

성능 부하 테스트 - Advanced

차용빈 (dr.coton@gmail.com)
손영수 (indigoguru@gmail.com)



강의 소개

이 강의에서 여러분은 실제 서버에 테스트 스크립트를 작성하여 성능 테스트와 부하 테스트를 진행하는 과정에 대해서 학습할 수 있습니다. 이 강의를 충분히 학습한다면, 다양한 서버 및 다양한 조건에서의 성능 테스트를 직접 운영할 수 있으며 이를 바탕으로 성능 개선 및 장애 대응을 할 수 있습니다.

이 강의는 테스트를 하는 방법에 대한 강의이며, 실습으로 강의를 구성되어 있습니다. **basic**과 **advanced** 과정으로 나뉘어 있으며,

basic 과정은 성능 부하 테스트에 간략한 개요와 간단한 실습에 대해서 다루고 있으며,

advanced 과정은 실제 서버를 생성하여 테스트 하는 방법을 다루고 있으며, 상세한 성능 부하 테스트의 지표들을 시각화 하고 관리 하는 방법에 대해서 소개하고 있습니다.

대규모 부하 테스트에 대한 내용은 별도의 강의로 구성하고 있으며, 간략한 개요는 **advanced** 과정 마지막 부분에 언급됩니다.

목차

Basic 과정

1. 성능, 부하 테스트 개요
2. Apache JMeter 개요
3. Apache JMeter 스크립트 작성 방법
4. Apache JMeter Plugin
5. 결과 데이터 분석 방법

Advanced 과정

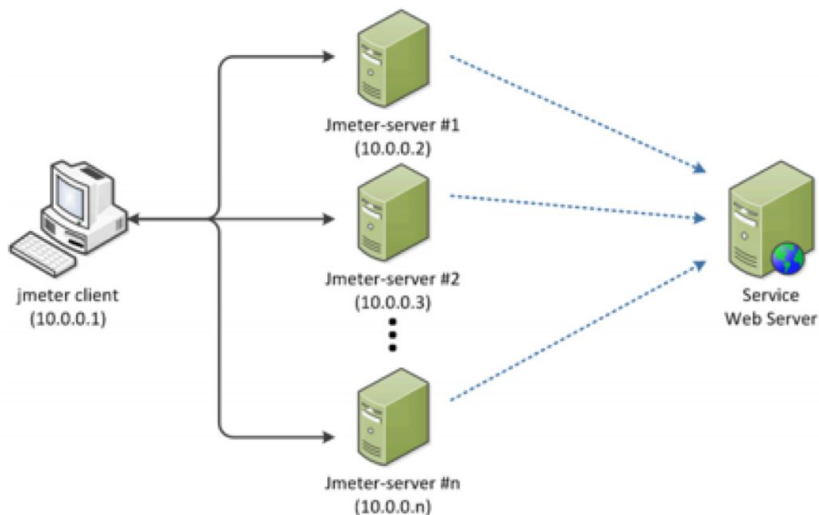
1. 대규모 부하 분산 테스트 개요
2. 웹 서버 설치
3. 실전 부하 테스트 스크립트 작성
4. JMeter 데이터 시각화
5. 실전 웹 서버 성능 진단

대규모 부하 분산 테스트 개요

대용량 부하 분산 테스트는 한 대의 부하발생기로는 불가능하고, 여러 대의 부하발생기를 병렬로

연결해서 동시에 테스트를 해야만 한다. JMeter는 이런 환경을 구축하기 위해 복수의

jmeter-server를 원격에서 실행할 수 있는 Remote Testing 기능을 제공한다.



JMeter Controller

실제로 부하를 발생시키는 jmeter-server에 명령을 전달하는 Master 역할을 수행하는 노드다.

JMeter Server

실제로 부하를 발생하는 노드로, jmeter controller에서 받은 명령을 수행하고 그 결과값을 다시 jmeter controller로 전달하는 Slave 역할을 한다.

JMeter Controller 설정

JMETER_HOME/bin/jmeter.properties 파일을 편집한다. jmeter-server 접속을 위한 IP 설정과 결과 데이터를 받아오는 방식을 설정한다.

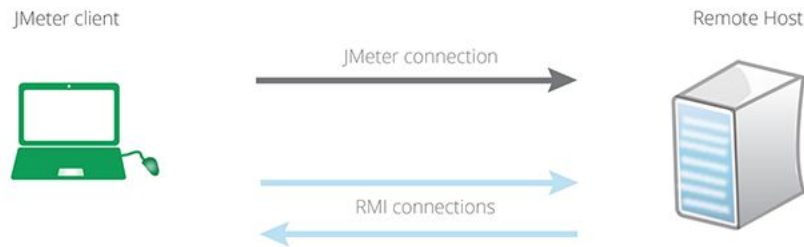
jmeter controller에서 접속할 remote jmeter-server의 정보를 입력하는 부분이다. 다음과 같이 콤마(,)로 구분된 IP 리스트를 적어준다. 포트 정보를 별도로 입력해주지 않으면 1099로 설정된다.

```
remote_hosts=10.0.0.2,10.0.0.3,...,10.0.0.n
```

통신 포트 설정 변경

jmeter controller와 jmeter-server 사이에 방화벽이 존재한다면 특정 포트를 별도로 설정해야 한다. 그래야만 방화벽에 해당 포트를 등록하고 통신할 수 있다.

