



Operation Manual for KubeManager — Sangfor PaaS Platform

(For Version 1.0)



SANGFOR

Operation Manual of Sangfor PaaS – KubeManager 1.0 Confidentiality: internal use

Sangfor Technologies Inc.

Revision



Revision Number	Author	Date	Brief Introduction
1.0	Zhao Zhenyang	June 30, 2020	Initial draft

Contents

1	Overview of KubeManager.....	1
1.1	Development of Docker.....	1
1.2	Kubernetes.....	1
1.3	Sangfor PaaS Platform — KubeManager.....	3
1.3.1	KubeManager Architecture.....	4
2	Environment.....	5
2.1	Environment Deployment.....	5
2.1.1	Environment Requirements.....	5
2.1.2	Installation of Manage Cluster.....	6
2.1.3	Installation of User Cluster.....	7
2.2	Environment Maintenance.....	10
2.2.1	Expansion and Addition of User Cluster.....	10
2.2.2	Maintenance and Deletion of Nodes.....	11
3	User System.....	13
3.1	Overview of User System.....	13
3.2	Local User.....	13
3.3	Integration with Third-party Account.....	14
4	Registry.....	16
4.1	Deployment and High Availability.....	16
4.2	Private Registry.....	16
4.3	Registry Configuration.....	16
4.4	Image Upload and Management.....	17
5	App Store.....	18
5.1	Deployment and High Availability.....	18
5.2	Configuration of App Store.....	18
5.3	Application Upload and Management.....	20
5.4	Multi-cluster Applications.....	20
6	Storage and Use.....	22



6.1	Storage Server.....	22
6.2	Creation of Storage Class.....	23
6.3	Creation of PV.....	25
6.4	Support to Other Storage.....	26
7	Multi-cluster Management.....	27
7.1	K8S Clusters on "Hosts from Cloud Service Providers".....	28
7.2	Cluster of "Kubernetes Hosting Service Providers".....	30
8	Project Configuration.....	31
8.1	Creation of Project.....	31
8.2	Namespace Management.....	32
9	System Tools.....	33
9.1	Global Settings.....	34
9.2	Cluster Settings.....	36
9.3	Project Settings.....	39
10	Other Configurations.....	39
10.1	Backup and Recovery.....	39
10.2	Security.....	41
10.3	Precautions.....	42



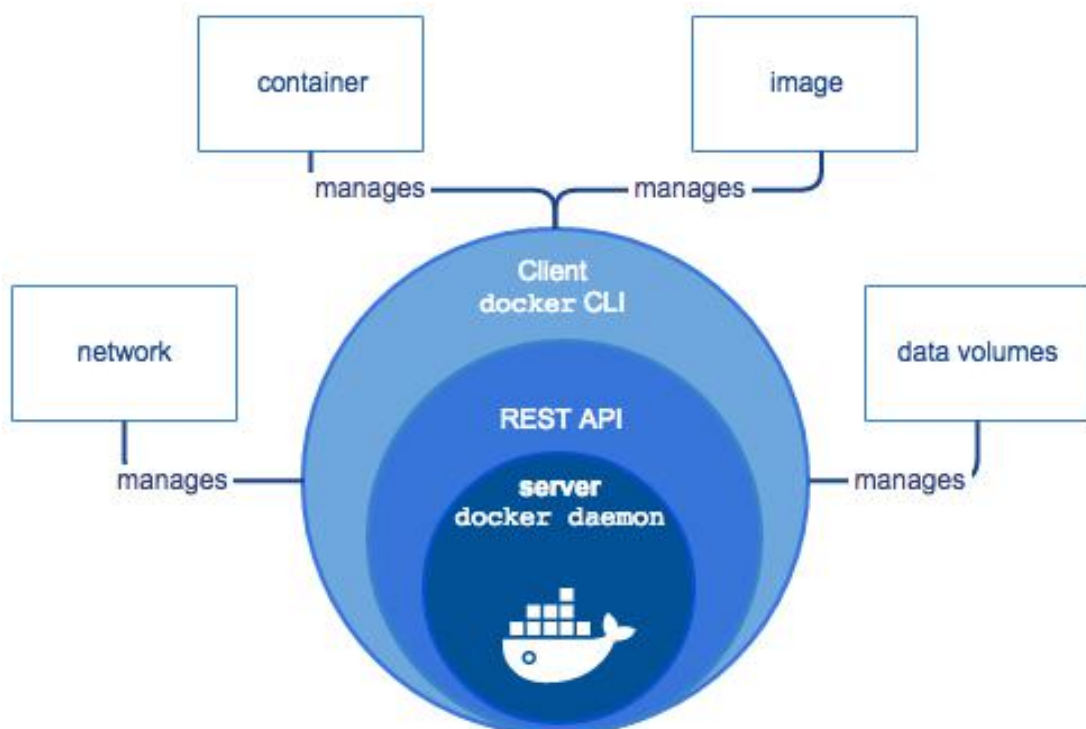
1 Overview of KubeManager

1.1 Development of Docker

The container technology originated from the namespace resource isolation technology and cgroup resource restriction technology of Linux kernel. Namespace saved path to Linux kernel in 2001. The container technology really became available after the release of LXC in 2008.

Linux Container is a kernel virtualization technology, which can provide lightweight virtualization and isolate processes and resources. Linux Container is abbreviated as LXC. LXC needs no instruction interpretation and full virtualization. LXC, the predecessor of docker, has been replaced by Libcontainer since version 0.9. Docker started with LXC, then optimized LXC, encapsulated components such as Libcontainerd and libnetwork, and managed them with runc.

The current Docker has three packages: containerd, docker-ce and docker-ce-cli. Docker-ce-cli replaces runc as the manager.

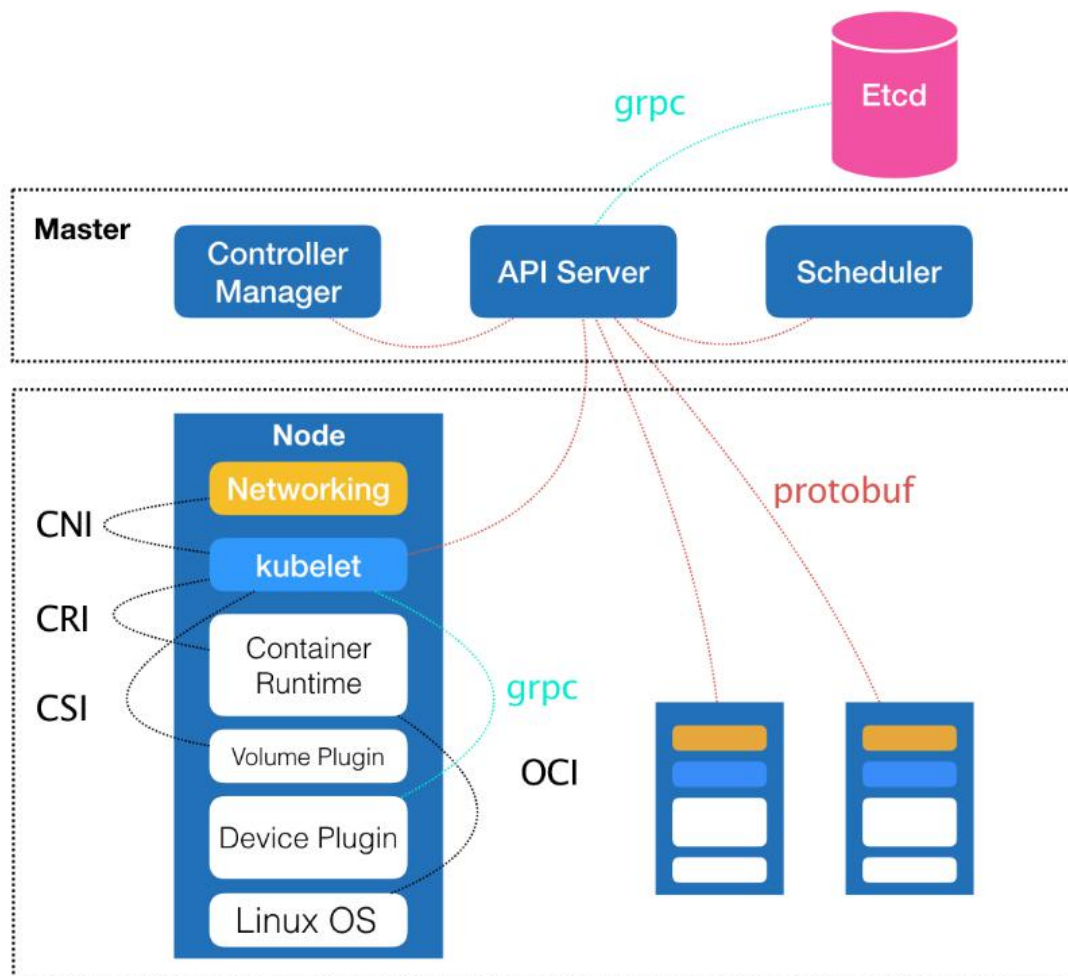


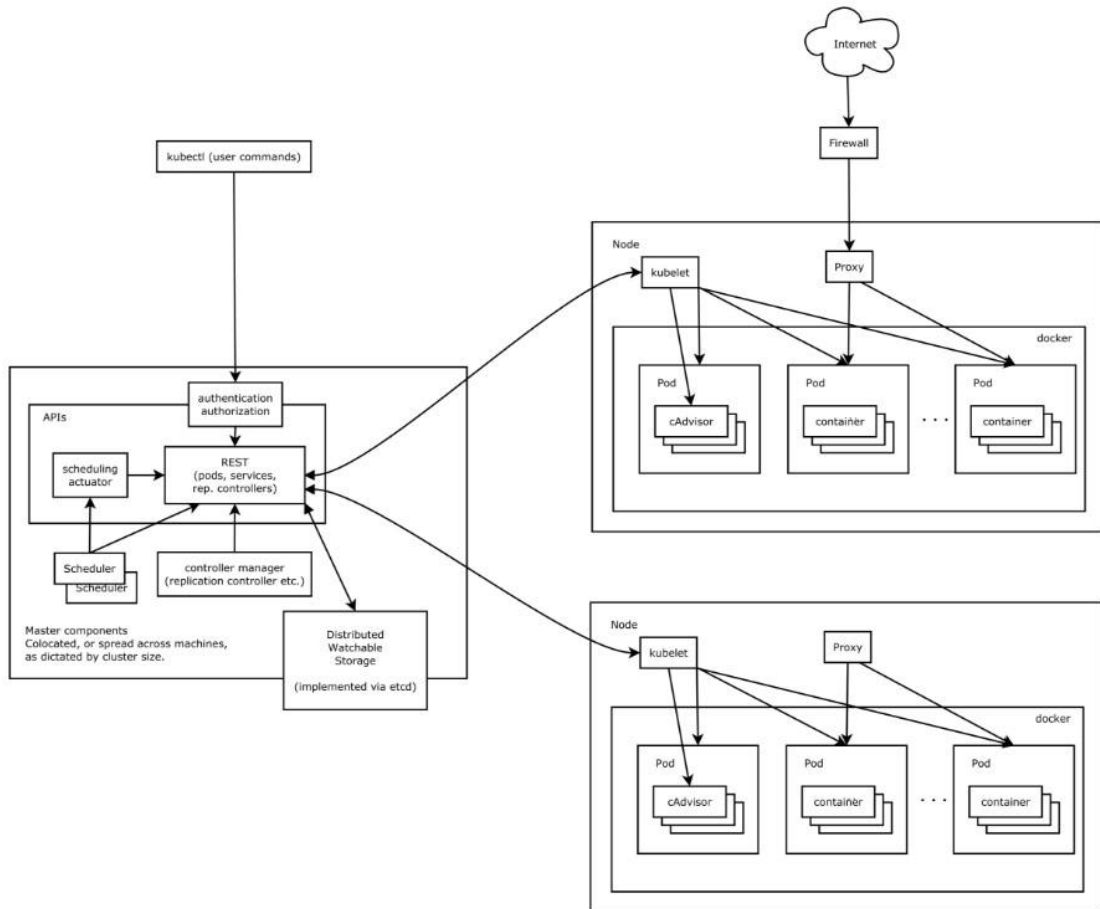
1.2 Kubernetes

Docker solves various problems caused by inconsistent environment in the process of

application deployment, but problems like container deployment, expansion and management on large clusters give birth to container orchestration engines, such as Kubernetes, Docker swarm and Mesos. Mesos was released by Twitter, swarm by Docker, and Kubernetes originated from the Borg system of Google. Kubernetes is abbreviated as K8S, in which 8 represents the 8 letters being omitted. K8S is portable, extensible and automatic.

K8S manages the above services with declarative API. K8S contains such components as ETCD, api server, controller manager, scheduler, kubectl, kubelet, kubeproxy, and docker. Generally, etcd, api server, controller manager, and scheduler are deployed in one node, which is usually called Master node. Other components are deployed in each node and become nodes. Master mainly stores the cluster configuration, schedules services in the cluster, and controls the cluster. Node is mainly responsible for the operation of containers, the hosting of services and the release of services.



