# Introduction

## Purpose

This document provides the instructions to be followed for the installation for a software used for the PoC as well as the steps on the verification of the components developed for the PoC.

## Scope

# This document is intended for the team, in order to up to date, relevant and clear install documents.

# Installation Manual

In order to facilitate the team tasks, it is vital to provide the delivery with accurate, exhaustive, complete and clear installation instructions.

## Pre-requisites

Following are the list of install prerequisites that must be fulfilled before the install can begin.

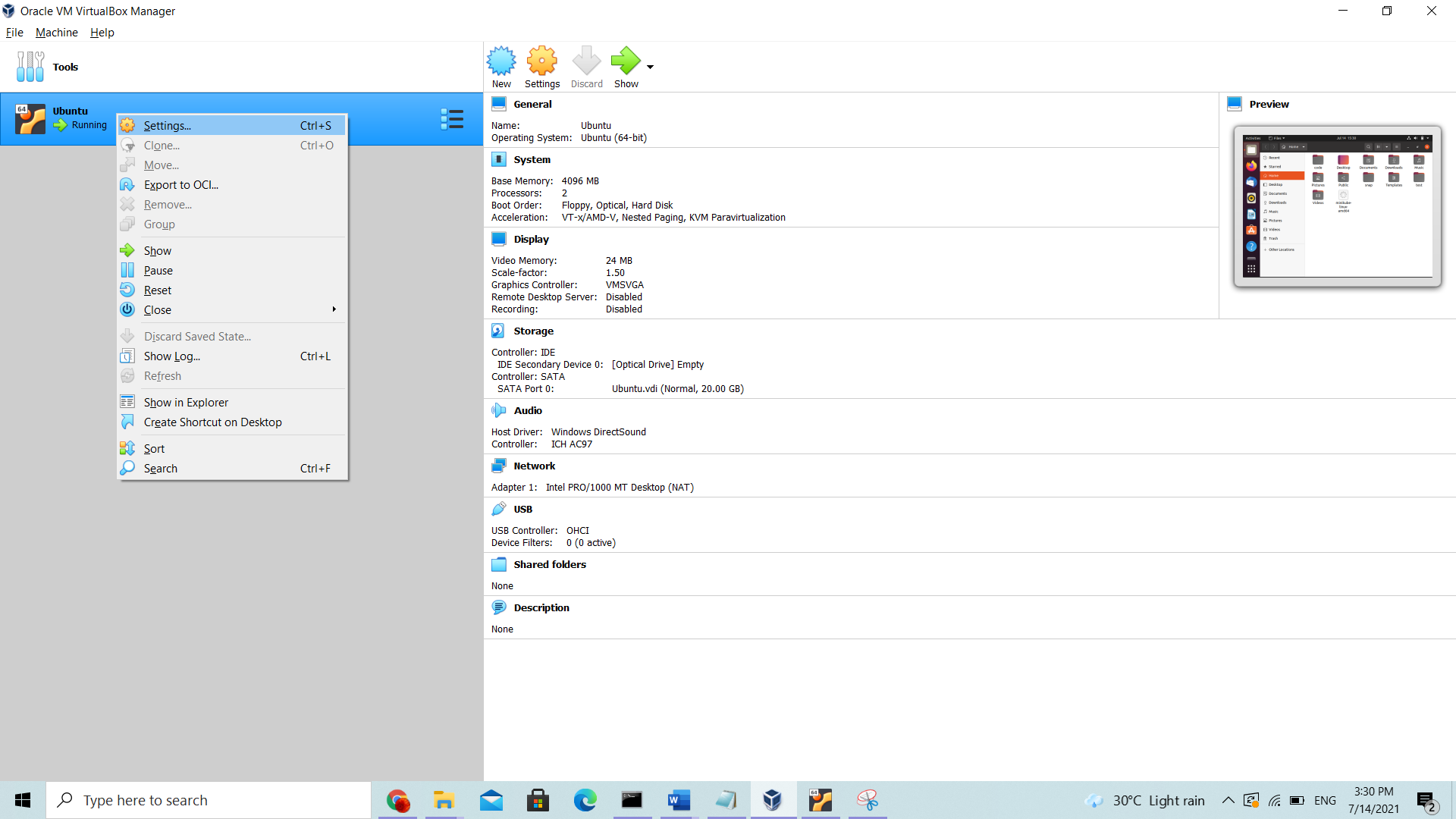
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S No | Type | Name with Version | Purpose | Path to download |
| 1 | Operating System | Windows 7/10 | Infrastructure for Virtual environment |  |
| 2 | Virtual Box | Oracle Virtual Box 6.1 | Infrastructure to virtualize, where Open Source software like Ubuntu can be installed | Download [Link](https://dl5.filehippo.com/48b/14e/9623941a95aa2dde0fabab26cb1885a88d/VirtualBox-6.1.18-142142-Win.exe?Expires=1621390018&Signature=5012eb7b87043fa053e10ae91ae1916cc5df4a09&url=https://filehippo.com/download_virtualbox/&Filename=VirtualBox-6.1.18-142142-Win.exe)  Alternate Path  <https://www.virtualbox.org/wiki/Downloads> |
| 3 | Operating System | Ubuntu 20.04.2 | Infrastructure where the PoC will run. | Download [Link](https://mirrors.piconets.webwerks.in/ubuntu-mirror/ubuntu-releases/20.04.2.0/ubuntu-20.04.2.0-desktop-amd64.iso)  Alternate Path  <https://ubuntu.com/download/desktop> |
| 4 |  |  |  |  |

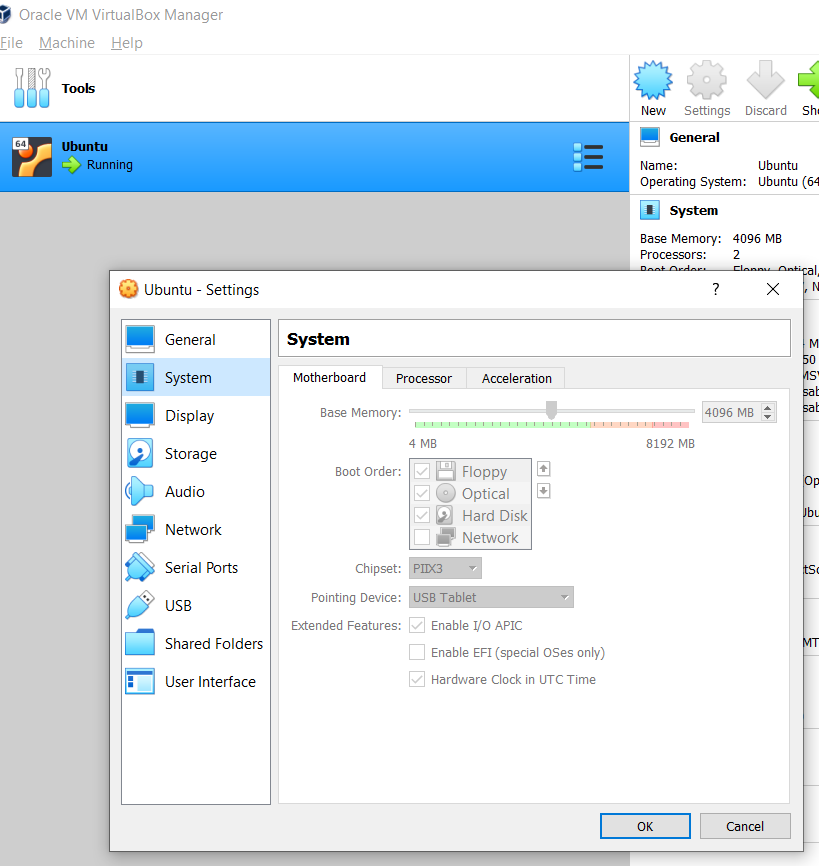
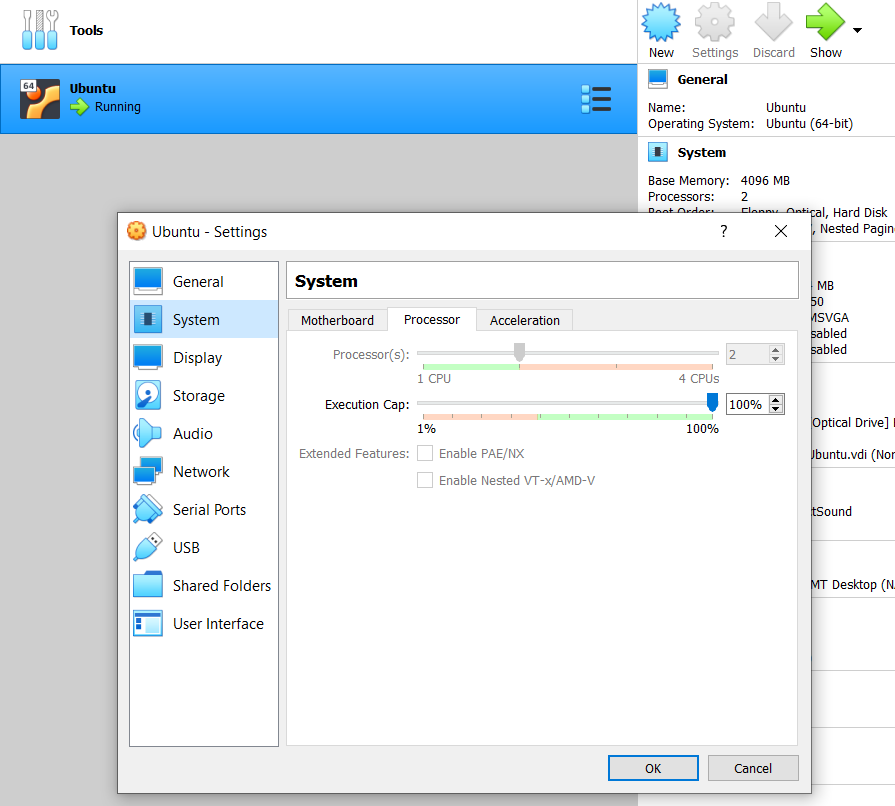
Follow the instructions from [here](https://itsfoss.com/install-linux-in-virtualbox/) to install pre-requisites without which we may not able to run the PoC.

**Note** : It is advisable to have the good configuration of Windows Machine so that the Operating System can be configured to run in Virtual Box without latency.

### Configuration of Virtual Machine

Optimal configuration of the Virtual Machine is 4096MB Base Memory, 2 Processor(s) shown below



Important : It is assumed that the Windows is Operating System used as Infrastructure for the evaluation of PoC. If you use the Mac or any Linux machines, the Pre-requisite can be skipped.

## Software Install procedure

Following software are required in the Ubuntu environment

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| S No | Name with Version | Usage | Path to download | Validate installation command line |
| 1 | Git 2.25.1 | To checkout code from Git. By default Ubuntu provides it and no installation is required |  | git version |
| 2 | Visual Studio Code | IDE for code | Download [link](https://az764295.vo.msecnd.net/stable/2aeda6b18e13c4f4f9edf6667158a6b8d408874b/code_1.58.1-1626158276_amd64.deb)  Alternate path  <https://code.visualstudio.com/download> |  |
| 3 | Docker | Used  for building and containerizing applications | Instruction available at [link](https://docs.docker.com/engine/install/ubuntu/#install-using-the-convenience-script)  Alternate path  <https://docs.docker.com/engine/install/ubuntu/> | docker version |
| 4 | Docker-compose | Tool for defining and running multi-container Docker applications | Instruction available at [link](https://docs.docker.com/compose/install/#install-compose-on-linux-systems)  Alternate path  <https://docs.docker.com/compose/install/#install-compose> | docker-compose version |
| 5 | Minikube and Kubernates | Minikube is open source tool that allows you to set up a single-node Kubernetes cluster. | Instruction available at [link](https://phoenixnap.com/kb/install-minikube-on-ubuntu) | minikube version  kubectl version -o json |
| 6 | Istio | Configure Service mesh with various parameters like traffic management, security, and policy | Instruction available at [link](https://istio.io/latest/docs/setup/getting-started/#download)  Alternate path  <https://istio.io/latest/docs/setup/getting-started/> | istioctl x precheck |

## Software accounts

We need to have account in the [docker hub](https://hub.docker.com/) for pulling or pushing images. Hence create account for yourself [here](https://hub.docker.com/).

Docker Hub image repository Path

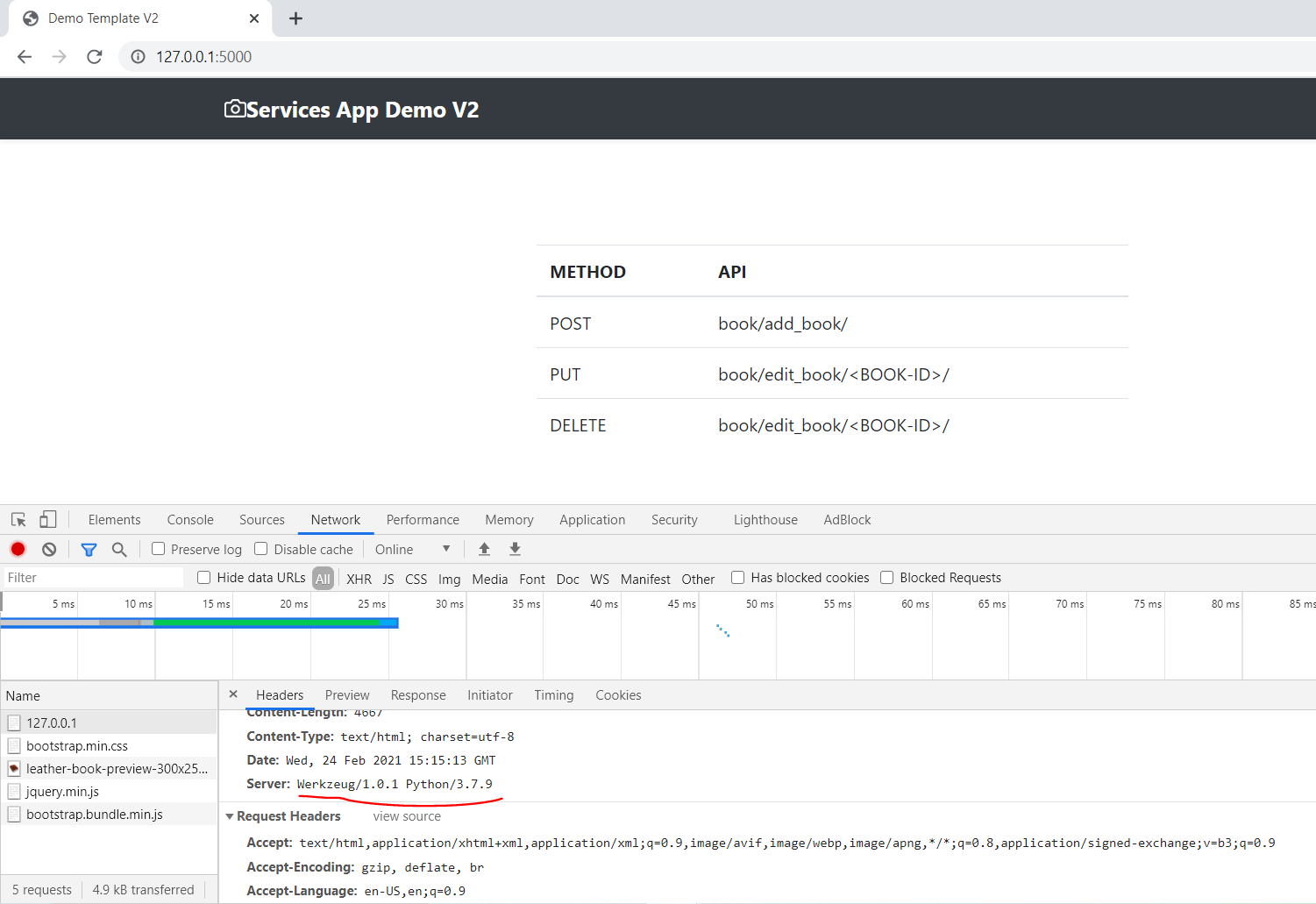
<https://hub.docker.com/r/ksrangar/services/tags?page=1&ordering=last_updated>

