

02. SDK LB 2

#0. 4

#1. 가 5

#2. 6

#3. Grafana Import.....17

#4. 41

02. SDK를 활용한 LB 모니터링 시각화

- Grafana 모니터링
 - API를 바로 호출하여 Prometheus 로 저장
 - SDK 를 활용하여 LB API 호출
 - 해당 데이터를 Prometheus 에 저장
 - Grafana 시각화
 - 모니터링 API로 서버에 저장한 데이터를 프로메테우스에서 읽어서 변경
 - API 를 호출하는 별도의 서버에 csv로 로그 저장
 - csv를 Prometheus에서 읽어들이
 - Grafana 시각화

• LB API 리스트

3. list

Name	Type	Description
listloadbalancersresponse	Object	listLoadBalancers 요청에 대한 응답 정보
count	Integer	load balancer 개수
loadbalancer	Array	load balancer 목록

3. 응답

Name	Type	Description
usageloadbalancerservicerresponse	Object	usageLoadBalancerService 요청에 대한 응답 정보
lists	Array	날짜별 사용량
name	String	Load Balancer 이름
outbound	String	outbound 네트워크 트래픽
inbound	String	inbound 네트워크 트래픽
date	String	날짜

Servers in LB : 조회

cursvrconnections	Integer	현재 서버에 연결된 connection 수
state	String	서버 상태
throughputrate	Integer	Throughput 조회 (Mbps)
avgsvrttfb	Integer	TTFB 조회(서버 응답 시간, 단위:msec)
requestsrte	Integer	Request Rate 조회

#0. 전체 구성

1. 시스템 개요

로드밸런서 모니터링 시스템은 다음과 같은 요소들로 구성

[Loadbalancer] → [SDK/API] → [Custom Exporter] → [Prometheus] → [Grafana] → [사용자]

원천 데이터 → 인터페이스 → 변환 → 저장소 → 시각화 → 제공

Custom exporter

- 역할: 로드밸런서 메트릭을 Prometheus 형식으로 변환
- 기능
 - SDK를 통해 로드밸런서 데이터 수집
 - Prometheus 메트릭 형식으로 데이터 변환

Prometheus

- 역할: 시계열 데이터베이스 및 메트릭 수집기
- 기능
 - Exporter로부터 주기적으로 메트릭 수집
 - 알람 규칙 처리

Grafana

- 역할: 데이터 시각화 및 대시보드 제공
- 기능
 - Prometheus 데이터소스 연결
 - 대시보드를 통한 시각적 모니터링

#1. 수집 가능 메트릭

1. list

Name	Type	Description
listloadbalancersresponse	Object	listLoadBalancers 요청에 대한 응답 정보
count	Integer	load balancer 개수
loadbalancer	Array	load balancer 목록

2. 응답

Name	Type	Description
usageloadbalancerserviceresponse	Object	usageLoadBalancerService 요청에 대한 응답 정보
lists	Array	날짜별 사용량
name	String	Load Balancer 이름
outbound	String	outbound 네트워크 트래픽
inbound	String	inbound 네트워크 트래픽
date	String	날짜

3. Servers in LB : 조회

Name	Type	Description
cursrvrconnections	Integer	현재 서버에 연결된 connection 수
state	String	서버 상태
throughputrate	Integer	Throughput 조회 (Mbps)
avgsvrttfb	Integer	TTFB 조회(서버 응답 시간, 단위:msec)
requestsrates	Integer	Request Rate 조회

#2. 설정파일

1. Custom Export설정 [🔗](#)

파일 경로

- SDK와 동일한 경로에 생성

/path/to/site-packages/ktcloud/lb-exporter.py

- 소스코드(주석 포함)

```
1  #!/usr/bin/env python3
2  """
3  kt Cloud LB Exporter
4  """
5
6  import os
7  import time
8  import logging
9  import threading
10 from datetime import datetime
11 from dotenv import load_dotenv
12 import kcldx as kcl
13 from prometheus_client import start_http_server, Gauge, Counter, Info
14 from prometheus_client.core import CollectorRegistry
15 import copy
16
17 # .env 파일에서 환경 변수 로드
18 load_dotenv()
19
20 # 환경 변수에서 KT Cloud 접속 정보 및 설정값 읽기
21 CLOUD_ID = os.getenv("CLOUD_ID")      # KT Cloud 계정 ID
22 CLOUD_PASSWORD = os.getenv("CLOUD_PASSWORD") # KT Cloud 계정 비밀번호
23 CLOUD_ZONE = os.getenv("CLOUD_ZONE")    # KT Cloud 존 정보 (예: DX-M1)
24 EXPORTER_PORT = 9105                    # Prometheus 메트릭 노출 포트
25 SCRAPE_INTERVAL = 60                   # 메트릭 수집 주기 (초)
26
27 # 로깅 설정 - 시간, 로거명, 레벨, 메시지 형태로 출력
28 logging.basicConfig(
29     level=logging.INFO,
30     format='%(asctime)s - %(name)s - %(levelname)s - %(message)s'
31 )
32 logger = logging.getLogger(__name__)
33
34 class AtomicKTCloudLBExporter:
35     """
36     KT Cloud Load Balancer 정보를 수집하여 Prometheus 메트릭으로 노출하는 익스포터 클래스
37
38     원자적 업데이트 방식을 사용하여 Prometheus 스크랩 중 데이터 일관성을 보장합니다.
39     - 데이터 수집과 메트릭 업데이트를 분리
40     - 스레드 안전성 보장
41     - 메트릭 업데이트 중 충돌 감지 및 처리
```

```

42 """
43
44 def __init__(self):
45     """익스포터 초기화 - 메트릭 정의 및 KT Cloud 연결 설정"""
46
47     # Prometheus 메트릭 레지스트리 생성 (독립적인 메트릭 관리)
48     self.registry = CollectorRegistry()
49
50     # 스레드 안전성을 위한 재귀 락 (Reentrant Lock)
51     # 같은 스레드에서 여러 번 락을 획득할 수 있음
52     self.update_lock = threading.RLock()
53
54     # 현재 업데이트 상태 플래그
55     self.is_updating = False
56
57     # 메트릭 저장소 (현재 사용되지 않지만 확장성을 위해 유지)
58     self.current_metrics = {}
59     self.temp_metrics = {}
60
61     # === Prometheus 메트릭 정의 ===
62
63     # 1. 전체 LB 개수
64     self.lb_count = Gauge(
65         'ktcloud_lb_total_count',
66         'Total number of load balancers', # 전체 로드밸런서 개수
67         registry=self.registry
68     )
69
70     # 2. LB 기본 정보 (상태, IP, 포트 등)
71     self.lb_info = Gauge(
72         'ktcloud_lb_info',
73         'Load balancer information (value: 1=UP, 0=DOWN)', # LB 정보 (1=정상, 0=다운)
74         ['lb_id', 'lb_name', 'service_ip', 'service_port', 'service_type',
75          'lb_option', 'healthcheck_type', 'zone'], # 레이블들
76         registry=self.registry
77     )
78
79     # 3. LB별 연결된 서버 개수
80     self.lb_server_count = Gauge(
81         'ktcloud_lb_server_count',
82         'Number of servers connected to load balancer', # LB에 연결된 서버 수
83         ['lb_id', 'lb_name', 'zone'],
84         registry=self.registry
85     )
86
87     # 4. LB 내 개별 서버 상태
88     self.lb_server_state = Gauge(
89         'ktcloud_lb_server_state',
90         'Server state in load balancer (1=UP, 0=DOWN)', # 서버 상태 (1=정상, 0=다운)
91         ['lb_id', 'lb_name', 'server_ip', 'server_port', 'zone'],
92         registry=self.registry
93     )
94
95     # === 서버 성능 메트릭들 ===
96
97     # 5. 서버별 현재 연결 수
98     self.server_connections = Gauge(
99         'ktcloud_server_current_connections',

