

# Rune

---

Opensource serverless framework

# Contents

목차



- 개발 배경
- 개발 목적
- Rune
- 개발 사항
- 영상 / 시연
- 차후 계획

## [ Server ]

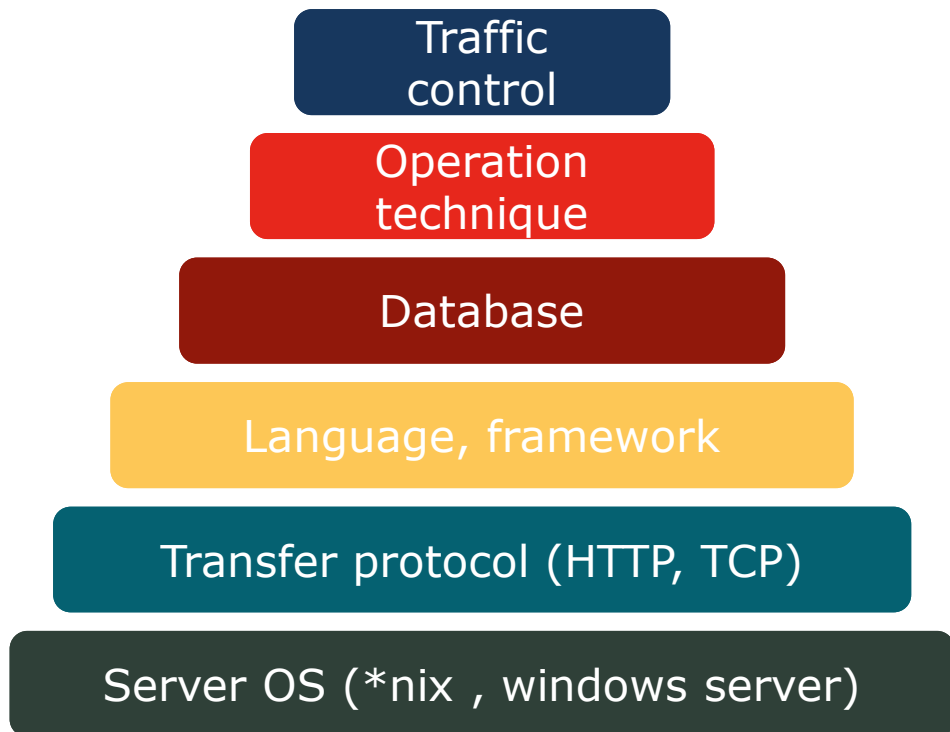
: client의 요청에 네트워크를 통해 응답을 하기 위한 일련의 infrastructure

## [ Server infrastructure ]

: client가 server 와 통신을 하기 위한 제반 요구사항

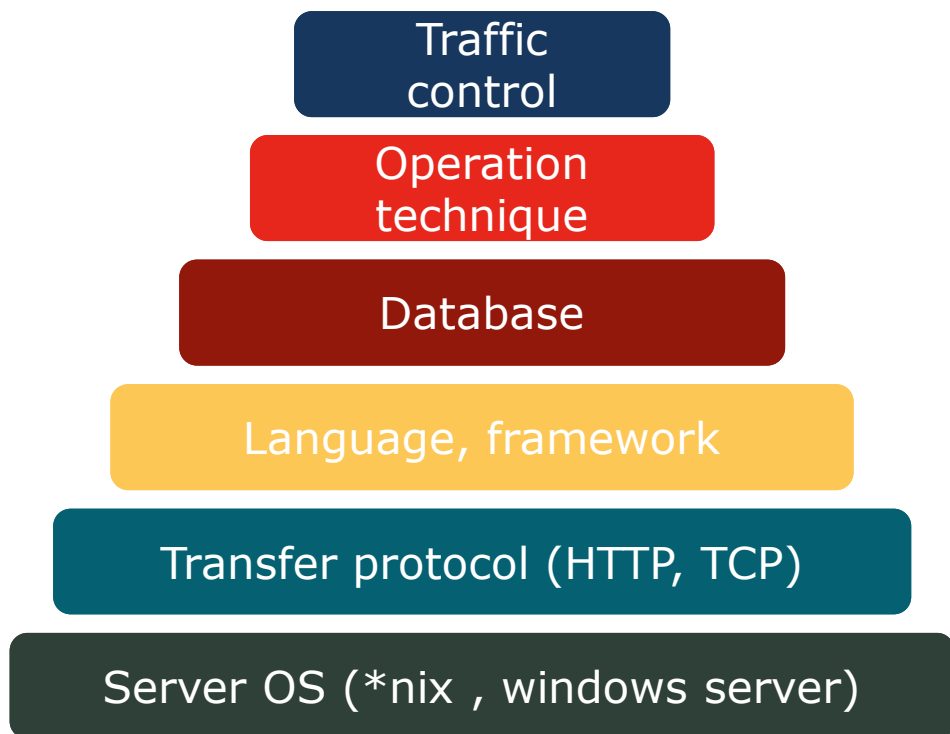
## [ Server infrastructure ]

