# CS5830: BIG DATA LABORATORY
# ASSIGNMENT 7 : REPORT

Saketh D.
ED19B010

## Task 1 - MNIST APP With Monitoring:

- The program from assignment 6 has been modified to include the monitoring of metrics such as the api runtime and total api call counter using prometheus_client.
- The address **localhost:10000** contains these metrics
- Downloaded the prometheus-monitoring and node-exporter precompiled binaries from https://prometheus.io/download/
- Modified the **prometheus.yml** file to include metrics from node-exporter and from the mnist app.
- **localhost:9100** is the default address where node-exporter logs the metrics and **localhost:9090** is the default address for prometheus
- Grafana was installed by following the instructions present in the following link: https://grafana.com/docs/grafana/latest/setup-grafana/installation/ debian/
- The prometheus data source is used to create the visualization in Grafana for the following metrics: API runtime, total number of API calls, API memory utilization, API CPU utilization and API network I/O bytes.
- Started the node-exporter using the following command:

```
saketh@saketh-IdeaPad-3-14IML05-U1a:~/Desktop/assignment7$ cd node_exporter-1.8.0.linux-amd64/
saketh@saketh-IdeaPad-3-14IML05-U1a:~/Desktop/assignment7/node_exporter-1.8.0.linux-amd64$ ./node_exporter
ts=2024-05-17T16:56:02.203Z caller=node_exporter.go:193 level=info msg="Starting node_exporter" version="(version=1.8.0, branch=HEAD, revision=cadb1d1190ad95c66b951758f01ff4c94e55e6ce)"
ts=2024-05-17T16:56:02.205Z caller=node_exporter.go:194 level=info msg="Build context" build_context="(go=go1.22.2, platform=linux/amd64, user=root@587d3f12650c, date=20240424-13:15:48, tags=unknown)"
ts=2024-05-17T16:56:02.208Z caller=filesystem_common.go:111 level=info collector=filesystem msg="Parsed flag --collector.filesystem.mount-points-exclude" flag=^/(dev|proc|run/credentials/.+|sys|var/lib/docker/.+|var/lib/containers/storage/.+)($|/)
ts=2024-05-17T16:56:02.209Z caller=filesystem_common.go:113 level=info collector=filesystem msg="Parsed flag --collector.filesystem.fs-types-exclude" flag=^(autofs|binfmt_misc|bpf|cgroup2?|configfs|debugfs|devpts|devtmpfs|fusectl|hugetlbfs|iso9660|mqueue|nsfs|overlay|proc|procfs|pstore|rpc_pipefs|securityfs|selinuxfs|squashfs|sysfs|tracefs)$
ts=2024-05-17T16:56:02.210Z caller=diskstats_common.go:111 level=info collector=diskstats msg="Parsed flag --collector.diskstats.device-exclude" flag=^(z?ram|loop|fd|(h|s|v|xv)d[a-z]|nvme\d+n\d+p)\d+$
ts=2024-05-17T16:56:02.210Z caller=node_exporter.go:111 level=info msg="Enabled collectors"
ts=2024-05-17T16:56:02.210Z caller=node_exporter.go:118 level=info collector=arp
```

- Started prometheus using the following commands:

```
saketh@saketh-IdeaPad-3-14IML05-U1a:~/Desktop/assignment7$ cd prometheus-2.45.5.linux-amd64/
saketh@saketh-IdeaPad-3-14IML05-U1a:~/Desktop/assignment7/prometheus-2.45.5.linux-amd64$ ./prometheus --config.file=./prometheus.yml
ts=2024-05-17T17:54:10.216Z caller=main.go:534 level=info msg="No time or size retention was set so using the default time retention" duration=15d
ts=2024-05-17T17:54:10.216Z caller=main.go:578 level=info msg="Starting Prometheus Server" mode=server version="(version=2.45.5, branch=HEAD, revision=2b052add78646ff
89d193dac84eae8855d11565a)"
ts=2024-05-17T17:54:10.216Z caller=main.go:583 level=info build_context="(go=go1.21.9, platform=linux/amd64, user=root@98598c5dfe5e, date=20240502-08:58:53, tags=netg
o,builtinassets,stringlabels)"
ts=2024-05-17T17:54:10.216Z caller=main.go:584 level=info host_details="(Linux 6.5.0-28-generic #29~22.04.1-Ubuntu SMP PREEMPT_DYNAMIC Thu Apr  4 14:39:20 UTC 2 x86_6
```

