

# GPU Enabled Serverless Computing on OpenStack Picasso

OpenStack Days 2017

KAIST 소프트웨어 대학원 김재욱  
(RESL/지도교수 김대영)

# 본 발표의 주요 주제

- Serverless Computing
- GPGPU
- GPU Enabled Serverless Computing
  - IronFunctions
  - OpenStack Picasso

# What is Serverless Computing?

# Serverless Computing

Compute  
Service

가상머신에 서버 환경 구축 및 관리

Permanent  
Process

가상머신이 Running 상태일시 과금

**Cloud Computing**

·  
·  
·  
·  
·  
·

Compute  
Service

서버 구축과 관리할 필요 없이  
함수 단위로 코드 실행

Event Driven

함수가 실행될때에만 과금

**Serverless Computing**

# Serverless Computing

Lambda > New function

Step 1: Select blueprint

Step 2: Configure function

Step 3: Review

## Configure function

A Lambda function consists of the custom code you want to execute. [Learn more](#) about Lambda functions.

Name\* HelloWorld  
Description test lambda function  
Runtime\* Node.js

## Lambda function code

Provide the code for your function. Use the editor if your code does not require custom libraries (other than the aws-sdk). If you need custom libraries, you can upload your code and libraries as a .ZIP file. [Learn more](#) about deploying Lambda functions.

Code entry type ☒ Edit code inline ☐ Upload a .ZIP file ☐ Upload a .ZIP from Amazon S3

```
1 console.log('Loading function');  
2  
3 exports.handler = function(event, context) {  
4   console.log('event: ', JSON.stringify(event));  
5   var name = event.myname || 'Anonymous';  
6   context.succeed('Hello World, ' + name);  
7 };
```

## Lambda function handler and role

Handler\* index.handler  
Role\* lambda\_basic\_execution

## Advanced settings

These settings allow you to control the code execution performance and costs for your Lambda function. Changing your resource settings (by selecting memory) or changing the timeout may impact your function cost. [Learn more](#) about how Lambda pricing works.

Memory (MB)\* 128  
Timeout\* 0 min 3 sec

\* These fields are required.

Cancel Previous Next

## 1. 함수 정보 입력

## 2. 코드 작성

## 3. 실행 환경 설정

Resources [Deploy API](#) / - POST - Setup [Delete Method](#)

Choose the integration point for your new method. ⓘ

Integration type ☒ Lambda Function  
☐ HTTP Proxy  
☐ Mock Integration  
[Show advanced](#)

Lambda Region ap-northeast-1  
Lambda Function HelloWorld [Save](#)

## 4. 함수 실행 API 설정

## test Stage Editor

Invoke URL: <https://edzgyy0dx8.execute-api.ap-northeast-1.amazonaws.com/test>

Settings Stage Variables SDK Generation Export Deployment History

Configure the metering and caching settings for the test stage.

## Cache Settings

Enable API cache ☐

## CloudWatch Settings

Enable CloudWatch Logs ☐

Enable CloudWatch Metrics ☐

## Throttling Settings

Rate 500

Burst Limit 1000

## Client Certificate

Select the client certificate that API Gateway will use to call your integration endpoints in this stage.

None None

[Save Changes](#)

