

Lab. Using CloudFront with AWS Route 53

목적

이번 실습에서는 생성한 웹 서버가 AWS CDN 서비스인 CloudFront에 등록되어 캐싱을 통해 전 세계 어디서든지 가장 가까운 Edge Location에서 유저에게 웹 서비스를 제공하도록 설정한다. 또한 Cache에 대해 알아보고, AWS Route 53을 통해 DNS 서비스를 이용하는 방법도 살펴본다.

사전 준비물

AWS Free-Tier 계정

WebServer EC2 인스턴스에 웹서버 생성하기

1. WebServer 인스턴스를 위해 VPC를 생성한다.

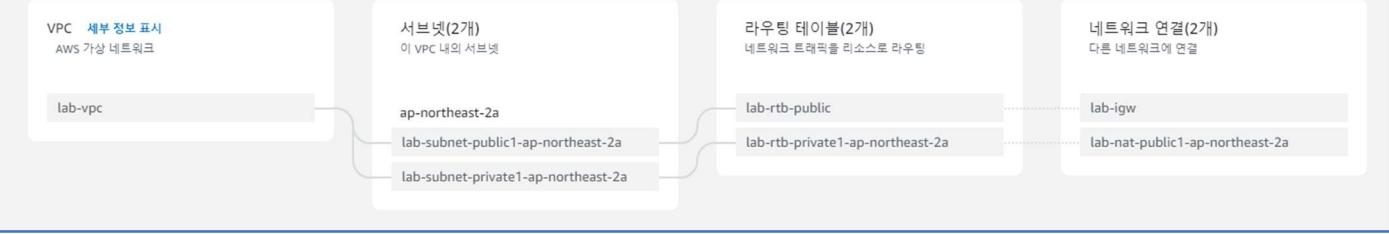
- A. [생성할 리소스] : VPC, 서브넷 등
- B. [이름 태그 자동 생성] : 자동 생성 체크, lab
- C. [IPv4 CIDR 블록] : 10.0.0.0/16
- D. [가용 영역] : 1 / ap-northeast-2a
- E. [퍼블릭 서스넷 수/프라이빗 서브넷 수] : 각 1개(10.0.10.0/24, 10.0.20.0/24)
- F. [NAT 게이트웨이] : AZ당 1개
- G. [DNS 옵션] : DNS 호스트 이름 활성화 체크/DNS 확인 활성화 체크

<p>VPC 설정</p> <p>생성할 리소스 정보 VPC 리소스 또는 VPC 및 기타 네트워킹 리소스만 생성합니다.</p> <p><input type="radio"/> VPC만 <input checked="" type="radio"/> VPC 등</p> <p>이름 태그 자동 생성 정보 이름 태그의 값을 입력합니다. 이 값은 VPC의 모든 리소스에 대한 이름 태그를 자동으로 생성하는 데 사용됩니다.</p> <p><input checked="" type="checkbox"/> 자동 생성 lab</p> <p>IPv4 CIDR 블록 정보 CIDR 표기법을 사용하여 VPC의 시작 IP와 크기를 결정합니다.</p> <p>10.0.0.0/16 65,536 IPs</p> <p>CIDR 블록 크기는 /16에서 /28 사이여야 합니다.</p> <p>IPv6 CIDR 블록 정보 <input checked="" type="radio"/> IPv6 CIDR 블록 없음 <input type="radio"/> Amazon 제공 IPv6 CIDR 블록</p> <p>테넌시 정보 기본값</p>	<p>가용 영역(AZ) 수 정보 서브넷을 프로비저닝할 AZ 수를 선택합니다. 고가용성을 위해서는 최소 2개 이상의 AZ를 사용하는 것이 좋습니다.</p> <table border="1"><tr><td>1</td><td>2</td><td>3</td></tr></table> <p>▼ AZ 사용자 지정</p> <p>첫 번째 가용 영역 ap-northeast-2a</p> <p>퍼블릭 서브넷 수 정보 VPC에 추가할 퍼블릭 서브넷 수입니다. 인터넷을 통해 공개적으로 액세스할 수 있어야 하는 웹 애플리케이션에는 퍼블릭 서브넷을 사용합니다.</p> <table border="1"><tr><td>0</td><td>1</td></tr></table> <p>프라이빗 서브넷 수 정보 VPC에 추가할 프라이빗 서브넷 수입니다. 프라이빗 서브넷을 사용하여 퍼블릭 액세스가 필요 없는 백엔드 리소스를 보호합니다.</p> <table border="1"><tr><td>0</td><td>1</td><td>2</td></tr></table> <p>▼ 서브넷 CIDR 블록 사용자 지정</p> <p>ap-northeast-2a 퍼블릭 서브넷 CIDR 블록 10.0.10.0/24 256 IPs</p> <p>ap-northeast-2a 프라이빗 서브넷 CIDR 블록 10.0.20.0/24 256 IPs</p>	1	2	3	0	1	0	1	2
1	2	3							
0	1								
0	1	2							

<p>NAT 게이트웨이(\$) 정보 NAT 게이트웨이를 생성할 가용 영역(AZ) 수를 선택합니다. 각 NAT 게이트웨이마다 요금이 부과됩니다.</p> <table border="1"><tr><td>없음</td><td>1개의 AZ에서</td><td>AZ당 1개</td></tr></table> <p>VPC 엔드포인트 정보 엔드포인트는 VPC에서 S3에 직접 액세스하여 NAT 게이트웨이 요금을 줄이고 보안을 강화할 수 있습니다. 기본적으로 모든 액세스 정책이 사용됩니다. 언제든지 이 정책을 사용자 지정할 수 있습니다.</p> <table border="1"><tr><td>없음</td><td>S3 게이트웨이</td></tr></table>	없음	1개의 AZ에서	AZ당 1개	없음	S3 게이트웨이
없음	1개의 AZ에서	AZ당 1개			
없음	S3 게이트웨이				

<p>DNS 옵션 정보</p> <p><input checked="" type="checkbox"/> DNS 호스트 이름 활성화 <input checked="" type="checkbox"/> DNS 확인 활성화</p>
--

미리 보기



2. 다음과 같이 **WebServer** 인스턴스를 생성한다.

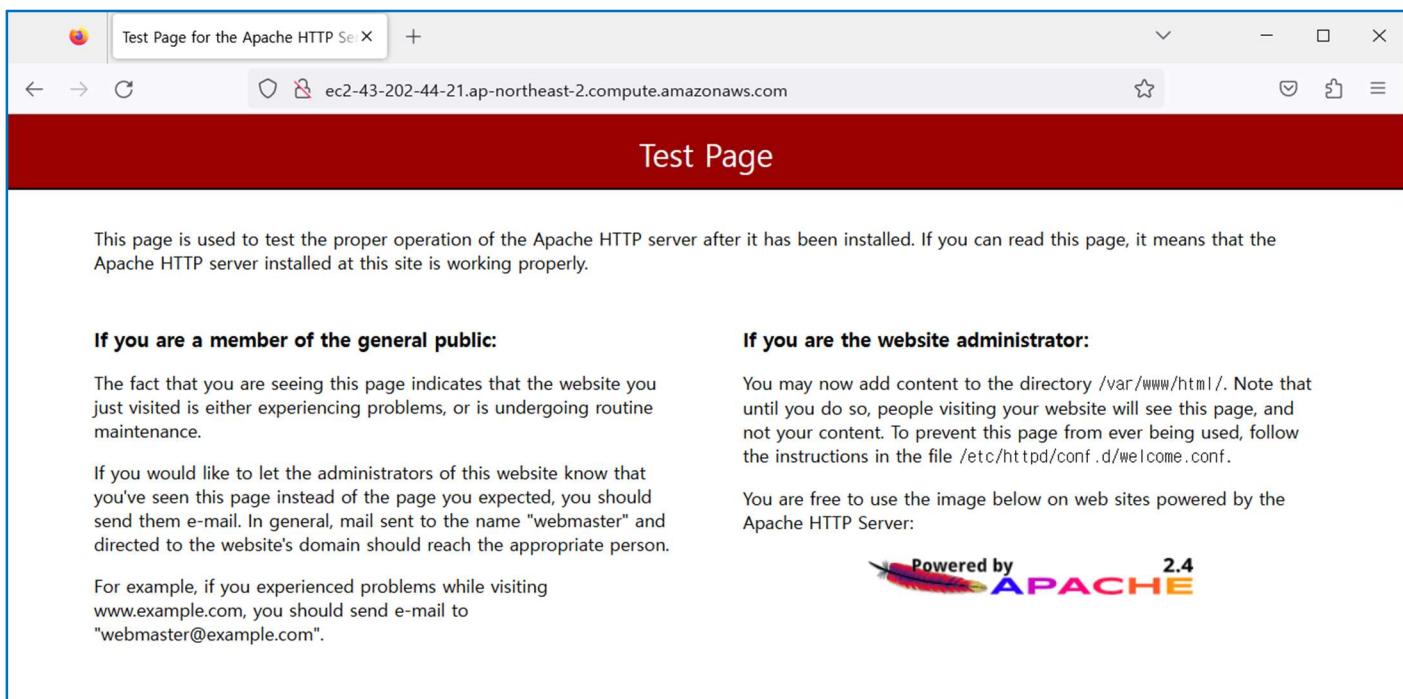
- A. [이름] : lab-webserver-ec2
- B. [AMI] : Amazon Linux 2 AMI(HVM) – Kernel 5.10, SSD Volume Type, 64비트(x86)
- C. [인스턴스 유형 선택] : t2.micro
- D. [키 페어(로그인)] > [새 키 페어 생성] : lab-webserver-ec2-key.pem
- E. [네트워크 설정] > [편집]
 - ① [네트워크] : lab-vpc
 - ② [서브넷] : lab-subnet-public1-ap-northeast-2a
 - ③ [퍼블릭 IP 자동 할당] : 활성화
- F. [방화벽(보안그룹)] > [보안 그룹 생성]
- G. [보안 그룹 이름] : lab-webserver-sg
 - ① [유형] : SSH, [프로토콜] : TCP, [포트 범위] : 22, [소스] : 위치 무관
 - ② [유형] : HTTP, [프로토콜] : TCP, [포트 범위] : 80, [소스] : 위치 무관
- H. [스토리지 구성] : 8 GiB, gp2
- I. [고급 세부 정보] > 사용자 데이터

사용자 데이터 - *optional* 정보
사용자 데이터가 포함된 파일을 업로드하거나 필드에 입력합니다.

Choose file

```
#!/bin/bash
yum update -y
amazon-linux-extras install -y lamp-mariadb10.2-php7.2 php7.2
yum install -y httpd
systemctl start httpd
systemctl enable httpd|
```

3. WebServer EC2 인스턴스는 PHP와 함께 Apache 웹 서버를 설치한다.
4. 웹 서버가 제대로 설치되고 시작되었는지 직접 웹 브라우저를 열고 WebServer EC2 인스턴스의 [퍼블릭 Domain Name System(DNS)] 이름을 입력한다.



5. Tabby와 같은 SSH Client Tool로 WebServer EC2에 접속한 다음, 다음과 같은 내용으로 /var/www/html/index.php 파일을 생성한다.

```
$ sudo nano /var/www/html/index.php
```

```
GNU nano 2.9.8                               /var/www/html/index.php

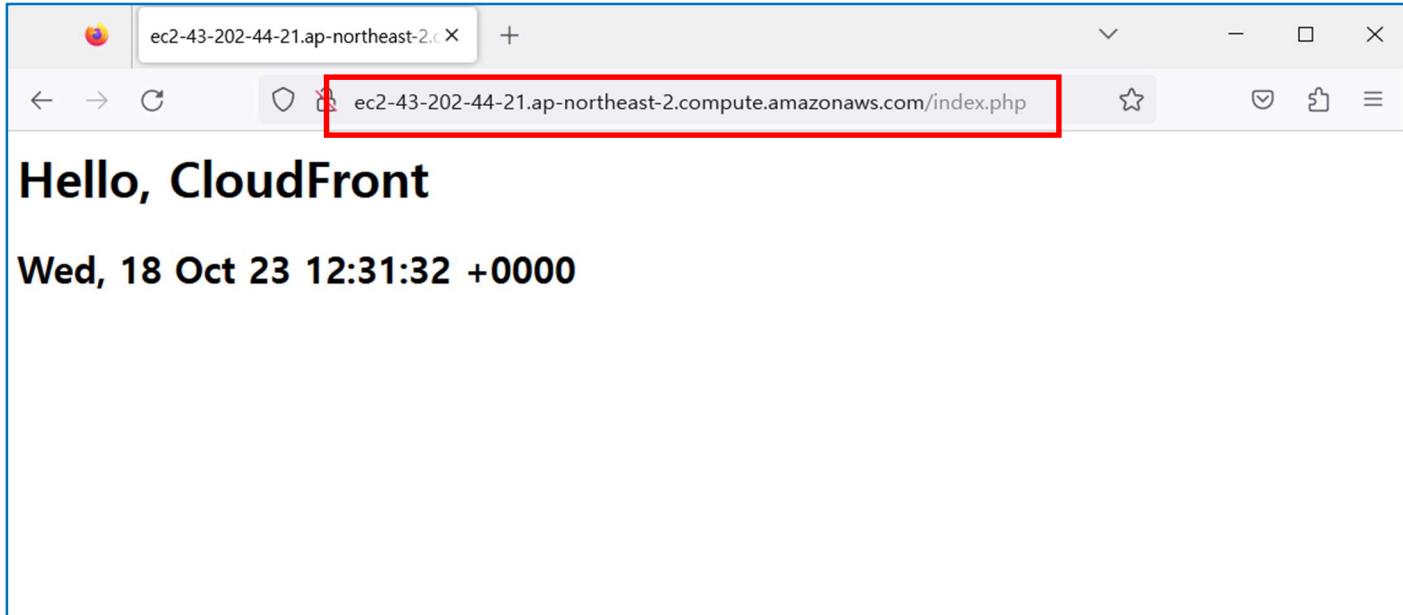
<?php
    sleep(1);
?>
<html>
    <body>
        <h1>Hello, CloudFront</h1>
        <h2><?php
            print(date(DATE_RFC822));
?></h2>
    </body>
</html>
```

[Read 11 lines]

