、Nginxサービスの作成を完了します。

## Nginxサービスにアクセス

次の二つの方式でNginxサービスにアクセスできます。

### Cloud Load Balancer IPでNginxサービスにアクセス

1. 左側ナビゲーションバーの**クラスター**をクリックし、「クラスター管理」ページに進みます。
2. Nginxサービスが所在するクラスターIDをクリックし、**サービスとルート > Service**を選択します。
3. Serviceリストページで、NginxサービスのCLB IPをコピーします。下図に示すとおりです。



Name	Labels	Type	Selector	IP address	Time created	Operation
kubernetes	componentapiserver providerkubernetes	ClusterIP	-	[IPV4] (Service IP)	2023-03-06 11:53:32	Update configuration Delete

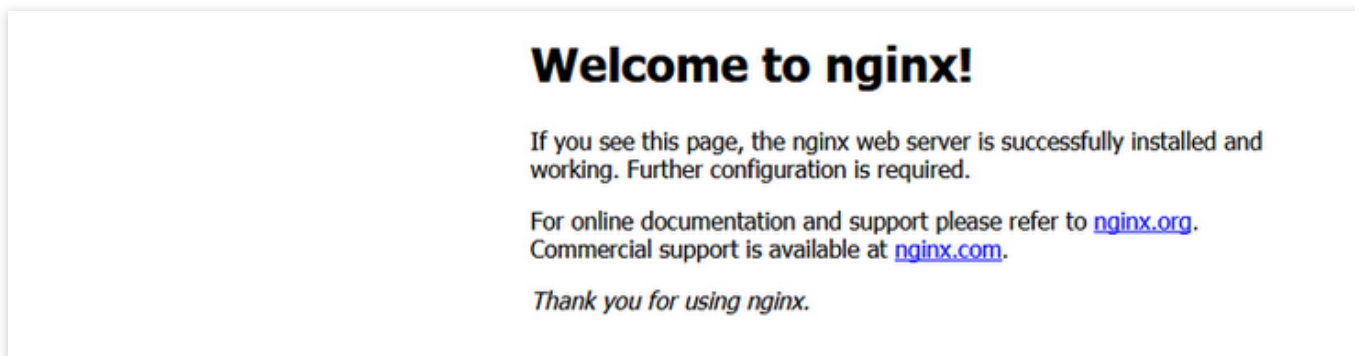
4. ブラウザのアドレス欄にCLB IPを入力し、「**Enter**」を押すとサービスにアクセスできます。

### サービスの名前でNginxサービスにアクセス

クラスター内の他のサービスやコンテナはサービスの名前で直接アクセスできます。

### Nginxサービス検証

サービスは作成されました。サービスにアクセスする時、直接にNginxサーバーのデフォルトウェルカムページに入ります。下図の通りです：



### その他のNginx設定

コンテナの作成に失敗した場合は、[イベントに関するよくあるご質問](#)をご参照ください。

# Hello Worldサービスの手動構築

最終更新日：：2023-05-06 19:41:07

## 概要

このドキュメントは、コンテナクラスター内のNode.js版のHello Worldのサービスをすばやく作成する方法を理解する助けとなるものです。Dockerイメージの作成方法に関する詳しいチュートリアルについては、[dockerイメージの作成方法](#)をご参照ください。

## 前提条件

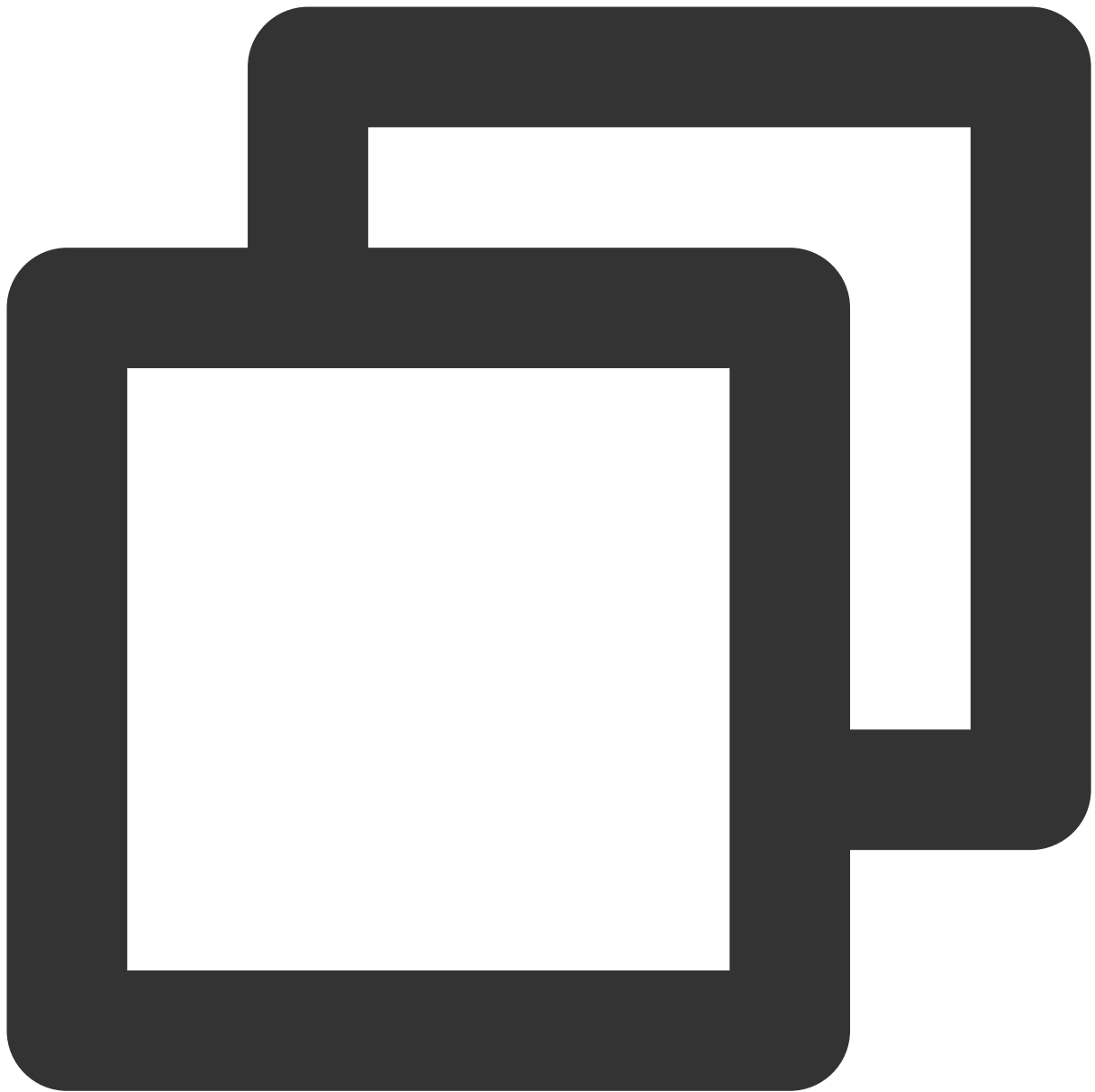
クラスターを作成済みであること。詳細については、[クラスターの作成](#)をご参照ください。  
ノードにログイン済みであり、そのノードにNode.jsがインストールされていること。

