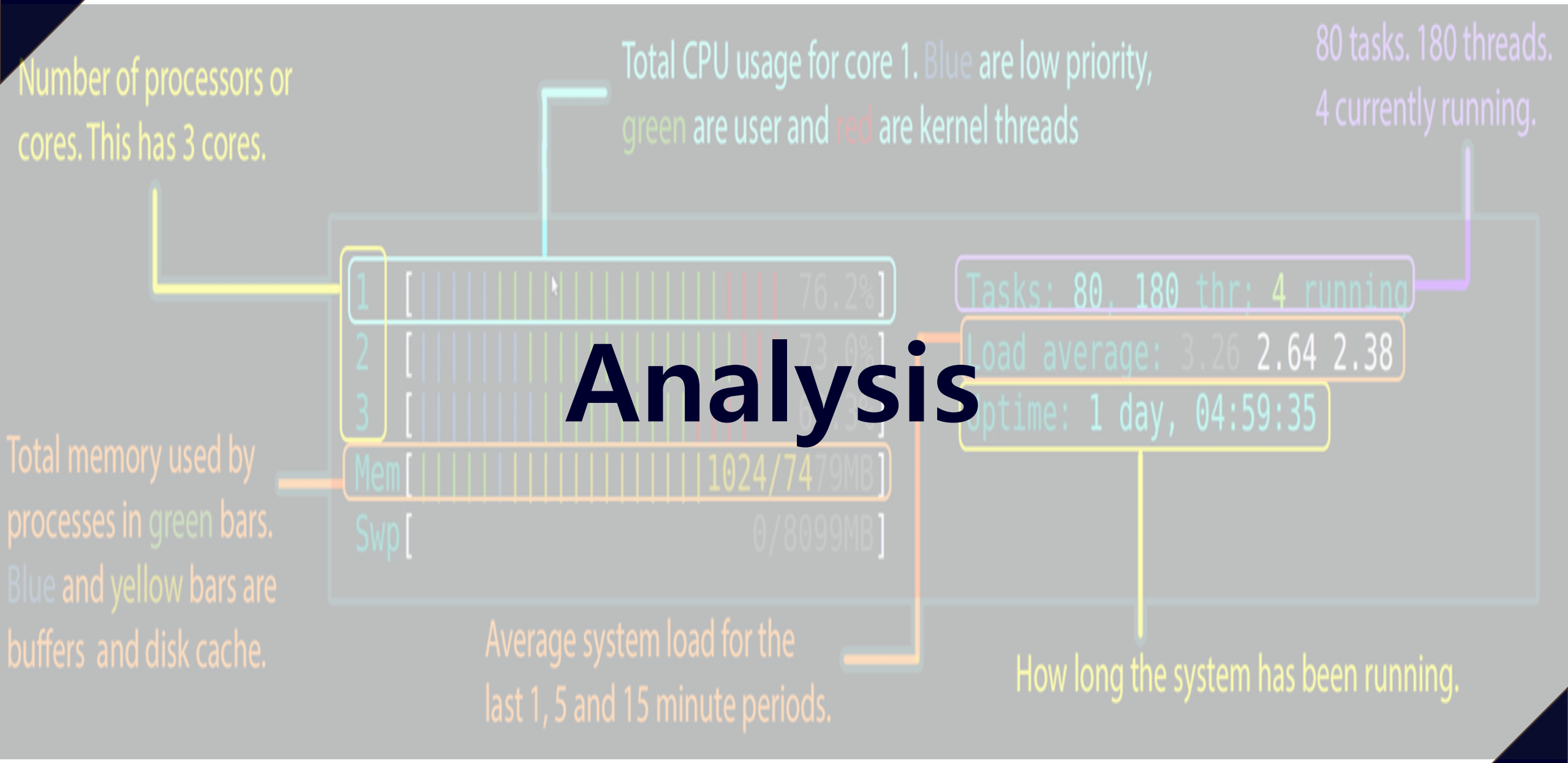
OpenInfra Monitoring with Prometheus

본 세션에서는...

- OpenInfra & Cloud Monitoring
- TSDB (Time Series Data Base)
- What is Prometheus
- Kubernetes Monitoring Demo
- 활용 사례 공유

WHY YOUR COMPANY NEEDS A MONITORING SYSTEM

Alerting

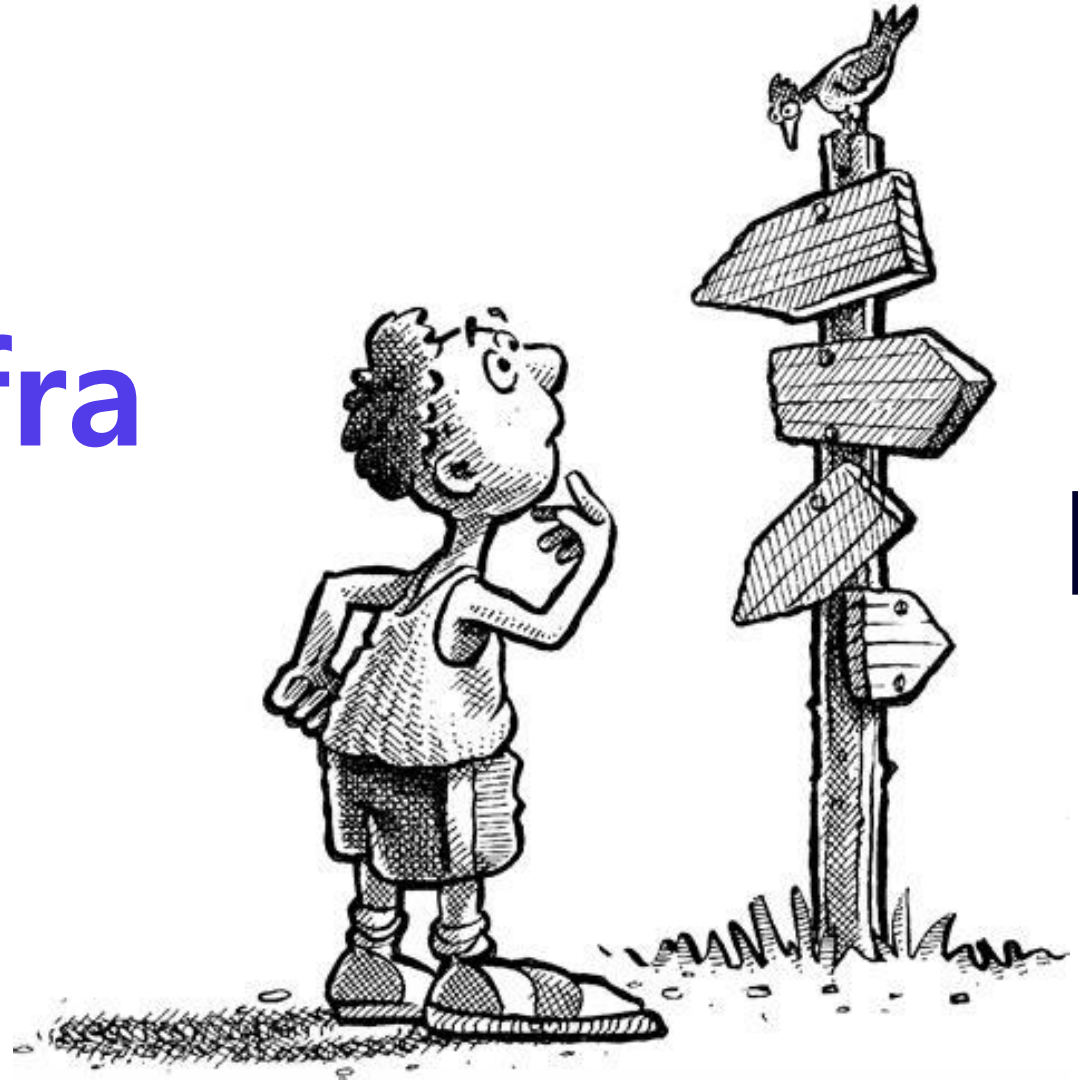

Number of processors or cores. This has 3 cores.
Total CPU usage for core 1. Blue are low priority, green are user and red are kernel threads
80 tasks. 180 threads. 4 currently running.
1 [76.2%]
2 [73.0%]
3 [
Mem[1024/7479MB]
Swp[0/8099MB]
Tasks: 80, 180 thr; 4 running
Load average: 3.26 2.64 2.38
Uptime: 1 day, 04:59:35
Total memory used by processes in green bars. Blue and yellow bars are buffers and disk cache.
Average system load for the last 1, 5 and 15 minute periods.
How long the system has been running.

Analysis


Big Dashboard
Zoom Out
Oct 4, 2015 11:29:09 to Oct 4, 2015 14:13:23 UTC
Logins
172
Sign ups
263
Sign outs
268
Support calls
80
Memory / CPU
logins
Memory / CPU
logins
memory
cpu
logins
logins (-1 hour)
server requests
upper_25 6 ms
upper_50 75 ms
upper_75 411 ms
upper_90 974 ms
upper_95 1.288 s
avg
server requests
web_server_01
web_server_02
web_server_03
web_server_04
Trend

Monitoring Tools

OpenInfra Cloud



Monitoring

Lifecycle

- Scale-IN/OUT
- Hostname, Floating IP DHCP
- VM Reload
- Location

Stateless

- Ephemeral Disk
- Container
- Metric, Log

Target

- HOST, Hypervisor, Docker
- Openstack, Kubernetes, Mesos/Marathon, Swarm
- VM, Container
- Service, Application
- Ping, Port, Http Check

How can this be solved

CLOUD NATIVE
COMPUTING
FOUNDATION

다음과 같은 것이 필요합니다.

