ucloud CDN(Contents Delivery Network) Customer Guidebook

[ucloud CDN고객가이드]

2012년 4월

olleh

목 차

1. ucloud CDN 서비스에 대한 이해

1.1. CDN (Contents Delivery Network) 이란

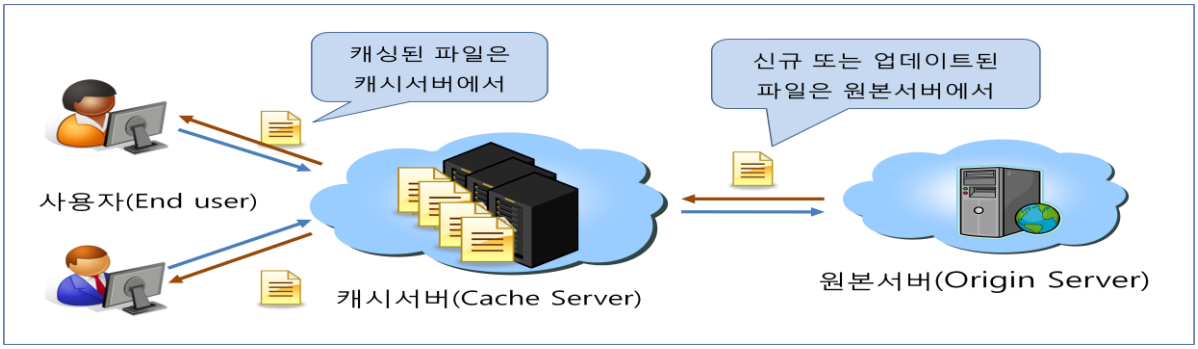
인터넷 사용자들에게 웹 컨텐츠를 빠르고 안정적으로 전달하기 위해서 등장한 서비스로서 다수의 CDN 서버를 구축하여 사용자가 가까운 위치에 있는 CDN 서버에서 컨텐츠를 다운로드 할 수 있도록 함으로써 컨텐츠의 전송 속도를 향상시키고 다수의 CDN 서버가 지속적으로 유지 되어 안정적인 서비스를 할 수 있는 서비스 입니다.

ucloud CDN 서비스는 CDN 서비스에 적합한 웹 캐시서버(Web Cache Server)를 구성하여 빠른 응답 속도와 반복적인 요청에 의한 부하 감소와 트래픽 절감 효과를 볼 수 있도록 하였습니다.

1.2. 웹 캐시 서비스 이해

웹 캐시 서비스는 이전 사용자(End User)의 요청에 의해 사용된 웹 컨텐츠를 저장하고 있다가 재요청이 있을 경우 원본서버(Origin Server)에 요청하지 않고 이전에 저장한 컨텐츠를 직접 전송하여 지연 시간을 최소화하여 신속한 웹 서비스를 제공하며, 반복되는 데이터 전송을 최소화 함으로써 원본서버의 부하를 감소시키고 네트워크 대역폭을 절약할 수 있는 서비스 입니다.

