Tencent Cloud Elastic Microservice

Getting Started

Product Documentation

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.

Contents

Getting Started
Step 1. Get the Access Permission
Access Acquisition by Root Account
Access Acquisition by Sub-account
Step 2. Create an Environment
Step 3. Create an Application
Step 4. Deploy the Application
Deploying a General Application
Deploying Microservice Application

Getting Started

Step 1. Get the Access Permission

Access Acquisition by Root Account

Last updated：2024-01-09 12:00:35

Overview

As TEM needs to access APIs of other Tencent Cloud products, you need to create service roles and grant TEM those roles.

Preparation

You have signed up for a Tencent Cloud account.

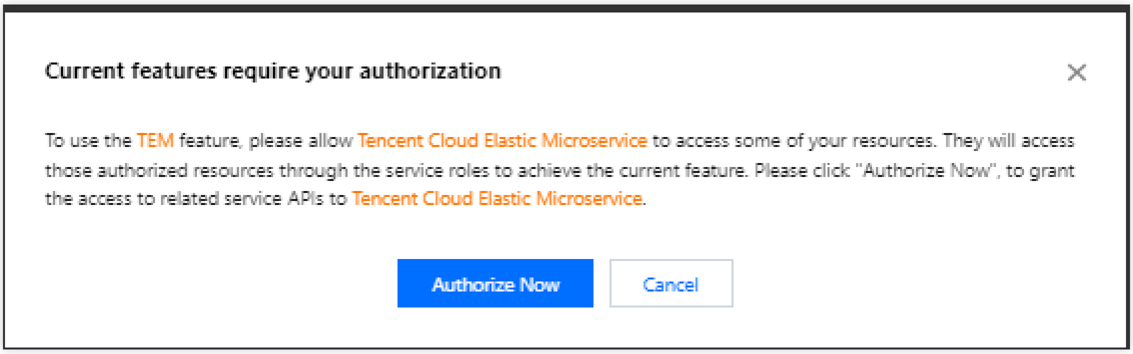
Note:

After you register a Tencent Cloud account, the system will create a root account for you by default, which is used to quickly access Tencent Cloud resources.

Directions

If you use TEM for the first time, you need to create the `TEM_QCSLinkedRoleInAccessCluster` and `TEM_QCSLinkedRoleInTEMLog` service roles as follows to grant you the permissions to access resources of other Tencent Cloud products.

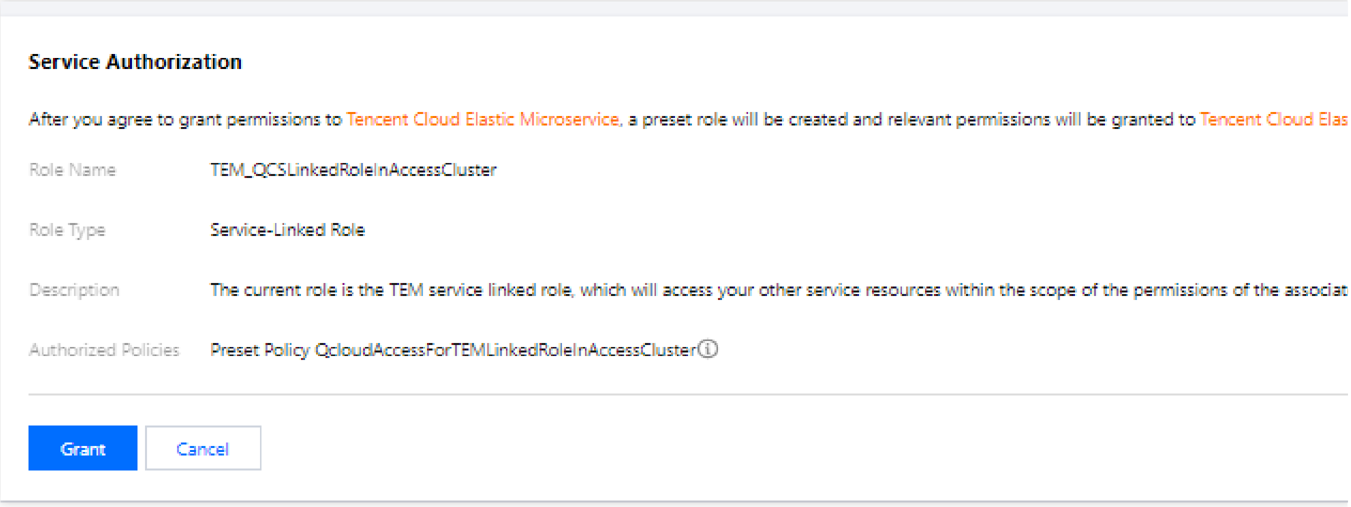
1. Log in to the TEM console with the **root account**.



Current features require your authorization

To use the TEM feature, please allow Tencent Cloud Elastic Microservice to access some of your resources. They will access those authorized resources through the service roles to achieve the current feature. Please click "Authorize Now", to grant the access to related service APIs to Tencent Cloud Elastic Microservice.

2. Click **Authorize Now** to enter the CAM console, click **Grant**, and the `TEM_QCSLinkedRoleInAccessCluster` service role will be created to grant you the permissions to access resources of Tencent Cloud products other than CLS.



Service Authorization

After you agree to grant permissions to Tencent Cloud Elastic Microservice, a preset role will be created and relevant permissions will be granted to Tencent Cloud Elas

Role Name	TEM_QCSLinkedRoleInAccessCluster
Role Type	Service-Linked Role
Description	The current role is the TEM service linked role, which will access your other service resources within the scope of the permissions of the associat
Authorized Policies	Preset Policy QcloudAccessForTEMLinkedRoleInAccessCluster ⓘ

3. Return to the TEM console and click **Authorized**.
4. On the left sidebar, click **Application management**. In the pop-up window, click **Authorize Now** to enter the CAM console, click **Grant**, and the `TEM_QCSLinkedRoleInTEMLog` service role will be created to grant you the permission to access CLS resources.

Access Acquisition by Sub-account

Last updated：2024-01-09 12:00:35

Basic CAM Concepts

A root account authorizes sub-accounts by binding policies. The policy settings can be specific to the level of **API, Resource, User/User Group, Allow/Deny, and Condition**.

User type

Root account: A root accounts owns all the resources in Tencent Cloud and can access any of these resources.
Sub-account: Sub-accounts include sub-users, WeCom sub-users, collaborators, and message recipients.
For detailed definitions of the root account and sub-accounts and descriptions of their features, see User Types.

Resources and Permissions

Resource: An object that Tencent Cloud services operate on, such as a CVM instance, a COS bucket, or a VPC instance.
Permission: An authorization to allow or forbid users to perform certain operations. By default, **a root account has full access to all resources under the account**, while **a sub-account does not have access to any resources under its root account**.
Policy: Syntax rule to define and describe one or more permissions. **A root account** performs authorization by **associating policies** with users/user groups.