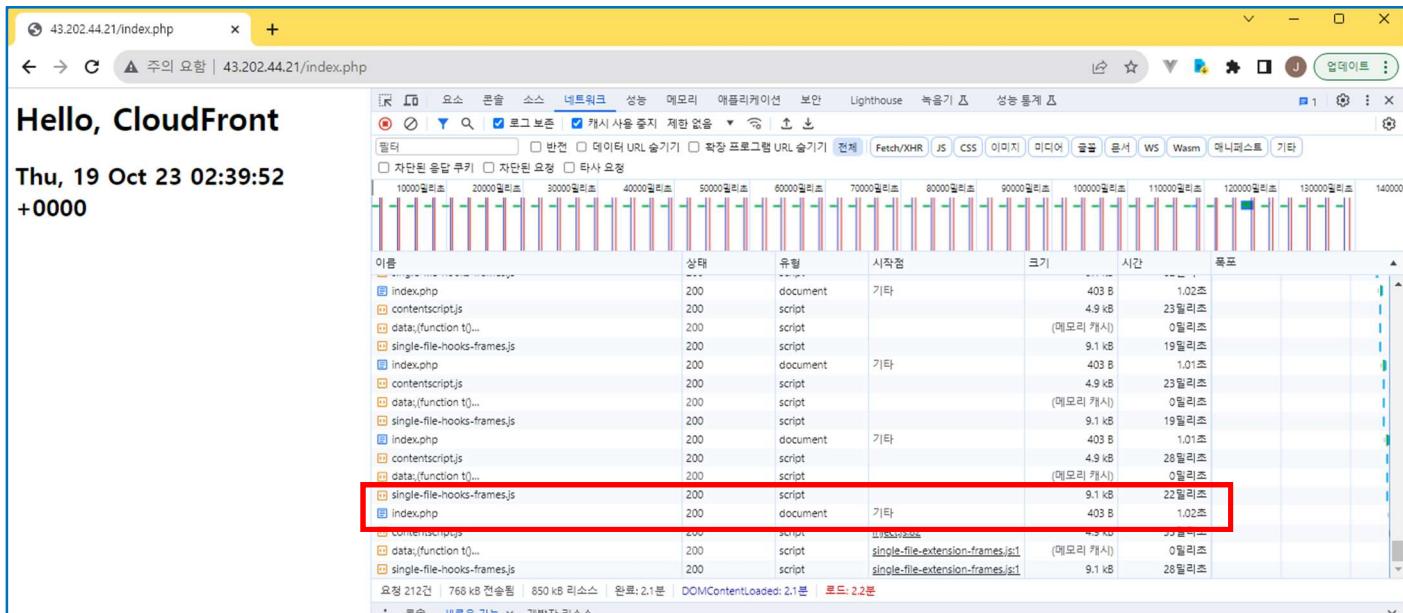
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line

6. 키보드의 **Ctr + O**를 눌러서 파일을 저장하고, **Ctrl + X**를 눌러서 파일을 나온다. 그리고 다음과 같이 [퍼블릭 IPv4 DNS 주소]를 이용하여 웹브라우저에서 접속한다.

[http://\[퍼블릭 IPv4 DNS 주소 or Public IP Address\]/index.php](http://[퍼블릭 IPv4 DNS 주소 or Public IP Address]/index.php)



7. Web Browser에서 키보드의 **F12**를 눌러서 브라우저의 개발자도구를 오픈한다. 그 중에 [네트워크] 탭을 선택한다. 현재 index.php 파일을 로딩하는 데 시간이 1초 조금 더 걸리는 것을 확인할 수 있다.



8. 그 이유는 코드상에서 1초간 정지하고 계속 다음을 진행하기 때문이다. 즉, 웹서버는 이 파일을 1초간 정지 했다가 클라이언트의 웹 브라우저에게 전송하기 때문에 클라이언트의 웹 브라우저는 1초 이상이 걸리는 것이다. 또한 Tabby 같은 SSH Client Tool을 통해 EC2 Instance로 접속하여 로그를 확인해 보자. Apache Web Server는 접속이 들어올 때마다 액세스 로그라는 텍스트 파일에 접속자에 대한 정보를 저장한다. 그래서 다음과 같이 해당 파일에 접속해서 파일의 끝 부분을 확인해 보기로 한다.

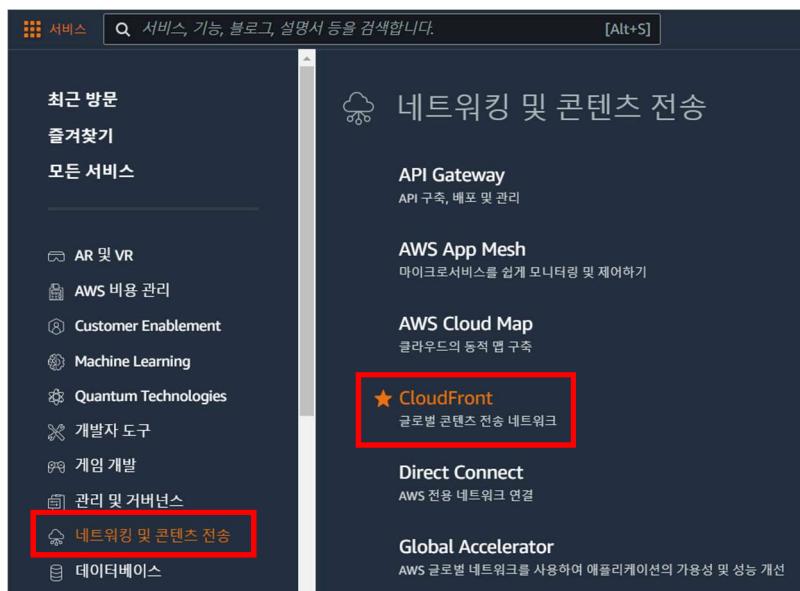
```
$ sudo tail -f /var/log/httpd/access_log
```

```
182.208.131.42 -- [19/Oct/2023:03:04:37 +0000] "GET /index.php HTTP/1.1" 200 203 "http://43.202.44.21/index.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36"
182.208.131.42 -- [19/Oct/2023:03:04:40 +0000] "GET /index.php HTTP/1.1" 200 203 "http://43.202.44.21/index.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36"
182.208.131.42 -- [19/Oct/2023:03:04:45 +0000] "GET /index.php HTTP/1.1" 200 203 "http://43.202.44.21/index.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36"
182.208.131.42 -- [19/Oct/2023:03:04:47 +0000] "GET /index.php HTTP/1.1" 200 203 "http://43.202.44.21/index.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36"
182.208.131.42 -- [19/Oct/2023:03:04:50 +0000] "GET /index.php HTTP/1.1" 200 203 "http://43.202.44.21/index.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36"
```

9. 클라이언트로부터 요청이 들어오는 순간 로그에도 계속 기록하는 것을 확인할 수 있다.

CloudFront 생성하기

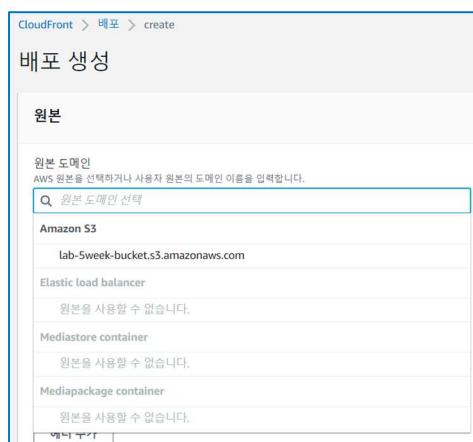
10. CloudFront를 생성하기 위해 [서비스] > [네트워킹 및 콘텐츠 전송] > [CloudFront] 메뉴를 클릭한다.



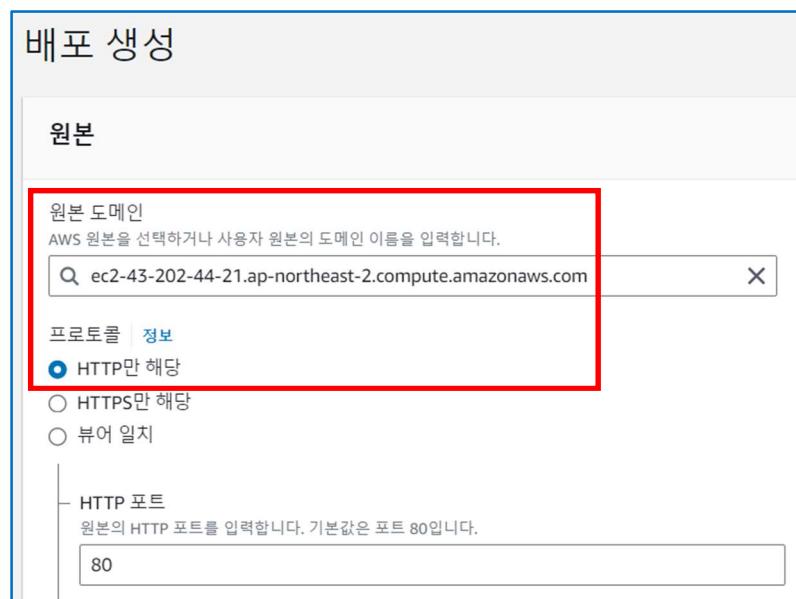
11. Amazon CloudFront 페이지로 들어왔다. 화면상 오른쪽에 있는 [CloudFront 배포 생성]을 클릭한다.



12. [배포 생성] 페이지로 들어왔다. [원본 도메인]은 오리진 웹 서버의 주소를 넣는다. 만일 S3를 사용하고 있다면 Bucket의 주소를 넣으면 된다.



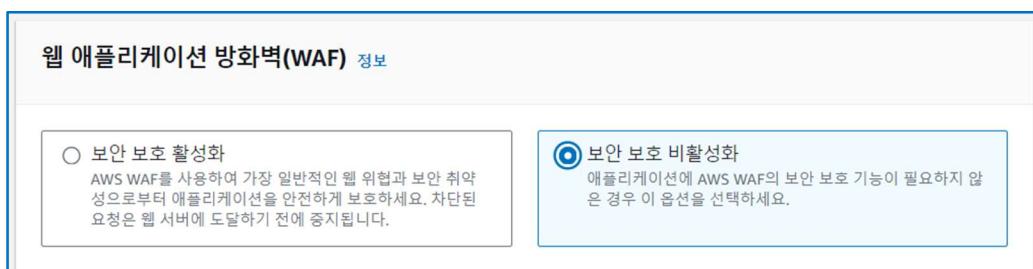
13. 그렇지 않고 이번 실습처럼 직접 웹 서버를 운영하고 있다면, 그 주소를 복사해서 붙여 넣으면 된다. 주의할 점은 http와 같은 프로토콜을 넣으면 안된다. 또한 파일까지의 전체 경로를 넣지 않고 도메인명까지만 입력해야 한다. 나머지 값은 기본값 그대로 사용한다. 또한 [프로토콜]은 **HTTP만 해당**을 선택한다.



14. 페이지를 계속 아래로 스크롤다운하여 **[캐시 키 및 원본 요청]** 섹션에서 **[캐시 정책]**은 목록에서 **CachingOptimized**를 선택한다.



15. **[웹 애플리케이션 방화벽(WAF)]**은 **보안 보호 비활성화**를 선택한다.



16. 나머지 값을 즉, [기본 캐시 동작] 섹션, [함수 연결] 섹션, 그리고 [설정] 섹션 모두 기본값 그대로 사용하기로 한다. [배포 생성]을 클릭한다.

지원되는 HTTP 버전
추가 HTTP 버전에 대한 지원을 추가합니다. HTTP/1.0 및 HTTP/1.1이 기본값으로 지원됩니다.
 HTTP/2

기본값 루트 객체 - 선택 사항
뷰어가 특정 객체 대신 루트 URL(/)을 요청할 때 반환할 객체(파일 이름)입니다.

표준 로깅
Amazon S3 버킷으로 전송된 뷰어 요청의 로그를 가져옵니다.
 끄기
 켜기

IPv6
 끄기
 켜기

설명 - 선택 사항

취소 **배포 생성**

17. 배포 생성이 성공됐다. 이 페이지에서 중요한 부분은 [일반] 탭의 [배포 도메인 이름] 값이다. 바로 이 주소가 CloudFront의 주소이다. 이 주소를 복사한다.

CloudFront > 배포 > E2T6KQZD0ZKAP2

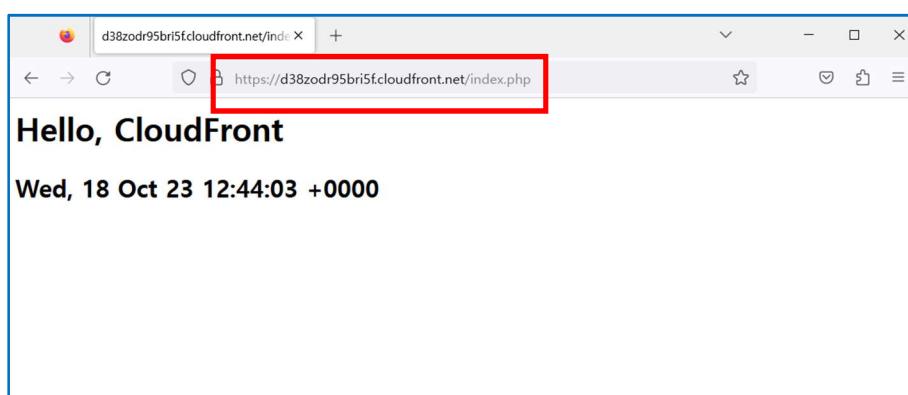
E2T6KQZD0ZKAP2

일반 | 원본 | 동작 | 오류 페이지 | 지리적 제한 | 무효화 | 태그

세부 정보

배포 도메인 이름 <input checked="" type="text"/> d38zodr95bri5f.cloudfront.net	ARN <input checked="" type="text"/> arn:aws:cloudfront::789534828835:distribution/E2T6KQZD0ZKAP2
--	---

18. 이 주소로 웹 브라우저에서 접속해 본다. 복사한 주소 제일 뒤에 /index.php를 붙이고 엔터를 치면 우리가 앞에서 만들었던 웹페이지를 확인할 수 있다.