: client가 server 와 통신을 하기 위한 제반 요구사항



## [ Server infrastructure ]

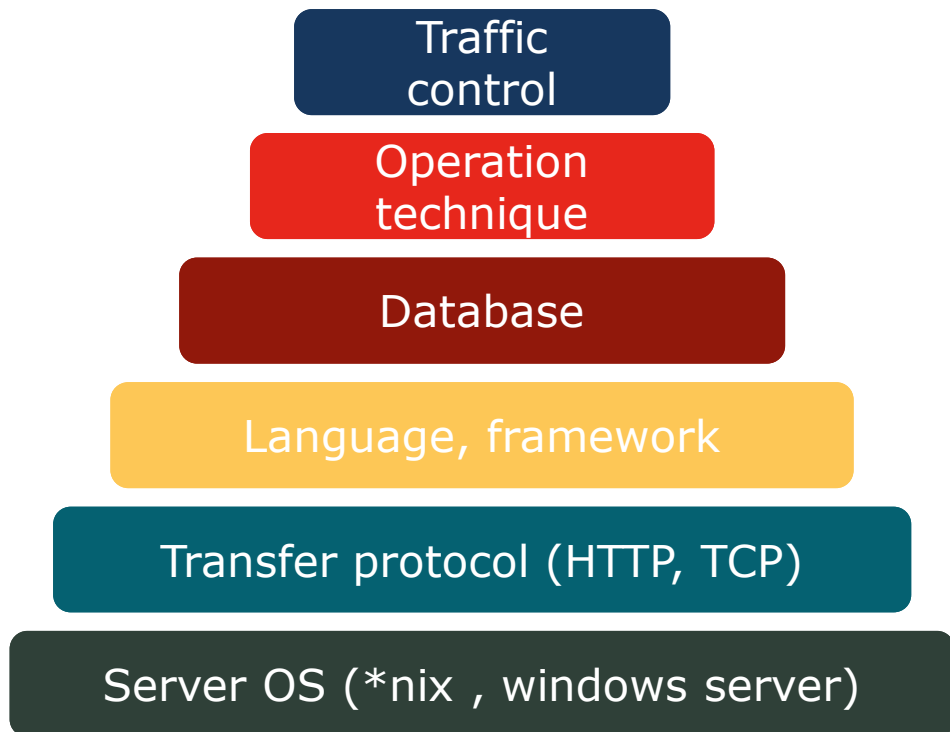
: client가 server 와 통신을 하기 위한 제반 요구사항



초기 환경 설정을 위한 프로그램 설치 및 방화벽 설정

## [ Server infrastructure ]

: client가 server 와 통신을 하기 위한 제반 요구사항

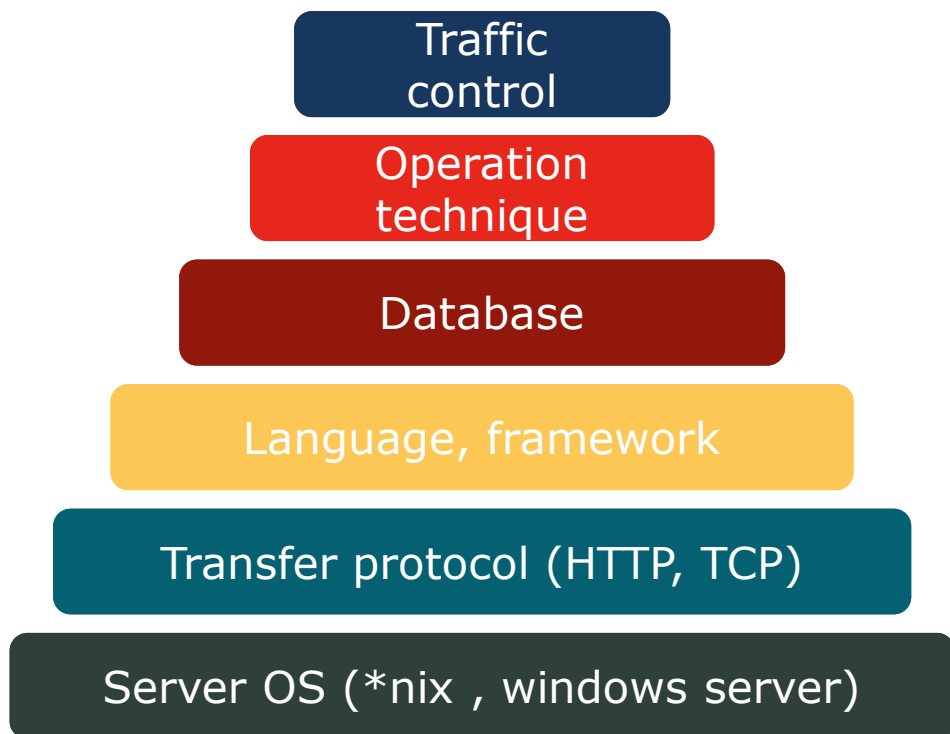


HTTP를 수신하기 위한 서버(nginx, apache) 에 대한 공부

초기 환경 설정을 위한 프로그램 설치 및 방화벽 설정

## [ Server infrastructure ]

: client가 server 와 통신을 하기 위한 제반 요구사항



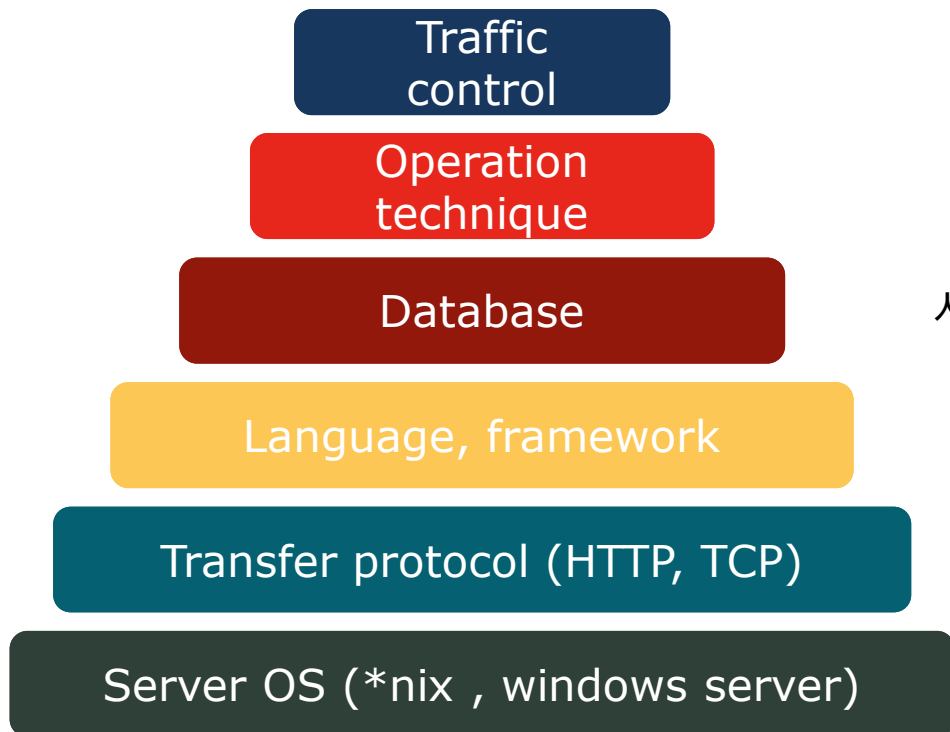
Client를 위한 Java, Swift 가 아닌 Server용 언어와 Framework

HTTP를 수신하기 위한 서버(nginx, apache) 에 대한 공부

초기 환경 설정을 위한 프로그램 설치 및 방화벽 설정

## [ Server infrastructure ]

: client가 server 와 통신을 하기 위한 제반 요구사항



사용자의 정보를 저장하기 위한 Database 설치, 사용 및 운용

Client를 위한 Java, Swift 가 아닌 Server용 언어와 Framework

HTTP를 수신하기 위한 서버(nginx, apache) 에 대한 공부

초기 환경 설정을 위한 프로그램 설치 및 방화벽 설정



# [ Server infrastructure ]

: client가 server 와 통신을 하기 위한 제반 요구사항

Traffic  
control

Operation  
technique

Database

Language, framework

Transfer protocol (HTTP, TCP)

Server OS (\*nix , windows server)

서버가 죽지 않도록 지속 가능한 서버로 만들기 위한 trial & error

사용자의 정보를 저장하기 위한 Database 설치, 사용 및 운용

Client를 위한 Java, Swift 가 아닌 Server용 언어와 Framework

HTTP를 수신하기 위한 서버(nginx, apache) 에 대한 공부

초기 환경 설정을 위한 프로그램 설치 및 방화벽 설정

# [ Server infrastructure ]

: client가 server 와 통신을 하기 위한 제반 요구사항

Traffic  
control

서비스가 흥했을 때 버티기 위한 서버 scaling 기술 확보

Operation  
technique

서버가 죽지 않도록 지속 가능한 서버로 만들기 위한 trial & error

Database

사용자의 정보를 저장하기 위한 Database 설치, 사용 및 운용

Language, framework

Client를 위한 Java, Swift 가 아닌 Server용 언어와 Framework

Transfer protocol (HTTP, TCP)

HTTP를 수신하기 위한 서버(nginx, apache) 에 대한 공부

Server OS (\*nix , windows server)

초기 환경 설정을 위한 프로그램 설치 및 방화벽 설정

## [ Server infrastructure ]

: client가 server 와 통신을 하기 위한 제반 요구사항

Traffic  
control

서비스가 흥했을 때 버티기 위한 서버 scaling 기술 확보

Operation  
technique

어렵다

제반가 족지 않도록 지속 가능한 서버로 만들기 위한 trial & error

Database

사용자의 정보를 저장하기 위한 Database 설치, 사용 및 운용

Language, framework

Client를 위한 Java, Swift 가 아닌 Server용 언어와 Framework

Transfer protocol (HTTP, TCP)

HTTP를 수신하기 위한 서버(nginx, apache) 에 대한 공부

Server OS (\*nix , windows server)

초기 환경 설정을 위한 프로그램 설치 및 방화벽 설정



개발 배경 Background

그러면  
PaaS  
로가자!