Using TEM with a Sub-account

To allow a sub-account such as collaborator to use TEM, you need to complete the following authorization operations:
1. To pass a role and its policies to TEM, the user must have the permission to **pass roles** to the service, i.e., the **PassRole** policy must be created. For detailed directions, see Granting the `PassRole` policy.
2. The permission to use TEM is required. You can grant the target sub-account the `QcloudTEMFullAccess` or `QcloudTEMReadOnlyAccess` policy to grant it full or read-only access to TEM. For detailed directions, see Granting the permission to use TEM.
3. TEM may call other Tencent Cloud products when used, so the root account needs to authorize the target sub-account accordingly. For more information, see Granting permissions to access other Tencent Cloud products.

Granting the PassRole policy

Step 1. Create policies

1. Log in to the CAM console.
2. On the left sidebar, click **Policies** to enter the policy management page.
3. Click **Create Custom Policy**.
4. In the **Select Policy Creation Method** pop-up window, click **Create by Policy Syntax**.
5. On the Create by Policy Syntax page, select **Blank Template** and click **Next**.
6. Enter the policy name and content and click **Create Policy**. Use the root account or a sub-account with admin permissions to create the following two custom policies:

Access to resources of Tencent Cloud products other than CLS:

```
{
     "version": "2.0",
     "statement": {
             "effect": "allow",
             "action": "cam:PassRole",
             "resource": "qcs::cam::uin/${OwnerUin}:role/tencentcloudServiceRoleNam
     }
}
```

Access to CLS resources:

```
{
     "version": "2.0",
     "statement": {
             "effect": "allow",
             "action": "cam:PassRole",
             "resource": "qcs::cam::uin/${OwnerUin}:role/tencentcloudServiceRoleNam
     }
}
```

Here, `${OwnerUin}` is the root account ID, which can be obtained on the Account Info page in the console.

Step 2. Bind the policies to the user

1. On the left sidebar, click **User** > **User List** to enter the user management page.
2. Select the target user and click **Authorize** in the **Operation** column.
3. Filter the policies created in **step 1** in the **Policy List**.
4. Click **OK** to bind the policies, which will be displayed in the **Policy List** of the user.

Granting the permission to use TEM

Full access policy

Grant a sub-user full access (including resource creation and management) to the TEM service.

```
    ]
}
```

```
{
    "version": "2.0",
    "statement": [
        {
            "action": "tem:*",
            "resource": "*",
            "effect": "allow"
        }
    ]
}
```

You can also configure the system's full read/write policy to support this permission.

1. Log in to the CAM console.
2. Click **Policies** on the left sidebar.
3. In the policy list, click **Create Custom Policy**.
4. In the **Select Policy Creation Method** pop-up window, select **Create by Syntax**.
5. In **Template Type**, search for "tem", select **QcloudTEMFullAccess** (full access to TEM), and click **Next**.
6. Click **Complete**.

Subsequent operation: Bind the created policy to the target user.

Read-only policy

Grant a sub-user read-only access to the TEM service.

```
{
    "version": "2.0",
    "statement": [
        {
            "action": [
                "tem:Describe*"
            ],
            "resource": "*",
            "effect": "allow"
        }
    ]
}
```

You can also configure the system's read-only policy to support this permission.

1. Log in to the CAM console.
2. Click **Policies** on the left sidebar.
3. In the policy list, click **Create Custom Policy**.
4. In the **Select Policy Creation Method** pop-up window, select **Create by Syntax**.
5. In **Template Type**, search for "tem", select **QcloudTEMReadOnlyAccess** (read-only access to TEM), and click **Next**.
6. Click **Create Policy**.

Granting permissions to access other Tencent Cloud products

TEM may call the following Tencent Cloud products when used, so the root account needs to authorize the target sub-account separately to ensure that the sub-account can use TEM product features normally:
Below is the sample code for authorization:

```
{
    "version": "2.0",
    "statement": [
        {
            "effect": "allow",
            "action": [
                "cam:DescribeRoleList",
                "cvm:DescribeSecurityGroups",
                "vpc:DescribeVpcEx",
                "vpc:DescribeSubnetEx",
                "tse:DescribeSREInstances",
                "cls:DescribeLogsets",
                "cls:DescribeTopics",
                "cfs:DescribeCfsFileSystems",
                "ssl:DescribeCertificate",
                "tcr:DescribeRepositoryOwnerPersonal",
                "tcr:DescribeRepositories",
                "tcr:DescribeInstances",
                "tcr:DescribeInternalEndpoints",
                "tcr:CreateInstanceToken"
            ],
            "resource":[
                "*"
            ]
        }
    ]
}
```

Step 2. Create an Environment

Last updated：2024-03-08 12:29:05

Overview

In TEM, an environment is a collection of computing, network, and storage resources. TEM provides the multi-environment management feature. You can create multiple isolated environments for development, testing, prerelease, and production according to your business needs and deploy applications separately. In this way, applications are isolated by environment, and those in the same environment can access each other through the Service mechanism in K8s or registries like ZooKeeper.
This document describes how to create an environment in the TEM console.