19. 웹 브라우저에서 F12를 눌러서 크롬 개발자 도구를 오픈한다. [네트워크] 탭으로 이동하여 웹 페이지를 다시 리프레시(F5)하면 로딩 시간이 표시되는데, 현재 17ms 즉 매우 짧은 시간에 로딩되는 것을 알 수 있다. 그런데, 여기서 주의해서 볼 점은 아무리 페이지를 다시 로딩해도 웹 페이지를 접속하는 시간을 표시하는 내용은 변하지 않는다는 것이다. 즉 꽤 오랜 시간 화면의 값이 변하지 않는다는 것을 알 수 있다. 이 상황의 이유는 **Cache** 때문이다.

이름	상태	유형	시작점	크기	시간
index.php	200	document	기타	376 B	17밀리초
contentscript.js	200	script		4.9 kB	33밀리초
single-file-hooks-frames.js	200	script		9.1 kB	29밀리초
index.php	200	document	기타	367 B	31밀리초
contentscript.js	200	script		4.9 kB	57밀리초
data:(function t()...	200	script		9.1 kB	0밀리초
single-file-hooks-frames.js	200	script		9.1 kB	56밀리초
index.php	200	document	기타	368 B	15밀리초
contentscript.js	200	script		4.9 kB	14밀리초
data:(function t()...	200	script		9.1 kB	0밀리초
single-file-hooks-frames.js	200	script		9.1 kB	10밀리초
index.php	200	document	기타	367 B	17밀리초
contentscript.js	200	script	inject.js:2	4.9 kB	14밀리초
data:(function t()...	200	script	single-file-extension-frames.js:1	9.1 kB	0밀리초
single-file-hooks-frames.js	200	script	single-file-extension-frames.js:1	9.1 kB	10밀리초

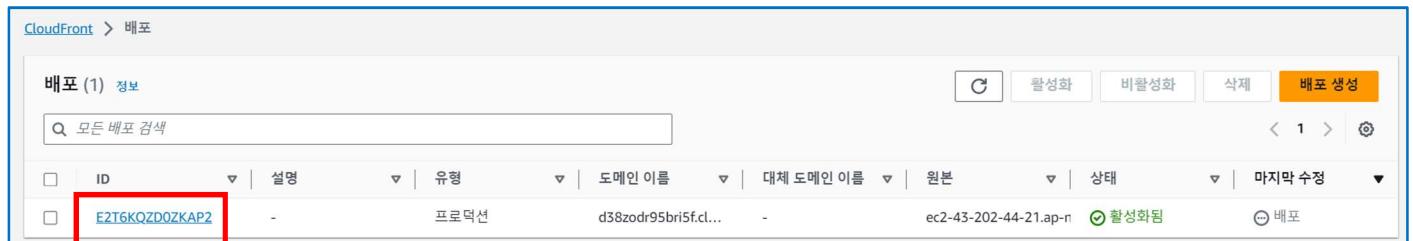
20. 확인하는 방법은 index.php를 선택하고 [헤더]의 값에서 X-Cache를 확인하면 된다. 값이 **Hit from cloudfront**라고 보인다.

이름	헤더	미리보기	응답	시작점	타이밍
index.php					
contentscript.js					
data:(function t()...					
single-file-hooks-frames.js					
index.php					
contentscript.js					
data:(function t()...					
single-file-hooks-frames.js					
index.php					
contentscript.js					
data:(function t()...					
single-file-hooks-frames.js					
index.php					
contentscript.js					
data:(function t()...					
single-file-hooks-frames.js					

21. 한편, 로그파일의 내용을 확인해 보자. 마지막 로그 시간 이후 계속 로깅이 되지 않는 것을 확인할 수 있다. 즉, CloudFront는 열심히 일을 하지만, WebServer는 전혀 일을 하지 않는다는 것을 확인할 수 있다.

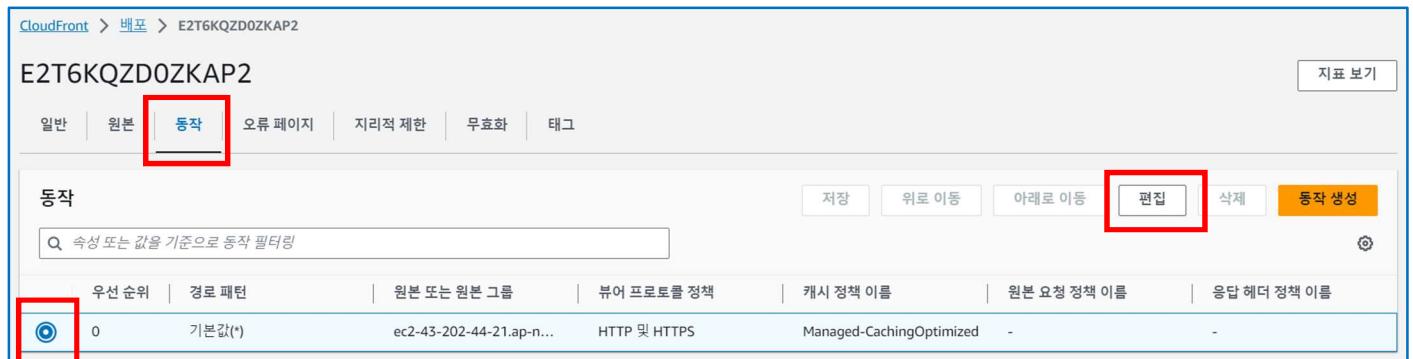
```
182.208.131.42 - - [19/Oct/2023:03:11:08 +0000] "GET /index.php HTTP/1.1" 200 203 "http://43.202.44.21/index.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36"
182.208.131.42 - - [19/Oct/2023:03:11:10 +0000] "GET /index.php HTTP/1.1" 200 203 "http://43.202.44.21/index.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36"
182.208.131.42 - - [19/Oct/2023:03:11:11 +0000] "GET /index.php HTTP/1.1" 200 203 "http://43.202.44.21/index.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36"
182.208.131.42 - - [19/Oct/2023:03:11:13 +0000] "GET /index.php HTTP/1.1" 200 203 "http://43.202.44.21/index.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36"
182.208.131.42 - - [19/Oct/2023:03:11:18 +0000] "GET /index.php HTTP/1.1" 200 203 "http://43.202.44.21/index.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36"
182.208.131.42 - - [19/Oct/2023:03:11:20 +0000] "GET /index.php HTTP/1.1" 200 203 "http://43.202.44.21/index.php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36"
```

22. 원본 페이지 즉 오리진의 내용이 변경되었음에도 배포가 그 내용을 반영하지 못하기 때문이다. 성능을 적당히 발휘하면서 그러면서도 최신의 내용을 배포하려면 어떻게 해야 할까? **CloudFront**의 **[배포]** 페이지로 이동한다. 배포 **[ID]**의 링크를 클릭한다.



The screenshot shows the CloudFront console with the 'Distribution' tab selected. A single distribution is listed with the ID 'E2T6KQZD0ZKAP2'. This ID is highlighted with a red box. The distribution is marked as '활성화됨' (Enabled). Other columns include 'ID', '설명', '유형', '도메인 이름', '대체 도메인 이름', '원본', '상태', and '마지막 수정'.

23. **[동작]** 탭을 클릭한다. **[경로 패턴]**을 보면 기본값(*)으로 되어 있다. 즉, **CloudFront**로 들어오는 모든 경로의 동작 방법을 설명하고 있다. 목록의 옵션 버튼을 클릭하면 **[편집]** 버튼이 활성화 된다. **[편집]**을 클릭한다.



The screenshot shows the 'Behaviors' section of the distribution configuration. The 'Path pattern' dropdown is set to '기본값(*)'. The 'Edit' button is highlighted with a red box. Other tabs visible include 'General', 'Origin', 'Custom Page', 'Geographic Restriction', 'Cache Settings', and 'Tags'. Below the tabs, there are buttons for 'Save', 'Move Up', 'Move Down', 'Edit', and 'Delete'.

24. **[동작 편집]** 페이지로 들어왔다. 페이지를 아래로 스크롤다운하여 **[캐시 키 및 원본 요청]** 섹션으로 이동한다. **[Legacy cache settings]**를 선택하면 아래 여러 개의 항목들이 활성화되는데, 여기서 **[객체 캐싱]** 항목을 확인한다. 기본 값은 **[Use origin cache headers]**가 선택되어 있다. 즉, 오리진의 캐시와 관련된 헤더를 사용해서 캐시를 **CloudFront**에서 얼마나 사용할 것인가를 지정하겠다는 의미이다. 여기서 말하는 헤더란 **Cache-Control**을 의미한다. 그 기간 동안 **CloudFront**는 오리진 웹 서버에게 요청하지 않는 것이다.

캐시 키 및 원본 요청

캐시 정책 및 원본 요청 정책을 사용하여 캐시 키 및 원본 요청을 제어할 것을 권장합니다.

Cache policy and origin request policy (recommended)

Legacy cache settings

- 헤더

캐시 키에 포함할 헤더를 선택합니다.

없음

- 쿼리 문자열

캐시 키에 포함할 쿼리 문자열을 선택합니다.

없음

- 쿠키

캐시 키에 포함할 쿠키를 선택합니다.

없음

- 캐시 캐싱

Use origin cache headers

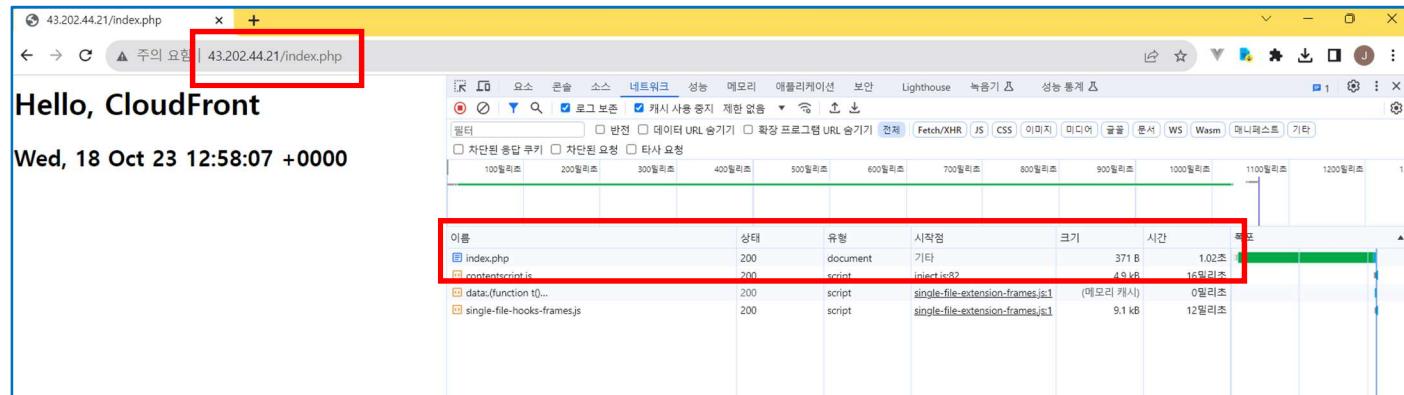
Customize

25. **Apache Web Server**는 접속이 들어올 때마다 **액세스 로그**라는 텍스트 파일에 접속자에 대한 정보를 저장한다. 그래서 다음과 같이 해당 파일에 접속해서 파일의 끝 부분을 확인해 보기로 한다.

```
$ sudo tail -f /var/log/httpd/access_log
```

```
[ec2-user@ip-10-0-10-158 ~]$ sudo tail -f /var/log/httpd/access_log
121.136.18.98 - - [18/Oct/2023:12:27:13 +0000] "GET / HTTP/1.1" 403 3630 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/118.0"
121.136.18.98 - - [18/Oct/2023:12:27:14 +0000] "GET /icons/apache_pb2.gif HTTP/1.1" 200 4234 "http://ec2-43-202-44-21.ap-northeast-2.compute.amazonaws.com/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/118.0"
121.136.18.98 - - [18/Oct/2023:12:27:14 +0000] "GET /favicon.ico HTTP/1.1" 404 196 "http://ec2-43-202-44-21.ap-northeast-2.compute.amazonaws.com/" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/118.0"
121.136.18.98 - - [18/Oct/2023:12:31:31 +0000] "GET /index.php HTTP/1.1" 200 182 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/118.0"
139.162.84.205 - - [18/Oct/2023:12:43:46 +0000] "GET / HTTP/1.0" 200 182 "-" "Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/109.0.0.0 Safari/537.36"
13.124.199.12 - - [18/Oct/2023:12:44:02 +0000] "GET /index.php HTTP/1.1" 200 182 "-" "Amazon CloudFront"
13.124.199.12 - - [18/Oct/2023:12:44:03 +0000] "GET /favicon.ico HTTP/1.1" 404 196 "-" "Amazon CloudFront"
13.124.199.12 - - [18/Oct/2023:12:45:23 +0000] "GET /favicon.ico HTTP/1.1" 404 196 "-" "Amazon CloudFront"
```

26. 그리고 **CloudFront**가 아닌 웹서버(EC2 WebServer Instance의 Public IP 사용)에 직접 새로 페이지를 로딩해 본다. 그러면 로딩 속도가 1초 이상 걸리는 것을 확인할 수 있고, 새롭게 로그 기록이 올라오는 것을 확인할 수 있다.

