ALIBABA CLOUD

Alibaba Cloud

容器服务Kubernetes版

Serverless Kubernetes 叢集使用者指南

Document Version: 20210225

Alibaba Cloud

Legal disclaimer

Alibaba Cloud reminds you to carefully read and fully understand the terms and conditions of this legal disclaimer before you read or use this document. If you have read or used this document, it shall be deemed as your total acceptance of this legal disclaimer.

1. You shall download and obtain this document from the Alibaba Cloud website or other Alibaba Cloud-authorized channels, and use this document for your own legal business activities only. The content of this document is considered confidential information of Alibaba Cloud. You shall strictly abide by the confidentiality obligations. No part of this document shall be disclosed or provided to any third party for use without the prior written consent of Alibaba Cloud.

2. No part of this document shall be excerpted, translated, reproduced, transmitted, or disseminated by any organization, company or individual in any form or by any means without the prior written consent of Alibaba Cloud.

3. The content of this document may be changed because of product version upgrade, adjustment, or other reasons. Alibaba Cloud reserves the right to modify the content of this document without notice and an updated version of this document will be released through Alibaba Cloud-authorized channels from time to time. You should pay attention to the version changes of this document as they occur and download and obtain the most up-to-date version of this document from Alibaba Cloud-authorized channels.

4. This document serves only as a reference guide for your use of Alibaba Cloud products and services. Alibaba Cloud provides this document based on the "status quo", "being defective", and "existing functions" of its products and services. Alibaba Cloud makes every effort to provide relevant operational guidance based on existing technologies. However, Alibaba Cloud hereby makes a clear statement that it in no way guarantees the accuracy, integrity, applicability, and reliability of the content of this document, either explicitly or implicitly. Alibaba Cloud shall not take legal responsibility for any errors or lost profits incurred by any organization, company, or individual arising from download, use, or trust in this document. Alibaba Cloud shall not, under any circumstances, take responsibility for any indirect, consequential, punitive, contingent, special, or punitive damages, including lost profits arising from the use or trust in this document (even if Alibaba Cloud has been notified of the possibility of such a loss).

5. By law, all the contents in Alibaba Cloud documents, including but not limited to pictures, architecture design, page layout, and text description, are intellectual property of Alibaba Cloud and/or its affiliates. This intellectual property includes, but is not limited to, trademark rights, patent rights, copyrights, and trade secrets. No part of this document shall be used, modified, reproduced, publicly transmitted, changed, disseminated, distributed, or published without the prior written consent of Alibaba Cloud and/or its affiliates. The names owned by Alibaba Cloud shall not be used, published, or reproduced for marketing, advertising, promotion, or other purposes without the prior written consent of Alibaba Cloud. The names owned by Alibaba Cloud include, but are not limited to, "Alibaba Cloud", "Aliyun", "HiChina", and other brands of Alibaba Cloud and/or its affiliates, which appear separately or in combination, as well as the auxiliary signs and patterns of the preceding brands, or anything similar to the company names, trade names, trademarks, product or service names, domain names, patterns, logos, marks, signs, or special descriptions that third parties identify as Alibaba Cloud and/or its affiliates.

6. Please directly contact Alibaba Cloud for any errors of this document.

Document conventions

Style	Description	Example
Danger	A danger notice indicates a situation that will cause major system changes, faults, physical injuries, and other adverse results.	**Danger:** Resetting will result in the loss of user configuration data.
Warning	A warning notice indicates a situation that may cause major system changes, faults, physical injuries, and other adverse results.	**Warning:** Restarting will cause business interruption. About 10 minutes are required to restart an instance.
Notice	A caution notice indicates warning information, supplementary instructions, and other content that the user must understand.	**Notice:** If the weight is set to 0, the server no longer receives new requests.
Note	A note indicates supplemental instructions, best practices, tips, and other content.	**Note:** You can use Ctrl + A to select all files.
>	Closing angle brackets are used to indicate a multi-level menu cascade.	Click **Settings**> **Network**> **Set network type**.
Bold	Bold formatting is used for buttons , menus, page names, and other UI elements.	Click **OK**.
`Courier font`	Courier font is used for commands	Run the `cd /d C:/window` command to enter the Windows system folder.
Italic	Italic formatting is used for parameters and variables.	`bae log list --instanceid` *Instance_ID*
[] or [a|b]	This format is used for an optional value, where only one item can be selected.	`ipconfig [-all|-t]`
{} or {a|b}	This format is used for a required value, where only one item can be selected.	`switch {active|stand}`

