**集群管理手册**

**版本 4.2.0**

目录

[一、产品概述 1](#_Toc29984)

[1.1产品简介 1](#_Toc5704)

[1.2产品版本 1](#_Toc12900)

[1.3操作系统 1](#_Toc11459)

[1.4功能简介 2](#_Toc10560)

[1.5术语图形界面元素引用约定 3](#_Toc10732)

[二、登录 5](#_Toc23214)

[三、集群管理 6](#_Toc12424)

[3.1集群信息 6](#_Toc13641)

[3.1.1集群列表 6](#_Toc6150)

[3.1.2创建集群 6](#_Toc20737)

[3.1.3查看日志 21](#_Toc9706)

[3.1.4创建非高可用的实验集群 21](#_Toc32082)

[3.1.5删除和释放资源 22](#_Toc13216)

[3.1.6检查已删除集群 23](#_Toc24146)

[3.1.7使用相同配置重建集群 23](#_Toc11227)

[3.1.8从集群中移除工作节点 24](#_Toc30482)

[3.1.9添加工作节点至集群 25](#_Toc25576)

[3.1.10集群备份及恢复 26](#_Toc13349)

[四、节点信息 34](#_Toc14494)

[4.1节点信息 34](#_Toc24400)

[4.1.1节点列表 34](#_Toc21462)

[4.1.2节点详情 34](#_Toc28068)

[4.1.3加入集群 35](#_Toc26250)

[4.1.4禁用节点 35](#_Toc25498)

[五、配置文件 35](#_Toc17679)

[5.1系统所使用的端口 35](#_Toc9966)

[5.2配置文件详细说明 37](#_Toc32267)

[5.2.1server端配置 37](#_Toc5178)

[5.2.2client端配置 45](#_Toc2570)

[六、常见故障 46](#_Toc7857)

[6.1工作节点加入集群失败 46](#_Toc24809)

[6.2集群搭建完成，跨节点的pod无法互相访问 47](#_Toc12902)

[6.3升级失败 47](#_Toc27505)

[6.4使用限制与注意事项 48](#_Toc5627)

[6.4.1云平台的挂载型设备的上限 48](#_Toc15053)

[6.4.2网络带宽 49](#_Toc11959)

[6.4.3节点数量 49](#_Toc31946)

[七、查看日志 49](#_Toc25165)

[7.1获取错误操作ID 50](#_Toc6913)

[7.2查看日志ssh 52](#_Toc27613)

[八、升级管理 52](#_Toc9302)

[8.1准备升级依赖 53](#_Toc26098)

[8.1.1准备依赖的软件包 53](#_Toc23415)

[8.1.2准备依赖的镜像 56](#_Toc6544)

[8.2版本配置 57](#_Toc23097)

[8.2.1创建版本配置 58](#_Toc25230)

[8.2.2编辑版本配置 61](#_Toc30989)

[8.2.3删除版本配置 61](#_Toc15305)

[8.3升级计划 62](#_Toc25308)

[8.3.1创建升级计划 62](#_Toc23034)

[8.3.2升级 63](#_Toc16655)

[九、用户管理 64](#_Toc12851)

[9.1用户 64](#_Toc2192)

[9.1.1创建用户 65](#_Toc29007)

[9.1.2编辑用户 66](#_Toc30717)

[9.1.3删除用户 66](#_Toc21468)

[9.2角色 67](#_Toc6260)

[9.2.1创建角色 67](#_Toc2625)

[9.2.2编辑角色 69](#_Toc23951)

[9.2.3删除角色 69](#_Toc29489)

[十、平台配置 70](#_Toc9880)

[10.1平台默认参数设置 70](#_Toc1711)

[10.2证书管理 70](#_Toc12791)

[10.2.1更新证书 71](#_Toc24105)

[10.2.2上传新的证书（证书有效） 72](#_Toc23906)

[10.2.3上传新的证书（证书已失效） 72](#_Toc27255)

[十一、pod网段的计算 73](#_Toc8294)

[11.1Calico IPAM 73](#_Toc11968)

[11.2Calico 网络模式 73](#_Toc23601)

[11.3Calico 网络模式之节点数量 74](#_Toc20367)

[11.4硬件资源 75](#_Toc13009)

[11.5建议规模 76](#_Toc32679)

[11.6 平台 dns 配置和使用 76](#_Toc26945)

[11.6.1创建域名组 77](#_Toc24658)

[11.6.2添加域名和地址解析 78](#_Toc13045)

[11.6.3删除地址解析 78](#_Toc22842)

[11.6.4删除域名 79](#_Toc31439)

[11.6.5删除域名组 79](#_Toc19479)

[十二、管理员须知 79](#_Toc7321)

[12.1 关于"从错误位置继续"并不是万能 79](#_Toc27314)

[12.2 关于默认存储的问题 80](#_Toc23421)

# 一、产品概述

## 1.1产品简介

CaaS是一个容器云管理平台，可以纳管和编排多达 10000 个服务器节点，并可以同时支持x86和arm架构的服务器。用户可以通过该平台方便地部署和管理多套 k8s 集群，执行集群扩缩容和版本升级等运维操作。

平台提供了对接各种存储的包括NFS，ceph RBD，Openstack Cinder等原生集成。此外，平台提供了在被纳管的k8s集群上部署标准的 "日志收集" "监控报警" 等功能。在网络方面，平台给到用户非常大的灵活度，其提供的overlay vxlan、overlay IP-in-IP以及 BGP(ibgp) 模式，方便用户在不同的主机和网络环境中部署集群，打通pod的网络，并且用k8s集群和其他业务的互动。

## **1.2产品版本**

本文档针对的软件及其版本为：

|  |  |
| --- | --- |
| **产品名称** | **产品版本号** |
| **CTSI CaaS** |  |

## **1.3操作系统**

本平台所支持的操作系统为Centos-7.x

## **1.4功能简介**

本产品是一个容器云管理平台，具有以下功能：

|  |  |  |
| --- | --- | --- |
| 模块 | 功能 | 描述 |
| 管理主机节点 | 禁用节点 | 通过界面的方式禁止该节点被任何集群使用(仅当该节点没有被任何集群占用时) |
| 启用节点 | 通过界面的方式允许该节点被搭建集群(仅当该节点被警用时) |
| 管理集群 | 创建集群 | 通过界面的方式创建的集群 |
| 配置附加组件 | 在安装集群的同时选择需要安装的组件 |
| 动态的添加/移除 工作节点 | 在不影响业务运行的情况下，添加或者安全的移除一个或多个工作节点 |
| 销毁集群 | 销毁集群并移除所有集群节点上的 k8s 相关的数据: 镜像,容器,配置文件，网络设备等 |
| 准备新 k8s 版本 | 当社区有新版本时,通过这个功能,配置所有新版本的依赖的软件包,如：对应 kubelet 的 rpm 包 |
| 在线升级 | 升级集群到指定的版本 |
| 升级管理 | 定制升级的版本 | 配置可用于升级的版本的基本信息,包括版本号,升级需要的依赖包等等 |
| 检查集群是否符合升级条件 | 通过自动检测指定集群是否已经准备好了升级必要的依赖 |
| 升级集群 | 升级指定的集群 |
| 证书管理 | 查看证书有效期 |  |
| 查看证书有效期 |  |
| 用户管理 |  |  |

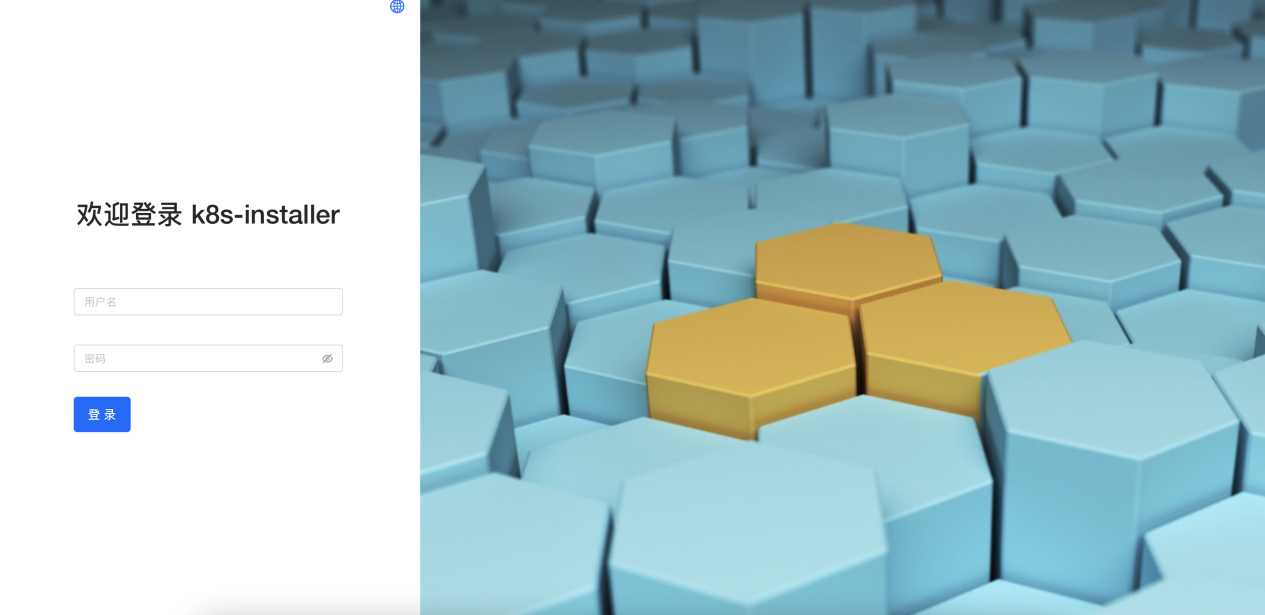
## **1.5术语图形界面元素引用约定**

|  |  |
| --- | --- |
| **格式** | **意义** |
| **“”** | 带双引号“”的格式表示各类界面控件名称和数据表，如单击“确定”。 |
| **>** | 多级菜单用“>”隔开。如选择“虚拟资源>云主机”，表示选择“虚拟资源”菜单下的“云主机”子菜单项。 |
| **Bastion/server** | 服务器 |
| **client** | 客户端 |
| **K8s** | kubernetes |
| **CTSI CaaS** | CTSI CaaS平台 |

# 二、登录

CTSI CaaS容器平台采取账号+密码的验证方式登陆。

注意，登陆CTSI CaaS容器平台登录页面可能会出现的以下错误提示：

* “请输入用户名/请输入密码”：未输入用户名或密码。
* “登录失败”：用户名或密码输入错误。

# 三、集群管理

## **3.1集群信息**

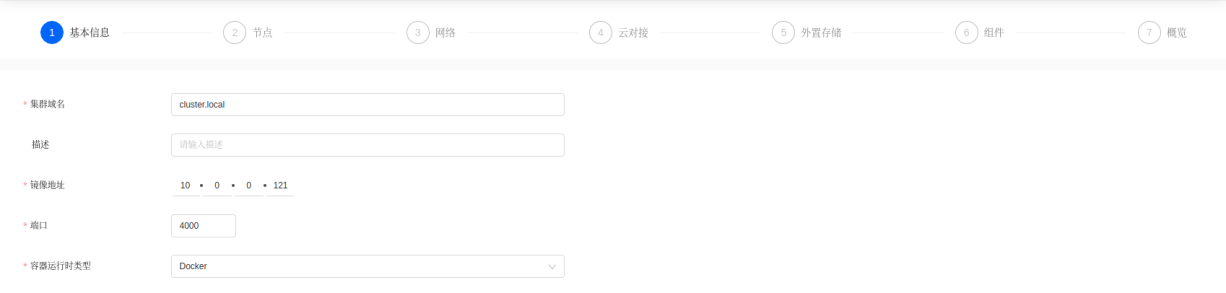
### 3.1.1集群列表

用户可以通过点击主面板右侧菜单“集群管理>集群信息”进至集群管理界面，查看集群列表。可查看集群名称、描述、控制节点数量、工作节点数量、状态、创建时间。



### **3.1.2创建集群**

点击左上角的“创建集群/create cluster”按钮来进入引导安装界面。

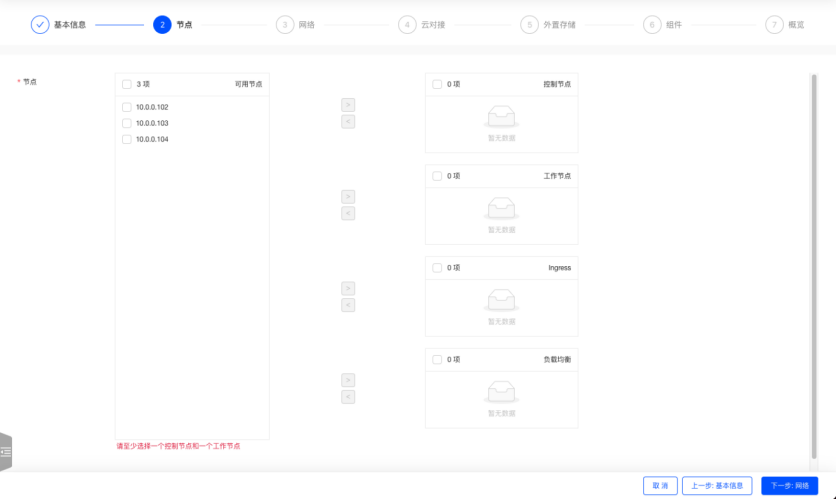
操作步骤：

* 基本信息：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 作用说明 | 验证 | 是否可以缺省 | 可选值 | 默认值 |
| 集群域名/Cluster name | 集群的域名会在搭建完成的集群中设置 kubelet参数 cluster-domain | DNS名 | 否 | N/A | cluster.local |
| 描述/Description | 集群的描述或别名 | N/A | 是 | N/A | 空 |
| 镜像地址/Private registry address | 安装过程中使用的镜像仓库地址 | N/A | 否 | N/A | N/A |
| 容器运行时/Container Runtime | k8s使用的底层容器运行时,目前只支持 docker | N/A | 否 | N/A | Docker |

* 节点配置：

界面的左侧列表中列出所有“可以使用的节点(没有被停用或者没有被其他集群占用的节点)”，右侧展示着整个集群的节点的配置，根据需求，用户可以安装 “实验环境的集群(all in one 或者单 master多worker)”或者“高可用的集群(3 master,多 worker）“。

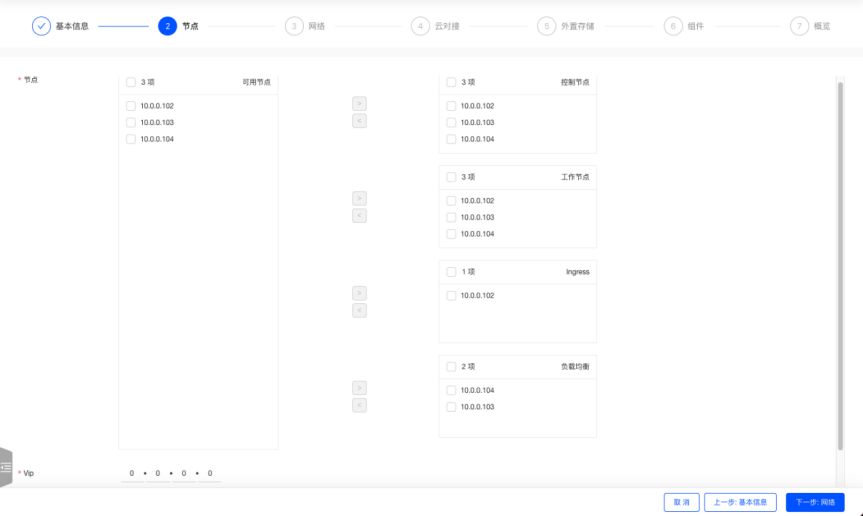


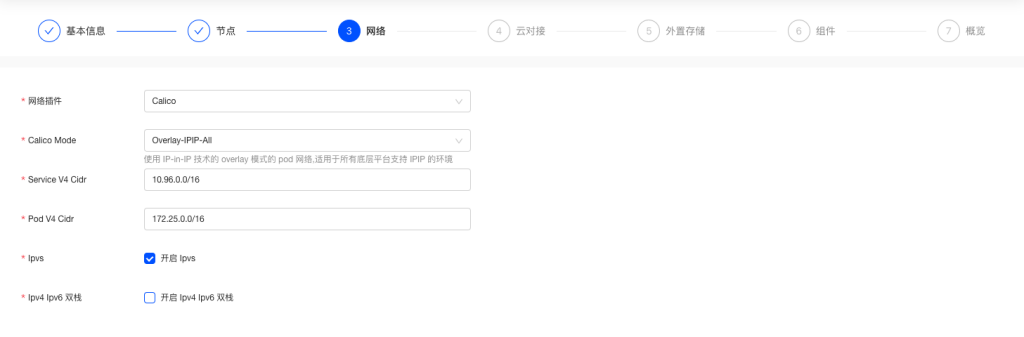
1. 在左侧中选中3个节点,然后点击 ">" 将节点加入 master（控制节点） 中。
2. 在左侧中选中3个节点(包括之前的master的意义是：master也当作工作节点来使用),然后点按钮 ">" 将节点加入 worker（工作节点） 中。
3. 在左侧节点列表中选择之前选择的3个 master ,然后点击 ">" 将节点加入 ingress 中。
4. 如果用户需要配置 "外部负载均衡" ,可以选择不和ingress冲突(ingress 和负载均衡不能共享节点) 的节点加入到 "负载均衡" 的输入框中,并在左下角输入 "Vip" 。

注意：

"Vip" 必须是一个不存在的IP，且用户需要确保物理链路是可达的，因为系统在安装之初是无法检测IP的可达性的。

为保证系统安全，"日志收集(EFK)" 的身份认证依赖于ingress的功能，如果用户不设置ingress的 "日志收集(EFK)" ，将没有基础身份认证，并且可以被任何人直接访问。