JMeter는 jmeter controller와 jmeter - server 사이에 양방향 통신이 가능해야 한다. Remote Start 버튼을 누르면 처음에는 jmeter controller가 jmeter-server의 서버 포트를 이용해서 접속하지만, 실제로 테스트가 실행되면 서로 RMI(Remote Method Invocation) 포트를 이용해서 정보를 주고받는다.



통신 포트 설정 변경

jmeter-server 설정

```
server_port=2099  
server.rmi.localport=4000
```

jmeter controller 설정

```
remote_hosts=10.0.0.2:2099  
client.rmi.localport=4001
```

SSH Tunnel 작업

JMeter client



FIREWALL 1:N NAT



JMeter connection

Internet

RMI connections

FIREWALL 1:1 NAT



Google Compute Engine Instances

SSH Tunnel 작업

Server-side Port Forwarding



Client-side Port Forwarding



SSH Tunnel 작업



```
$ export JVM_ARGS="-Djava.rmi.server.hostname=localhost"  
$ ssh -R 4001:localhost:4001 10.0.0.2
```

분산을 위한 서버 수 선정 방법

JMeter는 Java 기반이기 때문에 CPU보다는 Memory 크기에 상관관계가 크다. 보통 대개 Jmeter Server는 CPU 2Core 메모리 4GB로 환경을 맞춘다. 이 경우에 대부분의 테스트 경우에는 10,000 쓰레드까지는 문제 없이 테스트가 가능하다

원하는 부하 쓰레드 수에서 최대 10,000으로 나누어 Jmeter Server 를 준비하는 것이 좋다.

부하 테스트 시 주의 사항

사전 테스트는 본격적인 테스트를 하기 전에 테스트가 정상적으로 수행될 수 있을지 검사하는 과정이므로 많은 수의 가상 사용자로 오랫동안 테스트하는 것이 아니라 적은 수의 가상 사용자로 스크립트를 실행해 보는 것이다. 여러 대의 부하발생기로 테스트할 예정이라면 부하발생기도 정상 동작하는지 확인해야 한다.

결과는 View Results Tree를 활성화해서 확인한다. View Results Tree는 사전테스트에서 사용하려고 Test Plan에 추가한 것이므로 사전 테스트가 끝난 후에는 반드시 비활성화시켜야 한다.

이를 통해 확인 해야하는 사항은 다음 장에서 소개된다.

부하 테스트 시 주의 사항

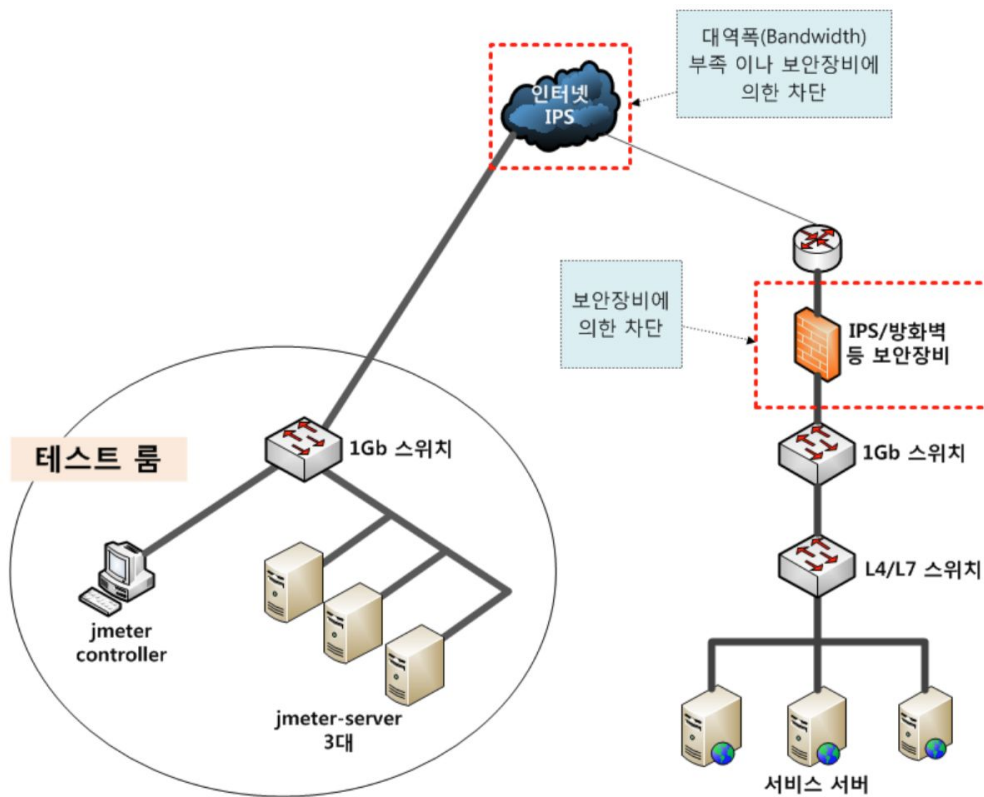
Case 1. Test Plan 스크립트는 정상 동작하는가 ?

