REFERENCE ARCHITECTURE–APRIL 2023

KubeSphere and VMware Cloud Native Solution

Table of Contents

Executive Summary

Business Case

Running container workloads on VMware vSphere® requires a solution that is cost-effective, highly scalable, and easy to manage. Although Kubernetes delivers a decent solution for container orchestration, its steep learning curve and CLI-based working mode put a huge burden on operation and maintenance (O&M) personnel. On top of Kubernetes, KubeSphere is a container platform that can work together with VMware Cloud Native Storage (CNS) and VMware Antrea to build a one-stop modern application solution.

KubeSphere is an application-oriented multi-tenant container platform, which provides full-stack automated O&M capabilities and simplifies the DevOps workflows for enterprises. KubeSphere can run on Linux VMs powered by VMware vSphere or VMware Tanzu Kubernetes Grid and visualize the management of Kubernetes resources. This solution enables features such as multi-cluster management, logging, monitoring, alerting, notifications, audits, events, and metering and billing. Moreover, KubeSphere provides platform-level capabilities, such as App Store, DevOps, edge computing, and microservices governance.

KubeSphere can work together with VMware CNS and Antrea to achieve infrastructure automation and platform extensibility, which simplifies and automates Day 0 to Day 2 operations with Kubernetes. This solution provides a complete, automated, and one-stop platform for developers and O&M personnel to modernize and manage applications during the whole lifecycle. KubeSphere also helps IT personnel to reduce the learning curve and O&M costs, improves operation efficiency, and fuels digital transformation.

In this solution, we provide infrastructure administrators, developers, and maintenance personnel with the design, deployment, and best practices of KubeSphere on VMware vSphere with VMware vSAN™, to help enterprises run and manage modern applications.

Business Values

Deploying KubeSphere on VMware vSphere with vSAN features the following benefits:

- Quick deployment. You can deploy KubeSphere on native Kuberenetes powered by VMware vSphere and vSAN at ease. vSphere Container Storage Interface (CSI) and Antrea are preinstalled in the KubeSphere App Store, so you can configure storage and networking in a few clicks.
- Independent O&M for IaaS and PaaS. For infrastructure administrators, you only need to operate and maintain Fibre Channel Storage Area Network (FC SAN) or Network Attached Storage (NAS) on vSphere or vSAN. For platform administrators, KubeSphere allows you to focus on how to manage container workloads and DevOps workflows.
- Cluster lifecycle management. KubeSphere simplifies the lifecycle management of Kubernetes clusters on vSphere with vSAN, and provides features such as cluster upgrade, certificate renewal, automatic backup, and parameter management.
- Storage and network security. You can use CNS for VMware vSphere to integrate storage of different types, including DAS, SAN, and NAS, and use a CSI for persistent storage on KubeSphere. Antrea powers KubeSphere to enable multi-layer network policies and traffic observability.
- Out-of-the-box PaaS. KubeSphere on VMware vSphere with vSAN is native to Kubernetes and provides out-of-the-box features, such as DevOps powered by Jenkins or Argo CD, microservices governance powered by Istio or Spring Cloud, containerized database management, and edge computing.
- Observability. KubeSphere provides various features for observability, such as unified logging, monitoring, and alerting, diverse notification channels, events, audits, and billing and metering.
- Multi-cluster or multi-cloud delivery. KubeSphere tailors delivery to meet business needs in active-active or geo-redundancy scenarios. To manage multiple clusters across clouds, KubeSphere supports cluster federation and enables observability for unified O&M.

Key Results

Powered by VMware vSphere and vSAN, KubeSphere achieves the following key results:

- Simplify resource management from the following aspects: Kubernetes-native O&M on VMware vSphere with vSAN, collaboration in open source ecosystems, integration with DevOps tools, and service governance.

- Increase sales for VMware and QingCloud and work together with VMware vSphere, vSAN, and NSX to provide a one-stop cloud native solution for existing enterprise users.

Audience

This solution is suitable for IT administrators, developers, O&M personnel, and engineers and experts engaged in planning, design, and deployment of VMware vSphere, vSAN, Kubernetes, and DevOps workloads. Before you read the following content, make sure you have some knowledge of VMware vSphere, vSAN, Kubernetes, and Docker.

Technology Overview

The following components work collaboratively to implement this solution:

- vSphere
- vSAN
- VMware CNS
- Antrea
- KubeSphere

VMware vSphere

VMware vSphere is a virtualization platform powered by VMware, which can transform a data center into a converged computing infrastructure that provides CPU, storage, and networking resources. VMware vSphere manages the infrastructure as a unified environment and provides you with the tools to manage the data center.

VMware vSAN

VMware vSAN uses a software-defined approach to creating shared storage for virtual machines (VMs). This service can virtualize local physical storage resources provided by ESXi servers and turn them into storage pools in which resources can be provisioned to VMs and applications on demand. vSAN is implemented in the ESXi hypervisor, and vSAN can be configured to work as a hybrid cluster or an all-flash cluster.

VMware CNS

VMware CNS for vSphere integrates with Kubernetes and allows Kubernetes to configure storage on demand on vSphere in a fully automated and scalable manner. This service also enables volume observability for administrators through vCenter.

VMware CNS allows administrators to run, monitor, and manage container and VM storage on the same platform. This simplifies container infrastructure storage, lifecycle management, and operations for administrators. VM and container storage provisioning on a unified platform reduces administrator learning costs and enables consistent operations across workloads and clouds. CNS can help you reduce time in managing infrastructure and focus more on building applications that deliver business value.

Antrea

Antrea is a Kubernetes-native network solution that uses Open vSwitch as the data plane at Layer 3 or 4 to provide network and security services for Kubernetes. Antrea can work as a Container Network Interface (CNI) plug-in to enable various network and security features, such as the configuration of static IP addresses, IP Address Management (IPAM), multi-cluster networking, overlay and underlay network construction, traffic encryption, layered and multi-dimensional network policies, network flow analysis and aggregation, egress gateways, LoadBalancer, and Traceflow.

KubeSphere

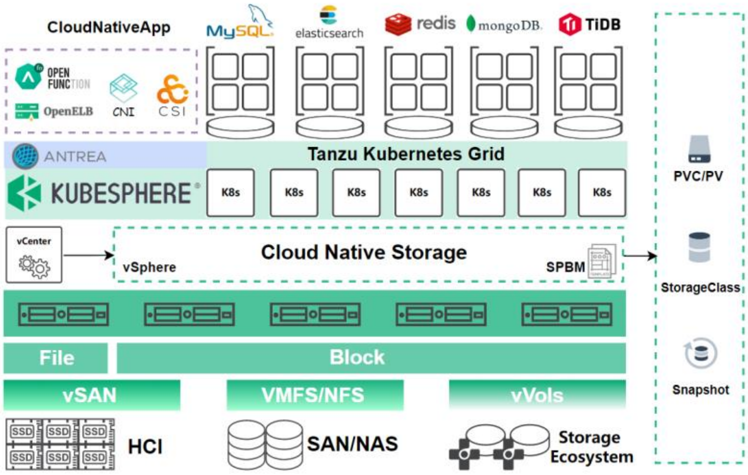
Built on top of Kubernetes, KubeSphere is a distributed operating system for cloud native applications. KubeSphere provides multiple features to build an enterprise-grade Kubernetes environment, such as multi-cloud and multi-cluster management, resource management, DevOps, application lifecycle management, microservices governance or service meshes, log query and collection, services and networks, multi-tenant management, monitoring and alerting, event and audit query, storage management, access control, GPU support, network policies, image registry management, and security management. KubeSphere abstracts away complex technical details of the underlying infrastructure, and helps enterprises seamlessly deploy, update, migrate, and manage existing containerized applications on various types of infrastructure. This way, KubeSphere allows developers to focus on application development, and O&M personnel can leverage enterprise-grade features to speed up DevOps workflows and delivery processes. The features include observability, troubleshooting, unified monitoring and log query, storage and network management, and easy-to-use CI/CD pipelines.

Solution Configuration

This section describes how to implement the solution from the following aspects:

- Architecture
- Hardware resources
- Software resources
- Network configuration

Architecture Diagram



Hardware Resources

Table 1. Hardware resources for KubeSphere on Linux

PROPERTY	SPECIFICATION
OS	Ubuntu 16.04/18.04/20.04 Debian Buster, Stretch CentOS 7.x Red Hat Enterprise Linux 7 SUSE Linux Enterprise Server 15 /openSUSE Leap 15.2
CPU	> 2 vCPUs
RAM	> 4 GB
CRI	Docker, Containerd, CRI-O

Dependency	Socat, Conntrack
Disks	> 40 GB

Software Resources

Table 2. Software resources for KubeSphere on Kubernetes

SOFTWARE	VERSION
Kubernetes	v1.19.x, v1.20.x, v1.21.x, v1.22.x, v1.23.x, and v1.24.x
StorageClass	Exist

Network Configuration

Table 3. Ports used by KubeSphere

SERVICE	PROTOCOL	ACTION	SOURCEPORT	DESTPORT
SSH	TCP	ALLOW	22	
ETCD	TCP	ALLOW	2379	2380
APISERVER	TCP	ALLOW	6443	
NODEPORT	TCP	ALLOW	30000	32767
MASTER	TCP	ALLOW	10250	10258
DNS	TCP	ALLOW	53	
DNS	UDP	ALLOW	53	
LOCAL-REGISTRY	TCP	ALLOW	5000	
LOCAL-APT	TCP	ALLOW	5080	
RPCBIND	TCP	ALLOW	111	
METRICS-SERVER	TCP	ALLOW	8443	

Solution Validation

Test Overview

This section validates the feasibility of integrating KubeSphere with vSphere CSI and Antrea.

Integrate KubeSphere with vSphere CSI

This section guides you through how to install native Kubernetes and Kubesphere on VMs powered by VMware vSphere. In this example, VMware CNS is used for persistent storage, and KubeSphere helps you create workloads backed by the vSphere CSI driver. For example, you can create a stateful MySQL application with a persistent volume claim (PVC), test volume attaching, create volume snapshots, and delete or recover volumes.

Create a Kubernetes Cluster

To create a Kubernetes cluster at ease, we recommend that you use the open source tool KubeKey. KubeKey is an open source tool powered by QingCloud. This tool can help you create KubeSphere clusters and install KubeSphere at ease. You can use this tool to add cluster nodes, install Harbor registries, deploy cluster networks, storage plug-ins, and etcd clusters, renew cluster certificates, and quickly upgrade clusters.

To obtain KubeKey, run the following command:

```
curl -sfL https://get-kk.kubesphere.io | VERSION=v3.0.2  sh -
```

Docker allows you to run a MySQL application with one command, and KubeKey allows you to run a Kubernetes cluster and KubeSphere with one command. Run the following command to install Kubernetes and KubeSphere:

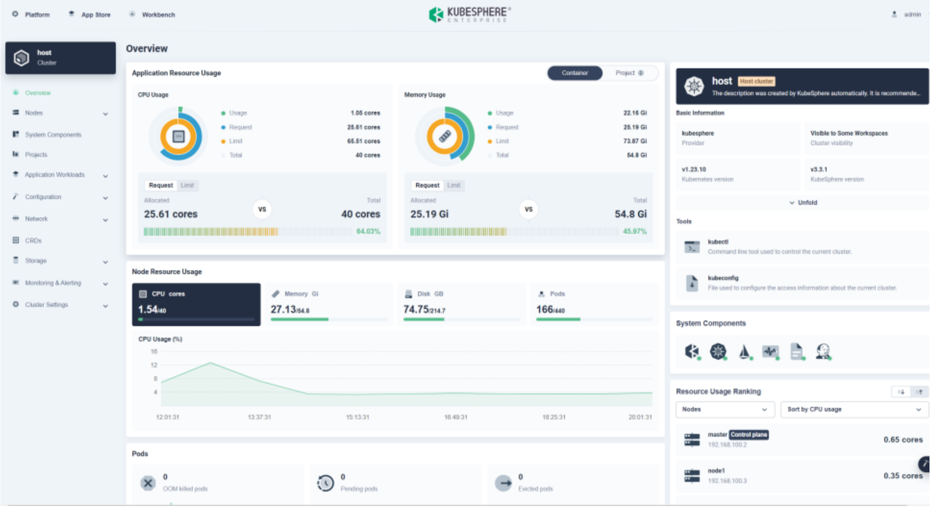
```
kk create cluster --with-kubernetes v1.23.8 --with-kubesphere v3.3.1
```

If you create a cluster with multiple nodes, run **kk create config** to create a configuration file. After you configure node information, run **kk create cluster -f config.yaml** to apply the file.

Sample code:

```
 hosts:
 - {name: master, address: 192.168.0.2, internalAddress: 192.168.0.2, user: ubuntu, password: Testing123}
 - {name: node1, address: 192.168.0.3, internalAddress: 192.168.0.3, user: ubuntu, password: Testing123}
 roleGroups:
   etcd:
   - master
   control-plane:
   - master
   worker:
   - node1
```

Wait until Kubernetes and KubeSphere are installed. This process might take about 20 minutes.



Install the storage plug-in for vSphere

vSphere CNS consists of the vCenter control plane and the storage plug-in for Kubernetes. The control plane is available for vCenter 6.7U3 and later, so you need only to install the CSI plug-in in the Kubernetes cluster backed by VMware vSphere.

Before you create a VM for Kubernetes, make sure the following conditions are met:

- ✓ VMware Tools is installed.
- ✓ disk.EnableUUID is set to true.
- ✓ The hardware version is 15 or later.
- ✓ A VMware Paravirtual SCSI controller is used for the VM.

Before you install the vSphere CSI driver, check its compatibility with vCenter and Kubernetes on the official website.

