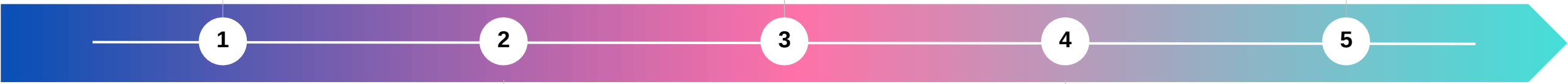
Model Training & Testing Guideline

MindSpore Challenge - Pathology Diagnosis

Competition Timeline



1

Training Workshop

Using personal laptop to train and test on the public data

2

3

Model Training

Using Huawei ModelArts to train the dataset

4

Model Evaluation

Submit code and checkpoint to MSC21 platform for model testing

5

Final Submission

Participants should upload their final source code

Final Pitching

Present to the Judges for the final round evaluation and scoring

Please refer to rule books for criteria

Model Training Stage

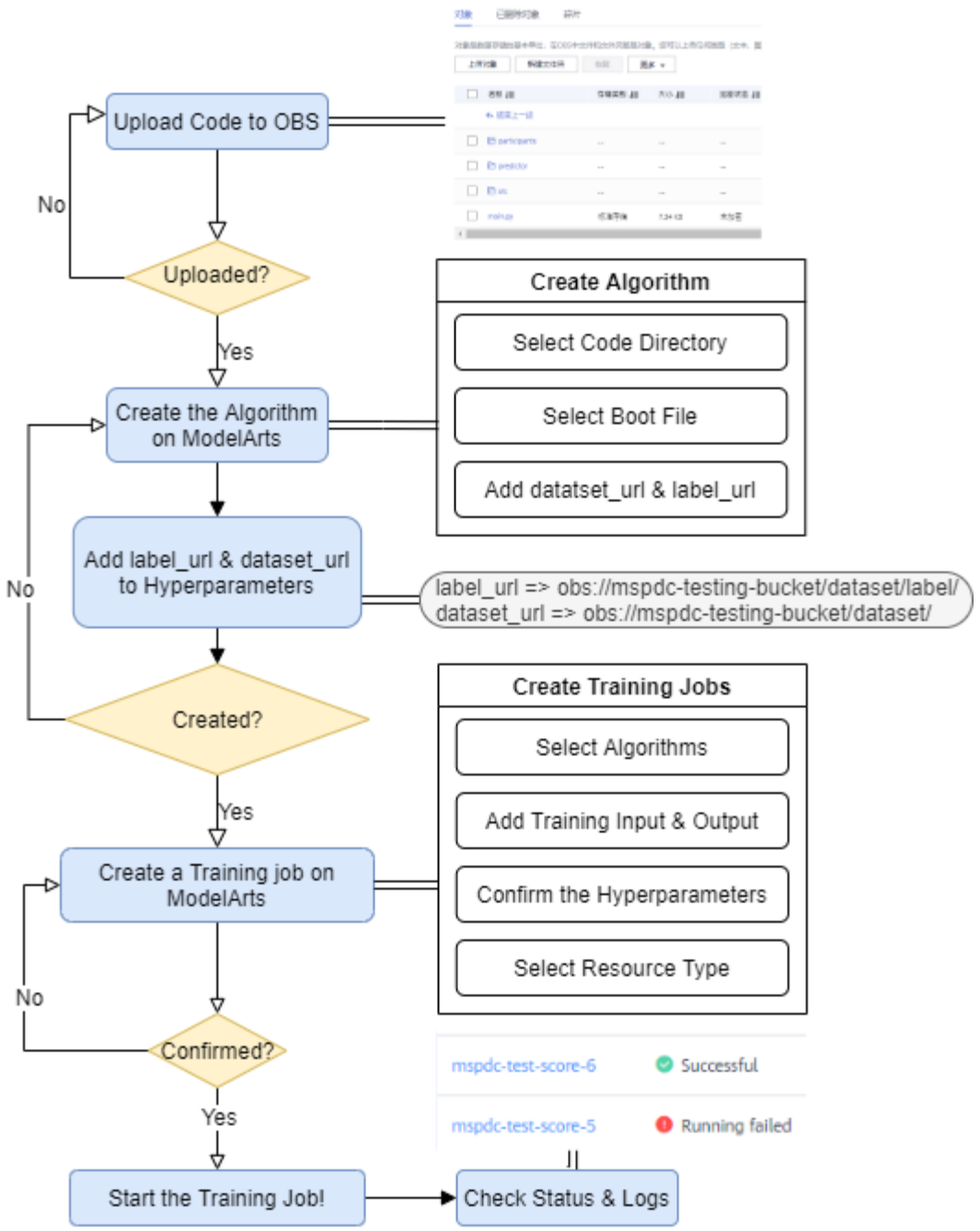
Detail about the training procedure on ModelArts

Training Flow Overview

Huawei Cloud	It's a **cloud service provider** with the features of data storage and AI Platform
ModelArts	A one-stop **development platform for AI** developers from data to output
OBS	**Cloud storage service** optimized for storing massive amounts of data

Main Procedures

1. **Upload code to OBS**
2. **Create Algorithm on Huawei Cloud ModelArts**
3. **Create Training Jobs on Huawei Cloud ModelArts**
4. **Start training & Post-Training**

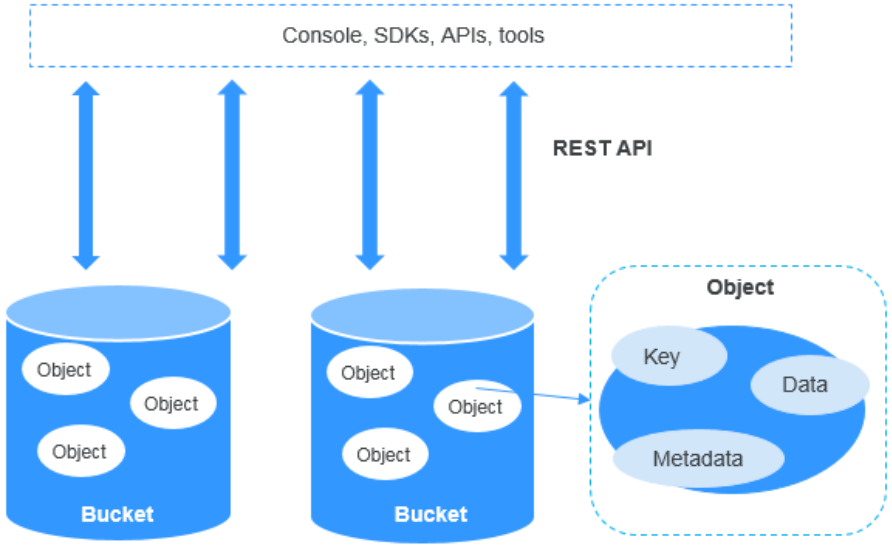


/01 Upload Code to OBS

1. What & Why OBS?
2. How to create a bucket and upload files?
3. How to manage the file structure

What & Why OBS?

- OBS Official Document
- Best OBS Practices
- Train a model with OBS



01

Cloud Storage

- Object Storage Service (OBS) is a **cloud storage service** optimized for storing massive amounts of data.
- It can **store your Code/ Dataset/ Training Output** and so on.

02

Bucket and Object

- **Bucket** = **container** for storing objects in OBS, which has specific storage class.
- **Object** = basic unit of **data** storage in OBS, which consists of a **key, metadata, and data.**

03

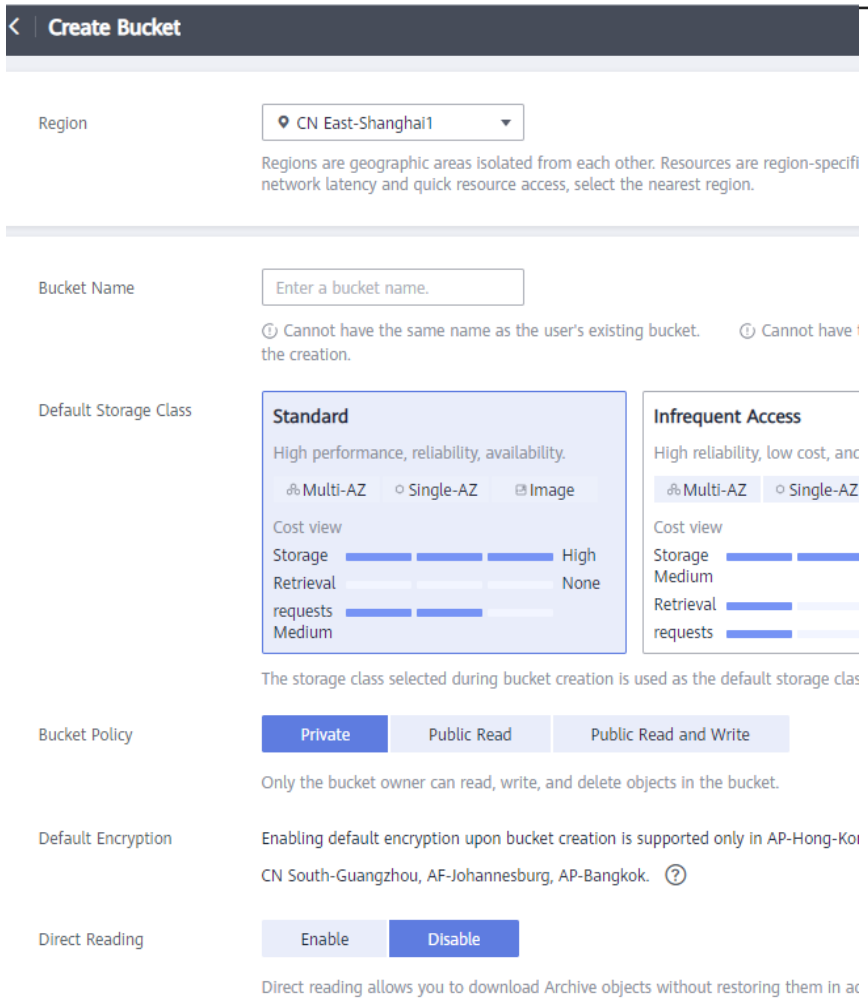
Why to use it?

- **Upload your training code** here for the connection to ModelArts!
- **Store output** of your **checkpoint, weight** from the trained models!
- Easily manage all your resources on cloud!

