



# 쿠버네티스 환경에서 오픈소스 기반의 서버리스와 메시지 큐의 구현

강상진

Akamai Technologies  
APJ Computing Specialist



# 오늘의 주제

## 그리고.. 쿠버네티스

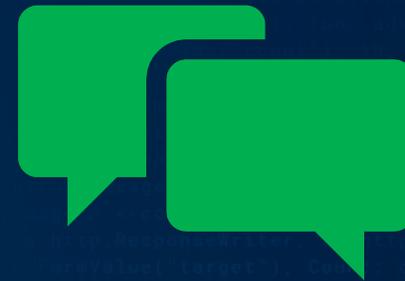
### 서버리스

서버리스 컴퓨팅, 종종 "서버리스"로 간단히 언급되며, 개발자가 기반 인프라를 관리하거나 걱정하지 않고 애플리케이션을 구축하고 실행할 수 있는 클라우드 컴퓨팅 패러다임



### 메시지 큐

메시지를 비동기적으로 보내고 받을 수 있도록 하여 다양한 컴퓨터 프로세스 또는 컴퓨터 시스템 간에 데이터나 메시지를 안전하게 전달하고 통신할 수 있도록 돕는 기술적인 메커니즘



서버리스

“서버리스는  
아키텍처 패턴이다.”

# 서버리스



모놀리딕



마이크로서비스



함수

줄어드는 인프라스트럭처에 대한 고민

# 서버리스

## 운영 관점



불필요한  
인프라 관리



관리형  
보안



사용한 만큼  
과금

## 개발 관점



서비스 기반



이벤트 기반



Stateless

# 함수 (Function)



- 짧은 생명 주기
- 하나의 목적
- 무상태 기반
- 오토 스케일링
- 다양한 개발 언어 지원

# 서버리스 + 함수

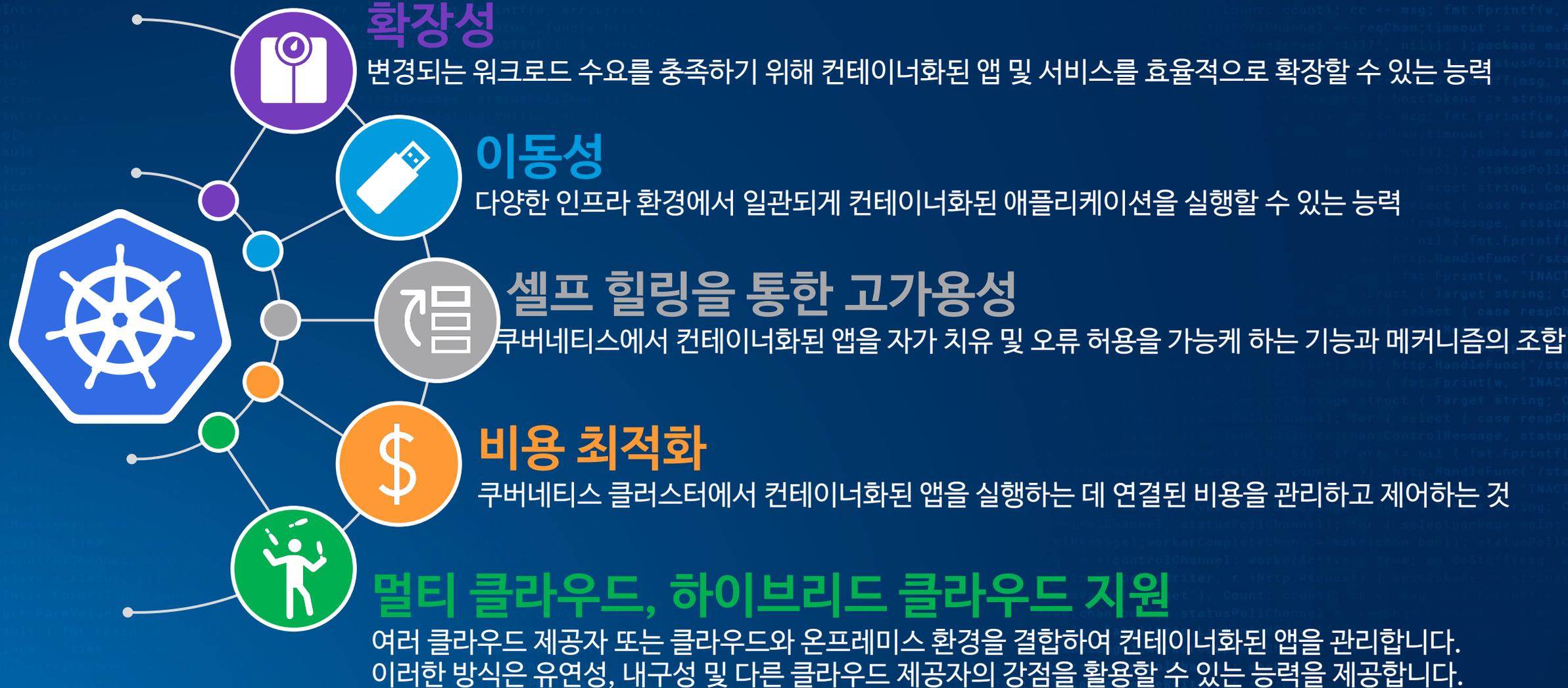
↑ 인프라 대신 워크로드 자체에 집중



- 빠른 시작 시간
- 더 나은 자원 활용
- 더 세밀한 관리
- 모놀리식 분할

→ 더 자동화되고 유연한 인프라 구현

# ..그리고 쿠버네티스!





## LKE (Linode Kubernetes Engine)

빠르고 간편한 Kubernetes 클러스터 배포  
완전 관리형 컨테이너 오케스트레이션 엔진으로  
쿠버네티스 클러스터 배포

# LKE (Linode Kubernetes Engine)

The screenshot shows the CNCF certification page for software conformance. The page lists several providers, each with their logo, product name, and company name. A red arrow points to the Linode logo.