```

```

100     'Current number of connections to the server', # 서버의 현재 연결 수
101     ['lb_id', 'lb_name', 'server_ip', 'server_port', 'zone'],
102     registry=self.registry
103 )
104
105 # 6. 서버별 처리량 (KB/s)
106 self.server_throughput_rate = Gauge(
107     'ktcloud_server_throughput_rate_kbps',
108     'Server throughput rate in KB/s', # 서버 처리량 (KB/초)
109     ['lb_id', 'lb_name', 'server_ip', 'server_port', 'zone'],
110     registry=self.registry
111 )
112
113 # 7. 서버별 평균 응답 시간 (TTFB - Time To First Byte)
114 self.server_avg_ttfb = Gauge(
115     'ktcloud_server_avg_ttfb_ms',
116     'Average server Time To First Byte in milliseconds', # 평균 첫 바이트 응답 시간 (ms)
117     ['lb_id', 'lb_name', 'server_ip', 'server_port', 'zone'],
118     registry=self.registry
119 )
120
121 # 8. 서버별 초당 요청 처리량
122 self.server_requests_rate = Gauge(
123     'ktcloud_server_requests_rate_per_sec',
124     'Server requests rate per second', # 서버별 초당 요청 수
125     ['lb_id', 'lb_name', 'server_ip', 'server_port', 'zone'],
126     registry=self.registry
127 )
128
129 # 9. 서비스 타입별 LB 개수 (HTTP, HTTPS, TCP 등)
130 self.service_type_count = Gauge(
131     'ktcloud_lb_service_type_count',
132     'Count of load balancers by service type', # 서비스 타입별 LB 개수
133     ['service_type', 'zone'],
134     registry=self.registry
135 )
136
137 # === 익스포터 운영 메트릭들 ===
138
139 # 10. 메트릭 수집 소요 시간
140 self.scrape_duration = Gauge(
141     'ktcloud_lb_scrape_duration_seconds',
142     'Time spent scraping KT Cloud LB metrics', # 메트릭 수집에 걸린 시간 (초)
143     registry=self.registry
144 )
145
146 # 11. 마지막 성공적인 수집 시간
147 self.last_scrape_timestamp = Gauge(
148     'ktcloud_lb_last_scrape_timestamp',
149     'Timestamp of last successful scrape', # 마지막 성공 수집 타임스탬프
150     registry=self.registry
151 )
152
153 # 12. 업데이트 충돌 발생 횟수 (Prometheus 스크랩과 충돌)
154 self.update_conflicts = Counter(
155     'ktcloud_lb_update_conflicts_total',
156     'Number of times Prometheus scraped during metric updates', # 업데이트 중 충돌 횟수
157     registry=self.registry

```



```

158 )
159
160 # 13. 성공적인 원자적 업데이트 횟수
161 self.atomic_updates = Counter(
162     'ktcloud_lb_atomic_updates_total',
163     'Number of successful atomic metric updates', # 성공적인 원자적 업데이트 횟수
164     registry=self.registry
165 )
166
167 # 14. 익스포터 정보 (버전, 설정 등)
168 self.exporter_info = Info(
169     'ktcloud_lb_exporter',
170     'KT Cloud Load Balancer Exporter Information', # 익스포터 정보
171     registry=self.registry
172 )
173
174 # KT Cloud 연결 초기화
175 self.zone_mgr = None # KT Cloud Zone Manager
176 self.network = None # Network Resource Manager
177 self.init_ktcloud_connection()
178
179 def init_ktcloud_connection(self):
180     """
181     KT Cloud 연결 초기화
182     - Zone Manager 생성
183     - Network Resource Manager 생성
184     - 익스포터 정보 설정
185     """
186     try:
187         logger.info("KT Cloud 연결 초기화 중...")
188
189         # KT Cloud Zone Manager 생성 (인증 및 존 관리)
190         self.zone_mgr = kcl.ZoneManager(CLOUD_ID, CLOUD_PASSWORD, CLOUD_ZONE)
191
192         # Network Resource Manager 생성 (LB 관련 API 호출)
193         self.network = self.zone_mgr.network_resource()
194
195         # 존 정보 가져오기
196         zone, zone_name = self.zone_mgr.get_zone()
197
198         # 익스포터 정보 메트릭 설정
199         self.exporter_info.info({
200             'version': '2.6.0', # 익스포터 버전
201             'zone': zone_name, # KT Cloud 존 이름
202             'port': str(EXPORTER_PORT), # 서비스 포트
203             'scrape_interval': str(SCRAPE_INTERVAL), # 수집 주기
204             'data_source': 'KT Cloud SDK Atomic', # 데이터 소스
205             'description': 'Atomic update version to prevent Prometheus scrape conflicts'
206         })
207
208         logger.info(f"KT Cloud 연결 성공 - Zone: {zone_name}")
209
210     except Exception as e:
211         logger.error(f"KT Cloud 연결 실패: {e}")
212         raise
213
214 def collect_data_to_temp(self):
215     """

```

```

216 1단계: 데이터를 임시 저장소에 수집 (메트릭 업데이트 없음)
217
218 KT Cloud API를 호출하여 LB 정보를 수집하지만,
219 Prometheus 메트릭은 업데이트하지 않고 임시 저장소에만 보관합니다.
220 이렇게 하면 데이터 수집 중에 Prometheus가 스크랩해도 일관된 데이터를 읽을 수 있습니다.
221
222 Returns:
223 dict: 수집된 데이터를 담은 딕셔너리
224     - lb_list: LB 목록
225     - lb_servers: LB별 서버 정보
226     - service_type_counts: 서비스 타입별 개수
227     - collection_time: 수집 시간
228 """
229 # 임시 데이터 저장소 초기화
230 temp_data = {
231     'lb_list': [],      # LB 기본 정보 목록
232     'lb_servers': {},   # LB별 서버 상세 정보
233     'service_type_counts': {}, # 서비스 타입별 카운트
234     'collection_time': time.time() # 수집 시작 시간
235 }
236
237 logger.info(f"

```

```

274     logger.info(f"✅ 임시 데이터 수집 완료")
275
276     return temp_data
277
278 def atomic_update_metrics(self, temp_data):
279     """
280     2단계: 수집된 데이터를 원자적으로 메트릭에 업데이트
281
282     임시 저장소의 데이터를 실제 Prometheus 메트릭으로 업데이트합니다.
283     락을 사용하여 업데이트 중에는 다른 스레드가 접근하지 못하도록 하고,
284     모든 메트릭을 한 번에 업데이트하여 일관성을 보장합니다.
285
286     Args:
287         temp_data (dict): collect_data_to_temp()에서 수집된 데이터
288     """
289     update_start = time.time()
290
291     # 락 획득 시도 (non-blocking)
292     if not self.update_lock.acquire(blocking=False):
293         # 락 획득 실패 시 (다른 스레드가 업데이트 중)
294         logger.warning("⚠️ 메트릭 업데이트 중 충돌 감지 - 대기")
295         self.update_conflicts.inc() # 충돌 카운터 증가
296         self.update_lock.acquire() # 블로킹 모드로 대기
297
298     try:
299         self.is_updating = True
300         logger.debug("🔧 메트릭 원자적 업데이트 시작")
301
302         # === 모든 메트릭 초기화 (이전 데이터 제거) ===
303         self.lb_info.clear()
304         self.lb_server_count.clear()
305         self.lb_server_state.clear()
306         self.server_connections.clear()
307         self.server_throughput_rate.clear()
308         self.server_avg_ttfb.clear()
309         self.server_requests_rate.clear()
310         self.service_type_count.clear()
311
312         # === 기본 메트릭 업데이트 ===
313
314         # 전체 LB 개수 설정
315         self.lb_count.set(len(temp_data['lb_list']))
316
317         # 존 정보 가져오기
318         zone, zone_name = self.zone_mgr.get_zone()
319
320         # === LB별 상세 정보 업데이트 ===
321         for lb in temp_data['lb_list']:
322             try:
323                 # LB 기본 정보 추출
324                 lb_id = str(lb['lb_id'])
325                 lb_name = lb['lb_name']
326                 service_ip = lb['service_ip']
327                 service_port = str(lb['service_port'])
328                 service_type = lb['service_type']
329                 lb_option = lb['lb_option'] # 로드밸런싱 알고리즘
330                 healthcheck_type = lb['healthcheck_type'] # 헬스체크 방식