Create Bucket and Upload Files

Create a Bucket

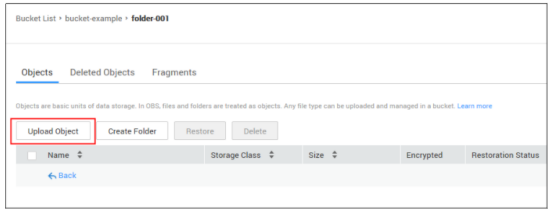
OBS Console > **Object Storage** > **Create Bucket** (upper right)



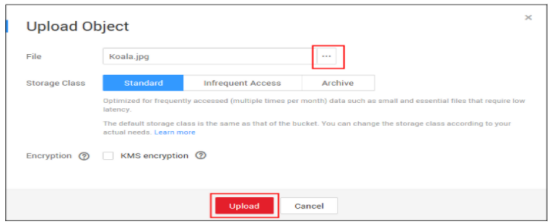
Upload to Bucket

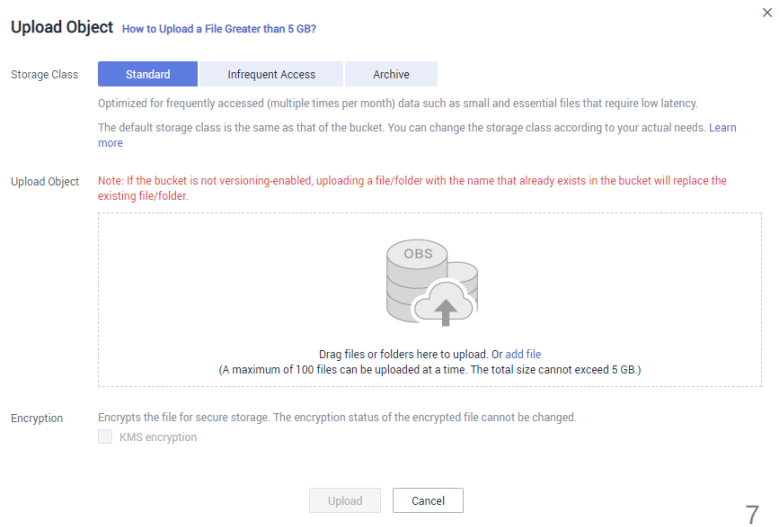
Object Storage > Overview > Objects > Upload Object

1 Uploading a file.



2 Selecting a file and uploading it.

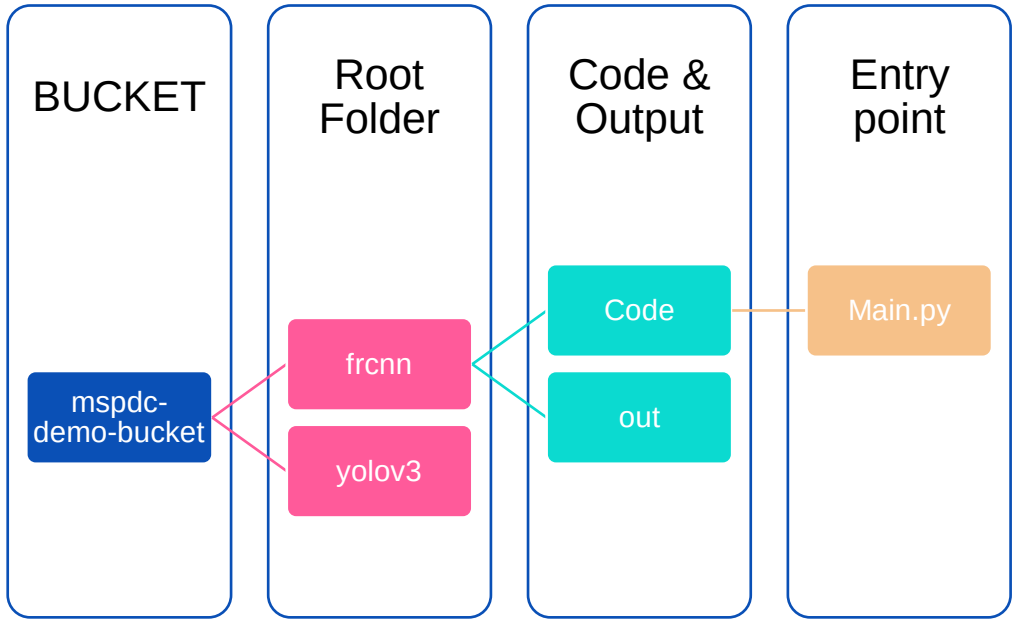




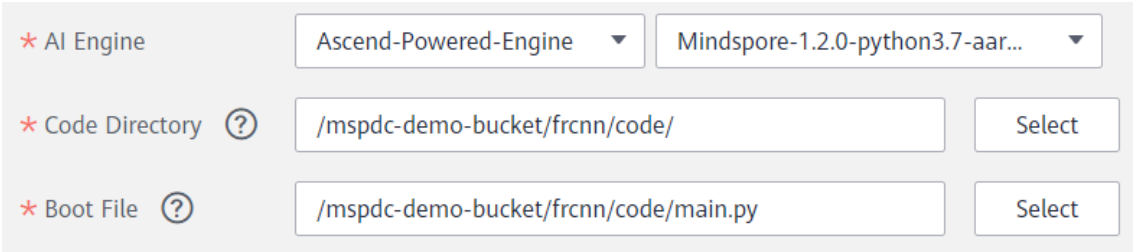
How to Manage the File Structure

Since Model Arts Algorithm creation **need to know your source code position**…

- Using different main folder to **separate different algorithms**
- There should be **one entry point** to handle parameters



Model Arts Algorithm



/02 Create Algorithm on ModelArts

1. What is ModelArts
2. How to create an Algorithm

What is ModelArts?

- ModelArts is a one-stop **development platform** for AI developers
- **MoXing deep learning framework** accelerates training
- Optimized **GPU & Ascend AI** chips achieve more efficient inference

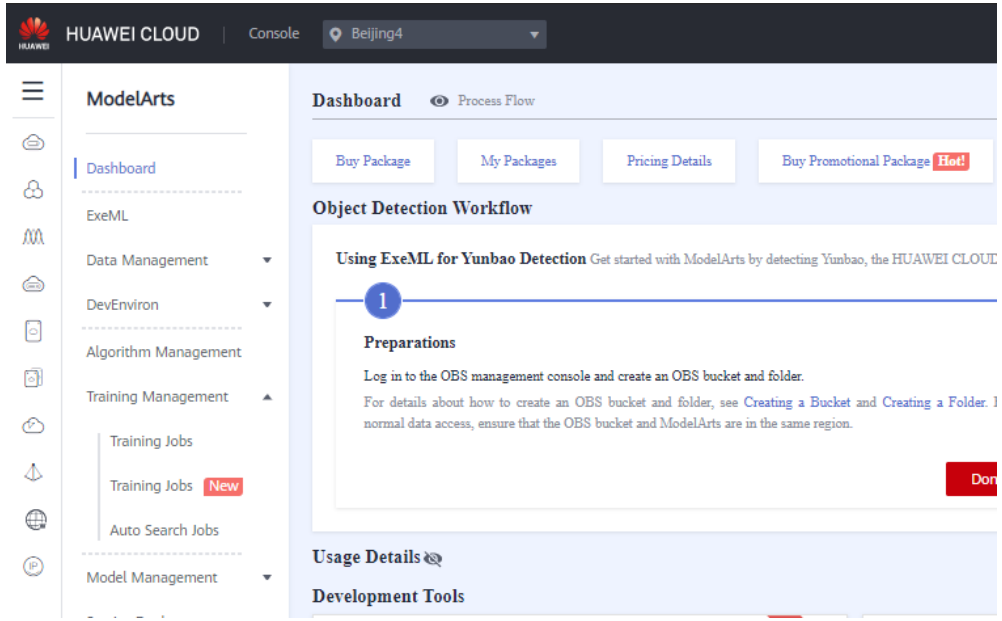
AI development lifecycle	Data management	• Datasets are stored in OBS. • The dataset labeling information is stored in OBS. • Data can be imported from OBS.
	Development environment	Data or code files in a notebook instance are stored in OBS.
	Model training	• The datasets used by training jobs are stored in OBS. • The running scripts of training jobs are stored in OBS. • The models generated by training jobs are stored in the specified OBS paths. • The run logs of training jobs are stored in the specified OBS paths.
	Model management	After a training job is completed, the generated model is stored in OBS. You can import the model from OBS.
	Service deployment	The models stored in OBS can be deployed as services.

What will we do on ModelArts?

- **Create an Algorithm**

  Upload locally developed algorithms or algorithms developed using other tools to ModelArts for unified management
- **Create a Training Job Using an Algorithm**

  Quickly create a training job on ModelArts to obtain the desired model using the algorithm you have created



- [Introduction to AI Lifecycle]
- [ModelArts Official Document]
- [ModelArts Algorithm Management]
- [ModelArts Model Training Management]
- [Moxing Framework Guideline]

How to Create an Algorithm?

Algorithm Management > My Algorithms > Create

01. Basic Information

- Name
- Description

02.Creating Way (Custom Script)

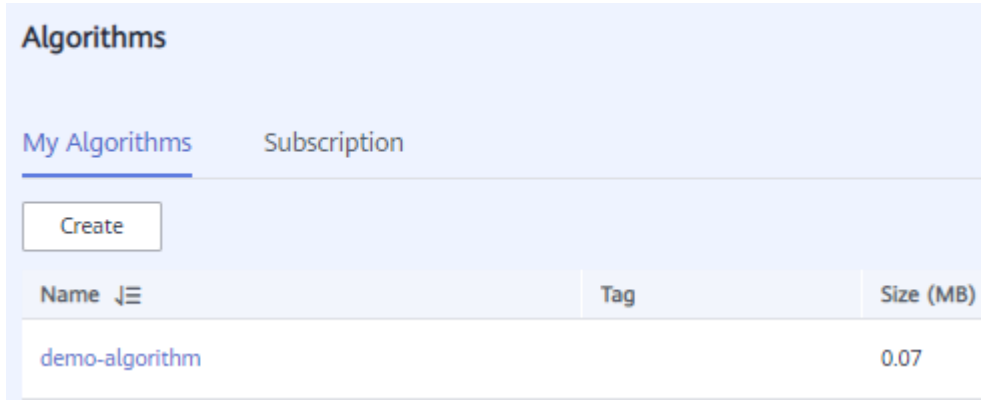
- AI Engine
- Code Directory + Boot File

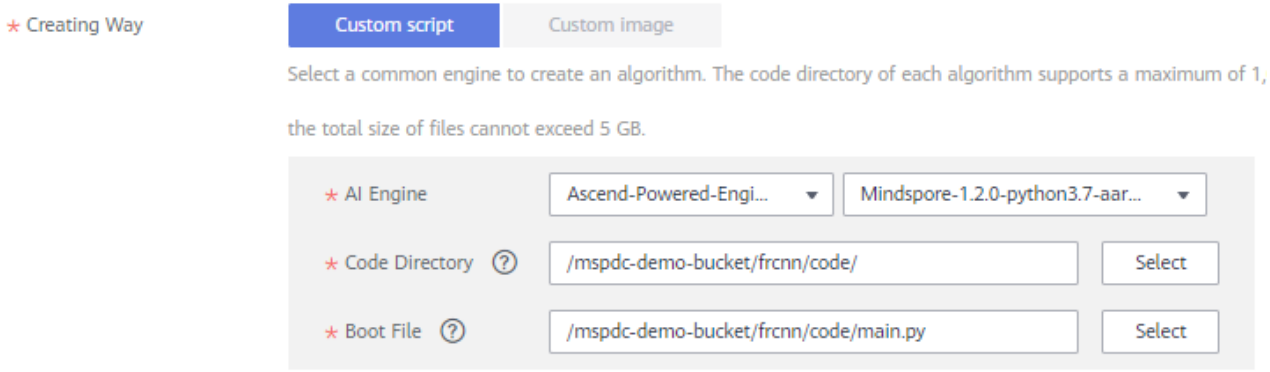
03. Hyperparameters

- dataset_url => to training images dataset
- label_url => to training images' label

04. In/Out Path Configuration

- Input path mapping
- Output path mapping (make sure the folder exists)





How to Create an Algorithm?

