

Configuration Manual

MSc Research Project
Cloud Computing

Prawal Saxena
Student ID: 21151334

School of Computing
National College of Ireland

Supervisor: Sean Heeney

National College of Ireland
Project Submission Sheet
School of Computing



Student Name:	Prawal Saxena
Student ID:	21151334
Programme:	Cloud Computing
Year:	2022
Module:	MSc Research Project
Supervisor:	Sean Heeney
Submission Due Date:	01/02/2023
Project Title:	Configuration Manual
Word Count:	759
Page Count:	10

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:	Prawal Saxena
Date:	27th January 2023

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	<input type="checkbox"/>
Attach a Moodle submission receipt of the online project submission , to each project (including multiple copies).	<input type="checkbox"/>
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.	<input type="checkbox"/>

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

Prawal Saxena
21151334

1 Architectural Diagram

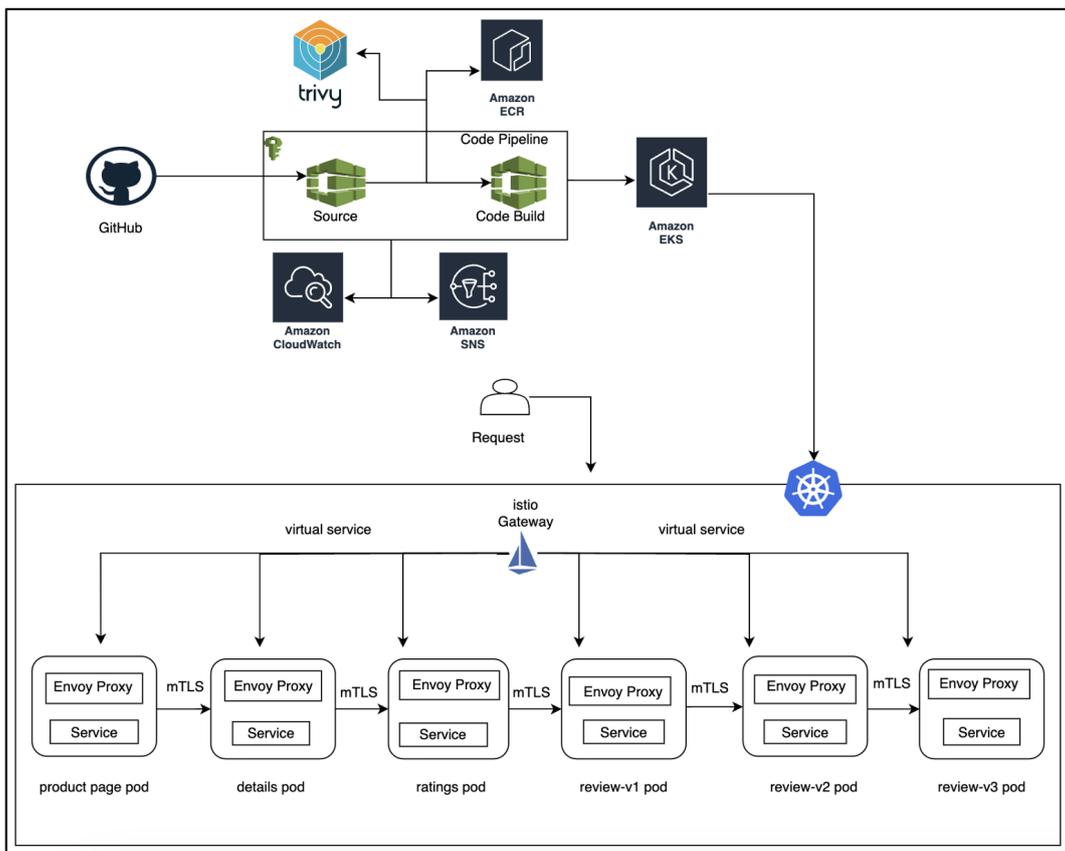


Figure 1: Architectural Diagram

2 Kubernetes Installation

In this project, I am installing Kubernetes on my local machine(macOS), it can be installed in ubuntu or any Linux OS. ([Kumar et al. \(2022\)](#))