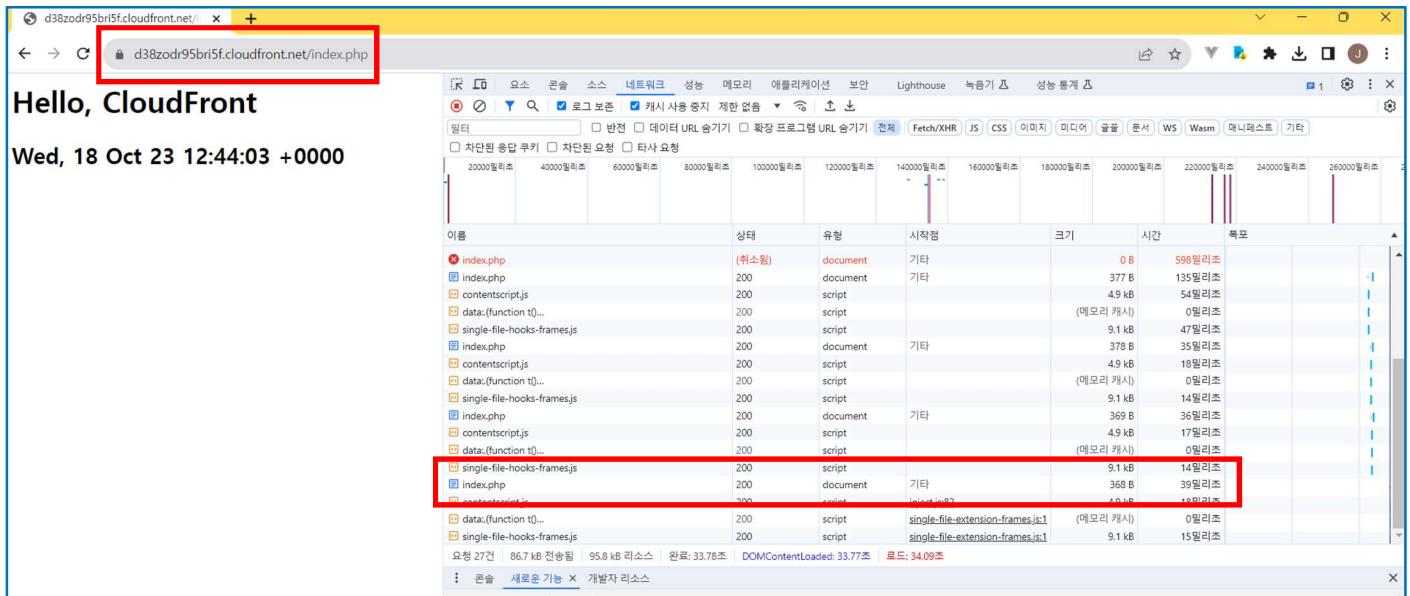


```

13.124.199.12 - - [18/Oct/2023:12:44:03 +0000] "GET /favicon.ico HTTP/1.1" 404 196 "-" "Amazon CloudFront"
13.124.199.12 - - [18/Oct/2023:12:45:23 +0000] "GET /favicon.ico HTTP/1.1" 404 196 "-" "Amazon CloudFront"
121.136.18.98 - - [18/Oct/2023:12:57:04 +0000] "GET /php HTTP/1.1" 404 196 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36"
121.136.18.98 - - [18/Oct/2023:12:57:05 +0000] "GET /favicon.ico HTTP/1.1" 404 196 "http://43.202.44.21/php" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36"
Safari/537.36"
121.136.18.98 - - [18/Oct/2023:12:57:08 +0000] "GET /php HTTP/1.1" 404 196 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36"
121.136.18.98 - - [18/Oct/2023:12:57:27 +0000] "GET /index.php HTTP/1.1" 200 102 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36"
121.136.18.98 - - [18/Oct/2023:12:57:35 +0000] "GET /index.php HTTP/1.1" 200 102 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36"
121.136.18.98 - - [18/Oct/2023:12:57:58 +0000] "GET /index.php HTTP/1.1" 200 102 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36"
121.136.18.98 - - [18/Oct/2023:12:58:06 +0000] "GET /index.php HTTP/1.1" 200 102 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36"

```

27. 하지만, **CloudFront**에 접속하는 페이지로 테스트해보면 접속시간은 매우 빠르지만 로그파일에는 아무 변화가 없는 것을 알 수 있다. 역시 웹서버는 작업하지 않지만, **CloudFront**는 일하고 있다.



```

121.136.18.98 - - [18/Oct/2023:12:57:08 +0000] "GET /php HTTP/1.1" 404 196 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36"
121.136.18.98 - - [18/Oct/2023:12:57:27 +0000] "GET /index.php HTTP/1.1" 200 102 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36"
121.136.18.98 - - [18/Oct/2023:12:57:35 +0000] "GET /index.php HTTP/1.1" 200 102 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36"
121.136.18.98 - - [18/Oct/2023:12:57:58 +0000] "GET /index.php HTTP/1.1" 200 102 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36"
121.136.18.98 - - [18/Oct/2023:12:58:06 +0000] "GET /index.php HTTP/1.1" 200 102 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36"
121.136.18.98 - - [18/Oct/2023:13:00:15 +0000] "GET /index.php HTTP/1.1" 200 102 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36"
121.136.18.98 - - [18/Oct/2023:13:00:20 +0000] "GET /index.php HTTP/1.1" 200 102 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36"
121.136.18.98 - - [18/Oct/2023:13:00:23 +0000] "GET /index.php HTTP/1.1" 200 102 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36"
121.136.18.98 - - [18/Oct/2023:13:00:24 +0000] "GET /index.php HTTP/1.1" 200 102 "-" "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36"

```

28. 이것이 원래 캐시의 목적이지만, 실습 목적 상 수정하려고 한다. 오리진 웹 서버의 정보가 사용자가 요청할 때마다 반응하는 즉, 캐시를 하지 않지는 않지만, 그렇다고 꽤 오랫동안 바뀌지는 않는 그런 상황이 아니라, 일정 시간 동안만 캐시를 하려고 한다. 그렇게 하려면 웹 브라우저에 설정을 해야 한다. 웹 서버에서 웹 브라우저에게 페이지를 다운로드할 때 그 캐시 시간을 지정하면 된다. **Index.php** 파일을 오픈해서 다음과 같이 입력한다.

```
$ sudo nano /var/www/html/index.php
```

```

<?php
    header("Cache-Control: max-age=10");
    sleep(1);
?>
<html>
    <head>
        <!-- <meta http-equiv="refresh" content="1"> -->
    </head>
    <body>
        <h1>Hello, CloudFront</h1>
        <h2><?php
            print(date(DATE_RFC822));
        ?></h2>
        <h3><a href="/index.php">To me</a></h3>
    </body>
</html>

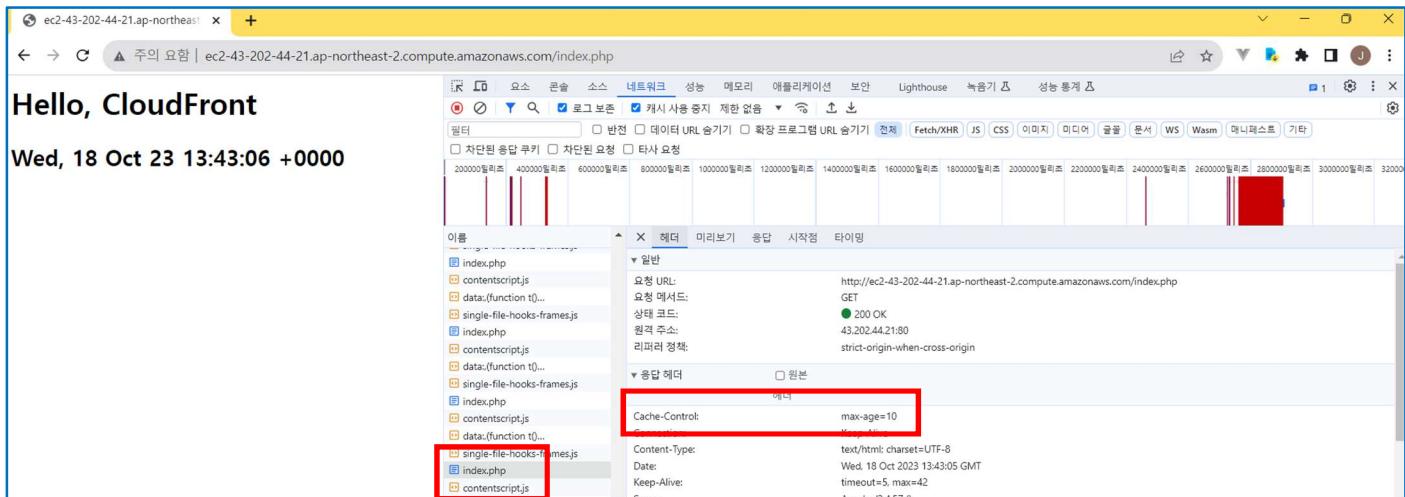
```

29. 파일을 저장하고 다시 웹 페이지를 새로 로딩해 본다. **CloudFront**를 사용하지 않는 웹 페이지이다. 로딩 시간은 디스크 캐시를 사용할 때는 2ms, 디스크 캐시를 사용하지 않으면 1초보다 조금 더 걸린다. 목록에서 **index.php**를 선택한 후, [헤더] 탭을 선택한다. 다음 이미지는 디스크 캐시를 사용할 때이다.

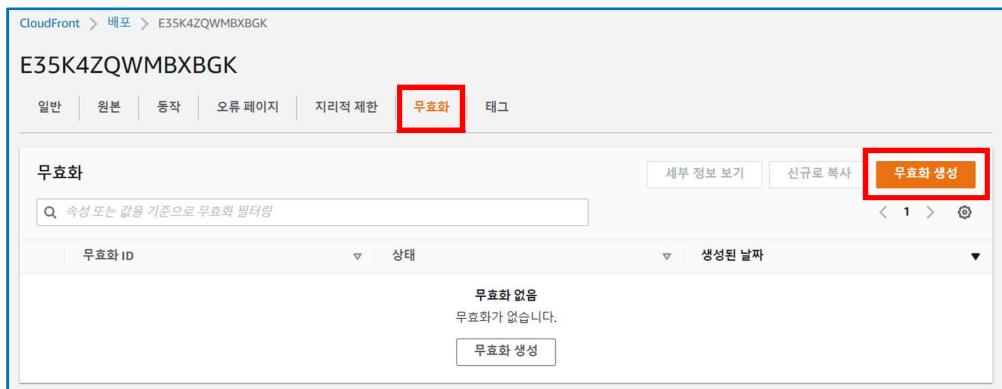
이름	상태	유형	시작점	크기	시간	목표
index.php	200	document	기타	509 B	1.02초	
contentscript.js	200	script		4.9 kB	17밀리초	(메모리 캐시)
data:(function t()...	200	script		9.1 kB	14밀리초	
single-file-hooks-frames.js	200	script		9.1 kB	3밀리초	
index.php	200	document	기타	509 B	1.02초	(디스크 캐시)
contentscript.js	200	script		4.9 kB	18밀리초	(메모리 캐시)
data:(function t()...	200	script		9.1 kB	14밀리초	
single-file-hooks-frames.js	200	script		9.1 kB	20밀리초	(메모리 캐시)
index.php	200	document	기타	509 B	1.02초	(디스크 캐시)
contentscript.js	200	script	single-file-extension-frames/s/1	4.9 kB	16밀리초	(메모리 캐시)
data:(function t()...	200	script	single-file-extension-frames/s/1	9.1 kB	0밀리초	
single-file-hooks-frames.js	200	script	single-file-extension-frames/s/1	9.1 kB	39밀리초	

상태 코드:	200 OK (디스크 캐시에서)
원격 주소:	43.202.44.21:80
리퍼러 정책:	strict-origin-when-cross-origin

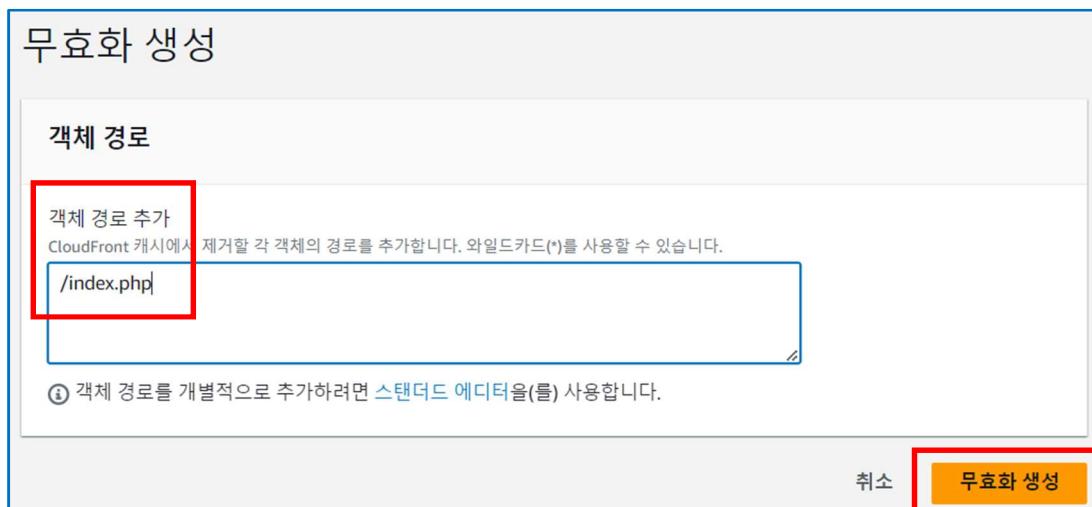
30. **Cache-Control**값을 확인할 수 있다. 즉, 디스크 캐시를 사용하면 로딩 속도는 많이 빠르게 되지만, 캐시를 사용하지 않으면 로딩 속도는 크게 차이가 나는 것을 확인할 수 있다. 현재는 최대 10초 동안 캐시를 하는 것을 알 수 있다.



31. 이번에는 캐시를 강제로 지우는 방법을 살펴보기로 한다. **CloudFront** 페이지에서 실습에서 생성한 배포 ID의 링크를 클릭해서 이동한 후, [무효화] 탭을 클릭한다. [무효화] 페이지에서 [무효화 생성]을 클릭한다.



32. [무효화 생성] 페이지에서 [객체 경로 추가]의 목록에 /index.php를 입력하고 [무효화 생성]을 클릭한다.



33. [무효화 세부 정보] 페이지에서 [상태]가 완료됨을 확인한다.

The screenshot shows the CloudFront Invalidation Details page. At the top, a green bar indicates the invalidation was successfully generated. Below it, the navigation path is CloudFront > 배포 > E2T6KQZD0ZKAP2 > I7TT3GDVKV8WB3EOTYT1UYEUMU. The main section is titled '무효화 세부 정보'. It displays the creation time as '2023년 10월 18일 오후 1시 45분 29초 UTC' and the object key as '/index.php'. A red box highlights the '상태' (Status) field, which shows '완료됨' (Completed).