때로는 Test Plan 스크립트를 작성한 시점과 실제 테스트하는 시점에서 애플리케이션의 로직이 변경되어 오류가 발생하곤 한다.

사전 테스트에서 문제가 발견되면 개발팀과 상의해서 변경사항을 Test Plan 스크립트에 추가 또는 변경해야 한다. Test Plan 스크립트를 작성한 PC와 부하발생기의 JMeter 버전이나 확장 모듈 내용이 달라서 작성한 Test Plan 스크립트가 구동되지 않는 경우도 있으므로 꼭 버전을 맞춰준다.

부하 테스트 시 주의 사항

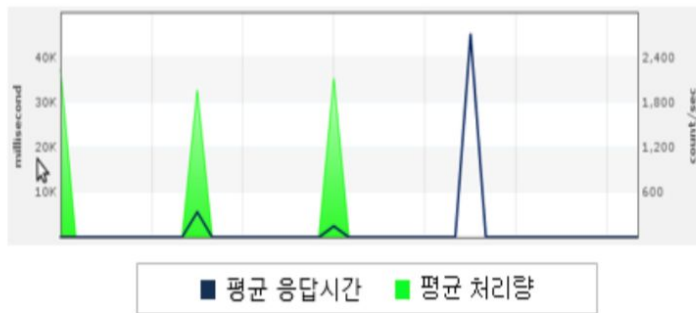
Case 2. 방화벽 및 L4 스위치 차단 확인



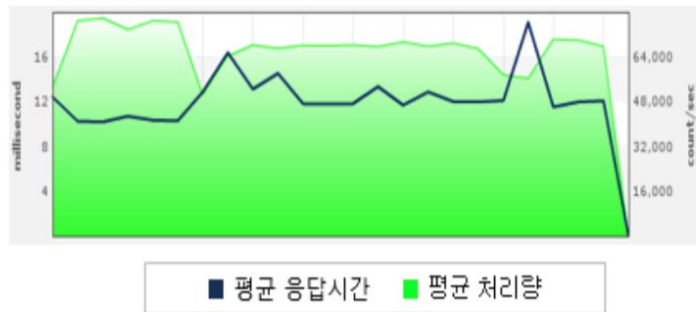
부하 테스트 시 주의 사항

Case 2. 방화벽 및 L4 스위치 차단 확인

■ Case.1 Firewall On



■ Case.2 Firewall Off



부하 테스트 시 주의 사항

Case 3. JMeter Controller의 CPU 사용률

분산 테스트를 할 때 jmeter controller는 Test Plan 스크립트의 로직을 수행하는 것이 아니라 단순히 jmeter-server에서 전달되는 결과를 취합해서 화면에 보여주는 역할을 한다. 그러므로 CPU 사용률이 높다면 결과 취합 과정에서 어떤 문제가 발생한 것이다.

이런 문제는 결과 취합 모드가 Standard일 때 자주 발생한다. Standard 모드일 때에는 테스트 결과를 실시간으로 전달받아서 바로 화면에 결과를 업데이트하므로 결과를 손쉽게 확인할 수 있지만, 여러 대의 jmeter-server의 결과를 한 대의 jmeter controller에서 취합하다 보니 용량이 부족할 때가 많다.

가장 쉬운 해결 방법은 결과 취합 모드를 Statistical 모드로 변경하는 것이다. Statistical 모드일 때에는 jmeter-server에서 결과를 취합한 후 평균과 합만 jmeter controller로 보내므로 jmeter controller의 부하가 눈에 띄게 줄어든다.

부하 테스트 시 주의 사항

Case 4. JMeter Server의 CPU 사용률

Jmeter-server는 실제로 Test Plan 스크립트의 로직이 실행되는 부분이므로 CPU를 많이 사용하는 로직이 스크립트에 포함되어 있다. 가장 쉬운 해결 방법은 jmeter-server를 증설하는 것이다.

서버 증설로 한 대에서 구동되는 가상 사용자의 수를 줄여주면 CPU 부하도 함께 줄어든다. 증설이 힘든 상황이라면 BeanShell이나 If Controller 등 CPU를 많이 사용하는 요소를 제거하고 Test Plan 스크립트를 재구성해야 한다.

부하 테스트 시 주의 사항

Case 5. 스위치의 IP Hash 방식


한 대의 서비스 서버로 테스트 요청이 몰리는 현상은 L4 스위치를 사용하는 환경에서 흔히 발생하는 문제다.

대부분 L4의 Hash 방식은 요청 클라이언트의 IP를 기준으로 서비스 서버를 할당하므로 한 대의 부하발생기에 아무리 많은 가상 사용자가 생성되어 있어도 모두 하나의 서버로 몰릴 수밖에 없게 된다.

이를 해결하는 방식은 최대한 여러개의 IP를 준비하는 것이다. 외부 환경에서 테스트하는 경우에는 각기 다른 공인IP를 가지고 있어야 하며, 테스트를 망 안에서 하는 경우에는 많은 수의 부하 발생기 (jmeter-server)를 준비하거나 HTTP Request 에서 IP를 여러개 추가 해주는 방식이 있다.