## 操作手順

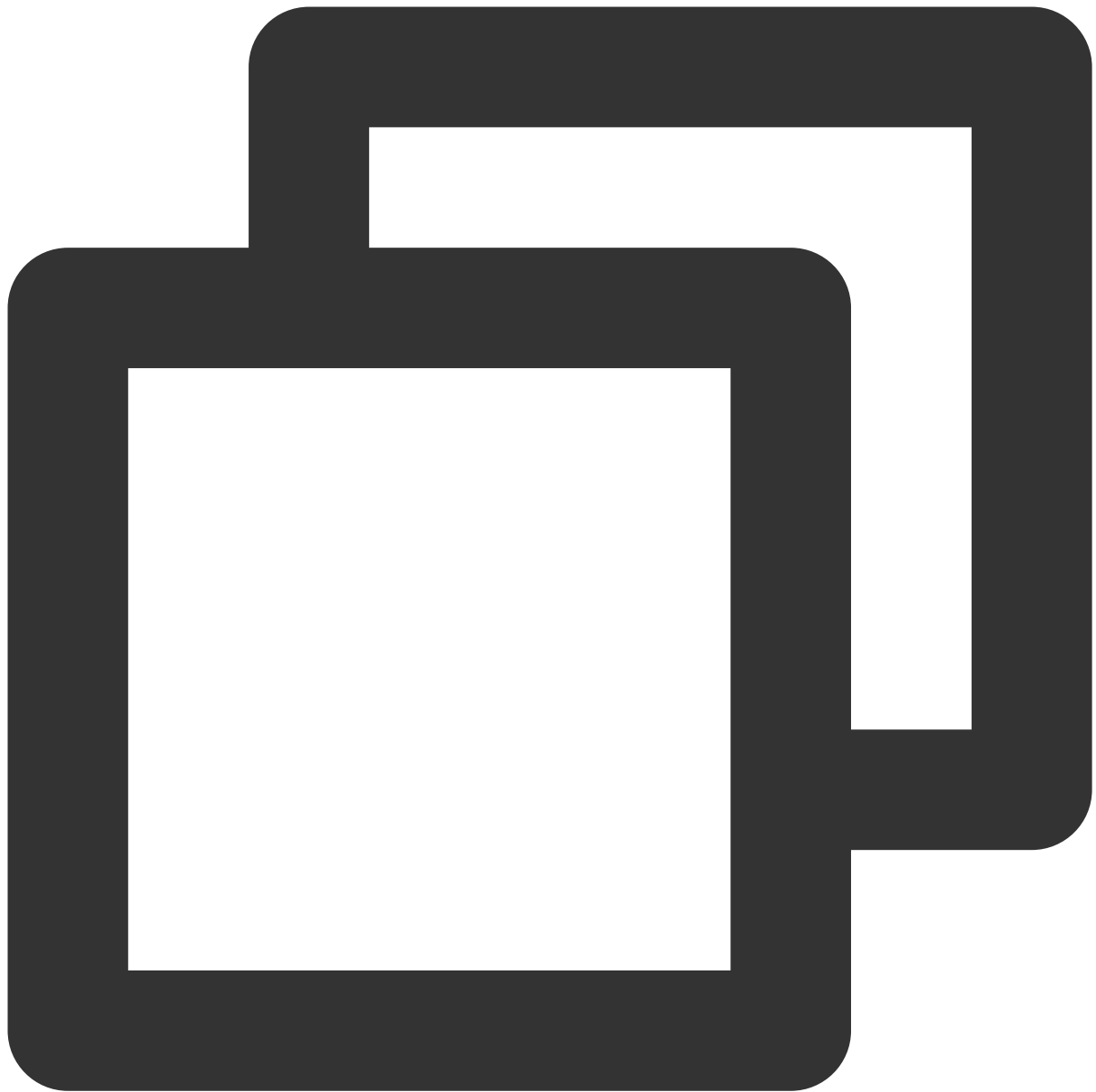
### コーディングとイメージ作成

#### アプリケーション作成

1. 以下のコマンドを順に実行し、hellonodeのフォルダを作成して入ります。

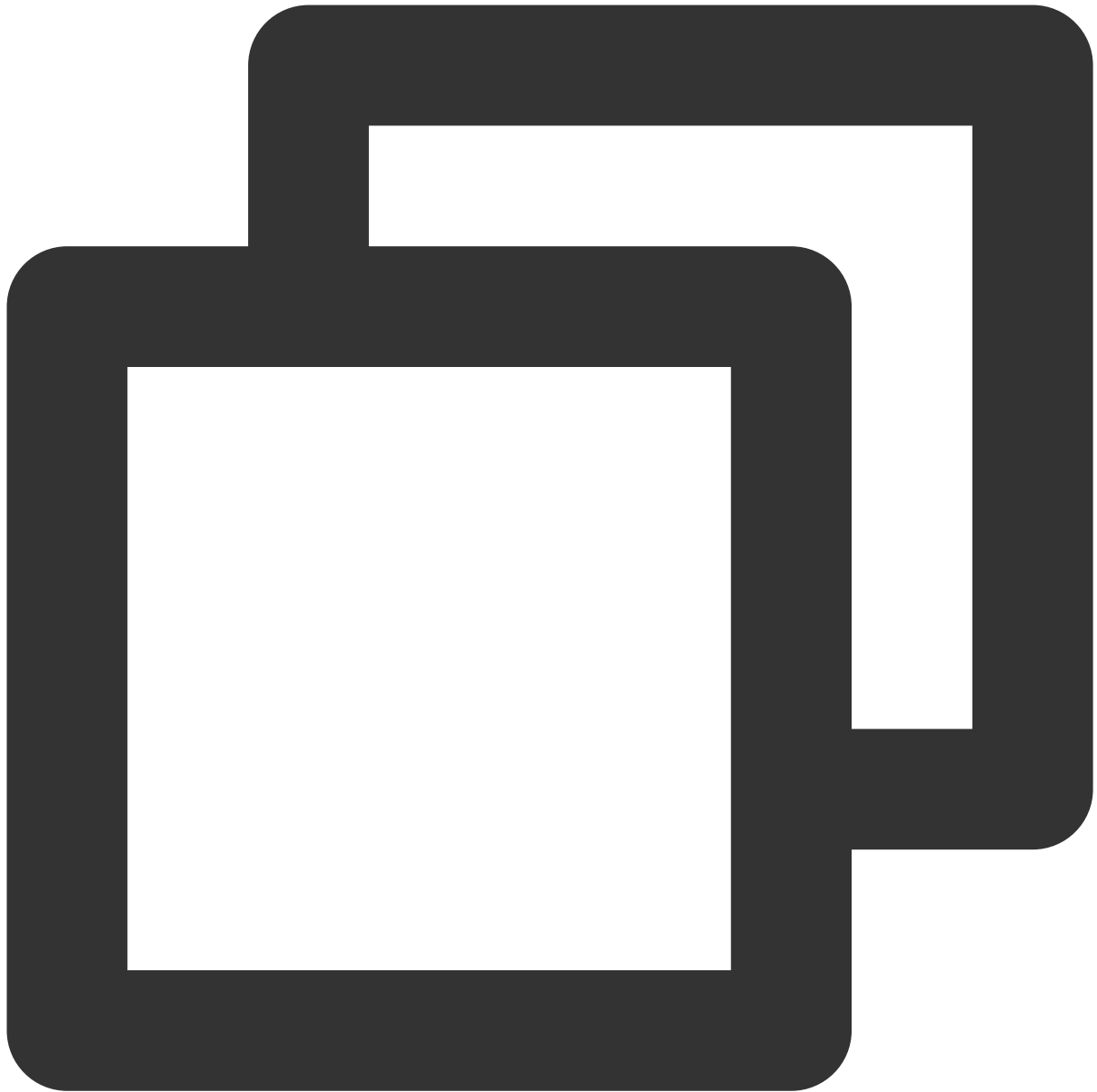


```
mkdir hellonode
```



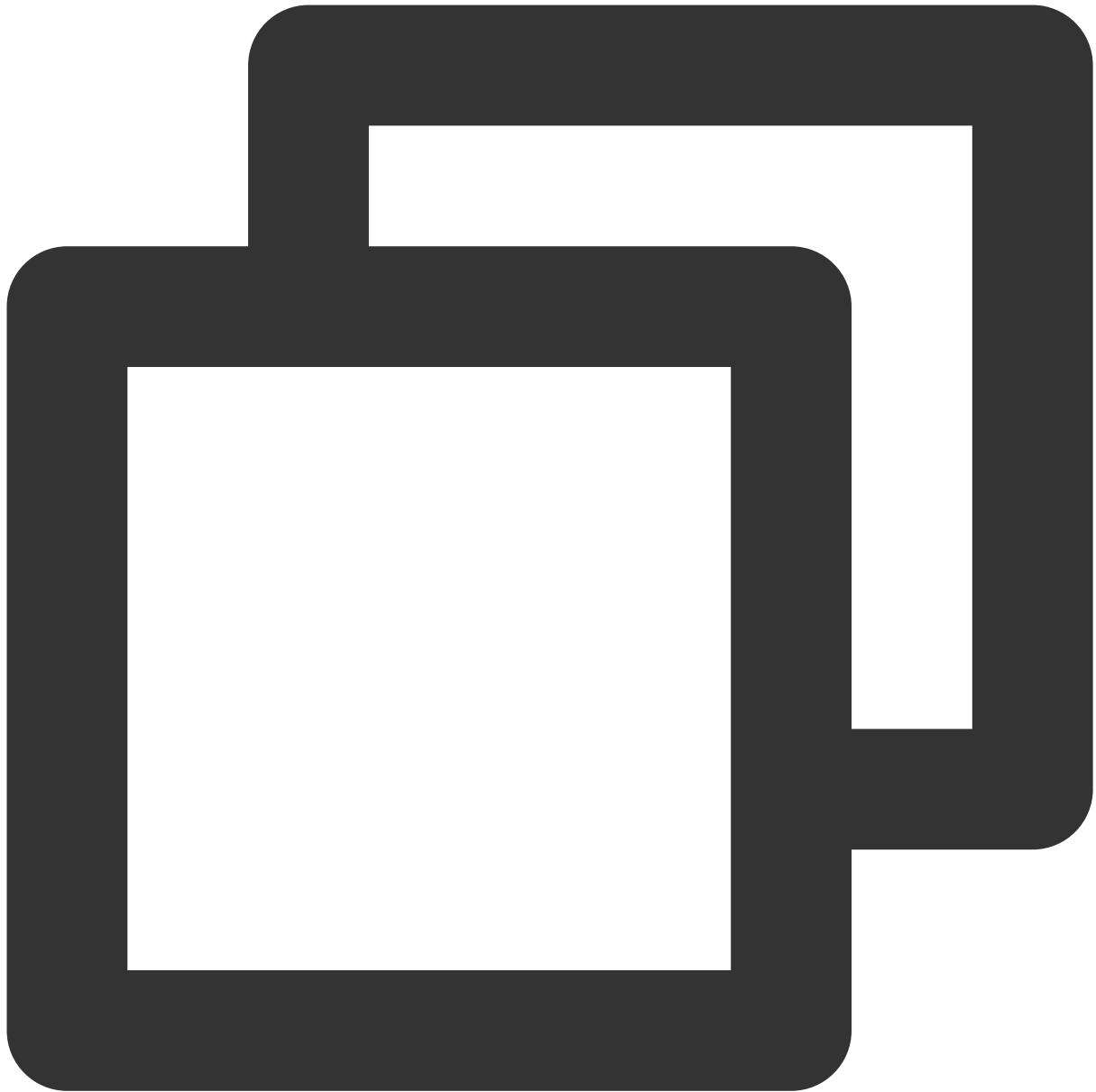
```
cd hellonode/
```

2. 以下のコマンドを順に実行し、`server.js`ファイルを新規作成して開きます。



```
vim server.js
```

3. **i**を押して編集モードに切り替え、以下の内容をserver.jsに入力します。

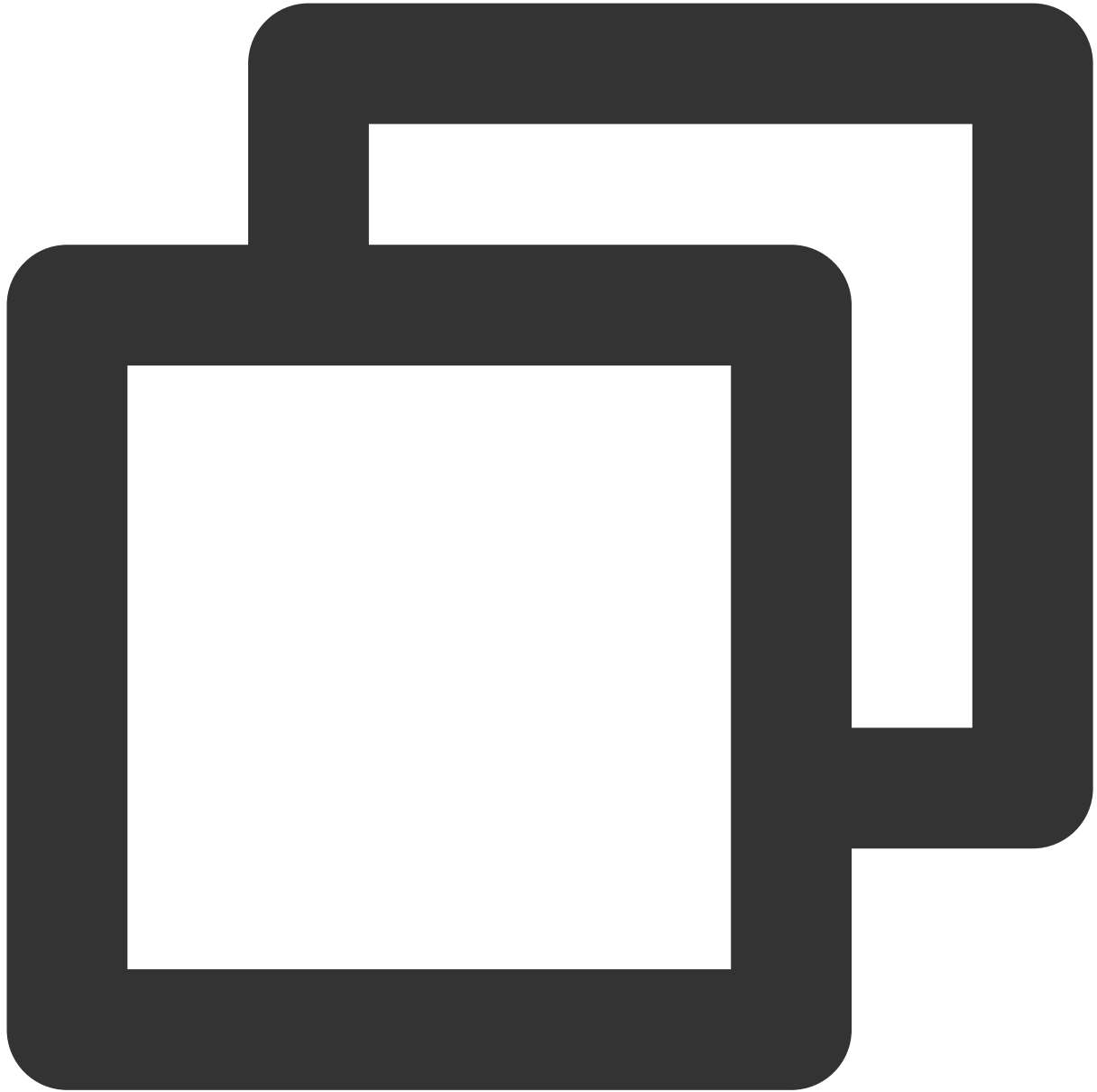


```
var http = require('http');
var handleRequest = function(request, response) {
  console.log('Received request for URL: ' + request.url);
  response.writeHead(200);
  response.end('Hello World!');
};
var www = http.createServer(handleRequest);
www.listen(80);
```

**Esc**を押して、**:wq**と入力し、ファイルを保存して戻ります。



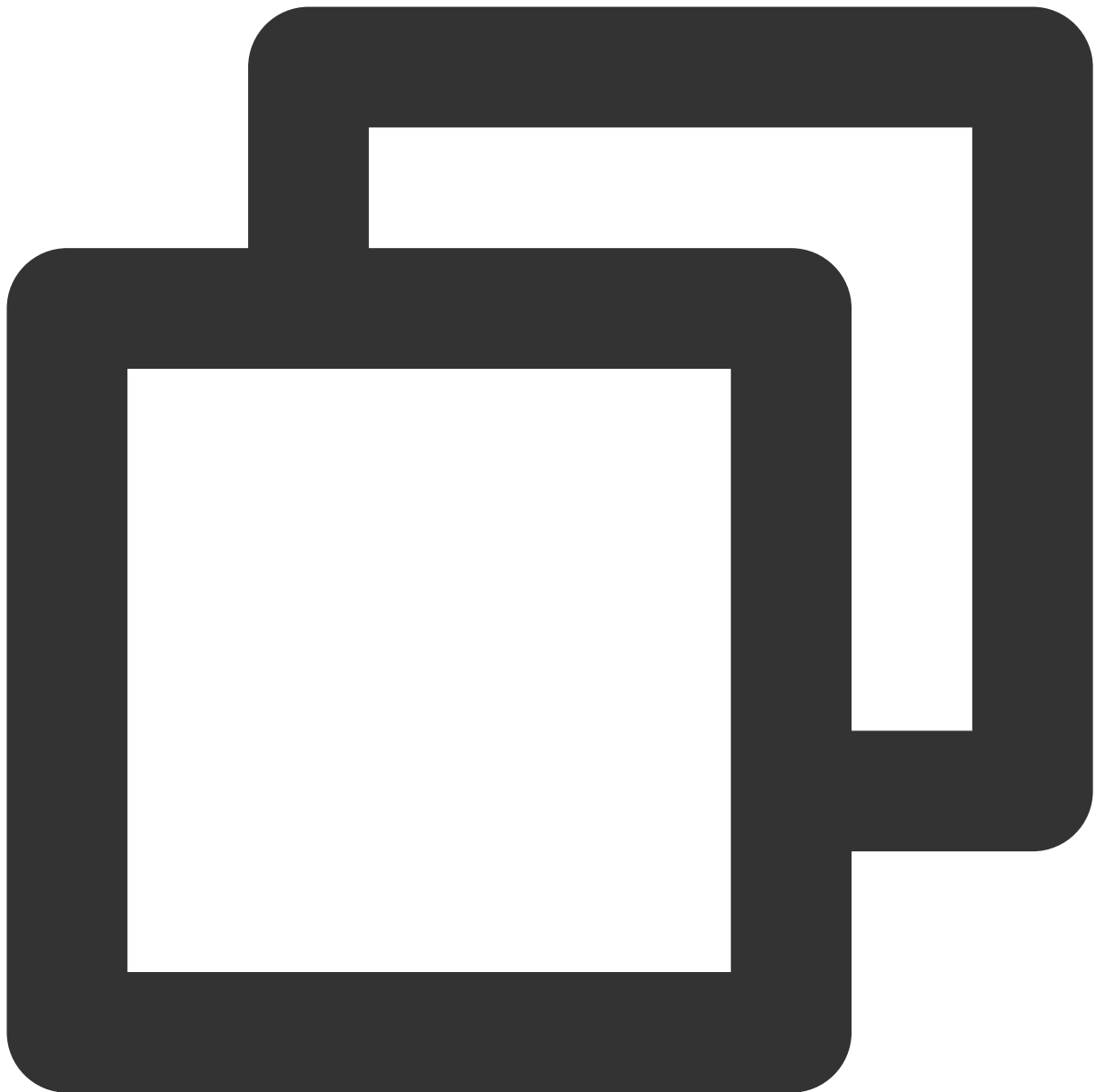
4. 以下のコマンドを順に実行し、`server.js`ファイルを実行します。



```
node server.js
```