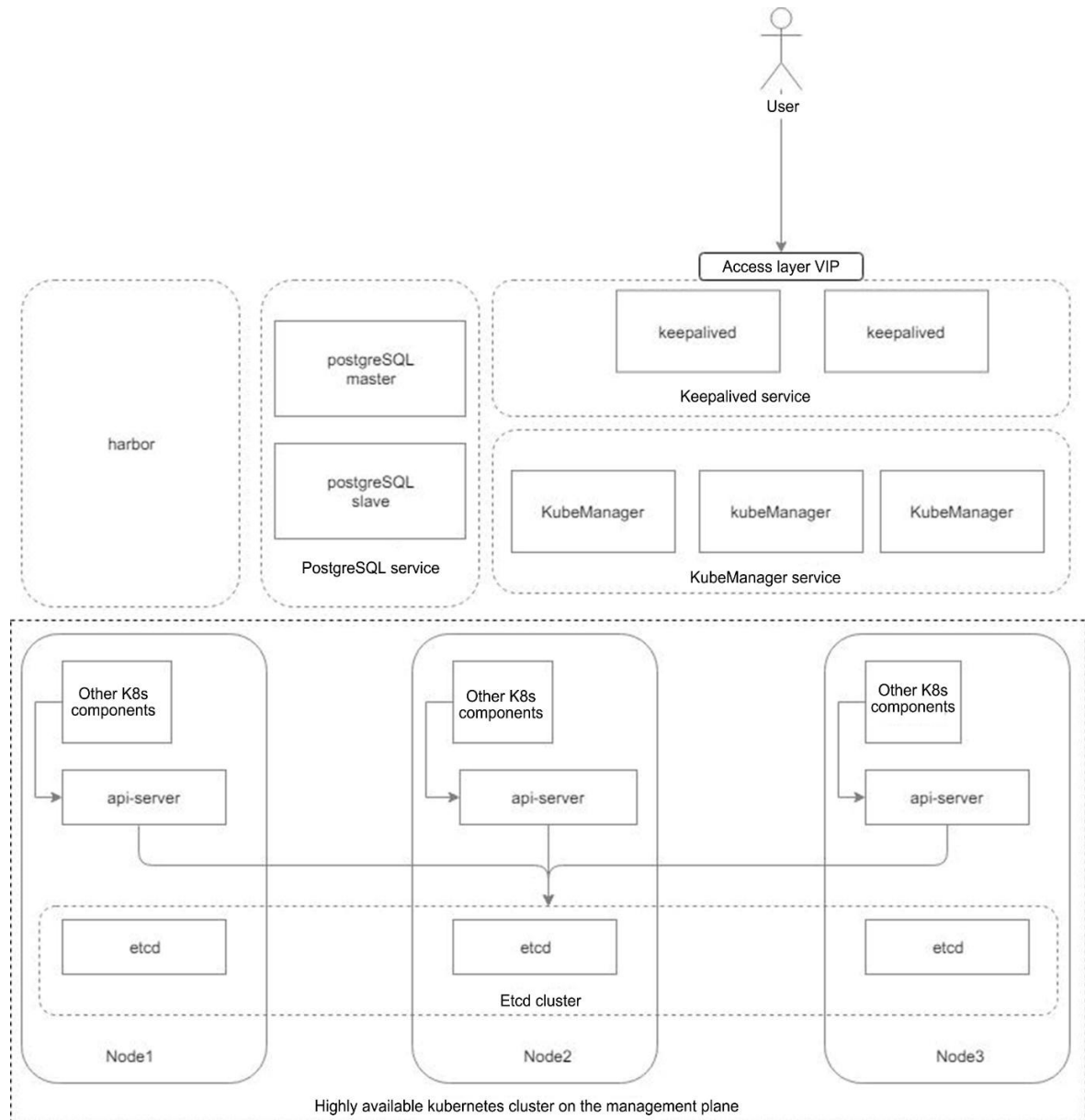


1.3 Sangfor PaaS Platform — KubeManager

KubeManager is a simple, easy-to-use and stable cloud native multi-K8S management platform released by Sangfor. By applying the cloud native technology, KubeManger runs all system components on K8S. KubeManager provides default registry and app store, supports integration with the third-party registry and app store, and supports third-party account system. KubeManager is a lightweight and stable commercial Kubernetes multi-cluster management system.



1.3.1 KubeManager Architecture



KubeManager is a PaaS system running on kubernetes. KubeManager provides multi-K8S management, registry, app store and other functions. Every component of the KubeManager system meets the high availability mechanism. On the underlayer, the high availability of kubernetes is transplanted to this system, to ensure the high availability of system, so that the system can provide external services normally when the duplicate of any node or any service fails.

The following subsystems ensure the high availability of the entire system:

- The keepalived service ensures the high availability of interfaces through floating IP addresses;



- The daemonset service ensures the access performance expansion and high availability of services;
- The postgresQL service component ensures the high availability of databases. The operator component allows automatic configuration of various database modes (single host, master-slave, and one-master-to-multiple-slave) and can be used to store audit logs or harbor user information.
- The ETCD cluster ensures the high availability of configuration data. All configuration information of KubeManager is stored in K8S at the underlayer.
- KubeManger runs on kubernetes as a service. Kubernetes guarantees the high availability of all components of KubeManager;

2 Environment

2.1 Environment Deployment

2.1.1 Environment Requirements

Non-highly available environment					
Host	Hostname	CPU	Memory	Disk	Installation component
Host 1	KubeMager	4C	8GB	80 GB system disk without partition; mount 50 GB	Manage clusters, registries, and app stores
Host 2	etcd,controller, worker	8C	8GB	100 GB system disk without partition; mount 150 GB	User cluster
Host 3	etcd,controller, worker	8C	8GB	100 GB system disk without partition; mount 150 GB	User cluster
Host 4	etcd,controller, worker	8C	8GB	100 GB system disk without partition; mount 150 GB	User cluster



Highly available environment					
Host	Hostname	CPU	Memory	Disk	Installation component
Host 1	KubeMager	4C	8GB	80 GB system disk without partition; mount 50 GB	Manage clusters, registries, and app stores
Host 2	KubeMager	4C	8GB	80 GB system disk without partition; mount 50 GB	Manage clusters, registries, and app stores
Host 3	KubeMager	4C	8GB	80 GB system disk without partition; mount 50 GB	Manage clusters, registries, and app stores
Host 4	etcd,controller, worker	8C	8GB	100 GB system disk without partition; mount 150 GB	User cluster
Host 5	etcd,controller, worker	8C	8GB	100 GB system disk without partition; mount 150 GB	User cluster
Host 6	etcd,controller, worker	8C	8GB	100 GB system disk without partition; mount 150 GB	User cluster
Host 7	worker	8C	8GB	100 GB system disk without partition; mount 150 GB	User cluster
Host 8	worker	8C	8GB	100 GB system disk without partition; mount 150 GB	User cluster

KubeManager can be installed on VM, bare metal, cloud server and other infrastructures, and supports various network topologies. For example, the deployment in VPC, single NIC deployment in classic network, multi-NIC deployment in classic network.

2.1.2 Installation of Manage Cluster

The installation of KubeManager cluster is very simple and convenient, including the following steps:

- Install the operating system



- Set disk partition and time
- Assign IP addresses and plan networks
- Configure ssh server
- Execute the installation program

The installation script has only one command. Then KubeManager can be installed based on the configured parameters. Once installation is completed, you can use the excellent service provided by KubeManager after simple initial setup.

You need to set the default password of admin when logging in for the first time:

The image shows a 'Reset Password' web form. At the top, it says 'Reset Password' and 'Please input your account information.' Below this, there is a 'Current Password' input field. Then, for 'New Password', there are two radio buttons: 'Create my own password' (which is selected) and 'Use a randomly generated password'. Below the radio buttons is a large text area for the 'New Password' with a password strength hint: 'Password should contain 8 - 64 characters. The characters should consist of any 4 of the following: uppercase letters, lowercase letters, digits and the special characters: ~!@#%&*<>~^\$.*+?~!{}[]\^_~. A maximum of 4 consecutive characters are allowed.' Below this is a 'Confirm Password' input field. At the bottom of the form is a 'Submit' button.

Select "Custom password" to define your own password or select "Random password" to generate a random password. After the password is set, it jumps to the homepage of KubeManager.

2.1.3 Installation of User Cluster

You can directly install the K8S cluster on the KubeManager interface, which is very convenient and quick.

First, login to the KubeManager platform, and then shift to the **Global** mode:




Click **Add cluster**, and select **Custom cluster**. If other clusters are required, select the corresponding options as prompted:




Sangfor KubeManager Global Clusters Apps Users Settings Security Tools English Default Admin (admin)


Add Cluster - Select Cluster Type





From existing nodes (Custom)
Create a new Kubernetes cluster using SKM, out of existing bare-metal servers or virtual machines.


With SKM and new nodes in an infrastructure provider

 Amazon EC2


 Azure


 DigitalOcean


 Linode

 vSphere

With a hosted Kubernetes provider

 Amazon EKS

 Azure AKS

 Google GKE

Cancel

Add node for the cluster:

Sangfor KubeManager Global Clusters Apps Users Settings Security Tools English Default Admin (admin)

Add Cluster - Custom

Cluster Name *
e.g. sandbox [Add a Description](#)

Note: Only nodes running on centos centos 7.7/908 are supported for now

Hostname Prefix	IP	etcd	Control Plane	Worker	Labels	Taints	Connection Status
<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Labels	Taints	Unchecked Quick Check

Number of nodes required: 1, 3, or 5 1 or more 1 or more

+ Add Node New Nodes Allowed 2

Next: Configure Cluster Cancel

Configure the node:



Edit

Target IP Address *

External IP Address *

Enter an external IP for application delivery

Internal IP Address *

SSH Port *

Username *

Password *

SaveCancel

Check the connectivity of node after configuration.

Test the connectivity of the node and set the role of node:

Sangfor KubeManager

Global

Clusters

Apps

Users

Settings

Security

Tools

English

Default Admin (admin)

Add Cluster - Custom

Cluster Name *

e.g. sandbox

Add a Description

Note: Only nodes running on centos 7.7/908 are supported for now.

Hostname Prefix	IP	etcd	Control Plane	Worker	Labels	Taints	Connection Status
					Labels	Taints	Unchecked Quick Check

Number of nodes required: 1, 3, or 5 1 or more 1 or more

+ Add Node New Nodes Allowed: 2

Next: Configure ClusterCancel

Configure clusters, including network, ingress port, private registries;



Add Cluster - Custom

Note: Only nodes running on centos centos 7.7.1908 are supported for now.

Hostname Prefix	IP	etcd	Control Plane	Worker	Labels	Taints	Connection Status
go	10.113.83.113.22	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Labels	Taints	Normal

Member Roles
Control who has access to the cluster and what permission they have to change it.

Labels & Annotations
Configure labels and annotations for the cluster. None

Cluster Options [Edit as YAML](#)

☐ Use an existing SKM Template and revision Expand All

Kubernetes Options
Customize the kubernetes cluster options

Private Registry
Configure a default private registry for this cluster. When enabled, all images required for cluster provisioning and system add-ons startup will be pulled from this registry.

Advanced Options
Customize advanced cluster options

Authorized Endpoint
Enabling the authorized cluster endpoint allows direct communication with the cluster, bypassing the API proxy. Authorized endpoints can be retrieved by generating a kubeconfig for the cluster.

[Create](#) [Cancel](#)

2.2 Environment Maintenance

2.2.1 Expansion and Addition of User Cluster

When users use K8S cluster, with the use of resources, they need to expand or add nodes. Add nodes following these steps:

Click a cluster to jump to the Nodes list page, and click the **Add Node** button:

Sangfor KubeManager local | Cluster **Nodes** | Storage | Projects/Namespaces | Members | Tools English | Default Admin (admin)

Nodes [Add Node](#)

☐ Cordon ☐ Drain

State	Name	Roles	Version	CPU	RAM	Pods
Active	XX 192.168.0.1 sangfor.com/regist...		v1.10.13 18.0.3	18/4 Cores	3.2/4.9 GiB	29/110

As with cluster creation, add the intervention information and roles of nodes in configuration.

Sangfor KubeManager local | Cluster **Nodes** | Storage | Projects/Namespaces | Members | Tools English | Default Admin (admin)

Add Node

Note: Only nodes running on centos centos 7.7.1908 are supported for now.

Hostname Prefix	IP	etcd	Control Plane	Worker	Labels	Taints	Connection Status
<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Labels	Taints	Unchecked Quick Check
<input type="text"/>	<input type="text"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Labels	Taints	Unchecked Quick Check

Number of nodes required: (An etcd node, a control plane node and a worker node already exist in the cluster.) 1, 3, or 5 1 or more 1 or more

[Add Node](#) New Nodes Allowed: 2

[Save](#) [Cancel](#)



Edit

Target IP Address *

External IP Address *

Enter an external IP for application delivery

Internal IP Address *

SSH Port *

Username *

Password *

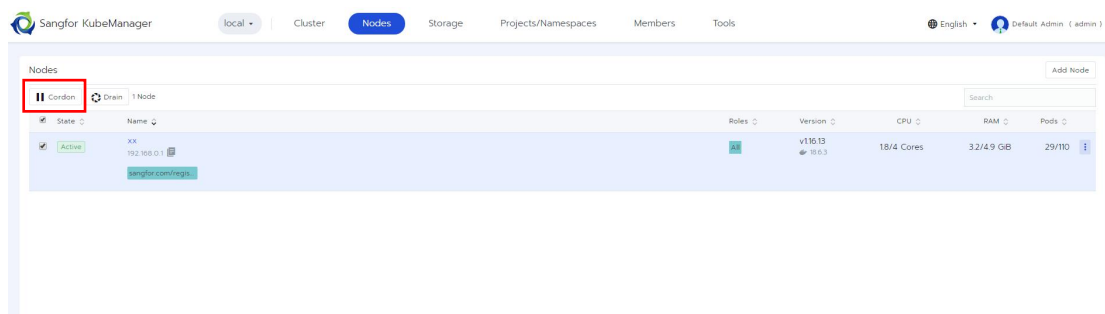
SaveCancel

Then, after checking the connectivity of nodes, install and configure nodes for the system and expand the cluster.

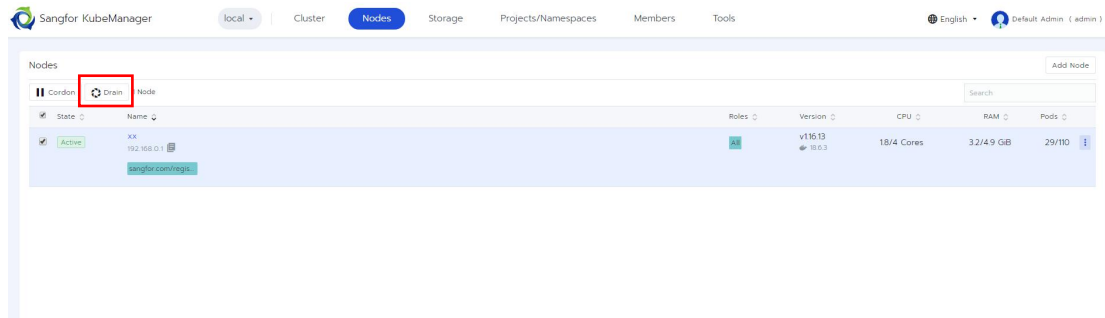
2.2.2 Maintenance and Deletion of Nodes

Maintain nodes when there is any fault. When a node failure cannot be fixed, delete this node and then add a new node to replace it.

First, **Cardon** the node. Then, the new service will not be scheduled to this node any more:



Then, **Drain** this node, to migrate the service on this node to other nodes seamlessly:



Drain "xx"

Mode

☒ **Safe**
If a node has standalone pods or ephemeral data it will be cordoned but not drained.

