INSTALLING ISTIO ON GOOGLE KUBERNETES ENGINE USING KUBERNETES INGRESS OBJECT

## Description:

Istio by default create the istio ingress gateway as a load balancer - those of us using google kubernetes engine who wish to take advantage of the google load balancing capabilities without having to sacrifice the istio experience. This tutorial enables you to utilize istio, change the istio ingress gateway to a nodeport, and configure a kubernetes ingress object with associated google load balancer.

[**Description:**](#_tzam08d1wljn) **1**

[**PREREQUISITES**](#_s3trt0oj7wu5) **2**

[**Create new cluster**](#_8v8029ygmg70) **2**

[**Connect to cluster**](#_qtjmylxngyq7) **2**

[**DOWNLOAD and SETUP initial istio**](#_9g7f98wdakjw) **3**

[**VERIFY ability to install istio**](#_31m9cggn8ul4) **3**

[**INSTALL ISTIO**](#_heg5u1gtokt3) **3**

[**Add a namespace label to instruct Istio to automatically inject Envoy sidecar proxies when you deploy your application later:**](#_ovla0t416aap) **3**

[**Deploy the application ( note: ensure you are still in istio-1.5.1 root folder )**](#_a09bojzdqv8f) **3**

[**Deploy the gateway**](#_ecgqpibufe6y) **3**

[**Deploy health virtual service**](#_chtpi4crpy8i) **4**

[**Patch istio ingress-gateway**](#_h2f178cmccyu) **4**

[**Create Kubernetes Ingress Object**](#_zf1ufsdc17d9) **5**

[**Configure Health Check on Load Balancer Backend**](#_1xmbg7mwiuis) **5**

[**Reserve the IP associated with the kubernetes ingress ( making it static )**](#_qrc6s3kgy2ed) **6**

[**Configure Load Balancer for SSL**](#_a23zkxa5m0r2) **6**

[**Visit your product page**](#_glt46t5b6t9h) **6**

[**LINKS OF IMPORTANCE**](#_wrurq4low121) **7**

## PREREQUISITES

1. CREATE A NEW PROJECT
2. SET CURRENT NAMESPACE TO DEFAULT

kubectl config set-context --current --namespace=default

1. ENSURE YOUR PROJECT ID IS SET PROPER

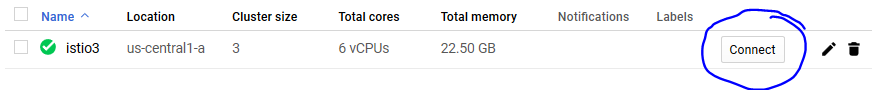
gcloud config set project {GCLOUD PROJECT ID HERE}

## Create new cluster

* MACHINE TYPE minimum n1-standard-2
* SELECT KUBERNETES RELEASE CHANNEL regular ( 1.15.9-gke.24 as of 4/16/2020 )
* DO NOT INSTALL ISTIO VIA GOOGLE CLOUD ADDONS
* enable HorizontalPodAutoscaling
* enable HttpLoadBalancing
* enable --enable-ip-alias option makes the cluster [VPC-native](https://cloud.google.com/kubernetes-engine/docs/how-to/alias-ips).
* enable --enable-stackdriver-kubernetes option

## Connect to cluster

ie: gcloud container clusters get-credentials istio3 --zone us-central1-a --project istiocodelab



1. DOWNLOAD and SETUP initial istio

<https://istio.io/docs/setup/getting-started/#download>

This will add the istioctl client to your path:

curl -L https://istio.io/downloadIstio | sh -

cd istio-1.5.1

export PATH=$PWD/bin:$PATH

## VERIFY ability to install istio

istioctl verify-install

## INSTALL ISTIO

***NOTE: For this installation, we use the demo*** [***configuration profile***](https://istio.io/docs/setup/additional-setup/config-profiles/)***. It’s selected to have a good set of defaults for testing, but there are other profiles for production or performance testing.***

istioctl manifest apply --set profile=demo

## Add a namespace label to instruct Istio to automatically inject Envoy sidecar proxies when you deploy your application later:

*kubectl label namespace default istio-injection=enabled*

## Deploy the application ( note: ensure you are still in istio-1.5.1 root folder )

*kubectl apply -f samples/bookinfo/platform/kube/bookinfo.yaml*

## Deploy the gateway

*kubectl apply -f samples/bookinfo/networking/bookinfo-gateway.yaml*

## Deploy health virtual service

*kubectl apply -f - <<EOF*

*apiVersion: networking.istio.io/v1alpha3*

*kind: VirtualService*

*metadata:*

*name: healthservice*

*spec:*

*gateways:*

*- istio-ingressgateway*

*hosts:*

*- "\*"*

*http:*

*- match:*

*- uri:*

*exact: /healthz/ready*

*route:*

*- destination:*

*host: istio-ingressgateway.istio-system.svc.cluster.local*

*port:*

*number: 15020*

*EOF*

## Patch istio ingress-gateway

Create a [JSON Patch](https://tools.ietf.org/html/rfc6902) file to make changes to the Istio ingress gateway:

*cat <<EOF > istio-ingress-patch.json*

*[*

*{*

*"op": "replace",*

*"path": "/spec/type",*

*"value": "NodePort"*

*},*

*{*

*"op": "remove",*

*"path": "/status"*

*}*

*]*

*EOF*

Apply the patch file and add the Istio ingress gateway as a backend:

*kubectl -n istio-system patch svc istio-ingressgateway \*

*--type=json -p="$(cat istio-ingress-patch.json)" \*

*--dry-run=true -o yaml | kubectl apply -f -*

*kubectl annotate svc istio-ingressgateway -n istio-system cloud.google.com/neg='{"exposed\_ports": {"80":{}}}'*

## Create Kubernetes Ingress Object

*kubectl apply -f - <<EOF*

*apiVersion: extensions/v1beta1*

*kind: Ingress*

*metadata:*

*name: primary-ingress*

*namespace: istio-system*

*spec:*

*backend:*

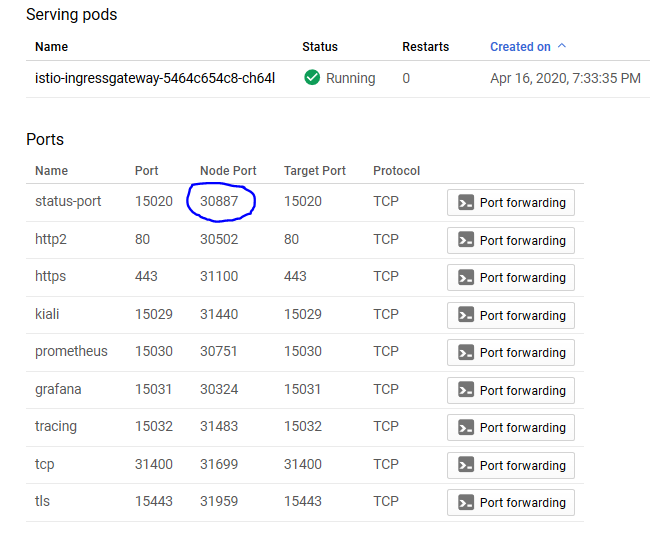
*serviceName: istio-ingressgateway*

*servicePort: 80*

*EOF*

## Configure Health Check on Load Balancer Backend

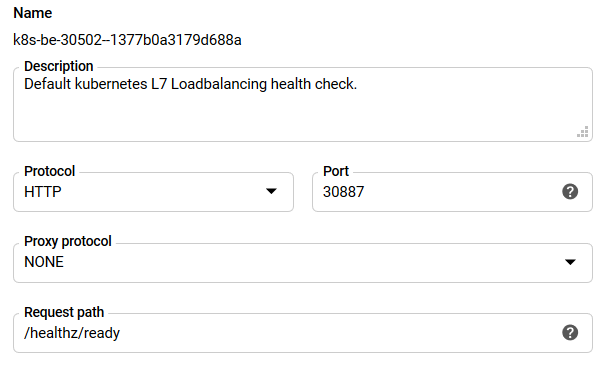
Target port is found in the istio-ingressgateway