5. Hello Worldプログラムをテストします。

方法1：ノードに再度ログインし、以下のコマンドを実行します。



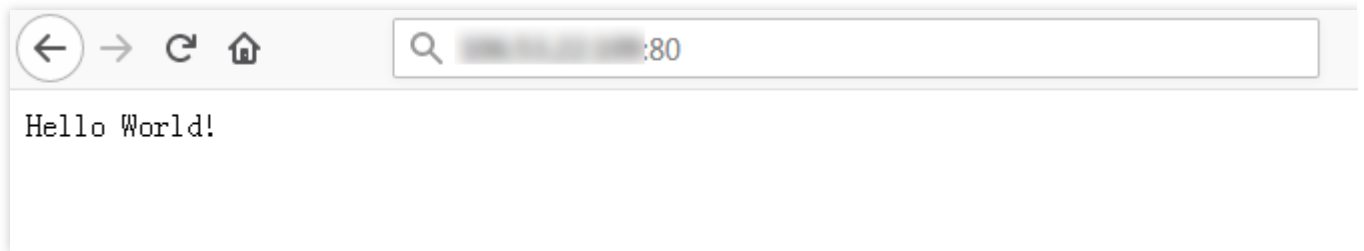
```
curl 127.0.0.1:80
```

次のように表示されれば、Hello Worldプログラムが実行されたことを示します。

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

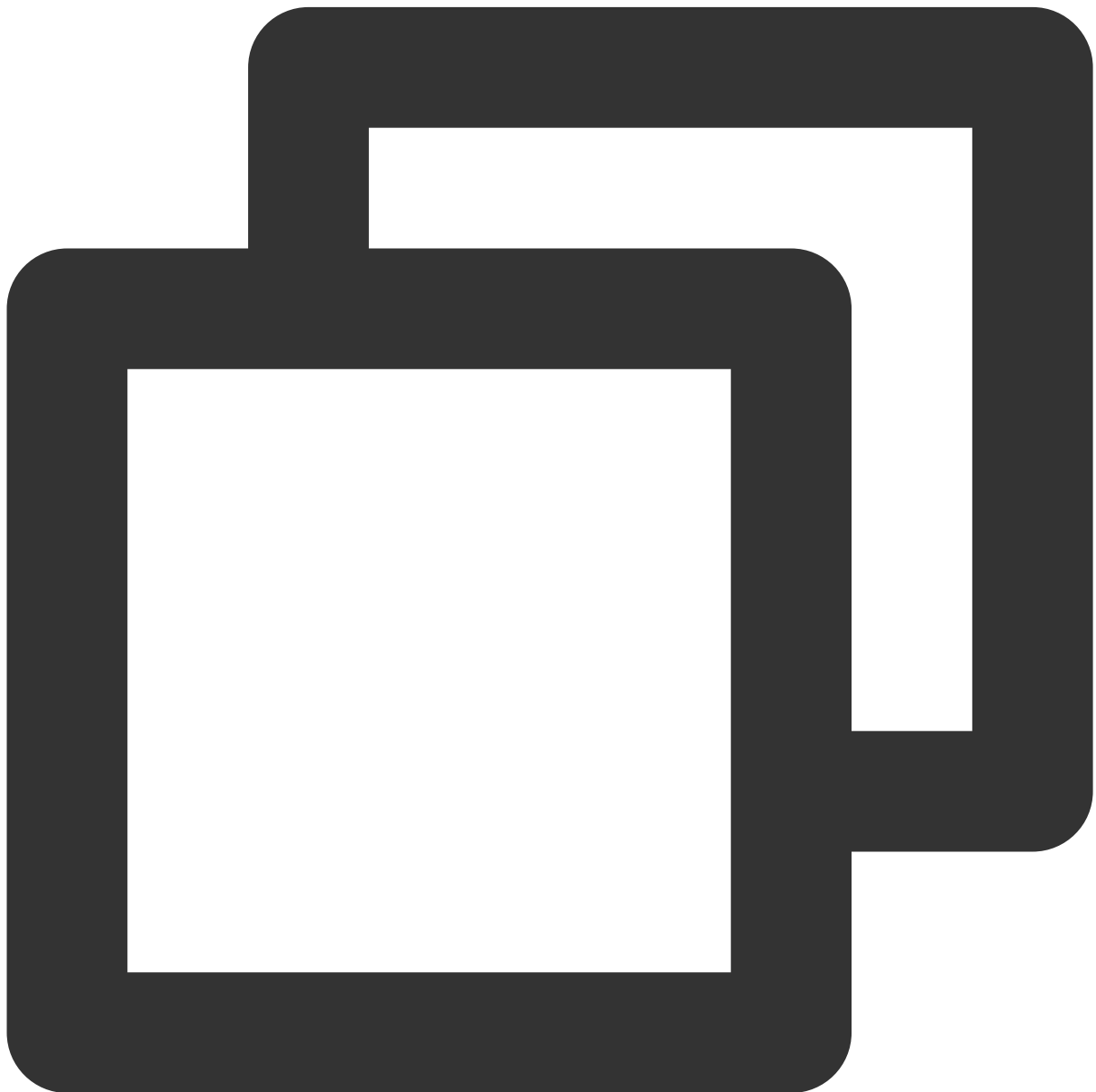
方法2：Cloud Virtual Machine (CVM)インスタンスのパブリックIPアドレスでローカルブラウザを開き、設定済みのポート番号の形式でアクセスします。ポート番号は80です。

次のように表示されれば、Hello Worldプログラムが実行されたことを示します。

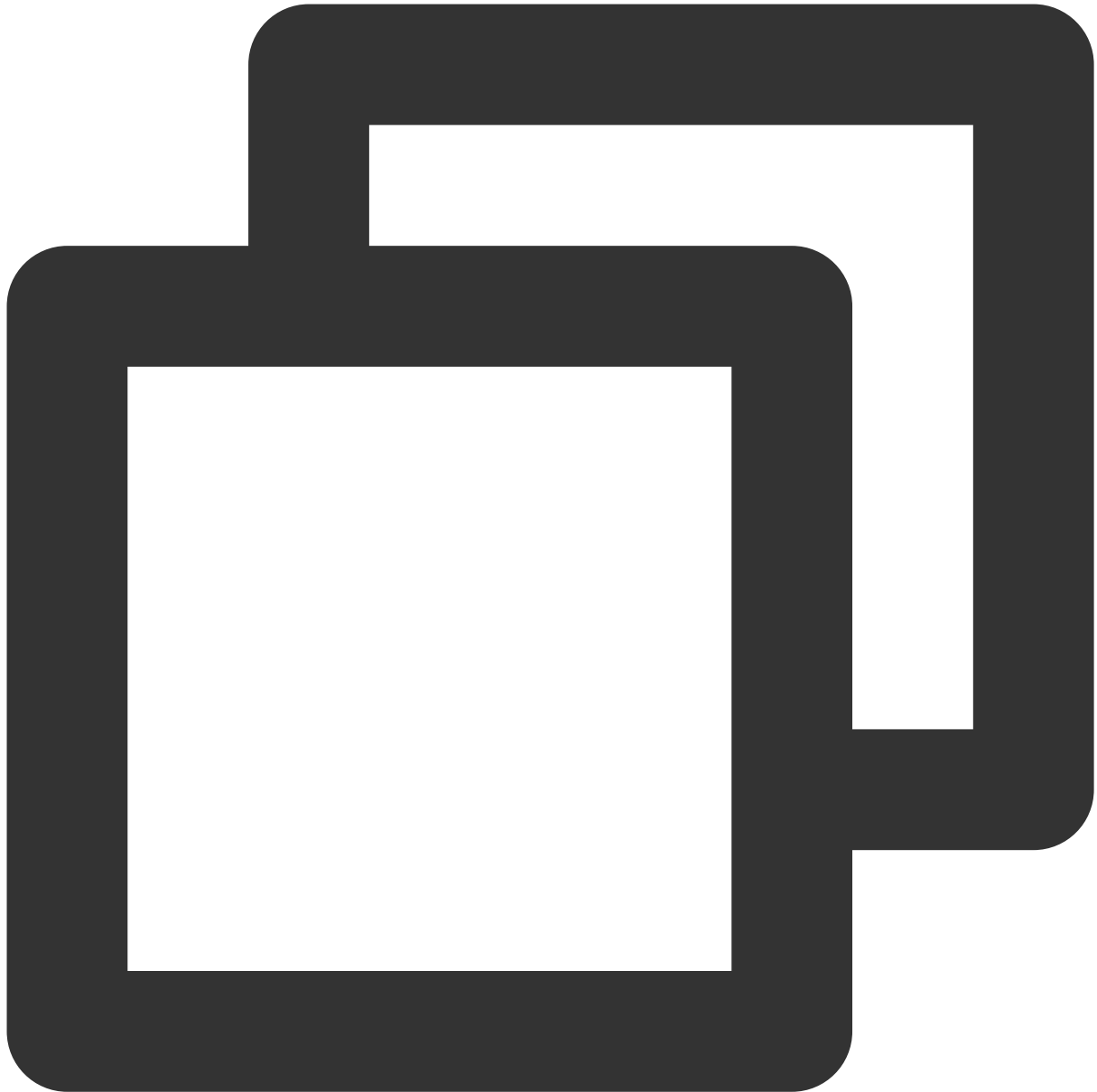


## Dockerイメージ作成

1. 以下のコマンドを順に実行し、hellonodeフォルダにDockerfileファイルを作成します。

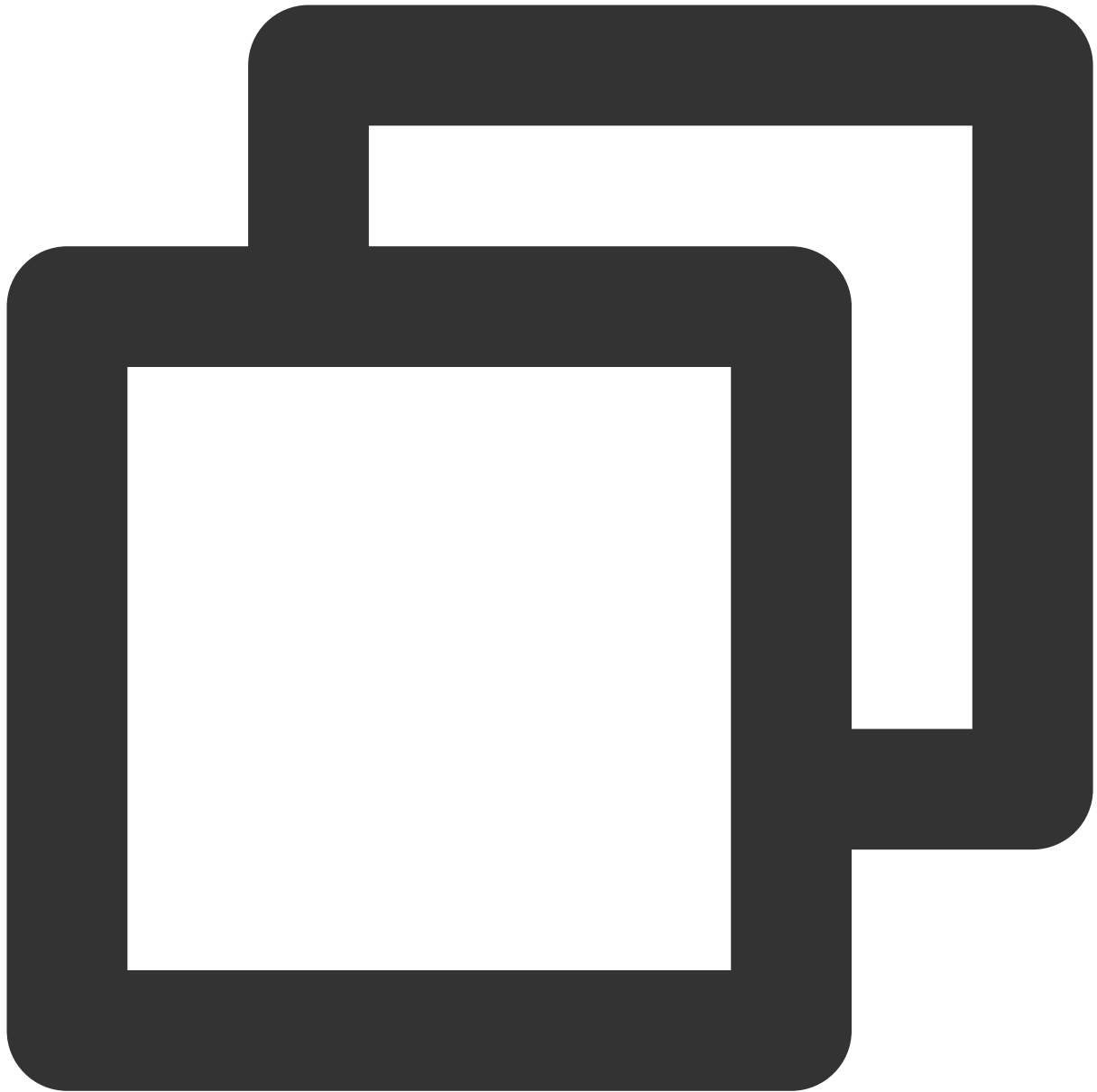


```
cd hellonode
```



```
vim Dockerfile
```

2. **i**を押して編集モードに切り替え、以下の内容をDockerfileファイルに入力します。



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

**Esc**を押して、**:wq**と入力し、ファイルを保存して戻ります。

3. このノードをdockerにインストールし、dockerサービスを起動します。



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

4. 次のコマンドを実行して、イメージを作成します。



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

5. 次のコマンドを実行して、作成したhello-nodeイメージを確認します。



```
docker images
```

結果は次のように表示されれば、hello-nodeイメージが成功的に構築されたことを示し、このIMAGE IDを記録します。下図の通りです：

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

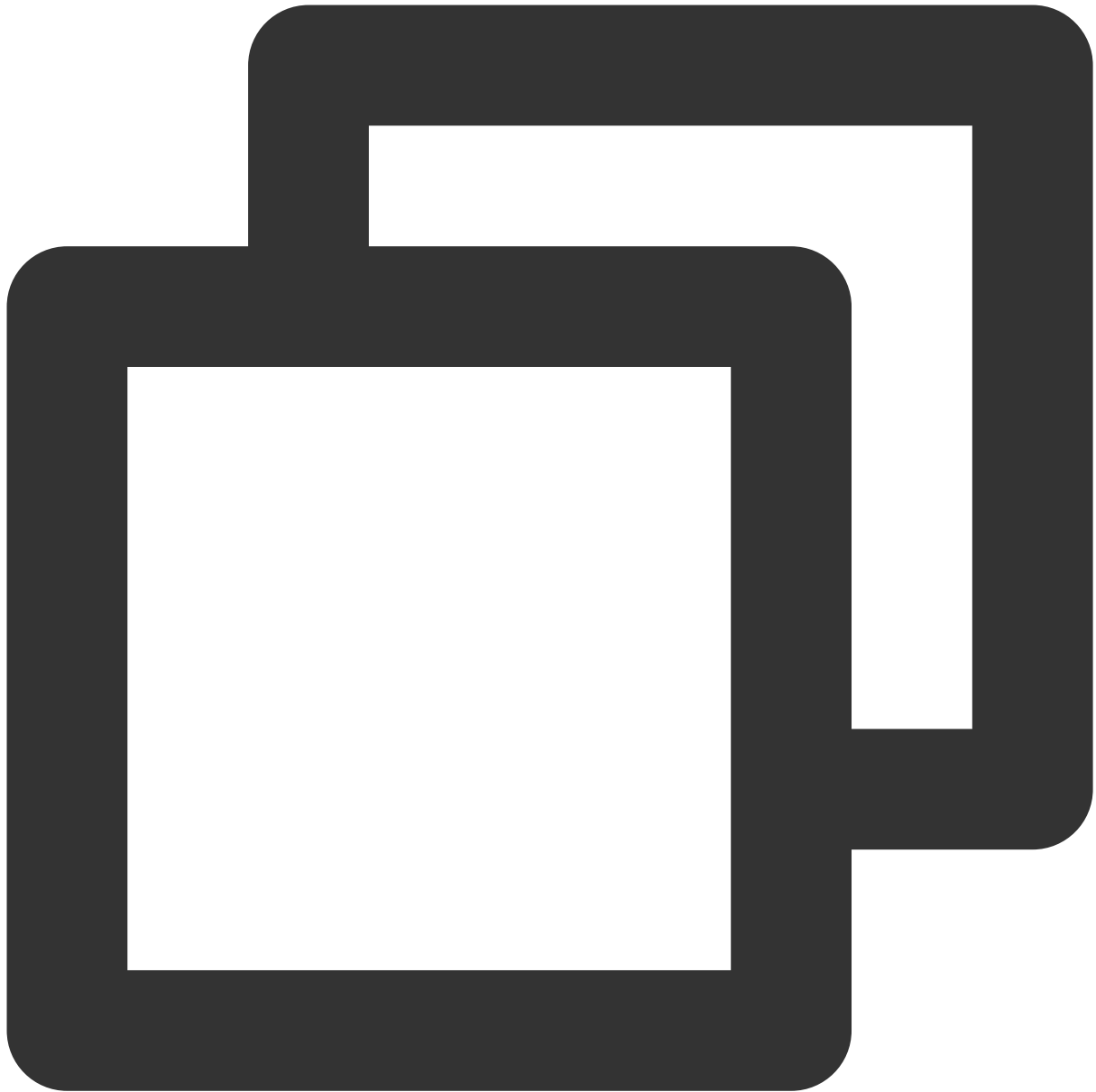
このイメージをqcloudイメージウェアハウスにアップロード

次のコマンドを順番に実行して、イメージをqcloudイメージウェアハウスにアップロードします。





```
docker tag IMAGEID ccr.ccs.tencentyun.com/ネームスペース/hello-node:v1
```



```
docker login ccr.ccs.tencentyun.com
docker push ccr.ccs.tencentyun.com/ネームスペース/hello-node:v1
```

