

# 임무 소프트웨어 Guide

WebRTC 영상 스트리밍

msw\_webrtc

# 목 차

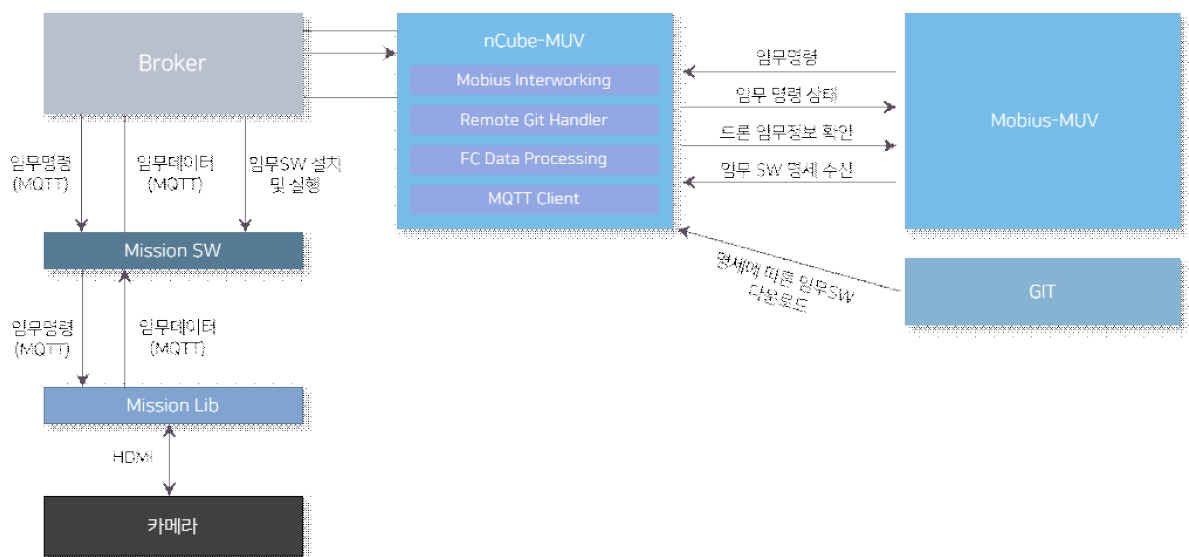
1	개요 .....	3
2	임무장비 .....	4
2.1	카메라 .....	4
3	임무Lib 및 임무SW 개발 .....	5
3.1	임무Lib, 임무SW와 개발도구 .....	5
3.2	임무Lib 개발 .....	5
3.3	임무SW 개발 .....	6

# 1 개요

이 문서는 임무 SW를 통해 카메라를 연동하고 WebRTC를 실행하기 위해 임무 SW를 구현하고 동작하는 방법을 설명하는 문서이다.

여기서 설명하는 WebRTC는 중앙 서버에 WebRTC 서버 오픈 소스인 Janus 기반 Signaling 서버가 구현되어 있고 미션 컴퓨터(Mission Computer, MC)에서 임무 SW를 통해 영상 스트리밍 애플리케이션을 실행한다. KETI-GCS에서는 앞서 실행된 임무 SW 기반 영상 스트리밍 애플리케이션을 통해 전송된 드론 카메라 영상을 확인할 수 있다.

이를 위해 임무장비(카메라)와 직접적으로 연결되어 제어하는 임무라이브러리(Library, Lib), 임무 Lib를 포함한 형태로 nCube-MUV와 연동되어 외부와 통신하며 임무 Lib를 제어하는 임무 소프트웨어(Software, SW)에 대해 설명하고 동작하기 위한 방법을 설명한다.



< WebRTC 임무 소프트웨어(msw\_webrtc) 아키텍처 >

## 2 임무장비

### 2.1 카메라

WebRTC를 위한 카메라로는 HDMI 포트가 지원되는 카메라 또는 인터넷 브라우저에서 웹캠으로 인식 가능한 카메라는 모두 사용 가능하다. 연결 방식은 비디오 캡처 장치를 통한 방식, CSI를 통한 방식과 USB를 통한 방식이 있다.



