

# Cloud Container Engine

## Primeiros passos

Edição 01  
Data 2023-11-13



**Copyright © Huawei Cloud Computing Technologies Co., Ltd. 2023. Todos os direitos reservados.**

Nenhuma parte deste documento pode ser reproduzida ou transmitida em qualquer forma ou por qualquer meio sem consentimento prévio por escrito da Huawei Cloud Computing Technologies Co., Ltd.

## **Marcas registadas e permissões**



HUAWEI e outras marcas registadas da Huawei são marcas registadas da Huawei Technologies Co., Ltd. Todas as outras marcas registadas e os nomes registados mencionados neste documento são propriedade dos seus respectivos detentores.

## **Aviso**

Os produtos, os serviços e as funcionalidades adquiridos são estipulados pelo contrato estabelecido entre a Huawei Cloud e o cliente. Os produtos, os serviços e as funcionalidades descritos neste documento, no todo ou em parte, podem não estar dentro do âmbito de aquisição ou do âmbito de uso. Salvo especificação em contrário no contrato, todas as declarações, informações e recomendações neste documento são fornecidas "TAL COMO ESTÃO" sem garantias ou representações de qualquer tipo, sejam expressas ou implícitas.

As informações contidas neste documento estão sujeitas a alterações sem aviso prévio. Foram feitos todos os esforços na preparação deste documento para assegurar a exatidão do conteúdo, mas todas as declarações, informações e recomendações contidas neste documento não constituem uma garantia de qualquer tipo, expressa ou implícita.

## **Huawei Cloud Computing Technologies Co., Ltd.**

Endereço: Huawei Cloud Data Center, Rua Jiaoxinggong  
Avenida Qianzhong  
Novo Distrito de Gui'an  
Guizhou 550029  
República Popular da China

Site: <https://www.huaweicloud.com/intl/pt-br/>

---

# Índice

---

<b>1 Introdução.....</b>	<b>1</b>
<b>2 Preparações.....</b>	<b>3</b>
<b>3 Criação um cluster do Kubernetes.....</b>	<b>7</b>
<b>4 Criação de uma Implementação (Nginx).....</b>	<b>11</b>
<b>5 Implementação de WordPress e MySQL que dependem um do outro.....</b>	<b>17</b>
5.1 Visão geral.....	17
5.2 Criação de uma carga de trabalho do MySQL.....	18
5.3 Criação de uma carga de trabalho do WordPress.....	22
<b>6 Implementação de WordPress usando o Helm.....</b>	<b>31</b>

# 1 Introdução

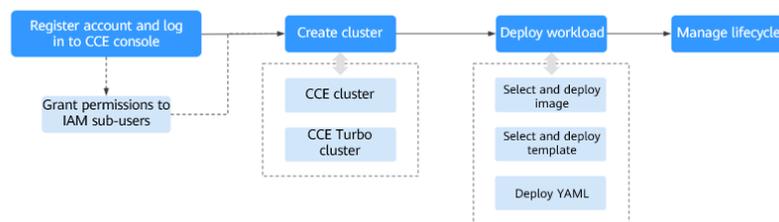
Esta seção descreve como usar o Cloud Container Engine (CCE) e fornece perguntas frequentes (FAQs) para ajudá-lo a começar rapidamente a usar o CCE.

Você pode usar o CCE usando o [console](#), [kubectl \(ferramenta de linha de comando do Kubernetes\)](#) e [as interfaces de programação de aplicações \(APIs\)](#). Para obter detalhes sobre conceitos de CCE, consulte [Conceitos básicos](#).

## Procedimento

Conclua as seguintes tarefas para começar a usar o CCE.

**Figura 1-1** Procedimento para começar com o CCE



### **Passo 1** Registre uma conta do Huawei Cloud e conceda permissões aos usuários do IAM.

As contas da Huawei Cloud têm permissões para usar o CCE. No entanto, os usuários do IAM criados por uma conta da Huawei Cloud não têm a permissão. Você precisa conceder manualmente a permissão aos usuários do IAM. Para obter detalhes, consulte [Visão geral de permissões](#).

### **Passo 2** Crie um cluster.

Para obter detalhes sobre como criar um cluster regular do Kubernetes, consulte [Criação um cluster do Kubernetes](#).

### **Passo 3** Crie uma carga de trabalho a partir de uma imagem ou gráfico.

- [Criação de uma Implementação \(Nginx\)](#)
- [Implementação de WordPress e MySQL que dependem um do outro](#)

**Passo 4** Exiba o status da carga de trabalho e os logs. Atualize, dimensione e monitore a carga de trabalho.

Para obter detalhes, consulte [Gerenciamento de cargas de trabalho e tarefas](#).

---Fim

## Perguntas frequentes

1. **O CCE é adequado para usuários que não estão familiarizados com o Kubernetes?**

Sim. O console do CCE é fácil de usar, e o guia *Primeiros passos* ajuda você a entender e usar rapidamente o CCE.

2. **Onde posso encontrar as instruções sobre como criar imagens do Docker para aplicações tradicionais?**

Para obter detalhes, consulte [Containerização de uma aplicação empresarial \(ERP\)](#) nas práticas recomendadas do CCE.

3. **O CCE é adequado para usuários que têm pouca experiência na construção de imagens?**

Além de armazenar imagens criadas por você em **My Images**, o CCE permite que você crie aplicações em contêiner usando imagens de código aberto. Para mais detalhes, consulte [Criação de uma Implementação \(Nginx\)](#).

4. **Como fazer para criar uma carga de trabalho usando o CCE?**

Crie um cluster e, em seguida, crie uma carga de trabalho no cluster. Para mais detalhes, consulte [Criação de uma Implementação \(Nginx\)](#).

5. **Como criar uma carga de trabalho acessível a redes públicas?**

O CCE fornece diferentes tipos de acesso à carga de trabalho para abordar diversos cenários. Para obter detalhes, consulte [Visão geral](#).

6. **Como permitir que várias cargas de trabalho no mesmo cluster acessem umas às outras?**

Selecione o tipo de acesso ClusterIP, que permite que cargas de trabalho no mesmo cluster usem seus nomes de domínio internos do cluster para acessar uns aos outros.

Os nomes de domínio interno do cluster estão no formato de <nome de serviço autodefinido>.<espaço de nome da carga de trabalho>.svc.cluster.local:<número da porta>. Por exemplo, nginx.default.svc.cluster.local:80.

Exemplo:

Suponha que a carga de trabalho A precise acessar a carga de trabalho B no mesmo cluster. Em seguida, você pode criar um Serviço **ClusterIP** para a carga de trabalho B. Depois que o Serviço ClusterIP é criado, a carga de trabalho B é alcançável em <nome do serviço autodefinido>.<namespace da carga de trabalho B>.svc.cluster.local:<número da porta>.

# 2 Preparações

---

Antes de usar o CCE, faça as seguintes preparações:

- [Registrar uma HUAWEI ID](#)
- [Recarregar sua conta](#)
- [Criar um usuário do IAM](#)
- [Obter permissões de recursos](#)
- [\(Opcional\) Criar uma VPC](#)
- [\(Opcional\) Criar um par de chaves](#)

## Registrar uma HUAWEI ID

Se você já tiver uma HUAWEI ID, pule este passo. Se você não tiver uma, proceda da seguinte forma:

1. Faça login no [site oficial da Huawei Cloud](#) e clique em **Register** no canto superior direito.
2. Na página exibida, registre uma conta conforme solicitado.  
Após o registro, o sistema redireciona você automaticamente para sua página de informações pessoais.

## Recarregar sua conta

Certifique-se de que sua conta tenha saldo suficiente. Para obter detalhes sobre as informações de definição de preço do CCE, consulte [Detalhes de preços de produto](#).

## Criar um usuário do IAM

Se quiser permitir que vários usuários gerenciem seus recursos sem compartilhar sua senha ou chaves, você pode criar usuários usando o IAM e conceder permissões aos usuários. Esses usuários podem usar links especificados e suas próprias contas para acessar a Huawei Cloud e ajudá-lo a gerenciar recursos de forma eficiente. Você também pode configurar políticas de segurança da conta para garantir a segurança dessas contas.

Suas contas têm as permissões para usar CCE. No entanto, os usuários do IAM criados por suas contas não têm as permissões. Você precisa conceder manualmente as permissões aos usuários do IAM. Para obter detalhes, consulte [Visão geral de permissões](#).

## Obter permissões de recursos

O CCE trabalha em estreita colaboração com vários serviços em nuvem para suportar funções de computação, armazenamento, rede e monitoramento. Quando você faz login no console do CCE pela primeira vez, o CCE solicita automaticamente permissões para acessar esses serviços de nuvem na região onde você executa suas aplicações. Especificamente:

- Serviços de computação

Quando você cria um nó em um cluster, um servidor de nuvem é criado de acordo. O pré-requisito é que o CCE obtenha as permissões para acessar o Elastic Cloud Service (ECS) e o Bare Metal Server (BMS).

- Armazenamento

O CCE permite que você monte armazenamento em nós e contêineres em um cluster. O pré-requisito é que o CCE tenha obtido as permissões para acessar serviços como Elastic Volume Service (EVS), Scalable File Service (SFS) e Object Storage Service (OBS).

- Serviços de rede

O CCE permite que os contêineres em um cluster sejam publicados como serviços que podem ser acessados por sistemas externos. O pré-requisito é que o CCE tenha obtido as permissões para acessar serviços como Virtual Private Cloud (VPC) e Elastic Load Balance (ELB).

- Serviços de contêineres e monitoramento

O CCE suporta funções como extração de imagem de contêiner, monitoramento e registro em logs. O pré-requisito é que o CCE tenha obtido as permissões para acessar serviços como o Software Repository for Container (SWR) e Application Operations Management (AOM).

