NexaVM nKU

User Guide

Nexavm Technologies AG

NEXAVM

catalogs

1 Introduction

Readership

This document details how to use nKU. This document is primarily intended for the following audience:

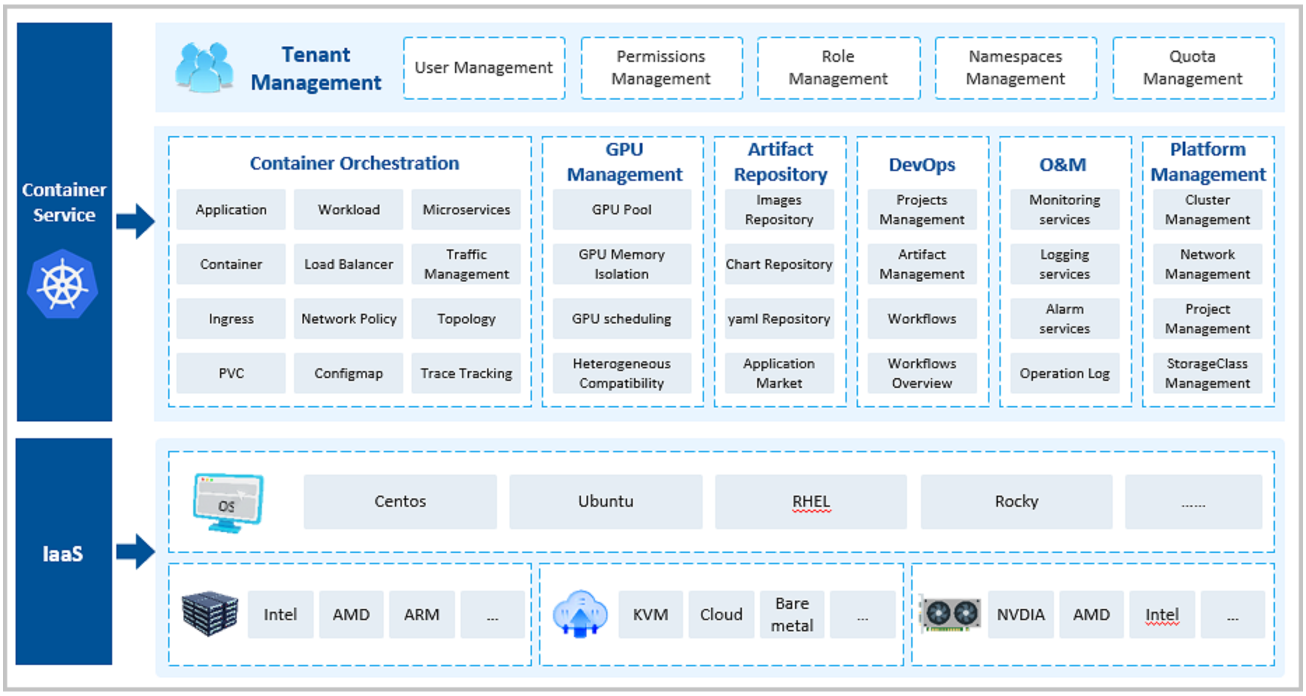
- Technical Support Engineer
- Deployment Operations Engineer
- Product Consulting Engineer
- For those interested in researching nKU

2 Product Overview

nKU takes "providing universal cloud-native products" as its core design concept, shields the complexity of container technology, simplifies it according to traditional user habits, deeply integrates computing/networking/storage/security and other basic components in a pluggable way, and provides enterprise-grade features such as multi-tenancy, multi-cluster, resource quota, CI/CD, micro-service governance, GPU management scheduling, disaster recovery, operation and maintenance management, etc. It allows enterprises to quickly get started and enjoy the resource optimization and efficiency improvement brought by cloud-native applications. It also provides multi-tenancy, multi-cluster, resource quota, CI/CD, microservices governance, GPU management and scheduling, disaster recovery and backup, operation and maintenance management, and other enterprise-level features, which enable enterprises to quickly get started and enjoy the resource optimization and efficiency improvement brought by cloud-native applications.

As shown in Figure 1: nKU Product Architecture:

Figure 1: nKU Product Architecture



Product Characteristics

- **Out-of-the-box, easy to operate and maintain**
  - nKU enables one-click installation, maximizes the shielding of underlying technical details, reduces O&M complexity, and provides an enterprise-grade

platform; it is deeply adapted to Cloud cloud platforms to achieve platform unified O&M.

  - nKU is built around the four elements of computing, storage, network and security with infrastructure as the core, and fully supports CPU and GPU resources of x86 and ARM architectures; it integrates core functions, including full lifecycle management of workloads, multi-tenant resource quota, multi-tier cluster management, monitoring and maintenance, service governance, and CI/CD, which makes it simpler to use containers, and makes it easier to use containers. A key to cross into the cloud native era.

- **Tenant autonomy and safety**
  - nKU supports three levels of privilege division, isolating users, applications and networks among tenants, which flexibly meets the application management and operation and maintenance in the cloud environment.
  - nKU has full lifecycle protection capabilities, covering pipeline scanning and detection during the application development phase, as well as repository detection during the image repository phase, and providing comprehensive defense capabilities at runtime.
  - nKU supports security capability coverage for full scenarios of cloud-native applications such as Images, runtimes, orchestration files, services/microservices, K8S clusters, container networks, etc. It also provides security operation capabilities such as big screen, asset management center, security compliance center, multi-tenant management view, and vulnerability operation center.
- **Cluster integrated**, **open and compatible**
  - nKU Cluster supports integrating external standard K8S clusters and supporting multi-cloud and multi-cluster, with no lock-in and all APIs open. nKU supports multiple clusters, associating multiple K8s engines, creating regular and cluster HA clusters, and supporting backup/recovery and upgrade operations. In addition, nKU also integrates public/private cloud K8s clusters regardless of vendor and version. For external integrated clusters, all it takes is one click to activate them, and the platform's management functions are synchronized to match the external clusters.
  - nKU fully supports mainstream letter architecture and is compatible with localized chips/operating systems. nKU adapts to mainstream local operating systems, and

completes compatibility certification with local chip vendors and hardware server vendors.

- **Application-driven**, **continuous delivery**
    - Covering the full life cycle management of application development and testing to improve iteration efficiency; providing full life cycle control of microservices governance through service mesh.
    - Support load of service governance, mainly in the update application - canary release; traffic management of various types of traffic routing, load policy settings; application access traffic visualization; application traffic link audit tracing tracking and so on.

3 Product Functions

nKU provides container orchestration, artifact repository, container operation and maintenance, cluster management and setup of a number of functional modules , this section focuses on the complete range of features provided by nKU is introduced .

Type	Characterization	Description
Dashboard	Resource Overview	Real-time monitoring and display of clusters, nodes, applications, workloads, jobs, worker containers, and Pod with allowed overviews
		Real-time statistics of multi-tenant, Repository information situation
	Unread alarm statistics for the past 7 days	Show unread alarm messages for the last 7 days
	Cluster Resource Utilization Statistics	Real-time monitoring and display of total cluster CPU and memory resource usage
		Showing Top 3 Cluster Rankings for CPU, Memory Usage
		Real-time monitoring and display of the cluster's CPU and memory resource utilization in the last 24 hours
	Resource utilization statistics within the cluster	Show the request and usage of GPU and GPU memory
		Showing Top 10 ranking of CPU, memory and pvc usage for Pod
container orchestration	application	Supports basic lifecycle management such as installing, deleting, etc. of applications
		Supports updating the application with the current deployment chart/new deployment chart and viewing the version comparison.
		Supports managing application history, including viewing, rollback, and deletion.
		Supports centralized management of application-related resources, including Pod, data volumes, services, and presses.
	workload	Supports Deployment, StatefulSet, and DaemonSet workload types.

		Supports visual configuration of health checks, environment variables, lifecycle callbacks for workloads
		Supports mounting persistentVolume templates (StatefulSet type only), pvc, hostpaths, configSets and Secret for workloads.
		Support for setting GPU configurations for workload Pod, including the number of GPUs used, GPU memory per GPU, and GPU memory percentage per GPU
		Supports selecting the GPU to be used by GPU specification, or directly specifying the GPU
		Supports multiple containers sharing a single GPU with isolated GPU memory and multiple GPUs for a single container.
		Support setting node scheduling policy, affinity, anti-affinity, and tolerance for workload Pod
		Support setting Update Policy for workload
		Support setting Network Configuration for workload, Configure and use the Pod External Network or the Pod External Network
		Supports basic lifecycle management such as creation, update, restart, and deletion of workloads
		Supports managing workload history, including viewing Yaml, rollbacks
		Supports modification of workload configurations via both Yaml and forms
		Support for modifying workload container images
		Supports elastic workload scaling, including manual scaling and autoscaling type of scaling
		Supports centralized management of workload-related resources, including: Pod, related services, etc.
		Support for centralized view of workload events
		Support exporting the workload list as a CSV file, facilitating statistical analysis
	job	Supports two types of jobs: Job, CronJob.
		Support for creating jobs via forms or templates
		Support for visually configuring health checks, environment variables for jobs
		Supports mounting persistentVolumeClaims, configmaps, and Secret for job

		Support for setting GPU configurations for jobs, including the number of GPUs used, GPU memory per GPU, and GPU memory percentage per GPU
		Supports selecting the GPU to be used by GPU specification, or directly specifying the GPU
		Supports multiple containers sharing a single GPU with isolated GPU memory and multiple GPUs for a single container.
		Supports basic lifecycle management such as creation, update, deletion, enablement and disablement of jobs
		Supports centralized management of job-associated resources, including: Pod, pvcs
	pod	Support for viewing and deleting Pod
		Supports centralized management of containers in a pod, including viewing Monitoring Data, container information, container logs, saving containers as an image, and so on.
		Support for accessing pod endpoints
		Support for centralized viewing of pod events
		Support exporting the pod list as a CSV file, facilitating statistical analysis
	service	Supports four service types: ClusterIP, NodePort, Load Balance and Headless.
		Supports customizing the external IP address for the LoadBalancer service.
		Supports basic lifecycle management such as creation, update, and deletion of services
		Supports modification of service configurations via both Yaml and forms
		Supports centralized view of service events
	ingress	Supports basic lifecycle management such as creation, update, and deletion of ingresses
		Supports centralized management of ingress rules, including adding, modifying, and deleting rules.
		Support for viewing routing gateway addresses and forward policy access urls
		Support for centralized view of ingress events

	network policy	Supports the creation of network policies for workloads to control the communication behavior of workloads and protect applications from network attacks.
		Supports basic lifecycle management such as creation and deletion of network policies
		Supports modification of network policy configurations via both Yaml and forms
	pvc	Supports distributed storage with pvcs for persistent storage of workload data.
		Supports basic persistentVolumes management such as creation, expansion, and deletion of pvc.
		Support for centralized view of pvc events
	configmap	Supports basic lifecycle management such as creation, update, and deletion of configmaps
	secret	Supports three types of Secret: Opaque, TLS certificate, and image repository login key.
		Supports basic lifecycle management such as creation, update, and deletion of Secret
	microservices application	A microservices application is a collection of resource objects such as workloads, services, and service mesh resources
		Supports basic lifecycle management such as creation and deletion of microservices applications
		Supports accessing microservices applications to the platform for visualization governance
		Supports grayscale release strategy for multiple microservices applications, can route traffic to new and old versions according to traffic weight and Request Content to ensure smooth application upgrades.
		Provide traffic monitoring for new and old versions during canary release process
	service topology	Provide visual application topology, intuitively displaying invocations, dependencies and traffic monitoring data between microservices applications
		Supports historical and real-time viewing of topology maps
		Supports configuration of load balancing, connection pooling, Circuit Breaking and other traffic management rules for microservices applications

	trace tracking	Provide trace tracking function, support to query the call chain between services in the current namespace by service name or TraceID.
		Comprehensively monitor the invocation status of services in the invocation chain, invocation time consumption and other key metrics
artifact repository	repository	Support for public and private repository types
		Supports basic lifecycle management such as creation, modification, and deletion of local repositories
		Support centralized management of image repository, including uploading, exporting, downloading images, and deleting image tags.
		Supports 3 types of Image upload: online upload, file upload, command line upload
		Support centralized management of Chart deployments in the repository, including uploading and downloading Chart, viewing and deleting Chart versions, and installing applications based on Chart.
		Support to set the project to which the Repository belongs to
	YAML template repo	Template repository for managing YAML templates
		Provide a variety of resources for reference, example YAML templates, users can create templates based on the example directly
		Supports basic lifecycle management such as creation, modification, and deletion of templates
	application market	Provide zookeeper, kafka, mysql, redis,deepseek, etc. official Charts
		Supports one-click publishing of Chart deployment packages from the application market to deploy as application instances
Container O&M	One-Click Inspection	Provide three major inspection categories: basic services, computing, and networking, covering key resources and services of both platform management clusters and business clusters.
		Has three built-in inspection levels: Normal, Warning, and Fault, as well as a four-level health scoring mechanism (inspection resources, inspection items,

		clusters, and platforms), displaying the overall health scores of clusters and the platform
		Supports providing inspection suggestions for inspection resources in Warning and Crashed states
		Provides inspection overview, inspection summary, and inspection results, and supports viewing details of abnormal inspection items as well as relevant inspection suggestions
		Supports one-click inspection without login
		Supports custom selection of clusters and inspection items by category for one-click inspection
		Supports setting up automatic inspection to perform scheduled checks on the platform
		Supports operations such as canceling one-click inspection, re-inspecting, and exporting Word inspection reports
	monitoring panel	Supports one-click jump to monitoring panel to view detailed monitoring data of the cluster
		Supports viewing the overall resource usage of the cluster, including CPU utilization, memory utilization, network traffic, etc.
		Support for viewing the CPU and memory resource usage of Pod on nodes
		Support for viewing workload resource utilization
		Support for viewing network resource usage for workloads, Pod
	log panel	Provide a unified portal for users to centrally view all container logs in the cluster
		Supports querying logs by cluster, namespace, workload, pod, container, keyword, time range
		Support for exporting logs
		Supports viewing log contexts; supports searching contexts by keywords; supports exporting log contexts
	Alarm messages	Supports centralized viewing and management of all alarm messages on the platform
		Support filtering alarm messages by resource name, time period
		Supports marking alarm messages as read

	alarm	Supports basic lifecycle management such as creating, enabling, disabling, deleting, etc. of alarms
		Supports three Emergency Levels: Emergency, Critical, and Alert
		Supports on-demand turn-on of recurring alarm notifications
		One alarm supports monitoring of multiple resources at the same time
	Endpoint	The Endpoint is the way for users to receive alarm messages, and supports four types of Endpoint: system, email notification, enterprise WeChat, and nails.
		Supports basic lifecycle management such as creation, modification, and deletion of Endpoint.
	current task	Supports centralized viewing and management of in-progress operations
		Support for viewing current job details
	Operation History	Support for centralized view of historical jobs
		Supports filtering historical Task by time period, job results, and operation source
		Support for viewing historical job details
cluster management	cluster	Supports basic lifecycle management clusters such as creation, retry installation, deletion, etc. of multiple clusters
		Support for Kubernetes Cluster HA
		Supports customized deployment of functional components in clusters，include：Microservices Governance Component, ZStone-RBD-CSI, ZCE-iSCSI-CSI, ZCE-NFS-CSI
		Support for Kubernetes version 1.24~1.30 of NAMI's Kubernets clusters
		Support CPU, GPU heterogeneous clusters unified construction management and resource unified planning and scheduling
		Support for node heterogeneity, nodes in the same cluster support the use of different operating systems and architectures
		Supports real-time view of cluster CPU, memory, storage and other resource utilization

		Support setting the retention time of container logs in the cluster and the size of the data disk for container logs.
		Support monitoring and alerting on the usage rate of the container log data disk, and automatically enables write protection when the threshold is reached, preventing service abnormalities caused by the data disk being full.
	Node Auto-Scaling Group	Supports automatic triggering of node elastic scaling based on Pod Pending status, CPU/memory usage rate, etc
		Can be associated with node selectors and node affinity/anti-affinity configurations to ensure scaled-out nodes meet Pod scheduling requirements
		Supports basic lifecycle management of node elastic scaling groups, such as creat, modify basic configuration, modify node configuration, fault recovery, and delete
		Supports viewing and exporting scaling logs of node auto-scaling groups
	node	Supports basic lifecycle control plan such as adding and deleting nodes.
		Supports stop/resume node scheduling to enable scheduling for node maintenance
		Supports real-time view of node's CPU/memory utilization rate, request rate, and limit rate, which facilitates timely understanding of node's resource allocation situation
		Supports centralized viewing of GPU information on nodes, including basic GPU information, memory usage, scheduled container pod information, real-time monitoring data, etc
		Support exporting the workload list as a CSV file, facilitating statistical analysis
	GPU pool	Supports viewing all GPUs under the current cluster by cluster, and the total display usage of all GPUs under the cluster, including: GPU memory utilization, GPU memory request rate, GPU utilization

		Support to view detailed information of GPU manufacturer, model, memory, memory utilization , memory request rate, GPU utilization, GPU request rate,etc.
		Supports visual monitoring of GPU, including: GPU utilization, GPU memory utilization, power consumption, temperature
		Supports centralized view of information about scheduled pods on GPUs
		Supported GPU manufacturers include: NVIDIA,Huawei,HYGON,Ilutavar
	storage class	Supports basic lifecycle management of storage class creation, update, deletion, etc.
	external network	Supports basic lifecycle management such as create and delete of external networks
		Supports viewing external network IP usage to improve network planning efficiency
		Supports two types of network segments: external service network and external pod network
		The external pod network supports direct access to the pod IP from outside the Kubernetes cluster
		Supports adding and removing network segments for external networks
		Supports setting up network segment sharing mode to allow global or designated projects to use network segment resources
	project	Supports basic lifecycle management such as project creation, editing, deletion, etc.
		Support for adding members and namespaces to projects and centralized management
		Supports resource segregation and privilege control through projects
	user	Support for local and 3rd-party users
		Supports basic lifecycle management such as creating, enabling, disabling, deleting, and resetting passwords for local users
		Supports basic lifecycle management such as synchronization and deletion of 3rd-party users

		Support for setting users as administrators to gain access to the platform
		Support for removing users as administrators
		Support for adding users to a project to get project permissions
		Support for removing users from projects
	namespace	Supports basic lifecycle management such as creation and deletion of namespaces
		Supports setting namespace resource quotas to effectively control user/team resource usage
		Supports setting GPU memory quota and GPU Count quota for namespaces by manufacturer to effectively control user/team GPU resource usage.
		Support for setting namespace default resource limits
		Support for adding namespaces to/removing them from projects
		Support for adding labels, annotations to namespaces
Settings	email server	Supports basic lifecycle management such as adding, modifying and deleting email servers.
		Supports STARTTLS, SSL/TLS and NONE encryption types.
	Theme Appearance	Support customized setting of platform theme appearance, such as appearance theme, platform title, Logo, etc.
	3rd-party authentication	Supports seamless access to 3rd-party authentication systems by adding 3rd-Party Authentication Servers to enable corresponding accounts to log in to the cloud platform without confidentiality.
		Support adding 3rd-Party Authentication Server Type: OIDC,CAS
		Supports setting synchronization mapping rules for 3rd-Party Authentication Servers, including: user mapping rules
		Supports basic lifecycle management such as adding, modifying, and deleting 3rd-Party Authentication Servers.
	AccessKey Management	Provide identity credentials to access the platform API with full creator permissions

		Supports basic lifecycle management such as generation, enabling, disabling, and deletion of AccessKey.

4 System Login

Users can access the nKU UI interface using a browser and log in using their account password. For the best experience, Chrome 66 and above is recommended.

Login Platform

The platform supports the following two login portals:

- **Account login**: local users log in from this portal
- **Unified identity authentication login**: 3rd-Party users login from this portal, you need to configure the 3rd-Party Authentication Server in advance in the settings> 3rd-Party Authentication Kind.

Manage Login

After logging in to nKU, you can click your username and avatar on the far right side of the main menu to expand the Personal Center, which supports the following login-related administrative operations:

- **Change Password**: Change the login password of the current user, when changing, you need to verify the old password. The updated password will take effect at the next login.

  Note: Password change is not supported for third party users.
- **Logout**: Ends the user's current session and returns to the login screen.

5 License management

5.1 General

To use the nKU feature, you need to apply for a license and install the license in advance.

In the nKU main menu, click **Personal Center** > **License Management** to enter the **license management** interface. This interface displays information about the current container service's license agreement, software version, and license status, and provides unified management of licenses.

Note: Only admin and administrator have license upload and management privileges.

Description of the authorization agreement

nKU offers a base license that authorizes all features of the platform and includes two licensing options:

- **vCPU Authorization**: authorizes the number of VM instances to be used in scenarios where the business cluster compute nodes are virtual machines.
- **CPU Authorization**: authorizes the number of physical CPUs slots for scenarios where the business cluster compute nodes are physical machines.

5.2 Installation of licenses

Request for authorization

1. Download request key.

   In the nKU main menu, click **Personal Center** > **License Management** to enter **the license management** interface. In the **License Management** interface, click **the Download** button after **the request key** to download the request key locally.

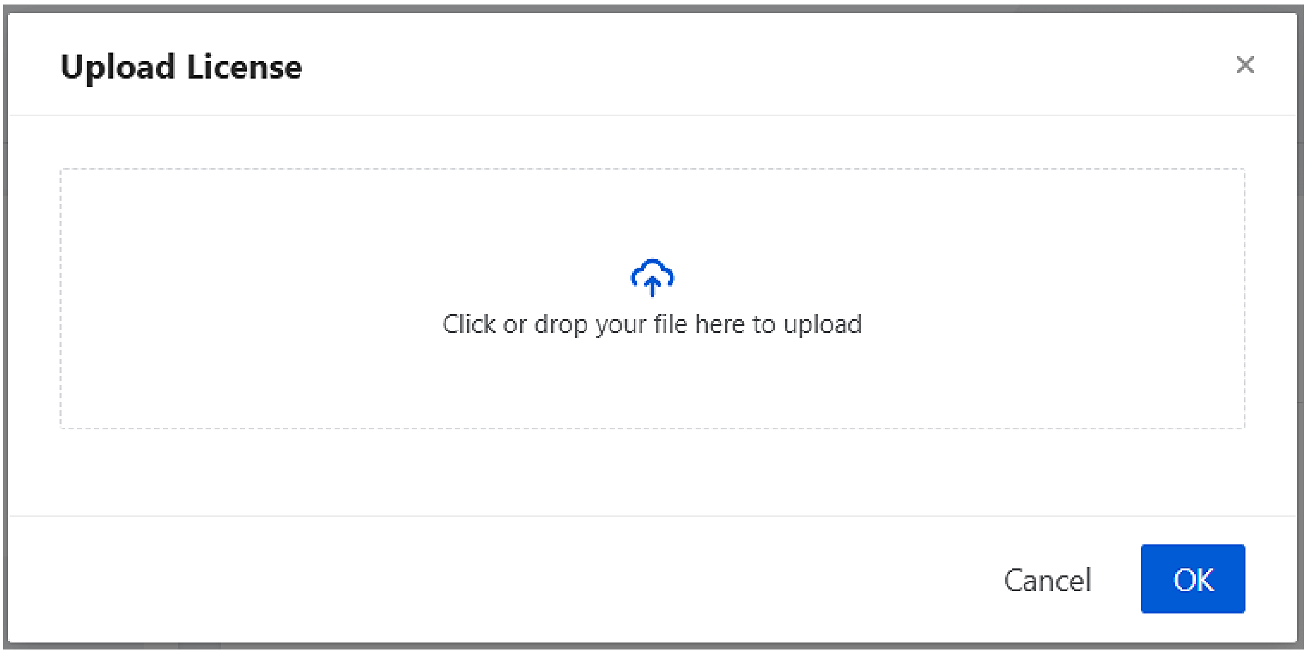
2. Send a request key to request authorization.

Uploading a license

In the **license management** interface, click **Upload License** in the upper right corner to bring up the **Upload License** interface, and upload the acquired license here.

As shown in Figure 4: Uploading a License:

Figure 4: Uploading a license



Upload License

Click or drop your file here to upload

Notes.

- After the license has been uploaded, it needs to be reloaded by clicking the Refresh button in the upper right corner of the **license management** screen.
- Only admin and administrators have permission to upload and manage licenses.

5.3 Managing licenses

In the nKU main menu, click **Personal Center** > **License Management** to enter **the license management** interface.

The license supports the following operations:

manipulate	**descriptive**
Upload license	Add licenses through the GUI.
download request key	Download the request key required to generate the license and use the request key to request/renew the license.
Refresh license	Click the Refresh button to reload the license.

Delete licenses	Click the Delete License button to delete the license. **Note:** After deleting the license, all the functions of the base license will be unavailable, please be careful.

Description of license status

The following statuses exist for the license:

state of affairs	brief
efficiently	When a valid license is added, the word "Valid" will be marked in the corresponding position, and then you can use the functions of the license pair normally.
Expires in XX days	When the remaining period of use of a license is less than 15 days, a banner will appear after logging into the base license. When the banner appears, you can click to jump to the **license management** interface to check the details. In order not to affect the normal use, please contact the official technical support to update the license in time.
expired	If the base license expires, the corresponding position will be marked with the word expired, in order not to affect the normal use, please contact the official technical support to update the license in time. **Notes.** •When a base license for the Platform expires, you cannot continue to use any of the features on the Platform. • When a base license expires, the business that has been created on the platform will continue to operate normally and will not be affected.
Insufficient authorization of quotas	When it detects that the actual usage of resources of any one of the licensing methods exceeds the license authorization quota, it will limit the use of the base functions. In this case, please contact the official technical support to update the license and expand the capacity in time.
exceptions	When an abnormal base license is detected, or the license management service is abnormal, the use of the platform functions will be restricted. In this case, please contact the official technical support as soon as possible.

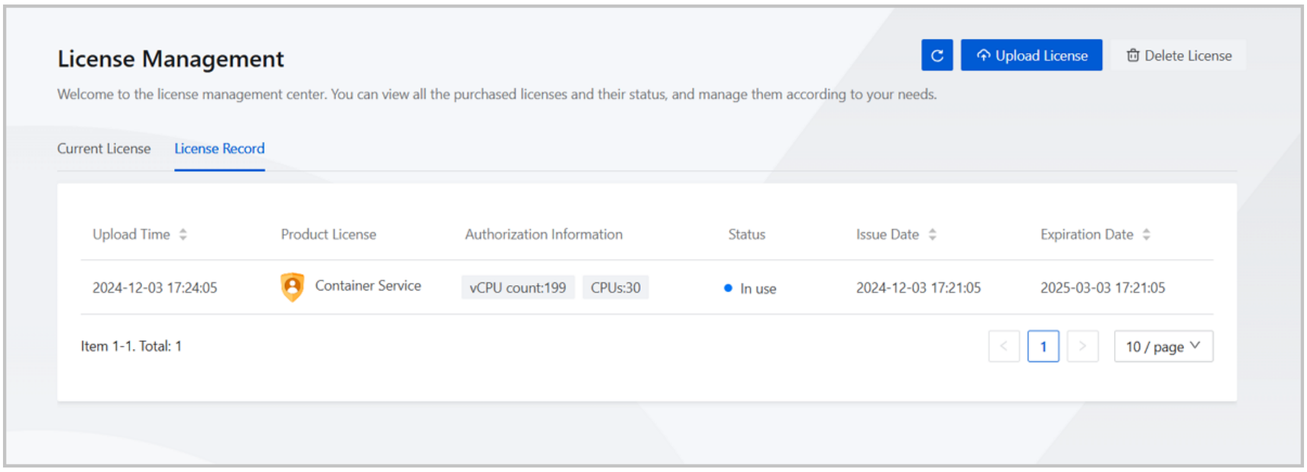
License Record

In the nKU main menu, click **Personal Center** > **License Management** > **License Record** to access the **License Record** screen.

This interface visually displays the platform's authorization history, including: license upload time, product license, authorization information, status, license issuance time, and license expiration time.

As shown in Figure 5: License Record:

Figure 5: License Record



License Management

Welcome to the license management center. You can view all the purchased licenses and their status, and manage them according to your needs.

Upload Time	Product License	Authorization Information	Status	Issue Date	Expiration Date
2024-12-03 17:24:05	Container Service	vCPU count:199 CPUs:30	● In use	2024-12-03 17:21:05	2025-03-03 17:21:05

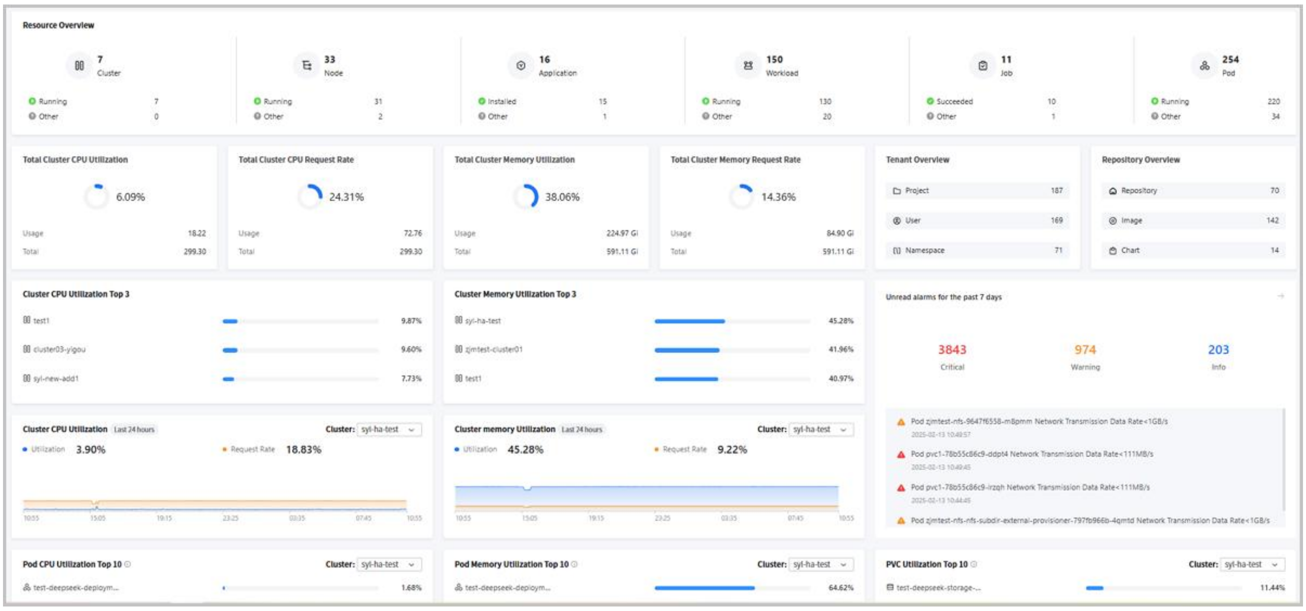
Item 1-1. Total: 1

6 Dashboard

Log in to nKU or click **Home** from the nKU main menu to access **the** Dashboard screen.

As shown in Figure 6: Dashboard:

Figure 6: Dashboard



Pod zjmtest-nfs-9647f6558-m8pmm Network Transmission Data Rate<1GB/s
2025-02-13 10:48:57

Pod pvc1-78b55c86c9-ddpt4 Network Transmission Data Rate<111MB/s
2025-02-13 10:49:45

Pod pvc1-78b55c86c9-lrzqh Network Transmission Data Rate<111MB/s
2025-02-13 10:44:45

Pod zjmtest-nfs-nfs-subdir-external-provisioner-797fb966b-4qmtd Network Transmission Data Rate<1GB/s

The nKU home page displays information in the form of cards such as resource overview, total cluster resource utilization, TOP ranking of cluster resource utilization, 24-hour monitoring of cluster resource utilization,GPU and memory utilization of cluster, TOP ranking of cluster pod resource utilization, TOP ranking of pvc utilization within the cluster, and statistics of unread alarms in the last 7 days.

- Dashboard automatically refreshes every 10 seconds to get the latest data and display it in real time.
- Status statistics charts adopt a unified color specification, with green indicating normal status, red indicating abnormal status, and grey indicating other statuses, making it easy for users to quickly identify the health status of resources.
- The percentage progress bar consists of three colors: blue, yellow, and red to visualize the current state of resource usage. The color for usage are: blue (less than 80%), yellow (greater than or equal to 80% and less than 90%), and red (greater than or equal to 90%).

7 Container orchestration

7.1 Application management

7.1.1 Application

7.1.1.1 General

An application is an instance installed by the Helm tool, a collection of resource objects such as workloads, services, storage, and so on. Support for unified resource management through applications.

Relevant definitions

The application involves the following core concepts:

- **Helm**: An open source tool initiated by Deis, **Helm** is a Kubernetes package manager that makes it easy to discover, share, and use Kubernetes-built applications.nKU integrates the Helm 3.0 tool and extends it to simplify the complexity of distributing and deploying Kubernetes applications, enabling visual application publishing, upgrades, rollbacks, versioning, and more.
- **Deployment packages**: i.e., Helm Chart. the Chart package describes the workloads, container images, dependencies, and resource definitions required to deploy and run an application, and may also contain definitions of services in a Kubernetes cluster. the relationship between Helm and Chart is similar to that of an APT and a DPKG, or a YUM and an RPM file. Users can use Chart to deploy a complete application.

7.1.1.2 Installing application

Publishing an application means publishing the Chart to the cluster and deploying an application instance. Users can install application in two ways in the **application** or **artifact repository** interface. This section focuses on the way of publishing through the **application** interface.

In the nKU main menu, click **Container Orchestration** > **Application Management** > **Applications** to enter the **application** interface. Click **Publish Application** to bring up the **Publish Application** screen.

You can refer to the following example to enter the appropriate contents:

- **Application Name**: Set the name of the application. Naming rules: length limit 2-20 characters, input can only contain lowercase letters, numbers and separator (-), and must start with a lowercase letter and end with a lowercase letter or number.
- **Version**: Setting the application version
- **Description**: optional, note application-related information
- **Cluster**: Select the cluster to which the application will be installed
- **Namespace** : Select the namespace that the application will install application to.
- **Chart Repository**: Select the source of Chart repository, you can choose Repository or application market

  If you choose **Repository**, please set the following parameters:

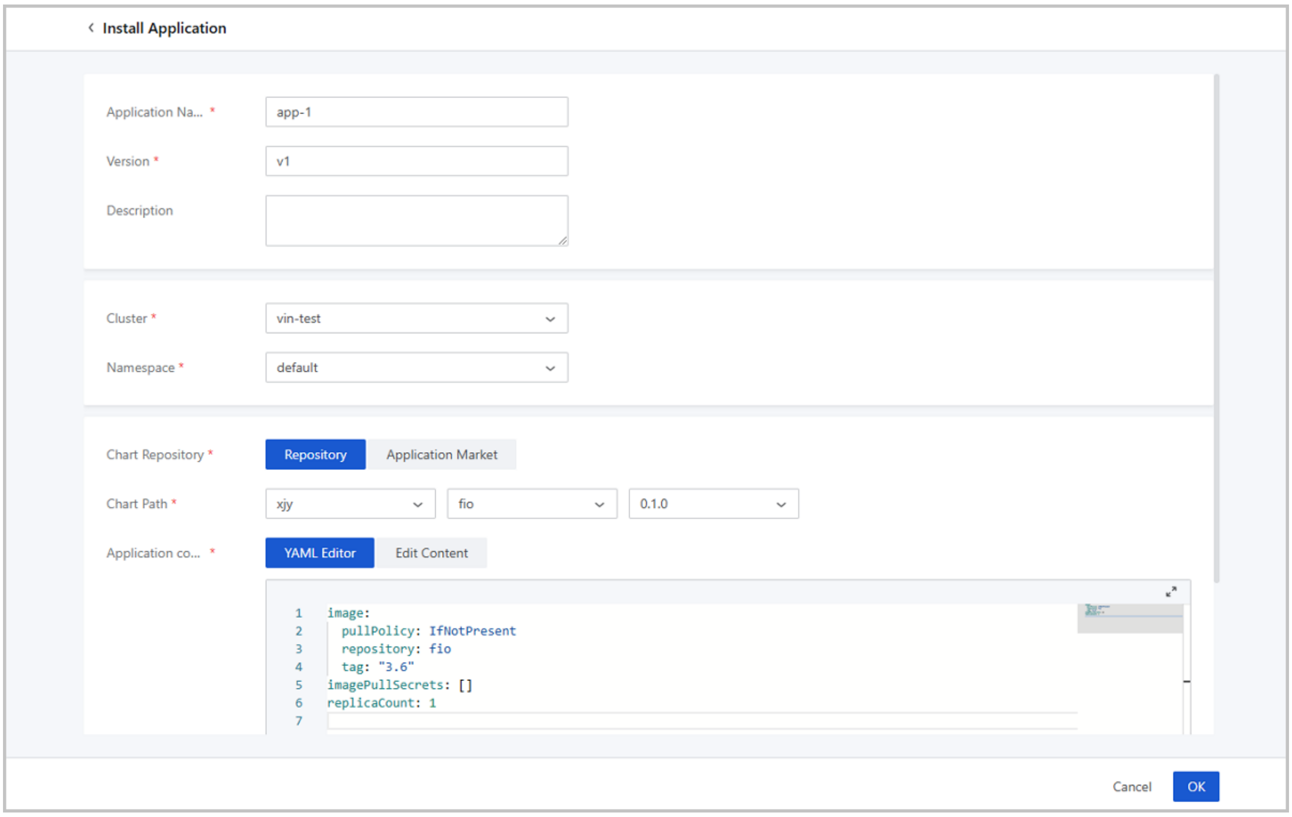
  - **Chart path:** select the path to locate the Chart
  - **Application Configuration:** Modify the configuration of the values file in the deployment chart. The deployment chart of the Repository only supports YAML editing as a modification method. Users can click on the modified content to see before and after the modification.

  If you choose **Application Market**, please set the following parameters:

  - **Chart**: select a chart
  - **Application Configuration**: Modify the configuration content of the VALUES file in the deployment chart. The deployment package of the application market supports two modification methods: **editing charts** and **YAML editing**. Users can click on **the modified content** to see the comparison between before and after modification

AS shown in Figure 7: Installing application| Chart Repository: repository, Figure 8: Installing application| Chart Repository: application market:

Figure 7: Installing application| Chart Repository: repository



Application Na... * app-1

Version * v1

Description

Cluster * vin-test

Namespace * default

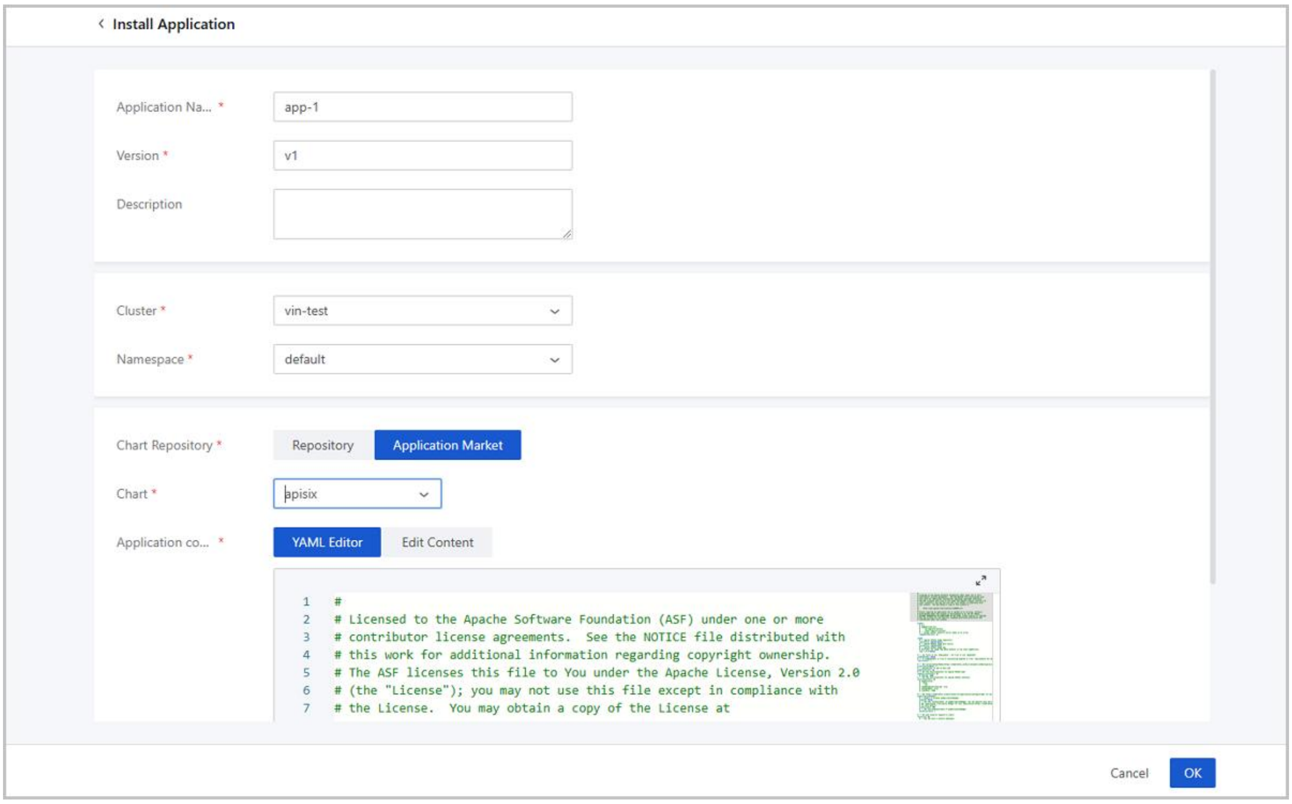
Chart Repository * Repository | Application Market

Chart Path * xjy | fio | 0.1.0

Application co... * YAML Editor | Edit Content

```
image:
  pullPolicy: IfNotPresent
  repository: fio
  tag: "3.6"
imagePullSecrets: []
replicaCount: 1
```

Figure 8: Installing application| Chart Repository: application market



Application Na... * app-1

Version * v1

Description

Cluster * vin-test

Namespace * default

Chart Repository * Repository | Application Market

Chart * apisix

Application co... * YAML Editor | Edit Content

```
#
# Licensed to the Apache Software Foundation (ASF) under one or more
# contributor license agreements.  See the NOTICE file distributed with
# this work for additional information regarding copyright ownership.
# The ASF licenses this file to You under the Apache License, Version 2.0
# (the "License"); you may not use this file except in compliance with
# the License.  You may obtain a copy of the License at
```

7.1.1.3 Management applications

In the nKU main menu, click **Container Orchestration** > **Application Management** > **Applications** to access the **Application** interface.

The application supports the following operations:

manipulate	descriptive
install application	Publish the Chart to the cluster and deploy an instance of the application.
update application	Update the application configuration based on the current deployment chart, or update the application with a new deployment chart. **Notes.** • Applications that are not console installations or updates do not support updates based on the current deployment chart. • Updates based on the current deployment chart are not supported when the current deployment chart has been deleted.
Application Details	Go to the Application Details page to view basic application information, version history, configuration resources, and related services, including: Pod, pvc, services, and ingress.
Rollback application	Rolls back the application to the specified version. **Notes.** • To perform this action, go to the application details page and click **History Version** > **Action** > **Rollback**. • Rollback to an **enabled** or installed Status is only supported.
Delete historical versions	Delete the historical version of the application. **Note:** To perform this operation, go to the application details page and click **History Version** > **Operation** > **Delete**.

Delete application	Remove the application from the cluster. **Notes.** • After deletion, the application-related resource objects are deleted synchronously. • PVCs created in the application through volumeClaimTemplates are not deleted synchronously. To delete, go to the list of pvcs and perform the delete operation separately.

7.1.2 Workload

7.1.2.1 General

A workload is an application running on Kubernetes that manages a set of Pod using the same image, including Deployment, StatefulSet, and DaemonSet types.

Relevant definitions

Workloads involve the following core concepts:

- **Workload types**: nKU supports three workload types:
  - **Deployment:** A workload of the Deployment type declares pod templates and runtime policies, and is primarily used to manage stateless applications, that is, individual applications that are completely independent but functionally identical.Each pod under Deployment has identical configuration except for different names and IP addresses.
  - **StatefulSet:** StatefulSet type workloads support orderly deployment, deletion, and persistent storage of containers, and are primarily used to manage stateful applications where the roles (or functions) of each application are different, such as a database master/backup role.StatefulSet type workloads have the following characteristics:
    - StatefulSet workloads provide a uniform, fixed name for each pod and add a 0-N suffix to the uniform name for resource differentiation. The name and HostName do not change after the pod is rescheduled.
    - The StatefulSet workload provides a fixed access domain name for each pod through the Headless service.
    - StatefulSet workloads create PVCs (PVCs) with fixed identifiers for persistent datastores so that the same data remains accessible after a pod is rescheduled.

  - **DeamonSet:** DeamonSet type workloads run one pod on each node of the cluster and only one pod, and are mainly used to manage system level applications such as logging collection, resource monitoring/logging etc. These types of applications need to run on every node, but do not require too many Pod.

- **Pod:** A pod is a collection of containers and is the smallest unit for scheduling and management in a Kubernetes cluster. Workloads manage a set of pods that use the same image, and the application actually runs in the pod.
  - **Replicas:** The number of pods that a workload contains. A Deployment or StatefulSet type workload can contain up to 100 pods.
  - **Containers**: Containers are the building blocks of pods. Normally, a pod contains only one container, but if the application requires multiple containers to work together and share resources, you can add multiple containers to a pod.
    - **Worker container**: In a pod, the worker container is responsible for running the actual application.
    - **Init container**: In a pod, the init container is responsible for performing initialization tasks and running to completion before the pod is ready. The worker container will only start after the init container has finished running. The init container can be used in the following scenarios:
      - Prepare the prerequisites for a group of applications to run through the init container, and only when the prerequisites are met will the worker container start and run the application.
      - The init container prepares utilities or configurations that do not need to be installed in the worker container and allows secure execution of utilities and custom code, reducing the risk of attacks on the worker container.

  Note: Init container does not support turning on livenessProbe, readinessProbe, startupProbe, and lifecycle callbacks.

7.1.2.2 Creating workloads

In the nKU main menu, click **Container Orchestration** > **Application Management** > **Workloads** to enter the **workload** interface. Click **Create Workload** to bring up the **Create Workload** screen.

Creating a workload is divided into the following five steps:

1. Basic Information

You can refer to the following example to enter the appropriate contents:

- **Name**: Set the name of the workload. Naming rules: length limit 2-50 characters, input can only contain lowercase letters, numbers and separator (-), and must start with a lowercase letter and end with a number or a lowercase letter

  Note: You cannot use the same name for the same type of workload under the same namespace.

- **Type**: Select the type of workload, support Deployment, StatefulSet, DaemonSet three types, the type of difference in detail see the Workload types
- **Replicas**: Sets the number of Pod the workload contains, up to a maximum of 100 replicas are supported. workloads of type DeamonSet do not need to set the number of replicas.
- **Workload Label**: set the workload label. The system will add two labels app, workloadType for the workload by default, the value of the label corresponds to the workload name and workload type respectively. These two labels cannot be edited or deleted. Users can also customize to add other labels, please set the following parameters:

# **key:** fill in the label key name

Notes.

  - Typically, key names can contain 1-63 characters, including letters, numbers, and the following special characters: (-)(_)

    (.) , special characters cannot be used at the beginning or end of a key name.

  - Supports prefixing key names and concatenating them with /, e.g. k8s.io/app, all characters filled in before / will be considered as key prefixes and all characters after will be considered as key names.
  - The key name prefix can contain 1-253 characters, including lowercase letters, numbers, and the following special characters: (-)(_)

    (.) , you cannot use 2 or more special characters in a row, and you cannot use special characters at the beginning or end of a prefix.

  - Please note that you do not have to use a key name prefix, but you must fill in the key name.

# **value**: fill in the value of the label

- **Workload Annotation**: optional, add workload annotations

  # **key:** fill in the annotation key name

  Notes.

  - Typically, key names can contain 1-63 characters, including letters, numbers, and the following special characters: (-)(_)(.) Special characters cannot be used at the beginning or end of a key name.
  - Supports prefixing key names and concatenating them with /, e.g. k8s.io/app, all characters filled in before / will be considered as key prefixes and all characters after will be considered as key names.
  - The key name prefix can contain 1-253 characters, including lowercase letters, numbers and the following special characters: (-)(_)(.) (-)(_)(.), 2 or more special characters cannot be used consecutively, and special characters cannot be used at the beginning or end of the prefix.
  - Please note that you do not have to use a key name prefix, but you must fill in the key name.

  # **value**: fill in the value of the annotation

- **Pod Label**: set the Pod Label. The system will add two labels app, workloadType to the pod by default, the value of the label corresponds to the workload name and workload type respectively. These two labels cannot be edited or deleted. Users can also customize to add other labels, please set the following parameters:

  # **key:** fill in the label key name

  Notes.

  - Typically, key names can contain 1-63 characters, including letters, numbers, and the following special characters: (-)(_)(.) Special characters cannot be used at the beginning or end of a key name.
  - Supports prefixing key names and concatenating them with /, e.g. k8s.io/app, all characters filled in before / will be considered as key prefixes and all characters after will be considered as key names.
  - The key name prefix can contain 1-253 characters, including lowercase letters, numbers and the following special characters: (-)(_)(.) (-)(_)(.), 2 or

more special characters cannot be used consecutively, and special characters cannot be used at the beginning or end of the prefix.

- Please note that you do not have to use a key name prefix, but you must fill in the key name.

# **value**: fill in the value of the label

- **Pod Annotation**: optional, add pod annotations

# **key:** fill in the annotation key name

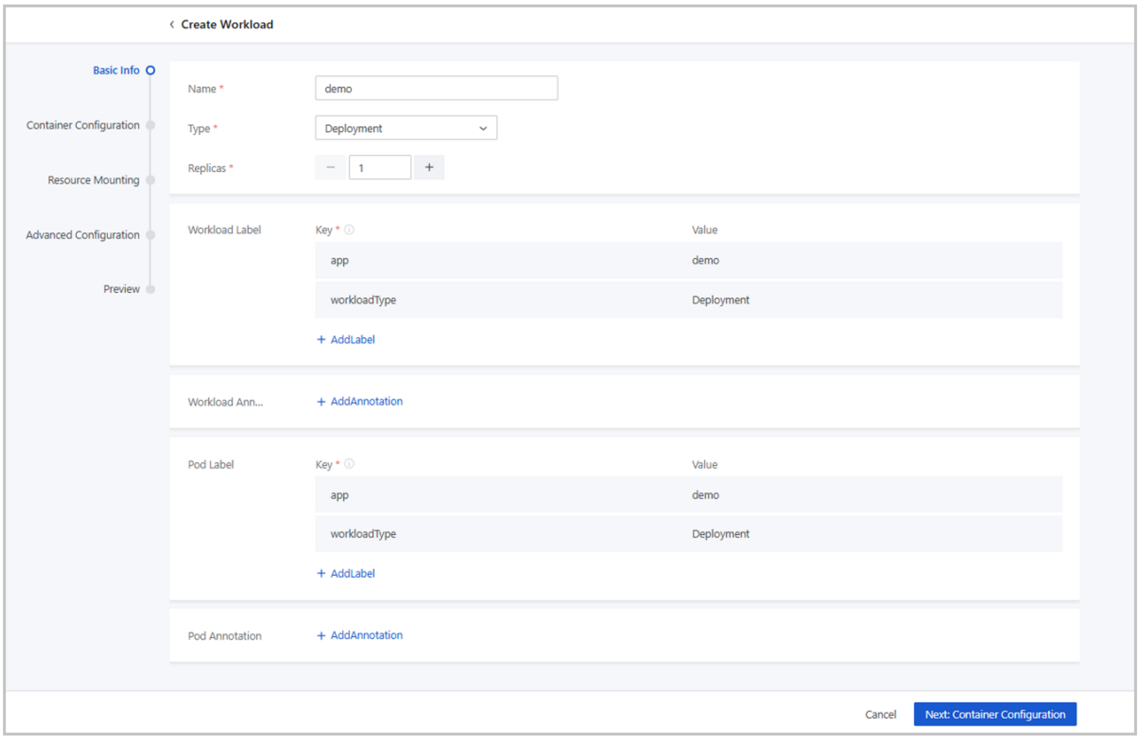
Notes.

- Typically, key names can contain 1-63 characters, including letters, numbers, and the following special characters: (-)(_)(.) Special characters cannot be used at the beginning or end of a key name.
- Supports prefixing key names and concatenating them with /, e.g. k8s.io/app, all characters filled in before / will be considered as key prefixes and all characters after will be considered as key names.
- The key name prefix can contain 1-253 characters, including lowercase letters, numbers and the following special characters: (-)(_)(.) (-)(_)(.), 2 or more special characters cannot be used consecutively, and special characters cannot be used at the beginning or end of the prefix.
- Please note that you do not have to use a key name prefix, but you must fill in the key name.

# **value:** fill in the value of the annotation

As shown in Figure 9: Basic Information

Figure 9: Basic Information



Name * demo

Type * Deployment

Replicas * 1

Workload Label

Key *	Value
app	demo
workloadType	Deployment

Workload Ann...

Pod Label

Key *	Value
app	demo
workloadType	Deployment

Pod Annotation

2. container configuration

Click **Add Container** and refer to the following example to enter the appropriate content. multiple container configurations can be added to a workload load:

- **Container Name**: Set the container name. Naming rules: length limit 1-63 characters, can only contain lowercase letters, numbers and separator (-), must start with a lowercase letter
- **Container type**: select the container type, supports two types: worker container and init container. Default is worker container.
- **Image source**: select the container image repository, support for Repositories and external repositories two kinds of image sources

  If you select the **Repository**, the container will be created using the image repository that has been uploaded to the nKU Repository, set the following parameters:

  - **Image**: Select container image
  - **Tag**: Select a image Tag

  If you select **External repository**, which will pull the image repository from outside nKU to create the container, set the following parameters:

  - **Image**: enter the detailed Image address, e.g. docker.io/library/nginx:latest

  - **Image pulling Secret**: optional, if the external repository to which the image belongs is a private repository, you need to select the Secret to be used for image repository authentication

- **Image PullPolicy**: Select an image pull policy to define whether the container image repository should be re-fetched every time the container is started.

  Like, the following three strategies are supported:

  # **Always**: the container refetches the specified Images every time it starts up

  # **IfNotPresent**: the container will refetch the image only if the specified container image does not exist on the node where the container is located.

  # Never: no matter whether the specified container image exists on the node where the container is located, it will not be reacquired; when there is no specified image available on the node where the container is located, the container will fail to startup

- **Privileged**: select whether to enable privileged mode for the container. When turned on, the node where the container resides will run the container process as the root user
- **Resource request**: optional, set container CPU/memory request value. unit of CPU: core, unit of memory: Mi, Gi

  Notes.

  - The CPU/memory request value is the minimum CPU/memory value required by the container, and when container scheduling is performed, the container will only be scheduled to the node if the total amount of CPU/memory allocable to the node is ≥ the container CPU/memory request value.
  - The CPU/Memory request value cannot be greater than the CPU/Memory limit.
  - The CPU/Memory request value cannot be greater than the CPU/Memory limit of the current namespace, or the workload will fail to be created.
  - If not set, the CPU request value defaults to be equal to the CPU/memory limit value.
  - Supports setting the CPU request value and memory request value at the same time, or setting the CPU request value or memory request value separately.

- **Resource limit**: optional, set container CPU/memory limit value. CPU unit: core, memory unit: Mi, Gi

Notes.

- The CPU/memory limit value is the maximum CPU/memory limit that the container can use. It is recommended to set the CPU/memory limit value for production environments to avoid system failures caused by container resource overruns.
- Containers that have used more CPU/memory than the limit may trigger a pod restart, affecting the normal operation of the workload, please set the limit according to the actual situation.
- If not set, the container will use the default resource limits for the current namespace.
- Supports setting CPU limit value and memory limit value at the same time, or setting CPU limit value or memory limit value separately.

- **Command**: optional, set the command to be run when the container starts, e.g., /bin/sh. If left blank, the container will run the default command of the image when it starts.
- **Argument**: optional, setup the arguments of the command, support to add more than one argument, and use half comma (,) to separate them.
- **Env**: optional, add container environment variables. Supports multiple ways to add variables:

# **Custom key-value pairs**: add container environment variables via custom key-value pairs

■ **Variable name**: Set the variable name

■ **Variable value**: Setting the value of a variable

# **Reference pod information**: reference pod Yaml field as container environment variable

■ **Variable name**: Set the variable name

■ **Pod pod field**: select the pod field to reference

# **Reference resource request/limit information**: references the container's resource request/limit field as an environment variable

■ **Variable name**: Set the variable name

■ **Resource field**: select the resource request/limit field that needs to be referenced

■ **Container name**: set the name of the container to be referenced, if left blank, the default resource request/limit field of the current container will be referenced.

# **import configmap key-value pairs**: references the specified configmap key-value pairs as environment variables

■ **Variable name**: Set the variable name

■ **ConfigMap**: select the configmap to be referenced

■ **ConfigMap key name**: Select the ConfigMap key name to be referenced

# **import configmap**: references the complete specified configmap as an env

■ **ConfigMap**: select the configmap to be referenced

■ **Prefix**: optional, fill in the prefix, the prefix will form the name of the env along with the configmap key name

# **import secret key-value pairs**: refer to the specified Secret key/value as an env

■ **Variable name**: Set the variable name

■ **Configmap**: select the Secret to be referenced

■ **Configmap key name**: select the Secret key name to be referenced

# **Import Secret**: references the full specified Secret as an env

■ **Secret**: select the Secret to be referenced

■ **Prefix**: optional, fill in the prefix, the prefix will form the name of the env together with the Secret key name

- **LivenessProbe**: Choose whether to turn on the livenessProbe to determine whether the container is alive or not, and the container will be restarted automatically when it detects that it is not alive. To enable livenessProbe, you need to set the following parameters:

# **Probe type**: Select probe type, support four probe types: httpGet, tcpSocket, exec, GPRC:

■ **httpGet**: This method is mainly used to probe the container providing HTTP/HTTPS service, the cluster will periodically launch HTTP/HTTPS GET request to the container, if the received return key is within the range of 200~399, then the probe is successful, or else the probe fails. To select the HTTPGet request method, you need to set the following parameters:

■ **Protocol**: select the detection protocol, support HTTP, HTTPS two types of protocols

- **Path:** optional, enter the probe path
- **Port:** Input probe port
- **Httpheaders:** optional, enter probe request headers

Example: select the httpGet protocol, the probe path is filled in /healthcheck, the probe port is filled in 8080, and the container IP is 192.168.0.32, then the cluster will initiate periodic GET requests to http://192.168.0.32:8080/healthcheck.

- **tcpSocket**: This probe type is mainly used to probe the container providing TCP service, the cluster will periodically establish TCP connection to the container, if the connection succeeds, it is a successful probe, otherwise it is a probe failure. To select the TcpSocket method, you need to set the following parameters:
  - **Port**: Enter the port used to establish the TcpSocket
- **Exec**: This probe type requires executing an executable command for the container, and the cluster will periodically execute the command in the container, if the command returns 0, it is a successful probe, otherwise, it is a failed probe. If the command returns 0, the detection is successful, otherwise the detection fails:
  - **Exec**: Enter a customized detection command
- **GRPC**: This probe type is mainly used for probing containers that implement GRPC health check protocol, the cluster will periodically execute remote procedure calls via GRPC, the response status is **SERVING** means the probe is successful, otherwise it is a probe failure. To select the GRPC method, you need to set the following parameters:
  - **Port**: Enter the port to be used for GRPC probing

# **InitialDelaySeconds**: set the initial detection delay time, the initial detection will be carried out after the container is started and the delay time has expired, the default is 3 seconds.

# **Period Seconds**: Set the periodSeconds, default is every 10 seconds.

# **Timeout**: set the detection timeout waiting time, the default is 1 second

# **failureThreshold**: Set the number of detection failures allowed, default is 3. When the set upper limit is reached, the application container will be recognized as not surviving and the container will be automatically restarted

- **ReadinessProbe**: Choose whether to turn on readinessProbe to determine whether the container is ready and can provide services to the outside world, and when it detects that it is not ready, the traffic will not be forwarded to the container. To enable readinessProbe, you need to set the following parameters:

# **Probe type**: Select probe type, support four probe types: httpGet, tcpSocket, exec, GPRC:

- ■ **httpGet**: This method is mainly used to probe the container providing HTTP/HTTPS service, the cluster will periodically launch HTTP/HTTPS GET request to the container, if the received return key is within the range of 200~399, then the probe is successful, or else the probe fails. To select the HTTPGet request method, you need to set the following parameters:
  - ■ **Protocol**: select the detection protocol, support HTTP, HTTPS two types of protocols
  - ■ **Path**: optional, enter the probe path
  - ■ **Port**: Input probe port
  - ■ **Httpheaders**: optional, enter probe request headers

  Example: select the httpGet protocol, the probe path is filled in /healthcheck, the probe port is filled in 8080, and the container IP is 192.168.0.32, then the cluster will initiate periodic GET requests to http://192.168.0.32:8080/healthcheck.

- ■ **tcpSocket**: This probe type is mainly used to probe the container providing TCP service, the cluster will periodically establish TCP connection to the container, if the connection succeeds, it is a successful probe, otherwise it is a probe failure. To select the TcpSocket method, you need to set the following parameters:
  - ■ **Port**: Enter the port used to establish the TcpSocket
- ■ **Exec**: This probe type requires executing an executable command for the container, and the cluster will periodically execute the command in the container, if the command returns 0, it is a successful probe, otherwise, it is a failed probe. If the command returns 0, the detection is successful, otherwise the detection fails:
  - ■ **Exec**: Enter a customized detection command
- ■ **GRPC**: This probe type is mainly used for probing containers that implement GRPC health check protocol, the cluster will periodically execute remote procedure calls via GRPC, the response status is **SERVING** means the probe is successful, otherwise it is a probe failure. To select the GRPC method, you need to set the following parameters:
  - ■ **Port**: Enter the port to be used for GRPC probing

# **InitialDelaySeconds**: set the initial detection delay time, the initial detection will be carried out after the container is started and the delay time has expired, the default is 3 seconds.

# **Period Seconds**: Set the periodSeconds, default is every 10 seconds.

# **Timeout**: set the detection timeout waiting time, default is 1 second

# **FailureThreshold**: Set the number of detection failures allowed, default is 3. When the set upper limit is reached, the container will be recognized as not ready and traffic will not be forwarded to this container

# **successThreshold:** Set the minimum number of successful detections a container needs to complete before it can be recognized as ready, the default is 1, if set to 2, the container needs to be detected successfully 2 times in a row before it can be recognized as ready.

- **Startup Probe**: Choose whether to turn on startupProbe to determine whether the container application has been initialized successfully; the container will restart automatically when it detects that the application has not been initialized successfully. If the container takes a long time to init container, it is recommended to turn on the startupProbe. To enable startupProbe, you need to set the following arguments:

# **Probe type**: Select probe type, support four probe types: httpGet, tcpSocket, exec, GPRC:

■ **httpGet**: This method is mainly used to probe the container providing HTTP/HTTPS service, the cluster will periodically launch HTTP/HTTPS GET request to the container, if the received return key is within the range of 200~399, then the probe is successful, or else the probe fails. To select the HTTPGet request method, you need to set the following parameters:

■ **Protocol**: select the detection protocol, support HTTP, HTTPS two types of protocols

■ **Path**: optional, enter the probe path

■ **Port**: Input probe port

■**httpHeaders**: optional, enter probe request httpHeaders

Example: select the httpGet protocol, the probe path is filled in /healthcheck, the probe port is filled in 8080, and the container IP is 192.168.0.32, then the cluster will initiate periodic GET requests to http://192.168.0.32:8080/healthcheck.

■ **tcpSocket**: This probe type is mainly used to probe the container providing TCP service, the cluster will periodically establish TCP connection to the container, if the connection succeeds, it is a successful probe, otherwise it is a probe failure. To select the TcpSocket method, you need to set the following parameters:

■ **Port**: Enter the port used to establish the TcpSocket

■ **Exec**: This probe type requires executing an executable command for the container, and the cluster will periodically execute the command in the container, if the command returns 0, it is a successful probe, otherwise, it is a failed probe. If the command returns 0, the detection is successful, otherwise the detection fails:

■ **Exec**: Enter a customized detection command

■ **GRPC**: This probe type is mainly used for probing containers that implement GRPC health check protocol, the cluster will periodically execute remote procedure calls via GRPC, the response status is **SERVING** means the probe is successful, otherwise it is a probe failure. To select the GRPC method, you need to set the following parameters:

■ **Port**: Enter the port used for GRPC probing

# **InitialDelaySeconds**: Set the initialDelaySeconds for probing, the initial probing will take place after the container is started and the delay time has expired, the default is 3 seconds

# **Period Seconds:** Set the periodSeconds, default is every 10 seconds.

# **Timeout**: set the detection timeout waiting time, default is 1 second

# **failureThreshold**: Set the number of detection failures allowed, default is 3. When the set number of times limit is reached, the container application will be recognized as not successfully initialized and the container will be automatically restarted

- **PostStart Callback**: Select whether to enable PostStart Callback for executing the specified initialization operation immediately after the container creation is completed. To enable PostStart callback, set the following arguments:

# **Callback method**: select the lifecycle hook, supports both HTTPGet and exec:

■**httpGet**: performs an HTTP request to a specific endpoint on the container. To select the HTTP request method, set the following parameters:

■ **Protocol**: select the protocol type, the current Kubernetes version only supports the HTTP protocol

■ **Request Address**: Set the HTttpGet request address. If left blank, the current container pod IP address will be used by default.

- ■ **path**: optional, set the path of httpGet request, default is /.
- ■ **Port**: Set the port for httpGet requests
- ■ **Httpheaders**: optional, set httpHeaders for HTTP requests

■ **Exec**: executes custom commands in container cgroups and namespaces. To select the custom command method, set the following parameters:

■ **Exec:** Set up custom executives. Supports setting multiple commands, each command is separated by a semicolon comma (,)

Note: The resources consumed by the exec will be counted towards the total container resource consumption.

- **PreStop Callback**: choose whether to enable PreStop Callback, which is used to perform specified cleanup operations before the container stops, such as saving data, closing connection, notifying other systems, etc. To enable PreStop callback, you need to set the following parameters:

# **Callback method**: select the lifecycle hook, supports both HTTPGet and exec:

■ **httpGet**: performs an HTTP request to a specific endpoint on the container. To select the HTTP request method, set the following parameters:

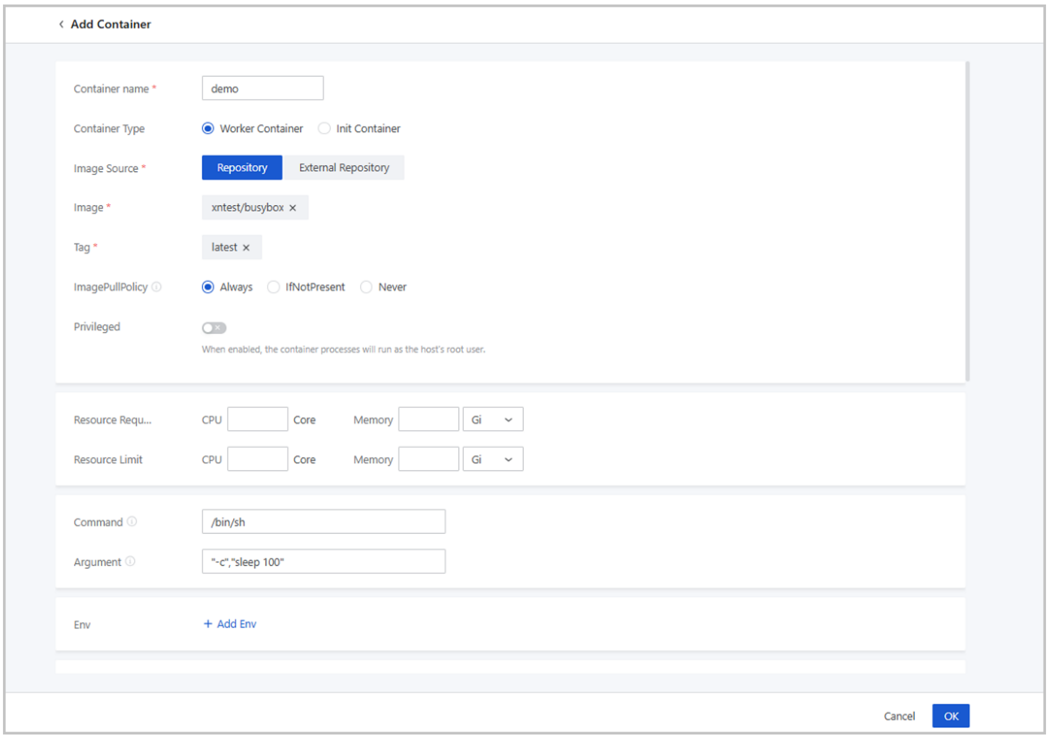
- ■ **Protocol:** select the protocol type, the current Kubernetes version only supports the HTTP protocol
- ■ **Request Address:** Set the HTttpGet request address. If left blank, the current container pod IP address will be used by default.
- ■ **path:** optional, set the path of httpGet request, default is /.
- ■ **Port:** Set the port for httpGet requests
- ■ **Httpheaders:** optional, set the HTttpHeaders for the request

■ **Exec:** executes custom commands in container cgroups and namespaces. To select the custom command method, set the following parameters:

■ **Exec:** Set up custom executives. Supports setting multiple commands, each command is separated by a semicolon comma (,)

Note: The resources consumed by the exec will be counted towards the total container resource consumption.

As shown in Figure 10: Container Configuration:



‹ Add Container

Container name * demo

Container Type ● Worker Container ○ Init Container

Image Source * [✓ Repository] External Repository

Image * xntest/busybox ×

Tag * latest ×

ImagePullPolicy ⓘ ● Always ○ IfNotPresent ○ Never

Privileged ○

When enabled, the container processes will run as the host's root user.

Resource Requ... CPU Core Memory Gi

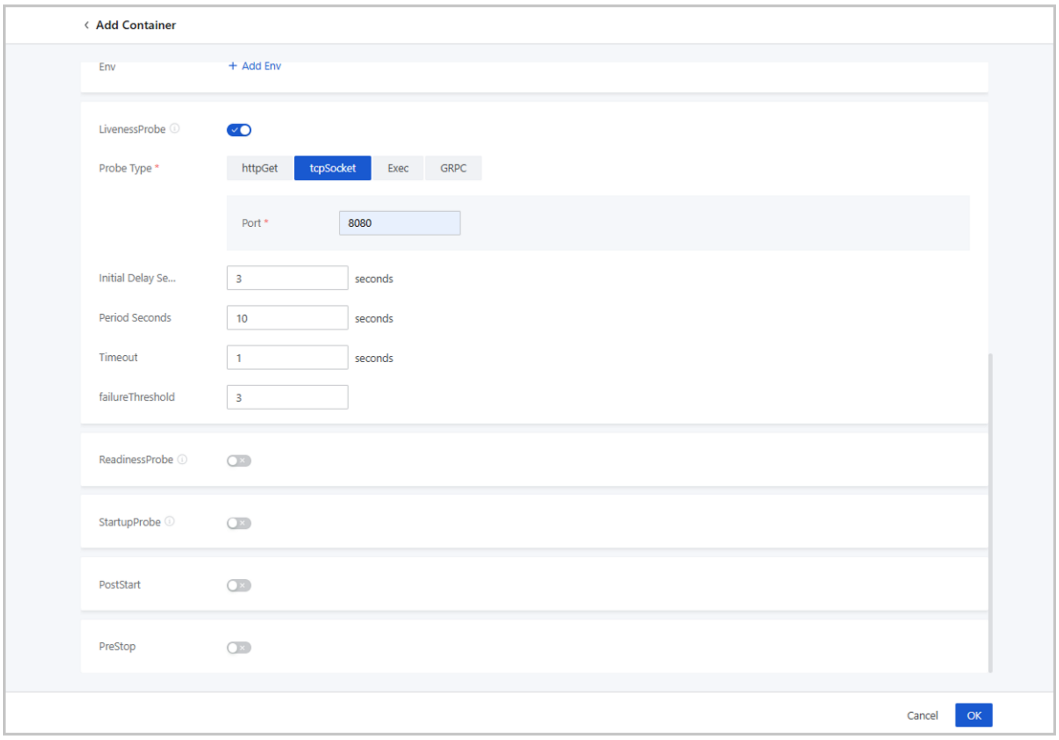
Resource Limit CPU Core Memory Gi

Command ⓘ /bin/sh

Argument ⓘ "-c","sleep 100"

Env + Add Env

Figure 10: Container Configuration



‹ Add Container

Env + Add Env

LivenessProbe ⓘ ●

Probe Type * httpGet [✓ tcpSocket] Exec GRPC

Port * 8080

Initial Delay Se... 3 seconds

Period Seconds 10 seconds

Timeout 1 seconds

failureThreshold 3

ReadinessProbe ⓘ ○

StartupProbe ⓘ ○

PostStart ○

PreStop ○

3. Resource mounting

a. Mount PVC. Two ways to mount PVC are supported: creating a new pvc and mounting an existing pvc.

- **Create a new pvc**: Create one pvc for each pod under the workload and mount persistentVolumeClaims separately via the pvc template. This method is only available for StatefulSet type workloads. Please click **Add PVC Template** and set the following parameters:

  # **PVC Name**: Set the name of the pvc. Naming rules: length limit 1-50 characters, can only contain lowercase letters, numbers and separator (-), and must start with a lowercase letter and end with a number or lowercase letter.

  # **Storageclass:** Select the storage class used to create the PVC

  Notes.

  - A storage class is a configuration template used in Kubernetes to dynamically create a pvc, defining the properties of the pvc, the creation policy, and the required storage plugins.
  - Storage classes are created and managed by the administrator. If there are no available storage classes, please contact the administrator.

  # **Access Mode**: selects the access mode for the pvc. The following three access modes are supported:

  ■ **ReadWriteOnce**: single node read/write support only

  ■ **ReadWriteMany**: support for multiple nodes read and write

  ■ **ReadOnlyMany**: Supports multiple nodes read-only

  # **Capacity**: Set the storage size of the pvc, unit: Gi, Ti

  # **Container**: select the container where the pvc will be mounted

  # **Mount path**: Select the mount path of the pvc on the container.

  # **Permission**: set pvc Permission, including read-only, read and write privileges.

  # **SubPath**: Optional, fill in the subpath of the pvc

  mounted into container.

  Notes.

  - A subPath refers to a directory or file in the pvc, and when it is filled in, only the content under that path will be mounted. You may need to fill in subPaths when mounting the same pvc to multiple containers or mounting persistentVolumeClaims into a container under multiple paths.

    - If left blank, the contents of the root directory of the mount persistentVolumeClaims will be mounted by default.

- **Mount PVC**: select an existing pvc to mount into container, all Pod under the workload will share the pvc at the same time. Please click **Add PVC** and set the following parameters:

  # **PVC**: Selects the PVC

  Note: A PVC of the exclusive disk type supports only one Pod read/write at a time, so it can only be mounted into one workload at most, and the number of pvc's for that workload must be one.

  # **Container**: select the container where the pvc will be mounted

  # **Mount path**: Set the mount path of the PVC on the container. Within the same container, the mount paths of all mounted resources cannot be the same.

  # **Permission**: set pvc Permission, including **read-only**, **read and write** permissions.

  # **subPath**: optional, fill in the subpath of the pvc mounted into container

  Notes.

    - A subPath refers to a directory or file in the pvc, and when it is filled in, only the content under that path will be mounted. You may need to fill in subPaths when mounting the same pvc to multiple containers or mounting persistentVolumeClaims into a container under multiple paths.
    - If left blank, the contents of the root directory of the mount persistentVolumeClaims will be mounted by default.

b. **Mount hostpath**: mounts the file directory of the host node to the workload. All Pod under the workload can use this hostpath. Please click **Add hostpath** and set the following parameters:

  - **Host path**: enter the host directory to be mounted

    Notes.