Table of Contents

1.概述

阿里雲 Serverless Kubernetes 讓您無需管理和維護叢集與伺服器，即可快速建立 Kuberentes 容器應用，並且根據應用實際使用的 CPU 和記憶體資源量進行按需付費。使用 Serverless Kubernetes，您可以專註於設計和構建應用程式，而不是管理運行應用程式的基礎設施。它基於阿里雲彈性計算基礎架構，並且完全相容 Kuberentes API 的解決方案，充分結合了虛擬化資源帶來的安全性、彈性和 Kubernetes 生態。

公測開放地區

目前，阿里雲Container Service Serverless Kubernetes 叢集處於公測階段。暫只開放部分地區，其他地區將相繼開放。

使用限制

- 只能建立一個 serverless kubernetes 叢集。
- 只能使用 2C4G 的 pod。
- 只能最多建立 5 個 pod。

產品定價

公測期間免費使用，統一預設選用 2C4G 規格。

與Container Service的對比

2.Kubernetes 功能支援

API 版本

支援 Kubernetes 1.9 API。

應用負載

- 支援Deployment、StatefulSet、Job/CronJob、Bare Pod。
- 不支援 DaemonSet。

Pod 定義

支援啟動多個容器，設定環境變數，設定 RestartPolicy，設定健全狀態檢查命令，掛載volumes 等。

負載平衡

- 支援建立 LoadBalancer 類型應用。
- 支援Ingress。
- 不支援 NodePort 類型。

配置

支援 Secret 和 Configmap。

儲存

- 支援 emptyDir 和 nfs volume 類型。
- 不支援 PersistentVolume 和 PersistentVolumeClaim。

命名空間

使用者只可看見 default 命名空間，不可添加命名空間。

節點

使用者不可查看 kubernetes 的節點資訊。

事件

使用者可查看 default 命名空間下的事件。

容器logs

使用者可通過**kubectl logs**即時查看容器的日誌。

容器exec/attach

使用者可通過**kubectl exec**進入容器執行命令。

3.快速入門

3.1. 建立 Serverless Kubernetes 叢集

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

前提條件

登入 Container Service管理主控台 和 RAM 管理主控台 開通相應的服務。

操作步驟

1. 登入Container Service管理主控台。
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 Serverless Kubernetes 叢集**。

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。

   > **說明** Serverless Kubernetes 叢集目前處於公測階段，僅支援華東 2（上海）地區。

6. 設定叢集的網路。
   - **網路類型**：Kubernetes 叢集僅支援專用網路。
   - **專用網路**：支援**自動建立**和**使用已有**的 VPC。

     > **說明**
     > - **自動建立**：叢集會自動建立一個VPC，並在VPC中自動建立NAT Gateway以及配置SNAT規則。
     > - **使用已有**：您可以在已有 VPC 列表中選擇所需的 VPC 和交換器。如需訪問公網，比如下載容器鏡像，要配置NAT Gateway，建議將容器鏡像上傳到叢集所在地區的阿里雲鏡像服務，並通過內網VPC地址拉取鏡像。

7. 勾選是否開啟基於PrivateZone的服務發現功能，該功能允許您在叢集VPC內通過服務網域名稱訪問服務。

   > **說明** 在使用勾選該功能前，請確認您已開通PrivateZone服務，參見雲端式解析 PrivateZone的服務發現。
8. 勾選**無伺服器 Kubernetes 服務合約**。
9. 在頁面右側，單擊**建立叢集**，啟動部署。

後續步驟

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 Serverless 叢集。

您可以單擊右側的**管理**，查看叢集的基本資料和串連資訊。

4.叢集管理

4.1. 建立 Serverless Kubernetes 叢集

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

前提條件

登入 Container Service管理主控台 和 RAM 管理主控台 開通相應的服務。

操作步驟

1. 登入Container Service管理主控台。
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 Serverless Kubernetes 叢集**。