```

```

332     # LB 상태 변환 ('UP' -> 1, 'DOWN' -> 0)
333     state = 1 if lb['state'] == 'UP' else 0
334
335     # LB 기본 정보 메트릭 설정
336     self.lb_info.labels(
337         lb_id=lb_id,
338         lb_name=lb_name,
339         service_ip=service_ip,
340         service_port=service_port,
341         service_type=service_type,
342         lb_option=lb_option,
343         healthcheck_type=healthcheck_type,
344         zone=zone_name
345     ).set(state)
346
347     # 해당 LB의 서버 정보 처리
348     servers = temp_data['lb_servers'].get(lb['lb_id'], [])
349     server_count = len(servers)
350
351     # LB별 연결된 서버 개수 설정
352     self.lb_server_count.labels(
353         lb_id=lb_id,
354         lb_name=lb_name,
355         zone=zone_name
356     ).set(server_count)
357
358     # === 서버별 상세 정보 처리 ===
359     for server in servers:
360         try:
361             # 서버 기본 정보 추출
362             server_ip = server['vm_ip']
363             server_port = str(server['vm_port'])
364
365             # 서버 상태 변환 ('UP' -> 1, 'DOWN' -> 0)
366             server_state = 1 if server['state'] == 'UP' else 0
367
368             # 서버 상태 메트릭 설정
369             self.lb_server_state.labels(
370                 lb_id=lb_id,
371                 lb_name=lb_name,
372                 server_ip=server_ip,
373                 server_port=server_port,
374                 zone=zone_name
375             ).set(server_state)
376
377             # === 성능 메트릭 처리 ===
378
379             def safe_float(value):
380                 """
381                 안전한 float 변환 함수
382                 None, 빈 문자열, 잘못된 형식의 값을 0으로 처리
383                 """
384                 try:
385                     return float(value) if value is not None and value != "" else 0
386                 except (ValueError, TypeError):
387                     return 0
388
389             # 성능 지표 추출 및 변환

```

```

390         connections = safe_float(server.get('cursvrconnections', 0)) # 현재 연결 수
391         throughput = safe_float(server.get('throughputrate', 0))    # 처리량 (KB/s)
392         ttfb = safe_float(server.get('avgsrvttfb', 0))              # 평균 TTFB (ms)
393         requests = safe_float(server.get('requestsrate', 0))       # 초당 요청 수
394
395         # 성능 메트릭들 설정
396         self.server_connections.labels(
397             lb_id=lb_id, lb_name=lb_name,
398             server_ip=server_ip, server_port=server_port,
399             zone=zone_name
400         ).set(connections)
401
402         self.server_throughput_rate.labels(
403             lb_id=lb_id, lb_name=lb_name,
404             server_ip=server_ip, server_port=server_port,
405             zone=zone_name
406         ).set(throughput)
407
408         self.server_avg_ttfb.labels(
409             lb_id=lb_id, lb_name=lb_name,
410             server_ip=server_ip, server_port=server_port,
411             zone=zone_name
412         ).set(ttfb)
413
414         self.server_requests_rate.labels(
415             lb_id=lb_id, lb_name=lb_name,
416             server_ip=server_ip, server_port=server_port,
417             zone=zone_name
418         ).set(requests)
419
420     except Exception as e:
421         logger.error(f"서버 {server.get('vm_ip', 'Unknown')} 메트릭 설정 실패: {e}")
422
423     except Exception as e:
424         logger.error(f"LB {lb.get('lb_name', 'Unknown')} 메트릭 설정 실패: {e}")
425
426     # === 서비스 타입별 카운트 메트릭 설정 ===
427     for service_type, count in temp_data['service_type_counts'].items():
428         self.service_type_count.labels(
429             service_type=service_type,
430             zone=zone_name
431         ).set(count)
432
433     # === 익스포터 상태 메트릭 업데이트 ===
434
435     # 마지막 성공적인 수집 시간 기록
436     self.last_scrape_timestamp.set(temp_data['collection_time'])
437
438     # 성공적인 원자적 업데이트 카운터 증가
439     self.atomic_updates.inc()
440
441     # 업데이트 소요 시간 계산 및 로깅
442     update_duration = time.time() - update_start
443     logger.info(f"🔒 원자적 업데이트 완료 - 소요시간: {update_duration:.3f}초")
444
445     finally:
446         # 락 해제 및 상태 플래그 초기화 (예외 발생 시에도 실행됨)
447         self.is_updating = False

```

```

448         self.update_lock.release()
449
450     def collect_metrics(self):
451         """
452         메트릭 수집 메인 함수 - 2단계 프로세스로 구성
453
454         1단계: 데이터 수집 (KT Cloud API 호출)
455         2단계: 원자적 메트릭 업데이트 (Prometheus 메트릭 갱신)
456
457         이렇게 분리함으로써 Prometheus 스크랩 중에도 일관된 데이터를 제공할 수 있습니다.
458         """
459         start_time = time.time()
460
461         try:
462             logger.info("🚀 메트릭 수집 시작")
463
464             # 1단계: 데이터 수집 (메트릭 업데이트 없음)
465             temp_data = self.collect_data_to_temp()
466
467             # LB가 없는 경우 처리
468             if not temp_data['lb_list']:
469                 self.lb_count.set(0)
470             return
471
472             collection_duration = time.time() - start_time
473             logger.info(f"🇰🇷 데이터 수집 완료 - 소요시간: {collection_duration:.3f}초")
474
475             # 2단계: 원자적 메트릭 업데이트
476             self.atomic_update_metrics(temp_data)
477
478             # 전체 소요시간 기록 (메트릭으로 노출)
479             total_duration = time.time() - start_time
480             self.scrape_duration.set(total_duration)
481
482             logger.info(f"✅ 전체 메트릭 처리 완료 - 총 소요시간: {total_duration:.3f}초")
483
484         except Exception as e:
485             logger.error(f"❌ 메트릭 수집 중 오류: {e}")
486             raise
487
488     def run(self):
489         """
490         익스포터 메인 실행 루프
491
492         - Prometheus HTTP 서버 시작
493         - 주기적으로 메트릭 수집 및 업데이트
494         - 예외 처리 및 재시도 로직
495         """
496         logger.info(f"🚀 KT Cloud LB Exporter (Atomic Version) 시작 - 포트: {EXPORTER_PORT}")
497         logger.info("원자적 메트릭 업데이트로 Prometheus 스크랩 충돌 방지")
498
499         # Prometheus HTTP 서버 시작 (메트릭 노출용)
500         start_http_server(EXPORTER_PORT, registry=self.registry)
501         logger.info(f"🌐 메트릭 서버 시작: http://localhost:{EXPORTER_PORT}/metrics")
502
503         # 메인 실행 루프
504         while True:
505             try:

```

```

506     cycle_start = time.time()
507
508     # 메트릭 수집 실행
509     self.collect_metrics()
510
511     # 다음 수집까지의 대기 시간 계산
512     elapsed = time.time() - cycle_start
513     sleep_time = max(0, SCRAPE_INTERVAL - elapsed)
514
515     if sleep_time > 0:
516         logger.info(f"🔋 다음 수집까지 {sleep_time:.1f}초 대기")
517         time.sleep(sleep_time)
518     else:
519         # 수집 시간이 설정된 주기를 초과한 경우 경고
520         logger.warning(f"⚠️ 수집 시간이 수집 주기를 초과: {elapsed:.1f}초")
521
522     except KeyboardInterrupt:
523         # 사용자가 Ctrl+C로 중단한 경우
524         logger.info("👋 사용자 중단 요청")
525         break
526     except Exception as e:
527         # 예상치 못한 오류 발생 시 10초 후 재시도
528         logger.error(f"💣 예상치 못한 오류: {e}")
529         logger.info("🔄 10초 후 재시도...")
530         time.sleep(10)
531
532 def main():
533     """
534     메인 함수 - 프로그램 진입점
535
536     환경 변수 검증 후 익스포터 실행
537     """
538     # 필수 환경 변수 존재 여부 확인
539     if not all([CLOUD_ID, CLOUD_PASSWORD, CLOUD_ZONE]):
540         logger.error("❌ 환경 변수 누락: CLOUD_ID, CLOUD_PASSWORD, CLOUD_ZONE")
541         return
542
543     # 익스포터 인스턴스 생성 및 실행
544     exporter = AtomicKTCLOUDLBExporter()
545     exporter.run()
546
547     # 스크립트가 직접 실행될 때만 main() 함수 호출
548     if __name__ == "__main__":
549         main()

```

- 이후 서비스를 등록하여 실행하거나 직접 pytho나파일을 동작

2. Prometheus 설정 [🔗](#)

파일 경로

/prometheus/prometheus-3.1.0.linux-amd64/prometheus.yml

```

1 # my global config
2 global:
3     scrape_interval: 5s # Set the scrape interval to every 15 seconds. Default is every 1 minute.

