

# Jonathan Flightbase™

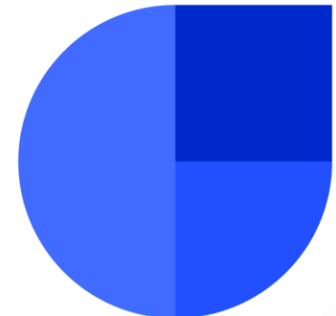
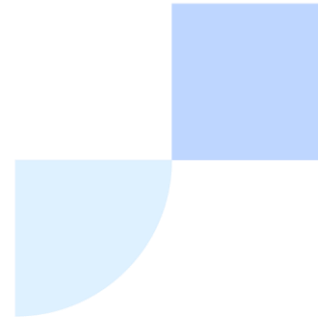
## 사용자 매뉴얼

acryl


STRICTLY CONFIDENTIAL

Copyright 2020 ACRYL. All rights reserved.

Created by : Edgar



# Contents

- 1 Jonathan Flightbase™ 소개
  - 2 Jonathan Flightbase™ 주요 기능
  - 3 Docker Image 생성 및 Dataset 업로드
  - 4 다양한 타입의 학습 생성
  - 5 학습 모델 배포 및 테스트
- 

# 1

## Jonathan Flightbase™ 소개

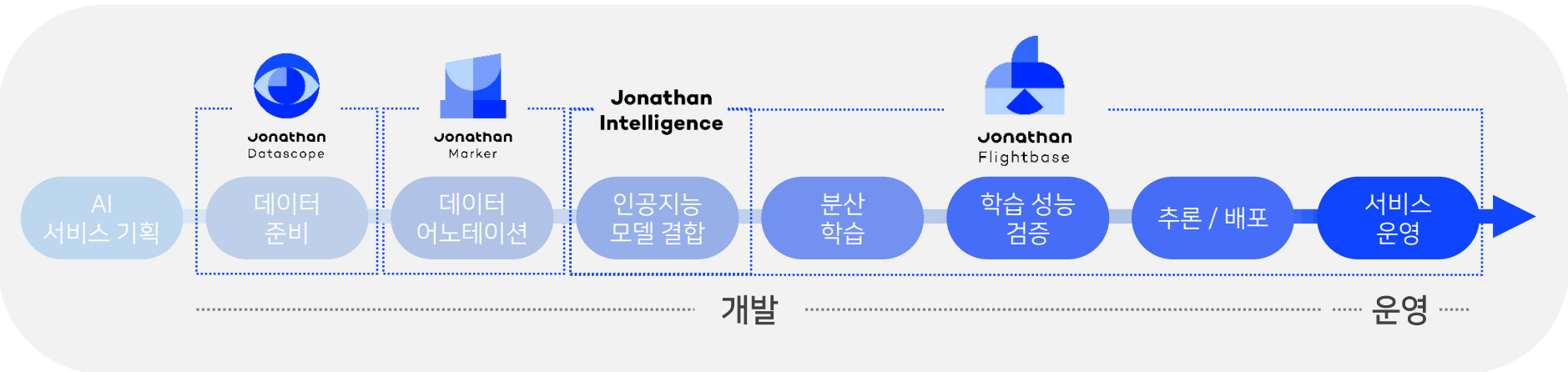
- 개요
- 구조
- 컨셉



하나의 플랫폼에서 학습을 위한 도구들을 탑재하고,  
사용자가 편리하게 모델 설계에 집중할 수 있도록 지원

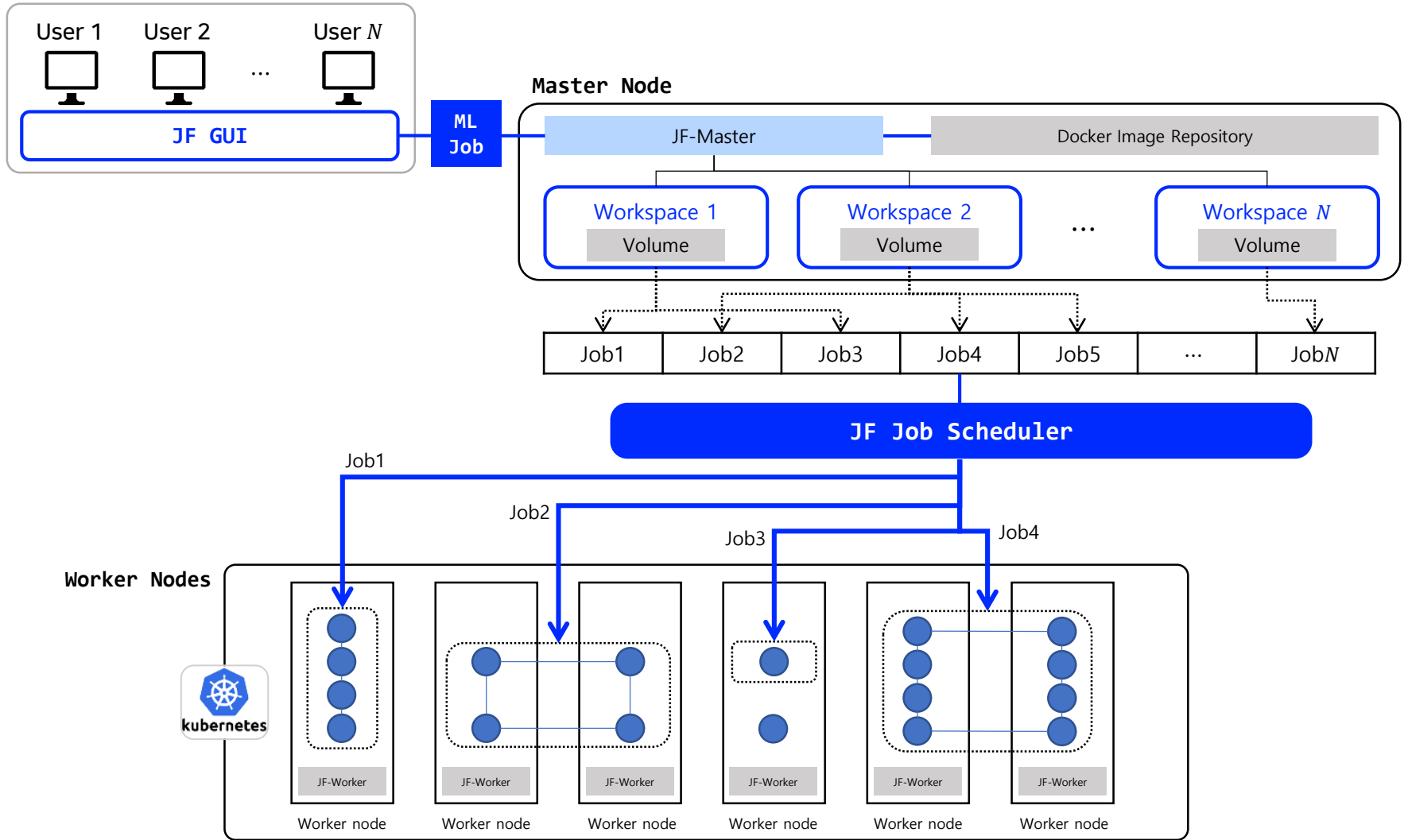


**Jonathan**  
[AI 구축 프로세스]



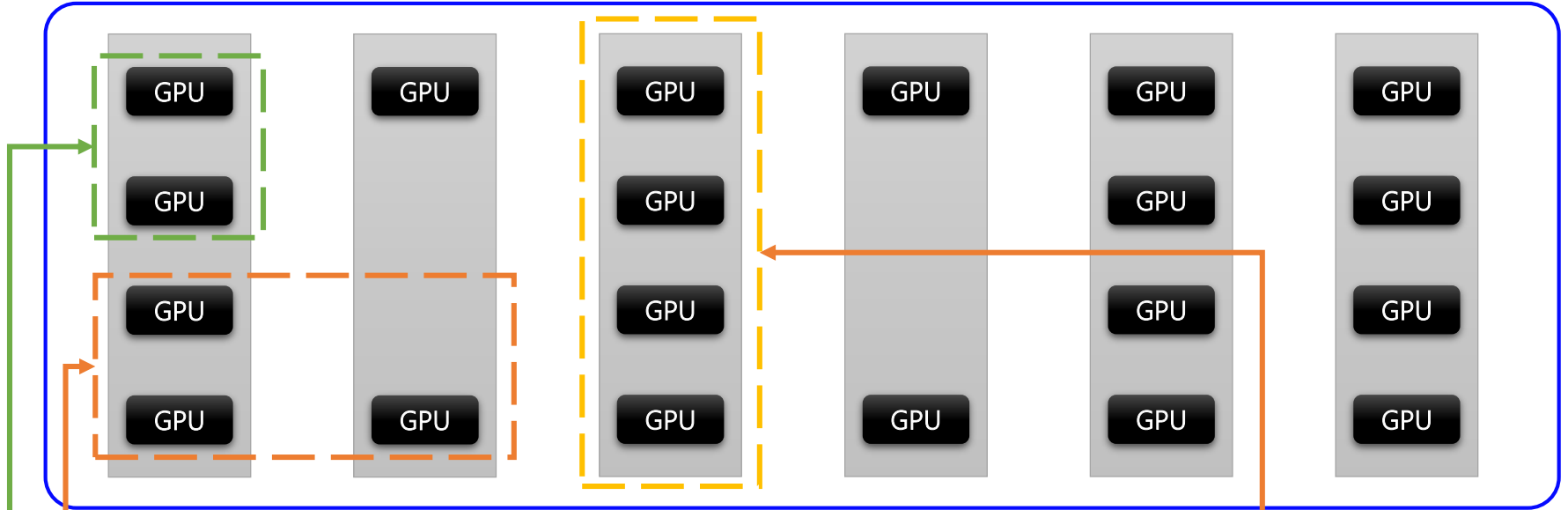
모델 학습 완료 후,  
플랫폼 상에서 이를 테스트하고 서비스 할 수 있도록 지원

# JFB 소개 - 구조



# JFB 소개 - 컨셉 : 자원 사용

## Worker Nodes



20개 GPU

① Workspace 생성 시에는 GPU를 논리적으로 수량만 나눈 상태

5개 GPU

10개 GPU

1개 GPU

4개 GPU

Workspace-A

Workspace-B

Workspace-C

Workspace-D

Job-A1  
2개 GPU

Job-A2  
3개 GPU

② 학습 실행 시, Workspace에 할당된 GPU 수 만큼만 입력 가능

↓  
전체 GPU Pool에서 JF Scheduler를 통해 GPU 배정

\* JF Scheduler - 가능하면 동일 서버 내 GPU 먼저 클러스터링

2

# Jonathan Flightbase™ 주요 기능



관리자 작업 단계

1) 서버에 JF 설치



2) Nodes 추가



3) 사용자 계정 생성



4) 워크스페이스 생성



5) 플랫폼 사용 기록 관리



## 사용자 작업 단계

- 1) **Docker Image 생성/업로드**
  - JF GUI에서 관리자로부터 부여받은 계정으로 로그인