<WebRTC를 위한 카메라 예시>

- 웹캠 인식 카메라
  - 별도의 비디오 캡처 장치 없이 MC와 바로 연결 가능하다.
- 그 외 카메라
  - 아래 그림과 같은 비디오 캡처 장치 또는 CSI 어댑터를 거쳐 MC와 연결해야한다.



<비디오 캡처 장치와 CSI 어댑터 예시>

### 3 임무Lib 및 임무SW 개발

#### 3.1 임무Lib, 임무SW와 개발도구

임무Lib는 임무 장비와 직접적으로 연결되어 임무 장비를 제어하는 소프트웨어이다. 필수 구현 요소로 임무 장비와 연결하기 위한 인터페이스(Serial, I2C, PWM 등) 연동, 내부적으로 임무SW와 통신하기 위한 MQTT 통신에 대한 내용이 있다. 또한 임무 장비, 사용자 또는 환경에 따라 데이터 모델 정의, 통신 모델 정의 등의 내용을 추가할 수 있다. 임무Lib를 임무SW에서 사용을 위해 개발자가 사전에 개발하여 응용프로그램 형태로 배포해야 한다.

임무SW는 임무Lib를 포함한 형태로 개발되며 외부와 통신하며 임무Lib를 제어하는 소프트웨어이다. 임무SW는 응용프로그램 형태의 임무Lib를 실행하기 위한 스크립트를 정의하는 내용이 필수이며, 또한 외부와 통신하기 위해 통신 모델을 정의하고 데이터 모델을 정의하는 내용을 필수로 한다.

개발자는 개발도구를 통해 임무SW를 좀 더 쉽게 개발하고 배포하여 드론에 탑재까지 가능하다. 개발도구는 블록코딩 형식으로 개발할 수 있으며 Github와 연동하여 이미 배포된 임무Lib와 임무SW를 재사용이 가능하고 개발된 임무SW를 배포할 수 있으며 드론과 임무에 대한 명세를 작성함으로써 자동으로 드론에 탑재와 동작이 가능하다.

#### 3.2 임무Lib 개발

WebRTC를 위한 임무Lib는 임무장비와 직접적으로 연결되어 통신하는 다른 임무Lib와 달리 WebRTC 클라이언트를 자동 제어하는 역할을 수행한다. 따라서 WebRTC 클라이언트 자동제어를 위해 웹 브라우저를 제어할 수 있는 임무Lib를 사전에 개발한다.

- 임무Lib는 파이썬 언어로 작성되었다. 또한, 다른 임무Lib와 달리 통신하거나 데이터를 분석하는 내용 대신 대부분이 Selenium 라이브러리를 사용하여 영상 스트리밍 애플리케이션을 실행하고 웹을 자동으로 제어하기 위한 내용이다.
- 임무Lib는 임무SW와 연동을 위해 실행파일 형태로 배포한다. 파이썬으로 개발된 WebRTC 임무Lib는 아래의 명령어를 통해 응용프로그램 형태로 만들 수 있다.

```
$ python3 -m PyInstaller -F lib_webrtc.py
```

- Python으로 개발되어 응용프로그램 형태로 배포된 WebRTC 임무Lib는 아래 Github 주소에서 확인할 수 있다.

[https://github.com/loTKETI/lib\\_webrtc.git](https://github.com/loTKETI/lib_webrtc.git)