## 5. API 생성

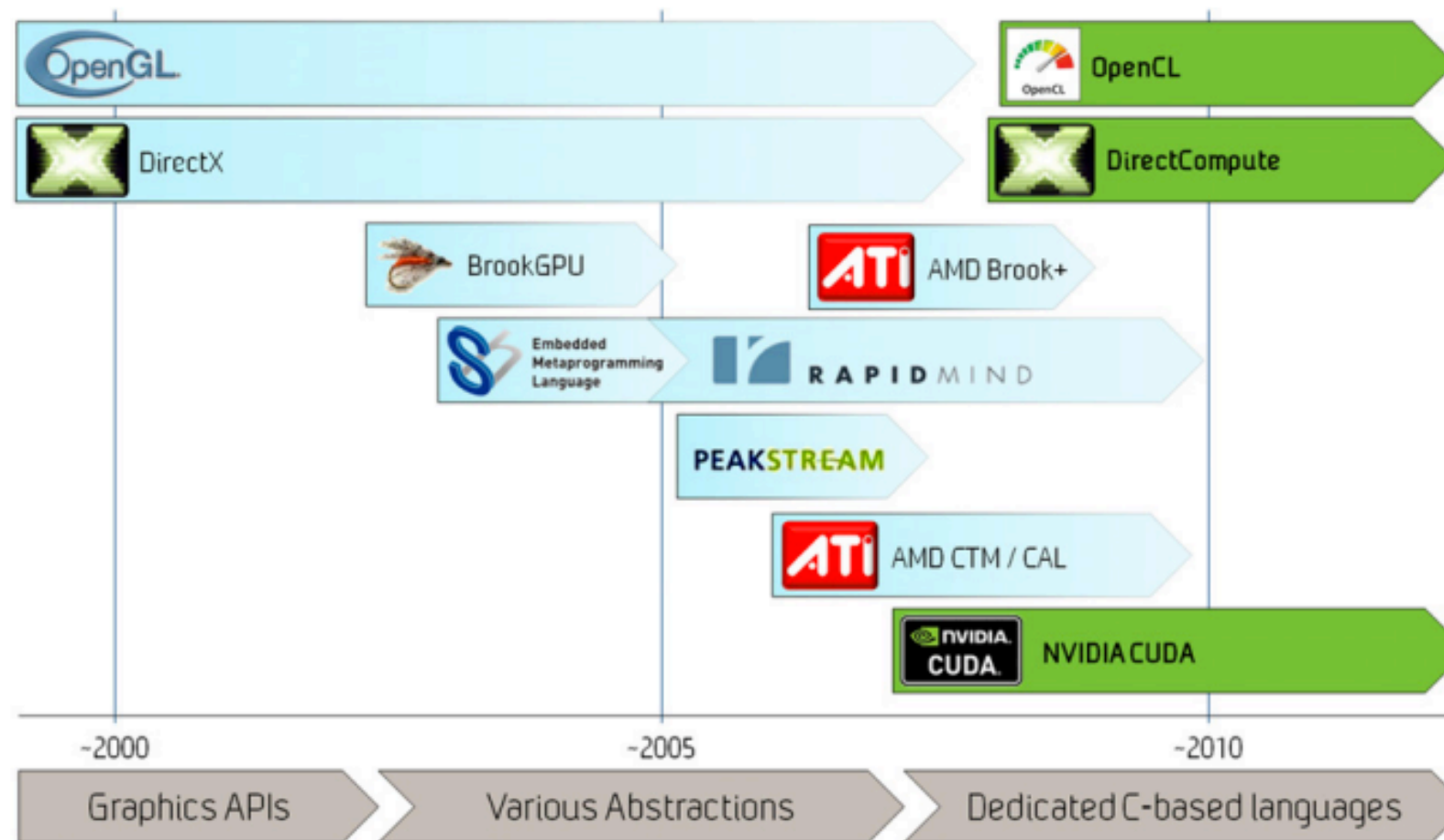
# Serverless Computing

예제 : AWS Lambda를 활용한 이미지 썸네일 만들기



# GPGPU

# History of programming languages for GPU computing




- 2000년대의 GPU는 주로 Graphic을 빠르게 표현하기 위해 사용되었다.
- 현재는 CUDA, OpenCL 등으로 Graphic 연산 뿐만 아니라 General한 목적으로 GPU가 사용되고 있다.
- 특히 요즘엔 머신러닝에 GPU가 많이 사용되고 있다.




# Serverless + GPGPU

## Is the GPU available in AWS Lambda?

 Reply

Posted by: [sajman](#)

Posted on: Jun 16, 2015 11:44 AM

 [lambda](#) , [opengl](#) , [gpu](#)

★ This question is **not answered**. Answer it to earn points.

Is a GPU available for AWS Lambda? Can I run OpenGL based code on it?

Edited by: [sajman](#) on Jun 16, 2015 11:45 AM

Lambda에서 GPU를 쓸수있나?

