인공지능 기술의 대중화(AI Democratization)를 위한

TANGO 커뮤니티 제1회 컨퍼런스

TANGO

TANGO 개발자 가이드

성 명 김홍숙 **소 속** ETRI

주관 ETRI 한국전자통신연구원 (TANGO) 주최 과학기술정보통신부 IITP 정보통신기획평가원 후원 lablup weda tesla System 한국인공지능협회 SNUH 서울대학교병원 고려대학교 KOREA UNIVERSITY 홍익대학교 HONGIK UNIVERSITY CAU 중앙대학교 Georgia Tech

TANGO

목차

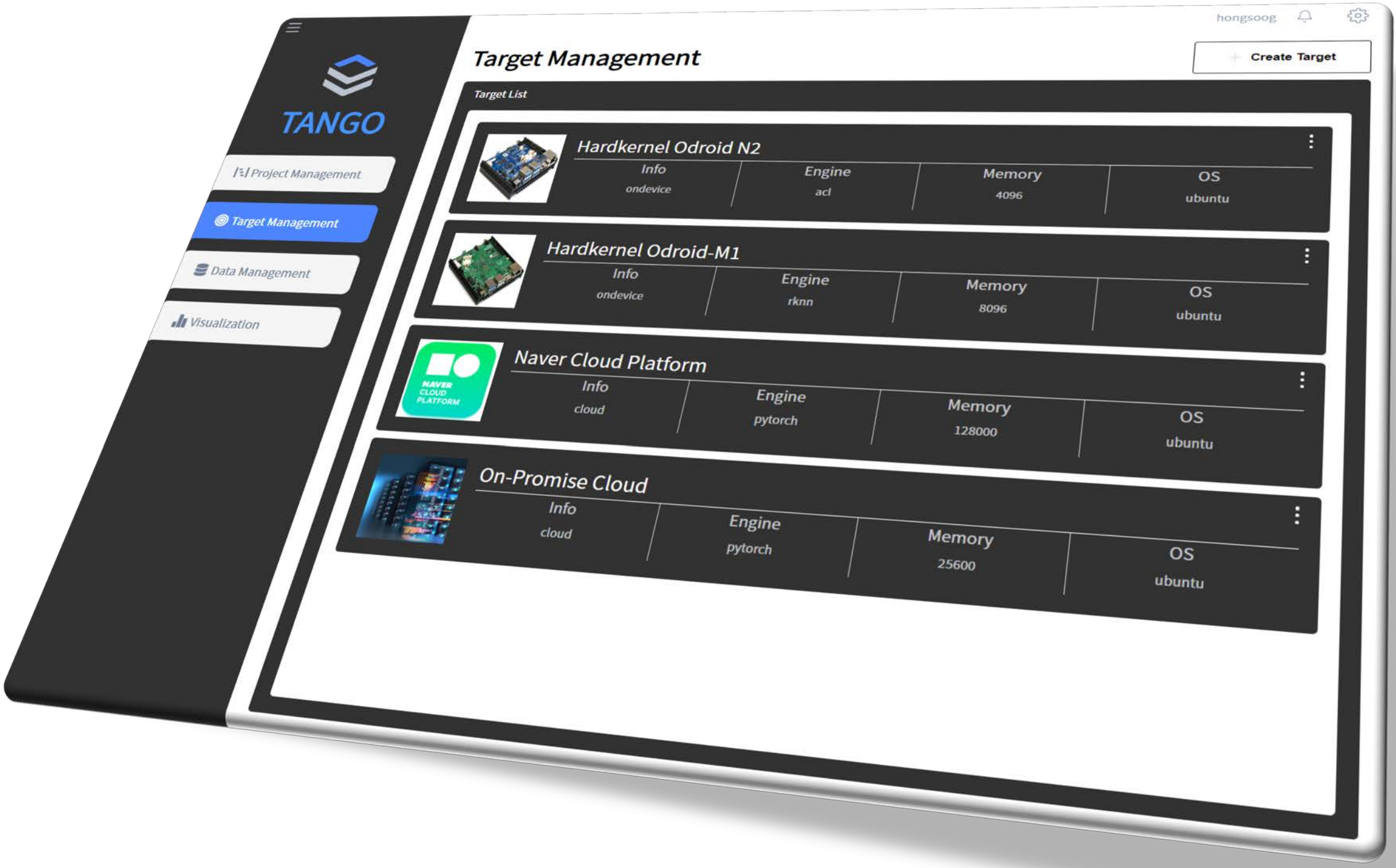
TANGO

TANGO

hongsoog

Log in

Don't have account? Register now


hongsoog
TANGO
Project Management
Target Management
Data Management
Visualization
Target Management
Create Target
Target List
Hardkernel Odroid N2
Info
ondevice
Engine
acl
Memory
4096
OS
ubuntu
Hardkernel Odroid-M1
Info
ondevice
Engine
rknn
Memory
8096
OS
ubuntu
Naver Cloud Platform
NAVER CLOUD PLATFORM
Info
cloud
Engine
pytorch
Memory
128000
OS
ubuntu
On-Promise Cloud
Info
cloud
Engine
pytorch
Memory
25600
OS
ubuntu