* 网络配置：

在“节点配置”界面点击“下一步”进入容器网络配置界面：

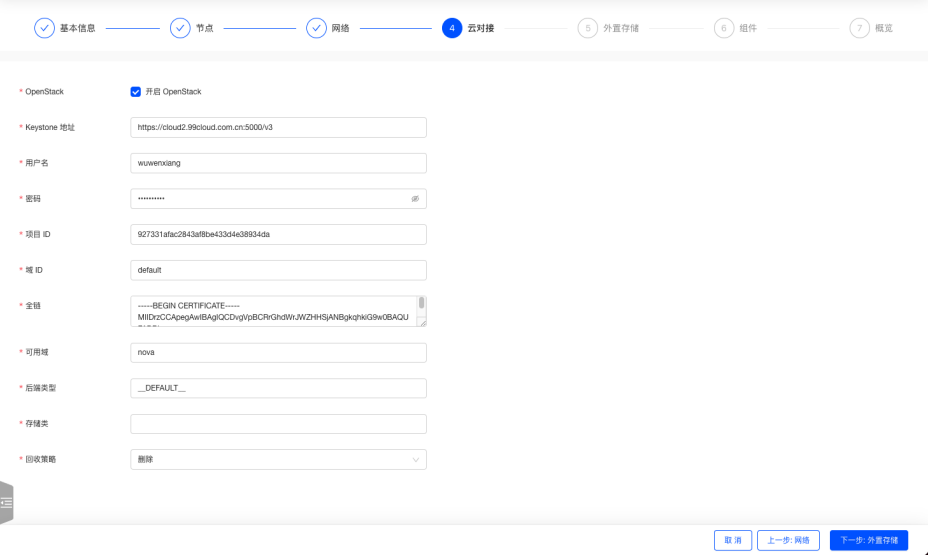
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 作用说明 | 验证 | 是否可以缺省 | 可选值 | 默认值 |
| 网络插件/Plugin | 集群使用的网络插件,目前仅支持 calico | N/A | 否 | N/A | Calico |
| 服务网段ipv4/Service V4 Cidr | k8s集群的IPv4的cluster id所在网段 | CIDR | 否 | N/A | 10.96.0.0/16 |
| pod网段ipv4/Service V4 Cidr | k8s集群pod的IPv4所在网段 | N/A | 否 | N/A | 172.25.0.0/16 |
| 开启IPvs/IPvs | 是否开启IPvs | N/A | 否 | 是/否 | 是 |
| 双栈/dual stack | 是否IPv4，IPv6 双栈 | N/A | 否 | 是/否 | 否 |

* 云对接：

在“网络配置”界面点击“下一步”进入云对接界面。

云对接主要是用来和各种云平台对接，当系统底层的主机运行在 Openstack云平台上时，就可以使用其与底层Openstack来做对接，方便使用 Cinder的存储。

注意：如果用户没有Openstack环境或者想部署一个实验环境，可取消勾选开启Openstack。



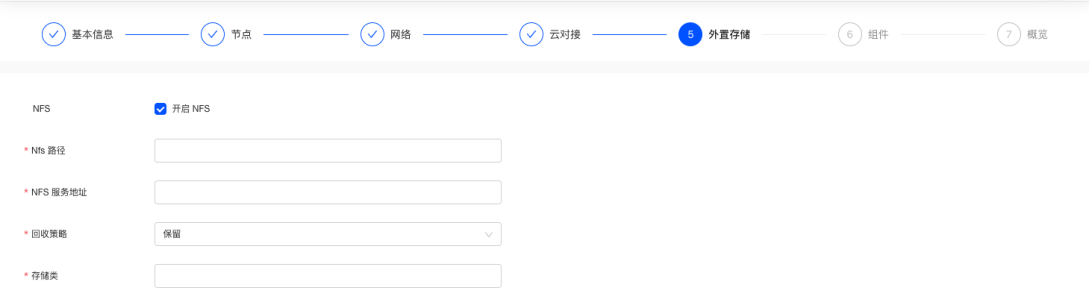
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 作用说明 | 验证 | 是否可以缺省 | 可选值 | 默认值 |
| 开启Openstack | 是否打开Openstack 对接 | N/A | 否 | 是/否 | 否 |
| keystone 地址/keystone address | Openstack 对接地址比如:http(s)://openstack:5000/v3 | url | 否 | N/A | 根据现场设置 |
| 用户名/username | Openstack的keystone中的用户名,用来和Openstack对接的用户名 | N/A | 否 | N/A | 根据现场设置 |
| 密码/password | Openstack的keystone中的用户名的密码,用来和Openstack对接的用户名 | N/A | 否 | N/A | 根据现场设置 |
| 项目 ID/Project Id | Openstack的keystone中的租户的id,用来和Openstack对接的租户id | N/A | 否 | N/A | 根据现场设置 |
| 域ID/Domain Id | Openstack的keystone中的域的id,用来和Openstack对接的域id | N/A | 否 | N/A | 根据现场设置 |
| 全链/Full Chain | 当keystone 的访问地址为https协议的时候,就需要配置受信的链 | N/A | 否 | N/A | 根据现场设置 |
| 后端类型/Cinder Backend | 对应 cinder volume type list 中的一个存储后端 | N/A | 否 | N/A | 根据现场设置 |
| 可用域/Availability Zone | 对应Openstack中的Availability Zone | 英文字符 或者英文字符-英文字符 | 否 | N/A | cinder-csi |
| 存储类/Storage Class Name | 完成对接后默认创建对接 cinder 的 storage class 对象的名字 | 英文字符 或者英文字符-英文字符 | 否 | N/A | cinder-csi |
| 回收策略/Reliam Policy | 使用 cinder-csi 的 pod 被删除磁盘回收机制,delete/删除 = pod 删除后删除磁盘,保留/retain = pod 删除后保留磁盘 | N/A | 否 | Delete(删除)/Retain(保留) | Delete(删除) |

* 外置存储：

在“云对接”界面点击“下一步”，可进入外置存储配置界面。

外接存储主要是用来和外部存储集群，比如对接Ceph、Glusterfs、NFS等,当前版本仅支持外接NFS.。

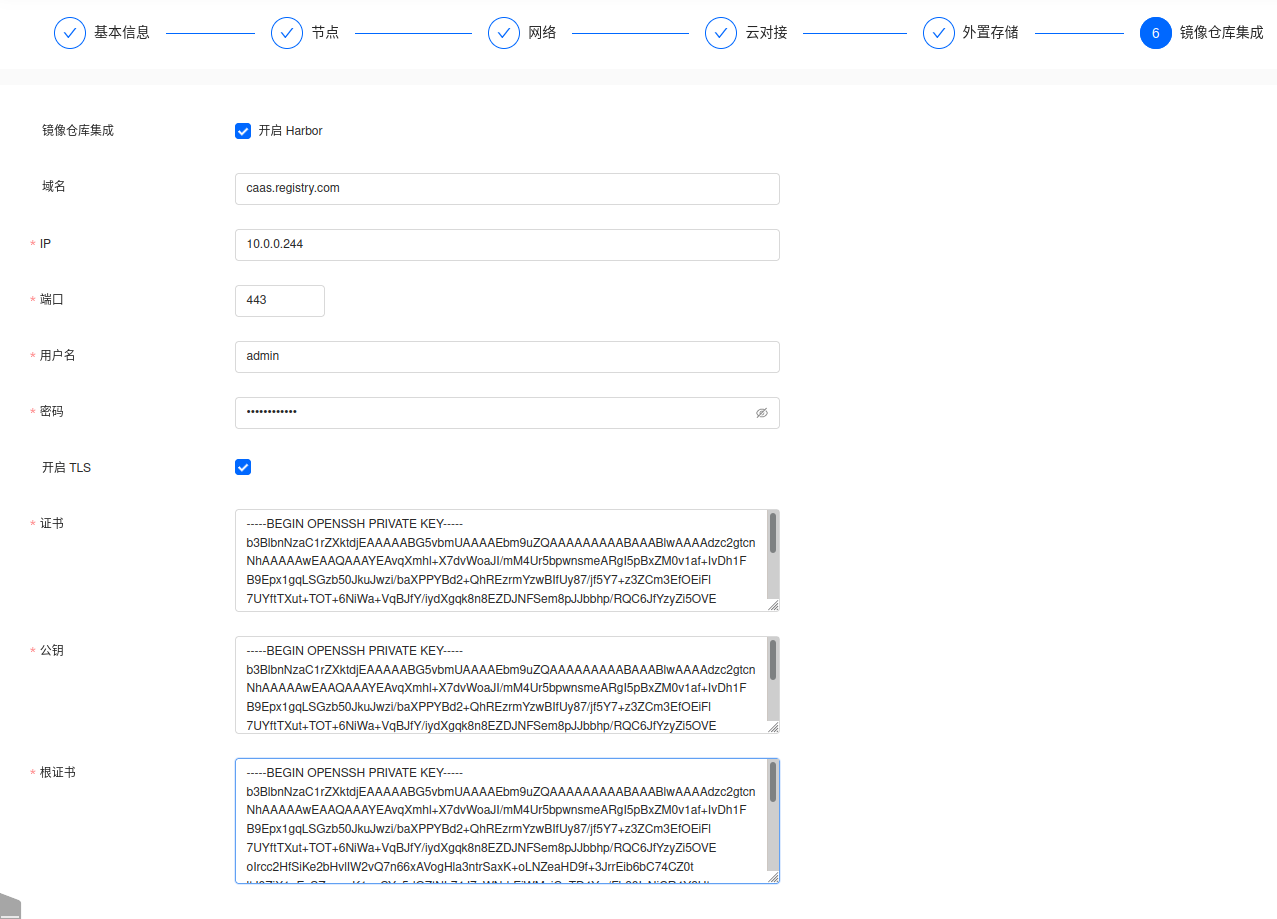
注意：如果不需要对接，用户可以保持不激活NFS对接，直接点击下一步跳过该配置。



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 作用说明 | 验证 | 是否可以缺省 | 可选值 | 默认值 |
| 开启 NFS | 是否打开 NFS 对接 | N/A | 否 | 是/否 | 否 |
| NFS Path/路径 | NFS配置的提供服务的路径,比如 /mnt/NFS\_data/ | N/A | 否 | N/A | 根据现场配置 |
| NFS Server address/Nfs 服务地址 | NFS配置的提供服务的地址,比如 10.0.0.100 | N/A | 否 | N/A | 根据现场配置 |
| Storage class Name/存储类 | 完成对接后默认创建对接cinder 的storage class 对象的名字 | 英文字符 或者英文字符-英文字符 | 否 | N/A | 根据现场配置 |
| Reclaim Policy/回收机制 | 是否删除PVC当使用NFS-csi的 pod被删除后, 是否删除对应的 PVC | N/A | 否 | 保留(retain)/删除(delete) | 根据现场配置 |

* 镜像仓库对接：

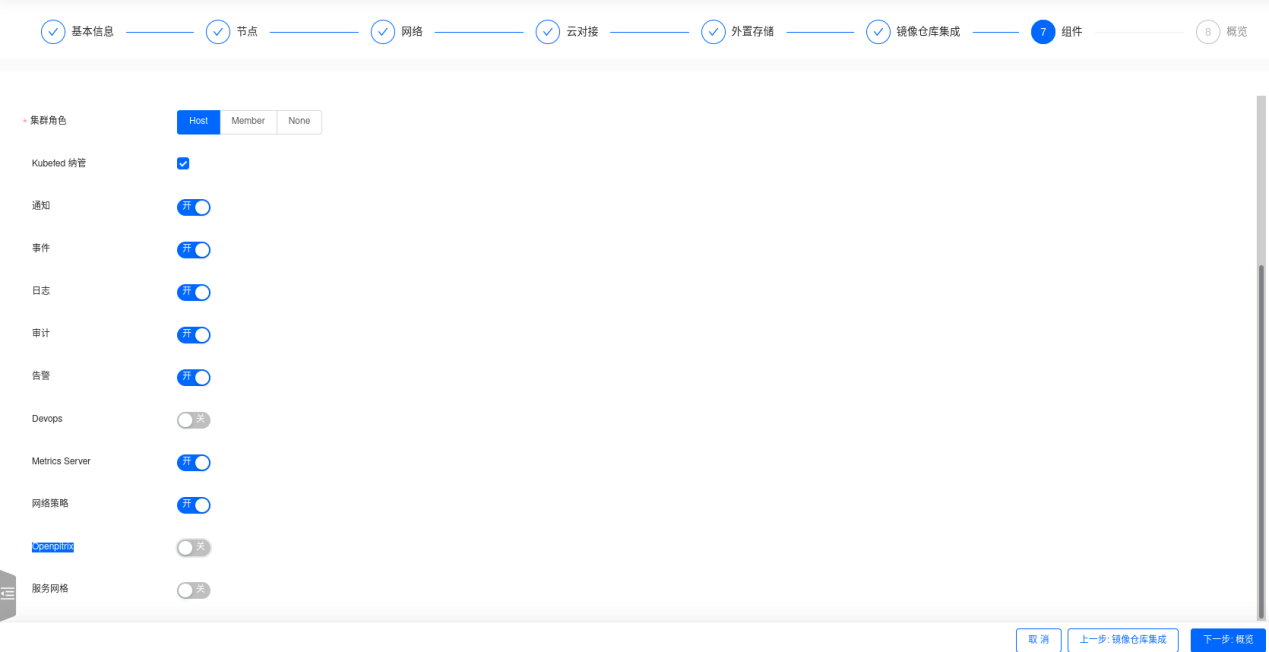
进入下一步配置 镜像仓库集成 ,点击右下角的 "下一步",如下图



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 作用说明 | 验证 | 是否可以缺省 | 可选值 | 默认值 |
| 开启 harbor | 是否打开 harbor 对接 | N/A | 否 | 是/否 | 否 |
| 域名/domain | harbor 的域名，如果填写，那么 installer 会自动在所有集群节点的 /etc/hosts 文件加入解析，您不用担心是否能否解析这个域名，但是如果 harbor 在搭建的时候配置额域名，那么这里必须填写和 harbor 在安装的时候相同的域名，比如 harbor 在安装时启用了 https 并且使用了 域名 abc.com,那么这里你也必须填写 abc.com | FQDN | 是 | N/A | N/A |
| 地址/ip | harbor 的 ip 地址 | ipv4/ipv6 | 否 | N/A | N/A |
| enable tls/开启 tls | harbor 安装时是否使用了 https 协议,如果是 | N/A | 否 | 是/否 | 是 |
| ca 证书/ca cert | harbor 安装时如果启用了 https 协议那么就会有这张证书他的位置在 harbor 安装节点上 /usr/share/harbor/tls/ca/ca.crt,当然也有可能是管理员提供 | N/A | 否 | 是/否 | 是 |
| 证书/cert | harbor 安装时如果启用了 https 协议那么就会有这张证书他的位置在 harbor 安装节点上 /usr/share/harbor/tls/server/server.cert,当然也有可能是管理员提供 | N/A | 否 | 是/否 | 是 |
| 证书 key/cert key | harbor 安装时如果启用了 https 协议那么就会有这张证书他的位置在 harbor 安装节点上 /usr/share/harbor/tls/server/server.key,当然也有可能是管理员提供 | N/A | 否 | 是/否 | 是 |

* 组件配置：

组件配置中主要是用来安装附加组件的包括多集群的组件和一些其他组件

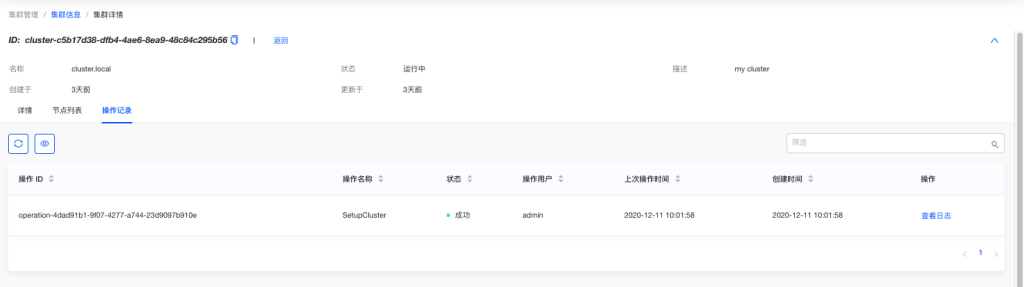


|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 作用说明 | 验证 | 是否可以缺省 | 可选值 | 默认值 |
| 开启 PaaS 平台 | 是否打开多集群的 paas 相关功能 | N/A | 否 | 是/否 | 否 |
| 存储类 | paas 应用依赖的存储类，必须要有一个，如果您开启了 openstack 对接或者 nfs 对接这里就会自动选中他们 | N/A | 否 | N/A | 根据现场配置 |
| Console 端口号 | 应用管理平台的管理界面的端口 | int | 否 | N/A | 30880 |
| 集群角色 | 设计集群的角色是 主集群(host) 还是 子集群(member/被纳管的集群) 又或者 不设置(none/干净的 k8s 集群不安装多集群的组件，但是可以安装下方选中的功能) | N/A | 否 | host/member/none | host |
| Kubefed 纳管 | 是否启用 kubefed，建议 "集群角色" 为 host 和 member 时，设置开启 | N/A | 否 | 是/否 | 否 |
| 通知 | 是开启通知功能，建议 "集群角色" 为 host 和 member 时，设置开启 | N/A | 否 | 是/否 | 否 |
| 事件 | 是开启事件功能，建议 "集群角色" 为 host 和 member 时，设置开启 | N/A | 否 | 是/否 | 否 |
| 日志 | 是开启事件功能，建议 "集群角色" 为 host 和 member 时，设置开启 | N/A | 否 | 是/否 | 否 |
| 审计 | 是开启审计功能，建议 "集群角色" 为 host 和 member 时，设置开启 | N/A | 否 | 是/否 | 否 |
| 告警 | 是开启告警功能，建议 "集群角色" 为 host 和 member 时，设置开启 | N/A | 否 | 是/否 | 否 |
| Devops | 是开启Devops功能，建议 "集群角色" 为 host 和 member 时，设置开启，除非你不想要这个功能 | N/A | 否 | 是/否 | 否 |
| Metrics Server | 是开启Metrics Server功能，建议 "集群角色" 为 host 和 member 时，设置开启，除非你不想要这个功能 | N/A | 否 | 是/否 | 否 |
| 网络策略 | 是开启网络策略(network policy)功能，建议 "集群角色" 为 host 和 member 时，设置开启，除非你不想要这个功能 | N/A | 否 | 是/否 | 否 |
| Openpitrix | 是开启Openpitrix(应用商店)功能，建议 "集群角色" 为 host 和 member 时，设置开启，除非你不想要这个功能 | N/A | 否 | 是/否 | 否 |
| 服务网格 | 是开启服务网格(istio)功能，建议 "集群角色" 为 host 和 member 时，设置开启，除非你不想要这个功能 | N/A | 否 | 是/否 | 否 |

* 概览：

在“概览”界面用户可以对整个安装配置做一个全面的了解，可以点击“确定”开始安装，也可以点击“上一步”来返回之前的页面进行配置修改，或点击某个配置项，返回该配置项界面进行修改。