Install vsphere-cloud-controller-manager

Add labels to all nodes:

```
kubectl taint node k8s node.cloudprovider.kubernetes.io/uninitialized=true:NoSchedule
```

Download **vsphere-cloud-controller-manager.yaml**:

```
wget https://raw.githubusercontent.com/kubernetes/cloud-provider-vsphere/release-1.23/releases/v1.23/vsphere-cloud-controller-manager.yaml
```

Modify the vCenter-related configuration in Secret and ConfigMap:

```
apiVersion: v1
kind: Secret
metadata:
  name: vsphere-cloud-secret
  namespace: kube-system
stringData:
  10.0.0.1.username: "<ENTER_YOUR_VCENTER_USERNAME>"
  10.0.0.1.password: "<ENTER_YOUR_VCENTER_PASSWORD>"
```

```
apiVersion: v1
kind: ConfigMap
metadata:
  name: vsphere-cloud-config
  namespace: kube-system
```

```
data:
  vcenter:
    <your-vcenter-name-here>:
      server: 10.0.0.1
      user: <use-your-vcenter-user-here>
      password: <use-your-vcenter-password-here>
      datacenters:
        - Datacenter01
```

Run the following command to apply the file:

```
→ kubectl apply -f vsphere-con-ma.yaml
serviceaccount/cloud-controller-manager created
secret/vsphere-cloud-secret created
configmap/vsphere-cloud-config created
rolebinding.rbac.authorization.k8s.io/servicecatalog.k8s.io:apiserver-authentication-reader created
clusterrolebinding.rbac.authorization.k8s.io/system:cloud-controller-manager created
clusterrole.rbac.authorization.k8s.io/system:cloud-controller-manager created
daemonset.apps/vsphere-cloud-controller-manager created

→ kubectl get pods -A | grep vsphere
kube-system          vsphere-cloud-controller-manager-km68c        1/1     Running   0          3m4s
```

Create a namespace named **vmware-system-csi**:

```
→ kubectl create ns vmware-system-csi
namespace/vmware-system-csi created
```

Create a configuration file for the CSI driver:

```
cat /etc/kubernetes/csi-vsphere.conf
[Global]
cluster-id = "<cluster-id>"#Enter the cluster name

[VirtualCenter "<IP or FQDN>"] #Enter the IP address or FQDN of vCenter
insecure-flag = "<true or false>"#Specify the CA file and fingerprint if you set the value to false.
user = "<username>"
password = "<password>"
port = "<port>"
datacenters = "<datacenter1-path>"#Enter the path of the data center where the cluster nodes reside.
```

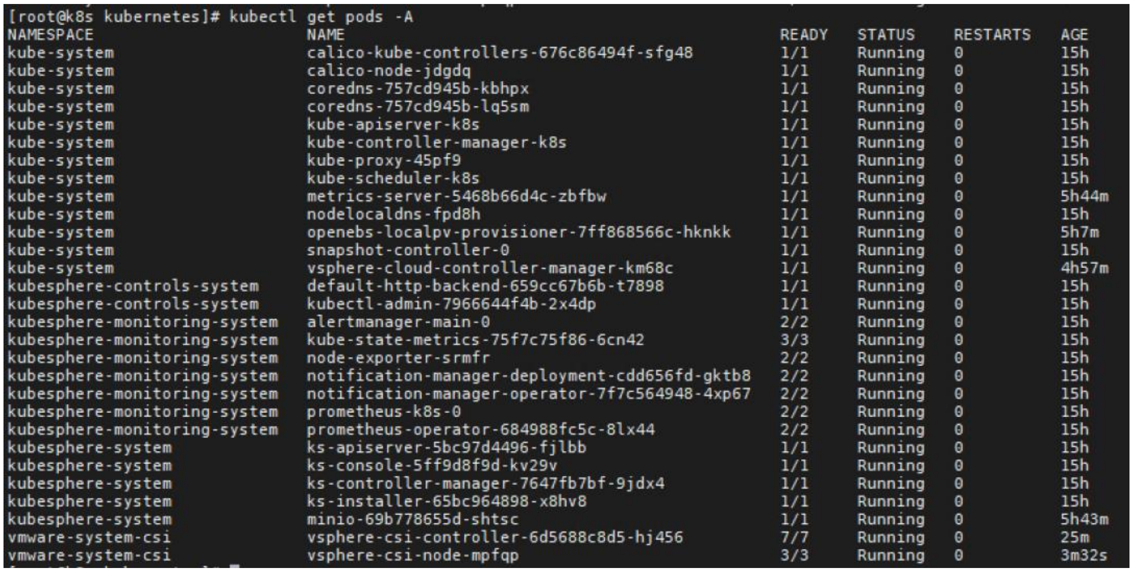
Generate a secret:

```
→ kubectl create secret generic vsphere-config-secret --from-file=csi-vsphere.conf --namespace=vmware-system-csi
```

Install **vsphere-csi-driver**:

```
→ kubectl apply -f https://raw.githubusercontent.com/kubernetes-sigs/vsphere-csi-driver/v2.6.0/manifests/vanilla/vsphere-csi-driver.yaml
```

Wait until all pods are running properly:



```
[root@k8s kubernetes]# kubectl get pods -A
NAMESPACE                      NAME                                             READY   STATUS    RESTARTS   AGE
kube-system                    calico-kube-controllers-676c86494f-sfg48         1/1     Running   0          15h
kube-system                    calico-node-jdgdq                                1/1     Running   0          15h
kube-system                    coredns-757cd945b-kbhpx                          1/1     Running   0          15h
kube-system                    coredns-757cd945b-lq5sm                          1/1     Running   0          15h
kube-system                    kube-apiserver-k8s                               1/1     Running   0          15h
kube-system                    kube-controller-manager-k8s                      1/1     Running   0          15h
kube-system                    kube-proxy-45pf9                                 1/1     Running   0          15h
kube-system                    kube-scheduler-k8s                               1/1     Running   0          15h
kube-system                    metrics-server-5468b66d4c-zbfbw                  1/1     Running   0          5h44m
kube-system                    nodelocaldns-fpd8h                               1/1     Running   0          15h
kube-system                    openebs-localpv-provisioner-7ff868566c-hknkk     1/1     Running   0          5h7m
kube-system                    snapshot-controller-0                            1/1     Running   0          15h
kube-system                    vsphere-cloud-controller-manager-km68c           1/1     Running   0          4h57m
kubesphere-controls-system     default-http-backend-659cc67b6b-t7898            1/1     Running   0          15h
kubesphere-controls-system     kubectl-admin-7966644f4b-2x4dp                   1/1     Running   0          15h
kubesphere-monitoring-system   alertmanager-main-0                              2/2     Running   0          15h
kubesphere-monitoring-system   kube-state-metrics-75f7c75f86-6cn42              3/3     Running   0          15h
kubesphere-monitoring-system   node-exporter-srmfr                              2/2     Running   0          15h
kubesphere-monitoring-system   notification-manager-deployment-cdd656fd-gktb8   2/2     Running   0          15h
kubesphere-monitoring-system   notification-manager-operator-7f7c564948-4xp67   2/2     Running   0          15h
kubesphere-monitoring-system   prometheus-k8s-0                                 2/2     Running   0          15h
kubesphere-monitoring-system   prometheus-operator-684988fc5c-8lx44             2/2     Running   0          15h
kubesphere-system              ks-apiserver-5bc97d4496-fjlbb                    1/1     Running   0          15h
kubesphere-system              ks-console-5ff9d8f9d-kv29v                       1/1     Running   0          15h
kubesphere-system              ks-controller-manager-7647fb7bf-9jdx4            1/1     Running   0          15h
kubesphere-system              ks-installer-65bc964898-x8hv8                    1/1     Running   0          15h
kubesphere-system              minio-69b778655d-shtsc                           1/1     Running   0          5h43m
vmware-system-csi              vsphere-csi-controller-6d5688c8d5-hj456          7/7     Running   0          25m
vmware-system-csi              vsphere-csi-node-mpfqp                           3/3     Running   0          3m32s
```

Query the status of csidriver and csinode:

```
→ kubectl get csidriver
NAME                  ATTACHREQUIRED   PODINFOONMOUNT   STORAGECAPACITY   TOKENREQUESTS   REQUIRESREPUBLISH
MODES        AGE
csi.vsphere.vmware.com   true            false           false           <unset>        false              Persistent   85m
→ kubectl get CSINODE
NAME   DRIVERS   AGE
k8s    1         16h
```

The vSphere CSI driver is installed. Then, configure the underlying storage and StorageClass.

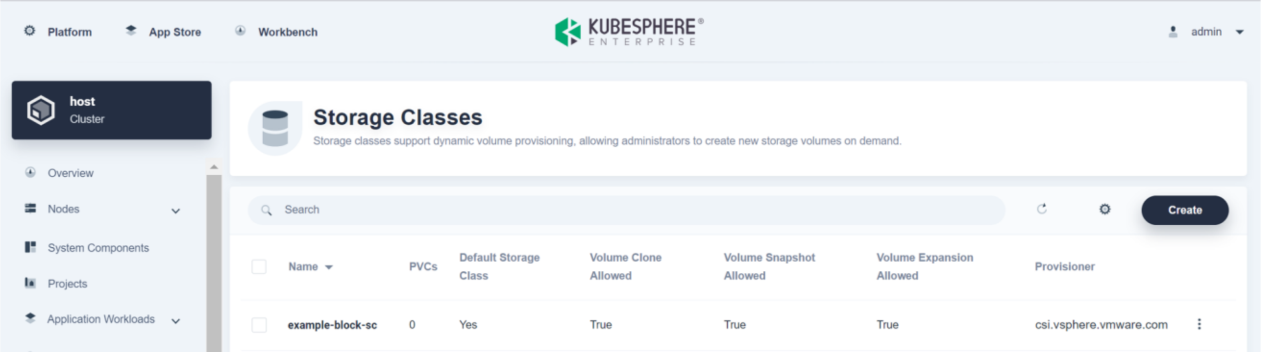
Create a StorageClass

CNS makes Kubernetes aware of how to dynamically provision PVs based on storage policies or StorageClasses.

Create a StorageClass:

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: example-block-sc
  annotations:
    storageclass.kubernetes.io/is-default-class: "true"
provisioner: csi.vsphere.vmware.com
parameters:
  storagepolicyname: "vSAN Default Storage Policy"  #The storage policy.
# datastoreurl: "ds:///vmfs/volumes/vsan:52cdfa80721ff516-ea1e993113acfc77/" #The data storage, for example, NFS mounted to ESXi,
FC/IP SAN, or local storage.
# csi.storage.k8s.io/fstype: "ext4" #The formatted file type.
```

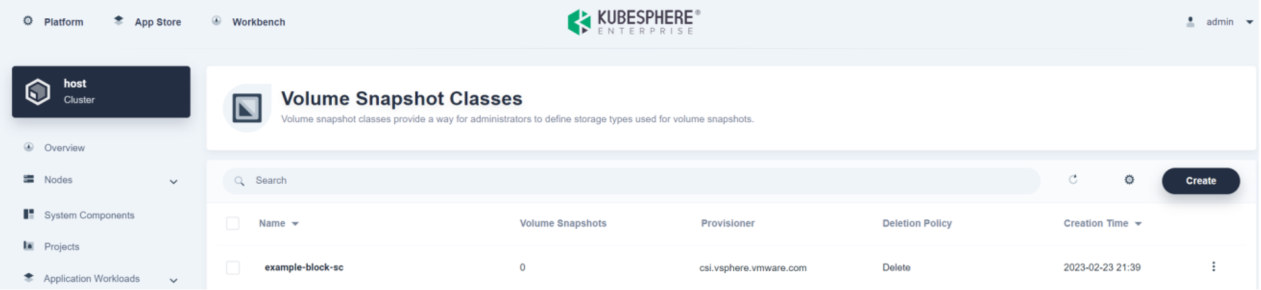
In the KubeSphere web console, check the storage class and snapshot class.



Storage Classes

Storage classes support dynamic volume provisioning, allowing administrators to create new storage volumes on demand.

Name	PVCs	Default Storage Class	Volume Clone Allowed	Volume Snapshot Allowed	Volume Expansion Allowed	Provisioner
example-block-sc	0	Yes	True	True	True	csi.vsphere.vmware.com



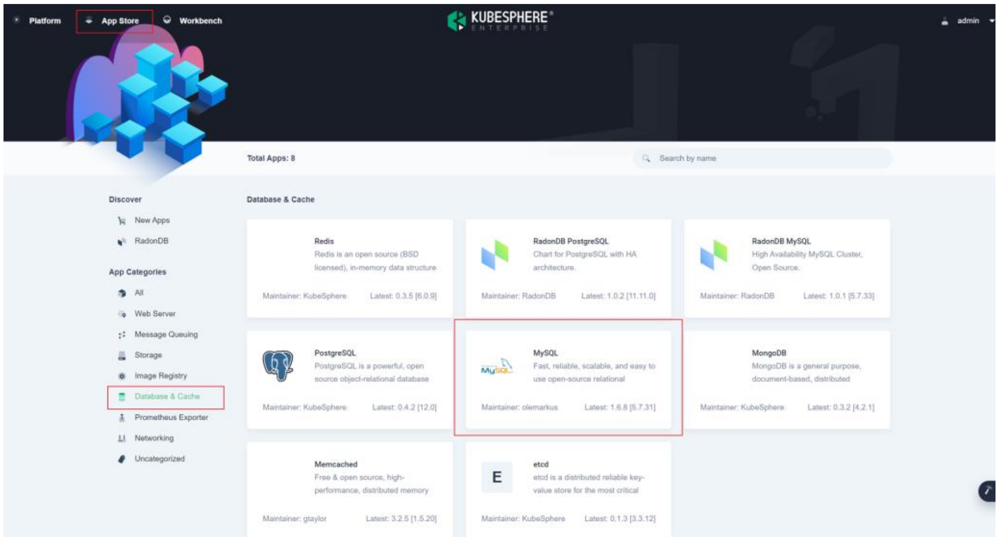
Volume Snapshot Classes

Volume snapshot classes provide a way for administrators to define storage types used for volume snapshots.

Name	Volume Snapshots	Provisioner	Deletion Policy	Creation Time
example-block-sc	0	csi.vsphere.vmware.com	Delete	2023-02-23 21:39

Run a Stateful Application

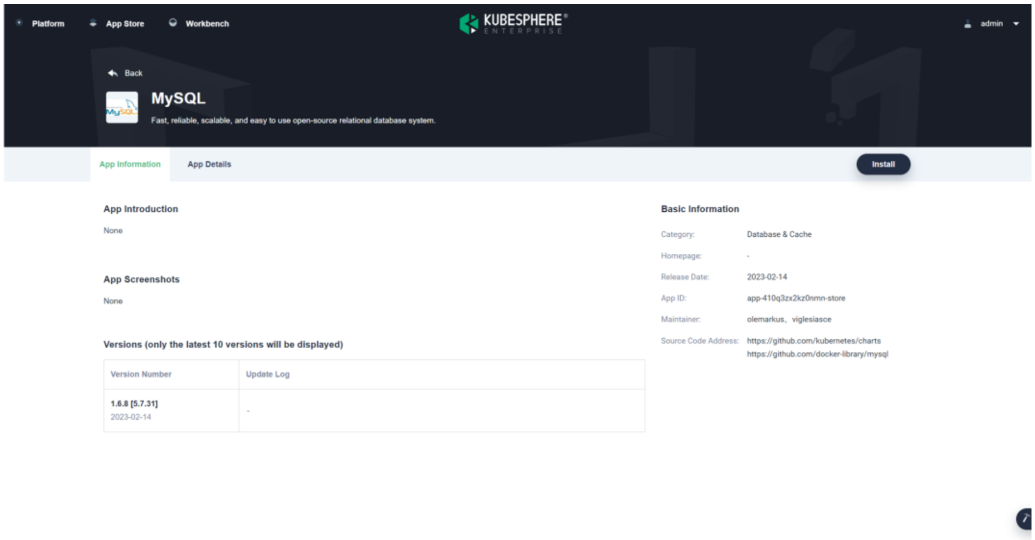
In the App Store of KubeSphere, deploy a MySQL application to check whether PVs work for vSphere.



Total Apps: 8

Database & Cache

App	Description	Maintainer	Latest
Redis	Redis is an open source (BSD licensed), in-memory data structure	Maintainer: KubeSphere	Latest: 0.3.5 [6.0.9]
RadonDB PostgreSQL	Chart for PostgreSQL with HA architecture.	Maintainer: RadonDB	Latest: 1.0.2 [11.11.0]
RadonDB MySQL	High Availability MySQL Cluster, Open Source.	Maintainer: RadonDB	Latest: 1.0.1 [5.7.33]
PostgreSQL	PostgreSQL is a powerful, open source object-relational database	Maintainer: KubeSphere	Latest: 0.4.2 [12.0]
MySQL	Fast, reliable, scalable, and easy to use open-source relational	Maintainer: olemarkus	Latest: 1.6.8 [5.7.31]
MongoDB	MongoDB is a general purpose, document-based, distributed	Maintainer: KubeSphere	Latest: 0.3.2 [4.2.1]
Memcached	Free & open source, high-performance, distributed memory	Maintainer: gtaylor	Latest: 3.2.5 [1.5.20]
etcd	etcd is a distributed reliable key-value store for the most critical	Maintainer: KubeSphere	Latest: 0.1.3 [3.3.12]



MySQL

Fast, reliable, scalable, and easy to use open-source relational database system.

App Introduction

None

App Screenshots

None

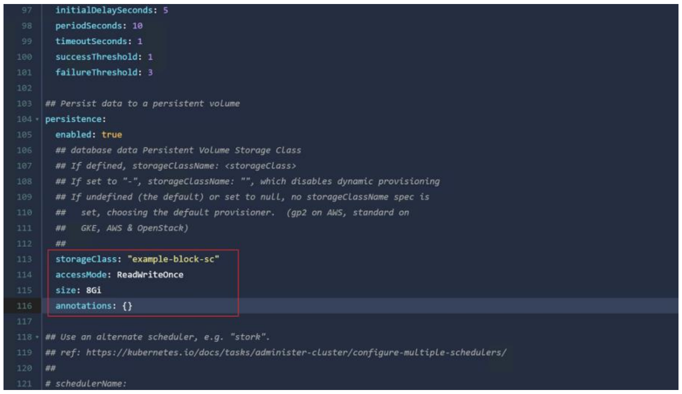
Versions (only the latest 10 versions will be displayed)

Version Number	Update Log
1.6.8 [5.7.31] 2023-02-14	-

Basic Information

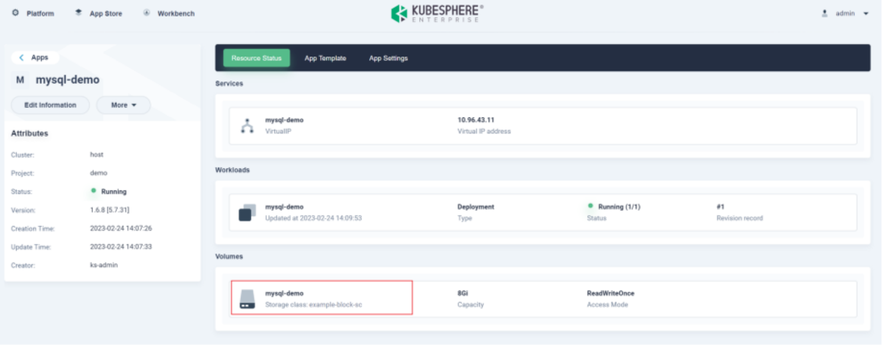
Category:	Database & Cache
Homepage:	-
Release Date:	2023-02-14
App ID:	app-410q3zx2kz0nmn-store
Maintainer:	olemarkus、viglesiasce
Source Code Address:	https://github.com/kubernetes/charts https://github.com/docker-library/mysql

Set **storageClass** to **example-block-sc**, which was created based on the vSphere CSI driver:



```
  initialDelaySeconds: 5
  periodSeconds: 10
  timeoutSeconds: 1
  successThreshold: 1
  failureThreshold: 3

## Persist data to a persistent volume
persistence:
  enabled: true
  ## database data Persistent Volume Storage Class
  ## If defined, storageClassName: <storageClass>
  ## If set to "-", storageClassName: "", which disables dynamic provisioning
  ## If undefined (the default) or set to null, no storageClassName spec is
  ##   set, choosing the default provisioner.  (gp2 on AWS, standard on
  ##   GKE, AWS & OpenStack)
  ##
  storageClass: "example-block-sc"
  accessMode: ReadWriteOnce
  size: 8Gi
  annotations: {}

## Use an alternate scheduler, e.g. "stork".
## ref: https://kubernetes.io/docs/tasks/administer-cluster/configure-multiple-schedulers/
##
# schedulerName:
```

The following figure shows that the application with a PVC is created.



M mysql-demo

Attributes

Cluster: host
Project: demo
Status: Running
Version: 1.6.8 [5.7.31]
Creation Time: 2023-02-24 14:07:26
Update Time: 2023-02-24 14:07:33
Creator: ks-admin

Services

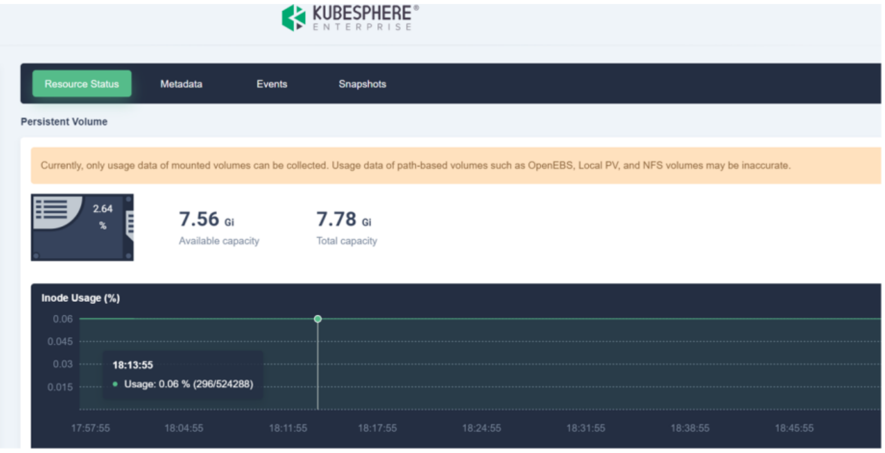
mysql-demo VirtualIP | 10.96.43.11 Virtual IP address

Workloads

mysql-demo Updated at 2023-02-24 14:09:53 | Deployment Type | Running (1/1) Status | #1 Revision record

Volumes

mysql-demo Storage class: example-block-sc | 8Gi Capacity | ReadWriteOnce Access Mode



Persistent Volume

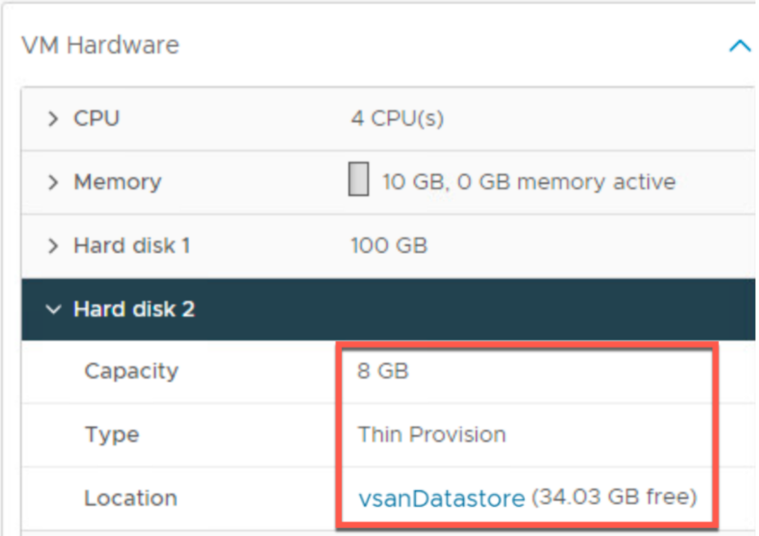
Currently, only usage data of mounted volumes can be collected. Usage data of path-based volumes such as OpenEBS, Local PV, and NFS volumes may be inaccurate.

7.56 Gi Available capacity

7.78 Gi Total capacity

18:13:55
Usage: 0.06 % (296/524288)

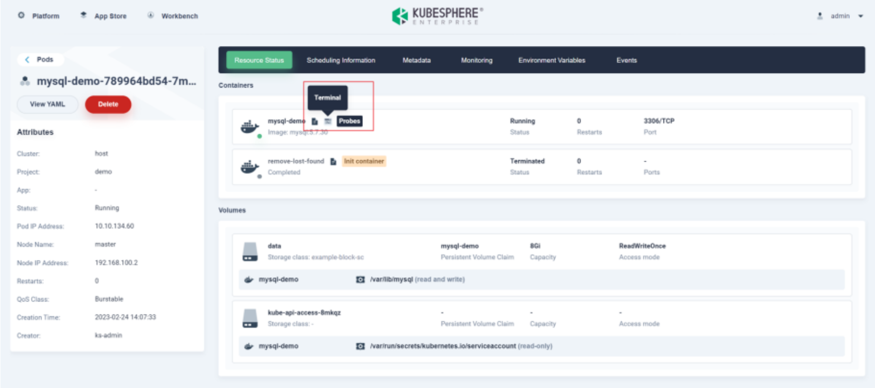
Go to vCenter, and you can view that an 8GB disk has been mounted to the VM. The capacity is the same as that defined in the PVC, and the backend storage is the same as that defined in the StorageClass.



VM Hardware

CPU	4 CPU(s)
Memory	10 GB, 0 GB memory active
Hard disk 1	100 GB
Hard disk 2	
Capacity	8 GB
Type	Thin Provision
Location	vsanDatastore (34.03 GB free)

Open Terminal to connect to the MySQL database:



Attributes

Cluster:	host
Project:	demo
App:	-
Status:	Running
Pod IP Address:	10.10.134.60
Node Name:	master
Node IP Address:	192.168.100.2
Restarts:	0
QoS Class:	Burstable
Creation Time:	2023-02-24 14:07:33
Creator:	ks-admin

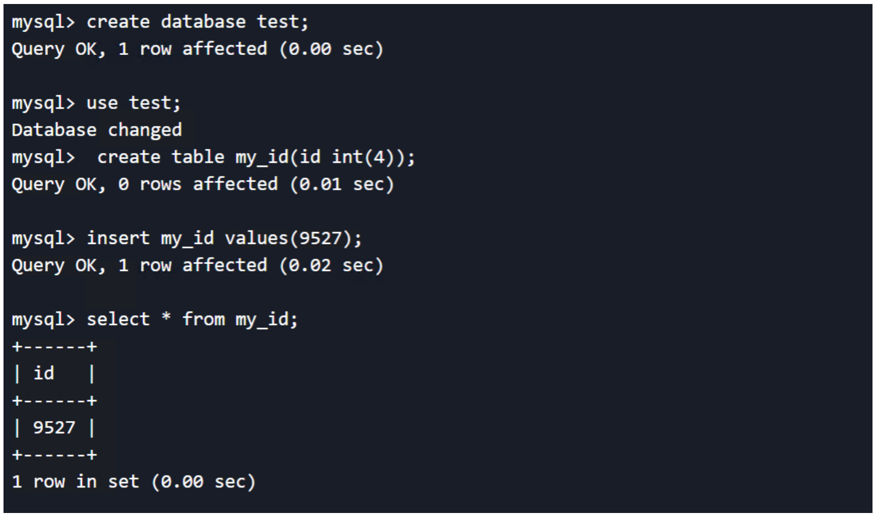
Containers

mysql-demo Image: mysql:5.7.30	Running Status	0 Restarts	3306/TCP Port
remove-lost-found Completed	Terminated Status	0 Restarts	- Ports

Volumes

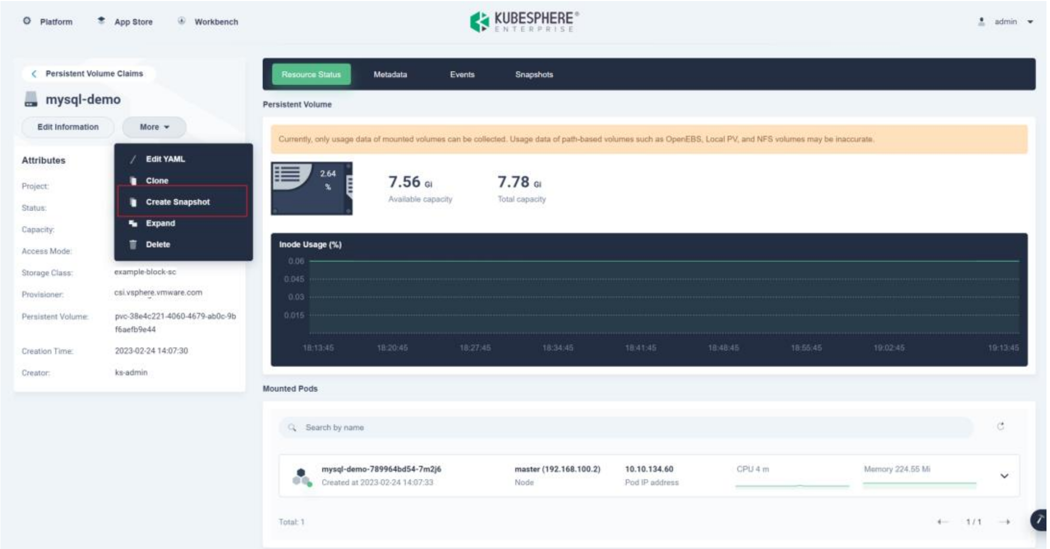
data Storage class: example-block-sc	mysql-demo Persistent Volume Claim	8Gi Capacity	ReadWriteOnce Access mode
mysql-demo	/var/lib/mysql (read and write)		
kube-api-access-8mkqz Storage class: -	- Persistent Volume Claim	- Capacity	- Access mode
mysql-demo	/var/run/secrets/kubernetes.io/serviceaccount (read-only)		

Create a table and insert example values:



```
mysql> create database test;
Query OK, 1 row affected (0.00 sec)

mysql> use test;
Database changed
mysql>  create table my_id(id int(4));
Query OK, 0 rows affected (0.01 sec)

mysql> insert my_id values(9527);
Query OK, 1 row affected (0.02 sec)

mysql> select * from my_id;
+------+
| id   |
+------+
| 9527 |
+------+
1 row in set (0.00 sec)
```

Create a snapshot for the PVC:



Attributes

Project:

Status:

Capacity:

Access Mode:

Storage Class: example-block-sc

Provisioner: csi.vsphere.vmware.com

Persistent Volume: pvc-38e4c221-4060-4679-ab0c-9bf6aefb9e44

Creation Time: 2023-02-24 14:07:30

Creator: ks-admin

Persistent Volume

Currently, only usage data of mounted volumes can be collected. Usage data of path-based volumes such as OpenEBS, Local PV, and NFS volumes may be inaccurate.

2.64 %

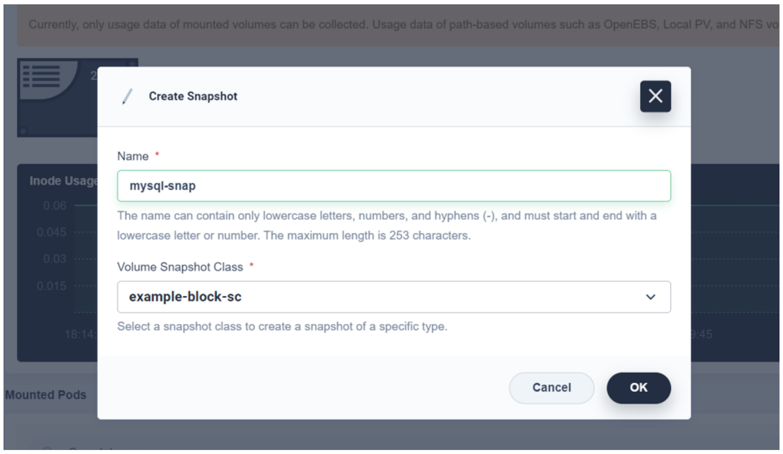
7.56 Gi Available capacity

7.78 Gi Total capacity

Mounted Pods

mysql-demo-789964bd54-7m2j6 Created at 2023-02-24 14:07:33 | master (192.168.100.2) Node | 10.10.134.60 Pod IP address | CPU 4 m | Memory 224.55 Mi

Total: 1



Currently, only usage data of mounted volumes can be collected. Usage data of path-based volumes such as OpenEBS, Local PV, and NFS vo

Create Snapshot

Name *

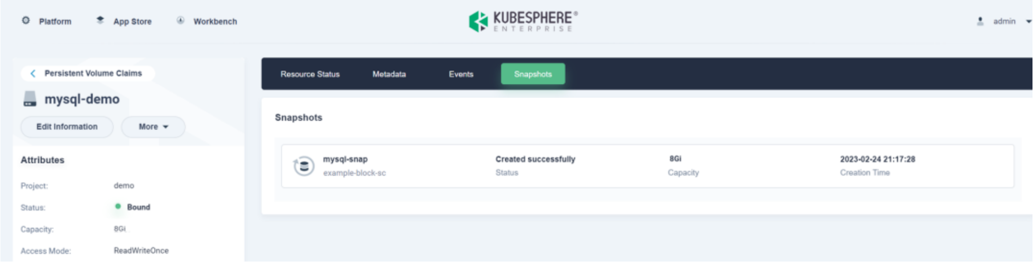
mysql-snap

The name can contain only lowercase letters, numbers, and hyphens (-), and must start and end with a lowercase letter or number. The maximum length is 253 characters.

Volume Snapshot Class *

example-block-sc

Select a snapshot class to create a snapshot of a specific type.



Attributes

Project: demo

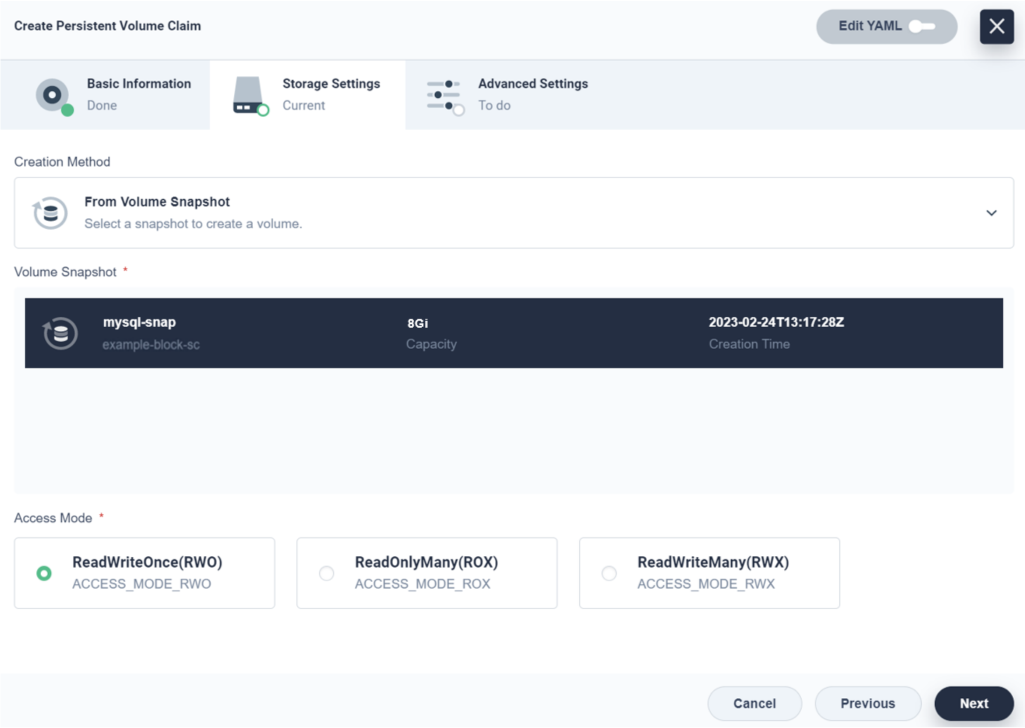
Status: ● Bound

Capacity: 8Gi

Access Mode: ReadWriteOnce

Snapshots

mysql-snap example-block-sc | Created successfully Status | 8Gi Capacity | 2023-02-24 21:17:28 Creation Time



Creation Method

From Volume Snapshot
Select a snapshot to create a volume.

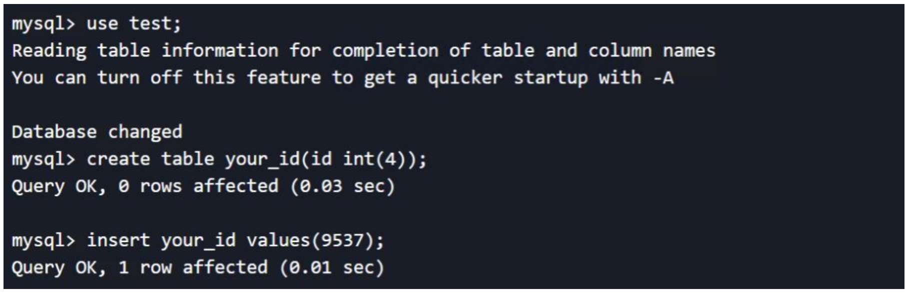
Volume Snapshot *

mysql-snap example-block-sc	8Gi Capacity	2023-02-24T13:17:28Z Creation Time

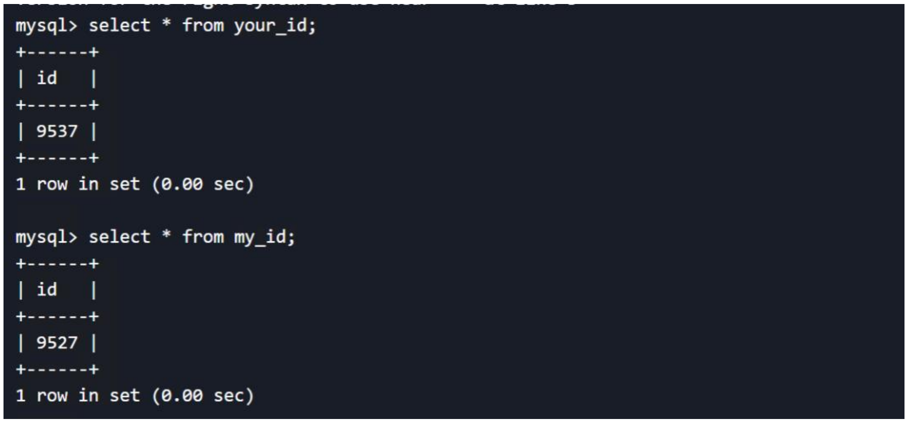
Access Mode *

● ReadWriteOnce(RWO) ACCESS_MODE_RWO
○ ReadOnlyMany(ROX) ACCESS_MODE_ROX
○ ReadWriteMany(RWX) ACCESS_MODE_RWX

Insert data to the MySQL database:

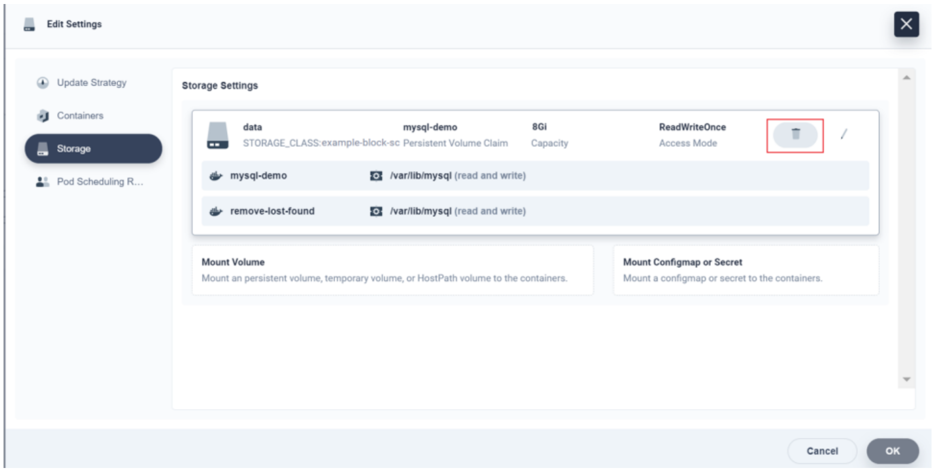


```
mysql> use test;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> create table your_id(id int(4));
Query OK, 0 rows affected (0.03 sec)

mysql> insert your_id values(9537);
Query OK, 1 row affected (0.01 sec)
```



```
mysql> select * from your_id;
+------+
| id   |
+------+
| 9537 |
+------+
1 row in set (0.00 sec)

mysql> select * from my_id;
+------+
| id   |
+------+
| 9527 |
+------+
1 row in set (0.00 sec)
```

Delete the PVC that is bound. Then, verify that the PVC cannot be found in the MySQL database.



Storage Settings

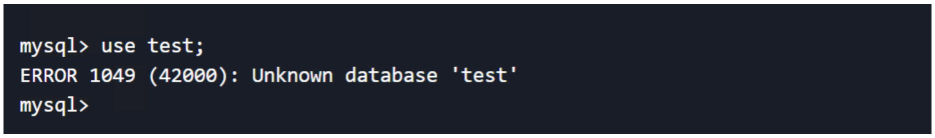
data STORAGE_CLASS:example-block-sc | mysql-demo Persistent Volume Claim | 8Gi Capacity | ReadWriteOnce Access Mode

mysql-demo /var/lib/mysql (read and write)

remove-lost-found /var/lib/mysql (read and write)

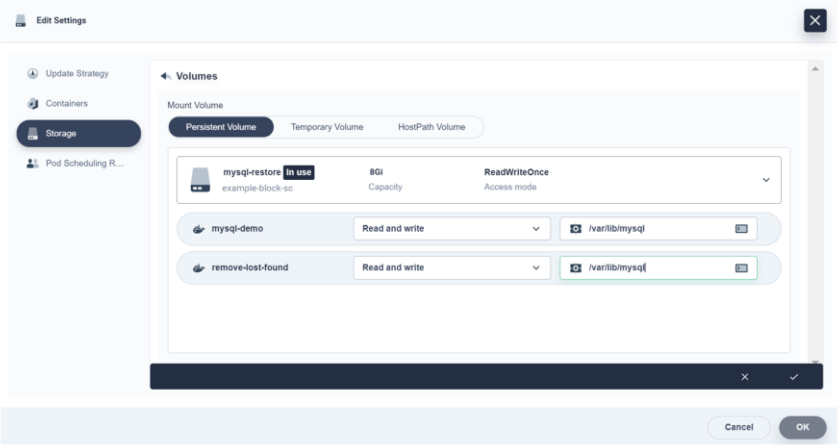
Mount Volume
Mount an persistent volume, temporary volume, or HostPath volume to the containers.

Mount Configmap or Secret
Mount a configmap or secret to the containers.



```
mysql> use test;
ERROR 1049 (42000): Unknown database 'test'
mysql>
```

Use the snapshot created for the PVC, and connect to the MySQL database to query data:

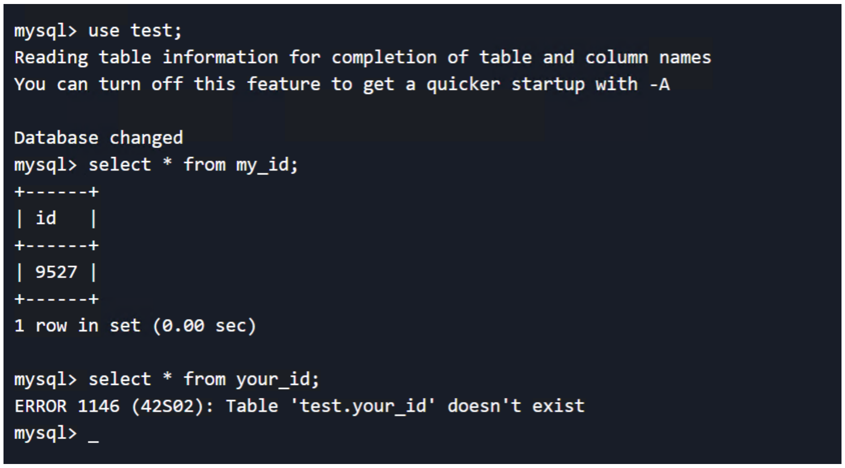


Mount Volume

mysql-restore In use example-block-sc | 8Gi Capacity | ReadWriteOnce Access mode

mysql-demo | Read and write | /var/lib/mysql

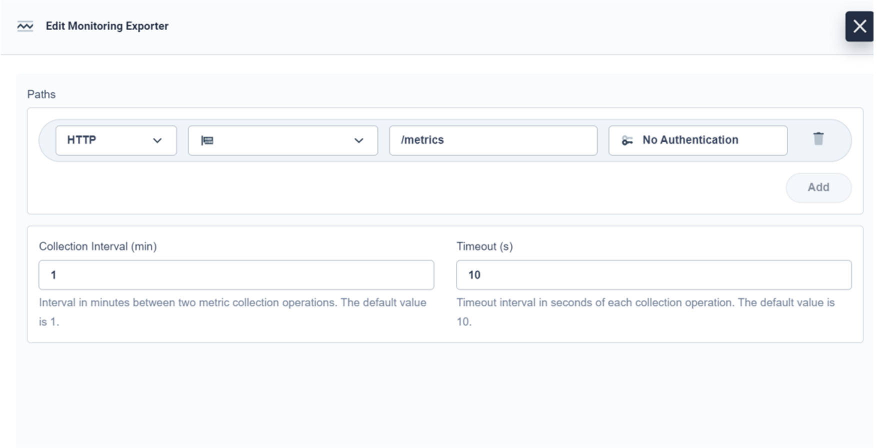
remove-lost-found | Read and write | /var/lib/mysql