#### 説明

コマンドのIMAGEIDを[イメージの確認](#)に記録されたIMAGEIDに置き換えてください。

コマンド内のネームスペースを作成したネームスペースに置換してください。ネームスペースがない場合は、まずネームスペースを作成してください。詳細については、[ネームスペースの作成](#)をご参照ください。

次の結果が表示されれば、イメージが成功的にアップロードされたことを示します。

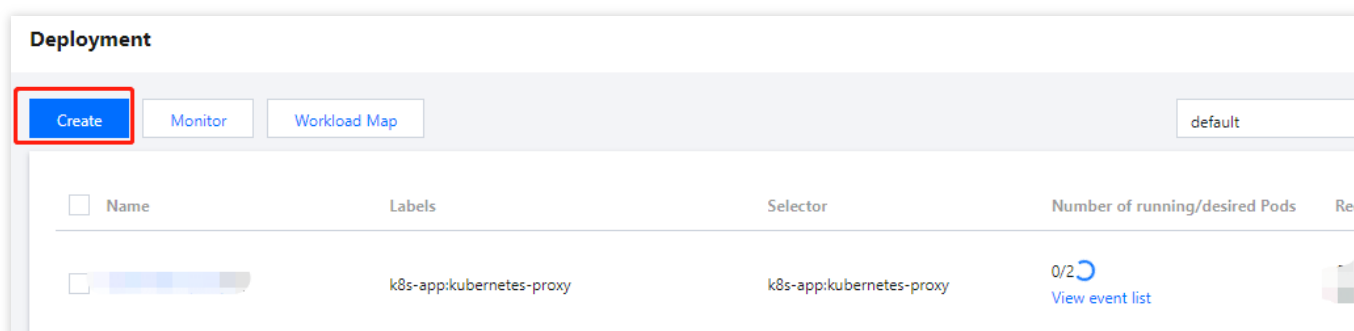
```
[root@VM_2_5_centos hellonode]# sudo docker tag ccr.ccs.tencentyun.com/
[root@VM_2_5_centos hellonode]# sudo docker push ccr.ccs.tencentyun.com/: test/hello
The push refers to repository [ccr.ccs.tencentyun.com/ test/helloworld]
7357e3a21b1f: Pushed
20a6f9d228c0: Pushed
80c332ac5101: Pushed
04dc8c446a38: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

## このイメージでHello Worldサービス作成

### 注意

Hello Worldサービスを使用して作成する前に、クラスターを持っている必要があります。クラスターがない場合は、まずクラスターを作成してください。詳細については、[クラスターの作成](#)をご参照ください。

1. [TKEコンソール](#)にログインし、左側ナビゲーションバーから**クラスター**を選択します。
2. **クラスター管理**ページで、作成したいサービスのクラスターIDを選択し、**クラスター詳細**ページに進みます。
3. **ワークロード > Deployment**を選択し、Deploymentページで**新規作成**をクリックします。下の図に示すとおりです。



4. **Deployment**の**新規作成**ページで、以下の情報に基づき、ワークロードの基本情報を設定します。下の図に示すとおりです。

The screenshot shows a configuration form for a container. The fields are as follows:

- Name:** helloworld. Below the input is a note: "Up to 63 characters, including lowercase letters, numbers, and hyphens ("-"). It must begin with a lowercase letter, and end with a nu".
- Description:** helloworld serve.
- Namespace:** default (dropdown menu).
- Labels:** k8s-app = helloworld. Below the input is a note: "The key name cannot exceed 63 chars. It supports letters, numbers, "/" and "-". "/" cannot be placed at the beginning. A prefix is supp. The label key value can only include letters, numbers and separators ("-", "\_", "."). It must start and end with letters and numbers."
- OS type:** Linux (dropdown menu with a refresh icon). Below the input is a note: "Configurations are initialized when you change the OS type for the container."
- Volume (optional):** Add volume. Below the input is a note: "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 mo".

**ワークロード名:** 作成するワークロードの名前を入力します。ここでは、**helloworld**を例として使用します。

**説明:** ワークロードの関連情報を入力します。

**ネームスペース:** 実際のニーズに応じて選択してください。

**タグ:** key = valueキーバリューペア。この例では、タグのデフォルト値は **k8s-app = helloworld**とします。

**OSタイプ:** 実際のニーズに応じて選択します。ここでは、**Linux**を例として使用します。

**データボリューム:** 実際のニーズに応じてワークロードのマウントボリュームを設定します。詳細については、[Volume管理](#)をご参照ください。

5. 以下の情報を参照して、「インスタンス内コンテナ」を設定してください。

5.1 インスタンス内コンテナ名を入力します。ここでは例として**helloworld**を使用します。

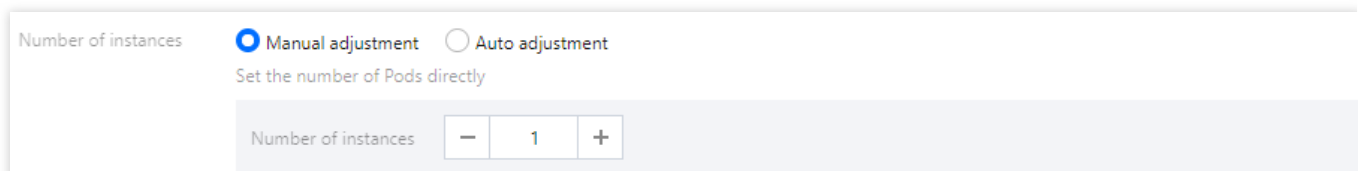
5.2 **イメージを選択**をクリックし、ポップアップボックスで**マイイメージ**を選択し、検索ボックス機能で **helloworld**イメージを探して**OK**をクリックします。

主なパラメータ情報は下記の通りです：

**イメージバージョン (Tag) :** デフォルト値**latest**を使用します。

**イメージプルポリシー:** **Always**、**IfNotPresent**および**Never**という3つのポリシーを提供しますので、必要に応じて選択してください。ここでは、何も設定していないデフォルトのポリシーを例として使用します。

6. 「インスタンス数」では、以下の情報に基づいてサービスのインスタンス数を設定します。下図に示すとおりです。



**手動調節**：インスタンス数を設定します。ここではインスタンス数を1とします。「+」または「-」をクリックすると、インスタンス数を制御できます。

**自動調節**：設定した条件のいずれかが満たされると、インスタンス(pod)を自動調整します。詳細については、[サービスの自動スケーリング](#)をご参照ください。

7. 以下の手順に従い、ワークロードへのアクセス設定を行います。

**Service**：「開始」にチェックを入れます。

**サービスアクセス方法**：「パブリックネットワークLBアクセス」を選択します。

**ロードバランサ**：実際のニーズに応じて選択します。

**ポートマッピング**：TCPプロトコルを選択し、コンテナポートとサービスポートをどちらも80に設定します。

### 注意

サービスが配置されているクラスターのセキュリティグループは、ノードネットワークとコンテナネットワークを開放している必要があり、さらにポート30000~32768も開放する必要があります。開放していないと、TKEを使用できないといった問題が発生する場合があります。詳細については、[TKEセキュリティグループの設定](#)をご参照ください。

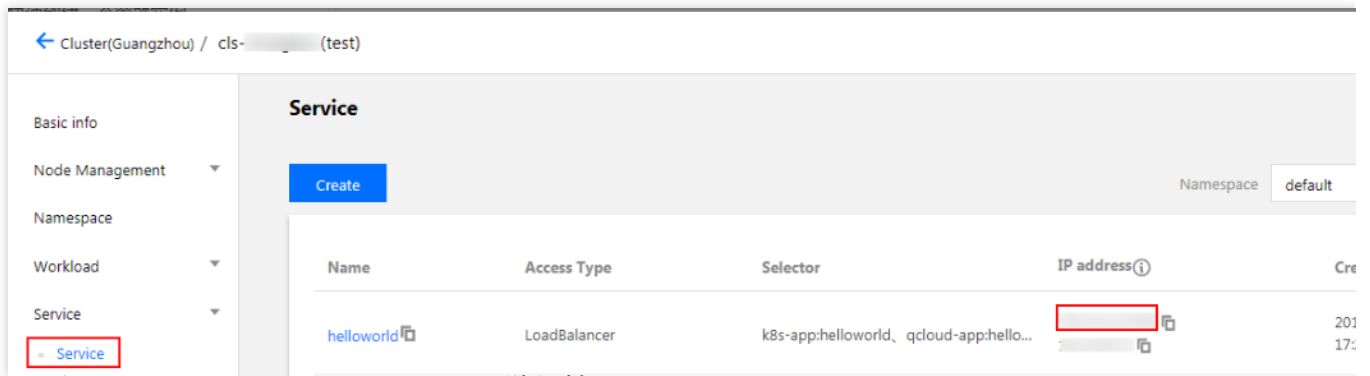
8. **Deployment**の作成をクリックし、Hello Worldサービスの作成を完了します。

## Hello Worldサービスにアクセス

次の二つの方式でHello Worldサービスにアクセスできます。

### Cloud Load Balancer IPでHello Worldサービスにアクセス

1. [TKEコンソール](#)にログインし、左側ナビゲーションバーから**クラスター**を選択します。
2. **クラスター管理**ページで、Hello WorldサービスのクラスターIDをクリックし、**クラスター詳細ページ**に進みます。
3. **Service** ページで、**サービスとルーティング > Service**を選択します。
4. サービス管理ページで、helloworldサービスのCloud Load Balancer IPをコピーします。下図に示すとおりです



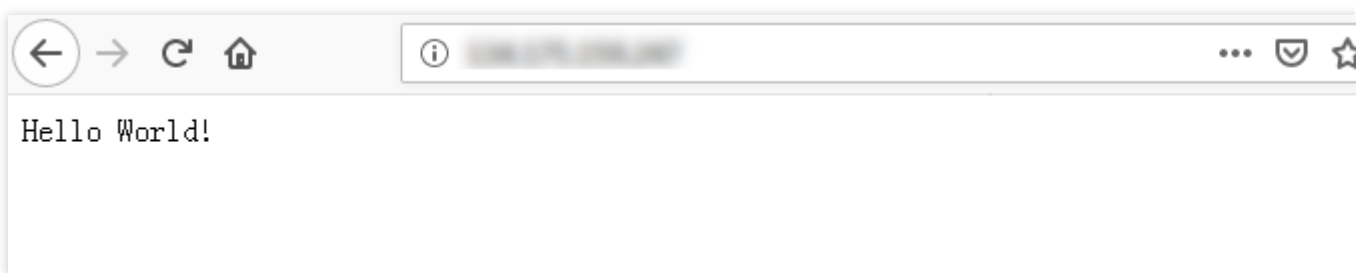
5. ブラウザに、helloworldサービスのCLB IPを貼り付けます。

### サービスの名前でHello Worldサービスにアクセス

クラスター内のその他サービスやコンテナは、直接サービス名を介してアクセスできます。

### Hello Worldサービス検証

サービスにアクセスする時、次のように表示されれば、Hello Worldサービスが作成されたことになります。



コンテナの作成に失敗した場合は、[イベントに関するよくあるご質問](#)をご参照ください。

# 単一インスタンス版のWordPress

最終更新日：2022-04-07 11:10:35