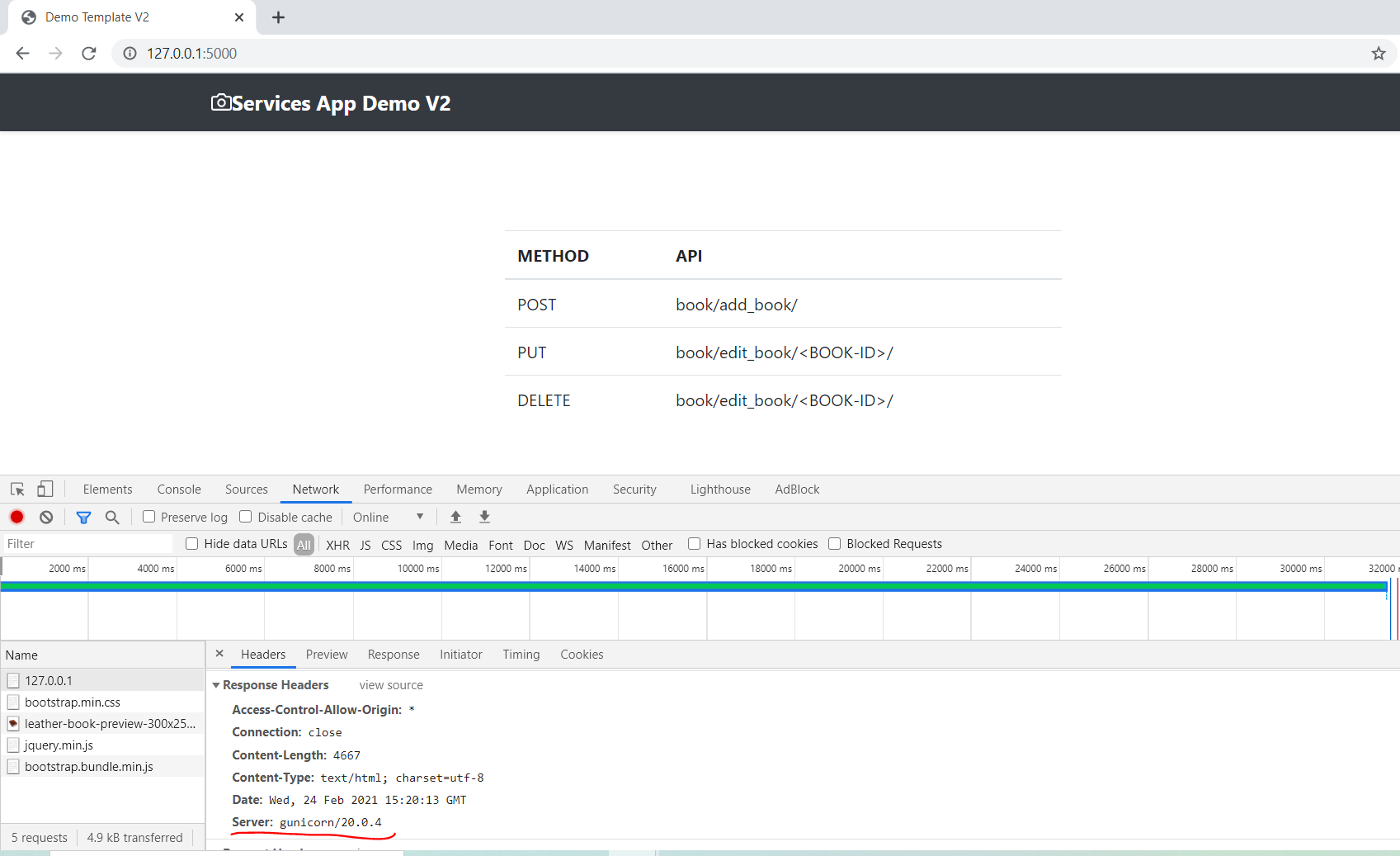
Source code Path

<https://github.com/ksrangarajan/service_mesh_poc>

**Service Mesh POC**

**1: Default Inbuild Flask Server:**



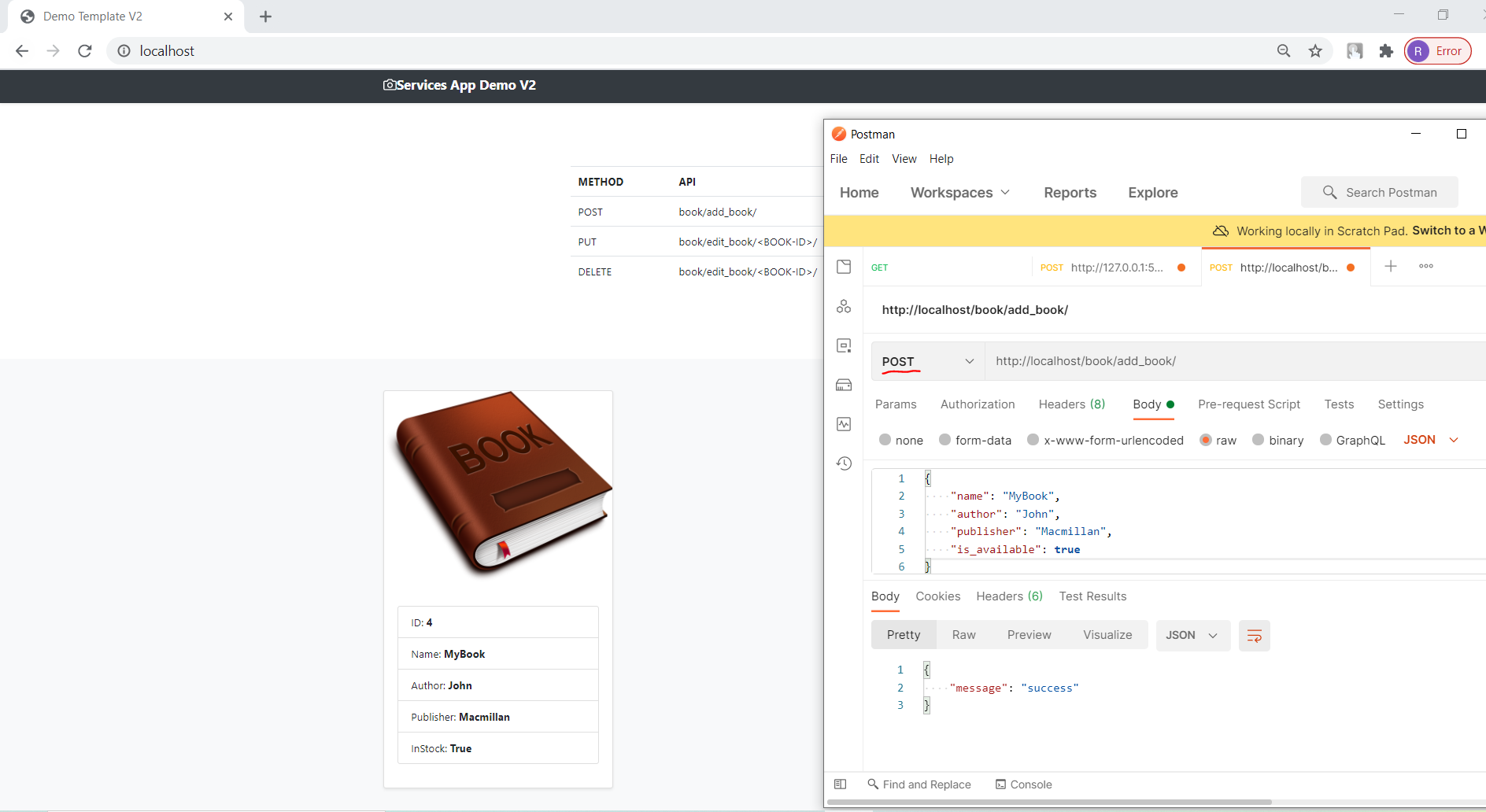
**2: Gunicorn Server:** 

**3: Nginx Server on Port 80:**

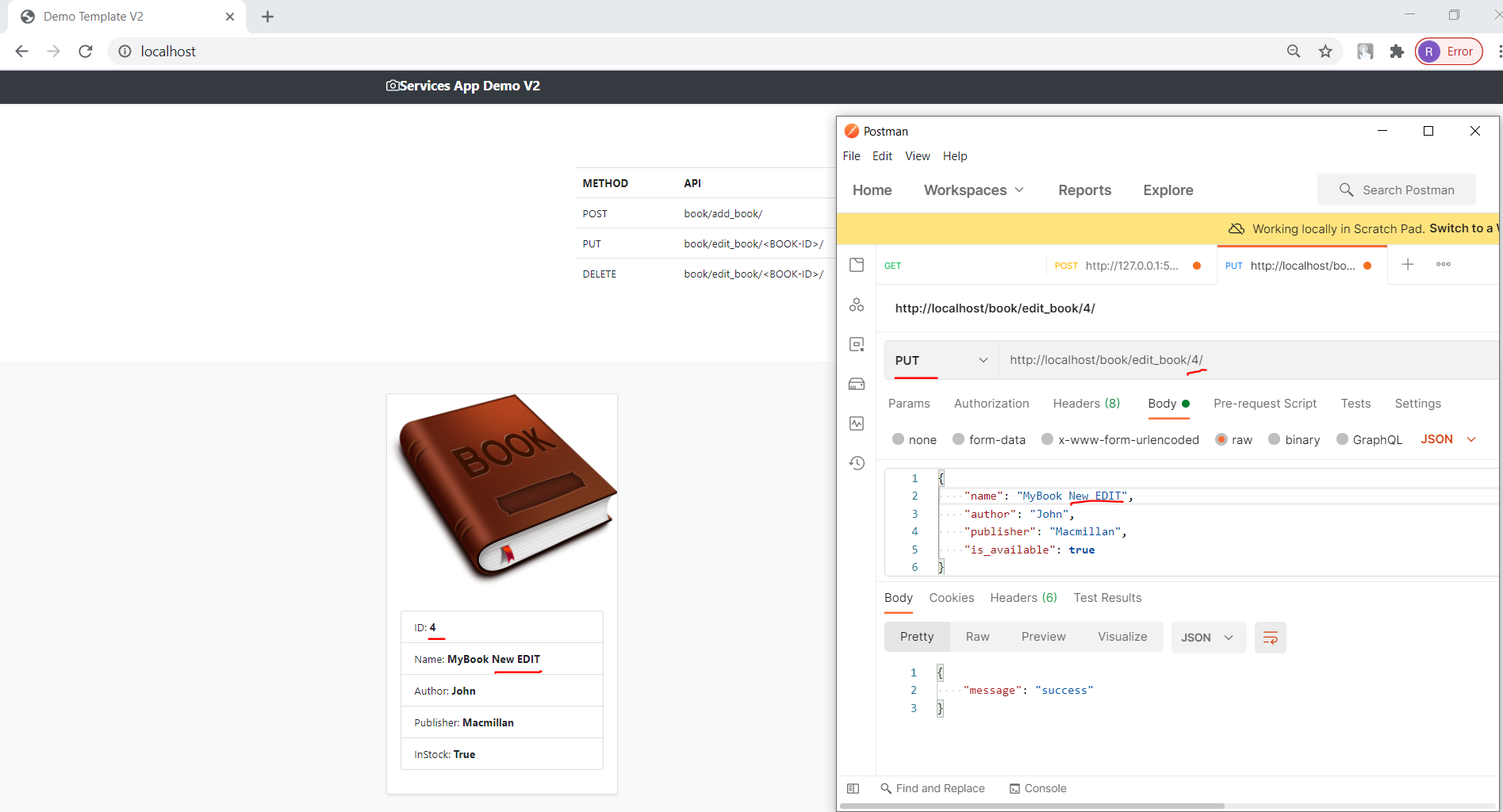


**REST Operations:**

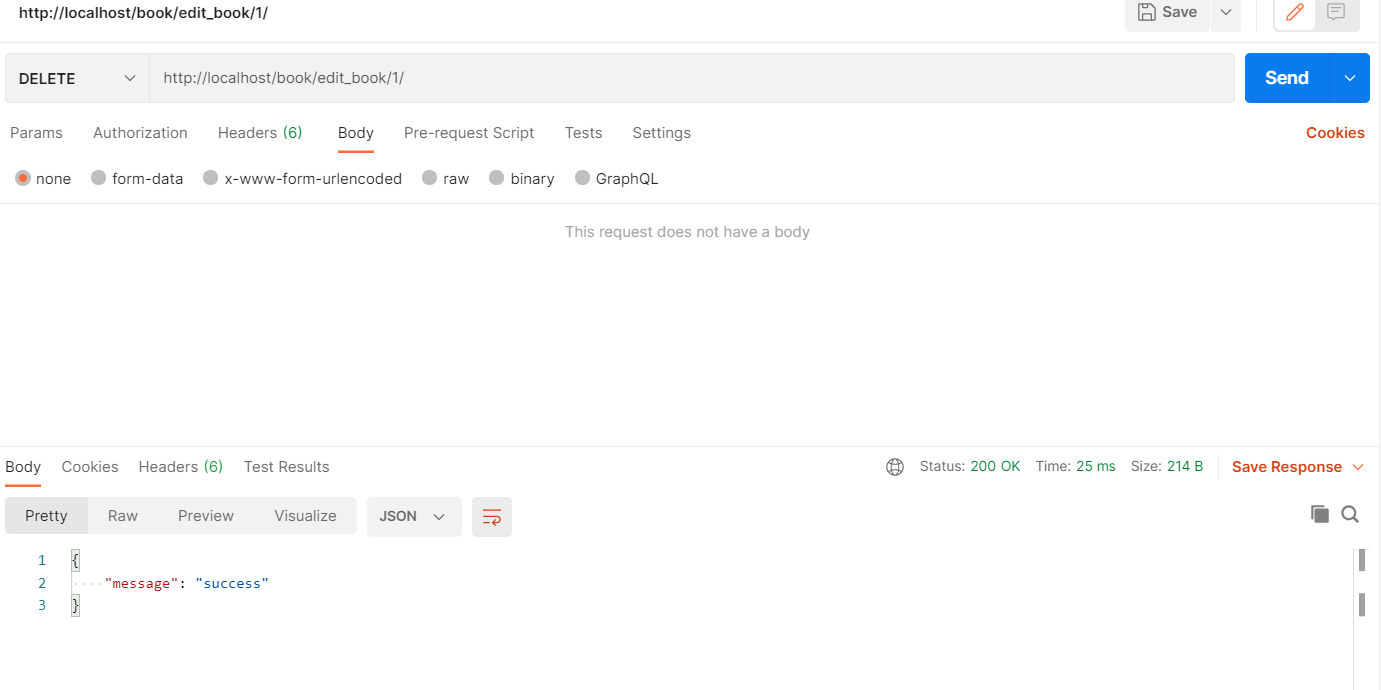
**1: POST:**

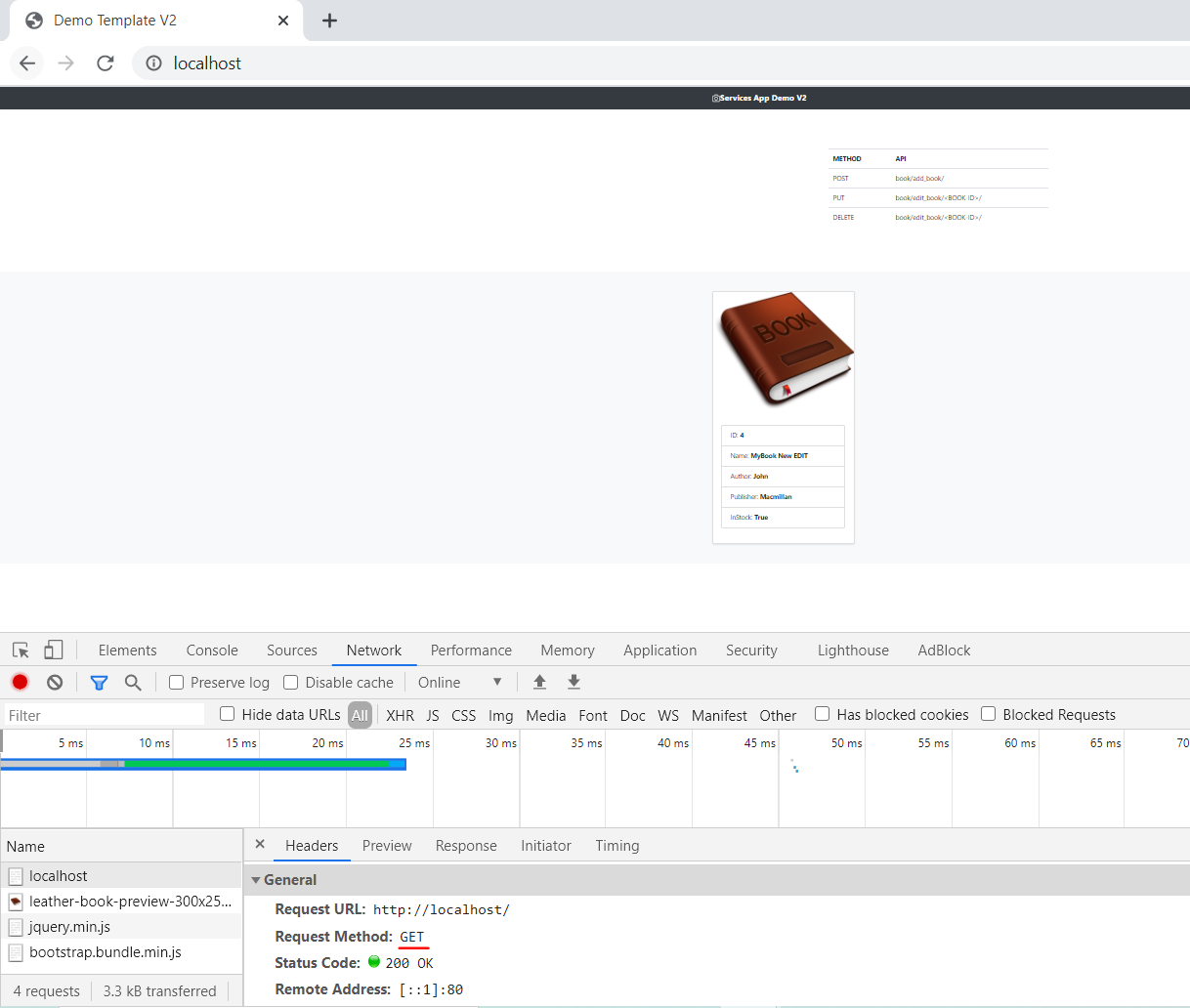


**2: PUT:**



**3: Delete:**



**4: GET (**Default method on UI home page**):**

**Nginx**

**1: Nginx as LoadBalancer:**

upstream services\_manager {

    server backend\_v1:5000;

    server backend\_v2:5001;

}

server {

    listen 80;

    location / {

        proxy\_pass http://services\_manager;

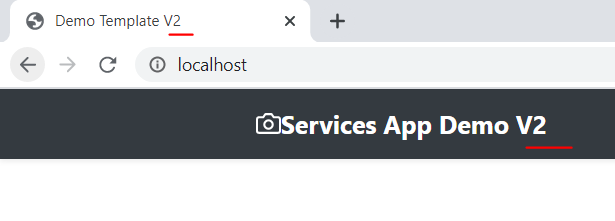
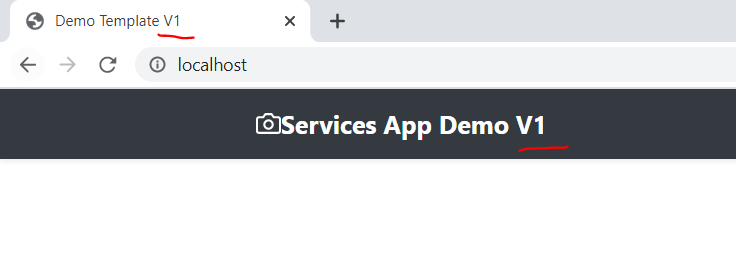
        proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

