

[DevOps -Task-5]

**Prometheus** is a free software application used for event monitoring and alerting. It records real-time metrics in a time series database built using a HTTP Pull model with flexible queries and real-time alerting. The project is written in Go and licensed under the Apache 2 License, with [source code](https://en.wikipedia.org/wiki/Source_code) available on GitHub, and is a graduated project of the [Cloud Native Computing Foundation](https://en.wikipedia.org/wiki/Cloud_Native_Computing_Foundation), along with [Kubernetes](https://en.wikipedia.org/wiki/Kubernetes) and Envoy.

**Grafana** is an open-source platform for data visualization, monitoring and analysis. Our company uses this tool, paired up with Graylog, to monitor technical state of software systems we use internally or build for our customers. Grafana allows users to create

dashboards with panels, each representing specific metrics over a set time-frame. Every dashboard is versatile, so it could be custom-tailored for a specific project or any development and/or business needs.

We mostly use Grafana with Elasticsearch and Influx DB, but there is a variety of other supported data sources (Prometheus, MySQL, Postgres to name just a few) for this software. For each data source, Grafana has a customized query editor and specific syntax.

Kubernetes (also known as k8s or “kube”) is an open source container orchestration platform that automates many of the manual processes involved in deploying, managing, and scaling containerized applications.

In other words, you can cluster together groups of hosts running Linux containers, and Kubernetes helps you easily and efficiently manage those clusters.

[Kubernetes clusters](https://www.redhat.com/en/topics/containers/what-is-a-kubernetes-cluster) can span hosts across on-premise, [public](https://www.redhat.com/en/topics/cloud-computing/what-is-public-cloud), [private](https://www.redhat.com/en/topics/cloud-computing/what-is-private-cloud), or [hybrid clouds.](https://www.redhat.com/en/topics/cloud-computing/what-is-hybrid-cloud) For this reason, Kubernetes is an ideal platform for hosting [cloud-native applications](https://www.redhat.com/en/topics/cloud-native-apps) that require rapid scaling, like real-time data streaming through[Apache Kafka](https://www.redhat.com/en/topics/integration/what-is-apache-kafka).

Kubernetes was originally developed and designed by engineers at Google. Google was one of the early contributors to Linux container technology and has talked publicly about how everything at google runs in container (This is the technology behind Google cloud services.)

**Task** : Integrate Prometheus and Grafana and perform in following way:

1. Deploy them as pods on top of Kubernetes by creating resources Deployment/ Replica Set/ Pods or Services

2. And make their data to be remain persistent .

3. And both of them should be exposed to outside world.

**Few Requirements are there like :**

* Virtual-box with RHEL-8 installed and configured in it.
* The VM should have docker and Kubectl configured in it.
* The VM and the base system must have Internet connectivity.
* The Base system must have Kubectl installed in it.
* Minikube in base OS (Windows in my case).

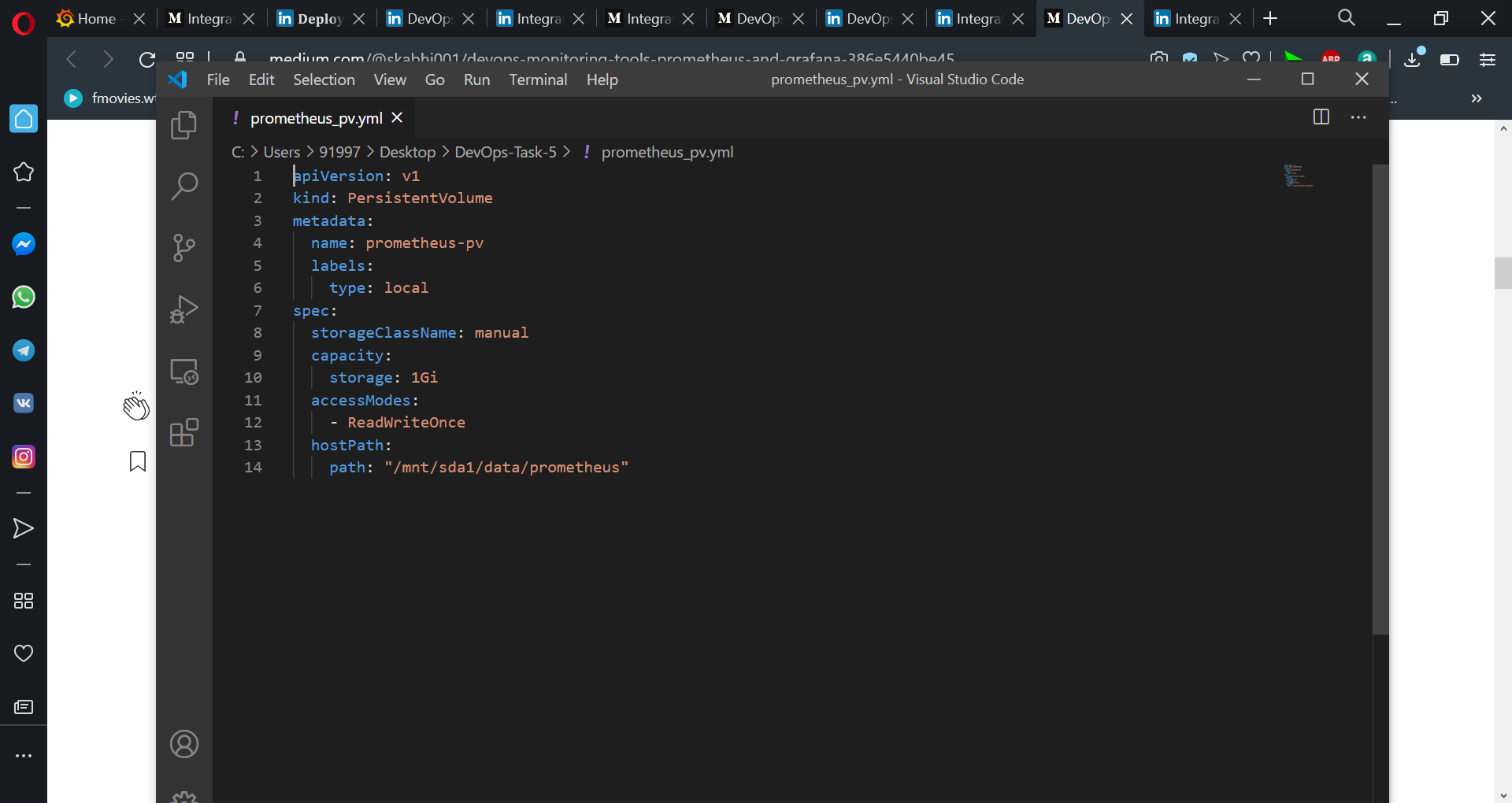
**Creation of Docker images for Prometheus and Grafana using Dockerfiles .**