## 操作場面

WordPressは、PHP言語を使用して開発されたブログプラットフォームです。ユーザーは、PHPとMySQLデータベースをサポートするサーバーに、自分のウェブサイトを設置することができます。また、WordPressをコンテンツ管理システム(CMS)として使用することも可能です。

このドキュメントはDocker Hubの公式イメージ `wordpress` で公的にアクセス可能なWordPressウェブサイトを作成する方法を紹介します。

## 前提条件

注意：

- `wordpress` このイメージにはWordPressのすべての運行環境が含まれているので、直接プルしてサービスを作成すればいいです。
- 単一インスタンスバージョンのWordPressを作成するのはテスト用のみです。データの持続的な保存は保証できません。自作のMySQLやTencentDBでデータを保存することをお勧めします。詳細は [TencentDBを使用しているWordPress](#) をご参照ください。

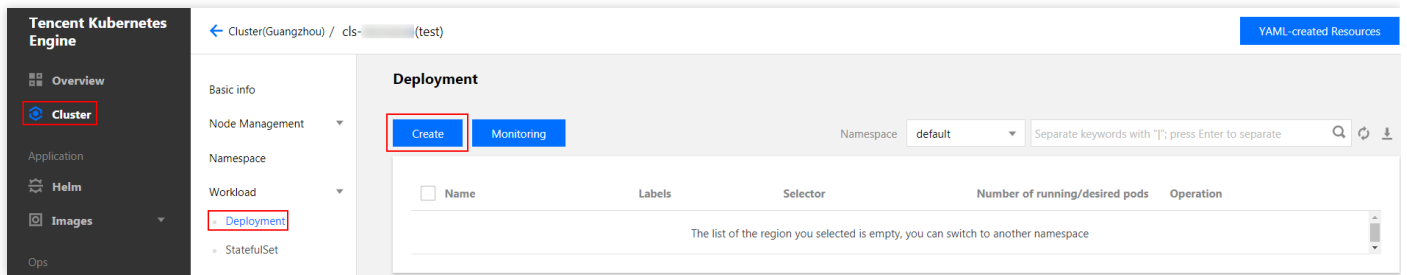
- [Tencent Cloudアカウント登録済み](#)です。

- クラスターが作成されました。クラスターの作成については、[クラスター作成](#)をご参照ください。

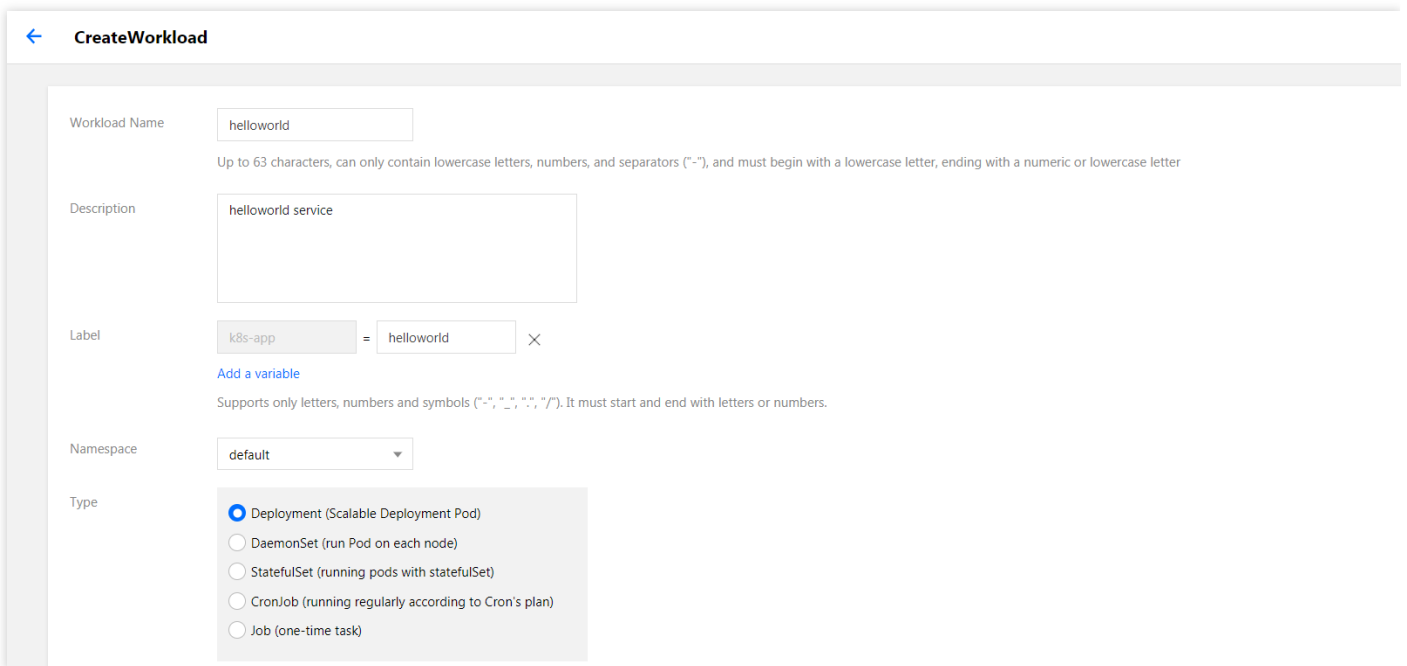
## 操作手順

### WordPressサービス作成

- Tencent Kubernetes Engineコンソールにログインし、左側のナビゲーションバーで\*\* [Cluster](#) \*\*を選択します。
- 「クラスター管理」ページでサービスを作成する必要があるクラスターIDを選択し、クラスターのロード「Deployment」ページに入って、\*\* [Create](#) \*\*をクリックします。下図の通りです：



3. 「Create Workload」 ページで次の情報によって、ロードの基本情報を設定します。下図の通りです：



- **ロード名**：作成するロードの名前を入力し、本文はwordpressを例として説明します。
- **説明**：ロードについての情報を記入します。
- **タグ**：key = value、この例ではタグのデフォルト値はk8s-app = **wordpress**です。
- **ネームスペース**：実際の必要に応じて選択します。
- **タイプ**：実際の必要に応じて選択します。
- **データボリューム**：実際の必要に応じてロードのマウントを設定します。詳細は[Volume管理](#)をご参照ください。

4. 次の提示によって、**インスタンスコンテナ**を設定します。

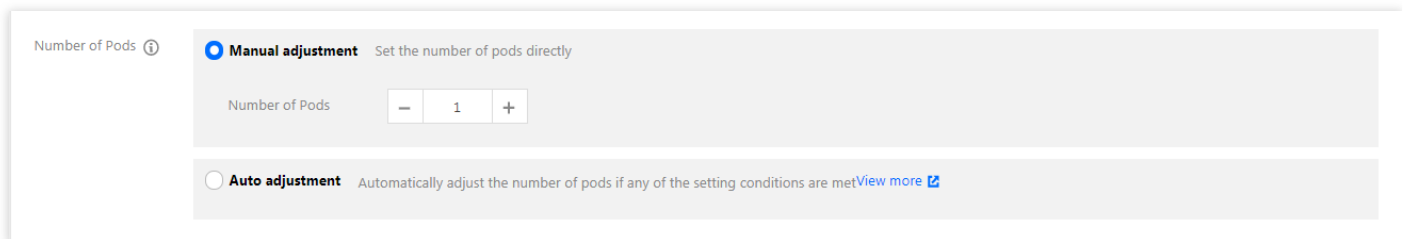
主なパラメータ情報は下記の通りです。他のオプションはデフォルト値を保ちます：

- **名前**：カスタマイズコンテナの名前を入力し、本文はtestを例として説明します。
- **イメージ**： `wordpress` を入力します。
- **イメージバージョン (Tag)**：latestを入力します。



- **イメージプルポリシー**：次の三つのポリシーを提供し、必要に応じて選択してください。本文は設定しなくてデフォルトポリシーを使用するのを例として説明します。  
イメージプルポリシーを設定しなくて、イメージバージョンはnullや `latest` の時、**Always**ポリシーを使用し、そうでなければ**IfNotPresent**ポリシーを使用します。
  - **Always**：いつもリモートでこのイメージをプルします。
  - **IfNotPresent**：ローカルイメージをデフォルトで使用し、ローカルにこのイメージがない場合、リモートでこのイメージをプルします。
  - **Never**：ローカルイメージだけを使用し、ローカルにこのイメージがない場合、異常を報告します。

6. 次の提示によって、サービスするインスタンス数量を設定します。下図の通りです：



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](#) ⓘ

- **手動調整**：インスタンス数量を設定し、本文のインスタンス数量は1に設定します。「+」や「-」をクリックして、インスタンス数量をコントロールします。
- **自動調整**：いずれかの設定条件を満たせば、インスタンス (pod) 数量を自動的に調整します。詳細は[サービスの自動的なスケールアウトとスケールイン](#)をご参照ください。

8. 次の提示によって、ロードの**アクセス設定 (Service)** をします。

- **Service**：「使用」を選択します。
- **サービスのアクセス方式**：「パブリックネットワークLBでアクセス」を選択します。
- **Cloud Load Balancer**：実際の必要に応じて選択します。
- **ポートマッピング**：TCPプロトコルを選択し、コンテナポートとサービスポートを80に設定します。

注意：

サービスのあるクラスターのセキュリティグループはノードネットワークとコンテナネットワークをインターネットにオープンしてください。同時に、30000 - 32768ポートをインターネットにオープンする必要もありますが、そうでなければTencent Kubernetes Engineが使用できない問題が発生する場合があります。詳細は[Tencent Kubernetes Engineセキュリティグループ設定](#)をご参照ください。

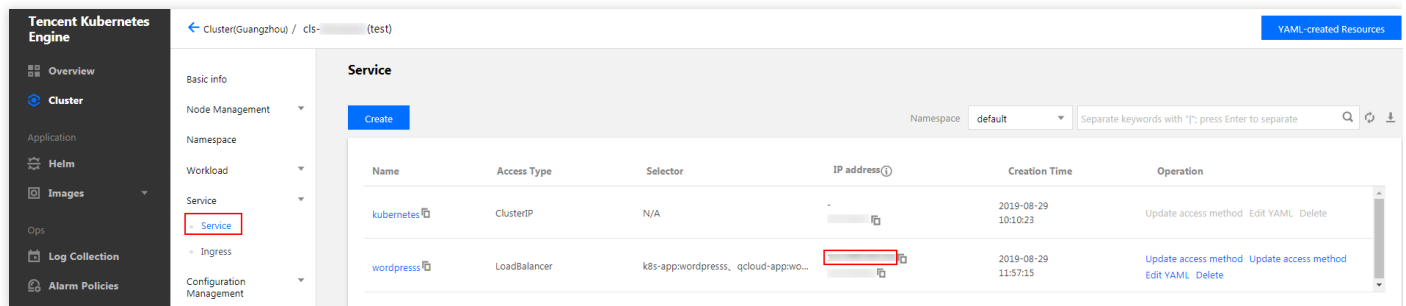
9. **Create Workload**をクリックし、wordpressサービスの作成を完了します。

## WordPressサービスにアクセス

次の二つの方式でWordPressサービスにアクセスできます。

### Cloud Load Balancer IPでWordPressサービスにアクセス

1. 左側ナビゲーションバーで\*\* Cluster\*\*をクリックして、「クラスター管理」ページに入ります。
2. WordPressサービスのあるクラスターIDをクリックして、**Services and Routes > Service**を選択します。
3. サービス管理のページに入って、WordPressサービスのCloud Load Balancer IPをコピーして、次の通りです。



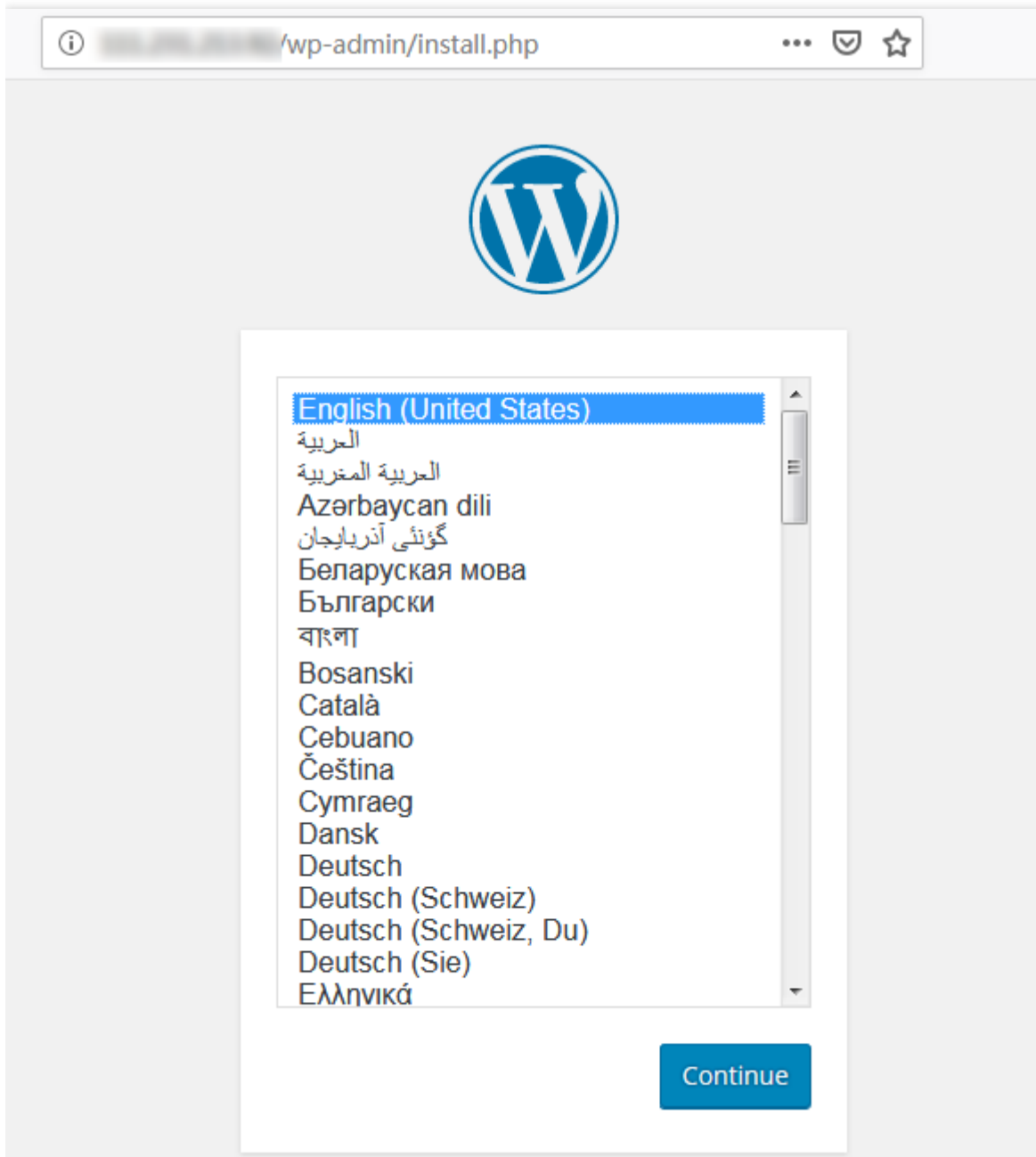
4. ブラウザのアドレスバーにCloud Load Balancer IPを入力し、「**Enter**」を押すと、サービスにアクセスできます。

### サービスの名前でサービスにアクセス

クラスター内の他のサービスやコンテナはサービスの名前で直接にアクセスできます。

## WordPressサービス検証

サービスは成功的に作成されました。サービスにアクセスする時、直接にWordPressサーバーの構成ページに入ります。次の通りです。



## その他のWordPress設定

コンテナの作成に失敗した場合、[よくある質問](#)をご参照ください。

# TencentDBを用いたWordPress

最終更新日：：2022-04-07 11:10:35

## 操作場面

[単一インスタンスバージョン WordPress](#) の例は迅速にWordPressサービスを作成する方法を示します。この方法で作成されたWordPressサービスの特徴は次の通りです：

- データは同じコンテナが運行されているMySQL DBに書き込みます。
- サービスは迅速に起動することができます。
- コンテナがある原因で停止すれば、DBとストレージファイルは失います。

MySQL DBでデータを永遠にストレージすることができます。DBはインスタンス／コンテナが再び起動した後、存在し続けます。このドキュメントは[TencentDB](#)でMySQL DBを設定する方法とTencentDBのWordPressサービスを作成して使用する方法を紹介します。

## 前提条件

- [Tencent Cloudアカウント登録済み](#)です。
- クラスターが作成されました。クラスターの作成については、[クラスター作成](#)をご参照ください。

説明：

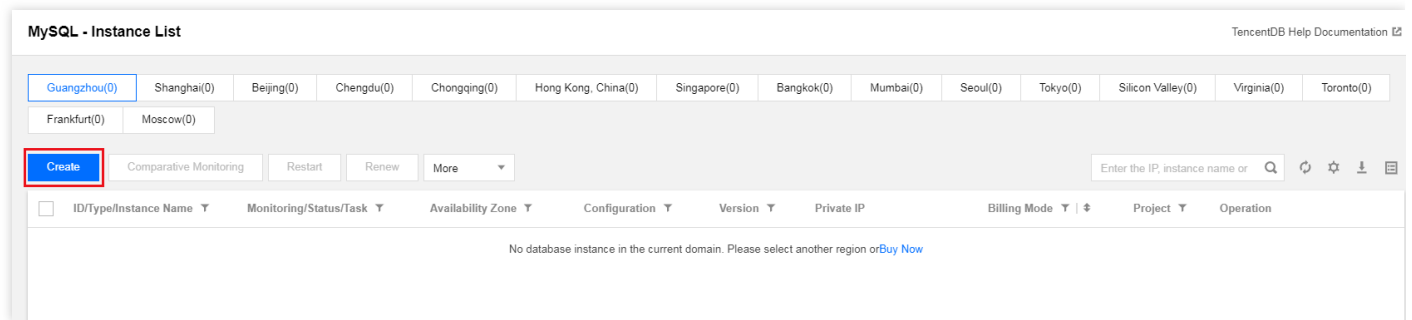
本文が使用されているDBは[MySQL](#)です。

## 操作手順

### WordPressサービス作成

#### TencentDB作成

1. [MySQLコンソール](#)にログインし、DBインスタンスリストの上で【Create】をクリックします。下図の通りです：



2. 購入する構成を選択します。詳細は[MySQL](#)をご参照ください。

注意：

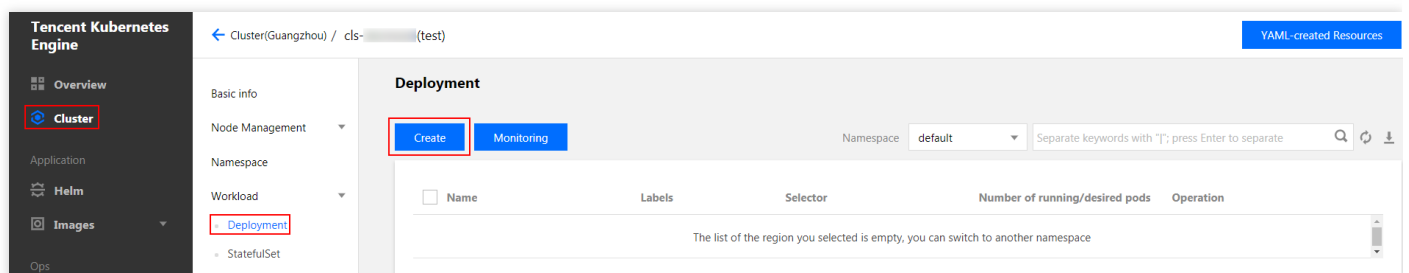
TencentDBのある地域はクラスターと同じです。そうでなければこのDBに接続できません。

3. DBが成功的に作成された後、[MySQL-インスタンスリスト](#)で確認できます。

4. DBについて初期化操作を行います。詳細は[MySQLの初期化](#)をご参照ください。

## TencentDBを使用するWordPressサービス作成

1. Tencent Cloud Tencent Kubernetes Engine コンソールにログインし、左側のナビゲーションバーで【[Cluster](#)】を選択します。