Depois que você concordar em delegar as permissões, uma agência chamada **cce\_admin\_trust** será criada para o CCE no Identity and Access Management (IAM). A conta do sistema **op\_svc\_cce** será delegada a função **Tenant Administrator** para executar operações em outros recursos do serviço de nuvem. O administrador de locatários tem as permissões em todos os serviços de nuvem, exceto o IAM, que chama os serviços de nuvem dos quais o CCE depende. A delegação tem efeito apenas na região atual. Para obter detalhes, consulte [Delegação de conta](#).

Para usar o CCE em várias regiões, solicite permissões de recursos de nuvem em cada região. Você pode acessar o console do IAM, escolher **Agências** e clicar em **cce\_admin\_trust** para exibir os registros de delegação de cada região.

### NOTA

CCE pode falhar a execução como esperado se a função Tenant Administrator não for atribuída. Portanto, não exclua nem modifique a agência **cce\_admin\_trust** ao usar o CCE.

## (Opcional) Criar uma VPC

Uma VPC fornece uma rede virtual isolada, configurável e gerenciável para clusters do CCE.

Antes de criar o primeiro cluster, verifique se uma VPC foi criada. Para obter detalhes, consulte [Criação de uma VPC](#).

Se você já tiver uma VPC disponível, ignore esta etapa.

**Passo 1** Faça login no console de gerenciamento.

**Passo 2** Clique em  no canto superior esquerdo e selecione uma região e um projeto.

**Passo 3** Em **Networking**, clique em **Virtual Private Cloud**.

**Passo 4** Clique em **Create VPC**.

**Passo 5** Na página **Create VPC**, configure os parâmetros conforme solicitado.

Uma sub-rede padrão será criada junto com uma VPC. Você pode clicar em **Add Subnet** para criar mais sub-redes para a VPC.

**Passo 6** Clique em **Create Now**.

----Fim

## (Opcional) Criar um par de chaves

A plataforma de nuvem usa criptografia de chave pública para proteger as informações de logon dos seus nós do CCE. Senhas ou pares de chaves são usados para autenticação de identidade durante o logon remoto nos nós.

- Você precisa especificar o nome do par de chaves e fornecer a chave privada ao efetuar logon nos nós do CCE usando SSH se escolher o modo de logon do par de chaves. Para obter detalhes, consulte [Criação de um par de chaves](#).
- Se você escolher o modo de logon por senha, pule esta tarefa.

### NOTA

Se você quiser criar pods em várias regiões, será necessário criar um par de chaves em cada região. Para obter mais informações sobre regiões, consulte [Regiões e AZs](#).

### Criar um par de chaves no console de gerenciamento

Se você não tiver nenhum par de chaves, crie um no console de gerenciamento. O procedimento é o seguinte:

**Passo 1** Faça logon no console de gerenciamento.

**Passo 2** Clique em  no canto superior esquerdo e selecione uma região e um projeto.

**Passo 3** Sob **Computing**, clique em **Elastic Cloud Server**.

**Passo 4** No painel de navegação à esquerda, escolha **Key Pair**.

**Passo 5** No painel direito, clique em **Create Key Pair**.

**Passo 6** Digite o nome da chave e clique em **OK**.

**Passo 7** Um nome de par de chaves consiste em duas partes: **KeyPair** e quatro dígitos aleatórios. Você pode inserir um nome fácil de lembrar, por exemplo, **KeyPair-xxxx\_ecs**.

**Passo 8** Baixe manualmente ou automaticamente o arquivo de chave privada. O nome do arquivo é um nome de par de chaves especificado com um sufixo de **.pem**. Armazene com segurança o arquivo de chave privada. Na caixa de diálogo exibida, clique em **OK**.

 **NOTA**

O arquivo de chave privada pode ser baixado apenas uma vez. Mantenha-o seguro. Ao criar um ECS, forneça o nome do par de chaves desejado. Cada vez que você fizer logon no ECS usando SSH, forneça a chave privada.

**----Fim**

# 3 Criação um cluster do Kubernetes

## Contexto

Esta seção descreve como criar rapidamente um cluster do CCE. Para obter detalhes, consulte [Compra de um cluster do CCE](#). Neste exemplo, as configurações padrão ou simples estão em uso.

## Criar um cluster

**Passo 1** Efetue logon no [console do CCE](#).

- Se você não criou um cluster, clique **Create** sob **CCE cluster** na página do assistente.
- Se você criou um cluster, escolha **Clusters** no painel de navegação e clique em **Buy** ao lado do **CCE cluster**.

**Passo 2** Na página **Configure Cluster**, configure os parâmetros do cluster.

Neste exemplo, a maioria dos parâmetros retém valores padrão. Apenas os parâmetros obrigatórios são descritos. Para mais detalhes, consulte [Tabela 3-1](#).

**Tabela 3-1** Parâmetros para criação de um cluster

Parâmetro	Descrição
<b>Basic Settings</b>	
*Cluster Name	Nome do cluster a ser criado. Um nome de cluster contém de 4 a 128 caracteres que começam com uma letra minúscula e não terminam com um hífen (-). Somente letras minúsculas, dígitos e hífens (-) são permitidos.
*Enterprise Project	Este parâmetro é exibido somente para usuários empresariais que ativaram o Enterprise Project Management.
*Cluster Version	Escolha a versão mais recente.
*Cluster Scale	Número máximo de <b>worker nodes</b> que podem ser gerenciados pelo cluster. Se você selecionar <b>50 nodes</b> , o cluster poderá gerenciar no máximo 50 nós de trabalho.

Parâmetro	Descrição
*High Availability	O valor padrão é <b>Yes</b> .
<b>Network Settings</b>	
*Network Model	Você pode selecionar <b>VPC network</b> ou <b>Tunnel network</b> . Mantenha o valor padrão.
*VPC	VPC onde o cluster será localizado. Se nenhuma VPC estiver disponível, clique em <b>Create VPC</b> para criar uma. Depois que a VPC for criada, clique em atualizar.
*Master Node Subnet	Sub-rede onde os nós principais do cluster estão localizados.
*Container CIDR Block	Bloco CIDR usado pelos contêineres. O valor determina o número máximo de contêineres no cluster. Mantenha o valor padrão.
*IPv4 Service CIDR Block	Bloco CIDR para Serviços usados por contêineres no mesmo cluster para acessar uns aos outros. O valor determina o número máximo de Serviços que você pode criar. O valor não pode ser alterado após a criação. Mantenha o valor padrão.

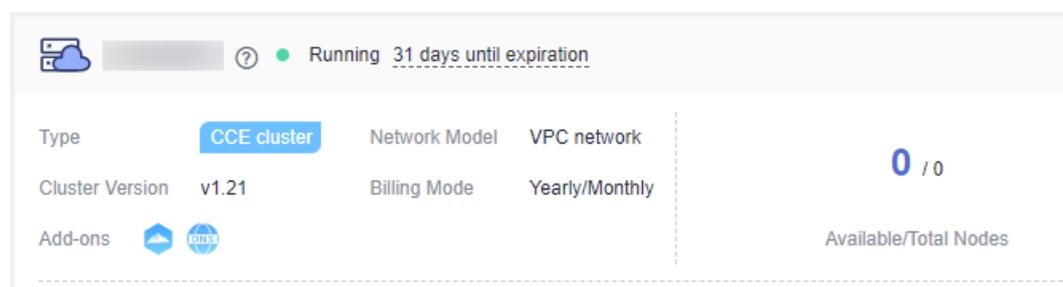
**Passo 3** Clique em **Next: Add-on Configuration**. Mantenha as configurações padrão.

**Passo 4** Clique em **Next: Confirm**. A lista de recursos do cluster é exibida. Confirme as informações e clique em **Submit**.

Demora cerca de 6 a 10 minutos para criar um cluster.

O cluster criado será exibido na página **Clusters** e o número de nós no cluster é 0.

**Figura 3-1** Criação bem-sucedida de clusters



----Fim

## Criar um nó

Depois que um cluster é criado, você precisa criar nós no cluster para executar cargas de trabalho.

**Passo 1** Efetue logon no console do CCE.

**Passo 2** Clique no cluster criado.

**Passo 3** No painel de navegação, escolha **Nodes**. Clique em **Create Node** no canto superior direito e configure os parâmetros do nó.

O que se segue descreve apenas parâmetros importantes. Para outros parâmetros, mantenha os padrões.

#### Compute Settings

- **AZ**: mantenha o valor padrão.
- **Node Type**: selecione **Elastic Cloud Server (VM)**.
- **Specifications**: selecione as especificações do nó que melhor se adequam às suas necessidades de negócios.
- **Container Engine**: selecione um mecanismo de contêiner conforme necessário.
- **OS**: selecione o sistema operacional (SO) dos nós a serem criados.
- **Node Name**: insira um nome de nó.
- **Login Mode**:
  - Se o modo de logon for **Password**: o nome do usuário padrão é **root**. Digite a senha para efetuar logon no nó e confirme a senha.  
Lembre-se da senha de logon do nó. Se você esquecer a senha, o sistema não poderá recuperar sua senha e você terá que redefinir a senha.
  - Se o modo de início de sessão for **Key pair**, selecione um par de chaves para efetuar o registro no nó e selecione a caixa de verificação para confirmar que obteve o arquivo de chave e sem este arquivo não poderá iniciar sessão no nó.  
Um par de chaves é usado para autenticação de identidade quando você entra remotamente em um nó. Se nenhum par de chaves estiver disponível, clique em **Create Key Pair**. Para obter detalhes, consulte [Criação de um par de chaves](#).

#### Storage Settings

- **System Disk**: configure o tipo de disco e a capacidade com base em seus requisitos. A capacidade padrão do disco é de 50 GiB.
- **Data Disk**: configure o tipo de disco e a capacidade com base em seus requisitos. A capacidade padrão do disco é de 100 GiB.