      - The hostpath must begin with **/**.

  - The following system directories cannot be mounted: /, /bin, /boot, /data, /dev, /etc, /home, /lib, / lib64, /opt, /proc, /root, /run, /sbin, /srv, /sys, /tmp, /usr, /var

- **Container**: Select the container to be mounted by the hostpath
- **Mount path**: Set the mount path of the host directory on the container. Within the same container, the mount paths of all mounted resources cannot be the same
- **Permissions**: Set hostpath permissions, including **read-only**, **read and write** permissions.
- **SubPath**: optional, fill in the subPath of the hostpath that is mounted into container

  Notes.

  - A subPath refers to a directory or file under the current hostpath; when filled in, only that directory or file will be mounted. You may need to fill in the subPath when mounting the same hostpath to multiple containers or mounting into a container under multiple paths.
  - If left blank, the entire contents of this hostpath will be mounted by default.

c. **Mount ConfigMap**: mounts an existing configMap to the workload. The configmap set is available to all pods under the workload. Please click **Add ConfigMap** and set the following parameters:

- **ConfigMap**: selects the ConfigMap
- **Mount content**: select the content to be mounted, you can choose to mount all keys under the mount configMap, or only mount one or more specific keys

If you choose to mount all, you need to set the following mount configMap parameters:

- **Containers**: Select the containers to be mounted by the configuration map.
- **Mount path**: Set the mount path of the configuration configuration on the container, the mount path of all resources mounting in the same container cannot be the same.
- **SubPath**: optional, fill in the subpath within the volume of the configmap to be mounted into container configuration

Notes.

- If you fill in the subPath, the configmap will generate the corresponding file under the penultimate level of the mount path, and the file name will be the last level of the mount path. For example, if you set the mount path to /data/config.conf, the configmap will be mounted into container /data/ directory, the file name is config.conf, and other files in the /data/ directory will not be overwritten.
- When the mount content is all keys of the configmap, the subPath needs to be filled as the key name of the configmap, a subPath can only correspond to one key name, users can click **Add Mount** to set the mount configMap for different keys.
- If you fill in the subPath, you need to restart the container pod after each update of the ConfigMap to synchronize the update.
- If left blank, the configMap will be mounted to the last level of the mount path. For example, if you set the mount path to / data/config, the configmap will be mounted in the /data/config directory, and the original files in the /data/config directory will be overwritten.

If you choose to mount a specific key, you need to set the following parameters:

- **Keys**: Select a specific key
  - **Path**: Define the path generated by the key in the mount directory, if you use subPath in the mount configMap, the path filled here should be the same as the subPath, don't use absolute path.

  Set the mount configMap parameters:
- **Containers**: Select the containers to be mounted by the configuration map.
- **Mount path**: Set the mount path of the configuration configuration on the container, the mount path of all resources mounting in the same container cannot be the same.
- **SubPath**: optional, fill in the subpath within the volume of the configmap to be mounted into container configuration

Notes.

- If you fill in the subPath, the configmap will generate the corresponding file under the penultimate level of the mount path, and the file name will be the last level of the mount path. For example, if you set the mount path to /data/config.conf, the configmap will be mounted into container /data/ directory, the file name is config.conf, and other files in the /data/ directory will not be overwritten.
- When the mount content is a specific key of the configmap, the subPath needs to be the same as the key path, and the user can click **Add Mount** to set the mount configurations for different keys.
- If you fill in the subPath, you need to restart the container pod after each update of the ConfigMap to synchronize the update.
- If left blank, the configMap will be mounted to the last level of the mount path. For example, if you set the mount path to / data/config, the configmap will be mounted in the /data/config directory, and the original files in the /data/config directory will be overwritten.

d. **Mount Secret**: mounts an existing Secret to the workload. The Secret is available to all pod under the workload. Please click **Add Secret** and set the following parameters:

- **Secret**: Select a Secret
- **Mount content**: select the content to be mounted, you can choose to mount all the keys under the Secret or only one or more specific keys

If you choose to mount **all**, you need to set the following mount config parameters:

- **Container**: select the container that the secret will be mounted on
- **Mount path**: set the mount path of the secret on the container, the mount path of all resources in the same container cannot be the same
- **SubPath**: optional, fill in the subpath within the secret volume to be mounted into container

Notes.

- If you fill in the subPath, this secret will generate the corresponding file under the penultimate level of the mount path, and the file name will be the last level of the mount path. For example, if you set the mount path to /data/config.conf, then the Secret will be mounted into container /data/

        directory with the file name config.conf, and other files in the /data/ directory will not be overwritten.

        - When the mount content is all keys of the secret, the subPath needs to be filled as the key name in the secret, a subPath can only correspond to one key name, users can click **Add Mount** to set the mount config for different keys.
        - If you fill in the subPath, after each update of the secret, you need to restart the container pod to synchronize the update.

    - If left blank, the secret will be mounted to the last level of the mount path. For example, if you set the mount path to /data/config, the secret will be mounted in the /data/config directory, and the original files in the /data/config directory will be overwritten.

If you choose to mount **a specific key**, you need to set the following parameters:

- **Keys**: Select a specific key
- **Path**: Define the path generated by the key in the mount directory, if you use subPath in the mount config, the path filled here should be the same as the subPath, don't use absolute path.

  Set the mount config parameters:
- **Container**: select the container that the secret will be mounted on
- **Mount path**: Set the mount path of the secret on the container, the mount path of all resources in the same container cannot be the same.
- **SubPath**: optional, fill in the subpath within the secret volume to be mounted into container

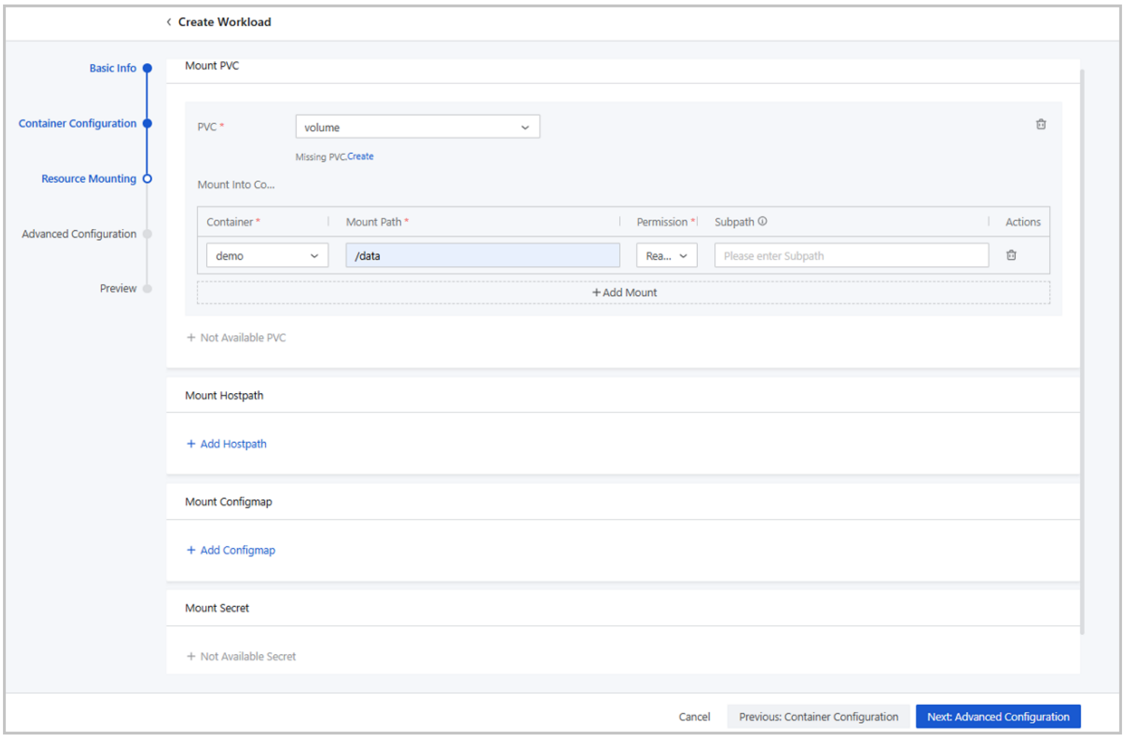
  Notes.

    - If you fill in the subPath, the secret will generate the corresponding file under the penultimate level of the mount path, and the file name will be the last level of the mount path. For example, if you set the mount path to /data/config.conf, the secret will be mounted into container /data/ directory, the file name is config.conf, and other files in the /data/ directory will not be overwritten.
    - When mount content is a secret specific key, the subPath needs to be the same as the key path.

- If you fill in the subPath, after each update of the secret, you need to restart the container pod to synchronize the update.
- If left blank, the secret will be mounted to the last level of the mount path. For example, if you set the mount path to /data/config, the secret will be mounted in the /data/config directory, and the original files in the /data/config directory will be overwritten.

As shown in Figure 11: Resource Mounting:

Figure 11: Resource Mounting



Mount PVC

PVC * volume

Missing PVC,Create

Mount Into Co...

Container *	Mount Path *	Permission *	Subpath ⓘ	Actions
demo	/data	Rea...	Please enter Subpath	🗑

+ Add Mount

+ Not Available PVC

Mount Hostpath

+ Add Hostpath

Mount Configmap

+ Add Configmap

Mount Secret

+ Not Available Secret

4. The advanced configuration can be entered accordingly by referring to the following example:

- GPU Configuration: Default is off, if you want to use GPU resources, it is on, you need to set the following parameters:

# **GPU manufacturer**: Select the manufacturer of the GPU manufacturer you want to use, currently supports NVIDIA ,Huawei,HYGON, Iluvatar.

When **NVIDIA** is selected as the GPU manufacturer, set the following parameters:

■ **GPU Selection Method:** Select the GPU selection method, which supports the following three methods: do not specify GPU, specify GPU model, and specify GPU.

■ do not specify GPU: The GPU to be used by the container pod is not specified, and the platform schedules it to any GPU that meets the GPU resource allocation according to the GPU resource allocation.

■ Specify GPU model: Select one or more GPU models to which the workload pods will be scheduled to the GPU that meet the GPU resource configuration of the selected model.

■ specify GPU: Select one or more GPUs. The workload pods will be scheduled to the GPUs that meet the GPU resource configuration.

■ **GPU model/GPU**: Specify the GPU model or GPU.

■ **Container**: select the container that will use the GPU resources

Note: When there are multiple containers in the same pod, only one container in the pod is supported to use the GPU.

■ **GPU Number**: sets the number of GPUs the container needs to use

Note: Containers can only use the GPU node of the node where the pod is located and cannot be used across nodes.

■ **GPU memory**: Sets the size of the memory of a single GPU to be used by the container. Selectable units: Gi, Mi

Notes.

- If the number of GPUs is set greater than 1, each GPU requests GPUs per GPU memory size.
- Per GPU memory cannot exceed the maximum video memory of a single GPU.

■**GPU Core Percentage**: Sets the percentage of arithmetic on a single GPU that the container needs to use

Notes.

- If the Number of GPUs is set greater than 1, each GPU can use up to a percentage of arithmetic per GPU core.

- If the per-GPU core percentage is not set, it defaults to 0. The container pod may be scheduled to any GPU that meets the GPU memory requirements.
- The container uses the GPU on a shared basis, and if exclusive GPU core is required, set the per GPU core percentage to 100%.

When you select **Huawei** for the GPU manufacturer, set the following parameters:

■ **GPU model**: Select the model of the GPU model to be used, and the available models are all the huawei GPU models in the current cluster.

■ **Container**: select the container that will use the GPU resources

Note: When there are multiple containers in the same pod, only one container in the pod is supported to use the GPU.

■ **GPU Number**: sets the number of GPUs the container needs to use

Note: Containers can only use the GPU node of the node where the pod is located and cannot be used across nodes.

■ **Exclusive GPU**: Select whether to exclusive GPU. The default is off, then the container uses the GPU in a shared way, if you need to use the whole card you can turn on this configuration

Note: Huawei does not support multi-card sharing, i.e., if the number of GPUs is configured greater than 1, you can only have an exclusive GPU.

■ **GPU memory**: Select the size of the memory of a single GPU to be used by the container.

Note: Huawei only allows GPUs to be sliced according to a fixed number of specifications, and the number of AI Core cores corresponding to the memory is also fixed, the slicing specifications of different GPUs are different, the specific specifications are shown in the drop-down box of GPU memory per GPU.

When you select **HYGON** for the GPU manufacturer, set the following parameters:

■ **GPU model**: Select the model of the GPU to be used, and the available models are all the HYGON GPU models in the current cluster.

Note: If the namespace is configured with Hygon GPU memory quota, the GPU model must be specified when using an exclusive GPU card; otherwise, the Pod will fail to be created.

- **Container**: select the container that will use the GPU resources

Note: When there are multiple containers in the same pod, only one container in the pod is supported to use the GPU.

- **GPU Number**: sets the number of GPUs the container needs to use

Note: Containers can only use the GPU node of the node where the pod is located and cannot be used across nodes.

- **Exclusive GPU**: Select whether to exclusive GPU. The default is off, then the container uses the GPU in a shared way, if you need to use the whole card you can turn on this configuration

Note: HYGON does not support multi-card sharing, i.e., if the number of GPUs is configured greater than 1, you can only have an exclusive GPU.

- **GPU memory**: Sets the size of the memory of a single GPU to be used by the container. Selectable units: Gi, Mi

Note: GPU memory cannot exceed the maximum memory of a single GPU.

- **GPU Core Percentage**: Sets the percentage of arithmetic on a single GPU that the container needs to use

When you select **Iluvatar** for the GPU manufacturer, set the following parameters:

- **GPU model**: Select the model of the GPU to be used, and the available models are all the Iluvatar GPU models in the current cluster.
- **Container**: select the container that will use the GPU resources

Note: When there are multiple containers in the same pod, only one container in the pod is supported to use the GPU.

- **GPU Number**: sets the number of GPUs the container needs to use

Note: Containers can only use the GPU node of the node where the pod is located and cannot be used across nodes.

■ **Exclusive GPU**: Select whether to exclusive GPU. The default is off, then the container uses the GPU in a shared way, if you need to use the whole card you can turn on this configuration

Note: Iluvatar does not support multi-card sharing, i.e., if the number of GPUs is configured greater than 1, you can only have an exclusive GPU.

■ **GPU memory**: Sets the size of the memory of a single GPU to be used by the container. Selectable units: Gi, Mi

Note: The GPU Memory size must be a multiple of 256 MiB and can be only an integer.

■ **GPU Core Percentage**: Sets the percentage of arithmetic on a single GPU that the container needs to use

Note: To use Iluvatar GPU, the Iluvatar software stack library needs to be pre-loaded into the container.

- **Node scheduling policy**: choose whether to set the node scheduling policy, the default is not on, that is, when the node scheduling, follow the default policy of Kubernetes, such as on, can be customized to set the scheduling policy, you need to set the following parameters:

# **Policy type**: select the policy type, support select by node name and select by node label.

■ Select by node name: select a node to which the workload container pod will be scheduled

■ Select by node label: fill in one or more node labels and the workload container pods will be dispatched to the nodes with the specified labels

# **nodes/node labels**: specify nodes or node labels

- **Pod affinity**: optional, sets the affinity of each pod under the current workload, or the affinity of the current workload with other workload Pod, specifying whether they are scheduled to the same node or not

# **Add Type**: two affinity addition methods are supported: basic and advanced.

When selecting **the basic** method, set the following parameters:

- ■ **Workload**: select workload, support to select current workload or other workloads
  - ■ select **current workload**，Indicates that the pods under the current workload should be scheduled to the same node.
  - ■ select **other workloads**，Indicates that the pod with the current workload and the pod with the specified workload are scheduled to the same node.

When you select the **Advanced** method, set the following parameters:

- ■ **Affinity policy**: select an affinity policy. Both preferred and non-preferred strategies are supported:
  - ■ Required: that affinity must be satisfied
  - ■ Preferred: this affinity is satisfied as much as possible, and when it cannot be satisfied, some of the container pods may be dispatched to different nodes
- ■ **Weight**: set the weight of the affinity, this parameter is required when the affinity policy is preferred to **non-preferred**
- ■ **Namespace**: select a namespace, and when you do, you can set the affinity between the pod of the current workload and the Pod under that namespace
- ■ **Namespace selector**: Users can also select namespaces via the namespace selector, which supports selecting by label, or selecting by label matchExpressions
  - ■ **Matching label**: fill in the label key and value, the system will automatically select the namespace containing the label; if you fill in more than one label, the namespace must contain these labels at the same time to be selected
  - ■ **Matching label expressions**: fill in the label expression key, value, operator, the system will automatically select the namespace that matches the label expression; if you fill in more than one label expression at the same time, the namespace must match these label expressions at the same time to be selected

Note: If you do not manually select a namespace or use the namespace selector to match namespaces, the system will default to the namespace of the current workload.

■ **Pod selector**: used to select a pod. Pod under the current namespace will be dispatched to the same node as the pod selected here. Supports selection by label, or by matchExpressions

■ **Matching label**: fill in the label key, value, the system will automatically select the pod containing the label; such as filling in multiple labels, the pod must contain these labels at the same time to be selected

■ **Matching label expression**: fill in the label expression key, value, operator, the system will automatically select the pod that matches the label expression; if you fill in more than one label expression at the same time, the pod must match these label expressions at the same time to be selected

Note: If not manually selected, the current pod will be scheduled to the same node as any pod in the selected namespace.

- **Pod Anti-affinity**: optional, sets the anti-affinity of each pod under the current workload, or the anti-affinity of the current workload with other workload Pod, used to specify whether they are dispatched to different nodes or not
- **Add Type**: supports two affinity addition methods: basic and advanced.

When selecting **the basic** method, set the following parameters:

■ **Workload**: select workload, support to select current workload or other workloads

■ select **current workload**，Indicates that the pods under the current workload should be scheduled to the same node.

■ select **other workloads**，Indicates that the pod with the current workload and the pod with the specified workload are scheduled to the same node.

When you select the **advanced** method, set the following parameters:

■ **Affinity policy**: select an affinity policy. Both preferred and non-preferred strategies are supported:

■ Required: that anti-affinity must be satisfied

  - Preferred: this anti-affinity is satisfied as much as possible, and when it cannot be satisfied, some of the pods may be scheduled to the same nodes

- **Weight**: set the anti-affinity weight, this parameter is required when the affinity policy is preferred to **non-preferred**
- **Namespace**: select a namespace, and when you do, you can set the inverse affinity between the pod of the current workload and the Pod under that namespace
- **Namespace selector**: Users can also select namespaces via the namespace selector, which supports selecting by label, or selecting by label matchExpressions
  - **Matching label**: fill in the label key and value, the system will automatically select the namespace containing the label; if you fill in more than one label, the namespace must contain these labels at the same time to be selected
  - **Matching label expressions**: fill in the label expression key, value, operator, the system will automatically select the namespace that matches the label expression; if you fill in more than one label expression at the same time, the namespace must match these label expressions at the same time to be selected

Note: If you do not manually select a namespace or use the namespace selector to match namespaces, the system will default to the namespace of the current workload.

- **Pod selector**: used to select a pod. Pod under the current namespace will be dispatched to a different node than the one selected here. Supports selection by label, or by matchExpressions
  - **Matching label**: fill in the label key, value, the system will automatically select the pod containing the label; such as fill in more than one label, the pod must contain these labels at the same time to be selected
  - **Matching label expression**: fill in the label expression key, value, operator, the system will automatically select the pod that matches the label expression; if you fill in more than one label expression at the same time, the pod must match these label expressions at the same time to be selected

Note: If not manually selected, the current pod will not be scheduled to the same node as any pod in the selected namespace.

- Toleration: optional, sets the workload pod's tolerance policy for node taint; when set, the pod will be allowed to be scheduled to nodes that contain taint

# **key**: fill in the key of the taint that needs to be tolerated, if you leave it blank, the default matches all the keys of the taint

# **Action**s: Select the taint matching method. If **Equal** is selected, the value and effect of the taint needs to be further matched; if **Exist** is selected, the effect of the taint needs to be further matched

# **Value**: Fill in the value of the taint to be tolerated, this parameter needs to be set only if **the operation** is **Equal**

# **Effect**: Fill in the effects of the taint that need to be tolerated, please fill in the effects that match the actual configuration of the taint. Taint may be configured with the following effects:

■ NoSchedule: the system does not schedule a pod to this node unless the pod is configured with a tolerance policy that matches the taint and the effect of the taint; a pod that has been scheduled on this node is not evicted.

■ PreferNoSchedule: the system tries to avoid scheduling the pod to the node unless the container configuration has a tolerance policy matching the taint and the effect of the taint; the effect does not completely prevent the pod from being scheduled to the node.

■NoExecute： The system will not schedule pods on this node, and will evict any existing pods on this node after the configured toleration seconds have expired, unless the pod has a toleration policy configured that matches the taint and taint effect.

# **Tolerance seconds**: this parameter needs to be set when the tolerance effect is **NoExecute**, and the existing container pods on the node of the taint will be evicted at the end of the tolerance seconds. Unit: seconds

Notes.

- If not set, the default can always be tolerated and existing Pod on that node will not be evicted.
- If set to 0 or a negative number, existing container pods on that node will be evicted immediately.

- **Service Name:** the default value is the workload name, which can be modified to other values. This parameter is only supported when the workload type is StatefulSet.

Validation Rules: The input content must be 1-63 characters in length, can only contain lowercase letters, numbers, and the special character "-", and must start with a lowercase letter and end with a number or lowercase letter

Notes: The service name refers to serviceName. This field must be configured if a Headless service is to be used, and the name of the Headless service must be consistent with the value of this field

- **Network Configuration:** Optional. Set the network used by the workload. The native Kubernetes container network is used by default

  - ■ Network Type: Select the network type. Three types are supported: pod network only, pod external network, pod network and external network

    Notes:

    - Pod network refers to the native Kubernetes container network and only supports access within the Kubernetes cluster
    - Pod External network supports direct access to the pod IP from outside the Kubernetes cluster
    - When the workload's network type is "pod external network ", LoadBalancer-type services are not supported

  - ■ Use Node Network: Supported when the network type is "pod network". After being enabled, the workload will use the node network and not be restricted by network policies
  - ■ Pod External Network: Required to select the pod external network segment to be used when the network type is "pod external network " or " pod network and external network "

- **Update Policy:** Optional. Set the strategy for workload updates. Different types of workloads require different parameters to be configured

  # When the workload type is Deployment, the parameters are as follows:

  - ■ **Update Type**: Select the workload update method. Optional values: RollingUpdate, Recreate

- **MaxSurge:** Optional. Set the maximum number or percentage of pod replicas allowed to exceed the desired number during the update. Default value: 25%. Optional units: %, Count. This parameter can be configured only when the update method is RollingUpdate

- **MaxUnavailable:** Optional. Set the maximum number or percentage of unavailable pod replicas allowed during the update. Default value: 25%. Optional units: %, Count. This parameter can be configured only when the update method is RollingUpdate

  Notes: MaxSurge and MaxUnavailable cannot both be set to 0.

- **MinReadySeconds:** Set the minimum waiting time for a pod to be considered available after it is ready. Unit: seconds

- **ProgressDeadlineSeconds:** The timeout period to wait for the update to complete before marking the Deployment as failed. Default value: 600 seconds

  Notes: The ProgressDeadlineSeconds must be greater than the MinReadySeconds.

- **Historical Versions to Retain:** Set the number of historical versions of the workload to retain. Default value: 10. Historical versions of the workload can be viewed on the workload details page

# When the workload type is StatefulSet, the parameters are as follows:

- **Update Type**: Select the workload update method. Optional values: RollingUpdate, OnDelete

- **Partition :** Optional. If a partition is set, only the pods with ordinals greater than or equal to the partition will be updated when the StatefulSet is updated. This parameter can be configured only when the update method is RollingUpdate

  Notes: If the partition is set to a value greater than the number of replicas of the StatefulSet, the StatefulSet will not be updated.

- **PodManagementPolicy :** Optional. Defines the strategy for scaling pods during scaling operations. Optional values: OrderedReady, Parallel

- **Historical Versions to Retain:** Set the number of historical versions of the workload to retain. Default value: 10. Historical versions of the workload can be viewed on the workload details page

# When the workload type is DaemonSet, the parameters are as follows:

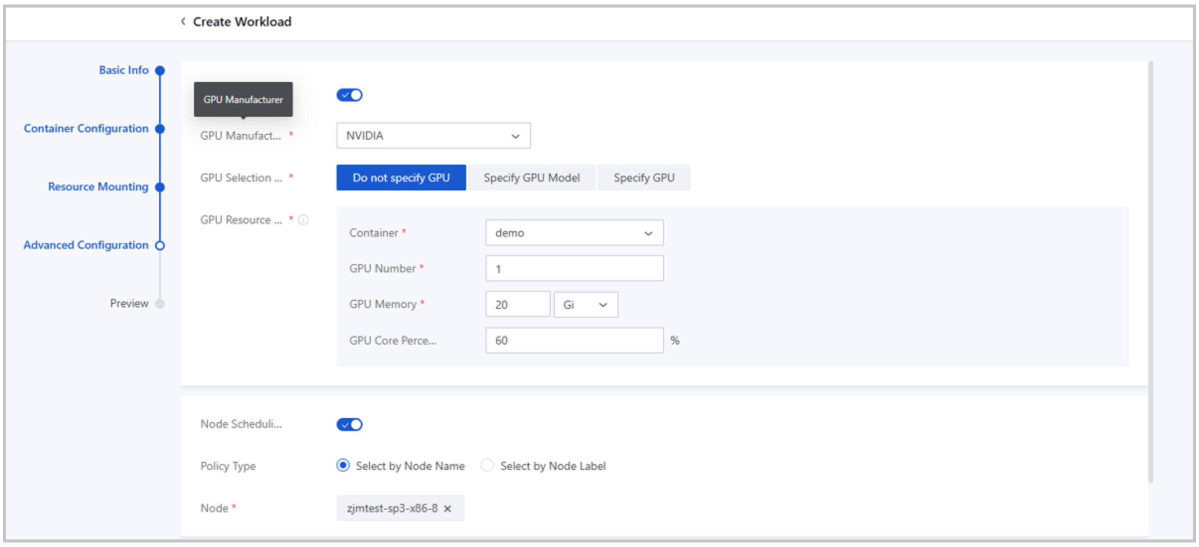
- **Update Type**: Select the workload update method. Optional values: RollingUpdate, OnDelete
- **MaxSurge:** Optional. Set the maximum number or percentage of pod replicas allowed to exceed the desired number during the update. Default value: 25%. Optional units: %, Count. This parameter must be configured when the update method is RollingUpdate
- **MaxUnavailable:** Optional. Set the maximum number or percentage of unavailable pod replicas allowed during the update. Default value: 25%. Optional units: %, Count. This parameter must be configured when the update method is RollingUpdate

  Notes: Only one of MaxSurge and MaxUnavailable must be set to 0, and the other cannot be 0.

- **MinReadySeconds:** Set the minimum waiting time for a pod to be considered available after it is ready. Unit: seconds
- **Historical Versions to Retain:** Set the number of historical versions of the workload to retain. Default value: 10. Historical versions of the workload can be viewed on the workload details page

As shown in Figure 12: Advanced Configuration:

Figure 12: Advanced Configuration



GPU Manufact... * NVIDIA

GPU Selection ... * Do not specify GPU

GPU Resource ... *

Container * demo

GPU Number * 1

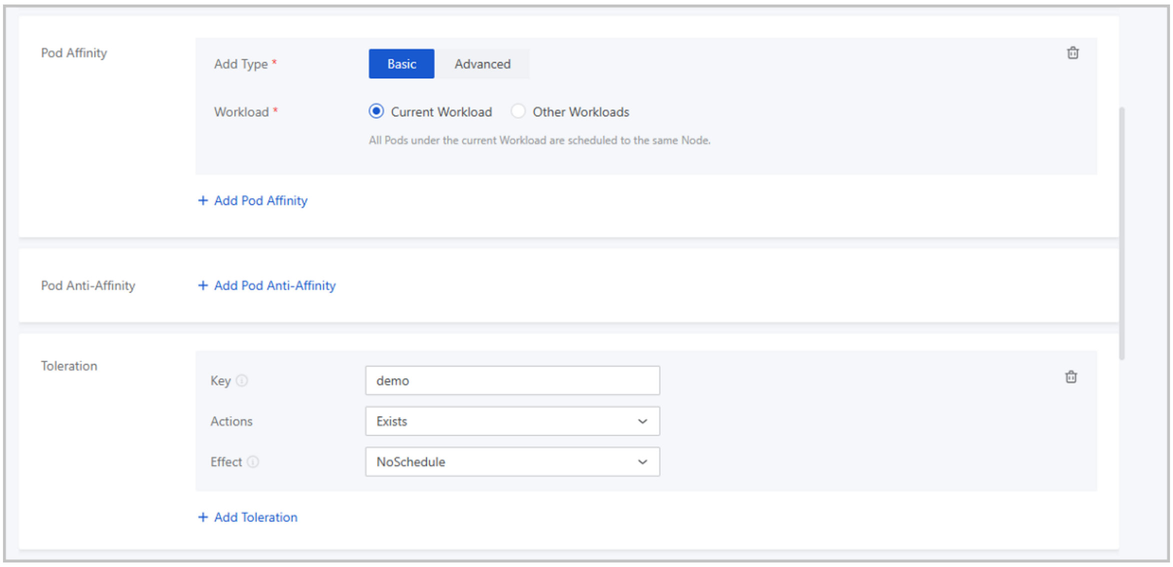
GPU Memory * 20 Gi

GPU Core Perce... 60 %

Node Scheduli...

Policy Type ● Select by Node Name ○ Select by Node Label

Node * zjmtest-sp3-x86-8 ×



Pod Affinity

Add Type * Basic

Workload * ● Current Workload ○ Other Workloads

All Pods under the current Workload are scheduled to the same Node.

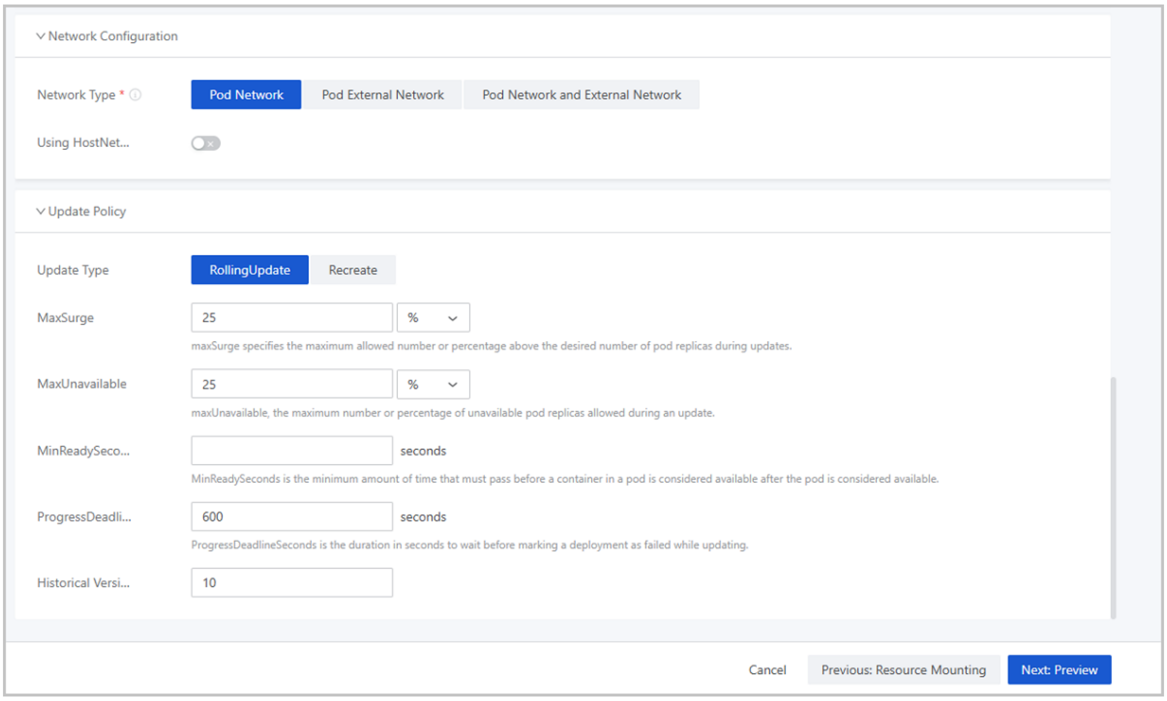
Pod Anti-Affinity

Toleration

Key demo

Actions Exists

Effect NoSchedule



Network Configuration

Network Type * Pod Network

Using HostNet...

Update Policy

Update Type RollingUpdate

MaxSurge 25 %

maxSurge specifies the maximum allowed number or percentage above the desired number of pod replicas during updates.

MaxUnavailable 25 %

maxUnavailable, the maximum number or percentage of unavailable pod replicas allowed during an update.

MinReadySeco... seconds

MinReadySeconds is the minimum amount of time that must pass before a container in a pod is considered available after the pod is considered available.

ProgressDeadli... 600 seconds

ProgressDeadlineSeconds is the duration in seconds to wait before marking a deployment as failed while updating.

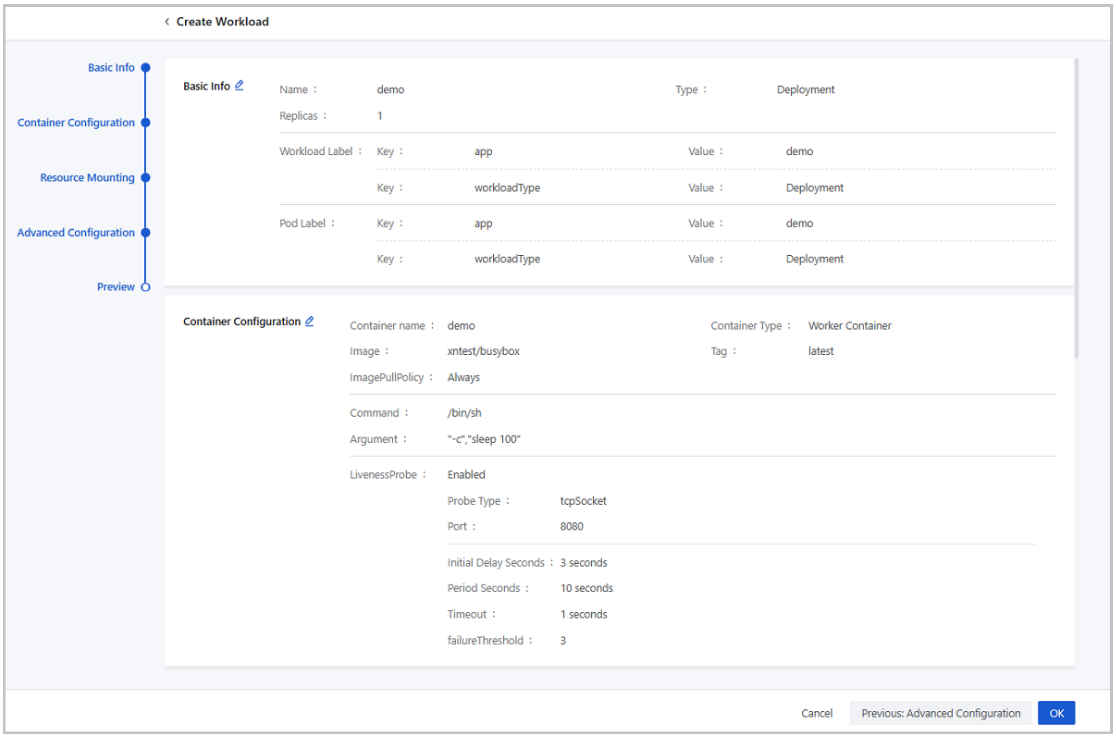
Historical Versi... 10

5. Preview

view the workload to be created and support jump to modify.

As shown in Figure 13: Preview:

Figure 13: Preview



< Create Workload

Basic Info

Name : demo
Type : Deployment
Replicas : 1

Workload Label : Key : app Value : demo
Key : workloadType Value : Deployment

Pod Label : Key : app Value : demo
Key : workloadType Value : Deployment

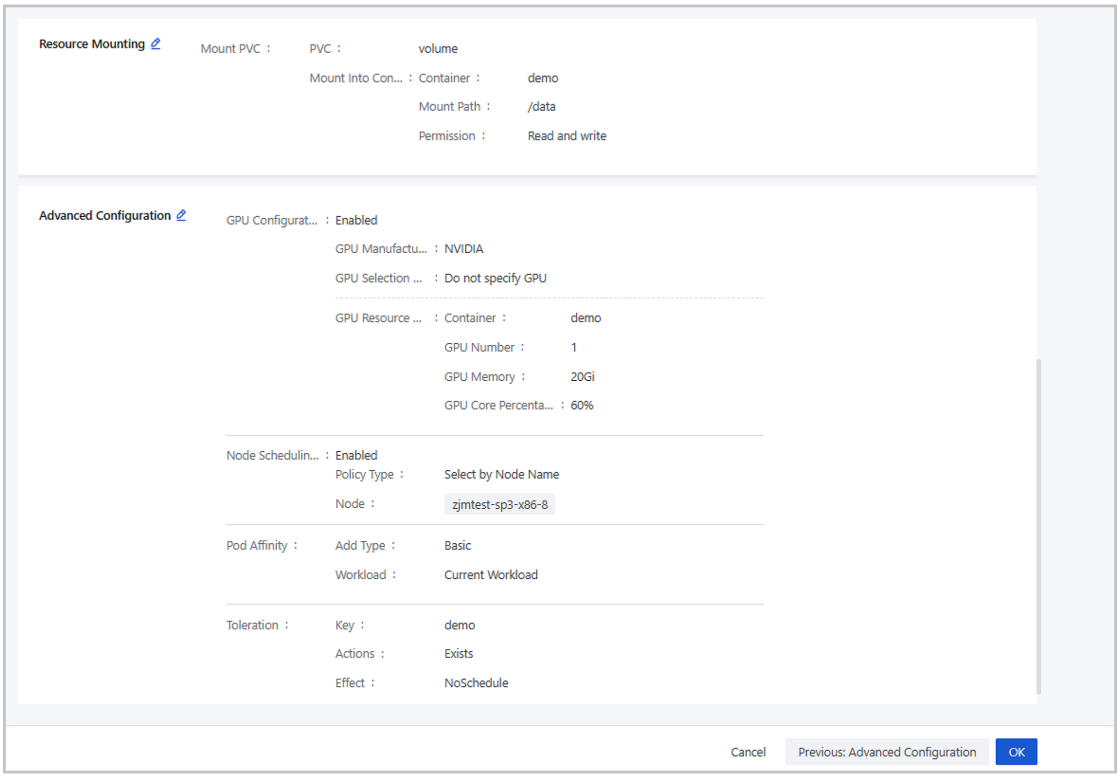
Container Configuration

Container name : demo
Container Type : Worker Container
Image : xntest/busybox
Tag : latest
ImagePullPolicy : Always

Command : /bin/sh
Argument : "-c","sleep 100"

LivenessProbe : Enabled
Probe Type : tcpSocket
Port : 8080

Initial Delay Seconds : 3 seconds
Period Seconds : 10 seconds
Timeout : 1 seconds
failureThreshold : 3



Resource Mounting

Mount PVC : PVC : volume
Mount Into Con... : Container : demo
Mount Path : /data
Permission : Read and write

Advanced Configuration

GPU Configurat... : Enabled
GPU Manufactu... : NVIDIA
GPU Selection ... : Do not specify GPU

GPU Resource ... : Container : demo
GPU Number : 1
GPU Memory : 20Gi
GPU Core Percenta... : 60%

Node Schedulin... : Enabled
Policy Type : Select by Node Name
Node : zjmtest-sp3-x86-8

Pod Affinity : Add Type : Basic
Workload : Current Workload

Toleration : Key : demo
Actions : Exists
Effect : NoSchedule

7.1.2.3 Managing workloads

In the nKU main menu, click **Container Orchestration** > **Application Management** > **Workloads** to access the **workload** interface. Workloads support the following actions:

manipulate	descriptive
Create Workload	Create a new workload.
Edit Workload	Update the workload configuration via a form.
Update Image	Update the workload container image.
Update Yaml	Update the workload configuration via Yaml form.
Restart Workload	Restart the workload. **Notes.** • After restarting, the system creates new Pod for the workloads according to the update policy and removes the current Pod. This process may cause business interruption, so please proceed with caution. • If the workload is of Deployment type and mount persistentVolumeClaims data disk type, please realize the reboot by manually deleting the pod under it.
Autoscaling	Adjusts the number of workload replicas and supports both manual scaling and autoscaling types of scaling. - Autoscaling supports two trigger indicators, CPU target utilization rate and memory target utilization rate.
Workload Details	Go to the workload details page to view basic information about the workload, worker Pod, historical versions, events, and other related services, including: services, pvcs, and so on.
Rollback Workload	Rolls back the workload to the specified version, and the system rebuilds the workload with the rollback version configuration. **Note:** To perform this action, go to the Workload Details page and click **Historical Versions** > **Action** > **Rollback**.
Delete Workload	Remove the workload from the cluster.
Export Workload information	Export Workload information as a CSV file. Supports exporting the current page or all pages.

7.1.3 Job

7.1.3.1 General

The job is responsible for batch processing of ephemeral operations and ensures that one or more containers used for batch processing of operations comprise the termination of the function, including one-time tasks and timed tasks of two types.

Type of job

- **One-time jobs**: i.e., Jobs in Kubernetes, which deal with one-time transient operations only, and the Pod responsible for running these operations are automatically exited and deleted after the operations are successfully completed.
- **Timed job**: i.e. Cronjob in Kubernetes, adds the concept of time to Job. A timed job will run the same business, such as pvc backups, sending emails, etc., over and over again according to a specified time period.

Batch versus long-term servicing

- **Batch operations**: batch operations have a clear run always and the pod stops running once the operation is successfully completed. In nKU, such operations and the Pod that run them are managed by jobs.
- **Long-term servo operations**: long-term servo operations have no explicit operational termination and run forever as long as the user does not stop them. In nKU, these types of businesses and the Pod running them are managed by Deployment or Statefulset type workloads.

7.1.3.2 Creating jobs

In the nKU main menu, click **Container Orchestration** > **Application Management** > **Tasks** to access the **Tasks** screen. Click **Create Job** to bring up the **Create Job** screen.

Creating a job is divided into the following five steps:

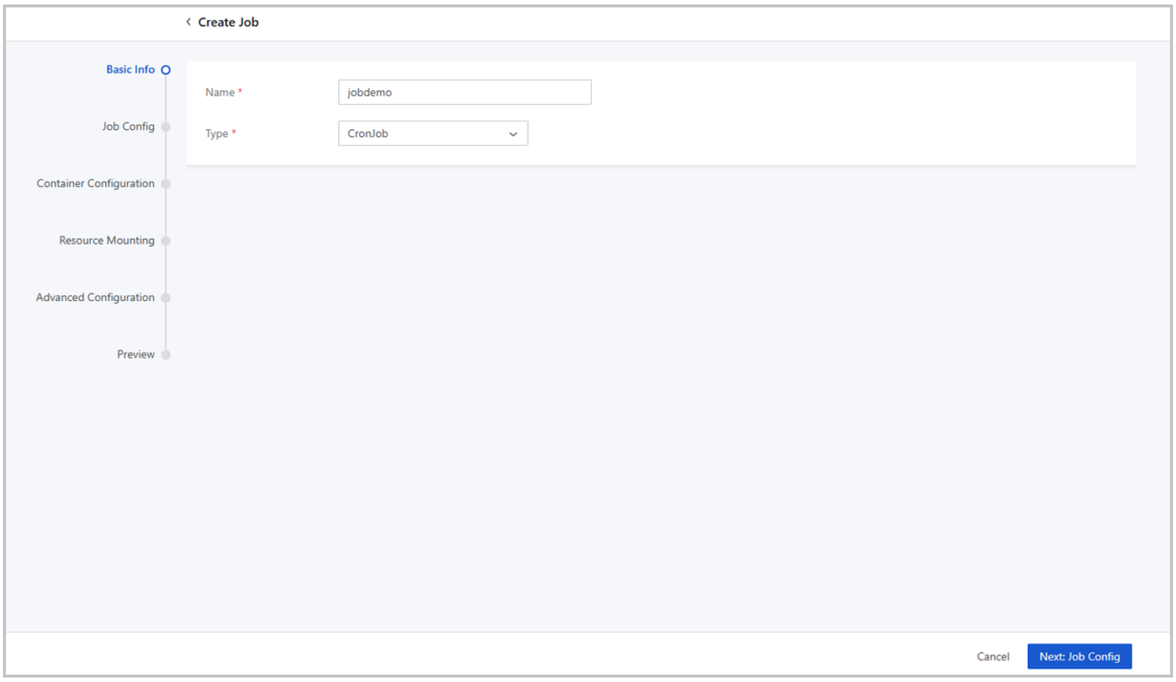
1. The basic information can be entered accordingly by referring to the following example:
    - **Name**: Set the name of the job. Naming rules: length limit 2-50 characters, input can only contain lowercase letters, numbers and separator (-), and must start with a lowercase letter and end with a number or lowercase letter.

Note: You cannot use the same name for the same type of job under the same namespace.

- **Type**: select the type of job, supports Job (one-time job) and CronJob (timed job).

As shown in Figure 14: Basic Information:

Figure 14: Basic Information



Name * jobdemo

Type * CronJob

2. The job configurations can be entered accordingly by referring to the following example:

- **concurrencyPolicy**: only CronJob type jobs need to set this parameter, it is used to define the processing strategy when the new and old jobs conflict with each other in a timed job, it supports three kinds of strategies: Forbid, Allow and Replace:

  # Forbid: no new job will be created when the previous one is not completed.

  # Allow: When a new job time point has been reached, but the previous job has not completed, allow a new job to be created and the old and new jobs to run in parallel.

  # Replace: When a new job time point has been reached, but the previous job has not been completed, a new job will be created and replace the previous one.

- **Schedule**: only CronJob type of job need to set this parameter, used to define the execution period of the timed task. Support quick selection by minutes, hours, days, weeks, months, etc., or custom rules, custom rules need to follow the Cron schedule syntax

  Note: The job is executed in UTC time by default, please take into account the time difference of the time zone you belong to.

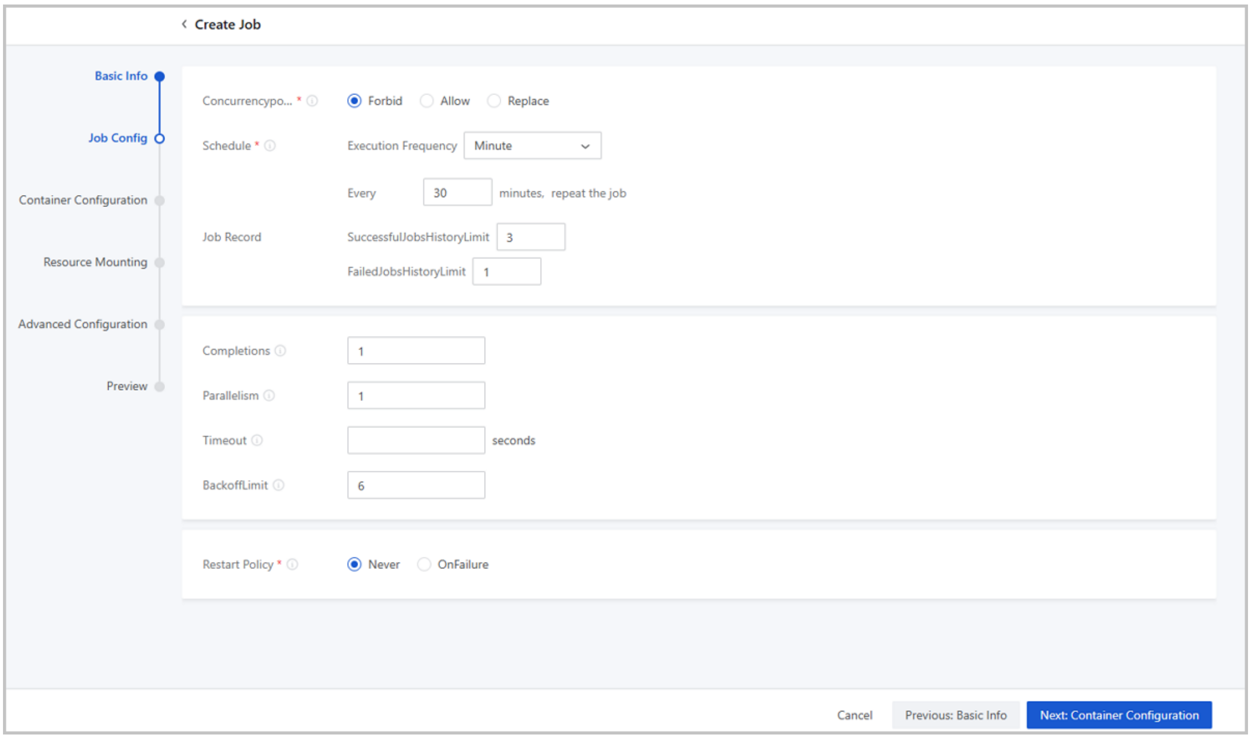
- **Job Record**: Optional, this item can be set for CronJob type jobs, it is used to define the number of success/failure execution records to be kept in the job details page. The default is 3 success records and 1 failure record.
- **Completions**: optional, defaults to 1, sets the number of container pods that need to be successfully completed before the job ends
- **Parallelism**: optional, default is 1, set the number of container pods running at the same time
- **Timeout**: optional, set the timeout time of the job, if the job is not finished after this time, the system will try to terminate all the container pods in the job, unit: second
- **BackoffLimit**: optional, default is 6, sets the upper limit of the number of times the job will try to execute, after exceeding this limit, the job will not continue to try to execute
- **Restart policy**: Set the restart policy when a container pod fails, supporting Never and Onfailure policies.

  # Never: no reboot when a pod fails, the system will create a new pod

  # Onfailure: will attempt to restart internal containers in case of pod failure

As shown in Figure 15: Job Configuration:

Figure 15: Job Configuration



Concurrencypo... * ● Forbid ○ Allow ○ Replace

Schedule * Execution Frequency: Minute

Every 30 minutes, repeat the job

Job Record SuccessfulJobsHistoryLimit: 3

FailedJobsHistoryLimit: 1

Completions: 1

Parallelism: 1

Timeout: seconds

BackoffLimit: 6

Restart Policy * ● Never ○ OnFailure

3. Container Configuration

Can be entered accordingly by referring to the following example:

- **Image Source**: Select the container image repository, support both Repository and external repository. If you select **Repository**, the container will be created using the image that has been uploaded to nKU Repository, please set the following parameters:
- **Image**: Select container image
- **Tag**: Select a Tag

If you select **external repository**, which will pull the image repository from outside nKU to create the container, set the following parameters:

  - **Image**: enter the detailed Image address, e.g. docker.io/library/nginx:latest
  - **Image pulling Secret**: If the external repository to which the image belongs is a private repository, you need to select a Secret for image repository authentication.

- **Resource request**: optional, set container CPU/memory request value. unit of CPU: core, unit of memory: Mi, Gi

Notes.

- The CPU/memory request value is the minimum CPU/memory value required by the container, and when container scheduling is performed, the container will only be scheduled to the node if the total amount of CPU/memory allocable to the node is ≥ the container CPU/memory request value.
- The CPU/Memory request value cannot be greater than the CPU/Memory limit.
- The CPU/Memory request value cannot be greater than the CPU/Memory limit of the current namespace, or the workload will fail to be created.
- If not set, the CPU request value defaults to be equal to the CPU/memory limit value.
- Supports setting the CPU request value and memory request value at the same time, or setting the CPU request value or memory request value separately.

- **Resource Limit**: optional, set container CPU/memory limit value. CPU unit: core, memory unit: Mi, Gi

Notes.

- The CPU/memory limit value is the maximum CPU/memory limit that the container can use. It is recommended to set the CPU/memory limit value for production environments to avoid system failures caused by container resource overruns.
- Containers that have used more CPU/memory than the limit may trigger a pod restart, affecting the normal operation of the workload, please set the limit according to the actual situation.
- If not set, the container will use the default resource limits for the current namespace.
- Supports setting CPU limit value and memory limit value at the same time, or setting CPU limit value or memory limit value separately.

- **Command**: optional, set the command to be run when the container starts, e.g., /bin/sh. If left blank, the container will run the default command of the image when it starts.

- **Argument**: optional, setup the arguments of the command, support to add more than one argument, and use half comma (,) to separate them.
- **Env**: optional, add container environment variables

# **key**: Fill in the key of the env, only alphanumeric and special characters (-)(_)(.) can be included.

# **value**: fill in the value of the env

- **LivenessProbe**: Choose whether to turn on the livenessProbe to determine whether the container is alive or not, and the container will be restarted automatically when it detects that it is not alive. To enable livenessProbe, you need to set the following parameters:

  # **Probe type**: Select probe type, support three probe types: httpGet, tcpSocket and exec:

  ■ **httpGet:** This method is mainly used to probe the container providing HTTP/HTTPS service, the cluster will periodically launch HTTP/HTTPS GET request to the container, if the received return key is within the range of 200~399, then the probe is successful, or else the probe fails. To select the HTTPGet request method, you need to set the following parameters:

  ■ **Protocol**: select the detection protocol, support HTTP, HTTPS two types of protocols

  ■ **Path**: optional, enter the probe path

  ■ **Port**: Input probe port

  ■**httpHeaders**: optional, enter probe request httpHeaders

  Example: select the httpGet protocol, the probe path is filled in /healthcheck, the probe port is filled in 8080, and the container IP is 192.168.0.32, then the cluster will initiate periodic GET requests to http://192.168.0.32:8080/healthcheck.

  ■ **tcpSocket**: This probe type is mainly used to probe the container providing TCP service, the cluster will periodically establish TCP connection to the container, if the connection succeeds, it is a successful probe, otherwise it is a probe failure. To select the TcpSocket method, you need to set the following parameters:

  ■ **Port**: Enter the port used to establish the TcpSocket

■ **Exec**: This probe type requires executing an executable command for the container, and the cluster will periodically execute the command in the container, if the command returns 0, it is a successful probe, otherwise, it is a failed probe. If the command returns 0, the detection is successful, otherwise the detection fails:

■ **Exec**: Enter a customized detection command

# **InitialDelaySeconds**: set the initial detection delay time, the initial detection will be carried out after the container is started and the delay time has expired, the default is 3 seconds.

# **Period Seconds**: Set the periodSeconds, default is every 10 seconds.

# **Timeout**: set the detection timeout waiting time, default is 1 second

# **failureThreshold**: Set the number of detection failures allowed, default is 3. When the set upper limit is reached, the application container will be recognized as not surviving and the container will be automatically restarted

- **ReadinessProbe**: Choose whether to turn on readinessProbe to determine whether the container is ready and can provide services to the outside world, and when it detects that it is not ready, the traffic will not be forwarded to the container. To enable readinessProbe, you need to set the following parameters:

# **Probe type**: Select probe type, support three probe types: httpGet, tcpSocket and exec:

■ **httpGet:** This method is mainly used to probe the container providing HTTP/HTTPS service, the cluster will periodically launch HTTP/HTTPS GET request to the container, if the received return key is within the range of 200~399, then the probe is successful, or else the probe fails. To select the HTTPGet request method, you need to set the following parameters:

■ **Protocol**: select the detection protocol, support HTTP, HTTPS two types of protocols

■ **Path**: optional, enter the probe path

■ **Port**: Input probe port

■ **Httpheaders**: optional, enter probe request headers

Example: select the httpGet protocol, the probe path is filled in /healthcheck, the probe port is filled in 8080, and the container IP is 192.168.0.32, then the cluster will initiate periodic GET requests to http://192.168.0.32:8080/healthcheck.

- **tcpSocket**: This probe type is mainly used to probe the container providing TCP service, the cluster will periodically establish TCP connection to the container, if the connection succeeds, it is a successful probe, otherwise it is a probe failure. To select the TcpSocket method, you need to set the following parameters:
  - **Port:** Enter the port used to establish the TcpSocket
- **Exec:** This probe type requires executing an executable command for the container, and the cluster will periodically execute the command in the container, if the command returns 0, it is a successful probe, otherwise, it is a failed probe. If the command returns 0, the detection is successful, otherwise the detection fails:
  - **Exec:** Enter a customized detection command

# **Initial Delay Seconds**: set the initial detection delay time, the initial detection will be carried out after the container is started and the delay time has expired, the default is 3 seconds.

# **Period Seconds**: Set the periodSeconds, default is every 10 seconds.

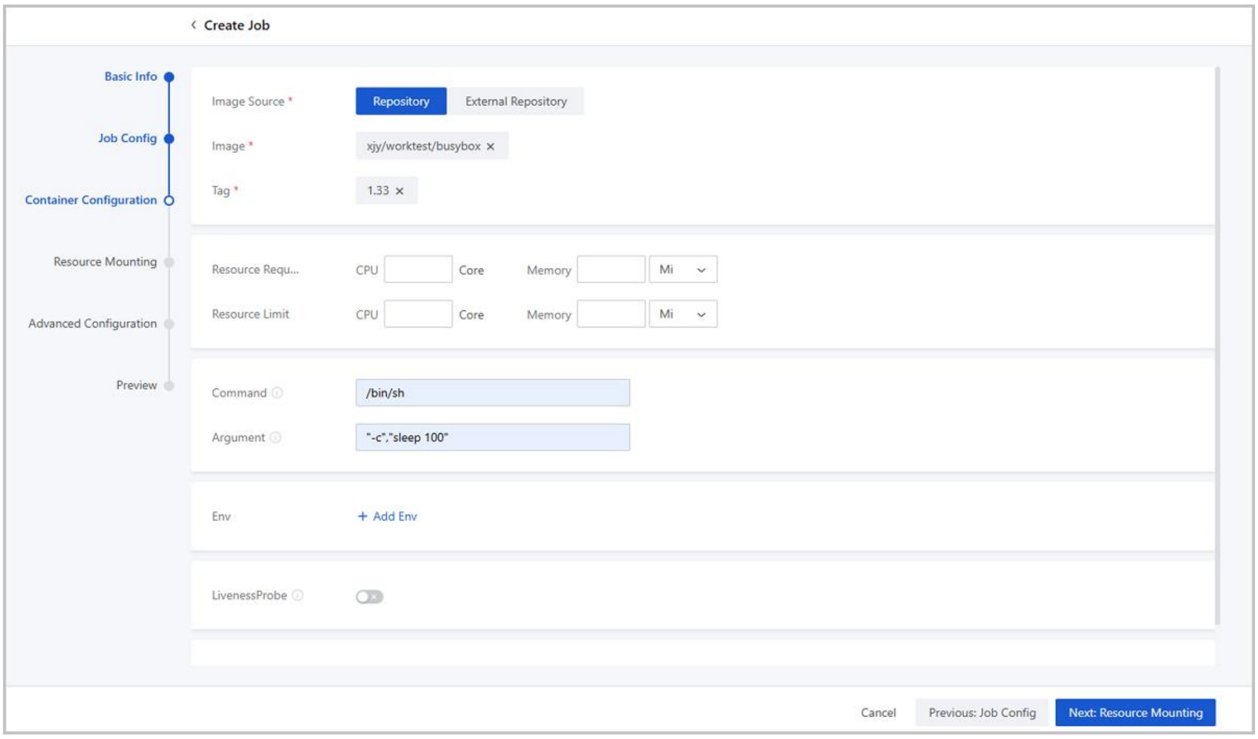
# **Timeout**: set the detection timeout waiting time, default is 1 second

# **failureThreshold**: Set the number of detection failures allowed, when the set number of times reaches the upper limit, the container will be recognized as not ready and the traffic will not be forwarded to this container. The default is 3 times.

# **successThreshold**: Set the minimum number of successful detections a container needs to complete before it can be recognized as ready, the default is 1, if set to 2, the container needs to be detected successfully 2 times in a row before it can be recognized as ready.

As shown in Figure 16: Container Configuration:

Figure 16: Container Configuration



‹ Create Job

Image Source * Repository | External Repository

Image * xjy/worktest/busybox ×

Tag * 1.33 ×

Resource Requ... CPU Core Memory Mi

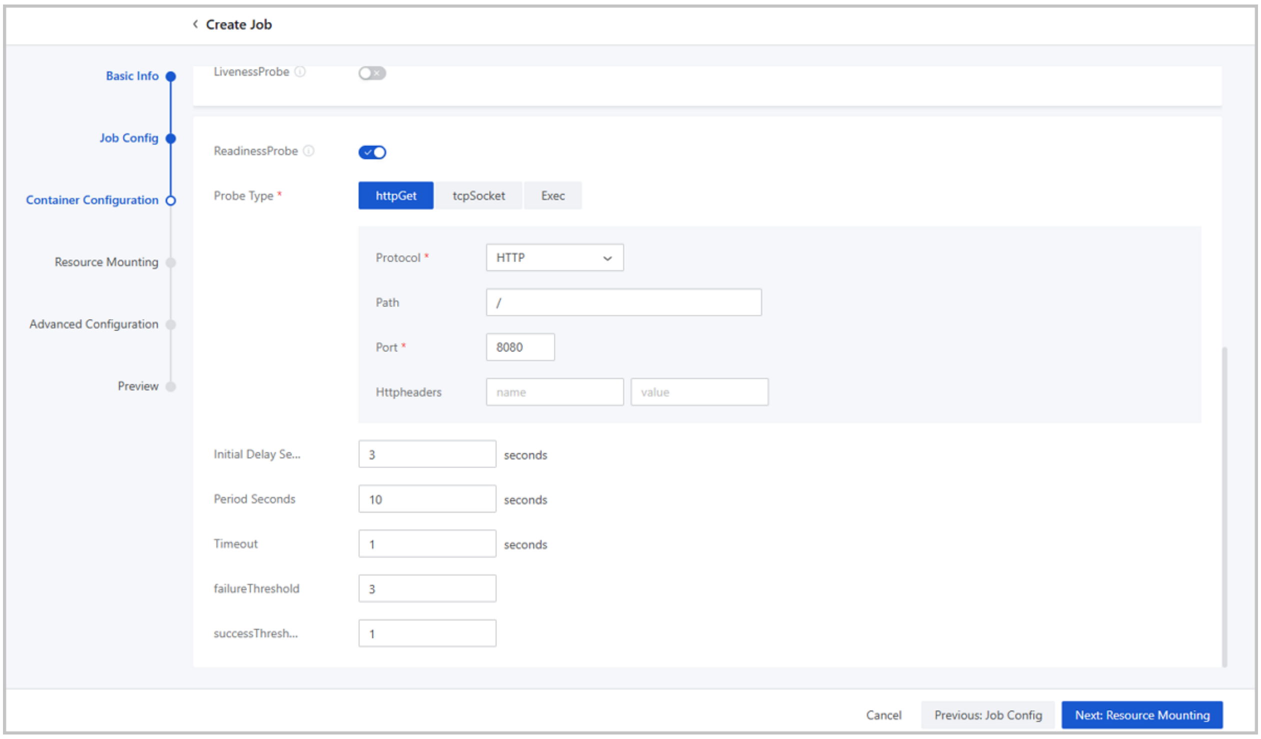
Resource Limit CPU Core Memory Mi

Command /bin/sh

Argument "-c","sleep 100"

Env + Add Env

LivenessProbe



‹ Create Job

LivenessProbe

ReadinessProbe

Probe Type * httpGet | tcpSocket | Exec

Protocol * HTTP

Path /

Port * 8080

Httpheaders name value

Initial Delay Se... 3 seconds

Period Seconds 10 seconds

Timeout 1 seconds

failureThreshold 3

successThresh... 1

4. resource mounting

- **Mount PVC:** mount a PVC for the job to provide persistent storage. Click **Add PVC** and refer to the following example to enter the appropriate contents:

# **PVC**: select the pvc, only the PVC under the namespace where the current job is located can be selected.

Note: A pvc can only be mounted to one job at a time.

# **Mount path**: Set the mount path of the pvc on the container.

# **Permission**: set pvc permission s, including read-only, read and write permission s.

- Mount ConfigMap: mount ConfigMap to the container to add configuration data for the job. Click **Add ConfigMap** and refer to the following example to enter the appropriate content:

  # **Configmap**: select a configmap

  # **Mount content**: select the content to be mounted, you can choose to mount all keys under the configmap, or only mount one or more specific keys

  If you choose to mount all, you need to set the following parameters:

  - **mount path**: set the mount path of the ConfigMap on the container, each key under the ConfigMap will generate a configuration file under the path, the file name is called the key name, the file content is the key value

  If you choose to mount **a specific key**, you need to set the following parameters:

  - **Key**: Select a specific key
  - **Mount path**: set the mount path of the key on the container
  - **SubPath**: select whether to enable subPath, default not enabled

    # If subPath is enabled, the key will generate a configuration file under the penultimate level of the mount path, the configuration file name is the last level of the mount path, and the file content is the key value. For example, if you fill in the mount path as /data/config.conf, the key will generate a file named config.conf under the /data/ directory of the container, and the original file under the /data/ directory will not be overwritten.

    # If subPath is not enabled, the key will generate a configuration file under the last level of the mount path with the key name and the file content as the key value, and the original file under the mount path will be overwritten.

- Mount Secret: Mount a Secret into container. Click **Add Secret** and refer to the following example to enter the corresponding content:

# **Secret**: select a Secret

# **Mount content**: select the content to be mounted, you can choose to mount all keys under the Secret or only one or more specific keys

If you choose to mount **all**, you need to

set the following parameters:

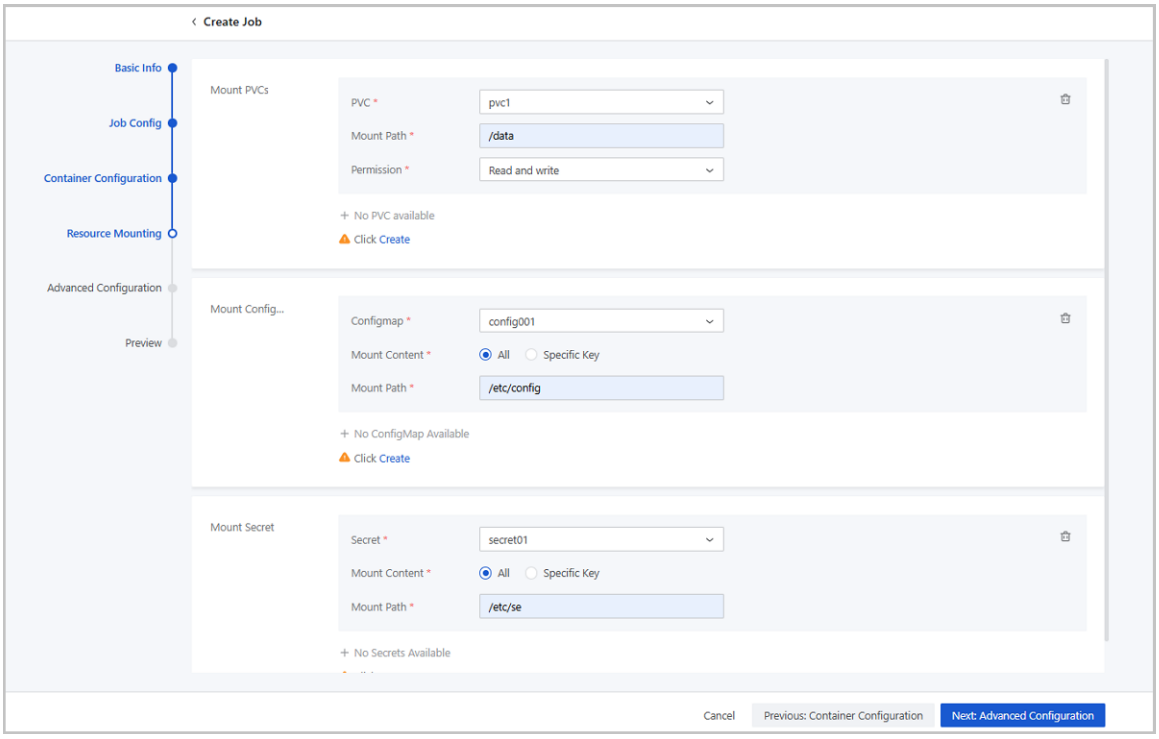
- **mount path**: set the mount path of the Secret on the container, each key under the Secret will generate a configuration file under the path, the file name is the key name, the file content is the key value

If you choose to mount **a specific key**, you need to set the following parameters:

- **Key**: Select a specific key
- **Mount path**: set the mount path of the key on the container
- **SubPath**: select whether to enable subPath, default is not enabled, the definition and usage of subPath is the same as the subPath when mounting configmap

As shown in Figure 17: Resource Mounting:

Figure 17: Resource Mounting



Mount PVCs
PVC *: pvc1
Mount Path *: /data
Permission *: Read and write

Mount Config...
Configmap *: config001
Mount Content *: ● All ○ Specific Key
Mount Path *: /etc/config

Mount Secret
Secret *: secret01
Mount Content *: ● All ○ Specific Key
Mount Path *: /etc/se

5. For advanced configuration, refer to the following example to enter the appropriate content:

- **GPU Configuration**: Default is off, if you want to use GPU resources, it is on, you need to set the following parameters:

  # **GPU manufacturer**: Select the manufacturer of the GPU manufacturer you want to use, currently supports NVIDIA and Huawei,HYGON,Iluvatar.

  When **NVIDIA** is selected as the GPU manufacturer, set the following parameters:

  ■ **GPU Selection Method**: Select the GPU selection method, which supports the following three methods: Do not specify GPU, Specify GPU Model, and Specify GPU.

  ■ Do not specify GPU： The GPU to be used by the pod is not specified. The platform schedules it to any GPU that meets the GPU resource configuration based on GPU resource allocation.

  ■ Specify GPU Model： Select one or more GPU models, and the workload pod will be scheduled to the GPUs that meet the GPU resource configuration of the selected model.

  ■ Specify GPU： Select one or more GPUs. The workload pods will be scheduled to the GPUs that meet the GPU resource configuration among the selected GPUs.

  ■ **GPU model/GPU**: Specify the GPU model or GPU.

  ■ **GPU Number**: sets the number of GPUs the container needs to use

  Note: Containers can only use the GPU node of the node where the pod is located and cannot be used across nodes.

  ■ **GPU memory**: Sets the size of the memory of a single GPU to be used by

  the container. Selectable units: Gi, Mi

  Notes.

  - If the number of GPUs is set greater than 1, each GPU requests GPUs per GPU memory size.
  - Per GPU memory cannot exceed the maximum video memory of a single GPU.

  ■**GPU Core Percentage**: Sets the percentage of GPU core on a

  single GPU that the container needs to use

Notes.

- If the Number of GPUs is set greater than 1, each GPU can use up to a percentage of arithmetic per GPU core.
- If the per-GPU core percentage is not set, it defaults to 0. The container pod may be scheduled to any GPU that meets the GPU memory requirements.
- The container uses the GPU on a shared basis, and if exclusive GPU core is required, set the per GPU core percentage to 100%.

When you select **huawei** for the GPU manufacturer, set the following parameters:

■ **GPU model**: Select the model of the GPU model to be used, and the available models are all the huawei GPU models in the current cluster.

■ **GPU Number** : sets the number of GPUs the container needs to use

Note: Containers can only use the GPU node of the node where the pod is located and cannot be used across nodes.

■ **Exclusive GPU**: Select whether to exclusive GPU. The default is off, then the container uses the GPU in a shared way, if you need to use the whole card you can turn on this configuration

Note: Huawei does not support multi-card sharing, i.e., if the number of GPUs is configured to be greater than 1, you can only have an exclusive GPU.

■ **GPU memory**: Select the size of the graphics memory of a single GPU to be used by the container.

Note: Huawei only allows GPUs to be sliced according to a fixed number of specifications, and the number of AI Core cores corresponding to the memory is also fixed, the slicing specifications of different GPUs are different, the specific specifications are shown in the drop-down box of GPU memory per GPU.

When you select **HYGON** for the GPU manufacturer, set the following parameters:

■ **GPU model**: Select the model of the GPU to be used, and the available models are all the HYGON GPU models in the current cluster.

Note: If the namespace is configured with Hygon GPU memory quota, the GPU model must be specified when using an exclusive GPU card; otherwise, the Pod will fail to be created.

- **GPU Number**: sets the number of GPUs the container needs to use

Note: Containers can only use the GPU node of the node where the pod is located and cannot be used across nodes.

- **Exclusive GPU**: Select whether to exclusive GPU. The default is off, then the container uses the GPU in a shared way, if you need to use the whole card you can turn on this configuration

Note: HYGON does not support multi-card sharing, i.e., if the number of GPUs is configured greater than 1, you can only have an exclusive GPU.

- **GPU memory**: Sets the size of the memory of a single GPU to be used by the container. Selectable units: Gi, Mi

Note: GPU memory cannot exceed the maximum memory of a single GPU.

- **GPU Core Percentage**: Sets the percentage of arithmetic on a single GPU that the container needs to use

When you select **Iluvatar** for the GPU manufacturer, set the following parameters:

- **GPU model**: Select the model of the GPU to be used, and the available models are all the Iluvatar GPU models in the current cluster.
- **GPU Number**: sets the number of GPUs the container needs to use

Note: Containers can only use the GPU node of the node where the pod is located and cannot be used across nodes.

- **Exclusive GPU**: Select whether to exclusive GPU. The default is off, then the container uses the GPU in a shared way, if you need to use the whole card you can turn on this configuration

Note: Iluvatar does not support multi-card sharing, i.e., if the number of GPUs is configured greater than 1, you can only have an exclusive GPU.

- **GPU memory**: Sets the size of the memory of a single GPU to be used by the container. Selectable units: Gi, Mi

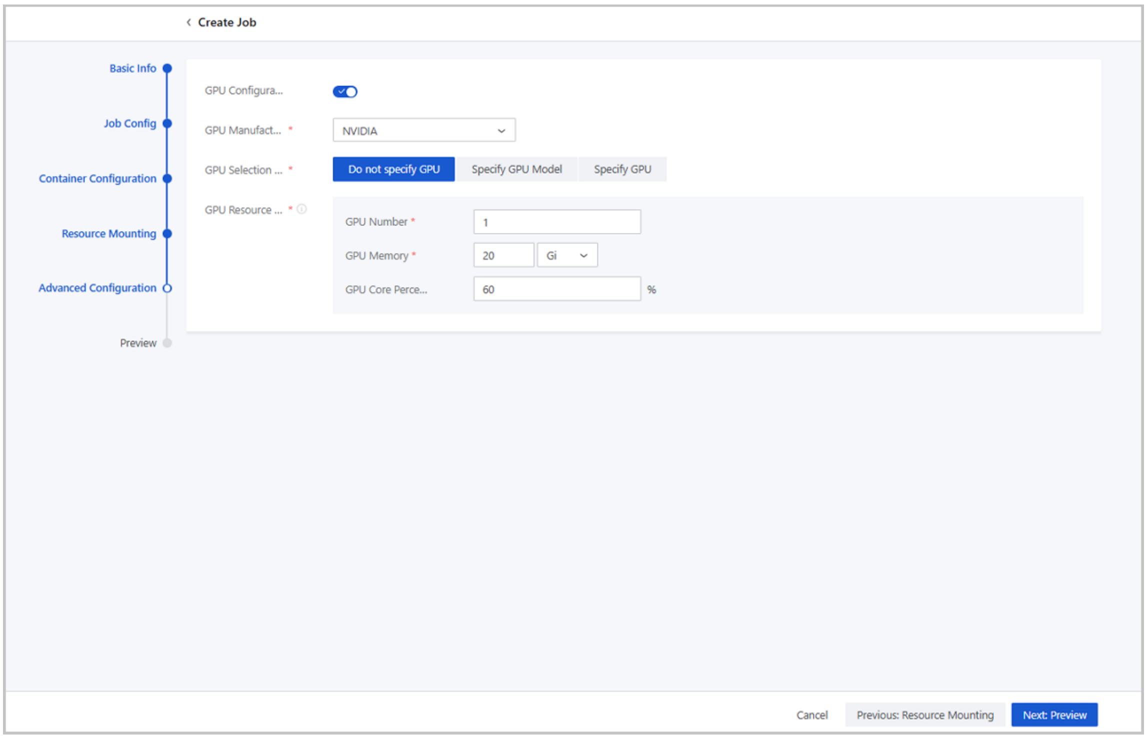
Note: The GPU Memory size must be multiple of 256 MiB and can be only an integer.