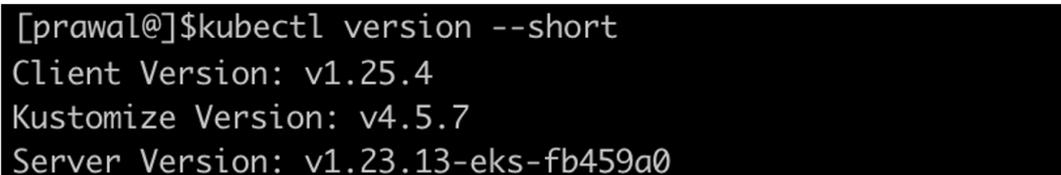
- I have configured AWS account in my local machine which has admin access to all AWS services. It can be done in another way as well in AWS create a Linux Ec2 then install Kubernetes in that. Attach IAM role with a policy that enables access to different services like ECR, EKS, and Codepipeline.

```

sudo apt update
sudo apt install -y docker.io
systemctl status docker
docker --version
sudo apt install -y apt-transport-https curl
curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg
| sudo apt-key add
sudo apt-add-repository "deb http://apt.kubernetes.io/ kubernetes-xenial
    main"
sudo apt update
sudo apt install -y kubelet kubeadm kubectl

```

- For checking if kubernetes successfully installed or not use kubectl version command:-



```

[prawal@]$kubectl version --short
Client Version: v1.25.4
Kustomize Version: v4.5.7
Server Version: v1.23.13-eks-fb459a0

```

Figure 2: Kubernetes version

2.1 EKS cluster creation and istio installation

- I am creating EKS cluster eksctl, so first I need to install eksctl by running these commands ([Paul SaveSoil \(2022\)](#)):

```

curl --silent --location
    "https://github.com/weaveworks/eksctl/releases/latest/download/eksctl_$(uname
    -s)_amd64.tar.gz" | tar xz -C /tmp
sudo mv /tmp/eksctl /usr/local/bin
export PATH=$PATH:/usr/local/bin
echo 'export PATH=$PATH:/usr/local/bin' >> ~/.bashrc
eksctl version

```

- After installing eksctl run the below command to create EKS cluster:-

```

eksctl create cluster --name research --version 1.23 --region us-east-1
    --nodegroup-name node-1-workers --node-type t2.medium --nodes 2
    --nodes-min 2 --nodes-max 2 --managed

```

- After successful installation go to AWS EKS service and see the details of the cluster
- In the local machine terminal run kubectl get nodes to check the nodes status