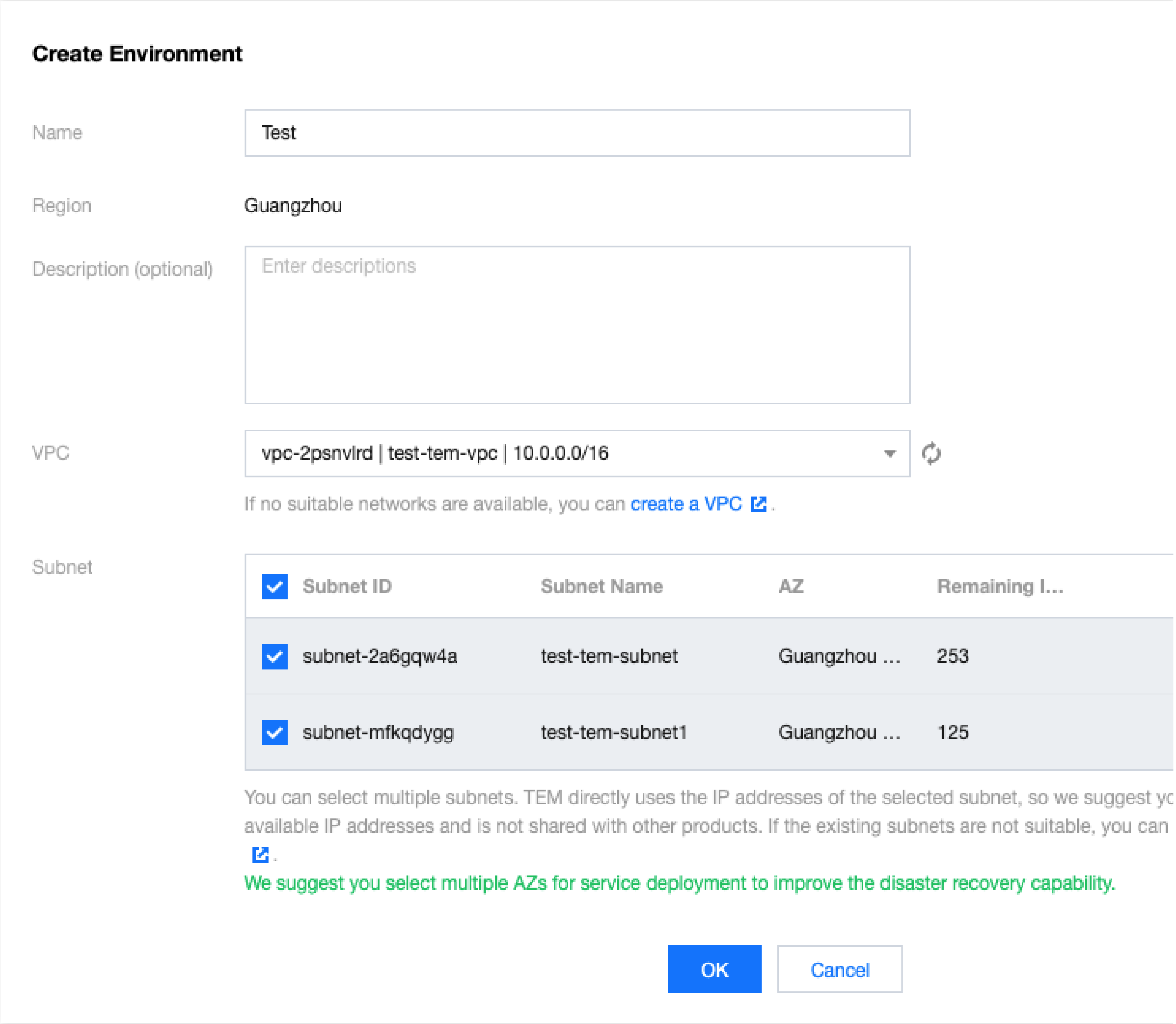
Directions

1. Log in to the TEM console and click **Manage Environment** on the left sidebar.
2. On the **Manage Environment** page, select a deployment region and click **Create**.

Note:

You can create up to five environments in the same region.

3. In the **Create environment** window, configure the environment information.



Create Environment

Name	Test
Region	Guangzhou
Description (optional)	Enter descriptions
VPC	vpc-2psnvlrd | test-tem-vpc | 10.0.0.0/16

If no suitable networks are available, you can create a VPC .

Subnet

☑	Subnet ID	Subnet Name	AZ	Remaining I...
☑	subnet-2a6gqw4a	test-tem-subnet	Guangzhou ...	253
☑	subnet-mfkqdygg	test-tem-subnet1	Guangzhou ...	125

You can select multiple subnets. TEM directly uses the IP addresses of the selected subnet, so we suggest yc available IP addresses and is not shared with other products. If the existing subnets are not suitable, you can .

We suggest you select multiple AZs for service deployment to improve the disaster recovery capability.

Name: enter up to 40 characters.

VPC: Select an existing VPC. If your existing VPCs are not suitable or you haven't created a VPC yet, you can click Create VPC to create one (note that the selected region must be the same as that of the environment), return to and refresh this page, and select it.

Subnet: Select an existing subnet. We recommend you choose multi-AZ deployment to improve the disaster recovery capabilities. If your existing subnets are not suitable or you haven't created a subnet yet, you can click Create Subnet to create one, return to and refresh this page, and select it.

CoreDNS is automatically deployed to support service discovery in the environment. Specifically, two replica nodes of `Deployment:coredns` are automatically deployed in the Kubernetes cluster namespace `kube-system` . This service is free of charge, and we recommend you not modify it.

4. Click **OK** and the environment will enter the initialization status. Wait a few minutes, and the environment will be created.

Step 3. Create an Application

Last updated：2024-01-09 12:00:35

Overview

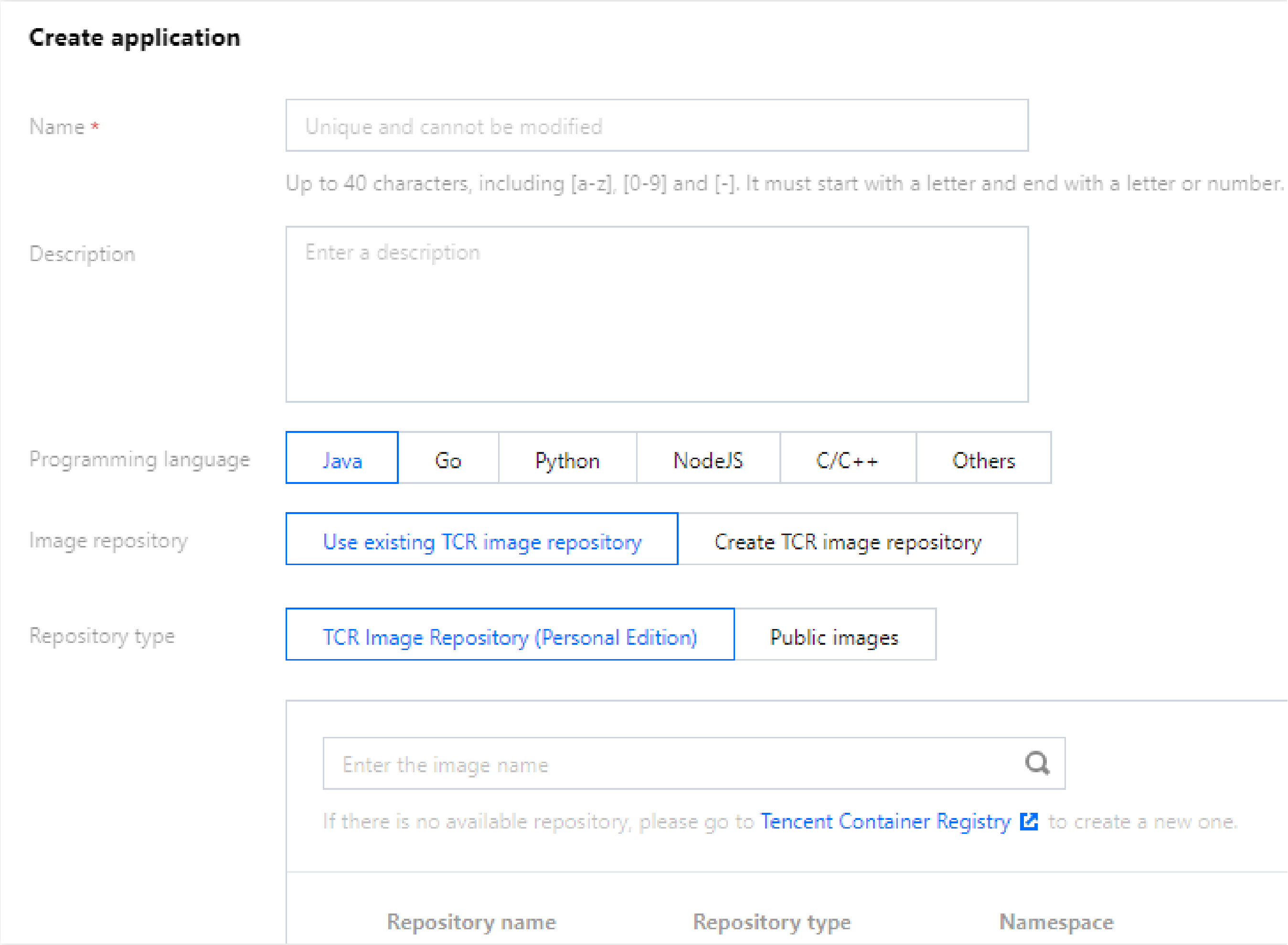
This document describes how to create an application in the TEM console.