hongsoog
Data Management
TANGO
Project Management
Target Management
Data Management
Visualization
BluAI
ML tool by weda
DataSet
Add New Dataset
COUNT: 1
test
IMAGE
CLASSIFICATION
x-ray
created by
at 2022-10-30 20:13
TRAINING
DataSet
1 Datasets are ready
Edit Dataset
Title
test
Data Type
Image
Video
Purpose
Classfication
Detection
Segmentation
Details
x-ray
Normal
Virus
Bacteria
Total : 21 / Success : 21 / Fail : 0
Label File Name Size Type Status
1 Nor··· NORMAL-1128157-··· 244.5··· jpeg
2 Nor··· NORMAL-471048-0··· 218.6··· jpeg
3 Nor··· NORMAL-4853105-··· 242.3··· jpeg
4 Nor··· NORMAL-5398062-··· 215.6··· jpeg
Apply


hongsoog
TANGO
Project Management
Target Management
Data Management
Visualization
Diagonosys of tuberculosis
Description
Diagonosys of tuberculosis based on X-ray images
설명 수정
Information
Dataset
Target
New Target
NAVER CLOUD PLATFORM
Configuration
Task Type
Classification
Detection
AutoNN Config
Dataset file :
dataset.yaml
Base Model :
basemodel.yaml
Nas Type
Backbone Nas
Neck Nas
신경망 자동 생성
Current Work - [ ]
BMS
Visualization
AutoNN
Image Gen
Image Deploy
Run Image

1. TANGO as MSA

TANGO based on MSA

- MSA (Micro Service Architecture) using Docker Container
  - structures an application as a collection of services

- A microservice [1]
  - is responsible for a single capability.
  - is individually deployable.
  - consists of one or more processes.
  - owns its own data store.
  - replaceable.

- A small team can maintain a few handfuls of microservices.

[1] Manning Microservice in .Net, 2021

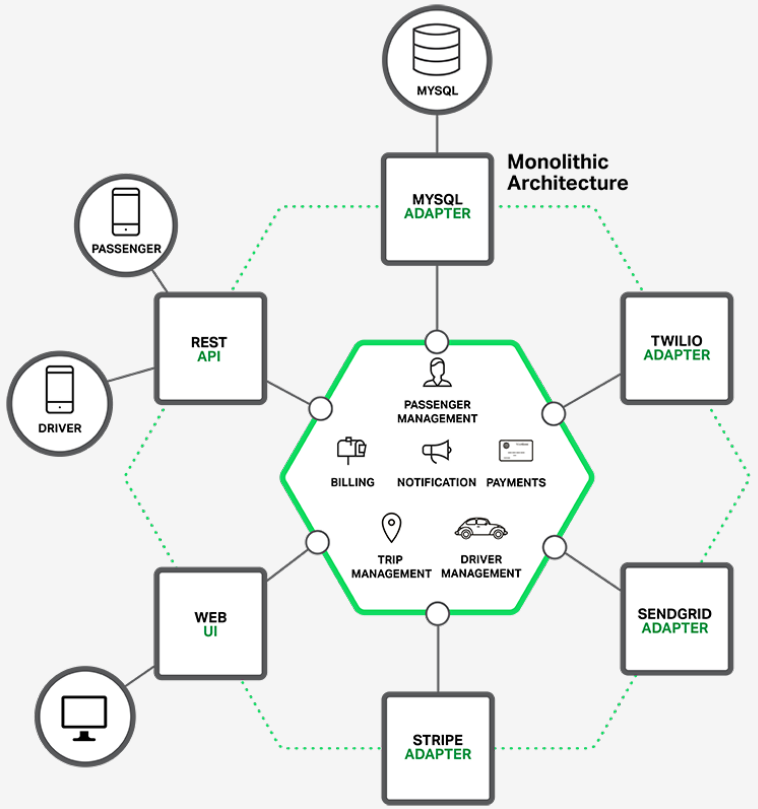
TANGO based on MSA

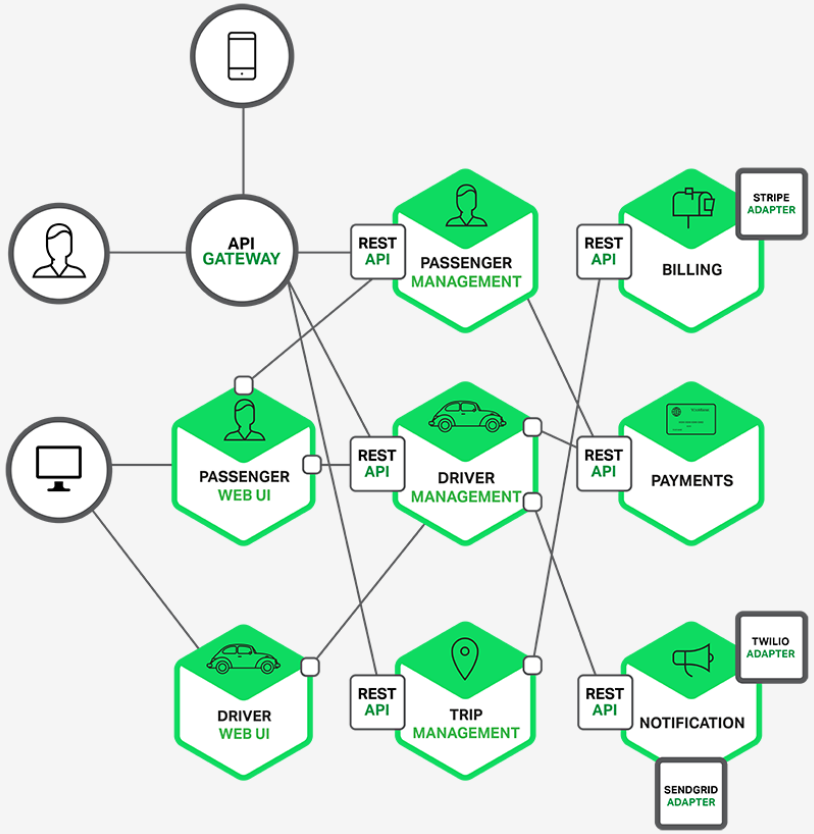
- Microservices are independently deployable modules[2].