## [ PaaS(Platform as a Service) ]

: 플랫폼 자체를 서비스로 제공하여 infrastructure에 대한 고민을 제거



Google App Engine

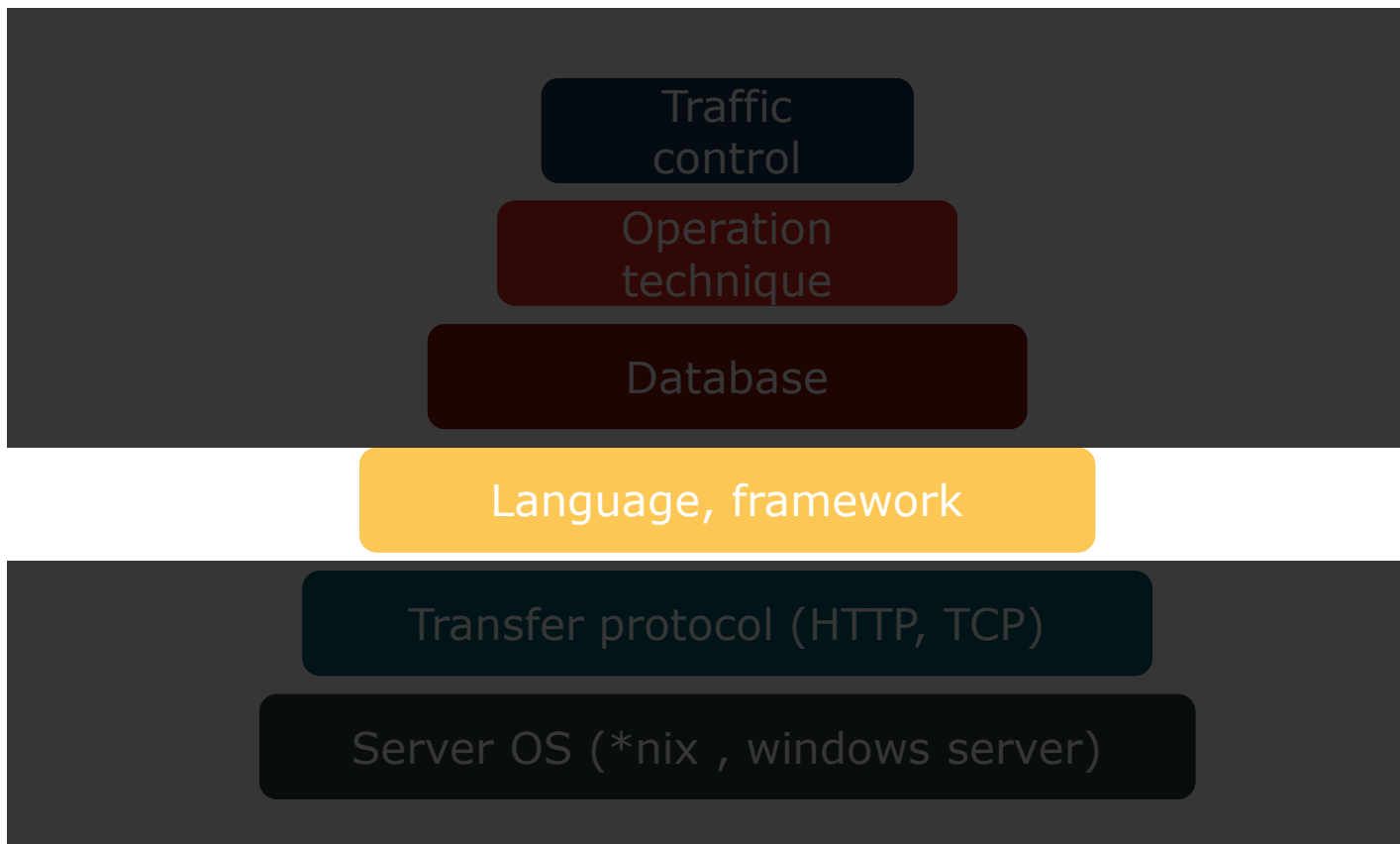


IBM Bluemix

IBM Bluemix



Microsoft Azure



## [ PaaS(Platform as a Service) ]

: 플랫폼 자체를 서비스로 제공하여 infrastructure에 대한 고민을 제거

-> 초창기 Service에는 편리하나 과도한 비용 제출 및 Platform dependencies의 과도한 증대

## [ PaaS(Platform as a Service) ]

: 플랫폼 자체를 서비스로 제공하여 infrastructure에 대한 고민을 제거

-> 초창기 Service에는 편리하나 과도한 비용 제출 및 Platform dependencies의 과도한 증대

월 10억번 호출 당 가격:

**₩4,202,359 (\$3686.28)**

사백이십이만삼백오십구원

기준 환율 1140

- \* DAU 10만 기준, 유저당 300 Request
- \* Google App Engine 기준





## [ PaaS(Platform as a Service) ]

: 플랫폼 자체를 서비스로 제공하여 infrastructure에 대한 고민을 제거

-> 초창기 Service에는 편리하나 과도한 비용 제출 및 Platform dependencies의 과도한 증대

# 비싸다

월 10억번 호출 당 가격:

**₩4,202,359 (\$3686.28)**

사백이십이만삼백오십구원

기준 환율 1140

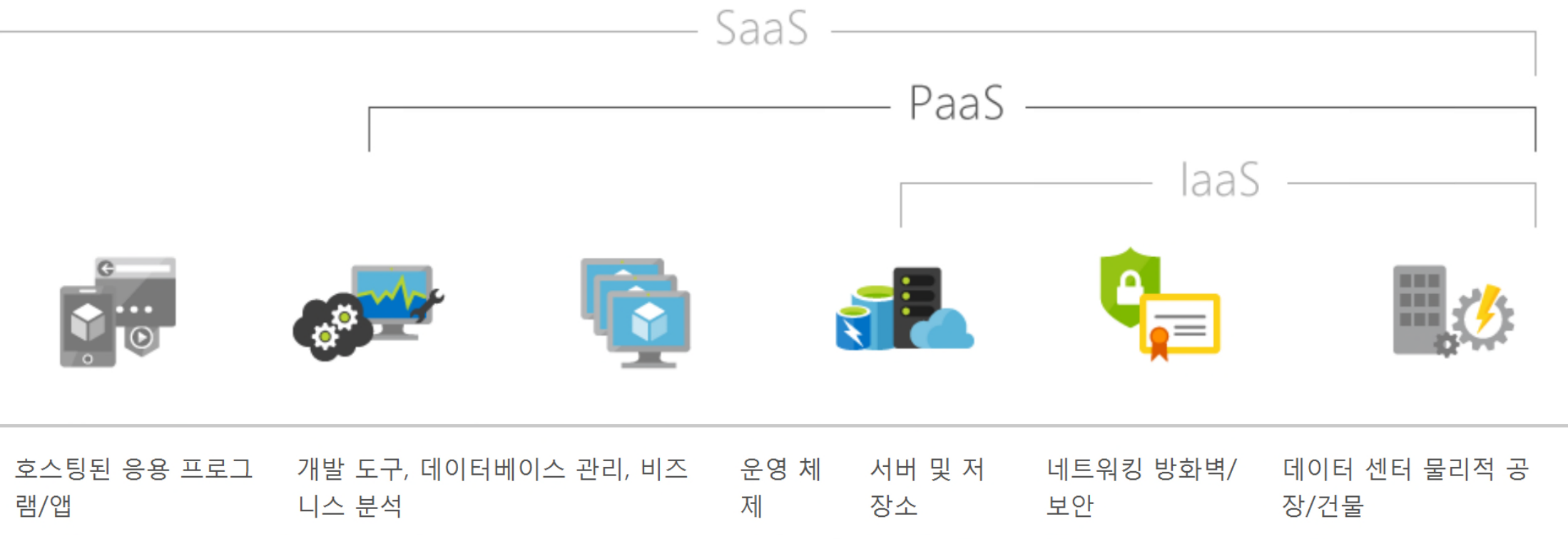
\* DAU 10만 기준, 유저당 300 Request

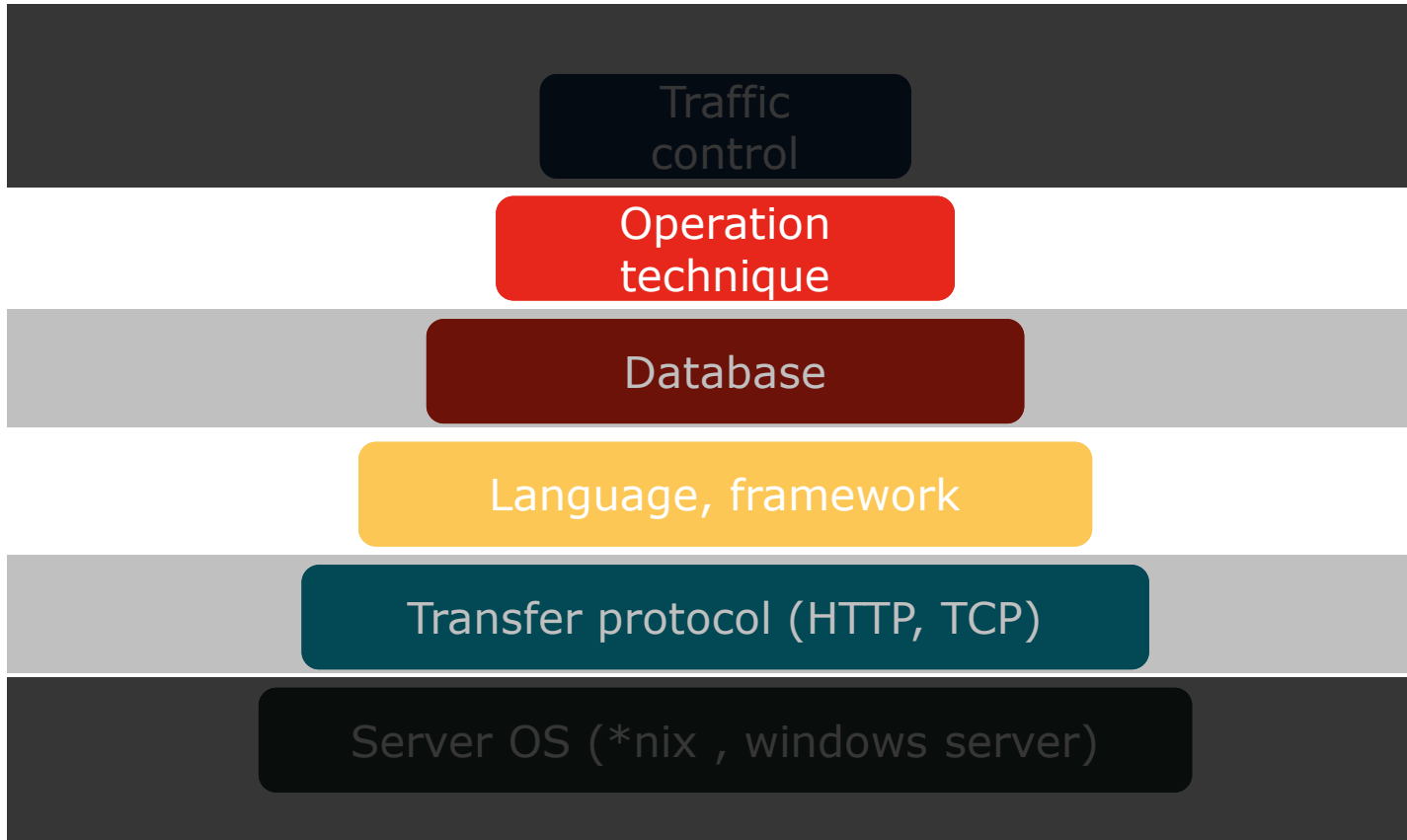


그러면  
IaaS  
로가자!

# [ IaaS(Infrastructure as a Service) ]

: PaaS에서 관리 및 분석 도구를 제거한 System 환경을 제공





## [ IaaS(Infrastructure as a Service) ]

: PaaS에서 관리 및 분석 도구를 제거한 System 환경만을 제공

월 10억번 호출 당 가격:

**₩1,767,193 (\$1550.17)**

백칠십육만칠천백구십삼원

기준 환율 1140

- \* DAU 10만 기준, 유저당 300 Request
- \* AWS EC2, Seoul region 기준



# [ IaaS(Infrastructure as a Service) ]

: PaaS에서 관리 및 분석 도구를 제거한 System 환경을 제공

월 10억번 호출 당 가격:  
**₩1,767,193 (\$1550.17)**  
백칠십육만칠천백구십삼원  
기준 환율 1140

덜 비싼데  
좀 불편하다

- \* DAU 10만 기준, 유저당 300 Request
- \* AWS EC2, Seoul region 기준

## [ Serverless ]

: server infrastructure에 대한 고민 없이 구현만 요구되는 architecture



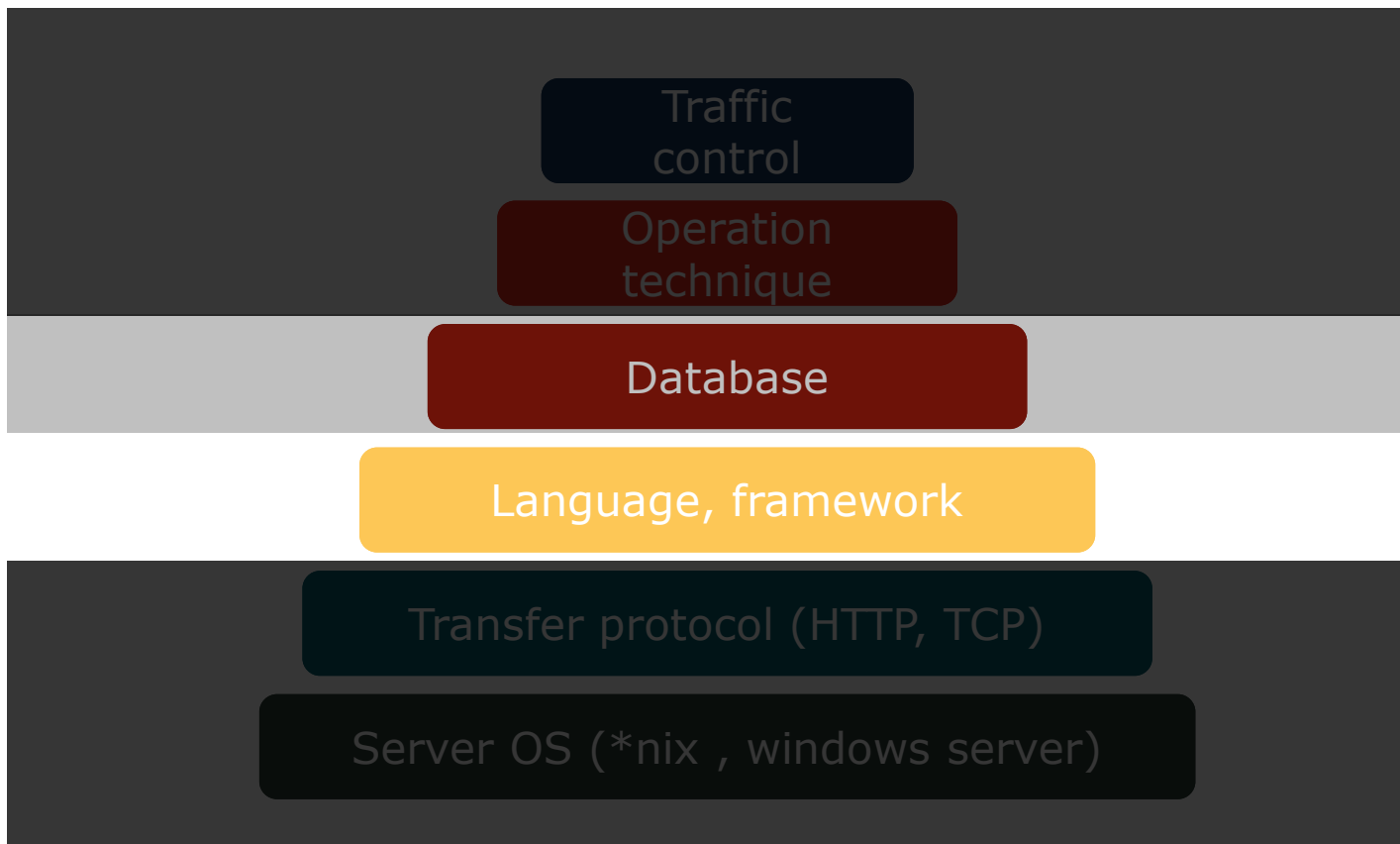
Google Cloud Function



IBM Openwhisk



Microsoft Azure function





## [ Serverless ]

: server infrastructure에 대한 고민 없이 구현만 요구되는 architecture

월 10억번 호출 당 가격:

**₩2,120,548 (\$1860.13)**

이백십이만오백사십팔원

기준 환율 1140

- \* DAU 10만 기준, 유저당 300 Request
- \* AWS Lambda, Seoul region 기준



# [ Serverless ]

: server infrastructure에 대한 고민 없이 구현만 요구되는 architecture

편하고  
월 10억번 호출 당 가격:  
**₩2,120,548 (\$1550.13)**  
이백십이만오백사십팔원  
기준 환율 1140

- \* DAU 10만 기준, 유저당 300 Request
- \* AWS Lambda, Seoul region 기준



## [ Serverless ]

: server infrastructure에 대한 고민 없이 구현만 요구되는 architecture

참 좋은데..  
월 10억번 호출 당 가격:  
**₩2,120,518 (\$1860.13)**  
더 쌀 수 있지 않나?  
이백십이만오백사십팔원  
기준 환율 1140

- \* DAU 10만 기준, 유저당 300 Request
- \* AWS Lambda, Seoul region 기준

## 현재 상용 Serverless들은 IaaS 위에 Serverless 기능을 추가

- ⇒ 기존 IaaS의 고비용 문제가 제거되지 않음
- ⇒ Serverless 기능을 위해 울며 겨자먹기 식으로 IaaS를 이용

현재 상용 Serverless들은 IaaS 위에 Serverless 기능을 추가

⇒ 기존 IaaS의 고비용 문제가 제거되지 않음

⇒ Serverless 기능을 위해 울며 겨자먹기 식으로 IaaS를 이용

그렇다면  
상용 IaaS를 이용하지 않는 방법은  
없을까?

Rune



만들자  
만들어서 쓰자

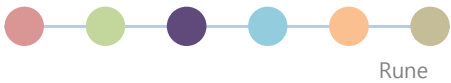


Private Cloud에 이용할 수 있는  
Opensource serverless platform을 해보자

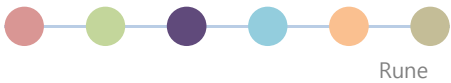


마법처럼 간단하게  
누구나 쓸 수 있도록 만들자





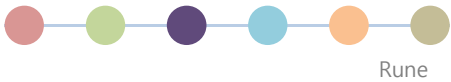
Machine만 있다면  
설치부터 운용까지 손쉽게 만들어보자



Python으로 이용 가능한  
Serverless framework

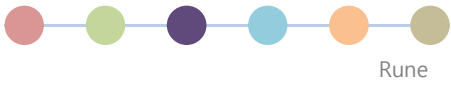


Openstack을 이용하여  
확장성과 유연함을 가질 수 있게

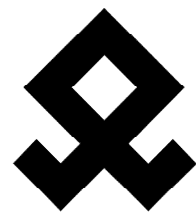
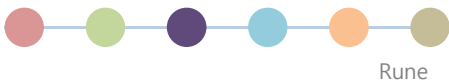