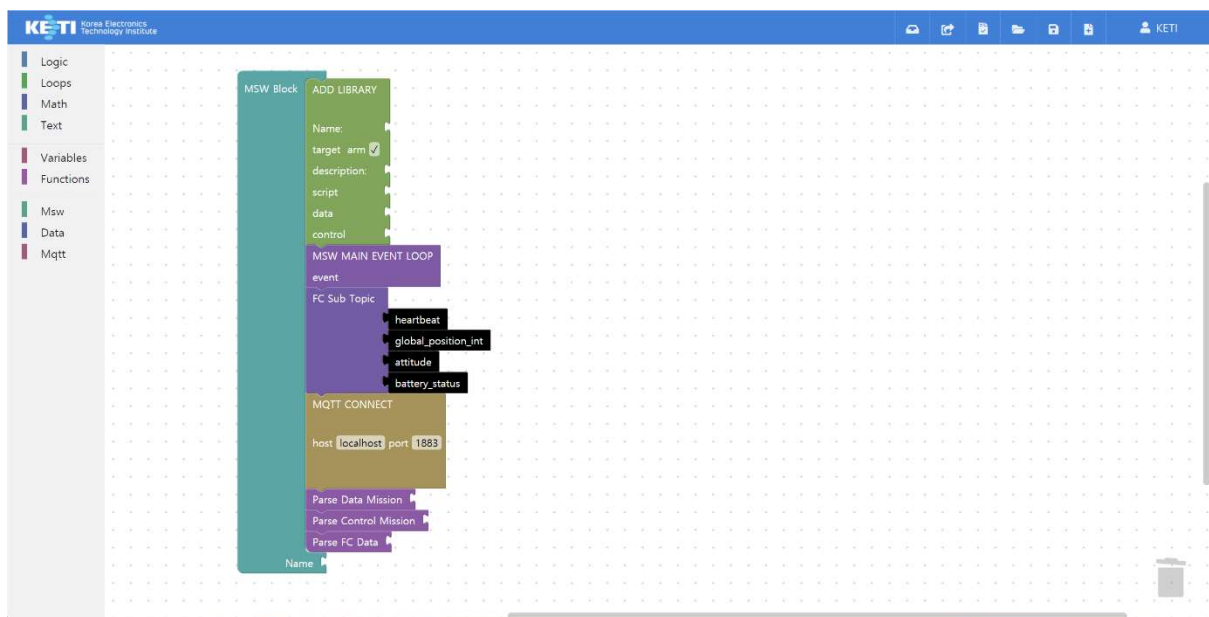
### 3.3 임무SW 개발

#### 3.3.1 MUV 개발도구

- 개발자가 임무SW를 손쉽게 개발하고 배포하여 MC에 자동으로 탑재까지 가능하도록 지원하는 블록코딩 형태의 웹 기반 지원도구이다.
- <http://203.253.128.177:7505> 주소로 접속하여 로그인하여 사용 가능하다.
- 코딩을 소스코드를 작성하는 형태가 아닌 블록코딩을 통해 쉽게 개발할 수 있으며 Github와 연동하여 업로드가 가능하고 approval이라는 명세를 입력함으로써 드론에 탑재가 가능하다.