2. 「クラスター管理」ページでサービスを作成する必要があるクラスターIDを選択し、クラスターのロード「Deployment」ページに入って、【[Create](#)】をクリックします。下図の通りです：



3. 「Workloadの新規作成」 ページで次の情報によって、ロードの基本情報を設定します。下図の通りです：

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

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

- **ロード名：**作成するロードの名前です。本文はwordpressを例として説明します。
- **説明：**ロードについての情報を記入します。
- **タグ：**key = value、この例ではタグのデフォルト値がk8s-app = **wordpress**です。
- **ネームスペース：**必要に応じて選択します。
- **タイプ：**必要に応じて選択します。
- **ボリューム：**必要に応じてロードのマウントを設定します。詳細は[Volume管理](#)をご参照ください。

4. 次の情報を参照して「インスタンスコンテナ」を設定します。次の通りです。

The screenshot shows the 'Containers in the pod' configuration window. It includes the following fields and values:

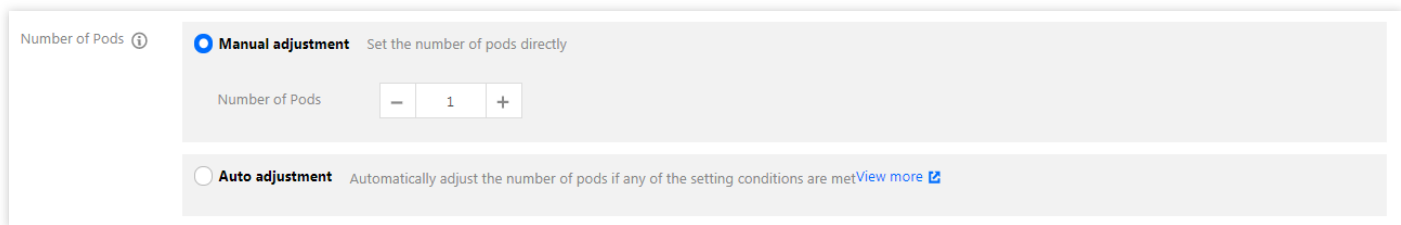
- Name:** test
- Image:** wordpress
- Image Tag:** latest
- CPU/memory limit:**
  - CPU Limit:** request 0.25, limit 0.5 core
  - Memory Limit:** request 256, limit 1024 MiB
- GPU limit:** 0
- Environment Variable:**
  - WORDPRESS\_DB\_HOST = [redacted]
  - WORDPRESS\_DB\_PASSWORD = [redacted]

At the bottom, there is a note: 'Note: After Workload is created, the container configuration information can be modified by updating YAML.' and an 'Add Container' button.

主なパラメータ情報は下記の通りです。他のオプションはデフォルト設定を保ちます：

- **名前：**カスタマイズコンテナの名前を入力し、本文はtestを例として説明します。
- **イメージ：** `wordpress` を入力します。
- **イメージバージョン (Tag)：** `latest` を入力します。
- **イメージプルポリシー：** 次の三つのポリシーを提供し、必要に応じて選択します。本文は設定しなくてデフォルトポリシーを使用するのを例として説明します。  
イメージプルポリシーを設定しなくて、イメージバージョンはnullやlatestの時、**Always**ポリシーを使用し、そうでなければ**IfNotPresent**ポリシーを使用します。
  - **Always：** いつもリモートでこのイメージをプルします。
  - **IfNotPresent：** ローカルイメージをデフォルトで使用し、ローカルにこのイメージがない場合、リモートでこのイメージをプルします。
  - **Never：** ローカルイメージだけを使用し、ローカルにこのイメージがない場合、異常を報告します。
- **環境変数：** 次の構成情報を順番に入力します：
  - WORDPRESS\_DB\_HOST = MySQLのプライベートネットワークIP
  - WORDPRESS\_DB\_PASSWORD = 初期化の時、記入されたパスワード

5. 次の提示によって、サービスするインスタンス数量を設定します。下図の通りです：



Number of Pods ⓘ

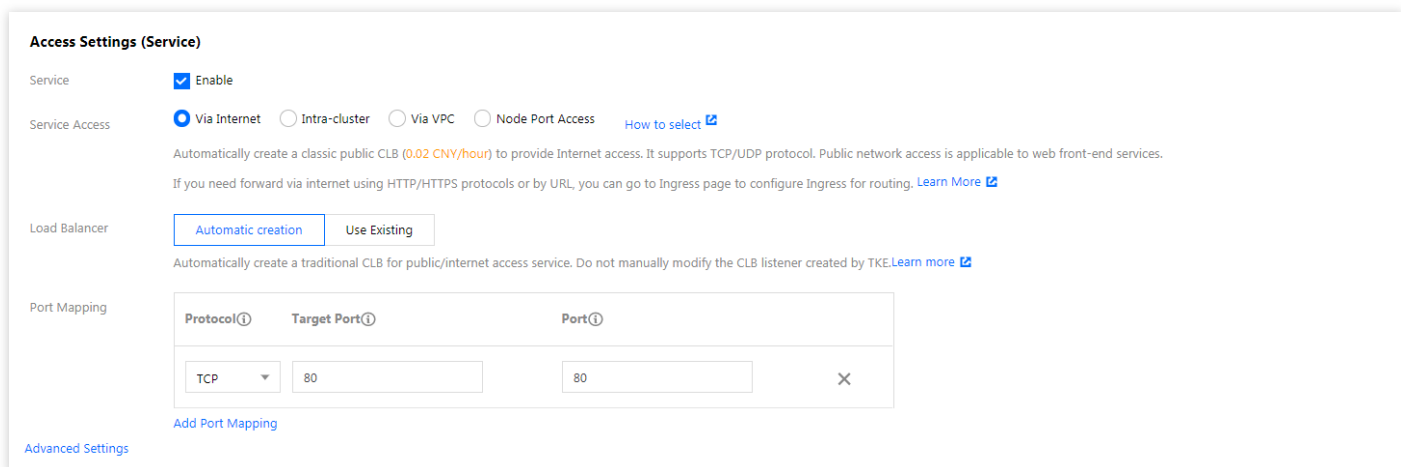
**Manual adjustment** Set the number of pods directly

Number of Pods

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

- **手動調整**：インスタンス数量を設定し、本文のインスタンス数量は1に設定します。「+」や「-」をクリックして、インスタンス数量をコントロールします。
- **自動調整**：いずれかの設定条件を満たせば、インスタンス（pod）数量を自動的に調整します。詳細は[サービスの自動的なスケールアウトとスケールイン](#)をご参照ください。

6. 次の提示によって、ロードのアクセス設定をします。下図の通りです：



**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  Automatic creation  Use Existing

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**：「使用」を選択します。
- **サービスのアクセス方式**：「パブリックネットワークを提供してアクセス」を選択します。
- **Cloud Load Balancer**：必要に応じて選択します。
- **ポートマッピング**：TCPプロトコルを選択し、コンテナポートとサービスポートを80に設定します。

注意：

サービスのあるクラスターのセキュリティグループはノードネットワークとコンテナネットワークをインターネットにオープンしてください。同時に、30000 - 32768ポートをインターネットにオープンする必要がありますが、そうでなければTencent Kubernetes Engineが使用できない問題が発生する場合があります。詳細は[Tencent Kubernetes Engineセキュリティグループ設定](#)をご参照ください。



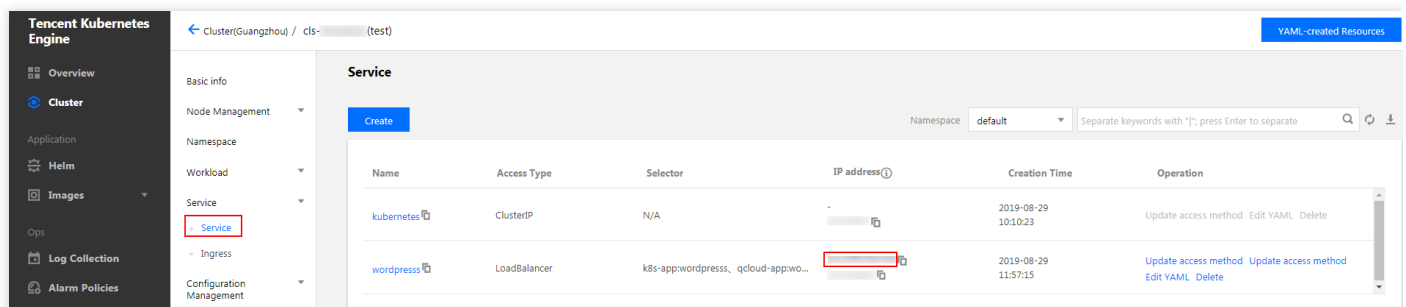
7. 【Create workload】をクリックして、WordPressサービスの作成を完了します。

## WordPressサービスにアクセス

次の二つの方式でWordPressサービスにアクセスできます。

### Cloud Load Balancer IPでWordPressサービスにアクセス

1. 左側ナビゲーションバーで【Clusters】をクリックして、「クラスター管理」ページに入ります。
2. WordPressサービスのあるクラスターIDをクリックして、【Service】>【Service】を選択します。
3. サービス管理のページに入って、WordPressサービスのCloud Load Balancer IPをコピーして、下図の通りです：



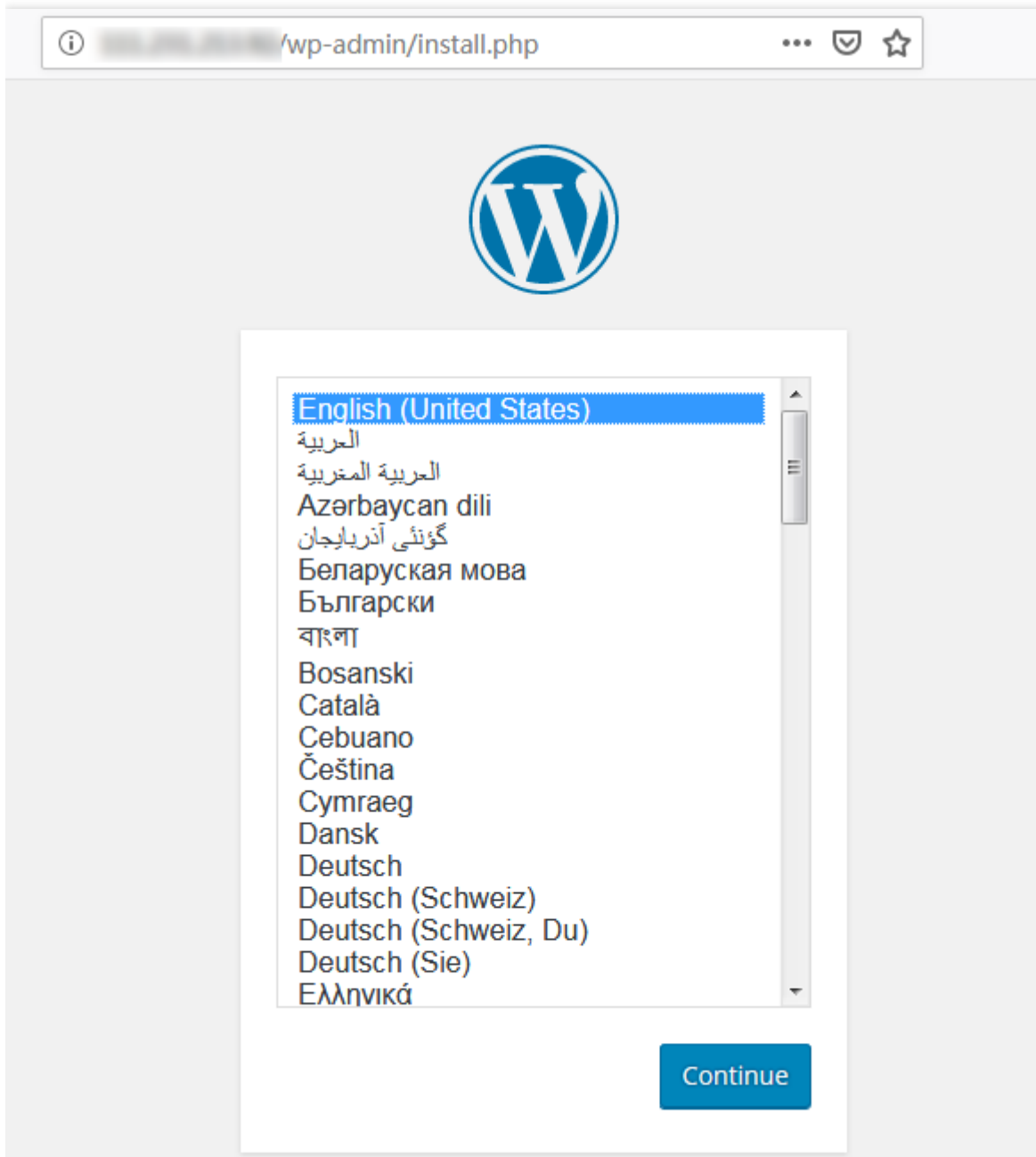
4. ブラウザのアドレスバーにCloud Load Balancer IPを入力し、**Enter**を押すと、サービスにアクセスできます。

### サービスの名前でサービスにアクセス

クラスター内の他のサービスやコンテナはサービスの名前で直接にアクセスできます。

## WordPressサービス検証

サービスは成功的に作成されました。サービスにアクセスする時、直接にWordPressサーバーの構成ページに入ります。下図の通りです：



## その他のWordPress設定

コンテナの作成に失敗した場合、[よくある質問](#)をご参照ください。

# 簡易Webアプリケーションの構築

最終更新日：：2022-12-14 16:43:12

## 運用シーン

ここでは、Tencent CloudのTKEを使用して簡単なWebアプリケーションを作成する方法をご紹介します。

Webアプリケーションは以下の2部分から構成されます：

- フロントエンドサービス。クライアント側の検索や書き込みリクエストの処理に使用されます。
- データベースストレージサービス。redisを使用します。書き込んだデータをredis-masterに保存し、redis-slaveにアクセスして読み取る処理を行います。redis-master と redis-slave はマスター/スレーブレプリケーションによりデータを同期します。

このアプリケーションはkubernetes固有のサンプルです。詳しくは、[Guestbook App](#)をご参照ください。

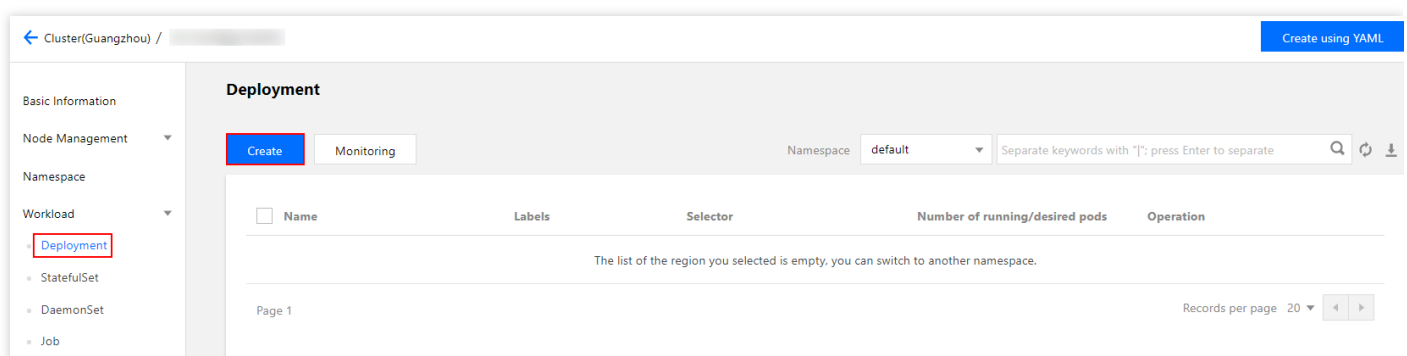
## 前提条件

- [Tencent Cloudアカウントの登録](#)が完了していること。
- クラスタが作成されていること。作成されていない場合、[クラスタ作成](#)をご参照ください。

## 操作手順

### redis-masterサービスの作成

1. TKEコンソールにログインし、左側のナビゲーションバーの **Cluster** を選択します。