Django로 제공되는  
User dashboard



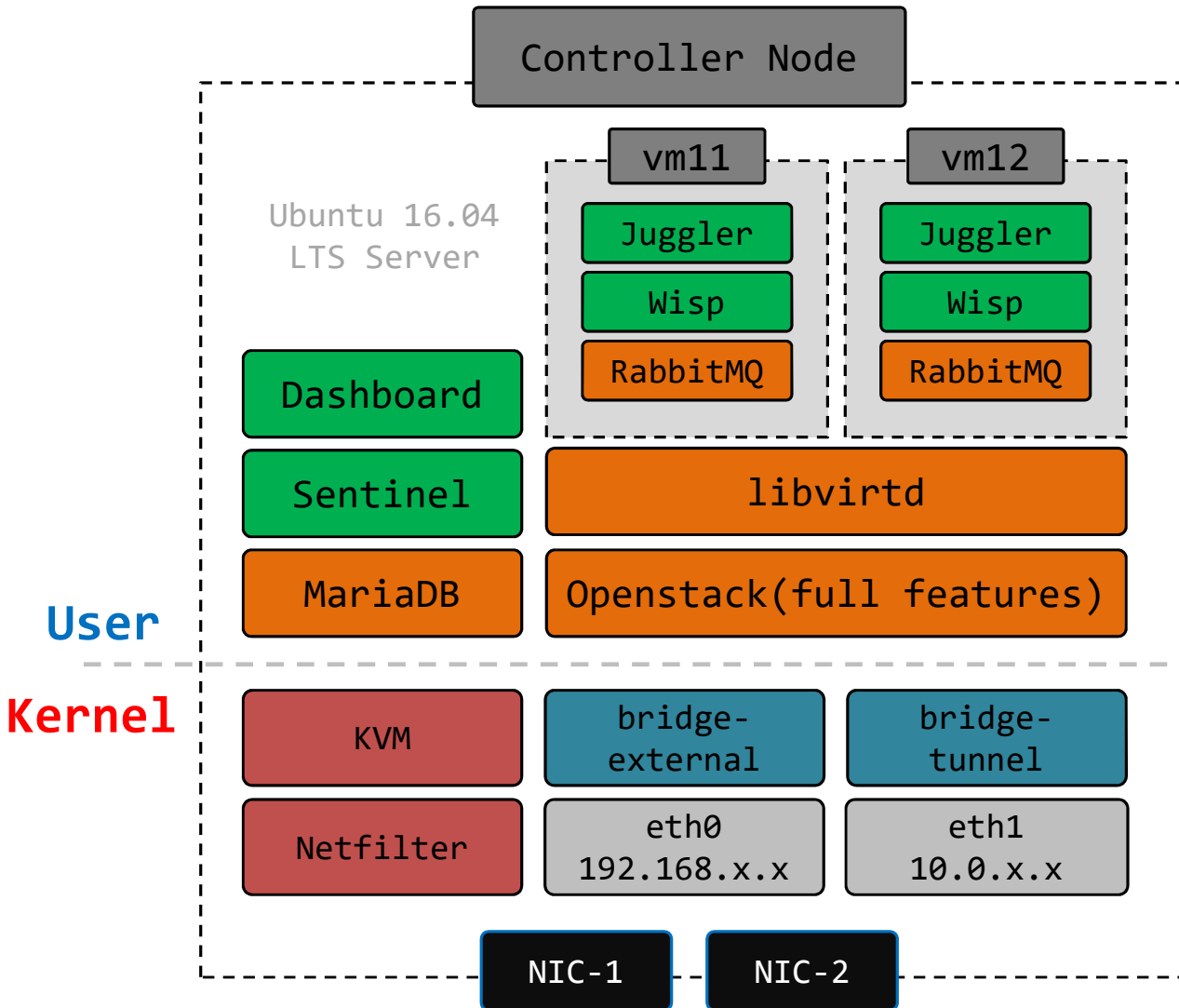


Rune



Rune

마법처럼  
편리하게

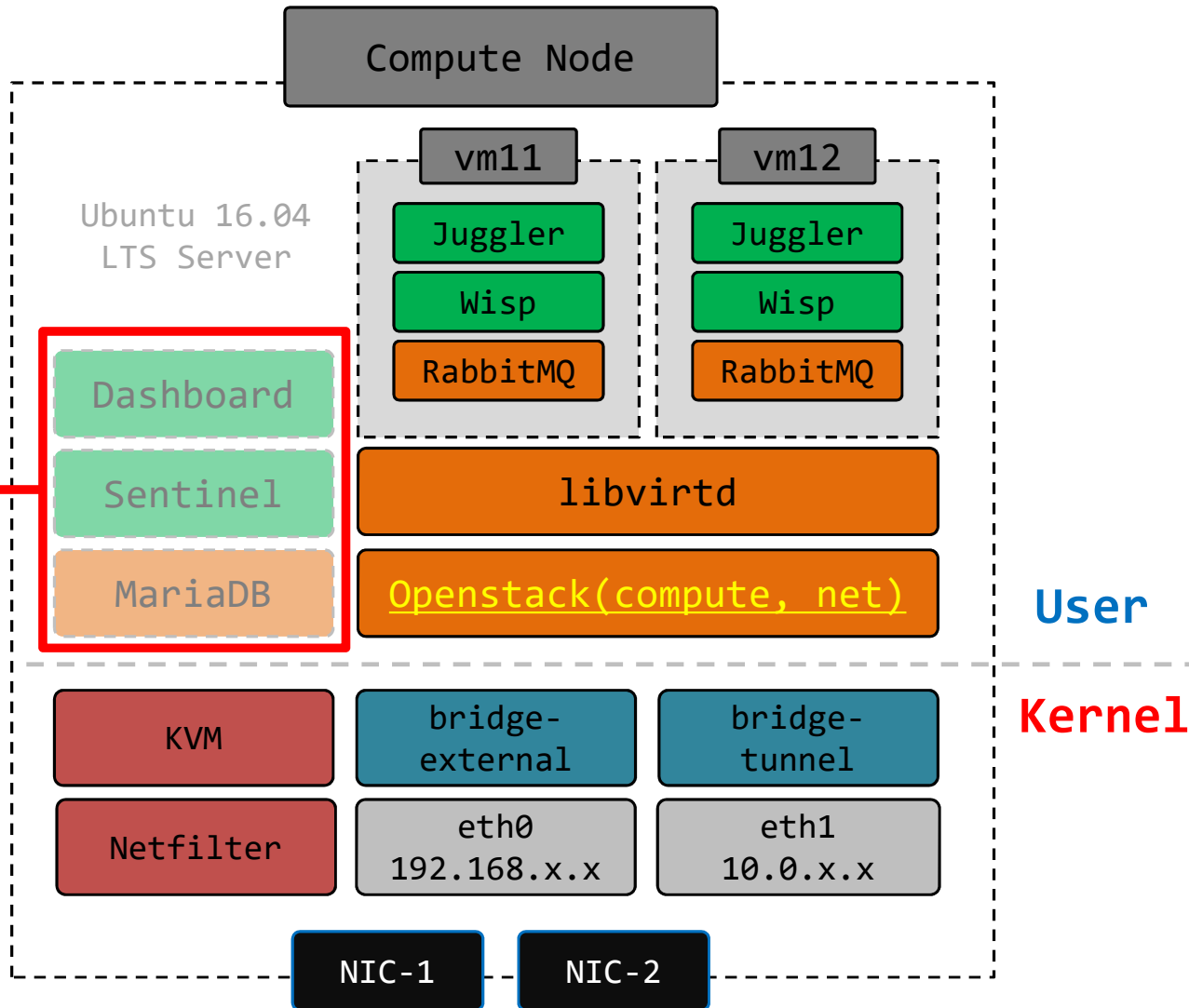


## Controller Node는 Openstack을 기반으로 Sentinel과 Dashboard를 실행

스크립트 1번 실행으로 Controller Node를 설치할 수 있으며  
Serverless 플랫폼을 제공 / 자체적으로 vm을 가질 수 있다.

# Compute Node는 최소한의 Openstack을 기반으로 Instance의 Scalability 제공

최소한의 Openstack 서비스를 설치하며  
스크립트 1번 실행으로 vm을 확장할 수 있다.





## Openstack(controller)

nova-api

neutron-  
server

nova-compute

neutron-  
dhcp-agent

nova-scheduler

neutron-  
l3-agent

nova-consoleauth

neutron-linux  
bridge-agent

nova-conductor

neutron-  
ml2-plugin

nova-novncproxy

neutron-  
metadata-agent

keystone

glance-api

glance-registry

## Openstack(compute)

nova-compute

neutron-linux  
bridge-agent

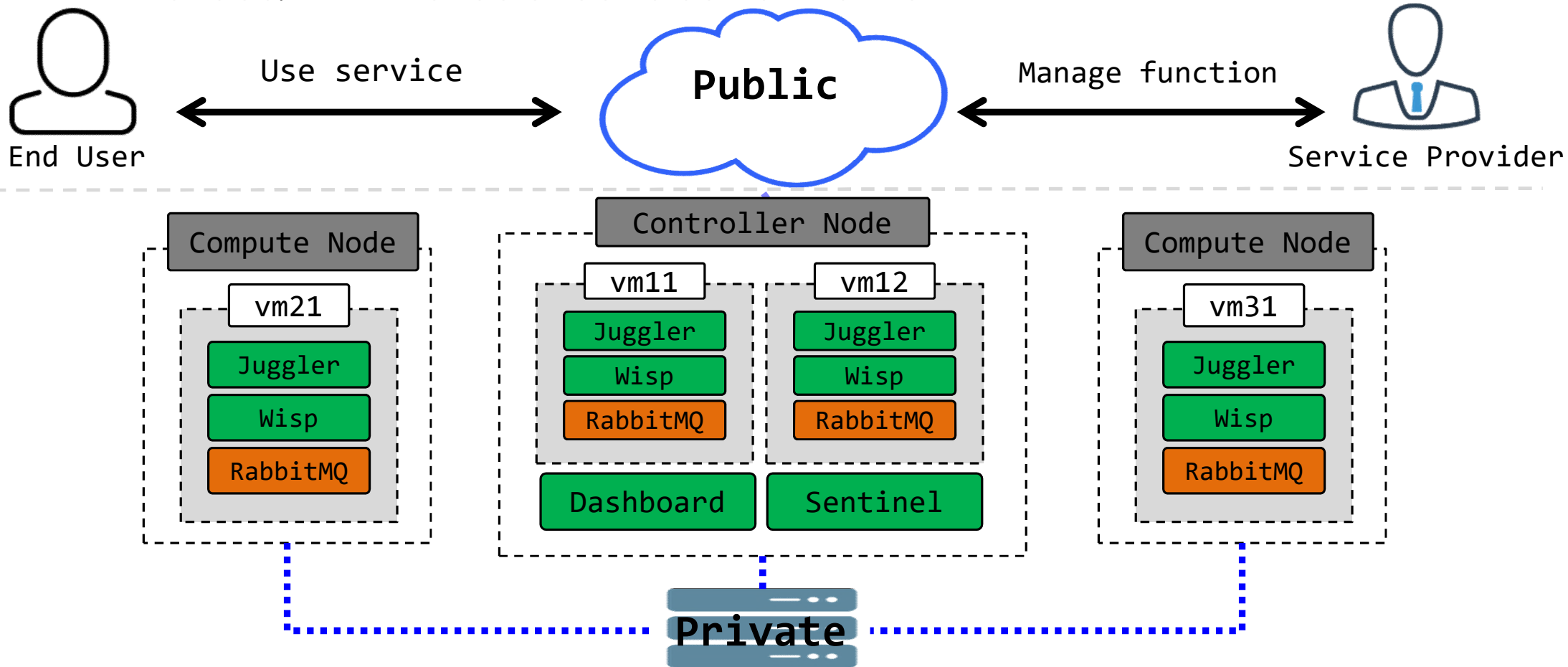
인증은 **Keystone** 이미지는 **Glance**  
컴퓨팅은 **Nova** 네트워크는 **Neutron**

