IBM

Open by design™

IBM开源技术微讲堂

Istio系列

第六讲

Istio mixer—policy, telemetry and extending

http://ibm.biz/opentech-ma

讲师介绍

张文涛 zwtzhang@cn.ibm.com

专注于IBM Cloud上的service监控与运维。

主要兴趣在kubernetes, istio, big data and ML.

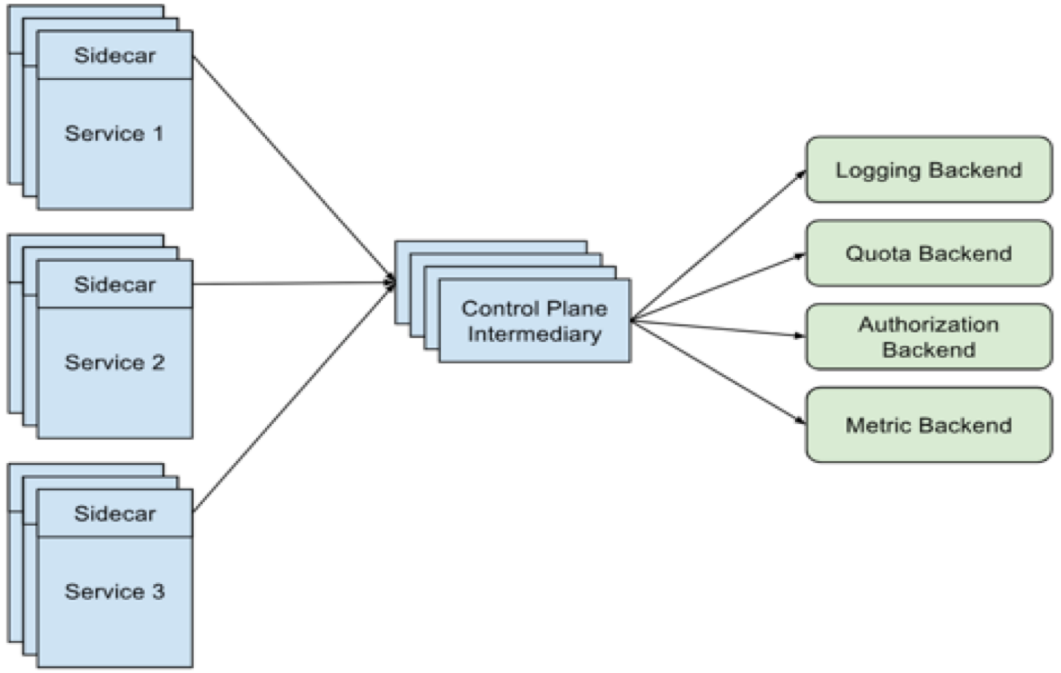
Github: https://github.com/wentao-zh

议程

- Origin
- Concepts
- Policy
- Telemetry
- Adapter

Origin

- Google API & Service Management System



Google's API & Service Management System

Reference

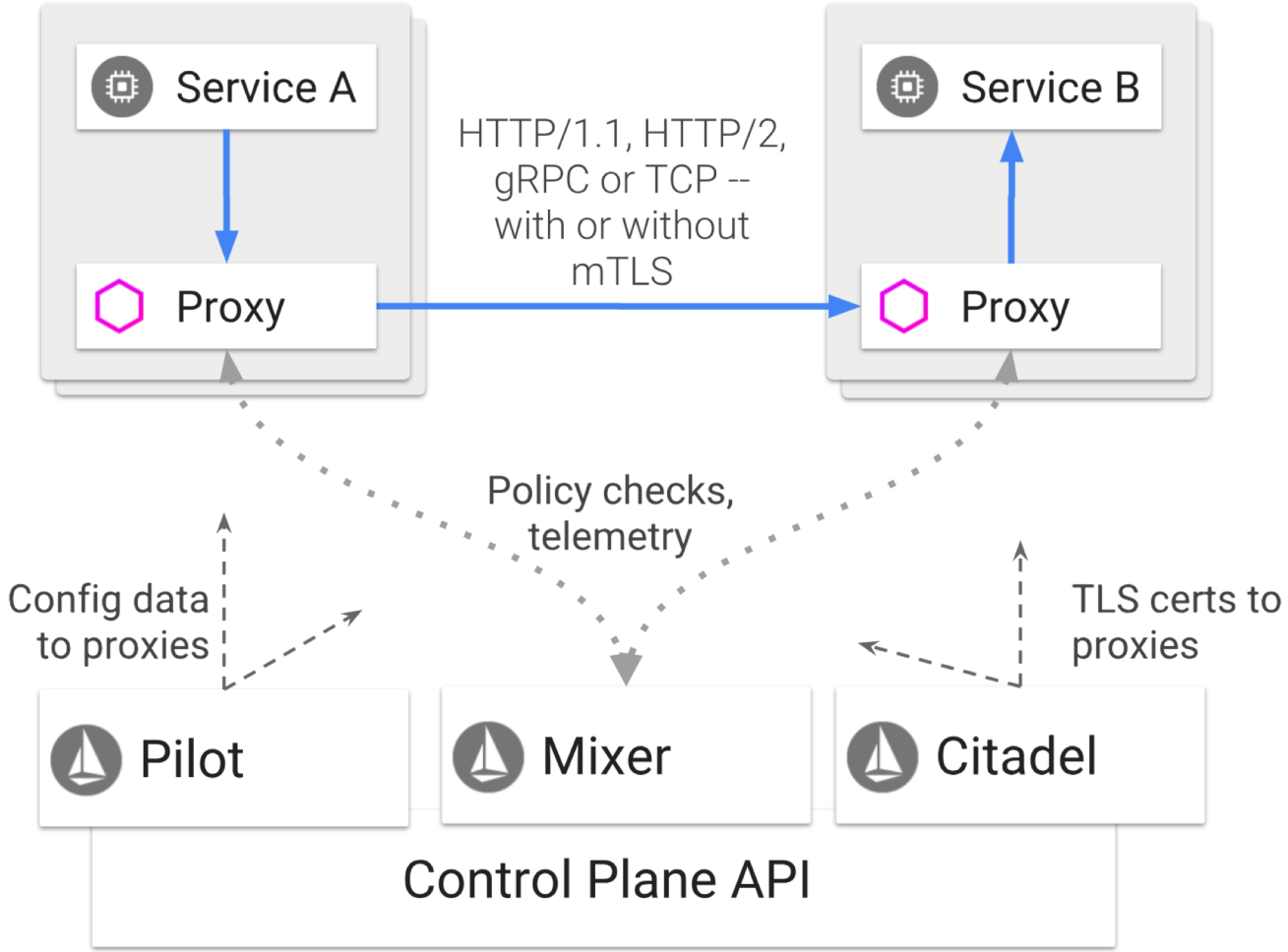
Mixer and the SPOF Myth

By MARTIN TAILLEFER