Provider	Product	Company	Additional Info
IBM	IBM Kubernetes Engine	IBM	
Inspur Group	Inspur Kubernetes Engine	Inspur Group	
Linode	Linode Kubernetes Engine	Linode	
Microsoft	Microsoft AKS Engine for Azure Stack	Microsoft	★ 6, MCap: \$2.1T
NHN Cloud	NHN Kubernetes Service (NKS)	NHN Cloud	Funding: \$121.7M
JD Cloud	JD Cloud Kubernetes Service	JD Cloud	
NIFCLOUD	NIFCLOUD Kubernetes Service Hatoba	Fujitsu	MCap: \$25.7B
Nutanix	Nutanix Karbon	Nutanix	MCap: \$5.7B
Oracle	Oracle Container Engine	Oracle	MCap: \$256B
Orka	Orka	MacStadium	Funding: \$65.7M
OVHcloud	OVH Managed Kubernetes Service	OVHcloud	MCap: \$2.1B

# LKE (Linode Kubernetes Engine)



## 극대화된 리소스 효율성

클라우드 리소스를 최대한 자동으로 활용하여 클러스터 전체에 리소스를 효율적으로 활용하며 컨테이너를 원활하게 배포

요구사항에 맞게 클러스터 자동 확장  
Horizontal Cluster Autoscaling으로 설정된  
한도에 따라 실시간으로 리소스를 생성하고 폐기.  
노드 풀을 효율적으로 관리함으로써 가용성과  
애플리케이션 안정성을 추구

## 관리 요금 없음

컴퓨팅 인스턴스, NodeBalancers, Block Storage Volumes에 대한  
비용만 지불. 기타 클라우드 사업자와 달리 관리형 쿠버네티스 컨트롤 플  
레인의 경우 관리비를 청구하지 않음  
*\*타 CSP의 경우 시간당 \$0.10 또는 월 \$73의 클러스터 관리 수수료를 부과*



## 개발자 친화적, 높은 이식성

API 를 통해 직접 LKE 클러스터를 생성하고 관리해 쿠버네티스 관리를 자  
체 인프라와 통합. 기존 애플리케이션과 워크로드를 빠르게 이전

Helm 차트, Operators, Controllers로 배포  
쿠버네티스 오픈 소스 생태계의 도구를 활용.  
Rancher, Helm, Operators 등 인기 있는  
쿠버네티스 관련 도구와의 통합을 지원

# LKE (Linode Kubernetes Engine)

## Akamai Connected Cloud의 쿠버네티스 클러스터를 사용한 클라우드 비용 효율화

**Region**  
You can use our [speedtest page](#) to find the best region for your current location.  
Singapore, SG (ap-south)

**Kubernetes Version**  
1.26

**HA Control Plane**  
Recommended for production workloads, a high availability (HA) control plane is replicated on multiple master nodes to 99.99% uptime. [Learn more about the HA control plane.](#)

Yes, enable HA control plane. (\$60.00/month)  
 No

**Add Node Pools**  
Add groups of Linodes to your cluster. You can have a maximum of 100 Linodes per node pool.

**Dedicated CPU** | Shared CPU | High Memory | Premium CPU

Dedicated CPU instances are good for full-duty workloads where consistent performance is important.

Plan	Monthly	Hourly	RAM	CPUs	Storage				
Dedicated 4 GB	\$36	\$0.054	4 GB	2	80 GB	-	0	+	Add
Dedicated 8 GB	\$72	\$0.108	8 GB	4	160 GB	-	3	+	Add
Dedicated 16 GB	\$144	\$0.216	16 GB	8	320 GB	-	3	+	Add
Dedicated 32 GB	\$288	\$0.432	32 GB	16	640 GB	-	3	+	Add

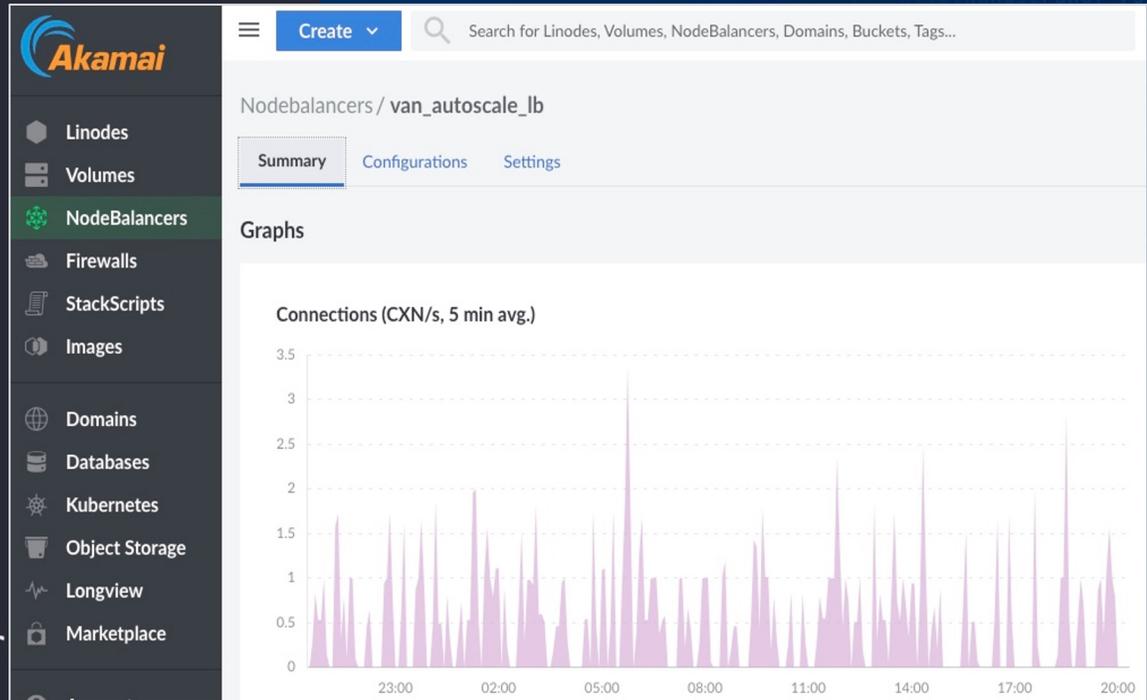
**Cluster Summary**  
Dedicated 4 GB Plan  
2 CPUs, 80 GB Storage  
- 3 +  
\$108.00/month  
\$108.00/mo  
**Create Cluster**