  - 워크스페이스 목록에서 사용하고자 하는 워크스페이스 선택
  - 메뉴에서 "Docker Image" 선택
  - Pull / Dockerfile Build / Tar 의 3가지 방식을 이용하여, 플랫폼 내 Docker Image 생성 가능
  
- 2) **Datasets 업로드**
  - 메뉴에서 "Datasets" 선택
  - 폴더 혹은 파일을 선택하여 업로드
  - Dataset 상세 페이지에서 파일 추가 가능
  
- 3) **Training/Jobs 생성**
  - 메뉴에서 "Trainings" 선택
  - 다양한 타입의 학습 생성 가능 (Basic / Advanced / Built-in Model / Jupyter Notebook)
  - Editor를 이용하여 학습 코드 생성 및 업로드
  - Basic / Advanced / Built-in Model 학습의 경우, Job 생성하여 학습
  
- 4) **학습 로그 확인**
  - Jobs 상세 페이지에서 학습 별 로그 확인 (그래프 확인 및 로그 파일 다운로드)
  
- 5) **학습 모델 배포**
  - 메뉴에서 "Deployment" 선택
  - "Create Deployment"를 선택하고, 학습 완료 모델의 체크포인트를 지정하여 배포 생성
  - 배포를 "On"하여 API 활성화

## JFB 주요 기능 - 사용자 기능

The screenshot displays the Jonathan Flightbase dashboard for user 'edgar'. The interface includes a sidebar with a search bar and a list of workspaces: 'ws-for-berkeley', 'test-ws', 'test-keit', 'storage-test-ws', and 'res-ws'. The main area shows a grid of workspace cards, each with a status indicator (e.g., '사용중' for 'In Use') and a timeline. Each card displays two donut charts for '학습 GPU 수' (Learning GPU count) and '배포 GPU 수' (Deployment GPU count), along with a table of '학습' (Learning) and '배포' (Deployment) counts, and '도커 이미지' (Docker images) and '데이터셋' (Datasets) counts.

