

Anthos Service Mesh, which is powered by [Istio](https://istio.io/docs/concepts/what-is-istio/) (<https://istio.io/docs/concepts/what-is-istio/>), is a framework for connecting, monitoring, and securing services running on Google Kubernetes Engine (GKE) and GKE On-Prem. It lets you create a network of deployed services with load balancing, service-to-service authentication, monitoring, and more, without requiring any changes in service code. You add Anthos Service Mesh support to services by deploying a sidecar proxy to each of your application's Pods. The sidecar proxy intercepts all network communication between microservices, and is configured and managed using Anthos Service Mesh's control plane functionality.

This guide explains how to install Anthos Service Mesh with the [supported core Istio features](/service-mesh/docs/supported-features) (</service-mesh/docs/supported-features>) enabled on either a new or existing GKE cluster and deploy a demo multi-service application.

This guide doesn't explain how to enable the [Anthos Service Mesh managed components](/service-mesh/docs/overview#managed_components) (/service-mesh/docs/overview#managed_components). The managed components are currently in beta. If you would like to join the Anthos Service Mesh beta, fill out the [Contact sales form](https://cloud.google.com/contact/?form=anthos) (<https://cloud.google.com/contact/?form=anthos>), and someone from the Anthos team will contact you. You must register your Google Cloud project and Google Cloud account to participate in the Anthos Service Mesh beta.

If you haven't already done so, create a [Google Cloud project](/resource-manager/docs/creating-managing-projects) (</resource-manager/docs/creating-managing-projects>).

If you plan to install Anthos Service Mesh on a new GKE cluster:

- The minimum [machine type](/compute/docs/machine-types#machine_types) (/compute/docs/machine-types#machine_types) required by Anthos Service Mesh is `n1-standard-4`, which has 4 vCPUs.
- Make sure you install a [supported cluster version](/service-mesh/docs/supported-features#supported_environments) (/service-mesh/docs/supported-features#supported_environments).
- After the setup, review [Requirements for Pods and Services](https://istio.io/docs/setup/kubernetes/additional-setup/requirements/) (<https://istio.io/docs/setup/kubernetes/additional-setup/requirements/>) before you deploy workloads.

If you plan to install Anthos Service Mesh on an existing GKE cluster:

- Make sure your cluster satisfies the requirements specified in [Requirements for Pods and Services](https://istio.io/docs/setup/kubernetes/additional-setup/requirements/) (<https://istio.io/docs/setup/kubernetes/additional-setup/requirements/>).
- Make sure your cluster master version is listed in [Supported environments](/service-mesh/docs/supported-features#supported_environments) (/service-mesh/docs/supported-features#supported_environments). To check your cluster version:

Output like the following is displayed:

If you need to upgrade your cluster to a supported version, see [Manually upgrading a cluster or node pool](/kubernetes-engine/docs/how-to/upgrading-a-cluster) (</kubernetes-engine/docs/how-to/upgrading-a-cluster>).


- The minimum machine type required by Anthos Service Mesh is `n1-standard-4`, which has 4 vCPUs. If the machine type for your cluster doesn't have at least 4 vCPUs, change the machine type as described in [Migrating workloads to different machine types](/kubernetes-engine/docs/tutorials/migrating-node-pool) (</kubernetes-engine/docs/tutorials/migrating-node-pool>)

[Multiple mesh deployments](https://istio.io/docs/ops/deployment/deployment-models/#mesh-models) (<https://istio.io/docs/ops/deployment/deployment-models/#mesh-models>) in a single Google Cloud project aren't supported.

You can follow the installation guides using [Cloud Shell](/shell/docs) (</shell/docs>), an in-browser command line interface to your Google Cloud resources, or locally on your own computer running.

