



AKS(Azure Kubernetes Service)와 쿠버네티스 툴들

MLSA, MCT 박민진

MCT, Gold MLSA 박민진

I love Cloud Infrastructure & DevOps ♥



Email: parkmin030505@naver.com

LinkedIn: <https://bit.ly/minjinpark>



Presentation / Lecture History

- [Korea MCT Summit 2025](#) | IaC in Azure: ARM vs Bicep vs Terraform (2025.05)
- [Docker & Linux on Windows: The Secret Link Between the Linux Kernel and Docker](#) (2025.03)
- [DDD Seoul\(2024\)](#) | Almost Everything Azure Serverless: From Static Web App Deployment to Container Orchestration (2024.12)
- [.NET Conf 2024](#) | ASP.NET Core and Blazor in .NET 9 [F/BE] (2024.11)
- [Exploring Kubernetes](#) (2024.08)
- [Deploying Web app by Container](#)(Container Serverless) (2024.07)
- [Seoul National University Science Computing Camp](#) | Teaching Assistant for "Cloud-Native Application Development with GitHub Copilot and Bicep" (2024.07)
- [Clutch \(Daegu Center for Creative Economy & Innovation Program\)](#) | AI-Powered Code Writing: GitHub Copilot (2024.05)
- [MS AI Odessey\(Microsoft\)](#) | Building Vision AI Solutions with Azure AI (2024.05)
- [2024 Global AI BootCamp](#) | Collecting Survey Data with Azure AI Document Intelligence (2024.03)

목차

1. 컨테이너 개념 이해

가볍고 빠른 실행 환경, VM과의 차이점

3. AKS 환경

Azure Kubernetes Service의 특징과 장점

2. 쿠버네티스 소개

컨테이너 오케스트레이션의 필요성과 핵심 기능

4. 핵심 도구

Helm, Prometheus, Grafana를 활용한 효율적 운영

컨테이너부터 마이크로서비스 운영 환경으로 쓰이는 쿠버네티스까지 개념을 함께 알아보시다.

1. 컨테이너: 가볍고 빠른 실행 환경

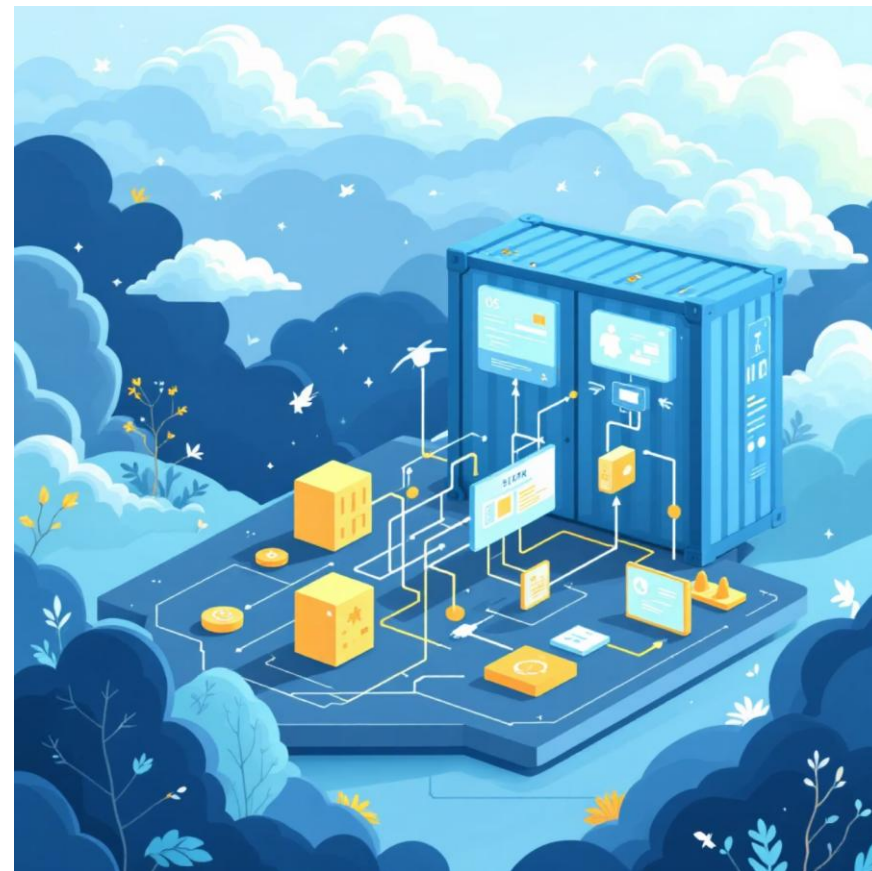
왜 컨테이너를 사용할까요?

[기존 - VM(Virtual Machine) 사용 방식]

- 무겁고 느림.
 - 개발 환경과 운영 환경의 불일치로 인한 문제가 빈번.
- "내 컴퓨터에서는 잘 되는데..."

[컨테이너]

- 애플리케이션과 필요한 라이브러리만 패키징하여 어떤 환경에서도 동일하게 실행될 수 있도록 보장.
- 호스트 OS의 커널을 공유하기 때문에 가볍고 빠르게 실행.



① 도커(Docker)는 컨테이너 기술의 사실상 표준으로, 애플리케이션을 컨테이너화하고 배포하는 가장 인기 있는 도구.

1. 컨테이너: 가볍고 빠른 실행 환경

잠자고 요리할 환경이 필요해!

