

ALIBABA CLOUD

阿里云

服务网格
多集群

文档版本：20210223

 阿里云

法律声明

阿里云提醒您 在阅读或使用本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

通用约定

| 格式 | 说明 | 样例 |
|--|------------------------------------|---|
|  危险 | 该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。 |  危险 重置操作将丢失用户配置数据。 |
|  警告 | 该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。 |  警告 重启操作将导致业务中断，恢复业务时间约十分钟。 |
|  注意 | 用于警示信息、补充说明等，是用户必须了解的内容。 |  注意 权重设置为0，该服务器不会再接受新请求。 |
|  说明 | 用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。 |  说明 您也可以通过按Ctrl+A选中全部文件。 |
| > | 多级菜单递进。 | 单击设置> 网络> 设置网络类型。 |
| 粗体 | 表示按键、菜单、页面名称等UI元素。 | 在结果确认页面，单击确定。 |
| Courier字体 | 命令或代码。 | 执行 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。 |
| 斜体 | 表示参数、变量。 | <code>bae log list --instanceid</code> <i>Instance_ID</i> |
| [] 或者 [a b] | 表示可选项，至多选择一个。 | <code>ipconfig [-all -t]</code> |
| { } 或者 {a b} | 表示必选项，至多选择一个。 | <code>switch {active stand}</code> |

目录


| | |
|-----------------------------|----|
| 1.通过ASM管理ASK集群上的ECI Pod应用 | 05 |
| 2.通过ASM管理ACK虚拟节点上的ECI Pod应用 | 06 |
| 3.通过ASM管理外部注册Kubernetes集群应用 | 08 |
| 4.部署应用示例到包含同VPC多集群的ASM实例 | 10 |
| 5.通过ASM实现跨地域多集群的全局负载均衡 | 23 |
| 6.ASM VM PROXY安装部署指南 | 25 |

1.通过ASM管理ASK集群上的ECI Pod应用

阿里云容器服务基于虚拟节点和ECI提供了多种Serverless Container产品形态，例如Serverless Kubernetes（ASK）集群实现了Kubernetes与弹性容器实例ECI的无缝连接。本文介绍如何在服务网格ASM中管理运行在ASK上的ECI Pod应用。

前提条件

- 已创建ASK集群。具体操作，请参见[创建Serverless Kubernetes集群](#)。

 **说明** 创建ASK集群时，您需要启用PrivateZone或者CoreDNS，确保ASK集群支持服务发现功能。

- 已添加ASK集群到ASM实例。具体操作，请参见[添加集群到ASM实例](#)。


启用自动注入

在ASM控制台中启动自动注入功能，可以在创建Pod的过程中，将Sidecar自动注入Proxy容器，以实现数据平面的网格化。

- 登录[ASM控制台](#)。
- 在左侧导航栏，选择**服务网格 > 网格管理**。
- 在**网格管理**页面，找到待配置的实例，单击实例的名称或在操作列中单击**管理**。
- 在**控制平面区域Namespace**页签下找到待注入的命名空间（本示例为default命名空间，如果不存在需要先创建），在**自动注入**列单击**启用Sidecar自动注入**。
- 在**确认对话框**，单击**确定**。

创建ECI Pod应用

在ASK集群中所有Pod都是ECI Pod，无需给Pod配置特殊标签。

 **说明** 创建ECI Pod应用后，ASM可以通过Sidecar对ECI Pod应用进行数据平面化管理。

- 执行以下命令，部署Nginx应用。

```
kubectl run nginx -n default --image nginx
```

- 执行以下命令，查看虚拟节点上的Pod信息。

```
kubectl get pod -n default -o wide|grep virtual-kubelet
```

常见问题

为什么ASK集群无法启用服务发现？

Sidecar Proxy容器日志中可以看到解析istiod.istio-system服务到错误的IP地址。说明您未开通云解析PrivateZone，您可以通过[提交工单](#)开启PrivateZone或者安装CoreDNS。

2.通过ASM管理ACK虚拟节点上的ECI Pod应用

阿里云容器服务基于虚拟节点和ECI提供了多种Serverless Container产品形态，例如通过部署ACK虚拟节点组件创建ECI Pod实现了Kubernetes与弹性容器实例ECI的无缝连接。您可以灵活动态地按需创建ECI Pod，免去集群容量规划的麻烦。本文介绍如何在服务网格ASM中管理运行在ACK虚拟节点上的ECI Pod应用。

前提条件


- ASM实例为v1.7.5.41-ge61a01c3-aliyun或以上版本。
- 已在ACK集群中部署ack-virtual-node组件，并确保运行正常。具体操作，请参见[步骤一：在ACK集群中部署ack-virtual-node组件](#)。
- 已添加ACK集群到ASM实例。具体操作，请参见[添加集群到ASM实例](#)。

启用自动注入

在ASM控制台中启动自动注入功能，可以在创建Pod的过程中，将Sidecar自动注入Proxy容器，以实现数据平面的网格化。

1. 登录[ASM控制台](#)。
2. 在左侧导航栏，选择[服务网格 > 网格管理](#)。
3. 在[网格管理](#)页面，找到待配置的实例，单击实例的名称或在操作列中单击[管理](#)。
4. 在控制平面区域Namespace页签下找到待注入的命名空间（本示例中包括2个命名空间default和vk，如果不存在需要先创建），在自动注入列单击启用Sidecar自动注入。
5. 在确认对话框，单击确定。