[그림1] 웹 캐시서버 동작 구조도

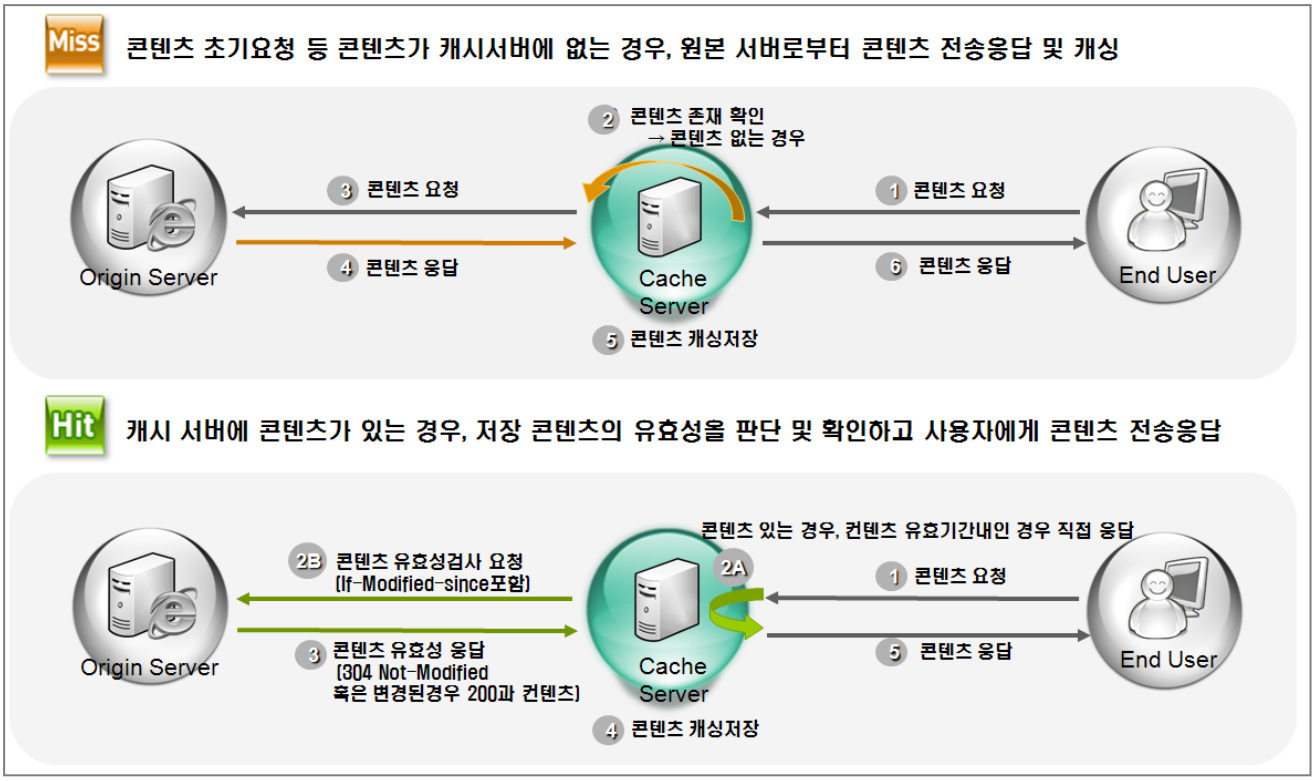


2. 캐시에 대한 오해와 이해

2.1. HIT과 MISS의 의미

캐시서버는 사용자가 요청하는 컨텐츠가 자신의 Memory나 HDD에 저장(cached)되어 있는지를 먼저 확인합니다. 이때 요청한 컨텐츠가 캐시서버에 저장되어 있다고 판단을 할 경우 원본서버로 요청을 하지 않고 캐시서버에 저장된 컨텐츠를 사용자에게 전송하며 'HIT' 이라고 응답을 합니다. 그리고 요청한 컨텐츠가 캐시서버에 저장되어 있지 않다고 판단을 하면 원본서버로 요청을 하게 되는데 이때 캐시서버는 'MISS'라고 응답을 합니다.

[그림 2] HIT과 MISS의 동작 구성도



2.2. Cache에 대한 이해

'MISS'의 경우는 이전에 요청된 이력이 없는 경우 처음에만 MISS가 발생하며 이후부터는 HIT이 발생합니다. 그러나 아래와 같이 캐시를 할 수 없는 경우도 있습니다.

캐시를 할 수 없는 경우

요청하는 METHOD가 POST 인 경우
원본서버에서 응답헤더에 Last-Modified가 없는 경우
캐시서버에 저장된 컨텐츠의 Last-Modified 시간이 원본서버의 컨텐츠 보다 최근일때
요청하는 HTTP 헤더에 no-cache, no-store, max-age=0 등을 포함 하고 있는 경우
원본서버에서 응답하는 상태코드가 200 OK가 아닌 경우

ucloud CDN 서비스의 웹 캐시서버는 GET, HEAD의 Method만 캐시를 POST Method는 캐시를 하지 않습니다.

웹 캐시서버는 원본서버에 갱신 여부를 확인하기 위해서 HTTP header의 Last-Modified 시간을 비교하므로 원본서버의 응답 헤더에 Last-Modified가 없으면 캐시를 하지 않으며 있더라도 원본서버에서 기존의 컨텐츠보다 더 오래된 컨텐츠로 변경한 경우에도 갱신을 하지 않고 기존의 저장된 컨텐츠를 전송합니다.

원본서버의 Cache-Control과 Pragma 설정이 되어 있는 경우 캐시서버는 자신의 설정보다 웹 서버의 설정을 우선으로 적용하게 됩니다. 그래서 no-cache와 같은 요청이 있을 경우 캐시서버에 저장되어 있는 컨텐츠를 사용하지 않고 'MISS'를 발생하게 됩니다.

웹 캐시서버는 원본서버로부터 정상적인 응답을 받지 않은 경우 저장을 하지 않으나 404 Not Found의 경우는 사용자들의 악의적인 요청이 빈번히 발생 할 수 있어 웹

캐시서버에서 30초간 원본서버로 질의 없이 사용자의 동일한 요청에 대해서 404 Not found를 응답합니다. 이를' NEGATIVE HIT'이라고 합니다.

캐시를 할 수 없는 경우의 요청이나 응답을 받은 경우에는 컨텐츠를 저장하지 않으며 캐시서버는 프록시(proxy) 역할만 합니다.