Replies: 2 | Pages: 1 - Last Post: Mar 16, 2017 2:58 PM by: [Even](#)

### Replies

#### Re: Is the GPU available in AWS Lambda?

 Reply

Posted by:  [WilliamG@AWS](#)

Posted on: Jun 19, 2015 10:58 AM

 in response to: [sajman](#)

Hi sajman,

아직은 안된다. 고려하겠다.

The GPU is not available at this time; however, we will consider it as a feature request. Thank you for your feedback!

#### Re: Is the GPU available in AWS Lambda?

 Reply

Posted by: [Even](#)

Posted on: Mar 16, 2017 2:58 PM

 in response to: [WilliamG@AWS](#)

 [lambda](#) , [gpu](#) , [deep](#) , [learning](#) , [tensorflow](#)

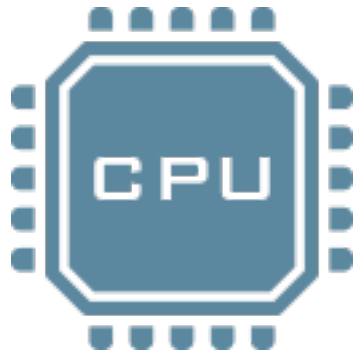
포스트 이후로 시간이 많이 지났는데  
아직 안되나? 나의 Lambda 서비스에  
서 Tensorflow를 쓰고싶다.

Hey William, I'm curious on the timeline. With the rise of deep learning there's going to be a huge demand for this sort of on demand gpu compute capacity.

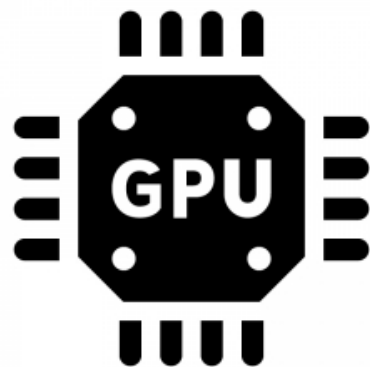
I'm currently looking into deploying tensorflow in lambda as it meets my use case perfectly, but without GPU support I'm worried it won't meet my performance needs.

Your post is a year and a half old. Is this something Amazon has in the works?

# Problem Statement

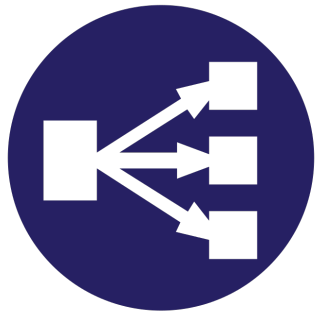


**CPU 기반의 코드를 실행하는데 머물러있는 현재 Serverless Computing의 활용성을 GPU로 확장하여 다양한 분야에서 빠르게 서비스를 제공한다.**



**현재 원격의 GPU를 사용하는 방법은 매우 복잡하다. Serverless 환경에서 쉽고 빠르게 원격의 GPU를 사용하자.**

## Use Case



**Microservice를 배포할 때 GPU Programming를 할수있도록 해서  
고성능의 Microservice를 배포할 수 있다.**

### MicroService



**GPU Programming을(예:Deep Learning) GPU가 없는 PC에서 원격의  
GPU를 사용해서 실행할 수 있다.**

### GPU Share

## Related Works

- AWS Lambda(Amazon), Google Cloud Function(Google), AzureFunction(MicroSoft), OpenWhisk(IBM), IronFunctions(iron.io), Picasso(OpenStack)  
현재 나와있는 Serverless Computing Framework들은 GPU를 지원하지 않는다.
- “rCUDA”, Reducing the number of GPU-based accelerators in high performance clusters(2010)  
GPU 가상화를 통해 원격의 GPU 자원을 사용할 수 있도록하는 연구이다. 원격의 GPU를 사용하여 여러 GPU를 병렬처리에 사용하여 HPC를 구현하는데 사용한다. 사용이 매우 복잡하다.
- Towards Serverless Execution of Scientific Workflows – HyperFlow Case Study(2016)  
Google Cloud Function에 HyperFlow 엔진을 연동하는 코드를 구현하여 Scientific Workflow를 실행하는 연구이다. Serverless 환경에서 Scientific Workflow를 구현할 수 있다고 말한다.