https://istio.io/blog/2017/mixer-spof-myth

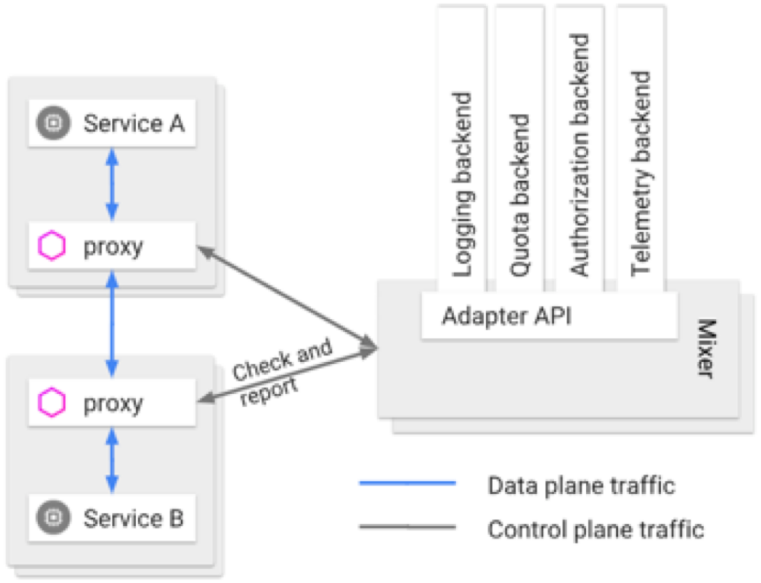
Concepts

- Quick view of Istio architecture



Concepts

- **Infrastructure backends**
  - provide support functionality used to build services
  - access control systems, telemetry capturing systems, quota enforcement systems, billing systems, and so forth



Mixer Topology

Concepts

- **Attribute**
  - An attribute is a small bit of data that describes a single property of a specific **service request** or the **environment for the request**.