캐시서버는 요청받은 컨텐츠가 저장되어 있더라도 1일이 지난 경우에는 해당 컨텐츠가 변경이 되었는지 여부를 확인하기 위해서 원본서버에 IMS(If-Modified-Since) 요청을 하여 변경 여부를 확인을 합니다. 이때 원본서버가 변경되지 않았다고 304 Not Modified 응답을 하면(이때 응답헤더만 받게 됨) 캐시서버는 저장된 컨텐츠를 사용자에게 전송하게 됩니다. 그리고 원본서버에서 변경이 된 경우에는 해당 컨텐츠를 원본서버로부터 컨텐츠를 다운로드 하게 되고 다운로드가 완료되면 원본서버에서 200 OK를 응답하게 되며 캐시서버는 내려 받은 최신 컨텐츠를 사용자에게 전송하게 됩니다. 만약 컨텐츠가 원본서버에서 삭제 된 경우에는 원본서버가 404 Not Found 라고 응답을 하고 캐시서버는 기존에 저장하고 있던 컨텐츠를 삭제하고 사용자에게 404 Not Found를 응답하게 됩니다.

웹 캐시서버와 원본서버간 네트워크가 단절되는 경우 또는 원본서버의 시스템 다운과 같이 웹 캐시서버가 원본서버로 갱신여부를 확인 할 수 없는 경우에는 캐시서버는 저장하고 있는 컨텐츠를 지속적으로 서비스 하게 됩니다. 물론 저장되어 있지 않은 컨텐츠의 요청에 대해서는 원본서버로부터 받은 HTTP 에러 코드를 보여주게 됩니다.

3. 효율적인 캐시서비스 방법

3.1. 효율적인 캐시정책의 기본

효율적인 캐시서비스를 위해 컨텐츠의 종류와 특성별로 분리를 하여 적절하게 캐시서비스를 이용하시는 것이 좋습니다. 아래에 소개되는 효율적인 캐시정책을 활용하신다면 CDN 서비스의 빠른 응답시간과 높은 Hit율을 보장할 수 있습니다.

'HIT', 'MISS' 컨텐츠를 분류해 도메인으로 분리합니다.

캐시서버에서 'HIT' 컨텐츠와 'MISS' 컨텐츠를 함께 사용하도록 한 도메인을 사용하게 되면 'HIT' 되는 컨텐츠는 캐시서버에서 바로 응답을 하여 빠른 전송을 할 수 있지만 그렇지 않은 'MISS' 되는 컨텐츠는 캐시서버를 거쳐 원본서버에서 응답을 하게 되므로 그만큼 응답 시간이 지연됩니다. 그래서 이런 경우는 도메인을 각각 분리하여 'HIT' 되는 컨텐츠의 도메인은 CDN 서비스를 이용하도록 설정을 하고 그렇지 않은 도메인은 웹 서버에서 직접 전송토록 하는 것이 효율적입니다.

3.2. 웹 서버에서 컨텐츠 만료기간 관리

ucloud CDN 서비스를 담당하는 웹 캐시서버는 원본 컨텐츠가 갱신되었는지 여부를 1일마다 확인하도록 되어 있습니다. 이 갱신주기 보다 짧거나 길게 설정을 하셔야 할 때는 웹 서버에서 컨텐츠 타입별 또는 디렉토리별로 Cache-Control 또는 Expires와 같은 설정을 하시면 관리가 용이합니다. CDN 서비스에 제공되는 웹 캐시서버는 원본서버에서 응답하는 컨텐츠의 만료기간을 더 우선으로 하고 있으므로 원본 서버의 설정을 따르게 됩니다.

갱신주기를 짧게 설정하는 경우 캐시서버가 원본서버로 갱신여부를 자주 질의하게 되므로 CDN 서비스를 이용하는 효과가 낮아 질 수 있으므로 웹 서버 설정시 충분한 고려가 필요합니다.

3.3. 웹 서버 설정

웹 캐시서버의 동작원리는 원본서버로부터 새로운 컨텐츠를 받아 저장(cache)하고, 그 컨텐츠에 대한 갱신주기를 판단해 요청 받은 컨텐츠를 적절하게 전송하는 것입니다. 적절한 갱신주기 설정은 효율적인 캐싱을 위해 중요한 요소입니다. 컨텐츠에 대한 갱신 정책은 웹 서버에서 설정할 수 있으며, 캐시서버에서도 해당 컨텐츠에 대한 유효성 여부를 검사할 수 있는 항목을 제공하고 있지만 웹 서버의 설정을 우선으로 합니다.

[표 1]에서 보는 내용들은 일반적으로 웹 페이지 구성시 캐시와 관련된 것이며, 이외에도 다양한 항목들이 컨텐츠 유효성 검사를 위한 캐시정책에 영향을 미치게 됩니다.