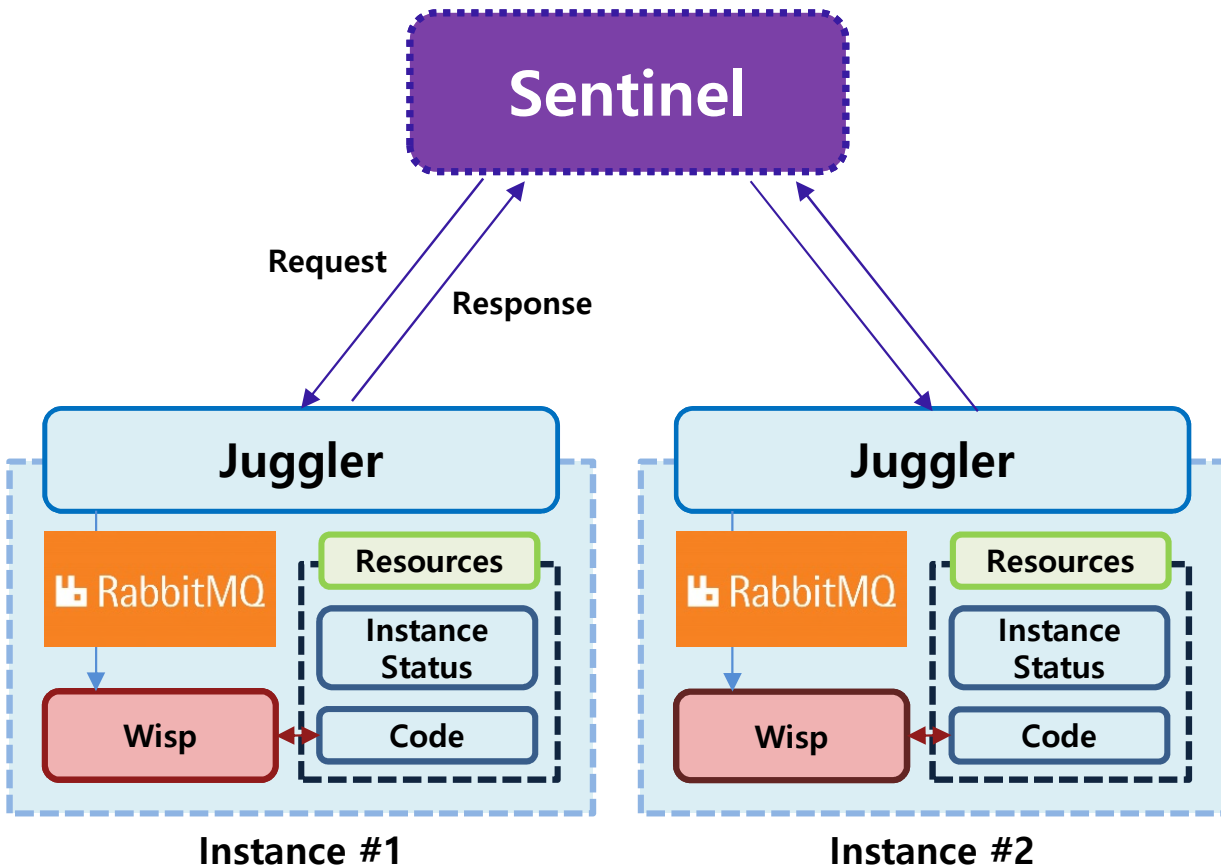
Controller Node는 Openstack의 필수적인 모든 패키지를 설치하고  
Compute Node는 확장에 필요한 최소한의 패키지만 설치한다.

Rune Project에서는 이 모든걸 스크립트 하나로 해결한다.

## 개발자에게는 핵심 로직에 집중할 수 있는 환경을 사용자에게는 끊임 없는 서비스를 제공

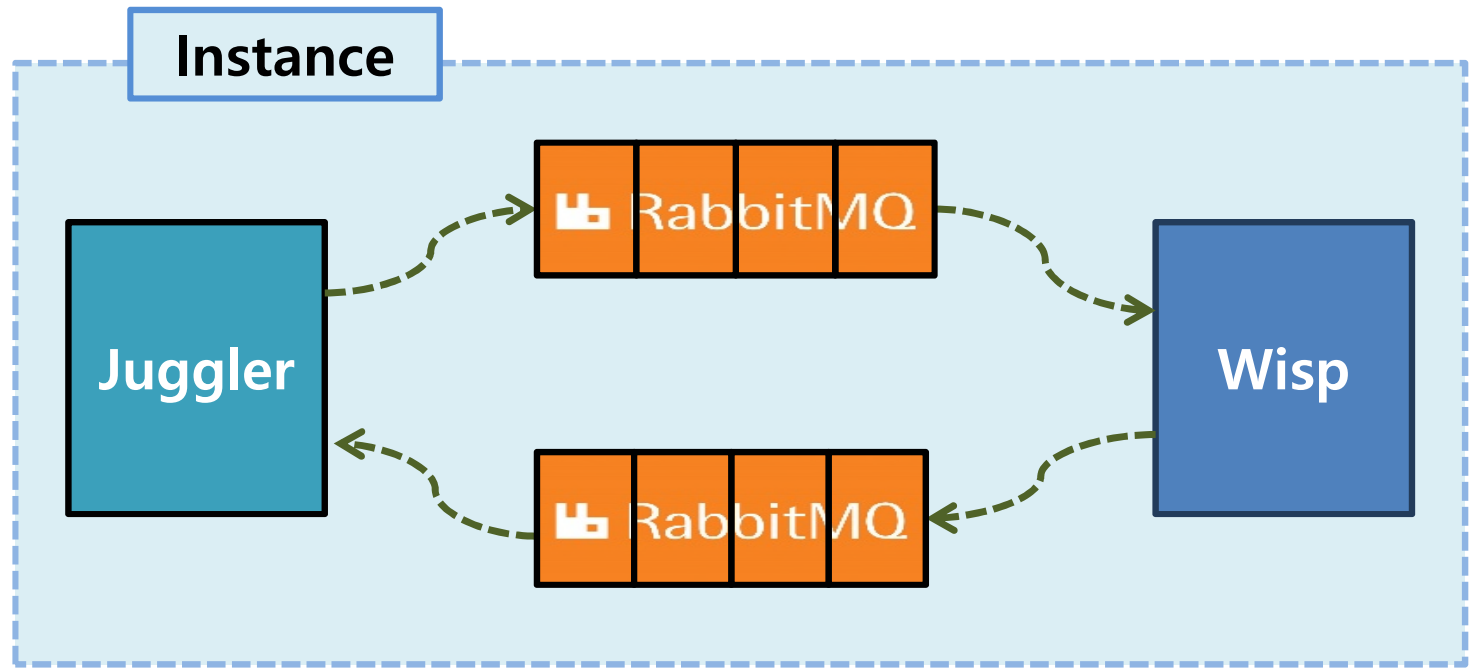
Dashboard는 개발자에게 / Sentinel은 개발자와 사용자와 모두에게 편리함을 제공한다.