### **3.1.3查看日志**

点击列表数据右侧“查看日志”，可查看操作记录。当集群正在安装时，也可查看目前安装集群到的步骤了,正在处理的事情。若发生错误，可截图发送运维人员进行排查。

### **3.1.4创建非高可用的实验集群**

通常系统将单master称之为实验环境的集群，因为一旦master发生磁盘级别的事故无法恢复的话，那么整个集群的数据也随之丢失。

创建实验环境的集群需要在“节点选择”页面中，只将一个节点加入到 "master" 和 "worker/工作节点" 中。具体操作参考“创建集群”操作。

### **3.1.5删除和释放资源**

用户可以通过删除集群的方式来释放资源。

注意：删除后的集群数据不可恢复。如果有自定义的name space使用了外部的“存储资源”比如Openstack Cinder等，需要主动释放资源。

点击集群右侧操作栏中，有“删除”与“删除和释放资源”操作。这里选择方式是根据所创建的集群决定。

如果没有额外的“存储资源”需要释放，用户可以选择使用“删除”；反之则选择“删除和释放资源”。

点击操作栏中的“删除和释放资源”，出现弹窗，输入需要释放的 name space，点击确定即可删除和释放资源。

### **3.1.6检查已删除集群**

已删除集群在列表中默认不展示。系统将所有被删除集群数据记录下载保留在 "数据库" 中，用户可以点击“状态/Status’列来打开过滤,取消勾选”隐藏已销毁/Hide Destroyed Cluster”来展示已经被删除的集群。

### **3.1.7使用相同配置重建集群**

用户可使用之前被销毁的集群相同的配置重新创建集群。

但需要注意以下情况：

* 需要被重建集群中的节点是否会其他集群占用；
* 需要被重建集群中的节点是否可用：可以通过左侧菜单 "节点信息" > "节点信息" 来确认 "状态/status 列" 是不是 "就绪/ready"， "客户端状态/client status 列" 是不是 "就绪/ready" ，"禁用/disable 列" 是不是 "启用/enable" ， "角色/role 列" 是不是为 "-"。
* 状态/status 列：节点是否被平台支持,如果节点的操作系统不被支持会呈现 "不支持/Not Support" 状态；
* 客户端状态/client status 列：代表客户端是否正常运行,如果没有启用这里会呈现为 "不可达/unreachable" 的状态；
* 禁用/disable 列：代表该节点是不是被禁用,如果呈现为 "禁用/disabled" 那么代表该节点被禁用；
* 角色/role 列：代表该节点在现有集群(正常运行中的集群)中扮演的角色,如果值为 "master,worker,ingress,external lb" 代表该节点已经被其他的集群占用了。

点击“已销毁”状态的集群右侧操作栏中的“重新安装”，出现弹窗，点击确定即可重新安装该配置的集群。页面将自动跳转至该集群详情中的“操作日志”列表页面。



### **3.1.8从集群中移除工作节点**

从集群中移除一个或多个工作节点（不支持移除master），此版本在后端已经做过安全处理，但是用户还是需要注意生产中的安全问题。在平台上操作的时需要注意以下内容，避免应用发生中断的问题。

* 要被移除的节点是否为该集群的最后一个工作节点。否则被移除节点上的业务将无处可迁移，后台处理程序将阻止用户将最后一个工作节点从集群中移除。
* 举例说明: 一个集群有1个master(只做master，非共存)，1个worker，这时移除这个worker，平台会拒绝执行本次操作。
* 举例说明: 一个集群有1个master(master+worker并存)，1个worker,这时移除这个worker，后台将允许本次操作。
* 业务是否仅有单副本，并且只运行在要被移除的节点上。
* 业务是单独的pod，没有复制器管理， 并且运行在将要被移除节点上，这种情况业务会彻底中断。
* 业务是单独的pod ，有复制器管理，并且运行在将要被移除节点上 ，这种情况业务会一定时间的中断，中断时间取决于用户的集群的硬件性能和当前的负载。

点击目标集群的名称，进入该集群的详情页面，点击进入“节点列表”的标签页。



当master+worker并存情况下，点击目标节点（worker）右侧的“删除”，出现弹窗，点击确认，即可将该节点从目标集群中移除。



### **3.1.9添加工作节点至集群**

用户如果觉得集群工作负载比较高，可以通过为集群的动态添加节点来达到主动扩容的目的，添加新的“节点”不会影响现有的业务的运行。

点击目标集群的名称，进入该集群的详情页面，点击进入“节点列表”的标签页。点击“添加到集群”按钮，出现弹窗。

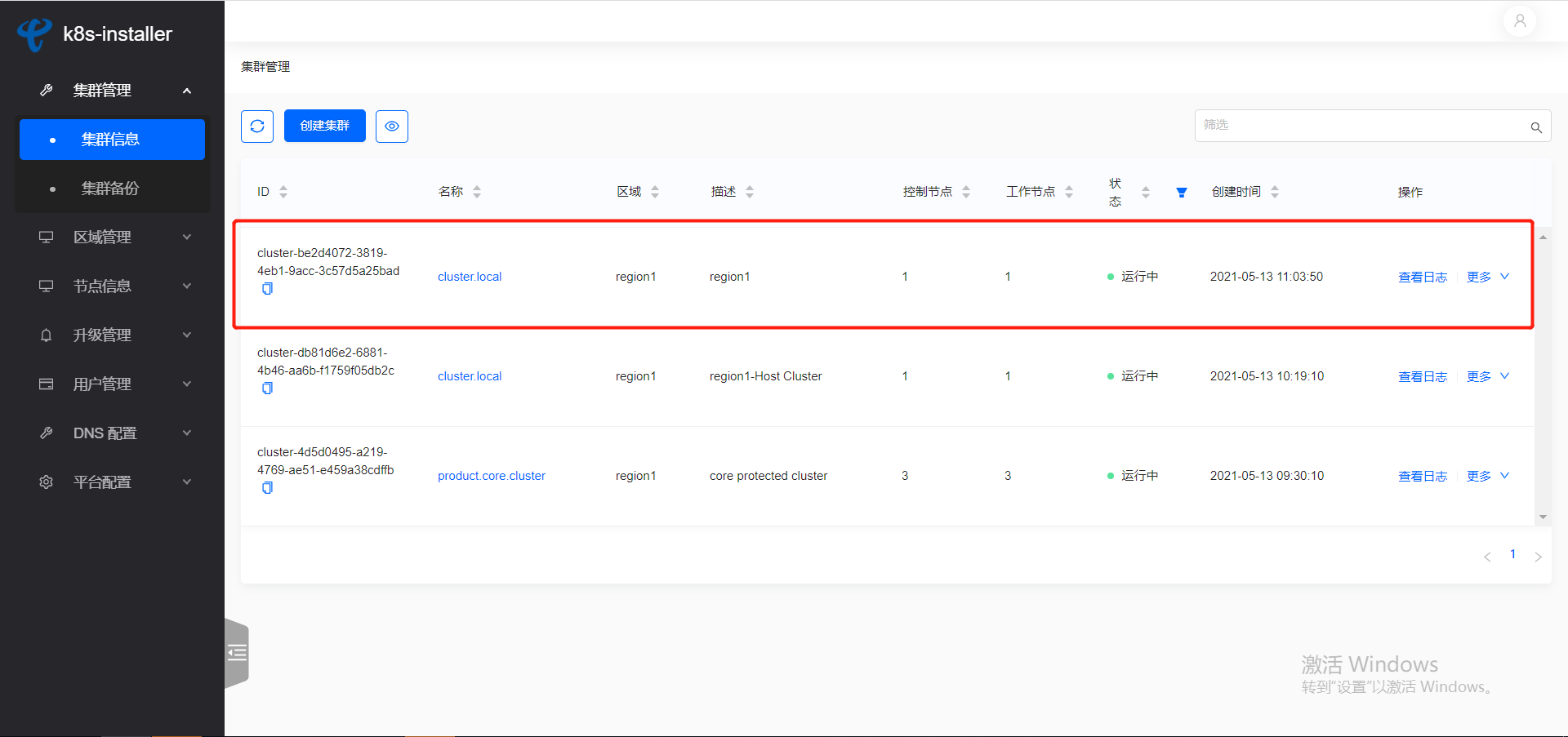
左侧为可选节点，右侧为已选节点，点击“>”添加，点击确定即可完成添加。节点添加过程中,该节点会呈现为“不可用/not ready”的状态,等待一段时间后,改节点的状态会变为“就绪/ready”，还可至 "操作日志" 进行详细内容的查看。

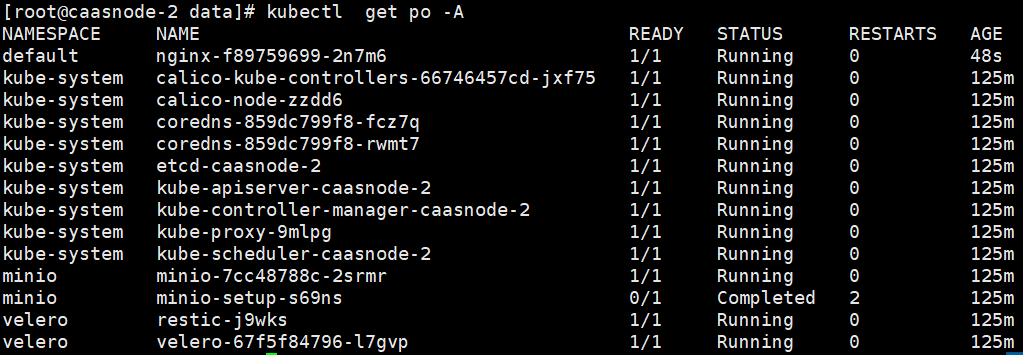
### **3.1.10集群备份及恢复**

CTSI CaaS对 k8s 集群的备份与传统备份有区别：k8s 集群的备份主要为备份 ETCD 数据库数据，以及 k8s 的资源对象备份，如 namespace，deplotment、configMap，对资源自身产生的文件和数据不做备份。例如集群中运行的 mysql pod，该 mysql pod 产生的数据和文件， CTSI CaaS不会为之备份，同理，文件类的 pv 对象下的文件， CTSI CaaS也不做备份，仅仅备份 pv 这个对象。目前 CTSI CaaS提供的备份功能属于热备份，备份期间不影响集群的使用。平台虽然不反对在集群 "繁忙期" 备份，但是同样也强烈不赞成在集群 "繁忙期" 备份。

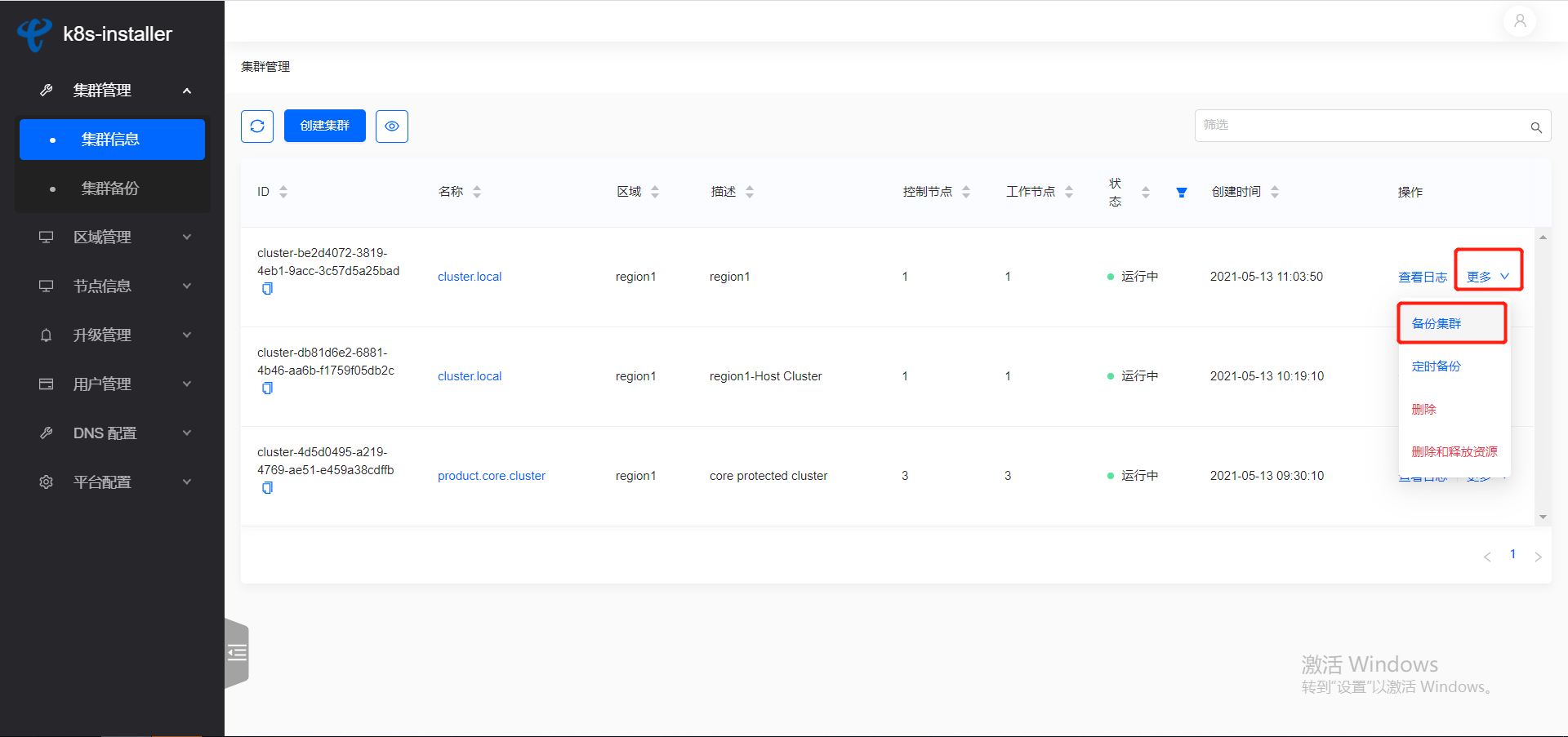
CTSI CaaS对 k8s 集群的恢复与传统恢复有区别：一、在集群正常运行期间进行恢复，则 CTSI CaaS将对该集群进行非覆盖式恢复，就是把已经删除掉的资源对象恢复回来即可。二、当集群被删除后，若想恢复集群，则需要先重装集群，方可对该集群使用恢复功能。目前主流的集群恢复的做法都是重建集群，再对其恢复（如 CaaS4.0、rancher，都是仅仅备份 etcd ，再重建集群后进行恢复）

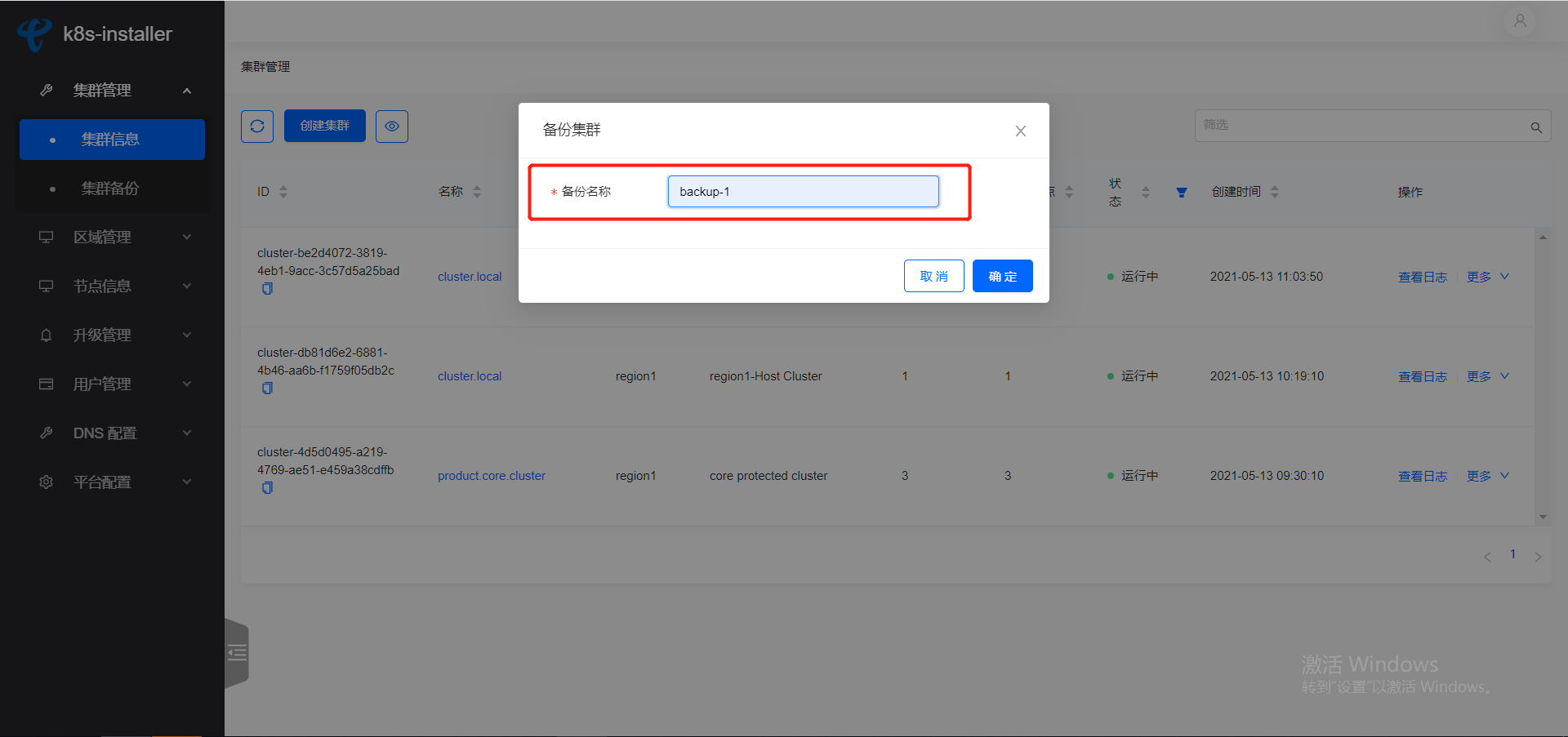
CTSI CaaS对 k8s 集群的定时备份说明：定时备份在创建时，会先进行一次集群的备份功能，后续再根据备份时间计划触发集群的备份功能。目前已知问题：启动定时备份时会立即触发备份机制，第一次的备份功能也因此提前被触发，该问题将在后续迭代修复。

1. 用 CTSI CaaS 创建一个集群
2. 在该集群的节点后台创建一个 nginx 的 deployment 对象: kubectl create deploy nginx --image=nginx 和 kubectl get pod -A

