1 ICA

Generated on: 2026-01-13 15:04:48 **Version:** 1.0

1.1 Table of Contents

1.2 Overview

The **Istio Certified Associate (ICA)** exam demonstrates a candidate's knowledge of Istio service mesh concepts and practical skills.

1.3 Exam Overview

Detail	Information
Exam Format	Multiple Choice
Number of Questions	60
Duration	90 minutes
Passing Score	75%
Certification Validity	3 years
Cost	$250 USD
Retake Policy	1 free retake

1.4 Exam Domains & Weights

Domain	Weight
Istio Installation, Upgrade & Configuration	7%
Traffic Management	40%
Resiliency and Fault Injection	10%
Securing Workloads	20%
Advanced Scenarios	13%
Observability	10%

1.5 Key Topics

1.5.1 Traffic Management

- Virtual Services and Destination Rules
- Gateways (Ingress/Egress)
- Traffic shifting and mirroring
- Request routing

1.5.2 Security

- mTLS configuration
- Authorization policies
- Peer and request authentication
- Certificate management

1.5.3 Observability

- Metrics with Prometheus
- Distributed tracing
- Access logging
- Kiali dashboard

1.6 Study Resources

- Istio Documentation
- ICA Curriculum
- Istio by Example

1.7 Navigation

- Next: Sample Questions →

1.8 Istio Architecture

Comprehensive guide to Istio service mesh architecture for ICA certification.

1.9 Overview

Istio is an open-source service mesh that provides:

- **Traffic Management** - Control traffic flow between services
- **Security** - Secure service-to-service communication
- **Observability** - Monitor and trace requests
- **Policy Enforcement** - Apply policies consistently

1.10 Architecture Components

1.10.1 Control Plane (istiod)

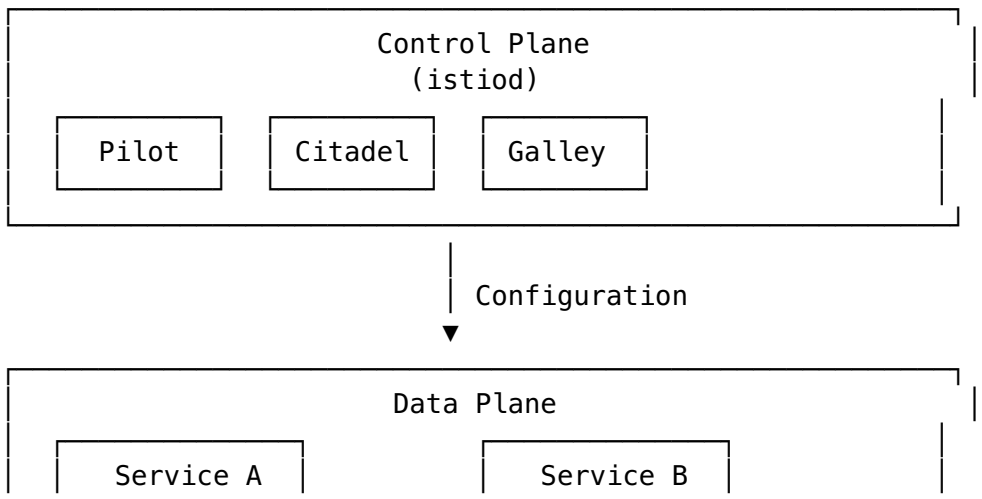
The control plane is consolidated into a single binary called **istiod**:

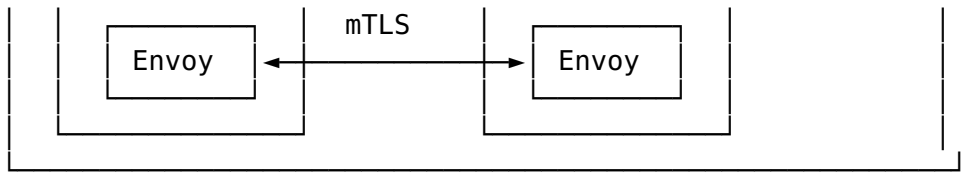
- **Pilot** - Service discovery, traffic management configuration
- **Citadel** - Certificate management, identity
- **Galley** - Configuration validation and distribution

1.10.2 Data Plane

- **Envoy Proxies** - Sidecar proxies deployed with each service
- Intercept all network traffic
- Apply policies and collect telemetry

1.11 Traffic Flow





1.12 Installation

1.12.1 Using istioctl

```
# Download Istio
curl -L https://istio.io/downloadIstio | sh -
cd istio-*
export PATH=$PWD/bin:$PATH

# Install with default profile
istioctl install --set profile=default -y

# Install with demo profile (includes more features)
istioctl install --set profile=demo -y

# Verify installation
istioctl verify-install
```

