

(四) 基于Istio on Kubernetes 云原生应用的最佳实践

2018-11-09

准备Kubernetes集群

- 阿里云容器服务Kubernetes 1.10.4目前已经上线，可以通过容器服务管理控制台非常方便地快速创建 Kubernetes 集群。具体过程可以参考[创建Kubernetes集群](#)。
- 安装和设置kubectl客户端，请参考不同的操作系统，如果已经安装完成请忽略：

- macos

```
1 curl -LO https://kubectl.oss-cn-hangzhou.aliyuncs.com/macos/kubectl
2 chmod +x ./kubectl
3 sudo mv ./kubectl /usr/local/bin/kubectl
4 kubectl --help
```

- linux

```
1 curl -LO https://kubectl.oss-cn-hangzhou.aliyuncs.com/linux/kubectl
2 chmod +x ./kubectl
3 sudo mv ./kubectl /usr/local/bin/kubectl
4 kubectl --help
```

- windows

把<https://kubectl.oss-cn-hangzhou.aliyuncs.com/windows/kubectl.exe> 放到系统PATH路径下

```
1 kubectl --help
```

- 配置kubectl连接Kubernetes集群的配置，可参考文档[通过kubectl连接Kubernetes集群](#)
- 示例中用到的文件请参考：[文件](#)， Clone下载到本地，切换到目录kubec2018sh。
查看本文件：<https://github.com/osswangxining/yunqi2018-workshop-istio/tree/master/kubec2018sh>

部署Istio

打开容器服务控制台，在左侧导航栏中选中集群，右侧点击更多，在弹出的菜单中选中 部署Istio。

在打开的页面中可以看到Istio默认安装的命名空间、发布名称；通过勾选来确认是否安装相应的模块，注意勾选下**Kiali**；

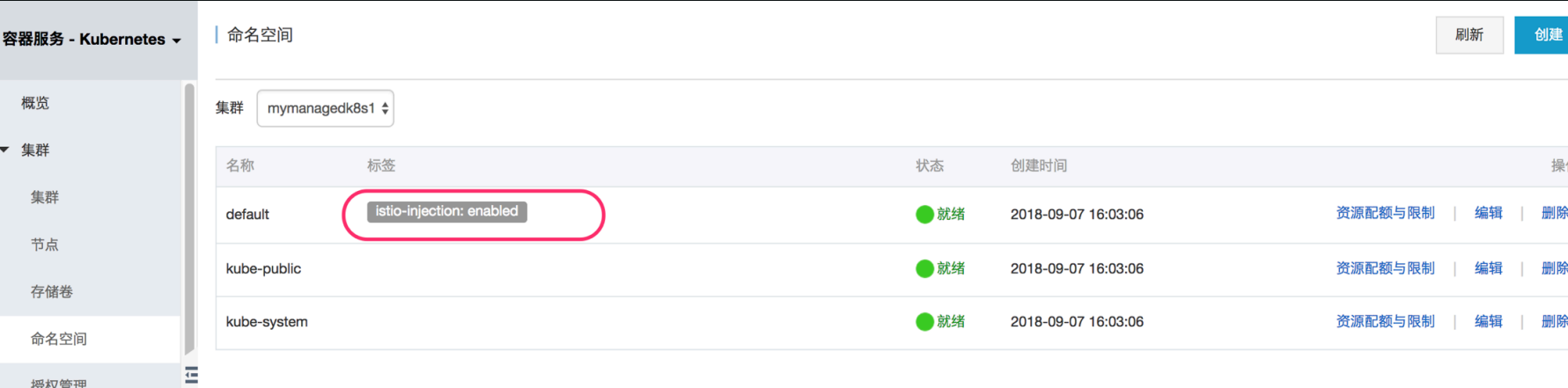
点击 部署Istio 按钮，几十秒钟之后即可完成部署。

自动 Sidecar 注入

查看namespace：



点击编辑，为 default 命名空间打上标签 istio-injection=enabled。



创建Ingress查看Kiali

路由-> 命名空间istio-system,点击创建。输入名称为kiali, 域名也为kiali, 选择服务kiali及端口20001。

后续步骤中会随时查看Kiali，来展示服务之间的调用关系。

部署应用

使用 kubectl 部署简单的服务

1

kubectl apply -f app.yaml

上面的命令会启动全部的3个服务，其中也包括了 addedvalues 服务的版本v1.

定义 Ingress gateway

1

kubectl apply -f gateway.yaml

确认所有的服务和 Pod 都已经正确的定义和启动

确认所有的服务已经正确的定义和启动

1

kubectl get services

确认所有的Pod 都已经正确的定义和启动

1

kubectl get pods

确认网关创建完成

1

kubectl get gateway

应用缺省目标规则

1

kubectl apply -f destination-v1.yaml

等待几秒钟，等待目标规则生效。这意味着上述3个微服务已经都部署在istio的环境中了。你可以使用以下命令来查看目标规则：

1

kubectl get destinationrules

查看Ingress Gateway的地址

点击左侧导航栏中的服务，在右侧上方选择对应的集群和命名空间，在列表中找到istio-ingressgateway的外部端点地址。

打开浏览器，访问<http://{GATEWAY-IP}/productpage>

部署v2 - 灰度发布

运行以下命令部署v2：

```
1 kubectl apply -f addedvalues-v2.yaml
```

部署DestinationRule：

```
1 kubectl apply -f destination-v2.yaml
```

部署VirtualService：

```
1 kubectl apply -f virtualservice-v1-v2.yaml
```

当前v1和v2的流量分别为50%.

