National College of Ireland

Configuration Manual

MSc Research Project
Cloud Computing

Abhishek Shitole
Student ID: x19206925

School of Computing
National College of Ireland

Supervisor: Rashid Mijumbi

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Abhishek Shitole		
Student ID:	x19206925		
Programme:	Cloud Computing	**Year:**	2022
Module:	MSc Research Project		
Supervisor:	Rashid Mijumbi		
Submission Due Date:	31/01/2022		
Project Title:	Configuration Manual		
Word Count:	495	**Page Count:**	6

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Abhishek Shitole
Date:	30th January 2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Configuration Manual

Abhishek Shitole
x19206925

1 Introduction

This configuration manual provides in detail, stepwise installation, configuration and deployment of all the software's, tools and files required for implementation of the proposed system.

2 Prerequisites

Below stated are hardware software and requirements for implementing this project.

Kubernetes Cluster on GKE	
Machine type	e2-standard-4
No of Nodes	3
Operating System	Debian Version 10
Total vCPU	12
Total Memory	48 GB
Cost	$0.134012/hr per node

Software	Description
Service Mesh	Istio 1.12.1
Monitoring	Prometheus
Data Visualization	Grafana
Kiali	Requires Istio installed
Locust	Requires Cluster with 3 node

3 Implementation

3.1 Deploying Kubernetes Cluster on Google Kubernetes Engine (GKE)

1. We first set environment variables for zones and cluster-name by using following command on cloud shell.

```
abhishekshitole2710@cloudshell:~ (ultimate-opus-333416)$ export CLUIST_LOCATION=us-central1-c
abhishekshitole2710@cloudshell:~ (ultimate-opus-333416)$ export IST_PROJ=$(gcloud config get-value project)
Your active configuration is: [cloudshell-1020]
```

Figure 1: Defining cluster location & zone

2. Run the below command to create a cluster named bookstore with 3 nodes and e2-standard machine type.

   gcloud container clusters create bookstoreistio --project=${CC_PROJ} --zone=${CLU_LOCATION} --machine-type=e2-standard-4 --num-nodes=2

3.1 Installation & Configuration of Service Mesh Istio

1. Create a installer.sh by typing vi installer.sh and paste the below codes in it.

```
echo "Downloading Istio"
curl -L https://istio.io/downloadIstio | sh -
cd istio-1.12.0
export PATH=$PWD/bin:$PATH
echo "Installing Istio"
istioctl install --set profile=demo -y
echo "Enabling istio proxy injection in default namespace"
kubectl label namespace default istio-injection=enabled
```

Figure 2: Installing Istio & Enabling proxy injection

2. Execute the script by typing **bash installer.sh**

3.2 Deploying Micro-services application

1. Clone the micro-services application from github by using following command

   git clone https://github.com/Abhishek-NCI/Mthesis.git

2. Create appinstaller.sh by typing vi appinstaller.sh and paste the below codes in it.

```
echo "Deploying Microservices on Pods"
kubectl apply -f Mthesis/bookinfo/platform/kube/bookinfo.yaml
echo "Checking if services are properly installed"
kubectl get services
echo "Checking if services are properly installed on pods"
kubectl get pods
echo "Applying networking configurations on Ingress-Gateway"
kubectl apply -f Mthesis/bookinfo/networking/bookinfo-gateway.yaml
```

Figure 3: Deploying bookinfo & applying gateway config

3. Execute the script by typing **bash appinstaller.sh** in cloud shell.

4. To check for validation issues in namespace following command is used

```
abhishekshitole2710@cloudshell:~ (ultimate-opus-333416)$ istioctl analyze

✓ No validation issues found when analyzing namespace: default.
```