HTTP Request 에서 Source IP를 가변으로 변경하면 된다. 이때 여러 개의 IP를 준비할 수 있는 상황이어야 한다. 이 IP를 CSV 파일로 준비하여 Source IP 에 추가하면 된다.

AWS EC2와 SSM을 활용한 분산 테스트

 서비스 ▼

AWS Systems Manager

×

빠른 설정

▼ 운영 관리

탐색기 신규

OpsCenter

CloudWatch 대시보드

PHD

▼ 애플리케이션 관리

리소스 그룹

AppConfig 신규

Parameter Store

▼ 작업 및 변경

자동화

일정 변경 신규

유지 관리 기간

▼ 인스턴스 및 노드

규정 준수

인벤토리

관리형 인스턴스

하이브리드 활성화

Session Manager

Run Command

관리

AWS Systems Manager 명령 실행

보안이 유지되고 안전한 규모별 원격 관리

기본 제공 안전 제어 및 감사 기능으로 Amazon EC2 인스턴스 및 온프레미스 서버 또는 VM(가상 머신)을 안전하게 관리합니다

인스턴스 관리

명령 실행

사용 사례 및 블로그 포스트

SSH 액세스 없이 대규모로 인스턴스 관리

보안이 유지되고, 확장 가능하며, 효율적인 인스턴스 관리

설명서

설명서

API 참조

FAQ

작동 방식

1 사전 정의된 명령 문서 사용 또는 직접 생성

2 수동으로 인스턴스 선택 또는 태그를 사용하여 선택

3 안전 제어 및 일정 선택

4 실행 후 결과 보기

기능 및 장점

모든 위치에서 인스턴스 관리

데이터 센터에서 규모별로 EC2 인스턴스 또는 온프레미스 서버 및 VM을 관리합니다.

확장 가능한 권한 및 액세스 제어를 사용하여 안전하게 관리

IAM 및 태그 기반 권한을 사용하여 특정 시스템에서 권한이 있는 작업을 수행할 수 있는 사용자를 제어합니다.

관리 오버헤드 감소

배치된 호스트 설치나 SSH 키 및 인증서 관리 없이 인스턴스 및 온프레미스 서버 또는 VM을 관리합니다.

플릿에서 명령 실행 제어

오류 임계값에 따라 명령을 배치로 실행하는 속도 제어를 사용하여 구성 변경을 안전하게 수행합니다.

Run Script 예시 - 초기 설정

```
# setup.sh

sudo apt-get update
sudo apt-get install -y openjdk-8-jre
sudo apt install -y openjdk-18-jdk-headless
sudo apt-get install -y htop
sudo apt install -y unzip

cd /

sudo mkdir jmeter

cd jmeter

sudo wget https://storage.s3.ap-northeast-2.amazonaws.com/apache-jmeter-5.3.zip
sudo unzip apache-jmeter-5.3.zip

sudo wget https://storage.s3.ap-northeast-2.amazonaws.com/test.jmx
```

Run Script 예시 - 테스트 진행

```
# start.sh
```

```
cd /
```

```
sudo ./jmeter/apache-jmeter-5.3/bin/shutdown.sh
```

```
sudo rm -rf /jmeter/test.jmx
```

```
cd /jmeter
```

```
sudo wget https://storage.s3.ap-northeast-2.amazonaws.com/test.jmx
```

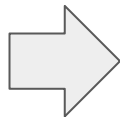
```
cd /
```

```
sudo nohup ./jmeter/apache-jmeter-5.3/bin/jmeter.sh -n -t /jmeter/test.jmx&
```

AWS EC2와 SSM을 활용한 분산 테스트

Q. 동시접속자 30만명 테스트 환경을 1시간 테스트 하면 얼마나 비용이 나올까?

- CPU 2코어, 메모리 4GB 가상 머신 사용
- 가상 머신 1대 당 동시접속자 1만명 배정
- 가상 머신 총 30대 사용



- AWS 기준, t3a.medium 사이즈 사용.
- 네트워크 비용은 계산하지 않음.

t3a.medium (서울) 기준, 시간 당 0.0468\$

=> $0.0468\$ \times 30\text{대} = 1.404\$$ (약 1,500원)

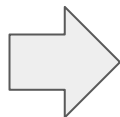
=> 스토리지 비용 및 네트워크 비용 합 대략 (600원)

=> **총 2,100원 예상**

AWS EC2와 SSM을 활용한 분산 테스트

Q. 동시접속자 3만명 테스트 환경을 1주일 장시간 테스트 하면 얼마나 비용이 나올까?

- CPU 2코어, 메모리 4GB 가상 머신 사용
- 가상 머신 1대 당 동시접속자 1만명 배정
- 가상 머신 총 3대 사용



- AWS 기준, t3a.medium 사이즈 사용.
- 네트워크 비용은 계산하지 않음.

t3a.medium (서울) 기준, 시간 당 0.0468\$

=> $0.0468\$ \times 3\text{대} \times 24\text{시간} \times 7\text{일} = 23.5872\$$ (약 28,000원)

