实验08-Persistence Storage

一、实验目的

- 1、了解 Kubernetes 存储类;
- 2、掌握 NFS 共享存储的配置;
- 3、掌握 Kubernetes NFS 存储类的基本应用。

二、实验学时

2学时

三、实验类型

设计性

四、实验任务

1、完成 NFS 共享存储的搭建;

- 2、完成集群使用NFS共享存储;
- 3、完成持久化服务应用发布;
- 4、完成 Wordpress 服务发布。

五、实验环境

1、硬件

本实验基于实验教学中心网络运维实验室服务器集群开展,每个实验小组分配集群中的1台物 理服务器作为实验基础平台,提供云计算资源。每个人配备计算机1台。(学生可根据自身情况 使用个人计算机)。

2、软件

Windows操作系统,或MacOS操作系统。 安装最新版本的浏览器,建议使用Edge、Chrome等。

3、网络

计算机使用无线网络接入局域网,能够访问实验教学中心网络运维实验室服务器集群,并支持 对互联网的访问。

4、工具

无

六、实验内容步骤

1、云数据中心存储规划

(1) NFS 服务器规划

共享存储服务器采用 NFS,服务器使用虚拟机,配置信息如表 8-1 所示。

表 8-1 共享存储服务器配置信息

序号	虚拟机配置	操作系统配置
1	虚拟机名称: Cloud-K8s-NFS CPU: 2核 内存: 2GB 硬盘: 50GB(系统盘)+5*20GB(共享存储) 网卡: Class-Cloud-VM-Network	主机名: Cloud-K8s-NFS 操作系统: openEuler 24.03 LT IP 地址: 172.16.125.106 子网掩码: 255.255.255.0 网关: 172.16.125.1 DNS: 172.16.125.3

(2) NFS 共享存储目录规划

使用1台虚拟机建设NFS服务器,虚拟机配置5个20GB硬盘用于发布NFS共享存储,在NFS服务器上创建5个目录作为挂载点,供K8s集群使用,允许5个K8s节点访问,共享存储挂载规划如表8-2所示。

表 8-2 共享存储挂载规划表

序号	存储名称	挂载目录	存储容量	
1	cloud-k8s-nfs-1	/K8s-NFS-1	20GB	र्भ
2	cloud-k8s-nfs-2	/K8s-NFS-2	20GB	Я
3	cloud-k8s-nfs-3	/K8s-NFS-3	20GB	र्भ
4	cloud-k8s-nfs-4	/K8s-NFS-4	20GB	Я
5	cloud-k8s-nfs-5	/K8s-NFS-5	20GB	Я

2、搭建NFS服务器

步骤1:安装NFS服务并创建共享目录

```
Shell

1 #安装NFS相关服务
2 [root@Cloud-K8s-NFS ~]# yum install -y nfs-utils rpcbind

4 #启动服务以及设置自启
5 [root@Cloud-K8s-NFS ~]# systemctl start nfs-server
6 [root@Cloud-K8s-NFS ~]# systemctl enable nfs-server
7 [root@Cloud-K8s-NFS ~]# systemctl start rpcbind
8 [root@Cloud-K8s-NFS ~]# systemctl enable rpcbind
9
10 #创建共享目录并授权
11 [root@Cloud-K8s-NFS ~]# mkdir /{K8s-NFS-1,K8s-NFS-2,K8s-NFS-3,K8s-NFS-4,K8s-NFS-5}
12 [root@Cloud-K8s-NFS ~]# chmod -R 777 /{K8s-NFS-1,K8s-NFS-2,K8s-NFS-3,K8s-NFS-3,K8s-NFS-3,K8s-NFS-5}
```

步骤 2: 格式化磁盘

```
1 #查看存储设备信息
2 [root@Cloud-K8s-NFS ~]# lsblk
3 NAME
                  MAJ:MIN RM SIZE RO TYPE MOUNTPOINTS
4 sda
                    8:0 0 50G 0 disk
5 – sda1
                    8:1
                         0
                             1M 0 part
6 \vdash sda2
                    8:2 0
                              1G 0 part /boot
7 └─sda3
                    8:3 0 19G 0 part
9 └─openeuler-swap 253:1 0 2G 0 lvm [SWAP]
                    8:16 0 20G 0 disk
10 sdb
11 sdc
                    8:32 0 20G 0 disk
12 sdc
                    8:48 0 20G 0 disk
13 sde
                    8:64 0 20G 0 disk
14 sdf
                    8:80 0 20G 0 disk
15 sr0
                   11:0 1 1024M 0 rom
16
17 #格式化磁盘
18 [root@Cloud-K8s-NFS ~]# mkfs.ext4 /dev/sdb
19 mke2fs 1.47.0 (5-Feb-2023)
20 丢弃设备块:完成
21 创建含有 5242880 个块(每块 4k)和 1310720 个 inode 的文件系统
22 文件系统 UUID: 964b918d-422e-474f-9d5a-e7710aaf9e74
23 超级块的备份存储于下列块:
24
         32768, 98304, 163840, 229376, 294912, 819200, 884736, 1605632,
  2654208,
25
         4096000
26
27 正在分配组表:完成
28 正在写入 inode表: 完成
29 创建日志 (32768 个块): 完成
30 写入超级块和文件系统账户统计信息:已完成
31 #格式化其他磁盘
32 [root@Cloud-K8s-NFS ~]# mkfs.ext4 /dev/sdc
33 [root@Cloud-K8s-NFS ~]# mkfs.ext4 /dev/sdd
34 [root@Cloud-K8s-NFS ~]# mkfs.ext4 /dev/sde
35 [root@Cloud-K8s-NFS ~]# mkfs.ext4 /dev/sdf
```

步骤3:将磁盘挂载到共享目录

```
1 #进行挂载
 2 [root@Cloud-K8s-NFS ~]# mount /dev/sdb /K8s-NFS-1
 3 [root@Cloud-K8s-NFS ~]# mount /dev/sdc /K8s-NFS-2
 4 [root@Cloud-K8s-NFS ~]# mount /dev/sdd /K8s-NFS-3
 5 [root@Cloud-K8s-NFS ~]# mount /dev/sde /K8s-NFS-4
 6 [root@Cloud-K8s-NFS ~]# mount /dev/sdf /K8s-NFS-5
 7
 8 #查看挂载情况
 9 [root@Cloud-K8s-NFS ~]# lsblk
10
11 #修改配置文件
12 [root@Cloud-K8s-NFS ~]# cat >> /etc/exports <<EOF
13 > /K8s-NFS-1 172.16.125.0/24(rw,sync,no_root_squash)
14 > /K8s-NFS-2 172.16.125.0/24(rw,sync,no_root_squash)
15 > /K8s-NFS-3 172.16.125.0/24(rw,sync,no_root_squash)
16 > /K8s-NFS-4 172.16.125.0/24(rw,sync,no_root_squash)
17 > /K8s-NFS-5 172.16.125.0/24(rw,sync,no_root_squash)
18 > EOF
19 cat /etc/exports
20
21 #重启服务
22 systemctl restart nfs-server
23 showmount -e
```

3、为 K8s 集群添加 NFS 共享存储

步骤1:在每个集群节点上安装NFS客户端

Shell

1 #在每个集群节点上安装NFS客户端

- 2 yum install nfs-utils -y
- 3 systemctl start nfs-server
- 4 systemctl enable nfs-server

步骤 2:为 K8s 集群添加 NFS 共享存储

(1) 登录 Kuboard,在"集群管理"的"概要"中,创建存储类,如图 8-1、8-2 所示。



图 8-1



图 8-2 创建存储类

(2) 创建时,依据给出的测试命令,在主机上进行测试。

Shell
<pre>1 #在任意节点验证成功挂载该 NFS 服务 2 [root@k8s-worker3 ~]# mkdir /tmp/testnfs \ 3 && mount -t nfs 172.16.125.106:/K8s-NFS-2 /tmp/testnfs \ 4 && echo "hello nfs" >> /tmp/testnfs/test.txt \ 5 && cat /tmp/testnfs/test.txt 6 hello nfs</pre>