# LKE (Linode Kubernetes Engine)

로드밸런서 형태의 외부 접속을 위한 서비스 리소스를 제공

```
~> kubectl get service
NAME                TYPE          CLUSTER-IP      EXTERNAL-IP      PORT(S)          AGE
hello1-service-lb  LoadBalancer 10.128.207.202  172.104.37.252  80:31437/TCP     18s
kubernetes          ClusterIP     10.128.0.1      <none>           443/TCP          27h

~> kubectl describe service hello1-service-lb
Name:                hello1-service-lb
Namespace:           default
Labels:              <none>
Annotations:         <none>
Selector:            app=hello-one
Type:                LoadBalancer
IP Family Policy:    SingleStack
IP Families:         IPv4
IP:                  10.128.207.202
IPs:                 10.128.207.202
LoadBalancer Ingress: 172.104.37.252
Port:                <unset> 80/TCP
TargetPort:         80/TCP
NodePort:            <unset> 31437/TCP
Endpoints:          <none>
Session Affinity:   None
External Traffic Policy: Cluster
Events:
  Type    Reason              Age   From              Message
  ----    -
  Normal  EnsuringLoadBalancer 65s   service-controller Ensuring load balancer
  Normal  EnsuredLoadBalancer 64s   service-controller Ensured load balancer
```



# LKE (Linode Kubernetes Engine)

블록 스토리지 형태의 지속적인 볼륨 리소스를 제공

```
~ kubectl get persistentvolume
NAME                CAPACITY  ACCESS MODES  RECLAIM POLICY  STATUS  CLAIM                STORAGECLASS          REASON  AGE
pvc-2e260ccc156a4674  10Gi      RWO           Retain          Released  default/pvc-example  linode-block-storage-retain  27h
pvc-3cba285159044a61  10Gi      RWO           Retain          Released  default/pvc-example7  linode-block-storage-retain  26h
pvc-50a5b14498e54fde  10Gi      RWO           Retain          Released  default/pvc-examplei4  linode-block-storage-retain  26h
pvc-7e537458b25b41bd  10Gi      RWO           Retain          Bound    default/pvc-example7  linode-block-storage-retain  73s
```

Volumes Docs Create Volume

Label	Status	Region	Size	Attached To	
temp-47345846	Active	Tokyo, JP	30 GB	Windows_VM_StackScripts2	Show Config Edit ...
temp-47338202	Active	Tokyo, JP	30 GB	windows-ap-northeast	Show Config Edit ...
pvc7e537458b25b41bd	Active	Singapore, SG	10 GB	<u>lke116964-173565-649e8fabd85b</u>	Show Config Edit ...

# 쿠버네티스 기반의 서버리스 오픈소스



nuclio



fission

# Knative

## Implement 서버리스 on 컨테이너



현대적인 서버리스 워크로드를 구축, 배포 및 관리하는 쿠버네티스 기반 오픈소스

# Knative

Products

Google  
Cloud Run

Pivotal  
Function  
Services

SAP Kyma

T-Mobile  
Jazz

OpenFaas

Primitives

Build

Serving

Eventing

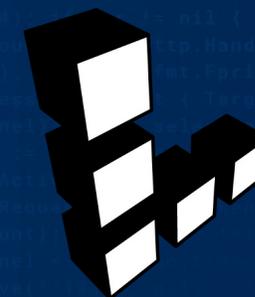
Gateway



Platform



kubernetes



# OpenFaaS



## Anywhere

OCI 이미지를 사용하여 온프레미스 또는 클라우드에 함수를 배포하세요.



## Any code

어떤 언어로든 함수를 작성하고 기존의 마이크로서비스도 함께 가져오세요.