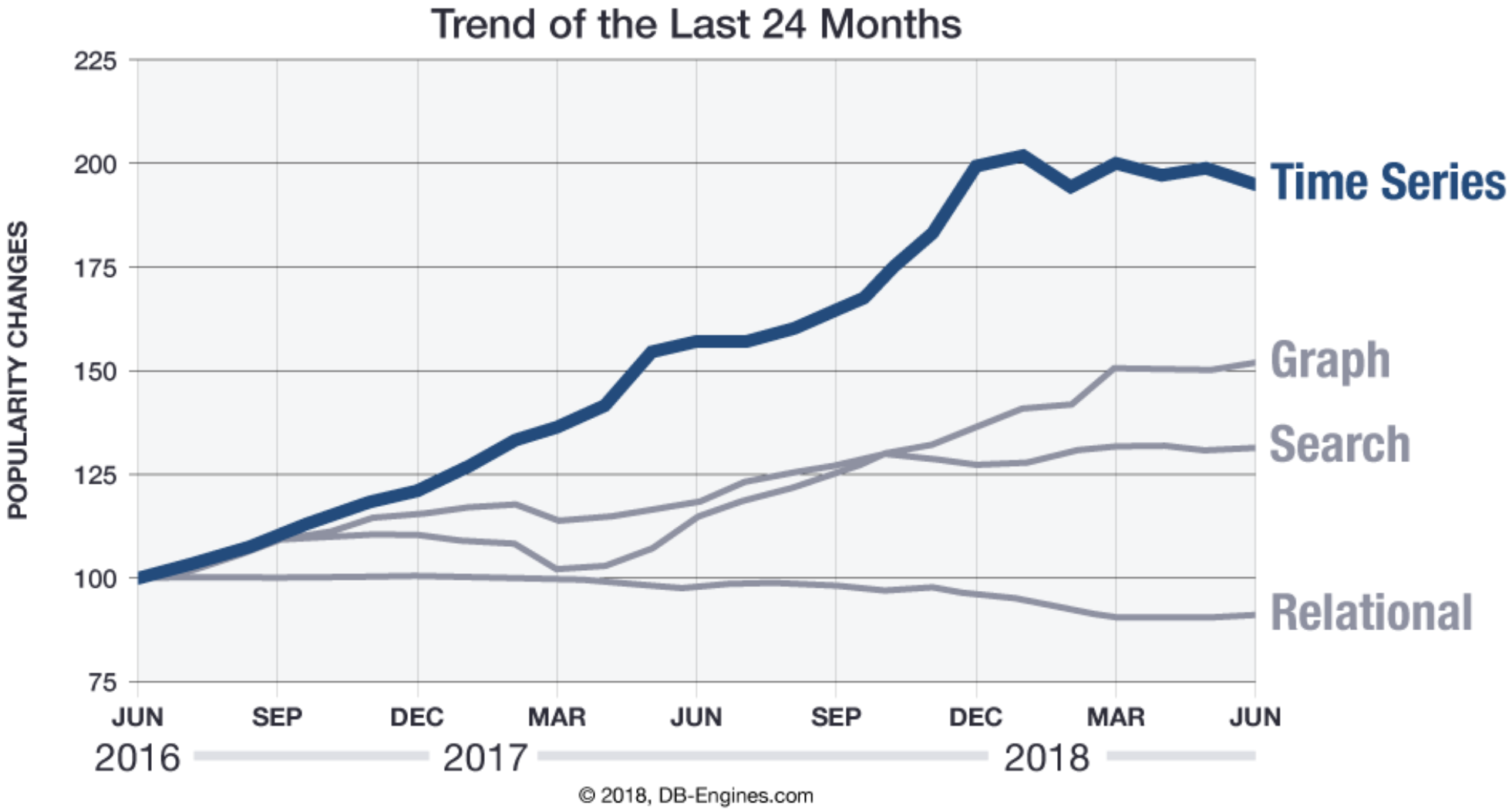
Metric수집

vs

Metric
Visualize

Alerting

TSDB (Time Series Data Base) 확산



시계열 데이터 저장/처리에 최적화 Database

Example

- H/W Metric
- OS Metric
- VM/Container Metric
- Platform Metric
- Application Metric

Metric Data

- High write Performance
- Quick to process
- Easy Range Query
- Data Compaction
- Cost Efficient

공통적인 특징을 요약하자면...

- Metric Storage (InfluxDB, Prometheus, OpenTSDB, Graphite …)
- Metric 수집을 위한 API 및 Interface 제공
- Metric 조회를 Query SQL / Web 관리 UI 제공

25 systems in ranking, June 2018

Rank			DBMS	Database Model	Score		
Jun 2018	May 2018	Jun 2017			Jun 2018	May 2018	Jun 2017
1.	1.	1.	IninfluxDB	Time Series DBMS	11.33	+0.33	+3.13
2.	2.	↑ 5.	Kdb+	Multi-model	3.02	-0.06	+1.44
3.	3.	↓ 2.	RRDtool	Time Series DBMS	2.67	-0.01	-0.35
4.	4.	↓ 3.	Graphite	Time Series DBMS	2.38	+0.12	+0.38
5.	5.	↓ 4.	OpenTSDB	Time Series DBMS	1.56	-0.06	-0.24
6.	6.	↑ 8.	Prometheus	Time Series DBMS	1.27	+0.14	+0.66
7.	7.	↓ 6.	Druid	Time Series DBMS	1.13	+0.12	+0.13
8.	8.	↓ 7.	KairosDB	Time Series DBMS	0.41	-0.02	-0.21
9.	9.	9.	eXtremeDB	Multi-model	0.28	-0.03	-0.09
10.	10.	↑ 11.	Riak TS	Time Series DBMS	0.21	-0.05	-0.03
11.	↑ 14.	↓ 10.	Axibase	Time Series DBMS	0.11	+0.06	-0.14
12.	↓ 11.	↑ 14.	FaunaDB	Multi-model	0.11	+0.00	+0.02
13.	↓ 12.	↑ 19.	Hawkular Metrics	Time Series DBMS	0.11	+0.00	+0.07
14.	↓ 13.	↑ 15.	Blueflood	Time Series DBMS	0.09	-0.01	+0.01

Metric vs LOG

‘Time + Counter’

- No shaping and Easy aggregation
- Quick to process & visualize
- Cost Efficient

- Great for Abnormal detection, trending

‘Time + events’

- Individual events
- Shaping before processing
- higher I/O and network requirements
- Scaling can be costly

- Great for Deep-dive and drill down to individual events

Metric vs LOG (Apache Access Log)

Requests						
Logs	Method	Path	Code	Response time	Browser info	
	GET	/foo	200	1234	true	
	POST	/endpoint	500	299	true	
	GET	/foo	200	399	flase	
	GET	/foo	200	273	true	
	GET	/foo	200	101	true	
	POST	/endpoint	500	300	true	
	GET	/foo	200	450	true	
	GET	/foo	200	2327	true	

Requests						
Metrics	GET : 6	/foo: 6	200: 6	response_sum : 4784	response_count : 6	mobile : 5
	POST: 2	/endpoint : 1	500: 1	response_sum : 599	response_count : 2	mobile : 2


WE ARE ONE TEAM • WE ARE ONE TEAM • WE ARE ONE TEAM •

prometheus.io

What is Prometheus

Features

- Time-series database
- Metrics collection
- Service Discovery
- Graphing
- Alerting

Performance

- Millions of Time series
- Thousands of targets

About Prometheus

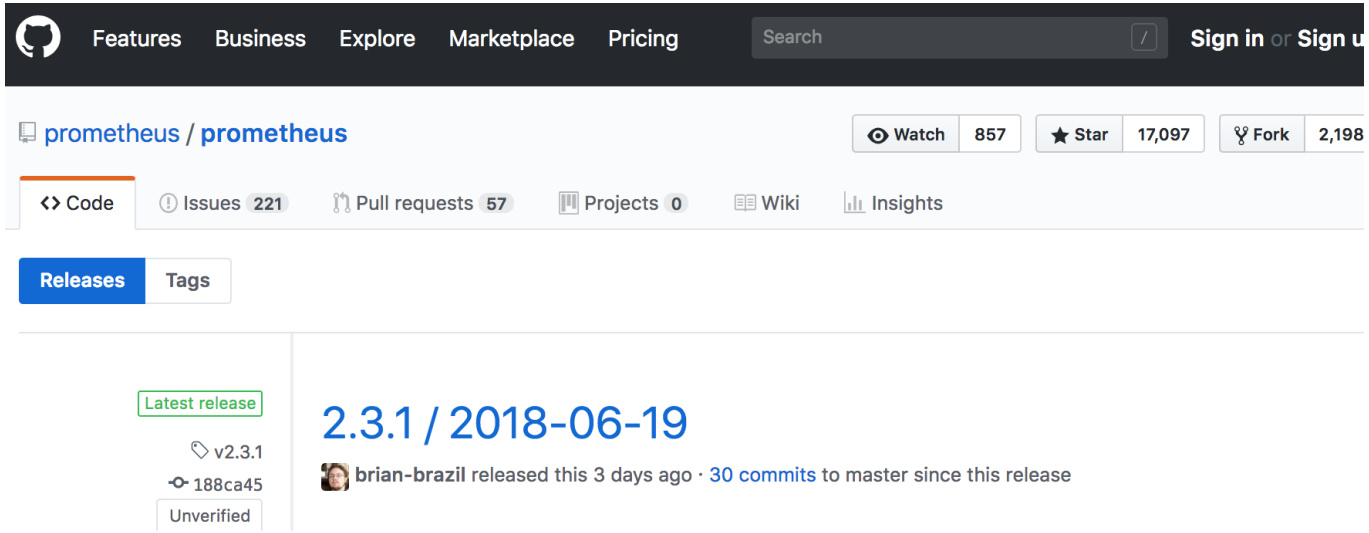
Open Source

Prometheus is 100% open source and community-driven. All components are available under the Apache 2 License on GitHub.