1.12.2 Installation Profiles

Profile	Description
default	Production deployment
demo	Full features for learning
minimal	Minimal control plane
remote	Remote cluster in multicluster
empty	Nothing installed

1.13 Sidecar Injection

1.13.1 Automatic Injection

```
# Label namespace for automatic injection
kubectl label namespace default istio-injection=enabled

# Verify label
kubectl get namespace -L istio-injection
```

1.13.2 Manual Injection

```
# Inject sidecar manually
istioctl kube-inject -f deployment.yaml | kubectl apply -f -
```

1.14 Core Resources

1.14.1 VirtualService

Controls traffic routing:

```
apiVersion: networking.istio.io/v1beta1
kind: VirtualService
metadata:
  name: reviews-route
spec:
  hosts:
  - reviews
  http:
  - match:
    - headers:
        end-user:
          exact: jason
    route:
    - destination:
        host: reviews
        subset: v2
  - route:
    - destination:
        host: reviews
        subset: v1
```

1.14.2 DestinationRule

Defines policies for traffic after routing:

```
apiVersion: networking.istio.io/v1beta1
kind: DestinationRule
metadata:
  name: reviews-destination
spec:
  host: reviews
  trafficPolicy:
    connectionPool:
      tcp:
        maxConnections: 100
  subsets:
  - name: v1
    labels:
```

```
      version: v1
  - name: v2
    labels:
      version: v2
```

1.14.3 Gateway

Manages inbound/outbound traffic:

```
apiVersion: networking.istio.io/v1beta1
kind: Gateway
metadata:
  name: my-gateway
spec:
  selector:
    istio: ingressgateway
  servers:
  - port:
      number: 80
      name: http
      protocol: HTTP
    hosts:
    - "*.example.com"
```

1.14.4 ServiceEntry

Adds external services to mesh:

```
apiVersion: networking.istio.io/v1beta1
kind: ServiceEntry
metadata:
  name: external-api
spec:
  hosts:
  - api.external.com
  ports:
  - number: 443
    name: https
    protocol: HTTPS
  resolution: DNS
  location: MESH_EXTERNAL
```

1.15 Security

1.15.1 PeerAuthentication

```
apiVersion: security.istio.io/v1beta1
kind: PeerAuthentication
metadata:
```

```yaml
  name: default
  namespace: istio-system
spec:
  mtls:
    mode: STRICT
```

1.15.2 AuthorizationPolicy

```yaml
apiVersion: security.istio.io/v1beta1
kind: AuthorizationPolicy
metadata:
  name: allow-read
  namespace: default
spec:
  selector:
    matchLabels:
      app: myapp
  action: ALLOW
  rules:
  - from:
    - source:
        principals: ["cluster.local/ns/default/sa/frontend"]
    to:
    - operation:
        methods: ["GET"]
```

1.16 Observability

1.16.1 Kiali Dashboard

```bash
# Install Kiali
kubectl apply -f samples/addons/kiali.yaml

# Access dashboard
istioctl dashboard kiali
```

1.16.2 Jaeger Tracing

```bash
# Install Jaeger
kubectl apply -f samples/addons/jaeger.yaml

# Access dashboard
istioctl dashboard jaeger
```

1.16.3 Prometheus & Grafana

```bash
# Install addons
kubectl apply -f samples/addons/prometheus.yaml
```

```
kubectl apply -f samples/addons/grafana.yaml

# Access dashboards
istioctl dashboard prometheus
istioctl dashboard grafana
```

1.17 Useful Commands

```
# Check proxy status
istioctl proxy-status

# Analyze configuration
istioctl analyze

# Debug proxy
istioctl proxy-config clusters <pod-name>
istioctl proxy-config routes <pod-name>
istioctl proxy-config listeners <pod-name>

# View mesh configuration
istioctl manifest generate --set profile=demo
```

← Back to ICA Overview

1.18 Sample Practice Questions

1.19 Practice Resources

- Istio Documentation
- Istio by Example
- Killercoda Istio Scenarios

1.20 Traffic Management (40%)

1.20.1 Question 1

Create a VirtualService that routes 80% of traffic to v1 and 20% to v2 of the reviews service.

Show Solution

```
apiVersion: networking.istio.io/v1beta1
kind: VirtualService
```

```
metadata:
  name: reviews
spec:
  hosts:
  - reviews
  http:
  - route:
    - destination:
        host: reviews
        subset: v1
      weight: 80
    - destination:
        host: reviews
        subset: v2
      weight: 20
```

1.20.2 Question 2

Create a DestinationRule that defines subsets v1 and v2 based on version labels.

Show Solution

```
apiVersion: networking.istio.io/v1beta1
kind: DestinationRule
metadata:
  name: reviews
spec:
  host: reviews
  subsets:
  - name: v1
    labels:
      version: v1
  - name: v2
    labels:
      version: v2
```

1.20.3 Question 3

Create an Istio Gateway for incoming HTTPS traffic on port 443.