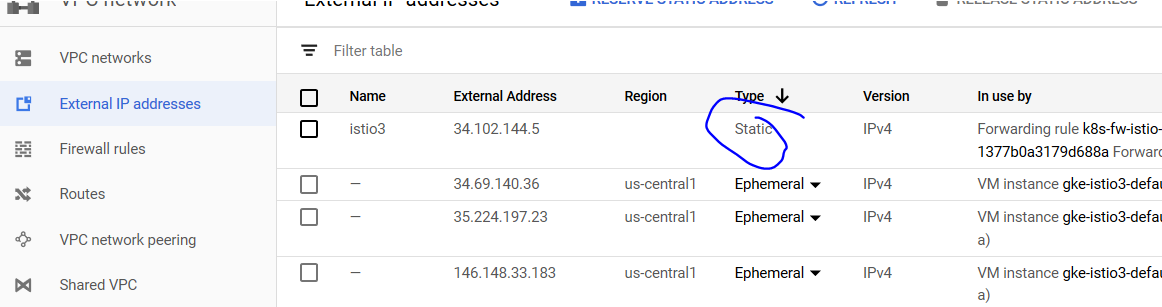
*Path /healthz/ready*

*Protocol HTTP*

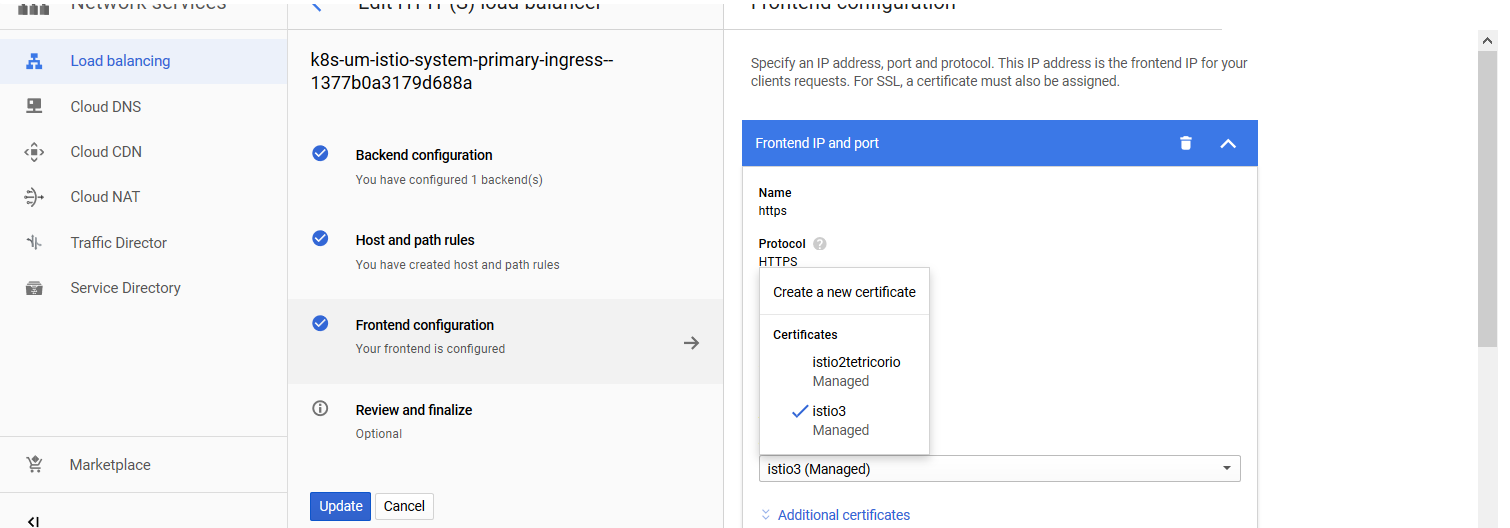
*Port 30887*

**

## Reserve the IP associated with the kubernetes ingress ( making it static )

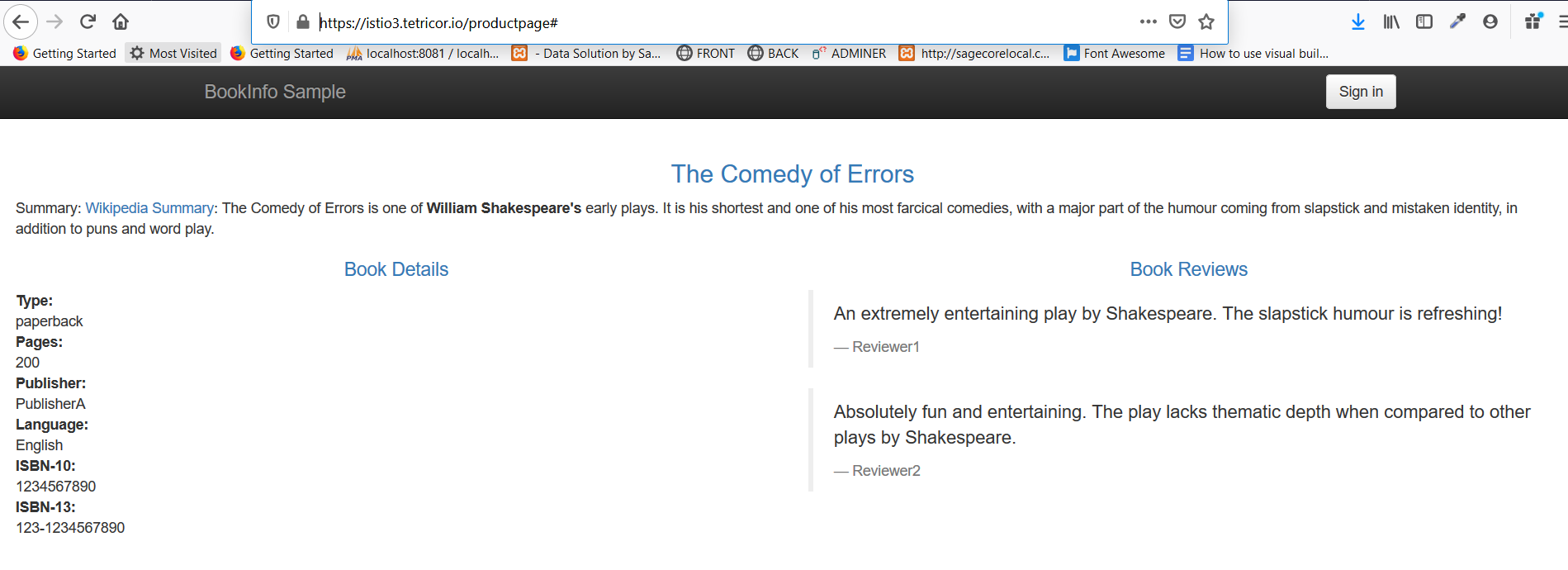


## Configure Load Balancer for SSL

* *Point a domain to the ip - wait for DNS to resolve*
* *Configure an HTTPS frontend - this is where your SSLs reside*

## Visit your product page

VISIT [https://{YOURDOMAIN}/productpage](about:blank) in your browser and verify you see

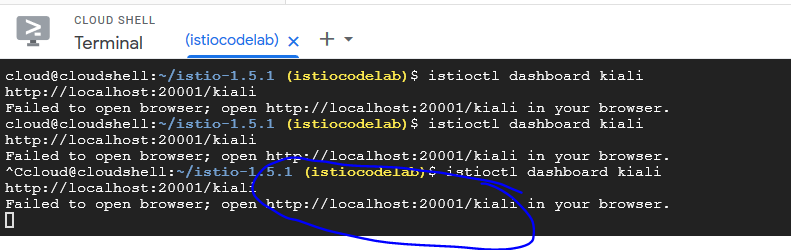


## Generate some test traffic in 1 tab

watch -n 1 curl [https://istiolab.tetricor.io/productpage](https://istio3.tetricor.io/productpage)

## Launch Kiali

istioctl dashboard kiali



## 

## NEXT STEPS

* Running a private cluster with same config
* kiali with NON default username

## 

## LINKS OF IMPORTANCE

1. ISTIO SETUP

<https://istio.io/docs/setup/getting-started/#download>

1. Create Health Request VirtualService

<https://cloud.google.com/solutions/integrating-https-load-balancing-with-istio-and-cloud-run-for-anthos-deployed-on-gke#handling_health_check_requests>

1. Modify istio-ingressgateway for use with kubernetes ingress

<https://cloud.google.com/solutions/integrating-https-load-balancing-with-istio-and-cloud-run-for-anthos-deployed-on-gke#modifying_the_istio_ingress_gateway_for_use_with_kubernetes_ingress>

1. Create kubernetes ingress object

<https://cloud.google.com/solutions/integrating-https-load-balancing-with-istio-and-cloud-run-for-anthos-deployed-on-gke#creating_a_kubernetes_ingress_object>

https://istio.io/docs/concepts/traffic-management/#introducing-istio-traffic-management

SHORTCUTS

KIALI

istioctl dashboard kiali

***CLICK*** THE LOCALHOST LINK IN CLOUD CONSOLE

