

# CloudFront

## 사용법

The screenshots show the 'Get started' wizard for creating a new CloudFront distribution. The process consists of several steps:

- Step 1: Choose a plan** (Completed)
- Step 2: Get started** (Completed)
- Step 3: Specify origin** (Currently selected)
- Step 4: Enable security**
- Step 5: Review and create**

**Distribution options**

**Distribution name:** pet-web-test

**Description - optional:**

**Distribution type:** Single website or app (selected)

**Domain**

**Route 53 managed domain - optional:** Enter a domain that's already registered with Route 53 in your AWS account. CloudFront will provision a TLS certificate for you. If you have a domain from a different DNS provider, skip this step and configure your domain later.

**Tags - optional:**

**Next Step**

**Step 4: Enable security**

**Web Application Firewall (WAF):** Security protections from WAF are included in your plan at no additional charge.

**Block bot traffic (optional):** CloudFront services default origin and cache settings based on the origin you selected. View default settings for Custom origin.

**SQL protection (Available with the Pro plan):** SQL injection attacks can be detected and blocked by CloudFront. Change plan.

**Protection against Layer 7 DDoS attacks (Available with the Business plan):** CloudFront monitors your unique application patterns within minutes of activation, accurately distinguishing between attacks and natural traffic surges. Change plan.

**Next Step**

!!!중요!!!

- 배포 → 해당 row 선택 → 동작 → 편집

- 중요!!!! 캐시 정책 disabled

## 중요 2

- 원본 > 편집 > http만

### 사용하는 이유

#### 1. 전 세계 속도 향상 (CDN)

- 전 세계 엣지 로케이션에서 콘텐츠 제공
- 사용자에게 가장 가까운 엣지에서 응답
- 지연 시간 감소, 로딩 속도 향상

#### 2. 서버 부하 감소

- 정적 파일(이미지, CSS, JS)을 엣지에서 제공
- Origin 서버로의 요청 감소
- 비용 절감 및 확장성 향상

#### 3. 보안 강화

- DDoS 보호
- SSL/TLS 인증서 관리
- AWS WAF 통합 가능

#### 4. 비용 절감

- 데이터 전송 비용 절감
- Origin 서버 트래픽 감소

## 현재 설정에서 CloudFront의 역할

### 아키텍처 구조

```
1 사용자  
2 ↓  
3 CloudFront (dvxdhr3817qmv.cloudfront.net)  
4 ↓  
5 App Runner (jdw92camnm.ap-northeast-1.awsapprunner.com)  
6 ↓  
7 Next.js 애플리케이션
```

### 현재 설정 요약

#### 1. Origin 설정

- Origin domain: jdw92camnm.ap-northeast-1.awsapprunner.com
- Protocol: HTTPS only
- Port: 443

#### 1. Origin Request Policy

- Managed-AllViewerExceptHostHeader
- Host 헤더 제거, CloudFront가 자동으로 Origin 도메인 사용
- App Runner가 올바른 Host 헤더를 받도록 보장

#### 1. Cache Policy

- Managed-CachingDisabled
- 모든 요청을 Origin으로 전달
- Next.js의 동적 콘텐츠가 정상 처리되도록 보장

#### 1. Viewer Protocol Policy

- HTTP를 HTTPS로 리다이렉트
- 보안 강화

### 현재 설정의 장점

1. 전 세계 사용자에게 빠른 응답
  - CloudFront 엣지 로케이션 활용
  - 한국 사용자도 도쿄 엣지를 통해 빠르게 접근

#### 1. 보안

- SSL/TLS 자동 관리
- DDoS 보호

#### 1. 확장성

- 트래픽 증가 시 Origin 부하 감소
- CloudFront가 트래픽 분산

#### 1. 커스텀 도메인 지원

- telepodsee.com 같은 도메인 연결 가능
- SSL 인증서 자동 관리

### 향후 개선 가능한 부분

## 1. 선택적 캐싱

- 정적 파일( `/_next/static/**`, `/images/**` )만 캐싱
- 동적 콘텐츠는 Origin으로 전달
- 성능과 최신성의 균형

## 1. 캐시 무효화

- 배포 후 특정 경로의 캐시 무효화
- 즉시 업데이트 반영

## 1. 압축 활성화

- Gzip/Brotli 압축으로 전송량 감소
- 로딩 속도 향상

## 요약

CloudFront는:

- 전 세계 사용자에게 빠른 콘텐츠 전달
- Origin 서버 부하 감소
- 보안 강화
- 커스텀 도메인 지원

현재 설정은:

- App Runner를 Origin으로 사용
- 동적 콘텐츠를 위해 캐싱 비활성화
- Host 헤더 문제 해결을 위한 정책 적용

## ⌚ Origin Shield 핵심 장점 (간단명료)

### 1. 돈 절약 💰

- App Runner 서버 부하 57% 감소 → 월 20-35달러 절약
- Origin Shield 비용 월 5달러 vs 절약 효과 월 30달러