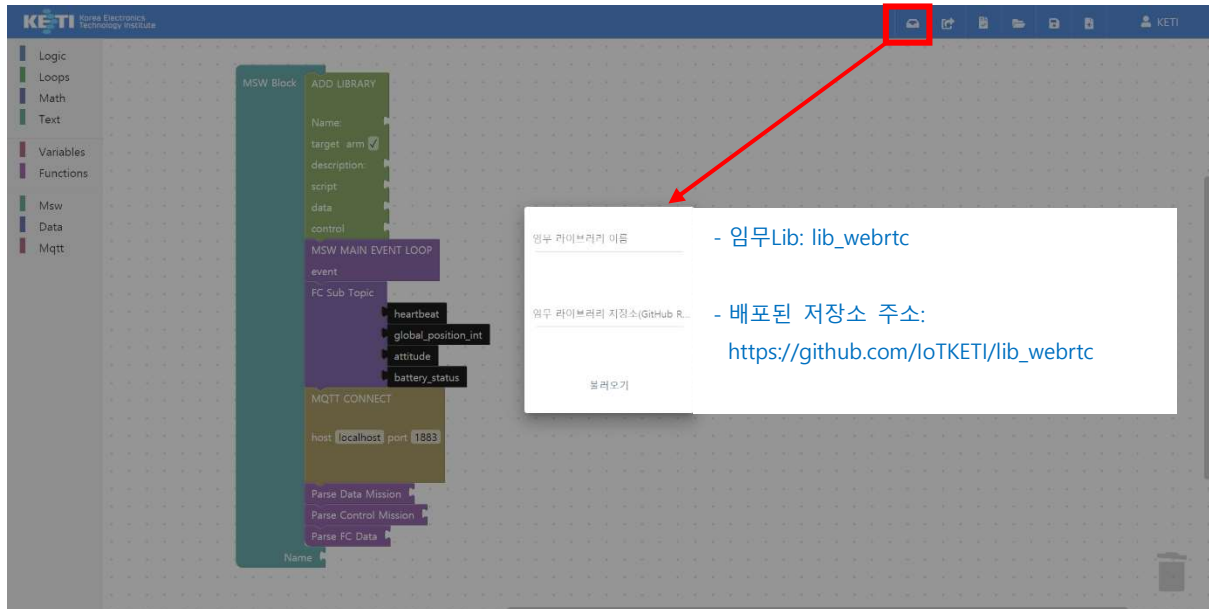
#### 3.3.2 임무 SW 개발과정

- ① 아래 사진과 같이 임무SW의 기본이 되는 형태까지 블록으로 코딩한다. 이 구조는 임무SW도 동일한 구조를 가진다.

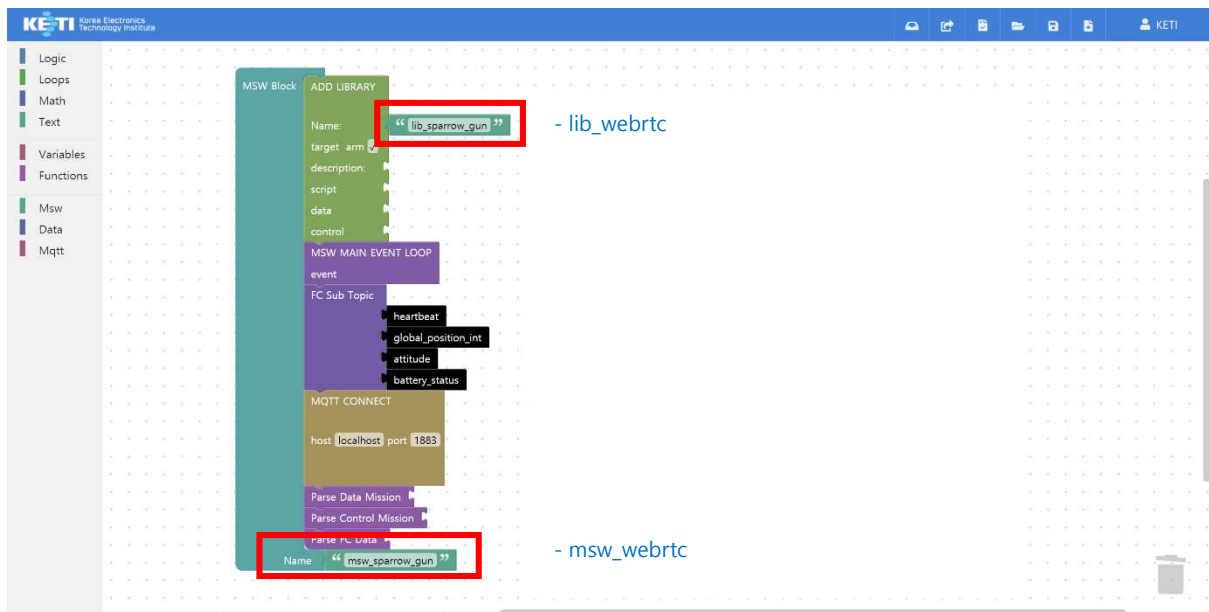


② 사전에 개발해서 배포한 임무Lib를 호출한다.