# Why Serverless doesn't support GPU?

- Serverless 환경에서 GPU 사용의 필요성을 못느낀다.
  - Serverless는 현재 매우 짧게 실행되는 함수에 주로 사용되기 때문에 GPU 사용의 필요성을 못느낀다.  
=> 짧게 실행되는 함수를 더 빨리 실행할 수 있다면 서비스를 이용하는 사람에게 더 빠르게 서비스를 제공할 수 있다.  
앞선 연구와 같이 Serverless의 활용성에 대한 연구가 진행되고 있다.

- Container 환경에서 GPU 사용이 어렵다.
  - Linux의 Container를 활용하기 위한 SW는 현재 Docker가 가장 활발하게 활용되고 있다.  
Docker Container에서 GPU를 사용하는 솔루션이 없었다.  
=> 2016년 NVIDIA에서 NVIDIA-Docker를 공개하여 Docker Container에서 NVIDIA GPU를 쉽게 사용할 수 있도록 Docker Middleware를 개발하였다.
- Container 환경에서 GPU 메모리 사용을 컨트롤할 수 없었다.
  - Container 환경에서 GPU의 메모리 사용을 제한하는 솔루션이 없었다.  
=> Daeyoun Kang, Jaewook Kim 외 "ConVGPU: GPU Management Middleware in Container Based Virtualized Environment" (2017) 연구에서 Docker Container 환경에서 GPU 메모리를 관리하는 솔루션을 제안하였다.

## Approach

**OpenStack Picasso**

**+**

**NVIDIA-Docker**

Picasso와 NVIDIA-Docker를 연동하여 Picasso에서 GPU를 사용할 수 있도록 하자.