Preparation

You have activated TCR before creating your application.

Directions

1. Select the application deployment region at the top of the **Application management** page in the TEM console.
2. Click **Create** to go to the application creation page and enter the application information.
3. Select the programming language of your application and the package upload method. Image, JAR, and WAR files can be uploaded in Java, while only image files can be uploaded in other languages.
4. Click **Submit** and click **OK** in the pop-up window.



Create application

Name *: Unique and cannot be modified

Up to 40 characters, including [a-z], [0-9] and [-]. It must start with a letter and end with a letter or number.

Description: Enter a description

Programming language: Java | Go | Python | NodeJS | C/C++ | Others

Image repository: Use existing TCR image repository | Create TCR image repository

Repository type: TCR Image Repository (Personal Edition) | Public images

Enter the image name

If there is no available repository, please go to Tencent Container Registry to create a new one.

Repository name	Repository type	Namespace

Step 4. Deploy the Application

Deploying a General Application

Last updated：2024-01-09 12:00:35

Overview

This document describes how to deploy a JAR package of the JAVA Demo sample application on Tencent Cloud Elastic Microservice and enable its public network access.

Prerequisites

Create an environment and application.

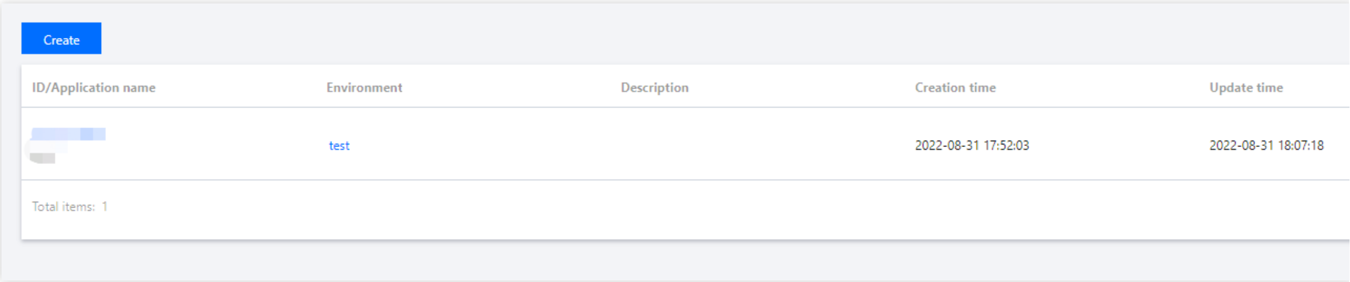
Directions

Step 1. Get the Demo

This document uses the demo application developed in Java Spring Boot as an example. Click here to download the JAR package.

Step 2. Deploy the Application

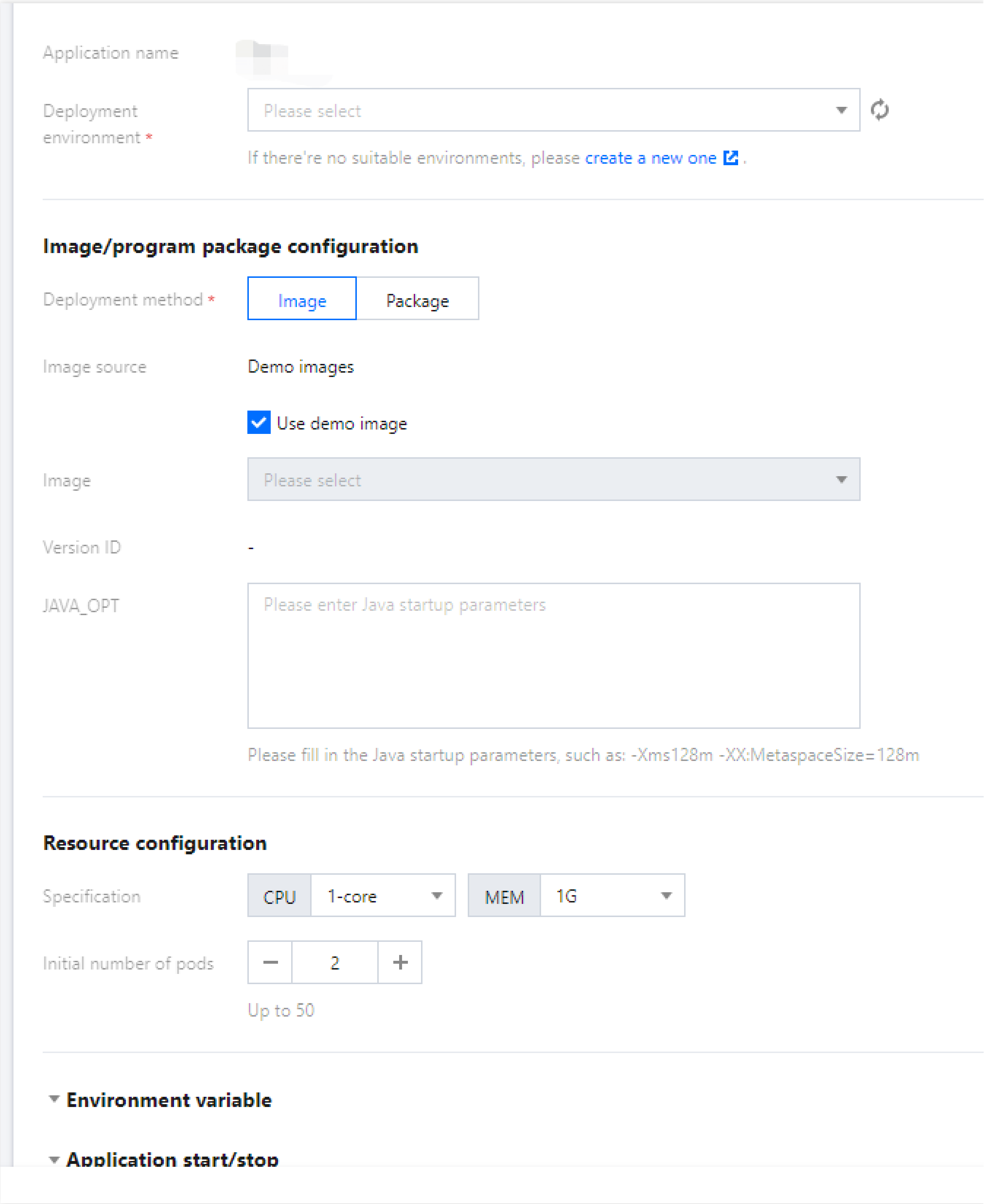
1. On the **Application management** page, select the target application and click **Deploy to new environment**.