VM(가상머신)

땅(호스트 컴퓨터의 OS)에 다가
집 짓고(가상화된 OS를 올려)
잠자고 요리할 환경 만들기(런타임 환경 설정)



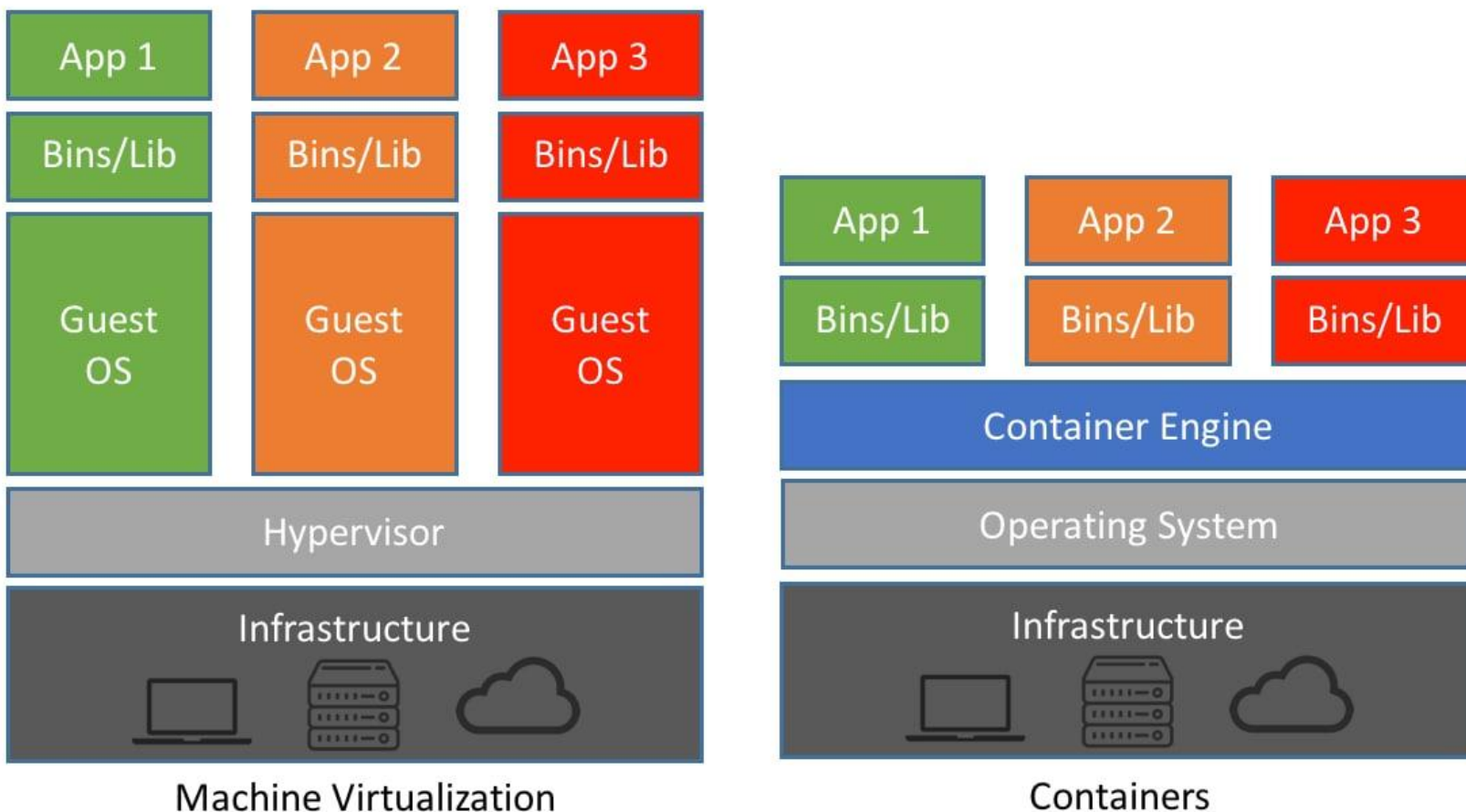
컨테이너

아무 곳(아무 컴퓨터)에 다가
[침대랑 버너](런타임 환경) 넣은 컨테이너 가져다 쓰기



① 도커(Docker)는 컨테이너 기술의 사실상 표준으로, 애플리케이션을 컨테이너화하고 배포하는 가장 인기 있는 도구.

1. 컨테이너: 가볍고 빠른 실행 환경



① 도커(Docker)는 컨테이너 기술의 사실상 표준으로, 애플리케이션을 컨테이너화하고 배포하는 가장 인기 있는 도구.

1. 컨테이너: 가볍고 빠른 실행 환경

가상 머신(VM) vs 컨테이너



가상 머신 (VM)

- 운영체제 전체를 가상화
- 하이퍼바이저 위에서 실행
- GB 단위의 크기
- 부팅 시간: 분 단위
- 자원 사용량이 큼



컨테이너

- 애플리케이션 계층만 가상화
- 호스트 OS 커널 공유
- MB 단위의 크기
- 시작 시간: 초 단위
- 자원 효율성 높음

컨테이너는 "한 번 작성하면 어디서나 실행"이라는 약속을 지켜줍니다!

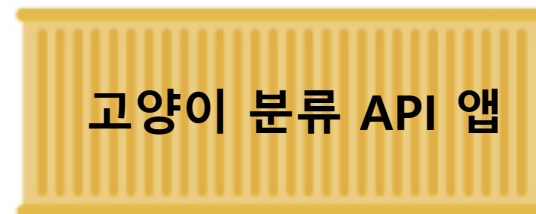
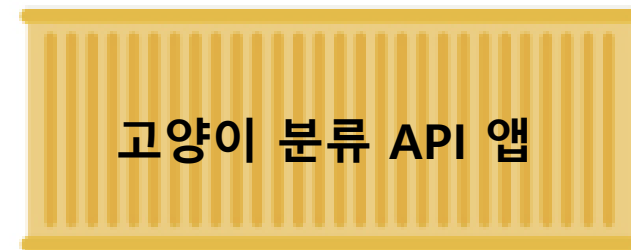
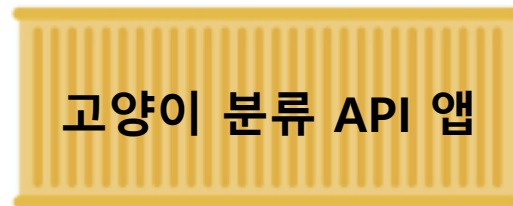
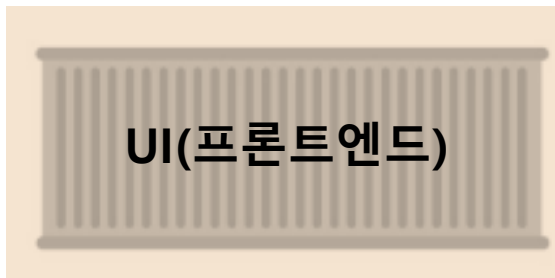
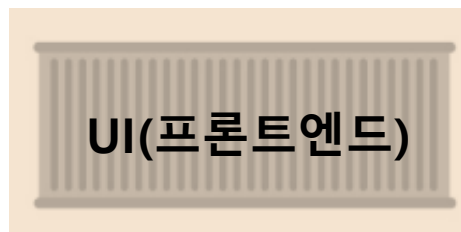
1. 컨테이너: 가볍고 빠른 실행 환경

(1) 애플리케이션과 필요한 라이브러리만 패키징하여 **어떤 환경에서도 동일**하게 실행될 수 있도록 보장.