然后，通过以下命令v2接管所有流量：

```
1 kubectl apply -f virtualservice-v2.yaml
```

完成灰度发布，切换到v2。

请求路由

接下来会把特定用户(登录名称以yunqi开头的)的请求发送给 v3 版本，其他用户则不受影响。
运行以下命令部署v2：

```
1 kubectl apply -f addedvalues-v3.yaml
```

部署DestinationRule：

```
1 kubectl apply -f destination-v3.yaml
```

部署VirtualService：

```
1 kubectl apply -f virtualservice-user-v2-v3.yaml
```

打开浏览器，访问<http://{GATEWAY-IP}/productpage>

不论刷新多少次页面，如果没有登录或者登录名不是以yunqi开头的，始终得到如下的显示内容，也就是上述提到的 第2 个版本的 addedvalues微服务。

资产管理之某XX理财产品尊享款

产品简介: 本理财产品属于保本固定收益型，风险级别偏低，适合投资风险承受能力较低，偏好安全性高的投资者购买。

产品明细

产品类别:
资产类
封闭期(天):
184
预期收益率:
4.85%
首次购买起点:
1,000,000.00
起息日:
2018-08-13
到期日:
2018-10-15

增值服务

本理财产品属于保本固定收益型，风险级别偏低，适合投资风险承受能力较低，偏好安全性高的投资者购买。在流动性方面，本产品为1月期理财产品，同时，银行允许客户随时赎回产品，所以流动性极佳。

— 风险星级
★★★★☆

收益率方面，本产品的收益规则为实际理财天数小于等于7天,则收益率为1.25%; 实际理财天数在7天到25天之内,收益率为1.40%; 实际理财天数大于25天或产品到期，收益率为1.78%。客户收益计算公式为:理财本金×收益率÷360×实际理财天数。同活期存款利率0.36%相比，本产品的收益率具备投资价值。

— 投资星级
★★★★☆

资产管理之某XX理财产品尊享款

产品简介: 本理财产品属于保本固定收益型，风险级别偏低，适合投资风险承受能力较低，偏好安全性高的投资者购买。

产品明细

产品类别:
资产类
封闭期(天):
184
预期收益率:
4.85%
首次购买起点:
1,000,000.00
起息日:
2018-08-13
到期日:
2018-10-15

增值服务

本理财产品属于保本固定收益型，风险级别偏低，适合投资风险承受能力较低，偏好安全性高的投资者购买。在流动性方面，本产品为1月期理财产品，同时，银行允许客户随时赎回产品，所以流动性极佳。

— 风险星级
★★★★☆

收益率方面，本产品的收益规则为实际理财天数小于等于7天,则收益率为1.25%; 实际理财天数在7天到25天之内,收益率为1.40%; 实际理财天数大于25天或产品到期，收益率为1.78%。客户收益计算公式为:理财本金×收益率÷360×实际理财天数。同活期存款利率0.36%相比，本产品的收益率具备投资价值。

— 投资星级
★★★★☆

当使用以yunqi开头的用户名登录时，就会看到如下页面内容， 也就是上述提到的 第3 个版本的addedvalues微服务。

资产管理之某XX理财产品尊享款

产品简介: 本理财产品属于保本固定收益型，风险级别偏低，适合投资风险承受能力较低，偏好安全性高的投资者购买。

产品明细

产品类别:
资产类
封闭期(天):
184
预期收益率:
4.85%
首次购买起点:
1,000,000.00
起息日:
2018-08-13
到期日:
2018-10-15

增值服务

本理财产品属于保本固定收益型，风险级别偏低，适合投资风险承受能力较低，偏好安全性高的投资者购买。在流动性方面，本产品为1月期理财产品，同时，银行允许客户随时赎回产品，所以流动性极佳。

— 风险星级
★★★★☆

收益率方面，本产品的收益规则为实际理财天数小于等于7天,则收益率为1.25%; 实际理财天数在7天到25天之内,收益率为1.40%; 实际理财天数大于25天或产品到期，收益率为1.78%。客户收益计算公式为:理财本金×收益率÷360×实际理财天数。同活期存款利率0.36%相比，本产品的收益率具备投资价值。

— 投资星级
★★★★☆

根据自身需求定制专属卡片

```
1 | kubectl delete -f .
```

About

This theme was developed by [Jonathan Klughertz](#). The source code is available on [Github](#). Create Websites. Make Magic.

Recent Posts

[Kubernetes动手实践沙龙](#)

[（一）Kubernetes弹性扩缩容实践](#)

[（二）基于Kubernetes的三种发布策略 -](#)

[（三）通过Serverless Kuberne](#)

