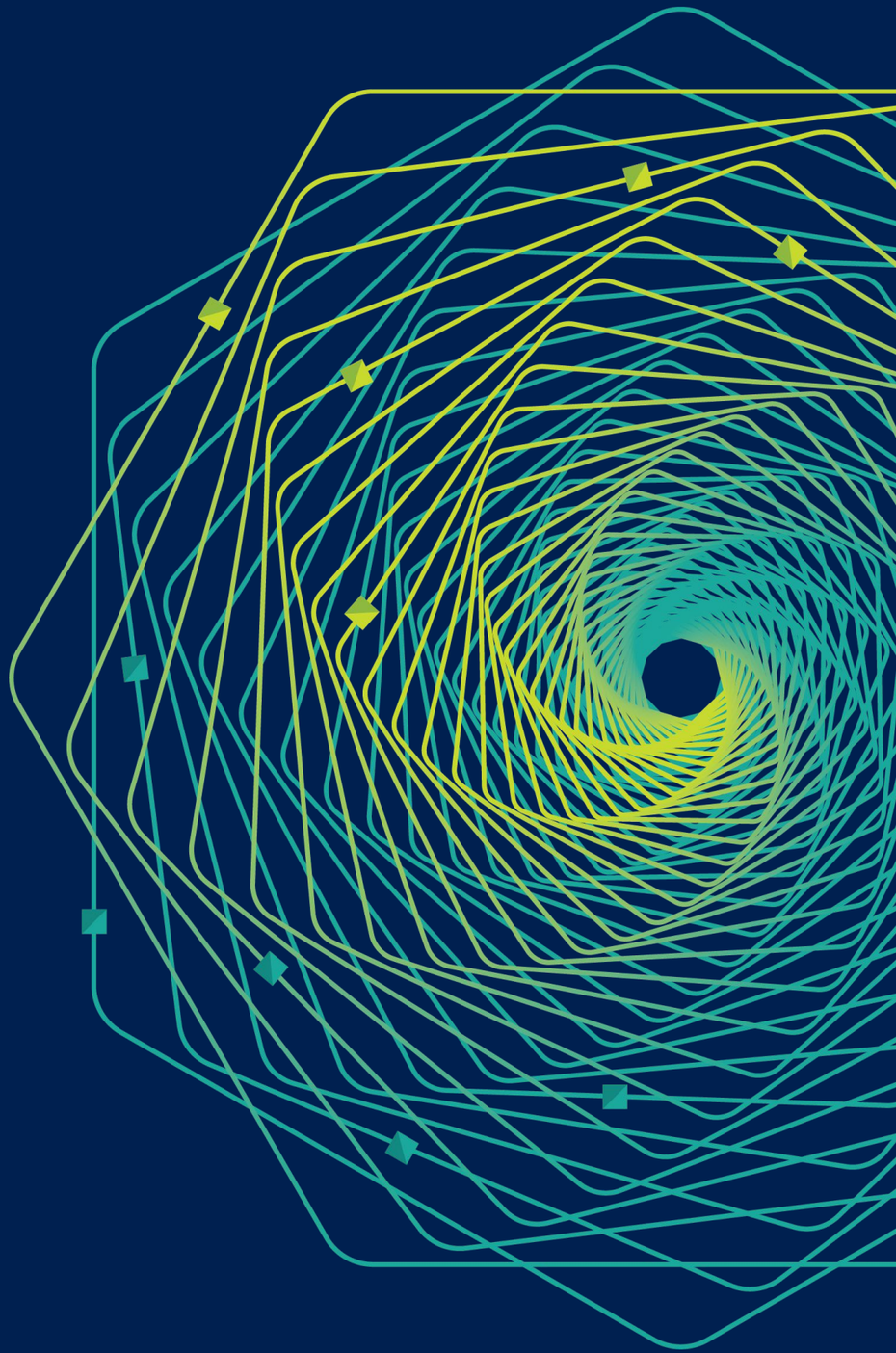
Microsoft



OpenPAI: The Open Source Initiative for AI Platform in China

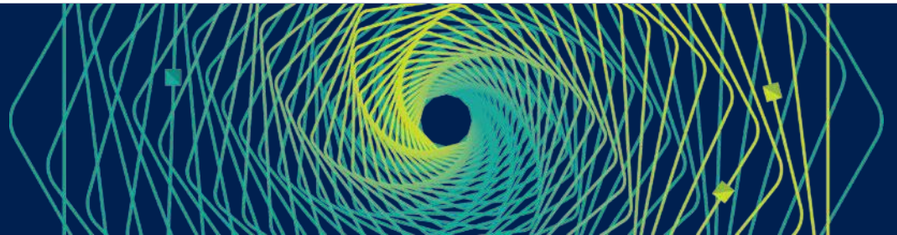
CQU CS1701

NLP Research Group

Background and Motivation

- Artificial Intelligence (AI) becomes one major focus and heated across academia and industries

- Major opportunity to democratize AI through innovations on AI infrastructure
  - Lower the entry bar for new comers
  - Facilitate AI education
  - Speed up AI research
  - Accelerate the penetration of AI across industries



Current Status and Challenges of AI Platform

- Still in early stage: ad hoc ways to build/deploy an AI platform
  - It is easy to build small-scale platforms with a narrowed, specific purpose

- Need an AI platform that works in different environment and application scenario
  - On-premise, cloud, and hybrid environment
  - Image/video, speech, language, vertical domain
  - High compatibility, extensibility, manageability, efficiency

The Importance of an AI Platform

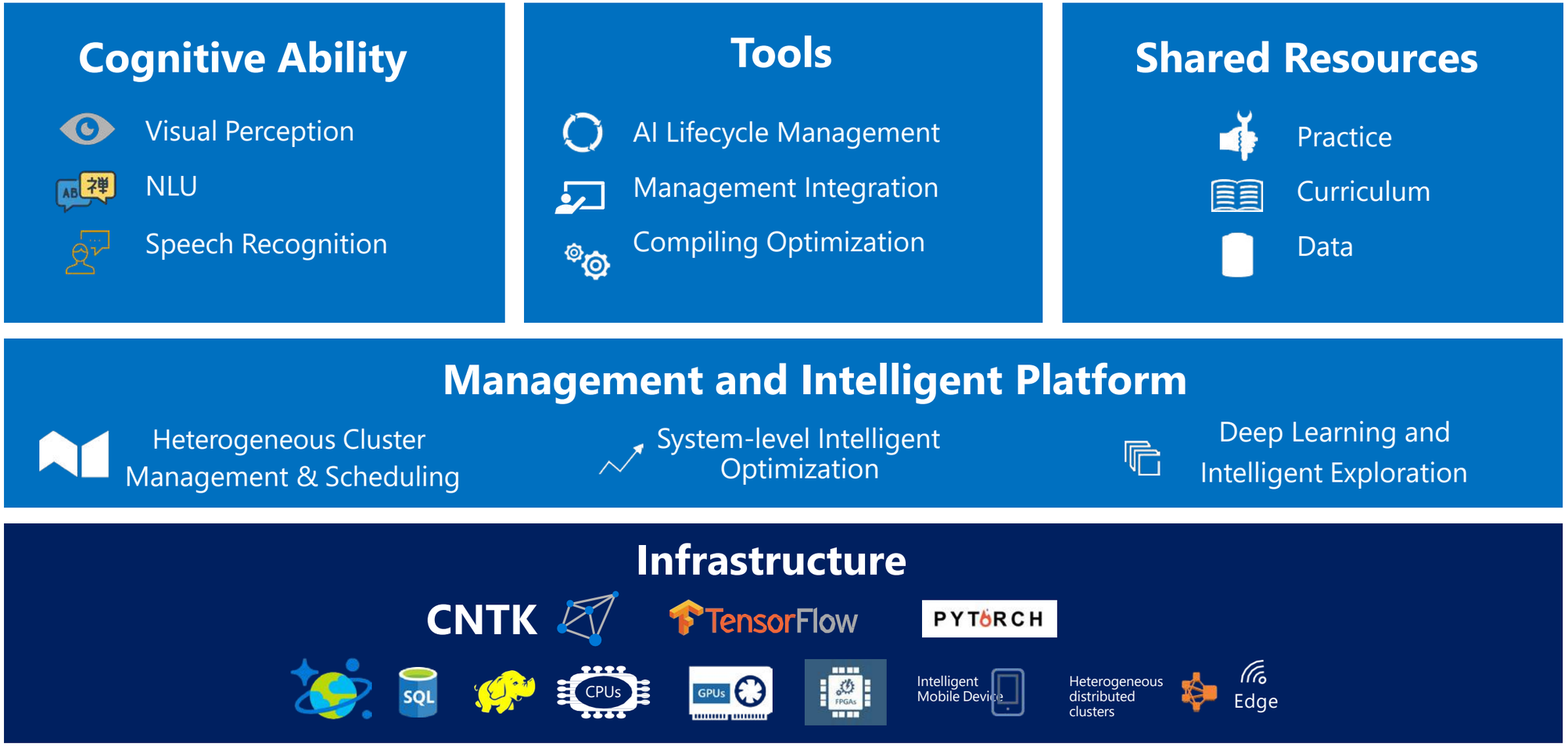
- Infrastructure support for the advance of artificial intelligence
  - Deep learning algorithms and frameworks run on
  - Manage hardware