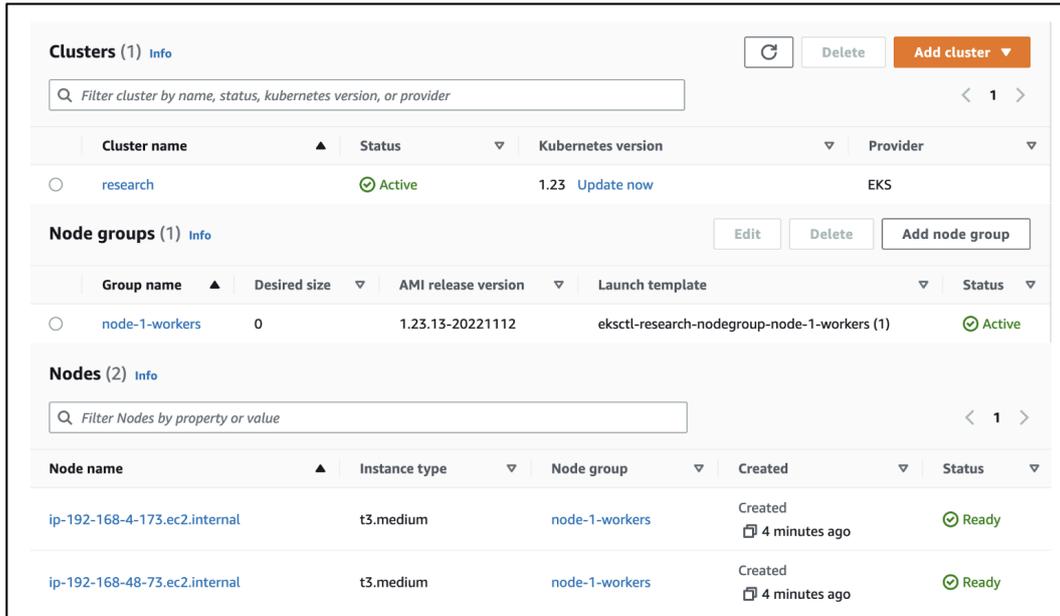


Figure 3: EKS Cluster Details

```
[prawal@]$ kubectl get nodes
NAME                                STATUS    ROLES    AGE   VERSION
ip-192-168-4-173.ec2.internal      Ready    <none>   22m   v1.23.13-eks-fb459a0
ip-192-168-48-73.ec2.internal      Ready    <none>   22m   v1.23.13-eks-fb459a0
[prawal@]$
```

Figure 4: EKS Cluster Nodes

- Next step to install Istio in the Kubernetes cluster in istio-system namespace ([istio \(2022\)](#)):

```
curl -L https://istio.io/downloadIstio | sh -
export PATH=$PWD/bin:$PATH
istioctl install --set profile=demo -y -n istio-system
```

- Create research namespace for all the application pods

```
kubectl label namespace research istio-injection=enabled
```

```
[prawal@]$ kubectl get ns --show-labels
istio-system      Active  9d   kubernetes.io/metadata.name=istio-system
research          Active  9d   istio-injection=enabled,kubernetes.io/metadata.name=research
```

Figure 5: Istio Enabled in Research Namespace

- Install Istio Addons that includes Kiali, Grafana dashboards.

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

3 Creating AWS Services

This section includes the creation of GitHub, ECR, and other AWS services.

- Creating Private GitHub Repository and upload the code and other files to it.
- Create Private ECR Repository in AWS.
- Next create Code Build Service.
 - Select GitHub as source.
 - Select Environment Image Managed image as choose Ubuntu as Operating system.
 - Create a new IAM role and provide full access policy of EKS and ECR in the role.
- Next creating the Code Pipeline in that select GitHub as source code then in build stage select the Code Build Project created in the above step.