```

```

4  evaluation_interval: 5s # Evaluate rules every 15 seconds. The default is every 1 minute.
5  # scrape_timeout is set to the global default (10s).
6  #query:
7  #lookback-delta: 30s
8  # Alertmanager configuration
9  #alerting:
10 # alertmanagers:
11 #   - static_configs:
12 #     - targets:
13 #       - localhost:9093
14 # Load rules once and periodically evaluate them according to the global 'evaluation_interval'.
15 rule_files:
16 # - "first_rules.yml"
17 # - "second_rules.yml"
18 # - "/prometheus/prometheus-3.1.0.linux-amd64/rules/*.yml"
19 # A scrape configuration containing exactly one endpoint to scrape:
20 # Here it's Prometheus itself.
21 scrape_configs:
22 # The job name is added as a label `job=<job_name>` to any timeseries scraped from this config.
23 - job_name: "prometheus"
24   # metrics_path defaults to '/metrics'
25   # scheme defaults to 'http'.
26   static_configs:
27     - targets: ["localhost:9090"]
28   # KT Cloud Load Balancer 익스포터
29 - job_name: 'ktcloud-lb'
30   static_configs:
31     - targets: ['localhost:9105']
32     labels:
33       environment: 'production'
34       service: 'ktcloud-lb'
35   scrape_interval: 60s
36   scrape_timeout: 20s
37   metrics_path: /metrics
38   honor_labels: true

```


#3. Grafana Import

1. JSON파일 [🔗](#)

- Grafana 대시보드의 모든 설정(패널, 쿼리, 레이아웃 등)을 저장한 텍스트 파일

```
1 {
2   "__inputs": [
3     {
4       "name": "k1c_PROMETHEUS",
5       "label": "Prometheus",
6       "description": "",
7       "type": "datasource",
8       "pluginId": "prometheus",
9       "pluginName": "Prometheus"
10    }
11  ],
12  "__requires": [
13    {
14      "type": "panel",
15      "id": "stat",
16      "name": "Stat",
17      "version": ""
18    },
19    {
20      "type": "panel",
21      "id": "timeseries",
22      "name": "Time series",
23      "version": ""
24    },
25    {
26      "type": "panel",
27      "id": "table",
28      "name": "Table",
29      "version": ""
30    },
31    {
32      "type": "panel",
33      "id": "piechart",
34      "name": "Pie chart",
35      "version": ""
36    },
37    {
38      "type": "grafana",
39      "id": "grafana",
40      "name": "Grafana",
41      "version": "11.4.0"
42    },
43    {
44      "type": "datasource",
45      "id": "prometheus",
46      "name": "Prometheus",
47      "version": "3.1.0"
48    }
49  ]
50 }
```

```

49 ],
50 "annotations": {
51   "list": [
52     {
53       "builtIn": 1,
54       "datasource": {
55         "type": "grafana",
56         "uid": "-- Grafana --"
57       },
58       "enable": true,
59       "hide": true,
60       "iconColor": "rgba(0, 211, 255, 1)",
61       "name": "Annotations & Alerts",
62       "type": "dashboard"
63     }
64   ]
65 },
66 "editable": true,
67 "fiscalYearStartMonth": 0,
68 "graphTooltip": 0,
69 "id": null,
70 "links": [],
71 "panels": [
72   {
73     "collapsed": false,
74     "gridPos": {
75       "h": 1,
76       "w": 24,
77       "x": 0,
78       "y": 0
79     },
80     "id": 1,
81     "panels": [],
82     "title": "전체 Loadbalacer 현황",
83     "type": "row"
84   },
85   {
86     "datasource": "${ktc_PROMETHEUS}",
87     "fieldConfig": {
88       "defaults": {
89         "color": {
90           "mode": "thresholds"
91         },
92         "mappings": [],
93         "thresholds": {
94           "mode": "absolute",
95           "steps": [
96             {
97               "color": "green",
98               "value": null
99             },
100            {
101              "color": "yellow",
102              "value": 5
103            },
104            {
105              "color": "red",
106              "value": 10

```

```

107     }
108   ]
109 },
110   "unit": "short"
111 },
112   "overrides": []
113 },
114   "gridPos": {
115     "h": 3,
116     "w": 5,
117     "x": 0,
118     "y": 1
119   },
120   "id": 2,
121   "options": {
122     "colorMode": "value",
123     "graphMode": "area",
124     "justifyMode": "auto",
125     "orientation": "auto",
126     "reduceOptions": {
127       "calcs": [
128         "lastNotNull"
129       ],
130       "fields": "",
131       "values": false
132     },
133     "text": {
134       "titleSize": 20,
135       "valueSize": 48
136     },
137     "textMode": "auto"
138   },
139   "pluginVersion": "11.4.0",
140   "targets": [
141     {
142       "datasource": "${ktc_PROMETHEUS}",
143       "expr": "ktcloud_lb_total_count",
144       "interval": "",
145       "legendFormat": "",
146       "refId": "A"
147     }
148   ],
149   "title": "총 LB 수",
150   "type": "stat"
151 },
152 {
153   "datasource": "${ktc_PROMETHEUS}",
154   "fieldConfig": {
155     "defaults": {
156       "color": {
157         "mode": "thresholds"
158       },
159       "mappings": [],
160       "thresholds": {
161         "mode": "absolute",
162         "steps": [
163           {
164             "color": "red",

```

```

165         "value": null
166     },
167     {
168         "color": "yellow",
169         "value": 1
170     },
171     {
172         "color": "green",
173         "value": 2
174     }
175 ]
176 },
177 "unit": "short"
178 },
179 "overrides": []
180 },
181 "gridPos": {
182     "h": 3,
183     "w": 2,
184     "x": 5,
185     "y": 1
186 },
187 "id": 3,
188 "options": {
189     "colorMode": "value",
190     "graphMode": "area",
191     "justifyMode": "auto",
192     "orientation": "auto",
193     "reduceOptions": {
194         "calcs": [
195             "lastNotNull"
196         ],
197         "fields": "",
198         "values": false
199     },
200     "text": {},
201     "textMode": "auto"
202 },
203 "pluginVersion": "11.4.0",
204 "targets": [
205     {
206         "datasource": "${ktc_PROMETHEUS}",
207         "expr": "sum(kcloud_lb_info)",
208         "interval": "",
209         "legendFormat": "",
210         "refId": "A"
211     }
212 ],
213 "title": "정상 LB 수",
214 "type": "stat"
215 },
216
217 {
218     "datasource": "${ktc_PROMETHEUS}",
219     "fieldConfig": {
220         "defaults": {
221             "color": {
222                 "mode": "thresholds"

```

```

223     },
224     "mappings": [],
225     "thresholds": {
226         "mode": "absolute",
227         "steps": [
228             {
229                 "color": "green",
230                 "value": null
231             },
232             {
233                 "color": "yellow",
234                 "value": 1
235             },
236             {
237                 "color": "red",
238                 "value": 2
239             }
240         ]
241     },
242     "unit": "short"
243 },
244 "overrides": []
245 },
246 "gridPos": {
247     "h": 3,
248     "w": 2,
249     "x": 7,
250     "y": 1
251 },
252 "id": 4,
253 "options": {
254     "colorMode": "value",
255     "graphMode": "area",
256     "justifyMode": "auto",
257     "orientation": "auto",
258     "reduceOptions": {
259         "calcs": [
260             "lastNotNull"
261         ],
262         "fields": "",
263         "values": false
264     },
265     "text": {},
266     "textMode": "auto"
267 },
268 "pluginVersion": "11.4.0",
269 "targets": [
270     {
271         "datasource": "${kvc_PROMETHEUS}",
272         "expr": "count(ktccloud_lb_info == 0)",
273         "interval": "",
274         "legendFormat": "",
275         "refId": "A"
276     }
277 ],
278 "title": "다운 LB 수",
279 "type": "stat"
280 },