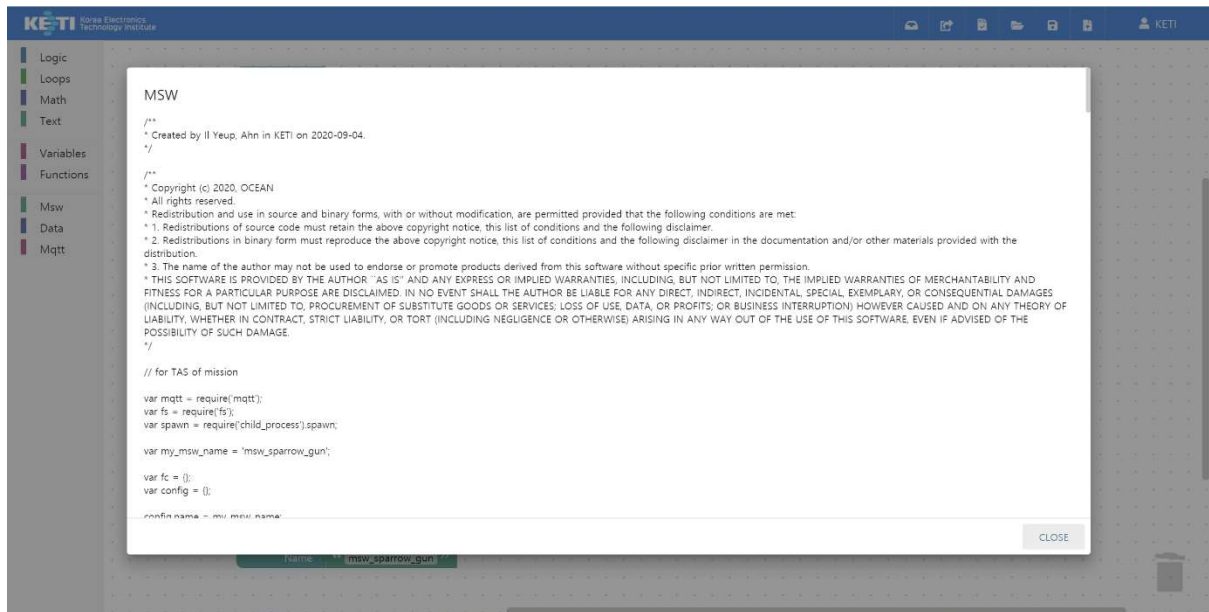
- ✓ 임무Lib 이름: 사전에 응용 프로그램 형태로 배포한 임무Lib의 이름
- ✓ 임무Lib 저장소: Github에 배포된 임무Lib를 가져오기 위한 주소



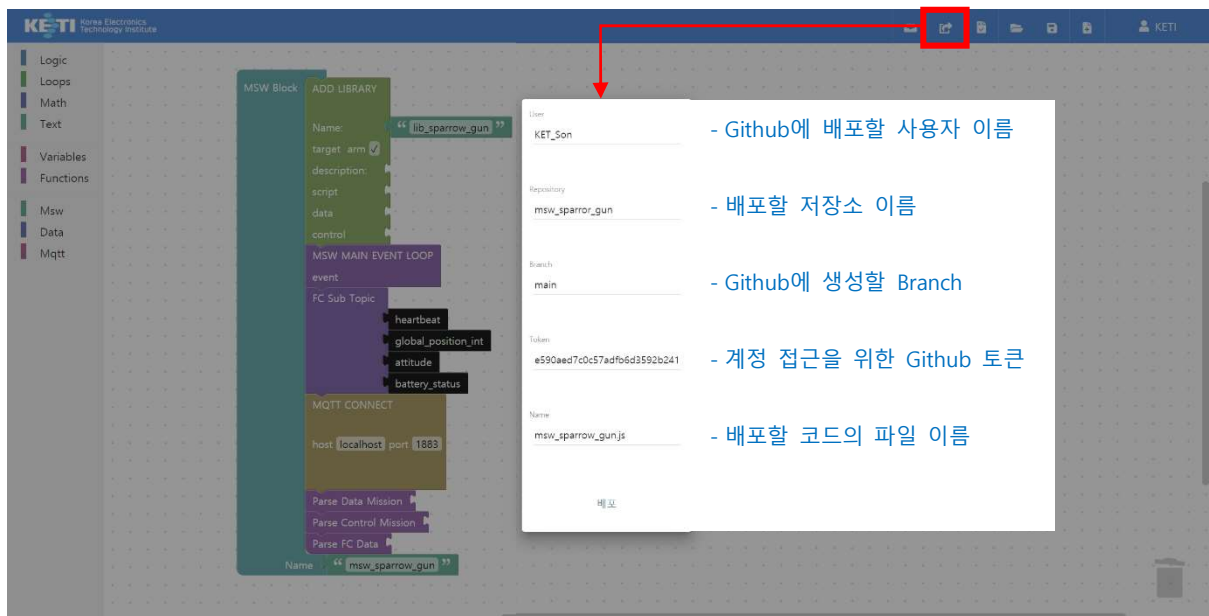
③ 생성된 임무Lib 블록 추가 및 임무SW의 이름 입력  
(lib\_sparrow\_gun 대신 lib\_webrtc)