(3)测试通过后,保存应用并依次添加 cloud-k8s-nfs-2、cloud-k8s-nfs-3、cloud-k8s-nfs-4、cloud-k8s-nfs-5存储资源,如图 8-3 所示。

Kuboard	首页 〉 Cloud-K8s [切换]	> 集群概赏页 🔺 设为默认页	3				Kubernetes : v1.28.0 Kuboard : v3.5.2.7	ß	kuboard-admin 👻
∯ Cloud-K8s	B	安名称筛选	▶ 按名称升序	1 按名称降序 1 创建	时间升序 \ 创建时间降序	🖉 显示所有名称空间			
☯ 集群导入 ✓		D default	k kube-node	lease	kube-public	K kube-syste	m	ard	
□ 集群管理 へ	J 📀 🛛	Active 大约 2 小	Active	大约 2 小时	Active	大约 2 小时	大约 2 小时 Active	大约 1	terb
只 概要									
■ 节点	计算资源				٩	存储资源 K8s存储管理 🛙 🛛 s	C5/5 PV5/5		
ᇦ存儲 ✓	未安装metrics-server	N k8s-master	N k8s-worker1	N k8s-worke	r2	• cloud-k8s-nfs-1		9 分钟	点击查看存储类
© 网络 ∽	安装 metrics-server 安装后请手动刷新页面	172.16.125.101 Ready: True	172.16.125.102 Ready: True	172.16.125.103 Ready: True			暂无存储卷 (PersistentVolume)		
至 自定义资源 ∨	CPU内存低却无法调度	A49 2 3993	Asy 2 (198)	A89.2.999		cloud-k8s-nfs-2		5 分钟	点击查看存储类
选择		k8s-worker3 172.16.125.104	k8s-worker4 172.16.125.105				暂无存储卷 (PersistentVolume)		
C. N-V38244		Ready: True 大约 2 小时	Ready: True 大约 2 小时			cloud-k8s-nfs-3	每十七日本卷(Damintan® (clume)	2 分钟	点击查看存储类
							首元仔储卷(Persistentvolume)		
[] 操作审计						 cloud-k8s-nfs-4 	东丁左战带(Demisterflahme)	1 分钟	点击查看存储类
							智元存储卷(Persistentvolume)		
						 cloud-k8s-nfs-5 		不到1分钟	点击查看存储类
							智尤存储卷 (PersistentVolume)		
517% »				https://je_t	d an				
* 104 <u>-</u>				nttps://kuboar	a.cn				

图 8-3 添加NFS存储类

4、创建用户和命名空间

步骤1: 创建命名空间,如图8-4、8-5所示

Kuboard	<u>首页</u> > <u>Cloud-K8s</u> [切换]	> 集群概意页 🍙 设为默认页	0				KubernetesKuboard	: v1.28.0 : v3.5.2.7	K kub	oard-admin 👻
🛱 Cloud-K8s	2 名称空间			> 1 按名称降序 1 创建时间升序	1 创建时间降序 🛛	显示所有名称空间				
◎ 集群导入 ~	1 创建名和	你空间					8	kuboard		Ð
□ 集群管理 へ	名称	namspace@	1					Active	大约 4 小时	
口 概要										
□ 节点	计算资源						✓ 保存 × 取消		创建有	-16 .
母存储 ✓	未安装metrics-server	N k8s-master	🛿 k8s-worker1	🛛 k8s-worker2		 cloud-k8s-nfs-1 			大約3小时 点曲	遭看存储类
◎ 网络 ~	安装 metrics-server 安装后请手动刷新页面	172.16.125.101 Ready: True 大约 5 小时	172.16.125.102 Ready: True 大約 5 小时	172.16.125.103 Ready: True 大约 5 小时			暂无存储卷 (Per			
幸 自定义资源 シン	CPU内存低却无法调度	k8s worker3	k8s worker4			 cloud-k8s-nfs-2 	STEFFER (Per		大约3小时 点击	這看存储关
- 送择		172.16.125.104 Ready: True	172.16.125.105 Ready: True			 cloud-k8s-nfs-3 			大约 2 小时 点击	這看存储类
合访问控制 ~		大到 5 小时	大约 4 小时				暂无存储卷 (Per			
[3 操作审计						• cloud-k8s-nfs-4			大约2小时 点击	·查看存储类
							暂无存储卷(Per:			
						 cloud-k8s-nfs-5 	暂无存储卷(Per		大约2小时 点进	查看存储关
《 收	2			https://kuboard.cn						

图 8-4 创建命名空间

Kuboard	首页 〉 Cloud-K8s	(1)) · 集群概赏页 A 设为默认页 🕄)				KubernetesKuboard	v1.28.0 v3.5.2.7	kuboard-admin 🕶
∯ Cloud-K8s		按名称筛选	▶ 按名称升序 、	1 按名称降序 1 创图	數时间升序 ↓ 创建时间降序	🛃 显示所有名称空间			1.0
它集群导入		Active 大约 5 小时	Active	大约 5 小时	Active	大約 5 小时 Active	大約 5 小时	Active	大约 4 小时
		namspace01	namspace0	2	namspace03		04	namspace	15
		2025-03-21 14:41:02 Active 1 2010	2025-03-21 14:41:20 Active	1 分钟	2025-03-21 14:41:42 Active	2025-03-21 14:41: 1 5%	i6 不到 1 分钟	2025-03-21 14:42:13 Active	不到 1 分钟
	计算资源				٩	存储资源 K8s存储管理 🛙 🛛	SC 5 / 5 PV 5 / 5		创建存储类
₿存儲	★安装metrics-server	N k8s-master	N k8s-worker1	N k8s-worke	er2	cloud-k8s-nfs-1		7	约3小时 点击查看存储类
@ 网络	 安装 mencs-server 安装后请手动刷新页面 	172.16.125.101 Ready: True 大约 5 小时	172.16.125.102 Ready: True 大约 5 小时	172.16.125.103 Ready: True 大约 5 小时			暂无存储卷 (Pers	sistentVolume)	
至 自定义资源	✓ CPU内存低却无法调度					cloud-k8s-nfs-2	毎天左縁券 (Pors	t	均 3 小时 点击查看存储类
- 送择		172.16.125.104 Ready: True	172.16.125.105 Ready: True			- aloud k0a mfa 2		, stone volumo y	
合 访问控制	~	大约 5 小时	大约 5 小时			• cloud-kos-liis-3	暂无存储卷 (Pers	z sistentVolume)	《约3小阳 息击宣有仔怀突
[] 操作审计						 cloud-k8s-nfs-4 		ť	约2小时 点击查看存储类
							暂无存储卷 (Pers	sistentVolume)	
						• cloud-k8s-nfs-5		7	均 2 小时 点击查看存储类
							暂无存储卷 (Pers	sistentVolume)	
				https://je-h-e-	rd en				

图 8-5 创建命名空间

步骤 2:为每个命名空间绑定存储类,点击"存储资源"的"查看存储类",选择"限定命名空间"并指定具体的命名空间,如图 8-6、8-7 所示。

Kuboard	首页 〉 Clo	oud-K8s [切換] 〉 集群概赏页	a 2580.0	Kubernetes : v1.28.0	K kuboard-admin 👻
🔅 Cloud-K8s	cloud-k8s-nfs-1 🛊	存儲类 🖸 🚱		•	3
◎ 集群导入	* 名称	cloud-k8s-nfs-1 🕄	22歳(),		
□ 集群管理	限定各称空间	☑ 限定 namspace0 在 Kuboard	1 ●		१९ ४ ग्रेक्ष
口 概要	* 存储类 类型	 NFS CephFS 	● CephFS Rook 使用其他类型的存储 If ●		
▦ 节点	NFS 参数	了解如何获取 NFS 参数	₫ .		创建存储类 ④
ᇦ 存储		* NFS Server	172.16.125.106		1时 点击查看存储类
@ 网络		* NFS Path	/K8s-MFS-1		
至 白定义资源		* NFS 可使用容量	2861		时 点击查看存储关
选择		MountOptions 🖻 🕲	38 30		
		请确保任意节点都可成3	挂载该 NFS 服务,可通过下面的命令测试:		时 点击查看存储类
11 功问控制		<pre>mkdir /tmp/testnfs && mount -t nfs 172 && echo "hello nfs"</pre>	\ .16.125.106:/K8s-HFS-1 /tmp/testnfs \ >/tmp/testnfs/test.tst \		
[] 操作审计		&& cat /tmp/testnfs	'test.txt		时 点击查着存储关
		★镜像 ○ eipwo ⊙ swr.cn	k/nfs-subdir-external-provisioner.v4.0.2 east-2.mvhuaweicloud.com/kuboard-dependencv/nfs-subdir-external-provisioner.v4.0.2		
		〇 从私有	皇帝合志與印政 nfs-client-provisioner 镜像		四日 点击宣复存储关
	* 回收策略	 回收后删除 	· Release		