☐ **Aggressive**
Permanently delete:

- Standalone Pods and their data
- Pods with "Empty Dir" volumes and their data

Grace period for pods to terminate themselves

☒ Honor the default from each pod

☐ Ignore the defaults and give each pod:

30 seconds

Drain timeout

☐ Keep trying forever

☒ Give up after:

60 seconds

Drain **Cancel**

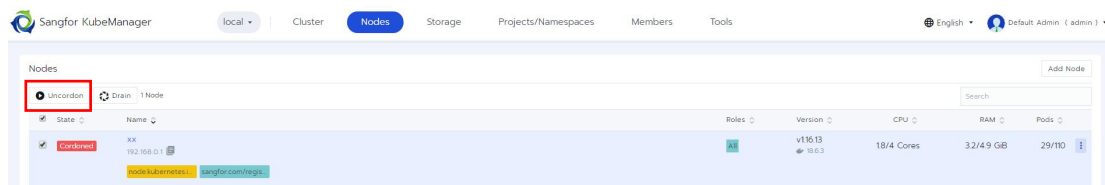
After the service on the node is evacuated, you can safely delete this node:

Are you sure you want to delete:

node1

Delete **Cancel**

Nodes that have been **pause** and **scatter** can be **Uncordon** after being fixed, and service scheduling is supported again:





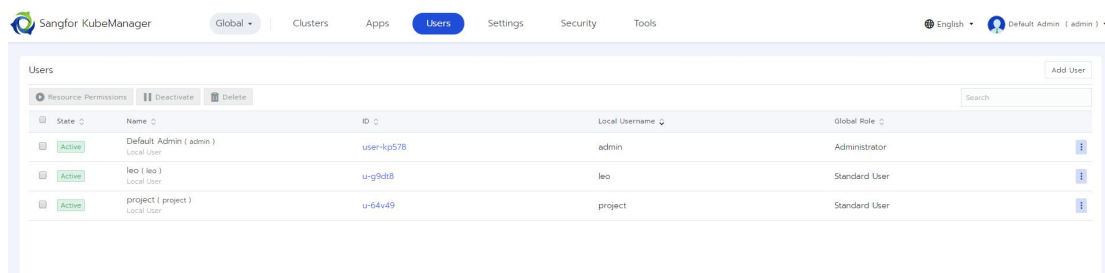
3 User System

3.1 Overview of User System

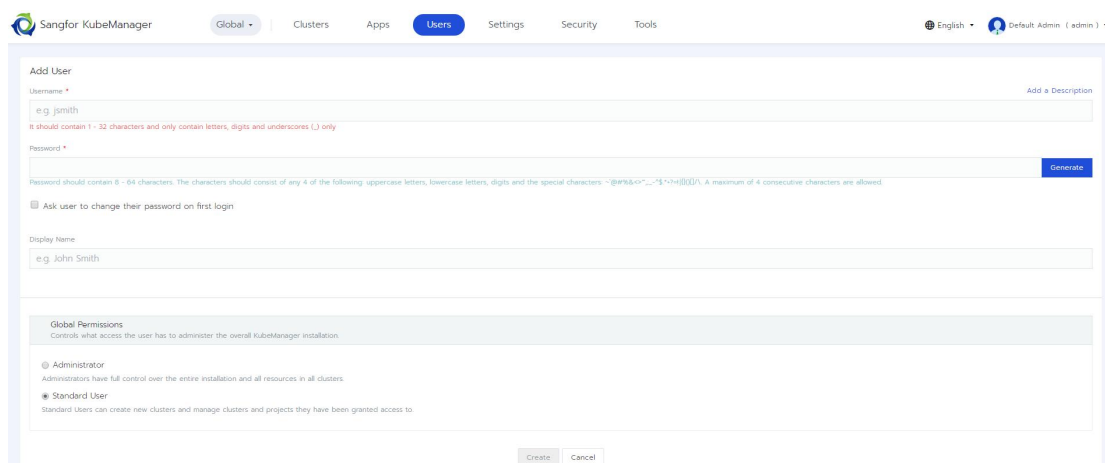
KubeManager supports a complete system of users and permissions. KubeManager supports the local user system, creation of local users, and integration with existing third-party permission systems, such as LADP and AD. Both local users and integrated third-party users have three-level permissions: **global**, **cluster** and **project**, and can customize permissions.

3.2 Local User

Click **Global** and view user list:

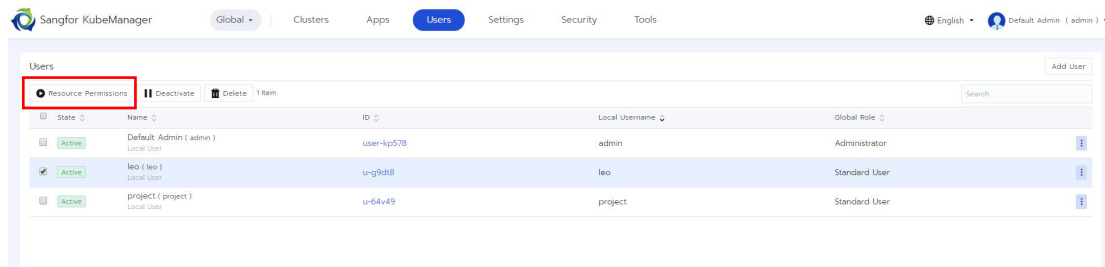


On the user list interface, add users:

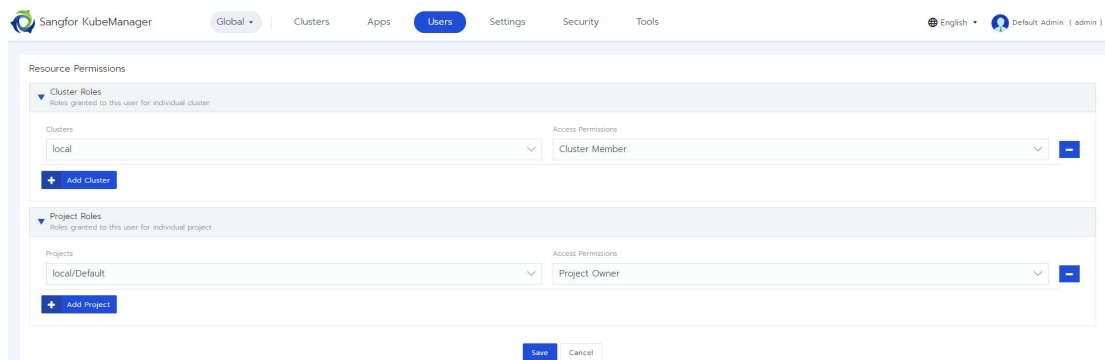


Create a local user, enter the username, set the password, or use the default password generated by the system, and set the user role to generate a local user.

Set accessible Resource Permissions for a user through resource authorization:



On the resource authorization page, select the cluster and project to be added, and choose the permissions for a user to access these resources:



After resource authorization, users can login to use and manage these resources and fulfill their service goals.

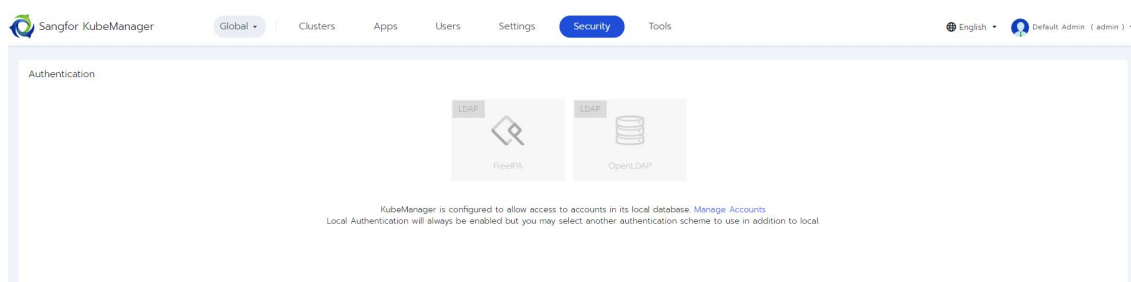
Similarly, for the user list at the **cluster** level, the global administrator can set an administrator and users for the cluster. The cluster administrator can set permissions for the cluster member list.

At the **Project** level, an administrator can add users and administrators for projects, and a project administrator can also set permissions for the project member list.

3.3 Integration with Third-party Account

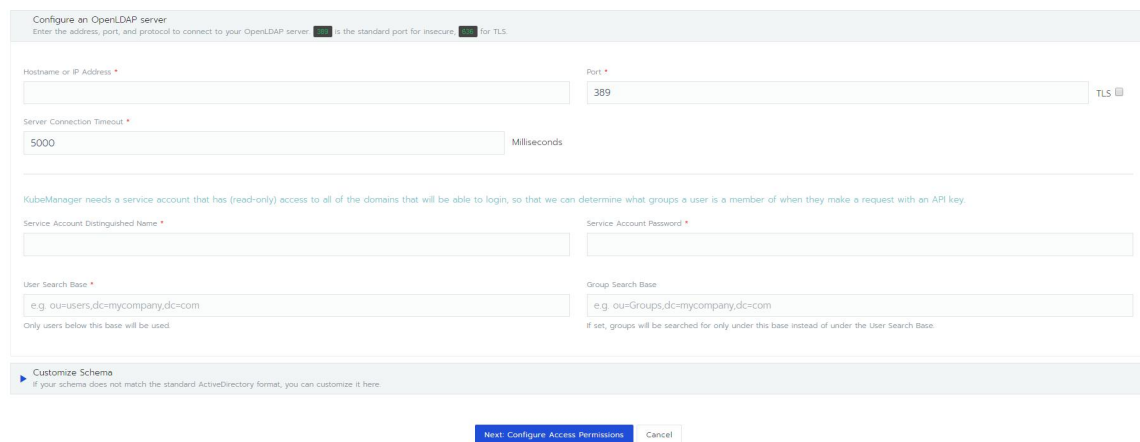
KubeManager supports the integration with third-party account systems such as LDAP, so that users can use their existing user systems directly without recreating users.

Choose **Authentication** from **Security** drop-down menu at the **Global** level, to integrate with the third-party user system:

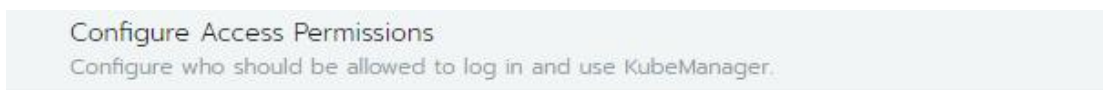


At present, accounts can be managed in a unified manner with LDAP. First, select **OpenLDAP**.

Configure the third-party account system:



Second, configure the access permission:



- ☒ Allow any valid Users
- ☐ Restrict access to only Authorized Users

There are two types of access permissions. One is to allow all users on LDAP to access KubeManager, namely **Allow all valid users**. Another one is to allow the authorized LDAP users to access it, namely **Allow authorized users only**. For the latter one, you need to authorize the users on LDAP before they login to KubeManager. Follow these steps:

- ☒ Restrict access to only Authorized Users

Authorized Users



 Add Users

After importing the third-party accounts into this system, you can use them as the ordinary local account.



4 Registry

4.1 Deployment and High Availability

The registry of KubeManager system is deployed together with KubeManager, and it also runs on K8S. The database of the registry also runs in pod, but the high availability is maintained by our operator.

4.2 Private Registry

Private registries refer to the registries of system images such as the installation package used when installing the cluster. For hybrid cloud and multi-cloud scenarios, there are multiple registries distributed in different areas. To install the cluster quickly, it is the best way to choose the nearest registry.

When creating or editing a cluster, you can configure the registry as follows:

▼ Private Registry
Configure a default private registry for this cluster. When enabled, all images required for cluster provisioning and system add-ons startup will be pulled from this registry.

Private Registry

☐ Disabled

☒ Enabled

URL

User

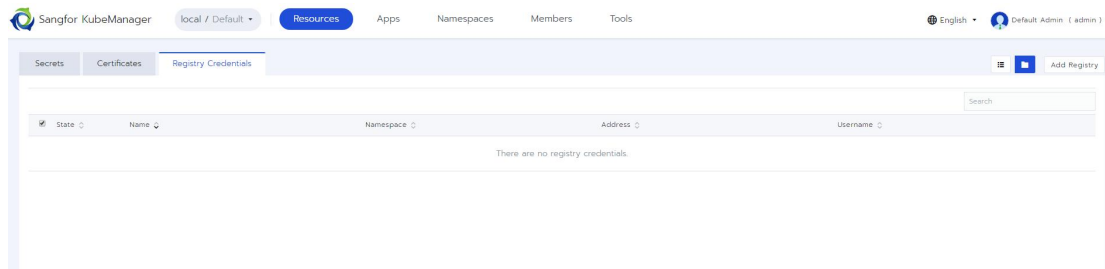
Password

4.3 Registry Configuration

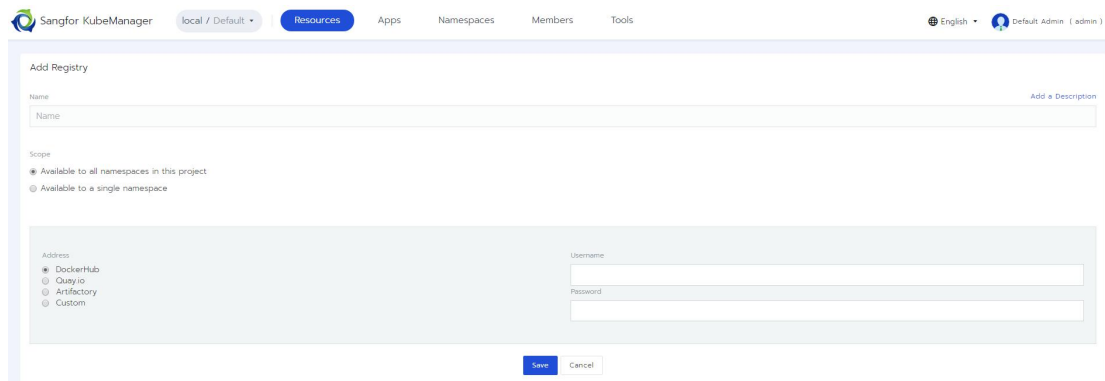
The private registry refers to the registry where system components are located when the cluster is installed. In reality, users often use multiple registries. The KubeManager successfully installed has a default registry, which can be accessed globally by default.

KubeManager allows customers to configure multiple different registries, that is, configuring one or more different registries in different projects. The registry can be either a public or a private one. For a private one, you need to provide a username and password.