#### Network Settings

- **VPC**: use o valor padrão, ou seja, a sub-rede selecionada durante a criação do cluster.
- **Node Subnet**: selecione uma sub-rede na qual o nó é executado.
- **Node IP**: endereço IP privado do nó. Selecione **Random**.
- **EIP**: permite o acesso à rede pública. Depois que um EIP é vinculado, o nó pode acessar a Internet, por exemplo, baixando imagens de um repositório externo.  
O valor padrão é **Do not use**. Você também pode selecionar **Use existing** ou **Auto create**.

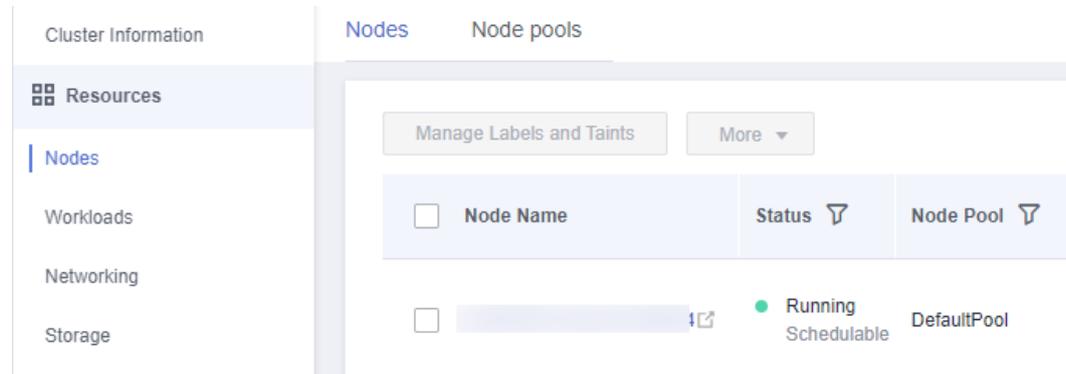
**Passo 4** Na parte inferior da página, selecione a quantidade do nó e o modo de cobrança e clique em **Next: Confirm**.

**Passo 5** Revise as especificações do nó, leia as instruções, selecione **I have read and understand the preceding instructions** e clique em **Submit**.

A adição de um nó leva de 6 a 10 minutos.

O nó criado será exibido na página **Nodes**.

**Figura 3-2** Nó criado com sucesso



----Fim

## Tutorial de vídeo

Você também pode aprender como criar um cluster do Kubernetes por meio do tutorial em vídeo de [criação de um cluster do Kubernetes](#).

# 4 Criação de uma Implementação (Nginx)

---

Você pode usar imagens para criar rapidamente uma carga de trabalho de um único pod que pode ser acessada a partir de redes públicas. Esta seção descreve como usar o CCE para implantar rapidamente uma aplicação de Nginx e gerenciar seu ciclo de vida.

## Pré-requisitos

Você criou um cluster do CCE que contém um nó com 4 vCPUs e 8 GiB de memória. O nó é vinculado a um EIP.

Um cluster é um grupo lógico de servidores em nuvem que executam cargas de trabalho. Cada servidor de nuvem é um nó no cluster.

Para obter detalhes sobre como criar um cluster, consulte [Criação um cluster do Kubernetes](#).

## Visão geral do Nginx

Nginx é um servidor Web leve. No CCE, você pode configurar rapidamente um servidor Web Nginx.

Esta seção usa a aplicação de Nginx como um exemplo para descrever como criar uma carga de trabalho. A criação leva cerca de 5 minutos.

Depois que o Nginx for criado com sucesso, você poderá acessar a página da Web do Nginx.

**Figura 4-1** Página da Web do Nginx

## 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](http://nginx.org).  
Commercial support is available at [nginx.com](http://nginx.com).

*Thank you for using nginx.*

## Criar Nginx no console

A seguir está o procedimento para criar uma carga de trabalho em contêiner a partir de uma imagem de contêiner.

**Passo 1** Efetue login no [console do CCE](#).

**Passo 2** Escolha o cluster de destino.

**Passo 3** No painel de navegação, escolha **Workloads**. Em seguida, clique em **Create Workload**.

**Passo 4** Configure os seguintes parâmetros e mantenha o valor padrão para outros parâmetros:

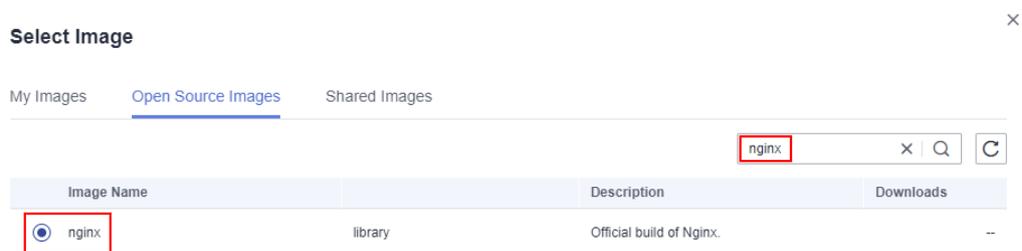
### Basic Info

- **Workload Type:** selecione **Deployment**.
- **Workload Name:** defina-o para **nginx**.
- **Namespace:** selecione **default**.
- **Pods:** defina a quantidade de pods para **1**.

### Container Settings

Na área **Basic Info** de **Container Information**, clique em **Select Image**. Na caixa de diálogo exibida, selecione **Open Source Images**, pesquise por **nginx** e selecione a imagem **nginx**.

**Figura 4-2** Selecionar a imagem de nginx



### Service Settings

Clique no sinal de mais (+) para criar um serviço para acessar a carga de trabalho de uma rede externa. Neste exemplo, crie um Serviço LoadBalancer. Configure os seguintes parâmetros:

- **Service Name:** nome do Serviço exposto a redes externas. Neste exemplo, o nome do Serviço é **nginx**.
- **Access Type:** selecione **LoadBalancer**.
- **Service Affinity:** mantenha o valor padrão.
- **Load Balancer:** se um balanceador de carga estiver disponível, selecione um balanceador de carga existente. Caso contrário, selecione **Auto create** para criar um.
- **Port:**
  - **Protocol:** selecione **TCP**.
  - **Service Port:** defina esse parâmetro como **8080**, que é mapeado para a porta do contêiner.
  - **Container Port:** porta na qual a aplicação escuta. Para contêineres criados usando a imagem de nginx, defina esse parâmetro como **80**. Para outras aplicações, defina este parâmetro para a porta da aplicação.

**Figura 4-3** Criar um Serviço

**Create Service**

Service Name:

Service Type:  ClusterIP  NodePort  LoadBalancer  DNAT

Service Affinity:  Cluster-level  Node-level

Load Balancer:   [Create Load Balancer](#)

Set ELB: Load balancing algorithm: Weighted round robin; Sticky session: Disable; Health check: Disable

I have read [Notes on Using Load Balancers](#).

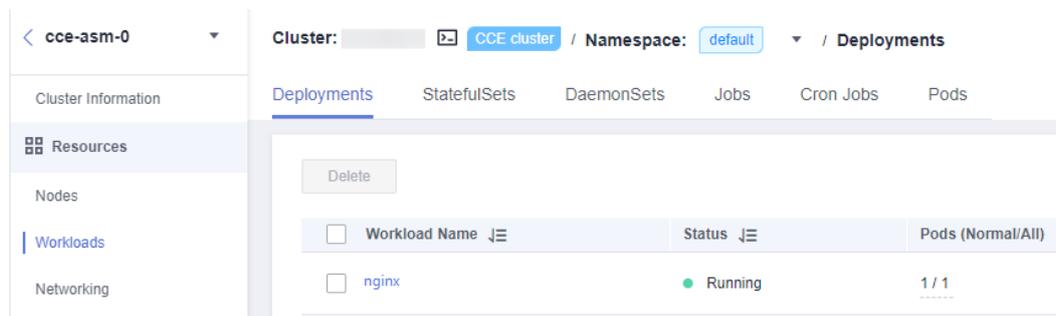
Protocol	Service Port	Container Port	Operation
TCP	8080	80	Delete

**Passo 5** Clique em **Create Workload**.

Aguarde até que a carga de trabalho seja criada.

A Implementação criada será exibida na guia **Deployments**.

**Figura 4-4** Carga de trabalho criada com sucesso



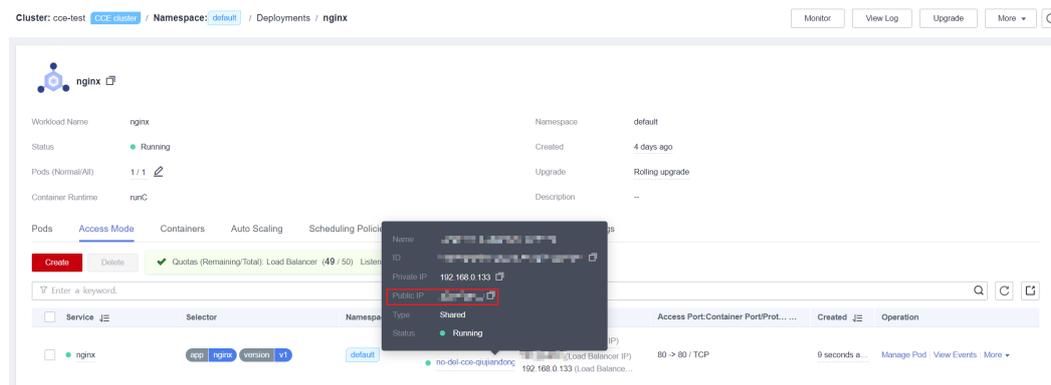
----Fim

## Acessar o Nginx

**Passo 1** Obtenha o endereço de acesso externo do Nginx.

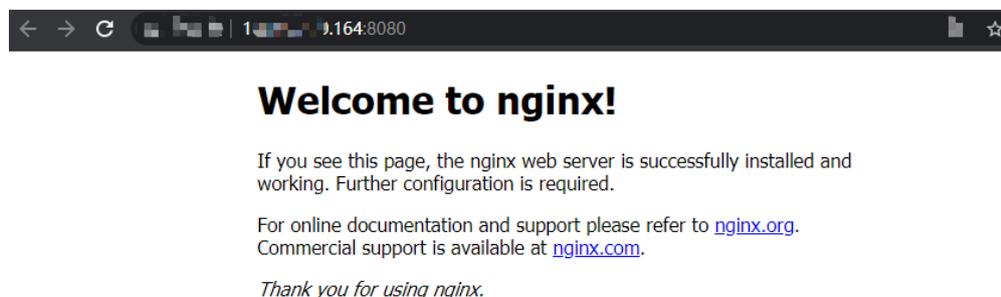
Clique na carga de trabalho do Nginx para entrar na página de detalhes. Na guia **Access Mode**, veja o endereço IP do Nginx. O endereço IP público é o endereço de acesso externo.