        proxy\_set\_header Host $host;

        proxy\_redirect off;

    }

}

* ***Default Waterfall Model***:
  + 

**2: Nginx as API-GateWay:**

upstream add\_book {

    server backend\_v1:5000;

}

upstream edit\_book {

    server backend\_v2:5001;

}

# LoadBalancer

upstream services\_manager {

    server backend\_v1:5000;

    server backend\_v2:5001;

}

server {

    listen 80;

    location / {

        proxy\_pass http://services\_manager;

        proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

        proxy\_set\_header Host $host;

        proxy\_redirect off;

    }

    # Add Book TO DB

    location /book/add\_book/ {

        proxy\_pass http://add\_book;

    }

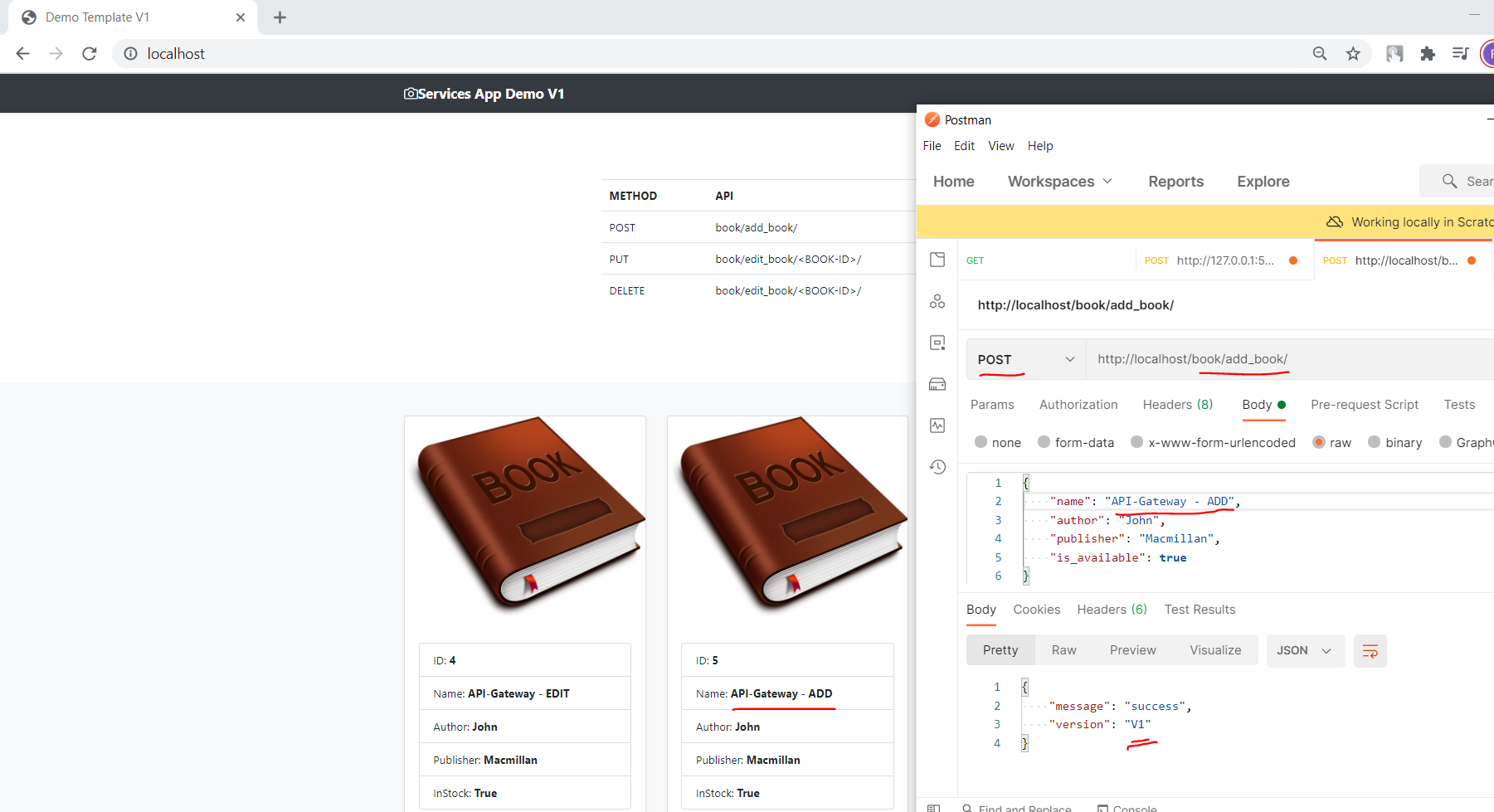
    # Edit & Delete book from DB

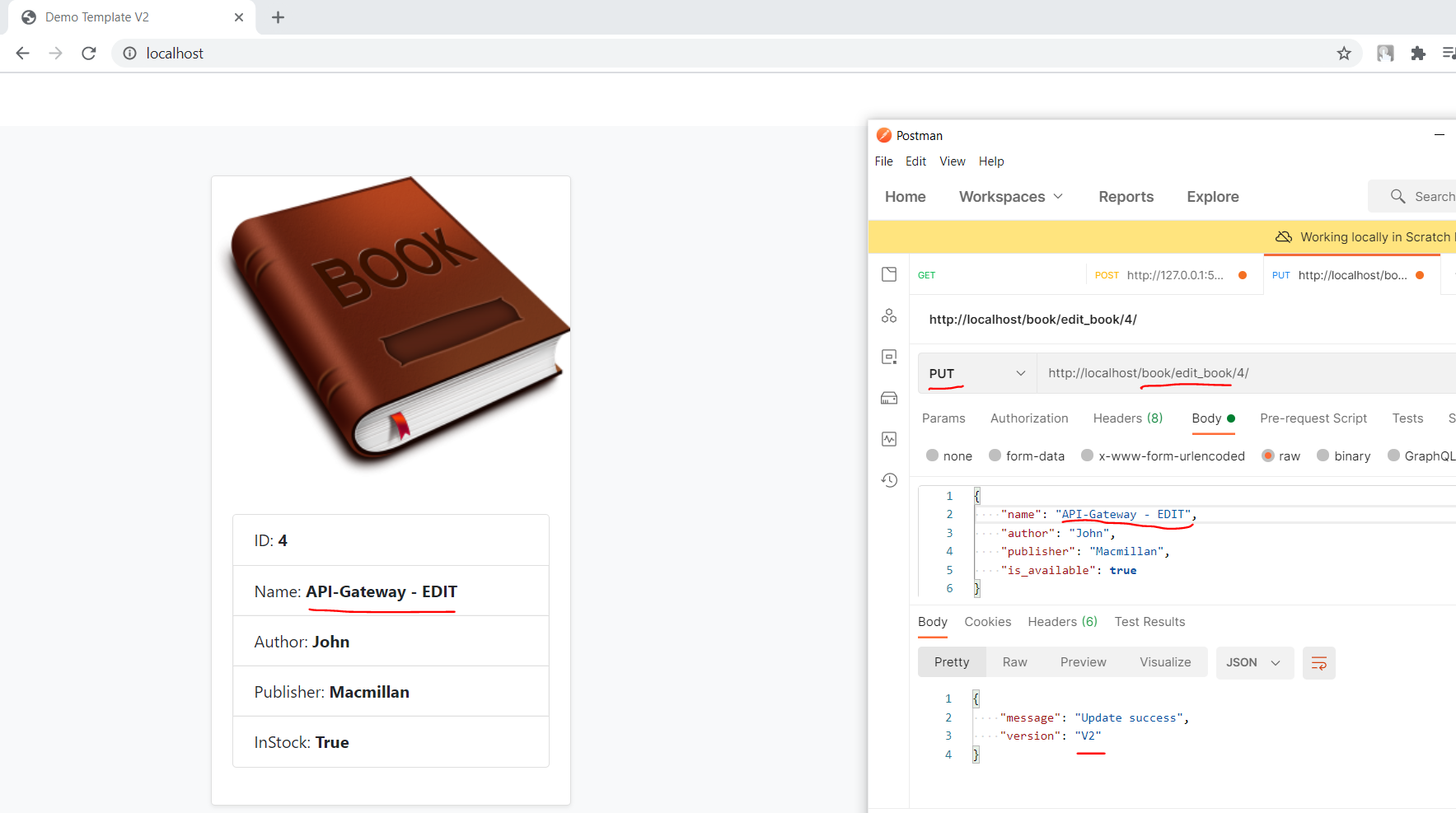
    location /book/edit\_book/ {

        proxy\_pass http://edit\_book;