Algorithm Management > My Algorithms > Create

01. Basic Information

- Name
- Description

02.Creating Way (Custom Script)

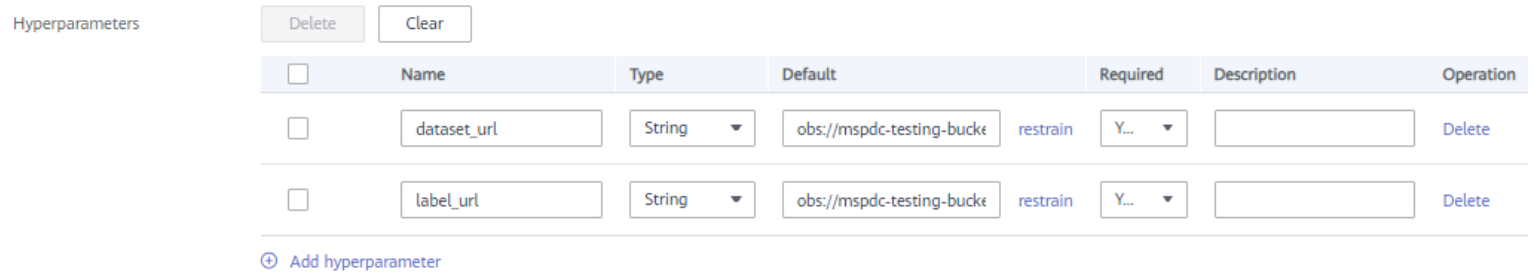
- AI Engine
- Code Directory + Boot File

03. Hyperparameters

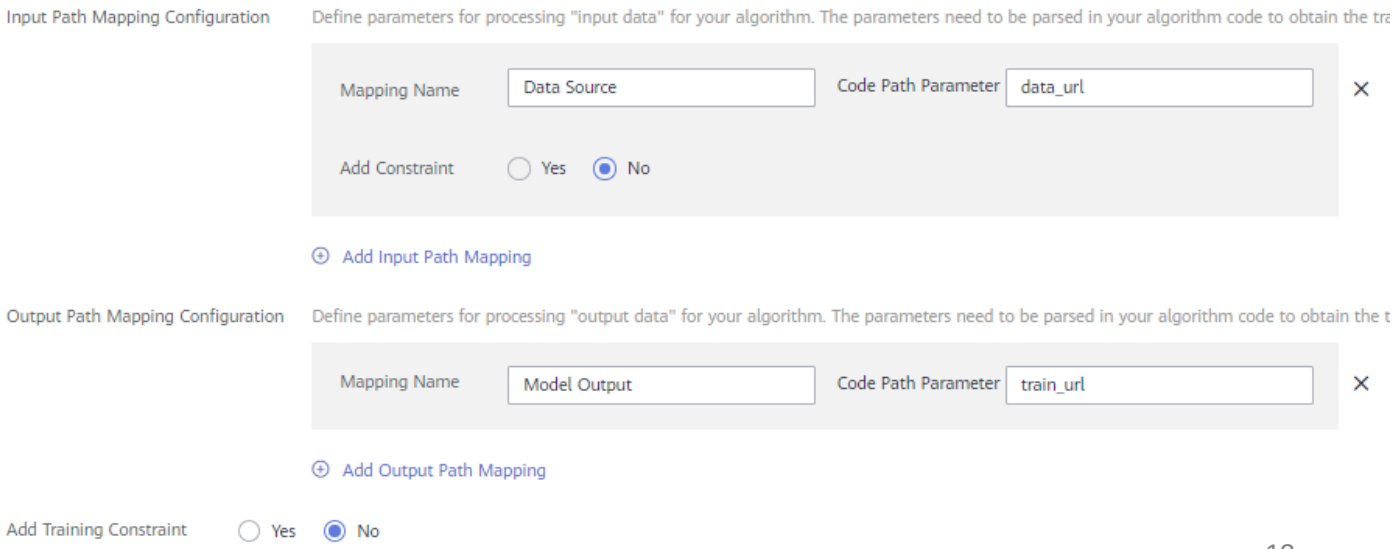
- dataset_url => to training images dataset
- label_url => to training images' label

04. In/Out Path Configuration

- Input path mapping
- Output path mapping (make sure the folder exists)



label_url => obs://mspdc-testing-bucket/dataset/label/
dataset_url => obs://mspdc-testing-bucket/dataset/



/03

Create training jobs on ModelArts

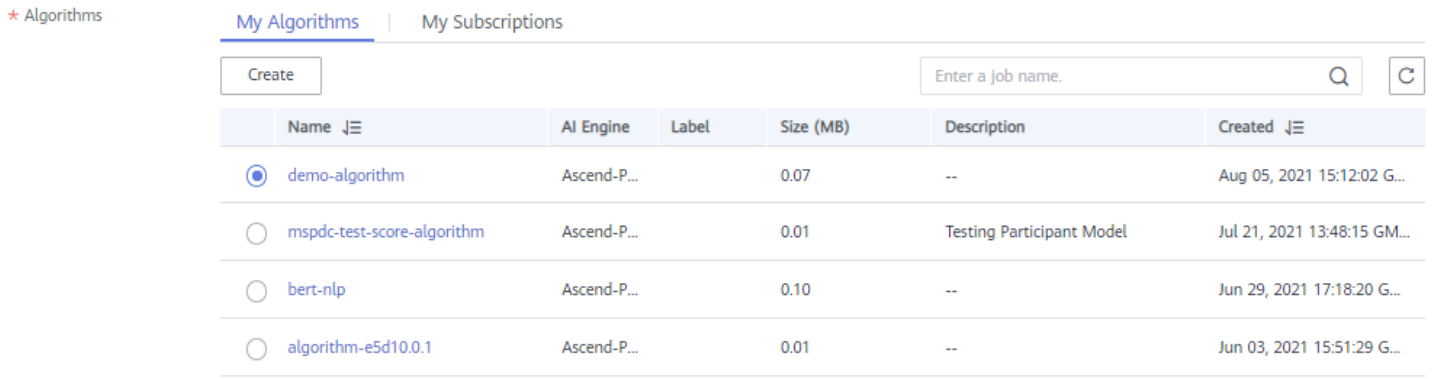
1. How to create a Training Job
2. Connect ModelArts & OBS - Moxing

How to Create a Training Job?

Algorithm Management > **My Algorithms > Create Training Job** (Operation Column)

01. Algorithms

- My Algorithms → choose the algorithm just created



02. Training Input/Output

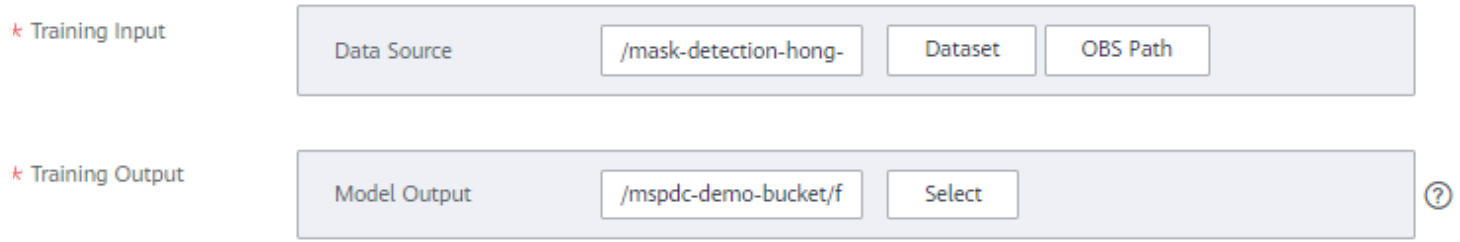
- Data Source = data_url (should be replaced by dataset_url)
- Model Output = train_url

03. Hyperparameter

- dataset_url => to training images dataset
- label_url => to training images' label

04. Resource Configuration

- Resource Pool & Type
- Instance & Compute Nodes
- Job Log Path



Data Source = data_url => any thing you like (will be replaced)
Model Output = train_url => obs://mspdc-demo-bucket/frcnn/out**/**

How to Create a Training Job?

Algorithm Management > **My Algorithms > Create Training Job** (Operation Column)

01. Algorithms

- My Algorithms → choose the algorithm just created

02. Training Input/Output

- Data Source = data_url (should be replaced by dataset_url)
- Model Output = train_url

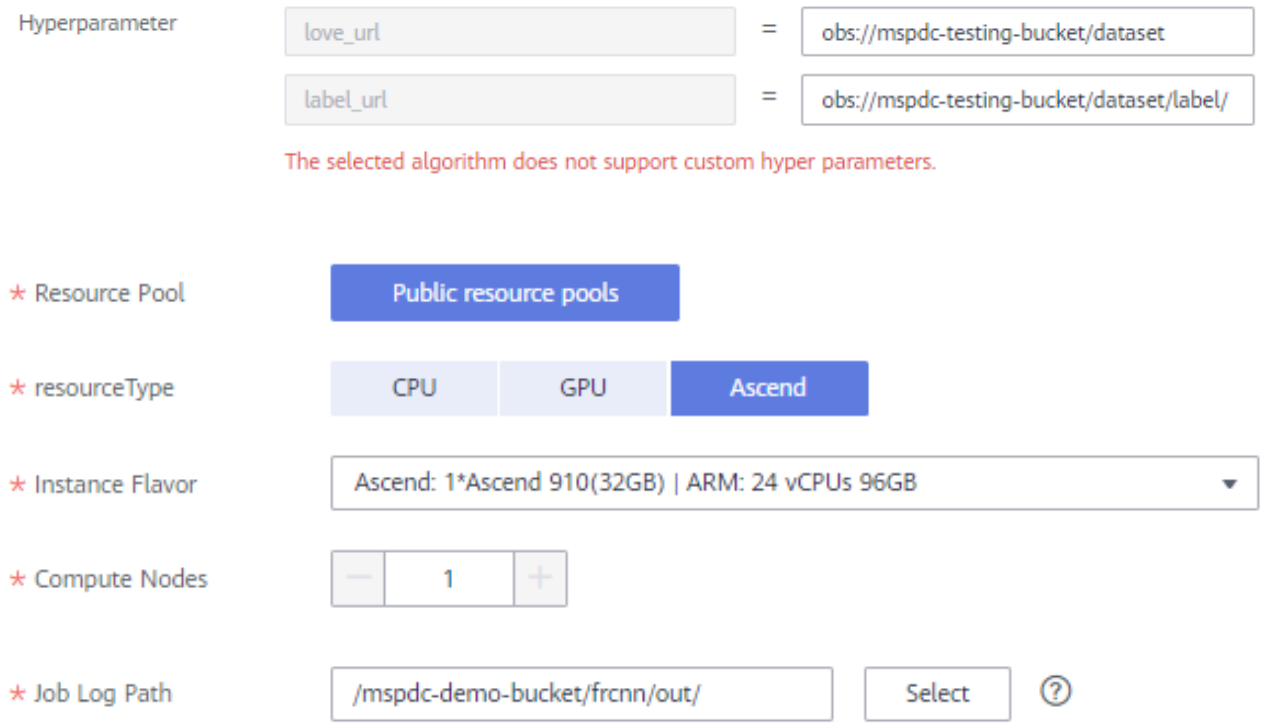
03. Hyperparameter

- dataset_url => to training images dataset
- label_url => to training images' label