- MSA Benefits[3]:
  - Highly maintainable and testable
  - Loosely coupled
  - Independently deployable
  - Organized around business capabilities
  - Owned by a small team

[2] Manning - Microservices - A Practical Guide on Principles, Concepts, and Recipes 2019
[3] https://microservices.io/

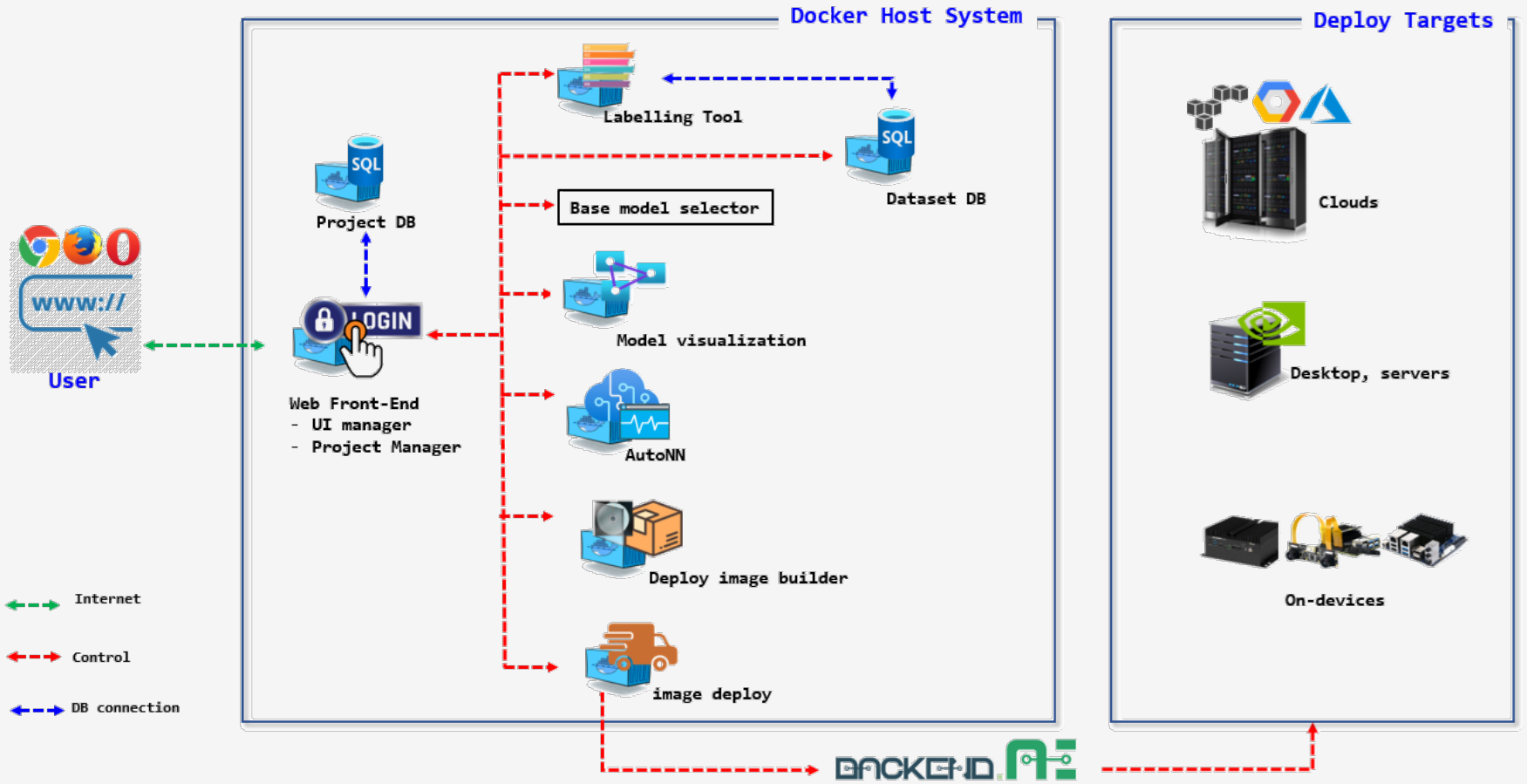
2. TANGO as MSA





source: https://www.nginx.com/blog/introduction-to-microservices/

MSA in TANGO



source: https://github.com/ML-TANGO/TANGO

2. Developer Guides

TANGO Wiki

- https://github.com/ML-TANGO/TANGO/wiki
- All Documents maintained as Wiki
    - Guides
        - TANGO Architecture
        - Exchange Data among Containers
        - Rest API
        - Container Port Map
    - HowTo
        - Common HowTos
    - References
        - Git, GitHub, Docker, Docker-compose