    }

}

**ADD Record (POST):**

**EDIT Record (PUT):**

**3: Nginx as Webserver:**

upstream services\_manager {

    server backend:5000;

}

server {

    listen 80;

    location / {

        proxy\_pass http://services\_manager;

        proxy\_set\_header X-Forwarded-For $proxy\_add\_x\_forwarded\_for;

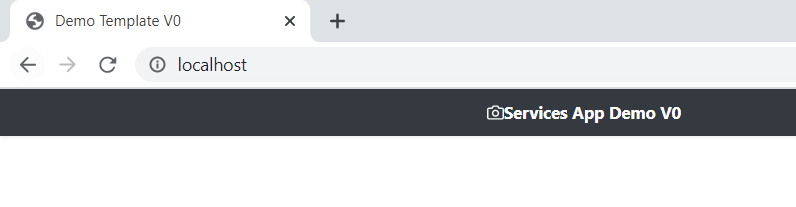
        proxy\_set\_header Host $host;

        proxy\_redirect off;

    }

}

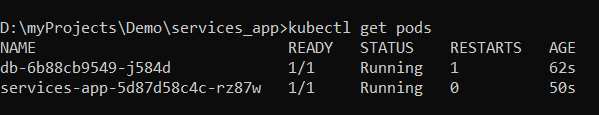
***Single Instance***



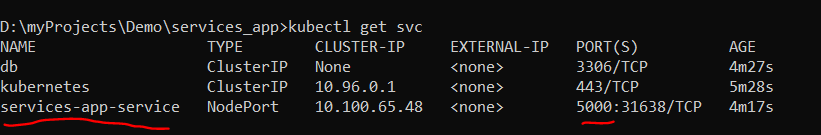
Note: Demo on Docker

[**Kubernetes**](https://kubernetes.io/)

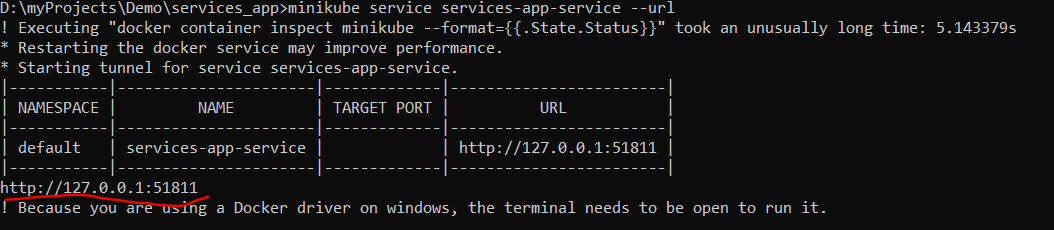
***1: Create Application and Database pods in K8s cluster***



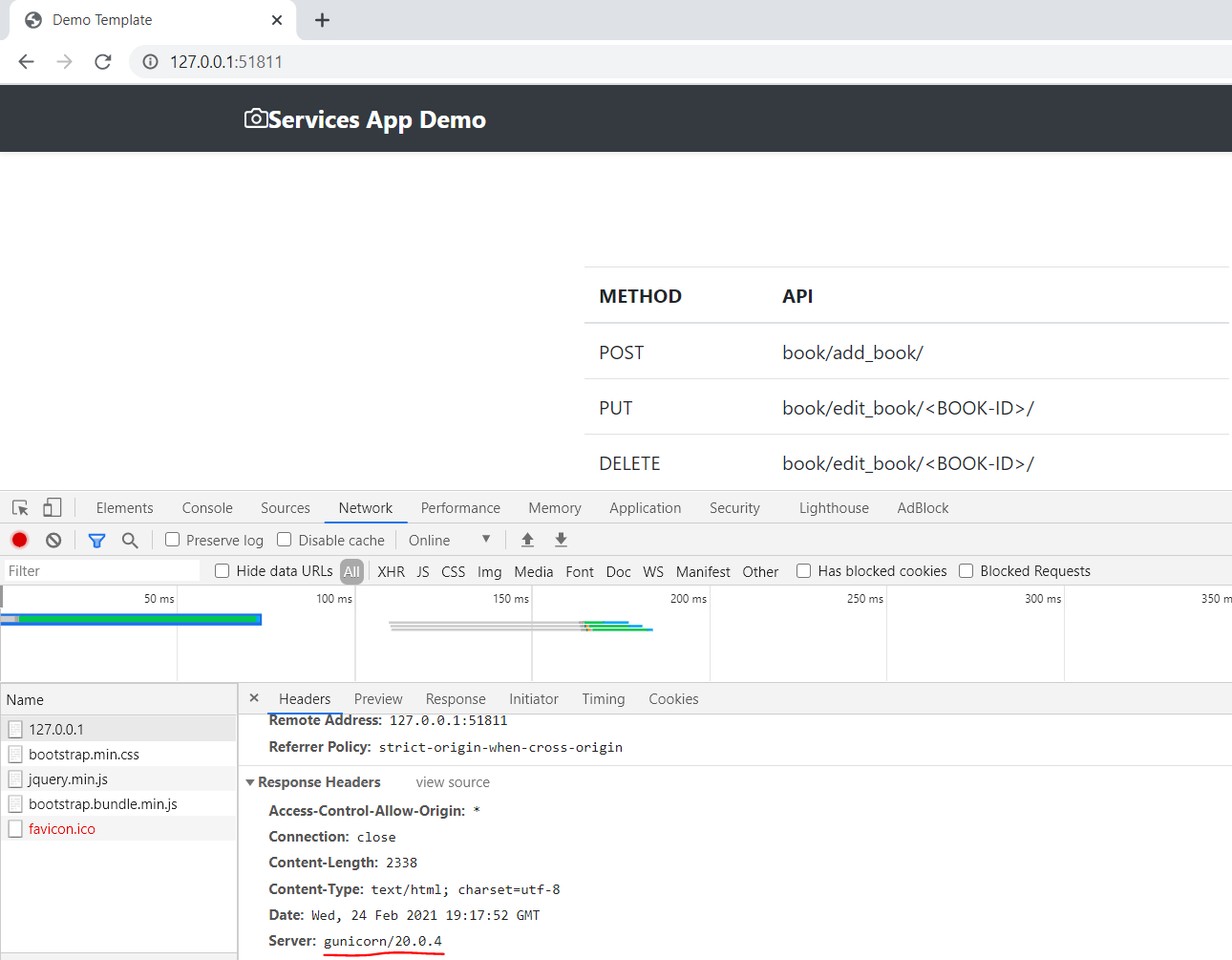
***2: Get Running Service for Application and Database***



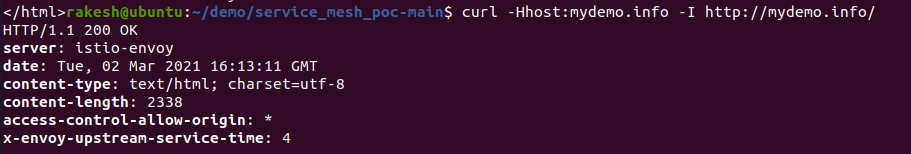
***3: Access Application inside K8s Cluster***