- Started the MNIST API with the following command:

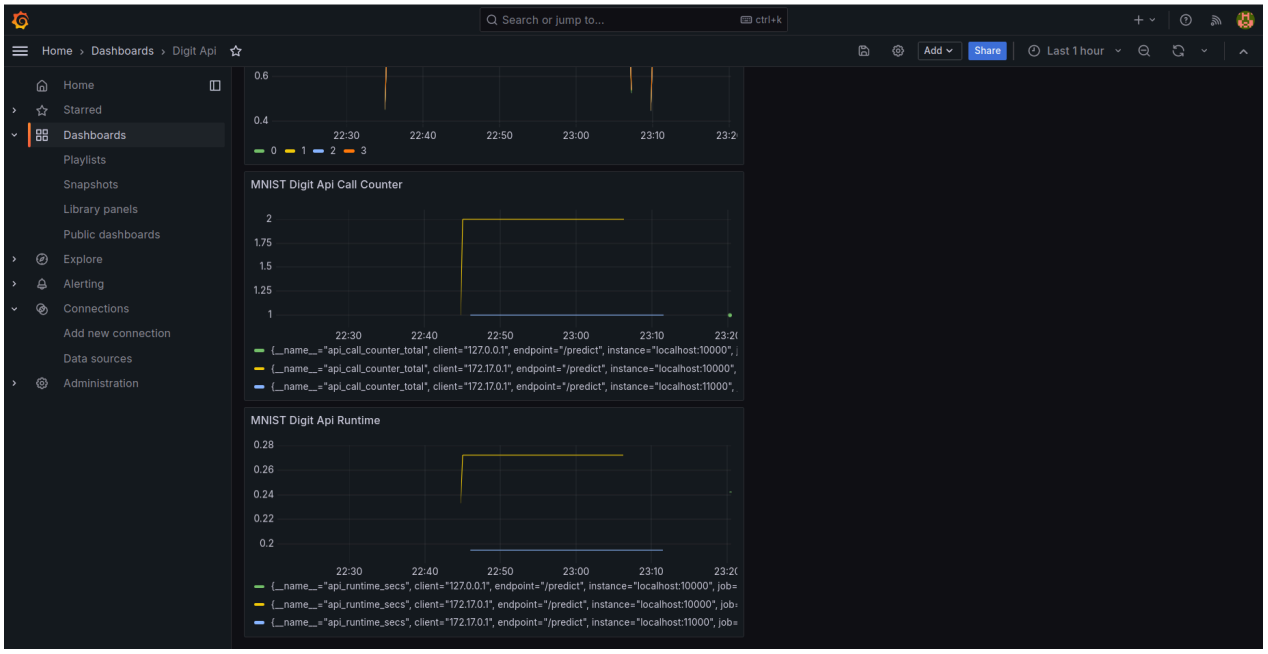
```
saketh@saketh-IdeaPad-3-14IML05-U1a:~/Desktop/assignment7$ python3 mnist_api.py /home/saketh/Desktop/assignment6/mymodel.keras
2024-05-14 22:40:17.964084: I external/local_tsl/tsl/cuda/cudart_stub.cc:32] Could not find cuda drivers on your machine, GPU will not be used.
2024-05-14 22:40:17.971804: I external/local_tsl/tsl/cuda/cudart_stub.cc:32] Could not find cuda drivers on your machine, GPU will not be used.
2024-05-14 22:40:18.088961: I tensorflow/core/platform/cpu_feature_guard.cc:210] This TensorFlow binary is optimized to use available CPU instructions in performance-
critical operations.
To enable the following instructions: AVX2 FMA, in other operations, rebuild TensorFlow with the appropriate compiler flags.
2024-05-14 22:40:19.950053: W tensorflow/compiler/tf2tensorrt/utils/py_utils.cc:38] TF-TRT Warning: Could not find TensorRT
INFO:     Started server process [10986]
INFO:     Waiting for application startup.
INFO:     Application startup complete.
INFO:     Uvicorn running on http://0.0.0.0:8000 (Press CTRL+C to quit)
INFO:     127.0.0.1:56796 - "GET / HTTP/1.1" 404 Not Found
INFO:     127.0.0.1:56796 - "GET /favicon.ico HTTP/1.1" 404 Not Found
INFO:     127.0.0.1:49992 - "GET /docs HTTP/1.1" 200 OK
INFO:     127.0.0.1:49992 - "GET /openapi.json HTTP/1.1" 200 OK
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 204ms/step
INFO:     127.0.0.1:43960 - "POST /predict HTTP/1.1" 200 OK
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 164ms/step
INFO:     127.0.0.1:59870 - "POST /predict HTTP/1.1" 200 OK
```