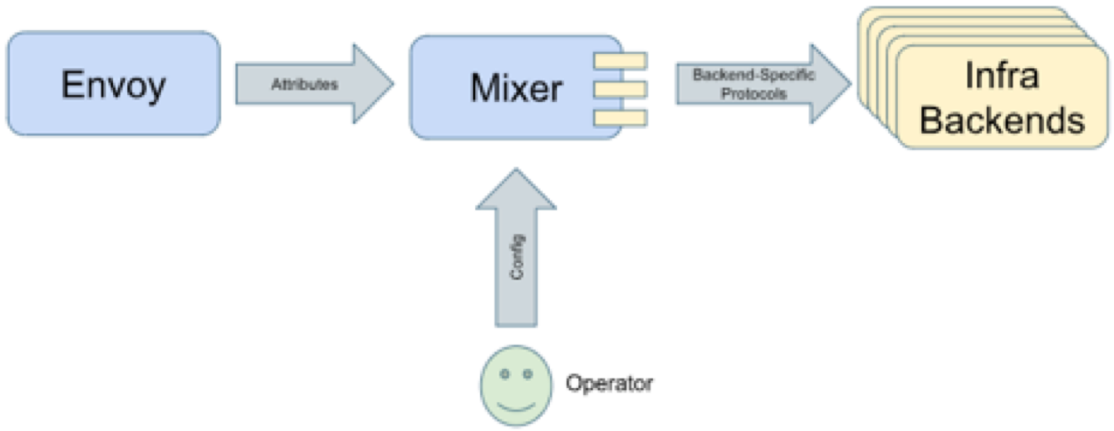
```
request.path: xyz/abc
request.size: 234
request.time: 12:34:56.789 04/17/2017
source.ip: 192.168.0.1
destination.service: example
```

  - Attribute Vocabulary: https://istio.io/docs/reference/config/policy-and-telemetry/attribute-vocabulary/

Name	Type	Description	Kubernetes Example
source.uid	string	Platform-specific unique identifier for the source workload instance.	kubernetes://redis-master-2353460263-1ecey.my-namespace
source.ip	ip_address	Source workload instance IP address.	10.0.0.117
source.labels	map[string, string]	A map of key-value pairs attached to the source instance.	version => v1
source.name	string	Source workload instance name.	redis-master-2353460263-1ecey
source.namespace	string	Source workload instance namespace.	my-namespace
source.principal	string	Authority under which the source workload instance is running.	service-account-foo
source.owner	string	Reference to the workload controlling the source workload instance.	kubernetes://apis/extensions/v1beta1/namespaces/istio-system/deployments/istio-policy

Concepts

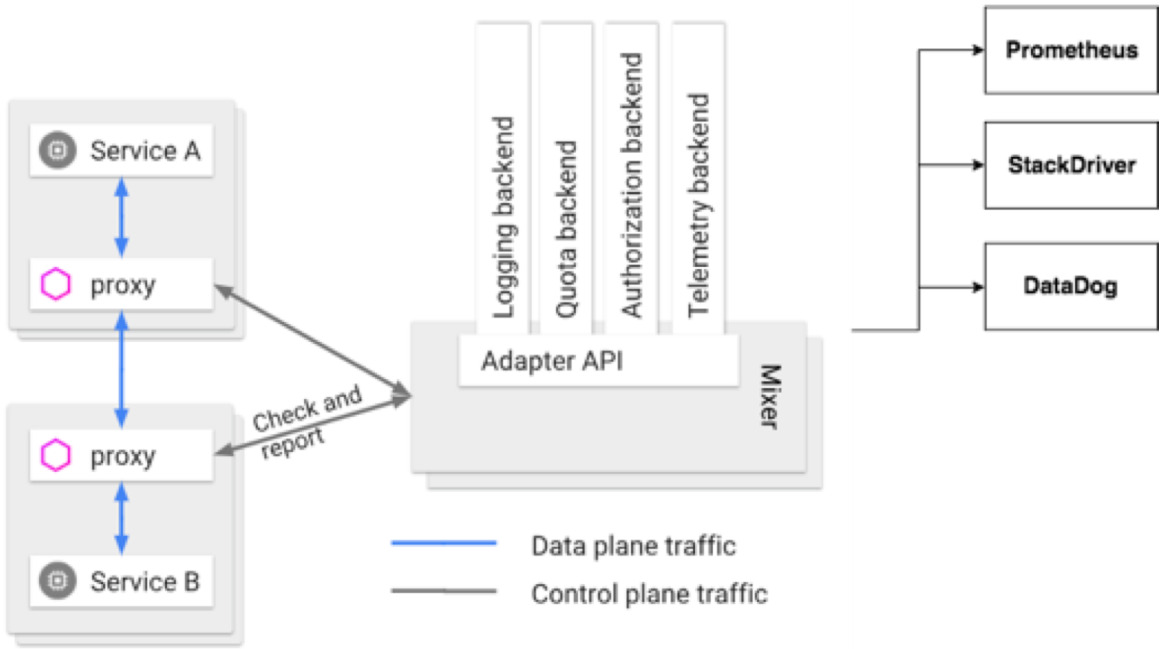
- **Attribute Machine**
  - The **Envoy sidecar**
    - invokes Mixer
    - giving Mixer a set of attributes that describe the request and the environment around the request