ID/Application name	Environment	Description	Creation time	Update time
[redacted]	test		2022-08-31 17:52:03	2022-08-31 18:07:18

Total items: 1

2. On the **Deploy Application** page, configure the relevant parameters as needed.



Application name [redacted]

Deployment environment * Please select

If there're no suitable environments, please create a new one .

Image/program package configuration

Deployment method * Image Package

Image source Demo images

☑ Use demo image

Image Please select

Version ID -

JAVA_OPT Please enter Java startup parameters

Please fill in the Java startup parameters, such as: -Xms128m -XX:MetaspaceSize=128m

Resource configuration

Specification CPU 1-core MEM 1G

Initial number of pods − 2 +

Up to 50

▾ Environment variable

▾ Application start/stop

Upload Package: upload your package.

Release Environment: select the environment just created.

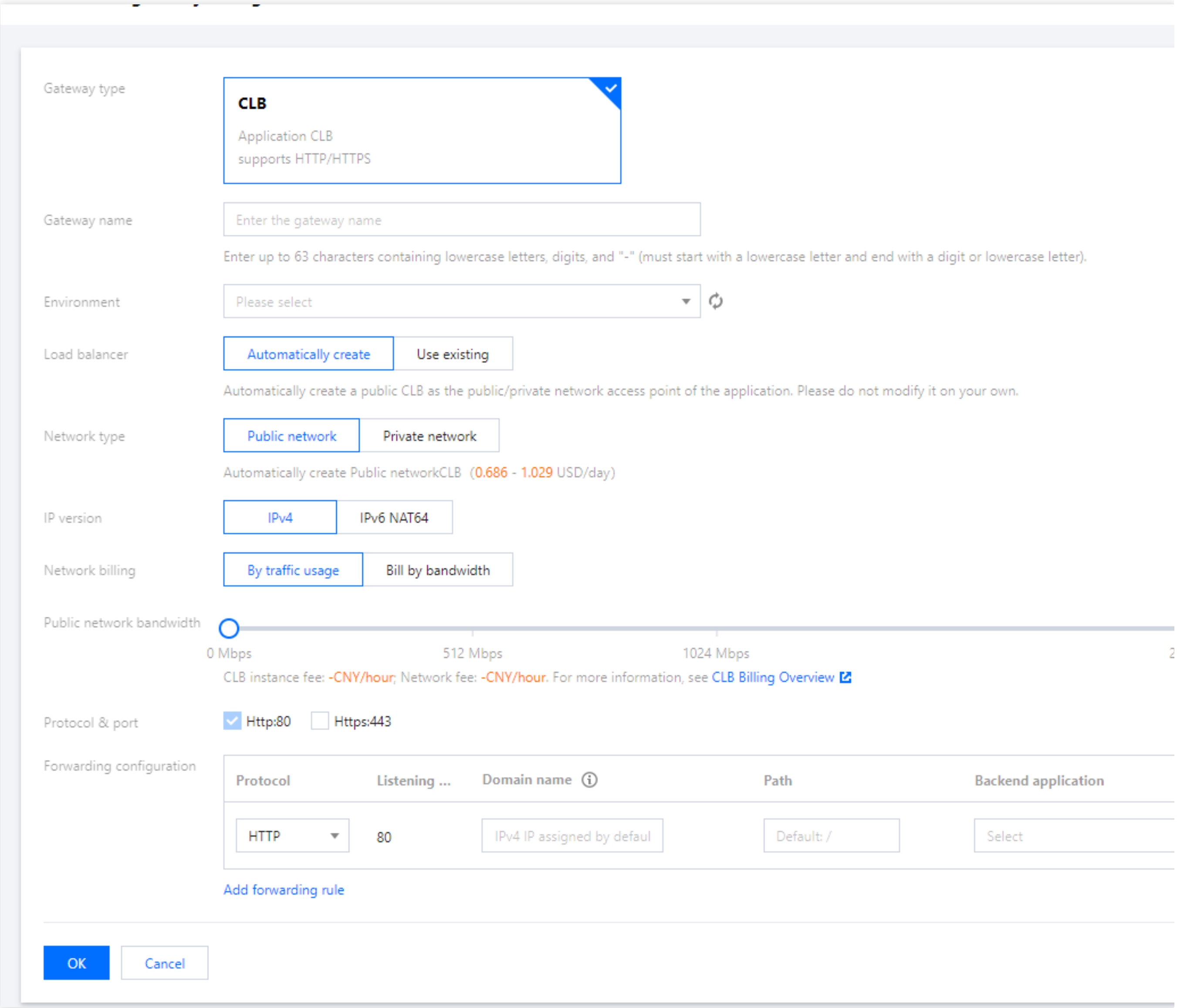
If you need to configure other advanced options for your application, please see Creating and Deploying Application.

3. Click **Deploy**.

Step 3. Access the Application

After the deployment, you need to create a configuration for it in the CLB gateway for internet access.

1. In the TEM console, click CLB Gateways in the left sidebar.
2. Click **Create** and enter the gateway configuration. Choose the correct **Environment**.



Gateway type: CLB, Application CLB supports HTTP/HTTPS

Gateway name: Enter the gateway name

Enter up to 63 characters containing lowercase letters, digits, and "-" (must start with a lowercase letter and end with a digit or lowercase letter).

Environment: Please select

Load balancer: [✓ Automatically create] Use existing

Automatically create a public CLB as the public/private network access point of the application. Please do not modify it on your own.

Network type: [✓ Public network] Private network

Automatically create Public networkCLB (0.686 - 1.029 USD/day)

IP version: [✓ IPv4] IPv6 NAT64

Network billing: [✓ By traffic usage] Bill by bandwidth

Public network bandwidth: 0 Mbps, 512 Mbps, 1024 Mbps, 2

CLB instance fee: -CNY/hour; Network fee: -CNY/hour. For more information, see CLB Billing Overview

Protocol & port: ☑ Http:80 □ Https:443

Forwarding configuration:

Protocol	Listening ...	Domain name	Path	Backend application
HTTP	80	IPv4 IP assigned by defaul	Default: /	Select

Add forwarding rule

OK Cancel

Network Type: Select **Public Network**. To configure an intra-environment access, see Creating and Deploying Application.

Load balancer: Create one automatically or use an existing CLB.

Protocol & port: **HTTP:80** and **HTTPS:443** are supported, and HTTPS domain names can be bound to certificates. Select **HTTP:80** for the Demo application.

IP version: Options include **IPv4** and **IPv6 NAT64**.

Forwarding configuration:

Domain name: Choose an existing domain name. If it’s not specified, a random IPv4 IP is assigned by default. For the demo application, use the assigned default IP address.