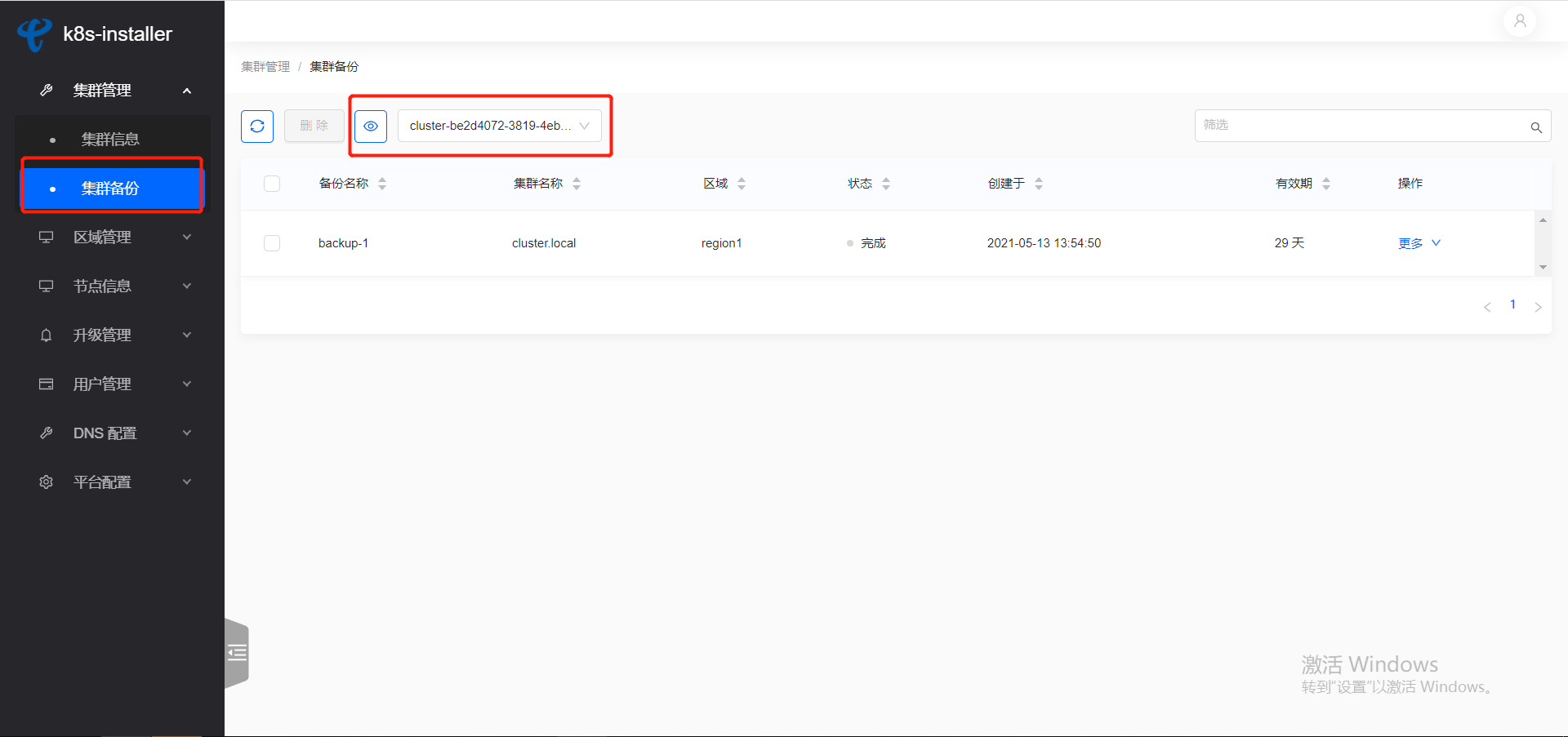


1. 创建备份

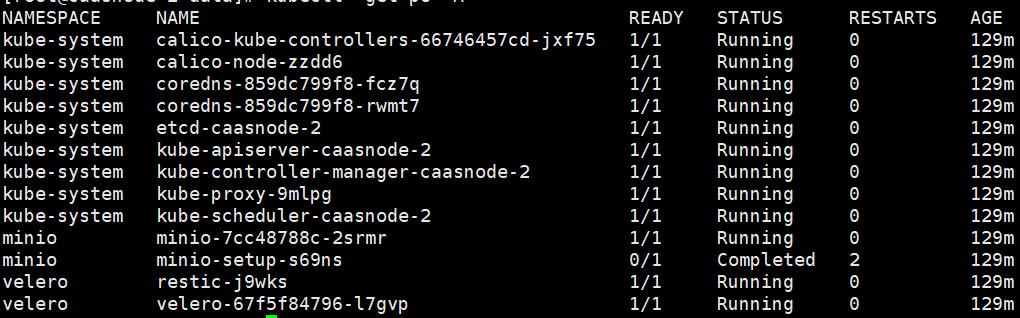




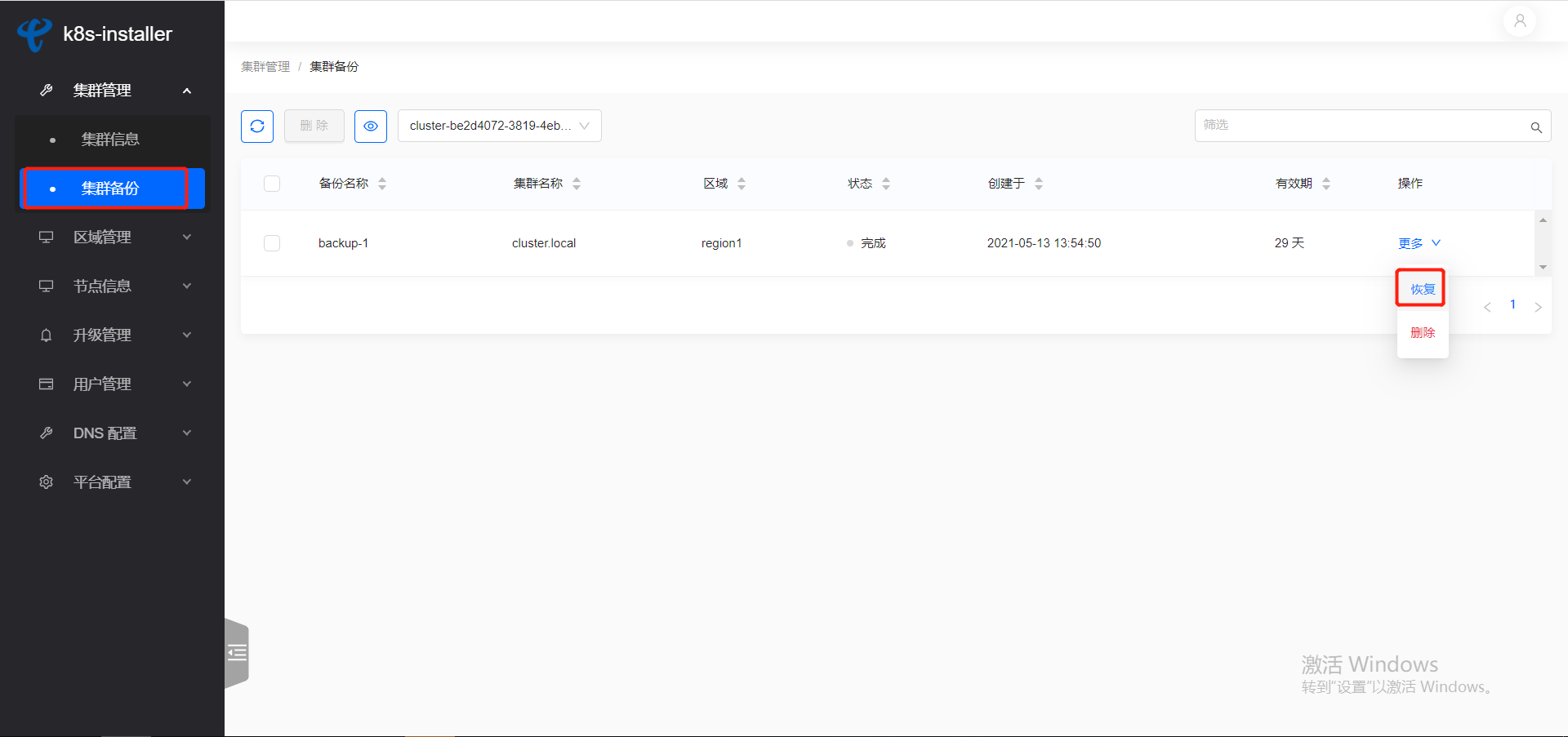
1. 查看备份结果



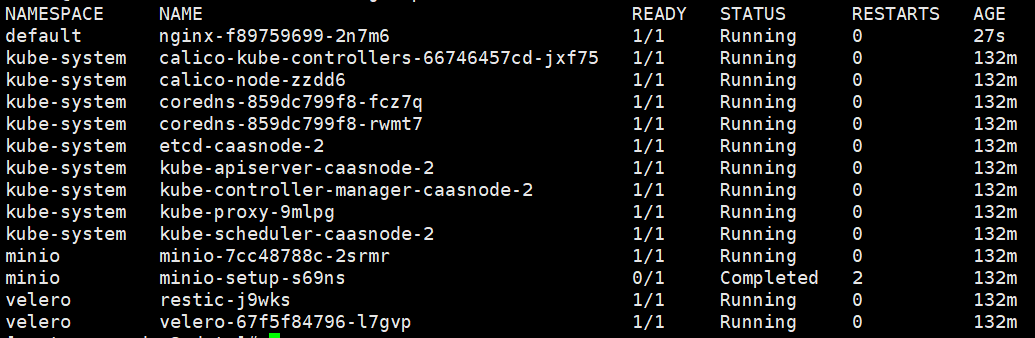
1. 删除 nginx 的 deployment 对象：kubectl delete deploy nginx 和 kubectl get pod -A