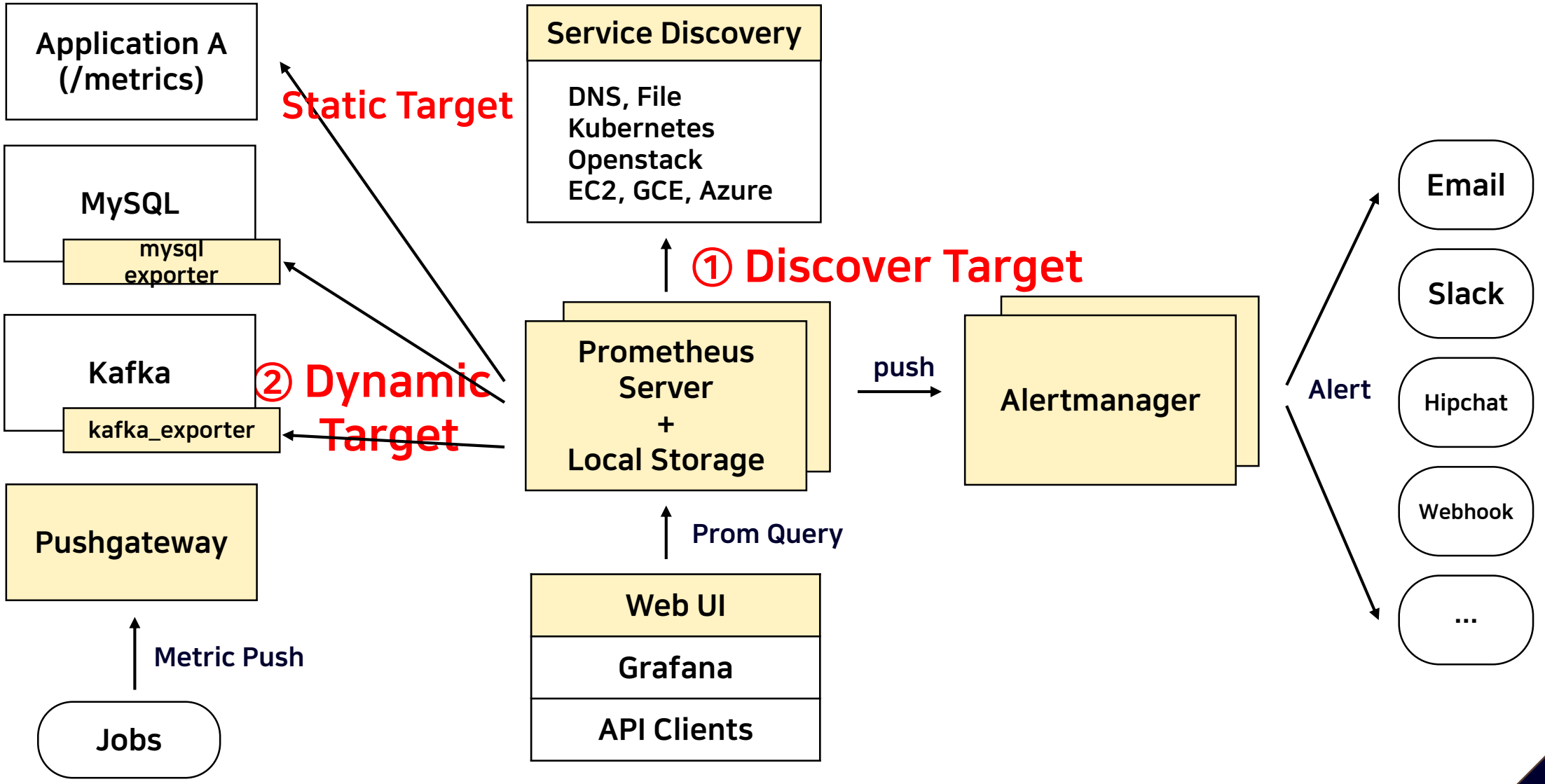
Star 16,985

We are a Cloud Native Computing Foundation member project.

CLOUD NATIVE COMPUTING FOUNDATION



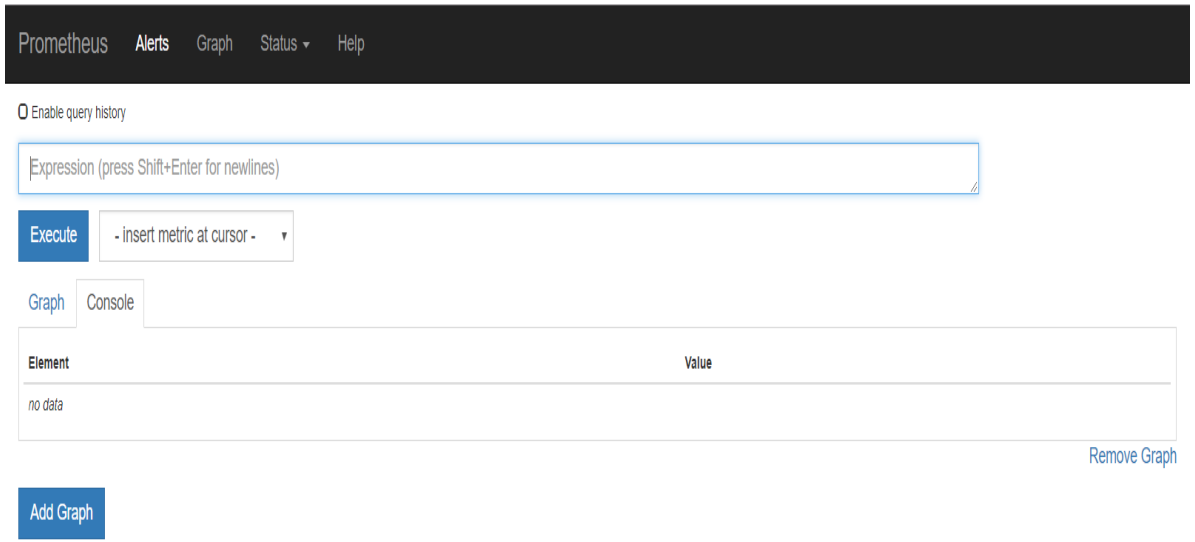
Prometheus Architecture



Prometheus Install & Config

Prometheus binary and run

```
$ wget https://github.com/.../prometheus-2.3.0.linux-amd64.tar.gz

$ tar xvfz prometheus-*.tar.gz
$ prometheus-*

./prometheus --config.file=prometheus.yml
```

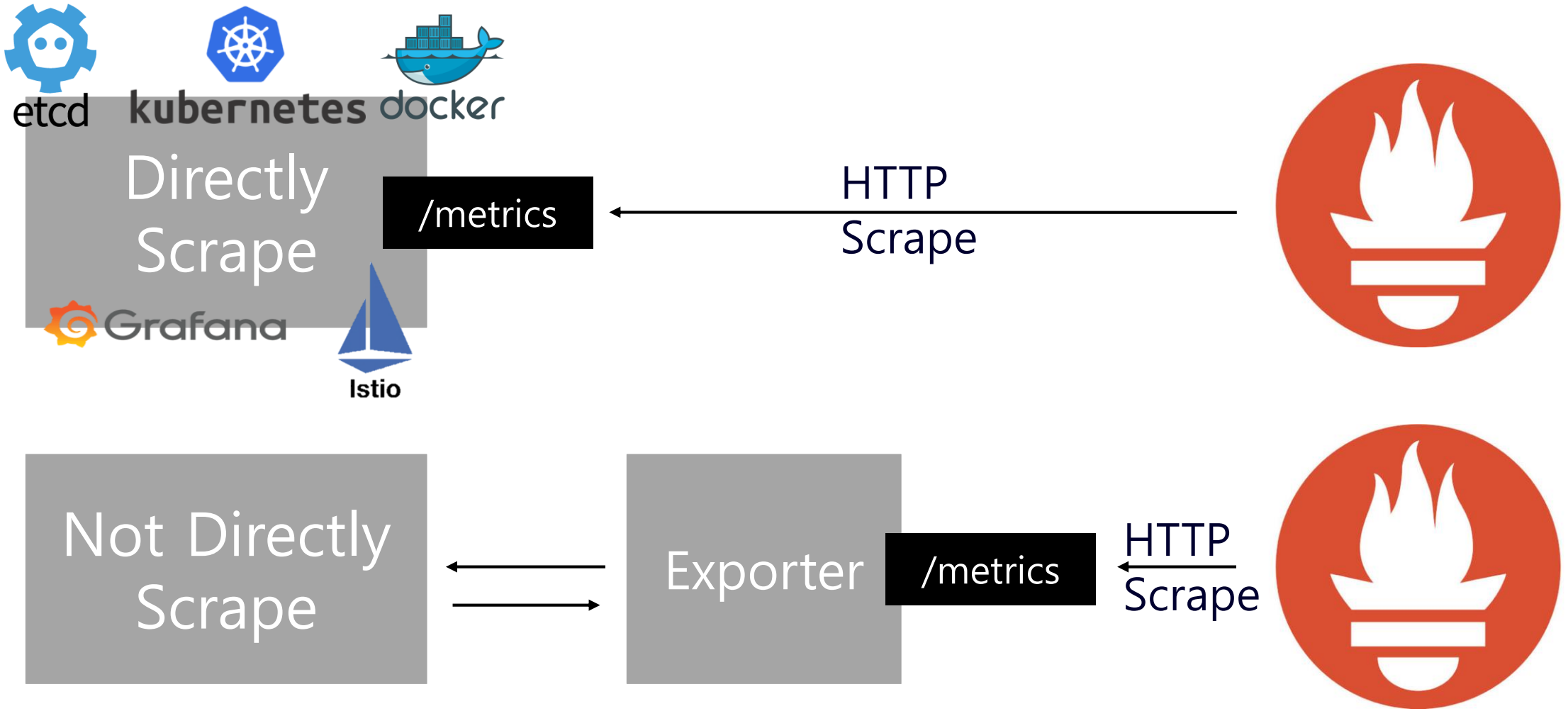


prometheus.yml

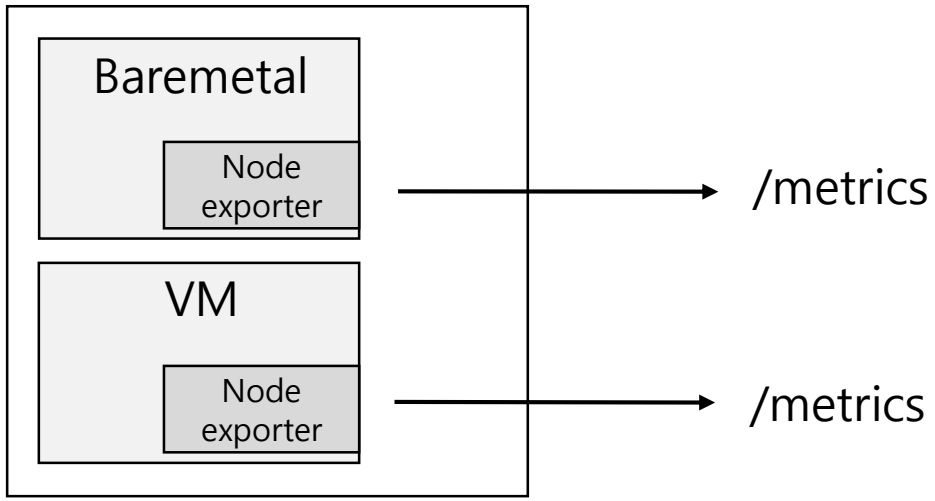
```
global:
  scrape_interval:     15s
  evaluation_interval: 15s
  scrape_timeout: 10s

alerting:
  alertmanagers:
  - static_configs:
    - targets:
       - alertmanager:9093

rule_files:
   - "first_rules.yml"
   - "rules_dir/*.rules"

scrape_configs:
- job_name: 'node'
    static_configs:
      - targets:
        - '192.168.0.100:9100'
        - '192.168.0.101:9100'
```