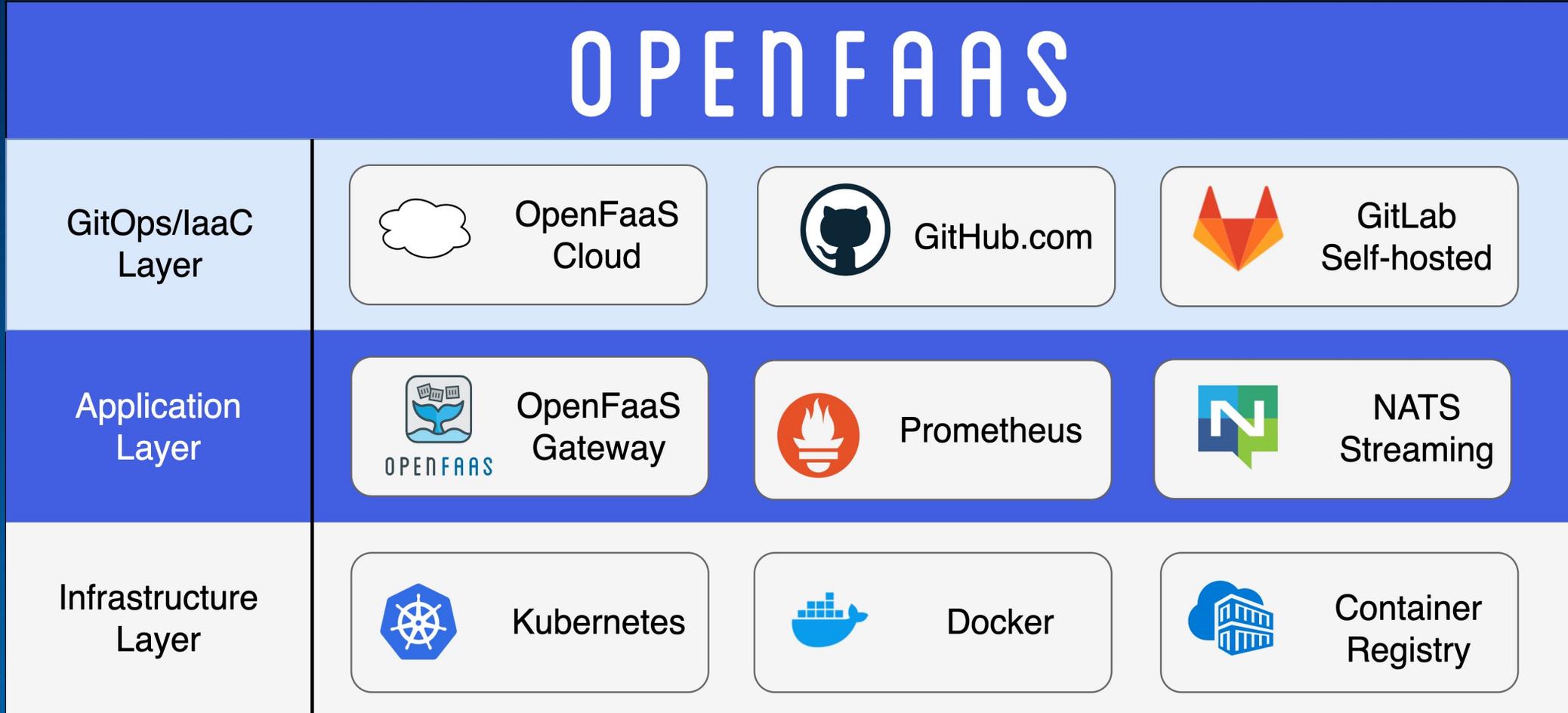


## Any scale

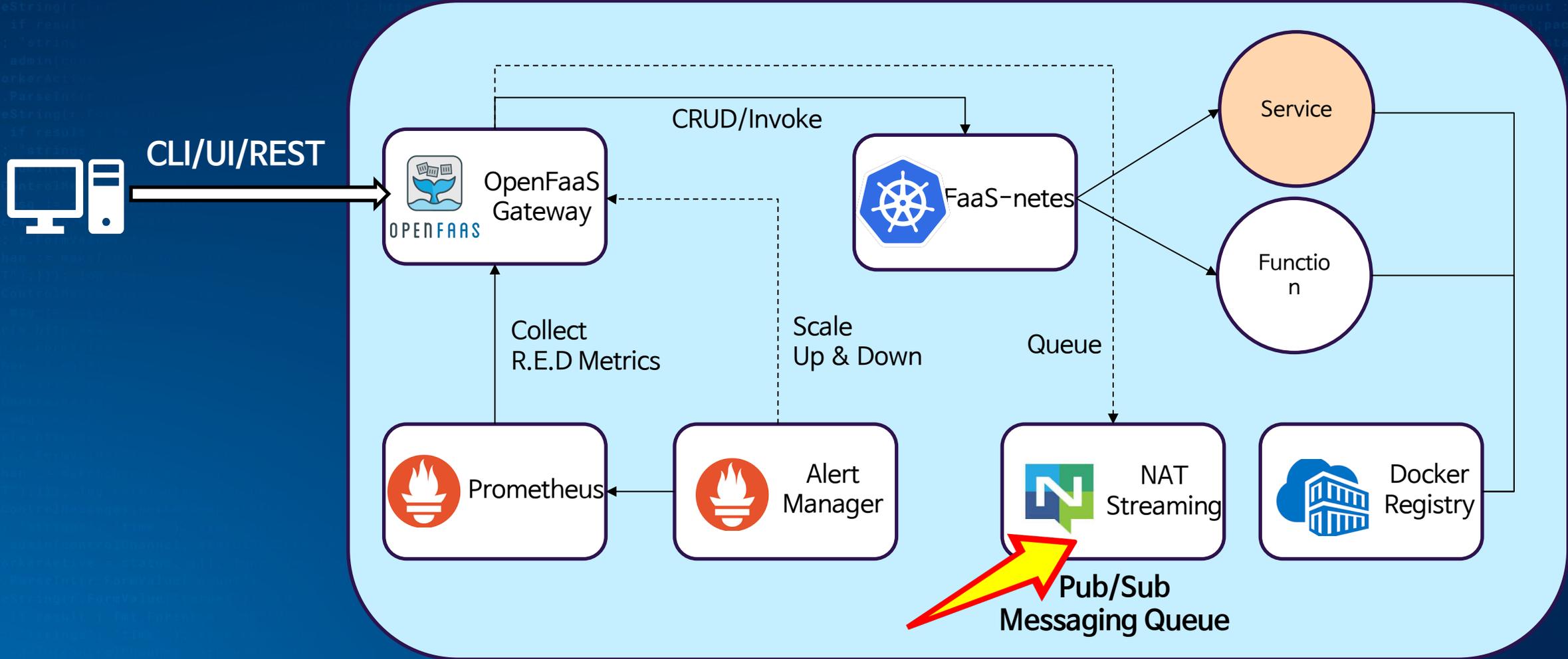
전문 기능을 사용하여 함수를 수요에 맞게 확장하고 유휴 상태일 때는 제로로 축소합니다.