2. Developer Guides

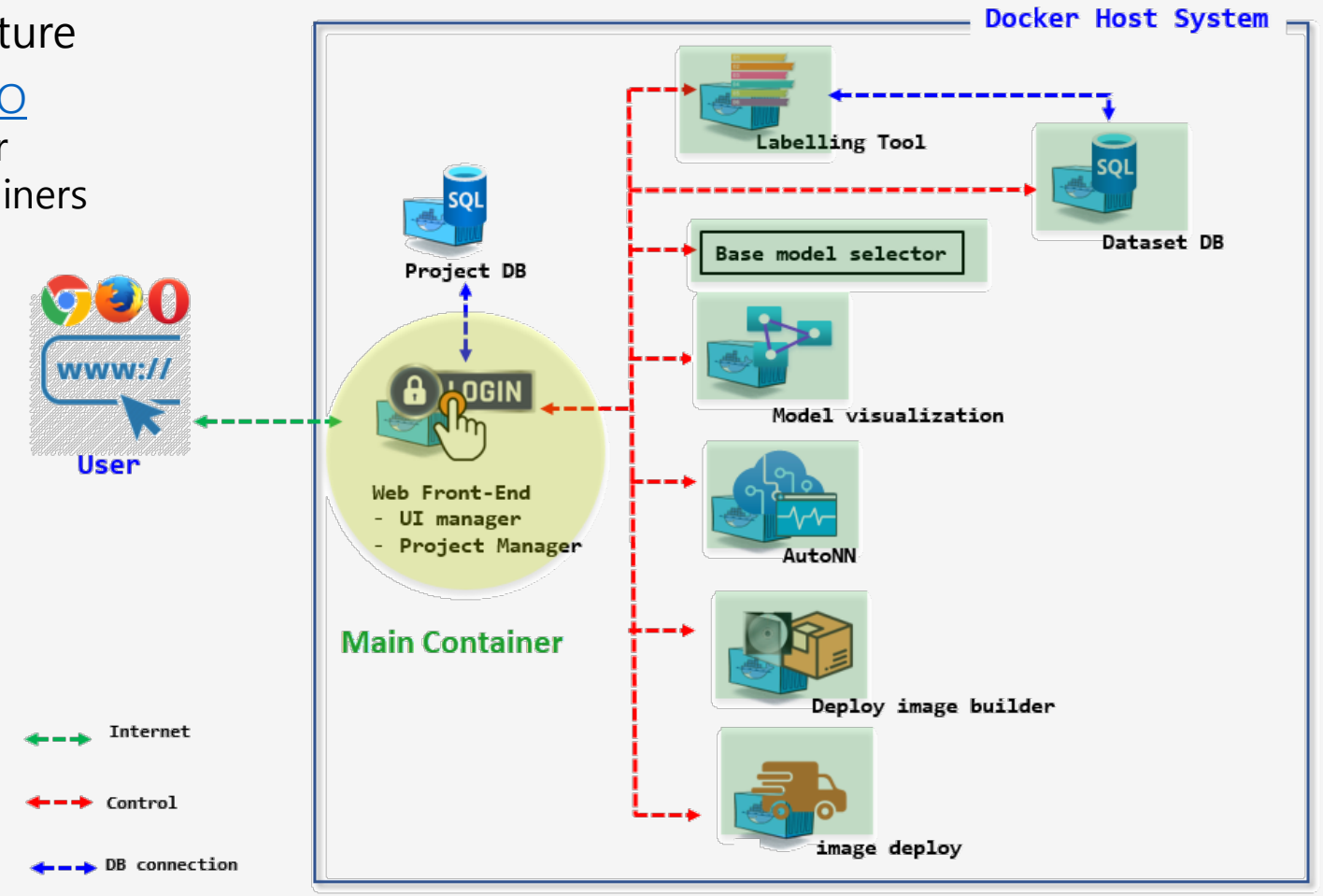
1/4 TANGO Architecture

- Containers in TANGO
- From TANGO Project Creation To Deployment onto Target Devices
- Overall Control Flow in TANGO Project

2. Developer Guides

1/4 TANGO Architecture

- Containers in TANGO
  - Main Container
  - Member Containers

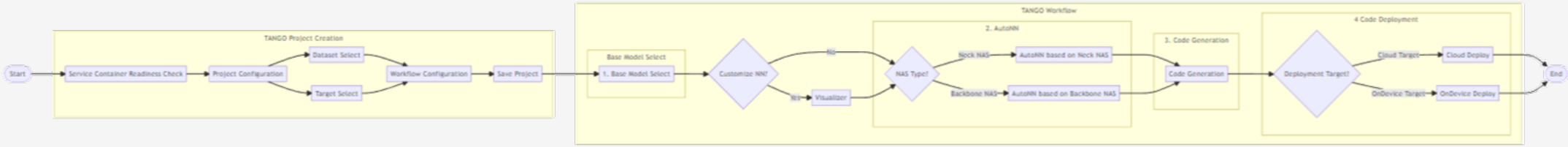


2. Developer Guides

1/4 TANGO Architecture

- From TANGO Project Creation To Deployment onto Target Devices
- Overall Control Flow in TANGO Project
  - Project Configuration
    - Member Container Readiness Check
    - Dataset, Target Selection
    - Workflow Definition

  - Project Workflow
    - using REST API: `start()`, `stop()`, `status_request()`, `status_report()`