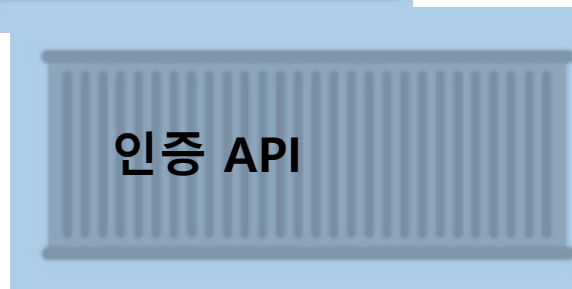
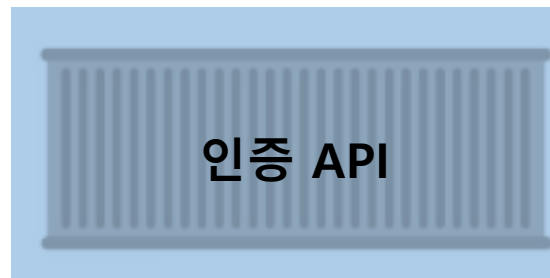
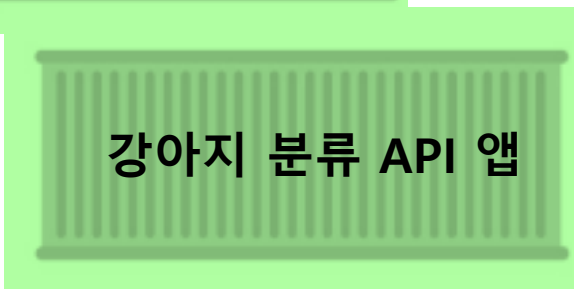
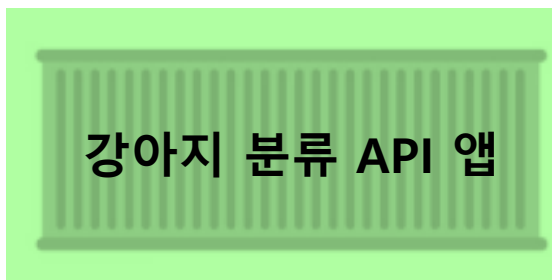
= 컴퓨터 1에서 만든 컨테이너를 컴퓨터 2도 쉽게 쓸 수 있다.

(2) 가볍고 빠르다

결론: 앱 배포 또는 실행을 컨테이너로 대부분 한다.



실제 서비스를 위해 많은 컨테이너가 동작





2. 쿠버네티스 : 컨테이너 오케스트레이션의 마법사

컨테이너 관리의 필요성

컨테이너가 수십, 수백 개로 늘어나면 복잡한 문제 발생!

- 컨테이너가 다운되면 어떻게 복구할까요?
- 업데이트는 어떻게 무중단으로 진행할까요?
- 트래픽이 증가하면 어떻게 확장할까요?
- 자원 할당은 어떻게 최적화할까요?
- 컨테이너 간 통신은 어떻게 관리할까요?
- 수백 개의 컨테이너를 일일이 관리할 수 있을까요?

"쿠버네티스는 컨테이너 선박을 효율적으로 관리하는 거대한 '컨테이너 터미널'과 같습니다."

2. 쿠버네티스 개념 소개

쿠버네티스의 핵심 기능



자동 복구 (Self-Healing)

컨테이너에 문제가 생기면 자동으로 재시작하거나 교체.
야간에 발생한 장애도 아침에 출근했을 때 이미 해결되어 있음.



자동 확장 (Scaling)

트래픽이나 CPU/메모리 사용량에 따라 컨테이너 수를 자동으로 조절. 갑작스러운 트래픽 증가에도 안정적인 서비스 가능.



서비스 검색 (Service Discovery)

컨테이너들이 서로를 쉽게 찾고 통신할 수 있도록 내부 DNS와 로드 밸런싱을 제공. 복잡한 네트워크 설정이 필요 없음.



무중단 배포 (Rolling Update)

