

# Building a Metrics Dashboard

REVIEW
CODE REVIEW
HISTORY

## Meets Specifications

### Dear Student,

Amazing work in this submission I was truly happy to see the results of all your learnings come so clearly into your final project. I really love to see your understanding of the `Jaeger` tracing and `PromQL` queries you wrote while building the dashboards.

## Extra Materials

Here are a few resources you might find useful for more insight and further learning.

- [6 Best Practices for Application Deployments](#)
- [10 Reasons Why Developers Love Docker](#)
- [Better Dashboarding – Grafana or SquaredUp?](#)
- [What is Grafana? Why Use It? Everything You Should Know About It](#)

Keep up the good work and congratulations on passing the project. 🎉

All the best for your next project

## Project and Cluster Staging

✓

The README includes a screenshot of the `kubectl get pods` output, showing successful installation of the components.

## Well Done!

You had provided the screenshots for the services and screenshots show that all the services are up and running. ✓

```
E:\Documents\udacity\SUSE\CNAND_nd064_C4_Observability_Starter_Files\Project_Starter_Files-Building_a_Metrics_Dashboard\manifests\app>kubectl get pods --all-namespaces
NAMESPACE   NAME                                     READY   STATUS    RESTARTS   AGE
kube-system  helm-install-traefik-nc94t             0/1     Completed 0          19d
monitoring   prometheus-kube-prometheus-operator-bcdfdbc79-v8lbt  1/1     Running   33         19d
kube-system  local-path-provisioner-7ff9579c6-2cb4m  1/1     Running   39         19d
monitoring   prometheus-prometheus-node-exporter-jzs2g  1/1     Running   32         19d
kube-system  metrics-server-7b4f8b595-gr2fv         1/1     Running   33         19d
observability jaeger-operator-5977dbf59f-5g56z        1/1     Running   32         19d
default      svc1b-backend-w7wk5                    1/1     Running   0          10h
kube-system  svc1b-traefik-4rznb                     2/2     Running   68         19d
```

```
monitoring      prometheus-kube-state-metrics-569d7854c4-mrct8      1/1      Running      33      19d
observability  my-jaeger-tracing-677f5cb694-nmfsf                 1/1      Running      1       11h
monitoring     alertmanager-prometheus-kube-prometheus-alertmanager-0  2/2      Running      64      19d
kube-system    coredns-88dbd9b97-85z52                             1/1      Running      32      19d
kube-system    traefik-5dd496474-h2zrf                             1/1      Running      32      19d
monitoring     prometheus-prometheus-kube-prometheus-prometheus-0    2/2      Running      64      19d
default        backend-d8fbd774f-664r6                             1/1      Terminating 0       10h
monitoring     prometheus-grafana-57589d7b8d-46cbt                 2/2      Running      64      19d
default        svc1b-frontend-n5ldr                                1/1      Running      0       10h
default        backend-6b865f87c8-8nlg2                             2/2      Running      3       10h
default        frontend-65b6cdb9d5-cr6dk                           1/1      Running      5       10h
E:\Documents\udacity\SUSE\CNAND_nd064_C4_Observability_Starter_Files\Project_Starter_Files-Building_a_Metrics_Dashboard\manifests\app>
```



The README has a screenshot showing the home page after login.

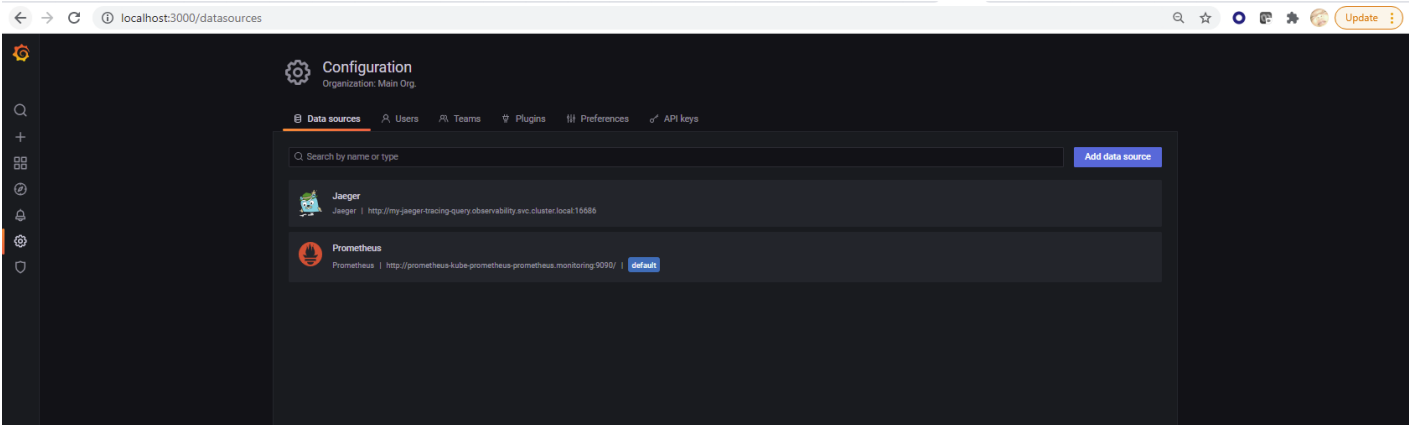
## Well done!