34. index.php 무효화한 후, 다시 브라우저에서 확인해 보자. 다음 그림을 디스크 캐시를 하지 않을 때 화면이다.

The screenshot shows the Network tab in a browser developer tools window for the URL d38zodr95bri5f.cloudfront.net/index.php. The timeline at the top shows various requests. The table below lists the network requests:

이름	상태	유형	시작점	크기	시간	통로
index.php	200	document	기타	507 B	17밀리초	
contentscript.js	200	script		4.9 kB	45밀리초	
data:(function t0...	200	script			(메모리 캐시)	0밀리초
single-file-hooks-frames.js	200	script		9.1 kB	40밀리초	
index.php	200	document	기타	502 B	1.02초	
contentscript.js	200	script		4.9 kB	26밀리초	
data:(function t0...	200	script			(메모리 캐시)	0밀리초
single-file-hooks-frames.js	200	script		9.1 kB	25밀리초	
index.php	200	document	기타	502 B	6밀리초	
contentscript.js	200	script		4.9 kB	42밀리초	
data:(function t0...	200	script			(메모리 캐시)	0밀리초
single-file-hooks-frames.js	200	script		9.1 kB	39밀리초	
index.php	200	document	기타	506 B	17밀리초	
contentscript.js	200	script	inject:is:t2	4.9 kB	61밀리초	
data:(function t0...	200	script			(메모리 캐시)	0밀리초
single-file-hooks-frames.js	200	script		9.1 kB	70밀리초	

요청 2434건 | 11.2 MB 전송됨 | 16.6 MB 리소스 | 원본: 31.4분 | 로드: 31.4분

35. 그리고, 디스크 캐시를 할 때의 그림이다.

The screenshot shows the Network tab in a browser developer tools window for the URL d38zodr95bri5f.cloudfront.net/index.php. The timeline at the top shows various requests. The table below lists the network requests:

이름	상태	유형	시작점	크기	시간	통로
index.php	200	document	(디스크 캐시)	507 B	6밀리초	
contentscript.js	200	script		4.9 kB	42밀리초	
data:(function t0...	200	script			(메모리 캐시)	0밀리초
single-file-hooks-frames.js	200	script		9.1 kB	39밀리초	
index.php	200	document	기타	506 B	17밀리초	
contentscript.js	200	script		4.9 kB	61밀리초	
data:(function t0...	200	script			(메모리 캐시)	0밀리초
single-file-hooks-frames.js	200	script		9.1 kB	70밀리초	
index.php	200	document	기타	501 B	1.03초	
contentscript.js	200	script		4.9 kB	30밀리초	
data:(function t0...	200	script			(메모리 캐시)	0밀리초
single-file-hooks-frames.js	200	script		9.1 kB	39밀리초	
index.php	200	document	기타	501 B	3밀리초	
contentscript.js	200	script	inject:is:t2	4.9 kB	16밀리초	
data:(function t0...	200	script			(메모리 캐시)	0밀리초
single-file-hooks-frames.js	200	script		9.1 kB	39밀리초	

요청 2442건 | 11.2 MB 전송됨 | 16.6 MB 리소스 | 원본: 32.1분 | DOMContentLoaded: 31.1분 | 로드: 32.1분

36. index.php를 무효화하면 다음과 같이 X-Cache의 값이 Miss from cloudfront로 변경됨을 알 수 있다.

The screenshot shows the Network tab of a browser's developer tools. A request for 'index.php' is selected. The response headers section shows the following:

Header	Value
Cache-Control	max-age=10
Content-Type	text/html; charset=UTF-8
Date	Thu, 19 Oct 2023 03:33:37 GMT
Server	Apache/2.4.57 (
Via	1.1 8fe450b14ab15570329ffffc38b89966.cloudfront.net (CloudFront)
X-Amz-Cf-Id	pvZTUI0DB_1fURoUj4Lo5mjaeTE2aZ2wTqvADlFVfr786aIC8G9Q==
X-Amz-Cf-Pop	ICN54-C3
X-Cache	Miss from cloudfront
X-Powered-By	PHP/7.2.34

37. index.php를 무효화하면 로그파일은 클라이언트가 요청할 때마다 계속 로깅이 된다. 하지만 무효화한 것을 취소하면 다시 일정 시간 로깅을 하지 않게 된다. 다음과 같이 [객체 경로]를 /*로 등록하고 로그 파일을 확인하면 일정시간 로깅이 되지 않는 것을 확인할 수 있다.

The screenshot shows the CloudFront Log Configuration page. It displays a log entry for the file 'index.php' with a red box highlighting the object key '/'. The log entry details are as follows:

생성된 날짜
2023년 10월 19일 오전 3시 39분 29초 UTC

상태
완료됨

객체 경로
/*

```
13.124.199.72 - - [19/Oct/2023:03:38:18 +0000] "GET /index.php HTTP/1.1" 200 213 "-" "Amazon CloudFront"
13.124.199.72 - - [19/Oct/2023:03:38:18 +0000] "GET /index.php HTTP/1.1" 200 213 "-" "Amazon CloudFront"
13.124.199.26 - - [19/Oct/2023:03:38:56 +0000] "GET /index.php HTTP/1.1" 200 213 "-" "Amazon CloudFront"
13.124.199.26 - - [19/Oct/2023:03:40:07 +0000] "GET /index.php HTTP/1.1" 200 213 "-" "Amazon CloudFront"
13.124.199.72 - - [19/Oct/2023:03:40:23 +0000] "GET /index.php HTTP/1.1" 200 213 "-" "Amazon CloudFront"
13.124.199.127 - - [19/Oct/2023:03:40:36 +0000] "GET /index.php HTTP/1.1" 200 213 "-" "Amazon CloudFront"
13.124.199.26 - - [19/Oct/2023:03:43:13 +0000] "GET /index.php HTTP/1.1" 200 213 "-" "Amazon CloudFront"
```

38. 보다 더 세밀하게 캐시를 CloudFront에서 수정해 보자. 배포 상세 페이지에서 [동작] 탭으로 이동한 후, 해당 동작을 선택하면 [편집] 버튼이 활성화된다. [편집] 버튼을 클릭한다.

The screenshot shows the CloudFront Distribution Settings page. The 'Actions' tab is selected, indicated by a red box. The 'Edit' button for the first action is also highlighted with a red box. The actions listed are:

우선 순위	경로 패턴	원본 또는 원본 그룹	불어 프로토콜 정책	캐시 정책 이름	원본 요청 정책 이름	응답 헤더 정책 이름
0	기본값(*)	ec2-43-202-44-21.ap-northeast-2.com...	HTTP 및 HTTPS	Managed-CachingOptimized	-	-

39. [동작 편집] 페이지에서 [캐시 키 및 원본 요청] 섹션으로 이동한 후, [Legacy cache settings]를 선택한다. 그리고 [객체 캐싱]을 [Customize]를 선택한다.

캐시 키 및 원본 요청
캐시 정책 및 원본 요청 정책을 사용하여 캐시 키 및 원본 요청을 제어할 것을 권장합니다.

Cache policy and origin request policy (recommended)
 Legacy cache settings

– 헤더
캐시 키에 포함할 헤더를 선택합니다.
없음

– 쿼리 문자열
캐시 키에 포함할 쿼리 문자열을 선택합니다.
없음

– 쿠키
캐시 키에 포함할 쿠키를 선택합니다.
없음

– 객체 캐싱
 Use origin cache headers
 Customize

최소 TTL
최소 TTL(Time to Live)(초)입니다.
0

최대 TTL
최대 TTL(Time to Live)(초)입니다.
31536000

기본 TTL
기본 TTL(Time to Live)(초)입니다.
86400

40. **TTL(Time to Live)**는 캐시를 얼마 동안 살아있게 할 것인가의 최소값이다. 예를 들어, 최소 TTL을 5로, 그리고 최대 TTL을 15로 지정하고 기본 TTL을 10으로 지정하면, 최소 5초 이상, 최대 15초 이하로 유지한다는 뜻이다. 하지만 Origin에서 프로그래밍으로 캐시 지정한 값(max-age)과 CloudFront의 캐시값은 서로 연관되어 있다. 다음과 같이 최소 TTL은 5, 최대 TTL은 15 그리고 기본 TTL은 10으로 지정한다.

– 객체 캐싱
 Use origin cache headers
 Customize

최소 TTL
최소 TTL(Time to Live)(초)입니다.
5

최대 TTL
최대 TTL(Time to Live)(초)입니다.
15

기본 TTL
기본 TTL(Time to Live)(초)입니다.
10

41. 그리고 index.php의 코드에서 max-age의 값을 20으로 수정하면 CloudFront의 범위 안에서만 사용 가능하다. 즉 5 ~ 15초까지 캐시 기간이 유지된다. 그 위에 [쿼리 문자열]을 확인해 보자.

캐시 키 및 원본 요청
캐시 정책 및 원본 요청 정책을 사용하여 캐시 키 및 원본 요청을 제어할 것을 권장합니다.

Cache policy and origin request policy (recommended)
 Legacy cache settings

– 헤더
캐시 키에 포함할 헤더를 선택합니다.
없음

– 쿼리 문자열
캐시 키에 포함할 쿼리 문자열을 선택합니다.
모두

42. 여기서 [쿼리 문자열]의 값을 모두로 변경하면, index.php?id=1, index.php?user=henry 처럼 쿼리 문자열이 다르게 접속할 때마다 CloudFront는 완전히 다른 요청이라고 판단하여 로그에 기록하게 된다.

```
13.124.199.2 - - [19/Oct/2023:03:57:49 +0000] "GET /index.php?id=1 HTTP/1.1" 200 213 "-" "Amazon CloudFront"
13.124.199.2 - - [19/Oct/2023:03:57:56 +0000] "GET /index.php?id=2 HTTP/1.1" 200 213 "-" "Amazon CloudFront"
```

CDN(Content Delivery Network)

- CloudFront 배포 ID의 링크를 클릭하여 상세 페이지로 이동한다. [편집] 버튼을 클릭한다.

The screenshot shows the AWS CloudFront distribution configuration page. At the top, it displays the distribution ID: E2T6KQZD0ZKAP2. Below this, there are tabs for '일반' (General), '원본' (Origin), '동작' (Behavior), '오류 페이지' (Error Page), '지리적 제한' (Geographic Restriction), '무효화' (Invalidate Cache), and '태그' (Tags). A 'List View' button is located in the top right corner. The main area is titled '세부 정보' (Detailed Information) and contains fields for '배포 도메인 이름' (Distribution Domain Name) set to d38zodr95bri5f.cloudfront.net, 'ARN' (Amazon Resource Name) set to arn:aws:cloudfront::789534828835:distribution/E2T6KQZD0ZKAP2, and '마지막 수정' (Last Modified) set to 2023년 10월 18일 오후 12시 42분 21초 UTC. The '설정' (Settings) section includes fields for '설명' (Description) and '대체 도메인 이름' (Alternate Domain Name), both currently empty. On the right side of the settings section, there are buttons for '표준 로깅' (Standard Logging), '끄기' (Turn Off), and '쿠키 로깅' (Cookie Logging). A red box highlights the '편집' (Edit) button in the top right corner of the settings section.

- [설정 편집] 페이지로 들어왔다. [가격 분류] 항목을 보면 현재 [모든 엣지 로케이션에서 사용(최고의 성능)] 옵션에 맞춰져 있다. 엣지 로케이션은 CloudFront의 캐시 서버라고 생각하면 된다.

The screenshot shows the 'Edit Settings' page for distribution E35K4ZQWMBXBGK. The title bar shows the path: CloudFront > 배포 > E35K4ZQWMBXBGK. The main section is titled '설정 편집' (Edit Settings). Under the '설정' (Settings) section, there is a '가격 분류' (Price Class) dropdown. The dropdown menu is open, showing two options: '모든 엣지 로케이션에서 사용(최고의 성능)' (Use in all edge locations (highest performance)) and '부미 및 유럽 및 아시아 및 중동 및 아프리카에서 사용' (Use in Europe and Asia and Middle East and Africa). The first option is highlighted with a red box. Below the dropdown, there is a note: '지불하려는 최고가와 연관된 가격 분류를 선택합니다.' (Select the price class associated with the highest price you are willing to pay.) and a note about AWS WAF Web ACL selection. A 'Web ACL 선택' (Select Web ACL) button is visible at the bottom of the dropdown menu. A red box highlights the '모든 엣지 로케이션에서 사용(최고의 성능)' option in the dropdown menu.

3. **Dotcom-tools.com** 페이지로 이동해 보자. 여기서는 이 사이트에서 운영하는 전 세계의 컴퓨터에서 실습에서 입력한 주소로 접속해서 속도를 테스트한다. [Start Test] 파란색 버튼을 클릭한다.