2. アプリケーションを作成するクラスターIDをクリックし、ワークロードDeployment画面に入って、\*\* Create\*\* を選択します。下図の通りです：



3. [Create Workload] 画面で、以下の案内に従って、ワークロードの基本情報を設定します。下図の通りです：

The screenshot shows the 'Create Workload' interface. The 'Workload Name' field contains 'redis-master'. The 'Description' field is empty. The 'Tag' field shows 'k8s-app = redis-master'. The 'Namespace' dropdown is set to 'default'. The 'Type' section has 'Deployment (Scalable Deployment Pod)' selected. The 'Volume (optional)' section has an 'Add Volume' link.

- **ワークロード名**：作成するワークロードの名前。本書では、**redis-master**とします。
- **詳細**：ワークロードの関連情報を記入します。
- **タグ**：key = value キーと値のペア。本例では、タグのデフォルト値は、**k8s-app = redis-master**とします。
- **ネームスペース**：必要に応じて選択してください。
- **タイプ**：必要に応じて選択してください。
- **データボリューム**：必要に応じて、ワークロードのマウントボリュームを設定してください。詳しくは、[Volume 管理](#)をご参照ください。

4. 以下の案内に従って、**インスタンスコンテナ**を設定します。下図の通りです：

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	request	0.25	-	limit	0.5	-	Memory Limit	request	256	-	limit	1024
						core		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](#)

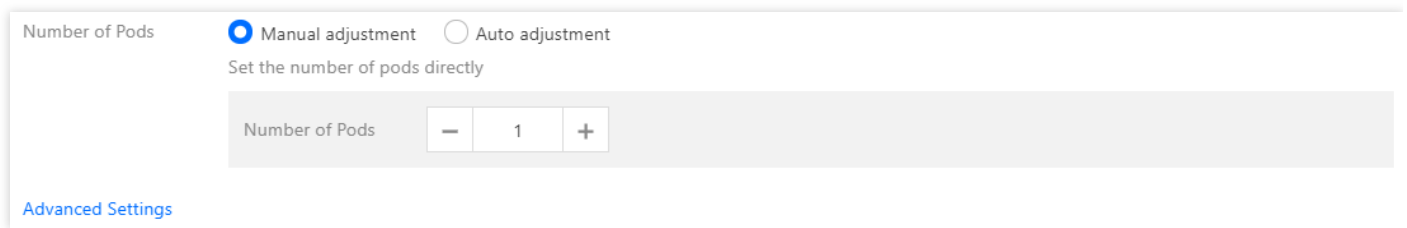
Note: After Workload is created, the container configuration information can be modified by updating YAML.

主なパラメータ情報は下記の通りです。ほかのオプションはデフォルトで構いません。

- **名前**：インスタンスコンテナの名前を入力します。本書では、**master**とします。
- **イメージ**：「**ccr.ccs.tencentyun.com/library/redis**」を入力します。
- **イメージのバージョン (Tag)**：**latest**を入力します。
- **イメージ取得ポリシー**：以下の3ポリシーが用意されているため、必要に応じて選択してください。本書では、デフォルトポリシーの適用を例とします。  
イメージ取得ポリシーを設定しない際、イメージのバージョンが空白または**latest**の場合、**Always**ポリシーを適用します。そうでない場合、**IfNotPresent**ポリシーを適用します。
  - **Always**：常にリモートからイメージを取得します。
  - **IfNotPresent**：デフォルトではローカルイメージを使用します。ローカルイメージがない場合、リモートからイメージを取得します。

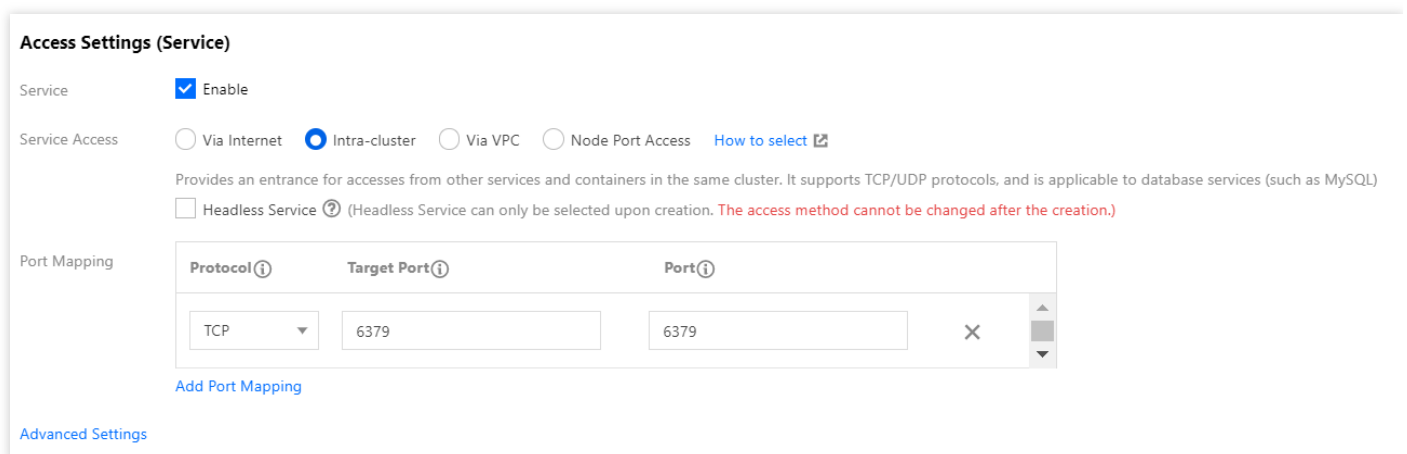
- **Never** : ローカルイメージだけを使用します。ローカルイメージがない場合、エラーを出力します。

6. インスタンス数を設定します。詳しくは下図の通りです：



- **手動調整** : インスタンス数を設定します。本書では、インスタンス数に1を設定します。「+」または「-」をクリックしてインスタンス数を設定することができます。
- **自動調整** : いずれかの設定条件を満たした場合、自動的にインスタンス (pod) 数を調整します。詳しくは、[サービスの自動拡張/縮小](#)をご参照ください。

7. 以下の案内に従って、ワークロードのアクセス設定を実施します。下図の通りです：

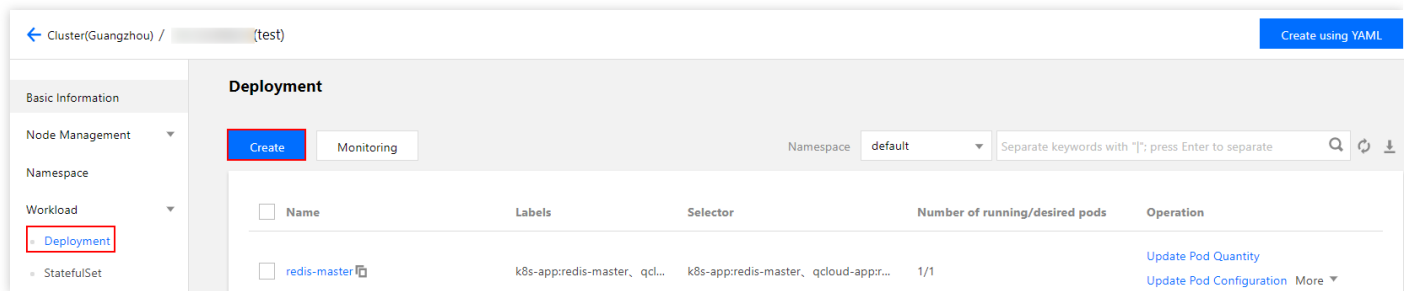


- **Service** : 「有効」にチェックを入れます。
- **サービスアクセス方式** : 「クラスター内部のアクセス」を選択します。
- **ロードバランサー** : 必要に応じて選択してください。
- **ポートマッピング** : TCPプロトコルを選択し、サービスポートとコンテナポート両方に6379を設定します。ほかのサービスは、サービス名redis-masterまたはポート6379を介してmasterコンテナにアクセスすることができます。

8. **Create Workload**をクリックし、redis-masterサービスの作成を完了します。

## redis-slaveサービスの作成

1. TKEコンソールにログインし、左側のナビゲーションバーの \*\* Cluster \*\*を選択します。
2. サービスを作成するクラスターIDをクリックし、ワークロードDeployment画面に入って、 **Create** を選択します。下図の通りです：



3. [New Workload] 画面で、以下の案内に従って、ワークロードの基本情報を設定します。下図の通りです：

Workload Name:

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:

Tag:  =  X

Add a variable

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

Namespace:

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](#)

- **ワークロード名**：作成するワークロードの名前。本書では、redis-slaveとします。
- **詳細**：ワークロードの関連情報を記入します。
- **タグ**：key = value キーと値のペア。本例では、タグのデフォルト値は、k8s-app = **redis-slave**とします。
- **ネームスペース**：必要に応じて選択してください。
- **タイプ**：必要に応じて選択してください。
- **データボリューム**：必要に応じて、ワークロードのマウントボリュームを設定してください。詳しくは、[Volume 管理](#)をご参照ください。

4. 以下の案内に従って、**インスタンスコンテナ**を設定します。以下の通りです。

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	request	0.25	-	limit	0.5	-	Memory Limit	request	256	-	limit	1024
	core				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.

主なパラメータ情報は下記の通りです。ほかのオプションはデフォルトでよいです。

- **名前**：インスタンスコンテナの名前を入力します。本書では、**slave**とします。
- **イメージ**：「`ccr.ccs.tencentyun.com/library/gb-redisslave`」を入力します。
- **イメージのバージョン (Tag)**：`latest`を入力します。
- **イメージ取得ポリシー**：必要に応じて選択してください。本書では、デフォルトポリシーの適用を例とします。
- **環境変数**：以下の設定情報を入力します：  
`GET_HOSTS_FROM = dns`



5. インスタンス数を設定します。下図の通りです：

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

Number of Pods

[Advanced Settings](#)

- **手動調整**：インスタンス数を設定します。本書では、インスタンス数に1を設定します。「+」または「-」をクリックしてインスタンス数を設定することができます。
- **自動調整**：いずれかの設定条件を満たした場合、自動的にインスタンス（pod）数を調整します。詳しくは、[サービスの自動拡張/縮小](#)をご参照ください。

6. 以下の案内に従って、ワークロードのアクセス設定を実施します。下図の通りです：

**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](#)

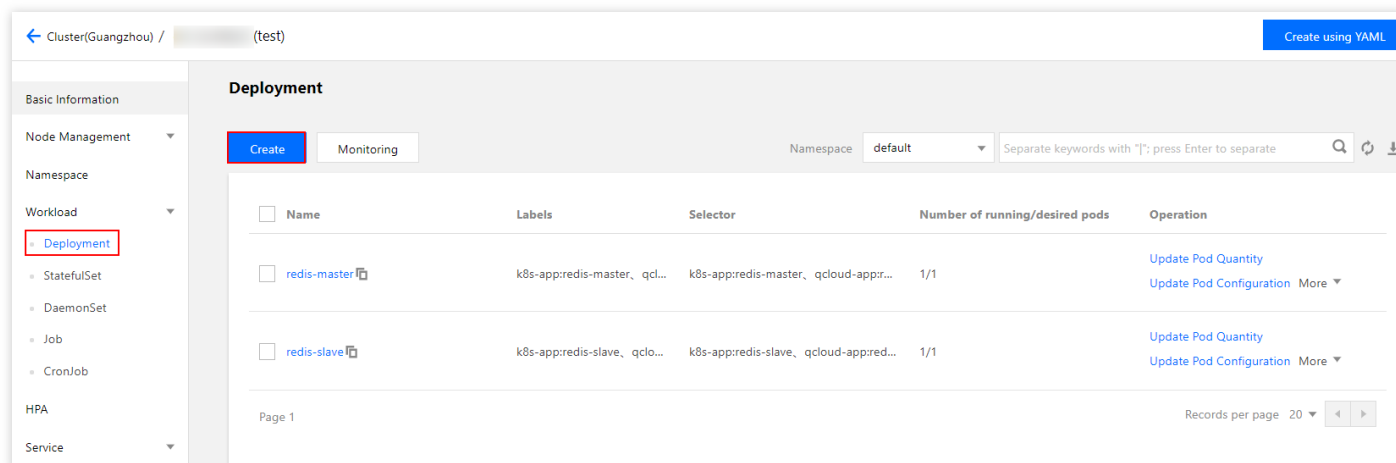
- **Service**：「有効」にチェックを入れます。
- **サービスアクセス方式**：「クラスター内部のアクセス」を選択します。
- **ロードバランサー**：必要に応じて選択してください。
