USABILITY DOCUMENT

A step-by-step guide to use the NSSF on your system.

Technologies Required:

Docker

Docker is a software platform that allows you to build, test, and deploy applications quickly. Docker packages software into standardized units called containers that have everything the software needs to run including libraries, system tools, code, and runtime.

Follow the link to install Docker Desktop on your system.

Grafana

Grafana is an open-source interactive data-visualization platform, developed by Grafana Labs, which allows users to see their data via charts and graphs that are unified into one dashboard (or multiple dashboards!) for easier interpretation and understanding.

Follow the link to install Grafana Desktop on your system.

Flask

Flask is used for developing web applications using python, implemented on Werkzeug and Jinja2. Advantages of using Flask framework are built-in development server and a fast debugger provided.

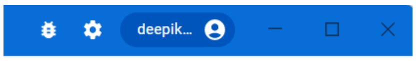
Follow the link to install Flask on your system.

After installing the above applications, Follow the steps given below to avail your services and monitor traffic:

STEP 1: Docker Desktop

Open Docker Desktop and make sure it is logged in and stays connected.

You can ensure it is logged in if your Docker Hub username is displayed at the top right corner:



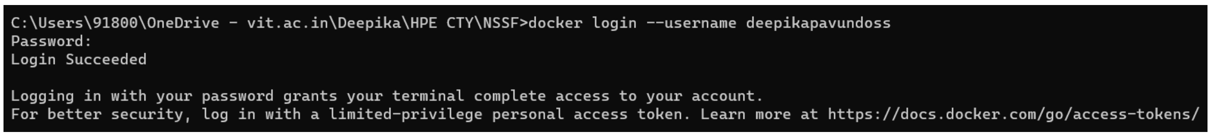
STEP 2: In CMD

Open Command Prompt.

To be able to pull the images from Docker Hub, firstly login to you Docker Hub account on your terminal using the following:

```
docker login --username <username>
```

It prompts you to enter your password, enter your password to login.

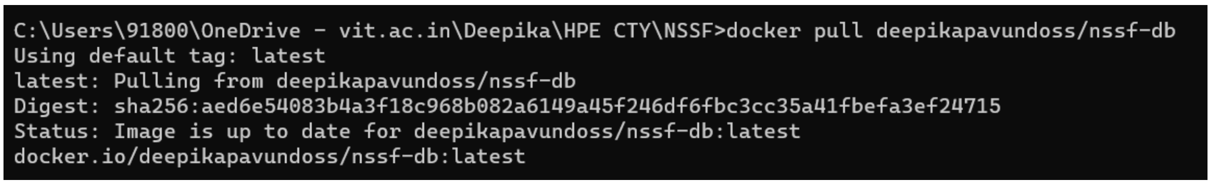

```
C:\Users\91800\OneDrive - vit.ac.in\Deepika\HPE CTY\NSSF>docker login --username deepikapavundoss
Password:
Login Succeeded

Logging in with your password grants your terminal complete access to your account.
For better security, log in with a limited-privilege personal access token. Learn more at https://docs.docker.com/go/access-tokens/
```

STEP 3: Pulling Images

Pull the images from Docker Hub by following:

```
docker pull deepikapavundoss/nssf-db

docker pull deepikapavundoss/nssf-host

docker pull deepikapavundoss/nssf-request

docker pull deepikapavundoss/nssf-automate
```

You will get similar output on doing so:


```
C:\Users\91800\OneDrive - vit.ac.in\Deepika\HPE CTY\NSSF>docker pull deepikapavundoss/nssf-db
Using default tag: latest
latest: Pulling from deepikapavundoss/nssf-db
Digest: sha256:aed6e54083b4a3f18c968b082a6149a45f246df6fbc3cc35a41fbefa3ef24715
Status: Image is up to date for deepikapavundoss/nssf-db:latest
docker.io/deepikapavundoss/nssf-db:latest
```

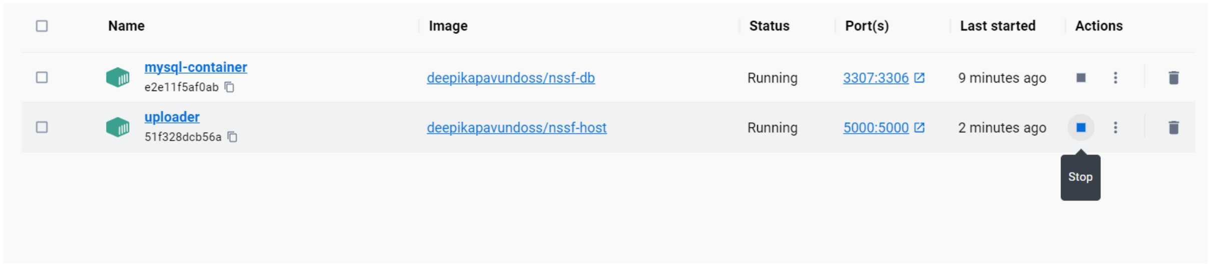
STEP 4: Create Network

Run the pulled docker images:

Firstly, we create a network, "mynetwork" so that all the images we'll be running can stay connected and communicate among them:

```
docker network create mynetwork
```

If the images are already running, you can stop them by clicking at the stop button beside each of the containers in the container section of Docker Desktop:



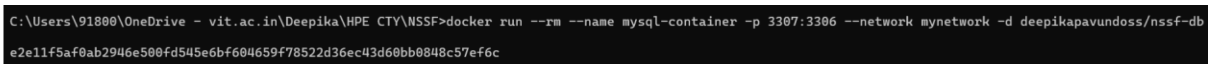
☐	Name	Image	Status	Port(s)	Last started	Actions
☐	mysql-container e2e11f5af0ab	deepikapavundoss/nssf-db	Running	3307:3306	9 minutes ago	■ ⋮ 🗑
☐	uploader 51f328dcb56a	deepikapavundoss/nssf-host	Running	5000:5000	2 minutes ago	■ ⋮ 🗑

STEP 5:

Then, we run the image for SQL database by creating a container for it, specifying the port on which it will run and subsequently running it:

```
docker run --rm --name mysql-container -p 3307:3306 --network mynetwork -d
deepikapavundoss/nssf-db
```