***Demo:***



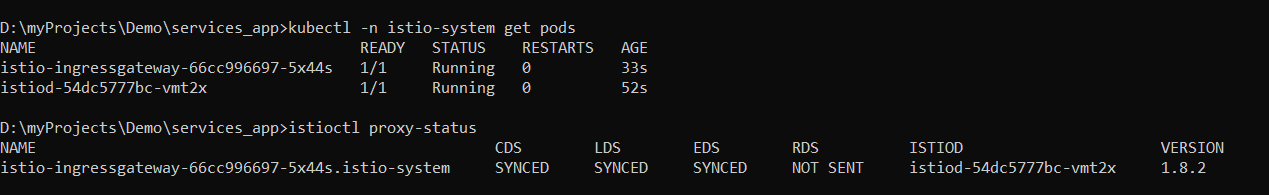
***4: Ingress controller:***



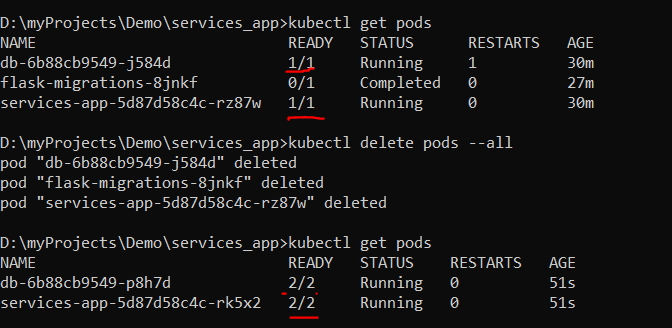
***Note: Demo on Minikub, Ubuntue***

[**Istio**](https://istio.io/latest/) **(Service Mesh):**

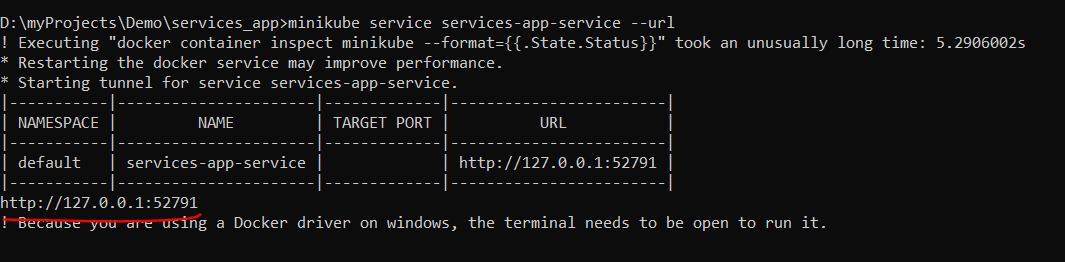
***1: Istio Pods and Service:***

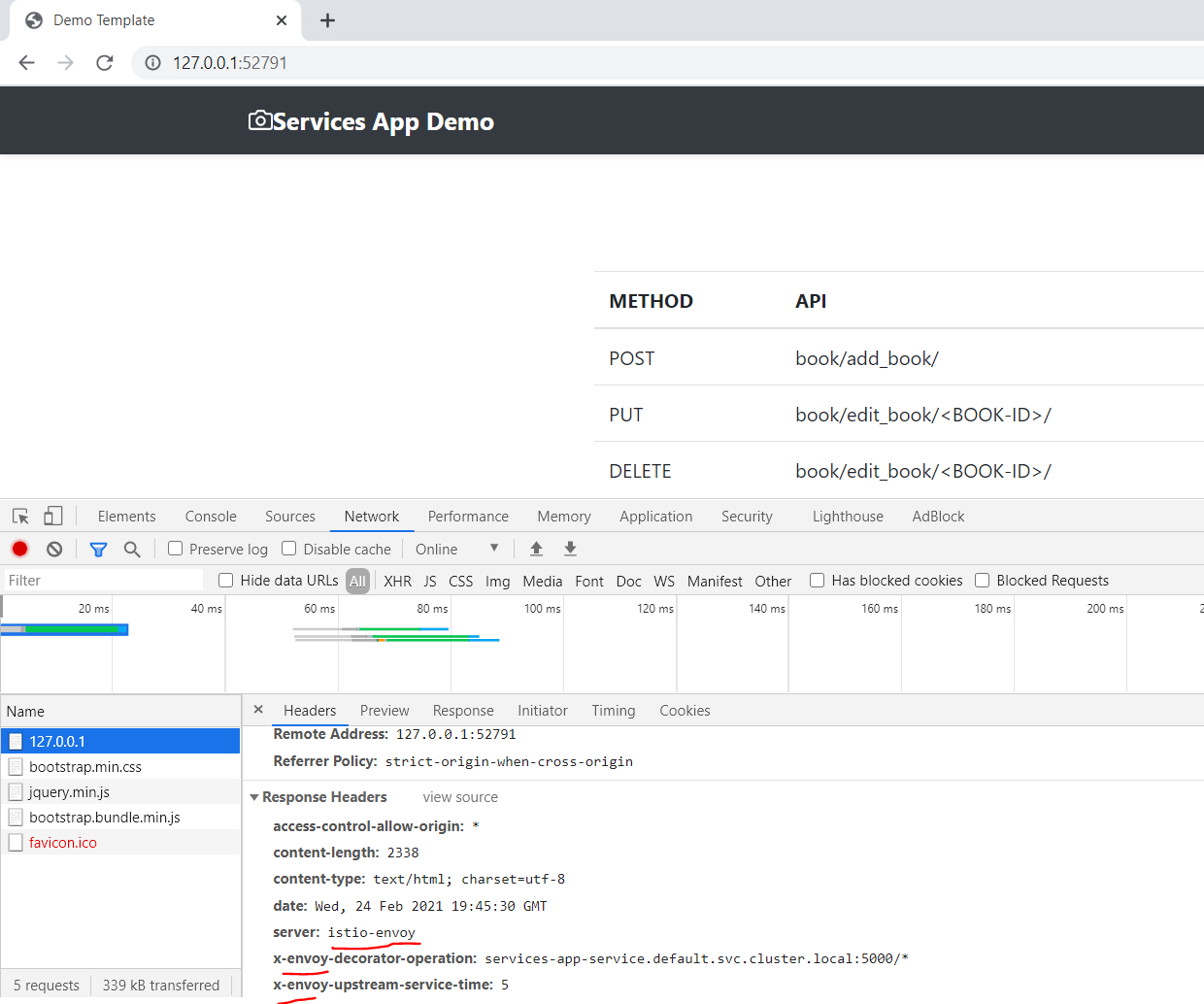


***2: Inject Istio side car (Envoy) to Application and DB pods.***

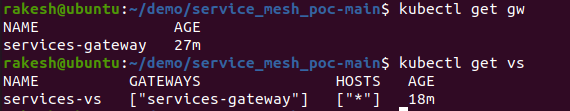


***3: Access Application inside K8s Cluster via Istio-envoy:***



***Demo:***

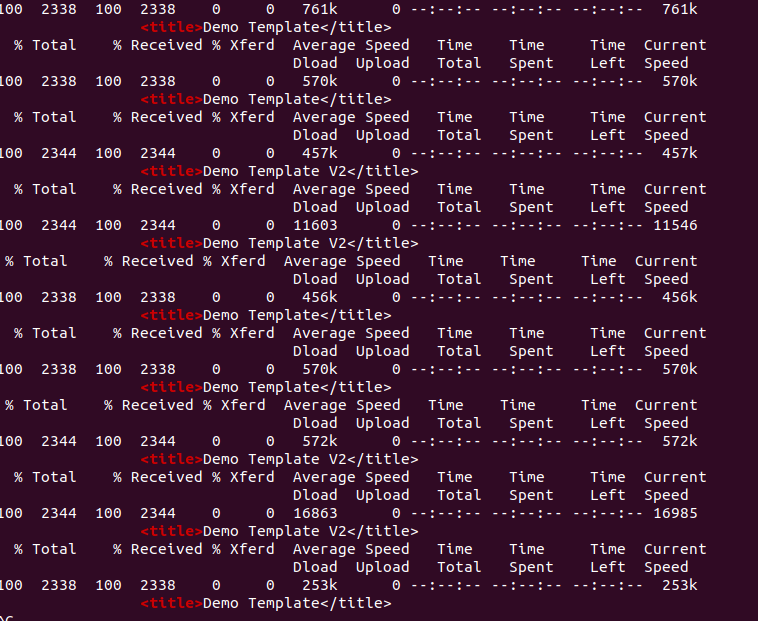
***4: Istio Gateway & VirtualService:***



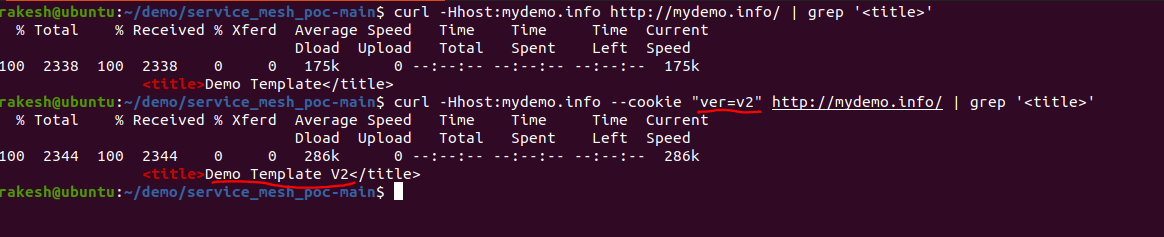
***5: Simple snippet to load test request and response***



***5:*** [***Traffic Split***](https://istio.io/latest/docs/tasks/traffic-management/traffic-shifting/) ***(50%(Demo Template)-50%(Demo Template V2)):***



***6: Canary deployment (If Cookie is set to v2 then redirect to V2 Service)***



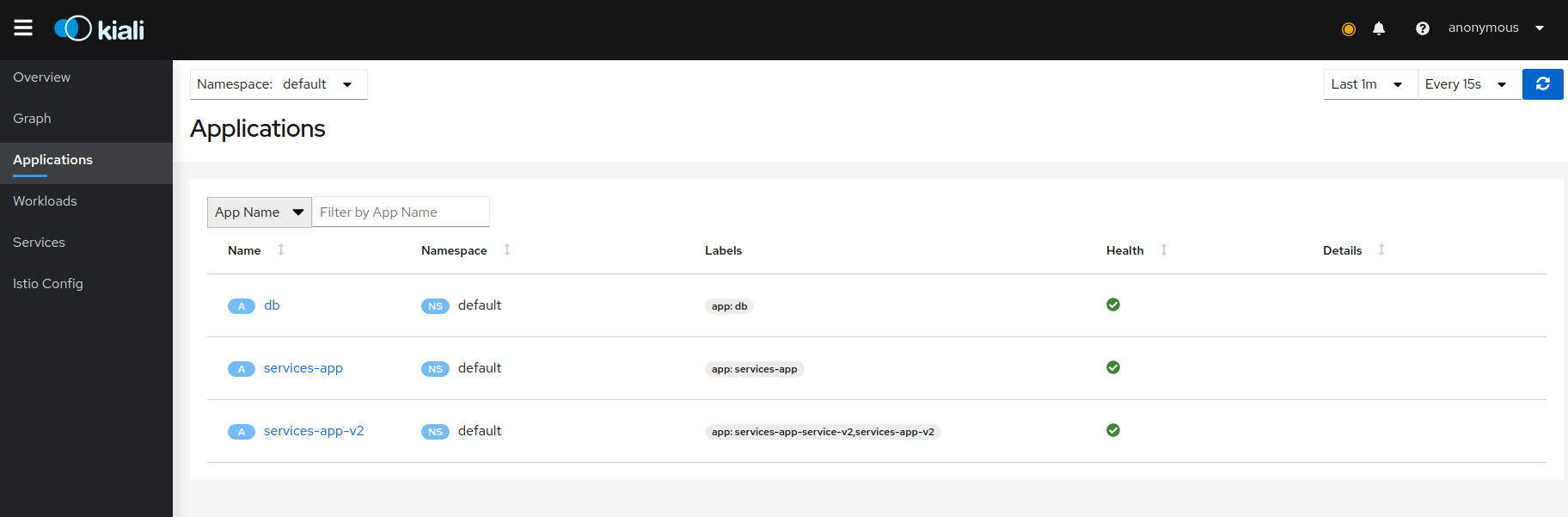
***TODO***

1. [***Explore other options***](https://istio.io/latest/docs/tasks/traffic-management/)

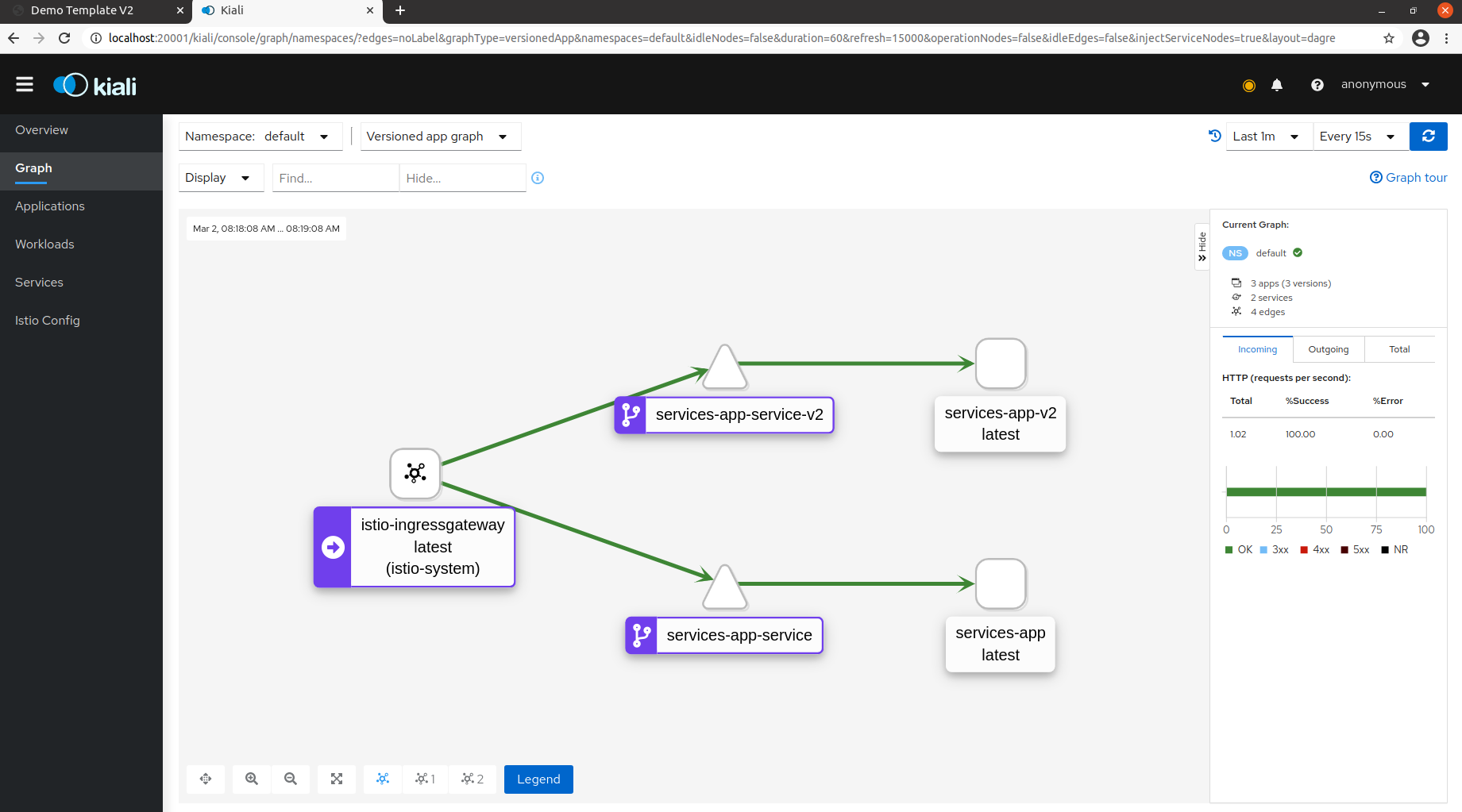
***Note: Demo on Minikube, Ubuntu***

[**Monitoring &Visualization**](https://istio.io/latest/docs/tasks/observability/)

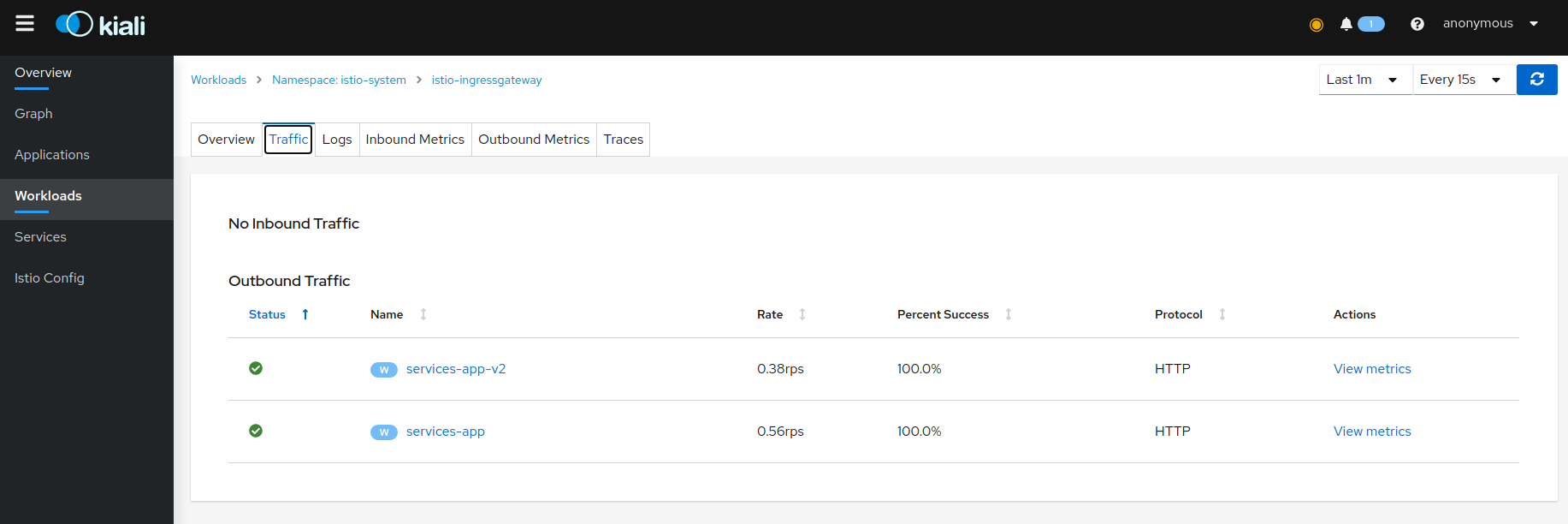
***1:*** [***Kiali***](https://kiali.io/) ***(Visualization Tool) 🡪 Real Time***