- A platform to boost AI innovation and productivity
  - Allow researchers/practitioners to focus on AI innovation, instead of the hassles of infra. construction, deployment, management, and optimization
  - Enable results sharing, build a community for mutual learning/leveraging, and rapid innovation

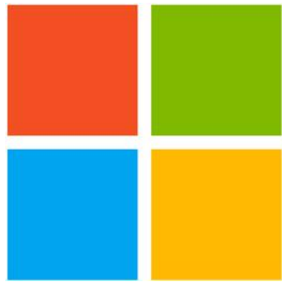
An Open Platform for AI R&D and Education

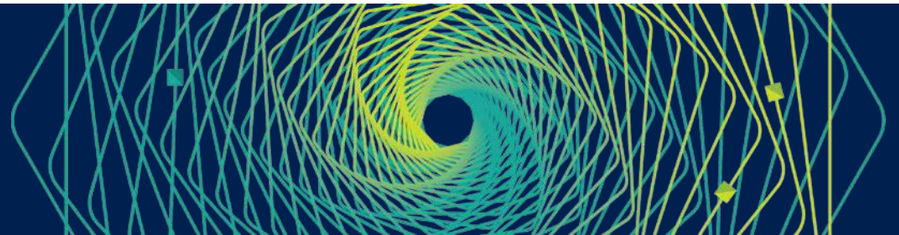
- The co-development of AI innovation, AI education, and AI platform evolution
  - Research, education, and production
- Design AI course project, perform training to grow AI talent pool
- Open source, result sharing, a community for collaborative innovation

Overview


Cognitive Ability
Visual Perception
NLU
Speech Recognition
Tools
AI Lifecycle Management
Management Integration
Compiling Optimization
Shared Resources
Practice
Curriculum
Data
Management and Intelligent Platform
Heterogeneous Cluster Management & Scheduling
System-level Intelligent Optimization
Deep Learning and Intelligent Exploration
Infrastructure
CNTK
TensorFlow
PYTORCH
SQL
CPUs
GPUs
FPGAs
Intelligent Mobile Device
Heterogeneous distributed clusters
Edge