=> 스토리지 비용 및 네트워크 비용 합 대략 (9,600원)

=> **총 37,600원 예상**

웹 서버 설치

Node.js
설치

MySQL
설치

MySQL
설정

DB 쿼리
생성

서버
클러스터
실행

웹 서버 설치

Code Clone

```
git clone https://github.com/dr-coton/Board-Restful-API-Server.git  
cd Board-Restful-API-Server
```

#

```
sudo chmod 755 install_env.sh  
sudo ./install_env.sh
```

password : root

```
mysql -u root -p
```

```
create database board;
```

Start

```
npm install  
npm run dev
```

API 구성

// 회원가입

```
routes.post('/signup', async ctx => { ...
```

// 로그인

```
routes.post('/signin', async ctx => { ...
```

// 글 작성하기

```
routes.post('/article/write', authorized, async ctx => { ...
```

// 글 전체 가져오기

```
routes.get('/article/all', authorized, async ctx => { ...
```

// 특정 글 가져오기

```
routes.get('/article/:aid', authorized, async ctx => { ...
```

// 글 검색하기

```
routes.get('/article/search/:keyword', authorized, async ctx => { ...
```

게시판 api (1)

회원가입

POST /signup? HTTP/1.1

Host: 127.0.0.1:5000

Content-Type: application/json

Content-Length: 67

```
{  
  "id" : "test",  
  "name" : "test",  
  "password" : "test"  
}
```

로그인

POST /signin HTTP/1.1

Host: 127.0.0.1:5000

Content-Type: application/json

Content-Length: 46

```
{  
  "id" : "test",  
  "password" : "test"  
}
```

게시판 api (2)

글쓰기

POST /article/write HTTP/1.1

Host: 127.0.0.1:5000

Authorization: bearer XXXXXXXXXXXXXXXXXXXX

Content-Type: application/json

Content-Length: 96

```
{
  "title" : "akldsijkljalkdjklada",
  "contents" : "asdfasfjakjl;dkasl;fjal;fjk;lajfkl; "
}
```

전체 게시글 가져오기

GET /article/all HTTP/1.1

Host: 127.0.0.1:5000

Authorization: bearer

XXXXXXXXXXXXXXXXXXXX

게시판 api (3)

특정 글 조회 하기

GET /article/1 HTTP/1.1

Host: 127.0.0.1:5000

Authorization: bearer XXXXXXXXXXXXXXXXXXXX

게시글 검색하기

GET /article/search/akldsjskljalkdjklada

HTTP/1.1

Host: 127.0.0.1:5000

Authorization: bearer

XXXXXXXXXXXXXXXXXXXX

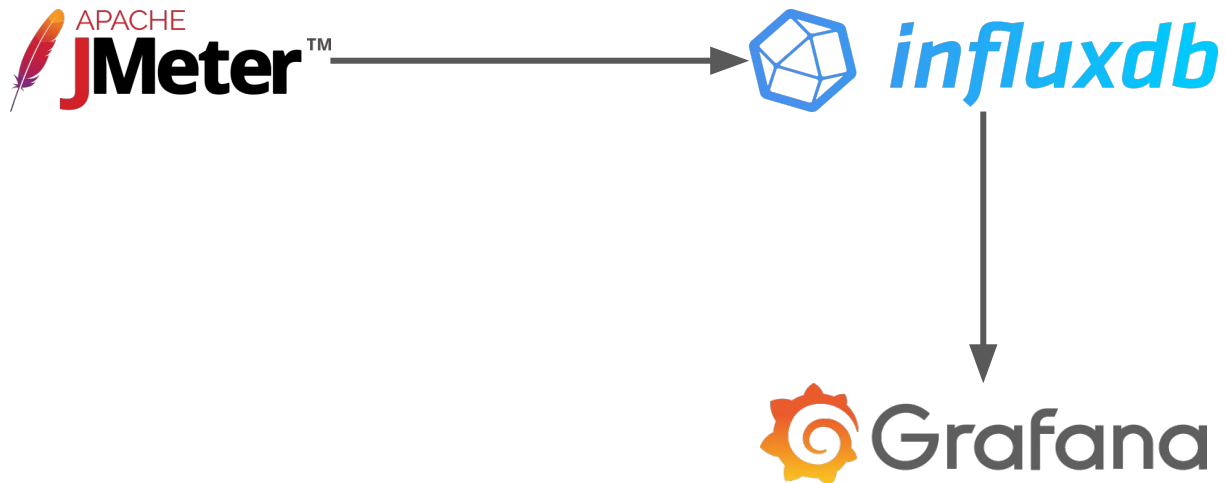
실전 부하 테스트 스크립트 작성

본 내용은 라이브로 진행됩니다.

실전 부하 테스트 스크립트 작성

본 내용은 라이브로 진행됩니다.