2. Developer Guides

2/4 Exchange Data among Containers

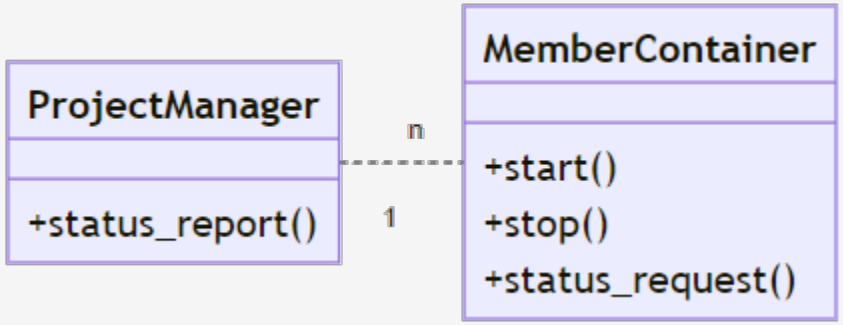
- using Docker host volume
- via volumes: in docker-compose.yaml

```
$ tree ./TANGO/
./TANGO/
└── shared
    ├── common
    │   └── user_id
    │       └── project_id
    │           ├── project_info.yaml     // generated by project manager
    │           ├── data_set.yaml         // generated by labelling
    │           ├── basemodel.yaml        // generated by Base Model Select
    │           ├── model_x.json          // generated by Visualizer
    │           ├── neural_net_info.yaml  // generated by AutoNN
    │           ├── best.onnx             // generated by AutoNN
    │           ├── best.pt               // generated by AutoNN
    │           ├── model.py              // generated by AutoNN
    │           ├── deployment.yaml       // generated by Code_Gen
    │           ├── nn_model              // folder genearted by Code_gen
    │           └── nn_model.zip          // zip of nn_model foler for OnDevice
    │
developers
    │
    └── datasets
        └── coco
            ├── train        // folder for train images
            ├── train.txt
            ├── test         // folder for test images
            ├── test.txt
            ├── val          // foler for val images
            └── val.txt
```

2. Developer Guides

3/4 REST API

- Exposed API from Main Container
  - `status_report()`
- Exposed API form Member Containers
  - `start()`
  - `stop()`
  - `status_request()`



Note on API provider and consumer

- Don't confuse between API provider and API consumer.

2. Developer Guides

3/4 REST API

Core APIs in TANGO

- In HTTP GET Message format

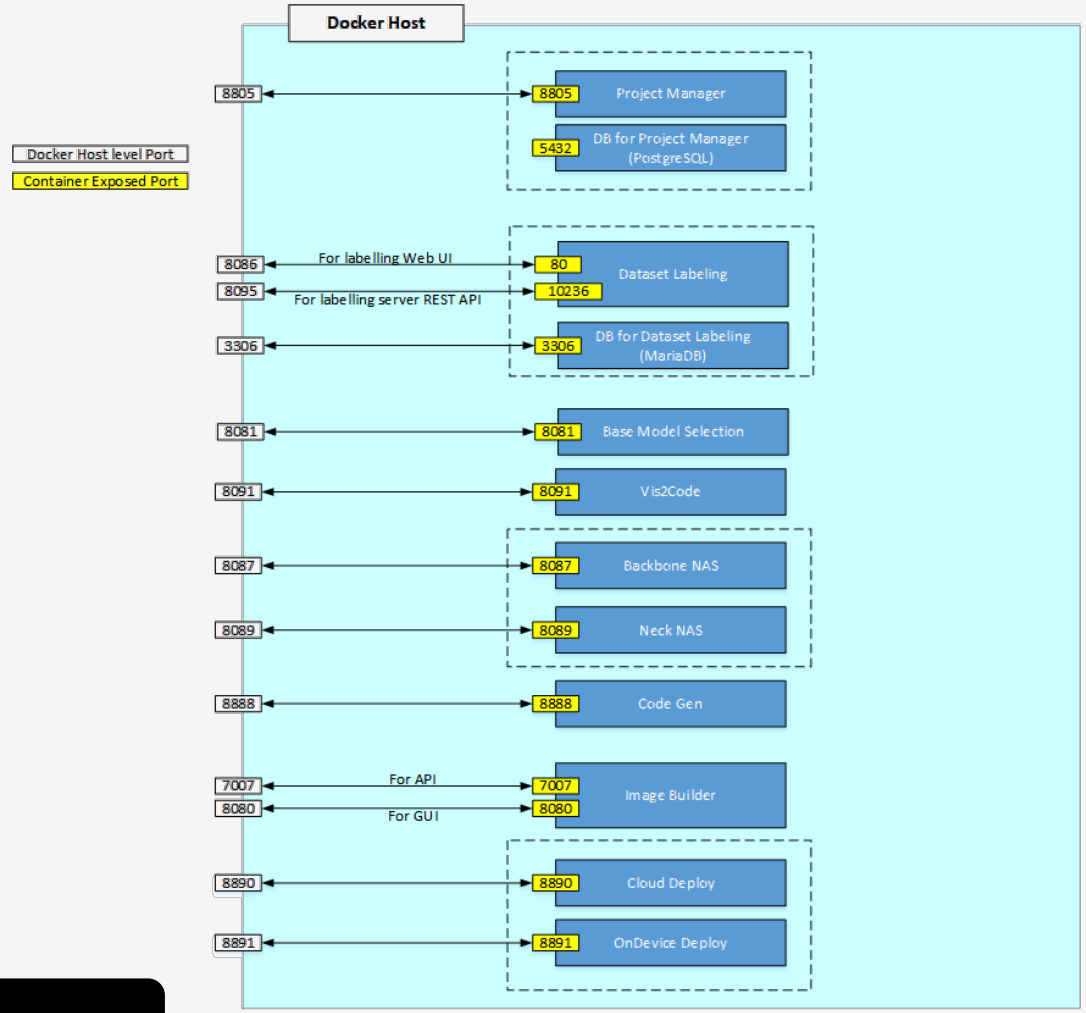
Type	Expose REST APIs
Main	http://<DOCKER_HOST_IP>:<PROJECT_MANAGER_PORT>/**status_report**? container_id=<container_id>&user_id=<user_id>&project_id=<project_id>&status=<status>
Member	http://<DOCKER_HOST_IP>:<MEMBER_CONTAINER_PORT>/**start**? user_id=<user_id>&project_id=<project_id>
	http://<DOCKER_HOST_IP>:<MEMBER_CONTAINER_PORT>/**stop**? user_id=<user_id>&project_id=<project_id>
	http://<DOCKER_HOST_IP>:<MEMBER_CONTAINER_PORT>/**status_request**? user_id=<user_id>&project_id=<project_id>