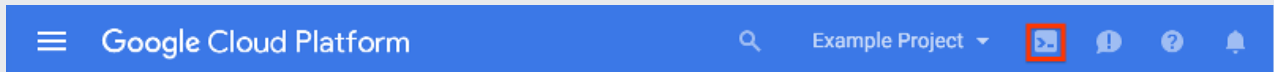
Cloud Shell provisions a [g1-small Compute Engine virtual machine \(VM\)](/compute/docs/machine-types) (</compute/docs/machine-types>) running a Debian-based Linux operating system. The advantages to using Cloud Shell are:

- Cloud Shell includes the `gcloud`, `kubectl` and `helm` command-line tools that you need.

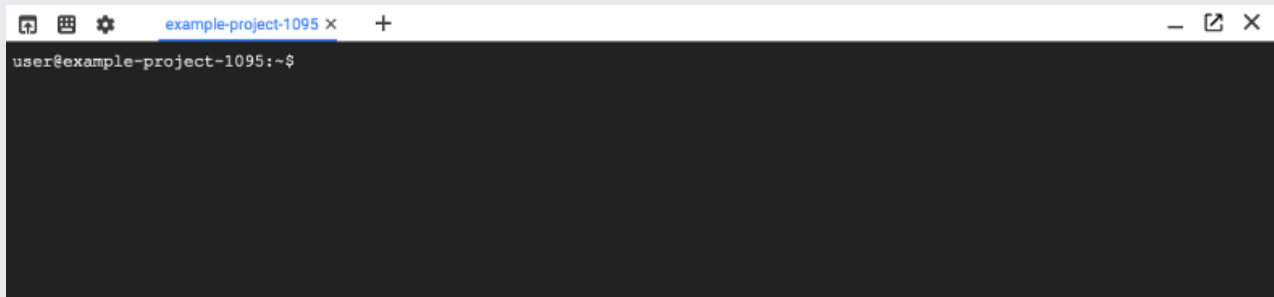
- Your Cloud Shell \$HOME directory has at least 5GB persistent storage space.
- You have your choice of [text editors](/shell/docs/features#tools) (/shell/docs/features#tools):
 - [Code editor](/shell/docs/features#code_editor) (/shell/docs/features#code_editor), which you access by clicking  at the top of the Cloud Shell window.
 - Emacs, Vim, or Nano, which you access from the command line in Cloud Shell.

To use Cloud Shell:

1. Go to the [Cloud Console](https://console.cloud.google.com/) (https://console.cloud.google.com/).
2. Select your Cloud project.
3. Click the **Activate Cloud Shell** button at the top of the Cloud Console window.



A Cloud Shell session opens inside a new frame at the bottom of the Cloud Console and displays a command-line prompt.



On your local machine, install the following tools if you don't already have them:

1. Install and initialize the [Cloud SDK](/sdk/docs/quickstarts) (/sdk/docs/quickstarts) (the `gcloud` command-line tool).

If you already have the Cloud SDK installed, make sure to update the components:

2. Install `kubectl`:

1. Authenticate with the Cloud SDK:
2. Get your [GCP project ID](#) ([/resource-manager/docs/creating-managing-projects#identifying_projects](#)) and create an environment variable for it:

★ **Note:** Many of the commands in this guide depend on this environment variable. If you disconnect from your shell environment and reconnect later, you will need to set up the variable again.

3. Set the default project ID for the `gcloud` command-line tool:
4. Set the required Cloud Identity and Access Management (Cloud IAM) roles. You need the following roles on the Cloud project:

Role name	Role ID	Description
Project Editor	roles/editor	Permissions for actions that modify state, such as changing existing resources.
Kubernetes Engine Admin	roles/container.admin	Provides access to full management of Container Clusters and their Kubernetes API objects.

If you are a *Project Owner*, you don't need to add those roles because the *Project Owner* role has all the necessary permissions. For details on setting the roles, see [Granting, changing, and revoking access to resources](#) ([/iam/docs/granting-changing-revoking-access](#)).

5. Enable the following APIs:

Enabling the APIs can take a minute or more to complete.

You can install Anthos Service Mesh on either a new or existing GKE cluster. If you are using an existing cluster, it must meet the requirements explained in the [Before you begin](#) (#before_you_begin) section.

This section explains the basics of creating a GKE cluster with the options that are required for Anthos Service Mesh.