China Open AI Platform Alliance

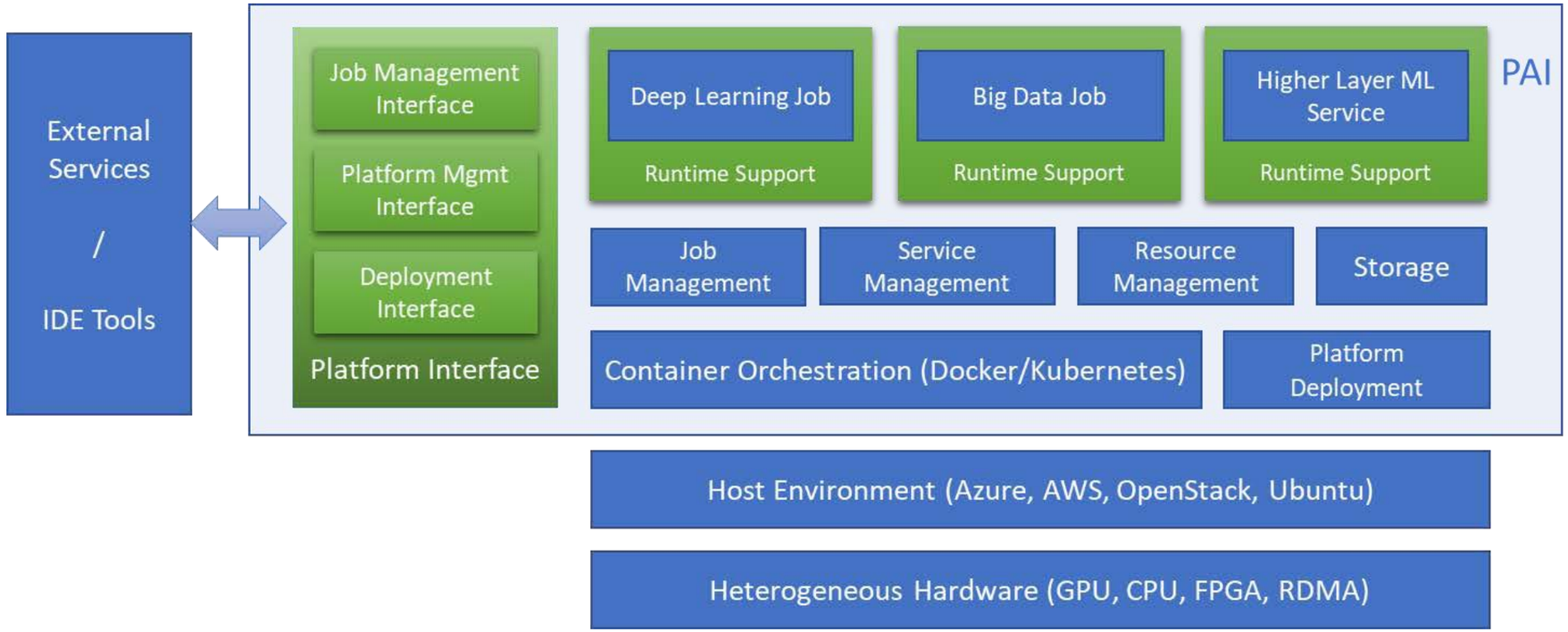




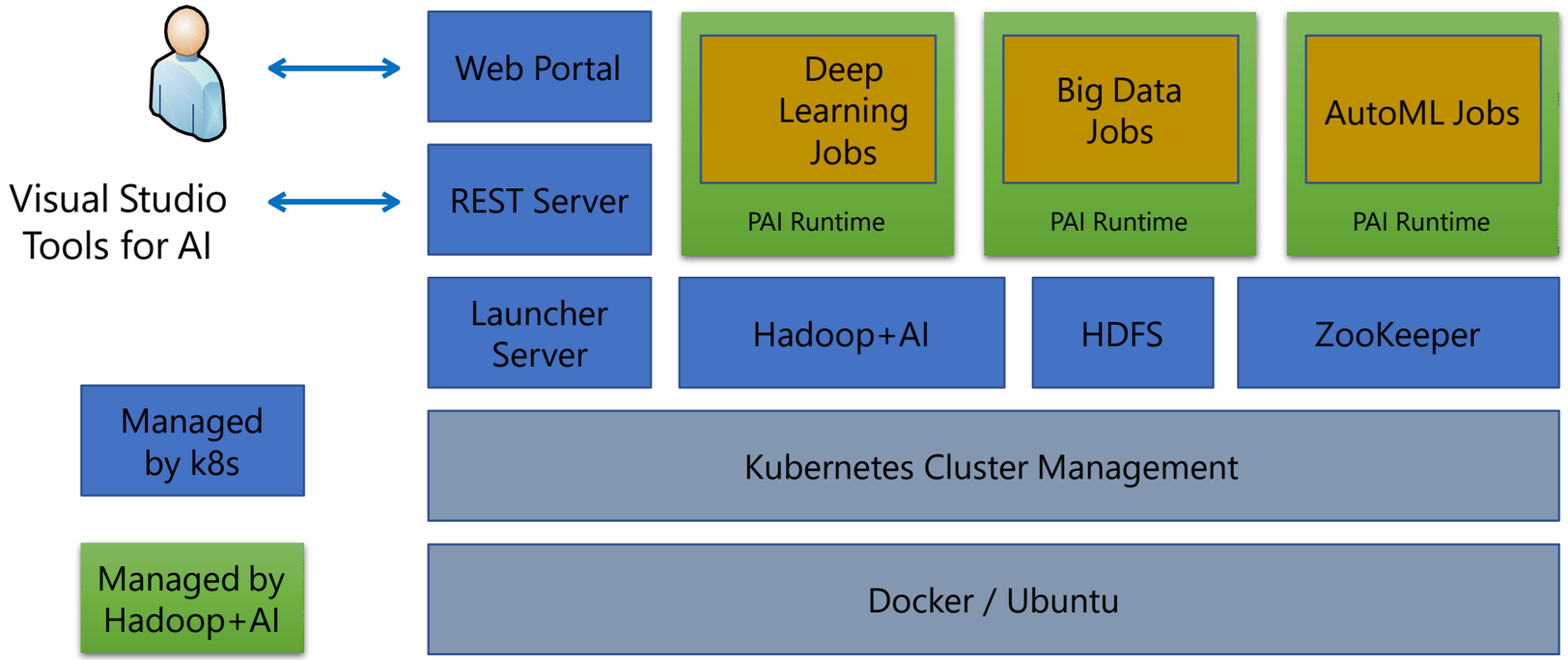
Open Platform for AI (OpenPAI)

- A platform for GPU cluster management
  - Openness: open source (MIT), open collaboration model
  - Extensibility: support all deep learning frameworks, GPU/FPGA/ASIC
  - Modularity: micro-service, different component choice (storage, scheduler)
  - Efficiency: fine-grained GPU scheduling, support IB/RDMA
  - Manageability：job and platform monitoring, deployment, upgrade, etc.
  - Robust: fault tolerance
  - Practicability: leverage the mature design and practice in Microsoft