- All TANGO member container should implement following APIs at least;
  - `start()`,
  - `stop()`, and
  - `status_request()`

2. Developer Guides

4/4 Container Port Map

- for identifying the containers
- all containers publish its port
  - via `ports:` in `docker-compose.yaml`



3. How service defined and TANGO app runs?

Dockerfle, Docker Compose, and docker-compose.yaml

- A **`Dockerfile`** is a simple text file that contains the commands a user could call to assemble an image
- **Docker Compose** is a tool for defining and running multi-container Docker applications.
- **Docker Compose**
  - define the services that make up your app in **`docker-compose.yml`** so they can be run together in an isolated environment.
  - gets an app running in one command by just running '*`docker-compose up`*'.
- **Docker compose** uses the `Dockerfile`
  - if you add the *`build`* command to your project's **`docker-compose.yml`** .

4. How to build TANGO

```
# Change working directory to TANGO top directory
$ cd ~/work/TANGO

# Build TANGO Docker images and run containers
$ docker-compose up -d --build
or
$ docker-compose up -d

# Cleanup TANGO images, containers and volumes
$ docker-compose down --rmi all --volumes
or
$ docker-compose down --rmi all
```

QnA
Quit and Adios!

주관 ETRI 한국전자통신연구원 (TANGO) 주최 과학기술정보통신부 IITP 정보통신기획평가원 후원 lablup weda tesla System 한국인공지능협회 SNUH 서울대학교병원 고려대학교 KOREA UNIVERSITY 홍익대학교 HONGIK UNIVERSITY CAU 중앙대학교 Georgia Tech

감사합니다.

Supple. - Screen shots

TANGO

TANGO

hongsoog

Log in

Don't have account? Register now

TANGO

- Project Management
- Target Management
- Data Management
- Visualization

hongsoog

Target Management

+ Create Target

Target List

Hardkernel Odroid N2

Info	Engine	Memory	OS
ondevice	acl	4096	ubuntu

Hardkernel Odroid-M1

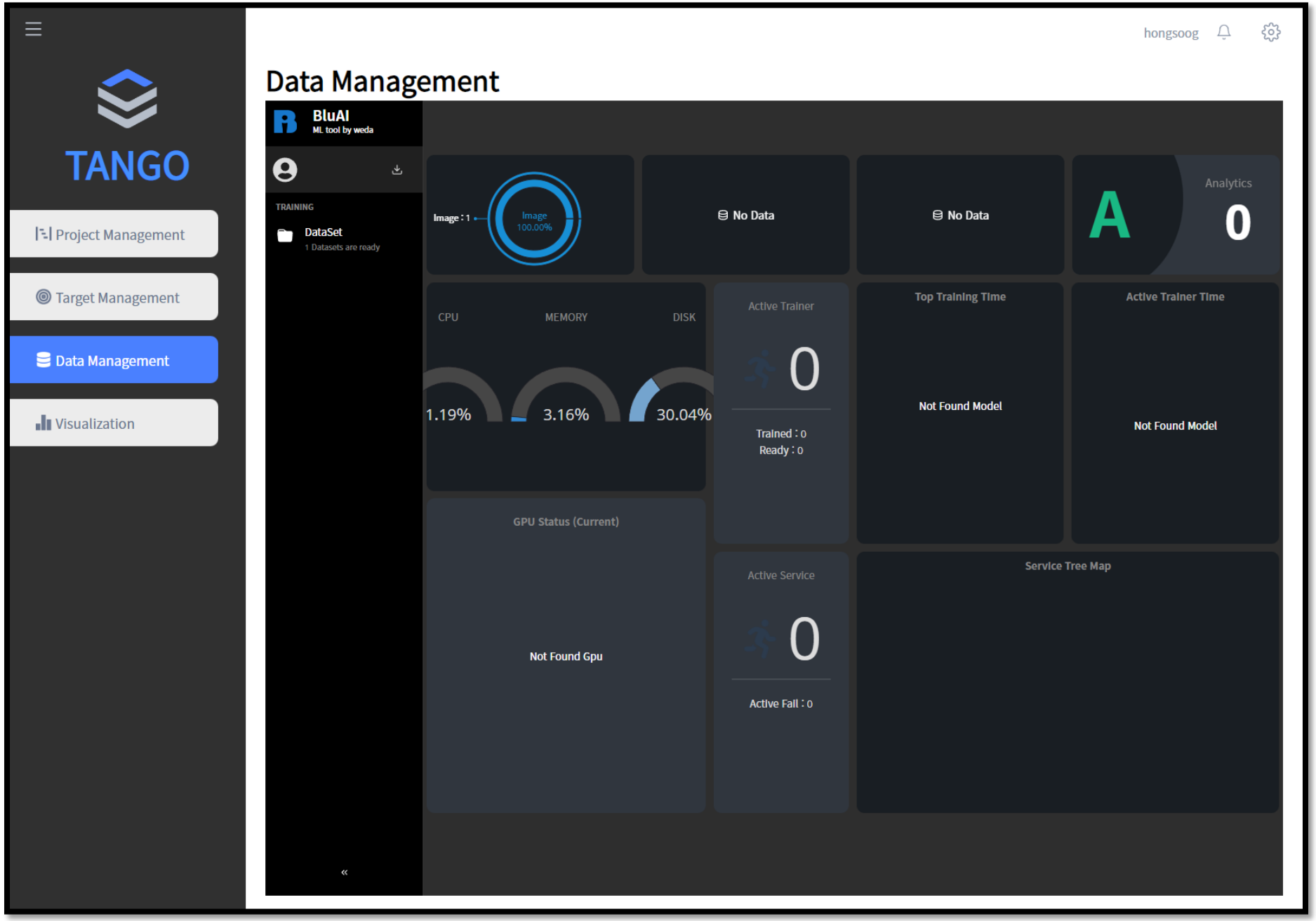
Info	Engine	Memory	OS
ondevice	rknn	8096	ubuntu