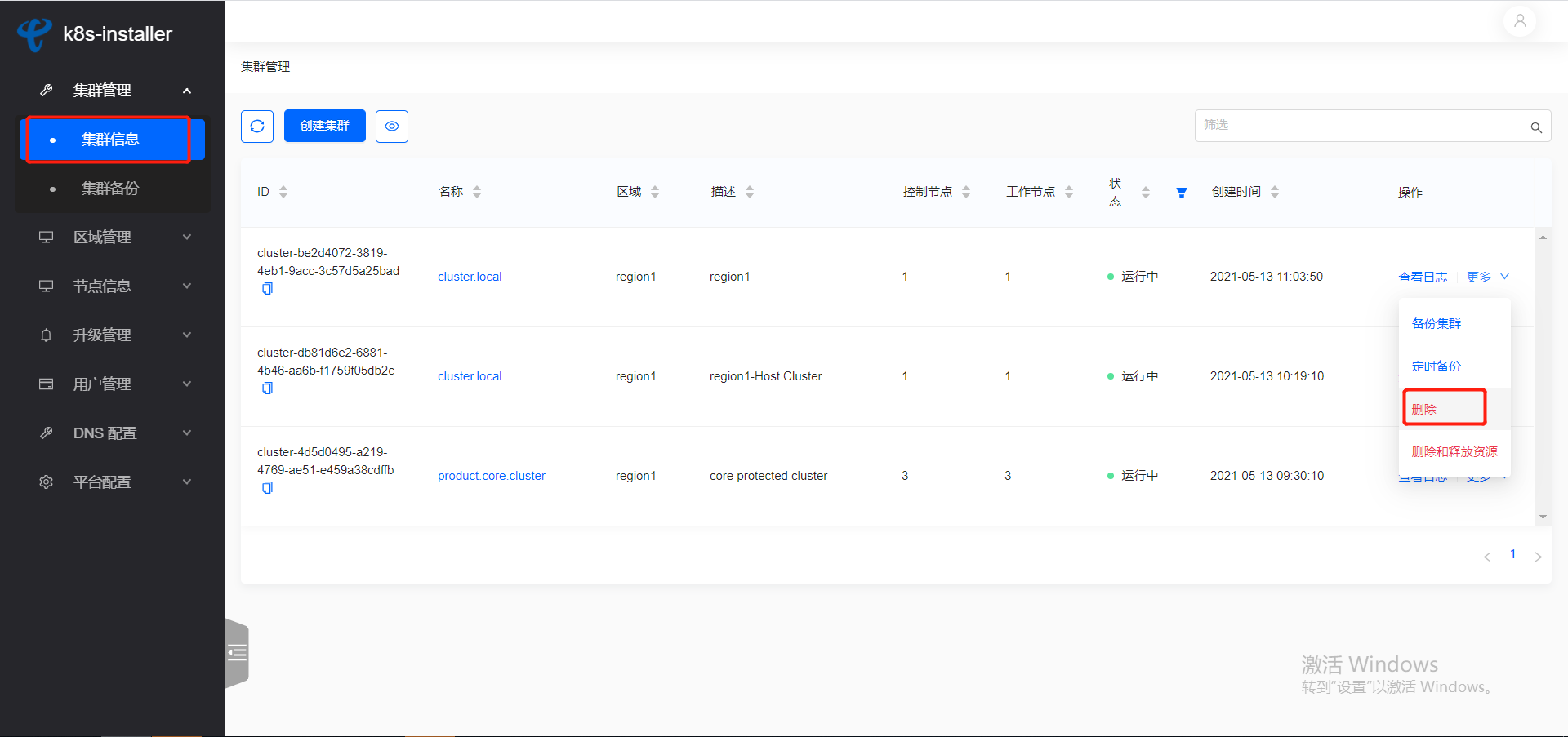
1. 集群恢复



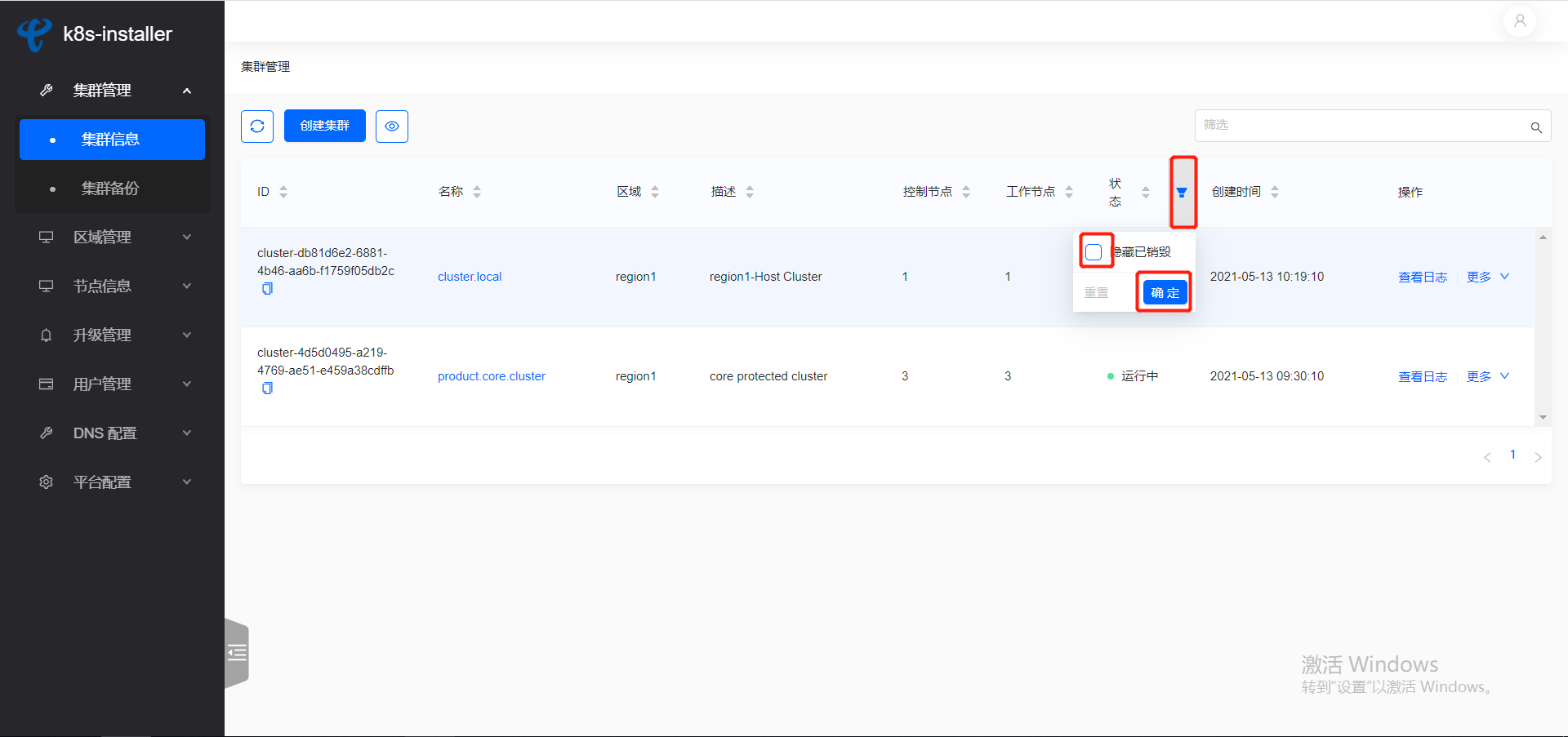
1. 在该集群的节点后台查看 nginx 是否恢复

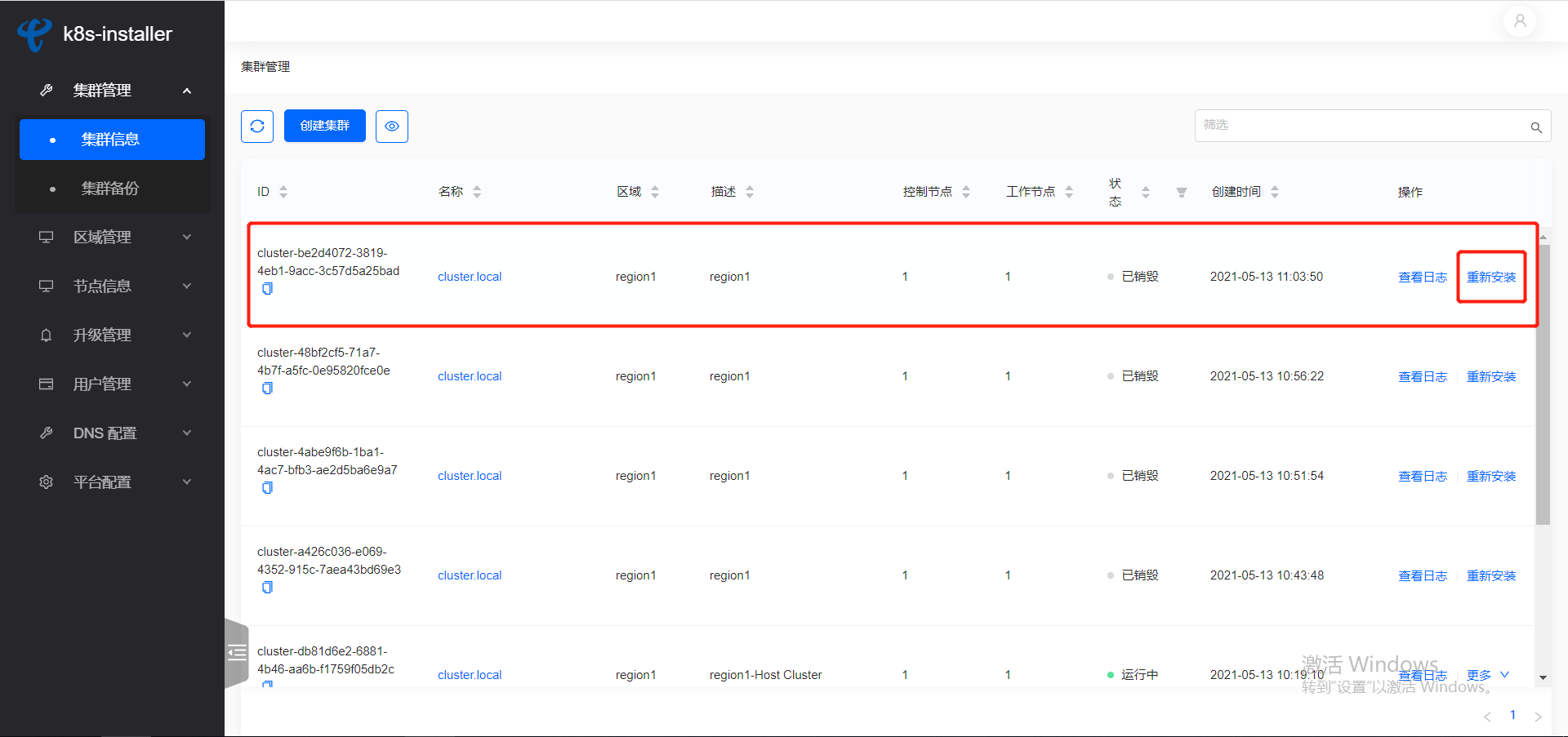


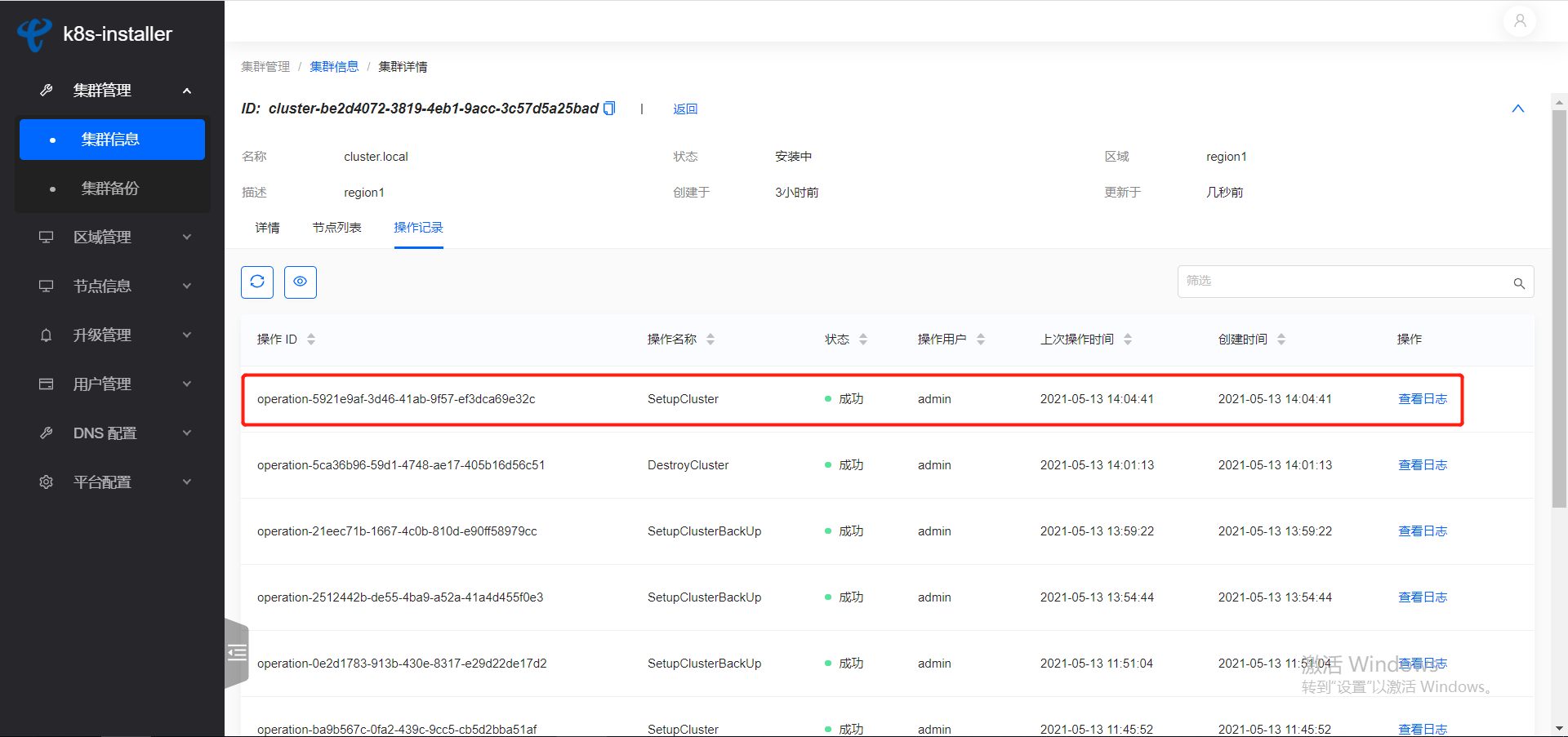
1. 删除整个集群



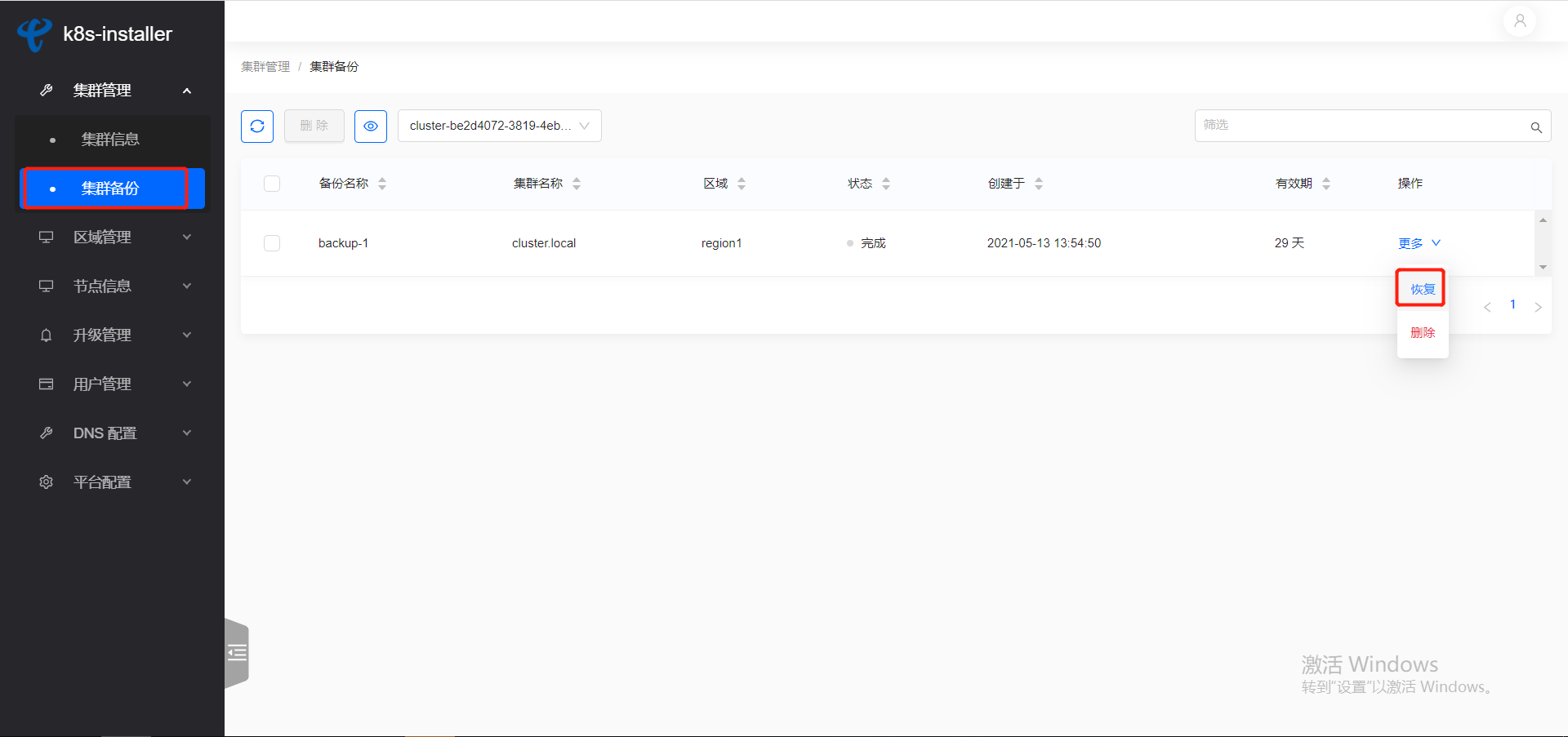
1. 重装集群

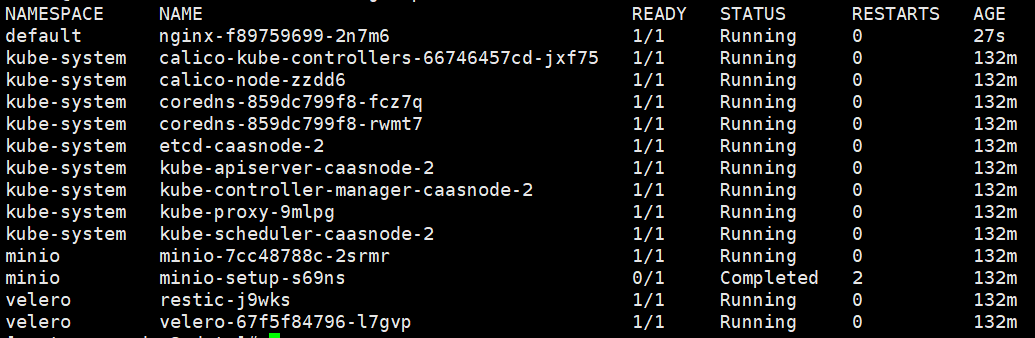


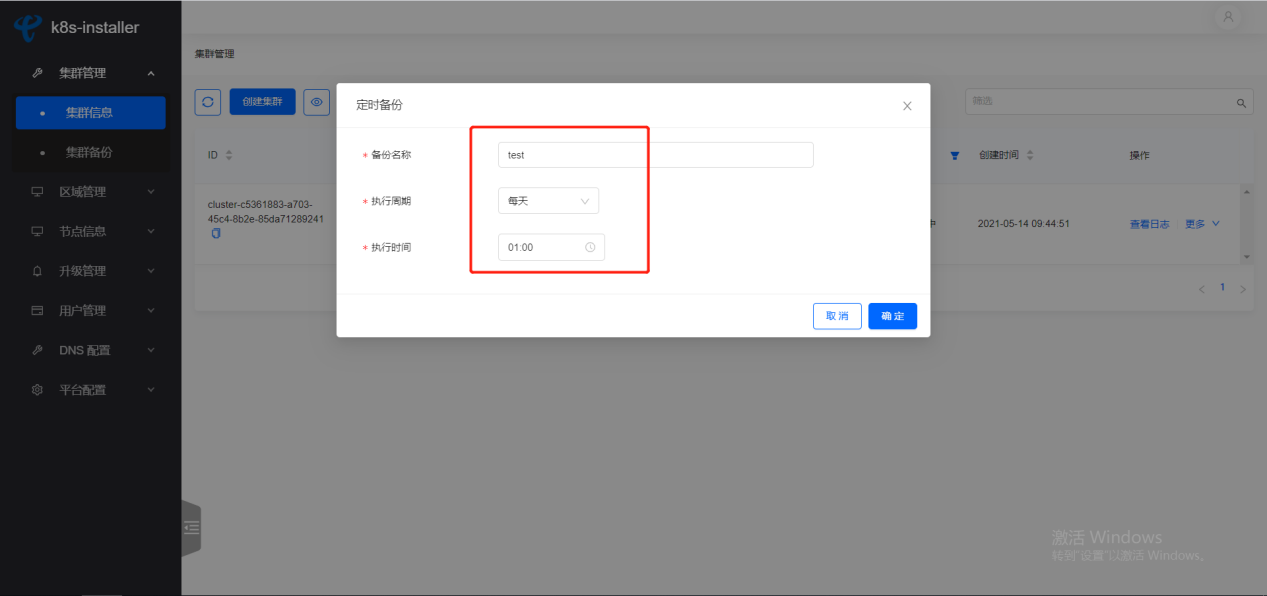
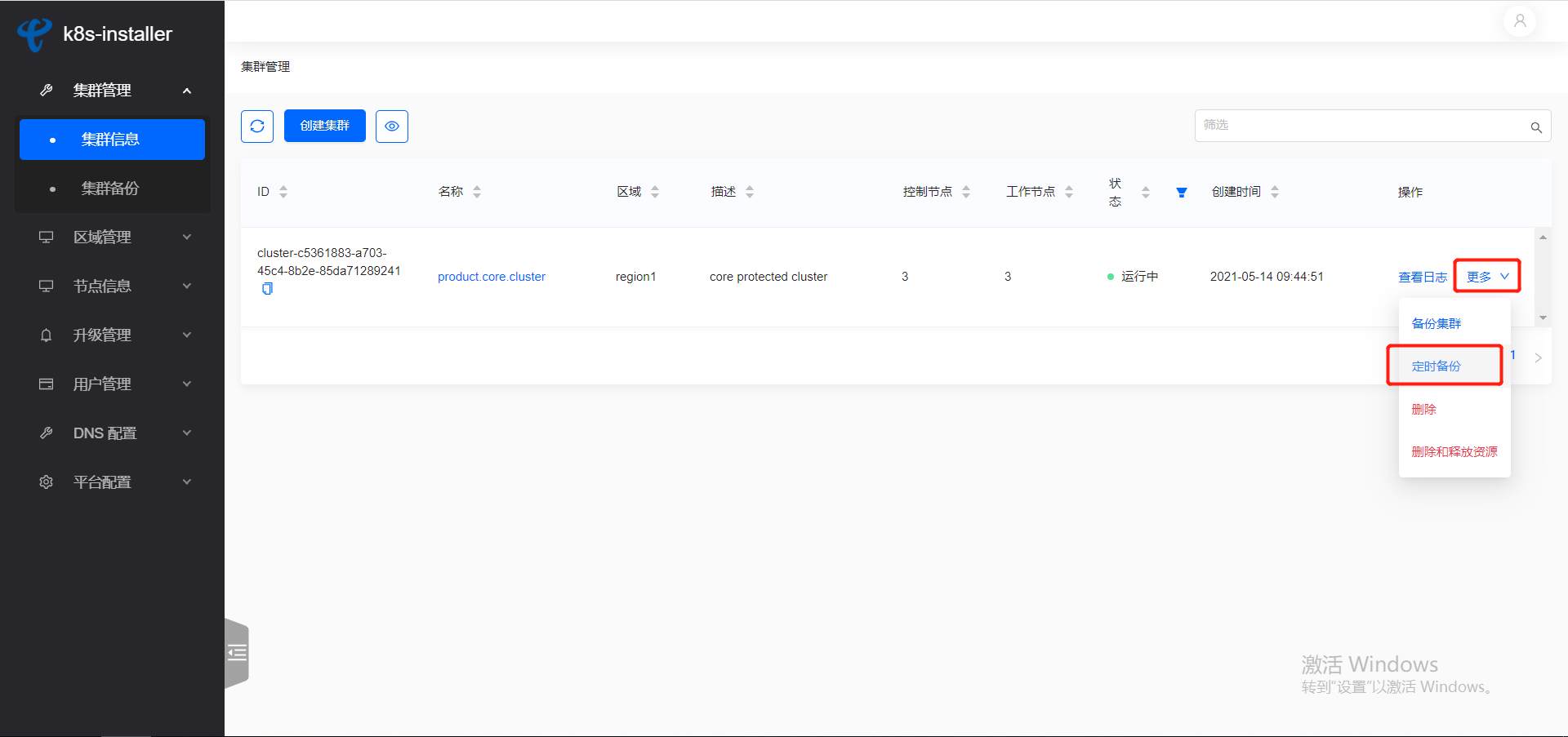