创建ECI Pod应用

 **说明** 创建ECI Pod应用后，ASM可以通过Sidecar对ECI Pod应用进行数据平面化管理。

通过配置Pod标签的方式创建ECI Pod应用

给Pod添加 `label alibabacloud.com/eci=true` 的标签，Pod将以ECI方式运行，并且所在的节点是虚拟节点。

1. 执行以下命令，确认default命名空间已包含 `istio-injection=enabled` 标签。

```
kubectl get ns default --show-labels
```

预期输出：

```
NAME    STATUS   AGE   LABELS
default Active   84d   istio-injection=enabled,provider=asm
```

2. 执行以下命令，部署Nginx应用。

```
kubectl run nginx -n default --image nginx -l alibabacloud.com/eci=true
```

3. 执行以下命令，查看虚拟节点上的Pod信息。

```
kubectl get pod -n default -o wide|grep virtual-kubelet
```

通过配置Namespace标签的方式创建ECI Pod应用

给Pod所在的命名空间添加 `label alibabacloud.com/eci=true` 标签，Pod将以ECI方式运行，并且所在的节点是虚拟节点。

1. 执行以下命令，确认vk命名空间已包含 `istio-injection=enabled` 标签。

```
kubectl get ns default --show-labels
```

预期输出：

```
NAME    STATUS   AGE   LABELS
default Active   84d   istio-injection=enabled,provider=asm
```

2. 执行以下命令，为vk命名空间添加标签。

```
kubectl label namespace vk alibabacloud.com/eci=true
```

3. 执行以下命令，部署Nginx应用。

```
kubectl -n vk run nginx --image nginx
```

4. 执行以下命令，查看虚拟节点上的Pod信息。

```
kubectl -n vk get pod -o wide|grep virtual-kubelet
```

3.通过ASM管理外部注册Kubernetes集群应用

您可以通过容器服务控制台注册外部Kubernetes集群，并通过服务网格ASM进行应用管理。

前提条件

- 一个可以访问公网地址的外部Kubernetes集群，并能够通过容器服务控制台进行注册，请参见[注册外部Kubernetes集群](#)。
- 已开通服务网格功能，请参见[创建ASM实例](#)。

操作步骤

1. 登录[ASM控制台](#)。
2. 在左侧导航栏，选择[服务网格 > 网格管理](#)。
3. 在[网格管理](#)页面，单击[创建新网格](#)。
4. 在[创建新网格](#)页面，填写网格的名称、选择相应的地域、专有网络VPC及交换机。

② 说明

- 根据注册的外部接入集群所在的地域信息，选择与之更近距离的地域。
- 在已有VPC列表中选中注册的外部接入集群所使用的VPC实例。
- 在交换机列表中选择所需的交换机。如果没有您需要的交换机，单击[创建交换机](#)进行创建，请参见[使用交换机](#)。

5. 设置是否开放使用公网地址暴露API Server。

② 说明

ASM实例的运行基于Kubernetes运行时，可以通过API Server定义执行各种网格资源，如虚拟服务、目标规则或者Istio网关等。

- 如果选择开放，会创建一个EIP，并挂载到私网SLB上。API Server会暴露6443端口，您可以在公网通过kubecfg来连接和操作集群，从而定义网格资源。
- 如果选择不开放，则不会创建EIP，您只能在VPC下通过kubecfg来连接和操作集群，从而定义网格资源。

6. 设置使用公网地址暴露Istio Pilot。

② 说明

请勾选此选项，否则注册的外部接入集群中的Pod无法连接到Pilot，导致应用无法正常使用。

7. 其他设置可保持默认值，暂不启用。单击[确定](#)，开始创建实例。

② 说明

一个ASM实例的创建时间约为2到3分钟。

8. 选择刚刚创建的实例，单击[管理](#)。
9. 在网格管理页面中的[数据平面](#)区域，单击[集群页签](#)下的[添加](#)。

10. 在集群添加页面，勾选需要添加的外部接入集群，然后单击**确定**。

说明 添加集群之后，ASM实例的状态变为**更新中**。数秒之后（时长与添加的外部接入集群网络访问速度有关），单击页面右上方的**刷新**，网格状态会变为**运行中**。在**数据平面**区域，可以查看已添加集群的信息。

11. 在**数据平面**区域，选择**入口网关服务**，单击**部署**。

i. 在**部署集群**下拉列表中选择要部署入口网关的外部接入集群。

ii. 选择**负载均衡**的类型，公网访问或内网访问。

说明 针对不同的外部接入集群，可能支持不同的负载均衡类型，例如不支持内网访问，需要根据实际情况选择类型。如果不支持负载均衡类型的服务，那么可以暂选择公网类型，待创建之后，修改对应的YAML内容重新指定服务类型，例如指定为Nodeport或者ClusterIP类型。