04. Resource Configuration

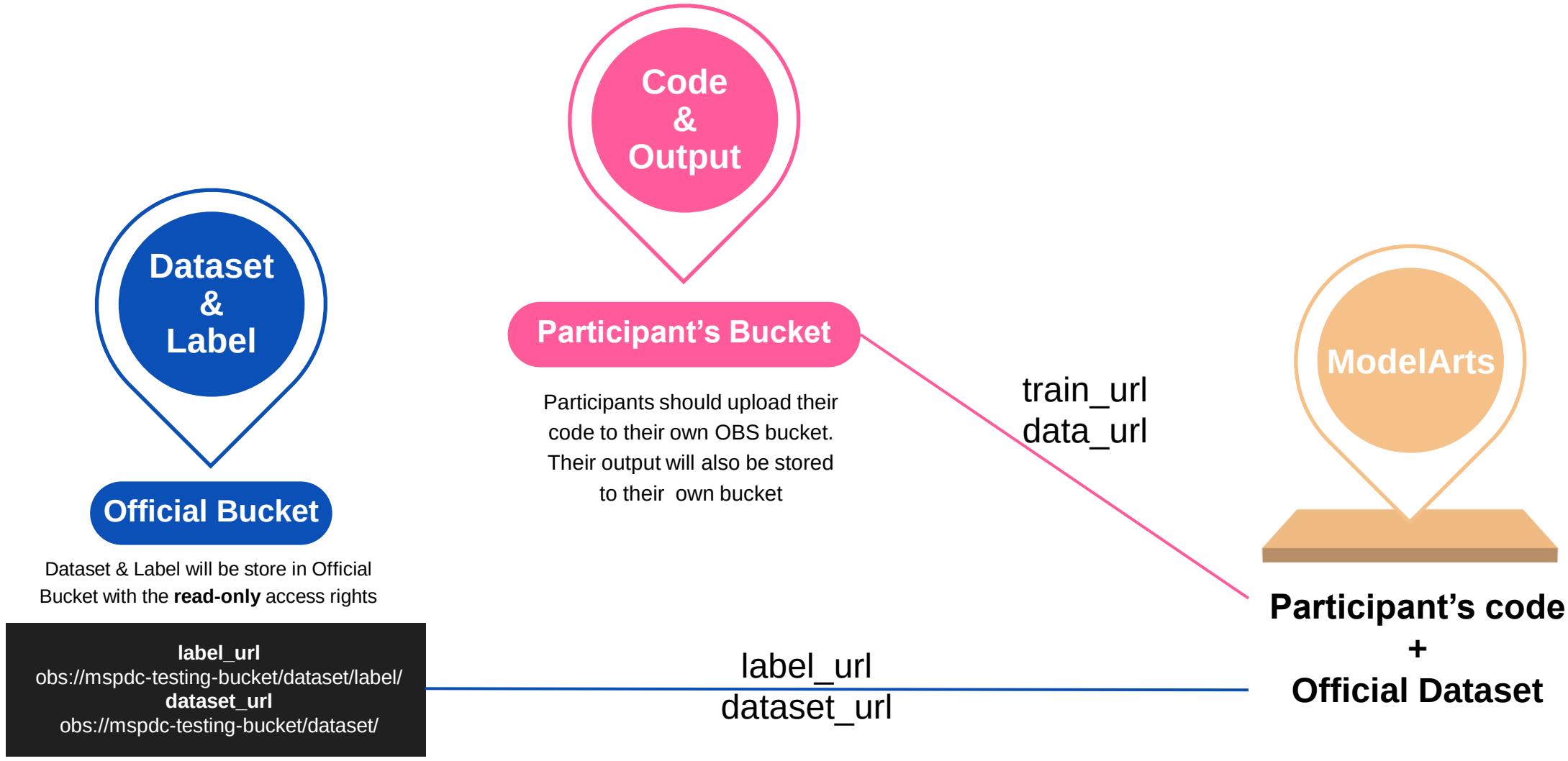
- Resource Pool & Type
- Instance & Compute Nodes
- Job Log Path

dataset_url & label_url => should be auto-complete based on the parameters from the Algorithm



Job Log Path => obs://mspdc-demo-bucket/frcnn/out/

MSC21 Training Structure



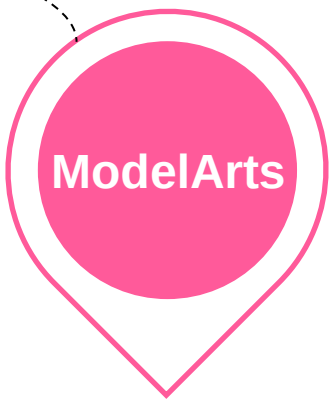
Connect ModelArts & OBS – MoXing Framwork

Source Code

In the uploaded source code, you will need MoXing to tell ModelArts how to interact with OBS

```python
import moxing as mox


if __name__ == '__main__':
  # ... parser argument
  args_opt = parser.parse_args()
  args_opt.data_url = args_opt.dataset_url  #replace to our dataset url

  data_path = './data/'
  label_path ='./label/'
  mox.file.copy_parallel(src_url=cfg.data_url, dst_url=data_path)
  mox.file.copy_parallel(src_url=cfg.label_url, dst_url=label_path)

  # ... create mindrecord from data_path
  # ... create dataset from mindrecord
  # ...  start training  ...
```

https://support.huaweicloud.com/intl/en-us/moxing-devg-modelarts/modelarts_11_0001.html



Algorithm

When training job starts, the algorithm will start execute the source code contain mox.file funciton

/04 Post-Training

1. Learn about the status and logs
2. Manage Training Jobs version

Training Jobs Status & Logs

Training Status

Algorithm Usage

Training Jobs New

Create Training Job

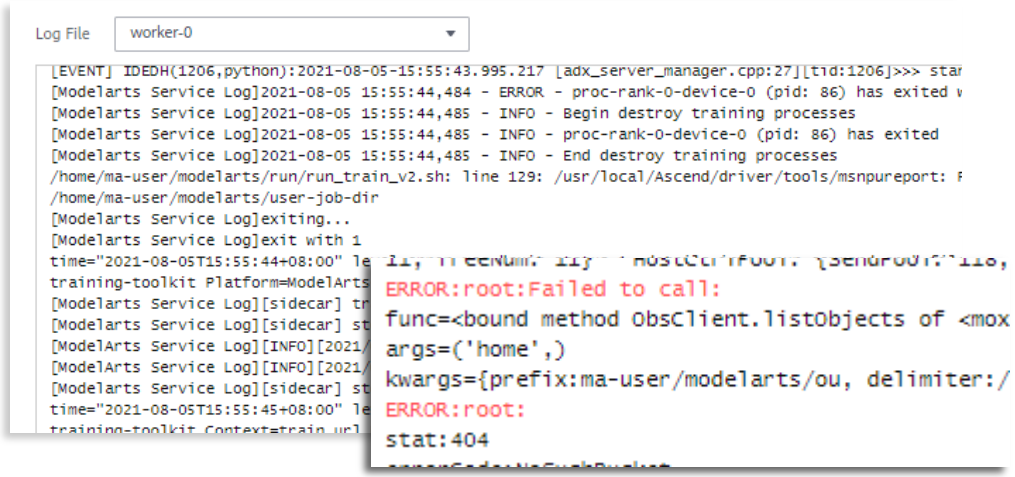
Display Only My Instances

Advanced Search

All statuses

Name	Job Type	Status	Duration (hh:mm:ss)	Created	Algorithm	Flavor	Nodes
demo-training-copy-f627	Training Job	Failed	00:11:16	Aug 05, 2021 15:43:52 ...	demo-algorithm	Ascend: 1*Ascend 910(32GB) | ARM: 24 vCPUs 96GB	1 | Regular
demo-training	Training Job	Terminated	00:12:24	Aug 05, 2021 15:19:50 ...	demo-algorithm	Ascend: 1*Ascend 910(32GB) | ARM: 24 vCPUs 96GB	1 | Regular
trainjob-crab-poc	Training Job	Completed	02:00:02	Jul 28, 2021 14:40:20 G...	物体检测-EfficientDet	GPU: 1*NVIDIA-V100(32GB) | CPU: 8 vCPUs 64GB	1 | Regular

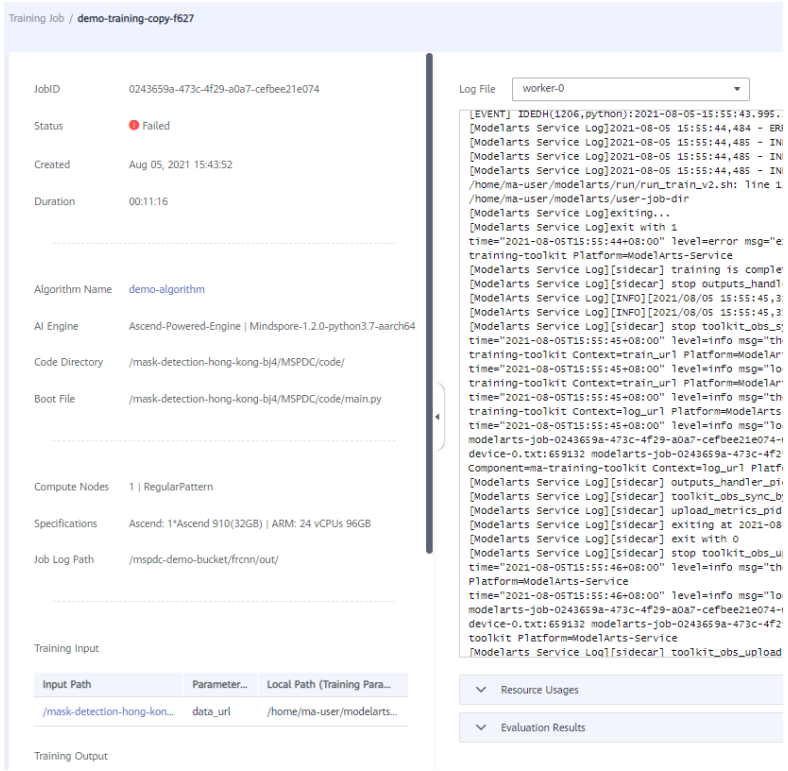
Click →
check Detail and Logs



Manage Training Jobs version

Viewing Training Job Versions

Training Management > Training Jobs



Comparing Versions of a Training Job

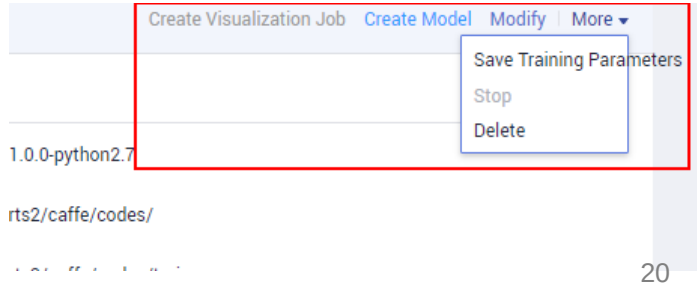
Version Manager > View Comparison Result