- **GPU Core Percentage**: Sets the percentage of arithmetic on a single GPU that the container needs to use

Note: To use Iluvatar GPU, the Iluvatar software stack library needs to be pre-loaded into the container.

As shown in Figure 18: Advanced Configuration:

Figure 18: Advanced Configuration



GPU Configura...: on

GPU Manufact... *: NVIDIA

GPU Selection ... *: [✓ Do not specify GPU] Specify GPU Model Specify GPU

GPU Resource ... *

GPU Number *: 1

GPU Memory *: 20 Gi

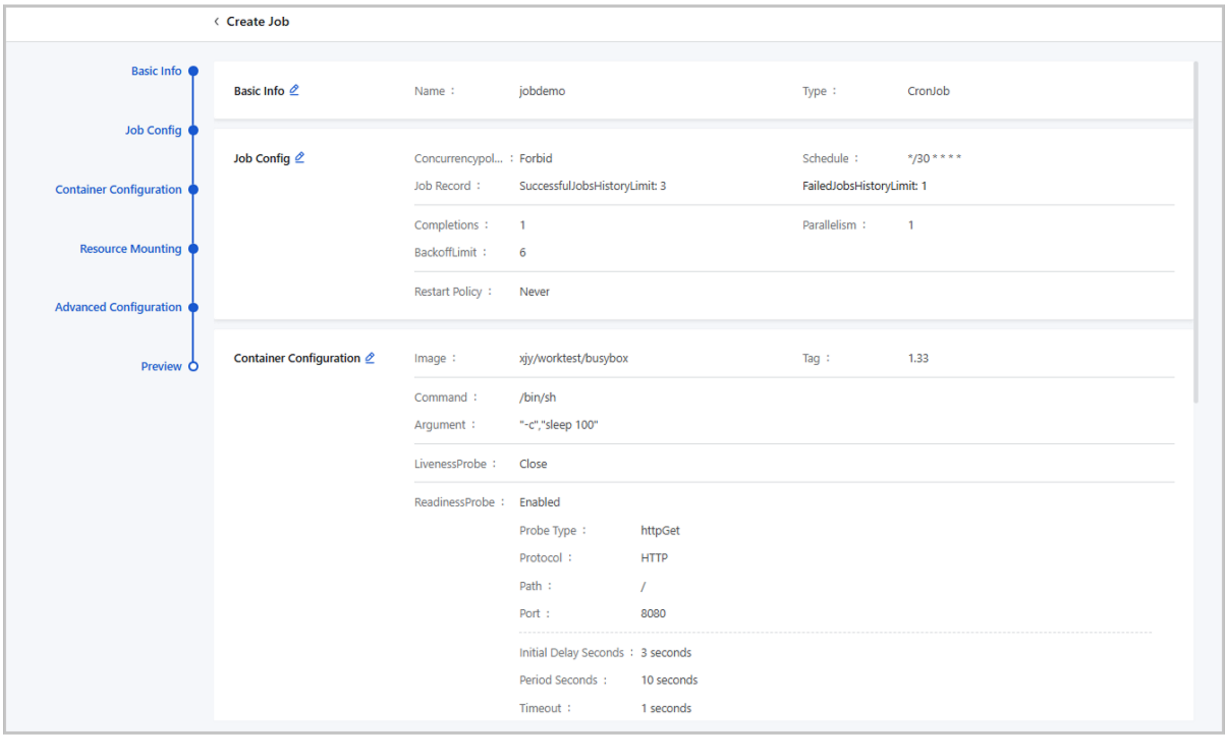
GPU Core Perce...: 60 %

6. Preview

View the job that will be created, with support for jump modifications.

As shown in Figure 19: Confirmation Message:

Figure 19: Confirmation Message



‹ Create Job

Basic Info

Name : jobdemo

Type : CronJob

Job Config

Concurrencypol... : Forbid

Schedule : */30 * * * *

Job Record : SuccessfulJobsHistoryLimit: 3 FailedJobsHistoryLimit: 1

Completions : 1

Parallelism : 1

BackoffLimit : 6

Restart Policy : Never

Container Configuration

Image : xjy/worktest/busybox

Tag : 1.33

Command : /bin/sh

Argument : "-c","sleep 100"

LivenessProbe : Close

ReadinessProbe : Enabled

Probe Type : httpGet

Protocol : HTTP

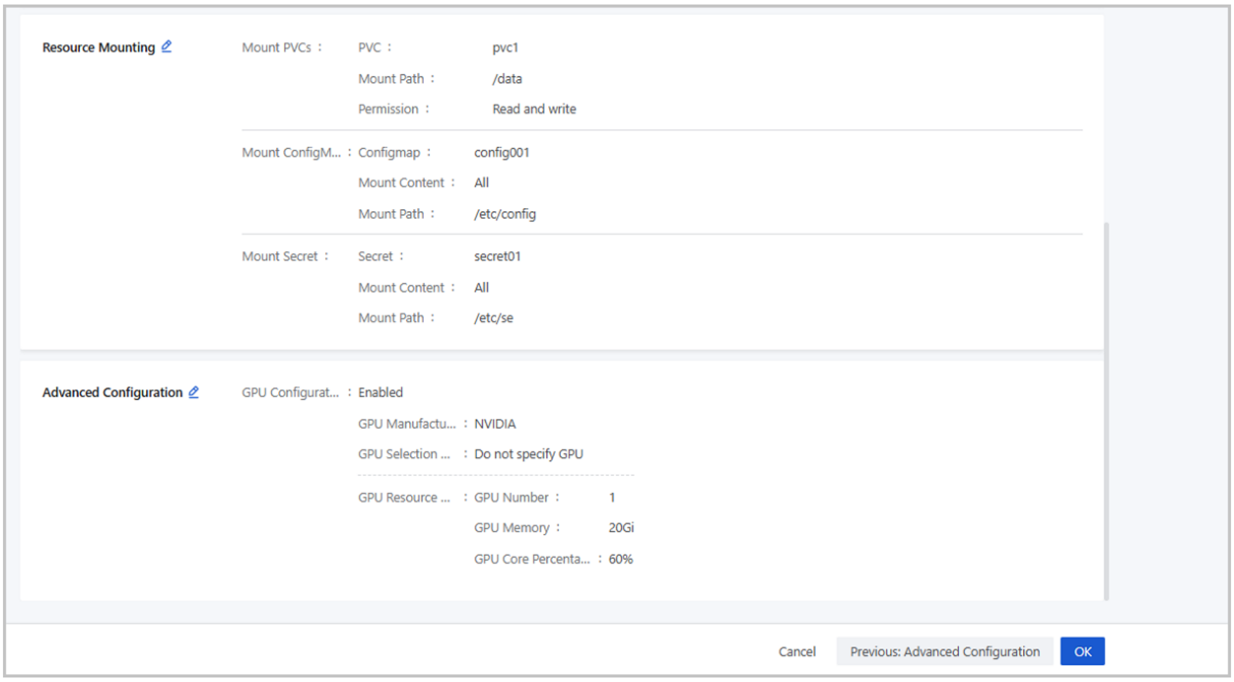
Path : /

Port : 8080

Initial Delay Seconds : 3 seconds

Period Seconds : 10 seconds

Timeout : 1 seconds



Resource Mounting

Mount PVCs : PVC : pvc1

Mount Path : /data

Permission : Read and write

Mount ConfigM... : Configmap : config001

Mount Content : All

Mount Path : /etc/config

Mount Secret : Secret : secret01

Mount Content : All

Mount Path : /etc/se

Advanced Configuration

GPU Configurat... : Enabled

GPU Manufactu... : NVIDIA

GPU Selection ... : Do not specify GPU

GPU Resource ... : GPU Number : 1

GPU Memory : 20Gi

GPU Core Percenta... : 60%

7.1.3.3 Management Job

In the nKU main menu, click **Container Orchestration** > **Application Management** > **Jobs** to access the **Jobs** screen.

The job supports the following operations:

manipulate	descriptive
Create job	Create a new job.
Update Yaml	Update the job via Yaml form. **Note:** Only CronJob supports updates.
Enabled Job	Enables a deactivated job. **Note:** Enablement is only supported for CronJob.
Disabeld Job	Deactivates an enabled job. **Note:** Deactivation is supported only for CronJob.
Job details	Go to the Job Details page to view an overview of the job, execution logs, pod information, and other associated resources, such as pvc.
Delete job	Remove the job from the cluster.

7.1.4 Pod

7.1.4.1 General

A pod is the smallest unit of cluster scheduling and management cluster. Containers in the same pod share the same network and storage. Usually, only one container runs in a pod, but in a scenario where multiple containers need to be coupled and share resources, multiple containers may also run in a pod.

In nKU, Pod actually run the user's application and are managed by workloads or Jobs. In practice, the user creates the corresponding pod during the process of creating the workload or job, and rarely creates the pod directly.

7.1.4.2 Managing Pod

In the nKU main menu, click **Container Orchestration** > **Application Management** > **Pod** to access the **Pod** screen.

The following operations are supported for Pod:

manipulate	descriptive
View Yaml	View the YAML of the Pod

Open a web terminal connection.	Enter the container terminal.
View Monitor	Jump to view the pod monitoring panel.
View Log	Jump to view container logs, and you can export the logs to save them locally.
Pod details	Go to the Pod Details page to view the pod overview, resource request, monitoring data, containers, events, and other associated resources, such as volumes.
save as an image	Generate an image repository based on the current container and upload it to the Repository. **Notes.** • To perform this operation, you need to go to the Pod Details page and click **Containers** > **Operation** > **Save as an image**. • Only running Images support this operation.
Delete pod	Delete the pod. • Directly created container pods will be completely deleted. • When a pod created through a workload or job is deleted, a new pod is automatically created. The delete operation in this scenario is equivalent to restarting the pod.
Export Pod information	Export Pod information as a CSV file. Supports exporting the current page or all pages.

7.2 Network management

7.2.1 Service

7.2.1.1 General

Service provides a unified access entry point for container services, allowing applications to easily implement service discovery and load balancing.

Functional principle

Typically, because Pod are often created and destroyed so quickly, even if each pod has a fixed IP address, direct access to a pod for application services is still an unreliable way to access a pod because the pod that is running an application now may be different from the pod that will be running that application the next moment.

To provide a unified access portal to these rapidly changing Pod, the concept of a service is introduced. The service has a separate fixed IP address and automatically forwards traffic accessing this IP address to the appropriate pod, which is selected by label. Even if the pod in which the application is running on the back-end changes, users can still access it quickly and without change perception through the service. At the same time, the service can help achieve load balancing between Pod.

Type of service

nKU supports the following four service types:

- ClusterIP: Services of type ClusterIP are not exposed outside the cluster. The system assigns it a cluster IP through which the service can be accessed from within the cluster.
- NodePort: NodePort type of service is exposed to the outside world in the form of cluster node IP + static port, which can be accessed from outside the cluster through NodeIP (any node IP of the cluster):NodePort (cluster port).

  Note: Services of type NodePort are automatically created and ingressed to a ClusterIP service.
- LoadBalancer: Services of type LoadBalancer are exposed externally through nKU's metallb load balancing feature. The system assigns an external network IP to it, through which the service can be accessed from outside the cluster.
- Headless: Headless type services do not have a fixed IP address, and after the client accesses the domain name of the service, it can receive the IP addresses of all Pod returned via DNS and can realize load balancing of the back-end Pod via DNS.Headless type services are usually paired with StatefulSet workloads for accessing url between replicas of each pod. Headless-type services are usually used with StatefulSet workloads for access between replicas of Pod.

7.2.1.2 Creating services

In the nKU main menu, click **Container Orchestration** > **Network Management** > **Services** to enter **the Services** interface. Click **Create Service** to bring up the **Create Service** screen.

You can refer to the following example to enter the appropriate contents:

- **Name**: service settings name. Naming rules: length limit 1-50 characters, can only contain lowercase letters, numbers and separator (-), and must start with a lowercase letter and end with a number or lowercase letter
- **Associated Workload**: Related workloads for services, traffic accessing the service will be forwarded to the container containers under the related workloads, supports two types of related methods: selecting workloads, related by labels

  # Select workloads: directly select existing workloads

  # Related by label: Fill in the pod label key and value of the workload, and the system will automatically filter the workload to be related services based on the label.

  Note: The service uses the label to determine which pod to forward traffic to, if the pod contains all the labels that the user has filled in here, then the service will forward traffic to that pod

- **Type**: Select the type of service, see Service Type for details on the type definition.

  Note: When the type is Headless and the associated workload is a StatefulSet , the service name must be set to be consistent with the service name in the advanced configuration of the StatefulSet.

- **IP Type**: Select the type of service IP address, currently only supports IPv4 type.
- **Service External network**: optional, you can fill in this item when the type is LoadBalancer, select an Service external network for LoadBalancer service, users can access LoadBalancer service from outside the cluster through the external network IP.
- **Specify IPv4**: Optional, you can fill in this item after selecting the Service external network to manually specify the external network IPv4 for the LoadBalancer service, if not, the external network will randomly assign an IPv4 for the service.

  Note: The specified IP address must be within the selected external network range and avoid address conflicts.

- **Service Port**: Set the following information:

  # **Port**: sets the port on which the service settings receive traffic

  # **Container port**: sets the container port to which the service will forward traffic

  # **Port name**: set the name of the port, if not set then the default is "ze-protocol-port-container-port", such as zetcp-80-80

# **Host port**: optional, this item can be set when the type is NodePort, host port is NodePort, users can access the NodePort service port from outside the cluster by using NodeIP:NodePort, the value range is 30000-32767, if you do not set it, the cluster will randomly allocate a port

# **Protocol**: set the access protocol type, support TCP, UDP two protocol types

- **Session affinity**: Choose whether to enable session hold, when enabled, access requests from the same client IP within the specified time will all be forwarded to the same container pod, default is not enabled

Note: If the client request is forwarded by nginx or other proxy servers, the system may recognize the proxy server IP as the client IP.

- **Session affinity timeout**: If session hold is enabled, you need to set the maximum session sticky time, the default is 10800, unit: second, value range: 300-86400 range of integers
- **Label**: optional, add service labels

# **key:** fill in the label key name

Notes.

- Typically, key names can contain 1-63 characters, including letters, numbers, and the following special characters: (-)(_)(.) Special characters cannot be used at the beginning or end of a key name.
- Supports prefixing key names and concatenating them with /, e.g. k8s.io/app, all characters filled in before / will be considered as key prefixes and all characters after will be considered as key names.
- The key name prefix can contain 1-253 characters, including lowercase letters, numbers and the following special characters: (-)(_)(.) (-)(_)(.), 2 or more special characters cannot be used consecutively, and special characters cannot be used at the beginning or end of the prefix.
- Please note that you do not have to use a key name prefix, but you must fill in the key name.

# **value**: fill in the value of the label

- **Annotation**: optional, add service annotations

# **key:** fill in the annotation key name

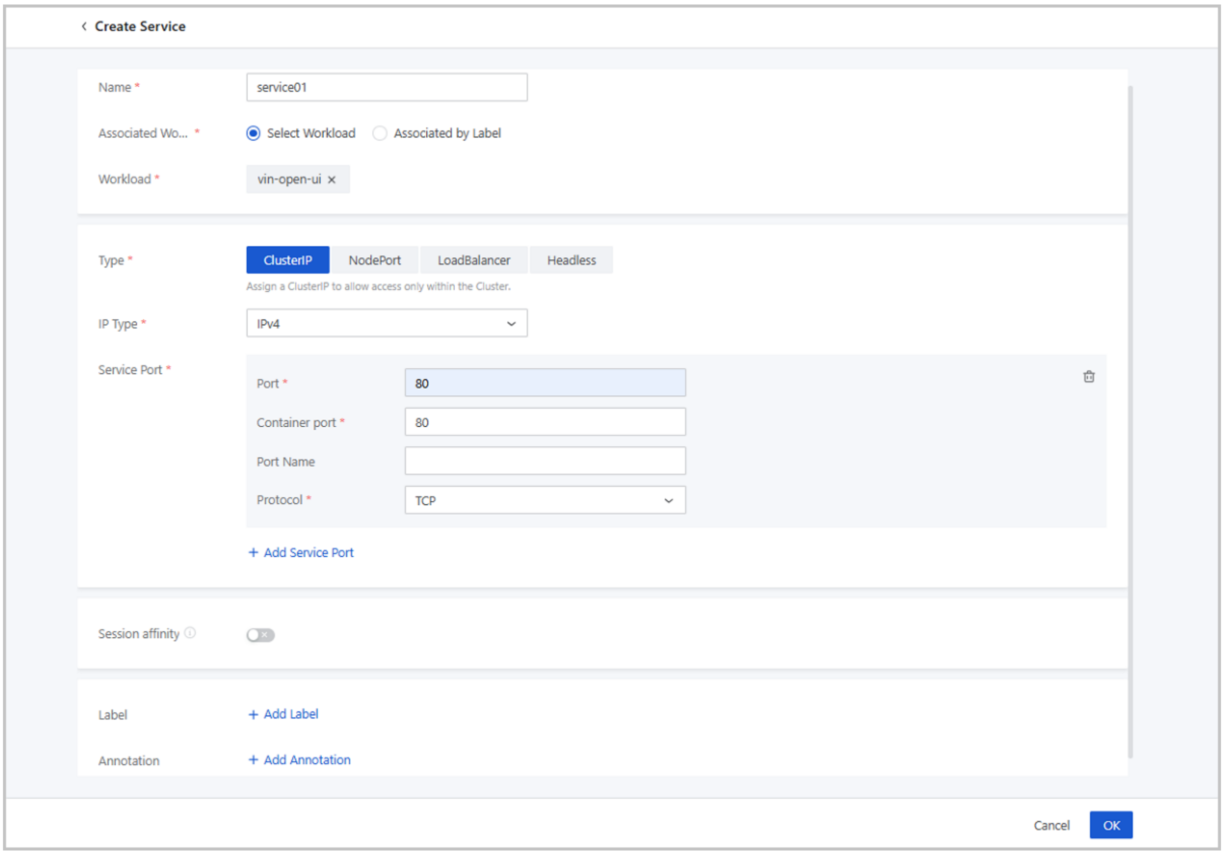
Notes.

- Typically, key names can contain 1-63 characters, including letters, numbers, and the following special characters: (-)(_)(.) Special characters cannot be used at the beginning or end of a key name.
- Supports prefixing key names and concatenating them with /, e.g. k8s.io/app, all characters filled in before / will be considered as key prefixes and all characters after will be considered as key names.
- The key name prefix can contain 1-253 characters, including lowercase letters, numbers and the following special characters: (-)(_)(.) (-)(_)(.), 2 or more special characters cannot be used consecutively, and special characters cannot be used at the beginning or end of the prefix.
- Please note that you do not have to use a key name prefix, but you must fill in the key name.

# **Value:** Fill in the value of the annotation

as shown in Figure 20: Creating Service:

Figure 20: Creating Service



Create Service

Name * service01

Associated Wo... * ● Select Workload ○ Associated by Label

Workload * vin-open-ui ×

Type * ClusterIP

Assign a ClusterIP to allow access only within the Cluster.

IP Type * IPv4

Service Port *

Port * 80

Container port * 80

Port Name

Protocol * TCP

Session affinity

7.2.1.3 Management services

In the nKU main menu, click **Container Orchestration** > **Network Management** > **Services** to enter **the services** interface.

The service supports the following operations:

manipulate	**descriptive**
Create service	Create a new service.
Edit service	Update the service configuration via a form. **Note:** If the current service is created through Yaml form and the value of spec.ipFamilies is **IPv6** or **IPv6,IPv4**, the service does not support modification through form. Users can click **Update Yaml** to modify the service configuration.
Update Yaml	Update the service configuration via Yaml form.
Service Details	Go to the Service Details page to view basic service information, port information, and events.
Delete service	Remove the service from the cluster.

7.2.2 Ingress

7.2.2.1 General

Routing (Ingress) is a set of HTTP or HTTPS routing rules used for external access to services (Servcie) within the cluster, providing externally accessible URLs, load balancing, SSL termination, and HTTP routing for cluster services. The ingress feature has 7 layers of load balancing capability to forward traffic to different services based on different URLs.

Functional realization

The routing functionality relies on Ingress Controller implementations such as Nginx Ingress Controller or Traefik, etc. Ingress Controller is an application running in a Kubernetes cluster that listens for changes in routing resources and updates the configuration of the load balancer or reverse proxy server. thereby enabling access to the service by external traffic according to the ingress rules.

nKU application market provides **nginx-ingress-controller** deployment chart, users can enter the application market, one-click to install the deployment chart, install Nginx Ingress Controller, you can also deploy your own Ingress Controller by other means.

7.2.2.2 Creating an ingress

In the nKU main menu, click **Container Orchestration** > **Network Management** > **Ingress**. In the **Routing** screen, click **Create Route** to bring up the **Create Route** screen.

You can refer to the following example to enter the appropriate contents:

- **Name**: Set the name of the ingress. Naming rules: length limit 1-50 characters, can only contain lowercase letters, numbers and separator (-), and must start with a lowercase letter and end with a number or lowercase letter
- **IngressClass**: Select IngressClass to specify the Ingress Controller that handles ingresses

  Note: Please make sure that Ingress Controller is installed in advance, if not, you can go to **the artifact repository** > **application market** to install **application nginx-ingress-controller** with one click.

- **Ingress Rule**: Configure routing rules, multiple routing rules are supported to be configured under one route, you can refer to the following example to enter the corresponding contents:

  # **Domain**: optional, fill in the actual access url of the domain name. After the ingress is created, you can configure the domain name resolution rules in the DNS service, so that the domain name is resolved to the gateway IP of the route, and the domain name access can be realized

  # **Protocol**: select the protocol type, support HTTP, HTTPS two protocol types

  # **Secret**: When the protocol type is **HTTPS**, you need to bind a Secret of TLS type to support HTTPS data transmission encryption authentication.

  # **PathType**: select the pathType. PathType determines how the path is matched, the following three path types are supported, the default is ImplementationSpecific:

  ■ ImplementationSpecific: the path matching method is determined by IngressClass, which may use a separate matching method or the same matching method as Exact or Prefix.

  ■ Exact: match the URL path exactly, and case-sensitive. For example, if the path is set to /foo and the request path is /foo, the request matches the path; if the request path is /foo/, the request does not match the path.

- Prefix: matches based on /-separated URL path prefixes. Matching is case-sensitive and done for elements in the path one by one. A path element is a list of labels in a path separated by /. A request matches a path if each part of the path is prefixed by an element of the requested path. For example, if the path is set to /foo and the request path is /foo, the request matches the path; if the request path is /foo/, the request also matches the path.

# **Path**: Specify the path of the URL that is allowed to be accessed, the default is the root path /

# **BackendService**: select the service being accessed

# **BackendService Port**: Select the service port to be accessed.

- **Label**: optional, add labels

# **key:** fill in the label key name

Notes.

- Typically, key names can contain 1-63 characters, including letters, numbers, and the following special characters: (-)(_)(.) Special characters cannot be used at the beginning or end of a key name.
- Supports prefixing key names and concatenating them with /, e.g. k8s.io/app, all characters filled in before / will be considered as key prefixes and all characters after will be considered as key names.
- The key name prefix can contain 1-253 characters, including lowercase letters, numbers and the following special characters: (-)(_)(.) (-)(_)(.), 2 or more special characters cannot be used consecutively, and special characters cannot be used at the beginning or end of the prefix.
- Please note that you do not have to use a key name prefix, but you must fill in the key name.

# **value**: fill in the value of the label

- **Annotation**: optional, add annotations

# **key:** fill in the annotation key name

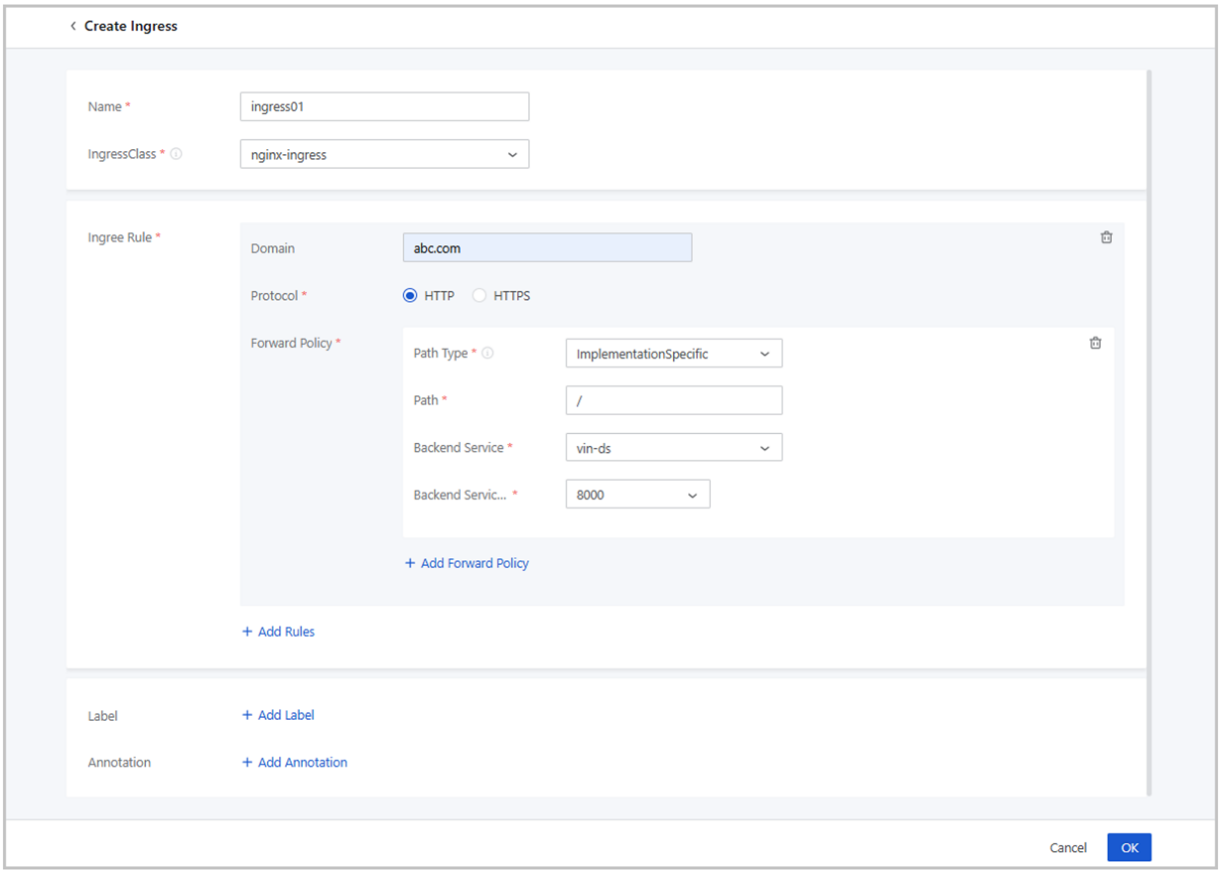
Notes.

- Typically, key names can contain 1-63 characters, including letters, numbers, and the following special characters: (-)(_)(.) Special characters cannot be used at the beginning or end of a key name.
- Supports prefixing key names and concatenating them with /, e.g. k8s.io/app, all characters filled in before / will be considered as key prefixes and all characters after will be considered as key names.
- The key name prefix can contain 1-253 characters, including lowercase letters, numbers and the following special characters: (-)(_)(.) (-)(_)(.), 2 or more special characters cannot be used consecutively, and special characters cannot be used at the beginning or end of the prefix.
- Please note that you do not have to use a key name prefix, but you must fill in the key name.

# **value:** fill in the value of the annotation

As shown in Figure 21: Creating an Ingress:

Figure 21: Creating an Ingress



Create Ingress

Name * ingress01

IngressClass * nginx-ingress

Ingree Rule *

Domain abc.com

Protocol * ● HTTP ○ HTTPS

Forward Policy *

Path Type * ImplementationSpecific

Path * /

Backend Service * vin-ds

Backend Servic... * 8000

7.2.2.3 Administering the ingress

In the nKU main menu, click **Container Orchestration** > **Network Management** > **Routing** to enter the **Routing** interface.

The following operations are supported by the ingress:

manipulate	descriptive
Create Ingress	Create a new ingress.
Update Yaml	Updates to the ingress via Yaml.
Edit IngressClass	Modify the IngressClass associated with the ingress.
Edit the routing rules	Modify the ingress rules.
Edit label and annotations	Editing labels, annotations of ingresses
Ingress Details	Go to the Route Details page to view basic route information and events.
Delete Ingress	Remove the ingress from the cluster.

7.2.2.4 Ingress instructions

7.2.2.4.1 Deploy Ingress Controller

background information

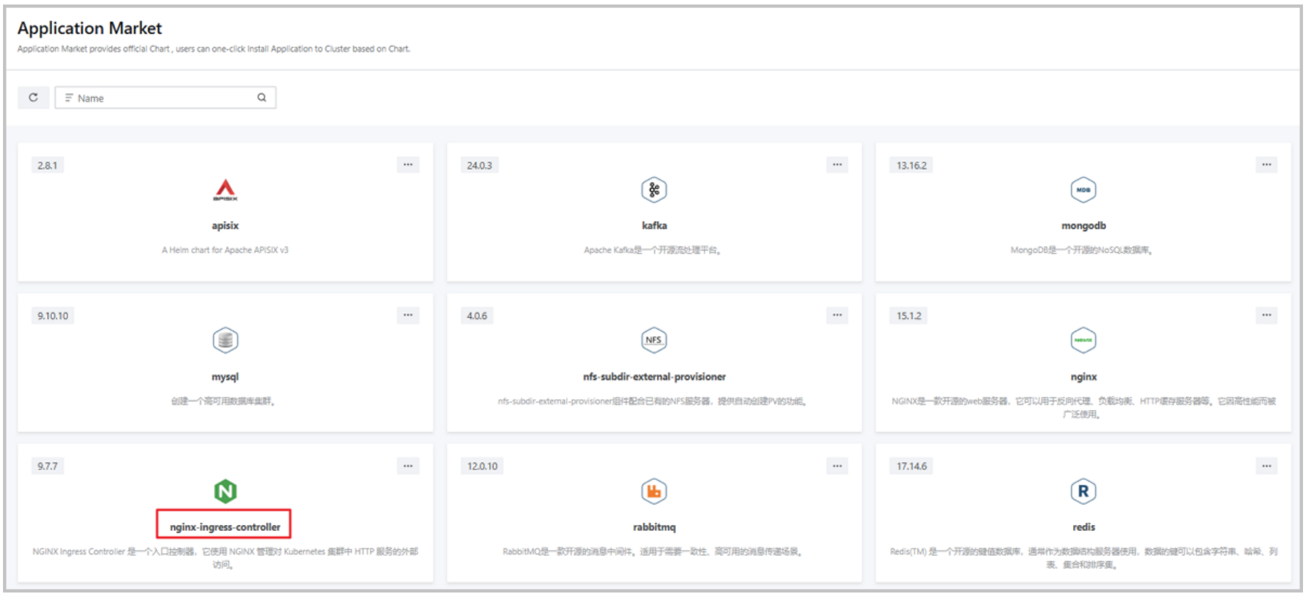
An ingress defines the rules for processing incoming traffic to the cluster over the HTTP and HTTPS protocols, and the Ingress Controller is responsible for enforcing these rules. Therefore, before creating a ingress, it is important to ensure that Ingress Controller has been properly installed in the cluster.This section focuses on how to install Ingress Controller via the **nginx-ingress-controller** deployment chart built into the nKU application market.

procedure

1. In the nKU main menu, click the **artifact repository** > **Application Market** to access the **application market** interface.
2. In the **application market**, find the **nginx-ingress-controller** chart and click on **Action** > **to install the application**.

As shown in Figure 22: nginx-ingress-controller chart:

Figure 22: nginx-ingress-controller chart



Application Market

Application Market provides official Chart , users can one-click Install Application to Cluster based on Chart.

2.8.1
apisix
A Helm chart for Apache APISIX v3

24.0.3
kafka
Apache Kafka是一个开源流处理平台。

13.16.2
mongodb
MongoDB是一个开源的NoSQL数据库。

9.10.10
mysql
创建一个高可用数据库集群。

4.0.6
nfs-subdir-external-provisioner
nfs-subdir-external-provisioner组件配合已有的NFS服务器，提供自动创建PV的功能。

15.1.2
nginx
NGINX是一款开源的web服务器。它可以用于反向代理、负载均衡、HTTP缓存服务器等。它因高性能而被广泛使用。

9.7.7
nginx-ingress-controller
NGINX Ingress Controller 是一个入口控制器，它使用 NGINX 管理对 Kubernetes 集群中 HTTP 服务的外部访问。

12.0.10
rabbitmq
RabbitMQ是一款开源的消息中间件，适用于需要一致性、高可用的消息传递场景。

17.14.6
redis
Redis(TM) 是一个开源的键值数据库，通常作为数据结构服务器使用，数据的键可以包含字符串、哈希、列表、集合和排序集。

3. On the **install application** screen, refer to the following example to enter the appropriate content:

   - **Application Name**: Set the name of the application. The length is limited to 2-20 characters, the input can only contain lowercase letters, numbers and special characters (-), and must start with a lowercase letter and end with a lowercase letter or number.
   - **Version**: Setting the application version
   - **Description**: optional, note application-related information
   - **Cluster**: Select the cluster to which the application is installed
   - **Namespace**: select the namespace to which the application is installed
   - **Chart version**: display the default Chart version, currently does not support changes
   - **Application Configuration**: Modify the configuration contents of the VALUES file in the deployment chart

     Notes.

     - If you are installing multiple Nginx Ingress Controllers in the same cluster, it is recommended that you modify the following configuration, as duplicate configurations of multiple Nginx Ingress Controllers may create conflicts:

       # nameOverride: IngressController name

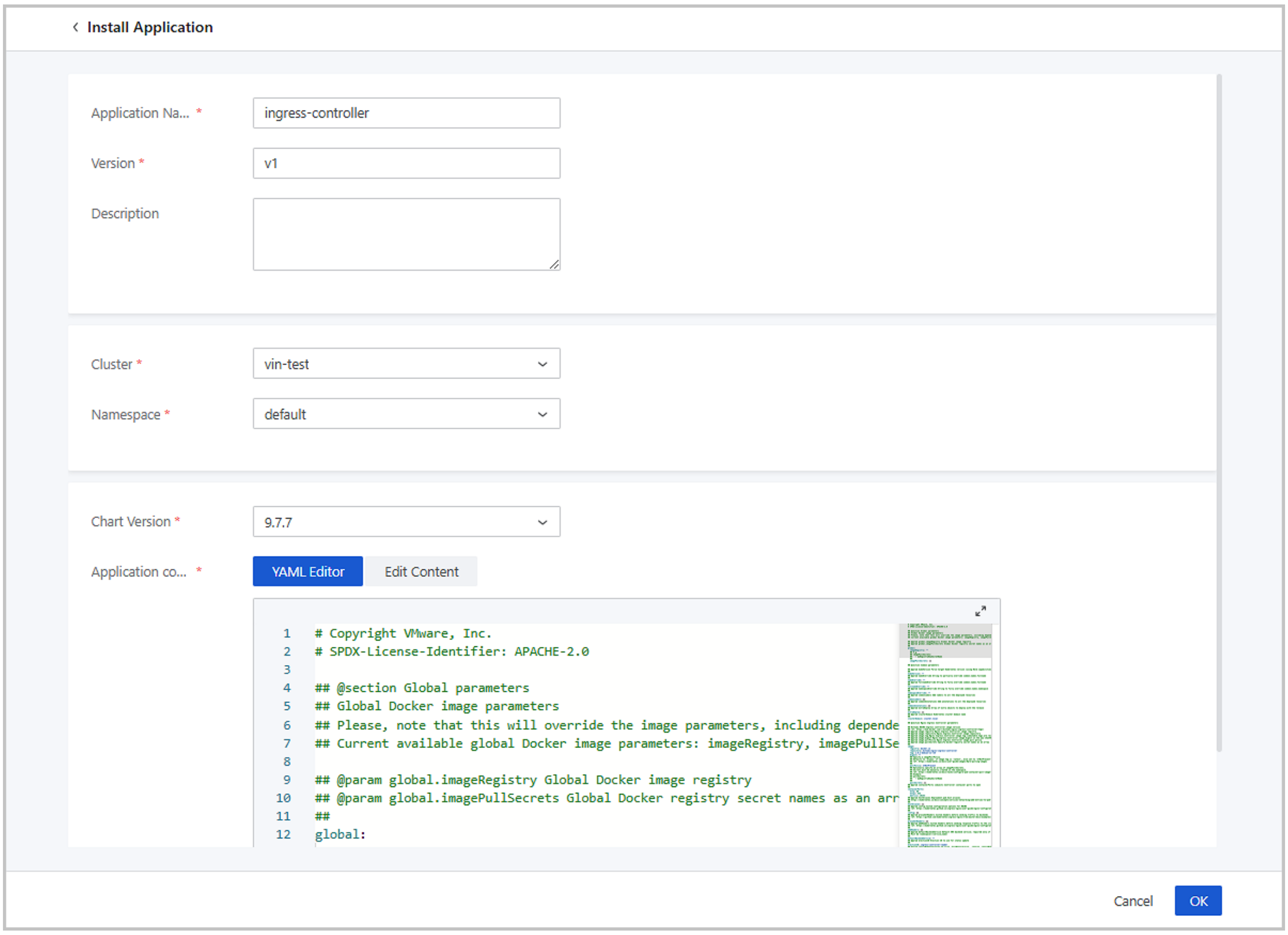
       # ingressClassResource.name: IngressClass name

# ingressClassResource.controllerClass: align with nameOverride

- When there is a lack of available external network IPs, you can set the value of the service.type parameter to NodePort, and when set, the gateway address of the ingress using this Ingress Controller will use the cluster Node IP.

As shown in Figure 23: Installing an application:

Figure 23: Installing an application



‹ Install Application

Application Na... * ingress-controller

Version * v1

Description

Cluster * vin-test

Namespace * default

Chart Version * 9.7.7

Application co... *

```
# Copyright VMware, Inc.
# SPDX-License-Identifier: APACHE-2.0

## @section Global parameters
## Global Docker image parameters
## Please, note that this will override the image parameters, including depende
## Current available global Docker image parameters: imageRegistry, imagePullSe

## @param global.imageRegistry Global Docker image registry
## @param global.imagePullSecrets Global Docker registry secret names as an arr
##
global:
```

7.2.2.4.2 Access to services via ingress

background information

This section describes how to access the service from the outside through the ingress.

procedure

1. Create the ingress.

   In the nKU main menu, click **Container Orchestration** > **Network Management** > **Ingress**. In the **Ingress** interface, click **Create Ingress** to bring up the **Create Ingress** interface, set the basic information of the route and configure the routing rules, see Creating an ingress for details.

2. View the routing gateway address and the domain name in the ingress rules.

   In the **Ingress** interface, click the Ingress created in Step 1 to enter the Ingress Details page to view the IP address of the routing gateway in the **Basic Information** and the domain name configured for each rule in the **Rules**.

3. Configure domain name resolution rules.

   Login to the DNS server or client and configure the domain name resolution rules in the hosts file so that the domain name resolves to the gateway IP address of the ingress.

4. View access url.

   Go to the Ingress Details page, expand Rules, and view the access url for the forward policy.

5. Access the corresponding backendService via the access url.

7.2.3 Network policy

7.2.3.1 General

NetworkPolicy is a policy that controls the behavior of workload-to-workload and workload-to-external network traffic to protect applications from network attacks by filtering inbound and outbound traffic.

Functional Features

Network Policy Rules

A network policy contains one or more rules that actually control the communication behavior of workloads. According to the direction of the controlled traffic, they can be categorized as ingress rules and egress rules:

- **Ingress rules**: control the source of traffic accessing the workload from the outside.nKU offers three ways to set up ingress rules:

# Allow all traffic: all traffic can access the current workload regardless of source

# Deny all traffic: All traffic is not allowed to access the current workload, regardless of source.

# Customize rules: Manually specify access rules for the current workload, which can include: allowed ingress from, access protocol, port

- ■ Traffic source: the source of traffic allowed ingress from the current workload, which can be specified as a namespace, a workload, or a CIDR
- ■ Access protocol: the access protocol used by the traffic source to access the workload, supporting two protocol types: TCP and UDP.
- ■ Port: the port on which the current workload can be accessed by the specified source

Note: Custom rules use a whitelisting mechanism, i.e., after adding a custom rule, no traffic other than that specified by the rule is allowed to access the current workload.

- **Egress rules**: control where traffic from the workload goes.nKU offers two ways to set up Egress rules:

# Allow all traffic: workloads can access any destination regardless of where the traffic is going

# Customized rules: Manually specify egress rules for the current workload, which can include: Allowed Egress To, access protocol, and port.

- ■ Allowed Egress To： The purpose for which the current workload is allowed access can be specified as one or more namespaces, workloads, or CIDR blocks.
- ■ Access protocol: the access protocol used by the workload to access the specified destination, supporting TCP and UDP access types.
- ■ Port: The port on the traffic destination that can be accessed by the current workload

Note: Custom rules use a whitelisting mechanism, i.e., after adding a custom rule, no other traffic from the workload is allowed except the traffic specified by the rule to go to the workload.

- **Number of rules and priority**: Multiple egress rules can be added under one network policy. Multiple rules are not prioritized and will take effect on loaded workloads at the same time.

Network Policy and Workloads

- If the network policy is not loaded, all outgoing and incoming traffic on the workload is allowed; when the network policy is loaded, the workload traffic will follow the network policy rules.
- Multiple network policies can be loaded for a workload. Multiple network policies have no priority of being in effect and will be in effect at the same time for that workload.
- When configuring a network policy for a workload, pay attention to the communication rules on the other end as well. For example, to enable workload A to access workload B, in addition to allowing outbound traffic from workload A to workload B, you need to ensure that workload B allows inbound traffic from workload A.

7.2.3.2 Creating a network policy

In the nKU main menu, click **Container Orchestration** > **Network Management** > **Network Policy**. In the **Network Policy** screen, click **Create Network Policy** to bring up the **Create Network Policy** screen.

You can refer to the following example to enter the appropriate contents:

- **Name**: Set the network policy name. Naming rules: length limit of 1-50 characters, can only contain lowercase letters, numbers and separator (-), and must begin with a lowercase letter and end with a number or a lowercase letter
- **Workload**: Select the workloads on which the current network policy works, either all workloads under the current namespace or only manually specified workloads.
- **Ingress Rule**: Setting up ingress rules, supporting three types of rules: Allow All Traffic, Reject All Traffic, and Custom Rules. When you select **Custom Rules**, you need to click **Add Ingress Rule** and set the following parameters:

  # **Allowed Ingress From**: Set the allowed ingress from, three types of traffic sources are supported: namespace, workload, and CIDR:

  ■ **Namespace**: Specify a namespace that allows traffic from that namespace to access the current workload

  ■ **Workload**: Specify a workload and allow traffic from that workload to access the current workload

  ■ **Network Segment**: Specify a network segment from which traffic is allowed CIDR access to the current workload

  ■ **Allowed CIDR**: The current IPv4 network segment only supports the specified 0.0.0.0/0, indicating that all network segments are allowed to access the workload

# **Protocol**: Set the protocol used by the traffic source to access the current workload, supports TCP and UDP protocol types.

# **Port**: Set the port on which the current workload can be accessed by the traffic source, supports setting ports in the range of 1-65535

- **Egress rules**: Set up egress rules, support two types of rules: Allow all traffic, Custom rules. When you choose **custom rules**, you need to click **Add egress rules** and set the following parameters:

# **Allowed Egress To**: Set allowed egress to three types of traffic: namespaces, workloads, and segments:

■ **Namespace**: Specify a namespace and allow the current workload to access that namespace

■ **Workload**: Specify a workload and allow the current workload to access that workload

■ **Segment**: Specify a segment to allow the current workload to access the segment

■ **Allowed CIDR**: Fill in the network segments that the current workload is allowed to access.

■ **Exclude CIDR**: Optional, fill in a sub-segment within the allowed CIDR range. The workload will not be allowed to access the excluded CIDR.

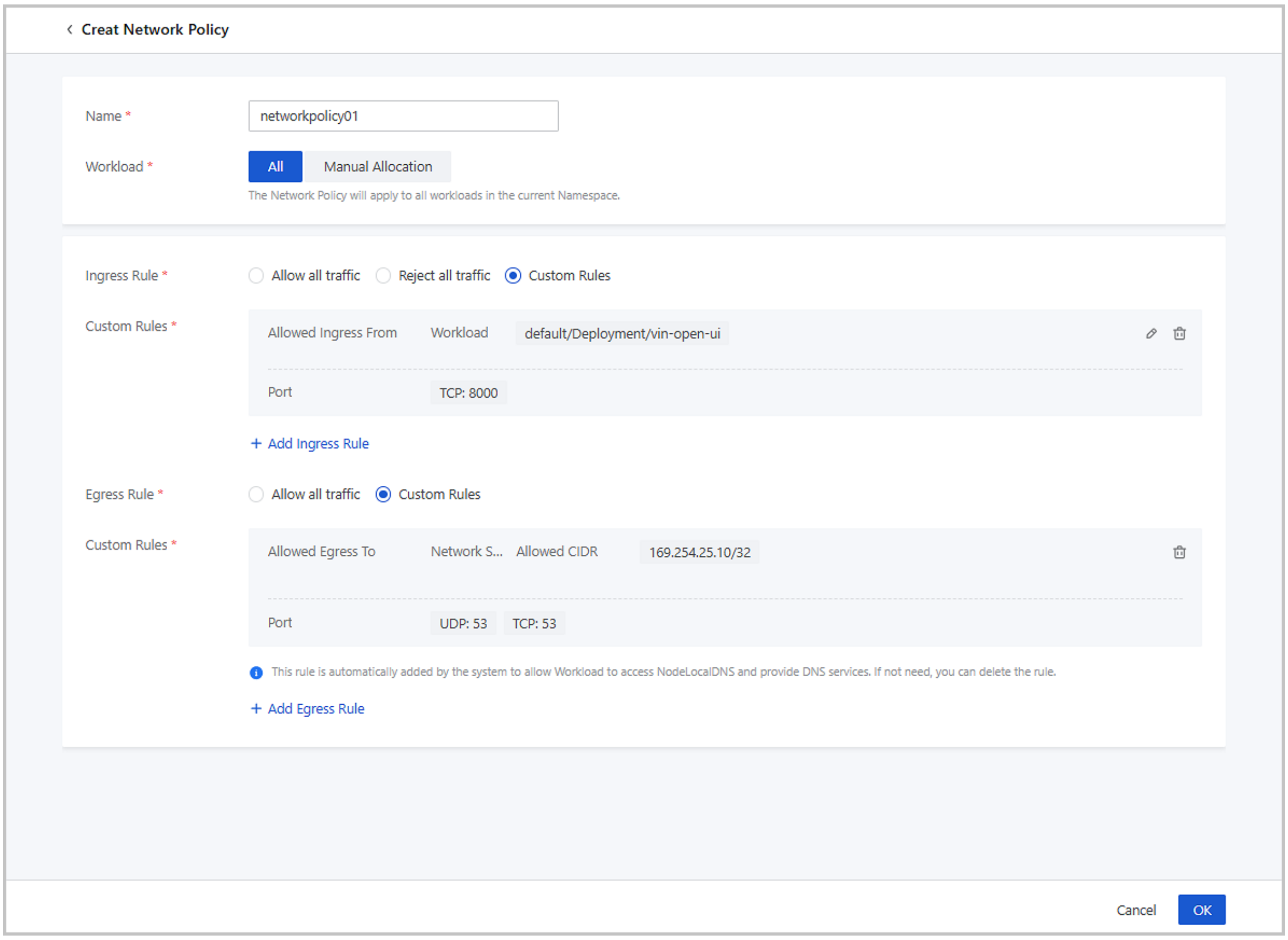
# **Protocol**: Specify the protocol used when the current workload accesses the specified destination, supporting TCP and UDP protocol types.

# **Port**: Specify the destination port that can be accessed by the current workload, supports setting ports in the range of 1-65535

Note: When customizing egress rules, the system will automatically add a rule for allowing workloads to access NodeLocalDNS for DNS service, if not needed, you can manually remove the rule.

As shown in Figure 24: Creating a Network Policy:

Figure 24: Creating a Network Policy



< Creat Network Policy

Name * networkpolicy01

Workload * All | Manual Allocation

The Network Policy will apply to all workloads in the current Namespace.

Ingress Rule * ○ Allow all traffic ○ Reject all traffic ● Custom Rules

Custom Rules * Allowed Ingress From: Workload default/Deployment/vin-open-ui

Port: TCP: 8000

+ Add Ingress Rule

Egress Rule * ○ Allow all traffic ● Custom Rules

Custom Rules * Allowed Egress To: Network S... Allowed CIDR 169.254.25.10/32

Port: UDP: 53 TCP: 53

This rule is automatically added by the system to allow Workload to access NodeLocalDNS and provide DNS services. If not need, you can delete the rule.

+ Add Egress Rule

Cancel OK

7.2.3.3 Management network policy

In the nKU main menu, click **Container Orchestration** > **Network Management** > **Network Policy** to enter the **Network Policy** interface.

The network policy supports the following operations:

manipulate	descriptive
Create network policy	Create a new network policy.
Edit network policy	Modify the network policy via a form, including mounted workloads, egress rules for outgoing/incoming traffic. **Note:** If the network policy was created in a non-formatted manner or has been modified in a non-formatted manner,modifying it in a formatted manner is no longer supported. You can modify the network policy configuration using Yaml.

Update Yaml	Modification of network policy via Yaml forms
Network Policy Details	Go to the external network details page to view basic network policy information and ingress/egress rules
Delete network policy	Delete the network policy. **Note:** After deletion, all rules under this network policy will be invalidated and will no longer control the communication behavior of the workload.

7.3 PersistentVolumes management

7.3.1 PVC

7.3.1.1 General

PersistentVolumeClaim is used to persistently store the data of a workload so that the data is retained when the pod is restarted or deleted.nKU is based on distributed storage and provides two types of data disks: exclusive disks and shared disks.

7.3.1.2 Creating PVC

In the nKU main menu, select **Container Orchestration** > **PersistentVolumes management** > **PVC**. In the **PVC** screen, click **Create PVC** to bring up the **Create PVC** screen.

You can refer to the following example to enter the appropriate contents:

- **Name**: Set the name of the pvc. Naming rules: length limit of 1-50 characters, can only contain lowercase letters, numbers and separator (-), and must begin with a lowercase letter, number or lowercase letter end
- **Storageclass**: Select the storage class used to create the pvc

  Notes.

    - A storage class is a configuration template used in Kubernetes to dynamically create a pvc, defining the properties of the pvc, the creation policy, and the required storage plugins.
    - Storage classes are created and managed by the administrator. If there are no available storage classes, please contact the administrator.
- **Access Mode**: selects the access mode for the pvc. The following three access modes are supported:

# **ReadWriteOnce**: single node read/write support only

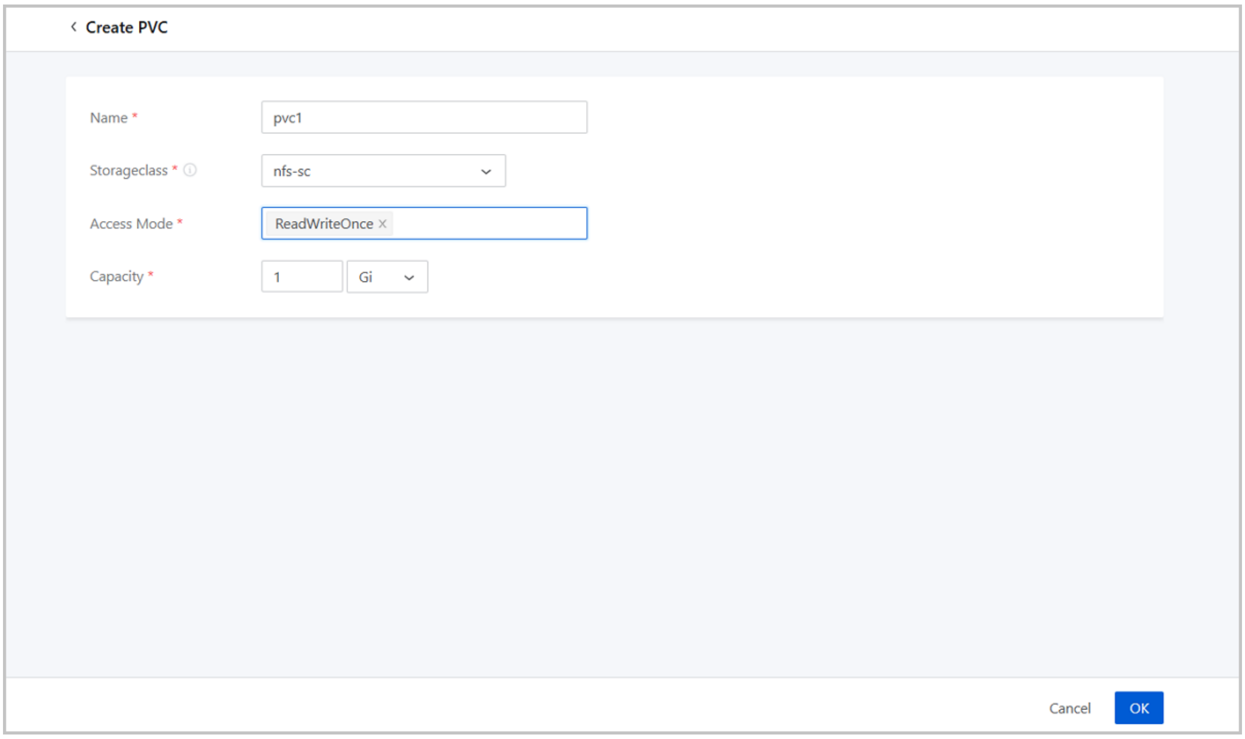
# **ReadWriteMany**: support for multiple node reads and writes

# **ReadOnlyMany**: support for multi-node read-only

- **Capacity**: Set the storage size of the pvc, unit: Gi, Ti

As shown in Figure 25: Create PVC:

Figure 25: Create PVC



Name * pvc1

Storageclass * ⓘ nfs-sc

Access Mode * ReadWriteOnce ×

Capacity * 1 Gi

7.3.1.3 PVC management

In the nKU main menu, click **Container Orchestration** > **PersistentVolumes management** > **PVC** to enter the **PVC** interface.

The pvc supports the following operations:

manipulate	descriptive
Create pvc	Create a new pvc.

Resize PVC	Expand the storage size of the pvc. **Notes.** • If the mounted persistentVolumeClaims have been mounted and the expanded storage size has not taken effect for a longer period of time, try restarting the mounted pod. • Expanding a pvc volume requires ensuring that the storage class used by the volume allows for expansion
Update Yaml	Updates to the ingress via Yaml.
PVC details	Go to the PVC Details page to view basic information, usage statistics, and events for the pvc.
Delete PVC	Remove the pvc from the cluster. **Notes.** - Deleting a pvc deletes both the PVC and the corresponding PV. - Deletion of mounted persistentVolumeClaims is not supported.

7.4 Configuration management

7.4.1 ConfigMap

7.4.1.1 General

Configmap stores configuration data in the form of key/value pairs, which can be mounted configMap and used as a configuration file in a workload. A configmap decouples the user's environment configuration information from the container configuration image, making it easy to modify the application configuration.

7.4.1.2 Create configmap

In the nKU main menu, select **Container Orchestration** > **Configuration Management** > **ConfigMap**. In the **ConfigMap** screen, click **Create ConfigMap** to bring up the **Create ConfigMap** screen.

You can refer to the following example to enter the appropriate contents:

- **Name**: Set the name of the configmap. Naming rules: length limit 2-50 characters, can only contain lowercase letters, numbers and separator (-), the separator can not be located at the beginning or end of the name

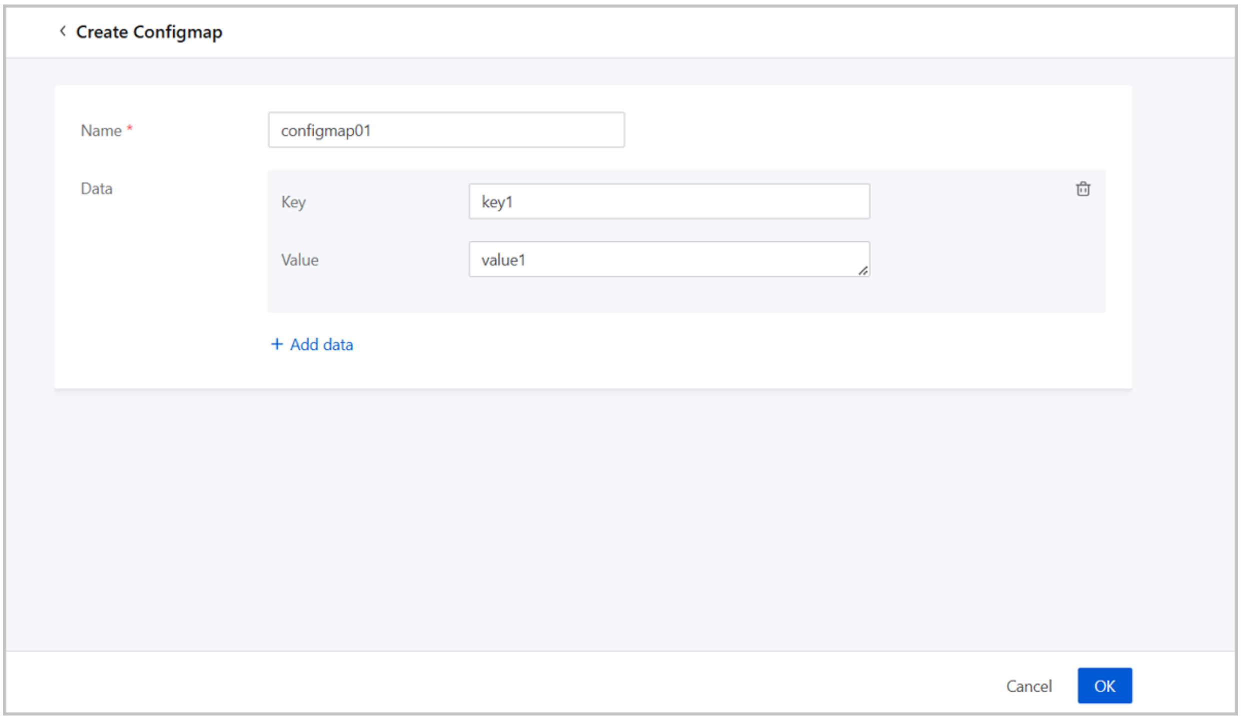
- **Data**: add configuration data in the form of key/value pairs, support for adding multiple key/value pairs

  # **key**: Fill in the key, the input should be within the range of 1-253 characters, and can only contain letters, numbers and the following special characters: (-)(.) (_). Cannot set duplicate key name

  # **Value:** Fill in the value, i.e. the content of the configuration, which can be multiple lines of text

As shown in Figure 26: Creating a configmap:

Figure 26: Creating a Configmap



< Create Configmap

Name * configmap01

Data

Key key1

Value value1

7.4.1.3 Managing configmaps

From the nKU main menu, select **Container Orchestration** > **Configuration Management** > **ConfigMap** to access the **ConfigMap** screen.

The configmap supports the following operations:

manipulate	**descriptive**
Create Configmap	Create a new configmap.

Edit configmap	Modify the configmap data. **Note:** Configmap names do not support modification.
Update Yaml	Updates to the configmap are made by way of Yaml.
Delete configmap	Remove the configmap from the cluster. **Note:** If the configuration configuration is already mounted into a container pod, the restart operation for the mounted container pod will fail after deletion, so please be careful.

7.4.2 Secret

7.4.2.1 General

Secret store sensitive configurations in a Kubernetes cluster, such as passwords, certificates, and so on, and can be used as files or envs in workloads. Because Secret can be independent of the container pods that use them there is less risk of exposing them and their data during the creation, viewing, and editing of container pods.

7.4.2.2 Creating a Secret

In the nKU main menu, select **Container Orchestration** > **Configuration Management** > **Secret**. In the **Secret** screen, click **Create Secret** to bring up the **Create Secret** screen.

Creating a Secret is divided into the following 3 scenarios:

- Creating a Opaque Secret
- Creating a TLS Secret
- Create rigestry login Secret

Creating a Opaque Secret

You can refer to the following example to enter the appropriate contents:

- **Name**: Set the name of the Secret. Naming rules: length limit 2-50 characters, can only contain lowercase letters, numbers and separator (-), separator can not be located at the beginning or end of the name
- **Type**: Select **Opaque**. The Opaque type of Secret is commonly used to store sensitive configurations, such as account passwords.

- **Data**: Configuration data is added in the form of key/value pairs, and values in the data are base64 encoded when the Secret is created.

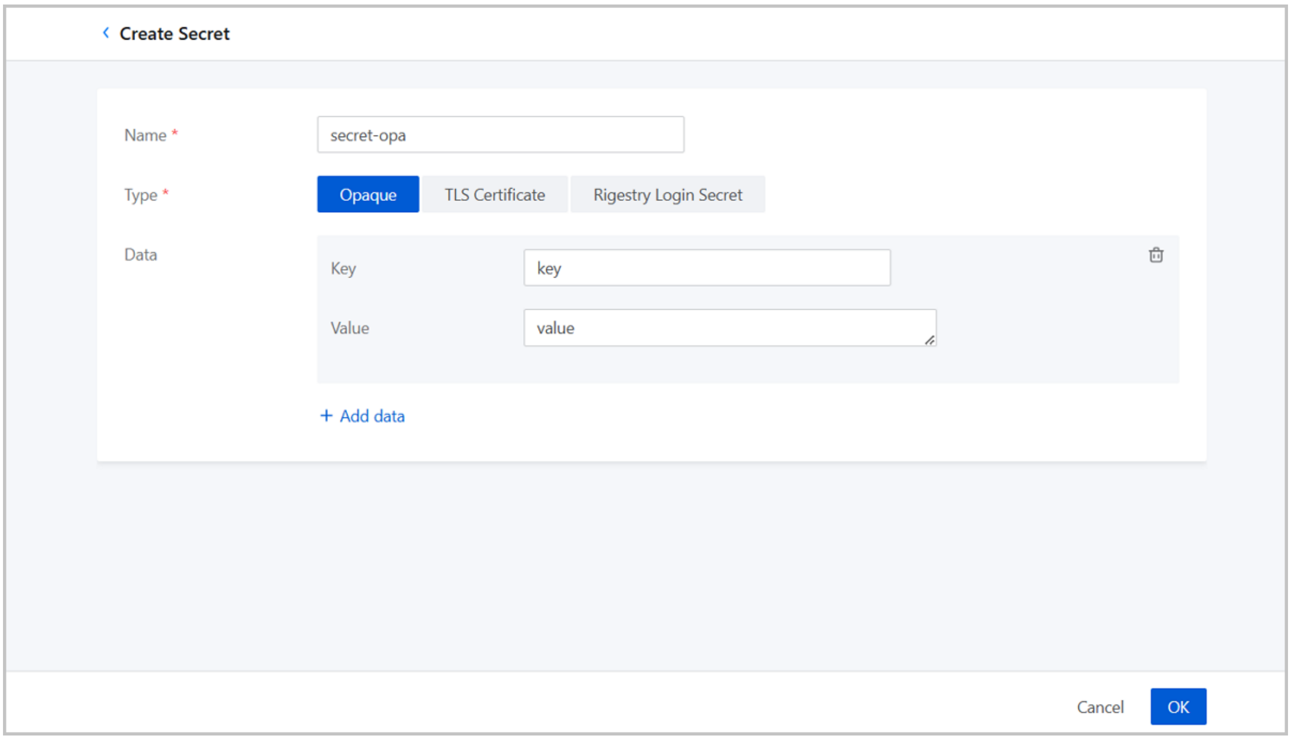
  # **key**: Fill in the key, the input should be within the range of 1-253 characters, and can only contain letters, numbers and the following special characters: (-)(.) (_). Cannot set duplicate key name

  # **Value**: Fill in the value, i.e. the configuration

content, which can be multi-line text.

As shown in Figure 27: Creating a Opaque Secret:

Figure 27: Creating a Opaque Secret



Create Secret

Name * secret-opa

Type * Opaque | TLS Certificate | Rigestry Login Secret

Data

Key: key

Value: value

Creating a TLS certificate Secret

You can refer to the following example to enter the appropriate contents:

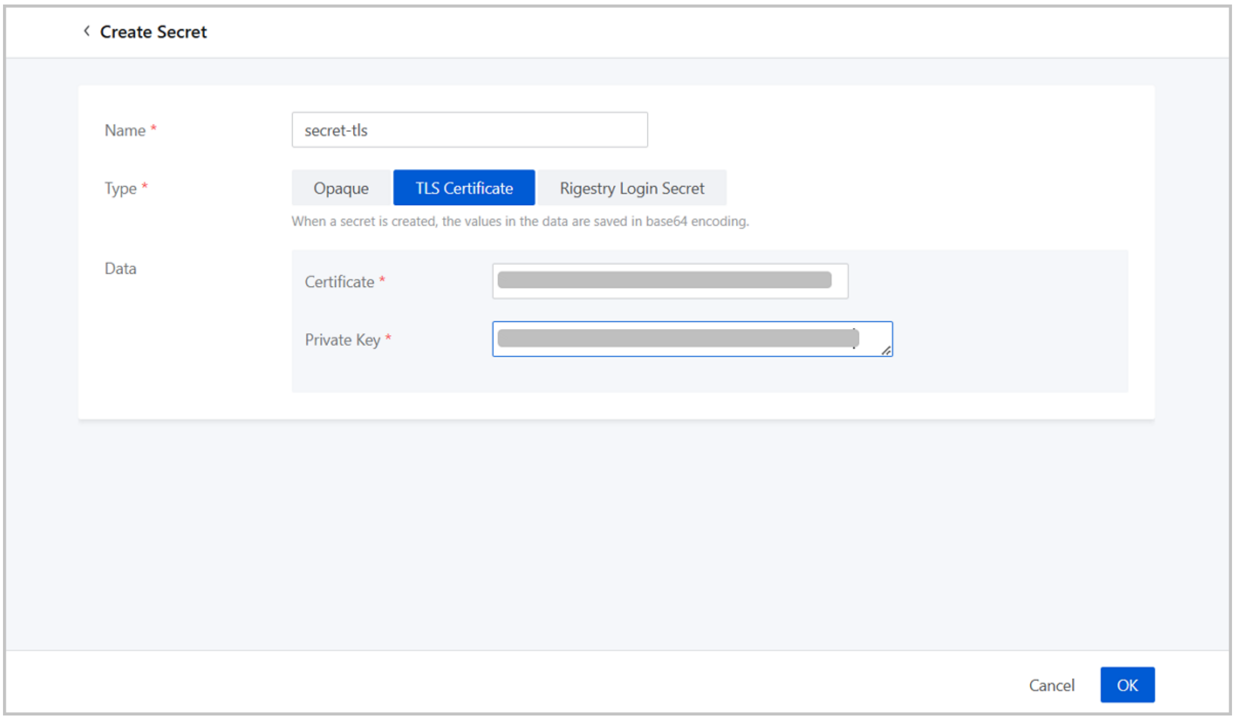
- **Name**: Set the name of the Secret. Naming rules: length limit 2-50 characters, can only contain lowercase letters, numbers and separator (-), separator can not be located at the beginning or end of the name
- **Type**: select TLS certificate, the Secret of TLS certificate type is mainly used to store TLS certificate
- **Data**: add configuration data in the form of certificates/private keys, the values in the data will be base64 encoded and saved when creating the Secret.

# **Certificates**: Fill in the TLS certificate file

# **private key**: fill in the TLS private key file

As shown in Figure 28: Creating a TLS certificate Secret:

Figure 28: Creating a TLS certificate Secret



‹ Create Secret

Name * secret-tls

Type * Opaque | TLS Certificate | Rigestry Login Secret

When a secret is created, the values in the data are saved in base64 encoding.

Data

Certificate *

Private Key *

Cancel OK

Create Rigestry login Secret

You can refer to the following example to enter the appropriate

contents:

- **Name**: Set the name of the Secret. Naming rules: length limit 2-50 characters, can only contain lowercase letters, numbers and separator (-), separator can not be located at the beginning or end of the name
- **Type**: Select the image repository login secret. The image repository login secret is used to store the authentication information required to log in to the image repository or pull images from the private image repository.
- **Data**: Add configuration data

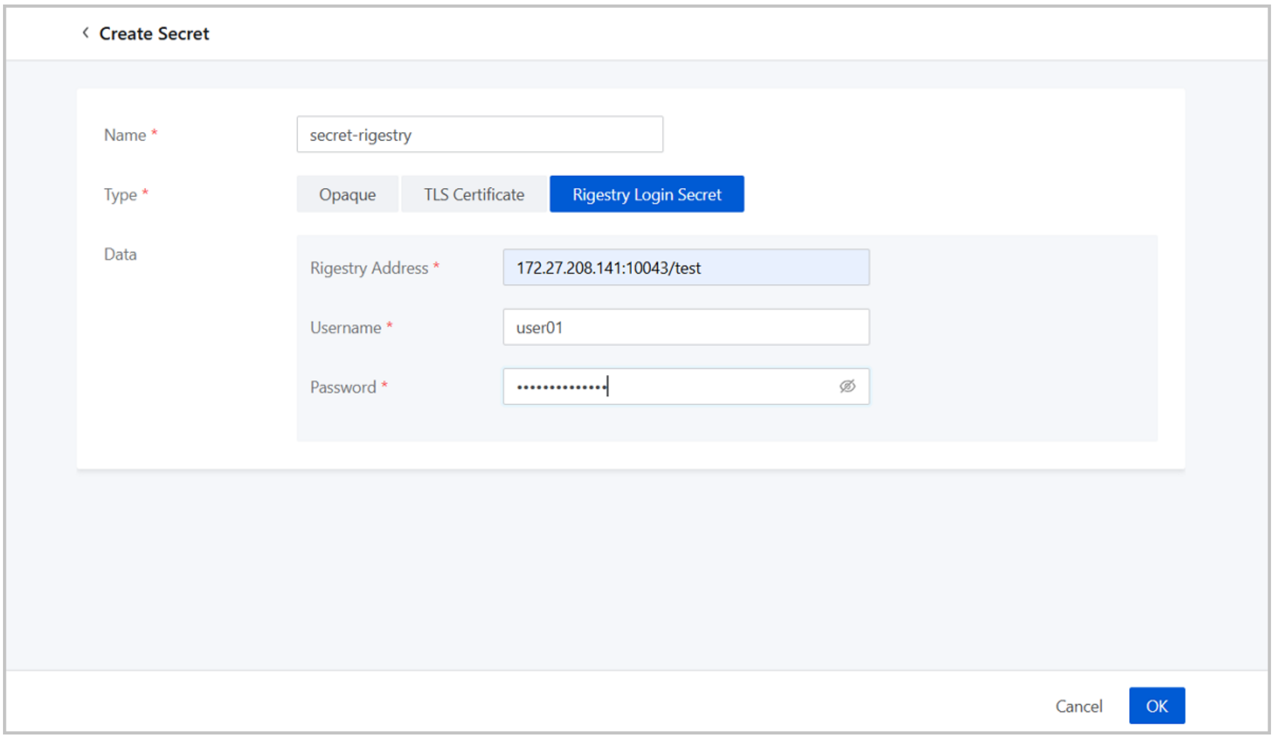
# **Rigestry address**: fill in the address of the image rigestry

# **Username**: Fill in the username used to log in to the image rigestry.

# **Password**: Fill in the password that corresponds to the username used to log in to the image rigestry.

As shown in Figure 29: Create Rigestry Login Secret:

Figure 29: Create Rigestry Login Secret



‹ Create Secret

Name * secret-rigestry

Type * Opaque | TLS Certificate | Rigestry Login Secret

Data

Rigestry Address * 172.27.208.141:10043/test

Username * user01

Password * ••••••••••••••

7.4.2.3 Managing the Secret

From the nKU main menu, select **Container Orchestration** > **Configuration Management** > **Secret** to access the **Secret** screen.

The Secret supports the following operations:

manipulate	**descriptive**
Create Secret	Create a new Secret.
Edit Secret	Modify Secret data. **Notes.** •The name of the Secret does not support modification. •Currently, only three types of Secret are supported for modifying Opaque, TLS certificate, and image repository login secret.

Update Yaml	Updates to Secret are made by means of Yaml, and all types of Secret support updates.
Delete Secret	Remove the Secret from the cluster. **Note:** If the Secret has been mounted into container pods, the restart operation for the mounted container pods will fail after deletion.

7.5 Microservices governance

7.5.1 Microservices application

7.5.1.1 General

Microservices application is a collection of resource objects such as a set of workloads, services, service mesh resources, etc. It supports accessing microservices applications to the platform for visualization governance. It supports creating new or selecting existing workloads and services to create microservices applications to support traffic governance, canary release, trace tracking, and topology viewing of the applications.

Note: To use the microservices governance feature in a cluster, you need to deploy the microservices governance component in the cluster first, you can either enable the microservices governance component when you create the cluster, or go to the feature component list on the cluster details page to deploy it.

7.5.1.2 Creating a microservices application

In the nKU main menu, select **Container Orchestration** > **Microservices governance** > **Microservices application**. In the **Microservices application** screen, click **Create Microservices Application** to bring up the **Create Microservices Application** screen.

You can refer to the following example to enter the appropriate contents:

- **Name**: Set the name of the microservices application. Naming rules: 1-50 characters in length, can only contain lowercase letters, numbers and separator (-), and must start with a lowercase letter and end with a number or lowercase letter.
- **Description**: optional, note information related to microservices application
- **Workload**: Specify the workload for the microservices application. Supports selecting an existing workload or creating a new one

  # Select Existing: Directly select an existing workload.

Note: Only one microservices application can be related to the same workload.

# Create : Create a new workload related to the microservices application, refer to Creating a workload

- **Workload Version**: Set the workload version. Version rules: length limit 1-50 characters, can only contain lowercase letters, numbers and the separator "-", "-" can not be used at the beginning or the end

  Note: If the workload already has a label zsedge.io/version, the value of this label will be used as the workload version by default, manual modification is supported and the modified value will be overwritten to the workload zsedge.io/version.