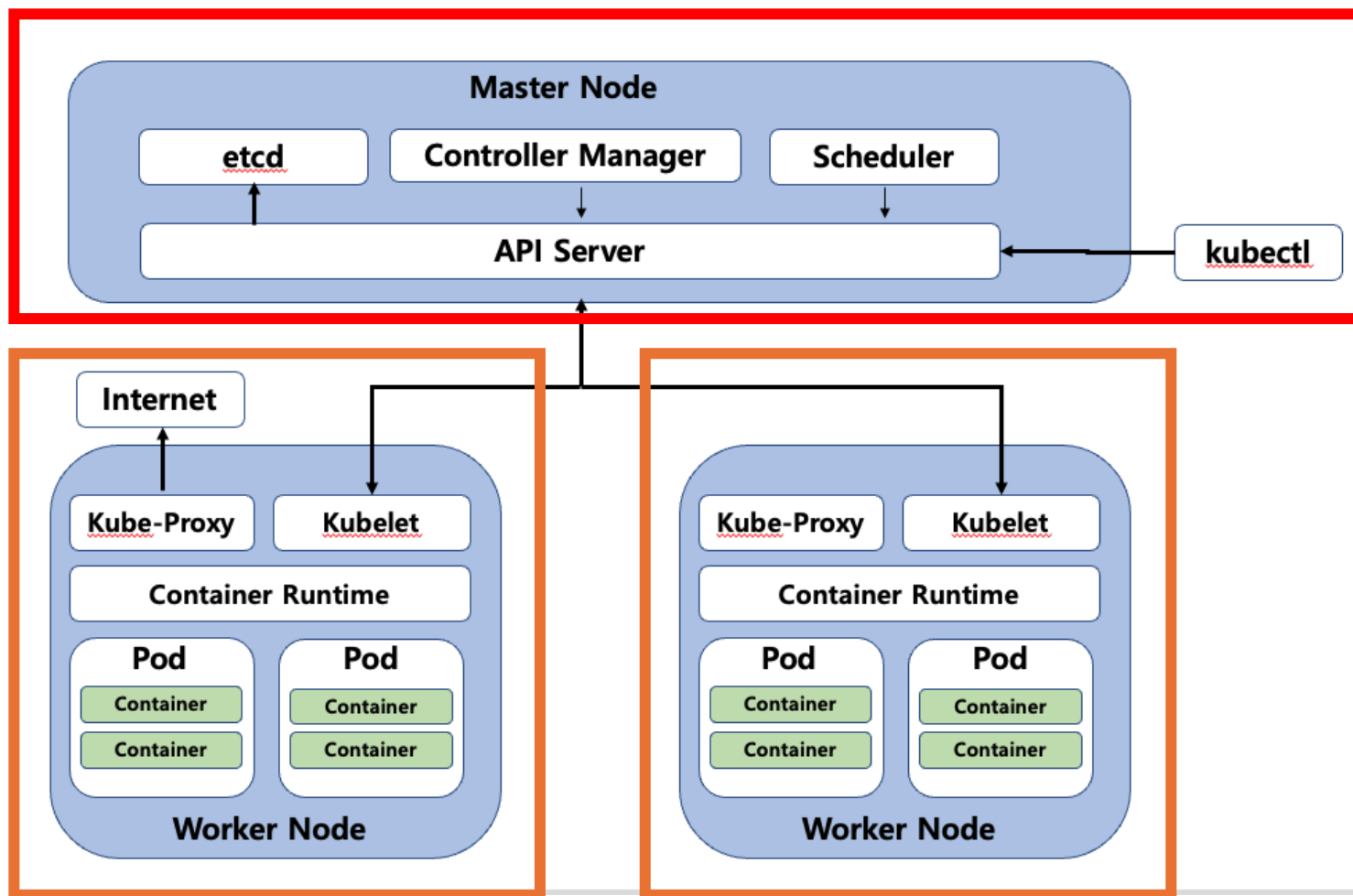
서비스 중단 없이 애플리케이션을 업데이트 가능.
사용자는 업데이트가 진행 중인지 알 수 없을 정도로 매끄럽게 진행됨.

이 모든 기능을 마이크로소프트의 [AKS\(Azure Kubernetes Service\)](#)에서 관리형 서비스로 쉽게 사용할 수 있음.

