# Alibaba Cloud Container Service for Kubernetes

使用者指南

檔案版本:20180929

为了无法计算的价值 | [-] 阿里云

# 目錄

1	Kubernete	s 叢集	1
	1.1 簡介		1
	1.1.1	概述	. 1
	1.1.2	阿里雲 Kubernetes VS 自建 Kubernetes	. 1
	1.2 授權管3	里	. 3
	1.2.1	使用子帳號	. 3
	1.2.2	建立自訂授權策略	5
	1.2.3	子帳號Kubernetes應用許可權配置指導	. 8
	1.3 叢集管3	里	10
	1.3.1	查看叢集概覽	10
	1.3.2	建立Kubernetes叢集	11
	1.3.3	VPC下 Kubernetes 的網路位址區段規劃	19
	1.3.4	通過 kubectl 串連 Kubernetes 叢集	22
	1.3.5	SSH訪問Kubernetes叢集	23
	1.3.6	SSH金鑰組訪問Kubernetes叢集	25
	1.3.7	升級叢集	27
	1.3.8	擴容和縮容叢集	29
	1.3.9	自動調整叢集	30
	1.3.1	0 刪除叢集	38
	1.4 節點管3	里	40
	1.4.1	添加已有節點	40
	1.4.2	查看節點列表	44
	1.4.3	節點監控	45
	1.4.4	節點標籖管理	47
	1.5 命名空	間管理	49
	1.5.1	設定資源配額和限制	50
	1.5.2	編輯命名空間	52
	1.6 應用管3	里	54
	1.6.1	使用鏡像建立無狀態Deployment應用	54
	1.6.2	通過 Kubernetes Dashboard 建立應用	69
	1.6.3	通過編排模板建立應用	71
	1.6.4	通過命令管理應用	75
	1.6.5	建立服務	75
	1.6.6	服務伸縮	80
	1.6.7	查看服務	82
	1.6.8	更新服務	83
	1.6.9	刪除服務	87
	1.6.1	0 查看容器	88
	1.6.1	1	91
	1.6.1	2	92
	1.6.1	3	96

1.7 負載平衡及路由管理	97
1.7.1 概述	97
1.7.2 通過負載平衡(Server Load Balancer)訪問服務	97
1.7.3 Ingress 支援	103
1.7.4 通過 Web 介面建立路由	108
1.7.5 變更路由	118
1.7.6 查看路由	119
1.7.7 刪除路由	120
1.8 配置項及密鑰管理	121
1.8.1 建立配置項	121
1.8.2 在 pod 中使用配置項	125
1.8.3 修改配置項	128
1.8.4 刪除配置項	132
1.8.5 建立密鑰	133
1.8.6 查看密鑰	135
1.8.7 編輯密鑰	136
1.8.8 刪除密鑰	137
1.9 儲存管理	138
1.9.1 概述	139
1.9.2 安裝外掛程式	139
1.9.3 使用阿里雲雲端硬碟	143
1.9.4 使用阿里雲 NAS	148
1.9.5 使用阿里雲 OSS	155
1.9.6 建立持久化儲存卷聲明	159
1.9.7 使用持久化儲存卷聲明	161
1.10 日誌管理	163
1.10.1 概述	163
1.10.2 查看叢集日誌	163
1.10.3 為 Kubernetes 和Log Service配置 Log4JAppender	164
1.10.4 利用 log-pilot + elasticsearch + kibana 搭建 kubernetes 日誌解決方案	169
1.10.5 Kubernetes與Log Service整合與使用	175
1.11 監控管理	179
1.11.1 與Cloud Monitor整合與使用	179
1.11.2 使用 Grafana 展示監控資料	184
1.11.3 使用HPAAuto Scaling容器	189
1.11.4 通過資源分組進行監控與警示	193
1.12 安全管理	197
1.12.1 概述	197
1.12.2 Kube-apiserver審計日誌	199
1.13 發行管理	203
1.13.1 基於Heml的發行管理	203
1.14 Istio管理	205
1.14.1 概述	205
1.14.2 部署Istio	207

1.14.3 更新Istio	212
1.14.4 刪除Istio	214
1.15 應用目錄管理	
1.15.1 應用目錄概述	
1.15.2 查看應用目錄列表	
1.16 服務類別目錄管理	217
1.16.1 概述	
1.16.2 開通服務類別目錄	218

# 1 Kubernetes 叢集

# 1.1 簡介

# 1.1.1 概述

Kubernetes 是流行的開源容器編排技術。為了讓使用者可以方便地在阿里雲上使用 Kubernetes 管 理容器應用,阿里雲Container Service提供了 Kubernetes 叢集支援。

您可以通過Container Service管理主控台建立一個安全高可用的 Kubernetes 叢集,整合阿里雲虛 擬化、儲存、網路和安全能力,提供高效能可伸縮的容器應用管理能力,簡化叢集的搭建和擴容等 工作,讓您專註於容器化的應用的開發與管理。

Kubernetes 支援對容器化應用程式的部署、擴充和管理。它具有以下功能:

- 彈性擴充和自我修複
- 服務發現和負載平衡
- 服務發布與復原
- 機密和組態管理

## 使用限制

- 目前 Kubernetes 叢集只支援 Linux 容器,對 Kubernetes 的 Windows 容器的支援在計劃中。
- 目前 Kubernetes 叢集只支援 VPC 網路。您可以在部署 Kubernetes 叢集時選擇建立一個新的 VPC 或者使用已有的 VPC。

## 相關開源項目

- 阿里雲 Kubernetes Cloud Provider 實現: https://github.com/AliyunContainerService/ kubernetes
- Flannel 的阿里雲 VPC 網路驅動: https://github.com/coreos/flannel/blob/master/Documentation /alicloud-vpc-backend.md

如果您對相關項目有問題或者建議, 歡迎在社區提交 Issue 或者 Pull Request。

# 1.1.2 阿里雲 Kubernetes VS 自建 Kubernetes

# 阿里雲 Kubnernetes 的優勢

# 便捷

• 通過 Web 介面一鍵建立 Kubernetes 叢集。

• 通過 Web 介面一鍵完成 Kubernetes 叢集的升級。

您在使用自建 Kubernetes 叢集的過程中,可能需要同時處理多個版本的叢集(包括 1.8.6、1.9 .4、以及未來的 1.10)。每次升級叢集的過程都是一次大的調整和巨大的營運負擔。Container Service的升級方案使用鏡像滾動升級以及完整元資料的備份策略,允許您方便地復原到先前版 本。

• 通過 Web 介面輕鬆地實現 Kubernetes 叢集的擴容和縮容。

使用Container Service Kubernetes 叢集可以方便地一鍵垂直伸縮容來快速應對資料分析業務的 峰值。

功能	說明
網路	<ul> <li>高效能 VPC 網路外掛程式。</li> <li>支援 network policy 和流控。</li> <li>Container Service可以為您提供持續的網路整合 和最佳的網路最佳化。</li> </ul>
負載平衡	支援建立Server Load Balancer執行個體(公 網、內網)。 如果您在使用自建 Kubernetes 叢集的過程使 用自建的 Ingress 實現,業務發布頻繁可能會 造成 Ingress 的配置壓力並增加出錯機率。 Container Service的 SLB 方案支援原生的阿里 雲高可用負載平衡,可以自動完成網路設定的修 改和更新。該方案經曆了大量使用者長時間的使 用,穩定性和可靠性大大超過使用者自建的入口 實現。
儲存	整合阿里雲雲端硬碟、Network Attached Storage、區塊存放裝置EBS,提供標準的 FlexVolume 驅動。 自建 Kubernetes 叢集無法使用雲上的儲存資 源,阿里雲Container Service提供了最佳的無縫 整合。
營運	<ul> <li>整合阿里雲Log Service、Cloud Monitor</li> <li>支援Auto Scaling</li> </ul>
鏡像倉庫	<ul> <li>高可用,支援大並發。</li> <li>支援鏡像加速。</li> <li>支援 p2p 分發。</li> </ul>

強大

功能	說明
	您如果使用自建的鏡像倉庫,在百萬級的用戶端 同時拉取鏡像的時候,會存在鏡像倉庫崩潰的可 能性。使用Container Service鏡像倉庫的專有雲 版本來提高鏡像倉庫的可靠性,減少營運負擔和 升級壓力。
穩定	<ul> <li>專門的團隊保障容器的穩定性。</li> <li>每個 Linux 版本,每個 Kubernetes 版本都 會在經過嚴格測試之後之後才會提供給使用 者。</li> </ul>
	Container Service提供了 Docker CE 兜底和 推動 Docker 修複的能力。當您遇到 Docker Engine hang、網路問題、核心相容等問題時, Container Service可以為您提供最佳實務。
高可用	<ul><li>• 提供多可用性區域支援。</li><li>• 支援備份和容災。</li></ul>
支援人員	<ul> <li>提供 Kubernetes 升級能力,新版本一鍵升級。</li> <li>阿里雲容器團隊負責解決在環境中遇到的各種容器問題。</li> </ul>

#### 自建 Kubernetes 的成本和風險

• 搭建叢集繁瑣。

您需要手動設定 kubernetes 相關的各種組件、設定檔、認證、密鑰、相關外掛程式和工具,整個叢集搭建工作需要專業人員數天到數周的時間。

• 在公用雲上,需要投入大量的成本實現和雲產品的整合。

和阿里雲上其他產品的整合,需要您自己投入成本來實現,如Log Service、監控服務和儲存管 理等。

- 容器是一個系統性工程,涉及網路、儲存、作業系統、編排等各種技術,需要專門的人員投入。
- 容器技術一直在不斷髮展,版本迭代快,需要不斷的踩坑、升級、測試。

# 1.2 授權管理

# 1.2.1 使用子帳號

使用子帳號登入Container Service控制台並進行相關操作之前,您需要賦予子帳號相應的許可權。

步驟1建立子帳號並開啟控制台登入

- 1. 登入 #########。
- 2. 單擊左側導覽列中的使用者管理並單擊頁面右上方的建立使用者。
- 3. 填寫子帳號的名稱並單擊確定。
- 4. 在使用者管理頁面,選擇建立的子帳號,單擊右側的管理。
- 5. 在Web控制台登入管理中, 單擊啟用控制台登入。
- 6. 輸入登入密碼並單擊確定。
- 步驟 2 授予子帳號訪問Container Service的許可權
  - 1. 在使用者管理頁面,選擇建立的子帳號,單擊右側的授權。

用户管理			新建用户 〇 刷新
用户名 🔻 teststs	搜	·	
用户名/显示名	备注	创建时间	操作
teststs		2017-12-05 17:04:04	管理 授权 删除 加入组

2. 將所需的策略授權給子帳號。

添加授权策略后,该账户即具有该条领	<b>策略的权限</b>	, 同·		能被重复添加。		
可选授权策略名称		긜		已选授权策略名称		类型
CS		Q		AliyunCSFullAccess		系统
管理Elasticsearch的权限		•		管理容器服务(CS)的权限		
AliyunCSFullAccess -	自定义		>			
AliyunACSResourcesAccess_xingy aliyun container s	自定义		<			
AliyunCSReadOnlyAccess 只读访问容器服务(CS)的权限	系统					
		•				
					确定	关闭

# 您可以使用系統預設的授權策略。

- AliyunCSFullAccess: Container Service的系統管理權限。
- AliyunCSReadOnlyAccess: Container Service的唯讀許可權。

或者根據您自己的需要自訂授權策略並授權給子帳號,參見########。

#### 步驟 3 子帳號登入Container Service控制台

使用子帳號登入 Container Service#####。

如果您之前已經給主帳號授予了 AliyunCSDefaultRole 和 AliyunCSClusterRole 角色,子帳號可以 直接登入到Container Service管理主控台,並進行相應的操作。

如果之前您沒有給主帳號授予 AliyunCSDefaultRole 和 AliyunCSClusterRole 角色,則會看到如下 提示。單擊同意授權。

云资源访问授权	
温馨提示:如需修改角色权限,请前往RAM控制台角色管理中设置,需要注意的是,错误的配置可能导致CS无法获取到必要的权限。	×
C3時來飲稅以同時16公支通貨的状況	
AliyunCSDefaultRole 描述: 容器服务默认使用此角色来访问您在其他云产品中的资源	~
权限描述:用于容器服务(CS)默认角色的授权策略	
AliyunCSClusterRole	~
描述: 容器服务集群使用此角色来访问您在其他云产品中的资源	
权限面述: 用于容器服务(CS) Cluster 角色的技权策略	
同意授权取消	

完成以上授權後,重新整理Container Service控制台,然後就可以進行操作了。

# 1.2.2 建立自訂授權策略

Container Service提供的系統授權策略的授權粒度比較粗,如果這種粗粒度授權策略不能滿足您的 需要,那麼您可以建立自訂授權策略。比如,您想控制對某個具體的叢集的操作許可權,您必須使 用自訂授權策略才能滿足這種細粒度要求。

建立自訂授權策略

在建立自訂授權策略時,您需要瞭解授權策略語言的基本結構和文法,相關內容的詳細描述請參 考########。

本文檔以授予子帳號查詢、擴容和刪除叢集的許可權為例進行說明。

操作步驟

- 1. 使用主帳號登入RAM#####。
- 2. 單擊左側導覽列中的策略管理並單擊頁面右上方的建立授權策略。
- 3. 選擇一個模板,填寫授權策略名稱並編寫您的授權策略內容。



```
{
  "Statement": [{
    "Action": [
        "cs:Get*",
        "cs:ScaleCluster",
        "cs:DeleteCluster"
    ],
    "Effect": "Allow",
    "Resource": [
        "acs:cs:*:*:cluster/叢集ID"
    ]
}],
"Version": "1"
}
```

其中:

• Action 處填寫您所要授與權限。



• Resource 有如下配置方式。

- 授予單叢集許可權

```
"Resource": [
"acs:cs:*:*:cluster/叢集ID"
]
```

- 授予多個叢集許可權

```
"Resource": [
"acs:cs:*:*:cluster/叢集ID",
"acs:cs:*:*:cluster/叢集ID"
]
```

— 授予您所有叢集的許可權

```
"Resource": [
"*"
]
```

其中,叢集ID 需要替換為您要授權的真實的叢集 ID。

4. 編寫完畢後, 單擊建立授權策略。

# 表 1-1: Container ServiceRAM Action

Action	說明
CreateCluster	建立叢集
AttachInstances	向叢集中添加已有ECS執行個體
ScaleCluster	擴容叢集
GetClusters	查看叢集列表
GetClusterById	查看叢集詳情
ModifyClusterName	修改叢集名稱
DeleteCluster	刪除叢集
UpgradeClusterAgent	升級叢集Agent
GetClusterLogs	查看叢集的動作記錄
GetClusterEndpoint	查看叢集存取點地址
GetClusterCerts	下載叢集認證
RevokeClusterCerts	吊銷叢集認證
BindSLB	為叢集綁定Server Load Balancer執行個體
UnBindSLB	為叢集解除綁定Server Load Balancer執行個體
ReBindSecurityGroup	為叢集重新綁定安全性群組

Action	說明
CheckSecurityGroup	檢測叢集現有的安全性群組規則
FixSecurityGroup	修複叢集的安全性群組規則
ResetClusterNode	重設叢集中的節點
DeleteClusterNode	移除叢集中的節點
CreateAutoScale	建立節點Auto Scaling規則
UpdateAutoScale	更新節點Auto Scaling規則
DeleteAutoScale	刪除節點Auto Scaling規則
GetClusterProjects	查看叢集下的應用
CreateTriggerHook	為應用建立觸發器
GetTriggerHook	查看應用的觸發器列表
RevokeTriggerHook	刪除應用的觸發器
CreateClusterToken	建立 Token

# 1.2.3 子帳號Kubernetes應用許可權配置指導

本文旨在協助您瞭解如何通過Container Service控制台配置子帳號對應的Kubernetes RAM叢集許可權和在叢集內相應的Kubernetes RBAC應用許可權。

配置說明

- 子帳號授權頁面僅主帳號可見。您需要擁有一個阿里雲主帳號,並有一個或若干個子帳號。
- 由於阿里雲RAM的安全限制,當您通過Container Service控制台的授權配置涉及到子帳號RAM 授權的修改時,需要您按照頁面上給出的參考策略內容和操作說明,在RAM控制台進行目標子 帳號的手動授權。

操作步驟

📕 说明 :

因為子帳號授權頁面僅主帳號可見,您需要用主帳號登入Container Service控制台。

- 1. 登入Container Service#####。
- 2. 在Kubernetes菜單下,單擊左側導覽列中的叢集 > 子帳號授權,進入子帳號授權頁面。
- 3. 在子帳號列表中選擇需要授權的子帳號,單擊授權。

容器服务 - Kubernetes ▼	子账号授权 + 返回列表					
概范	选择子 <del>账号</del>		授权资	源	>	更新授权策略
	用户名	显示名	香注	创建时间		
<b>集群</b>	and the second sec			2018-05-16 11:32:11		授权
节点	ing a			2018-05-16 11:33:19		授权
存储卷	+			2018-07-12 16:31:03		授权
前名空间	Angeology (1997)			2018-07-18 10:21:39		授权 3
于眾与沒仅 2						

 進入授權資源頁面,您可以通過單擊表格左上方的加號添加叢集或命名空間層級的許可權配置, 並選擇相應的預置角色;也可以單擊配置行首的減號刪除目標角色。

子账号授权	€ 返回列表					
	选择子账号		授权资源		更新授权策略	
0	集群/命名空间			访问权限		
•	集群 k8s-test ▼ 命名空间	所有命名空间 🔻		◎ 管理员 ◎ 运	5维人员 💿 开发人员 🔍 受限用户	2
•	集群 k8s-test ▼ 命名空间	default 🔻		◉开发人员 🔍	受限用户	
权限说明						
	集群管理权限		应用管理权限			
管理员	集群 <mark>读写</mark> 权限,可删除、伸缩集群,添加	加节点	对所有命名空间下资源的 <mark>读写</mark> 权限,对集群节	点,存储卷,命名空间,配籍	额的读写权限	
运维人员	集群读写权限,可删除、伸缩集群,添加	加节点	对所有命名空间下资源的 <mark>读写</mark> 权限,对集群节	点,存储卷,命名空间,配输	顿的只读权限	
开发人员	集群只读权限		对所有命名空间或所选命名空间下资源的读写	权限		
受限用户	集群只读权限		对所有命名空间或所选命名空间下资源的只读	权限		
					上一步	下一步

叢集和命名空間的預置角色定義可查看下面的角色許可權說明:

## 表 1-2: 角色許可權說明

	叢集系統管理權限	應用系統管理權限
管理員	叢集讀寫權限,可刪除、伸縮 叢集,添加節點	對所有命名空間下資源的讀寫 權限,對叢集節點,儲存卷, 命名空間,配額的讀寫權限
營運人員	叢集讀寫權限,可刪除、伸縮 叢集,添加節點	對所有命名空間下資源的讀寫 權限,對叢集節點,儲存卷, 命名空間,配額的唯讀許可權
開發人員	叢集唯讀許可權	對所有命名空間或所選命名空 間下資源的讀寫權限
受限人員	叢集唯讀許可權	對所有命名空間或所選命名空 間下資源的唯讀許可權

 完成配置後,如果涉及目標子帳號RAM叢集許可權的變更,在更新授權策略頁面會展示配置項 對應的Kubernetes叢集RAM許可權參考配置,您可以根據頁面中的指導在RAM控制台完成子帳 號授權的更新。

#### 補充說明

為了不影響子帳號對當前存量可訪問kubernetes叢集的正常使用,Container Service控制台會暫時 相容舊的叢集存取權限控制,在一段時間對原有子帳號可訪問kubernetes叢集不進行RBAC的應用 許可權校正,如果您是子帳號使用者,請您根據叢集類型和相容方式,及時聯絡主帳號進行授權操 作。

對於由當前子帳號自身建立的存量叢集,可以通過在叢集詳情頁面單擊升級當前叢集授權資訊,完 成叢集應用許可權的自動升級。

在公告期結束後,仍舊沒有由主帳號授權或進行許可權管理升級的子帳號,子帳號將被禁止訪問叢 集對應的應用控制台。

表 1-3: 相容叢集說明

相容叢集類型	相容方式
存量子帳號建立叢集	提示許可權管理升級公告,提供一鍵升級連結, 子帳號可通過點擊升級連結完成應用授權
存量RAM授權訪問叢集	提示許可權管理升級公告,請及時聯絡主帳號完 成應用授權
建立RAM授權訪問叢集	提示許可權管理升級公告,請及時聯絡主帳號完 成應用授權

# 1.3 叢集管理

# 1.3.1 查看叢集概覽

阿里雲Container Service Kubernetes 叢集提供叢集概覽功能,提供應用狀態、組件狀態和資源監 控等功能,方便您快速瞭解叢集的健康狀態資訊。

操作步骤

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下, 單擊左側導覽列中的概覽,進入 Kubernetes 叢集概覽頁面。
- 3. 選擇所需的叢集和命名空間,您可查看應用狀態、組件狀態和資源監控圖表。

- 應用狀態:顯示當前啟動並執行部署、容器組和複本集的狀態示意圖,綠色表徵圖代表正常,黃色表徵圖代表異常。
- 節點狀態:顯示當前叢集的節點狀態。
- 組件狀態:Kubernetes 叢集的組件通常部署在 kube-system 命名空間下,包括 scheduler、controller-manager和 etcd 等核心組件。
- 資源監控:提供 CPU 和記憶體的監控圖表。CPU 統計單位為 Cores(核),可顯示小數點後3位,最小統計單位是 millcores,即一個核的 1/1000;記憶體的統計單位是 Gi,顯示小數點後3位。更多相關資訊,請參見 Meaning of CPU和 Meaning of memory。
- 事件:顯示叢集的事件資訊,例如警告和錯誤事件等。



# 1.3.2 建立Kubernetes 叢集

您可以通過Container Service管理主控台非常方便地快速建立 Kubernetes 叢集。

使用須知

建立叢集過程中, Container Service會進行如下操作:

- 建立 ECS, 組態管理節點到其他節點的 SSH 的公開金鑰登入, 通過 CloudInit 安裝配置 Kubernetes 叢集。
- 建立安全性群組,該安全性群組允許 VPC 入方向全部 ICMP 連接埠的訪問。

- 如果您不使用已有的 VPC 網路,會為您建立一個新的 VPC 及 VSwitch,同時為該 VSwitch 建 立 SNAT。
- 建立 VPC 路由規則。
- 建立 NAT Gateway及 EIP。
- 建立 RAM 子帳號和 AK,該子帳號擁有 ECS 的查詢、執行個體建立和刪除的許可權,添加和刪 除雲端硬碟的許可權,SLB 的全部許可權,Cloud Monitor的全部許可權,VPC 的全部許可權, Log Service的全部許可權,NAS 的全部許可權。Kubernetes 叢集會根據使用者部署的配置相應 的動態建立 SLB,雲端硬碟,VPC路由規則。
- 建立內網 SLB,暴露 6443 連接埠。
- 建立公網 SLB,暴露 6443、8443和 22 連接埠(如果您在建立叢集的時候選擇開放公網 SSH 登入,則會暴露 22 連接埠;如果您選擇不開放公網 SSH 訪問,則不會暴露 22 連接埠)。

前提條件

您需要開通Container Service、Resource Orchestration Service(ROS)服務和存取控制(RAM) 服務。

登入 Container Service#####, ROS ##### 和 RAM ##### 開通相應的服務。



Container Service Kubernetes 叢集部署依賴阿里雲Resource Orchestration Service 的應用部署能力,所以建立 Kubernetes 叢集前,您需要開通 ROS。

#### 使用限制

- 隨叢集一同建立的Server Load Balancer執行個體只支援隨用隨付的方式。
- Kubernetes 叢集僅支援專用網路 VPC。
- 每個帳號預設可以建立的雲資源有一定的配額,如果超過配額建立叢集會失敗。請在建立叢集前 確認您的配額。如果您需要提高您的配額,請提交工單申請。
  - 每個帳號預設最多可以建立5個叢集(所有地區下),每個叢集中最多可以添加40個節點。
     如果您需要建立更多的叢集或者節點,請提交工單申請。
  - 每個帳號預設最多可以建立 100 個安全性群組。
  - 每個帳號預設最多可以建立 60 個隨用隨付的Server Load Balancer執行個體。
  - 每個帳號預設最多可以建立 20 個EIP。
- ECS 執行個體使用限制:

- 僅支援 CentOS 作業系統。

- 支援建立隨用隨付和訂用帳戶的ECS執行個體。

#### 操作步驟

- 1. 登入 Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列的叢集,進入叢集列表頁面。
- 3. 單擊頁面右上方的建立 Kubernetes 叢集,進入建立 Kubernetes 叢集頁面。

容器服务	- Kubernetes 🔻	集群列表				您最多可以创建 5 个绩	東群 , 每个集群最多可以添加 40 个节点	刷新 创建 Kubernetes 集群	]
概题	-	常见问题: Ø 如何创建集群 Ø 扩容和缩容集群	洋 🔗 通过 kubectl 连接 Kuberne	tes 集群 🔗 通过命令管理应	7月			2	
▼ 集群	0	名称 🔻							
集群		集群名称/ID	集群类型 地域 (全部) 👻	网络类型	集群状态	创建时间	Kubernetes 版本	操作	
节点									

預設進入kubernetes叢集配置頁面。

创建 Kuberne	tes 集群 🔒 返回集群	列表		
Kubernetes	多可用区 Kubernetes	Serverless Kubernetes ( 公测 )	𝔅 申请 Kubernetes 托管版公测	
* 集群名称	k8s-cluster			
	名称为1-63个字符,可	〕包含数字、汉字、英文字符,或"-"		

4. 填寫叢集的名稱。

叢集名稱應包含1-63個字元,可包含數字、漢字、英文字元或連字號(-)。

5. 選擇叢集所在的地區和可用性區域。

地域:	华北 1	华北 2	华东 1	华东 2	华南 1	亚太东北 1 (东京) 美国西部 1 (硅谷)
			亚太东南1(新加	美国东部1(弗吉	欧洲中部1(法兰	
	华北 3	香港	坡)	尼亚)	克福)	
可用区:	华东 1 可用区 F	-				

6. 設定叢集的網路。Kubernetes 叢集僅支援專用網路。

專用網路:您可以選擇自動建立(建立 Kubernetes 叢集時,同步建立一個 VPC)或者使用已 有(使用一個已有的 VPC)。選擇使用已有後,您可以在已有 VPC 列表中選擇所需的 VPC 和 交換器。

- 選擇自動建立,建立叢集時,系統會自動為您的 VPC 建立一個 NAT Gateway。
- 選擇使用已有,如果您使用的 VPC 中當前已有 NAT Gateway, Container Service會使用已 有的 NAT Gateway;如果 VPC 中沒有 NAT Gateway,系統會預設自動為您建立一個 NAT Gateway。如果您不希望系統自動建立 NAT Gateway,可以取消勾選頁面下方的為專用網路 配置 SNAT。

- 7. 設定節點類型, Container Service支援隨用隨付和訂用帳戶兩種節點類型。
- 8. 設定 Master 節點的配置資訊。

您需要選擇 Master 節點的系列和規格。

📕 说明:

- 目前僅支援 CentOS 作業系統。
- 目前僅支援建立3個 Master 節點。
- 支援為Master節點掛載系統硬碟,支援SSD雲端硬碟和高效雲端硬碟。

MASTER 配置							
实例规格	4核8G(ecs.sn1ne.xlarge)		•	T	数量	3台	
系统盘	SSD云盘 ▼	40	GiB 🌲				

9. 設定 Worker 節點的配置資訊。您可選擇新增執行個體或添加已有執行個體。

道 说明:
• 目前僅支援 CentOS 作業系統。
• 每個叢集最多可包含 37 個 Worker 節點。如果您需要建立更多的節點,請提交工單申請。
• 支援為Worker節點掛載一個資料盤,支援SSD雲端硬碟、高效雲端硬碟和普通雲端硬碟

a. 若您選擇新增執行個體,則需要選擇 Worker 節點的系列和規格,以及需要建立的 Worker 節點的數量(本樣本建立1個 Worker 節點)。

WORKER 配置									
实例规格	4核8G(ecs.sn1ne.xlarge)				• 7	数量	1	台	*
系统盘	SSD云盘	•	40	GiB	*				
✔ 挂载数据盘	SSD云盘	•	40	GiB	*				

# b. 若您選擇添加已有執行個體,則需要預先在此地區下建立 ECS 雲端服務器。

10.配置登入方式。

• 設定密鑰。

您需要在建立叢集的時候選擇金鑰組登入方式,單擊建立金鑰組,跳轉到ECS雲端服務器控 制台,建立金鑰組,參見## SSH ###。金鑰組建立完畢後,設定該金鑰組作為登入叢集的憑 據。

登录方式	设置密钥 设置密码	
密钥对		- <b>C</b>
	您可以访问 ECS 控制台 <mark>新建密钥对</mark>	
	⑧ 请选择密钥对	

• 設定密碼。

--- 登入密碼:設定節點的登入密碼。

- 確認密碼:確認設定的節點登入密碼。

11.設定Pod網路 CIDR 和Service CIDR。



12.設定是否為專用網路配置 SNAT Gateway。



若您选择的 VPC 不具备公网访问能力,将使用 NAT 网关、EIP 为该 VPC 配置 SNAT,期间可能创建 NAT 网关、EIP 等资源

13.設定是否開放公網 SSH 登入。

- 選擇開放公網 SSH 登入,您可以 SSH 訪問叢集。
- 選擇不開放公網 SSH 登入,將無法通過 SSH 訪問叢集,也無法通過 kubectl 串連 叢集。如 果您需要通過 SSH 訪問叢集執行個體,可以手動為 ECS 執行個體綁定 EIP,並配置安全性 群組規則,開放 SSH (22)連接埠,具體操作參見SSH##Kubernetes##。

SSH登录	✔ 开放公网SSH登录
	选择不开放时,如需手动开启 SSH 访问,请参考 SSH 访问 Kubernetes 集群

14. 設定是否啟用Cloud Monitor外掛程式。

您可以選擇在 ECS 節點上安裝Cloud Monitor外掛程式,從而在Cloud Monitor控制台查看所建立 ECS 執行個體的監控資訊。

云监控插件: ▼ 在ECS节点上安装云监控插件 在节点上安装云监控插件,可以在云监控控制台查看所创建ECS实例的监控信息

15. 設定是否啟用Log Service,您可使用已有Project或建立一個Project。

勾選使用**SLS**,會在叢集中自動設定Log Service外掛程式。建立應用時,您可通過簡單配置, 快速使用Log Service,詳情參見*Kubernetes#Log Service#####*。

日志服务	✔ 使用 SLS	
	使用已有 Project	创建新 Project
	将自动创建名称为 k8s-l	og-{ClusterID} 的 Proj

16.是否啟用進階選項。

- a. 設定啟用的網路外掛程式,支援Flannel和Terway網路外掛程式。
  - Flannel:簡單穩定的社區的Flannel cni外掛程式。
  - Terway:阿里雲Container Service自研的網路外掛程式,支援將阿里雲的彈性網卡分配給容器,支援Kubernetes的NetworkPolicy來定義容器間的存取原則,支援對單個容器做頻寬的限流,目前處於公測階段。
- b. 設定節點 Pod 數量,是指單個節點可運行 Pod 數量的上限,建議保持預設值。

c. 設定是否選擇自訂鏡像。或不選擇自訂鏡像,則 ECS 執行個體會安裝預設的 CentOS 版本。

目前您只能選擇基於 CentOS 的鏡像來快速部署您需要的環境,如基於 CentOS 7.4 的 LAMP 部署測試的鏡像。

**d.** 設定是否使用自訂叢集**CA**。如果勾選自訂叢集 CA,可以將 CA 憑證添加到 kubernetes 叢集 中,加強服務端和用戶端之間資訊互動的安全性。

集群CA: 自定义集群CA

17.單擊建立叢集,啟動部署。

说明:

-個包含多節點的 Kubernetes 叢集的建立時間一般需要十幾分鐘。

#### 查看叢集部署結果

叢集建立成功後,您可以在Container Service管理主控台的 Kubernetes 叢集列表頁面查看所建立 的叢集。

容器服务 - Kubernetes ▼	集群列表				绝	强多可以创建 5 个集群,每	个集群最多可以添加 40 个节点	刷新	创建 Kubernetes 集群
概览	常见问题: ②如何创建集群 ③ 扩容和缩容	集群 🔗 通过 kubectl 连接	そ Kubernetes 集群	Ø 通过命令管理应用					
▼ 集群	名称 ▼								
集群	集群名称/ID	集群类型	地域 (全部) 👻	网络类型	集群状态	创建时间	Kubernetes 版本		操作
节点 存储卷	k8s-cluster	Kubernetes	华东1	虚拟专有网络 vpc-bp1lkyevdjj	●运行中	2018-09-26 17:41:26	1.11.2	管理	<ul> <li>         查看日志   控制台         集群伸缩   更多 ▼     </li> </ul>

# 您可以單擊右側的查看日誌查看叢集的日誌資訊,您可單擊資源棧事件查看更詳細的資訊。

│ 集群日志信息: k8s-cluste	r < 近回無数列表	刷新
资源部署详细日志请参考:	急速线事件	
时间	在思	
2018-07-09 17:19:01	c07ea7a5639924917af756e7d3l95e0d5   Start to delete node	
2018-07-09 17:19:01	c07ea7a5639924917af756e7d3I95e0d5   Failed to delete node (i-bp1dmw3cynna4bef3rfa) Api server error.	
2018-07-09 17:18:59	c07ea7a5639924917af756e7d3l95e0d5   Start to delete node	
2018-07-09 17:18:59	c07ea7a5639924917af756e7d3I95e0d5   Failed to delete node (i-bp1dmw3cynna4bef3rfb) Api server error.	
2018-07-09 17:18:40	c07ea7a5639924917af756e7d3l95e0d5   Start to delete node	
2018-07-09 17:18:40	c07ea7a5639924917af756e7d3I95e0d5   Failed to delete node (i-bp1dmw3cymna4bsf3rfa) Api server error.	
2018-07-09 17:18:33	c07ea7a5639924917af756e7d395e0d5   Start to delete node	
2018-07-09 17:18:33	c07ea7a5639924917af756e7d3I95e0d5   Failed to delete node (i-bp1dmw3cynna4bef3rfb) Api server error.	

# 您可以單擊右側的管理,查看叢集的基本資料和串連資訊。

基本信息					
集群ID:	虚拟专有网络	●运行中	地域: 华东1		
连接信息					
API Server 公网连接端点	https://1 6443				
API Server 内网连接端点	https://1 6443				
Master 节点 SSH 连接地址	点 SSH 连接地址				
服务访问域名	).0	n-hangzhou.alicontainer.com			
集群资源					
资源编排 ROS	k8s-for-cs-c	8879			
公网 SLB	lb-bp				
虚拟专有网络 VPC	vpc-i				
NAT 网关	NAT 网关 ngw-				
週过 kubectl 连接 Kubernetes 集群					
1. 从 Kubernetes 版本页面 下载最新的 kubectl 客户端。					
2. 安装和设置 kubectl 客户端。有关详细信息,参见 安装和设置 kubectl					
3. 配置集群凭据:					
KubeConfig SSH					

# 其中:

• API Server 公網串連端點: Kubernetes 的 API server 對公網提供服務的地址和連接埠,可以通 過此服務在使用者終端使用 kubectl 等工具管理叢集。

- API Server 內網串連端點: Kubernetes 的 API server 對叢集內部提供服務的地址和連接埠。此 IP 為負載平衡的地址,後端有3台 Master 提供服務。
- Master 節點 SSH 串連地址:可以直接通過 SSH 登入到 Master 節點,以便對叢集進行日常維護。
- 服務訪問網域名稱:為叢集中的服務提供測試用的訪問網域名稱。服務訪問網域名稱尾碼是<</li>
   cluster\_id>.<region\_id>.alicontainer.com。

例如,您可以通過 SSH 登入到 Master 節點,執行kubect1 get node查看叢集的節點資訊。

login as: root				
root@1 3's password:				
Welcome to Alibaba Cloud Elastic Com	pute Servi	ce !		
[root@iZbp1d7yvpa3j183u0ur11Z ~]# ku	bectl get	node		
NAME	STATUS	ROLES	AGE	VERSION
cn-hangzhou.i	Ready	<none></none>	17m	v1.8.4
cn-hangzhou.i	Ready	master	19m	v1.8.4
cn-hangzhou.i	Ready	master	24m	v1.8.4
cn-hangzhou.i	Ready	master	22m	v1.8.4
[root@iZbp1d7yvpa3j183u0url1Z ~]#				

可以發現,一共有 4 個節點,包括 3 個 Master 節點和我們在參數設定步驟填寫的 1 個 Worker 節點。

# 1.3.3 VPC下 Kubernetes 的網路位址區段規劃

在阿里雲上建立 Kubernetes 叢集時,通常情況下,可以選擇自動建立專用網路,使用預設的網路 地址即可。在某些複雜的情境下,需要您自主規劃 ECS 地址、Kubernetes Pod 地址和 Service 地 址。本文將介紹阿里雲 VPC 環境下 Kubernetes 裡各種地址的作用,以及位址區段該如何規劃。

#### Kubernetes 網段基本概念

首先來看幾個和 IP 位址有關的概念。

VPC 網段

您在建立 VPC 選擇的位址區段。只能從 10.0.0.0/8、172.16.0.0/12、192.168.0.0/16 三者當中選擇 一個。

交換器網段

在 VPC 裡建立交換器時指定的網段,必須是當前 VPC 網段的子集(可以跟 VPC 網段地址一樣, 但不能超過)。交換器下面的 ECS 所分配到的地址,就是從這個交換器位址區段內擷取的。一個 VPC 下,可以建立多個交換器,但交換器網段不能重疊。

VPC 網段結構如下圖。



# Pod 位址區段

Pod 是 Kubernetes 內的概念,每個 Pod 具有一個 IP 位址。在阿里雲Container Service上建立 Kubernetes 叢集時,可以指定Pod 的位址區段,不能和 VPC 網段重疊。比如 VPC 網段用的是 172.16.0.0/12, Kubernetes 的 Pod 網段就不能使用 172.16.0.0/16,不能使用 172.17.0.0/16..., 這些地址都涵蓋在 172.16.0.0/12 裡了。

# Service 位址區段

Service 也是 Kubernetes 內的概念,每個 Service 有自己的地址。同樣,Service 位址區段也不 能和 VPC 位址區段重合,而且 Service 位址區段也不能和 Pod 位址區段重合。Service 地址只在 Kubernetes 叢集內使用,不能在叢集外使用。

Kubernetes 網段和 VPC 網段關係如下圖。



# 如何選擇位址區段

## 單 VPC+單 Kubernetes 叢集情境

這是最簡單的情形。VPC 地址在建立 VPC 的時候就已經確定,建立 Kubernetes 叢集時,選擇和 當前 VPC 不一樣的位址區段就可以了。

## 單 VPC+多 Kubernetes 叢集情境

一個 VPC 下建立多個 Kubernetes 叢集。在預設的網路模式下(Flannel),Pod 的報文需要通過 VPC 路由轉寄,Container Service會自動在 VPC 路由上配置到每個 Pod 位址區段的路由表。所有 Kubernetes 叢集的 Pod 位址區段不能重疊,但 Service 位址區段可以重疊。

VPC 地址還是建立 VPC 的時候確定的,建立 Kubernetes 的時候,為每個 Kubernetes 叢集選擇一個不重疊的位址區段,不僅不能和 VPC 地址重疊,也不和其他 Kubernetes Pod 段重疊。

需要注意的是,這種情況下 Kubernetes 叢集部分互連,一個叢集的 Pod 可以直接存取另外一個叢 集的 Pod 和 ECS,但不能訪問另外一個叢集的 Service。

#### VPC 互聯情境

兩個 VPC 網路互聯的情況下,可以通過路由表配置哪些報文要發送到對端 VPC 裡。以下面的情境 為例,VPC 1 使用位址區段 192.168.0.0/16,VPC 2 使用位址區段 172.16.0.0/12,我們可以通過 路由表,指定在 VPC 1 裡把 目的地址為 172.16.0.0/12 的報文都發送到 VPC 2。



在這種情況下,VPC 1 裡建立的 Kubernetes 叢集,首先不能和 VPC 1 的位址區段重疊,同時其 位址區段也不能和 VPC 2 的位址區段重疊。在 VPC 2 上建立 Kubernetes 叢集也類似。這個例子 中,Kubernetes 叢集 Pod 位址區段可以選擇 10.0.0.0/8 下的某個子段。

道 说明:

這裡要特別關注"路由到 VPC 2" 的位址區段,可以把這部分地址理解成已經佔用的地

址,Kubernetes叢集不能和已經佔用的地址重疊。

如果 VPC 2 裡要訪問 VPC 1 的 Kubernetes Pod,則需要在 VPC 2 裡配置到 VPC1 Kubernetes叢 集pod地址的路由。

VPC 網路到 IDC 的情境

和 VPC 互聯情境類似,同樣存在 VPC 裡部分位址區段路由到 IDC ,Kubernetes 叢集的 Pod 地址 就不能和這部分地址重疊。IDC 裡如果需要訪問 Kubernetes 裡的 Pod 地址,同樣需要在 IDC 端配 置到專線 VBR 的路由表。

# 1.3.4 通過 kubectl 串連 Kubernetes 叢集

如果您需要從用戶端電腦串連到 Kubernetes 叢集,請使用 Kubernetes 命令列用戶端 kubectl。

操作步骤

- 1. 從Kubernetes #### 下載最新的 kubectl 用戶端。
- 2. 安裝和設定 kubectl 用戶端。

有關詳細資料,參見 ##### kubectl。

3. 配置叢集憑據。

您可以使用scp命令安全地將主節點的配置從 Kubernetes 叢集主 VM 中的 /etc/kubernetes

/kube.conf 複製到本機電腦的 \$HOME/.kube/config(kubectl 預期憑據所在的位置)。

• 如果您在建立叢集時選擇密碼方式登入,請用以下方式拷貝 kubectl 設定檔。

```
mkdir $HOME/.kube
scp root@<master-public-ip>:/etc/kubernetes/kube.conf $HOME/.kube/
config
```

如果您在建立叢集時選擇金鑰組方式登入,請用以下方式拷貝 kubectl 設定檔。

```
mkdir $HOME/.kube
scp -i [.pem 私密金鑰檔案在本地機器上的儲存路徑] root@:/etc/kubernetes/
kube.conf $HOME/.kube/config
```

您可以在叢集資訊頁面查看叢集的 master-public-ip。

- a) 登入Container Service#####。
- b) 單擊Kubernetes進入 Kubernetes 叢集列表頁面。
- c) 選擇所需的叢集並單擊右側的管理

您可以在串連資訊處查看叢集的串連地址。

集群 : k8s-test					
基本信息					
集群ID: cde6	虚拟专有网络	●运行中	地域: 华东1	节点个数 4	集群伸缩
连接信息					
Master 节点 SSH 连接地址	101.37				
API Server 公网连接端点	https:/				
API Server 内网连接端点	https:/				

# 1.3.5 SSH訪問Kubernetes叢集

如果您在建立叢集時,選擇不開放公網 SSH 訪問,您將無法 SSH 訪問 Kubernetes 叢集,而且也 無法通過 kubectl 串連 Kubernetes 叢集。如果建立叢集後,您需要 SSH 訪問您的叢集,可以手動 為 ECS 執行個體綁定 EIP,並配置安全性群組規則,開放 SSH(22)連接埠。

操作步驟

- 1. 登入 Container Service#####。
- 2. 在 Kubernetes 菜單下, 單擊左側導覽列中的叢集, 進入 Kubernetes 叢集列表頁面。
- 3. 選擇所需的叢集並單擊右側的管理。
- 在叢集資源中,找到您的公網Server Load Balancer執行個體,並單擊執行個體 ID,頁面跳轉至 您的公網Server Load Balancer執行個體的詳細資料頁面。

集群:k8s-cluster					
基本信息					
集群ID:	虚拟专有网络	●运行中	地域: 华东1		
连接信息					
API Server 公网连接端点	Hard 1970 (1990)				
API Server 内网连接端点	Recolution (Comparison)				
Master 节点 SSH 连接地址	KIND OF THE PARTY				
服务访问域名	1-3-3-3-3-10-10-3-4-3-4-3-4-3-4-4-4-4-4-4-4-4-4-4-4-4-				
集群资源					
资源编排 ROS	A A LOUIS ANY MUSICIPALITY				
公网 SLB	1 L Ballon A B				
虚拟专有网络 VPC	a subscription and a subscription of the subsc				
NAT 网关	and the second second				

5. 單擊左側導覽列中的執行個體 > 執行個體管理, 單擊添加監聽。

- 6. 添加 SSH 監聽規則。
  - a. 前端協議(連接埠)選擇 TCP:22。
  - **b.** 後端協議(連接埠)選擇 TCP : 22。
  - c. 選擇 使用伺服器組 並選擇虛擬伺服器組。
  - d. 伺服器組ID 選擇sshVirtualGroup。
  - e. 單擊下一步 並單擊確認,建立監聽。

添加监听		$\times$
1.基本配置	2.健康检查配置 3.配置成功	
前端协议 [端口]:★	TCP     ▼     22       端口輸入范围为1-65535。       四层监听请选择TCP、UDP; 七层监听请选择HTTP、HTTPS; 查看详情	
后端协议 [端口] : 🗚	TCP : 22 端口输入范围为1-65535。	
带宽峰值:	不限制 配置 使用流量计费方式的实例默认不限制带宽峰值;峰值输入范围1-5000	
调度算法:	加权轮询	
使用服务器组: 🛛	● ● 什么是服务器组?	
服务器组类型:	◉ 虛拟服务器组 ❷ 🛛 🔍 主备服务器组 ❷	
服务器组ID:	sshVirtualGroup 🔻	
创建完毕自动启动监 听:	● 已开启	
→ 展开高级配置		
	下一步	取消

7. 監聽建立完成後,您就可以使用該負載平衡的服務地址 SSH 訪問您的叢集了。

K8sMasterSlbIntern * 返回负載均衡列表	☞ 使用限制和注意事项
基本信目	
()	
负载均衡名称: K8sMa	地域: 华东 1
地址类型: 公网	可用区: 华东1可用区 B(主)/华东1可用区 D(街)
网络类型: 经典网络	
付费信息	演奏明细 <mark>释放设置</mark> 个
付费方式: 按使用流量	创建时间: 2018-01-04 19:47:11
服务地址: 101	自动释放时间: 无

# 1.3.6 SSH金鑰組訪問Kubernetes叢集

阿里雲Container Service支援通過 SSH 金鑰對的方式登入叢集,保障遠程SSH訪問的安全性。

# 操作步骤

- 1. 登入 Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的叢集,進入 Kubernetes 叢集列表頁面。
- 3. 單擊右上方的建立 Kubernetes 叢集。

容器服务		集群列表	您最多可以创建 5	5 个集群,每个集群最多可以添加 4	0 个节点 刷新	申请使用Serverless Kubernetes集群	创建 Kubernetes 集群 🚽
Kubernetes	Swarm						
概院		常见问题: ⑧ 如何创建集群 ⑧ 扩容和	」缩容集群 📀 通过 kubectl 连接 Kuberne	etes 集群 🛛 ③ 通过命令管理应用			
▼ 集群		名称 ▼					
集群		集群名称/ID	集群美型 地域 (全部) マ 网络	格美型 集群状态	创建时间	Kubernetes 版本	操作
节点							