Click into the project where the registry is to be configured. Choose **Secerets** from the drop-down menu of **Resources**, and click **Add** in **Registry Credentials** to configure the registry:



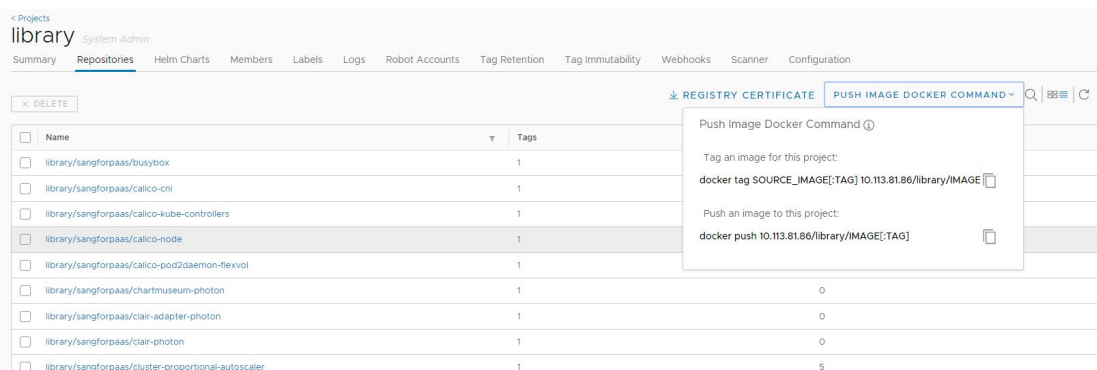
The configuration process is as follows:



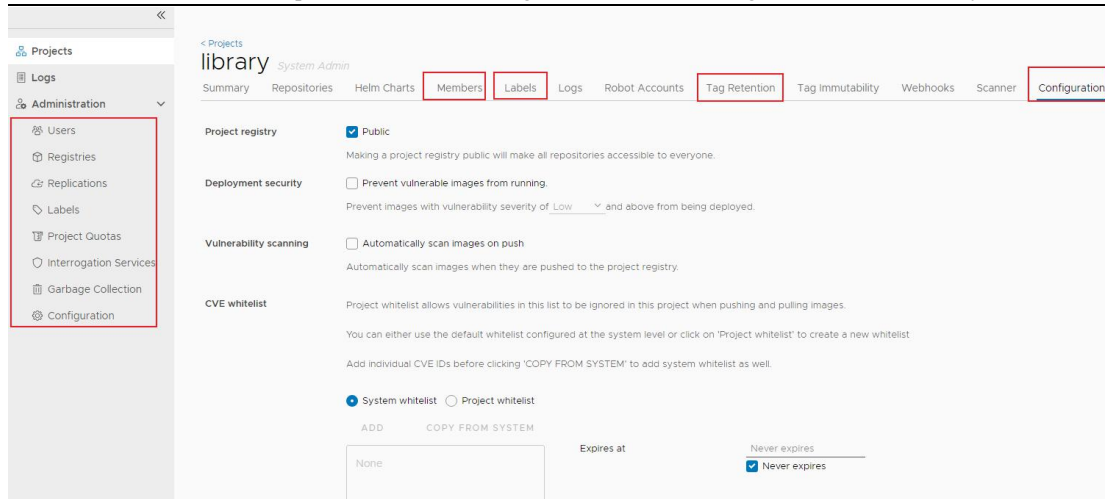
You can configure an effective namespace for the registry and set the registry type (any type is acceptable). We recommend using https registry as much as possible.

4.4 Image Upload and Management

We recommend uploading the image by docker push. First, download the https certificate from the registry, label the image with docker's command, and then push it to the registry.



In the registry, you can manage the image version, registry's permissions, and security policies. The permission and management system of the registry are independent of the use system of KubeManager.



5 App Store

5.1 Deployment and High Availability

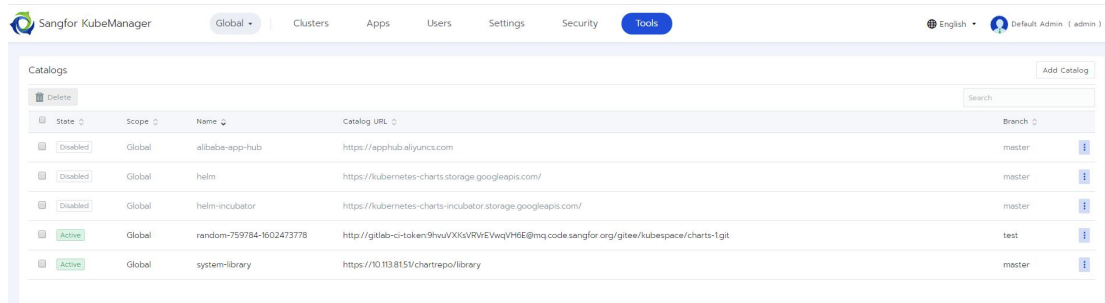
The app store of KubeManager system is deployed together with KubeManager, and it also runs on K8S. The database of the App Store also runs in pod, but the high availability is maintained by our operator.

By default, the system's app store and the registry are implemented through the same Harbor.

5.2 Configuration of App Store

Like other resources, the app store can be set at the **global**, **cluster** and **project** levels. The app store can be public or private.

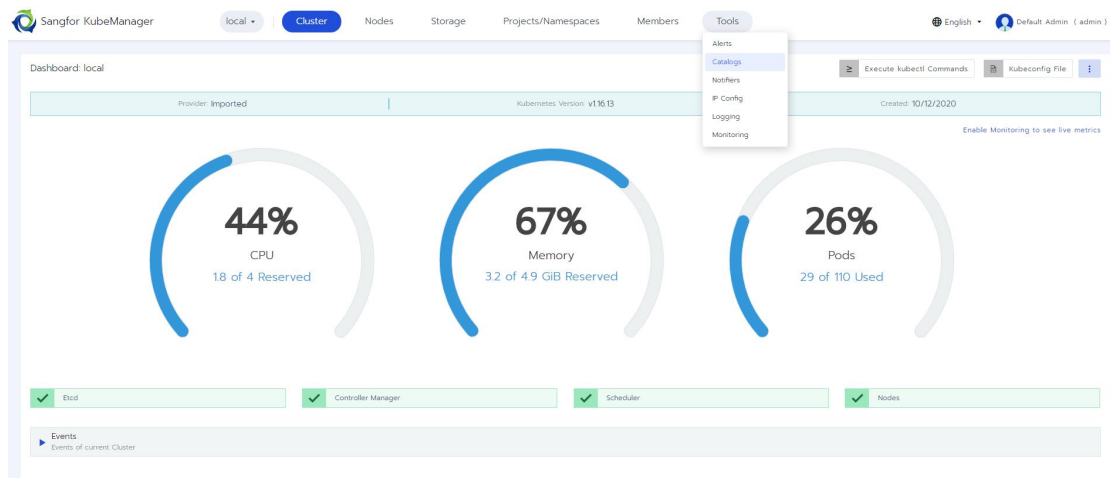
Choose **Catalogs** from the drop-down menu of **Tools** at the **Global** level, to configure the app store at the global level:



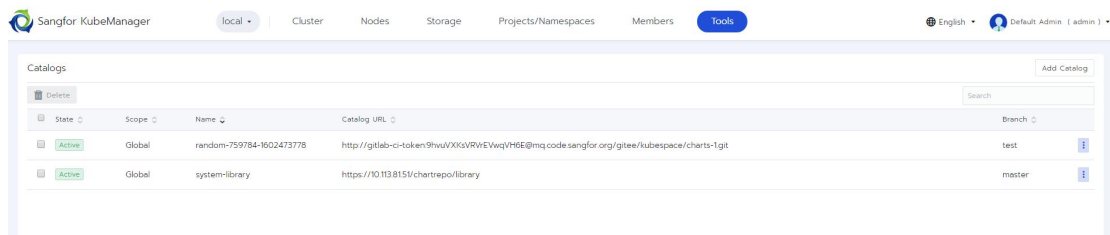
The app store at the global level can be used in all clusters. If some clusters cannot get some images from the app store at the global level, there will be an error of image acquisition failure.



Select the cluster, and choose **Catalogs** from the drop-down menu of **Tools**, to set the app store at the **cluster** level:



Then choose **Add Catalog** to set the store in the project:



Configure the app store as follows:

Add Catalog

Name

Catalog URL *

☒ Use private catalog

Username *

Your username

Password *

Your password

Branch

master

Scope

cluster

Create

Cancel

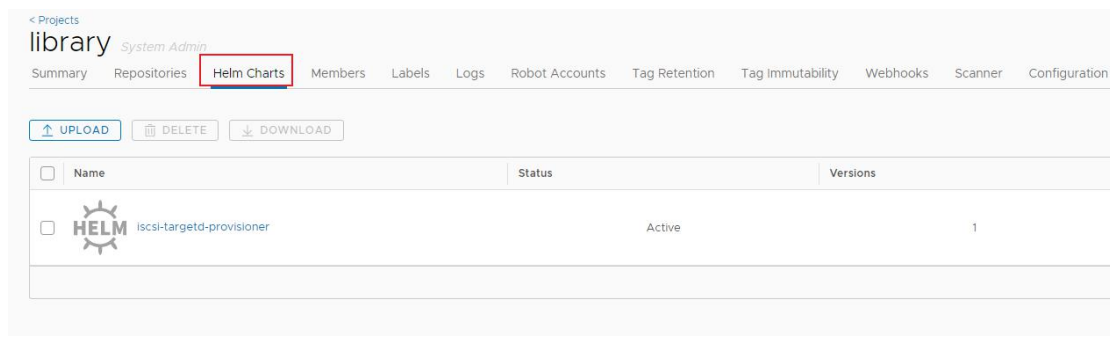


5.3 Application Upload and Management

To upload a new App developed by the existing application to the app store, follow these two steps:

- Upload the image required by the application to the corresponding registry
- Upload chat package to app store

The first step is the same as that for the registry. The second step is to operate on the Harbor page as follows:



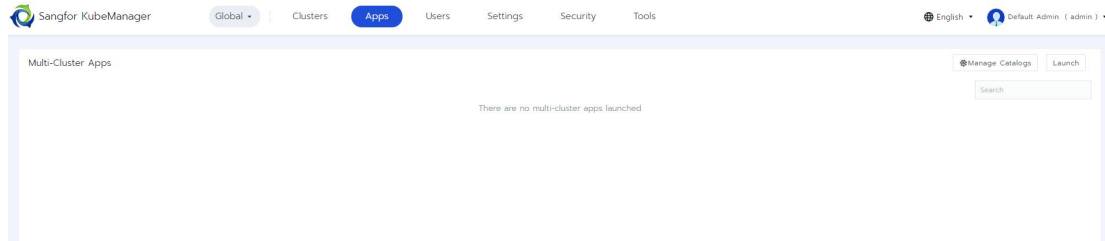
Upload Chart Files

Chart File	No file selected	BROWSE
Prov File	No file selected	BROWSE
		CANCEL UPLOAD

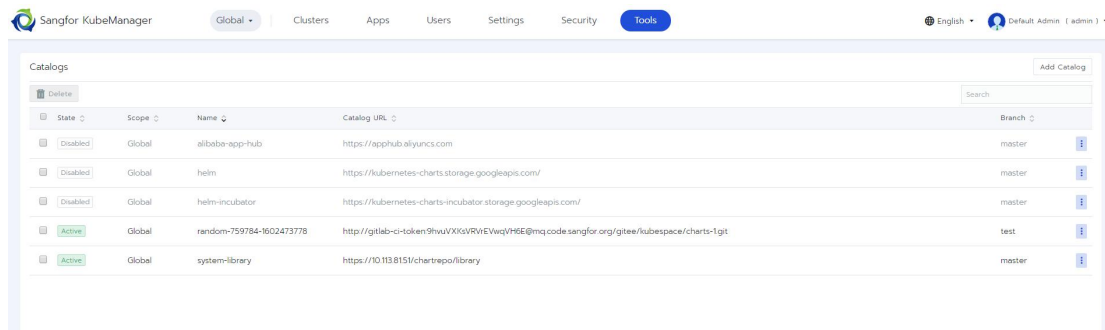
5.4 Multi-cluster Applications

KubeManager can create multi-cluster applications in addition to the creation of applications in the app store. The multi-cluster application allows the same application to be deployed in different projects of multiple clusters, and supports the unified management, upgrade, rollback and other maintenance operations. Multi-cluster applications at the global level: the application in the app store at the global level can be deployed in multi-cluster applications.

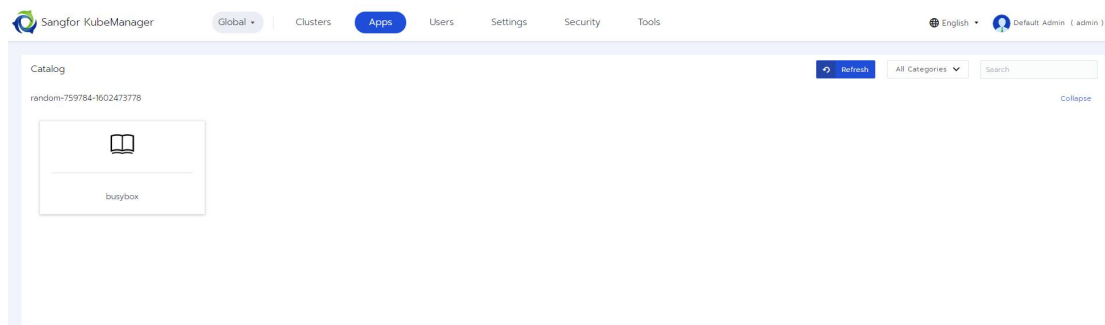
Click the **Apps** menu at the **Global** level:



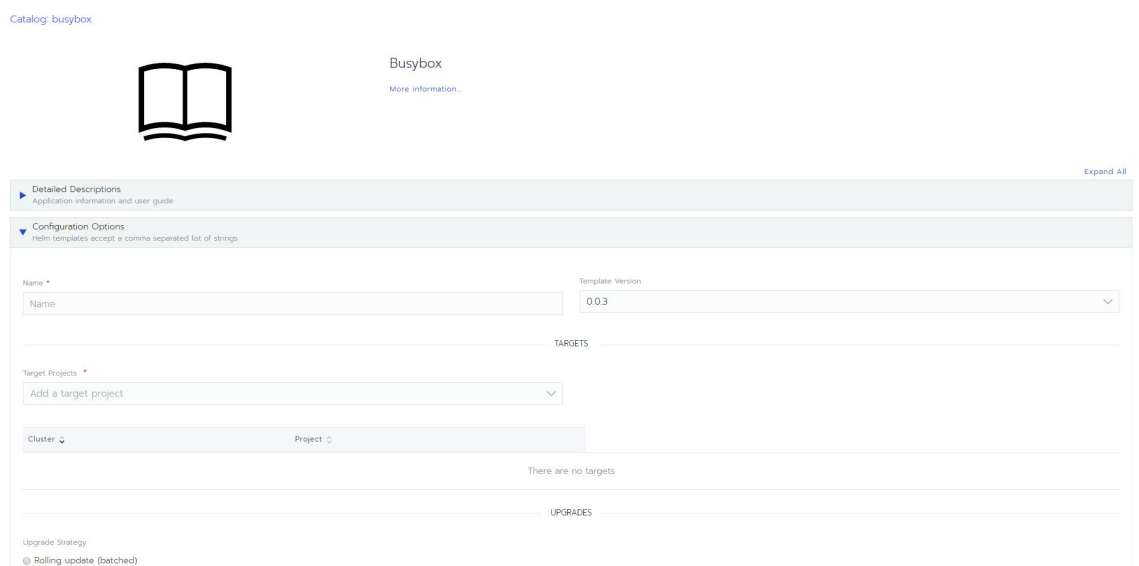
Click **Manage Catalogs** to configure and manage the applications in app store at the global level:



Click **Launch** to load all applications in the app store and classify them according to the store:



Click on the application in the app store to deploy multi-cluster applications:





6 Storage and Use

KubeManager supports a variety of common storage services through the built-in CSI driver, such as Amazon EBS Disk, Azure Disk, google persistent Disk, VMWare vSphere Volume, NFS, and sangfor asan. If deployed with Sangfor cloud devices such as Sangfor HCI, it could provide services by directly using HCI's virtual machine storage.

Moreover, it supports other types of storage too, such as glusterfs and ceph, as long as the user provides provisioner.

6.1 Storage Server

For NFS and sangfor asan, for the management convenience, we manage them via storage servers.