Workspace	Status	Timeline	Learning GPU	Deployment GPU	Learning Count	Deployment Count	Docker Images	Datasets		
ws-for-berkeley	사용중	2020-10-22 00:00 ~ 2020-11-30 23:59	0%	0%	0/11	0/0	2	0	10	0
storage-test-ws	사용중	2020-10-07 00:00 ~ 2020-10-31 23:59	0%	0%	0/0	0/0	0	0	10	1
res-ws	대기중	2020-12-01 00:00 ~ 2020-12-19 23:59	0%	0%	0/0	0/0	5	0	10	1
test-ws	사용중	2020-09-24 00:00 ~ 2020-12-31 23:59	100%	50%	2/2	1/2	24	11	13	14
test-keit	사용중	2020-09-01 00:00 ~ 2020-11-30 23:59	0%	100%	0/0	3/3	3	5	13	0

- 로그인 후 현재 사용 가능한 워크스페이스의 목록 및 각 워크스페이스의 현황 확인 가능

# JFB 주요 기능 - 사용자 기능

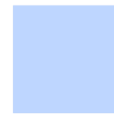
The screenshot shows the JFB workspace management interface. The left sidebar contains a navigation menu with 'Workspace 정보' (Workspace Info) and 'Workspace 메뉴 목록' (Workspace Menu List) callouts. The main content area includes:

- Workspace 정보**: A callout pointing to the workspace name 'test-ws' and its details.
- Workspace 메뉴 목록**: A callout pointing to the menu items: 워크스페이스 홈, 도커 이미지, 데이터셋, 학습, 배포, 서비스.
- 최근 Job 진행 상태 확인**: A callout pointing to the '최근 작업 현황' (Recent Job Status) table.
- GPU 최대 사용률 Chart**: A callout pointing to the 'GPU 사용 그래프' (GPU Usage Graph) chart.
- GPU 사용 현황**: A callout pointing to the '사용 중인 GPU 수' (Number of GPUs in Use) section, which shows 100% for both '학습용' (Learning) and '배포용' (Deployment) with a 2/2 count.
- 항목별 수 확인**: A callout pointing to the '항목별 수' (Item Counts) section, which shows counts for '도커 이미지' (2), '데이터셋' (11), '학습' (25), and '배포' (10).
- 최근 작업 내역 확인**: A callout pointing to the '최근 실행 기록' (Recent Execution History) log.

- 워크스페이스 홈 화면에서 현재 워크스페이스의 현황 확인 가능

# 3

## Docker Image 생성 및 Dataset 업로드



# 도커 이미지 업로드 - 도커 이미지 목록

**도커 이미지 생성/업로드**

**생성된 도커 이미지 목록**

**필터 및 검색**

상태	업로드 방식	공개 범위	도커 이미지 이름	크기	업로더	업로드 날짜	수정
준비됨	Built-in	전체	object_detection2		root	2020-10-20 19:52:16	
준비됨	Built-in	전체	object_detection		root	2020-10-20 19:52:16	
준비됨	Built-in	전체	image_empathy	11.80GB	root	2020-10-20 19:52:16	
준비됨	Pull	워크스페이스	tata	0.06GB	daehan	2020-10-08 17:26:34	
준비됨	Pull	전체	bqaweq	6.05GB	root	2020-08-31 19:11:48	
준비됨	Dockerfile Build	워크스페이스	docker-mecab	9.95GB	lyla	2020-08-13 15:13:09	
준비됨	Dockerfile Build	전체	horovod-cpu	4.08GB	lyla	2020-08-13 15:04:13	
준비됨	Pull	전체	test	6.05GB	edgar	2020-08-12 17:05:54	
준비됨	Pull	전체	ngc-docker-img	6.05GB	edgar	2020-08-07 18:23:17	
준비됨	Pull	전체	docker-test	6.05GB	edgar	2020-08-04 11:53:12	

**도커 이미지 수정**

- 해당 워크스페이스에 도커 이미지 (Docker Image)를 업로드할 수 있습니다.

## 도커 이미지 업로드 - 도커 이미지 목록 (상세 정보)

The screenshot shows the Docker Hub interface for a user named Jonethan Flightbase. The main content area displays a list of Docker images under the 'test-ws' namespace. The table below summarizes the visible entries:

선택	상태	업로드 방식	공개 범위	도커 이미지 이름	크기	업로더	업로드 날짜	수정
<input type="checkbox"/>	준비됨	Built-in	전체	object_detection2		root	2020-10-20 19:52:16	
<input type="checkbox"/>	준비됨	Built-in	전체	object_detection		root	2020-10-20 19:52:16	
<input type="checkbox"/>	준비됨	Built-in	전체	image_empathy	11.80GB	root	2020-10-20 19:52:16	
<input type="checkbox"/>	준비됨	Pull	워크스페이스	tata	0.06GB	daehan	2020-10-08 17:26:34	
<input type="checkbox"/>	준비됨	Pull	전체	bqaweq	6.05GB	root	2020-08-31 19:11:48	
<input type="checkbox"/>	준비됨	Dockerfile Build	워크스페이스	docker-mecab	9.95GB	lyla	2020-08-13 15:13:09	

Below the table, the detailed information for the 'docker-mecab' image is shown:

### docker-mecab의 상세정보

CUDA 버전: 10.0      TensorFlow 버전: 2.0.0      MPI 버전: 4.0.0      Pytorch 버전: -

docker-mecab on the System

Tag: kR8i9Q	이미지 ID: 01097e14c58a	저장소: 192.168.1.13:5000/jfb/bybuild-docker-mecab
-------------	----------------------	---

- 도커 이미지의 상세 정보를 통해, 해당 이미지에 설치되어 있는 프레임워크 및 라이브러리 버전 정보를 확인할 수 있습니다.