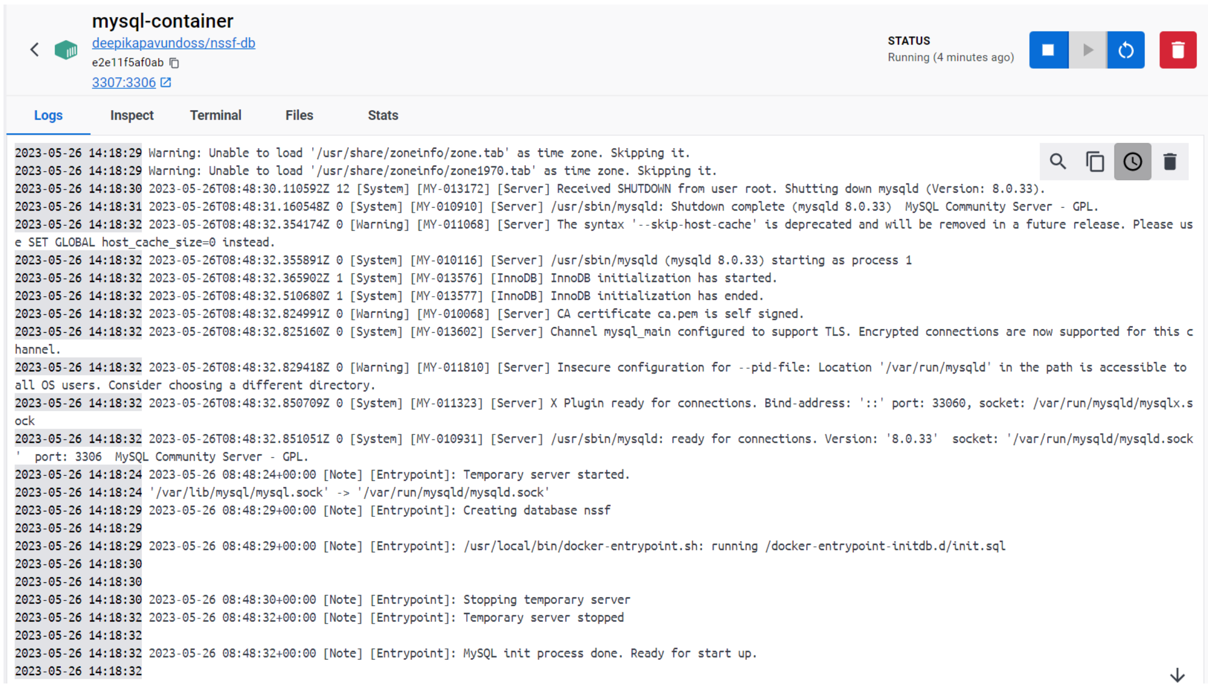
A similar output is displayed where the output displayed is basically the Container ID of the container created



```
C:\Users\91800\OneDrive - vit.ac.in\Deepika\HPE CTY\NSSF>docker run --rm --name mysql-container -p 3307:3306 --network mynetwork -d deepikapavundoss/nssf-db
e2e11f5af0ab2946e500fd545e6bf604659f78522d36ec43d60bb0848c57ef6c
```

STEP 6:

Wait for some time, 30 seconds or so, till the mysql-container is fully running. On clicking on the container in Docker Desktop, you'll get to see the logs. If you see a similar log, you're good to go to the next step:



```
2023-05-26 14:18:29 Warning: Unable to load '/usr/share/zoneinfo/zone.tab' as time zone. Skipping it.
2023-05-26 14:18:29 Warning: Unable to load '/usr/share/zoneinfo/zone1970.tab' as time zone. Skipping it.
2023-05-26 14:18:30 2023-05-26T08:48:30.110592Z 12 [System] [MY-013172] [Server] Received SHUTDOWN from user root. Shutting down mysqld (Version: 8.0.33).
2023-05-26 14:18:31 2023-05-26T08:48:31.160548Z 0 [System] [MY-010910] [Server] /usr/sbin/mysqld: Shutdown complete (mysqld 8.0.33)  MySQL Community Server - GPL.
2023-05-26 14:18:32 2023-05-26T08:48:32.354174Z 0 [Warning] [MY-011068] [Server] The syntax '--skip-host-cache' is deprecated and will be removed in a future release. Please us
e SET GLOBAL host_cache_size=0 instead.
2023-05-26 14:18:32 2023-05-26T08:48:32.355891Z 0 [System] [MY-010116] [Server] /usr/sbin/mysqld (mysqld 8.0.33) starting as process 1
2023-05-26 14:18:32 2023-05-26T08:48:32.365902Z 1 [System] [MY-013576] [InnoDB] InnoDB initialization has started.
2023-05-26 14:18:32 2023-05-26T08:48:32.510680Z 1 [System] [MY-013577] [InnoDB] InnoDB initialization has ended.
2023-05-26 14:18:32 2023-05-26T08:48:32.824991Z 0 [Warning] [MY-010068] [Server] CA certificate ca.pem is self signed.
2023-05-26 14:18:32 2023-05-26T08:48:32.825160Z 0 [System] [MY-013602] [Server] Channel mysql_main configured to support TLS. Encrypted connections are now supported for this c
hannel.
2023-05-26 14:18:32 2023-05-26T08:48:32.829418Z 0 [Warning] [MY-011810] [Server] Insecure configuration for --pid-file: Location '/var/run/mysqld' in the path is accessible to
all OS users. Consider choosing a different directory.
2023-05-26 14:18:32 2023-05-26T08:48:32.850709Z 0 [System] [MY-011323] [Server] X Plugin ready for connections. Bind-address: '::' port: 33060, socket: /var/run/mysqld/mysqlx.s
ock
2023-05-26 14:18:32 2023-05-26T08:48:32.851051Z 0 [System] [MY-010931] [Server] /usr/sbin/mysqld: ready for connections. Version: '8.0.33'  socket: '/var/run/mysqld/mysqld.sock
'  port: 3306  MySQL Community Server - GPL.
2023-05-26 14:18:24 2023-05-26 08:48:24+00:00 [Note] [Entrypoint]: Temporary server started.
2023-05-26 14:18:24 '/var/lib/mysql/mysql.sock' -> '/var/run/mysqld/mysqld.sock'
2023-05-26 14:18:29 2023-05-26 08:48:29+00:00 [Note] [Entrypoint]: Creating database nssf
2023-05-26 14:18:29
2023-05-26 14:18:29 2023-05-26 08:48:29+00:00 [Note] [Entrypoint]: /usr/local/bin/docker-entrypoint.sh: running /docker-entrypoint-initdb.d/init.sql
2023-05-26 14:18:30
2023-05-26 14:18:30
2023-05-26 14:18:30 2023-05-26 08:48:30+00:00 [Note] [Entrypoint]: Stopping temporary server
2023-05-26 14:18:32 2023-05-26 08:48:32+00:00 [Note] [Entrypoint]: Temporary server stopped
2023-05-26 14:18:32
2023-05-26 14:18:32 2023-05-26 08:48:32+00:00 [Note] [Entrypoint]: MySQL init process done. Ready for start up.
2023-05-26 14:18:32
```