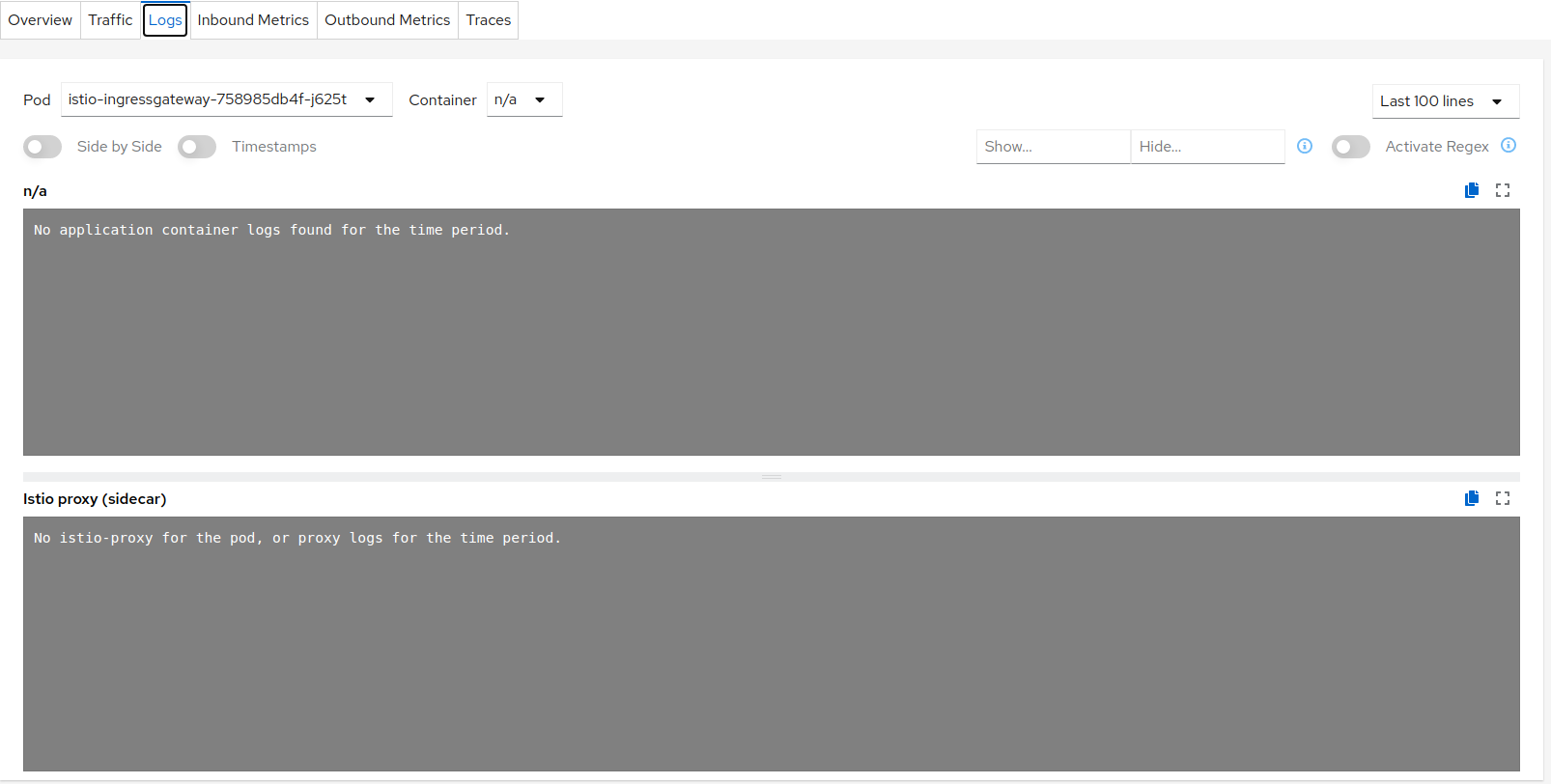
***Graph of request direction 50% to Service V1 and 50% to Service V2***



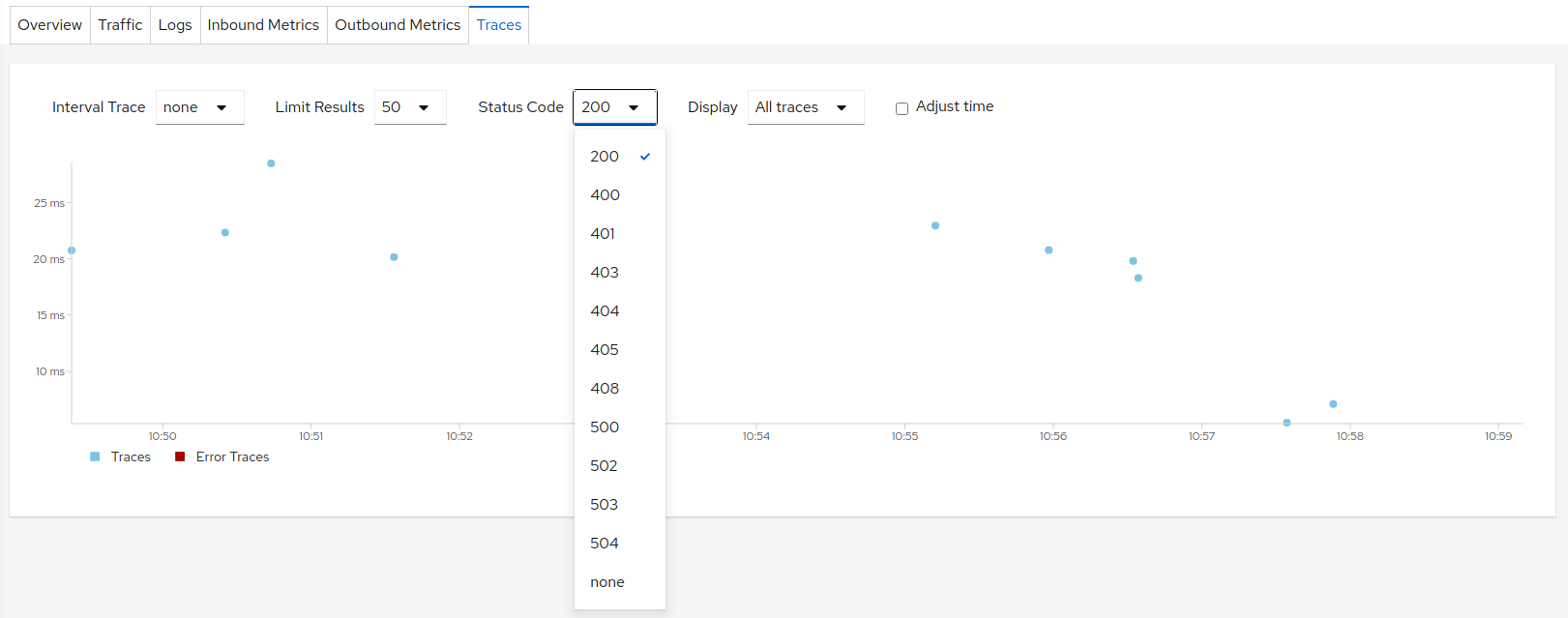
***Service Health***



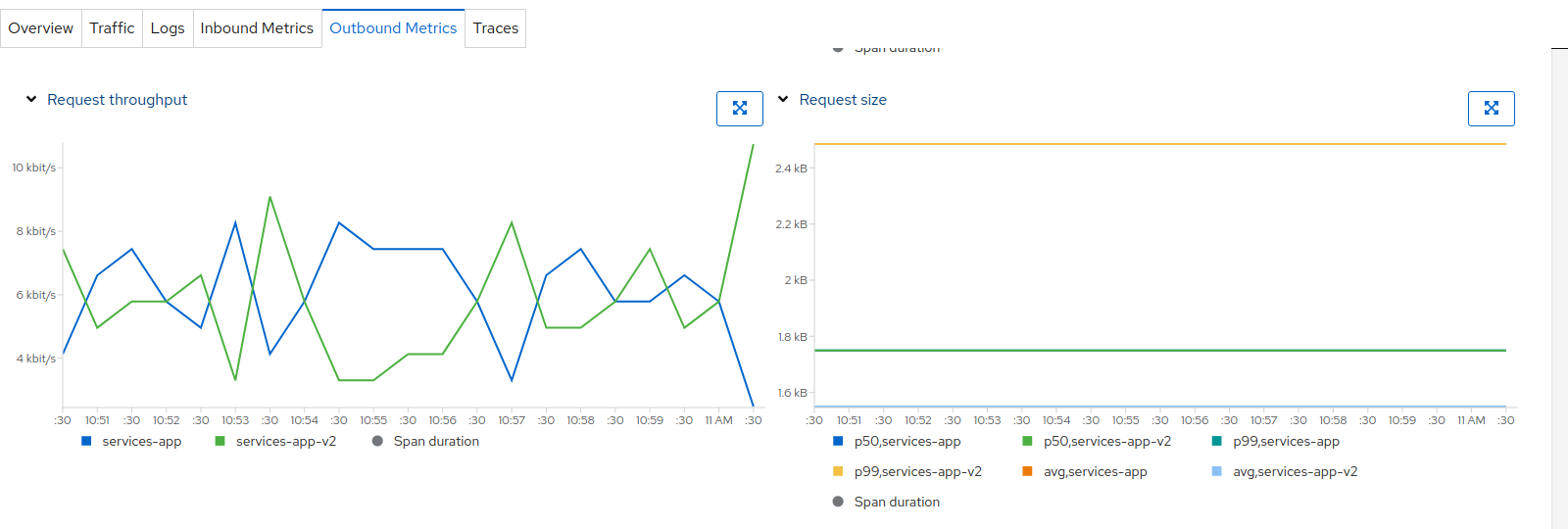
***Logs From Pods (Containers) (Not Configured Currently -TODO)***

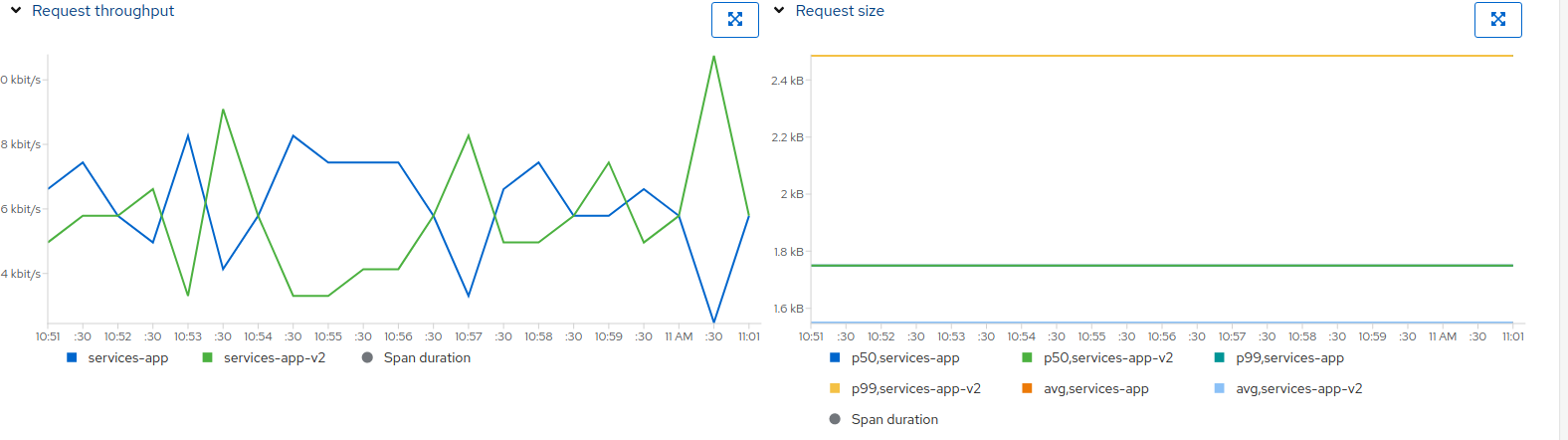


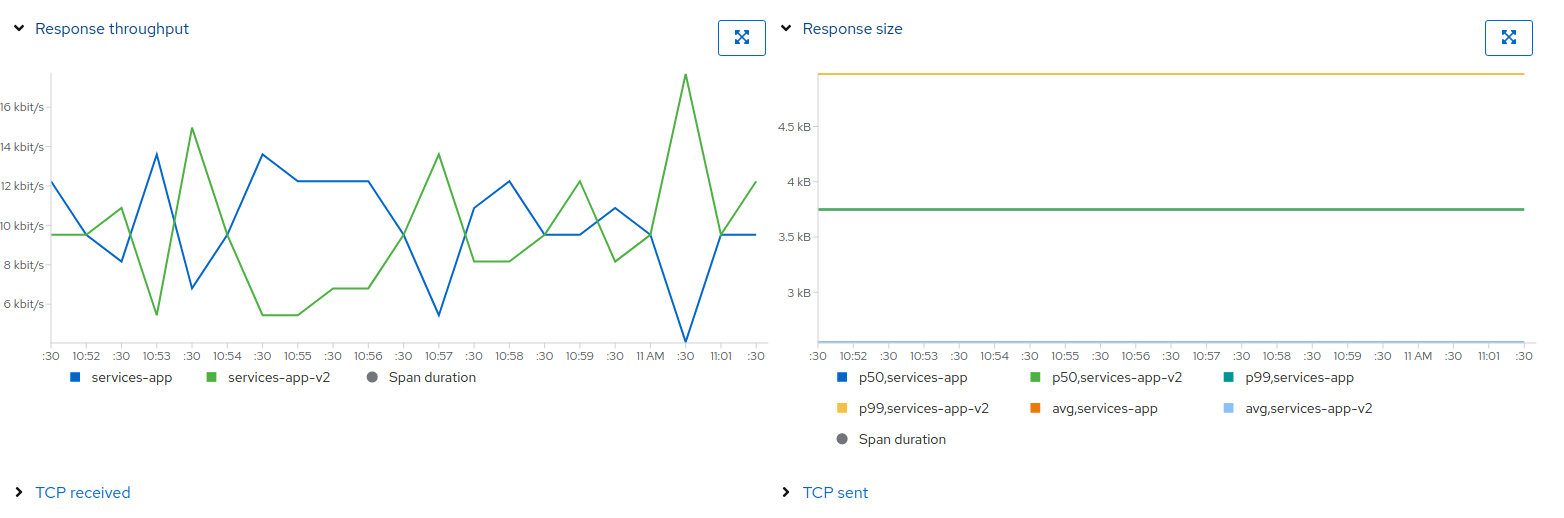
***Request Tracing (status codes and more)***