- **ポートマッピング**：TCPプロトコルを選択し、サービスポートとコンテナポート両方に6379を設定します。ほかのサービスは、サービス名redis-masterまたはポート6379を介してmasterコンテナにアクセスすることができます。

7. **Create Workload**をクリックし、redis-slaveサービスの作成を完了します。

## frontendサービスの作成

1. TKEコンソールにログインし、左側のナビゲーションバーの **Cluster** を選択します。

2. アプリケーションを作成するクラスターIDをクリックし、ワークロードDeployment画面に入って、**Create**を選択します。下図の通りです：



3. [New Workload] 画面で、以下の案内に従って、ワークロードの基本情報を設定します。下図の通りです：

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

- Workload Name: frontend
- Description: Up to 1000 characters
- Tag: k8s-app = frontend
- Namespace: default
- Type: Deployment (Scalable Deployment Pod)

Other options for Type include DaemonSet, StatefulSet, CronJob, and Job. The 'Add Volume' link is also visible.

- **ワークロード名**：作成するワークロードの名前。本書では、**frontend**とします。
- **詳細**：ワークロードの関連情報を記入します。
- **タグ**：key = value キーと値のペア。本例では、タグのデフォルト値は、**k8s-app = frontend**とします。
- **ネームスペース**：必要に応じて選択してください。
- **タイプ**：必要に応じて選択してください。
- **データボリューム**：必要に応じて、ワークロードのマウントボリュームを設定してください。詳しくは、[Volume 管理](#)をご参照ください。

5. 以下の案内に従って、**インスタンスコンテナ**を設定します。詳しくは、下図の通りです：

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	request	0.25	-	limit	0.5	-	Memory Limit	request	256	-	limit	1024
	core				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](#)

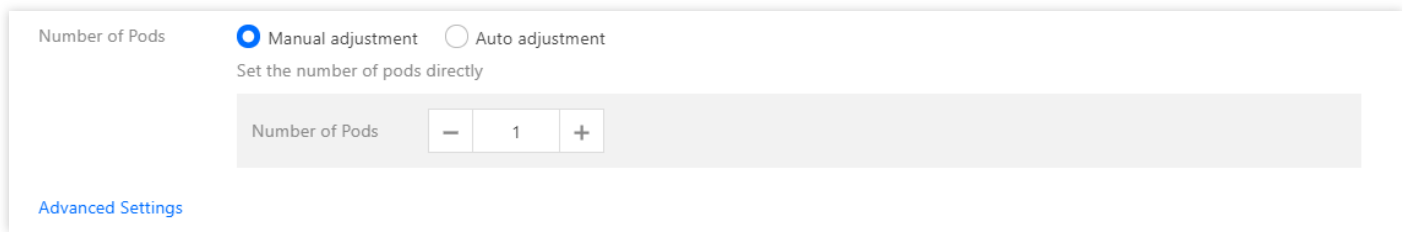
[Add Container](#)

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

主なパラメータ情報は下記の通りです。ほかのオプションはデフォルトで構いません：

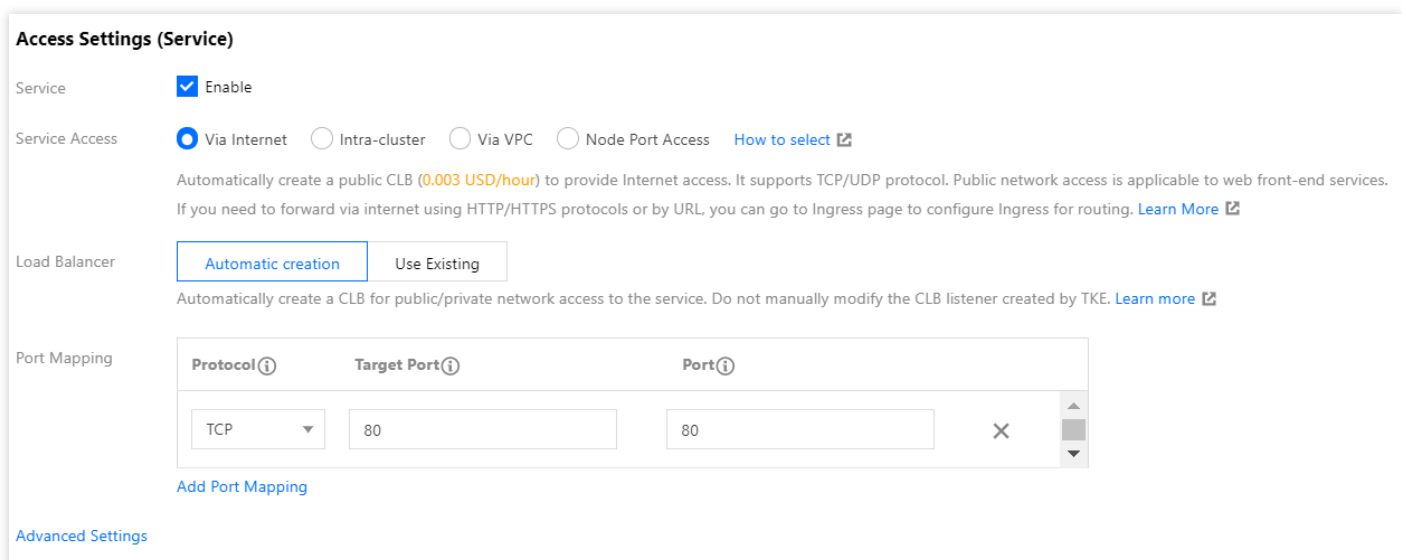
- **名前**：インスタンスコンテナの名前を入力します。本書では、**frontend**とします。
- **イメージ**：「**ccr.ccs.tencentyun.com/library/gb-frontent**」を入力します。
- **イメージのバージョン (Tag)**：**latest**を入力します。
- **イメージ取得ポリシー**：必要に応じて選択してください。本書では、**デフォルトポリシーの適用**を例とします。
- **環境変数**：以下の設定情報を入力します：  
**GET\_HOSTS\_FROM = dns**

6. インスタンス数を設定します。詳しくは下図の通りです：



- **手動調整**：インスタンス数を設定します。本書では、インスタンス数に1を設定します。「+」または「-」をクリックしてインスタンス数を設定することができます。
- **自動調整**：いずれかの設定条件を満たした場合、自動的にインスタンス（pod）数を調整します。詳しくは、[サービスの自動拡張/縮小](#)をご参照ください。

7. 以下の案内に従って、ワークロードのアクセス設定を実施します。下図の通りです：



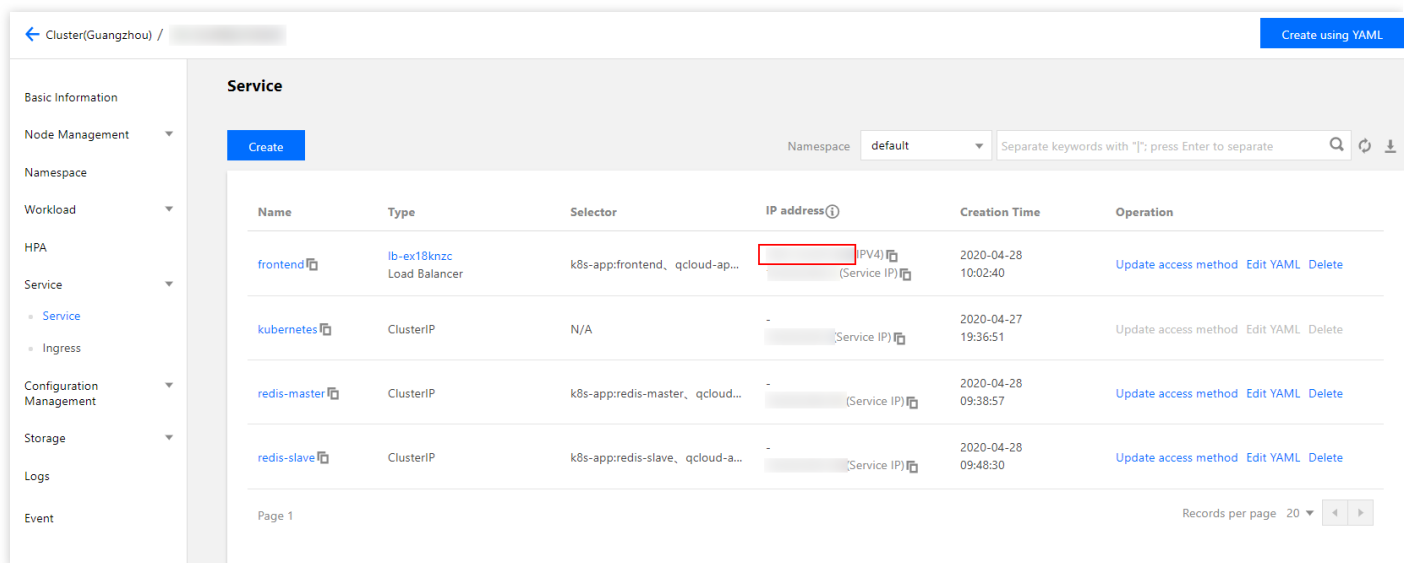
- **Service**：「有効」にチェックを入れます。
- **サービスアクセス方式**：「パブリックネットワークからのアクセスを許可」を選択します。
- **ロードバランサー**：必要に応じて選択してください。
- **ポートマッピング**：TCPプロトコルを選択し、サービスポートとコンテナポート両方に80を設定します。ユーザーはブラウザ経由でロードバランスIPにアクセスすることで、frontendコンテナにアクセスできます。

8. **Create Workload** をクリックし、frontendサービスの作成を完了します。

## Webアプリケーションの確認

1. 左側ナビゲーションバーで、**Cluster** をクリックし、「クラスター管理」画面に入ります。

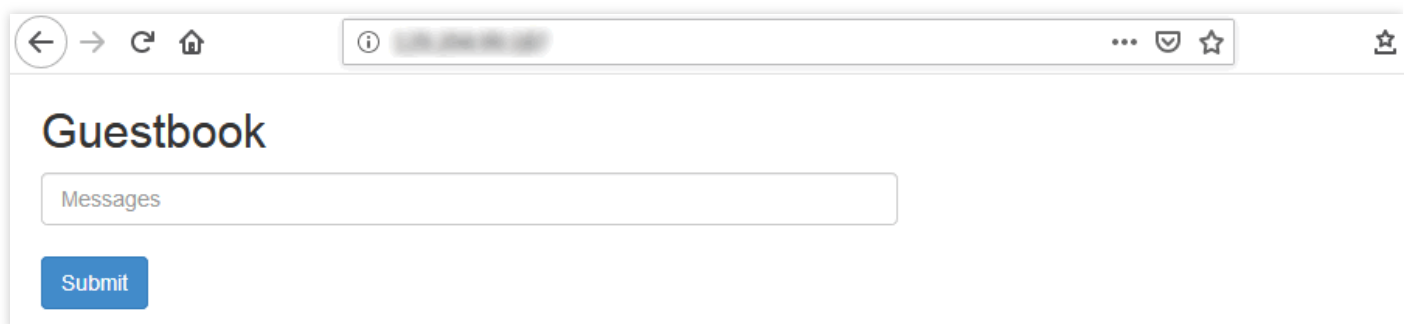
- 作成したサービスが存在するクラスターIDをクリックして、**Services>Service**の順に選択します。下図の通りです
- サービス管理画面に入って、frontendサービスのロードバランスIPをコピーします。下図の通りです：



説明：

- redis-master と redis-slave サービスを作成する時、**クラスター内部でのアクセス**のアクセス方式を設定したため、サービスは1つの内部ネットワークIPだけがあり、クラスター内部のサービスだけがアクセスできます。
- frontendサービスを作成する時、**パブリックネットワークからのアクセスを許可**のアクセス方式を設定したため、サービスはロードバランスIP（すなわち、パブリックネットワークIP）と内部ネットワークIPがあるため、クラスター内部のサービスからのアクセスも可能で、パブリックネットワーク経由でもアクセスできます。

- ブラウザ経由でfrontendサービスのロードバランスIPにアクセスし、下図に示している画面が返されれば、正常にfrontendサービスにアクセスできます。



5. 入力欄に任意の文字列を入力して **Submit** をクリックすると、入力された内容が保存され、画面の下側に表示されます。

ブラウザの画面をリフレッシュして、改めてこのサービスのIPアドレスにアクセスします。入力された内容が依然として存在すれば、入力された文字列がすでにredisに保存されています。

## 開発例

以下のサンプルコードは、Guestbook Appのfrontendサービスの完全なコードです。frontendサービスがHTTPリクエストを受信すると、setコマンドかを判断します：

- setコマンドと判断した場合、パラメータ中のkeyとvalueを取得して、redis-masterサービスに接続し、keyとvalueをredis-masterに設定します。
- setコマンドではないと判断した場合、redis-slaveサービスに接続し、パラメータkeyに対応するvalueの値を取得し、クライアントに返して表示します。

```
<?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: 443,
        ]);
        $client->set($_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: 443,
        ]);
    }
}
```

```
$value = $client->get($_GET['key']);  
print('{"data": "' . $value . '"}');  
}  
}else{  
phpinfo();  
} ?>
```

## 説明事項

- frontendサービスがredis-masterとredis-slaveサービスにアクセスする時、**サービス名とポート**に接続します。クラスター固有のdnsサービスはサービス名を対応するサービスIPに解析し、サービスIPでロードバランシングを実行します。  
例えば、redis-slaveサービスに3つのインスタンスがあるとします。redis-slaveサービスにアクセスする時、そのままredis-slaveと6379に接続します。dnsは自動的にredis-slaveをredis-slaveのサービスIP（すなわち、ロードバランス IPに類似するフローティングIP）に解析し、redis-slaveのサービスIPで自動的にロードバランシングを実行し、リクエストをあるredis-slaveサービスのインスタンスに送信します。
- コンテナ環境変数の設定：
- **デフォルト設定の適用（推奨）**：frontendコンテナが動作時に、取得したGET\_HOSTS\_FROM環境変数の値がdnsの場合、サービス名で接続します。
- **その他の設定**：ほかの環境変数でredis-masterまたはredis-slaveのドメイン名を取得してください。