- 對叢集登入方式進行配置,設定通過密鑰的登入方式。對叢集其他的參數進行配置,參見##
   Kubernetes##,最後單擊建立。
  - 1. 若您已在ECS雲端服務器控制台建立了密鑰,只需在金鑰組下拉框中進行選擇。
  - 2. 若您沒有密鑰,單擊建立金鑰組,前往ECS雲端服務器控制台建立金鑰組,參見## SSH ##
     #。

登录方式	设置密钥	设置密码		
密钥对	test11		- S	
您可以访问 ECS 控制台 <mark>新建密钥对</mark>				

5. 叢集建立成功後,在叢集列表中找到該叢集,單擊叢集右側的管理,在串連資訊中查看Master 節點SSH串連地址。

<	集群:test-k8s
基本信息	基本信息
	集群ID: c5
	连接信息
	API Server 公网连接端点
	API Server 内网连接端点
	Master 节点 SSH 连接地址
	服务访问域名

- 下載 . pem私密金鑰檔案,按照本地作業系統環境,如 Windows 和 linux 作業系統,進行相關配置,參見##SSH#####Linux####。以 linux 環境為例。
  - a) 找到您所下載的.pem私密金鑰檔案在本地機上的儲存路徑,如 /root/xxx.pem
  - b) 運行命令修改私密金鑰檔案的屬性: chmod 400 [.pem私密金鑰檔案在本地機上的儲存路 徑],如 chmod 400 /root/xxx.pem。
  - c) 運行命令串連至叢集: `ssh -i [.pem私密金鑰檔案在本地機上的儲存路徑] root@[ master-public-ip],其中 master-public-ip 即是 Master 節點 SSH 串連地址。如 ssh i /root/xxx.pem root@10.10.10.100。

# 1.3.7 升級叢集

您可以通過Container Service管理主控台升級您叢集的 Kubernetes 版本。

您可以在 Kubernetes 叢集列表頁面查看您的叢集的 Kubernetes 版本。

容器服务		集群列表				您最多可以创建	書 5 个集群,每个集群最多可以添加 20 个节点	周期	创建 Kubernetes 集群	•
Kubernetes	Swarm									
▼ 集群		常见问题: Ø 如何创建集群 Ø 扩容和缩容集群	⌀ 通过 kubec	tl 连接 Kubernetes 集群	⌀ 通过命令管理应用					
集群		名称 ▼								
节点		集群名称/ID	地域	网络类型	集群状态	创建时间	Kubernetes 版本		1	築作
存储		k8s-cluster	华东1	虚拟专有网络 vpc-bp1gnu8br4a	● 运行中	2018-04-23 11:41:46	1.9.3	管理	<ul> <li>1 査看日志   控制</li> <li>集群伸縮   更</li> </ul>	台 多▼

注意事項

- 叢集升級需要機器可以公網訪問,以便下載升級所需的軟體包。
- 叢集升級 Kubernetes 過程中,可能會有升級失敗的情況,為了您的資料安全,強烈建議您先打 快照然後再升級。有關 ECS 打快照的操作參見####。
- 叢集升級 Kubernetes 過程中, 叢集上部署中的服務會中斷, 同時無法進行叢集和應用的操作, 請您在升級之前安排好相關事宜。升級時間大約 5-30 分鐘, 升級完成後叢集會變成運行中狀 態。
- 準備工作

請在叢集升級前檢查叢集的健康情況,並且確保叢集健康。

登入 Master 節點,參見SSH##Kubernetes##和## kubectl ## Kubernetes ##。

1. 執行kubectl get cs命令,確保所有模組都處於健康狀態。

NAME	STATUS	MESSAGE		ERROR
scheduler	Healthy	ok		
controller-manager	Healthy	ok		
etcd-0	Healthy	{"health":	"true"}	
etcd-1	Healthy	{"health":	"true"}	
etcd-2	Healthy	{"health":	"true"}	

2. 執行 kubectl get nodes 命令,確保所有節點都處於 Ready 狀態。

kubectl get nodes					
NAME		STATUS	S R	ROLES	AGE
VERSION					
cn-shanghai.i-xxxxxx	Ready	master	38d	7	v1.9.3
cn-shanghai.i-xxxxxx	Ready	<none></none>	38d	7	v1.9.3
cn-shanghai.i-xxxxxx	Ready	<none></none>	38d	7	v1.9.3
cn-shanghai.i-xxxxxx	Ready	<none></none>	38d	7	v1.9.3
cn-shanghai.i-xxxxxx	Ready	master	38d	7	v1.9.3
cn-shanghai.i-xxxxxx	Ready	master	38d	7	v1.9.3

如果節點不正常可以自行修複,也可以通過提交工單,請阿里雲工程師協助修複。

#### 操作步驟

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的叢集,進入 Kubernetes 叢集列表頁面。
- 3. 選擇所需的叢集,並單擊更多 > 叢集升級。

容器服务	集群列表			您最多可以创建!	5 个集群,每个集群最多可以添加 20 个节点	创建 Kubernetes 集群 🗸
Kubernetes Swarm	常见问题: & 如何创建集群 Ø 扩容和谐容集群 名称 マ	♂ 通过 kubecti 连接 Kubernetes 集群	Ø 通过命令管理应用			
节点	集群名称/ID	地域 网络类型	集群状态	创建时间	Kubernetes 版本	操作
存储	k8stest 🖌	华东1 虚拟专有网络 vpc-bp1gnu8br4a	●运行中	2018-04-23 11:41:46	100	管理 查看日志 控制 在 集群仲缩 更多
▼ 应用						<b>删除</b> 添加已有节点
部署						3 果群升级
<ul> <li>★ 应用</li> <li>部署</li> <li>服务</li> </ul>						3         近日有市点           工作時分報         1000000000000000000000000000000000000

4. 在彈出的對話方塊中,單擊升級。

系統開始升級 Kubernetes 的版本。

升級完成後,您可以在 Kubernetes 叢集列表頁面查看叢集 Kubernetes 的版本,確認升級成功。

# 1.3.8 擴容和縮容叢集

通過Container Service管理主控台,您可以根據實際業務需要對 Kubernetes 叢集的 Worker 節點進 行擴容和縮容。

#### 背景信息

- 目前不支援叢集中 Master 節點的擴容和縮容。
- 叢集縮容只能縮減叢集建立和擴容時增加的 Worker 節點,通過######功能添加到叢集中的 Worker 節點不能被縮減。
- 縮減規則是按建立時間進行的,最新擴容出來的節點會被先回收。
- 必須有大於1個非手動添加的節點,才能進行縮容。

## 操作步骤

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下, 單擊左側導覽列中的叢集, 進入 Kubernetes 叢集列表頁面。
- 3. 選擇所需的叢集並單擊右側的叢集伸縮。

容器服务	集群列表		您最多可以创建 5	个集群,每个集群最多可以添加 20 个节点	刷新 创建 Kubernetes 集群 🚽
Kubernetes Swarm	<b>常见问题: &amp;</b> 如何创建集群 <b>&amp;</b> 扩容和缩容集	詳 🔗 通过 kubectl 连接 Kubernete	s 集群 🛛 ⑧ 通过命令管理应用		
集群 集群	名称 ▼				
节点	集群名称/ID	地域 网络类型	集群状态 创建时间	Kubernetes 版本	操作
存储	k8s-cluster	华东1 虚拟专有网络 vpc-bp13cwaie5z	●运行中 2018-04-20 11:0	02:31 1.9.3	管理   查看日志   控制台 2 集群件缩   更多▼

4. 選擇擴容 或者縮容,設定 Worker 節點的數量。

本樣本進行叢集擴容,將叢集的 Worker 節點數由 1 個擴容到 4 個。

集群名称	k8s-test
地域	华东1 可用区G
已有Worker数	1
伸缩	<b>扩容</b> 缩容
实例规格	2核16G(ecs.se1ne.large)
伸缩数量	3 台 🜩
伸缩后 Worker 数	4
* 登寻恋码	•••••••
32.8Cd 15	
RDS白名单	请选择你想要添加白名单的RDS实例

5. 填寫節點的登入密碼。

送 说明:

由於升級過程依賴登入到 ECS 來拷貝配置資訊,所以叢集伸縮時填寫的密碼必須和建立叢集時 填寫的密碼一致。

6. 單擊提交。

后续操作

伸縮完成後,單擊左側導覽列中的叢集 > 節點,查看節點列表頁面,您可以看到 Worker 節點的數 量從 1 變成了 4。

# 1.3.9 自動調整叢集

根據叢集負載的工作情況,自動調整叢集。

前提條件

您已經成功建立一個 Kubernetes 叢集,參見##Kubernetes##。

背景資訊

自動調整叢集與傳統基於資源閾值的#######不同,根據配置好的伸縮條件,當叢集工作負載的工 作情況達到配置條件時,叢集可進行自動調整。

#### 操作步驟

執行自動調整

- 1. 登入 Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的叢集,進入Kubernetes叢集列表頁面。
- 3. 選擇所需的叢集並單擊操作列的更多 > 自動調整。

·莱田·白杨y1D	集群类型	地域 (全部) 🔻	网络类型	集群状态	创建时间	Kubernetes 版本			操作
k8s-cluster ∠ c51b1b2bd791f4496ae28fd3a6b1ac32f	Kubernetes	华东1	虚拟专有网络 vpc-bp1kd7yn4qn	●运行中	2018-08-24 09:57:11	1.10.4	管理	查看日志   集群伸缩	控制台   更多▼
								删除 添加已有节 集群升级	抗
								自动伸缩 系统组件升 升级监控服	 †级 段务

#### 授權

- 開通ESS服務
  - 1. 單擊彈出對話方塊中的第一個連結,進入Auto Scaling服務 ESS頁面。

提示	×
自动伸缩依赖弹性伸缩(ESS)服务,启用自动伸缩前,您需要:	
1 开通该服务,并完成默认角色授权:弹性伸缩(ESS)	
2 跳转到访问控制(RAM)为当前集群添加 ESS 授权策略: 查看详细操作步骤 KubernetesWorkerRole	
请确认完成以上步骤,否则自动伸缩将无法正常启用	
	已完成

2. 單擊開通ESS服務,進入雲產品開通頁。

Sa	弹性伸缩服务 ESS <sup>您尚未开通弹性伸缩服务ESS</sup>
	开通ESS服务
概览	
• 弹性( 中体	伸缩(Auto Scaling),是根据用户的业务需求和策略,经济地自动调整弹性计算资源的管理服务。弹性伸缩不仅适合业务显不断波动的应用程序,同时也适合业务量稳定的应用程序
· 免费(	使用,所需资源按照相关价格计费
主要功能	
<ul> <li>弹性</li> <li>弹性</li> <li>弹性</li> <li>弹性</li> </ul>	f 展:自动调整弹性计算资源,在满足业务需求高峰增长时无罐地增加ECS实例 灾缩:自动调整弹性计算资源,在业务需求下降时自动减少ECS实例以节约成本 自愈:自动普换不健康的ECS实例使业务始终保持正常的负载,为业务保暖护航
参考文档	
• 产品i	
<ul> <li>快速/</li> <li>用户</li> </ul>	
<ul> <li>         ・         ・         ・</li></ul>	1首巻>

3. 選中我已閱讀並同意複選框,單擊立即開通。



4. 開通成功後,在開通完成頁簽,單擊管理主控台,進入Auto Scaling服務 ESS頁面。
| 确认订单    |                                |
|---------|--------------------------------|
|         |                                |
| $\odot$ | 恭喜,开通成功!<br>您订购的弹性伸缩服务正在努力开通中, |
|         | 管理控制台                          |

5. 單擊前往授權,進入雲資源訪問授權頁面,配置對雲資源的存取權限。

SV	弹性伸缩服务 ESS 弹性伸缩服务 ESS 并未得到您完整的 Open API 授权	
	前往授权	

6. 單擊同意授權。

一 云资源坊问授权	
温馨提示:如齋修改角色权限,请前往RAM控制台角色管理中设置,需要注意的是,错误的配置可能导致ESS无法获取到必要的权限。	×
ESS请求获取访问您云资源的权限 下方是系统创建的可供ESS使用的角色,授权后,ESS拥有对您云资源相应的访问权限。	
AliyunESSDefaultRole         描述:       ESS就认使用此角色未访问您在其他云产品中的资源         权限描述:	v.
同意结论	

預期結果

頁面自動跳轉至Auto Scaling控制台,說明授權成功。關閉頁面,繼續配置授權角色。

• 授權角色

1. 單擊彈出對話方塊中的第二個連結,進入角色授權策略頁面。

☐ 说明:	
此處需要以主帳號登入控制台。	
提示	$\times$

跳转到访问控制(RAM)为当前集群添加 FSS 授权策略 · 杏麦详细操作步骤
KubernetesWorkerRole
认完成以上步骤,否则自动伸缩将无法正常启用

2. 選擇目標授權策略名稱並單擊操作列查看許可權,進入授權策略詳情頁面。

<	KubernetesWorkerRole-21646e8a-2d64-4e5d-8332-4b200c72545f			编辑授权策略
角色详情				
da, da Lad IC Annair	授权策略名称	备注	美型	操作
用出这次完成	k8sWorkerRolePolicy-21646e8a-2d64-4e5d-8332-4b200c72545f		自定义	查看权限 解除授权

3. 單擊頁面右上方修改授權策略。

<	k8sWorkerRolePolicy-21646e8a-
授权策略详情	策略详情
版本管理	名称 k8sWorkerRolePolicy-21646e8
引用记录	备注
	1       {         2       "Version": "1",         3       "Statement": [         4       {         5       "Action": [         6       "ecs:AttachDisk         7       "ecs:DetachDisk         8       "ecs:DescribeDi         9       "ecs:CreateDisk         10       "ecs:CreateSnap         11       "ecs:DeleteDisk         12       "ecs:DescribeNetw         13       "ecs:DescribeNetw         14       "ecs:DetachNetw         15       "ecs:DeleteNetw         16       "ecs:DeleteNetw

4. 在策略內容的Action欄位,補充以下策略:

```
"ess:Describe*",
"ess:CreateScalingRule",
"ess:ModifyScalingGroup",
"ess:RemoveInstances",
"ess:ExecuteScalingRule",
"ess:ModifyScalingRule",
"ess:DeleteScalingRule",
"ecs:DescribeInstanceTypes",
```

"ess:DetachInstances"

说明: 需要在Action欄位的最後一行補充",",再添加以上內容。

5. 單擊修改策略。

## 配置自動調整

## 1. 在Auto Scaling配置頁面,填寫以下資訊,配置自動縮容:

配置	說明
叢集	目的地組群名稱。
縮容閾值	叢集中負載申請的資源量與叢集資源量的比 值。當叢集中負載申請的資源值小於等於配 置的縮容閾值時,叢集會自動縮容。預設情況 下,縮容閾值為50%。
縮容觸發時延	叢集滿足配置的縮容閾值時,在配置的縮容觸 發時延到達後,叢集開始縮容。單位:分鐘。 預設情況下是10分鐘。
靜默時間	在叢集添加或刪除節點後,在靜默時間內,叢 集不會再次觸發擴容或縮容,單位 :分鐘。預 設情況下是10分鐘。

## 2. 根據所需要Auto Scaling的資源類型(CPU/GPU),單擊操作列建立。

自动伸缩	€ 返回集群列表							刷新
基本信息								修改禁用
集群ID: cad	7e15c0784848a5be0	02443e9186ccb7	集群名称: test-r	nia		伸缩状态: 🛑 待激活		
缩容阈值: 5	0%		缩容触发时延::	0分钟		静默时间: 10分钟		
伸缩组	状态	实例规格	总实例数	最小实例数	最大实例数	加入中	移出中	操作
CPU								创建
GPU								创建

在伸縮配置頁面填寫以下資訊,建立對應的伸縮組:

配置	說明
地區	所建立伸縮組將要部署到的地區。與伸縮組所在叢集的地區一 致,不可變更。
可用性區域	所建立伸縮組的可用性區域。
專用網路	所建立伸縮組的網路,與伸縮組所在叢集的網路一致。

配置 Worker 節點的資訊。

配置	說明
執行個體規格	伸縮組內執行個體的規格。
系統硬碟	伸縮組的系統硬碟。
掛載資料盤	是否在建立伸縮組時掛載資料盤,預設情況下不掛載。
執行個體數	伸縮組所包含的執行個體數量。
	<ul> <li>说明:</li> <li>執行個體不包含客戶已有的執行個體。</li> <li>預設情況,執行個體的最小值是0台,超過0台的時候,叢 集會預設向伸縮組中添加執行個體,並將執行個體加入到 伸縮組對應的Kubernetes叢集中。</li> </ul>
金鑰組	登入伸縮後的節點時所使用的金鑰組。可以在ECS控制台建立 金鑰組。
	说明: 目前只支援通行金鑰對方式登入。
RDS白名單	Auto Scaling後的節點可以訪問的RDS執行個體。

3. 單擊確定,建立伸縮組。

預期結果

• 在自動調整頁面,可以看到CPU下已建立好一個伸縮組。

自动伸缩 • 返回集群列表	
基本信息	
集群ID: c51b1b2bd791f4496ae28fd3a6b1ac32f	
缩容阈值: 30%	
伸缩组	状态
CPU asg-bp193147i2s1udgawpwh	已启用
GPU	

- 1. 單擊左側導覽列應用 > 部署,進入部署列表頁面。
  - 2. 選擇目的地組群和kube-system命名空間,可以看到名稱為cluster-autoscaler的組件已建立成功。

容器服务 - Kubernetes ▼	部署列表				剧新使用镜像创建	使用模板创建
命名空间	集群 k8s-cluster 🔻 命名空间 kube-syste	m • 2				
授权管理	名称	标签	容器组数量	创建时间		操作
▼ 应用	alicloud-application-controller	app:alicloud-application-controller owner:aliyun	1/1	2018-08-24 10:06:01	详情	编辑   监控   更多▼
容器组	alicloud-disk-controller	app:alicloud-disk-controller	1/1	2018-08-24 10:06:01	详情	编辑   监控   更多▼
服务	alicloud-monitor-controller	k8s-app:alicloud-monitor-controller task:monitoring	1/1	2018-08-24 10:06:01	详情	编辑  监控 更多▼
路由	cluster-autoscaler	app:cluster-autoscaler	0/1 3	2018-09-04 11:37:50	详情	編輯   监控   更多▼
存储声明	coredns	k8s-app:kube-dns	2/2	2018-08-24 10:02:41	详情	编辑  监控 更多▼

# 1.3.10 刪除叢集

您可以通過Container Service管理主控台刪除不再使用的叢集。

#### 操作步骤

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的叢集,進入 Kubernetes 叢集列表頁面。
- 3. 選擇所需的叢集並單擊右側的更多 > 刪除。

容器服务		集群列表		您要	修可以创建 5 个集群,每	每个集群最多可以添加 20 个节点 刷新	i 创建 K	ubernetes 集群	•
Kubernetes SV ▼集群	warm	<b>常见问题:</b> & 如何创建集群 & 扩容和缩音	Weither 🖉 通过 kubectl 连接 Kubernet	ies 集群 🛛 通过命令管理	里应用				
集群 节点		集群名称/ID	地域网络类型	集群状态 创	建时间	Kubernetes 版本	en 18   ;	630± 1 00	2
存储 ▼ 应用		k8stest 🖉	华东1 虚拟专有网络	● 运行中 20	018-04-20 11:02:31	1.9.3	3	皇者口志 投 集群伸缩 删除	司合
部署服务								添加已有节点 集群升级 升级监控服务	

#### 后续操作

刪除叢集失敗

如果您在 ROS 建立的資源下手動添加了一些資源,ROS 是沒有許可權刪除這些資源的。比如在 ROS 建立的 VPC 下手動添加了一個 VSwitch,這樣在就會導致 ROS 刪除時無法處理該 VPC,從 而最終刪除叢集失敗。

Container Service提供了強制移除叢集的功能。通過強制移除功能,您可以在叢集刪除失敗後,強 制移除叢集記錄和 ROS 資源棧。但是,強制移除操作不會自動釋放您手動建立的這些資源,您需 要手動進行釋放。

叢集刪除失敗時,會顯示如下資訊。

集群列表			您最多可以创建	10 个集群,每个集群	群最多可以添加 50 个节点	刷新 创建Serverless Kubern	etes集群 创建 Kubernetes 集群 👻
<b>常见问题: Ø</b> 如何创建集群 Ø 扩容和缩容集群	Ø 通过 kubectl 连接 Kubernetes	集群 &	? 通过命令管理应用				
名称 ▼							
集群名称/ID	集群类型 1	地域	网络类型	集群状态	创建时间	Kubernetes 版本	操作
test-k8s c7837e45e82c8465ba91c981f39b301e5	Kubernetes d	华东1	虚拟专有网络 vpc-bp13cwaie5z	●删除失败	2018-05-08 17:23:27	1.9.3	管理   查看日志   控制台 集群伸缩   更多→

單擊刪除失敗的叢集對應的更多 > 刪除,在彈出的對話方塊裡,您可以看到刪除失敗的資源,勾 選強制移除 並單擊 確定,即可刪除叢集和 ROS 資源棧。



删除集群 -	k8stest					$\times$
	确定要删除集群	k8stest ?				
	☑ 强制删除 亿	又删除集群记录和资	原栈,您需要手动释放以下资源			
	资源ID		资源类型	状态	更新时间	
	sg-bp1jcj;		ALIYUN::ECS::SecurityGroup	● 创建完成	2018-05-08 17:23:40	
	lb-bp14g7		ALIYUN::SLB::LoadBalancer	●创建完成	2018-05-08 17:23:41	
	i-bp1ik2g	100	ALIYUN::ECS::Instance		2018-05-11 09:43:53	
	i-bp11cxjl	and the second	ALIYUN::ECS::Instance		2018-05-11 09:43:54	
	i-bp15mv		ALIYUN::ECS::Instance		2018-05-11 09:43:56	
	eip-bp1az		ALIYUN::VPC::EIPAssociation	●创建完成	2018-05-08 17:23:51	
	ngw-bp1i		ALIYUN::ECS::NatGateway	●创建完成	2018-05-08 17:23:40	
	rsp-bp1f2		ALIYUN::SLB::VServerGroup	●删除失败	2018-05-14 16:27:02	
	eip-bp1az		ALIYUN::VPC::EIP	●创建完成	2018-05-08 17:23:40	
	lb-bp1s3lz		ALIYUN::SLB::LoadBalancer	●创建完成	2018-05-08 17:23:40	
	lb-bp1s3lz		ALIYUN::SLB::BackendServerAt tachment	●创建完成	2018-05-08 17:27:55	
	Kubernet∉ d6-9e06-4 e55	21.22	ALIYUN::RAM::Role	●删除失败	2018-05-14 16:27:02	
	Kubernete d6-9e06-4 e55	10.000	ALIYUN::RAM::Role	● 创建完成	2018-05-08 17:23:41	
	snat-bp12 z	1000	ALIYUN::ECS::SNatEntry	● 创建完成	2018-05-08 17:23:55	
					确定取	消

# 1.4 節點管理

# 1.4.1 添加已有節點

您可以向已經建立的 Kubernetes 叢集中添加已有的 ECS 執行個體。目前,僅支援添加 Worker 節 點。

前提条件

- 如果之前沒有建立過叢集,您需要先##Kubernetes##。
- 需要先把待添加的 ECS 執行個體添加到 Kubernetes 叢集的安全性群組裡。

背景信息

- 預設情況下,每個叢集中最多可包含 40 個節點。如果您需要添加更多節點,請提交工單申請。
- 添加的雲端服務器必須與叢集在同一地區同一 VPC 下。
- 添加已有雲端服務器時,請確保您的雲端服務器有EIP (專用網路),或者相應 VPC 已經配置 了 NAT Gateway。總之,需要確保相應節點能正常訪問公網,否則,添加雲端服務器會失敗。
- Container Service不支援添加不同帳號下的雲端服務器。
- 僅支援添加作業系統為 CentOS 的節點。

操作步骤

- 1. 登入 Container Service#####。
- 2. 在Kubernetes菜單下,單擊左側導覽列中的叢集,進入 Kubernetes 叢集列表頁面。
- 3. 選擇所需的叢集並單擊右側的更多 > 添加已有節點。

進入添加節點頁面,您可以選擇自動添加或手動添加的方式,添加現有雲端服務器執行個體。

自動添加方式會列出當前帳號下可用的 ECS 雲端服務器,在 Web 介面進行選擇,安裝部署, 並自動添加到叢集中;手動添加方式要求您擷取安裝命令,登入到對應 ECS 雲端服務器上進行 安裝,每次只能添加一個 ECS 雲端服務器。

容器服务 - Kubernetes ▼	集群列表					您最多可以创建 5 个集	群,每个集群最多可以添加 40 个节点	刷新	创建 Kubernetes 集群
概览	常见问题: Ø如何创建集群 Ø 扩容和缩容	東群 🔗 通过 kubectl 送	车接 Kubernetes	集群 🔗 通过命令管理应用	Ð				
▼ 集群	名称 ▼								
集群	集群名称/ID	集群类型 地	城 (全部) 👻	网络类型	集群状态	创建时间	Kubernetes 版本		2
节点	k8s-cluster 🗾	Kubernetes 华族	东1	虚拟专有网络 vpc-bp1kd7yn4qn	●运行中	2018-08-24 09:57:11	1.10.4	管理	查看日志
命名空间								3	制除 添加已有节点

4. 您可選擇自動添加的方式,您可以一次性添加多個ECS雲端服務器。

a) 在已有雲端服務器的列表中,選擇所需的ECS雲端服務器,然後單擊下一步。

选择已有云服务器实例	<b>埴写实</b> 例	信息		添加完成
添加方式:				
<ul> <li>自动添加</li> <li>● 手动添加</li> </ul>				
容器服务添加已有ECS实例,会更换系统盘,磁盘ID会变更,原系统盘会被释放	τ.			
请注意: 1. 您原系统盘的用户快照会保留,自动快照则根据您该系统盘属性值"自动快	照是否随磁盘释放"的选项来判断是否保留	留或随磁盘删除;您可以进入磁盘列表	点击"修改属性"查看或修改属性值	•
<ol> <li>为保持足够的快照额度来顺利完成周期性的自动快照策略,您可以删除不認 3. 您在操作前做好相关备份,以免数据丢失给你造成损失。</li> </ol>	需要的用户快照和自动快照。			
实例ID ▼ 请输入实例ID进行精确查询 投资				
■ 实例ID 实例名称	IP地址(ID)	可用区	网络类型	实例规格
C test	- (3単性) (私有)	cn-hangzhou-b	虚拟专有网络	ecs.t5-lc2m1.nano
a shukun-ECS	(私有)	cn-hangzhou-b	虚拟专有网络	ecs.s2.xlarge
IZbp13rsab25rrzk2sb7rqZ	(私有)	cn-hangzhou-g	虚拟专有网络	ecs.sn2ne.large
ecsdocTest	(弹性) (私有)	cn-hangzhou-e	虚拟专有网络	ecs.n1.tiny
				下一步

b) 填寫執行個體資訊,設定登入密碼,然後單擊下一步。

选择已有云服务器实例	<b>填写实例信息</b>	<u> </u>	添加完成
集群ID/名称:	当前要添加的集群信息		
登录方式:	<ul> <li>设置密码</li> </ul>		
* 密码:	••••••• 密码为8-30个字符,必须同时包含三项(大、小写字母,数字和特殊符号),	不支持\"两个符号	
* 确认密码:	•••••		
实例信息:	实例ID	实例名称	
	the Array states	test	
	10 and 10 million	shukun-ECS	
			上-步 下-步

c) 在彈出的對話方塊中,單擊確定,選擇的 ECS 雲端服務器會自動添加到該叢集中。

选择已有云服务器实例	填写实例信息		添加完成
集群ID/名称:	当前要添加的集群信息		
登录方式:	<ul> <li>设置窓码</li> </ul>		
* 密码:	••••••• 密码为8-30个字符,必须同时包含三项(大、小写字母,数字和特殊符号),	不支持\"两个符号	
* 确认密码:	•••••		
实例信息:	实例ID	实例名称	
	- to - for many many many	test	
		shukun-ECS	
			上一步下一步

- 5. (可选) 選擇手動添加的方式。
  - a) 選擇所需的 ECS 雲端服務器,單擊下一步。您一次只能添加一個 ECS 雲端服務器。



b) 進入執行個體資訊頁面,確認無誤後,單擊下一步。

选择已有云服务器实例		填写实例信息	添加完成
集群ID/名称:	当前要添加的集群信息	/ k8s-cluster	
实例信息:	实例ID	实例名称	
	100.000	ALC: NAME OF TAXABLE PARTY.	
			上一步下一步

c) 進入添加節點頁面, 您需要複製其中的執行命令。

选择已有云服务器实例	>	填写实例信息		添加完成
仅支持添加同一 VPC 内,操作系统为 Cen	tOS 的节点			
登录要添加的节点,执行以下命令:				
curl http://aliacs-k8s-cn-hangzhou 41dfa0e9af7a7	.oss-cn-hangzhou.aliyun	cs.com/public/pkg/run/attach/attach_n	ode.sh   bash -sdoo	ker-version 17.06.2-ce-2token 84d9e8.e4b
				完成

- d) 最後單擊完成。
- e) 登入 ECS #####, 單擊左側導覽列中的執行個體, 選擇叢集所在的地區, 選擇需要添加的 ECS 執行個體。
- f) 單擊 ECS 執行個體右側的遠端連線。進入 ECS 執行個體遠端連線介面,根據頁面指導,輸入遠端連線密碼並單擊確定,成功後,輸入上面儲存的命令,單擊確定,開始執行指令碼。

CentOS Linux 7 (Core) Kernel 3.10.0-693.2.2.el7.	复制粘贴命令		$\times$	
izbp13dqaawoex9fws2wsoz lo Password: Last failed login: Tue Mar There were 5 failed login Welcome to Alibaba Cloud E Eroot@izbp13dqaawoex9fws2w	将内容粘贴至文本框中, "文本内容:	最大支持2000个字符,暂不支持中文等非标准键盘值特殊字符。 curl http://allacs-K8s-cn-hangzhou.oss-cn- hangzhou.allyuncs.com/public/pkg/run/attach/attach_node.sh   bash-sdocker-version 17.06.2-cg-1-token 5925bc.396eb/0069b32d8d5endpoint 172.16.253.150:6443 - cluster-gns 172.21.0.10		
		能定	取消	

g)等待指令碼執行成功,該雲端服務器即添加成功。您可以在叢集列表頁面單擊叢集的 ID 查看 該叢集下的節點列表。查看節點是否成功添加到叢集中。

## 1.4.2 查看節點列表

您可以通過命令、Container Service管理主控台的節點列表頁面或者 Kubernetes Dashboard 的節 點列表頁面查看 Kubernetes 叢集的節點列表。

```
通過命令查看
```

📕 说明:

使用命令查看 Kubernetes 叢集的節點列表頁面之前,您需要先設定## *kubectl ## Kubernetes* # #。

通過 kubectl 串連到 Kubernetes 叢集後,運行以下命令查看叢集中的節點。

kubectl get nodes

#### 輸出樣本:

\$ kubectl get nodes			
NAME	STATUS	AGE	VERSION
iz2ze2n6ep53tch701yh9zz	Ready	19m	v1.6.1-2+ed9e3d33a07093
iz2zeafr762wibijx39e5az	Ready	7m	v1.6.1-2+ed9e3d33a07093
iz2zeafr762wibijx39e5bz	Ready	7m	v1.6.1-2+ed9e3d33a07093
iz2zef4dnn9nos8elyr32kz	Ready	14m	v1.6.1-2+ed9e3d33a07093
iz2zeitvvo8enoreufstkmz	Ready	11m	v1.6.1-2+ed9e3d33a07093

### 通過Container Service管理主控台查看

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的叢集 > 節點,進入節點列表頁面。
- 3. 選擇所需的叢集,您可以查看該叢集下的節點列表。

#### 通過 Kubernetes Dashboard 查看

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下, 單擊左側導覽列中的叢集, 進入 Kubernetes 叢集列表頁面。
- 3. 選擇所需的叢集並單擊右側的控制台,進入 Kubernetes Dashboard。

容器服务		集群列表				您最多可以创建 5 个集群	¥ , 每个集群最多可以添加 20 个节点	刷新 创建 Kubernetes 集群 🚽
Kubernetes	Swarm							
▼ 集群		常见问题: Ø 如何创建集群 Ø 扩容和缩容集群	詳 🔗 通过	! kubectl 连接 Kubernetes 集	群 🕜 通过命令管	理应用		
集群	1	名称 ▼						
节点		集群名称/ID	地域	网络类型	集群状态	创建时间	Kubernetes 版本	2 =
存储		k8s-cluster	华东1	虚拟专有网络 vpc-bp13cwaie5z	●运行中	2018-04-20 11:02:31	1.9.3	管理   查看日志   控制台 集群伸缩   更多▼

4. 在 Kubernetes Dashboard 中, 單擊左側導覽列中的節點。即可查看叢集中的節點列表。



## 1.4.3 節點監控

kubernetes 叢集與阿里雲監控服務無縫整合,您可以查看 kubernetes 節點的監控資訊,瞭解 Kubernetes叢集下 ECS 執行個體的節點監控指標。

#### 操作步骤

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的叢集 > 節點,進入節點列表頁面。
- 3. 選擇所需的叢集, 在該叢集下選擇所需的節點。
- 4. 單擊監控,查看對應節點的監控資訊。

容器服务 - Kubernetes ▼	节点列表							刷新标签管理	集群伸缩 添加已有节点	
概范	常见问题 3	常见问题 3 显付最终包年包月								
▼ 集群 1	集群 k8s-clust	ter 🔻	标签过滤 🗸							
集群	IP地址	角色	实例ID/名称	配置	容器组 (已分配量)	CPU (请求量   使用量 )	内存 ( 请求量   使用量 )	更新时间	操作	
节点 2	0.00	Master	de contra de	按量付费 ecs.c5.large	8	47.50%   7.00%	7.12%   60.82%	2018-08-24 17:57:00	详情 监控 移除 调度设置	
命名空间		Worker	10112	按量付费 ecs.c5.large	30	38.00%   8.35%	59.83%   59.90%	2018-08-24 18:06:00	洋橋 監控 移除 调度设置	
授权管理		Master	Automotive that	按量付费 ecs.c5.large	10	42.50%   6.60%	5.28%   59.16%	2018-08-24 17:59:00	详情 监控 移除 调度设置	
部署	10000	Master	1000	按量付费 ecs.c5.large	8	47.50%   4.30%	7.12%   63.12%	2018-08-24 17:58:00	洋情 监控 移除 调度设置	

5. 進入Cloud Monitor管理主控台,查看對應 ECS 執行個體的基本監控資訊,包括 CPU使用率、 網路流入頻寬、網路流出頻寬、系統磁碟 BPS、系統硬碟 IOPS 等指標。



后续操作

要想查看關於作業系統層級的監控指標,需要安裝Cloud Monitor組件。參見######。

## 1.4.4 節點標籤管理

您可以通過Container Service Web 介面對節點進列標籤管理,包括大量新增節點標籤、通過標籤 篩選節點和快速刪除節點標籤。

關於如何使用節點標籤實現節點調度,請參見######。

前提條件

您已經成功建立一個 Kubernetes 叢集,參見##Kubernetes##。

大量新增節點標籤

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的叢集 > 節點,進入節點列表頁面。
- 3. 選擇所需的叢集, 在頁面右上方單擊標籤管理。

容器服务 - Kubernetes ▼	节点列表						刷新标签管理	集群伸缩 添加已有节点
概范	常见问题 3 量	付荔转包年包月					4	
▼ 康群 1	集群 k8s-cluster	▼ 标签过滤 ▼						
集群	IP地址 角	h色 实例ID/名称	配置	容器组(已分配量)	CPU(请求量 使用量)	内存(请求量 使用量)	更新时间	操作
节点 2	М	laster	按量付费 ecs.c5.large	8	47.50%   7.00%	7.12%   60.82%	2018-08-24 17:57:00	详情 监控 移除 调度设置
命名空间	w	Vorker	按量付费 ecs.c5.large	30	38.00%   8.35%	59.83%   59.90%	2018-08-24 18:06:00	洋情 监控 移除 调度设置
授权管理	м	laster	按量付费 ecs.c5.large	10	42.50%   6.60%	5.28%   59.16%	2018-08-24 17:59:00	详情 监控 移除 调度设置
部署	м	laster	按量付费 ecs.c5.large	8	47.50%   4.30%	7.12%   63.12%	2018-08-24 17:58:00	洋情 监控 修除 调度设置

4. 在節點列表中,批量選擇節點,然後單擊添加標籤。

标签管理   ← 返回		刷新
	IP地址	标签
🗭 cn-hangzhou.i-	,27	
	.26	
cn-hangzhou.i-	.29	
🕑 cn-hangzhou.i-	.28	
<ul> <li>添加标签</li> </ul>		

5. 在彈出的添加標籤對話方塊中,輸入標籤的名稱和值,然後單擊確定。

添加		$\times$
名称	group	
值	master	
	确定	关闭

## 您可以在標籤管理頁面,看到批量節點具有相同的標籤。

标签管理 (* 返回)		周朝
<ul> <li>———————————————————————————————————</li></ul>	IP地址	标签
cn-ha	.27	group : master 🔇
cn-ha	.26	group:master 🚳
On-ha	.29	
C cn-hz	.28	group : master 🕲
■ 液加后签		

#### 通過標籤篩選節點

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的叢集 > 節點,進入節點列表頁面。
- 3. 選擇所需的叢集, 在頁面右上方單擊標籤管理。

容器服务 - Kubernetes ▼	节点列表							刷新标签管理	集群伸缩 添加已有节点
概范	常见问题 3	量付费转包	年包月					4	
▼ 無群 1	集群 k8s-cluster	•	标签过滤 👻						
集群	IP地址	角色	实例ID/名称	配置	容器组(已分配量)	CPU (请求量   使用量 )	内存(请求量 使用量)	更新时间	操作
节点 2	0.00	Master	der erfeitet.	按量付费 ecs.c5.large	8	47.50%   7.00%	7.12%   60.82%	2018-08-24 17:57:00	洋情 监控 移除 调度设置
命名空间		Worker	10000	按量付费 ecs.c5.large	30	38.00%   8.35%	59.83%   59.90%	2018-08-24 18:06:00	洋情 监控 移除 调度设置
授权管理		Master	ne franciscus	按量付费 ecs.c5.large	10	42.50%   6.60%	5.28%   59.16%	2018-08-24 17:59:00	详情 监控 移除 调度设置
部署		Master	10000	按量付费 ecs.c5.large	8	47.50%   4.30%	7.12%   63.12%	2018-08-24 17:58:00	洋情 监控 移除 调度设置

4. 選擇某個節點,單擊右側的標籤,如 group:worker,可通過標籤來篩選節點。

您可看到通過 group:worker 標籤成功篩選出所需的節點。

标签管理   ← 返回		T group:worker ③ 周新
名称	IP地址	标签
n cn-hangzhou.	10110-00-00	group:worker 🚳
添加标签		

## 刪除節點標籖

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的叢集 > 節點,進入節點列表頁面。
- 3. 選擇所需的叢集, 在頁面右上方單擊標籤管理。

容器服务 - Kubernetes ▼	节点列表							刷新 标签管理	集群伸缩 添加已有节点
概范	常见问题 3	<b>第四時語</b> 3 量付最終色年包月 4							
▼ 康群 1	集群 k8s-clust	er 🔻	标签过滤 👻						
集群	IP地址	角色	实例ID/名称		容器组(已分配量)	CPU(请求量 使用量)	内存(请求量 使用量)	更新时间	操作
节点 2	10.00	Master	ale trick.	按量付费 ecs.c5.large	8	47.50%   7.00%	7.12%   60.82%	2018-08-24 17:57:00	详情 监控 移除 调度设置
命名空间		Worker	22222	按量付费 ecs.c5.large	30	38.00%   8.35%	59.83%   59.90%	2018-08-24 18:06:00	洋情 监控 修除 调度设置
授权管理		Master	An Installer for	按量付费 ecs.c5.large	10	42.50%   6.60%	5.28%   59.16%	2018-08-24 17:59:00	详情 监控 移除 调度设置
部署	-	Master		按量付费 ecs.c5.large	8	47.50%   4.30%	7.12%   63.12%	2018-08-24 17:58:00	洋情 监控 移除 调度设置

## 4. 選擇某個節點,單擊標籤的刪除表徵圖,如 group:worker。

标签管理		刷新
各次	IP地址	标签
0 m	27	group:master 🚳
() (n-	26	group:master 🕲
α.+	29	group:worker 🚳
a.	28	group:master 🚳
■ 添加标签		

## 您可以看到該節點右側的標籤消失,節點標籤被刪除。

标签管理		刷新
各称	IP地址	标签
0 an-	27	group:master 🚳
an-	26	group:master 🔇
0 ch-	29	group:worker 🕙 🚺
	28	group:master 🔇
(1) 液加标签		

# 1.5 命名空間管理

## 1.5.1 設定資源配額和限制

您可通過Container Service控制台,設定命名空間的資源配額和限制。

前提条件

- 您已成功建立一個 Kubernetes 叢集,參見##Kubernetes##。
- 您已成功建立一個樣本的命名空間test,參見######。
- 您已成功串連到叢集的Master節點地址,參見## kubectl ## Kubernetes ##。

#### 背景信息

在預設的情況下,運行中的 Pod 可以無限制的使用 Node 上的 CPU 和記憶體,這意味著任意一個 Pod 都可以無節制地使用叢集的計算資源,某個命名空間的 Pod 可能會耗盡叢集的資源。

命名空間的一個重要的作用是充當一個虛擬叢集,用於多種工作用途,滿足多使用者的使用需求, 因此,為命名空間配置資源額度是一種最佳實務。

您可為命名空間配置包括 CPU、記憶體、Pod 數量等資源的額度,更多資訊請參見 *Resource Quotas*。

#### 操作步骤

- 1. 登入 Container Service#####。
- 在 Kubernetes 菜單下,單擊左側導覽列的叢集 > 命名空間,選擇所需的叢集,然後單擊命名空間test右側的資源配額與限制。

容器服务 - Kubernetes ▼	命名空间				周新 创建
概览	集群 k8s-cluster	· 3			
	名称	标签	状态	创建时间	(14) (14)
集群	default		● 就绪	2018-08-24 10:02:37	资源配额与限制   編輯   删除
节点	istio-system	istio-injection: disabled	● 就绪	2018-08-30 18:03:28	资源配额与限制   編輯   删除
命名空间 2	kube-public		● 就绪	2018-08-24 10:02:37	资源配额与限制   编辑   删除
▼ 应用	kube-system		● 就绪	2018-08-24 10:02:37	资源配额与限制   编辑   删除
部署	test	np: test	● 就绪	2018-09-04 10:47:21	▲ 资源配额与限制   編編   删除

3. 在資源配額與限制對話方塊中,您可快速設定資源配額和預設資源限制。

道 说明:
對命名空間設定CPU/記憶體配額(ResourceQuota)後,建立容器組時,必須指定CPU/記憶體
資源限制,或為命名空間配置預設資源限制(LimitRange),詳情請參考: <i>Resource Quotas</i>
°

a) 您可為命名空間配置資源限額(ResourceQuota)。

资源配额与限制				$\times$
提示:对命名空间设置CPU/内存 制(LimitRange),详情请参考	配额(ResourceQuota)后,创 : Resource Quotas	]建容器组时,必须指示	主CPU/内存资源限制,或为命名空间配置默认资源限	
资源配额(ResourceQuota)	默认资源限制(LimitRange	)		
^ 计算资源限制				
☑ CPU限制	最大使用量	2	核	
☑ 内存限制	最大使用量	4Gi	0	
^存储资源限制				
☑ 存储空间	最大使用量	1024Gi	0	
✔ 存储声明数量	最大使用量	50	$\uparrow$	
~其他资源限制				
☑ 配置文件数量	最大使用量	100	$\uparrow$	
☑ 容器组数量	最大使用量	50	$\uparrow$	
☑ 服务数量	最大使用量	20	$\uparrow$	
☑ 负载均衡型服务数量	最大使用量	5	$\uparrow$	
✔ 保密字典数量	最大使用量	10	$\uparrow$	
			确定取消	Ś

b) 您可為該命名空間下的容器設定資源限制和資源申請(defaultRequest),從而控制容器的開 銷。詳情參見*https://kubernetes.io//memory-default-namespace/*。

答澒配额(R	esourceQuota )	野认资源限制(LimitRange)	
Densinder ( ···			
	CPU		内仔 🤍
资源限制	0.5	核	512Mi
祭源由请	0.1	核	256Mi
源申请	0.1	核	256Mi

 您已為該命名空間建立資源配額和限制,串連到 Master 節點串連地址,執行以下命令,查看該 命名空間的資源情況。

#kubectl get limitrange,ResourceQuota -n test NAME AGE limitrange/limits 8m NAME AGE resourcequota/quota 8m # kubectl describe limitrange/limits resourcequota/quota -n test Name: limits Namespace: test Type Resource Min Max Default Request Default Limit Max Limit/ Request Ratio \_\_\_\_\_ Container cpu - - 100m 500m -Container memory - - 256Mi 512Mi -Name: quota Namespace: test Resource Used Hard configmaps 0 100 limits.cpu 0 2 limits.memory 0 4Gi persistentvolumeclaims 0 50 pods 0 50 requests.storage 0 1Ti secrets 1 10 services 0 20 services.loadbalancers 0 5

## 1.5.2 編輯命名空間

#### 前提条件

- 您已成功建立一個 Kubernetes 叢集,參見##Kubernetes##。
- 您已成功建立一個樣本的命名空間test,參見######。

#### 背景信息

您可對命名空間進行編輯,對命名空間的標籤進行增、刪、改等操作。

#### 操作步骤

- 1. 登入 Container Service######。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列的叢集 > 命名空間,進入命名空間列表頁面。
- 3. 選擇所需的叢集,選擇所需的命名空間,單擊右側的編輯。

空间				刷新	创建
k8s-duster •					
标签	状态	创建时间			操作
lt	● 就绪	2018-08-24 10:02:37	资源配额与限制	编辑	一删除
ystem istio-injection: disabiled	●就绪	2018-08-30 18:03:28	资源配额与限制	编辑	一删除
public	● 就绪	2018-08-24 10:02:37	资源配额与限制	编辑	一删除
system	🔵 航緒	2018-08-24 10:02:37	资源配额与限制	4	删除
env: test	● 就绪	2018-09-04 10:47:21	资源配额与限制	編辑	删除
e It s	zia k8c-duster K8c-duster	للافة-clustr المحالية المحال المحالية المحالية المح المحالية المحالية ال محالية المحالية	kac-dustar     3       kac     kas       kas     kas	Red-cluster         3           km         Max         Max	المعالية     المعالية       المعالية

在彈出的對話方塊中,單擊編輯,對命名空間的標籤進行修改,如將標籤修改為env:test-V2
 ,然後單擊儲存。

编辑命名空间				$\times$
名称	test 长度为 1-63 个字符	, 只能包含数字、字母	:、和"-",且首尾只能是字母或数字	
标签	<del>变量</del> 名称 env	变量值 test-V2	操作 <u> 保存</u>   删除	]
	名称	值	添加	-
			确定	取消