1. 集群恢复，该集群的节点后台查看 nginx 是否被恢复





1. 定时备份流程：目前已知问题：启动定时备份时会立即触发备份机制，第一次的备份功能也因此提前被触发，该问题将在后续迭代修复。

# 四、节点信息

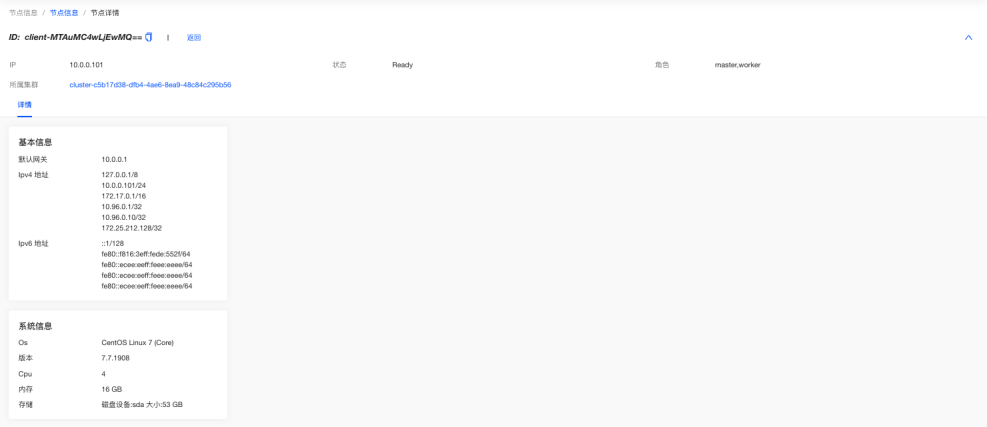
## **4.1节点信息**

### **4.1.1节点列表**

用户可以通过点击主面板右侧菜单“节点信息>节点信息”进至节点管理界面，查看节点列表。可查看IP、CPU、内存、状态、客户端状态、禁用、角色。

### **4.1.2节点详情**

点击目标节点IP，进入节点详情界面，可查看节点基本信息，系统信息等内容。



### **4.1.3加入集群**

点击目标节点右侧操作栏中的”加入集群“，在弹窗中选择需要加入的集群，点击确定即可。



### **4.1.4禁用节点**

点击目标节点右侧操作栏中的“禁用”按钮，出现二次确认弹窗，点击确定即可禁用该节点。

注意：已加入集群的节点不可禁用。



# 五、配置文件

## **5.1系统所使用的端口**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **端口号** | **作用说明** | **对应server的配置** | **缺省默认值** | **对应client的配置** | **缺省默认值** | **位置** |
| 8099 | 服务端提供管理端 api 服务的端口 | api-server  api-port: 8099 | 8099 | N/A | N/A | 服务端（bastion/  server） |
| 8079 | 服务端提供资源下载的 api 服务的端口 | api-server  resource-server-port: 8079 | 8079 | N/A | N/A | 服务端（bastion/  server） |
| 9090 | 服务端用来检测客户端是否正常启动着的检测端口 | signal-port: 9090 | 9090 | signal-port: 9090 | 9090 | 客户端（client） |

注意：表格中的字段 "对应server的配置"与"对应client的配置" 均为yaml格式，包含换行与缩进，请见下述解释：

举例1:

api-server.api-port: 8099

api-server.xxx-port: 8338

等于

api-server

api-port: 8099

xxx-port: 8338

举例2:

signal-port: 9090

等于

signal-port: 9090

## **5.2配置文件详细说明**

本文档没有提及的配置，代表该配置没有被使用或将在未来的版本中被移除。

注意：所有的配置均是yaml格式，表格中“父属性”代表“属性”的上一层级，请见下述解释：

举例1:

父属性：api-server

属性: api-port

对应 yaml 配置为

api-server

api-port:

举例2:

父属性：N/A

属性：signal-port

对应 yaml 配置为

signal-port:

### **5.2.1server端配置**

|  |  |  |  |
| --- | --- | --- | --- |
| **父属性** | **属性** | **说明** | **缺省默认值** |
| N/A | api-server | api-server 配置开始 | N/A |
| api-server | api-port | api管理服务所监听的端口号 | 8099 |
| api-server | jwt-expired-after-hours | javascript web token 过期时间 | 8 |
| api-server | jwt-sign-string | javascript web token 加密签名字符串 | 根据现场配置 |
| api-server | jwt-sign-method | javascript web token 加密算法 | HS256 |
| api-server | resource-server-cidr | 当bastion(server)多节点时,如果需要通过一个vip提供资源下载时,可以配置为vip的地址。若不需要则不要配置 | N/A |
| api-server | resource-server-filepath | 依赖软件包的本地目录(bastion) | N/A |
| api-server | resource-server-port | 提供依赖软件下载的服务端口号(bastion) | 8079 |
| api-server | signal-port | 检查客户存活的检测端口 | 9091 |
| api-server | system-info-pubkey | 证书的公钥 | N/A |
| api-server | disable-lazy-operation-log | 日志输出模式,true=立刻输出日志到数据库,false=操作出现错误或成功后一次性输出日志到数据库。  如果需要减少etcd数据库的压力可以设置为 false ,但不能立刻看到日志输出 | N/A |
| api-server | server-id | bastion(server)的id | 无需设置自动生成 |
| N/A | cache | 数据层配置开始 | N/A |
| cache | cache-runtime | no-cache=不使用缓存适合在多 bastion(server) 的情况下使用,local-ram=使用本地缓存,适合在单 bastion(server) 的情况下使用 | no-cache |
| N/A | etcd | 数据库 etcd 配置开始 | N/A |
| etcd | etcd-auth-mode | 数据库 etcd 认证方式, tls=tls认证方式(生产中),none=无认证直连(实验环境中) | tls |
| etcd | etcd-endpoints | 数据库 etcd 访问地址,如果 etcd-auth-mode: node 那么请使用 http 协议,如果是链接集群那么用 " ｜ " 分开。如https://etcd-node1:2379 | [https://etcd-node2:2379](https://etcd-node2:2379/) | https://etcd-node3:2379 | https://localhost:2379 |
| etcd | etcd-ca-file-path | 数据库 etcd ca 证书路径 | N/A |
| etcd | etcd-cert-file-path | 数据库 etcd cert 证书路径 | N/A |
| etcd | etcd-key-file-path | 数据库 etcd key 路径 | N/A |
| N/A | host-requirement | 系统检查配置开始 | N/A |
| host-requirement | check-kernel-version | 是否匹配内核版本,不要设置为 true 否则程序会把不符合 kernel-major,kernel-sub,kernel-ver 的内核的主机表为不支持 | false |
| host-requirement | kernel-major | 内核检查的主版本号 | 3 |
| host-requirement | kernel-sub | 内核检查的次版本号 | 10 |
| host-requirement | kernel-ver | 内核检查的版本号 | 0 |
| host-requirement | check-os-family | 是否匹操作系统版本,不要改为 false 目前我们 CTSI CaaS只支持 centos-7.x | true |
| host-requirement | support-os-family | 支持那些操作系统,如果有多个则用 "逗号" 隔开,如 "centos,ubuntu" | centos |
| host-requirement | support-os-family-version | 支持那些操作系统的版本,如果有多个则用 "逗号" 隔开,如 centos-7和ubuntu 16.06 则表示为 "7,16.04" | 7 |
| N/A | log | 系统日志配置开始 | N/A |
| log | log-level | 日志等级,越低的等级代表越少的输出,我们的应用最高为 6,请不要随意修改 | 6 |
| N/A | message-queue | 消息队列配置开始 | N/A |
| message-queue | auth-mode | 消息队列认证模式,目前只支持 basic=基础身份认证 | N/A |
| message-queue | cluster-port | 消息队列集群通信端口 | 9890 |
| message-queue | message-queue-group-name | 消息队列组消息频道 | caas-node-status-report-queue |
| message-queue | message-queue-leader | 消息队列集群的 leader | 根据现场配置 |
| message-queue | message-queue-node-status-report-in-subject | 节点汇报自身状态的消息频道 | caas-node-status-report-subject |
| message-queue | message-queue-server-address | 消息队列 server 的地址 | 127.0.0.1 |
| message-queue | message-queue-server-listening-on | 消息队列 server 的监听的地址 | 0.0.0.0 |
| message-queue | message-queue-subject-suffix | 消息队列普通消息频道 | k8s-install |
| message-queue | username | 消息队列用户名 | 根据现场配置 |
| message-queue | password | 消息队列用户名的密码 | 根据现场配置 |
| message-queue | port | 消息队列的监听端口 | 9889 |
| N/A | server-message-timeout-config | 任务超时设置开始 | N/A |
| server-message-timeout-config | task-basic-config-timeout | 基本工作的任务超时时间(秒)，如关闭firewalld等 | 5 |
| server-message-timeout-config | task-cri-timeout | 安装容器运行时的任务超时时间(秒) | 300 |
| server-message-timeout-config | task-kubeadm-init-first-control-plane-timeout | 初始化控制节点的任务超时时间(秒) | 300 |
| server-message-timeout-config | task-kubeadm-join-control-plane-timeout | 加入控制集群的任务超时时间(秒) | 300 |
| server-message-timeout-config | task-kubeadm-join-worker-timeout | 加入工作集群的任务超时时间(秒) | 300 |
| server-message-timeout-config | task-kubectl-timeout | kubectl的任务超时时间(秒) | 300 |
| server-message-timeout-config | task-rename-hostname-timeout | 重命名节点的任务超时时间(秒) | 300 |
| server-message-timeout-config | task-vip-timeout | 部署外部的负载均衡的任务超时时间(秒) | 300 |

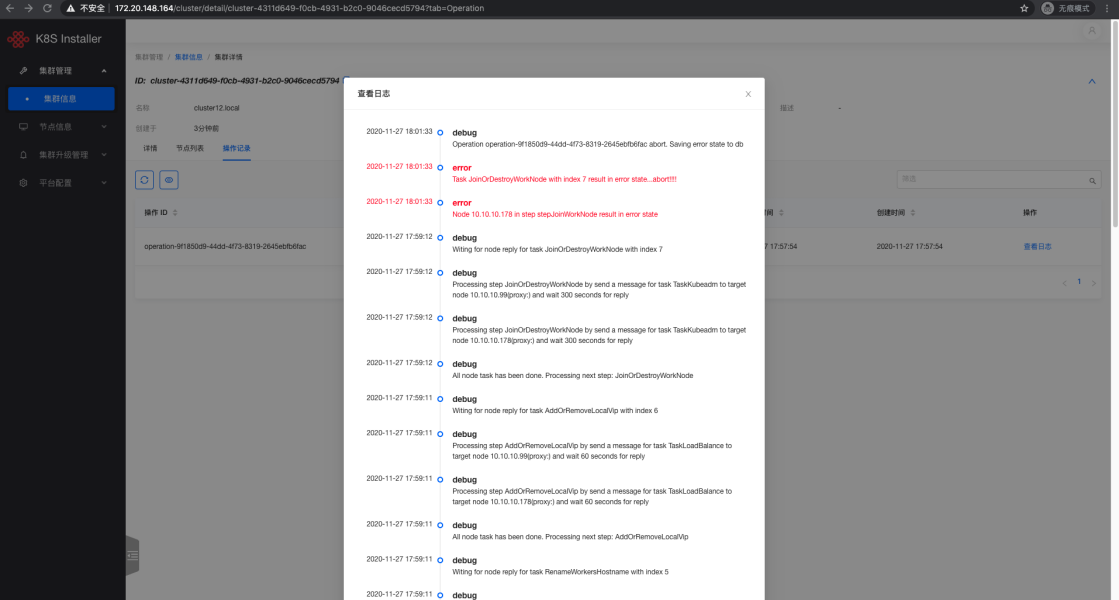
### **5.2.2client端配置**

|  |  |  |  |
| --- | --- | --- | --- |
| **父属性** | **属性** | **说明** | **缺省默认值** |
| N/A | api-server | api-server 配置开始 | N/A |
| api-server | signal-port | 检查客户存活的检测端口 | 9091 |
| api-server | client-id | Client ID | 无需设置自动生成 |
| N/A | log | 系统日志配置开始 | N/A |
| log | log-level | 日志等级,越低的等级代表越少的输出,我们的应用最高为 6,请不要随意修改 | 6 |
| N/A | message-queue | 消息队列配置开始 | N/A |
| message-queue | auth-mode | 消息队列认证模式,目前只支持 basic=基础身份认证 | N/A |
| message-queue | message-queue-group-name | 消息队列组消息频道 | caas-node-status-report-queue |
| message-queue | message-queue-node-status-report-in-subject | 节点汇报自身状态的消息频道 | caas-node-status-report-subject |
| message-queue | message-queue-server-address | 消息队列 server 的地址,如果链接接群使用 "逗号隔开",如 "10.10.10.24,10.10.10.41,10.10.10.63" | 127.0.0.1 |
| message-queue | message-queue-subject-suffix | 消息队列普通消息频道 | k8s-install |
| message-queue | username | 消息队列用户名 | 根据现场配置 |
| message-queue | password | 消息队列用户名的密码 | 根据现场配置 |
| message-queue | port | 消息队列的监听端口 | 9889 |

# 六、常见故障

## **6.1工作节点加入集群失败**

在特定的情况下，如网络不稳定等因素造成 "添加工作节点到集群" 操作出现错误，将看到“日志”出现“error”。



* 恢复方案：

在界面上点击 "删除" 集群，等待运行完成后,点击 "重新安装" ，即可修复错误的安装。

## **6.2集群搭建完成，跨节点的pod无法互相访问**

在某些云环境下，平台会出现跨节点的pod不能互相访问的情况。此时需要用户检查底层云平台是否允许了ipip，比如Openstack平台在使用了dpdk时，会出现此问题，此时用户需要关闭Openstack的防火墙功能，才能正常访问。

在有 openstack 命令行工具的节点上运行以下命令

找到所有集群的主机的 "端口号"，例如：用户 CTSI CaaS 平台由3个bastion(server)和3个client(10.0.0.10，10.0.0.11，10.0.0.12) 组成，用户需要找到3个client的端口号 $ openstack port list | grep -i "10.0.0.10" # 复制 id

$ openstack port list | grep -i "10.0.0.11" # 复制 id

$ openstack port list | grep -i "10.0.0.12" # 复制 id

关闭防火墙

$ openstack port set --disable-port-security --no-security-group [10.0.0.10 节点的 port 的id]

$ openstack port set --disable-port-security --no-security-group [10.0.0.12 节点的 port 的id]

$ openstack port set --disable-port-security --no-security-group [10.0.0.12 节点的 port 的id]

## **6.3升级失败**

CTSI CaaS集群升级功能在复杂的网络环境下，可能会因其他原因造成升级失败，最大可能的情况是“下载镜像超时”或“节点下线”造成的升级失败。若升级失败，平台“操作日志”界面中将会明确标注具体节点出现的问题。

* 恢复方案：

下载镜像超时：登录到错误的节点,尝试手动下载镜像,如：用户的升级的目标版本为 "1.19.2" , 那么用户可以尝试运行以下命令,完成后点击 "操作日志" 失败记录右侧的 "从错误位置继续" 按钮即可。

```console

# 手动下载镜像

$ docker pull [镜像仓库的ip]:[端口]/kube-proxy:1.19.2

$ docker push [镜像仓库的ip]:[端口]/kube-apiserver:1.19.2

$ docker push [镜像仓库的ip]:[端口]/kube-controller-manager:1.19.2

$ docker push [镜像仓库的ip]:[端口]/kube-scheduler:1.19.2

$ docker push [镜像仓库的ip]:[端口]/etcd:3.4.13-0

节点下线：尝试让节点重新上线，确认ki-client的systemd已经正常启动，完成后点击 "操作日志" 失败记录右侧的 "从错误位置继续" 按钮即可。