[표 1] 캐싱 컨텐츠 유효성 관련 항목

항목	내용	비고
마지막 변경 사항 (Last-Modified)	해당 컨텐츠의 마지막 변경 시간을 기록	GMT 기준
E-Tag	서버에서 생성되는 유일한 인자로 컨텐츠의 중복성과 캐싱 유효성 여부를 확인할 수 있는 항목	HTTP 1.1
캐시컨트롤 (Cache-Control)	해당 컨텐츠에 대한 캐싱 관련 정책을 제어하는 항목 (max-age, no-cache, must-revalidate 등이 있음)	HTTP 1.1
Age	해당 컨텐츠가 갱신된 시점을 기준으로 초기화되며 새롭게 갱신되는 시점까지 1초 단위로 증가함	
Expires	해당 컨텐츠가 얼마동안 유효한지를 표시해주는 항목으로 명시된 시간 이후 캐싱 유효성 재확인	GMT 기준

웹 서버의 특성에 따라 설정을 수정하실 때는 캐시정책과 관련 있는지 유무를 확인하셔야 합니다. 웹 서버의 설정은 캐시서버의 캐시정책에도 영향을 미치게 되므로 설정 시에 유념하시고 변경 시 의문점이 있으면 캐시서비스 담당자에게 문의를 하시기 바랍니다.

3.4. 캐시 컨트롤 HTTP 헤더

HTTP 1.1 에서는 Cache-conrol Response Header라는 새로운 헤더 클래스를 지원합니다. 이는 만료시간시간 설정에 대한 문제 해결책으로 웹 제작자가 직접 컨텐츠를 조절할 수 있게 해줍니다. 캐시 컨트롤 헤더에 포함되는 옵션은 아래와 같습니다.

- max-age=[초] : 객체가 업데이트가 필요 없다고 여길 수 있는 최대 시간을 정해줍니다. 만료시간과 마찬가지로 특정시간이라기 보다는 요청시간과 관련이 있습니다. 요청받은 시간으로부터 설정해준 [초]까지 컨텐츠가 업데이트 필요없는 객체라고 나타내줍니다.
- s-max-age=[초] : max-age와 비슷합니다
- public : 캐시할 수 있는 인증된 요청임을 표시하는 것으로 일반적으로 HTTP 인증이 요구되는 경우는 자동적으로 응답은 캐시하지 않습니다.
- no-cache : 매회 캐시된 객체를 사용자에게 전달하기 전에 캐시에서 원본서버로 객체를 요청합니다. 인증이 중요시 되는 경우 유용하며, 계속적으로 업데이트되어야 할 객체일 경우 유용합니다.
- no-store : 어떠한 경우라도 객체가 캐시되지 않도록 합니다.

- must-revalidate : 웹 캐시서버가 컨텐츠의 업데이트 여부에 대한 정보를 반드시 따르도록 정해줍니다. HTTP가 업데이트 되지 않은 컨텐츠에 대해서 특정 조건하에서 사용자에게 제공하도록 합니다.

3.5. Freshness (신선도)

'Freshness(신선도)'란 클라이언트의 요청이 'HIT'되었을 때 캐시서버에서 해당 컨텐츠를 직접 전달해도 되는지 판단하는 것을 말합니다. 이때 'Fresh(신선한)'와 'Stale(신선하지 않은)'이라는 용어도 함께 사용되는데 'Freshness'는 캐시나 웹에서 아주 중요합니다. 캐시서버에서 신선도를 판단하지 않는다면 원본서버에서 컨텐츠가 변경되었음에도 불구하고 캐시서버는 이전에 캐시된 컨텐츠를 계속해서 응답하게 되므로 사용자는 새로운 컨텐츠를 받을 수 없게 됩니다.

캐시서버가 확인하는 것은 응답했던 컨텐츠의 유효기간(Expiration time)을 제일 먼저 확인하는데 이 정보는 이전에 HTTP응답헤더(HTTP response header)의 max-age나 Expires로 판단합니다.

예를 들어 이전에 'Cache-control : max-age=86400' (24시간) 라는 값을 가지고 있었는데 아직 응답한지 24시간이 지나지 않았으면 이 컨텐츠는 신선(Fresh)하다고 판단하여 원본서버에 요청하지 않고 캐시서버에서 즉시 응답해 주고 만약에 이 컨텐츠가 유효기간이 지났다면 원본서버로 유효한지를 확인하는 과정을 거치게 됩니다.

3.6. Validation (유효성)

캐시서버의 'HIT' 컨텐츠가 신선하지 않은 컨텐츠로 판단되면 원본서버로 재사용여부를 묻게 됩니다. 이때 일반적으로 사용하는 부분이 마지막 수정시간(Last-Modified)이고

캐시가 원본서버로 보내는 헤더(HTTP Header)에 다음 [표 2]와 같은 내용을 포함하게 됩니다.