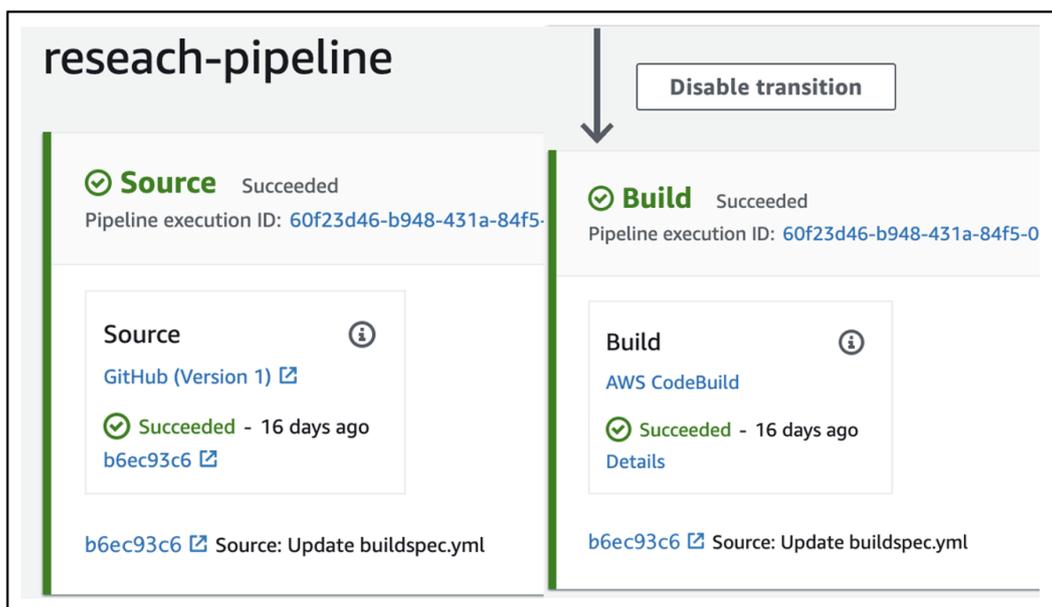


Figure 6: Code Pipeline

4 Important files that is needed in the Proejct

I have taken the application Code from Istio public GitHub repository. (Istio (2021))

- To run Code Build Project buildpsec.yml file is needed, in the file I need to write steps that will run in the project.
- Trivy installation command is written in buildspec file. (Trivy (2022))

```

apt-get install wget apt-transport-https gnupg
wget -qO - https://aquasecurity.github.io/trivy-repo/deb/public.key |
  apt-key add -
echo deb https://aquasecurity.github.io/trivy-repo/deb bionic main | tee
  -a /etc/apt/sources.list.d/trivy.list
apt-get update
apt-get install -y trivy

```

- Next step in buildspec is to login to ECR repository

```

aws ecr get-login-password --region us-east-1 | docker login --username
  AWS --password-stdin 436026820972.dkr.ecr.us-east-1.amazonaws.com

```

- Now building the four Dockerfiles there will create the different application images.

```

docker build -t research-product -f productpage/Dockerfile .
docker build -t research-details -f details/Dockerfile
docker build -t research-ratings -f ratings/Dockerfile .
docker build -t research-reviews -f reviews/reviews-wlpcfg/Dockerfile .

```

- After building the Dockerfile I will scan the builded image and see the output and check the severity of dependency and library of the image.

```

trivy image research-product:latest
trivy image research-details:latest
trivy image research-ratings:latest
trivy image research-reviews:latest

```

823	Python (python-pkg)					
824	=====					
825	Total: 2 (UNKNOWN: 0, LOW: 0, MEDIUM: 0, HIGH: 0, CRITICAL: 2)					
826						
827	-----					
828	Library	Vulnerability	Severity	Installed Version	Fixed Version	Title
829	-----					
830	Werkzeug (METADATA)	CVE-2022-29361	CRITICAL	0.15.5	2.1.1	** DISPUTED ** Improper parsing of
831	HTTP requests in Pallets					Werkzeug v2.1.0...
832						https://avd.aquasec.com/nvd/cve-
833						

Figure 7: Trivy scanning of Image

- I will check the output of the scanned image if the file safe then I will push it to ECR repository. In this Code Pipeline all the steps are included in once but in the industry level first with Trivy Docker image should check once verified then only deployed otherwise make the changes suggested by the scan then push the image to ECR.
- Tag the docker image with a suitable name.

```
docker tag research-product:latest
436026820972.dkr.ecr.us-east-1.amazonaws.com/test:research-product
docker tag research-details:latest
436026820972.dkr.ecr.us-east-1.amazonaws.com/test:research-details
docker tag research-ratings:latest
436026820972.dkr.ecr.us-east-1.amazonaws.com/test:research-ratings
docker tag research-reviews:latest
436026820972.dkr.ecr.us-east-1.amazonaws.com/test:research-reviews
```

- After tagging the image pushing the image to ECR Repository.

```
docker push
436026820972.dkr.ecr.us-east-1.amazonaws.com/test:research-product
docker push
436026820972.dkr.ecr.us-east-1.amazonaws.com/test:research-details
docker push
436026820972.dkr.ecr.us-east-1.amazonaws.com/test:research-ratings
docker push
436026820972.dkr.ecr.us-east-1.amazonaws.com/test:research-reviews
```