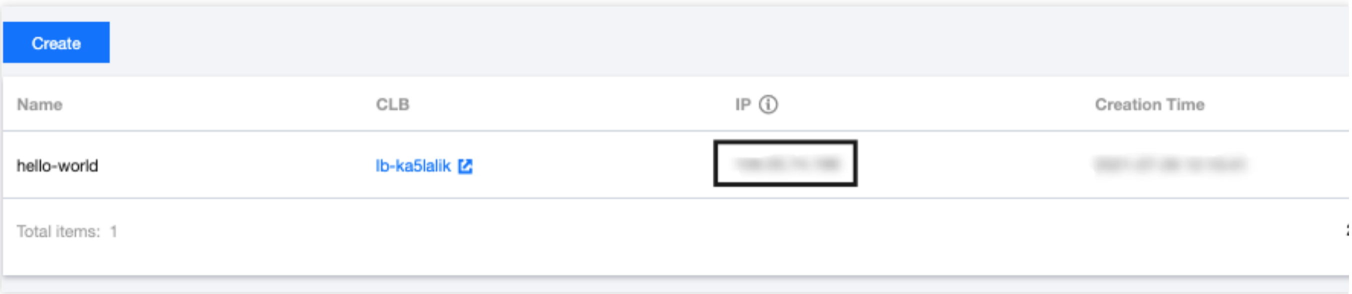
Path: It defaults to “/”. Configure it as needed.

Backend application: Select the demo application.

Application port: For the demo application, use port 8201.

Server certificate: If **HTTP** is selected, you need to select a server certificate. If existing certificates are not suitable, you can create one.

3. Click **Confirm** to complete the configuration. You can then check the IP for internet access in the list of CLB gateways.



Name	CLB	IP ⓘ	Creation Time
hello-world	lb-ka5lalik	[redacted]	[redacted]

Total items: 1

4. Enter the public IP of the application in a browser. If the following result is returned, the application is successfully deployed.

```
Hello World!
```

Deploying Microservice Application

Last updated：2024-01-09 12:00:35

Overview

This document describes how to deploy your microservice applications in TEM and make them mutually callable and accessible over the public network by using the provider/consumer demo applications developed in Spring Cloud with JAR file upload as an example.

Prerequisites

Create an environment and application.

Directions

Step 1. Get the demo application

TEM currently supports Nacos, Eureka (for existing users), and ZooKeeper registries. The following steps take the Nacos registry as an example. To demonstrate Spring Cloud microservice applications using a registry, we have prepared a set of provider/consumer demo applications using the Nacos registry. You can view their source code at GitHub or directly download them as JAR packages.

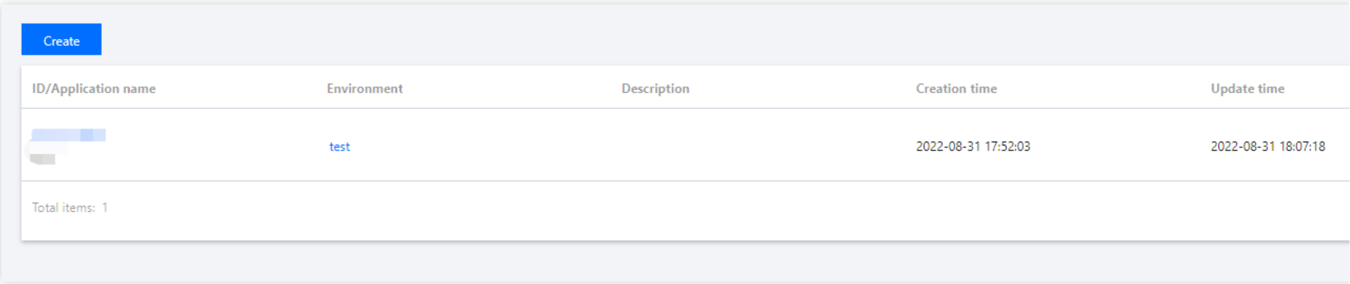
Nacos registry Demo application: Provider Demo and Consumer Demo。

Step 2. Configure the Registry

Before you start to deploy the Spring Cloud demo application, you need to configure the **registry** resource in the **environment**. Please add the registry corresponding to the application you select as instructed in Adding Environment Resource.

Step 3. Deploy the Application

1. On the **Application management** page, select the target application and click **Deploy to new environment**.

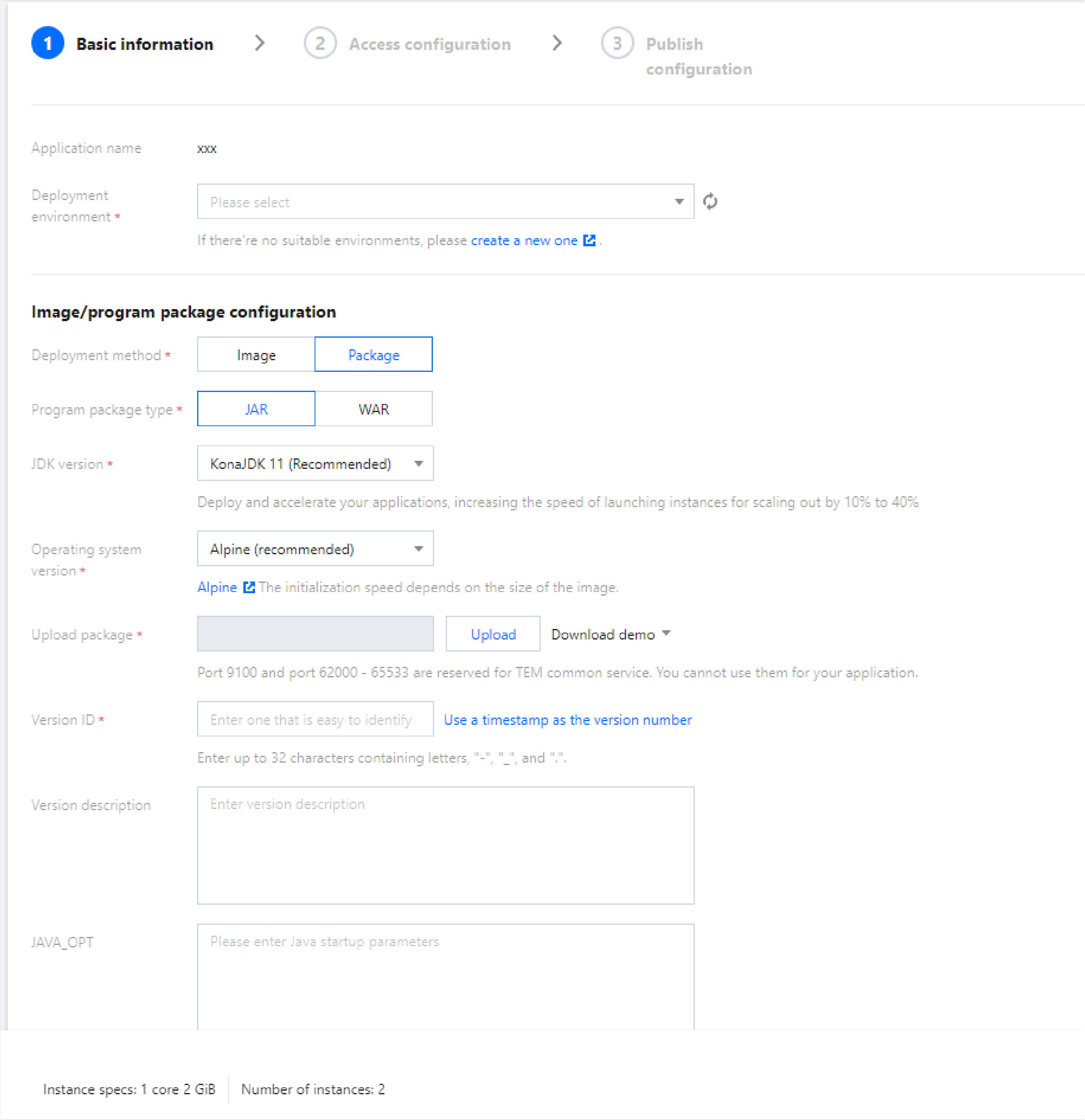


Create

ID/Application name	Environment	Description	Creation time	Update time
[redacted]	test		2022-08-31 17:52:03	2022-08-31 18:07:18