2. 쿠버네티스 개념 소개

쿠버네티스의 구조

관리자 = Control Plane = Master Node



일하는 사람들
= Data Plane
= Worker Node

2. 쿠버네티스 개념 소개

쿠버네티스의 구조

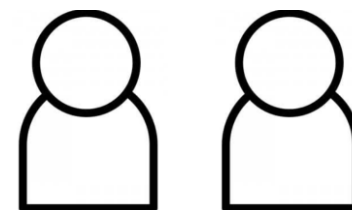
관리자 = Control Plane = Master Node



일꾼 = Data Plane = Worker Node

- 1 Master 3 (Worker) Node 구성
- 4 Node 클러스터

관리자 = Control Plane = Master Node

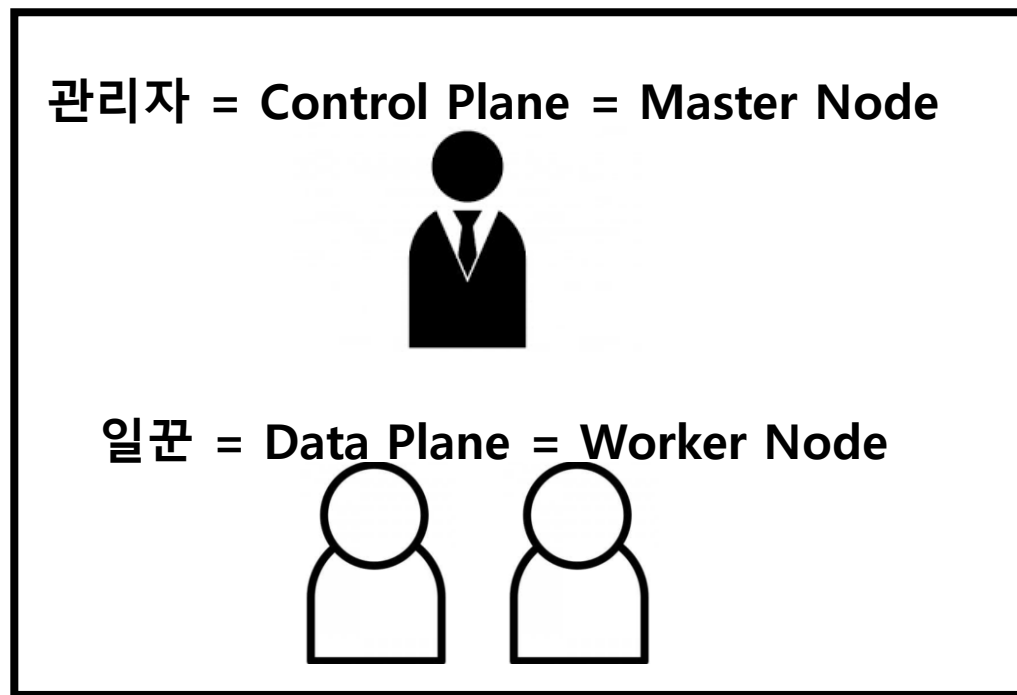


일꾼 = Data Plane = Worker Node

- 1 Master 2 (Worker) Node 구성
- 3 Node 클러스터

3. AKS 환경

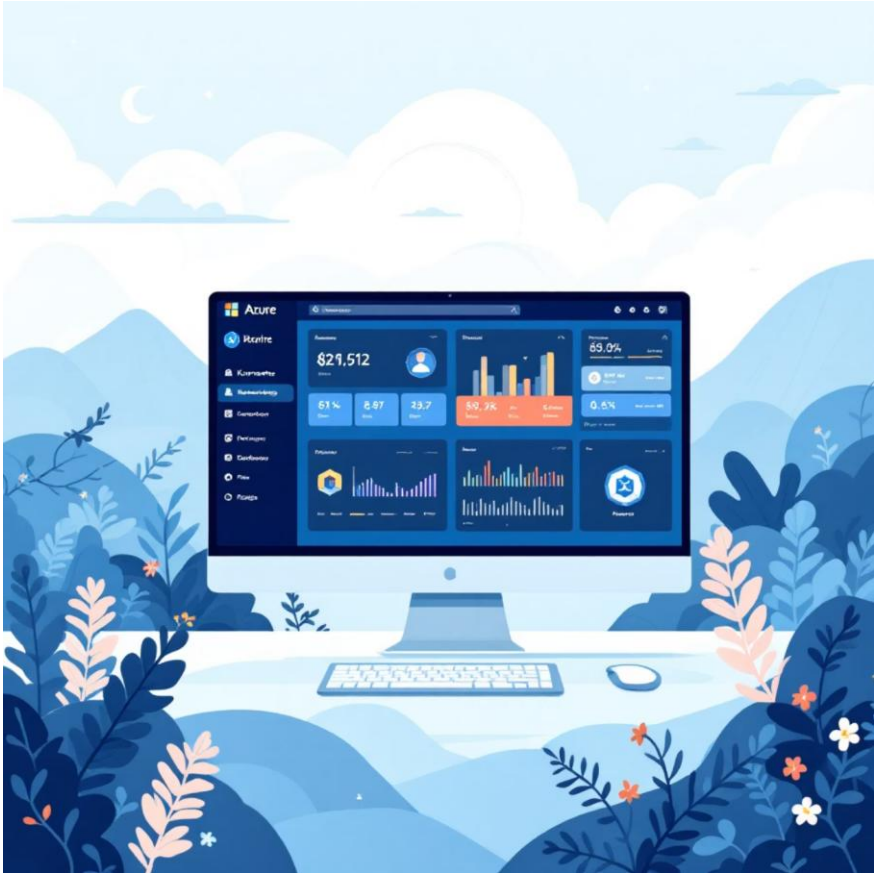
쿠버네티스 클러스터 직접 구축: 복잡!



- (1) 각 노드를 위한 서버 구축
- (2) 각 노드에 쿠버네티스 설치
- (3) 노드들(서버들)간 네트워크 연결

.
. .

AKS: Azure Kubernetes Service



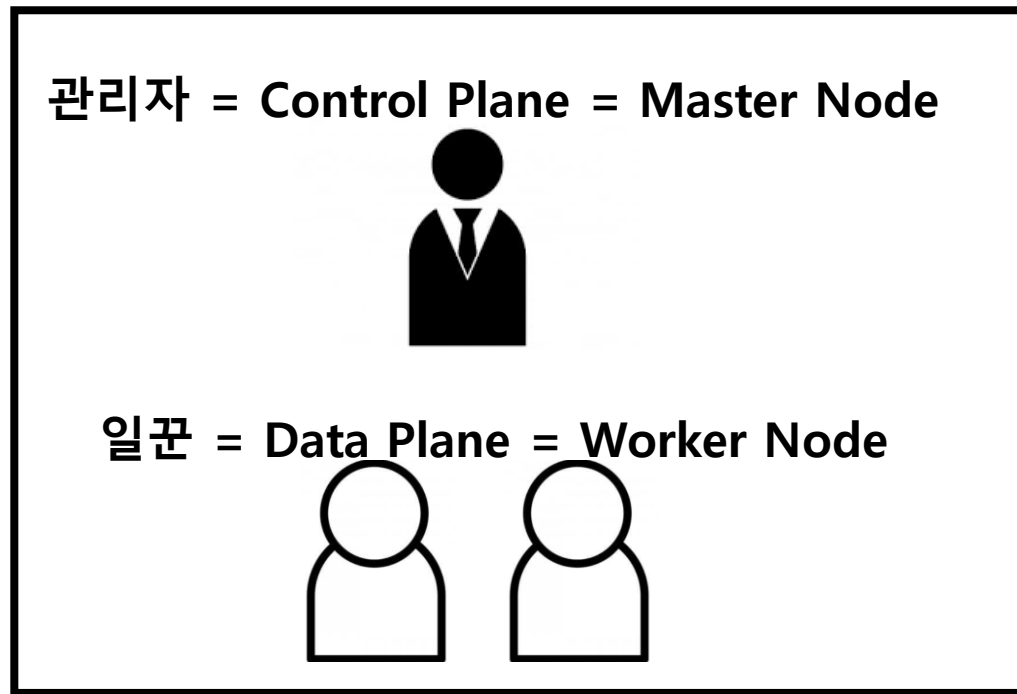
AKS의 주요 특징

- 완전 관리형 쿠버네티스 서비스
- 마스터 노드 무료 제공 (컨트롤 플레인)
- Azure 서비스와의 원활한 통합
- 보안 및 컴플라이언스 기능 내장
- 자동화된 업그레이드 및 패치
- 확장성이 뛰어난 인프라 제공

✔ **비용 절감 팁:** AKS는 워커 노드에 대해서만 비용을 지불하며, 컨트롤 플레인은 무료로 제공. 또한 Azure 예약 인스턴스와 함께 사용하면 최대 72%까지 비용을 절감할 수 있음.

3. AKS 환경

쿠버네티스 클러스터 직접 구축: 복잡!



- (1) 각 노드를 위한 서버 구축
- (2) 각 노드에 쿠버네티스 설치
- (3) 노드들(서버들)간 네트워크 연결

.
. .
. .