To set up a new cluster:

1. Select a [zone](#) (/compute/docs/regions-zones/#available) and a [machine type](#) (/compute/docs/machine-types) for the new cluster. The minimum machine type required by Anthos Service Mesh is [n1-standard-4](#) (/compute/docs/machine-types).

- a. To get a list of the available GCP zones:

- b. To get a list of machine types:

2. Set up the following environment variables:

3. Set the default zone for the `gcloud` command-line tool:

★ **Tip:** To make setting up your shell environment easier in the future, you can copy and past the `export` statements for each environment variable to a simple shell script that you `source` when you start a new shell. You can also add the `gcloud` commands that set the default zone and project to the script. Or you can use [gcloud config configurations](/sdk/gcloud/reference/config/configurations) (/sdk/gcloud/reference/config/configurations) to create and activate a named `gcloud` configuration.

4. Create the cluster with the options required by Anthos Service Mesh. The following command creates a cluster containing 4 nodes of machine type `n1-standard-4`, which has 4 vCPUs. This is the minimum machine type and number of nodes required for Anthos Service Mesh. You can specify another machine type as long as it has at least 4 vCPUs, and you can increase the number of nodes as needed for your system requirements.

Before installing Anthos Service Mesh, you need to:

- Set required credentials and permissions.
- Download and extract the Anthos Service Mesh installation file.

1. Get [authentication credentials](#)

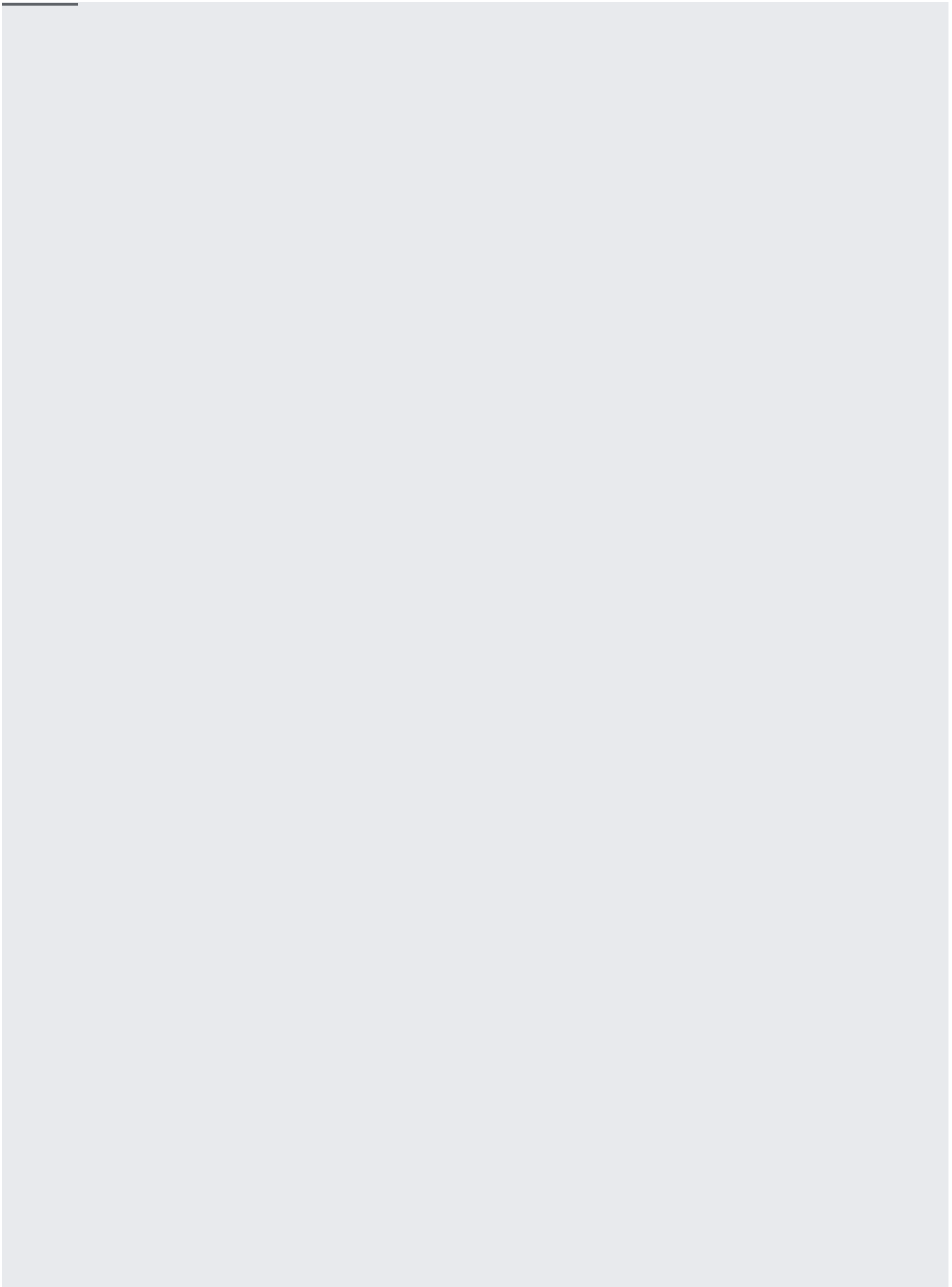
(/kubernetes-engine/docs/how-to/cluster-access-for-kubectI#default_cluster_kubectI) to interact with the