目前针对不同的外部接入集群，只能选择**新建负载均衡**。

iii. 配置**端口映射**。

说明

- 建议容器端口与服务端口一致，并在Istio网关资源定义中启用了该端口。
- 控制台默认提供了4个Istio常用的端口，但并不意味着必须从中选择，您可以根据需要自行添加或删除端口。

12. 单击**确定**，完成部署。成功添加入口网关后，可登录到外部接入集群查看详情。

部署应用到外部接入集群

使用Kubectll命令行或者登录到外部接入集群控制台（如果存在），部署应用到外部接入集群中，请参见[部署应用到ASM实例](#)。

定义Istio资源

在服务网格ASM控制台中可以定义Istio资源，请参见[定义Istio资源](#)。

4.部署应用示例到包含同VPC多集群的ASM实例

通过服务网格ASM，可以将一个应用的服务组件部署在同VPC的多个集群上。本文以Bookinfo应用为例，介绍如何将该应用部署到包含两个集群的ASM实例。

前提条件

- 在同一VPC下创建两个ACK集群（本例中m1c1和m1c2），详情请参见[创建Kubernetes专有版集群](#)。
- 创建一个ASM实例（本例中mesh1），详情请参见[创建ASM实例](#)。

步骤一：修改集群的安全组名称

将两个集群对应的安全组名称修改为易于辨识的名称，本例中为m1c1-sg和m1c2-sg。

1. 登录[ECS管理控制台](#)。
2. 在左侧导航栏，选择[网络与安全](#) > [安全组](#)。
3. 在顶部菜单栏左上角处，选择地域。
4. 在安全组列表页面中，找到需要修改的安全组，单击操作列下的[修改](#)。
5. 在弹出的对话框中，修改安全组名称和描述。
6. 单击[确定](#)。

修改后的名称，如下图所示。



| 安全组ID/名称 | 标签 | 所属专有网络 | 相关实例 | 可加入IP数 | 网络类型(全部) | 安全组类型 | 创建时间 | 描述 | 操作 |
|-----------------------------------|----|--------------------------------------|------|--------|----------|-------|-------------------|---------------------------|--|
| sg-8vbgqxxdwang4wh23zq m1c2-sg | | vpc-8vbgpnrbx1jzrcieenuh5 meshvpc | 6 | 1994 | 专有网络 | 普通安全组 | 2019年12月31日 14:01 | This is used by kubern... | 修改 克隆 还原规则 管理实例 配置规则 管理弹性网卡 |
| sg-8vben21m59x4nm6pgkg m1c1-sg | | vpc-8vbgpnrbx1jzrcieenuh5 meshvpc | 3 | 1976 | 专有网络 | 普通安全组 | 2019年12月31日 13:55 | security group of ACS ... | 修改 克隆 还原规则 管理实例 配置规则 管理弹性网卡 |

步骤二：配置集群的互访联通性

为了使两个集群能够互相访问，需要为彼此添加安全组访问规则。

1. 在m1c1-sg安全组配置界面，添加以m1c2-sg为授权对象的访问规则。详情请参见[添加安全组规则](#)。



| 授权策略 | 协议类型 | 端口范围 | 授权类型(全部) | 授权对象 | 描述 | 优先级 | 创建时间 | 操作 | |
|--------------------------|------|------|----------|-------|-------------------------------------|-----|------|-------------------|--|
| <input type="checkbox"/> | 允许 | 全部 | -1/-1 | 安全组访问 | sg-8vbgpnrbx1jzrcieenuh5 m1c2-sg | - | 1 | 2019年12月31日 14:37 | 修改 克隆 删除 |

2. 在m1c2-sg安全组规则配置界面，添加以m1c1-sg为授权对象的访问规则。



| 授权策略 | 协议类型 | 端口范围 | 授权类型(全部) | 授权对象 | 描述 | 优先级 | 创建时间 | 操作 | |
|--------------------------|------|------|----------|-------|-----------------------------------|-----|------|-------------------|--|
| <input type="checkbox"/> | 允许 | 全部 | -1/-1 | 安全组访问 | sg-8vben21m59x4nm6pgkg m1c1-sg | - | 1 | 2019年12月31日 14:37 | 修改 克隆 删除 |