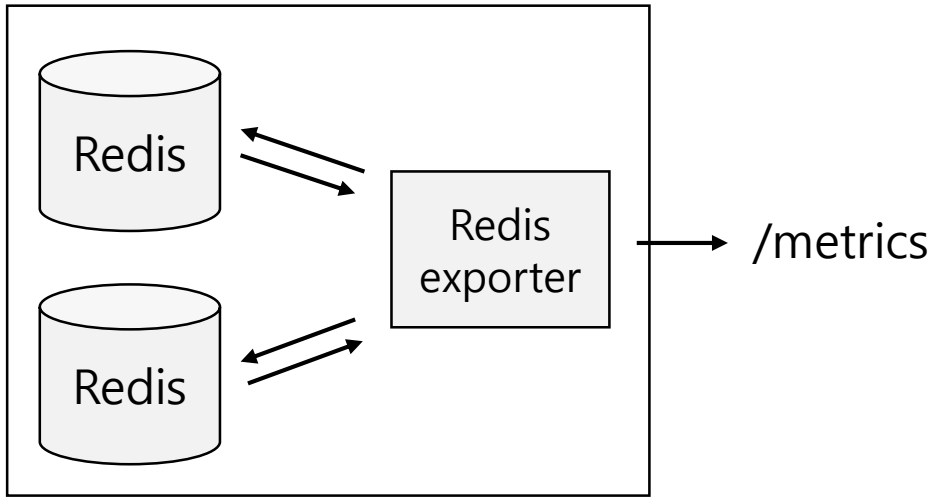
How Prometheus works?

Scrape metric



Exporters





Node-exporter

- Embedded Type
- Hardware and OS metrics 수집 / 노출
- Host CPU, Memory, Disk, Filesystem, vmstat, netstat, iostat, /proc/~

Redis-exporter

- Metric Server Type
- Prometheus exporter for Redis metrics.
- Supports Redis 2.x and 3.x
- 1:N

Exporters List Up

https://prometheus.io/docs/instrumenting/exporters/

Lots of official & 3rd-party exporters

OS – Node Exporter
- Linux, Windows

Orchestrator
- Kube-state-metric, BOSH, CloudFoundry...

Database
- Mysql, Postgres, CouchDB ...

Messaging
- Kafka, RabbiqMQ, NATS...

Logging
- ElasticSearch, Fluentd, Telegraf...

Key-Value
- Redis, Memcached...

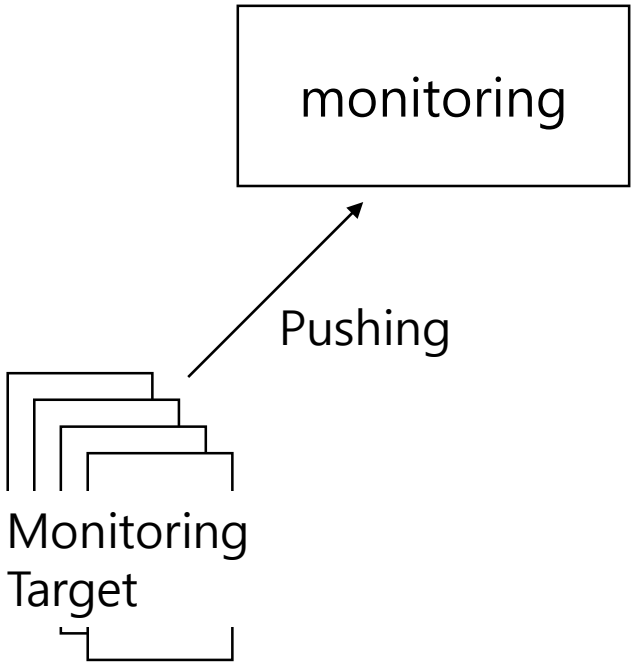
WebServer
- Apache, Nginx...

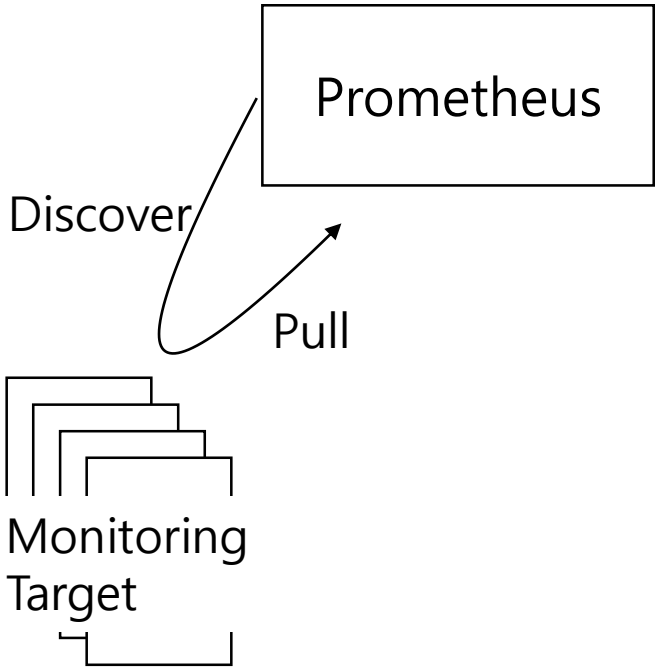
Proxy
- Haproxy, Varnish...

DNS
- BIND, PowerDNS, Unbound

Pull Oriented model

- Metric 수집 대상 Discovery
- Keep/Drop등의 설정에 따라 Target Register, Unregister, Update 수행





Target Service Discovery

Prometheus SD Config

- DNS, File
- Kubernetes
  - ✓ Node, Pod, Service, Ingress, Endpoint
- Openstack
  - ✓ instance
- EC2
- Mesos-Marathon
- Consul, Zookeeper
- …

```
scrape_configs:
- job_name: kubernetes-apiservers
  kubernetes_sd_configs:
  - role: node
  bearer_token_file: xxx
```

```
openstack_sd_configs:
  - identity_endpoint:
    https://openstack.example.com:5000/v2.0
    username: simon
    project_name: prometheus-lab
    password: supersecret
    role: instance
```

```
file_sd_configs:
  - files:
    - 'file-sd.json'
    - 'path/*.yaml'
    refresh_interval: 15s
```

Prometheus Metric & Prom QL

Metric name	Labels	Time	Value
node_cpu_total_seconds	{component="node-exporter",cpu="cpu0",instance="10.178.218.142:9100", mode="system"}	1529219351	65577.99
node_cpu_total_seconds	{component="node-exporter",cpu="cpu0",instance="10.178.218.142:9100", mode="user"}	1529219351	252458.42
node_cpu_total_seconds	{component="node-exporter",cpu="cpu0",instance="10.178.218.142:9100", mode="guest"}	1529219351	0
node_cpu_total_seconds	{component="node-exporter",cpu="cpu0",instance="10.178.218.142:9100", mode="guest_nice"}		1529219351
node_cpu_total_seconds	{component="node-exporter",cpu="cpu0",instance="10.178.218.142:9100", mode="idle"}	1529219351	1726297.47
node_cpu_total_seconds	{component="node-exporter",cpu="cpu0",instance="10.178.218.142:9100", mode="iowait"}	1529219351	22811.51
node_cpu_total_seconds	{component="node-exporter",cpu="cpu0",instance="10.178.218.150:9100", mode="system"}	1529219351	4507.3
node_cpu_total_seconds	{component="node-exporter",cpu="cpu0",instance="10.178.218.150:9100", mode="user"}	1529219351	64458.11
node_cpu_total_seconds	{component="node-exporter",cpu="cpu0",instance="10.178.218.150:9100", mode="guest"}	1529219351	0
node_cpu_total_seconds	{component="node-exporter",cpu="cpu0",instance="10.178.218.150:9100", mode="guest_nice"}		1529219351
node_cpu_total_seconds	{component="node-exporter",cpu="cpu0",instance="10.178.218.150:9100", mode="idle"}	1529219351	1326346.1
node_cpu_total_seconds	{component="node-exporter",cpu="cpu0",instance="10.178.218.150:9100", mode="iowait"}	1529219351	0

```
> node_cpu_total_seconds
```

```
> node_cpu_total_seconds{"instance="10.178.218.142:9100"}
```

```
> rate(node_cpu_total_seconds{"instance="10.178.218.142:9100“, mode="idle"}[5m])
```

```
> avg by (instance) (rate(node_cpu_total_seconds{"instance="10.178.218.142:9100“, mode="idle”}[5m]))
```

Demo

1. Prometheus, Grafana 배포 (Docker)
2. Node Exporter 통한 Metric 수집
3. CPU Usage Visualization (Grafana)
4. File_sd_conifg (Discovery)