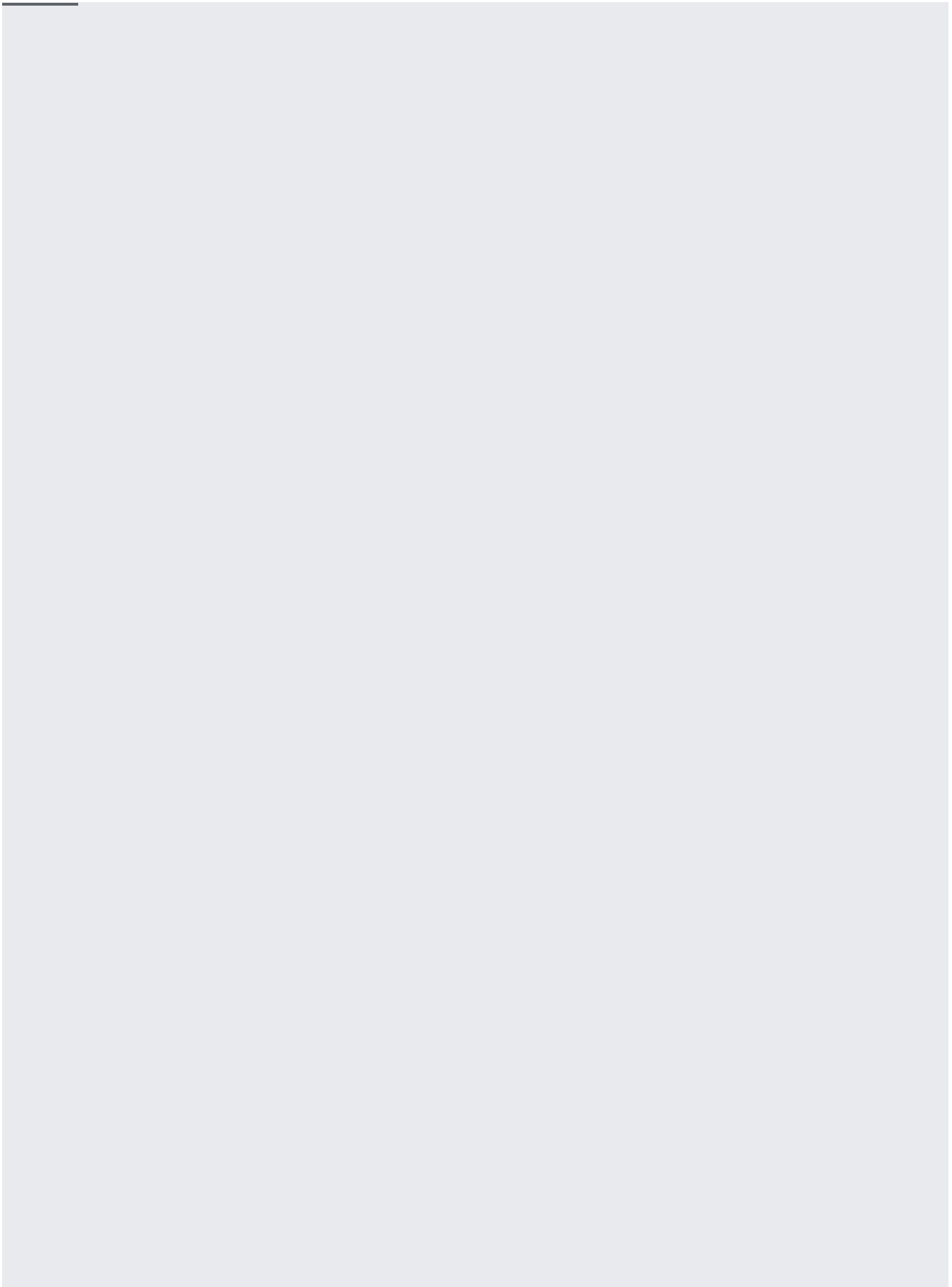
cluster:

This command configures `kubectl` to use the specified cluster.

2. Grant cluster admin permissions to the current user. You need these permissions to create the necessary role based access control (RBAC) (</kubernetes-engine/docs/role-based-access-control>) rules for Anthos Service Mesh:

If you see the "`cluster-admin-binding`" `already exists` error, you can safely ignore it and continue with the existing cluster-admin-binding.





To install Anthos Service Mesh using the ASM configuration profile:

The ASM profile contains the configuration options suitable for running your services in a production environment. For a full list of features enabled by the ASM profile see [Anthos Service Mesh supported features](/service-mesh/docs/supported-features) (/service-mesh/docs/supported-features).

The ASM configuration profile:

- Sets your services to run in **PERMISSIVE** mode.

! Caution: In **PERMISSIVE** mode, your services can accept both plaintext traffic and mutual TLS traffic at the same time and send plaintext intra-mesh traffic by default. To secure your service mesh, migrate your services to mutual TLS **STRICT** mode, as described in [Mutual TLS Migration](https://istio.io/docs/tasks/security/authentication/mtls-migration/) (<https://istio.io/docs/tasks/security/authentication/mtls-migration/>).

- Enables the node agent secret discovery service (SDS).

! Caution: If you don't enforce a Pod security policy, we recommend that you create and enable Pod security policies to secure SDS. For more information, see [Securing SDS with pod security policies](https://istio.io/docs/tasks/security/citadel-config/auth-sds/#securing-sds-with-pod-security-policies) (<https://istio.io/docs/tasks/security/citadel-config/auth-sds/#securing-sds-with-pod-security-policies>).

If you need to enable a supported feature that isn't enabled by default in the ASM profile, you can set configuration parameters individually on the command line using `--set values`. For example, to set mTLS to **STRICT** mode by default:

Alternatively, you can specify the configuration in a YAML file and pass it to `istioctl` using the `-f` option.

Whichever approach that you use to customize the ASM configuration profile, make sure the feature that you enable [is supported](https://istio.io/docs/service-mesh/docs/supported-features) ([/service-mesh/docs/supported-features](https://istio.io/docs/service-mesh/docs/supported-features)). For more information, see [Customizing the configuration](https://istio.io/docs/setup/install/istioctl/#customizing-the-configuration) (<https://istio.io/docs/setup/install/istioctl/#customizing-the-configuration>).