图 8-6 指定命名空间

Kuboard	<u>首页</u> > <u>Cloud-K8s</u> [切换]	> 集群概览页 🔺 设为默认页 💽	•		:	Kubernetes : v1.28.0 Kuboard : v3.5.2.7	K kuboard-admin 🕶
O Cloud-K8s		按名称筛选	▶ 按名称升序	↓ 按名称踪序 ト 创建时间升序 ↓ 创建时间降序	🖉 显示所有名称空间		
② 集群导入		D default	kube-node-	-lease	kube-system	kuboard	
□ 集群管理		2025-03-21 09:52:53 Active 大约 5 小时	2025-03-21 09:52:53 Active	大约 5 小时	大约 5 小时 Active 大	約 5 小时 Active	大约 4 小时
口 概要	(°	namsnace01	namsnace0		namenace04		05
前点	计算资源			٩	存储资源 K8s存储管理 🗹 🜒 SC 5/5	PV 5 / 5	创建存储类
⊜ 存储	★安装metrics-server	N k8s-master	N k8s-worker1	🔇 k8s-worker2	cloud-k8s-nfs-1		大约3小时 点击查看存储类
@ 网络	安装 metrics-server 安装后清手动刷新页面	172.16.125.101 Ready: True 大约 5 小时	172.16.125.102 Ready: True 大约 5 小时	172.16.125.103 Ready: True 大约 5 小时	暂无术 只能在名称空间 r	字储卷(PersistentVolume) namspace01 中选择此存储类(Kuboard O	nly)
至 自定义资源	✓ CPU内存低却无法调度●	N k8s-worker3	N k8s-worker4]	cloud-k8s-nfs-2		大约 3 小时 点击查看存储类
📕 名称空间 - 选择		172.16.125.104 Ready: True 大约 5 小时	172.16.125.105 Ready: True 大约 5 小时		暂无术 只能在名称空间 r	字储卷(PersistentVolume) namspace02 中选择此存储类(Kuboard O	nly)
合 访问控制	~			J	• cloud-k8s-nfs-3		大约 3 小时 点击查看存储类
[] 操作审计					暂无利 只能在名称空间 r	字储卷(PersistentVolume) namspace03 中选择此存储类(Kuboard O	nly)
					 cloud-k8s-nfs-4 		大约 3 小时 点击查看存储类
					暫元7 只能在名称空间 r	字储卷(PersistentVolume) namspace04 中选择此存储类(Kuboard O	nly)
							. J
«	KUZ			https://kuboard.cn			

图 8-7 命名空间绑定存储类

步骤3:创建用户并为用户指定命名空间。

(1) 创建用户 user01-user05, 如图 8-8 所示。

Kuboard	<u>首页</u> > 用户列表 A 设为默认页 😯				 Kuboard : v3.5.2.7 	A admin -
▲ Kubernetes 集群	Ⅰ 用户列表					
名 用户与权限 へ	援权用户访问 Kubernetes 集群中的某一个名称空间 🛙 🔮					+ 创建用户
用户	名称	创建时间	最后登录时间	是否启用	操作	
用户组	admin	大約 4 小时	大約 4 小时		と 査 看 区 修改商码 自 删除	
角色	user01	2.分钟	1		④ 查看 区 修改密码 自 删除	
⑦ 全局设置 >	user02	1.分钟			心 査 看 区 修改密码 自 删除	
W EMAR	user03	1.分钟	1		心 查看 IC 修改密码 自 删除	
⑦ 个人设置 ~	user84	不到 1 分钟			心 査 看 区 修改密码 合 删除	
[] 操作审计	user05	不到 1 分钟	÷.		查看 区 修改密码 自 删除	
Etre »						

图 8-8 创建用户

(2)创建用户组 user,并关联到用户,点击"角色绑定(集群)"创建角色绑定,将用户组绑 定到"sso-user"角色,此时用户组的用户可以访问到本地集群 Cloud K8s,如图 8-9、8-10 所 示。

Kuboard		首页 〉 用户组列表	>user ♠ 设为默认页	9			• Kuboard : v3.5.2.7 🔥 admin 🗸
▲ Kubernetes 集群		┃用户组					
名 用户与权限	^	基本信息					
用户		用户组名	user				
用户组		描述	user 区编辑				
角色		创建时间	1分钟				
◎ 全局设置	~						
〇 个人设置	~	关联用户	角色绑定(全局)	角色绑定(集群)			
「9 操作审计							+ 关联用户到用户组
		Group	User	关联时间		操作	
		user	user01	不到1分钟		合 删除	
		user	user02	不到1分钟		白 副除	
		user	user03	不到1分钟		☆ 删除	
		user	user04	不到1分钟		由 删除	1
		user	user05	不到1分钟		白 删除	•
	< 1018				https://kuboard.co		

图 8-9 关联用户

首页 〉 用户组列表 〉 user ▲	设为默认页 😳				• Ki	uboard : v3.5.2.7	admin 🔹		
Ⅰ 用户组									
基本信息									
用户组名 user									
描述 user IC 编载									
创建时间 2分钟									
关联用户 角色绑定(全局)) 角色绑定(集	群)							
授权用户访问 Kubernetes 集群中	的某一个名称空间 🖸 🖗					+ 创建角色频	定 (集群级别)		
朱群名称 ⇔ 主体	体类型 🗢	主体名称 💠	角色 ≑	关联时间 🗢	操作				
Cloud-K8s Kul	uboardAuthGroup	user	sso-user	不到1分钟	亩 删除				
							(e		
			https://kubpard.cp						
	 ・	RM / 加光型2000 / User ▲ 622003 で 日用产组 基本信息 用产组名 user 服法 user 回 病気 创趣时间 2.分钟 关联用户 角色频应 (全局) 角色频应 (集 级仅用户访问 Kubernetes 集群中的某一个名称空问 C ● 解释名称 : 主体実型 : Choue Kds KubeardkuthGroup	RAL 7 METADARA 7 USET ● 50200.00 で 日井宁组 基本信息 月戸報名 国志 USET 細志 USET 回 楽唱 創建2月月 2.分钟 実現用中 角色病症 (金周) 発気用中 角色病症 (金周) 単数用中 白色病症 (金周) 単数日 生 単数日 生 ● 単数日 ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● <th>展升 / 推出現決致 / User ▲ 62/2004 ● 基本信息 ● 用户组 基本信息 用户组 ● 服告 user 服告 user 服告 user 服告 user 服告 user 服告 (金属) 創設的词 2.分钟 经取用户访问 Kubernetes 集影中的某一个名称空问 CI ● 解释命 = 主体发标 = 角色 = Cbute KS KubeardAuthGroup user sto-user MubeardAuthGroup user sto-user https://kubeard.cd Ntps://kubeard.cd Ntps://kubeard.cd</th> <th>● 田戸田 基本信息 周中報名 user 画志 user 画志 user 創銀日同 2.分钟 実現用中角色病症(金原) ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●</th> <th>RA / RE-2004 / User * 102 KKK (1997) SA (1997)</th> <th>AR / MC2028 / USF * 502500 で * 00008 / V3.2.7 (■ 用户相 基本信息 用户相名 USF 2 編集 回题时 2 分钟 REC用PUSA Kubenetes 型部中知道——个名称空词 C ● ● ● ★ USEA + MA REC用PUSA Kubenetes 型部中知道——个名称空词 C ● ● ● ★ USEA + MA ERECR * 1452 * 1452 * 1452 * 1452 * 1452 * 1452 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 *</th>	展升 / 推出現決致 / User ▲ 62/2004 ● 基本信息 ● 用户组 基本信息 用户组 ● 服告 user 服告 user 服告 user 服告 user 服告 user 服告 (金属) 創設的词 2.分钟 经取用户访问 Kubernetes 集影中的某一个名称空问 CI ● 解释命 = 主体发标 = 角色 = Cbute KS KubeardAuthGroup user sto-user MubeardAuthGroup user sto-user https://kubeard.cd Ntps://kubeard.cd Ntps://kubeard.cd	● 田戸田 基本信息 周中報名 user 画志 user 画志 user 創銀日同 2.分钟 実現用中角色病症(金原) ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	RA / RE-2004 / User * 102 KKK (1997) SA (1997)	AR / MC2028 / USF * 502500 で * 00008 / V3.2.7 (■ 用户相 基本信息 用户相名 USF 2 編集 回题时 2 分钟 REC用PUSA Kubenetes 型部中知道——个名称空词 C ● ● ● ★ USEA + MA REC用PUSA Kubenetes 型部中知道——个名称空词 C ● ● ● ★ USEA + MA ERECR * 1452 * 1452 * 1452 * 1452 * 1452 * 1452 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 * 150 *		

