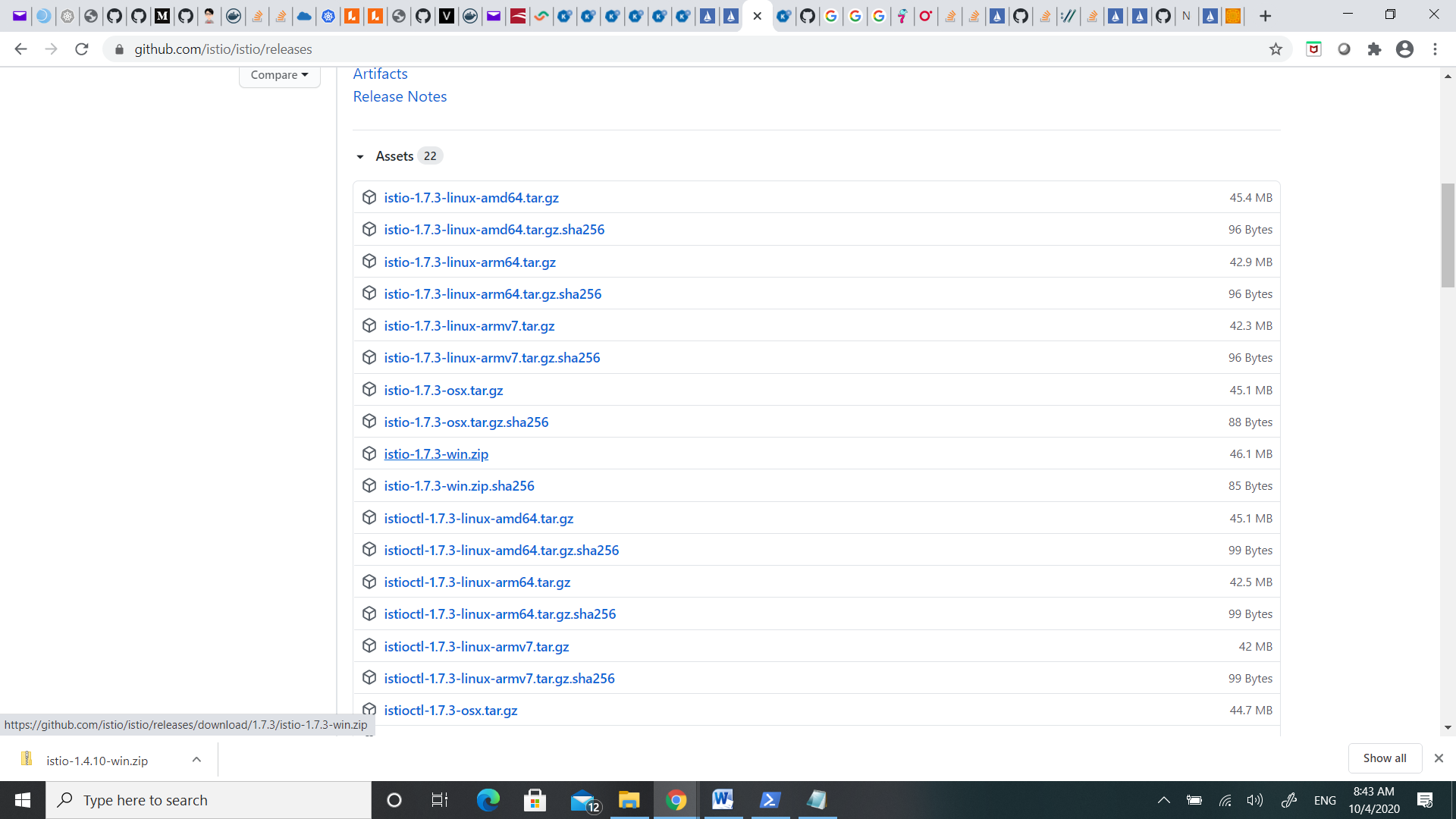
**INSTALL/CONFIGURE ISTIO FOR BLUE/GREEN OR CANARY DEPLOYMENTS**

Go to [**https://github.com/istio/istio/releases**](https://github.com/istio/istio/releases)

Select the appropriate download.

I selected [istio-1.7.3-win.zip](https://github.com/istio/istio/releases/download/1.7.3/istio-1.7.3-win.zip) since I have Windows10 Pro. Extract the zip file. Enter to the path to istioctl to your path. In windows, for example, add the path C:\bin\istio-1.7.0\bin to your environment variables. Verify you have istioctl in your path on the cmd cli. Issue the command which istioctl or istioctl version.



