Dokumentacja AI na dzień 2023-08-02

Wszystkie kody są wystawione na repozytorium:

<https://github.com/EOSC-AI4PHENO/AI4PhenoEOSC>

Serwer: 10.0.20.50

################# URUCHAMIANIE USŁUG #########

Cały deployment jest umieszczony: AI4PhenoEOSC\DeploymentFastAPICeleryRabbitMQ

Na ssh wchodzimy do AI4PhenoEOSC\DeploymentFastAPICeleryRabbitMQ

I wykonujemy:

docker-compose build # uwaga może to trwać 15 mins

potem:

docker-compose up

powinny wstać usługi: FastAPI, RABBITMQ, CELERY, REDIS

póki co Nvidia triton uruchamiamy ręcznie:

wszystkie modele powinne zostać zapisane w folderze:

AI4PhenoEOSC\DeploymentTriton\models

docker run --gpus=all --rm -p8000:8000 -p8001:8001 -p8002:8002 -v /home/kurekj/AI4PhenoEOSC/DeploymentTriton/models:/models nvcr.io/nvidia/tritonserver:23.06-py3 tritonserver --model-repository=/models --metrics-port=8002

Do uruchomienia Prometheus:

docker run -p 9090:9090 prom/Prometheus

docker cp /home/kurekj/prometheus.yml 55cc6709a05c:/etc/prometheus/prometheus.yml

#docker cp 8d9fb7c79241:/etc/prometheus/prometheus.yml /home/kurekj/prometheus.yml

prometheus.yml:

# my global config

global:

scrape\_interval: 15s # Set the scrape interval to every 15 seconds. Default is every 1 minute.

evaluation\_interval: 15s # Evaluate rules every 15 seconds. The default is every 1 minute.

# scrape\_timeout is set to the global default (10s).

# Alertmanager configuration

alerting:

alertmanagers:

- static\_configs:

- targets:

# - alertmanager:9093

# Load rules once and periodically evaluate them according to the global 'evaluation\_interval'.

rule\_files:

# - "first\_rules.yml"

# - "second\_rules.yml"

# A scrape configuration containing exactly one endpoint to scrape:

# Here it's Prometheus itself.

scrape\_configs:

# The job name is added as a label `job=<job\_name>` to any timeseries scraped from this config.

- job\_name: "prometheus"

# metrics\_path defaults to '/metrics'

# scheme defaults to 'http'.

static\_configs:

- targets: ["localhost:9090"]

- job\_name: 'triton'

scrape\_interval: 5s

static\_configs:

- targets: ['10.0.20.50:8002']

Uruchamianie grafany:

docker run -d -p 3000:3000 grafana/Grafana

przykładowy promql dla grafany:

(avg(rate(nv\_inference\_request\_duration\_us[5m])) by (model) / avg(rate(nv\_inference\_count[5m])) by (model)) / 1000000

Środowisko wirtualne conda: environment\_input.yml:

name: environment\_input

channels:

- conda-forge

- defaults

dependencies:

- \_libgcc\_mutex=0.1=conda\_forge

- \_openmp\_mutex=4.5=2\_gnu

- amqp=5.1.1=pyhd8ed1ab\_0

- annotated-types=0.5.0=pyhd8ed1ab\_0

- aom=3.5.0=h27087fc\_0

- astral=3.2=py310hff52083\_1

- backports=1.0=pyhd8ed1ab\_3

- backports.functools\_lru\_cache=1.6.5=pyhd8ed1ab\_0

- backports.zoneinfo=0.2.1=py310hff52083\_7

- billiard=4.1.0=py310h2372a71\_0

- blosc=1.21.4=h0f2a231\_0

- brotli=1.0.9=h166bdaf\_9

- brotli-bin=1.0.9=h166bdaf\_9

- brotli-python=1.0.9=py310hd8f1fbe\_9

- brunsli=0.1=h9c3ff4c\_0

- bzip2=1.0.8=h7f98852\_4

- c-ares=1.19.1=hd590300\_0

- c-blosc2=2.10.0=hb4ffafa\_0

- ca-certificates=2023.7.22=hbcca054\_0

- celery=5.3.1=pyhd8ed1ab\_0

- cfitsio=4.2.0=hd9d235c\_0

- charls=2.4.2=h59595ed\_0

- charset-normalizer=3.2.0=pyhd8ed1ab\_0

- click=8.1.6=unix\_pyh707e725\_0

- click-didyoumean=0.3.0=pyhd8ed1ab\_0

- click-plugins=1.1.1=py\_0

- dav1d=1.2.1=hd590300\_0

- ephem=4.1.4=py310h1fa729e\_0

- freetype=2.12.1=hca18f0e\_1

- geos=3.11.2=hcb278e6\_0

- giflib=5.2.1=h0b41bf4\_3

- idna=3.4=pyhd8ed1ab\_0

- imagecodecs=2023.7.10=py310h4c4fb95\_0

- imageio=2.31.1=pyh24c5eb1\_0

- importlib-metadata=6.8.0=pyha770c72\_0

- jxrlib=1.1=h7f98852\_2

- keyutils=1.6.1=h166bdaf\_0

- kombu=5.3.1=py310hff52083\_0

- krb5=1.21.1=h659d440\_0

- lazy\_loader=0.2=pyhd8ed1ab\_0

- lcms2=2.15=haa2dc70\_1

- ld\_impl\_linux-64=2.40=h41732ed\_0

- lerc=4.0.0=h27087fc\_0

- libaec=1.0.6=hcb278e6\_1

- libavif=0.11.1=h8182462\_2

- libblas=3.9.0=17\_linux64\_openblas

- libbrotlicommon=1.0.9=h166bdaf\_9

- libbrotlidec=1.0.9=h166bdaf\_9

- libbrotlienc=1.0.9=h166bdaf\_9

- libcblas=3.9.0=17\_linux64\_openblas

- libcurl=8.2.0=hca28451\_0

- libdeflate=1.18=h0b41bf4\_0

- libedit=3.1.20191231=he28a2e2\_2

- libev=4.33=h516909a\_1

- libffi=3.4.2=h7f98852\_5

- libgcc-ng=13.1.0=he5830b7\_0

- libgfortran-ng=13.1.0=h69a702a\_0

- libgfortran5=13.1.0=h15d22d2\_0

- libgomp=13.1.0=he5830b7\_0

- libjpeg-turbo=2.1.5.1=h0b41bf4\_0

- liblapack=3.9.0=17\_linux64\_openblas

- libnghttp2=1.52.0=h61bc06f\_0

- libopenblas=0.3.23=pthreads\_h80387f5\_0

- libpng=1.6.39=h753d276\_0

- libsqlite=3.42.0=h2797004\_0

- libssh2=1.11.0=h0841786\_0

- libstdcxx-ng=13.1.0=hfd8a6a1\_0

- libtiff=4.5.1=h8b53f26\_0

- libuuid=1.41.5=h5eee18b\_0

- libwebp-base=1.3.1=hd590300\_0

- libxcb=1.15=h0b41bf4\_0

- libzlib=1.2.13=hd590300\_5

- libzopfli=1.0.3=h9c3ff4c\_0

- lz4-c=1.9.4=hcb278e6\_0

- mpmath=1.3.0=pyhd8ed1ab\_0

- msgpack-python=1.0.5=py310hdf3cbec\_0

- ncurses=6.4=hcb278e6\_0

- networkx=3.1=pyhd8ed1ab\_0

- openjpeg=2.5.0=hfec8fc6\_2

- openssl=3.1.1=hd590300\_1

- packaging=23.1=pyhd8ed1ab\_0

- pandas=2.0.3=py310h7cbd5c2\_1

- pillow=10.0.0=py310h582fbeb\_0

- pip=23.2.1=pyhd8ed1ab\_0

- platformdirs=3.9.1=pyhd8ed1ab\_0

- pooch=1.7.0=pyha770c72\_3

- prompt-toolkit=3.0.39=pyha770c72\_0

- prompt\_toolkit=3.0.39=hd8ed1ab\_0

- pthread-stubs=0.4=h36c2ea0\_1001

- pydantic=2.1.1=pyhd8ed1ab\_0

- pydantic-core=2.4.0=py310hcb5633a\_0

- pysocks=1.7.1=pyha2e5f31\_6

- python=3.10.12=h955ad1f\_0

- python-dateutil=2.8.2=pyhd8ed1ab\_0

- python-multipart=0.0.6=pyhd8ed1ab\_0

- python-rapidjson=1.10=py310heca2aa9\_0

- python-tzdata=2023.3=pyhd8ed1ab\_0

- python\_abi=3.10=2\_cp310

- pytz=2023.3=pyhd8ed1ab\_0

- pywavelets=1.4.1=py310h0a54255\_0

- readline=8.2=h8228510\_1

- requests=2.31.0=pyhd8ed1ab\_0

- scikit-image=0.21.0=py310hc6cd4ac\_0

- scipy=1.11.1=py310ha4c1d20\_0

- setuptools=68.0.0=pyhd8ed1ab\_0

- shapely=2.0.1=py310h056c13c\_1

- six=1.16.0=pyh6c4a22f\_0

- snappy=1.1.10=h9fff704\_0

- sqlite=3.42.0=h2c6b66d\_0

- sympy=1.12=pyh04b8f61\_3

- tifffile=2023.7.18=pyhd8ed1ab\_0

- tk=8.6.12=h27826a3\_0

- tritonclient=2.34.0=pyhca7485f\_0

- typing-extensions=4.7.1=hd8ed1ab\_0

- typing\_extensions=4.7.1=pyha770c72\_0

- tzdata=2023c=h71feb2d\_0

- vine=5.0.0=pyhd8ed1ab\_1

- wcwidth=0.2.6=pyhd8ed1ab\_0

- wheel=0.41.0=pyhd8ed1ab\_0

- xorg-libxau=1.0.11=hd590300\_0

- xorg-libxdmcp=1.1.3=h7f98852\_0

- xz=5.4.2=h5eee18b\_0

- zfp=1.0.0=h27087fc\_3

- zipp=3.16.2=pyhd8ed1ab\_0

- zlib=1.2.13=hd590300\_5

- zlib-ng=2.0.7=h0b41bf4\_0

- zstd=1.5.2=hfc55251\_7

- pip:

- absl-py==1.4.0

- anyio==3.7.1

- asgiref==3.7.2

- asttokens==2.2.1

- astunparse==1.6.3

- async-timeout==4.0.2

- attrs==23.1.0

- backcall==0.2.0

- build==0.10.0

- cachecontrol==0.12.14

- cachetools==5.3.1

- certifi==2023.7.22

- cffi==1.15.1

- cleo==2.0.1

- click-repl==0.3.0

- crashtest==0.4.1

- cryptography==41.0.2

- decorator==5.1.1

- deprecated==1.2.14

- distlib==0.3.7

- dnspython==2.4.0

- dulwich==0.21.5

- email-validator==2.0.0.post2

- exceptiongroup==1.1.2

- executing==1.2.0

- fastapi==0.100.0

- filelock==3.12.2

- flatbuffers==23.5.26

- flower==2.0.0

- gast==0.4.0

- google-auth==2.22.0

- google-auth-oauthlib==1.0.0

- google-pasta==0.2.0

- grpcio==1.56.2

- h11==0.14.0

- h5py==3.9.0

- html5lib==1.1

- httpcore==0.17.3

- httptools==0.6.0

- humanize==4.7.0

- iniconfig==2.0.0

- installer==0.7.0

- ipython==8.14.0

- itsdangerous==2.1.2

- jaraco-classes==3.3.0

- jax==0.4.13

- jedi==0.18.2

- jeepney==0.8.0

- jinja2==3.1.2

- jsonschema==4.18.4

- jsonschema-specifications==2023.7.1

- keras==2.12.0

- keyring==23.13.1

- libclang==16.0.6

- lockfile==0.12.2

- markdown==3.4.4

- markupsafe==2.1.3

- matplotlib-inline==0.1.6

- ml-dtypes==0.2.0

- more-itertools==9.1.0

- numpy==1.24.3

- oauthlib==3.2.2

- opencv-python==4.8.0.74

- opencv-python-headless==4.8.0.74

- opt-einsum==3.3.0

- orjson==3.9.2

- parso==0.8.3

- pexpect==4.8.0

- pickleshare==0.7.5

- pkginfo==1.9.6

- pluggy==1.2.0

- poetry==1.5.1

- poetry-core==1.6.1

- poetry-plugin-export==1.4.0

- prometheus-client==0.17.1

- protobuf==4.23.4

- ptyprocess==0.7.0

- pure-eval==0.2.2

- py==1.11.0

- pyasn1==0.5.0

- pyasn1-modules==0.3.0

- pycparser==2.21

- pygments==2.15.1

- pyparsing==3.1.0

- pyproject-hooks==1.0.0

- pytest==7.4.0

- python-dotenv==1.0.0

- pyyaml==6.0.1

- rapidfuzz==2.15.1

- redis==4.6.0

- referencing==0.30.0

- requests-oauthlib==1.3.1

- requests-toolbelt==1.0.0

- rpds-py==0.9.2

- rsa==4.9

- secretstorage==3.3.3

- shellingham==1.5.0.post1

- sniffio==1.3.0

- stack-data==0.6.2

- starlette==0.27.0

- tensorboard==2.12.3

- tensorboard-data-server==0.7.1

- tensorflow==2.12.1

- tensorflow-estimator==2.12.0

- tensorflow-io-gcs-filesystem==0.32.0

- termcolor==2.3.0

- tomli==2.0.1

- tomlkit==0.11.8

- tornado==6.3.2

- traitlets==5.9.0

- trove-classifiers==2023.7.6

- ujson==5.8.0

- urllib3==1.26.16

- uvicorn==0.23.1

- uvloop==0.17.0

- virtualenv==20.24.1

- watchgod==0.8.2

- webencodings==0.5.1

- websockets==11.0.3

- werkzeug==2.3.6

- wrapt==1.14.1

prefix: /home/kurekj/anaconda3/envs/environment\_input

W folderze AI4PhenoEOSC\DeploymentFastAPICeleryRabbitMQ\fast-api-celery

Jeżeli chcemy zarządzać wirtualnym środowiskiem conda:

conda env create -f environment\_input.yml

conda activate environment\_input

conda env export --name environment\_input > environment\_input.yml

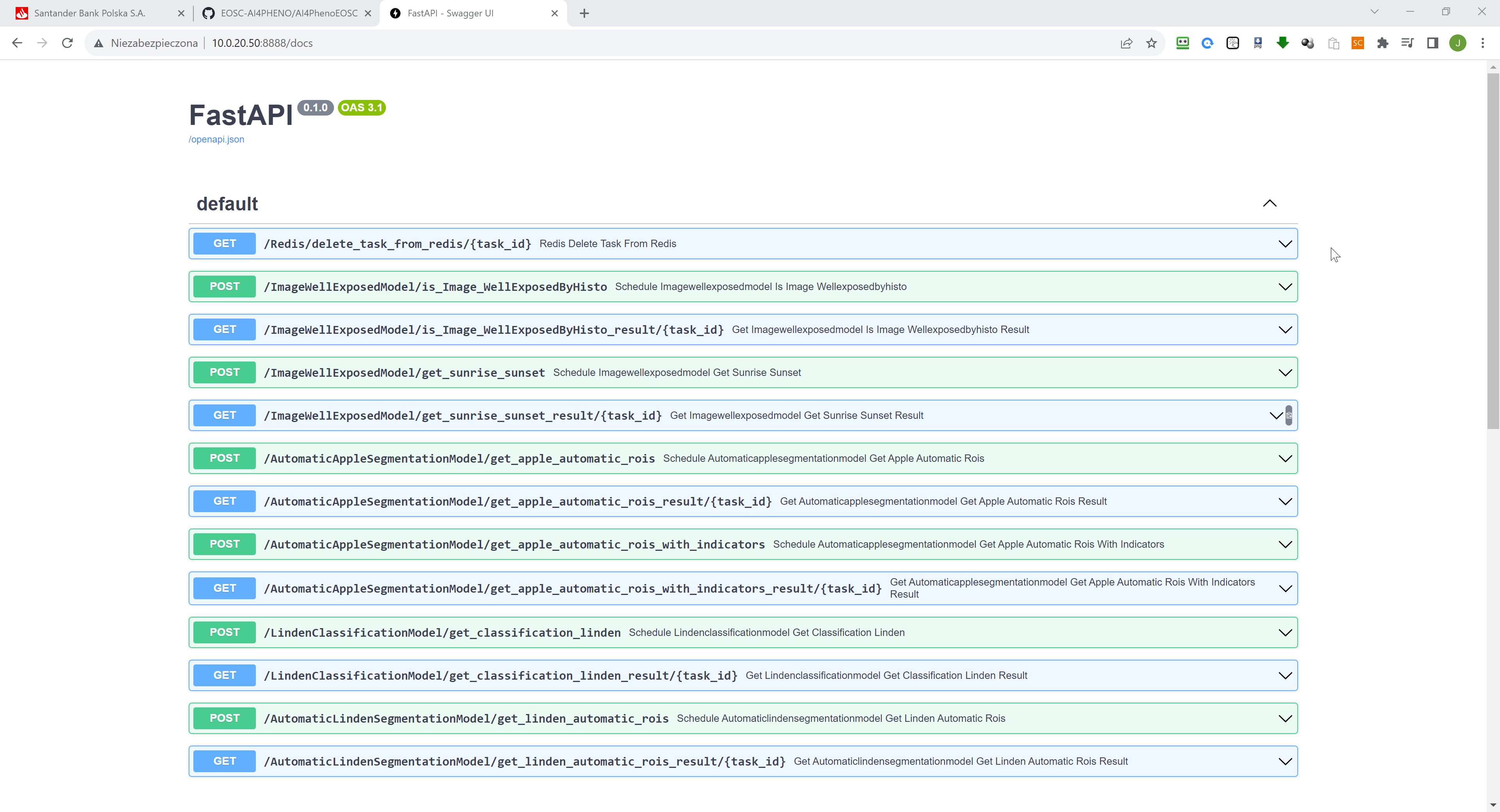
conda env remove --name environment\_input

#####################################

Usługi RESTAPI wystawione są na porcie:8888.