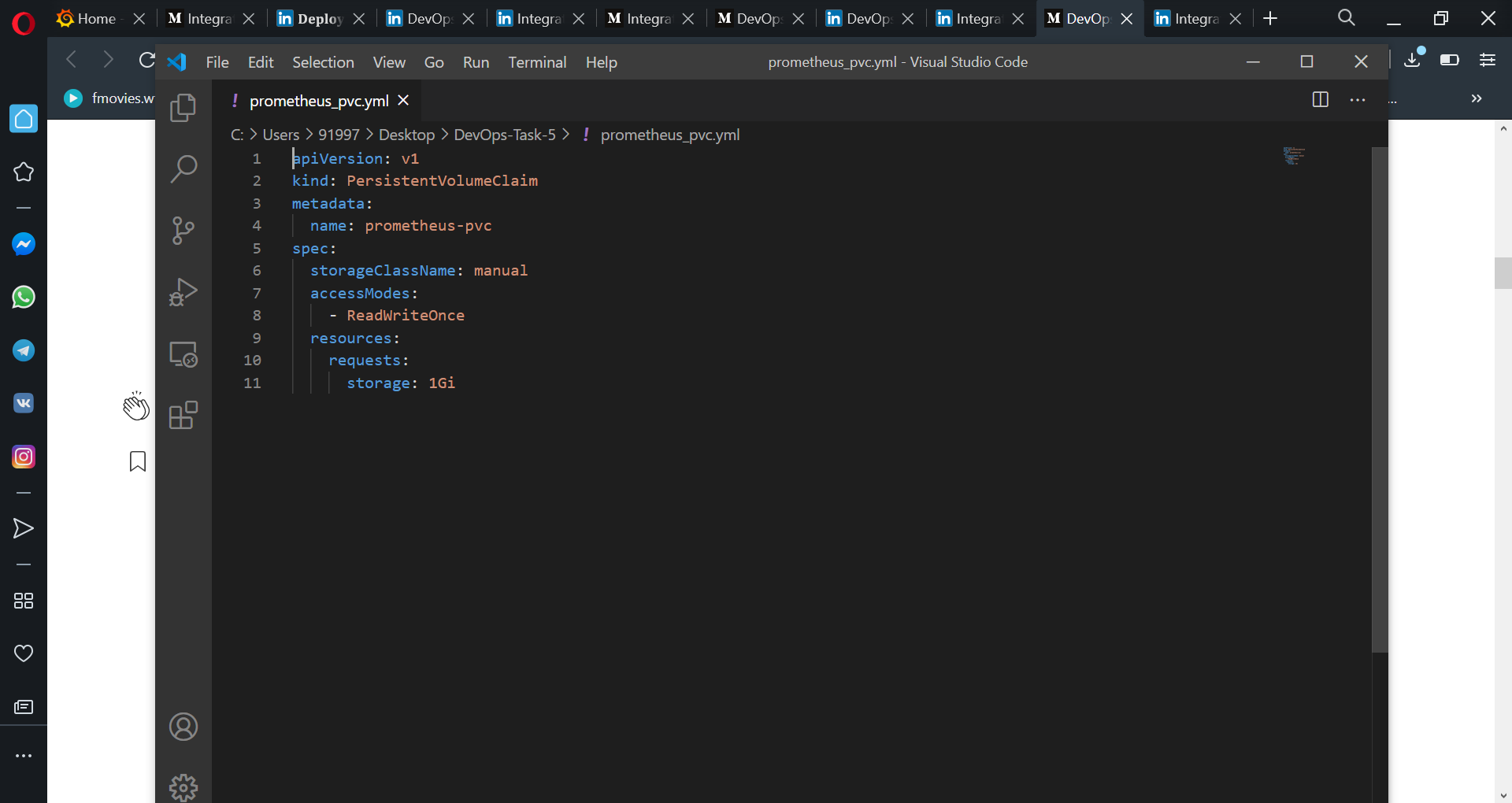
Build the Docker image with the following code below:

**docker build -t ravhitler/prometheus:v1**

Now, Persistent Volume (PV) will be created to make the data generated and stored by prometheus persistently.