```

```

281
282
283 {
284   "datasource": "${ktc_PROMETHEUS}",
285   "fieldConfig": {
286     "defaults": {
287       "color": {
288         "mode": "palette-classic"
289       },
290       "custom": {
291         "hideFrom": {
292           "legend": false,
293           "tooltip": false,
294           "vis": false
295         }
296       },
297       "mappings": []
298     },
299     "overrides": []
300   },
301   "gridPos": {
302     "h": 3,
303     "w": 5,
304     "x": 9,
305     "y": 1
306   },
307   "id": 5,
308   "options": {
309     "reduceOptions": {
310       "values": false,
311       "calcs": [
312         "lastNotNull"
313       ],
314       "fields": ""
315     },
316     "pieType": "pie",
317     "tooltip": {
318       "mode": "single",
319       "sort": "none"
320     },
321     "legend": {
322       "displayMode": "list",
323       "placement": "right",
324       "values": [
325         "value"
326       ]
327     }
328   },
329   "targets": [
330     {
331       "datasource": "${ktc_PROMETHEUS}",
332       "expr": "ktcloud_lb_service_type_count",
333       "interval": "",
334       "legendFormat": "{{service_type}}",
335       "refId": "A"
336     }
337   ],
338   "title": "서비스 타입별 로드밸런서",

```

```

339     "type": "piechart"
340 },
341 {
342     "datasource": "${kvc_PROMETHEUS}",
343     "fieldConfig": {
344         "defaults": {
345             "color": {
346                 "mode": "palette-classic"
347             },
348             "custom": {
349                 "axisLabel": "",
350                 "axisPlacement": "auto",
351                 "barAlignment": 0,
352                 "drawStyle": "line",
353                 "fillOpacity": 10,
354                 "gradientMode": "none",
355                 "hideFrom": {
356                     "legend": false,
357                     "tooltip": false,
358                     "vis": false
359                 },
360                 "lineInterpolation": "linear",
361                 "lineWidth": 2,
362                 "pointSize": 5,
363                 "scaleDistribution": {
364                     "type": "linear"
365                 },
366                 "showPoints": "never",
367                 "spanNulls": false,
368                 "stacking": {
369                     "group": "A",
370                     "mode": "none"
371                 },
372                 "thresholdsStyle": {
373                     "mode": "off"
374                 }
375             },
376             "mappings": [],
377             "thresholds": {
378                 "mode": "absolute",
379                 "steps": [
380                     {
381                         "color": "green",
382                         "value": null
383                     }
384                 ]
385             },
386             "unit": "short"
387         },
388         "overrides": []
389     },
390     "gridPos": {
391         "h": 11,
392         "w": 10,
393         "x": 14,
394         "y": 1
395     },
396     "id": 6,

```

```

397     "options": {
398         "legend": {
399             "calcs": [],
400             "displayMode": "list",
401             "placement": "bottom"
402         },
403         "tooltip": {
404             "mode": "multi",
405             "sort": "desc"
406         },
407         "cursor": {
408             "sync": "crosshair"
409         }
410     },
411     "targets": [
412         {
413             "datasource": "${ktc_PROMETHEUS}",
414             "expr": "ktcloud_lb_server_count{lb_name=~\"$lb_name\"}",
415             "interval": "",
416             "legendFormat": "{{lb_name}} - 연결된 서버 수",
417             "refId": "A"
418         }
419     ],
420     "title": "Load Balancer별 연결된 서버 수",
421     "type": "timeseries"
422 },
423
424 {
425     "datasource": "${ktc_PROMETHEUS}",
426     "fieldConfig": {
427         "defaults": {
428             "color": {
429                 "mode": "thresholds"
430             },
431             "custom": {
432                 "align": "center",
433                 "displayMode": "auto",
434                 "filterable": true,
435                 "inspect": false
436             },
437             "mappings": [
438                 {
439                     "options": {
440                         "0": {
441                             "color": "#F2495C",
442                             "text": "DOWN"
443                         },
444                         "1": {
445                             "color": "#73BF69",
446                             "text": "UP"
447                         }
448                     },
449                     "type": "value"
450                 }
451             ],
452             "thresholds": {
453                 "mode": "absolute",
454                 "steps": [

```



```

455     {
456         "color": "green",
457         "value": null
458     }
459 ]
460 }
461 },
462 "overrides": [
463     {
464         "matcher": {
465             "id": "byName",
466             "options": "lb_name"
467         },
468         "properties": [
469             {
470                 "id": "displayName",
471                 "value": "LB 이름"
472             },
473             {
474                 "id": "custom.width",
475                 "value": 200
476             }
477         ]
478     },
479     {
480         "matcher": {
481             "id": "byName",
482             "options": "상태"
483         },
484         "properties": [
485             {
486                 "id": "custom.width",
487                 "value": 80
488             },
489             {
490                 "id": "custom.displayMode",
491                 "value": "color-text"
492             },
493             {
494                 "id": "custom.cellOptions",
495                 "value": {
496                     "type": "color-text"
497                 }
498             }
499         ]
500     },
501     {
502         "matcher": {
503             "id": "byName",
504             "options": "service_ip"
505         },
506         "properties": [
507             {
508                 "id": "displayName",
509                 "value": "LB IP"
510             },
511             {
512                 "id": "custom.width",

```

```

513         "value": 140
514     }
515 ]
516 },
517 {
518     "matcher": {
519         "id": "byName",
520         "options": "service_port"
521     },
522     "properties": [
523         {
524             "id": "displayName",
525             "value": "서비스 포트"
526         },
527         {
528             "id": "custom.width",
529             "value": 100
530         }
531     ]
532 },
533 {
534     "matcher": {
535         "id": "byName",
536         "options": "service_type"
537     },
538     "properties": [
539         {
540             "id": "displayName",
541             "value": "서비스 타입"
542         },
543         {
544             "id": "custom.width",
545             "value": 100
546         }
547     ]
548 },
549 {
550     "matcher": {
551         "id": "byName",
552         "options": "lb_option"
553     },
554     "properties": [
555         {
556             "id": "displayName",
557             "value": "LB 옵션"
558         },
559         {
560             "id": "custom.width",
561             "value": 200
562         }
563     ]
564 },
565 {
566     "matcher": {
567         "id": "byName",
568         "options": "healthcheck_type"
569     },
570     "properties": [

```

```

571     {
572         "id": "displayName",
573         "value": "헬스체크"
574     },
575     {
576         "id": "custom.width",
577         "value": 110
578     }
579 ]
580 }
581 ]
582 },
583 "gridPos": {
584     "h": 8,
585     "w": 14,
586     "x": 0,
587     "y": 4
588 },
589 "id": 7,
590 "options": {
591     "footer": {
592         "enablePagination": true,
593         "fields": "",
594         "reducer": [],
595         "show": false,
596         "countRows": false
597     },
598     "showHeader": true,
599     "sortBy": [
600         {
601             "desc": false,
602             "displayName": "상태"
603         }
604     ]
605 },
606 "pluginVersion": "11.4.0",
607 "targets": [
608     {
609         "datasource": "${ktc_PROMETHEUS}",
610         "expr": "ktcloud_lb_info",
611         "format": "table",
612         "instant": true,
613         "interval": "",
614         "legendFormat": "",
615         "refId": "A"
616     }
617 ],
618 "title": "Load Balancer 현황",
619 "transformations": [
620     {
621         "id": "organize",
622         "options": {
623             "excludeByName": {
624                 "__name__": true,
625                 "Time": true,
626                 "job": true,
627                 "instance": true,
628                 "lb_id": true,