OpenPAI Architecture



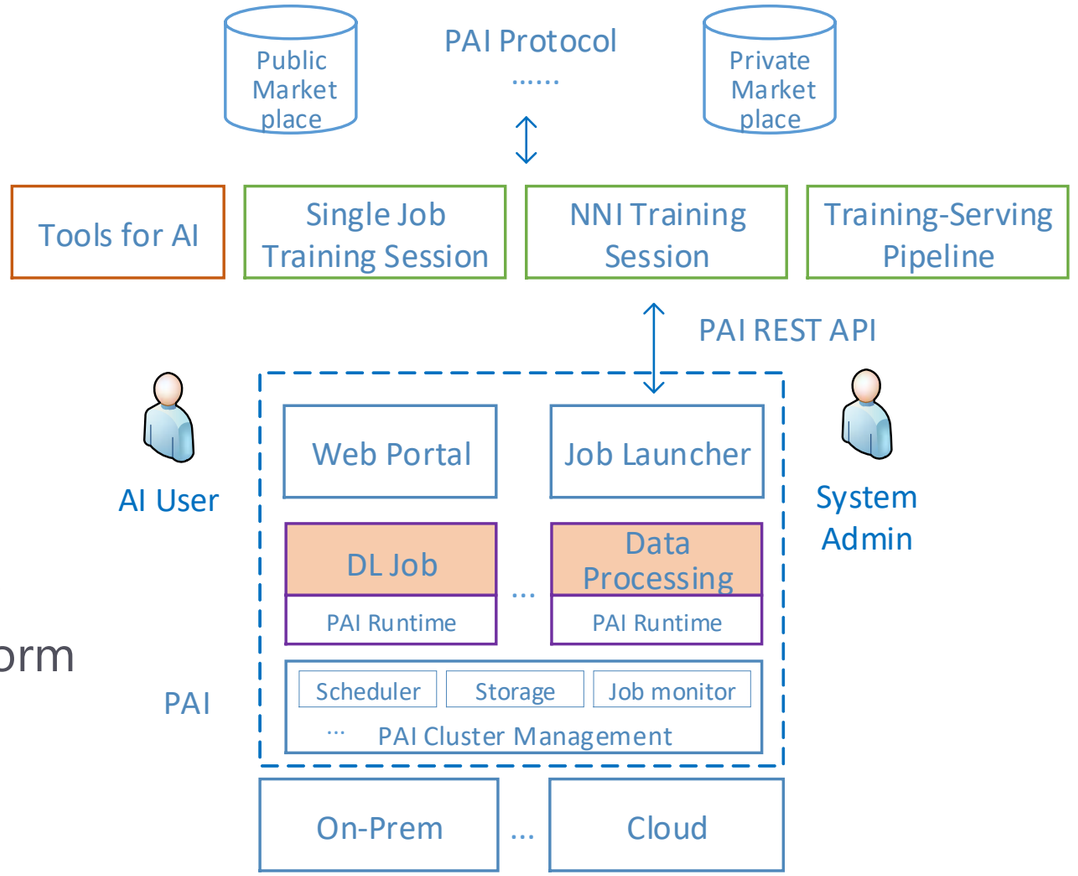
Implementation



https://github.com/Microsoft/pai

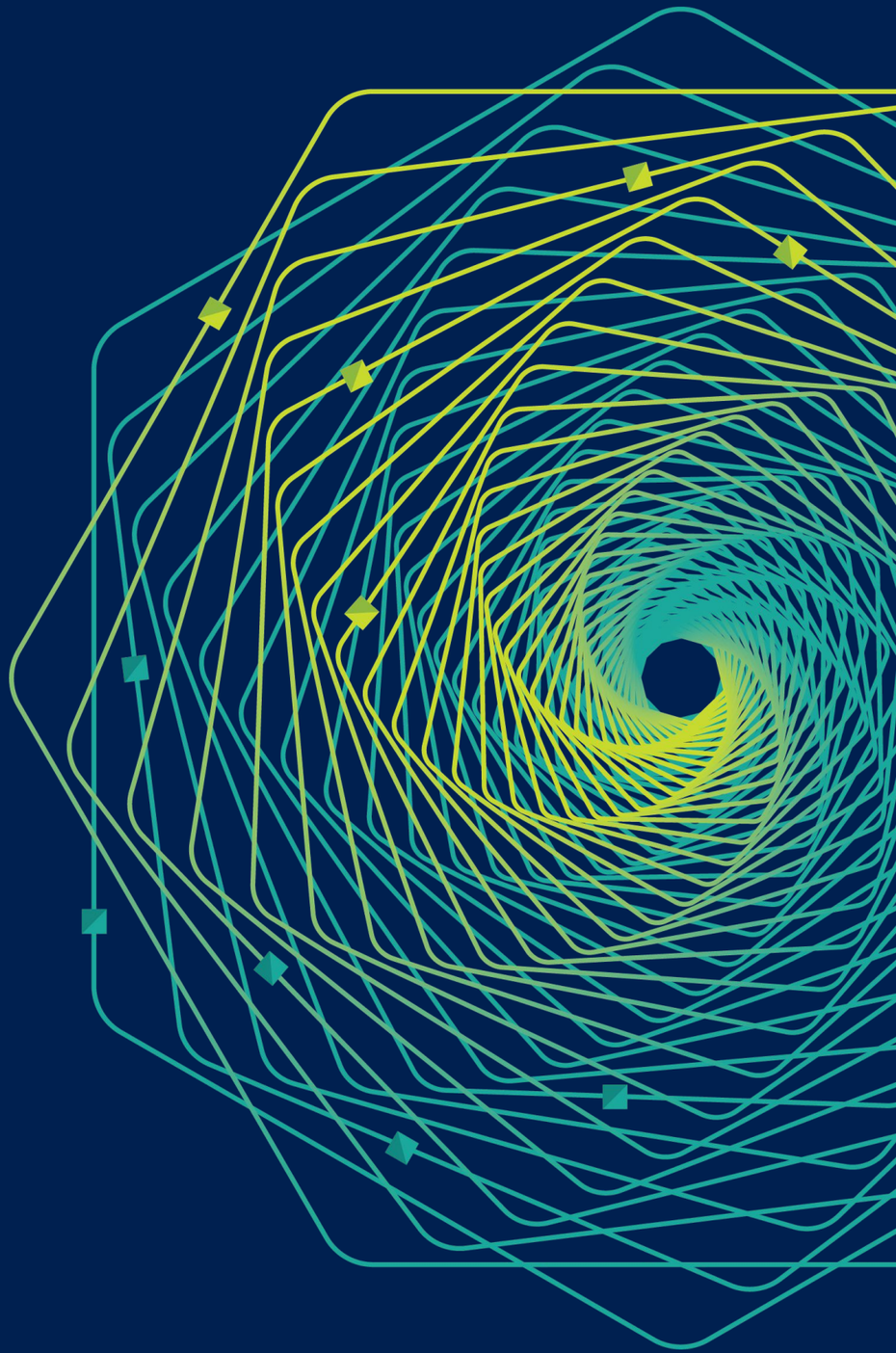
OpenPAI – Modularized (containerized) AI Platform

- PAI marketplace
  - AI asset sharing
- PAI protocol – resource specification
  - Data, code, docker image
  - Hardware requirement
- Job Launcher
  - Understand PAI protocol and execute the job accordingly
  - Onboard new AI workload w/o modifications to the platform
- Deployment in different environment
  - Single-box, cloud, on-prem, hybrid