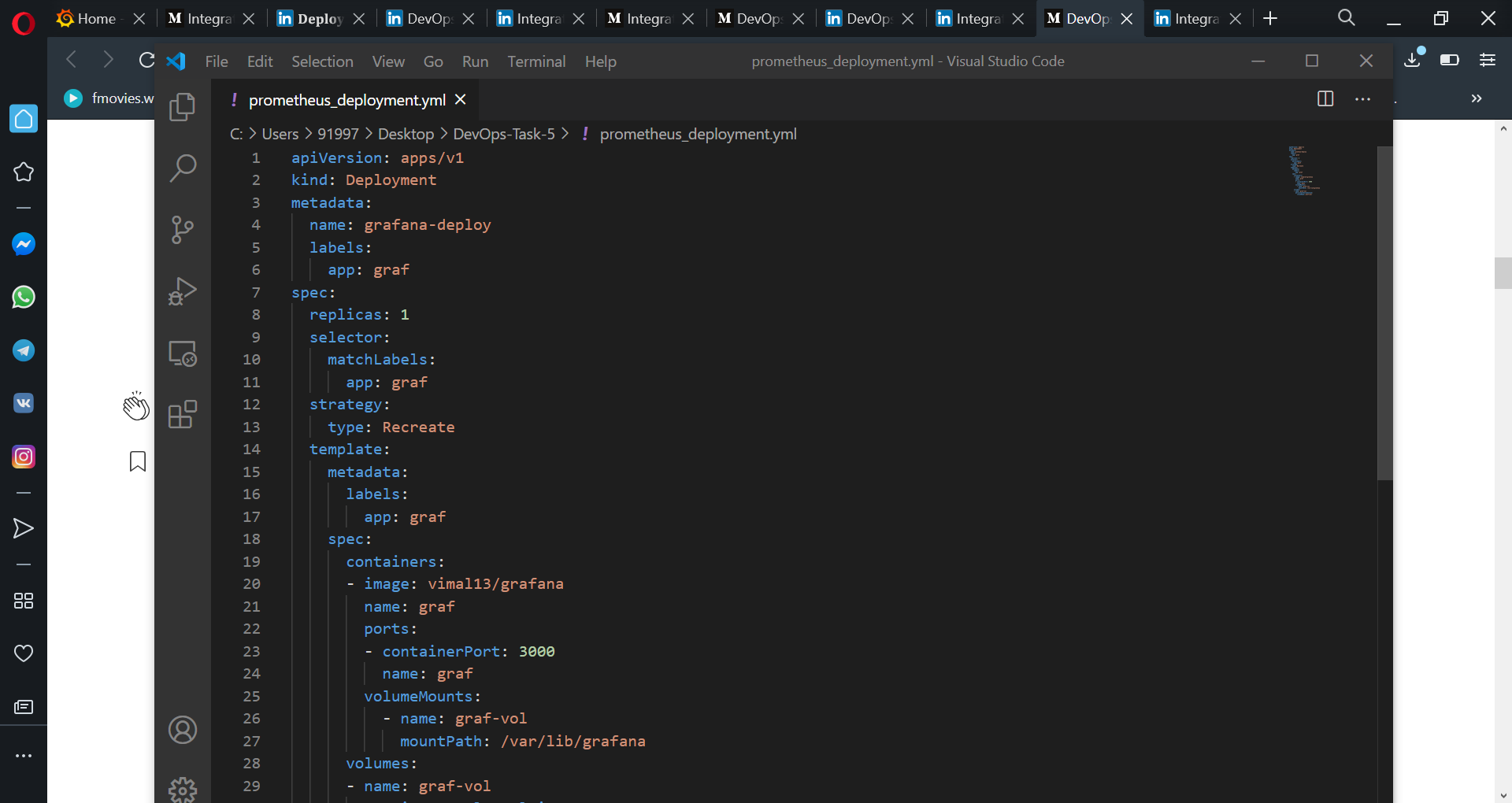


## **Persistent Volume Claim for Prometheus:**



## 

## Prometheus-deployment:



Now creating docker image of Grafana.

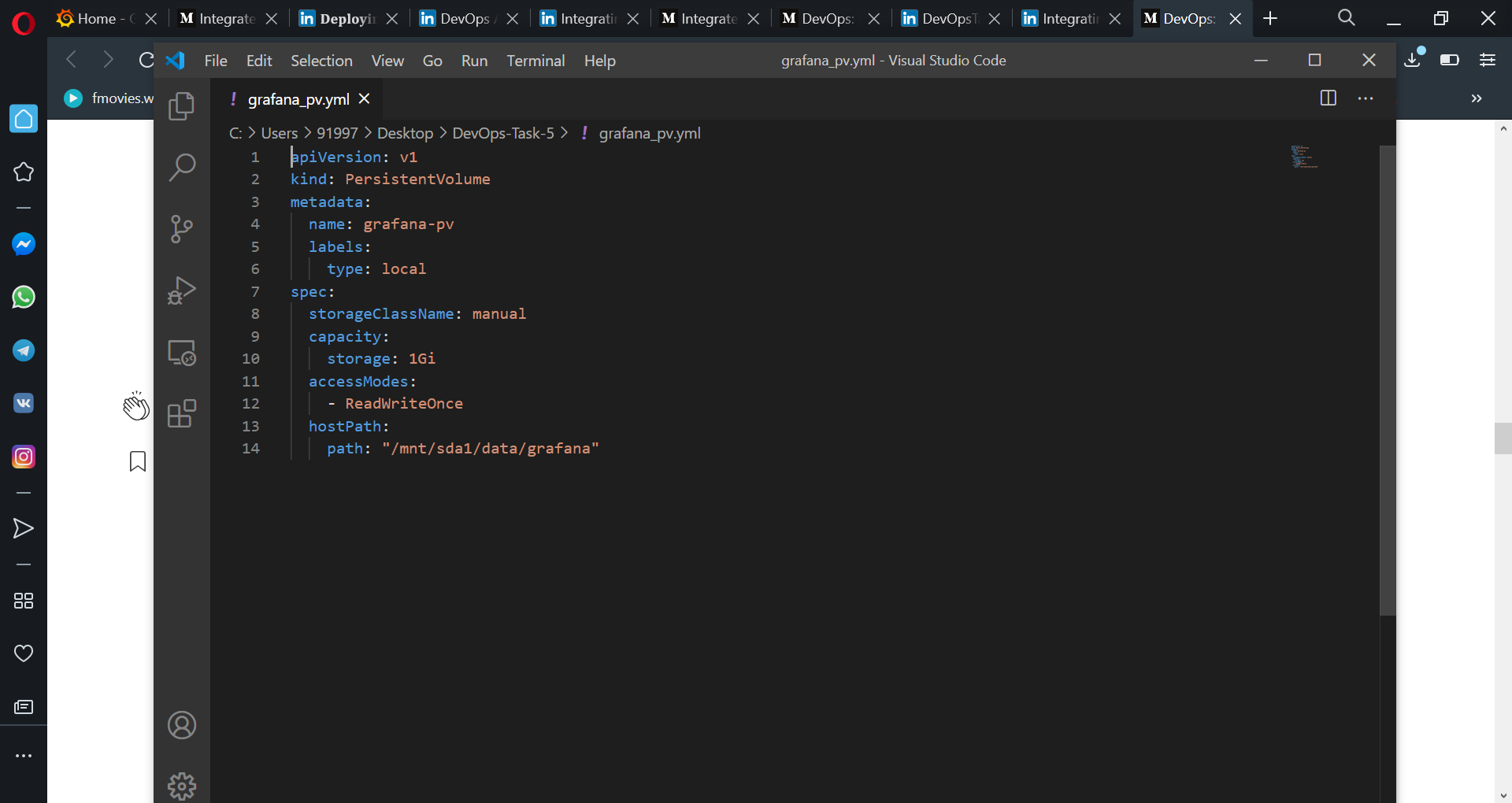
Dockerfile Grafana:



Build the above Docker image frm this command:

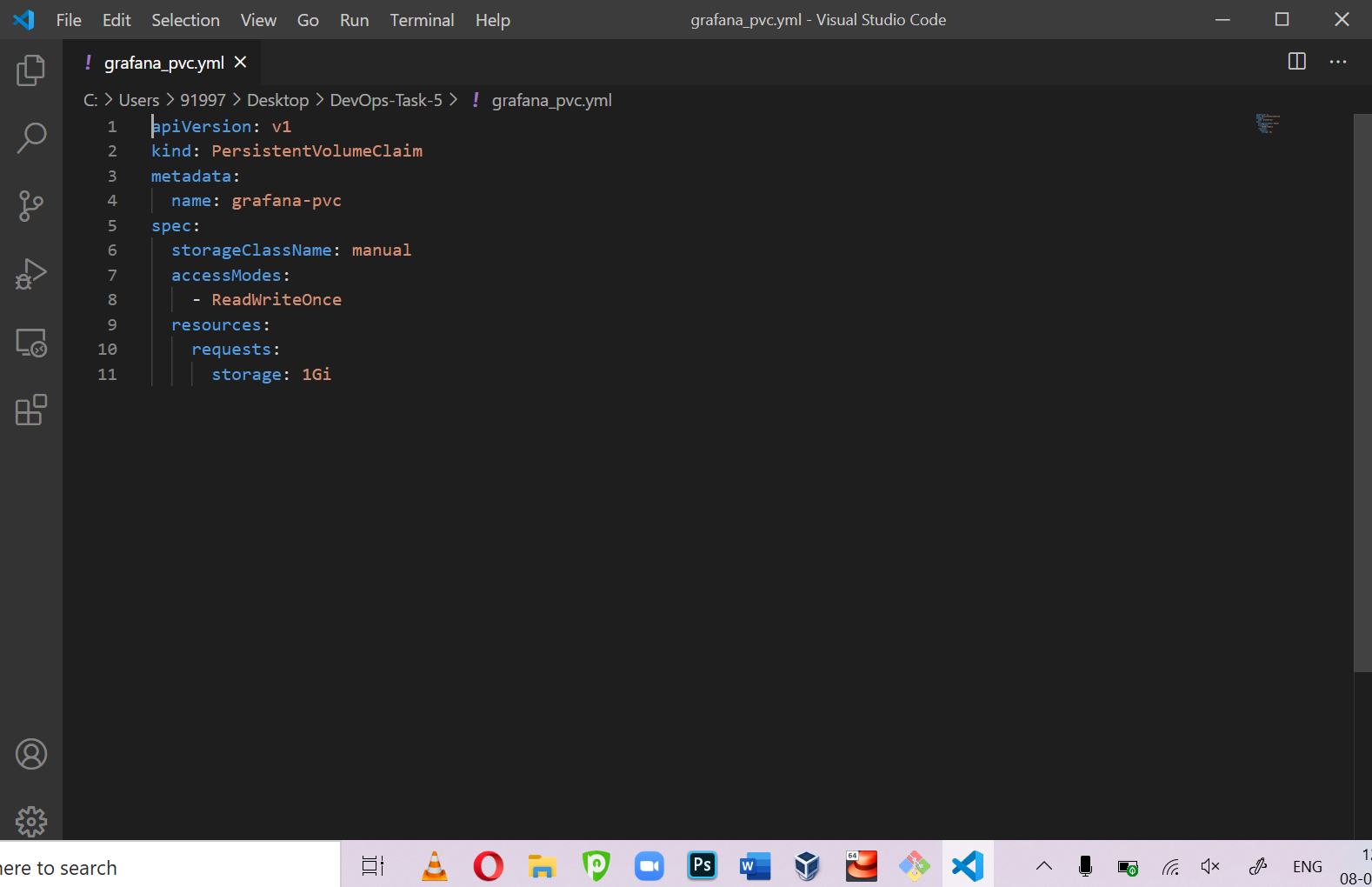
Docker build -t ravi/Grafana:latest

## Persistent Volume creation for grafana:



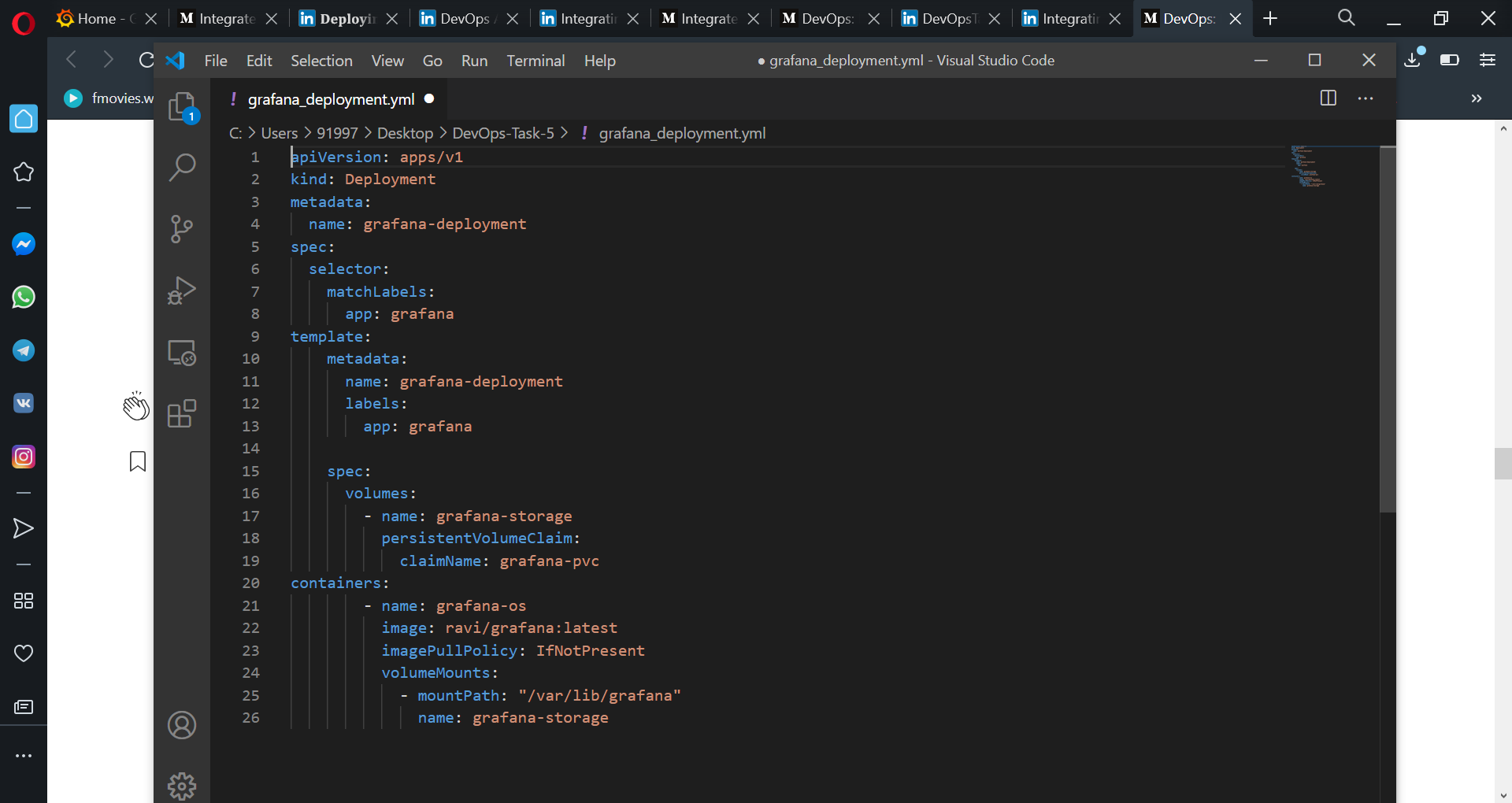
## 

## PersistentVolumeClaim creation for grafana:

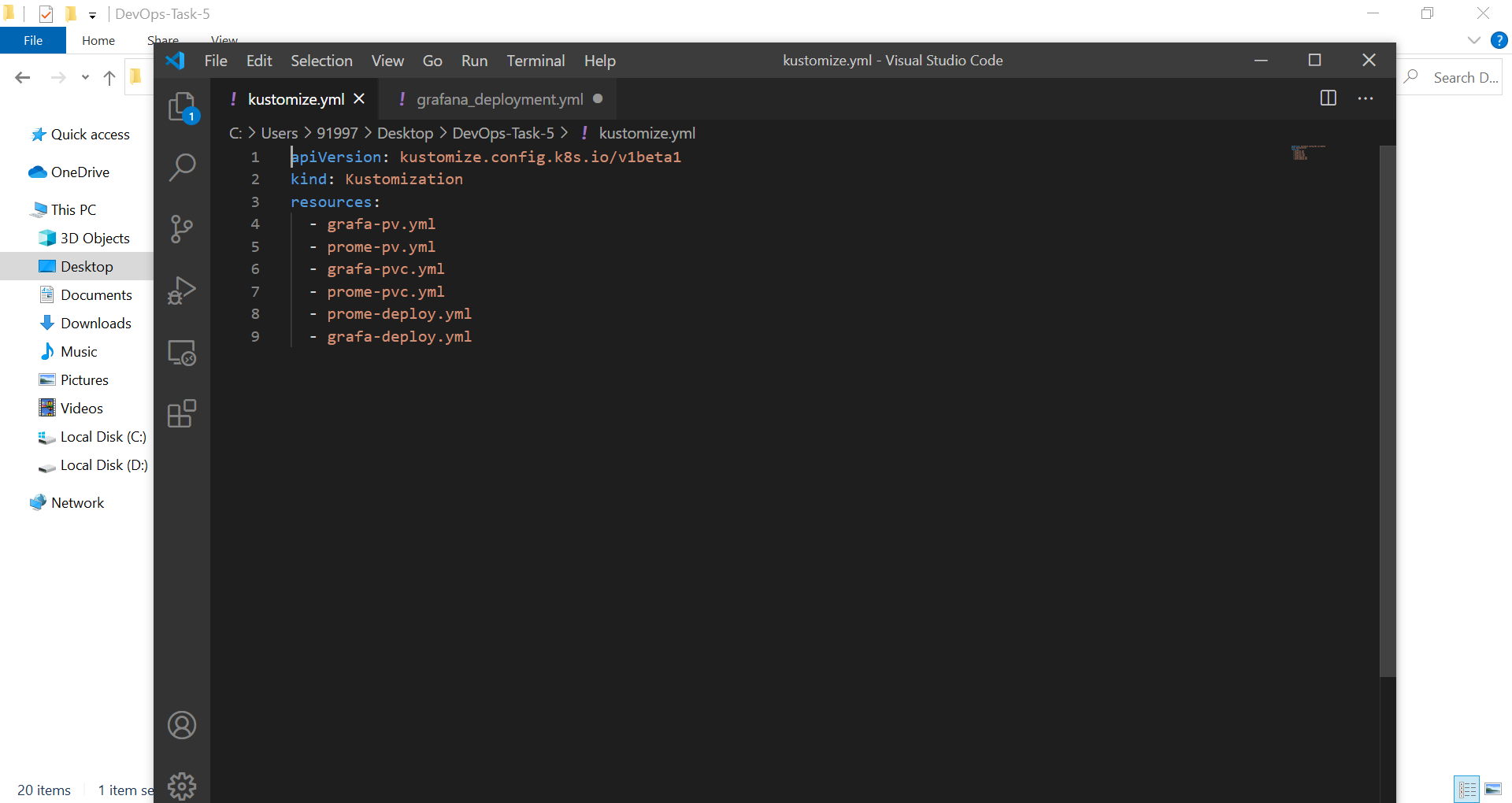


## 

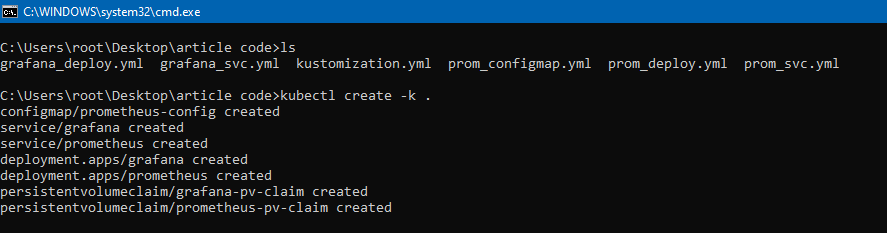
## Grafana-deployment creation code:



* **Kustomization** file for running all the above files using the single command :
* **Kustomize** is a tool that lets you create an entire Kubernetes application out of individual pieces
* **kustomization service to start all the services i.e pods, services, replica set .Deployment and persistent volume immediately without going and starting it individually.**