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。

   > **說明** Serverless Kubernetes 叢集目前處於公測階段，僅支援華東 2（上海）地區。

6. 設定叢集的網路。
   - **網路類型**：Kubernetes 叢集僅支援專用網路。
   - **專用網路**：支援**自動建立**和**使用已有**的 VPC。

     > **說明**
     > - **自動建立**：叢集會自動建立一個VPC，並在VPC中自動建立NAT Gateway以及配置SNAT規則。
     > - **使用已有**：您可以在已有 VPC 列表中選擇所需的 VPC 和交換器。如需訪問公網，比如下載容器鏡像，要配置NAT Gateway，建議將容器鏡像上傳到叢集所在地區的阿里雲鏡像服務，並通過內網VPC地址拉取鏡像。

7. 勾選是否開啟基於PrivateZone的服務發現功能，該功能允許您在叢集VPC內通過服務網域名稱訪問服務。

   > **說明** 在使用勾選該功能前，請確認您已開通PrivateZone服務，參見雲端式解析 PrivateZone的服務發現。
8. 勾選**無伺服器 Kubernetes 服務合約**。
9. 在頁面右側，單擊**建立叢集**，啟動部署。

後續步驟

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 Serverless 叢集。

您可以單擊右側的**管理**，查看叢集的基本資料和串連資訊。

4.2. 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。

操作步驟

1. 登入Container Service管理主控台。
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**刪除**。
4. 在彈出的確認對話方塊中，單擊**確定**。

4.3. 管理和訪問叢集

4.3.1. 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。

操作步驟

1. 從Kubernetes 版本頁面下載最新的 kubectl 用戶端。
2. 安裝和設定 kubectl 用戶端。

   有關詳細資料，參見 安裝和設定 kubectl。
3. 配置叢集憑據。

   您可以在叢集資訊頁面查看叢集憑據。

   i. 登入Container Service管理主控台。
   ii. 在 Kubernetes菜單下，單擊**叢集**進入 Kubernetes 叢集列表頁面。
   iii. 選擇所需的叢集並單擊右側的**管理**。
   iv. 您可以在**串連資訊**處查看叢集的串連地址。

   v. 拷貝叢集憑據到本地檔案中，您可建立並將叢集憑據儲存到 *$HOME/.kube/config*（kubectl 預期憑據所在的位置）。或者命名一個新的檔案，如 */tmp/kubeconfig*，並執行命令 `export KUBECONFIG=/tmp/kubeconfig` 。

vi. 執行上述操作後，您可執行以下命令，確認叢集串連情況。

```
# kubectl cluster-info
Kubernetes Master is running at https://xxxxxx.serverless-1.kubernetes.cn-shanghai.aliyuncs.com:6443
```

後續步驟

配置完成後，您即可使用 kubectl 從本機電腦訪問 Kubernetes 叢集。

4.4. 叢集管理最佳實務

4.4.1. 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。

> **說明** 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。

5.應用管理

5.1. 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，您需要先設定通過 kubectl 串連 Kubernetes 叢集。

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）：

```
root@master # kubectl run nginx --image=registry.cn-hangzhou.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。