④ 블록으로 코딩한 임무SW를 배포하기 전 소스코드 형태로 확인

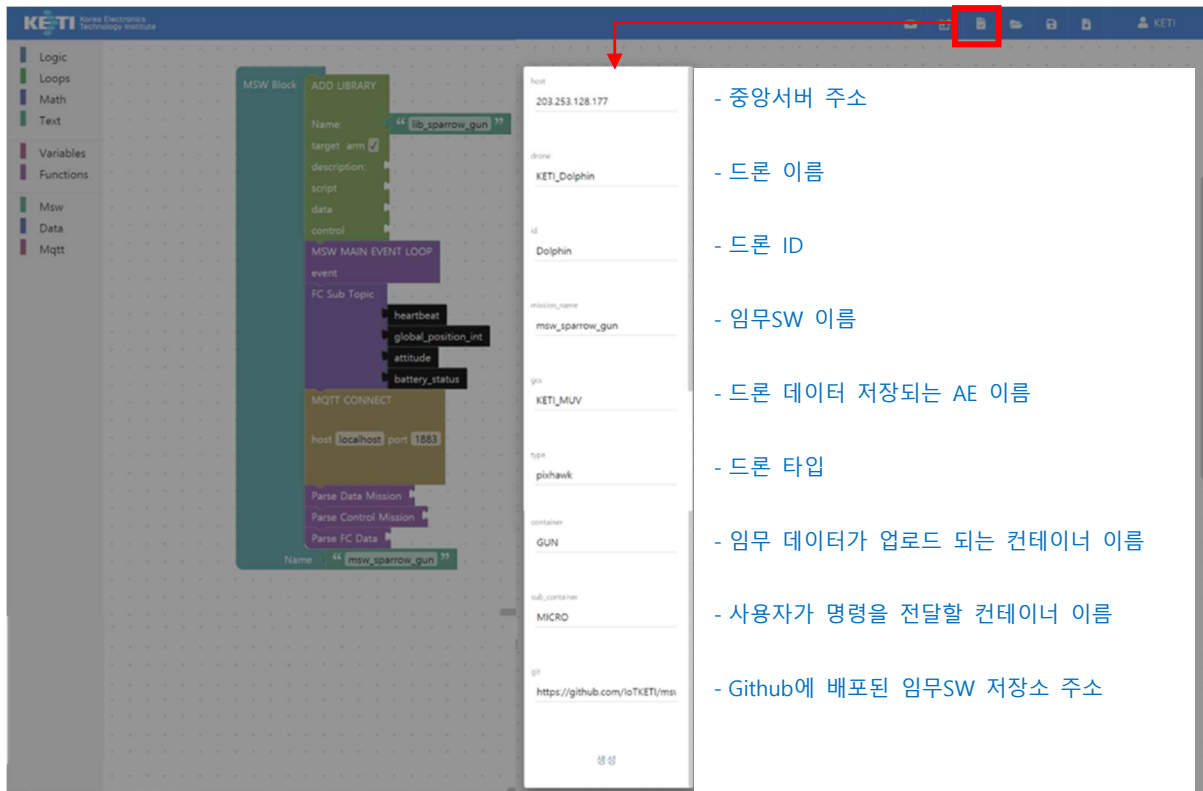


⑤ 임무SW 배포에 필요한 정보 입력하여 배포





⑥ 드론 탑재를 위한 드론 명세 작성



host: 203.253.128.177

drone: KETI\_Dolphin

id: Dolphin

mission\_name: msw\_sparrow\_gun

gcs: KETI\_MUV

type: pixhawk

container: GUN

sub\_container: MICRO