The image shows a web form titled "도커 이미지 업로드" (Docker Image Upload). On the left, five black callout boxes with white text point to specific fields in the form:

- 생성할 도커 이미지의 이름 입력** (Enter the name of the Docker image to be created) points to the "도커 이미지 이름" (Docker image name) field.
- 세부 설명 입력** (Enter detailed description) points to the "도커 이미지 정보 - 선택 항목" (Docker image information - optional) text area.
- 업로드 방식 선택** (Select upload method) points to the "도커 이미지 업로드 방식" (Docker image upload method) radio buttons.
- 방식에 맞는 파일 업로드** (Upload file matching the method) points to the "도커 이미지 URL을 입력하세요" (Enter Docker image URL) field.
- 공개 범위 선택** (Select public scope) points to the "공개 범위" (Public scope) radio buttons.

The form fields and options are:

- 도커 이미지 이름**: Text input with placeholder "도커 이미지 이름을 20자 이내로 입력하세요."
- 도커 이미지 정보 - 선택 항목**: Text area with placeholder "도커 이미지 정보를 입력하세요." and a character count "0/200".
- 도커 이미지 업로드 방식**: Radio buttons for  Pull,  Dockerfile build, and  Tar.
- 도커 이미지 URL을 입력하세요**: Text input.
- 공개 범위**: Radio buttons for  Workspace and  Global.

At the bottom right, there are two buttons: "취소" (Cancel) and "업로드" (Upload).

- "도커 이미지 업로드" 선택 시, [Pull / Dockerfile Build / Tar] 형태로 도커 이미지를 업로드할 수 있습니다.
- 접근 범위에서 "Workspace" 선택 시 해당 워크스페이스에서만 공개되며, Global 선택 시 플랫폼 사용자 전체가 사용 가능합니다.

# 데이터셋 생성 - 데이터셋 목록

The screenshot displays the 'test-ws' workspace in Jonathan Flightbase. The '데이터셋' (Dataset) tab is active, showing a list of datasets. A '데이터셋 생성' (Create Dataset) button is visible. The dataset list includes columns for selection, name, files, creator, update time, creation time, edit, and owner. A '데이터셋 수정' (Edit Dataset) callout points to the edit icon in the last row. A '필터 및 검색' (Filter and Search) callout points to the search bar. A '데이터셋 이름' (Dataset Name) callout points to the dataset name column. A '데이터셋 생성' (Create Dataset) callout points to the '데이터셋 생성' button. An '업로드한 데이터셋 목록' (Uploaded Dataset List) callout points to the dataset list. A '데이터셋 업로드' (Dataset Upload) callout points to the '데이터셋 생성' button.

<input type="checkbox"/>	접근 권한	데이터셋 이름	파일	생성자	수정 날짜	생성 날짜	수정	어노테이션
<input type="checkbox"/>	읽기&쓰기	test	1 files	lyla	2020-10-26 17:59:29	2020-10-26 17:59:29		Jonathan Marker >
<input type="checkbox"/>	읽기&쓰기	ier-image	8 files	edgar	2020-10-20 20:20:46	2020-10-20 19:07:24		Jonathan Marker >
<input type="checkbox"/>	읽기전용	cycle_gan	4,943 files	root	2020-09-24 18:20:43	2020-09-24 18:20:43		Jonathan Marker >
<input type="checkbox"/>	읽기전용	pixel_to_pixel	8,182 files	root	2020-09-24 18:20:43	2020-09-24 18:20:43		Jonathan Marker >
<input type="checkbox"/>	읽기전용	simple_forecasting	1 files	root	2020-09-24 18:20:43	2020-09-24 18:20:43		Jonathan Marker >
<input type="checkbox"/>	읽기전용	simple_text_classification	2 files	root	2020-09-24 18:20:43	2020-09-24 18:20:43		Jonathan Marker >
<input type="checkbox"/>	읽기전용	qa_classification	2 files	root	2020-09-24 18:20:43	2020-09-24 18:20:43		Jonathan Marker >
<input type="checkbox"/>	읽기전용	text_generation	1 files	root	2020-09-24 18:20:43	2020-09-24 18:20:43		Jonathan Marker >
<input type="checkbox"/>	읽기전용	neural_machine_translation	2 files	root	2020-09-24 18:20:43	2020-09-24 18:20:43		Jonathan Marker >
<input type="checkbox"/>	읽기전용	precedent_classification	2 files	root	2020-09-24 18:20:43	2020-09-24 18:20:43		Jonathan Marker >

- 해당 워크스페이스에 학습을 위한 데이터셋을 업로드할 수 있습니다.



## 데이터셋 생성 - 데이터셋 세부 목록

The screenshot displays the 'test-ws' dataset page. The left sidebar shows a list of datasets, with 'test-ws' selected. The main content area shows the 'ier-image' dataset details, including a '데이터 업로드' button and a '폴더 생성' button. Below this is a table of files and folders within the dataset.

유형	이름	크기	수정자	수정 날짜	수정
파일	JF_data_face.csv	0.00MB	root	2020-10-20 20:21:35	✎
폴더	face	0.32MB	root	2020-10-20 20:19:15	✎
폴더	Image	3.51MB	yeobie	2020-10-20 19:07:31	✎
파일	JF_data.csv	0.00MB	yeobie	2020-10-20 19:07:23	✎

Callouts in the image point to various UI elements:

- 데이터셋 정보**: Points to the dataset name 'test-ws' and the 'ier-image의 상세정보' link.
- 데이터셋 내 폴더 생성**: Points to the '폴더 생성' button.
- 데이터셋 내의 파일 목록**: Points to the table listing files and folders.
- 데이터셋 내 파일 추가**: Points to the '데이터 업로드' button.
- Dataset 내 File 삭제**: Points to the '삭제' button.
- 폴더/파일 이름 수정**: Points to the edit icons in the table.

- 업로드 된 데이터셋 내의 세부 파일 목록을 확인할 수 있습니다.
- 또한, 폴더를 생성하거나 이름 변경이 가능합니다.

**데이터셋 생성**

데이터셋 이름  
데이터셋 이름을 20자 이내로 입력하세요.

데이터셋 정보 - 선택 항목 0/200  
데이터셋 정보를 입력하세요.

접근 권한  
 읽기&쓰기  읽기전용

업로드 방식  
 파일  폴더

찾기

업로드할 파일 선택  
(다중 선택 가능)

취소 생성

- “데이터셋 생성” 버튼을 선택하고, 해당 정보를 입력하여 데이터셋을 생성합니다.
- 다수의 파일을 선택하거나 특정 폴더 전체를 업로드 할 수 있습니다.
- 또한, 접근 권한을 선택하여 파일의 수정을 제어할 수 있습니다.