[표 2] 캐싱 컨텐츠 유효성 확인을 위한 헤더 정보

GET http://www.abc.com/image.gif HTTP/1.1 If-Modified-Since: Sun, 8 Sep 2009 19:43:31 GMT

캐시의 If-Modified-Since 헤더를 받은 원본서버는 자신이 가지고 있는 컨텐츠와 비교하여 유효한지에 대해서 판단하게 됩니다. 그리고 사용 가능한 컨텐츠일 경우 캐시에게 컨텐츠를 전송하는 것이 아니라 '304 Not Modified' 라는 간단한 메시지만 보냅니다. 이 응답을 받은 캐시서버는 컨텐츠가 아직 유효하다고 결정, 캐시에 있는 새로운 컨텐츠에 대한 정보(Date, Expires)만 업데이트하게 되고 캐시에 있는 컨텐츠를 클라이언트에게 전송합니다.

만약 304가 아닌 응답 코드를 받았을 경우에는 실제 원본서버로부터 새 컨텐츠를 내려받고 자신의 컨텐츠와 교체한 다음 클라이언트에게 전송합니다. 컨텐츠 재사용이 유효한지 판단할 때는 If-Modified-Since 외에도 ETag 도 있는데 ETag는 서버가 응답할 때 자신의 메타정보를 가공해서 보낸 문자열로서, 캐시는 단지 ETag가 있을 경우에만 요청헤더에 ETag 문자열을 포함해서 보냅니다. 이때 요청 받은 서버는 단순하게 요청헤더에 포함된 ETag의 문자열 비교를 통해서 유효성(Validation)을 판단합니다.

4. PURGE

원본서버의 컨텐츠가 동일한 파일명으로 수정되었을 때 해당 컨텐츠가 CDN 캐시서버에 즉각적으로 교체될 수 있도록 PURGE를 제공하고 있습니다. 기존의 컨텐츠와 파일경로 및 파일이름이 다른 경우에는 해당 되지 않습니다.

4.1. PURGE의 의미

PURGE는 CDN 캐시서버에 저장되어있는 컨텐츠를 삭제하는 툴(tool)이며 원본서버의 컨텐츠가 변경되었을 때 즉각적으로 캐시서버에 반영하기 위한 방법으로 제공되고 있습니다.

캐시서버는 사용자로부터 요청받은 컨텐츠가 저장되어 있지 않을 경우 원본서버로부터 다운로드하여 캐시하도록 되어 있는데 이런 캐시서버의 특성을 이용하여 원본서버에서 변경된 컨텐츠가 캐시서버에 반영될 수 있도록 PURGE를 사용하여 캐시서버의 기존 컨텐츠를 삭제합니다. 만약 원본서버의 컨텐츠를 변경하신 후에 PURGE를 하지 않으면 캐시서버는 갱신주기에 의해서 원본서버의 컨텐츠가 변경되었음을 판단하기 전까지는 캐시서버에 저장된 컨텐츠를 계속 서비스 하게 됩니다.

PURGE는 CDN서비스를 하고 있는 캐시서버의 컨텐츠를 삭제하는 것이며 원본서버의 컨텐츠를 컨트롤 하지는 않습니다. 무분별한 PURGE 사용은 원본서버의 부하로 이어질 수 있으므로 한번에 많은 양의 PURGE 사용은 주의를 요합니다.

5. DNS 설정

ucloud CDN 서비스를 사용하기 위해서는 기존에 사용하시던 서비스 도메인을 CDN 도메인으로 연결 되도록 설정하는 것이 필요합니다. DNS의 솔루션은 현재 여러가지가 사용되고 있지만 본문 에서는 가장 흔히 사용되는 DNS 솔루션인 BIND를 위주로 설명합니다.

DNS의 설정 미숙으로 인한 장애가 발생 할 수 있으니 주의가 필요합니다.

5.1. CNAME 레코드의 의미

CNAME 레코드는 "Canonical NAME"의 약어로서 특정 도메인에 대한 서브도메인(sub domain)을 사용하기 위한 레코드 입니다. 가장 많이 사용하는 A 레코드는 특정 IP에 대해서 서브도메인을 사용하는 방법이지만 CNAME은 IP가 아니라 도메인 또는 동일 zone 파일내의 다른 호스트를 사용하기 위한 방법 이라는 점에서 '별칭' 또는 '별명' 이라는 말로 쓰이기도 합니다.

BIND가 설치된 시스템에서는 아래 [표 5]와 같이 zone 파일을 수정합니다. abc.com 의 도메인에서 www 호스트를 CDN 도메인 'ab01-ab0123.ktics.co.kr' 로 CNAME 레코드를 이용하여 설정한 것입니다. 이 설정이 적용되면 ucloud CDN 서비스로 트래픽이 발생되고 CDN 서비스가 시작 되는 것입니다. CNAME 으로 설정할 CDN 도메인은 CDN 신청하는 웹 페이지에서 'CDN 서비스 신청' 란에서 정보 입력후에 확인 가능합니다.

