



链滴

# istio 安装和 bookinfo 示例

作者: [lggl](#)

原文链接: <https://ld246.com/article/1591939925273>

来源网站: 链滴

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



## istio

istio从1.5版本开始回归单体架构，下面记录1.6.1版本安装测试。kubernetes版本为1.18

### 1.安装

安装请参考[官网](#)，安装包下载可以github上获取，如果能联网，也可以使用下面命令获取

```
$ curl -L https://istio.io/downloadIstio | sh -
```

切换到 Istio 包所在目录下。例如：Istio 包名为 **istio-1.6.1**，则：

```
$ cd istio-1.6.1
```

然后将 **istioctl** 放到系统路径下

```
$ mv ./bin/istioctl /usr/local/bin
```

接着就可以安装了

```
istioctl manifest apply --set profile=demo
```

验证是否安装成功

```
$ kubectl get svc -n istio-system
```

NAME	TYPE	CLUSTER-IP	EXTERNAL-IP	PORT(S)	AGE
grafana	ClusterIP	172.21.211.123	<none>	3000/TCP	2m
istio-citadel	ClusterIP	172.21.177.222	<none>	8060/TCP,15014/TCP	2m

```

istio-egressgateway ClusterIP 172.21.113.24 <none> 80/TCP,443/TCP,15443/TCP
2m
istio-galley ClusterIP 172.21.132.247 <none> 443/TCP,15014/TCP,9901/TCP
2m
istio-ingressgateway LoadBalancer 172.21.144.254 52.116.22.242 15020:31831/TCP,80:3
380/TCP,443:31390/TCP,31400:31400/TCP,15029:30318/TCP,15030:32645/TCP,15031:31933/T
P,15032:31188/TCP,15443:30838/TCP 2m
istio-pilot ClusterIP 172.21.105.205 <none> 15010/TCP,15011/TCP,8080/TCP
,15014/TCP 2m
istio-policy ClusterIP 172.21.14.236 <none> 9091/TCP,15004/TCP,15014/TCP
2m
istio-sidecar-injector ClusterIP 172.21.155.47 <none> 443/TCP,15014/TCP
2m
istio-telemetry ClusterIP 172.21.196.79 <none> 9091/TCP,15004/TCP,15014/TCP
CP,42422/TCP 2m
jaeger-agent ClusterIP None <none> 5775/UDP,6831/UDP,6832/UDP
2m
jaeger-collector ClusterIP 172.21.135.51 <none> 14267/TCP,14268/TCP
2m
jaeger-query ClusterIP 172.21.26.187 <none> 16686/TCP
2m
kiali ClusterIP 172.21.155.201 <none> 20001/TCP
2m
prometheus ClusterIP 172.21.63.159 <none> 9090/TCP
2m
tracing ClusterIP 172.21.2.245 <none> 80/TCP
2m
zipkin ClusterIP 172.21.182.245 <none> 9411/TCP

```

如果集群运行在一个不支持外部负载均衡器的环境中 (例如: minikube) , **istio-ingressgateway** 的 **XTERNAL-IP**将显示为 **<pending>** 状态。请使用服务的 **NodePort** 或 端口转发来访问网关。

```
$ kubectl edit svc istio-ingressgateway -n istio-system
```

```
...
spec
...
type: NodePort
```

请确保关联的 Kubernetes pod 已经部署, 并且 **STATUS** 为 **Running**:

```
$ kubectl get pods -n istio-system
NAME                                READY STATUS RESTARTS AGE
grafana-f8467cc6-rbjlg              1/1   Running 0       1m
istio-citadel-78df5b548f-g5cpw      1/1   Running 0       1m
istio-egressgateway-78569df5c4-zwtb5 1/1   Running 0       1m
istio-galley-74d5f764fc-q7nrk      1/1   Running 0       1m
istio-ingressgateway-7ddcfd665c-dmtqz 1/1   Running 0       1m
istio-pilot-f479bbf5c-qwr28        1/1   Running 0       1m
istio-policy-6fccc5c868-xhblv      1/1   Running 2       1m
istio-sidecar-injector-78499d85b8-x44m6 1/1   Running 0       1m
istio-telemetry-78b96c6cb6-ldm9q    1/1   Running 2       1m
istio-tracing-69b5f778b7-s2zvw     1/1   Running 0       1m
kiali-99f7467dc-6rvwp              1/1   Running 0       1m
prometheus-67cdb66cbb-9w2hm        1/1   Running 0       1m
```