STEP 7: Run Image for Flask Application

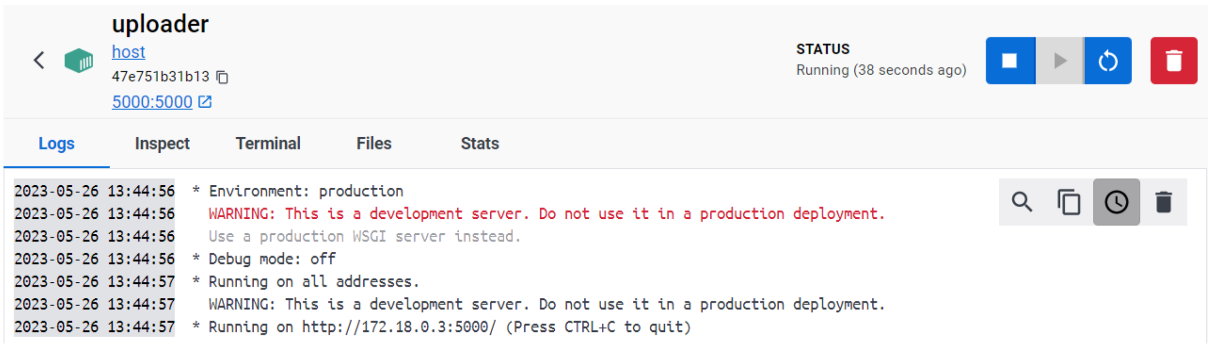
Run the next image by running the following on your terminal:

```
docker run -it --rm -d --name uploader --network mynetwork -p 5000:5000
deepikapavundoss/nssf-host
```

On running the above, the Container ID of the newly created Uploader container is displayed.

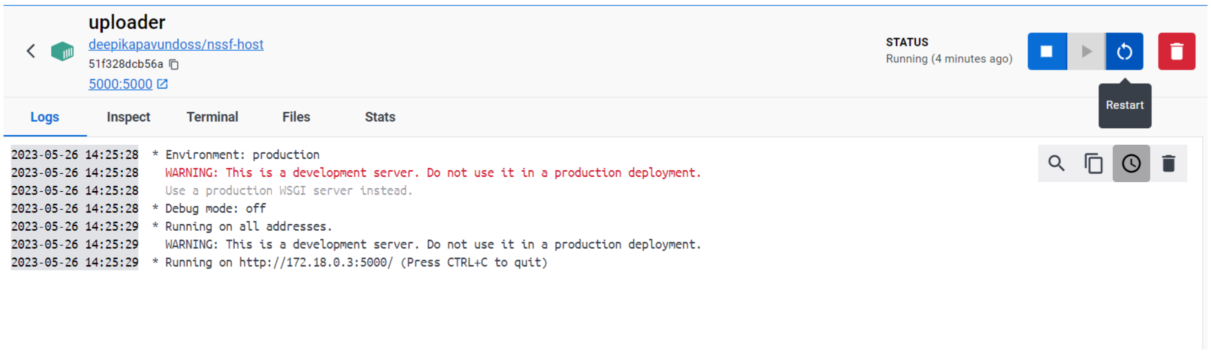
The flask Application (hosted for internal purposes) runs on localhost on port 5000,

- Make sure the Uploader is running well by checking its logs, if you find similar log, then you're good to go to the next step.



```
2023-05-26 13:44:56  * Environment: production
2023-05-26 13:44:56    WARNING: This is a development server. Do not use it in a production deployment.
2023-05-26 13:44:56    Use a production WSGI server instead.
2023-05-26 13:44:56  * Debug mode: off
2023-05-26 13:44:57  * Running on all addresses.
2023-05-26 13:44:57    WARNING: This is a development server. Do not use it in a production deployment.
2023-05-26 13:44:57  * Running on http://172.18.0.3:5000/ (Press CTRL+C to quit)
```

- If you find some error in the logs, go to the Images section of Docker Desktop, make sure the mysql-container is fully running, again click the Uploader image and you'll find the button to restart on the top right corner. Restart the Uploader container:



```
2023-05-26 14:25:28  * Environment: production
2023-05-26 14:25:28    WARNING: This is a development server. Do not use it in a production deployment.
2023-05-26 14:25:28    Use a production WSGI server instead.
2023-05-26 14:25:28  * Debug mode: off
2023-05-26 14:25:29  * Running on all addresses.
2023-05-26 14:25:29    WARNING: This is a development server. Do not use it in a production deployment.
2023-05-26 14:25:29  * Running on http://172.18.0.3:5000/ (Press CTRL+C to quit)
```

<u>STEP 8: Run the next Images to obtain user input.</u>

The application can be run in 2 ways:

Case 1. The user can give manual input by entering his user ID and application type for which he wants slice to be allocated.

Case 2. Or simply check the Output to see the Slices allocated for a particular set of Inputs.

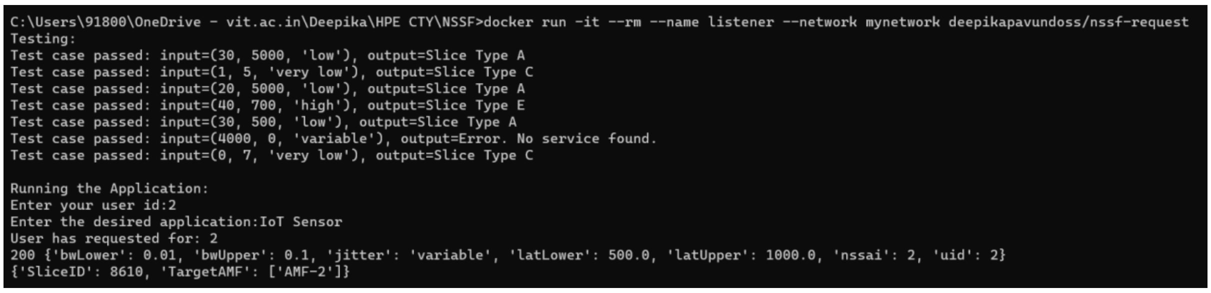
For Case 1: Stay in the same Step.