Go to [**https://istio.io/latest/docs/setup/install/istioctl/**](https://istio.io/latest/docs/setup/install/istioctl/) page and follow instructions to install istio using istioctl commands.

The simplest option is to install the default Istio [configuration profile](https://istio.io/latest/docs/setup/additional-setup/config-profiles/) using the following command. The default profile is recommended for production use.

**$ istioctl install**

**Output from the powershell:**

PS C:\eclipseWorkspace2020-09\MessageBoardApp> **istioctl install**

This will install the default Istio profile into the cluster. Proceed? (y/N) y

Detected that your cluster does not support third party JWT authentication. Falling back to less secure first party JWT. See https://istio.io/docs/ops/best-practices/security/#configure-third-party-service-account-tokens for details.

✔ Istio core installed

✔ Istiod installed

✔ Ingress gateways installed

✔ Installation complete

PS C:\eclipseWorkspace2020-09\MessageBoardApp>

**Check what’s installed**

The istioctl command saves the IstioOperator CR that was used to install Istio in a copy of the CR named installed-state. Instead of inspecting the deployments, pods, services and other resources that were installed by Istio, for example:

$ **kubectl -n istio-system get deploy**

*NAME READY UP-TO-DATE AVAILABLE AGE*

***istio-ingressgateway 1/1 1 1 49m***

***istiod 1/1 1 1 49m***

**Verify install:**

PS C:\eclipseWorkspace2020-09\MessageBoardApp\.istio\.deploy> **istioctl manifest generate > generated-manifest.yaml**

PS C:\eclipseWorkspace2020-09\MessageBoardApp\.istio\.deploy> **istioctl verify-install -f generated-manifest.yaml**

**Output from console:**

**PS C:\eclipseWorkspace2020-09\MessageBoardApp\.istio\.deploy> istioctl verify-install -f generated-manifest.yaml**

CustomResourceDefinition: adapters.config.istio.io.default checked successfully

CustomResourceDefinition: attributemanifests.config.istio.io.default checked successfully

CustomResourceDefinition: authorizationpolicies.security.istio.io.default checked successfully

CustomResourceDefinition: destinationrules.networking.istio.io.default checked successfully

CustomResourceDefinition: envoyfilters.networking.istio.io.default checked successfully

CustomResourceDefinition: gateways.networking.istio.io.default checked successfully

CustomResourceDefinition: handlers.config.istio.io.default checked successfully

CustomResourceDefinition: httpapispecbindings.config.istio.io.default checked successfully

CustomResourceDefinition: httpapispecs.config.istio.io.default checked successfully

CustomResourceDefinition: instances.config.istio.io.default checked successfully

CustomResourceDefinition: istiooperators.install.istio.io.default checked successfully

CustomResourceDefinition: peerauthentications.security.istio.io.default checked successfully

CustomResourceDefinition: quotaspecbindings.config.istio.io.default checked successfully

CustomResourceDefinition: quotaspecs.config.istio.io.default checked successfully

CustomResourceDefinition: requestauthentications.security.istio.io.default checked successfully

CustomResourceDefinition: rules.config.istio.io.default checked successfully

CustomResourceDefinition: serviceentries.networking.istio.io.default checked successfully

CustomResourceDefinition: sidecars.networking.istio.io.default checked successfully

CustomResourceDefinition: templates.config.istio.io.default checked successfully

CustomResourceDefinition: virtualservices.networking.istio.io.default checked successfully

CustomResourceDefinition: workloadentries.networking.istio.io.default checked successfully

ServiceAccount: istio-ingressgateway-service-account.istio-system checked successfully

ServiceAccount: istio-reader-service-account.istio-system checked successfully

ServiceAccount: istiod-service-account.istio-system checked successfully

ClusterRole: istio-reader-istio-system.default checked successfully

ClusterRole: istiod-istio-system.default checked successfully

ClusterRoleBinding: istio-reader-istio-system.default checked successfully

ClusterRoleBinding: istiod-pilot-istio-system.default checked successfully

ValidatingWebhookConfiguration: istiod-istio-system.default checked successfully

EnvoyFilter: metadata-exchange-1.6.istio-system checked successfully

EnvoyFilter: metadata-exchange-1.7.istio-system checked successfully

EnvoyFilter: stats-filter-1.6.istio-system checked successfully

EnvoyFilter: stats-filter-1.7.istio-system checked successfully

EnvoyFilter: tcp-metadata-exchange-1.6.istio-system checked successfully

EnvoyFilter: tcp-metadata-exchange-1.7.istio-system checked successfully

EnvoyFilter: tcp-stats-filter-1.6.istio-system checked successfully

EnvoyFilter: tcp-stats-filter-1.7.istio-system checked successfully

ConfigMap: istio.istio-system checked successfully

ConfigMap: istio-sidecar-injector.istio-system checked successfully

MutatingWebhookConfiguration: istio-sidecar-injector.default checked successfully

