

Anthos Service Mesh uses sidecar proxies to enhance network security, reliability, and visibility. With Anthos Service Mesh, these functions are abstracted away from the application's primary container and implemented in a common out-of-process proxy delivered as a separate container in the same Pod. To take full advantage of [Anthos Service Mesh's features](/service-mesh/docs/overview) (/service-mesh/docs/overview), each service in your application needs to have an [Envoy](https://www.envoyproxy.io) (https://www.envoyproxy.io) sidecar proxy running in its Pod. The Envoy proxy intercepts all inbound and outbound HTTP traffic to the service and communicates with Anthos Service Mesh.

You can inject an Envoy proxy manually by updating your Pods' Kubernetes configuration, or you can use the webhooks-based automatic sidecar injection. By default, sidecar auto-injection is disabled for all namespaces. To enable auto-injection:

where ***NAMESPACE*** is the name of the [namespace](https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/) (https://kubernetes.io/docs/concepts/overview/working-with-objects/namespaces/) for your application's services or `default` if you didn't explicitly create a namespace.

Because sidecars are injected when Pods are created, you must restart any running Pods for the change to take effect.

**Tip:** Unless you have a load balancer or router setup for [blue-green deployments](https://martinfowler.com/bliki/BlueGreenDeployment.html) (https://martinfowler.com/bliki/BlueGreenDeployment.html), make sure you test restarting Pods in a staging environment

that your services can handle any potential traffic interruption.

To restart Pods:

How you restart Pods depends on if they were created as part of a [Deployment](#) (/kubernetes-engine/docs/concepts/deployment).

1. If you used a Deployment, restart the Deployment, which restarts all Pods with sidecars:

If you didn't use a Deployment, delete the Pods, and they are automatically recreated with sidecars:

2. Check that all the Pods in the namespace have sidecars injected:

In the following example output from the previous command, notice that the `READY` column indicates there are two containers for each of your workloads: the primary container and the container for the sidecar proxy.

Learn more about:

- [Installing sidecars](https://istio.io/docs/setup/kubernetes/sidecar-injection/) (https://istio.io/docs/setup/kubernetes/sidecar-injection/)
- [Deploying workloads](#) (/kubernetes-engine/docs/how-to/deploying-workloads-overview)