3. AKS 환경

기본으로 제공되는
Master Node에서 “명령어 하나” 로 쉽게
Worker Node(일꾼들) 무한 증식 가능

```
Requesting a Cloud Shell. Succeeded.  
Connecting terminal...  
  
Welcome to Azure Cloud Shell  
  
Type "az" to use Azure CLI  
Type "help" to learn about Cloud Shell  
  
MOTD: SqlServer has been updated to Version 22!  
  
VERBOSE: Authenticating to Azure ...  
VERBOSE: Building your Azure drive ...  
  
PS /home/carolb>
```

Azure Portal의 Cloud Shell(온라인 ver Azure CLI),
Azure CLI,
Azure Developer CLI에서 AKS 사용 가능

3. AKS 환경

```
az group create --name MyAKSCluster --location eastus
```

```
az aks create --resource-group MyAKSCluster --name myAKSCluster --node-count 1 --enable-managed-identity --generate-ssh-keys
```

```
az aks get-credentials --resource-group MyAKSCluster --name myAKSCluster
```

```
kubectl get nodes
```

```
minjin [ ~ ]$ az group create --name MyAKSCluster --location eastus
{
  "id": "/subscriptions/0bd90d67-2234-4f80-8be8-c15043ac40be/resourceGroups/MyAKSCluster",
  "location": "eastus",
  "managedBy": null,
  "name": "MyAKSCluster",
  "properties": {
    "provisioningState": "Succeeded"
  },
  "tags": null,
  "type": "Microsoft.Resources/resourceGroups"
}
minjin [ ~ ]$ az aks create --resource-group MyAKSCluster --name myAKSCluster --node-count 1 --enable-managed-identity --generate-ssh-keys
SSH key files '/home/minjin/.ssh/id_rsa' and '/home/minjin/.ssh/id_rsa.pub' have been generated under ~/.ssh to allow SSH access to the VM.
Azure Cloud Shell without an attached file share, back up your keys to a safe location
minjin [ ~ ]$ az aks get-credentials --resource-group MyAKSCluster --name myAKSCluster
Merged "myAKSCluster" as current context in /home/minjin/.kube/config
minjin [ ~ ]$ kubectl get nodes
NAME                                STATUS    ROLES    AGE   VERSION
aks-nodepool1-39026178-vmss000000  Ready    <none>   2m51s v1.32.6
```

4. 핵심 도구

Helm: 쿠버네티스용 패키지 관리자

문제점

쿠버네티스에 애플리케이션을 배포하려면 다양한 리소스를 정의하는 여러 YAML 파일 작성해야 함

- Deployment
- Service
- ConfigMap
- Secret
- Ingress
- ServiceAccount

이런 파일들을 관리하고 배포하는 것은 복잡하고 오류가 발생하기 쉬움.

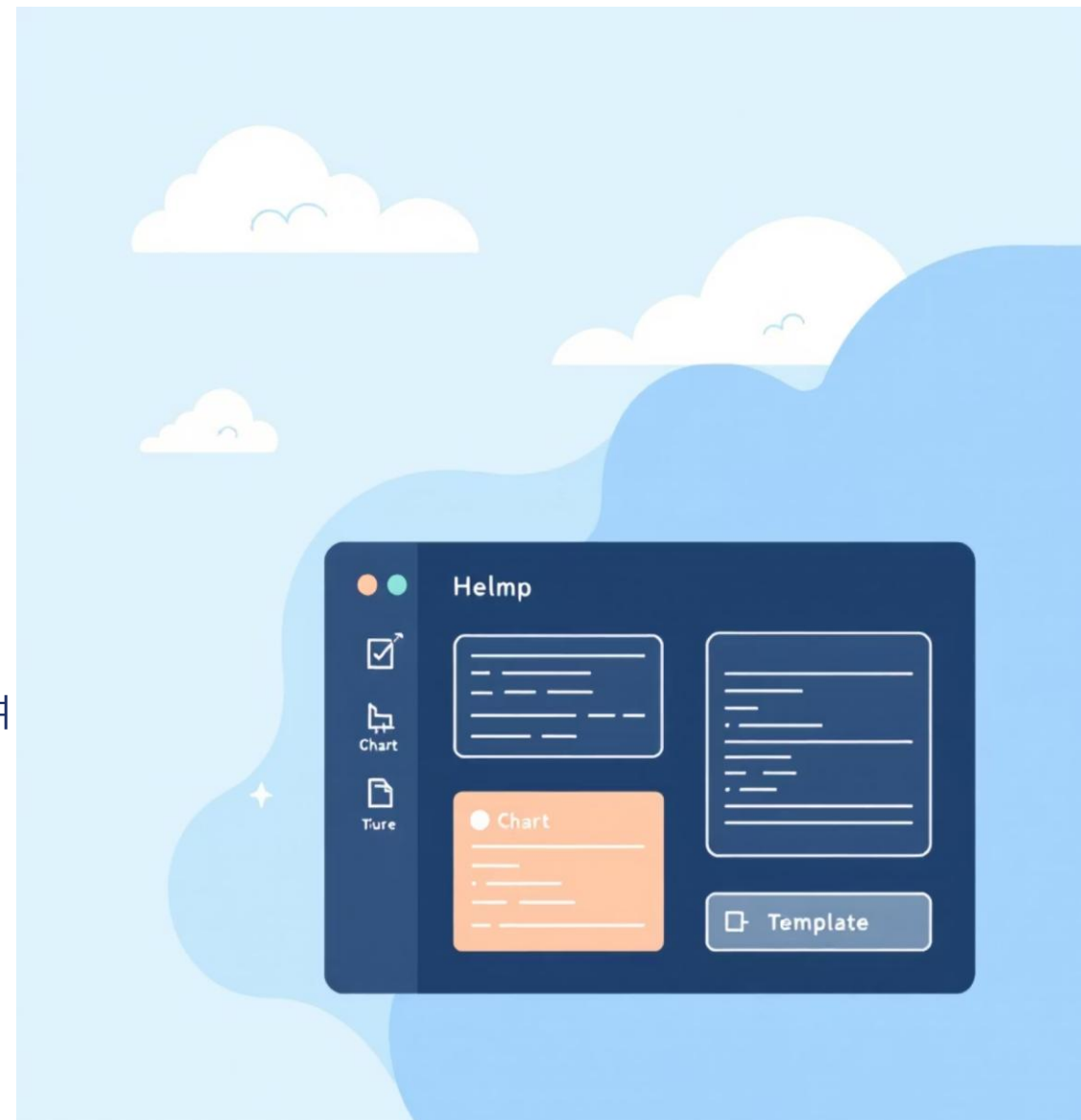
Helm의 해결책

Helm 차트(Chart)는 애플리케이션 배포에 필요한 모든 리소스를 패키징하여 단일 명령으로 설치할 수 있게 해줌.

```
helm install prometheus prometheus-community/prometheus
```