- **Service**: Specifies the service for the workload. Support for selecting existing services or creating new ones

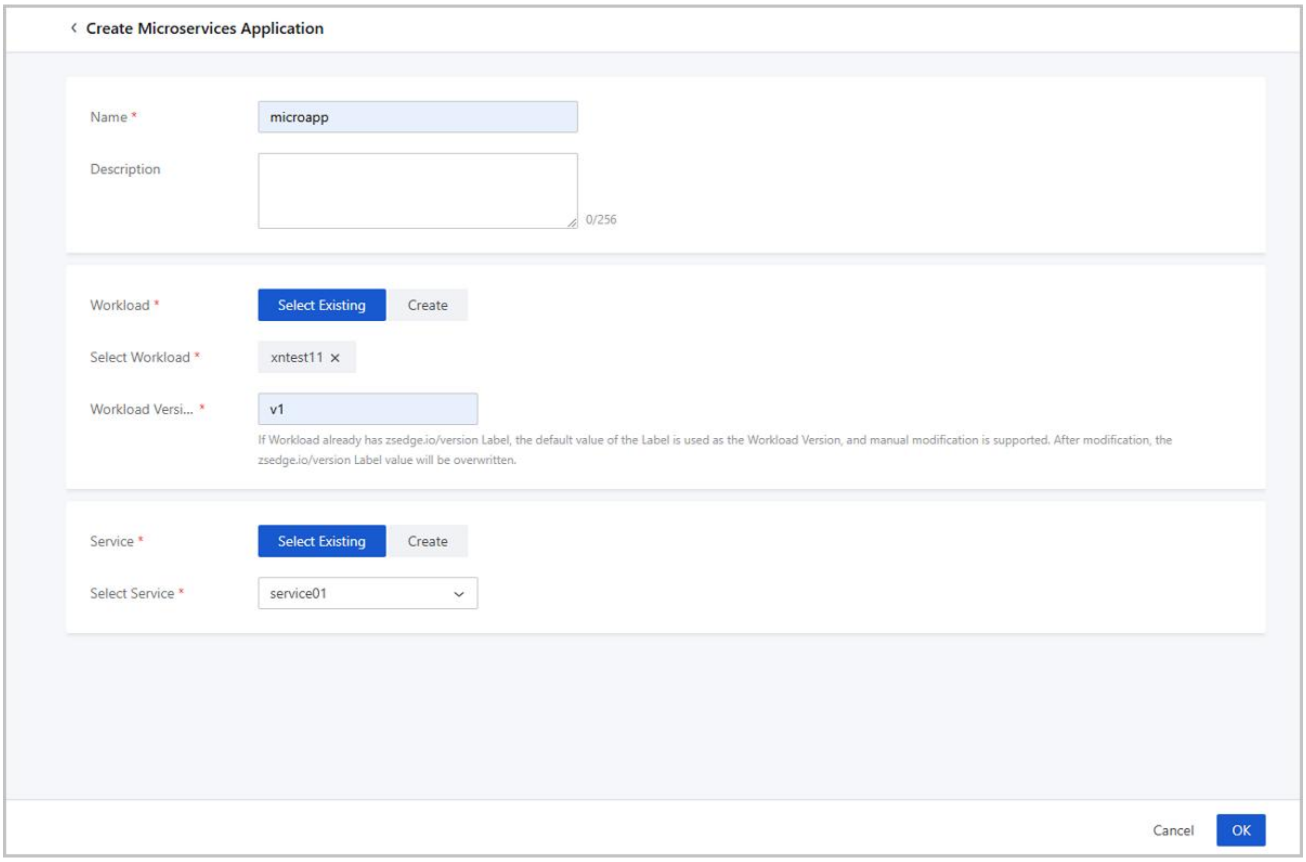
  # Select Existing: Directly select an existing service.

  Note: Existing services must be related services to the workload of the microservices application.

  # Create: Creates a new service. Refer to Creating a Service

As shown in Creating a microservices application:

Figure 30: Creating a Microservices Application



Name * microapp

Description 0/256

Workload * Select Existing

Select Workload * xntest11 ×

Workload Versi... * v1

If Workload already has zsedge.io/version Label, the default value of the Label is used as the Workload Version, and manual modification is supported. After modification, the zsedge.io/version Label value will be overwritten.

Service * Select Existing

Select Service * service01

7.5.1.3 Managing microservices applications

In the nKU main menu, click **Container Orchestration** > **Microservices Governance** > **Microservices Application** to access the **Microservices Application** screen.

The microservices application supports the following operations:

manipulate	descriptive
Create microservices applications	Create a new microservices application.
canary release	Create a gray release to update the workload under the microservices application by means of a canary release. **Notes.** **1.** Canary releases are not supported in the following cases: • There are no workloads under the microservices application. • There are no services under microservices application.

	• The service mesh resources for microservices applications had been modified through non-formal forms. • The microservices application already has a canary release job installed. • When services under the microservices application are associated with workloads via the zsedge.io/version label, the traffic policy for grayscale release will not take effect accurately, and you need to update the service Yaml to remove the zsedge.io/version label from spec.selector before you can perform grayscale release. **2.** Visual viewing and managing canary releases is not supported in the following cases: • The service mesh resources of the microservices application have been modified through non-formal forms or there are no service mesh resources under the microservices application • No workloads under microservices applications
Delete microservices applications	Delete the microservices application. **Note:** Deleting a microservices application will synchronize the deletion of service mesh resources under the microservices application.

7.5.1.4 Canary release

background information

Gray-scale release is a software deployment strategy designed to securely and gradually install new versions into production environments.nKU supports the visual operation of the complete grayscale release process, including: creating a grayscale release job, configuring traffic policy, monitoring the grayscale release process in real time, and taking the version down to achieve a smooth transition of version updates and risk control. This section mainly introduces the whole process of canary release.

The entire process of installing a canary release is divided into the following four steps:

1. Creating a canary release job
2. Monitor canary releases
3. Adjusting traffic policy

4. Take over all traffic and take down the version

procedure

1. Creating canary release jobs

In the nKU main menu, click **Container Orchestration** > **Microservices governance** > **Microservices application**. In the **Microservices application** interface, select the microservices application to be canary ed and click **Canary Release** to bring up **the Canary Release** interface.

Creating a canary release job is divided into the following two steps:

a) New version of workloads

Creating a new version of a workload supports Quick Create, Standard Create, and Select Existing:

i. Quick creation

Only set the name, replicas, and image configuration of the workload, and keep the rest of the configuration the same as the old workload.

You can refer to the following example to enter the appropriate contents:

- **Workload Name**: Specifies the name of the new version of the workload. Name rules: length limit of 2-50 characters and can only contain lowercase letters, numbers, and the separator "-", must start with a lowercase letter and end with a lowercase letter or number
- **Workload Version**: Specifies the version of the workload for the new release. Versioning rules: length limit 1-50 characters, can only contain lowercase letters, numbers and the separator "-", "-" can not be used at the beginning or the end
- **Replicas**: Sets the number of Pod the workload contains, up to a maximum of 100 replicas are supported. workloads of type DeamonSet do not need to set the number of replicas
- **Image configuration**: Configure the image of the container, the default settings are the same as the old workloads

# **Container**: container name, consistent with older workloads, not modifiable

# **Image Source**: Select the container image repository, which supports both local and external repositories.

If you select **repository**, the container will be created using the image repository that has been uploaded to the nKU Repository , please set the following parameters:

- **Image**: Select container image
- **Tag**: Select a Tag

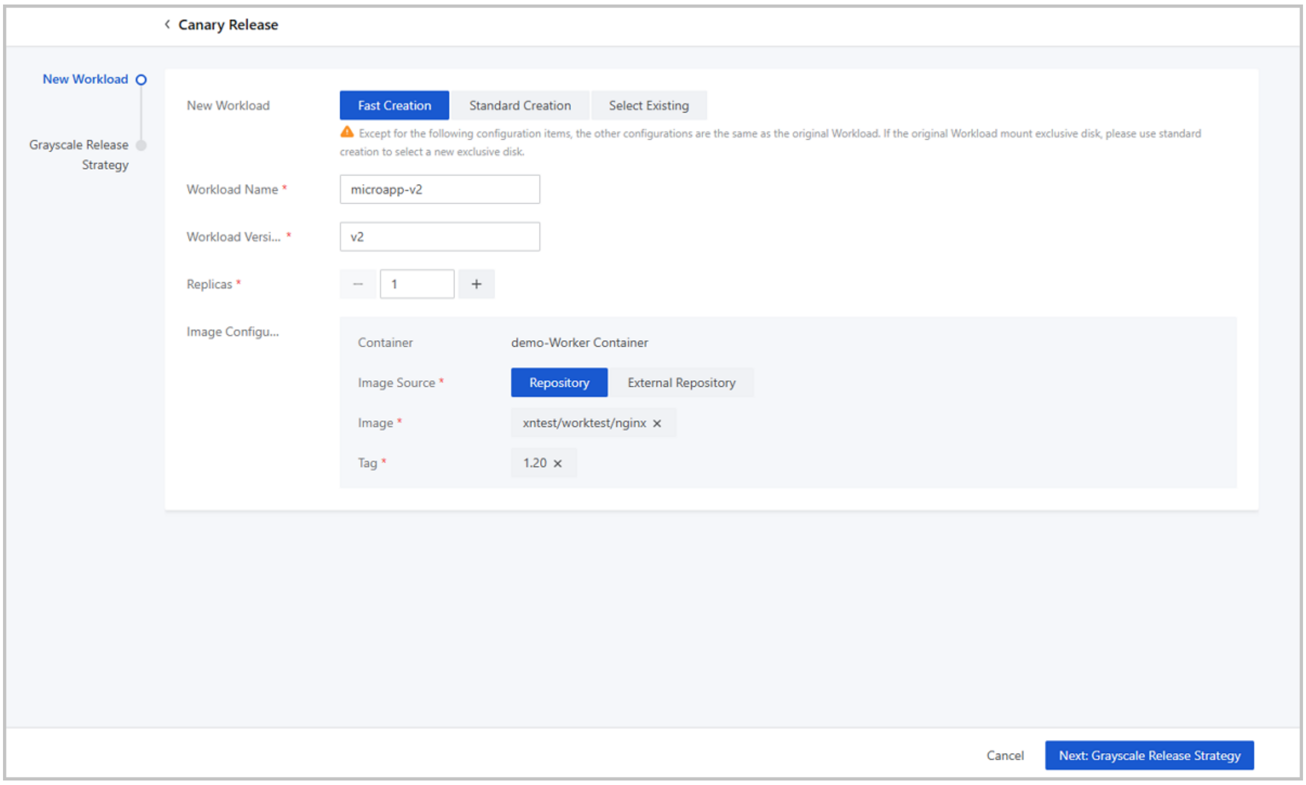
If you select **external repository**, which will pull the image repository from outside nKU to create the container, set the following parameters:

- **Image**: enter the detailed Image address, e.g. docker.io/library/nginx:latest
- **Image pulling Secret**: optional, if the external repository to which the image belongs is a private repository, you need to select the Secret to be used for the image repository authentication

Note: The Quick Create method maintains the same configuration as the old workload except for the name, replicas, and image configuration of the workload. If the mount persistentVolumeClaims of the old workload do not support multi-node reads or writes, use the Standard Create to select a new pvc.

As shown in Figure 31: Canary release-Quickly create a new version of the workload:

Figure 31: Canary release-quickly creating a new version of a workload



‹ Canary Release

New Workload: Fast Creation | Standard Creation | Select Existing

⚠ Except for the following configuration items, the other configurations are the same as the original Workload. If the original Workload mount exclusive disk, please use standard creation to select a new exclusive disk.

Workload Name *: microapp-v2

Workload Versi... *: v2

Replicas *: − 1 +

Image Configu...

Container: demo-Worker Container

Image Source *: Repository | External Repository

Image *: xntest/worktest/nginx ×

Tag *: 1.20 ×

ii. Standard creation

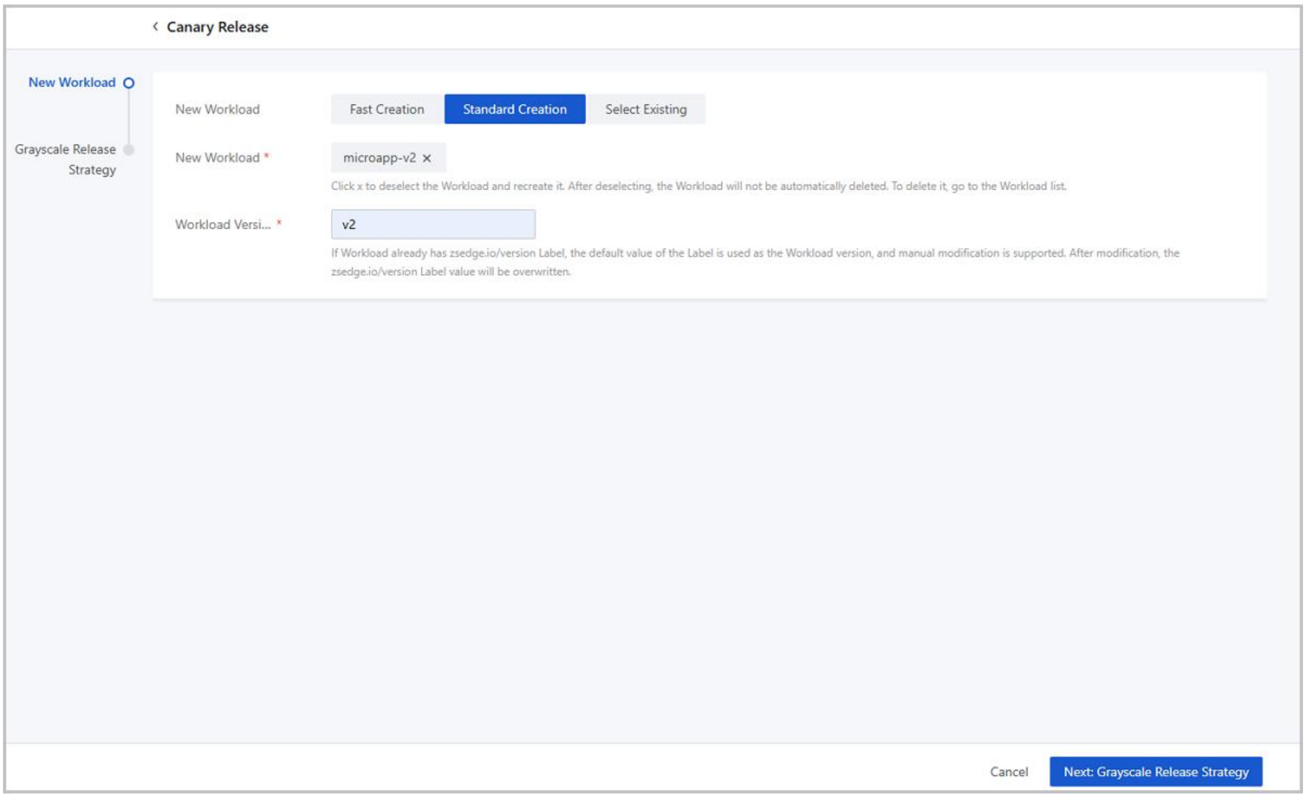
You can refer to the following example to enter the appropriate contents:

- **New Workload**: Create a new workload, refer to Creating a Workload
- **Workload Version**: Specify the version of the workload for the new release, version rules: length limit 1-50 characters, can only contain lowercase letters, numbers and separator "-", "-" can not be used in the beginning or end

Note: If the workload already has a label zsedge.io/version, the value of this label will be used as the workload version by default, manual modification is supported and the modified value will be overwritten to the workload zsedge.io/version.

As shown in Figure 32: Canary release-Standard Create New Version workload:

Figure 32: Canary release-standard creation of a new version of a workload



New Workload: Standard Creation

New Workload *: microapp-v2 ×

Click x to deselect the Workload and recreate it. After deselecting, the Workload will not be automatically deleted. To delete it, go to the Workload list.

Workload Versi... *: v2

If Workload already has zsedge.io/version Label, the default value of the Label is used as the Workload version, and manual modification is supported. After modification, the zsedge.io/version Label value will be overwritten.

iii. select existing

You can refer to the following example to enter the appropriate contents:

- **Select workload**: If the new version of the workload has already been created in another way, you can select it directly

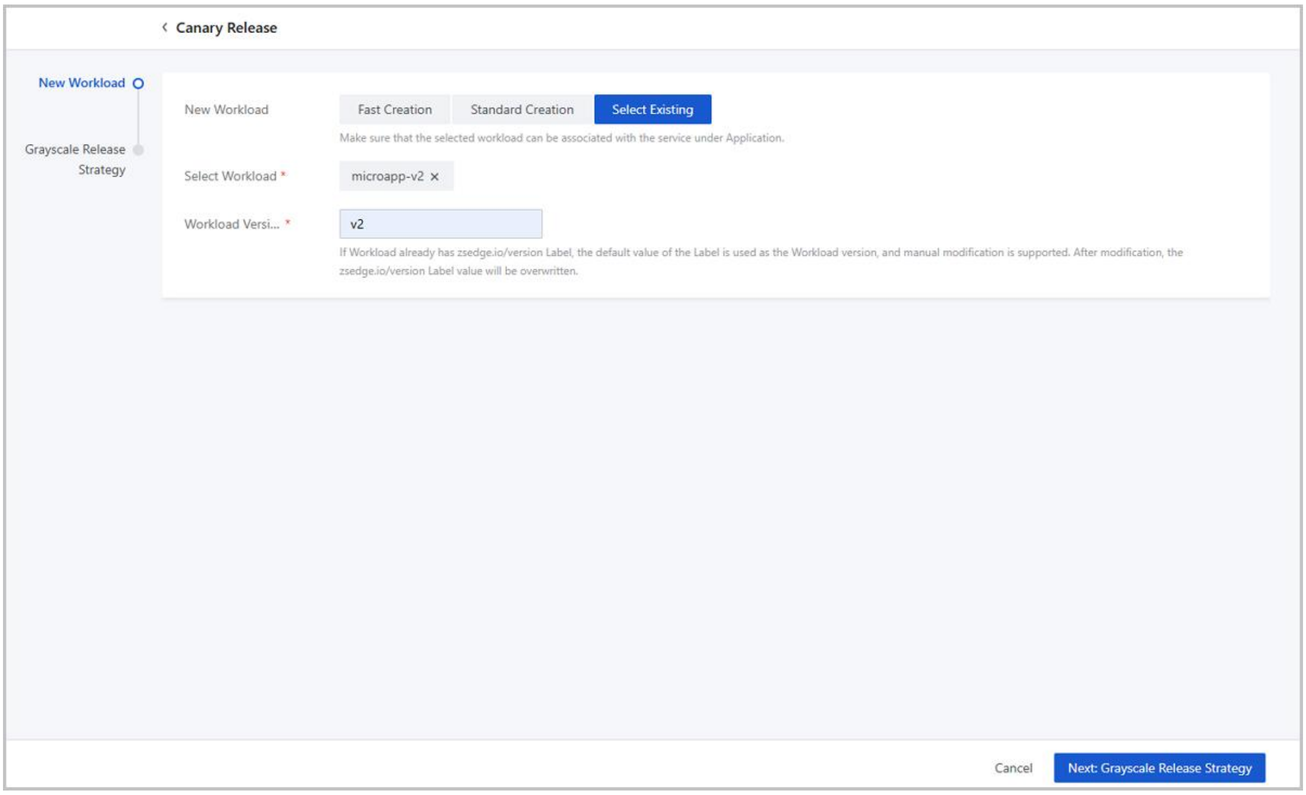
  Note: You need to ensure that the selected workload can be related services under the application.

- **Workload Version**: Specify the version of the workload for the new release, version rules: length limit 1-50 characters, can only contain lowercase letters, numbers and separator "-", "-" can not be used in the beginning or end

  Note: If the workload already has a label zsedge.io/version, the value of this label will be used as the workload version by default, manual modification is supported and the modified value will be overwritten to the workload zsedge.io/version.

As shown in Figure 33: Canary release-Select existing workload:

Figure 33: Canary release - selecting existing workloads



New Workload: Select Existing

Make sure that the selected workload can be associated with the service under Application.

Select Workload *: microapp-v2 ×

Workload Versi... *: v2

If Workload already has zsedge.io/version Label, the default value of the Label is used as the Workload version, and manual modification is supported. After modification, the zsedge.io/version Label value will be overwritten.

b) The grayscale release strategy can

refer to the following example to enter

the appropriate content:

- **Policy Type**: Select the type of traffic policy, support both trafic weight and Request Content to allocate traffic to the old and new versions.
  If you choose to **trafic weight**, the access traffic will be sent to the new version of the workload in the configured proportion and the remaining proportion of the traffic will be sent to the old version of the workload, set the following parameters:
  - **New version trafic weight**: Set the weight of the trafic to be sent to the new version, default is 50%, maximum is 100%.

  If **Request Content** is selected, traffic that matches the following Request Content configuration will be sent to the new version of the workload, and other traffic will be sent to the old version of the workload. set the following parameters:
  - **Request Content**: Set the Request Content requirements to be sent to the new version of the workload, only requests that meet this requirement will be sent to the new version of the workload, other requests will be sent to the

old version of the workload. Users can fill in URI, Cookie, Header and OS requirements at the same time, or only one or several of them.

# **URI**: set the URI matching method and content to be matched, that is, the URI of the request traffic access should be able to match the filled content according to the selected matching method.

# **Cookie**: set the cookie matching method and the content to be matched, that is, you need to be able to select the matching method in the request traffic in the cookie to match the content filled in.

# **Header**: set the matching method of the Header and the content to be matched, i.e., you need to be able to match the requested content in the Header of the request traffic according to the selected matching method.

# **OS**: Specifies which operating systems request traffic from can be sent to the new version, defaults to all operating systems, can specify Windows, Linux, Mac OS, Android, iOS

Note: The following three matching methods are supported:

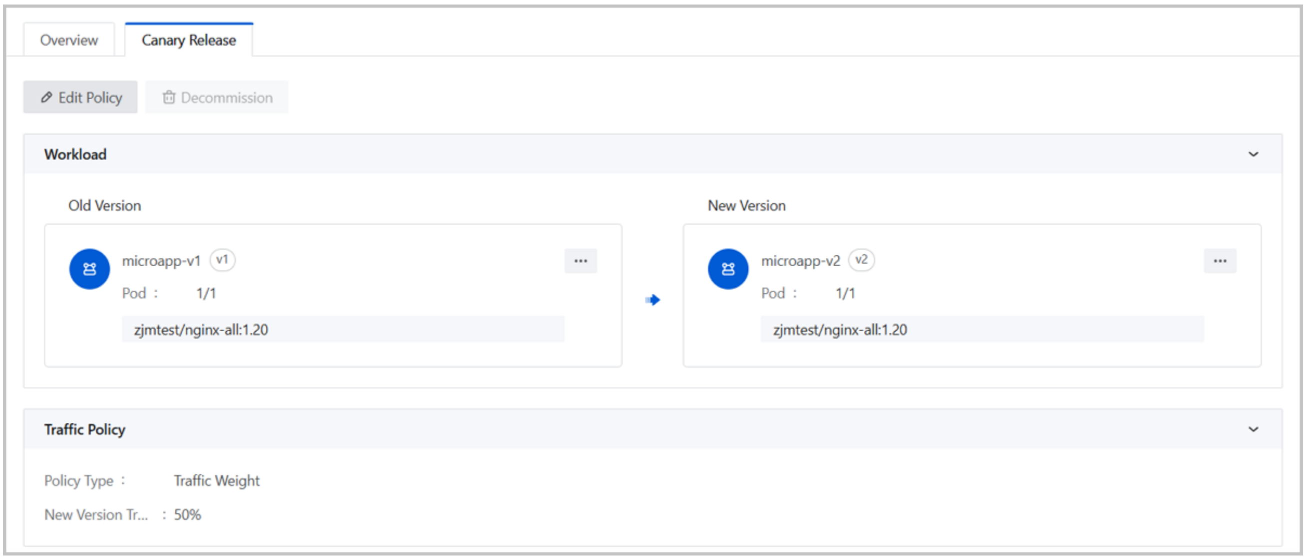
- Exact Match: The Request URI/Cookie/Header needs to be an exact match to the specified Content in order to be sent to the new version.
- Prefix Matching: The request URI/Cookie/Header needs to match the specified prefix requirements in order to be sent to the new version.
- Regex Matching: The request URI/Cookie/Header needs to match the specified regular expression in order to be sent to the new version.

2. Monitor canary releases

After you create a gray release job, in the nKU main menu, click **Container Orchestration** > **Microservices governance** > **Microservices application**. In the **Microservices application** interface, click the microservices application that has been installed for gray release, and go to the gray release page on the Microservices application details page, where you can view the details of the gray release job and the traffic monitoring between the old and new versions.

As shown in Figure 34: Canary release details:

Figure 34: Canary Release Details



Workload

Old Version

microapp-v1 v1

Pod : 1/1

zjmtest/nginx-all:1.20

New Version

microapp-v2 v2

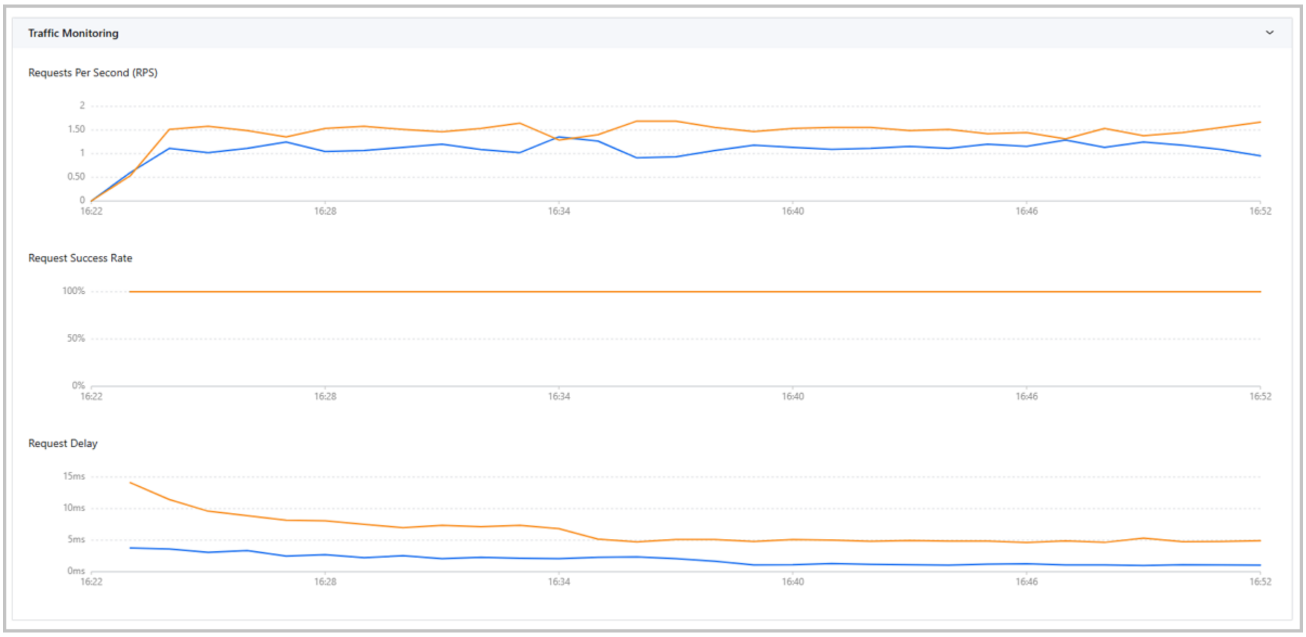
Pod : 1/1

zjmtest/nginx-all:1.20

Traffic Policy

Policy Type : Traffic Weight

New Version Tr... : 50%



3. Adjusting traffic policy

   In the nKU main menu, click **Container Orchestration** > **Microservices governance** > **Microservices application**. In the **microservices application** interface, click the microservices application that has been installed for Canary release, go to the Canary release page of the microservices application detail page, and click **Edit Policy** to bring up the **Edit Policy** interface. You can adjust the traffic policy according to the actual situation of the canary release by referring to the step of creating a canary release.

4. Taking over all traffic and versioning it down

   When you are sure that the new version is functionally stable, you can let the new version take over all traffic, or have the old version take over all traffic back if there are problems with the new version. In the nKU main menu, click **Container Orchestration** > **Microservices governance** > **Microservices application**. In the **microservices application** interface, click the microservices application that has been installed for canary release, go to the Canary release page of the microservices application details page, select the version that needs to take over all traffic, and click **Take over all traffic**, then all the request traffic will be sent to the version that takes over the traffic. At this point, you can click **Version Take Down** to take down the version that has no traffic access and remove the workload corresponding to the version.

7.5.2 Service topology

7.5.2.1 General

Service topology visualizes the invocations, dependencies and traffic monitoring data between microservices applications under the current namespace. Supports visualization to view historical and real-time topology diagrams, and supports configuration of load balancing, connection pooling, Circuit Breaking and other traffic management rules for microservices applications.

Note: To use the microservices governance feature in a cluster, you need to deploy the microservices governance component in the cluster first, you can either enable the microservices governance component when you create the cluster, or go to the cluster details page to deploy the feature component list.

7.5.2.2 Viewing the topology map

From the nKU main menu, select **Container Orchestration** > **Microservices Governance** > **Service Topology** to access the **Service Topology** screen.

The service topology by default displays a topology map of the last 1 minute of microservices applications under the selected namespace for the selected cluster, which supports the following:

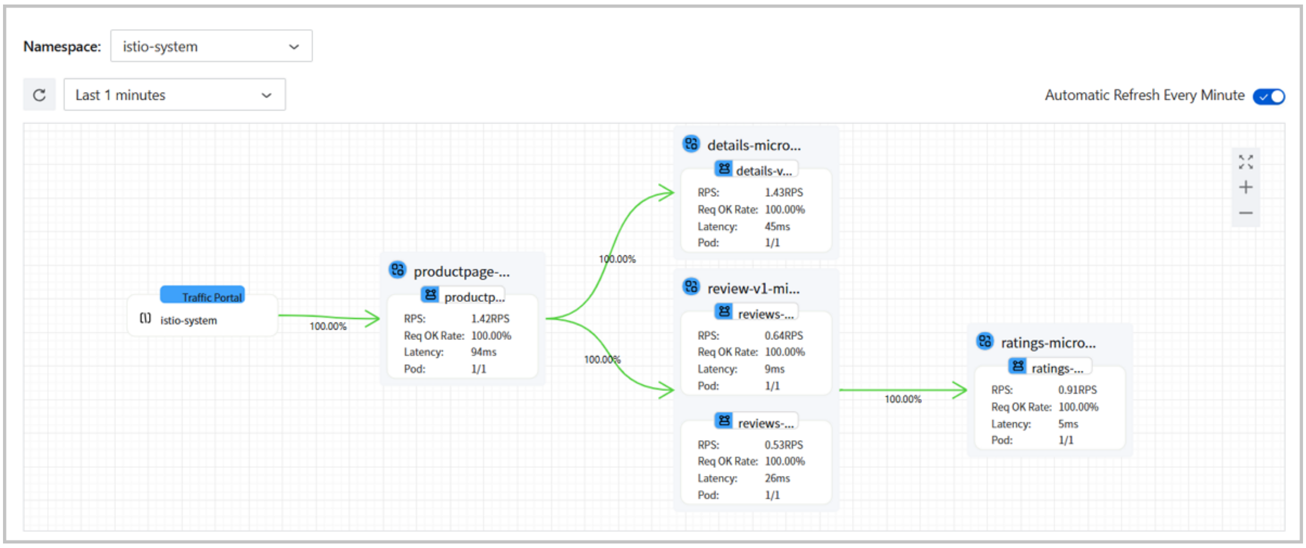
- Demonstrate the number of requests per second, request success rate, request latency, and number of worker containers in the workload
- Automatically refreshes every minute to get the latest data and display it in real time
- Supports selecting a time period to view the topology for the selected time period

Note: If there is no traffic access between the microservices applications during the selected time period, the topology graph shows only the nodes with no connectivity.

- The node connectivity includes green, yellow and red colors to visualize the request success rate. The color thresholds for request success rate are: green (greater than or equal to 99.9%), yellow (greater than or equal to 80% and less than 99.9%), and red (less than 80%).
- Support zoom in, zoom out, full screen view

As shown in Figure 35: Service Topology:

Figure 35: Service topology



Namespace: istio-system

Last 1 minutes

Automatic Refresh Every Minute

7.5.2.3 Traffic management

In the nKU main menu, select **Container Orchestration** > **Microservices Governance** > **Service Topology** to enter the **service topology** interface. Click the microservices application or workload on the topology map that you want to perform traffic governance to bring up the **traffic governance** interface.

You can refer to the following example to enter the appropriate contents:

- **Load Balancer**: Select whether to configure load balancing parameters, the default is off. If it is on, you need to set the following parameters:

# **Load balancing policy**: select a load balancing policy, support for least requests, polling, random, session hold four policies, the default is least requests

Notes.

- Least Requests: sends access traffic to the container pod with the least current requests.
- Polling: sends access traffic to individual container pods on a rotating basis.
- Random: sends access traffic randomly to any container pod.
- Session persistence algorithm: Based on the hash algorithm, traffic that meets the conditions is assigned to the same pod to ensure session continuity and consistency.

If you select **session hold**, the traffic that meets the conditions will be assigned to the same container pod based on the hash algorithm to ensure session continuity and consistency, please set the following parameters:

- **Session persistence algorithm**: set the session persistence algorithm, support for three algorithms based on HTTPHeaderName, HTTPCookie, UseSourceIP, the default is based on HTTPHeaderName.

  If you choose **HttpHeaderName**, the session is maintained based on the information in the Header, and requests with consistent values for the keys specified in the Header are sent to the same container pod, set the following parameters:

    - **HTTP Header key**: sets the name of the specified Header key to be used for HTTP Header hashing

  If you select **HttpCookie**, which maintains the session based on the cookie information and sends requests with consistent information specified in the cookie to the same pod, set the following parameters:

    - **Cookie name**: sets the cookie name to be used for HTTP cookie hashing
    - **Cookie path**: optional, set the cookie path to be used for HTTP cookie hashing
    - **Cookie valid time**: optional, set the cookie valid time to be used for HTTP cookie hashing, in seconds.

  If you choose to **UseSourceIp**, the session is maintained based on the HTTP source IP address information, and requests from the same source IP address are sent to the same pod

- **Connection Pooling**: Select whether to configure connection pooling parameters, the default is off. If it is on, you need to set the following parameters:

# **max pending requests**: set the max requests that the connection pool will wait for, can only be positive integers

# **Max Requests**: set the maximum requests allowed by the connection pool, can only be positive integers

# **Max requests per connection**: set the maximum requests per connection, can only be positive integers

Note: If max requests per connection is set to 1, session hold is disabled.

# **Maximum retries**: set the maximum number of retries, can only be a positive integer

# **Connection timeout**: set the connection timeout, in seconds, can only be a positive integer

- **Circuit Breaking**: Select whether to turn on the Circuit Breaking, the default is off. If it is turned on, the following parameters need to be set:

Note: Circuit Breaking is a mechanism to passively keep the HTTP service healthy. When a request sent to a pod returns 5xx response errors for a specified number of times, the pod will be fused and no client requests will be sent to the pod during the fusion.

# **Consecutive 5xx errors**: Specify how many consecutive 5xx error responses will fuse the pod, can only be a positive integer.

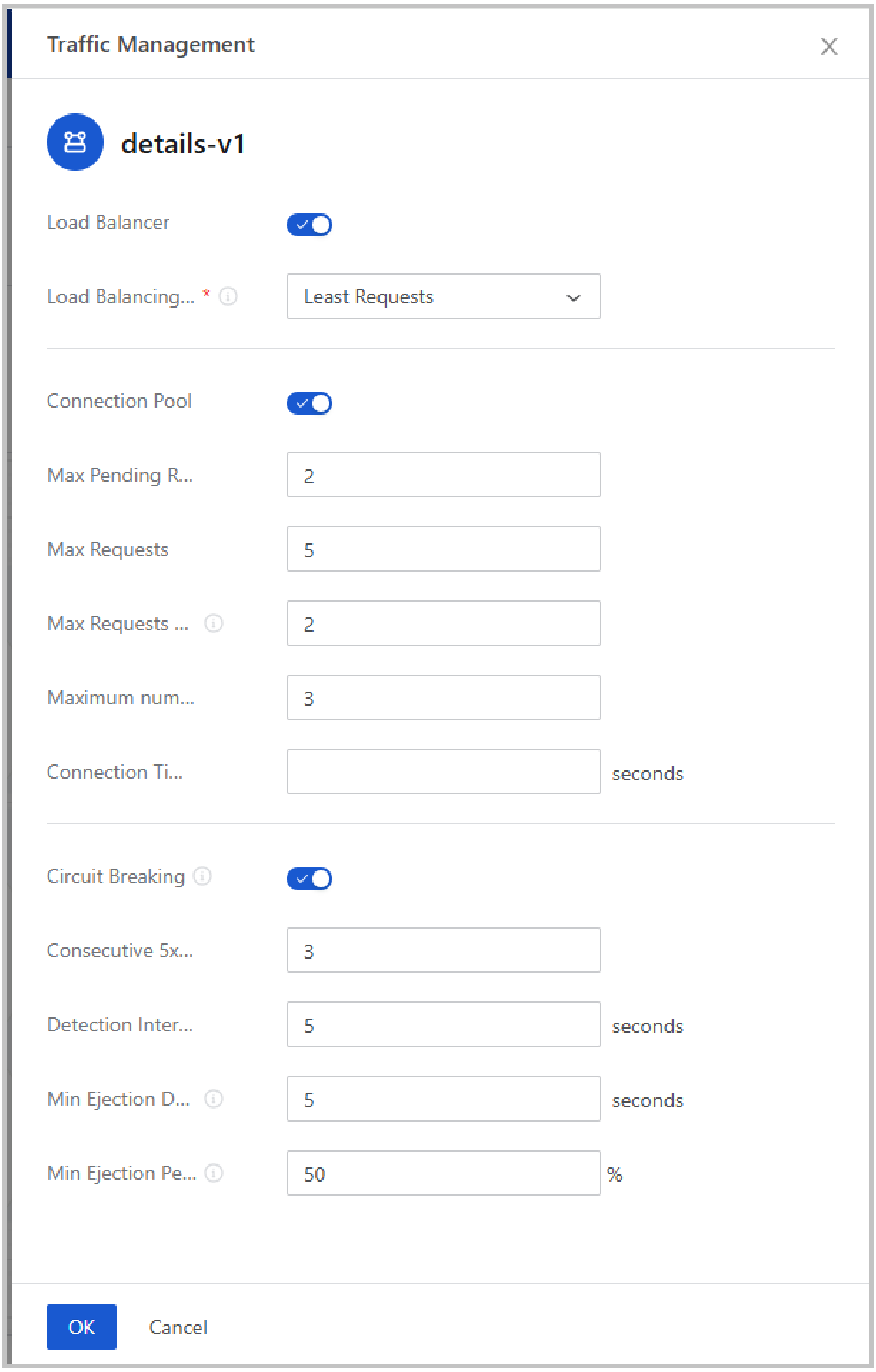
# **Detection interval**: set the detection interval in seconds, which can only be a positive integer.

# **Min ejection duration**: set the min ejection duration in seconds, can only be a positive integer. Pod fusing duration = min ejection duration * number of times this pod has been fused

# **Maximum Fusing Percentage**: Set the maximum min ejection percentage, which can only be an integer from 1 to 100. Number of fusible container pods = Total number of container pods * min ejection percent. When the result is not an integer, it will be automatically rounded down.

As shown in Figure 36: Traffic Management:

Figure 36: Traffic Management



Traffic Management

details-v1

Load Balancer: On

Load Balancing... *: Least Requests

Connection Pool: On

Max Pending R...: 2

Max Requests: 5

Max Requests ...: 2

Maximum num...: 3

Connection Ti...: seconds

Circuit Breaking: On

Consecutive 5x...: 3

Detection Inter...: 5 seconds

Min Ejection D...: 5 seconds

Min Ejection Pe...: 50 %

7.5.3 Trace tracking

7.5.3.1 General

Trace tracking integration Jager, provides call chain query between services, all-round monitoring of the call status of the services in the call chain, call time consumed and other key indicators, to help users understand the response of the nodes in the business request chain, in order to quickly locate the problem and the module where the problem occurs.

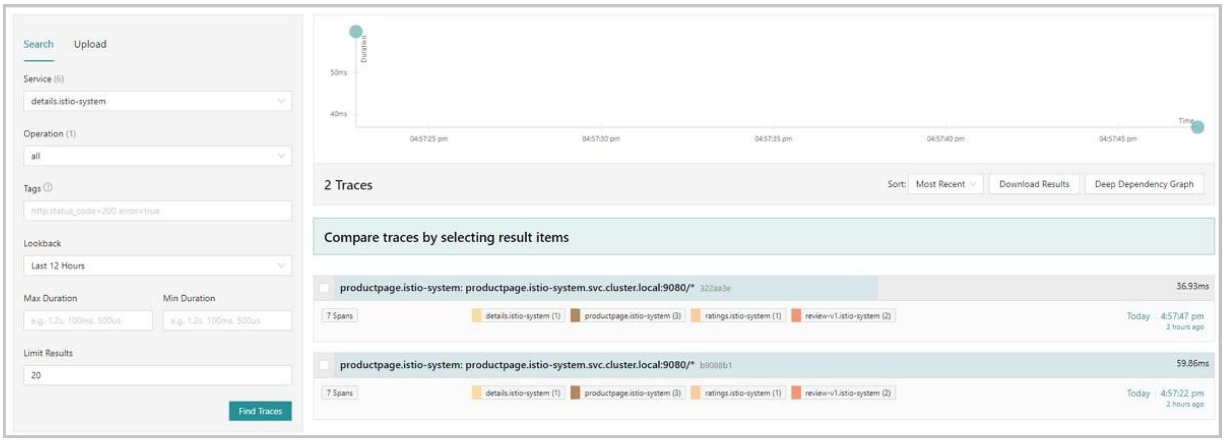
Note: To use the microservices governance feature in a cluster, you need to deploy the microservices governance component in the cluster first, you can either enable the microservices governance component when you create the cluster, or go to the cluster details page to deploy the feature component list.

7.5.3.2 Viewing Call Chain Data

In the nKU main menu, select **Container Orchestration** > **Microservices Governance** > **Trace Tracking** to enter the **trace tracking** interface. You can query the research chain by service name, TraceID, and request time, and support customizing the query time period. Each piece of data displays a request operation, including access time, request duration, number of spans, and information about each span.

As shown in Figure 37: Call Chain:

Figure 37: Call Chain



Service (6): details.istio-system

Operation (1): all

Tags: http.status_code=200 error=true

Lookback: Last 12 Hours

Max Duration: e.g. 1.2s, 100ms, 500us

Min Duration: e.g. 1.2s, 100ms, 500us

Limit Results: 20

2 Traces

Sort: Most Recent

Compare traces by selecting result items

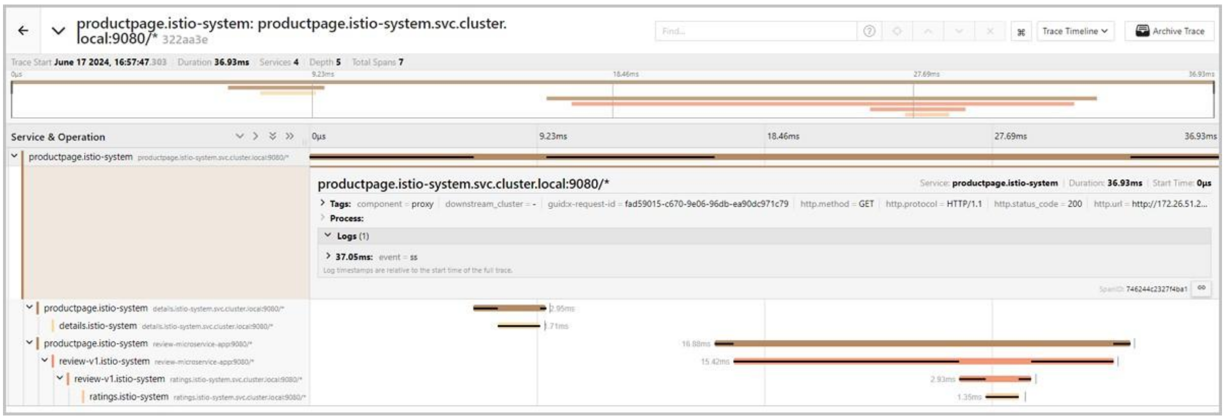
productpage.istio-system: productpage.istio-system.svc.cluster.local:9080/* 322aa3e — 36.93ms
7 Spans — details.istio-system (1) productpage.istio-system (3) ratings.istio-system (1) review-v1.istio-system (2) — Today 4:57:47 pm, 2 hours ago

productpage.istio-system: productpage.istio-system.svc.cluster.local:9080/* b9088b1 — 59.86ms
7 Spans — details.istio-system (1) productpage.istio-system (3) ratings.istio-system (1) review-v1.istio-system (2) — Today 4:57:22 pm, 2 hours ago

Click on the request to go to the details page where you can view the request details.

As shown in Figure 38: Call Chain Details:

Figure 38: Call Chain Details



productpage.istio-system: productpage.istio-system.svc.cluster.local:9080/* 322aa3e

Trace Start **June 17 2024, 16:57:47**.303 | Duration **36.93ms** | Services **4** | Depth **5** | Total Spans **7**

Service & Operation

productpage.istio-system productpage.istio-system.svc.cluster.local:9080/*

productpage.istio-system.svc.cluster.local:9080/*

Service: **productpage.istio-system** | Duration: **36.93ms** | Start Time: **0μs**

Tags: component = proxy | downstream_cluster = - | guid:x-request-id = fad59015-c670-9e06-96db-ea90dc971c79 | http.method = GET | http.protocol = HTTP/1.1 | http.status_code = 200 | http.url = http://172.26.51.2...

Process:

Logs (1)

37.05ms: event = ss

Log timestamps are relative to the start time of the full trace.

SpanID: 746244c2327f4ba1

productpage.istio-system details.istio-system.svc.cluster.local:9080/* 2.95ms

details.istio-system details.istio-system.svc.cluster.local:9080/* 1.71ms

productpage.istio-system review-microservice-app:9080/* 16.88ms

review-v1.istio-system review-microservice-app:9080/* 15.42ms

review-v1.istio-system ratings.istio-system.svc.cluster.local:9080/* 2.93ms

ratings.istio-system ratings.istio-system.svc.cluster.local:9080/* 1.35ms

8 Artifact repository

8.1 Repository

8.1.1 General

The Repository is used to store Docker images and charts uploaded by users. Users can upload their modified images and Charts to the Repository, or download the uploaded Docker images and Charts from the Repository.

8.1.2 Create Repository

In the nKU main menu, click **Artifact Repository** > **Repository**. In the **Repository** screen, click **Create Repository** to bring up the **Create Repository** screen.

You can refer to the following example to enter the appropriate contents:

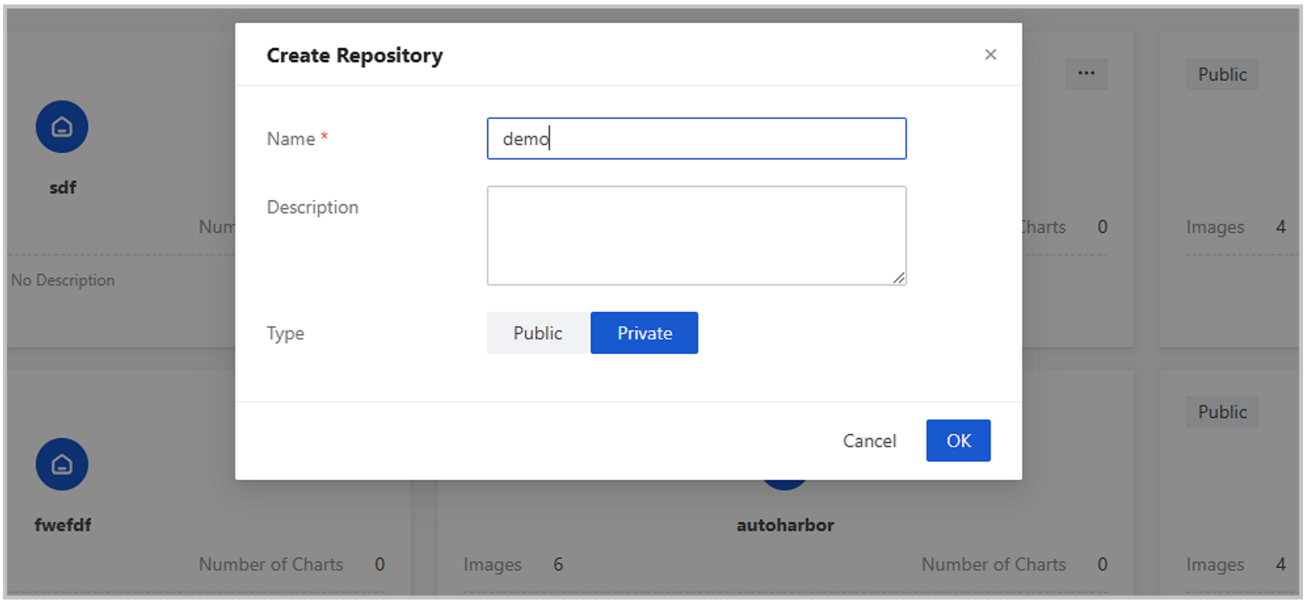
- **Name**: Set the name of the repository. Naming rules: length limit 2-50 characters, can only contain lowercase letters, numbers and special characters (_)(-) (.), special characters can not be used at the beginning or end and can not be used consecutively. Special characters can not be used at the beginning or the end and can not be used consecutively.
- **Description**: optional, note repository-related information
- **Type**: set the type of repository, support both public and private types

# **Public**: Public repositories do not require a login to pull an image repository, and all users of the platform can view the repository.

# **Private**: A private type of repository requires login credentials to pull an image repository, and can only be viewed by users who manage the project to which the repository belongs.

As shown in Figure 39: Create Repository:

Figure 39: Create Repository



Create Repository

Name * demo

Description

Type: Public | Private

8.1.3 Managing the Repository

In the nKU main menu, click **Artifact Repository** > **Repository** to enter the **Repository** interface.

The Repository supports the following operations:

manipulate	**descriptive**
Create Repository	Create a new repository.
Edit Repository	Modify the repository profile and type.
Reset Password	Reset the user password for the Repository, through which users can log in to the repository using the docker/nerdctl login command. **Notes.** • A platform user corresponds to a Repository user, and the user name is automatically generated by the platform. • When you use it for the first time and do not have a login password, you can use **Reset Password** to set your password, which does not require you to verify your old password.
Repository Details	Go to the Repository details page to view basic repository information and image exporting records, and centrally manage images and Charts in the repository.

Change project	Change the project to which the Repository belongs. **Note:** After changing projects, members of the original project will no longer be able to view or manage that Repository and the resources in the repository.
Delete Repository	Remove the Repository. **Note:** Deleting a repository requires you to delete all images and Charts from the repository.

8.1.4 Repository details

8.1.4.1 Docker images

8.1.4.1.1 Uploading Images

On the Repository details page, click **Images** to go to the **Images** tab, click **Upload** to bring up the **upload image** interface.

The following three Image upload methods are supported:

- online upload
- file upload
- command line upload

online upload

You can refer to the following example to enter the appropriate contents:

- **Upload method**: choose **online upload**
- **External Image address**: fill in the URL address of the external Image, if the URL does not contain the label of the Image, the system will use the latest tag by default.
- **Architecture**: Select the architecture of the image to be uploaded. You can choose linux/arm64 and linux/amd64, multiple choices are supported.

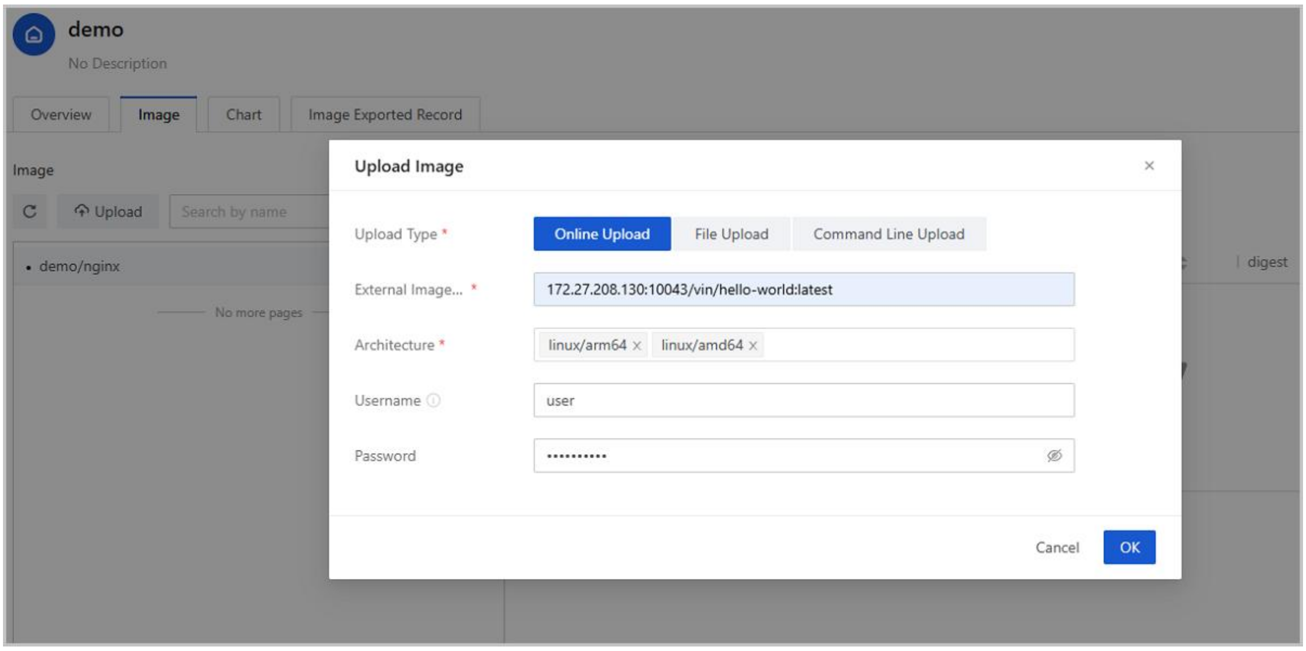
  Note: For multi-architecture Images, if only one architecture is selected, only the layer for the selected architecture will be uploaded.

- **Username**: Fill in the username of the repository where the image repository belongs to. If the image is stored in a private repository, you need to fill in this parameter in order to verify the identity of the repository when pulling the image.

- **Password**: Fill in the password that corresponds to your username. This parameter should be filled in when the image is stored in a private repository in order to verify your identity when pulling the image.

As shown in Figure 40: Upload Image - Online Upload:

Figure 40: Upload Image - Online Upload



Upload Image

Upload Type *: Online Upload | File Upload | Command Line Upload

External Image... *: 172.27.208.130:10043/vin/hello-world:latest

Architecture *: linux/arm64 × linux/amd64 ×

Username: user

Password: ••••••••••

file upload

You can refer to the following example to enter the appropriate contents:

- **Upload Method**: Select **File Upload**
- **File upload**: file upload, only support to upload the container management tool exported Image compressed package

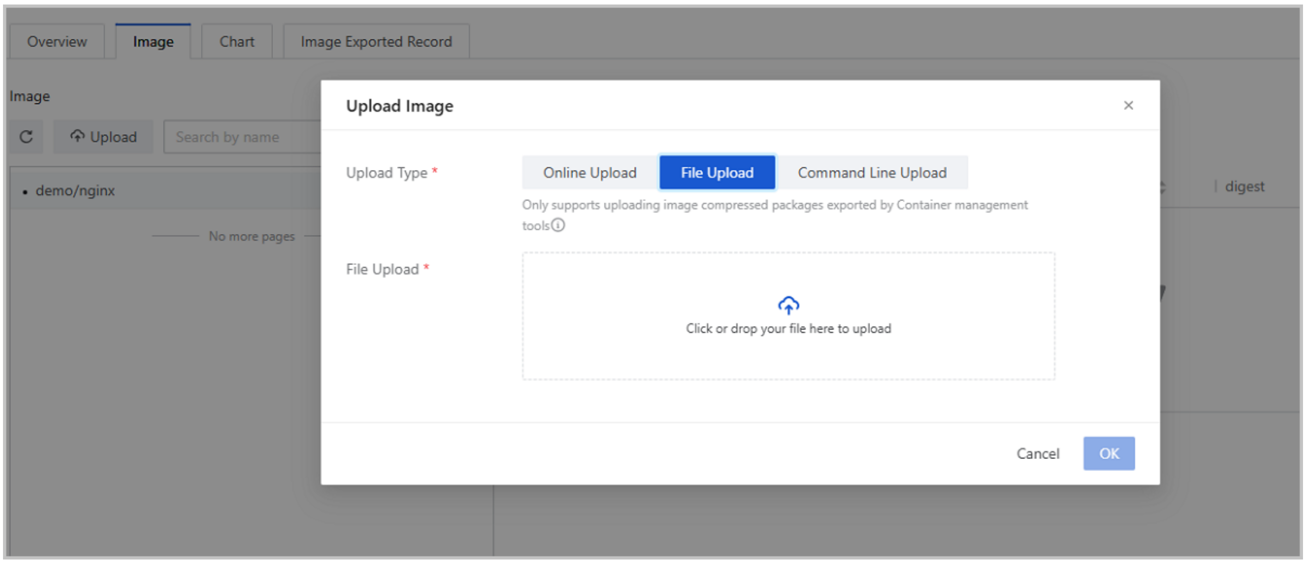
Note: Users can generate image tarballs using docker or nerdctl commands, for example:

```
docker save -o $OUTPUT_FILENAME $IMAGE:$TAG
  # $OUTPUT_FILENAME refers to the generated zip file name, and $IMAGE:$TAG refers to
  the image and tag to be compressed.
nerdctl save -o $OUTPUT_FILENAME $IMAGE:$TAG
```

```
# $OUTPUT_FILENAME refers to the generated zip file name, and $IMAGE:$TAG refers to the image and tag to be compressed.
```

As shown in Figure 41: Upload Image - File Upload:

Figure 41: Upload Image - File Upload



Upload Image

Upload Type * Online Upload | File Upload | Command Line Upload

Only supports uploading image compressed packages exported by Container management tools

File Upload * Click or drop your file here to upload

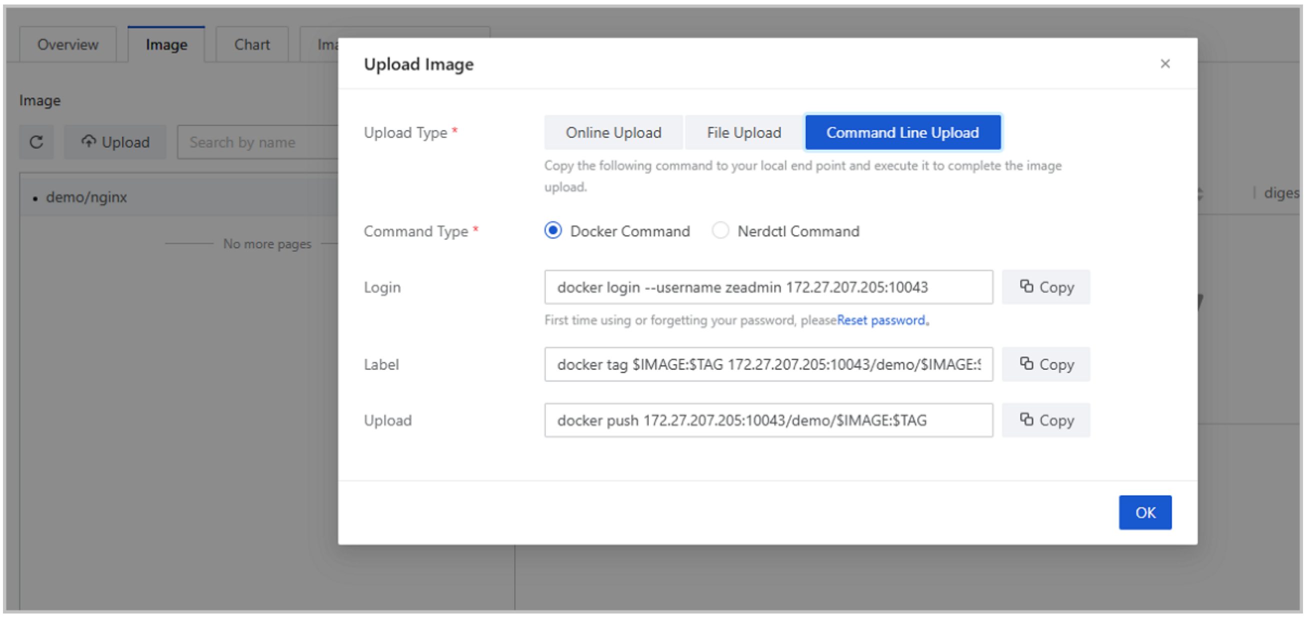
command line upload

Command line upload refers to pushing an image from an external repository to a Repository via the docker or nerdctl command, you can refer to the following steps to complete the upload:

1. In the **Upload Image** screen, select the upload method as **command line upload** and select the command type.
2. Copy the command line uploaded from the bottom of the interface to a web terminal that can be connected to an external repository, modify the variable parameters and execute the command to complete the image repository upload.

As shown in Figure 42: Upload Image-Command line upload:

Figure 42: Upload Image - Command Line Upload



Upload Type * Command Line Upload

Copy the following command to your local end point and execute it to complete the image upload.

Command Type * ● Docker Command ○ Nerdctl Command

Login: `docker login --username zeadmin 172.27.207.205:10043`

First time using or forgetting your password, pleaseReset password。

Label: `docker tag $IMAGE:$TAG 172.27.207.205:10043/demo/$IMAGE:$`

Upload: `docker push 172.27.207.205:10043/demo/$IMAGE:$TAG`

8.1.4.1.2 Downloading and pulling Images

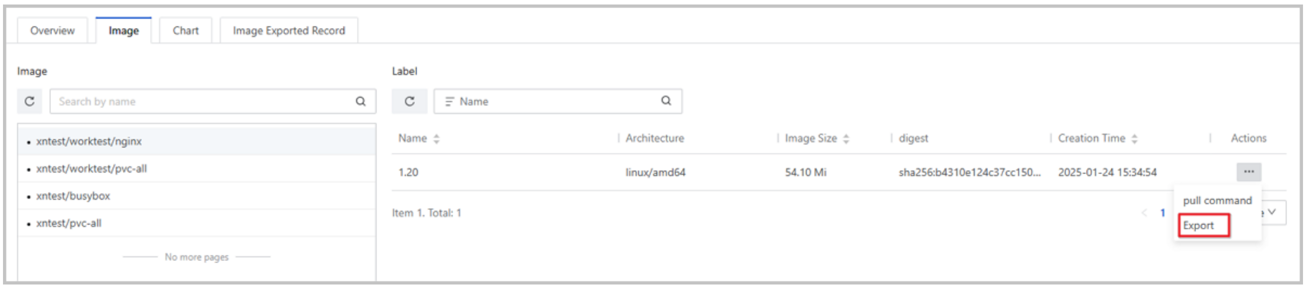
Download Image

Users can refer to the following steps to download the image from Repository for use:

1. On the Repository details page, go to the **Image** tab, find the image you need to download, and click **the action** > **Export**.

   As shown in Figure 43: Export Image:

   Figure 43: Export Image



- xntest/worktest/nginx
- xntest/worktest/pvc-all
- xntest/busybox
- xntest/pvc-all

Name	Architecture	Image Size	digest	Creation Time	Actions
1.20	linux/amd64	54.10 Mi	sha256:b4310e124c37cc150...	2025-01-24 15:34:54	···

Item 1. Total: 1

2. Go to the **Image Export Logs** tab of the Local Image Details page, locate the image exported in step 1, and click **the operation** > **Download** to download the image zip package locally.

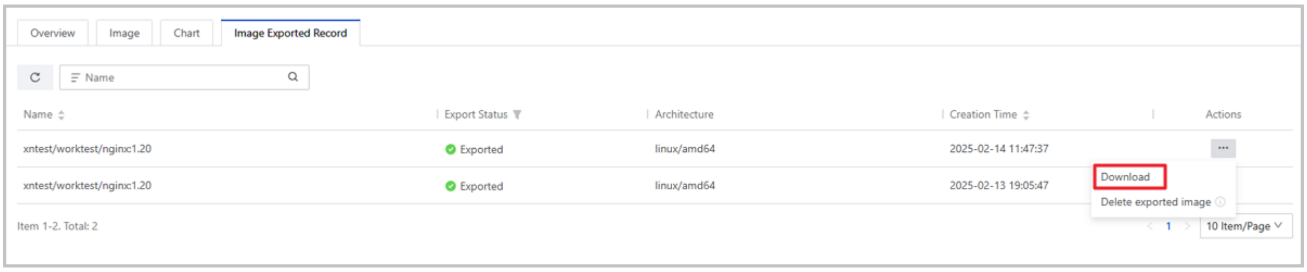
Notes.

- The exported image should be downloaded within 24 hours, otherwise the exported record will be invalid and need to be re-exported.

- When exporting a large number of larger Images for a short period of time, it may cause the storage space to be full and no more Images can be exported, so you can delete the exported Images that have been downloaded first.

As shown in Figure 44: Download Image:

Figure 44: Download Image



Name	Export Status	Architecture	Creation Time	Actions
xntest/worktest/nginx:1.20	Exported	linux/amd64	2025-02-14 11:47:37	···
xntest/worktest/nginx:1.20	Exported	linux/amd64	2025-02-13 19:05:47	

3. (Optional) Execute the docker load command to import the downloaded image for use in another repository, for example:

```
docker load -i $IMAGE.TAR
  # $IMAGE.TAR is the name of the image tarball
```

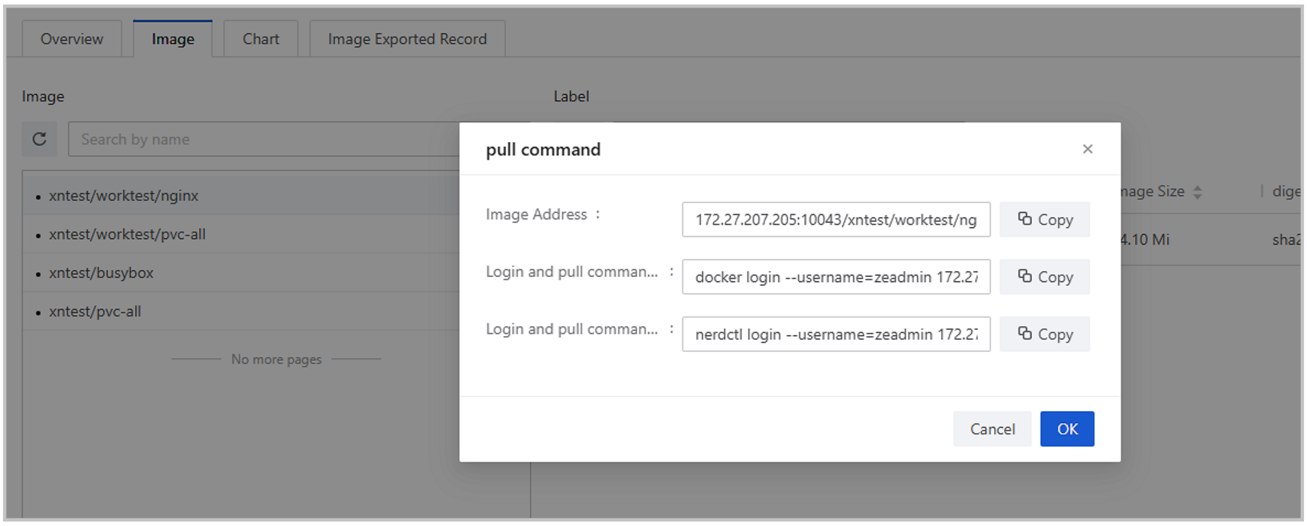
Pulling Images

Users can refer to the following steps to pull the image in their Repository:

1. On the Repository details page, go to the **Images** tab, find the image you need to download, and click **Action** > **Pull Commands** to view the image registry address, docker commands for logging into the repository and pulling images, and nerdctl commands.

As shown in Figure 45: Pull command:

Figure 45: Pull Command



pull command

Image Address ： 172.27.207.205:10043/xntest/worktest/ng

Login and pull comman... ： docker login --username=zeadmin 172.27

Login and pull comman... ： nerdctl login --username=zeadmin 172.27

2. The Copy command and is used for scenes that require pulling Images.

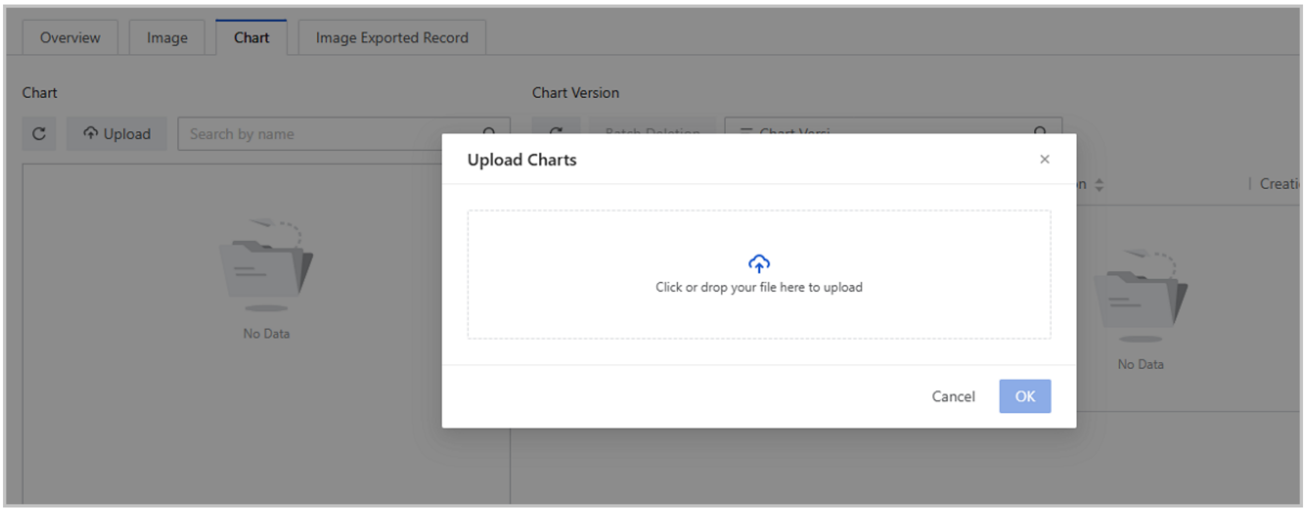
8.1.4.2 Chart

8.1.4.2.1 Uploading Chart

On the Repository details page, click **Chart**, go to the **Chart** label, click **Upload** to bring up the **Upload Chart** interface. Users can upload the local Chart to this interface.

As shown in Figure 46: Upload Chart:

Figure 46: Upload Chart



8.1.4.2.2 Installing applications

Publishing an application is to publish the Chart to the cluster and deploy an instance of the application. nKU supports one-click publishing of Chart from the Repository.

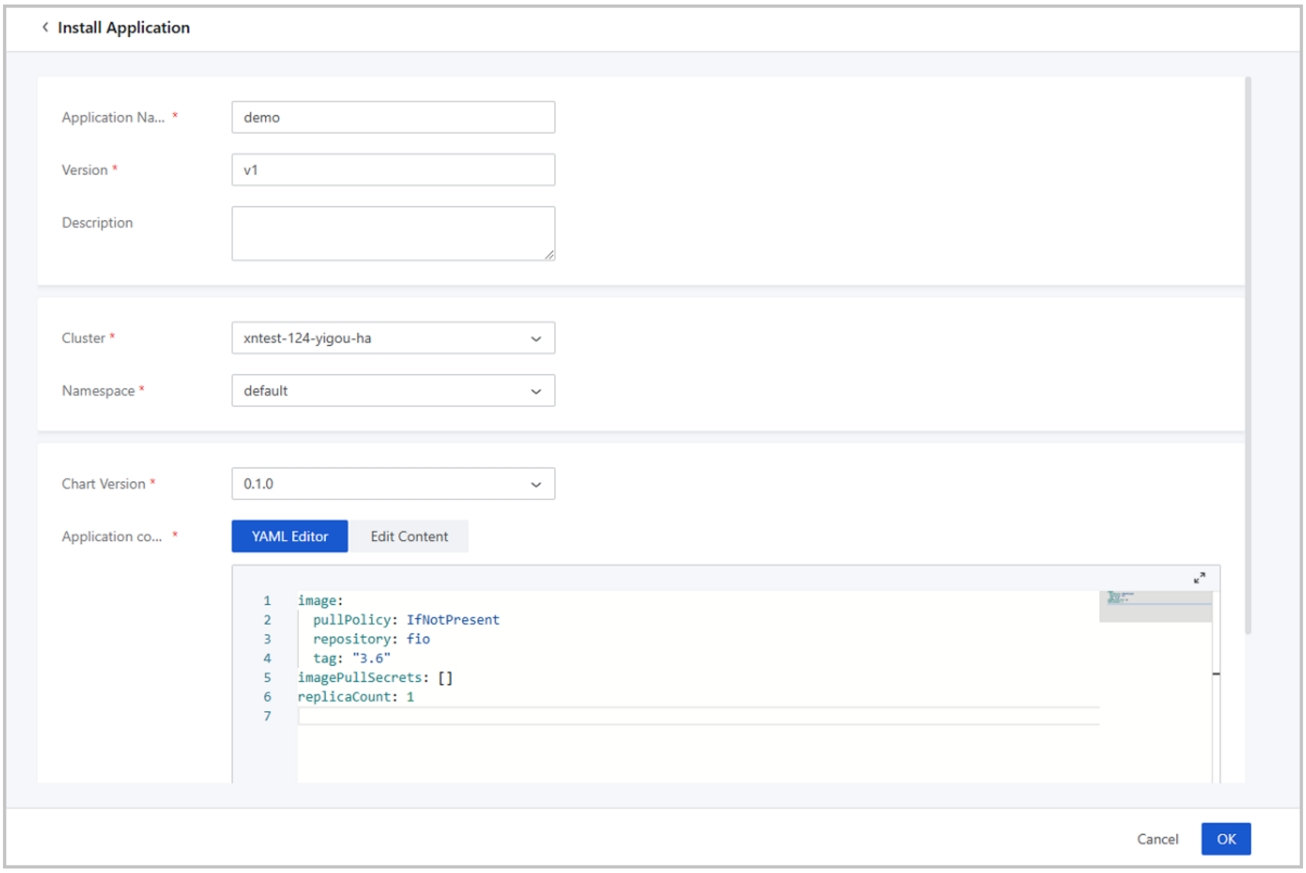
Go to the **Chart** label on the Repository details page, find the Chart you need to install and click on the **action** > **Install** to bring up the **Install application** interface.

You can refer to the following example to enter the appropriate contents:

- **Application Name**: Set the application name. The length is limited to 2~20 characters, the input can only contain lowercase letters, numbers and separator (-), and must start with a lowercase letter and end with a lowercase letter or number
- **Version**: Setting the application version
- **Description**: optional, note application-related information
- **Cluster**: Select the cluster to which the application is installed
- **Namespace**: select the namespace to which the application is installed
- **Chart version**: select the Chart version, default is the version shown in the Chart list
- **Application Configuration**: Modify the contents of the VALUES file in the deployment chart via YAML. Users can click on **the modified content** to see the comparison between before and after modification

As shown in Figure 47: Install application:

Figure 47: Install Application



Install Application

Application Na... * demo

Version * v1

Description

Cluster * xntest-124-yigou-ha

Namespace * default

Chart Version * 0.1.0

Application co... * YAML Editor | Edit Content

```
image:
  pullPolicy: IfNotPresent
  repository: fio
  tag: "3.6"
imagePullSecrets: []
replicaCount: 1

```

Note: After an application is installed, you can view and manage the application from the nKU main menu by clicking **Container Orchestration** > **Application Management** > **Application** to access the **application** interface.

8.2 YAML template repository

8.2.1 General

The YAML template repository is used to store YAML templates written by users and provides example YAML template repositories for Deployment, Statefulset, Service, Configmap, PersistentVolumeClaim, and more than a dozen other common Kubernetes resources. Users can combine the example YAML templates in the template repository to quickly write the required YAML text.