图 8-10 绑定集群角色

(3)进入集群概览页,设置访问控制的"第二阶段授权",点击"用户"->"为新 User 授 权",选择用户 user01 ,如图所示

Kuboard		<u>首页 〉 Cloud-K8s</u> [10時] 〉 user A 设力取入页 G			Kubernetes : v1.28.0 Kuboard : v3.5.2.7	🔇 kuboard-admin 👻
🛱 Cloud-K8s		指定名称空间中被援权的 User/Group				
② 集群导入	×	古你全时 请输入名称空间	✓ ⊕ ± #			
□ 累計官理 ■ 名称空间 - 选择	Ť	Users Groups				
合 访问控制	^	User Name 🗢	RoleBinding 数量 💠	ClusterRoleBinding 政문 💠		+ 为新 User 授权
✿ 第一阶段授权	~	system kube-controller-manager	清洗择名称空间	1	新输入 User Name : 下拉选择 手动输入	
角色		system kube-scheduler	请选择名称空间	2	user01 III 请选择用户	×取満 ×确定
ClusterRole Role						
用户 ServiceAccount						
用户						Ğ
用户组						
	《 收起			https://kuboard.cn		

图 8-11 选择授权用户user01

(4) 切换到 namspace01 名称空间,并点击 RoleBinding 后面的 添加 按钮,在
 RoleBinding 页面选择命名空间为 namspace01,关联的 Role 的 name 为 admin,如图 8-12、8-13 所示。

Kuboard	<u>首页</u> 〉 <u>Cloud-K8s</u> [辺絵] 〉 <u>User</u> 〉 user01 ▲ 设为默认页	0			Kubernetes : v1.28.0 Kuboard : v3.5.2.7	kuboard-admin -
∯ Cloud-K8s	User 🗈 😧 user01					
☑ 集群导入 ∨	User user 01 在当前面名称空间中获得的授权 东际空间	0 57	 此页面只 因为如下 	能显示直接授予给 User 的角色/权限,Use 源因,Kuboard 不能在界面上罗列 User 与	rr 还将获得其所在 Group 的角色/权限。 ; Group 之间的映射关系	
□ 集群管理 ~	Hello CELO Hello CEOT		• Kubo	ard 支持多种类型的用户信息库		
🔡 名称空间 - 选择	RoleBinding II 🕢 0 匝 湊加	名称空间级别权限				
合访问控制 个	添加 RoleBinding 可接予名称空间级别的权限	roleKind roleName	apiGroups resou	rces/nonResourceURLs	resourceNames	verbs
第一阶段授权				智无数据		
◎ 第二阶段授权 ^						
	ClusterRoleBinding I 🛛 🖉 🕢 🗈 添加	集群级别权限				
ClusterRole	system:basic-user 角色: system:basic-user	clusterRoleName	apiGroups	resources/nonResourceURLs	resourceNames	verbs
Role	授权给 system:authenticated system:discovery	system basic-user	authorization.k8s.io	setfsubjectaccessreviews setfsubjectrulesreviews		create
	用色: system:discovery 授权给 system:authenticated	system:basic-user	authentication.k8s.io	selfsubjectreviews		create
ServiceAccount	system:publc-info-viewer 角色: system:publc-info-viewer			/healthz /livez		
用户	netova system:autnenticated system:service-account-issuer-discovery	system:public-info-viewer		/readyz		get
用户组	角色: system:service-account-issuer-discovery 授权给 system:serviceaccounts			/version/		
一 可访问时段				/api /api/*		
« #0	a	https://kubo	pard.cn	/apis		

图 8-12 授权用户命名空间

Kuboard	首页 > Cloud-K8s (印刷 > User)	> user01 ♠ 设为默认页 💽		Kubernetes : v1.28.0 Kuboard : v3.5.2.7 Kuboard : v3.5.2.7
贷 Cloud-K8s	RoleBinding			⊗
 ○ 集群导入 □ 集群管理 ■ 各称空间・选择 	* 名称空间 namspace81 * 名称 user01-rolebinding-Sib20		XHR9 ClusterRole / Role apiGroup rbsc.authorization.kBs.io * kind ClusterRole * name antein	
合 访问控制				verbs
✿ 第一阶段授权 ◎ 第二阶段授权	被授权对象 (ServiceAccount, User, Group) 可在一个 RoleBinding 中为多个对象授权 apiGroup rbac.authorization.k8s.io	 通知 ServiceAccount 		
ClusterRole	kind User	ය) 添加 User [] 添加 Group		verbs
Role 用户				create
ServiceAccount				create
用户				get
用户组	预览/编辑 YAML			◇保存
前 可访问时段	< kuz	https:/	رمیہ امورٹ kuboard.cn	

图 8-13 设置用户命名空间的角色

(5) 使用 user01 登录集群,只能选择命名空间 namspace01 ,如图 8-14 所示。

Kuboard	前页 🔺 默认页 📀						• Ku	board : v3.5.2.7	U user01 -
▲ Kubernetes 集群	Kubernetes 集群列表								
② 个人设置 ~	Cloud-K8s 大约 4 小时	Cloud-K8s • 请选择访问集群时所使用的身份	授权用户访问名称空间 🗹 🔵	按名利	擁透 Q	 國示所有名称空间 			
	已就绪 K8S-v1.28.	 使用 ServiceAccount kuboard-ad 	lmin	Star	Name	Phase			
	本地K8s集群	 使用 ServiceAccount kuboard-vie 		☆	• -default	Active			
				☆	• -kube-node-lease	Active			
		 便用 ServiceAccount kuboard-ad 	lmin 扮演 user01	12 ~	• -kube public	Active			
		此时您只具备如下用户(组)的权限,不	具备 kuboard-admin 的权限	ਮ 	 kubezed 	Active			
		用户组 user01		ਮ ਨ	namenare@1	Active			
		A 查看用户在集群中的身份	详情	ਮ ਨ	-namsnace82	Active			
				~ \$	-namspace03	Active			
				☆	-namspace84	Active			
				슈	● -namspace05	Active			
		豆 集群概要	⊻ 集群导入信息						
						∠ 编辑名称空间列表			
						× 取満			
《收起			https://kuboard.cn						