Figure 4: Troubleshooting validation issues

3.3 Configuring Istio Ingress gateway & Exposing External traffic

1. Create a exttraffic.sh by typing vi exttraffic.sh and paste the below codes in it.

```
export INGRESS_HOST=$(kubectl -n istio-system get service istio-ingressgateway -o jsonpath='{.status.loadBalancer.ingress[0].ip}')
export INGRESS_PORT=$(kubectl -n istio-system get service istio-ingressgateway -o jsonpath='{.spec.ports[?(@.name=="http2")].port}')
export SECURE_INGRESS_PORT=$(kubectl -n istio-system get service istio-ingressgateway -o jsonpath='{.spec.ports[?(@.name=="https")].port}')
echo "Creating firewall rules for ports"
gcloud compute firewall-rules create allow-gateway-http --allow "tcp:$INGRESS_PORT"
gcloud compute firewall-rules create allow-gateway-https --allow "tcp:$SECURE_INGRESS_PORT"
export GATEWAY_URL=$INGRESS_HOST:$INGRESS_PORT
echo "$GATEWAY_URL"
echo "http://$GATEWAY_URL/productpage"
```

Figure 5: Configuration of istioingress gateway

2. Execute the script by typing **bash exttraffic.sh** in cloud shell.

3. Copy the output of this file and paste it onto the browser to see web application running.

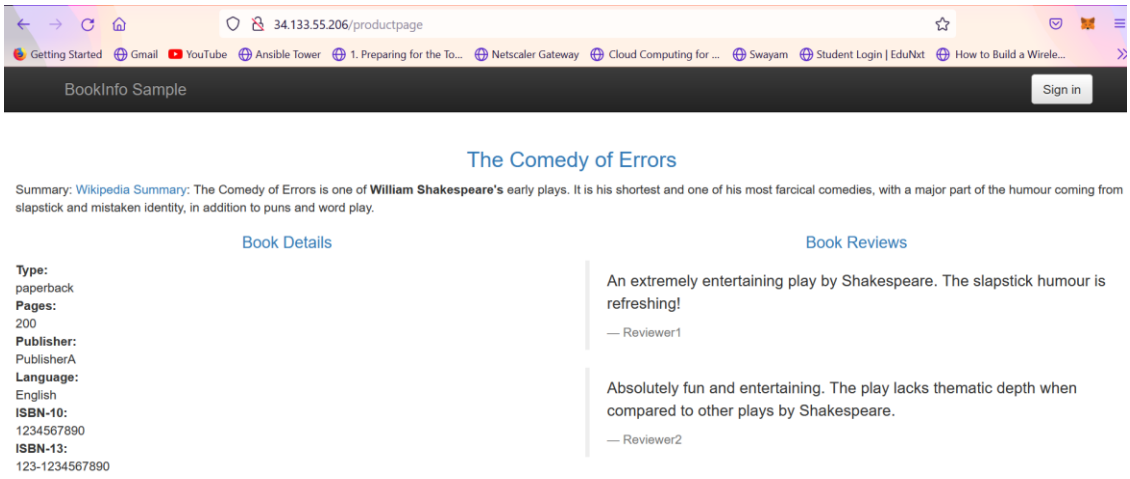


Figure 6: Accessing web application BookInfo

3.4 Installation of Monitoring & Visualisation Tools

1. Use below command to Install Kiali, Prometheus and Grafana as addon.

```
abhishekshitole2710@cloudshell:~ (ultimate-opus-333416)$ kubectl apply -f istio-1.12.1/samples/addons
```

Figure 7: Installing Addon's

2. Rollout deployments of Kiali, Prometheus and Grafana by using below commands.

```
abhishekshitole2710@cloudshell:~ (ultimate-opus-333416)$ kubectl rollout status deployment/kiali -n istio-system
deployment "kiali" successfully rolled out
abhishekshitole2710@cloudshell:~ (ultimate-opus-333416)$ kubectl rollout status deployment/prometheus -n istio-system
deployment "prometheus" successfully rolled out
abhishekshitole2710@cloudshell:~ (ultimate-opus-333416)$ kubectl rollout status deployment/grafana -n istio-system
deployment "grafana" successfully rolled out
```