View Comparison Result

Vers...	Running Parameter	F1 Score	Recall	Precision	Accuracy
V0005	data_path_suffix=codes				
V0004	data_path_suffix=codes				
V0003	data_path_suffix=codes				
V0002	data_path_suffix=codes				
V0001	data_path_suffix=codes				

Shortcut Options

Training Management > Training Jobs



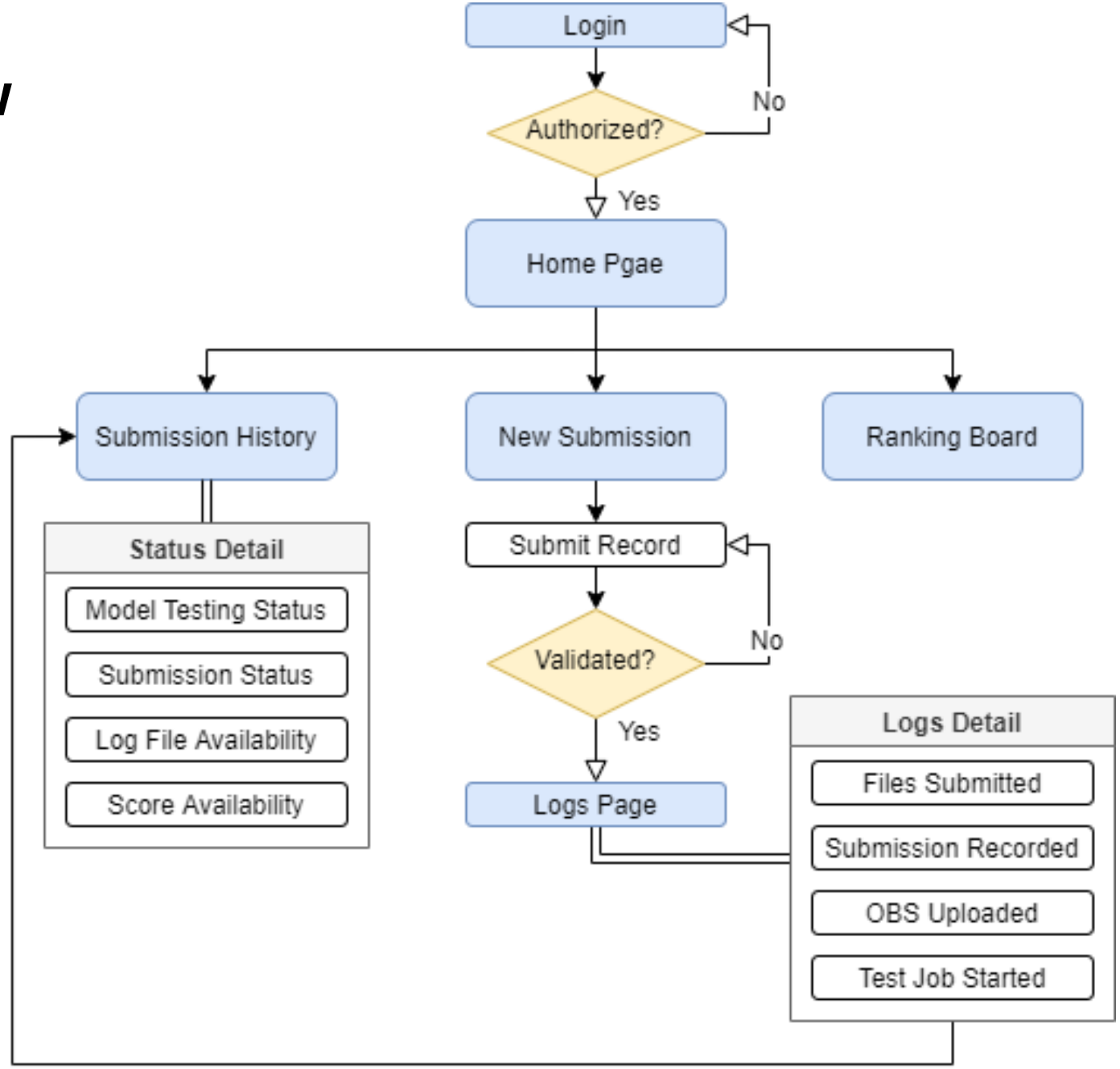
Model Evaluation Stage

Detail about the Testing and Evaluation procedure on your model!

Testing Flow Overview

Main Procedures

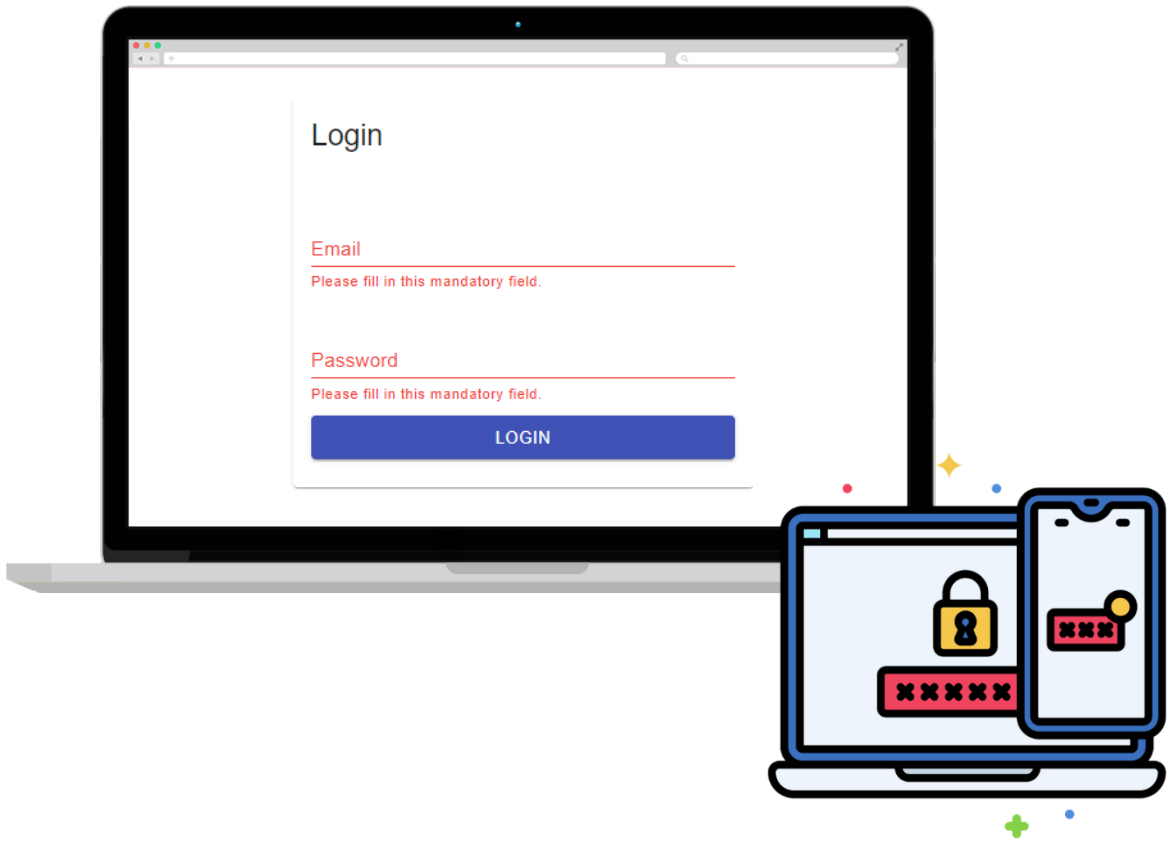
1. Login and Authentication
2. Create Testing Jobs on Huawei Cloud ModelArts
3. View Submission Records
4. Realtime Ranking Billboard
5. Logs and evaluation picture for optimizing model



/01 Login & Authentication

Secure your work and submission carefully

JWT Authentication



01. Login with JWT Token

- Login with password once
- Login with JWT Token for the rest of your journey

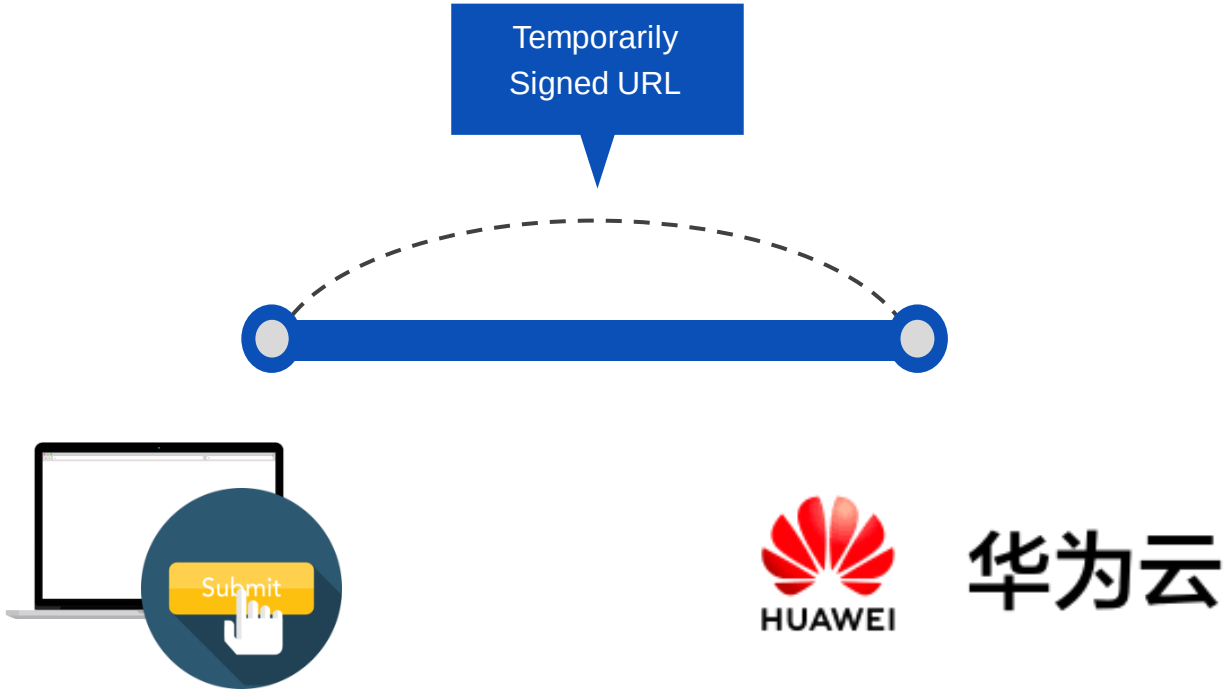
02. API Authentication

- Every Step will validate your ID and prevent any suspicious API

03.Change Password

- We will give you the password at first
- Change password to any kind you like

Signed URL Authentication



01. Secure Connection with Cloud

- Only connect when it’s authenticated
- Every user can only access to their submissions