## 5. 單擊確定,返回命名空間列表,該命名空間的標籤發生變化。

命名空间					刷	新	创建
集群 k8s-cluster 、	Y						
名称	标签	状态	创建时间				操作
default		● 就绪	2018-08-24 10:02:37	资源配额与限制		编辑	删除
istio-system	istio-injection: disabled	● 就绪	2018-08-30 18:03:28	资源配额与限制		编辑	删除
kube-public		<b>●</b> 就绪	2018-08-24 10:02:37	资源配额与限制		编辑	删除
kube-system		● 就绪	2018-08-24 10:02:37	资源配额与限制		编辑	删除
test	env: test-V2	<b>●</b> 就绪	2018-09-04 10:47:21	资源配额与限制		编辑	删除

# 1.6 應用管理

# 1.6.1 使用鏡像建立無狀態Deployment應用

您可以使用鏡像建立一個可公網訪問的nginx應用。

前提条件

建立一個 Kubernetes 叢集。詳情請參見##Kubernetes##。

### 操作步骤

- 1. 登入Container Service#####。
- 在Kubernetes菜單下,單擊左側導覽列中的應用 > 部署,然後單擊頁面右上方的使用鏡像建 立。
- 設定應用程式名稱、部署叢集和命名空間、副本數量和類型,副本數量即應用程式套件含的 Pod數量。然後單擊下一步進入容器配置頁面。

说明	:
	说明

本例中選擇無狀態類型,即Deployment類型。

如果您不設定命名空間,系統會預設使用 default 命名空間。

	应用基本信息	容器配置	$\rangle$	高级配置	$\rangle$	创建完成
应用名称	nginx	]				
		,且不能以-开头				
部署集群:	tuoguan 🔻					
命名空间:	default					
副本数量:	2					
类型	无状态					
						返回 下一步

4. 設定容器配置。

	说明 :
/ <b>-</b>	

您可為應用的Pod設定多個容器。

a) 設定容器的基本配置。

鏡像名稱:您可以單擊選擇鏡像,在彈出的對話方塊中選擇所需的鏡像並單擊確定,本例
 中為 nginx。

您還可以填寫私人 registry。填寫的格式為domainname/namespace/imagename:tag

- 鏡像版本:您可以單擊選擇鏡像版本 選擇鏡像的版本。若不指定,預設為 latest。
- 總是拉取鏡像:為了提高效率,Container Service會對鏡像進行緩衝。部署時,如果發現 鏡像 Tag 與本機快取的一致,則會直接複用而不重新拉取。所以,如果您基於上層業務便 利性等因素考慮,在做代碼和鏡像變更時沒有同步修改 Tag,就會導致部署時還是使用本 機快取內舊版本鏡像。而勾選該選項後,會忽略緩衝,每次部署時重新拉取鏡像,確保使 用的始終是最新的鏡像和代碼。

- 資源限制:可指定該應用所能使用的資源上限,包括 CPU 和 記憶體兩種資源,防止佔用 過多資源。其中,CPU 資源的單位為 millicores,即一個核的千分之一;記憶體的單位為 Bytes,可以為 Gi、Mi 或 Ki。
- 所需資源:即為該應用預留資源額度,包括 CPU 和 記憶體兩種資源,即容器獨佔該資源,防止因資源不足而被其他服務或進程爭搶資源,導致應用不可用。
- Init Container: 勾選該項,表示建立一個Init Container, Init Container包含一些實用的工具,具體參見https://kubernetes.io/docs/concepts/workloads/pods/init-containers/。

		应用基本信息	容器配置		高级配置
容器1	• 添加容器				
	镜像名称:	nginx	镜像版本:	latest	
		选择镜像		选择镜像版本	
本配置	资源限制:	CPU 如:500m 0 内存 如:128Mi 0			
R	所需资源:	CPU 500m 0 内存 如:128Mi 0			
	Init Container				

b) (可选) 配置資料卷資訊。

支援配置本機存放區和雲端儲存。

- 本機存放區:支援主機目錄(hostpath)、配置項(configmap)、保密字典(secret)和
   臨時目錄,將對應的掛載源掛載到容器路徑中。更多資訊參見 volumes。
- 雲端儲存:支援雲端硬碟/NAS/OSS三種雲端儲存類型。

本例中配置了一個雲端硬碟類型的資料卷,將該雲端硬碟掛載到容器中/*tmp* 路徑下,在該路徑下產生的容器資料會儲存到雲端硬碟中。

数据卷:	😏 増加本地存储			
	存储卷类型	挂载源	容器路径	
	😌 増加云存储			
	存储卷类型	挂载源	容器路径	
	云盘	<ul> <li>pvc-yunpan-test</li> </ul>	▼ /tmp	•

c) (可选) 配置Log Service, 您可進行採集配置和自訂Tag設定。

☐ 说明: 請確保已部署Kubernetes叢集,並且在此叢集上已安裝日誌外掛程式。 您可對日誌進行採集配置:

- 日誌庫:即在Log Service中產生一個對應的logstore,用於儲存採集到的日誌。
- 容器內日誌路徑:支援stdout和文本日誌。
  - stdout: stdout 表示採集容器的標準輸出日誌。
  - 文本日誌:表示收集容器內指定路徑的日誌,本例中表示收集/var/log/nginx下所有的文本日誌,也支援萬用字元的方式。

您還可設定自訂 tag,設定tag後,會將該tag一起採集到容器的日誌輸出中。自訂 tag 可協助 您給容器日誌打上tag,方便進行日誌統計和過濾等分析操作。

↔ 米東配置		
日志库	容器内日志路径(可设置为stdout)	
catalina	stdout	•
access	/var/log/nginx	•
<ul> <li>自定义Tag</li> </ul>		
Tag名称	Tag值	

d) (可选) 配置環境變數。

支援通過索引值對的形式為 Pod 配置環境變數。用於給 Pod 添加環境標誌或傳遞配置等,具 體請參見 *Pod variable*。

e) 配置生命週期。

您可以為容器的生命週期配置容器啟動項、啟動執行、啟動後處理和停止前處理。具體參見 https://kubernetes.io/docs/tasks/configure-pod-container/attach-handler-lifecycle-event/。

- 容器啟動項:勾選 stdin 表示為該容器開啟標準輸入;勾選 tty 表示為該容器分配一個虛擬 終端,以便於向容器發送訊號。通常這兩個選項是一起使用的,表示將終端 (tty) 綁定到容 器的標準輸入(stdin)上,比如一個互動程式從使用者擷取標準輸入,並顯示到終端中。
- 啟動執行:為容器設定啟動前置命令和參數。
- 啟動後處理:為容器設定啟動後的命令。
- 停止前處理:為容器設定預結束命令。

	容器启动项:	🔲 stdin 🔲 tty
	启动执行:	命令 "/bin/sh", "-c", "echo Hello from the postStart handler > /usr/share/message"
命周期		参数
Ψ	启动后处理:	命令
	停止前处理:	命令 "/usr/sbin/nginx","-s","quit"

## f) (可选) 設定健全狀態檢查

支援存活檢查(liveness)和就緒檢查(Readiness)。存活檢查用於檢測何時重啟容器; 就緒檢查確定容器是否已經就緒,且可以接受流量。關於健全狀態檢查的更多資訊,請參 見*https://kubernetes.io/docs/tasks/configure-pod-container/configure-liveness-readinessprobes*。

存活检查	☑ 开启						
		Http请求		TCP连接		命令行	~
	协议	нттр			v		
	路径						
	端口						
	Http头	name	value				
	延迟探测时间(秒)	3					
	执行探测频率(秒)	10					
	超时时间(秒)	1					
		1					
	健康國但						
	健康阈值	3					
就绪检查	健康閾値 不健康阈值 ✔ 开启	3					
就建检查	健康阈值 不健康阈值 ✔ 开启	- 3 Http请求		TCP连接		命令行	×
就绪检查	健東間望 不健康間值 ✔ 开启	- 3 Http請求	_	TCP连接		章令行	•
9 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3	健東間望 不健康順道 ✔ 开启 协议	- 3 Http講求 HTTP		TCP连接	•	命令行	v
4 就绪检查	健東周望 不健康樹直 ● 开启 协议 路径	- 3 нцряж нттр		TCP连接	•	章令行	
9 数结合面	健東周望 不健康調道 ● 开启 协议 路径 端口	- 3 нцр爾永 нттр		TCP)连接	•	命令行	~
就塘检查	健東周望 不健康順値 び 开启	- 3 Http请求 HTTP name	value	TCP连接		命令行	<b>v</b>
就達检查	「建東町山」 不健康順值 ● 开启 协议 路径 」第一 Http头 延迟探測时间(秒)	- 3 <u>Http調味</u> HTTP name 3	value	TCP连接	•	命令行	
教神检查	「建東町山」 不健康尚直 「一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一一	- 3 <u>Httpāţ</u> HTTP name 3 10	value	TCP连接		命令行	v
教神	(建東町山) 不健康調値 ● 开启 协议 路径 第口 Http\$ 延迟採刷时间(秒) 此行採測频率(秒) 超时时间(秒)	- 3 <u>Http湖;</u> HTTP лате 3 10 1	value	TCP连接		章令行	· ·
就達检查	健東南區 不健康調備 ✓ 开启 协议 路径 第口 Http头 延迟採興时间(秒) 执行探測誘案(秒) 超时时间(秒) 健康調備		value	TCP连接		命令行	· ·

請求類型	配置說明
HTTP請求	即向容器發送一個HTTPget 請求,支援的參 數包括:
	• 協議:HTTP/HTTPS

請求類型	配置說明
	<ul> <li>路徑:訪問HTTP server 的路徑</li> <li>連接埠:容器暴露的訪問連接埠或連接埠名,連接埠號碼必須介於1~65535。</li> <li>HTTP頭:即HTTPHeaders,HTTP請求中自訂的要求標頭,HTTP允許重複的header。支援索引值對的配置方式。</li> <li>延遲探測時間(秒):即initialDelaySeconds,容器啟動後第一次執行探測時需要等待多少秒,預設為3秒。</li> <li>執行探測頻率(秒):即periodSeconds,指執行探測的時間間隔,預設為10s,最低為1s。</li> <li>逾時時間(秒):即timeoutSeconds, 探測逾時時間。預設1秒,最小1秒。</li> <li>健康閾值:探測失敗後,最少連續探測成功多少次才被認定為成功。預設是1,最小值是1。對於存活檢查(liveness)必須是1。</li> <li>不健康閾值:探測成功後,最少連續探測 失敗多少次才被認定為失敗。預設是3。 最小值是1。</li> </ul>
TCP串連	即向容器發送一個TCP Socket,kubelet將嘗 試在指定連接埠上開啟容器的通訊端。如果 可以建立串連,容器被認為是健康的,如果 不能就認為是失敗的。支援的參數包括: • 連接埠:容器暴露的訪問連接埠或連接埠 名,連接埠號碼必須介於1~65535。 • 延遲探測時間(秒):即initialDel aySeconds,容器啟動後第一次執行探測 時需要等待多少秒,預設為15秒。 • 執行探測頻率(秒):即periodSeconds ,指執行探測的時間間隔,預設為10s, 最低為1s。 • 逾時時間(秒):即timeoutSeconds, 探測逾時時間。預設1秒,最小1秒。 • 健康閾值:探測失敗後,最少連續探測成 功多少次才被認定為成功。預設是1,最 小值是1。對於存活檢查(liveness)必須 是1。

	不健康閾值:探測成功後,最少連續探測 失敗多少次才被認定為失敗。預設是3。 最小值是1。
命令列	過在容器中執行探針檢測命令,來檢測容 的健康情況。支援的參數包括: 命令列:用於檢測容器健康情況的探測命 令。 延遲探測時間(秒):即initialDel aySeconds,容器啟動後第一次執行探測 時需要等待多少秒,預設為5秒。 執行探測頻率(秒):即periodSeconds ,指執行探測的時間間隔,預設為10s, 最低為1s。 逾時時間(秒):即timeoutSeconds, 探測逾時時間。預設1秒,最小1秒。 健康閾值:探測失敗後,最少連續探測成 功多少次才被認定為成功。預設是1,最 小值是1。對於存活檢查(liveness)必須 是1。 不健康閾值:探測成功後,最少連續探測 失敗多少次才被認定為失敗。預設是3。 最小值是1。

- 5. 完成容器配置後,單擊下一步。
- 6. 進行進階設定。
  - a) 設定訪問設定。

您可以設定暴露後端Pod的方式,最後單擊建立。本例中選擇ClusterIP服務和路由 (Ingress),構建一個可公網訪問的nginx應用。

📕 说明 :

針對應用的通訊需求,您可靈活進行訪問設定:

- 內部應用:對於只在叢集內部工作的應用,您可根據需要建立ClusterIP或NodePort類型的服務,來進行內部通訊。
- 外部應用:對於需要暴露到公網的應用,您可以採用兩種方式進行訪問設定:
  - 建立LoadBalancer類型的服務:使用阿里雲提供的負載平衡服務(Server Load Balancer, SLB),該服務提供公網訪問能力。

 建立ClusterIP、NodePort類型的服務,以及路由(Ingress):通過路由提供公網訪問 能力,詳情參見https://kubernetes.io/docs/concepts/services-networking/ingress/。

	应用基	本信息	>	容器配置	高级配置	创建完成
题	服务(Service)	创建				
动间机	路由(Ingress)	创建				
(中緒記器	容體組水平伸缩	□ 开启				
	节点亲和性	添加				
調度设置	应用亲和性	添加				
	应用非亲和性	添加				
						上一步创建

## 1. 在服務欄單擊建立, 在彈出的對話方塊中進行配置, 最後單擊建立。

创建服务		$\times$
名称:	nginx-svc	
类型:	虚拟集群IP   ▼	
端口映射:	<ul> <li>承加</li> <li>服务端口 容器端口 协议</li> <li>80</li> <li>80</li> <li>TCP ▼ ●</li> </ul>	
注解: 标签:	<ul> <li>◆ 添加</li> <li>◆ 添加</li> </ul>	
	创建	取消

- 名稱:您可自主設定,預設為applicationname-svc。
- 類型:您可以從下面3種服務類型中進行選擇。

- 虛擬叢集 IP:即 ClusterIP,指通過叢集的內部 IP 暴露服務,選擇該項,服務只能 夠在叢集內部可以訪問。
- 節點連接埠:即NodePort,通過每個Node上的IP和靜態連接埠(NodePort)
   暴露服務。NodePort服務會路由到ClusterIP服務,這個ClusterIP服務會自動建
   立。通過請求 <<u>NodeIP>:<NodePort></u>,可以從叢集的外部存取一個NodePort服務。
- 負載平衡:即 LoadBalancer,是阿里雲提供的負載平衡服務,可選擇公網訪問或內 網訪問。負載平衡可以路由到 NodePort 服務和 ClusterIP 服務。
- 連接埠映射:您需要添加服務連接埠和容器連接埠,若類型選擇為節點連接埠,還需要 自己設定節點連接埠,防止連接埠出現衝突。支援 TCP/UDP 協議。
- 標籤:您可為該服務添加一個標籤,標識該服務。
- 在路由欄單擊建立,在彈出的對話方塊中,為後端Pod配置路由規則,最後單擊建立。更 多詳細的路由配置資訊,請參見#####。



通過鏡像建立應用時,您僅能為一個服務建立路由(Ingress)。本例中使用一個虛擬機 器主機名稱作為測試網域名稱,您需要在hosts中添加一條記錄。在實際工作情境中,請 使用備案網域名稱。

101.37.224.146 foo.bar.com #即ingress的IP

名称:	nginx-ingress
规则:	◎ 添加
	<u>域名</u>
	foo.bar.com
	使用 **.cptpcaeu43e43420/b85982/a3e94c533.cn-nangzhou.alicontainer.com 或者 自定义 路径
	e.g./
	服务 • 添加
	名称 靖口 权重 权重比例
	nginx-svc 🔻 80 🔻 100 100.0% 🗢
灰度发布:	☑ 添加 设置灰度规则后,符合规则的请求将路由到新服务中。如果设置100以外的权重,满足灰度规则的 请求将会继续依据权重路由到新老版本服务中
注解:	● 添加 重定向注解
TLS:	□ 开启
标签:	● 添加
	创建

3. 在訪問設定欄中,您可看到建立完畢的服務和路由,您可單擊變更和刪除進行二次配置。

包運	NZ/FB									
		应用基本信息	> *	容器配置			高級配置		创建完成	
	服务(Service)	变更 制除								
		服务端口			容器端口			协议		
		80			80			ТСР		
的可设置	路由(Ingress)	交更 制除								
		域名		路径		服务名称		服务端口		
		foo.bar.com				nginx-svc		80		
<b>新加加</b>	容器组水平伸缩	◎开启								
	节点亲和性	添加								
建铁制度	应用亲和性	添加								
	应用非亲和性	添加								
										上一步 创建

b) (可选) 容器組水平伸縮。

您可勾選是否開啟容器組水平伸縮,為了滿足應用在不同負載下的需求,Container Service 支援服容器組(Pod)的Auto Scaling,即根據容器 CPU 和記憶體資源佔用情況自動調整容 器組數量。

	容器组水平伸缩	☑ 开启				
		指标:	CPU使用量		•	
<b>伸縮配置</b>		触发条件	: 使用量 70	%		
		最大副本刻	数: 10	可选范围:2-100		
		最小副本刻	数: 1	可选范围:1-100		

若要啟用自動調整,您必須為容器設定所需資源,否則容器自動調整無法生效。參見容器基 本配置環節。

- 指標:支援CPU和記憶體,需要和設定的所需資源類型相同。
- 觸發條件:資源使用率的百分比,超過該使用量,容器開始擴容。
- 最大容器數量:該Deployment可擴容的容器數量上限。
- 最小容器數量:該Deployment可縮容的容器數量下限。
- c) (可选) 設定調度親和性。

您可設定節點親和性、應用親和性和應用非親和性,詳情參見https://kubernetes.io/docs/ concepts/configuration/assign-pod-node/#affinity-and-anti-affinity。



親和性調度依賴節點標籤和Pod標籤,您可使用內建的標籤進行調度;也可預先為節 點、Pod配置相關的標籤。

1. 設定節點親和性,通過Node節點的Label標籤進行設定。

创建		$\times$
必须满足:		
	kubernetes.io/hostname  In  Cn-hangzhou.i-bp130w  kubernetes.io/hostname In  Cn-hangzhou.i-bp17wa	
尽量满足:	<ul> <li>运 法http://www.interstation.com/interstation</li></ul>	
	权重         ②           100         ③	
	选择器 S 添加 标签名 操作符 标签值	
	kubernetes.io/hostname 🔻 NotIn 🔻 cn-hangzhou.i-bp1fi3o 🗢	
	确定	取消

節點調度支援硬約束和軟約束(Required/Preferred),以及豐富的匹配運算式(In, NotIn, Exists, DoesNotExist. Gt, and Lt):

- 必須滿足,即硬約束,一定要滿足,對應requiredDuringSchedulingIgnore
   dDuringExecution,效果與NodeSelector相同。本例中Pod只能調度到具有對應
   標籤的Node節點。您可以定義多條硬約束規則,但只需滿足其中一條。
- 盡量滿足,即軟約束,不一定滿足,對應preferredDuringSchedulingIgnor
   edDuringExecution。本例中,調度會盡量不調度Pod到具有對應標籤的Node節點。您還可為軟約束規則設定權重,具體調度時,若存在多個合格節點,權重最高的節點會被優先調度。您可定義多條軟約束規則,但必須滿足全部約束,才會進行調度。
- 設定應用親和性調度。決定應用的Pod可以和哪些Pod部署在同一拓撲域。例如,對於相 互連信的服務,可通過應用親和性調度,將其部署到同一拓撲域(如同一個主機)中,減 少它們之間的網路延遲。

创建				×
必须満足:	)添加规则 命名夺间			8
1	default 拓扑域 kubernetes.io/hostname		•	]
2	选择器 ● 添加 查看应用列表 标签名	操作符	标签值	
3	арр	In •	nginx	
尽量满足:	) 添加规则			
	权重 100 命名空间			
	default 拓扑域		•	]
	Kubernetestio/nostname 选择器 ● 添加 查看应用列表	122 / L- MA	一种库	
	app	NotIn V	wordpress	
			确	<del>定</del> 取消

根據節點上啟動並執行Pod的標籤(Label)來進行調度,支援硬約束和軟約束,匹配的運 算式有:In, NotIn, Exists, DoesNotExist。

- 必須滿足,即硬約束,一定要滿足,對應requiredDuringSchedulingIgnore
   dDuringExecution,Pod的親和性調度必須要滿足後續定義的約束條件。
  - 命名空間:該策略是依據Pod的Label進行調度,所以會受到命名空間的約束。
  - 拓撲域:即topologyKey,指定調度時範圍,這是通過Node節點的標籤來實現的, 例如指定為kubernetes.io/hostname,那就是以Node節點為區分範圍;如果指 定為beta.kubernetes.io/os,則以Node節點的作業系統類型來區分。
  - 選取器:單擊選取器右側的加號按鈕,您可添加多條硬約束規則。
  - 查看應用列表:單擊應用列表,彈出對話方塊,您可在此查看各命名空間下的應
     用,並可將應用的標籤匯入到親和性配置頁面。

- 一 硬約束條件:設定已有應用的標籤、操作符和標籤值。本例中,表示將待建立的應
   用調度到該主機上,該主機啟動並執行已有應用具有app:nginx標籤。
- 盡量滿足,即軟約束,不一定滿足,對應preferredDuringSchedulingIgnor
   edDuringExecution。Pod的親和性調度會盡量滿足後續定義的約束條件。對於軟約
   束規則,您可配置每條規則的權重,其他配置規則與硬約束規則相同。

送 说明 :

權重:設定一條軟約束規則的權重,介於1-100,通過演算法計算滿足軟約束規則的節點的權重,將Pod調度到權重最高的節點上。

- 設定應用非親和性調度,決定應用的Pod不與哪些Pod部署在同一拓撲域。應用非親和性 調度的情境包括:
  - 將一個服務的Pod分散部署到不同的拓撲域(如不同主機)中,提高服務本身的穩定 性。
  - 給予Pod一個節點的獨佔存取權限來保證資源隔離,保證不會有其它Pod來分享節點資源。
  - 把可能會相互影響的服務的Pod分散在不同的主機上。

📋 说明:

應用非親和性調度的設定方式與親和性調度相同,但是相同的調度規則代表的意思不同, 請根據使用情境進行選擇。

- 7. 最後單擊建立。
- 建立成功後,預設進入建立完成頁面,會列出應用程式套件含的對象,您可以單擊查看應用詳 情進行查看。

ť	✓ 创建应用任务提交成功	
创建部署	nginx-deployment	成功
创建Service	nginx-svc	成功
创建Ingress	nginx-ingress	成功
	查看应用详情 继续创建	

## 預設進入建立的nginx-deployment的詳情頁面。

部署nginx-deployment 全返回列表				
基本信息				
名称:	nginx-deployment			
命名空间:	default			
创建时间:	2018-08-09 15:28:55			
标签:	app:nginx			
注解:	deployment.kubernetes.io/revision:1			
选择器:	app:nginx			
策略:	RollingUpdate			
状态:	已更新:1个,不可用:0个,共计:1个			
容器组 关联服务 事件	容器组水平伸缩器			
名称		状态	镜像	
nginx-deployment-64ff85b579-hqvjz		●运行中	nginx:latest	

9. 單擊左側導覽列的應用 > 路由,可以看到路由列表下出現一條規則。
| 容器服务 - Kubernetes ▼ | 路由 (Ingress)  |              |                           |                     | 刷新 创建                 |
|---------------------|---------------|--------------|---------------------------|---------------------|-----------------------|
| 概党                  | 集群 test-k8s ▼ | 命名空间 default | •                         |                     |                       |
| ▼ 集群                | 名称            | 議点           | 规则                        | 创建时间                | 擾作                    |
| 集群                  | nginx-ingress | 1 6          | foo.bar.com/ -> nginx-svc | 2018-08-09 15:28:56 | 详情   变更   查看YAML   删除 |
| 节点                  |               |              |                           |                     |                       |
| 存储卷                 |               |              |                           |                     |                       |
| 命名空间                |               |              |                           |                     |                       |
| ▼应用                 |               |              |                           |                     |                       |
| 部署                  |               |              |                           |                     |                       |
| 容器组                 |               |              |                           |                     |                       |
| 服务                  |               |              |                           |                     |                       |
| 路由                  |               |              |                           |                     |                       |

10.在瀏覽器中訪問路由測試網域名稱,您可訪問 nginx 歡迎頁面。

foo.bar.com/?spm=5176.2020520152.0.0.704061b1K4IJgO					
	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.				

# 1.6.2 通過 Kubernetes Dashboard 建立應用

您可以通過 Kubernetes Dashboard 建立應用。

#### 操作步骤

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的叢集,進入 Kubernetes 叢集列表頁面。
- 3. 選擇所需的叢集並單擊右側的控制台,進入 Kubernetes Dashboard。

容器服务 - Kubernetes	•	集群列表					您最多可以创建!	5 个集群,每个集群最多可以添加 40	个节点 刷新	创建 Kubernetes 集群
概范	-	常见问题: & 如何创建集群 & 扩容和缩容集群	🔗 通过 kubectl	连接 Kubernetes 集群	∂ 通过命令管理应用					
▼ <u>集群</u>	Į.	名称 ▼								
集群		集群名称/ID	集群类型	地域 (全部) 👻	网络类型	集群状态	创建时间	Kubernetes 版本		2 🕫
节点存储卷	l	k8s-cluster	Kubernetes	华东1	虚拟专有网络 vpc-bp1kd7yn4qn	●运行中	2018-08-24 09:57:11	1.10.4	管理	査君日志

4. 在 Kubernetes Dashboard 中,單擊頁面右上方的建立。



5. 在彈出的對話方塊中,設定應用的資訊。

您可以通過以下3種方法之一建立應用:

使用文本建立:直接輸入 YAML 或 JSON 格式的編排代碼建立應用。您需要瞭解對應的編排
 格式,如下所示是一個 YAML 格式的編排模板。

1	piVersion: apps/v1 # for versions before 1.9.0 use apps/v1beta2
2	.ind: Deployment
3 -	etadata:
4	name: nginx-depioyment
5 -	pec
6 -	selector:
/ -	matchLabels:
8	app: nginx
9	
10 -	template:
11 -	metadata:
12 -	labels:
13	app: nginx
14 -	spec:
15	containers:
16 -	- name: nginx
1/	image: nginx:1.7.9
18	ports:
19	- ContainerPort: 80

- 使用檔案建立:通過匯入已有的 YAML 或 JSON 設定檔建立應用。
- 使用配置建立:本例中通過指定下邊的設定建立應用。
  - 應用程式名稱:所建立應用的名稱。本樣本中為 nginx。
  - 容器鏡像:所要使用的鏡像的 URL。本樣本使用的是 Docker Nginx。
  - 容器組數量:建立的應用的 pod 個數。
  - 服務:可設定為外部或內部。外部表示建立一個可以從叢集外部存取的服務;內部表示建 立一個叢集內部可以訪問的服務。
  - 進階選項:您可以選擇顯示進階選項,對標籤、環境變數等選項進行配置。此設定將流量 負載平衡到三個 Pod。

使用文本创建	使用文件创建	使用配置创建		
应用名称 *				
nginx-test				一个认此为值的 'app' 标签将被添加到部署机服务上 了解更多 🖉
			10 / 24	
容靜镜像 *				绘)(江帝八开结伊服冬婴幼 LIPI) 武 Docker Livib Coogle Container Pagistay 古的私友结例 了會
nginx				ap(11)のA) かたまたののA and 5 OKE , Ap Docker Hub, Google Container Registry 中山544日1618 5 # 多 四
容靜坦数量 <sup>*</sup>				部署將全被创建,由它维护集群中容器组的所要数量 了解再多 13
3				
ate i				
al ar				可指定内部或外部服务端口,映射到容器所监听的端口 此服务的内部 DNS 名称将会是: nginx-te
ZTRP				了解更多 🛛
两口 <sup>。</sup>		目标端口 *	协议 *	
80		9080	TCP 👻	
			· · · · · · · · · · · · · · · · · · ·	
*		日标端口	TCP -	
			10	
日二方机进行				
並小同級建現				
金田田	TONIA			
HPTE	******			

#### 6. 單擊部署 部署這些容器和服務。

您也可以單擊顯示進階選項 展開進階選項進一步配置相關參數。

#### 后续操作

單擊部署後,您可以查看應用的服務或查看應用的容器。

單擊左側導覽列中的容器,您可以通過左側的表徵圖查看每個 Kubernetes 對象的狀態。



對象仍然處於部署狀態。 👝 表示對象已經完成部署。

# 1.6.3 通過編排模板建立應用

在Container Service kubernetes 模板編排中,您需要自己定義一個應用運行所需的資來源物件,通 過標籤選取器等機制,將資來源物件組合成一個完整的應用。

表示

#### 前提条件

建立一個 kubernetes 叢集,參見##Kubernetes##。

背景信息

本例示範如何通過一個編排模板建立 nginx 應用,包含一個 Deployment 和 Service,後端 Deployment會建立Pod 資來源物件, Service 會綁定到後端 Pod 上,形成一個完整的 nginx 應 用。

操作步骤

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的應用 > 部署,進入部署列表頁面。
- 3. 單擊頁面右上方的使用模板建立。

容器服务 - Kubernetes ▼	部署列表				利新 他用编集的建 他用模能的建
概范	集群 k8s-cluster 、	・ 命名空间 default ・			2
▼ 集群	名称	标签	容器组数量	创建时间	識作
集群	wordpress	app:wordpress	2/2	2018-08-28 17:09:06	详情   编辑   监控   更多▼
节点					
存储卷					
命名空间					
授权管理					
▼ 应用 部署					

- 4. 對模板進行相關配置,完成配置後單擊建立。
  - 叢集:選擇目的地組群。資來源物件將部署在該叢集內。
  - 命名空間:選擇資來源物件所屬的命名空間,預設是 default。除了節點、持久化儲存卷等底
     層計算資源以外,大多數資來源物件需要作用於命名空間。
  - 樣本模板:阿里雲Container Service提供了多種資源類型的 Kubernetes yaml 樣本模板,讓 您快速部署資來源物件。您可以根據 Kubernetes Yaml 編排的格式要求自主編寫,來描述您 想定義的資源類型。
  - 添加部署:您可通過此功能快速定義一個Yaml模板。
  - 使用已有模板:您可將已有編排模板匯入到模板配置頁面。

集群	kßs-cluster 🔹	
命名空间	default •	]
示例模版	自定义 <b>*</b>	]
模版	1       apiVersion: apps/vlbeta2 # for versions before 1.8.0 use apps/vlbeta1         2       kind: Deployment         3       metadata:         4       name: nginx-deployment         5       labels:         6       app: nginx         7       spec:         8       replicas: 2         9       selector:         10       matchlabels:         11       app: nginx         12       template:         13       metadata:         14       labels:         15       app: nginx         16       spec:         17       containers:         18       - name: nginx         19       jmage: nginx:1.7.9 # replace it with your exactly (image_name:tags)         20       ports:         21       - containerPort: 80         22          23          24       apiVersion: vl # for versions before 1.8.0 use apps/vlbetal         25       hidd: Service         26       mame: my-servicel #T000: to specify your service name         29       labels:         20       app: nginx         21       app: nginx	添加部署 使用已有模版
	保存模版 创建	] 1

下面是一個 nginx 應用的樣本編排,基於Container Service內建的編排模板。通過該編排模板, 即可快速建立一個屬於 nginx 應用的 deployment 。

# 🚺 说明 :

Container Service支援Kubernetes Yaml編排,支援通過---符號將資來源物件分隔,從而通過 一個模板建立多個資來源物件。

```
apiVersion: apps/vlbeta2 # for versions before 1.8.0 use apps/
v1beta1
kind: Deployment
metadata:
   name: nginx-deployment
    labels:
      app: nginx
spec:
   replicas: 2
    selector:
     matchLabels:
       app: nginx
    template:
      metadata:
        labels:
         app: nginx
      spec:
        containers:
        - name: nginx
```

```
image: nginx:1.7.9 # replace it with your exactly <</pre>
image_name:tags>
         ports:
          - containerPort: 80
_ _ _
apiVersion: v1  # for versions before 1.8.0 use apps/v1beta1
kind: Service
metadata:
                          #TODO: to specify your service name
  name: my-service1
  labels:
    app: nginx
spec:
  selector:
                          #TODO: change label selector to match
    app: nginx
your backend pod
  ports:
   - protocol: TCP
    name: http
    port: 30080
                           #TODO: choose an unique port on each
node to avoid port conflict
    targetPort: 80
                           ##本例中將type從Nodeport修改為LoadBalancer
   type: LoadBalancer
```

5. 單擊建立後。會提示部署狀態資訊。成功後,單擊Kubernetes 控制台前往Kubernetes

Dashboard 查看部署進度。

<	■ 工作负载 部署							十 创建
集群 命名空间	CPU 使用率			内存使用率(	)			
节点 持久化存储卷	0.001 0.001 0.0008			150 Mi 134 Mi				
角色	B.00005 0.0003 15:07 15:10 1	5:13 15:16	15:20 <b>15:21</b>	0 66.8 Mi 2 33.4 Mi 15:07	15:10	15:13	15:16	15:20 <b>15:21</b>
<b>参名空间</b>		BjiBJ				时间		
default *	部 <b>署</b> 	标签	容器组		日创建 🗢	镜像		Ŧ
工作负载	oginx-deployment-basic	app: nginx	2 / 2		2018-08-31 15:20:21	nginx:1.7.9		:
守护进程集	wordpress	app: wordpress	2/2		2018-08-28 17:09:06	wordpress:4		:
# <b>5</b>								

6. 在 Kubernetes Dashboard 裡,您可以看到 my-service1 服務已成功部署,並暴露了外部入口。 單擊外部入口的訪問地址。

服务						Ŧ
名称 🜲	标签	集群 IP	内部端点	外部端点	已创建 🗢	
my-service1	app: nginx	172.21.10.24	my-service1:30080 TCP my-service1:31035 TCP	:30080	2018-08-31 15:32:27	:

7. 您可以在瀏覽器中訪問 nginx 服務歡迎頁面。

() 1	
AND DESCRIPTION OF A DESCRIPTION OF A DESCRIPTION	and a second
	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.

后续操作

您也可返回Container Service首頁,單擊左側導覽列中的應用 > 服務,查看該nginx的服務。

### 1.6.4 通過命令管理應用

您可以通過命令建立應用或者查看應用的容器。

#### 前提條件

在本地使用命令前,您需要先設定## kubectl ## Kubernetes ##。

#### 通過命令建立應用

可以通過運行以下語句來運行簡單的容器(本樣本中為 Nginx Web 服務器):

```
# kubectl run -it nginx --image=registry.aliyuncs.com/spacexnice/
netdia:latest
```

此命令將為該容器建立一個服務入口,指定 --type=LoadBalancer 將會為您建立一個阿里雲負 載平衡路由到該 Nginx 容器。

```
# kubectl expose deployment nginx --port=80 --target-port=80 --type=
LoadBalancer
```

#### 通過命令查看容器

運行如下命令列出所有 default 命名空間裡正在啟動並執行容器。

root@master # kubectl get pods			
NAME	READY	STATUS	RESTARTS
AGE			
nginx-2721357637-dvwq3	1/1	Running	1
9h			

# 1.6.5 建立服務

Kubernetes Service 定義了這樣一種抽象:一個 Pod 的邏輯分組,一種可以訪問它們的策略,通常 稱為微服務。這一組 Pod 能夠被 Service 訪問到,通常是通過 Label Selector 來實現的。 在 Kubernetes 中,pod 雖然擁有獨立的 IP, 但 pod 會快速地建立和刪除,因此,通過 pod 直接 對外界提供服務不符合高可用的設計準則。通過 service 這個抽象,Service 能夠解耦 frontend(前 端)和 backend(後端) 的關聯,frontend 不用關心 backend 的具體實現,從而實現松耦合的微 服務設計。

更多詳細的原理,請參見 Kubernetes service。

#### 前提條件

您已經成功建立一個 Kubernetes 叢集,參見##Kubernetes##。

#### 步驟1 建立 deployment

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的應用 > 部署,單擊頁面右上方的使用模板建立。

容器服务 - Kubernetes ▼	部署列表				周新使用镜像创建	使用模板创建
概定	集群 k8s-cluster ▼	命名空间 default ▼				2
▼ 集群	名称	标签	容器组数量	创建时间		操作
集群	wordpress	app:wordpress	1/1	2018-08-28 17:09:06	详情	编辑  监控 更多▼
节点						
存储卷						
命名空间						
▼ 应用 部署						

#### 3. 選擇所需的叢集, 命名空間, 選擇範例模板或自訂, 然後單擊建立。

集群	k8s-cluster v	
命名空间	default 🔻	
示例模版	Resource - basic Deployment •	
模版	<pre>1 apiVersion: apps/v1beta2 # for versions before 1.8.0 use apps/v1beta1 2 kind: Deployment 3 ~ metadata: 4     name: nginx-deployment-basic 5 ~     labels: 6          ispp: nginx 7 ~ spec:</pre>	添加部署 使用已有機版
	<pre>8 replicas: 2 9 - selector: 10 - matchlabels: 11 app: nginx 12 - template: 13 - metadata: 14 - labels: 15 app: nginx 16 - spec: 17 containers: 18 - name: nginx 19 image: nginx:1.7.9 # replace it with your exactly <image_name:tags> 20 port=:</image_name:tags></pre>	
	20 ports. 21 - containerPort: 80 ##需要在服务中暴露该端口	
	创建成功,请前往控制台查看创建进度: Kubernetes 控制台	
	保存模版 创建	

本例中,樣本模板是一個 nginx 的 deployment。

```
apiVersion: apps/vlbeta2 # for versions before 1.8.0 use apps/
v1beta1
kind: Deployment
metadata:
  name: nginx-deployment-basic
  labels:
    app: nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
   template:
     metadata:
      labels:
        app: nginx
     spec:
      containers:
       - name: nginx
        image: nginx:1.7.9
                                          # replace it with your
exactly <image_name:tags>
        ports:
         - containerPort: 80
  ##需要在服務中暴露該連接埠
```

4. 單擊Kubernetes 控制台,進入 Kubernetes Dashboard,查看該 Deployment 的運行狀態。

<	■ 工作负载 部署								十 创建
集群 命名空间	CPU 使用率			内存使用率(	)				
节点	0.001 0.001 ₩ 0.0008			85.8 Mi 76.3 Mi					
角色	0.0005 0.0003			份 38.1 Mi 性 19.1 Mi					
存储类 	<b>14:35</b> 14:36 14:4	40 14:43 1 时间	4:46 14:49	14:35	14:36 14	1:40 时间	14:43	14:46	14:49
default 👻	部署								Ŧ
概況	名称 🜲	标签	容器组		已创建 ≑	領像			
工作负载	oginx-deployment-basic	app: nginx	2 / 2		2018-08-31 11:39:51	nginx:1.7.9			:
定时任务	Swordpress	app: wordpress	2 / 2		2018-08-28 17:09:06	wordpress:	ŧ		:
守护进程集									

步驟2 建立服務

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的應用 > 服務,進入服務列表頁面。
- 3. 選擇所需的叢集和命名空間, 單擊頁面右上方的建立。

容器服务 - Kubernetes ▼	服务列表						別新
概范	集群 k8s-cluster ▼	命名空间 default	• 3				4
▼ 集群	名称	类型	创建时间	集群IP	内部端点	外部端点	提作
集群	kubernetes	ClusterIP	2018-08-24 10:02:37	172.21.0.1	kubernetes:443 TCP		详情   更新   查看YAML   删除
节点	wordpress	LoadBalancer	2018-08-28 17:09:06	172.21.8.237	wordpress:80 TCP wordpress:30465 TCP	101.37.115.11:80	洋橋   更新   查看YAML   謝除
命名空间	wordpress-mysql	ClusterIP	2018-08-28 17:09:06	None	wordpress-mysql:3306 TCP	-	详情   更新   查看YAML   删除
・ 应用 1							
部署							
容器组							
服务 2 -							

4. 在彈出的建立服務對話方塊中,進行配置。

创建服务		$\times$
夕钟,		
白你,	nginx-svc	
类型:	负载均衡 ▼ 公网访问 ▼	
关联部署:	nginx-deployment-basic 🔻	
端囗映射:	●添加	
	服务端口 容器端口 协议	
	80 80 TCP 🔻 🗢	
注解:	● 添加 负载均衡配置参数	
	名称      值	
	service.beta.kubernetes.ic 20	
标签:	◆ 添加	
	名称      值	
	app nginx O	
	创建即	消

- 名稱:輸入服務的名稱,本例中為 nginx-vc。
- 類型:選擇服務類型,即服務訪問的方式,包括:

- 節點連接埠:即NodePort,通過每個Node上的IP和靜態連接埠(NodePort)暴露服務。NodePort服務會路由到ClusterIP服務,這個ClusterIP服務會自動建立。通過請求
   <NodeIP>:<NodePort>,可以從叢集的外部存取一個NodePort服務。
- 負載平衡:即 LoadBalancer,指阿里雲提供的負載平衡服務(SLB),可選擇公網訪問或 內網訪問。阿里雲負載平衡服務可以路由到 NodePort 服務和 ClusterIP 服務。
- · 關聯部署:選擇服務要綁定的後端對象,本例中是前面建立的 nginx-deployment-basic。若 不進行關聯部署,則不會建立相關的 Endpoints 對象,您可自己進行綁定,參見 services-*without-selectors*。
- 連接埠映射:添加服務連接埠和容器連接埠,容器連接埠需要與後端的 pod 中暴露的容器連 接埠一致。
- 標籤:您可為該服務添加一個標籤,標識該服務。
- 5. 單擊建立, nginx-svc 服務出現在服務列表中。

服务列表							刷新	创建
集群 k8s-cluster ▼	命名空间 default	T						
名称	英型	创建时间	集群IP	内部端点	外部端点			操作
kubernetes	ClusterIP	2018-08-24 10:02:37	172.21.0.1	kubernetes:443 TCP	-	详情   更新	查看YAML	│删除
nginx-svc	LoadBalancer	2018-08-31 11:46:35	172.21.12.168	nginx-svc:80 TCP nginx-svc:32072 TCP	:80	详情   更新	查看YAML	删除

6. 您可查看服務的基本資料,在瀏覽器中訪問 nginx-svc 的外部端點。

C 1 2/?spm=5176.2020520152.0.0	.49fd16ddLtgTUa	
	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.	