Total items: 1

2. On the **Deploy Application** page, configure the relevant parameters as needed.



Application name: xxx

Deployment environment *: Please select

If there're no suitable environments, please create a new one .

Image/program package configuration

Deployment method *: Image | Package

Program package type *: JAR | WAR

JDK version *: KonaJDK 11 (Recommended)

Deploy and accelerate your applications, increasing the speed of launching instances for scaling out by 10% to 40%

Operating system version *: Alpine (recommended)

Alpine The initialization speed depends on the size of the image.

Upload package *: Upload | Download demo

Port 9100 and port 62000 - 65533 are reserved for TEM common service. You cannot use them for your application.

Version ID *: Enter one that is easy to identify | Use a timestamp as the version number

Enter up to 32 characters containing letters, "-", "_", and ".".

Version description: Enter version description

JAVA_OPT: Please enter Java startup parameters

Instance specs: 1 core 2 GiB | Number of instances: 2

Upload Package: upload your package.

Release Environment: select the environment just created.

For information about advanced options, see Creating and Deploying Application.

3. For a Spring Cloud application, if a registry is associated with the selected **release environment**, you can select **Auto Inject Registry Info**. For more information, see Application Registration and Discovery.

4. Click **Deploy** to complete the deployment of the consumer application.

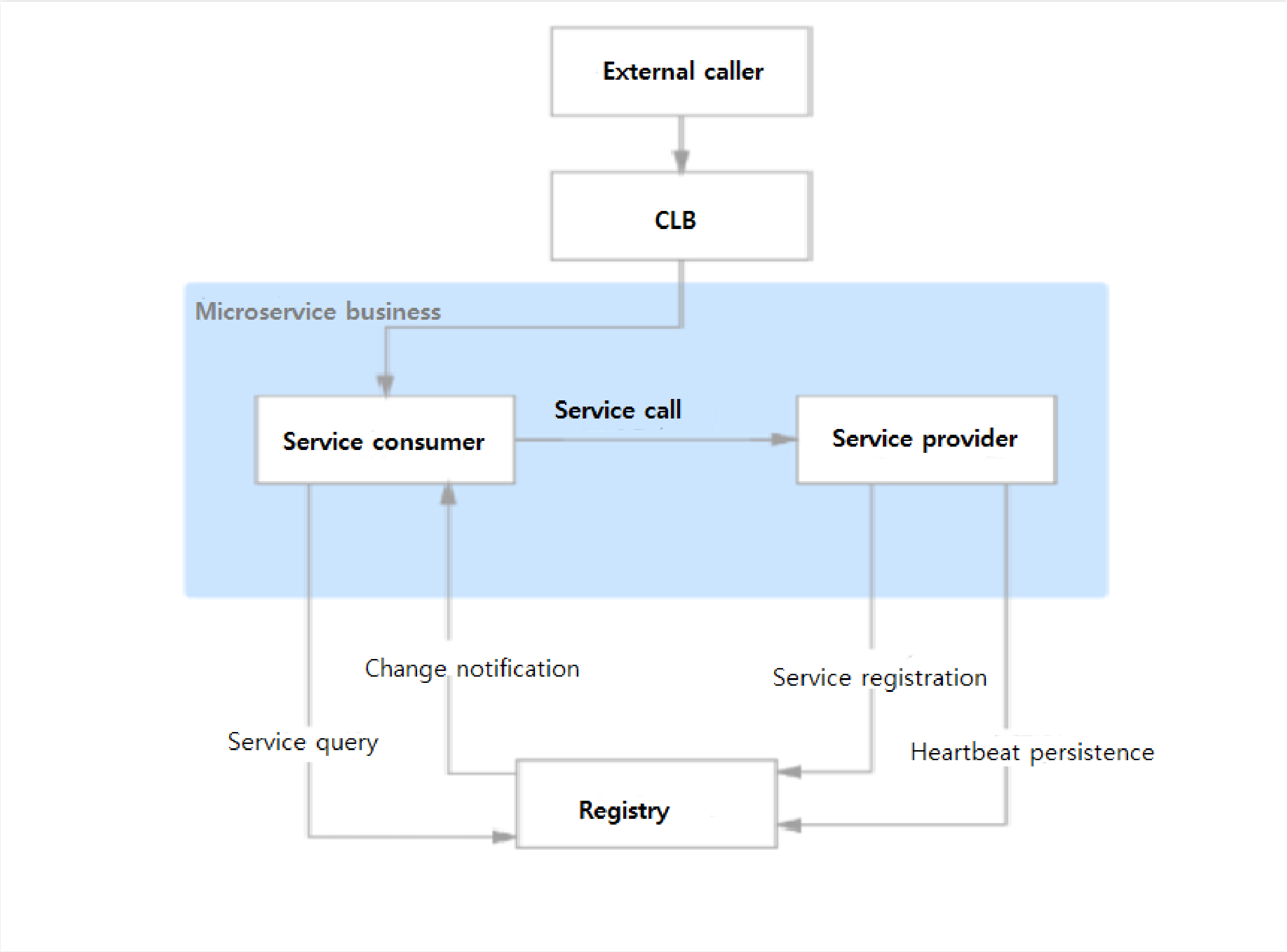
5. Repeat the four steps above to deploy the provider application.