步骤三：添加集群到ASM实例并部署集群的入口网关

将两个集群添加到ASM实例后，由于两个集群已实现访问互通，因此只需为一个集群部署入口网关。

1. 将两个集群添加到ASM实例，详情请参见[添加集群到ASM实例](#)。
2. 为m1c1集群部署入口网关，详情请参见[添加入口网关服务](#)。

步骤四：部署Bookinfo应用

为了演示ASM跨集群的应用部署能力，Bookinfo应用的不同微服务分别部署在两个集群上。

1. 在m1c2中部署不包含review-v3 deployment的Bookinfo应用，详情请参见[部署应用到ASM实例](#)。

说明 Review-v3 deployment对应的功能是书评中显示红色星。

对应的Yaml文件内容如下所示：

```
#####
#####
# Details service
#####
#####
apiVersion: v1
kind: Service
metadata:
  name: details
  labels:
    app: details
    service: details
spec:
  ports:
  - port: 9080
    name: http
  selector:
    app: details
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: bookinfo-details
  labels:
    account: details
---
apiVersion: apps/v1
```

```
kind: Deployment
metadata:
  name: details-v1
  labels:
    app: details
    version: v1
spec:
  replicas: 1
  selector:
    matchLabels:
      app: details
      version: v1
  template:
    metadata:
      labels:
        app: details
        version: v1
    spec:
      serviceAccountName: bookinfo-details
      containers:
        - name: details
          image: docker.io/istio/examples-bookinfo-details-v1:1.15.0
          imagePullPolicy: IfNotPresent
          ports:
            - containerPort: 9080
---
#####
#####
# Ratings service
#####
#####
apiVersion: v1
kind: Service
metadata:
  name: ratings
  labels:
    app: ratings
    service: ratings
spec:
  ports:
    - port: 9080
```

```

name: http
selector:
  app: ratings
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: bookinfo-ratings
labels:
  account: ratings
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: ratings-v1
labels:
  app: ratings
  version: v1
spec:
  replicas: 1
  selector:
    matchLabels:
      app: ratings
      version: v1
  template:
    metadata:
      labels:
        app: ratings
        version: v1
    spec:
      serviceAccountName: bookinfo-ratings
      containers:
      - name: ratings
        image: docker.io/istio/examples-bookinfo-ratings-v1:1.15.0
        imagePullPolicy: IfNotPresent
        ports:
        - containerPort: 9080
---
#####
#####
# Reviews service

```

```
# REVIEWS SERVICE
#####
#####
apiVersion: v1
kind: Service
metadata:
  name: reviews
  labels:
    app: reviews
    service: reviews
spec:
  ports:
    - port: 9080
      name: http
  selector:
    app: reviews
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: bookinfo-reviews
  labels:
    account: reviews
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: reviews-v1
  labels:
    app: reviews
    version: v1
spec:
  replicas: 1
  selector:
    matchLabels:
      app: reviews
      version: v1
  template:
    metadata:
      labels:
        app: reviews
```

```
    version: v1
  spec:
    serviceAccountName: bookinfo-reviews
    containers:
      - name: reviews
        image: docker.io/istio/examples-bookinfo-reviews-v1:1.15.0
        imagePullPolicy: IfNotPresent
    ports:
      - containerPort: 9080
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: reviews-v2
  labels:
    app: reviews
    version: v2
spec:
  replicas: 1
  selector:
    matchLabels:
      app: reviews
      version: v2
  template:
    metadata:
      labels:
        app: reviews
        version: v2
    spec:
      serviceAccountName: bookinfo-reviews
      containers:
        - name: reviews
          image: docker.io/istio/examples-bookinfo-reviews-v2:1.15.0
          imagePullPolicy: IfNotPresent
      ports:
        - containerPort: 9080
# ---
# apiVersion: apps/v1
# kind: Deployment
# metadata:
#   name: reviews-v3
```

```
# labels:
# app: reviews
# version: v3
# spec:
# replicas: 1
# selector:
# matchLabels:
# app: reviews
# version: v3
# template:
# metadata:
# labels:
# app: reviews
# version: v3
# spec:
# serviceAccountName: bookinfo-reviews
# containers:
# - name: reviews
# image: docker.io/istio/examples-bookinfo-reviews-v3:1.15.0
# imagePullPolicy: IfNotPresent
# ports:
# - containerPort: 9080
---
#####
#####
# Productpage services
#####
#####
apiVersion: v1
kind: Service
metadata:
  name: productpage
  labels:
    app: productpage
    service: productpage
spec:
  ports:
  - port: 9080
    name: http
  selector:
    app: productpage
```



```
  app: productpage
  ---
  apiVersion: v1
  kind: ServiceAccount
  metadata:
    name: bookinfo-productpage
  labels:
    account: productpage
  ---
  apiVersion: apps/v1
  kind: Deployment
  metadata:
    name: productpage-v1
  labels:
    app: productpage
    version: v1
  spec:
    replicas: 1
    selector:
      matchLabels:
        app: productpage
        version: v1
    template:
      metadata:
        labels:
          app: productpage
          version: v1
      spec:
        serviceAccountName: bookinfo-productpage
        containers:
          - name: productpage
            image: docker.io/istio/examples-bookinfo-productpage-v1:1.15.0
            imagePullPolicy: IfNotPresent
            ports:
              - containerPort: 9080
  ---
```

2. 在m1c1中部署review-v3以及rating service（review依赖的服务）。

对应的YAML文件内容如下所示：

```
#####
#####
```

```
# Reviews service
#####
#####
apiVersion: v1
kind: Service
metadata:
  name: reviews
  labels:
    app: reviews
    service: reviews
spec:
  ports:
    - port: 9080
      name: http
  selector:
    app: reviews
---
apiVersion: v1
kind: ServiceAccount
metadata:
  name: bookinfo-reviews
  labels:
    account: reviews
---
apiVersion: apps/v1
kind: Deployment
metadata:
  name: reviews-v3
  labels:
    app: reviews
    version: v3
spec:
  replicas: 1
  selector:
    matchLabels:
      app: reviews
      version: v3
  template:
    metadata:
      labels:
        app: reviews
```

```
app: reviews
  version: v3
spec:
  serviceAccountName: bookinfo-reviews
  containers:
  - name: reviews
    image: docker.io/istio/examples-bookinfo-reviews-v3:1.15.0
    imagePullPolicy: IfNotPresent
  ports:
  - containerPort: 9080
---
#####
#####
# Ratings service
#####
#####
apiVersion: v1
kind: Service
metadata:
  name: ratings
  labels:
    app: ratings
    service: ratings
spec:
  ports:
  - port: 9080
    name: http
  selector:
    app: ratings
```