8.2.2 Creating templates

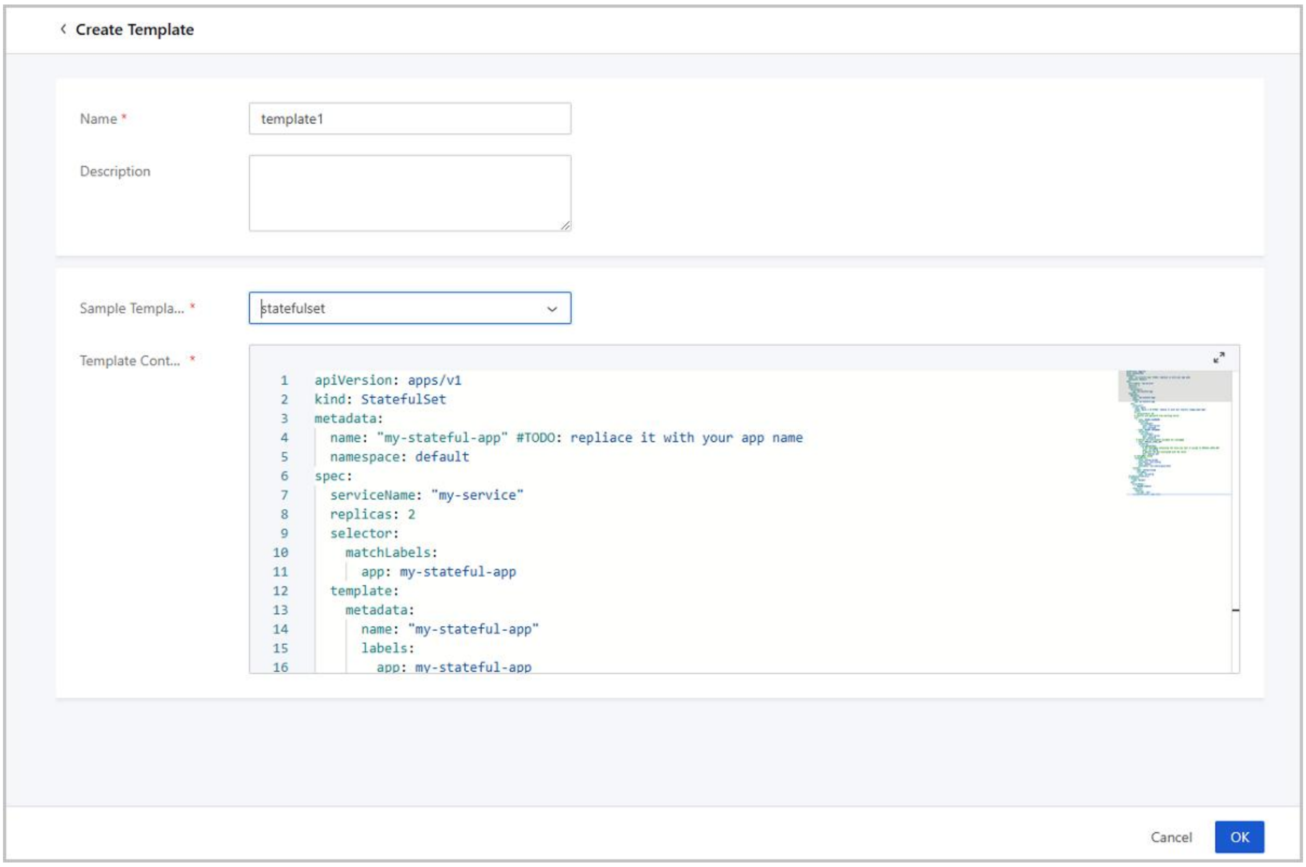
In the nKU main menu, click **Artifact repository** > **template repository** to enter the YAML **template repository** interface. Click **Create Template** to bring up the **Create Template** interface.

You can refer to the following example to enter the appropriate contents:

- **Name**: Set the name of the template. Naming rules: length limit of 2-50 characters, the input can only contain lowercase letters, numbers and separators (-), separators can not be used at the beginning or end of the name
- **Description**: optional, note template-related information
- **Example YAML**: Select Deployment, Statefulset, Service, Configmap, PersistentVolumeClaim and other common resource example YAML, or customize the template.
- **Template content**: when choosing a common resource example YAML template, you can modify it based on the YAML example; when choosing a customized template, you can fill in the YAML content by yourself. Support filling multiple resource YAMLs separated by (---).

As shown in Figure 48: Creating a template:

Figure 48: Creating a template



Name * template1

Description

Sample Templa... * statefulset

Template Cont... *

```yaml
apiVersion: apps/v1
kind: StatefulSet
metadata:
  name: "my-stateful-app" #TODO: repliace it with your app name
  namespace: default
spec:
  serviceName: "my-service"
  replicas: 2
  selector:
    matchLabels:
      app: my-stateful-app
  template:
    metadata:
      name: "my-stateful-app"
      labels:
        app: my-stateful-app
```

8.2.3 Managing templates

In the nKU main menu, click **Artifact Repository** > **template repository** to enter the YAML **template repository** interface.

The YAML template repository supports the following operations:

manipulate	descriptive
Create Template	Create a template from an example YAML, or create a customized template.
Edit template	Modify the profile and content of the template.
Change project	Change the project to which the template belongs. 📋 **Note:** After the change, members of the original project will no longer be able to view and manage the template.
Delete Template	Remove the template.

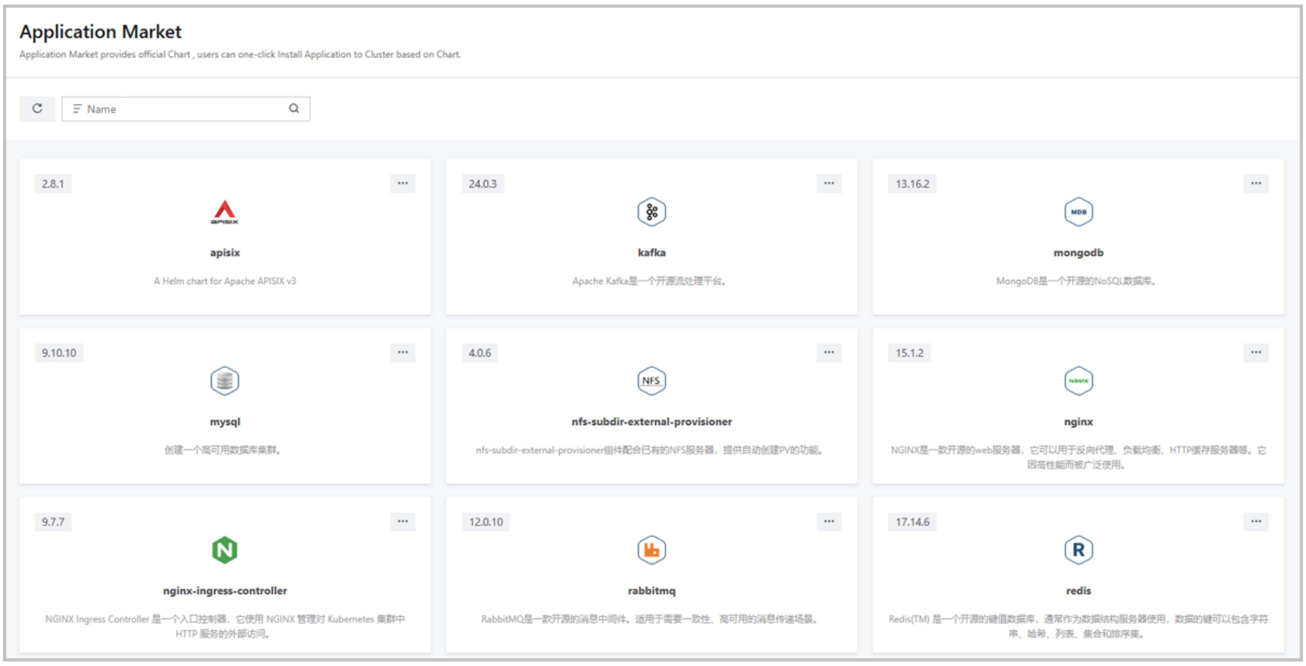
8.3 Application market

8.3.1 General

The application market provides rich official Chart such as kafka, zookeeper, mysql, redis, rabbitmq,deepseek, etc., which can be installed to the cluster and deployed as application instances with a single click.

As shown in Figure 49: Application market:

Figure 49: Application market



Application Market

Application Market provides official Chart , users can one-click Install Application to Cluster based on Chart.

2.8.1 apisix

A Helm chart for Apache APISIX v3

24.0.3 kafka

Apache Kafka是一个开源流处理平台。

13.16.2 mongodb

MongoDB是一个开源的NoSQL数据库。

9.10.10 mysql

创建一个高可用数据库集群。

4.0.6 nfs-subdir-external-provisioner

nfs-subdir-external-provisioner组件配合已有的NFS服务器，提供自动创建PV的功能。

15.1.2 nginx

NGINX是一款开源的web服务器，它可以用于反向代理、负载均衡、HTTP缓存服务器等。它因高性能而被广泛使用。

9.7.7 nginx-ingress-controller

NGINX Ingress Controller 是一个入口控制器，它使用 NGINX 管理对 Kubernetes 集群中 HTTP 服务的外部访问。

12.0.10 rabbitmq

RabbitMQ是一款开源的消息中间件，适用于需要一致性、高可用的消息传递场景。

17.14.6 redis

Redis(TM) 是一个开源的键值数据库，通常作为数据结构服务器使用，数据的键可以包含字符串、哈希、列表、集合和排序集。

8.3.2 Installing applications

To install application is to publish the Chart to the cluster and deploy an application instance, nKU supports one-click publishing of Chart packages from the application market.

You can refer to the following example to enter the appropriate contents:

- **Application Name**: Set the application name. The length is limited to 2~20 characters, the input can only contain lowercase letters, numbers and separator (-), and must start with a lowercase letter and end with a lowercase letter or number
- **Version**: Setting the application version
- **Description**: optional, note application-related information
- **Cluster**: Select the cluster to which the application is installed
- **Namespace**: select the namespace to which the application is installed

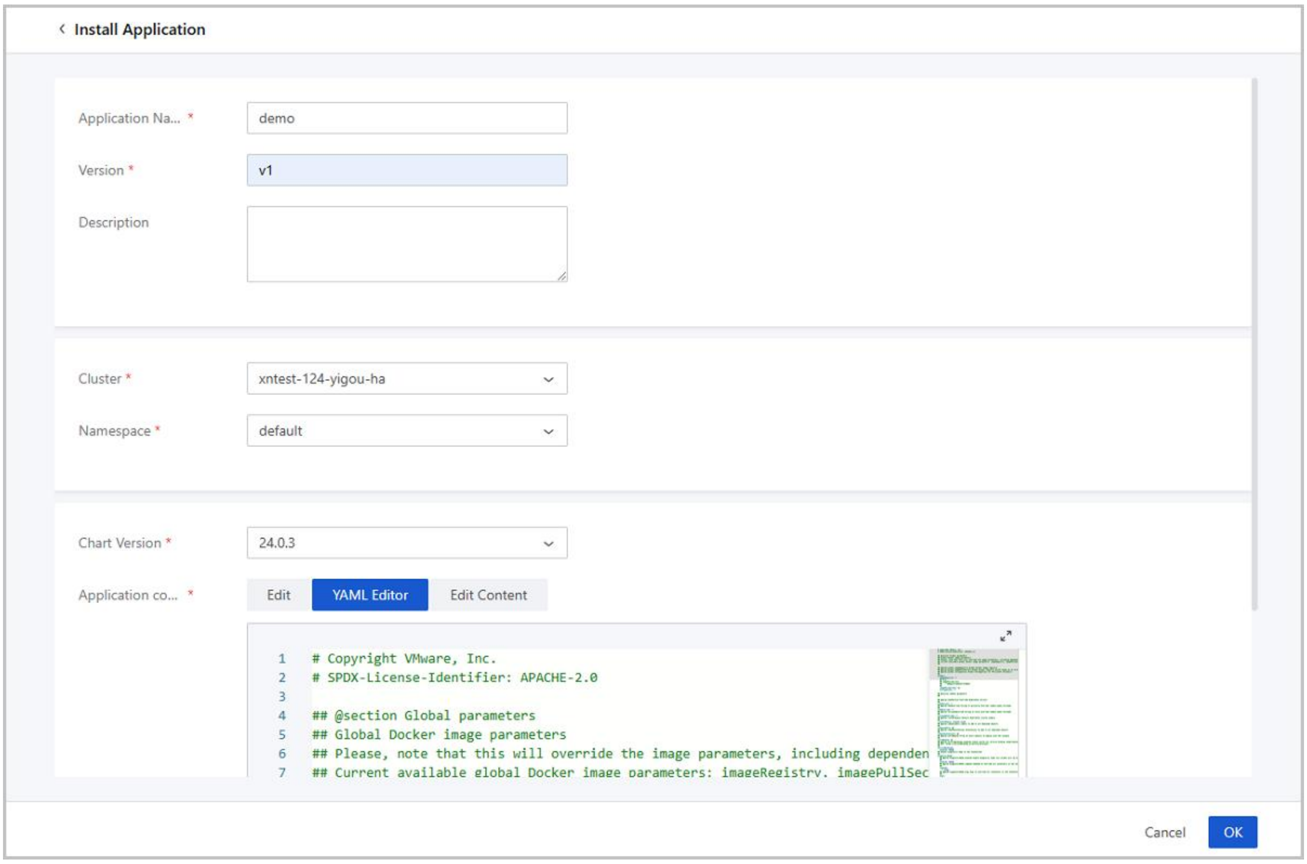
- **Chart version**: display the default Chart version, currently does not support changes
- **Application Configuration**: Modify the configuration content of the values file in the deployment package, supporting two modification methods: **chart editing** and **YAML editing**. Users can click on **the modified content** to see the comparison between before and after modification

  # Edit: Provides visual modification for common configurations, mainly used for simpler configuration modifications.

  # YAML Editor: Modify the application YAML in the YAML editor, mainly used for more complex configuration changes.

As shown in Figure 50: Install application:

Figure 50: Installing Applications



Application Na... * demo

Version * v1

Description

Cluster * xntest-124-yigou-ha

Namespace * default

Chart Version * 24.0.3

Application co... *

```
# Copyright VMware, Inc.
# SPDX-License-Identifier: APACHE-2.0

## @section Global parameters
## Global Docker image parameters
## Please, note that this will override the image parameters, including dependen
## Current available global Docker image parameters: imageRegistry, imagePullSec
```

Note: After an application is installed, you can view and manage the application from the nKU main menu by clicking **Container Orchestration** > **Application Management** > **Application** to access the **application** interface.

9 Container O&M

9.1 One-Click Inspection

9.1.1 General

Conduct comprehensive health inspections of key platform resources and services, and make optimization recommendations for risk items to ensure that platform resources and services are in the best condition and help efficient operation and maintenance.

Function Principle

- **Inspection Categories and Items:**

  One-click inspection provides three types of inspection items (Basic Services, Computing, Network) for management clusters and business clusters. It supports inspections of key platform resources and services such as cluster nodes, image servers, and network components.

  - Basic Services: Detect the operating status of cluster basic services and components.
  - Computing: Detect the usage and operating status of computing resources.
  - Network: Detect network configurations and status.

  Users can customize the selection of clusters and inspection items for one-click inspection. After starting the inspection, the platform will perform health checks on the resources or services involved in the selected items. For detailed information on inspection items, refer to the Inspection Item Overview.

- **Inspection Results:**

  One-click inspection provides four types of results for inspected resources or services: Normal, Warning, Fault, and Failure.

  - Normal: The inspected resource or service is in normal status, marked with a green icon.
  - Warning: The inspected resource or service is in poor status, which may affect the performance and stability of related resources and services to a certain extent, marked with a yellow icon.

  - Crash: The inspected resource or service is in a critical status, which may seriously affect business operations, marked with a red icon.
  - Failure: The inspection of the resource or service failed, which may seriously affect business operations, marked with a gray icon.

- **Health Scoring:**

  One-click inspection has a built-in health scoring mechanism that supports quantitative scoring of the health status of inspected resources or services, helping users intuitively and accurately grasp the overall operating status of the platform.

  - Resource/Service Score: Scored based on the inspection result of the resource or service.
    - If the inspection result of a resource or service is Normal, the score is 100 points.
    - If the inspection result is Warning, the score is 50 points.
    - If the inspection result is Fault or Failure, the score is 0 points.
  - Inspection Item Score: Calculated based on the scores of resources or services involved in the inspection item. Details are as follows:
    - Scoring Mechanism: (Resource 1 Score + Resource 2 Score + ... + Resource N Score) / (N * 100) * 100.
    - Example: Suppose an inspection item involves 3 resources with results Normal, Warning, and Fault/Failure, corresponding to scores 100, 50, and 0. The score of the inspection item is: (100 + 50 + 0) / (3 * 100) * 100 = 50 points.
  - Cluster Score: Comprehensive calculation based on the scores of each inspection item under the cluster and the weight of each item. Details are as follows:
    - Scoring Mechanism: (Inspection Item 1 Score * Inspection Item 1 Weight + Inspection Item 2 Score * Inspection Item 2 Weight + ... + Inspection Item N Score * Inspection Item N Weight) / (100 * Inspection Item 1 Weight + 100 * Inspection Item 2 Weight + ... + 100 * Inspection Item N Weight) * 100.
    - Example: Suppose a user selects 3 inspection items under a cluster, with scores 100, 50, and 0, and corresponding weights 1, 2, and 3. The cluster score is: (100*1 + 50*2 + 0*3) / (100*1 + 100*2 + 100*3) * 100 = 33 points.
  - Overall Platform Score: Average of the scores of all clusters.

  - Scoring Mechanism: (Cluster 1 Score + Cluster 2 Score + ... + Cluster N Score) / N.
  - Example: Suppose a user selects 3 clusters for one-click inspection, with scores 100, 50, and 0. The overall platform score is: (100 + 50 + 0) / 3 = 50 points.

- **Inspection Recommendations**:

  For resources and services with Warning or Crash status, one-click inspection analyzes potential risks and their impacts on related resources and services, and provides targeted repair recommendations. For detailed recommendations, refer to the Inspection Item Overview.

- **Inspection Report:**

  One-click inspection supports exporting inspection reports in WORD format. The report displays an overview of inspection information, result statistics, and summarizes details of all abnormal inspection items sorted by severity level, with inspection recommendations for each abnormal item.

Functional Advantages

nKU One-Click Inspection has the following advantages:

- Comprehensive, Efficient, and Customizable Inspection: Three categories of inspection items cover all key resources and services on the platform. It supports customized inspection selection and delivers results within minutes.
- Multi-Level Scoring Mechanism: Built-in four-level scoring mechanism (Resource/Service, Inspection Item, Cluster, Platform) helps users grasp the platform's operating status from micro to macro perspectives.
- Intelligent Risk Identification and Recommendations: Intelligently pushes resource-level risk analysis and response measures to facilitate accurate and efficient operation and maintenance.
- Inspection Without Login: Supports one-click inspection even when the platform service is abnormal, helping to quickly locate problems.

9.1.2 One-Click Inspection

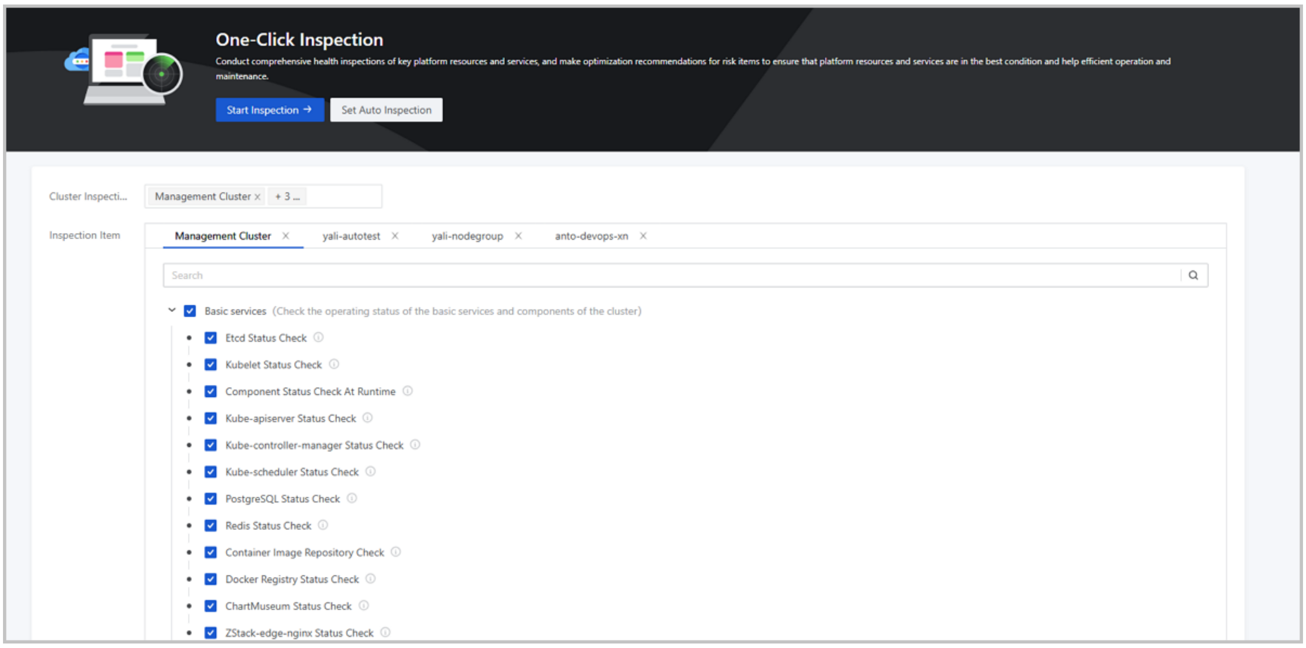
In the main menu of nKU, click **Container O&M > One-Click Inspection** to enter the One-Click Inspection interface.

One-Click Inspection supports the following operations:

Operation	Description
Start One-Click Inspection	After customizing the clusters and inspection items to be inspected, perform a one-click inspection on the selected items.
Cancel One-Click Inspection	When a one-click inspection is in progress, cancel the inspection of the selected items.
Retest	After a one-click inspection is completed, re-check the inspection items selected in the last inspection.
Export Inspection Report	Export an inspection report in WORD format.
Set Auto Inspection	Supports setting up automatic inspection. The platform will automatically inspect all inspection items according to the set inspection start time and cycle. After the inspection, users can enter the one-click inspection page to view or export detailed inspection reports. • Auto inspection is enabled by default, with a default cycle of 8 hours, and can be disabled as needed. • Supports setting the start time and cycle of auto inspection as needed. • Supports enabling email notifications as needed. If email notifications are enabled, emails will be sent to specified recipients when abnormal inspection items are found.

As shown in Figure: One-Click Inspection:

Figure: One-Click Inspection



One-Click Inspection

Conduct comprehensive health inspections of key platform resources and services, and make optimization recommendations for risk items to ensure that platform resources and services are in the best condition and help efficient operation and maintenance.

Cluster Inspecti... Management Cluster × + 3 ...

Inspection Item: Management Cluster × | yali-autotest × | yali-nodegroup × | anto-devops-xn ×

Search

☑ Basic services (Check the operating status of the basic services and components of the cluster)

- ☑ Etcd Status Check
- ☑ Kubelet Status Check
- ☑ Component Status Check At Runtime
- ☑ Kube-apiserver Status Check
- ☑ Kube-controller-manager Status Check
- ☑ Kube-scheduler Status Check
- ☑ PostgreSQL Status Check
- ☑ Redis Status Check
- ☑ Container Image Repository Check
- ☑ Docker Registry Status Check
- ☑ ChartMuseum Status Check
- ☑ ZStack-edge-nginx Status Check

9.1.3 View Inspection Results

In the main menu of nKU, click **Container Operation and Maintenance > One-Click Inspection** to enter the One-Click Inspection interface. After selecting the clusters and their corresponding inspection items to be inspected, click **Start One-Click Inspection**. Once the inspection is completed, you can view the inspection report.

The inspection results display the overall health status of the selected inspection items through quantitative scoring. They also categorize and summarize the selected items by status, facilitating users to quickly find abnormal ones. Meanwhile, the system clearly presents the information and status of resources involved in each inspection item in a list format, along with corresponding inspection recommendations.

1. **Overall Health Check Results**

   The upper section of the inspection results page summarizes the total number of inspection items and the number of abnormal items for the current inspection. It also calculates an overall score based on the health scoring mechanism, intuitively showing the overall health status of the inspected items. Additionally, the inspection results record the duration and completion time of the current one-click inspection, helping users schedule the next inspection. For details on the health scoring mechanism, refer to Health Scoring.

2. **Categorized Display of Inspection Item Statuses**

   The lower left section of the inspection results page summarizes all clusters and their inspection items for the current inspection. All items are classified into **Abnormal** and **Normal** based on the inspection results of the resources they involve. Users can freely switch between the Normal and Abnormal pages to quickly search and locate inspection items. The specific classification rules are as follows:

- If all inspection results of the resources or services involved in an item are Normal, the item is marked with a green icon and classified as Normal.
- If all results are Warning, or some are Warning and the rest are Normal, the item is marked with a yellow icon and classified as Abnormal.
- If all results are Fault, or some are Fault and the rest are Normal or Warning, the item is marked with a red icon and classified as Abnormal.
- If all results are Failure, or some are Failure and the rest are Normal, Warning, or Fault, the item is marked with a gray icon and classified as Abnormal.

3. Inspection Resource Information, Status, and Recommendations

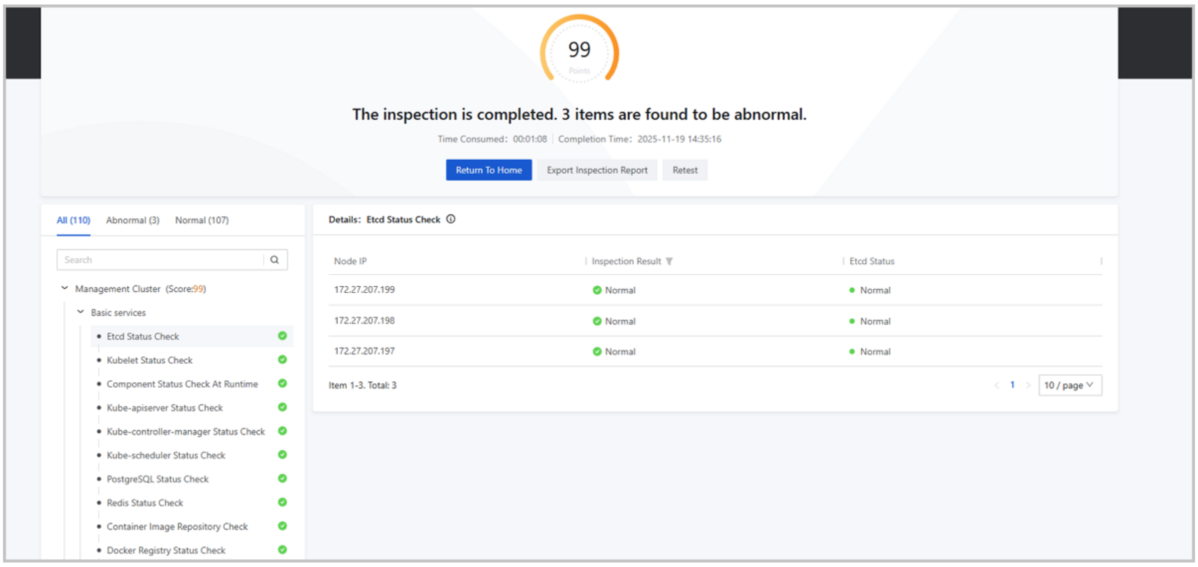
Users can click any inspection item in the lower left section of the inspection report page. The lower right section will then display the basic information and inspection results of the resources involved in that item in detail, along with inspection recommendations for resources with Warning or Fault statuses.

- List Information:
    - Basic Information: One-click inspection shows different basic information for different resources and services. For example, the "Node Time Synchronization Check" item can display node IP, node time, and chronyd service status.
    - Inspection Results: Display corresponding results based on the health status of resources and services, including Failure, Fault, Warning, and Normal.
        - Failure: Unable to inspect the resource or service, marked with a gray icon.
        - Crash: The inspected resource or service is in a critical status, which may seriously affect business operations, marked with a red icon.
        - Warning: The inspected resource or service is in poor status, which may affect the performance and stability of related resources and services to a certain extent but will not seriously impact business operations, marked with a yellow icon.
        - Normal: The inspected resource or service is in normal status, marked with a green icon.
    - Inspection Recommendations: For resources and services with Warning or Fault statuses, one-click inspection analyzes potential risks and their impacts, and provides targeted repair suggestions to support efficient operation and

maintenance. For detailed recommendations, refer to the Inspection Item Overview.

As shown in Figure: View Inspection Results:

Figure: View Inspection Results



The inspection is completed. 3 items are found to be abnormal.

Time Consumed: 00:01:08 | Completion Time: 2025-11-19 14:35:16

Management Cluster (Score:99)

Basic services

- Etcd Status Check
- Kubelet Status Check
- Component Status Check At Runtime
- Kube-apiserver Status Check
- Kube-controller-manager Status Check
- Kube-scheduler Status Check
- PostgreSQL Status Check
- Redis Status Check
- Container Image Repository Check
- Docker Registry Status Check

Details: Etcd Status Check

Node IP	Inspection Result	Etcd Status
172.27.207.199	Normal	Normal
172.27.207.198	Normal	Normal
172.27.207.197	Normal	Normal

Item 1-3. Total: 3   1   10 / page

9.1.4 Perform One-Click Inspection Without Login

When the platform service is abnormal and login is unavailable, one-click inspection of the management cluster is supported without login, facilitating quick problem location.

You can access the login-free one-click inspection in two ways:

- From the login page, click **One-Click Inspection** to enter.
- Directly access the login-free one-click inspection URL: http://${nKU IP or Domain}/ze/cs/inspection.

9.2 Monitoring Panel

9.2.1 General

The monitoring panel integrates with Grafana and has a variety of built-in monitoring panels that provide monitoring data of various resources such as CPU, memory, storage, network, etc. from multiple dimensions such as cluster, node, namespace, workload, and pod, which helps users to quickly grasp the resource usage of cluster, node, and pod, and improves the efficiency of operation and maintenance.

9.2.2 Viewing monitoring data

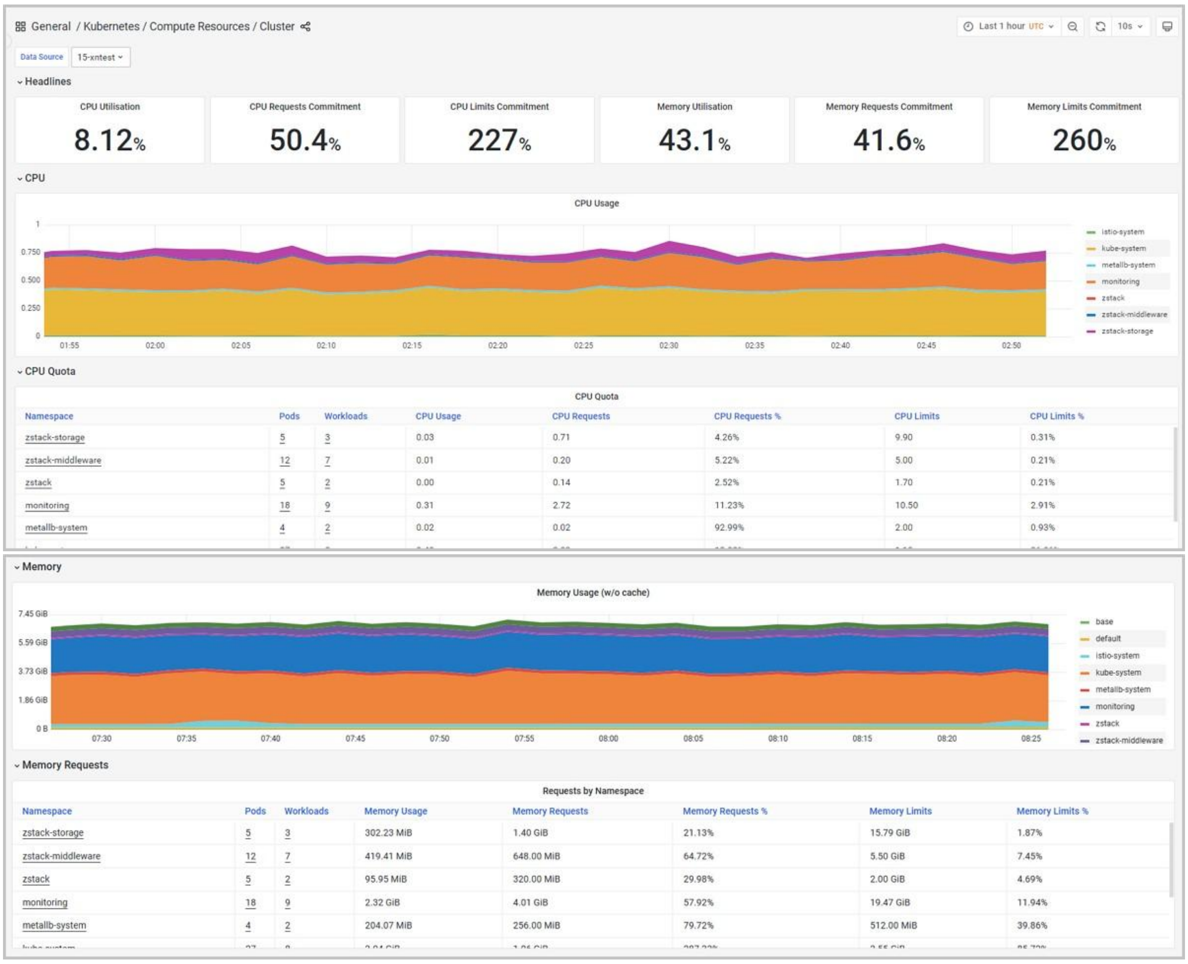
In the nKU main menu, click **Container Ops** > **Monitoring Panel** to enter the **Grafana** Dashboard, and click **General** in the upper-left corner of the page to display all the monitoring panels, and you can search for the monitoring panels by name or label, and the following describes the details of each monitoring panel:

- Kubernetes cluster resource monitoring

  Go to the **Kubernetes / Compute Resources / Cluster** monitoring panel to view Kubernetes cluster CPU utilization, memory utilization, and CPU and memory usage for each namespace.

  As shown in Figure 51: Kubernetes / Compute Resources / Cluster:

Figure 51: Kubernetes / Compute Resources / Cluster



CPU Utilisation	CPU Requests Commitment	CPU Limits Commitment	Memory Utilisation	Memory Requests Commitment	Memory Limits Commitment
8.12%	50.4%	227%	43.1%	41.6%	260%

CPU Quota

Namespace	Pods	Workloads	CPU Usage	CPU Requests	CPU Requests %	CPU Limits	CPU Limits %
zstack-storage	5	3	0.03	0.71	4.26%	9.90	0.31%
zstack-middleware	12	7	0.01	0.20	5.22%	5.00	0.21%
zstack	5	2	0.00	0.14	2.52%	1.70	0.21%
monitoring	18	9	0.31	2.72	11.23%	10.50	2.91%
metallb-system	4	2	0.02	0.02	92.99%	2.00	0.93%

Requests by Namespace

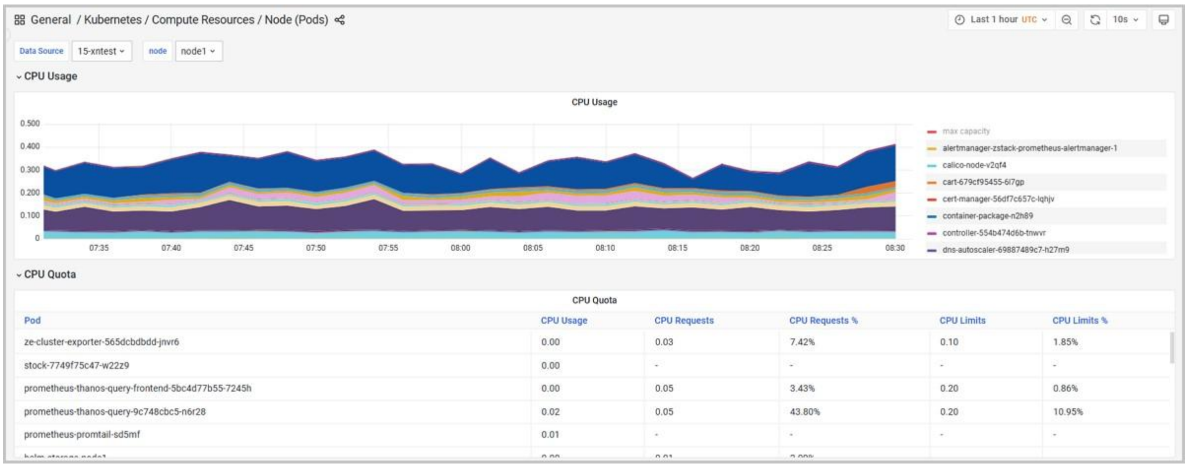
Namespace	Pods	Workloads	Memory Usage	Memory Requests	Memory Requests %	Memory Limits	Memory Limits %
zstack-storage	5	3	302.23 MiB	1.40 GiB	21.13%	15.79 GiB	1.87%
zstack-middleware	12	7	419.41 MiB	648.00 MiB	64.72%	5.50 GiB	7.45%
zstack	5	2	95.95 MiB	320.00 MiB	29.98%	2.00 GiB	4.69%
monitoring	18	9	2.32 GiB	4.01 GiB	57.92%	19.47 GiB	11.94%
metallb-system	4	2	204.07 MiB	256.00 MiB	79.72%	512.00 MiB	39.86%

- Resource monitoring for Pod on nodes

Go to **the Kubernetes / Compute Resources / Node(Pods)** monitoring panel to see the CPU and memory usage of Pod on a single node.

As shown in Figure 52: Kubernetes / Compute Resources / Node(Pods):

Figure 52: Kubernetes / Compute Resources / Node (Pods)



CPU Quota

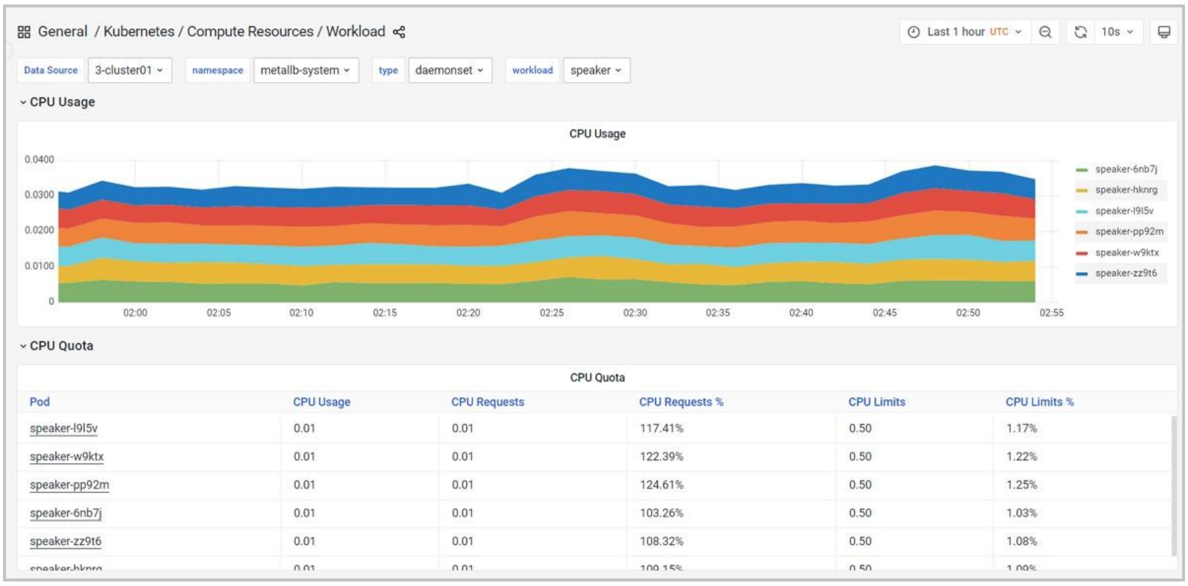
Pod	CPU Usage	CPU Requests	CPU Requests %	CPU Limits	CPU Limits %
ze-cluster-exporter-565dcbdbdd-jnvr6	0.00	0.03	7.42%	0.10	1.85%
stock-7749f75c47-w22z9	0.00	-	-	-	-
prometheus-thanos-query-frontend-5bc4d77b55-7245h	0.00	0.05	3.43%	0.20	0.86%
prometheus-thanos-query-9c748cbc5-n6r28	0.02	0.05	43.80%	0.20	10.95%
prometheus-promtail-sd5mf	0.01	-	-	-	-

- Workload Resource Monitoring

Go to the **Kubernetes / Compute Resources / Workload** monitoring panel to view individual workload resource usage.

As shown in Figure 53: Kubernetes / Compute Resources / Workload:

Figure 53: Kubernetes / Compute Resources / Workload



CPU Quota

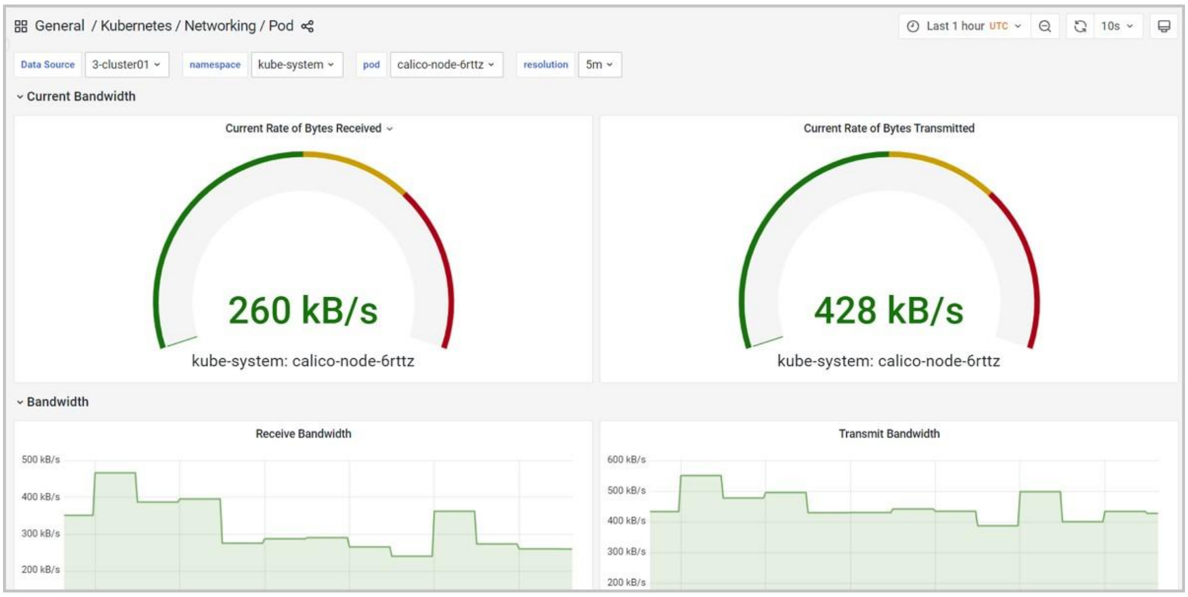
Pod	CPU Usage	CPU Requests	CPU Requests %	CPU Limits	CPU Limits %
speaker-l9l5v	0.01	0.01	117.41%	0.50	1.17%
speaker-w9ktx	0.01	0.01	122.39%	0.50	1.22%
speaker-pp92m	0.01	0.01	124.61%	0.50	1.25%
speaker-6nb7j	0.01	0.01	103.26%	0.50	1.03%
speaker-zz9t6	0.01	0.01	108.32%	0.50	1.08%
speaker-hknrg	0.01	0.01	109.15%	0.50	1.09%

- Pod Network Resource Monitoring

Go to the **Kubernetes / Networking / Pod** monitoring panel to view individual pod network resource usage.

As shown in Figure 54: Kubernetes / Networking / Pod:

Figure 54: Kubernetes / Networking / Pod



General / Kubernetes / Networking / Pod

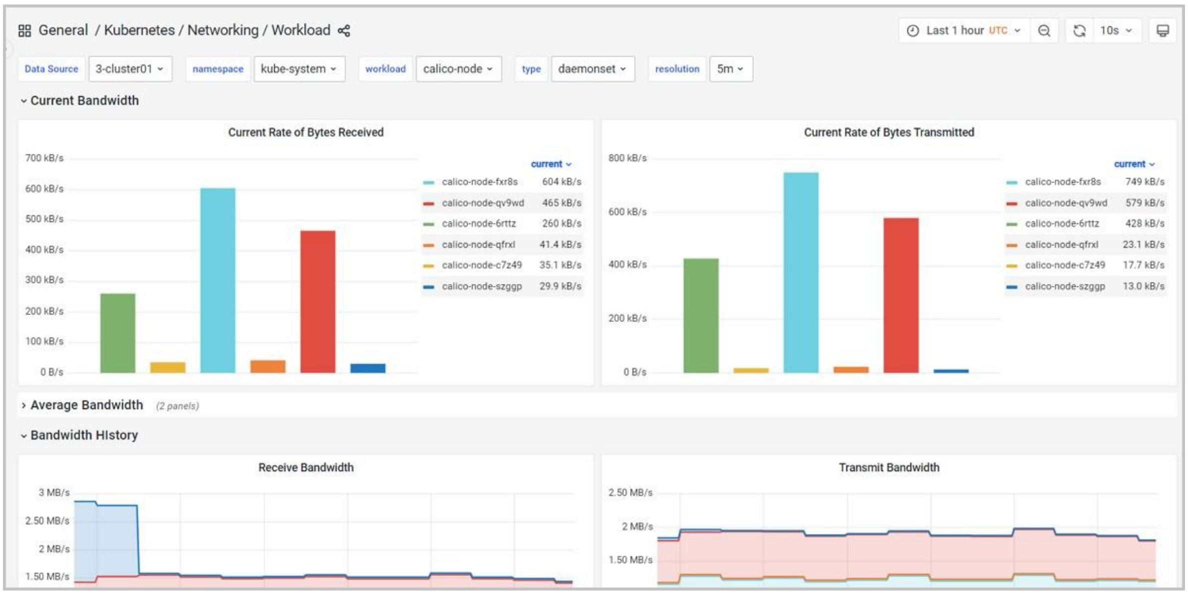
Data Source: 3-cluster01 · namespace: kube-system · pod: calico-node-6rttz · resolution: 5m

- Workload Network Resource Monitoring

Go to the **Kubernetes / Networking / Workload** monitoring panel to view individual workload network resource usage.

As shown in Figure 55: Kubernetes / Networking / Workload:

Figure 55: Kubernetes / Networking / Workload



General / Kubernetes / Networking / Workload

Data Source: 3-cluster01 · namespace: kube-system · workload: calico-node · type: daemonset · resolution: 5m

Current Rate of Bytes Received

	current
calico-node-fxr8s	604 kB/s
calico-node-qv9wd	465 kB/s
calico-node-6rttz	260 kB/s
calico-node-qfrxl	41.4 kB/s
calico-node-c7z49	35.1 kB/s
calico-node-szggp	29.9 kB/s

Current Rate of Bytes Transmitted

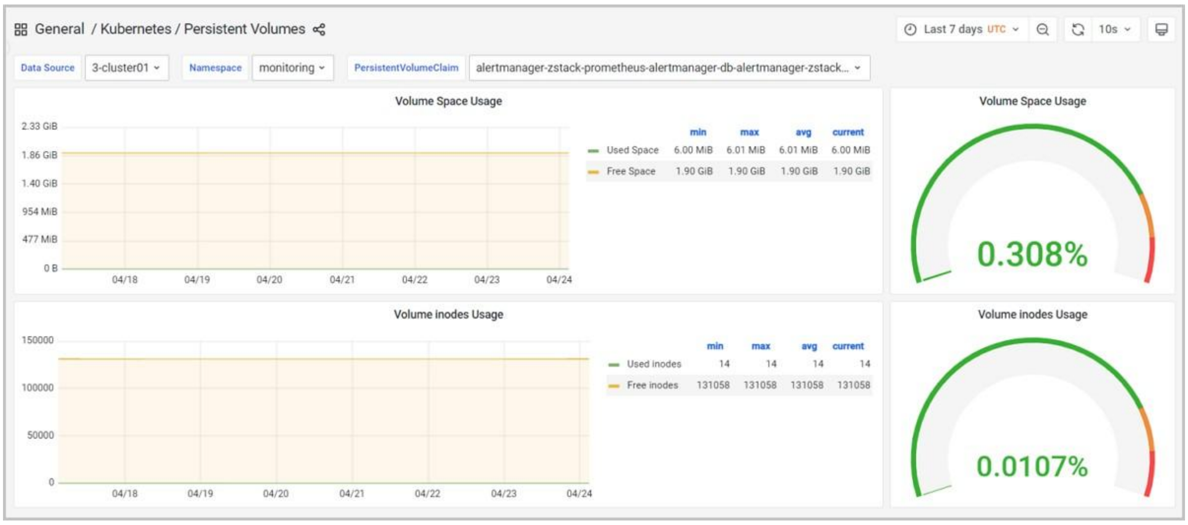
	current
calico-node-fxr8s	749 kB/s
calico-node-qv9wd	579 kB/s
calico-node-6rttz	428 kB/s
calico-node-qfrxl	23.1 kB/s
calico-node-c7z49	17.7 kB/s
calico-node-szggp	13.0 kB/s

- Pvc Resource Monitoring

  Go to the **Kubernetes / Persistent Volumes** monitoring panel to view individual pvc volume storage utilization and storage usage.

As shown in Figure 56: Kubernetes / Persistent Volumes:

Figure 56: Kubernetes / Persistent Volumes



General / Kubernetes / Persistent Volumes

Last 7 days UTC · 10s

Data Source: 3-cluster01 · Namespace: monitoring · PersistentVolumeClaim: alertmanager-zstack-prometheus-alertmanager-db-alertmanager-zstack...

Volume Space Usage

	min	max	avg	current
Used Space	6.00 MiB	6.01 MiB	6.01 MiB	6.00 MiB
Free Space	1.90 GiB	1.90 GiB	1.90 GiB	1.90 GiB

Volume Space Usage: 0.308%

Volume inodes Usage

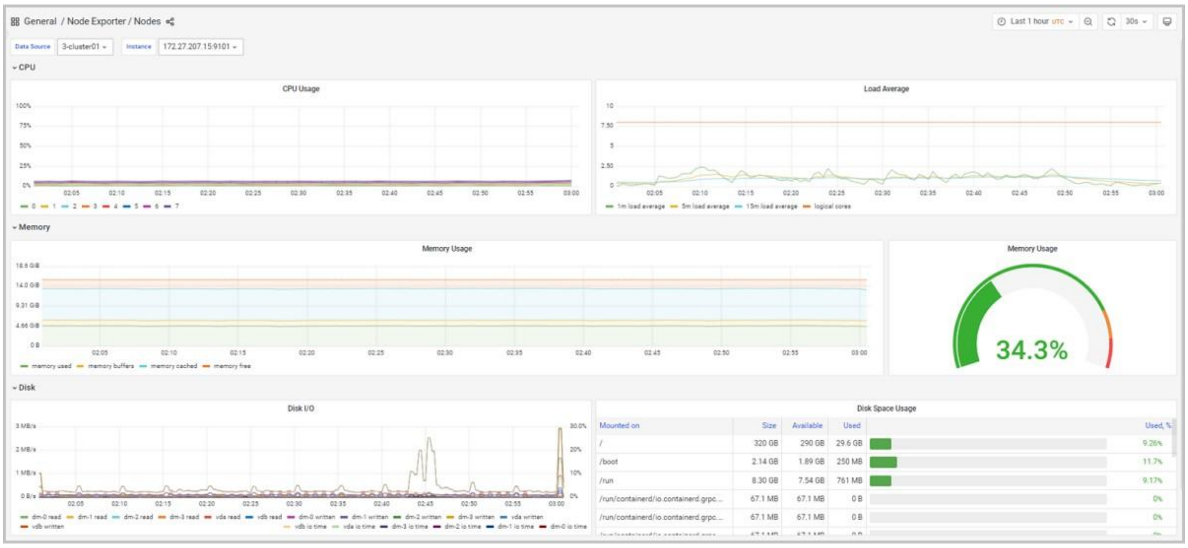
	min	max	avg	current
Used inodes	14	14	14	14
Free inodes	131058	131058	131058	131058

Volume inodes Usage: 0.0107%

- Node Monitoring Enter the **Nodes** monitoring panel to view the CPU, memory, disk, and network resource usage of a single node.

As shown in Figure 57: Nodes:

Figure 57: Nodes



General / Node Exporter / Nodes

Last 1 hour UTC · 30s

Data Source: 3-cluster01 · Instance: 172.27.207.15:9101

CPU

CPU Usage

Load Average

Memory

Memory Usage

Memory Usage: 34.3%

Disk

Disk I/O

Disk Space Usage

Mounted on	Size	Available	Used		Used, %
/	320 GB	290 GB	29.6 GB		9.26%
/boot	2.14 GB	1.89 GB	250 MB		11.7%
/run	8.30 GB	7.54 GB	761 MB		9.17%
/run/containerd/io.containerd.grpc....	67.1 MB	67.1 MB	0 B		0%
/run/containerd/io.containerd.grpc....	67.1 MB	67.1 MB	0 B		0%

9.3 Log panel

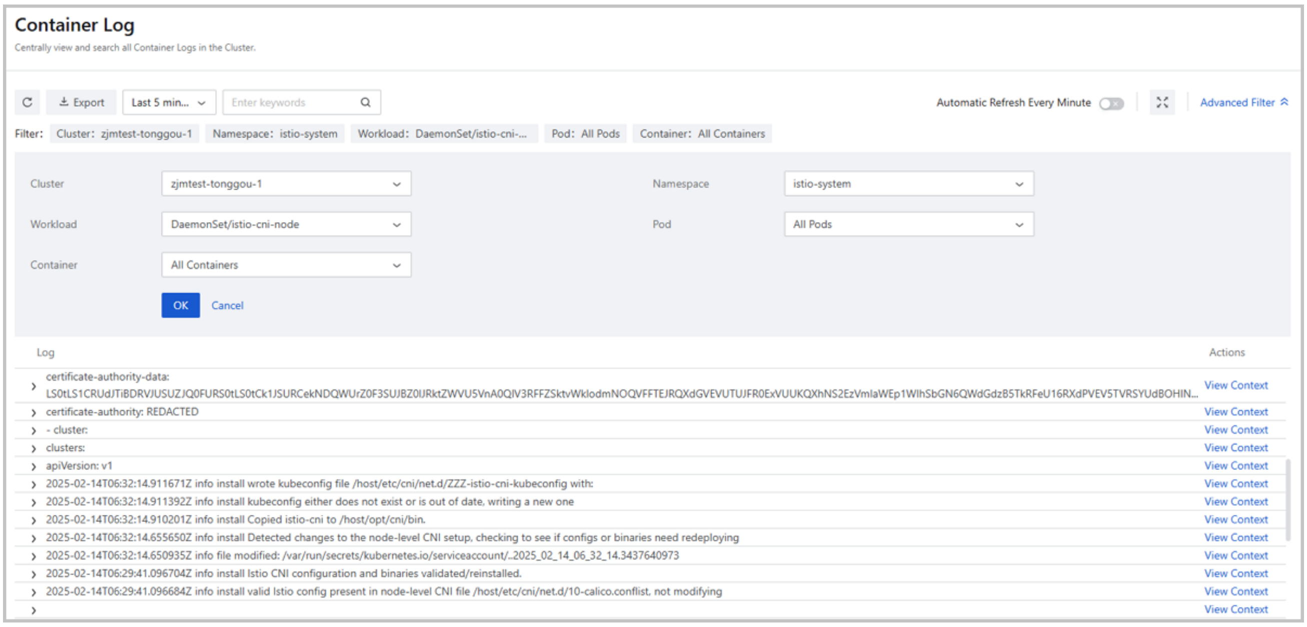
9.3.1 Container logs

In the nKU main menu, click **Container O&M**> **Log panel** > **Container Log** to access the **Container Log** interface.

This interface centralizes the display of all container logs in the cluster.

As shown in Figure 58: Container log:

Figure 58: Container Logs



Container Log

Centrally view and search all Container Logs in the Cluster.

Filter: Cluster：zjmtest-tonggou-1 Namespace：istio-system Workload：DaemonSet/istio-cni-... Pod：All Pods Container：All Containers

Cluster: zjmtest-tonggou-1

Namespace: istio-system

Workload: DaemonSet/istio-cni-node

Pod: All Pods

Container: All Containers

Log

```
certificate-authority-data:
LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURCekNDQWUrZ0F3SUJBZ0lJRktZWVU5VnA0QlV3RFFZSktvWklodmNOQVFFTEJRQXdGVEVUTUJFR0ExVUUKQXhNS2EzVmlaWEp1WlhSbGN6QWdGdzB5TkRFeU16RXdPVEV5TVRSYUdBOHIN...
certificate-authority: REDACTED
- cluster:
clusters:
apiVersion: v1
2025-02-14T06:32:14.911671Z info install wrote kubeconfig file /host/etc/cni/net.d/ZZZ-istio-cni-kubeconfig with:
2025-02-14T06:32:14.911392Z info install kubeconfig either does not exist or is out of date, writing a new one
2025-02-14T06:32:14.910201Z info install Copied istio-cni to /host/opt/cni/bin.
2025-02-14T06:32:14.655650Z info install Detected changes to the node-level CNI setup, checking to see if configs or binaries need redeploying
2025-02-14T06:32:14.650935Z info file modified: /var/run/secrets/kubernetes.io/serviceaccount/..2025_02_14_06_32_14.3437640973
2025-02-14T06:29:41.096704Z info install Istio CNI configuration and binaries validated/reinstalled.
2025-02-14T06:29:41.096684Z info install valid Istio config present in node-level CNI file /host/etc/cni/net.d/10-calico.conflist, not modifying
```

Users can perform search filtering and unified management of logs on this page:

- Supports selecting a time period to view container logs for the selected time period. Selectable time periods include: last 5 minutes, last 15 minutes, last 30 minutes, last 1 hour, last 6 hours, last 12 hours, last 24 hours, last 3 days, last 7 days, and customized, and the default is to show logs for the last 5 minutes.
- Supports searching logs by entering keywords.
- Supports searching logs by cluster, namespace, workload, pod, and container.
- Support for viewing logs in context. Users can search for context by keywords, which will be highlighted in the search results.
- Support for exporting logs and log contexts.

9.4 Alarm service

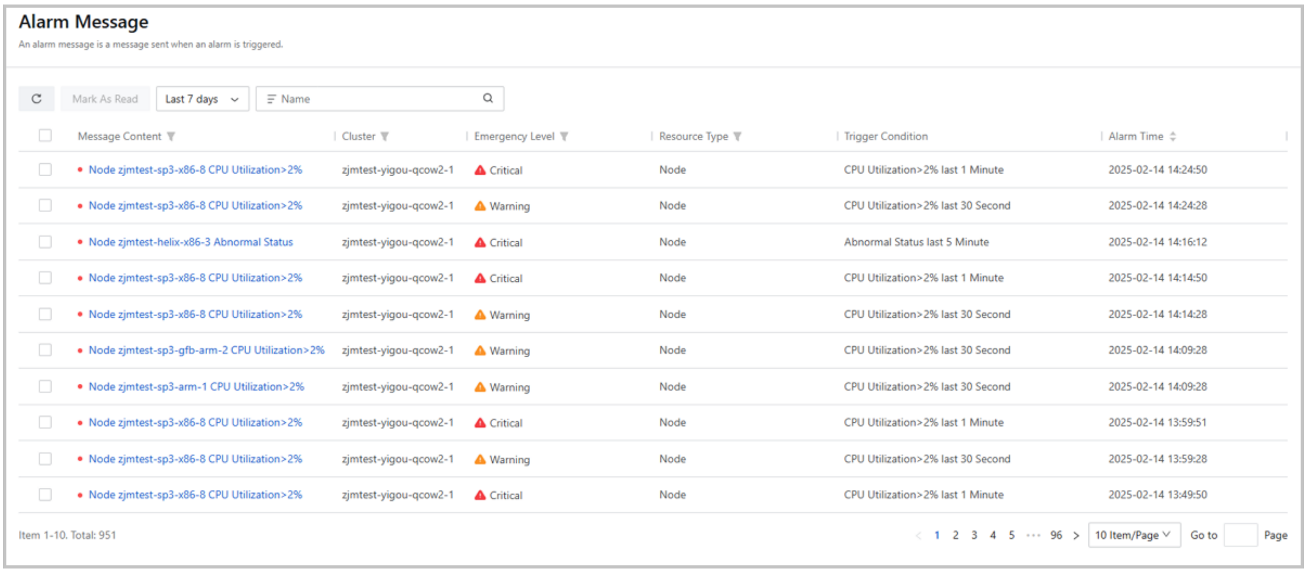
9.4.1 Alarm message

In the nKU main menu, click **Container O&M** > **Alarm Service** > **Alarm Message** to enter the **Alarm Message** interface.

This page centrally displays all alarm messages, including the message content, Emergency Level, Resource Type, Trigger Condition, and Alarm Time of the alarm message.

As shown in Figure 59: Alarm Messages:

Figure 59: Alarm Messages



Alarm Message

An alarm message is a message sent when an alarm is triggered.

□	Message Content	Cluster	Emergency Level	Resource Type	Trigger Condition	Alarm Time
□	Node zjmtest-sp3-x86-8 CPU Utilization>2%	zjmtest-yigou-qcow2-1	Critical	Node	CPU Utilization>2% last 1 Minute	2025-02-14 14:24:50
□	Node zjmtest-sp3-x86-8 CPU Utilization>2%	zjmtest-yigou-qcow2-1	Warning	Node	CPU Utilization>2% last 30 Second	2025-02-14 14:24:28
□	Node zjmtest-helix-x86-3 Abnormal Status	zjmtest-yigou-qcow2-1	Critical	Node	Abnormal Status last 5 Minute	2025-02-14 14:16:12
□	Node zjmtest-sp3-x86-8 CPU Utilization>2%	zjmtest-yigou-qcow2-1	Critical	Node	CPU Utilization>2% last 1 Minute	2025-02-14 14:14:50
□	Node zjmtest-sp3-x86-8 CPU Utilization>2%	zjmtest-yigou-qcow2-1	Warning	Node	CPU Utilization>2% last 30 Second	2025-02-14 14:14:28
□	Node zjmtest-sp3-gfb-arm-2 CPU Utilization>2%	zjmtest-yigou-qcow2-1	Warning	Node	CPU Utilization>2% last 30 Second	2025-02-14 14:09:28
□	Node zjmtest-sp3-arm-1 CPU Utilization>2%	zjmtest-yigou-qcow2-1	Warning	Node	CPU Utilization>2% last 30 Second	2025-02-14 14:09:28
□	Node zjmtest-sp3-x86-8 CPU Utilization>2%	zjmtest-yigou-qcow2-1	Critical	Node	CPU Utilization>2% last 1 Minute	2025-02-14 13:59:51
□	Node zjmtest-sp3-x86-8 CPU Utilization>2%	zjmtest-yigou-qcow2-1	Warning	Node	CPU Utilization>2% last 30 Second	2025-02-14 13:59:28
□	Node zjmtest-sp3-x86-8 CPU Utilization>2%	zjmtest-yigou-qcow2-1	Critical	Node	CPU Utilization>2% last 1 Minute	2025-02-14 13:49:50

Item 1-10. Total: 951

Users can filter and search alarm messages and manage them in a unified way on this page:

- Supports selecting a time period to view alarm messages for the selected time period. Selectable time periods include: last 12 hours, last 1 day, last 3 days, last 7 days, customized, and the default is to show the logs of the last 7 days.
- Supports searching alarm messages by resource name.
- Support for viewing alarm message details.
- Supports marking alarm messages as read.
- Support to adjust the number of alarm messages displayed on each page, the selectable values are: 10, 20, 50, 100, and support page turning operation.

9.4.2 Alarms

9.4.2.1 General

Alarms are used to monitor and respond to changes in the state of temporal data and push alarm messages to specified Endpoint through the notification service, and users can create alarms to monitor resources.

9.4.2.2 Creating an alarm

In the nKU main menu, click **Container O&M** > **Alarm Service** > **Alarms** to enter the **Alarms** interface. Click **Create Alarm** to bring up the **Create Alarm** screen.

You can refer to the following example to enter the appropriate contents:

- **Name**: Set the name of the alarm. Naming rules: length limit 1-128 characters, can only contain Chinese characters, English letters, numbers and the following 7 special characters: (-) (_) (.) (() () (:) (+)
- **Description**: optional, note alarm-related information
- **Resource Type**: Select the type of alarm resource, including: node, pod, pvc
- **Metric Item**: select alarm entries according to the selected resource type
- **Resource Scope**: Set the resource scope for the alarm application, support to apply to all resources or specified resources under the selected resource type:

  # Customize a single resource to create an alarm that monitors only the single resource under it, which meets the alarm conditions to trigger an alarm.

  # Customize batch resources to create an alarm which monitors the batch resources under it, and any one of them meets the alarm conditions to trigger an alarm.

  # Create an alarm for all resources, which monitors all resources of that resource type within the platform, and any one of those resources that meets the alarm conditions can trigger an alarm.

  Note: When the resource type is pod, it supports two resource selection methods: selecting by pod and selecting by workload. When selecting by pod, the selected pod will be monitored, and when selecting by workload, the pod under the selected workload will be monitored.

- **Alarm trigger rules**: set the alarm trigger rules

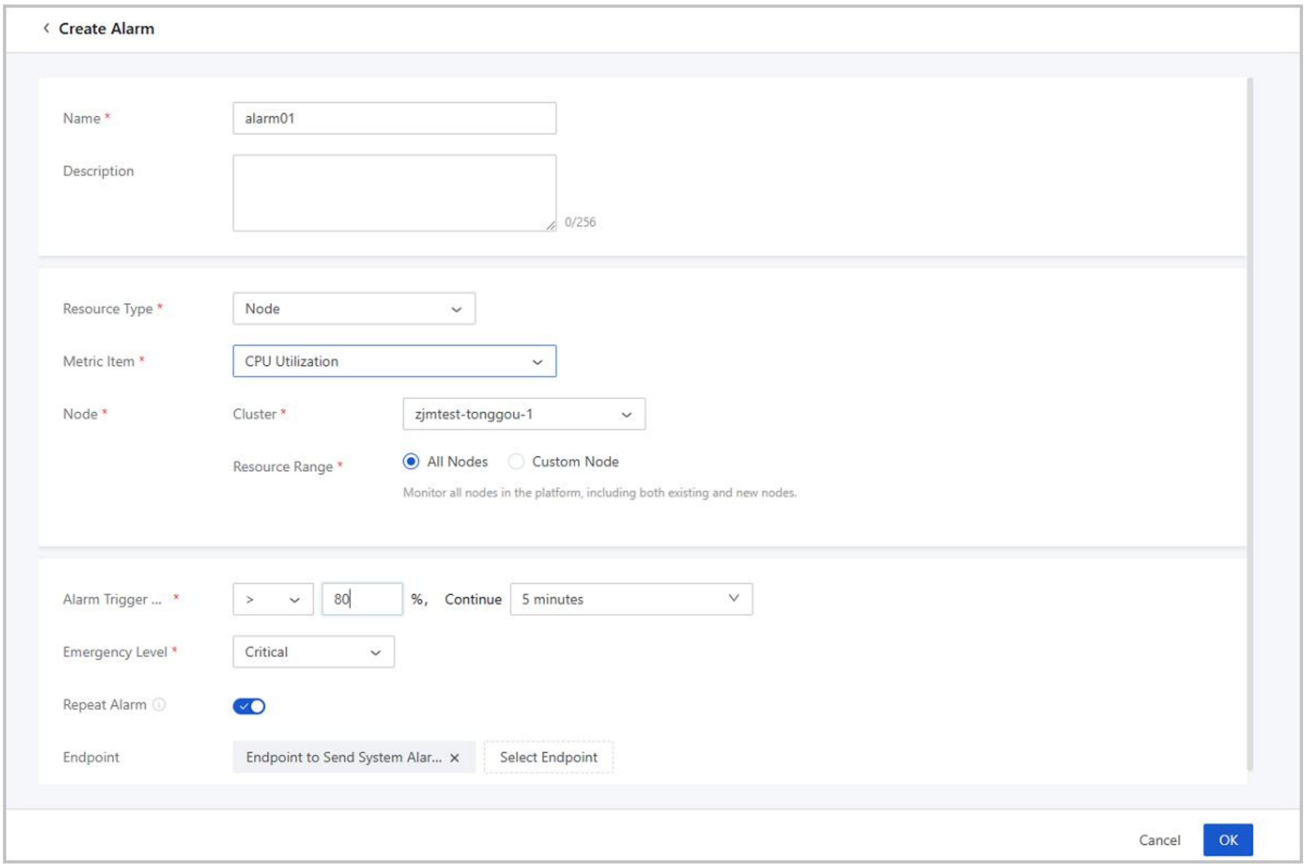
- **Emergency Level**: set the alarm level, support for Emergency, Serious, Tip three levels, different levels of alarm will send different levels of alarm messages
  - **Repeat Alarm**: Set whether to allow repeated alarm messages to be sent if the alarm resource is not restored in time after the first alarm

    Notes.

    - Turn on repeat alarms:
      - If an alarm mounts a single resource, and that resource triggers an alarm once and then continues to meet the alarm conditions, that alarm will alarm every 3 hours.
      - If an alarm mounts multiple resources, and one of the resources continues to meet the alarm conditions after triggering an alarm once, the alarm will alarm that resource every 3 hours, and if another resource meets the alarm conditions within the 3-hour interval, it will proceed directly to alarm the other resource.
    - Does not turn on repeat alarms:
      - If the alarm mounts a single resource, the resource triggers an alarm once and then continues to meet the alarm conditions, the alarm will no longer alarm, and if the resource returns to normal and then meets the alarm conditions again, it will trigger the alarm again.
      - If an alarm is mounted on multiple resources, and one of the resources triggers an alarm once and then continues to meet the alarm conditions, the alarm will no longer alarm that resource, and if there are other resources that meet the alarm conditions, an alarm for the other resources will be conducted.
- **Endpoint**: optional, default is the **system alarm Endpoint**, alarm messages will be sent to the specified Endpoint

As shown in Figure 60: Creating an Alarm:

Figure 60: Creating an Alarm



‹ Create Alarm

Name * alarm01

Description 0/256

Resource Type * Node

Metric Item * CPU Utilization

Node * Cluster * zjmtest-tonggou-1

Resource Range * ● All Nodes ○ Custom Node

Monitor all nodes in the platform, including both existing and new nodes.

Alarm Trigger ... * > 80 %, Continue 5 minutes

Emergency Level * Critical

Repeat Alarm ⓘ ✓

Endpoint: Endpoint to Send System Alar... ×

9.4.2.3 Managing alarms

In the nKU main menu, click **Container Ops** > **Alarms Services** > **Alarms** to enter **the Alarms** interface.

The alarm supports the following operations:

manipulate	descriptive
Create alarm	Create a new alarm.
Edit alarm	Modification of alarms, including alarm profile, resource range, alarm trigger rules, Alarm Emergency Level, Repeat Alarms, Endpoint.
Delete alarm	Delete the alarm. **Note:** After deletion, this alarm will no longer monitor resources and send alarm messages, so please proceed with caution.
Enabled alarm	Enable the deactivated alarm.

Disabled alarm	Deactivate the enabled alarms. **Note:** This alarm will still be monitored after deactivation. Deactivating the alarm still monitors the status of the alarm, but does not generate any alarm messages.

9.4.3 Endpoint

9.4.3.1 General

Endpoint refers to the way users get information about subscription topics, and the types of Endpoint include: system, email, WeChat, and DingTalk.

9.4.3.2 Creating a Endpoint

Creating a Endpoint is divided into the following three scenarios:

- Creating a email type Endpoint
- Creating an WeChat type Endpoint
- Creating a DingTalk type Endpoint

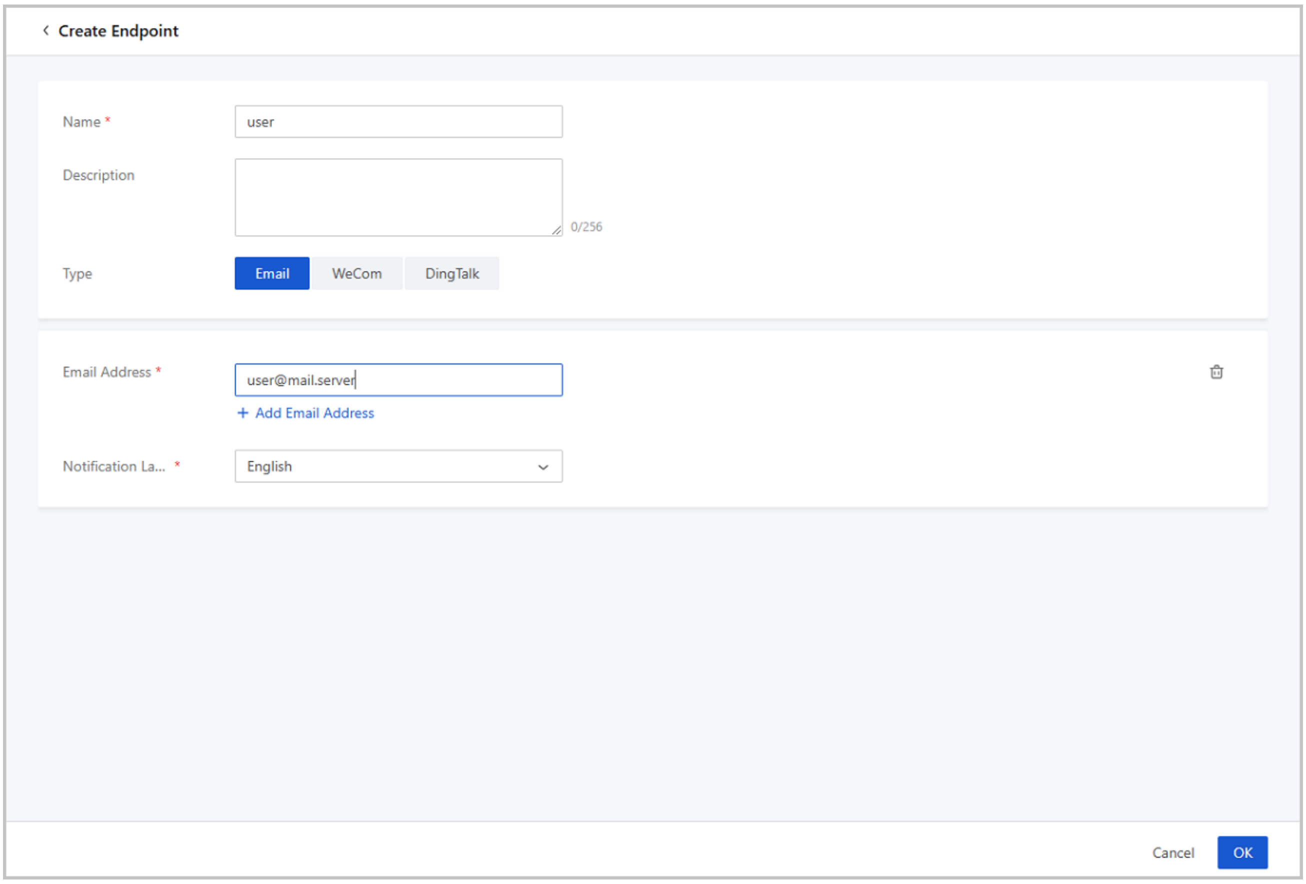
Creating a email type Endpoint

In the nKU main menu, click **Container O&M** > **Alarm Service** > **Endpoint** to enter the **Endpoint** interface, and click **Create Endpoint** to enter the **Create Endpoint** interface. You can refer to the following example to enter the corresponding content:

- **Name**: set the name of the Endpoint. Naming rules: length limit 1-128 characters, can only contain Chinese characters, English letters, numbers and the following 7 special characters: (-) (_) (.) (() ()) (:) (+)
- **Description**: optional, note information about the Endpoint
- **Type**: Select **Email**
- **Email address**: enter the e-mail address, support to add multiple e-mail addresses, can add up to 100
- **Notification language**: set the language to be used in the message sent to the Endpoint, support Simplified Chinese and English, the default is the same as the current platform language.