Figure 8: Rolling out Kiali, Prometheus & Grafana

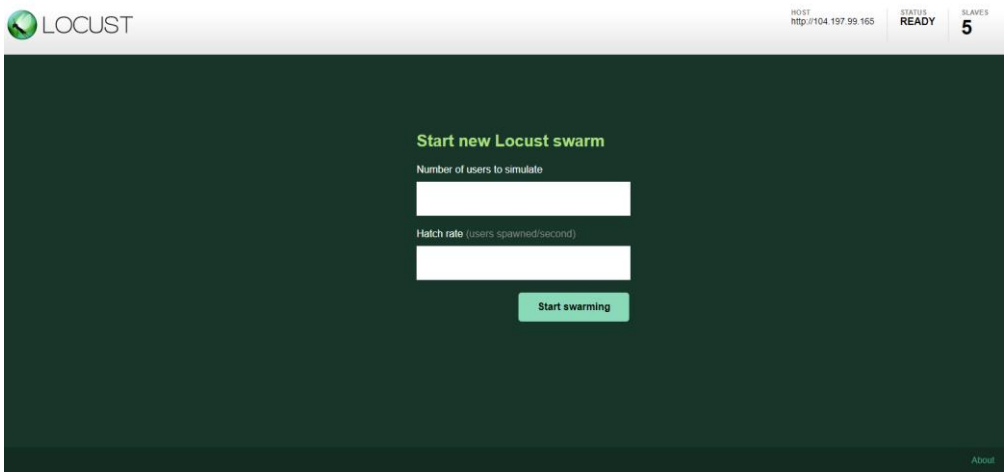
3. Kiali, Prometheus & Grafana can be accessed using below commands

```
abhishekshitole2710@cloudshell:~ (ultimate-opus-333416)$ istioctl dashboard kiali
abhishekshitole2710@cloudshell:~ (ultimate-opus-333416)$ istioctl dashboard prometheus
abhishekshitole2710@cloudshell:~ (ultimate-opus-333416)$ istioctl dashboard grafana
```

Figure 9: Accessing Kiali, Prometheus & Grafana dashboard

3.5 Generating Load on Application

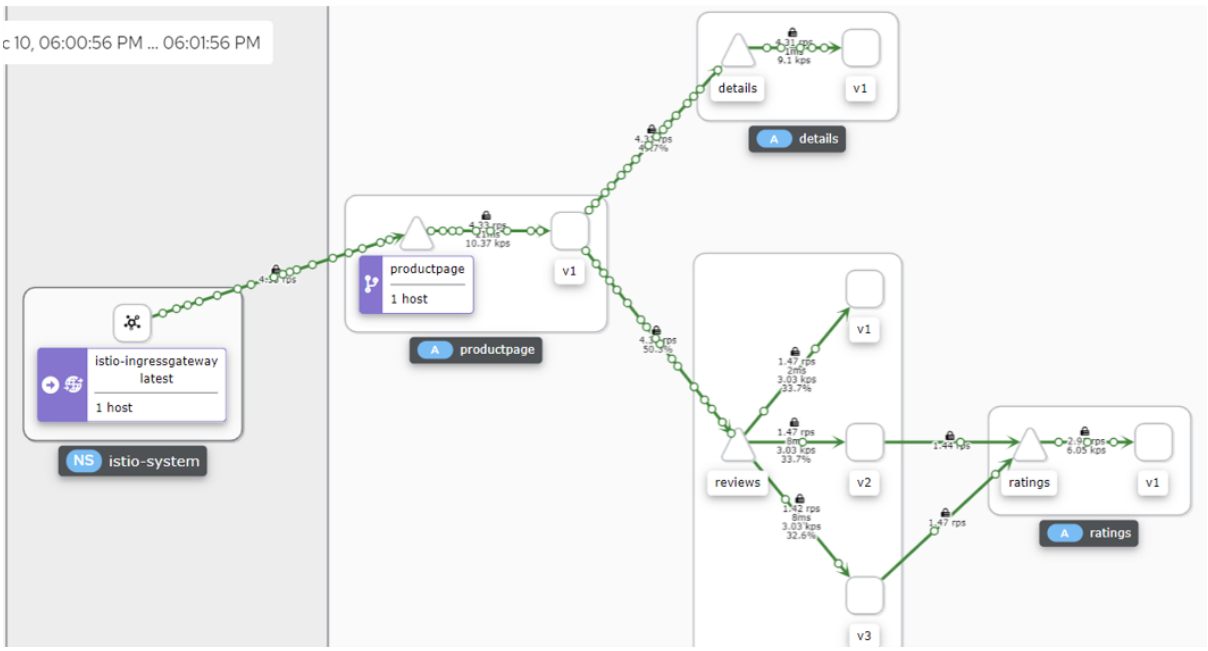
1. For Generation of load over application we first need to setup load testing tool locus on the Kubernetes cluster. [Refer to Cloud, G., 2021] for configuration and setting up load generator tool Locust.
2. After Locust has been setup, we now simulate the load over application for that we need to enter the overall users to simulate and users to produce per second.