Attribute Machine

Concepts

- **Adapter**
  - abstract away the details of different policy and telemetry backend systems
  - encapsulate the logic necessary to interface Mixer with a specific infrastructure backend
  - build in mixer/ out of mixer process



Concepts

- **Template**
  - define the schema for specifying request mapping from attributes to adapter inputs.
  - A given adapter may support any number of templates
  - canonical templates
    - check: apikey, authorization, checknothing, quota
    - report: metric, lolentry, reportnothing, tracespan

Concepts

- **Configuration Model**
  - **handler**: determine the set of adapters that are being used and how they operate
  - **instance**: describe how to map request attributes into adapter inputs. Instances represent a chunk of data that one or more adapters will operate on
  - **rule**: describe when a particular adapter is called and which instances it is given

IBM

Open by design™

Policy

- Precondition check
  - authentication: eg. OPA
  - authorization: eg. OPA
  - quota: eg. Memory quota/Redis Quota
  - rate limit:

Policy

- sample: denier

```yaml
apiVersion: "config.istio.io/v1alpha2"
kind: denier
metadata:
  name: handler
  namespace: istio-system
spec:
  status:
    code: 7
    message: Not allowed
---
apiVersion: "config.istio.io/v1alpha2"
kind: checknothing
metadata:
  name: denyrequest
  namespace: istio-system
spec:

---
apiVersion: "config.istio.io/v1alpha2"
kind: rule
metadata:
  name: denyingress
  namespace: istio-system
spec:
  match: source.labels["istio"] == "ingressgateway" && request.headers["x-user"] == "john"
  actions:
  - handler: handler.denier.istio-system
    instances: [ denyrequest.checknothing.istio-system ]
```