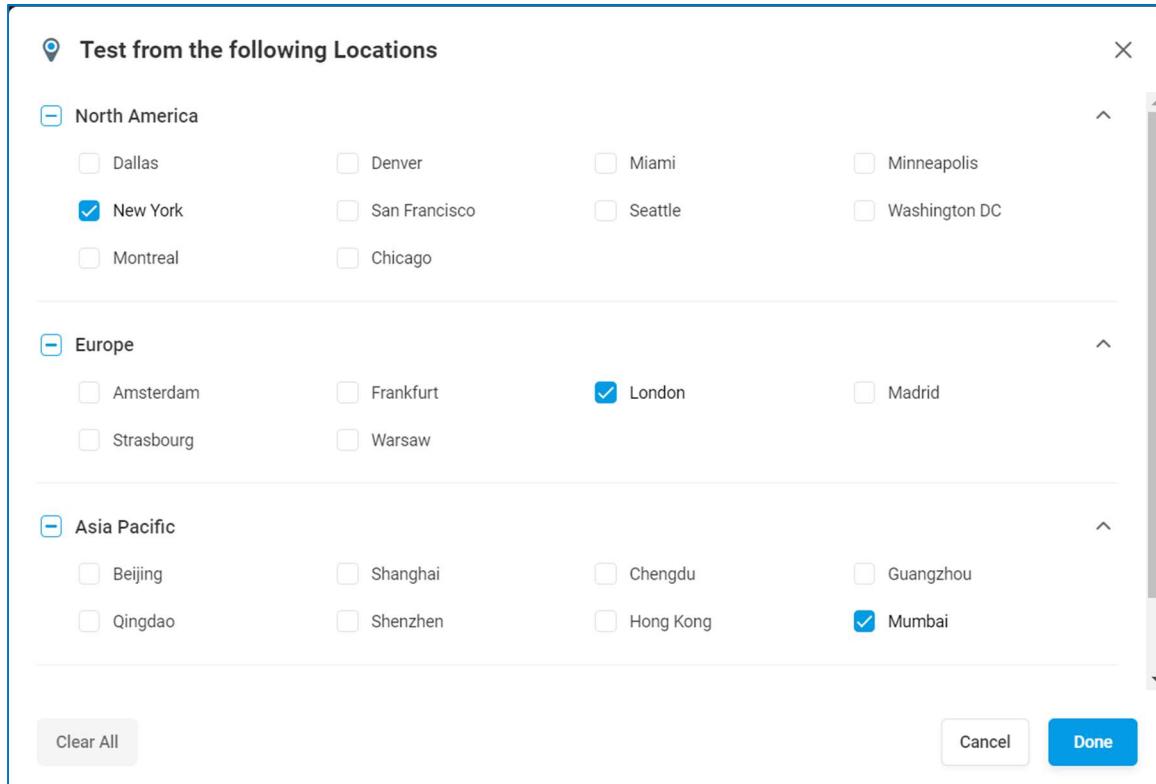
<https://www.dotcom-tools.com/>

The screenshot shows the Dotcom-tools website with a 'Web Server Test' section. A red box highlights the 'Start Test' button. To the right, a modal window titled 'Web Servers Test Summary Report' displays a table of test results from Madrid, Beijing, Sydney, Copenhagen, Seattle, IPv6 San Francisco, and Manila. One entry from Madrid has a 'DNS Error (2007)' message.

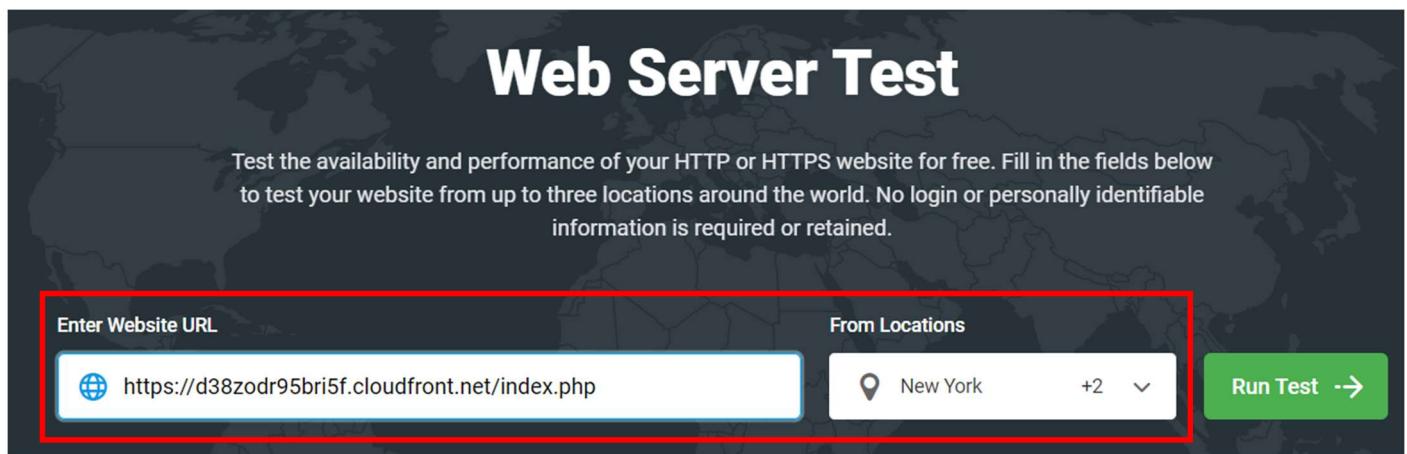
4. [Enter Host] 텍스트박스에 위 실습에서 생성했던 Web Server의 URL을 붙여넣는다. 이 주소는 현재 서울 리전에서 실행되고 있다. 이 사이트에서 무료계정으로 테스트하려면 3개의 지역을 선택할 수 있는데, [From Locations]의 목록을 클릭한다.

The screenshot shows the Dotcom-tools 'Web Server Performance Test' page. The 'Enter Website URL' field contains the IP address 'ec2-43-202-44-21.ap-northeast-2.compute.amazonaws.com'. The 'From Locations' dropdown menu is open, showing 'Denver' and '+2' options. A red box highlights the 'From Locations' dropdown.

5. 실습에서는 3개의 지역을 North America에서는 New York, Europe에서는 London, 그리고 Asia pacific에서는 Mumbai를 선택했다. [Done]을 클릭한다.



6. 그리고 [Enter Host]에 CloudFront 접속 주소를 입력했다.



7. 그리고 테스트하기 전에 CloudFront 캐시를 모두 무효화해야 한다. 위의 Cache 무효화를 참고하여 다음과 같이 [객체 경로]를 /*로 지정한 무효화를 생성한다. [무효화 생성]을 클릭한다.

CloudFront > 배포 > E35K4ZQWMBXBGK > 무효화 생성

무효화 생성

객체 경로

객체 경로 추가

CloudFront 캐시에서 제거할 각 객체의 경로를 추가합니다. 와일드카드(*)를 사용할 수 있습니다.

/*

(i) 객체 경로를 개별적으로 추가하려면 [스탠더드 에디터](#)(를) 사용합니다.

취소 무효화 생성

무효화 I6PZON93ACAYWUOAPY4UZG3SGL은(은) 성공적으로 생성했습니다.

CloudFront > 배포 > E2T6KQZD0ZKAP2 > I6PZON93ACAYWUOAPY4UZG3SGL

무효화 세부 정보

생성된 날짜
2023년 10월 18일 오후 1시 54분 21초 UTC

객체 경로
/*

상태
완료됨

신규로 복사

8. Index.php 코드를 아래와 같이 수정했다.

```
<?php
    header("Cache-Control: max-age=2");
    sleep(1);
?>
<html>
    <head>
        <meta http-equiv="refresh" content="1">
    </head>
    <body>
        <h1>Hello, CloudFront</h1>
        <h2><?php
            print(date(DATE_RFC822));
        ?></h2>
    </body>
</html>
```

9. 아래 그림은 첫 테스트의 결과이다.

The screenshot shows the 'Web Server Test Summary Report' interface. At the top, there are three call-to-action buttons: 'Enable Daily Tests', 'Save Test', and 'Allow up to 30 Locations'. Below these is the URL 'https://d38zodr95bri5f.cloudfront...'. A 'Test Time' section shows '10/18/2023 10:58:15 PM'. On the right, there's a 'Share' button and a 'New Test' button. The main area displays test results for three locations: Mumbai, New York, and London. Each location row includes 'Location', 'Date', 'Time', 'Duration', and 'Status' (OK). The entire table is highlighted with a red box. The 'OK' status is shown in green for all three locations.

Location	Date	Time	Duration	Status
Mumbai	10/18/2023	10:58:16 PM	1.77 s	OK
New York	10/18/2023	10:58:15 PM	1.49 s	OK
London	10/18/2023	10:58:15 PM	1.60 s	OK

10. 다음 그림은 2번째 테스트 결과이다. 첫 테스트의 평균 접속 시간은 **1.62초**이고, 두번째 테스트의 결과는 **2.84초**임을 알 수 있다.

The screenshot shows the 'Web Server Test Summary Report' interface. At the top, there are three call-to-action buttons: 'Enable Daily Tests', 'Save Test', and 'Allow up to 30 Locations'. Below these is the URL 'https://d38zodr95bri5f.cloudfront...'. A 'Test Time' section shows '10/18/2023 11:01:01 PM'. On the right, there's a 'Share' button and a 'New Test' button. The main area displays test results for three locations: Mumbai, New York, and London. Each location row includes 'Location', 'Date', 'Time', 'Duration', and 'Status' (OK). The entire table is highlighted with a red box. The 'OK' status is shown in green for all three locations. The average duration is listed as 2.84 seconds.

Location	Date	Time	Duration	Status
Mumbai	10/18/2023	11:01:01 PM	5.43 s	OK
New York	10/18/2023	11:01:01 PM	1.52 s	OK
London	10/18/2023	11:01:01 PM	1.57 s	OK

11. 그리고 다음 그림은 3번째 테스트 결과이다.

☰ ⏪ HTTP Web Server Test Summary Report Sign up Free

Enable Daily Tests Test results delivered daily by email to your inbox. [click to enable](#) →

Save Test Keep history of all tests run, compare and analyze the differences. [click to enable](#) →

Allow up to 30 Locations Enable all 30 locations around the world to check at once. [click to enable](#) →

<https://d38zodr95bri5f.cloudfront...> ⓘ Options

📅 Test Time: 10/18/2023 11:02:51 PM Share

3 of 3 Locations Checks Complete 0 of 3 Locations Errors From 1.61 Seconds Average Duration

Location	Date	Time	Duration	Status
New York	10/18/2023	11:02:51 PM	1.49 s	OK
London	10/18/2023	11:02:51 PM	1.58 s	OK
Mumbai	10/18/2023	11:02:51 PM	1.77 s	OK

HTTP New Test

Route 53에 도메인 등록하기

- [서비스] > [네트워킹 및 콘텐츠 전송] > [Route 53]를 선택한다.

The screenshot shows the AWS Management Console navigation pane on the left and the main content area on the right. The navigation pane lists various service categories like AR 및 VR, AWS 비용 관리, Customer Enablement, Machine Learning, Quantum Technologies, etc. A red box highlights the '네트워킹 및 콘텐츠 전송' (Networking & Content Delivery) category. Within this category, a red box highlights the '네트워킹 및 콘텐츠 전송' (Networking & Content Delivery) service itself. The main content area displays a list of services under this category, including API Gateway, AWS App Mesh, AWS Cloud Map, CloudFront, Direct Connect, Global Accelerator, Route 53, and VPC. The 'Route 53' service is highlighted with a red box and a star icon, indicating it is the target for domain registration.

- Route 53 대시보드 페이지에 들어왔다. 좌측 메뉴 중 [도메인] > [등록된 도메인]을 클릭한다.
- [등록된 도메인] 페이지이다. [도메인 등록]은 새로운 도메인을 등록하는 데 사용한다. [도메인 등록]을 클릭한다.

The screenshot shows the 'Registered Domains' page in the Route 53 console. At the top, there's a search bar labeled '이름으로 도메인 검색'. To the right of the search bar are buttons for '결제 보고서 다운로드', '내 송신 ▾', and a prominent orange button with a star icon labeled '도메인 등록'. Below these buttons is a pagination control with arrows and a page number '1'. The main table lists registered domains with columns for '도메인 이름', '만료 날짜', '자동 갱신', and '이전 잠금'. Two domains are listed: 'devtrainer.click' and 'henrytrainer.click', both of which are marked as '비활성' (Inactive).

도메인 이름	만료 날짜	자동 갱신	이전 잠금
devtrainer.click	7월 28, 2024, 00:08(UTC:+09:00)	비활성	비활성
henrytrainer.click	7월 28, 2024, 15:54(UTC:+09:00)	비활성	비활성

4. 먼저 등록하고 싶은 도메인을 검색부터 해야 한다. 누군가 이미 사용하고 있다면 당연히 등록할 수 없다. 아래 그림과 같아 **myaws.com**으로 검색하면 사용할 수 없고, AWS가 추천하는 다른 도메인 목록이 나온다.

제안된 사용 가능한 도메인 (10)		
도메인	요금/년	선택
myawsagency.com	13.00 USD	선택
myaws.info	23.00 USD	선택
sellmyaws.com	13.00 USD	선택
myawsnetwork.com	13.00 USD	선택
getmyaws.com	13.00 USD	선택
myaws.tv	32.00 USD	선택
myawssite.com	13.00 USD	선택
myaws.mobi	30.00 USD	선택
myawsapp.net	11.00 USD	선택
myaws.ninja	18.00 USD	선택

5. 이번에는 **myaws.world**로 검색했더니 사용 가능이라고 확인할 수 있다. [장바구니에 추가]를 클릭한다. [장바구니] 섹션에 1년에 12달러의 금액이 필요하다고 나온다. 확인 후 [결제 진행]을 클릭하여 다음 단계로 이동한다.

제안된 사용 가능한 도메인 (10)		
도메인	요금/년	선택됨
myaws.world	12.00 USD	선택됨

6. 다음 단계로 [1단계:요금] 페이지이다. 가격을 확인하고 [다음] 버튼을 클릭한다. [자동 갱신]이 기본적으로 체크되어 있지만, 본 예제에서는 체크 해제하기로 한다.

Route 53 > 등록된 도메인 > 도메인 등록 > 체크아웃

1단계
요금

2단계
연락처 정보

3단계
검토 및 제출

요금 정보

도메인 요금 옵션

도메인 이름	기간(가격)	자동 갱신
myaws.world	1년 (12.00 USD)	<input type="checkbox"/> 비활성

i 이 도메인에 대해 자동 갱신이 꺼져 있으며 선택한 등록 기간 후에 등록이 만료됩니다.
자세한 내용은 [도메인 등록 갱신](#) 섹션을 참조하세요.

소계: 12.00 USD
적용되는 세금은 결제 시 계산됩니다.

취소

다음

7. [2단계:연락처 정보] 페이지이다. 각각의 값을 입력한다.

Route 53 > 등록된 도메인 > 도메인 등록 > 체크아웃

1단계
요금

2단계
연락처 정보

3단계
검토 및 제출

연락처 정보

정보

ⓘ 연락처 양식은 가장 최근에 등록한 도메인(henrytrainer.click)에 대해 입력한 정보로 미리 채워져 있습니다. 다른 연락처 정보를 사용하려면 정보를 편집하거나 재설정할 수 있습니다.

재설정

등록 연락처

일반 정보

연락처 유형

개인

이름 성

종순

이메일

javaexpert@nate.com

전화번호

국가 코드 및 전화번호를 입력하세요.

8. 각 항목의 값을 입력 후 페이지 하단의 [계속]을 클릭한다.