# OpenStack Picasso



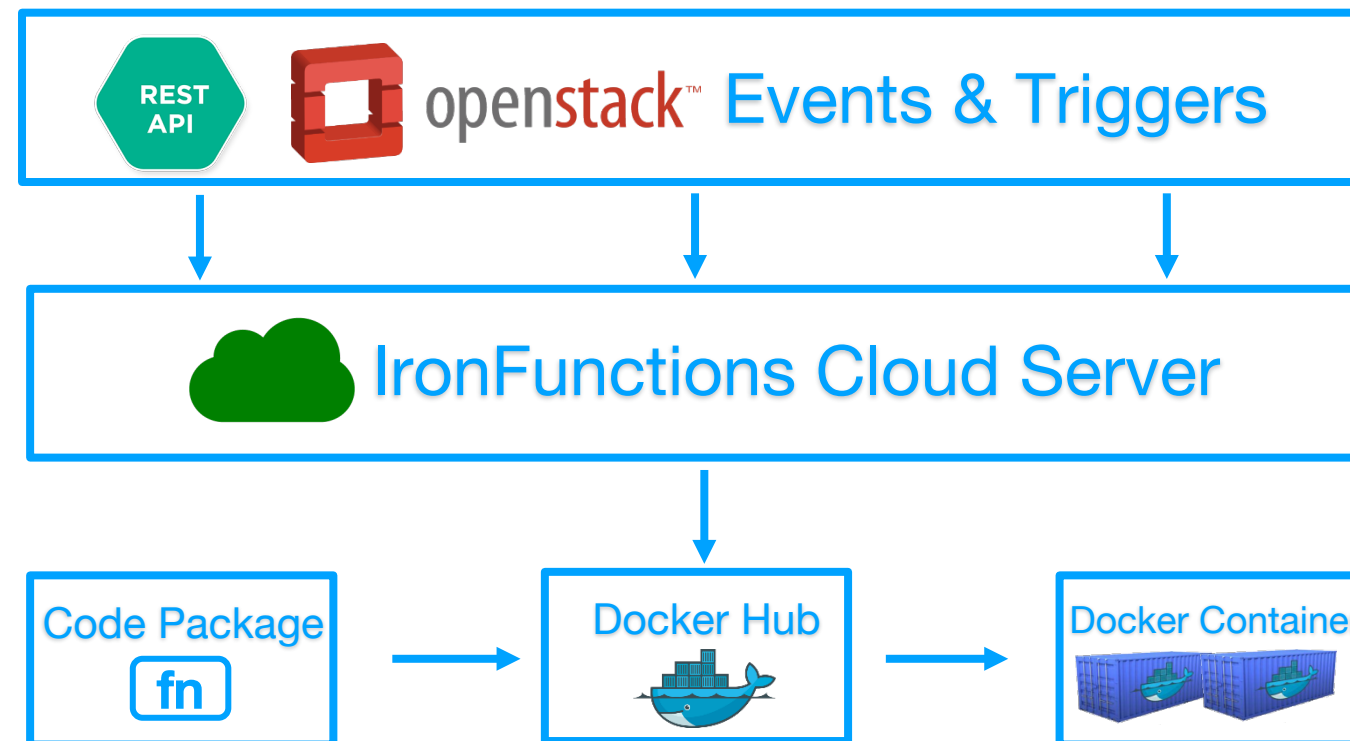
# OpenStack Picasso

- Picasso는 OpenStack의 공식 프로젝트로 OpenStack에서 Serverless 환경을 제공하도록 개발되었다.
- Picasso는 주요한 세개의 프로젝트로 구성되어있다.
  - A.Picasso API : Picasso의 API 서버로 Keystone과 연동하여 인증 처리를 하고 함수 정보를 데이터베이스에 저장하고 함수를 IronFunctions과 연동한다
  - B.Picasso Client : OpenStack CLI와 연동하여 Command line으로 Picasso API를 호출한다.
  - C.IronFunctions : Picasso는 백엔드 엔진으로 오픈소스 Serverless/FaaS 프레임워크인 IronFunctions을 사용한다.