02. Two-Factor Authentication

- Middleware handle identity of user requests
- Huawei Cloud authenticate the signature in the URL

/02 Create a New Submission

Create a new submission of code + .ckpt file for score-testing on models

Python Code Submission Structure

To automate the Testing Procedure and Protect the Privacy, please follow the code structure of the template to return the correct score

Name	Rules	Description
Submission File Type	.zip folder or .py file	Code for testing model
Entry Point	participant_model.py	Describe the model structure & optional functions
Model Class Name	class **Net**(mindspore.nn.Cell)	Class extends **mindspore.nn.Cell** which is the standard way to instantiate a network in **MindSpore**
***Pre-Process Function**	def **pre_process** (image_id, image)	**Preprocess** the dataset, so that the network can receive the correct **data format**
***Post-Process Function**	def **post_process**(image_id, prediction)	**Parse the output** from the model, with the **No. bounding boxes** as first dimension and **bounding boxes details** as the second dimension
***Explainable Function**	def **saliency_map**(net, image_id, image, predictions)	Output a saliency map from the model with the image. Utilize the mindspore.explainer toolkit.

*** = optional function***
Please refer to the Rule Book for detail guideline

Possible File Structure

```
.
├── ...
├── src
│   └── participant_model.py            # python code entry point
│       └── class Net(
│   └── dataset.py                      # data processing
│       ├── def pre_process:
│       └── def post_process:
│   └── utils.py                        # some other utils function
│       └── def saliency_map:
│
└── ...
```

Model Class Definition

```
import mindspore as ms

class Net(ms.nn.Cell):
  def __init__(self):
    super(Net, self).__init__()
  def construct(self, x):
```

Python Code Submission Structure

To automate the Testing Procedure and Protect the Privacy, please follow the code structure of the template to return the correct score

Explainable AI Template Code

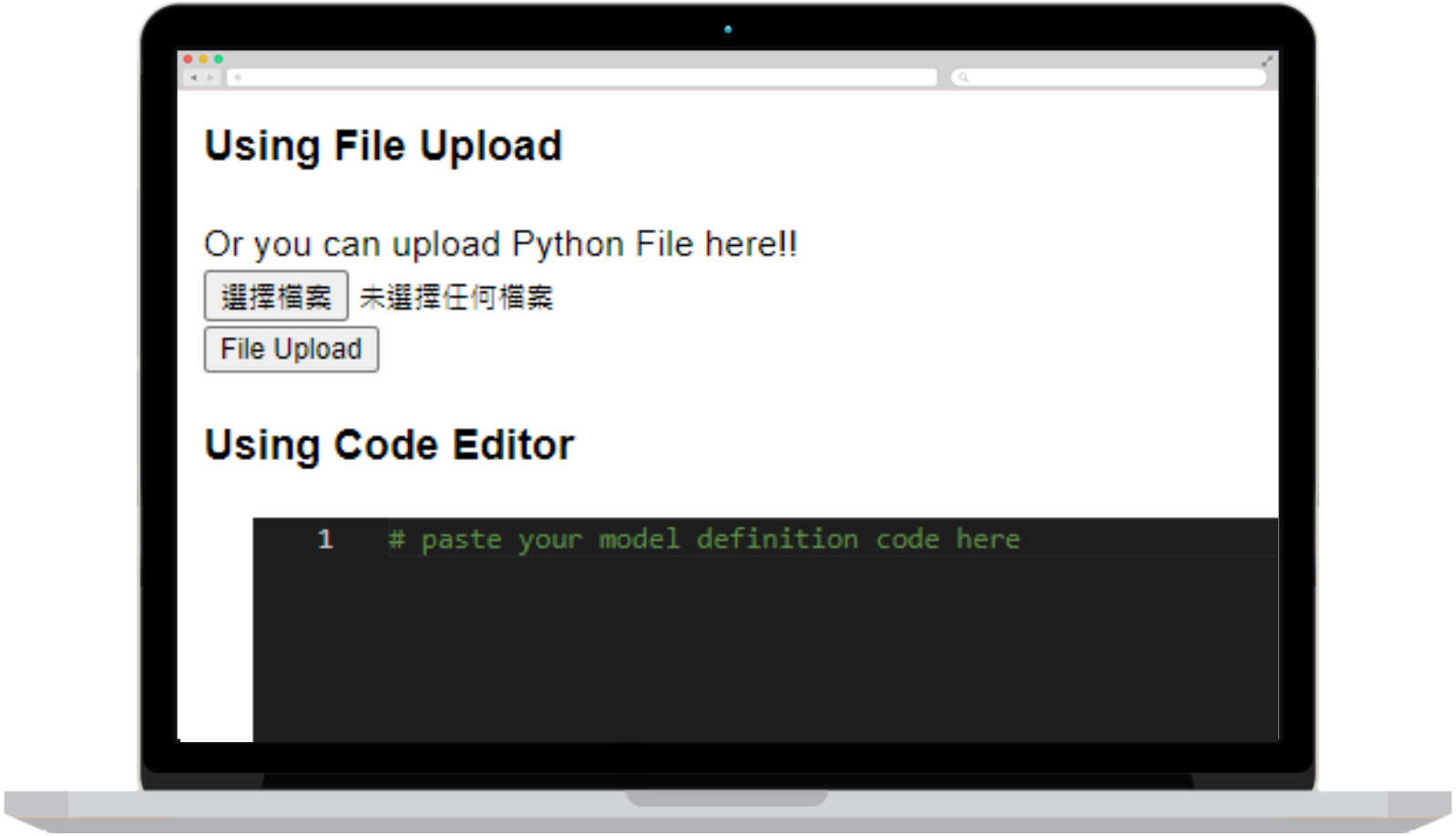
```
import numpy as np
import mindspore as ms


def saliency_map(
    net: Net,
    image_id: str,
    image: np.array,
    prediction: ms.Tensor
):
    """
    Keyword arguments:
    net -- ms.nn.Cell format, an instance of the Net class.
    image_id -- str format, for identifying an image.
    image -- np.array of CHW format.
    prediction -- the output of the network.

    Returns:
    result -- np.array, with the same shape of the image width and
height, but only one channel. Shape: (H, W)
                Should be the result from ms.explainer.*
    """
```

Data Processing Template Code

```
import numpy as np
import mindspore as ms

def pre_process(image_id: str, image: np.array):
    """
    Keyword arguments:
    image_id -- str format, for identifying an image.
    image -- np.array of CHW format.

    Returns:
    pre_processed_data -- dict or ms.Tensor, to be passed into
the input.
    """


def post_process(image_id: str, prediction: ms.Tensor):
    """
    Keyword arguments:
    image_id -- str format, for identifying an image.
    prediction -- the output of the network.

    Returns:
    results -- np.array or 2d-list,
        [xmin, ymin, xmax, ymax, p0, p1, p2, p3][]
        p0 to p3 implies probabilities for SCC, AC, SCLC, NSCLC
    """
```

Upload or Write Python Code

Multiple files in a zip file is also supported



S1 Python Code

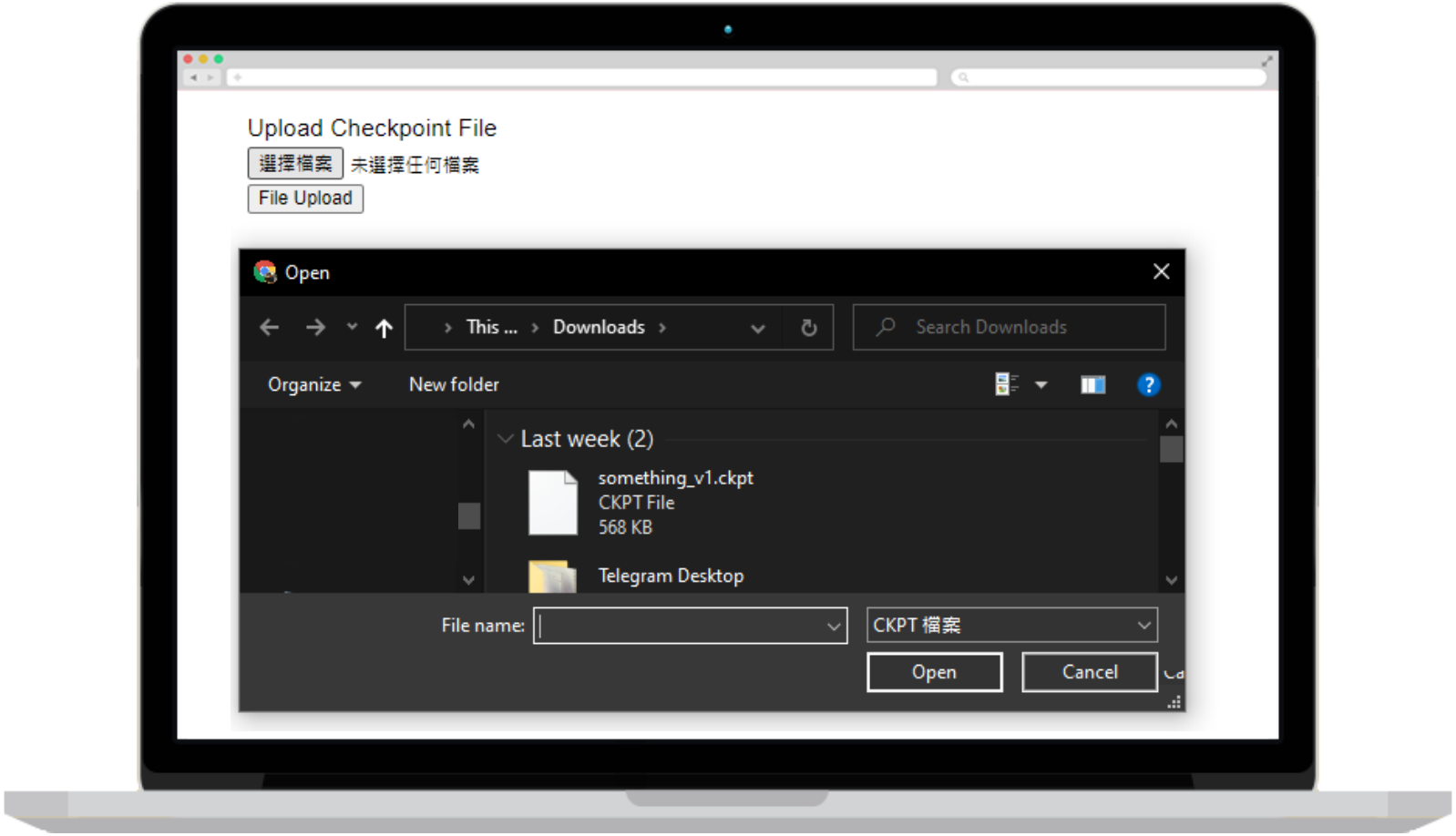
S2 Checkpoint File

S3 Double-Check Submission Detail

S4 Submission Logs

Upload Mindspore Checkpoint File (weights)