至此,您完成如何建立一個關聯到後端的 deployment 的服務,最後成功訪問 Nginx 的歡迎頁面。

### 1.6.6 服務伸縮

應用建立後,您可以根據自己的需求來進行服務擴容或縮容。

#### 操作步骤

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下, 單擊左側導覽列中的叢集, 進入 Kubernetes 叢集列表頁面。

3. 選擇所需的叢集並單擊右側的控制台,進入 Kubernetes Dashboard。

容器服务 - Kubernetes ▼	集群列表					您最多可以创建;	5 个集群,每个集群最多可以添加 40	) 个节点 刷新	创建 Kubernetes 集群
概流	常见问题: ② 如何创建集群 ③ 扩容和缩容集群	🖉 通过 kubectl	连接 Kubernetes 集群	⌀ 通过命令管理应用					
▼ <u>#</u> #	名称 ▼								
集群	集群名称/ID	集群类型	地域 (全部) 👻	网络类型	集群状态	创建时间	Kubernetes 版本		2
节点存储卷	k8s-cluster	Kubernetes	华东1	虚拟专有网络 vpc-bp1kd7yn4qn	●运行中	2018-08-24 09:57:11	1.10.4	管理	<ul> <li>査看日志   控制台</li> <li>集群伸缩   更多▼</li> </ul>

- 4. 在 Kubernetes Dashboard 中,單擊左側導覽列中的部署,查看部署的 Deployment。
- 5. 選擇所需的 Deployment, 單擊右側的操作表徵圖並單擊伸縮。



6. 在彈出的對話方塊中,將所需容器組數修改為2,並單擊確定。

此操作會擴容一個 Pod,將副本數升到 2。

伸缩部署	
资源 wordpress 将更新 当前状态:已创建 1 イ	所以反映所需数量 內, 需要 2 个
所需容器组数量 2	
取消	确定

#### 后续操作

您可以通過左側的表徵圖查看每個 Kubernetes 對象的狀態。 🚗 君

表示對象仍然處於部署狀

表示對象已經完成部署。 態。

應用部署完成後,您可以單擊某個部署的名稱查看正在啟動並執行 Web 服務的詳細資料。您可以 查看部署中的複本集以及這些複本集所使用的 CPU 和 Memory 資源的相關資訊。

2월 68	
1.H. PH	

如果看不到資源,請耐心等待幾分鐘。



# 1.6.7 查看服務

背景信息

您在建立應用時,如果配置了外部服務,除了運行容器,Kubernetes Dashboard 還會建立外部 Service,用於預配負載平衡器,以便將流量引入到叢集中的容器。

操作步骤

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的應用 > 服務,進入服務列表頁面。
- 3. 您可以選擇所需的叢集和命名空間,選擇所需的服務,單擊右側的詳情。

容器服务 - Kubernetes ▼	服务列表							用納	653
概范	無群 k8s-cluster v 命名空	间 default 🔻	3						
▼ 集群	名称	类型	创建时间	集群IP	内部姚点	外部跳点			操作
集群	kubernetes	ClusterIP	2018-08-24 10:02:37	19.545	kubernetes:443 TCP		洋橋   更新	查看YAML	- 819:
节点 存储器	ngirox-svc	LoadBalancer	2018-08-31 11:46:35		nginx-svc:80 TCP nginx-svc:30920 TCP	80	4 洋橋 更新	查看YAML	ente.
命名空间	wordpress	LoadBalancer	2018-08-28 17:09:06	10.000	wordpress:80 TCP wordpress:30465 TCP	180 IIII	洋橋   更新	查看YAML	
	wordpress-mysql	ClusterIP	2018-08-28 17:09:06	None	wordpress-mysql:3306 TCP		洋情   更新	查看YAML	
部署									
容器组									
168 <b>2</b>									

您可查看服務的名稱、類型、建立時間、叢集 IP 以及外部端點等資訊。本例中您可看到分配給服務的外部端點(IP 位址)。您可以單擊該 IP 位址訪問nginx應用。

nginx-svc   1 返回列表	
基本信息	
名称:	nginx-svc
命名空间:	default
创建时间:	2018-08-31 11:46:35
标签:	app:nginx-v2
注解:	service.beta.kubernetes.io/alicloud-loadbalancer-bandwidth:20
类型:	LoadBalancer
集群IP:	
内部端点:	nginx-svc:80 TCP nginx-svc:30920 TCP
外部端点:	where produce

(可选) 您也可進入叢集的 Kubernetes Dashboard。在左側導覽列中單擊服務,查看所有服務。

### 1.6.8 更新服務

您可以通過Container Service管理主控台的服務列表頁面或者 Kubernetes Dashboard 變更服務的 配置。

#### 通過Container Service控制台服務列表頁面

1. 登入Container Service#####。

- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的應用 > 服務,進入服務列表頁面。
- 3. 選擇叢集和命名空間,選擇所需的服務(本樣本中選擇 nginx-svc),單擊右側的更新。

容器服务 - Kubernetes ×	服务列表						Fallen (1938)
概況	集群 k8s-cluster v 命名空	Biël default 🔻	3				
▼ 東群	名称	與型	创建时间	集群IP	内部跳点	外部跳点	操作
魚群	kubernetes	ClusterIP	2018-08-24 10:02:37	10.00	kubernetes:443 TCP		洋橋   <u>4</u>   查看YAML   删除
节点	nginx-avc	LoadBalancer	2018-08-31 11:46:35	10.000	nginx-svc:80 TCP nginx-svc:32072 TCP	80	详稿   更新   查型YAML   删除
命名空间	wordpress	LoadBalancer	2018-08-28 17:09:06	10.00	wordpress:80 TCP wordpress:30465 TCP	1:80	洋橋   更新   查看YAML   删除
* 应用 1	wordpress-mysql	ClusterIP	2018-08-28 17:09:06	None	wordpress-mysql:3306 TCP		洋橋   更新   查看YAML   删除
##B							
容器组							
ES 2 =							

4. 在彈出的更新對話方塊中,進行配置修改,然後單擊確定。

更新服务		$\times$
名称;	nginx-svc	
类型:	负载均衡 ▼ 公网访问 ▼	
端口映射:	●添加	
	服务端口 容器端口 协议	
	80 80 TCP 🔻 🗢	
注解:	● 添加 负载均衡配置参数	
	名称	
	service.beta.kubernetes.ic 20	
标签:	● 添加	
	名称 值	
	app nginx-v2	
	更新	取消

 在服務列表中,找到所需的服務,單擊右側的詳情,查看服務變化的情況。本例中,對服務的標 籤進行修改。

nginx-svc 全返回列表	
基本信息	
名称:	nginx-svc
命名空间:	default
创建时间:	2018-08-31 11:46:35
标签:	app:nginx-v2
注解:	service.beta.kubernetes.io/alicloud-loadbalancer-bandwidth:20
类型:	LoadBalancer
集群IP:	
内部端点:	nginx-svc:80 TCP nginx-svc:32038 TCP
外部端点:	:80

#### 通過 Kubernetes Dashboard

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的叢集,進入 Kubernetes 叢集列表頁面。
- 3. 選擇所需的叢集並單擊右側的控制台,進入 Kubernetes Dashboard。

窨	器服务 - K	ubernetes 🕶	集群列表					您最多可以创建 5	个集群,每个集群最多可以添加 40 个节。	点刷新	创建 Kubernetes 集群
	概范	*	常见问题: Ø如何创建集群 Ø扩容和缩容集群	🔗 通过 kubecti	连接 Kubernetes 集群	Ø 通过命令管理应用					
-	集群	•	名称 ▼								
	集群	<b>-</b>	集群名称/ID	集群类型	地域 (全部) 👻	网络类型	集群状态	创建时间	Kubernetes 版本		2 🕫
	节点 存储卷		k8s-duster	Kubernetes	华东1	虚拟专有网络 vpc-bp1kd7yn4qn	●运行中	2018-08-24 09:57:11	1.10.4	管理	<ul> <li>查看日志   控制台</li> <li>集群伸缩   更多▼</li> </ul>

- 4. 在 Kubernetes Dashboard 中, 選擇所需的命名空間, 單擊左側導覽列中的服務。
- 5. 選擇所需的服務,單擊右側的操作表徵圖並單擊查看/編輯YAML。

≡	服务发现与负载均衡	服务					十创建
服	<u>Ş</u>						Ŧ
	名称 💲	标签	集群 IP	内部端点	外部端点	已创建 ≑	
0	kubernetes	component: apiser provider: kubernet	10.041	kubernetes:443 TCP		2018-04-19 16:57:11	:
0	nginx-default-svc		242	nginx-default-svc:80 TCP nginx-default-svc:32194 TCP	:80	2018-04-19 15:06:19	:
0	nginx-svc	-	5	nginx-svc:80 TCP nginx-svc:31000 TCP	:80 🖾	2018-04-18 1 删除	
0	test	-	193	test:80 TCP	-	2018-04-18 1 查看/約	扁辑 YAML

6. 在彈出的更新對話方塊中,修改配置資訊,如將 nodePort 修改為 31000,然後單擊更新。

修改 S	ervice	
<b>=</b> =		
1 0	r.	
1 * [		
2	Noncier, with	
5	apiversion . vi ,	
4 °	"name": "nainy-syc"	
5	name: ngrnx-svc,	
7	"selfink", "(ani/u)(namesnaces/default/services/nginy_su/"	
8	"uid", "00435504.4200-1148-3444-001636082656"	
9	"resourceVersion": "226655"	
10	"creationTimestamp": "2018-04-18T09:15:297"	
11	}.	
12 -	"spec": {	
13 -	"ports": [	
14 -	{	
15	"protocol": "TCP",	
16	"port": 80,	
17	"targetPort": 80,	
18	"nodePort": 31000	
19	}	
20	1,	
21 -	"selector": {	
22	"app": "nginx"	
23	},	
24	"clusterIP": "172.21.2.55",	
25	"type": "LoadBalancer",	
26	"sessionAffinity": "None",	
27	"externalTrafficPolicy": "Cluster"	
28	},	
29 -	"status": {	
30 -	"loadBalancer": {	
31 *	"ingress": [	
	取当して、「日本」	
	政府 友利 更初	

# 1.6.9 刪除服務

阿里雲 Kuberntes 服務提供 Web 介面,讓您快速對服務進行刪除。

前提条件

- 您已經成功建立一個 Kubernetes 叢集,參見##Kubernetes##。
- 您已經成功建立一個服務,參見####。

#### 操作步骤

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的應用 > 服務,進入服務列表頁面。
- 3. 選擇叢集和命名空間,選擇所需的服務(本樣本中選擇 nginx-svc),單擊右側的刪除。

容器服务 - Kubernetes ▼	服务列表							刷新创建
概返	集群 k8s-cluster ▼	命名空间 default	<b>v</b> 3					
▼ 集群	名称	类型	创建时间	集群IP	内部端点	外部端点		操作
集群	kubernetes	ClusterIP	2018-08-24 10:02:37	100.00	kubernetes:443 TCP	-	详情   更新   道	
节点	nginx-svc	LoadBalancer	2018-08-31 11:46:35	10.0	nginx-svc:80 TCP nginx-svc:32072 TCP	80	洋情   更新   酒	ē君YAML 删除
命名空间	wordpress	LoadBalancer	2018-08-28 17:09:06	10.0	wordpress:80 TCP wordpress:30465 TCP	:80	洋情   更新   3	語YAML   删除
▼ 应用 1	wordpress-mysql	ClusterIP	2018-08-28 17:09:06	None	wordpress-mysql:3306 TCP		洋情   更新   重	E看YAML   删除
部署								
容器组								
服务 2 Ξ								

4. 在彈出的視窗中,單擊確定,確認刪除,該服務在服務列表中消失。

确认		$\times$
1	删除nginx-svc ?	
	确定耳	汉消

### 1.6.10 查看容器

您可以通過Container Service管理主控台的容器組頁面或者 Kubernetes Dashboard 查看 Kubernetes 叢集的容器組頁面。

#### 通過容器組頁面進行查看

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的應用 > 容器組,進入容器組頁面。
- 3. 選擇所需的叢集和命名空間,選擇所需的容器組,單擊右側的詳情。

```
📃 说明:
```

您可對容器組進行更新和刪除操作,對於通過部署(deployment)建立的容器組,建議您通過 deployment 進行管理。

容器服务 - Kubernetes ▼	容器组								刷新
概览	集群 k8s-cluster v 命名空间 default v								
▼ 集群	名称 🔶	状态 🔷	Pod IP 🗢	节点 🗢	创建时间 🔶	CPU (核)	内存(字节)		
集群	hello-pod	●运行中	10.0.1.166	192.168.35.5	2018-08-30 19:58:47	0	1.469 Mi	详情	更多 ▼
存储卷	nginx-deployment-basic-75675f5897-8inxt	●运行中	10.0.1.167	192.168.35.5	2018-08-31 11:25:36	0	0	3 详情	更多 ▼
命名空间	nginx-deployment-basic-75675f5897-zg5zd	●运行中	10.0.1.168	192.168.35.5	2018-08-31 11:25:36	0	0	详情	●更多▼
<b>-</b> <u>∞</u> ∎ <b>1</b>	wordpress-6d6558cdfb-7kqgx	●运行中	10.0.1.144	192.168.35.5	2018-08-28 17:09:06	0	73.191 Mi	详情	更多 🗸
部署									
容器组 2									

#### 4. 進入容器組的詳情頁,您可查看該容器組的詳情資訊。

容器组 - nginx	器组 - nginx-deployment-basic-75675f5897-cx758									
基本信息										
名称: nginx-deployment-basic-75675f5897-cx758						命名空间: default				
状态: Running						创建时间: 2018-09-03 15:56:25				
节点: cn-han	jzhou.i-bp17w	ac8pqci7498oxqv				Pod IP : 10.0.1.222				
标签: app:	nginx   pod-t	emplate-hash: 312	23191453							
容器 事件	创建者	初始化容器	存储	日志						
名称	COME IN	DOM TO HE HE	镜像	H-D-			端口			
> nginx		nginx:1.7	7.9			TCP 80				

#### 通過 Kubernetes Dashboard 查看容器組

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的叢集,進入 Kubernetes 叢集列表頁面。
- 3. 選擇所需的叢集並單擊右側的控制台,進入 Kubernetes Dashboard。

容器服务 - Kubernetes ▼	集群列表					您最多可以创建 5 个集群,	每个集群最多可以添加 40 个节点	刷新	创建 Kubernetes 集群
概党	常见问题: Ø 如何创建集群 Ø 扩容和缩容集								
* 集群 1	名称 ▼								
集群	集群名称/ID	集群类型	地域 (全部) 👻	网络类型	集群状态	创建时间	Kubernetes 版本		2/=
节点存储卷	k8s-cluster	Kubernetes	华东1	虚拟专有网络 vpc-bp1kd7yn4qn	●运行中	2018-08-24 09:57:11	1.10.4	管理	<ul> <li>查看日志   控制台</li> <li>集群伸縮   更多▼</li> </ul>

4. 單擊左側導覽列中的容器組,查看叢集中的容器組。

或者,您可以單擊左側導覽列中的服務,選擇所需的服務並單擊服務的名稱,查看該服務下的容 器組。

命名空间	■ 服务发现与负载均衡	服务					十 创建
节点							
持久化存储卷	服务						Ŧ
角色	名称 💲	标签	集群 IP	内部端点	外部端点	日创建 ≑	
存储类	e ap 2		10.00	ap:80 TCP ap:30310 TCP	80 🖾	2018-09-04 14:53:21	:
命名空间 default <del>·</del>	kubernetes	component: apiserver provider: kubernetes		kubernetes:443 TCP	-	2018-08-24 10:02:37	:
概況							
工作负载							
定时任务							
守护进程集							
部署							
任务							
容蓋担							
副本集							
副本控制器							
有状态副本集							
服务发现与负载均衡							
访问权							
IRS 1							

5. 您可以通過左側的表徵圖查看每個 Kubernetes 對象的狀態。

表示對象仍然處於部署狀 ۵



# 6. 選擇所需的容器組,單擊容器組的名稱,您可以查看容器組的詳細資料以及容器組使用的 CPU

和 Memory。



7. 選擇所需的容器組,單擊右上方的日誌查看容器的日誌資訊。

容器日志 wordpress 🔹	立于 wordpress-6bbb7ffc-ld6gk ~	A	Tr	Ō	0	Ŧ
WordPress not found in /var/www Complete! WordPress has been se	w/html - copying now uccessfully copied to /var/www/html					
AH00558: apache2: Could not re message	liably determine the server's fully qualified domain name, using 172.16.5.11. Set the 'ServerName' directive	global	ly to	suppress	this	
AH00558: apache2: Could not re message	liably determine the server's fully qualified domain name, using 172.16.5.11. Set the 'ServerName' directive	global	ly to	suppress	this	
[Mon Oct 30 16:00:10.245827 20 [Mon Oct 30 16:00:10.245862 20	[7] [mpm_prefork:notice] [pid 1] AH00163: Apache/2.4.10 (Debian) PHP/5.6.31 configured resuming normal ope [7] [core:notice] [pid 1] AH00094: Command line: 'apache2 -D FOREGROUND'	eratior	S			

8. 您可以單擊右側的刪除刪除該容器組。

<	三 工作负载 容器组						+ 创建
集群	CPU 使用率			内存使用率 📵			
节点	0.146 0.130			215 Mi 191 Mi			
角色存储类	€ 0.088 2 0.085 0.033			印 143 Mi 任 95.4 Mi 47.7 Mi			
命名空间 default ▼	17:50 17:53	17:58 18:00 83140	18:04	17:50	17:53	17:58 18:00 ខ្យាំផ្ស	18:04
概览 工作 <del>会</del> 新	容器组						Ŧ
→1F5050 守护进程集	名称 ≑	点	状态 ≑	已重启	巳创 韋 CPU(核)	内存 (字节)	
部署	nginx-584b86d598-s7dsr	cn-hangzhou.l- bp16pdfarljspng93m97	Running	0	2018-01- 29 17:54:14	38 23.	234 Mi <b>2</b>
(19) 容器组	wordpress-c4ff4f5df-4bdh9	cn-hangzhou.i- bp16pdfarljspng93m97	Running	0	2018-01- 29 16:42:37	42 89.0	051 Mi 删除
副本集 副本控制器	wordpress-c4ff4f5df-655p8	cn-hangzhou.i- bp16pdfarljspng93m97	Running	0	2018-01- 29 16:42:37	41 74.0	查看/编辑 YAML
有状态副本集	http-svc-ct6j2	cn-hangzhou.i- bp16pdfarljspng93m97	Running	0	2018-01- 28 0 18:46:49	1.5	12 Mi 🚍 🗄
访问权	Shttp-svc-nxw6k	cn-hangzhou.i- bp16pdfarljspng93m97	Running	0	2018-01- 28 0 18:46:49	1.90	06 Mi 🔳 🗄
8235							

# 1.6.11 變更容器配置

您可以通過Container Service管理主控台變更容器組(Pod)的配置。

操作步骤

- 1. 登入Container Service######。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的應用 > 容器組,進入容器組列表。
- 3. 選擇所需的叢集和命名空間,選擇所需的容器組,單擊右側的更多 > 編輯。

容器服务 - Kubernetes ▼	容器组							刷新
概览	集群 k8s-cluster ▼ 命名空间 default ▼	3						
▼ 集群	名称 🔷	状态 🔷	Pod IP 💠	节点 🔷	创建时间 🔷	CPU (核)	内存(字节)	4
集群 节点	hello-pod	●运行中	10.0.1.166	192.168.35.5	2018-08-30 19:58:47	0	1.469 Mi	洋情   更多  ▼
存储卷	nginx-deployment-basic-75675f5897-5h884	●运行中	10.0.1.137	192.168.35.5	2018-08-24 11:42:47	0	1.406 MI	终端
命名空间	nginx-deployment-basic-75675f5897-mrkmw	●运行中	10.0.1.136	192.168.35.5	2018-08-24 11:42:47	0	1.391 Mi	日志
授权管理 ▼应用 1	tomcat-deployment-65799d5fc4-p8mcs	●运行中	10.0.1.140	192.168.35.5	2018-08-28 16:43:44	0	333.543 Mi	洋情 │ 更多 ▼
部署	wordpress-6d6558cdfb-7kqgx	●运行中	10.0.1.144	192.168.35.5	2018-08-28 17:09:06	0	64.082 MI	详情 │ 更多 ▼
容器组 2 三								

4. 更新容器組的配置並單擊更新。



# 1.6.12 指定節點調度

您可通過為節點設定節點標籤,然後通過配置 nodeSelector 對 pod 調度進行強制限制式,將 pod 調度到指定的 Node 節點上。關於 nodeSelector 的詳細實現原理,請參見 *nodeselector*。

出於業務情境需要,如您需要將管控服務部署到 Master 節點;或者將某些服務部署到具有 SSD 盤 的機器上。您可以採用這種方式實現指定節點調度。

前提條件

您已經成功部署一個 Kubernetes 叢集,參見##Kubernetes##。

#### 步驟1 為節點添加標籤

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的叢集 > 節點,進入節點列表頁面。
- 3. 選擇所需的叢集, 在頁面右上方單擊標籤管理。

容器服务 - Kubernetes ▼	节点列表							刷新标签管理	集群伸缩 添加已有节点
概范	常见问题 3	常见问题 3 显付器转包车包月 4							
<ul> <li>★ (##)</li> </ul>	集群 k8s-clust	er 🔻	标签过滤 👻						
集群	IP地址	角色	实例ID/名称		容器组(已分配量)	CPU(请求量 使用量)	内存 (请求量   使用量 )	更新时间	操作
节点 2	0.00	Master	the second second	按量付费 ecs.c5.large	8	47.50%   7.00%	7.12%   60.82%	2018-08-24 17:57:00	详情 监控 移除 调度设置
命名空间		Worker	1000	按量付费 ecs.c5.large	30	38.00%   8.35%	59.83%   59.90%	2018-08-24 18:06:00	洋情 监控 移除 调度设置
授权管理		Master	And the second s	按量付费 ecs.c5.large	10	42.50%   6.60%	5.28%   59.16%	2018-08-24 17:59:00	详情 监控 移除 调度设置
部署		Master	10000	按量付费 ecs.c5.large	8	47.50%   4.30%	7.12%   63.12%	2018-08-24 17:58:00	洋情 监控 移除 调度设置

4. 在節點列表中,選擇所需節點,然後單擊添加標籤。本例中選擇一個 worker 節點。

标签管理		刷新
□ 名称	IP地址	标签
Cn-hangzhou.	.27	group:master 😵
cn-hangzhou.	.26	group : master 🚳
Cn-hangzhou.	.29	
on-hangzhou.	.28	group:master 🛇
□ 添加标签 <b>2</b>		

5. 在彈出的添加標籤對話方塊中,輸入標籤的名稱和值,然後單擊確定。

添加		$\times$
名称	group	
值	worker	
	确定	关闭

您可以在標籤管理頁面,看到該節點具有 group:worker標籤。

标签管理		
□ 名称	IP地址	标签
G2	27	group:master 🔇
2ba	26	group : master 🚳
18cn	29	group:worker 🔕
9nrg	28	group : master 🚳
□ 液加标签		

您也可以通過命令 kubectl label nodes <node-name> <label-key>=<label-value> 為節點添加標籤。

#### 步驟2 將 pod 部署到指定節點

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的應用 > 部署,進入部署列表頁面。
- 3. 單擊頁面右上方的使用模板建立。

容器服务		部署列表				使用镜像创建 使用模板	创建	刷新
Kubernetes	Swarm	集群 k8s-cluster 🔻 命名空间 default	Ŧ			2		
集群		名称	标签	容器组数量	创建时间			操作
节点		nginx-deployment-basic	app:nginx	2	2018-04-23 13:57:50	详情	更新	删除
存储 ▼ 应用	1	test-mariadb	app:test-mariadb chart:mariadb-0.7.0 release:test heritage:Tiller	1	2018-04-23 14:12:26	详情	更新	删除
部署 容器组	-	test-wordpress	app:test-wordpress chart:wordpress-0.6.13 release:test heritage:Tiller	1	2018-04-23 14:12:26	详情	更新	删除
服务								

- 4. 對模板進行相關配置,部署一個 pod,完成配置後,單擊建立。
  - 叢集:選擇所需的叢集。
  - 命名空間:選擇資來源物件所屬的命名空間,本例中是 default。
  - 樣本模板:本樣本選擇自訂模板。

集群	k8s-cluster	•
命名空间	default	
示例模版	自定义	•
模版	<pre>1 apiVersion: v1 2 kind: Pod 3 metadata: 4 labels: 5 name: hello-pod 6 name: hello-pod 7 spec: 8 containers: 9 image: nginx 10 imagePullPolicy: INotPresent 11 name: hello-pod 12 - ports: 13 containerPort: 8080 14 portocol: TCP 15 resources: {} 16 securityContext: 17 capabilities: {} 18 privileged: false 19 terminationMessagePath: /dev/termination-log 20 dnsPolicy: ClusterFirst 21 restartPolicy: Always 22 nodeSelector: 23 metasis ClusterFirst 24 schoos: ClusterFirst 25 nodeSelector: 26 nodeSelector: 27 securityContext: 27 nodeSelector: 28 nodeSelector: 29 nodeSelector: 20 nodeSelector: 20 nodeSelector: 21 nodeSelector: 23 nodeSelector: 24 schoos: ClusterFirst 25 schoos: ClusterFirst 26 schoos: ClusterFirst 27 schoos: ClusterFirst 28 schoos: ClusterFirst 29 schoos: ClusterFirst 20 schoos: ClusterFirst 20 schoos: ClusterFirst 21 schoos: ClusterFirst 23 schoos: ClusterFirst 24 schoos: ClusterFirs</pre>	
	创建	

```
本樣本的編排模板如下。
```

```
apiVersion: v1
kind: Pod
metadata:
  labels:
    name: hello-pod
  name: hello-pod
spec:
  containers:
    - image: nginx
      imagePullPolicy: IfNotPresent
      name: hello-pod
      ports:
        - containerPort: 8080
         protocol: TCP
      resources: {}
      securityContext:
        capabilities: {}
        privileged: false
      terminationMessagePath: /dev/termination-log
  dnsPolicy: ClusterFirst
  restartPolicy: Always
  nodeSelector:
                                            ##注意與前面配置的節點標籤
    group: worker
一致
```

status: {}

### 5. 單擊建立 後,會提示部署狀態資訊。 成功後,單擊Kubernetes 控制台前往控制台查看部署狀 態。

<	≡ 概況				+ 818
	容器组				Ŧ
5500 5500	名称 🗘	节点 状态 🗣	已重启 已创建 ◆ CPU(核)	内存(李节)	
节点	S hello-pod	cn-hangzhou.i-bp1idqj54m9at16du8cn Running	0 2018-04-25 13:59:28 -	1.438 Mi	≡ :
持久化存储卷	test-mariadb-9bb8f87dd-v9n6f	cn-hangzhou.i-bp1idqj54m9at16du8cn Running	0 2018-04-23 0.002	138.035 M	≡ :
角色	test-wordpress-6945bc9556-mpdxn	cn-hangzhou.i-bp1idqj54m9at16du8cn Running	0 2018-04-23 0.004	171.188 Mi	≡ :
存储英	ginx-deployment-basic-6c54bd5869-l8trh	cn-hangzhou.i-bp1idqj54m9at16du8cn Running	0 2018-04-23 0 13:57:50 0	1.383 Mi	≡ :
命名空间 default · ·	oginx-deployment-basic-6c54bd5869-mw65q	cn-hangzhou.i-bp1idqj54m9at16du8cn Running	0 2018-04-23 0 13:57:50 0	1.367 MI	≡ :
概况	副本集				Ŧ

6. 您可單擊 pod 名稱,進入詳情頁,瞭解 pod 詳情。

您可看到 pod 的標籤、所處的節點 ID 等資訊,表明該 pod 已經成功部署到具有 group:

worker 標籤的指定節點上。

■ 工作负载 容器组 hello-pod	🔁 运行命令 🖙 日志 🧪 編輯   删除 🕂 创建
详情	
名称: hello-pod	网络
命名空间: default	节点:
标签: name: hello-pod	IP: 10
创建时间: 2018-08-30 19:58:47	
状态: Running	
QoS 等级: BestEffort	

# 1.6.13 查看鏡像列表

操作步骤

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中市場 > 鏡像,進入鏡像列表頁面。

容器服务		镜像列表 常用镜像 Dock	er官方捐俸	
Kubernetes	Swarm			
概览		• mongoDB	mongo	创建应用 →
▼ 集群			突至: 27T 光敏: DUCKEC_NUS 第781 MongoDB document databases provide high availability and easy scalability.	
来軒 节点 存砝		MySQL	mysql 类型:公开 未原:DOCKER_HUB ★1225 MySQL is a widely used, open-source relational database management system (RDBMS).	创建应用→
命名空间 ▼ 应用		NGINX	nginx 樂型:公开 未愿:DOCKER_HUB ★1155 Official build of Nginx.	创建应用 →
部署容器组		<b>↓</b> Derestr	php 美型:公开 未愿:DOCKER_HUB ★378 While designed for web development, the PHP scripting language also provides general-purpose use.	创建应用 →
服务 路由 发布		PostgreSQL	<b>postgres</b> 英型: 公开: 朱愿: DOCKER_HUB ★224 The PostgreSQL object-relational database system provides reliability and data integrity.	创建应用 →
配置项 保密字典		🧉 redis	redis 樂型:公开 朱厚:DOCKER_HUB ★1257 Redis is an open source key-value store that functions as a data structure server.	创建应用 →
▼ 市场 镜像	1	X	<b>tomcat</b> 典型:公开 来意:DOCKER_HUB ★793 Apache Tomcat is an open source implementation of the Java Serviet and JavaServer Pages technologies	创建应用 →

您可以查看鏡像的種類過。

- 常用鏡像: Container Service推薦的一些常用鏡像。
- Docker 官方鏡像: Docker Hub 提供的官方鏡像。

### 1.7 負載平衡及路由管理

1.7.1 概述

Kubernetes 叢集提供了多種訪問容器應用的方式,支援通過阿里雲的負載平衡服務(Server Load Balancer)或者 Ingress 方式來訪問內部服務的和實現負載平衡。

### 1.7.2 通過負載平衡(Server Load Balancer)訪問服務

您可以使用阿里雲負載平衡來訪問服務。



如果您的叢集的cloud-controller-manager版本大於等於v1.9.3,對於指定已有SLB的時候,系統預 設不再為該SLB處理監聽,使用者需要手動設定該SLB的監聽規則。

執行以下命令,可查看cloud-controller-manager的版本。

root@master # kubectl get po -n kube-system -o yaml|grep image:|grep cloud-con|uniq image: registry-vpc.cn-hangzhou.aliyuncs.com/acs/cloud-controllermanager-amd64:v1.9.3

#### 通過命令列操作

1. 通過命令列工具建立一個 Nginx 應用。

2. 為 Nginx 應用建立阿里雲負載平衡服務,指定 type=LoadBalancer 來向外網使用者暴露

Nginx 服務。

root@master # kubectl expose deployment nginx --port=80 --targetport=80 --type=LoadBalancer root@master # kubectl get svc NAME CLUSTER-IP EXTERNAL-IP PORT(S) AGE nginx 172.19.10.209 101.37.192.20 80:31891/TCP 4s

3. 在瀏覽器中訪問 http://101.37.192.20, 來訪問您的 Nginx 服務。

#### 通過 Kubernetes Dashboard 操作

1. 將下面的 yml code 儲存到 nginx-svc.yml檔案中。

```
apiVersion: v1
kind: Service
metadata:
    labels:
        run: nginx
    name: http-svc
    namespace: default
spec:
    ports:
    - port: 80
        protocol: TCP
        targetPort: 80
    selector:
        run: nginx
    type: LoadBalancer
```

2. 登入Container Service#####, 單擊目的地組群右側的控制台,進入 Kubernetes Dashboard 頁

面。

3. 單擊建立,開始建立應用。



- 4. 單擊使用檔案建立。選擇剛才儲存的nginx-svc.yml 檔案
- 5. 單擊上傳。

這樣會建立一個阿里雲Server Load Balancer執行個體指向建立的 Nginx 應用,服務的名稱為 http-svc。

6. 在 Kubernetes Dashboard 上定位到 default 命名空間,選擇服務。

可以看到剛剛建立的 http-svc 的 Nginx 服務和機器的負載平衡地址 http://114.55.79.24 :80。

<	☰ 服务发现与负载均衡	服务
集群	服务	
命名空间	2 2 2	行交
节点		10/22
持久化存储卷	Mttp-svc	run: ngi
角色	kubernetes	component: apiser. provider: kubernet
存储类		promotine
命名空间		
default 👻		
概览		
工作负载		
守护进程集		
部署		
任务		
容器组		
副本集		
副本控制器		
有状态副本集		
服务发现与负载均衡		
访问权		
服务 1		

7. 將該地址拷貝到瀏覽器中即可訪問該服務。

#### 更多資訊

#### 注釋

阿里雲可以通過注釋的形式支援豐富的負載平衡功能。

使用已有的內網 SLB

需要指定兩個annotation。注意修改成您自己的 Loadbalancer-id。

```
apiVersion: v1
kind: Service
metadata:
 annotations:
   service.beta.kubernetes.io/alicloud-loadbalancer-address-type:
intranet
    service.beta.kubernetes.io/alicloud-loadbalancer-id: your-
loadbalancer-id
 labels:
   run: nginx
 name: nginx
 namespace: default
spec:
 ports:
  - name: web
   port: 80
   protocol: TCP
   targetPort: 80
 selector:
   run: nginx
  sessionAffinity: None
  type: LoadBalancer
```

然後儲存為 slb.svc 後,執行 kubectl apply -f slb.svc。

#### 建立 HTTPS 類型的 Loadbalancer

先在阿里雲控制台上建立一個認證並記錄 cert-id,然後使用如下 annotation 建立一個 HTTPS 類型 的 SLB。

```
apiVersion: v1
kind: Service
metadata:
    annotations:
        service.beta.kubernetes.io/alicloud-loadbalancer-cert-id: your-
cert-id
        service.beta.kubernetes.io/alicloud-loadbalancer-protocol-port: "
https:443"
    labels:
        run: nginx
        name: nginx
```

```
namespace: default
spec:
    ports:
    - name: web
        port: 443
        protocol: TCP
        targetPort: 443
    selector:
        run: nginx
    sessionAffinity: None
    type: LoadBalancer
```



注釋的內容是區分大小寫。

注釋	描述	預設值
service.beta.kubernetes.io/ alicloud-loadbalancer-protocol -port	多個值之間由逗號分隔,比 如:https:443,http:80	無
service.beta.kubernetes.io/ alicloud-loadbalancer-address -type	取值可以是 internet 或者 intranet	internet
service.beta.kubernetes.io /alicloud-loadbalancer-slb- network-type	負載平衡的網路類型,取值可 以是 classic 或者 vpc	classic
service.beta.kubernetes.io/ alicloud-loadbalancer-charge- type	取值可以是 paybytraffic 或者 paybybandwidth	paybybandwidth
service.beta.kubernetes.io/ alicloud-loadbalancer-id	Server Load Balancer執行個體 的 ID。通過 loadbalancer-id 指 定您已有的 SLB,已有 listener 會被覆蓋, 刪除 service 時該 SLB 不會被刪除。	無
service.beta.kubernetes.io/ alicloud-loadbalancer-backend -label	通過 label 指定 SLB 後端掛哪 些節點。	無
service.beta.kubernetes.io/ alicloud-loadbalancer-region	負載平衡所在的地區	無
service.beta.kubernetes. io/alicloud-loadbalancer- bandwidth	負載平衡的頻寬	50

注釋	描述	預設值
service.beta.kubernetes.io/ alicloud-loadbalancer-cert-id	阿里雲上的認證 ID。您需要先 上傳認證	""
service.beta.kubernetes.io/ alicloud-loadbalancer-health- check-flag	取值是 on 或者 off	預設為 off。TCP 不需要改參 數。因為 TCP 預設開啟健全狀 態檢查,使用者不可設定。
service.beta.kubernetes.io/ alicloud-loadbalancer-health- check-type	參 見CreateLoadBalancerTCPListe	無 ner
service.beta.kubernetes.io/ alicloud-loadbalancer-health- check-uri	參 見CreateLoadBalancerTCPListe	無 ner
service.beta.kubernetes.io/ alicloud-loadbalancer-health- check-connect-port	參 見CreateLoadBalancerTCPListe	無 ner
service.beta.kubernetes.io/ alicloud-loadbalancer-healthy- threshold	參 見CreateLoadBalancerTCPListe	無 ner
service.beta.kubernetes. io/alicloud-loadbalancer- unhealthy-threshold	參 見CreateLoadBalancerTCPListe	無 ner
service.beta.kubernetes.io/ alicloud-loadbalancer-health- check-interval	參 見CreateLoadBalancerTCPListe	無 ner
service.beta.kubernetes.io/ alicloud-loadbalancer-health- check-connect-timeout	參 見CreateLoadBalancerTCPListe	無 ner
service.beta.kubernetes.io/ alicloud-loadbalancer-health- check-timeout	參 見 <i>CreateLoadBalancerTCPListe</i>	無 ner

# 1.7.3 Ingress 支援

在 Kubernetes 叢集中,Ingress是授權入站串連到達叢集服務的規則集合,為您提供七層負載平衡 能力。您可以給 Ingress 配置提供外部可訪問的 URL、負載平衡、SSL、基於名稱的虛擬機器主機 等。

#### 前置條件

### 為了測試複雜路由服務,本例中建立一個 nginx 的樣本應用,您需要事先建立 nginx 的 deployment ,然後建立多個 Service,用來觀察路由的效果。實際測試請替換成自己的服務。

```
root@master # kubectl run nginx --image=registry.cn-hangzhou.aliyuncs.
com/acs/netdia:latest
root@master # kubectl expose deploy nginx --name=http-svc --port=80 --
target-port=80
root@master # kubectl expose deploy nginx --name=http-svc1 --port=80
--target-port=80
root@master # kubectl expose deploy nginx --name=http-svc2 --port=80
--target-port=80
root@master # kubectl expose deploy nginx --name=http-svc3 --port=80
--target-port=80
```

#### 簡單的路由服務

通過以下命令建立一個簡單的 Ingress,所有對 /svc 路徑的訪問都會被路由到名為 http-svc 的服

務。nginx.ingress.kubernetes.io/rewrite-target: /會將/svc路徑重新導向到後端服

務能夠識別的/路徑上面。

```
root@master # cat <<EOF | kubectl create -f -
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
  name: simple
  annotations:
   nginx.ingress.kubernetes.io/rewrite-target: /
spec:
 rules:
  - http:
      paths:
       - path: /svc
        backend:
          serviceName: http-svc
          servicePort: 80
EOF
root@master # kubectl get ing
NAME
                HOSTS
                               ADDRESS
                                                 PORTS
                                                           AGE
simple
                               101.37.192.211
                                                 80
                                                           11s
```

現在訪問 http://101.37.192.211/svc 即可訪問到 Nginx 服務。

#### 基於網域名稱的簡單扇出路由

如果您有多個網域名稱對外提供不同的服務,您可以產生如下的配置達到一個簡單的基於網域名稱 的扇出效果。

```
root@master # cat <<EOF | kubectl create -f -
apiVersion: extensions/vlbetal
kind: Ingress
metadata:
    name: simple-fanout</pre>
```
```
spec:
 rules:
  - host: foo.bar.com
   http:
     paths:
      - path: /foo
        backend:
          serviceName: http-svcl
          servicePort: 80
      - path: /bar
        backend:
          serviceName: http-svc2
          servicePort: 80
  - host: foo.example.com
   http:
      paths:
      - path: /film
        backend:
          serviceName: http-svc3
          servicePort: 80
EOF
root@master # kubectl get ing
NAME
                HOSTS
                              ADDRESS
                                                PORTS
                                                          AGE
simple-fanout
                *
                              101.37.192.211
                                                          11s
                                                80
```

這時您可以通過 http://foo.bar.com/foo 訪問到 http-svc1 服務;通過 http://foo.bar .com/bar 訪問到 http-svc2 服務;通過 http://foo.example.com/film 訪問到 httpsvc3 服務。

```
📕 说明 :
```

- 如果是生產環境,您需要將您的這個網域名稱指向上面返回的 ADDRESS 101.37.192.211 。
- 如果是測試環境測試,您可以修改 hosts 檔案添加一條網域名稱映射規則。

```
101.37.192.211 foo.bar.com
101.37.192.211 foo.example.com
```

#### 簡單路由預設網域名稱

如果您沒有網域名稱地址也沒有關係,Container Service為 Ingress 服務綁定了一個預設網域名稱,您可以通過這個網域名稱來訪問服務。網域名稱的格式如下:\*.[cluster-id].[region-id].alicontainer.com。您可以直接在控制台叢集基本資料頁擷取到該地址。

您可以通過下面的配置藉助該預設網域名稱暴露兩個服務。

```
root@master # cat <<EOF | kubectl create -f -
apiVersion: extensions/vlbetal
kind: Ingress
metadata:
    name: shared-dns
spec:
    rules:</pre>
```

```
- host: foo.[cluster-id].[region-id].alicontainer.com ##替換為您叢集
預設的服務訪問網域名稱
   http:
     paths:
      - path: /
       backend:
         serviceName: http-svcl
         servicePort: 80
  - host: bar.[cluster-id].[region-id].alicontainer.com ##替換為您叢集
預設的服務訪問網域名稱
   http:
     paths:
      - path: /
       backend:
         serviceName: http-svc2
         servicePort: 80
EOF
root@master # kubectl get ing
               HOSTS
NAME
                             ADDRESS
                                              PORTS
                                                        AGE
           foo.[cluster-id].[region-id].alicontainer.com,bar.[
shared-dns
cluster-id].[region-id].alicontainer.com
                                                    47.95.160.171
80
         40m
```

這時您可以通過 http://foo.[cluster-id].[region-id].alicontainer.com/ 訪問到 http-svc1 服務;通過 http://bar.[cluster-id].[region-id].alicontainer.com 訪 問到 http-svc2 服務。

#### 配置安全的路由服務

支援多認證管理,為您的服務提供安全防護。

1. 準備您的服務憑證。

如果沒有認證,可以通過下面的方法產生測試認證。



網域名稱與您的 Ingress 配置要一致。

root@master # openssl req -x509 -nodes -days 365 -newkey rsa:2048 keyout tls.key -out tls.crt -subj "/CN=foo.bar.com/O=foo.bar.com"

上面命令會產生一個認證檔案 tls.crt、一個私密金鑰檔案tls.key。

然後用該認證和私密金鑰建立一個名為*foo.bar*的 Kubernetes Secret。建立 Ingress 時需要 引用這個 Secret。

root@master # kubectl create secret tls foo.bar --key tls.key --cert
tls.crt

2. 建立一個安全的 Ingress 服務。

root@master # cat <<EOF | kubectl create -f apiVersion: extensions/vlbeta1</pre>

```
kind: Ingress
metadata:
 name: tls-fanout
spec:
  tls:
  - hosts:
    - foo.bar.com
    secretName: foo.bar
  rules:
   - host: foo.bar.com
    http:
      paths:
      - path: /foo
        backend:
          serviceName: http-svcl
          servicePort: 80
      - path: /bar
        backend:
          serviceName: http-svc2
          servicePort: 80
EOF
root@master # kubectl get ing
NAME
                HOSTS
                               ADDRESS
                                                 PORTS
                                                           AGE
tls-fanout
                 *
                               101.37.192.211
                                                            11s
                                                 80
```

 按照 基於網域名稱的簡單扇出路由 中的注意事項,配置 hosts 檔案或者設定網域名稱來訪問該 tls 服務。

您可以通過 http://foo.bar.com/foo 訪問到 http-svc1 服務;通過 http://foo.bar. com/bar 訪問到 http-svc2 服務。

您也可以通過 HTTP 的方式訪問該 HTTPS 的服務。Ingress 預設對配置了 HTTPS 的 HTTP 訪問重新導向到 HTTPS 上面。所以訪問 http://foo.bar.com/foo 會被自動重新導向到 https://foo.bar.com/foo。

### 通過 Kubernetes Dashboard 部署 Ingress

1. 將下面的 yml code 儲存到 nginx-ingress.yml 檔案中。

```
apiVersion: extensions/vlbetal
kind: Ingress
metadata:
   name: simple
spec:
   rules:
    - http:
        paths:
        - path: /svc
        backend:
        serviceName: http-svc
        servicePort: 80
```

- 登入Container Service#####,在 Kubernetes 菜單下,在叢集列表頁面中,單擊目的地組群右 側的控制台,進入 Kubernetes Dashboard 頁面。
- 3. 單擊建立,開始建立應用。



- 4. 單擊使用檔案建立。選擇剛才儲存的 nginx-ingress.yml 檔案。
- 5. 單擊上傳。

這樣就建立了一個 Ingress 的七層代理路由到 http-svc 服務上。

6. 在 Kubernetes Dashboard 上定位到 default 命名空間, 選擇訪問權。

可以看到您剛剛建立的 Ingress 資源及其訪問地址 http://118.178.174.161/svc。

<	三 服务发现与负载均衡 访问权		+	创建
集群	访问权			=
命名空间	_ 名称 ≑ 2	λD	已创建 🗢	
持久化存储卷	simple	₿.174.161 <sup>亿</sup>	2018-01-24 18:29:30	
角色	gateway	.113.231	2018-01-23 11:57:52	
存储类				
命名空间				
default *				
備党				
守护进程集				
部署				
任务				
容器组				
副本控制器				
有状态副本集				
服务发现与负载均衡				
访问权				

7. 開啟瀏覽器輸入該地址即可訪問前面建立的 http-svc 服務。

# 1.7.4 通過 Web 介面建立路由

阿里雲Container Service Web 介面整合了路由(Ingress)服務,您可通過 Web 介面快速建立路由 服務,構建靈活可靠的流量接入層。

前提條件

• 您已經成功建立一個 Kubernetes 叢集,參見##*Kubernetes*##,並且叢集中 Ingress controller 正常運行。 • SSH 登入到 Master 節點,參見SSH##Kubernetes##。

### 步驟1 建立 deployment 和服務

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的應用 > 部署,進入部署列表頁面。
- 3. 單擊頁面右上方使用模板建立。

容器服务 - Kubernetes ▼	部署列表		局新	使用镜像创建使用模板创建
概览	集群 k8s-cluster v 命名空间 defaul	t v		2
▼ 集群	名称    标签	容器组数量	创建时间	掘作
集群				
节点		① 没有者	İ询到符合条件的记录	
存储卷				
命名空间				
▼ 应用 部署				

4. 選擇所需的叢集和命名空間,選擇範例模板或自訂,然後單擊建立。

本例中,樣本中建立3個nginx應用,一個代表舊的應用old-nginx,一個代表新的應用 newnginx,此外建立一個domain-nginx應用,用於測試叢集訪問網域名稱。

集群	k8s-cluster	Ŧ
A.47.47.10		
中省全国	default	*
云侧横断		-
13 17 3 (Benck		-
横版	1 apiversion: extensions/v1beta1	法加部案
	2 kind: beployment	TONDERPES
	4 name: old-neinx	
	5 × spec:	
	6 replicas: 2	
	7 * selector:	
	8 - matchLabels:	
	11 - metadata:	
	12 - labels:	
	13 run: old-nginx	
	14 spec:	
	15 Containers:	
	17 image/ullpolicy: Always	
	18 name: old-nginx	
	19 ports:	
	20 - containerPort: 80	
	21 protocol: ICP	
	22 restartpointy: Always	
	24 apiVersion: extensions/v1beta1	
	25 kind: Deployment	
	26 - metadata:	
	27 name: new-nginx	
	20 Special pendicas: 1	
	30 v selector:	
	31 - matchLabels:	
	32 run: new-nginx	
	33 v template:	
	35 + t metaduala:	
	36 cun: new-nginx	*
		创建 2

old-nginx的編排模板如下所示:

```
apiVersion: extensions/vlbetal
kind: Deployment
metadata:
   name: old-nginx
spec:
```

```
replicas: 2
  selector:
    matchLabels:
     run: old-nginx
  template:
    metadata:
      labels:
        run: old-nginx
    spec:
      containers:
      - image: registry.cn-hangzhou.aliyuncs.com/xianlu/old-nginx
        imagePullPolicy: Always
        name: old-nginx
        ports:
        - containerPort: 80
          protocol: TCP
      restartPolicy: Always
_ _ _
apiVersion: v1
kind: Service
metadata:
 name: old-nginx
spec:
  ports:
  - port: 80
   protocol: TCP
    targetPort: 80
  selector:
   run: old-nginx
  sessionAffinity: None
  type: NodePort
```

new-nginx的編排模板如下所示:

```
apiVersion: extensions/vlbetal
kind: Deployment
metadata:
 name: new-nginx
spec:
  replicas: 1
  selector:
    matchLabels:
      run: new-nginx
  template:
    metadata:
      labels:
        run: new-nginx
    spec:
      containers:
      - image: registry.cn-hangzhou.aliyuncs.com/xianlu/new-nginx
        imagePullPolicy: Always
        name: new-nginx
        ports:
        - containerPort: 80
          protocol: TCP
      restartPolicy: Always
apiVersion: v1
kind: Service
metadata:
 name: new-nginx
```

```
spec:
  ports:
  - port: 80
    protocol: TCP
    targetPort: 80
  selector:
    run: new-nginx
  sessionAffinity: None
  type: NodePort
```

domain-nginx應用的編排模板如下所示:

```
apiVersion: apps/vlbeta2 # for versions before 1.8.0 use apps/
v1beta1
kind: Deployment
metadata:
  name: domain-nginx
  labels:
   app: nginx
spec:
  replicas: 2
  selector:
    matchLabels:
      app: nginx
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.7.9 # replace it with your exactly <</pre>
image_name:tags>
        ports:
        - containerPort: 80
___
apiVersion: v1
kind: Service
metadata:
 name: domain-nginx
spec:
 ports:
  - port: 80
   protocol: TCP
    targetPort: 80
  selector:
    app: nginx
  sessionAffinity: None
  type: NodePort
```

5. 單擊左側導覽列中的應用 > 服務 ,進入服務列表頁面。

等待服務建立完成後,在服務列表,您可看到本樣本建立的服務。

服务列表						刷新创建
集群 k8s-cluster	▼ 命名空间	default 🔻				
名称	类型	创建时间	集群IP	内部端点	外部端点	操作
domain-nginx	NodePort	2018-07-11 17:43:32	10.000	domain-nginx:80 TCP domain-nginx:32347 TCP		详情   更新   查看YAML   删除
kubernetes	ClusterIP	2018-07-11 17:35:35		kubernetes:443 TCP	-	详情   更新   查看YAML   删除
new-nginx	NodePort	2018-07-11 17:37:01		new-nginx:80 TCP new-nginx:32637 TCP		详情   更新   查看YAML   删除
old-nginx	NodePort	2018-07-11 17:37:01	10.000	old-nginx:80 TCP old-nginx:32039 TCP	-	洋情   更新   查看YAML   删除

### 步驟2 建立路由

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的應用 > 路由,進入路由頁面。
- 3. 選擇所需的叢集和命名空間,單擊頁面右上方的建立。

容器服务 - Kubernetes ▼	路由 ( Ingress )				刷新创建
概览	集群 k8s-cluster v 命名空	剧 default 🔻 3			4
▼ 集群	名称	這点	规则	创建时间	操作
集群					
节点			<li>① 没有查询到符合我</li>	条件的记录	
存储卷					
命名空间					
▼ 应用 1					
部署					
容器组					
服务					
路由 2					

4. 在彈出的路由建立對話方塊中,首先配置路由名稱,本例為 nginx-ingress。

名称:	nginx-ingress	

5. 對路由規則進行配置。

路由規則是指授權入站到達叢集服務的規則,支援 http/https 規則,配置項包括網域名稱(虛擬機器主機名稱)、URL 路徑、服務名稱、連接埠配置和路由權重等。詳細的資訊請參見#######。 本例中配置添加一條複雜的路由規則,配置叢集預設的測試網域名稱和虛擬機器主機名稱,展示 基於網域名稱的路由服務。

则: 💿 添加			
域名			
test.c	5.cn-hangzhou	u.alicontainer.com	
使用 *.c0	.cn-hangzho	ou.alicontainer.com 或者 自治	定义
路径			
/			
服务 🕙 添加			
名称	端口	权重 权重比例	
		100.00	
domain-nginx	• 80 •	100 100.0%	
域名			
foo.bar.com			
使用 *.c(	5.cn-hangzho	ou.alicontainer.com 或者 自治	定义
路径			
/			
服务 😋 添加			
名称	端口	权重 权重比例	
	-	50.0%	_
new-nginx	* 80 *	50 50.0%	Ŭ
old-painy	¥ 80 ¥	50 50.0%	
Old-Hyllix	* 00 *	50 50.0%	-

- 基於預設網域名稱的簡單路由,即使用叢集的預設網域名稱對外提供訪問服務。
  - 網域名稱配置:使用叢集的預設網域名稱,本例中是 test.[cluster-id].[regionid].alicontainer.com。

在建立路由對話方塊中,會顯示該叢集的預設網域名稱,網域名稱格式是 \*.[clusterid].[region-id].alicontainer.com; 您也可在叢集的基本資料頁面中擷取。

- 服務配置:佈建服務的訪問路徑、名稱以及連接埠。
  - 訪問路徑配置:您可指定服務訪問的 URL 路徑,預設為根路徑/,本例中不做配置。每個路徑(path)都關聯一個 backend(服務),在阿里雲 SLB 將流量轉寄到 backend之前,所有的入站請求都要先匹配網域名稱和路徑。
  - 服務配置:支援服務名稱、連接埠、服務權重等配置,即 backend 配置。同一個訪問路徑下,支援多個服務的配置,Ingress的流量會被切分,並被轉寄到它所匹配的backend。

- 基於網域名稱的簡單扇出路由。本例中使用一個虛擬主機名稱作為測試網域名稱對外提供訪問服務,為兩個服務配置路由權重,並為其中一個服務設定灰階發布規則。若您在生產環境中,可使用成功備案的網域名稱提供訪問服務。
  - 網域名稱配置:本例中使用測試網域名稱 foo.bar.com。

您需要修改 hosts 檔案添加一條網域名稱映射規則。

118.178.108.143 foo.bar.com #IP即是Ingress的Address

- 服務配置:佈建服務的訪問路徑、服務名稱、服務連接埠和服務權重。

- 訪問路徑配置:指定服務訪問的 URL 路徑。本例中不做配置,保留根路徑/。
- 服務名稱:本例中設定新舊兩個服務 nginx-new 和 nginx-old。
- 服務連接埠:暴露80連接埠。
- 權重設定:設定該路徑下多個服務的權重。服務權重採用相對值計算方式,預設值為 100,如本例中所示,新舊兩個版本的服務權重值都是50,則表示兩個服務的權重比例 都是50%。
- 6. 配置灰階發布。

目前阿里雲Container ServiceKubernetes Ingress Controller需要0.12.0-5及其以上版本才支援流量切分特性。

Container Service支援多種流量切分方式,適用於灰階發布以及AB測試情境。

- 1. 基於Request Header的流量切分
- 2. 基於Cookie的流量切分
- 3. 基於Query Param的流量切分

設定灰階規則後,要求標頭中滿足灰階發布匹配規則的請求才能被路由到新版本服務new-nginx 中。如果該服務設定了100%以下的權重比例,滿足灰階規則的請求會繼續依據權重比例路由到 對應服務。

在本例中,設定Header要求標頭帶有foo=^bar\$的灰階發布規則,僅帶有該要求標頭的用戶端 請求才能訪問到new-nginx 服務。

灰度发布:	◆ 添加 设置灰度规则后,容 请求将会继续依据权重路由到	好合规则的请求将路B 削新老版本服务中	由到新服务中。如果谈
	服务	类型	名称 匹
	new-nginx 🔻	Header <b>•</b>	foo

- 服務:路由規則配置的服務。
- 類型:支援Header(要求標頭)、Cookie和Query(請求參數)的匹配規則。
- 名稱和匹配值:使用者自訂的請求欄位,名稱和匹配值為索引值對。
- 匹配規則:支援正則匹配和完全符合。
- 7. 配置註解。

單擊重新導向註解,可為路由添加一條典型的重新導向註解。即 nginx.ingress. kubernetes.io/rewrite-target:/,表示將/path 路徑重新導向到後端服務能夠識別的 根路徑/上面。

📕 说明:

本例中未對服務配置訪問路徑,因此不需要配置重新導向註解。重新導向註解的作用是使 Ingress以根路徑轉寄到後端,避免訪問路徑錯誤配置而導致的404錯誤。

您也可單擊添加按鈕,輸入註解名稱和值,即Ingress的annotation索引值對,Ingress的註解參 見*https://kubernetes.github.io/ingress-nginx/user-guide/nginx-configuration/annotations/*。

名称	值	
nginx.ingress.kubernetes.io/	/rewrite-target /	•

## 

### • 您可選擇使用已有密鑰。

TLS:	☑ 开启 💿 已有密钥 ◎ 新建密钥	
	foo.bar	•

1. 登入 master 節點, 建立 tls.key 和 tls.crt。

```
openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout tls
.key -out tls.crt -subj "/CN=foo.bar.com/O=foo.bar.com"
```

2. 建立一個 secret。

kubectl create secret tls foo.bar --key tls.key --cert tls.crt

- **3.** 執行命令 kubectl get secret,您可看到該 secret 已經成功建立。在 Web 介面可選 擇建立的*foo.bar*這個 secret。
- 您可選擇在 TLS 介面上利用已建立的 TLS 私密金鑰和認證,一鍵建立 secret。



**1.** 登入 master 節點, 建立 tls.key 和 tls.crt。

openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout tls
.key -out tls.crt -subj "/CN=foo.bar.com/O=foo.bar.com"

- 2. 執行 vim tls.key 和 vim tls.crt 擷取產生的私密金鑰和認證。
- 3. 將認證和私密金鑰的內容複寫到 TLS 建立密鑰的面板。
- 9. 添加標籤。

標籤的作用是為Ingress添加對應的標籤,標示該Ingress的特點。

你盘	● 添加		
	名称	值	
	node-role.kubernetes.io/ingress	true	•

10.最後單擊建立,返迴路由列表。

等待一段時間,可以看到一條路由。

路由 (Ingress)			刷新创建
集群 k8s-cluster ▼ 命名3	21회 default ·		
名称 端点	规则	创建时间	操作
nginx-ingress	test.cf 5.cn-hangzhou.alicontainer> domain-nginx foo.bar.com/ -> new-nginx foo.bar.com/ -> old-nginx	2018-07-11 17:49:46 详情	変更   查看YAML   删除

**11.**單擊路由中的訪問網域名稱 test.[cluster-id].[region-id].alicontainer.com,以

及 foo.bar.com,可訪問 nginx 的歡迎頁面。

Test.co.co.a.a.a.a.a.a.a.a.a.a.a.a.a.a.a.a.a						
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.						