```
root@master # 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
```

5.2. 使用鏡像建立應用

前提條件

建立一個 Serverless Kubernetes 叢集。詳情參見建立 Serverless Kubernetes 叢集。

操作步驟

1. 登入Container Service管理主控台。
2. 在 Kubernetes菜單下，單擊左側導覽列中的**應用 > 部署**，進入應用列表頁面。
3. 單擊頁面右上方的**使用鏡像建立**。

4. 設定**應用程式名稱**、**部署叢集**和**命名空間**，並單擊**下一步**進入應用配置頁面。

   如果您不設定**命名空間**，系統會預設使用 default 命名空間。

5. 選擇所要使用的鏡像和鏡像的版本。
   - **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**。您還可以填寫

私人 registry。填寫的格式為 `domainname/namespace/imagename:tag` 。本例中為 nginx。

- **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。

6. 設定**容器數量**。

   本樣本是單容器 Pod，若指定多個容器，會啟動相同數量的 Pod。

7. 設定容器的資源限制和所需資源。

   Serverless Kubernetes 目前處於公測階段，僅支援 2C4G 規格。

   - 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。
   - 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。

   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。

8. 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。

9. 配置容器資訊。

   您可以為運行在 Pod 中的容器配置 Command、Arguments。

   Command 和 Args：若不配置，會使用鏡像預設的設定；若配置，則會覆蓋鏡像裡的預設設定，並且如果只配置 Arguments，容器啟動時，預設的 Command 會執行新的 Arguments。

   Command 和 Arguments 在 Pod 建立之後不可更改。

10. 完成應用配置後，單擊**下一步**，進入訪問設定頁面，即設定一個綁定後端 pod 的服務。

    您可以選擇不建立服務，或者選擇服務類型，目前只支援負載平衡類型。

    - **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。
    - **名稱**：預設會產生一個應用程式名稱尾碼 `svc` 的服務名稱，本例中為 serverless-app-svc，您也可更改服務的名稱。
    - **連接埠映射**：您需要添加服務連接埠和容器連接埠，支援 TCP/UDP 協議。

11. 完成訪問配置後，單擊**建立**。

後續步驟

建立成功後，進入建立完成頁面，會列出應用程式套件含的對象，您可以前往部署列表查看。

您可以看到建立的 serverless-app-svc 出現在部署列表下。

單擊左側導覽列的**應用 > 服務** ，可以看到建立的服務 serverless-app-svc 出現在服務列表下。

在瀏覽器中訪問外部端點，您可訪問 nginx 歡迎頁面。

5.3. 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。

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 Serverless Kubernetes 叢集，參見建立 Serverless Kubernetes 叢集。

步驟 1 建立 deployment

使用鏡像建立一個 deployment，本例中建立 serverless-app-deployment。具體操作參見使用鏡像建立應用。

步驟 2 建立服務

1. 登入 Container Service管理主控台。
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**。

4. 在彈出的建立服務對話方塊中，進行配置。

   - **名稱**：輸入服務的名稱，本例中為 serverless-service。
   - **類型**：選擇服務類型，即服務訪問的方式。目前僅支援負載平衡，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。
   - **關聯部署**：選擇服務要綁定的後端對象，本例中使用 serverless-app-deployment。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。
   - **註解**：為該服務添加一個註解（annotation），配置負載平衡的參數，例如設定 `service.beta.kubernetes.io/alicloud-loadbalancer-bandwidth:20` 表示將該服務的頻寬峰值設定為20Mbit/s，從而控制服務的流量。更多參數請參見負載平衡。
   - **標籤**：您可為該服務添加一個標籤，標識該服務。
5. 單擊**建立**，serverless-service 服務出現在服務列表中。

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

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。

5.4. 刪除服務

您可以通過Container Service控制台快速對服務進行刪除。

前提條件

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

操作步驟

1. 登入Container Service管理主控台。
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 serverless-service），單擊右側的**刪除**。
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。

5.5. 查看容器

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

操作步驟

1. 登入Container Service管理主控台。
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**。

   > **說明** 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。

4. 進入容器組的詳情頁，您可查看該容器組的詳情資訊。

5.6. 查看服務

您在建立應用時，如果配置了外部服務，除了運行容器，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。

操作步驟

1. 登入Container Service管理主控台。
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，查看部署的服務。

   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）。

5.7. 雲端式解析 PrivateZone 的服務發現

阿里雲Serverless Kubernetes已經支援服務發現功能，目前支援內網SLB和Headless Service的服務發現。

關於雲解析PrivateZone

Alibaba Cloud DNS PrivateZone，是基於阿里雲Virtual Private Cloud（Virtual Private Cloud）環境的私人網域名稱解析和管理服務。您能夠在自訂的一個或多個專用網路中將私人網域名稱映射到IP資源地址，同時在其他網路環境無法訪問您的私人網域名稱。

前提條件

1. 需要先開通Alibaba Cloud DNS PrivateZone，在Alibaba Cloud DNS控制台中開通。
2. 您已成功建立一個 Serverless Kubernetes 叢集，參見建立 Serverless Kubernetes 叢集。
3. 您已成功串連到 Kubernetes 叢集，參見通過 kubectl 串連 Kubernetes 叢集。

操作步驟

1. 通過 kubectl 串連到 Kubernetes 叢集，執行如下命令，確認串連的叢集。

   ```
   kubectl cluster-info
   Kubernetes master is running at https://xxxxxx.serverless-1.kubernetes.cn-shanghai.aliyuncs.com:6443
   ```

2. 部署 deployment 和建立 Service。目前僅支援 Intranet Service 和 Headless Service。

   以 Intranet Service 為例。建立樣本*nginx-deployment-basic.yaml*檔案。

   ```
   vim nginx-deployment-basic.yaml
   ```

   範例模板如下所示，在yaml檔案中複製如下yaml代碼，然後執行 `kubectl create -f nginx-deployment-basic.yaml` 命令進行建立。