# OpenStack Picasso

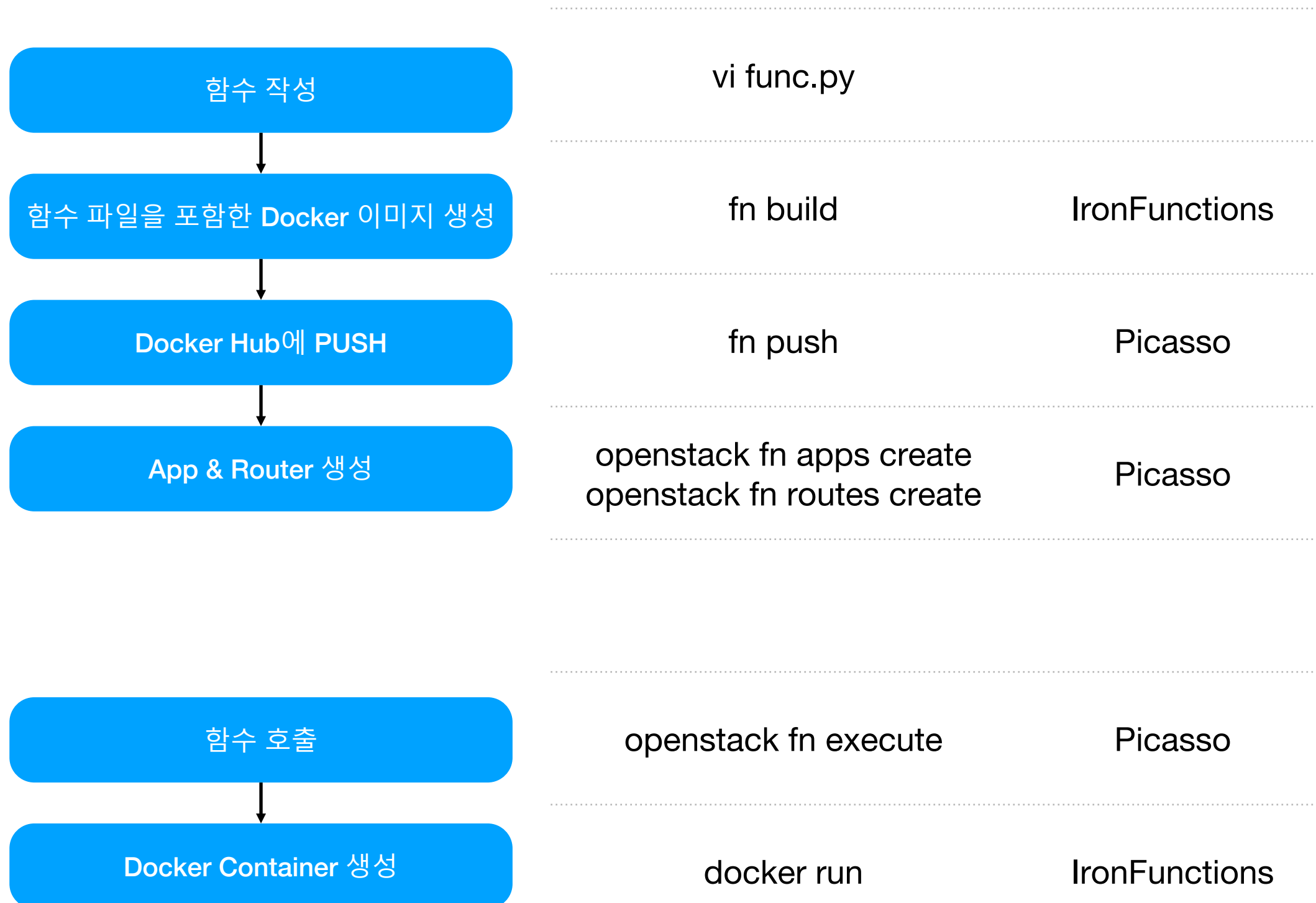
- 현재 Picasso에서 제공하는 API는 함수와 HTTP REST API를 연결하는 API Gateway 역할만 제공하고 있다.
- 함수 생성 및 실행은 IronFunctions를 통해서 이루어진다.
- System Requirements
  - Python 3.5 or greater
  - MySQL 3.7 or greater
  - IronFunctions
    - Go, glide, Docker
  - Ubuntu 16.04 + Devstack(stable/newton)