## 2.Profile配置

不同Profile, 安装的组件不同, 上面是为了测试, 指定了 **demo**配置, 具体不同版本请查看[官网](#), 下贴一下:

下表中标记为 **X** 的组件就是包含在配置文件里的内容:

	default	demo	minimal	sds
核心组件				
<code>istio-citadel</code>	X	X		X
<code>istio-egressgateway</code>		X		
<code>istio-galley</code>	X	X		X
<code>istio-ingressgateway</code>	X	X		X
<code>istio-nodeagent</code>				X
<code>istio-pilot</code>	X	X	X	X
<code>istio-policy</code>	X	X		X
<code>istio-sidecar-injector</code>	X	X		X
<code>istio-telemetry</code>	X	X		X
插件				
<code>grafana</code>		X		
<code>istio-tracing</code>		X		
<code>kiali</code>		X		
<code>prometheus</code>	X	X		X

为了进一步自定义 Istio 和安装插件，您可以在安装 Istio 时所使用的 `istioctl manifest` 命令中添加一个或多个 `--set <key>=<value>` 选项。 [安装选项](#) 中列出了完整的当前所支持的安装键值对集合。

## 2.bookinfo示例

官网链接为<https://istio.io/latest/zh/docs/examples/bookinfo/>

1.进入 Istio 安装目录。

2.Istio 默认自动注入 Sidecar. 请为 default 命名空间打上标签 istio-injection=enabled:

```
$ kubectl label namespace default istio-injection=enabled
```

3.使用 kubectl 部署应用:

```
$ kubectl apply -f samples/bookinfo/platform/kube/bookinfo.yaml
```

4.确认所有的服务和 Pod 都已经正确的定义和启动:

```
$ kubectl get services
NAME                CLUSTER-IP   EXTERNAL-IP   PORT(S)          AGE
details             10.0.0.31    <none>        9080/TCP         6m
kubernetes          10.0.0.1     <none>        443/TCP          7d
productpage        10.0.0.120   <none>        9080/TCP         6m
ratings            10.0.0.15    <none>        9080/TCP         6m
reviews            10.0.0.170   <none>        9080/TCP         6m
```

和

```
$ kubectl get pods
NAME                                READY   STATUS    RESTARTS   AGE
details-v1-1520924117-48z17        2/2     Running  0          6m
productpage-v1-560495357-jk1lz     2/2     Running  0          6m
ratings-v1-734492171-rnr5l        2/2     Running  0          6m
reviews-v1-874083890-f0qf0        2/2     Running  0          6m
reviews-v2-1343845940-b34q5       2/2     Running  0          6m
reviews-v3-1813607990-8ch52       2/2     Running  0          6m
```

5.要确认 Bookinfo 应用是否正在运行, 请在某个 Pod 中用 curl 命令对应用发送请求, 例如 ratings:

```
$ kubectl exec -it $(kubectl get pod -l app=ratings -o jsonpath='{.items[0].metadata.name}') -
ratings -- curl productpage:9080/productpage | grep -o "<title>.*</title>"
<title>Simple Bookstore App</title>
```

6.为应用程序定义 Ingress 网关:

```
$ kubectl apply -f samples/bookinfo/networking/bookinfo-gateway.yaml
```

7.确认网关创建完成:

```
$ kubectl get gateway
NAME                AGE
bookinfo-gateway   32s
```

8.通过istio网关访问应用

浏览器访问[http://\\$GATEWAY\\_URL/productpage](http://$GATEWAY_URL/productpage), 可以多刷新几次, 会有不同的效果。

\$GATEWAY\_URL=节点ip+nodeport端口

nodeport为istio-ingressgateway的svc的http2指定的端口, 通过

```
$ kubectl get svc istio-ingressgateway -n istio-system -oyaml
```

...

```
- name: http2  
  nodePort: 31624  
  port: 80  
  protocol: TCP
```

...

## 8.验证规则

```
$ kubectl apply -f samples/bookinfo/networking/destination-rule-all.yaml
```

## 9.查看规则

```
kubectl get destinationrules -o yaml
```