git: <https://github.com/loTKETI/msw>

생성

- 중앙서버 주소
- 드론 이름
- 드론 ID
- 임무SW 이름
- 드론 데이터 저장되는 AE 이름
- 드론 타입
- 임무 데이터가 업로드 되는 컨테이너 이름
- 사용자가 명령을 전달할 컨테이너 이름
- Github에 배포된 임무SW 저장소 주소

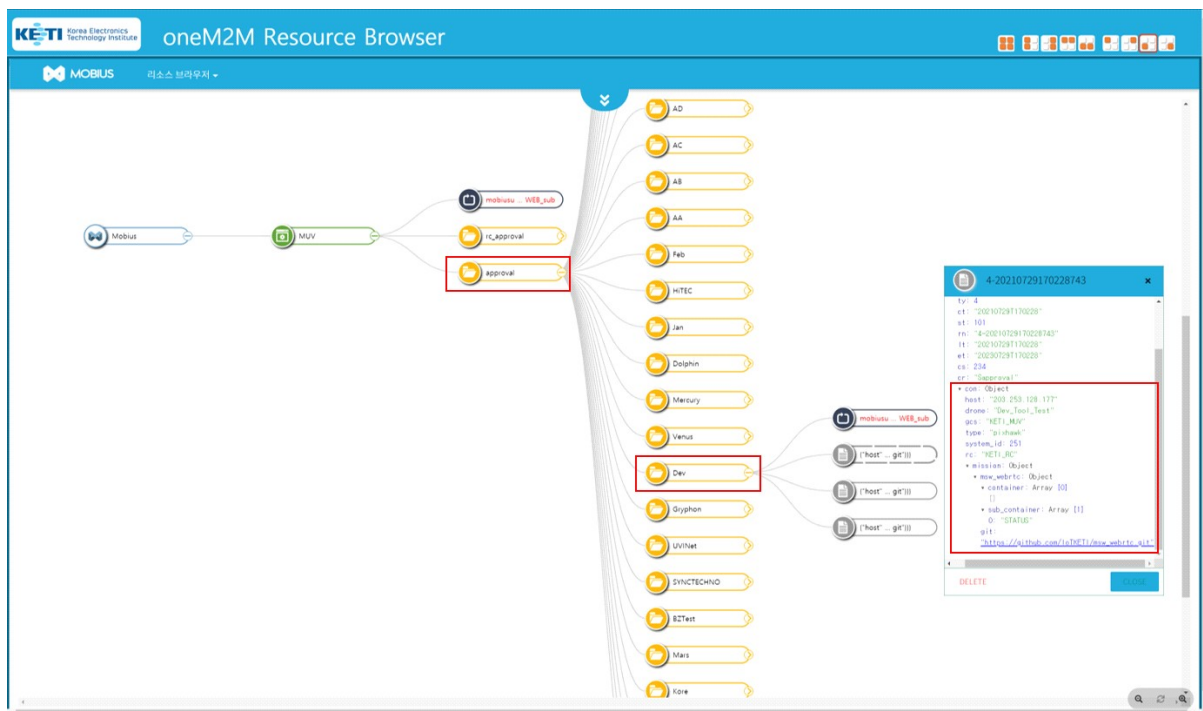
⑦ MC의 nCube-MUV 폴더에서 flight.json 정보 수정 (nCube-MUV 가이드 참고)