[표 3] CNAME 레코드 설정 예

```
$TTL 60
                    IN      NS          ns.abc.com.
ns                  IN      A           123.123.123.123
www                 IN      CNAME       ab01-ab0123.ktics.co.kr
abc.com.            IN      A           123.123.123.123
```

TIP) TTL 값이 큰 경우에는 CNAME 변경 후에도 TTL 시간만큼 CDN으로 서비스가 전환되는 시간이 길어 지게 됩니다. TTL 시간을 고려하여 CNAME 설정 전에 TTL을 줄여 놓고 설정을 하시면 서비스 전환이 빠르며 잘못된 설정으로 인한 장애시 복구 시간도 TTL 만큼 짧아 지게 됩니다. 적용후 정상적인 서비스가 된다면 TTL을 이전과 동일하게 설정하시면 됩니다.

5.2. DNS 설정 확인 방법

수정한 zone 파일의 설정이 DNS 동작에 문제가 없는지 확인합니다.

[표 4] zone file syntex 테스트 방법

```
# named-checkzone abc.com /var/named/abc_com.zone
zone localhost/IN: loaded serial 42
OK
```

zone 파일의 syntex 테스트가 문제가 없다면 수정한 zone 파일을 적용한후 dig 명령이나 nslookup 명령을 이용하여 수정된 CNAME 설정이 문제가 없는지 확인을 합니다.

[표 5] 네임서버 설정 확인 예

```
$ dig www.abc.com

; <<>> DiG 9.2.4 <<>> www.abc.com
;; global options:   printcmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 22958
;; flags: qr rd ra; QUERY: 1, ANSWER: 4, AUTHORITY: 2, ADDITIONAL: 2

;; QUESTION SECTION:
;www.abc.com.                    IN      A

;; ANSWER SECTION:
www.abc.com. 10478              IN      CNAME       ab01-ab0123.ktics.co.kr.
ab01-ab0123.ktics.co.kr. 60     IN      CNAME       cdn-ucloud2.ktics.co.kr.
cdn-ucloud2.ktics.co.kr. 0      IN      A           234.234.234.234
cdn-ucloud2.ktics.co.kr. 0      IN      A           234.234.234.235
cdn-ucloud2.ktics.co.kr. 0      IN      A           234.234.234.236

;; AUTHORITY SECTION:
bd-01.ktics.co.kr.          1705    IN      NS          ns2.bd-01.ktics.co.kr.
bd-01.ktics.co.kr.          1705    IN      NS          ns1.bd-01.ktics.co.kr.

;; Query time: 6 msec
;; SERVER: 168.126.63.1#53(168.126.63.1)
;; WHEN: Thu Dec   3 12:13:36 2009
;; MSG SIZE   rcvd: 201
```

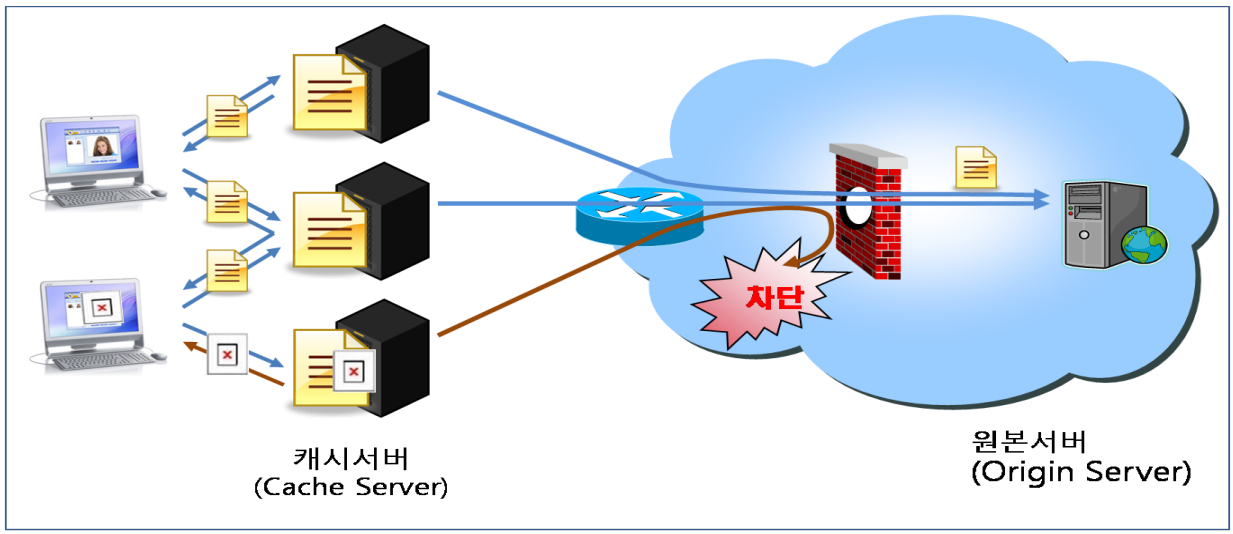
6. 방화벽 ACL 허용 (방화벽 사용 고객에 한함)