- Started the grafana server with the following command:

```
saketh@saketh-IdeaPad-3-14IML05-U1a:~/Desktop/assignment7$ sudo service grafana-server start
[sudo] password for saketh:
saketh@saketh-IdeaPad-3-14IML05-U1a:~/Desktop/assignment7$
```

- The Grafana dashboard is present at the address **localhost:3000** (default)

- The following images show the grafana dashboard which contains the visualizations of the metrics:

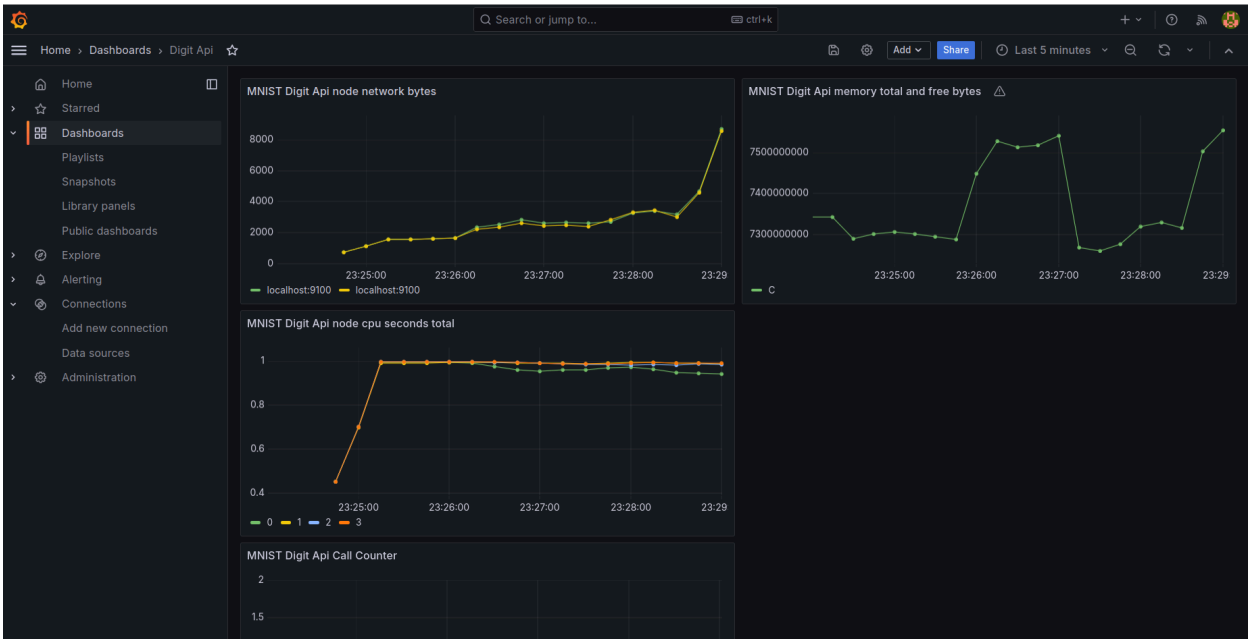



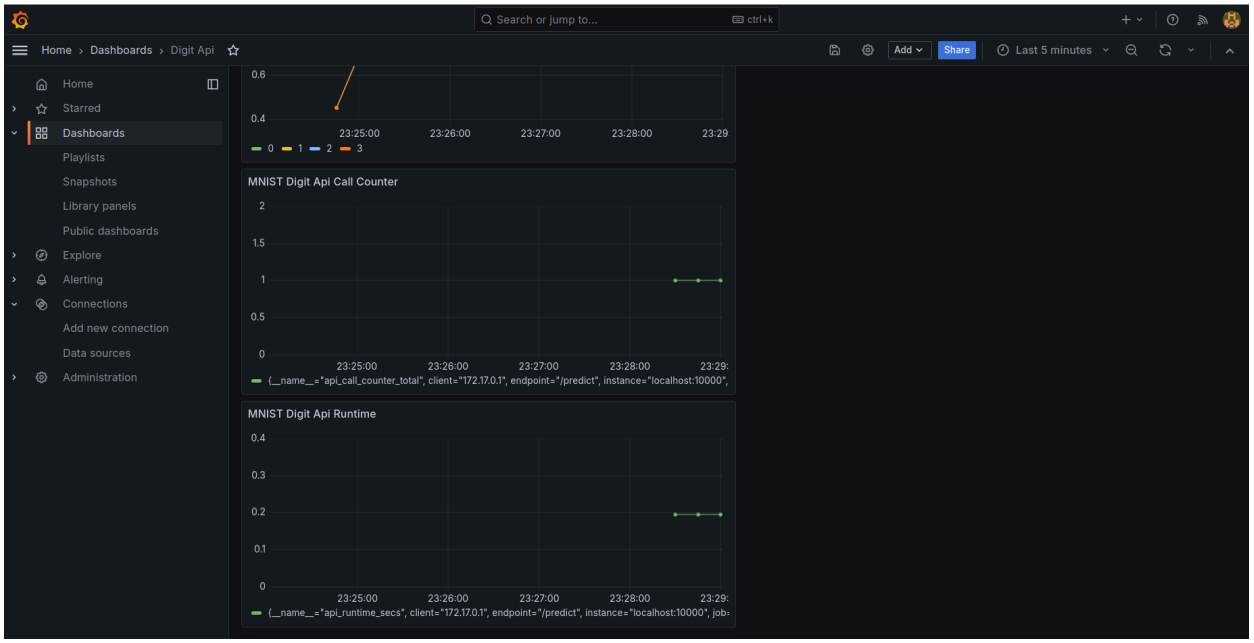
# Task 2 - Dockerization of the Mnist API:

- A Dockerfile and requirements.txt were created for dockerization.
- The docker image was built using the following command:
  **sudo docker build -t mnist_prediction_api .**
- The following command was used to run the docker container:

```
saketh@saketh-IdeaPad-3-14IML05-U1a:~/Desktop/assignment7$ sudo docker run -d -p 8000:8000 -p 10000:10000 mnist_prediction_api
[sudo] password for saketh:
ca644db1a4105a9b9068b6cf08eef67dc747854ce3d4c53b309eb060f72f2a43
saketh@saketh-IdeaPad-3-14IML05-U1a:~/Desktop/assignment7$ sudo docker ps
CONTAINER ID   IMAGE                  COMMAND                  CREATED             STATUS             PORTS
                    NAMES
ca644db1a410   mnist_prediction_api   "python3 mnist_api.p…"   10 seconds ago      Up 9 seconds       0.0.0.0:8000->8000/tcp, :::8000->8000/tcp, 0.0.0.0:10000->10000/
tcp, :::10000->10000/tcp   infallible_pare
7d43f1e05e48   mnist_prediction_api   "python3 mnist_api.p…"   About an hour ago   Up About an hour   0.0.0.0:9000->8000/tcp, :::9000->8000/tcp, 0.0.0.0:11000->10000/
tcp, :::11000->10000/tcp   gallant_hopper
saketh@saketh-IdeaPad-3-14IML05-U1a:~/Desktop/assignment7$
```