Step 4. Check Registered Applications

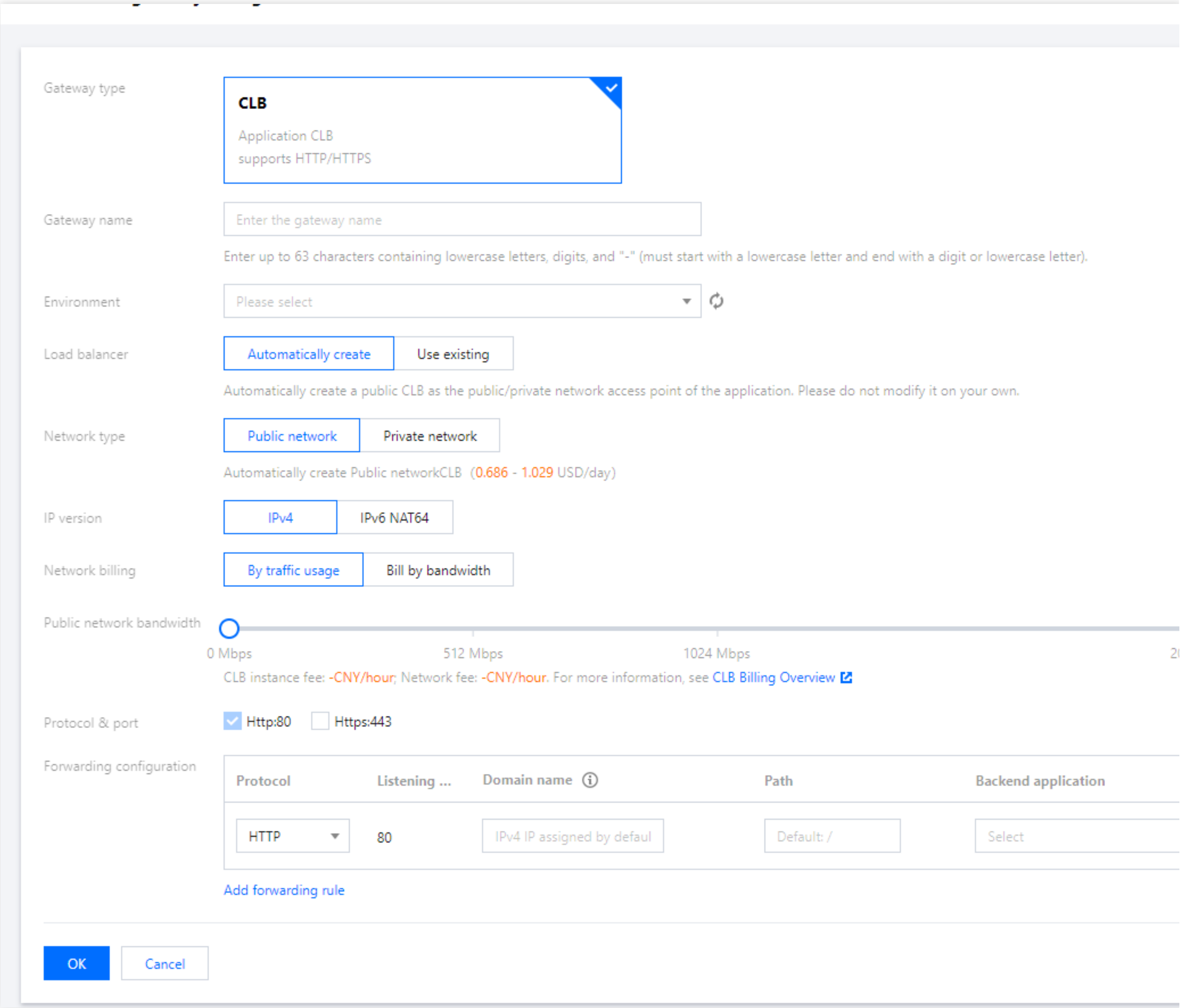
1. After the application instance to be deployed starts running, you can enter the Registry page in the TEM console and select the registry associated with your deployment environment.
2. On the registry details page, select the **Application Management** tab to check whether the provider and consumer applications are successfully registered.

Step 5. Verify the Access

After deployment of the provider and consumer applications, you can create an access configuration in the **CLB gateway** for the consumer application for internet access.



1. In the TEM console, click CLB Gateways in the left sidebar.
2. Click **Create** and enter the gateway configuration. Choose the correct **Environment**.



Gateway type: **CLB** ✓ Application CLB supports HTTP/HTTPS

Gateway name: Enter the gateway name

Enter up to 63 characters containing lowercase letters, digits, and "-" (must start with a lowercase letter and end with a digit or lowercase letter).

Environment: Please select

Load balancer: Automatically create | Use existing

Automatically create a public CLB as the public/private network access point of the application. Please do not modify it on your own.

Network type: Public network | Private network

Automatically create Public networkCLB (0.686 - 1.029 USD/day)

IP version: IPv4 | IPv6 NAT64

Network billing: By traffic usage | Bill by bandwidth

Public network bandwidth: 0 Mbps — 512 Mbps — 1024 Mbps — 2

CLB instance fee: -CNY/hour; Network fee: -CNY/hour. For more information, see CLB Billing Overview

Protocol & port: ☑ Http:80 □ Https:443

Forwarding configuration:

Protocol	Listening ...	Domain name	Path	Backend application
HTTP	80	IPv4 IP assigned by defaul	Default: /	Select

Add forwarding rule

Network Type: Select **Public Network**. To configure an intra-environment access, see Creating and Deploying Application.

Load balancer: Create one automatically or use an existing CLB.

Protocol & port: **HTTP:80** and **HTTPS:443** are supported, and HTTPS domain names can be bound to certificates. Select **HTTP:80** for the Demo application.

IP version: Options include **IPv4** and **IPv6 NAT64**.

Forwarding configuration:

Domain name: Choose an existing domain name. If it’s not specified, a random IPv4 IP is assigned by default. For the demo application, use the assigned default IP address.

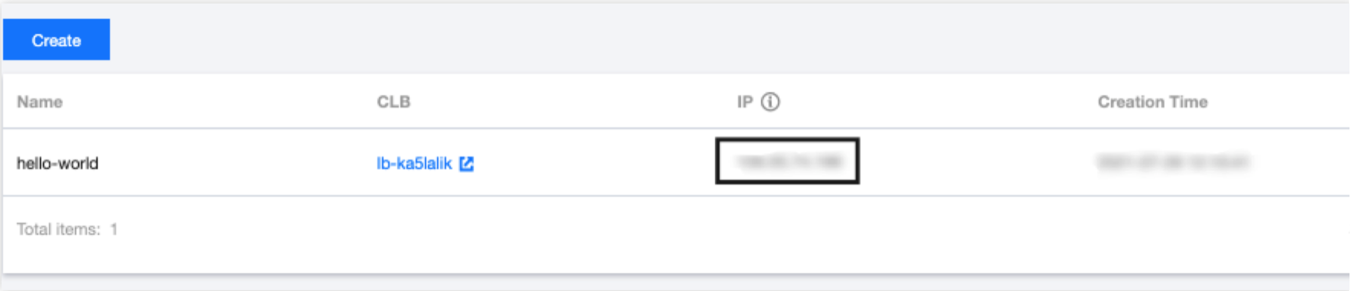
Path: It defaults to “/”. Configure it as needed.

Backend application: Select the consumer demo application you deployed.

Application port: Select port 19003 for the consumer demo application.

Server certificate: If **HTTP** is selected, you need to select a server certificate. If existing certificates are not suitable, you can create one.

3. Click **Confirm** to complete the configuration. You can then check the IP for internet access in the list of CLB gateways.



Name	CLB	IP ⓘ	Creation Time
hello-world	lb-ka5lalik	[redacted]	[redacted]

Total items: 1

4. Enter the following URL in a browser:

```
http://IP for internet access/echo/str
```

If the following result is returned, the application is successfully deployed.

```
Hello Nacos Provider str
```