```console

# 确认服务是 active running 状态

$ systemctl status ki-client

```

## **6.4使用限制与注意事项**

### **6.4.1云平台的挂载型设备的上限**

在特定的云平台上，单个节点的挂载磁盘的数量是有限制的，比如Openstack的上限为26(包含系统)，所以在使用 CTSI CaaS 平台时，需要注意业务调度，用户可以通过 "节点信息" 中的详细页面看到使用情况。



### **6.4.2网络带宽**

CTSI CaaS 平台可以高并发的安装多套集群，也意味着会同时有很多个主机同时从私有的镜像仓库中下载镜像。所以用户需要考虑物理带宽是否足够大，否则有可能造成下载镜像超时，从而导致安装失败。

* 解决方法：若用户的网络贷款有限，建议减少安装集群的规模。如：当需要安装100个节点的集群，用户可以先建立一个含有20个节点的集群，然后添加4次20个工作节点（共80个节点）到该集群中。
* 解决方法：增加带宽大小。

### **6.4.3节点数量**

CTSI CaaS平台目前不建议大于50个节点的集群，若实际使用情况中存在大于50个节点的集群，建议用户手动配置route reflection，用来优化集群中的节点的路由表所维护的稳定性。

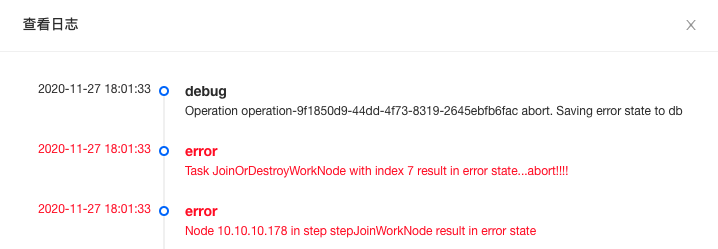
# 七、查看日志

在生产过程中快速定位至错误的日志可提高用户使用效率，在 CTSI CaaS平台中已经部署了Loki+Grafana，为用户收集操作日志，方便用户在遇到错误时快速定位。

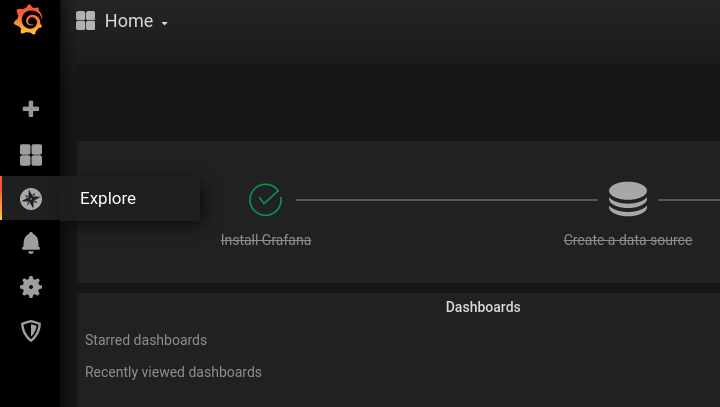
## **7.1获取错误操作ID**

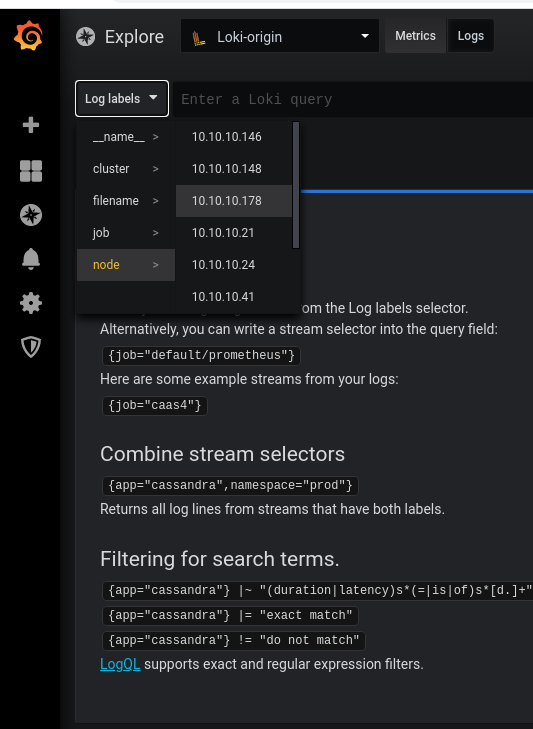
无论是创建集群、删除或者升级，都会和具体的 "操作ID" 产生关系。用户可以很容易的查看到其对集群的操作日志，操作日志会清晰的告诉用户错误来自于哪个“节点”、发生在哪个“步骤”。

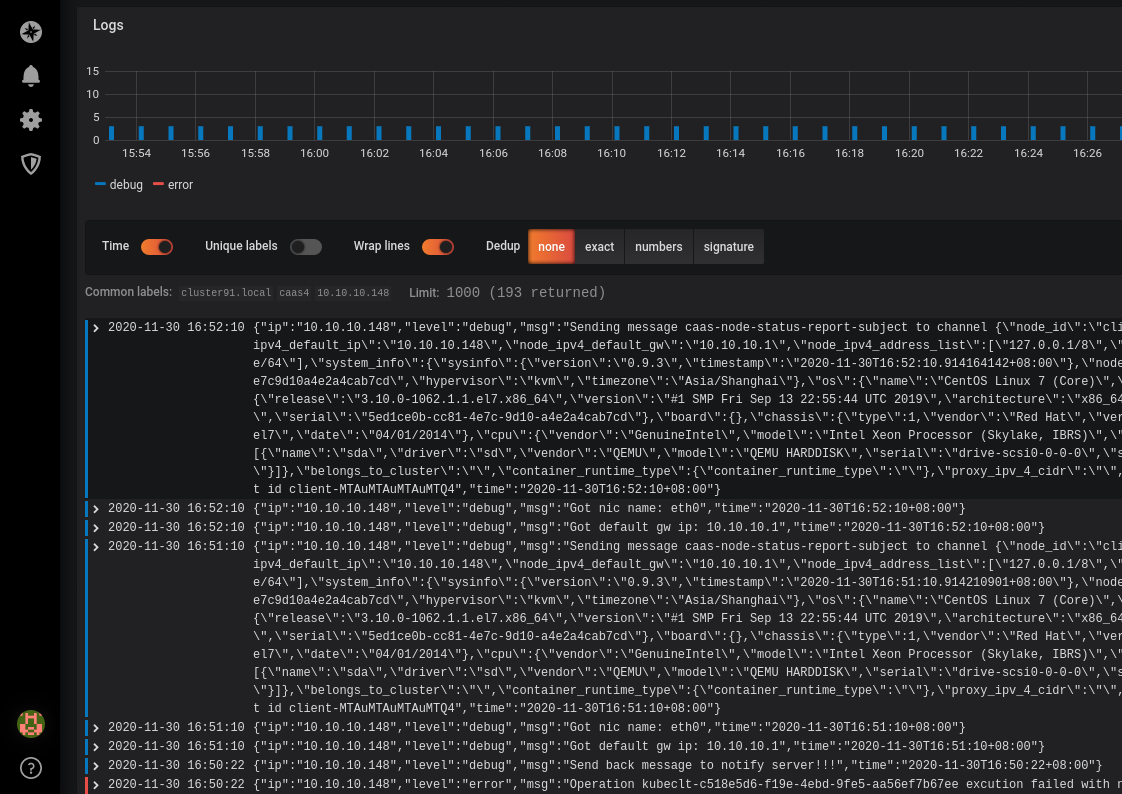
复制日志 "2020-11-27 18:01:33" 的内容。



登录日志收集界面，访问任意一台的bastion(server)的 IP:3000，如：[http://10.0.0.29:3000](http://10.0.0.29:3000/) ，将会看到Grafana的登录界面，输入用户名/密码 :admin/admin(默认)即可登录。



点击左侧栏目中的Explore，选择对应的IP地址，点击页面中的"error"，搜索在“2020-11-27 18:01:33”日志中复制的内容，即可定位到具体的错误。



## **7.2查看日志ssh**

作为传统运维，若没有软件辅助，用户可能需要更直接方法定位错误。

* 操作方法：

1. 登录发生错误的主机IP；
2. 输出日志到“临时文件”中。

$ journalctl -u ki-client > /tmp/ki-client-log.log

$ grep /tmp/ki-client-log.log | grep -i "Operation operation-f5c70eab-0c13-4b4b-b0ae-a57bc3366881"

# 八、升级管理

注意：此操作建议在本产品生产方的运维人员指导下完成，不建议用户在未经培训的情况下进行操作。

## **8.1准备升级依赖**

### **8.1.1准备依赖的软件包**

CTSI CaaS平台升级需要准备依赖的软件包-kubelet,kubeadm,kubectl以及周边的软件包。准备软件包的思路是在在线的环境上运行以下命令：

# 在在线环境下操作

# 配置 repo cat <<EOF > /etc/yum.repos.d/kubernetes.repo

[kubernetes] name=Kubernetes baseurl=https://packages.cloud.google.com/yum/repos/kubernetes-el7-x86\_64

enabled=1

gpgcheck=1

repo\_gpgcheck=1 gpgkey=https://packages.cloud.google.com/yum/doc/yum-key.gpg https://packages.cloud.google.com/yum/doc/rpm-package-key.gpg exclude=kubelet kubeadm kubectl

EOF

# 创建一个目录存放 rpm 软件包

$ mkdir k8s-1.19.2

# 安装 yum-utils

$ yum install -y yum-utils

# 下载依赖包

$ cd k8s-1.19.2

$ yumdownloader --resolve kubelet-1.19.2-0.x86\_64 kubeadm-1.19.2-0.x86\_64 kubectl-1.19.2-0.x86\_64 ebtables ipvsadm --disableexcludes=kubernetes

$ ls

# 输出

-rw-r--r-- 1 root root 191000 4月 3 2020 conntrack-tools-1.4.4-7.el7.x86\_64.rpm -rw-r--r-- 1 root root 5318270 9月 18 10:08 cri-tools-1.13.0-0.x86\_64.rpm -rw-r--r-- 1 root root 125448 4月 25 2018 ebtables-2.0.10-16.el7.x86\_64.rpm -rw-r--r-- 1 root root 45892 4月 3 2020 ipvsadm-1.27-8.el7.x86\_64.rpm

-rw-r--r-- 1 root root 8725678 9月 18 10:08 kubeadm-1.19.2-0.x86\_64.rpm -rw-r--r-- 1 root root 9450186 9月 17 18:17 kubectl-1.19.2-0.x86\_64.rpm

-rw-r--r-- 1 root root 20438926 9月 17 18:17 kubelet-1.19.2-0.x86\_64.rpm -rw-r--r-- 1 root root 19487362 9月 19 13:07 kubernetes-cni-0.8.7-0.x86\_64.rpm -rw-r--r-- 1 root root 18400 4月 3 2020 libnetfilter\_cthelper-1.0.0-11.el7.x86\_64.rpm

-rw-r--r-- 1 root root 18212 4月 3 2020 libnetfilter\_cttimeout-1.0.0-7.el7.x86\_64.rpm

-rw-r--r-- 1 root root 23584 8月 10 2017 libnetfilter\_queue-1.0.2-2.el7\_2.x86\_64.rpm

-rw-r--r-- 1 root root 296632 8月 10 2017 socat-1.7.3.2-2.el7.x86\_64.rpm

# 将这些文见复制到 bastion(server) 上,注意如果有多台 bastion 就需要全部复制

# ssh 到 bastion(server) 上

# 创建新的 k8s 版本资源文件 x86\_64

$ mkdir -p /usr/share/k8s-installer/resource/1.19.2/centos/7/x86\_64/package # 创建新的 k8s 版本资源文件 x86\_64 arm

$ mkdir -p /usr/share/k8s-installer/resource/1.19.2/centos/7/aarch64/package # 返回在线环境

$ scp -r k8s-1.19.2/\* xxx@bastionip:/usr/share/k8s-installer/resource/1.19.2/centos/7/aarch64/package/

注意：相同的操作系统版本,不要运行 yum update -y或者其他更新命令；需要能够访问google的网络环境。

### **8.1.2准备依赖的镜像**

当在线环境已经准备好了依赖包，需要通过kubeadm功能下载所需的镜像。

# 在在线环境下操作

# 安装 kubeadm

$ yum install kubeadm-1.19.2-0.x86\_64 --disableexcludes=kubernetes -y

# 下载镜像,以下命令会自动的下载所依赖的镜像

$ kubeadm config image pull

# 导出镜像

$ docker save k8s.gcr.io/kube-proxy:1.19.2 -o kube-proxy.tar

$ docker save k8s.gcr.io/kube-apiserver:1.19.2 -o kube-apiserver.tar

$ docker save k8s.gcr.io/kube-controller-manager:1.19.2 -o kube-controller-manager.tar

$ docker save k8s.gcr.io/kube-scheduler:1.19.2 -o kube-scheduler.tar

$ docker save k8s.gcr.io/etcd:3.4.13-0 -o etcd.tar

# 复制镜像到私有仓库

# 假设您的私有仓库地址为 10.0.0.100:4000

$ scp kube-proxy.tar kube-apiserver.tar kube-controller-manager.tar kube-scheduler.tar etcd.tar [user]@10.0.0.100:/path

# 将新版本的 kubeadm 二进制文件复制到 "bastion上的资源服务的路径"

$ scp /usr/bin/kubeadm xxx@bastionip:/usr/share/k8s-installer/resource/1.19.2/centos/7/aarch64/package/

# 在私有仓库中执行以下命令

# 在私有仓库中加载镜像

$ docker load -i kube-proxy.tar

$ docker load -i kube-apiserver.tar

$ docker load -i kube-controller-manager.tar

$ docker load -i kube-scheduler.tar $ docker load -i etcd.tar # 重命名镜像 $ docker tag k8s.gcr.io/kube-proxy:1.19.2 10.0.0.100:4000/kube-proxy:1.19.2

$ docker tag k8s.gcr.io/kube-apiserver:1.19.2 10.0.0.100:4000/kube-apiserver:1.19.2

$ docker tag k8s.gcr.io/kube-controller-manager:1.19.2 10.0.0.100:4000/kube-controller-manager:1.19.2

$ docker tag k8s.gcr.io/kube-scheduler:1.19.2 10.0.0.100:4000/kube-scheduler:1.19.2

$ docker tag k8s.gcr.io/etcd:3.4.13-0 10.0.0.100:4000/etcd:3.4.13-0

# 推送修改过的镜像到私有仓库

$ docker push 10.0.0.100:4000/kube-proxy:1.19.2

$ docker push 10.0.0.100:4000/kube-apiserver:1.19.2

$ docker push 10.0.0.100:4000/kube-controller-manager:1.19.2

$ docker push 10.0.0.100:4000/kube-scheduler:1.19.2

$ docker push 10.0.0.100:4000/etcd:3.4.13-0

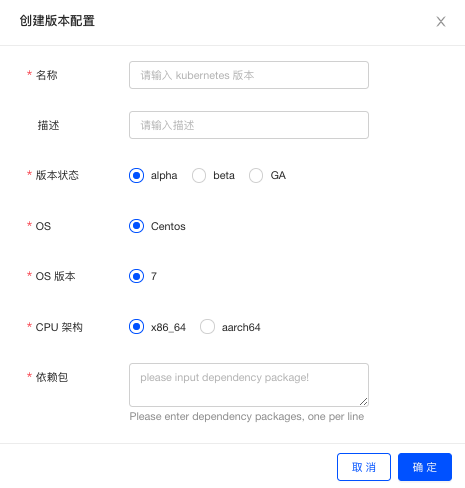
## **8.2版本配置**

当完成所有新版本的升级依赖，用户可以通过界面配置可升级的版本。点击左侧菜单“升级管理>版本配置”，可查看版本配置界面，查看版本名称、版本状态、描述、更新时间、修改者、创建时间。



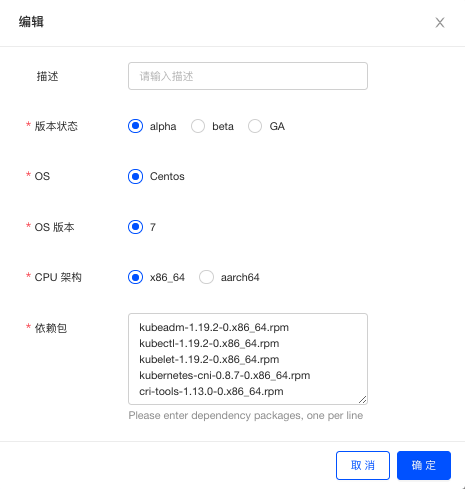
### **8.2.1创建版本配置**

点击界面上方的“创建版本配置”按钮，出现弹窗。在依赖包内输入所有在之前 "准备依赖的软件包"，如 "conntrack-tools-1.4.4-7.el7.x86\_64.rpm"。



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 作用说明 | 验证 | 是否可以缺省 | 可选值 | 默认值 |
| 名称/name | 审计版本的名称 | k8s版本认证表达式 `\+.)?(\d+.)?(\* | \d+)$` 比如 1.18.10,1.19.2 | 否 | N/A |
| 描述/description | k8s集群的ipv 4的cluster id所在网段 | CIDR | 否 | N/A | 10.96.0.0/16 |
| 版本状态 | alpha=alpha试用,beta=beta试用,GA=Gerneral avaliable可用,这里只是标注标注而已并无实际意义 | N/A | 否 | alpha,beta,GA | alpha |
| OS | 操作系统,目前只支持 Cetnos | N/A | 否 | Centos | Centos |
| OS 版本 | 操作系统版本,目前只支持 Cetnos-7.x | N/A | 否 | 7 | 7 |
| CPU架构 | CPU架构 | N/A | 否 | x86\_64,aarch64 | x86\_64 |
| 依赖包 | 依赖包列表,通过点击 "+" 添加更多以来,这里你需要输入之前 "准备依赖的软件包",比如 "conntrack-tools-1.4.4-7.el7.x86\_64.rpm" | N/A | 否 | x86\_64,aarch64 | x86\_64 |

### **8.2.2编辑版本配置**

当版本配置被创建出来之后，选择目标版本配置右侧“操作”栏中的“编辑”，出现弹窗，可编辑描述、版本状态、OS、OS版本、CPU架构、依赖包。点击确定即可。

### **8.2.3删除版本配置**

点击目标版本配置右侧“操作”栏中的“删除”按钮，出现确认弹窗，点击确定即可删除。

注意：如果该版本正在被某些集群做为升级操作时将无法删除。



## **8.3升级计划**

当用户完成成了 "准备依赖包"，"准备镜像"，"配置可升级版本"后，可以开始准备升级。

在升级之前平台会根据 "可升级版本" 对应的包去检查选中的集群是否满足升级条件。当满足时系统会在页面上显示集群待升级的记录，等待用户确认升级。当不满足时界面将显示其原因。

用户点击左侧菜单中的“升级管理>升级计划”，可查看升级计划的集群ID、目标版本、状态、操作ID、升级时间。



注意：当被升级的目标节点只有一个master，这种情况在升级的时，替换镜像的过程中api-server会有一段时间不可用。

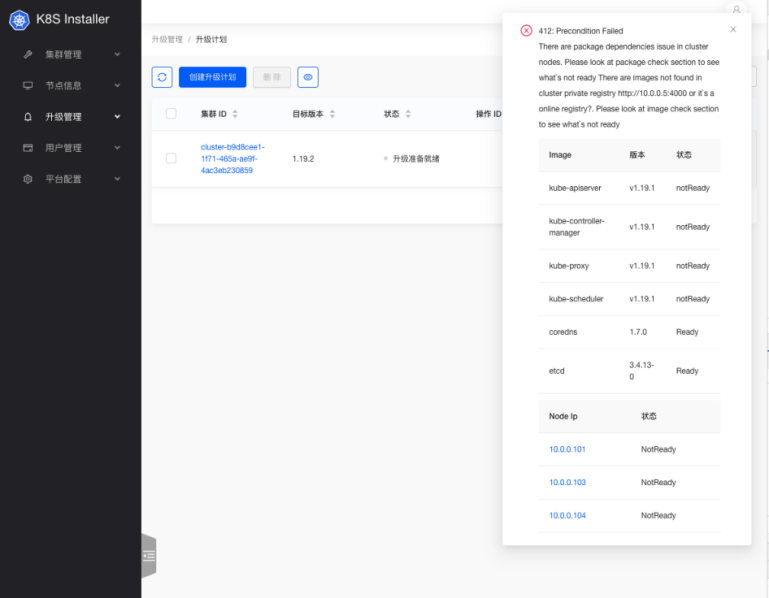
### **8.3.1创建升级计划**

点击左上角的“创建升级计划”按钮，然后在弹出的对话框中选择需要升级的 "集群" 和 "所要升级的版本"，点击确定即可完成创建。



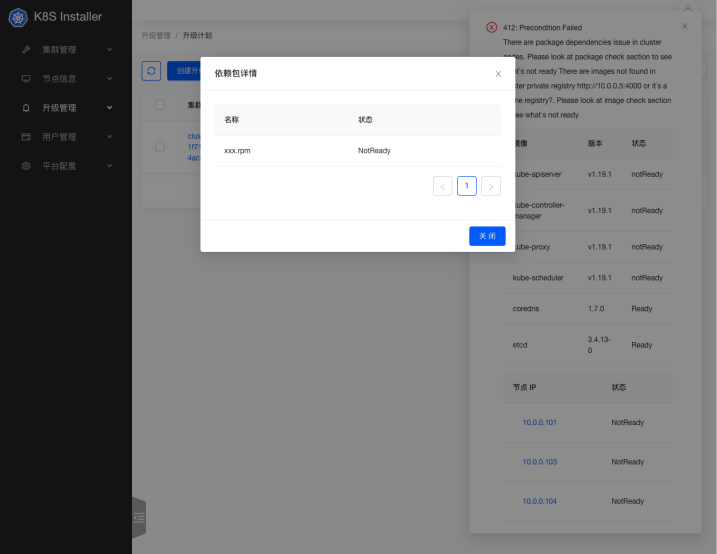
### **8.3.2升级**

当用户完成“创建升级计划”后，会出现一条“待升级”记录。若创建时检测失败，将明确告知用户哪一步出现错误。

* 检查镜像：获取cluster的私有镜像仓库，查询和"目标升级版本"匹配的镜像，如用户选择升级"1.19.2"，那么程序将会检查集群的"私有镜像仓库"中是否存在 "kube-apiserver:v1.19.2"等镜像。
* 检查依赖包：根据用户之前配置的"可升级的版本"中输入的依赖包，根据每个节点的操作系统和版本去检查bastion(server)上是否存在这些安装包。

在右侧用户可以看到镜像是否达标，若不达标将显示没有在镜像仓库中找到的镜像信息。

若有 "节点" 无法找到依赖包将会显示节点的"ip"的链接，点击链接出现弹窗，显示具体没有准备好的依赖包。



注意：由于安全原因，平台暂时阻止了使用公有镜像仓库的集群的升级功能，如果用户在线环境使用了gcr.k8s.io，将无法进行升级计划。

# 九、用户管理

平台在搭建好后默认为用户配置好2个"角色/role"，分别为 "admin" 和 "user" ，此外还有一个系统内置帐号 "admin" 的存在。

注意：admin用户和admin角色作为内置帐号不允许被删除。

## **9.1用户**

可以通过点击菜单 "用户管理>用户"，查看用户列表。可查看用户名、角色。



### **9.1.1创建用户**

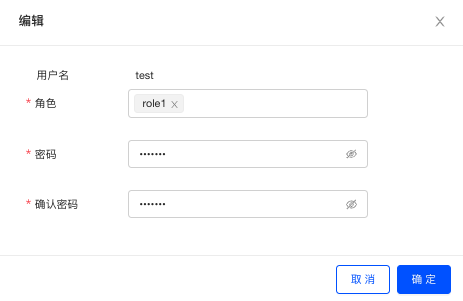
点击左上角按钮“创建用户”，出现弹窗，填写用户名、角色、密码以及确认密码，点击确定即可创建用户。



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 字段 | 作用说明 | 验证 | 是否可以缺省 | 可选值 | 默认值 |
| 用户名/username | 用户登录名 | N/A | 否 | N/A | N/A |
| 角色名称/role name | 角色名称,这里可以多选,比如 role 有权限 "创建集群", role2 有权限 "删除集群",那么如果你同时选择了 role 和 role2 那么这个帐号将有 "创建集群 和 删除集群" 的权限 | N/A | N/A | N/A | N/A |
| 密码/password | 密码 | N/A | 否 | N/A | N/A |
| 重复密码/password | 确认密码 | N/A | 否 | N/A | N/A |

### **9.1.2编辑用户**

点击目标用户右侧“操作”栏中的“编辑”按钮，出现编辑弹窗，可编辑角色、密码。

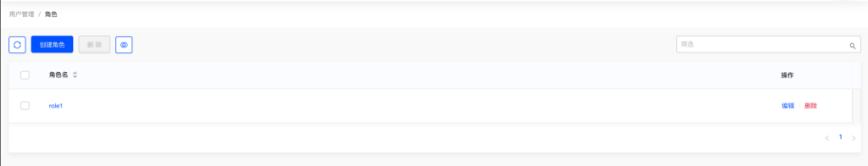


### **9.1.3删除用户**

点击目标用户右侧的“删除”按钮，出现二次弹窗，点击确定即可删除。系统中存在的admin角色无法被删除。



## **9.2角色**

点击左侧菜单中的“用户管理>角色”，可查看角色列表。可查看角色名称。

### **9.2.1创建角色**

点击页面右上角的“创建角色”按钮，出现弹窗，填写角色名以及权限可创建角色。

* 权限说明：

集群：

1. 查看：查看详细列表的权限
2. 创建：创建集群的权限
3. 删除 ：删除集群的权限
4. 添加节点：往运行中的集群添加节点的权限
5. 移除节点：从运行中的集群移除节点的权限
6. 继续失败任务：继续失败任务的权限

节点：

1. 查看：查看详细列表的权限
2. 开启关闭：禁用和启用节点的权限

用户：

1. 管理：管理用户增删改查的权限

角色：

1. 管理：管理角色增删改查的权限

升级：

1. 查看：查看详细列表的权限
2. 版本管理：管理可升级版本的权限
3. 升级管理：升级集群的权限

平台配置：

1. 平台配置：配置平台默认参数的权限
2. 证书管理：管理证书的权限



### **9.2.2编辑角色**

点击目标角色右侧的“编辑”按钮，可编辑角色的权限。

### **9.2.3删除角色**

点击目标角色右侧“删除”按钮，出现二次弹窗，点击确定即可删除角色。

# 十、平台配置

## **10.1平台默认参数设置**

点击左侧菜单中“平台配置>平台配置”，做为管理员需要修改这些默认的参数，方便后期普通用户可以方便的使用 CTSI CaaS平台。

镜像仓库为私有镜像仓库，用户可以添加多条记录。在部署的生产环境中默认会有3个，每次安装集群时前端界面都会随机的把其中一个镜像仓库变为默认镜像仓库，这样可以缓解由于单个镜像仓库压力过大的问题。

## **10.2证书管理**

用户在购买平台后，供应商将会向用户发出一份对应的证书，这份证书包含以下信息：

1. 证书有效期
2. 证书允许的节点数
3. 证书允许的CPU总数
4. 证书允许的内存总数

例如：

证书允许：1600 CPU 10000G 内存 50 个节点

用户环境：40 个节点,每个节点 32 个 CPU 和 150G 内存

结果：证书有效

证书允许：1600 CPU 10000G 内存 50 个节点

用户环境：60 个节点,每个节点 12 个 CPU 和 100G 内存

结果：证书无效，原因：节点总数超出了上限(50)

证书允许：1600 cpu 10000G 内存 50 个节点

用户环境：30 个节点,每个节点 128 个 cpu 和 100G 内存

结果：证书无效，原因： CPU总数 超出了上线。

证书失效后并 不会影响现有集群和业务的运行 ,只是整个管理平台将不可以使用.

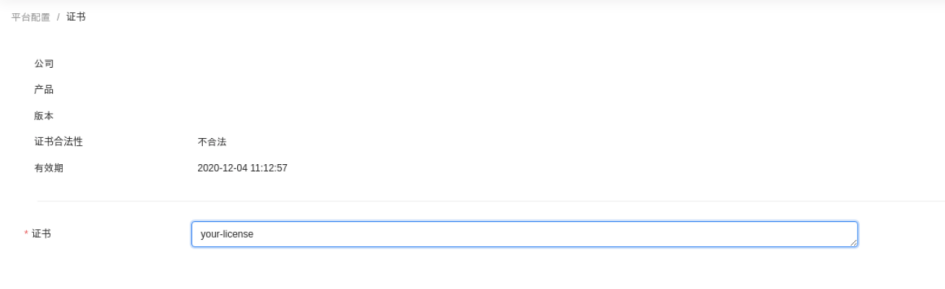
注意：请不要自行通过kubeadm join加入未经注册的节点，会造成证书不可用，导致平台无法管理。

### **10.2.1更新证书**

当用户证书失效时，可以通过菜单 "平台配置>证书" ，在界面“证书”输入框中输入证书信息。点击右下角的更新证书即可。

### **10.2.2上传新的证书（证书有效）**

当您的证书失效的时候,你可以通过菜单 "平台配置" -> "证书" 然后在证书输入框中输入您的证书,如下图。

完成后点击右下角的 "更新证书" 按钮。

### **10.2.3上传新的证书（证书已失效）**

由于证书已经生效了，这个时候您已无法从界面上更新证书了，这时需要链接到平台的任何一台 bastion 节点，并运行以下命令。