```
mysql> use test;
Reading table information for completion of table and column names
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> select * from my_id;
+------+
| id   |
+------+
| 9527 |
+------+
1 row in set (0.00 sec)

mysql> select * from your_id;
ERROR 1146 (42S02): Table 'test.your_id' doesn't exist
mysql> _
```

Monitoring

The vSphere CSI driver provides Prometheus-based monitoring metrics, including metrics for CNS components and storage. In the KubeSphere web console, you can use the monitoring exporter to create ServiceMontior:



Edit Monitoring Exporter

Paths

HTTP | | /metrics | No Authentication

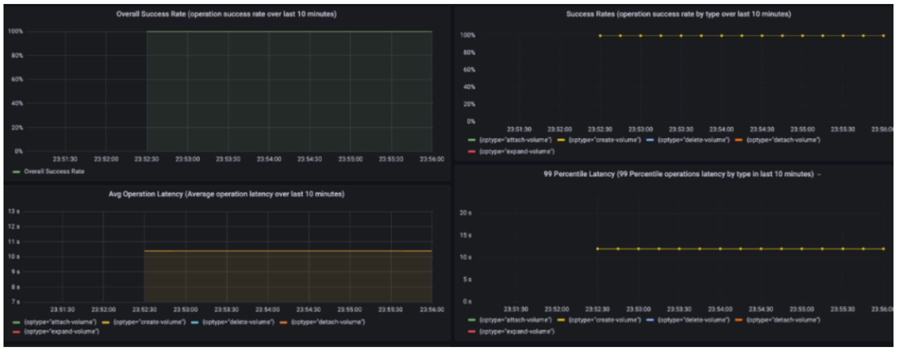
Collection Interval (min): 1

Interval in minutes between two metric collection operations. The default value is 1.

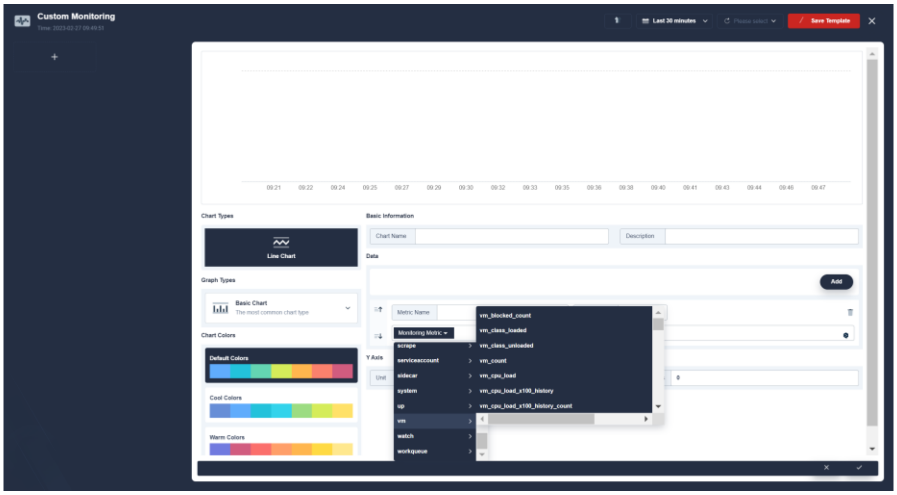
Timeout (s): 10

Timeout interval in seconds of each collection operation. The default value is 10.

vSphere provides Grafana monitoring templates available for use:



In the KubeSphere web console, you can configure custom dashboards based on monitoring metrics, such as the number of times that block storage is successfully created.



Administrators can also view CNS monitoring data in vCenter.

Note: This section introduces the Cloud Native Storage (CNS) features in vSAN for standard or 'vanilla' Kubernetes clusters. See the link below for more information on CNS for vSphere:

https://blogs.vmware.com/virtualblocks/2019/08/14/introducing-cloud-native-storage-for-vsphere/

Integrate KubeSphere with Antrea

Deploy Antrea and antctl

```
kubectl apply -f https://raw.githubusercontent.com/antrea-io/antrea/main/build/yamls/antrea.yml
kubectl get pod -n kube-system | grep antrea
antrea-agent-4gbn8                          2/2     Running     36d
antrea-agent-n8v2k                          2/2     Running     36d
antrea-agent-wznw6                          2/2     Running     36d
antrea-controller-879cc648-jgp5x            1/1     Running     40d
```

Use KubeKey to deploy a Kubernetes cluster and Antrea at ease

Create a cluster configuration file:

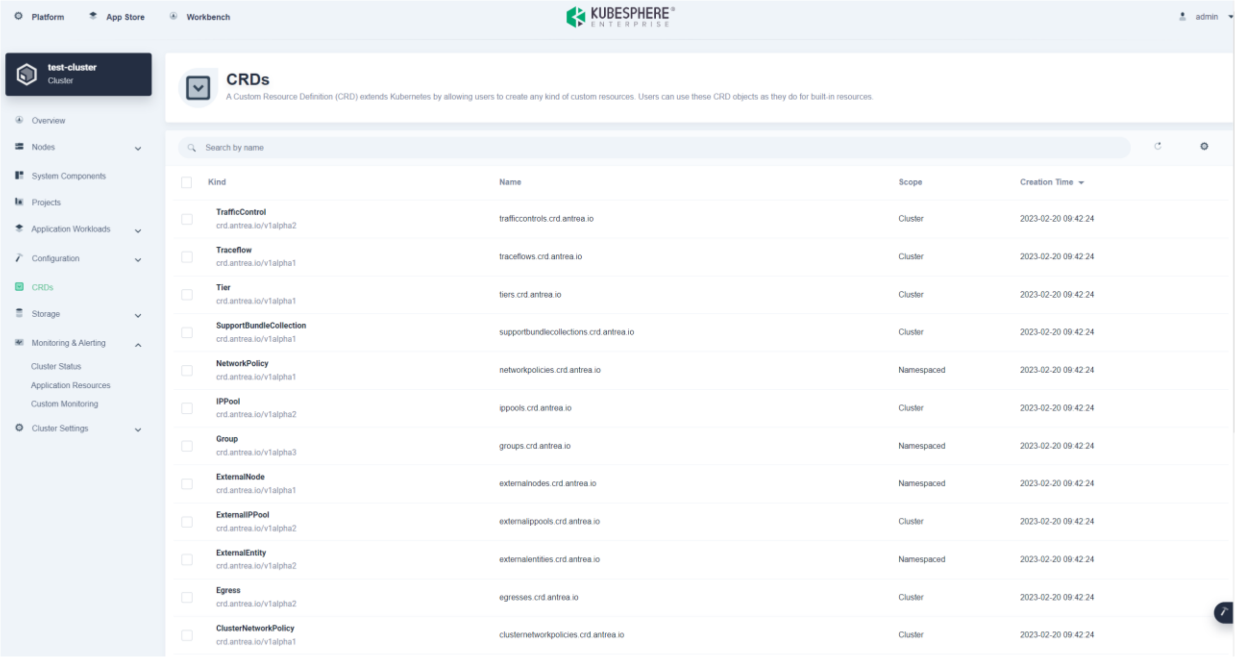
```
./kk create config -f config.yaml

vi config.yaml

apiVersion: kubekey.kubesphere.io/v1alpha2

kind: Cluster

metadata:

  name: example

spec:

  hosts:

  - {name: node1, address: 172.16.0.2, internalAddress: 172.16.0.2, privateKeyPath: "~/.ssh/id_rsa"}

  - {name: node2, address: 172.16.0.3, internalAddress: 172.16.0.3, privateKeyPath: "~/.ssh/id_rsa"}

  - {name: node3, address: 172.16.0.4, internalAddress: 172.16.0.4, privateKeyPath: "~/.ssh/id_rsa"}
```

```
...
  network:
    plugin: none
    kubePodsCIDR: 10.16.0.0/16
    kubeServiceCIDR: 10.96.0.0/16
...
addons:
  - name: antrea
    namespace: kube-system
    sources:
      chart:
        name: antrea
        repo: https://charts.antrea.io
```

Run the **kk create cluster -f config.yaml** command to deploy the Kubernetes cluster and Antrea.

KubeKey supports using yaml files or Helm charts to install Antrea as an add-on. Support for network plugin is in assessment and might be implemented in future versions.

You can view the CRD resources related to Antrea in the KubeSphere web console.

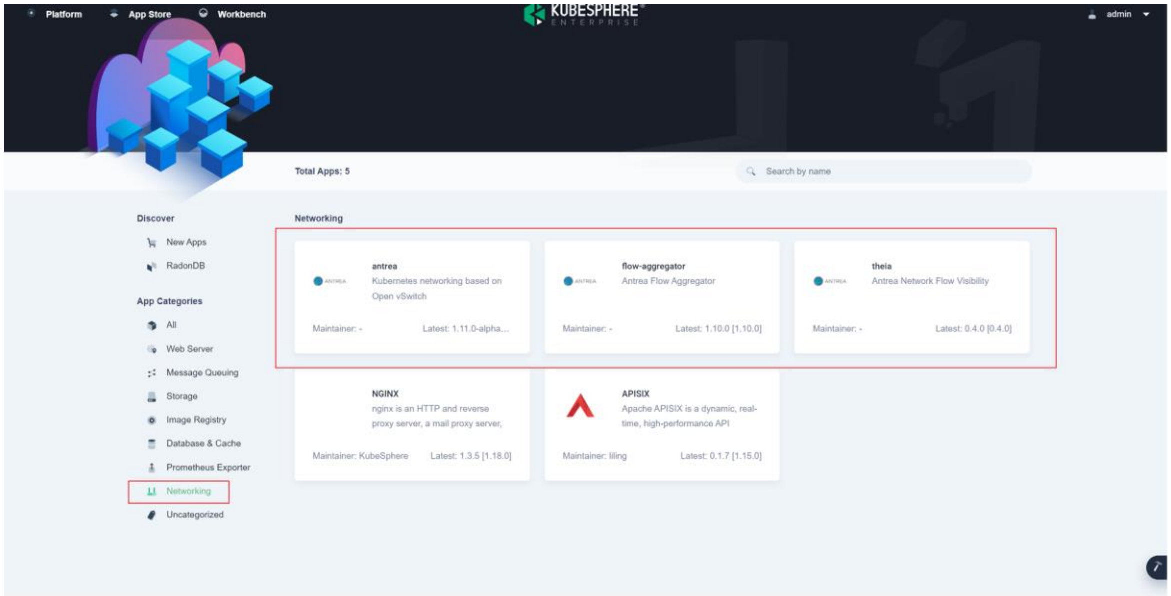


CRDs

A Custom Resource Definition (CRD) extends Kubernetes by allowing users to create any kind of custom resources. Users can use these CRD objects as they do for built-in resources.

Kind	Name	Scope	Creation Time
TrafficControl crd.antrea.io/v1alpha2	trafficcontrols.crd.antrea.io	Cluster	2023-02-20 09:42:24
Traceflow crd.antrea.io/v1alpha1	traceflows.crd.antrea.io	Cluster	2023-02-20 09:42:24
Tier crd.antrea.io/v1alpha1	tiers.crd.antrea.io	Cluster	2023-02-20 09:42:24
SupportBundleCollection crd.antrea.io/v1alpha1	supportbundlecollections.crd.antrea.io	Cluster	2023-02-20 09:42:24
NetworkPolicy crd.antrea.io/v1alpha1	networkpolicies.crd.antrea.io	Namespaced	2023-02-20 09:42:24
IPPool crd.antrea.io/v1alpha2	ippools.crd.antrea.io	Cluster	2023-02-20 09:42:24
Group crd.antrea.io/v1alpha3	groups.crd.antrea.io	Namespaced	2023-02-20 09:42:24
ExternalNode crd.antrea.io/v1alpha1	externalnodes.crd.antrea.io	Namespaced	2023-02-20 09:42:24
ExternalIPPool crd.antrea.io/v1alpha2	externalippools.crd.antrea.io	Cluster	2023-02-20 09:42:24
ExternalEntity crd.antrea.io/v1alpha2	externalentities.crd.antrea.io	Namespaced	2023-02-20 09:42:24
Egress crd.antrea.io/v1alpha2	egresses.crd.antrea.io	Cluster	2023-02-20 09:42:24
ClusterNetworkPolicy crd.antrea.io/v1alpha1	clusternetworkpolicies.crd.antrea.io	Cluster	2023-02-20 09:42:24

Three projects for Antrea, Flow Aggregator, and Theia will be rolled out on KubeSphere App Store. You can directly install these applications from the App Store in this way.



Total Apps: 5

Networking

antrea
Kubernetes networking based on Open vSwitch
Maintainer: - Latest: 1.11.0-alpha...

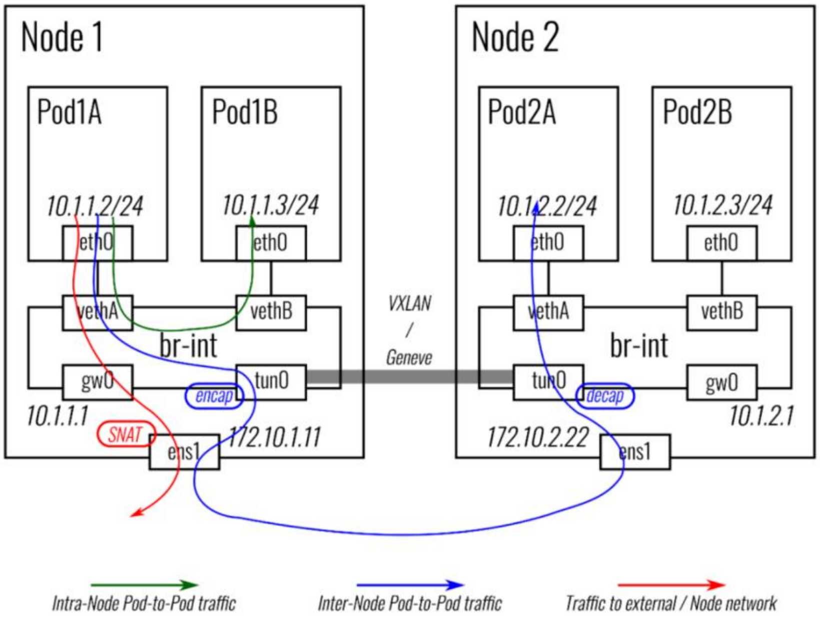
flow-aggregator
Antrea Flow Aggregator
Maintainer: - Latest: 1.10.0 [1.10.0]

theia
Antrea Network Flow Visibility
Maintainer: - Latest: 0.4.0 [0.4.0]

NGINX
nginx is an HTTP and reverse proxy server, a mail proxy server,
Maintainer: KubeSphere Latest: 1.3.5 [1.18.0]

APISIX
Apache APISIX is a dynamic, real-time, high-performance API
Maintainer: liling Latest: 0.1.7 [1.15.0]

Traffic walk



The image is cited from the Antrea official website, which can help you familiarize yourself with Antrea's communication mode.

- Intra-node traffic

Packets between two local pods will be forwarded by the OVS bridge directly.

- Inter-node traffic

Packets to a pod on another node will be first forwarded to the **antrea-tun0** port, encapsulated, and sent to the destination node through the tunnel; then they will be decapsulated, injected through the antrea-tun0 port to the OVS bridge, and finally forwarded to the destination pod.

- Pod to external traffic

Packets sent to an external IP or the Nodes' network will be forwarded to the **antrea-gw0** port (as it is the gateway of the local Pod subnet), and will be routed (based on routes configured on the Node) to the appropriate network interface of the Node (for example, a physical network interface for a bare metal node) and sent out to the Node network from there. Antrea Agent creates an iptables (MASQUERADE) rule to perform SNAT on the packets from Pods, so their source IP will be rewritten to the Node's IP before going out.

Security

Antrea ensures security from the following aspects:

- Control plane security

All API communication between Antrea control plane components is encrypted with TLS. You can generate the required certificate manually, or manage certificates by using cert-manager.

- Traffic encryption

Antrea supports encrypting tunnel traffic across nodes with IPsec ESP or WireGuard, to meet audit and security compliance for enterprises.

- Network policies

Network policies are native Kubernetes resources that include a whitelist of allowed egress rules. However, Kubernetes-native network policies do not support node-level control, events and logging, display of traffic rejection, cluster-level control, fine-grained action rules, and policy priorities.

Compared with Kubernetes-native network policies, Antrea abstracts a lot of network details and offers CRDs such as Tier, ClusterGroup, Group, ClusterNetworkPolicy, and NetworkPolicy. It defines policy priorities, groups resources by cluster or namespace, and supports cluster network policies. In contrast, Kubernetes-native network policies are similar to firewall rules for O&M personnel.

Antrea provides Tiers with specific priorities. You can attach a network policy to any Tier.

Emergency > SecurityOps > NetworkOps > Platform > Application > K8s NetworkPolicy > Baseline

In a ClusterGroup resource, you can specify labels, CIDR blocks, services, and member groups as needed.

Sample code:

```
apiVersion: crd.antrea.io/v1alpha3
kind: ClusterGroup
metadata:
  name: demo-cluster-group
spec:
  podSelector:
    matchLabels:
      role: db
  namespaceSelector:
    matchLabels:
      env: prod
  ipBlocks:
    - cidr: 10.0.10.0/24
  serviceReference:
    name: test-service
    namespace: default
  childGroups: [test-cg-sel, test-cg-ip-blocks, test-cg-svc-ref]
```

After the resource is defined, you can configure a network policy. In the following example, the policy priority is set to 5, and the policy applies to the resource group named **test-db**. Each ClusterNetworkPolicy might consist of zero or more ordered set of ingress or egress rules. Each rule, depending on the **action** field of the value, allows, drops, rejects, or passes traffic. Each policy can work together with multiple selectors and resources, such as ports, ipBlock, services, FQDN, ICMP, IGMP, and HTTP. You can enable logging for resources. Logs are stored in /var/log/antrea/networkpolicy/np.log, and you can use tools such as Filebeat to collect and analyze logs. The design of NetworkPolicy is similar to that of ClusterNetworkPolicy. This way, you can implement secure network policies at Layer 3 to Layer 7.

Sample code:

```
apiVersion: crd.antrea.io/v1alpha1
kind: ClusterNetworkPolicy
metadata:
  name: acnp-demo
spec:
  priority: 5
  tier: securityops
  appliedTo:
    - group: "test-db"
  ingress:
    - action: Allow #[“Allow”, “Drop”, “Reject”, “Pass”]
      from:
        - podSelector:#[“podSelector”, “namespaceSelector”, “nodeSelector”]
            matchLabels:
              role: frontend
      ports:
        - protocol: TCP
          port: 8080
          endPort: 9000
      name: AllowFromFrontend
      enableLogging: false
  egress:
    - action: Drop
      to:
        - ipBlock:
            cidr: 10.0.10.0/24
      ports:
        - protocol: TCP
          port: 5978
#      toServices:
#        - name: svcName
#          namespace: svcNamespace
#      protocols:
#        - icmp:
#            icmpType: 8
#            icmpCode: 0
#      l7Protocols:
#        - http:
#            path: "/api/v2/*"
#            host: "foo.bar.com"
#            method: "GET"

      name: DropToThirdParty
      enableLogging: true
```

IPAM

Antrea provides two types of IPAM capabilities, that is, NodeIPAM and AntreaIPAM.

NodeIPAM is a Kubernetes component, which manages IP address pool allocation per each node, when the node initializes. Antrea NodeIPAM controller can be executed in scenarios where the NodeIPAMController is disabled in kube-controller-manager.

When a Pod's IP is allocated from an IP Pool, the traffic from the Pod to Pods on another Node or from the Pod to external networks will be sent to the underlay network through the Node's transport network interface and will be forwarded or routed by the underlay network.

In real life practice, if you use underlay networks, use AntreaIPAM to allocate IP addresses and connect to physical networks. If you use overlay networks, use Multus for the second NIC and AntreaIPAM for VLAN connections.

Sample code:

```
{
  "cniVersion": "0.3.0",
  "name": "ipv4-net-1",
  "type": "macvlan",
  "master": "eth0",
  "mode": "bridge",
  "ipam": {
    "type": "antrea",
    "ippools": [ "ipv4-pool-1" ]
  }
}

kind: Pod
metadata:
 annotations:
  ipam.antrea.io/ippools: 'pod-ip-pool1'
```

Egress

When pods access critical applications such as databases outside the cluster, security audits are required. In this scenario, you can adopt egress gateways. Egress is a CRD API that manages external access from the Pods in a cluster. It supports specifying which egress (SNAT) IP the traffic from the selected Pods to the external network should use.

Sample code:

```
apiVersion: crd.antrea.io/v1alpha2
kind: Egress
metadata:
 name: egress-prod-web
spec:
 appliedTo:
  namespaceSelector:
   matchLabels:
    env: prod
  podSelector:
   matchLabels:
    role: web
 egressIP: 10.10.0.8
 externalIPPool: prod-external-ip-pool
```

You can use egressIP and externalIPPool to specify Egress resources. The specified IP address must be accessible over physical networks. High availability is provided automatically when the egressIP is allocated from an externalIPPool; for example, when the externalIPPool is specified. If the Node hosting the egressIP fails, another Node will be selected (from among the remaining Nodes selected by the nodeSelector of the externalIPPool) as the new egress Node of this Egress. It will take over the IP and send layer 2 advertisement (for example, Gratuitous ARP for IPv4) to notify the other hosts and routers on the network that the MAC address associated with the IP has changed.

NodePortLocal

In Kubernetes, you can use NodePort or LoadBalancer to expose services. LoadBalancers such as F5, Server Load Balancer (SLB), and OpenELB can help you create a service like ClusterIP and open a port on each node.NodePorts expose services on static ports that range from 30000 to 32767. However, the design of kube-proxy can cause issues in various scenarios, such as traffic distribution, session persistence, health check on external loads, and numerous ports occupied.

NodePortLocal creates mappings between *NodeIP:Port* and *PodIP:Port*, so each pod can access each other by using NodeIP and NodePort. This feature is suitable for the traditional business that transforms to cloud native, for example, clusters where microservices reside and interconnection between registries and clusters.

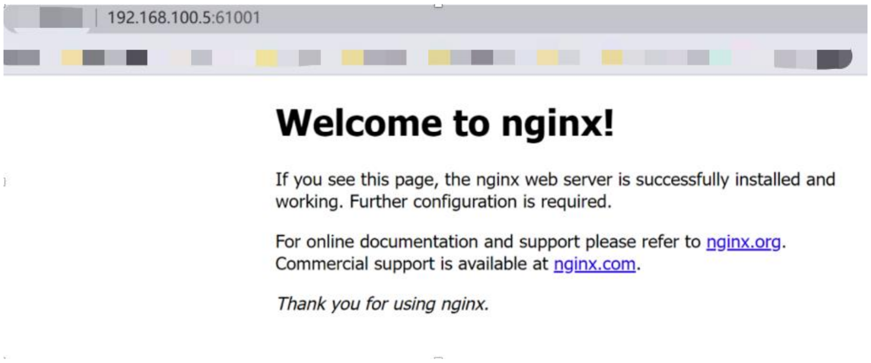
Add service annotations:

```
apiVersion: v1
kind: Service
metadata:
  name: nginx
  annotations:
    nodeportlocal.antrea.io/enabled: "true"
spec:
  ports:
  - name: web
    port: 80
    protocol: TCP
    targetPort: 80
  selector:
    app: nginx
```

View pod annotations:

```
[root@ks1-master ~]# kubectl get pods nginx-85b98978db-fwbmw -oyaml
apiVersion: v1
kind: Pod
metadata:
  annotations:
    nodeportlocal.antrea.io: '[{"podPort":80,"nodeIP":"192.168.100.5","nodePort":61001,"protocol":"tcp","protocols":["tcp"]}]'
  creationTimestamp: "2022-11-28T08:53:35Z"
  generateName: nginx-85b98978db-
  labels:
    app: nginx
    pod-template-hash: 85b98978db
  name: nginx-85b98978db-fwbmw
  namespace: default
```

Access the service in the *NodeIP:NodePort* format:



Welcome to nginx!

If you see this page, the nginx web server is successfully installed and working. Further configuration is required.

For online documentation and support please refer to nginx.org.
Commercial support is available at nginx.com.

Thank you for using nginx.

Query NAT rules:

```
conntrack -L -j  |grep 10.10.1.24
tcp      6 431970 ESTABLISHED src=192.168.255.254 dst=192.168.100.5 sport=7149 dport=61001 packets=2 bytes=92 src=10.10.1.24
dst=192.168.255.254 sport=80 dport=7149 packets=1 bytes=52 [ASSURED] mark=0 delta-time=29 use=1
```

TraceFlow

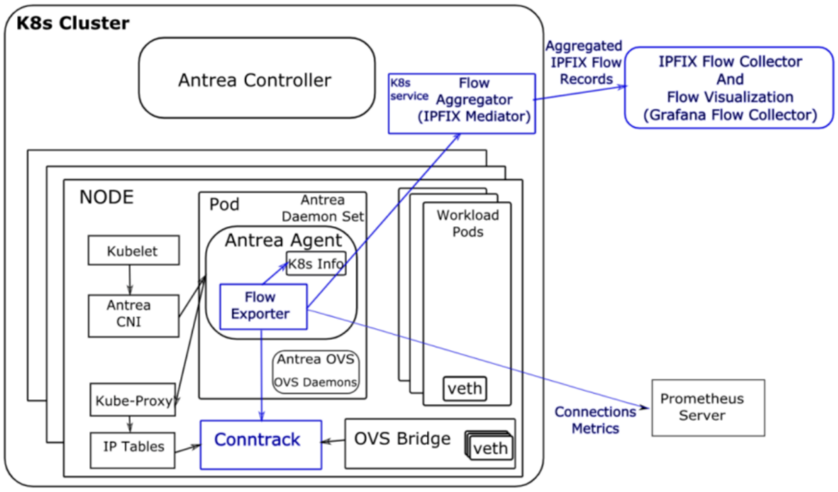
Antrea supports using Traceflow for network diagnosis. You must be familiar with this feature if NSX is not new to you. Creating a new Traceflow CRD triggers the Traceflow module to inject packet into OVS, provide various observation points along the packet's path, and populate these observations into the status field of the Traceflow CRD.

You can use antctl or octant to visualize Traceflow results:

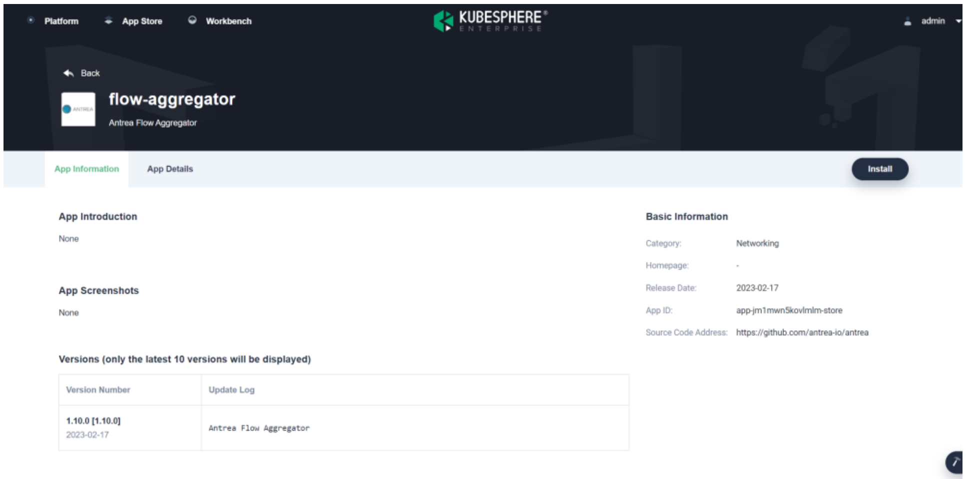
```
[root@ks1-master ~]#  antctl tf -S nginx-85b98978db-ckgtz -D nginx-85b98978db-fwbmw
name: nginx-85b98978db-ckgtz-to-nginx-85b98978db-fwbmw-rsc7pnr6
phase: Succeeded
source: default/nginx-85b98978db-ckgtz
destination: default/nginx-85b98978db-fwbmw
results:
- node: ks1-worker2
  timestamp: 1669629401
  observations:
  - component: Forwarding
    action: Received
  - component: Forwarding
    componentInfo: Output
    action: Delivered
- node: ks1-worker1
  timestamp: 1669629401
  observations:
  - component: SpoofGuard
    action: Forwarded
  - component: Forwarding
    componentInfo: Output
    action: Forwarded
    tunnelDstIP: 192.168.100.5
```

Network observability

To visualize network traffic, Antrea offers Flow Aggregator and Theia. Antrea monitors the flows in Linux conntrack module. These flows are converted to flow records, and then flow records are post-processed before they are sent to the configured external flow collector. In Antrea, the basic building block for the Network Flow Visibility is the Flow Exporter. Connections from the connection store are exported to the Flow Aggregator Service using the IPFIX protocol and displayed on Grafana through Theia.



Use KubeSphere App Store to deploy Flow Aggregator and Theia:



App Introduction

None

App Screenshots

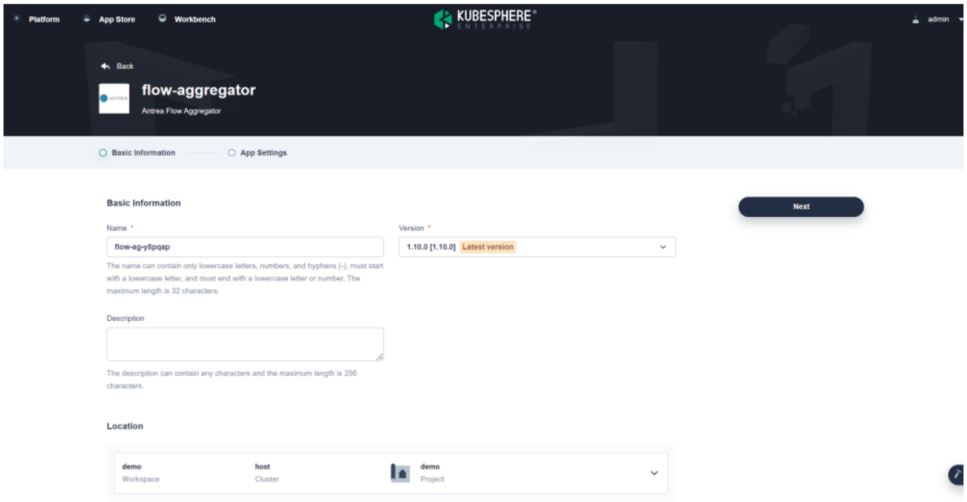
None

Versions (only the latest 10 versions will be displayed)

Version Number	Update Log
1.10.0 [1.10.0] 2023-02-17	Antrea Flow Aggregator

Basic Information

Category:	Networking
Homepage:	-
Release Date:	2023-02-17
App ID:	app-jm1mwn5kovlmlm-store
Source Code Address:	https://github.com/antrea-io/antrea



Basic Information

Name *

flow-ag-y8pqap

The name can contain only lowercase letters, numbers, and hyphens (-), must start with a lowercase letter, and must end with a lowercase letter or number. The maximum length is 32 characters.

Version *

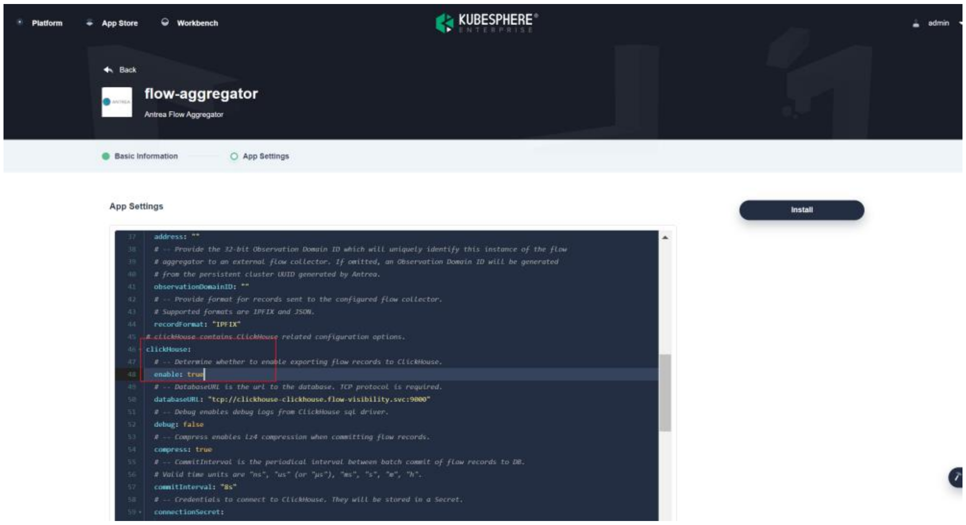
1.10.0 [1.10.0] Latest version

Description

The description can contain any characters and the maximum length is 256 characters.

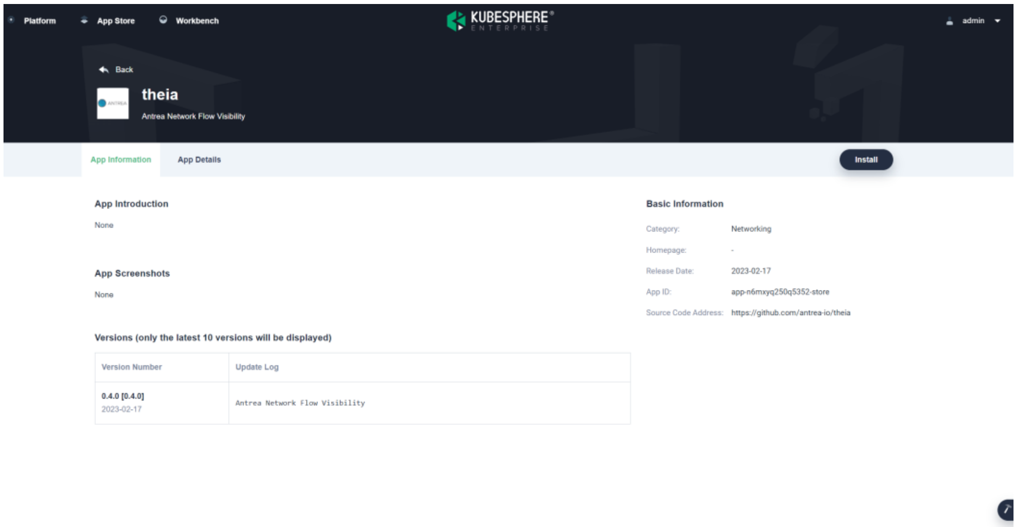
Location

demo Workspace	host Cluster	demo Project



App Settings

```
  address: ""
  # -- Provide the 32-bit Observation Domain ID which will uniquely identify this instance of the flow
  # aggregator to an external flow collector. If omitted, an Observation Domain ID will be generated
  # from the persistent cluster UUID generated by Antrea.
  observationDomainID: ""
  # -- Provide format for records sent to the configured flow collector.
  # Supported formats are IPFIX and JSON.
  recordFormat: "IPFIX"
# clickHouse contains ClickHouse related configuration options.
clickHouse:
  # -- Determine whether to enable exporting flow records to ClickHouse.
  enable: true
  # -- DatabaseURL is the url to the database. TCP protocol is required.
  databaseURL: "tcp://clickhouse-clickhouse.flow-visibility.svc:9000"
  # -- Debug enables debug logs from ClickHouse sql driver.
  debug: false
  # -- Compress enables lz4 compression when committing flow records.
  compress: true
  # -- CommitInterval is the periodical interval between batch commit of flow records to DB.
  # Valid time units are "ns", "us" (or "µs"), "ms", "s", "m", "h".
  commitInterval: "8s"
  # -- Credentials to connect to ClickHouse. They will be stored in a Secret.
  connectionSecret:
```



theia

Antrea Network Flow Visibility

App Introduction

None

App Screenshots

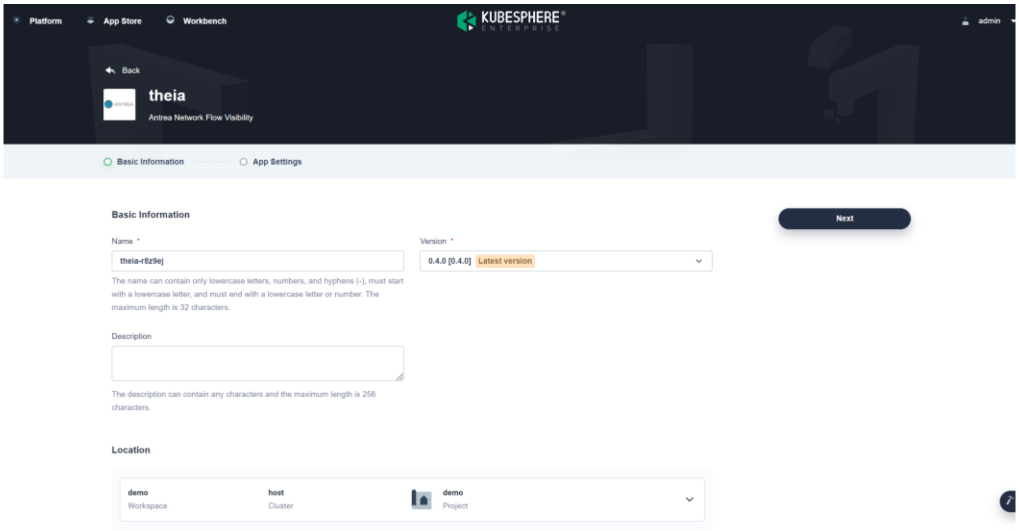
None

Versions (only the latest 10 versions will be displayed)

Version Number	Update Log
0.4.0 [0.4.0] 2023-02-17	Antrea Network Flow Visibility

Basic Information

Category:	Networking
Homepage:	-
Release Date:	2023-02-17
App ID:	app-n6mxyq250q5352-store
Source Code Address:	https://github.com/antrea-io/theia



theia

Antrea Network Flow Visibility

Basic Information

Name *

theia-r8z9ej

The name can contain only lowercase letters, numbers, and hyphens (-), must start with a lowercase letter, and must end with a lowercase letter or number. The maximum length is 32 characters.

Version *

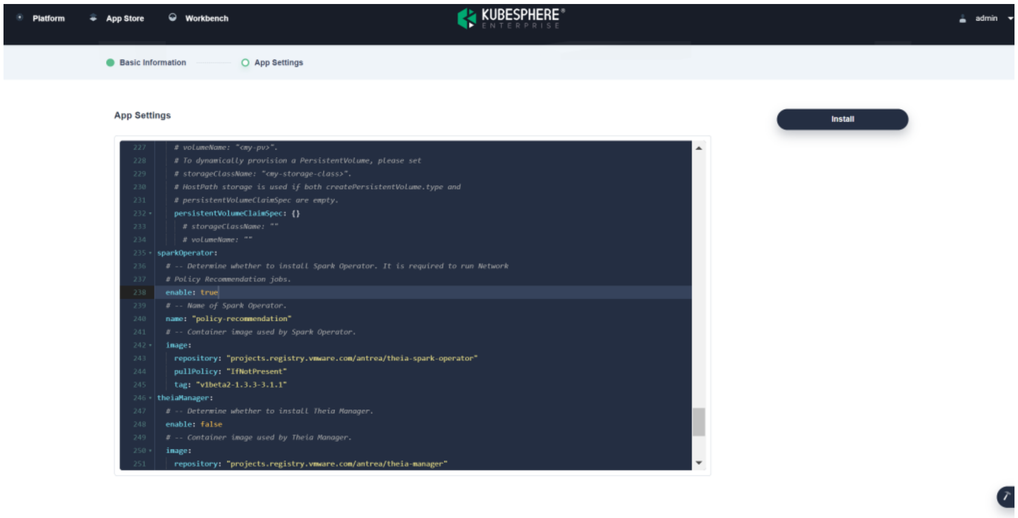
0.4.0 [0.4.0] Latest version

Description

The description can contain any characters and the maximum length is 256 characters.

Location

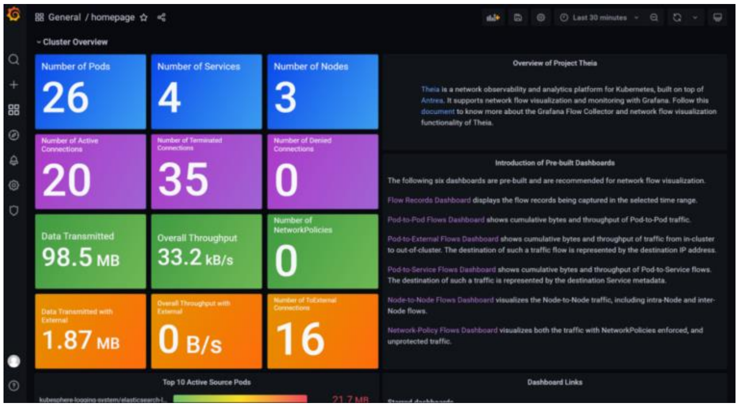
demo Workspace	host Cluster	demo Project



App Settings