Deployment: istio-ingressgateway.istio-system checked successfully

Deployment: istiod.istio-system checked successfully

PodDisruptionBudget: istio-ingressgateway.istio-system checked successfully

PodDisruptionBudget: istiod.istio-system checked successfully

Role: istio-ingressgateway-sds.istio-system checked successfully

Role: istiod-istio-system.istio-system checked successfully

RoleBinding: istio-ingressgateway-sds.istio-system checked successfully

RoleBinding: istiod-istio-system.istio-system checked successfully

HorizontalPodAutoscaler: istio-ingressgateway.istio-system checked successfully

HorizontalPodAutoscaler: istiod.istio-system checked successfully

Service: istio-ingressgateway.istio-system checked successfully

Service: istiod.istio-system checked successfully

Checked 21 custom resource definitions

Checked 1 Istio Deployments

**Istio is installed successfully**

**Install istio addons ( Grafana, Jaeger, Kiali, Prometheus): Move to the samples directory of the istio install folder. The addons directory is located in the samples directory. Apply all addons yaml file.**

**PS C:\bin\istio-1.7.0\samples> kubectl apply -f addons**

**Verify Grafana, Jaeger, Kiali, Prometheus are running:**

**PS C:\bin\istio-1.7.0\samples> kubectl -n istio-system get deploy**

**NAME READY UP-TO-DATE AVAILABLE AGE**

**grafana** 1/1 1 1 7m32s

istio-ingressgateway 1/1 1 1 12m

istiod 1/1 1 1 12m

**jaeger** 1/1 1 1 7m32s

**kiali**  1/1 1 1 7m32s

**prometheus** 1/1 1 1 7m31s

PS C:\eclipseWorkspace2020-09\MessageBoardApp\.k8s> **kubectl -n istio-system get pods**

NAME READY STATUS RESTARTS AGE

grafana-767c5487d6-wrn4c 1/1 Running 0 10h

istio-ingressgateway-746548c687-68tgn 1/1 Running 0 10h

istiod-6c5f6f55ff-xcl95 1/1 Running 0 10h

jaeger-566c547fb9-x9xgg 1/1 Running 1 10h

kiali-89fd7f87b-wqs97 1/1 Running 0 10h

prometheus-788c945c9c-bqpww 2/2 Running 0 10h

To start the grafana, kiali, and Prometheus dashboards, issue the commands:

**istioctl dashboard kiali**

**istioctl dashboard Prometheus**

**istioctl dashboard grafana**

You can inspect the installed-state CR, to see what is installed in the cluster, as well as all custom settings. For example, dump its content into a YAML file using the following command:

$ **kubectl -n istio-system get IstioOperator installed-state -o yaml > installed-state.yaml**

The installed-state CR is also used to perform checks in some istioctl commands and should therefore not be removed.

**Display the list of available profiles**

You can display the names of Istio configuration profiles that are accessible to istioctl by using this command:

**$ istioctl profile list**

*Istio configuration profiles:*

*minimal*

*preview*

*remote*

*default*

*demo*

*empty*

**Display the configuration of a profile**

You can view the configuration settings of a profile. For example, to view the setting for the demo profile run the following command:

**$ istioctl profile dump default**

*components:*

*egressGateways:*

*- enabled: true*

*k8s:*

*resources:*

*requests:*

*cpu: 10m*

*memory: 40Mi*

### Configure gateways

Gateways are a special type of component, since multiple ingress and egress gateways can be defined. In the [IstioOperator API](https://istio.io/latest/docs/reference/config/istio.operator.v1alpha1/), gateways are defined as a list type.

The default profile installs one ingress gateway, called istio-ingressgateway. You can inspect the default values for this gateway and save the output to a file:

**$ istioctl profile dump --config-path components.ingressGateways > components.ingressGateways.txt**

$ **istioctl profile dump --config-path values.gateways.istio-ingressgateway > values.gateways.istio-ingressgateway.txt**

**Uninstall Istio**

To completely uninstall Istio from a cluster, run the following command:

**$ istioctl x uninstall --purge**

The optional --purge flag will remove all Istio resources, including cluster-scoped resources that may be shared with other Istio control planes.