單擊指向new-nginx服務的路由地址,發現指向了old-nginx應用的頁面。



在瀏覽器中訪問路由地址,預設情況下,要求標頭(Header)中沒有前面步驟中定義的foo=^ bar\$,因此流量會導向old-nginx應用。

```
    ← → C (③ foo.bar.com/?spm=5176.2020520152.0.0.509c61b1iW1N16
    old
```

12.SSH登入到Master節點,執行以下命令,類比帶有特定要求標頭的訪問結果。

```
curl -H "Host: foo.bar.com" http://47.107.20.35
old
curl -H "Host: foo.bar.com" http://47.107.20.35
old
curl -H "Host: foo.bar.com" http://47.107.20.35
#類似於瀏覽器的訪問請求
old
curl -H "Host: foo.bar.com" -H "foo: bar" http://47.107.20.35
#類比帶有特有header的訪問請求, 會根據路由權重返回結果
new
curl -H "Host: foo.bar.com" -H "foo: bar" http://47.107.20.35
old
curl -H "Host: foo.bar.com" -H "foo: bar" http://47.107.20.35
old
curl -H "Host: foo.bar.com" -H "foo: bar" http://47.107.20.35
old
curl -H "Host: foo.bar.com" -H "foo: bar" http://47.107.20.35
```

new

# 1.7.5 變更路由

前提条件

- 您已經成功建立一個 Kubernetes 叢集,參見##Kubernetes##,並且叢集中 Ingress controller 正常運行。
- 您已經成功建立一個路由,參見## Web ######。

操作步骤

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的應用 > 路由,進入路由頁面。
- 3. 選擇所需的叢集和命名空間,選擇所需的路由,然後單擊路由右側的變更。

容器服务	路由 (Ingress)		创建    刷新
Kubernetes Swarm			
概览	来時 test-kbs V 即石立向 default V		
▼ 毎群	名称	创建时间	操作
集群	test.c         http-svc1           nginx-ingress         foo.bar.com/svc2         > http-svc2           foo.bar.com/svc3         > http-svc3	2018-04-28 11:38:05	洋情   変更   删除
节点			4
存储			
部署			
容器组			
服务			
<b>隆曲</b> 2			

4. 在彈出的對話方塊中,對路由的相關參數進行變更,然後單擊確定。本例中將 test.[ cluster-id].[region-id].alicontainer.com 修改為 testv2.[cluster-id].[ region-id].alicontainer.com。



### 后续操作

返迴路由列表,您可看到該路由的其中一條路由規則發生變化。

路由 ( Ingress )				创建	刷新
集群 test-k8s ▼	命名空间	default 🔹			
名称 端;	jū.	规则	创建时间		操作
nginx-ingress		testv2[c1 http://www.alicontaine> http://wwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwwww	2018-04-28 11:38:05	详情   变更	删除

# 1.7.6 查看路由

前提条件

- 您已經成功建立一個 Kubernetes 叢集,參見##Kubernetes##,並且叢集中 Ingress controller 正常運行。
- 您已經成功建立一個路由,參見## Web ######。

操作步骤

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的 應用 > 路由,進入路由頁面。
- 3. 選擇所需的叢集和命名空間,選擇所需的路由,然後單擊路由右側的詳情。

容器服务	)路由(Ingress)		创建刷新
Kubernetes Swarm			
概览	集群 test-k8s v 命名空间 default v 3		
▼ 集群	名称 就点 规则	创建时间	撮作
集群	test.c >> http-svc1 nginx-ingress > foo.bar.com/svc2 -> http-svc2 foo.bar.com/svc3 -> http-svc3	2018-04-28 11:38:05	洋情   変更   删除
节点			4
存储			
▼应用 1			
部署			
容器组			
服务			
路由 2			

nginx-ingress   t 返回列表				刷新
基本信息				
名称:	nginx-ingress			
命名空间;	default			
创建时间:	2018-04-28T03:38:05Z			
标签:				
注解:	nginx.ingress.kubernetes.io/rewrite-target:/			
端点:	13			
规则				
域名		路径	服务名称	服务端口
test.c1	1 cn-hangzhou.alicontainer.com	/svc1	http-svc1	80
foo.bar.com		/svc2	http-svc2	80
foo.bar.com		/svc3	http-svc3	80

## 進入路由詳情頁面,您可查看該路由的基本資料以及路由規則。

# 1.7.7 刪除路由

前提条件

- 您已經成功建立一個 Kubernetes 叢集,參見##*Kubernetes*##,並且叢集中 Ingress controller 正常運行。
- 您已經成功建立一個路由,參見## Web ######。

### 操作步骤

1. 登入Container Service#####。

- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的應用 > 路由,進入路由頁面。
- 3. 選擇所需的叢集和命名空間,選擇所需的路由,然後單擊路由右側的刪除。

容器服务		路由 ( Ingress )						创建	刷新
Kubernetes	Swarm								
概览		集群 test-k8s ▼ 命	诸空间 default	•	3				
▼ 集群		名称	规则				创建时间		操作
集群		nginx-ingress	test.c foo.bar.com/svc2 foo.bar.com/svc3	-> http-svc2 -> http-svc3	n-hangzhou.alicontainer>	http-svc1	2018-04-28 11:38:05	洋情   変更	删除
节点									4
存储									
▼应用	1								
部署									
容器组									
服务									
路由	2								

4. 在彈出的對話方塊中,單擊確定,完成刪除。

确认		$\times$
0	删除nginx-ingress ?	
	确定取	消

# 1.8 配置項及密鑰管理

# 1.8.1 建立配置項

在Container Service管理主控台上,您可以通過配置項菜單或使用模板來建立配置項。

## 通過配置項建立

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的應用 > 配置項,進入設定檔列表。
- 3. 在設定檔列表頁面,選擇需要建立配置項的叢集,然後單擊建立配置項。

容器服务	配置文件列表			刷新创建配置项
Kubernetes Swarm	集計· k8s_cluster y 3			
▼ 集群				_
集群	配置文件名称	命名空间	创建时间	操作
节点	test-mariadb	default	2018-04-23 14:12:26	删除   修改
存储	cluster-info	kube-public	2018-04-23 11:48:31	删除   修改
▼ 应用 1	extension-apiserver-authentication	kube-system	2018-04-23 11:48:29	删除   修改
部署	ingress-controller-leader-nginx	kube-system	2018-04-23 11:58:42	删除   修改
服务	kube-flannel-cfg	kube-system	2018-04-23 11:48:39	删除   修改
发布	kube-proxy-master	kube-system	2018-04-23 11:48:52	删除   修改
配置项 2	kube-proxy-worker	kube-system	2018-04-23 11:48:52	删除   修改

- 4. 填寫設定檔的資訊並單擊確定。
  - 命名空間:選擇該配置項所屬的命名空間。配置項(ConfigMap)是 kubernetes 資來源物件,需要作用於命名空間。
  - 設定檔名:指定配置項的檔案名稱,名稱可以包含小寫字母、數字、連字號(-)或者點號
     (.),名稱不可為空。其他資來源物件需要引用設定檔名來擷取配置資訊。
  - 配置項:填寫變數名稱和變數值後,需要單擊右側的添加。您也可以單擊編輯設定檔 在彈出的 對話方塊裡編寫配置項並單擊確定。

配置文件				
*命名空间:	default			٣
*配置文件名	test-config			
44. •	标: 名称只能包含小写字母数字,'-或者'.',名称不能为空。			
配置项:	变量名称	变量值		操作
	enemies	aliens	编辑	删除
	lives	3	编辑	删除
	名称	值		添加
	变量名不能重复 , 变量名和	变量值不能为空。		
	编辑配置文件			
		<b>确</b> 定 取消		

本樣本中設定了 enemies 和 lives 變數,分別用於傳遞 aliens 和 3 這兩個參數。

YAML格式的配置文件
1 - plata: 2 enemies: aliens 3 lives: '3' 4 metadata: 5 name: test-config 6 namespace: default 7 Namespace: default 8 BT董文件必须符合YAML格式
确定取消

## 5. 單擊確定後,您可以在設定檔列表中看到 test-config 設定檔。

配置文件列表			刷新创建配置项
集群: k8s-test ▼			
配置文件名称	命名空间	创建时间	操作
test-config	default	2018-02-05 11:03:42	删除   修改

### 使用模板建立

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的應用 > 部署,進入部署列表。
- 3. 單擊頁面右上方的使用模板建立。

容器服务	部署列表	使用编编创建使用模板创建 刷新
Kubernetes Swarm		
▼ 集群	集群 k8s-duster v 命名空间 default v	•
集群	名称 标签 容識組数量 创建时间	操作
节点	① 没有查询到符合条件的记录	
存储		
▼ 应用 部署 1		
服务		
发布		
配置项		

- 4. 在使用模版部署的頁面,設定設定檔的資訊並單擊部署。
  - 叢集:選擇需要建立配置項的叢集。
  - 命名空間:選擇該配置項所屬的命名空間。配置項(ConfigMap)是 kubernetes 資來源物件,需要作用於命名空間。
  - 樣本模板:您可以選擇自訂,根據 Kubernetes yaml 文法規則編寫 ConfigMap 檔案,或者選 擇樣本模板 resource-ConfigMap。該樣本模板的 ConfigMap 名稱為 aliyun-config,包含兩 個變數檔案 game.properties 和 ui.properties,您可以在此基礎上進行修改,然後單 擊部署。



5. 部署成功後,您可以在設定檔列表下看到 aliyun-config 設定檔。

配置文件列表			刷新创建配置项
集群: k8s-cluster ▼			
配置文件名称	命名空间	创建时间	操作
aliyun-config	default	2018-04-23 15:13:27	删除   修改

# 1.8.2 在 pod 中使用配置項

您可以在 Pod 中使用配置項,有多種使用情境,主要包括:

- 使用配置項定義 pod 環境變數
- 通過配置項設定命令列參數
- 在資料卷中使用配置項

更多關於配置項的資訊,可以參見 Configure a Pod to Use a ConfigMap。

### 使用限制

您在 pod 裡使用配置項時,需要兩者處於同一叢集和命名空間中。

### 建立配置項

本樣本建立配置項 special\_config, 包含 SPECIAL\_LEVEL: very 和 SPECIAL\_TYPE: charm 兩個索引值對。

### 使用編排模板建立配置項

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的應用 > 部署,然後單擊右上方的使用模板建立。
- 3. 選擇所需的叢集和命名空間,選擇範例模板或自訂,然後單擊建立。

您可以使用如下 yaml 樣本模板建立配置項。

```
apiVersion: v1
kind: ConfigMap
metadata:
    name: special-config
    namespace: default
data:
    SPECIAL_LEVEL: very
    SPECIAL_TYPE: charm
```

### 通過 Web 介面建立配置項

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的應用 > 配置項,進入配置項列表頁面。
- 3. 選擇所需的叢集和命名空間,然後單擊右上方的使用模板建立。

4. 輸入配置項名稱,然後單擊添加,輸入配置項,最後單擊確定。

容器服务		配置项					
Kubernetes	Swarm						
概览			集群	c152b4c3a188140ec8af63b34c23b	14a3		
▼ 集群			命名空间	default			
集群			*配置项名称:	special-config			
共占				名称只能包含小写字母数字,'-'或者	著.',名称不能为空。		
13 AR			配置项:	变量名称	变量值		操作
存储				SPECIAL_LEVEL	very	编辑	删除
▼ 应用					cham	(##2	1 20/84
部署				SPECIAL_TTPE	dam	에 귀리.	AUSR-F
容器组				名称	值		添加
				变量名不能重复,变量名和变量值7	不能为空。		
服务				编辑配置文件			
路由							
发布					确定 取消		
配置项							

#### 使用配置項定義 pod 環境變數

使用配置項的資料定義 pod 環境變數

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的應用 > 部署,然後單擊右上方的使用模板建立。
- 3. 選擇所需的叢集和命名空間,選擇範例模板或自訂,然後單擊建立。

您可以在 pod 中定義環境變數,使用 valueFrom 引用 SPECIAL\_LEVEL 的 value 值,從而定 義 pod 的環境變數。

下面是一個編排樣本。

```
apiVersion: v1
kind: Pod
metadata:
  name: config-pod-1
spec:
   containers:
     - name: test-container
      image: busybox
      command: [ "/bin/sh", "-c", "env" ]
      env:
         - name: SPECIAL_LEVEL_KEY
          valueFrom:
                                                 ##使用valueFrom來指
定env引用配置項的value值
            configMapKeyRef:
              name: special-config
                                                 ##引用的設定檔名稱
              key: SPECIAL_LEVEL
                                                 ##引用的配置項key
  restartPolicy: Never
```

同理,如果您需要將多個配置項的 value 值定義為 pod 的環境變數值,您只需要在 pod 中添加多個 env 參數即可。

### 將配置項的所有 key/values 配置為 pod 的環境變數

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的應用 > 部署,然後單擊右上方的使用模板建立。
- 3. 選擇所需的叢集和命名空間,選擇範例模板或自訂,然後單擊建立。

如果您想在一個 pod 中將配置項的所有 key/values 索引值對配置為 pod 的環境變數,可以使用 envFrom 參數,配置項中的 key 會成為 Pod 中的環境變數名稱。

下面是一個編排樣本。

```
apiVersion: v1
kind: Pod
metadata:
    name: config-pod-2
spec:
    containers:
        - name: test-container
        image: busybox
        command: [ "/bin/sh", "-c", "env" ]
        envFrom: ##引用 sepcial-config 設定檔的所有 key/
values 索引值對
        - configMapRef:
            name: special-config
restartPolicy: Never
```

通過配置項設定命令列參數

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的應用 > 部署,然後單擊右上方的使用模板建立。
- 3. 選擇所需的叢集和命名空間,選擇範例模板或自訂,然後單擊建立。

您可以使用配置項設定容器中的命令或者參數值,使用環境變數替換文法 \$(VAR\_NAME)來進行。

下面是一個編排樣本。

```
apiVersion: v1
kind: Pod
metadata:
  name: config-pod-3
spec:
   containers:
      - name: test-container
       image: busybox
       command: [ "/bin/sh", "-c", "echo $(SPECIAL_LEVEL_KEY) $(
SPECIAL TYPE KEY)" ]
       env:
         - name: SPECIAL_LEVEL_KEY
           valueFrom:
             configMapKeyRef:
               name: special-config
               key: SPECIAL_LEVEL
```

```
- name: SPECIAL_TYPE_KEY
    valueFrom:
    configMapKeyRef:
    name: special-config
    key: SPECIAL_TYPE
restartPolicy: Never
```

運行這個 pod 後,會輸出如下結果。

very charm

### 在資料卷中使用配置項

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的應用 部署,然後單擊右上方的使用模板建立。
- 3. 選擇所需的叢集和命名空間,選擇範例模板或自訂,然後單擊建立。

您也可以在資料卷裡面使用配置項,在 volumes 下指定配置項名稱,會將 key/values 的資料存 放區到 mountPath 路徑下(本例中是 /*etc/config*)。最終產生以 key 為檔案名稱,values 為檔案內容的設定檔。

下面是一個編排樣本。

```
apiVersion: v1
kind: Pod
metadata:
  name: config-pod-4
spec:
   containers:
     - name: test-container
       image: busybox
      command: [ "/bin/sh", "-c", "ls /etc/config/" ] ##列出該目錄
下的檔案名稱
      volumeMounts:
       - name: config-volume
        mountPath: /etc/config
  volumes:
     - name: config-volume
       configMap:
        name: special-config
  restartPolicy: Never
```

運行 pod 後,會輸出配置項的 key。

SPECIAL\_TYPE SPECIAL\_LEVEL

# 1.8.3 修改配置項

您可以修改配置項的配置。



修改設定檔會影響使用該設定檔的應用。

## 通過配置項進行修改

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的應用 > 配置項,進入設定檔列表。
- 3. 選擇所需的叢集和命名空間,選擇需要修改的配置項並單擊右側的修改。

容器服务	配置项列表			刷新创建配置项
Kubernetes Swarm				
▼ 集群	集群 k8s-cluster V 命名空间 default V	3		
集群	配置项名称	命名空间	创建时间	操作
节点	aliyun-config	default	2018-04-23 15:13:27	删除   4
存储	test-config	default	2018-04-23 15:25:33	删除   修改
	test-mariadb	default	2018-04-23 14:12:26	删除   修改
響暗				
容器组				
服务				
发布				
配置项 2				

4. 在彈出的確認對話方塊中,單擊確定。

修改确认		$\times$
0	修改配置文件会影响使用该配置文件的应用,您确定要修改吗?	
	确定	取消

- 5. 修改配置項。
  - 選擇需要修改的配置項並單擊右側的編輯,修改配置後單擊儲存。
  - 或者單擊編輯設定檔,完成編輯後單擊確定。

集群	real landstreak		
命名空间	default		
*配置项名称:	test-config		
配置项:	变量名称	变量值	操作
	enemies	aliens	编辑   删除
	lives	3	编辑   删除
	名称	值	添加
	变量名不能重复,变量名和变量值	不能为空。	
[	编辑配置文件		
		确定取消	

6. 完成配置項修改後,單擊確定。

## 通過 Kubernetes Dashboard 進行修改

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的叢集,進入叢集列表。
- 3. 選擇所需的叢集,並單擊右側的控制台。

容器服务		集群列表				您最多可以创建 5 个集器	洋,每个集群最多可以添加 20 个节点	刷新	创建 Kubernetes 集群 🛛 👻
Kubernetes	Swarm	常见问题: & 如何创建集群 & 扩容和缩容集	并 <i>8</i> 通过	t kubectl 连接 Kubernetes 集	群 🔗 通过命令管	理应用			
集群		名称 ▼							
节点		集群名称/ID	地域	网络类型	集群状态	创建时间	Kubernetes 版本		2 🗉
存储		kBs-cluster	华东1	虚拟专有网络 vpc-bp13cwaie5z	●运行中	2018-04-20 11:02:31	1.9.3	1	管理   查看日志   控制台 集群伸缩   更多▼

 在 Kubernetes Dashboard 頁面,單擊左側導覽列中的配置與儲存 > 配置字典,選擇所需的配置 字典,單擊右側的操作 > 查看/編輯 YAML 。

	持久化存储卷	■ 配置与存储 配置字典
	角色	
	存储类	配置字典
	命名空间	名称 🜲
	default 👻	test-config
	概况	alivun-config
	工作负载	
	定时任务	test-mariadb
	守护进程集	
	部署	
	任务	
	容器组	
	副本集	
	副本控制器	
	有状态副本集	
	服务发现与负载均衡	
	访问权	
文档版本	服务	
	配置与存储	

5. 彈出修改配置字典對話方塊,對組態變數進行修改後,單擊更新。



# 1.8.4 刪除配置項

您可以刪除不再使用的配置項。

通過配置項菜單進行刪除

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的應用 > 配置項,進入設定檔列表。
- 3. 選擇目的地組群,選擇需要修改的配置項並單擊右側的刪除。

容器服务	配置文件列表			剧新创建配置项
Kubernetes Swarm	集群: k8s-cluster 🔻 3			
* 葉群 集群	配置文件名称	命名空间	创建时间	操作
节点	aliyun-config	default	2018-04-23 15:13:27	删除   修改
存储	test-config	default	2018-04-23 15:25:33	4 删除 修改
▼应用 1	test-mariadb	default	2018-04-23 14:12:26	删除   修改
部署	cluster-info	kube-public	2018-04-23 11:48:31	删除   修改
服务	extension-apiserver-authentication	kube-system	2018-04-23 11:48:29	删除   修改
发布	ingress-controller-leader-nginx	kube-system	2018-04-23 11:58:42	删除   修改
配置项 2 =	kube-flannel-cfg	kube-system	2018-04-23 11:48:39	創除   修改

## 通過 Kubernetes Dashboard 進行刪除

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的叢集,進入 Kubernetes 叢集列表頁面。
- 3. 單擊左側導覽列中的叢集,選擇目的地組群,單擊右側的控制台。

容器服务		集群列表				您最多可以创建 5 个集群	, 每个集群最多可以添加 20 个节点	刷新	创建Kubernetes 集群 🛛 👻
Kubernetes	Swarm								
▼ 集群		常见问题: ② 如何创建集群 ③ 扩容和缩容集制	洋 🔗 通过 kubec	tl 连接 Kubernetes 集課	詳 🔗 通过命令管	理应用			
集群	1	名称 ▼							
节点		集群名称/ID	地域 网络	送型	集群状态	创建时间	Kubernetes 版本		2 🗉
存储		k8s-cluster	华东1    虚拟 vpc-	专有网络 bp13cwaie5z	●运行中	2018-04-20 11:02:31	1.9.3	管理	童石日志   控制台     集群伸缩   更多▼

 在 Kubernetes Dashboard 頁面,選擇所需的命名空間,單擊左側導覽列中的配置與儲存 > 配置 字典,單擊右側的的操作按鈕,在下拉框中單擊刪除。

节点	☴ 配置与存储 配置字典	十 创建
持久化存储卷		
角色	配置字典	=
存储类	68: ↓         ਓੱੱ	已创建 🗢
命名空间	test-config .	2018 15:2 删除 <b>3</b>
default 🝷 🚺	allyun-config .	2018 15:1 查看/编辑 YAML
成兄	test-mariado app: test-maria chart: mariado-0 heritage: Till release: t	2018-04-23 14:12:26
上作页载 中时/1条		
守护进程集		
戦争		
任务		
容器组		
副本焦		
副本控制器		
有状态副本集		
服务发现与负载均衡		
访问权		
服务		
配置与存储 配置字类		

5. 在彈出的對話方塊中,單擊刪除。

# 1.8.5 建立密鑰

## 前提条件

您已成功建立一個 Kubernetes 叢集,參見##Kubernetes##。

## 背景信息

若您需要在 Kubernetes 叢集中使用一些敏感的配置,比如密碼、認證等資訊時,建議使用密鑰(secret),即保密字典。

密鑰有多種類型,例如:

- Service Account:用來訪問 Kubernetes API,由 Kubernetes 自動建立,並且會自動掛載到 Pod 的/run/secrets/kubernetes.io/serviceaccount目錄中。
- Opaque: base64 編碼格式的 Secret,用來儲存密碼、認證等敏感資訊。

預設情況下,通過 Web 介面只能建立 Opaque 類型的密鑰。Opaque 類型的資料是一個 map 類型,要求value 是 base64 編碼格式。阿里雲Container Service提供一鍵建立密鑰的功能,自動將明 文資料編碼為 base64 格式。

您也可通過命令列手動建立密鑰,請參見 kubernetes secret 瞭解更多資訊。

### 操作步骤

- 1. 登入 Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的應用 > 保密字典,進入保密字典頁面。
- 3. 選擇所需的叢集和命名空間, 單擊右上方的建立。

容器服务	保密字典				刷新创建
Kubernetes Swarm					
概览	集群 k8s-cluster V 即名空间 de				
▼ 集群	名称 当	类型	命名空间	创建时间	操作
集群	default-token-46hv4 k	kubernetes.io/service-account-token	default	2018-05-18 17:22:31	洋情   编辑   删除
节点					
存储					
命名空间					
▼应用 1					
部署					
容器组					
服务					
路由					
发布					
配置项					
保密字典 2					

4. 配置新的保密字典。



命名空间 * 名称	default	
	名称长度为 1-253 字符 , 只能包含小写	字母、数字、'-和'.
* 数据	3 名称	值
	<ul> <li>password</li> </ul>	lf2dle2e67df
	lusername	admin
	名称只能包含数字、字母、"_″、"-"和" 对数据值进行 base64 编码	N.
		3 确定 取消

- 1. 名稱:輸入密鑰的名稱。名稱長度為 1-253 字元,只能包含小寫字母、數字、'-'和'.'
- 2. 配置密鑰的資料。單擊添加,在彈出的對話方塊中,輸入密鑰名稱和值,即索引值對。本例 中建立的密鑰包含兩個 value, username:admin和 passwrod:1f2d1e2e67df。
- 3. 單擊確定。
- 5. 預設返回保密字典頁面,您可看到建立的密鑰出現在列表中。

保密字典				局	判新	创建
集群 k8s-cluster ▼ 命名空间	default 🔻					
名称	类型	命名空间	创建时间			操作
account	Opaque	default	2018-05-18 18:18:20	详情	编辑	删除
default-token-46hv4	kubernetes.io/service-account-token	default	2018-05-18 17:22:31	详情	编辑	删除

# 1.8.6 查看密鑰

您可通過阿里雲Container Service Web 介面查看已建立的密鑰。

前提条件

- 您已成功建立一個 Kubernetes 叢集,參見##Kubernetes##。
- 您已建立一個密鑰,參見####。

## 操作步骤

- 1. 登入 Container Service######。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的應用 > 保密字典,進入保密字典頁面。
- 3. 選擇所需的叢集和命名空間,選擇所需的密鑰,並單擊右側的詳情。

容器服务	保密字典				刷新创建
Kubernetes Swarm	集群 test_kRs ▼ 命名空间	default 🔹			
概览					
▼ 集群	名称	美型	命名空间	创建时间	4 操作
集群	account	Opaque	default	2018-05-31 14:03:50	洋情 編輯 删除
节点	default-token-drpgj	kubernetes.io/service-account-token	default	2018-05-28 17:09:06	洋情   编辑   删除
存储					
命名空间					
▼应用 1					
部署					
容器组					
服务					
路由					
发布					
配置项					
保密字典 2					

4. 進入密鑰的詳情頁面,您可查看該密鑰的基本資料,以及密鑰包含的資料資訊。

## 單擊詳細資料中資料名稱右側的表徵圖,可查看資料的明文。

保密字典 - account		刷新编辑
基本信息		
名称: account		命名空间: default
创建时间: 2018-05-31 14:03:50		类型: Opaque
详细信息		
password		
username	•	

# 1.8.7 編輯密鑰

您可通過阿里雲Container Service Web 介面直接編輯已有密鑰。

前提条件

- 您已成功建立一個 Kubernetes 叢集,參見##Kubernetes##。
- 您已建立一個密鑰,參見####。

## 操作步骤

1. 登入 Container Service#####。

- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的應用 > 保密字典,進入保密字典頁面。
- 3. 選擇所需的叢集和命名空間,選擇所需的密鑰,並單擊右側的編輯。

容器服务	保密字典					刷新	创建
Kubernetes Swarm							
概览	集群 test-k8s v 命名空间	default 🔻 3					
▼ 集群	名称	类型	命名空间	创建时间		4	操作
集群	account	Opaque	default	2018-05-31 14:03:50	冸	青   编辑	删除
节点	default-token-drpgj	kubernetes.io/service-account-token	default	2018-05-28 17:09:06	洋	青   编辑	一删除
存储							
命名空间							
▼ 应用 1							
部署							
容器组							
服务	E						
路由							
发布							
配置项							
保密字典 2							

4. 在編輯密鑰的頁面,可對密鑰的資料進行編輯。

编辑保密字典	
命名空间	default
* 名称	account
* 数据	<ul> <li>各称</li> </ul>
	e username
	password
	名称只能包含数字、字母、"_"、"-"和"."

5. 最後單擊確定。

# 1.8.8 刪除密鑰

您可通過阿里雲Container Service Web 介面刪除已有密鑰。

## 前提条件

- 您已成功建立一個 Kubernetes 叢集,參見##Kubernetes##。
- 您已建立一個密鑰,參見####。

### 背景信息



叢集建立過程中產生的密鑰請勿刪除。

## 操作步骤

- 1. 登入 Container Service######。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的應用 > 保密字典,進入保密字典頁面。
- 3. 選擇所需的叢集和命名空間,選擇所需的密鑰,並單擊右側的刪除。

容器服务	保密字典				刷新创建
Kubernetes Swarm					
概览	集群 test-k8s ▼ 命名空间	default 🔹 3			
▼ 集群	名称	类型	命名空间	创建时间	4
集群	account	Opaque	default	2018-05-31 14:03:50	洋情 编辑 删除
节点	default-token-drpgj	kubernetes.io/service-account-token	default	2018-05-28 17:09:06	洋情   编辑   删除
存储					
命名空间					
▼ 应用 1					
部署					
容器组					
服务					
路由					
发布					
配置项					
保密字典 2					

4. 在彈出的對話方塊中,單擊確定,完成刪除。

提示		×
0	确定要删除 account 吗?	
	确定	取消

# 1.9 儲存管理

# 1.9.1 概述

Container Service支援 kubernetes Pod 自動綁定阿里雲雲端硬碟、NAS、 OSS 儲存服務。

目前,支援靜態儲存卷和動態儲存裝置卷。每種資料卷的支援情況如下:

阿里雲儲存	待用資料卷	動態資料卷
阿里雲雲端硬碟	您可以通過以下兩種方式使用 雲端硬碟靜態儲存卷: • 直接通過 volume 使用 • 通過 PV/PVC 使用	支援
阿里雲 NAS	NAS 靜態儲存卷又支援下面兩 種使用方式: • 通過 flexvolume 外掛程式使 用 — 通過 volume 方式使用 — 使用 PV/PVC • 通過 Kubernetes 的 NFS 驅 動使用	支援
阿里雲 OSS	您可以通過以下兩種方式使用 OSS 靜態儲存卷: • 直接使用 volume 方式 • 使用 PV/PVC	不支援

# 1.9.2 安裝外掛程式

通過下面的 yaml 配置部署阿里雲 Kubernetes 儲存外掛程式。

道 说明:

如果您的 Kubernetes 叢集是在 2018 年 2 月 6 日之前建立的,那麼您使用資料卷之間需要先安

裝阿里雲 Kubernetes 儲存外掛程式;如果您的 Kubernetes 叢集是在 2018 年 2 月 6 日之後建立

的,那麼您可以直接使用資料卷,不需要再安裝阿里雲 Kubernetes 儲存外掛程式。

使用限制

目前支援 CentOS 7 作業系統。

## 注意事項

使用 flexvolume 需要 kubelet 關閉 --enable-controller-attach-detach 選項。預設阿
 里雲 Kubernetes 叢集已經關閉此選項。

• 在 kube-system 使用者空間部署 flexvolume。

#### 驗證安裝完成:

在 master 節點上:

- 執行命令:kubectl get pod -n kube-system | grep flexvolume。輸出若干(節點 個數)Running 狀態的 Pod 列表。
- 執行命令:kubectl get pod -n kube-system | grep alicloud-diskcontroller。輸出一個 Running 狀態的 Pod 列表。

安裝步驟

安裝 Flexvolume:

```
apiVersion: apps/vl # for versions before 1.8.0 use extensions/vlbetal
kind: DaemonSet
metadata:
 name: flexvolume
 namespace: kube-system
 labels:
   k8s-volume: flexvolume
spec:
 selector:
   matchLabels:
     name: acs-flexvolume
  template:
   metadata:
      labels:
       name: acs-flexvolume
    spec:
      hostPID: true
      hostNetwork: true
      tolerations:
       - key: node-role.kubernetes.io/master
        operator: Exists
        effect: NoSchedule
      containers:
      - name: acs-flexvolume
        image: registry.cn-hangzhou.aliyuncs.com/acs/flexvolume:v1.9.7
-42e8198
        imagePullPolicy: Always
        securityContext:
         privileged: true
        env:
        - name: ACS_DISK
         value: "true"
        - name: ACS_NAS
          value: "true"
        - name: ACS_OSS
          value: "true"
        resources:
          limits:
            memory: 200Mi
          requests:
            cpu: 100m
            memory: 200Mi
```
```
volumeMounts:
  - name: usrdir
   mountPath: /host/usr/
  - name: etcdir
   mountPath: /host/etc/
  - name: logdir
   mountPath: /var/log/alicloud/
volumes:
- name: usrdir
 hostPath:
   path: /usr/
- name: etcdir
 hostPath:
   path: /etc/
- name: logdir
hostPath:
   path: /var/log/alicloud/
```

安裝 Disk Provisioner:

```
_ _ _
kind: StorageClass
apiVersion: storage.k8s.io/v1beta1
metadata:
 name: alicloud-disk-common
provisioner: alicloud/disk
parameters:
 type: cloud
kind: StorageClass
apiVersion: storage.k8s.io/v1beta1
metadata:
 name: alicloud-disk-efficiency
provisioner: alicloud/disk
parameters:
 type: cloud_efficiency
_ _ _
kind: StorageClass
apiVersion: storage.k8s.io/v1beta1
metadata:
 name: alicloud-disk-ssd
provisioner: alicloud/disk
parameters:
 type: cloud_ssd
kind: StorageClass
apiVersion: storage.k8s.io/v1beta1
metadata:
 name: alicloud-disk-available
provisioner: alicloud/disk
parameters:
type: available
kind: ClusterRole
apiVersion: rbac.authorization.k8s.io/v1beta1
metadata:
 name: alicloud-disk-controller-runner
rules:
  - apiGroups: [""]
    resources: ["persistentvolumes"]
    verbs: ["get", "list", "watch", "create", "delete"]
```

```
- apiGroups: [""]
   resources: ["persistentvolumeclaims"]
   verbs: ["get", "list", "watch", "update"]
  - apiGroups: ["storage.k8s.io"]
   resources: ["storageclasses"]
    verbs: ["get", "list", "watch"]
  - apiGroups: [""]
    resources: ["events"]
    verbs: ["list", "watch", "create", "update", "patch"]
apiVersion: v1
kind: ServiceAccount
metadata:
 name: alicloud-disk-controller
 namespace: kube-system
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1beta1
metadata:
 name: run-alicloud-disk-controller
subjects:
  - kind: ServiceAccount
   name: alicloud-disk-controller
   namespace: kube-system
roleRef:
 kind: ClusterRole
 name: alicloud-disk-controller-runner
 apiGroup: rbac.authorization.k8s.io
kind: Deployment
apiVersion: extensions/vlbetal
metadata:
 name: alicloud-disk-controller
 namespace: kube-system
spec:
 replicas: 1
  strategy:
    type: Recreate
  template:
   metadata:
      labels:
        app: alicloud-disk-controller
    spec:
      tolerations:
      - effect: NoSchedule
        operator: Exists
       key: node-role.kubernetes.io/master
      - effect: NoSchedule
        operator: Exists
        key: node.cloudprovider.kubernetes.io/uninitialized
      nodeSelector:
         node-role.kubernetes.io/master: ""
      serviceAccount: alicloud-disk-controller
      containers:
        - name: alicloud-disk-controller
          image: registry.cn-hangzhou.aliyuncs.com/acs/alicloud-disk-
controller:v1.9.3-ed710ce
          volumeMounts:
            - name: cloud-config
              mountPath: /etc/kubernetes/
            - name: logdir
              mountPath: /var/log/alicloud/
```

```
volumes:
    - name: cloud-config
    hostPath:
        path: /etc/kubernetes/
    - name: logdir
    hostPath:
        path: /var/log/alicloud/
```

## 1.9.3 使用阿里雲雲端硬碟

您可以在阿里雲Container Service Kubernetes 叢集中使用阿里雲雲端硬碟儲存卷。

目前,阿里雲雲端硬碟提供兩種 Kubernetes 掛載方式:

• #####

您可以通過以下兩種方式使用雲端硬碟靜態儲存卷:

- ####volume##
- ## PV/PVC ##
- #######

📕 说明:

對建立的雲端硬碟容量有如下要求:

- 普通雲端硬碟:最小5Gi
- 高效雲端硬碟:最小20Gi
- SSD雲端硬碟:最小20Gi

#### 靜態儲存卷

您可以直接通過volume使用阿里雲雲端硬碟儲存卷或者通過 PV/PVC 使用阿里雲雲端硬碟儲存卷。

前提條件

使用雲端硬碟資料卷之前,您需要先在 ECS 管理主控台上建立雲端硬碟。有關如何建立雲端硬碟,參見#####。

#### 使用說明

- 雲端硬碟為非共用儲存,只能同時被一個 pod 掛載。
- 使用雲端硬碟儲存卷前需要先申請一個雲端硬碟,並獲得磁碟 ID。參見 #####。
- volumeId: 表示所掛載雲端硬碟的磁碟ID; volumeName、PV Name要與之相同。
- 叢集中只有與雲端硬碟在同一個可用性區域(Zone)的節點才可以掛載雲端硬碟。

#### 直接通過 volume 使用

使用disk-deploy.yaml檔案建立 Pod。

```
apiVersion: extensions/vlbetal
kind: Deployment
metadata:
 name: nginx-disk-deploy
spec:
 replicas: 1
 template:
   metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx-flexvolume-disk
        image: nginx
        volumeMounts:
          - name: "d-bp1j17ifxfasvts3tf40"
            mountPath: "/data"
      volumes:
        - name: "d-bp1j17ifxfasvts3tf40"
          flexVolume:
            driver: "alicloud/disk"
            fsType: "ext4"
            options:
              volumeId: "d-bp1j17ifxfasvts3tf40"
```

通過 PV/PVC 使用

步驟 1 建立雲端硬碟類型的 PV

您可以使用 yaml 檔案或者控制台介面建立雲端硬碟類型的 PV。

#### 通過 yaml 檔案建立 PV

使用 disk-pv.yaml 檔案建立 PV。

**送** 说明:

pv name 要與阿里雲盤 ID 相同。

```
apiVersion: v1
kind: PersistentVolume
metadata:
 name: d-bp1j17ifxfasvts3tf40
 labels:
    failure-domain.beta.kubernetes.io/zone: cn-hangzhou-b
    failure-domain.beta.kubernetes.io/region: cn-hangzhou
spec:
 capacity:
    storage: 20Gi
 storageClassName: disk
 accessModes:
    - ReadWriteOnce
  flexVolume:
   driver: "alicloud/disk"
    fsType: "ext4"
    options:
```

volumeId: "d-bp1j17ifxfasvts3tf40"

## 通過控制台介面建立雲端硬碟資料卷

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下, 單擊左側導覽列中的叢集 > 儲存卷, 進入資料卷列表頁面。
- 3. 選擇所需的叢集,單擊頁面右上方的建立。

容器服务 - Kubernetes ▼	存储卷列表							剧新	创建
概览	集群 k8s-cluster v 3								4
	名称	总量	访问模式	回收策略	状态	存储类型	创建时间		操作
集群									
点									
存储卷 2									

- 4. 在建立資料卷對話方塊中,配置資料卷的相關參數。
  - 資料卷類型:本樣本中為雲端硬碟。
  - 訪問模式:預設為 ReadWriteOnce。
  - 雲端硬碟 ID:您可以選擇與叢集屬於相同地區和可用性區域下處於待掛載狀態的雲端硬碟。
  - 檔案系統類型:您可以選擇以什麼資料類型將資料存放區到雲端硬碟上,支援的類型包括 ext4、ext3、xfs、vfat。預設為 ext4。
  - 標籤:為該資料卷添加標籤。

创建数据卷		$\times$
数据卷类型:	● 云盘 ○ NAS ○ OSS	
访问模式:	ReadWriteOnce	
云盘ID:	d-t 选择云盘	
文件系统类型:	ext4 🔻	
标签:	● 添加标签	
	创建取消	ŧ

## 5. 完成配置後,單擊建立。

步驟 2 建立 PVC

## 使用 disk-pvc.yaml 檔案建立 PVC。

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
   name: pvc-disk
spec:
   accessModes:
    - ReadWriteOnce
   storageClassName: disk
   resources:
      requests:
      storage: 20Gi
```

## 步驟 3 建立 Pod

使用disk-pod.yaml檔案建立 pod。

```
apiVersion: v1
kind: Pod
metadata:
   name: "flexvolume-alicloud-example"
spec:
   containers:
        - name: "nginx"
        image: "nginx"
        volumeMounts:
            - name: pvc-disk
            mountPath: "/data"
volumes:
        - name: pvc-disk
        persistentVolumeClaim:
        claimName: pvc-disk
```

## 動態儲存裝置卷

動態儲存裝置卷需要您手動建立 StorageClass,並在PVC中通過 storageClassName 來指定期望的 雲端硬碟類型。

### 建立 StorageClass

```
kind: StorageClass
apiVersion: storage.k8s.io/v1beta1
metadata:
    name: alicloud-disk-common-hangzhou-b
provisioner: alicloud/disk
parameters:
    type: cloud_ssd
    regionid: cn-hangzhou
    zoneid: cn-hangzhou-b
```

### 參數說明:

• provisioner: 配置為 alicloud/disk,標識使用阿里雲雲端硬碟 Provsioner 外掛程式建立。

- type:標識雲端硬碟類型,支援 cloud、cloud\_efficiency、cloud\_ssd、available 四種類型;其 中 available 會對高效、SSD、普通雲端硬碟依次嘗試建立,直到建立成功。
- regionid:期望建立雲端硬碟的地區。
- reclaimPolicy: 雲端硬碟的回收策略,預設為Delete,支援Retain。
- zoneid:期望建立雲端硬碟的可用性區域。

### 建立服務

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
 name: disk-common
spec:
  accessModes:
    - ReadWriteOnce
  storageClassName: alicloud-disk-common-hangzhou-b
  resources:
   requests:
      storage: 20Gi
_ _ _
kind: Pod
apiVersion: v1
metadata:
 name: disk-pod-common
spec:
  containers:
  - name: disk-pod
   image: nginx
   volumeMounts:
      - name: disk-pvc
        mountPath: "/mnt"
  restartPolicy: "Never"
  volumes:
    - name: disk-pvc
      persistentVolumeClaim:
        claimName: disk-common
```