```
    # volumeName: "<my-pv>".
    # To dynamically provision a PersistentVolume, please set
    # storageClassName: "<my-storage-class>".
    # HostPath storage is used if both createPersistentVolume.type and
    # persistentVolumeClaimSpec are empty.
    persistentVolumeClaimSpec: {}
      # storageClassName: ""
      # volumeName: ""
sparkOperator:
  # -- Determine whether to install Spark Operator. It is required to run Network
  # Policy Recommendation jobs.
  enable: true
  # -- Name of Spark Operator.
  name: "policy-recommendation"
  # -- Container image used by Spark Operator.
  image:
    repository: "projects.registry.vmware.com/antrea/theia-spark-operator"
    pullPolicy: "IfNotPresent"
    tag: "v1beta2-1.3.3-3.1.1"
theiaManager:
  # -- Determine whether to install Theia Manager.
  enable: false
  # -- Container image used by Theia Manager.
  image:
    repository: "projects.registry.vmware.com/antrea/theia-manager"
```

View statistical data on Grafana:



Number of Pods 26

Number of Services 4

Number of Nodes 3

Number of Active Connections 20

Number of Terminated Connections 35

Number of Denied Connections 0

Data Transmitted 98.5 MB

Overall Throughput 33.2 kB/s

Number of NetworkPolicies 0

Data Transmitted with External 1.87 MB

Overall Throughput with External 0 B/s

Number of ToExternal Connections 16

Overview of Project Theia

Theia is a network observability and analytics platform for Kubernetes, built on top of Antrea. It supports network flow visualization and monitoring with Grafana. Follow this document to know more about the Grafana Flow Collector and network flow visualization functionality of Theia.

Introduction of Pre-built Dashboards

The following six dashboards are pre-built and are recommended for network flow visualization.

Flow Records Dashboard displays the flow records being captured in the selected time range.

Pod-to-Pod Flows Dashboard shows cumulative bytes and throughput of Pod-to-Pod traffic.

Pod-to-External Flows Dashboard shows cumulative bytes and throughput of traffic from in-cluster to out-of-cluster. The destination of such a traffic flow is represented by the destination IP address.

Pod-to-Service Flows Dashboard shows cumulative bytes and throughput of Pod-to-Service flows. The destination of such a traffic is represented by the destination Service metadata.

Node-to-Node Flows Dashboard visualizes the Node-to-Node traffic, including intra-Node and inter-Node flows.

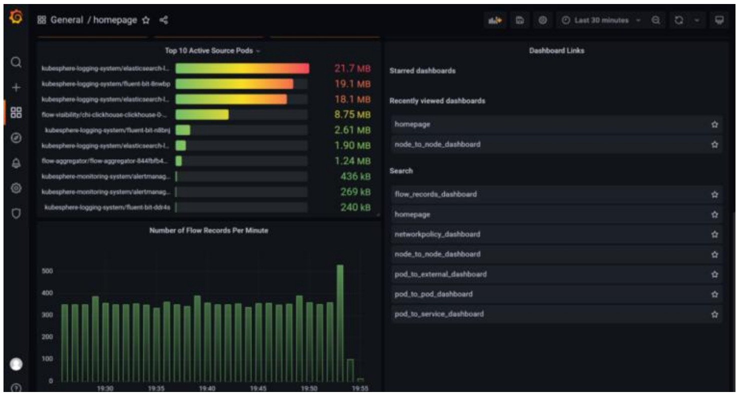
Network-Policy Flows Dashboard visualizes both the traffic with NetworkPolicies enforced, and unprotected traffic.

Top 10 Active Source Pods

kubesphere-logging-system/elasticsearch-l... 21.7 MB

Dashboard Links

Starred dashboards



Top 10 Active Source Pods

kubesphere-logging-system/elasticsearch-l...	21.7 MB
kubesphere-logging-system/fluent-bit-8nwbp	19.1 MB
kubesphere-logging-system/elasticsearch-l...	18.1 MB
flow-visibility/chi-clickhouse-clickhouse-0-...	8.75 MB
kubesphere-logging-system/fluent-bit-n8bnj	2.61 MB
kubesphere-logging-system/elasticsearch-l...	1.90 MB
flow-aggregator/flow-aggregator-844fbfb4...	1.24 MB
kubesphere-monitoring-system/alertmanag...	436 kB
kubesphere-monitoring-system/alertmanag...	269 kB
kubesphere-logging-system/fluent-bit-ddr4s	240 kB

Dashboard Links

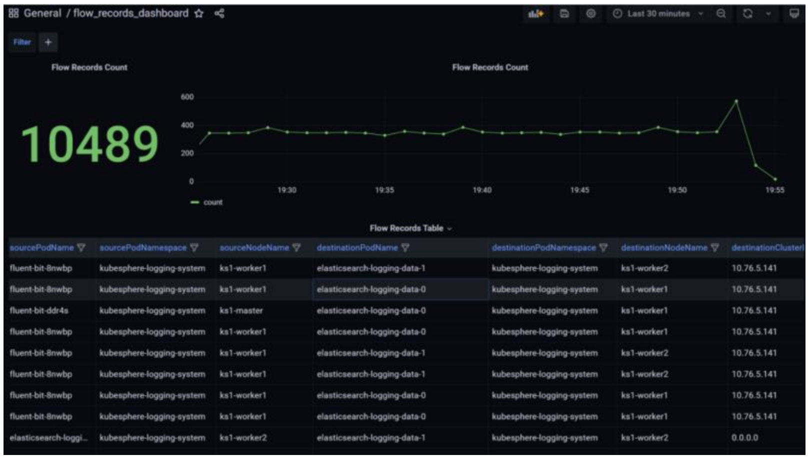
Starred dashboards

Recently viewed dashboards

- homepage ☆
- node_to_node_dashboard ☆

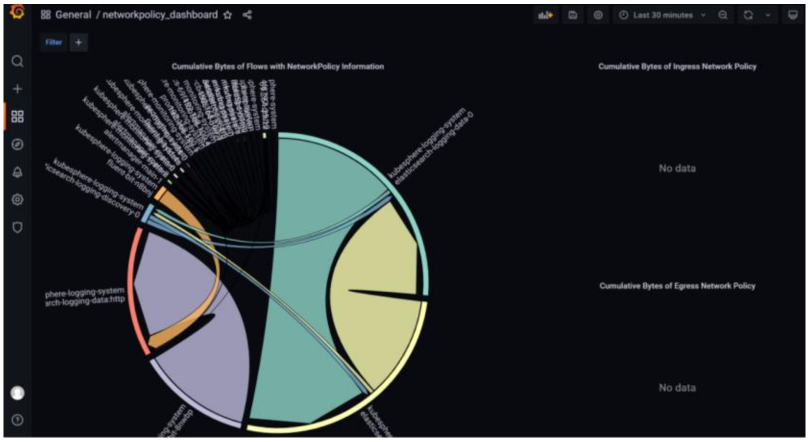
Search

- flow_records_dashboard ☆
- homepage ☆
- networkpolicy_dashboard ☆
- node_to_node_dashboard ☆
- pod_to_external_dashboard ☆
- pod_to_pod_dashboard ☆
- pod_to_service_dashboard ☆



Flow Records Table

sourcePodName	sourcePodNamespace	sourceNodeName	destinationPodName	destinationPodNamespace	destinationNodeName	destinationClusterI
fluent-bit-8nwbp	kubesphere-logging-system	ks1-worker1	elasticsearch-logging-data-1	kubesphere-logging-system	ks1-worker2	10.76.5.141
fluent-bit-8nwbp	kubesphere-logging-system	ks1-worker1	elasticsearch-logging-data-0	kubesphere-logging-system	ks1-worker1	10.76.5.141
fluent-bit-ddr4s	kubesphere-logging-system	ks1-master	elasticsearch-logging-data-0	kubesphere-logging-system	ks1-worker1	10.76.5.141
fluent-bit-8nwbp	kubesphere-logging-system	ks1-worker1	elasticsearch-logging-data-0	kubesphere-logging-system	ks1-worker1	10.76.5.141
fluent-bit-8nwbp	kubesphere-logging-system	ks1-worker1	elasticsearch-logging-data-1	kubesphere-logging-system	ks1-worker2	10.76.5.141
fluent-bit-8nwbp	kubesphere-logging-system	ks1-worker1	elasticsearch-logging-data-1	kubesphere-logging-system	ks1-worker2	10.76.5.141
fluent-bit-8nwbp	kubesphere-logging-system	ks1-worker1	elasticsearch-logging-data-0	kubesphere-logging-system	ks1-worker1	10.76.5.141
fluent-bit-8nwbp	kubesphere-logging-system	ks1-worker1	elasticsearch-logging-data-0	kubesphere-logging-system	ks1-worker1	10.76.5.141
elasticsearch-loggi...	kubesphere-logging-system	ks1-worker2	elasticsearch-logging-data-1	kubesphere-logging-system	ks1-worker2	0.0.0.0


General / networkpolicy_dashboard
Last 30 minutes
Filter
Cumulative Bytes of Flows with NetworkPolicy Information
kubesphere-logging-system
elasticsearch-logging-data-0
phere-logging-system
arch-logging-data:http
Cumulative Bytes of Ingress Network Policy
No data
Cumulative Bytes of Egress Network Policy
No data

Theia also provides suggestions on network policies and security hardening for platform components and system architectures.

Sample code:

```
theia policy-recommendation run
theia policy-recommendation list
CreationTime        CompletionTime      ID                                   Status
2022-10-08 06:35:04 2022-10-08 06:38:00 60a79979-19ed-40a7-bc11-2b389a8a43bb COMPLETED
theia policy-recommendation retrieve --id 60a79979-19ed-40a7-bc11-2b389a8a43bb
apiVersion: crd.antrea.io/v1alpha1
kind: ClusterNetworkPolicy
metadata:
  name: recommend-allow-acnp-kube-system-f2g5l
spec:
  appliedTo:
  - namespaceSelector:
      matchLabels:
        kubernetes.io/metadata.name: kube-system
  egress:
  - action: Allow
    to:
    - podSelector: {}
  ingress:
  - action: Allow
```

```
    from:
    - podSelector: {}
  priority: 5
  tier: Platform
---
apiVersion: crd.antrea.io/v1alpha1
kind: ClusterNetworkPolicy
metadata:
  name: recommend-allow-acnp-flow-aggregator-uny5c
spec:
  appliedTo:
  - namespaceSelector:
      matchLabels:
        kubernetes.io/metadata.name: flow-aggregator
  egress:
  - action: Allow
    to:
    - podSelector: {}
  ingress:
  - action: Allow
    from:
    - podSelector: {}
  priority: 5
  tier: Platform
---
apiVersion: crd.antrea.io/v1alpha1
kind: ClusterNetworkPolicy
metadata:
  name: recommend-allow-acnp-flow-visibility-ecj61
spec:
  appliedTo:
  - namespaceSelector:
      matchLabels:
        kubernetes.io/metadata.name: flow-visibility
  egress:
  - action: Allow
    to:
    - podSelector: {}
  ingress:
  - action: Allow
    from:
    - podSelector: {}
  priority: 5
  tier: Platform
```

Multi-cluster networking

Antrea Multi-cluster implements connectivity between pods or services. A Multi-cluster ClusterSet is comprised of a single leader cluster and at least two member clusters. Then, pods can access each other via pod IP addresses, and services can access each other via import or export.

Assume that the following two clusters exist:

Cluster A: Apiserver 192.168.100.3:6443, podCIDR 10.10.0.0/16

Cluster B: Apiserver 192.168.100.6, podCIDR 10.20.0.0/16

Enable the multi-cluster feature:

```
kubectl edit cm -n kube-system
  # Enable Antrea Multi-cluster Gateway to support cross-cluster traffic.
  # This feature is supported only with encap mode.
    Multicluster: true
```

Create a namespace named **antrea-multicluster** and use antctl to deploy ClusterSet:

```
kubectl create ns antrea-multicluster
antctl mc deploy leadercluster -n antrea-multicluster --antrea-version v1.10.0
antctl mc deploy membercluster -n kube-system --antrea-version v1.10.0
```

Create a ClusterSet in cluster A, which works as the leader cluster and has two member clusters.

```
antctl mc init --clusterset test-clusterset --clusterid test-cluster-leader -n antrea-multicluster --create-token -j join-config.yml
antctl mc join --clusterid test-cluster-leader -n kube-system --config-file join-config.yml
```

Query ClusterClaim and ClusterSet.

```
kubectl get clustersets.multicluster.crd.antrea.io -A
NAMESPACE           NAME              LEADER CLUSTER NAMESPACE   TOTAL CLUSTERS   READY CLUSTERS   AGE
antrea-multicluster   test-clusterset   antrea-multicluster        1                1                4m12s
kube-system         test-clusterset   antrea-multicluster        1                1                4m4s
kubectl get clusterclaims.multicluster.crd.antrea.io -A
NAMESPACE           NAME              VALUE               AGE
antrea-multicluster   clusterset.k8s.io   test-clusterset       4m32s
antrea-multicluster   id.k8s.io         test-cluster-leader   4m32s
kube-system         clusterset.k8s.io   test-clusterset       4m24s
kube-system         id.k8s.io         test-cluster-leader   4m24s
```

Specify the Gateway Node for cluster A, which is responsible for routing all cross-clusters traffic from the local cluster to other member clusters through tunnels.

```
kubectl annotate node ks1-master multicluster.antrea.io/gateway=true
```

Deploy member cluster B.

```
antctl mc deploy membercluster -k ~/.kube/config1 -n kube-system --antrea-version v1.10.0
```

Add cluster B to ClusterSet.

```
antctl mc deploy membercluster -k ~/.kube/config1 -n kube-system --antrea-version v1.9.0
```

Configure the Gateway Node for cluster B.

```
kubectl annotate node ks2 multicluster.antrea.io/gateway=true
```

Check the cluster status:

```
antctl mc get clusterset -A
CLUSTER-ID         NAMESPACE           CLUSTERSET-ID   TYPE     STATUS REASON
test-cluster-leader antrea-multicluster test-clusterset Ready    True   Connected
test-cluster-leader kube-system         test-clusterset IsLeader True   <NONE>
test-cluster-leader kube-system         test-clusterset Ready    True   <NONE>
test-cluster-member antrea-multicluster test-clusterset Ready    True   Connected
```

Check cluster import:

```
kubectl get clusterinfoimport -n kube-system
NAME                          CLUSTER ID          SERVICE CIDR   AGE
test-cluster-member-clusterinfo   test-cluster-member   10.86.0.0/16   28m
```

```
[root@ks2 ~]# kubectl get clusterinfoimport -n kube-system
NAME                          CLUSTER ID          SERVICE CIDR   AGE
test-cluster-leader-clusterinfo   test-cluster-leader   10.76.0.0/16   32m
```

In **antrea-mc-controller**, specify **podCIDR** for which you want to enable connectivity:

```
kubectl edit configmap -n kube-system antrea-mc-controller-config
data:
  controller_manager_config.yaml: |
    apiVersion: multicluster.crd.antrea.io/v1alpha1
    kind: MultiClusterConfig
    health:
      healthProbeBindAddress: :8080
    metrics:
      bindAddress: "0"
    webhook:
      port: 9443
    leaderElection:
      leaderElect: false
    serviceCIDR: ""
    podCIDRs:
      - "10.10.0.0/16"  #另一边为 10.20.0.0/16
    gatewayIPPrecedence: "private"
    endpointIPType: "ClusterIP"
```

Test pod connectivity:

```
kubectl get pods -o wide
NAME   READY   STATUS    RESTARTS   AGE     IP          NODE         NOMINATED NODE   READINESS GATES
box2   1/1     Running   0          5m29s   10.10.2.2   ks1-worker1   <none>           <none>
kubectl get pods -o wide
NAME   READY   STATUS    RESTARTS   AGE     IP          NODE   NOMINATED NODE   READINESS GATES
box1   1/1     Running   0          5m14s   10.20.0.4   ks2    <none>           <none>
kubectl exec box1 ping 10.10.2.2
PING 10.10.2.2 (10.10.2.2): 56 data bytes
64 bytes from 10.10.2.2: seq=0 ttl=61 time=2.832 ms
```

Best Practices

This guide leverages the advantages of VMware cloud-native infrastructure and QingCloud KubeSphere container platform to provide users with a full-lifecycle cloud-native solution. For information about implementation and design, see the following best practices:

- Best practices for cloud native storage
  - Best practices for deploying VMware CNS
- Best practices for cloud native networking:
  - Best practices for deploying Antrea
- Best practices for deploying KubeSphere on VMware vSphere
  - Deploy KubeSphere on VMware vSphere
- Best practices for container management
  - Manage containers on KubeSphere
- DevOps Best Practices

  - Best practices for KubeSphere DevOps

Conclusion

Powered by QingCloud, KubeSphere can work together with VMware in a seamless manner to provide cloud native solutions, for example, integration with vSAN and CNS to provide storage solutions, and with Antrea to provide network solutions. VMware features high performance, high availability, stability, and scalability, and KubeSphere is scalable, open source, and native to clouds. This indicates that KubeSphere can help you run containers on VMware and minimize the learning costs and O&M pressure, so you can care about only the upper-layer applications and accelerate digital transformation of your business.

Reference

- VMware CNS
- *Antrea*
- KubeSphere

About the Author

Wei MA, Solution Architect at KubeSphere, QingCloud

Shuang YU, Product Director at QingCloud

The following reviewers also contributed to the paper contents:

- Fei LIU, Senior Solution Architect at VMware
- Jiali XU, Senior Manager at Cloud Infrastructure Solutions Department, VMware
- Qin XU, Senior Solution Architect at Cloud Infrastructure Solutions Department, VMware
- Ting YIN, Senior Solution Architect at Cloud Infrastructure Solutions Department, VMware
- Min ZHANG, Senior Network Security Product Expert at VMware

VMware, Inc. 3401 Hillview Avenue Palo Alto CA 94304 USA Tel 877-486-9273 Fax 650-427-5001 www.vmware.com.