An Open Ecosystem: Engage with China AI Community

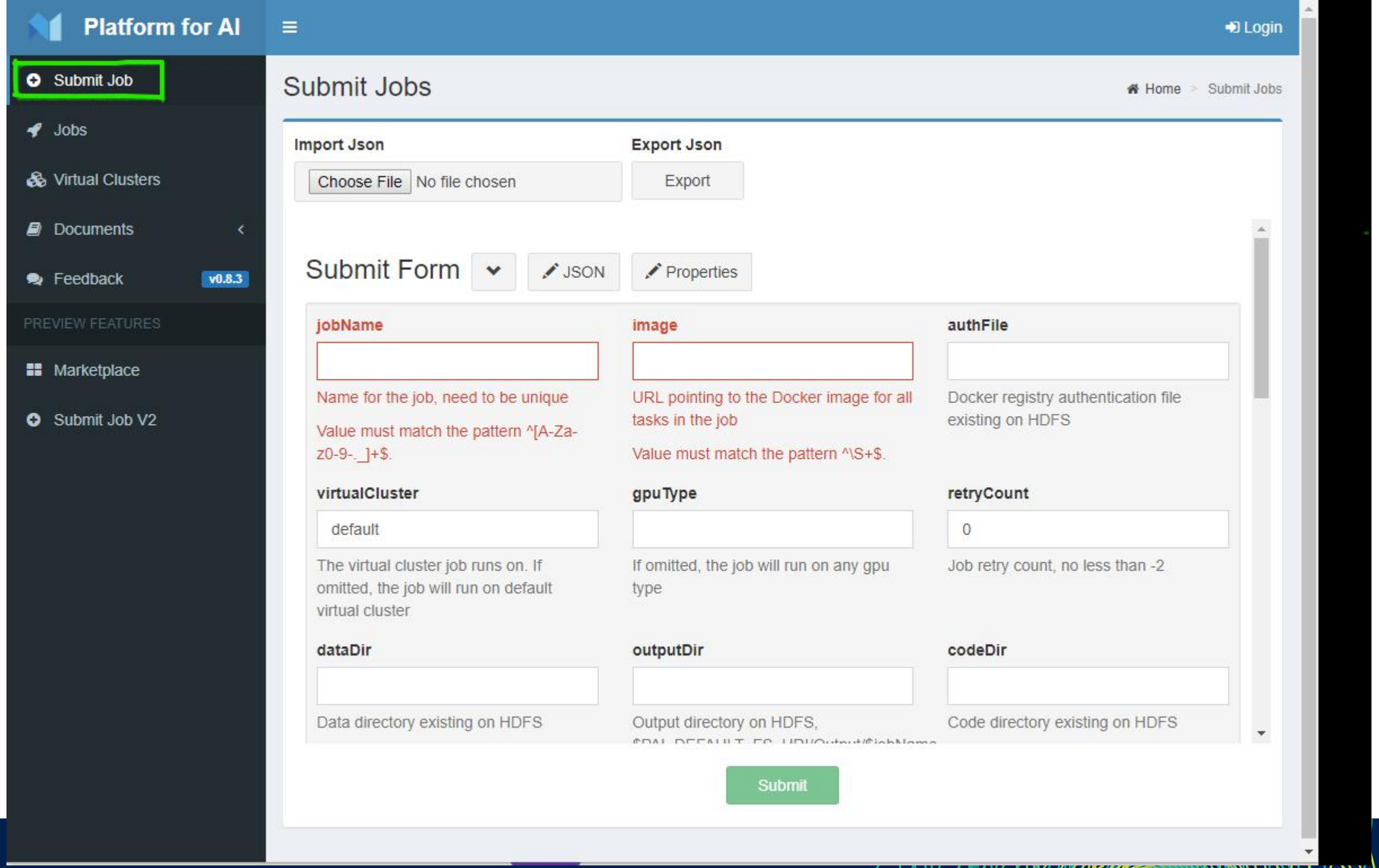
Microsoft



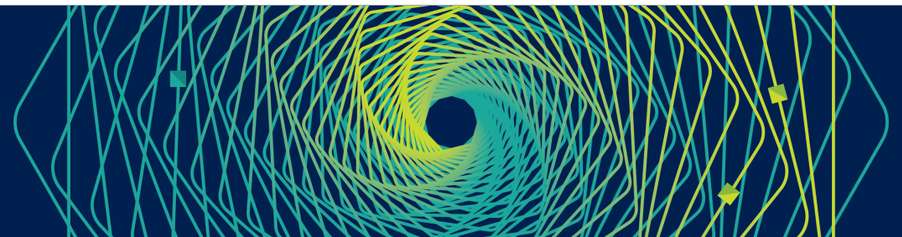
Train models on OpenPAI

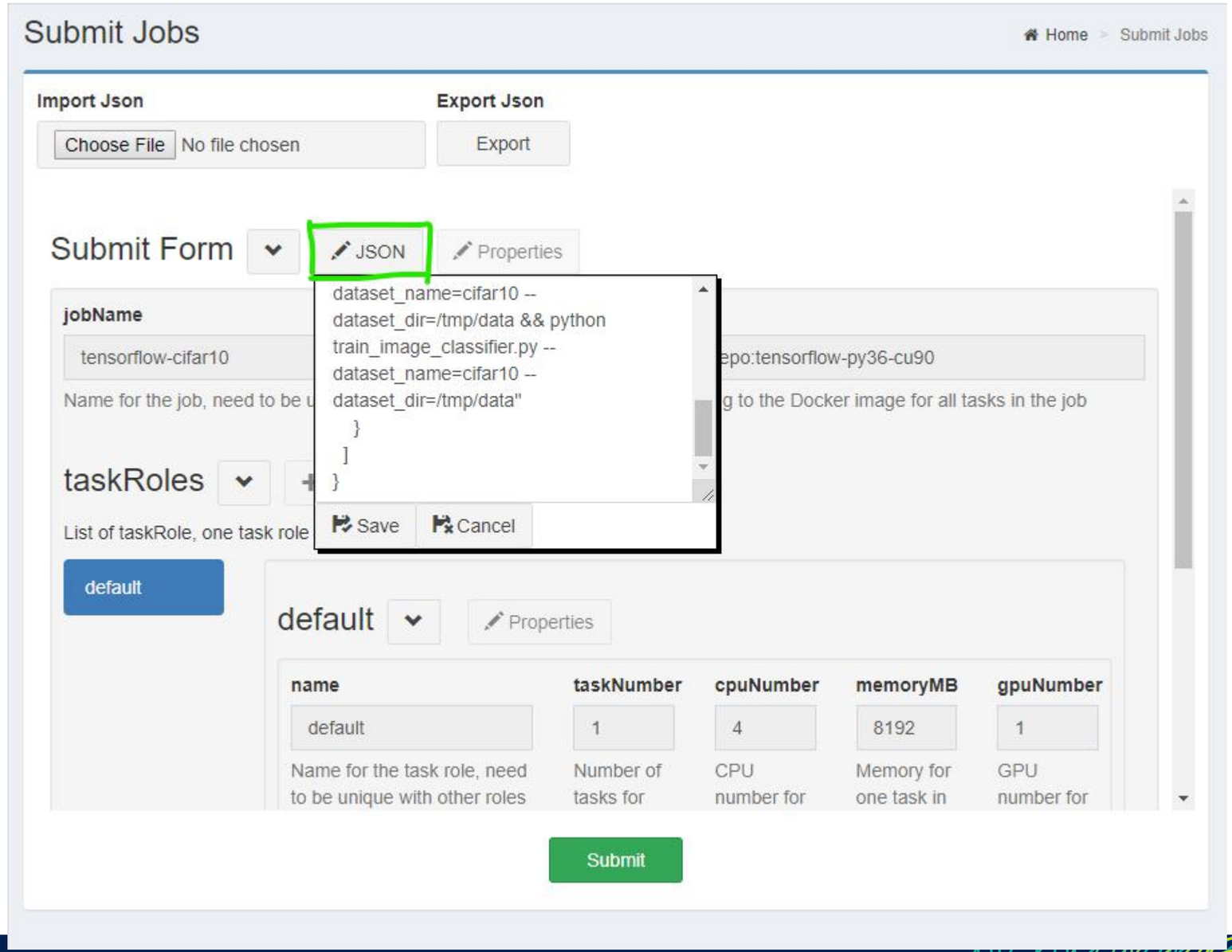
CQU CS1701

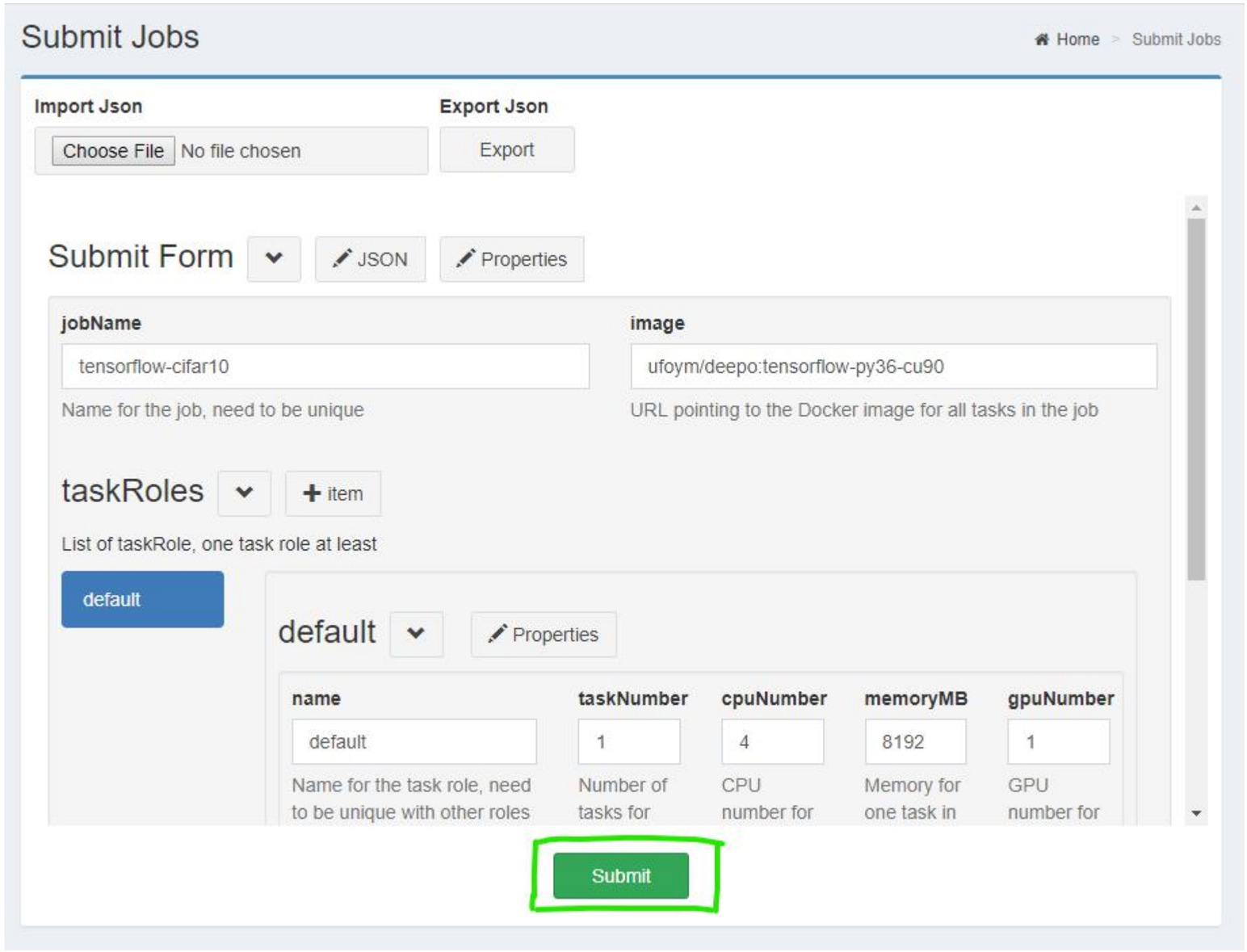
NLP Research Group


Platform for AI
Login
Submit Job
Jobs
Virtual Clusters
Documents
Feedback
v0.8.3
PREVIEW FEATURES
Marketplace
Submit Job V2
Submit Jobs
Home > Submit Jobs
Import Json
Choose File No file chosen
Export Json
Export
Submit Form
JSON
Properties
jobName
Name for the job, need to be unique
Value must match the pattern ^[A-Za-z0-9-._]+$.
image
URL pointing to the Docker image for all tasks in the job
Value must match the pattern ^\S+$.
authFile
Docker registry authentication file existing on HDFS
virtualCluster
default
The virtual cluster job runs on. If omitted, the job will run on default virtual cluster
gpuType
If omitted, the job will run on any gpu type
retryCount
0
Job retry count, no less than -2
dataDir
Data directory existing on HDFS
outputDir
Output directory on HDFS,
codeDir
Code directory existing on HDFS
Submit

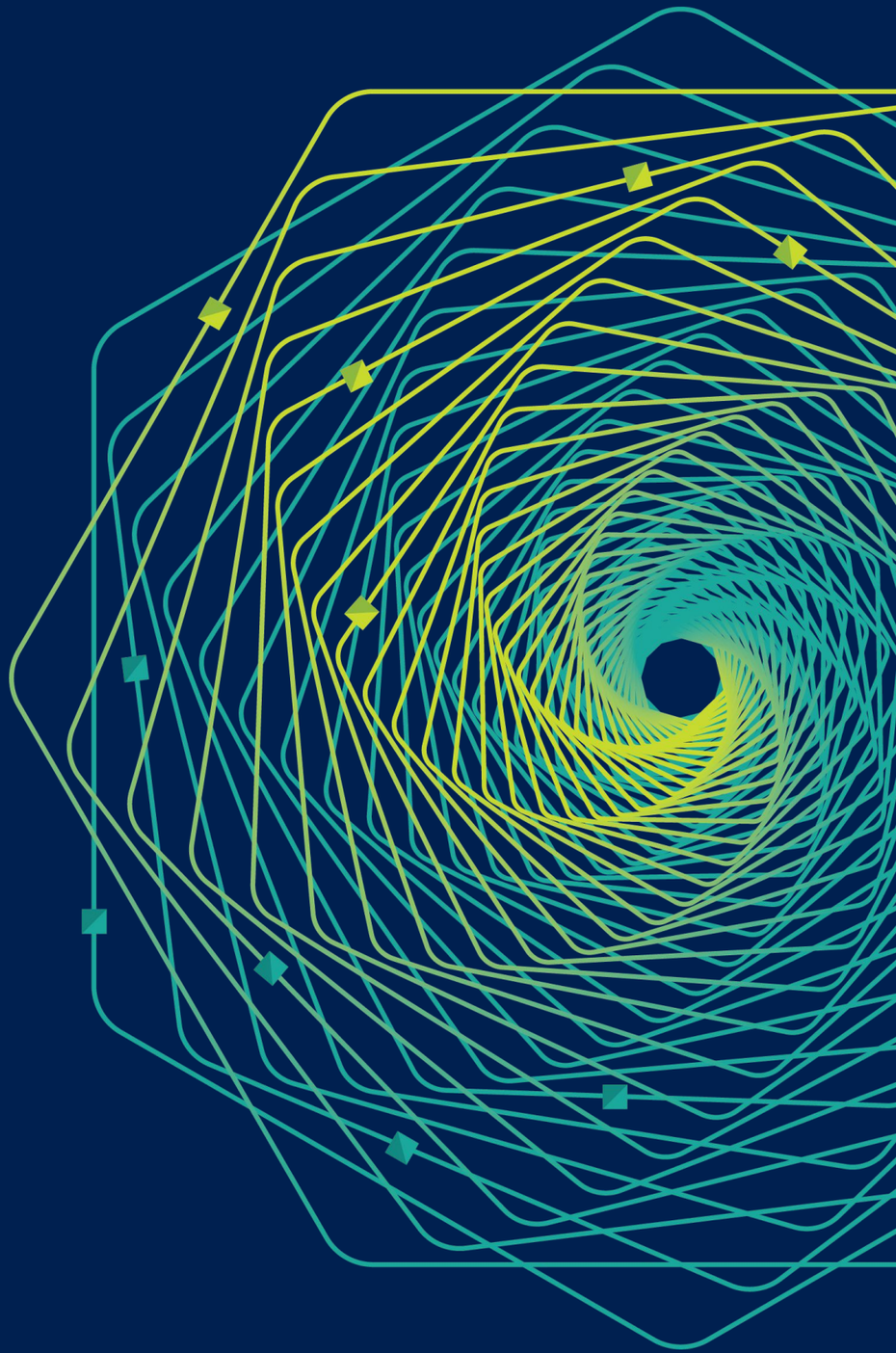
```
{
"jobName": "tensorflow-cifar10",