```
apiVersion: apps/v1beta2 # 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
---
apiVersion: v1
kind: Service
metadata:
  name: nginx-service-intranet       #可通過服務名作為短網域名稱進行訪問
  annotations:                  ##添加註解
    service.beta.kubernetes.io/alicloud-loadbalancer-address-type: intranet
spec:
  ports:
  - port: 80
    protocol: TCP
  selector:
    app: nginx
  type: LoadBalancer
```

您也可建立 Headless Service 類型的服務，樣本模板如下。

```
apiVersion: v1
kind: Service
metadata:
 name: nginx-service-headless
spec:
 ports:
 - port: 80
  protocol: TCP
 selector:
  app: nginx
 clusterIP: None
```

3. 執行以下命令，查看應用的健全狀態。

```
kubectl get svc,pod,deployment
```

4. 登入 Alibaba Cloud DNS控制台。
5. 在左側導覽列中單擊**PrivateZone > Zone 列表**，可看到該列表下自動產生一條記錄。

   您可在該 VPC 網路環境中通過私人網域名稱訪問Service（長網域名稱或者短網域名稱）。

   - 長網域名稱訪問：本例中是 `nginx-service-intranet.$NAMESPACE.svc.cluster.local` ，其中*$NAMESPACE*為Serverless叢集 ID，在控制台上可以看到，也可以在pod的yaml檔案中通過環境變數得到。

     ```
     env:
       - name: NAMESPACE
         valueFrom:
           fieldRef:
             fieldPath: metadata.namespace
     ```

   - 短網域名稱訪問：nginx-service-intranet 或 nginx-service-headless，即在yaml編排中定義的服務名。

更多資訊可參見serverless-k8s-examples

6.配置項

6.1. 建立配置項

在Container Service管理主控台上，您可以通過配置項菜單建立配置項。

操作步驟

1. 登入Container Service管理主控台。
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和命名空間，然後單擊**建立配置項**。

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**編輯配置**檔案 在彈出的對話方塊裡編寫配置項並單擊**確定**。

   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。

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

7.負載平衡管理

7.1. Ingress管理

7.2. SLB Ingress管理

7.2.1. 通過Ingress提供7層服務訪問

在阿里雲Serverless Kubernetes叢集中，我們可以通過LoadBalancer Service對外提供四層服務訪問，同樣您也可以通過Ingress來對外提供七層服務訪問，下面介紹如何在Serverless Kubernetes叢集中提供七層網域名稱服務 (DNS)訪問。

前提條件

- 您已建立一個serverless叢集，叢集的VPC需要配置NAT Gateway，從而訪問外網，下載容器鏡像。
- 您已通過kubectl串連到叢集，參見通過 kubectl 串連 Kubernetes 叢集。

使用說明

1. 不指定SLB執行個體情況下系統會自動幫您產生一個公網SLB執行個體。
2. SLB執行個體預設前端監聽連接埠為80（HTTP協議）和443（HTTPS協議）。
3. SLB執行個體HTTPS認證預設會初始化為第一個建立的Ingress配置的TLS認證，否則會初始化為系統預設認證；您可根據需要自行在SLB控制台上進行修改。
4. 當您指定使用已存在的SLB執行個體時，要求該SLB執行個體規格必須是效能保障型（支援ENI）；同時確保80和443連接埠當前沒有其他服務使用。

注釋說明

注釋	說明
service.beta.kubernetes.io/alicloud-loadbalancer-id	指定已存在的SLB ID

使用預設產生的SLB執行個體

當您不指定SLB執行個體時，系統會在第一個Ingress建立時自動產生一個效能保障型的公網SLB執行個體。

1、 部署測試服務

首先部署一個coffee service和tea service，編排模板如下：

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
 name: coffee
spec:
 replicas: 2
 selector:
  matchLabels:
```

```
    app: coffee
  template:
   metadata:
    labels:
     app: coffee
   spec:
    containers:
    - name: coffee
     image: registry.cn-hangzhou.aliyuncs.com/acs-sample/nginxdemos:latest
     ports:
     - containerPort: 80
---
apiVersion: v1
kind: Service
metadata:
 name: coffee-svc
spec:
 ports:
 - port: 80
  targetPort: 80
  protocol: TCP
 selector:
  app: coffee
 clusterIP: None
---
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
 name: tea
spec:
 replicas: 1
 selector:
  matchLabels:
   app: tea
 template:
  metadata:
   labels:
    app: tea
  spec:
   containers:
   - name: tea
```

```
    - name: tea
      image: registry.cn-hangzhou.aliyuncs.com/acs-sample/nginxdemos:latest
      ports:
      - containerPort: 80
---
apiVersion: v1
kind: Service
metadata:
 name: tea-svc
 labels:
spec:
 ports:
 - port: 80
   targetPort: 80
   protocol: TCP
 selector:
   app: tea
 clusterIP: None
```

```
$ kubectl apply -f cafe-service.yaml            #在cafe-service.yaml檔案中輸入上面的模板
deployment "coffee" created
service "coffee-svc" created
deployment "tea" created
service "tea-svc" created
# 部署完成後
$ kubectl get svc,deploy
NAME          TYPE     CLUSTER-IP  EXTERNAL-IP  PORT(S)  AGE
svc/coffee-svc  ClusterIP  <none>     <none>     80/TCP   1m
svc/tea-svc     ClusterIP  <none>     <none>     80/TCP   1m
NAME       DESIRED  CURRENT  UP-TO-DATE  AVAILABLE  AGE
deploy/coffee  2      2      2       2       1m
deploy/tea     1      1      1       1       1m
```