```

```

629     "zone": true,
630     "environment": true,
631     "service": true,
632     "established_connections": true
633 },
634 "indexByName": {
635     "lb_name": 0,
636     "Value": 1,
637     "service_ip": 2,
638     "service_port": 3,
639     "service_type": 4,
640     "lb_option": 5,
641     "healthcheck_type": 6
642 },
643 "renameByName": {
644     "Value": "상태"
645 }
646 }
647 }
648 ],
649 "type": "table"
650 },
651
652
653
654 {
655     "collapsed": false,
656     "gridPos": {
657         "h": 1,
658         "w": 24,
659         "x": 0,
660         "y": 12
661     },
662     "id": 10,
663     "panels": [],
664     "title": "Server in LB",
665     "type": "row"
666 },
667
668 {
669     "datasource": "${ktc_PROMETHEUS}",
670     "fieldConfig": {
671         "defaults": {
672             "color": {
673                 "mode": "thresholds"
674             },
675             "custom": {
676                 "align": "center",
677                 "displayMode": "auto",
678                 "filterable": true,
679                 "inspect": false
680             },
681             "mappings": [
682                 {
683                     "options": {
684                         "0": {
685                             "color": "#F2495C",
686                             "text": "DOWN"

```

```

687     },
688     "1": {
689         "color": "#73BF69",
690         "text": "UP"
691     }
692 },
693 "type": "value"
694 }
695 ],
696 "thresholds": {
697     "mode": "absolute",
698     "steps": [
699         {
700             "color": "green",
701             "value": null
702         }
703     ]
704 }
705 },
706 "overrides": [
707     {
708         "matcher": {
709             "id": "byName",
710             "options": "lb_name"
711         },
712         "properties": [
713             {
714                 "id": "displayName",
715                 "value": "LB 이름"
716             },
717             {
718                 "id": "custom.width",
719                 "value": 180
720             }
721         ]
722     },
723     {
724         "matcher": {
725             "id": "byName",
726             "options": "server_ip"
727         },
728         "properties": [
729             {
730                 "id": "displayName",
731                 "value": "서버 IP"
732             },
733             {
734                 "id": "custom.width",
735                 "value": 140
736             }
737         ]
738     },
739     {
740         "matcher": {
741             "id": "byName",
742             "options": "server_port"
743         },
744         "properties": [

```

```

745     {
746         "id": "displayName",
747         "value": "서버 포트"
748     },
749     {
750         "id": "custom.width",
751         "value": 100
752     }
753 ]
754 },
755 {
756     "matcher": {
757         "id": "byName",
758         "options": "Value"
759     },
760     "properties": [
761         {
762             "id": "displayName",
763             "value": "서버 상태"
764         },
765         {
766             "id": "custom.width",
767             "value": 100
768         },
769         {
770             "id": "custom.displayMode",
771             "value": "color-background"
772         },
773         {
774             "id": "custom.cellOptions",
775             "value": {
776                 "type": "color-text"
777             }
778         }
779     ]
780 }
781 ]
782 },
783 "gridPos": {
784     "h": 10,
785     "w": 8,
786     "x": 0,
787     "y": 13
788 },
789 "id": 11,
790 "options": {
791     "footer": {
792         "enablePagination": true,
793         "fields": "",
794         "show": false,
795         "countRows": false
796     },
797     "showHeader": true,
798     "sortBy": [
799         {
800             "desc": false,
801             "displayName": "서버 상태"
802         }

```

```

803     ]
804 },
805 "pluginVersion": "11.4.0",
806 "targets": [
807     {
808         "datasource": "${k1c_PROMETHEUS}",
809         "expr": "ktcloud_lb_server_state{lb_name=\"${selected_lb}\"}",
810         "format": "table",
811         "instant": true,
812         "interval": "",
813         "legendFormat": "",
814         "refId": "A"
815     }
816 ],
817 "title": "📉 Load Balancer 선택: ${selected_lb} | 연결된 서버 상태",
818 "transformations": [
819     {
820         "id": "organize",
821         "options": {
822             "excludeByName": {
823                 "__name__": true,
824                 "Time": true,
825                 "job": true,
826                 "instance": true,
827                 "lb_id": true,
828                 "zone": true,
829                 "environment": true,
830                 "service": true,
831                 "vm_id": true,
832                 "vm_name": true
833             },
834             "indexByName": {
835                 "lb_name": 0,
836                 "server_ip": 1,
837                 "server_port": 2,
838                 "Value": 3
839             },
840             "renameByName": {}
841         }
842     }
843 ],
844 "type": "table"
845 },
846 {
847     "datasource": "${k1c_PROMETHEUS}",
848     "fieldConfig": {
849         "defaults": {
850             "color": {
851                 "mode": "palette-classic"
852             },
853             "custom": {
854                 "axisLabel": "Response Time (ms)",
855                 "axisPlacement": "auto",
856                 "barAlignment": 0,
857                 "drawStyle": "line",
858                 "fillOpacity": 10,

```

```

861     "gradientMode": "none",
862     "hideFrom": {
863         "legend": false,
864         "tooltip": false,
865         "vis": false
866     },
867     "lineInterpolation": "linear",
868     "lineWidth": 2,
869     "pointSize": 5,
870     "scaleDistribution": {
871         "type": "linear"
872     },
873     "showPoints": "auto",
874     "spanNulls": false,
875     "stacking": {
876         "group": "A",
877         "mode": "none"
878     },
879     "thresholdsStyle": {
880         "mode": "off"
881     }
882 },
883 "mappings": [],
884 "thresholds": {
885     "mode": "absolute",
886     "steps": [
887         {
888             "color": "green",
889             "value": null
890         }
891     ]
892 },
893 "unit": "ms",
894 "min": 0
895 },
896 "overrides": []
897 },
898 "gridPos": {
899     "h": 10,
900     "w": 4,
901     "x": 8,
902     "y": 13
903 },
904 "id": 21,
905 "options": {
906     "legend": {
907         "calcs": ["lastNotNull", "max", "mean"],
908         "displayMode": "table",
909         "placement": "bottom",
910         "showLegend": true
911     },
912     "tooltip": {
913         "mode": "multi",
914         "sort": "none",
915         "shred": true
916     }
917 },
918 "cursor": {

```



```

919     "sync": "crosshair"
920   }
921 },
922 "targets": [
923   {
924     "datasource": "${ktc_PROMETHEUS}",
925     "expr": "ktcloud_server_avg_ttfb_ms{lb_name=\"${selected_lb}\"}",
926     "interval": "",
927     "legendFormat": "{{server_ip}}:{{server_port}} (TTFB)",
928     "refId": "A"
929   }
930 ],
931 "title": "TTFB (Time To First Byte)",
932 "type": "timeseries",
933 "description": "서버별 평균 응답 시간을 모니터링합니다."
934 },
935
936
937 {
938   "datasource": "${ktc_PROMETHEUS}",
939   "fieldConfig": {
940     "defaults": {
941       "color": {
942         "mode": "palette-classic"
943       },
944       "custom": {
945         "axisLabel": "Throughput",
946         "axisPlacement": "auto",
947         "barAlignment": 0,
948         "drawStyle": "line",
949         "fillOpacity": 10,
950         "gradientMode": "none",
951         "hideFrom": {
952           "legend": false,
953           "tooltip": false,
954           "vis": false
955         },
956         "lineInterpolation": "linear",
957         "lineWidth": 2,
958         "pointSize": 5,
959         "scaleDistribution": {
960           "type": "linear"
961         },
962         "showPoints": "auto",
963         "spanNulls": false,
964         "stacking": {
965           "group": "A",
966           "mode": "none"
967         },
968         "thresholdsStyle": {
969           "mode": "off"
970         }
971       },
972       "mappings": [],
973       "thresholds": {
974         "mode": "absolute",
975         "steps": [
976           {