4 Observations

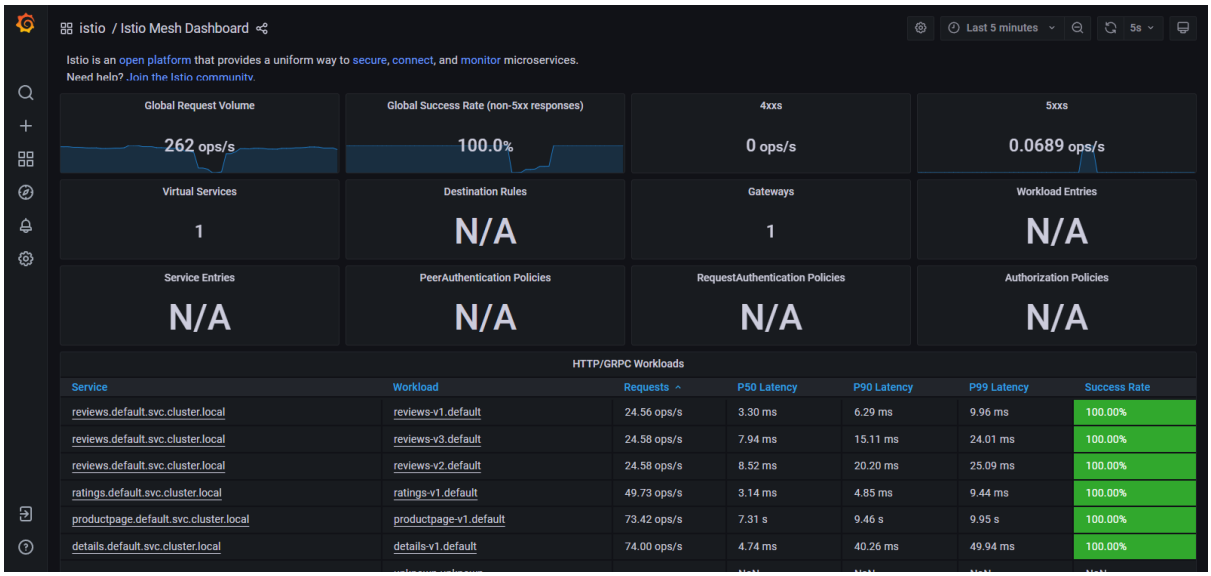
For observing the dynamic load balancing in microservices we open kiali dashboard. Detailed observation on incoming requests and resource consumption can been observed on Grafana dashboard.