JMeter 데이터 시각화



JMeter 데이터 시각화



설치 - Windows

InfluxDB Install

1. https://dl.influxdata.com/influxdb/releases/influxdb-1.8.3_windows_amd64.zip

다운로드 진행

2. 압축 해제
3. influxd.exe 실행
4. influx.exe 실행

Grafana Install

1. <https://dl.grafana.com/oss/release/grafana-7.3.3.windows-amd64.zip>

다운로드 진행

2. 압축 해제
3. grafana-server 실행
4. localhost:3000 접속 후 admin/admin
으로 접속

설치 - MacOS

InfluxDB Install (binaries)

1. https://dl.influxdata.com/influxdb/releases/influxdb-1.8.3_darwin_amd64.tar.gz
2. `tar zxvf influxdb-1.8.3_darwin_amd64.tar.gz`
3. `./usr/bin/influxd` (파인더에서 Command를 누른 상태에서 열기 추천)
4. `./usr/bin/influx` (파인더에서 Command를 누른 상태에서 열기 추천)

Grafana Install (binaries)

1. `https://dl.grafana.com/oss/release/grafana-7.3.3.darwin-amd64.tar.gz`
2. `tar -zxvf grafana-7.3.3.darwin-amd64.tar.gz`
3. `./bin/grafana-server web`

InfluxDB 생성

간단하게 **DB** 사용해보기

1. DB 생성 : `CREATE DATABASE testdb`
2. DB 사용 : `USE testdb`
3. DB 데이터 추가 : `INSERT monitoring`
`cpu=i9,core=16,memory=32000,disk=50000`
4. DB 데이터 확인 : `SELECT * FROM monitoring`

실습을 위한 **DB** 생성

```
CREATE DATABASE jmeter
```

JMeter Backend Listener

Backend Listener

Name: Backend Listener

Comments:


Backend Listener implementation `org.apache.jmeter.visualizers.backend.influxdb.InfluxdbBackendListenerClient`

Async Queue size 5000

Parameters

Name:	Value
influxdbMetricsSender	<code>org.apache.jmeter.visualizers.backend.influxdb.HttpMetricsSender</code>
influxdbUrl	<code>http://localhost:8086/write?db=jmeter</code>
application	Board API
measurement	jmeter
summaryOnly	false
samplersRegex	.*
percentiles	99;95;90
testTitle	Board API Load Test
eventTags	

Grafana 설정

 **Data Sources / InfluxDB**
Type: InfluxDB

Settings

Name ⓘ

InfluxDB

Default ☒

Query Language

InfluxQL

▼

HTTP

URL ⓘ

http://localhost:8086

Access

Server (default) ▼

Help >

Whitelisted Cookies ⓘ

Add Name

Add

Auth

Basic auth ☐

With Credentials ⓘ ☐

TLS Client Auth ☐

With CA Cert ⓘ ☐

Skip TLS Verify ☐

Forward OAuth Identity ⓘ ☐

Custom HTTP Headers

+ Add header

InfluxDB Details

Database

jmeter

User

Password

Password

HTTP Method ⓘ

Choose ▼

Grafana Dashboard 설정

Grafana의 Community Dashboard를 Import하기

- 별도 플러그인 설치 없이 사용하는 대시보드
<https://grafana.com/grafana/dashboards/5496>
- 별도 플러그인을 설치하여 사용하는 대시보드
<https://github.com/mderevyankoaa/jmeter-influxdb-listener-plugin>

md Backend Listener

Backend Listener

Name: Backend Listener

Comments:

Backend Listener implementation `org.md.jmeter.influxdb.visualizer.InfluxDatabaseBackendListenerClient`