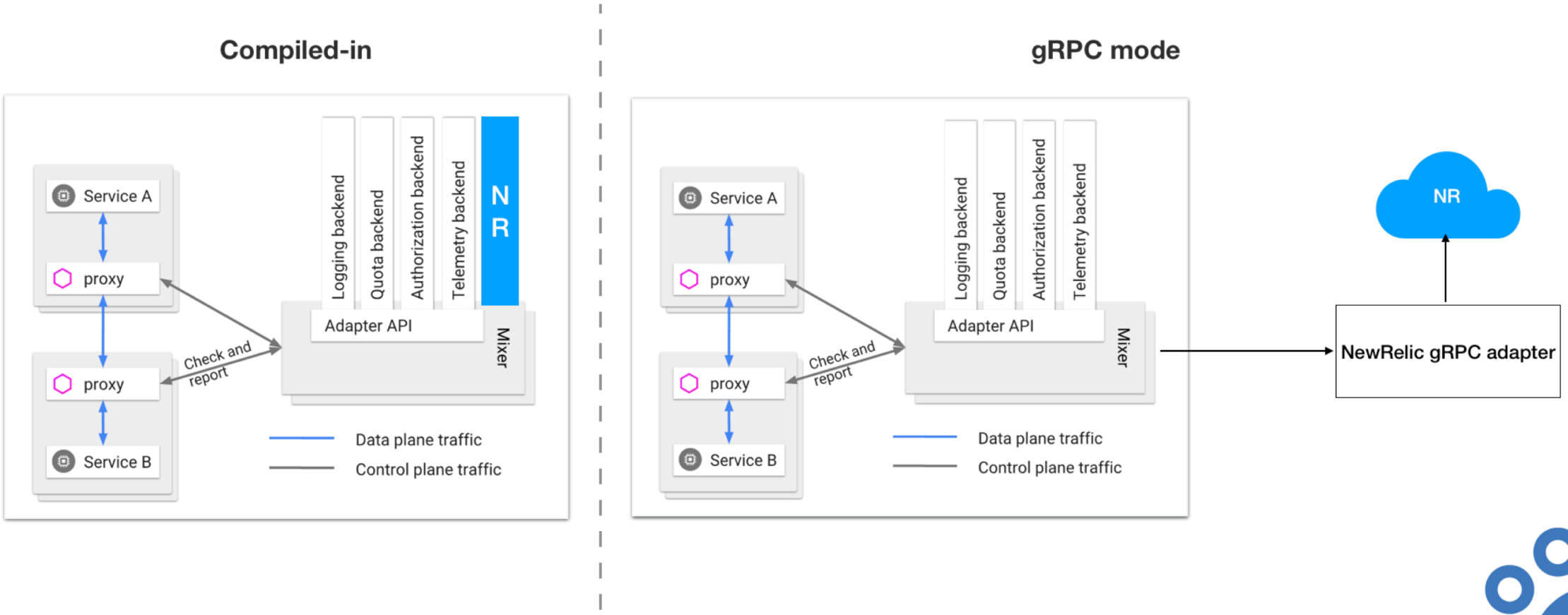
Telemetry

- Telemetry reporting
  - monitoring metrics: eg. StatsD
  - logging: eg. StackDriver/Fluentd

```yaml
apiVersion: "config.istio.io/v1alpha2"
kind: metric
metadata:
  name: doublerequestcount
  namespace: istio-system
spec:
  value: "2" # count each request twice
  dimensions:
    reporter: conditional((context.reporter.kind | "inbound") == "outbound", "client", "server")
    source: source.workload.name | "unknown"
    destination: destination.workload.name | "unknown"
    message: '"twice the fun!"'
  monitored_resource_type: '"UNSPECIFIED"'
---
# Configuration for a Prometheus handler
apiVersion: "config.istio.io/v1alpha2"
kind: prometheus
metadata:
  name: doublehandler
  namespace: istio-system
spec:
  metrics:
  - name: double_request_count # Prometheus metric name
    instance_name: doublerequestcount.metric.istio-system # Mixer instance name (fully-qualified)
    kind: COUNTER
    label_names:
    - reporter
    - source
    - destination
    - message
```

```yaml
---
# Rule to send metric instances to a Prometheus handler
apiVersion: "config.istio.io/v1alpha2"
kind: rule
metadata:
  name: doubleprom
  namespace: istio-system
spec:
  actions:
  - handler: doublehandler.prometheus
    instances:
    - doublerequestcount.metric
```

Adapter


Compiled-in
Service A
proxy
proxy
Service B
Check and report
Logging backend
Quota backend
Authorization backend
Telemetry backend
N R
Adapter API
Mixer
Data plane traffic
Control plane traffic
gRPC mode
Service A
proxy
proxy
Service B
Check and report
Logging backend
Quota backend
Authorization backend
Telemetry backend
Adapter API
Mixer
Data plane traffic
Control plane traffic
NR
NewRelic gRPC adapter

Adapter

- Mixer Out Of Process Adapter Dev Guide
  - https://github.com/istio/istio/wiki/Mixer-Out-Of-Process-Adapter-Dev-Guide
- Out of process Attribute Generating Adapter Developer Walkthrough
  - https://github.com/istio/istio/wiki/Out-of-process-Attribute-Generating-Adapter-Developer-Walkthrough
- https://github.com/IBM/newrelic-istio-adapter

```yaml
apiVersion: "config.istio.io/v1alpha2"
kind: handler
metadata:
 name: newrelic
 namespace: istio-system
spec:
 adapter: newrelic
 connection:
   address: "nristio.default.svc.cluster.local:8888"
```

```yaml
apiVersion: apps/v1
kind: Deployment
metadata:
  name: nristio-adapter
spec:
  selector:
    matchLabels:
      app: nristio
  replicas: 1
  template:
    metadata:
      labels:
        app: nristio
    spec:
      containers:
      - name: nristio
        image: registry.ng.bluemix.net/zwtzhang/nristio-adapter:0.1.3
        env:
        - name: NEW_RELIC_APIKEY
          valueFrom:
            secretKeyRef:
              name: newrelic-secret
              key: NEW_RELIC_APIKEY
        - name: NEW_RELIC_ACCOUNT
          valueFrom:
            secretKeyRef:
              name: newrelic-secret
              key: NEW_RELIC_ACCOUNT
        command: ["/nristioadapter"]
        args:
        - --port=41165
        - --maxworkers=1024
      imagePullSecrets:
      - name: nradaptertoken
```

IBM

Open by design™

IBM开源技术微讲堂

Istio系列

第六讲完

http://ibm.biz/opentech-ma