**Bookinfo Application**

 This example deploys a sample application composed of four separate microservices used to demonstrate various Istio features.

If you installed Istio using the [Getting Started](https://istio.io/latest/docs/setup/getting-started/) instructions, you already have Bookinfo installed and you can skip these steps.

The application displays information about a book, similar to a single catalog entry of an online book store. Displayed on the page is a description of the book, book details (ISBN, number of pages, and so on), and a few book reviews.

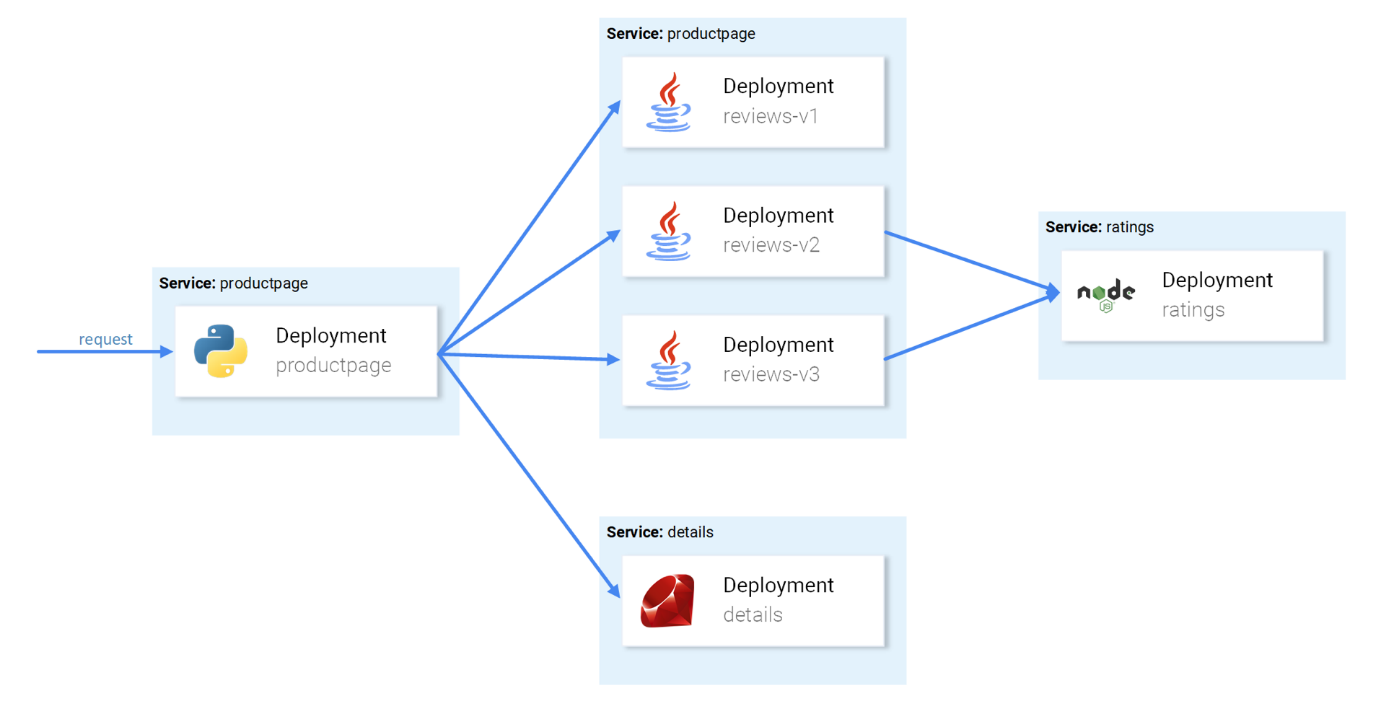
The Bookinfo application is broken into four separate microservices:

* productpage. The productpage microservice calls the details and reviews microservices to populate the page.
* details. The details microservice contains book information.
* reviews. The reviews microservice contains book reviews. It also calls the ratings microservice.
* ratings. The ratings microservice contains book ranking information that accompanies a book review.

There are 3 versions of the reviews microservice:

* Version v1 doesn’t call the ratings service.
* Version v2 calls the ratings service, and displays each rating as 1 to 5 black stars.
* Version v3 calls the ratings service, and displays each rating as 1 to 5 red stars.

The end-to-end architecture of the application is shown below.

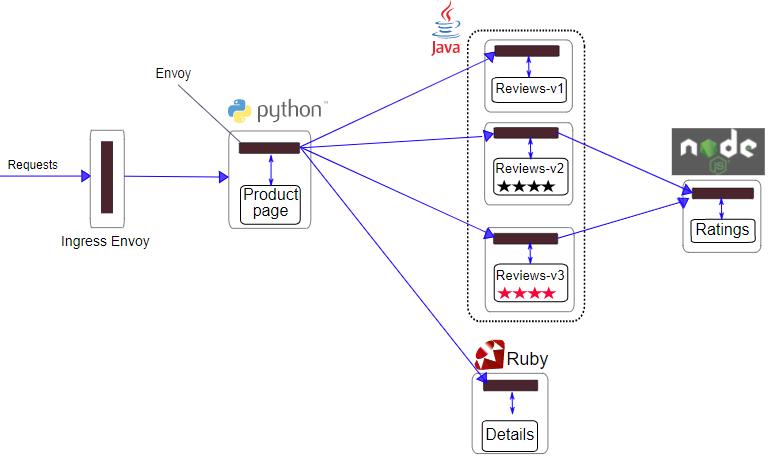


Bookinfo Application without Istio

This application is polyglot, i.e., the microservices are written in different languages. It’s worth noting that these services have no dependencies on Istio, but make an interesting service mesh example, particularly because of the multitude of services, languages and versions for the reviews service.

**Deploying the application**

To run the sample with Istio requires no changes to the application itself. Instead, you simply need to configure and run the services in an Istio-enabled environment, with Envoy sidecars injected along side each service. The resulting deployment will look like this:



Bookinfo Application

All of the microservices will be packaged with an Envoy sidecar that intercepts incoming and outgoing calls for the services, providing the hooks needed to externally control, via the Istio control plane, routing, telemetry collection, and policy enforcement for the application as a whole.

**Start the application services**

1. Change directory to the root of the Istio installation.
2. The default Istio installation uses [automatic sidecar injection](https://istio.io/latest/docs/setup/additional-setup/sidecar-injection/#automatic-sidecar-injection). Label the namespace that will host the application with istio-injection=enabled:

$ kubectl label namespace default istio-injection=enabled