- Next I need to login to the EKS cluster that I have created with the following commands.

```
export AWS_ACCESS_KEY_ID="$(echo ${CREDENTIALS} | jq -r
'.Credentials.AccessKeyId')"
export AWS_SECRET_ACCESS_KEY="$(echo ${CREDENTIALS} | jq -r
'.Credentials.SecretAccessKey')"
export AWS_SESSION_TOKEN="$(echo ${CREDENTIALS} | jq -r
'.Credentials.SessionToken')"
export AWS_EXPIRATION=$(echo ${CREDENTIALS} | jq -r
'.Credentials.Expiration')
aws eks update-kubeconfig --name research
kubectl config view --minify
```

- There can be an error of access denied while logging into EKS cluster because some policy is not added to the Code build IAM role. EKS describe role and update configmaps of kubernetes.
- I need to update the ECR image urls to deployment and gateway.yml file so that when I run kublet command to create pods the image I have scanned will spin up as container.
- In the deployment file I am creating four pods product, details, review, and ratings. In the gateway file, I are creating the ingress and gateway service of Istio which is used to secure pods and inter-service communication od pods.
- Now, run the command to deploy pods using the image mentioned in the deployment and gateway yml files.

```
kubectl apply -f deployment.yml -n research
kubectl apply -f gateway.yml -n research
```

5 Pods and Services in the Cluster

- After the deployment I can check the pods and services deployed in the cluster.

```
[prawal@]$ kubectl get pods -n research
NAME                                READY   STATUS    RESTARTS   AGE
details-v1-74dcf6fd88-dgm4j         2/2     Running   0           4d4h
productpage-v1-84b645657c-vpb9j     2/2     Running   0           4d4h
ratings-v1-5dd59fdb88-fzp8x        2/2     Running   0           4d4h
reviews-v1-86bd875db-7gpk9          2/2     Running   0           4d4h
reviews-v2-84999755b7-mqmdp         2/2     Running   0           4d4h
reviews-v3-775748b6dd-zbk5f        2/2     Running   0           4d4h
[prawal@]$
```

Figure 8: Pods Created in Research Namespace

```
[prawal@]$ kubectl get svc -n research
NAME            TYPE        CLUSTER-IP    EXTERNAL-IP   PORT(S)    AGE
details        ClusterIP   10.100.145.69 <none>        9080/TCP   4d5h
productpage    ClusterIP   10.100.224.82 <none>        9080/TCP   4d5h
ratings        ClusterIP   10.100.247.216 <none>        9080/TCP   4d5h
reviews        ClusterIP   10.100.174.210 <none>        9080/TCP   4d5h
[prawal@]$
```

Figure 9: Services Created in Research Namespace

```
[prawal@]$ kubectl get vs -n research
NAME            GATEWAYS          HOSTS          AGE
bookinfo        ["bookinfo-gateway"] ["*"]         4d19h
[prawal@]$
```

```
[prawal@]$ kubectl get gateway -n research
NAME            AGE
bookinfo-gateway 4d19h
[prawal@]$
```

Figure 10: Virtual Service and Gateway in Research Namespace

- If I describe the pod it can be seen Istio proxy is added to the pod.

```
istio-proxy:
  Container ID:   docker://6f9617db3546fdb9b19aa057974e0010a8c9224bf8f0cd1c8d261d21501d5590
  Image:          docker.io/istio/proxyv2:1.16.0
  Image ID:       docker-pullable://istio/proxyv2@sha256:f6f97fa4fb77a3cbe1e3eca0fa46bd462ad6b284c129cf57bf91575c4fb50cf9
  Port:           15090/TCP
  Host Port:      0/TCP
  AppArmor:       
```

Figure 11: Istio proxy on top of container

6 Dashboards and Metrics

- For Kiali and Grafana Dashboard I need to enable them to run the command.

```
istioctl dashboard kiali  
istioctl dashboard grafana
```