Dostęp do Swaggera:

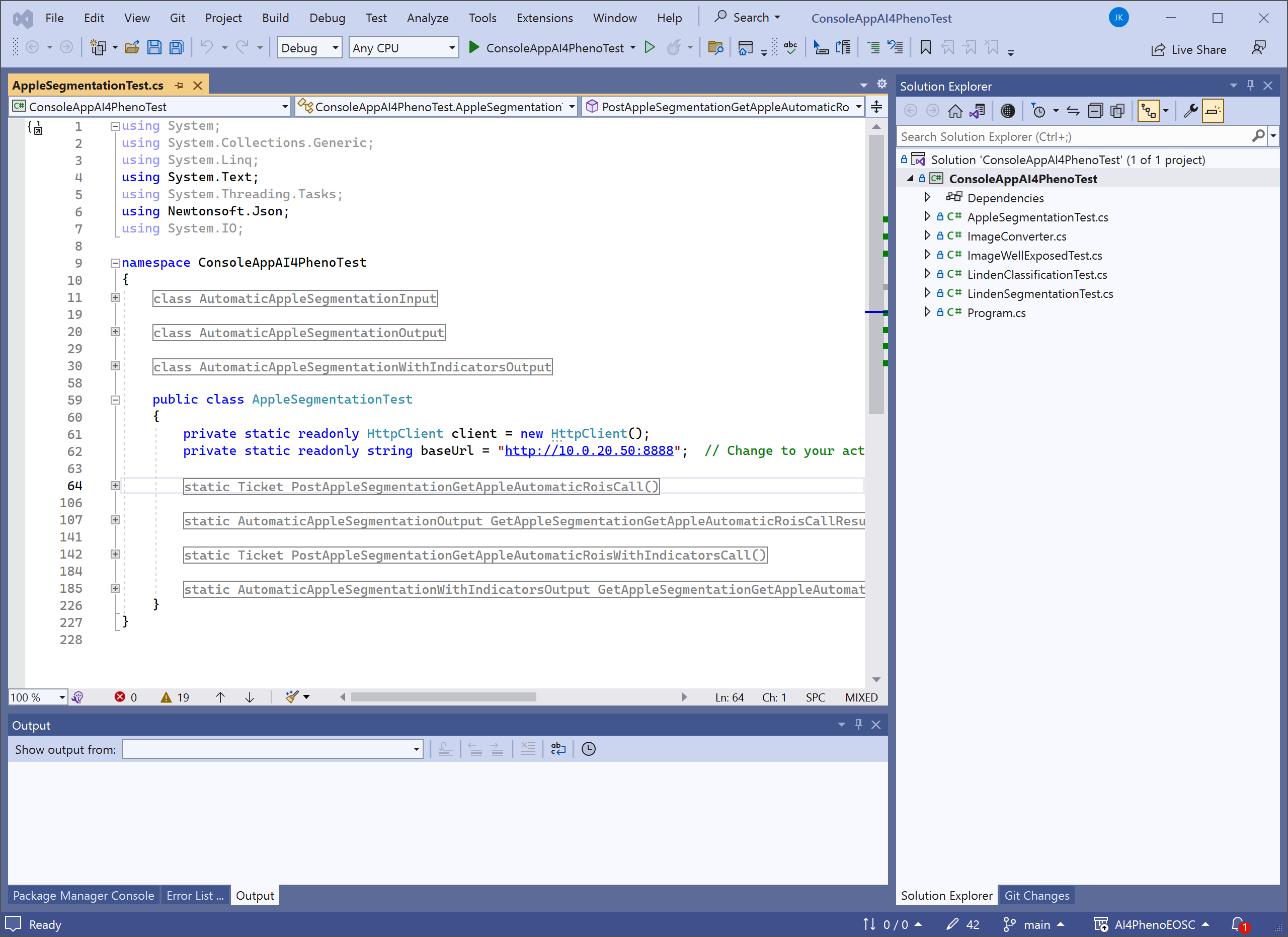
<http://10.0.20.50:8888/docs>



Wszystkie testy i przykłady użycia w C#.NET są w projekcie VisualStudio 2022 umieszczonym w

folderze: DeploymentFastAPICeleryRabbitMQTestC#\ConsoleAppAI4PhenoTest\ConsoleAppAI4PhenoTest.sln