4.1 Observation on Kiali

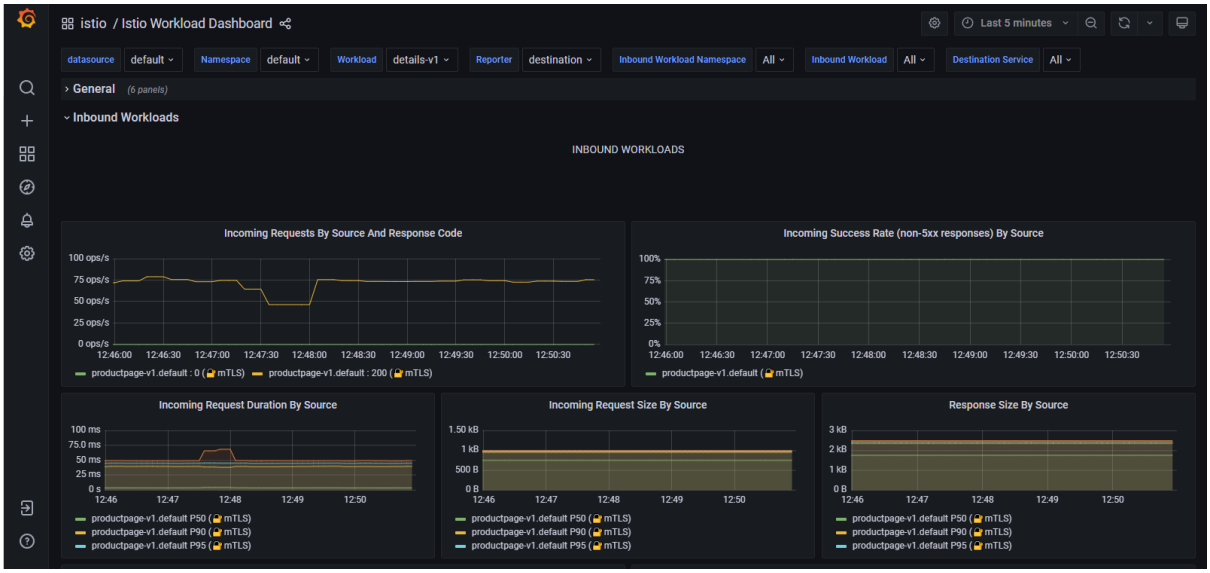


4.2 Observation on Grafana

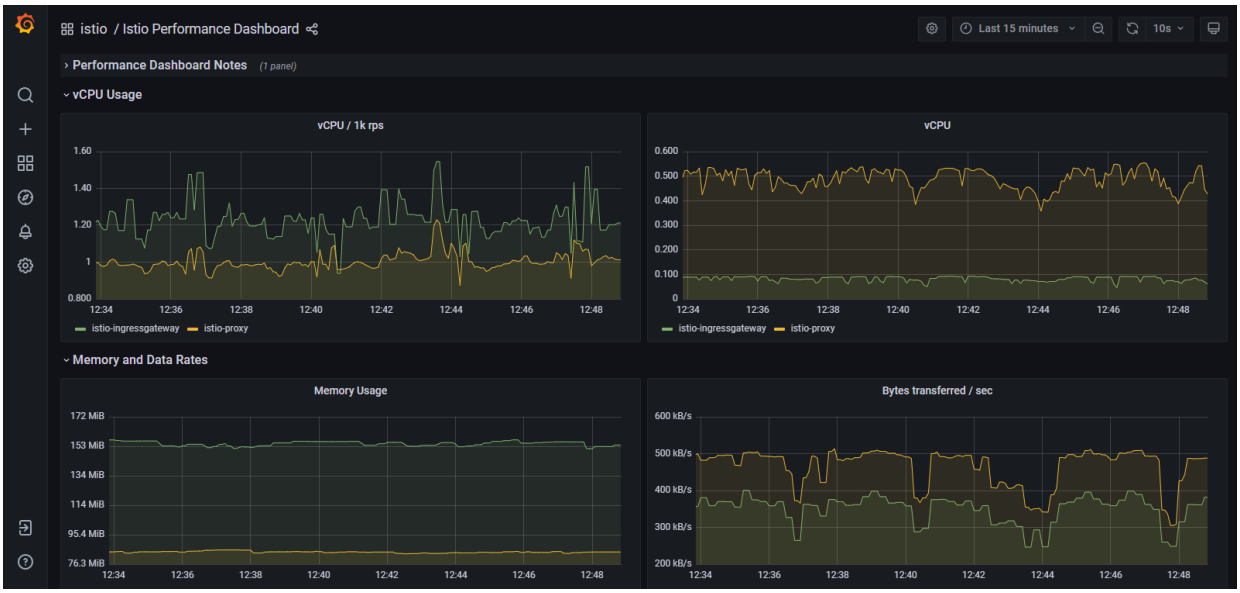
1. Mesh Dashboard



2. Workload Dashboard



3. Performance Dashboard



References

Cloud, G., 2021. Distributed load testing using Google Kubernetes Engine | Cloud Architecture Center | Google Cloud. [online] Google Cloud. Available at: <https://cloud.google.com/architecture/distributed-load-testing-using-gke>

Istio. 2021. Getting Started. [online] Available at: <https://istio.io/latest/docs/setup/getting-started/> [Accessed 16 December 2021].