Show Solution

```
apiVersion: networking.istio.io/v1beta1
kind: Gateway
metadata:
  name: my-gateway
spec:
  selector:
    istio: ingressgateway
  servers:
  - port:
      number: 443
```

```
      name: https
      protocol: HTTPS
    tls:
      mode: SIMPLE
      credentialName: my-tls-secret
    hosts:
    - "*.example.com"
```

1.20.4 Question 4

Configure request timeout of 10 seconds for the ratings service.

Show Solution

```
apiVersion: networking.istio.io/v1beta1
kind: VirtualService
metadata:
  name: ratings
spec:
  hosts:
  - ratings
  http:
  - timeout: 10s
    route:
    - destination:
        host: ratings
```

1.21 Securing Workloads (20%)

1.21.1 Question 5

Enable strict mTLS for the entire mesh.

Show Solution

```
apiVersion: security.istio.io/v1beta1
kind: PeerAuthentication
metadata:
  name: default
  namespace: istio-system
spec:
  mtls:
    mode: STRICT
```

1.21.2 Question 6

Create an AuthorizationPolicy that only allows GET requests from the frontend service.

Show Solution

```yaml
apiVersion: security.istio.io/v1beta1
kind: AuthorizationPolicy
metadata:
  name: allow-frontend-get
spec:
  selector:
    matchLabels:
      app: backend
  action: ALLOW
  rules:
  - from:
    - source:
        principals: ["cluster.local/ns/default/sa/frontend"]
    to:
    - operation:
        methods: ["GET"]
```

1.21.3 Question 7

Configure JWT authentication for a service.

Show Solution

```yaml
apiVersion: security.istio.io/v1beta1
kind: RequestAuthentication
metadata:
  name: jwt-auth
spec:
  selector:
    matchLabels:
      app: httpbin
  jwtRules:
  - issuer: "https://example.com"
    jwksUri: "https://example.com/.well-known/jwks.json"
```

1.22 Resiliency and Fault Injection (10%)

1.22.1 Question 8

Inject a 5-second delay for 50% of requests to the ratings service.

Show Solution

```yaml
apiVersion: networking.istio.io/v1beta1
kind: VirtualService
metadata:
  name: ratings
spec:
```

```
  hosts:
  - ratings
  http:
  - fault:
      delay:
        percentage:
          value: 50
        fixedDelay: 5s
    route:
    - destination:
        host: ratings
```

1.22.2 Question 9

Configure circuit breaker with max 100 connections and 1000 pending requests.

Show Solution

```
apiVersion: networking.istio.io/v1beta1
kind: DestinationRule
metadata:
  name: reviews
spec:
  host: reviews
  trafficPolicy:
    connectionPool:
      tcp:
        maxConnections: 100
      http:
        h2UpgradePolicy: UPGRADE
        http1MaxPendingRequests: 1000
    outlierDetection:
      consecutive5xxErrors: 5
      interval: 30s
      baseEjectionTime: 30s
```

1.22.3 Question 10

Inject HTTP 503 errors for 10% of requests.

Show Solution

```
apiVersion: networking.istio.io/v1beta1
kind: VirtualService
metadata:
  name: ratings
spec:
  hosts:
  - ratings
  http:
  - fault:
      abort:
```

```
        percentage:
          value: 10
        httpStatus: 503
    route:
    - destination:
        host: ratings
```

1.23 Observability (10%)

1.23.1 Question 11

What are the three pillars of observability in Istio?

Show Solution

1. **Metrics** - Collected by Prometheus, visualized in Grafana
2. **Distributed Tracing** - Using Jaeger or Zipkin
3. **Access Logs** - Envoy access logs for debugging

1.23.2 Question 12

How do you enable access logging in Istio?

Show Solution

```
apiVersion: telemetry.istio.io/v1alpha1
kind: Telemetry
metadata:
  name: mesh-default
  namespace: istio-system
spec:
  accessLogging:
  - providers:
    - name: envoy
```

Or via IstioOperator:

```
spec:
  meshConfig:
    accessLogFile: /dev/stdout
```

1.24 Installation & Configuration (7%)

1.24.1 Question 13

Install Istio with the demo profile using istioctl.

Show Solution

```
istioctl install --set profile=demo -y
```

1.24.2 Question 14

Enable automatic sidecar injection for a namespace.

Show Solution

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

1.25 Exam Tips

1. **Know VirtualService and DestinationRule** - These are heavily tested
2. **Understand mTLS modes** - STRICT, PERMISSIVE, DISABLE
3. **Practice traffic management** - Routing, splitting, mirroring
4. **Know AuthorizationPolicy** - ALLOW, DENY, CUSTOM actions
5. **Understand fault injection** - Delays and aborts

← Back to ICA Overview