For Case 2: Proceed to next Step (Step 9)

The image for Manual input must be run:

```
docker run -it --rm --name listener --network mynetwork deepikapavundoss/nssf-request
```

It will first display the testing details and then ask for User ID and Application desired for Slice Allocation in a similar way:



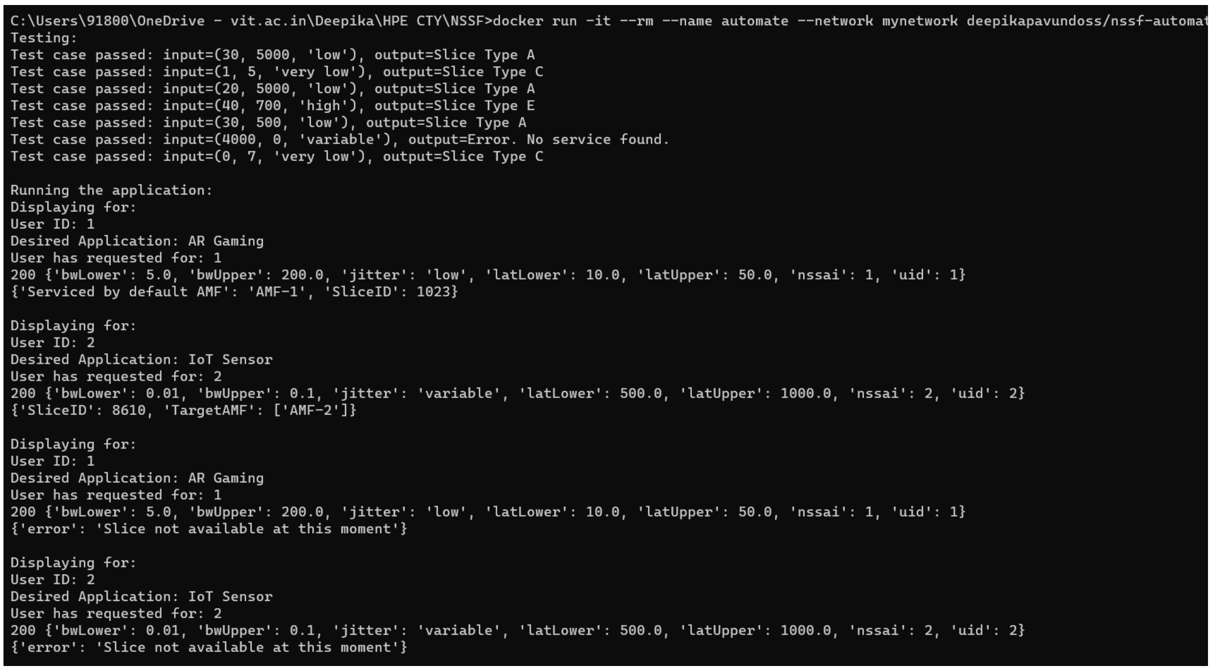
```
C:\Users\91800\OneDrive - vit.ac.in\Deepika\HPE CTY\NSSF>docker run -it --rm --name listener --network mynetwork deepikapavundoss/nssf-request
Testing:
Test case passed: input=(30, 5000, 'low'), output=Slice Type A
Test case passed: input=(1, 5, 'very low'), output=Slice Type C
Test case passed: input=(20, 5000, 'low'), output=Slice Type A
Test case passed: input=(40, 700, 'high'), output=Slice Type E
Test case passed: input=(30, 500, 'low'), output=Slice Type A
Test case passed: input=(4000, 0, 'variable'), output=Error. No service found.
Test case passed: input=(0, 7, 'very low'), output=Slice Type C

Running the Application:
Enter your user id:2
Enter the desired application:IoT Sensor
User has requested for: 2
200 {'bwLower': 0.01, 'bwUpper': 0.1, 'jitter': 'variable', 'latLower': 500.0, 'latUpper': 1000.0, 'nssai': 2, 'uid': 2}
{'SliceID': 8610, 'TargetAMF': ['AMF-2']}
```

<u>STEP 9:</u>

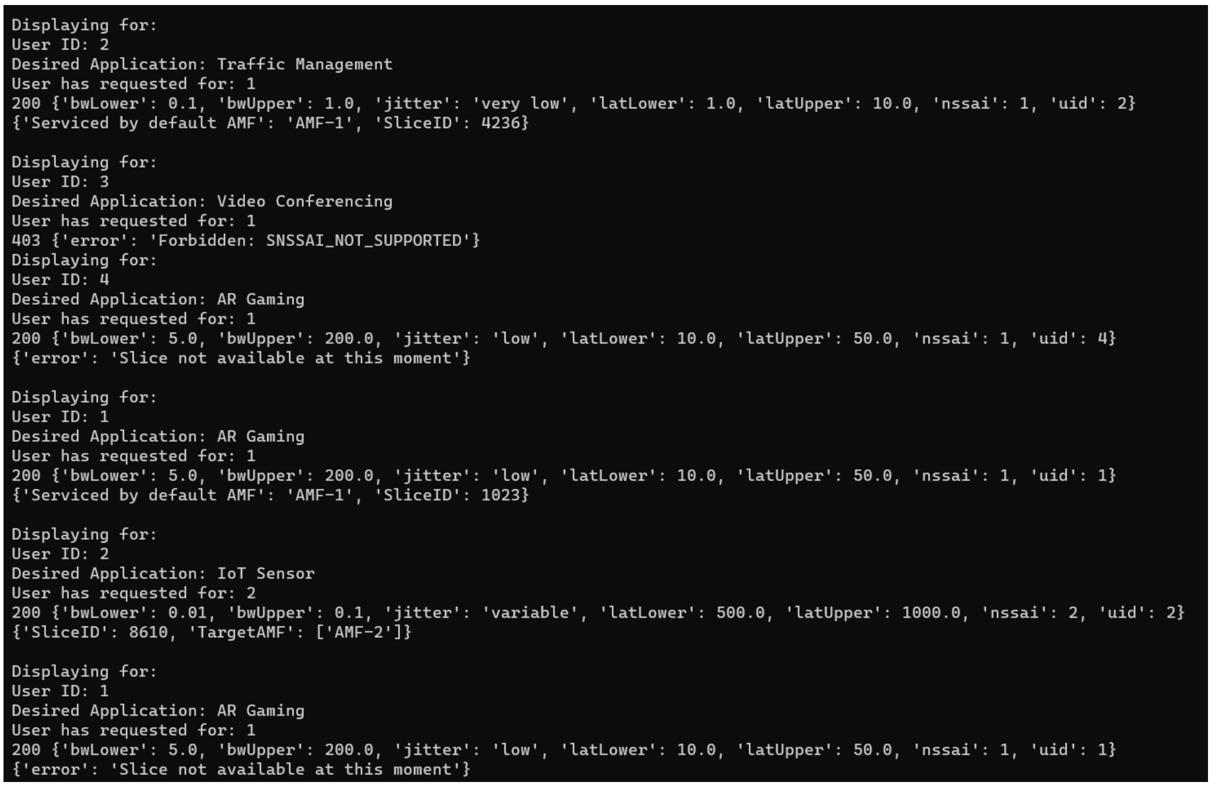
The image for Automated input has to be run:

```
docker run -it --rm --name automate --network mynetwork deepikapavundoss/nssf-automate
```

