

# What is Istio

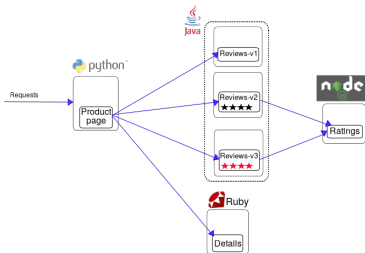
# What is Istio

- Service Mesh, Why do we need a Service Mesh?

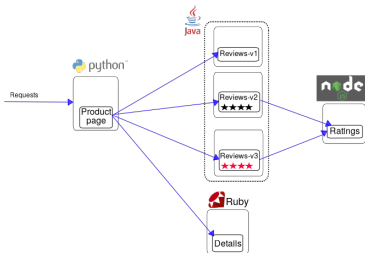
# What is Istio

- Service Mesh, Why do we need a Service Mesh?
- Let's take an example

# bookinfo

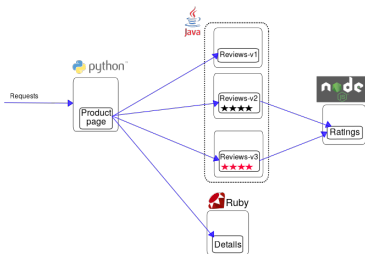


- Secure with mTLS

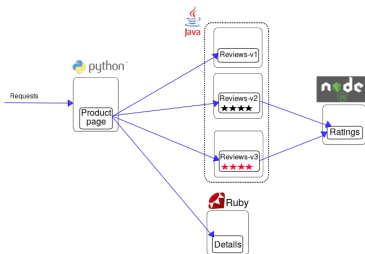


# bookinfo

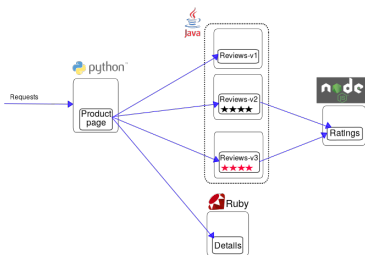
- Secure with mTLS
- Connect



# bookinfo

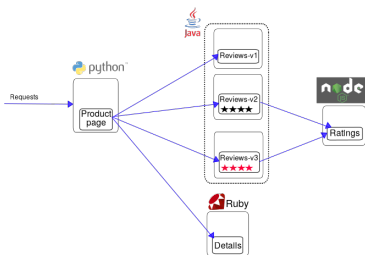


- Secure with mTLS
- Connect
- Service discovery, traffic management, test/canary versioning, incremental rollouts

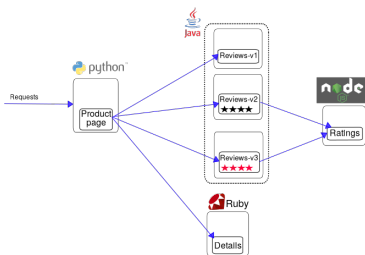


- Secure with mTLS
- Connect
- Service discovery, traffic management, test/canary versioning, incremental rollouts
- Retry policies, Circuit breaker

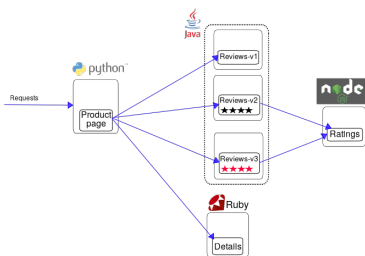




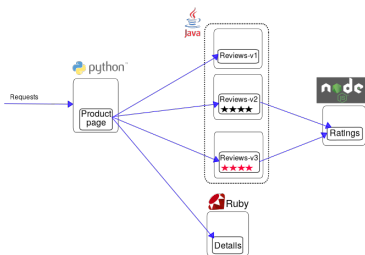
- Secure with mTLS
- Connect
- Service discovery, traffic management, test/canary versioning, incremental rollouts
- Retry policies, Circuit breaker
- Observe



- Secure with mTLS
- Connect
- Service discovery, traffic management, test/canary versioning, incremental rollouts
- Retry policies, Circuit breaker
- Observe
- See where the bottlenecks are, how traffic flows

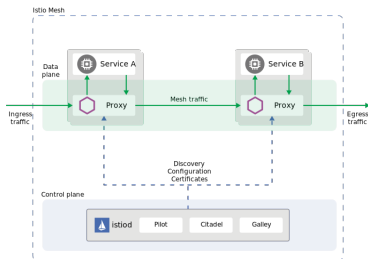


- Secure with mTLS
- Connect
- Service discovery, traffic management, test/canary versioning, incremental rollouts
- Retry policies, Circuit breaker
- Observe
- See where the bottlenecks are, how traffic flows
- Control

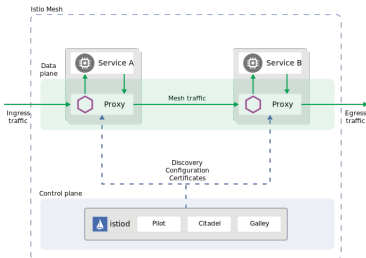


- Secure with mTLS
- Connect
- Service discovery, traffic management, test/canary versioning, incremental rollouts
- Retry policies, Circuit breaker
- Observe
- See where the bottlenecks are, how traffic flows
- Control
- Like who's allowed to talk to who, Rouge containers cannot talk to a service