Async Queue size 5000

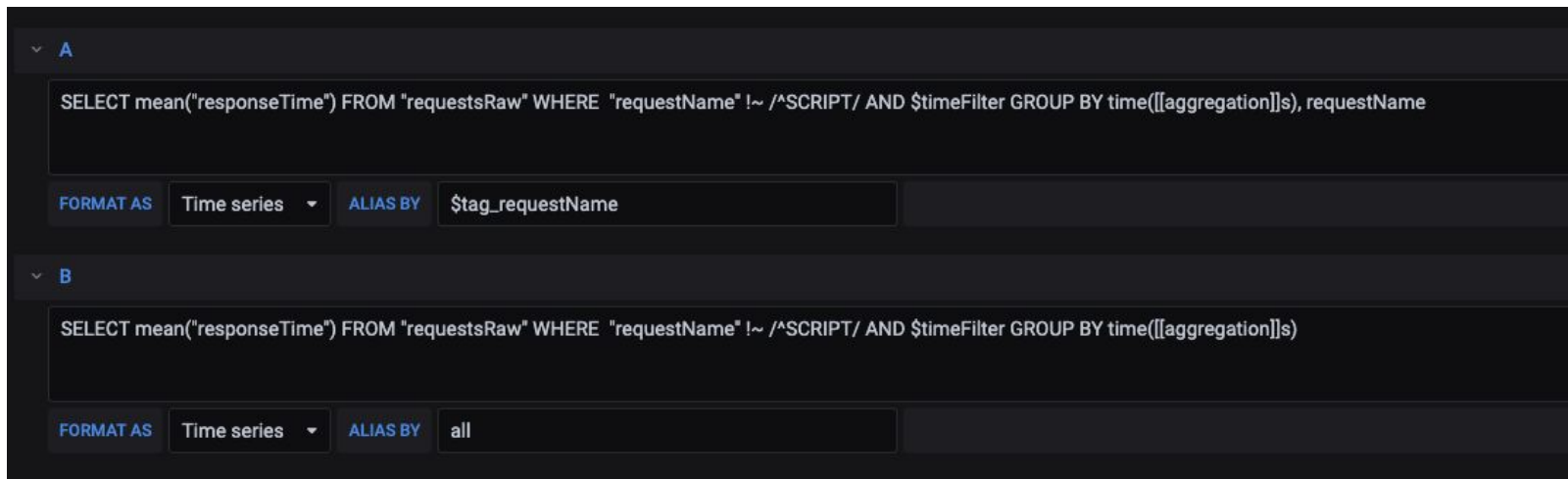
Parameters

Name:	Value
testName	Test
nodeName	Test-Node
runId	R001
influxDBHost	localhost
influxDBPort	8086
influxDBUser	
influxDBPassword	
influxDBDatabase	jmeter
retentionPolicy	autogen
samplersList	.*
useRegexForSamplerList	true
recordSubSamples	true
saveResponseBodyOfFailures	true

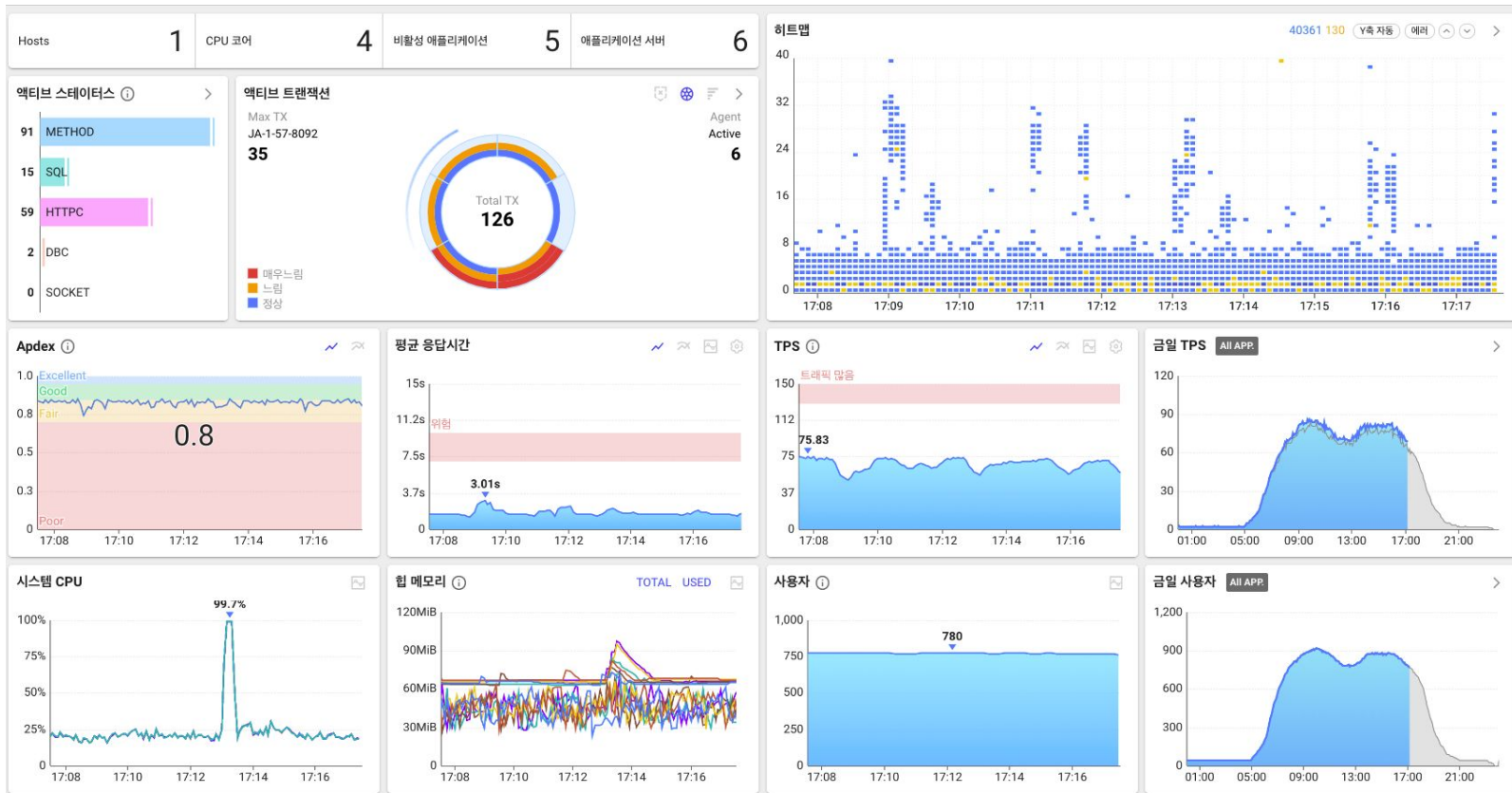
md 플러그인 Grafana 설정

md Grafana Dashboard 인 경우, 대시보드 일부 수정 (전체 평균 응답시간 지표 추가)