# OpenStack Picasso - IronFunctions

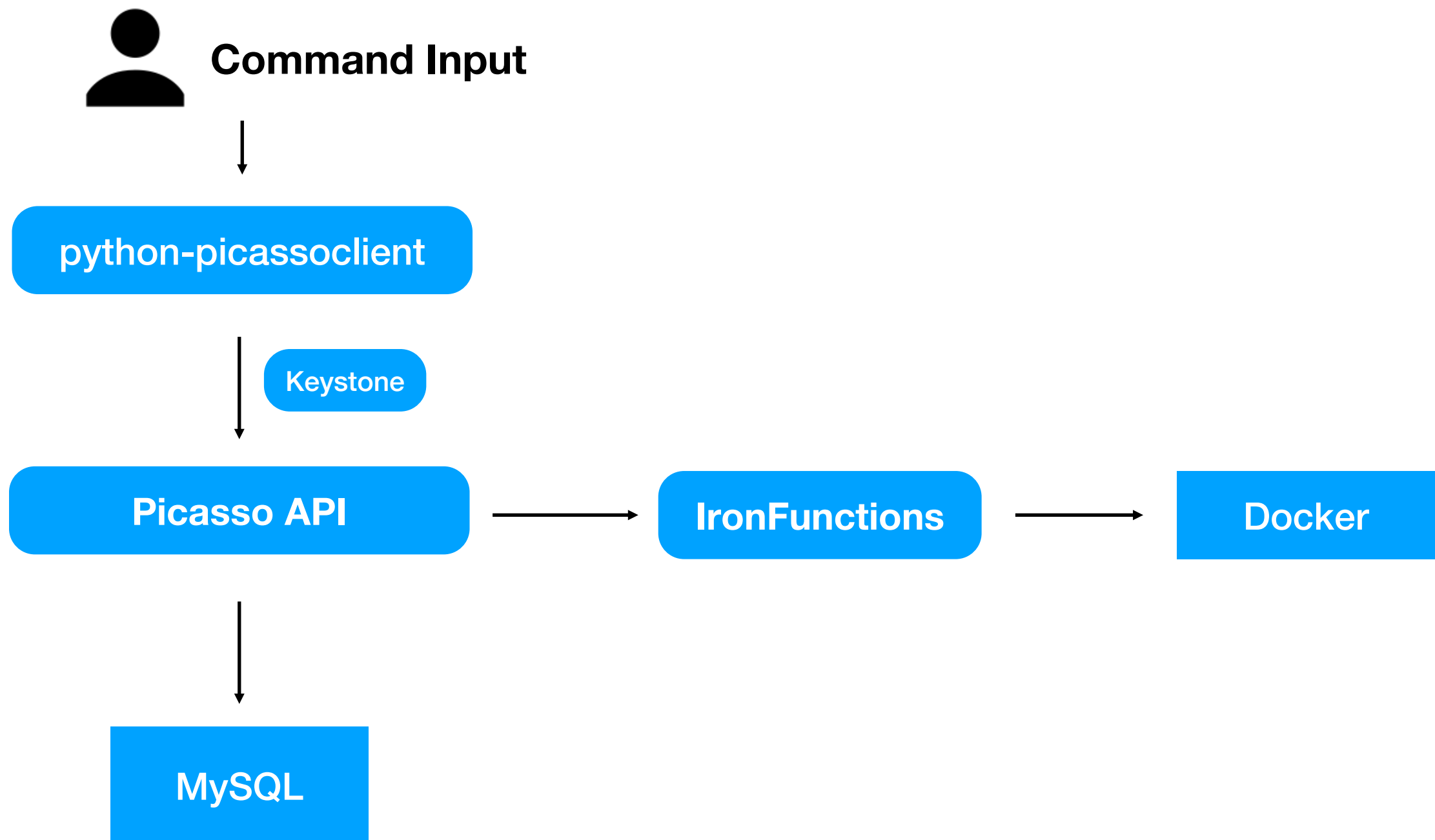


- IronFunctions는 iron.io에서 개발한 Open Source Serverless Computing Framework이다.
- 서버와 클라이언트 코드 모두가 Open Source이다.
- 함수 Code를 Docker Hub를 통해 서버/클라이언트가 주고 받는다.
- 코드의 실행은 Docker Container 내에서 실행한다.