## 預設選項

叢集預設提供了下面幾種 StorageClass,可以在單 AZ 類型的叢集中使用。

- alicloud-disk-common: 普通雲端硬碟。
- alicloud-disk-efficiency:高效雲端硬碟。
- alicloud-disk-ssd:SSD雲端硬碟。
- alicloud-disk-available:提供高可用選項,先試圖建立高效雲端硬碟;如果相應AZ的高效雲端硬 碟資源售盡,再試圖建立SSD盤;如果SSD售盡,則試圖建立普通雲端硬碟。

使用雲端硬碟建立多執行個體StatefulSet

## 使用 volumeClaimTemplates 的方式來建立,這樣會動態建立多個 PVC 和 PV 並綁定。

```
apiVersion: v1
kind: Service
metadata:
 name: nginx
 labels:
   app: nginx
spec:
 ports:
  - port: 80
   name: web
 clusterIP: None
 selector:
   app: nginx
apiVersion: apps/v1beta2
kind: StatefulSet
metadata:
 name: web
spec:
 selector:
   matchLabels:
     app: nginx
 serviceName: "nginx"
 replicas: 2
 template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
       image: nginx
        ports:
        - containerPort: 80
          name: web
        volumeMounts:
        - name: disk-common
          mountPath: /data
 volumeClaimTemplates:
  - metadata:
      name: disk-common
    spec:
      accessModes: [ "ReadWriteOnce" ]
      storageClassName: "alicloud-disk-common"
      resources:
        requests:
          storage: 10Gi
```

# 1.9.4 使用阿里雲 NAS

您可以在Container Service Kubernetes 叢集中使用阿里雲 NAS 資料卷。

目前阿里雲 NAS 提供兩種 Kubernetes 掛載方式:

• #####

其中,靜態儲存卷又支援下面兩種使用方式:

- 通過 flexvolume 外掛程式使用
  - 通過 volume 方式使用
  - 使用 PV/PV
- 通過 Kubernetes 的 NFS 驅動使用
- ########

#### 前提條件

使用 NAS 資料卷之前,您需要先在 NAS 管理主控台上建立檔案系統,並在檔案系統中添加 Kubernetes 叢集的掛載點。建立的 NAS 檔案系統需要和您的叢集位於同一 VPC。

#### 靜態儲存卷

您可以通過阿里雲提供的 flexvolume 外掛程式使用阿里雲 NAS 檔案儲存體服務或者通過 Kubernetes 的 NFS 驅動使用阿里雲 NAS 檔案儲存體服務。

## 通過 flexvolume 外掛程式使用

使用 flexvolume 外掛程式,您可以通過 volume 方式使用阿里雲 NAS 資料卷或者通過 PV/PVC 方 式使用阿里雲 NAS 資料卷。

📕 说明:

- NAS 為共用儲存,可以同時為多個 Pod 提供共用儲存服務。
- server:為 NAS 資料盤的掛載點。
- path:為串連 NAS 資料卷的掛載目錄,支援掛載 NAS 子目錄,且當子目錄不存在時,會自動 建立子目錄並掛載。
- vers: 定義 nfs 掛載協議的版本號碼, 支援3.0 和 4.0。
- mode:定義掛載目錄的存取權限,注意掛載 NAS 盤根目錄時不能配置掛載許可權。當 NAS 盤 中資料量很大時,配置 mode 會導致執行掛載非常慢,甚至掛載失敗。

## 通過 volume 方式使用

使用 nas-deploy.yaml 檔案建立 Pod。

```
apiVersion: v1
kind: Pod
metadata:
   name: "flexvolume-nas-example"
spec:
   containers:
        - name: "nginx"
        image: "nginx"
        volumeMounts:
```

```
- name: "nasl"
    mountPath: "/data"
volumes:
    name: "nasl"
    flexVolume:
    driver: "alicloud/nas"
    options:
        server: "0cd8b4a576-grs79.cn-hangzhou.nas.aliyuncs.com"
        path: "/k8s"
        vers: "4.0"
```

使用 PV/PVC

步驟1建立 PV

您可以使用 yaml 檔案或者通過阿里雲Container Service控制台介面建立 NAS 資料卷。

• 使用 yaml 檔案建立 PV

使用 nas-pv.yaml 檔案建立 PV。

```
apiVersion: v1
kind: PersistentVolume
metadata:
 name: pv-nas
spec:
 capacity:
   storage: 5Gi
 storageClassName: nas
 accessModes:
    - ReadWriteMany
 flexVolume:
   driver: "alicloud/nas"
    options:
      server: "0cd8b4a576-uih75.cn-hangzhou.nas.aliyuncs.com"
     path: "/k8s"
      vers: "4.0"
```

- 通過控制台介面建立 NAS 資料卷
  - 1. 登入Container Service#####。
  - 2. 在 Kubernetes 菜單下,單擊左側導覽列中的叢集 > 儲存,進入資料卷列表頁面。
  - 3. 選擇所需的叢集,單擊頁面右上方的建立。

容器服务	数据卷列表							刷新创建
Kubernetes Swarm	集群: k8s-cluster 🔻 3							4
#群	名称	总量	访问模式	回收策略	状态	存储类型	创建时间	攝作
节点	d-bp1gyzplgtreo4zpvwda	20Gi	ReadWriteOnce	Delete	Bound	alicloud-disk-efficiency	2018-04-23 14:12:29	删除
存储 2								

- 4. 在建立資料卷對話方塊中, 配置資料卷的相關參數。
  - 資料卷類型:本樣本中為NAS。
  - 資料卷名:建立的資料卷的名稱。資料卷名在叢集內必須唯一。本例為 pv-nas。

- 資料卷總量:所建立資料卷的容量。注意不能超過磁碟容量。
- 訪問模式:預設為 ReadWriteMany。
- 掛載點網域名稱: 叢集在 NAS 檔案系統中掛載點的掛載地址。
- 子目錄:NAS 路徑下的子目錄,以 / 開頭,設定後資料卷將掛載到指定的子目錄。
  - 如果 NAS 根目錄下沒有此子目錄,會預設建立後再掛載。
  - 您可以不填此項,預設掛載到 NAS 根目錄。
- 許可權:設定掛載目錄的存取權限,例如:755、644、777等。
  - 只有掛載到 NAS 子目錄時才能設定許可權, 掛載到根目錄時不能設定。
  - 您可以不填此項,預設許可權為 NAS 檔案原來的許可權。
- 標籤:為該資料卷添加標籤。

创建数据卷	>	<
数据卷类型:	◯ 云盘 ● NAS ◯ OSS	
数据卷名:	pv-nas	
总量:	20Gi	
访问模式:	ReadWriteMany ReadWriteOnce	
挂载点域名:	.nas.aliyuncs.com	
子目录:	如:/xxx , 默认为根目录	
权限:	如:755,默认为NAS权限	
标签:	●添加标签	
	创建取消	

5. 完成配置後, 單擊建立。

步驟 2 建立 PVC

使用 nas-pvc.yaml 檔案建立 PVC。

```
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
   name: pvc-nas
spec:
   accessModes:
    - ReadWriteMany
   storageClassName: nas
   resources:
      requests:
      storage: 5Gi
```

## 步驟 3 建立 Pod

使用nas-pod.yaml檔案建立 pod。

## 通過 Kubernetes 的 NFS 驅動使用

### 步驟 1 建立 NAS 檔案系統

登入 ###########, 建立一個 NAS 檔案系統。

📋 说明 :

建立的 NAS 檔案系統需要和您的叢集位於同一地區。

假設您的掛載點為 055f84ad83-ixxxx.cn-hangzhou.nas.aliyuncs.com。

## 步驟 2 建立 PV

您可以使用編排模板或者通過阿里雲Container Service控制台介面建立 NAS 資料卷。

• 使用編排模板

使用nas-pv.yam1檔案建立 PV。

執行以下命令建立一個類型為 NAS 的 PesistentVolume。

```
root@master # cat << EOF |kubectl apply -f -
apiVersion: v1
kind: PersistentVolume
metadata:
  name: nas
spec:
   capacity:
     storage: 8Gi
   accessModes:
     - ReadWriteMany
   persistentVolumeReclaimPolicy: Retain
   nfs:
     path: /
     server: 055f84ad83-ixxxx.cn-hangzhou.nas.aliyuncs.com
EOF
```

• 通過控制台介面建立 NAS 資料卷

具體操作參見 ##PV/PVC 中關於通過控制台介面建立 NAS 資料卷的內容。

步驟 2 建立 PVC

建立一個 PersistentVolumeClaim 來請求綁定該 PersistentVolume。

```
root@master # cat << EOF | kubectl apply -f -
apiVersion: v1
kind: PersistentVolumeClaim
metadata:
     name: nasclaim
spec:
     accessModes:
       - ReadWriteMany
     resources:
       requests:
         storage: 8Gi
EOF
```

#### 步驟 3 建立 Pod

建立一個應用來申明掛載使用該資料卷。

```
root@master # cat << EOF |kubectl apply -f -
apiVersion: v1
kind: Pod
metadata:
     name: mypod
spec:
     containers:
       - name: myfrontend
         image: registry.aliyuncs.com/spacexnice/netdia:latest
         volumeMounts:
         - mountPath: "/var/www/html"
           name: mypd
     volumes:
       - name: mypd
         persistentVolumeClaim:
```

claimName: nasclaim

至此,您就將 NAS 遠程檔案系統掛載到了您的 Pod 應用當中了。

#### 動態儲存裝置卷

EOF

使用動態 NAS 儲存卷需要您手動安裝驅動外掛程式,並配置 NAS 掛載點。

安裝外掛程式

```
apiVersion: storage.k8s.io/v1
kind: StorageClass
metadata:
 name: alicloud-nas
provisioner: alicloud/nas
apiVersion: v1
kind: ServiceAccount
metadata:
 name: alicloud-nas-controller
 namespace: kube-system
kind: ClusterRoleBinding
apiVersion: rbac.authorization.k8s.io/v1beta1
metadata:
 name: run-alicloud-nas-controller
subjects:
  - kind: ServiceAccount
   name: alicloud-nas-controller
   namespace: kube-system
roleRef:
 kind: ClusterRole
 name: alicloud-disk-controller-runner
 apiGroup: rbac.authorization.k8s.io
kind: Deployment
apiVersion: extensions/vlbetal
metadata:
 name: alicloud-nas-controller
 namespace: kube-system
spec:
 replicas: 1
  strategy:
    type: Recreate
  template:
    metadata:
      labels:
        app: alicloud-nas-controller
    spec:
      tolerations:
      - effect: NoSchedule
        operator: Exists
        key: node-role.kubernetes.io/master
      - effect: NoSchedule
        operator: Exists
        key: node.cloudprovider.kubernetes.io/uninitialized
      nodeSelector:
         node-role.kubernetes.io/master: ""
      serviceAccount: alicloud-nas-controller
```

```
containers:
        - name: alicloud-nas-controller
          image: registry.cn-hangzhou.aliyuncs.com/acs/alicloud-nas-
controller:v1.8.4
          volumeMounts:
           - mountPath: /persistentvolumes
            name: nfs-client-root
          env:
            - name: PROVISIONER_NAME
              value: alicloud/nas
            - name: NFS_SERVER
              value: 0cd8b4a576-mmi32.cn-hangzhou.nas.aliyuncs.com
            - name: NFS_PATH
              value: /
      volumes:
        - name: nfs-client-root
          nfs:
            server: 0cd8b4a576-mmi32.cn-hangzhou.nas.aliyuncs.com
            path: /
```

### 使用動態儲存裝置卷

```
apiVersion: apps/vlbetal
kind: StatefulSet
metadata:
 name: web
spec:
 serviceName: "nginx"
  replicas: 2
  volumeClaimTemplates:
  - metadata:
      name: html
    spec:
      accessModes:
        - ReadWriteOnce
      storageClassName: alicloud-nas
      resources:
        requests:
          storage: 2Gi
  template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:alpine
        volumeMounts:
        - mountPath: "/usr/share/nginx/html/"
          name: html
```

# 1.9.5 使用阿里雲 OSS

您可以在阿里雲Container Service Kubernetes 叢集中使用阿里雲 OSS 資料卷。

目前,僅支援 OSS 靜態儲存卷,不支援 OSS 動態儲存裝置卷。您可以通過以下方式使用 OSS 靜 態儲存卷:

- 直接使用 volume 方式
- 使用 PV/PVC

### 前提條件

使用 OSS 靜態儲存卷之前,您需要先在 OSS 管理主控台上建立 Bucket。

#### 使用說明

- OSS 為共用儲存,可以同時為多個 Pod 提供共用儲存服務。
- bucket:目前只支援掛載 Bucket,不支援掛載 Bucket 下面的子目錄或檔案。
- url: OSS endpoint, 掛載 OSS 的接入網域名稱。
- akld: 使用者的 access id 值。
- akSecret:使用者的 access secret 值。
- otherOpts: 掛載 OSS 時支援定製化參數輸入,格式為: -o \*\*\* -o \*\*\*。

#### 注意事項

如果您的 Kubernetes 叢集是在 2018 年 2 月 6 日之前建立的,那麼您使用資料卷之間需要先安裝# #####。使用 OSS 資料卷必須在部署 flexvolume 服務的時候建立 Secret,並輸入 AK 資訊。

#### 使用 OSS 靜態卷

## 直接使用 volume 方式

使用oss-deploy.yaml 檔案建立 Pod。

```
apiVersion: extensions/vlbetal
kind: Deployment
metadata:
 name: nginx-oss-deploy
spec:
 replicas: 1
 template:
    metadata:
      labels:
        app: nginx
    spec:
      containers:
      - name: nginx-flexvolume-oss
       image: nginx
        volumeMounts:
          - name: "oss1"
            mountPath: "/data"
      volumes:
        - name: "oss1"
          flexVolume:
            driver: "alicloud/oss"
            options:
              bucket: "docker"
              url: "oss-cn-hangzhou.aliyuncs.com"
              akId: ***
```

```
akSecret: ***
otherOpts: "-o max_stat_cache_size=0 -o allow_other"
```

使用 PV/PVC(目前不支援動態 pv)

#### 步驟1建立 PV

您可以使用 yaml 檔案或者通過Container Service控制台介面建立 PV。

#### 使用 yaml 檔案建立 PV

使用oss-pv.yaml檔案建立 PV。

```
apiVersion: v1
kind: PersistentVolume
metadata:
 name: pv-oss
spec:
 capacity:
   storage: 5Gi
 accessModes:
    - ReadWriteMany
 storageClassName: oss
 flexVolume:
   driver: "alicloud/oss"
    options:
      bucket: "docker"
      url: "oss-cn-hangzhou.aliyuncs.com"
      akId: ***
      akSecret: ***
      otherOpts: "-o max_stat_cache_size=0 -o allow_other"
```

通過控制台介面建立 OSS 資料卷

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的叢集 > 儲存,進入資料卷列表頁面。
- 3. 選擇所需的叢集,單擊頁面右上方的建立。

容器服务	数据卷列表							周新创建
Kubernetes Swarm	集群: k8s-cluster ▼ 3							4
	名称	总量	访问模式	回收策略	状态	存储类型	创建时间	摄作
节点	d-bp1gyzplgtreo4zpvwda	20Gi	ReadWriteOnce	Delete	Bound	alicloud-disk-efficiency	2018-04-23 14:12:29	删除
存储 2								

- 4. 在建立資料卷對話方塊中, 配置資料卷的相關參數。
  - 資料卷類型:本樣本中為 OSS。
  - 資料卷名:建立的資料卷的名稱。資料卷名在叢集內必須唯一。本例為 pv-oss。
  - 資料卷總量:所建立資料卷的容量。
  - 訪問模式:預設為 ReadWriteMany。
  - AccessKey ID、AccessKey Secret:訪問 OSS 所需的 AccessKey。

- Bucket ID: 您要使用的 OSS bucket 的名稱。單擊選擇Bucket, 在彈出的對話方塊中選擇 所需的 bucket 並單擊選擇。
- 訪問網域名稱:如果 Bucket 和 ECS 執行個體位於不同地區(Region),請選擇外網網域名 稱;如果位於相同地區,需要根據叢集網路類型進行選擇,若是 VPC 網路,請選擇VPC網域 名稱,若是傳統網路,請選擇內網網域名稱。
- 標籤:為該資料卷添加標籤。

创建数据卷	$\times$
数据卷类型: 数据卷名:	○ 云盘 ○ NAS ◎ OSS
总量:	20Gi
访问模式:	ReadWriteMany
AccessKey ID :	
AccessKey Secret :	
可选参数:	
	更多参数的填写格式可以 参考该文档
Bucket ID :	选择Bucket
访问域名:	○ 内网域名 ○ 外网域名 ● vpc域名 ②
标签:	●添加标签
	创建取消

## 5. 完成配置後,單擊建立。

## 步驟 2 建立 PVC

使用oss-pvc.yaml檔案建立 PVC。

kind: PersistentVolumeClaim

```
apiVersion: v1
metadata:
   name: pvc-oss
spec:
   storageClassName: oss
   accessModes:
        - ReadWriteMany
   resources:
        requests:
        storage: 5Gi
```

## 步驟 3 建立 Pod

使用oss-pod.yaml 建立 Pod。

```
apiVersion: v1
kind: Pod
metadata:
   name: "flexvolume-oss-example"
spec:
   containers:
        - name: "nginx"
        image: "nginx"
        volumeMounts:
            - name: pvc-oss
            mountPath: "/data"
volumes:
        - name: pvc-oss
        persistentVolumeClaim:
            claimName: pvc-oss
```

#### 使用 OSS 動態磁碟區

目前暫不支援。

## 1.9.6 建立持久化儲存卷聲明

您可通過Container Service控制台建立持久化儲存卷聲明(PVC)。

前提条件

- 您已建立一個Kubernetes叢集,參見##Kubernetes##。
- 您已建立一個儲存卷,本例中使用雲端硬碟建立一個雲端硬碟儲存卷,參見##########

預設根據標籤alicloud-pvname將儲存聲明和儲存卷綁定,通過Container Service控制台建立 資料卷時,會預設給儲存卷打上該標籤。如果儲存卷上沒有該標籤,您需要添加標籤後才可以選 擇關聯這個儲存卷。

背景信息

#### 操作步骤

- 1. 登入Container Service#####。
- 2. 在Kubernetes菜單下,單擊左側導覽列中的應用 > 儲存聲明,進入儲存聲明列表頁面。

3. 選擇所需的叢集和命名空間,單擊右上方的建立。

容器服务 - Kubernetes ▼	存储声明列表							Rjør dyst
概范	集群 K8S-cs ▼	命名空间	default 🔹 3					4
▼ 集群	名称	总量	访问模式	状态	存储类型	关联的存储卷	创建时间	操作
集群								
节点					① 没有查询到符	合条件的记录		
存储卷								
命名空间								
▼应用 1								
部署								
容器组								
服务								
路由								
存储声明 2								

4. 在建立儲存聲明對話方塊中進行配置,最後單擊建立。

创建存储声明	×
存储声明类型:	● 云盘 ○ NAS ○ OSS
名称 :	pvc-disk 名称必须以小写字母开头,只能包含小写字母、数 字、","和"-"
分配模式:	◎ 已有存储卷
已有存储卷 :	d-bp14ss6t80e7emx9lv6e, 20Gi 选择已有存储卷
总量:	20Gi
	创建取消

- 儲存宣告類型: 和儲存卷一致,包括雲端硬碟/NAS/OSS
- 名稱: 輸入儲存卷聲明名稱
- 分配模式: 目前只支援已有儲存卷
- 已有儲存卷: 選擇該類型下的儲存卷綁定
- 總量:聲明使用量,不能大於儲存卷的總量

▋ 说明:

若您的叢集中已有儲存卷,且未被使用,但在選擇已有儲存卷無法找到,則可能是未定義 alicloud-pvname標籤。

若無法找到可用的儲存卷,您可在左側導覽列中單擊叢集 > 儲存卷,找到所需儲存卷,單擊右側 的標籤管理,添加對應標籤,其中名稱為alicloud-pvname,值為儲存卷的名稱,雲端硬碟儲 存卷預設以雲端硬碟ID作為儲存卷的名稱。

标签管理	$\times$
● 添加标签	
名称	值
failure-domain.beta.kubernetes.io/zone	cn-hangzhou-g
failure-domain.beta.kubernetes.io/region	cn-hangzhou 🗢
alicloud-pvname	d-bp14ss6t80e7emx9lv6e
	确定关闭

5. 返回儲存聲明列表,您可看到建立的儲存聲明出現在列表中。

# 1.9.7 使用持久化儲存卷聲明

前提条件

- 您已建立一個Kubernetes叢集,參見##Kubernetes##。
- 您已建立一個儲存卷聲明,本例中使用雲端硬碟建立一個雲端硬碟儲存卷聲明pvc-disk,參見##
   #########

操作步骤

- 1. 登入Container Service#####。
- 在Kubernetes菜單下,單擊左側導覽列中的應用 > 部署,進入部署列表頁面,單擊右上方的使 用鏡像建立。

容器服务 - Kubernetes ▼	部署列表				周新	使用镜像创建	使用模	板创建
概范	集群 K8S-cs	▼ 命名空间 default ▼ 2				3		
▼ 集群	名称	标签	容器组数量	创建时间				操作
集群	grpc-service	run:grpc-service	1/1	2018-07-02 14:12:37		详情	编辑   监持	2│更多▼
节点								
存储卷								
命名空间								
▼ 应用 部署								

3. 在應用基本資料配置頁面,設定應用程式名稱、部署叢集和命名空間,然後單擊下一步。

创建应用							
	应用基本信息	应	用配置	$\rangle$	访问设置	$\rangle$	创建完成
应用名称	test-nginx						
	名称为1-64个字符,可包含数字、小	小写英文字符,或"-",且不能	能以-开头				
部署集群:	K8S-cs	٣					
命名空间:	default	٣					
							返回 下一步

在應用配置頁面,選擇鏡像,然後配置雲端儲存類型的資料卷,支援雲端硬碟/NAS/OSS 三種類型。本例中使用準備好的雲端硬碟儲存卷聲明,最後單擊下一步。

	应用	基本信息			应用	配置		访问设置	$\rightarrow$	创建分	記成
镜像名称	尔:	nginx				镜像版本:	latest			配置概述	
		选择镜像					选择镜像版本			基本配置	
容器数量	≣:	1								致 描卷 环境 变量	
资源限制	则:	CPU 500m	• 内谷	予 如:128Mi	0					容器配置 调度配置	
所需资源	<u>ছ</u> :	CPU	0 内石	₹ 如:128Mi	0					上一步	下一步
数据卷		🛨 増加本地存储									
		存储卷类型		挂载源		容器路径					
		增加云存储									
		存储卷类型		挂载源		容器路径					
		云盘		pvc-disk		v /tmp			•		

- 5. 配置test-nginx應用的服務,最後單擊建立。
- 6. 應用建立完畢後,單擊左側導覽列中的應用 > 容器組,找到該應用所屬容器組,單擊詳情。

杏醋组							刷新
集群 KBS-cs v 命名空间 default	Ŧ						
名称 🔷	状态 🔷	Pod IP 🜲	⇒ 点寸	创建时间 🜩	CPU (核)	内存(字节)	
grpc-service-75ccfff446-zsdvf	●运行中	10.0.1.138	192.168.34.189	2018-07-02 14:12:37	0	0	详情 │ 更多 ▼
test-nginx-deployment-85ffc89fb7-ndn4m	●运行中	10.0.1.139	192.168.34.189	2018-07-04 16:38:25	0	0	详情 │ 更多 ▼

7. 在容器組詳情頁面,單擊儲存,您可看到該容器組正確綁定了pvc-disk。

容器组 - test-nginx-deployment-85ffc89fb7-ndn4m						
基本信息						
- 名称: test-nginx-deployment-85ffc89fb7-ndn4m 命名空间: default						
状态: Running		创建时间: 2018-07-04 16:38:25				
节点: cn-hangzhou.i-bp153888e85yy0cyw659		Pod IP :				
标签: app: test-nginx pod-template-hash: 4199745963						
容器 事件 创建者 初始化容器 存储 日志						
名称	类型	详情				
volume-1530693170118	persistentVolumeClaim	claimName: pvc-disk				
default-token-w689p	secret	defaultMode: 420 secretName: default-token-w689p				

# 1.10 日誌管理

# 1.10.1 概述

阿里雲Container Service Kubernetes 叢集提供給您多種方式進行應用的日誌管理。

- 通過阿里雲Log Service#*unique\_97*,您可以方便地使用Log Service採集應用日誌,從而更好地 利用阿里雲Log Service提供給您的各種日誌統計分析等功能。
- 通過阿里雲Container Service提供的開源 *Log-pilot* 項目,#*unique\_98*,您可以方便地搭建自己的應用日誌叢集。

## 1.10.2 查看叢集日誌

## 背景信息

您可以通過Container Service的簡單Log Service查看叢集的動作記錄。

#### 操作步骤

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的叢集,進入 Kubernetes 叢集列表頁面。
- 3. 選擇所需的叢集並單擊右側的查看日誌。

容器服务	集群列表			您最多可以创题	書 5 个集群,每个集群最多可以添加 20 イ	节点  刷新	创建Kubernetes 集群 🗸 👻
Kubernetes Swarm							
▼集群	常见问题: Ø 如何创建集群 Ø 扩容和缩容集群 。	⑦ 通过 kubectl 连接 Kubernetes 集群	Ø 通过命令管理应用				
(生産) (生産) (生産) (生産) (生産) (生産) (生産) (生産)	名称 🔻						
节点	集群名称/ID	地域 网络类型	集群状态	创建时间	Kubernetes 版本		2 操作
存储	k8s-cluster	华东1 虚拟专有网络 vpc-bp1gnu8br4a	●运行中	2018-04-23 11:41:46	1.9.3	1	管理   査君日志   控制台 集群伸缩   更多▼

## 您可以查看該叢集的操作資訊。

集群日志信息: k8s-cluster 🔹 返回集群列表	8	刷新
资源部署详细日志请参考: 资源栈事件		
时间	信息	
2018-04-23 11:59:27	c98a72d1cf17549e28aeb94a09b5b1eb4   Set up k8s DNS configuration successfully	
2018-04-23 11:58:23	c98a72d1cf17549e28aeb94a09b5b1eb4   Stack CREATE completed successfully:	
2018-04-23 11:58:23	c98a72d1cf17549e28aeb94a09b5b1eb4   Start describeStackInfo	
2018-04-23 11:58:22	c98a72d1cf17549e28aeb94a09b5b1eb4   Start describeStackInfo	
2018-04-23 11:41:48	c98a72d1cf17549e28aeb94a09b5b1eb4   Successfully to CreateStack	
2018-04-23 11:41:48	c98a72d1cf17549e28aeb94a09b5b1eb4   Start to wait stack ready	
2018-04-23 11:41:46	c98a72d1cf17549e28aeb94a09b5b1eb4   Start to getNatGatway	
2018-04-23 11:41:46	c98a72d1cf17549e28aeb94a09b5b1eb4   Start to create cluster task	
2018-04-23 11:41:46	c98a72d1cf17549e28aeb94a09b5b1eb4   Start to CreateK8sCluster	
2018-04-23 11:41:46	c98a72d1cf17549e28aeb94a09b5b1eb4   Start to CreateStack	
2018-04-23 11:41:45	c98a72d1cf17549e28aeb94a09b5b1eb4   Start to validateCIDR	
2018-04-23 11:41:41	c98a72d1cf17549e28aeb94a09b5b1eb4   Start create cluster certificate	

## 1.10.3 為 Kubernetes 和Log Service 配置 Log4JAppender

Log4j 是 Apache 的一個開放原始碼項目。Log4j 由三個重要組件構成:日誌資訊的優先順序、日誌 資訊的輸出目的地、日誌資訊的輸出格式。通過配置 Log4jAppender,您可以控制日誌資訊輸送的 目的地是控制台、檔案、GUI 組件、甚至是套介面伺服器、NT 的事件記錄器、UNIX Syslog 守護 進程等。

本文介紹在不需要修改應用代碼的前提下,通過配置一個 yaml 檔案,將阿里雲Container Service Kubernetes 叢集中產生的日誌輸出到阿里雲Log Service。此外,通過在 Kubernetes 叢集上部署一 個樣本 API 程式,來進行示範。

前提條件

• 您已經開通Container Service, 並且建立了 Kubernetes 叢集。

本樣本中,建立的 Kubernetes 叢集位於華東1地區。

• 啟用 AccessKey 或 RAM,確保有足夠的存取權限。本例中使用 AccessKey。

### 步驟1在阿里雲Log Service上配置 Log4jAppender

- 1. 登入 Log Service#####。
- 2. 單擊頁面右上方的建立 Project, 填寫 Project 的基本資料並單擊確認進行建立。

本樣本建立一個名為 k8s-log4j,與 Kubernetes 叢集位於同一地區(華東 1)的 Project。

## 送明:

在配置時,一般會使用與 Kubernetes 叢集位於同一地區的Log Service Project。因為當 Kubernetes 叢集和Log Service Project 位於同一地區時,日誌資料會通過內網進行傳輸,從而 避免了因地區不一致而導致的資料轉送外網頻寬費用和耗時,從而實現即時採集、快速檢索的 最佳實務。

创建Project	$\times$
* Project名称: k8s-log4j	]
注释: <u>k8s-log4j</u> demo	
不支持<>"最多包含64个字符 * 所属地域: 华东1 ▼	
确认	取消

- 3. 建立完成後,k8s-log4j 出現在 project 列表下,單擊該 project 名稱,進入 project 詳情頁面。
- 4. 預設進入日誌庫頁面,單擊右上方的建立。

日志库 1					地域:华东1
▼ LogHub - 实时采集	Logstore列表			查看Endpoin	t 创建
[文档] 接入指南	请给入LonstoreS讲行摄解到 接去				2
Logtail配置	434831, (reofinin in HIST ) 1964875		日志消费模式		
Logtail机器组	Logstore名称 数据接入向导 监控 日志采集模式	日志消费	日志投递	查询分析	操作
<ul> <li>LogHub - 实时消费</li> <li>(立約) ※書任志</li> </ul>					
[又行] 清英百用	您还没有添加Logstore,立即创建				
<ul> <li>Search/Analytics</li> </ul>					
快速查询			共有0条 集而月		
报警配置			2019 0元, 中央生	2/3/ : 20/3R 《 〈	1 > >

5. 填寫日誌庫配置資訊並單擊確認。

本樣本建立名為 k8s-logstore 的日誌庫。

创建Logstore		×
* Logstore名称:	k8s-logstore	
Logstore属性		
* WebTracking:	WebTracking功能支持快速采集各种浏览器以及 iOS/Android/APP访问信息,默认关闭(帮助)	
* 数据保存时间:	30 目前Loghub保存时间和索引已经统一,数据生命周 期以Loghub设置为准(单位:天)	
* Shard数目:	2 ▼ 什么是分区(Shard)?	
* 计费:	参考计费中心说明	
	确认	取消

## 6. 建立完畢後,頁面會提示您建立資料接入嚮導。

创建		×
•	您已成功创建Logstore,请使用数据接入向导快速设置采集、分析等使用 方式	
	数据接入向导取消	i

7. 選擇自訂資料下的log4jAppender,根據頁面引導進行配置。

本樣本使用了預設配置,您可根據日誌資料的具體使用情境,進行相應的配置。

自定义数据								
	文本文件	SysLog	Web Tracking	API/SDK	● log4jAppender	Producer		
	Logtail自定义插件							
							下一步	

#### 步驟 2 在 Kubernetes 叢集中配置 log4j

本樣本使用 demo-deployment 和 demo-Service 樣本 yaml 檔案進行示範。

1. 串連到您的 Kubernetes 叢集。

具體操作參見SSH##Kubernetes## 或 ## kubectl ## Kubernetes ##。

採集。

demo-deployment.yaml 檔案的樣本編排如下。

```
apiVersion: apps/vlbeta2
kind: Deployment
metadata:
 name: log4j-appender-demo-spring-boot
 labels:
   app: log4j-appender
spec:
 replicas: 1
 selector:
   matchLabels:
      app: log4j-appender
  template:
   metadata:
    labels:
      app: log4j-appender
 spec:
    containers:
    - name: log4j-appender-demo-spring-boot
      image: registry.cn-hangzhou.aliyuncs.com/jaegertracing/log4j-
appender-demo-spring-boot:0.0.2
      env:
      - name: JAVA_OPTS
                                        ##注意
       value: "-Dproject={your_project} -Dlogstore={your_logstore
} -Dendpoint={your_endpoint} -Daccess_key_id={your_access_key_id} -
Daccess_key={your_access_key_secret}"
     ports:
      - containerPort: 8080
```

其中:

• -Dproject:您所使用的阿里雲Log Service Project 的名稱。本樣本中為 k8s-log4j。

- -Dlogstore: 您所使用的阿里雲Log Service Logstore 的名稱。本樣本中為 k8s-logstore。
- -Dendpoint: Log Service的服務入口,使用者需要根據日誌 Project 所屬的地區,配置自 己的服務入口,參見####進行查詢。本樣本中為 cn-hangzhou.log.aliyuncs.com。
- -Daccess\_key\_id: 您的 AccessKey ID。
- -Daccess\_key: 您的 AccessKey Secret。
- 3. 在命令列中執行以下命令,建立 deployment。

kubectl create -f demo-deployment.yaml

您不需要修改demo-Service.yaml中的配置。

kubectl create -f demo-service.yaml

## 步驟 3 測試產生 Kubernetes 叢集日誌

您可以使用 kubect1 get 命令查看資來源物件部署狀況,等待 deployment 和 service 部署成功

後,執行 kubect1 get svc 查看 service 的外部存取 IP,即 EXTERNAL-IP。

\$ kubectl get svc NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE log4j-appender-demo-spring-boot-svc LoadBalancer 172.21.XX.XX 120.55.XXX.XXX 8080:30398/TCP 1h

在本樣本中,通過運行login命令來測試產生 Kubernetes 叢集日誌。其中 K8S\_SERVICE\_IP 即

為 EXTERNAL-IP。



您可以在 GitHub log4j-appender-demo 中查看完整的 API 集合。

curl http://\${K8S\_SERVICE\_IP}:8080/login?name=bruce

### 步驟 4 在阿里雲Log Service中查看日誌

### 登入 Log Service#####。

進入對應的 Project 的詳情頁面,選擇對應的日誌庫 k8s-logstore,並單擊右側的查詢分析 - 查詢, 查看 Kubernetes 叢集輸出的日誌,如下所示。

<	l誌 k8s-logstore (居于k8s	s+log4j )					分享	查询分析属性	另存为快速查询	另存为报警
标签列表	请输入关键字进行搜索 1.2				0	1分钟 ~	2018-03-06 16:3	7:50 ~ 2018-03-06	6 16:38:50	搜索
i包 k8s-logstore 新建标签 🕑	0 37:550#0 37:555	B) 38:	⊖00₩) 38⊖05₩	38631080	38公15时 38公20时 38公 日志总条数1 查询状态:结果精确	325H) 385330H	38935	₿ 38;94	OE) 38☆45	ŧŷ
	快速分析	بمع ۲ 1 Q	时间 🖛	内容 ▼ source : 172.20.	3.8					4
	您还没有指定字般查 询,赶紧添加吧(查看 帮助)			topic: level: INFO location: com aliyun.1 message: User login thread: http-nio-8080 time: 2018-03-06T08	log4jappenderdemo.UserController.login(Us i successfully. requestID=hIAgSGKGyA nam 3-exec-7 3:38+0000	serController.java:17) te=bruce				

日誌的輸出內容對應上面的命令。本樣本示範了將樣本應用產生的日誌輸出到阿里雲Log Service的 操作。通過這些操作,您可以在阿里雲上配置 Log4JAppender,並通過阿里雲Log Service,實現 日誌即時搜集、資料過濾、檢索等進階功能。

## 1.10.4 利用 log-pilot + elasticsearch + kibana 搭建 kubernetes 日誌解決方案

開發人員在面對 kubernetes 分布式叢集下的日誌需求時,常常會感到頭疼,既有容器自身特性的原因,也有現有日誌採集工具的桎梏,主要包括:

- 容器本身特性:
  - 採集目標多:容器本身的特性導致採集目標多,需要採集容器內日誌、容器 stdout。對於容器內部的檔案日誌採集,現在並沒有一個很好的工具能夠去動態發現採集。針對每種資料來源都有對應的採集軟體,但缺乏一站式的工具。
  - Auto Scaling難: kubernetes 是分布式的叢集,服務、環境的Auto Scaling對於日誌採集帶來 了很大的困難,無法像傳統虛擬機器環境下那樣,事先配置好日誌的採集路徑等資訊,採集 的動態性以及資料完整性是非常大的挑戰。
- 現有日誌工具的一些缺陷:
  - 缺乏動態配置的能力。目前的採集工具都需要事先手動設定好日誌採集方式和路徑等資訊,
     因為它無法能夠自動感知到容器的生命週期變化或者動態漂移,所以它無法動態地去配置。
  - 日誌採集重複或丟失的問題。因為現在的一些採集工具基本上是通過 tail 的方式來進行日誌 採集的,那麼這裡就可能存在兩個方面的問題:一個是可能導致日誌丟失,比如採集工具在 重啟的過程中,而應用依然在寫日誌,那麼就有可能導致這個視窗期的日誌丟失;而對於這 種情況一般保守的做法就是,預設往前多採集 1M 日誌或 2M 的日誌,那麼這就又會可能引 起日誌採集重複的問題。
  - 未明確標記日誌源。因為一個應用可能有很多個容器,輸出的應用日誌也是一樣的,那麼當 我們將所有應用日誌收集到統一日誌儲存後端時,在搜尋日誌的時候,我們就無法明確這條 日誌具體是哪一個節點上的哪一個應用程式容器產生的。

本文檔將介紹一種 Docker 日誌收集工具 log-pilot,結合 Elasticsearch 和 kibana 等工具,形成一套 適用於 kubernetes 環境下的一站式日誌解決方案。

#### log-pilot 介紹

log-Pilot 是一個智能容器日誌採集工具,它不僅能夠高效便捷地將容器日誌採集輸出到多種儲存日 誌後端,同時還能夠動態地發現和採集容器內部的記錄檔。

針對前面提出的日誌採集難題,log-pilot 通過聲明式配置實現強大的容器事件管理,可同時擷 取容器標準輸出和內部檔案日誌,解決了動態伸縮問題,此外,log-pilot 具有自動探索機制, CheckPoint 及控制代碼保持的機制,自動日誌資料打標,有效應對動態配置、日誌重複和丟失以及 日誌源標記等問題。

目前 log-pilot 在 Github 完全開源,項目地址是 *https://github.com/AliyunContainerService/log-pilot*。您可以深入瞭解更多實現原理。

#### 針對容器日誌的聲明式配置

Log-Pilot 支援容器事件管理,它能夠動態地監聽容器的事件變化,然後依據容器的標籤來進行解 析,組建記錄檔採集設定檔,然後交由採集外掛程式來進行日誌採集。

在 kubernetes 下,Log-Pilot 可以依據環境變數 aliyun\_logs\_\$name = \$path 動態地組建記錄 檔採集設定檔,其中包含兩個變數:

- \$name 是我們自訂的一個字串,它在不同的情境下指代不同的含義,在本情境中,將日誌採集
   到 Elasticsearch 的時候,這個 \$name 表示的是 Index。
- 另一個是 \$path,支援兩種輸入形式,stdout 和容器內部記錄檔的路徑,對應日誌標準輸出和容
   器內的記錄檔。
  - 第一種約定關鍵字 stdout 表示的是採集容器的標準輸出日誌,如本例中我們要採集 tomcat
     容器日誌,那麼我們通過配置標籤 aliyun.logs.catalina=stdout 來採集 tomcat 標準
     輸出日誌。
  - 第二種是容器內部記錄檔的路徑,也支援萬用字元的方式,通過配置環境變數 aliyun\_log s\_access=/usr/local/tomcat/logs/\*.log來採集 tomcat 容器內部的日誌。當然如果 你不想使用 aliyun 這個關鍵字,Log-Pilot 也提供了環境變數 PILOT\_LOG\_PREFIX 可以指定 自己的聲明式日誌配置首碼,比如 PILOT\_LOG\_PREFIX: "aliyun,custom"。

此外,Log-Pilot 還支援多種日誌解析格式,通過 aliyun\_logs\_\$name\_format=<format> 標 籤就可以告訴 Log-Pilot 在採集日誌的時候,同時以什麼樣的格式來解析日誌記錄,支援的格式包 括:none、json、csv、nginx、apache2 和 regxp。 Log-Pilot 同時支援自訂 tag,我們可以在環境變數裡配置 aliyun\_logs\_\$name\_tags="K1=V1, K2=V2",那麼在採集日誌的時候也會將 K1=V1 和 K2=V2 採集到容器的日誌輸出中。自訂 tag 可 協助您給日誌產生的環境打上 tag,方便進行日誌統計、日誌路由和日誌過濾。

## 日誌採集模式

本文檔採用 node 方式進行部署,通過在每台機器上部署一個 log-pilot 執行個體,收集機器上所有 Docker 應用日誌。

該方案跟在每個 Pod 中都部署一個 logging 容器的模式相比,最明顯的優勢就是佔用資源較少,在 叢集規模比較大的情況下表現出的優勢越明顯,這也是社區推薦的一種模式。



## 前提條件

您已經開通Container Service,並建立了一個 kubernetes 叢集。本樣本中,建立的 Kubernetes 叢 集位於華東 1 地區。

## 步驟1 部署 elasticsearch

1. 串連到您的 Kubernetes 叢集。具體操作參見通過SSH##Kubernetes## 或 ## kubectl ## Kubernetes ##。 2. 首先部署 elasticsearch 相關服務,該編排模板包含一個 elasticsearch-api 的服

務、elasticsearch-discovery 的服務和 elasticsearch 的狀態集,這些對象都會部署在 kubesystem 命名空間下。

```
kubectl apply -f https://acs-logging.oss-cn-hangzhou.aliyuncs.com/
elasticsearch.yml
```

3. 部署成功後,kube-system 命名空間下會出現相關對象,執行以下命令查看運行情況。

-n=kube-system		
TYPE	CLUSTER-IP	
	AGE	
ClusterIP	172.21.5.134	<none></none>
22h		
ClusterIP	172.21.13.91	<none></none>
22h		
DESIRED CURRE	ENT AGE	
3 3 2	22h	
	-n=kube-system TYPE ClusterIP 22h ClusterIP 22h DESIRED CURRE 3 3 2	-n=kube-system TYPE CLUSTER-IP AGE ClusterIP 172.21.5.134 22h ClusterIP 172.21.13.91 22h DESIRED CURRENT AGE 3 3 22h

### 步驟2 部署 log-pilot 和 kibana 服務

1. 部署 log-pilot 日誌採集工具,如下所示:

```
kubectl apply -f https://acs-logging.oss-cn-hangzhou.aliyuncs.com/
log-pilot.yml
```

2. 部署 kibana 服務,該編排樣本包含一個 service 和一個 deployment。

```
kubectl apply -f https://acs-logging.oss-cn-hangzhou.aliyuncs.com/
kibana.yml
```

## 步驟3 部署測試應用 tomcat

在 elasticsearch + log-pilot + Kibana 這套日誌工具部署完畢後,現在開始部署一個日誌測試應用 tomcat,來測試日誌是否能正常採集、索引和顯示。

編排模板如下。

```
apiVersion: v1
kind: Pod
metadata:
 name: tomcat
 namespace: default
 labels:
   name: tomcat
spec:
 containers:
  - image: tomcat
   name: tomcat-test
    volumeMounts:
    - mountPath: /usr/local/tomcat/logs
     name: accesslogs
   env:
     - name: aliyun_logs_catalina
```

value: "stdout"	##
採集標準輸出日誌	
- name: aliyun_logs_access	
value: "/usr/local/tomcat/logs/catalina.*.log"	##
採集容器內記錄檔	
volumes:	
- name: accesslogs	
emptyDir: {}	

tomcat 鏡像屬於少數同時使用了 stdout 和檔案日誌的 Docker 鏡像,適合本文檔的示範。在上面的 編排中,通過在 pod 中定義環境變數的方式,動態地組建記錄檔採集設定檔,環境變數的具體說明 如下:

- aliyun\_logs\_catalina=stdout表示要收集容器的 stdout 日誌。
- aliyun\_logs\_access=/usr/local/tomcat/logs/catalina.\*.log 表示要收集容器內 /usr/local/tomcat/logs/目錄下所有名字匹配 catalina.\*.log 的檔案日誌。

在本方案的 elasticsearch 情境下,環境變數中的 <u>\$name</u> 表示 Index,本例中 <u>\$name</u>即是 catalina 和 access 。

### 步驟 4 將 kibana 服務暴露到公網

上面部署的 kibana 服務的類型採用 NodePort,預設情況下,不能從公網進行訪問,因此本文檔配 置一個 ingress 實現公網訪問 kibana,來測試日誌資料是否正常索引和顯示。

 通過配置 ingress 來實現公網下訪問 kibana 。本樣本選擇簡單的路由服務來實現,您可參 考Ingress ## 擷取更多方法。該 ingress 的編排模板如下所示。

apiVersion: extensions/vlbetal kind: Ingress metadata: name: kibana-ingress namespace: kube-system 於同一個 namespace	#要與	kibana	服務處
spec:			
rules:			
- http:			
paths:			
- path: /			
backend:			
serviceName: kibana	#輸入	kibana	服務的
名稱			
servicePort: 80	#輸入	kibana	服務暴
露的連接埠			

2. 建立成功後,執行以下命令, 擷取該 ingress 的訪問地址。

\$	kubectl	get	ingress	-n=kube-system			
1	JAME		HOSTS	ADDRESS	PORTS	AGE	
ŝ	shared-dn	s	*	120.55.150.30	80	5m	

3. 在瀏覽器中訪問該地址,如下所示。

← -	○ C ▲ 不安全	https://1	scover?_g=()&_a=(columns:!(_source),index:**,interval:auto,query:(query_string:( 🛚 🛧 💁 😵 🔘 🔅 🚦
	kibana	0 hits	New Save Open Share < 🛛 Last 15 minutes 🗲
	KIDalla	Search (e.g. status:200 AND exter	sion:PHP) Uses lucene query syntax
Ø	Discover	Add a filter 🕇	
Ш		* • •	
3		Selected Fields	No results found 🙂
8		? _source	Unfortunately I could not find any results matching your search. I tried really hard. I looked all over the place and frankly, I just couldn't find anything good. Help me, help you. Here are some
۶		Available Fields	ideas:
Ф			Expand your time range
			I see you are looking at an index with a date field. It is possible your query does not match anything in the current time range, or that there is no data at all in the currently selected time range. Click the button below to open the time picker. For future reference you can open the time picker by clicking on the Otime picker button in the top right corner of your screen.
			Refine your query

4. 單擊左側導覽列中的management ,然後單擊Index Patterns > Create Index Pattern。具體 的索引名稱會在 \$name 變數尾碼一個時間字串,您可以配合萬用字元 \* 進行建立。本例中使用 \$name \* 來建立 Index Pattern。