Figura 4-5 Obtenção do endereço de acesso externo



**Passo 2** Digite o **endereço de acesso externo** na caixa de endereço de um navegador. O seguinte mostra a página de boas-vindas se você acessar com êxito a carga de trabalho.

Figura 4-6 Acessar o Nginx



----Fim

## Criar o Nginx usando o kubectl

Esta seção descreve como usar o kubectl para criar uma Implementação e expor a Implementação na Internet por meio de um Serviço LoadBalancer.

**Passo 1** Use o kubectl para se conectar ao cluster. Para obter detalhes, consulte [Conexão a um cluster usando o kubectl](#).

**Passo 2** Crie um arquivo de descrição chamado **nginx-deployment.yaml**. **nginx-deployment.yaml** é um nome de arquivo de exemplo. Você pode renomeá-lo conforme necessário.

### vi nginx-deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
```

```
spec:
  containers:
  - image: nginx:alpine
    name: nginx
  imagePullSecrets:
  - name: default-secret
```

**Passo 3** Crie uma Implantação.

**kubectl create -f nginx-deployment.yaml**

Se as seguintes informações forem exibidas, a Implementação está sendo criada.

```
deployment "nginx" created
```

Verifique a Implementação.

**kubectl get deployment**

Se as informações a seguir forem exibidas, a Implementação está sendo executada.

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
nginx	1/1	1	1	4m5s

**Descrição do parâmetro**

- **NAME:** especifica o nome de uma carga de trabalho.
- **READY:** indica o número de pods disponíveis/pods esperados para a carga de trabalho.
- **UP-TO-DATE:** indica o número de réplicas que foram atualizadas.
- **AVAILABLE:** indica o número de pods disponíveis.
- **AGE:** indica o período de execução da Implementação.

**Passo 4** Crie um arquivo de descrição chamado **nginx-elb-svc.yaml**. Altere o valor do **selector** para **matchLabels (app: nginx** neste exemplo) no arquivo **nginx-deployment.yaml** para associar o Serviço à aplicação de back-end.

Para obter detalhes sobre os parâmetros no exemplo a seguir, consulte [Uso do kubectl para criar um Serviço\(Criação automática de um balanceador de carga\)](#).

```
apiVersion: v1
kind: Service
metadata:
  annotations:
    kubernetes.io/elb.class: union
    kubernetes.io/elb.autocreate:
      '{
        "type": "public",
        "bandwidth_name": "cce-bandwidth",
        "bandwidth_chargemode": "bandwidth",
        "bandwidth_size": 5,
        "bandwidth_sharetype": "PER",
        "eip_type": "5_bgp"
      }'
  labels:
    app: nginx
    name: nginx
spec:
  ports:
  - name: service0
    port: 80
    protocol: TCP
    targetPort: 80
  selector:
    app: nginx
  type: LoadBalancer
```

**Passo 5** Crie um Serviço.

**kubectl create -f nginx-elb-svc.yaml**

Se informações semelhantes às seguintes forem exibidas, o Serviço foi criado.

```
service/nginx created
```

**kubectl get svc**

Se informações semelhantes às seguintes forem exibidas, o tipo de acesso foi configurado e a carga de trabalho está acessível.

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.247.0.1	<none>	443/TCP	3d
<b>nginx</b>	<b>LoadBalancer</b>	<b>10.247.130.196</b>	<b>10.78.42.242</b>	<b>80:31540/TCP</b>	<b>51s</b>

**Passo 6** Digite o URL (por exemplo, **10.78.42.242:80**) na caixa de endereço de um navegador. **10.78.42.242** indica o endereço IP do balanceador de carga e **80** indica a porta de acesso exibida no console do CCE.

Nginx é acessível.

**Figura 4-7** Acessar o Nginx através do Serviço LoadBalancer

## 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](http://nginx.org).  
Commercial support is available at [nginx.com](http://nginx.com).

*Thank you for using nginx.*

----Fim

# 5 Implementação de WordPress e MySQL que dependem um do outro

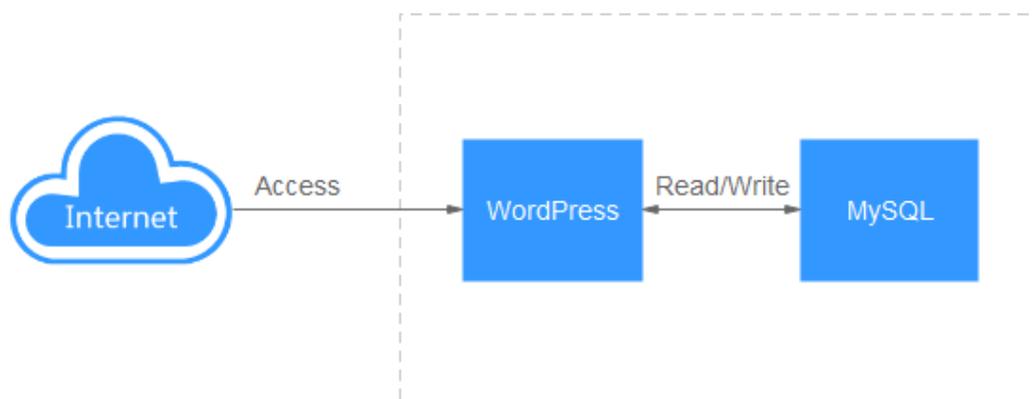
## 5.1 Visão geral

WordPress foi originalmente uma plataforma de blog baseada em PHP e MySQL. É gradualmente evoluído para um sistema de gerenciamento de conteúdo. Você pode configurar seu próprio blog em qualquer servidor que suporte PHP e MySQL. Milhares de plug-ins e inúmeros modelos de tema estão disponíveis para WordPress e fácil de instalar.

WordPress é uma plataforma de blog desenvolvida em pré-processador de hipertexto (PHP). Você pode configurar seus sites nos serviços que suportam bancos de dados PHP e MySQL ou usar o WordPress como um sistema de gerenciamento de conteúdo. Para mais informações sobre o WordPress visite <https://wordpress.org/>.

O WordPress deve ser usado em conjunto com o MySQL. O WordPress executa o programa de gerenciamento de conteúdo, enquanto o MySQL serve como um banco de dados para armazenar dados. Geralmente, o WordPress e o MySQL são executados em contêineres diferentes, conforme mostrado na figura a seguir.

**Figura 5-1** WordPress



Neste exemplo, duas imagens de contêiner estão envolvidas.

- **WordPress:** selecione `wordpress:php7.3` neste exemplo.
- **MySQL:** selecione `mysql:5.7` neste exemplo.

Quando o WordPress acessa o MySQL em um cluster, o Kubernetes fornece um objeto de recurso chamado Serviço para o acesso à carga de trabalho. Neste exemplo, um Serviço é criado para MySQL e WordPress respectivamente. Para obter detalhes sobre como criar e configurar um Serviço, consulte as seções a seguir.

## Tutorial de vídeo

Você também pode aprender como implementar WordPress e MySQL através do tutorial em vídeo de [implementação de WordPress e MySQL](#).

## 5.2 Criação de uma carga de trabalho do MySQL

O WordPress deve ser usado em conjunto com o MySQL. O WordPress executa o programa de gerenciamento de conteúdo, enquanto o MySQL serve como um banco de dados para armazenar dados.

### Pré-requisitos

Você criou um cluster do CCE que contém um nó com 4 vCPUs e 8 GiB de memória. Para obter detalhes sobre como criar um cluster, consulte [Criação um cluster do Kubernetes](#).

### Operações no console

**Passo 1** Efetue login no [console do CCE](#).

**Passo 2** Escolha o cluster de destino.

**Passo 3** No painel de navegação, escolha **Workloads**. Em seguida, clique em **Create Workload**.

**Passo 4** Defina parâmetros de carga de trabalho.

#### Basic Info

- **Workload Type:** selecione **Deployment**.
- **Workload Name:** defina-o para `mysql`.
- **Namespace:** selecione **Default**.
- **Pods:** altere o valor para **1** neste exemplo.

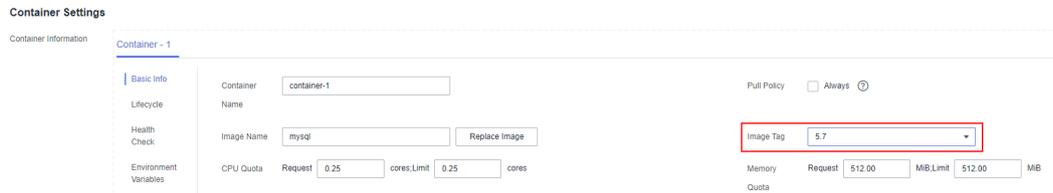
**Figura 5-2** Informações básicas sobre a carga de trabalho do MySQL

The screenshot shows the 'Basic Info' configuration page for a new workload. At the top, there are five options for 'Workload Type': Deployment (selected), StatefulSet, DaemonSet, Job, and Cron Job. Below this, a warning message states: 'Switching the workload type will require you to configure workload parameters again.' The 'Workload Name' field contains 'mysql'. The 'Namespace' is set to 'default' with a 'Create Namespace' link. The 'Pods' field is a numeric input set to '1'. The 'Time Zone' toggle is turned off. The 'Synchronization' section has a note: 'Allows containers to use the same time zone as the node where they run. (This function is realized by the local disks mounted to the containers. Do not modify or delete the local disks.)'

## Container Settings

Na área **Basic Info**, clique em **Select Image**. Na caixa de diálogo exibida, selecione **Open Source Images**, pesquise por **mysql**, selecione a imagem **mysql** e defina a tag de imagem como **5.7**.

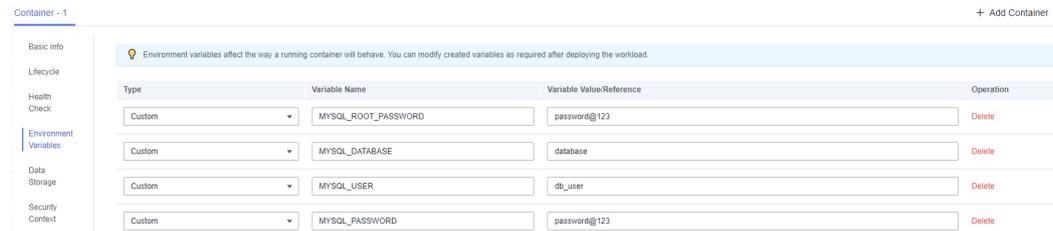
**Figura 5-3** Selecionar uma tag de imagem



Adicione as seguintes quatro variáveis de ambiente (detalhes disponíveis no [MySQL](#)):

- **MYSQL\_ROOT\_PASSWORD**: senha do usuário **root** do MySQL.
- **MYSQL\_DATABASE**: nome do banco de dados criado durante a inicialização da imagem.
- **MYSQL\_USER**: nome do usuário do banco de dados.
- **MYSQL\_PASSWORD**: senha do usuário do banco de dados.

**Figura 5-4** Configurar variáveis de ambiente



## Service Settings

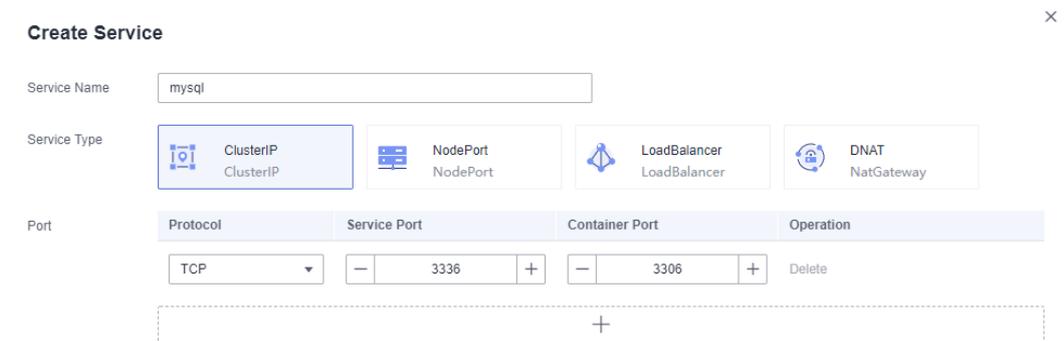
Clique no sinal de mais (+) para criar um serviço para acessar o MySQL a partir do WordPress.

Selecione **ClusterIP** para **Access Type**, defina **Service Name** como **mysql**, defina a **Container Port** e a **Service Port** como **3306** e clique em **OK**.

A porta de acesso padrão na imagem do MySQL é 3306. Neste exemplo, tanto a porta de contêiner quanto a porta de serviço são definidas como **3306** por conveniência. A porta de acesso pode ser alterada para outra porta.

Desta forma, a carga de trabalho do MySQL pode ser acessada através de **Service name:Access port (mysql:3306** neste exemplo) de dentro do cluster.

**Figura 5-5** Criar um Serviço

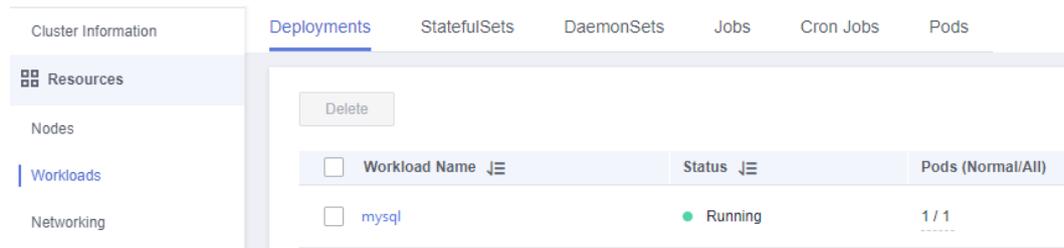


**Passo 5** Clique em **Create Workload**.

Aguarde até que a carga de trabalho seja criada.

A Implementação criada será exibida na guia **Deployments**.

**Figura 5-6** Carga de trabalho criada com sucesso



----Fim

## Operações através do kubectl

Esta seção descreve como usar o kubectl para criar uma Implementação e expor a Implementação por meio de um Serviço ClusterIP para permitir o acesso de dentro do cluster.

**Passo 1** Use o kubectl para se conectar ao cluster. Para obter detalhes, consulte [Conexão a um cluster usando o kubectl](#).

**Passo 2** Crie um arquivo de descrição chamado **mysql-deployment.yaml**. **mysql-deployment.yaml** é um nome de arquivo de exemplo. Você pode renomeá-lo conforme necessário.

**vi mysql-deployment.yaml**

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: mysql
  namespace: default
spec:
  replicas: 1
  selector:
    matchLabels:
      app: mysql
      version: v1
  template:
    metadata:
      labels:
        app: mysql
```

```

    version: v1
  spec:
    containers:
      - name: container-1
        image: mysql:5.7
        env:
          - name: MYSQL_ROOT_PASSWORD
            value: password@123
          - name: MYSQL_DATABASE
            value: database
          - name: MYSQL_USER
            value: db_user
          - name: MYSQL_PASSWORD
            value: password@123
        resources:
          requests:
            cpu: 250m
            memory: 512Mi
          limits:
            cpu: 250m
            memory: 512Mi
        imagePullSecrets:
          - name: default-secret