방화벽을 사용하는 고객이라면 방화벽에서 CDN 캐시서버 IP를 ACL에 허용해 주셔야 합니다. 이 절차가 빠진 상태에서 서비스를 개통하게 되면 방화벽에서 캐시서버의 요청을 차단해 서비스가 단절되는 문제가 발생할 수 있습니다. IPS 및 웹 방화벽 등 웹 서비스에 영향을 미칠 수 있는 보안 장비에 대해서도 화이트 리스트 등록 또는 ACL 허용이 필요합니다.

간혹, 캐시서비스를 이용하는 시작시기에는 방화벽에서 캐시서버 IP가 허용되도록 설정을 하였지만 ACL의 정책 만료기간이나 방화벽 특성에 따른 탐지오류로 인해 다시 차단되는 경우가 발생할 경우도 있으므로 이러한 우려가 없도록 ACL 허용 설정에 대해 정확한 룰셋 적용을 해 주셔야 합니다.

캐시서버가 웹 서버로 문제없이 연결될 수 있어야 안정적인 웹 캐시서비스를 이용하실 수 있습니다.

[그림 3] 방화벽 차단으로 발생되는 서비스 장애 예



7. CDN 서비스 테스트 방법

CDN 서비스가 정상적으로 서비스 되고 있는지 유무를 파악하기 위해서 테스트 하는 방법을 설명합니다.

리눅스 계열에서 사용할수 있는 curl 과 wget을 이용하는 방법과 윈도우에서 테스트 하실 때는 웹 브라우져에서 직접 url을 입력하여 테스트 할 수 있습니다.

7.1. wget을 이용한 방법

wget을 이용하여 'http://www.abc.com/image/13.jpg' 파일을 요청하는 기본적인 방법을 [표 6]에서 예를 들었습니다. 여러대의 캐시서버중 특정 캐시서버에 요청을 할때는 IP로 요청할 수 있지만 이때는 Host 정보에 정확한 도메인이 있어야 합니다.

[표 6] wget을 이용한 CDN 서비스 테스트 예

```
$ wget -S http://www.abc.com/image/13.jpg
$ wget -S --header='Host: www.abc.com' http://123.123.123.123/image/13.jpg
```

[표 7] wget 테스트 결과 예

```
--13:00:41--  http://123.123.123.123/image/13.jpg
              => `0.jpg'
Connecting to 123.123.123.123:80... connected.
HTTP request sent, awaiting response...
  HTTP/1.0 200 OK
  Date: Thu, 15 Dec 2011 04:00:30 GMT
  Expires: Sat, 14 Jan 2012 04:00:30 GMT
  Server: Apache/2.2.3 (CentOS)
  Last-Modified: Fri, 24 Sep 2010 16:50:13 GMT
  ETag: "288800c-400-491042c1f7b40"
  Accept-Ranges: bytes
  Content-Length: 1024
  Cache-Control: max-age=2592000
  Content-Type: image/jpeg
  X-Cache: HIT from cache.ktics.co.kr
  X-Cache: MISS from i1.cache.ktics.co.kr
  Connection: keep-alive
Length: 1,024 (1.0K) [image/jpeg]
```

응답 헤더 정보에서 200 OK 상태코드를 받아 정상적으로 서비스가 되고 있음을 확인하였다면 정상적으로 서비스가 되고 있음을 판단 할 수 있습니다.

[표 8] 에서는 '400 Bad Request'가 발생하였는데 이는 CDN 서비스를 하는 웹 캐시서버 또는 원본서버에서 잘못된 요청을 받았다고 판단 하였을 경우에 나타나며 서비스가 되지 않는 상황입니다. 주로 잘못된 도메인 설정으로 인해서 발생되는 경우가 많은데 아래와 같은 확인을 하시기 바랍니다.

- wget 사용시 도메인 또는 Host 정보를 정확하게 사용하였는지 확인 하십시오.
- dig를 이용하여 도메인 룩업시 wget에서 사용한 IP가 있는지 확인 하십시오.
- wget을 이용하여 웹 서버로 요청 하여 상태코드를 확인 하십시오

→ 웹서버로 요청시 정상적이라면 ucloud 고객센터로 문의해 주시기 바랍니다.