Only Mindspore is supported



S1 Python Code

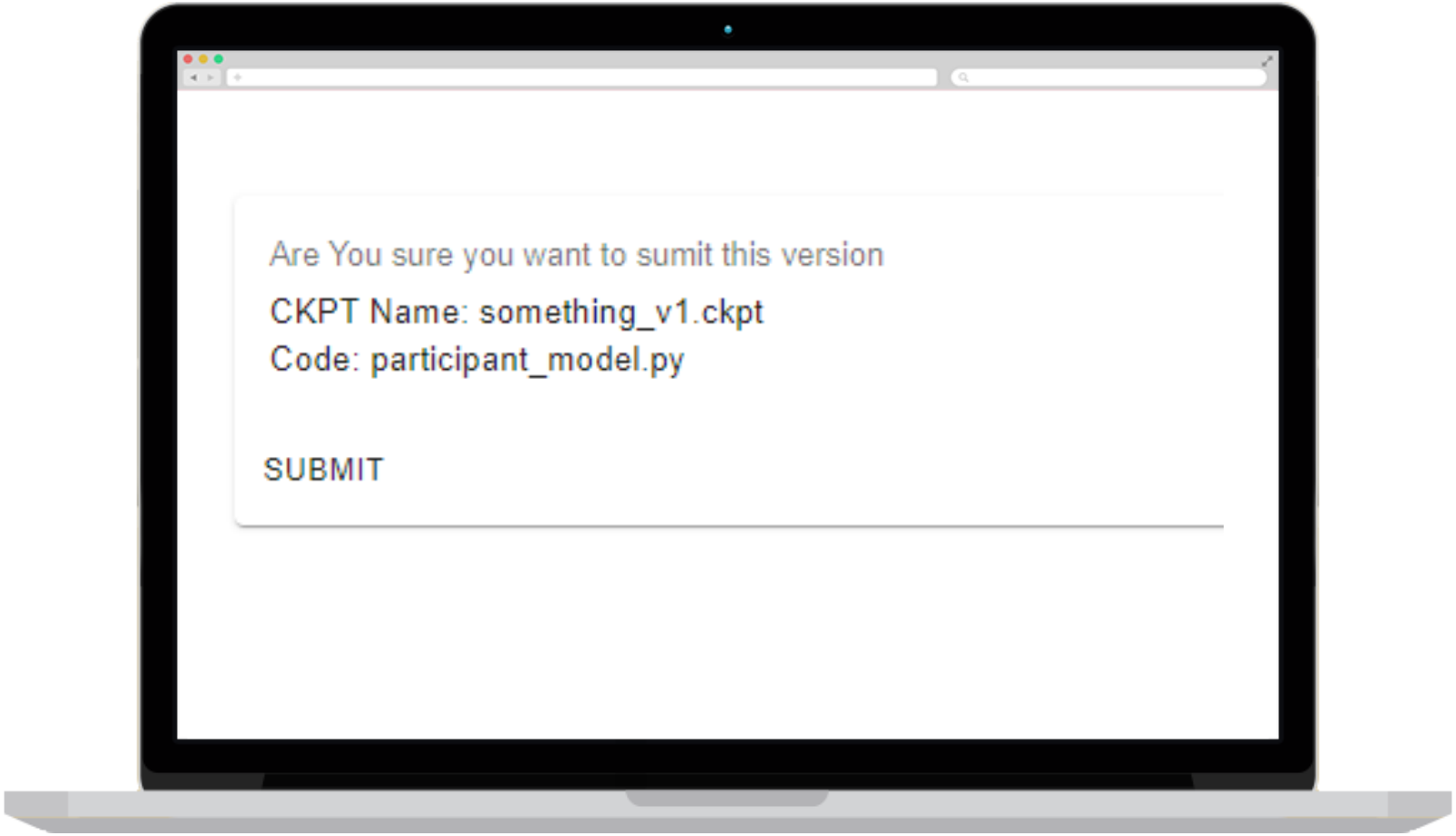
S2 Checkpoint File

S3 Double-Check Submission Detail

S4 Submission Logs

Double Check on the Submission Detail

This will show the file name you are going to submit



S1 Python Code

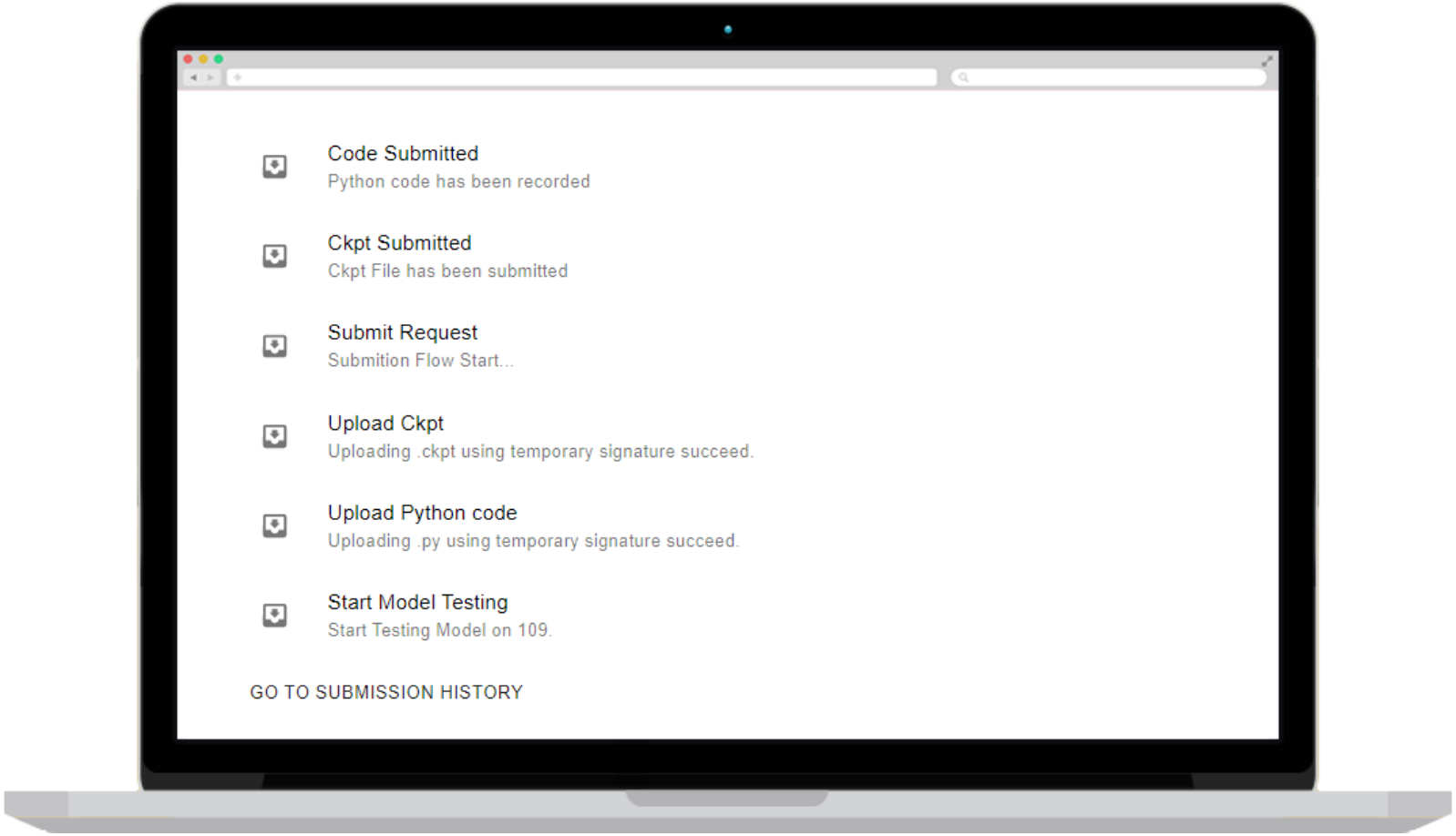
S2 Checkpoint File

S3 Double-Check Submission Detail

S4 Submission Logs

Inspect the Submission Status

Click on the “GO TO SUBMISSION” to check the Testing Status



S1 Python Code

S2 Checkpoint File

S3 Double-Check Submission Detail

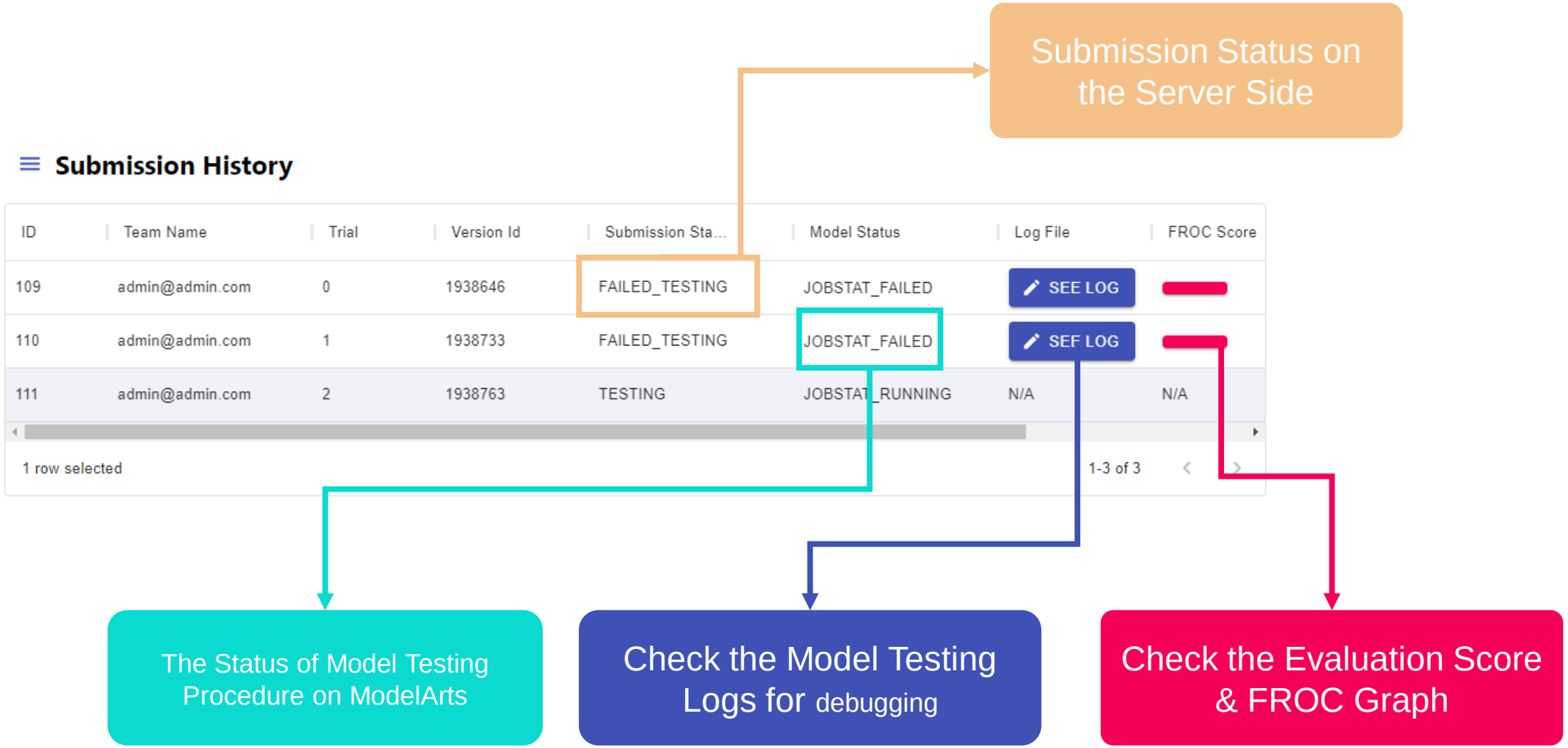
S4 Submission Logs

/03

Inspect on Submission History

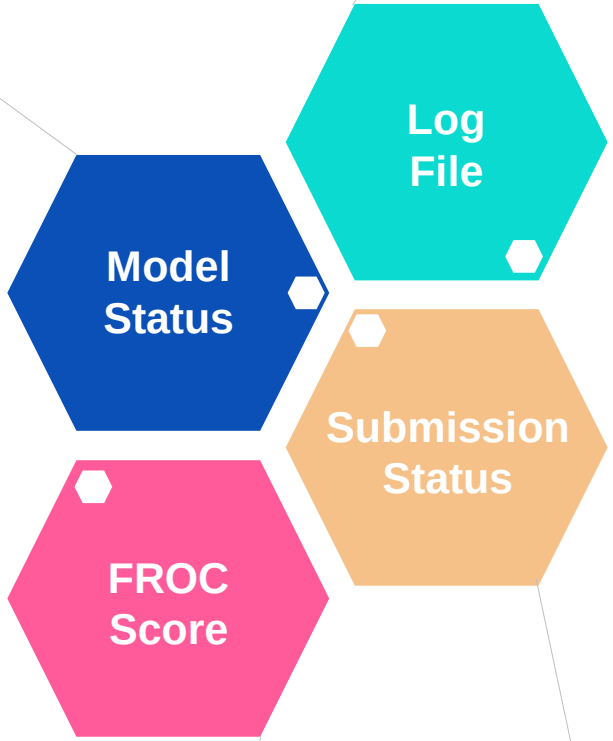
All the submissions will go to the Submission History Page for your record

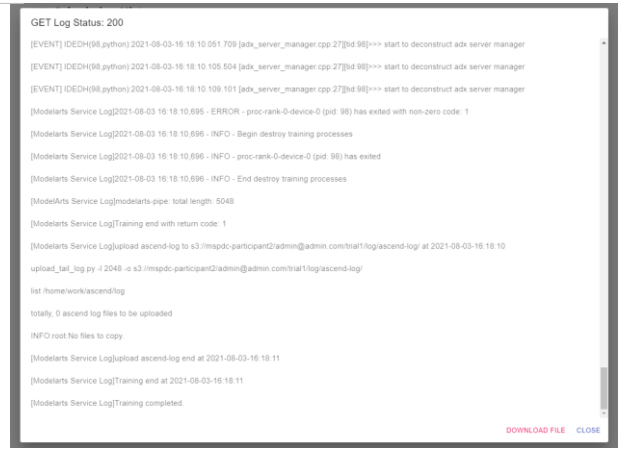
Overview of Submission History

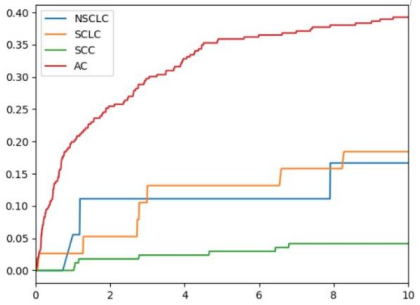


Status Elaboration

ID	Status Name	Meaning
0	JOBSTAT_UNKNOWN	?
1	JOBSTAT_INIT	初始化
2	JOBSTAT_IMAGE_CREATING	创建镜像
3	JOBSTAT_IMAGE_FAILED	创建镜像失败