```

**Passo 3** Crie uma carga de trabalho do MySQL.

**kubectl apply -f mysql-deployment.yaml**

Se as informações a seguir forem exibidas, isso indicará que a Implementação está sendo criada.

```
deployment "mysql" created
```

Verifique a Implementação.

**kubectl get deployment**

Se as informações a seguir forem exibidas, a Implementação está sendo executada.

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
mysql	1/1	1	1	4m5s

**Passo 4** Crie um arquivo de descrição chamado **mysql-service.yaml**. **mysql-service.yaml** é um nome de arquivo de exemplo. Você pode renomeá-lo conforme necessário.

**vi mysql-service.yaml**

```

apiVersion: v1
kind: Service
metadata:
  name: mysql
  namespace: default
spec:
  selector:
    app: mysql
    version: v1
  ports:
    - name: cce-service-0
      targetPort: 3306
      nodePort: 0
      port: 3306
      protocol: TCP
  type: ClusterIP

```

**Passo 5** Crie um Serviço.

**kubectl create -f mysql-service.yaml**

Se forem apresentadas informações semelhantes às seguintes, o Serviço foi criado.

```
service/mysql created
```

### kubectl get svc

Se informações semelhantes às seguintes forem exibidas, o tipo de acesso foi configurado e a carga de trabalho está acessível.

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.247.0.1	<none>	443/TCP	3d
mysql	ClusterIP	10.247.202.20	<none>	3306/TCP	51s

----Fim

## 5.3 Criação de uma carga de trabalho do WordPress

WordPress foi originalmente uma plataforma de blog baseada em PHP e MySQL. É gradualmente evoluído para um sistema de gerenciamento de conteúdo. Você pode configurar seu próprio blog em qualquer servidor que suporte PHP e MySQL. Milhares de plug-ins e inúmeros modelos de tema estão disponíveis para WordPress e fácil de instalar.

Esta seção descreve como criar um site público do WordPress a partir de imagens.

### Pré-requisitos

- Você criou um cluster do CCE que contém um nó com 4 vCPUs e 8 GiB de memória. Para obter detalhes sobre como criar um cluster, consulte [Criação um cluster do Kubernetes](#).
- O banco de dados MySQL foi criado seguindo as instruções em [Criação de uma carga de trabalho do MySQL](#). Neste exemplo, os dados do WordPress são armazenados no banco de dados MySQL.

### Operações no console

**Passo 1** Efetue logon no [console do CCE](#).

**Passo 2** Escolha o cluster de destino.

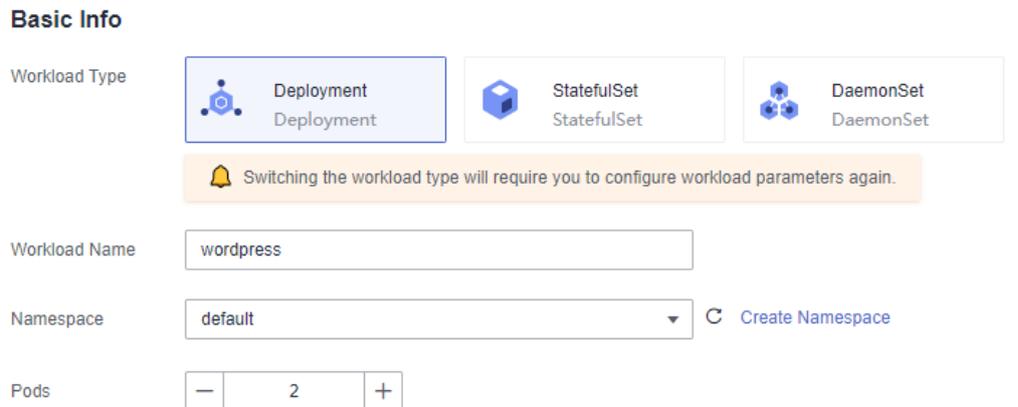
**Passo 3** No painel de navegação, escolha **Workloads**. Em seguida, clique em **Create Workload**.