步骤五：添加虚拟服务和Istio网关

1. 在ASM实例的default命名空间下新建一个虚拟服务，名为bookinfo，详情请参见[定义Istio资源](#)。
对应的Yaml文件内容如下所示：

```
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: bookinfo
spec:
  hosts:
  - "*"
  gateways:
  - bookinfo-gateway
  http:
  - match:
    - uri:
        exact: /productpage
    - uri:
        prefix: /static
    - uri:
        exact: /login
    - uri:
        exact: /logout
    - uri:
        prefix: /api/v1/products
  route:
  - destination:
      host: productpage
      port:
        number: 9080
```

2. 在ASM实例的default命名空间下新建一个Istio网关，名为bookinfo-gateway，详情参见[定义Istio资源](#)。
对应的Yaml文件内容如下所示：

```
apiVersion: networking.istio.io/v1alpha3
kind: Gateway
metadata:
  name: bookinfo-gateway
spec:
  selector:
    istio: ingressgateway # use istio default controller
  servers:
  - port:
      number: 80
      name: http
      protocol: HTTP
    hosts:
      - "*"

```

访问productpage页面，刷新页面时会轮流显示reviews的3个版本。虽然review-v3和其他服务不在同一个集群中，也可以正常显示。

（可选）步骤六：指定reviews总是用v3版本

通过定义目标规则和虚拟服务，可以定义Bookinfo应用的微服务部署策略。本例中将指定Bookinfo总是使用review v3版本。

1. 在ASM实例的default命名空间下新建一个目标规则，名为reviews。

Yaml文件的内容如下所示：

```
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: reviews
spec:
  host: reviews
  subsets:
  - name: v1
    labels:
      version: v1
  - name: v2
    labels:
      version: v2
  - name: v3
    labels:
      version: v3

```

2. 在ASM实例的default命名空间下新建一个虚拟服务，名为reviews。

对应的Yaml文件内容如下所示:

```
apiVersion: networking.istio.io/v1alpha3
kind: VirtualService
metadata:
  name: reviews
spec:
  hosts:
  - reviews
  http:
  - route:
    - destination:
        host: reviews
      subset: v3
```

此时访问productpage页面, reviews将始终使用v3版本, 即书评中为红色星。

The Comedy of Errors

Summary: [Wikipedia Summary](#): The Comedy of Errors is one of **William Shakespeare's** early plays. It is his shortest and one of his most farcical comedies, with a major part of the humour coming from slapstick and mistaken identity, in addition to puns and word play.

Book Details

Type:
paperback
Pages:
200
Publisher:
PublisherA
Language:
English
ISBN-10:
1234567890
ISBN-13:
123-1234567890

Book Reviews

An extremely entertaining play by Shakespeare. The slapstick humour is refreshing!
— Reviewer1
★★★★★

Absolutely fun and entertaining. The play lacks thematic depth when compared to other plays by Shakespeare.
— Reviewer2
★★★★☆

5.通过ASM实现跨地域多集群的全局负载均衡

服务网格ASM通过Envoy代理为应用服务提供了全局负载均衡能力，其能够以最佳方式将流量路由至某个服务位于多个地域的应用实例。本文以Bookinfo应用为例，介绍ASM跨地域多集群的全局负载均衡的能力。

前提条件

- 创建一个ASM实例（本例中mesh1），详情请参见[创建ASM实例](#)。
- 在集群创建之前做好网络规划，保证两个集群所使用的交换机IPv4网段不能重叠、Pod网络CIDR与Service CIDR不能重叠。
- 在两个VPC下分别创建两个ACK集群（本例中m1c1和m1c2），详情请参见[创建Kubernetes专有版集群](#)。
- 开通[云企业网（CEN）](#)。
- 两个VPC通过CEN实现互通，详情请参见[教程概览](#)。
- 在云企业网控制台中发布每一个网段，以确保这些网段在CEN中的状态为已发布并且不存在网段冲突。详情请参见[发布路由到云企业网](#)。

背景信息

- 本文所使用的网络规划如下所示：

| 对象 | 集群m1c1 | 集群m1c2 |
|---------------|----------------|----------------|
| VPC、交换机IPv4网段 | 192.168.0.0/24 | 192.168.1.0/24 |
| Pod网络CIDR | 172.22.0.0/16 | 172.30.0.0/16 |
| Service CIDR | 172.23.0.0/20 | 172.31.0.0/20 |

- 在ACK集群中，Pod的位置是通过其部署节点上的region和zone标签确定的，详情请参见[Well-Known Labels, Annotations and Taints](#)。ACK集群默认已经配置了这些信息。