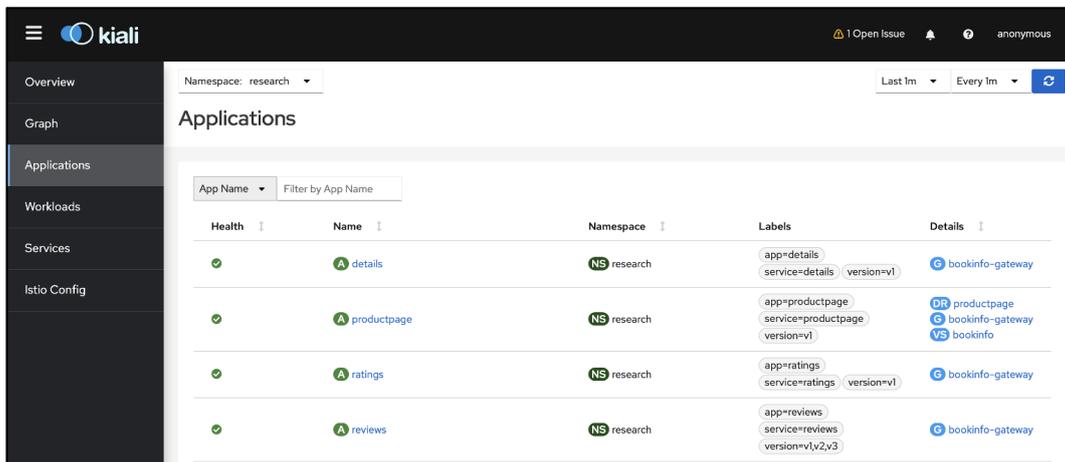


Figure 12: Kiali Dashboard

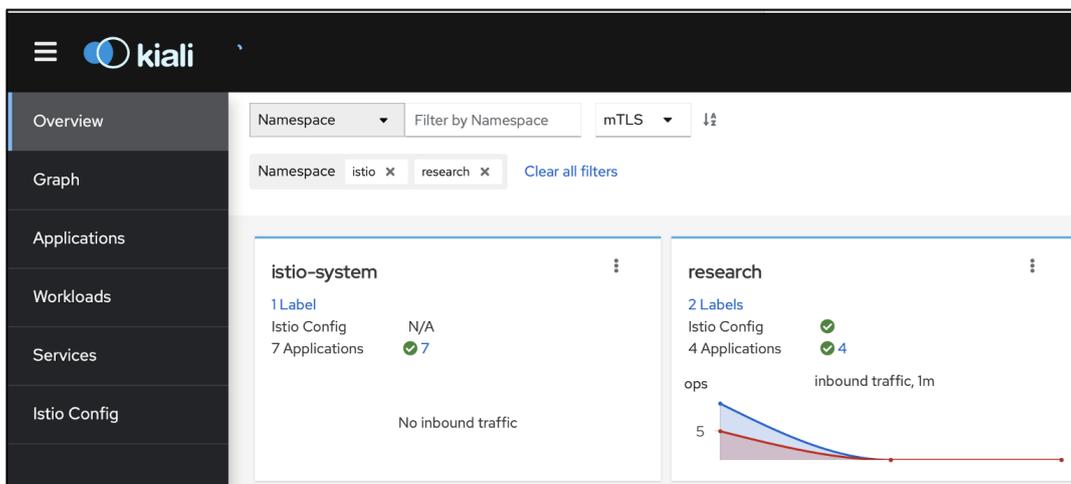


Figure 13: Kiali Dashboard Overview

- I can send traffic to the application to see different metrics.

```
for i in $(seq 1 100); do curl -s -o /dev/null  
"http://a63d50281fb894a5ab9309c12709b164-1903426299.us-east-1.elb.amazonaws  
.com/productpage"; done
```