```console

$ ETCDCTL\_API=3 etcdctl --endpoints=https://ki-etcd1:12379 --cert=/etc/ki-etcd/pki/etcd-client.crt --key=/etc/ki-etcd/pki/etcd-client.key --cacert=/etc/ki-etcd/pki/etcd-ca.crt put /license/ "{\"license\":\"替换为新证书\"}"

# 返回 1 表明成功

1

```

# 十一、pod网段的计算

不同的网络插件分配的地址池是截然不同的，每个cni组件都要实现自己的IPAM(IP allocation management）

## **11.1Calico IPAM**

Calico将用户设置的pod cidr分为若干个block(网段)，根据业务需求动态的分配给需要的节点，并在节点中通过 bgp peer维护集群中的节点的路由表。

例如：

容器的地址池: 172.25.0.0/16，动态分配的网段池: 172.25.0.0 - 172.25.255.192 (172.25.0.0/26 即 10 个比特位)，动态分配的网段数: 1023，每个网段的pod数量为: 61 (193-254)，总pod数量为1023 \* 61 = 62403，相对最大节点数(按照200业务pod为基准值)：312。

目前不建议大于50个节点的集群，大于50个节点的集群建议手动配置route reflection，用来优化集群中的节点的路由表维护的稳定性。

## **11.2Calico 网络模式**

默认使用calico作为网络插件时，平台会给若干个选项：

* Overlay-IPIP-All: 使用 IP-in-IP 技术打通不同节点的 pod 的网络,通常这样的方式使用在底层平台是 iaas 的环境之中,当然如果你底层网路环境直接是物理设备的也完全可以使用只不过效率和灵活度都会大打折扣,需要注意的是你需要确认底层网络环境(underlay)是支持 IPIP 协议的.(使用overlay的网络方式对网络性能造成一定的影响)
* Overlay-Vxlan-All: 使用 IP-in-IP 技术打通不同节点的 pod 的网络,通常这样的方式使用在底层平台是 iaas 的环境之中,当然如果你底层网路环境直接是物理设备的也完全可以使用只不过效率和灵活度都会大打折扣,他理论上可以在任何的网络环境上运行,通常在底层环境不支持 IPIP 协议的时候我们会使用他.(使用overlay的网络方式对网络性能造成一定的影响)
* BGP: 使用 IP-in-IP 技术打通不同节点的 pod 的网络,通常这样的方式使用在裸机的环境上,当然底 Iaas 平台支持 BGP 的话也是可以使用的,这种模式下 pod 的 ip 通信是通过 集群中的各个节点中互相交换路由表来完成 pod 之间的通信的,如果你需要手动打通多个集群之间的 pod 网络需要注意你分配的地址断不应该有冲突
* Overaly-IPIP-Cross-Subnet: 使用 IP-in-IP 技术打通不同节点的 pod 的网络,通常这样的方式使用在底层平台是 iaas 的环境之中,需要注意的是你需要确认底层网络环境(underlay)是支持 IPIP 协议的.和 Overlay-IPIP-All 的不同之处在于,如果 2 个不同节点但在同一个网段中的上 pod 互相通信时是通过路由表,这样可以提高在不同节点但在同一个网段中的上 pod 互相通信时的效率
* Overaly-Vxlan-Cross-Subnet: 和 Overaly-IPIP-Cross-Subnet 逻辑相似不在做重复的解释

## **11.3Calico 网络模式之节点数量**

calico 使用 bgp 协议交换路由当模式设为 :Overlay-IPIP-All/Overlay-IPIP-Cross-Subnet/BGP(vxlan不使用bgp协议) ,默认使用 full-mesh 算法这种情况下如果节点数量大于 100 个的时候会造成性能下降，建议手动配置 route reflection(目前产品没有自己配置 route reflection 功能，将在将来产品版本中加入)

|  |  |  |  |
| --- | --- | --- | --- |
| Calico模式 | 建议最大集群节点数(full-mesh) | 建议最大集群节点数(route-reflection) | 性能等级(more stars more better) |
| Overlay-IPIP-All/Overlay-IPIP-Cross-Subnet | 100 | 1000 | \* \* \* \* |
| Overlay-Overlay-All/Overlay-Overlay-Cross-Subnet | 500 | N/A | \* \* \* |
| BGP | 100 | 1000 | \* \* \* \* \* |

## **11.4硬件资源**

注意：bastion(server)的CPU和RAM是由被纳管的节点数量所决定的。

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 节点 | CPU | RAM | OS Disk | Pod Data Disk | Etcd Disk | Docker Disk or Conatianerd Disk |
| bastion1 | 8 | 16G | 1X100G SSD mount / | N/A | 1X150G mount /etc/etcd/ | 1X100G mount /var/lib/docker/ |
| bastion2 | 8 | 16G | 1X100G SSD mount / | N/A | 1X150G mount /etc/etcd/ | 1X100G mount /var/lib/docker/ |
| bastion3 | 8 | 16G | 1X100G SSD mount / | N/A | 1X150G mount /etc/etcd/ | 1X100G mount /var/lib/docker/ |
| client N(建议少于500个,包括将来部署的 k8s 节点和外部的负载均衡) | 8 | 32G | 1X100G SSD mount / | 1X100G mount /var/lib/kubelet/ | N/A | 1X150G mount /var/lib/docker/ |

## **11.5建议规模**

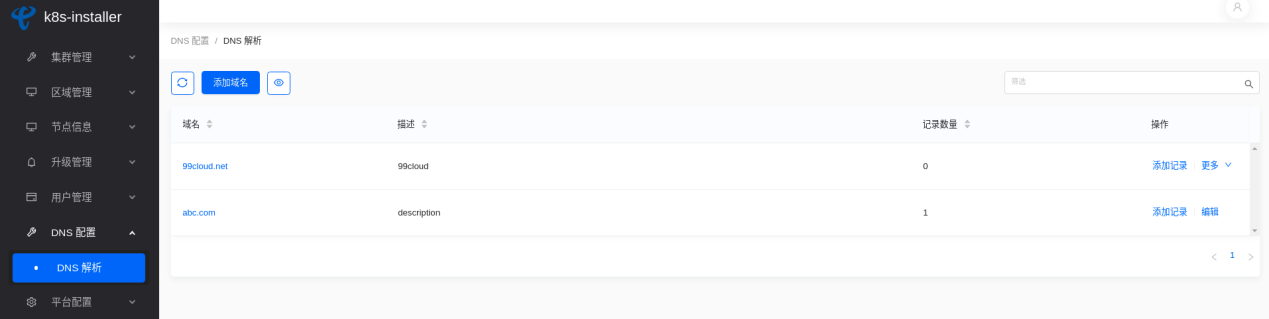
|  |  |  |  |
| --- | --- | --- | --- |
| 被纳管的节点数量(client) | 建议 bastion 数量 | 建议CPU数量(单台) | 建议RAM(G)数量(单台) |
| 0-1000 | 3 | 8 | 16 |
| 1001-3000 | 5 | 16 | 32 |
| 3001-5000 | 7 | 32 | 64 |

## **11.6 平台 dns 配置和使用**

* 注意事项1：在使用平台的 dns 需要注意的切勿不可和 k8s 集群的内域名发生冲突，比如你创建了1条 test.default.svc.cluster.local 的域名并解析到 1.1.1.1 那么如果 集群A 中正好有一个服务叫 test 他在 default 的 namespace 并且该集群的 domain 使用默认的 cluster.local 那么他们的就会产生冲突，在这个集群里对 test.default.svc.cluster.local 永远也不会解析到 1.1.1.1 上 。
* 注意事项2：dns 解析负载均衡默认 关闭，比如 [www.test.com](http://www.test.com/) 你编写了 3 条解析 10.0.0.100 10.0.0.101 10.0.0.102，那么对这个地址的解析请求会平均的分摊的到这个3个地址上，如果需要开启可联系供应商提供服务。
* 注意事项3：避免3个以上对同一个条域名的解析，比如 [www.test.com](http://www.test.com/) 你编写了 3 条解析 10.0.0.100 10.0.0.101 10.0.0.102 ,这样是 ok 的但是不建议超过 3 条的解析，这样在开启地址负载均衡时会影响 dns 算法效率。
* 注意事项4：泛域名的层级，平台 dns 提供对泛域名的支持，比如你输入 \*.text.com -> 10.0.0.1 ,之后你又创建 abc.test.com -> 10.0.0.200，那么对 abc.test.com 的解析请求 dns 会应答 10.0.0.200, 如果对 zzz.abc.com 进行解析请求那么 dns 会返回 10.0.0.1 ，同理 aaa.bbb.abc.com , ggg.aaa.bbb.abc.com 但是需要注意的是我们应该避免这样额多层级的域名，会减低 dns 相应的速度。

### **11.6.1创建域名组**

a. 点击 "DNS 配置",您会看到如下的页面

b.点击做上角的 "添加域名" 按钮，你会看到弹出如下对话框



### **11.6.2添加域名和地址解析**

a. 点击你要操作的 域名组 的右侧按钮 "添加记录"



### **11.6.3删除地址解析**

a. 您可以点击记录前的 "-" 按钮来删除解析，，只需要注意的是如果删除了所有额地址解析，那么这个 域名 将自动被删除

### **11.6.4删除域名**

a. 您可以选中域名的单选框后点击 "删除" 按钮，一并删除他们



### **11.6.5删除域名组**

a. 点击你要删除的 域名组 的右侧按钮 "更多" -> "删除"， 注意删除 域名组 前你必须删除其所有的域名解析，否则程序将阻止你



# 十二、管理员须知

## **12.1 关于"从错误位置继续"并不是万能**

在创建集群时，当集群搭建出现错误的时候，可以点击“从错误位置继续”，但是在特定的情况下还是会报错，并不能继续执行下去。如果 “从错误位置继续”不能解决问题那么可以通过删除集群重新创建，以下场景是点击“从错误位置继续”不能解决问题的情况：

* 缺少某些依赖文件，在具体的错误节点上，管理员会看到缺失的软件包，需要联系平台开发方将缺失的包补上；
* 在 join control plane 的时候集群的时候出现了 "download certs timeout" 的错误,这种通常都是网络不稳定造成的。其他节点从第一个 master 上下载证书超时，通常这个位置发生了错误，即使 "从错误位置继续" 也不能一定能解决问题。考虑以下场景：如果有2个节点等待加入集群，第一个成功了，然而第二失败了，那么任务就会停止，等待处理。点击“从错误位置继续”会重新尝试将这2个节点加入到集群，这里就将发生错误，因为第一个节点已经加入到集群中，无法再次加入集群。发生这样情况，可以通过两种方式来解决 ：

1. 删除集群然后重新安装 。
2. ssh 到第一台 master 上手动运行 kubectl delete node xxxxx，然后点击 "从错误位置继续"。

在更新集群的时候,为了满足稳定性,每个节点更新升级都是逐个完成的,而不是并发同时更新的，可以放心的使用 "从错误位置继续"

注意：不要尝试修改已经安装了集群的主机的主机名,会造成集群瘫痪的问题；

不要尝试推送新的镜像到 bastion 的私有仓库,在 bastion 上的私有仓库通常都存有平台搭建集群需要的镜像,随意的推送镜像到私有仓库会造成同名同版本的覆盖,造成集群管理功能的不稳定。

## **12.2 关于集群默认存储的问题**

Caas 平台目前支持 Cinder、NFS、Ceph 作为 K8S 集群的 StorageClass，且同时存在多个。**但是，我们在 K8S 集群中选择安装 Pass 平台的时候，由于 Paas 平台在集群中无法兼容多个默认存储，会导致安装失败**。因此我们建议在安装带有 Paas 平台的 K8S 集群的时候，**只选择一个存储作为 StorageClass 即可。如果存在多个存储，且存在 cinder 存储，那么其他存储将无法被设置为默认存储。**