관리 연락처

등록 연락처와 동일

기술 연락처

등록 연락처와 동일

개인 정보 보호

myaws.world의 모든 연락처에 대해 개인 정보 보호 켜기
개인 정보 보호를 켜면 연락처 정보가 WHOIS 쿼리에서 숨겨집니다.

취소 이전 다음

9. 다음 단계는 [3단계:검토 및 제출] 페이지이다. [이용 약관] 항목을 체크하고 [제출]를 클릭한다.

이용 약관

Amazon Route 53에서는 AWS 계정을 사용하여 도메인 이름을 등록하고 이전할 수 있습니다. Route 53은 Amazon Registrar 및 다른 등록 협력사를 이용하여 등록 및 이전 서비스를 수행합니다. 도메인 등록 대행자는 귀하가 지정한 등록 연락처에 정기적으로 연락하여 연락처 세부 정보를 확인하고 등록을 갱신합니다. [Amazon Route 53 도메인 이름 등록 최종 사용자 계약](#)에서 자세한 내용을 참조하세요.

Amazon Route 53 도메인 이름 등록 최종 사용자 계약을 읽었으며 이에 동의합니다.

취소 이전 제출

10. [다음 도메인에 대한 등록 요청이 진행 중입니다. 각 도메인의 등록 상태에 관한 이메일을 받게 됩니다.] 메시지를 확인할 수 있다.

The screenshot shows the AWS Route 53 'Registered Domains' page. At the top, a blue banner displays the message: '① 다음 도메인에 대한 등록 요청이 진행 중입니다. 각 도메인의 등록 상태에 관한 이메일을 받게 됩니다.' with a bullet point '• myaws.world'. To the right of the banner are buttons for '상태 확인' (Status Check) and a close 'X' button. Below the banner, the main content area has a header '등록된 도메인 정보' and a search bar '검색' (Search). The table lists three domains: 'devtrainer.click' registered on July 28, 2024, at 00:08 UTC; 'henrytrainer.click' registered on July 28, 2024, at 15:54 UTC; and 'myaws.world' which is currently listed as '진행 중' (In Progress). There are buttons for '결제 보고서 다운로드' (Download Payment Report), '내 송신' (My Translations), and '도메인 등록' (Domain Registration).

11. [상태 확인]버튼을 클릭하면 다음과 같이 진행 중으로 확인된다.

The screenshot shows the AWS Route 53 'Requests' page. The table lists a single request for domain 'myaws.world' with the status '진행 중' (In Progress). Other columns include '작업 ID' (Job ID), '도메인 이름' (Domain Name), '메시지' (Message), '유형' (Type), and '제출함' (Submitted). The row for 'myaws.world' is highlighted with a red box.

12. 등록한 이메일을 확인해 보면 다음과 같이 등록이 성공했음을 확인할 수 있다.

The screenshot shows an AWS email confirmation for the registration of 'myaws.world'. The subject line is 'Registration of myaws.world succeeded'. The body of the email reads:

Dear AWS customer,
We successfully registered the myaws.world domain. We also created a hosted zone for the domain.
Your next step is to add records to the hosted zone. These records contain information about how to route traffic for your domain and any subdomains. To learn how to add records to your hosted zone, see [Working with Records](#).
If you did not request this change, contact [Amazon Web Services Customer Support](#) immediately.
Regards,
Amazon Route 53

13. 도메인 등록이 완료되었다.

The screenshot shows the AWS Route 53 'Registered Domains' page again. The table now includes the newly registered domain 'myaws.world' with the status '비활성' (Deactivated). The other domains listed are 'devtrainer.click' and 'henrytrainer.click', both of which are '비활성' (Deactivated). The row for 'myaws.world' is highlighted with a red box.

14. 등록 완료된 도메인의 [도메인 이름]을 클릭해서 상세 페이지로 들어간다.

Route 53 > 등록된 도메인 > myaws.world

myaws.world 정보

송신 ▾ 도메인 삭제

세부 정보 정보 작업 ▾

등록 날짜 10월 18, 2023, 23:24(UTC:+09:00)	자동 갱신 비활성	도메인 상태 코드 addPeriod ok	이름 서버 ns-580.awsdns-08.net ns-1432.awsdns-51.org ns-181.awsdns-22.com ns-1638.awsdns-12.co.uk
만료 날짜 10월 18, 2024, 23:24(UTC:+09:00)	이전 잠금 비활성	DNSSEC 상태 구성되지 않음	

Freenom에 무료 도메인 생성하기

1. 무료 도메인을 발급 받기 위해 freenom 사이트(<https://www.freenom.com/en/index.html?lang=en>)를 방문한다. 검색창에 도메인을 입력하고, 검색하기 위해 [Check Availability] 파란색 원을 클릭한다. 검색 결과에서 입력한 도메인을 사용하기 위해 해당 도메인의 [Get it now!]를 선택한 후, [Selected] 초록색 버튼을 확인하고, 상단의 [Checkout] 초록색 버튼을 클릭한다.

Get one of these domains. They are free!			
devtrainer .tk	• FREE	USD 0.00	<input checked="" type="button"/> Selected
devtrainer .ml	• FREE	USD 0.00	<input type="button"/> Get it now!
devtrainer .ga	• FREE	USD 0.00	<input type="button"/> Get it now!

2. 기본이 3개월 무료이고, 최대 12개월까지 무료로 사용할 수 있다. 12개월을 선택하고 [Continue] 파란색 버튼을 클릭한다.

3. 금액은 무료이기 때문에 0\$이다. 만일 로그인을 하지 않았다면 로그인한다. 구글, Facebook 가능하다. [Complete Order]를 클릭하여 주문을 완성한다.

Review & Checkout

Description	Price
Domain Registration - devtrainer.tk	\$0.00USD
Subtotal:	\$0.00USD
Total Due Today:	\$0.00USD

I have read and agree to the Terms & Conditions Complete Order

4. 주문이 마무리되었다. 등록한 메일로 인증 메일이 올 수 있다. 페이지 아래의 [Click here to go to your Client Area] 파란색 버튼을 클릭한다.

Order Confirmation

Thank you for your order. You will receive a confirmation email shortly.

Your Order Number is: 4823458296

If you have any questions about your order, please open a support ticket from your client area and quote your order number.

Click here to go to your Client Area

5. 방금 등록 완료된 도메인 정보를 확인할 수 있다. 혹은 로그인 후, 상단의 메뉴 [Services] > [My Domains]를 선택해도 된다.

My Domains

View & manage all the domains you have registered with us from here...

Filter

Domain	Registration Date	Expiry date	Status	Type
devtrainer.tk	2022-05-28	2023-05-28	ACTIVE	Free

Results Per Page: 1 Records Found, Page 1 of 1

Route 53을 이용한 도메인 네임 서버 생성

1. 이번 실습은 **Route 53**에 도메인을 가지고 있는 상태에서 **Route 53**을 네임 서버로 이용해서 어떤 특정 도메인과 연결하는 방법을 살펴보자. 좌측 메뉴 중 **[호스팅 영역]**을 클릭한다. 이미 위 실습에서 등록했던 도메인이 보인다. 선택하여 **[삭제]** 버튼을 클릭하여 **Freenom**에서 등록한 도메인으로 실습을 계속 진행하기로 한다.

The screenshot shows the AWS Route 53 console. In the top navigation bar, 'Route 53 > 호스팅 영역' is selected. Below it, a table lists a single hosting zone: 'myaws.world' (Type: Public, Region: Route 53, Record Count: 2). At the top of the table, there are several buttons: 'C' (Create), '세부 정보 보기' (View details), '편집' (Edit), '삭제' (Delete) which is highlighted with a red box, and '호스팅 영역 생성' (Create hosting zone). A search bar below the table contains the placeholder '속성 또는 값을 기준으로 호스팅 영역 필터링'. The bottom of the table has dropdown menus for '도메인 이름', '유형', '생성자', '레코드 수', '설명', and '호스팅 영역 ID'.

2. **[호스팅 영역...을 삭제하시겠습니까?]** 창에서 **삭제**라고 입력하고 **[삭제]** 버튼을 클릭한다.

The screenshot shows a confirmation dialog box titled '호스팅 영역 myaws.world을(를) 삭제하시겠습니까?'. It contains the message '호스팅 영역을 영구적으로 삭제하시겠습니까? 이 작업은 실행 취소할 수 없습니다. 인터넷에서 도메인을 사용하지 못할 수도 있습니다.' Below the message is a text input field containing '삭제' with a red box around it. At the bottom right of the dialog is a large orange '삭제' button also highlighted with a red box. To the left of the '삭제' button is a '취소' (Cancel) button.

3. **[호스팅 영역 생성]**을 클릭한다.

The screenshot shows the AWS Route 53 console again. The top navigation bar shows 'Route 53 > 호스팅 영역'. The table below is empty, displaying '호스팅 영역 없음' and the message '이 계정에 대해 생성된 호스팅 영역이 없습니다.' At the top of the table, the '호스팅 영역 생성' button is highlighted with a red box. The rest of the interface is identical to the previous screenshot, including the 'Create', 'View details', 'Edit', and 'Delete' buttons.

4. [호스팅 영역 생성] 페이지에서 [도메인 이름]은 Freenom.com에서 생성한 도메인을 넣는다.

Route 53 > 호스팅 영역 > 호스팅 영역 생성

호스팅 영역 생성

호스팅 영역 구성

호스팅 영역은 example.com 같은 도메인과 관련 하위 도메인에 대한 트래픽을 라우팅하는 방식에 대한 정보를 포함하는 컨테이너입니다.

도메인 이름 | 정보
트래픽을 라우팅할 도메인의 이름입니다.
devtrainer.tv

규칙: a-z, 0-9 및 ! " # \$ % & ' () * + , - : ; < = > ? @ [\] ^ _ ` { } . ~

설명 - 선택 사항 | 정보
이 값을 사용하면 이름이 동일한 호스팅 영역을 구별할 수 있습니다.
호스팅 영역이 사용되는 경우...

설명은 최대 256자입니다. 0/256

유형 | 정보
유형은 인터넷 또는 Amazon VPC에서 트래픽을 라우팅할지 여부를 가리킵니다.

퍼블릭 호스팅 영역
퍼블릭 호스팅 영역은 인터넷에서 트래픽을 라우팅하는 방식을 결정합니다.

프라이빗 호스팅 영역
프라이빗 호스팅 영역은 Amazon VPC 내에서 트래픽을 라우팅하는 방식을 결정합니다.

5. 페이지를 아래로 스크롤다운하여 [태그] 섹션에 [키]는 Name으로, [값]은 lab-hosting이라고 입력하고 [호스팅 영역 생성]을 클릭한다.

태그 정보

호스팅 영역에 태그를 적용하면 호스팅 영역을 쉽게 구성하고 식별할 수 있습니다.

키	값 - 선택 사항
Name	lab-hosting

태그 추가

최대 49개의 태그를 더 추가할 수 있습니다.

취소 **호스팅 영역 생성**

6. 다음과 같이 네임서버가 생성되었다.

The screenshot shows the AWS Route 53 console. In the top navigation bar, 'Route 53' > '호스팅 영역' > 'devtrainer.tv'. Below the navigation, there are tabs: '파블릭' (Public) selected, 'devtrainer.tv' (DNS), and '정보' (Information). On the right, there are buttons for '영역 삭제' (Delete zone), '레코드 테스트' (Test record), '쿼리 로깅 구성' (Configure query logging), and '호스팅 영역 편집' (Edit hosted zone). A sub-menu '▶ 호스팅 영역 세부 정보' (Hosted zone details) is open. Under the 'DNSSEC 서명' (DNSSEC signature) tab, it says '호스팅 영역 태그(1)'. The main content area shows a table of DNS records:

레코드 (2) 정보	Automatic 모드는 최상의 필터 결과에 최적화된 현재 검색 동작입니다. 모드를 변경하려면 설정(settings)으로 이동합니다.				
속성 또는 값을 기준으로 레코드 필터링	유형	라우팅 정책	별칭	선택	
	레코드 이름	유형	라우팅 정책	차별화된 대상	값/트래픽 라우팅 대상
<input type="checkbox"/>	devtrainer.tv	NS	단순	-	ns-1438.awsdns-51.org. ns-1576.awsdns-05.co.uk. ns-633.awsdns-15.net. ns-60.awsdns-07.com.
<input type="checkbox"/>	devtrainer.tv	SOA	단순	-	ns-1438.awsdns-51.org. awsdns-hostmaster.amazon.com. 1 7200 900 1209600 86400

7. 이번에는 네임 서버를 등록해 보자. 다시 **Freenom.com**의 **My Domain** 페이지로 이동한다. [Manage Domain] 옆의 톱니바퀴 아이콘을 클릭한다.

The screenshot shows the Freenom 'My Domains' page. The title is 'My Domains' and the subtitle is 'View & manage all the domains you have registered with us from here...'. There is a search bar 'Enter Domain to Find' and a 'Filter' button. Below is a table of registered domains:

Domain	Registration Date	Expiry date	Status	Type	Action
devtrainer.tk	2022-05-28	2023-05-28	ACTIVE	Free	Manage Domain

At the bottom left, 'Results Per Page:' is set to 10. At the bottom center, it says '1 Records Found, Page 1 of 1'.

8. 또는 등록한 도메인의 **[Management Tools]** > **[Nameservers]** 메뉴를 선택한다.

The screenshot shows the 'Managing devtrainer.tk' page. On the left, there is an 'Information' section with a link to 'Back to Domains List'. In the center, there is a 'Management Tools' dropdown menu. The 'Nameservers' option is highlighted with a red box. To the right, there is a 'Manage Freenom DNS' section showing domain details:

Domain:	devtrainer.tk	Status:	ACTIVE
Registration Date:	2022-05-28	Expiration Date:	2023-05-28
Dot TK Apps:		Dot TK Ambassadors:	
Cancel domain			

9. 다음 그림과 같은 페이지로 이동한다. 현재 [Use default...]로 선택되어 있는데, 방금 생성한 도메인의 관리를 **Freenom.com**이 아닌 **AWS Route 53** 서비스가 대행할 예정이기 때문에 [**Use custom nameservers (enter below)**]를 선택한다.