# 4

## 다양한 타입의 학습 생성



# 다양한 타입의 학습 생성 - 학습 목록

The screenshot displays the Flightbase interface for a workspace named 'test-aws'. The main area shows a grid of learning tasks, each with a title, creation time, and details. Callouts point to various UI elements: '학습 생성' (Learning Generation) points to the '학습 생성' button; '학습 타입 필터' (Learning Type Filter) points to the filter tabs (Built-in, Basic, Advanced, Jupyter Notebook); '학습 목록' (Learning List) points to the list of tasks; '정렬 및 검색' (Sort and Search) points to the sort and search options; '학습 상세 정보' (Learning Detailed Information) points to the detailed view of the 'ji-test' task; 'SSH 접속 포트 정보' (SSH Connection Port Information) points to the SSH port number; '최근 실행 작업 상태' (Recent Execution Job Status) points to the progress bar and status of the 'jf-built-in-test' task; '코드 작성/수정을 위한 Jupyter Notebook 실행 (CPU)' (Jupyter Notebook Execution for Code Writing/Editing (CPU)) points to the CPU toggle and Jupyter icon; '학습을 위한 Jupyter Notebook 실행 (GPU)' (Jupyter Notebook Execution for Learning (GPU)) points to the GPU toggle and Jupyter icon.

- “학습” 메뉴에서 현재 생성된 학습 목록들을 확인하고 현재 작업 상태를 확인할 수 있습니다.
- 현재 Flightbase를 통해 4가지 타입의 학습을 생성할 수 있습니다.

## 다양한 타입의 학습 생성 - 학습 타입별 목록 정보

### Advanced

Advanced ⋮

**benchmark-test-22**

🕒 2020-10-14 17:33:00 | lyla

---

도커 이미지  
**jf-default**

인스턴스 / 구성  
**GPU \* 2개 / -**

네트워크  
-

---

**Jupyter Notebook**

CPU 편집용  jupyter >

코드 편집을 위해 주피터 노트북을 활성화해주세요.

SSH - 30803

---

**최근 작업** 🕒 2020-10-14 17:33:23

대기 0 종료 1 중지 (1/1)

[작업 목록으로 이동 >](#)

### Built-in Model

Built-in ⋮

**jf-built-in-test**

🕒 2020-10-20 19:41:24 | robert

---

도커 이미지  
**jf-default**

인스턴스 / 구성  
**GPU \* 1개 / -**

네트워크  
-

---

**Simple Text Classification의 정보**

입력 유형 -

LSTM 을 사용한 문장 분류 엔진

---

**최근 작업** 🕒 2020-10-20 19:45:37

대기 0 종료 1 중지 (1/1)

[작업 목록으로 이동 >](#)

### Jupyter

Jupyter ⋮

**gpu5**

🕒 2020-10-14 18:29:43 | lyla

---

도커 이미지  
**jf-default**

인스턴스 / 구성  
**GPU \* 1개 / GeForce GTX 1080 Ti**

네트워크  
-

---

**Jupyter Notebook**

CPU 편집용  jupyter >

주피터 노트북을 실행하여 코드를 편집하세요.

SSH - 30587

---

GPU 학습용  jupyter >

주피터 노트북을 실행하여 코드를 편집하세요.

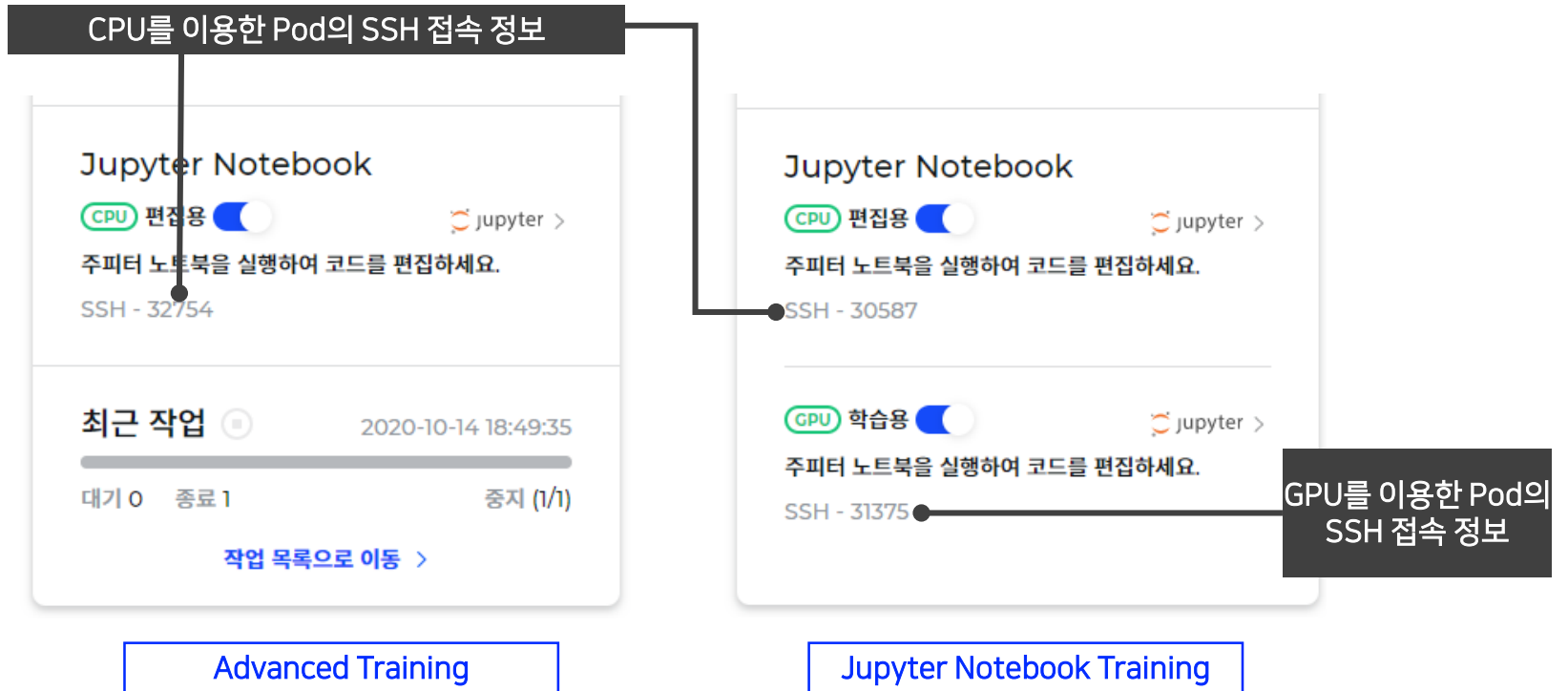
SSH - 31375

- 사용자 학습 코드를 이용한 학습 실행
- 멀티 GPU를 이용한 분산 학습을 지원하는 Training Type
- Flightbase내 탑재된 기본 이미지를 사용