```

```

977     "color": "green",
978     "value": null
979 }
980 ]
981 },
982 "unit": "KBs",
983 "min": 0
984 },
985 "overrides": []
986 },
987 "gridPos": {
988   "h": 10,
989   "w": 4,
990   "x": 12,
991   "y": 13
992 },
993 "id": 22,
994 "options": {
995   "legend": {
996     "calcs": ["lastNotNull"],
997     "displayMode": "list",
998     "placement": "bottom",
999     "showLegend": true
1000   },
1001   "tooltip": {
1002     "mode": "multi",
1003     "sort": "desc"
1004   },
1005   "cursor": {
1006     "sync": "crosshair"
1007   }
1008 },
1009 "targets": [
1010   {
1011     "datasource": "${ktc_PROMETHEUS}",
1012     "expr": "ktcloud_server_throughput_rate_kbps{lb_name=\"${selected_lb}\"}",
1013     "interval": "",
1014     "legendFormat": "{{server_ip}}:{{server_port}}",
1015     "refId": "A"
1016   }
1017 ],
1018 "title": "Throughput",
1019 "type": "timeseries",
1020 "description": "서버별 네트워크 처리량"
1021 },
1022
1023
1024
1025 {
1026   "datasource": "${ktc_PROMETHEUS}",
1027   "fieldConfig": {
1028     "defaults": {
1029       "color": {
1030         "mode": "palette-classic"
1031       },
1032       "custom": {
1033         "axisLabel": "Requests/sec",
1034         "axisPlacement": "auto",

```

```

1035     "barAlignment": 0,
1036     "drawStyle": "line",
1037     "fillOpacity": 10,
1038     "gradientMode": "none",
1039     "hideFrom": {
1040         "legend": false,
1041         "tooltip": false,
1042         "vis": false
1043     },
1044     "lineInterpolation": "linear",
1045     "lineWidth": 2,
1046     "pointSize": 5,
1047     "scaleDistribution": {
1048         "type": "linear"
1049     },
1050     "showPoints": "auto",
1051     "spanNulls": false,
1052     "stacking": {
1053         "group": "A",
1054         "mode": "none"
1055     },
1056     "thresholdsStyle": {
1057         "mode": "off"
1058     }
1059 },
1060 "mappings": [],
1061 "thresholds": {
1062     "mode": "absolute",
1063     "steps": [
1064         {
1065             "color": "green",
1066             "value": null
1067         }
1068     ]
1069 },
1070 "unit": "reqps",
1071 "min": 0
1072 },
1073 "overrides": []
1074 },
1075 "gridPos": {
1076     "h": 10,
1077     "w": 4,
1078     "x": 16,
1079     "y": 13
1080 },
1081 "id": 23,
1082 "options": {
1083     "legend": {
1084         "calcs": ["lastNotNull"],
1085         "displayMode": "list",
1086         "placement": "bottom",
1087         "showLegend": true
1088     },
1089     "tooltip": {
1090         "mode": "multi",
1091         "sort": "desc"
1092     },

```

```

1093     "cursor":{
1094         "sync": "crosshair"
1095     }
1096 },
1097 "targets": [
1098     {
1099         "datasource": "${ktc_PROMETHEUS}",
1100         "expr": "ktcloud_server_requests_rate_per_sec{lb_name=\"${selected_lb}\"}",
1101         "interval": "",
1102         "legendFormat": "{{server_ip}}:{{server_port}}",
1103         "refId": "A"
1104     }
1105 ],
1106 "title": "Requests",
1107 "type": "timeseries",
1108 "description": "서버별 초당 요청 처리량"
1109 },
1110
1111 {
1112     "datasource": "${ktc_PROMETHEUS}",
1113     "fieldConfig": {
1114         "defaults": {
1115             "color": {
1116                 "mode": "palette-classic"
1117             },
1118             "custom": {
1119                 "axisLabel": "Active Connections",
1120                 "axisPlacement": "auto",
1121                 "barAlignment": 0,
1122                 "drawStyle": "line",
1123                 "fillOpacity": 20,
1124                 "gradientMode": "opacity",
1125                 "hideFrom": {
1126                     "legend": false,
1127                     "tooltip": false,
1128                     "vis": false
1129                 },
1130                 "lineInterpolation": "linear",
1131                 "lineWidth": 2,
1132                 "pointSize": 5,
1133                 "scaleDistribution": {
1134                     "type": "linear"
1135                 },
1136                 "showPoints": "auto",
1137                 "spanNulls": false,
1138                 "stacking": {
1139                     "group": "A",
1140                     "mode": "none"
1141                 },
1142                 "thresholdsStyle": {
1143                     "mode": "line"
1144                 }
1145             },
1146             "mappings": [],
1147             "thresholds": {
1148                 "mode": "absolute",
1149                 "steps": [

```

```

1151     {
1152         "color": "green",
1153         "value": null
1154     },
1155     {
1156         "color": "yellow",
1157         "value": 50
1158     },
1159     {
1160         "color": "red",
1161         "value": 100
1162     }
1163 ]
1164 },
1165 "unit": "short",
1166 "min": 0
1167 },
1168 "overrides": [
1169     {
1170         "matcher": {
1171             "id": "byName",
1172             "options": "Total"
1173         },
1174         "properties": [
1175             {
1176                 "id": "color",
1177                 "value": {
1178                     "mode": "fixed",
1179                     "fixedColor": "purple"
1180                 }
1181             },
1182             {
1183                 "id": "custom.lineWidth",
1184                 "value": 3
1185             }
1186         ]
1187     }
1188 ],
1189 },
1190 "gridPos": {
1191     "h": 10,
1192     "w": 4,
1193     "x": 20,
1194     "y": 13
1195 },
1196 "id": 24,
1197 "options": {
1198     "legend": {
1199         "calcs": ["lastNotNull", "max", "mean"],
1200         "displayMode": "table",
1201         "placement": "bottom",
1202         "showLegend": true
1203     },
1204     "tooltip": {
1205         "mode": "multi",
1206         "sort": "desc"
1207     }
1208 },

```

```

1209     "targets": [
1210         {
1211             "datasource": "${ktc_PROMETHEUS}",
1212             "expr": "ktcloud_server_current_connections{lb_name=\"${selected_lb}\"} > 0",
1213             "interval": "",
1214             "legendFormat": "{{server_ip}}:{{server_port}}",
1215             "refId": "A"
1216         },
1217         {
1218             "datasource": "${ktc_PROMETHEUS}",
1219             "expr": "sum(ktcloud_server_current_connections{lb_name=\"${selected_lb}\"} > 0)",
1220             "interval": "",
1221             "legendFormat": "Total",
1222             "refId": "B"
1223         }
1224     ],
1225     "title": "Connections",
1226     "type": "timeseries",
1227     "description": "서버별 현재 활성 연결 수를 모니터링합니다."
1228 }
1229
1230
1231 ],
1232 "refresh": "30s",
1233 "schemaVersion": 40,
1234 "style": "dark",
1235 "tags": ["ktcloud", "loadbalancer", "monitoring"],
1236 "templating": {
1237     "list": [
1238         {
1239             "name": "lb_name",
1240             "type": "query",
1241             "datasource": {
1242                 "type": "prometheus",
1243                 "uid": "${ktc_PROMETHEUS}"
1244             },
1245             "query": {
1246                 "query": "label_values(ktcloud_lb_info, lb_name)",
1247                 "refId": "PrometheusVariableQueryEditor-VariableQuery"
1248             },
1249             "definition": "label_values(ktcloud_lb_info, lb_name)",
1250             "refresh": 1,
1251             "regex": "",
1252             "hide": 0,
1253             "includeAll": true,
1254             "multi": true,
1255             "sort": 1,
1256             "label": "Load Balancer",
1257             "allValue": ".*"
1258         },
1259         {
1260             "name": "selected_lb",
1261             "type": "query",
1262             "datasource": {
1263                 "type": "prometheus",
1264                 "uid": "${ktc_PROMETHEUS}"
1265             },
1266             "query": {