A storage server can manage multiple PVs and SCs.

Click a cluster, and choose **Storage Servers** from the drop-down menu of **Storage**. The list of storage servers is displayed:

State	Name	Type	IP Address
Active	test-random-202928-1602474018	Sangfor NFS	3.3.3
Active	test-random-984621-1602474019	Sangfor aSAN	https://3.3.3.4430
Active	test-random-727305-1602473982	Sangfor aSAN	https://3.3.3.4430
Active	test-random-306511-1602473991	Sangfor NFS	3.3.3
Active	test-random-65798-1602474013	Sangfor aSAN	https://3.3.3.4430
Active	test-random-880462-1602474008	Sangfor NFS	3.3.3
Active	test-random-185853-1602474009	Sangfor NFS	3.3.3
Active	test-random-445861-1602474002	Sangfor aSAN	https://3.3.3.4430
Active	test-random-370561-1602473988	Sangfor aSAN	https://3.3.3.4430

Click **Add Storage Server** to add a storage server:

Add Storage Server

Name *

Type *

Sangfor NFS

Server Info
Basic configuration info of the server

Server Address *
Example: 10.10.3.211

Storage Path *
Example: /var/nfs

Version *
Choose a version

Save Cancel

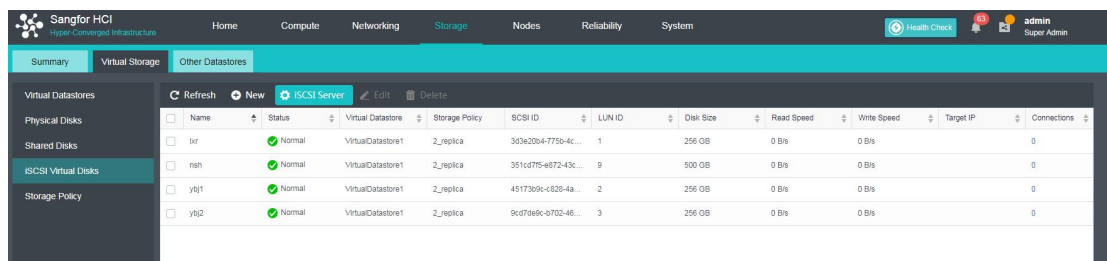
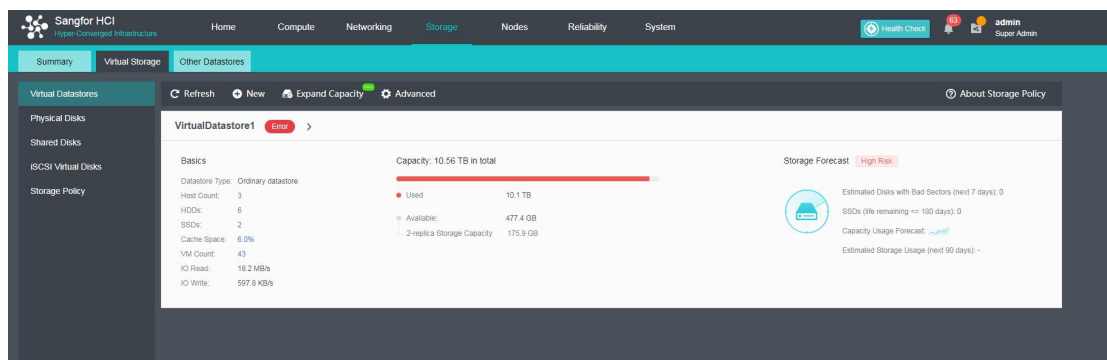
The storage server supports two types of servers, namely sangfor asan and sangfor nfs. The

sangfor asan is the asan storage provided by Sangfor. The angfor nfs is the standard NFS CSI provided by us, being able to integrate with general NFS servers and provide NFS CSI support.

6.2 Creation of Storage Class

The storage class is the most common way for PaaS platform to use storage. The process of creating a storage class is as follows:

1. Create a storage server, as described above;
2. Go to sangfor HCI, create storage volume, and configure iSCSI server:





iSCSI Server

Refresh

Virtual Datastore	Authentication	Target IP	Operation
VirtualDatastore1	iqn.2015-08.21130e11.com.sangfor.asan	<div>Not configured</div>	Settings

Close

Configure iSCSI Server (VirtualDatastore1)

Authentication

Target Portal

Target Name Prefix:

iqn.2015-08.21130e11.com.sangfor.asan

CHAP Username:

admin

CHAP Password:

Confirm Password:

[Change Password](#)

OK

Cancel



3. Create a storage class based on the created storage server, storage volume and ISCSI server:

The screenshot shows the 'Add Storage Class' interface in Sangfor KubeManager. The top navigation bar includes 'local', 'Cluster', 'Nodes', 'Storage' (selected), 'Projects/Namespace', 'Members', and 'Tools'. The user is logged in as 'Default Admin (admin)'. The form has the following sections:

- Name:** A text input field with the placeholder 'e.g. storage' and a link 'Add a Description'.
- Provisioner:** A dropdown menu currently showing 'Amazon EBS Disk'.
- Parameters:** A section titled 'Configure the provider-specific parameters for the storage class' containing three columns of options:
 - Volume Type:** Radio buttons for 'GP2 - General Purpose SSD' (selected), 'IO1 - Provisioned IOPS SSD', 'ST1 - Throughput-Optimized HDD', and 'SC1 - Cold-Storage HDD'.
 - Availability Zone:** Radio buttons for 'Automatic: Zones the cluster has a node in' (selected) and 'Manual: Choose specific zones'.
 - Encryption:** Radio buttons for 'Enabled' and 'Disabled' (selected).
- Customize:** A section titled 'Customize advanced options' containing:
 - Reclaim Policy:** Radio buttons for 'Delete volumes and underlying device when volume claim is deleted' (selected) and 'Retain the volume for manual cleanup'.
 - Mount Options:** A button labeled '+ Add Option'.

At the bottom of the form are 'Save' and 'Cancel' buttons.

Configure the storage class according to the configuration in HCI.

6.3 Creation of PV

You can create PV through SC or the static volume.

Creation of PV via SC: When SC is referenced by creating **Volumes**, PV will be automatically created when **Volumes** is used. The operating steps are as follows:

The top screenshot shows the 'Volumes' tab in Sangfor KubeManager. The navigation bar includes 'test / Default', 'Resources' (selected), 'Apps', 'Namespaces', 'Members', and 'Tools'. The user is logged in as 'Default Admin (admin)'. The 'Volumes' tab has a search bar and a table with columns: State, Claim Name, Size, Persistent Volume, Storage Class, and Related Workloads. The table is empty, with a message 'There are no persistent volume claims defined'.

The bottom screenshot shows the 'Add Volume Claim' form. It includes the following fields and sections:

- Name:** A text input field with the placeholder 'e.g. myvolume'.
- Namespace:** A dropdown menu currently showing 'busybox' and a link 'Add to a new namespace'.
- Source:** Radio buttons for 'Use a Storage Class to provision a new persistent volume' and 'Use an existing persistent volume' (selected).
- Persistent Volume:** A dropdown menu with the placeholder 'Select a persistent volume...'.
- Customize:** A section titled 'Customize advanced options' containing:
 - Access Modes:** Radio buttons for 'Single Node Read-Write' (selected), 'Many Nodes Read-Only', and 'Many Nodes Read-Write'.

At the bottom of the form are 'Create' and 'Cancel' buttons.

Directly create PV with the storage volume, as shown below:



The screenshot shows the Sangfor KubeManager interface. At the top, there's a navigation bar with tabs: test, Cluster, Nodes, Storage (selected), Projects/Namespace, Members, and Tools. Below the navigation bar, there's a dropdown menu for 'Storage' with options: Persistent Volumes, Storage Classes, and Storage Servers. The main content area is titled 'Persistent Volumes' and contains a table with columns: State, Name, Persistent Volume Claim, and Source. Below the table, it says 'There are no persistent volumes defined'. Below the table, there's a form titled 'Add Persistent Volume'. The form has fields for Name (e.g. myvolume), Volume Plugin (Choose a volume plugin...), Capacity (10 GB), Plugin Configuration (Choose a Volume Source above...), Customize (Access Modes: Single Node Read-Write, Many Nodes Read-Only, Many Nodes Read-Write; Mount Options: Add Option; Assign to Storage Class: None; Node Affinity: Add Node Selector), and buttons for Save and Cancel.

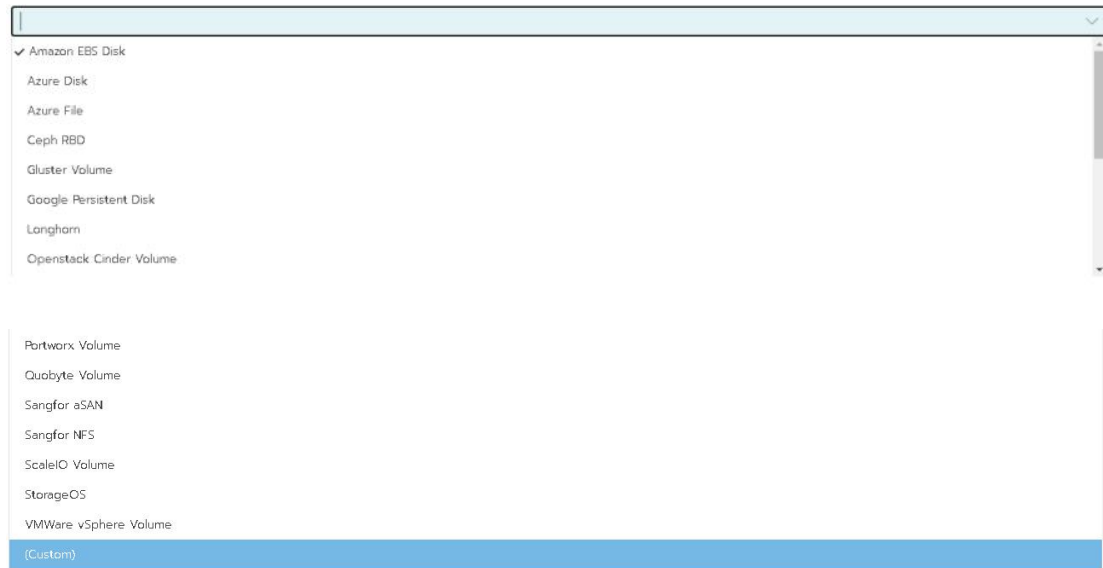
6.4 Support to Other Storage

KubeManager allows to create SC and PV via a variety of common storage services, and provides built-in CSI plug-ins.

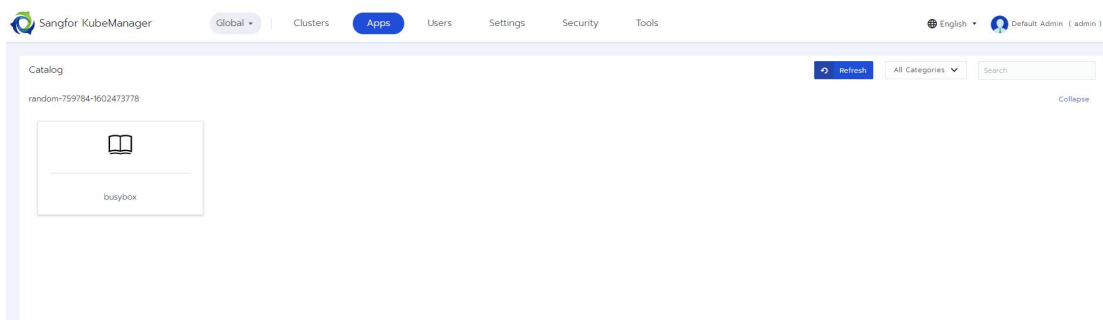
Commonly used volume plug-ins for creating PV are as follows:

The screenshot shows the 'Volume Plugin' dropdown menu. The menu is open, showing a list of options: Choose a volume plugin..., Amazon EBS Disk, Azure Disk, Azure Filesystem, Google Persistent Disk, Local Node Disk, Local Node Path, Longhorn, Sangfor aSAN, Sangfor NFS, and VMWare vSphere Volume.

Commonly used provisioners for creating SC are as follows:



Custom is a user-defined provisioner, which can be provided in the app store, for example:



After creating provisioner, you can provide the name of provisioner in a Custom way, create a storage class and then use the specific storage.

7 Multi-cluster Management

KubeManager provides powerful multi-cluster management capability, which is one of the core functions of KubeManager, including multi-cluster application, creating multi-cluster and supporting various K8S cluster forms.

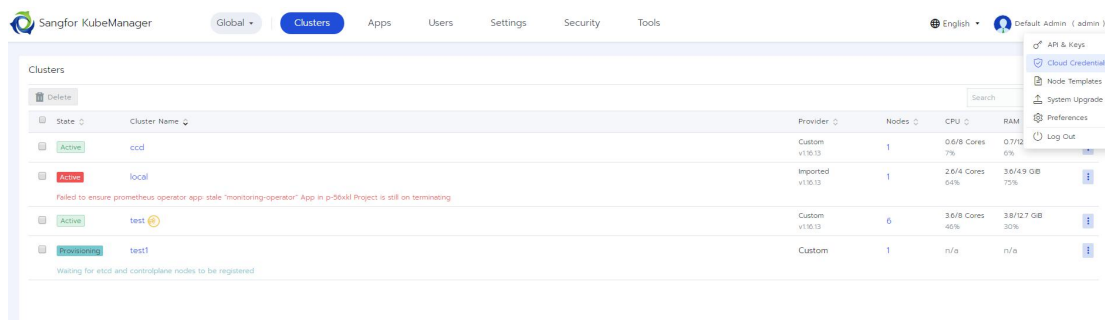
As the multi-cluster applications have been described in the section of **App Store**, this section mainly describes the other two. KubeManager allows a variety of K8S clusters to provide PaaS layer services for users, including custom clusters, hosts from cloud service providers, and Kubernetes hosting services.



7.1 K8S Clusters on "Hosts from Cloud Service Providers"

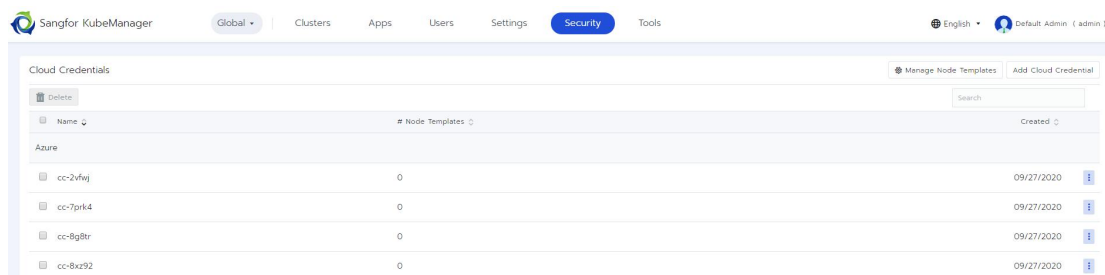
For the hosts provided by cloud service providers, we only need to integrate the drivers provided by cloud service providers through cloud provisioner, and configure the corresponding accounts to KubeManager. In this way, we can seamlessly create and manage the virtual machines on another platform and the Kubernetes on the virtual machines through the KubeManager platform.