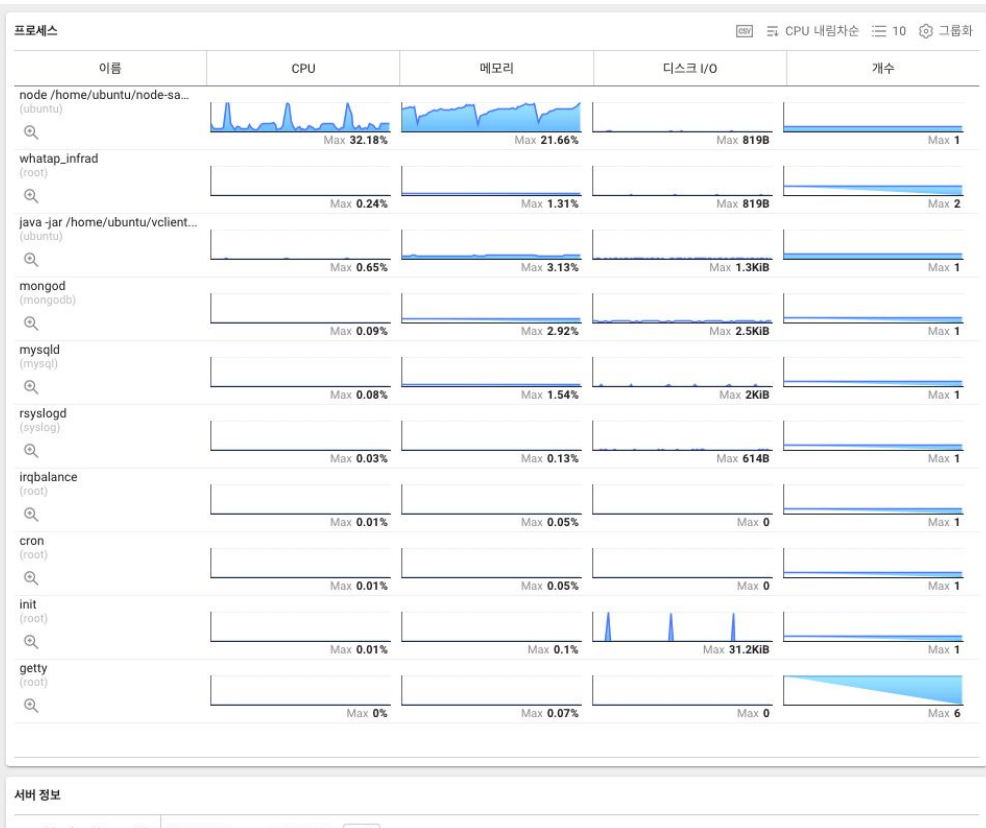
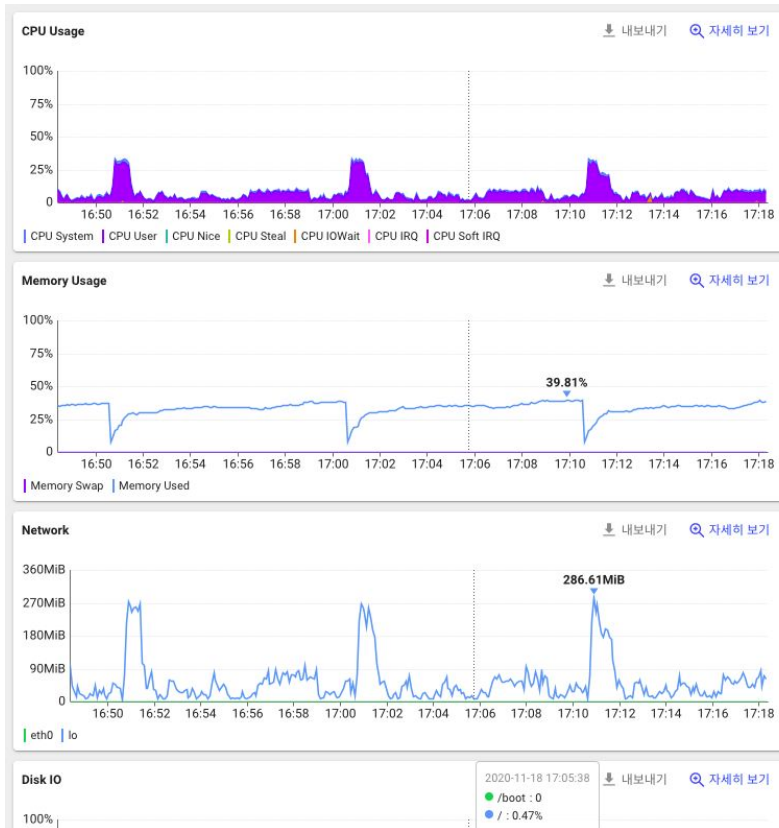
=> Average Response Times 판넬에서 Edit 후 아래와 같이 쿼리 추가



웹 서버 모니터링



웹 서버 모니터링



실전 웹 서버 성능 진단

본 내용은 라이브로 진행됩니다.