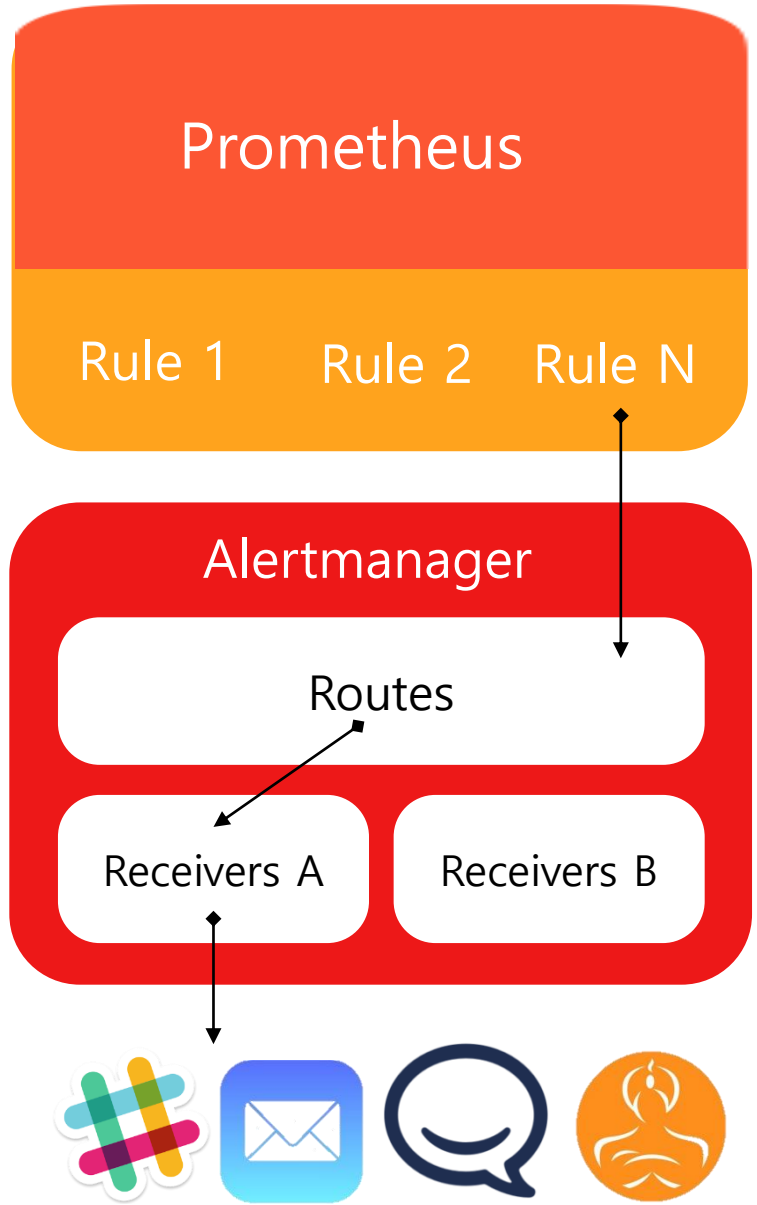
Alerting

Prometheus

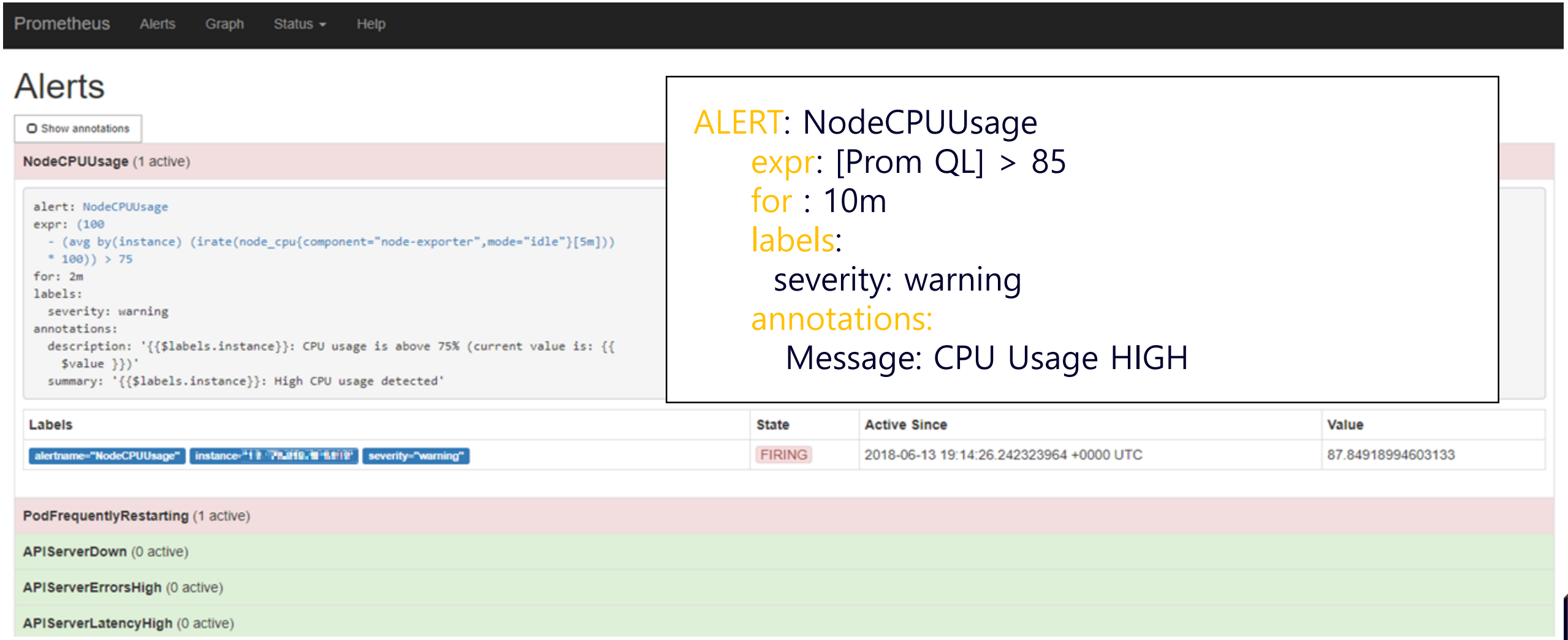
- Alert Rules setting
- Alert Trigger

Alertmanager

- Notification Channel Integration
- Send to Notification Channel
- Alert De-Duplication
- Alert Routing
- Silence

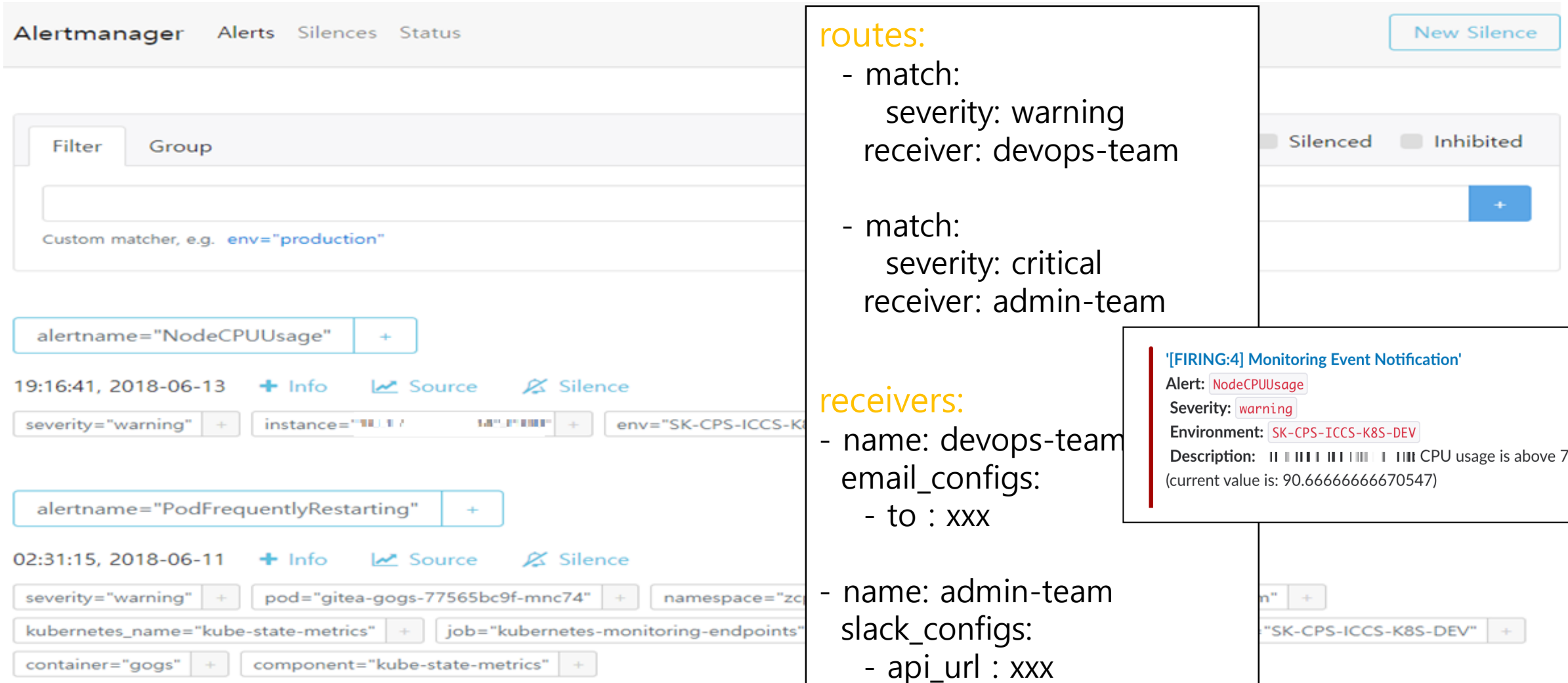


Alert Rules



ALERT: NodeCPUUsage
expr: [Prom QL] > 85
for : 10m
labels:
severity: warning
annotations:
Message: CPU Usage HIGH

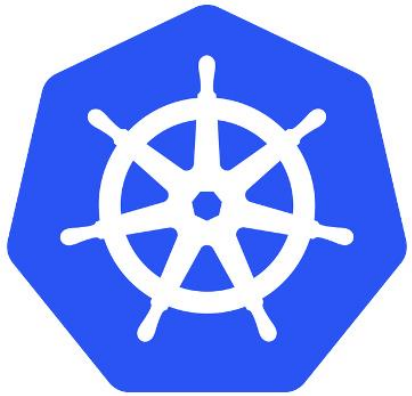
Alertmanager




```
routes:
  - match:
      severity: warning
    receiver: devops-team

  - match:
      severity: critical
    receiver: admin-team

receivers:
- name: devops-team
  email_configs:
    - to : xxx

- name: admin-team
  slack_configs:
    - api_url : xxx
```

'[FIRING:4] Monitoring Event Notification'
Alert: NodeCPUUsage
Severity: warning
Environment: SK-CPS-ICCS-K8S-DEV
Description: CPU usage is above 75%
(current value is: 90.66666666670547)