"image": "tensorflow/tensorflow:1.12.0-gpu-py3",
"taskRoles": [
    {
    "name": "default",
    "taskNumber": 1,
    "cpuNumber": 4,
    "memoryMB": 8192,
    "gpuNumber": 1,
    "command": "apt update && apt install -y git && git clone https://github.com/tensorflow/models && cd
models/research/slim && python download_and_convert_data.py --dataset_name=cifar10 --dataset_dir=/tmp/data &&
python train_image_classifier.py --dataset_name=cifar10 --dataset_dir=/tmp/data --max_number_of_steps=1000"
    }
]
}
```




Submit Jobs
Home > Submit Jobs
Import Json
Choose File No file chosen
Export Json
Export
Submit Form
JSON
Properties
dataset_name=cifar10 --
dataset_dir=/tmp/data && python
train_image_classifier.py --
dataset_name=cifar10 --
dataset_dir=/tmp/data"
}
]
}
Save
Cancel
jobName
tensorflow-cifar10
epo:tensorflow-py36-cu90
Name for the job, need to be u
g to the Docker image for all tasks in the job
taskRoles
List of taskRole, one task role
default
default
Properties
name
taskNumber
cpuNumber
memoryMB
gpuNumber
default
1
4
8192
1
Name for the task role, need to be unique with other roles
Number of tasks for
CPU number for
Memory for one task in
GPU number for
Submit