Tip. 김치찌개 끓이기

: 두부, 돼지고기, 파 등 사기 vs 밀키트 사기

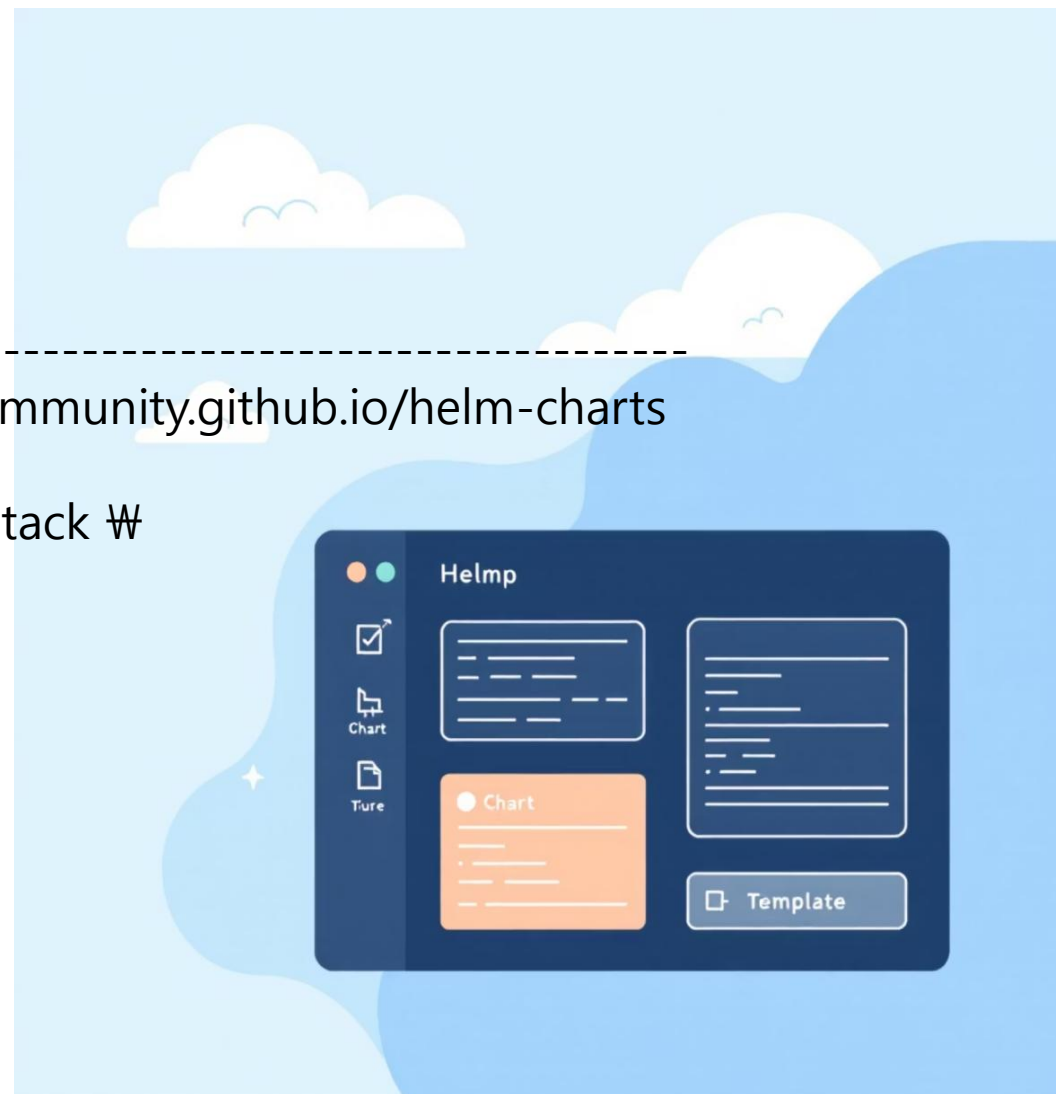


4. 핵심 도구

Helm: 쿠버네티스용 패키지 관리자

```
helm repo add bitnami https://charts.bitnami.com/bitnami  
helm repo update  
helm install my-nginx bitnami/nginx
```

```
helm repo add prometheus-community https://prometheus-community.github.io/helm-charts  
helm install ₩  
prometheus-stack prometheus-community/kube-prometheus-stack ₩  
--namespace monitoring --create-namespace
```



4. 핵심 도구

Prometheus & Grafana: 모니터링 솔루션



Prometheus

시계열 데이터베이스로 클러스터의 다양한 메트릭을 수집하고 저장.
강력한 쿼리 언어(PromQL)를 통해 데이터 분석 및 알림 설정이 가능합니다.



Grafana

Prometheus에서 수집한 데이터를 대시보드로 시각화.
실시간 모니터링으로 클러스터 상태를 한눈에 파악할 수 있습니다.

- ❏ **실무 팁:** Prometheus와 Grafana는 Helm을 통해 쉽게 설치할 수 있으며, 기본 대시보드와 알림 규칙이 포함된 [kube-prometheus-stack](#) 차트를 사용하는 것이 좋습니다.