**Passo 4** Defina parâmetros de carga de trabalho.

#### Basic Info

- **Workload Type:** selecione **Deployment**.
- **Workload Name:** defini-lo para **wordpress**.
- **Namespace:** selecione **Default**.
- **Pods:** defina este parâmetro como **2** neste exemplo.

**Figura 5-7** Configurar as informações básicas sobre a carga de trabalho



### Container Settings

Na área **Basic Info**, clique em **Select Image**. Na caixa de diálogo exibida, selecione **Open Source Images**, procure por **wordpress**, selecione a imagem **wordpress** e defina a tag de imagem para **php7.3**.

**Figura 5-8** Selecionar uma tag de imagem

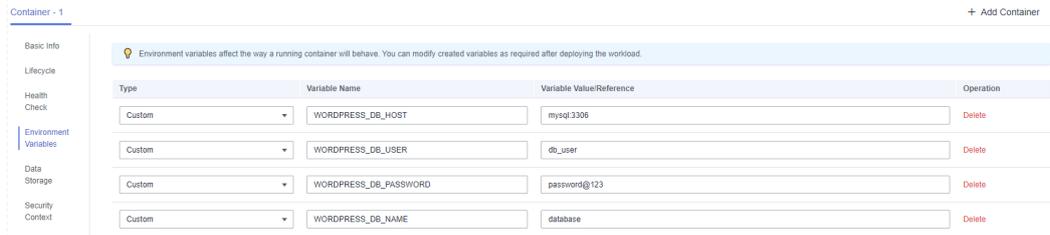


Adicione as seguintes variáveis de ambiente:

(Essas variáveis permitem que o WordPress saiba as informações sobre o banco de dados MySQL.)

- **WORDPRESS\_DB\_HOST**: endereço para acessar o banco de dados, que pode ser encontrado no Serviço (na página de guia **Services**) da carga de trabalho do MySQL. Você pode usar o nome de domínio interno **mysql.default.svc.cluster.local:3306** para acessar o banco de dados ou usar apenas **mysql:3306** omitindo **.default.svc.cluster.local**.
- **WORDPRESS\_DB\_USER**: nome do usuário para acessar o banco de dados. O valor deve ser o mesmo do **MYSQL\_USER** em [Criação de uma carga de trabalho do MySQL](#), que é usado para se conectar ao MySQL.
- **WORDPRESS\_DB\_PASSWORD**: senha para acessar o banco de dados. O valor deve ser o mesmo que o de **MYSQL\_PASSWORD** em [Criação de uma carga de trabalho do MySQL](#).
- **WORDPRESS\_DB\_NAME**: nome do banco de dados a ser acessado. O valor deve ser o mesmo do **MYSQL\_DATABASE** em [Criação de uma carga de trabalho do MySQL](#).

**Figura 5-9** Configurar variáveis de ambiente



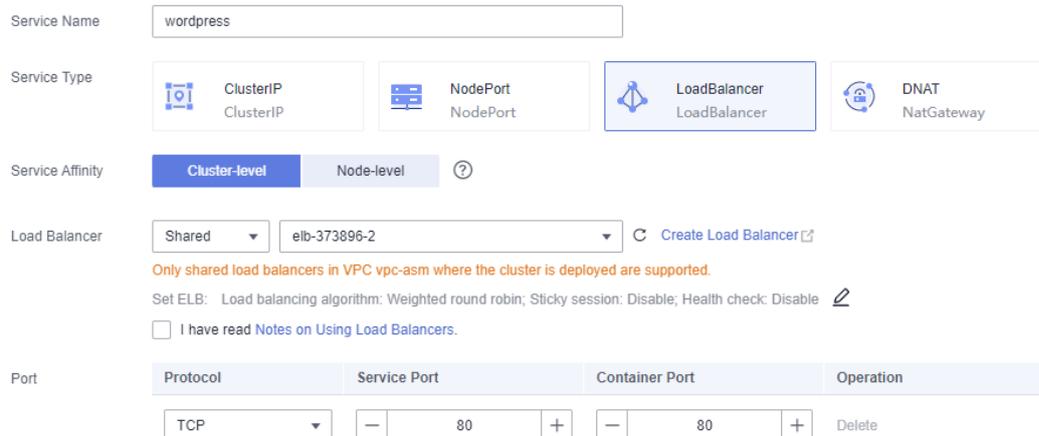
**Service Settings**

Clique no sinal de mais (+) para criar um serviço para acessar a carga de trabalho de uma rede externa. Neste exemplo, crie um Serviço LoadBalancer. Configure os seguintes parâmetros:

- **Service Name:** nome do Serviço exposto a redes externas. Neste exemplo, o nome do Serviço é **wordpress**.
- **Access Type:** selecione **LoadBalancer**.
- **Service Affinity:** mantenha o valor padrão.
- **Load Balancer:** se um balanceador de carga estiver disponível, selecione um balanceador de carga existente. Caso contrário, clique em **Create Load Balancer** para criar um no console do ELB.
- **Port:**
  - **Protocol:** selecione **TCP**.
  - **Service Port:** defina esse parâmetro como **80**, que é mapeado para a porta do contêiner.
  - **Container Port:** porta na qual a aplicação escuta. Para contêineres criados usando a imagem de nginx, defina esse parâmetro como **80**. Para outras aplicações, defina este parâmetro para a porta da aplicação.

**Figura 5-10** Criar um Serviço

**Create Service**

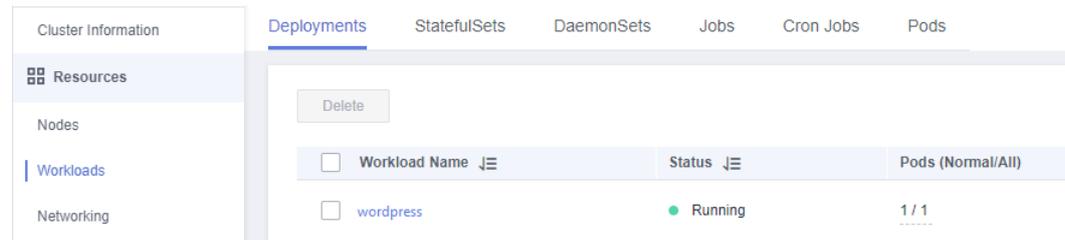


**Passo 5** Clique em **Create Workload**.

Aguarde até que a carga de trabalho seja criada.

A Implementação criada será exibida na guia **Deployments**.

**Figura 5-11** Carga de trabalho criada com sucesso



----Fim

## Operações através do kubectl

Esta seção descreve como usar o kubectl para criar uma Implementação e expor a Implementação na Internet por meio de um Serviço LoadBalancer.

- Passo 1** Use o kubectl para se conectar ao cluster. Para obter detalhes, consulte [Conexão a um cluster usando o kubectl](#).
- Passo 2** Crie um arquivo de descrição chamado **wordpress-deployment.yaml**. **wordpress-deployment.yaml** é um nome de arquivo de exemplo. Você pode renomeá-lo conforme necessário.

### vi wordpress-deployment.yaml

```
apiVersion: apps/v1
kind: Deployment
metadata:
  name: wordpress
  namespace: default
spec:
  replicas: 2
  selector:
    matchLabels:
      app: wordpress
      version: v1
  template:
    metadata:
      labels:
        app: wordpress
        version: v1
    spec:
      containers:
        - name: container-1
          image: wordpress:php7.3
          env:
            - name: WORDPRESS_DB_HOST
              value: mysql:3306
            - name: WORDPRESS_DB_USER
              value: db_user
            - name: WORDPRESS_DB_PASSWORD
              value: password@123
            - name: WORDPRESS_DB_NAME
              value: database
      resources:
        requests:
          cpu: 250m
          memory: 128Mi
        limits:
          cpu: 250m
          memory: 128Mi
      imagePullSecrets:
        - name: default-secret
```

### Passo 3 Crie uma carga de trabalho do WordPress

#### kubectl apply -f wordpress-deployment.yaml

Se as informações a seguir forem exibidas, isso indicará que a Implementação está sendo criada.

```
deployment "wordpress" created
```

Verifique a Implementação.

#### kubectl get deployment

Se as informações a seguir forem exibidas, a Implementação está sendo executada.

NAME	READY	UP-TO-DATE	AVAILABLE	AGE
wordpress	1/1	1	1	4m5s

### Passo 4 Crie um arquivo de descrição chamado **wordpress-service.yaml**. **wordpress-service.yaml** é um nome de arquivo de exemplo. Você pode renomeá-lo conforme necessário.

#### vi wordpress-service.yaml

```
apiVersion: v1
kind: Service
metadata:
  name: wordpress
  namespace: default
  annotations:
    kubernetes.io/elb.class: union
    kubernetes.io/elb.autocreate:
      '{
        "type": "public",
        "bandwidth_name": "cce-wordpress",
        "bandwidth_chargemode": "bandwidth",
        "bandwidth_size": 5,
        "bandwidth_sharetype": "PER",
        "eip_type": "5_bgp"
      }'
```

```
spec:
  selector:
    app: wordpress
  externalTrafficPolicy: Cluster
  ports:
    - name: cce-service-0
      targetPort: 80
      nodePort: 0
      port: 80
      protocol: TCP
  type: LoadBalancer
```

### Passo 5 Crie um Serviço.

#### kubectl create -f wordpress-service.yaml

Se forem apresentadas informações semelhantes às seguintes, o Serviço foi criado.

```
service/wordpress created
```

#### kubectl get svc

Se informações semelhantes às seguintes forem exibidas, o tipo de acesso foi configurado e a carga de trabalho está acessível.