- ✓ flight에 ⑥의 과정에서 업로드한 명세의 드론 ID를 추가한다

```

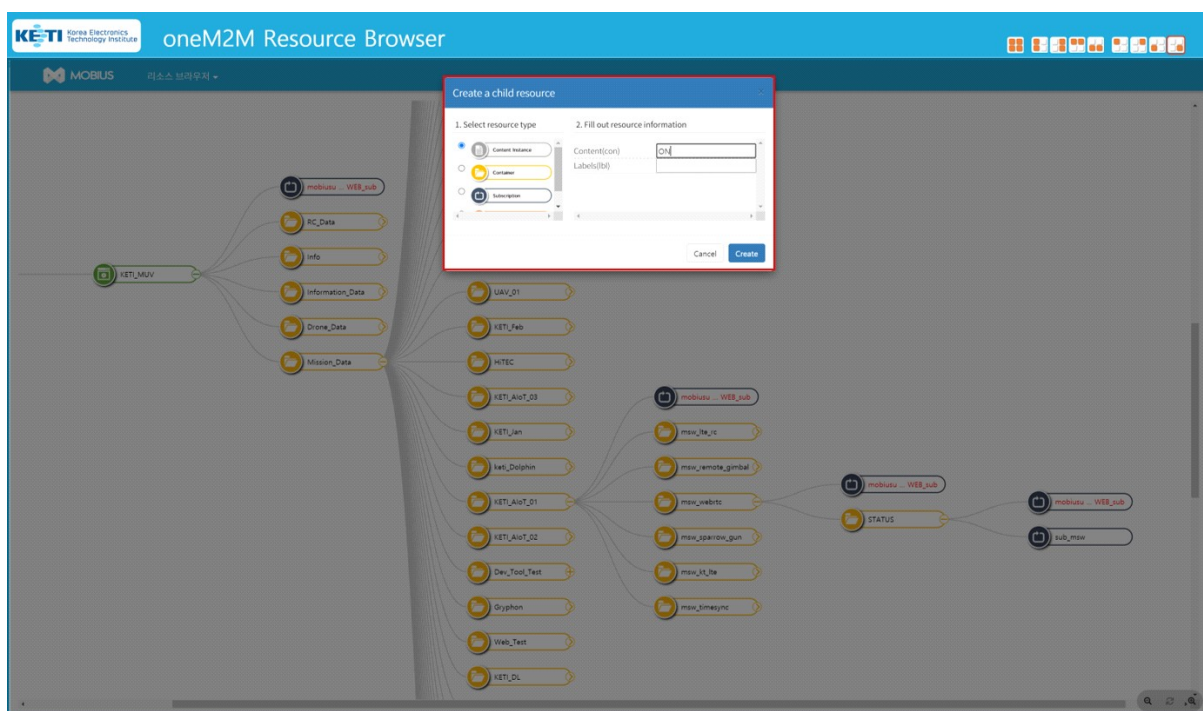
1  {
2      "approval_gcs": "MUV",
3      "flight": "Dev"
4  }
    
```

### ⑧ 명세 업로드 확인



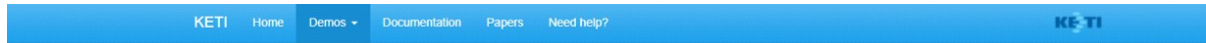
⑨ WebRTC 시작 명령 (제어 명령) 전송

- ✓ /Mobius/{GCS 이름}/Mission\_Data/{드론 이름}/msw\_webrtc/**STATUS** 컨테이너  
아래에 Content Instance를 생성하여 con값에 "ON"을 입력한다.



⑩ WebRTC 스트리밍 화면

- ✓ 임무Lib의 Selenium 라이브러리를 통해 인터넷 브라우저를 자동화하여 WebRTC 실행에 필요한 정보를 자동으로 입력하고 실행까지 자동으로 한다.
- ✓ 최종적으로 사용자는 드론에 전원을 넣고 WebRTC 시작 명령만 전달하면 자동으로 WebRTC가 실행되며 영상을 전송한다.



WebRTC VideoRoom Stop