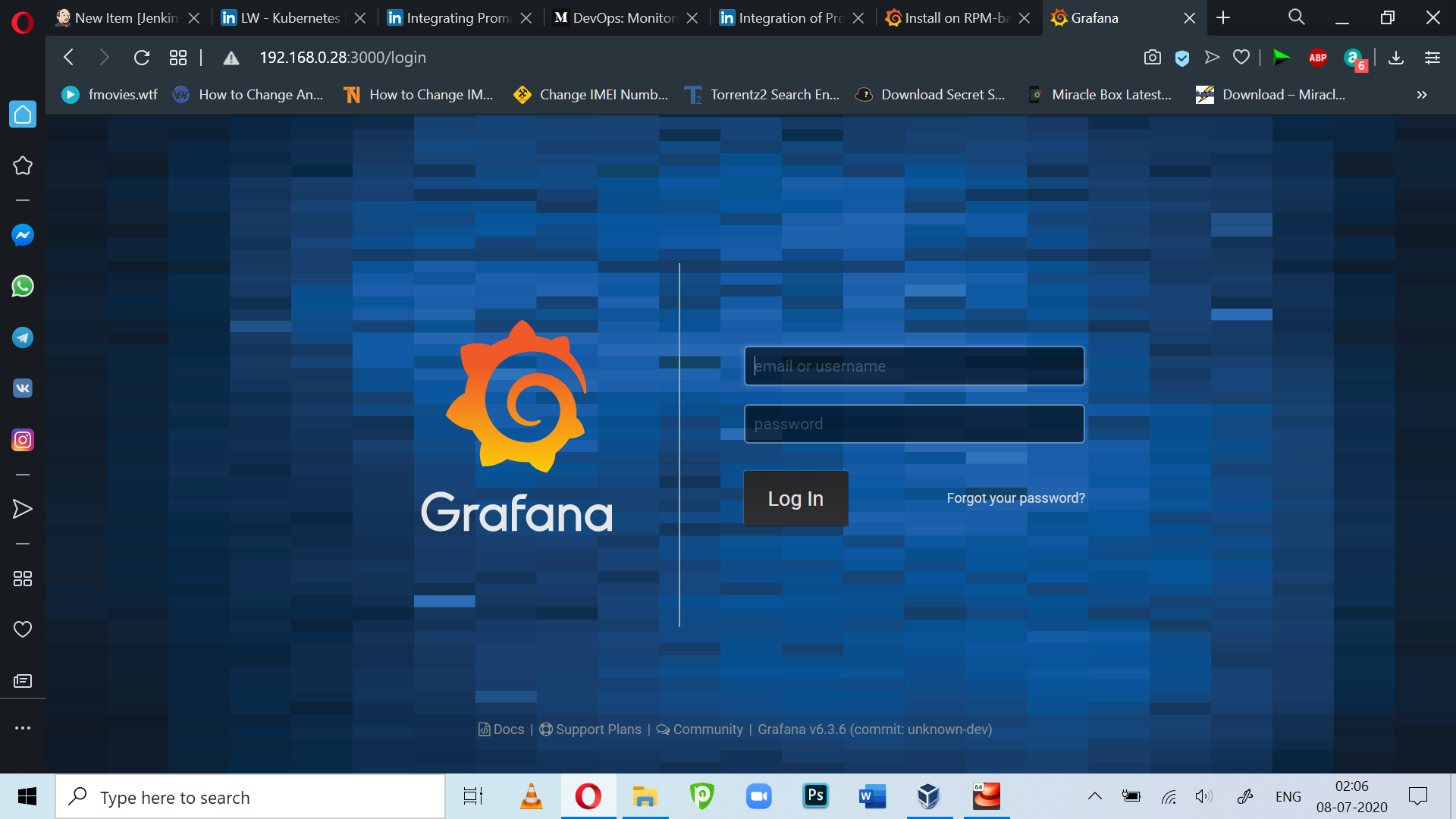


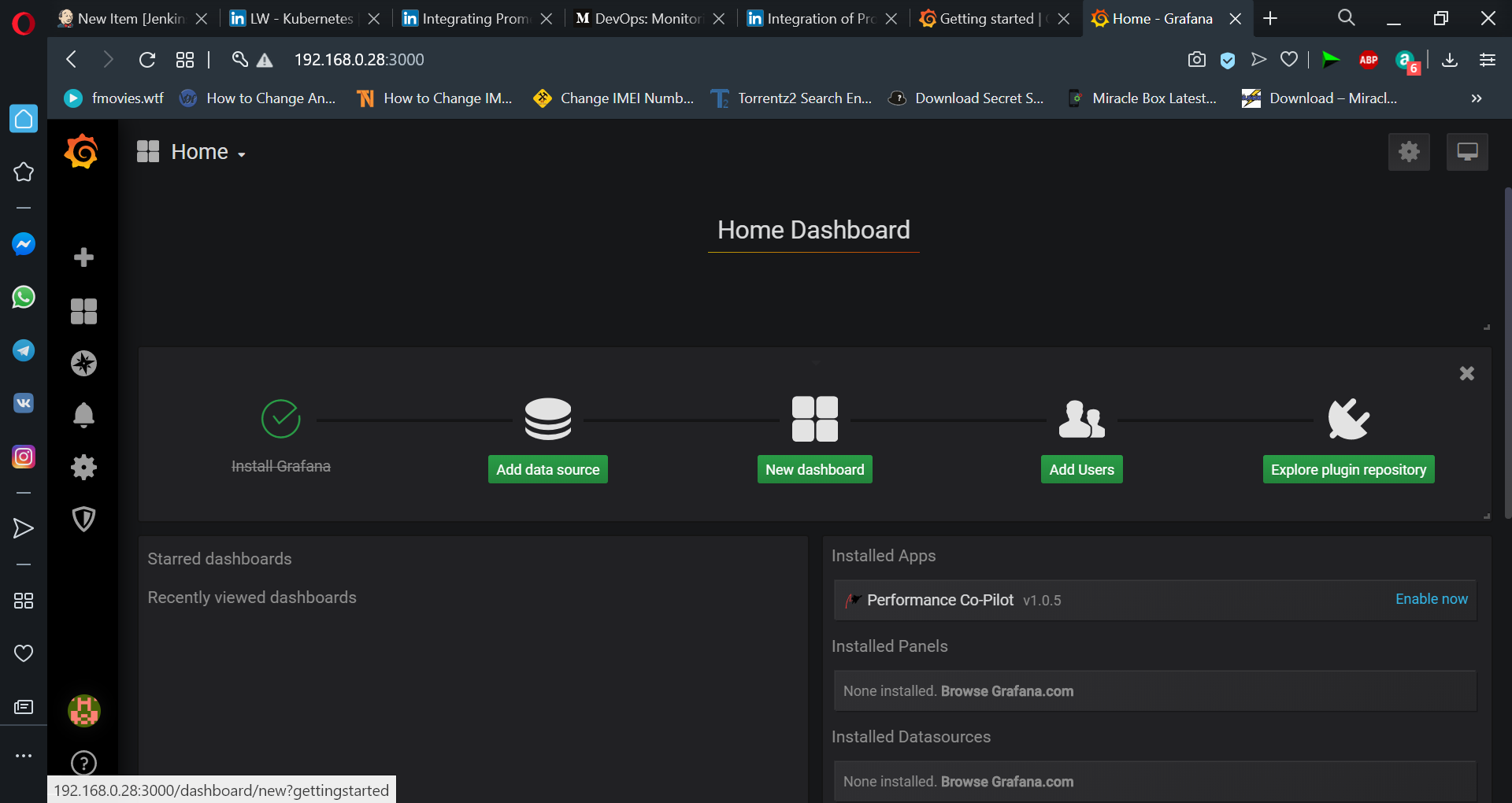
Setup for Kustomization file using **kubectl create -k .** command