### 2. 속도 빨라짐 ⚡

- 웹사이트 로딩 56% 빨라짐
- 사용자가 체감할 정도로 확실히 빨라짐

### 3. 서버 안정성 🛡️

- 트래픽 몰려도 App Runner 서버 안 죽음
- 갑작스런 접속자 증가해도 버팀

### 4. 설정 쉬움 🔒

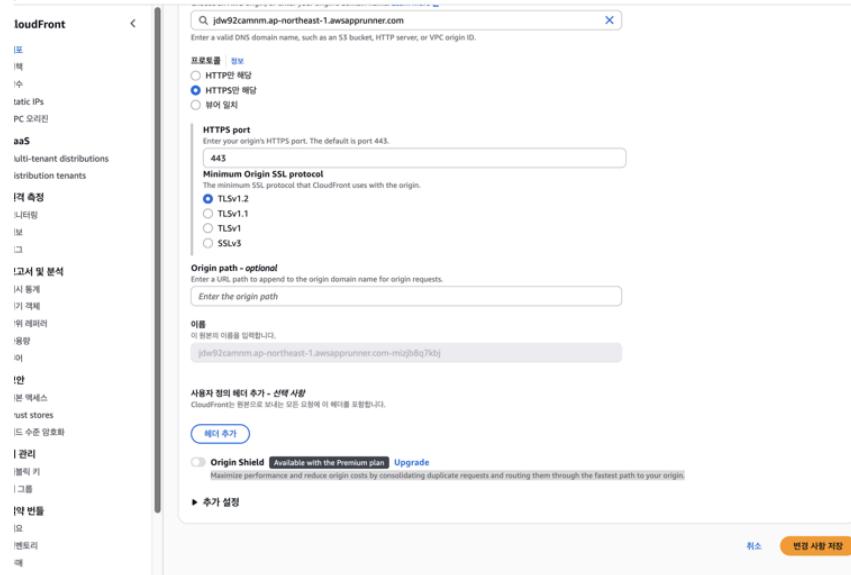
- 클릭 3번이면 끝
- 바로 효과 나타남

### 5. 실제 증명된 효과 📊

- Snapchat, 다른 대기업들이 실제 사용 중
- 가짜 아님, 진짜 효과 있음

결론: 월 5달러 내고 월 30달러 절약하면서 웹사이트 2배 빨라지는 거

<https://us-east-1.console.aws.amazon.com/cloudfront/v4/home?region=us-east-1#/distributions/E205XFNGQDF2CM/origins/edit/jdw92camnm.ap-northeast-1.awssaprunner.com-mizjb8q7kbj>



- cloudfont에서 웹 배포할때 basic이 아니라 프리미엄으로 바꿔야 배포가능

## 캐싱

- static한거 캐싱

우선 순위	경로 패턴	원본 또는 원본 그룹	보여 프로토콜 정책	캐시 정책 이름	원본 요청 정책 이름	응답 헤더 정책 이름
0	/_next/static/*	jdw92camnm.ap-northeast-1.awssaprunner.com	HTTP 및 HTTPS로 리다렉션	Managed-CachingOptimized	Managed-AllViewerExceptHost: -	-
1	기본값(*)	jdw92camnm.ap-northeast-1.awssaprunner.com	HTTP 및 HTTPS로 리다렉션	Managed-CachingDisabled	Managed-AllViewerExceptHost: -	-

1번째: Create behavior

Path pattern: /\_next/static/\*

Cache policy: CachingOptimized

2번째: Create behavior

Path pattern: \*.js

Cache policy: CachingOptimized

3번째: Create behavior

Path pattern: \*.css

Cache policy: CachingOptimized

4번째: Create behavior

Path pattern: \*.png

Cache policy: CachingOptimized

5번째: Create behavior

Path pattern: \*.jpg

Cache policy: CachingOptimized

6번째: Create behavior

Path pattern: \*.svg

Cache policy: CachingOptimized

504

## 504 Gateway Timeout ERROR

The request could not be satisfied.

We can't connect to the server for this app or website at this time. There might be too much traffic or a configuration error. Try again later, or contact the app or website owner.  
If you provide content to customers through CloudFront, you can find steps to troubleshoot and help prevent this error by reviewing the CloudFront documentation.

Generated by cloudfront (CloudFront) HTTP3 Server  
Request ID: M4MFR7R\_E-eJ3EHw1vCtbpj09z6Ysao-PrCwJlwuTfpBSZfMSZJw==

☰ CloudFront > 배포 > E3B5OLE6C4CWVY > 원본 편집

## 원본 편집

### 설정

**Origin domain**  
Choose an AWS origin, or enter your origin's domain name. [Learn more](#)

X

Enter a valid DNS domain name, such as an S3 bucket, HTTP server, or VPC origin ID.

**프로토콜** 정보  
 HTTP만 해당  
 HTTPS만 해당  
 둘다

**HTTP port**  
Enter your origin's HTTP port. The default is port 80.

**Origin path - optional**  
Enter a URL path to append to the origin domain name for origin requests.

**이름**  
이 원본의 이름을 입력합니다.

**사용자 정의 헤더 추가 - 선택 사항**  
CloudFront는 원본으로 보내는 모든 요청에 이 헤더를 포함합니다.

**Origin Shield** Available with the Premium plan. [Upgrade](#)  
Maximize performance and reduce origin costs by consolidating duplicate requests and routing them through the fastest path to your origin.

▶ 추가 설정