Submit Jobs
Home > Submit Jobs
Import Json
Choose File No file chosen
Export Json
Export
Submit Form
JSON
Properties
jobName
tensorflow-cifar10
Name for the job, need to be unique
image
ufoym/deepo:tensorflow-py36-cu90
URL pointing to the Docker image for all tasks in the job
taskRoles
item
List of taskRole, one task role at least
default
default
Properties
name
default
Name for the task role, need to be unique with other roles
taskNumber
1
Number of tasks for
cpuNumber
4
CPU number for
memoryMB
8192
Memory for one task in
gpuNumber
1
GPU number for
Submit

Microsoft



Build a Docker Image

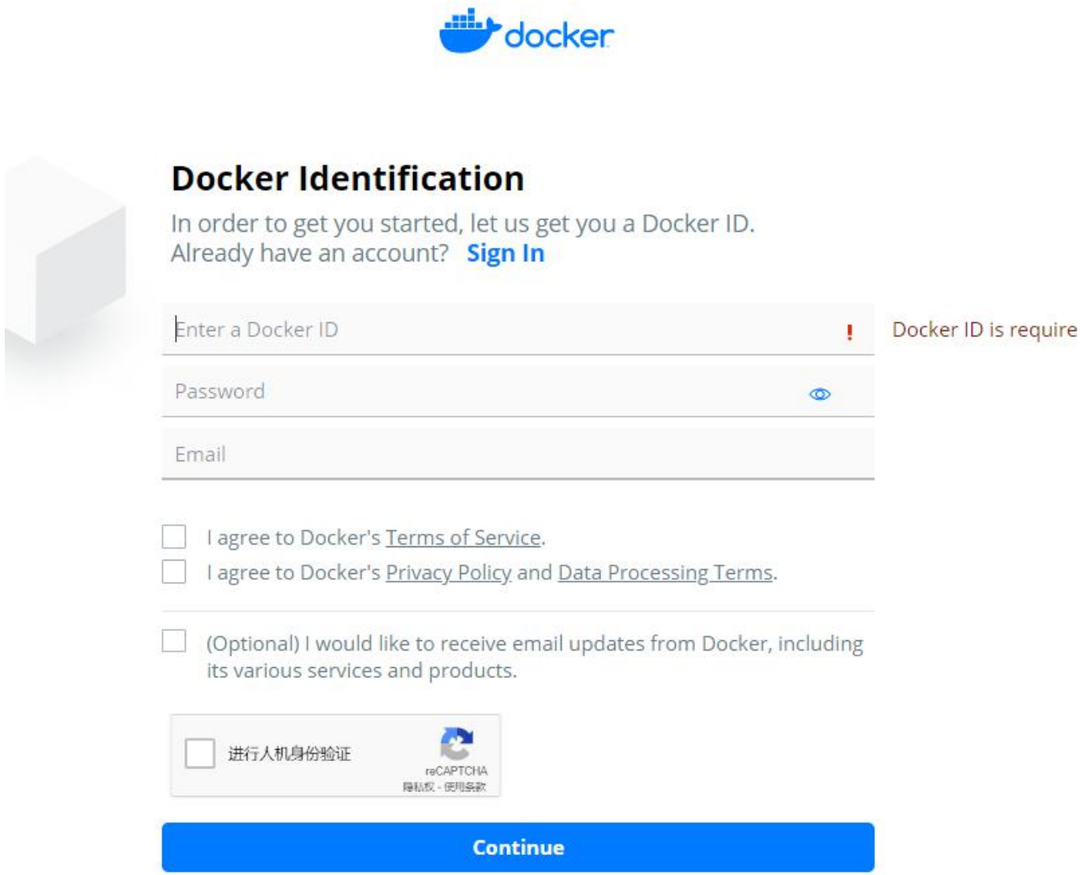
CQU CS1701

NLP Research Group

Regist a docker-hub account

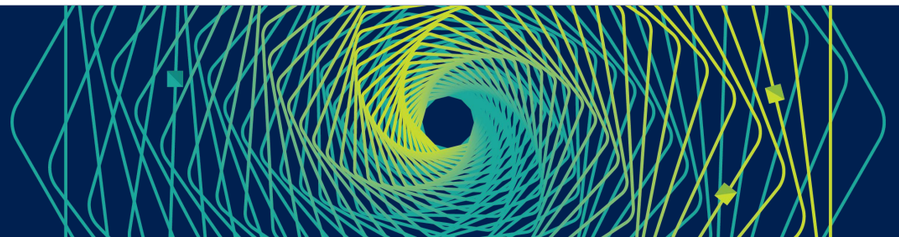
Regist url: https://hub.docker.com/signup

Sign in url: https://hub.docker.com/signup



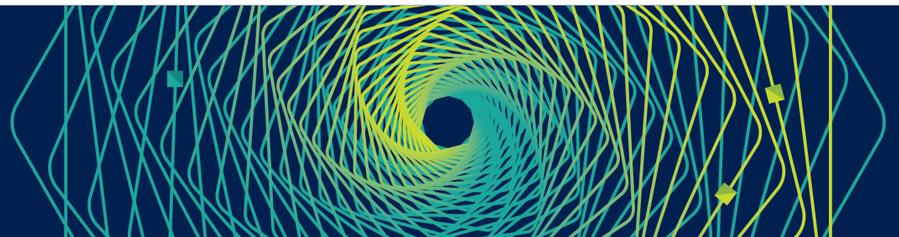
Install Docker-CE(Ubuntu)

- Step0: Install the Windows Subsystem for Linux(If use windows10)
- Step1: Uninstall old versions
- Step2: Install Docker Engine – Community
  - Install using the repository
  - Install from a package



Install the Windows Subsystem for Linux

1. Open PowerShell as Administrator and run:
   - Enable-WindowsOptionalFeature -Online -FeatureName Microsoft-Windows-Subsystem-Linux
2. Restart your computer when prompted.
3. Download Ubuntu and install from the Microsoft
4. Run Ubuntu Subsystem



Microsoft Store

Microsoft Store

Home Apps Games Devices Movies & TV Books Edge Departments Search

Run Linux on Windows

Install and run Linux distributions side-by-side on the Windows Subsystem for Linux (WSL).

Ubuntu
★★★★½
Installed

openSUSE Leap 42
★★★★☆
Installed

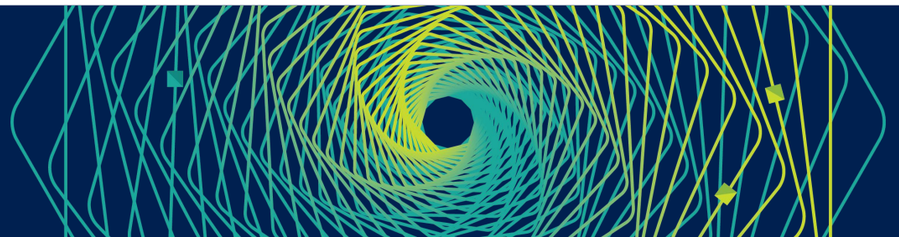
SUSE Linux Enterprise Server 12
★★★★☆
Owned