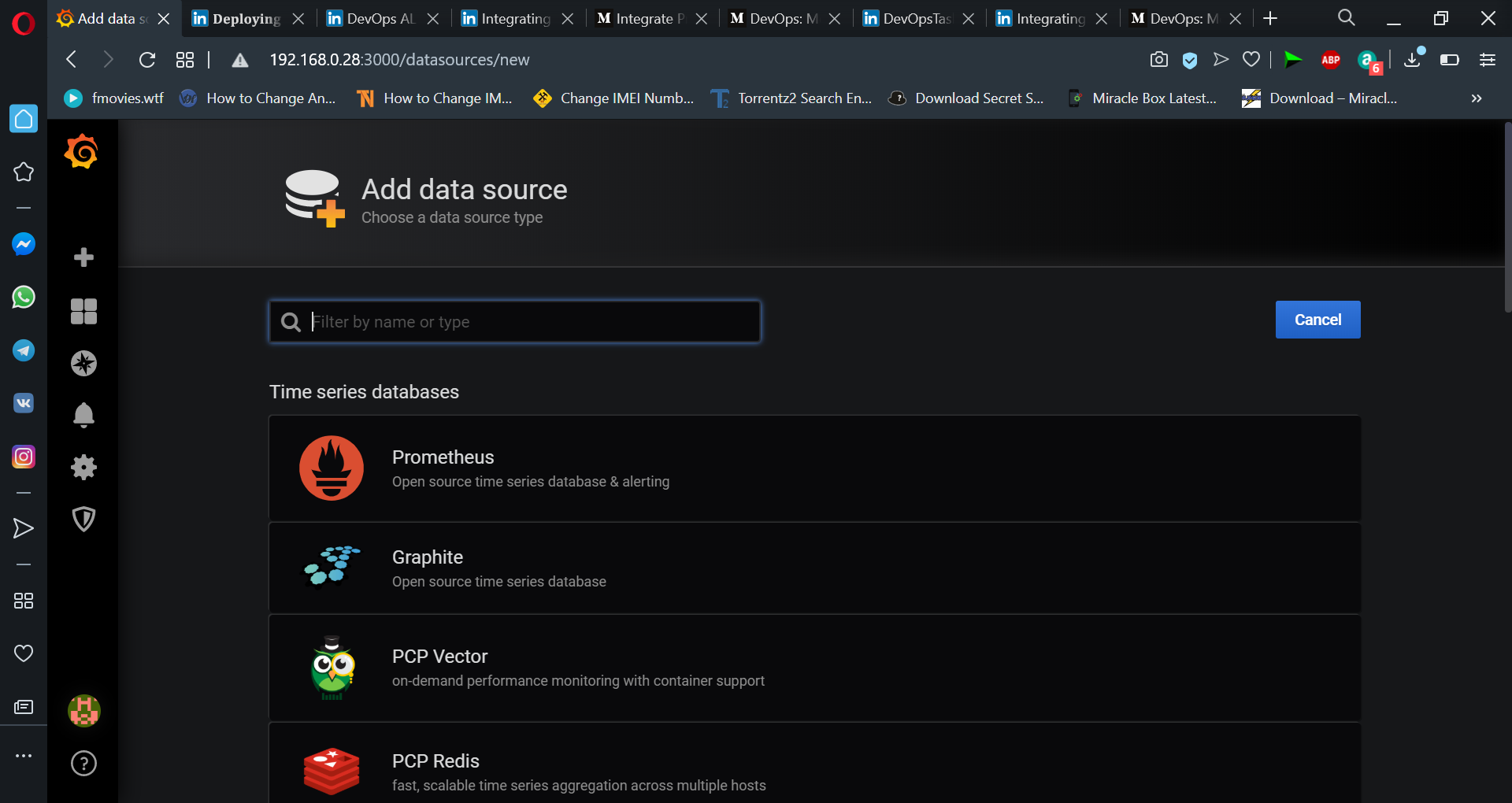
**[kubectl get all ] -** Grafana and Prometheus Pods, services, deployment are running .

Now we can access Grafana 192.168.0.28:3000. 3000 is bydefault port assigned to grafana.