步骤一：部署应用示例到跨地域的多个集群上

部署Bookinfo应用到跨地域的不同VPC的多个集群上。

步骤二：启用服务就近访问

为了确定服务对应的任一实例何时异常，Envoy代理需要在每个服务的目标规则中进行异常检测配置，详情请参见[DestinationRule](#)。在此情况下，ASM会根据发送请求的Envoy代理位置，针对目标服务的工作负载实例，进行优先级排序。当所有应用实例都正常时，请求将保留在同一位置，即保持服务就近访问。

1. 登录[ASM控制台](#)。
2. 在左侧导航栏，选择[服务网格 > 网格管理](#)。
3. 在[网格管理](#)页面，找到待配置的实例，单击实例的名称或在操作列中单击[管理](#)。
4. 在[基本信息](#)页面单击右上角的[功能设置](#)。
5. 在[功能设置更新](#)页面中勾选[启用服务就近访问](#)。
6. 单击[确定](#)。
在[基本信息](#)页面可以看到[服务就近访问](#)的状态变为[开启](#)。

此时没有在服务reviews对应的目标规则中定义异常检测配置，因此访问productpage页面，刷新页面时会轮流显示reviews的3个版本，即：

- v1版本不会调用Ratings服务。
- v2版本会调用Ratings服务，并使用1到5个黑色星形图标来显示评分信息。
- v3版本会调用Ratings服务，并使用1到5个红色星形图标来显示评分信息。

步骤三：定义异常检测配置

启用了服务就近访问的负载均衡之后，需要定义目标规则中的异常检测配置，Envoy代理才能确定应用实例是否正常，进而实现服务就近访问。

1. 登录ASM控制台。
2. 在左侧导航栏，选择服务网格 > 网格管理。
3. 在网格管理页面，找到待配置的实例，单击实例的名称或在操作列中单击管理。
4. 在控制平面区域，选择DestinationRule页签，然后单击新建。
5. 在新建页面，输入DestinationRule的信息，单击确定。
 - i. 在命名空间下拉列表中，选择待新建DestinationRule的命名空间。
 - ii. 在文本框中，输入DestinationRule的配置信息。

对应的Yaml文件内容如下所示：


```
apiVersion: networking.istio.io/v1alpha3
kind: DestinationRule
metadata:
  name: reviews
spec:
  host: reviews
  trafficPolicy:
    outlierDetection:
      consecutiveErrors: 5
      interval: 2m
      baseEjectionTime: 5m
```

在DestinationRule页签，可以看到新建的DestinationRule。

执行结果

由于productpage服务与reviews服务的版本v1和v2位于同一个位置，而v3则是位于不同的地域，因此按照就近访问原则，在reviews服务版本v1和v2正常情况下，流量不会路由到v3。访问productpage页面，刷新页面时会轮流显示reviews的2个版本，即：

- v1版本不会调用Ratings服务。
- v2版本会调用Ratings服务，并使用1到5个黑色星形图标来显示评分信息。

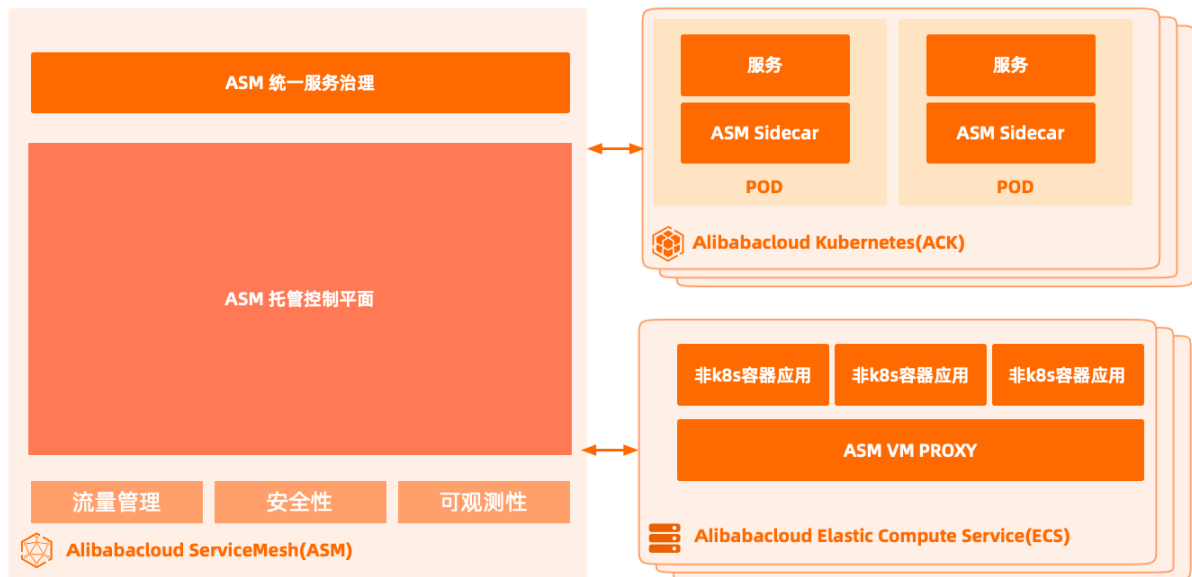
 **说明** 假设位于集群m1c1的reviews服务出现故障（可以通过扩容Pod数量为0进行模拟），再次访问productpage页面，则可以看到页面中的reviews服务调用了v3版本，即使用1到5个红色星形图标来显示评分信息。