```

```

1267     "query": "label_values(ktcloud_lb_info, lb_name)",
1268     "refId": "PrometheusVariableQueryEditor-VariableQuery"
1269 },
1270 "definition": "label_values(ktcloud_lb_info, lb_name)",
1271 "refresh": 1,
1272 "regex": "",
1273 "hide": 0,
1274 "includeAll": false,
1275 "multi": false,
1276 "sort": 1,
1277 "label": "상세 조회 LB (선택)",
1278 "current": {
1279     "selected": false,
1280     "text": "All",
1281     "value": "$__all"
1282 }
1283 },
1284 {
1285     "name": "zone",
1286     "type": "query",
1287     "datasource": {
1288         "type": "prometheus",
1289         "uid": "${ktc_PROMETHEUS}"
1290     },
1291     "query": {
1292         "query": "label_values(ktcloud_lb_info, zone)",
1293         "refId": "PrometheusVariableQueryEditor-VariableQuery"
1294     },
1295     "definition": "label_values(ktcloud_lb_info, zone)",
1296     "refresh": 1,
1297     "regex": "",
1298     "hide": 0,
1299     "includeAll": true,
1300     "multi": true,
1301     "sort": 1,
1302     "label": "Zone",
1303     "allValue": ".*"
1304 }
1305 ]
1306 },
1307 "time": {
1308     "from": "now-1h",
1309     "to": "now"
1310 },
1311 "timepicker": {},
1312 "timezone": "Asia/Seoul",
1313 "title": "kt Cloud_Dashboard - Load Balancer",
1314 "uid": "ktcloud-lb-enhanced-dashboard",
1315 "version": 1,
1316 "weekStart": ""
1317 }

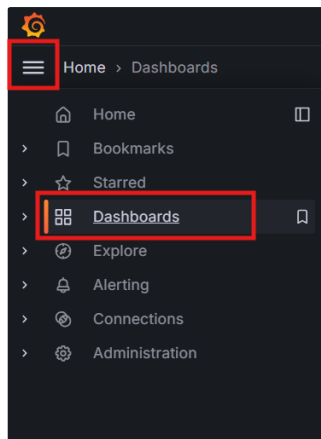
```

2. 대시보드 Import

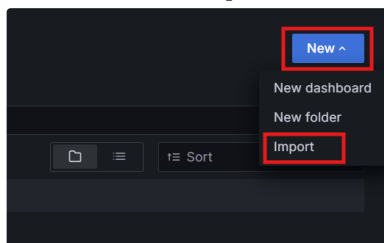
1. Grafana URL 접속

[Grafana IP]:3000

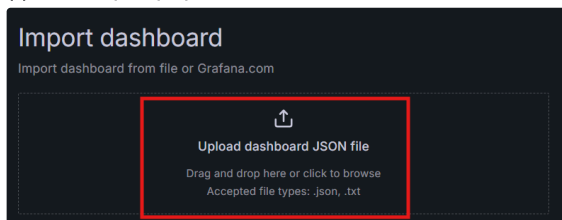
2. 좌측 상단에 [≡] → [Dashboards] 클릭



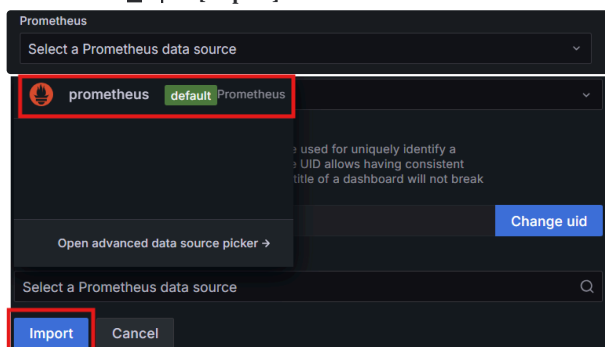
3. 우측 상단 [New] → [Import] 클릭



4. 업로드할 파일 추가



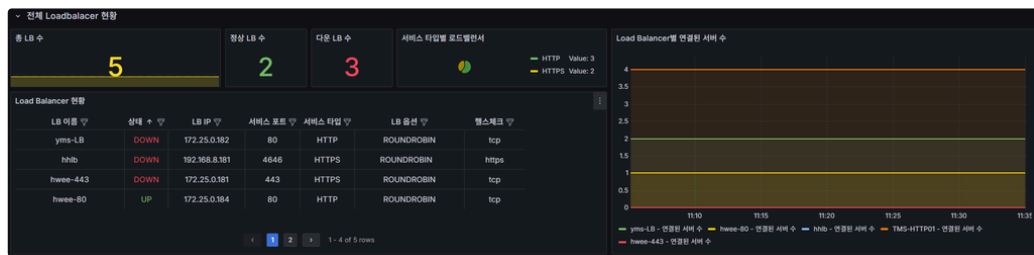
5. Prometheus 선택 → [Import]



#4. 완성 대시보드

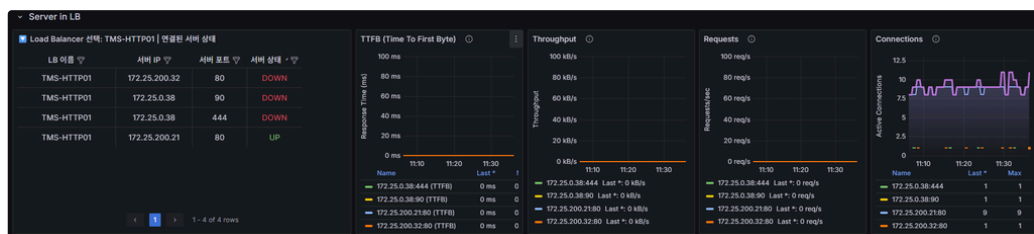
1. LB 전체 현황

- 수집된 계정 내 전체 LB의 상태 조회
 - LB의 IP, Port, Type, option, Healthcheck 조회



2. Server in LB

- LB에 연결된 서버의 상세 정보 확인 가능
 - 서버에 연결된 VM 현황
 - 각 서버별 TTFB, Throughput, Requests, Connections

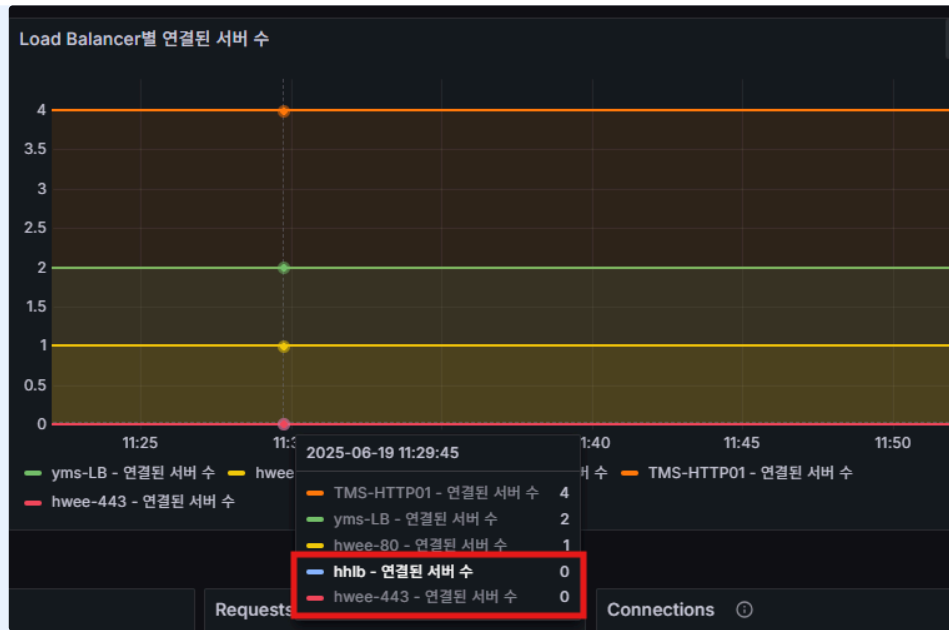


주의사항

그래프에 마우스 커서를 올렸을 경우, 수치에 해당하는 값들을 표기해 주는데 동일한 수치를 가지고 있는 값이 여러개일 경우 하나만 하이라이트 표시

ex) LB별 연결된 서버 수

- 실제 연결된 서버 수가 0인 LB가 2개가 존재하지만 하이라이트 표시는 하나만 표현
- Grafana특성 상 여러개의 데이터를 하이라이트 표기 불가



위와 같이 [hhlb], [hwee-443] 두 개의 LB가 0 값을 가지고 있지만 하나만 하이라이트 표현