图 8-14 用户登录集群

步骤 4: 为每个用户的命名空间指定独立的 pod 网段

需要为每个用户创建一个地址池,该地址池设置的网段就是用户所在的 pod 网段,之后为命名 空间指定地址池。

```
1 #查看当前存在的IP池
 2 kubectl get ippools.crd.projectcalico.org
 3
 4 #查看地址池的详细信息
 5 kubectl get ippools.crd.projectcalico.org pool-name -o yaml
 6
 7 #创建5个地址池
 8 cat > ippools.yaml <<EOF</pre>
9 ---
10 apiVersion: crd.projectcalico.org/v1
11 kind: IPPool
12 metadata:
13 name: ipv4-ippool01
14 spec:
15 cidr: 192.168.1.0/24
16 ipipMode: Always
17 natOutgoing: true
18 nodeSelector: all()
19 vxlanMode: Never
20
21 ---
22 apiVersion: crd.projectcalico.org/v1
23 kind: IPPool
24 metadata:
25
    name: ipv4-ippool02
26 spec:
27 cidr: 192.168.2.0/24
28 ipipMode: Always
29 natOutgoing: true
30 nodeSelector: all()
31 vxlanMode: Never
32
33 ---
34 apiVersion: crd.projectcalico.org/v1
35 kind: IPPool
36 metadata:
37
    name: ipv4-ippool03
38 spec:
39 cidr: 192.168.3.0/24
40
    ipipMode: Always
41 natOutgoing: true
42
    nodeSelector: all()
```

```
vxlanMode: Never
43
44
45 ---
46 apiVersion: crd.projectcalico.org/v1
47 kind: IPPool
48 metadata:
49
    name: ipv4-ippool04
50 spec:
51
    cidr: 192.168.4.0/24
    ipipMode: Always
52
53
    natOutgoing: true
54 nodeSelector: all()
55
    vxlanMode: Never
56
57 ---
58 apiVersion: crd.projectcalico.org/v1
59 kind: IPPool
60 metadata:
61
    name: ipv4-ippool05
62 spec:
63 cidr: 192.168.5.0/24
64 ipipMode: Always
65 natOutgoing: true
    nodeSelector: all()
66
67 vxlanMode: Never
68 EOF
69 #应用ippools.yaml文件,创建地址池
70 [root@k8s-master ~]# kubectl apply -f ippools.yaml
71 ippool.crd.projectcalico.org/ipv4-ippool01 created
72 ippool.crd.projectcalico.org/ipv4-ippool02 created
73 ippool.crd.projectcalico.org/ipv4-ippool03 created
74 ippool.crd.projectcalico.org/ipv4-ippool04 created
75 ippool.crd.projectcalico.org/ipv4-ippool05 created
76
77 #为命名空间指定地址池
78 kubectl annotate namespace namspace01 "cni.projectcalico.org/ipv4pool
   s"='["ipv4-ippool01"]'
79 kubectl annotate namespace namspace02 "cni.projectcalico.org/ipv4pool
   s"='["ipv4-ippool02"]'
80 kubectl annotate namespace namspace03 "cni.projectcalico.org/ipv4pool
   s"='["ipv4-ippool03"]'
81 kubectl annotate namespace namspace04 "cni.projectcalico.org/ipv4pool
   s"='["ipv4-ippool04"]'
```

```
82 kubectl annotate namespace namspace05 "cni.projectcalico.org/ipv4pool s"='["ipv4-ippool05"]'
```

5、部署并发布 Nginx 测试服务

步骤1:安装配置MetalLB。

修改 kube-proxy 代理模式

Shell
1 #修改两处 2 kubectl edit configmap kube-proxy -n kube-system 3
4 strictARP: true 5 mode: "ipvs" 6
7 #重启kube-proxy代理 8 kubectl rollout restart daemonset kube-proxy -n kube-system

安装 MetalLB,并配置地址池

```
1 #安装负载均衡器MetalLB
 2 kubectl apply -f https://raw.githubusercontent.com/metallb/metallb/v0.1
  4.9/config/manifests/metallb-frr.yaml
 3
 4 #创建对外发布的地址池
 5 vi matellbpool-l2.yaml
 6 -----
 7 apiVersion: metallb.io/v1beta1
 8 kind: IPAddressPool
 9 metadata:
    name: default-ip-pool
10
    namespace: metallb-system
11
12 spec:
13
    addresses:
14 - 172.16.125.151-172.16.125.160
15 ---
16
17 apiVersion: metallb.io/v1beta1
18 kind: L2Advertisement
19 metadata:
20
    name: default-l2
21
    namespace: metallb-system
22 spec:
23 ipAddressPools:
24 - default-ip-pool
25
26 #应用matellbpool-12.yaml文件,创建地址池
27 [root@k8s-master ~]# kubectl apply -f matellbpool-l2.yaml
28 ipaddresspool.metallb.io/default-ip-pool created
29 l2advertisement.metallb.io/default-l2 created
```

步骤2: 创建测试服务

```
1 [root@k8s-master ~]# cat matellbnginx.yaml
 2 apiVersion: apps/v1
 3 kind: Deployment
 4 metadata:
     name: nginx-deployment
 5
     labels:
 6
 7
       app: nginx
 8 spec:
 9
     replicas: 3
     selector:
10
       matchLabels:
11
12
         app: nginx
13
     template:
14
       metadata:
15
         labels:
           app: nginx
16
17
       spec:
18
         containers:
19
         - name: nginx
20
           image: docker.io/nginx:latest
21
           ports:
22
           - containerPort: 80
23
24 ---
25
26 apiVersion: v1
27 kind: Service
28 metadata:
29
     name: nginx-service
30
     labels:
31
       app: nginx
32 spec:
33
     selector:
34
      app: nginx
35
     ports:
36
     - name: nginx-port
37
       protocol: TCP
38
       port: 80
39
       targetPort: 80
40
     type: LoadBalancer
41
42 #查看服务地址
```

```
43 [root@k8s-master ~]# kubectl get svc
44 NAME
                TYPE
                            CLUSTER-IP
                                           EXTERNAL-IP
                                                          PORT(S)
        AGE
45 kubernetes
              ClusterIP 10.96.0.1
                                         <none>
                                                          443/TCP
        9h
46 nginx-service LoadBalancer 10.98.176.99 172.16.125.151
                                                          80:3051
  5/TCP 7m8s
47
```

💡 提醒:

此处发布测试服务主要验证发布服务所使用的地址是 MetalLB 从地址池中分配的 IP 地址,这里分配的 IP 地址是 172.16.125.151。

使用浏览器访问 http://172.16.125.151,可以看到 nginx 服务默认页,如图 8-15 所示。

← ♂ ▲ 不安全 172.16.125.151	② ☆ Q、 点此搜索
	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 <u>nginx.org</u> . Commercial support is available at <u>nginx.com</u> .
	Thank you for using nginx.

图 8-15 访问nginx服务

6、部署并发布 Wordpress 服务

步骤1:用创建的用户 user01 登录 Kuboard,进入命名空间 namespace01。

步骤 2: 创建存储卷声明 PVC,用于挂载 MySQL 的数据目录,点击"存储"->"创建存储卷声明",配置如图 8-16 所示,保存并应用,如图 8-17 所示。

Kuboard		首页 〉 Cloud-	K8s 时间 〉 namspace0	1 (1994) > 存続電声明 🔥 成力変化页 🙃	•	Kubernetes : v1.28.0 uboard : v3.5.2.7	U user01 -
∯ Cloud-K8s		存储卷声明	存储卷声明: namsp	ace01/创建	\otimes		
11 谷桥2590 namspace01 ▼ ▲ 私要 □ 常用服件 ▼ 島用販水	~ ~ ~	今段选择器 标签选择器 自また こ 名	名称20月 「 存録第月 明 ・名称 ・存録类 ・分配模式 ・広写模式 ・の量	amspace01 ayselds cloud+88-offs-1 ● 动参分配 ○ 月能接带力流氓写 ○ 可被多节点只读 ● 可被多节点误写 251		COES & YAML	
 ■ MSRP40 ● 存編 ① 事件 ● 名称空间设置 	Ŭ		存儲整声明 YAML		取消		
	《收起						