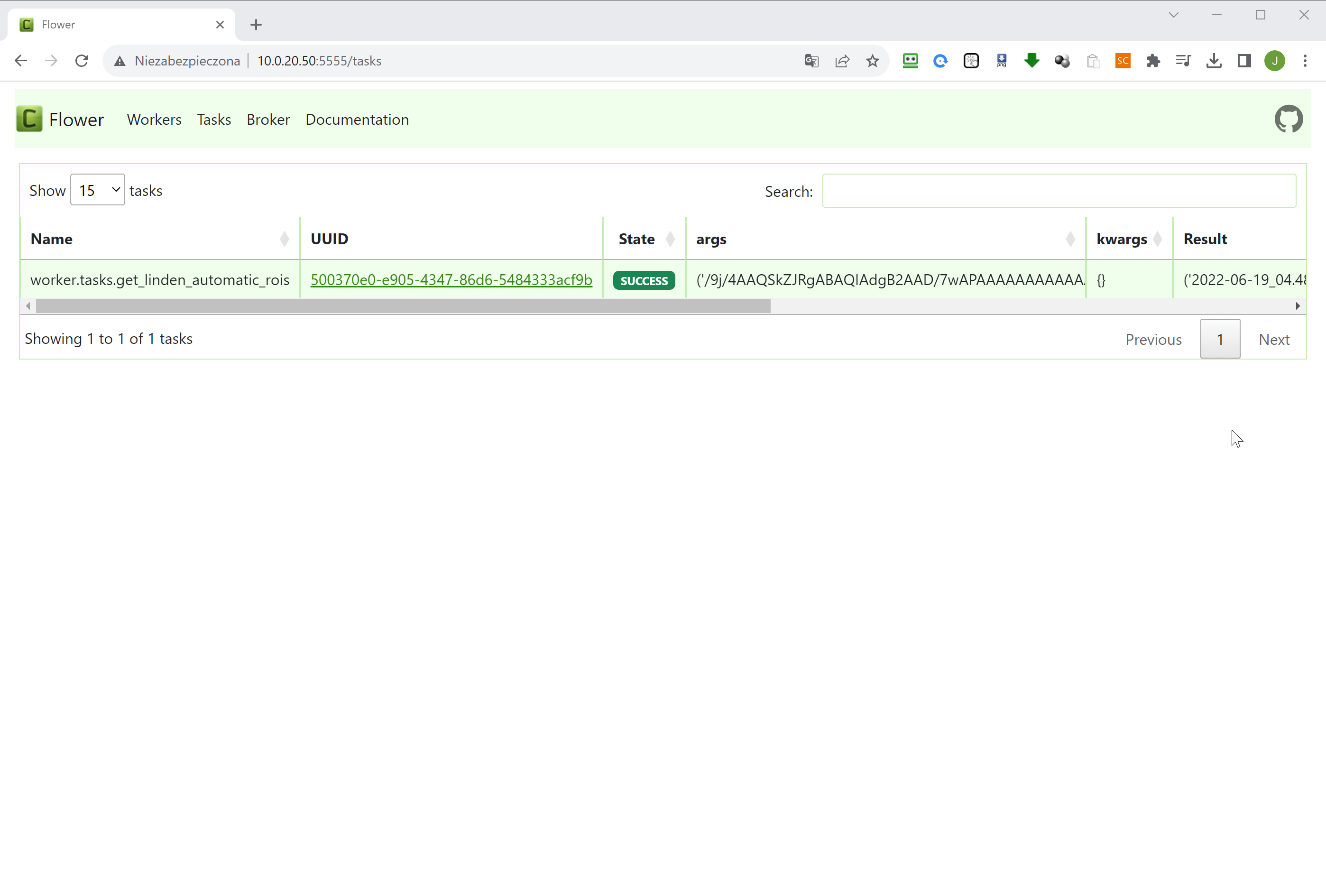
W Visual Studio odpalamy powyższy plik solucji.



FastAPI przekazuje request to RabbitMQ a następnie do Celery i jeżeli Model wymaga użycia karty GPU to wywołuje model z Nvidia Triton. Wynik jest zwracany do REDIS i utrzymywany przez 24h, chyba że się samemu wyrzuci.

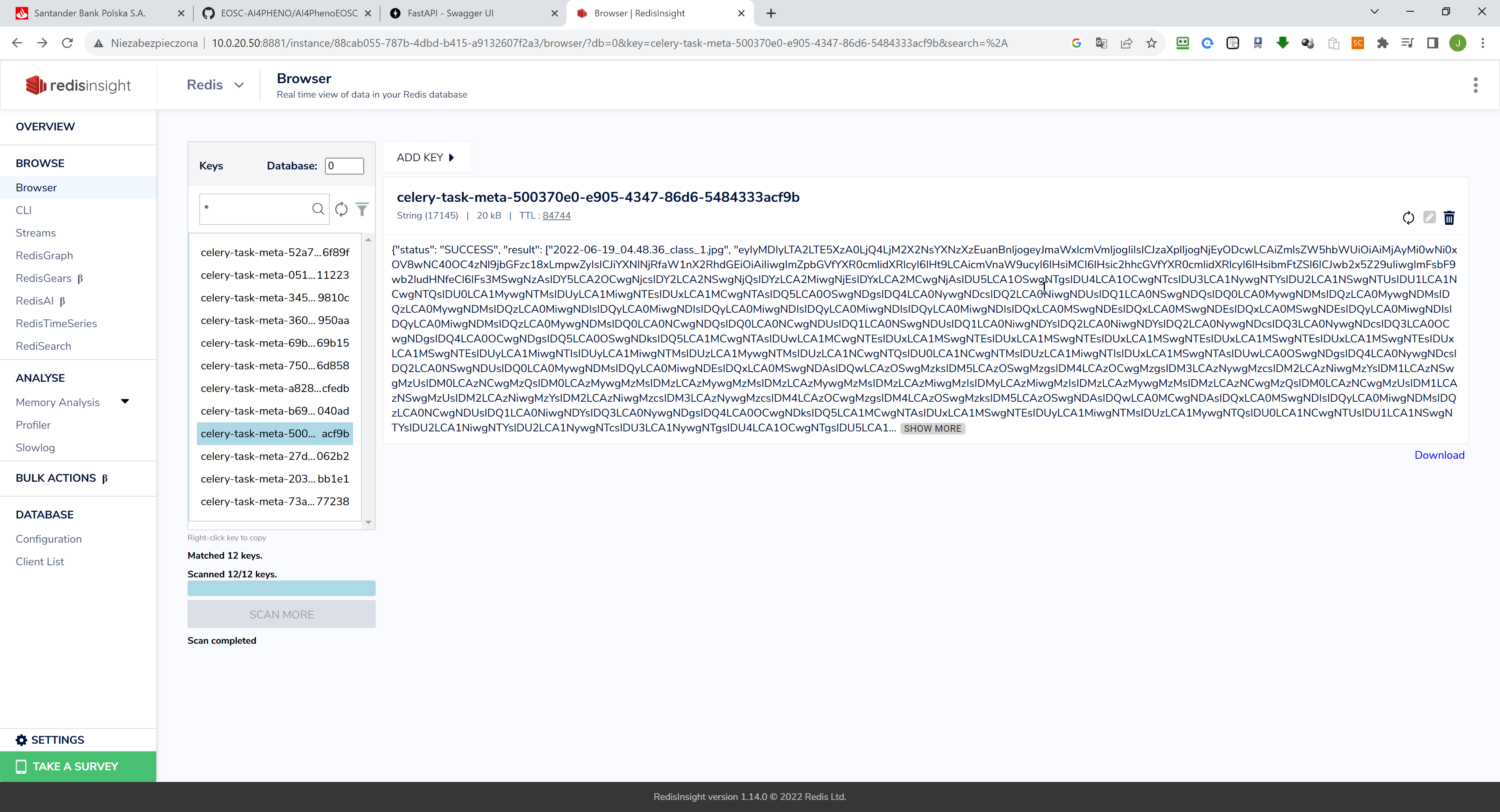
Nakładka graficzna Flower pozwala nadzorować zadania w Celery:

<http://10.0.20.50:5555/tasks>



Możemy przegląda wyniki, które są przechowywane w REDI za pomocą RedisInsight:

<http://10.0.20.50:8881/instance/88cab055-787b-4dbd-b415-a9132607f2a3/browser/?db=0&key=celery-task-meta-d8c786f6-6d98-4f60-b09c-6a77f8983e1a&search=%2A>

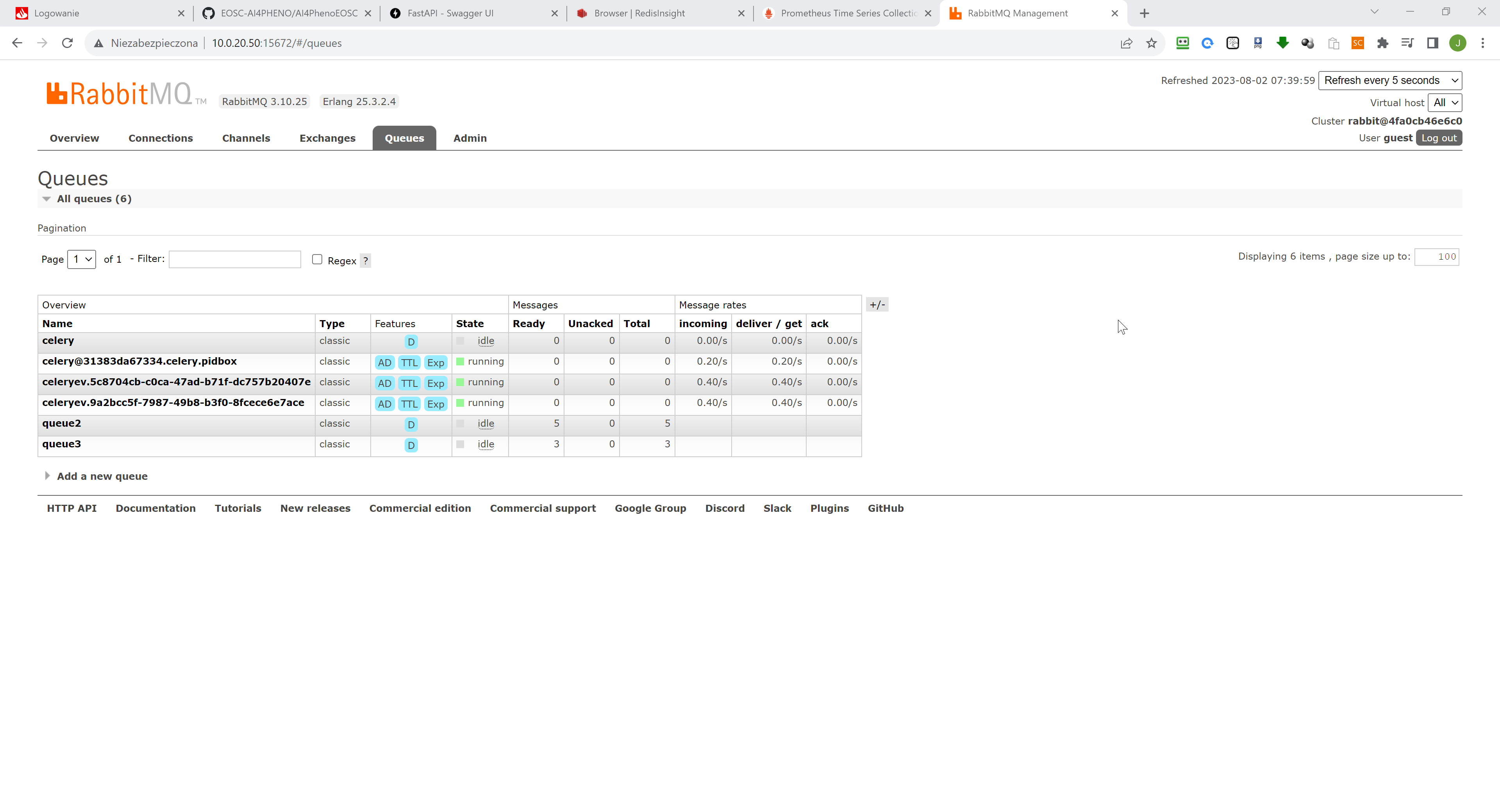


Dodatkowo można przeglądać RABBITMQ za pomocą:

<http://10.0.20.50:15672/#/nodes/rabbit%404fa0cb46e6c0>

login:guest

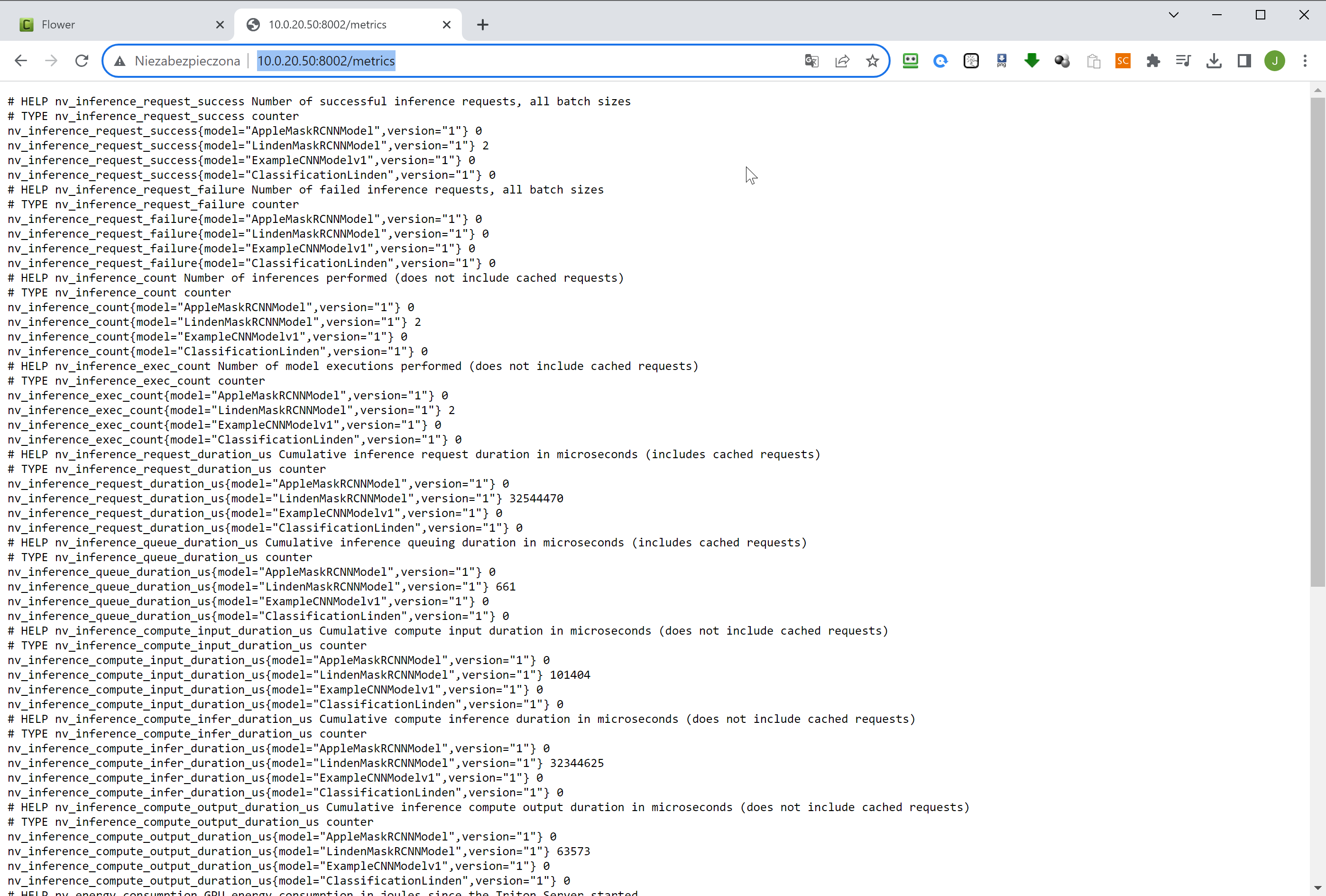
password:guest



Nvidia Triton jest odpalony na porcie:

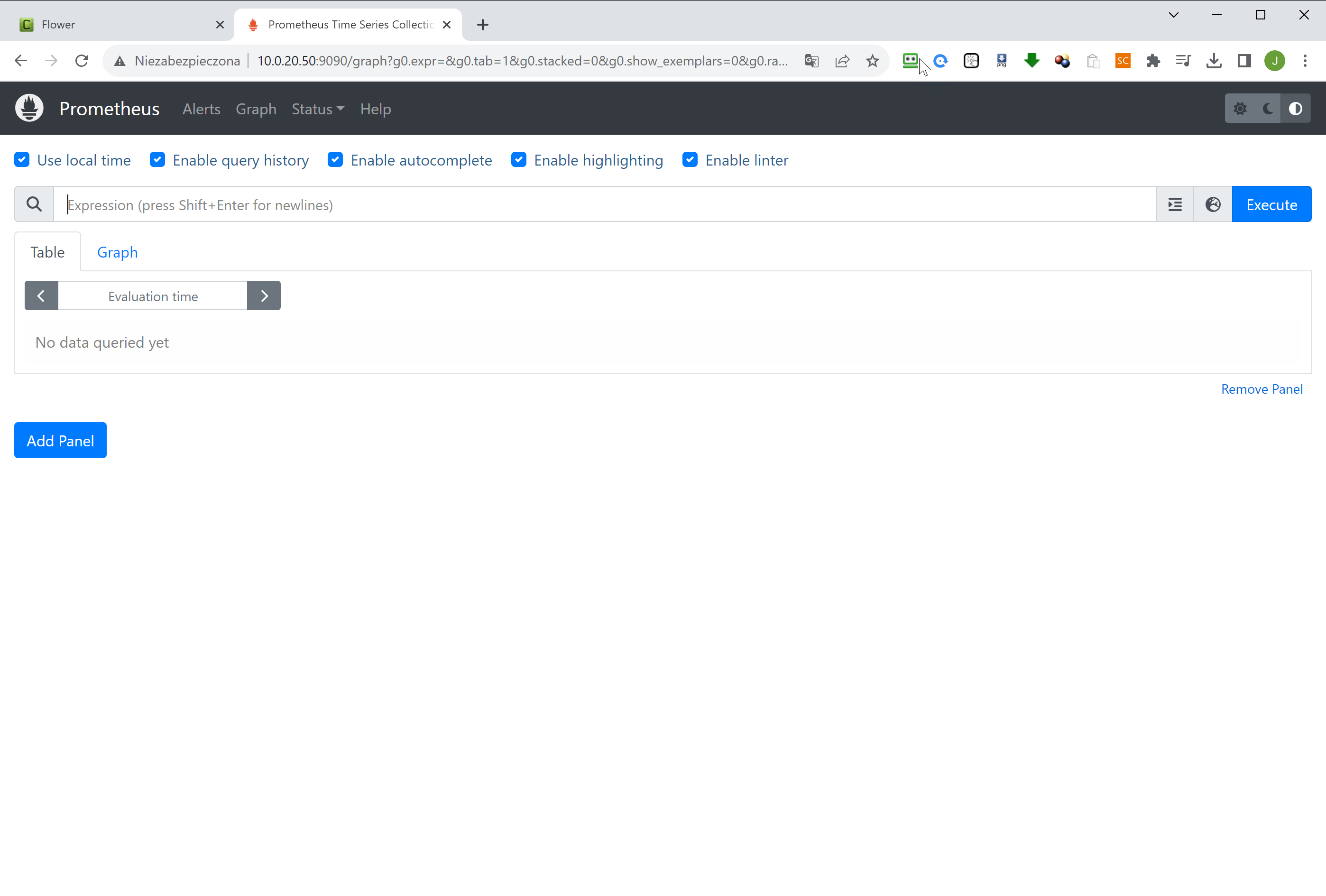
1. 8000 dla RESTAPI - nie używamy
2. 8001 dla GRPC - używamy
3. 8002 dla metryk – używamy dla Prometeusza