***Outbound Metrics:***



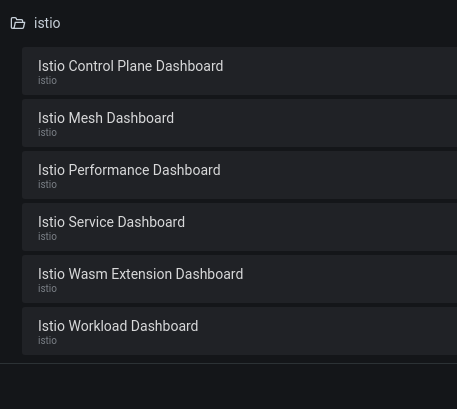




***Note: Demo on Minikub, Ubuntu***

***2:*** [***Grafana***](https://grafana.com/) ***(Visualization Tool) 🡪 Real Time:***

***List of available metrics and real time monitoring and visualization***

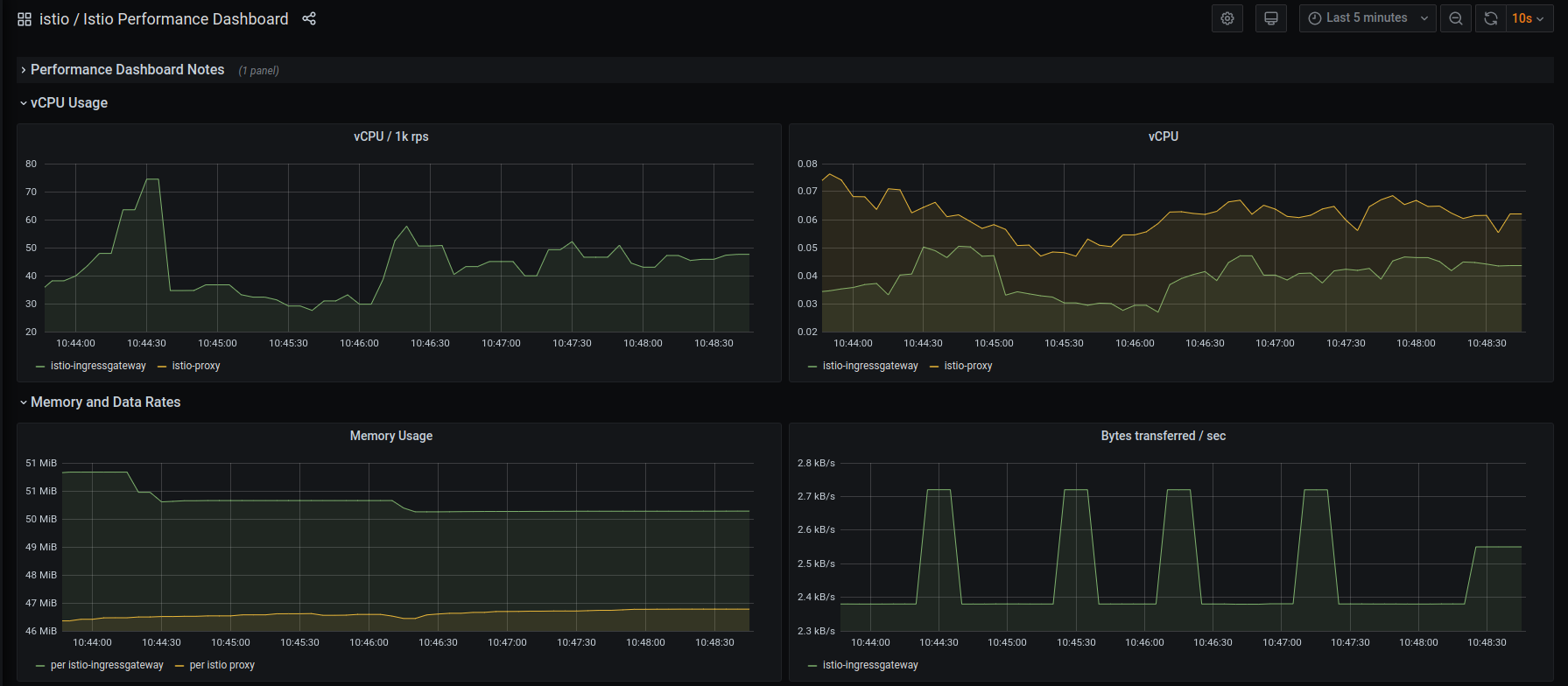


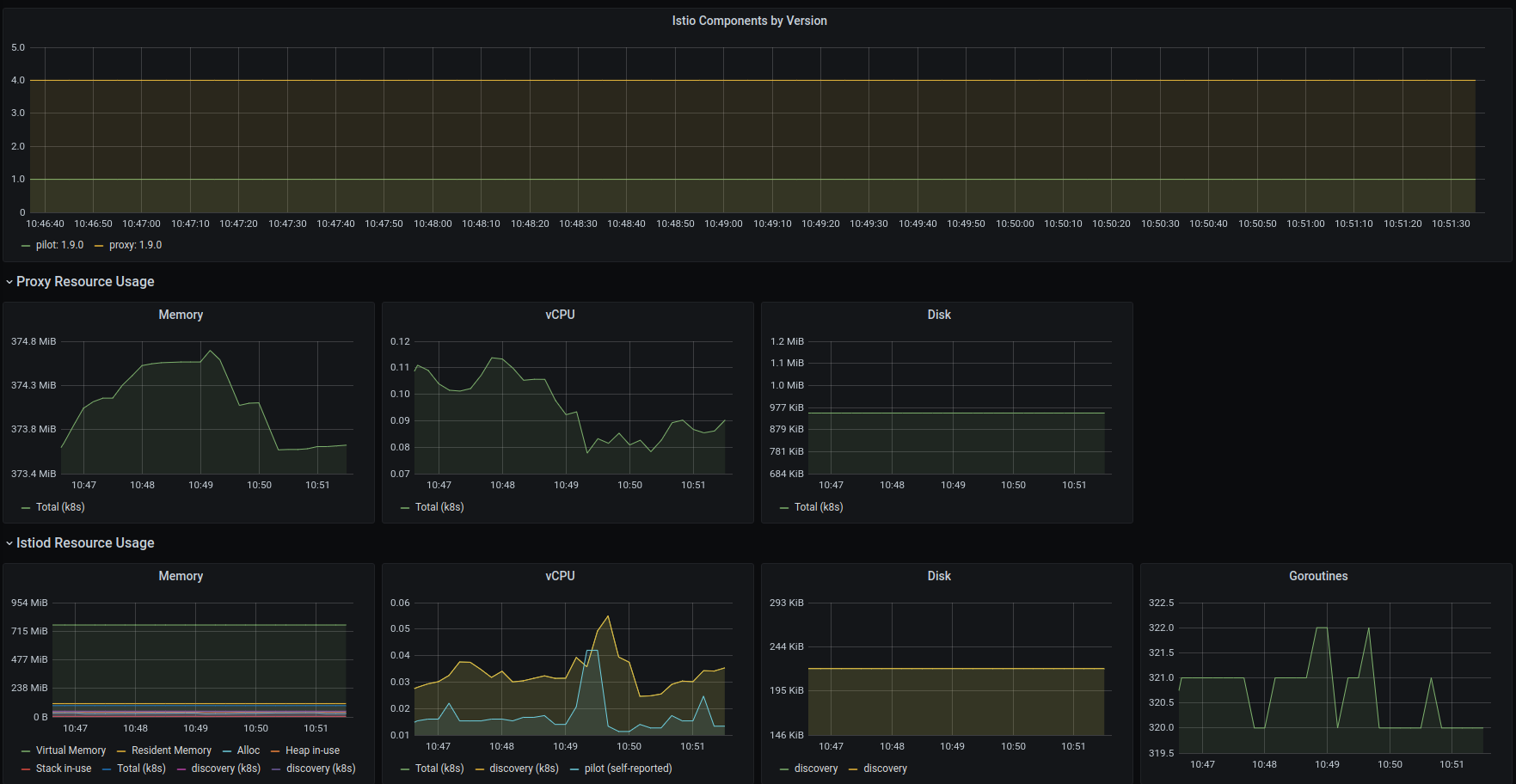
***Control Plane Dashboard***



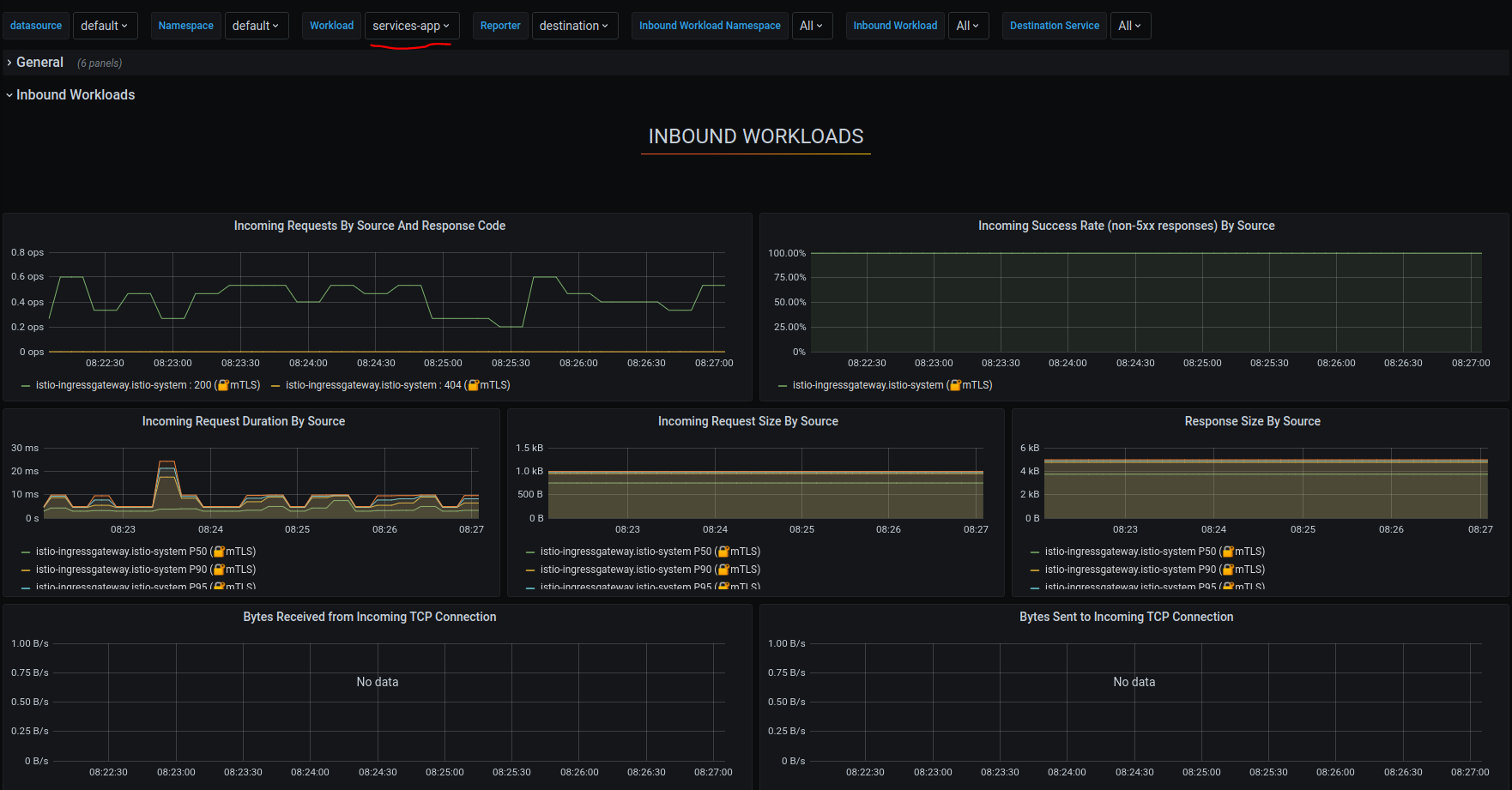


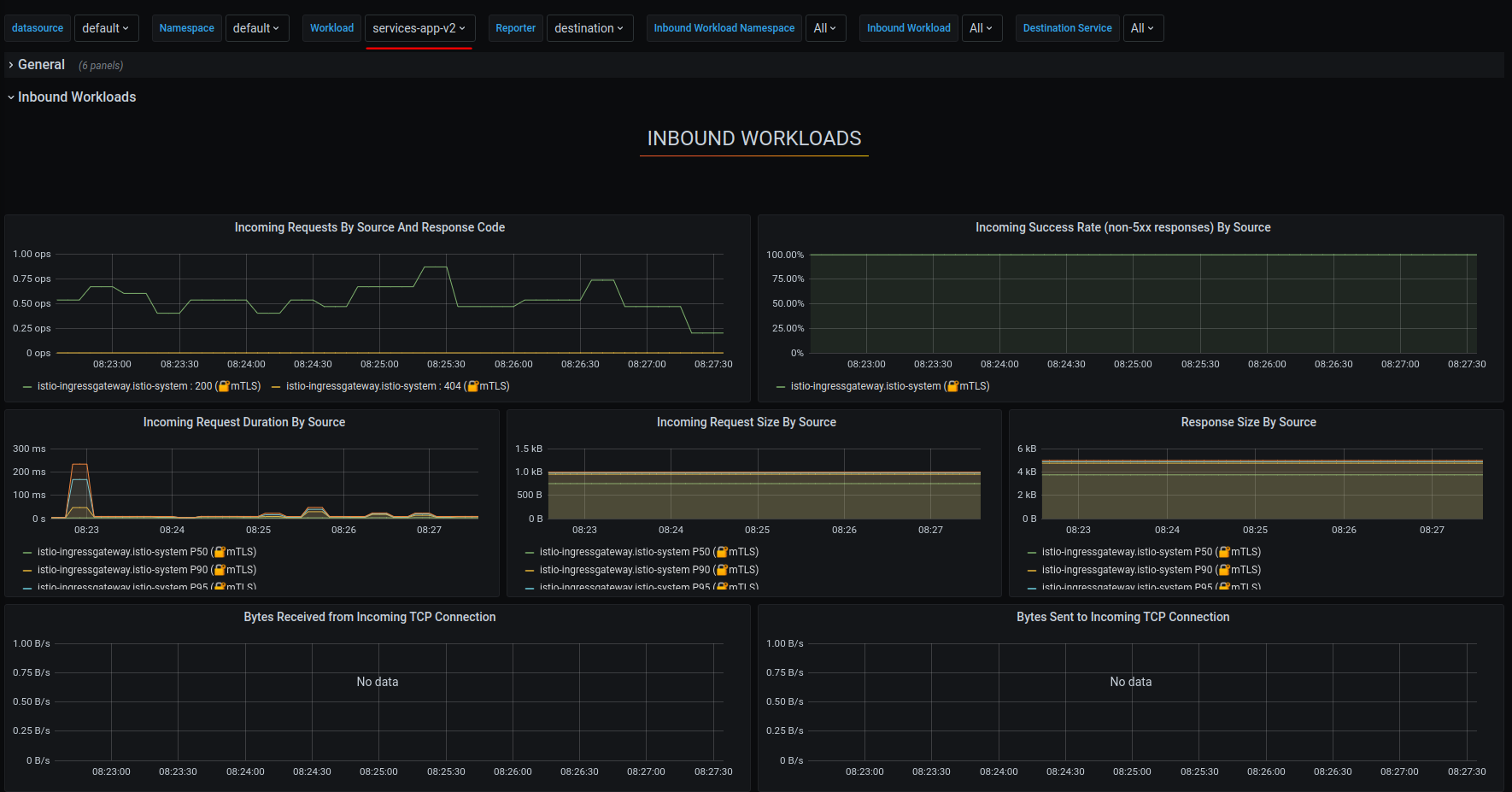
***Performance Dashboard (Metrics for CPU, Memory, IO utilization)***