4. 핵심 도구

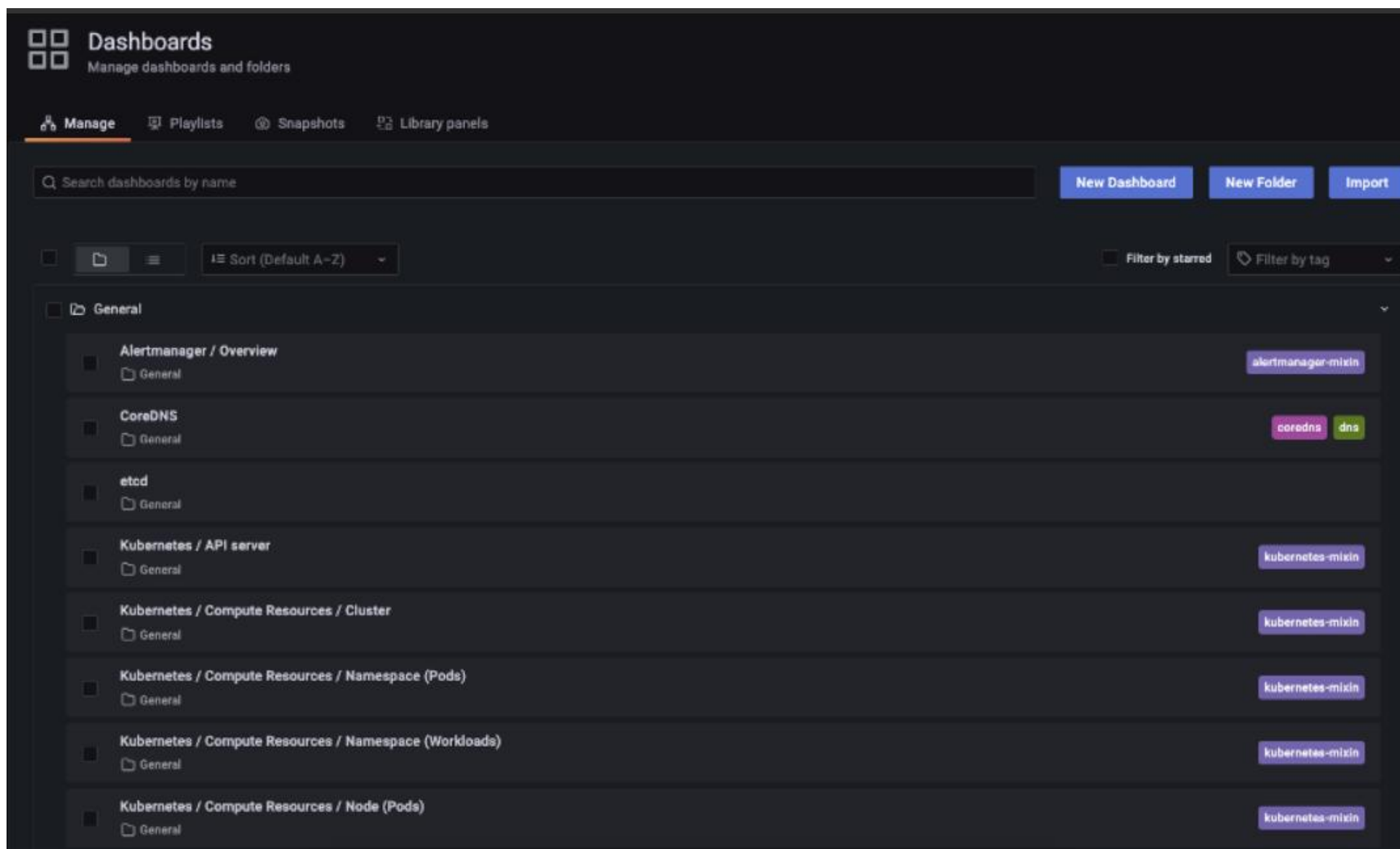
Prometheus & Grafana: 모니터링 솔루션

```
localhost:9100/metrics
# HELP node_cooling_device_max_state Maximum throttle state of the cooling device
# TYPE node_cooling_device_max_state gauge
node_cooling_device_max_state{name="0",type="Processor"} 0
node_cooling_device_max_state{name="1",type="Processor"} 0
# HELP node_cpu_guest_seconds_total Seconds the CPUs spent in guests (VMs) for each mode.
# TYPE node_cpu_guest_seconds_total counter
node_cpu_guest_seconds_total{cpu="0",mode="nice"} 0
node_cpu_guest_seconds_total{cpu="0",mode="user"} 0
node_cpu_guest_seconds_total{cpu="1",mode="nice"} 0
node_cpu_guest_seconds_total{cpu="1",mode="user"} 0
# HELP node_cpu_seconds_total Seconds the CPUs spent in each mode.
# TYPE node_cpu_seconds_total counter
node_cpu_seconds_total{cpu="0",mode="idle"} 2.45064998e+06
node_cpu_seconds_total{cpu="0",mode="iowait"} 787.09
node_cpu_seconds_total{cpu="0",mode="irq"} 0
node_cpu_seconds_total{cpu="0",mode="nice"} 3.28
node_cpu_seconds_total{cpu="0",mode="softirq"} 5512.32
node_cpu_seconds_total{cpu="0",mode="steal"} 50390.63
node_cpu_seconds_total{cpu="0",mode="system"} 55072.81
node_cpu_seconds_total{cpu="0",mode="user"} 184165.67
node_cpu_seconds_total{cpu="1",mode="idle"} 2.45099891e+06
node_cpu_seconds_total{cpu="1",mode="iowait"} 786.08
node_cpu_seconds_total{cpu="1",mode="irq"} 0
node_cpu_seconds_total{cpu="1",mode="nice"} 2.35
node_cpu_seconds_total{cpu="1",mode="softirq"} 5707.27
node_cpu_seconds_total{cpu="1",mode="steal"} 50413.99
node_cpu_seconds_total{cpu="1",mode="system"} 55017.49
node_cpu_seconds_total{cpu="1",mode="user"} 183888.57
```



4. 핵심 도구

Prometheus & Grafana: 모니터링 솔루션



4. 핵심 도구

Prometheus & Grafana: 모니터링 솔루션



AKS(Azure Kubernetes Service)와 쿠버네티스 툴들



컨테이너

가볍고 일관된 실행 환경으로 "어디서나 동일하게 작동"하는 애플리케이션 배포 방식



쿠버네티스

컨테이너를 자동으로 배포, 확장, 관리하는 오케스트레이션 플랫폼



AKS

클라우드 환경에서 쿠버네티스를 쉽게 구축하고 운영할 수 있는 관리형 서비스



핵심 도구

Helm, Prometheus, Grafana로 배포 자동화 및 모니터링 강화

"이 모든 기술은 **마이크로서비스 아키텍처**를 안정적이고 효율적으로 운영하기 위한 기반.
작은 서비스들이 독립적으로 개발되고 배포되어 함께 강력한 시스템을 구성할 수 있게 합니다."

감사합니다.



LinkedIn



AKS