No user interaction is needed and a similar output is displayed:



```
C:\Users\91800\OneDrive - vit.ac.in\Deepika\HPE CTY\NSSF>docker run -it --rm --name automate --network mynetwork deepikapavundoss/nssf-automat
Testing:
Test case passed: input=(30, 5000, 'low'), output=Slice Type A
Test case passed: input=(1, 5, 'very low'), output=Slice Type C
Test case passed: input=(20, 5000, 'low'), output=Slice Type A
Test case passed: input=(40, 700, 'high'), output=Slice Type E
Test case passed: input=(30, 500, 'low'), output=Slice Type A
Test case passed: input=(4000, 0, 'variable'), output=Error. No service found.
Test case passed: input=(0, 7, 'very low'), output=Slice Type C

Running the application:
Displaying for:
User ID: 1
Desired Application: AR Gaming
User has requested for: 1
200 {'bwLower': 5.0, 'bwUpper': 200.0, 'jitter': 'low', 'latLower': 10.0, 'latUpper': 50.0, 'nssai': 1, 'uid': 1}
{'Serviced by default AMF': 'AMF-1', 'SliceID': 1023}

Displaying for:
User ID: 2
Desired Application: IoT Sensor
User has requested for: 2
200 {'bwLower': 0.01, 'bwUpper': 0.1, 'jitter': 'variable', 'latLower': 500.0, 'latUpper': 1000.0, 'nssai': 2, 'uid': 2}
{'SliceID': 8610, 'TargetAMF': ['AMF-2']}

Displaying for:
User ID: 1
Desired Application: AR Gaming
User has requested for: 1
200 {'bwLower': 5.0, 'bwUpper': 200.0, 'jitter': 'low', 'latLower': 10.0, 'latUpper': 50.0, 'nssai': 1, 'uid': 1}
{'error': 'Slice not available at this moment'}

Displaying for:
User ID: 2
Desired Application: IoT Sensor
User has requested for: 2
200 {'bwLower': 0.01, 'bwUpper': 0.1, 'jitter': 'variable', 'latLower': 500.0, 'latUpper': 1000.0, 'nssai': 2, 'uid': 2}
{'error': 'Slice not available at this moment'}
```



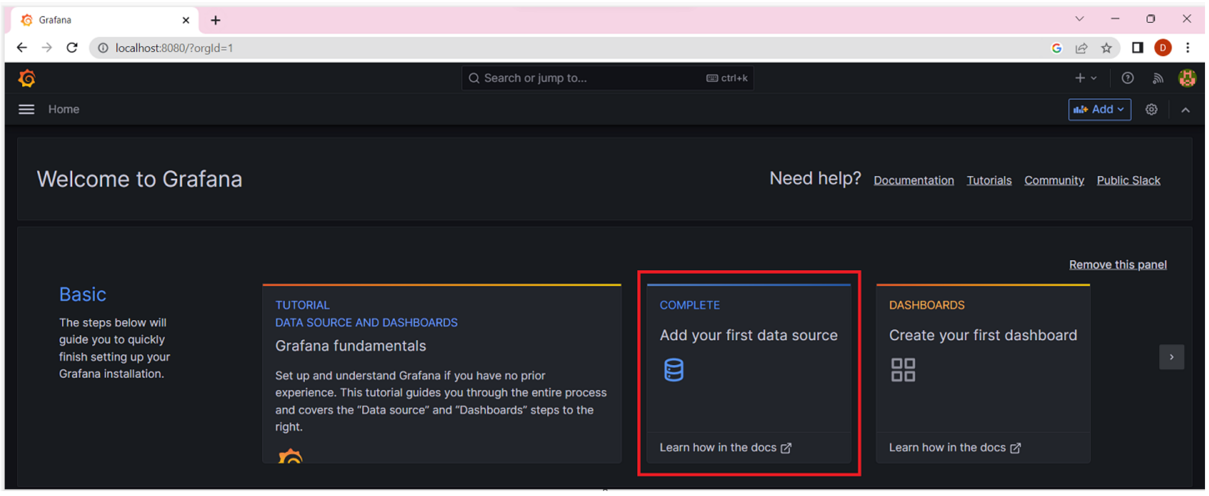
```
Displaying for:
User ID: 2
Desired Application: Traffic Management
User has requested for: 1
200 {'bwLower': 0.1, 'bwUpper': 1.0, 'jitter': 'very low', 'latLower': 1.0, 'latUpper': 10.0, 'nssai': 1, 'uid': 2}
{'Serviced by default AMF': 'AMF-1', 'SliceID': 4236}

Displaying for:
User ID: 3
Desired Application: Video Conferencing
User has requested for: 1
403 {'error': 'Forbidden: SNSSAI_NOT_SUPPORTED'}
Displaying for:
User ID: 4
Desired Application: AR Gaming
User has requested for: 1
200 {'bwLower': 5.0, 'bwUpper': 200.0, 'jitter': 'low', 'latLower': 10.0, 'latUpper': 50.0, 'nssai': 1, 'uid': 4}
{'error': 'Slice not available at this moment'}

Displaying for:
User ID: 1
Desired Application: AR Gaming
User has requested for: 1
200 {'bwLower': 5.0, 'bwUpper': 200.0, 'jitter': 'low', 'latLower': 10.0, 'latUpper': 50.0, 'nssai': 1, 'uid': 1}
{'Serviced by default AMF': 'AMF-1', 'SliceID': 1023}

Displaying for:
User ID: 2
Desired Application: IoT Sensor
User has requested for: 2
200 {'bwLower': 0.01, 'bwUpper': 0.1, 'jitter': 'variable', 'latLower': 500.0, 'latUpper': 1000.0, 'nssai': 2, 'uid': 2}
{'SliceID': 8610, 'TargetAMF': ['AMF-2']}

Displaying for:
User ID: 1
Desired Application: AR Gaming
User has requested for: 1
200 {'bwLower': 5.0, 'bwUpper': 200.0, 'jitter': 'low', 'latLower': 10.0, 'latUpper': 50.0, 'nssai': 1, 'uid': 1}
{'error': 'Slice not available at this moment'}
```