Alternatively, to remove only a specific Istio control plane, run the following command:

$ **istioctl x uninstall <your original installation options>**

or

**$ istioctl manifest generate <your original installation options> | kubectl delete -f -**

The control plane namespace (e.g., istio-system) is not removed by default. If no longer needed, use the following command to remove it:

$ **kubectl delete namespace istio-system**

**To quickly evaluate istio follow this guide:** <https://istio.io/latest/docs/setup/getting-started/>

In a couple minutes you will have

1. [Download and install Istio](https://istio.io/latest/docs/setup/getting-started/#download)
2. [Deploy the sample application](https://istio.io/latest/docs/setup/getting-started/#bookinfo)
3. [Open the application to outside traffic](https://istio.io/latest/docs/setup/getting-started/#ip)
4. [View the dashboard](https://istio.io/latest/docs/setup/getting-started/#dashboard)

When you’re finished experimenting with the Bookinfo sample, uninstall and clean it up using the following instructions:

1. Delete the routing rules and terminate the application pods
2. $ [samples/bookinfo/platform/kube/cleanup.sh](https://raw.githubusercontent.com/istio/istio/release-1.7/samples/bookinfo/platform/kube/cleanup.sh)
3. Confirm shutdown
4. $ kubectl get virtualservices #-- there should be no virtual services
5. $ kubectl get destinationrules #-- there should be no destination rules
6. $ kubectl get gateway #-- there should be no gateway

$ kubectl get pods #-- the Bookinfo pods should be deleted

USE PORT-FORWARD TO FORWARD THE CONTAINER PORT TO OUTSIDE THE KUBERNETES CLUSTER FOR ACCESSING THE WEBAPP VIA BROWSER.

EITHER PORT-FORWARD USING THE POD NAME TO A PORT (8083) THAT YOU CAN USE IN YOUR BROWSER:

kubectl port-forward messageboardapp-v1-9d8657f55-g74fn 8083

OR, PORT-FORWARD THE ISTIO INGRESSGATEWAY (80) TO THE PORT YOU WANT TO USE IN YOUR BROWSER (8088):

kubectl port-forward svc/istio-ingressgateway 8088:80 -n istio-system

**INSTALL ISTIO ADDONS: (grafana, kiali, prometheus, jaeger)**

**MOVE TO THE SAMPLES IN THE ISTIO TOP DIRECTORY:**

PS C:\bin\istio-1.7.0\samples> **kubectl apply -f addons**

serviceaccount/grafana unchanged

configmap/grafana unchanged

service/grafana unchanged

deployment.apps/grafana configured

configmap/istio-grafana-dashboards configured

configmap/istio-services-grafana-dashboards configured

deployment.apps/jaeger unchanged

service/tracing unchanged

service/zipkin unchanged

customresourcedefinition.apiextensions.k8s.io/monitoringdashboards.monitoring.kiali.io unchanged