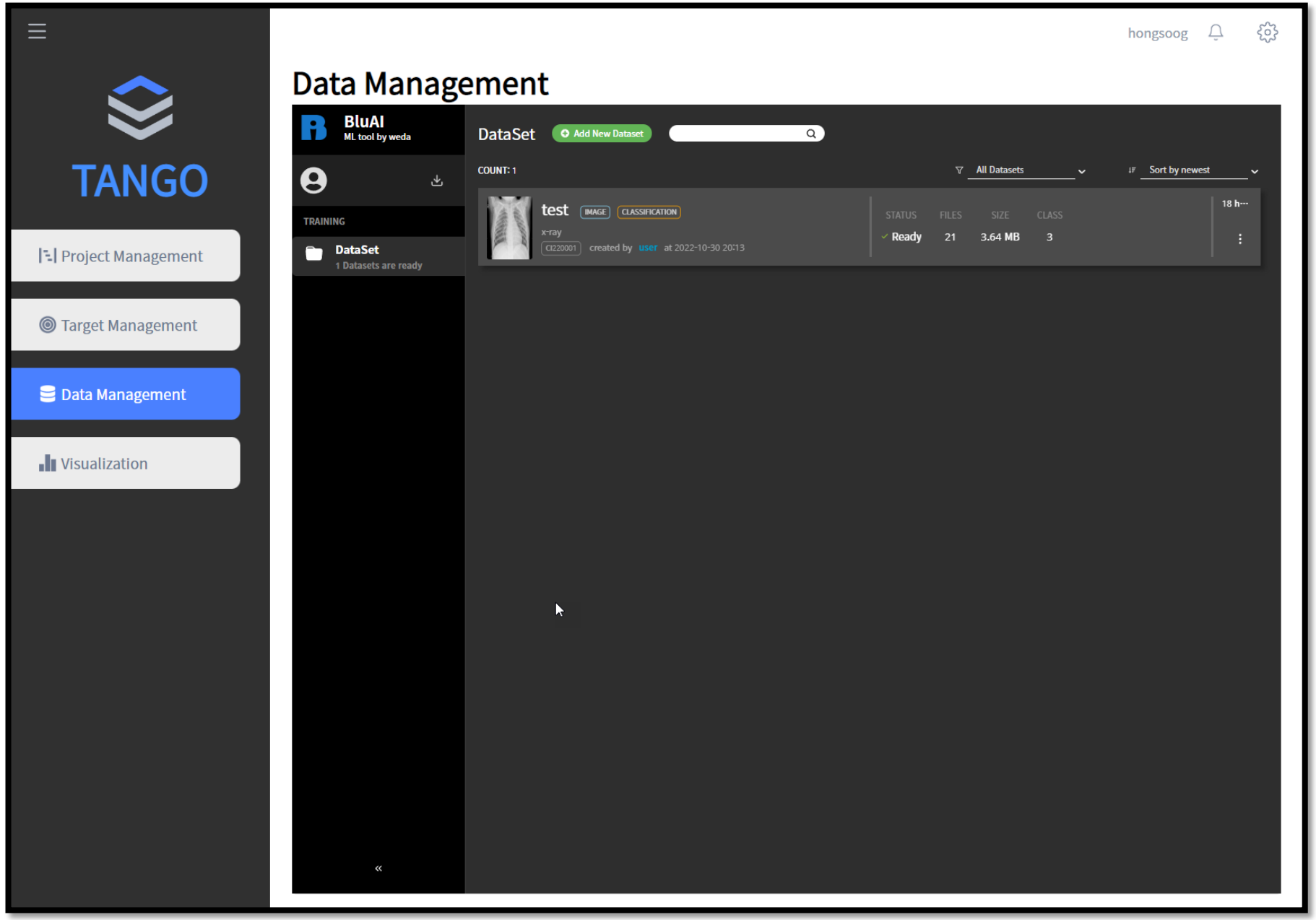
Naver Cloud Platform

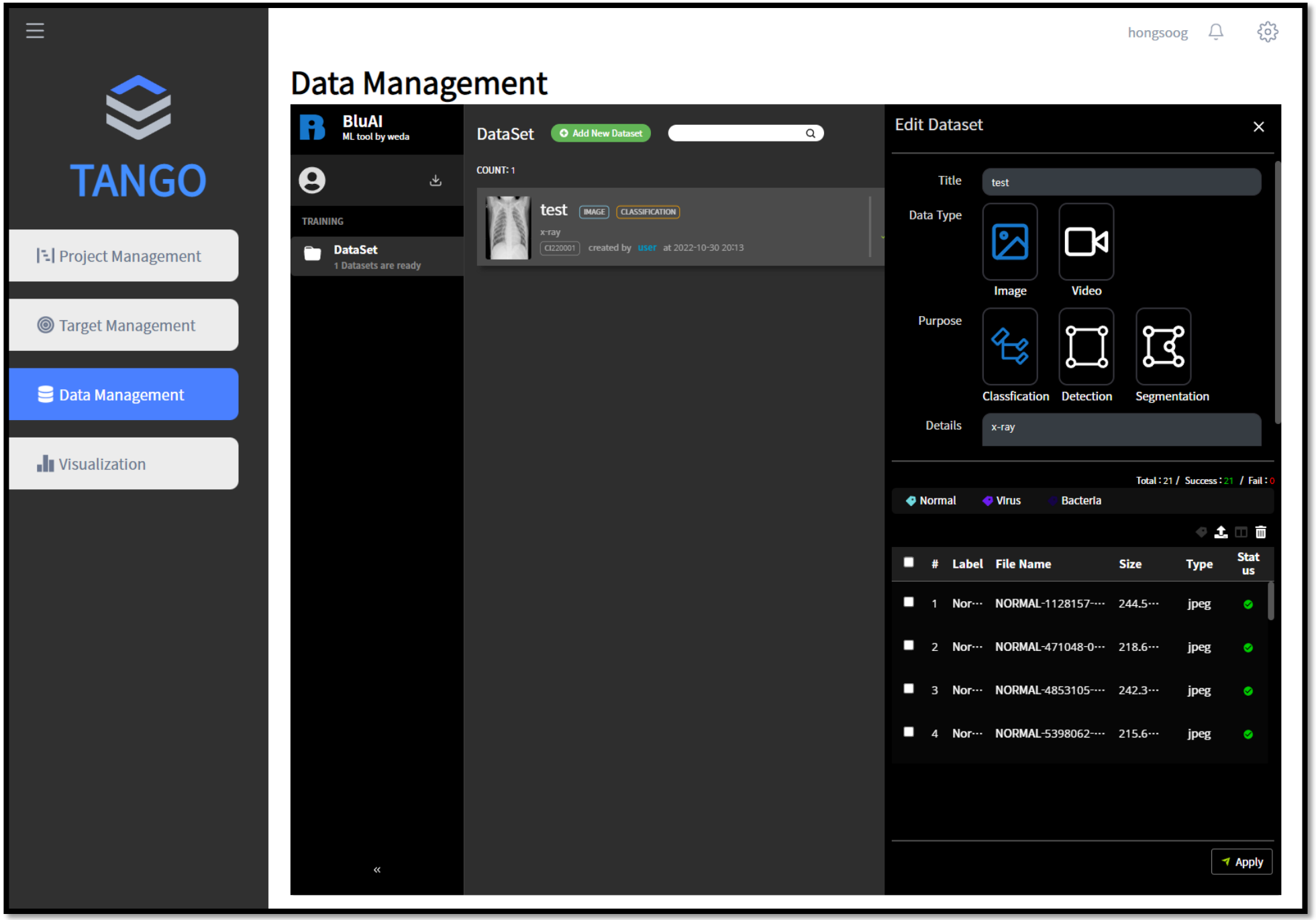
Info	Engine	Memory	OS
cloud	pytorch	128000	ubuntu

On-Promise Cloud

Info	Engine	Memory	OS
cloud	pytorch	25600	ubuntu


TANGO
Project Management
Target Management
Data Management
Visualization
hongsoog
Data Management
BluAI
ML tool by weda
TRAINING
DataSet
1 Datasets are ready
Image : 1
Image
100.00%
No Data
No Data
A
Analytics
0
CPU
MEMORY
DISK
1.19%
3.16%
30.04%
Active Trainer
0
Trained : 0
Ready : 0
Top Training Time
Not Found Model
Active Trainer Time
Not Found Model
GPU Status (Current)
Not Found Gpu
Active Service
0
Active Fail : 0
Service Tree Map


TANGO
Project Management
Target Management
Data Management
Visualization
hongsoog
Data Management
BluAI
ML tool by weda
TRAINING
DataSet
1 Datasets are ready
DataSet
Add New Dataset
COUNT: 1
All Datasets
Sort by newest
test
IMAGE
CLASSIFICATION
x-ray
CI220001
created by user at 2022-10-30 20:13
STATUS
FILES
SIZE
CLASS
Ready
21
3.64 MB
3
18 h


TANGO
Project Management