The container keeps on running unless you stop in Docker Desktop.

STEP 10: Visualizing the N22 Network Traffic on Grafana.

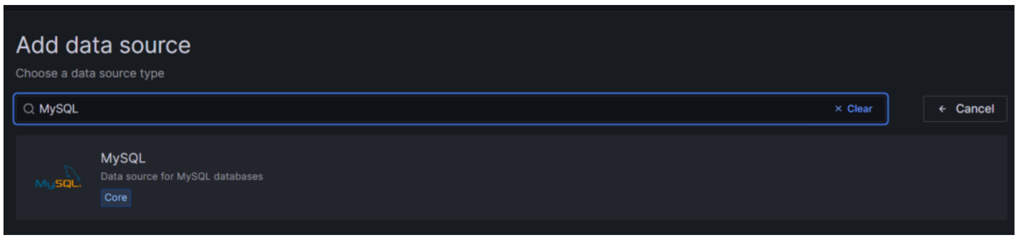
To visualize the network traffic of N22 interface, follow the following steps:

1. If you do not have an account for Grafana, Register yourself on Grafana Labs.
2. Make sure you have Grafana Desktop installed on your system.
3. Open http://localhost:3000/ on your web browser which is default http port for Grafana labs.
4. If the above doesn't work, try for http://localhost:8080/
5. Create a Grafana Data source first by clicking on the following:



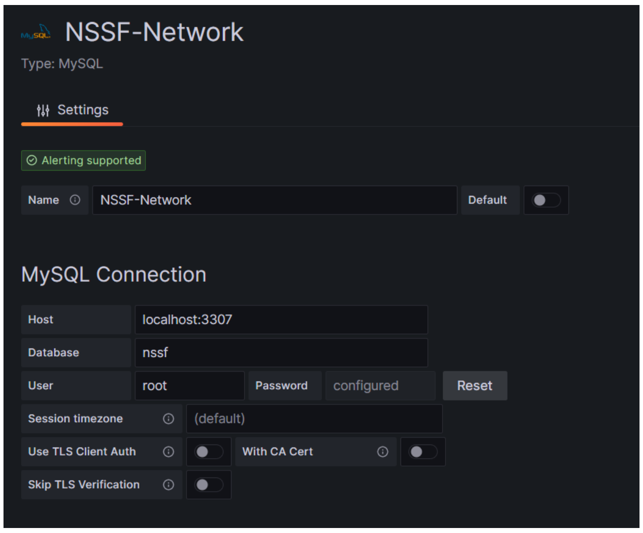
Grafana
localhost:8080/?orgId=1
Search or jump to...
Home
Add
Welcome to Grafana
Need help? Documentation Tutorials Community Public Slack
Remove this panel
Basic
The steps below will guide you to quickly finish setting up your Grafana installation.
TUTORIAL
DATA SOURCE AND DASHBOARDS
Grafana fundamentals
Set up and understand Grafana if you have no prior experience. This tutorial guides you through the entire process and covers the "Data source" and "Dashboards" steps to the right.
COMPLETE
Add your first data source
Learn how in the docs
DASHBOARDS
Create your first dashboard
Learn how in the docs

6. Search for MySQL and click on it to create SQL data source:



7. Enter the following details:
   Name: anything of your choice (here: NSSF-Network)
   host: localhost:3307
   Database: nssf
   User: root
   Password: password

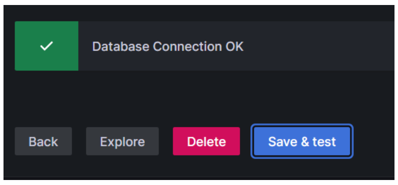
   The setup looks as follows:


NSSF-Network
Type: MySQL
Settings
Alerting supported
Name: NSSF-Network
Default

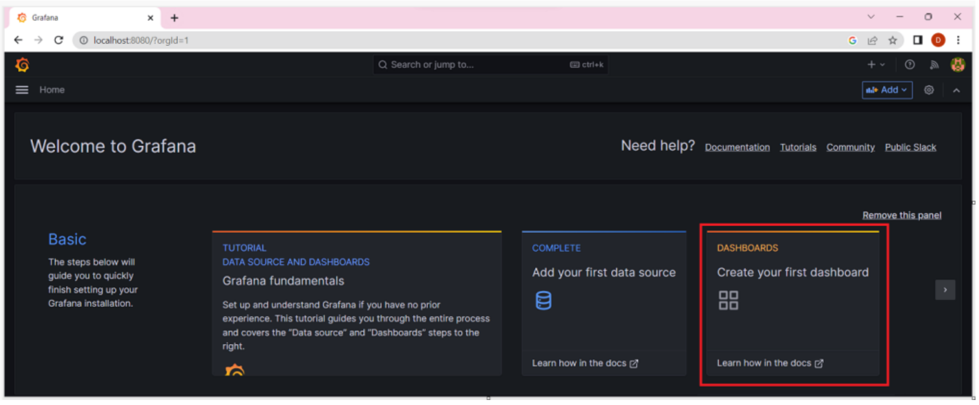
MySQL Connection

Host	localhost:3307
Database	nssf
User	root
Password	configured
Session timezone	(default)
Use TLS Client Auth	off
With CA Cert	off
Skip TLS Verification	off