→ 웹서버에서도 동일한 응답 코드를 받았다면 웹서버의 도메인 설정을 확인 하십시오

[표 8] 비정상적인 응답헤더 정보 예

```
--11:57:35--  http://123.123.123.123/500K/13.jpg
              => `13.jpg'
Connecting to 123.123.123.123:80... connected.
HTTP request sent, awaiting response...
  HTTP/1.0 400 Bad Request
  Server: ics-cache/3.5.1
  Date: Thu, 03 Dec 2009 02:57:35 GMT
  Content-Type: text/html
  Content-Length: 1312
  Expires: Thu, 03 Dec 2009 02:57:35 GMT
  X-Cache-Error: ERR_INVALID_REQ 0
  X-Cache: MISS from i2.cache.ktics.co.kr
  Proxy-Connection: close
11:57:35 ERROR 400: Bad Request.
```

응답헤더는 상황에 따라 다양하게 나타날 수 있으며 응답코드 또한 결과에 따라 다양한 코드로 표기 됩니다. 이에 대해 대표적인 응답헤더와 응답코드를 사전에 파악해 놓으시면 관리에 편리합니다. 응답코드 정보는 인터넷에서 'HTTP response code'를 검색하시면 상세한 내용을 보실 수 있습니다.

7.2. curl을 이용하는 방법

curl을 사용하는 방법도 wget과 옵션의 사용법이 다를 뿐 크게는 차이가 없고 어느 툴을 사용하느냐에 따라서 결과가 달라지는것은 아니지만 보여주는 내용에는 다소 차이가 있으므로 사용이 편리한 툴을 선택하셔서 사용하시면 됩니다.

[표 9] curl을 이용한 CDN 서비스 테스트 예

```
$ curl -v http://www.abc.com/image/13.jpg
$ curl -v -H 'Host: www.abc.com' http://123.123.123.123/image/13.jpg
```

[표 10]은 curl을 이용하여 테스트를 한 결과이다. wget의 결과와는 다소 차이가 있는것으로 보이지만 결과에는 차이가 없고 보여 주는 내용에는 약간의 차이가 있음을 볼 수 있다.

[표 10] curl 테스트 결과 예

```
* About to connect() to 123.123.123.123 port 80
*     Trying 123.123.123.123... * connected
* Connected to 123.123.123.123 (123.123.123.123) port 80
> GET /image/13.jpg HTTP/1.1
User-Agent: curl/7.12.1 (i686-redhat-linux-gnu) libcurl/7.12.1 OpenSSL/0.9.7a zlib/1.2.1.2
libidn/0.5.6
Pragma: no-cache
Accept: */*
Host: image.test.com

< HTTP/1.0 200 OK
< Date: Thu, 15 Dec 2011 04:00:30 GMT
< Server: Apache/2.2.3 (CentOS)
< Last-Modified: Fri, 24 Sep 2010 16:50:13 GMT
< ETag: "288800c-400-491042c1f7b40"
< Accept-Ranges: bytes
< Content-Length: 1024
< Cache-Control: max-age=2592000
< Expires: Sat, 14 Jan 2012 03:55:07 GMT
< Content-Type: image/jpeg
< X-Cache: MISS from cache.ktics.co.kr
< X-Cache: MISS from i2.cache.ktics.co.kr
< Connection: close
* Closing connection #0
```

8. 서비스 가입시 유의 사항

기존에 타사 CDN 서비스를 이용하시다가 ucloud CDN 서비스를 이용하실 때는 원본서버의 요청수와 트래픽이 발생할 수 있음을 유념을 하셔야 합니다.

기존에 타사에서 이용하시던 CDN 서비스에서는 많은 컨텐츠가 캐시된 상태에서 서비스를 하고 있었으므로 원본서버로의 요청을 많이 하지 않았지만 새로운 CDN 서비스를 이용 하실 때는 캐시된 컨텐츠가 없는 상태이므로 CDN 서버가 컨텐츠를 캐시 할 때 까지는 원본서버로 요청과 트래픽이 발생하여 시스템 부하가 발생 할 수 있습니다.

이런 상황으로 서비스의 품질이 저하되는 원인이 될 가능성이 있으므로 기존에 사용량이나 트래픽이 높은 경우 사이즈가 큰 컨텐츠를 사용하셨다면 서비스 신청 이전에 고객센터로 문의해 주시기 바랍니다.

기존에 CDN 서비스를 이용하지 않았던 고객으로서 처음 CDN 서비스를 이용하시는 고객이라면 트래픽과 부하에 대해서는 크게 문제 되지 않습니다. CDN 서비스는 기존에 CDN 서비스를 이용하지 않았을 때 보다 많은 트래픽을 발생하거나 부하를 더 주는 경우가 없습니다.