6.ASM VM PROXY安装部署指南

ASM-VM-PROXY是连接ASM和虚拟机的工具，负责与ASM通信并执行网格化操作。本文介绍如何安装部署ASM-VM-PROXY。

背景信息

ASM-VM-PROXY是连接ASM和虚拟机的工具，运行在虚拟机节点中，负责与ASM通信获取xDS信息、劫持非容器应用流量并执行网格化操作，如上报链路追踪数据、请求安全认证等。为了支持跨操作系统，以Docker镜像的方式进行分发。



安装前环境准备

为了演示跨操作系统的功能，本文使用三种最常见的Linux分发版本创建ECS实例，详情见下表。

| 名称 | 虚拟机名称 | 公网IP | 内网IP | OS-release版本 | 内核版本 |
|------|---------------------|---------------|--------------|---------------|------------------------------|
| 虚拟机1 | asm-vm-proxy-aliOS | 59.110.**.** | 192.168.*.** | alinux 2.1903 | 4.19.91-21.al7.x86_64 |
| 虚拟机2 | asm-vm-proxy-centos | 39.106.1**.** | 192.168.*.** | centos 8 | 4.18.0-193.14.2.el8_2.x86_64 |
| 虚拟机3 | asm-vm-proxy-ubuntu | 47.94.1**.** | 192.168.*.** | ubuntu 20.04 | 5.4.0-31-generic |

步骤一：安装Docker

ASM-VM-PROXY以Docker镜像的方式进行分发，因此需要在虚拟机上安装Docker。

1. 在虚拟机1中安装v19.03.12版本Docker。

```
yum install -y https://download.docker.com/linux/fedora/30/x86_64/stable/Packages/containerd.io-1.2.6-3.3.fc30.x86_64.rpm
yum install -y yum-utils
yum-config-manager \
  --add-repo \
  https://download.docker.com/linux/centos/docker-ce.repo
yum update && yum install -y docker-ce docker-ce-cli containerd.io
systemctl start docker
docker version
```

2. 在虚拟机2中安装v19.03.12版本Docker。


```
yum install -y https://download.docker.com/linux/fedora/30/x86_64/stable/Packages/containerd.io-1.2.6-3.3.fc30.x86_64.rpm
yum install -y yum-utils
yum-config-manager \
  --add-repo \
  https://download.docker.com/linux/centos/docker-ce.repo
yum update && yum install -y docker-ce docker-ce-cli containerd.io
systemctl start docker
docker version
```

3. 在虚拟机3中安装v19.03.8版本Docker。

```
apt-get update && apt-get install -y docker.io
docker version
```

步骤二：设置启动参数

设置启动参数，[步骤四：启动ASM-VM-PROXY](#)将使用该启动参数实现以下功能。

 **说明** 设置启动参数前，请确保用户身份ACCESS_KEY_ID具备网格实例MESH_ID的查询权限。关于授权管理，请参见[授权概述](#)。

- 使用启动参数获取Kubernetes实例的Service CIDR，用于启动ASM-VM-PROXY。
- 使用启动参数获取服务网格实例istioD地址，用于启动ASM-VM-PROXY。
- 使用启动参数获取服务网格实例中间证书，并生成支持SPIFFE的虚拟机侧证书，用于启动ASM-VM-PROXY和数据平面通信。

分别在3个虚拟机的 `/opt/asm_vm_proxy/asm_vm_proxy.env` 路径下填写以下变量，并保存文件。

```
ACCESS_KEY_ID=
ACCESS_KEY_SECRET=
REGION=
MESH_ID=
NS=
SA=
POD_NAME=
```

| 参数 | 说明 |
|-------------------|-------------------------------|
| ACCESS_KEY_ID | 阿里云账号ID |
| ACCESS_KEY_SECRET | 阿里云账号密码 |
| REGION | 地域标识，请参考 地域标识 |
| MESH_ID | 服务网格实例ID |
| NS | 虚拟机要加入的命名空间（Namespace） |
| SA | 虚拟机要使用的服务账号（ServiceAccount） |
| POD_NAME | 虚拟机要使用的Pod的名称 |

步骤三：获取镜像路径

以下为ASM-VM-PROXY镜像的路径格式。

```
registry.{地域标识}.aliyuncs.com/acs/asm-vm-proxy:{版本标识}-aliyun
```

例如，地域名称为杭州、版本为v1.6.9的镜像路径为 `registry.cn-hangzhou.aliyuncs.com/acs/asm-vm-proxy:v1.6.9-aliyun`。