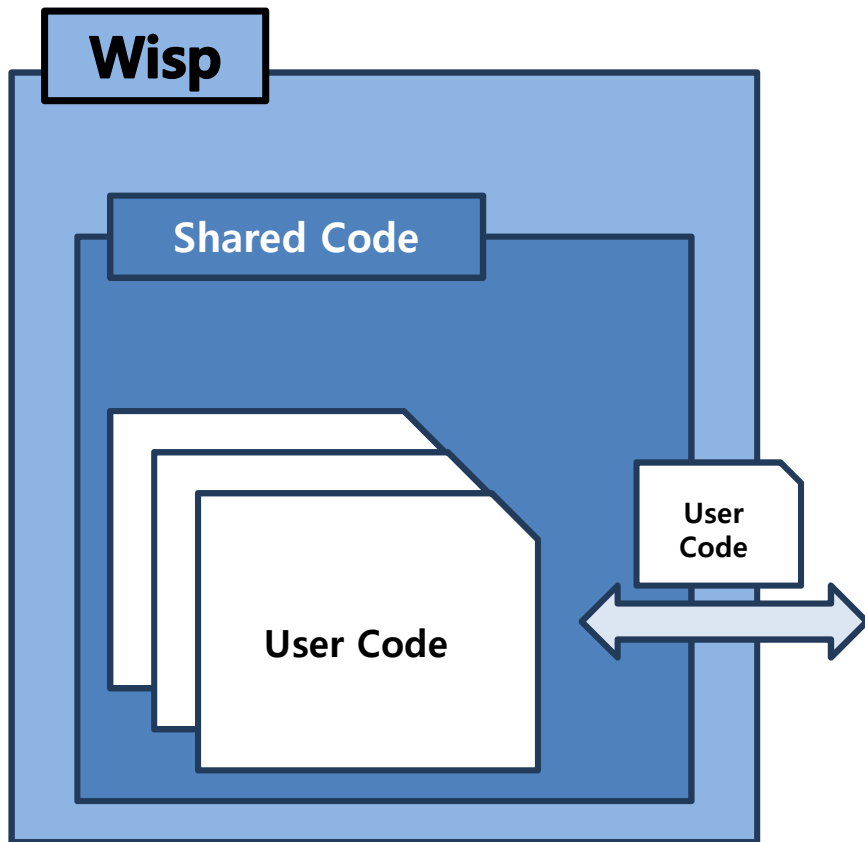
## Sentinel은 Juggler를 통해 Instance에서 Function을 실행

Juggler는 Instance의 자원과 Function Code를 관리  
Sentinel의 요청 시 Wisp을 이용하여 Code를 실행한다.



## Wisp은 MQ를 통해 Juggler의 요청을 비동기적으로 처리

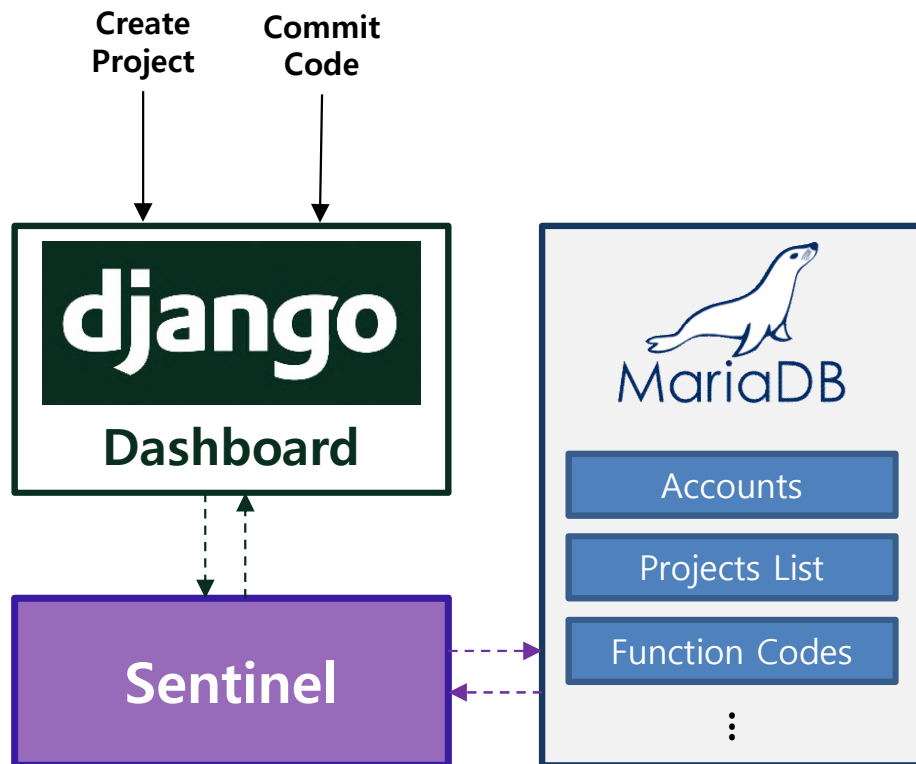
Wisp은 Juggler Message Queue를 통해 요청한 파라미터와 Code를 실행하여 그 결과를 Message Queue로 다시 전달한다.



속도향상을 위해..

- 공용코드는 메모리 상주
- User 코드는 MRU + 시간정책으로 관리
  - BUT! 코드 업데이트 요청시 강제로 Code Reload 진행.





## User의 Project와 Source 관리를 돕는 Rune Dashboard

Dashboard는 Sentinel API에 대한 User Interface 를 제공합니다.



## Dash board

- 로그인 및 회원가입 페이지

### Rune dashboard

#### Sign in

☐ Remember me

Sign in

Sign Up

### Rune dashboard

#### Sign Up

Sign up



## Dash board

### - 프로젝트 리스트 페이지

Rune project list instance list download logout				
PROJECT LIST				
ID	userid	user name	name	
36	11	wonggss@naver.com	rune	remove
37	11	wonggss@naver.com	test_project	remove
			<input type="text"/>	add





# Dash board

- 인스턴스 리스트 페이지

Rune	project list	instance list	download	logout
INSTANCE LIST				
UUID	address	core	memory	storage

## Dash board

- 코드 리스트 페이지
- 코드 추가 페이지

Runeproject listinstance listdownloadlogout

### CODE LIST

ID	name	last update time	
76	run.py	Nov. 10, 2016, 10:33 p.m.	remove
77	form.html	Nov. 10, 2016, 10:23 p.m.	remove
			add

Runeproject listinstance listdownloadlogout

### Add code

title

code

add



개발 항목 Rune

Rune project list instance list download logout

## Update code

title

run.py

code

```
#!/usr/bin/env python3
import sys, os, socket
from socketserver import ThreadingMixin
from http.server import SimpleHTTPRequestHandler, HTTPServer
from sentinelhttphandler import *
```

```
HOST = socket.gethostname()
```

```
class ThreadingSimpleServer(ThreadingMixin, HTTPServer):
```

update

```
#!/usr/bin/env python3
import sys, os, socket
from socketserver import ThreadingMixin
from http.server import SimpleHTTPRequestHandler, HTTPServer
from sentinelhttphandler import *
```

```
HOST = socket.gethostname()
```

```
class ThreadingSimpleServer(ThreadingMixin, HTTPServer):
```

## Dash board

-코드 수정 페이지

-Prism.js를 사용하여 코드 하이라이팅 구현



# 개발 사항 영상 및 시연

---

Debian 계열의 Linux 뿐만이 아니라 Cent, Windows 등의 여러가지 OS를 설치하여 사용할 수 있게 제공

02

Various O/S support

Dashboard improve

01

Dashboard 에 서 User, Project, Code 만을 관리하는 게 아닌 Instance에 대한 설정을 상세히 할 수 있게

현재 구현된 단순한 Load balancing 이 아닌 물리적 Load balancing 장비와 유사한 Load balancing 구현. Haproxy 도입 혹은 유사한 성능 제공

Load balancing Algorithm

03

04

Database dependency remove

다양한 형태의 Database를 사용할 수 있게 하여 DB 종속성을 제거하고 필요한 License만 구매하여 DB이용

# 감사합니다

---

Thank you  
děkuji vám  
Благодаря  
Дзякуй  
dankie

kiitos  
谢谢  
trugarez  
gracias  
شكراً  
Shenorhagal em

tak  
gràcies  
hvala  
eskerrik asko  
faleminderit

#1

**자체 Private cloud를 구축할 경우  
10억 Request에 대해서  
2 ~ 3대의 서버로 감당할 수 있습니다.**

**평균 회선 비용과 서버 구입/임대료 감안시  
월 150만원 정도의 비용으로 운영 가능**