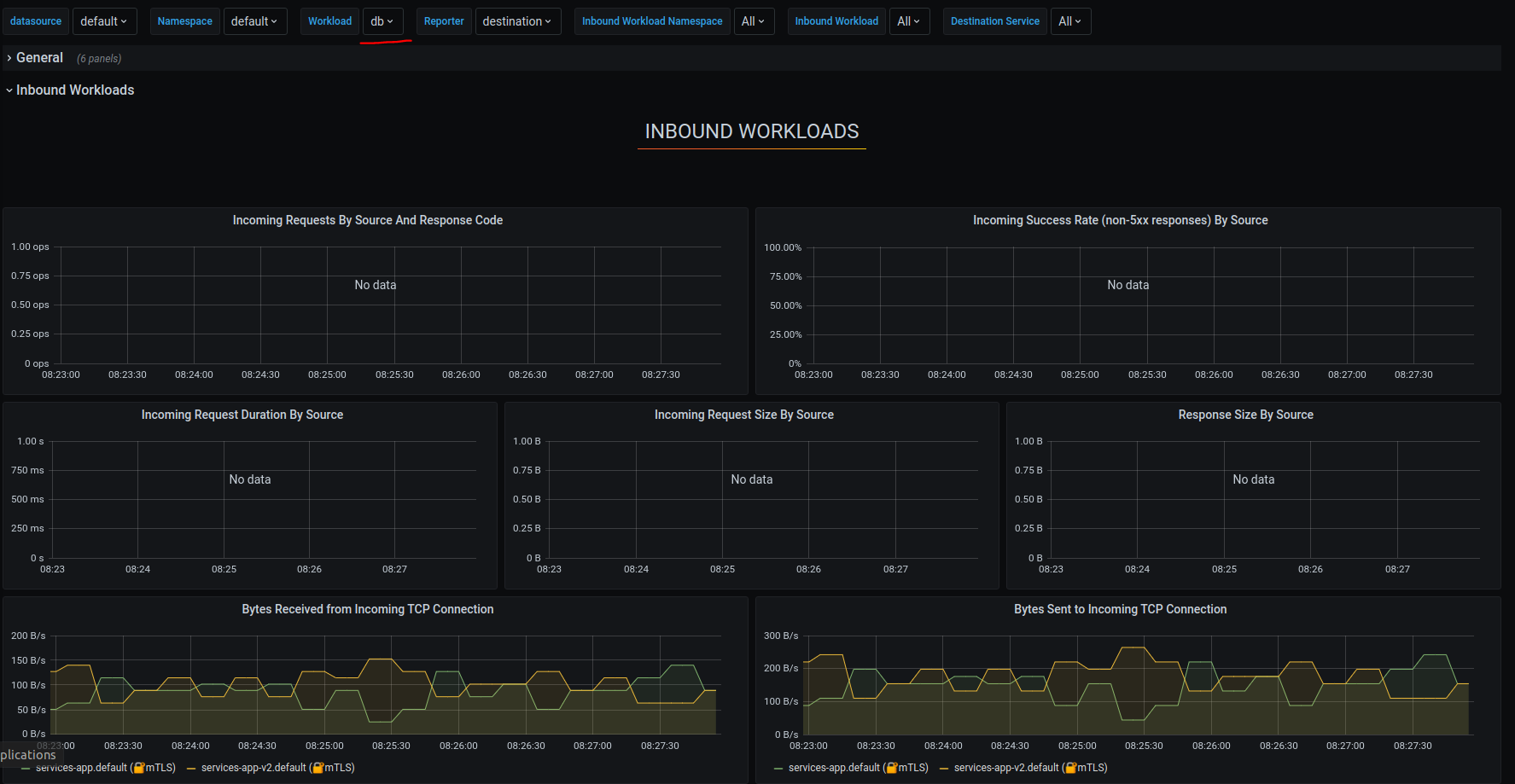




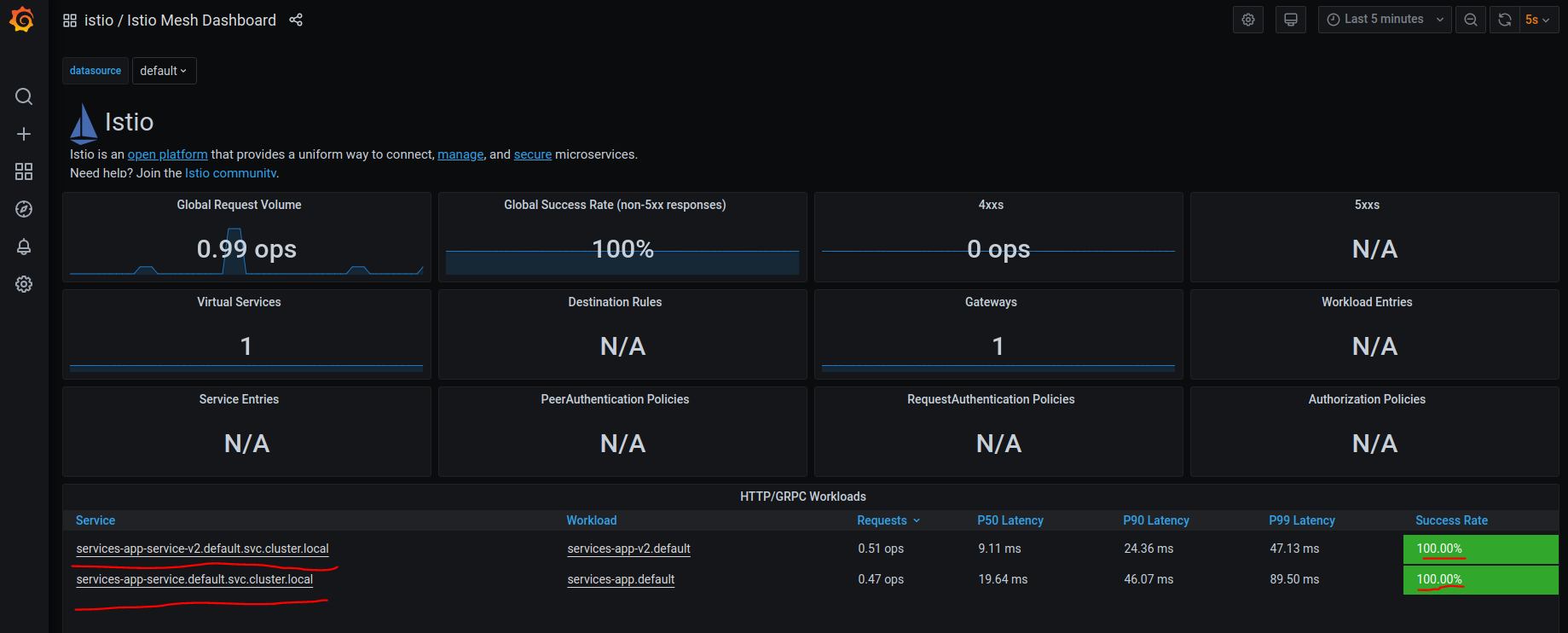
***Inbound Workload***





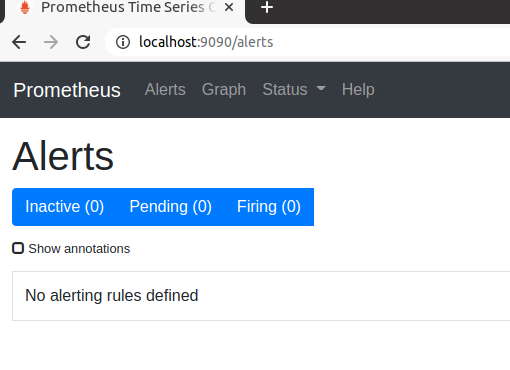


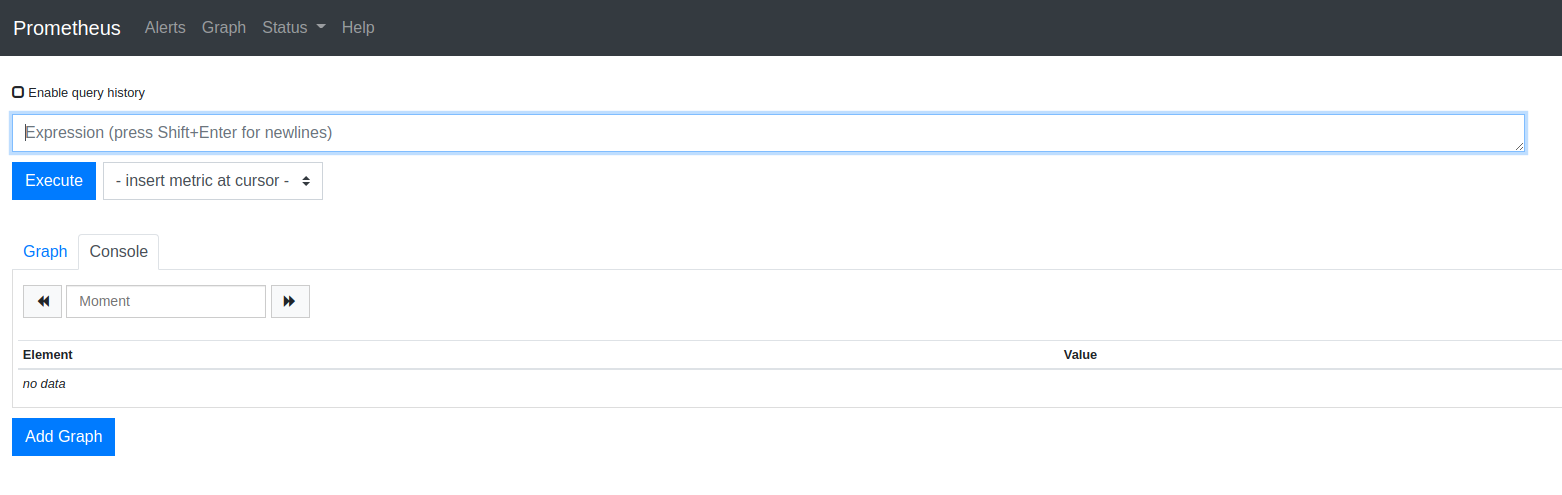
***Service Health***



***Note: Demo on Minikub, Ubuntu***

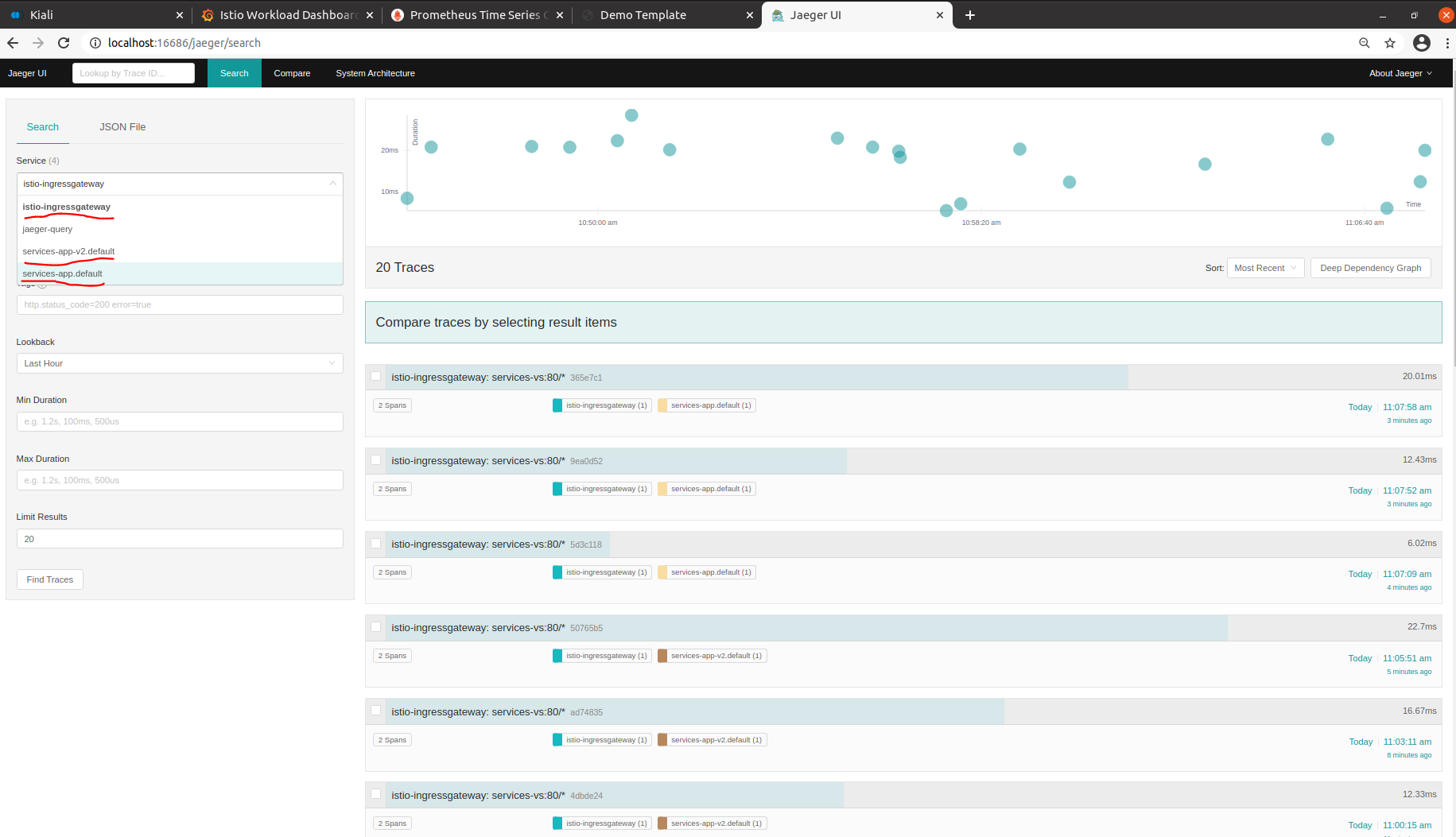
***3:*** [***Prometheus***](https://prometheus.io/) ***(Monitoring): 🡪 (TODO:Explore Options, Alerts, Monitoring)***





***Note: Demo on Minikub, Ubuntu***

***4:*** [***Jaeger***](https://www.jaegertracing.io/) ***(Service Tracing)(TODO:Explore Options)***



***Note: Demo on Minikub, Ubuntu***

***TODO***

1. ***Zipkin (Service Tracing)***
2. ***Service Discovery***
3. ***Log Management***