1. 确定地域标识。

ASM-VM-PROXY镜像开放了如下地域，请根据就近原则拉取镜像。

| 地域标识 | 地域名称 |
|----------------|-----------|
| cn-hangzhou | 华东1（杭州） |
| cn-shanghai | 华东2（上海） |
| cn-qingdao | 华北1（青岛） |
| cn-beijing | 华北2（北京） |
| cn-zhangjiakou | 华北3（张家口） |
| cn-huhehaote | 华北5（呼和浩特） |

| 地域标识 | 地域名称 |
|----------------|------------|
| cn-shenzhen | 华南1（深圳） |
| cn-chengdu | 西南1（成都） |
| cn-hongkong | 中国（香港） |
| ap-southeast-1 | 新加坡 |
| ap-southeast-2 | 澳大利亚（悉尼） |
| ap-southeast-3 | 马来西亚（吉隆坡） |
| ap-southeast-5 | 印度尼西亚（雅加达） |
| ap-northeast-1 | 日本（东京） |
| eu-central-1 | 德国（法兰克福） |
| eu-west-1 | 英国（伦敦） |
| us-west-1 | 美国（硅谷） |
| us-east-1 | 美国（弗吉尼亚） |
| ap-south-1 | 印度（孟买） |
| me-east-1 | 阿联酋（迪拜） |

2. 确定版本标识。

ASM-VM-PROXY后续的版本升级将会与ASM管控面一致。您可以在网格实例详情页获取版本信息。

- i. 登录ASM控制台。
- ii. 在左侧导航栏，选择服务网格 > 网格管理。
- iii. 在网格管理页面，找到待配置的实例，单击实例的名称或在操作列中单击管理。
- iv. 在网格实例详情页获取版本信息。



步骤四：启动ASM-VM-PROXY

执行以下命令，启动ASM-VM-PROXY。

```
docker run -d \  
--name=asm_vm_proxy \  
--network=host \  
--restart=always \  
--env-file /opt/asm_vm_proxy/asm_vm_proxy.env \  
--cap-add=NET_ADMIN \  
registry.{地域标识}.aliyuncs.com/acs/asm-vm-proxy:{版本标识}-aliyun
```

步骤五：检测ASM-VM-PROXY

1. 检测 `ISTIO_INBOUND` 信息。

```
iptables -t nat -L -v |grep -A5 "Chain ISTIO_INBOUND"
```

以下为预期结果。

```
Chain ISTIO_INBOUND (1 references)  
pkts bytes target prot opt in out source destination  
28 1428 RETURN tcp -- any any anywhere anywhere tcp dpt:ssh  
0 0 RETURN tcp -- any any anywhere anywhere tcp dpt:15020  
5 300 ISTIO_IN_REDIRECT tcp -- any any anywhere anywhere
```

2. 检测 `ISTIO_IN_REDIRECT` 信息。

```
iptables -t nat -L -v |grep -A3 "Chain ISTIO_IN_REDIRECT"
```

以下为预期结果。

```
Chain ISTIO_IN_REDIRECT (3 references)  
pkts bytes target prot opt in out source destination  
6 360 REDIRECT tcp -- any any anywhere anywhere redir ports 15006
```

3. 检测 `ISTIO_OUTPUT` 信息。

 说明 172.**.**/20是ASM-VM-PROXY中获取的Kubernetes实例的Service CIDR。

```
iptables -t nat -L -v |grep -A10 "Chain ISTIO_OUTPUT"
```

以下为预期结果。

```
Chain ISTIO_OUTPUT (1 references)
pkts bytes target prot opt in out source destination
0 0 RETURN all -- any lo 127.0.0.6 anywhere
0 0 ISTIO_IN_REDIRECT all -- any lo anywhere !localhost owner UID match 1337
7 420 RETURN all -- any lo anywhere anywhere !owner UID match 1337
0 0 RETURN all -- any any anywhere anywhere owner UID match 1337
0 0 ISTIO_IN_REDIRECT all -- any lo anywhere !localhost owner GID match 1337
0 0 RETURN all -- any lo anywhere anywhere !owner GID match 1337
0 0 RETURN all -- any any anywhere anywhere owner GID match 1337
0 0 RETURN all -- any any anywhere localhost
6 360 ISTIO_REDIRECT all -- any any anywhere 172.**.*./20
```

步骤六：诊断ASM-VM-PROXY

如果启动出现问题，您可以执行以下命令查看容器启动和Proxy运行的日志，定位启动问题。

1. 查看容器启动日志。

```
docker logs asm_vm_proxy
```

2. 查看Proxy日志。

```
docker exec -it asm_vm_proxy tail -n 100 /var/log/istio/istio.log
```

结果验证

- ASM-VM-PROXY启动成功后，会与ASM通信。登录ASM控制台，单击概览，在概览页面可以看到3个ECS节点。

| 服务网格 ASM | 名称 | 版本 | 路由配置 | 监听配置 | 集群配置 | 端点配置 |
|----------|------------|-------|--------|--------|--------|--------|
| 概览 | vm1.asm-vm | 1.6.9 | SYNCED | SYNCED | SYNCED | SYNCED |
| 服务网格 | vm2.asm-vm | 1.6.9 | SYNCED | SYNCED | SYNCED | SYNCED |
| 网络管理 | vm3.asm-vm | 1.6.9 | SYNCED | SYNCED | SYNCED | SYNCED |

- 登录ECS管理控制台，选择实例与镜像 > 实例。在实例列表页面单击目标实例操作列的管理，在实例详情页可以看到ASM-VM-PROXY为ECS实例标注了asm.cluster.id的标签，标签的值为ASM实例的ID。