您也可以執行以下命令,進入 elasticsearch 對應的 pod,在 index 下列出 elasticsearch 的所有 索引。

\$ kubectl get <u>p</u> <b>對應的</b> pod	pods -n=ku	lbe-system	#	找到 elas	stic	search
\$ kubectl exec elasticsearch	-it elas 的一個 pod	sticsearch-1	l bash	#進ノ		
 \$ curl 'localho 引	ost:9200/_	_cat/indices	5?v'	#‡	⊧ 列出	出所有索
health status	index		uuid		pri	rep
docs.count docs	s.deleted	store.size	pri.store.size			
green open	.kibana		x06jj19PS4Cim6A	jo51PWg	1	1
4	0	53.6kb	26.8kb			
green open	access-20	18.03.19	txd3tG-NR6-guqm	MEKKzEw	5	1
143	0	823.5kb	411.7kb			
green open	catalina-	2018.03.19	ZgtWd16FQ7qqJNN	WXxFPcQ	5	1
143	0	915.5kb	457.5kb			

5. 索引建立完畢後,單擊左側導覽列中的Discover,然後選擇前面建立的 Index,選擇合適的時間 段,在搜尋欄輸入相關欄位,就可以查詢相關的日誌。



至此,在阿里雲 Kubernetes 叢集上,我們已經成功測試基於 log-pilot、elasticsearch 和 kibana 的 日誌解決方案,通過這個方案,我們能有效應對分布式 kubernetes 叢集日誌需求,可以協助提升營 運和運營效率,保障系統持續穩定運行。

## 1.10.5 Kubernetes與Log Service整合與使用

阿里雲Container ServiceKubernetees叢集整合了Log Service(SLS),您可在建立叢集時啟用Log Service,隨後只要對應用進行簡單配置,即可無縫使用阿里雲Log Service。

前提條件

請確保開通了阿里雲Log Service。

## 步驟1 安裝日誌組件

- 1. 登入 Container Service#####。
- 在 Kubernetes 菜單下,單擊左側導覽列的叢集,進入叢集列表頁面,單擊右上方建立 Kubernetes叢集。

日志服务	✔ 使用 SLS			
	使用已有 Project	创建新 Project	to by advantific provide the second	-

 在建立叢集頁面,選擇Kubernetes叢集模式。勾選使用SLS,您可選擇已有日誌project;或者 直接建立一個,預設建立名稱為k8s-log-{ClusterID}的project。其他叢集配置參數,請 參見##Kubernetes##。

日志服务	✔ 使用 SLS	
	使用已有 Project	创建新 Project
	将自动创建名称为 k8s-	log-{ClusterID} 的 Proj

- 4. 最後單擊右上方建立叢集,啟動部署,等叢集部署完成。
- 5. 單擊左側導覽列中應用 > 部署,進入部署列表,選擇所需的叢集和kube-system命名空間,您可 看到alibaba-log-controller組件為運行狀態。

容器服务 - Kubernetes ▼	集群列表				燈	?最多可以创建 5 个集群,€	每个集群最多可以添加 40 个节点	刷新	创建 Kubernetes 集群
概选	常见问题: ② 如何创建集群 ② 扩容和缩	諸集群 🛛 ② 通过 kubectl 注	接 Kubernetes 集群	∂ 通过命令管理应用					
▼ 集群	名称 ▼								
集群	集群名称/ID	集群类型	地域 (全部) 👻	网络类型	集群状态	创建时间	Kubernetes 版本		操作
节点	logtest-cluster	Kubernetes	华东1	虚拟专有网络 vpc-bp1e67w2gx5	●运行中	2018-09-20 16:12:26	1.11.2	管理	- 查看日志

6. 進入Log Service管理主控台,您可看到日誌project被建立出來。

	应用基本信息	容器配置	>	高级配置	$\rightarrow$	创建完成
应用名称	nginx					
	名称为1-64个字符,可包含数字、小写	英文字符,或"-",且不能以-开头				
部署集群:	logtest-cluster	v				
命名空间:	default	T				
副本数量:	2					
类型	无状态	•				
						返回 下一步

## 步驟2 在部署頁配置採集資訊

- 1. 登入 Container Service#####。
- 在Kubernetes菜單下,單擊左側導覽列中的應用 > 部署,進入部署列表頁面,單擊右上方使用 鏡像建立。
- 進入應用基本資料頁面,配置應用基本資料,包括應用程式名稱、部署叢集和命名空間,最後單 擊下一步。

	日志服务:	注意:请确保集群已部署[日志插件]		
		● 采集配置		
		日志库	容器内日志路径 (可设置为stdout)	
		access	/usr/local/tomcat/logs/catalina.*.log	•
開始		catalina	stdout	•
在應用配置頁面,首先配置鏡像名稱、鏡像版本、容器數量。本例中建立一個tomcat應用,來測 試收集容器標準輸出日誌,以及容器內文本日誌。

说明:

tomcat 鏡像屬於少數同時使用 stdout 和檔案日誌的容器鏡像,適合本例中的示範。

Tan名称	Tan值	
- gann		
release	1.0.0	e

5. 配置資料卷,將容器內的日誌目錄掛載到臨時目錄,用於儲存文本日誌。



<		回Project列表						地域:华东1
	Logstore列表					学习	路径 查看End	point创建
<ul> <li>LogHub - 实时采集</li> <li>[文档] 接入指南</li> </ul>	请输入Logstore名进行模糊器 搜索							
Logtail配置	Lonstone文字	新捉達入南导	吃肉	日志采集構式		日志消费模式		绿作
Loctal M Wills	LUGSTOLEEN	\$20013277P3+3	mlx	HANTWOOD	日志消费	日志投递	查询分析	28(1)-
▼ LogHub - 实时消费	access	8	Ľ	Logtail配置(管理)  诊断   更多 -	预览	MaxCompute   OSS	查询	修改 删除
[文档] 消费指南	another-logstore-name	8	⊵	Logtail配置(管理)  诊断   更多 -	预览	MaxCompute   OSS	查询	修改 删除
消费组管理 ▼ Search/Applytics -	audit- c90e6551305c74a97b6184f783ed5486b	8	⊭	Logtall配置(管理)  诊断   更多 -	预览	MaxCompute   OSS	查询	修改 删除
· SearchyAnalytics 快速查询	catalina	8	Ł	Logtail配置(管理)  诊断   更多 →	预览	MaxCompute   OSS	查询	修改 删除

6. 配置Log Service,您可進行採集配置和自訂Tag設定。

🗟 catalina (雇于k8s-log-c	c90e6551305c74a97b	6184f783ed5486b )		①1小时(相对) 🔻	分享	查询分析属性	另存为快速查询	另存为告警
1 1.0.0							0	搜索
48 0 16801分	16时08分	16尉15会	1687259; 1687259; 1687569;	16时43分		16时20分	16时57分	
			日志总条数:44 查询状态:结果精确					
原始日志统计	图表						列访	25 🗍
快速分析	<	时间 🔺	内容 🔻					
(四五役有指定字校畫 间,赶紧然加旺(宣音 帮助)	1 Q	09-28 16:11:46	source 172.20.1.130 tag ::hostname: logtal-ds-twggx topic _container_jb: 172.20.1.135 containername: tomcat imagename: tomcat@sha256.8d120de5102cc12310de741299e9bf namespace_: celfault podname: tomcatPc756c746-d655v poduid: 084fe1ahc2611e8-9734-00163e00b2ae0 source_: stderr ttme_: 2018-09-28108.11:21.1609930092 content: 28-Sep-2018 08.11:21.159 INFO [main] org.apache.catalina.st release: 10.0	5139d2d67466312ce4e6f0f18 artup.Catalina.start Server s	31ccfeebe tartup in 7	7 '30 ms		
	2 Q	09-28 16:11:46	source 172.20.1.130 taghostname: logtall-ds-twggx toptal container_ip: 172.20.1.135 container_iname: tomcat@sha256.8d120de5102cc12310de741299e9bf namespace_: default podname: tomcatPc756c746-d655v pod_uid: 08f4e1ab-25f1e8-9734-00163e0b2ae0 source_: stder ttme: 2018-09-28108.11:21.157085805Z content: 28-5ep-2018 08.11:21.156 INFO (main) org.apache.coyote.Abs release: 10.0	5139d2d67466312ce4e6f0f18 stractProtocol.start Starting I	31ccfeebe ProtocolH	7 andler ("ajp-nio-800	19"]	

您可對日誌進行採集配置:

• 日誌庫:即在Log Service中產生一個對應的logstore,用於儲存採集到的日誌。

- 容器內日誌路徑:支援stdout和文本日誌。
  - stdout: stdout 表示採集容器的標準輸出日誌。
  - — 文本日誌:表示收集容器內指定路徑的日誌,支援萬用字元的方式,本例中表示收集符合/
     usr/local/tomcat/logs/catalina.\*.log名稱規則的文本日誌。

您還可設定自訂 tag,設定tag後,會將該tag一起採集到容器的日誌輸出中。自訂 tag 可協助您 給容器日誌打上tag,方便進行日誌統計和過濾等分析操作。

- 7. 完成應用配置後,單擊下一步。
- 8. 進入訪問設定頁面,本例中不進行訪問設定,單擊建立,頁面提示建立成功。

	✓ 创建应用任务	提交成功		
创建部署	tomcat-o	deployment	成功	
	查看应用详情	继续创建		

9. 進入Log Service控制台,在對應的日誌project中,您可看到成功建立的logstore。選擇所需的 logstore,單擊查詢。

<		▲返回Project列表	ž					地域:华北2
日志库	Logstore列表						查看Endpoint	创建
▼ LogHub - 实时采集		10.1						
[文档] 接入指南	请输入Logstore名进行模糊查试	9 搜索						
Logtail配置	Logestoro 名初	新提该入内导	防約	日本立傳練ゴ		日志消费模式		過作
	Logstore	8739326-226-344	III / Prive Diff.	日志消费	日志投递	查询分析	DRTF	
Logtall机器组	access	8	⊵	Logtail配置 (管理)  诊断   更多 <del>-</del>	预览	MaxCompute   OSS	查询	修改   删除
▼ LogHub - 头时消费 (立地) 消费指索	config-operation-log	8	⊵	Logtali配置 (管理)  诊断 更多 <del>-</del>	预览	MaxCompute   OSS	查询	修改   删除
消费组管理	catalina		⊭	Logtali配置 (管理)  诊断 更多 マ	预览	MaxCompute   OSS	查询	修改   删除
▼ Search/Analytics - 查						共有3条, 每页显示: 10		1 > >
快速查询								

10.頁面跳轉到日誌列表頁,選擇日誌查詢時間,每條日誌都被打上了tag app=tomcat。

<	퀎 access	
捜索 Q	B access (馬干/ds-to_tests (m)) ▼ 分享 査询分析属性 另存为快速查询 另存	7为告警
> 日志库 協	1 0 28	8
access	48	
config-operation-log	0 04992789 06991589 08991589 10991589 12991589 14991589 16991589 18991589	
> 快速查询 Q	日志总条数-44 查询状态结果精确 画始日志 统计图表	
> 収表盛 [1]	快速分析           内容 ▼	÷
	地区没有指定字段度彻。赶紧 荡动地은(食有和法))         1 ② 08-08 18:15:19	
	2 Q 08-06 18:15:19 <u>source172.16.1.131</u> tagbetweet_defined_id K85-group-c38eb05c07697450aadc68cc348ce tagbetwtwis_cat/homeat/logs/catalina_2018-08-08.log tagbetweet_defined_id K85-group-c38eb05c07697450aadc68cc348ce tagbetweet_defined_id K85-group-c38eb05c07697450aadc68cc348ce tagbetweet_defined_id K85-group-c38eb05c07697450aadc68cc348ce tagbetweet_defined_id K85-group-c38eb05c07697450aadc68cc348ce tagbetweet_and tag intege_name torocate tagbetweet_and tag intege_name torocatege_name torocatege_name torocatege_name torocatege_name torocatege_name torocatege_name torocatege_name torocatege_name torocatege_nam torocatege_nam	咨询,建议

# 1.11 監控管理

# 1.11.1 與Cloud Monitor整合與使用

前提條件

請先檢查kube-system命名空間下是否已經部署了alicloud-monitor-controller,若未部

署,請進行舊版本叢集升級。

使用方式

- 1. 登入 Container Service#####。
- 2. 在Kubernetes菜單下,單擊左側導覽列中的部署,進入部署列表頁面。
- 3. 選擇所需的deployment,單擊右側的監控,或者在內建的Kubernetes Dashboard的部署頁面中 單擊監控。

容器服务 - Kubernetes ▼	部署列表			刷新	使用镜像创建	使用模板创建
概赏	集群 KBS-cs v 命名空间 kube-system v 2					
▼ 集群	各称    标签	容器组数量	创建时间			3 操作
集群	alicloud-disk-controller app:alicloud-disk-controller	1/1	2018-06-27 17:53:56		详情 🏼 编	續 监控 更多▼
节点	alicloud-monitor-controller k8s-app:alicloud-monitor-controller task:monitoring	1/1	2018-06-28 14:06:30		详情   編	蟲  监控 更多▼
命名空间	default-http-backend app:default-http-backend	1/1	2018-06-27 17:53:56		详情   编	輯  监控 更多▼
▼ 应用 <b>1</b>	heapster k8s-app:heapster task:monitoring	1/1	2018-06-27 17:53:56		详情   編	攝  监控 更多▼
部署	kube-dns k8s-app:kube-dns	1/1	2018-06-27 17:53:51		洋情 丨 编	巖  监控 更多▼

<	➡ 工作负载 部署					十 创建
集群	部署					Ŧ
命名空间	名称 🔷	标签	容器组	已创建 🔶	镜像	
<sup>17 点</sup> 持久化存储卷	monitoring-grafana	k8s-app: grafana task: monitoring	1/1	2018-07-10 16:04:34	registry.cn-hangzhou.aliyuncs.com/acs,	:
角色	alicloud-monitor-controller	k8s-app: alicloud-monitor-controller task: monitoring	1/1	2018-06-28 14:06:30	registry.cn-hangzhou.aliyuncs.com/acs,	:
\$775B	V tiller-deploy	app: helm name: tiller	1/1	2018-06-27 17:53:58	registry-vpc.cn-hangzhou.aliyuncs.com	3:
kube-system	alicloud-disk-controller	app: alicloud-disk-controller	1/1	2018-06-27 17:53:56	registry-vpc.cn-hangzhou.aliyur 监控	
概況	leapster	k8s-app: heapster task: monitoring	1/1	2018-06-27 17:53:56	registry.cn-hangzhou.aliyuncs.c 伸缩	
工作负载	monitoring-influxdb	k8s-app: influxdb task: monitoring	1 / 1	2018-06-27 17:53:56	registry-vpc.cn-hangzhou.aliyur 删除	
~~	oginx-ingress-controller	app: ingress-nginx	1/1	2018-06-27 17:53:56	查看/ registry-vpc.cn-hangzhou.aliyur	/编辑 YAML
部署 2	default-http-backend	app: default-http-backend	1/1	2018-06-27 17:53:56	registry-vpc.cn-hangzhou.aliyuncs.com	:

此時會跳轉到Cloud Monitor的相應的監控視圖頁面。

<	k8s-c <sup>1</sup> Control Con						
组内资源	1///时 2//时 6//时 12//时 1开 3开 7开 14开 ( 自动脱游・						
● 监控视图	建度: 分组 实例						

### 4. 應用分組支援分組和執行個體兩個維度監控。





### 5. 如需警示設定,分組層級的指標以group開頭,執行個體層級的指標以pod開頭。

产品:	容體服务Kubernetes版 ▼	
资源范围:	应用分组	击 查看报警模板最佳实践
分组:	k8s-c07ea7a5639924917af756e7d3f95e 💌	
设置报警规则		
使用模板:	◎ 是 ⑧ 否	1.00
规则名称:	Deployment CPU利用率监控	
规则描述:	group.cpu.usage_rate 🔺 S分钟 🔹 总计 💌 >= 💌 調報	<u>ā</u>
		0.00
十添加报警规	group.cpu.limit	-0.50
通道沉默时	group.cpu.request	
	group.cpu.usage	-1.00 14:10:00 14:15:00 14:23:20 14:31:40 14:40:00 14:50:0(
通信后报警:	group.cpu.usage_rate	● group.cpu.usage_rate—Sum—应用分组—152343
	group.disk.io_read_bytes	

### 舊版本叢集升級

- 1. 登入 Container Service#####。
- 在Kubernetes菜單下,單擊左側導覽列中的應用 > 部署,進入部署列表頁面,單擊右上方的使 用模板建立。

容器服务 - Kubernetes ▼	部署列表			刷新使用镜像创建	使用模板创建
概览	集群 k8s-test ▼ 命名空间 default	¥			2
▼ 集群	名称    标签	容器组数量	创建时间		操作
集群					
节点		① 没有查	间到符合条件的记录		
存储					
命名空间					
▼ 应用 部署					

3. 選擇所需的叢集,Kube-system命名空間,使用以下的樣本模板,然後單擊建立。

☐ 说明:	
根據自己的叢集替換REGION與CLUSTER_ID,並重新部署Heapster的yaml編排。	

集群	k8s-test v	
命名空间	kube-system r	
示例模版	自定义 🔹	
欄版	<pre>1 apiVerSin: extensions/vibeta1 2 kind: Deployment 3 - metadata: 4 name: heapster 5 namespace: kube-system 6 - spec: 7 replicas: 1 8 - template: 9 - metadata: 10 - labels: 11 task: monitoring 12 k0s-app: heapster 13 - annotations: 14 scheduler.alpha.kubernetes.io/critical-pod: '' 15 - spec: 16 serviceAccount: admin 17 containers: 18 name: heapster 19 inage: registry.cn-hangzhou.aliyuncs.com/acs/heapster-amd64:v1.5.1.1 10 inagePulPolicy: ThOPPeent 21 - /meapster 23sourceskubernetes:https://kubernetes.default 24historical-sourcesinfluxdb:http://monitoring.influxdb:8086 25sink=socket:tcp://monitor.csk.cn-hangzhou.aliyuncs.com/80932clusterId= 24 pipublic=true 25 26sink=socket:tcp://monitor.csk.cn-hangzhou.aliyuncs.com/80932clusterId= 26 pipublic=true 27 28 29 29 29 29 29 29 20 20 20 20 20 20 20 20 20 20 20 20 20</pre>	
	65 <b>8</b>	

heapster樣本編排模板如下。若叢集中已有舊版本的heapster,您也可登入到Kubernetes叢集,

```
執行kubectl apply -f xxx.yaml命令進行更新。
```

```
apiVersion: extensions/vlbetal
kind: Deployment
metadata:
 name: heapster
 namespace: kube-system
spec:
  replicas: 1
  template:
    metadata:
      labels:
        task: monitoring
        k8s-app: heapster
      annotations:
        scheduler.alpha.kubernetes.io/critical-pod: ''
    spec:
      serviceAccount: admin
      containers:
      - name: heapster
       image: registry.##REGION##.aliyuncs.com/acs/heapster-amd64:
v1.5.1.1
        imagePullPolicy: IfNotPresent
        command:
        - /heapster
        - -- source=kubernetes: https://kubernetes.default
        - --historical-source=influxdb:http://monitoring-influxdb:
8086
        - --sink=influxdb:http://monitoring-influxdb:8086
```

- --sink=socket:tcp://monitor.csk.##REGION##.aliyuncs.com: 8093?clusterId=##CLUSTER\_ID##&public=true

alicloud-monitor-controller 的樣本編排如下,執行kubect1 create -f xxx.yaml命令部署 alicloud-monitor-controller。

```
apiVersion: extensions/vlbetal
kind: Deployment
metadata:
 name: alicloud-monitor-controller
 namespace: kube-system
spec:
 replicas: 1
  template:
    metadata:
      labels:
        task: monitoring
        k8s-app: alicloud-monitor-controller
      annotations:
        scheduler.alpha.kubernetes.io/critical-pod: ''
    spec:
      hostNetwork: true
      tolerations:
      - effect: NoSchedule
        operator: Exists
       key: node-role.kubernetes.io/master
      - effect: NoSchedule
        operator: Exists
        key: node.cloudprovider.kubernetes.io/uninitialized
      serviceAccount: admin
      containers:
      - name: alicloud-monitor-controller
        image: registry.##REGION##.aliyuncs.com/acs/alicloud-monitor
-controller:v1.0.0
        imagePullPolicy: IfNotPresent
        command:
        - /alicloud-monitor-controller
        - agent
        - --regionId=##REGION##
        - --clusterId=##CLUSTER ID##
        - --loqtostderr
        - --v=4
```

 更新完畢後,進入Kubernetes 控制台,在kube-system命名空間中,可看到這兩個Deployment 處於運行中,即升級完畢。

集群	kube-flannel-ds	app: flannel tier: node	4 / 4	2018-05-28 17:08:49	registry-vpc.cn-hangzhou.aliyuncs.com/acs/flannel:v0.8.0 registry-vpc.cn-hangzhou.aliyuncs.com/acs/flannel:v0.8.0	e :
命名空间						
节点	部署					Ŧ
持久化存储器	毎時 令	标签	容髓组	日日建 🗢	領領	
角色	alicloud-monitor-controller	k8s-app: alicloud-monitor-controller task: monitoring	1/1	2018-06-26 17:10:32	registry.cn-hangzhou.aliyuncs.com/acs/alicloud-monitor-contr-	
存储类	S albaba-log-controller	kubernetes.lo/cluster-service: true	1/1	2018-06-22 17:27:08	registry.cn-hangzhou.aliyuncs.com/log-service/alibabacloud-lc	:
命名空间	S tiller-deploy	app: helm name: tiller	1/1	2018-05-28 17:08:52	registry-vpc.cn-hangzhou.aliyuncs.com/acs/tiller:v2.8.2	1
kube-system *	alicloud-disk-controller	app: alicloud-disk-controller	1/1	2018-05-28 17:08:49	registry-vpc.cn-hangzhou.aliyuncs.com/acs/alicloud-disk-contr	÷
概况	default-http-backend	app: default-http-backend	1/1	2018-05-28 17:08:49	registry-vpc.cn-hangzhou.allyuncs.com/acs/defaultbackend:1.	:
工作负载	leapster	kBs-app: heapster task: monitoring	1/1	2018-05-28 17:08:49	registry.cn-hangzhou.aliyuncs.com/acs/heapster-amd64:v1.5.	:

對於不清楚自己REGION資訊的開發人員,可以通過如下的方式快速查詢,開啟ECS控制台,選擇 自己叢集所在的地區,頁面地址URL中最後一段即是REGION。

G	● 安全	https://ecs	console.aliyun.com/#/server/region/cn-hangzhou						
管理	控制台	产品与服务	▼	搜索		Q	🌲 <mark>24</mark>	费用	工单
云月	踌器 ECS		2 ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) ( ) (						
根	题 例		▼ 选择实例属性项搜索,或者输入关键字识别搜索		Q	标签			

# 1.11.2 使用 Grafana 展示監控資料

前提条件

- 您已經成功部署一個 Kubernetes 叢集,參見##Kubernetes##。
- 本樣本使用的 Grafana 的鏡像地址是 registry.cn-hangzhou.aliyuncs.com/acs/ grafana:5.0.4,內建了相關監控模板。

#### 背景信息

在 kubernetes 的監控方案中,Heapster+Influxdb+Grafana 的組合相比 prometheus 等開源方案而 言更為簡單直接。而且 Heapster 在 kubernetes 中承擔的責任遠不止監控資料的採集,控制台的監 控介面、HPA的 POD Auto Scaling等都依賴於 Heapster 的功能。因此 Heapster 成為 kubernetes 中一個必不可少的組件,在阿里雲的 Kubernetes 叢集中已經內建了 Heapster+Influxdb 的組合,如 果需要將監控的資料進行展示,只需要配置一個可用的 Grafana 與相應的 Dashboard 即可。

#### 操作步骤

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的應用 > 部署,進入部署列表頁面。
- 3. 單擊頁面右上方的使用模板建立。

容器服务 - Kubernetes ▼	部署列表		刷新	使用镜像创建使用模板创建
概览	集群 k8s-cluster ▼ 命名空间 default	Ŧ		2
▼ 集群	名称    标签	容器组数量	创建时间	攝作
集群		0.554		
点		① 没有	查询到符合条件的记录	
存储卷				
命名空间				
▼ <u>应用</u> 1				
部署				

- 4. 對模板進行相關配置,部署 Grafana 的 deployment 和 service,完成配置後,單擊建立。
  - 叢集:選擇所需的叢集。
  - 命名空間:選擇資來源物件所屬的命名空間,必須是kube-system。

樣本模板:本樣本選擇自訂模板,其中包含一個 deployment 和 service。



#### 本樣本的編排模板如下。

```
apiVersion: extensions/vlbetal
kind: Deployment
metadata:
  name: monitoring-grafana
   namespace: kube-system
spec:
   replicas: 1
   template:
     metadata:
       labels:
         task: monitoring
         k8s-app: grafana
     spec:
       containers:
       - name: grafana
         image: registry.cn-hangzhou.aliyuncs.com/acs/grafana:5.0.4
         ports:
         - containerPort: 3000
           protocol: TCP
         volumeMounts:
         - mountPath: /var
           name: grafana-storage
         env:
         - name: INFLUXDB_HOST
           value: monitoring-influxdb
       volumes:
       - name: grafana-storage
         emptyDir: {}
apiVersion: v1
kind: Service
```

```
metadata:
    name: monitoring-grafana
    namespace: kube-system
spec:
    ports:
    - port: 80
        targetPort: 3000
    type: LoadBalancer
    selector:
        k8s-app: grafana
```

5. 完成部署後,返回部署頁面,選擇所需叢集,然後選擇 kube-system,查看其下部署的應用。

突器服务 - Kubernetes ★	部署列表				刷新使用镜像创建使用模板创建
概览	集群 k8s-cluster ▼ 命名空间	kube-system 🔻 🙎			
▼ 集群	名称	标签	容器组数量	创建时间	1日 1日 1日
集群	alicloud-disk-controller	app:alicloud-disk-controller	1/1	2018-06-27 17:53:56	洋情 │ 编辑 │监控│更多▼
节点 存储类	alicloud-monitor-controller	k8s-app:alicloud-monitor-controller task:monitoring	1/1	2018-06-28 14:06:30	详情   编辑  监控 更多▼
命名空间	default-http-backend	app:default-http-backend	1/1	2018-06-27 17:53:56	详情   编辑  监控 更多▼
▼应用	heapster	k8s-app:heapster task:monitoring	1/1	2018-06-27 17:53:56	详情   编辑  监控 更多▼
部署	kube-dns	k8s-app:kube-dns	1/1	2018-06-27 17:53:51	详情   编辑  监控 更多▼
容器组 服务	monitoring-grafana	k8s-app:grafana task:monitoring	1/1	2018-07-10 16:04:34	详情   编辑  监控 更多▼
路由	monitoring-influxdb	k8s-app:influxdb task:monitoring	1/1	2018-06-27 17:53:56	洋情   编辑  监控 更多▼

6. 單擊 monitoring-grafana 的名稱,查看部署狀態,等待運行狀態變為 running。

创建monitoring-grafana 🔒	返回列表			刷新
基本信息				
名称:	monitoring-grafana			
命名空间:	kube-system			
创建时间:	2018-04-25T07:32:02Z			
标签:	k8s-app:grafana task:monitoring			
注解:	deployment.kubernetes.io/revision:1			
选择器:	k8s-app:grafana task:monitoring			
策略:	RollingUpdate			
状态:	已更新:1个,不可用:0个,共计:1个			
容器组 关联服务				
名称		状态	<b>鏡</b> 像	事件
monitoring-grafana-675dc8448c-nqj	4s	Running	registry.cn-hangzhou.aliyuncs.com/acs/grafana:5.0.4	

 7. 單擊左側導覽列中的應用 > 服務,進入服務列表,選擇所需的叢集和命名空間(kubesystem),查看外部端點。

這個地址是通過 LoadBalancer 類型的 service 自動建立的,對於要求更安全存取原則的開發人員而言,建議考慮添加 IP 白名單或者使用配置認證等方式增強安全性。

容器服务 - Kubernetes ▼	服务列表							刷新	创建
概览	集群 k8s-cluster	▼ 命名空间	kube-system 🔹 3						
▼ 集群	名称	类型	创建时间	集群IP	内部端点	外部端点			操作
集群	default-http-backend	ClusterIP	2018-06-27 17:53:56		default-http-backend:80 TCP		详情   更新	查看YAML	删除
节点	heapster	ClusterIP	2018-06-27 17:53:56	10.00	heapster:80 TCP	-	详情   更新	查看YAML	删除
存储卷 命名空间	kube-dns	ClusterIP	2018-06-27 17:53:51		kube-dns:53 UDP kube-dns:53 TCP	-	详情   更新	查看YAML	删除
▼ 应用 1	monitoring-grafana	LoadBalancer	2018-07-10 16:04:34	10.00	monitoring-grafana:80 TCP monitoring-grafana:32746 TCP	80	详情   更新	查看YAML	删除
部署	monitoring-influxdb	ClusterIP	2018-06-27 17:53:56	10.000	monitoring-influxdb:8086 TCP		详情   更新	查看YAML	删除
容器组 2 三	nginx-ingress-lb	LoadBalancer	2018-06-27 17:53:56	-	nginx-ingress-lb:80 TCP nginx-ingress-lb:30883 TCP nginx-ingress-lb:443 TCP nginx-ingress-lb:23280 TCP	80	详情   更新	查看YAML	删除

8. 選擇 monitoring-grafana 服務,單擊右側的外部端點,登入 Grafana 監控介面。

預設的 Grafana 的使用者名和密碼都是 admin,建議在登入後先修改為更複雜的密碼。



9. 您可選擇內建的監控模板 ,查看 Pod 和 Node 的監控 Dahsboard。

本樣本使用的 Grafana 版本內建了兩個模板,一個負責展示節點層級的實體資源,一個負責展示 Pod 相關的資源。開發人員也可以通過添加自訂的 Dashboard 的方式進行更複雜的展現,也可以基於 Grafana 進行資源的警示等。

Ó	Kube	rnetes No	de监控 ◄		
	node_name	cn-hangzho	u.i-t	J.	
+	🗸 Dashboa	ard Row			
Ļ		Uptime		C	PU Cores
*	1.	<b>84 da</b>	<b>y</b>		2
	✓ History				
	1.0 B			Filesy	stem Available
	0.5 B			Na	data pointo
	0 B ———				
	-0.5 B				
	-1.0 B	15:00	15	5:10	15:20
				Mem	ory Utilization
	92.500%				
	92.000%				
	91.000%				

91.000%

90.500%

👪 Kubernetes Pod监控 🗸			11.1× 🕸 🕅	* * < Q > (	DLast 1 hour Refresh every 5s C
namespace default - pod_name	test-wordpress-6945bc9556-mpdxn 🔻				
✓ Status					
Uptime	CPU Usage Rate	Memory Usage	Network Tx Rate	Network Rx Rate	
2.07 day	-(4-)	181 MB	1.815 kBps	7.61 kBps	
✓ Resource Limit and Request →	¢ 🖞				
CPU Limit	CPU Request	Memory Limit	Memory Request		
0	300	0 B	537 MB		
↓ History					
4.5	CPU Usage Rate		182.5 MB	Memory Usage	
			182.0 MB		
3.5			181.5 MB		- A A
2.5 15:00 15:10 — cpu/usage_rate.mean	15:20 15:30	15:40 15:50	181.0 MB 15:00 15 — memory/usage.mean	i10 15:20 15:30	15:40 15:50

# 1.11.3 使用HPAAuto Scaling容器

阿里雲Container Service支援在控制台介面上快速建立支援HPA的應用,實現容器資源的Auto Scaling。您也可通過定義HPA(Horizontal Pod Autoscaling)的yaml配置來進行配置。

前提條件

- 您已成功建立一個Kubernetes叢集,參見##Kubernetes##。
- 您已成功串連到Kubernetes叢集的Master節點。

#### 方法1 通過Container Service控制台建立HPA應用

在阿里雲Container Service中,已經整合了HPA,開發人員可以非常簡單地通過Container Service 控制台進行建立。

- 1. 登入 Container Service#####。
- 2. 在Kubernetes菜單下,單擊左側導覽列中的應用 > 部署,單擊右上方的使用鏡像建立。

容器服务 - Kubernetes ▼	部署列表					剧新	使用镜像创建	使用模板创建
概览	集群 k8s-cluster v 命名空间	default 🔻					2	
▼ 集群								
集群	47.944	4	<b>六明/2巻</b> 月		4/78e+1/5]			4日.//-
节点	白你	9702	會輸出設建		62%F13161			1981 F
存储卷				① 没有查询到符合条件的记录				
命名空间								
▼ <u> 应用</u> 新署								

- 3. 填寫應用的名稱,設定應用部署叢集和命名空間,單擊下一步。
- 4. 首先進行應用設定,設定副本數量,然後勾選開啟自動調整,設定伸縮的條件和配置。

- 指標:支援CPU和記憶體,需要和設定的所需資源類型相同。
- 觸發條件:資源使用率的百分比,超過該使用量,容器開始擴容。
- 最大容器數量:該Deployment可擴容的容器數量上限。
- 最小容器數量:該Deployment可縮容的容器數量下限。

应用配置		
副本数量:	1	
自动伸缩:	☑ 开启	
	指标: CPU使用量 v	
	触发条件: 使用量 70 %	
	最大副本数: 10 可选范围:2-100	
	最小副本数: 1 可选范围:1-100	

5. 進行容器設定,選擇鏡像,並設定所需的資源。然後單擊下一步

道 说明:	
您必須為Deployment設定所需資源,	否則無法進行容器自動調整。

con	atainer0				
	镜像名称:	nginx	镜像版本:	latest 总是拉取镜像	
		选择镜像		选择镜像版本	
悲本配置	资源限制:	CPU 如:500m			
-	所需资源:	CPU 500m 0 内存 如:128Mi			
	Init Container				

6. 進入訪問設定頁面,本例中不進行訪問設定,單擊建立。

此時一個支援HPA的Deployment就已經建立完畢,您可在部署的詳情中查看伸縮組資訊。

基本信息	ļ.										
名称:		nginx-deployment									
命名空间:		default									
创建时间:		2018-08-23 10:07:23									
标签:		app:nginx									
注解:		deployment.kubernetes.io/revision:1									
选择器:	选择器: app:nginx										
策略:	策略: RollingUpdate										
状态:	状态: 已更新:1个,不可用:1个,共计:1个										
触发器	約分器 1.每件学型的转发器只能创建1个 のほきが決落 へ										
目前没有任何常	触发器,点击右上角按钮	创建触发器									
CPU				Memory							
		当前暂无监	語空数据		当前	暂无监控数据					
容器组	访问方式 事件	容器组水平伸缩器									
容 <b>器</b> 组 名称	访问方式 事件	<b>容聽組水平伸縮議</b> 目标使用率	最小副本数	最大副本数	创建时间	操作					

 在實際使用環境中,應用會根據CPU負載進行伸縮。您也可在測試環境中驗證Auto Scaling,通 過給Pod進行CPU壓測,可以發現Pod在半分鐘內即可完成水平的擴充。

容器组	关联服务	事件	容器组水平伸缩器			
名称				状态	镜像	
nginx-test-	deployment-656	5fcb7d4c-4d	c4xt	●运行	nginx:latest	
nginx-test-	deployment-656	öfcb7d4c-kv	wqnh	●运行	h nginx:latest	

#### 方法2 通過kubectl命令進行使用

您也可通過編排模板來手動建立HPA,並將其綁定到要伸縮的Deployment對象上,通過kubect1 命令實現容器自動調整配置。

下面針對一個Nginx應用進行舉例, Deployment的編排模板如下,執行kubectl create -f xxx .yml命令進行建立。

```
apiVersion: apps/vlbeta2 # for versions before 1.8.0 use apps/vlbetal
kind: Deployment
metadata:
   name: nginx
   labels:
      app: nginx
spec:
   replicas: 2
   selector:
      matchLabels:
```

```
app: nginx
 template:
   metadata:
      labels:
       app: nginx
    spec:
      containers:
      - name: nginx
        image: nginx:1.7.9 # replace it with your exactly <image_name:</pre>
tags>
        ports:
         - containerPort: 80
        resources:
                                             ##必須設定,不然HPA無法運行
          requests:
            cpu: 500m
```

#### 然後建立HPA,通過scaleTargetRef設定當前HPA綁定的對象,在本例中綁定是名叫nginx的

#### Deployment.

```
apiVersion: autoscaling/v2beta1
kind: HorizontalPodAutoscaler
metadata:
 name: nginx-hpa
 namespace: default
spec:
                                               ###定名為nginx的
 scaleTargetRef:
Deployment
    apiVersion: apps/v1beta2
   kind: Deployment
   name: nginx
 minReplicas: 1
 maxReplicas: 10
 metrics:
  - type: Resource
    resource:
     name: cpu
      targetAverageUtilization: 50
```

说明:

HPA需要給Pod設定request資源,如果沒有request資源,HPA不會運行。

#### 執行kubectl describe hpa [name]會發現有類似如下的warnning。

Warning FailedGetResourceMetric 2m (x6 over 4m) horizontal-pod -autoscaler missing request for cpu on container nginx in pod default /nginx-deployment-basic-75675f5897-mqzs7 Warning FailedComputeMetricsReplicas 2m (x6 over 4m) horizontal-pod -autoscaler failed to get cpu utilization: missing request for cpu on container nginx in pod default/nginx-deployment-basic-75675f5

建立好HPA後,再次執行kubectl describe hpa [name]命令,可以看到如下資訊,則表示 HPA已經正常運行。

Normal SuccessfulRescale 39s horizontal-pod-autoscaler New size: 1; reason: All metrics below target

此時當Nginx的Pod的利用率超過本例中設定的50%利用率時,則會進行水平擴容,低於50%的時候 會進行縮容。

### 1.11.4 通過資源分組進行監控與警示

前提条件

- 如果之前沒有建立過叢集,您需要##Kubernetes##。
- Kubernetes 版本需要在 1.8.4 及以上,若叢集版本過低,您可以先對叢集進行升級,然後通過 升級監控服務的方式快速建立資源警示分組。

#### 背景信息

在 IT 系統基礎設施營運中,監控警示一直是保證可靠性和安全性的基礎,有助於日常營運、系統監 測、故障排除和調試。

在 Kubernetes 情境下,傳統的容器監控方案通過靜態配置化的監控 agent 或中心化的 server 進行 資源監控和警示,會遇到很大的難題。例如,由於容器更多的是在資源集區中調度,宿主機部署監 控 agent 會造成缺乏必要資訊來識別監控對象;容器的生命週期與傳統應用相比而言會更加短暫, 而由容器抽象的上層概念如 kubernetes 中的 ReplicaSet、Deployment 等則沒有太好的辦法從採集 的資料中進行反向的抽象,造成單純的容器監控資料無法有效地進行監控資料的彙總和警示,一旦 應用的發布可能會導致原有的監控與警示規則無法生效。

阿里雲Container Service Kubernetes 與Cloud Monitor進行了深度整合,用應用分組來抽象邏輯概 念,實現邏輯的概念和物理概念在監控資料、生命週期上面的統一。此外,阿里雲Cloud Monitor服 務提供豐富的功能特性和自訂工具,協助您快速實現 Kuberbetes 資源監控和警示的最佳實務。

#### 操作步骤

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的叢集,在叢集列表下找到目的地組群。
- 查看 kubernetes 版本和叢集 ID 等資訊,單擊叢集右側的更多 > 升級監控服務,快速建立資源 警示分組。

道 说明 若您的叢銅	月: 集版本低於 1.8.4,言	青先單	『擊叢集升	十級進行チ	十級。		
容器服务	集群列表				您最多可以创题	25 个集群,每个集群最多可以添加 20 个节	点 刷新 创建 Kubernetes 集群 👻
Kubernetes Swarm	常见问题: Ø 如何创建集群 Ø 扩容和缩容集群	🖉 通过 kube	ctl 连接 Kubernetes 集群	♂ 通过命令管理应用			
▼ 集群							
集群	当称·*						
节点	集群名称/ID	地域	网络类型	集群状态	创建时间	Kubernetes 版本	操作
存储	k8s-cluster 🖌	华东1	虚拟专有网络 vpc-bp1gnu8br4a	●运行中	2018-04-23 11:41:46	1.9.3	
▼ 应用							删除
部署							添加已有节点
服务							3         并级监控服务

4. 登入 Cloud Monitor#####。在左側導覽列中單擊應用分組,在應用分組中,可以看到包含叢集

ID 資訊的 kubernetes 資源分組。

云监控	应用分组					€ 刷新	创建组 当前版本: 升级版本,	免费版 获得更多配额和功能	升级
概览	業型 ▼ 请输入要搜索的组名(支持模糊搜索)	搜索	R						
Dashboard	分组名称 / 分组ID	健康状况 🕜	类型	服务器总数 🕗	资源类型总数 🕗	不健康实例数❷	创建时间		操作
主机监控	test / 27016	٢	自建	2	2	0	2017-10-22 03:28:43	管理	更多,
事件监控	k8s-ce7t L-master / 53473	۲	kubernetes	3	2	0	2018-04-03 09:56:01	管理	更多,
自定义监控	k8s-ce7t I-worker / 53474	•	kubernetes	1	1	0	2018-04-03 09:56:01	管理	更多,
日志监控							共 3条	10 🔻 « < 1	> >>
▶ 站点监控专业版									

5. 單擊分組名稱,進入具體的分組頁面,您可以查看組內的各項監控視圖。以 kubernetes 的 master 分組為例,預設顯示分組的拓撲視圖。

Kubernetes 節點從職能上分為 Worker 和 Master 兩種不同的節點。Master 節點上面通常會部署 管控類型的應用,整體的資源要求以強魯棒性為主;而 Worker 節點更多的承擔實際的 Pod 調 度,整體的資源以調度能力為主。當你建立資源警示分組時,Container Service會為你自動建立 兩個資源分組,一個是 Master 組,一個是 Worker 組。Master 組中包含了 Master 節點以及與 其相關的負載平衡器;Worker 組包含了所有的工作節點。

<	k8s-ce7b578f229754f4783ce693839bdf031-master	1小时	2小时	6小时 12小	时 1天	3天 7天 14天	2018-04-	03 05:42:25	- 2018-04-03 菌
组内资源									
• 监控视图					□□副新	新建报警规则	修改组	删除组	应用模板到组
故障列表									
事件监控									
可用性监控		<b></b>							
日志监控				负载均衡					
自定义监控									
报警历史			3		]				
报警规则			J						
				云服务器ECS	]				
	监控图表						每行展示 3	个圏 🔹	添加监控图表

 您若想瞭解分組內詳細的機器資訊,請單擊列表視圖的表徵圖,您可以查看分組內的各雲產品的 詳細資料。

<	k8s-4									
组内资源										
• 监控视图	_						こ別新	新建报警规则	修改组 删除组	应用模板到组
故障列表	Δ 🗉 🕙									
事件监控	云服务器ECS (3)	进入:	云服务器ECS 控制台	负载均衡 (1)		进入 负载均衡 控制台				
可用性监控	实例名称	健康状况 🖉	资源描述	实例名称	健康状况 🛛	资源描述				
日志监控	933		10	lb-	●正常状态					
日定义监控		■ 11.44/ASS	.10							
投管历史	10.0				共 1条, 毎	页显示10条 〈 1 →				
INEXCAS	Table 1	●正常状态	.11							
	44.04									
	Table 1	♥正常状态	.12							
_		共 3条, 每页显示1	.0条 < 1 →							
					查看全部					
	监控图表								每行展示 3 个图 🔹	添加监控图表

7. 單擊右下角的添加監控圖表,您可以添加更多的監控圖表。

┃ 添加图表	×
1 选择图表类型	
折线图 面积图 热力图 饼图	
2 选择监控项	
云产品监控 日志监控 自定义监控	
云服务器ECS ▼ <b>X服务器ECS</b> Y轴显示范围: 0 auto	
无数据	
监控项: 插件无心跳	
十添加监控项	
发布取消	

8. 在左側導覽列中單擊監控視圖,您可以查看分組內各雲產品的詳細監控指標。



- 在左側導覽列中單擊警示規則,該頁面列出當前分組中已有的警示規則列表,預設會在 Master 分組中設定所有節點的核心組件的健全狀態檢查。
  - 1. 您可以單擊建立警示規則,根據業務需求,來建立屬於該分組的警示規則。

<	k8s	-(		¦er ≄ )8	回应用分组	1小时 2小时 6少	时 12小时 1天 3天 7	天 14天 2018-04-	03 07:41:52 - 2018	3-04-03	3 🗰
组内资源 ● 监控规图	请城	认报警规则名称			搜索		2 刷新 新建	报警规则修改组	删除组 应用	月模板 3	到組
故障列表		规则名称	状态 (全部) -	启用	维度	报警规则	产品名称 (全部) -	2 1st			操作
事件监控 可用性监控		kube-proxy- TelnetLatency.Average	●正常状态	已启用	分組維度:k8s- ce7b578f229754f4783ce693839bdf031- worker	1分钟 平均值>1000 连续 3 次 则报警	云监控-可用性监控	云账号报警联 系人 查看	11	附上	查看 禁用
日志监控 自定义监控 招警历史	۰	kube-proxy- TelnetStatus.Value	●正常状态	已启用	分組維度:k8s- ce7b578f229754f4783ce693839bdf031- worker	1分钟 监控值>400 连续 3 次 则报警	云监控-可用性监控	云账号报警联 系人 查看	何	<b>时</b> 文	查看 禁用
报警规则 1		kubelet- TelnetLatency.Average	●正常状态	已启用	分組維度:k8s- ce7b578f229754f4783ce693839bdf031- worker	1分钟 平均值>1000 连续 3 次 则报警	云监控-可用性监控	云账号报警联 系人 查看	11	対け	查看 禁用
		kubelet- TelnetStatus.Value	♥正常状态	已启用	分組維度:k8s- ce7b578f229754f4783ce693839bdf031- worker	1分钟 监控值>400 连续 3 次 则报警	云监控-可用性监控	云账号报警联 系人 查看	10	100	查看 禁用
Ξ		启用 禁用 册	削除					共 4条	20 ¥ « < 1	>	>

- 2. 進入警示規則頁面,您需要設定警示規則。
  - 您可選擇警示關聯對象。
  - 您可選擇是否使用模板建立警示規則。若選擇使用模板建立警示規則,您可以在選擇模板
     下拉框中選擇已有的警示模板;或者可以單擊建立警示模板,建立新的自訂警示模板,參
     見,然後再進行選擇。
  - 您可設定通知方式,如通過釘釘、郵件、SMS的方式在第一時間擷取到 Kubernetes 的叢 集狀態。

创建报警	规则 t 返回									
1	关联资源									
	产品:	容器服务Kubernetes版	-							
	资源范围:	应用分组								
	分组:	k8s-c51b1b2bd791f4496ae28	k8s-c51b1b2bd791f4496ae28fd3a6							
2	设置报警规则									
	使用模板:	● 是 ○ 否								
	选择 <mark>模</mark> 板:	常用基础模板 🗙		▼ 创建报	警模板					
		常用基础模板_cpu_total	Host.cpu.totalUsed •	1m •	连续3次 🔻	> *	90	%		
		常用基础模板_diskusage_utili	Host.disk.utilization -	1m 👻	连续3次 🔻	> •	90	%		
		常用基础模板_memory_usedu	Host.mem.usedutilization -	1m -	连续3次 🝷	> -	90	%		
		常用基础模板_InternetOutRat	公网流出带宽使用率   ▼	1m •	连续3次 ▼	> •	90	%		
		常用基础模板_agent_heartbe	插件无心跳	1m -				-		
	通道沉默时 间:	24小时	- @							
	生效时间:	00:00 ▼ 至 23:59	•							
3	通知方式									
	通知对象:	联系人通知组	全选 已选组 1~	ŕ	全	先				
		搜索	Q 云账号报	警联系人						
			→							
			+							
		快速创建联系人组								

3. 最後單擊確認,本例中建立的警示規則會出現在規則列表下。

后续操作

在左側導覽列中,您可以探索更多符合自己資源監控需求的功能特性,如故障列表、事件監控、可 用性監控、日誌監控等。

# 1.12 安全管理

# 1.12.1 概述

### 授權管理

Kubernetes 叢集支援叢集層級的操作的子帳號授權。

#### 詳細操作參見#####。

#### 全鏈路 TLS 認證

在Container Service提供的 Kubernetes 叢集中,存在的以下通訊鏈路,均會進行 TLS 認證校正, 以保證通訊不被竊聽或篡改。

- 位於 Worker 節點上的 kubelet 主動串連位於 Master 節點上的 apiserver 時
- 位於 Master 節點上的 apiserver 主動串連位於 Worker 節點上的 kubelet 時

在初始化過程中,發起初始化的 Master 節點會通過 SSH 隧道的方式串連到其他節點的 SSH 服務 (22 連接埠)進行初始化。

#### 原生 Secret&RBAC 支援

Kubernetes Secret 用於儲存密碼、OAuth Token、SSH Key 等敏感資訊。明文地將這些敏感資訊 寫在 Pod YAML 檔案中或者寫在 Dockerfile 固化到鏡像中,會有資訊泄露的可能,使用 Secret 能 有效地避免這些安全隱患。

#### 詳細資料參見Secret。

Role-Based Access Control (RBAC) 使用 Kubernetes 內建的 API 組來驅動授權鑒權管理,您可以 通過 API 來管理不同的 Pod 對應到不同的角色,以及各自的角色擁有的存取權限。

詳細資料參見Using RBAC Authorization。

#### 網路隔離

Kubernetes 叢集中不同節點間 Pod 預設是可以互相訪問的。在部分情境中,不同業務之間不應該 網路互連,為了減少風險,您需要引入網路隔離(Network Policy)。在 Kubernetes 叢集中,您可 以使用 Canal 網路驅動實現網路隔離支援。

#### 鏡像安全掃描

Kubernetes 叢集可使用Container Registry進行鏡像管理,鏡像服務支援鏡像安全掃描。

鏡像安全掃描可以快速識別鏡像中存在的安全風險,減少 Kubernetes 叢集上啟動並執行應用被攻 擊的可能性。

詳細描述參見 ######。

#### 安全性群組與公網訪問

每個建立的叢集會被預設分配一個新的、安全風險最小化的安全性群組。該安全性群組對於公網入 方向僅允許 ICMP。 建立叢集預設不允許公網 SSH 連入,您可以參考*SSH##Kubernetes*## 配置通過公網 SSH 連入到 叢集節點中。

叢集節點通過 NAT Gateway訪問公網,可進一步減少安全風險。

# 1.12.2 Kube-apiserver審計日誌

在Kubernetes叢集中,apiserver的審計日誌可以協助叢集管理員記錄或追溯不同使用者的日常操 作,是叢集安全營運中重要的環節。本文旨在協助您瞭解阿里雲Kubernetes叢集apiserver審計日誌 的相關配置,以及如何通過SLSLog Service收集和搜尋指定的日誌內容。

配置介紹

當前建立Kubernetes叢集會預設開啟apiserver審計功能,相關的參數配置功能如下:



登入到Master節點, apiserver設定檔的目錄是/etc/kubernetes/manifests/kube-

apiserver.yaml.