4	JOBSTAT_SUBMIT_TRYING	提交作业
5	JOBSTAT_SUBMIT_FAILED	提交作业失败
6	JOBSTAT_DELETE_FAILED	删除作业失败
7	JOBSTAT_WAITING	等待
8	JOBSTAT_RUNNING	运行中
9	JOBSTAT_KILLING	杀死中
10	JOBSTAT_COMPLETED	完成
11	JOBSTAT_FAILED	失败
12	JOBSTAT_KILLED	杀死
13	JOBSTAT_CANCELED	取消
14	JOBSTAT_LOST	丢失
15	JOBSTAT_SCALING	自动伸缩
16	JOBSTAT_SUBMIT_MODEL_FAILED	提交模型失败
17	JOBSTAT_DEPLOY_SERVICE_FAILED	部署服务失败
18	JOBSTAT_CHECK_INIT	检查初始化
19	JOBSTAT_CHECK_RUNNING	检查运行中
20	JOBSTAT_CHECK_RUNNING_COMPLETED	检查运行中完成
21	JOBSTAT_CHECK_FAILED	检查失败





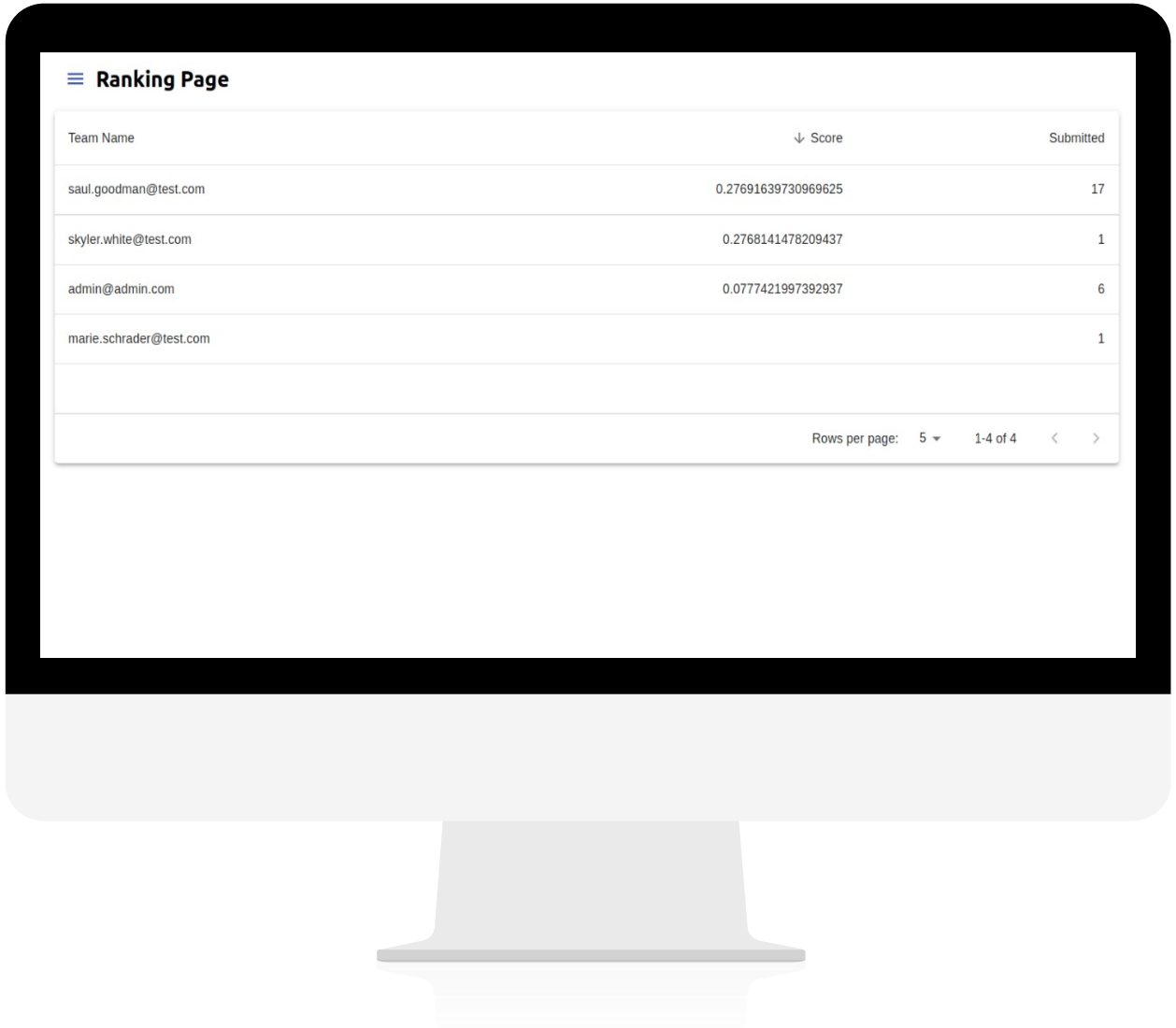


Status Name
UPLOADING
TESTING
SCORING
LOGGING
COMPLETED
FAILED_UPLOAD
FAILED_TESTING
FAILED_SCORING
FAILED_LOGGING
FAILED

/04 Check Your Ranking Board!

The highest score’s submission from each team will be display on the Ranking Board

Ranking Board Overview



Ranking Page Detail

Team Name	FROC Score	Submission Count
…	…	…

How to Score High?

1. Submit Earlier!
2. Higher FROC!

(Check out Rule book for more scoring detail)

Thanks

Join Discord & Checkout GitHub Repos