Create Cloud Credentials:



The cloud credential page has two options: **Add Cloud Credential** and **Manage Node Templates**.

The cloud credential is the authentication credential for accessing cloud service providers:



Add Cloud Credential:

Add Cloud Credential

Name

Add a Description

Cloud Credential Type

Amazon

Region *

Select a region

Access Key *

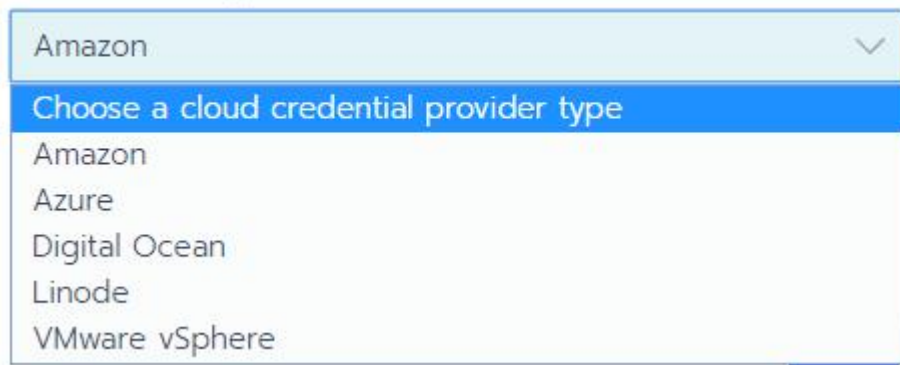
Secret Key *

Add

Cancel

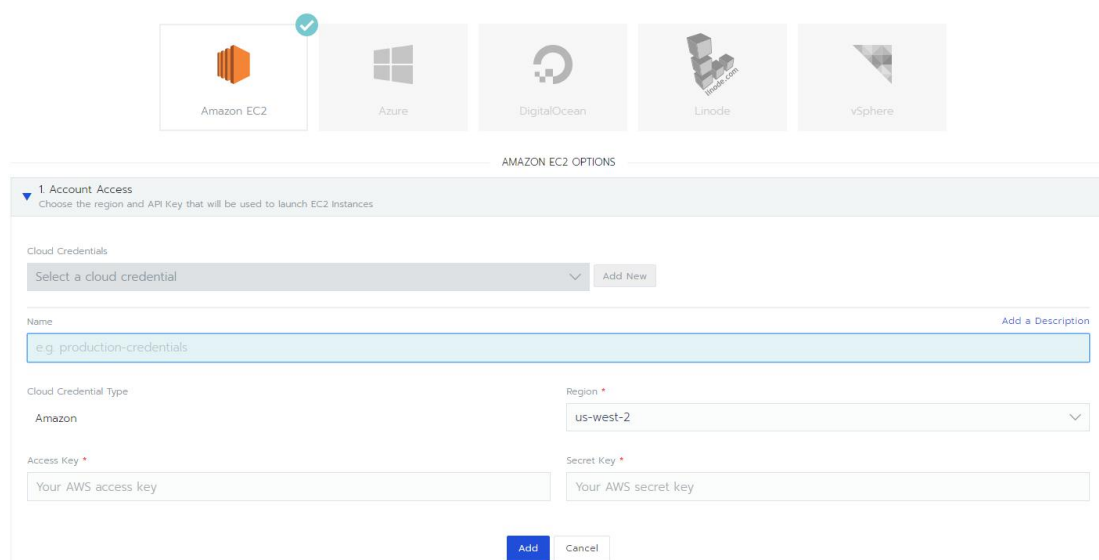


Cloud Credential Type

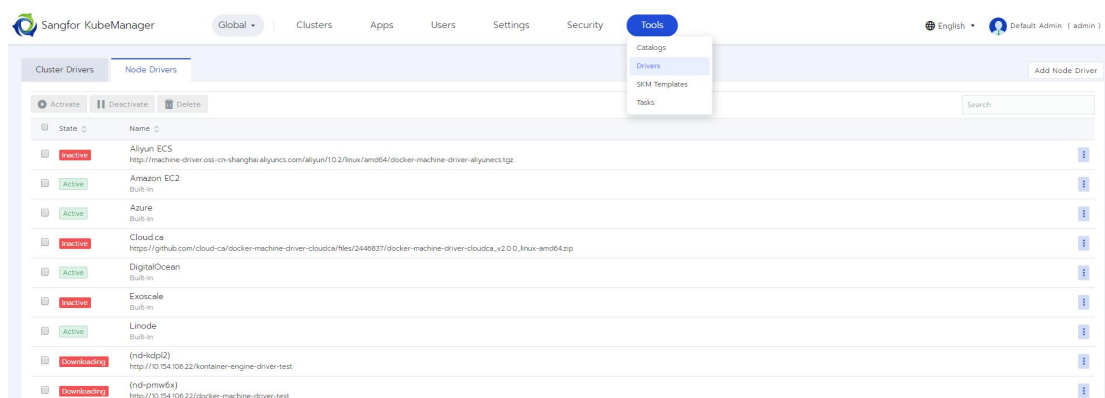


Manage Node Templates, which is used for creating a host at a cloud service provider, including such information as region and model:

Add Node Template



After configuring host credentials and cluster templates, we can create clusters of service providers and K8S services.

Add Node Drivers:

Create a cluster of service provider's virtual machine:

Sangfor KubeManager Global Clusters Apps Users Settings Security Tools English Default Admin (admin)

Add Cluster - Select Cluster Type

From existing nodes (Custom)
Create a new Kubernetes cluster using SKM, out of existing bare-metal servers or virtual machines.

With SKM and new nodes in an infrastructure provider

Amazon EC2 Azure DigitalOcean Linode vSphere

With a hosted Kubernetes provider

Amazon EKS Azure AKS Google GKE

Cancel

Sangfor KubeManager Global Clusters Apps Users Settings Security Tools English Default Admin (admin)

Add Cluster - Amazon EC2

Cluster Name *
e.g. sandbox

Add a Description

Name Prefix	Count	Template	Auto Replace	etcd	Control Plane	Worker	Taints
	1	Add Node Template	0 minutes	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Taints

Number of nodes required: 1, 3, or 5 1 or more 1 or more

[Add Node Pool](#) New Nodes Allowed 1

Member Roles
Control who has access to the cluster and what permission they have to change it.

Labels & Annotations
Configure labels and annotations for the cluster. None

Cluster Options [Edit as YAML](#)

☐ Use an existing SKM Template and revision

Kubernetes Options [Expand All](#)
Customize the Kubernetes cluster options

Kubernetes Version
v1.16.13-sangfor-1

7.2 Cluster of "Kubernetes Hosting Service Providers"

Add Cluster Driver:

Sangfor KubeManager Global Clusters Apps Users Settings Security Tools English Default Admin (admin)

Cluster Drivers **Node Drivers**

[Activate](#) [Deactivate](#) [Delete](#)

[Catalogs](#)
[Drivers](#)
[SKM Templates](#)
[Tasks](#)

[Add Cluster Driver](#)

State	Name	URL
Inactive	Alibaba ACK	https://github.com/rancher/kontainer-engine-driver-aliyun/releases/download/v0.2.5/kontainer-engine-driver-aliyun-linux
Active	Amazon EKS	Built-in
Active	Azure AKS	Built-in
Inactive	Baidu CCE	https://github.com/rancher/kontainer-engine-driver-baidu/releases/download/v0.2.0/kontainer-engine-driver-baidu-linux
Active	Google GKE	Built-in
Inactive	Huawei CCE	https://github.com/rancher/kontainer-engine-driver-huawei/releases/download/v0.1.2/kontainer-engine-driver-huawei-linux
Downloading	(k8s-25scf)	http://xxx

Create a cluster of K8S services:



Sangfor KubeManager Global Clusters Apps Users Settings Security Tools English Default Admin (admin)

Add Cluster - Select Cluster Type

From existing nodes (Custom)
Create a new Kubernetes cluster using SKM, out of existing bare-metal servers or virtual machines.

With SKM and new nodes in an infrastructure provider

Amazon EC2 Azure DigitalOcean Linode vSphere

With a hosted Kubernetes provider

Amazon EKS Azure AKS Google GKE

Cancel

Sangfor KubeManager Global Clusters Apps Users Settings Security Tools English Default Admin (admin)

Add Cluster - Amazon EKS

Cluster Name * Add a Description
e.g. sandbox

Note: Currently Amazon EKS will not create an ingress controller when launching a new cluster. If you need this functionality, you will have to create an ingress controller manually after cluster creation.

Member Roles
Control who has access to the cluster and what permission they have to change it.

Labels & Annotations
Configure labels and annotations for the cluster. None

Account Access
Choose the region and API key that will be used to launch Amazon EKS.

Region: us-west-2 Access Key: Your AWS access key Secret Key: Your AWS secret key Session Token Optional: Your AWS session token

Put in your AWS key pair here. Use only IAM access keys, using keys generated from the root user will make the cluster unreachable.

Next: Configure Cluster Cancel

8 Project Configuration

For a project, the smallest system of the permission system, you can create or delete a project, grant permissions to a project, or restrict the use resources and quotas of a project.

8.1 Creation of Project

Select the corresponding cluster, and click **Projects/Namespaces**, to go to the project list page. Then, you can see all the projects and their namespaces. Click **Add Project** to create new project:



Sangfor KubeManager local Cluster Nodes Storage **Projects/Namespaces** Members Tools English Default Admin (admin)

Projects/Namespaces

Move Download YAML Delete

State Namespace Name Created

Not in a project Add Namespace

Active	cattle-global-nt	09/27/2020
Active	cattle-prometheus-p-2q785	09/27/2020
Active	cattle-prometheus-p-2vnggh	09/27/2020
Active	cattle-sangfor-logging-p-259qy	09/27/2020
Active	logging	09/26/2020
Active	pgo	09/26/2020
Active	random-482155-1601197349	09/27/2020
Active	skm-system	09/26/2020

Project: Default
Default project created for the cluster

Active default 09/26/2020

Sangfor KubeManager local Cluster Nodes Storage **Projects/Namespaces** Members Tools English Default Admin (admin)

Add Project

Project Name * e.g. lab Add a Description

Members
Configure who has access to the resources in this project and what permissions they have

Name	Role
Default Admin (admin) Local User	Project Owner

+ Add Member

Resource Quotas
Configure how much of the resources the project can consume

Container Default Resource Limit
Configure how much of the resources the container can consume by default

Labels & Annotations
Configure labels and annotations for the project. None

Create Cancel

8.2 Namespace Management

On the project list page of the cluster, you can create and delete a namespace.

Sangfor KubeManager local Cluster Nodes Storage **Projects/Namespaces** Members Tools English Default Admin (admin)

Projects/Namespaces

Move Download YAML Delete Namespace

State Namespace Name Created

Not in a project Add Namespace

Active	cattle-global-nt	09/27/2020	⋮
Active	cattle-prometheus-p-2q785		
Active	cattle-prometheus-p-2vnggh		
Active	cattle-sangfor-logging-p-259qy		
Active	logging	09/26/2020	

Edit
Move
View YAML
View in API
Delete



Add Namespace

Name *

Project

Container Default Resource Limit

CPU Limit

Memory Limit

CPU Reservation

Memory Reservation

Labels & Annotations

Create Cancel

You can limit a namespace by quota.

Similarly, you can also manage namespaces in a project. For non-system default namespaces, you can move them from one project to another:

Projects/Namespaces

Move Download YAML Delete 1 Namespace

State Namespace Name Created

Not in a project

Active cattle-global-nt 09/27/2020

Active cattle-prometheus-p-2q785

Active cattle-prometheus-p-2vng

Active cattle-sangfor-logging-p-259qv

Active logging 09/26/2020

Move

Move namespace: cattle-global-nt

To project:

- None
- test-random-430357-1601128572
- tsft-aalldom-923347-1601128570
- dxxd2
- tsft-aalldom-923347-1601128570
- tsft-aalldom-923347-1601128570
- tsft-aalldom-923347-1601128570
- test-random-023347-1601128570

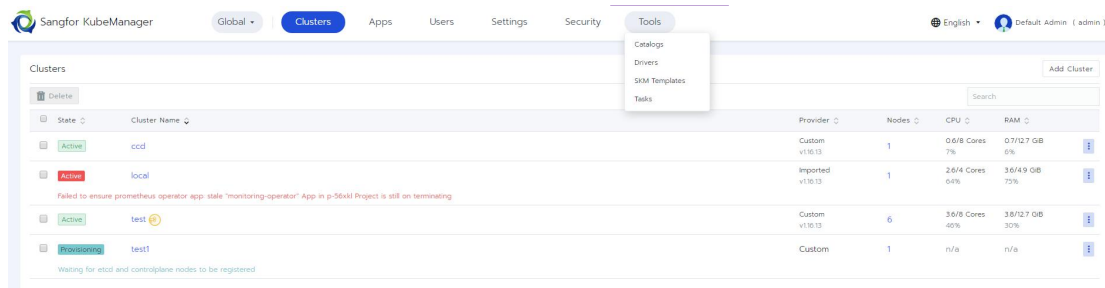
9 System Tools

Most of the auxiliary functions of the platform are set in **Tools**, through which you can set the corresponding functions at **Global**, **Cluster** and **Project** levels.



9.1 Global Settings

In the **Global** mode, there are four options, **Store settings**, **Driver management**, **SKM cluster template**, and **Operation audit**. Among them, Store settings and Driver management have been introduced before. We mainly introduce **SKM cluster management** and **Operation audit** in this section.

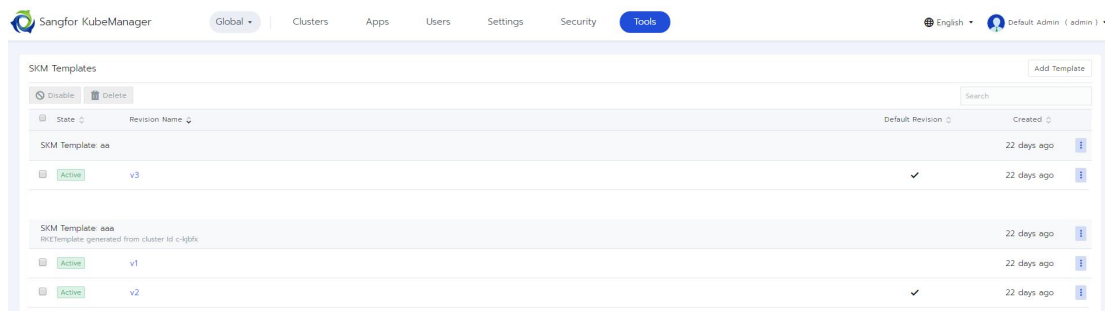


The screenshot shows the 'Clusters' page in Sangfor KubeManager. The top navigation bar includes 'Global', 'Clusters', 'Apps', 'Users', 'Settings', 'Security', and 'Tools'. The 'Tools' dropdown menu is open, showing 'Catalogs', 'Drivers', 'SKM Templates', and 'Tasks'. The main content area displays a table of clusters with columns for State, Cluster Name, Provider, Nodes, CPU, and RAM. There are four clusters listed: 'cccd' (Active), 'local' (Active), 'test' (Active), and 'test1' (Provisioning). A red error message is visible for the 'test1' cluster: 'Failed to ensure prometheus operator app: stale "monitoring-operator" App in p-S6xkl Project is still on terminating'. A green message at the bottom states: 'Waiting for etcd and controlplane nodes to be registered'.