# OpenFaaS 아키텍처



# OpenFaaS 아키텍처



# OpenFaaS 함수 제작

1. `$ faas new --lang python3 http-status`

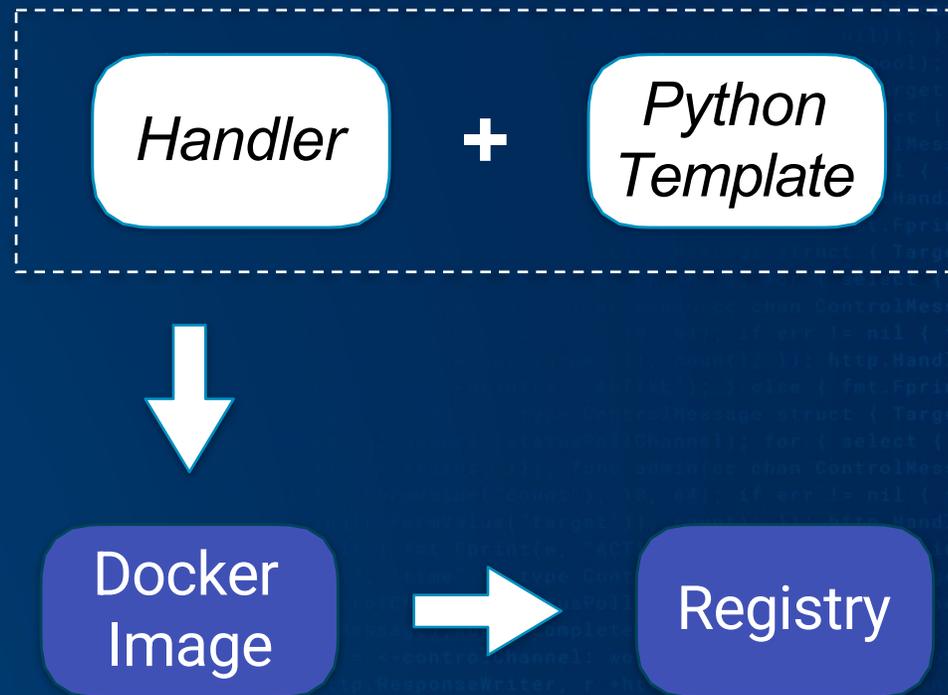
```
def handler(req):  
    r = requests.get(req)  
    return str(r.status_code)
```

handler.py

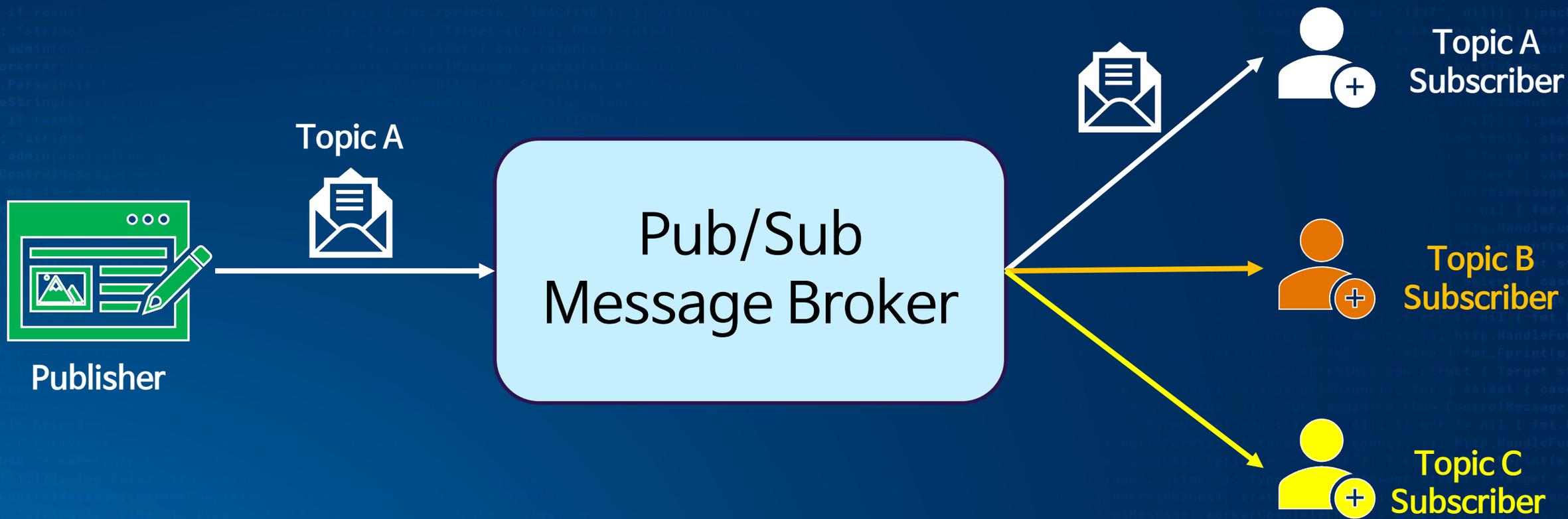
requests

requirements.txt

2. `$ faas up`



# Pub/Sub 메시징



# Pub/Sub 메시징

- 개발 관심사의 분리
- 애플리케이션 확장성
- 메시징 신뢰성
- 이벤트-주도 개발
- 브로드캐스팅 방식의 메시지 배포

Pub/Sub 메시징의 일반적인 구현에는 메시지 큐, 메시지 브로커 및 발행/구독 시스템이 포함됩니다.

Pub/Sub 메시징 시스템의 잘 알려진 예시로는 Apache Kafka, RabbitMQ, ActiveMQ 및 MQTT (Message Queuing Telemetry Transport)가 있습니다.