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
kubernetes	ClusterIP	10.247.0.1	<none>	443/TCP	3d

mysql	ClusterIP	10.247.202.20	<none>	3306/TCP	8m
wordpress	LoadBalancer	10.247.130.196	**.*.*.*.*	80:31540/TCP	51s

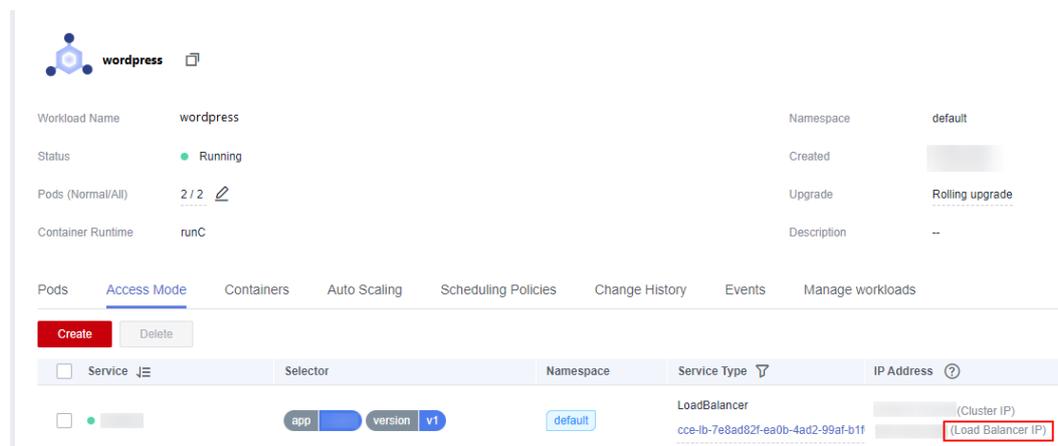
----Fim

## Acessar WordPress

**Passo 1** Obtenha o endereço de acesso externo do WordPress.

Clique na carga de trabalho do wordpress para entrar na página de detalhes. Na página de guia **Access Mode**, exiba o endereço IP do WordPress. O endereço IP do balanceador de carga é o endereço de acesso externo.

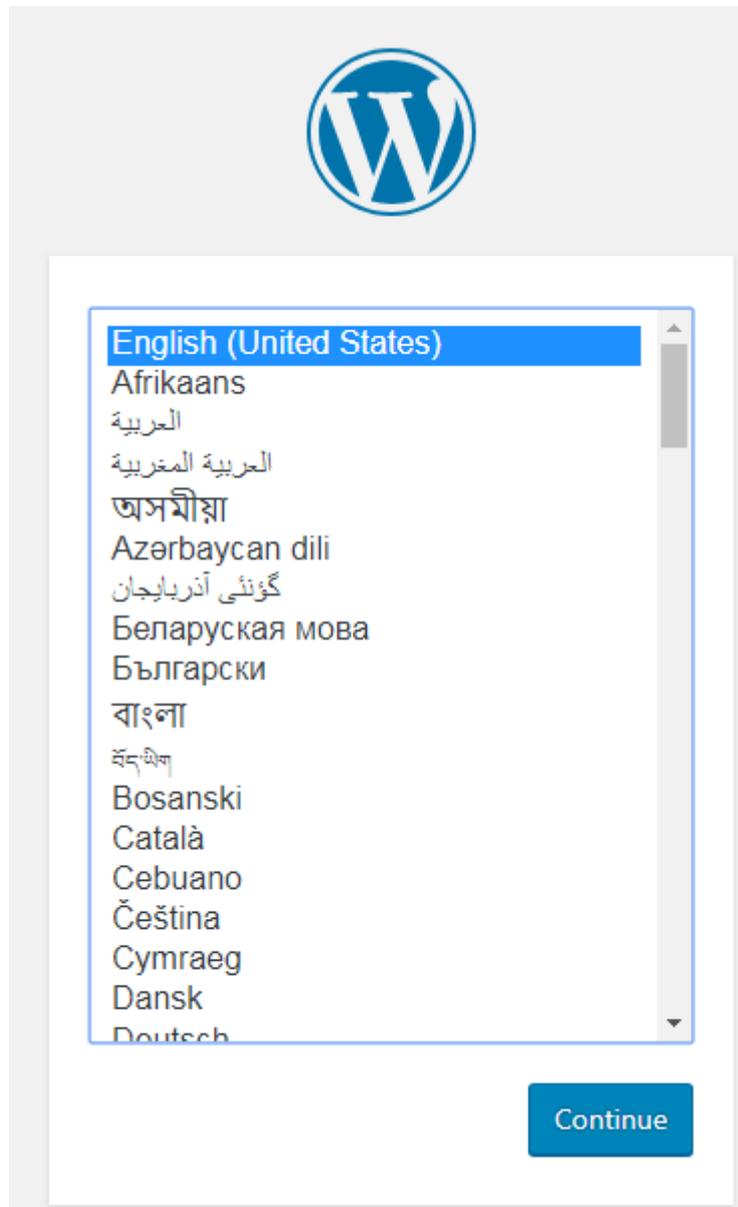
**Figura 5-12** Acessar WordPress



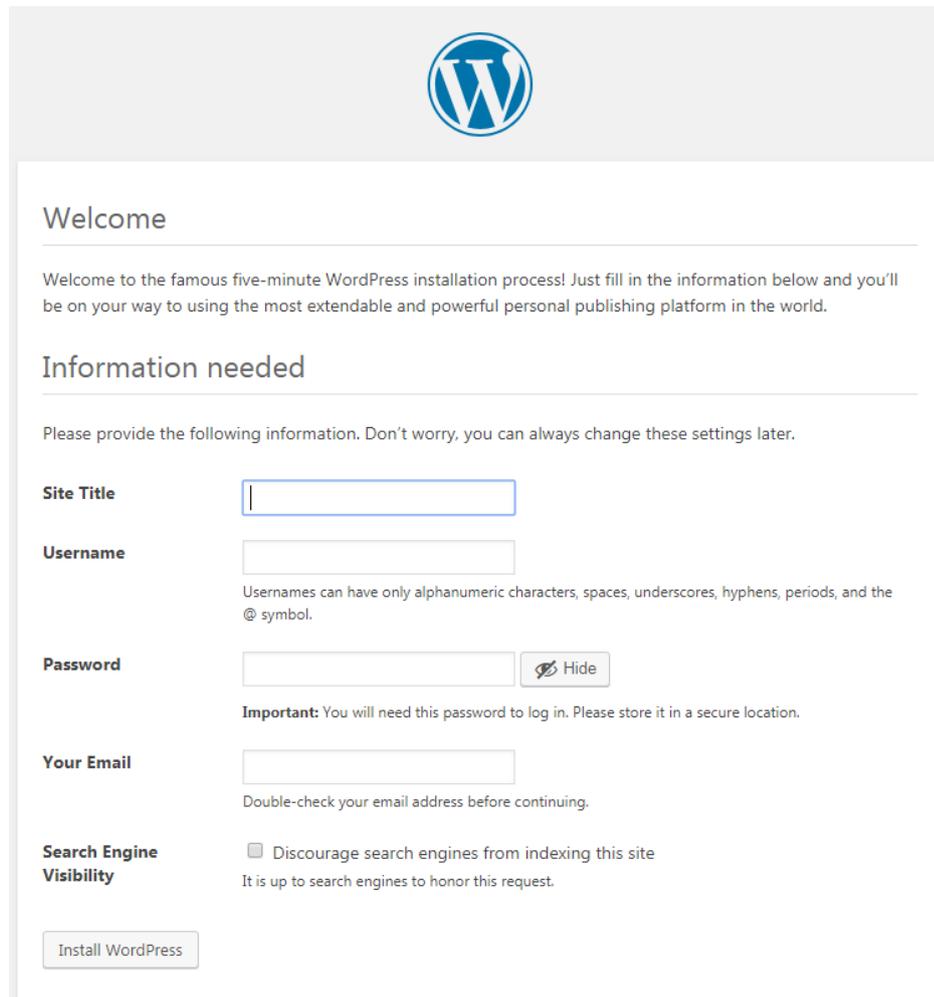
**Passo 2** Digite o endereço de acesso externo na caixa de endereço de um navegador para se conectar à carga de trabalho.

A figura a seguir mostra a página de WordPress acessada.

**Figura 5-13** Carga de trabalho do WordPress



**Figura 5-14** WordPress



The image shows the WordPress installation 'Information needed' screen. At the top is the WordPress logo. Below it is a 'Welcome' section with a message: 'Welcome to the famous five-minute WordPress installation process! Just fill in the information below and you'll be on your way to using the most extendable and powerful personal publishing platform in the world.' The main section is titled 'Information needed' and contains the following fields and options:

- Site Title:** A text input field.
- Username:** A text input field with a note: 'Usernames can have only alphanumeric characters, spaces, underscores, hyphens, periods, and the @ symbol.'
- Password:** A text input field with a 'Hide' button (eye icon).
- Your Email:** A text input field with a note: 'Double-check your email address before continuing.'
- Search Engine Visibility:** A checkbox labeled 'Discourage search engines from indexing this site' with a note: 'It is up to search engines to honor this request.'

At the bottom left is an 'Install WordPress' button.

----Fim

## Excluir recursos

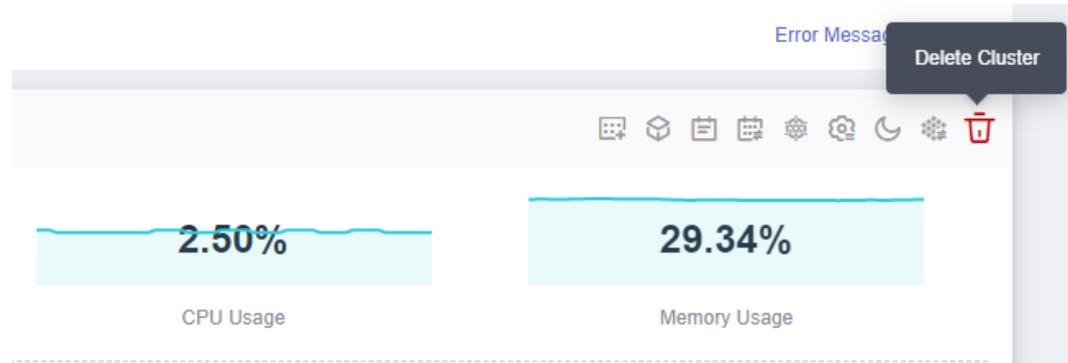
Até agora, você completou todas as instruções de Primeiros passos e entendeu como usar o CCE. As taxas são incorridas enquanto os nós estão em execução. Se os clusters usados nos guias de Primeiros passos não estiverem mais em uso, execute as etapas a seguir para excluí-los. Se você vai continuar os percursos manuais do CCE, retenha os clusters.