8. Click on "Save & Test" to make sure it works perfectly:

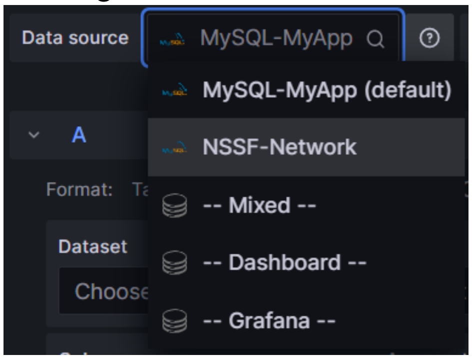

Database Connection OK

9. Go back to Home to create Dashboard by clicking on the following:

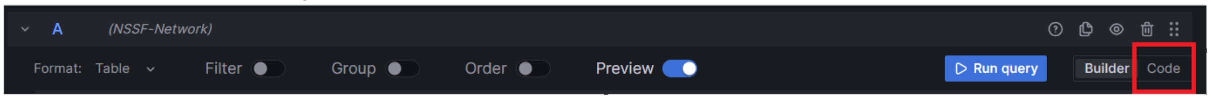


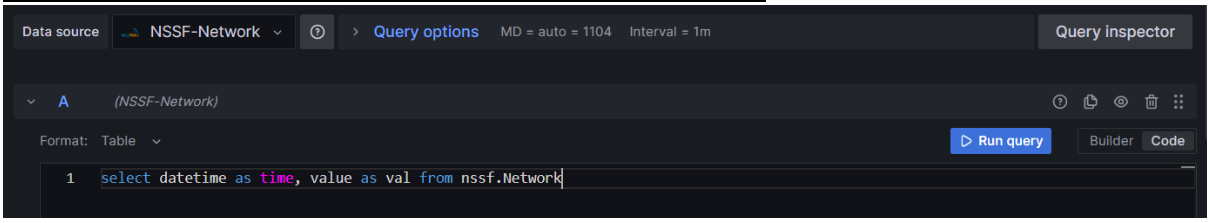
Welcome to Grafana
Need help? Documentation Tutorials Community Public Slack
Remove this panel
Basic
The steps below will guide you to quickly finish setting up your Grafana installation.
TUTORIAL
DATA SOURCE AND DASHBOARDS
Grafana fundamentals
Set up and understand Grafana if you have no prior experience. This tutorial guides you through the entire process and covers the "Data source" and "Dashboards" steps to the right.
COMPLETE
Add your first data source
Learn how in the docs
DASHBOARDS
Create your first dashboard
Learn how in the docs

10. Click on "Add Visualization"
11. In the Data source option right below the panel, click on the Data source created a while ago:

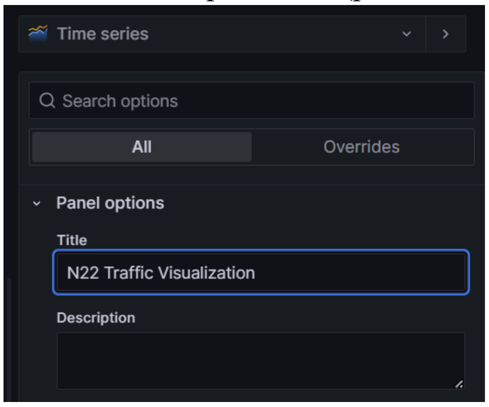


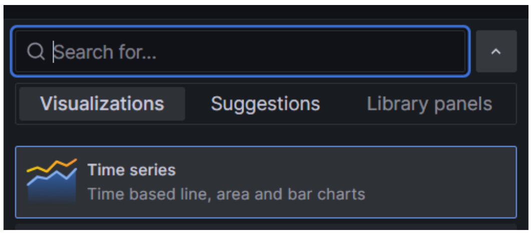
12. Click on code button right below:

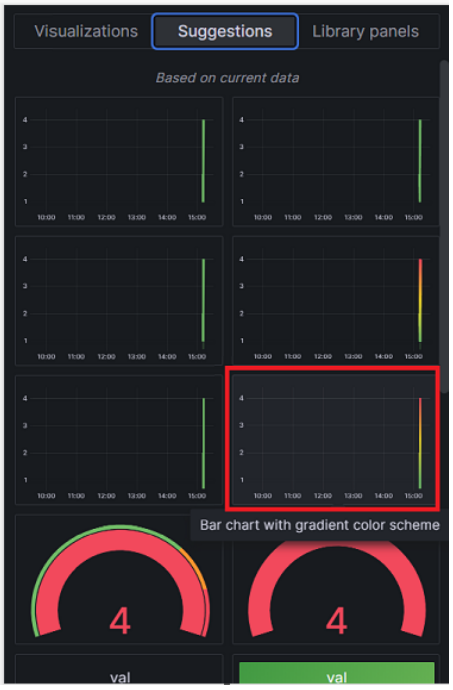


13. Enter the following SQL query and the click on “Run query”:
select datetime as time, value as UserID from nssf.Network

Data source NSSF-Network Query options MD = auto = 1104 Interval = 1m Query inspector
A (NSSF-Network)
Format: Table Run query Builder Code

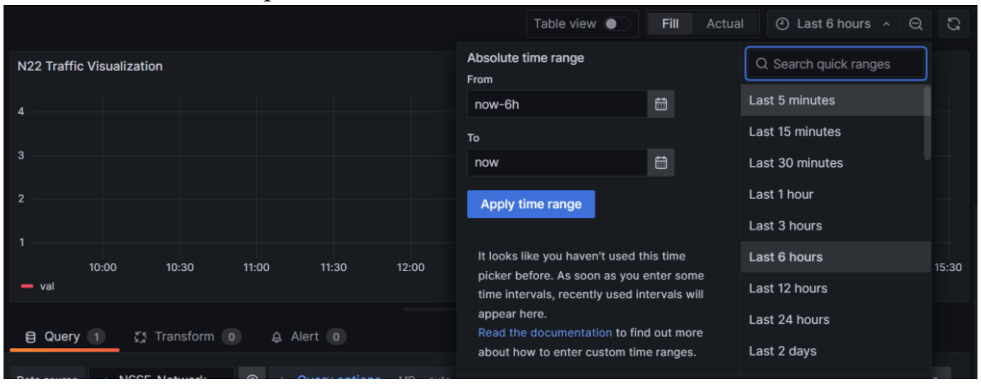
```
select datetime as time, value as val from nssf.Network
```