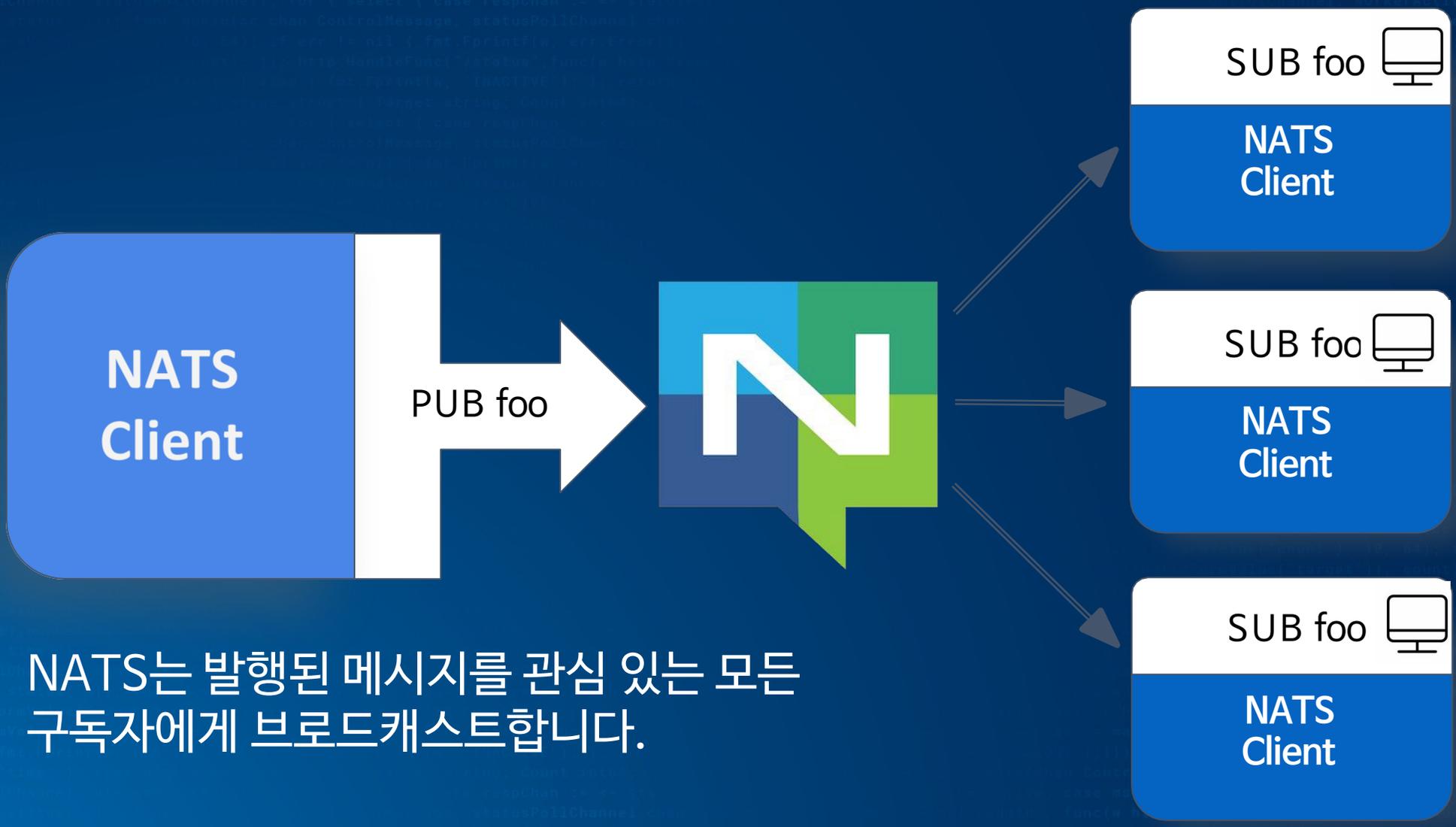
# 쿠버네티스 기반의 메시징 오픈소스



KubeMQ



# NATS 메시징



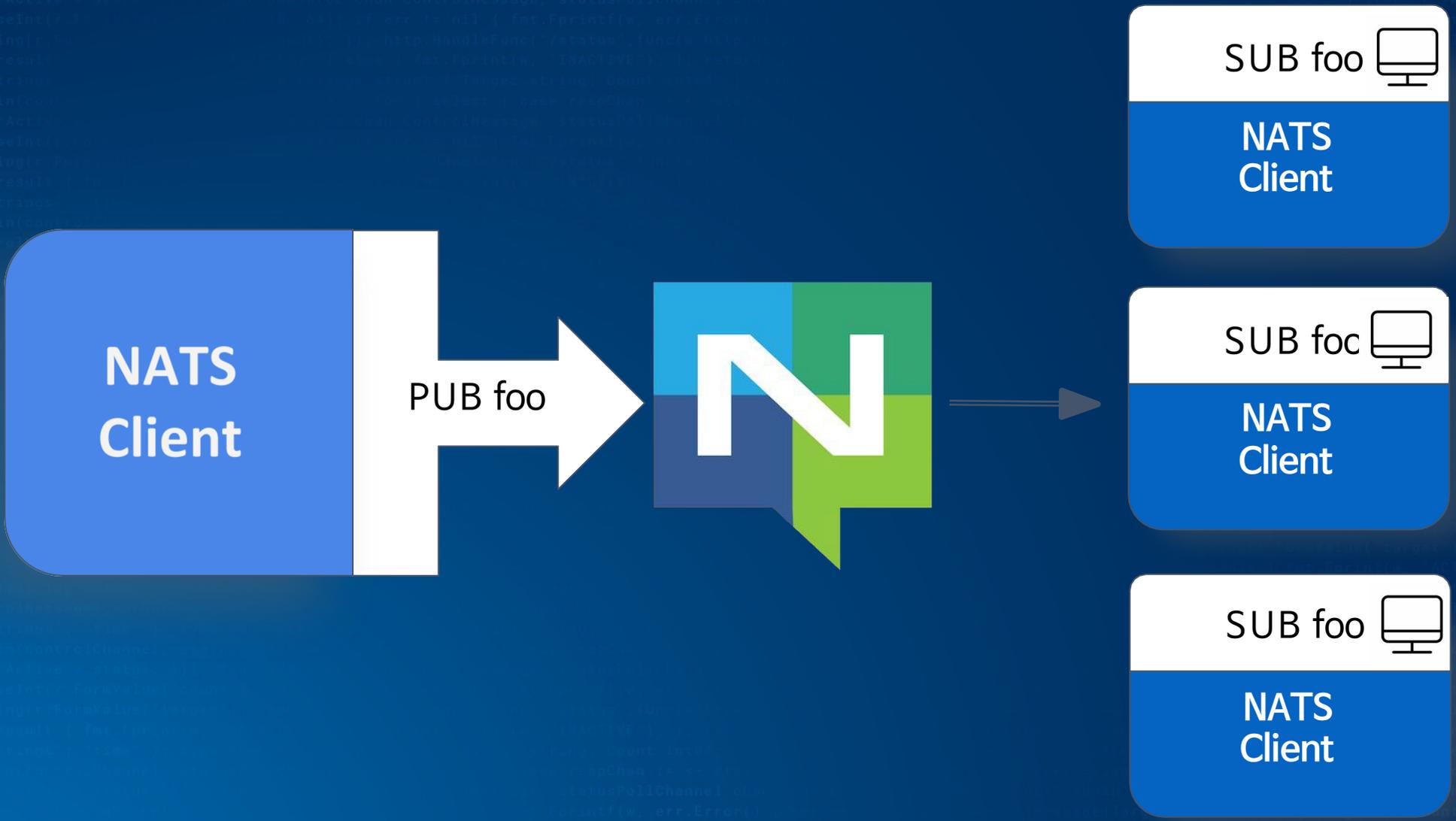
NATS는 발행된 메시지를 관심 있는 모든 구독자에게 브로드캐스트합니다.