State	Cluster Name	Provider	Nodes	CPU	RAM
Active	cccd	Custom v1.16.13	1	0.6/8 Cores 7%	0.7/12.7 GB 6%
Active	local	Imported v1.16.13	1	2.6/4 Cores 0.4%	3.6/4.9 GB 73%
Active	test	Custom v1.16.13	6	3.6/8 Cores 40%	3.6/12.7 GB 30%
Provisioning	test1	Custom	1	n/a	n/a

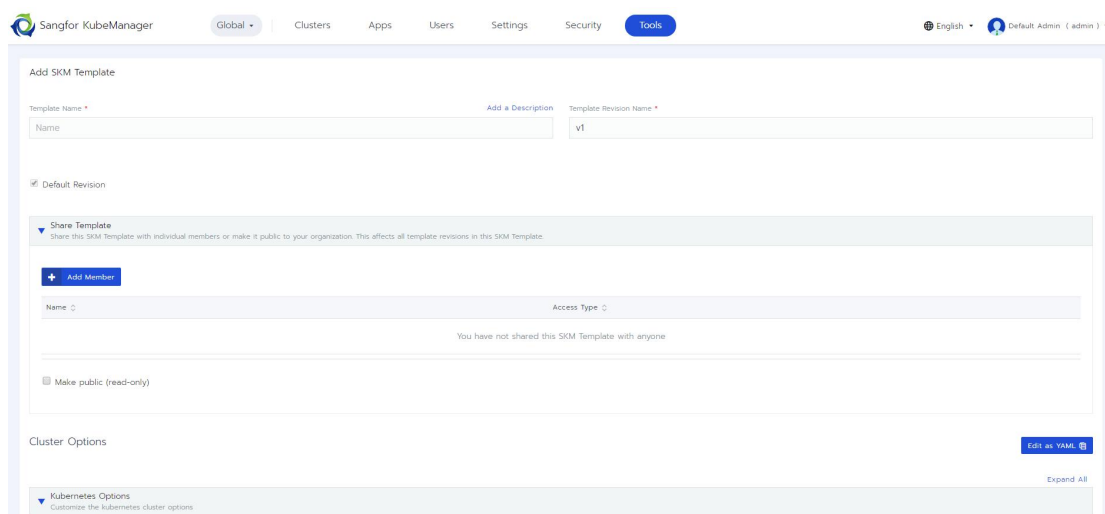
The cluster template is used to set the cluster by applying the previous configuration in this template, and quickly create a cluster.

SKM cluster list page:



The screenshot shows the 'SKM Templates' page in Sangfor KubeManager. The top navigation bar is the same as the previous screenshot. The main content area displays a table of SKM Templates with columns for State, Revision Name, Default Revision, and Created. There are five templates listed: 'aa' (Active), 'v3' (Active), 'aaa' (Active), 'v1' (Active), and 'v2' (Active). The 'v3' template has a checkmark in the 'Default Revision' column. A message at the bottom states: 'Waiting for etcd and controlplane nodes to be registered'.

State	Revision Name	Default Revision	Created
Active	aa		22 days ago
Active	v3	✓	22 days ago
Active	aaa		22 days ago
Active	v1		22 days ago
Active	v2	✓	22 days ago



The screenshot shows the 'Add SKM Template' page in Sangfor KubeManager. The top navigation bar is the same as the previous screenshots. The page has a form for adding a new template. The 'Template Name' field is labeled 'Name'. The 'Template Revision Name' field is labeled 'v1'. The 'Default Revision' checkbox is checked. The 'Share Template' section has a 'Share this SKM Template with individual members or make it public to your organization. This affects all template revisions in this SKM Template.' message. There is an 'Add Member' button. The 'Access Type' dropdown is set to 'You have not shared this SKM Template with anyone'. The 'Make public (read-only)' checkbox is unchecked. The 'Cluster Options' section has a 'Kubernetes Options' subsection with a 'Customize the kubernetes cluster options' message. There is an 'Expand All' button.

Template Name *

Name

Template Revision Name *

v1

☒ Default Revision

Share Template

Share this SKM Template with individual members or make it public to your organization. This affects all template revisions in this SKM Template.

Add Member

Name

Access Type

You have not shared this SKM Template with anyone

☐ Make public (read-only)

Cluster Options

Kubernetes Options

Customize the kubernetes cluster options

Expand All



Kubernetes Version

V1.16.13-sangfor-1.1

Network Provider

Calico

Networking

☒ Auto ☐ Custom

Cloud Provider

If your cloud provider is not listed, please use the **Custom** option.

☒ None

☐ Amazon

☐ Azure

☐ Custom

☐ External

Private Registry

Configure a default private registry for this cluster. When enabled, all images required for cluster provisioning and system add-ons startup will be pulled from this registry.

Private Registry

☒ Disabled

☐ Enabled

Advanced Options

Customize advanced cluster options

Nginx Ingress

☒ Enabled

☐ Disabled

Node Port Range

30000-32767

Metrics Server Monitoring

☒ Enabled

☐ Disabled

Pod Security Policy Support

☐ Enabled

☒ Disabled

Default Pod Security Policy

None

Docker version on nodes

☐ Require a supported Docker version

☒ Allow unsupported versions

Docker Root Directory

/var/lib/docker

etcd Snapshot Backup Target

☒ local

☐ s3

snaphots only exist locally, no external backups are performed

etcd snapshots will occur locally, subsequently the snapshot will be backed up to the configured s3 target.

Recurring etcd Snapshot Enabled

☒ Yes ☐ No

Recurring etcd Snapshot Interval

12

hours

Recurring etcd Snapshot Retention

Keep the last 6

Authorized Endpoint

Enabling the authorized cluster endpoint allows direct communication with the cluster, bypassing the API proxy. Authorized endpoints can be retrieved by generating a kubeconfig for the cluster.

Authorized Cluster Endpoint

☒ Enabled

☐ Disabled

Operation audit: You can trace all the operation history based on the operation records of all users of this platform. Operation audit supports searching and sorting.

Sangfor KubeManager

Global

Clusters

Apps

Users

Settings

Security

Tools

English

Default Admin (admin)

All Tasks

Last 7 days

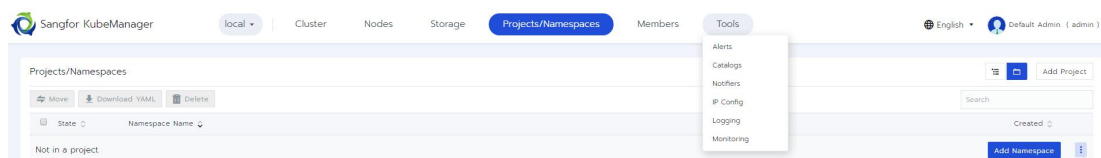
All Object Types

Search

Action	Object	Object Type	Cluster	Project	Operator	Time
Log In	-	API & Keys	-	-	admin (172.23.4.225)	9:11 AM
Log In	-	API & Keys	-	-	admin (172.23.4.225)	Yesterday at 8:03 PM
Edit authentication	openldap	Third-party Authentication	-	-	admin (172.23.12.120)	Yesterday at 10:20 AM
Edit authentication	openldap	Third-party Authentication	-	-	admin (172.23.12.120)	Yesterday at 10:16 AM
Test server	openldap	Third-party Authentication	-	-	admin (172.23.12.120)	Yesterday at 10:16 AM
Edit authentication	openldap	Third-party Authentication	-	-	admin (172.23.12.120)	Yesterday at 10:13 AM
Test server	openldap	Third-party Authentication	-	-	admin (172.23.12.120)	Yesterday at 10:13 AM
Log In	-	API & Keys	-	-	admin (172.23.12.120)	Yesterday at 10:11 AM
Add secret	eee	Secret	ccd	Default	admin (172.23.12.120)	Last Friday at 4:40 PM
Log In	-	API & Keys	-	-	admin (172.23.12.120)	Last Friday at 4:34 PM
Log In	-	API & Keys	-	-	admin (172.23.0.105)	Last Friday at 11:33 AM
Delete cluster	c-qg6z6	Cluster	c-qg6z6	-	admin (10.113.83.13)	Last Thursday at 7:17 PM
Add cluster	test-random-802368-1605181323	Cluster	test-random-802368-1605181323	-	admin (10.113.83.13)	Last Thursday at 7:17 PM

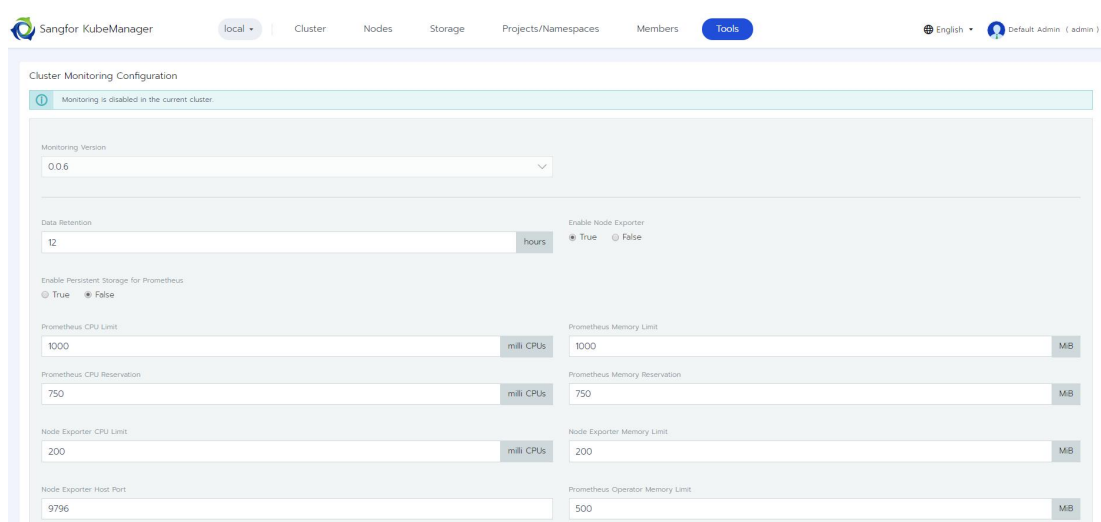
9.2 Cluster Settings

The **Tools** menu in the **Cluster** page mainly includes **Alarm**, **Notification**, **Monitoring**, **Log**, **Store settings**, **Network exit**, etc.



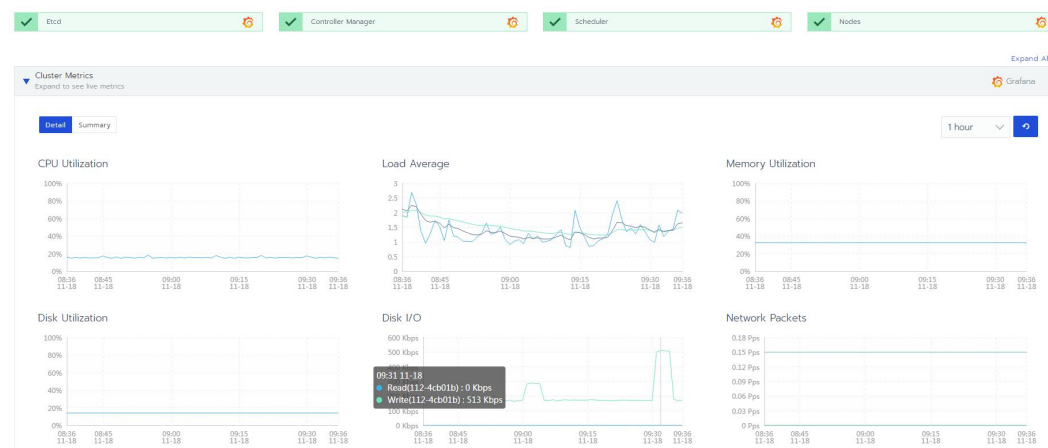
Store settings has been described above, so it will not be further described.

Cluster monitoring:



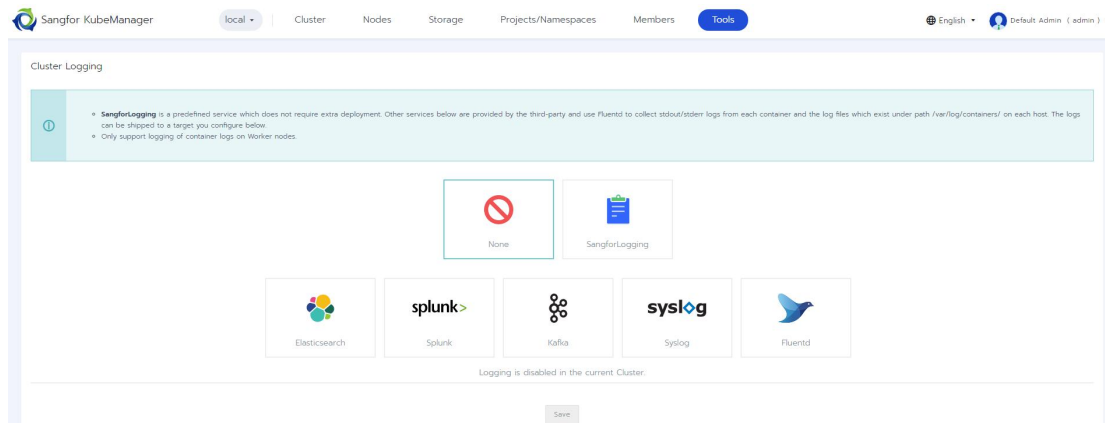
Cluster monitoring allows to use persistent storage and local storage, add node selectors, set tolerance, and support the persistent storage for grafana. Once Cluster monitoring is enabled, you can monitor resources in the cluster.