serviceaccount/kiali unchanged

configmap/kiali unchanged

clusterrole.rbac.authorization.k8s.io/kiali-viewer unchanged

clusterrole.rbac.authorization.k8s.io/kiali unchanged

clusterrolebinding.rbac.authorization.k8s.io/kiali unchanged

service/kiali unchanged

deployment.apps/kiali unchanged

monitoringdashboard.monitoring.kiali.io/envoy created

monitoringdashboard.monitoring.kiali.io/go created

monitoringdashboard.monitoring.kiali.io/kiali created

monitoringdashboard.monitoring.kiali.io/micrometer-1.0.6-jvm-pool created

monitoringdashboard.monitoring.kiali.io/micrometer-1.0.6-jvm created

monitoringdashboard.monitoring.kiali.io/micrometer-1.1-jvm created

monitoringdashboard.monitoring.kiali.io/microprofile-1.1 created

monitoringdashboard.monitoring.kiali.io/microprofile-x.y created

monitoringdashboard.monitoring.kiali.io/nodejs created

monitoringdashboard.monitoring.kiali.io/quarkus created

monitoringdashboard.monitoring.kiali.io/springboot-jvm-pool created

monitoringdashboard.monitoring.kiali.io/springboot-jvm created

monitoringdashboard.monitoring.kiali.io/springboot-tomcat created

monitoringdashboard.monitoring.kiali.io/thorntail created

monitoringdashboard.monitoring.kiali.io/tomcat created

monitoringdashboard.monitoring.kiali.io/vertx-client created

monitoringdashboard.monitoring.kiali.io/vertx-eventbus created

monitoringdashboard.monitoring.kiali.io/vertx-jvm created

monitoringdashboard.monitoring.kiali.io/vertx-pool created

monitoringdashboard.monitoring.kiali.io/vertx-server created

serviceaccount/prometheus unchanged

configmap/prometheus unchanged

clusterrole.rbac.authorization.k8s.io/prometheus unchanged

clusterrolebinding.rbac.authorization.k8s.io/prometheus unchanged

service/prometheus unchanged

deployment.apps/prometheus configured

**VERIFY THAT THE ADDOS ARE RUNNING** (IN THE ISTIO-SYSTEM NAMESPACE)

PS C:\bin\istio-1.7.0\samples> **kubectl get pods -n istio-system**

NAME READY STATUS RESTARTS AGE

grafana-767c5487d6-tvkqc 1/1 Running 0 3m35s

istio-ingressgateway-746548c687-vw57s 1/1 Running 0 8m10s

istiod-6c5f6f55ff-nrmqs 1/1 Running 0 8m46s

jaeger-566c547fb9-pzg8d 1/1 Running 0 3m35s

kiali-89fd7f87b-fvgp6 1/1 Running 0 3m35s

prometheus-788c945c9c-r4kgt 2/2 Running 0 3m35s

**START KIALI DASHBOARD**

**PS C:\Users\monica>istioctl dashboard kiali**

**Open the kiali dashboard in a browser:**

[**http://localhost:20001/kiali/console/overview?duration=60&refresh=15000**](http://localhost:20001/kiali/console/overview?duration=60&refresh=15000)

**START GRAFANA DASHBOARD**

PS C:\bin\istio-1.7.0\samples> **kubectl -n istio-system get pod -l app=grafana -o jsonpath='{.items[0].metadata.name}'**

**grafana-767c5487d6-tvkqc**

PS C:\bin\istio-1.7.0\samples> **kubectl -n istio-system port-forward grafana-767c5487d6-tvkqc 3010:3000**

Forwarding from 127.0.0.1:3010 -> 3000

Forwarding from [::1]:3010 -> 3000

**Or issue this command ( using Cygwin, this command gives an error in the powershell ):**

**kubectl -n istio-system port-forward $(kubectl -n istio-system get pod -l app=grafana -o jsonpath='{.items[0].metadata.name}') 3000:3000 &**

**Open the links in a browser.**

**This opens the istio service grafana dashboard:**

[**http://localhost:3000/d/LJ\_uJAvmk/istio-service-dashboard?orgId=1&refresh=10s**](http://localhost:3000/d/LJ_uJAvmk/istio-service-dashboard?orgId=1&refresh=10s)

**This opens the istio mesh grafana dashboard:**

[**http://localhost:3000/d/G8wLrJIZk/istio-mesh-dashboard?orgId=1&refresh=5s**](http://localhost:3000/d/G8wLrJIZk/istio-mesh-dashboard?orgId=1&refresh=5s)

**START PROMETHEUS DASHBOARD**

**istioctl dashboard prometheus**

**VERIFY PROMETHEUS SERVICE IS AVAILABLE:**

PS C:\eclipseWorkspace2020-09\MessageBoardApp2> **kubectl -n istio-system get svc prometheus**

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

prometheus ClusterIP 10.108.52.209 <none> 9090/TCP 4h49m

**VERIFY GRAFANA SERVICE IS AVAILABLE:**

PS C:\eclipseWorkspace2020-09\MessageBoardApp2> **kubectl -n istio-system get svc grafana**

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

grafana ClusterIP 10.110.0.153 <none> 3000/TCP 4h51m

**VERIFY KIALI SERVICE IS AVAILABLE:**

PS C:\eclipseWorkspace2020-09\MessageBoardApp2> **kubectl -n istio-system get svc kiali**

NAME TYPE CLUSTER-IP EXTERNAL-IP PORT(S) AGE

kiali ClusterIP 10.97.69.77 <none> 20001/TCP,9090/TCP 4h54m

**TO TURN ON AUTOMATIC SIDECAR INJECTION:**

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

**TO VERIFY SIDECAR INJECTION IS ON:**

PS C:\eclipseWorkspace2020-09\MessageBoardApp2> **kubectl get namespace -L istio-injection**

NAME STATUS AGE ISTIO-INJECTION

default Active 5h10m enabled

docker Active 5h9m

istio-system Active 4h52m disabled

kube-node-lease Active 5h10m

kube-public Active 5h10m

kube-system Active 5h10m

kubernetes-dashboard Active 4h56m