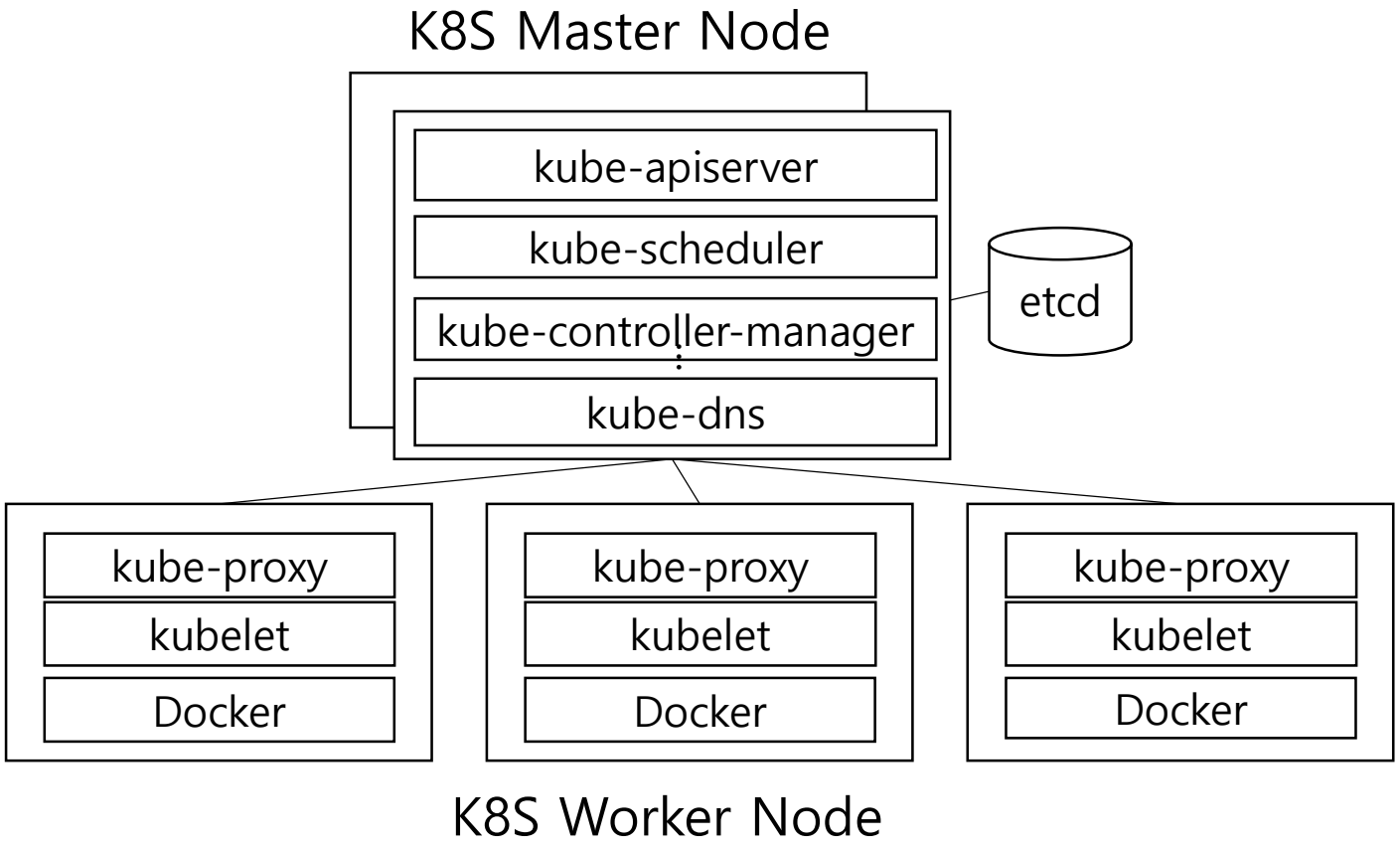
Kubernetes with Prometheus



- Native Monitoring
- 빠르고 작은 규모로 모니터링 시작
- 별도의 복잡한 모니터링 시스템 필요 없음

Kubernetes Components

- All Components Expose Metrics (/metrics)
- Ready to Monitoring with Prometheus



Kubernetes Discovery

Discovery Target

- Nodes
- Pods
- Endpoint/Service
- Ingress

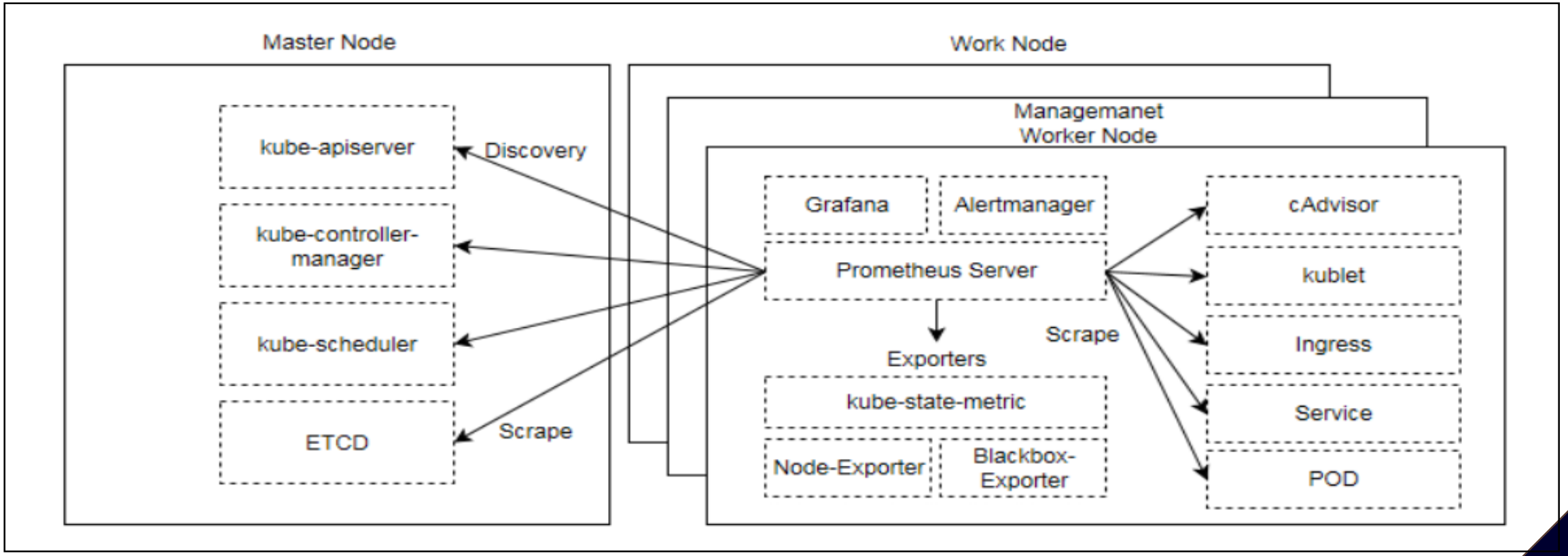
Automations

- Register, Unregister, Update

Kubernetes & Exporter

- Node-Exporter : Master, Worker Node OS Metric
- Kube-state-metric : kubernetes object metric, Cluster state metrics

  (kubectl get ~) node, service, deployment, replicasets, pods, pv, pvc, configmap, quotas, secret, etc



Prometheus Alerts Graph Status Help

Targets

kube-prometheus-exporter-kube-api

Endpoint	State	Labels	Last Scrape	Error
https://172.20.52.160:443/metrics	UP	endpoint="https-metrics" instance="172.20.52.160:443" namespace="kube-system" pod="kube-apiserver-ip-172-20-52-160.eu-west-1.compute.internal" service="kube-prometheus-exporter-kube-api"	9.667s ago	

kube-prometheus-exporter-kube-controller-manager

Endpoint	State	Labels	Last Scrape	Error
http://172.20.52.160:10252/metrics	UP	endpoint="http-metrics" instance="172.20.52.160:10252" namespace="kube-system" pod="kube-controller-manager-ip-172-20-52-160.eu-west-1.compute.internal" service="kube-prometheus-exporter-kube-controller-manager"	7.496s ago	

kube-prometheus-exporter-kube-dns

Endpoint	State	Labels	Last Scrape	Error
http://100.114.221.67:10054/metrics	UP	endpoint="http-metrics-dnsmasq" instance="100.114.221.67:10054" namespace="kube-system" pod="kube-dns-479524115-mpls2" service="kube-prometheus-exporter-kube-dns"	5.105s ago	
http://100.114.221.67:10055/metrics	UP	endpoint="http-metrics-skydns" instance="100.114.221.67:10055" namespace="kube-system" pod="kube-dns-479524115-mpls2" service="kube-prometheus-exporter-kube-dns"	11.565s ago	
http://100.114.237.75:10054/metrics	UP	endpoint="http-metrics-dnsmasq" instance="100.114.237.75:10054" namespace="kube-system" pod="kube-dns-479524115-d0pht" service="kube-prometheus-exporter-kube-dns"	12.443s ago	
http://100.114.237.75:10055/metrics	UP	endpoint="http-metrics-skydns" instance="100.114.237.75:10055" namespace="kube-system" pod="kube-dns-479524115-d0pht" service="kube-prometheus-exporter-kube-dns"	1.027s ago	

kube-prometheus-exporter-kube-etcd

Endpoint	State	Labels	Last Scrape	Error
http://172.20.52.160:4001/metrics	UP	endpoint="http-metrics" instance="172.20.52.160:4001" namespace="kube-system" pod="etcd-server-ip-172-20-52-160.eu-west-1.compute.internal" service="kube-prometheus-exporter-kube-etcd"	11.948s ago	

kube-prometheus-exporter-kube-scheduler

Endpoint	State	Labels	Last Scrape	Error
http://172.20.52.160:10251/metrics	UP	endpoint="http-metrics" instance="172.20.52.160:10251" namespace="kube-system" pod="kube-scheduler-ip-172-20-52-160.eu-west-1.compute.internal" service="kube-prometheus-exporter-kube-scheduler"	9.993s ago	

kube-prometheus-exporter-kube-state

Endpoint	State	Labels	Last Scrape	Error
http://100.114.221.80:8080/metrics	UP	endpoint="kube-state-metrics" instance="100.114.221.80:8080" namespace="monitoring" pod="kube-prometheus-exporter-kube-state-1644029036-g6qc6" service="kube-prometheus-exporter-kube-state"	11.323s ago	
http://100.114.237.82:8080/metrics	UP	endpoint="kube-state-metrics" instance="100.114.237.82:8080" namespace="monitoring" pod="kube-prometheus-exporter-kube-state-2696859725-4p4nl" service="kube-prometheus-exporter-kube-state"	13.274s ago	

kube-prometheus-exporter-node

Endpoint	State	Labels	Last Scrape	Error
http://172.20.52.160:9100/metrics	UP	endpoint="metrics" instance="172.20.52.160:9100" namespace="monitoring" pod="kube-prometheus-exporter-node-mmbnv" service="kube-prometheus-exporter-node"	1.691s ago	
http://172.20.63.18:9100/metrics	UP	endpoint="metrics" instance="172.20.63.18:9100" namespace="monitoring" pod="kube-prometheus-exporter-node-0h292" service="kube-prometheus-exporter-node"	11.739s ago	
http://172.20.84.163:9100/metrics	UP	endpoint="metrics" instance="172.20.84.163:9100" namespace="monitoring" pod="kube-prometheus-exporter-node-7xb78" service="kube-prometheus-exporter-node"	6.366s ago	