As shown in Figure 61: Creating a Endpoint:

Figure 61: Creating a Endpoint



‹ Create Endpoint

Name *: user

Description: 0/256

Type: Email | WeCom | DingTalk

Email Address *: user@mail.server

+ Add Email Address

Notification La... *: English

Cancel | OK

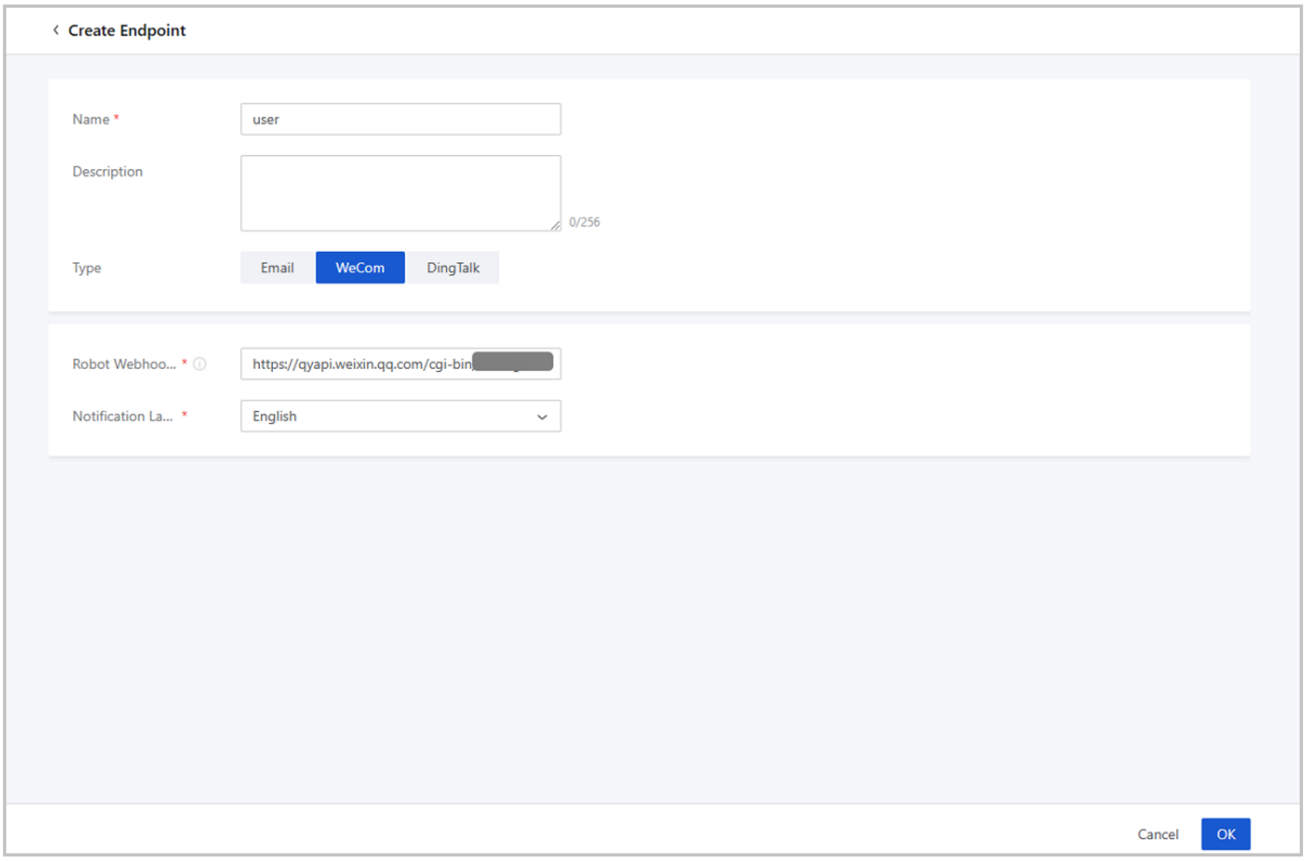
Creating an WeChat type Endpoint

In the nKU main menu, click **Container O&M** > **Alarm Service** > **Endpoint** to enter the **Endpoint** interface, and click **Create Endpoint** to enter the **Create Endpoint** interface. You can refer to the following example to enter the corresponding content:

- **Name**: set the name of the Endpoint. Naming rules: length limit 1~128 characters, can only contain Chinese characters, English letters, numbers and the following 7 special characters: (-) (_) (.) (() ()) (:) (+)
- **Description**: optional, note information about the Endpoint
- **Type**: Select WeChat
- **Robot Webhook Address**: Please create a group robot in the enterprise WeChat group and fill in the webhook address of the group robot here. After the creation, the platform alarm message will be pushed to the enterprise WeChat group through the group robot.
- **Notification language**: set the language to be used in the message sent to the Endpoint, support Simplified Chinese and English, the default is the same as the current platform language.

As shown in Figure 62: Creating a Endpoint:

Figure 62: Creating a Endpoint



‹ Create Endpoint

Name * user

Description 0/256

Type Email WeCom DingTalk

Robot Webhoo... * https://qyapi.weixin.qq.com/cgi-bin[redacted]

Notification La... * English

Cancel OK

Creating a DingTalk Endpoint

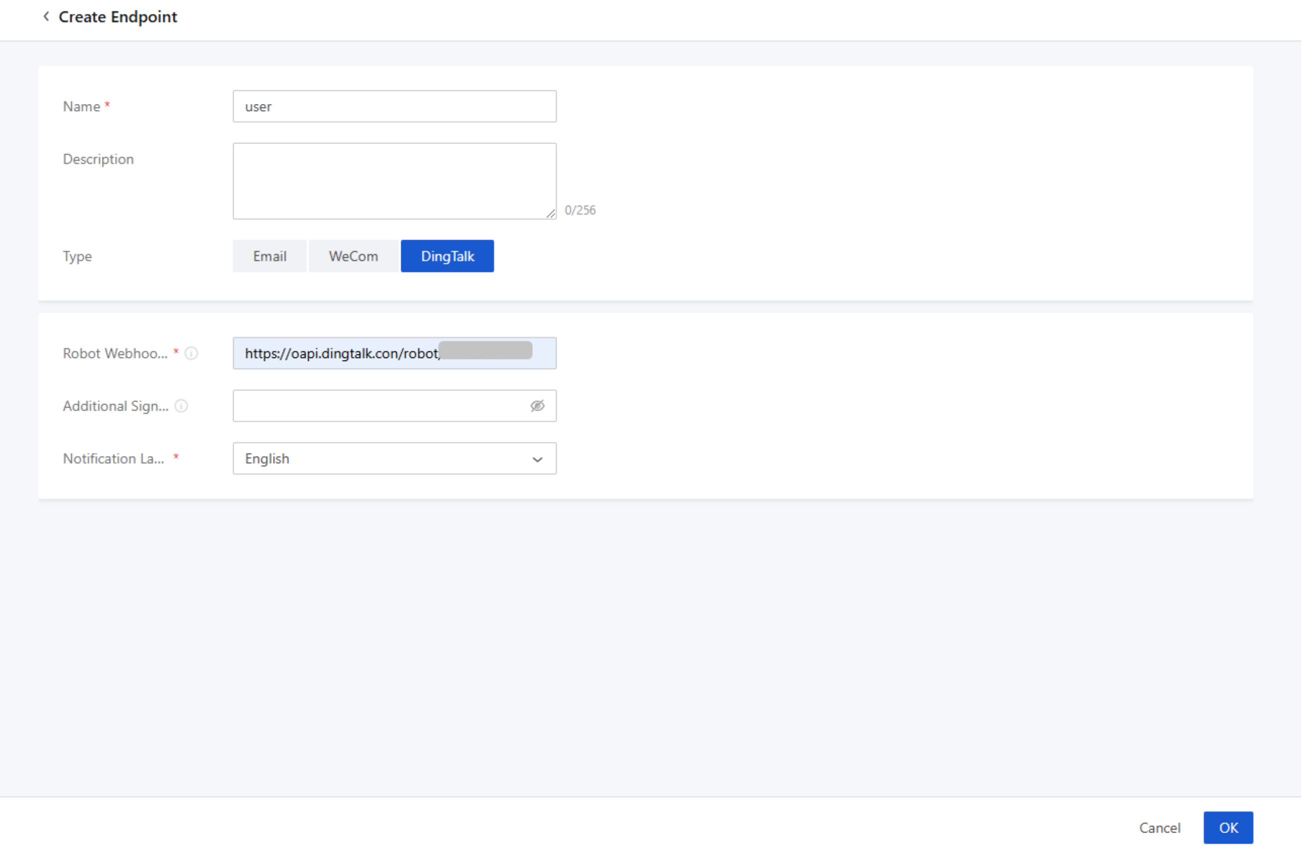
In the nKU main menu, click **Container O&M** > **Alarm Service** > **Endpoint** to enter the **Endpoint** interface, and click **Create Endpoint** to enter the **Create Endpoint** interface. You can refer to the following example to enter the corresponding content:

- **Name**: set the name of the Endpoint. Naming rules: length limit 1-128 characters, can only contain Chinese characters, English letters, numbers and the following 7 special characters: (-) (_) (.) (() ()) (:) (+)
- **Description**: optional, note information about the Endpoint
- **Type**: Select DingTalk
- **Robot Webhook Address**: please create a customized robot within the pinned group and fill in the robot webhook address here. After creation, the platform alarm message will be pushed to the pinned group through the robot

- **Additional Signature Key**: If you are using a version of Nail that requires security settings for group bots, please select the **Additional Signature** method and fill in the Additional Signature Key here.
- **Notification language**: set the language to be used in the message sent to the Endpoint, support Simplified Chinese and English, the default is the same as the current platform language.

As shown in Figure 63: Creating a Endpoint:

Figure 63: Creating a Endpoint



‹ Create Endpoint

Name *: user

Description: 0/256

Type: Email | WeCom | DingTalk

Robot Webhoo... *: https://oapi.dingtalk.con/robot/[redacted]

Additional Sign...:

Notification La... *: English

Cancel | OK

9.4.3.3 Managing Endpoint

In the nKU main menu, click **Container O&M** > **Alarm Service** > **Endpoint** to enter the **Endpoint** interface.

The Endpoint supports the following operations:

manipulate	descriptive
Creating a Endpoint	Creates a new Endpoint.

modifications	Modifies the Endpoint, and all contents except the Endpoint type can be modified.
removing	Delete the Endpoint. After deleting the Endpoint, the related subscription messages will not be sent to the Endpoint anymore, so please be careful.

9.5 Operation log

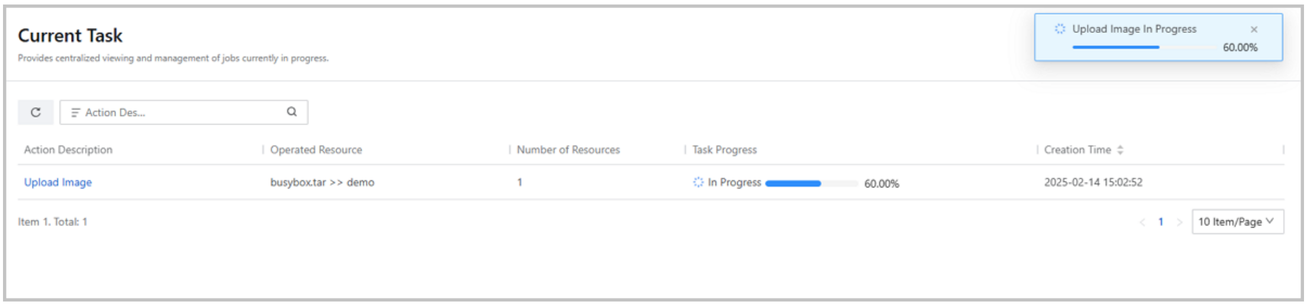
9.5.1 Current Task

In the nKU main menu, click **Container O&M** > **Operation Log** > **Current Task** to access the **Current Task** screen.

This page displays information about the operation that is currently in progress, including the operation description, operation resources, number of resources, job progress, operator, and creation time.

As shown in Figure 64: Current Task:

Figure 64: Current Task



Current Task

Provides centralized viewing and management of jobs currently in progress.

Upload Image In Progress 60.00%

Action Description	Operated Resource	Number of Resources	Task Progress	Creation Time
Upload Image	busybox.tar >> demo	1	In Progress 60.00%	2025-02-14 15:02:52

Item 1. Total: 1

Users can perform search filtering and unified management of current Task on this page:

- Supports searching for the current job by operation description, operation resource, and operator.
- Support for viewing job details.
- Supports adjusting the number of in-progress Task displayed on each page, with selectable values: 10, 20, 50, 100, and supports page turning.

9.5.2 Operation history

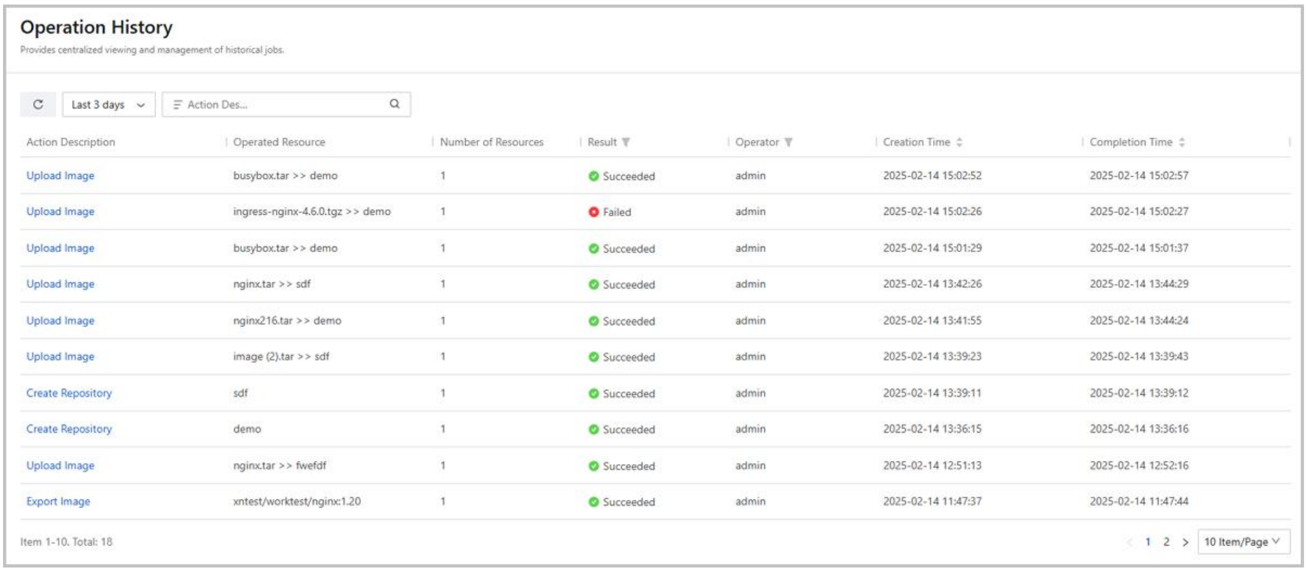
Operation History displays completed operations, providing centralized viewing and management.

In the nKU main menu, click **Container O&M** > **Operation Log** > **Operation History** to access the **Operation History** screen.

This page displays information about all completed Task, including job description, resource, number of resources, job result, operator, creation time, and completion time.

As shown in Figure 65: Operation History:

Figure 65: Operation History



Operation History

Provides centralized viewing and management of historical jobs.

Action Description	Operated Resource	Number of Resources	Result	Operator	Creation Time	Completion Time
Upload Image	busybox.tar >> demo	1	✓ Succeeded	admin	2025-02-14 15:02:52	2025-02-14 15:02:57
Upload Image	ingress-nginx-4.6.0.tgz >> demo	1	✗ Failed	admin	2025-02-14 15:02:26	2025-02-14 15:02:27
Upload Image	busybox.tar >> demo	1	✓ Succeeded	admin	2025-02-14 15:01:29	2025-02-14 15:01:37
Upload Image	nginx.tar >> sdf	1	✓ Succeeded	admin	2025-02-14 13:42:26	2025-02-14 13:44:29
Upload Image	nginx216.tar >> demo	1	✓ Succeeded	admin	2025-02-14 13:41:55	2025-02-14 13:44:24
Upload Image	image (2).tar >> sdf	1	✓ Succeeded	admin	2025-02-14 13:39:23	2025-02-14 13:39:43
Create Repository	sdf	1	✓ Succeeded	admin	2025-02-14 13:39:11	2025-02-14 13:39:12
Create Repository	demo	1	✓ Succeeded	admin	2025-02-14 13:36:15	2025-02-14 13:36:16
Upload Image	nginx.tar >> fwefdf	1	✓ Succeeded	admin	2025-02-14 12:51:13	2025-02-14 12:52:16
Export Image	xntest/worktest/nginx:1.20	1	✓ Succeeded	admin	2025-02-14 11:47:37	2025-02-14 11:47:44

Item 1-10. Total: 18

Users can perform search filtering and centralized management of historical Task from this page:

- Supports selecting a time period to view logs of completed operations for the selected time period. Selectable time periods include: the last 3 days, the last 7 days, the last 1 month, and customized, and the default is to display the logs of the last 3 days.
- Supports searching history logs by operation description, operation resource, operator, and job result.
- Supports sorting of completed operations by creation time/completion time.
- Supports adjusting the number of completed operation logs displayed on each page, with selectable values: 10, 20, 50, 100, and supports page-turning operation.

10 Cluster Management

10.1 Cluster Management

10.1.1 Cluster

10.1.1.1 General

Clusters mainly refer to Kubernetes clusters running container applications. nKU provides standardized, highly available and stable Kunernetes clusters and simplifies operations and maintenance operations such as deployment, expansion, upgrading, and high availability of clusters, so that users can focus on their own upper-tier container applications.

10.1.1.2 Creating Cluster

Preliminary

The hosts used to create the cluster need to meet the following conditions:

- Ensure that the nodes you want to use to create the cluster have a customized version of the Cloud operating system installed, and contact official technical support to obtain a customized version of the operating system for your current platform version.
- Ensure that each node acting as a management node mounts one 500G data disk each for storing monitoring/logging data for the cluster
- Ensure that the networks of the nodes to be used to create the cluster are interoperable and that all nodes are accessible to the management cluster

Creating a Cluster

In the nKU main menu, click **Cluster Management** > **Cluster Management** > **Cluster** to enter the **Cluster** interface. Click **Create Cluster** to enter the **Create Cluster** interface.

Creating a cluster is divided into the following three steps:

1. Node Configuration

   can be entered accordingly by referring to the following example:

   - **Management network and business network Reuse**: select whether the management network multiplexes the business network. When it is turned on, the management network multiplexes the business network, and the management cluster will access the nodes and manage the business clusters through the business network IP of the nodes, so please make sure that the management cluster is connected to the business network.

- **Cluster HA**: Select whether to turn on Cluster HA. The default is on. If it is off, the cluster HA does not support high availability, and only 1 management cluster node is needed. It is not recommended to turn off cluster HA for production environments.
- **Node Information**: Fill in information about the nodes to be used to create the cluster

# **Add Type**: The type for adding nodes include IP and IP range

# **Business network IPv4**: fill in the business network IPv4 address of the node. Fill in this field when the Add Type is IP.

# **Management network IPv4**: Fill in the IPv4 address of the management network of the node. Fill in this field when the Add Type is IP.The management cluster accesses and manages the business cluster nodes through the management network IPv4 address, so make sure that the management cluster is connected to this address. When the management network is multiplexed with the business network, the management network IPv4 address and the business network IPv4 address are the same.

# **Business network IPv4 Range**: fill in the business network IPv4 range of the node. Fill in this field when the Add Type is IP Range.

# **Management network IPv4**: Fill in the IPv4 range of the management network of the node. Fill in this field when the Add Type is IP Range. When the management network is multiplexed with the business network, the management network IPv4 range and the business network IPv4 range are the same.

Notes.

- Please make sure that the IP address you fill in is the same as the actual configuration of the machine.
- Make sure that the management network IPv4 address and the management cluster can be connected. If the management network is multiplexed with the business network, make sure that the business network IPv4 address is connectable to the management cluster.
- Make sure that the business network IPv4 addresses of the nodes are connectable to each other.

- The management network IPv4 addresses of different nodes cannot be duplicated; the business network IPv4 addresses of different nodes cannot be duplicated; if the management network is not multiplexed with the business network, the management network IPv4 address and the business network IPv4 address cannot be duplicated.
- If the Management Network is not reused for Business Network, make sure that the IPs in the Management Network IPv4 Range correspond one-to-one to the IPs in the Business Network IPv4 range. IPs in the same order belong to the same node. For example, the first IP in the Business Network IPv4 Range and the first IP in the Management Network IPv4 Range belong to the first node.
- Make sure that the network card name of the network card where the business network IPv4 address is located is the same for all 3 management nodes, and that the network card where the management network IPv4 address is located is the same for all 3 management nodes.

# **Node Name**: Set the name of the node. Naming rules: length limit 2-50 characters, can only contain lowercase letters, numbers and separator (-), must start with a lowercase letter and cannot end with a separator.When the Add Type is IP range, you need to specify a starting suffix. The node name consists of a base name and a suffix, and the suffix will automatically increment sequentially starting from the starting suffix, such as "-1/-2/-3" and so on.

Notes.

- The add node process sets the hostname of the host to the node name.
- The names used for the nodes in this batch cannot be duplicated.

# **Node Role**: Select the role of the node, optional: control plane, worker, GPU node. When Cluster HA is enabled, a cluster must have 3 control planes; when Cluster HA is disabled, a cluster must have 1 control plane.

Notes.

- The management node can also act as a compute node, then the business container pods can also be scheduled to the management node.

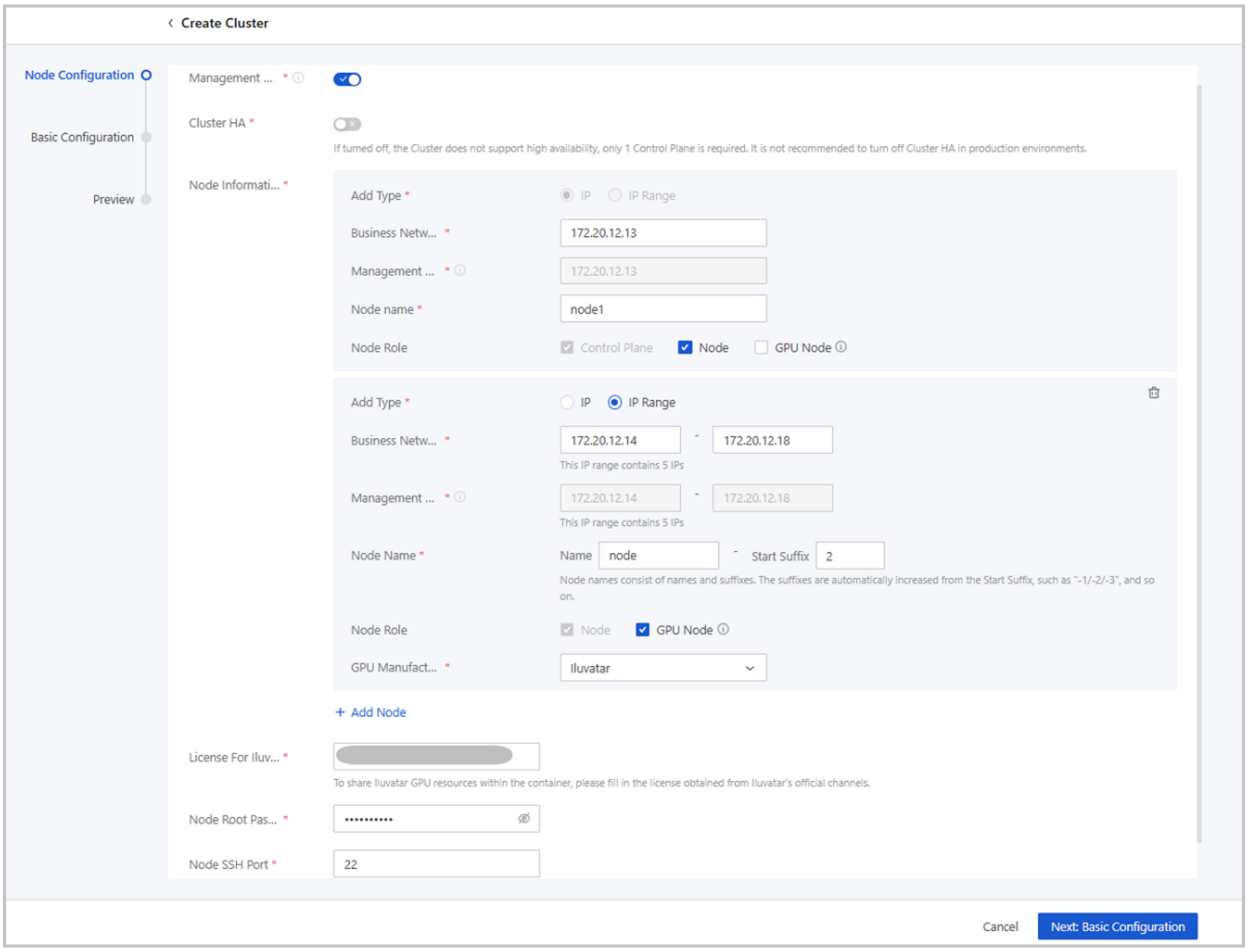
    - The GPU node will automatically install the corresponding components during the deployment process, please choose whether to install the machine as a GPU node according to the actual situation.
    - If the GPU node role is checked, please make sure that the GPU driver has been installed correctly on the node. If the driver is not installed, only the corresponding components will be installed and the node will not be set to the GPU node role.

  # **GPU Manufacturer**: Fill in this field when checking the GPU node for the node role, select the manufacturer of the GPU manufacturer on the node, and the system will install the GPU-related components of the corresponding manufacturer, currently supported: NVIDIA, Huawei,HYGON,Iluvatar.

- **License For Iluvatar GPU Virtualization:** When there are GPU nodes and the GPU manufacturer is Iluvatar, this field must be filled in. Enter the license for Iluvatar GPU virtualization. This license is required if you need to share Iluvatar GPUs in containers, and it can be obtained from Iluvatar official sources.
- **Node ROOT password**: Fill in the node root password to facilitate the process of adding nodes, SSH to the node to perform the installing steps, please ensure that the ROOT password of each node is the same.
- **Node SSH port**: Fill in the port used by the SSH node to facilitate the process of adding nodes, SSH to the node to perform the installing steps, please make sure that the port of each node can be connected.

As shown in Figure 66: Creating a Cluster-Node Configuration:

Figure 66: Creating a Cluster-Node Configuration



‹ Create Cluster

Management ... * ●

Cluster HA * ○

If turned off, the Cluster does not support high availability, only 1 Control Plane is required. It is not recommended to turn off Cluster HA in production environments.

Node Informati... *

Add Type * ● IP ○ IP Range

Business Netw... * 172.20.12.13

Management ... * 172.20.12.13

Node name * node1

Node Role ☑ Control Plane ☑ Node □ GPU Node

Add Type * ○ IP ● IP Range

Business Netw... * 172.20.12.14 - 172.20.12.18

This IP range contains 5 IPs

Management ... * 172.20.12.14 - 172.20.12.18

This IP range contains 5 IPs

Node Name * Name node - Start Suffix 2

Node names consist of names and suffixes. The suffixes are automatically increased from the Start Suffix, such as "-1/-2/-3", and so on.

Node Role ☑ Node ☑ GPU Node

GPU Manufact... * Iluvatar

License For Iluv... * [redacted]

To share Iluvatar GPU resources within the container, please fill in the license obtained from Iluvatar's official channels.

Node Root Pas... * ••••••••••

Node SSH Port * 22

2. Base Configuration

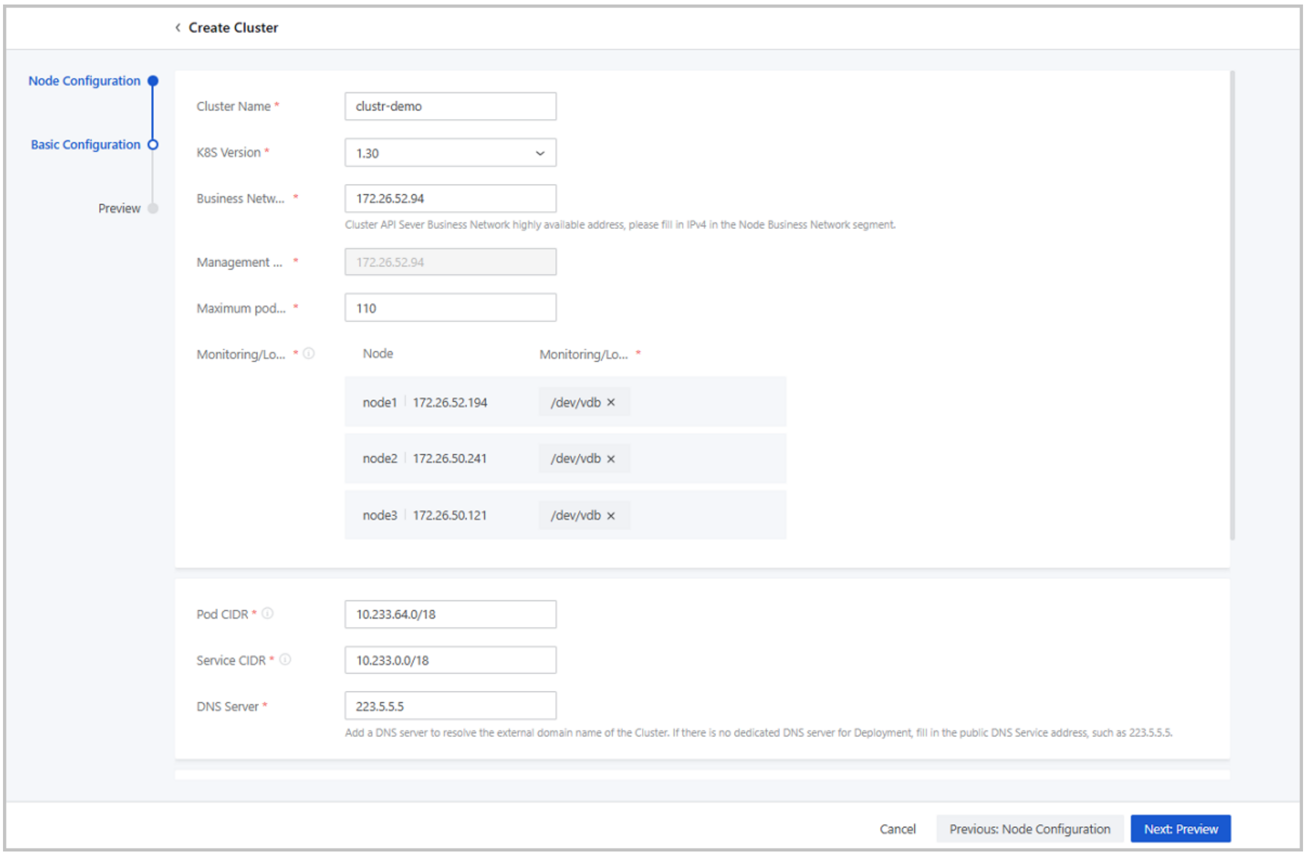
can be entered accordingly by referring to the following example:

- **Cluster Name**: Set the name of the cluster. Naming rules: length limit 2-50 characters, can only contain lowercase letters, numbers and separator (-), must start with a lowercase letter, can not end with a separator
- **K8S version**: select the Kubernetes version of the cluster, you can choose 1.24, 1.30 version
- **Business network VIP**: Fill in the highly available address of the cluster API Server in the business network, please fill in the IPv4 address in the business network segment.
- **Management Network VIP**: Fill in the cluster HA address of the cluster API Server in the management network, please fill in the IPv4 address in the management network segment. If the management network is multiplexed with the business network, the VIP of the management network is the same as the VIP of the business network.
- **Maximum pods per node**: fill in the maximum number of container pods allowed to run on each node, the default is 110, and you can only fill in an integer of 50~1000

- **Monitoring/logging data disk**: Select the disk used for storing cluster monitoring data and container log data, make sure it is loaded on each management cluster node.
- **Pod CIDR**: Fill in the network segment used by the pod in the cluster, which is used to assign the IP address used by the pod, and the default is 10.233.64.0/18. If you modify the default pod segment, please avoid overlapping with the intranet segment and other segments used in the creation of this cluster.
- **Service CIDR**: Fill in the service CIDR in the cluster, used to assign the ClusterIP for the service, the default is 10.233.0.0/18. If you modify the default service CIDR, please avoid overlapping with the Intranet CIDR and other CIDRs used in the creation of the cluster.
- **DNS server**: Add a DNS server to provide DNS resolution service to resolve domain names external to the cluster. If you have not deployed a dedicated DNS server, you can fill in the public DNS server address, such as 223.5.5.5.
- **Microservices governance**: choose whether to enable microservices governance, when enabled, the deployment of microservices governance related components in the cluster, support the use of microservices governance related functions in the cluster
- **Container Image Disk**: optional, select the Image data disk used to store the Image data. If no container images disk is set, the Image data is stored in the system disk by default. When more and larger Images are used, it is recommended to load a special volume for the node to store the Images, so as to avoid the Image data from filling up the system disk, resulting in system abnormalities.

As shown in Figure 67: Creating a Cluster-Basic Configuration:

Figure 67: Creating a Cluster-Basic Configuration



Cluster Name * clustr-demo

K8S Version * 1.30

Business Netw... * 172.26.52.94

Cluster API Sever Business Network highly available address, please fill in IPv4 in the Node Business Network segment.

Management ... * 172.26.52.94

Maximum pod... * 110

Monitoring/Lo... *

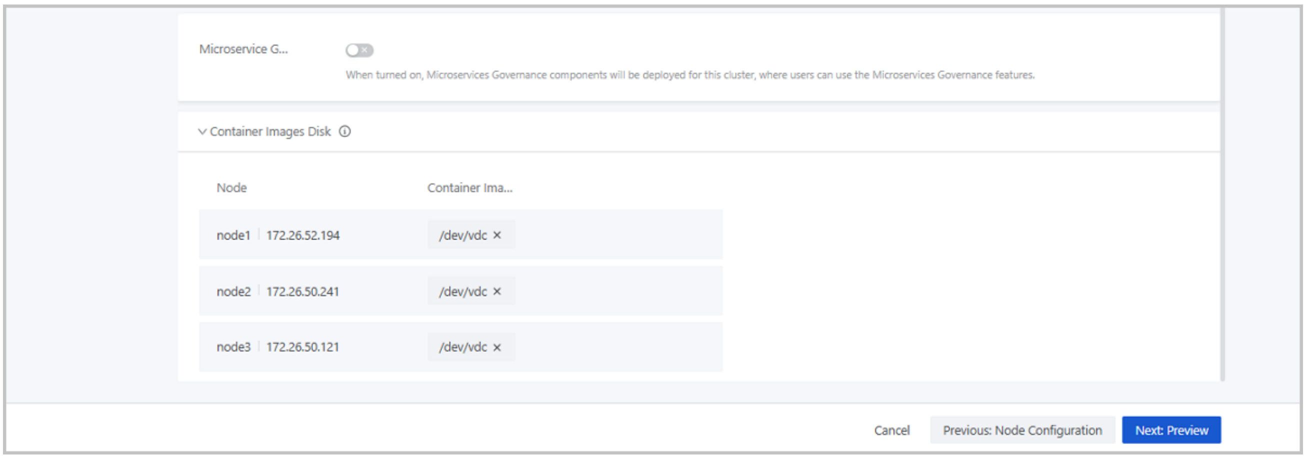
Node	Monitoring/Lo... *
node1 | 172.26.52.194	/dev/vdb ×
node2 | 172.26.50.241	/dev/vdb ×
node3 | 172.26.50.121	/dev/vdb ×

Pod CIDR * 10.233.64.0/18

Service CIDR * 10.233.0.0/18

DNS Server * 223.5.5.5

Add a DNS server to resolve the external domain name of the Cluster. If there is no dedicated DNS server for Deployment, fill in the public DNS Service address, such as 223.5.5.5.



Microservice G...

When turned on, Microservices Governance components will be deployed for this cluster, where users can use the Microservices Governance features.

Container Images Disk

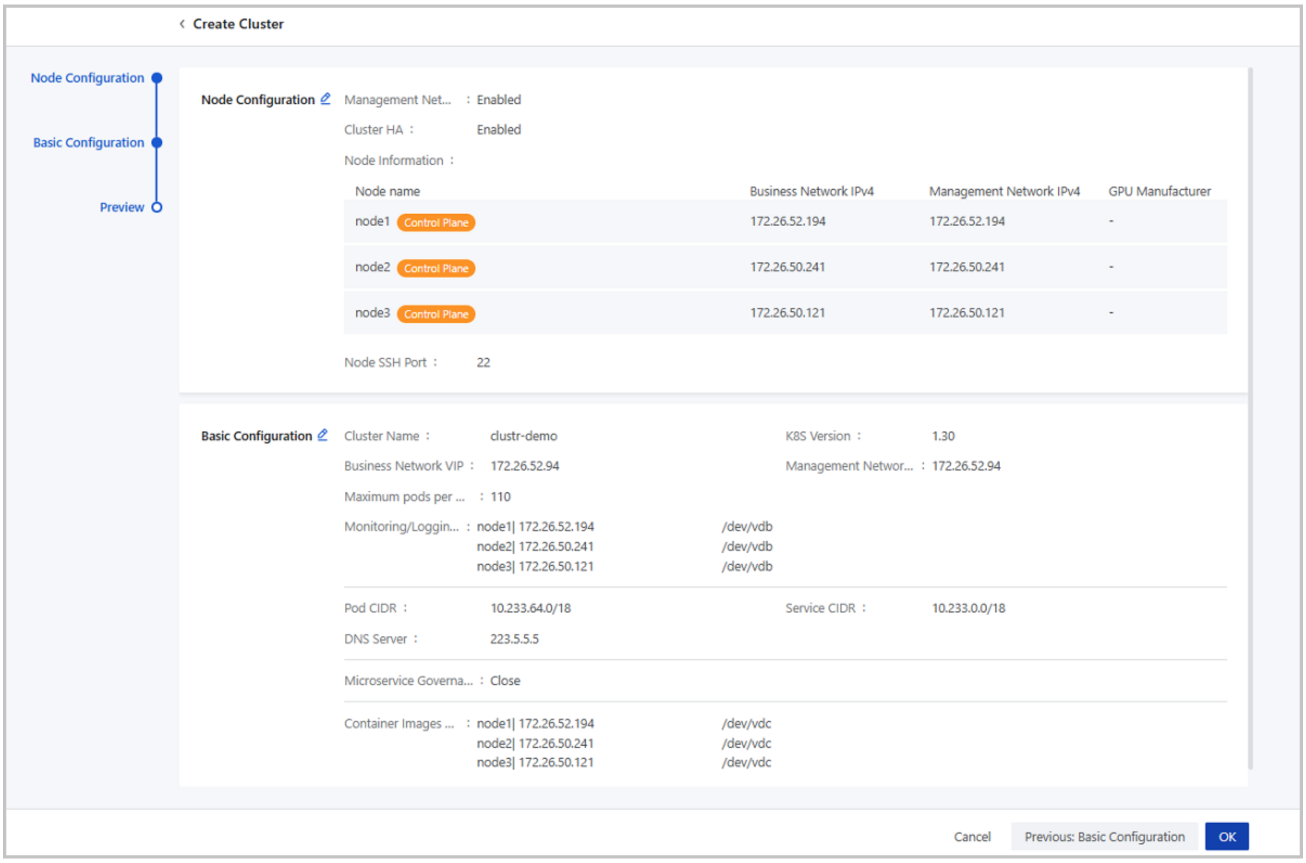
Node	Container Ima...
node1 | 172.26.52.194	/dev/vdc ×
node2 | 172.26.50.241	/dev/vdc ×
node3 | 172.26.50.121	/dev/vdc ×

3. Preview

Confirm the configuration information of the cluster ,support jump to modify.

As shown in Figure 68: Creating a Cluster-Preview:

Figure 68: Creating a Cluster-Preview



‹ Create Cluster

Node Configuration

Management Net... : Enabled

Cluster HA : Enabled

Node Information :

Node name	Business Network IPv4	Management Network IPv4	GPU Manufacturer
node1 Control Plane	172.26.52.194	172.26.52.194	-
node2 Control Plane	172.26.50.241	172.26.50.241	-
node3 Control Plane	172.26.50.121	172.26.50.121	-

Node SSH Port : 22

Basic Configuration

Cluster Name : clustr-demo

K8S Version : 1.30

Business Network VIP : 172.26.52.94

Management Networ... : 172.26.52.94

Maximum pods per ... : 110

Monitoring/Loggin... : node1| 172.26.52.194 /dev/vdb
node2| 172.26.50.241 /dev/vdb
node3| 172.26.50.121 /dev/vdb

Pod CIDR : 10.233.64.0/18

Service CIDR : 10.233.0.0/18

DNS Server : 223.5.5.5

Microservice Governa... : Close

Container Images ... : node1| 172.26.52.194 /dev/vdc
node2| 172.26.50.241 /dev/vdc
node3| 172.26.50.121 /dev/vdc

After clicking **OK**, the system will start to create the cluster, and the user clicks **Cluster Management** > **Cluster Management** > **Cluster** to enter the **cluster** interface and view the creation of the cluster.

10.1.1.3 Management cluster

In the nKU main menu, click **Cluster Management** > **Cluster Management** > **Clusters** to access the cluster interface.

The cluster supports the following operations:

manipulate	descriptive
Create Cluster	Create a new cluster.
Reinstall cluster	Reinstall the cluster that failed to create.
Cluster Details	Enter the cluster details page to view the basic information of the cluster, resource usage, kubeconfig, and cluster operation logs, including: logs of creating clusters, adding nodes, removing nodes, and so on.

functional component	Go to the cluster details page and click **Functional Components** to view the functional components supported by the platform and the deployment of the components in the cluster, and deploy the components as needed.
integrated cluster	Integrate existing Kubernetes clusters into the platform for management. **Notes.** **1.** For Kubernetes clusters that support kubernetes version 1.24-1.30, make sure that the platform management cluster is connected to the network of the managed cluster. **2.** The following features are not supported by the integrated cluster at this time: • Dashboard resource statistics are not supported. • Saving a container as an image is not supported. • Microservices governance is not supported. • Viewing of monitoring panel, log panel and alarm service is not supported. • Adding/removing nodes is not supported. • Adding and managing external networks is not supported. • GPU resource management scheduling is not supported
Cancel integrated	Cancellation of a integrated cluster, which, when canceled, will not allow the cluster to continue to be used and managed on the current platform.
Delete Cluster	Remove the cluster. **Note:** Deleting a cluster will synchronize the deletion of all resources within the current cluster and clear the cluster configuration on the nodes. The cleared resources and configurations are not recoverable, so please proceed with caution.

10.1.1.4 Cluster Access Configuration

Support for connecting to a Kubernetes cluster via kubectl can be found in the following steps:

1. Download and install the latest version of the kubectl client, see Installing and deploying the kubectl client for details.
2. Check out the cluster Kubeconfig.
   - In the nKU main menu, click **Personal Center** > **KubeConfig** in the upper right corner to view the current cluster KubeConfig.
     - Administrators can also go directly to the cluster details page to view the cluster KubeConfig.

   As shown in Figure 69: Kubeconfig-Personal Center, Figure 70: Kubeconfig-Cluster Details page:

   Figure 69: Kubeconfig-Personal Center

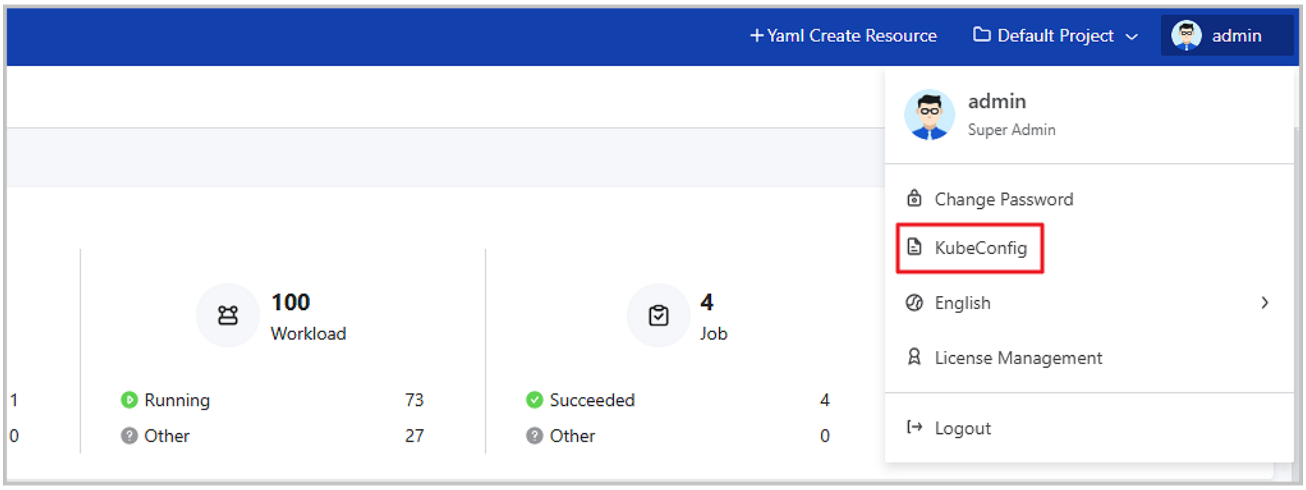
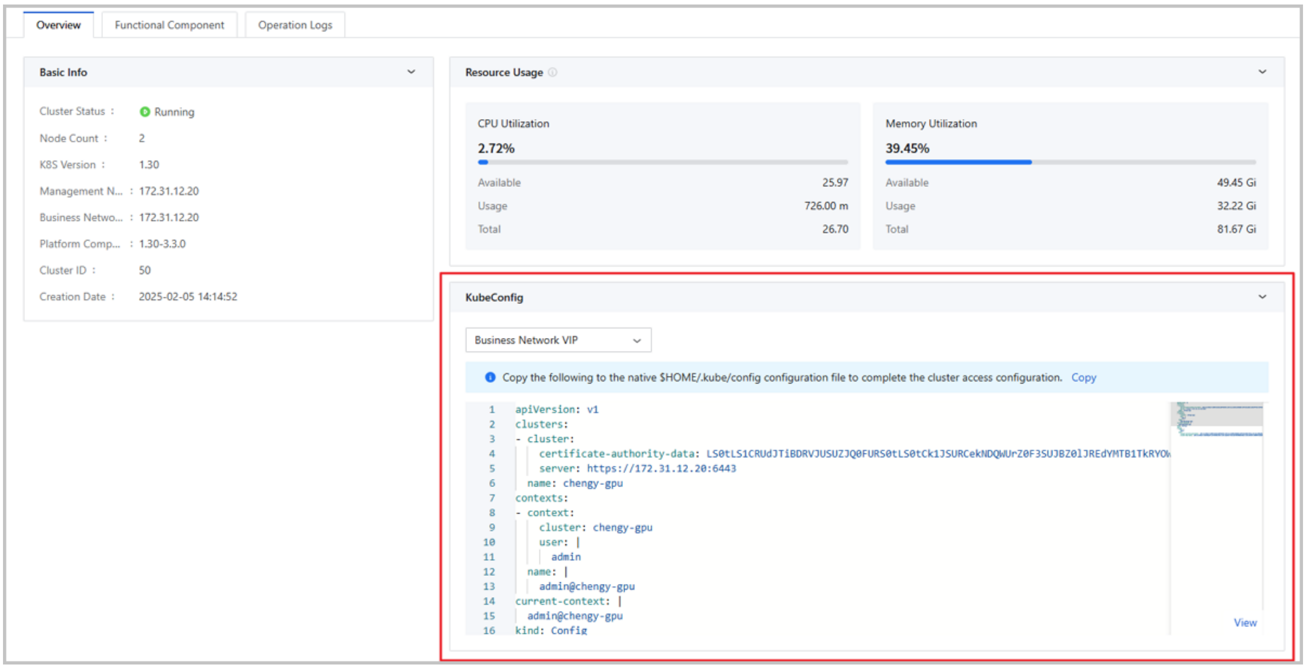


Figure 70: Kubeconfig-Cluster Details Page



Basic Info

Cluster Status : Running

Node Count : 2

K8S Version : 1.30

Management N... : 172.31.12.20

Business Netwo... : 172.31.12.20

Platform Comp... : 1.30-3.3.0

Cluster ID : 50

Creation Date : 2025-02-05 14:14:52

Resource Usage

CPU Utilization	2.72%
Available	25.97
Usage	726.00 m
Total	26.70

Memory Utilization	39.45%
Available	49.45 Gi
Usage	32.22 Gi
Total	81.67 Gi

KubeConfig

Business Network VIP

Copy the following to the native $HOME/.kube/config configuration file to complete the cluster access configuration. Copy

```
apiVersion: v1
clusters:
- cluster:
    certificate-authority-data: LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURCekNDQWUrZ0F3SUJBZ0lJREdYMTB1TkRYOW
    server: https://172.31.12.20:6443
  name: chengy-gpu
contexts:
- context:
    cluster: chengy-gpu
    user: |
      admin
  name: |
    admin@chengy-gpu
current-context: |
  admin@chengy-gpu
kind: Config
```

3. Depending on the network of the cluster, you can choose to use the business network VIP or management network VIP to access the cluster, select the corresponding KubeConfig, copy the content in **Kubeconfig** and paste it into the $HOME/.kube/config directory of the kubectl user terminal to complete the access configuration.

10.1.1.5 Cluster Advanced Setting

In the nKU main menu, click **Cluster Management** > **Cluster Management** > **Cluster** to access the cluster interface. Click on the cluster name to enter the cluster details page, then click Advanced Settings to access the Advanced Settings subpage.

The cluster supports the following advanced settings:

Name	Descriptive
Container Log Retention Period	• Default is 15 days, used to set the duration for which container logs are retained. Container logs exceeding this duration will be deleted. • It is recommended to set this period reasonably based on the Container Log Data Disk Size and the size of the generated container logs per day. If the total size of logs to be retained exceeds Container Log Data Disk Size before the retention period expires, it may cause abnormal log service.

Container Log Data Disk Size	• Default is 250 Gi, used for setting the size of the PVC for container logs. • It is recommended to set according to the size of the generated container logs per day and the number of days the logs need to be retained. If the total size of logs before the retention period exceeds the size of the log data disk, it may cause log service exceptions.
Restore Container Log Writing	**1.** If the container log write forbidden alarm is triggered for a cluster, the platform will forbid the container logs from being reported. In this case, expand the Container Log Data Disk Size or shorten the Container Log Retention Period to trigger automatic cleanup of the generated log files. **2.** If the utilization of theContainer Log Data Disk is less than the threshold of the container log write forbidden alarm, the container log data write forbidden will be lifted and the container logs will be reported as usual.

10.1.1.6 Functional Component

The function component list displays the function components supported by the platform and their deployment status in the cluster. Users can deploy components as needed. Currently, the following 4 function components are supported: Microservice Governance Component, ZStone-RBD-CSI, ZCE-iSCSI-CSI, ZCE-NFS-CSI.

10.1.1.6.1 Microservice Governance Component

The Microservice Governance Component includes all components required for microservice governance such as istiod and visualization plugins. It provides microservice governance capabilities, and relevant microservice governance functions can be used in the cluster after deploying this component.

Deploy the Microservice Governance Component

In the main menu of nKU, click **Cluster Management > Cluster Management > Cluster** to enter the **Cluster** interface. Click the cluster name to enter the Cluster Details page, then click **Function Components** to access the Function Components sub-page. Click the **Deploy** button next to the Microservice Governance Component in the function component list to open the Deploy Microservice Governance Component pop-up window.

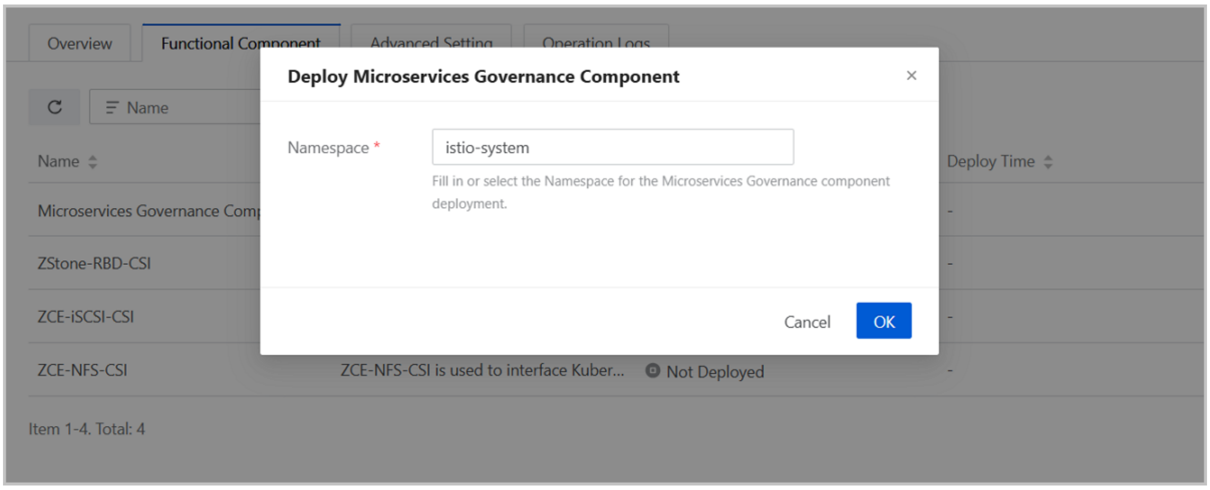
In the pop-up window, fill in or select the namespace for deploying the Microservice Governance Component, then click **Confirm**

Note:

- The default namespace is istio-system. If a new namespace is entered, it will be created automatically, and the Microservice Governance Component will be deployed in this namespace.
- Ensure no istio-related components are deployed under the filled-in namespace to avoid conflicts.
- If the deployment of the Microservice Governance Component fails, you can redeploy it, but the namespace cannot be changed at this time.
- Clusters with the Microservice Governance Component successfully deployed cannot be redeployed.

As shown in Figure: Deploy Microservice Governance Component:

Figure: Deploy Microservice Governance Component



ZCE-NFS-CSI is used to interface Kuber... Not Deployed

Deploy Microservices Governance Component

Namespace * istio-system

Fill in or select the Namespace for the Microservices Governance component deployment.

10.1.2 Node Auto-Scaling Group

10.1.2.1 General

The Node Auto-Scaling Group is a core resource scheduling component adapted for Kubernetes clusters. Its core function is to automatically create/delete cloud server nodes by connecting to the cloud platform, and dynamically adjust the number of cluster nodes based on business loads. This achieves precise matching between resource supply and business demands, serving as a key tool to ensure the efficient and stable operation of K8s clusters.

Function Principle

The core principle of the Node Auto-Scaling Group is to realize automatic node scaling through a "monitoring-judgment-execution" closed loop, without manual intervention throughout the process. Relying on the linkage with K8s native scheduling mechanisms and the cloud platform, it achieves dynamic closed-loop management of node resources. Details are as follows:

- Scaling Out: When the system detects insufficient cluster resources (with pending Pods to be scheduled), it automatically selects eligible Node Auto-Scaling Groups. Through cloud platform APIs, it creates cloud servers according to the preset node configurations of the group (such as computing specifications, images, networks, storage, etc.), and adds them to the cluster for Pod scheduling after initialization.
- Scaling In: When the system detects long-term idle cluster resources (e.g., CPU/memory utilization rate continuously below 50%), it automatically selects eligible Node Auto-Scaling Groups. It first migrates schedulable Pods on the target nodes using K8s native eviction mechanisms, then removes the nodes and synchronously deletes the corresponding cloud servers to release resources.
- Auto-Scaling Group Selection: During the scaling process, the system randomly selects target groups from all Node Auto-Scaling Groups that meet preset conditions to execute corresponding operations.

Key Features

- Multi-dimensional Trigger Conditions: Supports scaling based on Pod Pending status, CPU/memory utilization rate, etc.
- Compatibility with K8s Native Rules: Can be associated with node selectors and node affinity/anti-affinity configurations to ensure scaled-out nodes meet Pod scheduling requirements.
- Graceful Scaling Mechanism: Before scaling in, automatically evicts schedulable Pods on nodes to other healthy nodes to avoid business interruptions; built-in node health checks to automatically replace abnormal nodes.

Typical Application Scenarios

- Periodic Task Scenarios: Computational load peaks occur during fixed daily periods (e.g., morning peak data processing, nightly backups). The Node Auto-Scaling Group scales out according to load metrics and scales in quickly after tasks are completed to improve resource utilization.

- Sudden Traffic Scenarios: Sudden access surges caused by APP version updates or hot events trigger rapid scaling out through real-time load monitoring to avoid cluster overload; automatic scaling in when traffic subsides, no manual intervention required.
- E-commerce Promotion Scenarios: Traffic surges sharply during promotional activities, leading to a surge in Pod scheduling requests. The Node Auto-Scaling Group triggers scaling out to add nodes, ensuring the normal operation of core services such as orders and payments; automatic scaling in after the event to reduce idle resource costs.

Usage Restrictions

- Currently only supports clusters running Kubernetes 1.30.
- Only supports self-built clusters, not Integrated clusters.
- During elastic scaling out, the total vCPU of the nodes to be scaled out cannot exceed the available quota of the license.
- Nodes initialized when creating the cluster and manually added in the node list are not managed by the Node Auto-Scaling Group and cannot be automatically scaled in.

10.1.2.2 Create Node Auto-Scaling Group

In the main menu of nKU, click **Cluster Management > Cluster Management > Node Auto-Scaling Group** to enter the Node Auto-Scaling Group interface.Click **Create Node Auto-Scaling Group** to access the creation interface.

Creating a Node Auto-Scaling Group involves the following three steps:

1. Basic Configuration

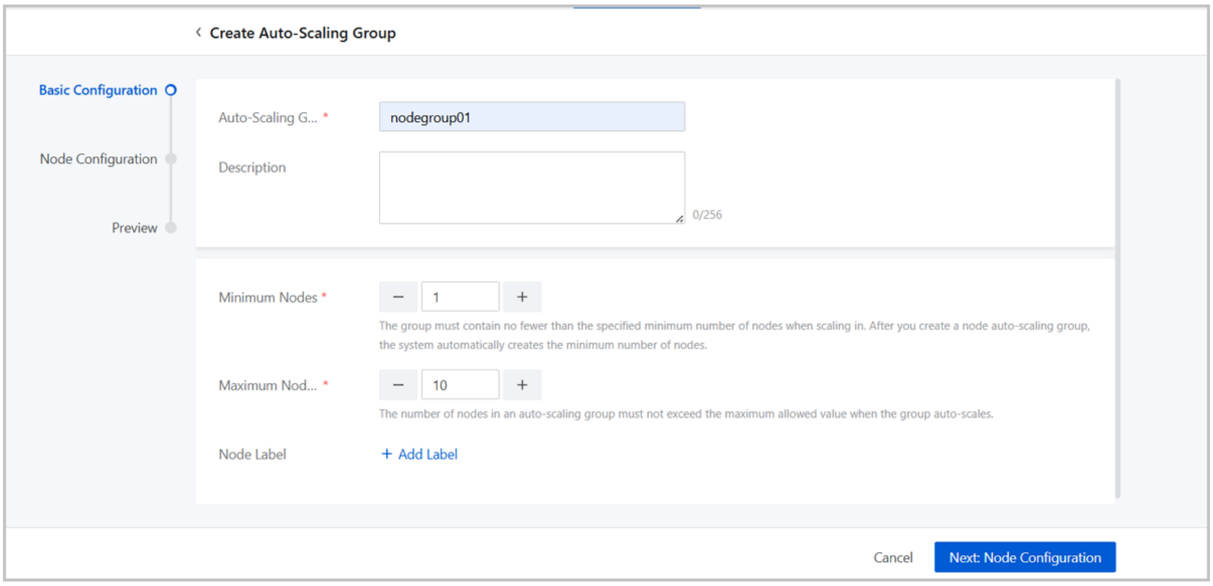
Enter the corresponding content with reference to the following examples:

- **Auto-Scaling Group Name:** Set the name of the Node Auto-Scaling Group. Naming Rules: 1-50 characters in length; can only contain lowercase letters, numbers, and the special character "-"; must start with a lowercase letter and end with a number or lowercase letter.
- **Description:** Optional. Add relevant remarks.
- **Minimum Nodes:** Set the minimum number of nodes. The system will ensure the number of nodes in the Node Auto-Scaling Group never falls below this value.
- **Maximum Nodes:** Set the maximum number of nodes. During elastic scaling out, the number of nodes in the group will not exceed this value.

- **Node Labels:** Add node labels. The set labels will be automatically assigned to nodes during scaling out. Meanwhile, the system will follow the node selector specified by the Pod and the requiredDuringSchedulingIgnoredDuringExecution constraint in node affinity, only considering Node Auto-Scaling Groups with node labels that meet these constraints.

As shown in Figure: Create Node Auto-Scaling Group-Basic Configuration:

Figure: Create Node Auto-Scaling Group-Basic Configuration



Auto-Scaling G... * nodegroup01

Description 0/256

Minimum Nodes * − 1 +

The group must contain no fewer than the specified minimum number of nodes when scaling in. After you create a node auto-scaling group, the system automatically creates the minimum number of nodes.

Maximum Nod... * − 10 +

The number of nodes in an auto-scaling group must not exceed the maximum allowed value when the group auto-scales.

Node Label + Add Label

2. Node Configuration

Configure node settings. During elastic scaling, cloud servers will be created on the cloud platform based on these node configurations. Refer to the following examples to enter the corresponding information:

- **Node Name Prefix:** Set the prefix for node names. Node names consist of this prefix plus a random number automatically generated by the system. Cloud server names are composed of the K8s cluster name and the node name.
- **VM Zone :** Select the Zone of the cloud platform. VMs will be created in this region.
- **CPU Architecture:** Set the CPU architecture of the VM.
- **Operating System:** Select the operating system for the VM.
- **CPU:** Set the CPU specifications of the VM.
- **Memory:** Set the memory size of the VM.
- **Root Volume Size:** Set the capacity of the Root Volume Size for the VM.

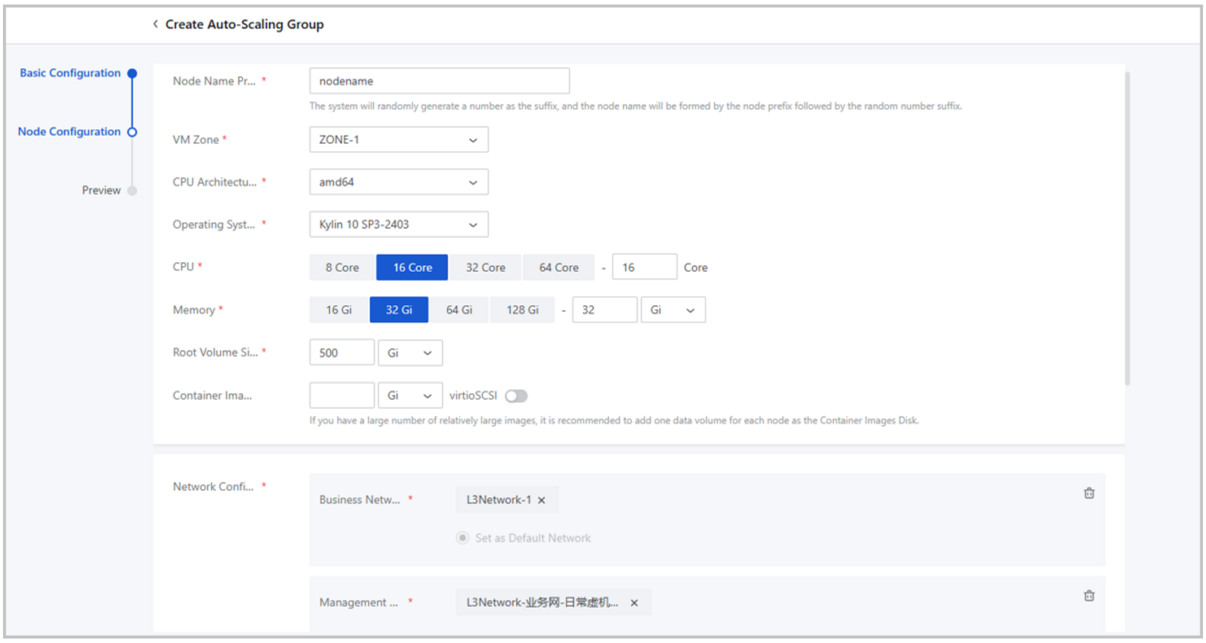
- **Container Images Disk:** Optional.If you have a large number of relatively large images, it is recommended to add one data volume for each node as the Container Images Disk.
- **Network Configuration:** Select the Layer 3 networks for the business network and management network. The business network will serve as the default network for the VM.

  Note:

  - Ensure the Layer 3 network used for the business network can communicate with other nodes in the cluster.
  - If the cluster's management network and business network are reuse, ensure the service network is accessible by the management cluster.
  - The management network only needs to be configured when the cluster's management network and service network are not reuse. In this case, the management network must use the same network segment as the management cluster's network.
- **ROOT Password:** Set the password for the ROOT account of the VM.
- **SSH Port:** Set the port used for SSH access to the VM. The default is 22.
- **DNS Server:** Add a DNS server to resolve domain names outside the K8s cluster. If no dedicated DNS server is deployed, you can enter a public DNS server address (e.g., 223.5.5.5).
- **High Mode**： Optional,Set the high availability mode of the VM. Supports two options: None and NeverStop. The default is None.
- **VM Cluster :** Optiona. Select the cluster of the cloud platform. VM will be created in this cluster. If not specified, the cloud platform will assign it automatically.
- **Root Volume Primary Storage**: Optiona. Set the primary storage for the root volume of the VM. If not specified, the cloud platform will assign it automatically.
- **Primary Storage for Data Volume:** Optiona. Set the primary storage for the data volume of the VM. If not specified, the cloud platform will assign it automatically.

As shown in Figure: Create Node Auto-Scaling Group-Node Configuration:

Figure: Create Node Auto-Scaling Group-Node Configuration



‹ Create Auto-Scaling Group

Node Name Pr... * nodename

The system will randomly generate a number as the suffix, and the node name will be formed by the node prefix followed by the random number suffix.

VM Zone * ZONE-1

CPU Architectu... * amd64

Operating Syst... * Kylin 10 SP3-2403

CPU * 8 Core | 16 Core | 32 Core | 64 Core - 16 Core

Memory * 16 Gi | 32 Gi | 64 Gi | 128 Gi - 32 Gi

Root Volume Si... * 500 Gi

Container Ima... Gi virtioSCSI

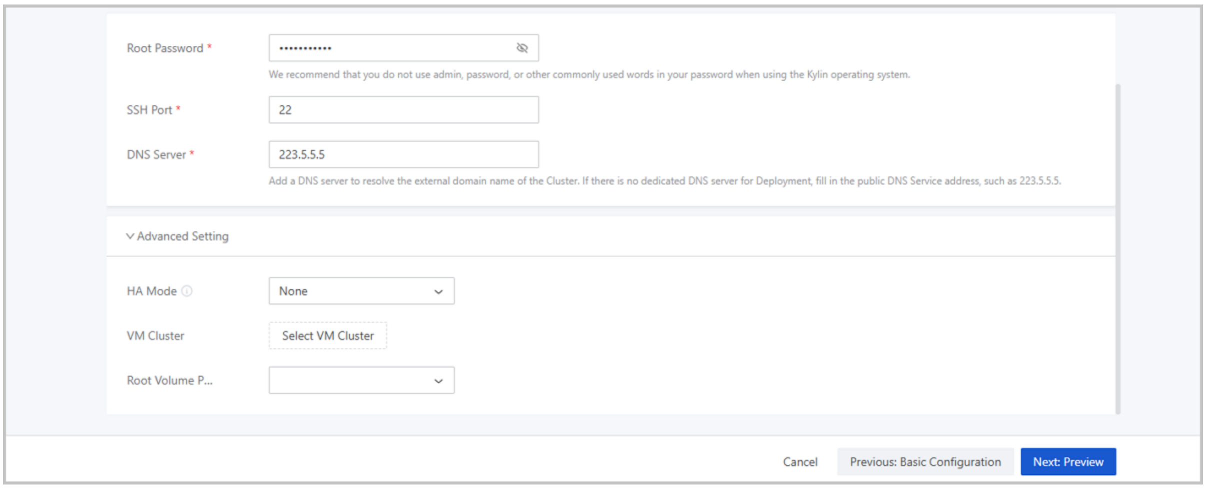
If you have a large number of relatively large images, it is recommended to add one data volume for each node as the Container Images Disk.

Network Confi... *

Business Netw... * L3Network-1 ×

○ Set as Default Network

Management ... * L3Network-业务网-日常虚机... ×



Root Password * ••••••••••••

We recommend that you do not use admin, password, or other commonly used words in your password when using the Kylin operating system.

SSH Port * 22

DNS Server * 223.5.5.5

Add a DNS server to resolve the external domain name of the Cluster. If there is no dedicated DNS server for Deployment, fill in the public DNS Service address, such as 223.5.5.5.

Advanced Setting

HA Mode None

VM Cluster Select VM Cluster

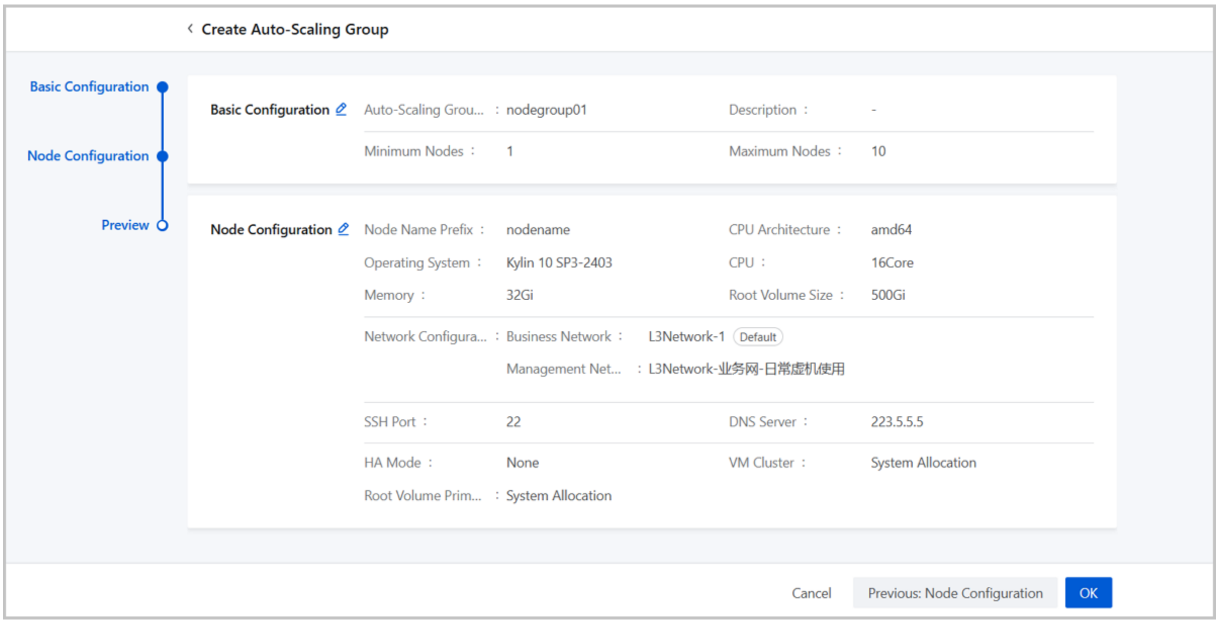
Root Volume P...

3. Preview

Confirm the configuration information of the cluster ,support jump to modify.

As shown in Figure: Create Node Auto-Scaling Group-Preview:

Figure: Create Node Auto-Scaling Group-Preview



‹ Create Auto-Scaling Group

Basic Configuration

Auto-Scaling Grou... :	nodegroup01	Description :	-
Minimum Nodes :	1	Maximum Nodes :	10

Node Configuration

Node Name Prefix :	nodename	CPU Architecture :	amd64
Operating System :	Kylin 10 SP3-2403	CPU :	16Core
Memory :	32Gi	Root Volume Size :	500Gi
Network Configura... :	Business Network : L3Network-1 Default Management Net... : L3Network-业务网-日常虚机使用		
SSH Port :	22	DNS Server :	223.5.5.5
HA Mode :	None	VM Cluster :	System Allocation
Root Volume Prim... :	System Allocation		

10.1.2.3 Mangement Node Auto-Scaling Group

In the nKU main menu, click **Cluster Management** > **Cluster Management** > **Node Auto-Scaling Group** to access the **Node Auto-Scaling Group** interface.

The Node Auto-Scaling Group supports the following operations:

Operation	Description
Create Node Auto-Scaling Group	Add a node auto-scaling group to the cluster to achieve the effect of automatically scaling nodes.
Modify Basic Configuration	Modify the basic configuration of the node auto-scaling group. **Note:**The basic configuration cannot be modified when the status of the node auto-scaling group is abnormal.
Modify Node Configuration	Modify the node configuration of the node auto-scaling group. **Note:** • After modifying the node configuration of the node auto-scaling group, it only takes effect for newly

	scaled nodes. Nodes already existing under the node auto-scaling group will not be changed. • The Zone, CPU architecture, operating system, CPU, and memory in the node configuration are not allowed to be modified.
Fault Recovery	When elastic scaling fails repeatedly due to cloud platform failures, license over-quota, and other reasons, it may cause the node auto-scaling group to fail. After troubleshooting and resolving the fault cause, you can use fault recovery to make the node auto-scaling group automatically return to normal.
Node Auto-Scaling Details	1. Enter the node auto-scaling details page and the overview sub-page to view the basic information and node configuration of the node auto-scaling group. 2. Enter the node sub-page to view the node list under the current node auto-scaling group. 3. Enter the scaling log sub-page to view the scaling logs of the node auto-scaling group, including task results, the number of successfully/expectedly scaled nodes, the number of nodes after scaling, etc. You can view and export detailed scaling logs.
Delete Node Auto-Scaling Group	Delete the node auto-scaling group. **Note:** Deleting the node auto-scaling group will simultaneously remove the nodes under the scaling group and delete the corresponding VMs of the nodes. This may cause the pods originally running on the nodes to be abnormal due to insufficient resources. Please operate with caution.

10.1.3 Node

10.1.3.1 General

Nodes are the basic building blocks of a Kubernetes cluster and are categorized as Master and Node. Users can add or remove compute nodes to the cluster according to their actual needs.

10.1.3.2 Add node

Adding nodes mainly refers to adding compute nodes to the Kubernetes cluster for scaling.

Preliminary

Before adding a node, ensure the node meets the following conditions:

- Before adding a host as a worker node to the cluster, ensure the host has installed the Cloud customized operating system. Contact official technical support to obtain the customized operating system matching the current platform version.
- Ensure the network cards corresponding to the external network of all pod external netwrok exist on the host, so as to guarantee the normal operation of relevant network services.

Add Node

In the nKU main menu, click **Cluster Management** > **Cluster Management** > **Node** to enter the **Node** interface. Click **Add Node** to enter the **Add Node** interface.

Adding a node is divided into the following three steps:

1. Node Configuration

can be entered accordingly by referring to the following example:

- **Cluster name**: displays the current cluster name by default
- **Management network and business network Reuse**: Default display whether the current cluster management network and business network multiplexing is enabled, if it is enabled, the management network IPv4 address of the node is the same as the IPv4 address of the business network.
- **Node Information**: Fill in the information of the node to be added.