14. You can set the panel title (present on right) as follows:

Time series
Search options
All Overrides
Panel options
Title: N22 Traffic Visualization
Description

15. Right above “Panel Options” exist a drop-down menu with “Time series” on top. For better Visualization of data click on it and click on “suggestions”:


Click on the following – “Bar chart with gradient colour scheme”


16. Select “Last 5 minutes” in the drop down menu right above the Panel. The menu has “Last 6 hours” on top:



Absolute time range

From

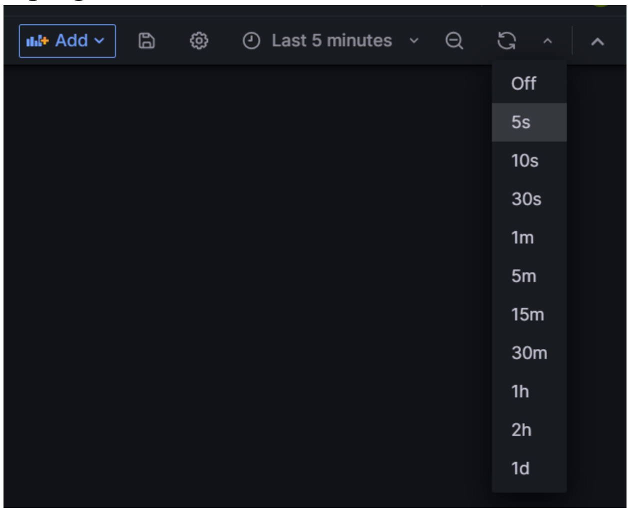
now-6h

To

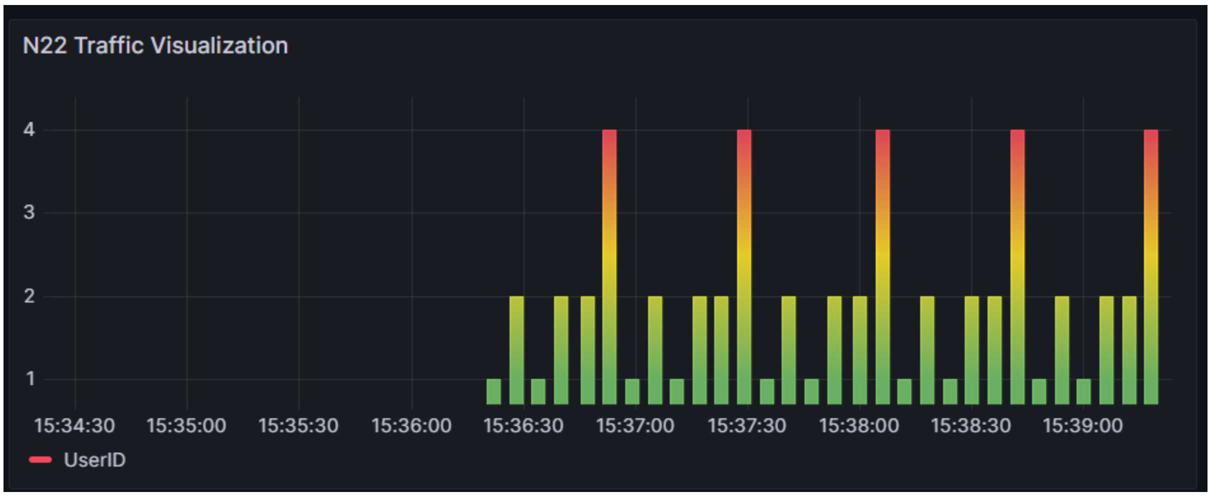
now

It looks like you haven't used this time picker before. As soon as you enter some time intervals, recently used intervals will appear here. Read the documentation to find out more about how to enter custom time ranges.

17. Click on “apply” on top right corner.
18. To get live visualization, click on “5s” in the following drop down menu present at top right corner:

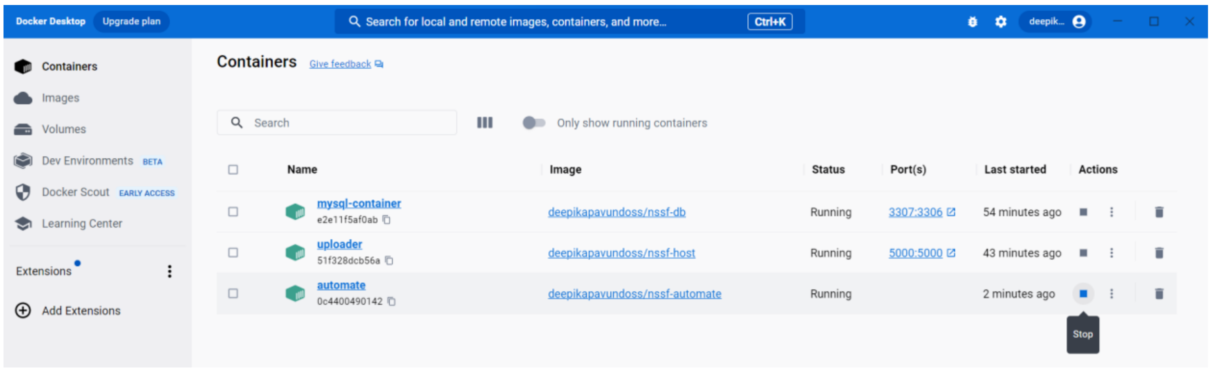


19. You’re finally done with Grafana setup and you’ll get to such Visualization of traffic on N22 interface. Each bar represents a communication via the N22 interface. The X-axis represents the timestamp of each such traffic and the Y-axis represents the UserID of an authorized User, who is requesting for the slice allocation.



STEP 11: Stopping the Containers

To stop the execution of each of the containers, click on Stop button for the container in Docker Desktop. On stopping them, they'll automatically get deleted.



Name	Image	Status	Port(s)	Last started
mysql-container e2e11f5af0ab	deepikapavundoss/nssf-db	Running	3307:3306	54 minutes ago
uploader 51f328dcb56a	deepikapavundoss/nssf-host	Running	5000:5000	43 minutes ago
automate 0c4400490142	deepikapavundoss/nssf-automate	Running		2 minutes ago

Thank You!