Your provided screenshot clearly states that you had successfully exposed the Grafana on the port `3000` .



The README includes a screenshot of the dashboard in Grafana that shows Prometheus as a source.

That's so nice to see that you added the `Jaeger` and `Prometheus` sources on `Grafana` .



## Create Dashboards to Measure SLIs



The README contains a definition of the SLIs, based on an SLO of monthly uptime and request response time.

### Good Work!

You had clearly defined the **monthly uptime** and **request response time** by writing the `SLO` and `SLIs` for both metrics.



The README contains KPI metrics based on the SLIs given in the project instructions.

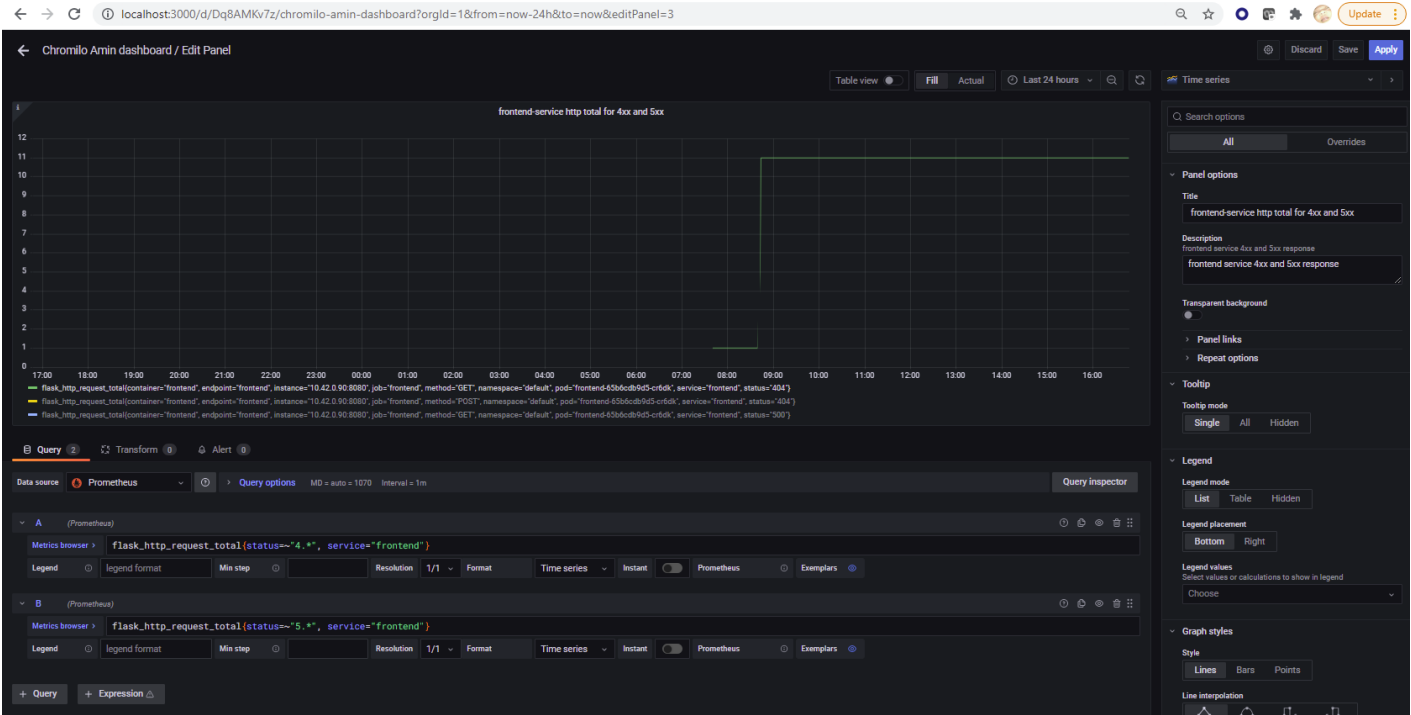
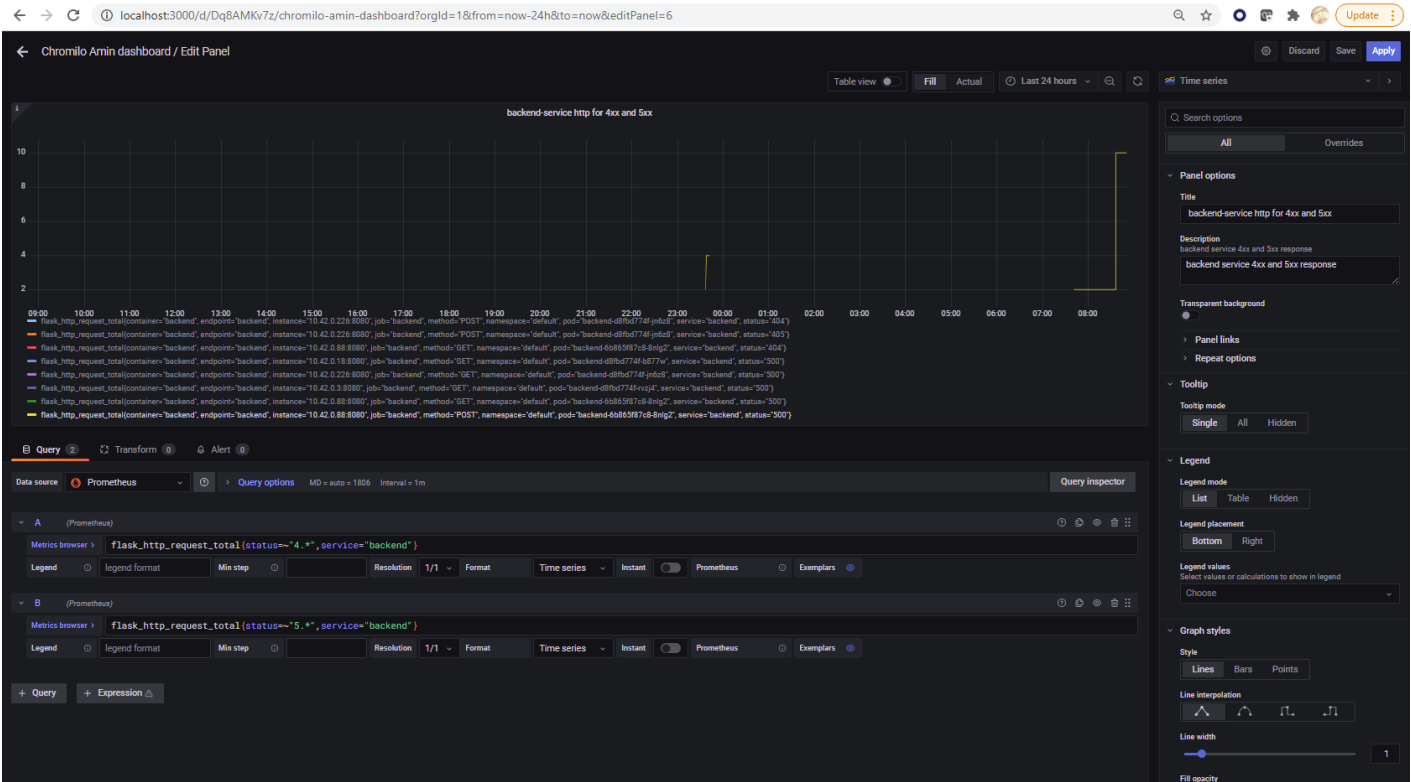
### Good Job!