  # **Add Type**: The type for adding nodes include IP and IP range

# **Business network IPv4**: fill in the business network IPv4 address of the node. Fill in this field when the Add Type is IP.

# **Management network IPv4**: Fill in the IPv4 address of the management network of the node. Fill in this field when the Add Type is IP.The management cluster accesses and manages the business cluster nodes through the management network IPv4 address, so make sure that the management cluster is connected to this address. When the management network is multiplexed with the business network, the management network IPv4 address and the business network IPv4 address are the same.

# **Business network IPv4 Range**: fill in the business network IPv4 range of the node. Fill in this field when the Add Type is IP Range.

# **Management network IPv4**: Fill in the IPv4 range of the management network of the node. Fill in this field when the Add Type is IP Range. When the management network is multiplexed with the business network, the management network IPv4 range and the business network IPv4 range are the same.

Notes.

- Please make sure that the IP address you fill in is the same as the actual configuration of the machine.
- Make sure that the management network IPv4 address and the management cluster can be connected. If the management network is multiplexed with the business network, make sure that the business network IPv4 address is connectable to the management cluster.
- Make sure that the business network IPv4 addresses of the nodes are connectable to each other.
- If the Management Network is not reused for Business Network, make sure that the IPs in the Management Network IPv4 Range correspond one-to-one to the IPs in the Business Network IPv4 range. IPs in the same order belong to the same node. For example, the first IP in the Business Network IPv4 Range and the first IP in the Management Network IPv4 Range belong to the first node.
- When adding nodes in batch, the IPv4 address of the management network of the nodes in this batch cannot be duplicated; the IPv4 address of the business network of the nodes in this batch cannot be duplicated; if

the management network and the business network are not multiplexed, the IPv4 address of the management network and the IPv4 address of the business network cannot be duplicated.

- The management network IPv4 address and business network IPv4 address of the nodes in this batch cannot be duplicated with existing nodes in the cluster.

# **Node Name**: Set the name of the node. Naming rules: length limit 2-50 characters, can only contain lowercase letters, numbers and separator (-), must start with a lowercase letter and cannot end with a separator.When the Add Type is IP range, you need to specify a starting suffix. The node name consists of a base name and a suffix, and the suffix will automatically increment sequentially starting from the starting suffix, such as "-1/-2/-3" and so on.

Notes.

- The add node process sets the hostname of the host to the node name.
- Node names cannot be duplicated with nodes already in the cluster.
- When adding nodes in a batch, the names used for nodes in this batch cannot be duplicated.

# **Node role:** worker by default, optional GPU node

Notes.

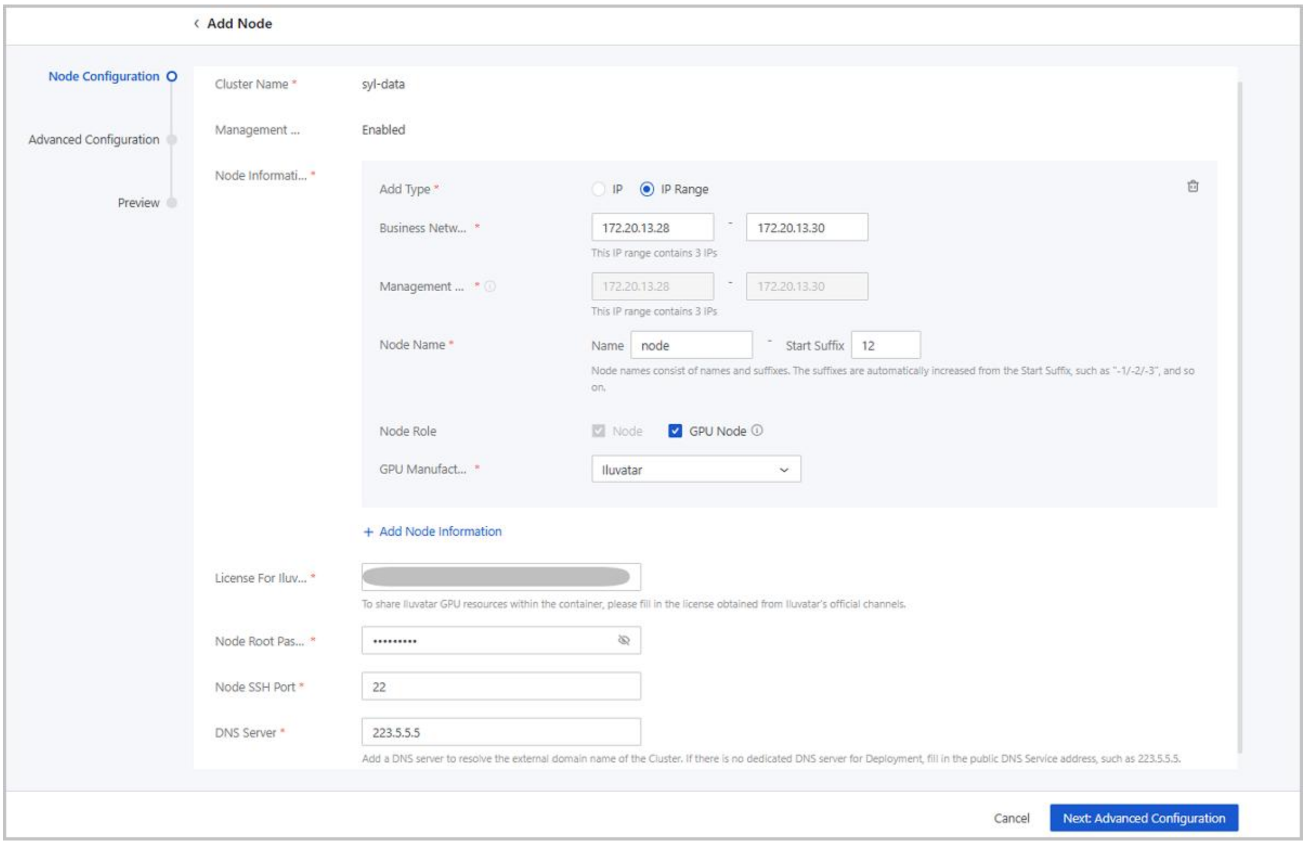
- The GPU node will automatically install the corresponding components during deployment, so please choose whether to install the machine as a GPU node according to the actual situation.
- If the GPU node role is checked, please make sure that the GPU driver has been installed correctly on the node. If the driver is not installed, only the corresponding components will be installed and the node will not be set to the GPU node role.

# **GPU Manufacturer**: Fill in this field when checking the GPU node for the node role, select the manufacturer of the GPU manufacturer on the node, and the system will install the GPU-related components of the corresponding manufacturer, currently supported: NVIDIA, Huawei, HYGON,Iluvatar.

- **License For Iluvatar GPU Virtualization:** This field must be filled in when there are no Iluvatar GPUs in the cluster yet, and the nodes to be added this time include GPU nodes with Iluvatar as the GPU manufacturer. Enter the license for Iluvatar GPU virtualization. This license is required if you need to share Iluvatar GPUs in containers, and it can be obtained from Iluvatar official sources.
- **Node ROOT password**: Fill in the node root password to facilitate the process of adding nodes, SSH to the node to perform the installing steps, please ensure that the ROOT password of each node is the same.
- **Node ssh port**: Fill in the port used by the SSH node to facilitate the process of adding nodes, SSH to the node to perform the installing steps, please make sure that the port of each node can be connected.
- **DNS server**: Add a DNS server to provide DNS resolution service to resolve domain names external to the cluster. If you have not deployed a dedicated DNS server, you can fill in the public DNS server address, such as 223.5.5.5.

As shown in Figure 71: Adding a Node - Node Configuration:

Figure 71: Add Node - Node Configuration



Cluster Name * syl-data

Management ... Enabled

Node Informati... *

Add Type * ○ IP ● IP Range

Business Netw... * 172.20.13.28 - 172.20.13.30

This IP range contains 3 IPs

Management ... * 172.20.13.28 - 172.20.13.30

This IP range contains 3 IPs

Node Name * Name node - Start Suffix 12

Node names consist of names and suffixes. The suffixes are automatically increased from the Start Suffix, such as "-1/-2/-3", and so on.

Node Role ☑ Node ☑ GPU Node

GPU Manufact... * Iluvatar

License For Iluv... * [redacted]

To share Iluvatar GPU resources within the container, please fill in the license obtained from Iluvatar's official channels.

Node Root Pas... * ••••••••••

Node SSH Port * 22

DNS Server * 223.5.5.5

Add a DNS server to resolve the external domain name of the Cluster. If there is no dedicated DNS server for Deployment, fill in the public DNS Service address, such as 223.5.5.5.

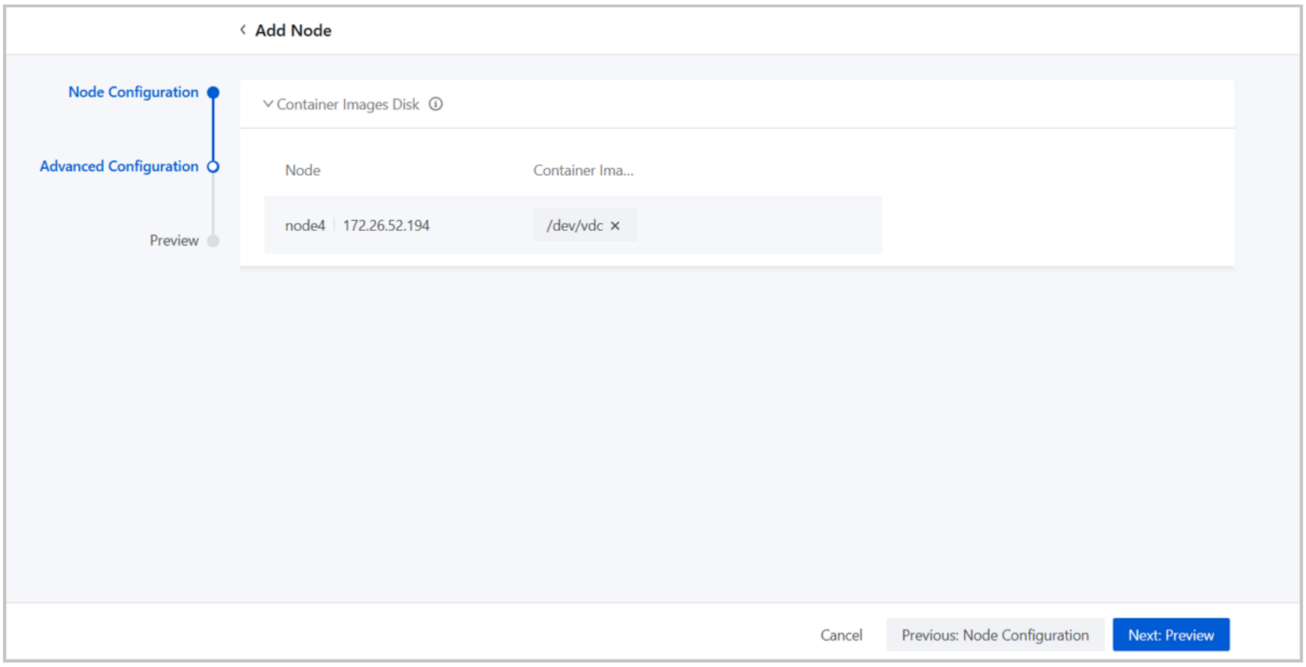
2. Advanced Configuration

refer to the following example to enter the appropriate content:

- Container Image Disk: optional, select the Image data disk used to store the Imageed data. If the container images disk is not set, the Image data is stored in the system disk by default. When more and larger Images are used, it is recommended to load a dedicated volume for the node to store the Images, so as to avoid the Image data from filling up the system disk, resulting in system abnormalities.

As shown in Figure 72: Adding Nodes-Advanced Configuration:

Figure 72: Adding Nodes-Advanced Configuration



‹ Add Node

Container Images Disk

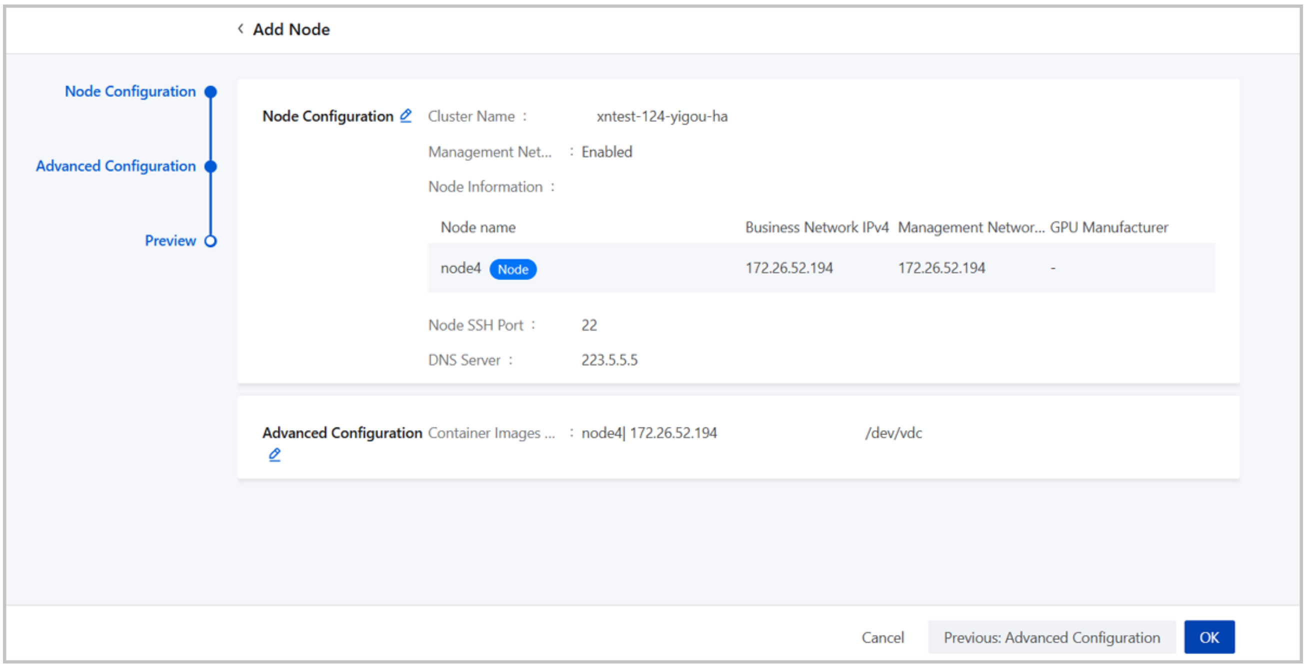
Node	Container Ima...
node4 | 172.26.52.194	/dev/vdc ×

3. Confirm configuration

Confirm the configuration information of the node , support jump to modification.

As shown in Figure 73: Adding a Node-Confirming Configuration:

Figure 73: Adding Nodes - Confirming Configuration



Node Configuration

Cluster Name : xntest-124-yigou-ha

Management Net... : Enabled

Node Information :

Node name	Business Network IPv4	Management Networ...	GPU Manufacturer
node4 Node	172.26.52.194	172.26.52.194	-

Node SSH Port : 22

DNS Server : 223.5.5.5

Advanced Configuration

Container Images ... : node4\| 172.26.52.194 /dev/vdc

After clicking **OK**, the system will start adding nodes and the cluster status changes to **Scaling**. Users click **Cluster Management** > **Cluster Management** > **Cluster** to enter the cluster interface and view the addition of nodes.

Notes.

- Currently, only worker nodes and GPU nodes are supported to be added, not control planes.
- Adding nodes is not allowed when the cluster is in an abnormal or deleting state.

10.1.3.3 Mangement Node

In the nKU main menu, click **Cluster Management** > **Cluster Management** > **node** to access **the node** interface.

The node supports the following operations:

manipulate	descriptive
Add Node	Add compute nodes to the cluster for capacity expansion. **Notes.** • Currently, only worker nodes and GPU nodes are supported to be added, not control planes. • Adding nodes is not allowed when the cluster is in an abnormal or deleting state. • Before the nodes join the cluster, you need to install the

	Cloud customized operating system . • Ensure the network cards corresponding to the external network of all pod external netwrok exist on the host, so as to guarantee the normal operation of relevant network services
Remove nodes	Remove the node from the cluster. **Notes.** • Currently, only compute nodes and GPU nodes are supported for removal, not control plane nodes. • Removing nodes is not allowed when the cluster is in an abnormal or deleting state. • The nodes will be cleaned up when you remove them, but they may not be cleaned up completely. If you need to add nodes to the cluster again, it is recommended that you reinstall the operating system first. • Nodes under the Node Auto-Scaling Group do not support manual removal.
disable schedule	After you stop scheduling a node, newly deployed applications will not be able to be scheduled to that node. This feature is mainly used in scenarios where nodes need to be maintained. **Notes.** - Disable schedule nodes only when the cluster state is **running**. - Only nodes in the **running** state can disable scheduling.
enable schedule	After the scheduling node is restored, newly deployed applications can be scheduled to that node. **Notes.** - A schedule node can only be enabled schedule when the cluster state is **running**. - Only nodes **in the running** state can enable scheduling.
Node details	**1.** Go to the node details page to view basic node information, including node status, scheduling status, CPU utilization, memory utilization, metadata, and so on. **Notes.**

	• CPU utilization is the actual CPU utilization of the current node. • The CPU request rate is the total number of all pod request values for the current node/total node CPU. • The CPU limit rate is the total number of limit values for all Pod on the current node/total node CPU. • Memory Usage is the actual current node memory usage. • The memory request rate is the total number of all pod request values for the current node/total node memory. • The memory limit rate is the total number of limit values for all Pod on the current node/total node memory. • Total node CPU and memory refers to the total amount of Kunbernetes that can be allocated, which may be less than the actual CPU and memory of the node. **2.** For GPU nodes, you can go to the GPU page to view information about all GPUs on the current node and monitoring data for GPUs
Export node information	Export node information as a CSV file. Supports exporting the current page or all pages.

10.1.4 GPU pool

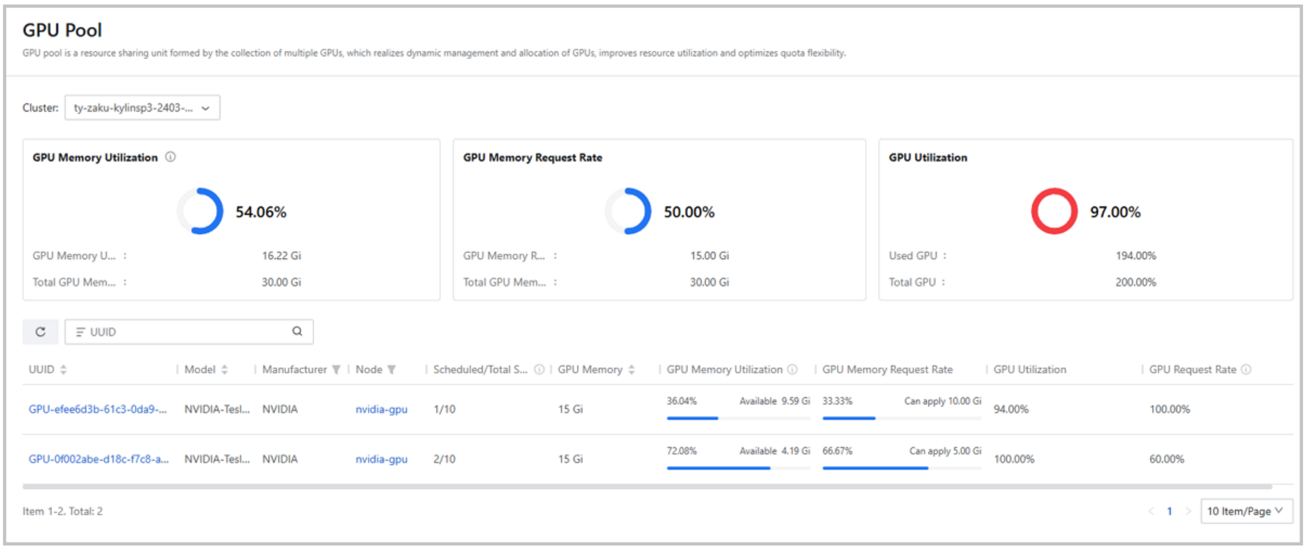
GPU pool is a resource sharing unit formed by collecting multiple GPUs to achieve dynamic management and allocation of GPUs, improve resource utilization and optimize the flexibility of resource allocation.

Viewing the GPU pool

In the nKU main menu, click **Cluster Management** > **Cluster Management** > **GPU pools** to access the **GPU pools** interface.

As shown in Viewing GPU pool:

Figure 74: Viewing GPU pool



GPU Pool

GPU pool is a resource sharing unit formed by the collection of multiple GPUs, which realizes dynamic management and allocation of GPUs, improves resource utilization and optimizes quota flexibility.

Cluster: ty-zaku-kylinsp3-2403-...

GPU Memory Utilization: 54.06%
GPU Memory U... : 16.22 Gi
Total GPU Mem... : 30.00 Gi

GPU Memory Request Rate: 50.00%
GPU Memory R... : 15.00 Gi
Total GPU Mem... : 30.00 Gi

GPU Utilization: 97.00%
Used GPU : 194.00%
Total GPU : 200.00%

UUID

UUID	Model	Manufacturer	Node	Scheduled/Total S...	GPU Memory	GPU Memory Utilization	GPU Memory Request Rate	GPU Utilization	GPU Request Rate
GPU-efee6d3b-61c3-0da9-...	NVIDIA-Tesl...	NVIDIA	nvidia-gpu	1/10	15 Gi	36.04% Available 9.59 Gi	33.33% Can apply 10.00 Gi	94.00%	100.00%
GPU-0f002abe-d18c-f7c8-a...	NVIDIA-Tesl...	NVIDIA	nvidia-gpu	2/10	15 Gi	72.08% Available 4.19 Gi	66.67% Can apply 5.00 Gi	100.00%	60.00%

Item 1-2. Total: 2

The GPU pool interface consists of two parts: a monitoring chart and a GPU list.

- Monitoring Chart: Statistics on the total GPU memory usage, GPU memory request and GPU usage of all GPUs under the selected cluster, which is convenient to understand the current total GPU usage of the cluster.
- GPU List: Displays information about all GPUs in the selected cluster, including: UUID, model, node it belongs to, number of scheduled/total schedulable containers, GPU memory, GPU memory utilization, GPU memory request rate,GPU utilization, GPU request rate.
- GPU Details: You can click on the GPU UUID to enter the GPU details page and view the monitoring graphs of GPU utilization, GPU memory utilization, temperature, power consumption, and information about the currently scheduled pods on the GPU.

10.2 Storage management

10.2.1 Storageclass

10.2.1.1 General

The StorageClass is a configuration template used in Kubernetes to dynamically create a PVC that defines the properties of the volume, the creation policy, and the required storage plugins.

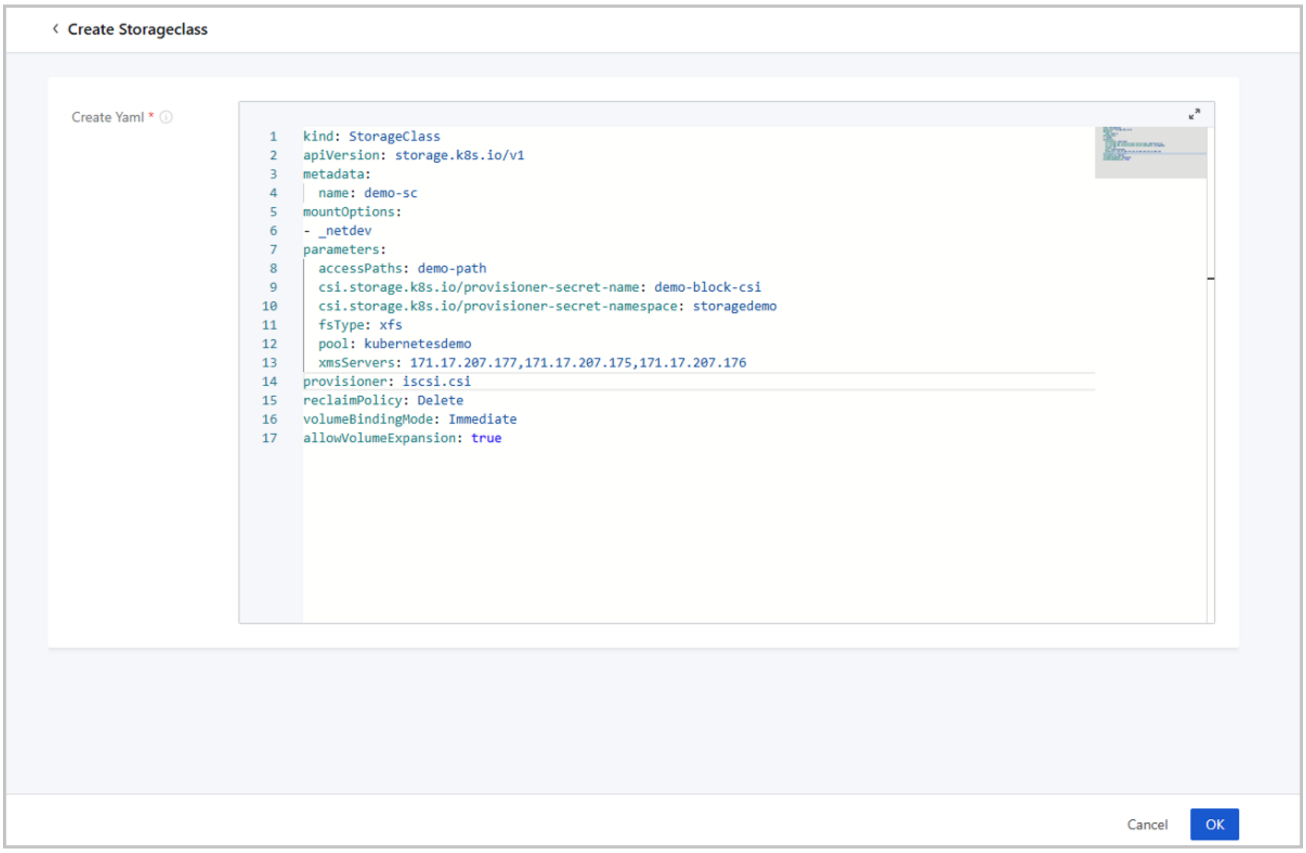
10.2.1.2 Creating storage classes

In the nKU main menu, click **Cluster Management** > **Storage Management** > **Storage Classes**. In the **Storage Classes** screen, click **Create Storage Class** to bring up the **Create Storage Class** screen.

Enter Yaml for the storage class and click **OK** to create it.

As shown in Figure 75: Creating a storage class:

Figure 75: Creating a Storage Class



```yaml
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: demo-sc
mountOptions:
- _netdev
parameters:
  accessPaths: demo-path
  csi.storage.k8s.io/provisioner-secret-name: demo-block-csi
  csi.storage.k8s.io/provisioner-secret-namespace: storagedemo
  fsType: xfs
  pool: kubernetesdemo
  xmsServers: 171.17.207.177,171.17.207.175,171.17.207.176
provisioner: iscsi.csi
reclaimPolicy: Delete
volumeBindingMode: Immediate
allowVolumeExpansion: true
```

10.2.1.3 Managing storage classes

In the nKU main menu, click **Cluster Management > Storage Management > Storage Classes** to access the **Storage Classes** screen.

The storage class supports the following operations:

manipulate	descriptive
Creating storage classes	Create a new storage class.

Update Yaml	Updates to storage classes by way of Yaml
Deleting a storage class	Remove the storage class. **Notes.** • Before deleting a storage class, ensure that no pvc or persistent volume created based on that storage class exists in the cluster. • After deletion, the storage class cannot continue to be used to create pvcs. If there is no other storage class in the cluster, users will not be able to create pvcs, so please be careful.

10.3 Network management

10.3.1 External network

10.3.1.1 General

The external network provides IP addresses for Pods and loadbalancer services within the cluster, enabling access to Pods or loadbalancer services from outside the Kubernetes cluster.

The external network supports unified planning and management of networks used by LoadBalancer services and Pods. It divides the IP ranges for LoadBalancer services and Pods through different network segments. The external network supports the following two types of network segments:

1. Service External Network

- The service external network is used to assign IPs for LoadBalancer-type services, which support access from outside the Kubernetes cluster.
- To use the service external network, all Control Plane Nodes must have the same network card name, and each network card must be configured with an IP.

2. Pod External Network

- The Pod external network is used to assign IPs for the external network of Pods, enabling direct access to Pod IPs from outside the Kubernetes cluster.

- By default, Pods use the native Kubernetes container network. You can configure the network for Pods in Workload > Advanced Configuration > Network Settings.
- To use the Pod external network, all nodes must have the same network card name.

10.3.1.2 Creating external network

In the nKU main menu, click **Cluster Management** > **Network Management** > **External Network**. In the **External Network** screen, click **Create External Network** to bring up the **Create External Network** screen.

You can refer to the following example to enter the appropriate contents:

- **Name**: Set the external network name. Naming rules: length limit of 1-50 characters, can only contain lowercase letters, numbers and separator (-), and must begin with a lowercase letter and end with a number or lowercase letter
- **Description**: optional, note external network related information
- **NIC name**: Select the NIC to be used for the external network. Only network cards with the same name on all Control Plane Nodes and not used by other external networks are supported

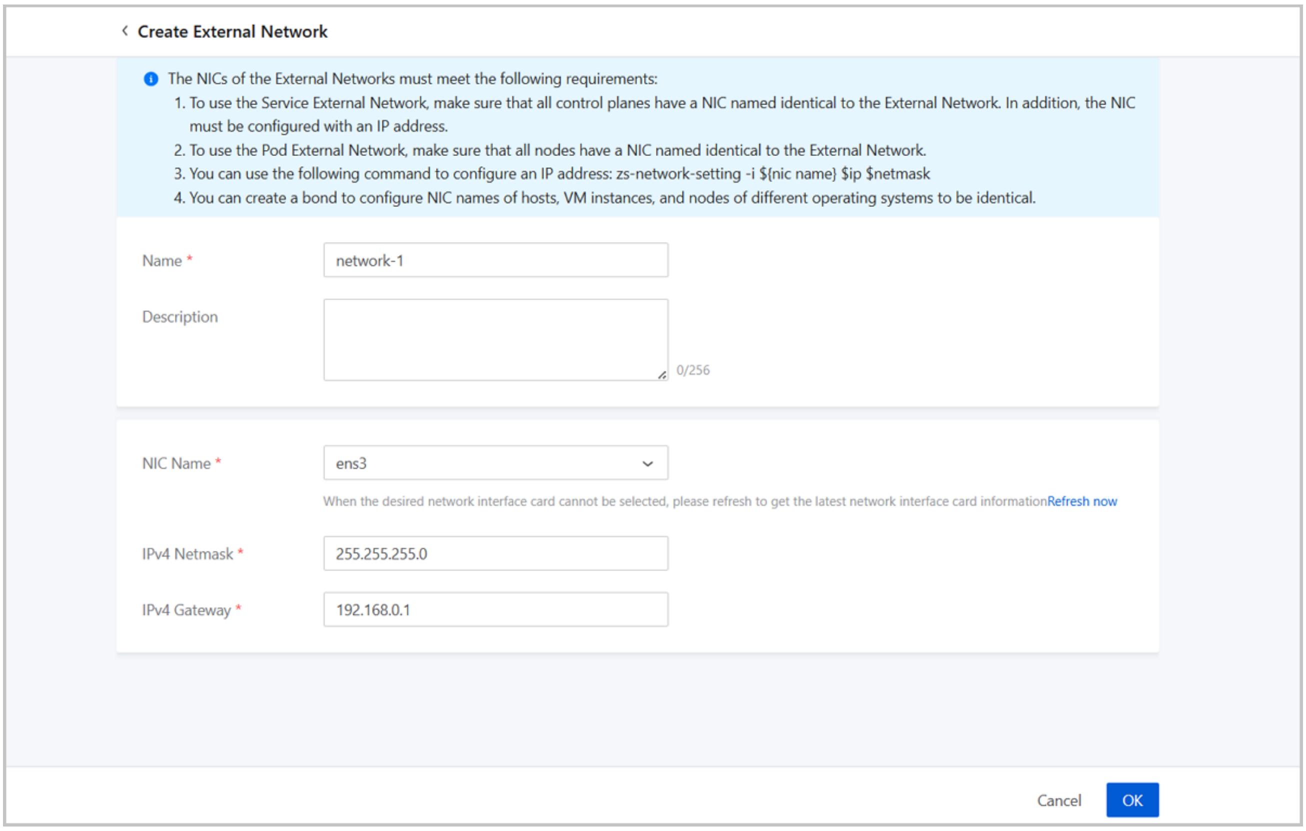
  Note: External network cards must meet the following requirements:

  1. To use the service external network, ensure all management nodes have network cards with the same name, and each network card is configured with an IP address.
  2. To use the Pod external network, ensure all nodes have network cards with the same name.
  3. You can use the following command to configure the IP: zs-network-setting -i ${network card name} $ip $netmask
  4. You can create a bond to keep the network card names consistent across physical machines, virtual machine nodes, or nodes with different operating systems.
- **Netmask**: Set the subnet mask of the external network segment, e.g. 255.255.0.0.
- **Gateway**: Set the gateway of the external network segment, for example: 172.20.0.1

  Note: The CIDR of the external network is determined by the IPv4 subnet mask and IPv4 gateway, and the CIDRs of different external networks must not overlap.

As shown in Figure 76: Creating an External Network:

Figure 76: Creating an External Network



‹ Create External Network

The NICs of the External Networks must meet the following requirements:

1. To use the Service External Network, make sure that all control planes have a NIC named identical to the External Network. In addition, the NIC must be configured with an IP address.
2. To use the Pod External Network, make sure that all nodes have a NIC named identical to the External Network.
3. You can use the following command to configure an IP address: zs-network-setting -i ${nic name} $ip $netmask
4. You can create a bond to configure NIC names of hosts, VM instances, and nodes of different operating systems to be identical.

Name * network-1

Description 0/256

NIC Name * ens3

When the desired network interface card cannot be selected, please refresh to get the latest network interface card informationRefresh now

IPv4 Netmask * 255.255.255.0

IPv4 Gateway * 192.168.0.1

Note: After the external network is successfully created, you need to navigate to the external network details page and add a network segment before the network can be used.

10.3.1.3 Managing external network

In the nKU main menu, click **Cluster Management** > **Network Management** > **External Network** to access the **External Network** interface.

The external network supports the following operations:

manipulate	descriptive
Create external network	Create a new external network.
Add Network Range	Add a new network segment to the external network. Two types of network segments are supported: service external network segments and Pod external network segments.

	Notes. 1. Creating a service external network is not supported when there are abnormal management nodes in the cluster. 2. Creating a Pod external network is not supported when the cluster is scaling or has abnormal nodes. 3. Adding a service external network segment under an external network requires that all management nodes have the external network's network card, and each has been configured with an IP. You can use the following command to configure the IP: zs-network-setting -i ${network card name} $ip $netmask. 4. Adding a Pod external network segment under an external network requires that all nodes have the external network's network card. You can create a bond to keep network card names consistent across physical machines, virtual machine nodes, or nodes with different operating systems. 5. The new network segment must be within the IPv4 CIDR range of the current external network. This CIDR is determined by the gateway and subnet mask filled in when creating the external network, and you can view the external network's IPv4 CIDR on the external network main list or details page. 6. The network segment must not include the gateway, broadcast address, network address, or occupied IP addresses. 7. The network segment must not conflict with the IP ranges of existing network segments added under the current external network.
External network details	Go to the external network details page to view basic external network information, IP usage, and manage network segments under the external network.
Delete external networks	Delete the external network.

10.3.1.4 External network details

10.3.1.4.1 External network Range

In the **external network** interface, click the external network name to enter the external network details page, select **a Network Range** to enter the **Network Range** label, where users can view the configuration of external network network segments and IP utilization, and centrally manage network segments, including adding and deleting them:

manipulate	descriptive
Add Network range	Add a new network segment to the external network. Two types of network segments are supported: service external network segments and Pod external network segments. **Notes.** 1. Creating a service external network is not supported when there are abnormal management nodes in the cluster. 2. Creating a Pod external network is not supported when the cluster is scaling or has abnormal nodes. 3. Adding a service external network segment under an external network requires that all management nodes have the external network's network card, and each has been configured with an IP. You can use the following command to configure the IP: zs-network-setting -i ${network card name} $ip $netmask. 4. Adding a Pod external network segment under an external network requires that all nodes have the external network's network card. You can create a bond to keep network card names consistent across physical machines, virtual machine nodes, or nodes with different operating systems. 5. The new network segment must be within the IPv4 CIDR range of the current external network. This CIDR is determined by the gateway and subnet mask filled in when creating the external network, and you can view the external network's IPv4 CIDR on the external network main list or details page.

	6. The network segment must not include the gateway, broadcast address, network address, or occupied IP addresses. The network segment must not conflict with the IP ranges of existing network segments added under the current external network.
Management IP Range	Manage the IP ranges of network segments, including deleting existing IP ranges and adding new IP ranges. **Notes.** • The IP ranges between all network segments under an external network must not conflict. • If any IP within an IP range is in use, the IP range cannot be deleted. • A service external network requires at least one IP range.
Set Sharing Mode	Set the sharing mode of the network segment. Currently, the following two modes are supported: • Global Sharing: Share the network segment globally, allowing all projects to use it. • Designated Sharing: Share the network segment with specific projects, allowing only the designated projects to use it. **Notes.** • Service External Network: If you cancel the designated project sharing mode for a service external network, the system will reassign IP addresses to the services under the project that used the service external network segment. If there are insufficient IPs, some services will have no available IP addresses. Please operate with caution.

	• Pod External Network: After modifying the sharing mode, the IPs already in use in the original project will be retained until the Pods using the IPs are deleted. If the Pods are created by a StatefulSet, you need to delete the Pods by reducing the number of replicas or deleting the StatefulSet to release the IPs.
Delete network range	Delete the network segment. **Notes.** • Deleting a network segment is not allowed when any IP within the segment is in use.

10.4 Project Management

10.4.1 Project

10.4.1.1 General

Project is a kind of tenant, which is used to realize resource isolation and user rights management, after the user joins the project, he/she can have the rights to view or operate the resources under the project.

- Once a user joins a project, he or she has permission to view and manage resources under that project.
- All resources created by the user under the current project belong to this project.
- The same resources can be viewed by different users under the same project, but resources between different projects are isolated from each other.
- Supports adding users belonging to the same team or with the same function to the same project, thus ensuring that the permissions between different teams are clearly defined, resources will not interfere with each other, and project data security and confidentiality are guaranteed.

10.4.1.2 Creating project

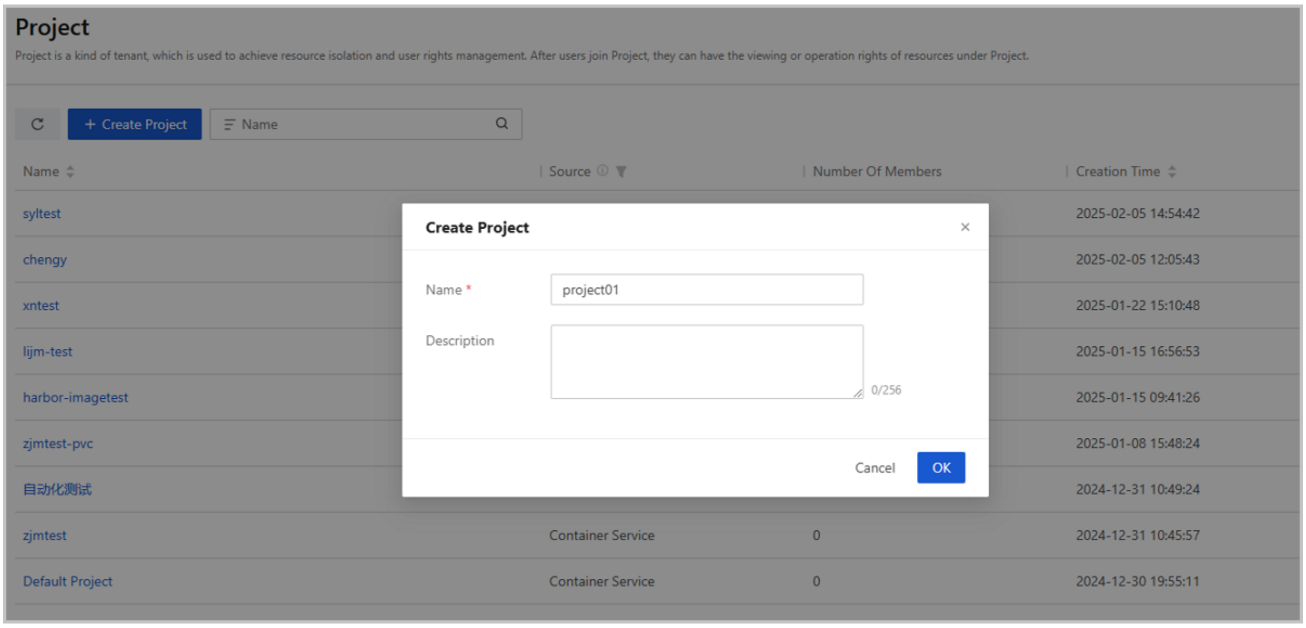
In the nKU main menu, click **Cluster Management** > **Project Management** > **Project** to enter the **Project** screen. Click **Create Project** to bring up the **Create Project** page.

You can refer to the following example to enter the appropriate contents:

- **Name:** Set the name of the project. Naming rules: length limit 1~128 characters, can only contain Chinese characters, English letters, numbers and the following 7 English characters (-) (_) (.) (() ()) (:) (+)
- **Description:** optional, note the project phase information

As shown in Figure 77: Create Project:

Figure 77: Create Project



Project

Project is a kind of tenant, which is used to achieve resource isolation and user rights management. After users join Project, they can have the viewing or operation rights of resources under Project.

Name	Source	Number Of Members	Creation Time
syltest			2025-02-05 14:54:42
chengy			2025-02-05 12:05:43
xntest			2025-01-22 15:10:48
lijm-test			2025-01-15 16:56:53
harbor-imagetest			2025-01-15 09:41:26
zjmtest-pvc			2025-01-08 15:48:24
自动化测试			2024-12-31 10:49:24
zjmtest	Container Service	0	2024-12-31 10:45:57
Default Project	Container Service	0	2024-12-30 19:55:11

Create Project

Name * project01

Description

10.4.1.3 Managing project

In the nKU main menu, click **Cluster Management** > **Project Management** > **Project** to access **the Project** screen.

The project supports the following operations:

manipulate	descriptive
Create project	Create a new project.
Edit project	Edit the name and profile of the project.
Add Member	Add existing users to the current project and assign permissions, supporting read-only and read-write permissions: • Read-only: Allows the user to view the resources under the current project.

	• Read/Write: Allows users to view and manipulate resources under the current project, including creation, modification, deletion, etc. **Note:** Supports adding up to 50 members at once.
Remove member	Remove the user from the project; after removal, the user no longer has viewing and management rights to the resources under the project.
Add Namespace	Adding a namespace to the current project enables project members to view and manage resources under that namespace. • Members with read-only permissions can view resources under the namespace. • Members with read/write permissions can view and manipulate resources under the namespace, including creating, modifying, and deleting. **Note:** Multiple namespaces can be added to a project, and only one project can be added to a namespace.
Remove namespaces	Removes the namespace from the project. After removal, project members no longer have permission to view and manage resources under that namespace.
Delete project	Delete the project. **Notes.** • Deletion is not supported when the following resources exist under the project: Repository, namespace, Endpoint. • Deleting a project will remove all members of the project and synchronize the deletion of templates and alarms under the project, so please proceed with caution.

10.4.2 User

10.4.2.1 General

nKU's users include both local and 3rd-party users:

- Local user: the user created in the platform, support enable, disable, reset password, set as administrator, cancel administrator, join project and other operations.
- 3rd-party users: users synchronized to the platform through 3rd-party authentication, support for setting as administrator, canceling administrator, joining projects and other operations.

User permissions users have different permissions according to their roles, including a total

of three roles: super administrator, administrator, ordinary users:

- admin : Super administrator with all platform privileges.
- Administrator: admin can set the user as administrator, then the user has administrator privileges, in addition to not being able to manage other administrators and 3rd-party authentication, the administrator has all the other privileges of the platform.
- Ordinary User : You need to join the project to have the platform privileges.

# Regular users have operational access to the following functional modules and resources:

▪ Container orchestration

▪ Artifact repository

▪ Container O&M

# Regular users do not have operational access to the following platform-level functional modules or resources:

■ Cluster management cluster management

■ Settings

Note: General users do not have administrative privileges for external networks, but they can use external network resources that are shared globally by admin or shared to their own projects.

# General user permissions to specific resources are determined by the project they are joining and the namespace of the project:

- Project : Ordinary users can view and manage the resources under the project they are added to, and the project can assign read-only or read-write permissions to resources for ordinary users.
  - Read-only permission : Allow normal users to view resources under the project only.
  - Read and Write Permission : Allow normal users to view and operate the resources under the project, including creation, modification, deletion, etc.
- Namespaces : Normal users can view and manage resources in namespaces under the project they are joining.
  - If the project assigns read-only permissions to a regular user, the regular user can view the resources in the namespace
  - If the project assigns read/write permissions to a normal user, the normal user can view and manipulate the resources in the namespace, including creating, deleting, modifying, etc.

10.4.2.2 Creating local user

In the nKU main menu, click Cluster **Management** > **Project Management** > **User** to access the **User** interface. Click **Create User** to go to the **Create User** page.

You can refer to the following example to enter the appropriate contents:

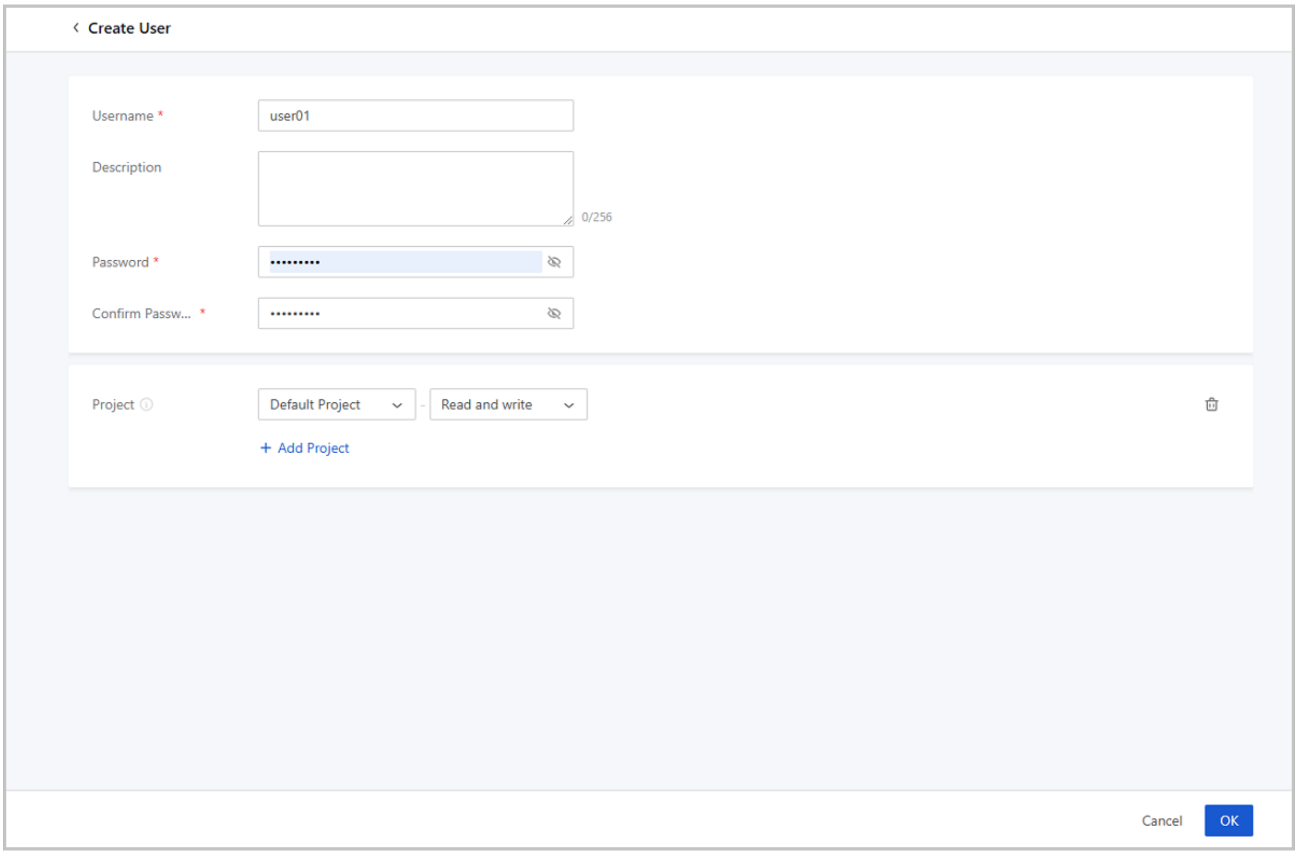
- **User name**: set the user name. Naming rules: length limit 2~30 characters, input can only contain lowercase letters, numbers and special characters (_) (-), and must start with a letter
- **Description**: optional, note user-related information
- **Password**: Set user's password. Password rules: the length is limited to 8~32 characters, including at least 3 kinds of upper case letters, lower case letters, numbers and special characters.
- **Confirm password**: reconfirm the user's password
- **Project**: Select the project that the user joins and set permissions for it. By default, users don't have any permissions, only after joining a project can they have the resource permissions under the project, a user can join multiple projects.

# Read-only permissions: only allows users to view resources under the project.

Read and write permissions: allow users to view and manipulate resources under the project, including creation, modification, and deletion.

As shown in Figure 78: Creating Users:

Figure 78: Creating Users



Create User

Username * user01

Description 0/256

Password * •••••••••

Confirm Passw... * •••••••••

Project Default Project - Read and write

10.4.2.3 Managing user

Managing Local User

In the nKU main menu, click **Cluster Management** > **Project Management** > **Users** to access **the local user** interface.

The following operations are supported for local users:

manipulate	**descriptive**
Create User	Create a new user.
disabled User	Disable the user, and when disabled, the user will not be able to log in to the platform.
Enabled User	Enable the user, and when enabled, the user can log in to the platform normally.

Reset Password	Reset the user password, no need to verify the old password when resetting.
Set as Admin	Setting a user as an administrator gives the administrator all the other privileges of the platform, except for not being able to manage other administrators and 3rd-party authentication.
Revoke Admin	Removing a user as an administrator will remove all of the user's privileges.
Add project	Add users to one or more projects.
Remove from project	Removes the user from the joined project.
Edit Permissions	Modify the user's permission configuration in the project.
User details	Go to the user details page to view basic information about the user and the projects they have joined.
Delete User	Remove the user. When deleted, all the user's privileges will be taken back synchronously.

Managing 3rd-party users

From the nKU main menu, click Cluster **Management** > **Project Management** > **Users** to access **the 3rd-party user** interface.

Third party users support the following operations:

manipulate	**descriptive**
Set as administrator	Setting a user as an administrator gives the administrator all the other privileges of the platform, except for not being able to manage other administrators and 3rd-party authentication.
Cancel Administrator	Removing a user as an administrator will remove all of the user's privileges.
Add project	Add users to one or more projects.
Remove from project	Removes the user from the joined project.
Modify permissions	Modify the user's permission configuration in the project.
User details	Go to the user details page to view basic information about the user and the projects they have joined.

Delete User	Remove the user. When deleted, all the user's privileges will be taken back synchronously. **Note:** After a 3rd-party user is deleted, if the user is not deleted in the 3rd-party authentication center, the platform will synchronously create the user again when the user logs in the platform again.

10.4.3 Namespaces

10.4.3.1 General

Namespaces provide virtual isolation for Kubernetes clusters, where resources in different namespaces are isolated from each other.

Namespaces have the following characteristics:

Resource Segregation and Privilege Control

- Resource isolation: Users can create multiple namespaces according to business requirements, and different namespaces manage resources for different purposes, so as to realize the effective distribution of workspace, for example, resources for the development environment, testing environment, and co-coordination environment are placed under different namespaces.
- Permission Control: Namespaces can be assigned to different projects so that members under the project can view and manage resources under the namespace. A namespace can only be assigned to one project at a time and cannot be viewed and managed by users other than the members and administrators under that project.

Note: When no projects are added to a namespace, Kubernetes resources under that namespace cannot be viewed and managed.

Resource usage control

Users can set resource quotas for namespaces, which can be used to effectively control resource usage when multiple teams or users share cluster resources.

The following resource quotas are supported. When the corresponding resource in the namespace exceeds the set quota, it will not be able to continue to create new resources, and the existing resources will not be affected:

- CPU request quota: the upper limit of the sum of CPU request values for all Pod under this namespace, if not set, there is no limit by default.

- CPU limit quota: the upper limit of the sum of CPU limit values for all Pod under this namespace, if not set, there is no limit by default.
- Memory request quota: the upper limit of the sum of the memory request values for all Pod under this namespace, if not set, there is no limit by default.
- Memory limit quota: the upper limit of the sum of the memory limit values of all Pod under this namespace, if not set, the default is no limit.
- Storage size quota: the upper limit of the total storage size of the pvc that can be created under this namespace, if you do not set it, it is unlimited by default.
- Pod quota: the upper limit of the total number of Pod that can be created under this namespace, if not set, the default is unlimited.
- Service Quota: the upper limit of the total number of services that can be created under this namespace, if you don't set it, there is no limit by default.
- LoadBalance service quota: the upper limit of the total number of LoadBalance services that can be created under this namespace, if not set, the default is unlimited.

  Note: LoadBalance service is a type of service, therefore LoadBalance service quota cannot exceed the service quota.

- ConfigMap Quota: the upper limit of the total number of config sets that can be created under this namespace, if not set, the default is unlimited.
- Secret Quota: the upper limit of the total number of Secret that can be created under this namespace, if not set, the default is unlimited.
- GPU Memory Quota: the total amount of GPU memory of the corresponding manufacturer that can be used under this namespace, if not set, the default is unlimited, and the default is unlimited for unset manufacturers.

Note: If a CPU/Memory resource quota is set for a namespace, the pod CPU/Memory limit value must be specified when the pod is created under that namespace. Users can also set a default CPU/Memory limit value for the namespace so that when a pod is created under that namespace, the pod will use this CPU/Memory limit value by default.

10.4.3.2 Creating namespaces

In the nKU main menu, click **Cluster Management** > **Project Management** > **Namespaces** to enter **the Namespaces** interface. Click **Create Namespace** to bring up the **Create Namespace** screen.

You can refer to the following example to enter the appropriate contents:

- **Name**: set the namespace name. Naming rules: length limit 1-50 characters, can only contain lowercase letters, numbers and separator (-), and must begin with a lowercase letter and end with a number or lowercase letter
- **Project**: adds the namespace to the project, allowing project members to view and manage Kubernetes resources under that namespace

  Note: If not specified manually, the namespace will be added to the default project.

- **Label**: add labels to namespaces in the form of key/value pairs

  # **key:** fill in the label key

  Notes.

  - Typically, key names can contain 1-63 characters, including letters, numbers, and the following special characters: "-", "_", "." but special characters cannot be used at the beginning or end of a key name.
  - Supports prefixing key names and concatenating them with "/", e.g., "k8s.io/app", all characters before the "/" will be considered as the All characters before "/" will be considered as key name prefix, and all characters after will be considered as key name.
  - The key name prefix can contain 1-253 characters, including lowercase letters, numbers and the following special characters "-", "...". , You cannot use two or more special characters in a row, and you cannot use a special character to start or end a prefix.
  - Please note that you do not have to use a key name prefix, but you must fill in the key name.

  # **value**: fill in the label value

- **Annotation**: add annotations to namespaces in the form of key/value pairs

  # **key:** fill in the annotation key

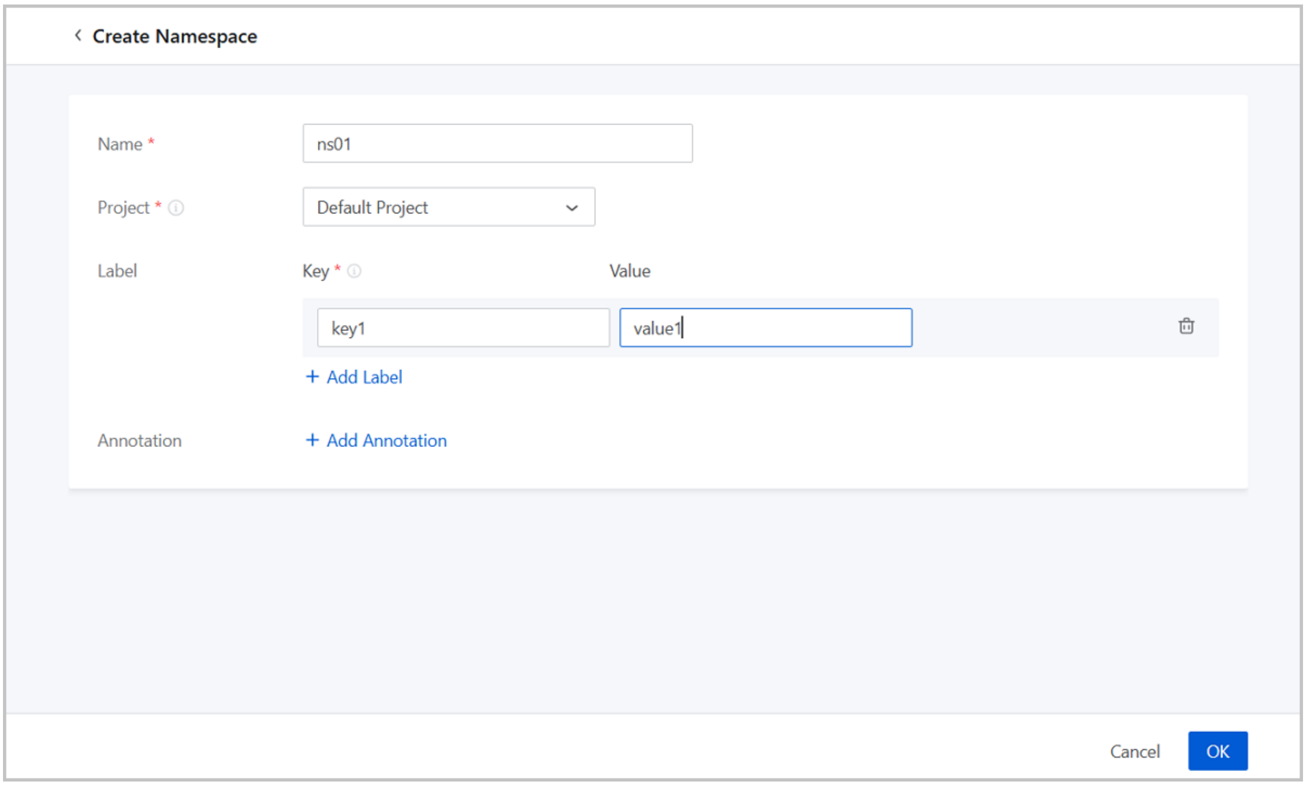
  Notes.

- Typically, key names can contain 1-63 characters, including letters, numbers, and the following special characters: "-", "_", "." but special characters cannot be used at the beginning or end of a key name.
- Supports prefixing key names and concatenating them with "/", e.g., "k8s.io/app", all characters before the "/" will be considered as the All characters before "/" will be considered as key name prefix, and all characters after will be considered as key name.
- The key name prefix can contain 1-253 characters, including lowercase letters, numbers and the following special characters "-", "...". ，不能连续使用 2 个及以上特殊字符，且不能将特殊字符用于前缀开头或结尾。
- Please note that you do not have to use a key name prefix, but you must fill in the key name.

# **value**: fill in the annotation value

As shown in Figure 79: Creating a Namespace:

Figure 79: Creating a Namespace



‹ Create Namespace

Name * ns01

Project * Default Project

Label Key * key1 Value value1

10.4.3.3 Managing namespaces

In the nKU main menu, click Cluster **Management** > **Project Management** > **Namespaces** to access **the Namespaces** interface.

The namespace supports the following operations:

manipulate	descriptive
Create Namespace	Create namespaces under the cluster.
Set resource quotas	Set a resource quota for a namespace, after setting it, the number of resources created/used under that namespace cannot exceed that quota.
Set default resource limits	Sets the default CPU/memory resource limit for Pod under the namespace. When set, Pod created under this namespace will use this CPU/memory limit by default.
Join project	Adding a namespace to a project allows project members to view and manage Kubernetes resources under that namespace: • Project members with read-only permissions can view Kubernetes resources under this namespace. • Project members with read and write permissions can view, create, modify, or delete Kubernetes resources under this namespace.
Remove from project	Remove the namespace from the project. After removal, the original project members cannot continue to view and manage Kubernetes resources under that namespace. **Note:** When a namespace does not belong to any project, Kubernetes resources under that namespace cannot be viewed and managed.
Edit label/annotation	Editing namespace labels, annotations
Namespace details	Go to the Namespace Details page to view basic information about the namespace.
Delete namespace	Remove the namespace from the cluster. **Note:** This operation will synchronize the deletion of all resources under the current namespace, so please be careful.

11 Settings

11.1 Theme Appearance

11.1.1 General

Users can customize the appearance of the platform theme, such as the platform title, Logo, etc.

11.1.2 Customizing theme appearance

This section describes how to customize the platform's Logo and release information.

1. Enter the Theme Appearance screen

In the nKU main menu, click **Settings > Theme Appearance** to access the T**heme Appearance** screen.

2. On the Theme Appearance screen, you can refer to the following example to set the appropriate contents:

- Global Appearance

# Appearance Theme: Set the interface theme color, eight theme colors are available.

Note: The theme color is effective for the interface appearance of all administrators and users of the platform.

- Title Setting

# Browser:

■ Chinese title: customize the browser's Chinese title text

Note: Title text should be 30 characters or less.

■ English title: customize the English title text of the browser

Note: Title text should be 30 characters or less.

■ Favicon: customize browser icons

Note: Only .ico format is supported, and the file size cannot exceed 2M.

# Login portal:

■ Chinese title: customize the Chinese title text of the login screen

Note: Title text should be 30 characters or less.

■ English title: customize the English title text of the login screen

Note: Title text should be 30 characters or less.

■ Logo: Customized Logo image for login screen

Note: Only .jpg/ .jpeg/ .png/ .svg format is supported, the size of the image should be within 250*70px, and the file size should not exceed 2M.

# Platform Interface:

▪ Chinese title: customize the Chinese title text of the platform interface

Note: Title text should be 30 characters or less.

▪ English title: customize the English title text of the platform interface

Note: Title text should be 30 characters or less.

■ Text size: customize the text size of the platform interface title, can be set to large, medium and small, the default is medium

■ Logo: Customize the Platform Interface Header Logo Image

Notes.

- Support .jpg/ .jpeg/ .png/ .svg format, the image size should be within 110*40 px, and the file size should not exceed 2M.
- On dark backgrounds, it is recommended to use white or light-colored Logo.

3. Reset theme appearance

In the nKU main menu, click **Settings** > **Theme Appearance** to enter the **Theme Appearance** interface. Click **Reset to Default Settings** to clear the current customization settings with one click and revert to the default settings.

11.2 Email server

11.2.1 General

Email server: The platform creates Endpoint of email type and needs to set up a email server for sending alarm messages and other message emails.

11.2.2 Adding a email server

In the nKU main menu, click **Settings** > **Email Servers** to enter **the Email Servers** interface. Click **Add Email Server** to enter the **Add Email Server** interface.

You can refer to the following example to enter the appropriate contents:

- **Name**: set the name of email server. Naming specification: the length is limited to 1~128 characters, and the input content can only contain Chinese characters, English letters, numbers and the following 7 kinds of English characters (-) (_) (.) (() ()) (:) (+)
- **Description**: optional, note email server related information
- **Email server type**: system default is SMTP
- **Email server**: enter the email server address
- **Port**: input email server port, default is 25
- **EMail Username**: enter the user name of email server, the sender will use this user name when sending mails.
- **Password**: Enter the password corresponding to the user name

  Note: If you use the third party email as email server, you need to enable SMTP service in the third party email settings in advance, and you should fill in the obtained authorization code in the password box.

- **Encryption type**: optional, set encrypted connection to email server port, encryption type: STARTTLS, SSL/TLS, NONE

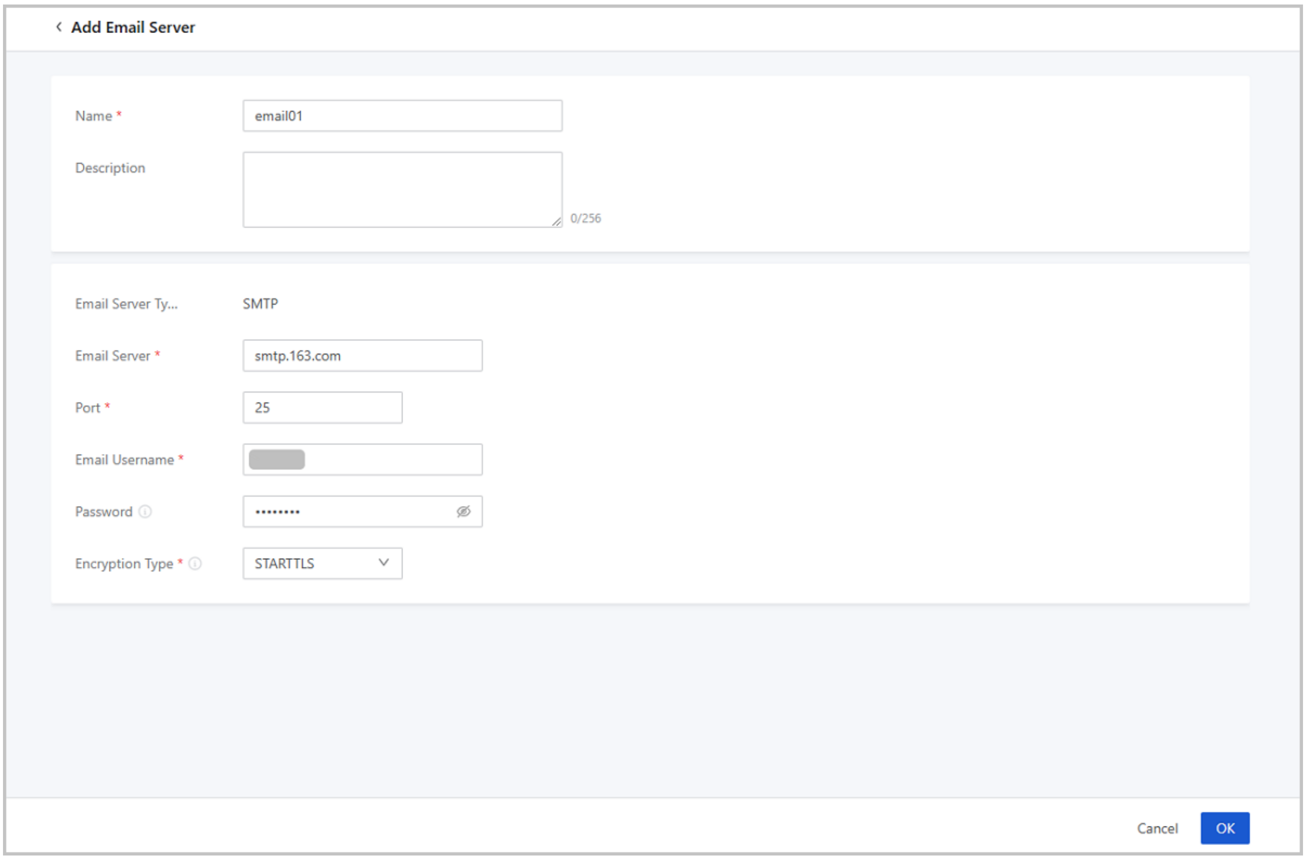
  # Select the STARTTLS encryption type by default. port 25.

  # Port default 465 when SSL/TLS encryption type is selected.

  # If the SMTP server does not use encrypted connections, you can select NONE.

As shown in Figure 81: Adding a Email Server:

Figure 81: Adding a Email Server



‹ Add Email Server

Name * email01

Description

0/256

Email Server Ty... SMTP

Email Server * smtp.163.com

Port * 25

Email Username * [redacted]

Password ⓘ ••••••••

Encryption Type * ⓘ STARTTLS

11.2.3 Managing email servers

Click **Setup > Email Server** to enter the **Email Server** screen.

The email server supports the following operations:

manipulate	**descriptive**
Edit Configuration	Modify the name, profile, email server, email server port, mail username, password, and encryption type of the email server.
Delete email server	Remove the email server. **Note:** After deleting the email server, the messages that need to be sent by e-mail will not be sent, so please be careful with the operation.

11.3 3rd-party authentication

11.3.1 General

3rd-party authentication is a 3rd-party login authentication service provided by the platform, which supports seamless access to the 3rd-party login authentication system, so that the corresponding 3rd-party users can log in to the platform in a single point without password and conveniently use the platform resources.

11.3.2 Adding a 3rd-Party Authentication Server

In the nKU main menu, click **Settings** > **3rd-party authentication** to enter the **3rd-party authentication** screen. Click **Add 3rd-Party Authentication Server** to enter the **Add 3rd-Party Authentication Server** screen.

Adding a 3rd-party authentication server is divided into the following types:

- Add an OIDC server
- Add a CAS server

Adding an OIDC Server

You can refer to the following example to enter the appropriate contents:

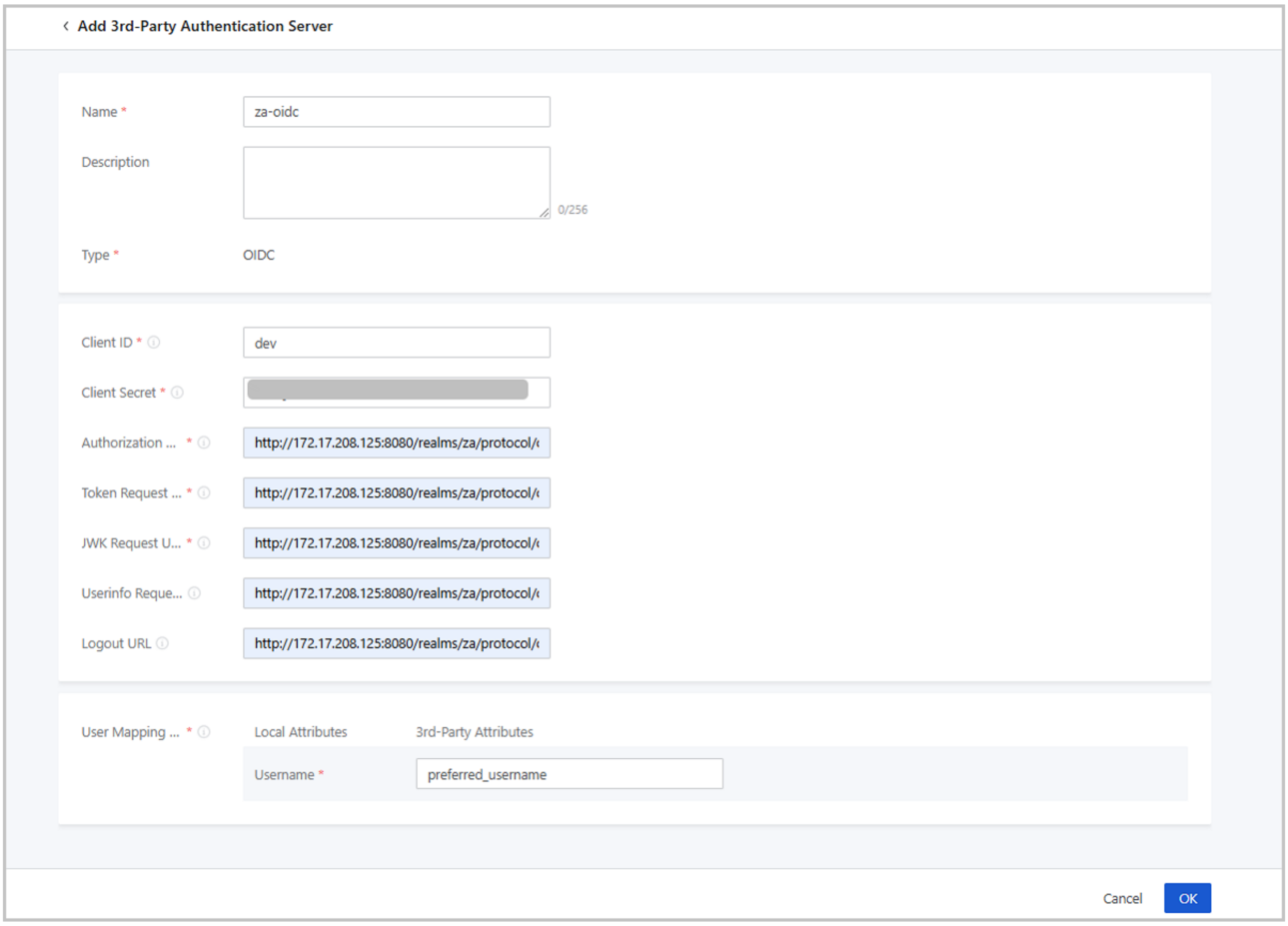
- **Name**: Set the 3rd-Party Authentication Server name. Naming specification： Length limit 1~128 characters, input content can only contain Chinese characters, English letters, numbers and the following 7 English characters (-) (_) (.) (() ()) (:) (+)
- **Description**: Optional , note 3rd-Party Authentication Server related information
- **Type**: Select OIDC
- **Client ID**: unique identifier assigned to the platform by the authentication system
- **Client Secret**: key assigned to the platform by the authentication system
- **Authorization Request URL**: request key for obtaining authorization in Authorization Code mode.
- **Token Request URL**: request URL to get an Access Token from the authentication server
- **JWK Request URL**: the request URL to get the signing key from the certification server
- **Userinfo Request URL**: the request URL for obtaining user information from the authentication server
- **Logout URL**: Used to call Logout URL to cancel the session after logging out of the platform, so that you need to log in to the 3rd-party authentication system again when logging in to the platform next time. If left blank, the logout information will not be cleaned up immediately after logging out of the platform, and you can still log in to the platform again within the validity period of the session without password.

- **User mapping Rule**: 3rd-party users synchronized to the platform will have local attributes of the platform. Mapping rules are used to establish the mapping relationship between 3rd-party attributes and local attributes.

  # **Username:** Sets the mapping between the platform user's user name and an attribute of the user in the OIDC authentication server. The username uniquely identifies a user globally within the platform, and it is necessary to ensure that the authentication system is filled in with a user attribute that also uniquely identifies the user. For example, if the username maps to preferred_username, the username of the user synchronized to the platform will use the corresponding value of preferred_username (e.g., xiaoming).