- Flightbase 내 탑재된 학습 모델을 이용하여 학습 실행
- 멀티 GPU를 이용한 분산 학습을 지원하는 Training Type
- Flightbase내 탑재된 기본 이미지를 사용

- Jupyter Notebook을 이용하여 코드를 작성하고 바로 학습 가능
- 단일 GPU를 지원하는 Training Type
- 사용자 이미지를 이용하여 학습 가능

## 다양한 타입의 학습 생성 - SSH 접속 방법



```
Shell
[COMMAND]
ssh [사용자 ID]@[Mater Server IP] -p [SSH Port]
```

- “Advanced”와 “Jupyter” 타입의 학습은 CPU 및 GPU Pod (Container)에 SSH로 접속하여 사용 가능합니다.

## 다양한 타입의 학습 생성 - 학습 생성

The image shows a screenshot of a workspace management interface. A modal window titled "학습 생성" (Create Learning) is open in the center. The modal contains the following sections:

- 기본 정보** (Basic Information):
  - 학습 이름** (Learning Name): A text input field with a placeholder "학습 이름을 20자 이내로 입력하세요."
  - 학습 정보 - 선택 항목** (Learning Information - Selected Items): A text input field with a placeholder "학습 정보를 입력하세요." and a character count "0/200".
- 학습 유형** (Learning Type): Three radio button options: "Basic", "Advanced" (selected), and "Jupyter Notebook".
- GPU 사용량** (GPU Usage): A dropdown menu showing "총 2 개 할당 가능 / 현재 0 개 사용 가능".
- 접근 설정** (Access Settings): A section with "취소" (Cancel) and "생성" (Create) buttons.

The background interface shows a list of workspaces, including "ji-test" and "gpu4".

- "학습 생성"을 통해 학습을 생성합니다. 학습 유형으로 4가지 (Basic, Advanced, Jupyter Notebook, Built-in Model)를 제공합니다.
- GPU 사용량을 입력하여 학습 시 다수의 GPU를 사용할 수 있습니다.

## 다양한 타입의 학습 생성 - 학습 생성

The screenshot displays the '학습 생성' (Create Learning) modal in the Jonathon Flightbase interface. The modal is overlaid on a workspace named 'test-ws'. The modal contains the following sections:

- 학습 정보 입력:** A text input field for entering learning information.
- 학습 유형 (Learning Type):** Radio buttons for 'Basic', 'Advanced' (selected), and 'Jupyter Notebook'. There are also radio buttons for 'Jonathan Intelligence' and 'Built-in Model'.
- GPU 사용량 (GPU Usage):** A dropdown menu showing '총 2개 할당 가능 / 현재 0개 사용 가능'.
- 접근 설정 (Access Settings):** A section for '접근 권한' (Access Permission) with radio buttons for 'Public' (selected) and 'Private'.
- 소유자 (Owner):** A dropdown menu showing 'edgar'.

Buttons for '취소' (Cancel) and '생성' (Create) are located at the bottom right of the modal.

- 접근 권한을 Public으로 설정 시, 워크스페이스의 모든 사용자가 접근이 가능하며, Private으로 설정 시, 선택된 사용자만 접근이 가능합니다.



## 다양한 타입의 학습 생성 - 학습 목록

The screenshot displays the 'test-ws' workspace in the Jonathan Flightbase interface. The main content area shows a list of training jobs under the heading 'jf-built-in-test의 작업'. A callout box labeled 'GPU 옵션 선택 여부' points to the '사용 중인 GPU 수 1' field in the job details. Another callout box labeled '작업 실행 정보' points to the '2020-10-20 19:45:37' timestamp. A third callout box labeled '작업 생성' points to the '작업 생성' button. A fourth callout box labeled '작업 실행 파라미터 확인' points to the 'epoch 20 batch\_size 64 learning\_rate 0.001 data /user\_dataset/' parameters. A fifth callout box labeled '선택 작업 삭제' points to the '삭제' button. A sixth callout box labeled '작업 그룹 상태' points to the '종료' button. The interface also shows a search bar, a sidebar with workspace navigation, and a footer with copyright information.

- “Advanced” 학습 및 “Built-in Model” 학습은 학습 작업 (Job)을 생성하여 자동 실행할 수 있습니다.
- 작업은 다수의 파라미터를 한번에 입력하여 그룹 단위로 실행 가능합니다.

## 다양한 타입의 학습 생성 - 학습 작업 생성 (Advanced)

The screenshot displays the '작업 생성' (Job Creation) modal in the Jonathan Flightbase interface. The modal is centered over a background showing a workspace named 'test-ws' with a 'benchmark-test-2' job. The modal contains the following fields and options:

- 작업 이름** (Job Name): A text input field with the placeholder '작업 이름을 입력하세요.' (Enter job name).
- 도커 이미지** (Docker Image): A dropdown menu currently showing 'jf-default'.
- 데이터셋** (Dataset): A dropdown menu currently showing 'simple\_image\_classification'.
- 실행 코드** (Execution Code): A dropdown menu with the placeholder '실행 코드를 선택하세요.' (Select execution code).
- 추가 기능 - 선택 항목** (Additional Features - Selection Items):
  - GPU 가속 통신** (GPU Accelerated Communication): GPU 가속 통신 기술을 활성화하여 학습 속도를 향상시킬 수 있습니다.
  - 통합 메모리** (Unified Memory): 통합 메모리 기능 사용시 학습 속도가 저하될 수 있으며, 이는 모델 및 서버 사양에 따라 달라질 수 있습니다.
  - 인피니밴드를 통한 RDMA (원격 직접 메모리 접근) GPU 가속 통신** (RDMA GPU Accelerated Communication via InfiniBand): RDMA GPU 가속 통신 기술은 인피니밴드로 연결되어 있는 서버에서만 사용할 수 있습니다.

Buttons for '취소' (Cancel) and '생성' (Create) are located at the bottom right of the modal.

- “Advanced” 유형의 학습에서 작업 (Job) 생성 시 데이터셋과 에디터 (CPU-Jupyter Notebook)을 통해 업로드한 Python 코드를 선택하여 학습할 수 있습니다.