配置	說明
audit-log-maxbackup	審計日誌最大分區儲存10個記錄檔
audit-log-maxsize	單個審計日誌最大size為100MB
audit-log-path	審計日誌輸出路徑為/var/log/kubernetes/ kubernetes.audit
audit-log-maxage	審計日誌最多儲存期為7天
audit-policy-file	審計日誌配置策略檔案,檔案路徑為:/etc/ kubernetes/audit-policy.yml

登入Master節點機器,審計配置策略檔案的目錄是/etc/kubernetes/audit-policy.yml,內 容如下:

```
apiVersion: audit.k8s.io/vlbetal # This is required.
kind: Policy
# Don't generate audit events for all requests in RequestReceived
stage.
omitStages:
    - "RequestReceived"
rules:
    # The following requests were manually identified as high-volume and
low-risk,
    # so drop them.
    - level: None
    users: ["system:kube-proxy"]
    verbs: ["watch"]
    resources:
        - group: "" # core
```

```
resources: ["endpoints", "services"]
  - level: None
   users: ["system:unsecured"]
   namespaces: ["kube-system"]
    verbs: ["get"]
    resources:
      - group: "" # core
       resources: ["configmaps"]
  - level: None
    users: ["kubelet"] # legacy kubelet identity
    verbs: ["get"]
    resources:
      - group: "" # core
        resources: ["nodes"]
  - level: None
    userGroups: ["system:nodes"]
    verbs: ["get"]
   resources:
      - group: "" # core
        resources: ["nodes"]
  - level: None
   users:
      - system:kube-controller-manager
      - system:kube-scheduler
      - system:serviceaccount:kube-system:endpoint-controller
    verbs: ["get", "update"]
   namespaces: ["kube-system"]
    resources:
      - group: "" # core
       resources: ["endpoints"]
  - level: None
   users: ["system:apiserver"]
    verbs: ["get"]
    resources:
      - group: "" # core
       resources: ["namespaces"]
  # Don't log these read-only URLs.
  - level: None
    nonResourceURLs:
      - /healthz*
      - /version
      - /swagger*
  # Don't log events requests.
  - level: None
    resources:
      - group: "" # core
       resources: ["events"]
  # Secrets, ConfigMaps, and TokenReviews can contain sensitive &
binary data,
  # so only log at the Metadata level.
  - level: Metadata
    resources:
      - group: "" # core
       resources: ["secrets", "configmaps"]
      - group: authentication.k8s.io
       resources: ["tokenreviews"]
  # Get repsonses can be large; skip them.
  - level: Request
   verbs: ["get", "list", "watch"]
   resources:
      - group: "" # core
      - group: "admissionregistration.k8s.io"
```

```
- group: "apps"
    - group: "authentication.k8s.io"
    - group: "authorization.k8s.io"
    - group: "autoscaling"
    - group: "batch"
    - group: "certificates.k8s.io"
    - group: "extensions"
    - group: "networking.k8s.io"
    group: "policy"group: "rbac.authorization.k8s.io"
    - group: "settings.k8s.io"
    - group: "storage.k8s.io"
# Default level for known APIs
- level: RequestResponse
  resources:
    - group: "" # core
    - group: "admissionregistration.k8s.io"
    - group: "apps"
    - group: "authentication.k8s.io"
    - group: "authorization.k8s.io"
    - group: "autoscaling"
    - group: "batch"
    - group: "certificates.k8s.io"
    - group: "extensions"
    - group: "networking.k8s.io"
    - group: "policy"
    - group: "rbac.authorization.k8s.io"
    - group: "settings.k8s.io"
    - group: "storage.k8s.io"
# Default level for all other requests.
- level: Metadatak8s.io"
    - group: "storage.k8s.io"
# Default level for all other requests.
- level: Metadata
```

- 在收到請求後不立即記錄日誌,當返回體header發送後才開始記錄。
- 對於大量冗餘的kube-proxy watch請求,kubelet和system:nodes對於node的get請求,kube組件在kube-system下對於endpoint的操作,以及apiserver對於namespaces的get請求等不作審計。
- 對於/healthz\*, /version\*, /swagger\*等唯讀url不作審計。
- 對於可能包含敏感資訊或二進位檔案的secrets, configmaps, tokenreviews介面的日誌等級設為 metadata,該level只記錄請求事件的使用者、時間戳記、請求資源和動作,而不包含請求體和 返回體。
- 對於一些如authenticatioin、rbac、certificates、autoscaling、storage等敏感介面,根據讀寫記錄相應的請求體和返回體。

#### 採集和檢索

在使用Kube-apiserver審計日誌之前,請確保在建立叢集的配置中開啟了SLSLog Service,並成功 建立對應的日誌Project和Logstore。

- 1. 登入 Log Service#####。
- 2. 單擊左側導覽列中Project管理,選擇建立叢集時設定的日誌Project,單擊名稱進入日誌Project 頁面。

日志服务	Project列表				学习路径创建	Project
Project管理	输入Project名称进行模糊查: 搜索					
	Project名称-	注释	地域	创建时间-		操作
	k8s-log	k8s log project, create	华东 1	2018-09-17 15:14:11	修改   全球加速	删除

 在Project詳情頁中,預設進入日誌庫頁面,查看名為audit-\${clustered}的日誌庫(logstore), 單擊右側的查詢,叢集對應的審計日誌會收集在該日誌庫中。

<b>道</b> 说明:	
在您建立過程中,指定的日誌Project中會自動添加一個名稱為audit-\${clustered}的日誌庫。	

<	@ k8s-log-cbbaab59f1e164a     ▲ 返回Project列表										
	Logstore列表					学习路径	查看Endpoint	创建			
<ul> <li>LogHub - 实的未来</li> <li>[文档] 接入指南</li> </ul>	请输入Logstore名进行模糊 搜索										
Logtail配置	Lonstore实际	数据接入向导	些挖	日志采作模式		日志消费模式		操作			
Logtail机器组	Lugaron carry.	20040307 01010	111112	HISTORICS	日志消费	日志投递	2	DR.IP			
▼ LogHub - 实时消费	audit- caaf06031015d		Ľ	Logtail配置(管理)  诊断   更多-	预览	MaxCompute   OSS	查询	修改 删除			

4. 當叢集管理員需要關注某一子帳號的行為時,可以輸入相應子帳號id,追溯其操作。

				i ,	
audit-cce5036					
🗟 audit-co	0.000	<b>3</b> (周于 k8s-)	0g.cce503673abe9414cbbc09d518c97a76)	使速查询	另存为告望
1			@ Ø	报	北京
40 0 15#j14分	158 <b>4</b> 525	3	16#532 <sup>(2)</sup> 17#512 <sup>(2)</sup> 18#532 <sup>(2)</sup> 18#532 <sup>(2)</sup>		19时12分
			日志总条数:44 查询状态:结果精确		
原始日志 统计器	图表			列设置	Ū.
快速分析	<	时间▲▼	内容 ▼		
您还没有指定学段登词。 赶紧 须加吧(金者有助)	1 Q	09-13 19:13:43	source_:172.16.0.2 tsg :_hostmame_:igglalde-t8kd tsg :_bast_:/var/og/xubenetes/xubernetes.audit tsg :_eser_defined_id_: K8-group-coe503673abe8114dbbb08dt516:67a76 tsg :_continer_name; kube-opieever tsg :_mange_name; sha256.82150741e6/22adt5dfb1ddbb08dt516:67a76 tsg :_continer_name; kube-opieever tsg :_mange_name; sha256.82150741e6/22adt5dfb1ddbb0ed518e57a76 tsg :_continer_name; kube-opieever tsg :_mange_name; sha256.82150741e6/22adt5dfb1ddbb0ed518e57a76 tsg :_pod_name; sha256.82150741e6/22adt5dfb1ddbb0ed5458ea31a8s3895a06df794c0tf785cdae2 tsg :_mange_name; sha256.82150741e6/22adt5dfb1ddbb0ed54518e7 tsg :_pod_name; sha256.82150741e6/22adt5dfb1ddbb0ed54518e7 tsg :_audit: poisever _tpgic_: continer_('ndr':'Event', 'apl/vesion':'audit.k8s.io/v1beta1', 'metadata1', ['creationTimestamp':'2018-09-13T1113.4322', 'avent': 'Request', 'timestamp':'2018-09- 13T1113.4322', 'audit: poisever _tpgic_: content_('ndr':'Levnt', 'apl/vesion':'audit.k8s.io/v1beta1', 'imetadata1', ['creationTimestamp':'2018-09-13T1113.43123', 'avent': 'Request', 'timestamp':'2018-09- 13T1113.4312', 'audit:', 'apl/vesion': 'audit.k8s.io/v1beta1', 'imetadata1', ['creationTimestamp':'2018-09-13T1113.43123', 'avent': 'Request', 'timestamp':'2018-09- 13T1113.43112', 'audit:', 'apl/vesion': 'audit:', 'apl/vesion': 'audit:', 'apl/vesion': 'arantestad', 'incort ': 'aple:'', tandit:', 'apl/vesion': 'arantestad', 'incort 'arantestad', 'apl/vesion': 'arantestad', 'apl/vesion': 'arantestad', 'apl/vesion':	'responseSta Role \*cs:adr	atus*: min\* to
	2 🔍	09-13 18:05:46	source_172.16.12 tagtonisma: jogali-dis-948vf tag : path : /var/ko/kubernetes.audit		

5. 當叢集管理員關注某一具體資來源物件時,可以輸入相應的資源名稱,檢索時間段內的指定操

作。

🗟 config-oper												
🗟 audit-c	 1	() )属于 k8s-log	g-cb	1.00	d )		①15分	钟(相对) 🔻	分享	查询分析属性	另存为快速查询	另存为告警
1 tomcat											© Ø	搜索
			domain beta hangchou: b type"than jor 112: 168. ("podCIDR" storage".rule type".ouc from the type".ouc from the type".cont description.ouc from the type type from the type type type from the type type type from type type type type type type from type type type type type type from type type type type type type type from type type type type type type type type	kubenetes iol op to 2,00% vijfer pel 2,00% vijfer 144,464 K." 144,464 K." H44,464 K." 144,464 K." 144,464 K." 144,464 K." 144,464 K." 144,464 K." 172,202,128 174,662 K." 174,662	region "ich-Han zcybler," inn with the scybler, and scybl	ngzhou", "failui actions", "failui actions", "failui actions", "failui actions", "failui actions", "failui actions", "failui actions, "failui actions, actio	Ine-domain Ennel, aipha c: nel, aipha c: cket"	leta kuberneti rores. com/ba rores. com/ba rores. com/ba rores. com/ba romemory. 32 romemory.	es. lo/zone" ckend-dats reos.com/n n.sock", "nc 181692Ki", 181692Ki", 181692Ki", 181692Ki", 181692Ki", 181692Ki", 181692Ki, 18	""null", "fanel alp bublic- "'null", "fanel alp bublic- "("capacity", "(copu gods", 1107, 'alloc pods", 'alloc pods", 'alloc pods", 'alloc pods,	Rubernetes io/nostma           Rubernetes io/nostma           accress com/backe           es.io/ttl"'0")}"spec":           "2" rephemeral- atable"; (rcpur'2"; repl           atable"; (rcpur'2"; repl           18-09-           8-09-           f6":           7/E 74", "bootID":"3cat           'operatingSystem":"lin           'sizeBytes":46266           ib bo5e750867", "regis           :cn-           1266d51e35bb", "regis           33d38379b5726da","           pr186d7d7dbf72e8","	me":"cn- nd- nemeral- ux","archite 8363}, stry.cn- try-vpc.cn- registry- registry-

### 支援第三方日誌解決

您可以在叢集Master各節點,在/var/log/kubernetes/kubernetes.audit 路徑下找到審計 日誌的源檔案。該檔案是標準的json格式,您可以在部署叢集時選擇不使用阿里雲的Log Service, 根據需要對接其他的日誌解決方案,完成相關審計日誌的採集和檢索。

### 1.13 發行管理

## 1.13.1 基於Heml的發行管理

在阿里雲 Kubernetes 服務提供了豐富的雲市場,其中的應用目錄和服務類別目錄功能整合了 helm 包管理工具,協助您快速構建雲上應用,因為 chart 可以多次發布(release),這就帶來一個發布 版本管理的問題。因此,阿里雲 Kubernetes 服務提供了發布功能,通過 Web 介面的方式對通過 helm 發布的應用進行管理。

#### 前提條件

- 您已經成功建立一個 Kubernetes 叢集,參見##Kubernetes##。
- 您已經使用應用目錄或服務類別目錄功能,安裝了 helm 應用,參見## Helm ######。本例中是 一個 tf-model 應用。

#### 查看發布的詳情

- 1. 登入Container Service#####。
- 在 Kubernetes 菜單下,單擊左側導覽列中的應用 > 發布,選擇所需的叢集,進入發布列表頁 面。

您可看到該叢集下通過 helm 包管理工具發布的應用及服務等。

3. 以 tf-model 為例,您可查看發布的詳情資訊,單擊右側的詳情,進入該發布的詳情頁面。

你可以查看該發布的目前的版本和曆史版本等資訊,目前的版本為1,無曆史版本。您還可查看 tf-model 的資源資訊,如資源名稱,資源類型,以及查看 YMAL 資訊。



單擊資源名稱,可進入 Kubernetes Dashboard 頁面,查看對應資源的詳細運行狀態。

4. 單擊參數,你可查看該 helm 包安裝的參數配置。

#### 更新發布的版本

- 1. 登入Container Service#####。
- 在 Kubernetes 菜單下,單擊左側導覽列中的應用 > 發布,選擇所需的叢集,進入發布列表頁 面。

您可看到該叢集下通過 helm 包管理工具發布的應用及服務等。

- 3. 以 tf-model 為例,您可更新該發布,單擊右側的更新,彈出更新發布對話方塊。
- 4. 在對話方塊中修改相關參數,隨後單擊更新,可對該發布進行更新。

在發布列表頁面,您可以看到目前的版本變為 2,您可以在曆史版本菜單下找到版本1,單擊 復 原到該版本,可進行復原。

#### 刪除發布

- 1. 登入Container Service#####。
- 在 Kubernetes 菜單下,單擊左側導覽列中的應用 > 發布,選擇所需的叢集,進入發布列表頁 面。

您可看到該叢集下通過 helm 包管理工具發布的應用及服務等。

3. 以 tf-model 為例,您可刪除該發布,單擊右側的刪除,彈出刪除對話方塊。

**4.** 勾選是否清除發布記錄,然後單擊確定,您可以刪除 tf-model 應用,其包含的 service、deployment 等資源都會一併刪除。

## 1.14 Istio管理

### 1.14.1 概述

Istio是一個提供串連、保護、控制以及觀測微服務功能的開放平台。

微服務目前被越來越多的IT企業重視。微服務是將複雜的應用切分為若干服務,每個服務均可以獨 立開發、部署和伸縮;微服務和容器組合使用,可進一步簡化微服務的交付,提升應用的可靠性和 延展性。

隨著微服務的大量應用,其構成的分布式應用架構在營運、調試、和安全管理等維度變得更加複 雜,開發人員需要面臨更大的挑戰,如:服務發現、負載平衡、故障恢複、指標收集和監控,以及 A/B測試、灰階發布、藍綠髮布、限流、存取控制、端到端認證等。

Istio應運而生。Istio是一個提供串連、保護、控制以及觀測微服務功能的開放平台,提供了一種簡 單的建立微服務網路的方式,並提供負載平衡、服務間認證以及監控等能力,同時Istio不需要修改 服務即可實現以上功能。

lstio提供如下功能:

- 流量管理:控制服務之間的流量及API調用。增強系統的可靠性。
- 鑒權及安全保護:為網格中的服務提供身分識別驗證,並保護服務的流量。增強系統的安全性。
- 策略執行:控制服務之間的存取原則,且不需要改動服務。
- 可觀察性: 擷取服務之間的流量分布及調用關係。 快速定位問題。

#### lstio架構

Istio在邏輯上分為控制層面和資料層面:

- 控制層面:管理代理程式(預設為Envoy),用於管理流量路由、運行時策略執行等。
- 資料層面:由一系列代理(預設為Envoy)組成,用於管理和控制服務之間的網路通訊。



lstio主要由以下組件構成:

- Istio管理器(Pilot):負責收集和驗證配置,並將其傳播到各種Istio組件。它從策略執行模組 (Mixer)和智能代理(Envoy)中抽取環境特定的實現細節,為他們提供使用者服務的抽象表 示,獨立於底層平台。此外,流量管理規則(即通用4層規則和7層HTTP/gRPC路由規則)可以 在運行時通過Pilot進行編程。
- 策略執行模組(Mixer):負責在服務網格上執行存取控制和使用原則,並從智能代理(Envoy))
   )和其他服務收集遙測資料。依據智能代理(Envoy)提供的屬性執行策略。
- Istio安全模組:提供服務間以及使用者間的認證,確保在不需要修改服務代碼的前提下,增強服
   務之間的安全性。包括3個組件:
  - 身份識別:當Istio運行在Kubernetes時,根據容器Kubernetes提供的服務帳號,識別運行服務的主體。
  - -Key管理:提供CA自動化產生和管理key和認證。
  - 通訊安全:通過智能代理(Envoy)在用戶端和服務端提供通道(tunnel)保證服務的安全
     性。
- 智能代理(Envoy):作為一個獨立的組件與相關微服務部署在同一個Kubernetes的pod上,並 提供一系列的屬性給策略執行模組(Mixer)。策略執行模組(Mixer)以此作為執行策略的依 據,並發送到監控系統。

# 1.14.2 部署Istio

為解決微服務的分布式應用架構在營運、調試、和安全管理等維度存在的問題,可通過部署Istio建 立微服務網路,並提供負載平衡、服務間認證以及監控等能力,同時Istio不需要修改服務即可實現 以上功能。

#### 前提條件

您已經成功建立一個 Kubernetes 叢集,參見##Kubernetes##。

背景資訊

- 阿里雲Container ServiceKubernetes 1.10.4及之後版本支援部署Istio,如果是1.10.4之前的版本,請先升級到1.10.4或之後版本。
- 一個叢集中,worker節點數量需要大於等於3個,保證資源充足可用。

#### 操作步驟

#### 通過叢集介面部署Istio

- 1. 登入 Container Service#####。
- 2. 單擊左側導覽列中的叢集,進入叢集列表頁面。
- 3. 選擇所需的叢集並單擊操作列更多 > 部署lstio。

集群名称/ID	集群类型	地域 (全部) 💌	网络类型	集群状态	创建时间	Kubernetes 版本			操作
20 page and the rest of	Kubernetes	华东1	虚拟专有网络 vpc-bp1e67w2gx5	●运行中	2018-09-20 16:12:26	1.11.2	管理	查看日志   集群伸缩	控制台   更多▼
test-mia	Kubernetes	华东1	虚拟专有网络 vpc-bp1lkyevdjj	●运行中	2018-09-17 11:37:55	1.11.2	管理	查看日志   集群伸缩	控制台 更多▼
								開除	
								#群升级	177
								自动伸缩	
								系统组件升	版
								升级监控服	绣
								部署 Istio	

4. 根據如下資訊,部署Istio:

配置	說明
叢集	部署Istio的目的地組群。
命名空間	部署Istio的命名空間。
發布名稱	發布的Istio名稱。
啟用 Prometheus 度量日誌收集	是否啟用Prometheus 收集度量日誌。預設情況下啟 用。
啟用 Grafana 度量展示	是否啟用Grafana 展示指標的度量資料。預設情況 下啟用。

配置	說明
啟用 ServiceGraph 可視化部署	是否啟用ServiceGraph 進行可視化部署。預設情況 下啟用。
啟用 Sidecar 自動注入	是否啟用 Sidecar 進行容器的自動注入。預設情況 下啟用。
啟用阿里雲Log Service SLS 及 Jaeger	是否啟用阿里雲的Log ServiceSLS 及 Jaeger。預 設情況下不啟用。 服務入口地址(Endpoint):根據配置的Log Service所在region選擇對應的地址。具體請參見## ## 項目名稱(Project):採集日誌所在的項目名稱。 日誌庫名稱(Logstore):採集日誌所在的日誌庫 名稱。 AccessKeyID:訪問Log Service時,所使用的訪問 秘鑰ID。
	说明: 需要選擇有許可權訪問Log Service的 AccessKeyID。
	AccessKeySecret:訪問Log Service時,所使用的 訪問秘鑰Secret。

5. 單擊部署 Istio, 啟動部署。

在部署頁面下方,可即時查看部署進展及狀態。

步骤	状态	
创建 Istio 资源定义	🔴 成功	53 / 53
部署 Istio	**运行中	
部署		

### 預期結果

可通過以下方法查看部署是否成功:

• 在部署 Istio頁面下方,部署 Istio變為已部署。

步骤	状态
创建 Istio 资源定义	等待开始
部署 Istio	等待开始
Ē	1部署

- • 單擊左側導覽列應用 > 容器組,進入容器組頁面。
  - 選擇部署Istio的叢集及命名空間,可查看到已經部署Istio的相關容器組。

<b>容器服务 - Kubernetes →</b>	容器组								周閉
₩近 * 集詳	集群 test-mia 🔻 命名空间 istio-system	n 🔻							
魚群	名称 🔷	状态 🔷	Pod IP 💠	节点 🔷	创建时间 💠	CPU (核)	内存(字节)		
节点	grafana-648d46c555-2llnt	●运行中	10.00	10.000	2018-09-19 13:51:41	0.005	91.469 Mi	详情	更多 ▼
存储卷	istio-citadel-5f494dccbb-w45br	●运行中	(21.01) (4)	*****	2018-09-19 13:51:42	0	10.074 Mi	详情	更多 ▼
命名空间 授权管理	istio-egressgateway-6b598867d8-hdncp	●运行中	10.00		2018-09-19 13:51:41	0.002	26.555 Mi	详情	更多 🔻
7 应用	istio-galley-845597d49c-8wlpp	●运行中	100.000	10.000	2018-09-19 13:51:41	0.025	13.031 Mi	详情	更多 🔻
部署	istio-ilbgateway-7bcb956785-q8m6c	●运行中			2018-09-19 13:51:41	0.003	28.305 Mi	详情	更多 🗸
容器组	istio-ingress-5f95fbf49f-gxcch	●运行中			2018-09-19 13:51:41	0.005		详情	<b>更</b> ≶ ▼
服务	istio-ingress-5f95fbf49f-tbzsz	●运行中	1000		2018-09-22 05:02:50	0.005	27.855 Mi	详情	更多 🗸

- • 單擊左側導覽列應用 > 服務,進入服務列表頁面。
  - 選擇部署Istio的叢集及命名空間,可查看到已經部署Istio相關服務所提供的訪問地址。

容器服务 - Kubernetes +	服务列表						刷新 创建
楸览	集群 test-mia ▼ 命名经	2间 istio-system	*				
▼ 魚群	名称	举型	613#84181	生用IP	内部端点	外部端点	操作
集群	grafapa	ClusterIP	2018-09-19 13:51:41		orafana:3000 TCP		送信   東新   査 <b>濟</b> YAMI   删除
节点	ground	Challen	2010 07 17 15/51/11		grandina booo r cr		And I want I make I was
存储卷	istio-citadel	ClusterIP	2018-09-19 13:51:41	10.000	istio-citadel:8060 TCP istio-citadel:9093 TCP	-	详情   更新   查看YAML   删除
命名空间	istio-egressgateway	ClusterIP	2018-09-19 13:51:41	10.000	istio-egressgateway:80 TCP istio-egressgateway:443 TCP	-	详情   更新   查看YAML   删除
授权管理 ▼ 应用	istio-galley	ClusterIP	2018-09-19 13:51:41	12,000	istio-galley:443 TCP istio-galley:9093 TCP		详情   更新   查看YAML   删除
部署 有状态副本集 容器组 脱务	istio-ilbgateway	LoadBalancer	2018-09-19 13:51:41	12,255,25	istio-ilbgateway:15011 TCP istio-ilbgateway:22/16 TCP istio-ilbgateway:15010 TCP istio-ilbgateway:2500 TCP istio-ilbgateway:28060 TCP istio-ilbgateway:28061 TCP istio-ilbgateway:2513 TCP istio-ilbgateway:51461 TCP		译情   更新   查看YAML   删除
路由 存储声明	istio-ingress	LoadBalancer	2018-09-19 13:51:41	12.2.2	istio-ingress:80 TCP istio-ingress:32000 TCP istio-ingress:443 TCP istio-ingress:32664 TCP	20125	详情   更新   查看YAML   删除

### 通過應用目錄部署Istio

- 1. 登入 Container Service######。
- 2. 單擊左側導覽列中的市場 > 應用目錄,進入應用目錄頁面。

容器服务 - Kubernetes -	应用目录			
▼ 应用				
部響	<b>Q</b> <sub>O</sub>	<b>₽</b> <sub>0</sub>	<b>Q</b> O	<b>₽</b> <sub>0</sub>
容器组				
服务	ack-hyperledger-fabric 1.1.0 incubator	ack-istio 1.0.0 incubator	ack-istio-remote 1.0.0 incubator	ack-openmpi 3.1.0 incubator
仔Z/编户图时 Helm	Ø	Ø	<b>\$</b>	Ø
发布 配置项 保密字典	ack-springcloud-configserver 1.5.13.RELEASE incubator	ack-springcloud-eureka 1.5.13.RELEASE incubator	ack-springcloud-hystrix 1.5.13.RELEASE incubator	ack-springcloud-turbine 1.5.13.RELEASE incubator
▼ 市场	O	en O	••••	m O
資像	$\Omega_0^-$	$\Omega_0^-$	$\Omega_0^-$	$\Omega_0^-$
碼排候做 应用目录	ack-springcloud-zipkin 1.5.13.RELEASE incubator	ack-springcloud-zuul 1.5.13.RELEASE incubator	ack-tensorflow-dev 1.5.0 incubator	ack-tensorflow-serving 1.4.0 incubator

- 3. 單擊ack-istio,進入應用目錄 ack-istio頁面。
- 4. 單擊參數頁簽,進行參數配置。

应用日录 - ack-istio	
ack-istio incubator Helm chart of all istio components for Kubernetes on Alibaba Cloud Container Service	
Image: Service in the image is a service is a service in the image is a service is a service in the image is a service is a service in the image is a service is a service in the image is a service is a service is a service in the image is a service is a service in the image is a service is a service in the image is a service in the service is a serv	シンプレン・シング・シング・シング・シング・シング・シング・シング・シング・シング・シン
☐ 说明:	
• 通用參數的含義、取值及預設情況,可參考說明頁簽Configu	iration欄位。
• 也可以針對特定參數進行定製化配置,例如:是否啟用grafa	ana, prometheus, tracing
、weave-scope以及kiali等,可參考:	
<pre># # # addons configuration # grafana: enabled: true replicaCount: 1 image: istio-grafana</pre>	

service:

```
name: http
type: ClusterIP
externalPort: 3000
internalPort: 3000
. . . .
prometheus:
enabled: true
replicaCount: 1
image:
repository: registry.cn-hangzhou.aliyuncs.com/aliacs-app-catalog/
istio-prometheus
tag: latest
. . . .
tracing:
enabled: true
jaeger:
enabled: true
. . . .
weave-scope:
enabled: true
qlobal:
# global.image: the image that will be used for this release
image:
repository: weaveworks/scope
tag: "1.9.0"
# global.image.pullPolicy: must be Always, IfNotPresent, or Never
pullPolicy: "IfNotPresent"
 . .
kiali:
enabled: true
replicaCount: 1
image:
repository: registry.cn-hangzhou.aliyuncs.com/aliacs-app-catalog/
istio-kiali
taq: dev
```

5. 在右側建立地區,填寫以下基本資料:

配置	說明
叢集	部署Istio的目的地組群。
命名空間	部署Istio的命名空間。預設情況下為 default。
發布名稱	發布的Istio名稱。

6. 單擊建立,啟動部署。

預期結果

- • 單擊左側導覽列應用 > 容器組,進入容器組頁面。
  - 選擇部署Istio的叢集及命名空間,可查看到已經部署Istio的相關容器組。

李器服务 - Kubernetes ▼	容器组							周時
戦送 ▲	集群 test-mia ¥ 命名空间 istio-system ¥							
魚群	名称 🗢	状态 🔷	Pod IP 💠	节点 🔷	创建时间 🔶	CPU (核)	内存(字节)	
节点	grafana-648d46c555-2lint	●运行中			2018-09-19 13:51:41	0.005	91.469 Mi	详情 │ 更多 ▼
存储卷	istio-citadel-5f494dccbb-w45br	●运行中	(1).0.00		2018-09-19 13:51:42	0	10.074 Mi	详情   更多 ▼
命名空间授权管理	istio-egressgateway-6b598867d8-hdncp	●运行中	10.00		2018-09-19 13:51:41	0.002	26.555 Mi	详情   更多 ▼
7 应用	istio-galley-845597d49c-8wlpp	●运行中		10.000	2018-09-19 13:51:41	0.025	13.031 Mi	详情 │ 更多 ▼
部署 有状本副本集	istio-ilbgateway-7bcb956785-q8m6c	●运行中		10.000.00	2018-09-19 13:51:41	0.003	28.305 Mi	详情 │ 更多 ▼
容器组	istio-ingress-5f95fbf49f-gxcch	●运行中			2018-09-19 13:51:41	0.005	35.383 Mi	详情 │ 更多 ▼
服务	istio-ingress-5f95fbf49f-tbzsz	●运行中	1000	-	2018-09-22 05:02:50	0.005	27.855 Mi	详情   更多 ▼

- •• 單擊左側導覽列應用 > 服務,進入服務列表頁面。
  - 選擇部署Istio的叢集及命名空間,可查看到已經部署Istio相關服務所提供的訪問地址。

容器服务 - Kubernetes +	服务列表						刷新 创建
敬范▲	集群 test-mia v 命名空	间 istio-system	•				
▼ 集群	名称	类型	创建时间	集群IP	内部端点	外部端点	操
集群	grafana	ClusterIP	2018-09-19 13:51:41	12.00.0	grafana:3000 TCP		详情   更新   查看YAML   删除
节点	istio-citadel	ClusterIP	2018-09-19 13:51:41	12.04	istio-citadel:8060 TCP istio-citadel:9093 TCP	-	详情   更新   查看YAML   删除
命名空间	istio-egressgateway	ClusterIP	2018-09-19 13:51:41	10.000	istio-egressgateway:80 TCP istio-egressgateway:443 TCP		详情   更新   查看YAML   删除
夜秋言堆 ▼ 应用	istio-galley	ClusterIP	2018-09-19 13:51:41	10.000	istio-galley:443 TCP istio-galley:9093 TCP		详情   更新   查看YAML   删除
部署 有状态副本集 容器组 服务	istio-ilbgateway	LoadBalancer	2018-09-19 13:51:41	12,000,00	istio-ilbgateway:15011 TCP istio-ilbgateway:25016 TCP istio-ilbgateway:15010 TCP istio-ilbgateway:25080 TCP istio-ilbgateway:28060 TCP istio-ilbgateway:28161 TCP istio-ilbgateway:531461 TCP		译情   更新   查看YAML   删除
路由 存储声明	istio-ingress	LoadBalancer	2018-09-19 13:51:41	12.2.2	istio-ingress:80 TCP istio-ingress:32000 TCP istio-ingress:443 TCP istio-ingress:32664 TCP	20125	详情   更新   查看YAML   删除

# 1.14.3 更新Istio

您可以通過更新操作,編輯已部署的Istio。

前提条件

- 您已成功建立一個 Kubernetes 叢集,參見##Kubernetes##。
- 您已成功建立一個Istio,參見##/stio。

### 操作步骤

- 1. 登入 Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列的應用 > Helm,進入發布列表頁面。
- 3. 選擇所需的叢集,選擇待更新的Istio,單擊操作列的更新。



• 通過叢集介面部署的Istio,發布名稱為istio,更新的內容與部署時的選項一致。
通過應用目錄部署的Istio,發布名稱為使用者建立時指定的名稱,更新的內容與部署時的參 數一致。

发布列表							刷新
集群 test-mia ▼							
发布名称	状态	命名空间	Chart 名称	Chart 版本	应用版本	更新时间	操作
istio	● 已部署	istio-system	ack-istio	1.0.2	1.0.2	2018-09-26 11:36:03	详情 更新 删除
aliacs-service-catalog	● 已部署	catalog	catalog	0.1.9		2018-09-17 15:15:07	详情 更新 删除

4. 在彈出的對話方塊中,對Istio的參數進行編輯後,單擊更新。

此處以更新通過叢集頁面部署的Istio為例:

<pre>1 * prometheus: 2 enabled: true 3 * grafana: 4 enabled: true 5 * servicegraph: 6 enabled: false 7 * global: 8 * proxy: 9 autoinject: enabled 10 * sidecarinjectorWebhook: 11 enabled: true 12 * tracing-on-sls: 13 enabled: false 14</pre>	更新发布	×
	<pre>1 * prometheus: 2 enabled: true 3 * grafana: 4 enabled: true 5 * servicegraph: 6 enabled: false 7 * global: 8 * proxy: 9 autoInject: enabled 10 * sidecarInjectorWebhook: 11 enabled: true 12 * tracing-on-sls: 13 enabled: false 14</pre>	

### 预期结果

可通過以下兩種方式,查看更新後的內容:

- 更新完成後,頁面自動跳轉到發布列表頁面,在資源頁簽,可以查看更新的內容。
- 在 Kubernetes 菜單下,單擊左側導覽列的應用 > 容器組,進入容器組頁面,選擇目的地組群和
   目標空間進行查看。

# 1.14.4 刪除Istio

您可以通過刪除操作,刪除已部署的Istio。

前提条件

- 您已成功建立一個 Kubernetes 叢集,參見##Kubernetes##。
- 您已成功建立一個Istio,參見##/stio。

### 操作步骤

- 1. 登入 Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列的應用 > Helm,進入發布列表頁面。
- 3. 選擇所需的叢集及待刪除的Istio,單擊操作列的刪除。

发布列表								刷新
集群 test-mia 🔻								
发布名称	状态	命名空间	Chart 名称	Chart 版本	应用版本	更新时间		操作
istio	●已部署	istio-system	ack-istio	1.0.0	1.0.0	2018-09-19 13:51:40	详情 更新	f删除

4. 在彈出的對話方塊中,單擊確定。

删除应	並用									$\times$
		确定要删隙 清除发布	余应用 isti 记录	o ?						
								确定	取消	
<b>自</b> ・ 不如	说明: []] []]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]]	布記錄:								
	<b>贺                                    </b>	曾做刪陈:								刷新
	集群 test-mia ▼ 发布名称	状态	命名空间	Chart 名称	Chart 版本	应用版本	更新时间			操作
	istio	●已删除	istio-system	ack-istio	1.0.2	1.0.2	2018-09-26 14:12:48		详情 更	新删除
_	此Istio的名	稒不可被再	次使用。							

	通過叢集介面重新部署Istio時,顯示已完成。	
	步骤	状态
	创建 Istio 资源定义	等待开始
	部署 Istio	等待开始
	е	部署
	通過應用目錄部署lstio時,提示:已存在同行	名的部署或資源,請修改名稱。
	提示	×
	已存在同名的部署或资源,请修改名称 Can't install release with errors: rpc erro -service-catalog already exists. Run: hell s of the release Or run: helm delpurge	r: code = Unknown desc = a release named aliacs m lsall aliacs-service-catalog; to check the statu e aliacs-service-catalog; to delete it
		确定
• 勾道	<b>選清除發布記錄,會同時刪除所有發布記錄,</b>	且此Istio的名稱可被再次使用。
建議保	持預設狀態,勾選清除發布記錄。	

### 预期结果

返回發布列表頁面,可以看到該Istio已被刪除。

## 1.15 應用目錄管理

## 1.15.1 應用目錄概述

微服務是容器時代的主題,應用微服務化給部署和管理帶來極大的挑戰。通過將龐大的單體應用拆 分成一個個微服務,從而使各個微服務可被獨立部署和擴充,實現敏捷開發和快速迭代。雖然微服 務帶來了很大的好處,但同時,由於應用拆分成許多組件,對應著龐大數量的微服務,開發人員不 得不面對這些微服務的管理問題,如資源管理、版本管理、組態管理等。

針對 Kubernetes 編排下微服務管理問題,阿里雲Container Service引入 Helm 開源項目並進行整合,協助簡化部署和管理 Kubernetes 應用。

Helm 是 Kubernetes 服務編排領域的開源子項目,是 Kubernetes 應用的一個包管理工具, Helm 通過軟體打包的形式,支援發布的版本管理和控制,簡化了 Kubernetes 應用部署和管理的複雜 性。

#### 阿里雲應用目錄功能

阿里雲Container Service應用目錄功能整合了 Helm,提供了 Helm 的相關功能,並進行了相關功能 擴充,如提供圖形化介面、阿里雲官方 Repository 等。

應用目錄首頁 chart 列表的資訊包含:

- chart 名稱:一個 Helm 包,對應一個目標應用,其中包含了運行一個應用所需要的鏡像、依賴 和資源定義等。
- 版本: chart 的版本號碼。
- Repository:用於發布和儲存 Chart 的倉庫,例如官方倉庫 stable、incubator 等。

各個 chart 詳情頁包含的資訊不盡相同,例如,可能包含:

- chart 簡介
- chart 詳細資料
- chart 安裝到叢集的前提條件,如預先配置持久化儲存卷(pv)。
- chart 安裝命令
- chart 卸載命令
- chart 參數配置項

目前,您可以通過 helm 工具部署和管理應用目錄中的 chart,具體請參見## Helm ######。

### 1.15.2 查看應用目錄列表

操作步骤

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的市場 > 應用目錄,進入應用目錄列表。

您可以查看該頁面下 chart 列表,每個 chart 對應一個目標應用,包含一些基本資料,如應用程 式名稱、版本號碼和來源倉庫等資訊。



#### 后续操作

您可進入單個 chart,瞭解具體的 chart 資訊,可根據相關資訊,利用 Helm 工具部署應用,具體請 參見## Helm ####### 。

## 1.16 服務類別目錄管理

### 1.16.1 概述

雲平台上啟動並執行應用需要使用一些基礎服務,如資料庫、應用伺服器等通用的基礎軟體。譬如 一個 wordpress 應用,作為一個 web 應用,後端需要一個資料庫服務,如 MariaDB。傳統方式是 在 wordpress 應用的編排中也建立應用依賴的 MariaDB 服務,並與 Web 應用進行整合。這種雲上 應用開發的方式,就需要開發人員花費精力解決所依賴的基礎設施軟體的部署和配置,增加應用託 管和遷移的成本。

阿里雲Container Service支援並整合了服務類別目錄的功能,該功能旨在接入和管理 Service Broker,使 kubernetes 上啟動並執行應用可以使用 service broker 所代理的託管服務。服務類別目 錄功能將支援一系列基礎設施軟體,應用開發人員可以不用關心這些軟體的可用性和伸縮能力,也 不用對其進行管理,開發人員可以簡單的將其作為服務使用,只需專註開發核心的應用程式。

服務類別目錄通過 Kubernetes 的 Open Service Broker API 與 Service Broker 進行通訊,並作為 Kubernetes API Server 的中介,以便協商首要規定(initial provisioning)並擷取應用程式使用託管 服務的必要憑據。關於服務類別目錄的具體實現原理,請參考 *service catalog*。

## 1.16.2 開通服務類別目錄

操作步骤

- 1. 登入Container Service#####。
- 2. 在 Kubernetes 菜單下,單擊左側導覽列中的市場 > 服務類別目錄,在右上方選擇目的地組群。
- 3. 若您還未部署服務類別目錄功能,介面上會提示您先進行安裝。

容器服务	÷	服务目录	集群: k8s-cluster ▼
Kubernetes	Swarm		
▼ 集群		① 该集群尚未部署服务目录, 请点击安装	<b>3</b>
集群		4	
节点			
存储			
▼ 应用			
部署			
服务			
发布			
配置项	-		
▼市场	1		
应用目	<b>录</b>		
服务目	₹ 2		

**4.** 安裝完成後,服務類別目錄頁面中會顯示預設安裝的 Service Broker,您可以進入 mariadbbroker 瞭解詳情。

▋ 说明:

服務類別目錄具體實現為一個擴充 API server 和一個 controller,阿里雲Container Service安裝 服務類別目錄功能後,會建立一個 catalog 命名空間。

容器服务 Kubernetes	Swarm	服务目录			集群: k8s-cluster ▼
▼ 集群		_	_	_	
集群		<b>₽</b> <sup>0</sup>	¢°	<b>₽</b> <sup>0</sup>	<b>₽</b> <sup>0</sup>
节点					
存储 ▼ 应用		incubator	incubator	incubator	spark-Droker incubator
部署		·			
服务					
发布					
配置项	Ξ				
▼ 市场					
应用目录					
服务目录					

5. 在左側導覽列中單擊叢集,單擊目的地組群右側的控制台。

容器服务		集群列表				您最多可以创建 5 个集器	羊, 每个集群最多可以添加 20 个节点	刷新	创建 Kubernetes 集群 🛛 👻
Kubernetes	Swarm	常见问题: Ø 如何创建集群 Ø 扩容和缩容集群	詳 <i>8</i> 通过	t kubectl 连接 Kubernetes 绯	# 8 通过命令管	理应用			
<ul> <li>集群 集群</li> </ul>	名称 ▼								
节点		集群名称/ID	地域	网络类型	集群状态	创建时间	Kubernetes 版本		2 🗉
存储		k8s-cluster	华东1	虚拟专有网络 vpc-bp13cwaie5z	●运行中	2018-04-20 11:02:31	1.9.3	管理	童看日志   控制台     集群伸縮   更多▼

6. 進入 Kubernetes Dashboard 頁面,在命名空間中選擇catalog,可以看到該命名空間下安裝了 catalog apiserver 和 controller 相關的資來源物件。

	(IT)						1 413
<	— (60元)   工作品報评本						- Bua
集群	TIPSERVON						
命名空间							
节点							
持久少存後業	100.00%		100.00%			100.00%	
6A	電路		容器组			副本街	
用巴						23400	
存储英	空間						_
命名空间							-
catalog 🕆 🚺	名称 🗘	标签	容器组	Ē	3创建 \$	镜像	
概况		app: aliacs-service-catalog-catalog-apise			2010 04 04 00 24 21	registry.cn-hangzhou.aliyuncs.com/service-cata	
工作负载	aliacs-service-catalog-catalog-apiserver	release: aliacs-service-catal	1/1		2010-04-04 09:54:51	registry.cn-hangzhou.aliyuncs.com/service-cata	•
定时任务		app: aliacs-service-catalog-catal					
守护进程集	<ul> <li>aliacs-service-catalog-catalog-controller-manage</li> </ul>	chart: catalog-0 heritage: Till	1/1	1	2018-04-04 09:34:31	registry.cn-hangzhou.aliyuncs.com/service-cata	:
		release: allacs-service-catal					
174 (14	容器组						÷
1135							
容證组	名称 ≑	节点    状态	÷	已重启	CPU(核) 建 <sup>€</sup> CPU(核)	内存(字节)	
副本集	aliacs-service-catalog-catalog-aniserver-86456dc	589- cn-hangzhou.i- Run	ling	0	2018-04-	4 23.637 Mi	. :
副本控制器		bp1iabxy7s0uwpzmaa2e		-	09:34:32	201007 111	•••
有状态副本集	aliacs-service-catalog-catalog-controller-manage	-Sct hangzhou.i-	ing	0	2018-04- 04 0.00	12 15.094 Mi	
服务发现与负载均衡		opinacky/souwpzmaaze			09:34:32		

### 后续操作

至此,您已經成功開通服務類別目錄功能。接下來可以通過服務類別目錄下的 Service Broker 來建 立託管服務的執行個體,並將其應用到您自己的應用程式中。