The cluster page after Cluster monitoring is enabled:

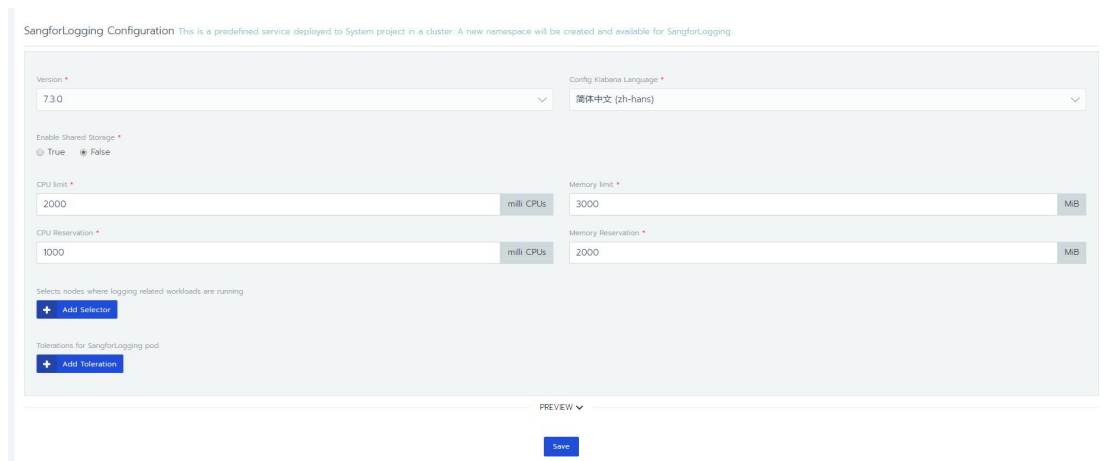


Just like monitoring, you can collect logs through Sangfor Logging or other log collection

methods:

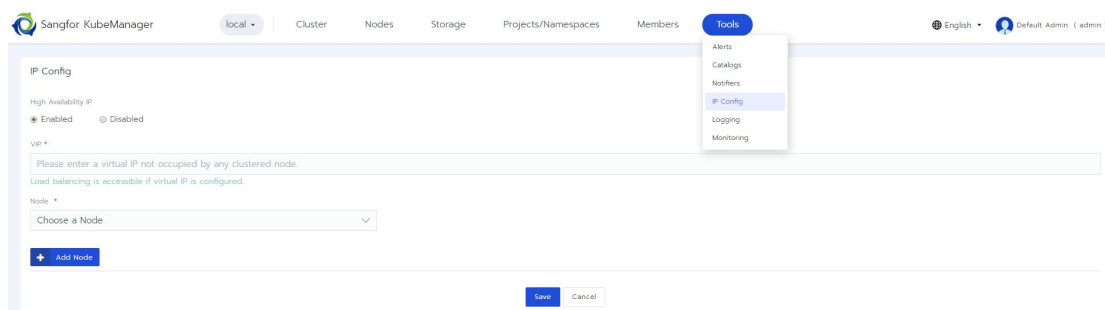


Sangfor Logging allows to configure the persistent storage and local storage:



If local storage is selected, 3 duplicates will be created, and high availability will be guaranteed through ES.

The network exit aims to ensure the high availability of service release. Two nodes are selected to ensure the high availability of the network in the form of vip, and guarantee that the IP for service release can ensure the high availability at different nodes:



The platform supports the alarm via Slack, E-mail, PagerDuty, Webhook, Enterprise WeChat and other means:



Slack

Email

PagerDuty

Webhook

WeChat

Name *

Add a Description

Smtip Server

Sender *

Host *

e.g. 192.168.1.121

Port *

587

☒ Use TLS

Username

e.g. John

Password

Your Password

Default Recipient Address *

e.g. admin@example.com

When configuring an Alert, the recipient can be overridden.

Send Resolved Alerts

☐ Enable

TEST

Add

Cancel

The alarm function aims to alarm the notification group in the notification according to the set rules:

Sangfor KubeManager

local Cluster Nodes Storage Projects/Namespaces Members Tools

English Default Admin (admin)

Cluster Alerts

Add Alert Group

Deactivate Delete

Search

State	Name	Target	Condition	Notifier
A set of alerts for etcd				
Alert for etcd leader existence, db size				
Active	A high number of leader changes within the etcd cl...	Metric	Greater Than 3	Not Configured
Active	Database usage close to the quota 500M	Metric	Greater Than 524288000	Not Configured
Active	Etcd is unavailable	System Service	Unhealthy	Not Configured
Active	Etcd member has no leader	Metric	Not Equal 1	Not Configured
A set of alerts for kube components				
Alert for kube components api server, scheduler, controller manager				
Active	Controller Manager is unavailable	System Service	Unhealthy	Not Configured
Active	Scheduler is unavailable	System Service	Unhealthy	Not Configured
A set of alerts when event happened				
Alert for receiving resource event				
Active	Get warning deployment event	Event Deployment Event	Happens	Not Configured

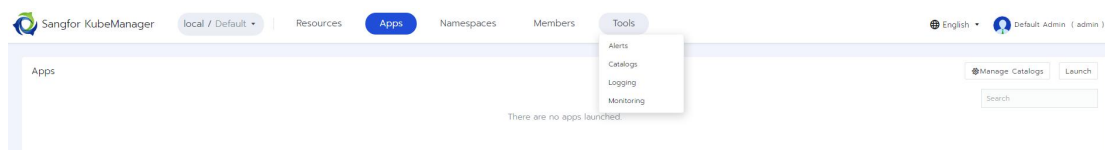
Add Alert Group:



The alarm group mainly consists of alarm rules and notification objects.

9.3 Project Settings

The Tools menu on the project page is the same as that of the cluster page. The only difference is that it applies to the project only. So it will not be further described here.

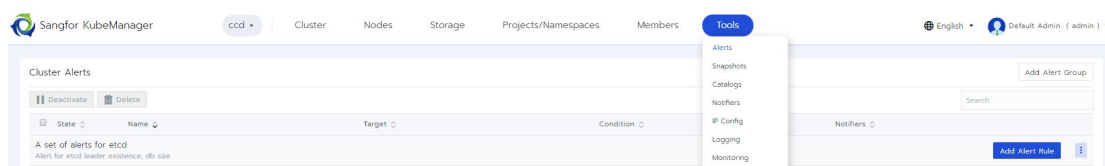


10 Other Configurations

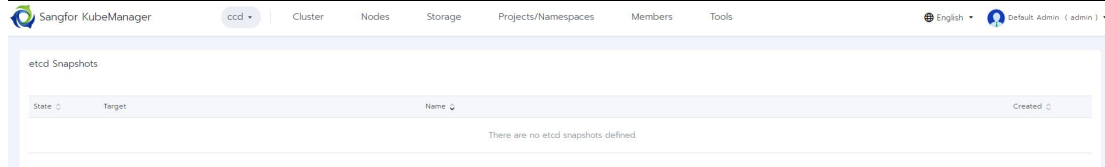
KubeManager also supports some other management functions.

10.1 Backup and Recovery

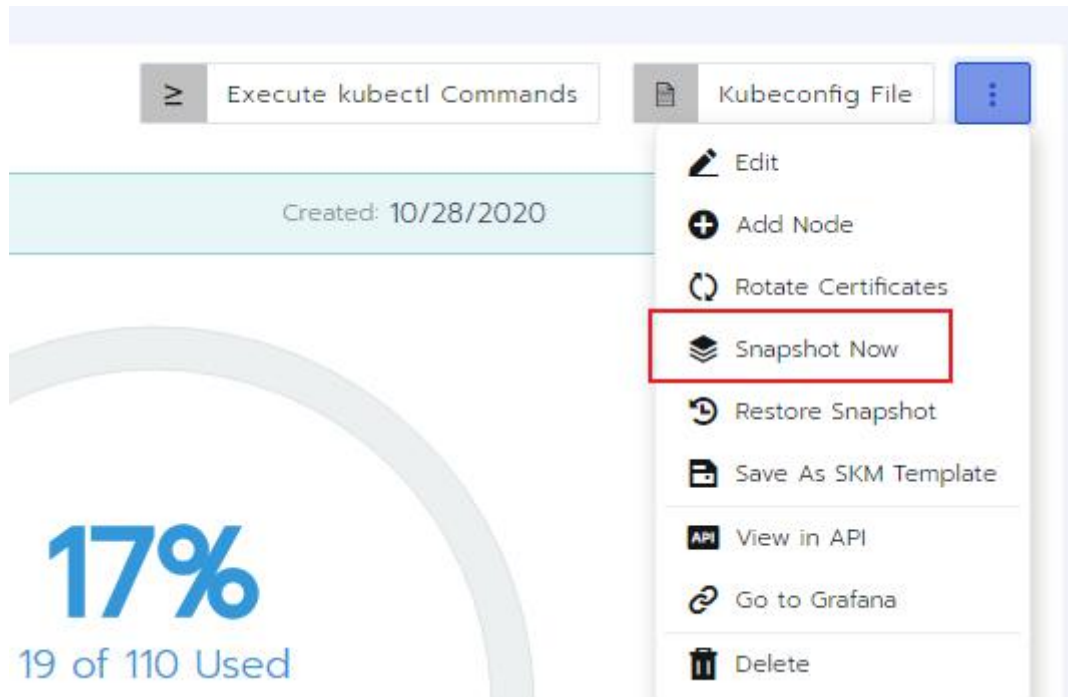
There are two backup methods for cluster. One is local backup, and the other is to backup to S3 storage.



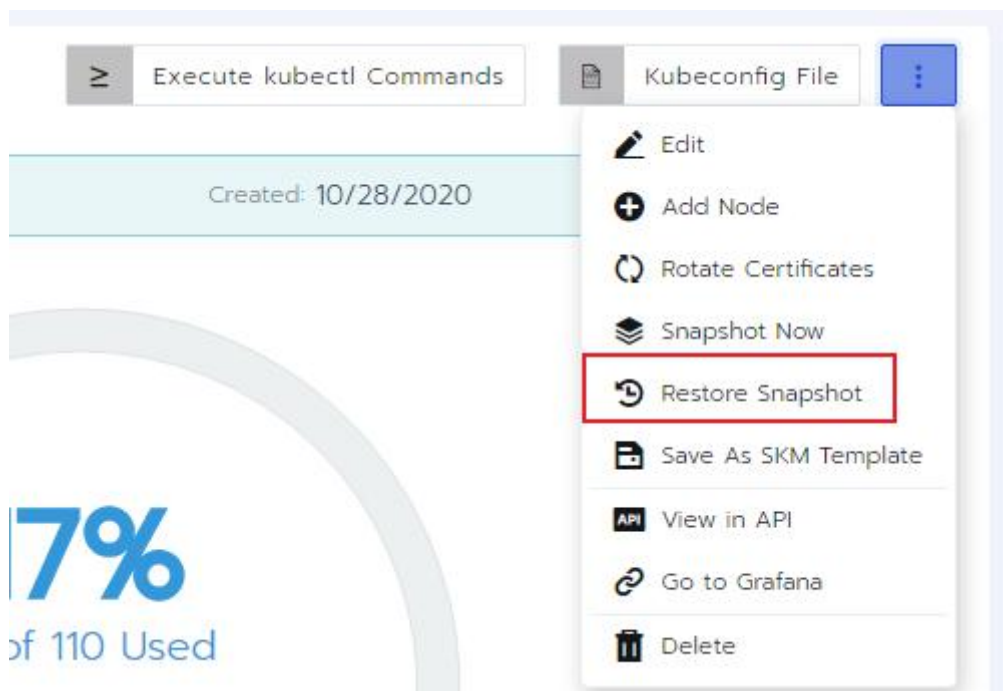
Set the backup mode and policy in the cluster configuration:

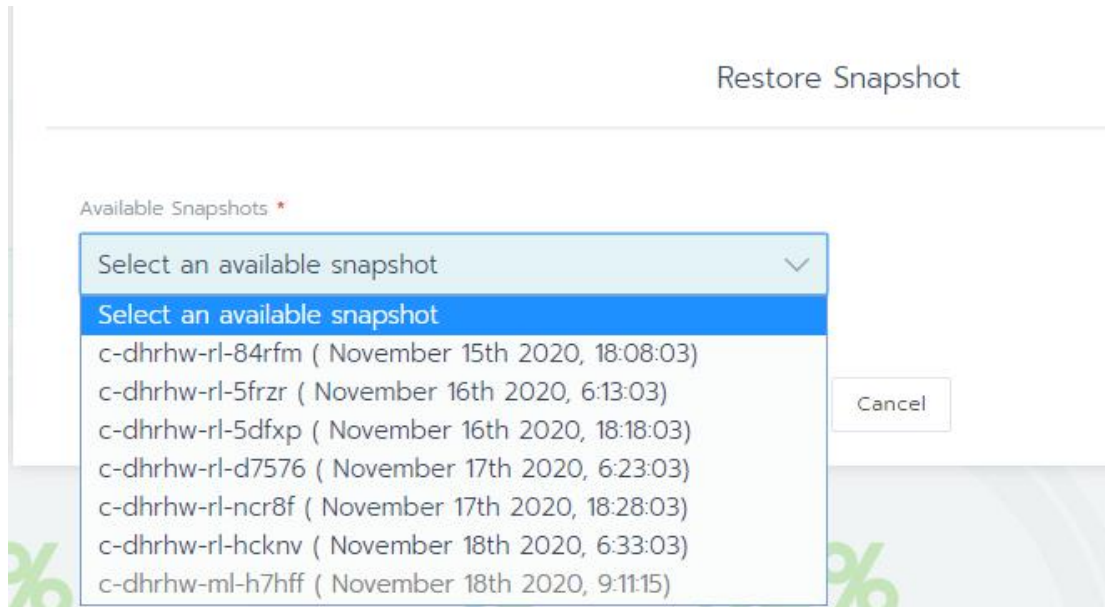


Make a manual backup on the cluster page:



Recover it on the cluster page:

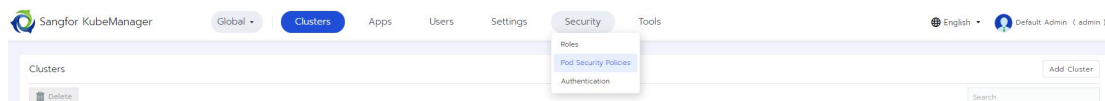




10.2 Security

There are two main types of KubeManager security, namely the image security that has been described in the registry part and the container security. Container security is mainly guaranteed by PSP security policy.

Security policy list:



Add security policy:

Add Policy

Name

eg. policy

Add a Description

Expand All

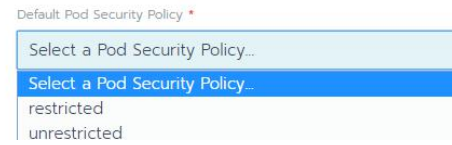
- Basic Policies
Config basic pod security policies
- Capability Policies
Config set of capability policies
- Volume Policy
Control the usage of volume types
- Allowed Host Paths Policy
Whitelist of allowed host paths
- FS Group Policy
Allocating an FSGroup that owns the pod's volumes
- Host Ports Policy
The use of host ports
- Run As User Policy
The user ID
- SELinux Policy
The SELinux context of the container
- Supplemental Groups Policy
Configuring allowable supplemental groups
- Labels & Annotations
Key/Value pairs that can be used to label/annotate resources

Create Cancel

When creating a cluster, you can enable the security policy and select the security policy to be



enabled:



10.3 Precautions

KubeManager provides a sound multi-cluster management function and a user-friendly interface.

During use, please ensure the network access and sufficient system resources, and guarantee that the registry could be connected to the node network.