## 다양한 타입의 학습 생성 - 학습 작업 생성 (Advanced)

**작업 생성**

추가 기능 - 선택 항목

- GPU 가속 통신  
GPU 가속 통신 기술을 활성화하여 학습 속도를 향상시킬 수 있습니다.
- 통합 메모리  
통합 메모리 기능 사용시 학습 속도가 저하될 수 있으며, 이는 모델 및 서버 사양에 따라 달라질 수 있습니다.
- 인피니밴드를 통한 RDMA (원격 직접 메모리 접근) GPU 가속 통신  
RDMA GPU 가속 통신 기술은 인피니밴드로 연결되어 있는 서버에서만 사용할 수 있습니다.

파라미터

- batch\_size=2, --num\_epoch=50
- batch\_size=4, --num\_epoch=50
- batch\_size=8, --num\_epoch=50
- batch\_size=16, --num\_epoch=50
- batch\_size=32, --num\_epoch=50
- batch\_size=64, --num\_epoch=50

추가

취소 생성

- 2개 이상의 GPU를 사용하는 학습의 경우, GPU 간 가속 통신 기능을 체크만으로 활성화하여 사용할 수 있습니다.
- 작업의 경우, 다수의 파라미터를 한번에 입력 받아 플랫폼에서 자동으로 순차 실행합니다.

## 다양한 타입의 학습 생성 - 학습 작업 생성 (Built-in Model)

**작업 생성 - Burn Detection**

작업 이름  
작업 이름을 입력하세요.

도커 이미지  
jf-default

데이터셋  
image    데이터셋을 선택하세요.

실행 코드  
built\_in\_run\_code

추가 기능 - 선택 항목

- GPU 가속 통신  
GPU 가속 통신 기술을 활성화하여 학습 속도를 향상시킬 수 있습니다.
- 통합 메모리  
통합 메모리 기능 사용시 학습 속도가 저하될 수 있으며, 이는 모델 및 서버 사양에 따라 달라질 수 있습니다.
- 인피니밴드를 통한 RDMA (원격 직접 메모리 접근) CPU 가속 통신  
RDMA GPU 가속 통신 기술은 인피니밴드로 연결되어 있는 서버에서만 사용할 수 있습니다.

취소    생성

작업 이름	시간	사용자
hine_translation	2020-09-25 11:52:36	edgar
hine_translation	2020-09-25 11:47:13	edgar
hine_translation	2020-09-25 11:45:51	edgar
hine_translation	2020-09-25 11:35:57	edgar
hine_translation	2020-09-25 09:52:28	daehan

- “Built-in Model” 유형의 학습에서 작업 (Job) 생성 시에는 정해진 유형의 데이터셋을 선택하여 학습할 수 있습니다.
- 실행 코드는 Built-in 으로 자동 설정됩니다.

## 다양한 타입의 학습 생성 - 학습 작업 생성 (Built-in Model)

The screenshot shows a web interface for job management. A modal window titled "작업 생성 - Burn Detection" is open, allowing users to configure job parameters. The modal includes a "추가 기능 - 선택 항목" section with three checked options: "GPU 가속 통신", "통합 메모리", and "인피니밴드를 통한 RDMA (원격 직접 메모리 접근) GPU 가속 통신". Below this is a "파라미터" section with a table for configuring epochs, batch sizes, and learning rates. The background shows a list of jobs with columns for job name, creation time, and user.

### 작업 생성 - Burn Detection

추가 기능 - 선택 항목

- GPU 가속 통신  
GPU 가속 통신 기술을 활성화하여 학습 속도를 향상시킬 수 있습니다.
- 통합 메모리  
통합 메모리 기능 사용시 학습 속도가 저하될 수 있으며, 이는 모델 및 서버 사양에 따라 달라질 수 있습니다.
- 인피니밴드를 통한 RDMA (원격 직접 메모리 접근) GPU 가속 통신  
RDMA GPU 가속 통신 기술은 인피니밴드로 연결되어 있는 서버에서만 사용할 수 있습니다.

파라미터

epoch	batch size	learning rate	
2	32	0.001	—
4	32	0.001	—
8	32	0.001	—
16	32	0.001	—
32	32	0.001	—

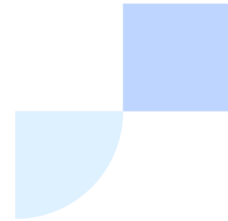
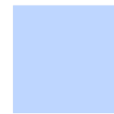
추가

취소 생성

- 2개 이상의 GPU를 사용하는 학습의 경우, GPU 간 가속 통신 기능을 체크만으로 활성화하여 사용할 수 있습니다.
- 작업의 경우, 정해진 파라미터들 여러 개를 한번에 입력 받아 플랫폼에서 자동으로 순차 실행합니다.

# 5

## 학습 모델 배포 및 테스트



# 학습 모델 배포 및 테스트 - 배포 목록

The screenshot displays the 'test-ws' workspace in the Jonathon Flightbase interface. The main content area shows a list of four deployed models:

- ji-ier-image-trained** (Jonathon Intelligence): Created 2020-10-20 20:33:39 by yeoble. Model: Image Empathy(image). Input: image. Docker image: image\_empathy.
- ji-ier-image** (Jonathon Intelligence): Created 2020-10-20 20:22:53 by yeoble. Model: Image Empathy(image). Input: image. Docker image: image\_empathy.
- simple-classificati** (Built-in): Created 2020-10-20 20:11:40 by yeoble. Model: Simple Text Classification. Input: text. Docker image: jf-default.
- adf** (Built-in): Created 2020-10-19 11:22:17 by daehan. Model: Burn Detection. Input: image. Docker image: jf-default.

Each model card includes a '모델 배포-서비스' (Model Deploy-Service) toggle and detailed deployment information such as status (배포중), node name (jf-node-04, jf-node-08), GPU configuration (GeForce GTX 1080 Ti, GeForce RTX 2080), and API details.

- 학습 완료된 체크포인트를 이용하여 모델을 배포할 수 있습니다.
- 현재는 플랫폼 내 API 가이드라인으로 인해 Built-in Model의 배포 기능만 지원하고 있습니다.

## 학습 모델 배포 및 테스트 - 배포 생성

### 배포 생성

#### 기본 정보

배포 이름

배포 정보 - 선택 항목 0/200

배포 유형

Jonathon Intelligence\*  Built-in Model  Custom Model

1. 타입과 인텔리전스 카테고리 선택하세요.