图 8-16 创建PVC

Kuboard	首页 > Cloud-K8s (切响) > namspace	01 [[19] > 存储卷声明 A 设为默认页 😯				 Kubernetes : v1.2 Kuboard : v3.5 	8.0 .2.7 U user01 ▼
∯ Cloud-K8s	Ⅰ 存储卷声明列表						
: 名称空间 /	∧ 字段选择器 metadata.name		8				
namspace01 🔻	标签选择器 标签选择器 ピ ❷	□ 添加					
▲ 概要					了解什么是 Persistent	VolumeClaim I 🛿 😧 🔍 查询	∠ YAML
□ 常用操作 、	合 删除						
▲ 应用程序		存储类	容量 状态	Source	创建时间	操作	
	mysqldb	cloud-k8s-nfs-1	Pendin	8	不到 1 分钟		隆回快照图YAML自删除
□ 事件							
〇 名称空间设置							
« #	牧起		https://kuboard.cn				

图 8-17 创建完成

步骤 3: 创建 MySQL 工作负载。

(1) 点击"常规操作"->"创建工作负载",在"基本信息"选项卡中选择工作负载类型,填 写工作负载名称,以及添加标签的配置如图 8-18 所示。

Kuboard	首页 〉 Cloud-K8s (切换	ii > <u>namspace01</u> [初與i > 创建 Deploy	ment			Kubernetes : v1.28.0 Kuboard : v3.5.2.7	U user01 -
✿ Cloud-K8s	創建工作负载 ○ 5	2満 ◎ 保存					
名称空间 へ	基本信息	容器信息 存储挂载 高级设置	服务/应用路由				
namspace01 🔻	基本信息						
▲ 概要	* 工作负载类型	部署 (Deployment) 有状态副本集 (Sta	tefulSet) 守护进程集 (DaemonSet) 定时任务	(CronJob) 任务((dot)		
□ 常用操作 へ	工作负载分层 ♂ ❷	持久层					
从YAML创建	* 工作负载名称	mysql					
创建工作负载	服务描述	请输入服务描述					0/50
从其他工作负载复制	注解	+ 添加注解					
导出工作负载	*标签	k8s.kuboard.cn/layer	db	Ē	k8s.kuboard.cn/name	mysql	±
导入工作负载		app	mysql	1	+ 液加标签		
调整编像版本	选择标签	k8s.kuboard.cn/layer	db	Ē	k8s.kuboard.cn/name	mysql	1
★ 应用程序 ✓		app	mysql				
□ 配置中心 ~		工作负载编辑器默认将 spec.selector.matchLa	bels 字段设置为与 metadata.labels 字段相同的值				
ᇦ 存储	*副本数	- 1 +					
へ			https://kuboard.cn				

图 8-18 设置基本信息

(2)在"容器信息"	选项卡中,	添加工作容器	,设置容器名称	、容器镜像为	registry.cn-		
hangzhou.aliyuncs.com/my-common-images/mysql:8.4.3 ,选择镜像拉取策略、添加用于							
数据库初始化的环境	变量 MYSQL_	ROOT_PASSWORD	MYSQL_DATABASE	MYSQL_USER	MYSQL_PASSWORD		
以及添加容器端口为	3306,配	置信息如图 8-1	.9、8-20所示。				

Kuboard	構成 > Cloud-K8s (20時) > namspace01 (20時) > 含肥 Deployment ・ Kubernets : ・ Kubernet ・ Kubernet	v1.28.0 User01 -
✿ Cloud-K8s	● 設証作免戦 ◎ 敷満 ◎ 教存	
名称空问 へ	基本信息 容識信息 存儀挂載 高级设置 服务应用路由	
namspace01 🔻	□ 添加約時代報報 □ 添加工作容器 工作容器	1
♠ 概要	工作総合 mysql	
	mysql * 容器快歩 registry.cn-hangzhou.aliyuncs.com/my-common-images/mysql:8.4.3	ImagePullSecret 选择镜像合库密码 · · · + 创建
从YAML创建	織像拉取策略 放修拉取新确像(Always) 本地不存在封拉取确像(IfNotPresent) 从不拉取确像(Never)	
创建工作负载	□ 命令/參数 如果填写了 『命令』 字段、将暂代撤缴中的默认 ENTRYPOINT 或 CMD,	
从其他工作负载复制	✓ 环境委量 ✓ 设置客器的环境支量	
导出工作负载	MISQL_NOOT_PASSAGNO = 🕅 🗸 wordpress	•
导入工作负载	MYSQL_DATABASE = @ v wordpress	8
调整镜像版本	MISQLUSER = @ v wordpress	(
▲ 応用程度 >>	MISQL_MASSADRD = @ V wordpress	â (
	+ 名極时 + 配置	
	● 治理論求限制 设置容易的资源需求 此本本語時間等等の計(1)等等時中部計(1)	帮助 ピ ●
9 存储		
へ ★ ## 《 收起	https://kuboard.cn	

图 8-19 设置容器信息

Kuboard	<u>首页 〉 Cloud-K8s</u> [回詞] > <u>namspace01</u> [回詞] > 创建 Deployment	Kubernetes : v1.28.0 Kuboard : v3.5.2.7 Uuser01 ▼
ộ Cloud-K8s	(1)課工作负载 ◎ 数 滴	
名称空问 ヘ	基本信息 容器信息 存谦控载 高级设置 服务应用路由	
namspace01 🔻	▶ 清加机油化器器 □ 清加工作器器 ✓ 环境变量 ✓ 设置容器的环境变量	
▲ 概要	【I作版語】	8
	MYSQL_DATABASE = @ v wordpress	۵
从YAML创建	MYSQL_USER = @ vondpress	ů
创建工作负载	MYSQL_PASSIORD = 🕼 🗸 wordpress	8
从其他工作负载复制	+ 30001 + 配置	
导出工作负载	資源請求限制 設置容易的資源需求 出生就心向市场有所習習業時代以為密慮求原則 / 機構對以	報助 ピ ●
导入工作负载	✓ 投资容易依据日	
调整镜像版本		
★ 应用程序	健康检查 意时检查容器健康状况	
白 配置中心 🗸 🗸	生命兩期回調 容器的生命用期回调	税助 🖸 🖌
₿ 存储	寄議会全上下文 安全上下文(Security Context)可以限制不可信智器的行为,保护系统和其他智慧不受共影响。	報助 ピ ●
^ ≪ 804		1
《收款	https://kuboard.cn	

图 8-20 设置容器信息

(3) 在"挂载存储"选项卡中,定义存储卷名称、选择存储卷声明以及挂载到容器内的路径, 配置信息如图 8-21 所示。

Kuboard	前方 〉 <u>Cloud-K8s</u> (1038) > <u>namspace01</u> (1038) > 含規 Deployment	Kubernetes : v1.28.0 Kuboard : v3.5.2.7	ser01 🔻
∲ Cloud-K8s	创建工作负载 ◎ 取満 ◎ 载存		
器 名称空间 ∧	基本信息 容識信息 存储控制 高级设置 服务应用路由		
namspace01 🔻	数据卷 定义数据卷,并将其建载到工作负载的容器中 数据卷 ℃ ●		
♠ 概要			
□ 常用操作 へ			
从YAML创建	自動除 存種標本到 能計目录(EmplyOir) HostPath NFS 配田字典 敬文		
创建工作负载	74725		
从其他工作负载复制	Triater → → → → → → → → → → → → → → → → → → →		
导出工作负载			
导入工作负载	終数振程注载到容器 ○ B新辛齢[20]		
调整镜像版本			
★ 应用程序 ~	10 /var/lib/mysal 民族 故与 @		<
□ 配置中心 ∨	LIGBNJ Subrath / Subratheopr, SLYANE, WHEN BERNENHENSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBERHESSUBE		
8 存储)
	North Associate		

图 8-21 设置存储挂载

(4) 在"服务/用于路由"选项卡中,点击"服务"复选框,展开服务发布,设置服务类型为 ClusterIP,设置 Port和 targetPort的值为 3306 ,配置信息如图 8-22 所示,点击保存、应 用,等待镜像拉取并成功运行,如图 8-23 所示。