Figure 14: CPU and Memory Usage

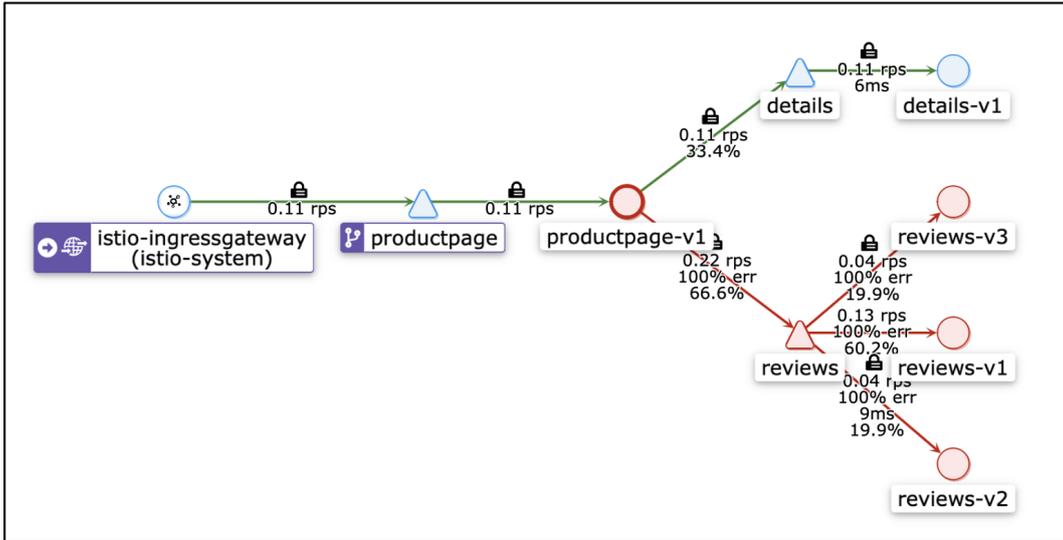


Figure 15: Request Flow in the cluster

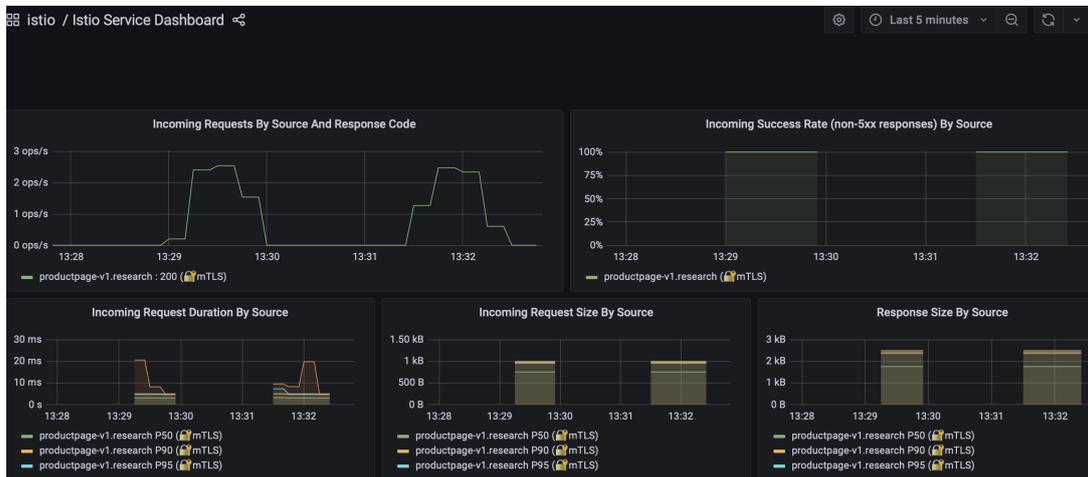


Figure 16: mTLS Connection Between Pods

References

Istio (2021). Istio/samples/bookinfo at master · istio/istio.

URL: <https://github.com/istio/istio/tree/master/samples/bookinfo>

istio (2022). Istio installation.

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

Kumar, P., Fanda, Daniel, Shriparv, S., Amey, James, Grodrigo, Muturi, N., Vignesh, Venkat and et al. (2022). How to install kubernetes (k8s) cluster on ubuntu 20.04.

URL: <https://www.linuxtechi.com/install-kubernetes-k8s-on-ubuntu-20-04/>

Paul SaveSoil, R. (2022). Creating an amazon eks cluster from scratch using eksctl.

URL: <https://dev.to/aws-builders/creating-an-amazon-eks-cluster-from-scratch-using-eksctl-540i>

Trivy (2022). Installation.

URL: <https://aquasecurity.github.io/trivy/v0.18.3/installation/>