Check that the control plane Pods in `istio-system` are up:

Expect to see output similar to the following:

Once Istio is installed and all its components are running, you can try deploying one of the sample applications provided with the installation. In this tutorial, we'll install [BookInfo](https://istio.io/docs/guides/bookinfo.html) (<https://istio.io/docs/guides/bookinfo.html>). This is a simple mock bookstore application made up of four services that provide a web product page, book details, reviews (with several versions of the review service), and ratings - all managed using Istio. You can find the source code and all the other files used in this example in your Istio installation's [samples/bookinfo](https://github.com/istio/istio/tree/master/samples/bookinfo) (<https://github.com/istio/istio/tree/master/samples/bookinfo>) directory.

Following these steps deploys the BookInfo application's services in an Istio-enabled environment, with Envoy sidecar proxies injected alongside each service to provide Istio functionality.

1. Ensure you're still in the root of the Istio installation directory on your cluster admin machine.
2. Deploy the application using `kubectl apply` and `istioctl kube-inject`. The `kube-inject` command updates the BookInfo deployment so that a sidecar is deployed in each application Pod along with the service.
3. Confirm that the application has been deployed correctly by running the following commands:

Output:

and

Output:

4. Finally, define the ingress gateway routing for the application:

Now that it's deployed, you can see the BookInfo application in action.

To see if the BookInfo application is working, you need to send traffic to the ingress gateway. Get the external IP address of the ingress gateway as follows:

Output:

In this example, the address of the ingress service is `35.239.7.64`:

1. Check that the BookInfo app is running with `curl`:

If the response shows `200`, it means the application is working properly with Istio.

2. Now point your browser to `http://$GATEWAY_URL/productpage` to view the BookInfo web page. If you refresh the page several times, you should see different versions of reviews shown in the product page, presented in a round robin style (red stars, black stars, no stars), since we haven't yet used Istio to control the version routing.

If you want to try deploying one of your own applications, just follow the same procedure with your own YAML deployment: Istio requires no changes to the application itself. Note that the application must use HTTP/1.1 or HTTP/2.0 protocol for all its HTTP traffic because the Envoy proxy doesn't support HTTP/1.0: it relies on headers that aren't present in HTTP/1.0 for routing.

You can either use `kube-inject` to add the sidecars when deploying the application, as in our example, or enable [Istio's automatic sidecar injection](#)

([/istio/docs/istio-on-gke/installing#enabling_sidecar_injection](https://istio/docs/istio-on-gke/installing#enabling_sidecar_injection)) for the namespace where your application is running.

Stackdriver Monitoring in the Cloud Console is now Generally Available and the default experience. For a limited period you also have the option to use the classic Stackdriver Monitoring console. For more information, see [Stackdriver Monitoring in the Cloud Console](https://cloud.google.com/monitoring/docs/monitoring_in_console) ([/monitoring/docs/monitoring_in_console](https://cloud.google.com/monitoring/docs/monitoring_in_console)).

By default, Anthos Service Mesh sends metrics related to your services (such as the number of requests received by a particular service) to [Stackdriver Monitoring](https://cloud.google.com/monitoring/docs/) ([/monitoring/docs/](https://cloud.google.com/monitoring/docs/)), where they appear in the Metrics Explorer. You can use these metrics to create charts and alerts, letting you monitor your services over time and receive alerts when, for example, a service is nearing a specified number of requests. You can also combine these metrics using filters and aggregations with Stackdriver Monitoring's built-in metrics to get new insights into your service behavior.

To view metrics:

1. In the Google Cloud Console, select **Monitoring**, or use the following button:

[Go to Monitoring](https://console.cloud.google.com/monitoring) (<https://console.cloud.google.com/monitoring>)

2. Select **Resources > Metrics Explorer**.

For a full list of metrics, see [Istio metrics](https://cloud.google.com/monitoring/api/metrics_istio) ([/monitoring/api/metrics_istio](https://cloud.google.com/monitoring/api/metrics_istio)) in the Stackdriver Monitoring documentation.

To uninstall the Anthos Service Mesh components, run the following command from your cluster admin machine:

Anthos Service Mesh is powered by Istio. You can learn more about Istio on the [Istio documentation site](https://istio.io) (<https://istio.io>).