2、配置 Ingress

通過Ingress配置coffee service和tea service對外暴露的網域名稱和Path路徑：

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
 name: cafe-ingress
spec:
 rules:
 # 配置七層網域名稱
 - host: foo.bar.com
  http:
   paths:
   # 配置Context Path
   - path: /tea
    backend:
     serviceName: tea-svc
     servicePort: 80
   # 配置Context Path
   - path: /coffee
    backend:
     serviceName: coffee-svc
     servicePort: 80
```

```
$ kubectl apply -f cafe-ingress.yaml
ingress "cafe-ingress" created
# 部署完成後，ADDRESS為自動產生的SLB執行個體IP
$ kubectl get ing
NAME      HOSTS     ADDRESS      PORTS   AGE
cafe-ingress  foo.bar.com  139.224.76.211  80     1m
```

3、測試服務訪問

> **說明** 目前我們需要自行將網域名稱解析到SLB執行個體IP上

本例中在*hosts*中添加一條DNS網域名稱解析規則，用於測試服務訪問。建議您在工作環境中對網域名稱進行備案。

```
139.224.76.211  foo.bar.com
```

通過瀏覽器測試訪問coffee服務。

通過命令列方式測試訪問coffee服務。

```
curl -H "Host: foo.bar.com" http://139.224.76.211/coffee
```

通過瀏覽器測試訪問tea服務。

通過命令列方式測試訪問tea服務。

```
curl -H "Host: foo.bar.com" http://139.224.76.211/tea
```

使用指定的SLB執行個體

我們可以通過注釋 `service.beta.kubernetes.io/alicloud-loadbalancer-id` 來指定使用已存在的SLB執行個體，但要求該SLB執行個體必須為效能保障型規格（支援ENI）。

> **說明** 系統會自動初始化SLB執行個體的80和443連接埠，請確保當前沒有其他服務使用。

1、 部署測試服務

首先部署一個tomcat測試應用，編排模板如下：