Kubernetes Monitoring Demo 영상

1. IBM ICCS Cluster 배포
2. Kubernetes Cluster 상태 점검

2. Prometheus 배포
   - ✓ Target 확인
   - ✓ Metric 수집 확인
   - ✓ Pre-Setup Alert
3. Exporter 배포
   - ✓ Node / kube-state-metric 배포
   - ✓ Metric 수집 확인

4.Grafana 배포
   - ✓ Pre-Setup Dashboard

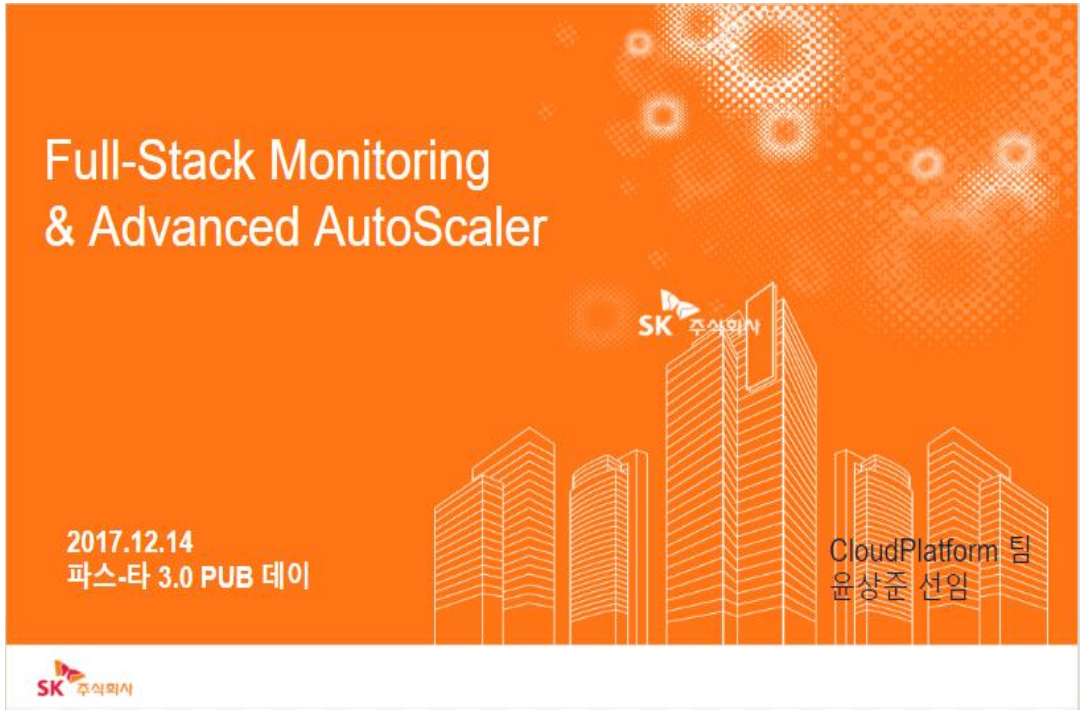
5. Service Discovery
   - ✓ Work Node 추가/삭제
   - ✓ Pod Metric 동적 수집

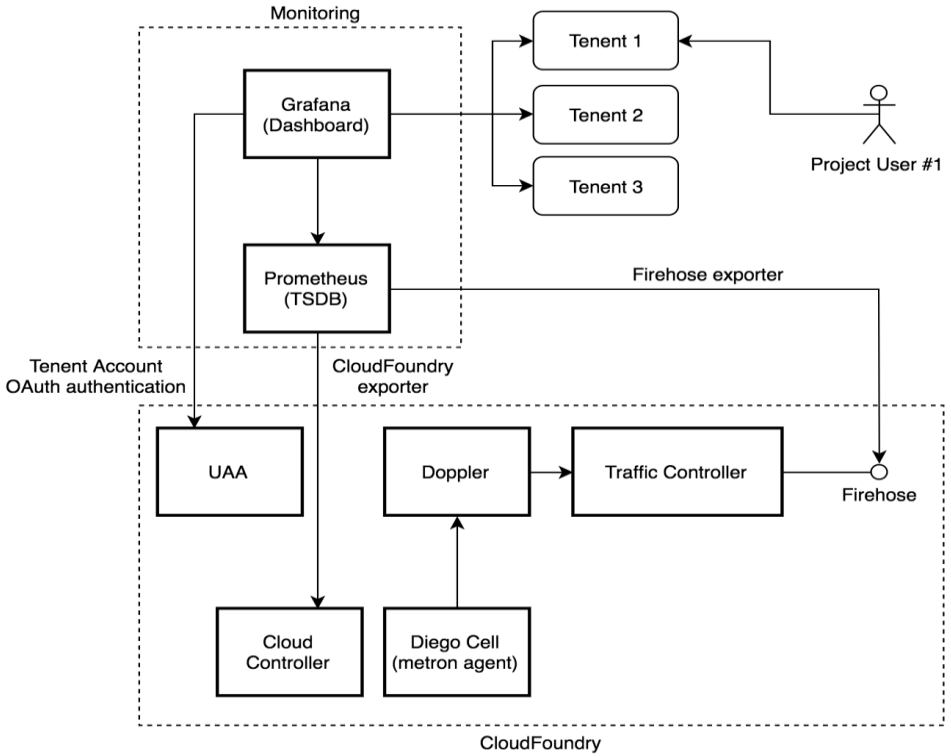
활용 사례 공유

PaaS-TA Monitoring 적용 및 기여

- CloudFoundry 기반 Prometheus Monitoring

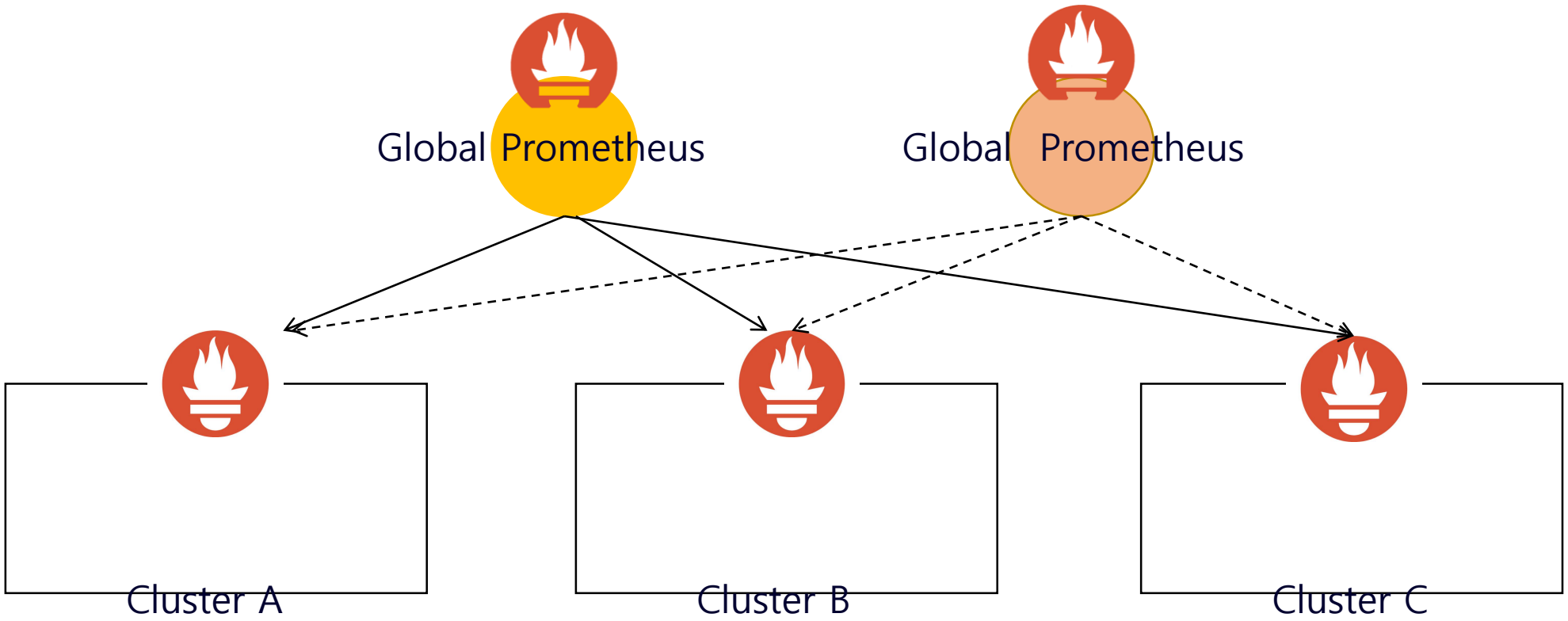
  https://github.com/PaaS-TA-Incubator/OpenPaaS-OWL