# NATS 요청 & 응답 패턴

고유한 응답 주제를 사용하여 클라이언트는 요청한 사람에게만 응답하는 서비스에 요청을 보낼 수 있으며, 이로써 1대1 관계를 생성합니다.



# NATS 메시징 큐

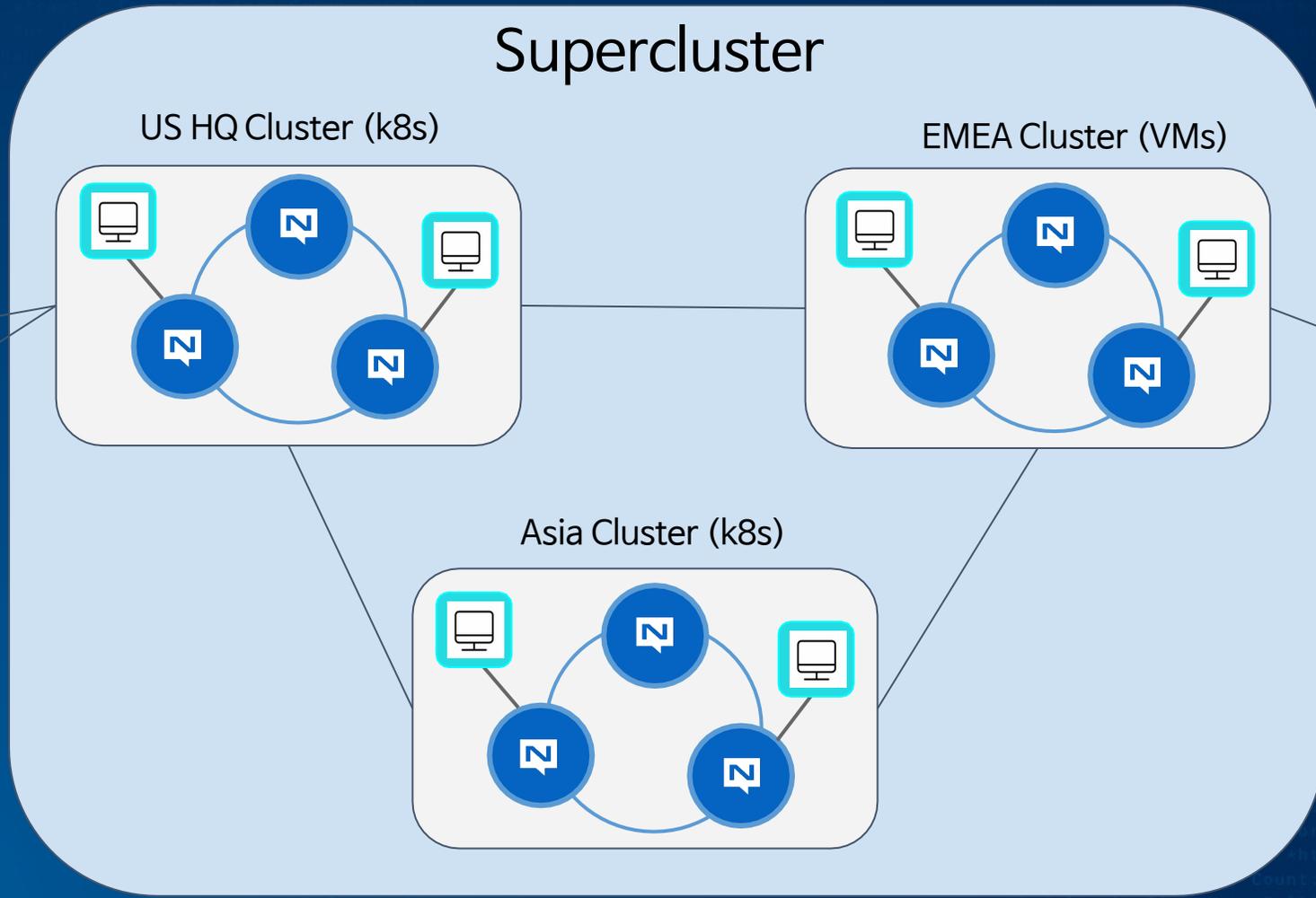


임의로  
분산된  
SUB foo  
작업자

# NATS 글로벌 배포

-  Devices
-  NATS Server

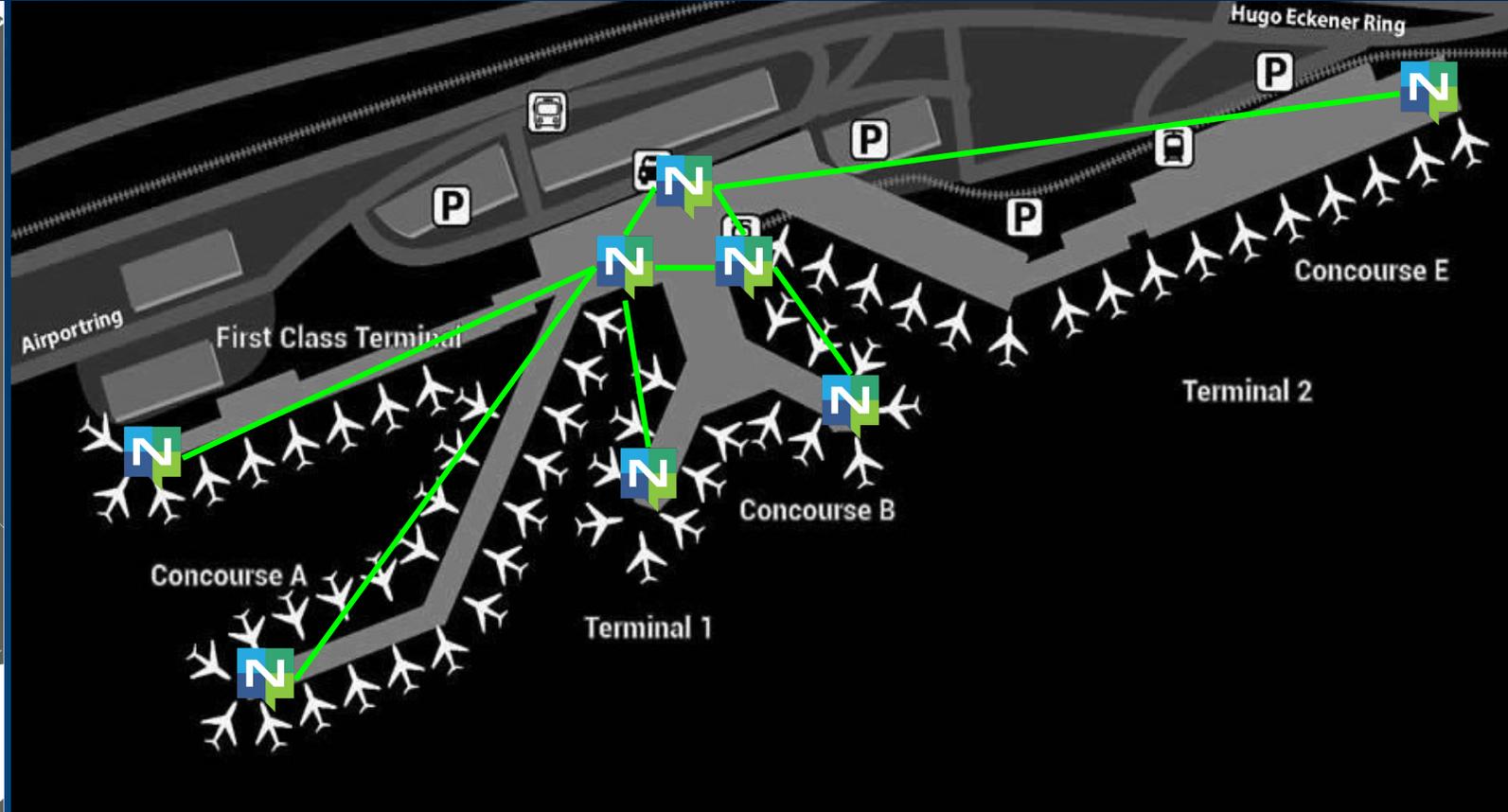
LeafNode in a Remote Cluster



LeafNode



# NATS 글로벌 배포 예제



공항이 연결되어 리드 노드를 가진 슈퍼클러스터가 형성되고, 게이트 메시징 통신은 비행기까지 확장됩니다.

# 관리형 클라우드 제품과 오픈소스 비교

	관리형 서비스	관리형 메시지 큐	LKE와 오픈소스
제품	Lambda, Cloud Functions, Serverless Function, ...	SNS, Cloud Pub/Sub, Service Bus, Event Hubs, ...	OpenFaaS, Knative, NATS, Kafka, ..
비용 체계	<ul style="list-style-type: none"> <li>호출 횟수</li> <li>실행 시간</li> <li>할당된 메모리</li> <li>외부 네트워크 전송 데이터양</li> </ul>	<ul style="list-style-type: none"> <li>게시 요청 수</li> <li>메시지 전송</li> <li>주제 구독</li> <li>메시지 필터링</li> <li>추가 기능</li> </ul>	<ul style="list-style-type: none"> <li>워커 노드 VM 비용 only</li> <li><b>단순하고 예상 가능한 비용</b></li> </ul>
고민?	<ul style="list-style-type: none"> <li>느린 스타트 (cold start)</li> <li>제한된 실행 시간</li> <li>자원 제한</li> <li><b>벤더 종속성</b></li> <li><b>복잡한 비용 체계</b></li> </ul>	<ul style="list-style-type: none"> <li>제한된 메시지 보관 기간</li> <li>메시지 크기 제한</li> <li>제한된 메시지 변환</li> <li>제한된 메시지 지속성</li> <li>제한된 프로토콜 지원</li> <li><b>복잡한 비용 체계</b></li> </ul>	<ul style="list-style-type: none"> <li>DIY 모델 설치/구성/운영/모니터링/유지보수</li> </ul>



# - DEMO -

