

# Prometheus + Grafana 모니터링

---

VPS 상태를 실시간으로, 사전 감지로 장애 차단

● 문제

● 도구

○ 아키텍처

○ 구축

○ 메트릭

○ 대시보드

○ 알림

# Prometheus vs Grafana

Prometheus

**수집 + 저장**

Pull 방식, 시계열 DB, PromQL 쿼리, CNCF 졸업

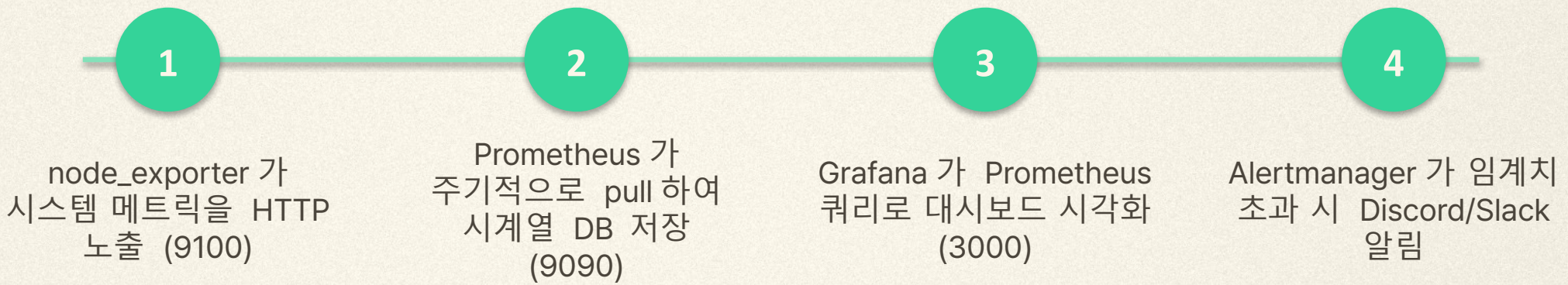
Grafana

**시각화**

그래프 / 대시보드, 커뮤니티 템플릿, 전 세계 수만 개 기업 사용

- 문제
- 도구
- 아키텍처
- 구축
- 메트릭
- 대시보드
- 알림

# 모니터링 데이터 흐름



- 문제
- 도구
- 아키텍처
- **구축**
- 메트릭
- 대시보드
- 알림

# docker-compose 서비스 정의

```
# docker-compose.yml ( 요약 )
services:
  prometheus:
    image: prom/prometheus
    ports: ["9090:9090"]
    volumes: [ './prometheus.yml:/etc/prometheus/prometheus.yml' ]

  grafana:
    image: grafana/grafana
    ports: ["3000:3000"]

  node_exporter:
    image: prom/node-exporter
    ports: ["9100:9100"]
```

- 문제
- 도구
- 아키텍처
- **구축**
- 메트릭
- 대시보드
- 알림

# prometheus.yml

```
global:
  scrape_interval: 15s # 기본 15 초, 실시간 필요 시 5s

scrape_configs:
  - job_name: 'node'
    static_configs:
      - targets: ['localhost:9100']
  # 다중 서버: 타겟 추가만 하면 됨
  - targets: ['worker2:9100']
```

- 문제
- 도구
- 아키텍처
- 구축
- **메트릭**
- 대시보드
- 알림

## node\_exporter 수집 메트릭

- CPU 사용률 (코어별)
- 메모리 사용량 + 스왑
- 디스크 여유 공간 (파일시스템별)
- 네트워크 트래픽 (수신 / 송신)
- 로드 평균값
- 수백 개 메트릭이 자동 수집 — 스크립트 불필요

- 문제
- 도구
- 아키텍처
- 구축
- 메트릭
- **대시보드**
- 알림

## Grafana PromQL 예시

# CPU 사용률 (%)

```
100 - (avg(rate(node_cpu_seconds_total{mode="idle"}  
[5m])) * 100)
```

# 메모리 사용률 (%)

```
(node_memory_MemAvailable_bytes /  
node_memory_MemTotal_bytes) * 100
```

# 디스크 여유 공간

```
node_filesystem_avail_bytes{mountpoint="/"}
```

# 네트워크 초당 수신 바이트

```
rate(node_network_receive_bytes_total[5m])
```

- 문제
- 도구
- 아키텍처
- 구축
- 메트릭
- 대시보드
- 알림

# Alertmanager 알림 조건

CPU 90%+ 5 분

CriticalAlert

디스크 여유  
<10%

DiskFull

메모리 95%+

MemoryWarning

서버 무응답 1 분

ServerDown → Discord/Slack

# 모니터링으로 장애 대응 시간 80% 단축

 [Vultr \\$100 크레딧 \( 설명란 \)](#)

 [전체 설정 가이드는 블로그에](#)

 [구독 & 좋아요 부탁드립니다](#)

구독과 좋아요 부탁드립니다! 