Custom Built-in Model 카테고리

2. 배포할 모델을 선택하세요.

- jf-built-in-test** CPU GPU Simple Text Classification
- burn-det-cpu CPU GPU Burn Detection
- info-tst2
- info-test CPU GPU Precedent Classification
- deptest CPU GPU Simple Text Classification

3. 체크포인트를 선택하세요.

test1 0 check/13-0.71.hdf5

취소 **생성**

인스턴스 유형

CPU  GPU

도커 이미지

접근 설정

접근 권한

Public  Private

소유자

취소 **생성**

- 배포 유형에서 사용자가 직접 학습한 "Custom" 혹은 미리 학습되어 있는 "Default"를 선택하여 배포할 수 있습니다.
- 또한, 1개의 CPU 혹은 다수의 GPU를 이용하여 배포할 수 있으며, 접근 권한 설정은 학습의 접근 권한 설정과 동일합니다.



# 학습 모델 배포 및 테스트 - 서비스 목록

The screenshot shows the 'test-ws' workspace in the Jonathon Flightbase interface. The sidebar on the left lists various workspaces, with 'test-ws' selected. The main area displays a list of services under the 'test-ws' workspace. The services are categorized into '서비스 예시' (Service Examples) and '사용자가 배포한 서비스' (Services Deployed by User). The services listed are:

- precedent-classificatio...**: Created on 2020-09-15 12:57:50. Description: 예규판례 분류 서비스 세팅과 관련된 문장/질의를 입력했을 때, 문장에 포함되어 있는 세팅에 대해 출력해주는 서비스. Model name: -. Data input type: text.
- behavior-recognition-s...**: Created on 2020-09-24 13:54:03. Description: 행동 인식 서비스 사람의 전신 이미지를 업로드 했을 때, 해당 이미지에 서 나타난 사람의 행동을 인식하는 서비스. Model name: -. Data input type: image.
- burn-degree**: Created on 2020-10-20 20:09:14. Description: 화상 심도 분류 서비스 환부 이미지를 업로드하고 서비스를 실행 했을 때, 화상의 정도를 알려주는 서비스... Model name: Burn degree classification. Data input type: canvas-coordinate, canvas-image.
- simple-classificati**: Built-in service. Created on 2020-10-20 20:11:46. Description: qwe. Model name: Simple Text Classification. Data input type: text.
- ji-ier-image**: Created on 2020-10-20 20:33:05. Description: wqe. Model name: Image Empathy(image). Data input type: image.
- ji-ier-image-trained**: Created on 2020-10-20 20:35:21. Description: qweqwe1. Model name: Image Empathy(image). Data input type: image.

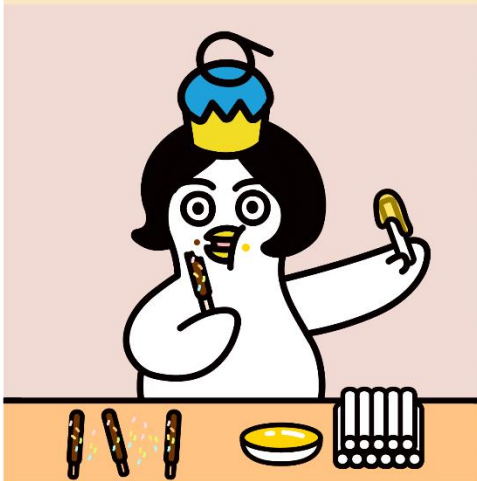
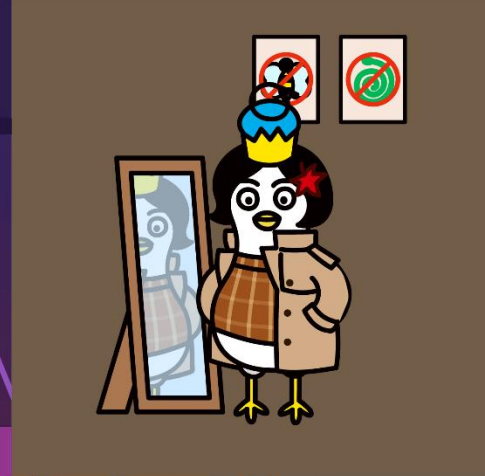
- 실행된 배포는 서비스 화면에서 목록으로 확인 가능합니다.
- 각 서비스 카드를 선택하여 테스트 화면으로 이동할 수 있습니다.

## 학습 모델 배포 및 테스트 - 학습 모델 테스트

The screenshot displays the 'ws-for-berkeley' workspace in the Jonathan Flightbase interface. The main content area shows the 'ji-text-classifier' model with a 'Text Classifier' model name. The input field contains the text '맛집 가고 싶다'. A '분석 시작' (Start Analysis) button is visible. Below, the '분석 결과' (Analysis Results) section shows a donut chart and a table of results.


Category	Percentage
음식/맛집	98.6%
의료/건강/다이얼트/운동	1.9%
국내	1.1%
없음	1.1%
주거/생활/가전	0.4%

- 테스트 화면은 API 가이드라인에 따라 작성된 경우, 자동적으로 생성됩니다.
- 입력 데이터를 입력 후 "분석 시작" 버튼을 클릭하면 모델의 결과를 다양한 형태로 보여주며, JSON 형태로도 확인이 가능합니다.



## Follow Us

 [iacryl.com/acryl/](http://iacryl.com/acryl/)

 [facebook.com/iacryl/](https://facebook.com/iacryl/)

 @i\_acryl

# acryl

KyungJun Lee | Jf

11F, Cheongdam Building, 704, Seolleung-ro, Gangnam-gu, Seoul, Republic of Korea. 06069

Tel: 82 2 557 4958 Fax: 82 2 558 4958

[support@iacryl.com](mailto:support@iacryl.com)

The contents in this document are © copyright 2020 Acryl Inc. All rights are reserved.

A license is hereby granted to download and print a copy of this document for personal use only.

No other license to any other intellectual property rights is granted herein.

Unless expressly permitted herein, reproduction, transfer, distribution of storage of part or all of the contents in any form without the prior written permission of ACRYL Inc. is prohibited.

The information presented in this document does not form part of any quotation or contract, is believed to be accurate and reliable and may be changed without notice.

No liability will be accepted by the publisher for any consequence of its use. Publication thereof does not convey nor imply any license under patent or other industrial or intellectual property rights.