# OpenStack Picasso Process

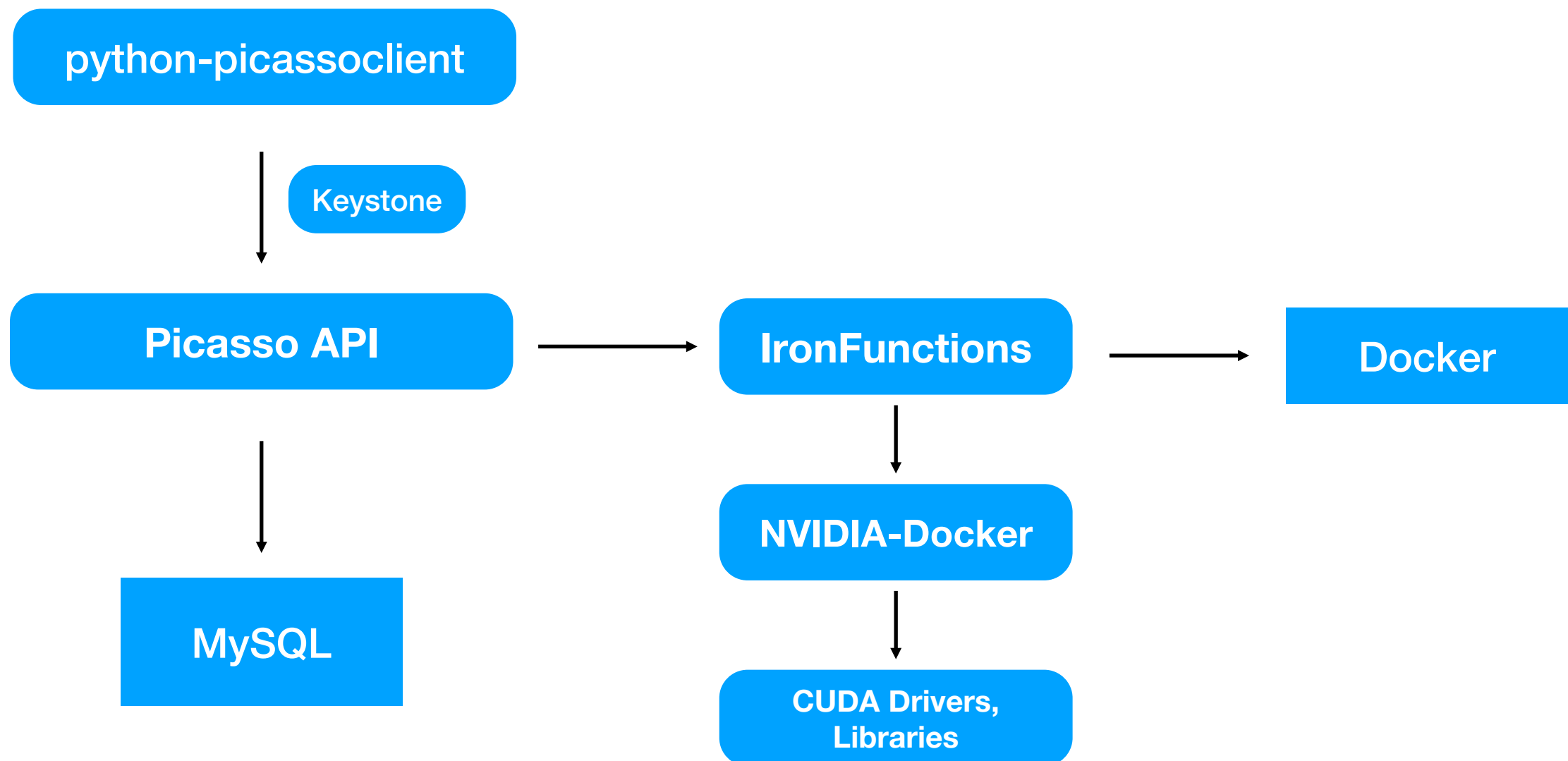


# OpenStack Picasso Architecture



# GPU Enabled OpenStack Picasso

- Picasso는 IronFunctions를 백엔드 엔진으로 사용한다. 따라서 IronFunctions과 NVIDIA-Docker를 연동한다.



# IronFunctions (기존 Architecture)

## Modified IronFunctions (Architecture 변경)



**Demo**

## Conclusion

- IronFunctions와 NVIDIA-Docker를 연동하여 OpenStack 기반의 Serverless Computing 환경에서 GPU를 사용할 수 있는 프레임워크를 제시하였다.
- 이미지, 비디오 프로세싱과 같은 GPU를 사용했을 때 고성능을 내는 함수의 경우 GPU Serverless 기반으로 배포한다면 더 빠르게 서비스를 제공할 수 있다.
- 서버의 GPU를 사용해서 GPU가 없는 PC에서도 손쉽게 GPU Programming을 할 수 있다.
- 해외 Conference paper submitted

## Limitation & Future works

- IronFunctions의 성능이 다른 Serverless Framework보다 느리다. 따라서 Picasso도 성능이 좋지 않다.
- Picasso의 프로젝트 비활성화(최근 커밋이 4달전....)
- UseCase 부족 (그래서 이걸 어디에 쓸건데?)
- GPU Resource Handling 불가능
- Picasso와 ConVGPU를 연동하여 GPU enabled Serverless Framework에서 GPU 메모리를 컨트롤 할 수 있도록 한다.
- OpenStack Picasso에서 백엔드 엔진으로 IronFunctions 대신 OpenStack Magnum을 사용하도록 하여 OpenStack내에서 Container를 컨트롤 할 수 있도록한다.
- Journal Extend(Preparing)

감사합니다