# proxies

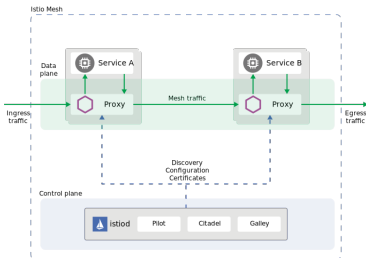


# proxies



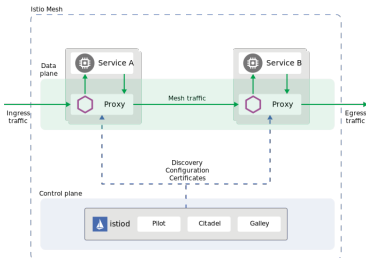
- Injected envoy proxies to each containers in the system

# proxies



- Injected envoy proxies to each containers in the system
- It runs next to your containers in the same kubernetes pod

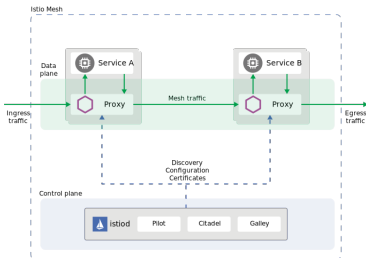
# proxies



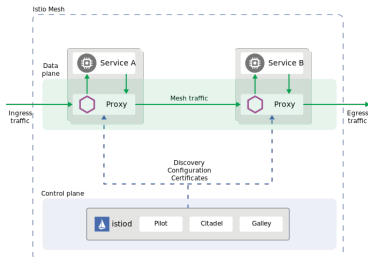
- Injected envoy proxies to each containers in the system
- It runs next to your containers in the same kubernetes pod
- Intercepts all requests and responses, apply any policies defined



# Istio tools

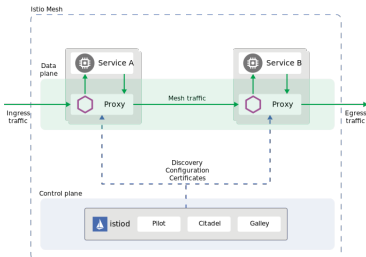


# Istio tools



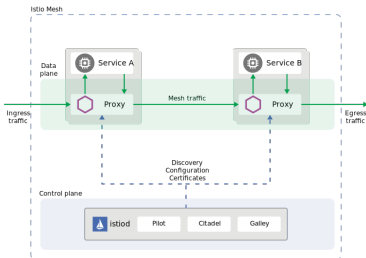
- Galley

# Istio tools



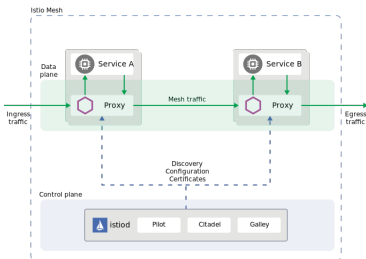
- Galley
- Pilot

# Istio tools



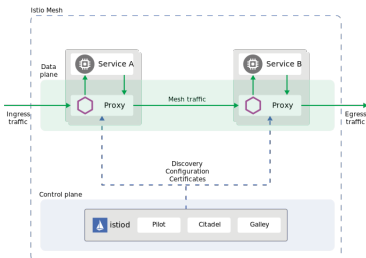
- Galley
- Pilot
- Policy

# Istio tools



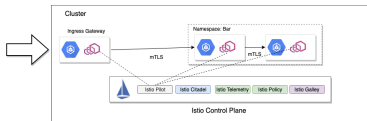
- Galley
- Pilot
- Policy
- Telemetry

# Istio tools



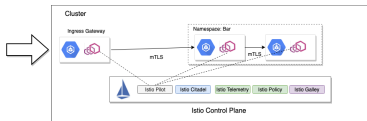
- Galley
- Pilot
- Policy
- Telemetry
- Citadel

# Istio Components



# Istio Components

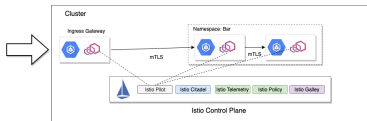
- Gateway



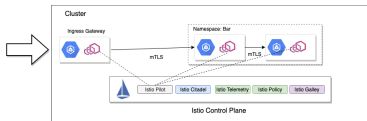


# Istio Components

- Gateway
- Virtual Service

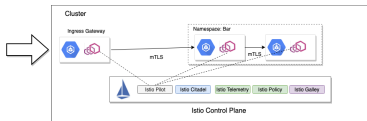


# Istio Components



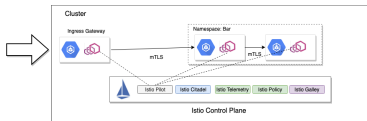
- Gateway
- Virtual Service
- VS can be from Gateway to a service
- VS can be from service to another service

# Istio Components



- Gateway
- Virtual Service
- VS can be from Gateway to a service
- VS can be from service to another service
- VS is where u define the percentile traffic split for different versions

# Istio Components



- Gateway
- Virtual Service
- VS can be from Gateway to a service
- VS can be from service to another service
- VS is where u define the percentile traffic split for different versions
- DestinationRules
- For applying rules like Circuit breaker/retry/etc

- Let's do a demo