Kuboard		首页 〉 Cloud-K8s [切]	甸 > namspace01	[[切换] > 创建 Dep	ployment • Kubernetes : v1.28.0 • Kubernet : v3.5.2.7	user01 🔻	
¢	Cloud-K8s		创建工作负载 ◎	取満 ⊘保存			
	名称空间	^	基本信息 容器信	· 傳 存储挂载	高级设置	服务应用器由	
	namspace01 🔻		☑ 服务	> 服务发布			
	♠ 概要			*名称	mysql		
	C) 常用操作	~		☑ 元数据	へ 注解/标签		
				🕢 服务定义	~ 服务定义		
	0/26 T //- 0 45				选择器	kils.kuboard.cn/layer db 🖹 (kils.kuboard.cn/name mysql	
	初建工作页载					app mysql	
	从其他工作负载复制				服务类型	Headless (None) O ClusterIP NodePort ExternalName LoadBalancer	
	导出工作负载				未就绪发布		
	导入工作负载				端 囗	ps8txw TCP 3386 3396 ⊕ + 7600380	
	调整镜像版本				会话保持	None ClientiP	
	★ 应用程序	~			externalIPs	+ 1650 externallP	
		Ť.	应用路由	应用路由			
/	₿存储						
		《收起				https://kuboard.cn	

图 8-22 服务设置

Kuboard	前页 > <u>Cloud-K8s</u> (2018) > <u>namspace01</u> (2018) > 使預別表 > mysql ▲ 昭元和33 20	Kubernetes : v1.28.0 Vabeard : v3.5.2.7 User01 ▼
∯ Cloud-K8s	Deployment 区 编辑 预选编辑 YAML 2. 清整确例版本 创 停缩 📮 日志 /终端 回 复考	M C/CD 集成 C 業局 合制除
- 名称空间	- 元数据 <u>运行时</u> 服务 应用路由	
namspace01 🔻	✓ 展开 伸縮: (1/1) 目前伸縮 ◎ 皆得更新 (3 1个关联事件) ±	安被 Kuboard 霰件
▲ 概要	历史版本 副本集 mysql-617545ffc6 所管理的容器组 当前副本集致 1 最大历史副本集致 10 点 编 編 回 删除该副本集下所有的容器组	Reason Time Count Message Slanded 不明1分钟 1 Started container mysql ×
D5 带用增作	● 已创建: 不到 1 分钟 mysq1-6f7545ffc6-469xk Rea	ady Created 不到1分钟 1 Created container mysql ×
	mysq1-6f7545ffc6 #1 不到1分钟 C 192.168.1.14 附172.16.125.103	Pulled 不到1分钟 1 Successfully pulled image "registry.cn-hangzhou.aliyuncs.com/my-common-images/mysq_×
★ 应用程序	利益:1 当前:1 飲坊:1 回 YAML 目 副 Mar Reason Time Count Message	Pulling 小到 分钟 1 Pulling image "registry.cn-nanginou.aliyuncs.com/my-common-images/mysql:6.4.3" × Scheduled 不到 1分钟 1 Successfully assigned namspace81/mysql:6f7545ffc6-469xk to k8s-worker2 ×
工作负载	Successful 不평 1 @ Created ×	已显示 5 条, 共 5 条
容器组 服务	已显示14。共1条	全接 metrics-scraper 用于在 Kuboard 中型示一級対局以内的 CPU内分布時後
应用路由		
高 配置中心	· · ·	SA就用 mysql-6f7945ftc6-469xk 不到1分钟
		所在节点 容器组P 状态 k8s-worker2 (172.16.125.183) 192.168.1.14 Running
₩ 17 10		● 已阅焼 2025-03-22 16:37:02 ● 已初始化 2025-03-22 16:37:02
♀ 事件		● 容器已減損 2025-03-22 16:37:05 ● 容器组已減損 2025-03-22 16:37:05
〇 名称空间设置		と 安装 Kuboard 寄件
		🚓 📾 mysql 🖬 🖬 🖬 🖓 🙆 🖓
	《收起 https://kuboard.c	л

图 8-23 运行的Pod

步骤 4: 创建 wordpress 工作负载并对外发布服务。

(1) 创建 wordpress 工作负载,配置信息如图 8-24 所示。

Kuboard	首页 〉 Cloud-K8s (初編	a 〉 <u>namspace01</u> [初始] 〉 创建 Deploy	ment			Kubernetes : v1.28.0 Kuboard : v3.5.2.7	U user01 •
🔅 Cloud-K8s	创建工作负载 ◎ 日	21消 ◎ 保存					
# 名称空间 へ	基本信息	容器信息 存储挂载 高级设置	服务/应用路由				
namspace01 🔻	基本信息						
▲ 概要	* 工作负载类型	部署 (Deployment) 有状态副本集 (Sta	tefulSet) 守护进程集 (DaemonSet) 定	时任务 (CronJob) 任务 (Job)			
□ 常用操作 へ	工作负载分层 🖸 🔮	展现层					
从YAML创建	* 工作负载名称 wordpress						
创建工作负载	服务描述	清逾入服务规定					0/50
从其他工作负载复制	注解	+ #5002846					
导出工作负载	*标签	k8s.kuboard.cn/layer	web	Ê	k8s.kuboard.cn/name	wordpress	ŧ
导入工作负载		app	wordpress	1	+ 添加标签		
调整镜像版本	选择标签	k8s.kuboard.cn/layer	web	Û	k8s.kuboard.cn/name	wordpress	Û
★ 应用程序 ~		app		wordpress			
□ 配置中心 ~		した実態編輯器数以将 spec selector matchLabels 学校设置为与 metadata labels 学校相同的值					
8 存储	*副本数	- 1 +					
∧ ★ ## ≪ 收起			https://kuboard	.cn			

图 8-24 设置基本信息

(2) 添加工作容器,设置连接数据库环境变量和容器端口,配置信息如图 8-25 所示。

Kuboard	Migi > Cloud-Kits Image > Billits > Billits </th <th>: v1.28.0 : v3.5.2.7</th> <th>U user01 •</th>	: v1.28.0 : v3.5.2.7	U user01 •
∯ Cloud-K8s	创建工作负载 ◎ 取消		
名称空间 へ	基本信息 容器信息 存储挂载 高级设置 服务应用路由		
namspace01 🔻	 D 法加工作部署 - 容器現象 ・ 報告 	ImagePullSecret	1 41 28
▲ 概要	工作容器 ・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・・	753+HE186 (2+2+00.1v3	- H3 XA
	命令/参数 如思境写了【命令】李良、将器代娘像中的放队 ENTRYPOINT 或 CMD。		
从YAML创建 创建工作负载	✓ 环境交量 ✓ 设置容器的环境更量		
从其他工作负载复制			
导出工作负载	MORDMESS_UM_DOSH MORDMESS MORDMESS		8
导入工作负载	NORDPRESS_DB_NAVE = 個 wordpress		
调整镜像版本	+ 名箴对 + 配置		
★ 应用程序 >>	资源请求限制 设置容器的资源需求 此生称空间中没有配置容器的飲以资质资素限制 2. 编编数以		森助 ピ 🛛
	✓ 容然流口 ✓ 投資容素的執口	_	
ᇦ 存储	□ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □ □		
◇ ★ ₩ 《 收起	https://kuboard.cn		