As shown in Figure 82: Adding OIDC Server:

Figure 82: Adding OIDC Server



‹ Add 3rd-Party Authentication Server

Name * za-oidc

Description 0/256

Type * OIDC

Client ID * ⓘ dev

Client Secret * ⓘ [redacted]

Authorization ... * ⓘ http://172.17.208.125:8080/realms/za/protocol/(

Token Request ... * ⓘ http://172.17.208.125:8080/realms/za/protocol/(

JWK Request U... * ⓘ http://172.17.208.125:8080/realms/za/protocol/(

Userinfo Reque... ⓘ http://172.17.208.125:8080/realms/za/protocol/(

Logout URL ⓘ http://172.17.208.125:8080/realms/za/protocol/(

User Mapping ... * ⓘ

Local Attributes	3rd-Party Attributes
Username *	preferred_username

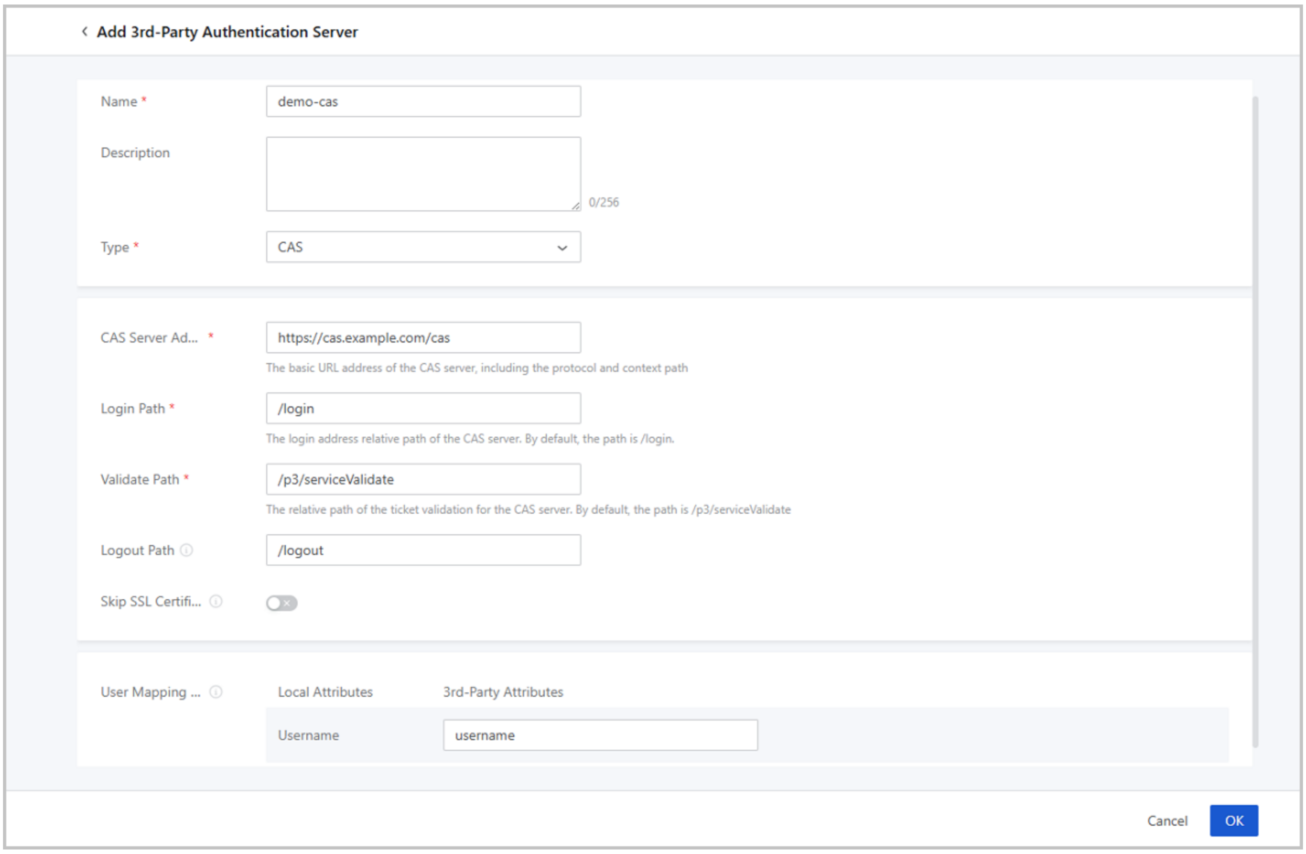
Adding a CAS Server

Refer to the following examples to enter the corresponding information:

- **Name**: Set the name of the 3rd-party authentication server. Naming rules: Length is limited to 1–128 characters. The input can only contain Chinese characters, English letters, numbers, and the following 7 English characters: (-) (_) (.) (() ()) (:) (+)
- **Description (Optional)**: Add remarks about the 3rd-party authentication server.
- **Type**: Select CAS.
- **CAS Server Address**: The base URL of the CAS server, which must include the protocol and context path (e.g.,https://cas.example.com/cas).
- **Login Path**: The relative path of the CAS server's login address. The default is generally /login.
- **Validation Path**: The relative path of the CAS server's ticket validation address. The default is generally /p3/serviceValidate.
- **Logout Path (Optional)**: The relative path of the CAS server's logout address. The default is generally /logout. It is used to call and invalidate the session after logging out of the platform. You will need to re-log in to the 3rd-party authentication system when logging into the platform next time. If left blank, the login information will not be cleared immediately after logging out of the platform, and you can log in again without entering credentials within the session validity period.
- **Skip SSL Certificate Check**: Set whether to skip SSL certificate checks. If enabled, the platform will skip all SSL certificate checks when connecting to the CAS server.
- **User Mapping Rules**: After 3rd-party users are synchronized to the platform, they will have local platform attributes. Mapping rules are used to establish the mapping relationship between 3rd-party attributes and local attributes.
    - **Username**: Set the mapping relationship between the platform username and a user attribute in the CAS authentication server. The username uniquely identifies a user globally on the platform. Ensure the user attribute filled in has a unique identifier in the authentication system. Example: If the local username is mapped to the 3rd-party "username", the username of the 3rd-party user on this platform will use the corresponding "username" value from the 3rd-party platform.

As shown in Figure 82: Adding OIDC Server:

Figure 82: Adding OIDC Server



‹ Add 3rd-Party Authentication Server

Name * demo-cas

Description 0/256

Type * CAS

CAS Server Ad... * https://cas.example.com/cas
The basic URL address of the CAS server, including the protocol and context path

Login Path * /login
The login address relative path of the CAS server. By default, the path is /login.

Validate Path * /p3/serviceValidate
The relative path of the ticket validation for the CAS server. By default, the path is /p3/serviceValidate

Logout Path /logout

Skip SSL Certifi...

User Mapping ...

Local Attributes	3rd-Party Attributes
Username	username

Cancel OK

11.3.3 Managing 3rd-Party Authentication Servers

In the nKU main menu, click **Settings > 3rd-party authentication** to access the **3rd-party authentication** screen.

The 3rd-Party Authentication Server supports the following operations:

manipulate	descriptive
Edit Configuration	Modify the name, profile, and other configuration information of the 3rd-Party Authentication Server. **Notes.** • If the 3rd-Party Authentication Server is replaced, the system will synchronize the users under the new server to this platform, and the users who have been synchronized before the server is replaced will not be able to continue to log in this platform. • If a user under the new server is renamed to a previously synchronized user, the user will inherit all the privileges of the original user with the same name on this platform.

	• If the 3rd-Party Authentication Server has not been changed, users who have been synchronized can still log in to the Platform normally and will not be affected.
Delete 3rd-Party Authentication Server	Remove the 3rd-Party Authentication Server. **Notes.** • Before deleting the 3rd-party authentication Server, you need to delete all the 3rd-party users in the platform, delete the 3rd-party users in the platform, and the users in the source 3rd-party authentication Server are not affected. • After deletion, users will not be able to log in to this platform through the 3rd-Party Authentication Server, so please be careful.

11.4 AccessKey Management

11.4.1 General

An AccessKey is an identity credential for accessing the platform APIs with platform privileges consistent with the current user, including: AccessKey ID and AccessKey Secret.

Notes.

- The AccessKey is a key factor in the platform's secure authentication of API requests, so please keep it safe.
- If there is a risk of leakage of a particular AccessKey, it is recommended that the AccessKey be deleted and a new AccessKey be generated in a timely manner.
- AccessKey has full rights of the creator, each user including administrators can only view and manage the AccessKey they created.
- Users can enable, disable or delete their own created AccessKey at any time.

11.4.2 Managing AccessKey

In the nKU main menu, click **Settings** > **AccessKey Management** to access **the AccessKey Management** screen.

AccessKey supports the following operations:

manipulate	**descriptive**
Generate AccessKey	Generate a new AccessKey.
Enabled AccessKey	Enables an AccessKey that is in the deactivated state.
Disabled AccessKey	Deactivate the selected AccessKey. A deactivated AccessKey cannot call the API to access platform resources.
Delete AccessKey	Delete the selected AccessKey. the deleted AccessKey cannot call the API to access platform resources.

12 Best Practices

12.1 Deploying applications in a conventional manner

background information

This section describes the step-by-step deployment of a web application on nKU in the usual way.

This scenario takes the deployment of a WordPress application as an example. WordPress is a blogging platform developed using the PHP language and needs to be used with MySQL. WordPress is used to run the content management program and MySQL is used as the database to store the data.

Deployment of a WordPress application in the regular way is divided into the following three steps:

1. Upload Image
2. Deployment of MySQL
3. Deployment of WordPress

procedure

1. Uploading Images

a) Preparing MySQL and WordPress Images

In this scenario, MySQL and WordPress images can be obtained directly from Docker Hub and do not need to be made separately.

Note: In other scenarios, users can make the required image via dockerfile or save an existing container as an image for use.

b) Creating a Repository

In the nKU main menu, click **Artifact Repository** > **Repository**. In the **Repository** screen, click **Create Repository** to bring up the **Create Repository** screen.

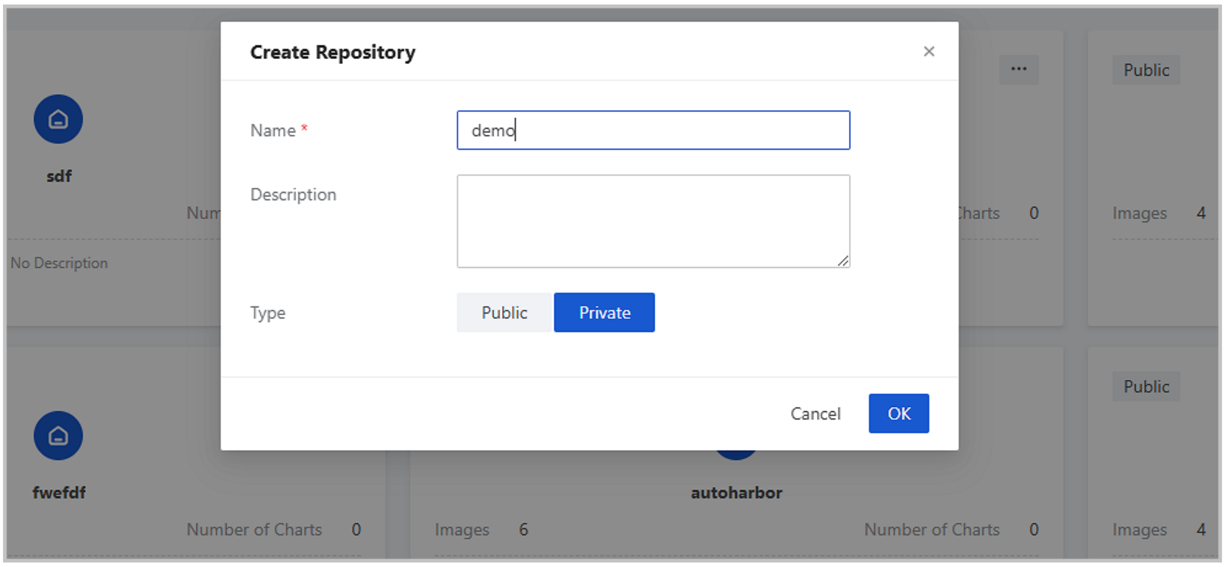
You can refer to the following example to enter the appropriate contents:

- **Name**: Set repository name

- **Description**: optional, repository related information for the Repository
- **Type**: This scenario is set to private

As shown in Figure 83: Creating a Repository:

Figure 83: Creating a Repository



Create Repository

Name * demo

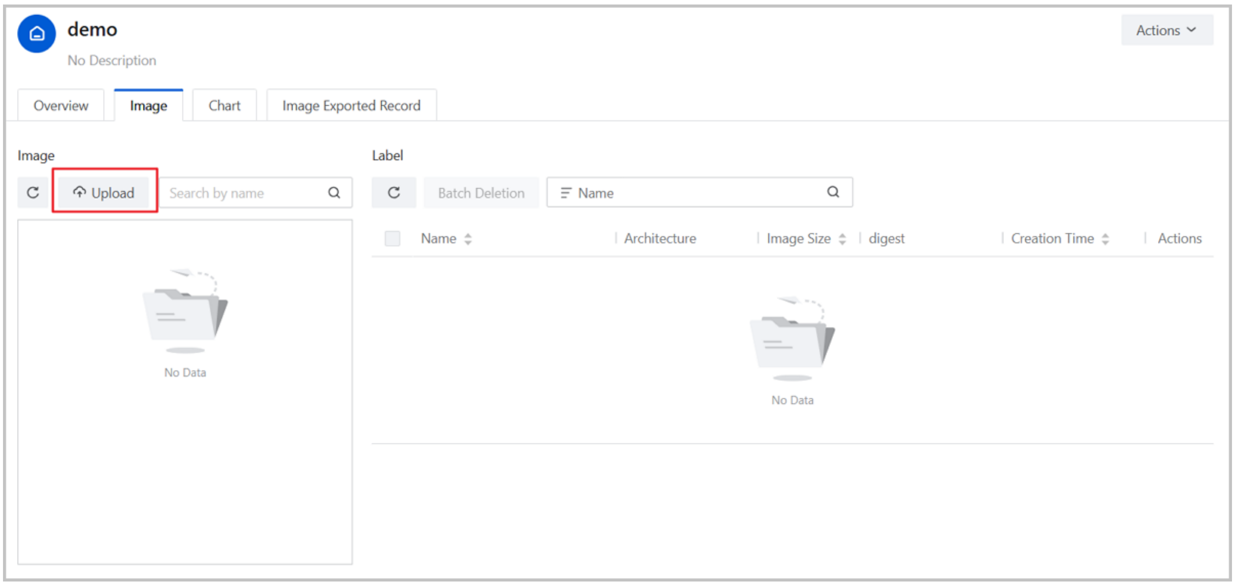
Description

Type: Public / Private (selected)

c) Upload the MySQL image to the Repository Go to the **Image** tag on the Repository details page and click **Upload**.

As shown in Figure 84: Uploading a Image Image:

Figure 84: Uploading a Image

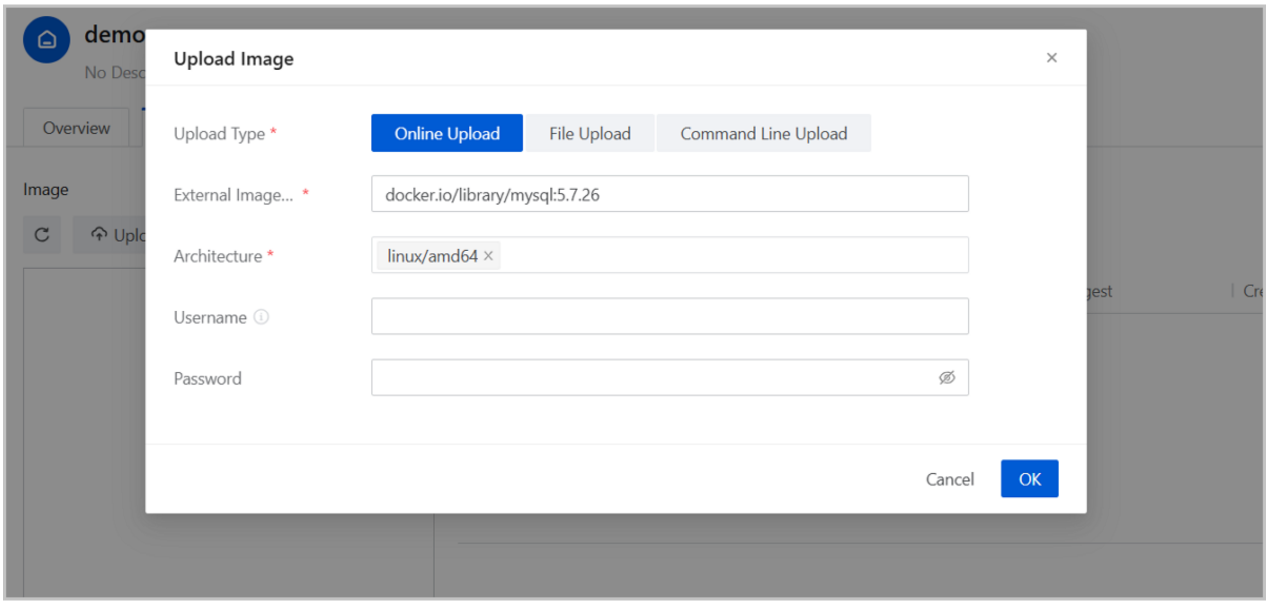


In the **Upload Image** screen, refer to the following example to enter the appropriate content:

- **Upload Type**: Online Upload is selected for this scenario
- **External Image address**: this scenario fills docker.io/library/mysql:5.7.26
- **Architecture**: Select the same architecture as the node where the application will be deployed
- **Username**: this scenario can be left blank
- **Password**: this scene can be left blank

As shown in Figure 85: Uploading a MySQL image:

Figure 85: Uploading a MySQL Image



Upload Image

Upload Type * Online Upload | File Upload | Command Line Upload

External Image... * docker.io/library/mysql:5.7.26

Architecture * linux/amd64 ×

Username

Password

d) Uploading a WordPress image repository to your Repository

Go to the **image** repository label on the Repository details page and click **Upload**.

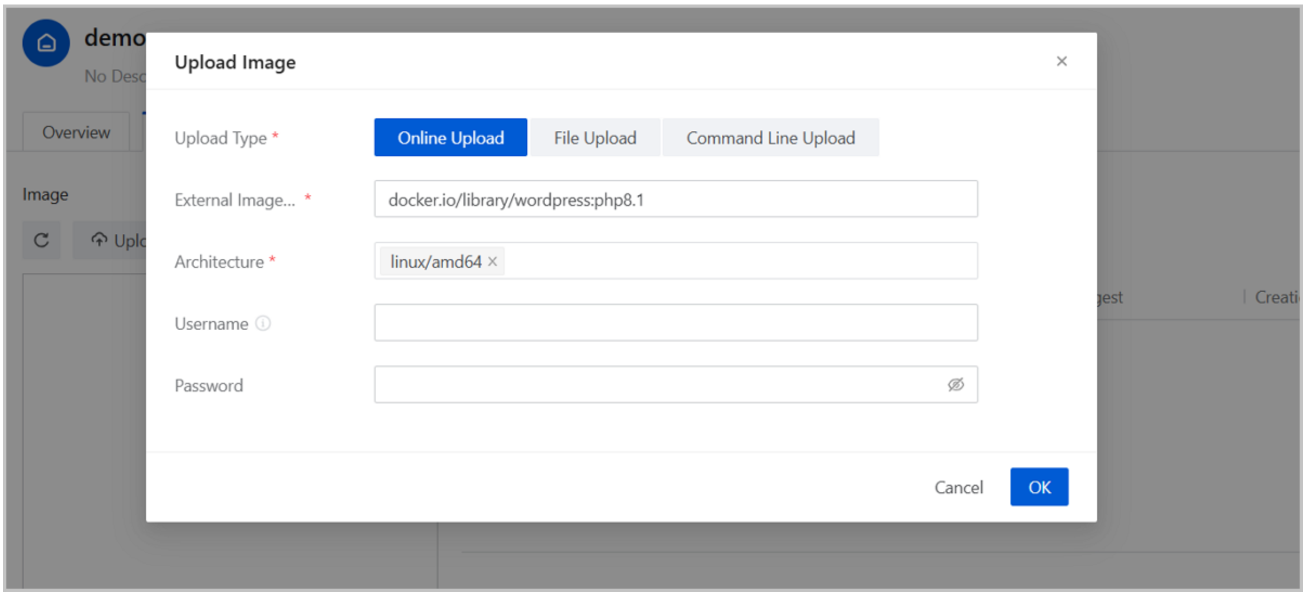
On the **Upload Image** screen, refer to the following example to enter the appropriate content:

- **Upload Type**: Online Upload is selected for this scenario
- **External Image address**: this scene fill docker.io/library/wordpress:php8.1
- **Architecture**: Select the same architecture as the node where the application will be deployed
- **Username**: this scenario can be left blank

- **Password**: this scene can be left blank

As shown in Figure 86: Uploading a WordPress image:

Figure 86: Uploading a WordPress image



Upload Image

Upload Type *: Online Upload

External Image... *: docker.io/library/wordpress:php8.1

Architecture *: linux/amd64

Username:

Password:

Note: In addition to online upload, nKU also supports file upload, command line upload two image repository upload, users can upload the image zip to the Repository, or through the command to push the image directly to the Repository, for details, please refer to Uploading Images.

2. Deployment of MySQL

a) Creating a pvc

In the nKU main menu, select **Container Orchestration** > **PersistentVolumes management** > **PVC**. In the **PVC** screen, click **Create PVC** to bring up the **Create PVC** screen.

Refer to the following example to enter the appropriate contents:

- **Name**: set to data-mysql in this scenario
- **StorageClass**: Select the storage class used to create the pvc volume.

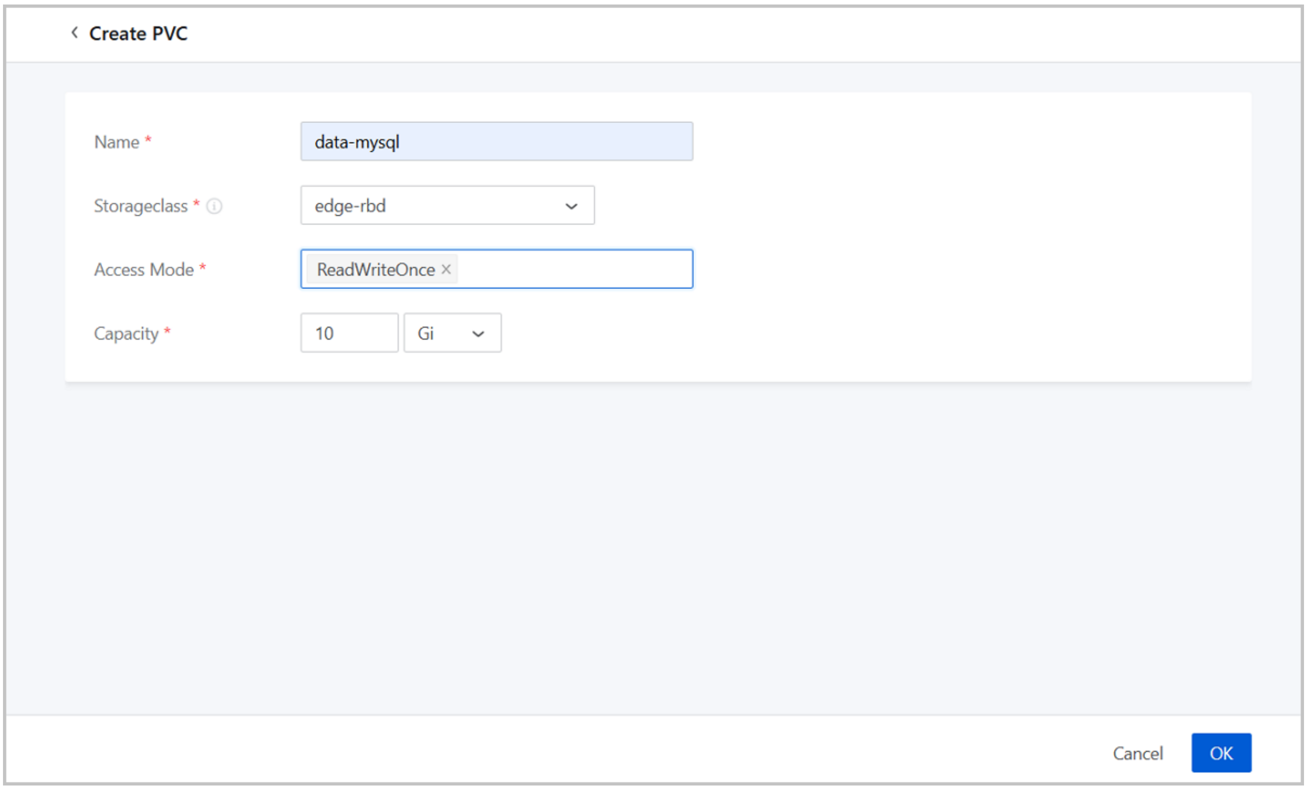
Notes.

- A storage class is a configuration template used in Kubernetes to dynamically create a pvc, defining the properties of the pvc, the creation policy, and the required storage plugins.

  - o Storage classes are created and managed by the administrator, if there are no available storage classes, please contact the administrator.

- **Access Mode**: selects the access mode allowed for the pvc. The setting for this scenario is ReadWriteOnce
- **Capacity**: Set the appropriate storage size of the pvc, unit: Gi, Ti
- As shown in Figure 87: Creating a pvc:

Figure 87: Creating a Pvc



Name * data-mysql

Storageclass * ⓘ edge-rbd

Access Mode * ReadWriteOnce ×

Capacity * 10 Gi

b) Creating workloads

In the nKU main menu, click **Container Orchestration** > **Application Management** > **Workloads** to enter the **workload** interface. Click **Create Workload** to bring up the **Create Workload** screen.

1. The basic information can be entered

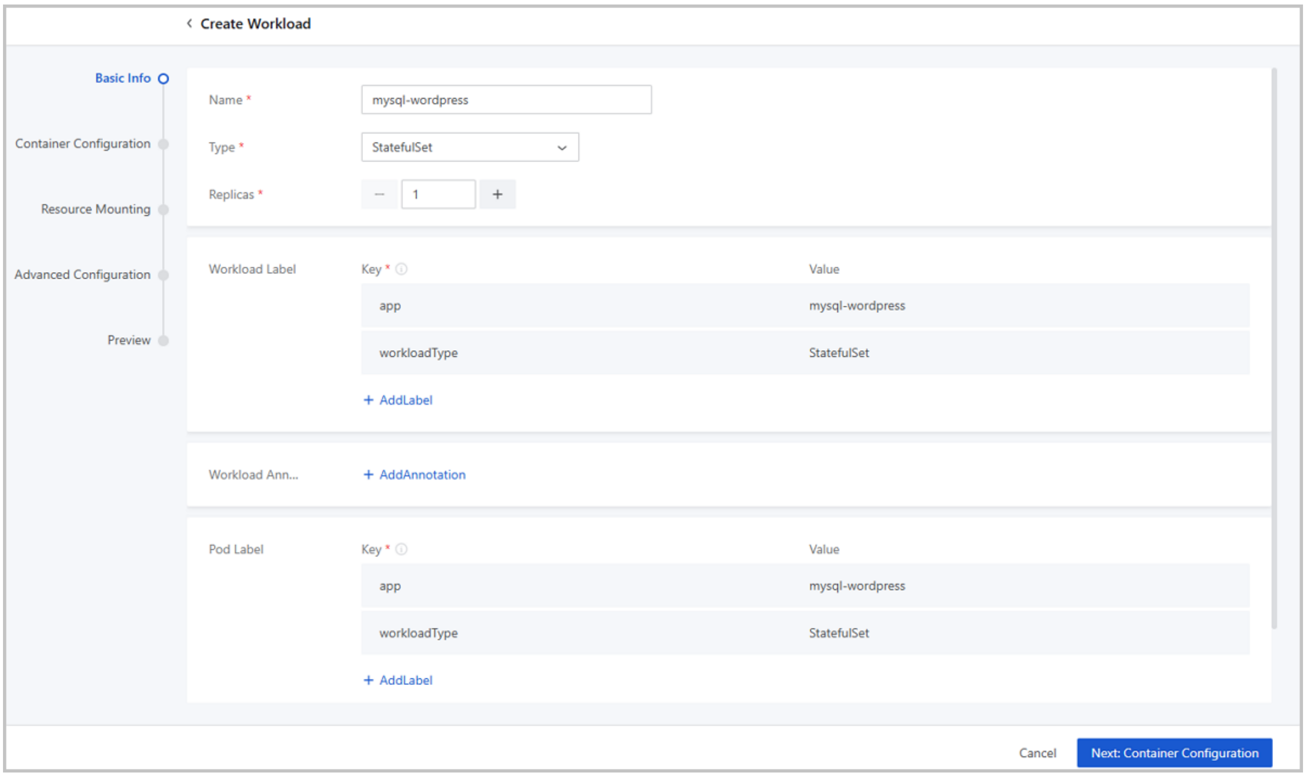
as shown in the following example:

- **Name**: set to mysql-wordpress for this scenario
- **Type**: Select StatefulSet
- **Replicas**: set to 1

- Default values can be used for other settings.

As shown in Figure 88: Creating a MySQL Workload-Basic Information:

Figure 88: Creating a MySQL Workload-Basic Information



Name * mysql-wordpress

Type * StatefulSet

Replicas * — 1 +

Workload Label

Key *	Value
app	mysql-wordpress
workloadType	StatefulSet

Workload Ann...

Pod Label

Key *	Value
app	mysql-wordpress
workloadType	StatefulSet

2. Container Configuration

   Click **Add Container** and refer to the following example to enter the appropriate content:

   - **Container name**: set the container name
   - **Container type**: Selection of the worker container
   - **Image Source**: select Repository
   - **Image**: Select MySQL Image
   - **Tag**: select 5.7.26
   - **Resource Request**: may be left blank
   - **Resource Limit**: may be left blank
   - **Command**: can be left blank
   - **Argument**: fill in **--ignore-db-dir=lost+found**

- **Env**: Select **custom key-value pairs** to add the following four environment variables:

  # Key: MYSQL_ROOT_PASSWORD; Value: MySQL root user password

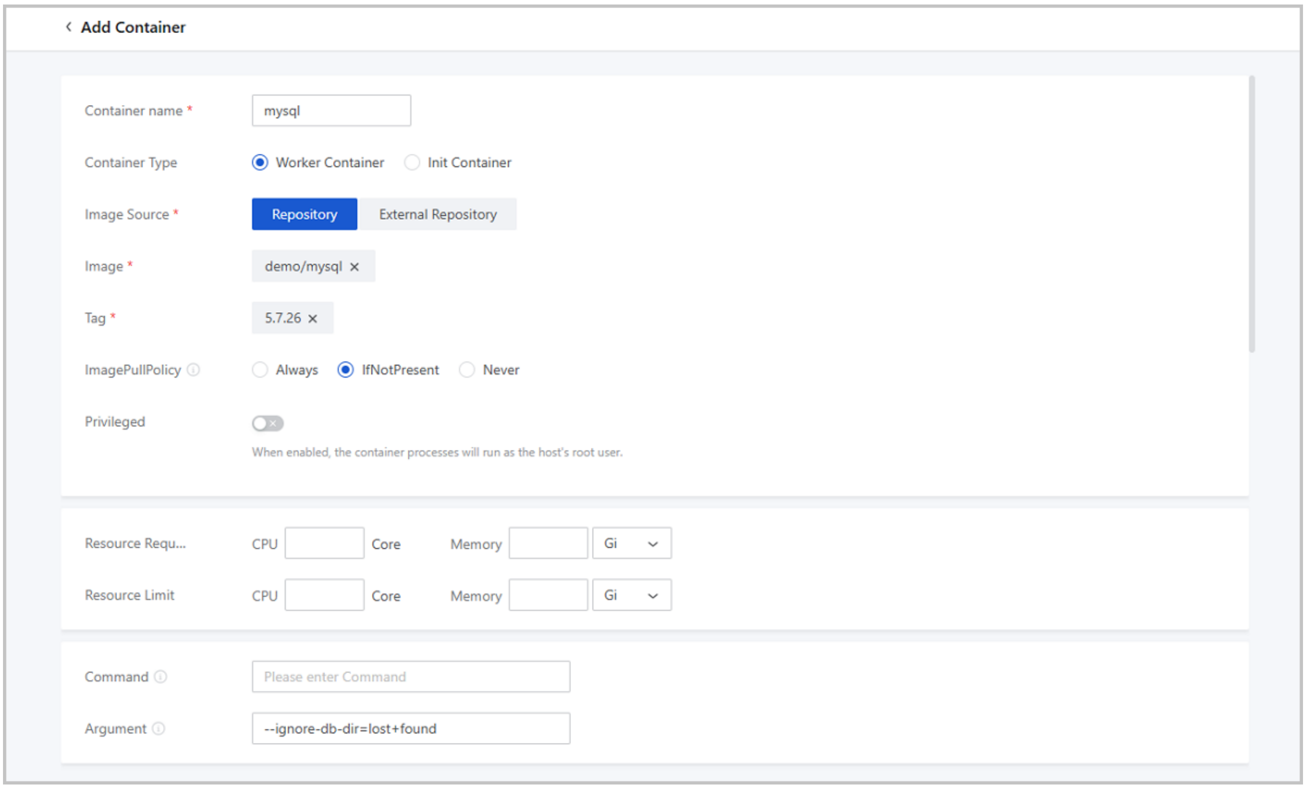
  # Key: MYSQL_DATABASE; Value: the name of the database to be created when MySQL starts up

  # Key: MYSQL_USER; Value: database user name

  # Key: MYSQL_PASSWORD; Value: database user password
- **LivenessProbe**: not enabled in this scenario
- **ReadinessProbe**: not enabled in this scenario
- **StartupProbe**: not enabled for this scenario
- **PostStart**: not enabled for this scenario
- **PreStop**: not enabled in this scenario

As shown in Figure 89: Creating a MySQL workload-container configuration:

Figure 89: Creating a MySQL workload-container configuration



‹ Add Container

Container name * mysql

Container Type ● Worker Container ○ Init Container

Image Source * [✓ Repository] External Repository

Image * demo/mysql ×

Tag * 5.7.26 ×

ImagePullPolicy ○ Always ● IfNotPresent ○ Never

Privileged

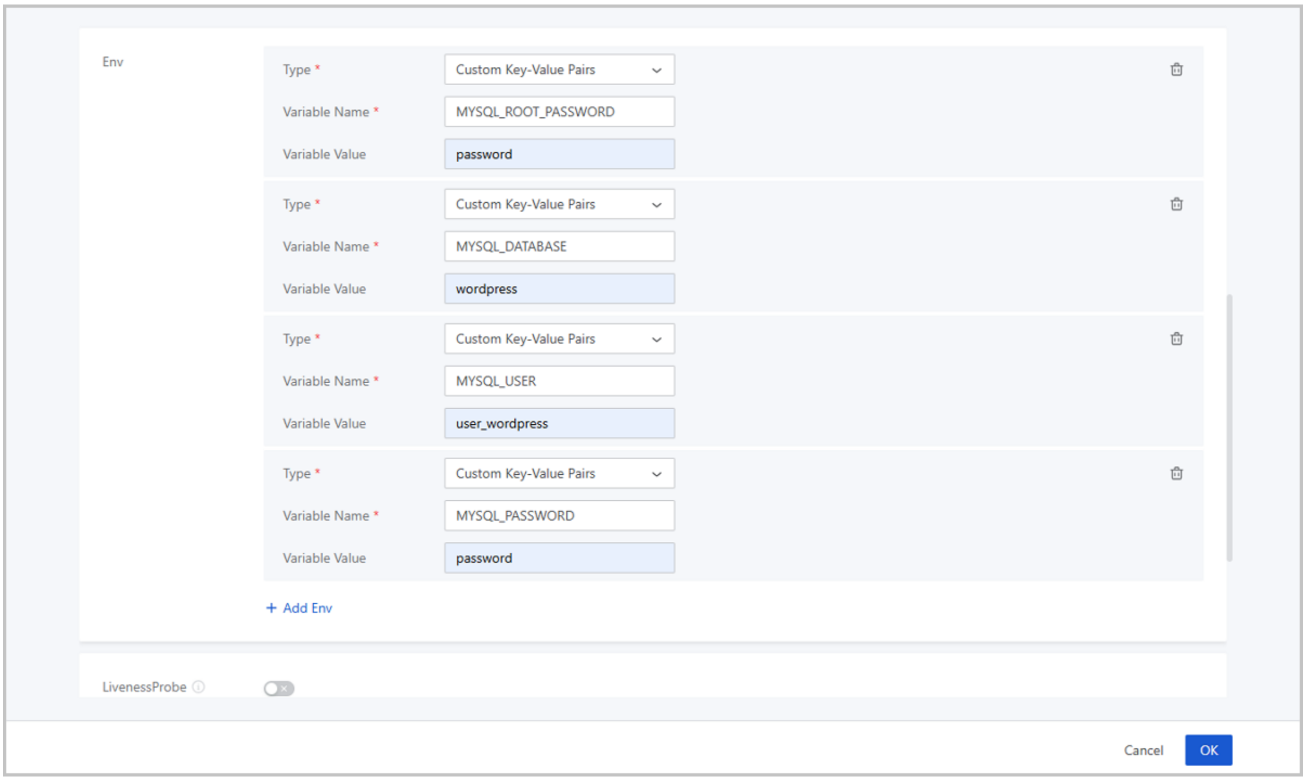
When enabled, the container processes will run as the host's root user.

Resource Requ... CPU Core Memory Gi

Resource Limit CPU Core Memory Gi

Command Please enter Command

Argument --ignore-db-dir=lost+found



Env

Type * Custom Key-Value Pairs
Variable Name * MYSQL_ROOT_PASSWORD
Variable Value password

Type * Custom Key-Value Pairs
Variable Name * MYSQL_DATABASE
Variable Value wordpress

Type * Custom Key-Value Pairs
Variable Name * MYSQL_USER
Variable Value user_wordpress

Type * Custom Key-Value Pairs
Variable Name * MYSQL_PASSWORD
Variable Value password

3. resource mounting

- Mount PVC: click **Add PVC** and refer to the following example to enter the corresponding contents:

  # **PVC**: Select the pvc created in step a.

  # **Container**: select the container created in the container configuration

  # **Mount path**: fill in /var/lib/mysql.

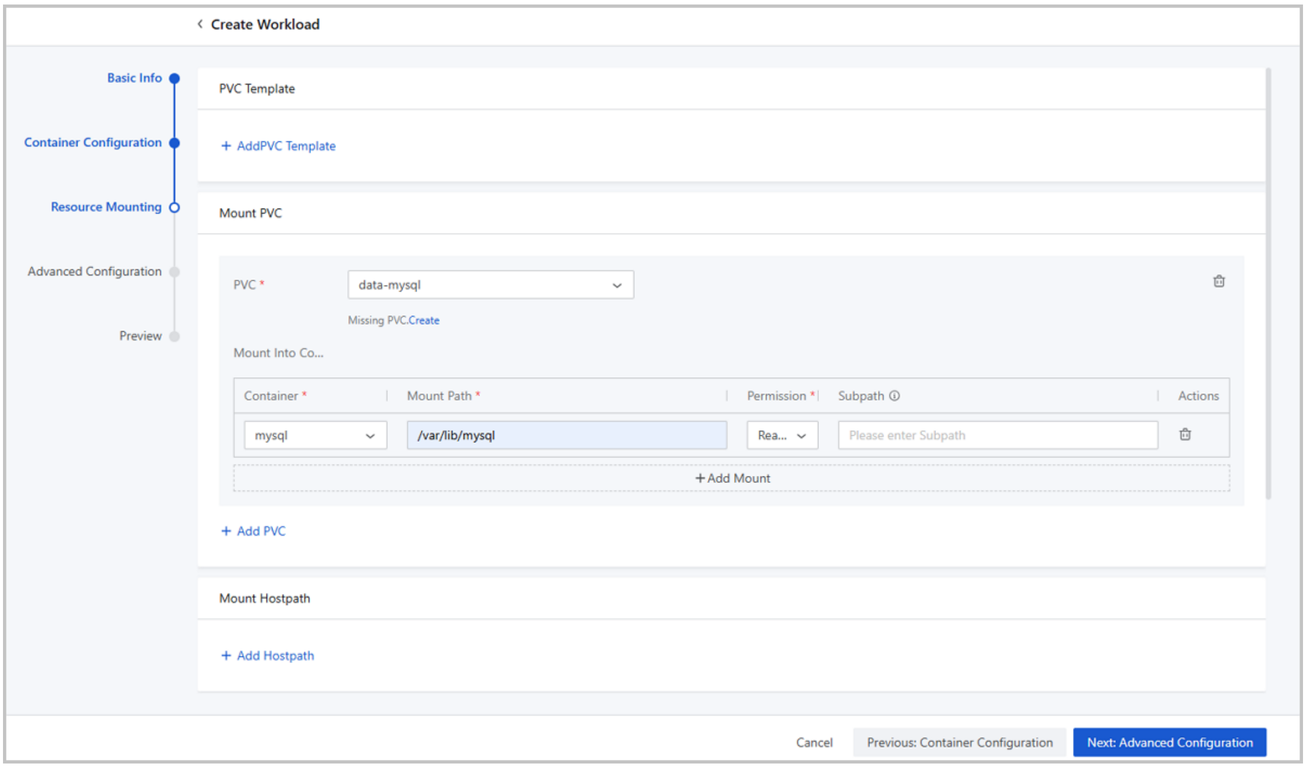
  # **Permission**: set to read/write

  # **SubPath**: not filled in for this scenario

- No other resources are mounted in this scenario.

As shown in Figure 90: Creating a MySQL Workload - Resource Mounting:

Figure 90: Creating a MySQL Workload - Resource Mounting



PVC *: data-mysql

Mount Into Co...

Container *	Mount Path *	Permission *	Subpath	Actions
mysql	/var/lib/mysql	Rea...	Please enter Subpath	

4. Advanced Configuration: In this scenario, the default configuration is maintained
5. Preview

   View the workload that will be created and confirm the creation.

c) Creating services

In the nKU main menu, click **Container Orchestration** > **Network Management** > **Service** to enter the **Service** interface. Click **Create Service** to bring up the **Create Service** screen.

You can refer to the following example to enter the appropriate contents:

- **Name**: This scenario is set to mysql-wordpress
- **Associated Workload**: select the workload created in step b
- **Type**: Select ClusterIP

  Note: In this scenario, the MySQL service is mainly used by the WordPress service, which is accessed within the cluster, so you can select the service type as ClusterIP.
- **IP Type:** Select IPv4
- **Service Port**: Set the following information:

# **Port**: 3306 is set for this scenario

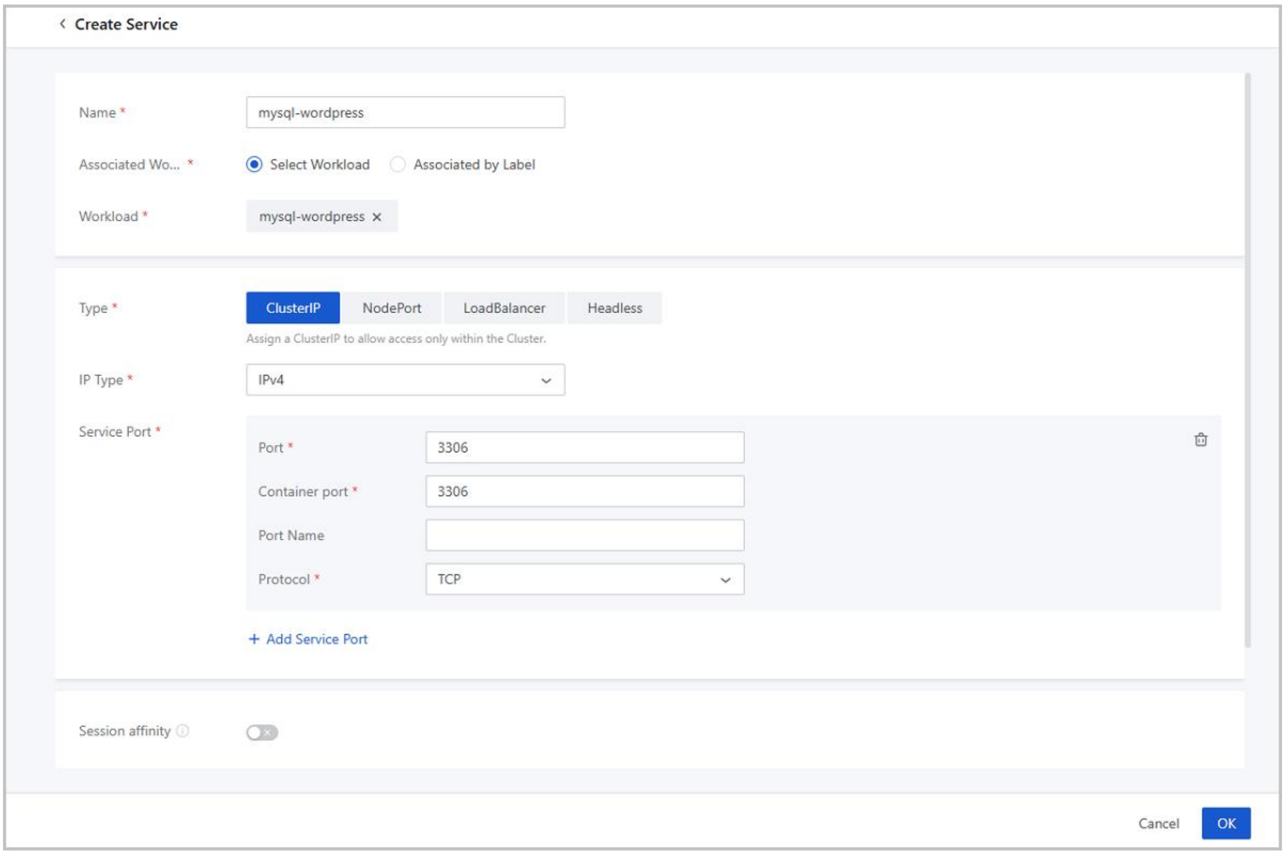
# **Container port**: set to 3306 for this scenario

# **Protocol**: This scenario is set to TCP

- **Session affinity**: Not enabled for this scenario

As shown in Figure 91: Creating the MySQL service:

Figure 91: Creating the MySQL service



‹ Create Service

Name *: mysql-wordpress

Associated Wo... *: ● Select Workload ○ Associated by Label

Workload *: mysql-wordpress ×

Type *: ClusterIP | NodePort | LoadBalancer | Headless

Assign a ClusterIP to allow access only within the Cluster.

IP Type *: IPv4

Service Port *:

Port *: 3306

Container port *: 3306

Port Name:

Protocol *: TCP

+ Add Service Port

Session affinity: off

3. Deploy WordPress

a) Create workloads

In the nKU main menu, click **Container Orchestration** > **Application Management** > **Workload** to enter the **Workload** interface. Click **Create Workload** to bring up the **Create Workload** screen.

1. Basic information

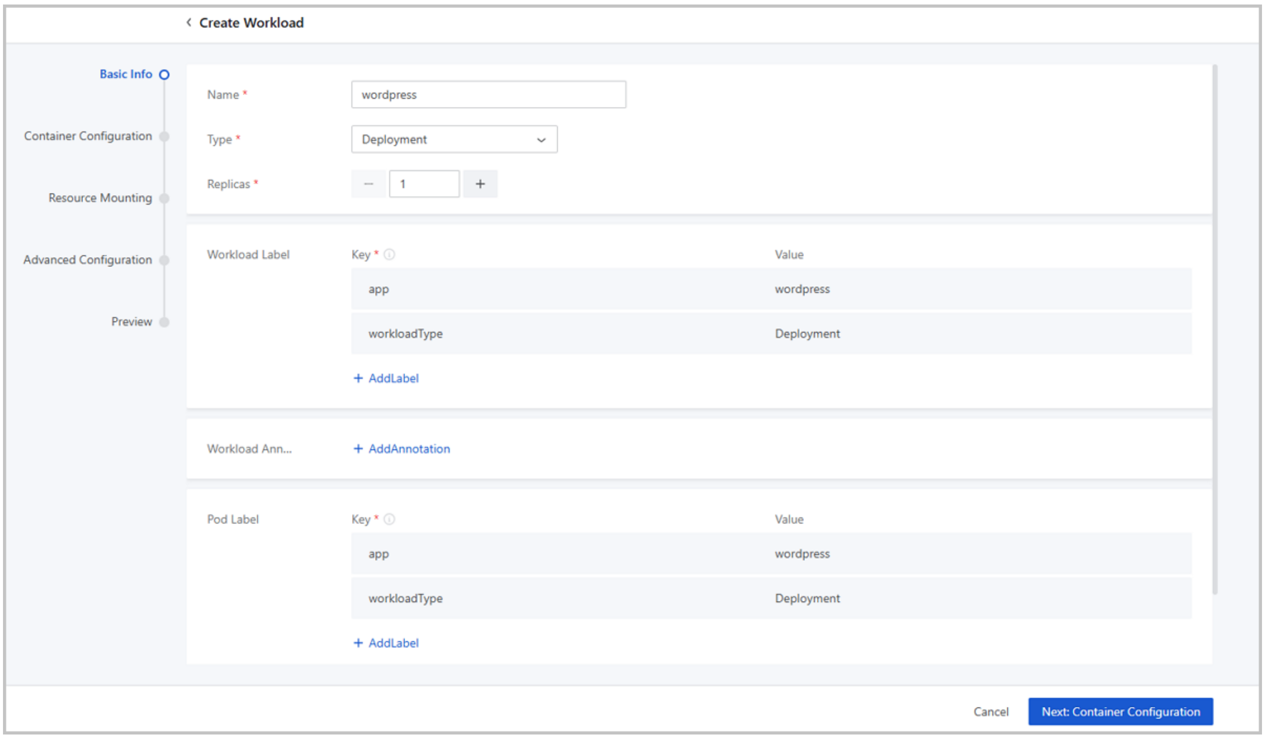
You can refer to the following example to enter the appropriate contents:

- **Name**: this scenario is set to wordpress
- **Type**: Select Deployment

- **Replicas**: set to 1
- Default values can be used for other settings.

As shown in Figure 92: Creating a Wordpress Workload - Basic Information:

Figure 92: Creating a Wordpress Workload - Basic Information



Name * wordpress

Type * Deployment

Replicas * 1

Workload Label

Key *	Value
app	wordpress
workloadType	Deployment

Workload Ann...

Pod Label

Key *	Value
app	wordpress
workloadType	Deployment

2. Container Configuration Click Add Container and refer to the following example to enter the appropriate content:

    - **Container name**: set the container name
    - **Container type**: Selection of the worker container
    - **Image Resource**: select Repository
    - **Image**: Choose a WordPress Image
    - **Tag**: select php8.1
    - **Resource Request**: may be left blank
    - **Resource Limit**: may be left blank
    - **Command**: can be left blank
    - **Argument**: can be left blank
    - **Env**: Select **custom key-value pairs** to add the following four environment variables:

# key: WORDPRESS_DB_Host; value: database access url, here fill in mysql-wordpress:3306

# Key: WORDPRESS_DB_USER; Value: the username for accessing the database, fill in the same value here as in the MySQL workload environment variable MYSQL_USER

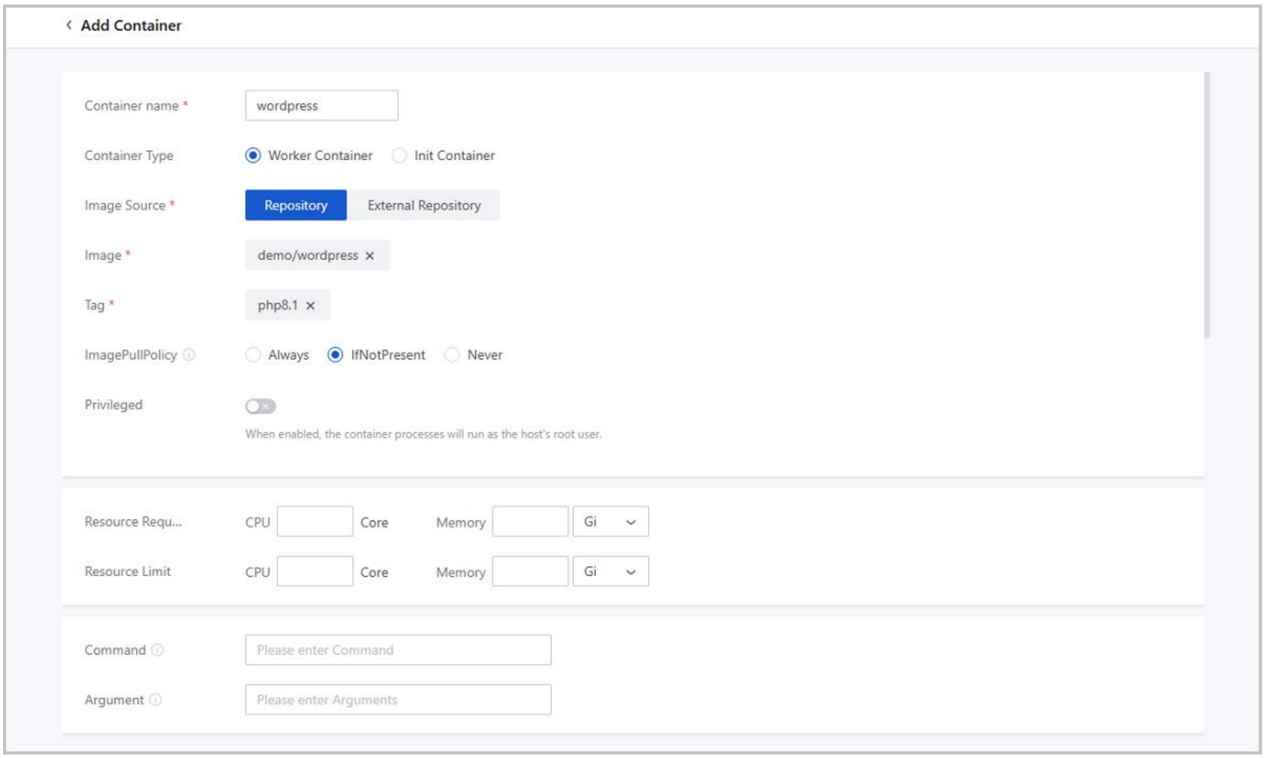
# Key: WORDPRESS_DB_PASSWORD; Value: the username to access the database, fill in the value here that matches the value in the MySQL workload environment variable MYSQL_PASSWORD

# Key: WORDPRESS_DB_NAME; Value: database name, here fill in the value consistent with that in the MySQL workload environment variable MYSQL_DATABASE

- **LivenessProbe**: not enabled in this scenario
- **ReadinessProbe**: not enabled in this scenario
- **StartupProbe**: not enabled for this scenario
- **PostStart**: not enabled for this scenario
- **PreStop**: not enabled in this scenario

As shown in Figure 92: Creating a Wordpress Workload-Basic Information:

Figure 93: Creating a Wordpress workload - container configuration



‹ Add Container

Container name * wordpress

Container Type ● Worker Container ○ Init Container

Image Source * Repository | External Repository

Image * demo/wordpress ×

Tag * php8.1 ×

ImagePullPolicy ○ Always ● IfNotPresent ○ Never

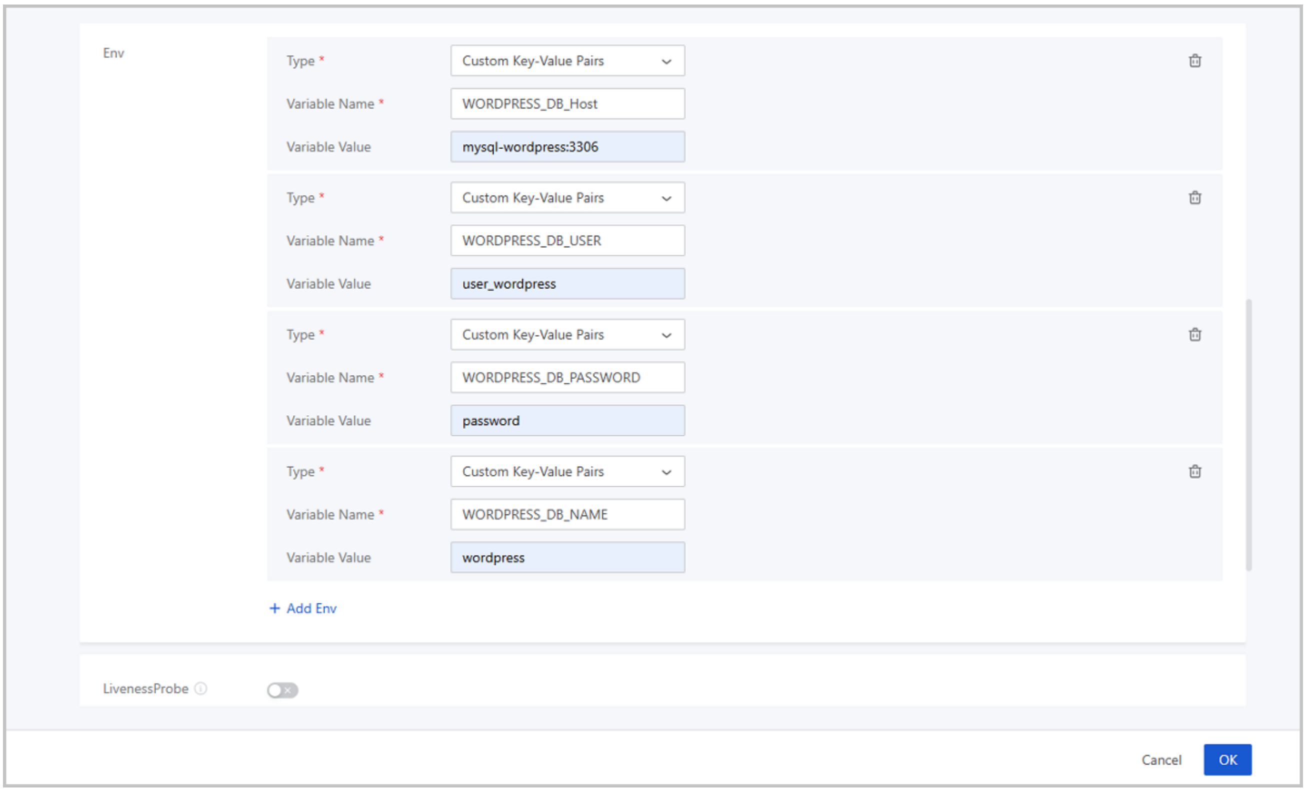
Privileged
When enabled, the container processes will run as the host's root user.

Resource Requ... CPU Core Memory Gi

Resource Limit CPU Core Memory Gi

Command Please enter Command

Argument Please enter Arguments



Env

Type * Custom Key-Value Pairs
Variable Name * WORDPRESS_DB_Host
Variable Value mysql-wordpress:3306

Type * Custom Key-Value Pairs
Variable Name * WORDPRESS_DB_USER
Variable Value user_wordpress

Type * Custom Key-Value Pairs
Variable Name * WORDPRESS_DB_PASSWORD
Variable Value password

Type * Custom Key-Value Pairs
Variable Name * WORDPRESS_DB_NAME
Variable Value wordpress

3. Resource mounting: In this scenario, there is no need to mount resources here.
4. Advanced Configuration: In this scenario, the default configuration is maintained
5. Confirmation Message: View the workload to be created and confirm the creation.

b) Creating services

In the nKU main menu, click **Container Orchestration** > **Network Management** > **Services** to enter the **Service** interface. Click **Create Service** to bring up the **Create Service** screen.

You can refer to the following example to enter the appropriate contents:

- **Name**: this scene is set to wordpress
- **Associated Workload**: select the workload created in step b
- **Type**: Select NodePort
- **IP Type:** Select IPv4
- **Service Port**: Set the following information:

  # **Port**: set to 80 for this scenario

  # **Container port**: set to 80 for this scenario

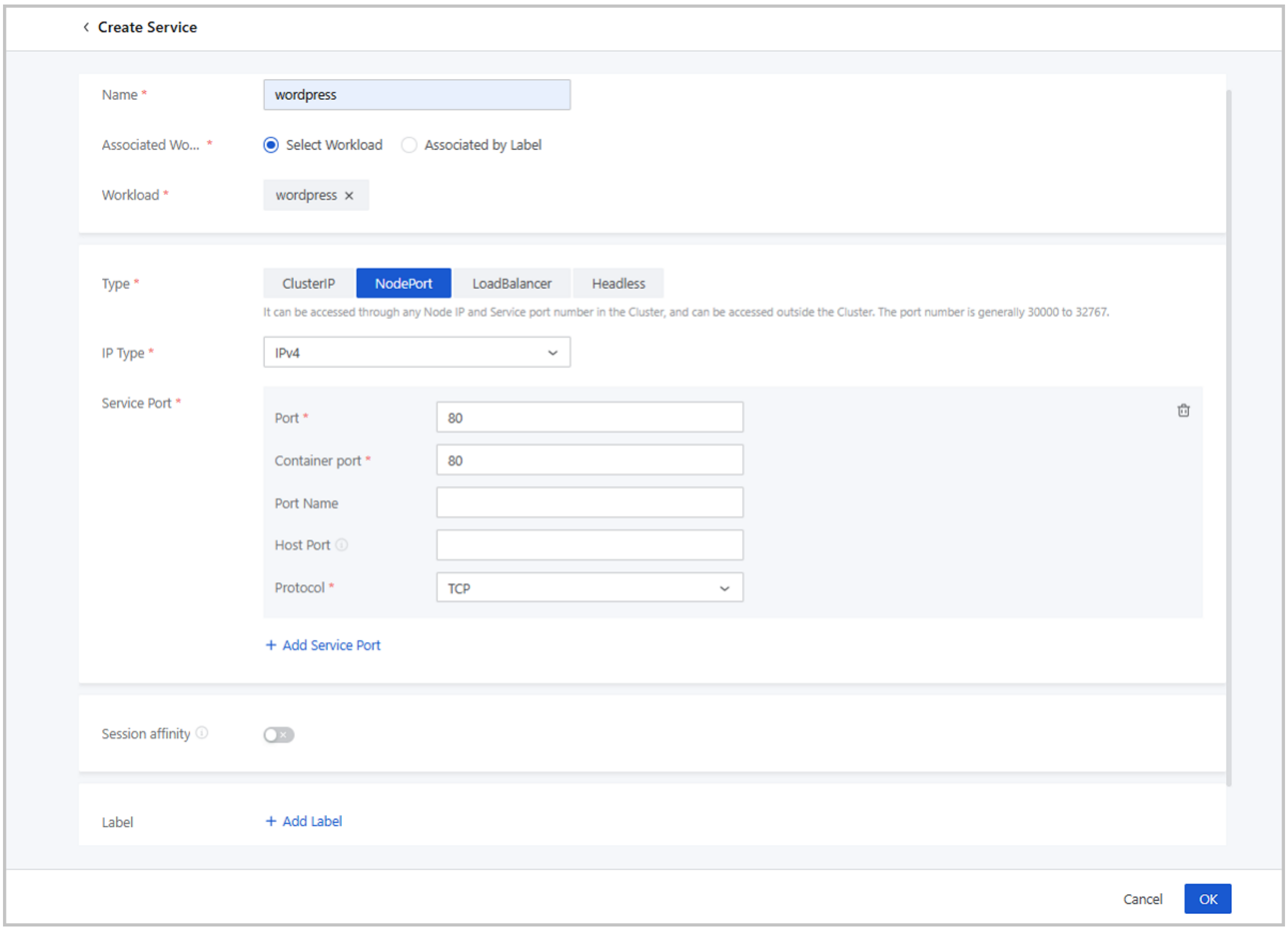
  # **Host port**: can be left blank

# **Protocol**: This scenario is set to TCP

- **Session affinity**: Not enabled for this scenario

As shown in Figure 94: Creating a Wordpress service:

Figure 94: Creating a Wordpress service



‹ Create Service

Name * wordpress

Associated Wo... * ● Select Workload ○ Associated by Label

Workload * wordpress ×

Type * ClusterIP | NodePort | LoadBalancer | Headless

It can be accessed through any Node IP and Service port number in the Cluster, and can be accessed outside the Cluster. The port number is generally 30000 to 32767.

IP Type * IPv4

Service Port *

Port * 80

Container port * 80

Port Name

Host Port

Protocol * TCP

Session affinity

Label

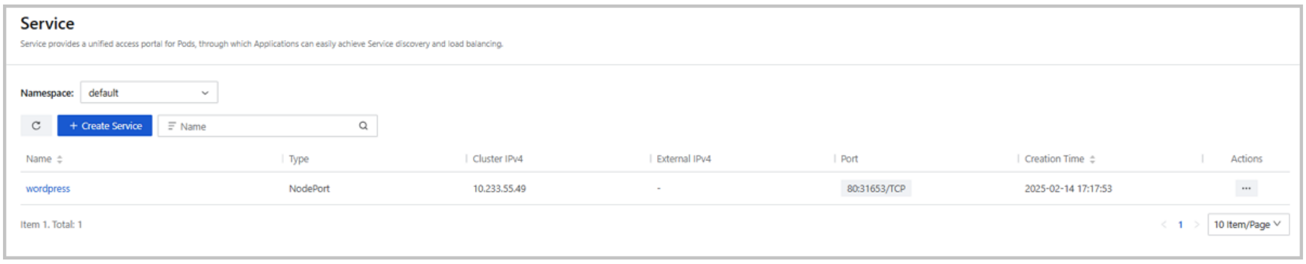
follow-up operation

Go to the list of **services** and check the port of the WordPress service, type ${NoteIP}:${Port} in your browser to access the WordPress service.

Note: ${NoteIP} is the IP address of any node in the cluster and ${Port} is the WordPress service port. In the image below, the WordPress service port is 31710.

As shown in Figure 95: Service Port

Figure 95: Service Ports



Service

Service provides a unified access portal for Pods, through which Applications can easily achieve Service discovery and load balancing.

Namespace: default

Name	Type	Cluster IPv4	External IPv4	Port	Creation Time	Actions
wordpress	NodePort	10.233.55.49	-	80:31653/TCP	2025-02-14 17:17:53	···

Item 1. Total: 1

12.2 One-click deployment of applications

background information

This section describes the method of deploying applications in nKU Express. This scenario takes the deployment of a MySQL application as an example, which consists of the following two main steps:

1. Uploading the MySQL Chart
2. **Steps to** install application in one click

1. Uploading the MySQL Chart

a) Preparing the MySQL Charts

b) Creating a Repository

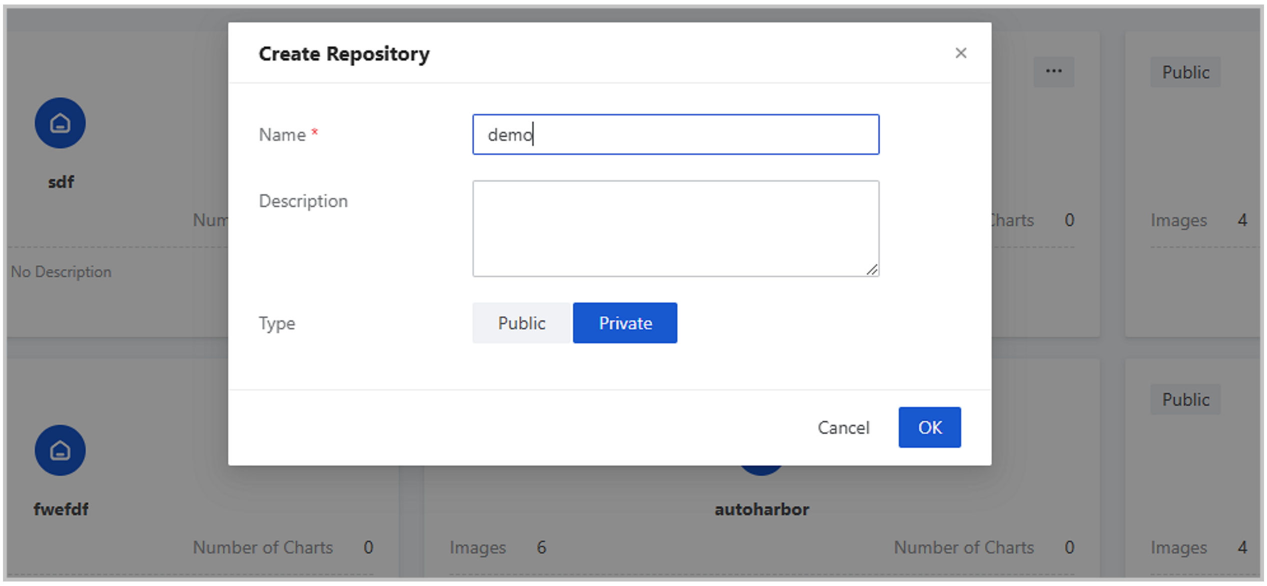
In the nKU main menu, click **Artifact Repository** > **Repository**. In the **Repository** screen, click **Create Repository** to bring up the **Create Repository** screen.

You can refer to the following example to enter the appropriate contents:

- **Name**: Set repository name
- **Description**: optional, note repository-related information
- **Type**: This scenario is set to private

As shown in Figure 96: Creating a Repository:

Figure 96: Creating a Repository

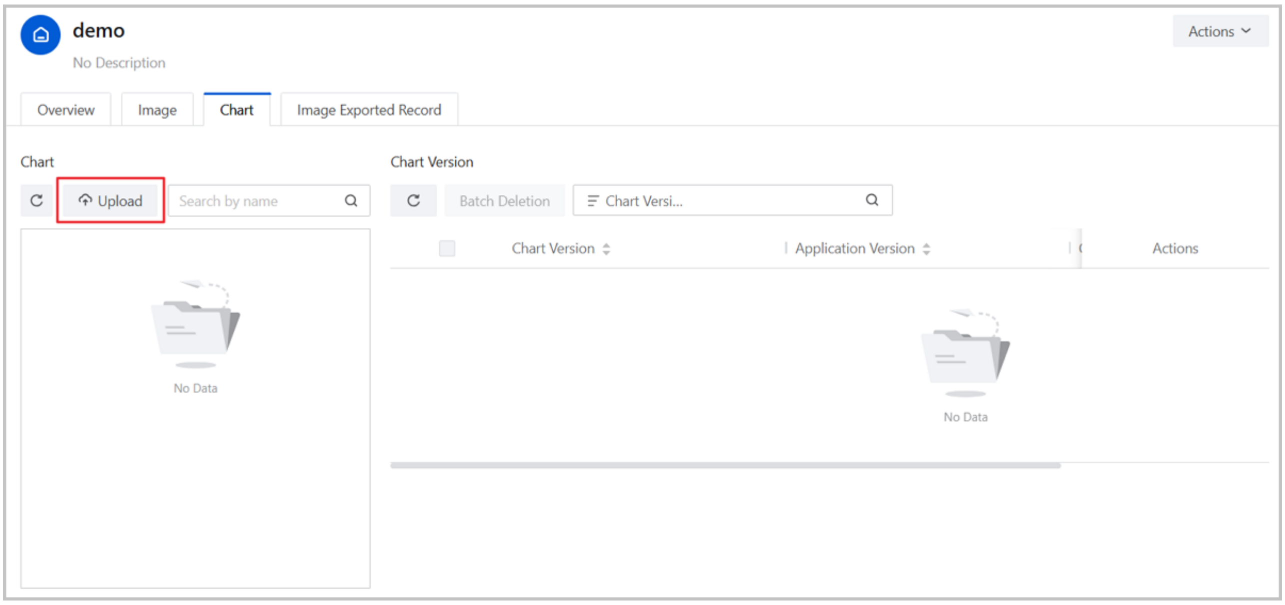


c) Uploading the MySQL Chart

Go to the **Chart** tab of the Repository details page and click **Upload** to upload the image repository zip prepared in step a to that repository.

As shown in Figure 97: Uploading the Chart:

Figure 97: Uploading a Chart



2. One-click to install application

In the Chart list, find the uploaded MySQL Chart, click **Action** > **to install**, in the pop-up **publish application** interface, fill in the configuration parameters as required and click **OK**.

follow-up operation

After the application is installed, users can view and manage the application in the **application** interface by going to the nKU main menu and clicking **Container Orchestration** > **Application Management** > **Application**.

Note: In addition to the chart paths described in this section, nKU also provides a variety of official Chart deployment packages in the application market. Users can enter **the artifact repository** > **application market** and install these deployment packages with one click. In this case, there is no need to create a repository and upload your own Charts, which is faster and more convenient.

Glossary

application

An application is an instance installed by the Helm tool, a collection of resource objects such as workloads, services, storage, and so on.

workload

A workload is an application running on Kubernetes that manages a set of Pod using the same image, including Deployment, StatefulSet, and DaemonSet types.

job

The job is responsible for batch processing of ephemeral operations and ensures that one or more containers used for batch processing of operations comprise the termination of the functionality, including one-time tasks and timed tasks of two types.

pod

A pod is the smallest unit of cluster scheduling and management cluster. Containers in the same pod share the same network and storage.

service

Service provides a unified access entry point for container services, allowing applications to easily implement service discovery and load balancing.

ingress

Ingress is a set of HTTP or HTTPS routing rules used for external access to services (Servcie) within a cluster, providing externally accessible URLs, load balancing, SSL termination, HTTP ingress for cluster services.

network policy

NetworkPolicy is a policy that controls the behavior of workload-to-workload and workload-to-external network traffic to protect applications from network attacks by filtering inbound and outbound traffic.

pvc

A PVC (PersistentVolumeClaim) is used to persistently store the data of a workload so that it is retained when the pod is restarted or deleted.

configmap

Configmap stores configuration data in the form of key/value pairs, which can be mounted configMap and used as a configuration file in a workload. A configmap decouples the user's environment configuration information from the container configuration image, making it easy to modify the application configuration.

secret

Secret stores sensitive configurations in the Kubernetes cluster, such as passwords, certificates, and so on, and can be used as files or environment variables in workloads.

microservices application

A microservices application is a set of workloads, services, service mesh resources, and other resource objects that support the access of microservices applications to the platform for visual governance.

service topology

Service topology visualizes invocations, dependencies and traffic monitoring data between microservices applications in the current namespace.

repository

The Repository is used to store Docker images and Charts uploaded by users. Users can upload their modified images and Charts to the Repository, or download the uploaded Docker images and Charts from the Repository.

YAML template repo

The YAML template repository is used to store YAML templates written by users and provides example YAML template repositories for Deployment, Statefulset, Service, Configmap, PersistentVolumeClaim, and more than a dozen other common Kubernetes resources. Users can combine the example YAML templates in the template repository to quickly write the required YAML text.

application market

The application market provides rich official Chart deployment applications such as kafka, zookeeper, mysql, redis, rabbitmq, etc., which can be installed to the cluster and deployed as application instances with a single click.

one-click inspection

Perform a comprehensive one-click health check on the platform's critical resources and services. It scores the health of the inspected resources and services based on the check results, and provides inspection suggestions and reports to support efficient operation and maintenance, ensuring the platform's resources and services are in optimal condition.

alarm

Alarms are used to monitor and respond to changes in the state of temporal data and push alarm messages to specified Endpoint through the notification service, and users can create alarms to monitor resources.

Endpoint

Endpoint refers to the way users get information about subscription topics, and the types of Endpoint include: system, email, enterprise WeChat, and nail.

cluster

Clusters mainly refer to Kubernetes clusters running container applications. nKU provides standardized, highly available and stable Kunernetes clusters and simplifies operations and maintenance operations such as deployment, expansion, upgrading, and high availability of clusters, so that users can focus on their own upper-tier container applications.

node auto-scaling group

A Node Auto Scaling Group is a node management unit that can automatically adjust the number of nodes based on the business load of a cluster, enabling on-demand resource allocation and cost optimization.

node

Nodes are the basic building blocks of a Kubernetes cluster and are categorized into management nodes (Master) and compute nodes (Node).

GPU pool

GPU pool is a resource sharing unit formed by collecting multiple GPUs to achieve dynamic management and allocation of GPUs, improve resource utilization and optimize the flexibility of resource allocation.

storage class

The StorageClass is a configuration template used in Kubernetes to dynamically create a PVC that defines the properties of the volume, the creation policy, and the required storage plugins.

external network

An External Network provides IP addresses for pod and Loadbalancer Services in the cluster and allows access from outside the Kubernetes cluster.

project

Project is a kind of tenant, which is used to realize resource isolation and user rights management, after the user joins the project, he/she can have the rights to view or operate the resources under the project.

User

Users are created by the administrator or synchronized from a 3rd-party authentication system and managed by the administrator. Users need to join the project or be set as an administrator to have access to operate the platform.

namespace

Namespace provides virtual isolation for Kubernetes clusters, where resources in different namespaces are isolated from each other.

AccessKey

The AccessKey is an identity credential for accessing the platform APIs with platform privileges consistent with the current user, including: AccessKey ID and AccessKey Secret.