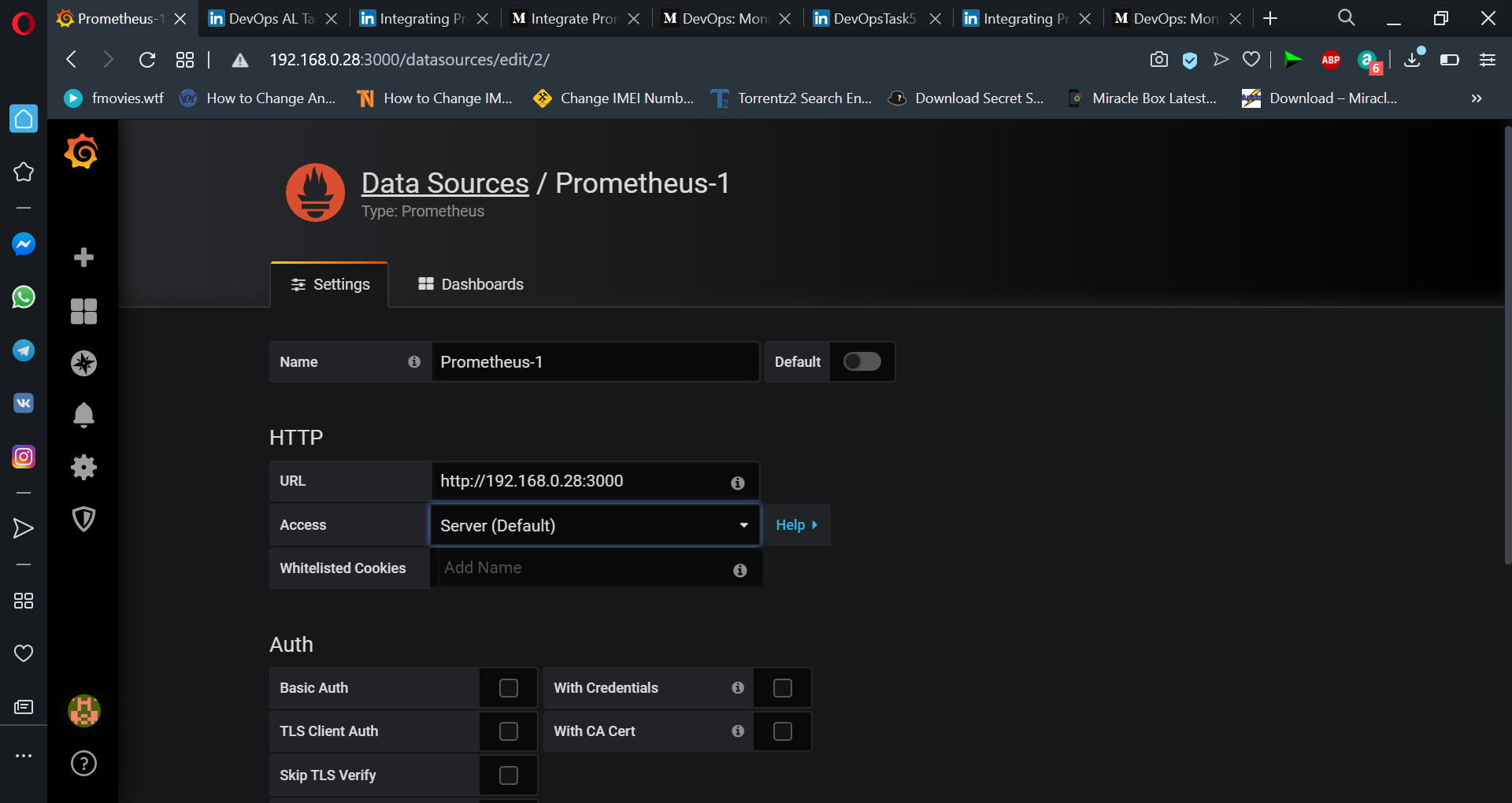


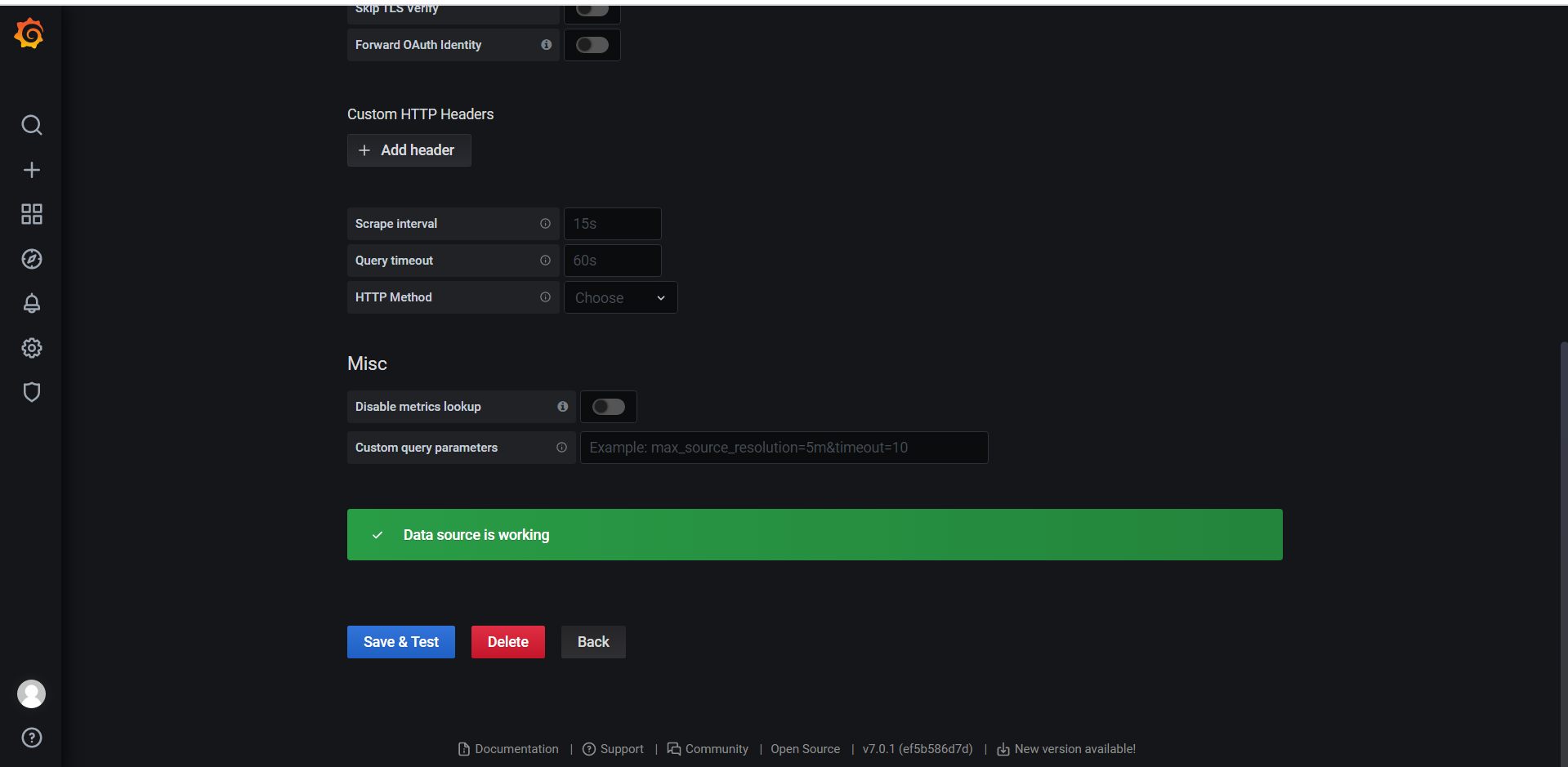
Bydefault username and password are – admin, but once you login. You must change it.

After login, we can add Prometheus as a data source for Grafana.



we need to provide the URL of Prometheus and save it as a data source.





Query Dashboard:



Final result showing



For more information and regarding yml file and other documents relate to this project you can check out the below GitHub account of mine :

GitHub URL : https://github.com/raviiraj/DevOps-Task-5