```
apiVersion: extensions/v1beta1
kind: Deployment
metadata:
 name: tomcat
spec:
 replicas: 1
 selector:
  matchLabels:
   run: tomcat
 template:
  metadata:
   labels:
    run: tomcat
  spec:
   containers:
   - image: tomcat:7.0
    imagePullPolicy: Always
    name: tomcat
    ports:
    - containerPort: 8080
     protocol: TCP
   restartPolicy: Always
---
apiVersion: v1
kind: Service
metadata:
 name: tomcat
spec:
 ports:
 - port: 8080
  protocol: TCP
  targetPort: 8080
 selector:
  run: tomcat
 clusterIP: None
```

```
$ kubectl apply -f tomcat-service.yml          #在tomcat-service.yaml中輸入上面的模板
deployment "tomcat" created
service "tomcat" created
# 部署完成後
$ kubectl get svc,deploy tomcat
NAME      TYPE     CLUSTER-IP  EXTERNAL-IP  PORT(S)   AGE
svc/tomcat  ClusterIP  <none>    <none>     8080/TCP  1m
NAME       DESIRED  CURRENT  UP-TO-DATE  AVAILABLE  AGE
deploy/tomcat  1     1     1       1       1m
```

2、申請SLB執行個體

您需要在叢集同Region下自行申請一個**效能保障型**SLB執行個體（如slb.s2.small），可以是私網也可以是公網（依據具體需求）。本例中申請一個公網SLB執行個體，記錄SLB執行個體的ID。

3、配置TLS認證

您需要配置TLS認證實現HTTPS訪問。

> **說明** 系統自動依據第一個建立的Ingress的TLS認證來初始化SLB的HTTPS預設認證，若需要修改HTTPS預設認證，可在SLB控制台自行修改；若需配置多個認證，可在SLB控制台HTTPS監聽擴充網域名稱下自行添加。

```
# 產生測試TLS認證
$ openssl req -x509 -nodes -days 365 -newkey rsa:2048 -keyout tls.key -out tls.crt -subj "/CN=bar.foo.com/O=
bar.foo.com"# 建立TLS認證Secret
$ kubectl create secret tls cert-example --key tls.key --cert tls.crt
secret "cert-example" created
# 查看建立TLS認證
$ kubectl get secret cert-example
NAME       TYPE          DATA    AGE
cert-example   kubernetes.io/tls  2      12s
```

4、配置 Ingress

通過Ingress配置tomcat service對外暴露的網域名稱和Path路徑，編排模板如下：

```
apiVersion: extensions/v1beta1
kind: Ingress
metadata:
 name: tomcat-ingress
 annotations:
  # 配置使用指定的SLB執行個體（SLB ID）
  service.beta.kubernetes.io/alicloud-loadbalancer-id: lb-xxxxxxxxxx        ##替換為你的SLB ID
spec:
 tls:
 - hosts:
  - bar.foo.com
  # 配置TLS認證
  secretName: cert-example
 rules:
 # 配置七層網域名稱
 - host: bar.foo.com
  http:
   paths:
   # 配置Context Path
   - path: /
    backend:
     serviceName: tomcat
     servicePort: 8080
```

```
$ kubectl apply -f tomcat-ingress.yml                #在tomcat-ingress.yaml中輸入上面的模板
ingress "tomcat-ingress" created
# 部署完成後，ADDRESS為指定的SLB IP地址
$ kubectl get ing tomcat-ingress
NAME          HOSTS     ADDRESS     PORTS   AGE
tomcat-ingress  bar.foo.com  47.101.20.67  80, 443  1m
```

5、測試服務訪問

> **說明** 目前我們需要自行將網域名稱解析到SLB執行個體IP上。

本例中在*hosts*中添加一條DNS網域名稱解析規則，用於測試服務訪問。建議您在工作環境中對網域名稱進行備案。

```
47.101.20.67  bar.foo.com
```

通過瀏覽器測試訪問tomcat服務：

通過命令列方式測試訪問tomcat服務：

```
curl -k -H "Host: bar.foo.com" https://47.101.20.67
```

7.3. 网络管理最佳实践

8.日誌管理