Target Management
Data Management
Visualization
hongsoog
Data Management
BluAI
ML tool by weda
TRAINING
DataSet
1 Datasets are ready
DataSet
Add New Dataset
COUNT: 1
test
IMAGE
CLASSIFICATION
x-ray
CI220001
created by user at 2022-10-30 20:13
Edit Dataset
Title
test
Data Type
Image
Video
Purpose
Classfication
Detection
Segmentation
Details
x-ray
Total : 21 / Success : 21 / Fail : 0
Normal
Virus
Bacteria
Label File Name Size Type Status
1 Nor··· NORMAL-1128157···· 244.5··· jpeg
2 Nor··· NORMAL-471048-0··· 218.6··· jpeg
3 Nor··· NORMAL-4853105···· 242.3··· jpeg
4 Nor··· NORMAL-5398062···· 215.6··· jpeg
Apply

TANGO

- Project Management
- Target Management
- Data Management
- Visualization

hongsoog

Diagonosys of tuberculosis

Description: Diagonosys of tuberculosis based on X-ray images [설명 수정]

Information

Dataset

Target

New Target

Configuration

Task Type	(●) Classification ( ) Detection
AutoNN Config	Dataset file : dataset.yaml Base Model : basemodel.yaml
Nas Type	( ) Backbone Nas (●) Neck Nas

신경망 자동 생성

Current Work - [ ]