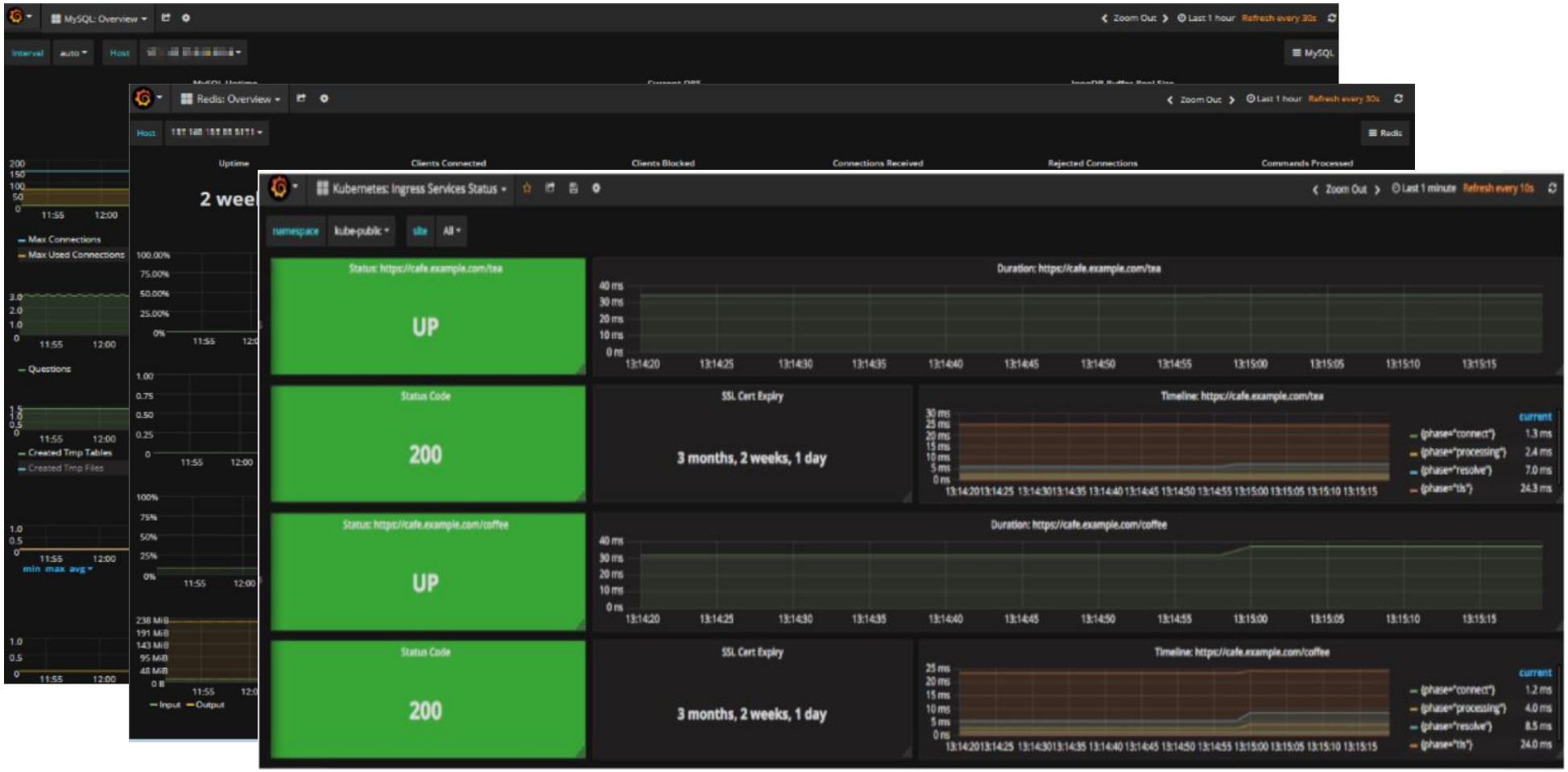
Prometheus Federation

- Kubernetes Cluster 별 Prometheus 배포/관리 수행
- Monitoring 통합 운영 준비



Exporter 확대

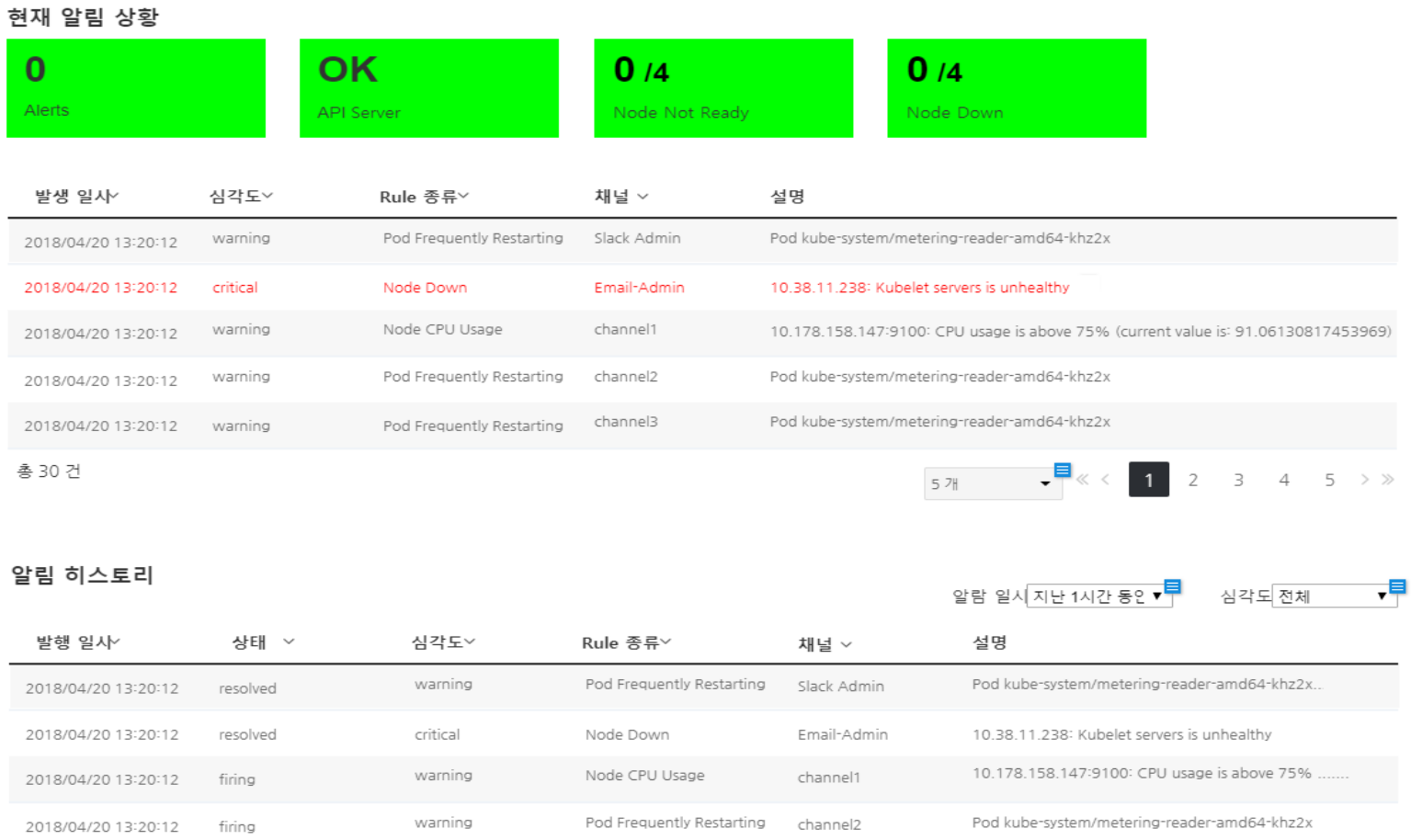
- Metric 확장 및 Monitoring 대상 추가
  (Mysql, Postgres, Redis, ElasticSearch, BlackBox Exporter)



Bosh exporter

Alertmanager 기능 강화

- Kubernetes Alert 설정/관리 기능 제공
- Alert History 기능



Kubernetes Custom Metrics Adaptor

- Prometheus–hpa-adaptor
- Custom Metric 지표 수집 및 HPA Metric 추가

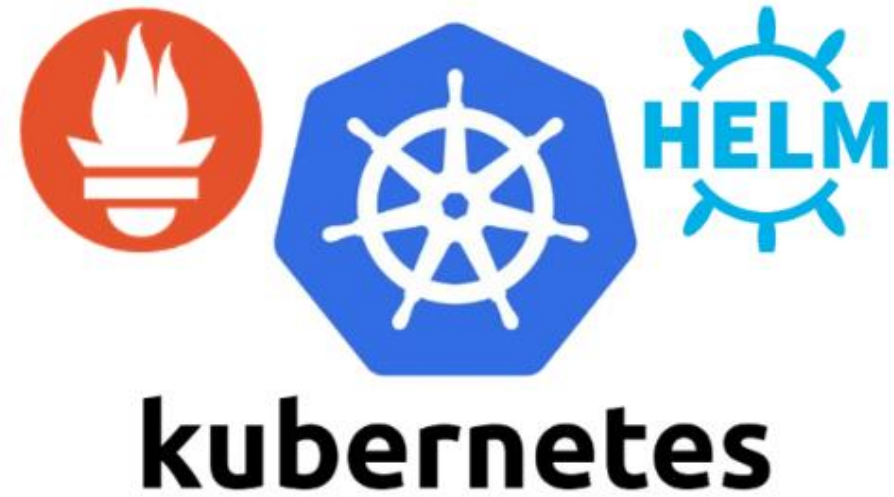
```
root@instance-1:~# kubectl describe hpa -n default
Name:                           spring-music-hpa
Namespace:                      default
Labels:                         <none>
Annotations:                    <none>
CreationTimestamp:              Mon, 11 Jun 2018 08:31:53 +0000
Reference:                      Deployment/spring-music
Metrics:                        ( current / target )
  "tomcat_requestcount" on pods:  0 / 100
Min replicas:                   1
Max replicas:                   10
Conditions:
  Type            Status  Reason            Message
  ----            ------  ------            -------
  AbleToScale     True    ReadyForNewScale  the last scale time was sufficiently old as to warrant a new scale
  ScalingActive   True    ValidMetricFound  the HPA was able to successfully calculate a replica count from pods metric tomcat_requestcount
  ScalingLimited  True    TooFewReplicas    the desired replica count is increasing faster than the maximum scale rate
Events:           <none>
```

요약

- 어렵다... Monitoring , Logging 통합 ? 구분을 짓자 !
- 오픈 인프라/클라우드를 잘~ 지원하는 Prometheus
- 내 입맛에 맞게 모니터링을 작은 규모로 빠르게 시작해보자

Tips

- Helm Chart
- Prometheus Operator



An Operator represents human operational knowledge in software, to reliably manage an application.

CoreOS

감사합니다