<http://10.0.20.50:8002/metrics>



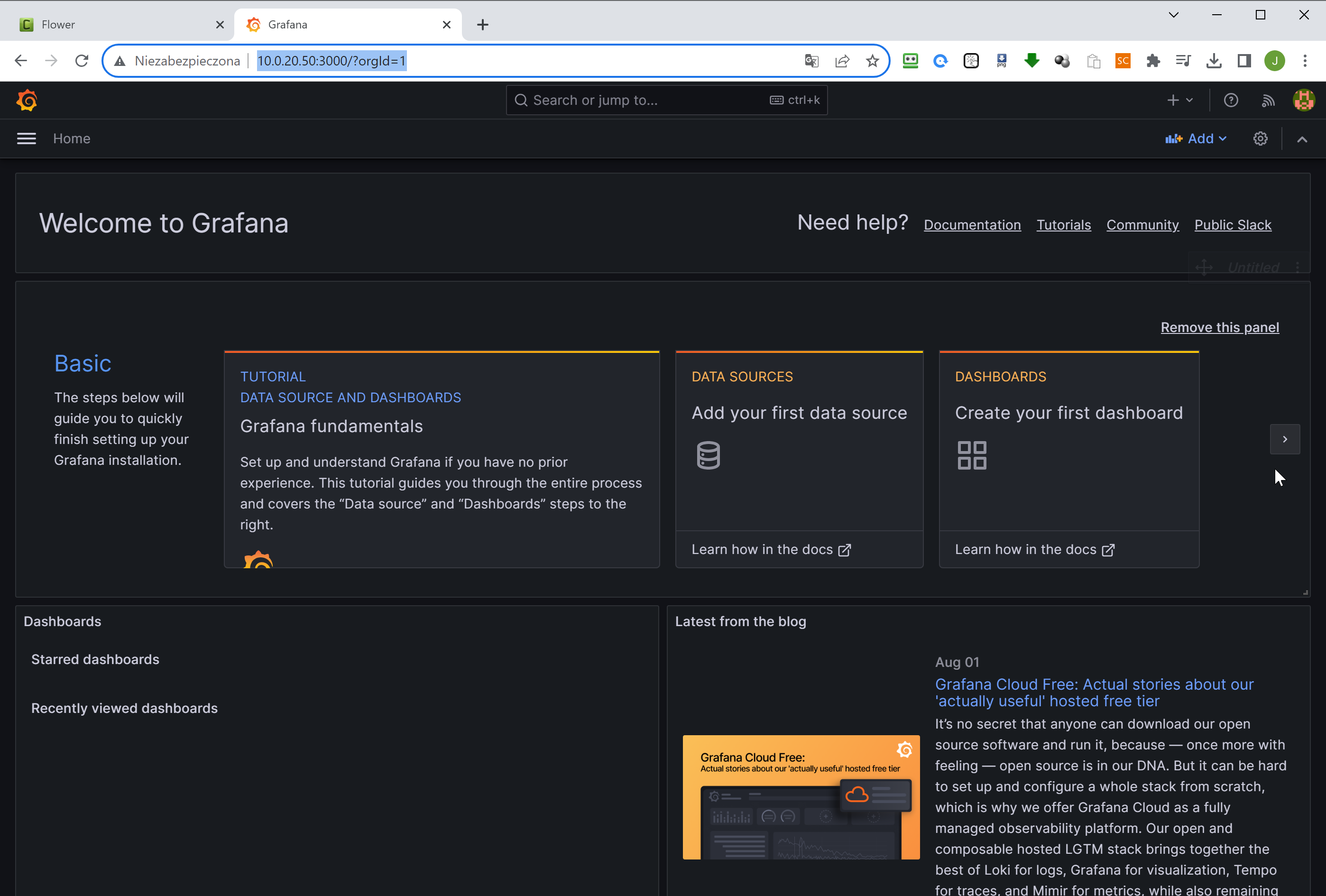
Te metryki są zapisywane w Prometeuszu, czyli Prometeusz pobiera jest co 15s

<http://10.0.20.50:9090/config>



Nakładką na Prometeusza jest Grafana:

<http://10.0.20.50:3000/?orgId=1>



Przygotowanie modeli do developmentu: AI4PhenoEOSC\DeploymentTriton