Debian GNU/Linux
★★★★☆
Installed

Kali Linux
★★★★☆
Owned

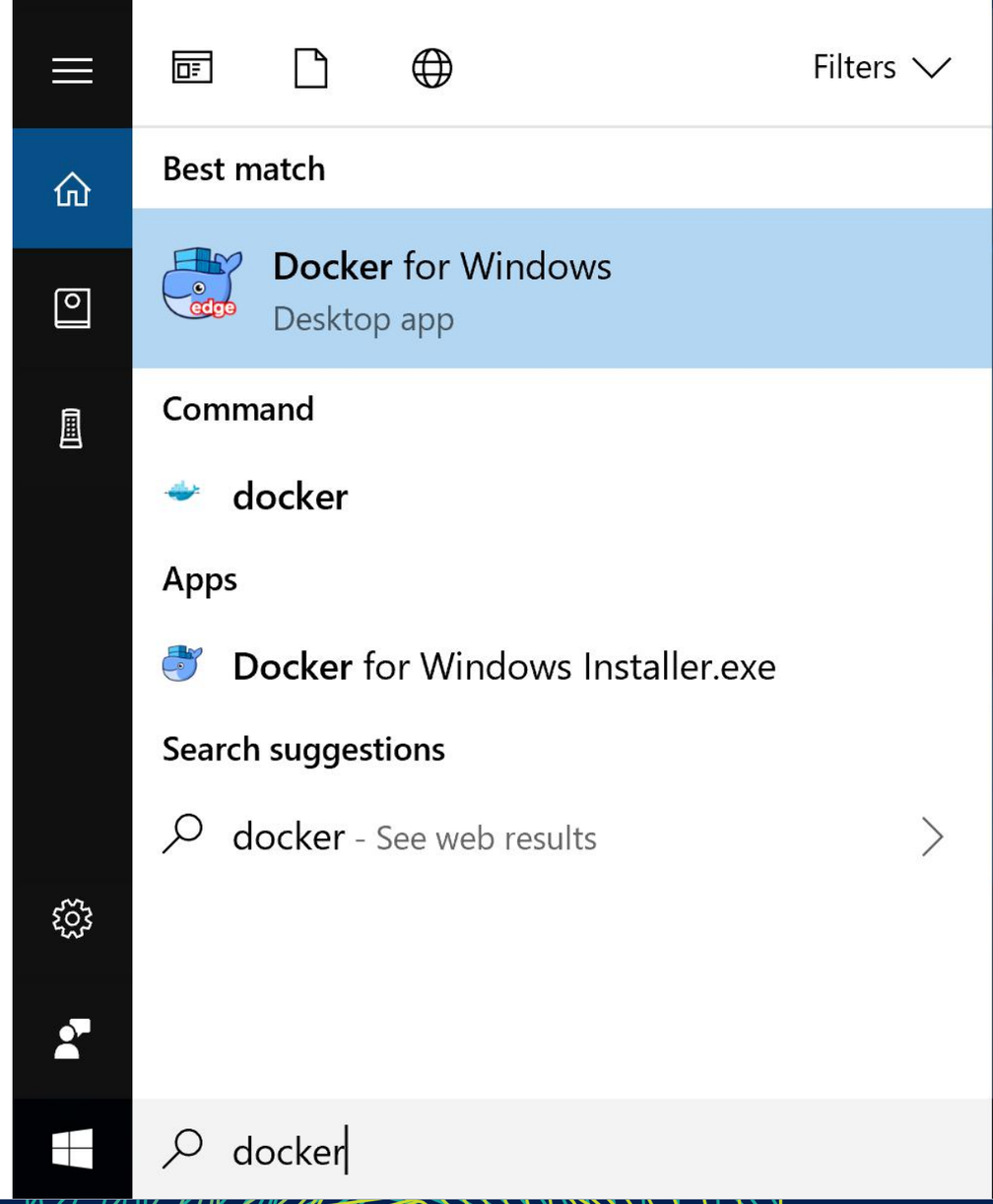
Install from a package

1. Go to https://download.docker.com/linux/ubuntu/dists/, choose your Ubuntu version, browse to pool/stable/, choose amd64 and download the .deb file for the Docker Engine - Community version you want to install.
2. Install Docker Engine - Community, changing the path below to the path where you downloaded the Docker package.
   - sudo dpkg -i /path/to/package.deb
3. Verify that Docker Engine - Community is installed correctly by running the hello-world image.
   - sudo docker run hello-world



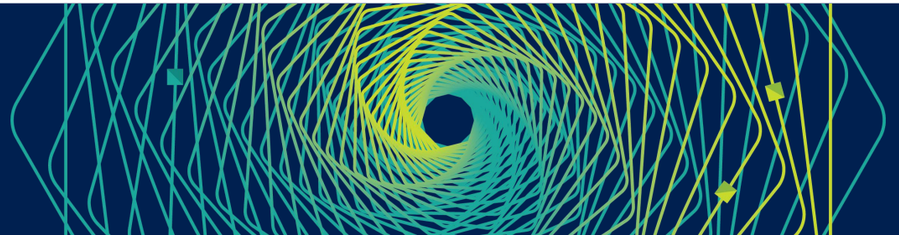
Install Docker Desktop(Windows)

- Download Docker Desktop(CE version)
  - https://hub.docker.com/editions/community/docker-ce-desktop-windows
- Install Docker Desktop for Windows desktop app
- Start Docker Desktop for Windows
  - Run Docker
  - Input command in Powershell



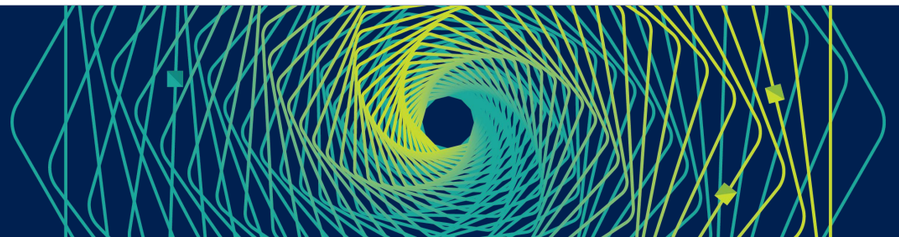
Build your own docker environment

- Download base Dockerfile
- Build Docker image
  - cd ../Dockerfiles/cuda8.0-cudnn6
  - sudo docker build -f Dockerfile.build.base -t pai.example.jupyter .
- Push the Docker image to Docker-Hub
  - sudo docker tag pai.example.jupyter USER/pai.example.jupyter
  - sudo docker push USER/pai.example.jupyter



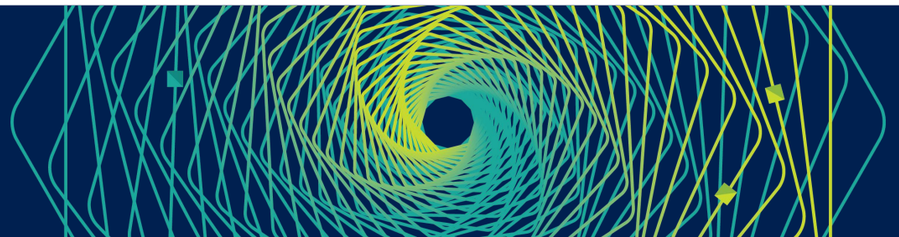
Use your own Docker image to submit jobs

```
{
  "jobName": "test_store",
  "image": "lvyufengh/pai.example.jupyter",
  "virtualCluster": "default",
  "taskRoles": [
    {
      "name": "test_store",
      "taskNumber": 1,
      "cpuNumber": 4,
      "memoryMB": 32000,
      "gpuNumber": 1,
      "command": "sleep infinity"
    }
  ]
}
```



Some commands

- apt install nfs-common && mkdir /models && mount -t nfs4 <server address>:<server path> /models
- showmount -e 202.202.5.138



Thank you!