- The ports were mapped from the host to the container.
- The following images show the grafana dashboard (with docker) which contains the visualizations of the metrics:






- The new client ip is 172.17.0.1 which is different from 127.0.0.1 . This confirms that the app is running  docker container

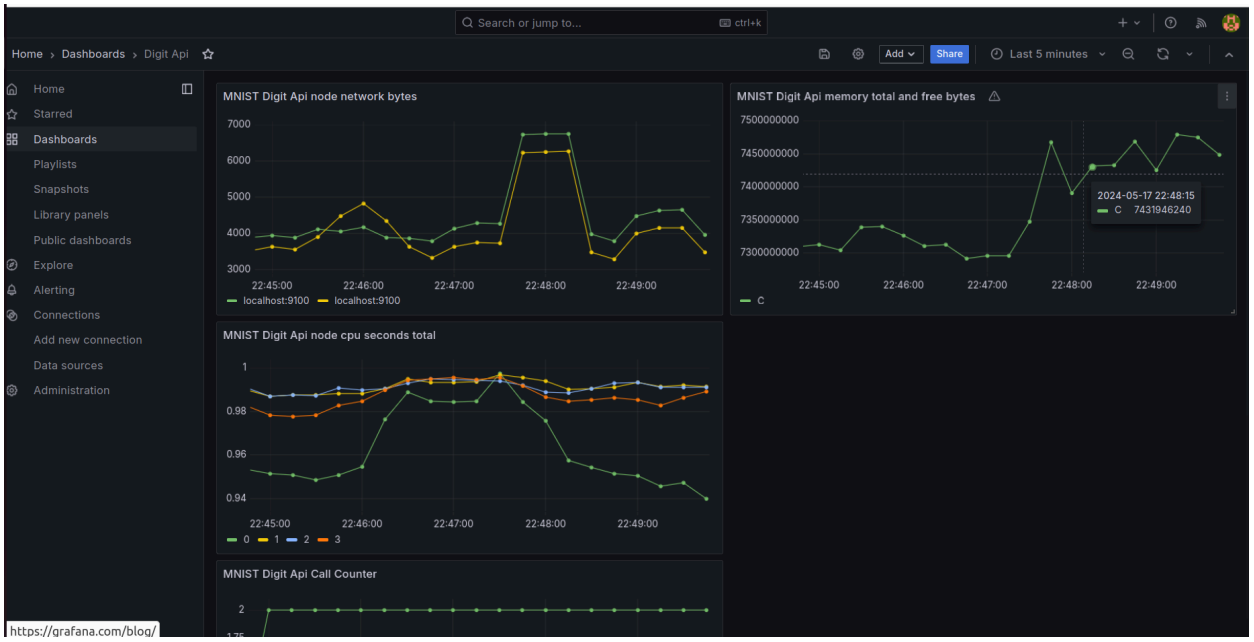
Running Docker Cluster:

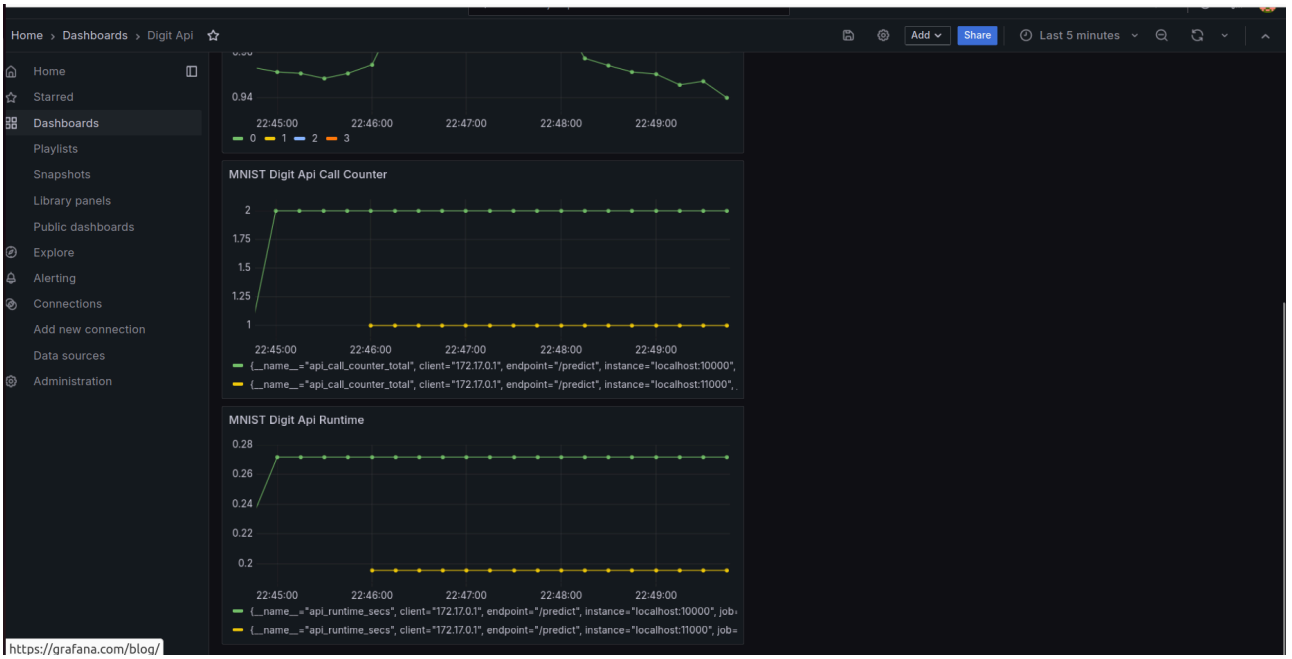
- The following commands were used to run multiple docker containers and obtain a cluster of FastAPI servers:

```
saketh@saketh-IdeaPad-3-14IML05-U1a:~/Desktop/assignment7$ sudo docker run -d -p 8000:8000 -p 10000:10000 --cpus 1 mnist_prediction_api
20d0d156c79d3e5f64fe77c286d93bf555662b25d9765ac2d115e3cea37
saketh@saketh-IdeaPad-3-14IML05-U1a:~/Desktop/assignment7$ sudo docker ps
CONTAINER ID   IMAGE                 COMMAND                  CREATED          STATUS          PORTS
                NAMES
21a3820d0d15   mnist_prediction_api   "python3 mnist_api.p…"   14 seconds ago   Up 13 seconds   0.0.0.0:8000->8000/tcp, :::8000->8000/tcp, 0.0.0.0:10000->10000/
::10000->10000/tcp   beautiful_ptolemy
saketh@saketh-IdeaPad-3-14IML05-U1a:~/Desktop/assignment7$ sudo docker run -d -p 9000:8000 -p 11000:10000 --cpus 1 mnist_prediction_api
7d43f1e05e484a73716be1ed2813a3ba03d057ce777758b373ea1fbb3699e616
```

- The FastApi server is mapped to different ports to get a cluster of servers
- Modified the prometheus.yml file to include metrics from different servers.
- The configuration file was named **prometheus_cluster.yml**

- The following images show the grafana dashboard (with multiple docker containers) which contains the visualizations of the metrics:




- Here, the client 172.17.0.1 has two different instances from ports 10000(8000) and 11000(9000)

# GitHub Repo:

https://github.com/Saketh-D/BDL-Assignment-7.git