图 8-25 设置容器信息

(3)保存并应用,待成功创建 wordpress 工作负载,如图 8-26 所示。

Kuboard	首項 > Cloud-K8s [1288] > namspace01 [1288] > 部語別表 > wordpress A 紀力部以页 ②	• Kubernetes : v1.28.0 • Kubernet : v3.5.2.7 Uuser01	÷
Ø Cloud-K8s	Deployment び 編載 预定 編載 YAML 企 環路限金版本 自 伸縮 □ 日志 / 终請 回 复制	○ 単紀 白 割除	
# 名称空间	へ 元数据 <u>运行时</u> 服务 应用路由		
namspace01 🔻	● 展开 伸缩: 〈 1/1 〉 自动伸缩 ● 暂停更新 じ 1 个关联事件 ≥ 安枝	を Kuboard	
▲ 概要	历史版本 副本集 wordpress-6581669989 所管理的容器组 当前副本集数 1 最大历史副本集数 10 ℓ 编 編 回 删除该副本集下所有的容器组	Reason Time Count Message Started 자편 1 Started container wordpress	×
	已创建: 不到 1 分钟 wordpress-658 f669989-lrgtx Ready	Created 不到1分钟 1 Created container wordpress	×
	wordpress-658f669989 #1 不到1分钟 C 192.168.1.128 # 172.16.125.185	Pulled The 1 Successfully pulled image "registry.cn-hangzhou.aliyuncs.com/my-common-images/word_	×
✓ 应用程序	林型:1当前:1就結:1 BYAU B 目除 Reason Time Count Message	Pulling 不到1分钟 Pulling image "registry.cn-hangzhou.aliyuncs.com/my-common-images/wordpressilatest" Scheduled 不到1分钟 1 Successfully assigned namspace@1/wordpress-658f669999-lrqtx to k8s-worker4	×
工作负载	Successful <u>不평비 1 21 여</u> 1 Created ×	已显示 5 条, 共 5 条	
容器组	已靈示1處,共1象	安装metrics-scraper 用于在 Kuboard 中型示一般时间以内的 CPU内疗曲线 用于在 Kuboard 中型示一般时间以内的	
服务			Ш
应用路由			
		容器组 wordpress-6581669989-irqtx 不到1分钟	
L HDE-H-D	*	所在节点 容器组印 状态	4
ᇦ 存储		k8s-worker4 (172.16.125.105) 192.168.1.128 Running	
♀ 事件		● 已開就 2025-03-22 17:58:52 ● 已別取化 2025-03-22 17:58:52 ● 含菜已受給 2025-03-22 17:58:52 ● 含菜用豆塗瘡 2025-03-22 17:58:52	
〇 名称空间设置		生 安装 Kuboard 副件	
		🚓 🗱 wordpress 🖬 🖬 🖬 🖬 🕼 🔘 🔘	
	枚起 https://kuboard.cn		

图 8-26 wordpress工作负载

(4)发布 wordpress 服务。在"服务"页面,点击 YAML ,通过 yaml 文件部署服务,如图 8-27 所示。

Kuboard	首页 > Cloud-K8s Inthei > namspace01 Inthei > 服务列表 ▲ 设先取认页 の	• Kubernetes : v1.28.0	U user01 -
☆ Cloud-K8s	从 YAML 创建 此功能与 导入工作负载 ♂ ● 功能的区别是,此处不对 YAML 做任何预处通。	8	
 器 名称空间 namspace01 ▼ ● 截要 □ 常用操作 	1 aptiverion: v1 2 kind: Service 3 metsdata: 4 name: wordpress 5 namespace: namespaceO1 6 label:si: 8 spec: 9 solector: 9 solector: 10 ap: wordpress 11 ports: 12 name: wordpress 12 ports:		Q 호함 순. YAML
 ★ 多用程序 工作失载 容器组 服务 应用器曲 ● 客器地会 	Li pressoli Lu li port li port li targelont: 80 15 typi Loudbalancer		YAML 合動除 YAML 合動除
 □ ##4 ○ 本件 ○ 本件 ○ 本体のの設置 	ERFYANLS(F	×取勝 く鶴辺	

图 8-27 用yaml文件部署服务

1 apiVersion: v1 2 kind: Service 3 metadata: 4 name: wordpress 5 labels: 6 app: wordpress 7 spec: 8 selector: 9 app: wordpress 10 ports: 11 - name: wordpress protocol: TCP 12 port: 80 13 targetPort: 80 14 15 type: LoadBalancer

💡 提醒:

通过 Kuboard 界面发布的服务类型只有 Headless、ClusterIP、NodePort。所以 LoadBalancer 类型的服务只能使用 yaml 文件创建。

(5)发布完成后,点击 wordpress 服务后面的 YAML 查看分配的 IP 地址,这里分配的 IP 地址 是 172.16.125.153,如图 8-28 所示。

Kuboard	首页 > Cloud-K8s rimer > namspace01 rimer > 部務列表 A 设大歌以页 👩	Kubernetes : v1.28.0	U user01 -
🔅 Cloud-K8s	Service		
各称空向	魔動影響の使 2000時間的 VAML 回 除痛 status 学校 9 name : wordpress 1 name: space1 1 name:space1 1 namespace1 1 namesp	双面后均屬機 TAML	
namspace01 🔻	12 spc: 13 spc: 13 allocationdBlancerModePorts: true		
♠ 概要	10 CulsterP1 10.99.100.129 11 CulsterP19: 10 - 10.99.150.129		Q 查询 《 YAML
□ 常用操作	27 externalInstiticPolicy: Cluster is internalInstiticPolicy: Cluster 19 ipFamilies:		
★ 应用程序	20 - 1Pv4 1 ipFmilyPolicy: SingleStack 22 ports:		
工作负载	23 - name: wordpress 24 nodePort: 32378 25 port: 88		YAML 创 删 除
容器组	26 protocal TCP 27 targetPort: 80 28 selector:		YAML 自删除
服务	20 app: wordpress 30 sessionAffinity: None 31 type: LoadBalancer		
应用路由	22 Status: 33 LoodBalancer: 34 ingress:		
高 配置中心	35 - ip: 172.16.125.153 36 37		(
⊜ 存储			
□ 事件 ● <i>@</i> ??????		× wa	
o skence			
	《 收起 https://wboard.cn		

图 8-28 查看分配的服务地址

(6)初始化 WordPress。浏览器访问 http://172.16.125.153,按照提示安装 WordPress,如 图 8-29、8-30 所示。



图 8-29 安装WordPress

🔹 🔲 🕔 仪表盘 ‹ demo	o — WordPress × +						- O	×
← C ▲ 不安全	172.16.125.153/wp-admi	n/			ē 💬	□ Q 点此搜索	☆ (頭新 %) …	0
(1) A demo ↔ 4 🛡	0 十 新建						總好 , admin 显示选项 ▼ 帮助 ▼	٦ _أ
首页	WordPress 6.7.2 现已可用	请立即更新。						
^{更新} 1	义表盘							
★ 文章 5) 媒体 ■ 页面	¥.	r仰使用 Wo	rdPress!				★ 不再显示	
♥ 评论	道	田了解 6.7.1 版本。						
No 1764								
▶ 用户								
■ 1.0								
11 设置	使用区块和区块样板创作丰富的内容			使用区块主题定制整个站点使用样式变更站点的外观和风格				
• 收起菜单	2	区块样板是预先配置好的区块布局。; 在极短时间内创建新页面。	通过区块样板获得灵康或	获得灵感或 上至页眉下至页脚,整个站点都可递过区块获区块样板进行设 计。			目新的调色盘或字体发挥创意!	
	新页面		打开站点编辑器			编辑样式		
	站点健康状态	~ ~ *	快速草稿	^ ¥ *				
	站在健康检查会自动定期运行承取得有 共无法自己。		标题					<
			<u>А</u> ра		拖动模块至此		拖动模块至此	
	概范	~ ~ *	在想些什么?					
	★ 1 篇文章	📕 1 个页面						
	■ 1 學评论					11		

图 8-30 使用WordPress

七、实验讲解

本实验配套讲解视频,访问课程学习平台。

八、实验考核

实验考核为【实验随堂查】。

实验随堂查:每个实验设置3-5考核点,学生现场进行演示和汇报讲解。

1、考核点

考核点1:完成NFS服务器搭建。(20分)

考核点 2: 完成为 K8s 集群添加 NFS 共享存储。(30分)

考核点3:创建用户和命名空间并设置用户访问范围。(20分)

考核点 4:完成在 K8s 集群上部署数据持久化的 WordPress 服务。(30分)

2、考核方式

以实验小组为单位进行考核,每个小组由1位同学进行实验成果汇报,小组其他成员回答教师 提问。根据汇报和答疑情况,对小组成员进行逐一打分。

由教师进行评分。