You had done a nice job in writing the `KPI` based on the written `SLIs` as expected.



The README will include a screenshot of the finished dashboard containing panels with graphs that measure the required metrics provided by the course.

Well done for providing the screenshot for the `backend` service and also measuring 40x and 50x errors over a 24 hour period.



# Tracing



The sample Python file contains a trace and span code to perform Jaeger traces on the Python services.

## Good work!

Provided python file inside the trail directory contains required `trace` and `span` code. This is showing your great understanding of the tracing.



The README contains a screenshot displaying the trace history of the application.

## Good Job!

You provided the screenshot for the trace history of the `Jaeger` as required.



The README contains a comprehensive trouble ticket written for developers.

Good Job!

You had created a ticket for the error that occurred while running the application.

Creating Your own Dashboard



The README contains a list of four SLIs and SLOs, created by the student based on the criteria provided in the instructions.

Well Done!

You had correctly defined the `SLI` and `SLO` .



The README contains a list of 2-3 KPIs per SLI as well as a description of why those KPIs were chosen.

Great work

You had created `KPIs` to accurately measure the metrics.



The README contains a screenshot of the completed dashboard as well as a brief description of each graph.

Well Done!

You had created the required dashboards after listing the SLIs and SLOs and KPIs for the SLIs.

[DOWNLOAD PROJECT](#)

RETURN TO PATH

Rate this review

START