Managing devtrainer.tk

Information Upgrade Management Tools ▾ Manage Freenom DNS

Nameservers

You can change where your domain points to here. Please be aware changes can take up to 24 hours to propagate.

Use default nameservers (Freenom Nameservers)

Use custom nameservers (enter below)

Change Nameservers

10. AWS Route 53 페이지에서 등록한 도메인의 NS 유형의 값 4개를 복사해서 **Freedon.com**에 붙여넣는다.

Route 53 > 호스팅 영역 > devtrainer.tv

파블릭 devtrainer.tv 정보

▶ 호스팅 영역 세부 정보

호스팅 영역 편집

레코드(2) DNSSEC 서명 호스팅 영역 태그(1)

레코드 (2) 정보

Automatic 모드는 최상의 필터 결과에 최적화된 현재 검색 등작입니다. 모드를 변경하려면 설정(settings)으로 이동합니다.

레코드 이름	유형	라우팅...	차별...	값/트래픽 라우팅 대상
devtrainer.tv	NS	단순	-	ns-1438.awsdns-51.org. ns-1576.awsdns-05.co.uk. ns-633.awsdns-15.net. ns-60.awsdns-07.com.
devtrainer.tv	SOA	단순	-	ns-1438.awsdns-51.org. awsdns-hostmaster.amazon.com. 1 7200 900 1209600 86400

11. 여기서 주의할 점은 반드시 [유형]이 NS의 값만 복사해서 넣어야 한다. 또한 마지막 .은 붙여넣지 않는다. 입력을 마치면 [Change Nameservers] 파란색 버튼을 클릭한다.

Nameservers

You can change where your domain points to here.
Please be aware changes can take up to 24 hours to propagate.

Use default nameservers (Freenom Nameservers)

Use custom nameservers (enter below)

Nameserver 1
ns-1438.awsdns-51.org

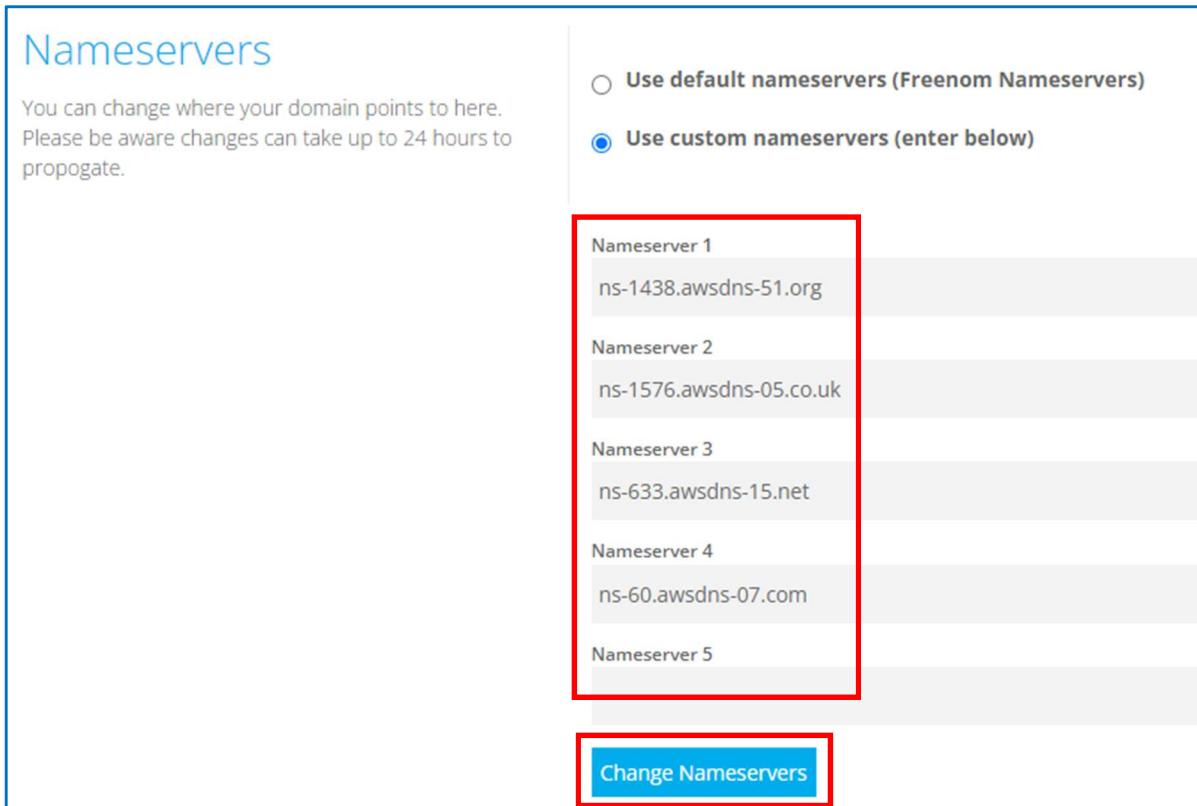
Nameserver 2
ns-1576.awsdns-05.co.uk

Nameserver 3
ns-633.awsdns-15.net

Nameserver 4
ns-60.awsdns-07.com

Nameserver 5

Change Nameservers

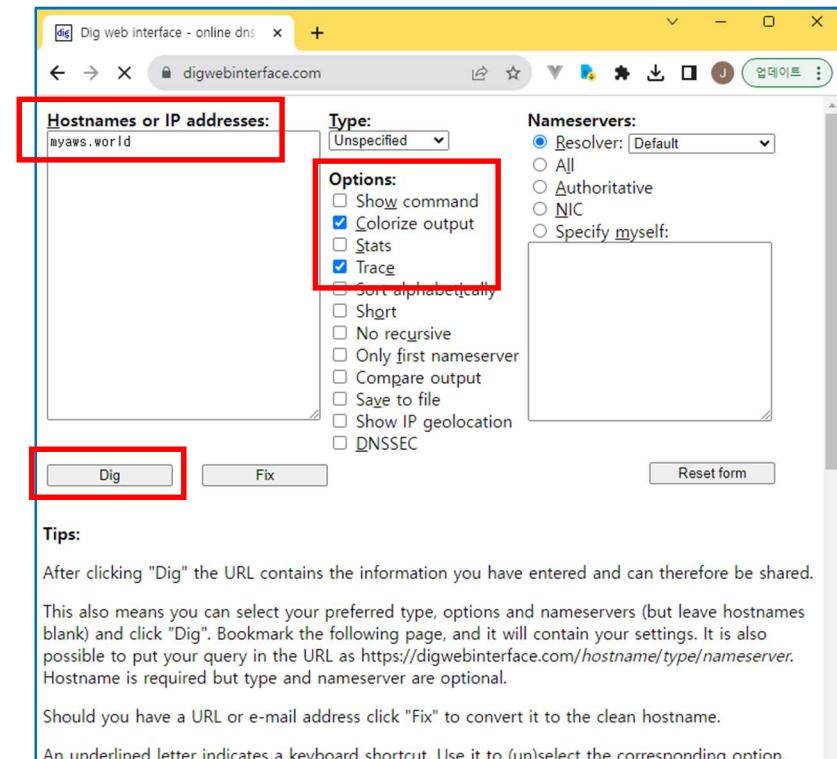


12. 새로 등록한 네임서버 정보가 전체 DNS 서버들을 통해 업데이트하는데는 24시간 정도의 시간이 필요하다.

DNS 디버그

1. **Dig**이라는 도구를 통해 DNS 설정을 확인하는 실습을 한다. **Dig**은 Windows를 제외한 다른 OS에서 사용할 수 있는 명령어이다. Windows에서는 다음과 같은 과정을 통해 사용할 수 있다. 웹 브라우저를 열고 다음의 URL을 입력하여 이동한다. **[Hostnames or IP addresses:]** 항목에 등록 테스트할 도메인을 입력하고, **[Options]**에서는 **Colorize output, Trace**를 체크하고 **[Dig]** 버튼을 클릭한다.

<http://www.digwebinterface.com>



2. 다음과 같은 결과를 얻을 수 있다. 만일 OS가 macOS라면 Terminal에서 다음과 같은 명령어를 사용한다.

```
$ dig +trace {domain name}
```

도메인 이름과 IP 주소 연결

1. 이제 등록한 도메인과 IP 주소를 연결하는 작업을 한다. 다시 AWS Route 53 페이지의 등록한 도메인의 상세 정보페이지로 이동한다. 오른쪽에 있는 [레코드 생성]을 클릭한다.

Route 53 > 호스팅 영역 > myaws.world

파블릭 myaws.world 정보

▶ 호스팅 영역 세부 정보

호스팅 영역 편집

레코드(2) 정보 DNSSEC 서명 호스팅 영역 태그(0)

레코드 (2) 정보

Automatic 모드는 최상의 필터 결과에 최적화된 현재 검색 동작입니다. 모드를 변경하려면 설정(settings)으로 이동합니다.

속성 또는 값을 기준으로 레코드 필터링 유형 라우팅 정책 별칭 값/트래픽 라우팅 대상 TTL(초) 상태 확장 대상 레코드 ID

myaws.world NS 단순 아니요 ns-580.awsdns-08.net. ns-1432.awsdns-51.org. ns-181.awsdns-22.com. ns-1638.awsdns-12.co.uk. 172800

myaws.world SOA 단순 아니요 ns-580.awsdns-08.net. awsd... 900

레코드 생성

2. [1단계:라우팅 정책 선택] 페이지에서 [빠른 생성으로 전환] 링크를 클릭한다.

Route 53 > 호스팅 영역 > myaws.world > 레코드 생성

1단계 라우팅 정책 선택

2단계 레코드 구성

라우팅 정책 선택 정보

라우팅 정책

빠른 생성으로 전환

단순 라우팅 모든 클라이언트가 동일한 응답을 수신하도록 하려면 사용합니다.

가중치 기반 동일한 작업을 수행하는 리소스가 여러 개 있고 각 리소스로 할당하는 트래픽의 비율을 지정하려는 경우 사용합니다. 예: EC2 인스턴스 2개 이상.

지리적 위치 사용자의 위치를 기반으로 트래픽을 라우팅하려는 경우 사용합니다.

지연 시간 여러 AWS 리전에 리소스가 있고 가장 짧은 지연 시간을 제공하는 리전으로 트래픽을 라우팅하려는 경우 사용합니다.

장애 조치 리소스가 정상일 때 해당 리소스로 트래픽을 라우팅하거나 첫 번째 리소스가 비정상일 때 다른 리소스로 트래픽을 라우팅하려는 경우 사용합니다.

다중값 응답 Route 53이 DNS 쿼리에 무작위로 선택된 최대 8개의 정상 레코드로 응답하도록 하려는 경우 사용합니다.

IP 기반 CIDR 표기법의 IP 주소 범위 위치로 트래픽을 라우팅하는 데 사용합니다.

취소 다음

3. [레코드 생성] 페이지에서 다음과 같이 각각의 값을 설정한다. 먼저, [레코드 이름]에는 만일 www 서비스를 하려면 www라고 입력한다. 그러면 유저들은 www.myaws.world로 접속할 것이다. 그 다음 [레코드 유형]에서는 가장 중요한 A 레코드를 선택한다.

The screenshot shows the 'Record Creation' section of the Route 53 console. The 'Record Type' dropdown is set to 'A - IPv4 주소 및 일부 AWS 리소스로 트래픽 라우팅'. The 'Name' field contains 'www' followed by '.myaws.world'. The 'TTL(초)' field is set to 300. The 'Value' field shows the IP address 192.0.2.235. The 'Create' button is highlighted in orange at the bottom right.

4. 다음은 [값]을 넣어야 하는데, 현재 Web Server EC2 인스턴스의 요약 페이지에 있는 [퍼블릭 IPv4 주소]를 복사해서 넣으면 된다. 그리고 [TTL] 값은 기본값 300초로 설정한다. 그리고 [레코드 생성]을 클릭한다.

The screenshot shows the EC2 instance summary for i-0498c3a243c260b68. The 'Public IPv4 Address' field is highlighted with a red box and contains '43.202.44.21'. This value is copied and pasted into the 'Value' field on the Route 53 record creation page. The 'TTL(초)' field is set to 300. The 'Create' button is highlighted in orange at the bottom right.

5. 레코드 생성이 마무리되었다. 하지만 아직 접속이 안 될 수 있다. 왜냐하면 캐시 때문에 저장된 정보들이 갱신되는 데 시간이 오래 걸릴 수 있다. 하루 이상 걸릴 수도 있다. 그래서 **Dig**을 통해 DNS 디버깅하는 방법이 좋다.

레코드 이름	유형	라우팅 정책	별칭	값/트래픽 라우팅 대상	TTL(초)	상태 확인	대상	레코드 생성
myaws.world	NS	단순	아니요	ns-580.awsdns-08.net. ns-1432.awsdns-51.org. ns-181.awsdns-22.com. ns-1638.awsdns-12.co.uk.	172800	-	-	-
www.myaws.world	A	단순	아니요	ns-580.awsdns-08.net.awsdns-51.org.ns-1432.awsdns-51.org.ns-181.awsdns-22.com.ns-1638.awsdns-12.co.uk.	900	-	-	-
www.myaws.world	A	단순	아니요	43.202.44.21	300	-	-	-

6. myaws.world에 대한 레코드가 생성되었습니다. 메시지박스안에 있는 [상태 보기] 버튼을 클릭해보자.

myaws.world에 대한 레코드가 생성되었습니다.
Route 53은 60초 이내에 모든 Route 53 신뢰할 수 있는 DNS 서버에 변경 사항을 전파합니다. '상태 보기(View status)' 버튼을 사용하여 전파 상태를 확인합니다.

7. 현재 [상태]는 INSYNC임을 확인할 수 있다.

ID	제출 날짜:
/change/C00722093RPCZSEDEK57O	10월 19, 2023, 09:49(UTC+09:00)
상태	설명
INSYNC	-

Microsoft Windows [Version 10.0.22621.2428]
(c) Microsoft Corporation. All rights reserved.

```
C:\Users\MZC01-HENRY>nslookup
기본 서버: Unknown
Address: 192.168.0.1

> www.myaws.world
서버: UnKnown
Address: 192.168.0.1

권한 없는 응답:
이름: www.myaws.world
Address: 43.202.44.21
```