**Passo 1** Efetue login no console do CCE.

**Passo 2** No painel de navegação à esquerda, escolha **Clusters**.

**Passo 3** Clique em  ao lado de um cluster e exclua o cluster conforme solicitado.

**Figura 5-15** Excluir um cluster



----Fim

# 6 Implementação de WordPress usando o Helm

---

O Helm é um gerenciador de pacotes que simplifica a implementação, atualização e gerenciamento de aplicações de Kubernetes. Helm usa charts (gráficos, um formato de empacotamento que define os recursos do Kubernetes) para encapsular todos os elementos implantados pelo Kubernetes, incluindo código de aplicação, dependências, arquivos de configuração e instruções de implementação. Com o Helm, você pode implementar e gerenciar facilmente aplicações complexas do Kubernetes, simplificando o desenvolvimento e a implementação de aplicações.

Esta seção descreve como implementar a carga de trabalho do WordPress usando Helm.

## Pré-requisitos

Você criou um cluster do CCE que contém um nó com 4 vCPUs e 8 GiB de memória. O nó tem um EIP para puxar a imagem do WordPress de um repositório externo. Para obter detalhes sobre como criar um cluster, consulte [Criação um cluster do Kubernetes](#).

## Preparações

1. Faça download e configure o kubectl para se conectar ao cluster.  
Efetue login no console do CCE e clique no nome do cluster de destino. Na área **Connection Information** da página **Cluster Information**, clique em **Configure** ao lado de **kubectl** e configure o kubectl conforme as instruções.

Figura 6-1 kubectl

### Connection Information

Private IP	https://192.168.0.198:5443 
EIP	-- Bind
Custom SAN	-- 
kubectl	<b>Configure</b>
Certificate Authentication	X.509 certificate <a href="#">Download</a>

## 2. Instale Helm 3.

## Implementação de WordPress

**Passo 1** Adicione o repositório oficial de WordPress.

```
helm repo add bitnami https://charts.bitnami.com/bitnami
```

**Passo 2** Execute os seguintes comandos para criar uma carga de trabalho de WordPress:

```
helm install myblog bitnami/wordpress \
--set mariadb.primary.persistence.enabled=true \
--set mariadb.primary.persistence.storageClass=csi-disk \
--set mariadb.primary.persistence.size=10Gi \
--set persistence.enabled=false
```

### NOTA

- mariadb usa um volume persistente (PV) para armazenar dados. O PV é criado automaticamente com um disco EVS de 10 GiB configurando **StorageClassName**.
- A persistência de dados não é necessária para WordPress. Defina **persistence.enabled** como **false** para o volume de dados.

Informação semelhante à seguinte foi exibida:

```
coalesce.go:223: warning: destination for
mariadb.networkPolicy.egressRules.customRules is a table. Ignoring non-table
value ([])
NAME: myblog
LAST DEPLOYED: Mon Mar 27 11:47:58 2023
NAMESPACE: default
STATUS: deployed
REVISION: 1
TEST SUITE: None
NOTES:
CHART NAME: wordpress
CHART VERSION: 15.2.57
APP VERSION: 6.1.1

** Please be patient while the chart is being deployed **

Your WordPress site can be accessed through the following DNS name from within
your cluster:

    myblog-wordpress.default.svc.cluster.local (port 80)

To access your WordPress site from outside the cluster follow the steps below:

1. Get the WordPress URL by running these commands:
```

```
NOTE: It may take a few minutes for the LoadBalancer IP to be available.
      Watch the status with: 'kubectl get svc --namespace default -w myblog-
wordpress'

      export SERVICE_IP=$(kubectl get svc --namespace default myblog-wordpress --
template "{{ range (index .status.loadBalancer.ingress 0) }}{{ . }}{{ end }}")
      echo "WordPress URL: http://$SERVICE_IP/"
      echo "WordPress Admin URL: http://$SERVICE_IP/admin"

2. Open a browser and access WordPress using the obtained URL.

3. Login with the following credentials below to see your blog:

      echo Username: user
      echo Password: $(kubectl get secret --namespace default myblog-wordpress -o
jsonpath="{.data.wordpress-password}" | base64 -d)
```

A saída do comando mostra como obter o URL de WordPress e o nome de usuário e senha para fazer login no plano de fundo de WordPress.

----Fim

## Acessar WordPress

**Passo 1** Modifique a configuração do Serviço de WordPress.

É necessária uma configuração de anotação adicional para usar um Serviço de LoadBalancer no CCE. No entanto, **bitnami/wordpress** não tem essa configuração. Você precisa modificar manualmente a configuração.

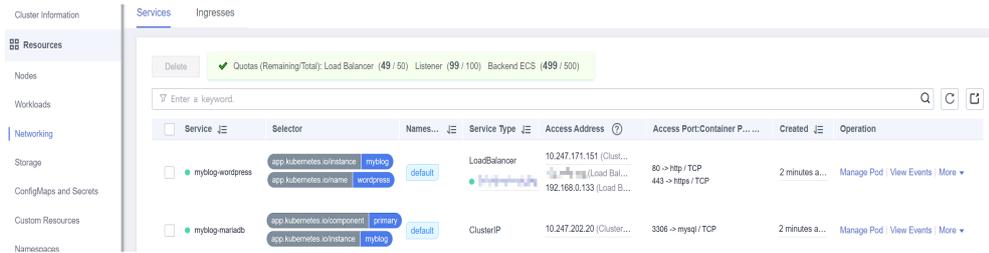
### kubectl edit svc myblog-wordpress

Adicione **kubernetes.io/elb.autocreate** e **kubernetes.io/elb.class** a **metadata.annotations** e salve a modificação. As duas anotações são usadas para criar um balanceador de carga compartilhado para que a carga de trabalho de WordPress possa ser acessada por meio do EIP do balanceador de carga.

```
apiVersion: v1
kind: Service
metadata:
  name: myblog-wordpress
  namespace: default
  annotations:
    kubernetes.io/elb.autocreate: '{ "type": "public", "bandwidth_name": "myblog-
wordpress", "bandwidth_chargemode": "bandwidth", "bandwidth_size": 5,
"bandwidth_sharetype": "PER", "eip_type": "5_bgp" }'
    kubernetes.io/elb.class: union
spec:
  ports:
    - name: http
  ...
```

**Passo 2** Obtenha o endereço IP do balanceador de carga de rede público do Serviço **myblog-wordpress** e acesse-o.

Na guia **Services**, localize **myblog-wordpress** e localize o endereço IP do balanceador de carga de rede público.



Na caixa endereço de um navegador, digite **<Public IP address of the load balancer>:80** para acessar WordPress.

User's Blog!

Sample Page

Mindblown: a blog about philosophy.

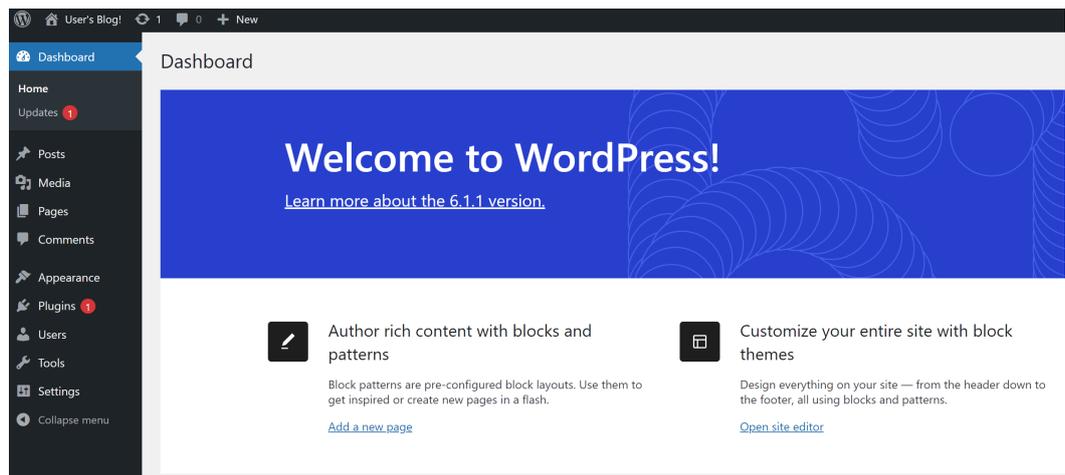
Hello world!

Welcome to WordPress. This is your first post.  
Edit or delete it, then start writing!

March 27, 2023

Na caixa endereço de um navegador, digite **<Public IP address of the load balancer>:80/login** para acessar o plano de fundo de WordPress como usuário **user**. Execute o seguinte comando para obter a senha do **user**:

```
kubectl get secret --namespace default myblog-wordpress -o jsonpath="{.data.wordpress-password}" | base64 -d
```



----Fim

## Excluir a carga de trabalho de WordPress

**Passo 1** Execute o seguinte comando para excluir a carga de trabalho de WordPress:

```
helm delete myblog
```

Informação semelhante à seguinte foi exibida:

```
release "myblog" uninstalled
```

**Passo 2** Exclua PersistentVolumeClaim (PVC) usado pelo componente mariadb em WordPress.

**kubectl delete pvc data-myblog-mariadb-0**

Informação semelhante à seguinte foi exibida:

```
persistentvolumeclaim "data-myblog-mariadb-0" deleted
```

**---Fim**